GEOGRAPHIC INFORMATION SCIENCE

Built for GEOG 370



David Parr



Geographic Information Science

The textbook for GEOG 370

David Parr

2026-11-27



Geographic Information Science

Copyright © 2026 David Parr. All rights reserved.

Published by David Parr. Available at https://daveparr.com/geographic-information-science/

First Edition

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (https://creativecommons.org/licenses/by-nc/4.0/).



For my sweet Abigail, who loves cake, and also insisted that I remind everyone that I can't live without her.

Table of contents

Preface 1
- About this book . . . 1
- How to use this book . . . 1
  - What's in each chapter . . . 1
  - QGIS and ArcGIS Pro . . . 2
  - Key terms and the glossary . . . 2
- How this book is organized . . . 2
- Fonts and reading settings . . . 3
- Acknowledgments . . . 3

I Unit 1: Data Models and Projections 4
- Chapter 1: What Is GIS? . . . 5
- Chapter 2: What Is Geographic Data? . . . 5
- Chapter 3: Location and Coordinate Systems . . . 5
- Chapter 4: Map Projections . . . 5
- Chapter 5: Attribute Data and Queries . . . 6
- Chapter 6: Databases and Table Joins . . . 6
- Chapter 7: Spatial Joins . . . 6

1 What Is GIS? 7
- 1.1 GIS in recent events . . . 7
  - 1.1.1 Wildfires in Los Angeles, January 2025 . . . 7
  - 1.1.2 An earthquake in Tibet, January 2025 . . . 8
  - 1.1.3 Spacing pilgrims in Mecca, 2021 . . . 9
  - 1.1.4 Hurricane Katrina, 2005: a problem in space and time . . . 9
- 1.2 Defining a geographic problem . . . 11
- 1.3 What questions can GIS answer? . . . 11
  - 1.3.1 Where things are: the Deepwater Horizon oil spill, 2010 . . . 11
  - 1.3.2 How much: where Americans live . . . 12
  - 1.3.3 How things have changed: the 2004 Indian Ocean tsunami . . . 12
  - 1.3.4 How to get there: route optimization . . . 12
  - 1.3.5 Five kinds of GIS questions . . . 13
- 1.4 What is GIS? . . . 14
  - 1.4.1 Where geography meets computer science . . . 14
  - 1.4.2 Does GIS mean…? . . . 14
  - 1.4.3 What is a GISystem? . . . 14
  - 1.4.4 Is a map a GIS? John Snow and cholera, 1854 . . . 15
  - 1.4.5 The five components of a GIS . . . 16
  - 1.4.6 Geographic Information Science (GIScience) . . . 16
  - 1.4.7 GIScience or GISystems? . . . 17
  - 1.4.8 Why "geographic"? . . . 17
- 1.5 A brief history of GIS . . . 18
- 1.6 The GIS process . . . 19
- 1.7 Who uses GIS? . . . 19
- 1.8 GIS and society . . . 20

1.9 Our toolbox . . . 20
1.10 What this book will teach you . . . 21
1.11 What are GIS questions? . . . 22
1.12 Check your understanding . . . 22
1.13 Summary . . . 22
1.14 Exercises . . . 23
1.15 Further reading and online resources . . . 23
1.15.1 Books and open textbooks . . . 23
1.15.2 What is GIS? Introductions and overviews . . . 24
1.15.3 Videos . . . 24
1.15.4 Software and tutorials . . . 24
1.15.5 The events in this chapter . . . 25
1.15.6 Where to find geographic data . . . 25
1.15.7 GIS and society . . . 25
1.15.8 Careers and professional organizations . . . 26

2 What Is Geographic Data? 27
2.1 How a map represents the world . . . 27
2.1.1 Entities and features . . . 27
2.1.2 Data abstraction . . . 28
2.1.3 Geographic data models . . . 29
2.2 What types of data do we use? . . . 29
2.2.1 Discrete objects and continuous data . . . 29
2.2.2 Points, lines, polygons, and annotation . . . 30
2.2.3 What is an entity? . . . 30
2.2.4 Spatial and non-spatial data . . . 31
2.2.5 Attribute data . . . 31
2.2.6 Thematic layers . . . 32
2.3 How do data models describe geography? . . . 32
2.3.1 Vector and raster . . . 32
2.3.2 The vector model: one entity, several features . . . 34
2.3.3 Vector terms . . . 35
2.3.4 The raster model: a grid of cells . . . 36
2.3.5 Cell size and cell coordinates . . . 37
2.4 How is geographic data stored? . . . 38
2.4.1 Data structures . . . 38
2.4.2 The Esri shapefile . . . 38
2.4.3 Metadata . . . 39
2.5 Vector or raster? . . . 40
2.6 Check your understanding . . . 41
2.7 Summary . . . 41
2.8 Exercises . . . 41
2.9 Further reading and online resources . . . 42
2.9.1 Books and open textbooks . . . 42
2.9.2 Data models: vector, raster, and TIN . . . 43
2.9.3 File formats . . . 43
2.9.4 Metadata . . . 43
2.9.5 Raster data and resolution . . . 44
2.9.6 Software and tutorials . . . 44
2.9.7 Where to find geographic data . . . 44

3 Location and Coordinate Systems 45
3.1 Where are we? . . . 45

3.2 Modeling the shape of the Earth . . . 46
3.2.1 Geodesy . . . 46
3.2.2 Sphere or ellipsoid? . . . 46
3.2.3 The geoid . . . 48
3.2.4 Three heights . . . 48
3.3 Geodetic datums . . . 49
3.3.1 What is a datum? . . . 49
3.3.2 NAD27 . . . 50
3.3.3 NAD83 . . . 50
3.3.4 WGS84 . . . 50
3.3.5 Datum shift . . . 51
3.4 Geographic coordinates . . . 52
3.4.1 Building a coordinate system . . . 52
3.4.2 Parallels and meridians . . . 52
3.4.3 How far is a degree? . . . 53
3.4.4 Writing degrees . . . 53
3.4.5 Converting between them . . . 54
3.4.6 Precision and location . . . 55
3.5 Lines on the globe . . . 55
3.5.1 Great circles . . . 55
3.5.2 The graticule . . . 56
3.6 From globe to flat map . . . 56
3.7 Putting it together . . . 57
3.8 Check your understanding . . . 58
3.9 Summary . . . 58
3.10 Exercises . . . 59
3.11 Further reading and online resources . . . 59
3.11.1 Books and open textbooks . . . 59
3.11.2 Geodesy, the geoid, and heights . . . 60
3.11.3 Datums . . . 60
3.11.4 Latitude, longitude, and great circles . . . 61
3.11.5 Software and tutorials . . . 61

4 Map Projections 62
4.1 What is a map projection? . . . 63
4.2 Every projection distorts something . . . 64
4.2.1 Four properties . . . 64
4.2.2 Seeing distortion: Tissot's indicatrix . . . 64
4.2.3 Mercator and reality . . . 64
4.2.4 Web Mercator: for display, not measurement . . . 66
4.3 How projections are made . . . 67
4.3.1 Three developable surfaces . . . 67
4.3.2 Tangent and secant . . . 67
4.3.3 Projections are named for what they preserve . . . 68
4.4 Projections for the United States . . . 69
4.4.1 Lambert Conformal Conic . . . 69
4.4.2 Transverse Mercator . . . 69
4.4.3 Universal Transverse Mercator (UTM) . . . 70
4.4.4 The State Plane Coordinate System . . . 71
4.5 Coordinate reference systems and EPSG codes . . . 72
4.6 Choosing a projection . . . 73
4.7 Projections in QGIS and ArcGIS Pro . . . 73
4.8 Check your understanding . . . 74
4.9 Summary . . . 74

4.10 Exercises . . . 74
4.11 Further reading and online resources . . . 75
4.11.1 Books and open textbooks . . . 75
4.11.2 Understanding map projections . . . 75
4.11.3 Mercator, Web Mercator, and the true size of countries . . . 76
4.11.4 UTM, State Plane, and coordinate reference systems . . . 76
4.11.5 Software and tutorials . . . 76

5 Attribute Data and Queries **78**
5.1 Attribute data . . . 79
5.1.1 Attribute tables . . . 79
5.1.2 Data types . . . 80
5.1.3 Why fields have limits: binary . . . 81
5.1.4 Levels of measurement . . . 82
5.2 Querying a table . . . 83
5.2.1 What is a query? . . . 83
5.2.2 SQL . . . 83
5.2.3 Simple and compound queries . . . 84
5.2.4 AND, OR, and NOT . . . 84
5.2.5 Selecting rows also selects features . . . 87
5.2.6 Complex queries and brackets . . . 87
5.2.7 Where you put NOT matters . . . 87
5.2.8 Beyond = and > . . . 88
5.2.9 What to do with a selection . . . 89
5.3 Thinking in sets . . . 91
5.3.1 Every condition names a set . . . 91
5.3.2 What do you get? . . . 92
5.3.3 Each feature is tested separately . . . 92
5.4 Null values and missing data . . . 93
5.5 Check your understanding . . . 94
5.6 Summary . . . 94
5.7 Exercises . . . 95
5.8 Further reading and online resources . . . 95
5.8.1 Books and open textbooks . . . 95
5.8.2 Attribute data, data types, and measurement . . . 96
5.8.3 Queries and SQL . . . 96
5.8.4 Sets and logic . . . 96
5.8.5 Tutorials and data for practice . . . 97

6 Databases and Table Joins **98**
6.1 Databases and keys . . . 98
6.1.1 From one table to many . . . 98
6.1.2 What is a database management system? . . . 99
6.1.3 Relational databases . . . 100
6.1.4 Primary and foreign keys . . . 100
6.2 Joining tables . . . 101
6.2.1 What a join does . . . 101
6.2.2 Tables are bags, not sets . . . 101
6.2.3 Which rows does a join keep? . . . 102
6.2.4 A second example: people and cars . . . 103
6.3 Cardinality . . . 103
6.3.1 Four kinds of relationship . . . 103
6.3.2 The rule of joining . . . 104

6.4 Joining census data in QGIS . . . 105
6.4.1 The task . . . 105
6.4.2 The key: GEOID . . . 105
6.4.3 Step by step . . . 106
6.4.4 Why joins fail . . . 107
6.4.5 Attribute joins and spatial joins . . . 108
6.5 Check your understanding . . . 108
6.6 Summary . . . 108
6.7 Exercises . . . 109
6.8 Further reading and online resources . . . 109
6.8.1 Books and open textbooks . . . 109
6.8.2 Databases . . . 109
6.8.3 Joins in QGIS and ArcGIS Pro . . . 110
6.8.4 SQL joins . . . 110
6.8.5 Census data and keys . . . 110

7 Spatial Joins 111
7.1 What is a spatial join? . . . 111
7.1.1 Location instead of a key . . . 111
7.1.2 Spatial relationships . . . 112
7.2 Four kinds of spatial join . . . 113
7.2.1 Simple or summarized, inside or distance . . . 113
7.2.2 A simple distance join: cities and airports . . . 113
7.2.3 An inside join: septic systems and geology . . . 115
7.3 Cardinality decides simple or summarized . . . 115
7.3.1 The rule of joining, again . . . 115
7.3.2 Which layer is the destination? . . . 115
7.4 Coordinate systems and distance joins . . . 116
7.4.1 Distances come out in the data's units . . . 116
7.4.2 Projecting on the fly is not enough . . . 117
7.5 Spatial joins in practice . . . 117
7.5.1 Six questions to answer first . . . 117
7.5.2 Spatial joins in QGIS . . . 118
7.5.3 Spatial joins in ArcGIS Pro . . . 118
7.5.4 Reading the output . . . 120
7.6 Worked examples . . . 120
7.6.1 Example 1: congressional districts with more than 10 earthquake deaths 120
7.6.2 Example 2: a pollution risk index for rivers . . . 120
7.6.3 Example 3: the closest volcano to each U.S. city . . . 121
7.7 Check your understanding . . . 121
7.8 Summary . . . 121
7.9 Exercises . . . 121
7.10 Further reading and online resources . . . 122
7.10.1 Books and open textbooks . . . 122
7.10.2 Spatial joins in QGIS . . . 122
7.10.3 Spatial joins in ArcGIS Pro . . . 123
7.10.4 Spatial relationships . . . 123
7.10.5 Data for practice . . . 123

II Unit 2: Spatial Data Quality 124
Chapter 8: Census Data . . . 125
Chapter 9: Geocoding and Georeferencing . . . 125
Chapter 10: OpenStreetMap and Data Quality . . . 125

8 Census Data **126**
8.1 Population and place data . . . 126
8.1.1 Authoritative and asserted data . . . 126
8.1.2 Four kinds of population and place data . . . 127
8.2 The U.S. Census . . . 127
8.2.1 The constitutional count . . . 127
8.2.2 The big three surveys . . . 128
8.2.3 Why census data matter . . . 128
8.2.4 The census is not perfect . . . 128
8.3 Census geography . . . 129
8.3.1 The spine . . . 129
8.3.2 Tracts, block groups, and blocks . . . 129
8.3.3 GEOIDs . . . 131
8.4 Census products and variables . . . 131
8.4.1 What is published where . . . 131
8.4.2 ACS 1-year and 5-year estimates . . . 132
8.4.3 Counts and estimates . . . 132
8.4.4 Choosing a product and a geography . . . 133
8.5 Getting census data into a GIS . . . 134
8.5.1 Where to get the data . . . 134
8.5.2 Tutorial: a census join in ArcGIS Pro . . . 134
8.6 Check your understanding . . . 135
8.7 Summary . . . 135
8.8 Exercises . . . 135
8.9 Further reading and online resources . . . 136
8.9.1 Books and guides . . . 136
8.9.2 Census Bureau resources . . . 136
8.9.3 Other sources and tools . . . 137

9 Geocoding and Georeferencing **138**
9.1 Geocoding and address matching . . . 138
9.1.1 An everyday example . . . 138
9.1.2 How U.S. addresses are structured . . . 139
9.1.3 Reference files . . . 139
9.1.4 Linear interpolation along a street . . . 141
9.1.5 Geocoding in practice . . . 142
9.2 Georeferencing . . . 142
9.2.1 Why georeference? . . . 142
9.2.2 Five steps . . . 143
9.2.3 Ground control points . . . 143
9.2.4 Transformations . . . 144
9.2.5 Resampling . . . 145
9.2.6 Residuals and RMSE . . . 145
9.2.7 Georeferencing in ArcGIS Pro . . . 147
9.2.8 Georeferencing in QGIS . . . 148
9.3 Check your understanding . . . 148
9.4 Summary . . . 148
9.5 Exercises . . . 148
9.6 Further reading and online resources . . . 149
9.6.1 Books and guides . . . 149
9.6.2 Geocoding tools and data . . . 149
9.6.3 Georeferencing tools and historic maps . . . 149

10 OpenStreetMap and Data Quality **151**
10.1 Defining data quality . . . 151
10.1.1 Quality is fitness for use . . . 151
10.1.2 Seven elements of quality . . . 152
10.2 Volunteered geographic information . . . 153
10.2.1 Anyone can be a data producer . . . 153
10.2.2 Types of VGI . . . 153
10.2.3 Big Data . . . 154
10.2.4 FAIR data . . . 154
10.3 OpenStreetMap . . . 154
10.3.1 A free, editable map of the world . . . 154
10.3.2 Who uses OpenStreetMap? . . . 155
10.3.3 OSM's data model . . . 155
10.4 Measuring the quality of OSM . . . 156
10.4.1 Extrinsic and intrinsic measures . . . 156
10.4.2 Tools for checking OSM . . . 157
10.5 Crisis mapping . . . 157
10.5.1 Haiti, 2010 . . . 157
10.5.2 The Humanitarian OpenStreetMap Team . . . 158
10.5.3 Nepal, 2015 . . . 158
10.5.4 The problem of success . . . 159
10.6 Getting involved . . . 160
10.7 Check your understanding . . . 160
10.8 Summary . . . 160
10.9 Exercises . . . 161
10.10 Further reading and online resources . . . 161
10.10.1 Readings . . . 161
10.10.2 OpenStreetMap . . . 162
10.10.3 Quality tools . . . 162
10.10.4 Humanitarian mapping . . . 162

III Unit 3: Spatial Analysis **164**
Chapter 11: Data Classing and Choropleth Maps . . . 165
Chapter 12: Density and Neighborhoods . . . 165
Chapter 13: Spatial Analysis: Selection, Buffers, and Overlay . . . 165
Chapter 14: Raster Analysis . . . 165
Chapter 15: Spatial Interpolation . . . 166
Chapter 16: Kriging . . . 166

11 Data Classing and Choropleth Maps **167**
11.1 Choropleth maps . . . 167
11.1.1 Shading areas by value . . . 167
11.1.2 Normalize first . . . 168
11.1.3 Look at the distribution first . . . 169
11.2 How many classes? . . . 169
11.2.1 To class or not to class . . . 169
11.2.2 How many can readers tell apart? . . . 169
11.2.3 Classing is generalization . . . 170
11.3 Where to put the breaks . . . 171
11.3.1 Classing as modeling . . . 171
11.3.2 Equal interval . . . 171
11.3.3 Quantile . . . 172
11.3.4 Arithmetic progression . . . 173

11.3.5 Geometric progression . . . 173
11.3.6 Natural breaks (Jenks) . . . 173
11.3.7 Standard deviation . . . 173
11.3.8 Manual (defined) intervals . . . 174
11.3.9 Matching the method to the data . . . 174
11.4 Measuring the fit . . . 174
11.5 Choropleth maps in a GIS . . . 175
11.5.1 Color schemes . . . 175
11.5.2 ArcGIS Pro . . . 175
11.5.3 QGIS . . . 176
11.6 Check your understanding . . . 176
11.7 Summary . . . 176
11.8 Exercises . . . 177
11.9 Further reading and online resources . . . 177
11.9.1 Books and articles . . . 177
11.9.2 Online resources . . . 177

12 Density and Neighborhoods 179
12.1 Density: things per area . . . 179
12.2 Neighborhoods . . . 180
12.2.1 What is a neighborhood? . . . 180
12.2.2 Tobler's first law . . . 180
12.2.3 By distance . . . 181
12.2.4 By contiguity . . . 181
12.2.5 By number of neighbors . . . 182
12.3 Kernel density estimation . . . 183
12.3.1 From points to a surface . . . 183
12.3.2 Kernel shape . . . 183
12.3.3 Bandwidth . . . 186
12.3.4 Cell size and units . . . 186
12.4 Kernel density in a GIS . . . 186
12.4.1 ArcGIS Pro . . . 186
12.4.2 QGIS . . . 187
12.5 Check your understanding . . . 187
12.6 Summary . . . 187
12.7 Exercises . . . 188
12.8 Further reading and online resources . . . 188
12.8.1 Books and articles . . . 188
12.8.2 Online resources . . . 189

13 Spatial Analysis: Selection, Buffers, and Overlay 190
13.1 What is spatial analysis? . . . 190
13.1.1 A problem: where would Batman live? . . . 190
13.1.2 Spatial operations and cartographic models . . . 191
13.2 Spatial scope . . . 192
13.3 Preparing layers . . . 192
13.3.1 Selection . . . 192
13.3.2 Joins . . . 193
13.3.3 Reclassification . . . 193
13.3.4 Dissolve . . . 193
13.4 Buffers . . . 193
13.4.1 Proximity . . . 193
13.4.2 Line and polygon buffers . . . 194

13.5 Overlay . . . 195
13.5.1 Combining layers by location . . . 195
13.5.2 Clip, intersect, and union . . . 195
13.5.3 Slivers . . . 197
13.6 Raster overlay: map algebra . . . 197
13.7 Other kinds of spatial analysis . . . 198
13.8 Check your understanding . . . 198
13.9 Summary . . . 198
13.10 Exercises . . . 199
13.11 Further reading and online resources . . . 199
13.11.1 Books . . . 199
13.11.2 ArcGIS Pro . . . 199
13.11.3 QGIS . . . 200
13.11.4 Other resources . . . 200

14 Raster Analysis **201**
14.1 Raster basics . . . 201
14.1.1 Cells, cell size, and extent . . . 201
14.1.2 What's in a cell? . . . 201
14.1.3 NoData . . . 202
14.2 Combining rasters . . . 202
14.2.1 Map algebra . . . 202
14.2.2 When the grids don't line up . . . 203
14.2.3 Resampling . . . 204
14.2.4 Cell size, extent, and snap raster . . . 204
14.3 The scope of raster functions . . . 205
14.4 Local functions . . . 205
14.4.1 Mathematical functions . . . 205
14.4.2 Logical functions . . . 206
14.4.3 Reclassification . . . 208
14.4.4 Conditional functions . . . 208
14.5 Beyond local . . . 209
14.5.1 Neighborhood functions . . . 209
14.5.2 The moving window . . . 209
14.5.3 Aggregation . . . 209
14.5.4 Zonal functions . . . 211
14.5.5 Global functions: distance . . . 211
14.6 Check your understanding . . . 211
14.7 Summary . . . 211
14.8 Exercises . . . 212
14.9 Further reading and online resources . . . 212
14.9.1 Books . . . 212
14.9.2 ArcGIS Pro . . . 213
14.9.3 QGIS . . . 213
14.9.4 Data and other resources . . . 213

15 Spatial Interpolation **214**
15.1 What is spatial interpolation? . . . 214
15.1.1 Five steps . . . 214
15.1.2 Exact and inexact interpolators . . . 215
15.2 Step 1: Interpolation methods . . . 216
15.2.1 Nearest neighbor and Thiessen polygons . . . 216
15.2.2 Average neighbor . . . 217
15.2.3 Inverse distance weighted (IDW) . . . 217

15.2.4 Polynomial (trend surface) interpolation . . . 219
15.2.5 Radial basis functions (splines) . . . 220
15.2.6 Comparing the methods . . . 220
15.3 Steps 2 and 3: Neighborhoods and sampling . . . 221
15.3.1 The search neighborhood . . . 221
15.3.2 Sampling design . . . 221
15.4 Step 5: Estimating the error . . . 221
15.4.1 Cross validation . . . 221
15.4.2 Root mean square error . . . 223
15.4.3 Four questions . . . 224
15.5 Interpolation in ArcGIS Pro and QGIS . . . 224
15.6 Check your understanding . . . 224
15.7 Summary . . . 224
15.8 Exercises . . . 225
15.9 Further reading and online resources . . . 225
15.9.1 Books . . . 225
15.9.2 ArcGIS Pro . . . 225
15.9.3 QGIS . . . 226
15.9.4 Data . . . 226

16 Kriging **227**
16.1 Semivariance . . . 227
16.1.1 Measuring dissimilarity with distance . . . 227
16.1.2 Direction matters . . . 228
16.2 The semivariogram . . . 228
16.2.1 Semivariogram models . . . 229
16.3 Kriging . . . 230
16.3.1 A surface in three parts . . . 230
16.3.2 Weights from the semivariogram . . . 230
16.3.3 Anisotropy . . . 231
16.3.4 Trends . . . 232
16.3.5 Types of kriging . . . 233
16.4 Judging the result . . . 233
16.4.1 Cross-validation statistics . . . 233
16.4.2 The kriging workflow . . . 234
16.5 Kriging in ArcGIS Pro and QGIS . . . 234
16.6 Check your understanding . . . 234
16.7 Summary . . . 235
16.8 Exercises . . . 235
16.9 Further reading and online resources . . . 235
16.9.1 Books and papers . . . 235
16.9.2 ArcGIS Pro . . . 236
16.9.3 QGIS and other tools . . . 236

IV GIS Software **237**

17 Getting to Know QGIS **238**
17.1 What is QGIS? . . . 238
17.1.1 Free and open source . . . 239
17.2 A short history . . . 239
17.2.1 Long-term and latest releases . . . 240
17.2.2 Installing QGIS . . . 240

17.3 The QGIS window . . . 241
17.3.1 Navigating the map . . . 241
17.4 Projects and data . . . 242
17.4.1 File formats . . . 242
17.5 Adding data . . . 243
17.5.1 Points from a table of coordinates . . . 243
17.5.2 Basemaps and web services . . . 244
17.6 Entering data . . . 244
17.6.1 Creating a new layer . . . 244
17.6.2 Digitizing features . . . 246
17.6.3 The attribute table . . . 247
17.6.4 Calculating fields . . . 247
17.6.5 Joins . . . 249
17.7 Selecting and querying . . . 249
17.8 Making a basic map . . . 249
17.8.1 Symbology . . . 250
17.8.2 Labels . . . 251
17.8.3 The print layout . . . 251
17.9 The Processing Toolbox . . . 253
17.10 Plugins . . . 254
17.11 Getting help . . . 254
17.12 Check your understanding . . . 255
17.13 Summary . . . 255
17.14 Exercises . . . 255
17.15 Further reading and online resources . . . 256
17.15.1 QGIS documentation . . . 256
17.15.2 Tutorials and courses . . . 256
17.15.3 Community and history . . . 256
17.15.4 Data used in this chapter . . . 257

18 Getting to Know ArcGIS Pro 258
18.1 ArcGIS and ArcGIS Pro . . . 258
18.2 A short history . . . 259
18.3 Getting ArcGIS Pro . . . 260
18.3.1 Licensing . . . 260
18.3.2 Installing and signing in . . . 261
18.4 The ArcGIS Pro window . . . 261
18.4.1 Navigating the map . . . 262
18.5 Projects . . . 262
18.5.1 The geodatabase . . . 263
18.6 Adding data . . . 263
18.6.1 Points from a table of coordinates . . . 263
18.6.2 Basemaps . . . 264
18.7 Entering data . . . 264
18.7.1 Creating a feature class . . . 264
18.7.2 Editing features . . . 264
18.7.3 The attribute table . . . 265
18.7.4 Calculating fields . . . 265
18.7.5 Joins . . . 265
18.8 Selecting and querying . . . 265
18.9 Making a basic map . . . 266
18.9.1 Symbology . . . 266
18.9.2 Labels . . . 266
18.9.3 Layouts . . . 266

18.10 Geoprocessing . . . 267
18.11 Sharing to the web . . . 267
18.12 QGIS and ArcGIS Pro compared . . . 269
18.13 Getting help . . . 269
18.14 Check your understanding . . . 270
18.15 Summary . . . 270
18.16 Exercises . . . 270
18.17 Further reading and online resources . . . 271
18.17.1 ArcGIS Pro documentation . . . 271
18.17.2 Lessons and courses . . . 271
18.17.3 History and community . . . 271

Appendices **272**

Glossary **272**

About the author **288**
About this book . . . 288

References **289**

Preface

This book is an open educational resource textbook designed for my course, GEOG 370, Geographic Information Science, built from the course lectures.

About this book

This is an interactive textbook written for GEOG 370. It's designed for students who are new to GIS and to spatial data. Chapters explain the concepts behind geographic information and show how to put them into practice in QGIS and ArcGIS Pro.

How to use this book

Read the book in order if you can. Each chapter builds on the ones before it, and later chapters assume you have met the ideas and terms introduced earlier. Each unit opens with a short page that says what its chapters cover, so you can see where you are going before you start.

What's in each chapter

Most chapters follow the same pattern, so you know where to look:

- **Learning objectives** at the top list what you should be able to do by the end of the chapter. They are a good checklist when you review for an exam.
- **Numbered sections** teach the material, with maps, figures, and tables. Cross-references such as "Section 1.2" or "Figure 2.1" are links you can follow.
- **Think about it** boxes pose a question to stop and consider before reading on. There is no grade attached: they are there to help the idea stick.
- **Did you know..?** boxes and **Trivia time** questions add a fact or a quick puzzle that connects the idea to the world around you.
- **Check your understanding** questions near the end let you test yourself. You can answer them as many times as you like, and they are not graded.
- A **Summary** recaps the chapter's main points, **Exercises** give you practice, and **Further reading and online resources** points you to books, websites, data, and tutorials that go deeper.

QGIS and ArcGIS Pro

This book teaches with two GIS programs: **QGIS**, which is free and open source, and **ArcGIS Pro**, made by Esri. Chapter 1 introduces both. The ideas in each chapter are the same whichever program you use, and the skills you learn in one carry over to the other.

Names you would type or see in the software, such as file names like `roads.shp`, field names, and expressions, are printed in a monospaced font.

Key terms and the glossary

Important terms are printed in **bold** where they are first explained.

All of the key terms and their definitions are collected in the Glossary at the end of the book.

The books, articles, and documentation the chapters cite are listed together on the References page at the end of the book.

How this book is organized

The chapters are grouped into units. Each unit opens with a page that summarizes its chapters.

Unit 1: Data Models and Projections (Chapters 1–7) lays the foundation. It explains what a geographic information system is, the science behind it, the questions it can answer, and who uses it, then turns to how a GIS stores the world as data, puts it in the right place with datums, coordinate systems, and map projections, asks questions of its attribute tables, and combines layers with attribute and spatial joins.

Unit 2: Spatial Data Quality (Chapters 8–10) is about where data come from and how good they are. It covers census data and their margins of error, turning addresses and scanned maps into GIS layers with geocoding and georeferencing, and the quality of volunteered data in OpenStreetMap.

Unit 3: Spatial Analysis (Chapters 11–16) turns data into answers. It covers classing data for choropleth maps and measuring density with neighborhoods and kernels, the core vector operations of selection, buffers, and overlay, map algebra and the other raster functions, and estimating values between observations with deterministic interpolation and kriging.

GIS Software (Chapters 17 and 18) introduces the two programs used in this course, QGIS and ArcGIS Pro: their history, how each is organized, and how to do common tasks in each, from entering data to making a finished map. Read them early, alongside Unit 1, and come back to them as a reference.

More chapters are being written now and will appear here as they are added.

Fonts and reading settings

The online version of this book sets its text in **Atkinson Hyperlegible Next**, and names you would type into the software in **Atkinson Hyperlegible Mono**. Both were designed by the Braille Institute of America for readers with low vision, and both are free to use. Many common letters and numbers look alike in ordinary fonts: a capital I, a lowercase l, and the number 1, or a capital O and a zero. Atkinson Hyperlegible gives each of these characters its own distinct shape, so they are hard to mistake for one another. That matters in a book full of coordinates, file names, and field names, where one misread character puts a point in the wrong place or points the software at a file that doesn't exist. The zero in the monospaced font has a slash through it, so you can always tell 0 from O.

This PDF is set in **Ubuntu**, a clear sans-serif font designed by Dalton Maag and free to use. A font that is easy for one reader can be hard for another. The online version of this book, at https://daveparr.com/geographic-information-science/, lets you change the font and text size with the **Aa** button beside the search box. The choices include Lexend, designed to reduce visual stress, and OpenDyslexic, which some readers with dyslexia find easier to read.

Acknowledgments

I'd like to acknowledge the hard work of my students, without whom this would not be possible.



Part I

Unit 1: Data Models and Projections

This unit lays the foundation for everything else in the book. It starts with the big picture: what a geographic information system is, the science behind it, and the questions it can answer. It then turns to how a GIS stores the world as data, and how coordinate systems and map projections put that data in the right place on a round Earth. By the end of the unit you can open, explore, edit, query, and join spatial data in QGIS and ArcGIS Pro.

Chapter 1: What Is GIS?

GIS helps people decide *where*: where a fire is spreading, where rescuers should search, and how to get help there. This chapter looks at GIS in recent events, the four questions that define a geographic problem, and the kinds of questions a GIS can answer. It defines a geographic information system and its five components, separates GISystems from GIScience, and traces the history of GIS, who uses it, and the ethical questions it raises. It ends with the tools used in this book, QGIS and ArcGIS Pro, and how to write a GIS question of your own.

Chapter 2: What Is Geographic Data?

How do you store a tree, a road, or a mountain in a computer? This chapter follows the steps that turn the real world into data: entities become features, and features are stored in a data model, a data structure, and finally machine code. It separates discrete objects from continuous data, and spatial data from attributes, then introduces the two main data models, vector and raster, with the vocabulary of each, from nodes and vertices to cell size and resolution. It ends with how data are stored in files such as the shapefile, why metadata matters, and how to choose between vector and raster.

Chapter 3: Location and Coordinate Systems

How do you say exactly where something is on the Earth? This chapter starts with relative and absolute location, then models the Earth's shape: Eratosthenes's measurement, the ellipsoid, and the geoid, with the three heights a GPS and a map report. It explains geodetic datums (NAD27, NAD83, and WGS84) and why the same coordinates can land 100 meters apart in two of them. It ends with latitude and longitude: how they are built, how to write and convert them, how precise they need to be, and why great circles and the graticule matter.

Chapter 4: Map Projections

Every flat map distorts the Earth. This chapter shows how a projection flattens latitude and longitude onto a plane, a cylinder, or a cone, and what each kind keeps and gives up: shape, area, distance, or direction. It explains why Greenland looks as big as Africa on a Mercator map, why web maps are for display and not measurement, and how scale error grows away from standard lines. It then covers the systems used most in North Carolina, UTM and State Plane, reads EPSG codes, and gives a practical method for choosing a projection.

Chapter 5: Attribute Data and Queries

Geometry says where; attributes say what. This chapter looks at the attribute table, its records and fields, the data types that store them and the binary limits behind them, and the levels of measurement that decide which questions make sense. It then teaches attribute queries in SQL, from simple comparisons to AND, OR, NOT, brackets, LIKE, BETWEEN, and IN, with a way to check any query using sets and Venn diagrams, and the trap of NULL values.

Chapter 6: Databases and Table Joins

Most GIS data live in databases. This chapter explains what a database management system does, how relational databases link tables with primary and foreign keys, and how codes such as FIPS codes and GEOIDs serve as keys. It then teaches table joins: the destination and source tables, inner, left, right, and full joins, and how cardinality decides what a join can do, with ways to handle one-to-many relationships. It ends with a census join in QGIS, and the problems that most often break a join.

Chapter 7: Spatial Joins

When two layers share no key, location can link them. This chapter explains spatial predicates such as intersects, within, and contains, and the four kinds of spatial join, from simple inside joins to summarized distance joins. Worked examples match North Carolina towns to their nearest airports and septic systems to the geology beneath them. It covers cardinality, why distances need a projected coordinate system, six questions to plan any spatial join, and the tools that do the work in QGIS and ArcGIS Pro.

1 What Is GIS?

Learning objectives

By the end of this chapter you should be able to:

1. Explain how maps and spatial data are used in real events, such as wildfires, earthquakes, and hurricanes.
2. Describe a geographic problem by its scale, time, space, and purpose.
3. Name the kinds of questions a GIS can answer, and recognize a GIS question when you see one.
4. Define a geographic information system, and name its five components.
5. Explain the difference between GISystems and GIScience, and why we say *geographic* rather than *spatial*.
6. Outline the history of GIS, who uses it today, and the ethical questions it raises.

Before you read on, take a moment to answer a question: *what do you think a GIS is?* Write your answer down. You may think of an app on your phone, a map on the news, or software you have heard of at work. All of those are part of the answer, but none of them is the whole answer. We will come back to your answer at the end of the chapter.

This first chapter looks at the big picture. It starts with recent events where maps and spatial data mattered, then defines what a geographic information system is, the science behind it, and the people who use it. It ends with the tools you will use in this book, and the kinds of questions you will be able to answer by the end of it.

1.1 GIS in recent events

The easiest way to see what GIS does is to watch it at work. In each of the events below, people had to make fast decisions about *where*: where the danger is, where people are, and where to send help.

1.1.1 Wildfires in Los Angeles, January 2025

On January 7, 2025, two wildfires broke out in Los Angeles County and spread quickly in strong Santa Ana winds. The Palisades Fire burned through Pacific Palisades on the west side of the city, and the Eaton Fire burned through Altadena and the hills above Pasadena. Together they destroyed more than 16,000 homes and other buildings (Figure 1.1), among the most destructive fires in California's history.

Think about how maps and spatial data would be used during an event like this:

- **Fire perimeters and spread forecasts.** Firefighters map the edge of the fire several times a day, from aircraft and satellites, and models forecast where wind will push it next.



Figure 1.1: The Palisades Fire seen from Palisades Drive in Pacific Palisades, January 7, 2025, the day it started. Photo: Ariam23, Wikimedia Commons, CC BY 4.0, cropped.

- **Evacuation zones and routes.** Emergency managers divide the area into zones and tell each one when to leave, and which roads to take.
- **Shelters and resources.** Maps show where shelters, water, fuel, and fire engines are, and how to get them where they are needed.
- **Damage assessment.** After the fire passes, satellite and aerial imagery show which buildings were damaged or destroyed.

Damage assessment is a good example of GIS at work. Smoke and cloud block ordinary cameras, but **radar** satellites send out their own microwave signal, which passes through smoke. Researchers compared images from the European Space Agency's Copernicus Sentinel-1 radar satellite taken before and after the Eaton Fire, and mapped which buildings in Altadena had changed. Each damaged building became a shape on a map, published within days by *The New York Times* and by NASA's Disasters program. Insurers, the Federal Emergency Management Agency (FEMA), utilities, residents, and rebuilding offices could all use maps like these to plan aid and recovery before crews could safely reach every street.

1.1.2 An earthquake in Tibet, January 2025

On the same day, January 7, 2025, a magnitude 7.1 earthquake struck Dingri County in Shigatse, Tibet, close to the Nepal border. The shaking was felt in Nepal, Bhutan, and India, and the earthquake killed more than 120 people. In the first hours after an earthquake, rescue teams have one urgent question: *where should we search first?*

The region has been hit before. On April 25, 2015, the magnitude 7.8 Gorkha earthquake struck central Nepal, killing nearly 9,000 people and destroying or damaging hundreds of thousands of homes, temples, and historic buildings in and around Kathmandu (Figure 1.2).



Figure 1.2: Collapsed shrines in the Pashupatinath Temple area of Kathmandu after the April 2015 Nepal earthquake. Photo: Nabin K. Sapkota, Wikimedia Commons, Creative Commons license.

No single map answers that. You need to combine several layers of information: where the epicenter was and how strongly the ground shook in each place, where the buildings are and how they were built, how many people live in each village, and which roads through the mountains are still open. Putting layers like these together, to find the places that meet several conditions at once, is one of the oldest and most important things a GIS does. The U.S. Geological Survey publishes a ShakeMap and an estimate of likely losses (PAGER) for large earthquakes worldwide within an hour or so, for exactly this purpose.

1.1.3 Spacing pilgrims in Mecca, 2021

Not every geographic problem is a disaster. During the COVID-19 pandemic, Saudi Arabia limited the Hajj pilgrimage in 2021 to about 60,000 vaccinated residents, compared with more than two million pilgrims in a typical year. Pilgrims at the Grand Mosque in Mecca walked around the Kaaba on marked paths, spaced apart (Figure 1.3).

Planning that spacing is a spatial problem. How many people fit safely in a space? Where should the markers and paths go? How do crowds move through the entrances and exits? The same questions come up when planning a football game, a graduation, or a concert on your own campus.

1.1.4 Hurricane Katrina, 2005: a problem in space and time

Hurricane Katrina made landfall on the Gulf Coast on August 29, 2005 (Figure 1.4). Its storm surge, ocean water pushed ashore by the storm, flooded the Mississippi coast and much of New Orleans, and the storm killed well over a thousand people. A satellite view of the storm in the Gulf shows *before*; a map of the flooded coast shows *after*. The same place looks completely different from one week to the next.

The storm, the surge, and the recovery all happened in specific places at specific times. That is what makes Katrina a *geographic* problem: it has a **where** and a **when**.



Figure 1.3: Pilgrims spaced apart around the Kaaba at the Grand Mosque in Mecca, July 2021, with barriers keeping the crowd small. Photo: AhmanElq, Wikimedia Commons, Creative Commons license.

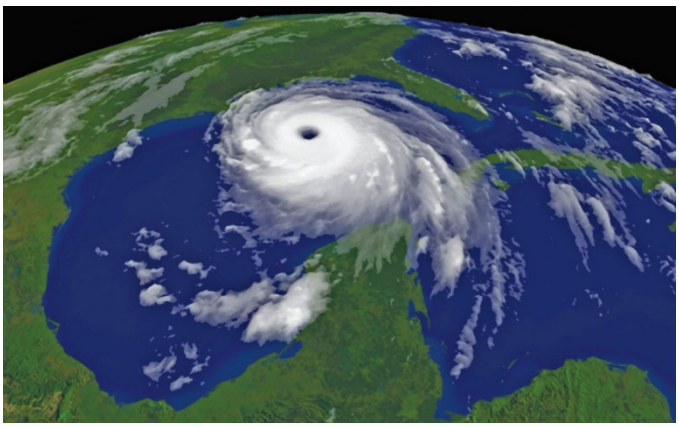


Figure 1.4: Hurricane Katrina in the Gulf of Mexico, August 2005. A geographic problem has a location, an extent, and a time. Image: NASA.

1.2 Defining a geographic problem

Four questions help describe any geographic problem. Table 1.1 applies them to Katrina.

Table 1.1: Four questions that define a geographic problem, applied to Hurricane Katrina.

Question	What it asks	Katrina
Scale	Does the problem look different at different levels of geographic detail?	One flooded home, one county, the whole Gulf Coast
Time	Are the consequences short-term or long-term?	Hours of storm surge; years of rebuilding
Space	Does the problem affect a small area or a large one?	Damage stretched across several states
Purpose	Are we advancing scientific knowledge, or solving a practical problem?	Better storm surge models, or faster evacuations

Scale matters because a decision made for a whole region may be wrong for a single neighborhood, and the data you need change with it: a county-level flood map can't tell a family whether their street will flood. Time matters because some problems last hours and others decades, and because the data you have describe one moment. Space matters because large problems need data from many sources that must fit together. Purpose matters because research and practice ask for different kinds of answers: a scientist wants to know *why* the surge reached so far inland, and an emergency manager wants to know *who must leave now*.

Think about it

Apply the four questions to the Los Angeles fires or the Tibet earthquake. At what scales does the problem matter? How long will its consequences last? How large an area does it cover? Who would use a map of it, and for what purpose?

1.3 What questions can GIS answer?

GIS helps answer a few broad kinds of questions. One way to remember them is: *where things are, how much is there, how things change, and how to get there.*

1.3.1 Where things are: the Deepwater Horizon oil spill, 2010

On April 20, 2010, the Deepwater Horizon drilling rig exploded in the Gulf of Mexico, killing 11 workers. Oil flowed from the well for 87 days, until it was capped in July, making it the largest marine oil spill in U.S. history. At its largest, oil on the surface covered tens of thousands of square miles of the Gulf.

The simplest GIS question is *where is it?* Agencies mapped the oil every day, from satellites, aircraft, and ships, and used those maps to close fishing areas in its path (at the peak, more than a third of the federal waters in the Gulf), to protect beaches and marshes, and to send cleanup crews where they were needed. Knowing where the oil was, and where currents would carry it next, drove almost every decision.

1.3.2 How much: where Americans live

GIS can also show *how much* of something is in each place. A map of U.S. population density, the number of people per square mile or square kilometer, shows at a glance that most Americans live in a few metropolitan areas: along the East Coast from Boston to Washington, around the Great Lakes, in Florida, Texas, and California. Large parts of the interior West are nearly empty (Figure 1.5). Maps like this are made from U.S. Census data, which you will map yourself in Unit 2.

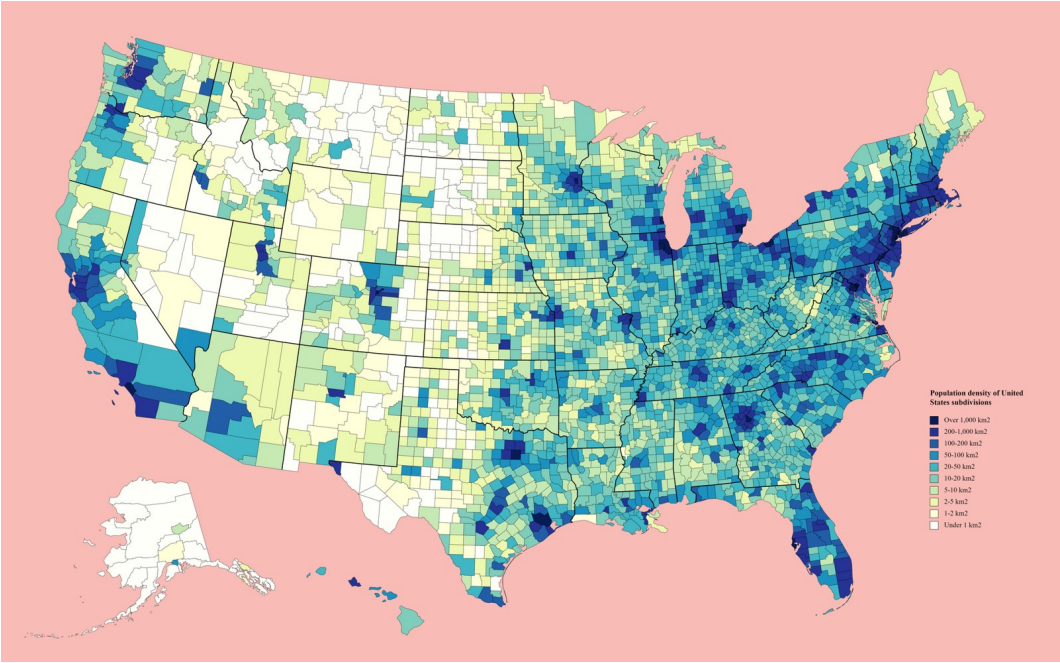


Figure 1.5: Population density of U.S. counties. Each county is shaded by its number of people per square kilometer (the legend's "km2"), from under 1 (white) to over 1,000 (darkest blue). Map: Austiger, Wikimedia Commons, Creative Commons license.

1.3.3 How things have changed: the 2004 Indian Ocean tsunami

On December 26, 2004, a magnitude 9.1 earthquake off the coast of Sumatra, Indonesia, set off a tsunami that crossed the Indian Ocean and killed about 230,000 people in more than a dozen countries. Satellite images of the same coastline before and after the tsunami show where beaches, buildings, roads, and trees were swept away (Figure 1.6).

Comparing images or maps of one place at two times is called **change detection**. It measures what was lost, and it guides rebuilding and planning for the next disaster. The same method tracks slower changes too: forests cut or regrown, cities spreading into farmland, glaciers retreating, and coastlines eroding.

1.3.4 How to get there: route optimization

Beyond mapping, GIS solves problems. **Route optimization** is one you probably use every day. When you ask your phone for the fastest route to Miami, the software treats roads as a **network** of lines and intersections, each road segment with a length and a travel time, and

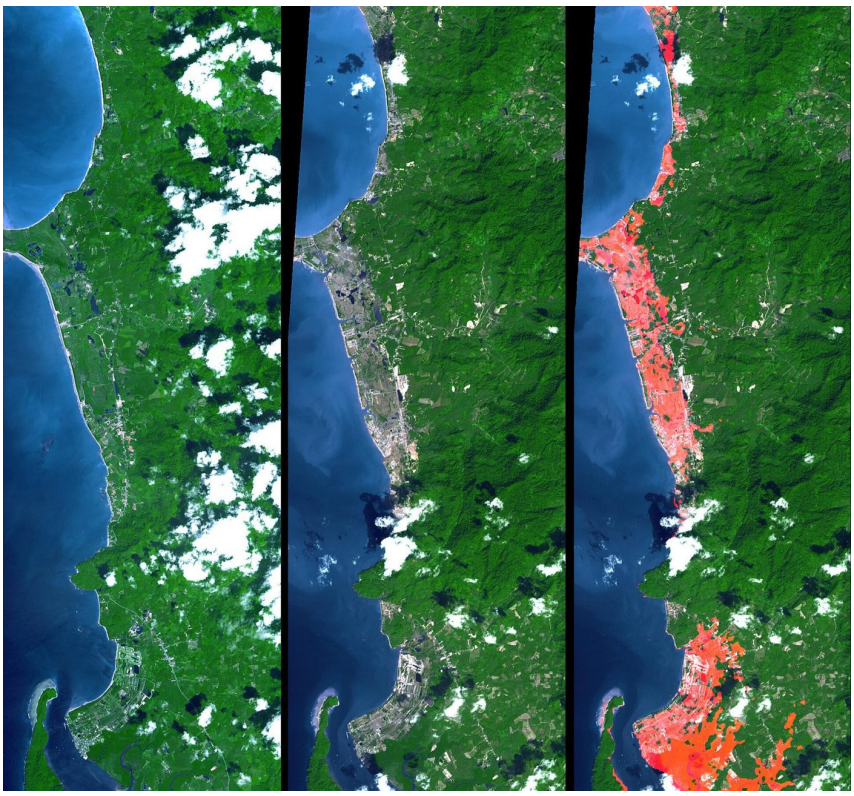


Figure 1.6: The coast north of Phuket, Thailand, seen by satellite in 2000 (left), in 2002 (middle), and after the December 2004 tsunami (right), where red marks the land the tsunami flooded and damaged. Comparing images like these is change detection. Images: NASA, public domain, via Wikimedia Commons.

searches for the path with the lowest total cost. (Most routing software uses descendants of an algorithm that the computer scientist Edsger Dijkstra published in 1959.)

What did your phone need to know to answer? The road network, which roads connect where, speed limits, turn restrictions, live traffic, and your own location from GPS. The same **network analysis** plans delivery routes that visit many stops, measures how long an ambulance or fire engine takes to reach each neighborhood, and helps choose where to put a new store, clinic, or fire station, a task called **site selection**.

1.3.5 Five kinds of GIS questions

A widely used list from the early GIS literature sorts the questions a GIS can answer into five kinds (Table 1.2).

Table 1.2: Five kinds of questions a GIS can answer.

Kind	The question	Example
Location	*What is at* this place?	Who owns the parcel at 123 Main Street?
Condition	*Where is* the place that meets these conditions?	Where is there flat, forested land within 1 km of a road and not in a floodplain?
Trends	*What has changed* here since then?	How much forest was lost in the Blue Ridge after Hurricane Helene?
Patterns	*What spatial patterns* exist?	Do cancer cases cluster near a particular factory?

Kind	The question	Example
Modeling	*What if…?*	What happens to traffic if this bridge closes? Where will the fire spread?

1.4 What is GIS?

1.4.1 Where geography meets computer science

So what is GIS? It sits where two fields overlap (Figure 5.8). **Geography** asks where things are, and why they are there. **Computer science** gives us the data structures, algorithms, and software to store and analyze large amounts of data. GIS is the overlap: using computers to capture, store, analyze, and display geographic data.

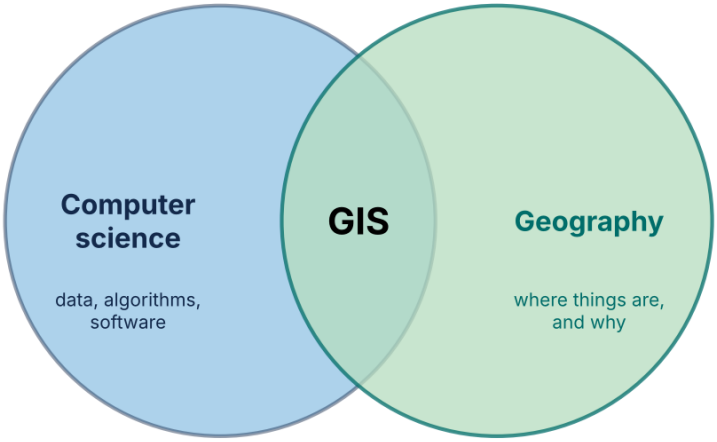


Figure 1.7: GIS is where geography and computer science overlap.

1.4.2 Does GIS mean…?

Ask a room of people what GIS stands for, and you'll get more than one answer:

- **Geographic Information Systems**: the tools. The software, hardware, and data for working with maps and spatial data. This is what most people mean.
- **Geographic Information Science**: the theory and methods behind the tools.
- **Guaranteed Income Source**: a joke, but with some truth in it. GIS skills are in demand in government, business, the environment, health, and planning (see Section 1.7).

This book covers the first two.

1.4.3 What is a GISystem?

There are many definitions of a geographic information system. One of the clearest comes from Paul Bolstad's textbook, *GIS Fundamentals* (Bolstad 2022):

> "A GIS is a computer-based system to aid in the collection, maintenance, storage, analysis, output, and distribution of spatial data and information."

Two parts of the definition are worth slowing down for. First, **spatial data**: data tied to a location on Earth, such as the address of a house, the outline of a county, or the elevation at a point. Second, the six verbs, *collect, maintain, store, analyze, output,* and *distribute*. They describe the whole life of spatial data, from the moment it is measured to the moment someone uses a map of it to make a decision. You will practice each of them in this book.

Other definitions stress different things. Esri, the company that makes ArcGIS, calls GIS a system that creates, manages, analyzes, and maps all types of data, connecting data to a map. Others stress that a GIS is a way of *thinking* about problems, not just software. All of them agree on the core: data with a location, and a computer to work with it.

1.4.4 Is a map a GIS? John Snow and cholera, 1854

In late August and September 1854, cholera killed more than 500 people within a few blocks of Broad Street, in the Soho district of London. Most doctors at the time believed that cholera spread through bad air, a "miasma". The physician John Snow suspected contaminated water instead.

Snow marked each death at its home address on a street map, along with the locations of the public water pumps where people got their drinking water (Figure 1.8). The deaths clustered around the pump on Broad Street. Snow persuaded local officials to remove the pump's handle on September 8, and the outbreak, already waning, ended. His map, published in 1855 (Snow 1855), became a classic example of spatial analysis, and Snow is remembered as one of the founders of **epidemiology**, the study of how diseases spread through populations.



Figure 1.8: John Snow's map of cholera deaths in Soho, London, 1854 (detail). Each dot is a death; each × is a public water pump. Map: John Snow, 1855, public domain.

So, is Snow's map a GIS? It has spatial data (deaths and pumps, each at a location), it has analysis (comparing the pattern of deaths with the pumps), and it led to a decision (remove the handle). But it has no computer. By Bolstad's definition, which says *computer-based*, it is not. By the spirit of the definition, it does everything a GIS does. Decide for yourself, and notice that the *thinking* behind Snow's map, combining layers of information to find a pattern, is the same thinking you will use in GIS software.

1.4.5 The five components of a GIS

A geographic information system is more than software. It has five components, arranged around data (Figure 16.5).

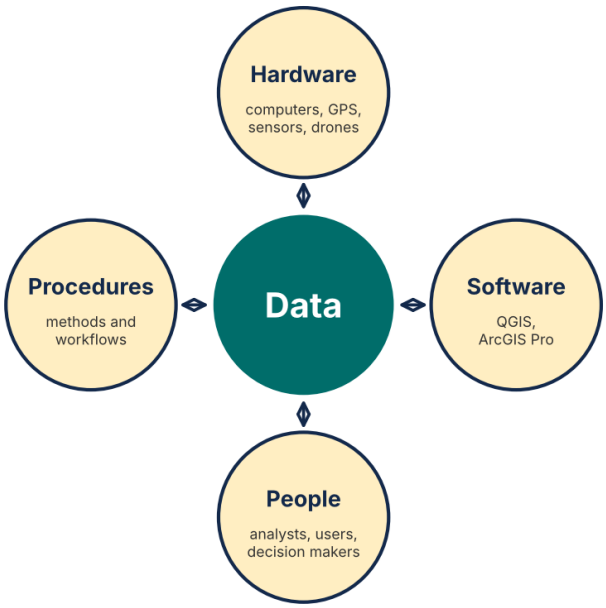


Figure 1.9: The five components of a GIS, with data at the center.

- **Data** are the center of every GIS. Without good data, nothing else matters. Data include the spatial features themselves and their **attributes**, the facts we know about each one, and **metadata**, the record of who made the data, when, and how.
- **Hardware** can be as simple as pen and paper, or as complex as GPS units, drones, satellites, sensors, laptops, and servers in the cloud.
- **Software** includes programs such as QGIS and ArcGIS Pro, but also the methods and analysis the programs carry out.
- **People** are often the most important part, and the most overlooked: the analysts who build the data and maps, the users who read them, and the decision makers who act on them.
- **Procedures** are the agreed methods and workflows that make results reliable and repeatable, so that two analysts given the same data and the same question get the same answer.

1.4.6 Geographic Information Science (GIScience)

Geographic information science, or **GIScience**, is the theory of GIS: the research behind the tools. The geographer Michael Goodchild coined the term in 1992, arguing that the hard

problems of GIS were scientific questions in their own right, not just questions of software (Goodchild 1992). GIScience covers:

- **Analysis techniques**: how to measure distance, find clusters, model surfaces, and predict change.
- **Data management, storage, and entry**: how to represent the real world in a computer, and how to keep data accurate.
- **Visualization and cartography**: how maps and graphics communicate, and how they can mislead.
- **GIS and society**: ethics, privacy, equity, and who is and isn't represented in the data.

GIScience asks how we can improve GISystems, so the tools of tomorrow come from the science of today.

1.4.7 GIScience or GISystems?

What is the difference, and what kinds of problems does each solve?

Table 1.3: GISystems and GIScience compared.

	GISystems	GIScience
What it is	The tools that store, analyze, and display spatial data	The questions and methods behind the tools
Kind of problem	Practical tasks	Research questions
Wildfire example	Mapping today's fire perimeter	Building a better model of where fires will spread
Hurricane example	Mapping evacuation routes	Improving storm surge forecasts

The two depend on each other. Every tool in a GISystem started as a research question in GIScience, and the practical problems people bring to GISystems raise new questions for GIScience.

1.4.8 Why "geographic"?

Why do we say *geographic* information, rather than just *spatial* information?

- **Spatial** can mean any space. Computer-aided design (**CAD**) software is spatial: it measures shapes and distances in a drawing of a building or an engine part. But those drawings aren't tied to places on Earth.
- **Geographic** means locations on or near Earth's surface. Geographic data come with coordinates, which place them on a round Earth, and **map projections**, which flatten it onto a screen or a page. Handling those is a big part of what makes GIS different from other graphics software, and it is the subject of later chapters.
- **Geospatial** is often used to mean the same thing as geographic, especially in industry, the military, and government. The U.S. National Geospatial-Intelligence Agency (NGA) is one example.

1.5 A brief history of GIS

People have made maps for thousands of years, and overlaying maps to find patterns is much older than computers. But GIS as we know it began in the 1960s, when computers first became powerful enough to store maps (Table 1.4).

Table 1.4: Some milestones in the history of GIS.

When	What happened
1854	John Snow maps cholera deaths and water pumps in London.
1960s	Roger Tomlinson leads the **Canada Geographic Information System** (CGIS), built to inventory land across Canada. Tomlinson coins the term *geographic information system* and is often called the father of GIS.
1965	Howard Fisher founds the Harvard Laboratory for Computer Graphics and Spatial Analysis, whose software and graduates shape the field for decades.
1969	Jack and Laura Dangermond found Esri in Redlands, California. Landscape architect Ian McHarg's book *Design with Nature* (McHarg 1969) popularizes **overlay analysis**: stacking transparent maps of soils, slope, water, and land use to find the best places to build.
1972	The first Landsat satellite launches, beginning the longest continuous record of Earth from space.
1980s	Desktop and workstation GIS software spreads, including Esri's ARC/INFO (1982) and the open-source GRASS GIS, first developed by the U.S. Army Corps of Engineers.
1990	The U.S. Census Bureau's TIGER files put every street in the country into digital form for the 1990 census.
1992	Michael Goodchild names the field of geographic information science.
1995–2000	The **Global Positioning System** (GPS) becomes fully operational, and in 2000 the U.S. turns off the deliberate error that had limited civilian accuracy.
2002	Gary Sherman begins writing Quantum GIS, now QGIS.
2004–2005	OpenStreetMap is founded, and Google launches Google Maps and Google Earth, putting interactive maps in front of hundreds of millions of people.
2007 onward	Smartphones with GPS make nearly everyone a producer of location data.
2015	Esri releases ArcGIS Pro.
2020s	Cloud platforms, drones, and artificial intelligence (**GeoAI**) process satellite imagery and map features at a scale no team of people could.

Two threads run through this history. One is the move from expensive mainframes run by specialists to software on every laptop and maps on every phone. The other is the explosion of data: from a few hand-digitized maps to daily satellite images of the whole planet, and billions of location records from phones and cars.

1.6 The GIS process

Using a GIS to solve a problem is a cycle that connects people, data, and the real world (Figure 1.10).

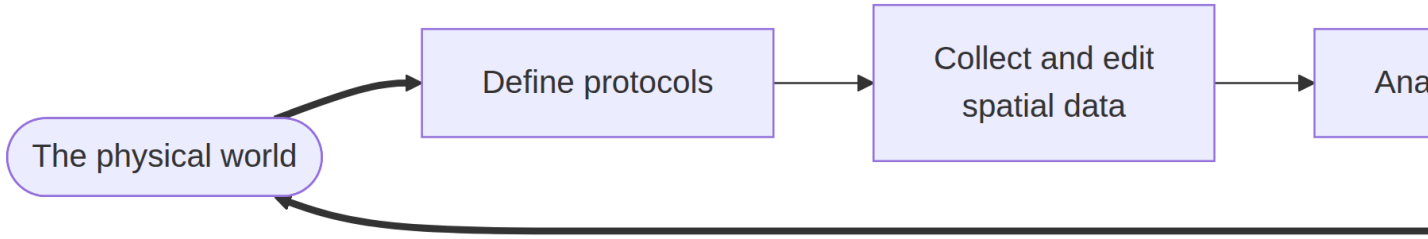


Figure 1.10: The GIS process as a cycle. Data come in from the physical world, and decisions go back out to it. Those two links are where GIS meets the real world.

1. **Define protocols.** Decide what to measure, how, where, and to what accuracy. Good procedures here save trouble later.
2. **Collect and edit spatial data.** Gather data from the field, from GPS, from imagery, or from existing sources, and clean it.
3. **Analyze.** Use the GIS to answer the question: overlay layers, measure distances, find clusters, or model change.
4. **Report results.** Make maps, tables, and reports that communicate what you found.
5. **Decide and act.** People use the results to make decisions, which change the physical world, and the cycle starts again.

The two links to the physical world (the thick arrows) are where GIS connects to reality: data come *in* from the world, and decisions go back *out* to it. Everything in between is an interaction between computers, people, and data.

1.7 Who uses GIS?

GIS is used in almost every sector of the economy (Table 1.5).

Table 1.5: Some of the sectors that use GIS.

Sector	Examples
Business	Banking, insurance, logistics, media, marketing, real estate, retail, economics
Government and safety	Politics, government, homeland security, military, emergency management, law enforcement
Health, education, and culture	Health care, public health, education, research, libraries, museums, archaeology
Environment and resources	Agriculture, forestry, conservation, restoration, biology, mining, energy exploration
Infrastructure and utilities	Transportation, engineering, electricity, gas, water, wastewater, telecommunications, surveying

A few examples of what that work looks like: a county planner checks a proposed subdivision against floodplain maps; a public health analyst maps which neighborhoods are more than 30 minutes by bus from a clinic; an insurer prices policies by wildfire risk; a retailer chooses

where to open its next store; a utility tracks every pipe and valve in its water system; a conservation group maps habitat corridors for black bears; a newsroom maps where public money was spent.

Some people work as GIS analysts, technicians, or developers, but many more use GIS as one skill among others in a job with a different title: planner, epidemiologist, ecologist, engineer, or market analyst. The U.S. Department of Labor's Geospatial Technology Competency Model describes the skills employers look for, and the GIS Certification Institute offers a professional certification, the **GISP**, for experienced practitioners.

> **Think about it**
>
> Find your own major, or the career you hope to have, in Table 1.5. If it isn't there, how might location still matter in that field?

1.8 GIS and society

Maps are powerful, and they can do harm as well as good. GIScience takes this seriously, and so should anyone who makes maps.

- **Privacy.** Your phone records where you go, and that data is bought and sold. A few location points are often enough to identify a person, find where they live and work, and reveal visits to a clinic, a place of worship, or a protest. Mapping individuals' data, even with names removed, can expose them.
- **Equity and representation.** Data describe some places and people better than others. Rural areas, poorer neighborhoods, and countries with fewer resources are often mapped in less detail, so they are easier to overlook in decisions. Volunteer projects such as the Humanitarian OpenStreetMap Team map buildings and roads in places that commercial maps miss.
- **Maps can be used to discriminate.** In the 1930s, the federal Home Owners' Loan Corporation drew maps grading city neighborhoods for lending risk, outlining many Black and immigrant neighborhoods in red. This practice, now called **redlining**, shaped where banks lent money for decades, and its effects can still be seen in home values, health, and tree cover in the same neighborhoods today.
- **Maps can mislead.** Choices about color, classes, scale, and what to leave out can make the same data tell very different stories. A good analyst makes those choices honestly and explains them.

Professional organizations such as the GIS Certification Institute and URISA publish codes of ethics for GIS professionals. A simple test to start with: who could be helped by this map, who could be harmed by it, and who was left off it?

1.9 Our toolbox

This book uses two desktop GIS programs (Table 1.6).

Table 1.6: The two desktop GIS programs used in this book.

	QGIS	ArcGIS Pro
Made by	A volunteer community (the QGIS project, part of OSGeo)	Esri
Cost	Free and **open source**	Commercial; universities often provide student licenses
Runs on	Windows, Mac, and Linux	Windows only
Where you'll see it	Governments, nonprofits, universities, and anyone on a budget	Most U.S. government agencies and many companies

Many employers use ArcGIS Pro, so it is worth knowing both. The good news is that the skills transfer: **layers**, attribute tables, coordinate systems, and geoprocessing tools work the same way in both programs, and once you know one, you can learn the other quickly. When a chapter gives step-by-step instructions, it gives them for both.

GIS doesn't stop at desktop software. You will also meet web maps (ArcGIS Online, Google Maps, OpenStreetMap), cloud platforms for satellite imagery (Google Earth Engine), and programming languages with GIS libraries (Python and R). The concepts in this book apply to all of them.

1.10 What this book will teach you

By the end of the book, you will be able to:

1. **Use GIS software** to design, develop, and carry out a GIS for a scientific spatial research project.
2. **Understand the concepts behind GIS**, including data models, map projections, coordinate systems, measurement, and data types.
3. **Think critically about spatial data**: how it is collected and used, who it affects, and the ethics of working with it.

The book follows the course's three units, each ending with a project (Table 1.7).

Table 1.7: The three units of the course.

Unit	Modules	Project
1. Data Models and Projections	Getting started with GIS; working with geographic information	Editing and querying spatial data
2. Spatial Data Quality	Census data; georeferencing	Mapping Census data
3. Spatial Analysis	Density analysis; measurements; weather data	Your own GIS project

Unit 1 builds the foundation: how a GIS stores the world as data, and how coordinate systems and **data models** put that data in the right place. Unit 2 is about data quality: where data come from, how to judge whether they are good enough, and how to bring old maps and new data into a GIS. Unit 3 puts it all to work with spatial analysis, and ends with a project of your own.

1.11 What are GIS questions?

A GIS question is one where the answer depends on location. Here are five from North Carolina. The key word in each names the kind of question.

Table 1.8: Examples of GIS questions.

Question	Key word	Kind of analysis
Where are there higher clusters of theft and burglary in Chapel Hill?	Where	Mapping and finding clusters
Are incomes **different** between Durham and Raleigh?	Difference	Comparing places
What location **factors** contribute to housing prices in Charlotte?	Factors	Explaining a pattern
How has forest cover **changed** in the Blue Ridge Mountains after Hurricane Helene?	Change	Comparing data over time
Which neighborhoods have less **accessibility** to hospitals?	Accessibility	Distance and travel time

Good GIS questions have a few things in common. They are specific about *what* is being measured and *where*. They can be answered with data that exist, or that you can collect. And the answer would be useful to someone: a planner, a business, a public health department, or the people who live there.

Thought experiment: a GIScience project in your field

What is a geographic information science project related to your own area of study or interest? GIS is used in business, computer science, environmental science, geography, law, geology, real estate, public health, the social sciences, urban planning, communications and journalism, government, transportation, and much more.
Take two minutes and write down one idea. Then answer three questions about it: What is the location question? What data would you need? Who would use the answer? Ideas like this can grow into your final project in Unit 3.
If you're stuck, here are three to start from. Public health: which neighborhoods can't reach a clinic by bus in 30 minutes? Real estate: what explains differences in housing prices between neighborhoods? Journalism: where in the state was disaster relief money spent?

1.12 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

1.13 Summary

- Maps and spatial data guide decisions in real events, such as wildfires, earthquakes, hurricanes, and crowd planning: where the danger is, where people are, and where to send help.

- A **geographic problem** can be described by its **scale**, **time**, **space**, and **purpose**.
- GIS answers questions about **where things are**, **how much** is in each place, **how things change**, and **how to get there**, and more generally about location, condition, trends, patterns, and "what if" models.
- A **geographic information system** is "a computer-based system to aid in the collection, maintenance, storage, analysis, output, and distribution of spatial data and information." It has five components: **data**, **hardware**, **software**, **people**, and **procedures**.
- **GIScience** is the theory and methods behind the tools. GISystems solve practical problems; GIScience asks research questions and improves the tools.
- *Geographic* means tied to places on or near Earth's surface, unlike other *spatial* data such as CAD drawings. *Geospatial* is a common synonym.
- GIS grew from the Canada Geographic Information System and university labs in the 1960s to free software, GPS, and maps on every phone today.
- GIS is used in almost every sector, and maps can harm as well as help: think about privacy, equity, and honesty whenever you make one.
- This book uses **QGIS** and **ArcGIS Pro**. Their concepts and skills transfer to each other and to other GIS tools.

Now look back at the answer you wrote down at the start of the chapter: *what do you think a GIS is?* Has your answer changed?

1.14 Exercises

1. **GIS in the news.** Find a news story from the past year in which a map or spatial data played a role. What was the geographic question? Describe the problem using the four questions in Table 1.1.
2. **Layers for a decision.** An earthquake strikes a mountain region. List five layers of data you would want in order to decide where rescuers should search first, and say where you might find each one.
3. **Is it a GIS?** Is a paper road atlas a GIS? A weather app on your phone? A spreadsheet of customer addresses? Use Bolstad's definition and the five components to argue each case.
4. **The five kinds of questions.** Write one question of each kind in Table 1.2 (location, condition, trends, patterns, modeling) about your hometown or campus.
5. **Systems or science?** For a topic you care about, give one example of a GISystem task and one example of a GIScience research question.
6. **Ethics.** Your phone's location history for the past month is put on a map. What could someone learn about you from it? Who should be allowed to see it?
7. **Your project.** Write your idea from the thought experiment as one clear GIS question. Underline its key word (where, different, factors, change, accessibility, or another), list the data you would need, and say who would use the answer.

1.15 Further reading and online resources

1.15.1 Books and open textbooks

- *GIS Fundamentals: A First Text on Geographic Information Systems*, by Paul Bolstad (Bolstad 2022). A thorough introduction to the whole field, and the source of this chapter's definition of a GIS.

- *Understanding GIS through Sustainable Development Goals*, by Paul Holloway (Holloway 2023). Chapter 1, "Introduction and Context," introduces GIS through the United Nations' Sustainable Development Goals.
- *Essentials of Geographic Information Systems*, by Jonathan Campbell and Michael Shin. A free, open textbook: https://open.umn.edu/opentextbooks/textbooks/67, also on LibreTexts at https://geo.libretexts.org/Bookshelves/Geography_(Physical)/Essentials_of_Geographic_Information_Systems_(Campbell_and_Shin).
- *Intro to GIS and Spatial Analysis*, by Manuel Gimond. Free online notes that explain GIS concepts clearly: https://mgimond.github.io/Spatial/.
- *The Nature of Geographic Information*, by David DiBiase and colleagues at Penn State. A free online text on how geographic data are made and used: https://www.e-education.psu.edu/natureofgeoinfo/.
- *Learning to Think Spatially*, by the National Research Council (National Research Council 2006). Why spatial thinking matters, and how GIS supports it. Free to read at https://nap.nationalacademies.org/catalog/11019.
- *Design with Nature*, by Ian McHarg (McHarg 1969). The classic book on overlay analysis for planning.
- *The Ghost Map*, by Steven Johnson (2006). A readable history of John Snow and the 1854 Broad Street cholera outbreak.

1.15.2 What is GIS? Introductions and overviews

- Esri, "What is GIS?": https://www.esri.com/en-us/what-is-gis/overview, and "The history of GIS": https://www.esri.com/en-us/what-is-gis/history-of-gis.
- National Geographic Education, "GIS (Geographic Information System)": https://education.nationalgeographic.org/resource/geographic-information-system-gis/.
- U.S. Geological Survey, "What is a geographic information system (GIS)?": https://www.usgs.gov/faqs/what-geographic-information-system-gis.
- QGIS, *A Gentle Introduction to GIS*: https://docs.qgis.org/latest/en/docs/gentle_gis_introduction/.
- The **GIS&T Body of Knowledge**, from the University Consortium for Geographic Information Science (UCGIS): short, peer-reviewed articles on hundreds of GIS topics, written for students. https://www.ucgis.org/site/gis-t-body-of-knowledge.
- Michael Goodchild's paper that named the field, "Geographical information science" (Goodchild 1992).

1.15.3 Videos

- *Geospatial Revolution*, a video series from Penn State Public Media (WPSU) on how digital mapping changes the way we live, including the episode *Mapping the Pandemic*: https://www.geospatialrevolution.psu.edu/ and https://wpsu.psu.edu/digital/geospatial-revolution.

1.15.4 Software and tutorials

- QGIS, free and open source: https://qgis.org. Install the Long Term Release. The *QGIS Training Manual* is at https://docs.qgis.org/latest/en/docs/training_manual/.
- ArcGIS Pro: get started at https://pro.arcgis.com/en/pro-app/latest/get-started/get-started.htm. Free, guided lessons are at Learn ArcGIS, https://learn.arcgis.com.

- OSGeo, the Open Source Geospatial Foundation, home of QGIS, GRASS GIS, and other open-source GIS projects: https://www.osgeo.org.
- Google Earth Engine, for analyzing satellite imagery in the cloud: https://earthengine.google.com.
- geojson.io, a simple website for drawing features on a map and seeing the data behind them: https://geojson.io.

1.15.5 The events in this chapter

- NASA Disasters program, maps and data for the January 2025 Southern California wildfires, including Sentinel-1 damage maps: https://disasters.nasa.gov/what-we-do/disasters/disasters-activations/southern-california-wildfires-jan-2025.
- U.S. Geological Survey Earthquake Hazards Program, with ShakeMaps and PAGER loss estimates for earthquakes worldwide: https://earthquake.usgs.gov.
- Copernicus Emergency Management Service, rapid maps of disasters around the world: https://emergency.copernicus.eu.
- NOAA Office of Response and Restoration, on the Deepwater Horizon oil spill: https://response.restoration.noaa.gov.
- National Hurricane Center, tropical cyclone reports, including Hurricane Katrina: https://www.nhc.noaa.gov.
- UCLA Department of Epidemiology, "John Snow": a large collection of material on Snow and his maps, https://www.ph.ucla.edu/epi/snow.html.

1.15.6 Where to find geographic data

- NC OneMap, North Carolina's geospatial data portal: https://www.nconemap.gov.
- U.S. Census Bureau, TIGER/Line shapefiles and other mapping files: https://www.census.gov/geographies/mapping-files.html.
- USGS EarthExplorer, for Landsat and other satellite imagery: https://earthexplorer.usgs.gov.
- NASA Earthdata: https://www.earthdata.nasa.gov.
- FEMA Flood Map Service Center: https://msc.fema.gov.
- Natural Earth, free world map data at small scales: https://www.naturalearthdata.com.
- OpenStreetMap, the free, editable map of the world: https://www.openstreetmap.org.

1.15.7 GIS and society

- *Mapping Inequality: Redlining in New Deal America*, from the University of Richmond's Digital Scholarship Lab: https://dsl.richmond.edu/panorama/redlining/.
- Humanitarian OpenStreetMap Team, which maps places in need for disaster response and development: https://www.hotosm.org. Anyone can volunteer to map.
- GIS Certification Institute, Code of Ethics: https://www.gisci.org/Portals/0/Ethics/CodeOfEthics_PR.pdf.

1.15.8 Careers and professional organizations

- U.S. Bureau of Labor Statistics, *Occupational Outlook Handbook*, "Cartographers and Photogrammetrists": https://www.bls.gov/ooh/architecture-and-engineering/cartographers-and-photogrammetrists.htm.
- GIS Certification Institute, which awards the GISP: https://www.gisci.org.
- URISA, a professional association for GIS in government and planning: https://www.urisa.org.
- American Association of Geographers: https://www.aag.org.
- University Consortium for Geographic Information Science (UCGIS): https://www.ucgis.org.
- GIS Day, held every November, with events on campuses around the world: https://www.gisday.com.

2 What Is Geographic Data?

Learning objectives

By the end of this chapter you should be able to:

1. Explain how a map abstracts the real world, turning entities into features.
2. Describe the steps of data abstraction, from the real world to a data model, a data structure, and machine code.
3. Tell discrete objects from continuous data, and spatial data from attribute data.
4. Name the four main geographic data models, and describe the vector and raster models and their terms.
5. Work out a raster cell's coordinates from its row, column, and cell size, and explain how cell size sets resolution.
6. Explain how a shapefile stores a vector layer, and what metadata records.
7. Weigh the advantages and disadvantages of the vector and raster models, and choose one for a task.

Start with a scenario. Your new GIS consulting firm has been hired to make a map of campus for new students. *What data would you collect? How would you collect it?* (And what would you name your firm?) Take two minutes and make a list. You might write down buildings, sidewalks, parking lots, bus stops, and the dining halls. Keep your list: we will come back to it several times in this chapter.

Whatever is on your list, you made choices. You didn't list every tree, every bench, or every crack in the sidewalk, and you didn't say *how* a building should be drawn. This chapter is about those choices: how we turn the messy, continuous real world into data a computer can store and analyze. It answers five questions. How does a map represent the world? What types of data do we use? How do data models describe geography? How is geographic data stored? And what are the advantages and disadvantages of each type?

2.1 How a map represents the world

2.1.1 Entities and features

Start with the real world. An **entity** is a real-world object: a tree, a building, a light post, a sidewalk, a lawn. A **feature** is how a map represents an entity, usually as a point, a line, or a polygon (Figure 2.1).

Notice what the map leaves out. The building's windows, the color of its bricks, the shape of the tree's crown, and the people walking by are all gone. Every map is a set of choices about what to keep, what to leave out, and how to draw what is kept. Look back at your campus list: which entities made it onto your map, and which did you leave off without even thinking about it?

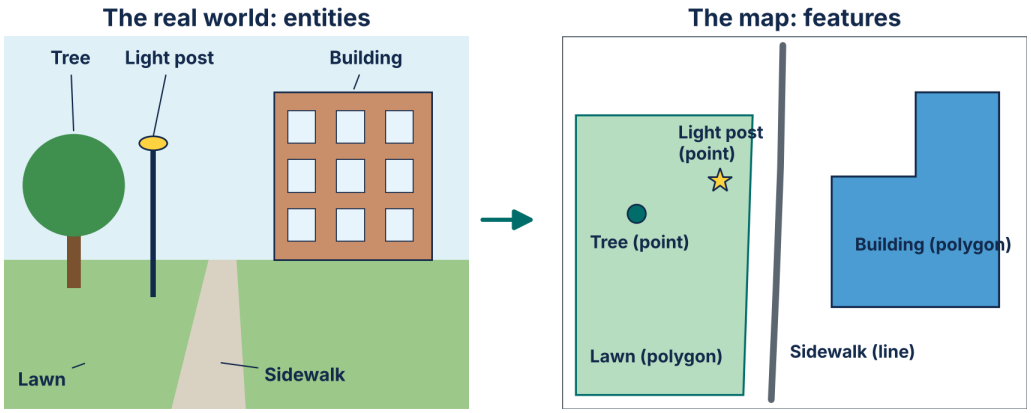


Figure 2.1: The same scene as entities in the real world (left) and as features on a map (right). The tree and light post become points, the sidewalk a line, and the building and lawn polygons.

2.1.2 Data abstraction

A computer can't store a tree. It can only store numbers, and in the end, only ones and zeros. **Data abstraction** is the chain of steps that turns the real world into something a computer can store (Figure 2.2):

1. **The real world**: the wetland, with its water, plants, and soggy edges.
2. **A data model**: the way *people* think about it in a GIS, such as polygons for each wetland and a table that gives each one an ID, an area, and a type.
3. **A data structure**: the way the model is laid out in a computer, such as a list of the *x, y* coordinate pairs around each polygon's edge.
4. **Machine code**: the ones and zeros the computer actually stores.

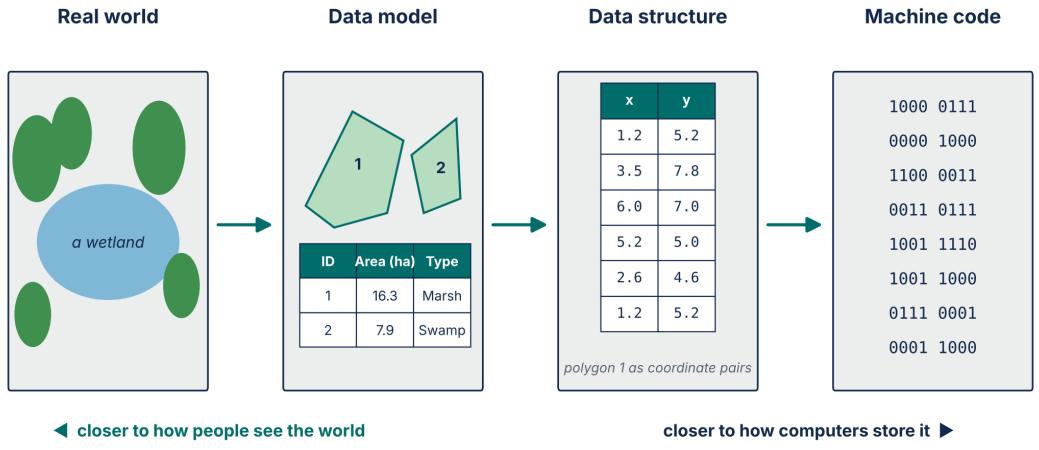


Figure 2.2: Data abstraction. Each step to the right looks less like the world to a person and suits the computer better.

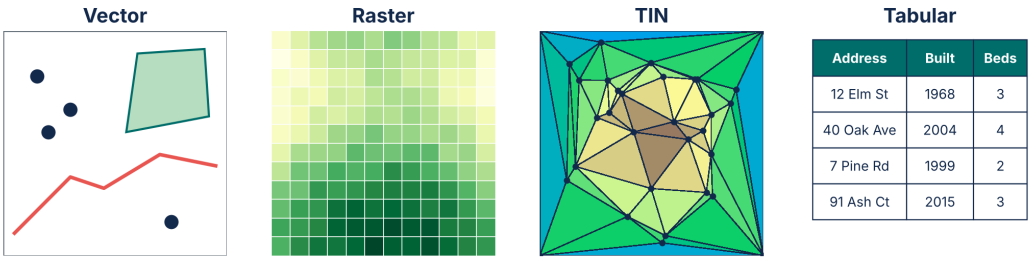
Humans work with data models; computers store data structures and binary code. The good news is that GIS software hides the right-hand steps. When you open a layer in QGIS or ArcGIS Pro, you see polygons and tables, not lists of coordinates, and certainly not ones and zeros. But knowing the steps are there helps you understand why files are organized the way they are, why some operations are slow, and why the same data can be stored in more than one way.

2.1.3 Geographic data models

A **data model** sits where human understanding and computer understanding overlap. People think about roads, rivers, and parcels; computers handle numbers and coordinates. The data model is the meeting point. Paul Bolstad, in *GIS Fundamentals*, describes a data model as the objects in a spatial database plus the relationships among them, and says its purpose is "to provide a formal means of representing and manipulating spatially-referenced information" (Bolstad 2022). *Formal* here means mathematical: a data model has exact rules, so a computer can follow them without guessing.

GIS uses four main data models (Figure 2.3):

- **Vector**: points, lines, and polygons, each stored as coordinates.
- **Raster**: a grid of cells, each holding a value.
- **TIN** (triangulated irregular network): triangles that connect sampled elevation points into a surface.
- **Tabular**: tables of attributes, linked to features.



Address	Built	Beds
12 Elm St	1968	3
40 Oak Ave	2004	4
7 Pine Rd	1999	2
91 Ash Ct	2015	3

Figure 2.3: The four main geographic data models: vector, raster, TIN, and tabular.

Most of this chapter is about vector and raster, the two models you will use most. A TIN is a special model for surfaces such as terrain: it puts more triangles where the ground is rough and fewer where it is flat, so it can be more efficient than a grid. Tables aren't a picture of the world on their own, but they hold the attributes that make vector and raster data useful.

2.2 What types of data do we use?

2.2.1 Discrete objects and continuous data

There are two ways of thinking about what is on a map (Figure 2.4).

Discrete objects exist in one place and not in another. They have clear boundaries, and at any location they are either present or absent: a road is either here or it isn't. Roads, houses, lakes, and state boundaries are discrete objects.

Continuous data describe a quantity that exists across the whole map. It is always present, and its value changes from place to place. Every point on Earth has an elevation, a temperature, and an amount of rainfall, even if the value is zero. You can't point to where elevation "ends."

Some things can be thought of either way, and the choice depends on your question. Is a forest discrete or continuous? A forest boundary drawn as a polygon is discrete: you are either inside the forest or not. But the percent of tree canopy over each patch of ground is

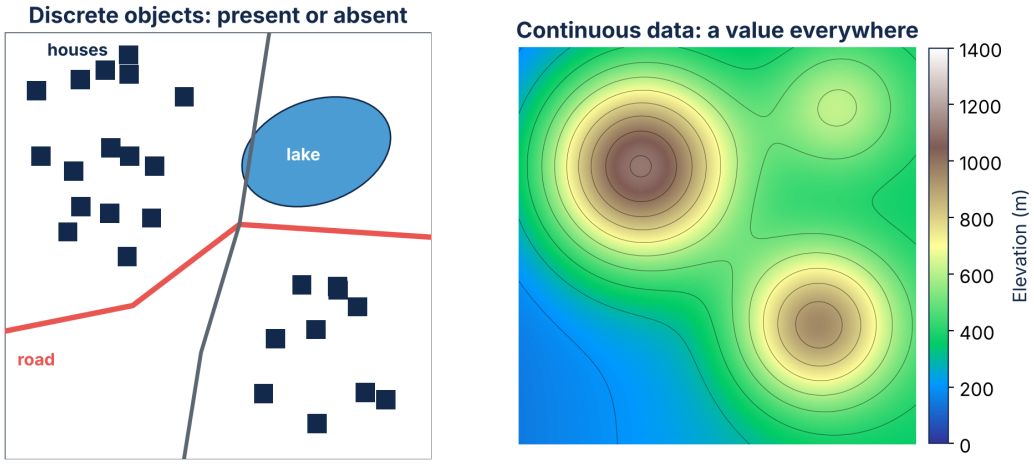


Figure 2.4: Discrete objects (left) are present or absent: roads, houses, and a lake. Continuous data (right) have a value everywhere: here, elevation in meters, with contour lines.

continuous, and it shades gradually from dense forest to open field. As you will see, this choice largely decides whether you use the vector or the raster model.

> **Think about it**
>
> Sort these into discrete objects and continuous data: fire hydrants, air pollution, county boundaries, snow depth, rivers, soil moisture, noise from an airport. Which ones could be either, depending on the question?

2.2.2 Points, lines, polygons, and annotation

Maps portray the world with four kinds of features (Figure 2.5):

- A **point** is a location with no area, such as a school or a well.
- A **line** has length but no area, such as a road or a stream.
- A **polygon** encloses an area, such as a lake or a park.
- **Annotation** is the text placed on a map, such as place names and street names.

The same entity can be any of these, depending on **scale**. A city is a point on a map of the United States, and a polygon on a map of its county. A river is a line on a state map, and a polygon on a detailed map of its banks. You will see this idea again in the vector model, below.

2.2.3 What is an entity?

Every entity carries five kinds of information. Take a lake as an example:

- **A spatial location**: where it is, as an address or coordinates. The lake's center is at a particular latitude and longitude.
- **An extent**: how big it is, its size, boundary, and dimensions. The lake covers 40 hectares.

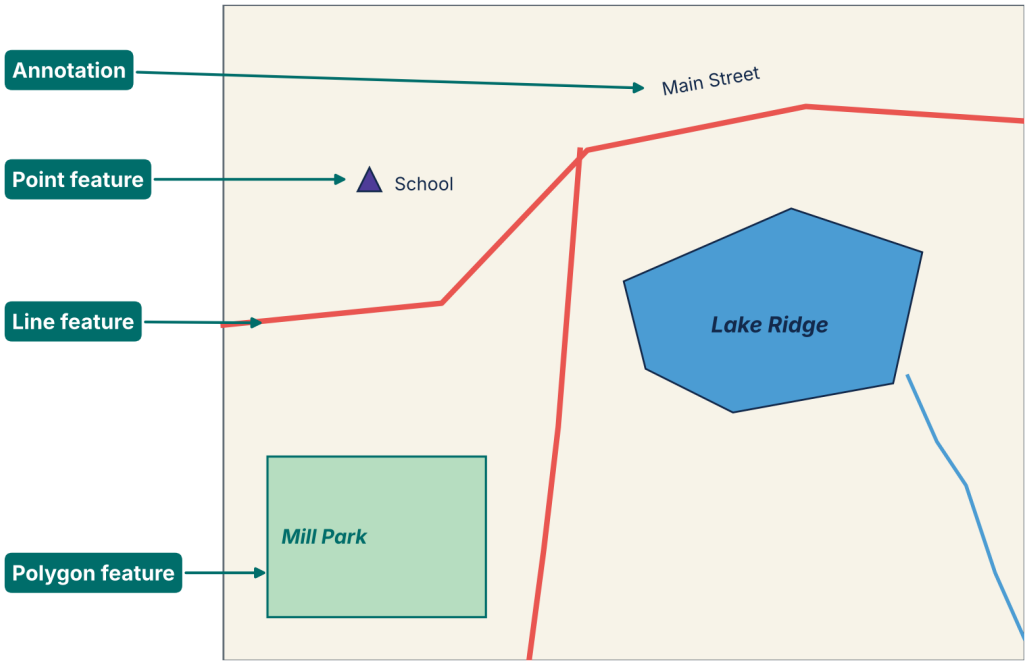


Figure 2.5: The four kinds of map features: points, lines, polygons, and annotation.

- **Non-spatial attributes**: facts that describe it without saying where it is, such as a name, a number, a quality, or a color. The lake is called Lake Ridge, and it is 12 meters deep.
- **Time**: when it existed, and when the data were collected. The shoreline was mapped from aerial photos taken in 2023, and a dry summer could move it.
- **A geometry**: its shape. The lake is stored as a polygon.

2.2.4 Spatial and non-spatial data

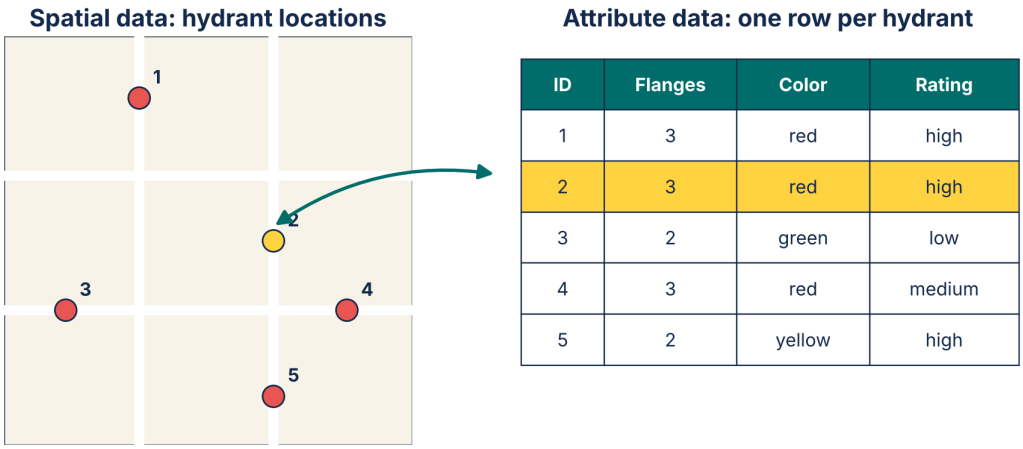
Spatial data have a location that can be put on a map: the buildings on your campus, the rainfall measured across North Carolina, the U.S. highway network. **Non-spatial data** have no location of their own: the parts of a bicycle, a list of student ID numbers, your class grades.

When non-spatial information describes a spatial entity, we call it an **attribute**. A building's name and the year it was built are attributes of the building's polygon. Quick check: is a student's home address spatial or non-spatial? On its own it is just text, but once software matches it to a location on a map (a step called **geocoding**), it becomes spatial.

2.2.5 Attribute data

Attribute data describe features. They have no location on their own; they borrow the location of the feature they are linked to. Attributes are usually stored in a table, with one row per feature and one column per attribute. Every feature in a layer has the same set of attributes, even if some values are blank.

Take a layer of fire hydrants (Figure 2.6). Each dot on the map is one hydrant, and each row of the table describes one of those dots: its ID, how many flanges it has (the outlets hoses connect to), its color, and its condition rating. Select hydrant 2 on the map, and its row lights up in the table; select the row, and the hydrant lights up on the map.



ID	Flanges	Color	Rating
1	3	red	high
2	3	red	high
3	2	green	low
4	3	red	medium
5	2	yellow	high

Figure 2.6: Fire hydrants as spatial data (left) and attribute data (right). Each row of the table describes one hydrant on the map.

That consistency, the same columns for every feature, is what lets a GIS **query** a layer (find every hydrant rated low) and **symbolize** it (color each hydrant by its rating).

2.2.6 Thematic layers

A **layer** groups similar features, such as all the roads, all the parcels, or all the hydrants. Stacking layers that cover the same area lets us compare themes and find where they meet (Figure 2.7). This is the overlay idea from Chapter 1, put into practice.

The two programs in this book organize layers slightly differently:

- **ArcGIS Pro**: the layers you display together make up a *map*. A project (an `.aprx` file) can hold several maps, along with layouts and other items. (Older ArcMap called a map a *data frame*, and saved its work as a *map document*, an `.mxd` file.)
- **QGIS**: layers belong to a *project* (a `.qgz` file), and are drawn in the *map canvas*. The Layers panel lists them, top to bottom, in the order they are drawn.

In both programs, a layer *points to* data stored somewhere else, in a file or a database. It does not hold the data itself. Deleting a layer from your map does not delete the data, and moving the data can break the layer. Keep that in mind when you move a project to another computer.

Think about it

Go back to your campus map list. Name three layers you would need. For each one, would its features be points, lines, or polygons? What attributes would you give them?

2.3 How do data models describe geography?

2.3.1 Vector and raster

The two main data models represent the world in very different ways (Figure 2.8; Table 2.1).

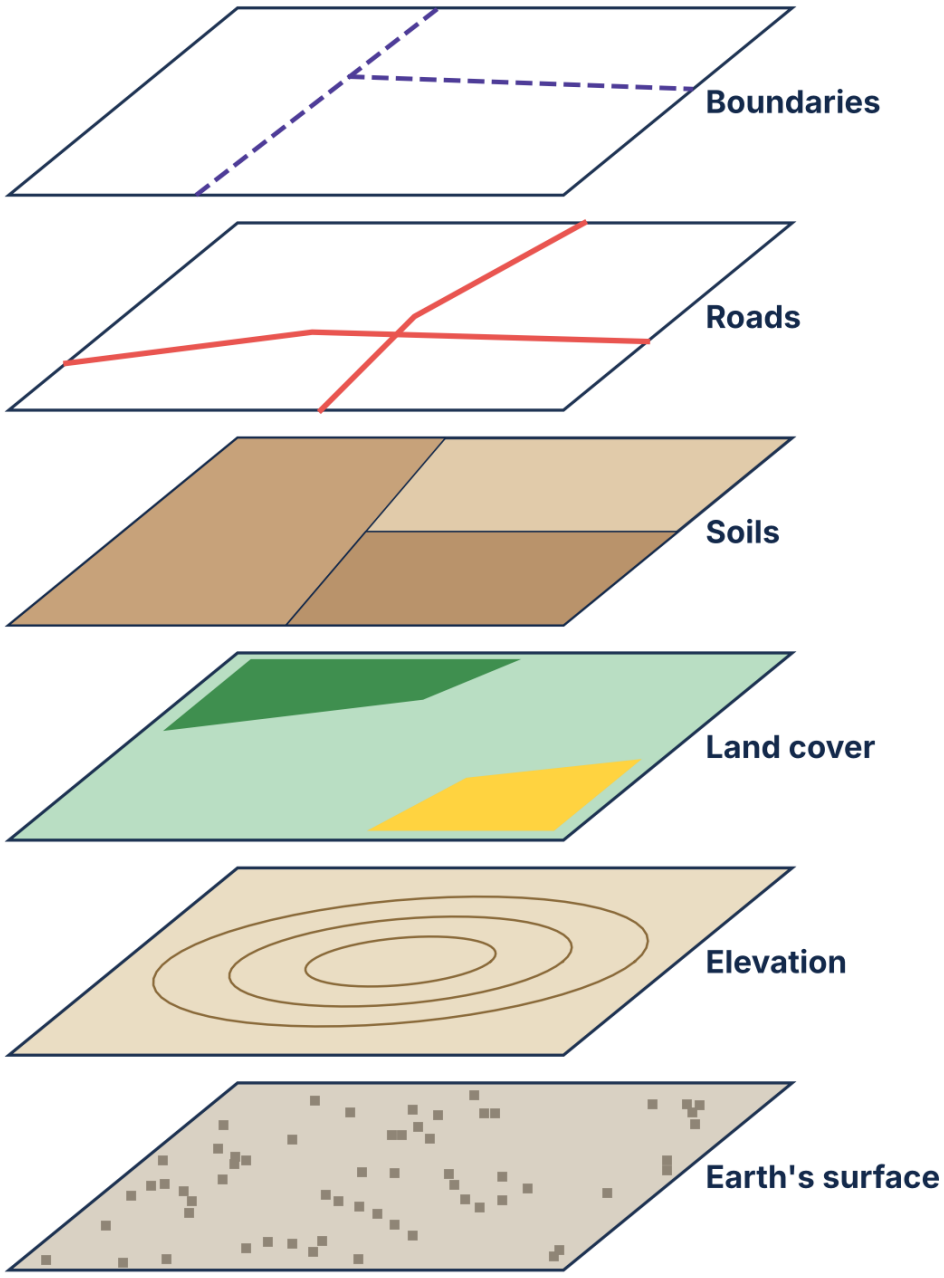


Figure 2.7: Thematic layers stacked over the same area, from the Earth's surface at the bottom up through elevation, land cover, soils, and roads to political boundaries.

The **vector** model stores geometry as coordinates. A point is a single *x, y* pair; a line is an ordered list of pairs; a polygon is a list that closes back on itself. Vector suits discrete objects, and each feature carries its own attributes.

The **raster** model divides space into a regular grid of **cells**, each holding one value. Raster suits continuous data. How much detail it keeps depends on the size of the cells. (In an image, a cell is called a **pixel**, short for *picture element*.)

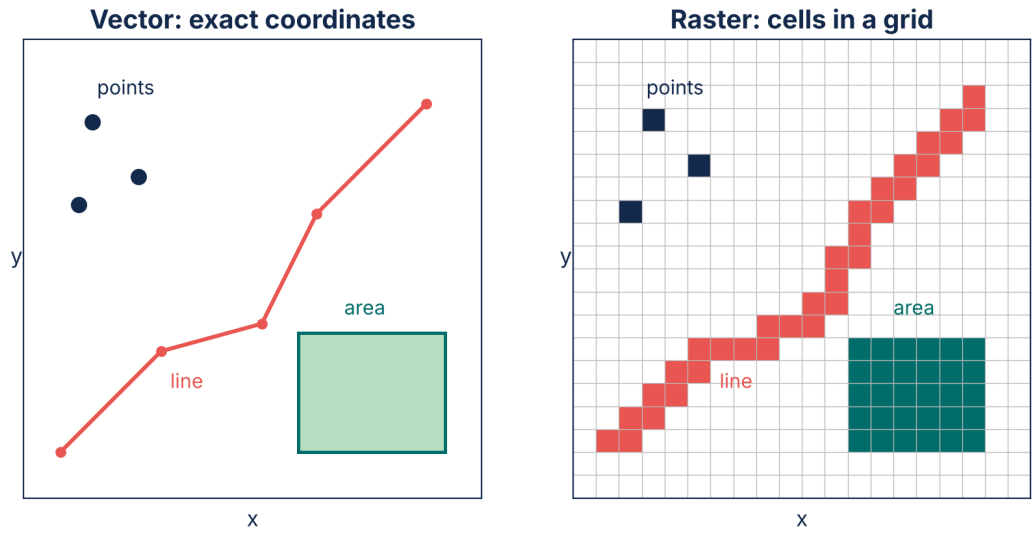


Figure 2.8: The same points, line, and area stored as vector (left) and raster (right). In the raster, each feature becomes a group of cells, and the line becomes a staircase.

Table 2.1: The vector and raster data models compared.

	Vector	Raster
Stores	Points, lines, and polygons	A regular grid of cells
Best for	Discrete objects	Continuous data
Location	*x, y* coordinates	The cell's row and column in the grid
Detail set by	The precision of the coordinates	The cell size
Attributes	Each feature has its own row of attributes	Each cell holds one value
Examples	Roads, parcels, wells, county boundaries	Elevation, satellite imagery, land cover, temperature

Before you read the examples in the table, try to think of your own: what would you store as vector, and what as raster?

2.3.2 The vector model: one entity, several features

The vector model gives you choices. The buildings in a neighborhood can be stored as points, as outlines (lines), or as filled areas (polygons) (Figure 2.9).

The right choice depends on scale and purpose. A citywide map of schools only needs points. A campus map for finding building entrances needs footprints. What is the difference between a line and an area? *Enclosure*: an area is a closed shape with an inside, so you can measure how big it is and ask whether something is in it. An outline drawn as a line has no inside, only a length.

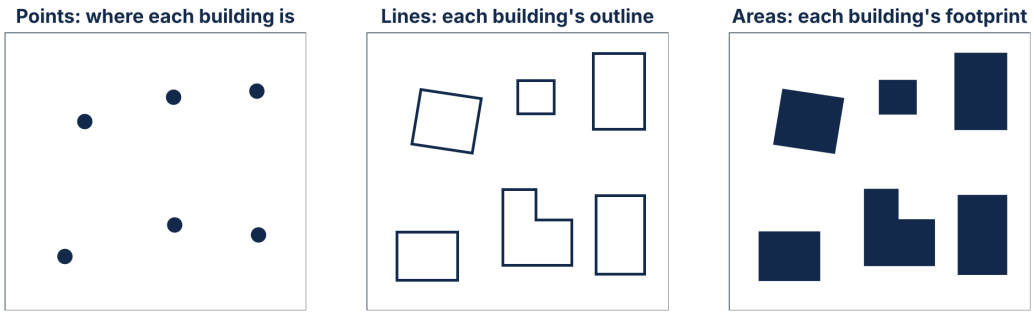


Figure 2.9: The same buildings stored three ways in the vector model: as points, as outlines, and as filled areas.

2.3.3 Vector terms

A few terms describe the parts of vector features (Figure 2.10):

- A **point** is a standalone location.
- A **node** is the start or end point of a line.
- A **vertex** (plural **vertices**) is an intermediate point where a line changes direction.
- A **line segment** is the straight piece between two vertices or nodes.
- A **polygon** is a closed line that encloses an area. Its corners are vertices.

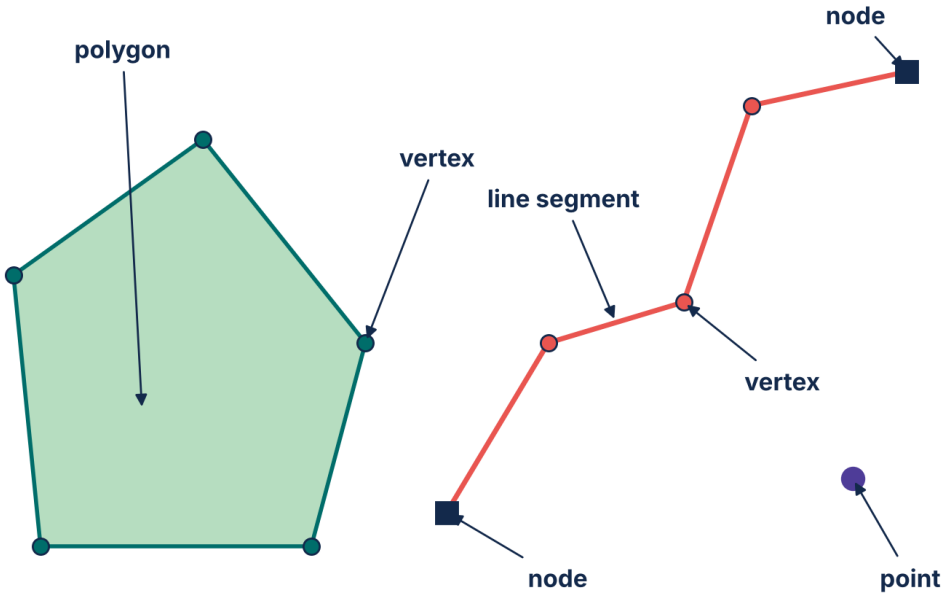


Figure 2.10: The parts of vector features: a polygon and its vertices, a line with a node at each end and vertices between, its line segments, and a point.

Nodes matter for **networks**. Roads meet at nodes, so routing software, like the route optimization in Chapter 1, uses them to find paths from one road to the next.

When you edit vector data in QGIS or ArcGIS Pro, you will move, add, and delete vertices one at a time. The more vertices a line has, the more closely it can follow a curving river, and the larger its file.

2.3.4 The raster model: a grid of cells

A **raster** is rows and columns of equal-sized cells, laid on an *x, y* coordinate system (Figure 2.11). Four things define it:

- **Each cell holds one value.** Most values are numbers, such as an elevation in meters. A value can also stand for a category, such as a land cover class (1 = water, 2 = forest, 3 = farmland), usually stored as a numeric code with a table that says what each code means. The cell marked N in the figure is a reminder that a value need not be a measurement.
- **The cell size** is the real-world length of one side of a cell, such as 30 meters. It sets the raster's **resolution**.
- **The coordinates of the lower-left cell** anchor the grid to a place on Earth.
- **The number of rows and columns** sets how far the grid extends.

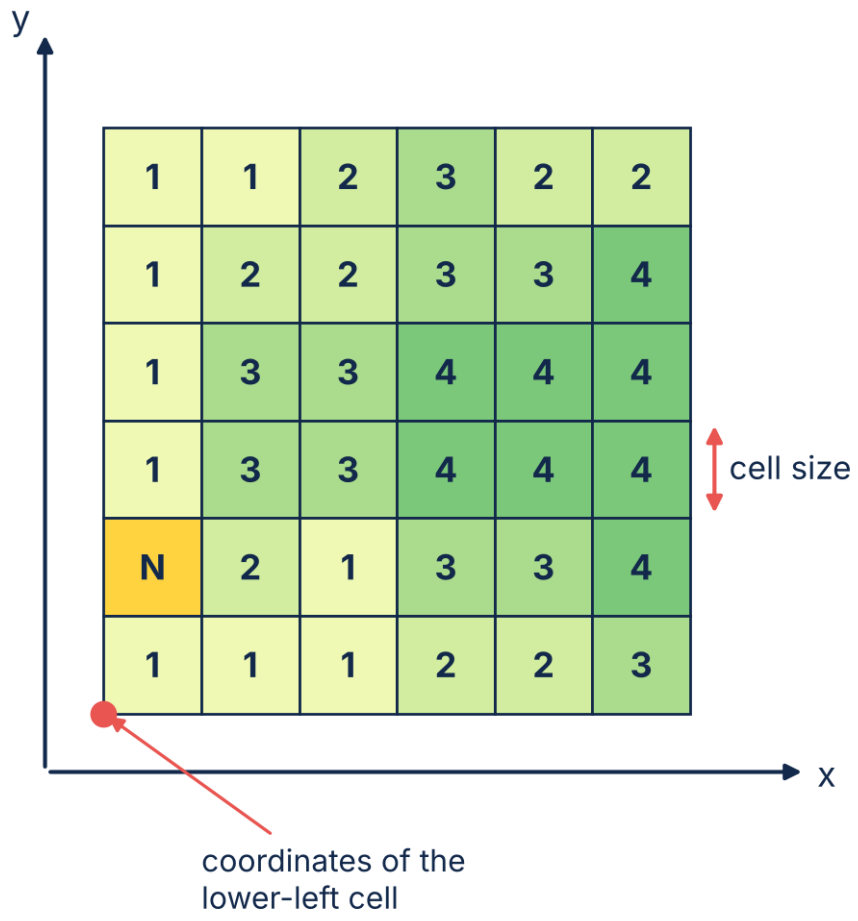


Figure 2.11: A raster grid. Each cell holds one value, and the grid is located by the coordinates of its lower-left cell.

Because every cell is the same size, you don't need to store the coordinates of each cell. You only need the coordinates of one corner and the cell size, and you can work out where every other cell is. That is a big part of why rasters are simple and fast to process.

2.3.5 Cell size and cell coordinates

Cell size is the ground distance covered by one side of a cell. Smaller cells mean more cells over the same area, which means finer resolution and larger files. Figure 14.5 shows the same 200-meter square at three cell sizes. Halving the cell size multiplies the number of cells by four: 4, then 16, then 64.

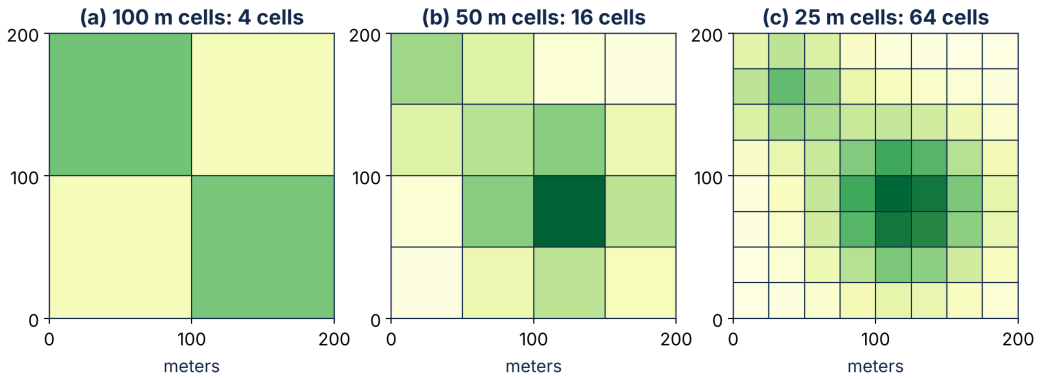


Figure 2.12: The same 200 m square at three cell sizes: (a) 100 m cells, 4 cells; (b) 50 m cells, 16 cells; (c) 25 m cells, 64 cells.

To find where a cell is, start from the lower-left cell and count. Number rows and columns from 0, starting at the lower-left cell. Then:

$$N_{cell} = N_{lower\text{-}left} + (\text{row} \times \text{cell size})$$

$$E_{cell} = E_{lower\text{-}left} + (\text{column} \times \text{cell size})$$

Here *N* is the **northing**, a distance north (the *y* coordinate), and *E* is the **easting**, a distance east (the *x* coordinate). You will meet eastings and northings again when we study coordinate systems.

For example, suppose the lower-left cell of a raster is at E 500,000 m, N 4,400,000 m, and the cells are 25 meters on a side. The cell in row 2, column 3 is at:

- N = 4,400,000 + (2 × 25) = **4,400,050 m**
- E = 500,000 + (3 × 25) = **500,075 m**

Trivia time: which cell size gives the best resolution?

Look at Figure 14.5 again. If you were buying a new TV, which of these would give you the best picture: (a) 100 m cells, (b) 50 m cells, or (c) 25 m cells?
Show the answer
(c), the 25-meter cells. Smaller cells mean more cells over the same area, so finer detail. A TV works the same way: more, smaller pixels on the same size screen give a sharper picture. The cost? Grid (c) has sixteen times as many cells as grid (a), so its file is larger and slower to process.

Did you know..?

Each cell in an image from the Landsat satellites, which have photographed Earth continuously since 1972, is 30 meters on a side. That is about the size of a baseball infield, whose bases are 90 feet (27.4 meters) apart. A single cell can't show you a

house, but a grid of them covering every place on Earth, every 16 days for over 50 years, can show you forests being cut, cities growing, and lakes drying up.

2.4 How is geographic data stored?

2.4.1 Data structures

The data model is the concept; the **data structure** is how a computer actually stores it. Figure 2.13 shows one way a vector polygon can be stored. The polygon is built from three **arcs** (A1, A2, and A3), each a short line. Each arc is stored as an ordered list of coordinate pairs at particular addresses in the computer's memory, and a **pointer** tells the computer which arc comes next.

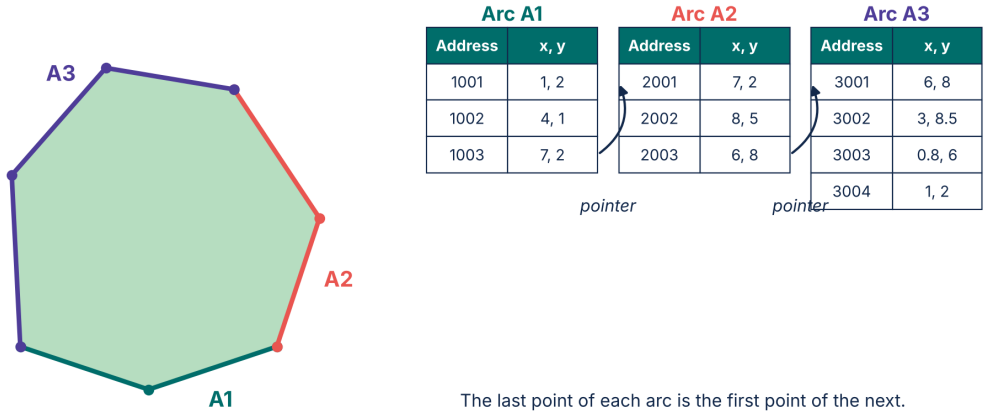


Figure 2.13: A polygon stored as three arcs. Each arc is a list of coordinates in memory, and pointers link one arc to the next.

Notice that the last coordinate of each arc is the same as the first coordinate of the next. That shared point is what joins the arcs into a closed polygon. Storing shared edges once, and recording how they connect, is called **topology**. It saves space, and it guarantees that two neighboring polygons, like two counties, share exactly the same border, with no gaps or overlaps between them. You will meet topology again when you edit data.

You rarely need to think about data structures. But they explain things you will notice: why some file formats are larger than others, why a file can be corrupted if one part is missing, and why some tools need data in a particular format.

2.4.2 The Esri shapefile

The data structure you will use most often this semester is the **shapefile**, a vector format that Esri created in the early 1990s and published as an open specification in 1998 (Environmental Systems Research Institute 1998). Despite its name, a shapefile is not one file. It is a group of files that share a name and together store one vector layer (Table 2.2).

Table 2.2: The files that make up a shapefile named roads.

File	What it holds	Required?
`roads.shp`	The feature geometry: the coordinates of each point, line, or polygon	Yes
`roads.shx`	A positional index that links each shape to its record	Yes
`roads.dbf`	The attribute table, in dBASE format	Yes
`roads.prj`	The coordinate system and projection	No, but you almost always need it
`roads.sbn`, `roads.sbx`	A spatial index for faster searches	No
`roads.cpg`	The character set used for text in the `.dbf`	No
`roads.shp.xml`	Metadata for the layer	No

Three files are required: `.shp`, `.shx`, and `.dbf`. Without the `.prj` file, software can't tell where on Earth the coordinates are, so the layer may show up in the wrong place, or not at all.

The most common student mistake is to email, upload, or move only the `.shp` file. The other files are left behind, and the shapefile won't open. **Always copy, move, and rename all of a shapefile's files together**, or zip the folder that holds them. Better yet, use the tools in QGIS (the Browser panel) or ArcGIS Pro (the Catalog pane), which treat the group of files as one layer.

Did you know..?

The shapefile is over thirty years old, and it shows its age. Its attribute table is a `.dbf` file from dBASE, a database program from the early 1980s, so field names can be no more than **10 characters** long (`POPULATION_2020` becomes `POPULATION`), a table can have at most 255 fields, and the `.shp` and `.dbf` files can each be no larger than **2 GB**. Each shapefile can also hold only one kind of geometry. Some GIS users run a website, *Switch from Shapefile*, that lists these limits and urges everyone to move to newer formats. The most popular replacement is the **GeoPackage** (`.gpkg`), an open standard from the Open Geospatial Consortium that stores many layers, rasters as well as vectors, in a single file. It has been QGIS's default format since 2018. Yet the shapefile is still everywhere, because nearly every GIS program ever written can read it.

2.4.3 Metadata

Before you trust a dataset, read its **metadata**: data about the data. Good metadata answers questions like these:

- **Who** created the data?
- **When** were they created, and for what time period?
- **How** were they created, and how accurate are they?
- **What** coordinate system do they use?
- **Where** can the data be found, and **can I use them**? Are there limits on how they may be shared or used?

Metadata is how you judge whether a dataset fits your purpose. A road layer made from 1990s paper maps may be fine for a state overview and useless for routing an ambulance. A land cover raster made with 30-meter cells can't tell you how many trees are in your yard.

In the United States, many public datasets follow the metadata standard of the Federal Geographic Data Committee (FGDC); internationally, the ISO 19115 standard is common. In QGIS you can read a layer's metadata in its **Layer Properties**, and in ArcGIS Pro in the **Metadata** tab of the Catalog pane or a layer's properties.

Think about it

Your consulting firm has finished its campus map. What metadata would you write, so that a student three years from now could decide whether to reuse your data?

2.5 Vector or raster?

Neither model is better in general. Each has strengths and weaknesses (Table 2.3), and you choose the one that fits the data and the question.

Table 2.3: Advantages and disadvantages of the vector and raster models.

Model	Advantages	Disadvantages
Vector	Precise locations and boundaries	A more complex data structure
	Compact storage for discrete features	Slower for overlay and surface analysis
	Each feature carries its own attributes	A poor fit for continuous data
	Looks good at any zoom level	
Raster	A simple grid structure	Precision limited by cell size
	Fast overlay and **map algebra** (math on grids, cell by cell)	Large files at fine resolution
	A natural fit for continuous data and imagery	Lines and edges look blocky
	Satellite and drone imagery is already raster	Each cell holds only one value

A memory aid that generations of GIS students have used for exams:

> **Vector is corrector, but raster is faster.**

Vector keeps exact coordinates, so shapes and boundaries are more correct. Raster's simple grid makes overlay and analysis faster. Here are a few choices to try:

- **Property parcel boundaries**: vector. A parcel's boundary is a legal line, and it needs to be exact.
- **A snow-depth map**: raster. Snow depth is continuous; there is a value everywhere.
- **A city's bus routes and stops**: vector. Lines and points, with schedules and route numbers as attributes.
- **A satellite image of a wildfire**: raster. Imagery is collected as a grid of pixels.
- **Land cover for a whole state**: usually raster, because it covers everything, with no gaps.



Software can convert between the two. Turning vector data into a raster is called **rasterizing**, and turning a raster into vector features is called **vectorizing**. Each conversion loses some information: rasterizing loses precision, and vectorizing produces blocky, staircase-shaped lines that follow the cell edges.

Did you know..?

The word *raster* comes, through German, from the Latin *rastrum*, a rake. Early television and computer screens drew their pictures as a set of parallel lines swept across the screen, like the lines a rake leaves in soil, and that pattern became known as a raster scan. *Vector* comes from the Latin *vehere*, to carry: in mathematics, a vector carries a point from one place to another, which is just what the line segments of a vector feature do.

2.6 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

2.7 Summary

- A map abstracts the world. Real-world **entities** become **features**: points, lines, polygons, and annotation. Every map is a set of choices about what to keep.
- **Data abstraction** runs from the real world to a **data model** (how people think about the data), a **data structure** (how a computer lays it out), and machine code.
- **Discrete objects** are present or absent, with clear boundaries; **continuous data** have a value everywhere.
- Every entity has a location, an extent, attributes, a time, and a geometry. **Attributes** are non-spatial facts that describe a spatial feature, stored in a table with one row per feature.
- **Layers** group similar features, and stacked layers make a map.
- The four main data models are **vector**, **raster**, **TIN**, and **tabular**. Vector stores points, lines, and polygons as coordinates, built from **nodes**, **vertices**, and **line segments**. Raster stores a grid of **cells**; the **cell size** sets its resolution, and every cell's location can be worked out from the lower-left cell.
- **Shapefiles** store a vector layer in several files (`.shp`, `.shx`, `.dbf`, and usually `.prj`) that must stay together. **Metadata** describes who made the data, when, how, and how accurate it is.
- *Vector is corrector, but raster is faster.* Choose the model that fits the data and the question.

Now return to your campus map list from the start of the chapter. Which entities would you store as vector features, and which as rasters? What attributes and metadata would you record?

2.8 Exercises

1. **Your campus map.** For each item on the list you made at the start of the chapter, say whether you would store it as a point, a line, a polygon, or a raster, and why. List two attributes for each.

2. **Five kinds of information.** Pick an entity near you, such as your dorm, a bus stop, or a creek. Describe its location, extent, attributes, time, and geometry.
3. **Discrete or continuous?** Give three examples of discrete objects and three of continuous data from your own major or interests. Then find one thing that could be either, and explain when you would treat it each way.
4. **Scale changes everything.** Describe how a river, an airport, and a university could each be represented as a point, a line, or a polygon, depending on the scale of the map.
5. **Cell arithmetic.** A raster's lower-left cell is at E 650,000 m, N 3,950,000 m, and its cells are 30 m. Find the easting and northing of the cell in row 4, column 7. How many cells would it take to cover a 3 km by 3 km square? How many with 10 m cells?
6. **Read the metadata.** Download a layer from NC OneMap (https://www.nconemap.gov) and open its metadata. Who made it, when, and how? What coordinate system does it use? Would you trust it for an ambulance routing system? For a state overview map?
7. **Shapefile detective.** Download a shapefile from the U.S. Census Bureau's TIGER/Line page, unzip it, and list every file in the folder. What does each one hold? Then open the layer in QGIS or ArcGIS Pro and look at its attribute table. Which field names look cut short to 10 characters?
8. **Vector or raster?** For each of these, choose vector or raster and explain why: hiking trails, wildfire burn severity, voting precincts, nighttime lights seen from space, the locations of cell towers, and annual rainfall.

2.9 Further reading and online resources

2.9.1 Books and open textbooks

- *GIS Fundamentals: A First Text on Geographic Information Systems*, by Paul Bolstad (Bolstad 2022). Chapter 2, "Data Models," covers the vector, raster, and TIN models, topology, and data structures in depth.
- *Understanding GIS through Sustainable Development Goals*, by Paul Holloway (Holloway 2023). Its chapters on spatial data introduce vector and raster data with worked examples.
- *Essentials of Geographic Information Systems*, by Jonathan Campbell and Michael Shin. Chapter 4, "Data Models for GIS," and Chapter 5, "Geospatial Data Management," are free online: https://open.umn.edu/opentextbooks/textbooks/67, also on LibreTexts at https://geo.libretexts.org/Bookshelves/Geography_(Physical)/Essentials_of_Geographic_Information_Systems_(Campbell_and_Shin).
- *Intro to GIS and Spatial Analysis*, by Manuel Gimond. See the chapters "Feature Representation" and "GIS Data Formats": https://mgimond.github.io/Spatial/.
- *The Nature of Geographic Information*, by David DiBiase and colleagues at Penn State. Chapter 1, "Data and Information," and its sections on vector and raster representations: https://www.e-education.psu.edu/natureofgeoinfo/.
- *GIS Commons: An Introductory Textbook on Geographic Information Systems*, by Michael Schmandt. A free online text with a chapter on data models: https://giscommons.org.
- *Geographic Information Systems and Science*, by Paul Longley, Michael Goodchild, David Maguire, and David Rhind. The chapters on representing geography and on data models are classics of the field.

2.9.2 Data models: vector, raster, and TIN

- The GIS&T Body of Knowledge, from UCGIS, has short articles on this chapter's ideas. Search it for "Vector Data Models," "Raster Data Models," "The Triangulated Irregular Network (TIN)," "Conceptual Models of Space," and "Topology": https://www.ucgis.org/site/gis-t-body-of-knowledge.
- QGIS, *A Gentle Introduction to GIS*, the chapters "Vector Data," "Vector Attribute Data," and "Raster Data": https://docs.qgis.org/latest/en/docs/gentle_gis_introduction/.
- Esri, "Introduction to image and raster data," in the ArcGIS Pro help: https://pro.arcgis.com/en/pro-app/latest/help/data/imagery/introduction-to-raster-data.htm.
- Esri, "TIN in ArcGIS Pro": https://pro.arcgis.com/en/pro-app/latest/help/data/tin/tin-in-arcgis-pro.htm.
- National Geographic Education, "Vector and raster data" and related resources on GIS: https://education.nationalgeographic.org/resource/geographic-information-system-gis/.
- GIS Geography, "Vector vs Raster in GIS: What's the Difference?": https://gisgeography.com/spatial-data-types-vector-raster/.
- Earth Lab, University of Colorado Boulder, "Introduction to spatial data: vector and raster" lessons: https://www.earthdatascience.org.
- Data Carpentry, "Introduction to Geospatial Concepts," a short lesson on vector and raster data, metadata, and file formats (no longer updated, but still useful): https://datacarpentry.github.io/organization-geospatial/.

2.9.3 File formats

- Esri, *ESRI Shapefile Technical Description* (Environmental Systems Research Institute 1998), the original specification: https://www.esri.com/content/dam/esrisites/sitecore-archive/Files/Pdfs/library/whitepapers/pdfs/shapefile.pdf.
- Esri, "Shapefiles" in the ArcGIS Pro help, including which files make up a shapefile: https://pro.arcgis.com/en/pro-app/latest/help/data/shapefiles/working-with-shapefiles-in-arcgis-pro.htm, and "Geoprocessing considerations for shapefile output," on its limits: https://pro.arcgis.com/en/pro-app/latest/tool-reference/appendices/geoprocessing-considerations-for-shapefile-output.htm.
- Switch from Shapefile, a community site that lists the shapefile's limits and the alternatives: http://switchfromshapefile.org.
- GeoPackage, the open standard from the Open Geospatial Consortium (OGC): https://www.geopackage.org.
- GDAL, the open-source library behind most GIS software's ability to read and write data, with a list of every vector and raster format it supports: https://gdal.org/en/stable/drivers/vector/index.html and https://gdal.org/en/stable/drivers/raster/index.html.
- GeoJSON, a simple text format for vector data on the web: https://geojson.org. Try drawing features and watching the coordinates appear at https://geojson.io.
- Cloud Optimized GeoTIFF, a raster format built for the cloud: https://cogeo.org.

2.9.4 Metadata

- Federal Geographic Data Committee (FGDC), geospatial metadata standards: https://www.fgdc.gov/metadata.
- U.S. Geological Survey, "Metadata creation" and data management guidance: https://www.usgs.gov/data-management/metadata-creation.

- QGIS User Guide, “Metadata Properties” for a layer: https://docs.qgis.org/latest/en/docs/user_manual/working_with_vector/vector_properties.html#metadata-properties.
- ArcGIS Pro, “View metadata”: https://pro.arcgis.com/en/pro-app/latest/help/metadata/view-and-edit-metadata.htm.
- Data.gov, the U.S. government’s open data catalog, where every dataset has metadata you can read: https://data.gov.

2.9.5 Raster data and resolution

- USGS, “Landsat Satellite Missions,” the history of every Landsat satellite and its sensors: https://www.usgs.gov/landsat-missions/landsat-satellite-missions.
- NASA Earth Observatory, “How to Interpret a Satellite Image: Five Tips and Strategies”: https://science.nasa.gov/earth/earth-observatory/how-to-interpret-a-satellite-image/.
- USGS 3D Elevation Program (3DEP), elevation rasters for the whole country at several resolutions: https://www.usgs.gov/3d-elevation-program.
- Multi-Resolution Land Characteristics Consortium, the National Land Cover Database (NLCD), a 30-meter land cover raster of the United States: https://www.mrlc.gov.

2.9.6 Software and tutorials

- QGIS Training Manual, “Create a Basic Map” and “Vector Data” modules: https://docs.qgis.org/latest/en/docs/training_manual/.
- QGIS User Guide, “Working with Vector Data” and “Working with Raster Data”: https://docs.qgis.org/latest/en/docs/user_manual/.
- ArcGIS Pro, “Get started with ArcGIS Pro”: https://pro.arcgis.com/en/pro-app/latest/get-started/get-started.htm, and free lessons at Learn ArcGIS: https://learn.arcgis.com.

2.9.7 Where to find geographic data

- NC OneMap, North Carolina’s geospatial data portal: https://www.nconemap.gov.
- U.S. Census Bureau, TIGER/Line shapefiles: https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html.
- USGS EarthExplorer, for Landsat imagery and elevation rasters: https://earthexplorer.usgs.gov.
- USGS National Map Downloader, for elevation, hydrography, and other national layers: https://apps.nationalmap.gov/downloader/.
- Natural Earth, free vector and raster world data: https://www.naturalearthdata.com.
- OpenStreetMap, the free, editable vector map of the world: https://www.openstreetmap.org.

3 Location and Coordinate Systems

Learning objectives

By the end of this chapter you should be able to:

1. Tell relative location from absolute location, and explain why a GIS needs absolute coordinates.
2. Describe how geodesy models the Earth's shape with a sphere, an ellipsoid, and the geoid.
3. Explain the difference between ellipsoid height, orthometric height, and geoid height, and use $h = H + N$.
4. Explain what a geodetic datum is, compare NAD27, NAD83, and WGS84, and describe a datum shift.
5. Convert coordinates between decimal degrees and degrees, minutes, and seconds, and choose a sensible number of decimal places.
6. Describe how latitude and longitude are built from the poles, the equator, and a prime meridian, and define parallels, meridians, great circles, and the graticule.
7. Explain why plotting latitude and longitude as x and y (the Plate Carrée) distorts the map.

Chapter 2 asked how a computer can store a tree, a road, or a mountain. The answer was coordinates: a point is an *x, y* pair, and a line or polygon is a list of them. This chapter asks the next question. *What are those numbers measured from?* A coordinate pair like 35.91, −79.05 means nothing until you know what the Earth's shape is assumed to be, where that shape sits, and how the angles are counted. This is the first central problem of GIS: how do we represent a round, lumpy, spinning world in a computer?

Every GIS layer starts with location, and every location depends on three choices: a **model of the Earth's shape**, a **datum** that pins that model to the ground, and a **coordinate system** for writing positions. This chapter builds those three layers from the ground up. Chapter 4 adds the fourth, the map projection that flattens it all onto a screen.

3.1 Where are we?

People describe where things are all the time, and most of the time they do it relative to something else. "You are here" on a mall map. "Ten paces north of the old oak" on a treasure map. A pile of stones, a cairn, that marks a trail above the treeline. "Third floor, past the elevators" on a building directory. Each is a **relative location**: it describes a place by its relation to a landmark. Relative locations work well, but only if you already know the landmark. Hand the treasure map to someone in another county, cut down the oak, and the directions are useless.

A street address is a step better. 1600 Pennsylvania Avenue NW, Washington, DC, finds the White House for anyone who knows how Washington numbers its streets. But an address

depends on a local naming system, and it doesn't work in the ocean, in a national forest, or in a new subdivision whose streets haven't been named yet.

An **absolute location** names a place in a system that means the same thing to everyone, everywhere. The White House is at 38° 53′ 52″ N, 77° 02′ 11″ W. Anyone with a GPS receiver, in any country, can go to that spot without knowing a single landmark in Washington. That is what a GIS needs: coordinates that resolve to the same spot no matter who reads them, so that layers made by different people at different times line up.

> **Think about it**
>
> Write down three ways to tell a friend where your dorm or apartment is. Which are relative, which are absolute, and which would still work for a visitor from another country?

3.2 Modeling the shape of the Earth

3.2.1 Geodesy

Before we can give anything a coordinate, we need a mathematical model of the Earth to measure against. **Geodesy** is the science of measuring the Earth's size, shape, orientation in space, and gravity field. It is one of the oldest sciences, and it began with a clever bit of geometry.

Around 240 BCE, Eratosthenes, the head of the great library at Alexandria, in Egypt, heard that at noon on the summer solstice the sun shone straight down a well at Syene (today's Aswan), far to the south, so that it cast no shadow. At the same moment in Alexandria, a vertical pole did cast a shadow, at an angle of about 7.2°. Because the sun is so far away, its rays arrive at the Earth parallel to each other. So the shadow angle at Alexandria is the same as the angle between the two cities measured at the Earth's center (Figure 3.1). And 7.2° is one-fiftieth of a full circle (360° ÷ 7.2° = 50). The distance from Alexandria to Syene, about 5,000 *stadia*, must therefore be one-fiftieth of the Earth's circumference, which makes the circumference about 250,000 stadia.

Nobody knows exactly how long Eratosthenes's stadion was, so we can't say exactly how close he came. Depending on which length you use, his answer is within roughly 2 to 15 percent of today's value of about 40,075 km around the equator. Not bad for a stick, a well, and some geometry, more than 2,200 years ago.

Modern geodesy uses satellites, lasers, radio telescopes, and the Global Positioning System (GPS) to measure positions to the centimeter, and to watch the continents drift by a few centimeters a year. That precision is what makes modern datums possible.

3.2.2 Sphere or ellipsoid?

The simplest model of the Earth is a **sphere**, with one radius, about 6,371 km, everywhere. For a world map on a page, a sphere is good enough.

But the Earth spins, and spinning makes it bulge at the equator and flatten at the poles. Its shape is an **oblate spheroid**, or **ellipsoid**: an ellipse rotated about its short, polar axis (Figure 3.2). Two numbers define an ellipsoid:

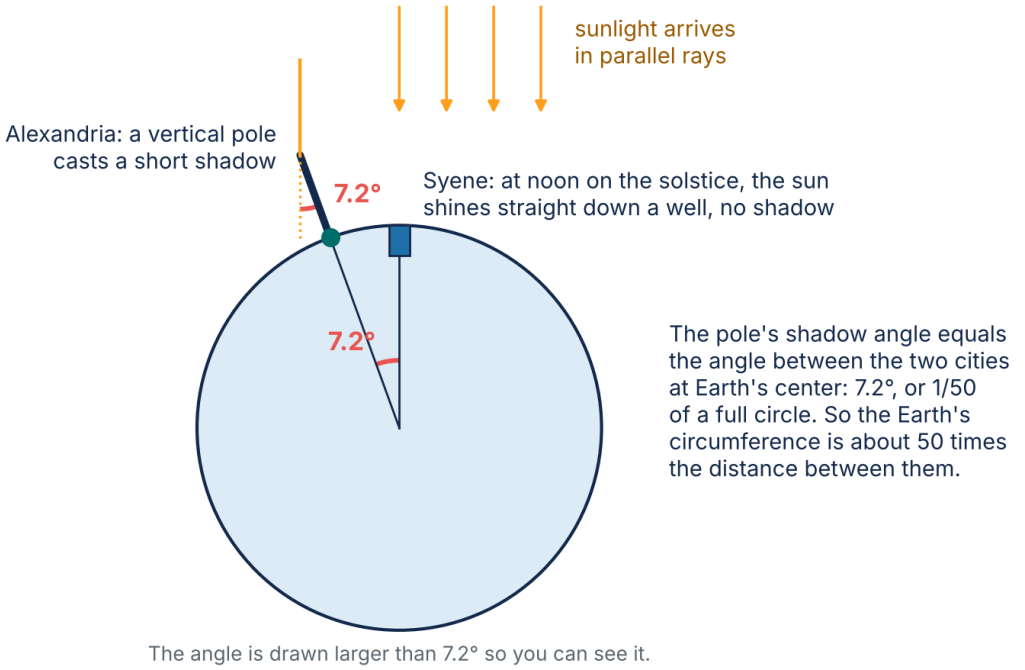


Figure 3.1: How Eratosthenes measured the Earth. The sun's rays are parallel, so the angle of a pole's shadow at Alexandria equals the angle between Alexandria and Syene at the Earth's center.

- The **semi-major axis**, *a*, is the equatorial radius: about 6,378 km.
- The **semi-minor axis**, *b*, is the polar radius: about 6,357 km.

How squashed the ellipsoid is, its **flattening**, is

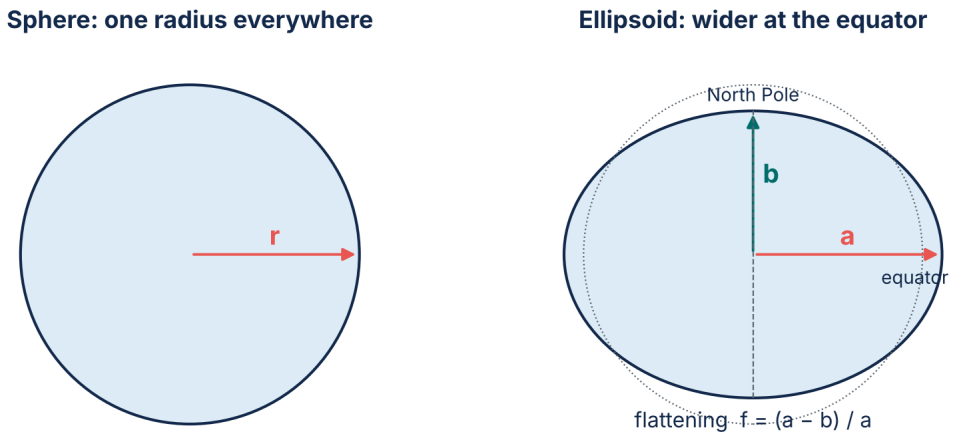
$$f = \frac{a - b}{a}$$

For the Earth, f is about 1/298. The equatorial radius is only about 21 km longer than the polar radius, a third of one percent, which is why the Earth looks perfectly round from space. But 21 km is a big error when you want positions to the meter, so detailed mapping always uses an ellipsoid.

Over the years, surveyors have fitted many ellipsoids to the Earth, each from the best measurements of its day, and some fitted to one region rather than the whole planet. Three you will meet:

Table 3.1: Three common ellipsoids. GRS 80 and WGS 84 differ by only a fraction of a millimeter in *b*.

Ellipsoid	Semi-major axis *a*	Semi-minor axis *b*	Used by
Clarke 1866	6,378,206.4 m	6,356,583.8 m	NAD27, the older North American datum
GRS 80	6,378,137.0 m	6,356,752.3 m	NAD83
WGS 84	6,378,137.0 m	6,356,752.3 m	WGS84, the datum GPS uses



Flattening is exaggerated. On the real Earth, a is only about 21 km longer than b.

Figure 3.2: A sphere (left) has one radius. An ellipsoid (right) has a longer equatorial radius, a, and a shorter polar radius, b. The flattening is greatly exaggerated here.

Did you know..?

Mount Everest is the highest mountain above sea level, at 8,849 m. But because the Earth bulges at the equator, the summit of Chimborazo, a volcano in Ecuador just one degree south of the equator, is the point on Earth's surface farthest from the Earth's center: about 6,384 km, roughly 2 km farther out than Everest's summit, even though Chimborazo is only about 6,263 m above sea level. Stand on top of Chimborazo, and you are the person on land closest to the stars.

3.2.3 The geoid

An ellipsoid is smooth and mathematical. The real Earth isn't. Mountains, ocean trenches, and dense rock deep inside the planet make gravity a little stronger in some places and weaker in others, and water flows "downhill" toward the stronger pull.

Imagine the oceans perfectly still, with no tides, winds, or currents, and extended in canals under the continents. Their surface would settle along a surface of equal gravity. That surface is the **geoid**, and it is the best match to global **mean sea level**. Because mass inside the Earth is unevenly spread, the geoid is lumpy: relative to the WGS 84 ellipsoid, it sits as much as about 106 m below, in the Indian Ocean south of India, and about 85 m above, near New Guinea.

The geoid matters because it is what *elevation* is measured from. When a topographic map says a summit is 2,037 m above sea level, that height is measured from the geoid. Geodesists publish geoid models that give the geoid's height everywhere: **EGM2008** for the whole world, and **GEOID18** for the United States.

3.2.4 Three heights

So every point on the ground has three related heights (Figure 3.3):

- **Ellipsoid height**, *h*: the height above the ellipsoid. This is what a GPS receiver measures.
- **Orthometric height**, *H*: the height above the geoid. This is the elevation printed on maps, the "height above sea level."
- **Geoid height**, *N*: the distance between the two surfaces at that spot, from the ellipsoid up to the geoid.

They are related by a simple equation:

$$h = H + N$$

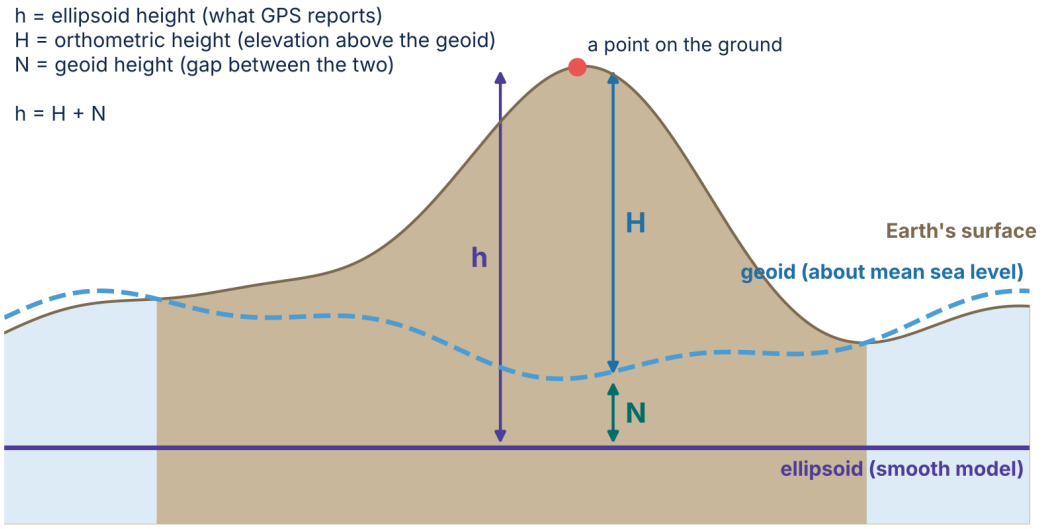


Figure 3.3: Three heights at one point. GPS measures ellipsoid height, h. Maps show orthometric height, H, above the geoid. The geoid height, N, is the gap between the two surfaces.

In the lower 48 United States the geoid lies *below* the ellipsoid, so *N* is negative, roughly −30 m. In North Carolina it is roughly −30 to −35 m. Rearranging the equation, $H = h - N$, so elevation is about 30 m *higher* than the raw height a GPS receiver reports. That is why surveyors and GPS apps always apply a geoid model before they report an elevation.

Think about it

A GPS receiver on a hilltop in Chapel Hill reports an ellipsoid height of 120 m. The geoid height there is about −33 m. What is the hilltop's elevation above sea level?
Show the answer
$H = h - N = 120 - (-33) =$ **153 m**. Subtracting a negative number adds it, so the elevation is higher than the GPS height.

3.3 Geodetic datums

3.3.1 What is a datum?

An ellipsoid on its own is just a shape floating in space. A **geodetic datum** fixes an ellipsoid's size, shape, and position relative to the real Earth: where its center is, and how it is oriented. Only then do latitude and longitude point to real places on the ground. A datum combines:

- an **ellipsoid**,
- an **origin and orientation**, which say where the ellipsoid sits, and
- a **network of control points** on the ground whose coordinates have been measured precisely.

Those **control points** are how a datum is *realized*, made real on the ground. For two centuries surveyors measured them with instruments like the theodolite, a telescope that measures horizontal and vertical angles very precisely, and by **triangulation**: measuring the angles of a chain of triangles between hilltop stations, and from a few carefully measured baselines working out the distance to every other station, across a whole continent. At each station they set a bronze disk, a **benchmark** or survey marker, in rock or concrete, often with a witness post nearby warning people not to disturb it. You can find these disks on mountaintops, bridges, and courthouse steps all over the country, and the National Geodetic Survey publishes the coordinates of each one.

Today the control network is mostly satellites and permanent GPS stations, but the idea is the same. Every coordinate you collect is only meaningful relative to a particular datum.

3.3.2 NAD27

The **North American Datum of 1927 (NAD27)** used the Clarke 1866 ellipsoid, which was fitted to North America rather than to the whole Earth. Its origin was a single survey station on Meades Ranch, in Kansas, near the middle of the contiguous United States. Its coordinates came from a huge adjustment of the triangulation networks of the day. Many older maps, including the classic USGS paper topographic quadrangles, use NAD27.

3.3.3 NAD83

The **North American Datum of 1983 (NAD83)** uses the GRS 80 ellipsoid and is *earth-centered*: the center of its ellipsoid is at the Earth's center of mass. That made it compatible with satellite positioning. NAD83 is fixed to the North American tectonic plate, so the coordinates of a benchmark stay the same as the continent drifts.

NAD83 has been readjusted several times as measurements improved. Each version is a *realization*, written with its year in parentheses: NAD83(HARN), NAD83(2007), and **NAD83(2011)**, the current one. A nationwide network of **Continuously Operating Reference Stations (CORS)**, GPS receivers bolted to buildings and concrete piers that run around the clock, keeps it accurate.

NAD83 is the datum of most U.S. federal and state GIS data, including the U.S. Census Bureau's TIGER/Line files and North Carolina's NC OneMap.

3.3.4 WGS84

The **World Geodetic System 1984 (WGS84)** is a global, earth-centered datum maintained by the U.S. Department of Defense, and it is the datum GPS uses. Coordinates from your phone, from Google Maps, and from most websites are almost always WGS84.

In 1984, WGS84 and NAD83 agreed to within about a meter. But NAD83 moves with the North American plate, while WGS84 is tied to the whole Earth, so as the plate has drifted (and been realized again with better measurements) the two have separated. In North America today they differ by about 1 to 2 meters. For a statewide map, that is too small to see. For a survey of a property line, it matters a lot.

3.3.5 Datum shift

Here is the key point: **the same latitude and longitude point to different places in different datums** (Figure 3.4). The difference between where a coordinate lands in one datum and where it lands in another is a **datum shift**.

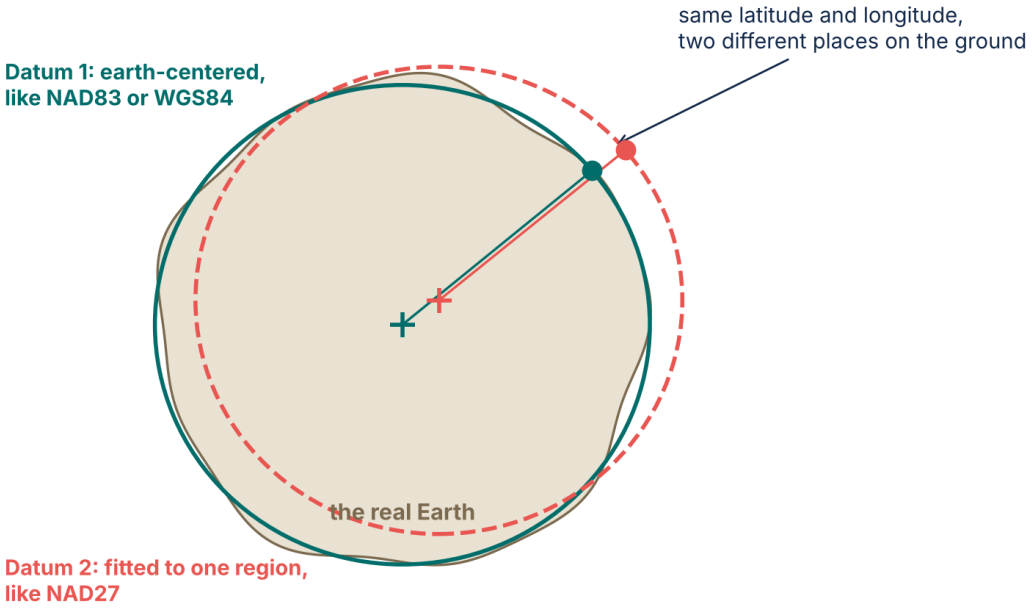


Figure 3.4: A datum shift. Two datums use different ellipsoids in different positions, so the same latitude and longitude land at two different places on the real Earth.

Between NAD27 and NAD83, the shift is anywhere from a few meters to over 100 meters across the United States. In one well-known USGS example from the Hayward, California, quadrangle, a single coordinate pair plotted in NAD27 and in NAD83 lands nearly 100 meters apart. That is a city block, plenty to put a well on the wrong side of a property line or a fire hydrant in the middle of a street.

This is one of the most common errors in GIS. Two layers that should line up, such as roads and parcels, are offset by a constant amount, because one is in NAD27 and the other in NAD83. The fix is a **datum transformation**: a set of equations, or a grid of measured shifts, that converts coordinates from one datum to another. QGIS and ArcGIS Pro apply datum transformations when you project or import data, and they will sometimes ask you to choose one. Before you combine layers, **check each layer's datum**.

Did you know..?

NAD83 is about to retire. The National Geodetic Survey is replacing it with a new set of datums, the modernized National Spatial Reference System. **NATRF2022** (the North American Terrestrial Reference Frame of 2022) will replace NAD83 for latitude and longitude, and **NAPGD2022** will replace NAVD 88 for elevations. Because NAD83's center turned out to be about 2 meters off the Earth's true center of mass, coordinates in the new datum will move by about 1 to 2 meters across the country, and elevations will change by up to a meter or more. Beta tools were released in 2026 and adoption is expected around 2027, so this is a datum change you will live through in your own career.

3.4 Geographic coordinates

3.4.1 Building a coordinate system

With a datum in place, we can write positions as latitude and longitude. Let's build that coordinate system from scratch.

1. **The poles.** The Earth spins around an axis. Where the axis meets the surface are the North and South Poles: two natural reference points.
2. **The equator.** Halfway between the poles is the equator, the natural zero line for north and south.
3. **A prime meridian.** For east and west there is no natural zero line, so we have to choose one. Many countries once used their own: Paris, Washington, Philadelphia, Cádiz. In 1884, the International Meridian Conference in Washington, DC, chose the meridian through the Royal Observatory at Greenwich, England, as the **prime meridian** for the world.

With those reference lines in place (Figure 3.5):

- **Latitude** is the angle north or south of the equator, measured at the Earth's center. It runs from 0° at the equator to 90° N at the North Pole and 90° S at the South Pole.
- **Longitude** is the angle east or west of the prime meridian. It runs from 0° at Greenwich to 180° east or west. 180° E and 180° W are the same line, which the International Date Line roughly follows.

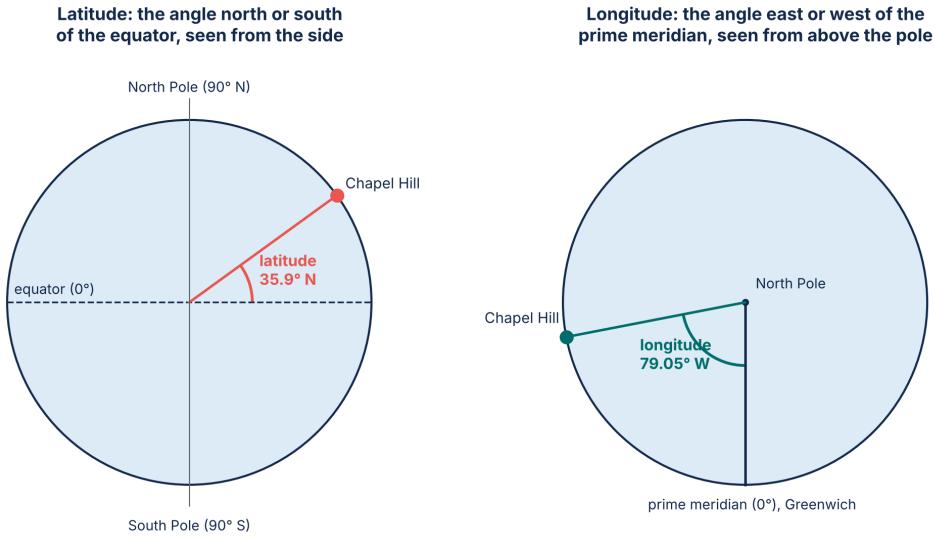


Figure 3.5: Latitude (left) is an angle north or south of the equator. Longitude (right) is an angle east or west of the prime meridian, here seen from above the North Pole. Chapel Hill is at about 35.9° N, 79.05° W.

Strictly, on an ellipsoid the latitude angle is measured between the equator and a line *perpendicular to the ellipsoid's surface*, which doesn't pass quite through the center. This **geodetic latitude** is what GIS software and GPS use. The difference from the simple picture is at most about a fifth of a degree, at 45°.

3.4.2 Parallels and meridians

Now draw the grid (Figure 3.6).

- **Parallels** are lines of equal latitude. They run east and west, they are parallel to the equator and to each other, and they get shorter toward the poles. They measure latitude.
- **Meridians** are lines of equal longitude. They run north and south, from pole to pole, and all of them meet at the poles. They measure longitude.

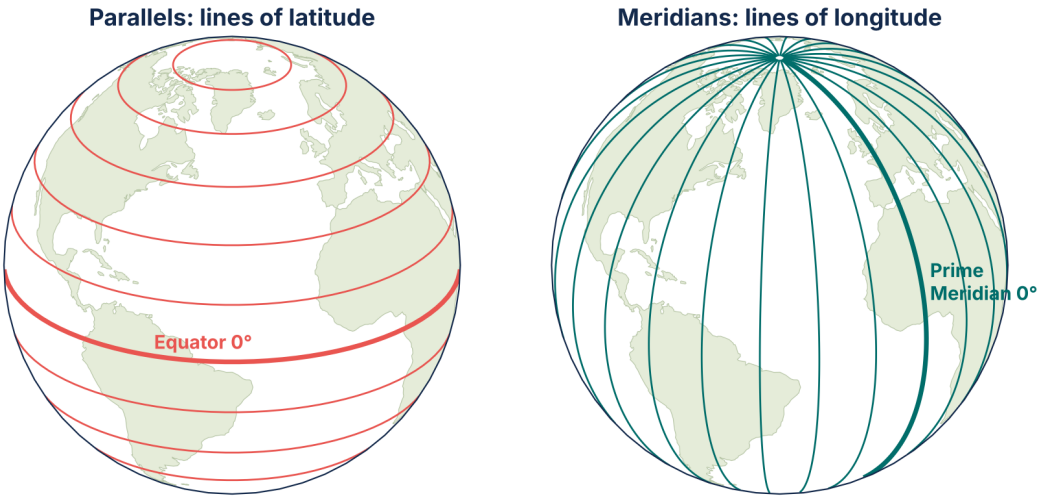


Figure 3.6: Parallels (left) are lines of equal latitude; the equator is the parallel at 0°. Meridians (right) are lines of equal longitude; the prime meridian is the meridian at 0°.

The equator and the prime meridian are both zero lines, so a place on the equator is at 0° N, which is the same as 0° S, and a place on the prime meridian is at 0° E, which is the same as 0° W. Both answers are right.

3.4.3 How far is a degree?

Because the parallels are evenly spaced, one degree of latitude is always about the same distance: **about 111 km**, or 69 miles. (It varies a little, from 110.6 km at the equator to 111.7 km at the poles, because the ellipsoid is flatter there.)

A degree of longitude is different. The meridians come together at the poles, so the distance between them shrinks as you move away from the equator. One degree of longitude is about 111 km at the equator, about 90 km at Chapel Hill's latitude, 56 km at 60° N, and zero at the poles. (The rule is 111 km × the cosine of the latitude.) Remember this. It is why latitude and longitude are angles, not distances, and why you can't measure distance or area in degrees.

3.4.4 Writing degrees

Latitude and longitude are angles, and there are two common ways to write them (Table 3.2).

Decimal degrees (DD) write the angle as one signed number. Positive numbers are north latitudes and east longitudes; **negative numbers are south latitudes and west longitudes**. Chapel Hill is at 35.9132, −79.0558. This is the format most GIS software and GPS units use, and the one you should type into a GIS.

Degrees, minutes, and seconds (DMS) split the angle the way we split an hour: there are 60 minutes (′) in a degree and 60 seconds (″) in a minute, so a second is 1/3,600 of a degree.

Instead of a sign, DMS uses a letter for the hemisphere: N, S, E, or W. Chapel Hill is at 35° 54′ 48″ N, 79° 03′ 21″ W.

Table 3.2: Two ways to write the same longitude, a meridian that runs just east of Denver, Colorado.

	Decimal degrees	Degrees, minutes, seconds
Example	−104.54°	104° 32′ 24″ W
Hemisphere	A sign: negative is south or west	A letter: N, S, E, or W
Used by	GIS software, GPS, spreadsheets, the web	Paper maps, nautical charts, surveys, deeds

3.4.5 Converting between them

Decimal degrees to DMS. Take −104.54°:

1. The whole number is the degrees: **104°**.
2. Multiply the decimal part by 60 to get minutes: 0.54 × 60 = 32.4, so **32′**.
3. Multiply the leftover decimal by 60 again to get seconds: 0.4 × 60 = **24″**.
4. The sign becomes the hemisphere letter. A negative longitude is west: **104° 32′ 24″ W**.

DMS to decimal degrees. Take the White House's latitude, 38° 53′ 52″ N:

$$DD = D + \frac{M}{60} + \frac{S}{3600}$$

$$38 + \frac{53}{60} + \frac{52}{3600} = 38 + 0.8833 + 0.0144 = 38.8978°$$

It is north, so the number stays positive: **38.8978°**. For a longitude west of Greenwich or a latitude south of the equator, put a minus sign in front.

Keep at least four to six decimal places when you convert, or the point will move. The next section shows how far.

Think about it

Convert Chapel Hill's longitude, −79.0558°, to degrees, minutes, and seconds. Then convert 35° 54′ 48″ N to decimal degrees.

Show the answers

79°; 0.0558 × 60 = 3.348, so 3′; 0.348 × 60 = 20.88, about 21″. Negative, so west: **79° 03′ 21″ W**.

35 + 54/60 + 48/3600 = 35 + 0.9 + 0.01333 = **35.9133°**.

3.4.6 Precision and location

How many decimal places should a coordinate have? Each decimal place makes a coordinate ten times more precise (Table 3.3). Since a degree of latitude is about 111 km, each place you add divides that distance by ten.

Table 3.3: How precise is a decimal degree? Distances are for latitude, and for longitude at the equator; a degree of longitude covers less ground away from the equator.

Decimal places	Example	Distance at the equator (approx.)	Enough to find
0	36°	111 km	A large region
1	35.9°	11 km	A town
2	35.91°	1.1 km	A neighborhood
3	35.913°	110 m	A large field or a city block
4	35.9132°	11 m	A house lot
5	35.91320°	1.1 m	A tree
6	35.913200°	11 cm	A survey marker, with survey-grade GPS
7	35.9132000°	1.1 cm	Plate tectonics research

A phone's GPS is usually accurate to about 3 to 5 meters in the open, so **five decimal places**, about a meter, is already more than enough. Reporting eight decimal places claims millimeter precision that a phone doesn't have. Extra decimals don't make your data more accurate; they only make it look more accurate than it is. On the other hand, cutting coordinates to two or three decimal places, which spreadsheets sometimes do on their own, can move your points by a kilometer.

3.5 Lines on the globe

3.5.1 Great circles

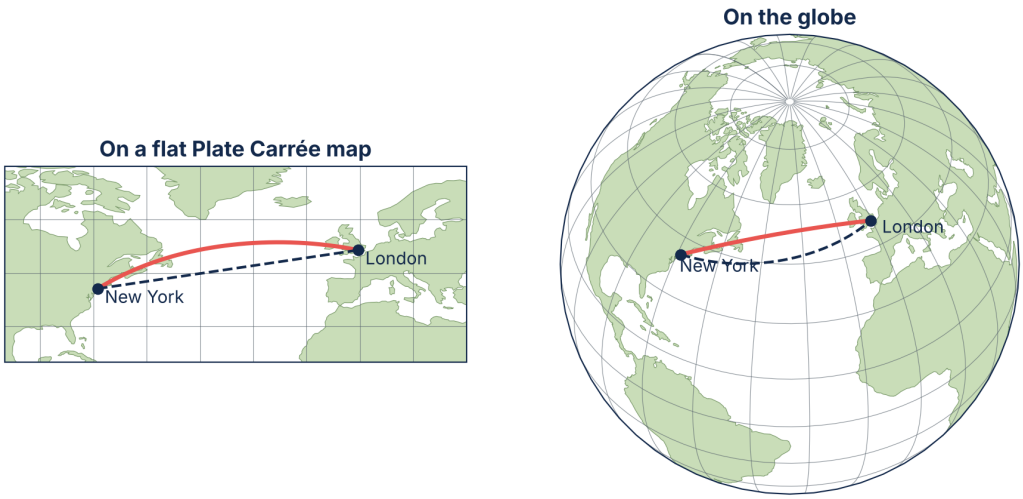
A **great circle** is the largest circle you can draw on a sphere. Its plane passes through the Earth's center, and it cuts the globe into two equal halves. Any circle whose plane doesn't pass through the center is a *small circle*.

Great circles matter for one big reason: **the shortest route between two points on a sphere follows a great circle**. That is why airline routes look curved on flat maps. A flight from New York to London heads northeast, passes near Newfoundland, and approaches Britain from the northwest. It looks like a detour on a flat map, but on the globe it is the straight path (Figure 3.7).

Which parallels and meridians are great circles?

Show the answer

Every meridian, together with the meridian on the opposite side of the globe (180° away), makes a great circle, because its plane passes through the poles and the Earth's center. Of the parallels, **only the equator** is a great circle. Every other parallel is a small circle, and the parallels get smaller toward the poles.



Solid red: the great-circle route, the shortest path (about 5,560 km). Dashed: a straight line on the flat map, about 5,790 km.

Figure 3.7: The great-circle route from New York to London (solid red) and a straight line on a flat Plate Carrée map (dashed). The great circle looks curved on the flat map, but it is the shortest path; the "straight" line is about 230 km longer.

3.5.2 The graticule

The **graticule** is the network of parallels and meridians drawn on a map. It tells readers how the map has bent the globe to flatten it, and it lets them estimate the latitude and longitude of any place.

Choosing which lines to draw is a design decision (Figure 3.8). Too many lines clutter the map and compete with the data; too few, and the grid stops being useful. A common rule of thumb is every 15° or 30° on a world map, every 5° or 10° on a map of a continent, and every degree or less on a map of a state. Always ask what the reader needs the grid for. A world map of climate zones might only need the equator, the tropics, and the Arctic and Antarctic Circles.

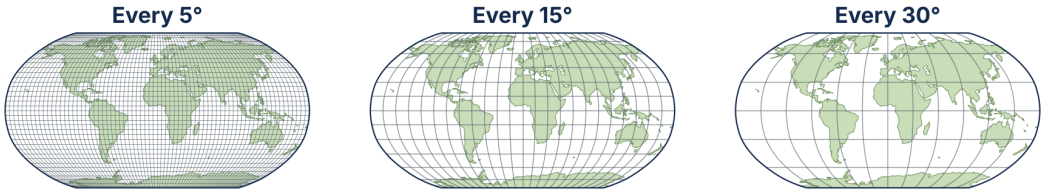


Figure 3.8: The same world map with a graticule every 5°, every 15°, and every 30°. The 5° grid crowds the map; the 30° grid barely helps you read a location.

3.6 From globe to flat map

Latitude and longitude are angles measured on a curved surface. They are not x and y distances on a flat plane. But a computer screen is flat, and so is a sheet of paper. So what happens if we simply pretend they are?

Plot longitude as x and latitude as y on an even grid, and you get the **Plate Carrée**, French for "flat square," also called the *equirectangular projection* (Figure 3.9). Every 1° by 1° cell is

a square on the map. It is what you see when you add a layer in latitude and longitude to a GIS without choosing a projection.

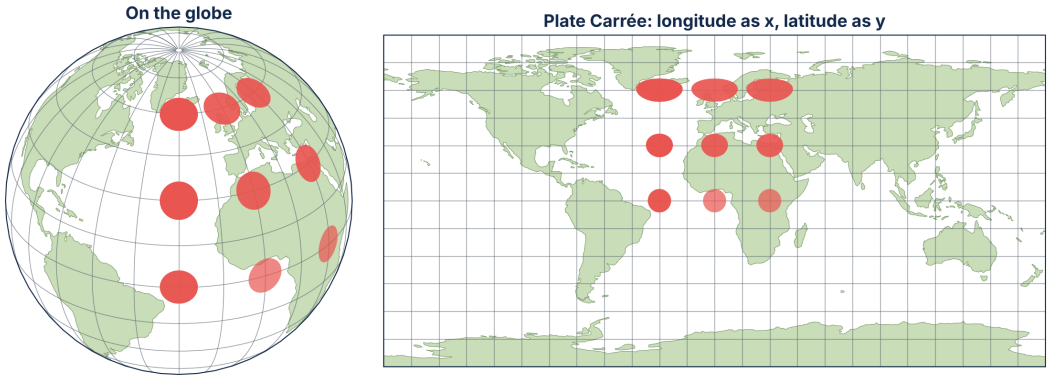


Figure 3.9: On the globe (left), each red circle covers the same area on the ground. On the Plate Carrée (right), where longitude is x and latitude is y, the circles stretch east to west more and more toward the poles.

The grid is simple, but the cells are only square on the map, not on the globe. On the Earth, a 1° cell at the equator is about 111 km on each side; at 60° N it is still 111 km tall but only 56 km wide. The Plate Carrée draws both as squares, so it stretches everything east to west more and more toward the poles. Greenland looks wide and squat, and Antarctica becomes a long band across the bottom of the map.

That raises the question for the next chapter: *how do we represent a round Earth on a flat surface?* Every answer, called a **map projection**, distorts something. Chapter 4 shows what each kind of projection keeps and what it gives up.

Did you know..?

If you visit the Royal Observatory in Greenwich and stand astride the famous brass line of the prime meridian, a GPS app will tell you that you are not at 0°. It reads about 0.0015° W: the zero of longitude that GPS uses is about **102 meters east** of the line. Nothing moved. The brass line marks the meridian astronomers defined in 1851 with a telescope that pointed along the local direction of gravity, which is tilted slightly by the local geoid. The modern earth-centered datums, WGS84 among them, define longitude from the Earth's center of mass instead, and that small difference in the vertical becomes about 100 meters on the ground.

3.7 Putting it together

Think of location in a GIS as a stack, with each layer built on the one below (Figure 4.1):

1. **The Earth**, the real, irregular surface, approximated by the geoid.
2. **An ellipsoid**, a smooth mathematical model of the Earth, such as GRS 80.
3. **A datum**, which pins the ellipsoid to the Earth and defines latitude and longitude, such as NAD83.
4. **A map projection**, which turns latitude and longitude into flat *x, y* coordinates, such as UTM.

A coordinate system that stops at the third layer, giving positions in latitude and longitude on a datum, is a **geographic coordinate system (GCS)**. NAD83 and WGS84, used for latitude

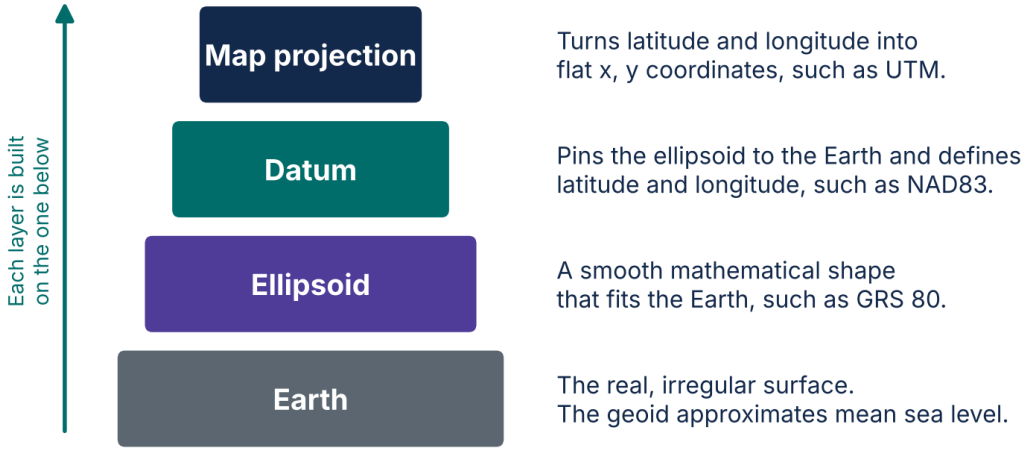


Figure 3.10: From the Earth to a map. Each layer is built on the one below. This chapter covered the bottom three; Chapter 4 covers map projections.

and longitude, are both geographic coordinate systems. A coordinate system that adds the fourth layer is a *projected coordinate system*, the subject of Chapter 4.

The most common point of confusion in this part of the course is the difference between a **datum** and a **projection**. A datum answers *where is the Earth, and what shape is it?* A projection answers *how do I flatten it?* Changing the datum moves your points on the ground. Changing the projection only changes how they are drawn on a flat map.

3.8 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

3.9 Summary

- A GIS needs **absolute locations**, coordinates that mean the same thing to everyone, rather than **relative locations** tied to landmarks.
- **Geodesy** measures the Earth's size and shape. Eratosthenes estimated its circumference around 240 BCE from shadow angles.
- The Earth is modeled as an **ellipsoid**, defined by its **semi-major axis** a and **semi-minor axis** b, with a **flattening** of about 1/298. Common ellipsoids are Clarke 1866, GRS 80, and WGS 84.
- The **geoid** is the surface of equal gravity that best matches mean sea level. Elevations are measured from it. **Ellipsoid height** (from GPS), **orthometric height** (elevation), and **geoid height** are related by $h = H + N$.
- A **geodetic datum** fixes an ellipsoid to the Earth through **control points**. **NAD27** used Clarke 1866 and an origin in Kansas; **NAD83** uses GRS 80 and is earth-centered; **WGS84** is the global datum GPS uses. The same coordinates land in different places in different datums, so check each layer's datum and apply a **datum transformation** when needed.

- **Latitude** is measured north and south from the **equator**, and **longitude** east and west from the **prime meridian** at Greenwich. **Parallels** are lines of latitude and **meridians** are lines of longitude; the network of both on a map is the **graticule**.
- Coordinates can be written in **decimal degrees** (negative for south and west) or **degrees, minutes, and seconds**. Each decimal place makes a coordinate ten times more precise; five places is about a meter.
- The shortest route between two places follows a **great circle**.
- Treating latitude and longitude as x and y gives the **Plate Carrée**, which stretches the map toward the poles. Flattening the Earth well takes a map projection, the subject of Chapter 4.

3.10 Exercises

1. **Describe a place three ways.** Pick a place on campus. Describe it with a relative location, a street address, and an absolute location in both decimal degrees and DMS. (Find its coordinates by right-clicking it in Google Maps or OpenStreetMap.)
2. **Eratosthenes at home.** Suppose that at noon a pole in your town casts a shadow at an angle of 5°, while at the same moment the sun is straight overhead in a town 550 km due south. What is the Earth's circumference by Eratosthenes's method? How close is it to 40,075 km?
3. **Conversions.** Convert to DMS: 35.2271, −80.8431 (Charlotte) and −33.8568, 151.2153 (the Sydney Opera House). Convert to decimal degrees: 40° 41′ 21″ N, 74° 02′ 40″ W (the Statue of Liberty).
4. **Heights.** Look up the elevation and the geoid height for your hometown with the NGS GEOID18 calculator (https://geodesy.noaa.gov/GEOID/GEOID18/computation.html). What ellipsoid height would a GPS receiver report there?
5. **Find a benchmark.** Use the NGS Data Explorer (https://geodesy.noaa.gov/NGSDataExplorer/) to find a survey marker near campus. What datum are its coordinates given in? When was it last recovered?
6. **Datum detective.** In QGIS, open the Layer Properties of any layer and find its coordinate reference system on the Information tab. In ArcGIS Pro, look in the layer's Properties, on the Source tab. Which datum does it use? Find one layer from a state or federal source and one from a website or a phone app. Do they use the same datum?
7. **Precision.** A classmate collects tree locations with a phone and records them to ten decimal places. Explain, using Table 3.3, why that is misleading. How many decimal places would you keep, and why?
8. **Great circles.** Use the Great Circle Mapper (http://www.gcmap.com) to draw the route from Raleigh-Durham (RDU) to Tokyo (NRT). Which states and countries does it pass over? Why doesn't it head due west?

3.11 Further reading and online resources

3.11.1 Books and open textbooks

- *GIS Fundamentals: A First Text on Geographic Information Systems*, by Paul Bolstad (Bolstad 2022). Chapter 3, "Geodesy, Projections, and Coordinate Systems," covers ellipsoids, the geoid, datums, and coordinates in depth.

- *Understanding GIS through Sustainable Development Goals*, by Paul Holloway (Holloway 2023). Its chapter on coordinate systems and projections works through these ideas in QGIS and ArcGIS Pro.
- *The Nature of Geographic Information*, by David DiBiase and colleagues at Penn State. Chapter 2, "Scales and Transformations," explains coordinate systems, the geoid, the ellipsoid, and datums: https://www.e-education.psu.edu/natureofgeoinfo/c2.html.
- *Essentials of Geographic Information Systems*, by Jonathan Campbell and Michael Shin. Chapter 2, "Map Anatomy," covers the Earth's shape, coordinate systems, and datums: https://open.umn.edu/opentextbooks/textbooks/67.
- *Intro to GIS and Spatial Analysis*, by Manuel Gimond. The chapter "Coordinate Systems" is short and clear: https://mgimond.github.io/Spatial/chp09_0.html.
- *GIS Commons*, by Michael Schmandt. The chapter "Coordinate Systems and Map Projections": https://giscommons.org.
- *Map Projections: A Working Manual*, by John P. Snyder (Snyder 1987), USGS Professional Paper 1395. The first chapters explain the Earth's shape, datums, and the geometry behind coordinates: https://pubs.usgs.gov/publication/pp1395.

3.11.2 Geodesy, the geoid, and heights

- NOAA National Geodetic Survey (NGS), "What is geodesy?": https://oceanservice.noaa.gov/facts/geodesy.html.
- NOAA National Ocean Service, "Geodesy" tutorial, a short illustrated course on the Earth's shape, the geoid, and datums: https://oceanservice.noaa.gov/education/tutorial_geodesy/.
- NGS, the GEOID18 model and an online calculator for geoid height anywhere in the United States: https://geodesy.noaa.gov/GEOID/GEOID18/.
- NGS, "What is the geoid?": https://geodesy.noaa.gov/GEOID/geoid_def.html.
- National Geospatial-Intelligence Agency (NGA), Earth Gravitational Model 2008 (EGM2008): https://earth-info.nga.mil/.
- Esri, "Earth's shape and the geoid," in the ArcGIS Pro help on vertical coordinate systems: https://pro.arcgis.com/en/pro-app/latest/help/mapping/properties/vertical-coordinate-systems.htm.
- The GIS&T Body of Knowledge has articles on "Geodesy," "Earth Geometry," "Datums," "Plane Coordinate Systems," and "Georeferencing Systems": https://www.ucgis.org/site/gis-t-body-of-knowledge.

3.11.3 Datums

- NGS, "Datums and reference frames," and the NGS Datums page: https://geodesy.noaa.gov/datums/index.shtml.
- NGS, "New Datums," on NATRF2022 and NAPGD2022, the replacements for NAD83 and NAVD 88: https://geodesy.noaa.gov/datums/newdatums/index.shtml.
- NGS, the Continuously Operating Reference Station (CORS) network: https://geodesy.noaa.gov/CORS/.
- NGS Data Explorer, a map of survey markers and their datasheets: https://geodesy.noaa.gov/NGSDataExplorer/.
- NGS Coordinate Conversion and Transformation Tool (NCAT), for converting coordinates between datums and formats: https://geodesy.noaa.gov/NCAT/.
- QGIS User Guide, "Working with Projections," including datum transformations: https://docs.qgis.org/latest/en/docs/user_manual/working_with_projections/working_with_projections.html.

- U.S. Geological Survey, “What is the difference between NAD27 and NAD83?” and related FAQs: https://www.usgs.gov/faqs.
- Esri ArcGIS Blog, articles on preparing data for the modernized National Spatial Reference System: https://www.esri.com/arcgis-blog/.

3.11.4 Latitude, longitude, and great circles

- National Geographic Education, “Latitude” and “Longitude” encyclopedia entries: https://education.nationalgeographic.org/resource/latitude/ and https://education.nationalgeographic.org/resource/longitude/.
- Malys, Seago, Pavlis, Seidelmann, and Kaplan, “Why the Greenwich meridian moved,” *Journal of Geodesy* (Malys et al. 2015), the explanation of the 102-meter offset at Greenwich: https://doi.org/10.1007/s00190-015-0844-y.
- Great Circle Mapper, for drawing great-circle routes between airports: http://www.gcmap.com.
- *Longitude*, by Dava Sobel (Walker, 1995), the story of how sailors learned to find their longitude at sea.

3.11.5 Software and tutorials

- QGIS Training Manual, “Vector Analysis: Reprojecting and Transforming Data”: https://docs.qgis.org/latest/en/docs/training_manual/vector_analysis/reproject_transform.html.
- QGIS, *A Gentle Introduction to GIS*, the chapter “Coordinate Reference Systems”: https://docs.qgis.org/latest/en/docs/gentle_gis_introduction/coordinate_reference_systems.html.
- Esri, “Coordinate systems, map projections, and transformations” in the ArcGIS Pro help: https://pro.arcgis.com/en/pro-app/latest/help/mapping/properties/coordinate-systems-and-projections.htm.
- Learn ArcGIS, “Choose the right projection” and related lessons: https://learn.arcgis.com.
- epsg.io, a searchable list of coordinate reference systems and their parameters: https://epsg.io.

4 Map Projections

Learning objectives

By the end of this chapter you should be able to:

1. Explain what a map projection is, and where it sits on top of an ellipsoid and a datum.
2. Name the four properties a projection can preserve (shape, area, distance, and direction) and explain why no flat map preserves them all.
3. Describe the three developable surfaces (the plane, the cylinder, and the cone) and tangent and secant projections, and explain how scale error grows away from standard lines.
4. Explain why the Mercator and Web Mercator projections should not be used to compare or measure areas.
5. Describe the Lambert Conformal Conic and Transverse Mercator projections, the UTM system, and the State Plane Coordinate System, and name the zones that cover North Carolina.
6. Read an EPSG code and tell a geographic coordinate system from a projected one.
7. Choose a projection for a study area and a measurement, and tell defining a projection from projecting data.

Chapter 3 built the bottom three layers of location in a GIS: the real Earth, an ellipsoid that models it, and a datum that pins the ellipsoid to the ground and gives us latitude and longitude. This chapter adds the top layer (Figure 4.1). A **map projection** converts latitude and longitude, angles on a curved surface, into flat *x, y* coordinates on a screen or a sheet of paper.

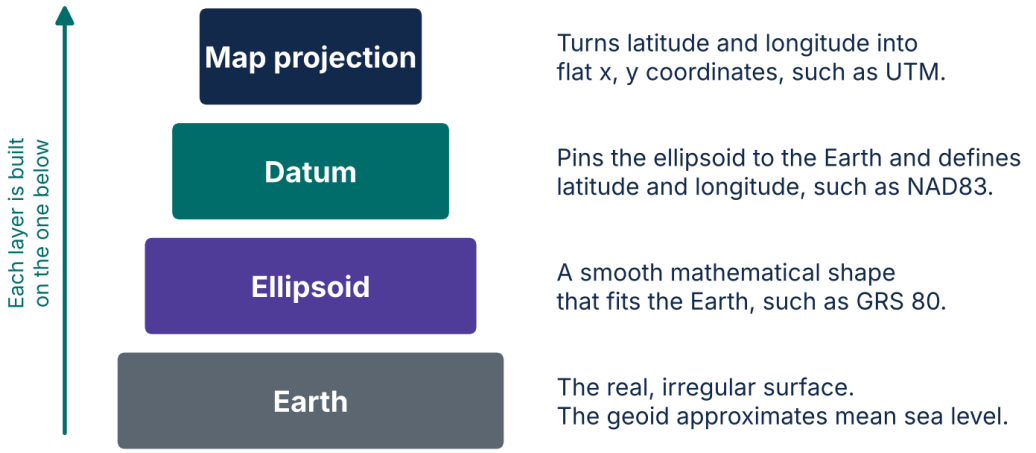


Figure 4.1: From the Earth to a map. Each layer is built on the one below. A map projection, the top layer, flattens the result.

There is a catch, and it is the main idea of this chapter: **every flat map distorts the Earth**. You cannot flatten an orange peel without stretching or tearing it, and you cannot flatten the Earth without distorting shape, area, distance, or direction. Choosing a projection means choosing which errors you can live with. By the end of this chapter you will know how to make that choice for your own GIS work.

4.1 What is a map projection?

A **map projection** is a set of equations that converts latitude and longitude on the ellipsoid into *x, y* coordinates on a flat plane. Every projected map, every web map, and every measurement of distance or area in a GIS depends on one.

The classic way to picture a projection is with light. Put a light bulb inside a transparent globe, hold a screen against it, and the light casts the coastlines and the graticule onto the screen (Figure 4.2). Where the screen touches the globe, the image is exact. Farther away, the shadows stretch, and evenly spaced points on the globe spread farther and farther apart on the screen.

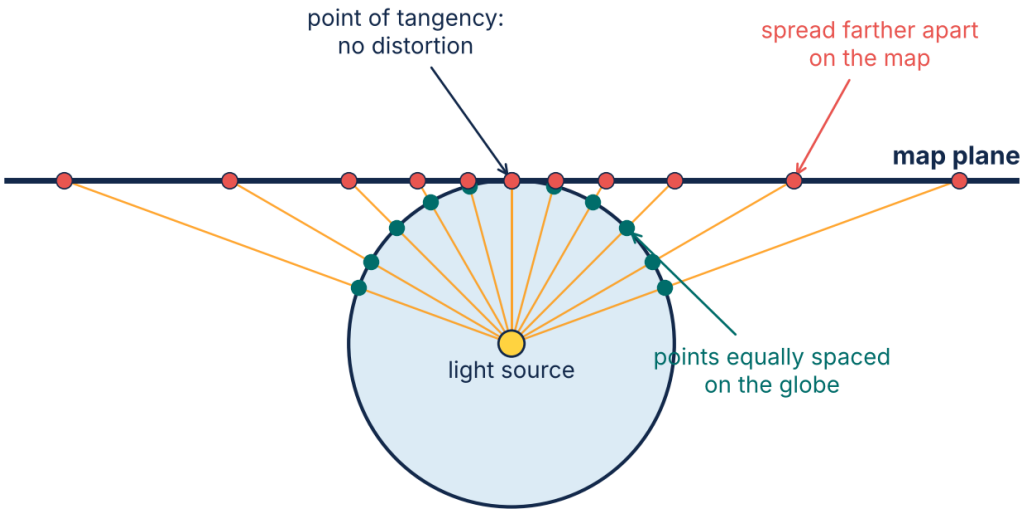


Figure 4.2: The light-bulb picture of a projection. A light at the globe's center casts points onto a flat map plane. Near the point where the plane touches the globe there is no distortion; farther away, points that are evenly spaced on the globe spread apart on the map.

The light bulb is an analogy. Most modern projections are defined by equations, not by real shadows, but the analogy explains why distortion grows with distance from the place where the map surface touches the Earth.

> **Think about it**
>
> Peel a clementine or an orange in as few pieces as you can, and try to press the peel flat on a table. Where does it tear? Where does it stretch? Every map projection makes the same kinds of compromises, just with mathematics instead of fingers.

4.2 Every projection distorts something

4.2.1 Four properties

A map can preserve four properties of the globe, each one named for the kind of projection that keeps it:

- **Shape.** A **conformal** projection keeps local angles true, so small shapes look right. A tiny circle on the globe is still a circle on the map, though it may be larger or smaller.
- **Area.** An **equal-area** (or *equivalent*) projection keeps relative size true. A county twice as large as another on the ground is twice as large on the map.
- **Distance.** An **equidistant** projection keeps true scale along certain lines, usually from one or two points. No projection keeps true distance between *every* pair of places.
- **Direction.** An **azimuthal** projection keeps true directions, or bearings, from a center point.

No flat map keeps all four. In particular, **no projection can be both conformal and equal-area**. Some projections preserve none of the four exactly, but keep every error moderate: these are **compromise** projections.

4.2.2 Seeing distortion: Tissot's indicatrix

How can you *see* what a projection does? In 1859, the French mathematician Nicolas Auguste Tissot came up with an elegant tool, now called **Tissot's indicatrix**: draw identical small circles all over the globe, then project them. On the map, each circle becomes an ellipse whose size and shape show the distortion at that spot. If the circles stay round, shapes are preserved there; if they all cover the same area, area is preserved.

Figure 4.3 shows Tissot's circles, each 500 km across on the ground, on four world maps.

4.2.3 Mercator and reality

The most famous map projection is the one Gerardus Mercator, a Flemish mapmaker, published in 1569 for sailors. On a **Mercator** map, a line of constant compass bearing, called a *rhumb line*, is a straight line. A navigator could draw a straight line from port to port, read its angle, and steer that compass course the whole way. The Mercator is conformal, so small shapes and angles are right, which is what made it so useful at sea.

The price is area. To keep shapes right while the meridians stay parallel, Mercator stretches the map north and south as much as it stretches east and west, and that stretch grows with latitude. On a Mercator map, the scale at any latitude is the *secant* of the latitude (1 ÷ cos), and areas are enlarged by the square of that. At 60° north or south, a place looks **four times** larger than it is. Near the poles, the enlargement goes to infinity, which is why Mercator maps are always cut off before the poles.

The classic example is Greenland and Africa (Figure 4.4). On a Mercator map they look about the same size. In fact, Africa, about 30 million km², is roughly **14 times** larger than Greenland, about 2.2 million km². Countries near the equator, like Brazil, Indonesia, and the Democratic Republic of the Congo, barely change size; countries far from it, like Canada, Russia, and Greenland, swell.

Use Mercator for navigation, not for comparing areas.

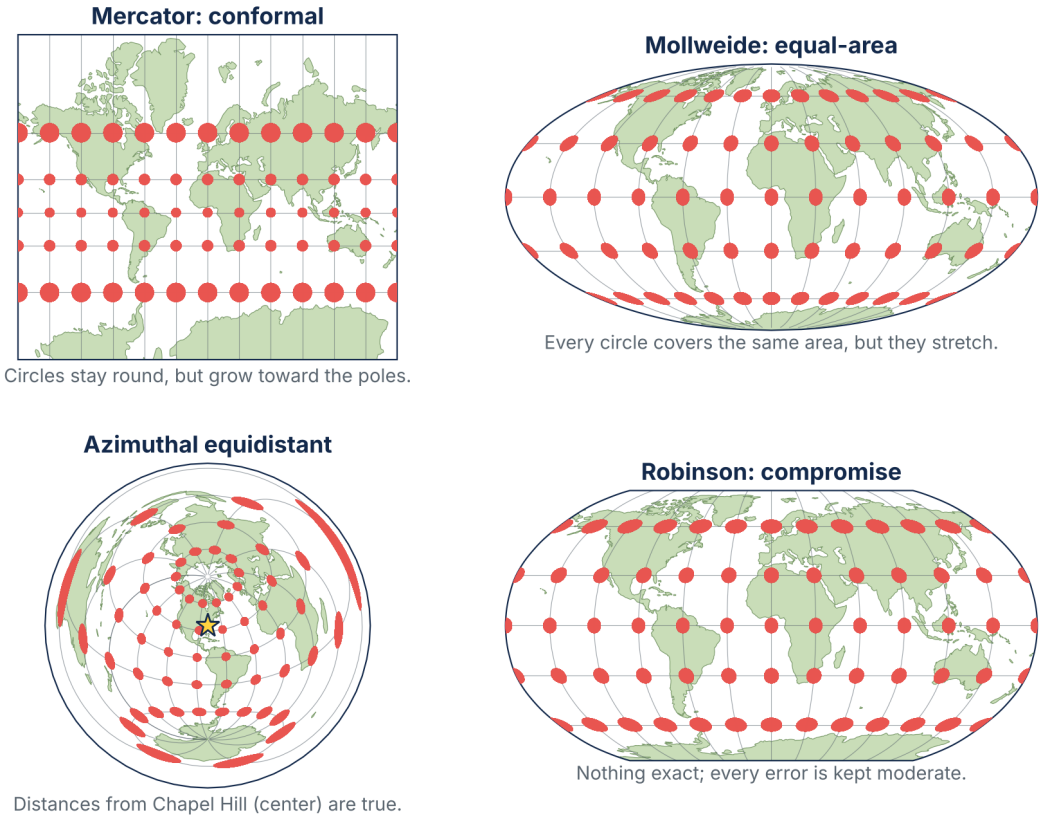


Figure 4.3: Four projections of the same world, with identical circles (each 500 km across on the ground) drawn every 30°. On the conformal Mercator they stay round but grow toward the poles. On the equal-area Mollweide they keep their area but stretch. On an azimuthal equidistant map centered on Chapel Hill, distances from the center are true. On the Robinson compromise, no property is exact.

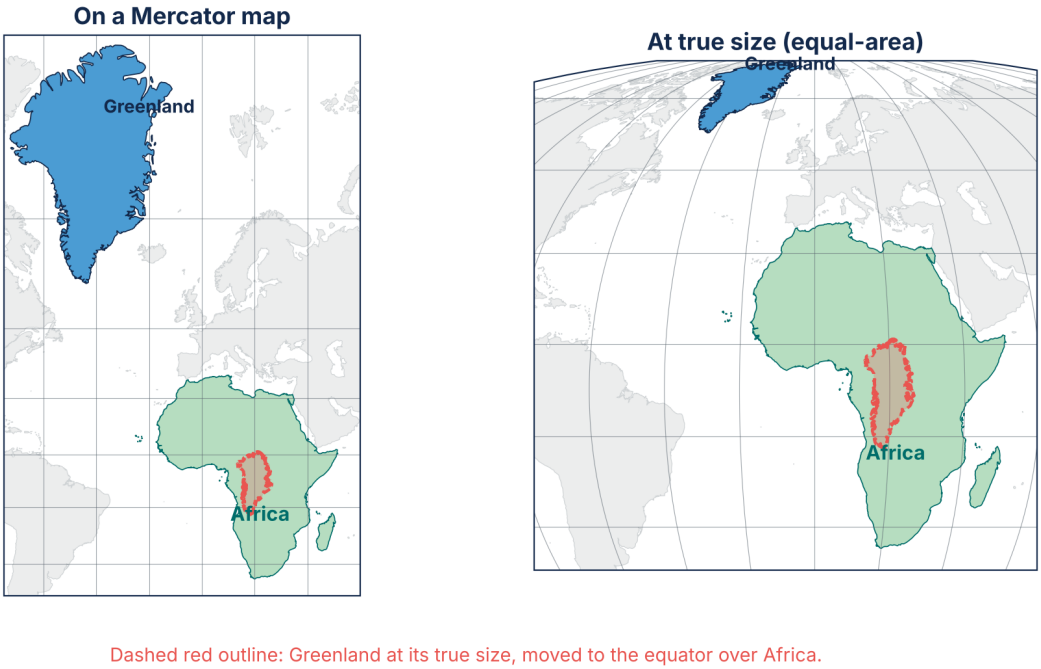


Figure 4.4: Greenland (blue) and Africa (green) on a Mercator map (left) and an equal-area map (right). The dashed red outline is Greenland at its true size, moved to the equator over Africa.

Did you know..?

In 2017, the Boston Public Schools began replacing Mercator wall maps in their classrooms with the equal-area **Gall-Peters** projection. Teachers said the Mercator had shrunk Africa and South America in students' minds for generations. The Gall-Peters itself is controversial: when the historian Arno Peters promoted it in the 1970s as a fairer map, cartographers pointed out that a Scottish clergyman, James Gall, had published the same projection in 1855, and that it badly stretches shapes near the equator. Many cartographers would rather see a good compromise projection on the wall. Either way, the choice of a map projection turns out to be a choice about how people picture the world.

4.2.4 Web Mercator: for display, not measurement

Look at almost any web map, such as Google Maps, OpenStreetMap, Bing Maps, or an Esri basemap, and you are looking at **Web Mercator**, code **EPSG:3857**. It treats the Earth as a sphere while using WGS84 latitude and longitude, which makes the math fast. Its great strength is that the whole world fits in a square, which can be cut into square **tiles** at every zoom level: one tile at zoom 0, four at zoom 1, sixteen at zoom 2, and so on (Figure 4.5). Your browser only downloads the tiles you are looking at, which is why web maps scroll and zoom so smoothly.

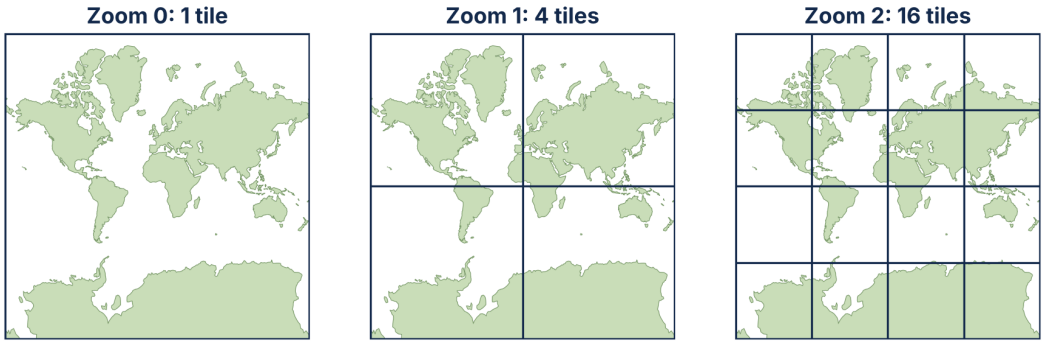


Figure 4.5: Web Mercator divides the world into square tiles. Each zoom level splits every tile into four.

To make the world square, Web Mercator cuts the map off at about 85.05° north and south, so the North and South Poles are not on the map at all. And it inherits all of Mercator's area enlargement. Use it for:

- web basemaps, and
- smooth zooming and panning.

Avoid it for:

- measuring area or distance,
- comparing regions at different latitudes, and
- anything near the poles.

If you collect data from a web map, reproject it to an equal-area or a local projection, such as UTM or State Plane, before you calculate areas or distances.

4.3 How projections are made

4.3.1 Three developable surfaces

A **developable surface** is a shape that can be unrolled flat without stretching or tearing. Three are used for map projections: the plane, the cylinder, and the cone (Figure 4.6).

- **Planar** (or **azimuthal**): a flat plane touches the globe at one point. Distortion grows in rings around that point, so planar projections suit the polar regions, or maps centered on one place.
- **Cylindrical**: a cylinder wraps the globe. In its normal position it touches the equator, which suits the tropics, or a world map. Turned on its side, as in the Transverse Mercator below, it suits regions that run north to south.
- **Conic**: a cone sits over one hemisphere like a hat and touches it along a parallel. Conic projections suit mid-latitude regions that run east to west, such as the contiguous United States.

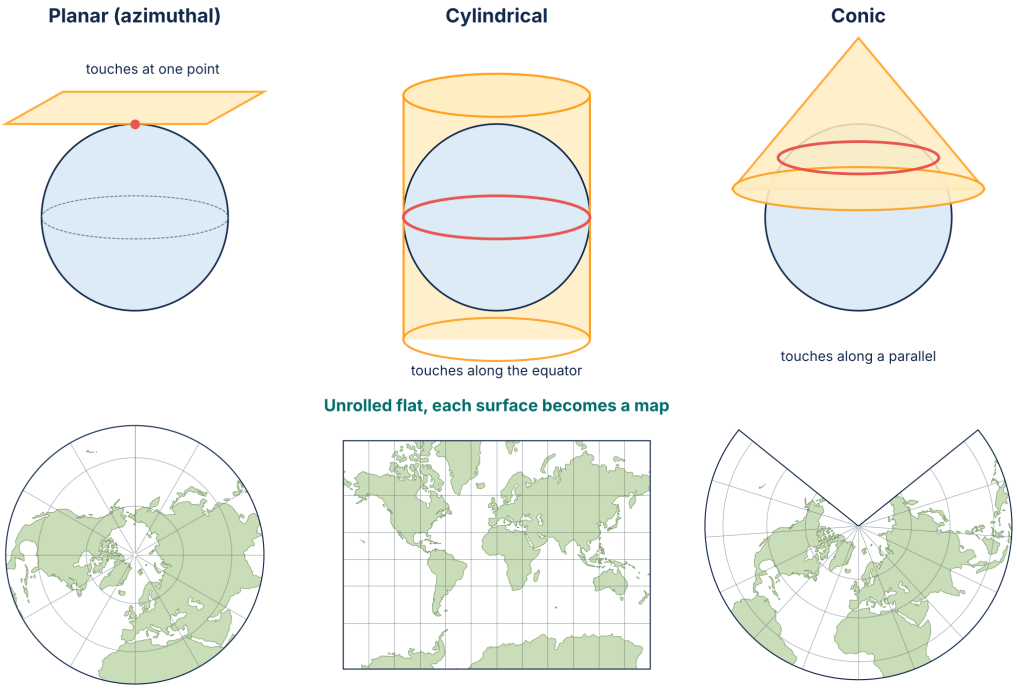


Figure 4.6: The three developable surfaces (top) and the maps they make when unrolled flat (bottom): a polar azimuthal map, a cylindrical (Mercator) map, and a conic (Lambert Conformal Conic) map.

The line where a surface is sliced so it can be unrolled becomes the outer edge of the map.

4.3.2 Tangent and secant

A projection surface can touch the globe in one place or cut through it (Figure 4.7).

- A **tangent** surface touches the Earth along one line, or at one point for a planar projection.
- A **secant** surface cuts through the Earth along two lines, or along a circle for a planar projection.

Where the surface meets the Earth, scale is exact. These are the **standard lines** (or a standard point). When standard lines follow parallels of latitude, as on a conic projection, they are called **standard parallels**. The ratio of distance on the map to distance on the ground, compared with the map's stated scale, is the **scale factor**: exactly 1 on a standard line, less than 1 where the map shrinks the Earth, and more than 1 where it stretches it.

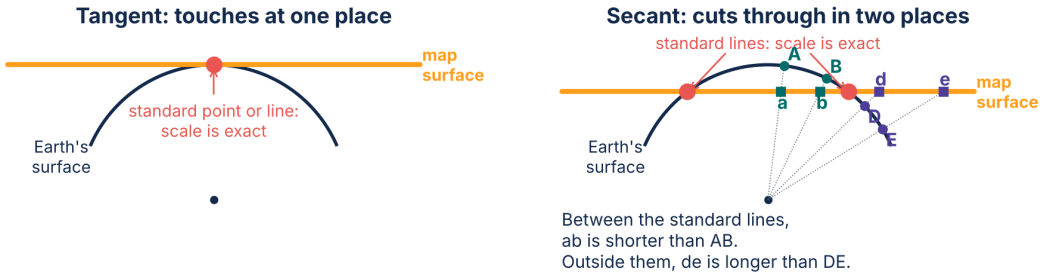


Figure 4.7: Tangent (left) and secant (right) projection surfaces in cross-section. Scale is exact where the surface meets the Earth. Between a secant surface's standard lines, map distances are a little too short (ab is shorter than AB); outside them, a little too long (de is longer than DE).

Why cut through the Earth on purpose? A secant surface spreads the error across a wider area. A tangent surface is perfect on one line and gets steadily worse everywhere else. A secant surface is slightly too small between its standard lines and slightly too large outside them, which keeps the largest error over the whole map much smaller. Good projection design puts the standard lines so that the area you care about stays close to zero error.

4.3.3 Projections are named for what they preserve

Putting it together, projections are grouped by the property they keep (Table 4.1).

Table 4.1: Projections are named for the property they preserve. No projection can be both conformal and equal-area.

Property preserved	Class	Examples	Typical use
Shape (local angles)	Conformal	Mercator, Lambert Conformal Conic, Transverse Mercator	Navigation, topographic and engineering maps
Area	Equal-area	Albers Equal-Area Conic, Mollweide	Thematic maps of density or land cover
Distance from one or two points	Equidistant	Azimuthal Equidistant, Equidistant Conic	Distances from a city or an airport
Direction from a center point	Azimuthal	Gnomonic, Azimuthal Equidistant	Polar maps, great-circle routes
None exactly; all errors moderate	Compromise	Robinson, Winkel Tripel	World maps for general reference

On a **gnomonic** projection, every great circle (Chapter 3) is a straight line. Navigators once used gnomonic charts to plot the shortest route between two ports, then copied it onto a Mercator chart as a series of short rhumb lines they could actually steer.

Did you know..?

For most of the twentieth century, the National Geographic Society, whose wall maps hung in classrooms across the country, used the Van der Grinten projection, which enlarges the polar regions almost as much as Mercator. In 1988 it switched to the Robinson projection, which Arthur Robinson had designed in 1963 by eye, adjusting the shape until the world simply *looked* right, rather than starting from an equation. Ten years later, in 1998, National Geographic switched again, to the **Winkel Tripel**, which has even less overall distortion and is still its standard world map.

4.4 Projections for the United States

4.4.1 Lambert Conformal Conic

The **Lambert Conformal Conic** projection, published by Johann Heinrich Lambert in 1772, uses a secant cone that cuts the ellipsoid along two standard parallels. Scale is exact on both, slightly too small between them, and too large outside them. For the contiguous United States, standard parallels at 33° N and 45° N keep the scale error within about 1 percent across most of the country (Figure 4.8, left).

On the map, meridians are straight lines that converge toward a point beyond the pole, and parallels are arcs of circles centered on that point, which gives the map its fan shape. Because it is conformal, Lambert Conformal Conic keeps local angles and shapes. That is why it is used for U.S. aeronautical charts, where pilots need true angles, and for the State Plane zones of states that are wider east to west than north to south, like North Carolina and Tennessee.

4.4.2 Transverse Mercator

The **Transverse Mercator** is the Mercator turned 90°. Instead of touching the equator, the cylinder touches the Earth along a chosen *central meridian*. Scale is true along that meridian, or along two lines parallel to it in the secant version, and error grows east and west of it (Figure 4.8, right). Like the Mercator, it is conformal.

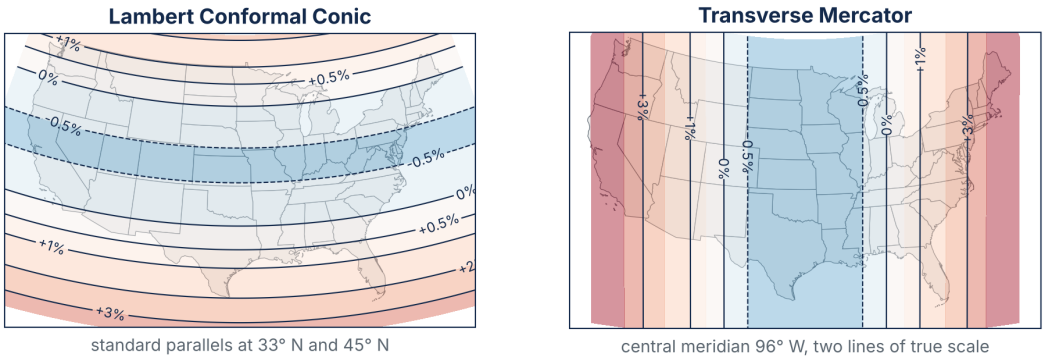


Figure 4.8: Scale error across the contiguous United States on a Lambert Conformal Conic projection with standard parallels at 33° N and 45° N (left), and a secant Transverse Mercator with a central meridian at 96° W (right). Contours show the error in percent: 0% is a line of true scale, blue areas are shrunk, and red areas are stretched.

A single Transverse Mercator for the whole country would be too distorted at the coasts. But in a narrow strip along its central meridian it is very accurate. So it is used in narrow zones: the State Plane zones of north–south states such as Illinois and Vermont, and the sixty zones of UTM.

4.4.3 Universal Transverse Mercator (UTM)

The **Universal Transverse Mercator (UTM)** system divides the world, from 80° S to 84° N, into **60 zones**, each **6° of longitude** wide. The zones are numbered eastward from 180° longitude, so zone 1 runs from 180° W to 174° W. The contiguous United States falls in zones 10 to 19 (Figure 4.9). Each zone has a north and a south half, such as 17N and 17S, on either side of the equator.

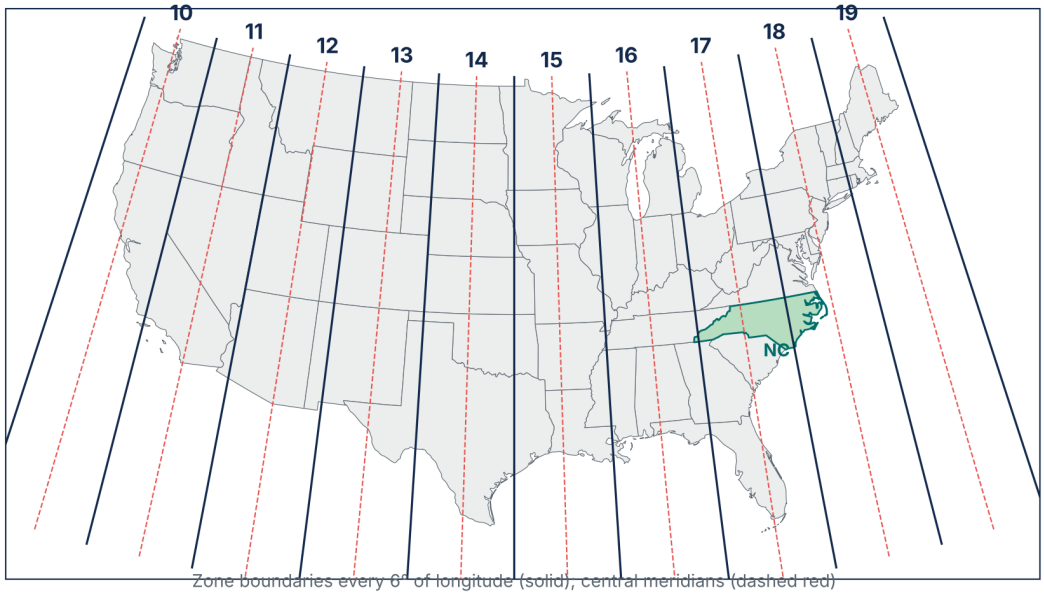


Figure 4.9: UTM zones 10 to 19 across the contiguous United States. Each zone is 6° of longitude wide (solid lines), with its central meridian (dashed red) down the middle. North Carolina (green) straddles zones 17 and 18.

Each zone is its own secant Transverse Mercator projection:

- The **scale factor on the central meridian is 0.9996**, so the two lines of true scale sit about 180 km on either side of it, and the error stays below about 1 part in 1,000 across the zone.
- Coordinates are in **meters**: an easting and a northing.
- The central meridian is given a **false easting of 500,000 m**, so every easting in the zone is positive. In the Northern Hemisphere, northings are measured from the equator; in the Southern Hemisphere, the equator is given a false northing of 10,000,000 m, so southern northings stay positive too.

A UTM coordinate needs its zone to make sense. Chapel Hill is at about 676,000 m E, 3,976,000 m N in zone 17N. The same numbers in zone 18N would land about 540 km farther east, in the Atlantic Ocean, because each zone measures its eastings from its own central meridian.

North Carolina spans two zones, split at 78° W: zone **17N** to the west, including Asheville, Charlotte, Chapel Hill, and Raleigh, and zone **18N** to the east, including Greenville, Wilmington, and the Outer Banks (Figure 4.10). Data that crosses a zone boundary is a common

source of error. For a project that spans both zones, pick one zone for the whole study area, or use State Plane instead.

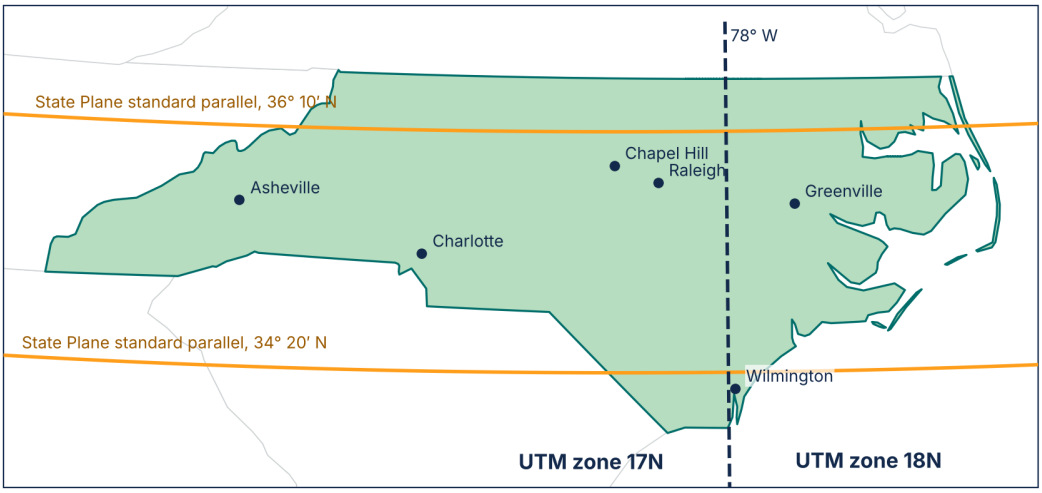


Figure 4.10: North Carolina's coordinate system zones. The dashed line at 78° W splits UTM zones 17N and 18N. The orange lines are the two standard parallels of the state's single State Plane zone, at 34° 20′ N and 36° 10′ N.

4.4.4 The State Plane Coordinate System

The **State Plane Coordinate System (SPCS)** was designed in the 1930s by the U.S. Coast and Geodetic Survey, now the National Geodetic Survey, so that surveyors and engineers could work with flat *x, y* coordinates with very little distortion. It exists in two main versions, based on NAD27 and on NAD83.

- Each state is covered by one or more zones, about 120 in all, whose boundaries follow state and county lines.
- States that are wide east to west, like Tennessee, use Lambert Conformal Conic; states that are long north to south, like Illinois, use Transverse Mercator. The Alaska panhandle uses an oblique Mercator.
- Zones are kept small enough that scale error stays at **1 part in 10,000 or better**: no more than 1 meter in 10 kilometers.

North Carolina has a single State Plane zone, a Lambert Conformal Conic with standard parallels at 34° 20′ N and 36° 10′ N (Figure 4.10). In a GIS you will usually meet it as **EPSG:32119**, NAD83 / North Carolina in meters, or **EPSG:2264**, the same projection in U.S. survey feet. They are the same projection with different units, so always check which units a dataset uses. Most North Carolina state and county data, including NC OneMap's, is in State Plane feet.

Think about it

Your project maps every stream in Dare County, on the Outer Banks. Which would you choose: UTM zone 17N, UTM zone 18N, or NC State Plane? What if your project covered the whole state?

Show the answer

For Dare County alone, UTM zone 18N or NC State Plane both work well; zone 17N would put the county outside its own zone. For the whole state, choose NC State Plane, which was designed to cover North Carolina in one zone.

4.5 Coordinate reference systems and EPSG codes

In software, a projection never travels alone. A **coordinate reference system (CRS)** bundles a datum, a projection with its parameters (the standard parallels, central meridian, scale factor, false easting, and so on), and the units.

There are two kinds of CRS:

- A **geographic coordinate system** gives positions in latitude and longitude, in **degrees**, on a datum (Chapter 3).
- A **projected coordinate system** gives positions in *x* and *y*, in **linear units** such as meters or feet, on a projection built on a datum.

Since there are thousands of coordinate reference systems, each is given a short numeric code. The **EPSG** registry, started by the European Petroleum Survey Group and now maintained by the International Association of Oil and Gas Producers (IOGP), lists them all, and QGIS, ArcGIS Pro, PostGIS, and most web tools use its codes (Table 4.2).

Table 4.2: EPSG codes you will meet most often in North Carolina.

EPSG code	Name	Type and use	Units
4326	WGS 84	Geographic: GPS and web data	Degrees
4269	NAD83	Geographic: U.S. federal data	Degrees
3857	WGS 84 / Pseudo-Mercator	Projected: web maps	Meters
26917	NAD83 / UTM zone 17N	Projected: NC west of 78° W	Meters
26918	NAD83 / UTM zone 18N	Projected: NC east of 78° W	Meters
32119	NAD83 / North Carolina	Projected: NC State Plane	Meters
2264	NAD83 / North Carolina (ftUS)	Projected: NC State Plane	U.S. survey feet
5070	NAD83 / Conus Albers	Projected: equal-area, contiguous U.S.	Meters

Two rules follow from this table:

1. **Never measure distance or area in a geographic CRS.** A degree of longitude is 111 km at the equator and 90 km in Chapel Hill (Chapter 3), so “square degrees” mean nothing on the ground. Measure in a projected CRS in meters or feet.
2. **Never measure in Web Mercator (3857).** It uses meters, but its meters are stretched, by about 24 percent in each direction at Chapel Hill’s latitude and more farther north.

When layers won’t line up, the first thing to check is whether each one has the correct CRS defined.

4.6 Choosing a projection

So what is the best projection? There is no single answer. The best projection depends on two questions: **where is your study area**, and **what will you measure?**

1. Pick a projection designed for your area. A local system keeps distortion smallest.

- Working in eastern North Carolina, east of 78° W? UTM zone 18N.
- Working in Chapel Hill, or across North Carolina? North Carolina State Plane.
- Working across North America or the contiguous United States? A North American Lambert Conformal Conic, or an Albers Equal-Area Conic if you are comparing areas.

2. Pick a projection that preserves the properties you need, where you need them.

- Measuring area, or mapping density? Choose **equal-area**.
- Measuring distance from a point? Choose **equidistant**, such as an azimuthal equidistant centered on that point.
- Measuring direction from a starting point? Choose **azimuthal**, centered on that point.
- Need shapes and angles to look right? Choose **conformal**. A conformal projection does not preserve area, so it is not a substitute for equal-area when you measure area.

3. Using elevation data? Pick a projection whose horizontal units match your elevation units. Slope and hillshade calculations divide a change in height by a distance across the ground. A digital elevation model with heights in feet on a projection in meters produces slopes that are off by a factor of about 3.3, the number of feet in a meter.

For a small study area, such as a county or a city, any good local projection works, and the differences among them are smaller than the errors in most data. The bigger your study area, the more the choice matters.

4.7 Projections in QGIS and ArcGIS Pro

Both programs reproject layers **on the fly**: each layer keeps its own CRS, and the software converts it to the map's CRS as it draws. In QGIS, the project's CRS is shown in the lower-right corner of the window; click it to change it. In ArcGIS Pro, the map's CRS is set in the map's Properties, on the Coordinate Systems tab. By default, both programs take the CRS of the first layer you add.

On-the-fly reprojection is convenient, but it can hide problems. Two tasks are easy to confuse:

- **Defining** a projection tells the software what CRS a layer is *already* in, when that information is missing or wrong, for example when a shapefile has lost its `.prj` file (Chapter 2). The coordinates don't change. In QGIS this is **Assign Projection** (or setting the CRS in Layer Properties); in ArcGIS Pro it is the **Define Projection** tool.
- **Projecting** (or reprojecting) a layer converts its coordinates into a new CRS and saves a new layer. In QGIS this is **Reproject Layer**; in ArcGIS Pro it is the **Project** tool, or **Project Raster** for rasters.

Defining the wrong CRS is a classic mistake. A layer in latitude and longitude that is wrongly defined as State Plane feet is read as a few feet from the projection's origin, so it shows up as a tiny speck far from where it belongs; in a world map, one wrongly defined in meters appears near 0° latitude and 0° longitude, in the ocean off West Africa. A layer in State Plane

feet wrongly defined as degrees may not appear at all, because its numbers, in the millions, are far outside the range of latitude and longitude. If a layer shows up in the wrong place, check its defined CRS first. Project a layer, rather than relying on on-the-fly reprojection, before you measure lengths or areas, run overlay tools, or share the data.

4.8 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

4.9 Summary

- A **map projection** is a set of equations that converts latitude and longitude into flat *x, y* coordinates. It sits on top of an ellipsoid and a datum.
- **Every flat map distorts** shape, area, distance, or direction. **Conformal** projections keep local shapes, **equal-area** projections keep area, **equidistant** projections keep distance from one or two points, and **azimuthal** projections keep direction from a center. No projection is both conformal and equal-area. **Tissot's indicatrix** shows the distortion.
- **Mercator** is conformal and great for navigation, but it enlarges areas toward the poles: four times at 60°. **Web Mercator** (EPSG:3857), the projection of web maps, is for display, not measurement.
- Projections are built on three **developable surfaces**: the plane, the cylinder, and the cone. A **tangent** surface touches the Earth; a **secant** surface cuts it, spreading the error. Scale is exact on **standard lines**, and error grows away from them.
- **Lambert Conformal Conic** suits mid-latitude regions that run east to west; **Transverse Mercator** suits regions that run north to south.
- **UTM** divides the world into 60 zones, 6° wide, each a secant Transverse Mercator in meters. North Carolina spans zones 17N and 18N, split at 78° W.
- The **State Plane Coordinate System** has about 120 zones, with error held to 1 part in 10,000. North Carolina's single zone is a Lambert Conformal Conic, EPSG:32119 in meters or 2264 in feet.
- A **coordinate reference system** bundles a datum, a projection, and units, and is identified by an **EPSG code**. **Geographic** systems are in degrees; **projected** systems are in meters or feet. Never measure in degrees or in Web Mercator.
- Choose a projection designed for your area and one that preserves what you will measure. **Define** a projection to fix missing CRS information; **project** a layer to convert its coordinates.

4.10 Exercises

1. **Tissot at home.** Open QGIS, add a world countries layer (Natural Earth works well), and change the project CRS to Mercator, Mollweide, Robinson, and an azimuthal equidistant centered on Chapel Hill. Describe how Greenland, Antarctica, and Africa change on each.
2. **True size.** Use The True Size Of… (https://www.thetruesize.com) to drag North Carolina, Texas, and Greenland to the equator and to the Arctic. Describe what happens and explain why, using the word *conformal*.

3. **Measure an area three ways.** In QGIS or ArcGIS Pro, calculate the area of North Carolina's counties in EPSG:3857, EPSG:32119, and EPSG:5070. Which answers agree? Which is far off, and by about how much? Why?
4. **Find the zone.** Which UTM zone are each of these in: Asheville, Wilmington, Denver, Seattle, and your hometown? (Zone number = the whole number part of (longitude + 180) ÷ 6, plus 1.)
5. **Read a CRS.** Look up EPSG:32119 at https://epsg.io/32119. Find its datum, projection, standard parallels, latitude of origin, central meridian, false easting, and units.
6. **Choose a projection.** For each project, choose a projection or CRS and explain why: (a) the area of every wetland in Orange County, NC; (b) a map of flight distances from RDU airport; (c) a world map of population density; (d) a map of the Arctic Ocean's sea ice; (e) a statewide map of North Carolina for a highway department.
7. **Wrong CRS.** A classmate's layer of North Carolina schools shows up as a tiny cluster in the ocean off West Africa, near 0° latitude and 0° longitude. What has gone wrong, and how would you fix it?
8. **Units and slope.** You have a digital elevation model with elevations in feet, in a projection in meters. What will happen to a slope calculation? How would you fix it?

4.11 Further reading and online resources

4.11.1 Books and open textbooks

- *GIS Fundamentals: A First Text on Geographic Information Systems*, by Paul Bolstad (Bolstad 2022). Chapter 3, "Geodesy, Projections, and Coordinate Systems," covers developable surfaces, UTM, and State Plane in depth.
- *Understanding GIS through Sustainable Development Goals*, by Paul Holloway (Holloway 2023). Its chapter on coordinate systems and projections includes worked examples in QGIS and ArcGIS Pro.
- *Map Projections: A Working Manual*, by John P. Snyder (Snyder 1987), USGS Professional Paper 1395. The standard reference for the mathematics and use of dozens of projections: https://pubs.usgs.gov/publication/pp1395.
- *Flattening the Earth: Two Thousand Years of Map Projections*, by John P. Snyder (University of Chicago Press, 1993), the history of map projections.
- *How to Lie with Maps*, by Mark Monmonier (3rd ed., University of Chicago Press, 2018). Its chapter on projections shows how a projection can shape a message.
- *Rhumb Lines and Map Wars: A Social History of the Mercator Projection*, by Mark Monmonier (University of Chicago Press, 2004), on the Mercator and the Peters debate.
- *The Nature of Geographic Information*, by David DiBiase and colleagues at Penn State. Chapter 2 covers plane coordinate systems, UTM, and State Plane: https://www.e-education.psu.edu/natureofgeoinfo/c2.html.
- *Intro to GIS and Spatial Analysis*, by Manuel Gimond. The chapter "Coordinate Systems" includes a gallery of projections: https://mgimond.github.io/Spatial/chp09_0.html.
- *Essentials of Geographic Information Systems*, by Jonathan Campbell and Michael Shin. Chapter 2 covers map projections: https://open.umn.edu/opentextbooks/textbooks/67.

4.11.2 Understanding map projections

- USGS, *Map Projections*, a poster comparing the properties and uses of common projections: https://pubs.usgs.gov/publication/70047422.

- The GIS&T Body of Knowledge, "Map Projections," "Map Projection Properties," "Map Projection Classes," and "Plane Coordinate Systems": https://www.ucgis.org/site/gis-t-body-of-knowledge.
- Axis Maps, "Cartography Guide: Map Projections": https://www.axismaps.com/guide/map-projections.
- PROJ, the open-source library behind QGIS's projections, with an illustrated list of every projection it supports: https://proj.org/en/stable/operations/projections/index.html.
- Wolfram MathWorld, "Map Projection," with the equations for common projections: https://mathworld.wolfram.com/MapProjection.html.
- Jason Davies, interactive map projection transitions and demos: https://www.jasondavies.com/maps/.
- Projection Wizard, a tool that recommends a projection for your area and purpose: https://projectionwizard.org.
- Map Projection Transitions, an animated tour of projections: https://www.jasondavies.com/maps/transition/.
- xkcd, "What your favorite map projection says about you": https://xkcd.com/977/.

4.11.3 Mercator, Web Mercator, and the true size of countries

- The True Size Of…, drag countries around a Mercator map to see their real size: https://www.thetruesize.com.
- Neil Kaye's animated "Mercator projection vs. the true size of each country," widely shared since 2019. Search for "Neil Kaye true size Mercator."
- Engaging Data, "Mercator Puzzle Redux," a game of placing countries at their true size: https://engaging-data.com/country-sizes-mercator/.
- OpenStreetMap Wiki, "Web Mercator" and "Slippy map tilenames," on how web map tiles work: https://wiki.openstreetmap.org/wiki/Slippy_map_tilenames.
- *The Guardian*, "Boston public schools map switch aims to amend 500 years of distortion" (March 2017), on the switch to Gall-Peters.
- *The West Wing*, season 2, "Somebody's Going to Emergency, Somebody's Going to Jail" (2001), the famous scene in which cartographers argue for the Gall-Peters projection.

4.11.4 UTM, State Plane, and coordinate reference systems

- NGS, State Plane Coordinate System of 2022 (SPCS2022) and the State Plane zones of 1983: https://geodesy.noaa.gov/SPCS/.
- NGS, *State Plane Coordinate System of 1983*, NOAA Manual NOS NGS 5, by James Stem: https://geodesy.noaa.gov/library/pdfs/NOAA_Manual_NOS_NGS_0005.pdf.
- epsg.io, a searchable list of coordinate reference systems; for example, North Carolina State Plane in meters: https://epsg.io/32119.
- The EPSG Geodetic Parameter Dataset, the official registry: https://epsg.org.
- Spatial Reference, another searchable list of coordinate systems with their definitions in many formats: https://spatialreference.org.

4.11.5 Software and tutorials

- QGIS User Guide, "Working with Projections": https://docs.qgis.org/latest/en/docs/user_manual/working_with_projections/working_with_projections.html.

- QGIS, *A Gentle Introduction to GIS*, “Coordinate Reference Systems”: https://docs.qgis.org/latest/en/docs/gentle_gis_introduction/coordinate_reference_systems.html.
- QGIS Training Manual, “Reprojecting and Transforming Data”: https://docs.qgis.org/latest/en/docs/training_manual/vector_analysis/reproject_transform.html.
- ArcGIS Pro, “Define Projection” and “Project” tools: https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/define-projection.htm and https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/project.htm.
- ArcGIS Pro, “Specify a coordinate system for a map”: https://pro.arcgis.com/en/pro-app/latest/help/mapping/properties/specify-a-coordinate-system.htm.
- Learn ArcGIS, “Choose the right projection”: https://learn.arcgis.com/en/projects/choose-the-right-projection/.
- Natural Earth, free world data for trying out projections: https://www.naturalearthdata.com.

5 Attribute Data and Queries

Learning objectives

By the end of this chapter you should be able to:

1. Describe how an attribute table stores data, with one record per feature and one attribute per column.
2. Choose a data type for a field in QGIS or ArcGIS Pro, and explain how the number of bits behind a field limits what it can hold.
3. Tell the four levels of measurement apart (nominal, ordinal, interval, and ratio) and say which questions each one supports.
4. Write SQL WHERE clauses with comparison operators, AND, OR, NOT, brackets, LIKE, BETWEEN, IN, and IS NULL.
5. Predict which records a query selects, using sets and Venn diagrams, and spot queries that can only select everything or nothing.
6. Explain how NULL values behave in a query.
7. Use a selection: zoom to it, summarize it, calculate a field from it, and export it.

Chapters 2 to 4 were about *where*: how a GIS stores geometry, and how coordinates pin it to the Earth. This chapter is about the other half of GIS data, the half that says *what*. A polygon tells you where a county is. Its attributes tell you its name, how many people live there, how old they are, and how many homes stand empty. Most of the questions people bring to a GIS start with attributes: *Which counties lost population? Which parcels are zoned commercial and larger than five acres? Which schools have no address?*

To answer them, you need to know how attributes are stored, what kinds of values they hold, and how to write a **query**, a precise question the software can test against every record. We will finish with a way to check your own queries using sets, and with the trap of missing values.

Five terms carry the whole chapter:

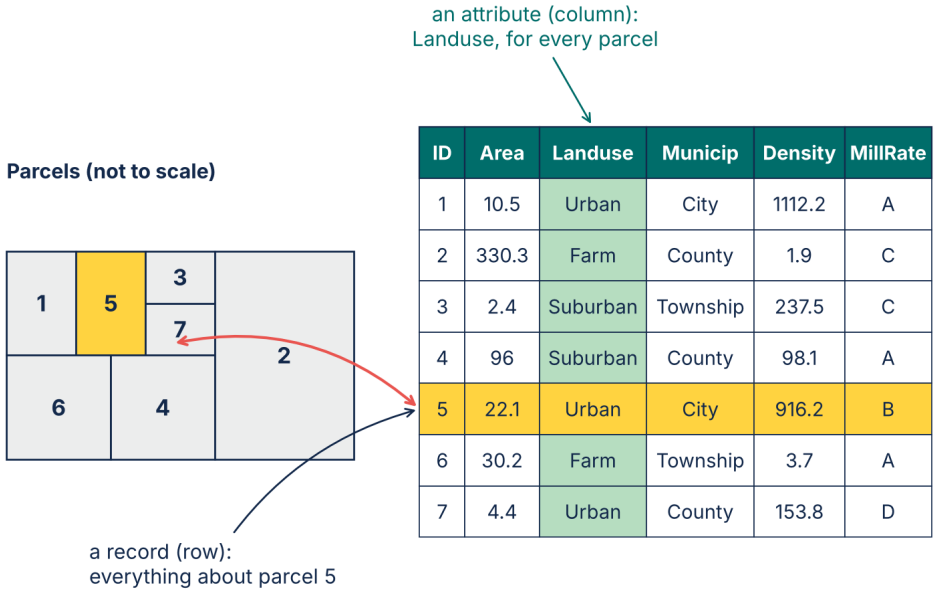
- A **table** is a grid that holds the attributes of every feature in a layer.
- An **attribute**, also called a **field** or a column, is one variable, measured the same way for every feature.
- A **record**, or row, is everything known about one feature.
- A **data type** is how a field is stored: as text, an integer, a decimal number, a date, and so on.
- A **query**, usually written in **SQL** (Structured Query Language), is a written condition that selects records.

5.1 Attribute data

5.1.1 Attribute tables

In Chapter 2, data abstraction turned a wetland into a data model: polygons for the geometry, and a table for everything else. That table is the **attribute table**, the non-spatial half of a vector layer. It follows three rules (Figure 5.1):

- **One row per feature.** Every parcel on the map has exactly one record in the table, and every record has a parcel.
- **One column per attribute.** Each column holds one kind of measurement, such as area or land use, for every feature.
- **The table and the map are linked.** Select a record, and its feature lights up on the map; select a feature, and its record lights up in the table. The table is where you ask the question; the map is where the answer is drawn.



ID	Area	Landuse	Municip	Density	MillRate
1	10.5	Urban	City	1112.2	A
2	330.3	Farm	County	1.9	C
3	2.4	Suburban	Township	237.5	C
4	96	Suburban	County	98.1	A
5	22.1	Urban	City	916.2	B
6	30.2	Farm	Township	3.7	A
7	4.4	Urban	County	153.8	D

Figure 5.1: Seven parcels and their attribute table. A record is a row: everything known about one parcel, here parcel 5. An attribute is a column: one variable for every parcel, here Landuse. Selecting record 5 in the table also selects parcel 5 on the map. (The parcels are not drawn to scale.)

We will use these seven parcels throughout the chapter. Their attributes are:

- **ID**, a number that identifies each parcel.
- **Area**, in acres.
- **Landuse**: Urban, Suburban, or Farm.
- **Municip**, the kind of local government the parcel is in: City, Township, or County.
- **Density**, in people per square mile.
- **MillRate**, a property tax class from A to D.

Software uses several words for the same things. A column may be called an attribute, a field, or an item. A row may be called a record, a row, or (because each row is one feature) a feature. In this book, *field* and *attribute* mean the same thing, and so do *record* and *row*.

In QGIS, open a layer's table by right-clicking it in the Layers panel and choosing **Open Attribute Table**. In ArcGIS Pro, right-click the layer in the Contents pane and choose **Attribute Table**. Click a column header to sort the table by that column; click again to sort the other way. Sorting a layer of world cities by population, largest first, is a quick way to explore a new dataset.

5.1.2 Data types

Every field has a **data type**, chosen when the field is created. The data type decides what a field can store and how much space each value takes. QGIS and ArcGIS Pro name their types a little differently (Table 5.1).

Table 5.1: Common field data types. Which types are available also depends on the format: a shapefile, a GeoPackage, a file geodatabase, or a database such as PostGIS. ArcGIS Pro's Big Integer, Date Only, and Time Only types need version 3.2 or later.

What it stores	QGIS	ArcGIS Pro	Range or example
Text, any characters	Text (string)	Text	John Doe, 27514
Small whole numbers (16-bit)	(none)	Short Integer	−32,768 to 32,767
Whole numbers (32-bit)	Integer (32 bit)	Long Integer	−2,147,483,648 to 2,147,483,647
Large whole numbers (64-bit)	Integer (64 bit)	Big Integer	about ±9.2 × 10^{18} in QGIS; ±9,007,199,254,740,991 in ArcGIS Pro
Decimal numbers, about 7 significant digits	(none)	Float	about ±3.4 × 10^{38}
Decimal numbers, about 15 significant digits	Decimal number (double)	Double	about ±1.8 × 10^{308}
True or false	Boolean	(none; use Short Integer 0 or 1)	true, false
Dates and times	Date, Date & Time	Date, Date Only, Time Only	12/31/2012 11:59:59 PM
A unique ID for each record	(fid)	Object ID	1, 2, 3, …
The feature's shape	(geometry)	Geometry (Shape)	point, line, polygon

Two terms describe numeric fields:

- **Precision** is the total number of digits a field can hold.
- **Scale** is how many of those digits fall after the decimal point.

A field with a precision of 6 and a scale of 2 can hold numbers up to 9,999.99. In ArcGIS Pro, you choose the type, and for some formats the precision and scale, when you create a field, and **they can't be changed later**. To change a field's type, you add a new field of the right type and calculate the values into it.

Some rules of thumb for choosing a type:

- Store **identifiers that are never used in arithmetic as text**, even if they look like numbers. ZIP codes, FIPS codes, phone numbers, and parcel IDs all belong in text fields. Stored as a number, the ZIP code of Boston's City Hall, 02201, becomes 2201, and Orange County's FIPS code, 37135, can no longer be matched to another table that stores it as text.
- Use an **integer** type for counts, such as population or number of rooms.
- Use **Double** for measurements that need decimals, such as area or density. Float saves a little space, but it keeps only about 7 significant digits, so it can round large numbers.

5.1.3 Why fields have limits: binary

Why do field types have such strange limits, like 32,767 or 2,147,483,647? Because everything in an attribute table is ultimately stored in **binary**, as ones and zeros, and each field gets a fixed number of **bits** (binary digits).

In binary, each column is worth twice the one to its right: ones, twos, fours, eights, and so on (Figure 5.2). So the binary number 1101 means 8 + 4 + 0 + 1 = 13.

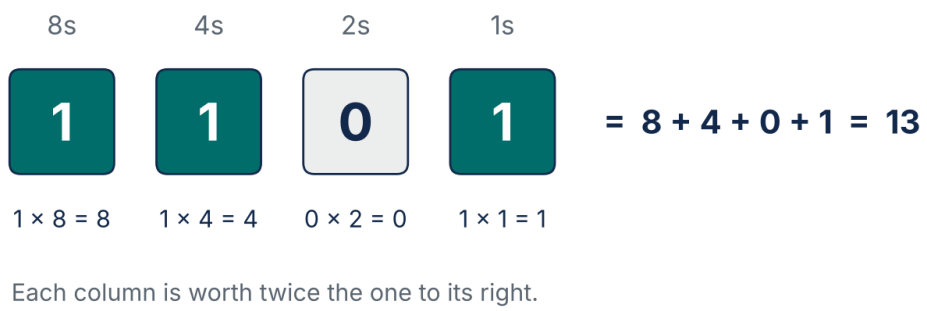


Figure 5.2: The binary number 1101. Each column is worth twice the one to its right, so 1101 is 8 + 4 + 0 + 1 = 13.

With n bits you can write 2^n different values. If one bit is used for the sign, plus or minus, half of them are negative:

- A **16-bit** short integer has $2^{16} = 65{,}536$ values, from −32,768 to 32,767.
- A **32-bit** integer has about 4.3 billion values, from about −2.1 billion to 2.1 billion.
- A **64-bit** integer has $2^{64} = 18{,}446{,}744{,}073{,}709{,}551{,}616$ values, about $\pm 9.22 \times 10^{18}$.

Modern computers process data in 64-bit *words*: that is what "a 64-bit processor" means. It refers to how many bits the processor handles at once, not to how much memory the computer has. An integer field is only as wide as the bits behind it, so the population of a county fits easily in a 32-bit integer, but the population of the world, about 8.2 billion, does not.

Did you know..?

In December 2014, the music video for PSY's "Gangnam Style" became the first video on YouTube to be watched more than 2,147,483,647 times, the largest number a signed 32-bit integer can hold. YouTube's engineers had seen it coming and had already switched the view counter to a 64-bit integer, which can count past nine quintillion. YouTube joked that it had never thought a video would be watched more times than a 32-bit integer could count. The same limit is behind the "Year 2038

problem”: many older computer systems count time as the number of seconds since January 1, 1970, in a signed 32-bit integer, which will run out on January 19, 2038.

5.1.4 Levels of measurement

The data type says what the computer can *store*. The **level of measurement** says what you may legitimately *ask*. They are different questions. The number 3 stored in an integer field could be a count of bedrooms, a rank, a temperature, or a code for “forest,” and the arithmetic that makes sense is different for each. The psychologist S. S. Stevens described four levels of measurement in 1946 (Stevens 1946), and they are still the standard way to sort attributes (Table 5.2).

Table 5.2: The four levels of measurement. Each level supports everything the levels above it support, and more.

Level	What it is	Examples	What you may ask
Nominal	Categories with no order. Two values are only ever the same or different.	Landuse, county name, land cover code	= and ≠. Counts and the most common value (the mode). No averages.
Ordinal	Ranked categories, but the distance between ranks is unknown.	MillRate (A, B, C, D), a hurricane’s category, “low, medium, high”	= ≠ < >. A median, not a mean.
Interval	Equal steps, but the zero point is arbitrary.	Temperature in °F or °C, calendar year	Differences. Sums and ratios mislead: 20 °C is not twice as hot as 10 °C.
Ratio	Equal steps and a true zero, so ratios hold.	Population, area, density, income, rainfall	Everything, including ratios and normalization, such as density = population ÷ area.

A few notes:

- A **nominal** attribute can be stored as numbers, like land cover codes 11 for water and 41 for forest, but the average land cover code of a county is meaningless.
- An **ordinal** attribute has an order, but not a size. A Category 4 hurricane is stronger than a Category 2, but it isn’t “twice as strong.”
- An **interval** attribute’s zero doesn’t mean “none.” Zero degrees Celsius is not the absence of temperature, and the year 0 is not the beginning of time, so ratios of interval values make no sense.
- A **ratio** attribute’s zero does mean “none”: zero people, zero acres. Only ratio data supports statements like “twice as many.”

The level of measurement also guides how you map an attribute: nominal data gets different colors or symbols for each category, while ordinal, interval, and ratio data get graduated colors or sizes. You will meet these ideas again when we make maps of census data.

Think about it

What level of measurement is each of these: a parcel's zoning code, a restaurant's health inspection grade (A, B, or C), median household income, the year a building was built, elevation in meters above sea level, and a student's class rank?

5.2 Querying a table

5.2.1 What is a query?

A **query** is a written condition, not a click. It asks the table a question, such as *which parcels are larger than 20 acres?*, and four ideas follow from that:

- The software tests **every record separately, one at a time**, against the condition.
- The answer is a **set of records**, and therefore a set of features on the map.
- The records that pass make up the **selected set**, or **selection**. They are highlighted in the table and on the map.
- The logical operators **AND**, **OR**, and **NOT** decide how several conditions combine.

5.2.2 SQL

SQL, Structured Query Language, is the standard language for asking a database a question. A full SQL statement reads:

```
SELECT * FROM parcels WHERE "Area" > 20
```

In words: *select all the columns (*) from the table named parcels, for the records where Area is greater than 20.*

In a GIS you rarely type the whole statement. The layer you are working on supplies the `FROM` part, and the selection tool supplies the `SELECT` part, so **you write only the WHERE clause**, the condition:

```
"Area" > 20
```

- In **QGIS**, use **Select Features by Expression** (on the attribute table's toolbar, or under Edit > Select). Type the condition into the expression box, or build it by double-clicking fields and operators in the lists.
- In **ArcGIS Pro**, use **Select By Attributes** (on the Map tab, or in the attribute table), which runs the **Select Layer By Attribute** tool. Build the clause with drop-down lists, or switch to SQL mode to type it.

A WHERE clause uses **comparison operators** (Table 5.3).

Table 5.3: Comparison operators.

Operator	Meaning	Example
=	equal to	`"Landuse" = 'Urban'`
<> or !=	not equal to	`"Municip" <> 'City'`

Operator	Meaning	Example
> and >=	greater than, greater than or equal to	`"Area" > 20`
< and <=	less than, less than or equal to	`"Density" <= 100`

Quoting is not decoration. It tells the software what kind of thing each part of the clause is:

- **Field names** take double quotes: `"Area"`, `"Landuse"`.
- **Text values** take single quotes: `'Urban'`.
- **Numbers** take no quotes: 20.

A number in quotes is text, and text sorts alphabetically, so `'9'` comes after `'10'`. A text value without quotes is read as a field name, which usually gives an error. Whether a text comparison cares about capital letters depends on the format: `'urban'` may or may not match `Urban`, so match the capitalization in the data.

5.2.3 Simple and compound queries

A **simple query** uses one attribute and one comparison. `"Area" > 20` tests each parcel's area and selects four of the seven: parcels 2, 4, 5, and 6 (Figure 5.3, top).

A **compound query** joins two or more conditions with a logical operator. With **AND**, a record must pass *both* tests. `"Landuse" = 'Urban' AND "Municip" = 'City'` selects only parcels 1 and 5, the only ones that are both urban and in a city (Figure 5.3, bottom).

AND narrows. Adding a condition with AND can never make a selection larger. Every extra condition is one more filter a record has to get through.

5.2.4 AND, OR, and NOT

There are three logical operators:

- **AND**: both conditions must be true for the same record. The result is where the two sets overlap.
- **OR**: either condition is enough. The result is everything in either set. **OR widens**: it can never make a selection smaller.
- **NOT**: reverses a condition, and selects every record the condition *doesn't* describe, its *complement*. It is written in front of the condition: `NOT ("Landuse" = 'Urban')`, which is the same as `"Landuse" <> 'Urban'`.

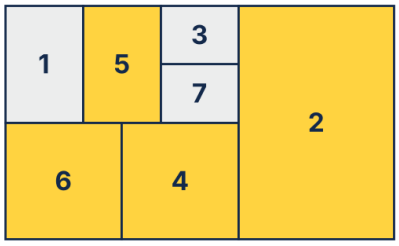
Figure 5.4 shows OR and NOT on the same seven parcels, and Table 5.4 compares all three.

Table 5.4: The three logical operators on the seven parcels.

Operator	Expression	Records selected	Count
AND	`"Landuse" = 'Urban' AND "Municip" = 'City'`	1, 5	2 of 7
OR	`"Area" > 20 OR "Municip" = 'City'`	1, 2, 4, 5, 6	5 of 7
NOT	`NOT ("Landuse" = 'Urban')`	2, 3, 4, 6	4 of 7

A simple query

```
"Area" > 20
```

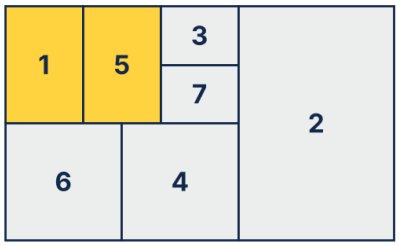


Selected: 4 of 7 (records 2, 4, 5, 6)

ID	Area	Landuse	Municip	Density	MillRate
1	10.5	Urban	City	1112.2	A
2	330.3	Farm	County	1.9	C
3	2.4	Suburban	Township	237.5	C
4	96	Suburban	County	98.1	A
5	22.1	Urban	City	916.2	B
6	30.2	Farm	Township	3.7	A
7	4.4	Urban	County	153.8	D

A compound query with AND

```
"Landuse" = 'Urban' AND "Municip" = 'City'
```



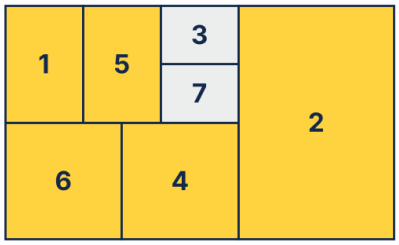
Selected: 2 of 7 (records 1, 5)

ID	Area	Landuse	Municip	Density	MillRate
1	10.5	Urban	City	1112.2	A
2	330.3	Farm	County	1.9	C
3	2.4	Suburban	Township	237.5	C
4	96	Suburban	County	98.1	A
5	22.1	Urban	City	916.2	B
6	30.2	Farm	Township	3.7	A
7	4.4	Urban	County	153.8	D

Figure 5.3: A simple query (top) selects parcels larger than 20 acres: four of seven. A compound query with AND (bottom) selects parcels that are both urban and in a city: two of seven.

OR: either condition is enough

```
"Area" > 20 OR "Municip" = 'City'
```

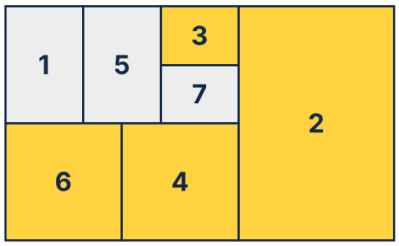


Selected: 5 of 7 (records 1, 2, 4, 5, 6)

ID	Area	Landuse	Municip	Density	MillRate
1	10.5	Urban	City	1112.2	A
2	330.3	Farm	County	1.9	C
3	2.4	Suburban	Township	237.5	C
4	96	Suburban	County	98.1	A
5	22.1	Urban	City	916.2	B
6	30.2	Farm	Township	3.7	A
7	4.4	Urban	County	153.8	D

NOT: the complement

```
NOT ("Landuse" = 'Urban')
```



Selected: 4 of 7 (records 2, 3, 4, 6)

ID	Area	Landuse	Municip	Density	MillRate
1	10.5	Urban	City	1112.2	A
2	330.3	Farm	County	1.9	C
3	2.4	Suburban	Township	237.5	C
4	96	Suburban	County	98.1	A
5	22.1	Urban	City	916.2	B
6	30.2	Farm	Township	3.7	A
7	4.4	Urban	County	153.8	D

Figure 5.4: OR (top) selects parcels that are larger than 20 acres or in a city, or both: five of seven. NOT (bottom) selects every parcel that isn't urban: four of seven.

Why does OR leave out parcels 3 and 7? Parcel 3 is 2.4 acres in a township, and parcel 7 is 4.4 acres in a county: each fails both tests.

5.2.5 Selecting rows also selects features

Selecting records in a table and selecting features on a map are the same act. That makes queries a powerful way to find *places*. Figure 5.5 shows a made-up county of forty townships. Each of the first three maps is one simple condition, and each selects a large, plausible-looking region. The fourth map ANDs all three together. Only the townships that pass all three tests remain, and there are far fewer of them than any one map suggests.

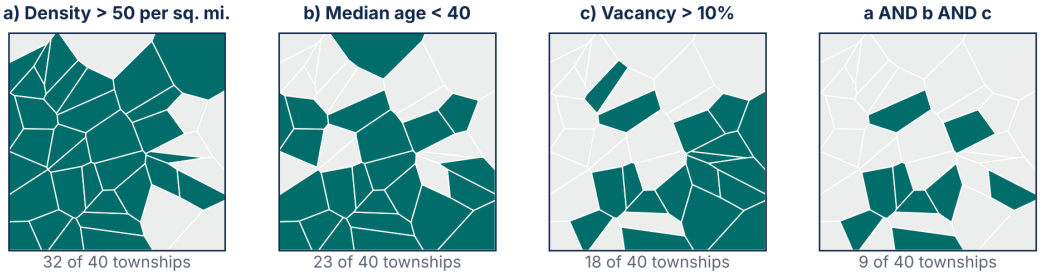


Figure 5.5: A made-up county of forty townships. Each of the first three maps selects townships that pass one condition. The fourth selects townships that pass all three: a AND b AND c.

Before you run a query like this, predict the result. If the answer is very different from what you expected, either the data or your query will teach you something.

5.2.6 Complex queries and brackets

A **complex query** mixes AND, OR, and NOT. Then the order in which the parts are evaluated matters, and **brackets** (parentheses) decide it. Each bracketed group is worked out on its own before the groups are joined (Figure 5.6):

```
("Landuse" = 'Urban' AND "MillRate" = 'B')
  OR (NOT "Municip" = 'City' AND "Density" > 200)
```

Read it in two groups. In the first, only parcel 5 is urban with a mill rate of B. In the second, only parcel 3 is outside a city with a density above 200 (it is a township with 237.5 people per square mile). The OR then joins the groups, so the query selects parcels 3 and 5.

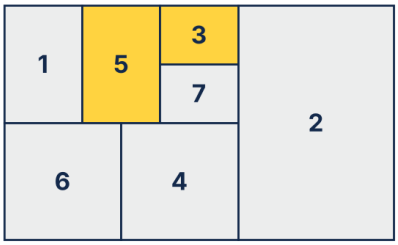
Without brackets, SQL evaluates NOT first, then AND, then OR. That happens to give the same answer here, but most of the time, removing brackets changes the answer. **Always write the brackets**, even when you think you don't need them. They make the query say what you mean, and they make it easy for the next person to read.

5.2.7 Where you put NOT matters

The position of NOT can change a query's answer completely (Figure 5.7).

A complex query with brackets

```
("Landuse" = 'Urban' AND "MillRate" = 'B')
  OR (NOT "Municip" = 'City' AND "Density" > 200)
```



Selected: 2 of 7 (records 3, 5)

ID	Area	Landuse	Municip	Density	MillRate
1	10.5	Urban	City	1112.2	A
2	330.3	Farm	County	1.9	C
3	2.4	Suburban	Township	237.5	C
4	96	Suburban	County	98.1	A
5	22.1	Urban	City	916.2	B
6	30.2	Farm	Township	3.7	A
7	4.4	Urban	County	153.8	D

Figure 5.6: A complex query with brackets selects parcels 3 and 5.

- `NOT ("Landuse" = 'Urban' AND "Municip" = 'County')` negates the whole bracket. Only parcel 7 is both urban and in a county, so negating that selects everything else: **six** records.
- `NOT "Landuse" = 'Urban' AND NOT "Municip" = 'County'` negates each condition, then joins them with AND. A parcel must be neither urban nor in a county. Only parcels 3 and 6 qualify: **two** records.

This is **De Morgan's law**, from logic: NOT (A AND B) is the same as (NOT A) OR (NOT B), *not* (NOT A) AND (NOT B). If you move a NOT inside a bracket, flip each AND to an OR, and each OR to an AND.

5.2.8 Beyond = and >

Four more operators make everyday queries shorter and clearer (Table 5.5).

Table 5.5: Four more operators.

Operator	Example	What it does
`LIKE`	`"NAME" LIKE 'North%'`	Matches a text pattern. % stands for any number of characters (including none), and _ for exactly one. This finds North Carolina, North Dakota, and Northampton.
`BETWEEN`	`"Pop2020" BETWEEN 5000 AND 9000`	Selects an inclusive range: **both** 5,000 and 9,000 are selected. It is a tidy way to write `"Pop2020" >= 5000 AND "Pop2020" <= 9000`.

Operator	Example	What it does
`IN`	`"Landuse" IN ('Urban', 'Suburban')`	Tests membership in a list. It is a tidy way to write a chain of ORs on the same field.
`IS NULL`	`"Pop2020" IS NULL`	The only correct way to test for a missing value (see Section 5.4). `IS NOT NULL` finds records that have a value.

Watch BETWEEN when you divide data into classes. If one class is `BETWEEN 0 AND 5000` and the next is `BETWEEN 5000 AND 10000`, a county with exactly 5,000 people lands in both. In QGIS, `ILIKE` works like LIKE but ignores capital letters.

5.2.9 What to do with a selection

A query only creates a selection. What you do next is what produces a result:

1. **Zoom to the selection** and look at it. This is the cheapest check there is. If the selected features are in implausible places, the expression is wrong, not the data. (QGIS: **Zoom Map to Selected Rows** in the attribute table. ArcGIS Pro: right-click the layer, **Selection > Zoom To Selection**.)
2. **Read the summary statistics** for the selected set: the count, sum, mean, and range. Read the **count** first. It tells you whether the query did what you meant. (QGIS: the attribute table's title bar shows how many features are selected, and the **Statistics** panel summarizes a field for selected features only. ArcGIS Pro: the bottom of the attribute table shows the count, and right-clicking a field and choosing **Statistics** makes a chart and summary.)
3. **Calculate a field** for the selection. The **field calculator** writes values into an attribute, for the selected records only. For example, set a new field, `Flag`, to 1 for the selected parcels. That turns a query into a permanent category you can map and query again. (QGIS: **Field Calculator**, with *Only update selected features* checked. ArcGIS Pro: **Calculate Field**.)
4. **Export the selection** to a new layer or table, so it survives. (QGIS: right-click the layer, **Export > Save Selected Features As**. ArcGIS Pro: right-click the layer, **Data > Export Features**; with a selection, only the selected features are exported.)

A selection is temporary. Clearing it erases it, and so does closing the project, or sometimes a careless click on the map. If you will need the result again, calculate it into a field or export it.

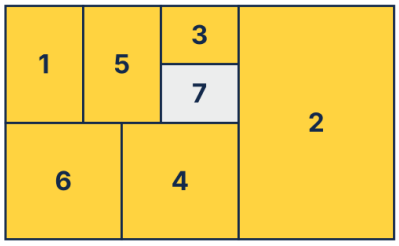
When you want a layer to show only some features, all the time, use a filter instead of a selection: in QGIS, **Filter** (in the layer's right-click menu), and in ArcGIS Pro, a **definition query** (in the layer's Properties). A filter uses the same WHERE clause, but it hides the records that fail it until you remove it.

Did you know..?

SQL was invented at IBM in the early 1970s by Donald Chamberlin and Raymond Boyce, to query the "relational" databases their colleague Edgar F. Codd had described in 1970 (Codd 1970). They first called it SEQUEL, for Structured English Query Language, but another company already had a trademark on that name, so it was shortened to SQL. That is why people still argue about whether to say "sequel" or "S-Q-L." (Both are

NOT the whole bracket

```
NOT ("Landuse" = 'Urban' AND "Municip" = 'County')
```

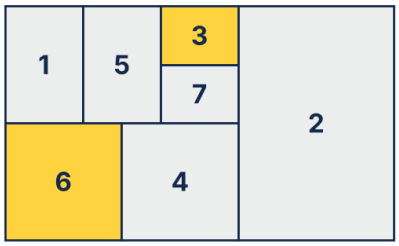


Selected: 6 of 7 (records 1, 2, 3, 4, 5, 6)

ID	Area	Landuse	Municip	Density	MillRate
1	10.5	Urban	City	1112.2	A
2	330.3	Farm	County	1.9	C
3	2.4	Suburban	Township	237.5	C
4	96	Suburban	County	98.1	A
5	22.1	Urban	City	916.2	B
6	30.2	Farm	Township	3.7	A
7	4.4	Urban	County	153.8	D

NOT each condition, joined by AND

```
NOT "Landuse" = 'Urban' AND NOT "Municip" = 'County'
```



Selected: 2 of 7 (records 3, 6)

ID	Area	Landuse	Municip	Density	MillRate
1	10.5	Urban	City	1112.2	A
2	330.3	Farm	County	1.9	C
3	2.4	Suburban	Township	237.5	C
4	96	Suburban	County	98.1	A
5	22.1	Urban	City	916.2	B
6	30.2	Farm	Township	3.7	A
7	4.4	Urban	County	153.8	D

Figure 5.7: Where the NOT sits changes the answer. Negating the whole bracket (top) selects six parcels; negating each condition and joining them with AND (bottom) selects two.

fine.) More than fifty years later, SQL is still one of the most widely used programming languages in the world, and every GIS program you will use speaks some form of it.

5.3 Thinking in sets

5.3.1 Every condition names a set

Here is a way to check a query before you trust it: think in **sets**. Each condition in a query selects a set of records.

- **AND** keeps only the records in both sets: the **intersection**.
- **OR** keeps the records in either set: the **union**.
- **NOT** keeps the records outside a set: the **complement**.

A **Venn diagram** draws each set as a circle, inside a rectangle that holds every record. Sketching one before you write a query catches most logic mistakes. Try it with students. Let:

- T = all students at the university (the whole table),
- A = students from New York,
- B = Geography majors, and
- C = English majors.

The query `"Home_State" = 'NY' AND "Major" = 'Geography'` selects A AND B: students who are in both sets, the overlap of the circles. Change one word, `"Home_State" = 'NY' OR "Major" = 'Geography'`, and it selects A OR B: everyone inside either circle, including the overlap (Figure 5.8).

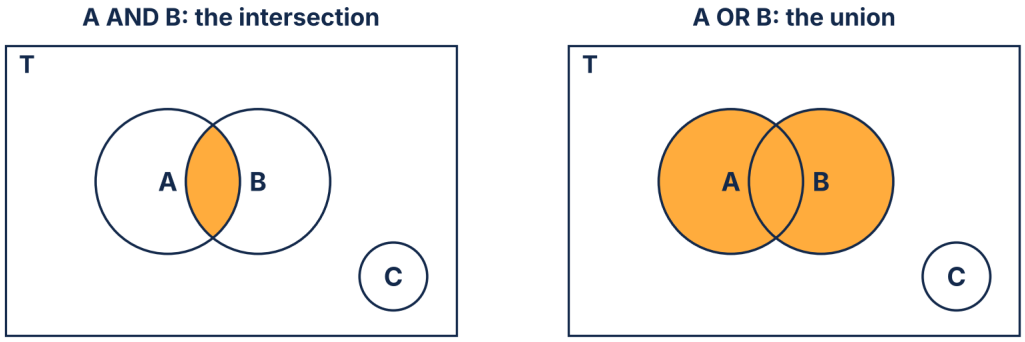


Figure 5.8: AND is the intersection of two sets (left); OR is their union (right). T is every student; A is students from New York; B is Geography majors; C is English majors.

AND can never select more records than either condition alone; OR can never select fewer. If an OR query returns almost the whole table, check whether you meant AND. If an AND query returns nothing, check whether the circles can overlap at all.

5.3.2 What do you get?

Try these four before you look at Figure 5.9. For each, which students are selected?

1. `"Major" = 'Geography' AND "Major" = 'English'`
2. `"Major" = 'Geography' OR "Major" = 'English'`
3. `"Home_State" = 'NY' AND "Major" = 'English'`
4. `"Home_State" = 'NY' OR "Major" = 'English'`

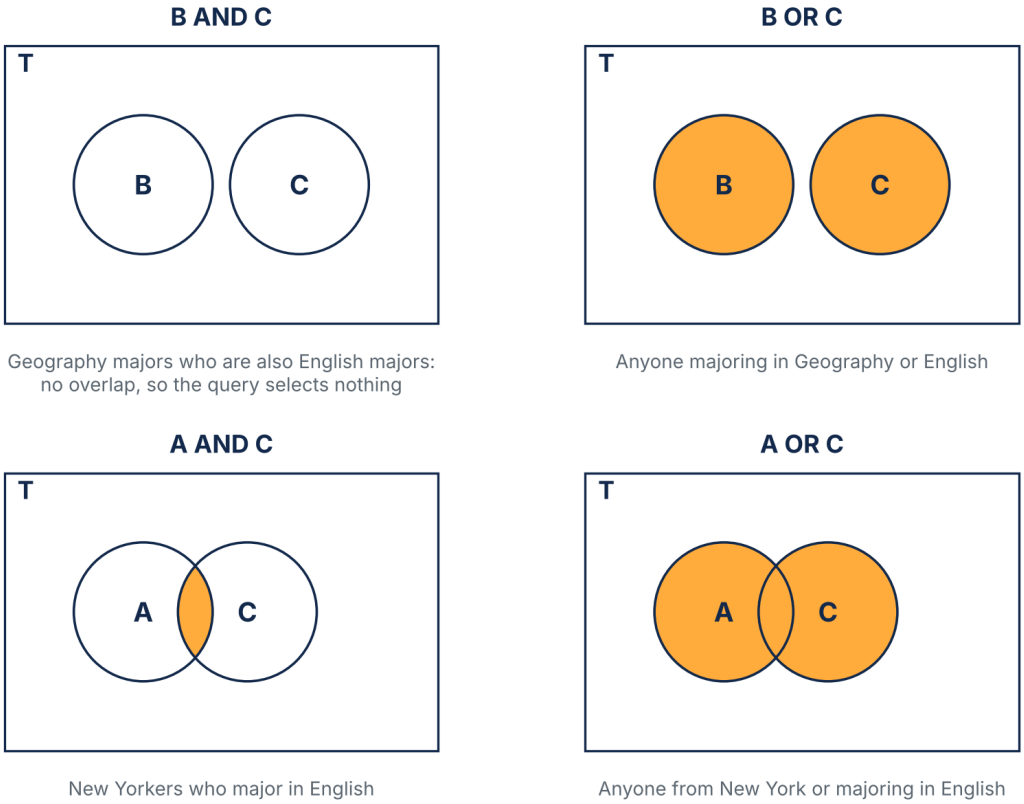


Figure 5.9: The four queries as Venn diagrams. B AND C selects nothing, because in this table a student has only one major, so the circles don't overlap.

The first is the surprise. Each record has only one value in the Major field, so no single student is both a Geography major and an English major *in this table*. The circles don't overlap, and the query selects nothing. **An AND on two different values of the same field always selects nothing.**

5.3.3 Each feature is tested separately

That leads to the two most common mistakes in writing queries, both caused by forgetting that **each record is tested on its own** (Table 5.6).

Table 5.6: Two classic mistakes.

Wrong	Why it fails	Right
`"Landuse" = 'Urban' AND "Landuse" = 'Farm'`	No single parcel is both urban and farm, so **nothing** is selected.	`"Landuse" = 'Urban' OR "Landuse" = 'Farm'`, or `"Landuse" IN ('Urban', 'Farm')`
`"Pop2020" >= 5000 OR "Pop2020" < 9000`	Every number is either at least 5,000 or below 9,000, so **everything** is selected.	`"Pop2020" >= 5000 AND "Pop2020" < 9000`

The rule of thumb: **two values of one attribute take OR; two bounds on one attribute take AND.**

The trouble comes from how we talk. "Show me the urban parcels and the farm parcels" uses *and* in English, but it means OR in SQL: you want a parcel if it is urban *or* if it is farm. Translate the question into "a parcel is selected if…" before you write the query.

5.4 Null values and missing data

One last trap. A **NULL** is not a value. It is the record telling you it doesn't know.

- **NULL is not zero, and not blank.** NULL means no value was recorded. Zero is a measurement, and an empty text string is a value. A Density of 0 says "nobody lives here"; a Density of NULL says "we never counted."
- **NULL is equal to nothing**, not even another NULL. So `"Density" = NULL` selects no records at all, even if half the table is NULL. Use `IS NULL` or `IS NOT NULL` instead.
- **NULL fails every comparison.** A record with a NULL Pop2020 is selected by neither `"Pop2020" >= 5000` nor `"Pop2020" < 5000`. Two queries that look like they split the table in two no longer add up to the whole table.
- **NOT doesn't rescue it.** `NOT ("Landuse" = 'Urban')` drops the NULL records along with the urban ones. If you want them, say so: `NOT ("Landuse" = 'Urban') OR "Landuse" IS NULL` (Figure 5.10).

Shapefiles make this worse: the old dBASE format can't store a true NULL in a numeric field, so missing numbers are often stored as 0, or as a code like –9999. Always read the metadata to find out how missing values are coded.

Build a habit: **before you trust any count, query `IS NULL` on each field you used**, and see how many records were never in the running.

Think about it

A table of 100 counties has 12 with a NULL median income. `"MedIncome" > 50000` selects 40 counties. How many does `"MedIncome" <= 50000` select? How many does `NOT ("MedIncome" > 50000)` select?

Show the answer

48 for both. The 12 counties with NULL income fail both comparisons, and NOT doesn't bring them back: 40 + 48 + 12 = 100.

ID	Area	Landuse	Municip
1	10.5	Urban	City
2	330.3	Farm	County
3	2.4	*NULL*	Township
4	96	Suburban	County
5	22.1	Urban	City
6	30.2	*NULL*	Township
7	4.4	Urban	County

```
NOT ("Landuse" = 'Urban')
```

selects 2 and 4 only. Parcels 3 and 6 are dropped: NULL fails every test.

```
... OR "Landuse" IS NULL
```

adds them back: 2, 3, 4 and 6.

Figure 5.10: Parcels 3 and 6 have no recorded land use. NOT (“Landuse” = ‘Urban’) selects only parcels 2 and 4; adding OR “Landuse” IS NULL brings back 3 and 6.

5.5 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

5.6 Summary

- An **attribute table** stores one **record** (row) per feature and one **attribute** (field, column) per variable. Selecting a record selects its feature on the map.
- A field’s **data type** limits what it can store. Integer types are only as wide as the **bits** behind them: a 32-bit integer tops out at 2,147,483,647. Store identifiers like ZIP codes and FIPS codes as **text**.
- The **level of measurement** limits what you may ask: **nominal** (categories), **ordinal** (ranks), **interval** (equal steps, arbitrary zero), and **ratio** (a true zero).
- A **query** is a WHERE clause the software tests against every record, one at a time. In **SQL**, field names take double quotes, text values single quotes, and numbers none.
- **AND** narrows (the **intersection**), **OR** widens (the **union**), and **NOT** takes the **complement**. Brackets and the placement of NOT change the answer, so write them out.
- **LIKE**, **BETWEEN**, **IN**, and **IS NULL** shorten everyday queries.
- Draw a **Venn diagram** to check a query. Two values of one attribute take OR; two bounds on one attribute take AND.
- **NULL** means no value was recorded. It fails every comparison, so run IS NULL before you trust a count.
- A **selection** is temporary. Zoom to it, read its count, then calculate a field or export it to keep the result.

5.7 Exercises

1. **Explore a table.** Download Natural Earth's populated places layer (https://www.naturalearthdata.com) and open its attribute table. How many records does it have? Sort it by `pop_max`. What are the ten largest cities? Find a field with NULL values.
2. **Data types.** For each of these fields, choose a data type and explain why: a county FIPS code, the number of students in a school, a stream's length in kilometers, whether a road is paved, a building's construction date, and a phone number.
3. **Levels of measurement.** For each field in exercise 2, name its level of measurement. Which could you average?
4. **Write the query.** Using the seven parcels, write a WHERE clause for each, and list the parcels it selects: (a) farm parcels larger than 50 acres; (b) parcels in a township or a county; (c) urban parcels with a density over 1,000 or a mill rate of A; (d) every parcel that is not suburban.
5. **Predict, then run.** In QGIS or ArcGIS Pro, with the populated places layer, write one query with AND, one with OR, and one with IS NULL. Predict the count before you run each one. How close were you?
6. **Find the mistake.** Explain what's wrong with each query, and fix it: (a) `"STATE" = 'NC' AND "STATE" = 'SC'`; (b) `"POP" > 1000 OR "POP" < 5000`; (c) `"NAME" = Durham`; (d) `"INCOME" = NULL`.
7. **De Morgan.** Rewrite `NOT ("Landuse" = 'Urban' OR "Municip" = 'City')` without a bracket after the NOT. Which parcels does it select?
8. **Keep the answer.** Select the North Carolina counties with more than 100,000 people. Calculate a field, `Big`, set to 1 for those counties, and export them to a new layer. Then clear the selection. Which of the two results survived?

5.8 Further reading and online resources

5.8.1 Books and open textbooks

- *GIS Fundamentals: A First Text on Geographic Information Systems*, by Paul Bolstad (Bolstad 2022). Chapter 8, "Attribute Data and Tables," covers attribute tables, SQL queries, and Boolean logic.
- *Understanding GIS through Sustainable Development Goals*, by Paul Holloway (Holloway 2023). Includes worked attribute queries in QGIS and ArcGIS Pro.
- *Essentials of Geographic Information Systems*, by Jonathan Campbell and Michael Shin. Chapter 5, "Geospatial Data Management," covers attribute tables, data types, and queries: https://open.umn.edu/opentextbooks/textbooks/67.
- *The Nature of Geographic Information*, by David DiBiase and colleagues at Penn State. Chapter 3, "Census Data and Thematic Maps," discusses levels of measurement: https://www.e-education.psu.edu/natureofgeoinfo/c3.html.
- *Intro to GIS and Spatial Analysis*, by Manuel Gimond. The chapter "GIS Data Formats" covers attribute tables and data types: https://mgimond.github.io/Spatial/.
- *Getting Started with SQL*, by Thomas Nield (O'Reilly, 2016), a short, friendly introduction to SQL.
- *PostGIS in Action*, by Regina Obe and Leo Hsu (3rd ed., Manning, 2021), for when you are ready to query spatial databases directly.

5.8.2 Attribute data, data types, and measurement

- QGIS User Guide, "Working with the Attribute Table": https://docs.qgis.org/latest/en/docs/user_manual/working_with_vector/attribute_table.html.
- QGIS, *A Gentle Introduction to GIS*, "Vector Attribute Data": https://docs.qgis.org/latest/en/docs/gentle_gis_introduction/vector_attribute_data.html.
- ArcGIS Pro, "ArcGIS field data types": https://pro.arcgis.com/en/pro-app/latest/help/data/geodatabases/overview/arcgis-field-data-types.htm.
- ArcGIS Pro, "Add Field" tool, including precision and scale: https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/add-field.htm.
- Stevens, S. S., "On the Theory of Scales of Measurement," *Science* (Stevens 1946), the original paper on levels of measurement: https://doi.org/10.1126/science.103.2684.677.
- The GIS&T Body of Knowledge, "Scales of Measurement," "Database Management Systems," and "Structured Query Language (SQL) and Attribute Queries": https://www.ucgis.org/site/gis-t-body-of-knowledge.

5.8.3 Queries and SQL

- QGIS User Guide, "Expressions," the full list of operators and functions you can use in Select by Expression: https://docs.qgis.org/latest/en/docs/user_manual/expressions/expression.html.
- QGIS User Guide, "Selecting features," including Select by Expression: https://docs.qgis.org/latest/en/docs/user_manual/introduction/general_tools.html#selecting-features.
- ArcGIS Pro, "Select Layer By Attribute" tool: https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/select-layer-by-attribute.htm.
- ArcGIS Pro, "SQL reference for query expressions used in ArcGIS": https://pro.arcgis.com/en/pro-app/latest/help/mapping/navigation/sql-reference-for-elements-used-in-query-expressions.htm.
- ArcGIS Pro, "Write a query in the Query Builder": https://pro.arcgis.com/en/pro-app/latest/help/mapping/navigation/write-a-query-in-the-query-builder.htm.
- ArcGIS Pro, "Use definition queries": https://pro.arcgis.com/en/pro-app/latest/help/mapping/layer-properties/definition-query.htm.
- SQLBolt, free interactive SQL lessons in your browser: https://sqlbolt.com.
- W3Schools SQL Tutorial, with a try-it-yourself editor for every operator in this chapter: https://www.w3schools.com/sql/.
- Khan Academy, "Intro to SQL: Querying and managing data": https://www.khanacademy.org/computing/computer-programming/sql.
- SQLite Tutorial, "SQLite NULL," on how NULL behaves: https://www.sqlitetutorial.net/sqlite-is-null/.
- Codd, E. F., "A Relational Model of Data for Large Shared Data Banks" (Codd 1970), the paper that started relational databases: https://doi.org/10.1145/362384.362685.

5.8.4 Sets and logic

- Math is Fun, "Sets and Venn Diagrams": https://www.mathsisfun.com/sets/venn-diagrams.html.
- Math is Fun, "Introduction to Sets" and set operations: https://www.mathsisfun.com/sets/sets-introduction.html.
- Wikipedia, "De Morgan's laws": https://en.wikipedia.org/wiki/De_Morgan%27s_laws.

- Stanford Encyclopedia of Philosophy, “Propositional Logic,” for a deeper look at AND, OR, and NOT: https://plato.stanford.edu/entries/logic-propositional/.

5.8.5 Tutorials and data for practice

- Learn ArcGIS, lessons on querying and selecting data, such as “Get started with ArcGIS Pro”: https://learn.arcgis.com.
- Natural Earth, free world data with rich attribute tables to practice on: https://www.naturalearthdata.com.
- NC OneMap, North Carolina’s data portal: https://www.nconemap.gov.
- U.S. Census Bureau, TIGER/Line shapefiles, whose attribute tables are a good place to practice queries on FIPS codes and names: https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html.

6 Databases and Table Joins

Learning objectives

By the end of this chapter you should be able to:

1. Explain what a database management system (DBMS) does, and why GIS data is often stored in one.
2. Describe a relational database, and tell a primary key from a foreign key.
3. Join two tables on a shared key field, and say which table is the destination and which is the source.
4. Predict which records an inner, left, right, or full outer join keeps.
5. Name the cardinality of a join (one-to-one, many-to-one, one-to-many, or many-to-many), and decide whether a simple join will work.
6. Join a census table to census tract boundaries in QGIS, and fix the problems that make joins fail.

Chapter 5 asked questions of a single attribute table. Real projects rarely stop at one table. The census tract boundaries you download from the Census Bureau carry a code and a name, but no population. The population sits in a separate table, from a separate website, with no geometry at all. A county's parcels live in one table, and its owners in another. A forest's trails are in one table, and the forest's size and ranger office in a second.

To map any of these, you have to bring the two tables together. That operation is a **table join**, and it is one of the most common and most error-prone steps in GIS work. This chapter starts with how databases store and link tables, then shows how joins work, how they fail, and how to check them.

6.1 Databases and keys

6.1.1 From one table to many

An attribute table follows the rules from Chapter 5: one row per feature, and one column per attribute. A layer of Washington counties might have a FIPS code, a name, the 1990 population, the land area, and the population density, all in one table.

That works until the questions get bigger. Suppose you also track every county's hospitals, every hospital's doctors, and every doctor's specialties. You could keep adding columns to the county table, but you would soon need a column for "hospital 1," "hospital 2," and so on, with most of them empty. A better design splits the data into several tables, each about one kind of thing, and links them through shared values. Storing and linking many tables is the job of a database.

6.1.2 What is a database management system?

A **database management system (DBMS)** is software that stores data once and lets many programs and many people query it. Common examples include:

- **Oracle** and **Microsoft SQL Server**, large commercial systems used by governments and companies.
- **MySQL** and **PostgreSQL**, free, open-source servers. **PostGIS** is an extension that adds geometry types and spatial functions to PostgreSQL.
- **SQLite**, a small database that lives in a single file. **SpatiaLite** adds spatial functions to it, and a **GeoPackage** (Chapter 2) is a SQLite database laid out to a published standard. QGIS saves new layers as GeoPackages by default.

Without a DBMS, every program reads its own files directly (Figure 6.1, left). The mapping app reads one copy of the parcels, the permit office keeps another in a spreadsheet, and a reporting script reads last year's export. The copies drift apart, and no one controls who edits what.

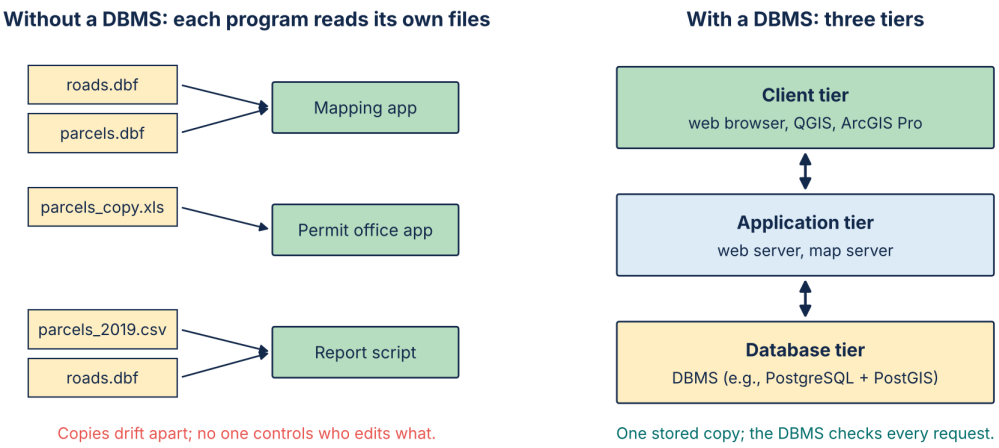


Figure 6.1: Left: without a DBMS, each program reads its own copies of the files. Right: with a DBMS, clients talk to an application tier, which talks to one stored copy of the data in the database tier.

A DBMS changes four things:

- **Client/server.** Client programs such as QGIS, ArcGIS Pro, or a web browser send requests to a database server, which does the work and sends back the answer.
- **Multiple tiers.** Large systems are usually built in three tiers (Figure 6.1, right): the client, an application tier (for example a web server or map server), and the database tier.
- **No direct write access.** Users change data only through the DBMS, which checks their permissions first. A summer intern can be allowed to read the parcel table but not to edit it.
- **Concurrency.** Many users can read and edit at the same time. The DBMS makes sure that two people editing the same record don't overwrite each other's work.

Did you know..?

SQLite is probably the most widely used database engine in the world. It is built into every Android and iPhone, every major web browser, and many desktop programs, so a typical smartphone holds hundreds of SQLite databases. Every GeoPackage you save from QGIS is one more (SQLite Project 2024).

6.1.3 Relational databases

Most databases are **relational databases**: they store data in tables, and they link the tables through fields that hold the same values. The idea goes back to the paper by E. F. Codd that we met in Chapter 5 (Codd 1970).

Figure 6.2 shows a small example from North Carolina's national forests. One table lists the four forests: each one's name, ranger office, rough size, and an ID number. A second table lists trails, and each trail records the ID of the forest it is in. The two tables share no names, but they share the `Forest_ID` field, and that is enough to link every trail to its forest.

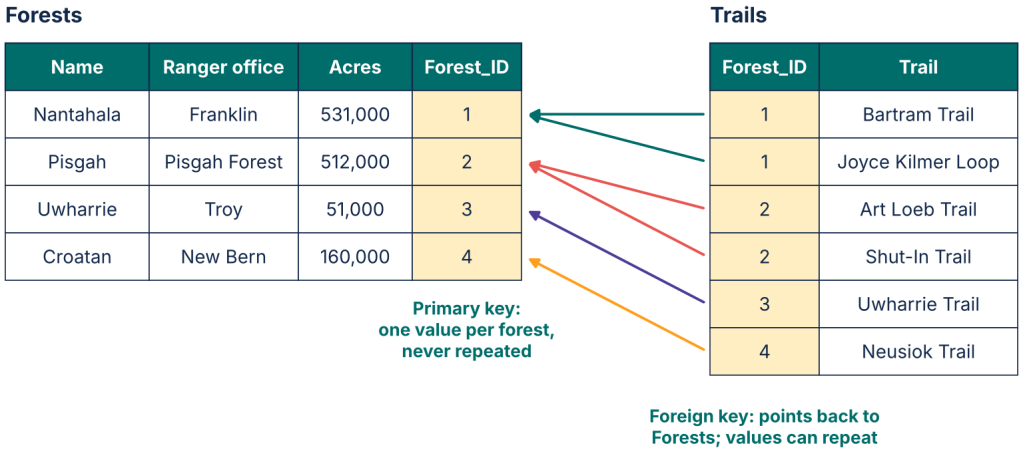


Figure 6.2: Two tables in a relational database. Each trail's Forest_ID points to one row in the Forests table. Acreages are rounded.

6.1.4 Primary and foreign keys

A field that links tables is called a **key**. There are two kinds:

- A **primary key** is a field whose values identify each row of a table uniquely. No value may repeat, and none may be missing. `Forest_ID` is the primary key of the Forests table: each forest has one ID, and no two forests share one.
- A **foreign key** is a field in another table that holds values of that primary key. `Forest_ID` in the Trails table is a foreign key. Its values can repeat, because a forest can have many trails.

The names of the two fields don't have to match. A Trails table could call the field `FOREST` and the Forests table could call it `Forest_ID`. What matters is that the values match, and that the two fields have the same data type.

Good keys are codes, not names. Names are spelled in different ways ("Pisgah," "Pisgah NF," "Pisgah National Forest"), and different places can share a name: about 30 states have a Washington County. In U.S. data, the most common keys are **FIPS codes**, the numeric codes the federal government gives each state, county, and census area. The Census Bureau's **GEOID** strings these codes together, so that each place has its own code (Section 6.4).

> **Think about it**
>
> Your university gives you a student ID number (a PID at UNC). Why does the registrar's database use that number as the key, rather than your name or your email address? What could go wrong with each of the other two?

6.2 Joining tables

6.2.1 What a join does

A **table join** appends the fields of one table to another, by matching the values in a key field. The two tables have names that tell you their role:

- The **destination table** (ArcGIS calls it the target, or input, table) is the one that receives the new fields. Its rows are all kept, in their own order. In GIS, the destination is usually the layer with the geometry, because you want to map the result.
- The **source table** (the join table) supplies the fields. Each of its values is copied to the destination row whose key matches.

It helps to picture the destination on the left and the source on the right. Figure 6.3 shows the trails joined to the forests: every trail now carries its forest's name, ranger office, and acreage. Nantahala's values appear twice, because two trails point to it.

Destination: Trails (kept as is)		Appended from the source: Forests			
Forest_ID	Trail	Name	Ranger office	Acres	Forest_ID
1	Bartram Trail	Nantahala	Franklin	531,000	1
1	Joyce Kilmer Loop	Nantahala	Franklin	531,000	1
2	Art Loeb Trail	Pisgah	Pisgah Forest	512,000	2
2	Shut-In Trail	Pisgah	Pisgah Forest	512,000	2
3	Uwharrie Trail	Uwharrie	Troy	51,000	3
4	Neusiok Trail	Croatan	New Bern	160,000	4

Figure 6.3: The Trails table (destination) after a join with the Forests table (source). The purple fields came from Forests.

Three things about joins are easy to forget:

1. **A join matches values, not positions.** The two tables can be sorted in any order. The software looks up each destination key in the source.
2. **A join is temporary.** In QGIS and ArcGIS Pro, a join is a live link between two tables. Nothing is written to the files on disk. If you remove the join, or open the layer in a new project, the fields are gone. To keep them, export the joined layer to a new file.
3. **A join changes neither table.** The source table is not altered, and the destination only borrows the source's fields while the join is in place.

6.2.2 Tables are bags, not sets

In Chapter 5 we treated the records a query selects as a **set**: a collection of distinct things, with no duplicates. A database table is not always a set. Unless a primary key forbids it, a table can hold two identical rows. A collection that can contain duplicates is called a **bag** (or a multiset).

That matters for joins. The table below, TAKES, records which students take which courses. Zhang appears twice, because Zhang takes two courses. When the TAKES table is joined to a table of students, Zhang's row has two matches. A join does not choose between them: it

either repeats Zhang's row once for each match, or it keeps only the first one. We return to this problem under cardinality (Section 6.3).

6.2.3 Which rows does a join keep?

Here are two small tables from a student records system. STUDENT has one row per student. TAKES has one row per enrollment. One enrollment, for ID 33301, belongs to a student who has since left and is no longer in STUDENT.

Table 6.1: STUDENT

Id	Name
00128	Zhang
12345	Shankar
11122	Parr
22211	McKeon

Table 6.2: TAKES

Id	Course_id	Sec_id
00128	CS-347	1
00128	CS-101	2
12345	HIS-222	1
33301	GEOG-370	1

Parr and McKeon have no enrollments. The student with ID 33301 has an enrollment but no row in STUDENT. A join has to decide what to do with records like these, which have no partner in the other table. There are four kinds of join, and they differ only in which unmatched records they keep (Figure 6.4):

- An **inner join** keeps only rows that match in both tables. Here that is Zhang's two enrollments and Shankar's one.
- A **left join** (or left outer join) keeps every row of the left table, the destination, and fills the source fields with NULL where there is no match. Here that adds Parr and McKeon, with NULL courses.
- A **right join** keeps every row of the right table, the source. Here that is the three matched enrollments plus 33301's enrollment, with a NULL name.
- A **full outer join** keeps every row of both tables.

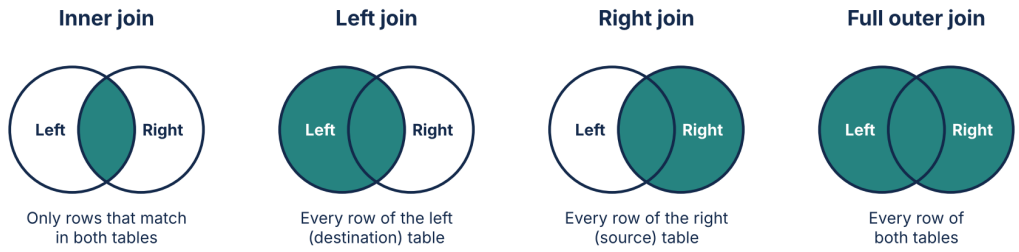


Figure 6.4: Four kinds of join drawn as Venn diagrams. The shaded part of each diagram is what the join keeps.

The left join of STUDENT to TAKES gives Table 6.3. Every student is there, once for each enrollment, and the two students with no enrollments are kept with NULL values.

Table 6.3: The left join of STUDENT (left) and TAKES (right).

Id	Name	Course_id	Sec_id
00128	Zhang	CS-347	1
00128	Zhang	CS-101	2
12345	Shankar	HIS-222	1

Id	Name	Course_id	Sec_id
11122	Parr	NULL	NULL
22211	McKeon	NULL	NULL

The left join is the default in GIS. When you join a table to a layer in QGIS or ArcGIS Pro, every feature of the layer is kept, matched or not. That is what you usually want: a county with no data should still appear on the map, so that you can see the gap. To get an inner join instead:

- In **ArcGIS Pro**, uncheck *Keep All Target Features* in the Add Join tool.
- In **QGIS**, use the *Join Attributes by Field Value* tool from the Processing Toolbox, and check *Discard records which could not be joined.*

Right and full outer joins are rare in desktop GIS, because the extra rows they keep have no geometry to draw. You will meet them in SQL databases.

6.2.4 A second example: people and cars

The key fields don't need the same name. In the tables below, a People table is the destination and a Cars table is the source. They are linked by matching `Name` in People to `Owner` in Cars.

Table 6.4: People (destination)

Name	Address	Kids
Dave	10 Main St	0
Lola	1 Walnut Av	3
Todd	5 LBJ	1
Melanie	7 Sessom St	9

Table 6.5: Cars (source)

Car	Year	Owner
Hummer	1999	Dave
Fiat	2012	Dave
Toyota	2002	Todd
Miata	2009	Melanie
VW Bus	1968	Melanie

Run a left join, and the result has six rows. Dave and Melanie each own two cars, so each appears twice. Lola owns no car, so her row is kept with NULL in the Car and Year fields. Notice that the result has *more* rows than the People table: the "one row per person" rule no longer holds. In a GIS, where each row is a feature, that is a problem, and it is the subject of the next section.

6.3 Cardinality

6.3.1 Four kinds of relationship

Cardinality describes how many records in one table match each record in the other. Name the destination first, and there are four possibilities (Figure 6.5):

- **One-to-one (1:1).** Each destination record matches exactly one source record. *States to governors*: each state has one governor, and each governor runs one state.
- **Many-to-one (M:1).** Many destination records match the same source record, but each destination record matches only one. *Counties to states*: many counties share a state, but each county is in one state.

- **One-to-many (1:M).** One destination record matches many source records. *States to counties*: each state has many counties.
- **Many-to-many (M:M).** Records on both sides match many records on the other. *Students to classes*: each student takes many classes, and each class has many students.

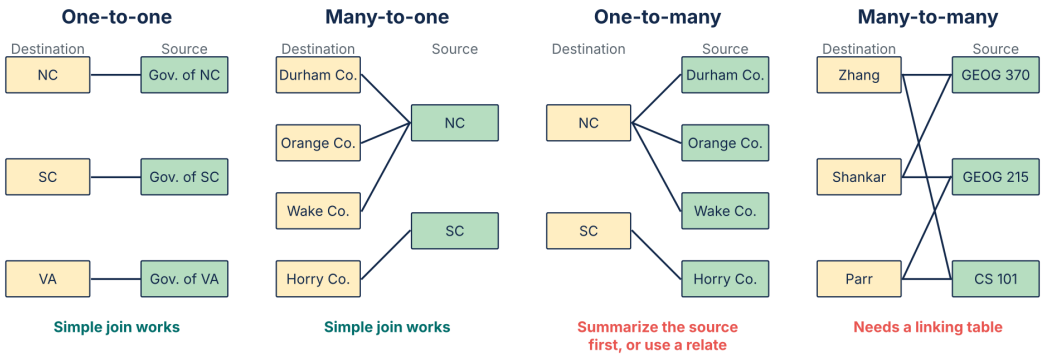


Figure 6.5: The four kinds of cardinality, with the destination table on the left of each panel and the source on the right.

The order matters. Counties to states is many-to-one; states to counties is one-to-many. The same two tables give a different cardinality depending on which one is the destination.

6.3.2 The rule of joining

One-to-one and many-to-one joins work directly. Every destination record finds at most one match, so there is exactly one value to copy into each field. In the Trails-to-Forests join (Figure 6.3), each trail is in one forest, so each trail gets one set of forest attributes. That many forest values are repeated doesn't matter.

One-to-many and many-to-many joins break the rule of a GIS layer, that each record is one feature. Suppose the destination is a states layer and the source is a table of counties (Figure 6.6). North Carolina matches three counties here (and 100 in reality). Which county's population should go in North Carolina's row?

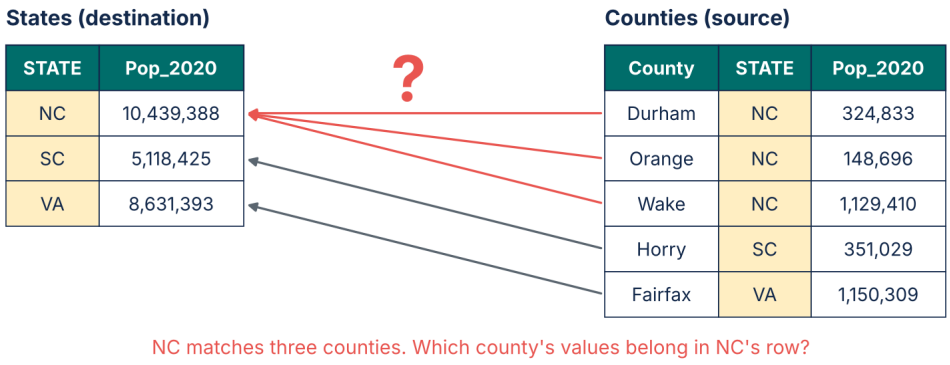


Figure 6.6: A one-to-many join. The North Carolina record matches three county records, so there is no single value to copy.

Software answers that question in different ways, and none of them is what you want:

- **QGIS** copies the first matching record it finds and ignores the rest. You get one county's value in each state, with no warning.

- **ArcGIS Pro**'s Add Join tool also keeps only the first match by default (its *Join Operation* is set to one-to-first), so it too drops records without a warning.
- **A SQL database** repeats the state's row once for each county, so North Carolina appears 100 times. On a map, that means 100 copies of the North Carolina polygon stacked on top of each other.

There are three ways to handle a one-to-many relationship properly:

1. **Summarize the source first.** If you want the total county population in each state, use a summary tool (*Aggregate* or *Statistics by categories* in QGIS, *Summary Statistics* in ArcGIS Pro) to make a table with one row per state. Then join that table, which is now one-to-one.
2. **Swap the tables.** Make the counties the destination and the states the source. Counties to states is many-to-one, so each county gets its state's attributes.
3. **Use a relate.** A **relate** (ArcGIS Pro) or a **relation** (QGIS) links the two tables without copying fields. Select North Carolina, and you can open all its counties in the related table. A relate never breaks the one-row-per-feature rule, because it never adds fields to the layer.

Many-to-many relationships need a third table. A **linking table** (a junction table) has one row for each pair: TAKES is a linking table between STUDENT and a COURSES table. Each of the two original relationships, students to enrollments and courses to enrollments, is then one-to-many.

Think about it

You have a layer of North Carolina's 100 counties and a table of every hospital in the state, with the county each hospital is in. What is the cardinality of a join from counties to hospitals? How would you make a map of the number of hospital beds in each county?

6.4 Joining census data in QGIS

6.4.1 The task

The most common join in American GIS work is attaching census statistics to census boundaries. The Census Bureau publishes the two separately. Boundary files, called TIGER/Line shapefiles, have geometry and identifiers but almost no statistics. Statistics come as tables, downloaded from data.census.gov, with identifiers but no geometry. A join is the bridge (Figure 6.7). Chapter 8 covers the census itself; here we focus on the join.

6.4.2 The key: GEOID

Each census tract has an 11-character **GEOID** made of three FIPS codes strung together:

Table 6.6: The parts of an 11-character tract GEOID, 37135010701.

Part	Digits	Example	Meaning
State	2	37	North Carolina
County	3	135	Orange County

Part	Digits	Example	Meaning
Tract	6	`010701`	Tract 107.01 (the decimal is implied)

The GEOID is a code, not a quantity, so it must be stored as **text** (Chapter 5). Stored as text, California's tract `06001400100` keeps its leading zero. Stored as a number, it becomes `6001400100`, and it no longer matches anything.

6.4.3 Step by step

The steps below join 2020 tract populations to North Carolina's census tracts. The same steps work for any state, and for counties or block groups.

1. **Load the destination layer.** Download the tract shapefile for your state from the Census Bureau's TIGER/Line page, and add it to QGIS. Open its attribute table, and find the `GEOID` field. Check that it is a text field (*Layer Properties > Fields* shows the type).
2. **Get the source table.** On data.census.gov, find table P1 (Race) from the 2020 Decennial Census for all tracts in the state, and download it as a CSV. The `GEO_ID` column looks like `1400000US37135010701`: the GEOID with a prefix in front. The total population is in the column `P1_001N`. (Older downloads, from before 2020, call the GEOID field `GEO.id2`.) Clean the table first if needed: delete the second header row, and make a column holding just the last 11 characters of `GEO_ID`.
3. **Add the CSV as a table.** Use *Layer > Add Layer > Add Delimited Text Layer*. Set the file format to CSV, check *First record has field names*, and under *Geometry definition* choose *No geometry (attribute only table)*. Under *Record and fields options*, uncheck *Detect field types*, so that the GEOID stays text. Click *Add*.
4. **Join.** Open the tract layer's *Properties* and go to the *Joins* tab. Click the green plus. Set *Join layer* to the CSV, *Join field* to its GEOID column, and *Target field* to `GEOID`. You can choose which fields to bring across, and set a short prefix for their names. Click *OK*.
5. **Check the join.** Reopen the tract layer's attribute table. The joined fields appear at the right end, named with the CSV's name or your prefix, such as `pop_P1_001N`. Sort by the joined field and look for NULL values. Every NULL is a tract that found no match. Never assume a join worked: always look.
6. **Make it permanent.** Export the joined layer (*right-click > Export > Save Features As*) to a new GeoPackage. The joined fields are now part of the new file.
7. **Map it.** Style the tracts with *Graduated* symbology on the population field. Chapter 11 covers how to choose the classes.

In ArcGIS Pro the steps are the same, except that you add the table with *Add Data* and join it with *Add Join* (right-click the layer, then *Joins and Relates*).

Did you know..?

Spreadsheets cause so many join failures that scientists once renamed genes to avoid them. Excel turned gene symbols such as SEPT1 and MARCH1 into the dates 1-Sep and 1-Mar, and a 2016 study found the error in about a fifth of the genetics papers that came with Excel files (Ziemann et al. 2016). In 2020 the committee that names human genes gave 27 genes new symbols, such as SEPTIN1 and MARCHF1, partly to stop it. Opening a census CSV in Excel does the same thing to GEOIDs: it drops the leading zeros.

Tract layer (destination)

GEOID	NAMELSAD	geometry
37135010701	Tract 107.01	polygon
37135010702	Tract 107.02	polygon
37135011200	Tract 112	polygon

+

Population CSV (source)

GEOID	POP
37135010701	4,812
37135011200	3,205
37135010702	5,147

→

Joined layer

GEOID	NAMELSAD	geometry	POP
37135010701	Tract 107.01	polygon	4,812
37135010702	Tract 107.02	polygon	5,147
37135011200	Tract 112	polygon	3,205

Match on GEOID: both fields are text, 11 characters, leading zeros kept.
The rows can be in any order; the join matches on values, not on position.

Figure 6.7: A census join: the tract layer (destination) and the population table (source) share the GEOID key, and the join adds the POP field to each tract. The population values are made up for the example.

6.4.4 Why joins fail

When a join gives NULL for every record, or for some of them, the cause is almost always one of the four in Table 6.7.

Table 6.7: Common join problems and their fixes.

Problem	What you see	Fix
The key fields have different types (text and number)	The join fails, or every value is NULL	Convert one field so that both keys have the same type
Leading zeros were lost (`06001` became `6001`)	Many NULLs, often a whole state	Keep IDs as text. Don't open the CSV in Excel; set the field types with a `.csvt` file, or turn off *Detect field types*
Extra spaces or different capitalization (`'Wake '` vs. `'WAKE'`)	Some rows fail to match	Trim spaces and make the case the same before the join
Duplicate keys in the source	Wrong or missing values	The join is one-to-many: summarize the source first

Figure 6.8 shows the second problem, the most common of all. The tract layer stores GEOID as text, with the leading zero. The CSV was read as numbers, which have no leading zeros. Not a single key matches.

Tracts

GEOID (text)
06001400100
06001400200
06001400300

CSV opened as numbers

GEOID (number)
6001400100
6001400200
6001400300

Join result

POP
NULL
NULL
NULL

The leading zero of California's state code (06) was dropped, so no key matches.

6.4.5 Attribute joins and spatial joins

Everything in this chapter is an **attribute join**: the tables are matched by the values of a shared key. Sometimes two layers share no key at all. A table of EMS calls has coordinates but no neighborhood code; a layer of neighborhoods has no call numbers. They can still be linked, by location, with a **spatial join**. That is the subject of Chapter 7.

Table 6.8: Attribute joins and spatial joins compared.

	Attribute join	Spatial join
Matches records by	A shared key field	Location
Example	Census CSV to tracts by GEOID	Count the schools in each district
Needs	A common field of the same type	Both layers in a suitable coordinate system
In QGIS	*Layer Properties > Joins*	*Join Attributes by Location*
In ArcGIS Pro	*Add Join*	*Spatial Join*

6.5 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

6.6 Summary

- A **database management system (DBMS)** stores data once and controls who can read and change it. It works as a client/server system, often in three tiers, and lets many users work at the same time.
- A **relational database** links tables through **keys**. A **primary key** identifies each row uniquely; a **foreign key** in another table points back to it and can repeat.
- A **table join** appends the fields of a **source table** to a **destination table** by matching key values. The key fields can have different names, but their values and data types must match.
- A join is temporary, matches on values rather than row order, and changes neither table. Export the result to keep it.
- An **inner join** keeps only matched rows. A **left join** keeps every destination row, with NULL where there is no match; it is the default in QGIS and ArcGIS Pro.
- **Cardinality** is how many source records match each destination record. **One-to-one** and **many-to-one** joins work directly. For **one-to-many**, summarize the source first, swap the tables, or use a **relate**. **Many-to-many** needs a linking table.
- Census statistics join to census boundaries on the **GEOID**, which must stay text so that its leading zeros survive.
- Most failed joins come from mismatched types, lost leading zeros, stray spaces, or duplicate keys. Always check the result for NULLs.

6.7 Exercises

1. **Design the keys.** A city tracks parks, the playgrounds in each park, and the inspections of each playground. Sketch three tables. Give each a primary key, and show the foreign keys that link them.
2. **Predict the join.** Using the People and Cars tables, write out the result of an inner join, a left join, and a right join. How many rows does each have?
3. **Name the cardinality.** For each pair, name the destination and source and give the cardinality: (a) counties to a table of county unemployment rates; (b) ZIP codes to a table of businesses with their ZIP code; (c) census tracts to the counties they are in; (d) rivers to the counties they flow through.
4. **Census join.** Follow the steps in Section 6.4 for North Carolina's census tracts. How many tracts are there? How many have NULL population after the join, and why? Export the result and make a graduated map.
5. **Break it on purpose.** Repeat exercise 4, but open the CSV in a spreadsheet program first and save it again. Then add it to QGIS with *Detect field types* checked. What happens to the join? For which state would this mistake do the most damage?
6. **One-to-many.** Join a table of North Carolina counties to a layer of the 50 states, with the states as the destination. What value does North Carolina get? Then summarize the county table by state and join it again. Compare the two results.
7. **Relates.** In QGIS, set up a relation between a layer of states and a table of counties (*Project Properties > Relations*). Use the identify tool on North Carolina. What do you see that a join could not show?

6.8 Further reading and online resources

6.8.1 Books and open textbooks

- *GIS Fundamentals: A First Text on Geographic Information Systems*, by Paul Bolstad (Bolstad 2022). Chapter 8, "Attribute Data and Tables," covers relational databases, keys, and table joins.
- *Understanding GIS through Sustainable Development Goals*, by Paul Holloway (Holloway 2023). Includes worked table joins in QGIS and ArcGIS Pro.
- *Essentials of Geographic Information Systems*, by Jonathan Campbell and Michael Shin. Chapter 5, "Geospatial Data Management," covers databases, keys, and joins: https://open.umn.edu/opentextbooks/textbooks/67.
- *Database System Concepts*, by Abraham Silberschatz, Henry Korth, and S. Sudarshan (7th ed., McGraw-Hill, 2019), the source of the STUDENT and TAKES example, and a standard text on relational databases.
- *PostGIS in Action*, by Regina Obe and Leo Hsu (3rd ed., Manning, 2021), for joins inside a spatial database.

6.8.2 Databases

- PostgreSQL, "Joins Between Tables," from the official tutorial: https://www.postgresql.org/docs/current/tutorial-join.html.
- PostGIS, the spatial extension for PostgreSQL: https://postgis.net.
- SQLite, "Most Widely Deployed and Used Database Engine": https://www.sqlite.org/mostdeployed.html.

- OGC GeoPackage, the standard behind the format: https://www.geopackage.org.
- Codd, E. F., "A Relational Model of Data for Large Shared Data Banks" (Codd 1970): https://doi.org/10.1145/362384.362685.
- The GIS&T Body of Knowledge, "Database Management Systems" and "Relational Database Management Systems": https://www.ucgis.org/site/gis-t-body-of-knowledge.

6.8.3 Joins in QGIS and ArcGIS Pro

- QGIS User Guide, "The Vector Properties Dialog," including the Joins tab: https://docs.qgis.org/latest/en/docs/user_manual/working_with_vector/vector_properties.html.
- QGIS User Guide, "Creating one or many to many relations": https://docs.qgis.org/latest/en/docs/user_manual/working_with_vector/joins_relations.html.
- QGIS Tutorials and Tips, "Performing Table Joins," by Ujaval Gandhi, a step-by-step census join: https://www.qgistutorials.com/en/docs/3/performing_table_joins.html.
- ArcGIS Pro, "Add Join" tool: https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/add-join.htm.
- ArcGIS Pro, "Joins and relates": https://pro.arcgis.com/en/pro-app/latest/help/data/tables/joins-and-relates.htm.
- GDAL, "Comma Separated Value (.csv)," including how `.csvt` files set field types: https://gdal.org/en/stable/drivers/vector/csv.html.

6.8.4 SQL joins

- W3Schools, "SQL Joins," with a try-it-yourself editor for inner, left, right, and full joins: https://www.w3schools.com/sql/sql_join.asp.
- SQLBolt, lessons 6 and 7 on joins: https://sqlbolt.com.
- Wikipedia, "Join (SQL)": https://en.wikipedia.org/wiki/Join_(SQL).
- Wikipedia, "Cardinality (data modeling)": https://en.wikipedia.org/wiki/Cardinality_(data_modeling).

6.8.5 Census data and keys

- U.S. Census Bureau, "Understanding Geographic Identifiers (GEOIDs)": https://www.census.gov/programs-surveys/geography/guidance/geo-identifiers.html.
- U.S. Census Bureau, TIGER/Line shapefiles: https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html.
- data.census.gov, where the census tables are downloaded: https://data.census.gov.
- Ziemann, Eren, and El-Osta, "Gene name errors are widespread in the scientific literature" (Ziemann et al. 2016): https://doi.org/10.1186/s13059-016-1044-7.

7 Spatial Joins

Learning objectives

By the end of this chapter you should be able to:

1. Explain how a spatial join differs from an attribute join, and recognize questions that need one.
2. Tell the four kinds of spatial join apart: simple or summarized, inside or distance.
3. Use cardinality to decide between a simple and a summarized join.
4. Explain why a distance join must be run on projected data, and why projecting on the fly is not enough.
5. Plan a spatial join by naming the destination layer, the relationship, and the cardinality before opening a tool.
6. Run spatial joins in QGIS and ArcGIS Pro, and read their output.

Chapter 6 joined tables that share a key, such as a census table and census tracts that both carry a GEOID. Many questions involve two layers with no shared field at all:

- **How many EMS calls happened in each neighborhood?** The calls are points with an address and a time. The neighborhoods are polygons with a name. Neither table holds the other's ID.
- **How many students attend school in each district?** The schools are points with an enrollment count. The districts are polygons.
- **How much revenue does a coffee chain earn in each sales region?** Revenue is recorded at each store, a point. The regions are polygons.

In each case, the only thing linking the two layers is *where* their features are. A call belongs to the neighborhood it falls in; a school belongs to the district that contains it. A **spatial join** uses that location to match the records, and then appends attributes just like the joins in Chapter 6.

7.1 What is a spatial join?

7.1.1 Location instead of a key

An attribute join matches records whose key fields hold the same value (Figure 7.1, left). A **spatial join** matches features whose geometries have a spatial relationship (Figure 7.1, right). Instead of asking "does this FIPS code equal that one?", the software asks "does this point fall inside that polygon?" or "which of these airports is closest to this city?"

Everything else about a join stays the same:

- The **destination layer** (ArcGIS calls it the target features) is the one whose features you keep, and to which fields are added.

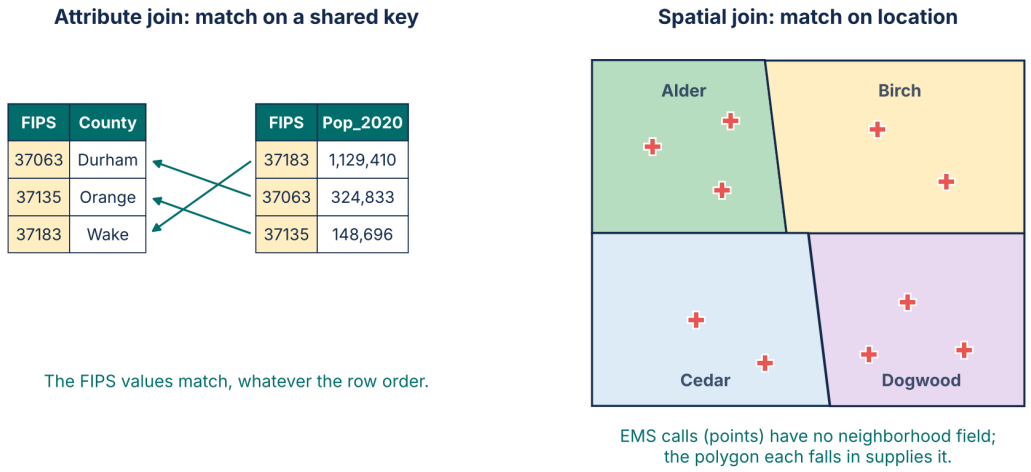


Figure 7.1: Left: an attribute join matches rows by a shared key. Right: a spatial join matches features by location, here EMS calls (points) falling inside counties (polygons). The counties are made up.

- The **source layer** (the join features) supplies the fields.
- The output is one table: every field of the destination layer, followed by the fields of the matching source feature. Destination features with no match are kept, with NULL in the new fields, unless you ask for matched features only.

Unlike an attribute join in the *Joins* tab, a spatial join usually writes a **new layer**. Your original layers are not changed.

7.1.2 Spatial relationships

Every spatial join needs a rule for deciding which source features match a destination feature. The rules come in two families:

- **Inside** relationships test containment or overlap: a point inside a polygon, a line crossing a polygon, a polygon within another polygon. In this chapter we call a join that uses them an **inside join**.
- **Distance** relationships test proximity: the closest feature, or every feature within a set distance. A join that uses them is a **distance join**.

The software calls these rules **spatial predicates** (QGIS's *geometric predicates*, ArcGIS Pro's *match options*). The names you will see in the tool dialogs are:

Table 7.1: Spatial predicates in QGIS and ArcGIS Pro.

Predicate	True when the two geometries…
intersects	share any point at all (the most general test)
contains	one has the other entirely inside it
within (*are within*)	one lies entirely inside the other (the reverse of *contains*)
touches	share a boundary, but their interiors don't overlap
overlaps	share some, but not all, of their area, and are the same kind of geometry
crosses	a line passes through a polygon or across another line
equals	are exactly the same geometry
disjoint	share no point at all

Predicate	True when the two geometries…
within a distance	are closer than a set distance
closest (nearest neighbor)	are the nearest pair

For points in polygons, *intersects* and *within* give almost the same answer. They differ only for a point that sits exactly on a boundary, which *intersects* counts and *within* does not.

Did you know..?

To test whether a point is inside a polygon, most GIS software draws an imaginary line from the point out to infinity and counts how many times it crosses the polygon's edge. An odd number of crossings means the point is inside; an even number means it is outside. The *ray-casting* test works for any polygon, however twisted, and it has been used since the earliest computer graphics in the 1960s.

7.2 Four kinds of spatial join

7.2.1 Simple or summarized, inside or distance

Every spatial join can be described with two choices (Figure 7.2):

- **Inside or distance:** is the relationship containment or proximity?
- **Simple or summarized:** does each destination feature receive the attributes of *one* source feature, or a *summary* of many?

Reading the four panels:

- **Simple inside join** (top left). *Which county is each school in?* The schools are the destination; the counties are the source. Each school gets the name and attributes of the one county that contains it.
- **Summarized inside join** (top right). *How many schools are in each county?* Now the counties are the destination and the schools are the source. Each county gets a count of the schools inside it.
- **Simple distance join** (bottom left). *Which attraction is closest to each hotel, and how far away is it?* Each hotel gets the attributes of its nearest attraction, plus a field holding the distance.
- **Summarized distance join** (bottom right). *How many attractions are closest to each hotel?* Each attraction is assigned to its nearest hotel, and each hotel gets a count of the attractions assigned to it.

Notice that the same two layers can answer different questions. Schools-to-counties and counties-to-schools are both inside joins; which layer is the destination depends on what one row of the answer should be.

7.2.2 A simple distance join: cities and airports

Figure 7.3 shows a simple distance join with real data. The destination layer is North Carolina's cities, and the source layer is the region's major airports. Each city receives the code



Figure 7.2: The four kinds of spatial join. In each panel, the destination is the layer the question asks about. The counties, hotels, and attractions are made up.

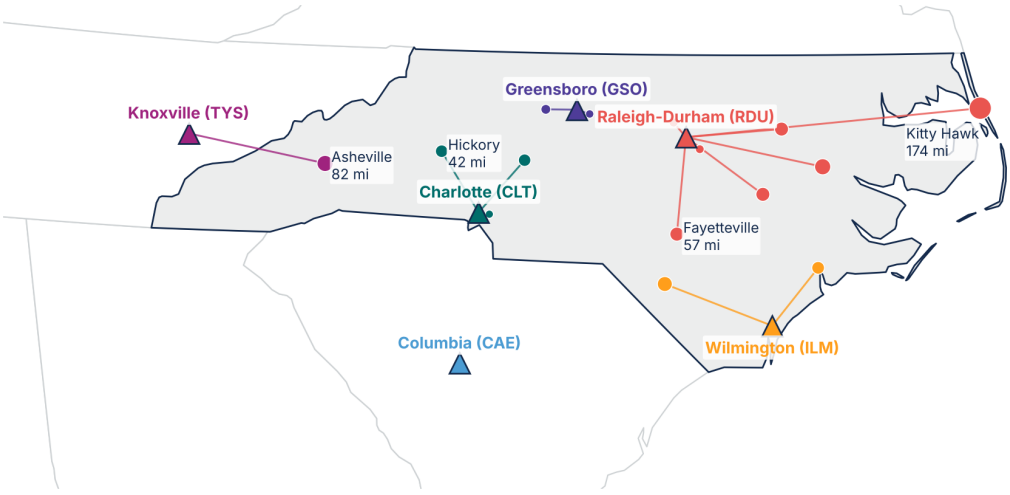


Figure 7.3: A simple distance join: each North Carolina city in Natural Earth's populated places layer joined to its nearest airport in Natural Earth's airports layer, with distances measured in NC State Plane (EPSG:32119). The airports layer includes only larger airports, so Asheville's own airport is missing and Asheville is matched to Knoxville, Tennessee.



of its nearest airport and the distance to it. The cities are colored by their airport and sized by the distance.

Two things stand out. First, a distance join pays no attention to boundaries: Asheville's nearest airport in this layer is across the state line in Tennessee. Second, the answer is only as good as the source layer. Asheville has a regional airport, but this small-scale layer leaves it out, so the join gives an answer that is correct for the data and wrong for the world. Always ask whether the source layer contains every feature that could be the right answer.

7.2.3 An inside join: septic systems and geology

A county health department wants to know the bedrock under every septic system, because septic systems on permeable rock can leak into groundwater. The destination is the septic systems (points), and the source is a geology map (polygons). Each septic system falls inside exactly one geology unit, so each receives exactly one rock type. The output table is the septic table with the geology unit's name and symbol added to each row, ready for a query such as `"ROCKTYPE" = 'Limestone'`.

7.3 Cardinality decides simple or summarized

7.3.1 The rule of joining, again

Chapter 6's rule of joining applies to spatial joins too. Ask how many source features can match each destination feature:

- **One-to-one or many-to-one: a simple join.** Each septic system lies in one geology unit; many septic systems can share a unit. Each city has one nearest airport; many cities can share one. There is never more than one set of attributes to copy, so a simple join copies them directly.
- **One-to-many or many-to-many: a summarized join.** Each county contains many schools. Each river flows through many counties. There is no single set of attributes to copy.

A **summarized join** solves the one-to-many problem by collapsing all the matches into statistics. It finds every source feature that matches each destination feature, summarizes them, and appends one row of summary values. Every summarized join reports a **count** of matches (ArcGIS Pro names the field `Join_Count`). You can also ask for the **sum**, **mean**, **minimum**, **maximum**, and other statistics of any numeric field in the source. Summing an enrollment field over the schools in each county gives the number of students in each county, which answers the school district question from the start of the chapter.

7.3.2 Which layer is the destination?

The surest way to choose the destination is to picture the answer. **What should one row of the output represent?** That row is a feature of the destination layer.

- "Which county is each school in?" One row per school: schools are the destination.
- "How many schools are in each county?" One row per county: counties are the destination.

- "Which volcano is closest to each city?" One row per city: cities are the destination.

Most failed spatial joins come from picking the wrong destination layer, or from running a simple join where several features match. In the second case, the software quietly keeps one match (usually the first it finds) and throws the rest away.

Think about it

A summarized distance join counts the attractions closest to each hotel. If you drew a boundary around all the land closer to each hotel than to any other, what shape would each hotel's area be? (Hint: look up *Thiessen polygons*, also called Voronoi polygons.)

Did you know..?

One of the most famous maps in history is a summarized distance join done by hand. After the 1854 cholera outbreak in London, John Snow mapped the homes of the people who died, then drew a line around the area closer, by walking distance, to the Broad Street pump than to any other pump. Most of the deaths fell inside that area (Snow 1855). Chapter 1 tells more of the story.

7.4 Coordinate systems and distance joins

7.4.1 Distances come out in the data's units

A distance join reports distances in the **units of the source data's coordinate system**. If the layers are projected in meters, the distances are in meters. If the layers are stored in a geographic coordinate system (GCS), such as WGS84 (EPSG:4326) or NAD83 (EPSG:4269), the distances are in **decimal degrees**. The software does not warn you. You simply get a `Distance` field full of numbers like 0.47.

Degrees cannot be turned into miles or kilometers with one conversion factor. As Chapter 3 showed, a degree of latitude is always about 111 km (69 miles), but a degree of longitude shrinks toward the poles: about 111 km at the equator, 90 km at Chapel Hill, 79 km at 45° N, and 56 km at 60° N (Figure 7.4, left). A distance in degrees mixes two different scales.

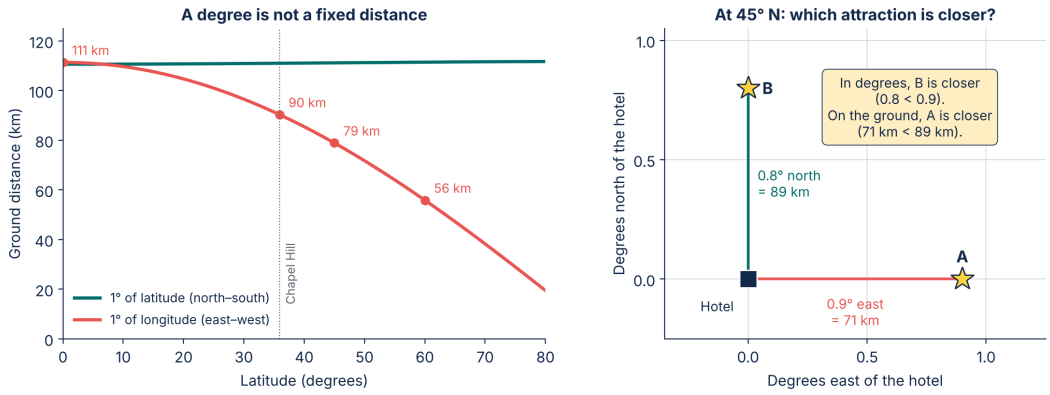


Figure 7.4: Left: the ground length of one degree of latitude and one degree of longitude, by latitude, on the WGS84 ellipsoid. Right: at 45° N, attraction B is closer to the hotel in degrees, but attraction A is closer on the ground.

The units are not just a labeling problem. They can change *which feature is nearest*. At 45° N, an attraction 0.9° east of a hotel is 71 km away, and one 0.8° north is 89 km away (Figure 7.4, right). In degrees, the northern attraction looks closer (0.8 < 0.9). On the ground, the eastern one is closer by 18 km. A distance join run on unprojected data would match the hotel to the wrong attraction.

The fix: **before a distance join, project both layers into a coordinate system that preserves distance well for your study area**, such as the right UTM zone or State Plane zone (Chapter 4). The distances then come out in meters or feet, and the nearest feature is the nearest on the ground.

> **Geodesic distances**
>
> Some newer tools can measure distances on the ellipsoid even when the data are in a GCS. ArcGIS Pro's Spatial Join, for example, has geodesic match options such as *Within a distance geodesic* and *Closest geodesic*. QGIS's *Join Attributes by Nearest* does not: it measures in the layer's own units. Unless you know what the tool does, projecting first is the safe habit.

7.4.2 Projecting on the fly is not enough

QGIS and ArcGIS Pro project layers **on the fly** (Chapter 4). If the map is set to NC State Plane, layers stored in WGS84 are drawn in State Plane too, and the map looks correct.

But the join tools read the coordinates **stored in the layers**, not the coordinates on the screen. Changing the map's coordinate system changes the picture, not the data. If the data on disk are still in degrees, the distances will be too.

To fix it, make projected copies of the layers and run the join on those:

- In **QGIS**, use *Vector > Data Management Tools > Reproject Layer*, or *Export > Save Features As* with a new CRS.
- In **ArcGIS Pro**, use the *Project* tool.

Then check the layer's properties to confirm the new CRS, and run the join again.

7.5 Spatial joins in practice

7.5.1 Six questions to answer first

Before opening any tool, work through these questions on paper:

1. **Sketch the layers.** What kinds of geometry are they: points, lines, or polygons?
2. **What should the output look like?** What does one row represent, and which fields should it have?
3. **Which layer is the destination?** The layer whose features you keep and add fields to: the one that one row of the output represents.
4. **Inside or distance?** Inside when one feature lies within, or crosses, another; distance when they don't touch.
5. **What is the cardinality?** Can a destination feature match more than one source feature?



6. **Simple or summarized?** One-to-one or many-to-one: simple. One-to-many or many-to-many: summarized, with the statistics you need.

And for a distance join, a seventh: **are both layers in a projected coordinate system with sensible units?**

7.5.2 Spatial joins in QGIS

QGIS splits spatial joins into separate tools, all in the Processing Toolbox and under *Vector > Data Management Tools*:

- **Join Attributes by Location** is the simple inside join. *Join to features in* is the destination layer. The *Features they* check boxes set the predicates. *By comparing to* is the source layer. The *Join type* sets the cardinality: *Create separate feature for each matching feature (one-to-many)* repeats a destination feature for every match, and *Take attributes of the first matching feature only (one-to-one)* keeps one. Check *Discard records which could not be joined* to drop unmatched features.
- **Join Attributes by Location (Summary)** is the summarized inside join. Choose the fields to summarize and the statistics (count, sum, mean, min, max, and others).
- **Join Attributes by Nearest** is the distance join. Set the maximum number of nearest neighbors (1 for the closest) and, optionally, a maximum distance. It adds fields with the distance and the neighbor's position.
- **Count Points in Polygon** is a shortcut for the most common summarized join: it adds a count of points to each polygon.

7.5.3 Spatial joins in ArcGIS Pro

ArcGIS Pro does every kind of spatial join with one **Spatial Join** tool, in the Geoprocessing pane, or from the layer's *Data* tab under *Add Spatial Join*. The kind of join depends on its settings:

- **Target features** is the destination layer; **Join features** is the source.
- **Join operation**: *Join one to one* gives one output row per target feature. When several join features match, it adds a `Join_Count` field and combines their values with the **merge rules** in the field map (first, sum, mean, minimum, maximum, and others). *Join one to many* writes one output row for each match.
- **Keep all target features**: checked, unmatched targets are kept (a left join); unchecked, they are dropped.
- **Match option**: the predicate, such as *Intersect*, *Completely contains*, *Within a distance*, or *Closest*.
- **Search radius**: for *Within a distance* and *Closest*.
- **Distance field name**: for *Closest*, the field that will hold the distance.

For example, to count the motorcycle crashes within 30 feet of each road, set Roads as the target, the crashes as the join features, *Join one to one*, *Keep all target features*, and *Within a distance* with a 30-foot radius. The `Join_Count` of each road is the number of crashes near it (Figure 7.5).

Table 7.2 summarizes which tool to use for each kind of join.

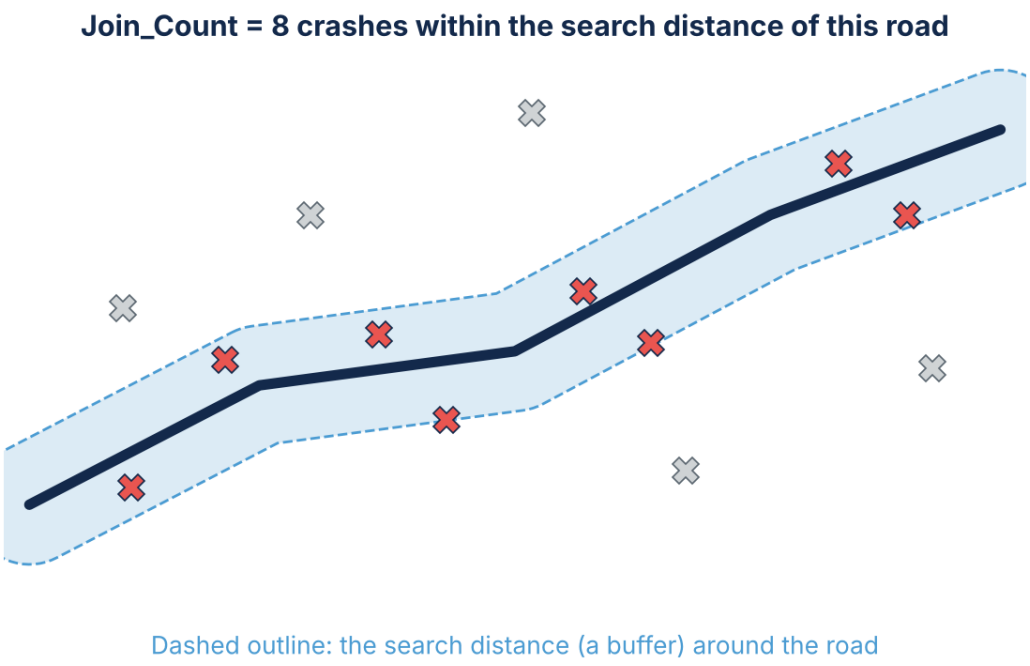


Figure 7.5: A "within a distance" join: every crash inside the road's search distance (the dashed outline) counts toward the road's Join_Count. The road and crashes are made up.

Table 7.2: Tools for each kind of spatial join, as named in QGIS 3 and ArcGIS Pro 3.

Join	QGIS	ArcGIS Pro Spatial Join
Inside, simple	Join Attributes by Location, first matching feature only	Match option *Intersect* or *Within*; Join one to one
Inside, summarized	Join Attributes by Location (Summary), or Count Points in Polygon	Join one to one with merge rules (Sum, Mean); adds `Join_Count`
Distance, closest	Join Attributes by Nearest, 1 neighbor; adds a distance field	Match option *Closest*; set a distance field name
Distance, count within a radius	Buffer, then Join Attributes by Location (Summary)	Match option *Within a distance*; Join one to one

7.5.4 Reading the output

Open the attribute table of the output layer. The destination's fields come first, then the source's. Then check three things:

- **The row count.** A simple join with every destination feature kept should have exactly as many rows as the destination layer. More rows means a one-to-many join repeated some features; fewer means unmatched features were dropped.
- **NULLs.** Rows with NULL in the joined fields had no match. Is that what you expect? A point in the ocean, outside every county, has no county.
- **Distances.** If there is a distance field, are the values sensible? Numbers below 1 for features kilometers apart mean the data were in degrees.

7.6 Worked examples

For each example, answer the six questions before reading the solution.

7.6.1 Example 1: congressional districts with more than 10 earthquake deaths

Find every congressional district that has had more than 10 earthquake deaths. You have earthquake points with a `DEATHS` field, and congressional district polygons.

Solution

- **Output:** one row per district, with total deaths.
- **Destination:** congressional districts. **Source:** earthquakes.
- **Relationship:** inside (points in polygons).
- **Cardinality:** one-to-many; a district can contain many earthquakes.
- **Join:** summarized, with the **sum** of `DEATHS`.
- Then select the districts with an attribute query (Chapter 5), such as `"DEATHS_sum" > 10` (QGIS names the summed field this way).

7.6.2 Example 2: a pollution risk index for rivers

Score each river by the total number of people living in the counties it flows through. You have river lines and county polygons with a population field.

Solution

- **Output:** one row per river, with the total population.
- **Destination:** rivers (lines). **Source:** counties (polygons).
- **Relationship:** inside, using *intersects* (a river crosses each county it flows through).
- **Cardinality:** one-to-many; each river crosses many counties.
- **Join:** summarized, with the **sum** of population. Rivers that flow past more people score higher.

7.6.3 Example 3: the closest volcano to each U.S. city

Make a table of the volcano closest to each city in the United States, and how far away it is.

Solution

- **Output:** one row per city, with its nearest volcano and the distance.
- **Destination:** cities. **Source:** volcanoes.
- **Relationship:** distance (closest feature).
- **Cardinality:** many-to-one; each city has one closest volcano, though many cities can share it.
- **Join:** simple distance join.
- **Coordinate system:** project both layers first, using an equidistant or equal-area projection for the whole country (such as USA Contiguous Albers, EPSG:5070) so that the distances are in meters, not degrees. Better still, use a tool that measures geodesic distances.

7.7 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

7.8 Summary

- A **spatial join** attaches the attributes of a **source layer** to a **destination layer** by location, when the two share no key field.
- An **inside join** matches features that contain, lie within, or intersect each other. A **distance join** matches the closest feature, or every feature within a set distance, and can record the distance.
- The rules are called **spatial predicates**: intersects, contains, within, touches, overlaps, crosses, equals, disjoint, within a distance, and closest.
- A **simple join** copies one source feature's attributes to each destination feature. Use it for **one-to-one** and **many-to-one** relationships.
- A **summarized join** counts and summarizes all the matching source features. Use it for **one-to-many** and **many-to-many** relationships. It always adds a count, and can add sums, means, minimums, and maximums.
- Choose the destination layer by asking what **one row of the output** should represent.
- A distance join reports distances in the stored units of the data. **Project the layers** before a distance join; projecting the map on the fly changes the display, not the data.
- QGIS has a tool for each kind of join; ArcGIS Pro's Spatial Join tool does them all through its settings. Check the output's row count, NULLs, and distances.

7.9 Exercises

1. **Name the join.** For each question, name the destination and source layers, the relationship, the cardinality, and the kind of join: (a) Which watershed is each well in? (b) How many fast-food restaurants are within 1 km of each school? (c) What is



the total length of streams in each county? (d) Which fire station is closest to each house? (e) How many houses does each fire station serve as the closest station?

2. **Count points in polygons.** Download Natural Earth's populated places and country boundaries. In QGIS, count the populated places in each country, and sum their `pop_max`. Which country has the most places? Does that match the country with the largest summed population?
3. **Simple inside join.** Join the countries to the populated places, so that each place gets its country's name and continent. How many places fall in no country, and why?
4. **Degrees versus meters.** Using Natural Earth's populated places and airports, run *Join Attributes by Nearest* on the layers as downloaded (EPSG:4326). Then project both into a suitable projected CRS for your study area and run it again. Compare the distance fields. Did any place get a different nearest airport?
5. **Within a distance.** Using North Carolina road and crash data from NC OneMap or NCDOT, count the crashes within 30 meters of each road segment. Which segments have the most? What would change with a 100-meter radius?
6. **Plan before you click.** Work through the six questions for the earthquake, river, and volcano examples in Section 7.6, then run one of them with data from the USGS and Natural Earth.

7.10 Further reading and online resources

7.10.1 Books and open textbooks

- *GIS Fundamentals: A First Text on Geographic Information Systems*, by Paul Bolstad (Bolstad 2022). Chapter 9, "Basic Spatial Analyses," covers spatial selection and joins.
- *Understanding GIS through Sustainable Development Goals*, by Paul Holloway (Holloway 2023). Includes spatial joins in QGIS and ArcGIS Pro.
- *Intro to GIS and Spatial Analysis*, by Manuel Gimond, the chapter on vector operations: https://mgimond.github.io/Spatial/.
- *The Nature of Geographic Information*, by David DiBiase and colleagues at Penn State: https://www.e-education.psu.edu/natureofgeoinfo/.
- *PostGIS in Action*, by Regina Obe and Leo Hsu (3rd ed., Manning, 2021), on spatial joins in SQL with `ST_Intersects`, `ST_Contains`, and `ST_DWithin`.

7.10.2 Spatial joins in QGIS

- QGIS User Guide, "Vector general" algorithms, including Join Attributes by Location, Join Attributes by Location (Summary), and Join Attributes by Nearest: https://docs.qgis.org/latest/en/docs/user_manual/processing_algs/qgis/vectorgeneral.html.
- QGIS User Guide, "Vector analysis" algorithms, including Count Points in Polygon: https://docs.qgis.org/latest/en/docs/user_manual/processing_algs/qgis/vectoranalysis.html.
- QGIS Tutorials and Tips, "Performing Spatial Joins," by Ujaval Gandhi: https://www.qgistutorials.com/en/docs/3/performing_spatial_joins.html.
- QGIS Tutorials and Tips, "Nearest Neighbor Analysis": https://www.qgistutorials.com/en/docs/3/nearest_neighbor_analysis.html.

7.10.3 Spatial joins in ArcGIS Pro

- ArcGIS Pro, "Spatial Join" tool: https://pro.arcgis.com/en/pro-app/latest/tool-reference/analysis/spatial-join.htm.
- ArcGIS Pro, "Select Layer By Location" tool, with diagrams of every spatial relationship: https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/select-layer-by-location.htm.
- ArcGIS Pro, "Summarize Within" tool, another way to run a summarized inside join: https://pro.arcgis.com/en/pro-app/latest/tool-reference/analysis/summarize-within.htm.

7.10.4 Spatial relationships

- Wikipedia, "DE-9IM," the model behind the spatial predicates: https://en.wikipedia.org/wiki/DE-9IM.
- Wikipedia, "Point in polygon," on the ray-casting test: https://en.wikipedia.org/wiki/Point_in_polygon.
- PostGIS, "Spatial Relationships," a clear illustrated guide to the predicates: https://postgis.net/docs/using_postgis_query.html.
- GeoPandas User Guide, "Merging data," including spatial joins in Python: https://geopandas.org/en/stable/docs/user_guide/mergingdata.html.
- The GIS&T Body of Knowledge, "Topological relationships" and "Overlay & combination operations": https://www.ucgis.org/site/gis-t-body-of-knowledge.

7.10.5 Data for practice

- Natural Earth, populated places, airports, and country boundaries: https://www.naturalearthdata.com.
- USGS Earthquake Hazards Program, earthquake catalog downloads: https://earthquake.usgs.gov/earthquakes/search/.
- Smithsonian Global Volcanism Program, volcano locations: https://volcano.si.edu.
- NC OneMap, North Carolina's data portal: https://www.nconemap.gov.
- U.S. Census Bureau, TIGER/Line shapefiles, including congressional districts and school districts: https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html.



Part II

Unit 2: Spatial Data Quality

Every dataset has a source, a method, and limits. This unit is about where spatial data come from and how good they are. It starts with the largest source of free data about people and places in the United States, the census, and how to read its estimates honestly. It then turns to two ways of making new GIS layers from things that aren't yet spatial data, addresses and scanned maps, and how to measure the error each one adds. It ends with data made by volunteers, in OpenStreetMap, and the ways the crowd checks its own work. By the end of the unit you can find, join, create, and judge spatial data in QGIS and ArcGIS Pro.

Chapter 8: Census Data

The U.S. Census Bureau decides how many members of Congress each state gets and where trillions of federal dollars go. This chapter separates authoritative from asserted data and describes the decennial census, the American Community Survey, and the Economic Census. It explains the hierarchy of census geography, from states to blocks, and the GEOIDs that identify each area. It shows how to choose between decennial, ACS 1-year, and ACS 5-year data, how to read a margin of error, and how to join a census table to its boundaries in ArcGIS Pro.

Chapter 9: Geocoding and Georeferencing

Many records describe a location without coordinates: a street address, or a place on an old paper map. This chapter covers both ways of giving them coordinates. Geocoding matches addresses against reference files, from ZIP Code centroids to parcels, and interpolates positions along streets; match scores and match rates tell you how well it worked. Georeferencing ties a scanned map to the ground with control points and a transformation, and residuals and RMSE measure its error. Both are shown step by step in ArcGIS Pro and QGIS.

Chapter 10: OpenStreetMap and Data Quality

What makes geographic data good enough to use? This chapter defines spatial data quality by its seven elements, from lineage to completeness, then looks at volunteered geographic information, Big Data, and the FAIR principles. It explains how OpenStreetMap works, from its license to its nodes, ways, and tags, and how its quality is measured against authoritative data and from its own history. It ends with crisis mapping, from Haiti in 2010 to the 2015 Nepal earthquake, where thousands of new volunteers mapped a country in weeks.

8 Census Data

Learning objectives

By the end of this chapter you should be able to:

1. Tell authoritative data from asserted data, and name the main kinds of population and place data.
2. Describe what the decennial census, the American Community Survey (ACS), and the Economic Census collect.
3. Explain the hierarchy of census geography, from the nation to the block, and which areas sit off the spine.
4. Choose the right census product (decennial, ACS 1-year, or ACS 5-year) for a geography and a question.
5. Read a margin of error, and judge whether an estimate is reliable enough to map.
6. Find a census table, build or read its GEOID, and join it to census boundaries in ArcGIS Pro or QGIS.

Chapter 6 joined a census table to census tracts to learn how joins work. This chapter turns to the data itself. The U.S. Census Bureau is the largest source of free, consistent information about people and places in the country. Its numbers decide how many members of Congress each state sends to Washington, where trillions of federal dollars go, where companies open stores, and how public health workers turn case counts into rates.

Census data are also easy to misuse. The same variable can come from three different products, published for eight levels of geography, and many values are estimates with sizable errors. This chapter is about choosing the right product and geography for a question, reading the numbers honestly, and getting them onto a map.

This chapter starts the unit on spatial data quality. Every dataset has a source, a method, and limits. The census documents all three unusually well, so it is a good place to learn to ask about them.

8.1 Population and place data

8.1.1 Authoritative and asserted data

Data about people and places come from two very different kinds of source:

- **Authoritative data** come from official agencies, such as the Census Bureau or the U.S. Geological Survey. They are collected under documented standards, and they ship with **metadata** that describe their sources, methods, dates, and accuracy.
- **Asserted data**, also called volunteered geographic information, are contributed by the public. OpenStreetMap is the best-known example (Chapter 10). Their accuracy is undocumented and uneven, and they often have little or no metadata. Busy, popular places tend to be mapped well; quiet ones may be sparse. Where contributors are

active, asserted data can be more current, and even more accurate, than old official data.

Neither kind is always better. Before using any dataset, ask who made it, how, when, and whether there is metadata that tells you how accurate it is.

8.1.2 Four kinds of population and place data

Most data about people and places fall into four groups:

- **Demographic data** describe who lives in a place: counts of people and households, and their age, sex, race and ethnicity, and household type. The main sources are the decennial census, the ACS, and the Census Bureau's annual population estimates.
- **Socioeconomic data** describe economic and social conditions: education, income, employment, poverty, and housing costs. They are often used to characterize neighborhoods, for example in **deprivation indices**, and as stand-ins for health, because they track health outcomes closely. The main source is the ACS, so every value is an estimate.
- **Environmental and neighborhood data** describe the places themselves: air and water pollution, flood zones, temperature, the habitats of disease-carrying insects, and amenities such as parks, grocery stores, and sidewalks. They usually come from outside the Census Bureau (the EPA, NOAA, and local governments), and are linked to census geography to describe the people exposed.
- **Health data** include vital statistics (births and deaths), morbidity (illness), and the locations of hospitals, clinics, and doctors. They are often published only for counties or states, to protect privacy. Census counts supply the **denominators** that turn health counts into comparable rates, such as deaths per 100,000 residents.

A rule runs through all of them: **the finer the geography, the fewer variables are published, and the less precise they are.** A state has thousands of detailed census tables; a block has a handful of counts.

8.2 The U.S. Census

8.2.1 The constitutional count

The census is written into the U.S. Constitution. Article I, Section 2 requires an "actual Enumeration" of the population every ten years, to divide the seats in the House of Representatives among the states. The first census was taken in 1790; the 2020 Census was the 24th, and the next is in 2030.

The **decennial census** counts every resident, whatever their age, citizenship, or housing status. Since 2010, it has asked only a short list of questions: age, sex, race, Hispanic origin, each person's relationship to the householder, and whether the home is owned or rented. Everything else, from income and education to commuting and home values, now comes from a separate survey, the ACS.

Did you know..?

Your individual census answers are sealed for 72 years. Until then, the Census Bureau may publish only statistics, and it cannot share your answers with any other agency, including the police, immigration authorities, or the IRS. In April 2022, the National

Archives released the individual records of the 1950 census, and the 2020 records will be opened in 2092. Genealogists wait for each release eagerly.

8.2.2 The big three surveys

The Census Bureau runs dozens of surveys, but three matter most for mapping (Table 8.1).

Table 8.1: The three Census Bureau surveys used most in GIS.

Survey	How often	What it covers	Kind of numbers
Decennial census	Every 10 years (2010, 2020, 2030)	Every resident: age, sex, race, Hispanic origin, household relationship, owning or renting	Counts of the whole population
American Community Survey (ACS)	Every year, published as 1-year and 5-year estimates	About 3.5 million addresses a year: income, education, employment, commuting, housing costs, language, disability, and much more	Estimates from a sample, each with a margin of error
Economic Census	Every 5 years, in years ending in 2 and 7	Businesses: number of firms, employees, payroll, and sales, by industry	Counts and totals for businesses

Most of the socioeconomic variables you will map come from the ACS. The Bureau also publishes annual **population estimates** between censuses, built from the last census plus records of births, deaths, and migration.

8.2.3 Why census data matter

Census numbers drive decisions in three arenas:

- **Political power.** The decennial count divides the 435 seats in the House among the states (apportionment), and states use the census to redraw congressional and legislative districts (redistricting).
- **Money.** In fiscal year 2021, 353 federal programs, including Medicaid, SNAP, and Head Start, used census data to distribute more than $2.8 trillion (U.S. Census Bureau 2023).
- **Business, planning, and health.** Companies use census demographics to choose store sites and gauge labor markets. Planners use commuting times to plan transit, and the age and tenure of housing to plan for growth. Health departments use census counts to compute rates and to find groups that are hit harder by disease. Emergency managers use them to find residents, such as older adults or households without a car, who may need help to evacuate.

8.2.4 The census is not perfect

Three limits are worth knowing:

- **Undercounts and overcounts.** Some groups are missed more than others. After each census, the Bureau runs a **Post-Enumeration Survey** to measure the errors. For 2020, it estimated that the census missed about 5 percent of Hispanic residents and 3.3 percent of Black residents, and an even larger share of American Indians living on reservations, while overcounting non-Hispanic White and Asian residents (U.S. Census Bureau 2022). Young children are also undercounted, census after census.
- **Differential privacy.** For 2020, the Bureau began adding carefully calibrated statistical noise to its published tables, so that individuals cannot be identified by combining them. Totals for states are exact, but counts for small areas, especially blocks, can look odd: a block with houses but no people, or a few children with no adults.
- **Change.** Boundaries, questions, and methods change from census to census. Comparing 2010 and 2020 tract data, for example, requires care, because many tracts were split or redrawn.

The Census Bureau's Vintage 2025 population estimates, released in January 2026, are a reminder that the population itself keeps changing: U.S. growth slowed to about half a percent between 2024 and 2025, mostly because fewer people moved to the country from abroad. Estimates like these are revised every year, so always check which vintage you are using.

8.3 Census geography

8.3.1 The spine

Census data are published for a hierarchy of areas (Figure 8.1). The main line, called the **spine**, runs from the nation through regions, divisions, states, counties, census tracts, and block groups, down to blocks. **Each level nests completely inside the one above.** A block is always inside one block group, which is inside one tract, which is inside one county. Because of that, you can add up the blocks in a tract to get the tract, or the tracts in a county to get the county.

Other areas sit **off the spine**. They are built from blocks (or, for metropolitan areas, from counties), but they don't fit neatly inside the spine's levels:

- **Places**: incorporated cities and towns, and **census-designated places** for unincorporated communities.
- **ZIP Code Tabulation Areas (ZCTAs)**: approximations of ZIP Code areas built from blocks. ZIP Codes are mail routes, not areas, so a ZCTA is never an exact match.
- **Metropolitan and micropolitan statistical areas**: groups of counties tied together by commuting.
- **School districts, and congressional and state legislative districts.**
- **American Indian, Alaska Native, and Native Hawaiian areas**, which can cross state lines.

Off-spine areas cross the spine's boundaries: a city can spill across two counties, and a ZCTA can cross tracts. You can't get a ZCTA's value by adding up tracts.

8.3.2 Tracts, block groups, and blocks

Three small levels are used most in local studies (Figure 8.2):

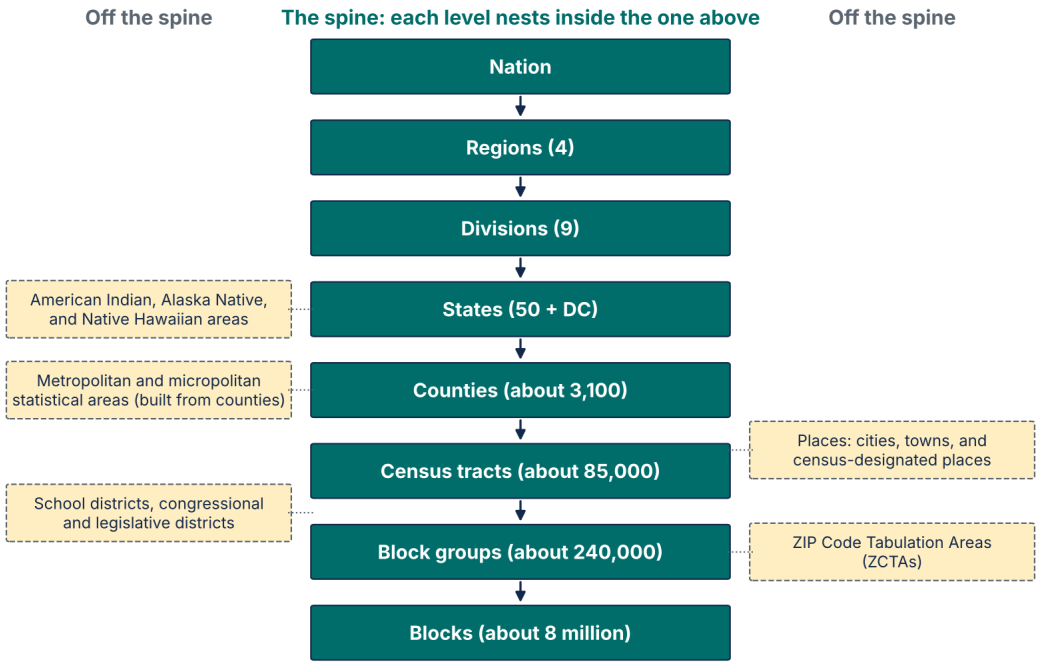


Figure 8.1: The hierarchy of census geography. The levels on the spine nest inside each other; the areas off the spine are built from blocks or counties but cross the spine's boundaries. Redrawn after the Census Bureau's "Standard Hierarchy of Census Geographic Entities."

- **Census tracts** are relatively permanent subdivisions of a county, with 1,200 to 8,000 people and about 4,000 on average. There are about 85,000 of them. They are updated each decade as the population changes, by splitting tracts that grew and merging ones that shrank. Tracts are the usual unit for neighborhood and health studies.
- **Block groups** are clusters of blocks within a tract, usually with 600 to 3,000 people. There are about 240,000. They are the smallest areas with ACS estimates, and those estimates often carry large errors.
- **Blocks** are the smallest areas, bounded by streets, streams, rail lines, and other visible features. The 2020 Census had more than 8 million of them, many with no one living there at all. Blocks get decennial counts only, with no ACS estimates.

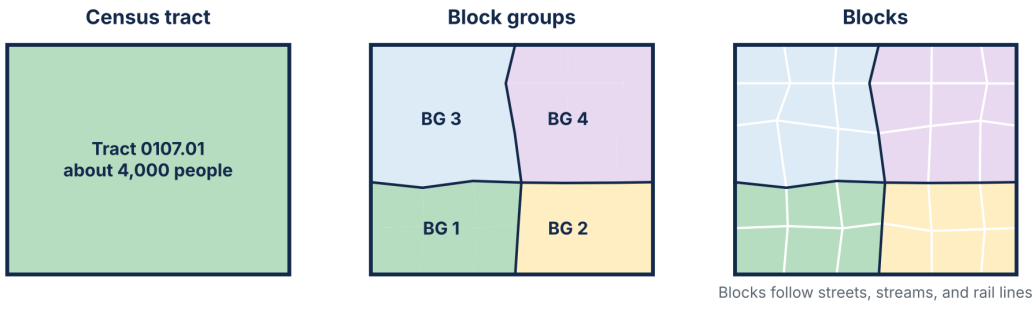


Figure 8.2: A census tract (left) is divided into block groups (middle), which are divided into blocks (right). Each level nests completely in the one above. The boundaries are made up.

8.3.3 GEOIDs

Every census area has a **GEOID**, made by stringing together the FIPS codes of the levels above it (Chapter 6). The GEOID is the key that links census tables to boundaries.

Table 8.2: GEOID structure for the spine's main levels.

Geography	GEOID structure	Digits	Example
State	SS	2	37 (North Carolina)
County	SS + CCC	5	37135 (Orange County)
Tract	SS + CCC + TTTTTT	11	37135010701
Block group	SS + CCC + TTTTTT + B	12	371350107011
Block	SS + CCC + TTTTTT + BBBB	15	371350107011003

Figure 8.3 breaks a block GEOID into its parts. The block group is not a separate piece: it is the first digit of the four-digit block number.

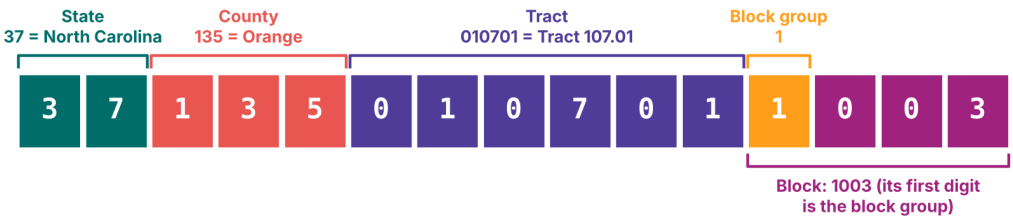


Figure 8.3: The 15 digits of a block GEOID. The block group (orange) is the first digit of the block number.

Store GEOIDs as **text**. As a number, Alabama's state code 01 becomes 1, and California's tracts lose their leading zero.

8.4 Census products and variables

8.4.1 What is published where

Smaller geographies get fewer tables (Table 8.3). The decennial census publishes the same few thousand variables for every level, down to the block. The ACS publishes far more, but not for every area.

Table 8.3: Approximate number of variables published for each geography. Counts vary by year.

Geography	Decennial (2020)	ACS 1-year	ACS 5-year
Block	9,067	none	none
Block group	9,067	none	4,108
Tract	9,067	none	17,168
County	9,067	36,400	18,123
State	9,067	36,400	19,564

8.4.2 ACS 1-year and 5-year estimates

The ACS samples addresses every month, and pools them into two products (Figure 8.4):

- **1-year estimates** use 12 months of data. They are the most current, but because the sample is small, they are published only for areas with **65,000 or more people**, and they are the least reliable. No tract or block group has 1-year estimates.
- **5-year estimates** pool 60 months of data. They are published for every area down to block groups, and they are the most reliable, but the least current: the 2020–2024 estimates describe the average over five years, not 2024.

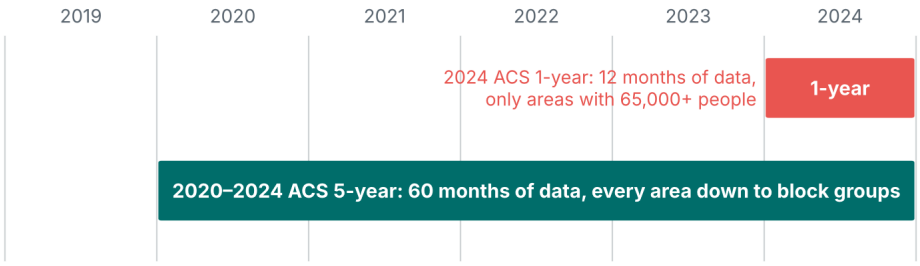


Figure 8.4: ACS 1-year and 5-year estimates released in 2025. The 1-year estimates cover 2024 alone; the 5-year estimates pool 2020 through 2024.

Use 1-year estimates for large areas when timeliness matters, such as tracking a state's income after a recession. Use 5-year estimates for small areas or small population groups. Never compare a 1-year estimate with a 5-year estimate, and don't compare 5-year estimates whose periods overlap (2015–2019 with 2017–2021, for example), because they share years of data.

8.4.3 Counts and estimates

Decennial counts are full counts: everyone was asked. ACS values are **estimates** from a sample of about one household in 40 each year, and every one comes with a **margin of error (MOE)**.

A margin of error works like the one on an opinion poll. If a poll finds that 55 percent of people prefer Coffee A, with a margin of error of ±3 points, the true share is probably between 52 and 58 percent. The Census Bureau's margins of error use a **90 percent confidence level**: if the survey were repeated 100 times, about 90 of the ranges would contain the true value.

Figure 8.5 shows five tract estimates of median household income. Tract D's estimate, $39,800, looks lower than Tract A's $48,200. But D's margin of error is $15,300, so its true value could be anywhere from about $24,500 to $55,100. The two ranges overlap widely, so the data can't tell you which tract has the lower income.

A common way to judge an estimate is its **coefficient of variation (CV)**, the standard error as a percentage of the estimate:

$$\text{CV} = \frac{\text{MOE}/1.645}{\text{estimate}} \times 100$$

The 1.645 turns a 90 percent margin of error into a standard error. A widely used rule of thumb treats a CV under 12 percent as reliable, 12 to 40 percent as usable with caution,

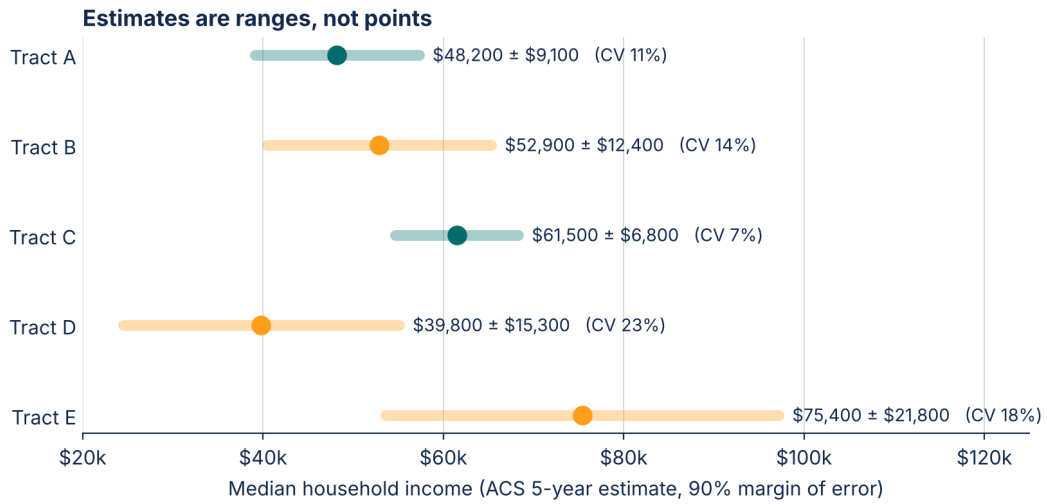


Figure 8.5: Made-up ACS estimates of median household income for five tracts, with their 90 percent margins of error. Green marks a coefficient of variation (CV) under 12 percent; orange, 12 to 40 percent.

and over 40 percent as unreliable. When many of your estimates are unreliable, combine them: use larger areas (tracts instead of block groups), or broader categories (all people under 18 instead of each age).

When you add estimates together, the margin of error of the sum is approximately the square root of the sum of the squared margins of error:

$$\text{MOE}_{\text{sum}} = \sqrt{\text{MOE}_1^2 + \text{MOE}_2^2 + \cdots}$$

So combining two tracts with MOEs of 300 and 400 people gives an MOE of 500, not 700.

Think about it

You are mapping the share of households without a car in each block group of Durham County, to plan bus service. Half the block groups have a CV over 40 percent. What could you do instead? What would you lose by doing it?

8.4.4 Choosing a product and a geography

Put the pieces together like this:

1. **What do you need to measure?** If it's a basic count (people, age, race, housing units), the decennial census has it, down to blocks. If it's income, education, commuting, or anything else social or economic, you need the ACS.
2. **How small are your areas?** Blocks get decennial counts only. Tracts and block groups get ACS 5-year estimates. Areas of 65,000 or more people also get ACS 1-year estimates.
3. **How current must it be?** For large areas, 1-year estimates are newer. For everything else, use the latest 5-year estimates.
4. **Do your areas nest?** Stay on the spine when you add up areas or compare levels. If you need ZCTAs or school districts, download them directly rather than building them from tracts.

5. **Are the estimates reliable enough?** Check the margins of error before you map.

Did you know..?

The 2020 Census was the first in which most households could answer online. About two-thirds of the households that responded on their own did so, and about 80 percent of those used the internet. The rest were counted by mail, by phone, or by census takers who visited the homes that didn't respond.

8.5 Getting census data into a GIS

8.5.1 Where to get the data

- **data.census.gov** is the Census Bureau's main site for tables. Search by topic, place, or table ID (such as `S1901`, income in the past 12 months), filter by geography, year, and survey, and download the table as a CSV.
- **The Census Data API** (api.census.gov) returns any table for any geography from a URL. Packages for R and Python use it to download tables with their boundaries in one step.
- **IPUMS NHGIS** (nhgis.org) offers census tables back to 1790, with boundary files for each year, and tables already adjusted to consistent boundaries over time.
- **TIGER/Line shapefiles** are the official boundaries for every census geography, with the GEOID in every record. The Bureau also publishes **cartographic boundary files**, simplified and clipped to the shoreline, which are better for small-scale maps.

8.5.2 Tutorial: a census join in ArcGIS Pro

This tutorial attaches median household income from the ACS to North Carolina's 100 counties, in ArcGIS Pro. The QGIS steps were covered in Chapter 6 (Section 6.4).

1. **Get the data.** Download a county boundary layer for North Carolina (for example the TIGER/Line or cartographic boundary file for U.S. counties, selected to `STATEFP = '37'`). On data.census.gov, find table **S1901** (Income in the Past 12 Months) from the latest ACS 5-year estimates, filter to all counties in North Carolina, and download it as a CSV.
2. **Start a project, add the data.** Create a new map project and add the county layer and the CSV. The CSV appears under *Standalone Tables* in the Contents pane.
3. **Check the key fields.** Open the county layer's fields view (*right-click > Data Design > Fields*). The county layer's `GEOID` is text, 5 characters. Open the CSV: ArcGIS Pro reads its ID column, after you clean it to the 5-digit code, as a number (Long integer). **The two key fields must have the same type**, so one has to change.
4. **Match the types.** A CSV is read-only in ArcGIS Pro, so add a field to the county layer instead. In the county attribute table, click *Add*, name the field `fips5`, set its type to *Long*, and save. Right-click the new column and choose *Calculate Field*, with the expression `fips5 = !GEOID!`. Each text code is converted to a number. (Because both keys are now numbers, the missing leading zeros don't matter.)
5. **Join.** Right-click the county layer, choose *Joins and Relates > Add Join*. Set the *Input Join Field* to `fips5`, the *Join Table* to the CSV, and the *Join Table Field* to its ID column. Click *Validate Join* to check how many records match: it should be all 100. Click *OK*.

6. **Check and make it permanent.** Open the attribute table and look for NULLs in the joined fields. Then right-click the layer and choose *Data > Export Features* to save a new feature class with the joined fields.
7. **Map it.** Symbolize the counties with *Graduated Colors* on the median household income field. Add the margin of error to the pop-up, so that readers can see how precise each estimate is.

8.6 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

8.7 Summary

- **Authoritative data** follow documented standards and come with metadata; **asserted data** are contributed by the public and their accuracy is undocumented.
- Population and place data are **demographic**, **socioeconomic**, **environmental**, or **health** data. The finer the geography, the fewer variables and the larger the errors.
- The **decennial census** counts everyone every ten years with a short form. The **American Community Survey** samples about 3.5 million addresses a year for social, economic, and housing data. The **Economic Census** profiles businesses every five years.
- Census data apportion the House, guide trillions of federal dollars, and serve business, planning, and public health. The census also has limits: uneven undercounts, differential privacy noise in small areas, and changing boundaries.
- Census geography has a **spine** (nation, regions, divisions, states, counties, tracts, block groups, blocks) whose levels nest, and **off-spine** areas such as places and ZCTAs that don't.
- A **GEOID** strings FIPS codes together: 2 digits for a state, 5 for a county, 11 for a tract, 12 for a block group, 15 for a block. Keep it as text.
- **ACS 1-year** estimates are current but cover only areas of 65,000+ people. **ACS 5-year** estimates cover every area down to block groups and are more reliable.
- Every ACS value has a **margin of error** at 90 percent confidence. Check the **coefficient of variation**, and combine areas or categories when estimates are unreliable.
- Join census tables to boundaries on the GEOID, with **matching field types**, and check the result for NULLs.

8.8 Exercises

1. **Authoritative or asserted?** Classify each source, and say what metadata you would look for: the 2020 Census, OpenStreetMap, a county's parcel layer, a crowd-sourced map of potholes, and the USGS National Hydrography Dataset.
2. **Pick the product.** For each question, choose the product (decennial, ACS 1-year, or ACS 5-year) and the geography: (a) the number of children under 5 in each block of Chapel Hill; (b) median rent in each tract of Wake County; (c) the poverty rate in North Carolina in the latest year; (d) the share of commuters who walk to work in each block group of Orange County.
3. **Read a GEOID.** What does the GEOID `371830541091` identify? Which state, which county, which tract, and which level of geography?

4. **Margins of error.** Download median household income (table B19013) for every tract in Orange County, NC, from the latest ACS 5-year estimates, including the margins of error. Compute the CV of each tract. How many are over 12 percent? Over 40 percent?
5. **County join.** Follow the ArcGIS Pro tutorial in Section 8.5 (or the QGIS steps in Chapter 6) to map median household income for North Carolina's counties. Which county has the highest income, and which the lowest? Are their margins of error small enough to be sure?
6. **Rates.** Download county population from the 2020 Census and a count of something by county (for example, hospitals, or COVID-19 cases from the state health department). Compute and map a rate per 100,000 residents. Why is a map of the raw counts misleading?

8.9 Further reading and online resources

8.9.1 Books and guides

- U.S. Census Bureau, *Understanding and Using American Community Survey Data: What All Data Users Need to Know* (2020), the official handbook, including how to work with margins of error: https://www.census.gov/programs-surveys/acs/library/handbooks/general.html.
- *Analyzing US Census Data: Methods, Maps, and Models in R*, by Kyle Walker (CRC Press, 2023), free online. The concepts chapters on census geography and margins of error are useful even without R: https://walker-data.com/census-r/.
- *GIS Fundamentals: A First Text on Geographic Information Systems*, by Paul Bolstad (Bolstad 2022). Chapter 7 discusses census data and other digital data sources.
- *The Nature of Geographic Information*, by David DiBiase and colleagues at Penn State. Chapter 3, "Census Data and Thematic Maps": https://www.e-education.psu.edu/natureofgeoinfo/c3.html.

8.9.2 Census Bureau resources

- data.census.gov, for finding and downloading tables: https://data.census.gov.
- Census Bureau, "Geography Program," including the hierarchy of census geographies: https://www.census.gov/programs-surveys/geography.html.
- Census Bureau, "Understanding Geographic Identifiers (GEOIDs)": https://www.census.gov/programs-surveys/geography/guidance/geo-identifiers.html.
- Census Bureau, TIGER/Line shapefiles: https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html.
- Census Bureau, cartographic boundary files: https://www.census.gov/geographies/mapping-files/time-series/geo/cartographic-boundary.html.
- Census Bureau, American Community Survey: https://www.census.gov/programs-surveys/acs.
- Census Bureau, Census Data API: https://www.census.gov/data/developers.html.
- Census Bureau, "Uses of Census Bureau Data in Federal Funds Distribution" (U.S. Census Bureau 2023).
- Census Bureau, Post-Enumeration Survey results for the 2020 Census (U.S. Census Bureau 2022): https://www.census.gov/library/stories/2022/03/who-was-undercounted-overcounted-in-2020-census.html.



- Census Bureau, "Disclosure Avoidance and the 2020 Census," on differential privacy: https://www.census.gov/programs-surveys/decennial-census/decade/2020/planning-management/process/disclosure-avoidance.html.
- Census Academy, free videos and courses on using census data: https://www.census.gov/data/academy.html.
- National Archives, the 1950 census records: https://1950census.archives.gov.

8.9.3 Other sources and tools

- IPUMS NHGIS, historical census tables and boundaries: https://www.nhgis.org.
- Social Explorer, census maps and tables back to 1790 (many universities subscribe): https://www.socialexplorer.com.
- censusdis, a Python package for downloading census data with boundaries: https://github.com/censusdis/censusdis.
- Esri, "Living Atlas" ACS layers, ready to map in ArcGIS: https://livingatlas.arcgis.com.
- North Carolina Office of State Budget and Management, State Demographer's population estimates and projections: https://www.osbm.nc.gov/facts-figures/population-demographics.
- Carolina Demography, at UNC's Carolina Population Center, on North Carolina's census data: https://carolinademography.cpc.unc.edu.

9 Geocoding and Georeferencing

Learning objectives

By the end of this chapter you should be able to:

1. Explain what geocoding is, and how address matching turns a street address into a point.
2. Compare the reference files used for address matching, from parcels to ZIP Codes, and how each affects accuracy.
3. Place an address on a street segment by linear interpolation.
4. Judge the quality of a geocoded layer from its match rate and match scores.
5. Explain what georeferencing is, and choose good ground control points.
6. Describe affine and polynomial transformations, and how many control points each needs.
7. Compute residuals and the root mean square error (RMSE), and use them to check a georeferenced image.

Most information doesn't arrive with coordinates. A table of patients has street addresses. A list of restaurant inspections has addresses. A historic map of Chapel Hill is a scan: a grid of pixels that knows its rows and columns, but nothing about where on Earth it belongs. Before any of these can be mapped with other layers, they have to be given a location.

This chapter covers the two most common ways to do that. **Geocoding** turns descriptions of places, most often street addresses, into coordinates. **Georeferencing** ties an image or scanned map to a coordinate system. Both are estimates, and both have errors you can measure. The thread running through the chapter: the reference data you choose, and how carefully you use them, decide how accurate the result is.

9.1 Geocoding and address matching

9.1.1 An everyday example

Every time you type an address into a map app, the app geocodes it. The text "201 E Franklin St, Chapel Hill, NC 27514" has to become a latitude and longitude before the app can put a pin on the map or draw a route. That everyday step is the same one a health department uses to map thousands of disease cases, or a city uses to map every 911 call.

Geocoding is assigning geographic coordinates to data that describe a location in some other way. It is a form of data capture. The most common kind is **address matching**: placing an address at a point or in an area by comparing it with a **reference file** whose features already carry address information. The output is a point layer that keeps all the original attributes, ready to be mapped and analyzed with other layers.

9.1.2 How U.S. addresses are structured

U.S. addresses are **street-based**: a house number, a street name and type, a city, a state, and a ZIP Code. Two conventions make them easy for a GIS to place:

- **House numbers increase along each street**, usually in blocks of 100. The 1100 block of Main Street comes after the 1000 block.
- **Odd numbers are on one side of the street, and even numbers on the other.**

That ordered, ranged numbering is what lets software place an address along a street line. Not every country works this way. Some historic British houses had names rather than numbers, and many villages numbered houses in any order.

Did you know..?

In most of Japan, streets have no names. A traditional Japanese address names a district, then a numbered block, then a building number, and the buildings on a block were often numbered in the order they were built, not their order along the street. Finding a house by its address alone can be hard even for locals, which is why Japanese police boxes keep detailed neighborhood maps for giving directions.

9.1.3 Reference files

Address matching compares each address with a reference file. There are three common kinds:

Table 9.1: The three common kinds of reference file for address matching.

Reference file	Each feature holds	Example
Point or polygon	One discrete address	Land parcels, address points
Point or polygon	A group of addresses	ZIP Code areas, ZIP+4 segments
Line	A range of addresses along a street segment	TIGER/Line roads, local street centerlines

Each kind places an address differently, with a different accuracy:

- **ZIP Code centroids.** Every address in a five-digit ZIP Code is placed at the same point, the ZIP Code's center. It matches almost every address, and quickly, but it is the least precise option: in a large rural ZIP Code, the error can be several kilometers. (Remember that ZIP Codes are mail delivery routes, not areas. The Census Bureau's ZIP Code Tabulation Areas, Chapter 8, approximate them.)
- **ZIP+4 and ZIP+2.** The four extra digits of a ZIP+4 identify a short piece of a delivery route: one side of a block, a single building, or a large mail recipient. The address is placed at the center of that segment. A ZIP+2 (the first two of the four digits) groups several segments.
- **Street centerlines.** Each street segment stores the range of addresses on its left and right sides. The address is placed by **interpolation** along the segment (Section 9.1.4). The Census Bureau's free, nationwide TIGER/Line files make this the most common method in the United States.

- **Parcels.** Each parcel polygon is one property with its own address, so a match finds the exact lot. The point can be placed at the parcel's **centroid**, inside the property rather than on the street. Parcels are the most accurate reference, but counties maintain them, so coverage and quality vary from place to place.
- **Address points.** Many states and counties, including North Carolina, now keep a point for every address, often on the rooftop or the building entrance. Where they exist, address points are as accurate as parcels.

Figure 9.1 shows how the same address lands in four different places, depending on the reference file.

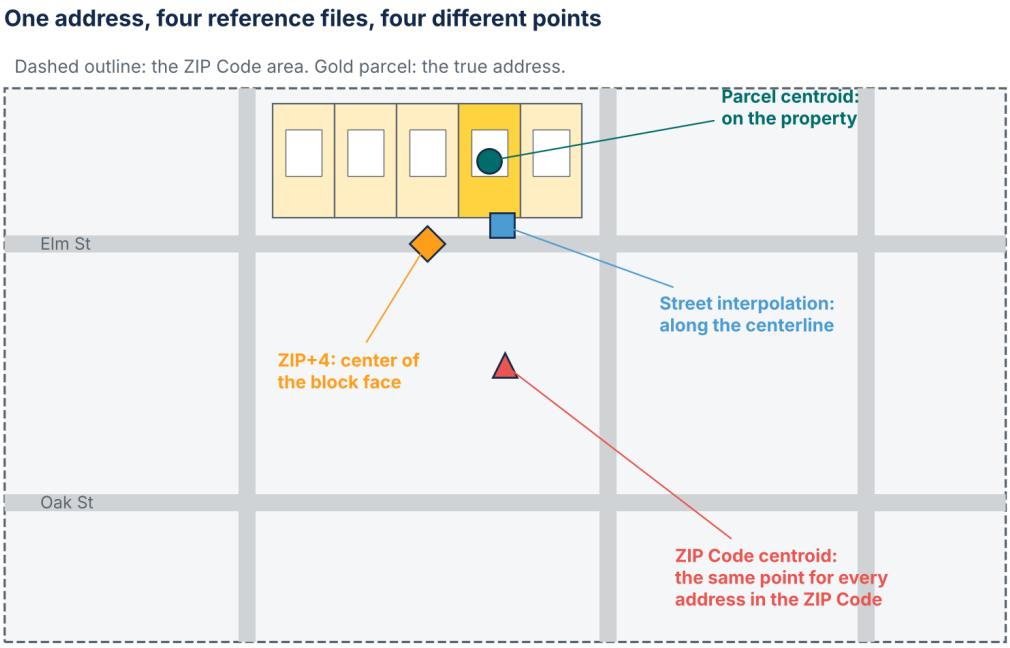


Figure 9.1: One address placed with four reference files: at the ZIP Code's centroid, at the center of its ZIP+4 block face, interpolated along the street centerline, and at the parcel's centroid. The streets and parcels are made up.

The trade-off is summarized in Table 9.2. Moving down the table, matching gets easier and cheaper, and the locations get less precise.

Table 9.2: Accuracy and effort of address-matching reference data.

Reference data	Address is placed	Accuracy	Effort and cost
Parcels or address points	On the property itself	Highest	Highest
Street centerlines	Interpolated along the street	High	High
ZIP+4	Center of a block-face segment	Medium	Medium
ZIP+2	Center of a group of segments	Medium-low	Medium-low
ZIP Code (5 digits)	ZIP Code centroid	Low	Low
ZIP Code (3 digits)	Regional centroid	Lowest	Lowest

9.1.4 Linear interpolation along a street

Street centerline matching takes three steps (Figure 9.2):

1. **Attribute query.** Find the street segment whose name, type, and address range include the address: for 1147 Main St, the segment of Main Street that holds 1100 to 1199.
2. **Choose the side of the street.** The segment stores a range for each side. Here the left side holds 1100 to 1198 (even), and the right side 1101 to 1199 (odd). 1147 is odd, so it is on the right.
3. **Interpolate.** Assume the addresses are spaced evenly along the segment, and find how far along it 1147 falls.

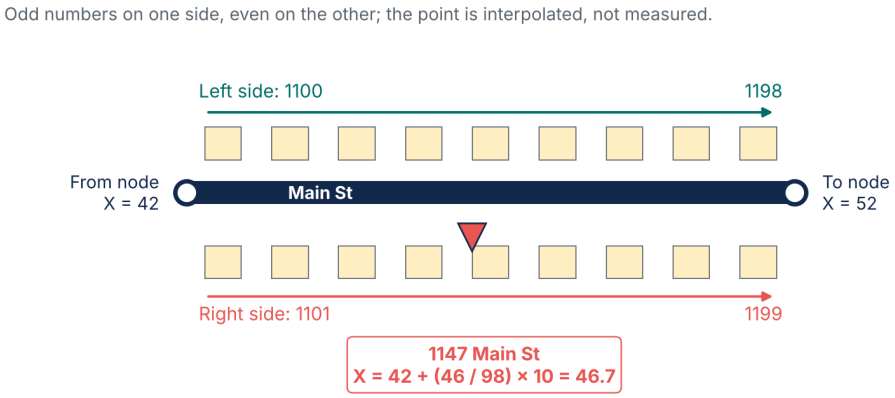


Figure 9.2: Linear interpolation along a TIGER street segment. 1147 is odd, so it is on the right side, 47 percent of the way from 1101 to 1199.

The share of the way along the segment is

$$A\% = \frac{A_{\#} - A_{\text{from}}}{A_{\text{to}} - A_{\text{from}}}$$

where $A_{\#}$ is the target house number and A_{from} and A_{to} are the house numbers at the segment's two ends, on that side of the street. The coordinates are then

$$X_{\#} = X_{\text{from}} + A\% \times (X_{\text{to}} - X_{\text{from}}), \qquad Y_{\#} = Y_{\text{from}} + A\% \times (Y_{\text{to}} - Y_{\text{from}})$$

For 1147 Main St:

- Address difference: 1147 − 1101 = 46.
- Address range: 1199 − 1101 = 98.
- Share along the segment: 46 ÷ 98 = 0.47, or 47 percent.
- Change in X: 52 − 42 = 10.
- Interpolated X: 42 + (0.47 × 10) = 46.7.

Y is found the same way. Software then moves the point a small **offset** distance (often 5 to 10 meters) to the correct side of the centerline, so that houses on opposite sides of the street don't land on top of each other.

Interpolation is an estimate. It assumes the lots are equal in size, that every number in the range is used, and that the street is straight. A long rural driveway, a big lot at one end of a

block, or a range of 1100 to 1199 with only six houses can each put the point tens of meters from the building.

> **Think about it**
>
> Where is 1184 Main St, on the same segment? Which side of the street is it on, what is A%, and what is its interpolated X? (Answer: the left side, since 1184 is even; A% = 84 ÷ 98 = 0.86; X = 42 + 8.6 = 50.6.)

9.1.5 Geocoding in practice

Individual and batch mode. Online tools geocode one address at a time. The **Census Geocoder** (geocoding.geo.census.gov) returns the coordinates of an address and the census tract, block group, and block it is in, and it can match a batch of up to 10,000 addresses from a CSV. GIS software geocodes a whole table at once in **batch mode**: ArcGIS Pro's *Geocode Table* tool, or plugins and processing tools in QGIS such as MMQGIS.

Standardization. Before matching, the software parses each address into parts (number, prefix, name, type, suffix, unit) and standardizes them, so that "201 East Franklin Street" and "201 E Franklin St" are treated alike.

Match scores. Each candidate gets a **match score**, often from 0 to 100, measuring how closely the address agrees with the reference. Spelling differences ("Fraklin"), a missing street type, or a wrong ZIP Code lower the score. The software accepts the best candidate above a minimum score, and reports each address as:

- **Matched**: one good candidate.
- **Tied**: two or more candidates with the same score, such as "Main St" in two towns.
- **Unmatched**: no candidate good enough.

Match rate. The **match rate** is the share of addresses that were matched. A 70 percent match rate means three addresses in ten are missing from your map, and they are rarely missing at random: rural route and PO Box addresses, new subdivisions not yet in the reference file, and addresses typed carelessly all fail more often. Review unmatched and tied addresses by hand (ArcGIS Pro's *Rematch Addresses* pane does this), and always report the match rate with your results.

> **Think about it**
>
> A health department geocodes the home addresses of people with a disease, and posts a map of the points online. What could go wrong? How might the department show the pattern while protecting people's privacy? (Look up *geomasking* and *aggregation*.)

9.2 Georeferencing

9.2.1 Why georeference?

A scanned map, an aerial photo, or a raw satellite image is a raster that knows only its own **file coordinates**: pixel columns and rows, counted from the top-left corner. Its features

don't sit at their true positions on the Earth. To overlay it on other layers, measure on it, or digitize from it, you have to give it map coordinates (Figure 9.3).

Georeferencing transforms a raster from file coordinates to map coordinates in a coordinate system and projection. It is a kind of **image rectification**. Common uses include:

- Overlaying historic maps on modern data, to see how a city, coastline, or forest has changed.
- Digitizing features from a scanned paper map, such as old property lines or the boundaries of redlined neighborhoods.
- Aligning a drone image or a scanned air photo with other layers.

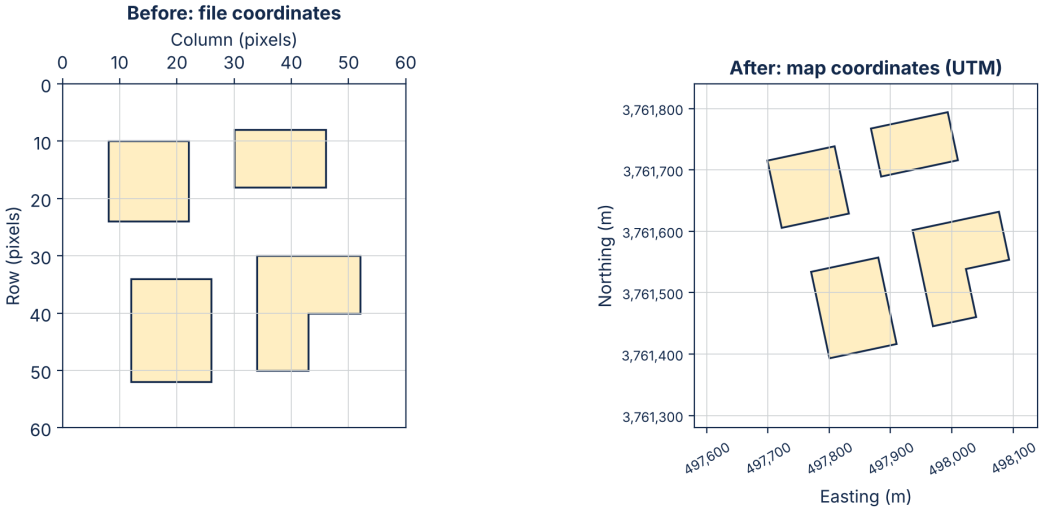


Figure 9.3: Building outlines on a scanned image in file coordinates (left), and the same outlines after georeferencing, in UTM coordinates (right). The buildings are made up.

9.2.2 Five steps

Georeferencing follows five steps:

1. **Get to know the data.** What does the image show, what is its scale and date, and what reference layers cover the same area?
2. **Set up ground control points (GCPs)**: features you can find on both the image and a reference layer.
3. **Transform the raster.** This has two parts. A **coordinate transformation** (spatial interpolation) works out where each cell belongs on the map. **Resampling** (intensity interpolation) then gives each output cell a value.
4. **Validate** the result with the residuals and the root mean square error (RMSE).
5. **Rectify**: save the georeferenced raster.

9.2.3 Ground control points

Ground control points (GCPs) are features that can be identified on the image and located accurately on a reference layer, such as a basemap, an orthophoto, or a road layer. Good GCPs are:

- **Easy to identify** on both: road intersections, building corners, bridges, and dams.

- **Spatially accurate** in the reference data.
- **Evenly distributed** across the whole image, including near the corners and edges. GCPs bunched in one area leave the rest of the image unconstrained.
- **Stable.** Avoid features that move or change: shorelines, river channels, vehicles, shadows, and trees. On a historic map, a church or a road junction that still exists is better than a field boundary.

9.2.4 Transformations

A **transformation** is a set of equations that converts each image position (x', y') into map coordinates (x, y). Its coefficients are solved from the GCPs, usually by **least squares**, which finds the coefficients that make the errors at the GCPs as small as possible overall.

First-order (affine) transformation. For moderate distortion over a small area, a first-order polynomial with six coefficients is enough (Jensen 2015):

$$x = a_0 + a_1 x' + a_2 y', \qquad y = b_0 + b_1 x' + b_2 y'$$

The six coefficients model the distortions in Figure 9.4: a shift in x and y (**translation**), a change of **scale** in x and in y, **skew**, and **rotation**. An affine transformation keeps straight lines straight and parallel lines parallel. It needs at least **3 GCPs**.

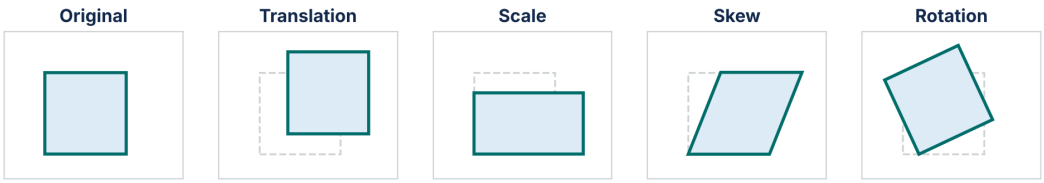


Figure 9.4: The distortions an affine transformation corrects. Dashed: the original square.

Higher-order polynomials. Nonlinear distortions, such as those from terrain relief, lens curvature, or a paper map that has shrunk unevenly, need a higher-order polynomial, which lets the image bend (Figure 9.5). Each order needs more GCPs:

Table 9.3: Polynomial transformations and the minimum number of GCPs.

Order	Can model	Minimum GCPs
1st (affine)	Shift, scale, skew, rotation	3
2nd	Simple curves	6
3rd	More complex bends	10

Two rules follow:

- **Collect more than the minimum.** With exactly 3 GCPs, an affine transformation fits them perfectly, and you have no way to measure the error. Ten or more well-spread GCPs are a good habit.
- **Use the lowest order that fits.** A high-order polynomial bends to fit the GCPs, but it can warp badly in areas between and beyond them, as the right edge of the third panel in Figure 9.5 shows. Try first order, and move up only if the error is too large.



Figure 9.5: Fitting the same GCPs with 1st-, 2nd-, and 3rd-order polynomials. Higher orders bend to fit the curve. The distortion and GCPs are made up.

Software offers other transformations too. A **Helmert** (similarity) transformation allows only shift, rotation, and uniform scale. A **spline** (thin plate spline or rubber sheeting) fits every GCP exactly and stretches the image between them; it is useful for badly distorted historic maps, but it hides errors at the GCPs, because their residuals are all zero.

9.2.5 Resampling

After the transformation, the output grid doesn't line up with the input pixels. Each output cell is traced back to its position in the input image, and given a value by **resampling**:

- **Nearest neighbor** takes the value of the closest input cell. It keeps the original values, so use it for categorical rasters such as land cover.
- **Bilinear interpolation** averages the four closest cells, weighted by distance. It gives a smoother look for continuous images such as air photos and scanned maps.
- **Cubic convolution** uses the 16 closest cells. It is sharper than bilinear, but slower.

9.2.6 Residuals and RMSE

How good is the fit? After the transformation is solved, it is applied to each GCP's image position, and the result is compared with where the GCP really is on the reference layer. The distance between the two is the GCP's **residual** (Figure 9.6):

$$R_i = \sqrt{(x_i - X_i)^2 + (y_i - Y_i)^2}$$

where (X_i, Y_i) are the reference coordinates of GCP i, and (x_i, y_i) are the coordinates the transformation gives it. A residual much larger than the others flags a GCP that may be misplaced, such as a click on the wrong intersection.

The **root mean square error (RMSE)** combines all the residuals into one number, in map units:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} R_i^2}$$

Square each residual, average the squares, and take the square root. For four GCPs with residuals of 56.81, 92.94, 0.96, and 35.17 meters:

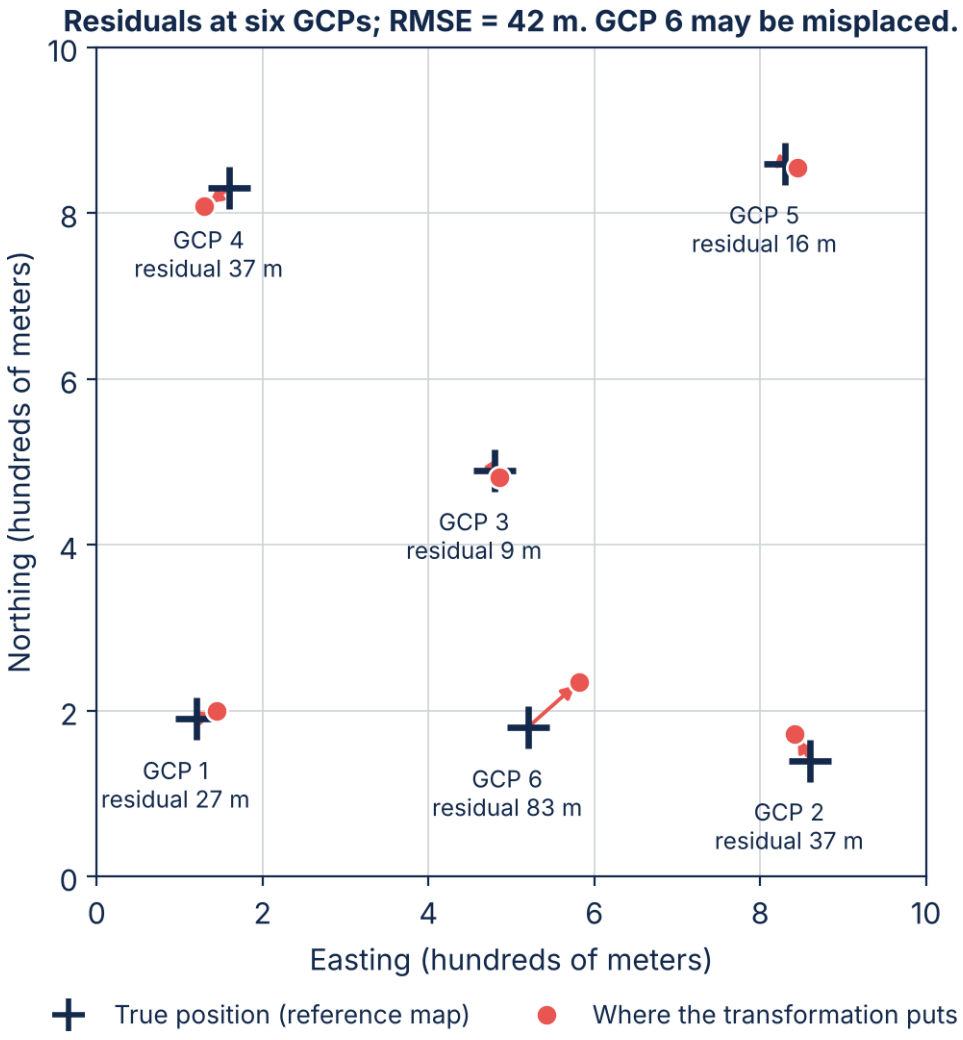


Figure 9.6: Residuals at six ground control points. Each arrow runs from a GCP's true position to where the transformation puts it. GCP 6's residual is more than twice the others, which suggests it was placed on the wrong feature. The GCPs are made up.

Table 9.4: Computing the RMSE from four residuals.

GCP	Residual (R_i)	R_i^2
1	56.81	3,227.38
2	92.94	8,637.84
3	0.96	0.92
4	35.17	1,236.93
Mean of R_i^2		3,275.77
RMSE		**57.23**

Lower is better. A common target is an RMSE smaller than the cell size of the raster: for an air photo with 1-meter cells, under 1 meter. For a historic map, the target depends on the map's scale and condition, and a larger RMSE may be the best that is possible. Either way, **report the RMSE and the transformation order with the result**, so users know how far to trust it.

The RMSE measures the fit at the GCPs only. A low RMSE from a few bunched GCPs, or from a high-order polynomial, can hide large errors elsewhere. The best check is to hold back a few **check points** that are not used to fit the transformation, and measure their errors too.

Did you know..?

Between 1867 and the 1960s, the Sanborn Map Company drew detailed fire insurance maps of about 12,000 American cities and towns, showing the outline, building material, and number of stories of almost every building. The Library of Congress has scanned tens of thousands of these sheets, including many of North Carolina towns, and historians and planners georeference them to trace how neighborhoods have changed.

9.2.7 Georeferencing in ArcGIS Pro

1. **Open the georeference tools.** Add the scanned map to a map that already has reference layers, such as a basemap or orthophoto. Select the image in the Contents pane, and on the *Imagery* tab, click *Georeference*.
2. **Bring the image into view.** Zoom the map to the area the scan covers, and click *Fit to Display* on the *Georeference* tab. The image is stretched over the current extent, a rough starting position.
3. **Add control points.** Click *Add Control Points*. Click a feature on the scan, then the same feature on the reference layer. Repeat across the whole image.
4. **Check the control point table.** Open the *Control Point Table* to see each GCP's residual and the total RMS error. Re-place or delete points with large residuals, and recheck. Change the transformation (*1st Order Polynomial* by default) only if the fit isn't good enough.
5. **Save or export.** *Save* stores the transformation with the original raster in an auxiliary file, without changing its pixels. *Save as New* writes a new, rectified raster: choose the resampling method (nearest neighbor for categories, bilinear or cubic for images) and an output cell size close to the source's. *Export Control Points* keeps a text file of the GCPs, so the work can be checked or repeated.

9.2.8 Georeferencing in QGIS

QGIS has a built-in **Georeferencer** (*Layer > Georeferencer*). Open the raster in the Georeferencer window, and click *Add Point* to place GCPs: click a feature on the image, then click *From Map Canvas* to pick its location on the main map. Under *Transformation Settings*, choose the transformation type (Linear, Helmert, Polynomial 1 to 3, Thin Plate Spline, or Projective), the target CRS, the resampling method, and the output file. The GCP table shows each point's residual, and the status bar shows the mean error. *Start Georeferencing* writes the rectified raster, and the GCPs can be saved to a `.points` file.

9.3 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

9.4 Summary

- **Geocoding** assigns coordinates to data that describe a location in another way. **Address matching** places an address by comparing it with a **reference file**.
- U.S. addresses can be interpolated because house numbers increase along each street, with odd and even numbers on opposite sides.
- Reference files range from **parcels** and address points (most accurate) through **street centerlines** and **ZIP+4** to **ZIP Code centroids** (least accurate). The same address lands in different places with each.
- **Linear interpolation** places an address along a street segment in proportion to its number within the segment's range: $A\% = (A_{\#} - A_{\text{from}})/(A_{\text{to}} - A_{\text{from}})$.
- Batch geocoding standardizes addresses and scores each candidate. Report the **match rate**, and review tied and unmatched addresses, which are rarely missing at random.
- **Georeferencing** transforms a raster from file coordinates (rows and columns) to map coordinates, using **ground control points** that are easy to identify, accurate, stable, and spread across the image.
- A **first-order (affine)** transformation corrects shift, scale, skew, and rotation, and needs at least 3 GCPs. **Second-** and **third-order** polynomials bend to fit nonlinear distortion and need 6 and 10. Use the lowest order that fits.
- **Resampling** gives the output cells their values: nearest neighbor for categories, bilinear or cubic for images.
- A GCP's **residual** is the distance between where the transformation puts it and where it really is. The **RMSE** combines the residuals into one number. Check it, fix outliers, and report it with the result.

9.5 Exercises

1. **Interpolate.** A street segment runs from (100, 200) to (300, 260). Its left side holds 200 to 298 and its right side 201 to 299. Find the coordinates of 250 and 275, and say which side each is on.
2. **Compare reference files.** Use the Census Geocoder to geocode your department's address, or any public building. Then find the same building on a parcel map from NC OneMap or the county GIS website. How far apart are the two points?

3. **Batch geocode.** Make a CSV of 20 public addresses in Chapel Hill (schools, libraries, fire stations). Geocode them in batch with the Census Geocoder or in ArcGIS Pro. What is the match rate? Why did any addresses fail?
4. **Georeference a historic map.** Download a historic map of Chapel Hill or another North Carolina town from the Library of Congress or the North Carolina Collection's digital maps. Georeference it in QGIS or ArcGIS Pro with at least 8 GCPs and a first-order transformation. Record the RMSE. Then try a second-order transformation. Did the RMSE fall? Does the image look better or worse away from the GCPs?
5. **Find the bad GCP.** In exercise 4, deliberately place one GCP on the wrong intersection. What happens to its residual and to the RMSE? Delete it and compare.
6. **RMSE by hand.** Five GCPs have residuals of 2.1, 3.4, 1.8, 2.6, and 9.7 meters. Compute the RMSE. Compute it again without the fifth point. What would you do next?

9.6 Further reading and online resources

9.6.1 Books and guides

- *GIS Fundamentals: A First Text on Geographic Information Systems*, by Paul Bolstad (Bolstad 2022). Chapter 4 covers registration, control points, transformations, and RMSE.
- *Introductory Digital Image Processing: A Remote Sensing Perspective*, by John R. Jensen (Jensen 2015), the standard source on geometric correction, polynomial transformations, and resampling.
- Goldberg, D. W., *A Geocoding Best Practices Guide* (Goldberg 2008), written for cancer registries, and the most thorough guide to geocoding quality: https://www.naaccr.org/gis-resources/.
- *The Nature of Geographic Information*, by David DiBiase and colleagues at Penn State, on address matching and the TIGER files: https://www.e-education.psu.edu/natureofgeoinfo/.
- The GIS&T Body of Knowledge, "Geocoding" and "Georeferencing": https://www.ucgis.org/site/gis-t-body-of-knowledge.

9.6.2 Geocoding tools and data

- U.S. Census Bureau, Census Geocoder: https://geocoding.geo.census.gov.
- U.S. Census Bureau, TIGER/Line shapefiles, including address ranges: https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html.
- ArcGIS Pro, "Geocode Table" and "Rematch Addresses": https://pro.arcgis.com/en/pro-app/latest/help/data/geocoding/what-is-geocoding-.htm.
- QGIS, the MMQGIS plugin, with geocoding tools: https://plugins.qgis.org/plugins/mmqgis/.
- NC OneMap, North Carolina's address points and parcels: https://www.nconemap.gov.
- USPS, "ZIP Code — The Basics," on what ZIP and ZIP+4 codes are: https://faq.usps.com.
- Nominatim, the geocoder for OpenStreetMap data: https://nominatim.org.

9.6.3 Georeferencing tools and historic maps

- ArcGIS Pro, "Overview of georeferencing": https://pro.arcgis.com/en/pro-app/latest/help/data/imagery/overview-of-georeferencing.htm.



- QGIS User Guide, "Georeferencer": https://docs.qgis.org/latest/en/docs/user_manual/managing_data_source/georeferencer.html.
- QGIS Tutorials and Tips, "Georeferencing Topo Sheets and Scanned Maps," by Ujaval Gandhi: https://www.qgistutorials.com/en/docs/3/georeferencing_basics.html.
- Library of Congress, Sanborn Maps collection: https://www.loc.gov/collections/sanborn-maps/.
- Library of Congress, Geography and Map Division, map collections: https://www.loc.gov/maps/.
- David Rumsey Map Collection, with an online georeferencer: https://www.davidrumsey.com.
- North Carolina Maps, from UNC Libraries and partners, with historic maps of the state: https://web.lib.unc.edu/nc-maps/.
- Allmaps, open-source tools for georeferencing maps from library collections: https://allmaps.org.
- Mapping Inequality, georeferenced redlining maps of U.S. cities, including Durham, Greensboro, and Winston-Salem: https://dsl.richmond.edu/panorama/redlining/.

10 OpenStreetMap and Data Quality

Learning objectives

By the end of this chapter you should be able to:

1. Define spatial data quality as fitness for use, and name its seven elements.
2. Explain what volunteered geographic information (VGI) is, why it appeared in the 2000s, and how its main types differ.
3. Describe Big Data by its three Vs, and the FAIR principles for good data.
4. Explain how OpenStreetMap is organized: its license, its users, and its data model of nodes, ways, relations, and tags.
5. Compare extrinsic and intrinsic ways of measuring the quality of OpenStreetMap, and name the tools the community uses to check it.
6. Use the 2015 Nepal earthquake to explain how crisis mapping works, and how a flood of new volunteers affects data quality.

Chapter 8 divided data about people and places into two kinds: **authoritative data**, made by official agencies to documented standards, and **asserted data**, contributed by the public. This chapter is about the second kind, and its best-known example, OpenStreetMap.

OpenStreetMap is a free map of the world that anyone can edit. Every street, building, and footpath on the UNC campus in OpenStreetMap was added by a volunteer, some walking with a GPS, most tracing satellite imagery from a desk. The result is used by governments, the largest technology companies, and aid agencies. So how good is it? To answer that, we first need a way to describe the quality of any geographic dataset.

10.1 Defining data quality

10.1.1 Quality is fitness for use

There is no such thing as a perfect dataset. Every map leaves things out, generalizes shapes, and goes out of date. **Spatial data quality** is how well a dataset fits the use you have in mind. A road layer that is fine for a tourist map may be dangerous for routing ambulances. A building layer drawn from 2015 imagery may be good enough to estimate a city's population, and useless for finding houses built since.

Quality questions come from both sides of a dataset:

- As a **data consumer**, how do you judge the quality of data you download?
- As a **data creator**, how do you make data good, and tell others how good it is?

Many things can go wrong. A road may not exist, or may not connect to the road it crosses. An address may be in the wrong place. A building may be labeled a school after it closed. A layer may come with no metadata at all, or in a coordinate system you didn't expect (Chapter 4).

10.1.2 Seven elements of quality

Van Oort (2006) compared the quality elements used in mapping standards around the world, and found seven that they share (Figure 10.1). They are now written into the international standard ISO 19157, *Geographic information: Data quality*.

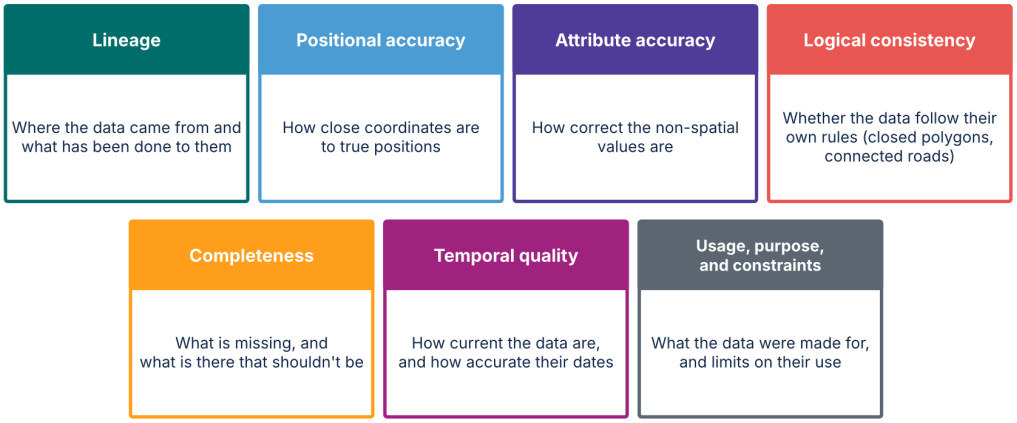


Figure 10.1: The seven elements of spatial data quality, after van Oort (2006).

Take a layer of fire hydrants as an example:

- **Lineage** is the history of the dataset. Who collected the hydrants, with what equipment, and how were the points processed and edited since?
- **Positional accuracy** is how close the coordinates are to the true locations. Is the hydrant where the map says it is, or across the street?
- **Attribute accuracy** is how correct the non-spatial values are. Is the feature labeled a hydrant, with the right flow rate?
- **Logical consistency** is whether the data agree with their own rules. Are hydrants on the water mains they connect to? In a road layer, do roads meet at intersections?
- **Completeness** is "a measure of the absence of data and the presence of excess data" (Oort 2006, 23). Are all the hydrants there, and are there any that were removed long ago?
- **Temporal quality** is whether the dates are right, and the data current. When was each hydrant last checked?
- **Usage, purpose, and constraints** tell users what the data were made for, and what they should not be used for.

The first and last elements are not measurements of the data at all. They are information *about* the data, which is why metadata matter so much. A dataset with poor positional accuracy can still be useful if its metadata tell you how poor. A dataset with no metadata leaves you guessing.

Think about it

Which of the seven elements matters most for each of these uses: (a) routing an ambulance; (b) estimating the number of people in a flood zone from building footprints; (c) a historic map of the town in 1950? Is the answer the same for all three?



10.2 Volunteered geographic information

10.2.1 Anyone can be a data producer

Until the 2000s, almost all map data came from governments and a few companies, made by trained staff to written standards. Three developments changed that:

- **User-generated content.** "Web 2.0" sites let readers create content, not just read it: blogs, wikis, photo sharing, and social networks (O'Reilly 2005).
- **Cheap GPS.** Handheld receivers, and then phones, made it cheap and easy to record a location.
- **Mobile computing.** Fast internet access everywhere let people upload what they recorded, from wherever they were.

Together they produced what Goodchild (2007) called **volunteered geographic information (VGI)**: geographic data created by individuals, most of them not trained in mapping, and shared on the internet. Goodchild described the volunteers as "citizens as sensors," billions of people able to observe and report on the places around them.

VGI changed the quality question. When a government agency makes a map, you can judge it partly by the agency's reputation and standards. When anyone can contribute, quality can no longer be assumed from who made the data. It has to be measured.

10.2.2 Types of VGI

VGI comes in three main forms, and the form affects its quality:

Table 10.1: The three main types of VGI.

Type	Examples	How location is recorded	Quality
Location-based social media	Geotagged posts and photos, check-ins	As a by-product: people are not trying to make a map	Very uneven; location may be the phone's, not the subject's
Collaborative mapping	OpenStreetMap, Wikimapia	Deliberately: mapping is the goal	Contributors can see and correct each other's work
Citizen science	eBird, iNaturalist, CoCoRaHS rain gauges, reports of flooded road crossings	By following a protocol set by scientists	Quality control is built into the protocol

Research on VGI has followed several strands. Some studies ask why people volunteer, and who they are. Others measure data quality, as we do below. Others study bias: OpenStreetMap contributors have been mostly young, male, urban, and from Europe and North America, and their interests shape what gets mapped and where. Others study VGI in disasters, the subject of the last part of this chapter.

10.2.3 Big Data

VGI is one source of **Big Data**, datasets too large, too fast, or too varied for traditional tools. Big Data is usually defined by three Vs (Beyer and Laney 2012):

- **Volume**: from terabytes to petabytes, more than one computer can store or process.
- **Velocity**: data arrive continuously and must be handled in near real time.
- **Variety**: mixed formats, such as text, images, GPS traces, and sensor readings.

OpenStreetMap has all three: billions of nodes, edits arriving every minute, and everything from roads to opening hours. Most of the data from phones, vehicles, and sensors carries a location, so geographic data make up a large share of Big Data.

Some authors add a fourth V, **veracity**: can the data be trusted? That is this chapter's question. At this scale, no one can check every feature by hand, so quality checks have to be automated and statistical. More data does not mean better data.

10.2.4 FAIR data

The **FAIR principles** (Wilkinson et al. 2016) say that good scientific data should be:

- **Findable**: data and metadata are easy to find, by people and by computers.
- **Accessible**: once found, users know how to get the data, including any login or permission needed.
- **Interoperable**: the data can be combined with other data, software, and workflows.
- **Reusable**: data and metadata are described well enough to be reused, or combined in new settings.

Many funders and journals now require FAIR data. FAIR doesn't guarantee that data are accurate, but good metadata let users judge quality for themselves. OpenStreetMap does well on findable, accessible, and reusable: it has an open license and a public API. It is weaker on consistent metadata, because contributors set their own tags.

10.3 OpenStreetMap

10.3.1 A free, editable map of the world

OpenStreetMap (OSM) was founded in 2004 by Steve Coast, then a student in London, to create free, open road data. It now maps far more than streets: buildings, land use, trails, shops, water points, and much more.

- **Anyone can edit.** You can add, change, or delete data, and your edits go live immediately. Each set of edits is saved as a **changeset**, with the editor's username and a comment.
- **The data are open.** OSM is published under the **Open Database License (ODbL)**. You can use, change, and share the data for any purpose, including commercial use, as long as you credit "© OpenStreetMap contributors" and share any database you build from it under the same license.
- **It is big.** OSM has more than 10 million registered accounts, and roughly 10 billion nodes. Only a small share of accounts have ever edited, and a much smaller core does most of the work.



Did you know..?

OpenStreetMap began because map data in Britain were expensive. The Ordnance Survey, the national mapping agency, charged for its data, so Steve Coast set out to build a free map from GPS traces he and his friends recorded on foot, by bicycle, and by car. Some of the first OSM mapping parties, in 2006, sent volunteers around a town with GPS receivers to map it in a weekend.

10.3.2 Who uses OpenStreetMap?

OSM is no longer a hobby project:

- **Governments.** The U.S. Census Bureau has compared OSM with its TIGER/Line road files (Chapter 8). Many national and local governments use or contribute to OSM.
- **Technology companies.** Apple, Meta, Microsoft, and Amazon all use OSM data. In 2022, Amazon, Meta, Microsoft, and TomTom founded the **Overture Maps Foundation**, which builds open map data partly on OSM.
- **Navigation and logistics.** Routing engines, delivery, and ride-hailing apps.
- **Humanitarian organizations.** The Red Cross and Médecins Sans Frontières use OSM to plan disaster response and health campaigns.
- **Researchers and students.** OSM is free, global, and consistent enough for spatial analysis anywhere.

There is also a public-good argument. Whoever makes the map decides which places exist on it. An open map, which anyone can improve, keeps that power from resting with one or two companies (Wroclawski 2014).

10.3.3 OSM's data model

OSM's data model is simple. It has three kinds of element, plus tags (Figure 10.2):

- A **node** is a single point, with a latitude and longitude. A node can stand alone, as a café or a hydrant, or be one of the vertices of a way.
- A **way** is an ordered list of nodes, forming a line such as a road or a stream. A **closed way**, whose first and last nodes are the same, forms an area, such as a building or a park.
- A **relation** groups elements and describes how they relate, such as all the roads of a bus route, or the outer and inner rings of a lake with islands.
- A **tag** is a `key=value` pair that describes an element, such as `highway=residential` or `name=Franklin Street`. Any element can carry any number of tags.

All of the meaning is in the tags, and there is no fixed list of them. Anyone can invent a tag, and the community agrees on conventions and documents them on the OSM Wiki. That flexibility lets OSM map almost anything. It also creates inconsistency, a logical consistency problem in van Oort's terms: one mapper tags a path `highway=footway`, another `highway=path`, a third `highway=track`. Because the data structure is loose, the community has to define and enforce its own quality standards.

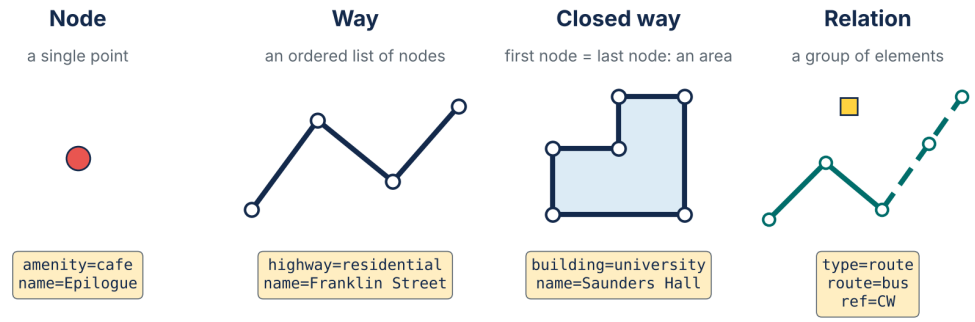


Figure 10.2: The elements of OpenStreetMap's data model. The places and tags are examples.

10.4 Measuring the quality of OSM

10.4.1 Extrinsic and intrinsic measures

There are two ways to measure the quality of OSM (Figure 10.3):

- **Extrinsic** measures compare OSM with an **authoritative** reference dataset, such as the Ordnance Survey's maps in Britain or TIGER/Line in the United States. You can measure how far OSM roads are from the reference roads (positional accuracy), and what each dataset has that the other lacks (completeness).
- **Intrinsic** measures use only OSM's own history: how many people edited an area, how many versions each feature has, how recently it was edited, and whether the work was checked by a second mapper.



Figure 10.3: Extrinsic measures compare OSM with an authoritative reference; intrinsic measures use OSM's own editing history. The values on the right are examples.

The early studies were extrinsic. Haklay (2010) compared OSM's roads in England with the Ordnance Survey's. He found them "fairly accurate": OSM's roads were, on average, within about 6 meters of the Ordnance Survey's, and about 80 percent of the motorways overlapped. Completeness was uneven, though. Wealthy and central areas were well mapped, while poorer and rural areas lagged. Later studies found the same in other countries, and found high completeness for building footprints in Munich (Fan et al. 2014).

Extrinsic measures have a limit: they need a trusted reference, and in the places where OSM matters most, there often isn't one. That is why intrinsic measures are useful. Haklay and colleagues (2010) found that the more people had mapped an area, the more accurate its roads were. This is unique to collaborative data. A traditional dataset has one producer, so there is no crowd to count.

Did you know..?

In open-source software, **Linus's Law** says that "given enough eyeballs, all bugs are shallow." It was named for Linus Torvalds, who created Linux. Haklay and colleagues (2010) tested it on maps. In London, the positional accuracy of OSM roads improved quickly as the first contributors mapped an area, and once about 15 contributors had mapped a square kilometer, the roads were reliably accurate.

10.4.2 Tools for checking OSM

Quality control in OSM happens after edits are made, through software and people:

- **Osmose** and **KeepRight** run automated checks across the database, flagging problems such as roads that don't connect, buildings that overlap, and missing or conflicting tags.
- **OSMCha** (OSM Changeset Analyzer) lets experienced mappers review recent changesets, to catch vandalism and mistakes.
- **MapRoulette** turns lists of detected errors into small fix-it tasks that anyone can do in a few minutes.
- The **HOT Tasking Manager** divides a mapping project into small squares. When one mapper finishes a square, a second, experienced mapper **validates** it.
- The **ohsome** dashboard, from the Heidelberg Institute for Geoinformation Technology (HeiGIT), shows how much was mapped in an area over time, an intrinsic quality signal.

Your GIS also checks data quality. ArcGIS Pro and QGIS can validate geometry and find features that break topology rules (QGIS's *Check Validity* and *Topology Checker*, ArcGIS Pro's *Check Geometry* and topology rules).

Think about it

Open OpenStreetMap at openstreetmap.org and zoom to a place you know well, such as your hometown or your street. What is mapped, and what is missing? Which of van Oort's quality elements would you worry about if you used it for a project there?

10.5 Crisis mapping

10.5.1 Haiti, 2010

On 12 January 2010, a magnitude 7.0 earthquake struck near Port-au-Prince, Haiti. Before the earthquake, Port-au-Prince was nearly blank in OpenStreetMap. Within days, satellite companies released new imagery, and hundreds of volunteers around the world began tracing roads, buildings, and camps for displaced people. Within weeks, OSM had the most detailed map of the city available anywhere, and aid workers on the ground used it (Zook et

al. 2010). It was the first large test of **crisis mapping**, mapping by volunteers to support disaster response.

10.5.2 The Humanitarian OpenStreetMap Team

A small group of OSM mappers had formed the **Humanitarian OpenStreetMap Team (HOT)** in 2009. The Haiti response turned it into a formal organization in 2010. HOT is activated when a disaster or other humanitarian need is declared, and it works with partners such as the Red Cross and Médecins Sans Frontières. It has an elected board, working groups for activation, communications, training, technology, and more, and local mapping communities in the affected countries.

HOT's work happens in three steps (Figure 10.4):

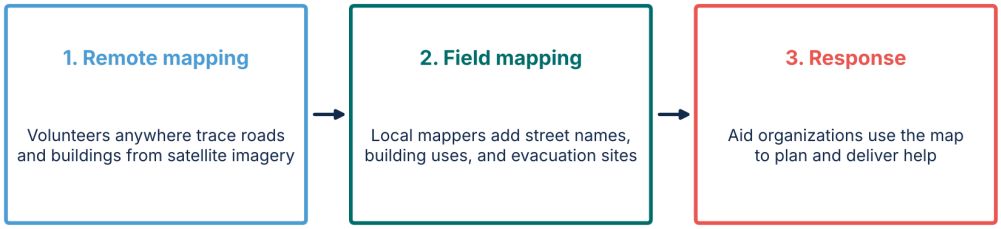


Figure 10.4: The three steps of a HOT mapping project, after LearnOSM.

1. **Remote mapping.** Volunteers anywhere in the world trace roads and buildings from satellite imagery, using the Tasking Manager so that they don't overlap. This step is fast and can involve thousands of people. But remote mappers can't see street names, or what a building is used for.
2. **Field mapping.** Local mappers, who know the place, add names, building uses, and evacuation sites, and check the remote work on the ground. This step is also quality control.
3. **Response.** Aid organizations use the map to plan routes, count buildings, estimate populations, and deliver help.

Did you know..?

You don't need a disaster to map for a cause. **Missing Maps**, founded in 2014 by HOT, the American and British Red Cross, and Médecins Sans Frontières, maps the places most at risk *before* disasters happen. Many of its volunteers map at **mapathons**, events where people trace imagery together for a few hours, often on college campuses.

10.5.3 Nepal, 2015

On 25 April 2015, a magnitude 7.8 earthquake struck central Nepal, followed by a magnitude 7.3 aftershock on 12 May. Nearly 9,000 people died, and more than half a million homes were destroyed or damaged. Many of the hardest-hit districts were rural and mountainous, hard to reach, and poorly mapped. Responders needed up-to-date maps of roads, villages, and helicopter landing sites within days.

HOT activated within hours. **Kathmandu Living Labs**, a civic technology group in Kathmandu, became the local hub, linking remote volunteers with responders in Nepal. HOT called for new volunteers, mapathons, and donations, and the call worked: universities,

companies, and community groups around the world held mapathons, and thousands of people created OSM accounts for the first time. Daily edits in Nepal jumped about tenfold after the earthquake, and volunteers edited roads across the central districts in under three weeks.

Parr (2015) counted the contributors who had mapped in Nepal by the year they joined OSM (Figure 10.5). In the ten years before the earthquake, 8,725 people had mapped there. In the 12 days after it, 9,989 more joined. Newcomers suddenly made up more than half of everyone who had ever mapped the country.

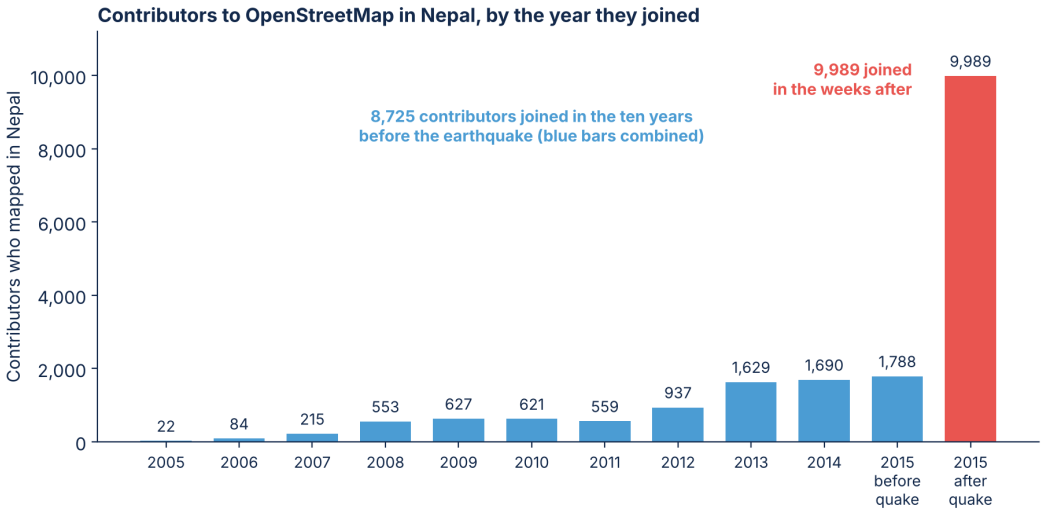


Figure 10.5: Contributors to OpenStreetMap in Nepal, by the year they joined. Data from Parr (2015).

10.5.4 The problem of success

The flood of new volunteers created tension within HOT. On its public mailing list, one experienced member wrote in April 2015 that they were "finding totally inexperienced mappers (some with no completed tiles and others with only one or two completed tiles) messing around with validating tiles or unlocking already validated tiles and working on them," at a time when "people's lives are at stake." Another wrote in May: "We need more people. At the same time, we need to adapt in various ways to crowdsourcing as we see various problems arising."

That is the dilemma of open data in a crisis. The response needed more mappers, not fewer, but beginners make mistakes, and a mistake in a crisis map can send a truck down a road that doesn't exist.

The newcomers' work also looked different (Figure 10.6). On average, each newcomer edited about half as many features as an earlier contributor, and changed or deleted far fewer existing ones. Newcomers mostly added new features. Experienced mappers did most of the correcting, and a small core of them did a very large share of it. The quality of the map depended on that core.

HOT's answer was both social and technical:

- **Social.** HOT improved its training guides at LearnOSM, and asked experienced mappers to review newcomers' work.

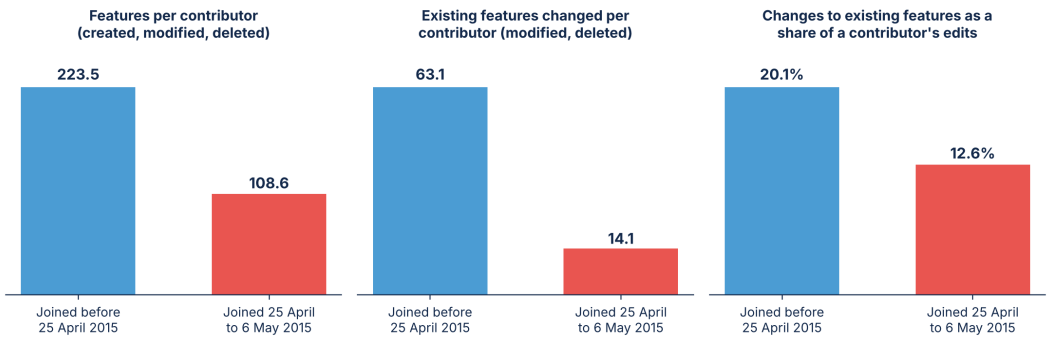


Figure 10.6: How contributors who joined before the earthquake compare with those who joined in the 12 days after it. Data from Parr (2015).

- **Technical.** The Tasking Manager gained a separate validation step, and flags for experienced mappers. A square is marked done only after a second, experienced mapper checks it.

The result was a new layer of data reviewers, the **validators**. It is a familiar pattern for quality control in open projects: open contribution, followed by expert review. It keeps the door open to beginners while protecting the data that responders rely on.

10.6 Getting involved

OpenStreetMap is one of the few datasets in this book that you can improve yourself:

- **Use the data.** Download extracts for a country, state, or region from Geofabrik (download.geofabrik.de) as shapefiles or in OSM's own formats. In QGIS, the **QuickOSM** plugin downloads features for the map's extent by tag, such as every `amenity=hospital` in a county.
- **Join OSM.** Create a free account at openstreetmap.org.
- **Start editing.** The **iD** editor runs in your web browser and needs no installation; it includes a short walkthrough. **JOSM** is a desktop editor for advanced work.
- **Map for a cause.** Pick a project in the HOT Tasking Manager (tasks.hotosm.org), or join a mapathon.
- **Meet the community.** State of the Map US is held every year, and the OSM community forum is at community.openstreetmap.org.

Whatever you use OSM for, credit it: "© OpenStreetMap contributors."

10.7 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

10.8 Summary

- **Spatial data quality** is fitness for use. Van Oort's seven elements are **lineage**, **positional accuracy**, **attribute accuracy**, **logical consistency**, **completeness**, **temporal quality**, and **usage, purpose, and constraints**.

- **Volunteered geographic information** appeared in the 2000s with Web 2.0, cheap GPS, and mobile internet. It comes as location-based social media, collaborative mapping, and citizen science. Its quality has to be measured, not assumed.
- **Big Data** is defined by volume, velocity, and variety; veracity is the quality question. The **FAIR** principles call for data that are findable, accessible, interoperable, and reusable.
- **OpenStreetMap** was founded in 2004. It is open under the **ODbL**, used by governments, technology companies, and aid agencies, and built from **nodes**, **ways**, **relations**, and free-form **tags**.
- **Extrinsic** measures compare OSM with authoritative data; **intrinsic** measures use OSM's own history. OSM's roads are often accurate, completeness is uneven, and areas with more editors tend to be better mapped.
- Tools such as Osmose, OSMCha, MapRoulette, and the HOT Tasking Manager check OSM's quality after edits are made.
- **Crisis mapping** began in Haiti in 2010 and led to the **Humanitarian OpenStreetMap Team**. In Nepal in 2015, newcomers outnumbered all earlier mappers, and **validation** by experienced mappers protected the map's quality.

10.9 Exercises

1. **Quality elements.** Find the metadata for a dataset you have used in this course (for example a TIGER/Line file or a layer from NC OneMap). Which of van Oort's seven elements does it describe? Which are missing?
2. **Tag a feature.** Look up a building and a road you know on openstreetmap.org. Click each (or use the *Query features* tool) and list its tags. Are any important attributes missing? Look up the recommended tags on the OSM Wiki.
3. **Extrinsic comparison.** Download OSM roads for Orange County, NC, from Geofabrik or with QuickOSM, and the TIGER/Line roads for the same county. Display them together and zoom in on three places: downtown Chapel Hill, a new subdivision, and a rural area. Where do they disagree? Which dataset is more complete in each place?
4. **Intrinsic history.** Pick a building on the UNC campus in OSM and view its history (*View History* on the feature page). How many versions does it have? How many different people edited it, and when?
5. **Map something.** Create an OSM account and complete the iD walkthrough. Then add or fix one feature near where you live, with a clear changeset comment. Or map one square on a HOT Tasking Manager project.
6. **Crisis mapping.** Read the Nepal case in this chapter. Write one paragraph on how you would balance openness and data quality if you ran the next activation.

10.10 Further reading and online resources

10.10.1 Readings

- Goodchild, "Citizens as Sensors: The World of Volunteered Geography" (Goodchild 2007), the article that named VGI.
- Haklay, "How Good Is Volunteered Geographical Information?" (Haklay 2010), the first large comparison of OSM with authoritative data.
- Haklay and others, "How Many Volunteers Does It Take to Map an Area Well?" (Haklay et al. 2010), on Linus's Law and the number of contributors.

- Parr, "Crisis Mapping and the Nepal Earthquake: The Impact of New Contributors" (Parr 2015), the source of the Nepal figures in this chapter.
- Zook and others, "Volunteered Geographic Information and Crowdsourcing Disaster Relief: A Case Study of the Haitian Earthquake" (Zook et al. 2010).
- Wilkinson and others, "The FAIR Guiding Principles for Scientific Data Management and Stewardship" (Wilkinson et al. 2016): https://www.nature.com/articles/sdata201618.
- *GIS Fundamentals: A First Text on Geographic Information Systems*, by Paul Bolstad (Bolstad 2022). Its chapter on spatial data standards and quality covers accuracy measures.
- GO FAIR, the FAIR principles explained: https://www.go-fair.org/fair-principles/.
- ISO 19157-1, *Geographic information: Data quality*: https://www.iso.org/standard/78900.html.

10.10.2 OpenStreetMap

- OpenStreetMap: https://www.openstreetmap.org.
- OpenStreetMap Wiki, the community's documentation, including tags: https://wiki.openstreetmap.org.
- OSM Wiki, "Elements," on nodes, ways, and relations: https://wiki.openstreetmap.org/wiki/Elements.
- OSM Wiki, "Map features," the most common tags: https://wiki.openstreetmap.org/wiki/Map_features.
- OSM Wiki, "Stats," on users and data: https://wiki.openstreetmap.org/wiki/Stats.
- OpenStreetMap Foundation, copyright and license (ODbL): https://www.openstreetmap.org/copyright.
- LearnOSM, beginner guides to mapping: https://learnosm.org.
- OpenStreetMap community forum: https://community.openstreetmap.org.
- OpenStreetMap US, including State of the Map US: https://www.openstreetmap.us.
- Geofabrik downloads, free OSM extracts: https://download.geofabrik.de.
- Overpass Turbo, for querying OSM data by tag in the browser: https://overpass-turbo.eu.
- QuickOSM plugin for QGIS: https://plugins.qgis.org/plugins/QuickOSM/.
- JOSM, the Java OpenStreetMap editor: https://josm.openstreetmap.de.
- Overture Maps Foundation: https://overturemaps.org.

10.10.3 Quality tools

- Osmose: https://osmose.openstreetmap.fr.
- KeepRight: https://www.keepright.at.
- OSMCha: https://osmcha.org.
- MapRoulette: https://maproulette.org.
- ohsome dashboard, from HeiGIT: https://ohsome-dashboard.heigit.org.

10.10.4 Humanitarian mapping

- Humanitarian OpenStreetMap Team: https://www.hotosm.org.
- HOT Tasking Manager: https://tasks.hotosm.org.
- Missing Maps: https://www.missingmaps.org.
- Kathmandu Living Labs: https://kathmandulivinglabs.org.

- YouthMappers, a network of university mapping chapters: https://www.youthmappers.org.



Part III

Unit 3: Spatial Analysis

With data in hand and its quality understood, this unit turns to answering questions with it. It starts with two ways of summarizing data over space: classing values for choropleth maps, and measuring density with neighborhoods and kernels. It then covers the core toolkit of GIS analysis, from selection, buffers, and overlay on vectors to map algebra on rasters. It ends by estimating values where nothing was measured, first with deterministic interpolation and then with kriging. By the end of the unit you can plan a chain of spatial operations, carry it out in QGIS and ArcGIS Pro, and judge how much to trust the result.

Chapter 11: Data Classing and Choropleth Maps

The same data can tell different stories depending on how they are grouped into classes. This chapter explains why choropleth maps should show rates and densities rather than counts, how to choose the number of classes, and how the main classification methods work: equal interval, quantile, natural breaks (Jenks), arithmetic and geometric intervals, and standard deviation. Using the population density of North Carolina's 100 counties, it compares the methods with histograms and the goodness of variance fit.

Chapter 12: Density and Neighborhoods

Density is a count divided by an area, but points have no area of their own. This chapter introduces Tobler's first law and spatial autocorrelation, and three ways of defining a neighborhood: by distance, by contiguity (rook's and queen's case), and by k nearest neighbors. It ends with kernel density estimation, the method behind hot-spot and heat maps, and the choices of kernel shape and bandwidth that change what the map shows.

Chapter 13: Spatial Analysis: Selection, Buffers, and Overlay

Most GIS questions are answered by chaining a few operations together. Starting with a light problem, where Batman should build his cave in Austin, this chapter introduces cartographic modeling and spatial scope (local, neighborhood, and global), then covers selection, joins, reclassification, dissolve, buffers of every kind, and overlay: point-in-polygon, clip, intersect, and union. It explains why overlay produces slivers and how to prevent them, and closes with map algebra, raster overlay cell by cell.

Chapter 14: Raster Analysis

A raster stores the world as a grid, and almost every raster tool repeats one operation cell by cell. This chapter covers cells, cell size, extent, and NoData; map algebra; and how to line up rasters with resampling, extent, and snap settings. It sorts raster functions by scope (local, neighborhood, zonal, and global) and works through mathematical, logical, reclassification, and conditional functions, moving windows, aggregation, zonal statistics, and distance.



Chapter 15: Spatial Interpolation

Weather stations, soil samples, and wells measure the world only at points. This chapter covers the five steps of interpolation and the main deterministic methods: nearest neighbor (Thiessen polygons), average neighbor, inverse distance weighting, polynomial trend surfaces, and radial basis functions. It shows how search neighborhoods and sampling designs shape the result, and how cross validation and RMSE compare methods objectively.

Chapter 16: Kriging

Kriging measures how values in the data are related at different distances before deciding how to weight them. This chapter builds the semivariogram from semivariance, explains its nugget, sill, and range, and shows how kriging sets its weights, handles anisotropy and trends, and differs in its ordinary, simple, universal, and cokriging forms. It ends with the cross-validation statistics and prediction standard errors used to judge a kriged surface.

11 Data Classing and Choropleth Maps

Learning objectives

By the end of this chapter you should be able to:

1. Explain what a choropleth map shows, and why it needs normalized data rather than raw counts.
2. Decide whether to class data at all, and how many classes a map can carry.
3. Describe how six classification methods set class breaks: equal interval, quantile, arithmetic and geometric progression, natural breaks (Jenks), and standard deviation.
4. Read a histogram to choose a method that fits a data distribution and the purpose of a map.
5. Measure how well classes fit the data with the goodness of variance fit.
6. Class and symbolize a choropleth map in ArcGIS Pro and QGIS.

Unit 2 was about getting good data. Unit 3 is about what you do with it, and the simplest analysis of all is a map. Most thematic maps of places, from election results to disease rates, are **choropleth maps**: maps that shade each area by its value. They look easy to make, and every GIS will make one in a few clicks. But behind every choropleth map are two decisions that can change its story completely: how many classes to use, and where to put the breaks between them.

This chapter maps one dataset, the population density of North Carolina's 100 counties, many different ways. Every difference you see between the maps comes from those two decisions, not from the data.

11.1 Choropleth maps

11.1.1 Shading areas by value

A **choropleth map** shades each **enumeration unit**, an area for which data are collected and reported, such as a county, a census tract, or a state, by its value. Darker or more saturated shades usually mean higher values. The areas are fixed in advance: their boundaries come from the data source, not from the pattern being mapped.

Choropleth maps are everywhere because so much data is published for administrative areas. They are good at showing how a rate or a share varies from place to place, and at helping readers find their own area. They also have weaknesses, which this chapter returns to: they show each area as uniform inside, and large areas dominate them whatever their values.

Did you know..?

The first choropleth map was made in 1826 by the French mathematician Charles

Dupin. It shaded each French *département* by the number of boys in school relative to its population, from white for the best-educated to black for the least, and it gave the French their first picture of a divided country: an educated north and a poorer south. The name *choropleth*, from the Greek *choros* (place) and *plethos* (multitude), was coined by the American geographer John K. Wright more than a century later, in 1938.

11.1.2 Normalize first

The most common mistake on a choropleth map is to shade the areas by a **raw count**, such as the number of people, cases, or crimes. Big areas tend to have big counts simply because they are big, or because more people live in them. A reader's eye is also drawn to large areas. So a map of raw counts exaggerates large units twice, and mostly shows where the people are.

The fix is to **normalize** the counts before mapping them:

- **Divide by area** to get a density, such as people per square mile.
- **Divide by population** to get a rate or a share, such as deaths per 100,000 residents or the percentage of votes cast.
- **Use a summary statistic**, such as a median income or an average age.

Figure 11.1 shows the difference. On the left, the raw population count makes the large urban counties stand out, along with big counties such as Buncombe and Onslow. On the right, population density (people per square mile of land) shows how crowded each county is.

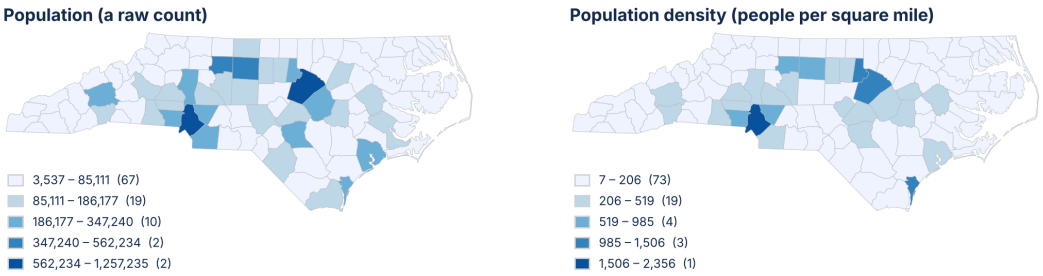


Figure 11.1: North Carolina counties by population (left) and population density (right), both in five natural breaks classes. Population: U.S. Census Bureau estimates for 2025. Land area: 2020 Census Gazetteer.

A rule of thumb: **if a number would grow just because the area or its population grew, normalize it before you make a choropleth map.** Raw counts belong on other kinds of map, such as proportional symbol maps, where the symbol's size shows the count.

Think about it

A news site maps the number of COVID-19 cases in each North Carolina county, and Wake and Mecklenburg are the darkest. Does that tell you where the risk of catching COVID-19 was highest? What would you map instead?

11.1.3 Look at the distribution first

Before choosing classes, look at how the values are spread. The best tool is a **histogram**, a bar chart of how many areas fall in each range of values (Figure 11.2).

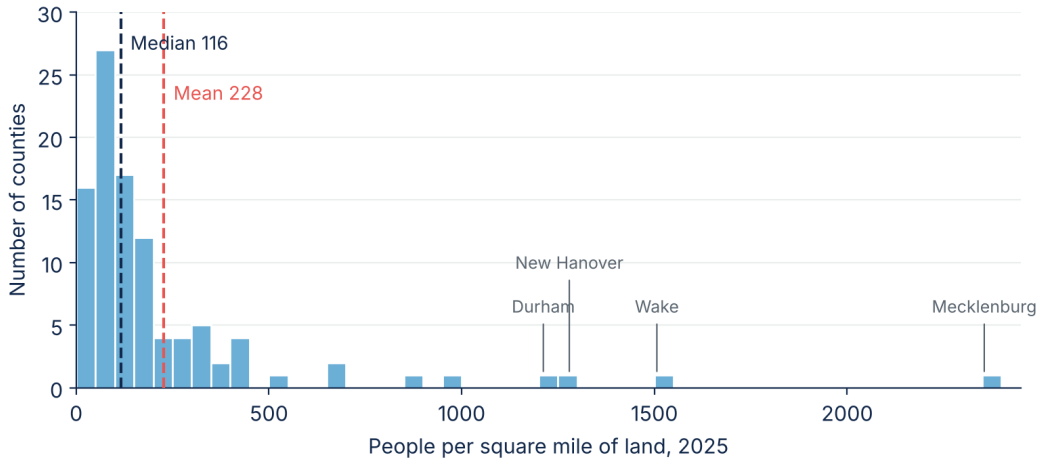


Figure 11.2: The distribution of population density in North Carolina's 100 counties, 2025.

North Carolina's county densities are strongly **skewed**. Half of the counties have fewer than 116 people per square mile, but a few urban counties have many more: Mecklenburg has 2,356. The mean (228) is twice the median (116), pulled up by those few counties. Skewed data are the rule, not the exception, for counts and densities of people, and the shape of the histogram decides which classing methods will work.

11.2 How many classes?

11.2.1 To class or not to class

The first decision is whether to class the data at all. In an **unclassed map**, each area gets its own shade, in proportion to its value. In a **classed map**, the values are grouped into a few classes, and every area in a class gets the same shade.

In 1973, Waldo Tobler argued that once computers could shade each area by its exact value, classes were no longer needed (Tobler 1973). The data are continuous, he said, so the symbols should be too, and the reader cares about the overall pattern, not exact values. The case for classing is about the reader. People match a few distinct shades to a legend far more accurately than they read a value from a smooth ramp; a handful of classes is easier to remember; and class breaks can mark values that matter, such as a poverty line. Most published choropleth maps are classed, but unclassed maps are a good choice when the overall pattern is what matters.

11.2.2 How many can readers tell apart?

Figure 11.3 maps the same densities in 2, 5, and 9 classes, and unclassed. With two classes, the map is simple, but it hides almost everything: a county with 117 people per square mile looks the same as Mecklenburg. With nine, it keeps more detail, but try to match the middle shades of the map to the legend.

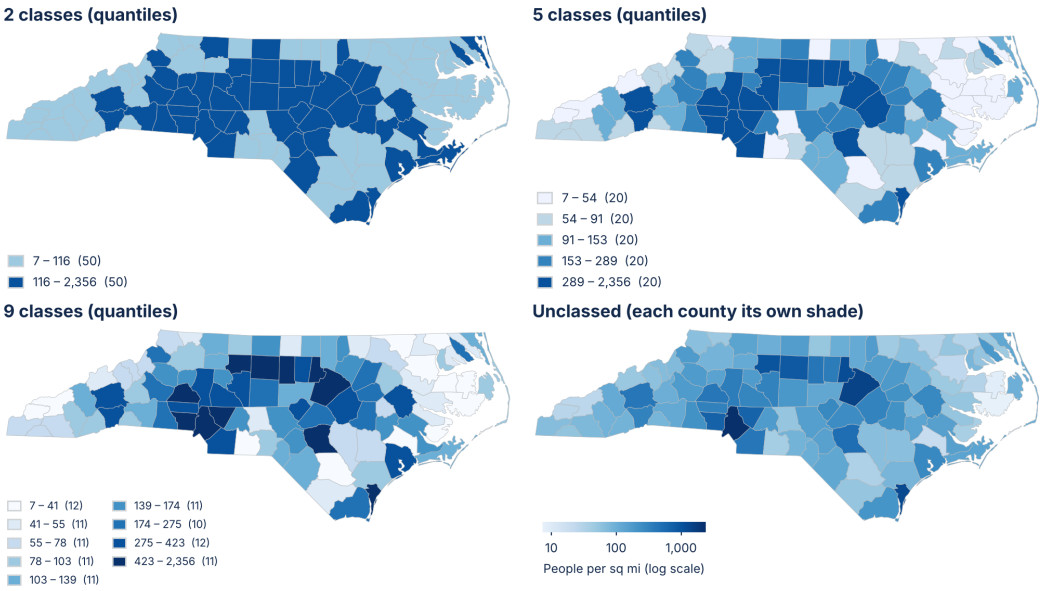


Figure 11.3: North Carolina population density in 2, 5, and 9 quantile classes, and unclassed. The unclassed map uses a logarithmic scale.

Some guidelines:

- **Most readers can tell apart about five to seven shades** of a single color. Beyond that, they start to confuse neighboring classes.
- **Use fewer classes** when there are few areas, when the map will be printed in grayscale, or when the audience is general.
- **Use more classes** only when readers need the detail, and test that they can tell the shades apart.
- **Sturges' rule** gives a statistical starting point: $k \approx 1 + 3.3 \log_{10}(n)$, where n is the number of areas (Sturges 1926). For North Carolina's 100 counties, $k \approx 1 + 3.3 \times 2 = 7.6$, about 8. That is at the limit of what readers can tell apart, so in practice you would use five or six.
- **ColorBrewer** (colorbrewer2.org) offers color schemes for 3 to 9 classes that have been tested for legibility and for color-blind readers (Harrower and Brewer 2003).

11.2.3 Classing is generalization

Classing is a kind of **map generalization**, the cartographer's toolkit for deciding what to keep and what to leave out. Generalization has three broad operations: **simplification** removes detail, **classification** groups values or features together, and **enhancement** adds detail or emphasis back where the reader needs it. When we class data, the goal is a map whose symbols fit the real distribution of the data, with as little of the important detail lost as possible.

One kind of detail is lost on every choropleth map, classed or not. A county's density is an average. It doesn't mean that people are spread evenly across the county: Mecklenburg has dense neighborhoods and nearly empty ones, and most of Hyde County's people live in a few small towns. Assuming that everyone in an area shares its average value is the **ecological fallacy**. Maps that redistribute values within each area, such as **dasymetric maps**, which place people only on land where people live, are one answer to it.

11.3 Where to put the breaks

11.3.1 Classing as modeling

The second decision is where to put the **class breaks**, the values that divide one class from the next. Each classification method puts them somewhere different, and so highlights a different side of the data. It is rarely obvious in advance which will work best, so treat classing as modeling:

1. **Model the data.** Plot a histogram, and look for skew, gaps, and outliers.
2. **Try several methods.** A pattern that appears under every method is probably real. One that appears under only one method may be an artifact of that method.
3. **Measure the error.** Check how well the classes fit the data, for example with the goodness of variance fit (Section 11.4).

Figure 11.4 maps North Carolina's county densities with six methods, each with five or six classes. Figure 11.5 shows where each method puts its breaks on the histogram. The numbers in brackets in the legends are the number of counties in each class.

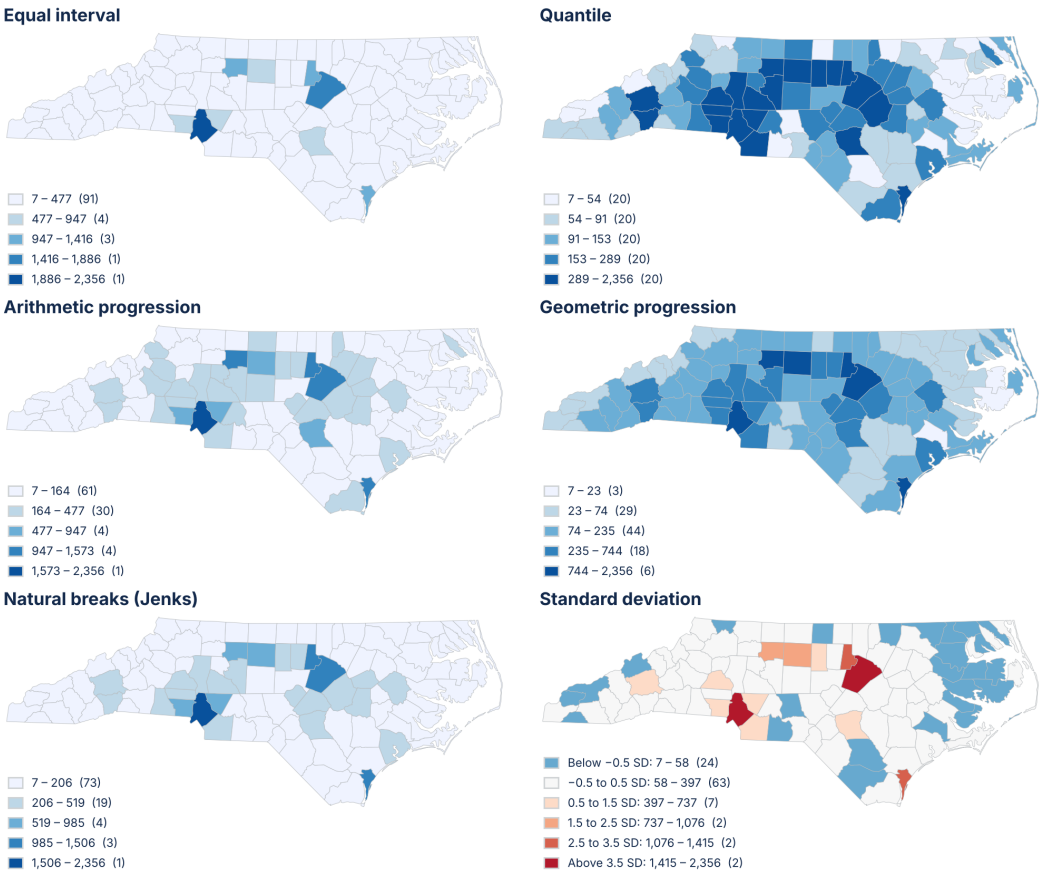


Figure 11.4: North Carolina population density by county, 2025, with six classification methods.

11.3.2 Equal interval

Equal interval divides the range of the data, from the minimum to the maximum, into classes of equal width. North Carolina's densities run from 7 to 2,356, so five equal classes are each about 470 people per square mile wide.

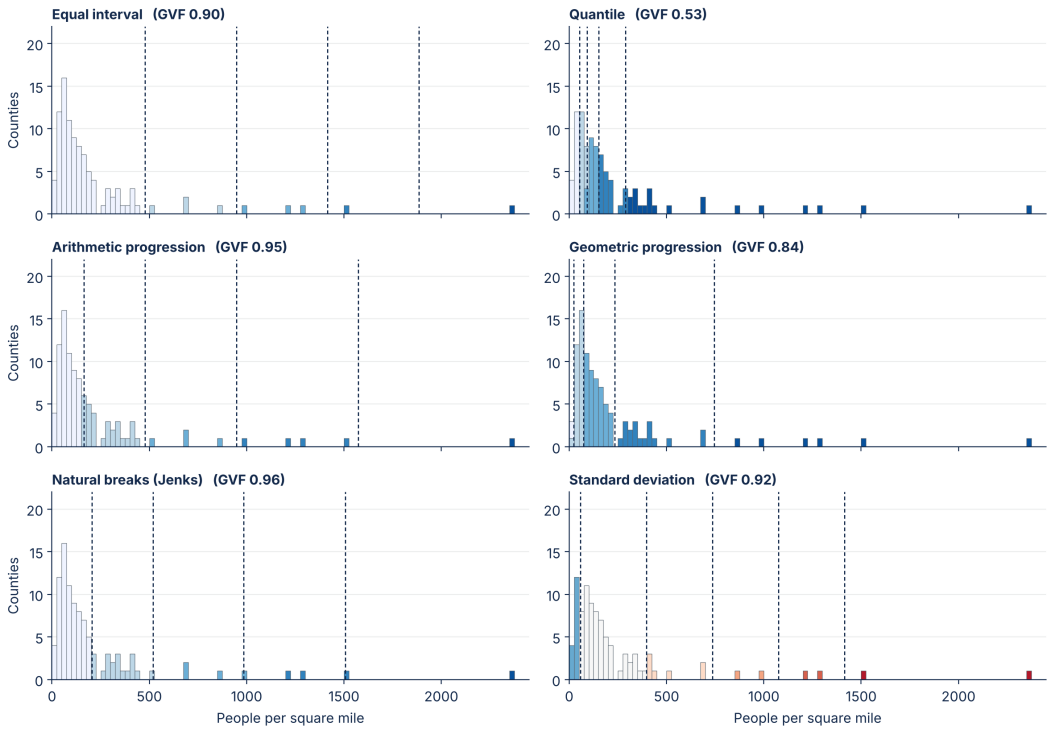


Figure 11.5: Where each method puts its class breaks (dashed lines) on the histogram of county densities, with the goodness of variance fit (GVF) of each.

- **Good for:** data spread fairly evenly across their range, and legends that are easy to read, especially when the breaks are rounded to familiar values (0, 500, 1,000, …).
- **Watch out for:** skewed data. Here, 91 of the 100 counties fall in the lowest class, and the map says little except that a few cities are dense.

11.3.3 Quantile

Quantile classing puts the same number of areas in each class. With five classes these are **quintiles**, 20 counties each; with four, **quartiles**.

- **Good for:** ranking places (a dark county is in the top fifth), and maps where every shade is used. Readers find quantile maps easy to use, and they work well for comparing a series of maps (Brewer and Pickle 2002).
- **Watch out for:** breaks that ignore the gaps in the data. Counties with nearly the same density can land in different classes, while very different counties share one: the top class here runs from 289 to 2,356, so Mecklenburg looks the same as counties with an eighth of its density.

Did you know..?

Which method do readers find easiest to use? Cynthia Brewer and Linda Pickle tested seven methods on maps of death rates, asking people to read and compare them (Brewer and Pickle 2002). Quantile classing, often criticized by cartographers, was among the best. The popular natural breaks method did less well, partly because its breaks change from map to map, so the same shade means different values on each one.

11.3.4 Arithmetic progression

In an **arithmetic progression**, each class is wider than the one before by a constant amount: if the first class is w wide, the next is $2w$, then $3w$, and so on.

- **Good for:** data that thin out steadily toward high values. It gives more classes to the low end than equal interval does.
- **Watch out for:** strongly skewed data. Here it helps a little, but 61 counties are still in the first class and 30 in the second.

11.3.5 Geometric progression

In a **geometric progression**, each class is wider than the one before by a constant *ratio*. Equivalently, the breaks are evenly spaced on a logarithmic scale. North Carolina's densities span about 2.5 orders of magnitude (7 to 2,356), so each of five breaks is about 3.2 times the one before: 23, 74, 235, and 744.

- **Good for:** strongly skewed data, such as population density or income, whose values span orders of magnitude. Compare it with equal interval: the same data now show a clear gradient from rural counties to the cities.
- **Watch out for:** irregular break values that are harder for readers to interpret.

11.3.6 Natural breaks (Jenks)

The **natural breaks**, or **Jenks**, method groups similar values together. It searches for the breaks that make the classes as alike inside as possible: it minimizes the variance within each class and maximizes the variance between them (Jenks 1977). In practice, it tends to put breaks at the gaps in the histogram.

- **Good for:** data with clusters and gaps. It is the default method in ArcGIS Pro.
- **Watch out for:** comparing maps. Because the breaks are fitted to one dataset, two natural breaks maps of different years or places usually have different breaks, and can't be compared directly. On skewed data, natural breaks also spends most of its classes on the few high values: here, 73 counties share the lowest class.

(The method is named for George Jenks, a cartographer at the University of Kansas, so it is *Jenks*, not *Jenk's*.)

11.3.7 Standard deviation

Standard deviation classing sets breaks at fixed distances from the mean, such as every half or whole standard deviation. A county in the class "0.5 to 1.5 SD" is between half and one and a half standard deviations above the mean.

- **Good for:** showing which areas are above or below average. Because the classes have a meaningful midpoint, the mean, this is one of the few cases where a **diverging color scheme** fits: one hue for values below the midpoint, another for values above it, and a light neutral color in the middle.

- **Watch out for:** skewed data. The method assumes a roughly normal, bell-shaped distribution. Here, the lowest class, below −0.5 SD, already reaches down to the least dense county: no county could be even 0.7 standard deviations below the mean, which would be a negative density. Mecklenburg, meanwhile, is more than six standard deviations above it.

11.3.8 Manual (defined) intervals

Finally, you can set the breaks yourself, at values that mean something: a poverty threshold, a regulatory limit, or round numbers. **Manual** or **defined intervals** are also the best choice when several maps must be compared, such as the same variable across census years, because every map then uses the same classes.

11.3.9 Matching the method to the data

Table 11.1: Classification methods and when to use them.

Method	How the breaks are set	Best for
Equal interval	Equal-width steps across the range	Evenly spread data; round break values
Quantile	The same number of areas in each class	Ranking places; comparing a series of maps
Arithmetic progression	Widths grow by a constant amount	Moderately skewed data
Geometric progression	Widths grow by a constant ratio	Strongly skewed data
Natural breaks (Jenks)	Minimize the variance within classes	Clustered data with gaps
Standard deviation	Fixed steps from the mean	Above or below average; near-normal data
Manual (defined)	The analyst sets meaningful thresholds	Policy cutoffs; maps compared over time

The best method depends on two things: the shape of the distribution, which you see in the histogram, and the purpose of the map.

Think about it

You are making a map for county commissioners showing which counties are more densely populated than the state as a whole (about 230 people per square mile). Which method would you use, and what color scheme? Would your answer change if the map were for a report comparing 2010 and 2025?

11.4 Measuring the fit

How do you know whether a set of classes fits the data? One measure is the **goodness of variance fit (GVF)**, which compares how spread out the values are within each class with how spread out they are overall:

$$\text{GVF} = 1 - \frac{\text{SDCM}}{\text{SDAM}}$$

- **SDAM**, the sum of squared deviations from the array mean, adds up the squared difference between each value and the mean of all the values.
- **SDCM**, the sum of squared deviations from the class means, adds up the squared difference between each value and the mean of its own class.

If every class held only identical values, SDCM would be 0 and the GVF would be 1. The closer the GVF is to 1, the better the classes fit the data. The GVF of each method for North Carolina's densities is shown in Figure 11.5.

Natural breaks has the highest GVF (0.96), as it should: it is designed to maximize it. Quantile has the lowest (0.53), because its top class lumps together counties from 289 to 2,356. But the GVF is not the only test of a good map. The quantile map shows far more of the state's geography than the natural breaks map, in which 73 counties look alike. A high GVF means the classes describe the *values* well; whether they tell the *story* well is a judgment about the map's purpose.

11.5 Choropleth maps in a GIS

11.5.1 Color schemes

Choose the color scheme to fit the data:

- **Sequential** schemes, in light-to-dark steps of one hue (or a few related hues), fit most data, which run from low to high. Light usually means low.
- **Diverging** schemes, two hues that meet at a light neutral color, fit data with a meaningful midpoint, such as values above and below a mean, a median, or zero (a change from one year to the next).
- **Qualitative** schemes, of distinct hues, fit categories, such as land-cover types, not quantities.

11.5.2 ArcGIS Pro

1. **Prepare the data.** Join your table to its boundaries (Chapter 6), and add a field with the normalized value, or plan to normalize in the symbology.
2. **Open the symbology.** Right-click the layer, choose *Symbology*, and set *Primary symbology* to **Graduated Colors**.
3. **Choose the field.** Set *Field* to the value to map. To divide by another field on the fly, set *Normalization* (for example, a population field to map a count per person, or a land-area field to map a density).
4. **Choose the method and classes.** Set *Method* (Natural Breaks (Jenks) is the default; the others include Equal Interval, Quantile, Standard Deviation, Geometrical Interval, Defined Interval, and Manual Interval) and the number of *Classes*.
5. **Check the histogram.** The *Histogram* tab shows the distribution with the breaks marked; you can drag breaks to new values, which switches the method to Manual Interval.
6. **Choose a color scheme**, and edit the class labels into rounded, readable values.

ArcGIS Pro's *Geometrical Interval* is its own method, designed for skewed data. It is similar to, but not exactly the same as, the geometric progression described above.

11.5.3 QGIS

1. **Prepare the data.** Join and normalize as needed. You can also type an expression instead of a field, such as `"POP_2025" / "ALAND_SQMI"`.
2. **Open the symbology.** Open *Layer Properties* > *Symbology* and choose **Graduated**.
3. **Choose the value and ramp.** Set *Value* to the field or expression and choose a *Color ramp*.
4. **Choose the mode and classes.** Set *Mode*: Equal Interval, Equal Count (Quantile), Natural Breaks (Jenks), Standard Deviation, Pretty Breaks, or Logarithmic scale. Set the number of *Classes*, and click *Classify*.
5. **Check the histogram.** The *Histogram* tab plots the values with the breaks; click *Load values* to see it. Double-click a class's values to edit its breaks.

A checklist for choropleth maps

- Is the value normalized (a rate, share, density, or average), not a raw count?
- Did you look at the histogram before choosing a method?
- Can readers tell the classes apart (usually 3 to 7)?
- Does the color scheme fit the data: sequential, or diverging around a meaningful midpoint?
- Are the class labels rounded and readable, with no gaps or overlaps?
- If maps will be compared, do they share the same classes?

11.6 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

11.7 Summary

- A **choropleth map** shades each enumeration unit by its value. **Normalize** counts first, by area, by population, or with a summary statistic, because raw counts mostly show where the people are.
- Look at the **histogram** before choosing classes. Counts and densities of people are usually strongly **skewed**.
- **Unclassed** maps show every value; **classed** maps are easier to read and remember. Most readers can tell apart **five to seven** shades; **Sturges' rule** gives a statistical starting point.
- Classing is **generalization**. Every choropleth map also hides variation inside each area, and assuming everyone shares the average is the **ecological fallacy**.
- **Equal interval** gives equal-width classes; **quantile**, equal counts; **arithmetic** and **geometric progressions**, classes that widen by a constant amount or ratio; **natural breaks (Jenks)**, classes of similar values; **standard deviation**, steps from the mean; and **manual** intervals, meaningful thresholds.
- The **goodness of variance fit** measures how well the classes fit the values; the purpose of the map decides whether they tell the right story.
- Use **sequential** color schemes for most data and **diverging** schemes for data with a meaningful midpoint.

11.8 Exercises

1. **Normalize.** For each variable, say whether it can go on a choropleth map as it is, and if not, how you would normalize it: (a) number of hospitals per county; (b) median age; (c) acres of farmland; (d) number of people without health insurance; (e) the percentage of adults with a bachelor's degree.
2. **Equal intervals by hand.** A variable ranges from 12 to 87. Compute the breaks for four equal-interval classes. Then compute the breaks for four classes in a geometric progression (hint: the ratio is $(87/12)^{1/4}$).
3. **Six methods.** Download the population of North Carolina's counties and their land areas (Chapter 8), compute population density, and map it in ArcGIS Pro or QGIS with equal interval, quantile, natural breaks, and standard deviation classes. Which map would you publish, and why?
4. **Histograms.** Open the histogram of the same data in your GIS. Then map the natural log of density with equal intervals. How does the histogram change? How does the map compare with the geometric progression map in Figure 11.4?
5. **Number of classes.** Make the same quantile map with 3, 5, 7, and 9 classes. At what point can you no longer match every shade on the map to the legend?
6. **Map a series.** Map the same variable (for example, the percentage of people in poverty) for two years. Class each with natural breaks, then both with the same manual breaks. Which pair of maps lets you see change correctly?

11.9 Further reading and online resources

11.9.1 Books and articles

- *Thematic Cartography and Geovisualization*, by Terry Slocum, Robert McMaster, Fritz Kessler, and Hugh Howard (Slocum et al. 2022), the standard textbook on classing and choropleth maps.
- *How to Lie with Maps*, by Mark Monmonier (Monmonier 2018), on how choices such as class breaks change a map's message.
- *Designing Better Maps: A Guide for GIS Users*, by Cynthia Brewer (Brewer 2016), with practical advice on classes and color.
- Tobler, "Choropleth Maps Without Class Intervals?" (Tobler 1973).
- Brewer and Pickle, "Evaluation of Methods for Classifying Epidemiological Data on Choropleth Maps in Series" (Brewer and Pickle 2002).
- Harrower and Brewer, "ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps" (Harrower and Brewer 2003).
- *GIS Fundamentals: A First Text on Geographic Information Systems*, by Paul Bolstad (Bolstad 2022), on classification of attribute data.

11.9.2 Online resources

- ColorBrewer 2.0, color schemes for maps: https://colorbrewer2.org.
- Axis Maps, *Cartography Guide*, on choropleth maps, normalization, and data classification: https://www.axismaps.com/guide/choropleth.
- The Nature of Geographic Information, Penn State, on thematic maps and data classification: https://www.e-education.psu.edu/natureofgeoinfo/c3_p13.html.
- *Geography 486: Cartography and Visualization*, Penn State, on classing and color: https://courses.ems.psu.edu/geog486/.

- The GIS&T Body of Knowledge, “Statistical Mapping (Enumeration, Normalization, Classification)” and “Choropleth Maps”: https://www.ucgis.org/site/gis-t-body-of-knowledge.
- Esri, “Data classification methods” in ArcGIS Pro: https://pro.arcgis.com/en/pro-app/latest/help/mapping/layer-properties/data-classification-methods.htm.
- Esri, “Graduated colors” symbology in ArcGIS Pro: https://pro.arcgis.com/en/pro-app/latest/help/mapping/layer-properties/graduated-colors.htm.
- QGIS User Guide, “Graduated Renderer”: https://docs.qgis.org/latest/en/docs/user_manual/working_with_vector/vector_properties.html#graduated-renderer.
- Kenneth Field, *Cartography.* (Esri Press, 2018), a richly illustrated reference with entries on classification and choropleth maps.
- Datawrapper Academy, “What to consider when creating choropleth maps”: https://academy.datawrapper.de/article/134-what-to-consider-when-creating-choropleth-maps.
- U.S. Census Bureau, 2020 Census Gazetteer files (land area for every geography): https://www.census.gov/geographies/reference-files/time-series/geo/gazetteer-files.html.

12 Density and Neighborhoods

Learning objectives

By the end of this chapter you should be able to:

1. Calculate a density as a count divided by an area, and explain why the choice of area changes the answer.
2. State Tobler's first law of geography, and explain how it relates to spatial autocorrelation.
3. Define neighborhoods by distance, by contiguity (rook's and queen's case), and by number of nearest neighbors, and name the strengths and pitfalls of each.
4. Explain how kernel density estimation turns points into a continuous density surface.
5. Describe how bandwidth, kernel shape, and cell size change a density surface.
6. Run kernel density in ArcGIS Pro and QGIS, and interpret the result.

Chapter 11 mapped the population density of North Carolina's counties. The counties were ready-made areas to divide by. But much of the data we care about arrives as points with no area around them: crimes, trees, cases of a disease, car crashes, or 911 calls. Where are they concentrated? To answer that, we need an area to divide by, and when no area exists, we have to define one. This chapter is about how to do that: with **neighborhoods**, and with **kernel density estimation**, which turns a set of points into a smooth surface of density.

12.1 Density: things per area

Density is a count divided by an area. **Population density** is the number of people divided by the area they live in. North Carolina had 10,439,388 people in the 2020 Census, so:

$$\text{density} = \frac{10{,}439{,}388 \text{ people}}{53{,}819 \text{ mi}^2} \approx 194 \text{ people per mi}^2$$

But 53,819 square miles is North Carolina's *total* area, including its sounds, lakes, and rivers. The Census Bureau divides by *land* area alone, 48,618 square miles, which gives 214.7 people per square mile. Same people, same state, different answer.

This is the theme of the chapter: **the area you divide by changes the density.** For a state, the choice is between total and land area. For points, there is no area at all until we define one, and every choice of area gives a different result.

Think about it

Think of a place you know well in Chapel Hill. Where would you expect the most people, crimes, or trees per square mile? What area would you divide by to measure it?

12.2 Neighborhoods

12.2.1 What is a neighborhood?

Ask people to describe their neighborhood, and you get three kinds of answer:

- **A distance**: "everything within a 10-minute walk."
- **Shared borders**: "my street, and the streets that touch it."
- **The nearest neighbors**: "the four houses around mine."

In spatial analysis, a **neighborhood** is a zone of analysis drawn around a location, and those are the three ways to draw it: by **distance**, by **contiguity**, or by a **number of neighbors**. We need neighborhoods because many things we map show **neighborhood effects**. A house's price tracks the prices of the houses around it; dense blocks sit next to dense blocks; species cluster in suitable habitat; and a disease spreads to nearby people first.

12.2.2 Tobler's first law

Why are nearby places alike? Waldo Tobler put it in one sentence in 1970:

> "Everything is related to everything else, but near things are more related than distant things." (Tobler 1970)

This is **Tobler's first law of geography**. Its measurable version is **spatial autocorrelation**: values at nearby locations are more similar (**positive** spatial autocorrelation) or more different (**negative** spatial autocorrelation) than chance would predict. Most geographic data show positive spatial autocorrelation, which is what makes maps useful: if nearby places were no more alike than distant ones, a map would look like random noise.

The law leaves open what counts as *near*. A neighborhood definition is how we decide, so the rest of this section is about making that choice explicit. None of the three definitions is correct in general. The right one depends on the process you are studying: walking distance for access to a grocery store, shared borders for a policy that spills over from one county to the next.

Did you know..?

Tobler's first law first appeared in a paper about a computer movie. In 1970, Tobler made one of the first computer-animated films in geography, simulating the growth of the Detroit region's population from 1910 to 2000. To keep the model simple, he assumed that each place was influenced most by the places closest to it, and stated the assumption as a "first law." It has been quoted, and argued over, ever since. Tobler later said he had called it a law a little tongue in cheek.

12.2.3 By distance

A **distance-based neighborhood** includes every location within a set radius, d, of the location in question (Figure 12.1). In the simplest, **binary** version, each pair of locations gets a weight of 1 if they are neighbors and 0 if they are not. Repeating this for every location builds a table of who neighbors whom, often stored as a **spatial weights matrix**.

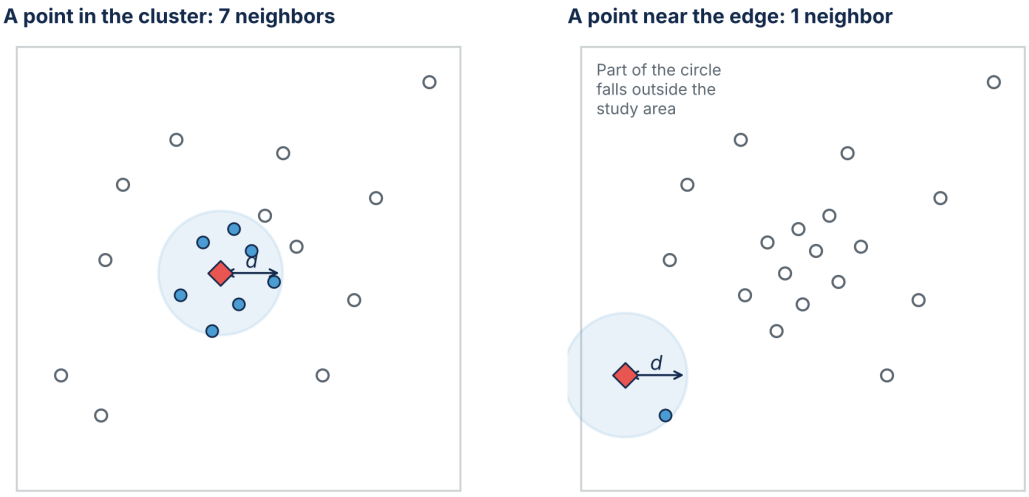


Figure 12.1: Distance neighborhoods. Every point within a radius *d* of the red diamond (filled blue) is its neighbor; the hollow points are not. The points are made up.

Because the cutoff is a fixed distance, the data must be in a **projected coordinate system** (Chapter 4), so that d is in meters or feet, not degrees.

Distance neighborhoods have two weaknesses:

- **Isolated locations may have no neighbors at all.** A point in an empty corner of the map gets nothing within d.
- **The edge effect.** A point near the edge of the study area has part of its circle outside the data, so it finds fewer neighbors than it should (Figure 12.1, right). Its true neighbors exist, but we never observed them. Boundary locations look less dense than they are. The fix is to collect data in a buffer beyond the study area, or to apply an edge correction.

12.2.4 By contiguity

For polygons, such as counties or census tracts, we can define neighbors by **contiguity**: polygons that touch are neighbors. No distance is needed: shared borders define the neighborhood. Contiguity is a *relative*, topological measure of nearness. Two counties are neighbors because they touch, not because their centers are within some distance.

There are two ways for polygons to touch, named for chess pieces (Figure 12.2):

- In **rook's case** contiguity, neighbors share an *edge*. The rook moves only in straight lines.
- In **queen's case** contiguity, neighbors share an edge *or a corner*. The queen moves straight or diagonally. Queen's case always finds the same or more neighbors than rook's case. It is the default in many tools, such as GeoDa.

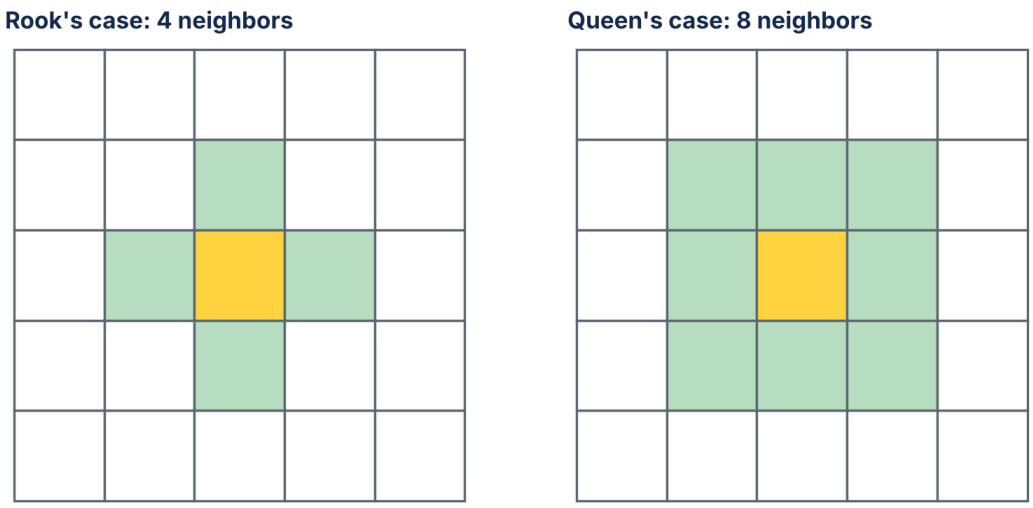


Figure 12.2: Rook's and queen's case contiguity on a grid. The yellow cell's first-order neighbors are green.

The neighbors that touch a polygon directly are its **first-order** neighbors. The neighbors of those neighbors are **second-order** neighbors, and so on. On real maps, few polygons meet only at a corner, so rook's and queen's case often agree. Orange County has the same five first-order neighbors either way (Figure 12.3). The famous exception is the Four Corners, where Arizona and Colorado, and Utah and New Mexico, touch only at a point.

Contiguity works only for polygons, and it struggles with odd shapes and **islands**: an island, or a county split by water, may touch fewer neighbors than it is really connected to by bridges and ferries, or none at all.

12.2.5 By number of neighbors

A **k-nearest-neighbor** neighborhood fixes the *number* of neighbors, k, instead of the distance. Each location reaches out until it has found its k closest locations (Figure 12.4). Distance doesn't matter, only rank: the nearest wins.

This is both the strength and the weakness of the method. Every location gets exactly k neighbors, however sparse its surroundings, so no point is left out. But an isolated point's neighbors may be far away, and not really related to it. With distance neighborhoods, the size of the neighborhood is fixed and the number of neighbors varies; with k nearest neighbors, the number is fixed and the size varies with the density of points. Larger k smooths the results but blurs local detail.

Table 12.1: Three ways to define neighborhoods.

Definition	Neighbors are	Works best for	Watch out for
Distance	All locations within radius *d*	Points; density estimation	Edge effects; isolated points get none
Contiguity	Polygons sharing an edge (rook) or an edge or corner (queen)	Polygons such as counties	Odd shapes and islands

Definition	Neighbors are	Works best for	Watch out for
k nearest	The *k* closest locations	Uneven or sparse points	Neighbors may be far away

Think about it

You are studying whether a county's opioid overdose rate is related to the rates in the counties around it. Which neighborhood definition would you use? What about studying whether a coffee shop's sales depend on the other coffee shops nearby?

12.3 Kernel density estimation

12.3.1 From points to a surface

Density estimation almost always uses distance neighborhoods, because density is a quantity per unit of area. **Kernel density estimation (KDE)** goes one step further: it lets nearer points count more than farther ones, and it produces a continuous surface, a **raster**, rather than one value per point.

Picture a grid of cells laid over the study area (Figure 12.5). A **kernel**, a window with a set radius called the **bandwidth** or **search radius**, is centered on each cell in turn. At each cell, the tool:

1. finds the points that fall inside the kernel;
2. gives each one a weight, from the kernel's shape: more for nearby points, less for distant ones;
3. adds up the weights and divides by the kernel's area;
4. writes the result to that cell of the output raster.

The result is a smooth surface whose value in each cell is the density of points around it, in points per square mile or per square kilometer. Maps of such surfaces are often called **heat maps**.

The idea was developed in statistics in the 1950s and 1960s, to estimate the shape of a distribution from a sample without assuming it was normal (Silverman 1986). Geographers applied the same idea in two dimensions.

12.3.2 Kernel shape

The kernel's shape decides how much weight each point gets (Figure 12.6):

- A **uniform** kernel gives every point inside the bandwidth the same weight, like a cookie cutter. It is the simplest, and gives the same result as counting points in a circle.
- A **quartic** kernel gives the most weight to points at the center and less and less toward the edge, reaching zero at the bandwidth. It is the default in ArcGIS Pro and QGIS.
- A **Gaussian** (normal) kernel is bell-shaped, and in theory never quite reaches zero.

The shape of the kernel changes the result far less than its size does (Silverman 1986).

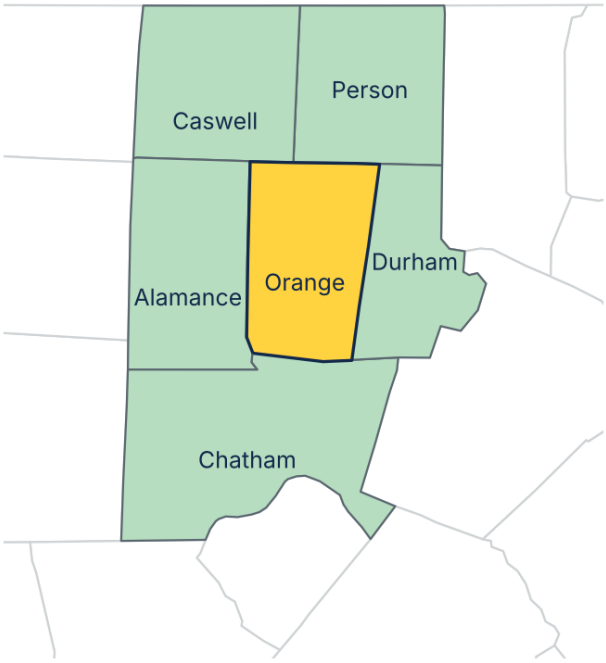


Figure 12.3: Orange County, North Carolina (yellow), and its five first-order neighbors (green).

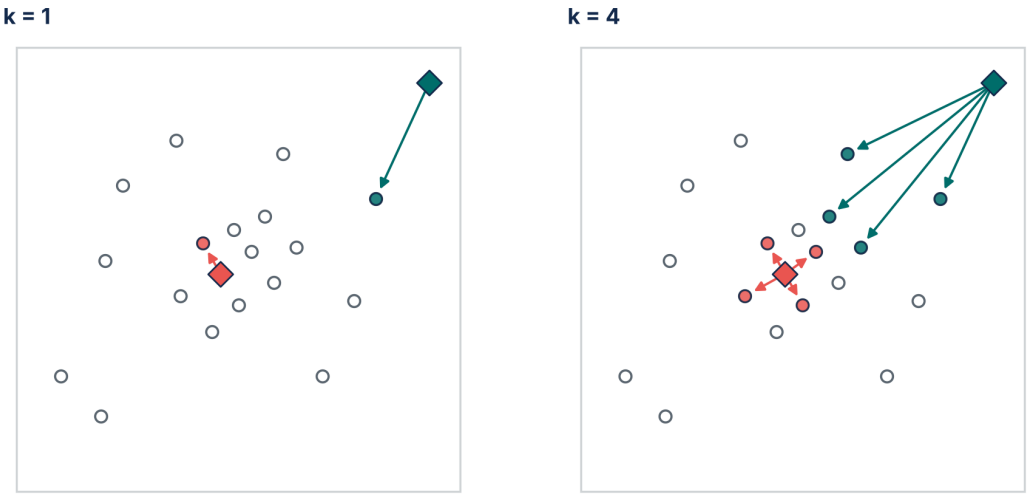


Figure 12.4: k nearest neighbors with k = 1 and k = 4, for a point in the cluster (red) and an isolated point in the corner (green). The points are the same as in Figure 12.1.

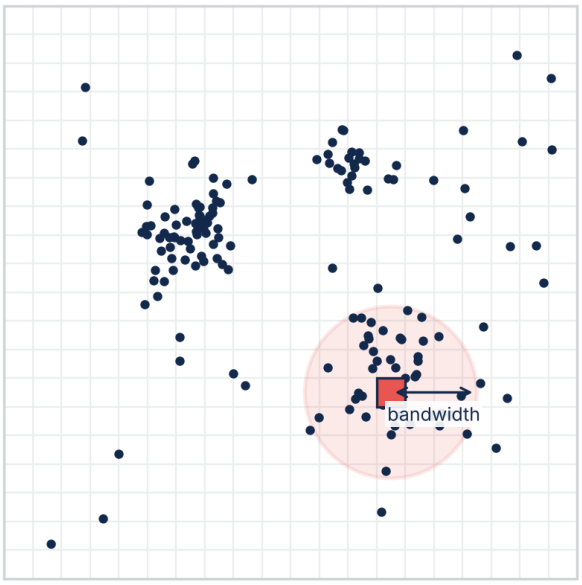


Figure 12.5: A kernel of a set bandwidth (red circle) centered on one cell (red square) of a grid laid over made-up incident points. The kernel moves to every cell in turn.

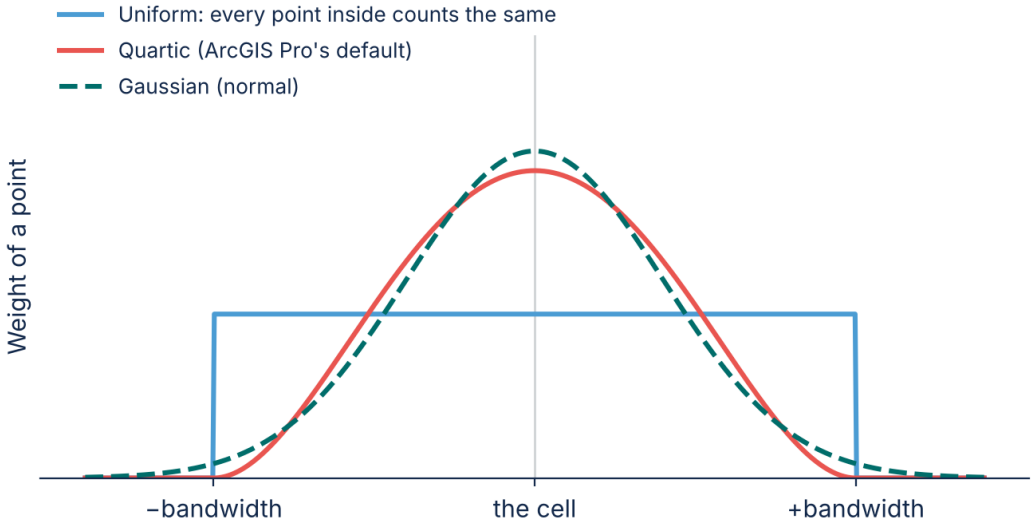


Figure 12.6: Three kernel shapes. Each shows how much weight a point gets, depending on its distance from the cell.

12.3.3 Bandwidth

Bandwidth is the most important choice in kernel density estimation (Figure 12.7):

- A **small bandwidth** lets each point influence only a short distance. It shows local hot spots, but if it is too small, it produces a spiky surface of isolated peaks, one for each point.
- A **large bandwidth** spreads each point's influence widely. It shows the broad regional pattern, but if it is too large, it produces one smooth blob that hides the hot spots.

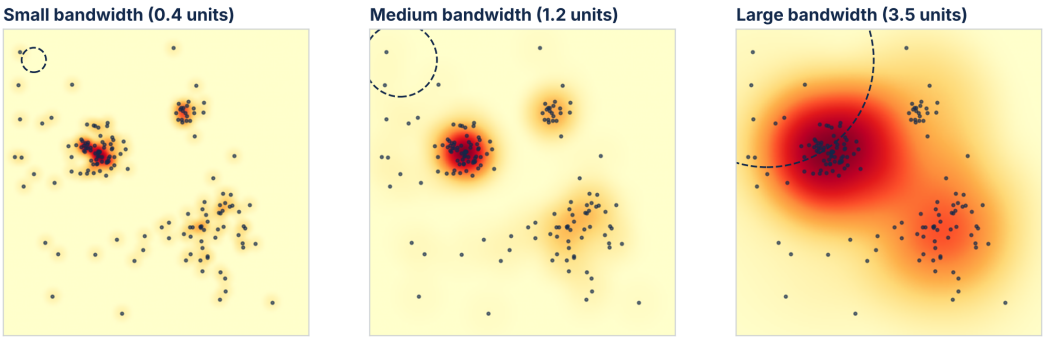


Figure 12.7: Kernel density surfaces of the same made-up incident points with three bandwidths. The dashed circle in the upper-left corner shows the size of the bandwidth. Darker red means a higher density.

A good bandwidth matches the scale of the process you are studying: a few blocks for street crime, several miles for wildfire ignitions. Try several, and report which you used.

12.3.4 Cell size and units

The **cell size** of the output raster controls how fine the surface looks, but not the pattern. Smaller cells give a smoother-looking surface and a larger file. Because the cell size and the search radius are both distances, **project your data first.** If the data are in a geographic coordinate system, both are measured in degrees, whose length changes with latitude (Chapter 3).

Did you know..?

Kernel density surfaces are often used to direct police patrols to crime "hot spots." The method is powerful, but it can also mislead. A surface built from *reported* crimes shows where crimes were reported and recorded, which partly reflects where police were already patrolling. Sending more patrols there produces more reports, which can make the hot spot look hotter: a feedback loop. Researchers and police departments now look for ways to check hot-spot maps against other data.

12.4 Kernel density in a GIS

12.4.1 ArcGIS Pro

The **Kernel Density** tool (*Spatial Analyst* toolbox, *Density* toolset) needs a Spatial Analyst license.



1. **Project your data** to a projected coordinate system, such as NC State Plane (EPSG:32119).
2. Open **Kernel Density**, and set the **Input point or polyline features**. (It takes points or lines, not polygons.)
3. **Population field** (optional): a field that makes one feature count more than once, for example a point that stands for an apartment building with 12 units. Use *NONE* to count each feature once.
4. **Output cell size**: the size of each raster cell, in map units.
5. **Search radius**: the bandwidth, in map units. If you leave it blank, the tool estimates one from the spread of the points, with a version of Silverman's rule of thumb.
6. **Area units**: the units of the output, such as square miles.
7. Run the tool, and symbolize the output raster with a color ramp. Compare a few search radii before you choose one.

ArcGIS Pro's **Point Density** tool is the simpler, uniform-kernel version: it counts the points in a neighborhood (a circle, rectangle, ring, or wedge) around each cell and divides by its area.

12.4.2 QGIS

The **Heatmap (Kernel Density Estimation)** tool is in the *Processing Toolbox*, under *Interpolation*.

1. Project the points, as above.
2. Set the **Point layer** and the **Radius** (the bandwidth), in map units.
3. Set the **Output raster size** (pixel size).
4. Optionally, choose a **Weight from field**, and a **Kernel shape** (quartic is the default; the others are triangular, uniform, triweight, and Epanechnikov).
5. Run the tool, and symbolize the output with *Singleband pseudocolor*.

For a quick look, without creating a new raster, set the layer's symbology to the **Heatmap** renderer, which draws a kernel density surface on the fly.

12.5 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

12.6 Summary

- **Density** is a count divided by an area. The area you divide by, such as total or land area, changes the answer.
- **Tobler's first law** says near things are more related than distant things. **Spatial autocorrelation** measures how much nearby values are alike.
- A **neighborhood** is a zone of analysis around a location, defined by **distance** (everything within *d*), **contiguity** (polygons that touch, in **rook's** or **queen's** case), or **number of neighbors** (the *k* nearest). Match the definition to the process you are studying.

- Distance neighborhoods suffer from the **edge effect** and leave isolated points without neighbors; contiguity struggles with islands and odd shapes; *k* nearest neighbors may reach far away.
- **Kernel density estimation** slides a **kernel** across a grid, weights the points inside by their distance, and divides by the kernel's area, producing a continuous density surface.
- **Bandwidth** matters far more than kernel shape: small bandwidths show local hot spots, large ones the regional pattern. Project your data so that cell size and bandwidth are in meters or feet.

12.7 Exercises

1. **Which area?** Look up the population, total area, and land area of Dare County and of Orange County, North Carolina. Compute the density of each with both areas. Why is the difference so much larger for Dare County?
2. **Neighbors by hand.** Draw a 4 by 4 grid of cells. How many rook's case neighbors does a corner cell have? An edge cell? An interior cell? Repeat for queen's case.
3. **County neighbors.** In QGIS or ArcGIS Pro, select the counties that touch Wake County with *Select by Location* (*touching the boundary of*, Chapter 7). How many first-order neighbors does it have? Which are its second-order neighbors?
4. **Kernel density.** Download a set of points for Chapel Hill or another town, such as crime incidents or traffic crashes from the town's open data portal. Project them, and make kernel density surfaces with three search radii: 250, 750, and 2,000 feet. Describe how the hot spots change.
5. **Kernels and counts.** Run ArcGIS Pro's Point Density tool and its Kernel Density tool on the same points with the same radius. Where do the two surfaces differ most, and why?
6. **Hot spots and fairness.** Read about the debate over predictive policing. Write a paragraph on how a hot-spot map could reinforce existing patterns of policing, and what you would check before using one.

12.8 Further reading and online resources

12.8.1 Books and articles

- Tobler, "A Computer Movie Simulating Urban Growth in the Detroit Region" (Tobler 1970), the source of the first law.
- Silverman, *Density Estimation for Statistics and Data Analysis* (Silverman 1986), the classic text on kernel density estimation.
- *Geographic Information Analysis*, by David O'Sullivan and David Unwin (O'Sullivan and Unwin 2010), on point patterns, neighborhoods, and density.
- *GIS Fundamentals: A First Text on Geographic Information Systems*, by Paul Bolstad (Bolstad 2022), on neighborhood operations and density.
- *Intro to GIS and Spatial Analysis*, by Manuel Gimond, free online, with chapters on point pattern analysis and spatial autocorrelation: https://mgimond.github.io/Spatial/.

12.8.2 Online resources

- Esri, "How Kernel Density works": https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/how-kernel-density-works.htm.
- Esri, "Kernel Density" tool reference: https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/kernel-density.htm.
- Esri, "Point Density" tool reference: https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/point-density.htm.
- QGIS User Guide, "Heatmap (Kernel Density Estimation)": https://docs.qgis.org/latest/en/docs/user_manual/processing_algs/qgis/interpolation.html.
- QGIS Tutorials, "Creating Heatmaps": https://www.qgistutorials.com/en/docs/3/creating_heatmaps.html.
- GeoDa, free software for exploring spatial data, neighborhoods (spatial weights), and spatial autocorrelation: https://geodacenter.github.io.
- GeoDa workbook, "Contiguity-Based Spatial Weights": https://geodacenter.github.io/workbook/4a_contig_weights/lab4a.html.
- The GIS&T Body of Knowledge, "Kernels and Density Estimation," "Spatial Autocorrelation," and "Neighborhoods": https://www.ucgis.org/site/gis-t-body-of-knowledge.
- Town of Chapel Hill open data portal, with police incident data: https://opendata.townofchapelhill.org.
- Penn State, *GEOG 586: Geographic Information Analysis*, on point pattern analysis and density: https://courses.ems.psu.edu/geog586/.

13 Spatial Analysis: Selection, Buffers, and Overlay

Learning objectives

By the end of this chapter you should be able to:

1. Define spatial analysis, and break a problem into a chain of spatial operations, as a cartographic model.
2. Tell local, neighborhood, and global operations apart by their spatial scope.
3. Use selection, joins, reclassification, and dissolve to prepare layers for analysis.
4. Create fixed, dissolved, multiple-ring, and variable-distance buffers, and spot the problems they can cause.
5. Explain what overlay does to geometry and attributes, and choose between clip, intersect, and union.
6. Recognize sliver polygons, and describe how map algebra performs overlay on rasters.

So far, this book has been about getting data into a GIS and mapping it well. This chapter turns to answering questions with it. Most GIS questions, from "which homes are in the flood zone?" to "where should the new clinic go?", are answered with a small set of tools used over and over: selection, joins, reclassification, dissolve, buffers, and overlay. The art is in chaining them together.

13.1 What is spatial analysis?

13.1.1 A problem: where would Batman live?

Start with a light problem. Suppose Batman is moving to Austin, Texas. He has two requirements:

- **Limestone soils**, soft enough to dig a Batcave, and prone to forming caves on their own.
- **A high-crime area nearby**, so that he can fight crime.

Austin is a good choice. It sits on the Balcones Escarpment, where limestone is common, and it is home to one of the largest urban bat colonies in North America. To find the best site, we need two data layers, soils and ZIP Codes, and a table of crimes by ZIP Code. Then we solve the problem in four steps (Figure 13.1):

1. **Select** the limestone soils from the soils layer, to create a new layer.
2. **Table join** the crime table to the ZIP Code layer, on the ZIP Code (Chapter 6).
3. **Select** the ZIP Code with the most crimes, to create a new layer.

4. **Spatial join** the limestone layer to the highest-crime ZIP Code, to find the closest limestone area (Chapter 7).

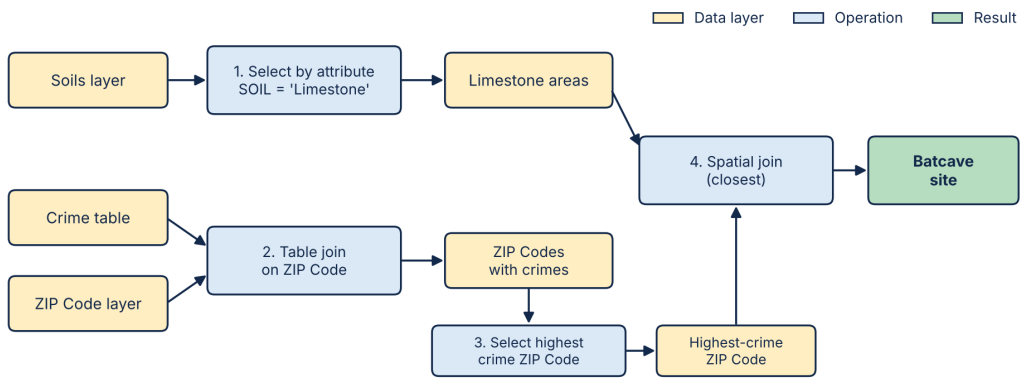


Figure 13.1: The Batcave problem as a cartographic model. Each operation's output becomes the next operation's input.

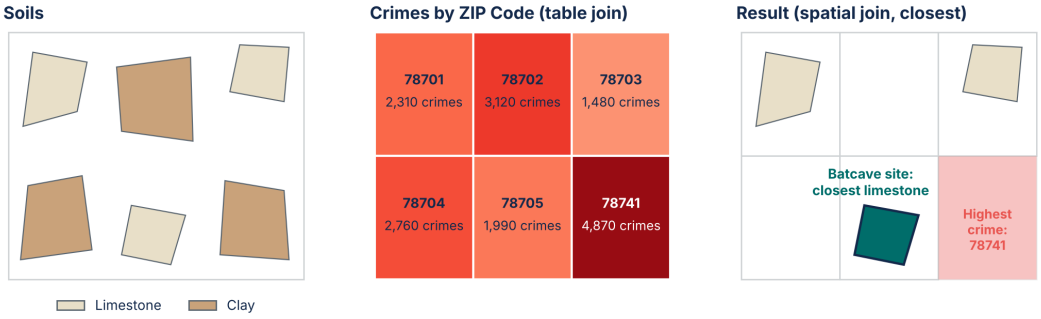


Figure 13.2: The Batcave problem in a made-up city: soils (left), crimes by ZIP Code after the table join (middle), and the closest limestone area to the highest-crime ZIP Code (right).

The answer depends on our choices. If we used crimes *per person* rather than total crimes (Chapter 11), a different ZIP Code might win, and so might a different site. The same workflow, with different criteria, is used to site real facilities, such as schools, clinics, and landfills.

Did you know..?

Austin really is a good place for bats. From March to October, about 1.5 million Mexican free-tailed bats live under the Congress Avenue Bridge downtown, and at dusk they stream out in a cloud to hunt insects. The colony moved in after the bridge was rebuilt in 1980 with narrow crevices underneath, just the right size for roosting. Crowds now gather every summer evening to watch.

13.1.2 Spatial operations and cartographic models

Bolstad defines **spatial analysis** as the "application of operations to coordinate data and related attribute data, often to solve a problem" (Bolstad 2022). Two things stand out in that definition. Operations work on both the geometry (*coordinate data*) and the table (*attribute data*), and they usually change both. And analysis is driven by a *problem*, not by the tools.

A **spatial operation**, also called a spatial function, takes one or more input layers and produces an output layer. Real problems are solved by chaining operations, each output becoming the next input. Planning such a chain is called **cartographic modeling**, a term introduced by Dana Tomlin (Tomlin 1990). A flowchart like Figure 13.1 is the plan: the inputs, the operations, the intermediate layers, and the final output.

Sketch the model before you open the software. Then build it in a tool such as ArcGIS Pro's **ModelBuilder** or QGIS's **Model Designer**, which let you draw the flowchart and run it. A model documents every step, and can be rerun when the data change.

13.2 Spatial scope

Spatial scope is "the extent or area of the input data that are used in determining the values at output locations" (Bolstad 2022). It tells you which input locations feed each output value. There are three kinds (Figure 13.3):

- **Local** operations use only the input at the same location. Utah's population density uses only Utah's population and area; its neighbors' values don't matter.
- **Neighborhood** operations use the input at a location and at nearby locations. Counting the states that border Utah is a neighborhood operation: the answer, 6, depends on which states touch it. Buffers and moving windows are neighborhood operations.
- **Global** operations use every location in the layer. Ranking the states by population is global: Utah's rank depends on every other state's value. Changing any one input can change every output.

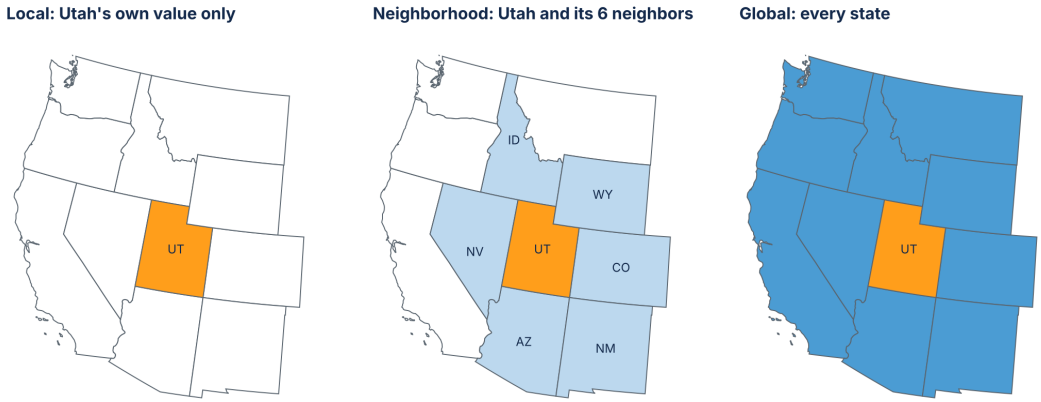


Figure 13.3: Spatial scope, with Utah as the example location. Shaded states are the inputs used to compute Utah's output value.

Utah's six neighbors include New Mexico, which it touches only at the Four Corners. Counted in rook's case (Chapter 12), Utah would have five. The same three scopes appear in raster analysis as local, focal, and global functions (Section 13.6).

13.3 Preparing layers

13.3.1 Selection

Selection is usually the first step. An **attribute query** selects features by their values in the table, such as `SOIL = 'Limestone'` (Chapter 5). A **spatial query** selects them by location,

such as the ZIP Codes that touch a river (Chapter 7). Both combine conditions with `AND`, `OR`, and `NOT`. Export the selection to a new layer to feed the next operation.

13.3.2 Joins

Joins attach attributes from one dataset to another. A **table join** links records through a shared key, such as a ZIP Code (Chapter 6). A **spatial join** links them by location, such as *contains*, *intersects*, or *closest*, and works when the layers share no key (Chapter 7). The Batcave problem uses both.

13.3.3 Reclassification

Reclassification groups values into fewer, more meaningful classes, such as elevation into *low* (below 500 m), *medium*, and *high* (above 1,500 m). It is a local operation: each feature's new class depends only on its own value. In suitability models, each input layer is usually reclassified onto a common scale, such as 1 (poor) to 5 (good), before the layers are combined.

13.3.4 Dissolve

Dissolve merges features that share a value in a field, removing the boundaries between them (Figure 13.4). Dissolving U.S. states on their Census region turns 49 polygons into 4. Dissolve is how you aggregate to a coarser unit: counties to states, or parcels to zoning districts.

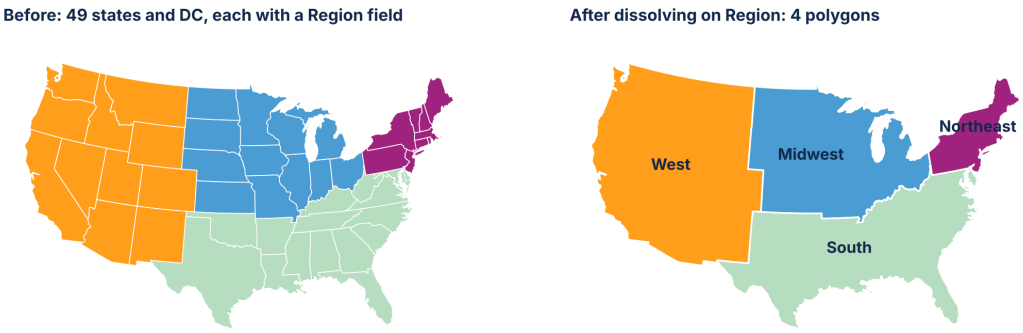


Figure 13.4: The lower 48 states and DC, before and after dissolving on a Region field.

The benefit is a simpler map and a smaller dataset; the cost is the detail of the merged features. When you dissolve, you can keep **summary statistics**, such as the sum of population, so that the new polygons carry totals of the old ones.

13.4 Buffers

13.4.1 Proximity

Proximity functions work with distance. The main one is the **buffer**: the region within a specified distance of a feature. You can buffer points, lines, or polygons, to answer questions such as "which homes are within 500 meters of a highway?"

There are several kinds of buffer (Figure 13.5):

- **Fixed buffers** use one distance for every feature. Where buffers overlap, they stay separate polygons, so an area near two features is covered twice.
- **Dissolved buffers** merge overlapping buffers into one region. Use them when you want the total area within the distance, and don't care which feature each part came from.
- **Multiple-ring** (nested) buffers create rings at several distances, such as 1, 2, and 3 kilometers, for questions about distance bands.
- **Variable-distance buffers** take each feature's distance from a field, so that larger rivers or busier roads get wider buffers (Figure 13.6, right).

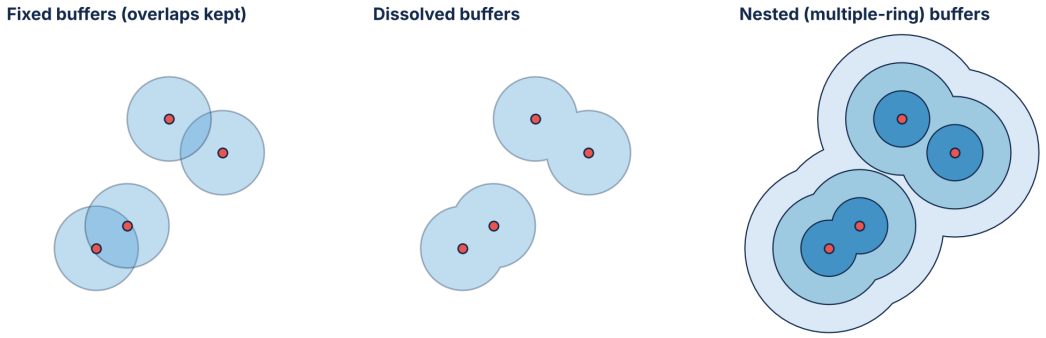


Figure 13.5: Fixed, dissolved, and nested buffers around the same four points. The nested buffers are 1, 2, and 3 km.

13.4.2 Line and polygon buffers

A line buffer extends the same distance on both sides of the line. Where lines loop back or run close together, their buffers merge, and the merged buffer can surround an area that is farther than the buffer distance from every line (Figure 13.6, left). This hole is *not* in the buffer. Polygons can also be buffered outward or inward (a negative distance). Check the output for holes and islands before using it in the next step.

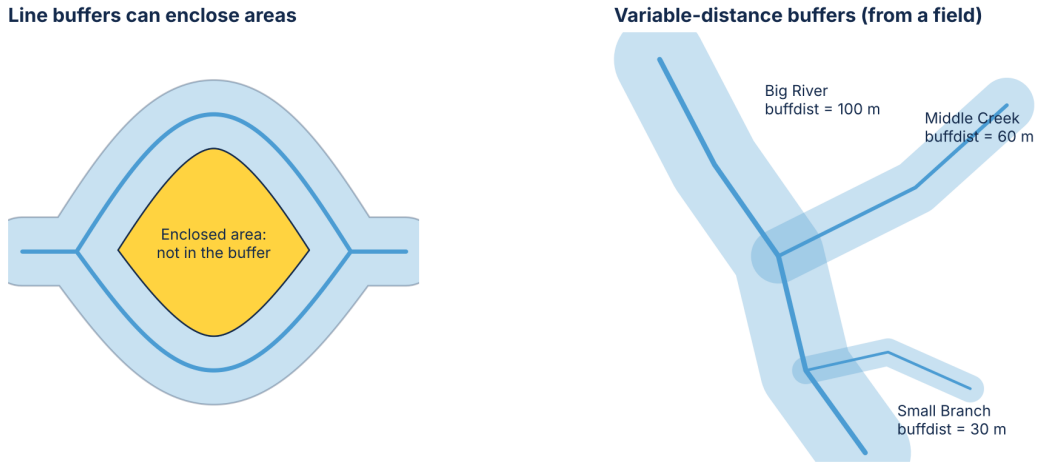


Figure 13.6: Left: two branches of a stream whose buffers enclose an area that is not in the buffer. Right: variable-distance buffers, with each river's distance read from a field.

Buffers must be built in a **projected coordinate system**, or with a *geodesic* option, so that the distance is in real meters or feet (Chapters 4 and 7).

> **Think about it**
>
> A city requires a 30-meter vegetated buffer along every stream, and 60 meters along rivers. A developer's parcel has a stream on two sides. How would you find the buildable area of the parcel? Which buffer type, and which overlay operation, would you use?

13.5 Overlay

13.5.1 Combining layers by location

Overlay combines two or more layers at the same locations. Boundaries from each layer are stacked, and wherever they cross, polygons are split. Each output polygon carries the attributes of every input polygon it came from (Figure 13.7). So overlay changes both the geometry and the table.

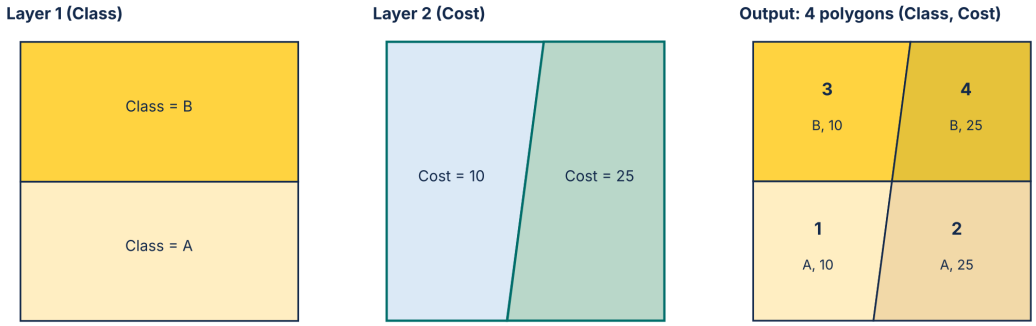


Figure 13.7: Polygon overlay. Two layers with two polygons each produce four output polygons, each with a Class from layer 1 and a Cost from layer 2.

The layers **must share a coordinate system**, or their features won't line up. (ArcGIS Pro and QGIS will project on the fly for display, but the tools may still give unexpected results; reproject first.)

The type of the output decides what can be kept. In a **point-in-polygon** overlay, each point inherits the attributes of the polygon it falls in, such as each crime point gaining the name and type of its county (Figure 13.8). There is no conflict, because each point is in one county. Going the other way, from points to a polygon, the polygon can hold only one value per field, so you need a rule: a count, a sum, the majority, or the first.

13.5.2 Clip, intersect, and union

Three overlay operations are used most (Figure 13.9):

- **Clip** keeps only the parts of the data layer that fall inside the bounding layer, like a cookie cutter. The output has only the data layer's attributes. Use it to cut statewide data down to a county or study area.

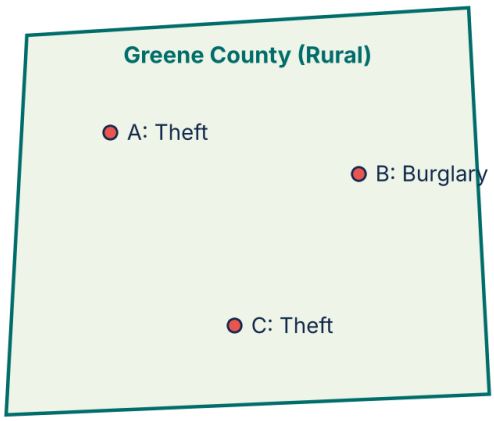


Figure 13.8: Point-in-polygon overlay. Each crime point keeps its own type and gains the county's name and type.

- **Intersect** keeps only the areas covered by *both* layers, and the output has attributes from both. Swapping the inputs gives the same areas. Intersect is the overlay version of a Boolean `AND`. For example, intersecting road centerlines with the 100-year floodplain gives the road segments at risk of flooding, with the road's name and the flood zone in each record.
- **Union** keeps *all* areas from both layers, split where they overlap, with attributes from both. Where one layer is absent, its fields are blank. Union is the overlay version of `OR`. In ArcGIS Pro, both inputs must be polygons.

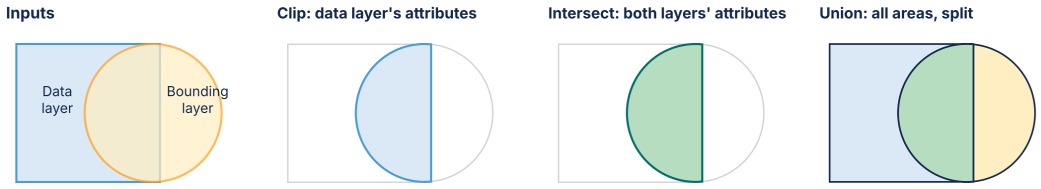


Figure 13.9: Clip, intersect, and union of the same data layer (blue) and bounding layer (orange).

Table 13.1: Clip, intersect, and union.

Operation	Keeps	Attributes from	Like
Clip	The data layer inside the bounding layer	The data layer only	A cookie cutter
Intersect	Areas in both layers	Both layers	`AND`
Union	All areas in either layer	Both layers (blank where absent)	`OR`

Other overlay tools include **Erase** (keep the parts of the data layer *outside* the other layer) and **Identity** (keep all of the input layer, with the attributes of the other layer where they overlap).

13.5.3 Slivers

Vector overlay has a well-known problem. When two layers contain the same boundary, such as a county line, digitized at different times or scales, the two versions never match exactly. Overlay splits polygons wherever they cross, so the result is full of **slivers**: thin, tiny polygons along the shared boundary that represent nothing real (Figure 13.10).



Figure 13.10: Slivers. The same boundary, digitized twice (left), produces many thin polygons after overlay (right).

Fixes include:

- setting a **snapping** or **cluster tolerance** when you run the overlay, so that vertices closer than the tolerance are treated as the same;
- using the **same source** for shared boundaries in every layer;
- removing polygons below a minimum area afterward, for example with ArcGIS Pro's **Eliminate** tool, which merges each sliver into its neighbor.

13.6 Raster overlay: map algebra

Rasters can be overlaid too, and more simply. Because every input uses the same grid of cells, overlay is done **cell by cell**, with **map algebra**, a framework also developed by Tomlin (Tomlin 1990). Its functions follow the same scopes as vector operations (Figure 13.11):

- **Local** functions use the same cell in each input, such as adding a slope score and a soil score.
- **Focal** functions use a moving window around each cell, such as the mean of a 3×3 window.
- **Zonal** functions summarize the cells within each zone, such as the mean elevation of each watershed.
- **Global** functions use every cell, such as the distance from each cell to the nearest road.

Raster overlay produces **no slivers**, since every layer shares one grid. The price is precision: no boundary can be more precise than the cell size. Chapter 14 covers raster analysis in detail.

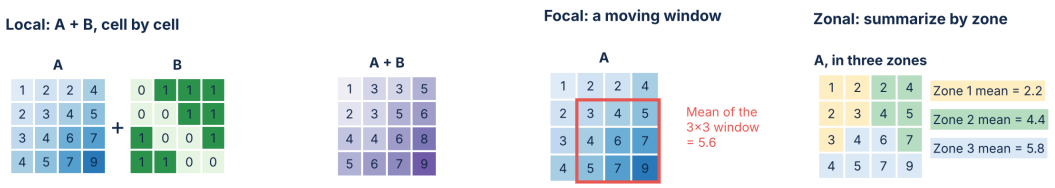


Figure 13.11: Map algebra on a made-up 4×4 raster: local, focal, and zonal functions.

13.7 Other kinds of spatial analysis

Overlay and proximity are only part of the toolkit. Other families of methods include:

- **Network analysis**: shortest routes, closest facilities, and service areas along roads or utility lines.
- **Interpolation**: estimating values between sample points, such as temperatures from weather stations (Chapters 15 and 16).
- **Spatial statistics**: testing whether patterns, such as clusters of disease, are more concentrated than chance would produce, with measures such as Moran's I and Getis-Ord Gi*.
- **Terrain analysis**: deriving slope, aspect, and viewsheds from elevation data (Chapter 14).

13.8 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

13.9 Summary

- **Spatial analysis** applies operations to geometry and attributes to solve a problem. Each **spatial operation** turns inputs into an output; a **cartographic model** chains them, and tools such as ModelBuilder make it repeatable.
- **Spatial scope** is **local** (the same location), **neighborhood** (the location and nearby ones), or **global** (every location).
- **Selection**, **joins**, **reclassification**, and **dissolve** prepare layers for analysis. Dissolve merges features with a shared value, and can carry summary statistics.
- A **buffer** is the area within a distance of a feature. Buffers can be **fixed**, **dissolved**, **multiple-ring**, or **variable-distance**; line buffers can enclose areas outside the buffer.
- **Overlay** splits polygons where layers cross and combines their attributes. **Clip** cuts like a cookie cutter, **intersect** keeps what is in both (`AND`), and **union** keeps everything (`OR`). Layers must share a coordinate system.
- Inconsistent shared boundaries produce **slivers**; snapping tolerances, common sources, and Eliminate fix them.
- **Map algebra** overlays rasters cell by cell with **local**, **focal**, **zonal**, and **global** functions, with no slivers, but no more precision than the cell size.

13.10 Exercises

1. **Draw a model.** Sketch a cartographic model for this problem: find the parcels in Orange County, NC, that are larger than 5 acres, within 1 mile of a major road, and outside the 100-year floodplain. Name the data layers, each operation, and each intermediate layer.
2. **Scope.** Classify each operation as local, neighborhood, or global: (a) converting elevation from feet to meters; (b) finding each county's highest-population neighbor; (c) computing each county's share of the state's population; (d) the average slope in a 3×3 window.
3. **Buffers.** In QGIS or ArcGIS Pro, buffer the schools in a county by 1 mile, once without dissolving and once with. Compare the number of polygons and the total area. Which would you use to find the share of the county within a mile of a school, and why?
4. **Clip, intersect, union.** Download a county boundary and the FEMA flood zones (from the FEMA Flood Map Service Center or NC Flood Risk Information System) for part of it. Clip, intersect, and union them, and compare the attribute tables of the three outputs.
5. **Roads at risk.** Intersect a road layer with the 100-year floodplain. What is the total length of road in the floodplain? Which road segments are longest?
6. **Slivers.** Overlay two versions of a county boundary from different sources (for example, TIGER/Line and NC OneMap). How many slivers do you get? Try again with a cluster tolerance of 5 meters.

13.11 Further reading and online resources

13.11.1 Books

- *GIS Fundamentals: A First Text on Geographic Information Systems*, by Paul Bolstad (Bolstad 2022). The chapter on basic spatial analysis covers selection, dissolve, buffers, and overlay.
- *Geographic Information Systems and Cartographic Modeling*, by C. Dana Tomlin (Tomlin 1990), the original account of map algebra.
- *The Esri Guide to GIS Analysis, Volume 1: Geographic Patterns and Relationships*, by Andy Mitchell (Mitchell 2020), a clear, practical guide to overlay and proximity.
- *Geospatial Analysis: A Comprehensive Guide*, by Michael de Smith, Michael Goodchild, and Paul Longley, free online: https://www.spatialanalysisonline.com.
- *Intro to GIS and Spatial Analysis*, by Manuel Gimond, free online: https://mgimond.github.io/Spatial/.

13.11.2 ArcGIS Pro

- Esri, "An overview of the Overlay toolset": https://pro.arcgis.com/en/pro-app/latest/tool-reference/analysis/an-overview-of-the-overlay-toolset.htm.
- Esri, "Buffer" tool reference: https://pro.arcgis.com/en/pro-app/latest/tool-reference/analysis/buffer.htm.
- Esri, "Multiple Ring Buffer" tool reference: https://pro.arcgis.com/en/pro-app/latest/tool-reference/analysis/multiple-ring-buffer.htm.
- Esri, "Dissolve" tool reference: https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/dissolve.htm.



- Esri, “What is ModelBuilder?”: https://pro.arcgis.com/en/pro-app/latest/help/analysis/geoprocessing/modelbuilder/what-is-modelbuilder-.htm.

13.11.3 QGIS

- QGIS User Guide, “Vector overlay” algorithms (Clip, Intersection, Union, Difference): https://docs.qgis.org/latest/en/docs/user_manual/processing_algs/qgis/vectoroverlay.html.
- QGIS User Guide, “Vector geometry” algorithms (Buffer, Dissolve, Multi-ring buffer): https://docs.qgis.org/latest/en/docs/user_manual/processing_algs/qgis/vectorgeometry.html.
- QGIS User Guide, “The graphical modeler”: https://docs.qgis.org/latest/en/docs/user_manual/processing/modeler.html.
- QGIS Tutorials, “Performing Spatial Queries” and other analysis tutorials: https://www.qgistutorials.com/en/.

13.11.4 Other resources

- The GIS&T Body of Knowledge, “Overlay & Combination Operations,” “Buffers,” and “Cartographic Modeling”: https://www.ucgis.org/site/gis-t-body-of-knowledge.
- Bat Conservation International: https://www.batcon.org.

14 Raster Analysis

Learning objectives

By the end of this chapter you should be able to:

1. Describe a raster by its cells, cell size, extent, and NoData values.
2. Combine rasters with map algebra, and predict how NoData passes through.
3. Line up rasters before analysis by resampling, and by setting the cell size, extent, and snap raster.
4. Classify a raster function by its scope: local, neighborhood, zonal, or global.
5. Use mathematical, logical, reclassification, and conditional functions.
6. Compute neighborhood and zonal statistics, and aggregate a raster to a coarser cell size.

Chapter 13 ended with a glimpse of raster overlay. This chapter takes rasters on their own terms. A raster stores the world as a grid, and almost every raster tool does the same thing: it repeats one operation cell by cell. What changes from tool to tool is *which* input cells feed each output cell, and *what* is done with them. Once you see that, the long list of raster tools in ArcGIS Pro and QGIS becomes much easier to understand.

14.1 Raster basics

14.1.1 Cells, cell size, and extent

A **raster** represents the world as a regular grid of **cells** (also called pixels, short for *picture elements*). Each cell holds a value, such as an elevation, a temperature, or a land-cover code. Two numbers describe the grid (Figure 14.1):

- **Cell size**, also called resolution, is the ground length of one side of a cell, such as 30 m or 1 km.
- **Extent** is the ground area the whole grid covers, from its lower-left to its upper-right corner.

The software does not store coordinates for every cell. It stores the location of one corner and the cell size, and works out any cell's position from its row and column. So a raster 100 cells wide with 30 m cells has an extent 3 km wide. It also explains why the cell size matters so much: change it, and every cell's footprint changes.

14.1.2 What's in a cell?

A cell can hold a whole number (such as a land-cover code), a decimal (such as rainfall in inches), or, in some formats, text. Most rasters hold one value per cell. **Multiband** rasters, such as satellite images, hold one value per band in each cell, for example red, green, blue, and near-infrared.

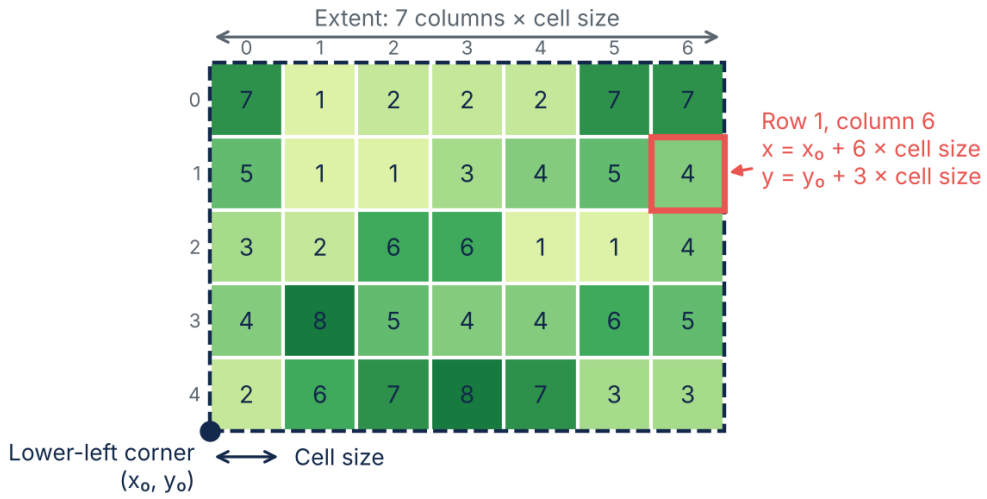


Figure 14.1: A raster is a grid of cells. The software stores the lower-left corner and the cell size, and computes every cell's location from its row and column.

Cell values are either **categorical** (also called discrete or thematic), where numbers are codes for classes such as land cover or soil type, or **continuous**, where numbers are measurements, such as elevation or temperature. Many choices in this chapter depend on which kind you have.

14.1.3 NoData

A cell is never truly empty. A cell without a measurement holds a special code, **NoData** (Null in some software). There are two rules for NoData:

1. **NoData in, NoData out.** In map algebra, any operation that touches a NoData cell writes NoData to the output: 5 + NoData = NoData.
2. **Statistics skip NoData.** The mean of 2, 4, and NoData is 3, not 2.

The difference between NoData and 0 matters. In an elevation raster, 0 means sea level; NoData means "unknown." If missing values are stored as 0, every mean that includes them is too low, and the map is wrong.

Did you know..?

The National Land Cover Database (NLCD) maps the lower 48 states with 30 m cells. Each cell covers 900 m², about one-sixth of a football field, so the lower 48's roughly 8 million km² takes about 9 billion cells. The land-cover codes used later in this chapter, such as 41 for deciduous forest and 82 for cultivated crops, are NLCD codes.

14.2 Combining rasters

14.2.1 Map algebra

Map algebra treats rasters as variables in an equation, and evaluates it cell by cell (Tomlin 1990). In Figure 14.2, each cell of raster C is the sum of the cells at the same position in A and

B. The top-left cell is 1 + (−1) = 0. The bottom-right cell of B is NoData, so the bottom-right cell of C is NoData too, whatever A holds.

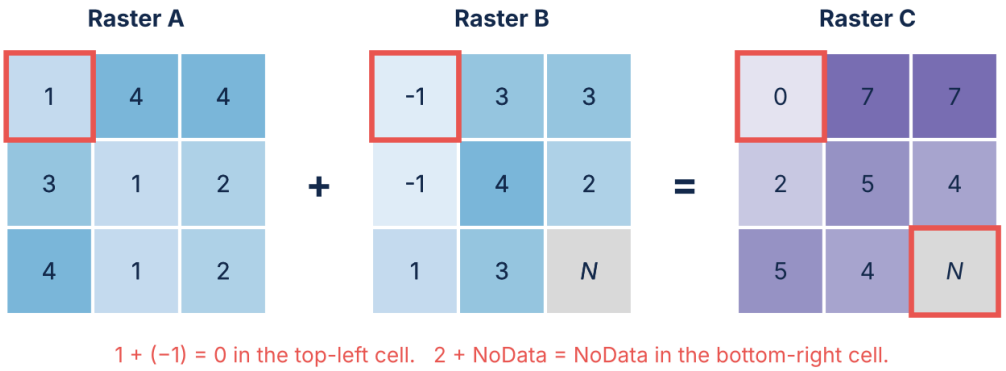


Figure 14.2: Map algebra: raster A plus raster B equals raster C, cell by cell. N marks NoData.

An operation with one input raster, such as A × 2 or the square root of A, is **unary**. One with two input rasters, such as A + B, is **binary**. (A constant such as 2 is not a raster, so it doesn't count as an input.)

14.2.2 When the grids don't line up

Map algebra assumes the rasters line up exactly: the same cell size, the same extent, and cells in the same places. Real data rarely do (Figure 14.3).

- **Mismatched cell sizes.** A coarse cell sits over parts of many fine cells. Which of them should it be added to? There is no one answer.
- **Different extents.** Outside the overlap, one input has no cells at all, so the output there can only be NoData.

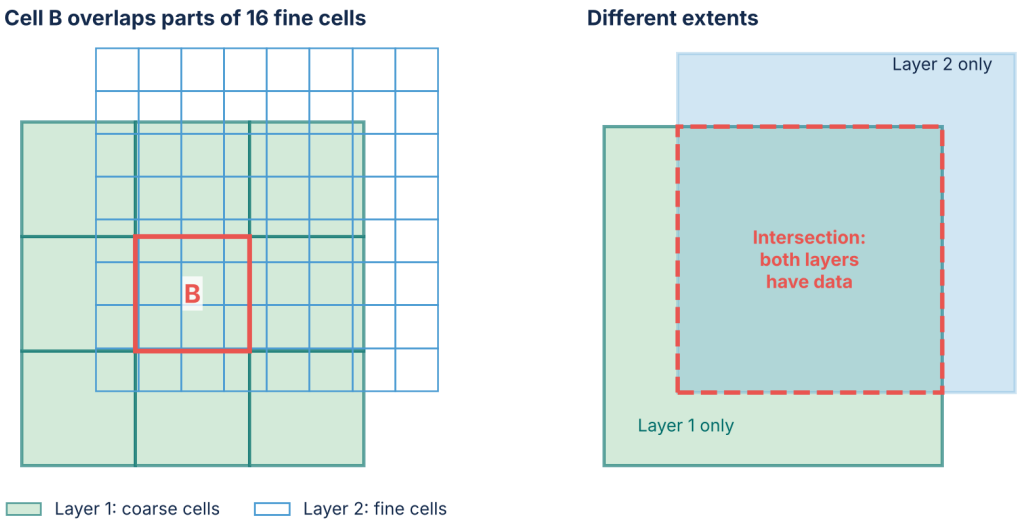


Figure 14.3: Two problems for map algebra: mismatched cell sizes (left) and different extents (right).

14.2.3 Resampling

Resampling rebuilds a raster on a new grid, so that every layer shares one cell size and one set of cell positions. For each new cell, it decides what value the cell should get. There are three standard methods (Table 14.1):

Table 14.1: Resampling methods.

Method	How it picks the new value	Use it for
Nearest neighbor	Copies the value of the closest input cell center	Categories: land cover, soils, zones
Bilinear	Distance-weighted mean of the 4 nearest cells	Continuous surfaces: elevation, temperature
Cubic convolution	A smooth curve through the 16 nearest cells	Continuous surfaces where smoothness matters

Nearest neighbor never invents values, so it is the only safe choice for categories: you can't average forest and water. Bilinear and cubic blend nearby cells, which gives smoother surfaces for continuous data, but creates values that were never measured. On a land-cover raster, they produce nonsense such as class 2.25 (Figure 14.4).

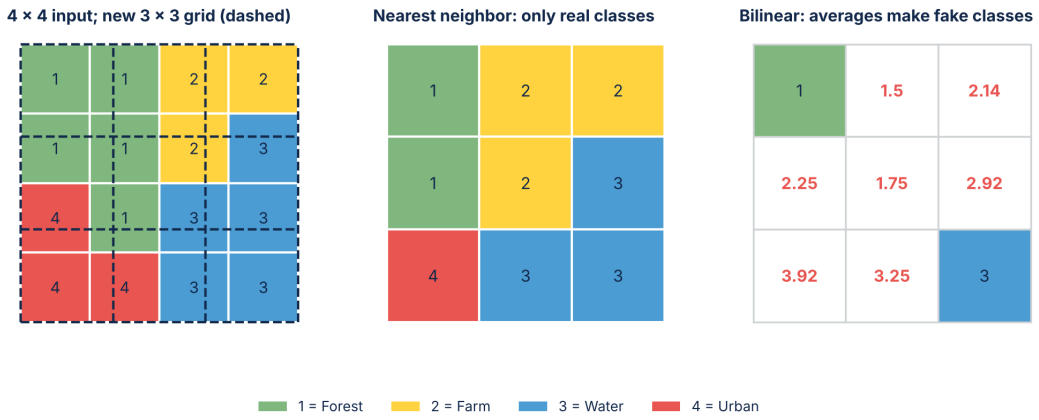


Figure 14.4: Resampling a 4 × 4 land-cover raster to a 3 × 3 grid. Nearest neighbor keeps real classes; bilinear produces in-between values that are not classes at all.

14.2.4 Cell size, extent, and snap raster

Three settings make the grids line up before any map algebra runs. In ArcGIS Pro they are **environment settings**, used by every raster tool; QGIS asks for them in each tool's dialog.

1. **Cell size.** Choose one output cell size, usually the coarsest input. Resampling to finer cells adds no real detail.
2. **Extent.** *Intersection* keeps only the area where every input has data; *union* keeps everything, with NoData in the gaps. Intersection is the safer default.
3. **Snap raster.** Even with the same cell size, two grids can be offset by part of a cell. Snapping aligns the output's cell edges to one input's grid.

The output can only be as detailed as its coarsest input (Figure 14.5). If you add a 30 m land-use raster to a 1 km elevation raster, the result is a blocky 1 km grid, and choosing 30 m output cells would only copy each 1 km value into a thousand smaller cells. If you need

30 m results, find a better elevation source, such as a 10 m or 30 m digital elevation model from USGS 3DEP.

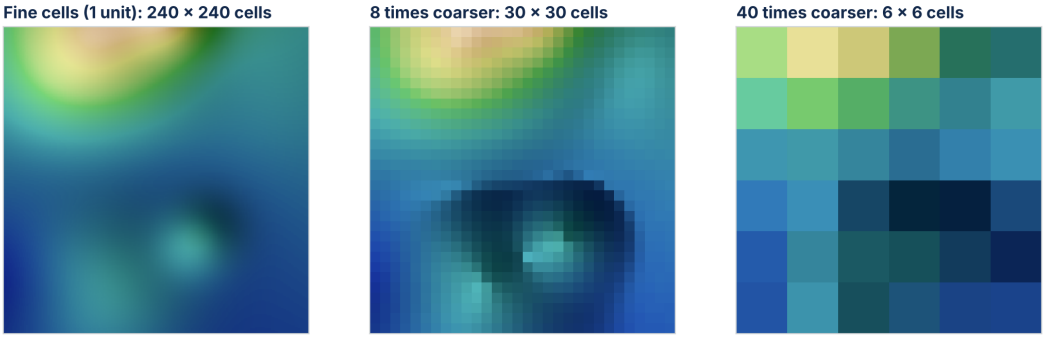


Figure 14.5: The same made-up terrain at three cell sizes. Each coarser raster averages blocks of finer cells.

14.3 The scope of raster functions

Scope is the set of input cells that feed each output cell. Chapter 13 introduced local, neighborhood, and global scope; rasters add a fourth, **zonal** (Figure 14.6):

- **Local** functions use only the input cell at the same location. Adding 4 to every cell is local: 10 becomes 14.
- **Neighborhood** (focal) functions use a window of cells around each location, usually 3 × 3. The window's result is written to its center cell, and then the window moves.
- **Zonal** functions use every cell in the same zone, where zones come from a second layer. Each cell in zone *a* gets the mean of all cells in zone *a*.
- **Global** functions can use every cell in the raster. The global maximum is written to every cell; more useful global functions, such as distance to the nearest road, give a different answer for each cell but still need to look everywhere.

Notice the edge cells in the neighborhood example: a window centered on a corner holds only four cells, so edge cells always have fewer neighbors.

> **Think about it**
>
> Is each of these local, neighborhood, zonal, or global? Converting elevation from feet to meters; smoothing a noisy elevation surface; computing the mean elevation of each county; finding the distance from every cell to the nearest hospital.

14.4 Local functions

14.4.1 Mathematical functions

Mathematical functions are the most common map algebra. You write rasters as letters in an **expression**, such as `A * B`, `(A * 2) / B`, or `A + B + C`, and the tool evaluates it cell by cell. You can also apply functions to a single raster: absolute value, exponents and logarithms, sine and cosine, rounding, square roots, and powers. In ArcGIS Pro, the **Raster Calculator** tool evaluates expressions; in QGIS, it is also called the Raster Calculator.

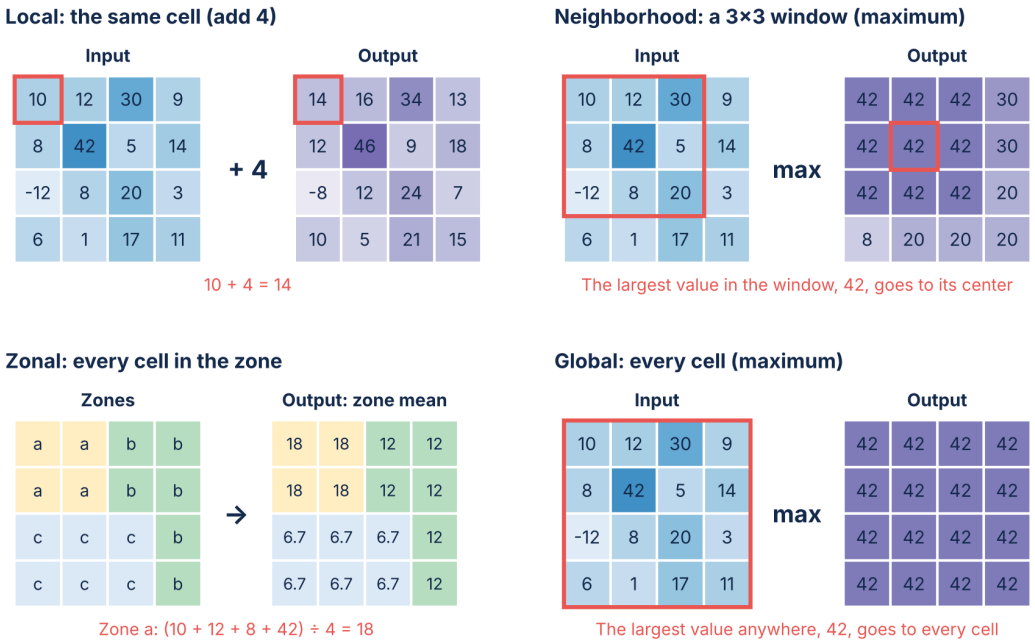


Figure 14.6: The four scopes of raster functions, on the same made-up input.

Watch two things. Trigonometric functions expect radians in most software, so convert slope or aspect in degrees first. And when you combine variables with different units, one can dominate. Multiplying elevation (100 to 1,000 m) by slope (0 to 31 degrees) gives a result driven mostly by elevation. **Rescale** both to the same range, such as 0 to 1, first (Figure 14.7).

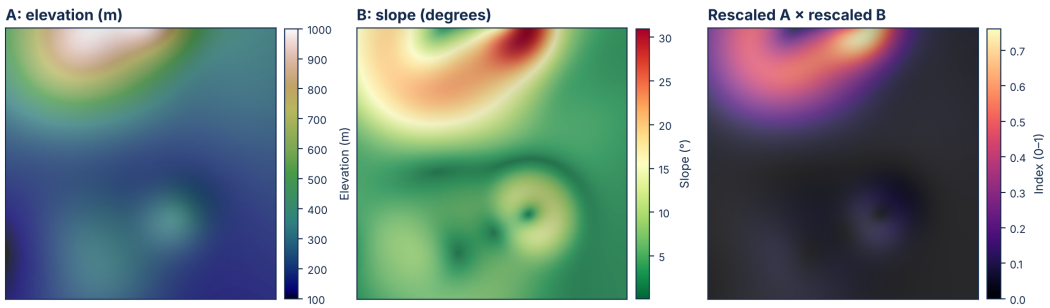


Figure 14.7: A binary local operation on made-up terrain: elevation (A) times slope (B), after rescaling each to 0–1. The result is high only where the land is both high and steep.

14.4.2 Logical functions

Relational operators compare two rasters, or a raster and a constant, cell by cell: less than, greater than, equal, and not equal. They write 1 where the comparison is true and 0 where it is false. **Boolean** operators, such as AND, OR, and NOT, combine those 0s and 1s (Figure 14.8).

Typical uses:

- Find buildable land: `Slope < 15`.
- Find land-cover change: `LC2010 != LC2020` gives 1 wherever the class changed.
- Find cells between two values: `(Slope > 5) AND (Slope < 15)`.

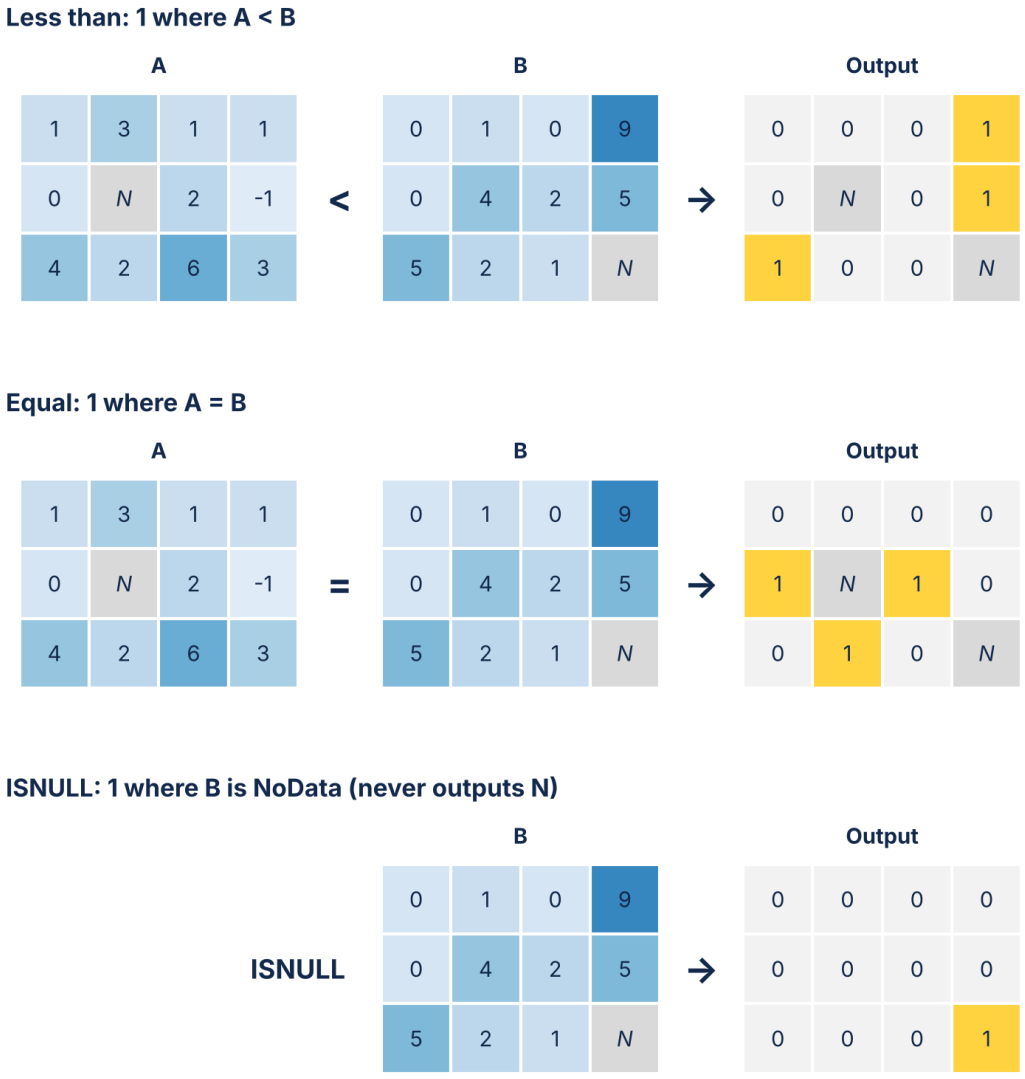


Figure 14.8: Logical functions on made-up rasters. Less than and equal output NoData wherever either input is NoData; ISNULL never does.

For continuous data, avoid testing equality: 1.30000 and 1.30001 are not equal. Test whether the difference is small instead.

ISNULL is unary: it writes 1 where a cell is NoData and 0 everywhere else. It is the one test that never outputs NoData, which makes it the tool for finding gaps. (You can't find NoData with `A = NoData`, since any comparison with NoData gives NoData.)

14.4.3 Reclassification

Reclassification replaces each cell's value with a new value from a rule you supply (Figure 14.9). It is local and unary, and one of the most used raster tools.

- **By table**, each exact value maps to a new value. Use this for categorical data, such as converting land-cover codes to a suitability score.
- **By ranges**, every value within a range maps to one new value. Use this for continuous data, such as slope.

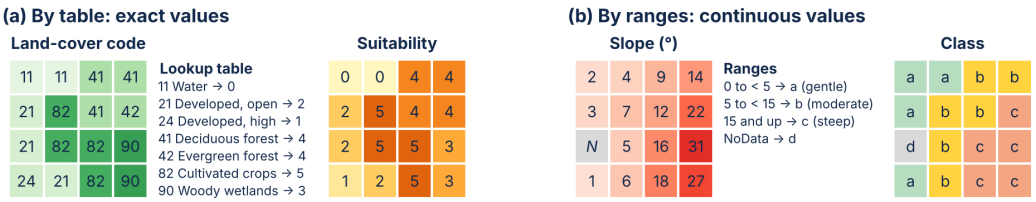


Figure 14.9: Reclassification by table (a), for categorical land-cover codes, and by ranges (b), for continuous slope. NoData can stay NoData or get its own class.

Watch the boundaries: is a slope of exactly 5 in class *a* or *b*? Software has a rule, and you should check it. Write ranges so that every value falls into exactly one class.

14.4.4 Conditional functions

The **conditional** function, CON in ArcGIS Pro, is the raster version of if-then-else. It takes a condition, a value or raster to use where the condition is true, and one to use where it is false. In Figure 14.10, the expression is `Con(A == 1, B, C)`: where A is 1, copy B; everywhere else, copy C. A decides, but its values never appear in the output.

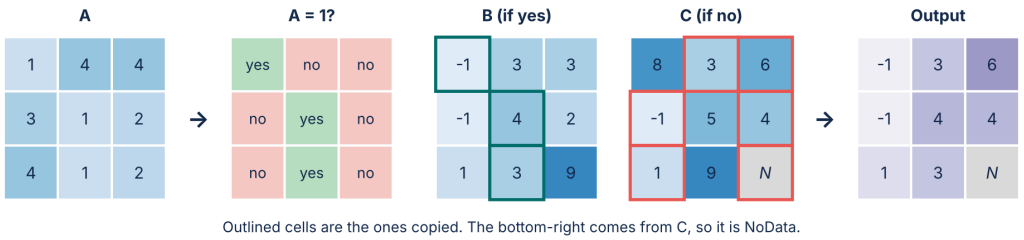


Figure 14.10: A conditional function: Con(A == 1, B, C). Outlined cells are the ones copied to the output.

The output's bottom-right cell is NoData because it comes from C, which is NoData there. B has a value in that cell, but B wasn't chosen. A common pattern fills gaps: `Con(IsNull(A), 0, A)` replaces NoData with 0 and keeps every other value.

14.5 Beyond local

14.5.1 Neighborhood functions

A **neighborhood function** computes a statistic from a window of cells and writes it to the center cell (Figure 14.11). The window is usually 3 × 3, but can be larger, a circle, or an annulus. In ArcGIS Pro, the tool is **Focal Statistics**; in QGIS, it is *r.neighbors* (from GRASS).

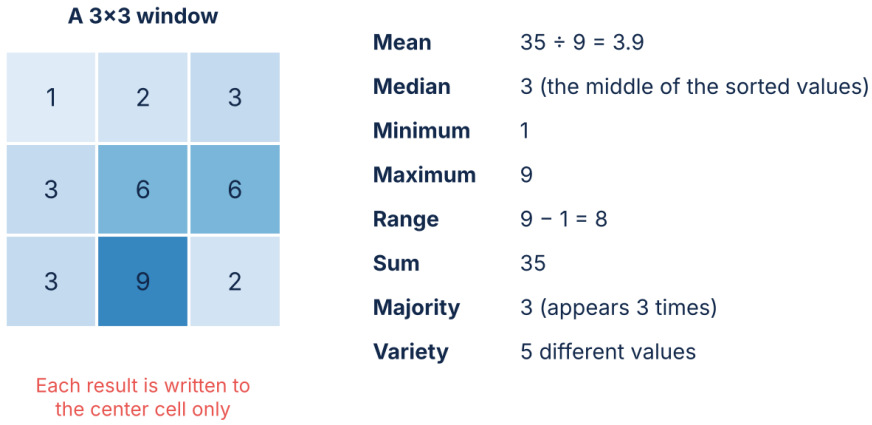


Figure 14.11: Statistics of one 3 × 3 window. Each result is written to the window's center cell.

Choose the statistic to fit the data. Mean and median smooth continuous surfaces; maximum and minimum find peaks and pits; **majority** suits categories, since a mean of land-cover codes is meaningless. Slope and aspect are neighborhood functions too: they fit a surface through the window and report how steep it is and which way it faces.

14.5.2 The moving window

Neighborhood functions run by sliding the window across the raster one cell at a time, left to right and top to bottom. A **majority filter** uses this to clean up classified satellite images, which often have single misclassified cells, called "salt and pepper" noise (Figure 14.12). Each cell takes the value shared by most of its neighbors; if there's no majority, it keeps its value.

Filters have side effects. Here, two other cells changed as well, including a water cell on the edge of a small water body. That is why filters are used with care, and checked afterward.

14.5.3 Aggregation

Aggregation (rescaling to a coarser cell size) groups blocks of cells and writes one value per block (Figure 14.13). It is the cleanest way to bring a fine raster to a coarser one's cell size. In ArcGIS Pro, the tool is **Aggregate**; in QGIS, set the resolution and a resampling method such as *average* in the **Align Rasters** tool.

Use the mean or sum for continuous data, such as rainfall or population, and the majority for categories.

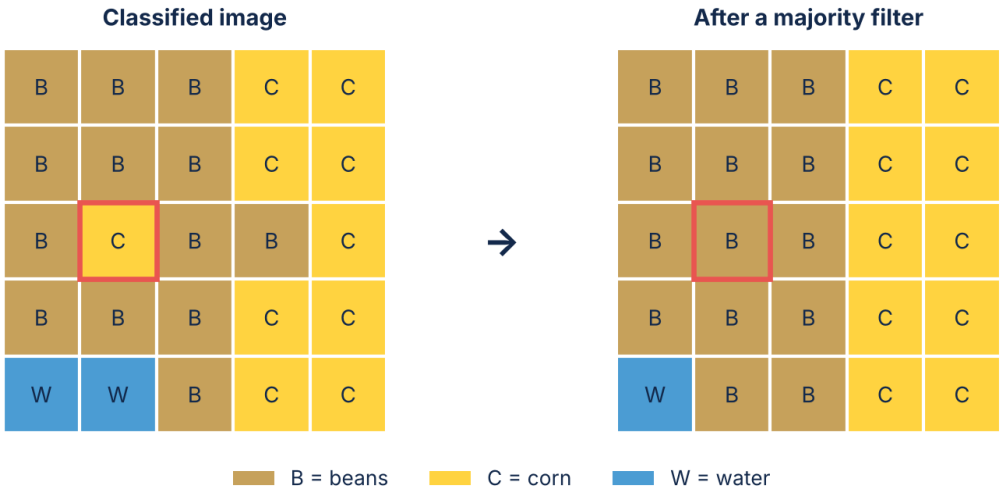


Figure 14.12: A majority filter using each cell's four rook's-case neighbors. The lone corn cell becomes beans, and two other cells change too.

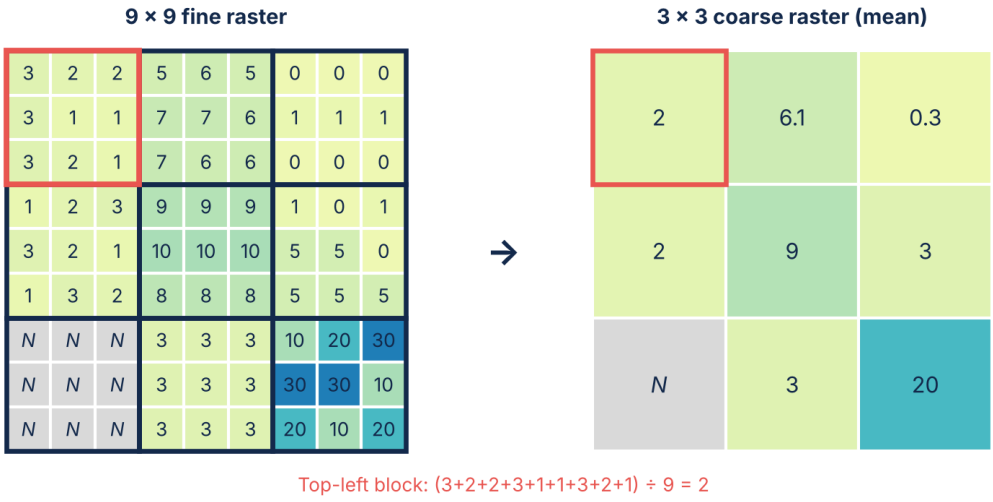


Figure 14.13: Aggregating a 9 × 9 raster to 3 × 3 with the mean of each 3 × 3 block.

14.5.4 Zonal functions

Zonal functions compute a statistic for every cell in a zone. Every neighborhood statistic (mean, sum, maximum, majority, and so on) can be computed by zone instead of by window. Zones can come from a raster, as in Figure 14.6, or from polygons, such as counties or watersheds. ArcGIS Pro's **Zonal Statistics** writes the result to a raster; **Zonal Statistics as Table** writes one row per zone. QGIS's **Zonal statistics** tool adds the results as fields in a polygon layer.

When the zones are polygons, the software converts them to cells behind the scenes. A polygon smaller than a cell may get no cells at all, and drop out of the results. Use a cell size that is small relative to your smallest polygon.

14.5.5 Global functions: distance

The most useful global functions measure **distance** (Figure 14.14). Euclidean distance gives every cell its straight-line distance to the nearest source cell, such as a river or a road. To know which source is nearest, the tool has to consider all of them, so the scope is global. **Cost distance** works the same way, but accumulates a cost, such as travel time across terrain, rather than meters.

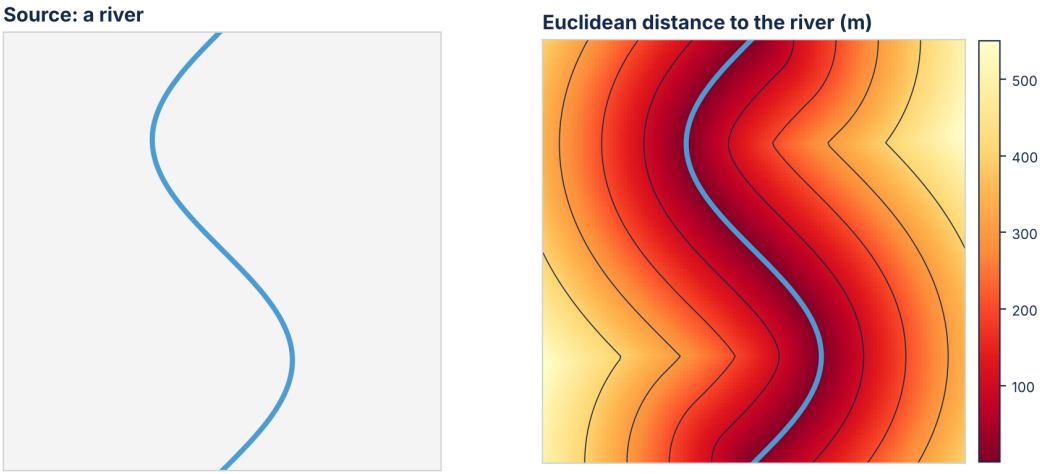


Figure 14.14: Euclidean distance from a made-up river, with contours every 100 m.

Distance rasters are often reclassified (into bands such as 0–100 m and 100–500 m) and combined with other layers in a suitability model, the raster version of the buffer and overlay workflow in Chapter 13.

14.6 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

14.7 Summary

- A **raster** is a grid of **cells**. The **cell size** and the corner of the **extent** locate every cell. Values can be **categorical** or **continuous**.



- **NoData** means "no value." It passes through map algebra (NoData in, NoData out), and statistics skip it. Don't store missing data as 0.
- **Map algebra** combines rasters cell by cell. **Unary** operations have one input raster; **binary** ones have two.
- Before combining rasters, line up the grids: **resample** to one cell size (nearest neighbor for categories, bilinear or cubic for surfaces), choose an **extent**, and set a **snap raster**. The output is only as detailed as its coarsest input.
- Raster functions have four scopes: **local** (one cell), **neighborhood** (a moving window), **zonal** (all cells in a zone), and **global** (every cell).
- Local functions include **math**, **logical** (relational and Boolean), **reclassification**, and **conditional** (CON) functions.
- **Neighborhood** functions compute statistics in a moving window, such as a **majority filter**. **Aggregation** rescales to coarser cells. **Zonal** functions summarize by zone. **Distance** is the most common global function.

14.8 Exercises

1. **Extent.** A DEM has 3,601 columns and 3,601 rows of 1-arc-second cells. About how far across is it in degrees? At the equator, about how many kilometers?
2. **Map algebra by hand.** For A and B in Figure 14.2, compute A − B and A × B. Where does NoData appear?
3. **Resampling.** In QGIS or ArcGIS Pro, download an NLCD land-cover raster for your county from the MRLC viewer, and resample it to 300 m twice, once with nearest neighbor and once with bilinear. Open the attribute tables or histograms. What happened to the classes?
4. **Suitability.** Using a DEM for Orange County, NC, from USGS 3DEP or NC OneMap, compute slope, then find the cells with a slope under 10 degrees and an elevation above 150 m. Write the expression you used.
5. **Fill the gaps.** Write a CON expression that replaces NoData in a rainfall raster with the raster's global mean, and leaves every other value unchanged.
6. **Zonal statistics.** Compute the mean elevation of each county in North Carolina with Zonal Statistics as Table. Which county is highest, and which is lowest?

14.9 Further reading and online resources

14.9.1 Books

- *GIS Fundamentals*, by Paul Bolstad (Bolstad 2022). The chapter on raster analyses covers map algebra, scope, and resampling.
- *Geographic Information Systems and Cartographic Modeling*, by C. Dana Tomlin (Tomlin 1990), the source of map algebra and its local, focal, and zonal functions.
- *GIS for Environmental Applications*, by Xuan Zhu, has clear chapters on raster analysis and suitability modeling.
- *Intro to GIS and Spatial Analysis*, by Manuel Gimond, free online, includes chapters on raster data and map algebra: https://mgimond.github.io/Spatial/.

14.9.2 ArcGIS Pro

- Esri, “Understanding Map Algebra”: https://pro.arcgis.com/en/pro-app/latest/help/analysis/spatial-analyst/mapalgebra/what-is-map-algebra.htm.
- Esri, “Raster Calculator” tool reference: https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/raster-calculator.htm.
- Esri, “Con” tool reference: https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/con-.htm.
- Esri, “Focal Statistics”: https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/focal-statistics.htm.
- Esri, “Zonal Statistics as Table”: https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/zonal-statistics-as-table.htm.
- Esri, “Cell size and resampling in analysis”: https://pro.arcgis.com/en/pro-app/latest/help/analysis/spatial-analyst/performing-analysis/cell-size-and-resampling-in-analysis.htm.

14.9.3 QGIS

- QGIS User Guide, “Raster Calculator”: https://docs.qgis.org/latest/en/docs/user_manual/working_with_raster/raster_analysis.html.
- QGIS User Guide, “Raster analysis” algorithms (Reclassify by table, Zonal statistics): https://docs.qgis.org/latest/en/docs/user_manual/processing_algs/qgis/rasteranalysis.html.
- QGIS Training Manual, “Raster analysis” lessons: https://docs.qgis.org/latest/en/docs/training_manual/rasters/index.html.

14.9.4 Data and other resources

- Multi-Resolution Land Characteristics Consortium (NLCD): https://www.mrlc.gov.
- USGS 3D Elevation Program (3DEP): https://www.usgs.gov/3d-elevation-program.
- USGS EarthExplorer, for Landsat and elevation data: https://earthexplorer.usgs.gov.
- The GIS&T Body of Knowledge, “Map Algebra” and “Raster Data Models”: https://www.ucgis.org/site/gis-t-body-of-knowledge.

15 Spatial Interpolation

Learning objectives

By the end of this chapter you should be able to:

1. Define spatial interpolation, and list the five steps every interpolation follows.
2. Tell exact interpolators from inexact ones, and deterministic methods from geostatistical ones.
3. Explain and compute nearest neighbor, average neighbor, and inverse distance weighted (IDW) estimates.
4. Describe polynomial (trend surface) and radial basis function interpolation, and their trade-offs.
5. Choose a search neighborhood and a sampling design.
6. Use cross validation and root mean square error (RMSE) to compare interpolation methods.

Weather stations measure temperature only where the stations are. Soil samples are taken at a few dozen spots in a field. Wells reveal the depth to groundwater at scattered points. Yet we want maps of temperature, soil chemistry, and groundwater *everywhere*. Spatial interpolation fills the gaps. This chapter covers the deterministic methods; Chapter 16 adds kriging, which brings in statistics.

15.1 What is spatial interpolation?

Spatial interpolation is estimating the value of an attribute at unsampled locations from a set of observations. An **observation** is a location where the attribute was measured. In Figure 15.1, four observations surround one unsampled location. Every method in this chapter answers the same question: how should those four values be combined to estimate the fifth?

The assumption behind all of it is Tobler's first law (Chapter 12): near things are more related than distant things. If values at nearby locations were unrelated, no method could estimate anything between them.

Interpolation estimates values *within* the area covered by the observations. **Extrapolation** estimates values *beyond* it, such as past the last station or above the highest observed value. Some methods can extrapolate; others can't. Most interpolation outputs a raster (Chapter 14), with an estimate in every cell.

15.1.1 Five steps

Every interpolation, whatever the method, follows five steps:

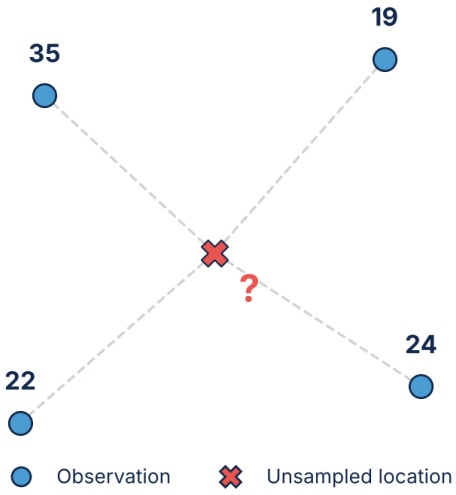


Figure 15.1: Four observations surround an unsampled location. Interpolation estimates its value.

1. **Choose a function and its parameters**: the method, such as IDW or a polynomial, and settings such as the power or the order.
2. **Define the neighborhood**: how far to search for observations, and in what shape.
3. **Find the points in the neighborhood**: which observations fall inside it. This depends on how the data were sampled.
4. **Evaluate the function** at every unknown location, usually every cell of the output raster.
5. **Estimate the error**, with cross validation and RMSE.

15.1.2 Exact and inexact interpolators

An **exact interpolator** returns the observed value at every observation: the surface passes through every point. An **inexact interpolator** smooths the data, so its surface may miss the observations (Figure 15.2). Inexact isn't necessarily worse. If the measurements contain error, forcing a surface through each one fits the noise.



Figure 15.2: An exact interpolator passes through every observation; an inexact one, here a straight trend line, does not.

The methods in this chapter are **deterministic**: the surface is fully determined by a formula and the parameters you choose. Kriging (Chapter 16) is **geostatistical**: it models the spatial autocorrelation in the data, and also estimates the uncertainty of each prediction.

15.2 Step 1: Interpolation methods

15.2.1 Nearest neighbor and Thiessen polygons

The simplest method, **nearest neighbor**, gives each location the value of the closest observation. The result is the same as drawing **Thiessen polygons** (also called Voronoi polygons): each polygon contains exactly one observation, and every location inside it is closer to that observation than to any other. The boundaries are the perpendicular bisectors between neighboring observations (Figure 15.3).

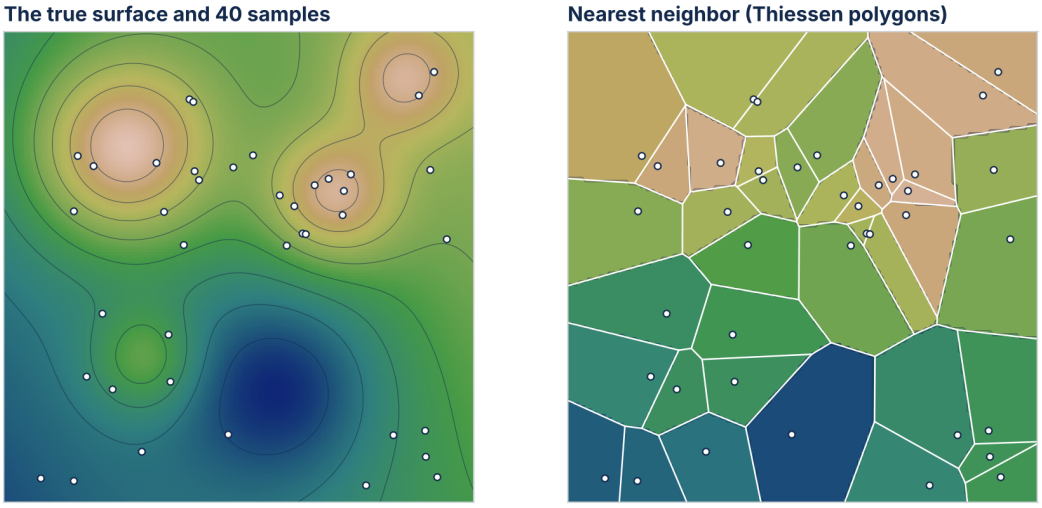


Figure 15.3: A made-up surface and 40 samples (left), and the nearest-neighbor surface, with its Thiessen polygons outlined (right).

Nearest neighbor is the right choice for **categorical** data, such as soil type or land-cover class, where averaging values would make no sense. For continuous data, its weakness is the surface itself: values jump in steps at each polygon boundary instead of changing gradually (Figure 15.9, top left).

Thiessen polygons are useful in their own right. Figure 15.4 draws them around the major airports in the lower 48 states. Many U.S. weather stations, part of the Automated Surface Observing System (ASOS), are at airports, so a map like this shows which station's reading nearest-neighbor interpolation would use where. The polygons range from about 3,000 km² in the Northeast to over 240,000 km² in the interior West. **Large polygons show where estimates are least reliable**, since they come from a station that may be hundreds of kilometers away.

Did you know..?

Thiessen polygons are named for Alfred H. Thiessen, a meteorologist with the U.S. Weather Bureau. In 1911 he published a way to estimate the average rainfall over a large area from unevenly spaced rain gauges: weight each gauge by the area of its polygon (Thiessen 1911). Hydrologists still use "Thiessen weights." Mathematicians knew the same shapes as Voronoi diagrams, after Georgy Voronoy, and they turn up

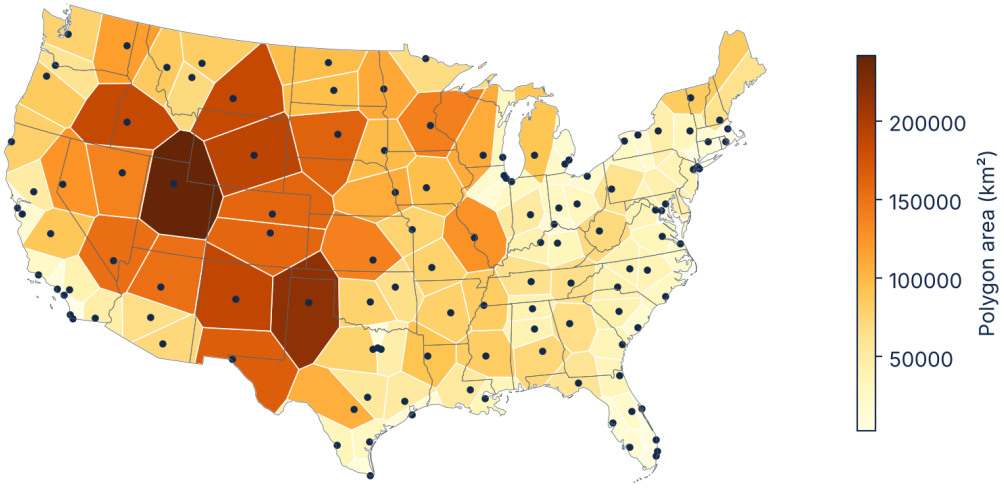


Figure 15.4: Thiessen polygons around 121 major airports in the lower 48 states, shaded by area. Data: Natural Earth.

everywhere from cell biology to cellular phone networks.

15.2.2 Average neighbor

Average neighbor interpolation draws a circle of radius *r* around each unknown location, and averages every observation inside it:

$$\hat{z} = \frac{\sum z_i}{k_r}$$

where $\hat{z}$ is the estimate, z_i are the observed values within the circle, and k_r is the number of observations found. In Figure 15.5, the circle captures four observations, 1, 2, 3, and 3, so the estimate is 9 ÷ 4 = 2.25.

It is fast and smooths the data. But the radius is a subjective choice; a point at the edge of the circle counts as much as one next door; the method is not exact; and as an average, it can never estimate a value above the highest or below the lowest observation.

15.2.3 Inverse distance weighted (IDW)

Inverse distance weighted (IDW) interpolation refines the average by giving nearer observations more weight (Shepard 1968):

$$\hat{z} = \frac{\sum z_i / d_i^p}{\sum 1 / d_i^p}$$

where d_i is the distance to observation i, and p is the **power**. The power models the *friction of distance*. With $p = 2$, the most common default, nearby points are far more influential than distant ones. As p grows, IDW approaches nearest neighbor; as p approaches 0, it approaches a simple average.

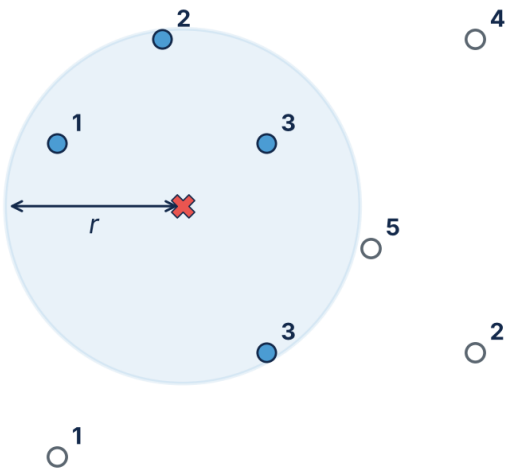


Figure 15.5: Average neighbor: the estimate is the mean of the observations within radius r.

A worked example. Suppose we want the temperature at a location 40 km from Austin (19 °C), 50 km from San Antonio (22 °C), and 150 km from Houston (21 °C), with $p = 1$ (Figure 15.6):

$$\hat{z} = \frac{19/40 + 22/50 + 21/150}{1/40 + 1/50 + 1/150} = \frac{1.055}{0.05167} \approx 20.4\,°\text{C}$$

Houston, three times farther away, has little influence. (With $p = 2$, the estimate is 20.2 °C: San Antonio and Austin matter even more. Avoid rounding the sums too early; rounding them to two places gives 21.)

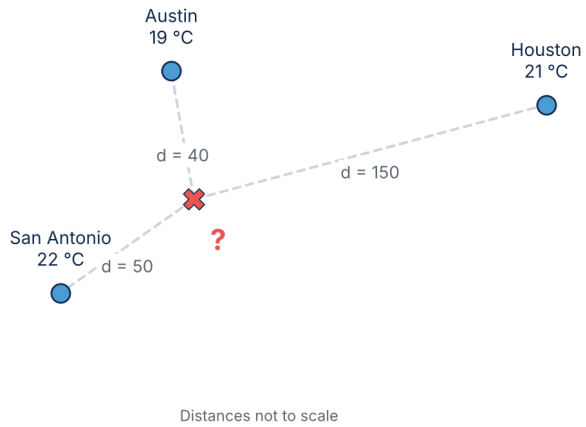


Figure 15.6: The IDW worked example. Three stations surround an unknown location.

IDW is exact: at an observation the distance is 0, so that observation gets all the weight. It produces a continuous surface. But you must choose both the power and the number of neighbors, and the result is sensitive to both (Figure 15.7). Like average neighbor, it can't predict values outside the observed range, so peaks and pits between observations are flattened. And it ignores direction. Its signature is the **bull's-eye**: a ring around an observation whose value differs from its neighbors'.

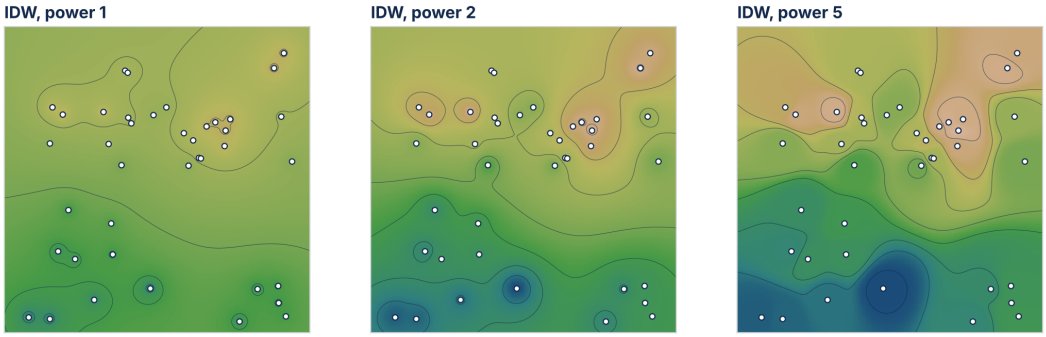


Figure 15.7: IDW on the same 40 samples with powers 1, 2, and 5. Higher powers give each observation a larger area of influence, and sharper bull's-eyes.

Think about it

A city has 30 air-quality monitors, 25 of them in one neighborhood near the port. If you interpolate pollution with IDW and 12 neighbors, what happens to estimates on the far side of the city? How could the search neighborhood (Section 15.3) help?

15.2.4 Polynomial (trend surface) interpolation

Polynomial interpolation fits one mathematical surface, a **trend surface**, to all the observations, usually by least squares. The **order** of the polynomial, the highest power of x and y in its equation, sets how many bends the surface can have (Figure 15.8):

- **First order** is a tilted plane: $\hat{z} = \beta_0 + \beta_1 x + \beta_2 y$.
- **Second order** can bend once, like a bowl or a ridge: $\hat{z} = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 x^2 + \beta_4 xy + \beta_5 y^2$.
- **Third order** can bend twice, adding the x^3, x^2y, xy^2, and y^3 terms.

Each β is a coefficient fit from the observations. Higher orders fit more closely, but quickly become unstable, especially near the edges of the data. Orders above 3 or 4 are rarely useful.

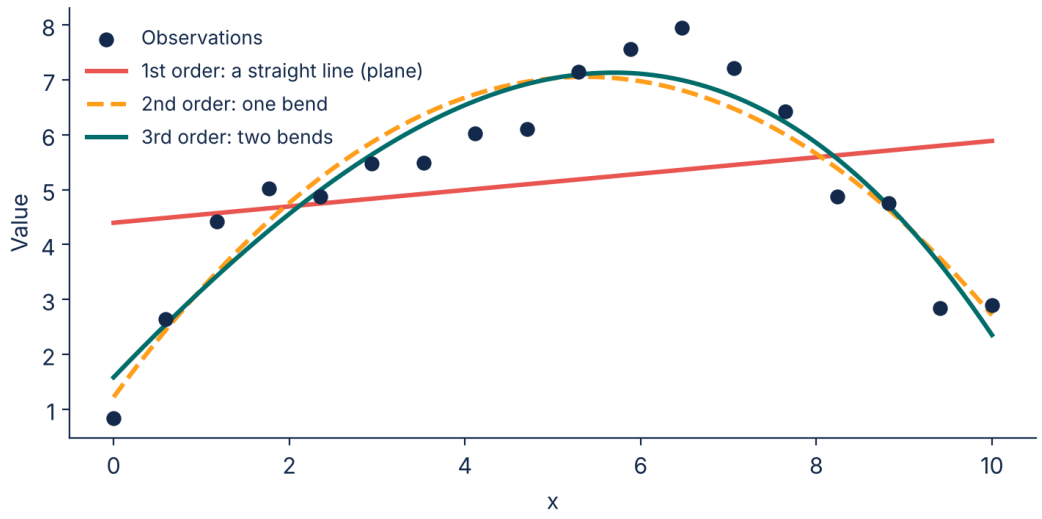


Figure 15.8: Polynomials of order 1, 2, and 3 fit to the same observations along a profile.

A **global polynomial** uses one formula for the whole study area. It is simple, with only the order to choose, and can extrapolate, but it is not exact and captures only the broad, regional pattern. That makes it useful for mapping a trend, or for removing one before interpolating what's left (the residuals) with another method. A **local polynomial** fits a separate surface to each overlapping neighborhood and blends them. It follows complex surfaces and comes close to the observations, but requires more choices: the order, and the size and shape of the neighborhood.

15.2.5 Radial basis functions (splines)

Radial basis function (RBF) interpolation, often called a **spline**, fits a smooth surface that passes exactly through every observation while bending as little as possible. Think of a thin, flexible sheet pinned at every data point. RBFs are exact, smooth, and can extrapolate: between observations, the surface can rise above or dip below the observed values (Figure 15.9, bottom right). Their weakness is surfaces with abrupt changes, such as cliffs or faults, where a smooth sheet overshoots.

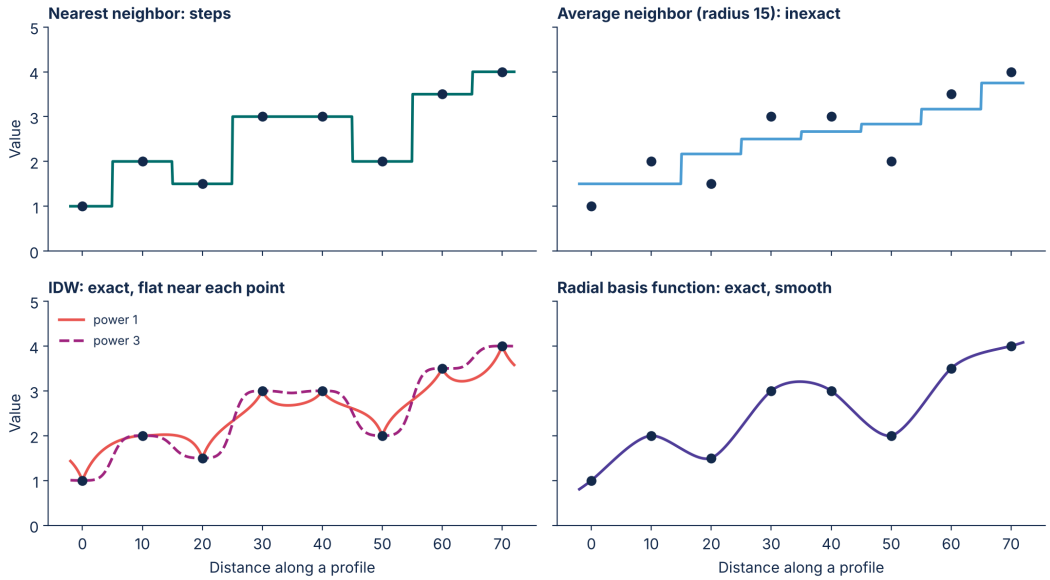


Figure 15.9: Four methods along the same profile of eight observations.

15.2.6 Comparing the methods

Figure 15.10 interpolates the same 40 samples from a made-up surface with five methods. All of them get the broad pattern. They differ in detail: Thiessen edges, IDW bull's-eyes, the global polynomial's smooth bowl, and the RBF's curves. Because every surface looks plausible, we can't pick the best by eye. That is what step 5, cross validation, is for (Section 15.4).

Table 15.1: Choosing an interpolation method.

Method	Exact?	Extrapolates?	Smooth?	Main choices
Nearest neighbor	Yes	No	No (steps)	None
Average neighbor	No	No	Yes	Radius r
IDW	Yes	No	Yes	Neighbors k, power p
Global polynomial	No	Yes	Yes	Order

Method	Exact?	Extrapolates?	Smooth?	Main choices
Local polynomial	Nearly	Yes	Yes	Order, neighborhood
Radial basis function	Yes	Yes	Yes	Function, smoothing

15.3 Steps 2 and 3: Neighborhoods and sampling

15.3.1 The search neighborhood

The **search neighborhood** decides which observations feed each estimate. Its settings can change the result as much as the choice of method (Figure 15.11):

- **Radius.** A *fixed* radius uses every observation within a set distance, which may be very few in sparse areas. A *variable* radius grows until it captures k points.
- **Sectors.** Dividing the circle into quadrants or octants, and taking a few points from each, stops one dense cluster on one side from dominating the estimate.
- **Shape.** A circle treats all directions equally. An ellipse suits a **directional trend**, where values change faster in one direction than another, such as across a valley rather than along it.

15.3.2 Sampling design

An interpolation can only be as good as its samples. **Sampling** is choosing where observations are collected, and there are four common designs (Figure 15.12):

- **Systematic**: points on a regular grid. Coverage is even and easy to plan, but a pattern that repeats at the grid spacing can be missed entirely.
- **Random**: every location has an equal chance. It avoids that bias, but by chance leaves some areas empty and others crowded.
- **Clustered**: points in groups, usually to save travel time in the field. It describes local variation well, but leaves gaps between clusters.
- **Adaptive** (multi-pass): a sparse first pass, then more samples where the first pass shows rapid change. It puts the effort where the surface is most complex, and is often the most efficient.

A useful rule: put more samples where the surface changes quickly, and fewer where it is uniform.

15.4 Step 5: Estimating the error

15.4.1 Cross validation

An interpolated surface always looks plausible, so we need a number that says how wrong it is likely to be. We usually can't go back into the field, so we test the surface with the observations we already have. **Cross validation** (leave-one-out) works like this:

1. Remove one observation from the data.

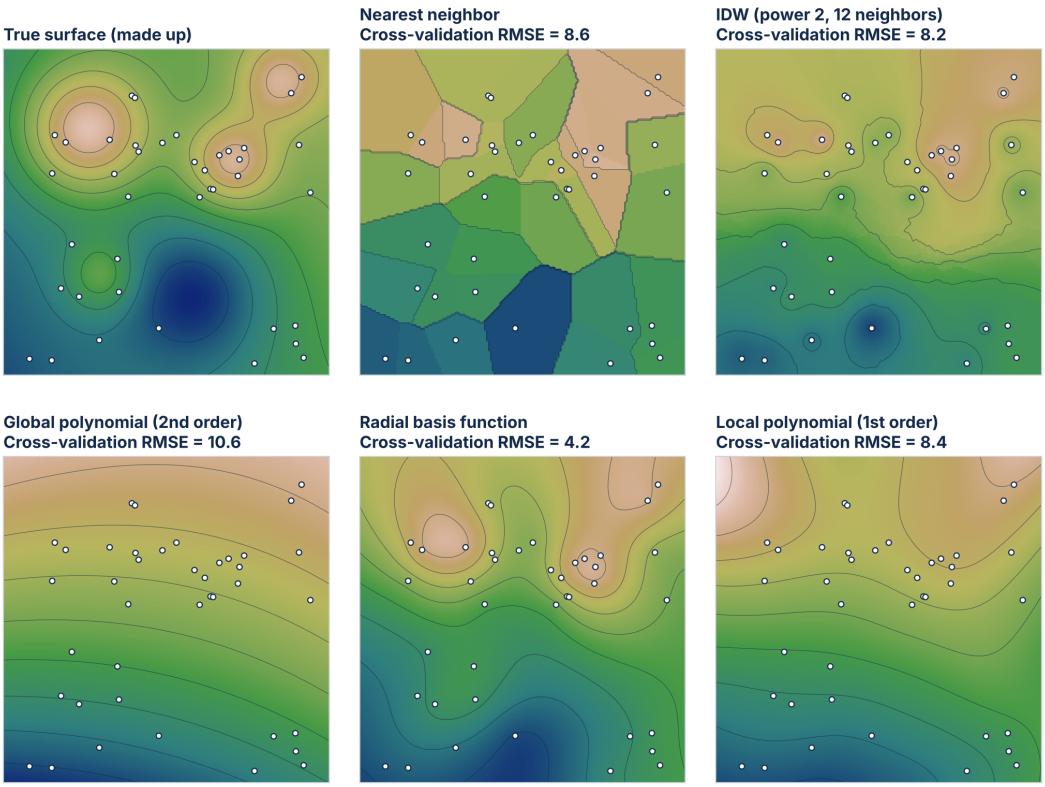


Figure 15.10: A made-up surface, and five interpolations of the same 40 samples, with their cross-validation RMSE. Lower is better.

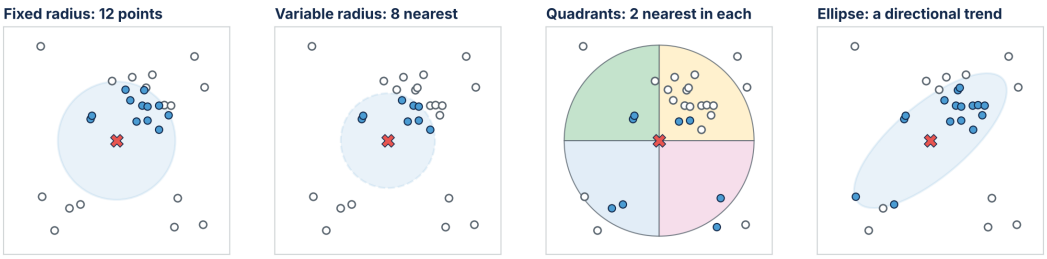


Figure 15.11: Four ways to choose the observations (blue) used to estimate the red X.

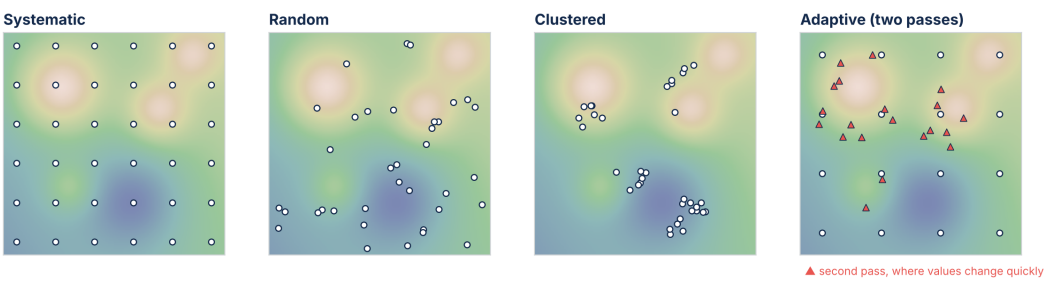


Figure 15.12: Four sampling designs over the same made-up surface.

2. Interpolate a value at its location from the remaining observations.
3. Compare the estimate with the observed value, and save the error.
4. Put the observation back, and repeat for every observation.

In Figure 15.13, the observation of 3.0 is removed. IDW with power 1, using its four neighbors (4 and 3 at distance 10; 4 and 2 at 14.14), estimates 3.29, an error of 0.29.

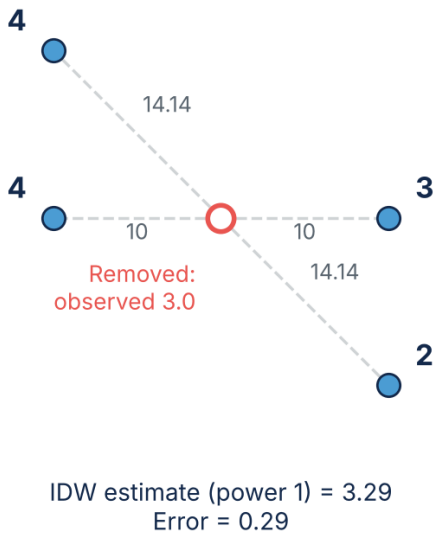


Figure 15.13: One step of cross validation.

When there are enough observations, an alternative is **validation**: hold back a separate test set, build the surface from the rest, and measure the error only at the held-back points.

15.4.2 Root mean square error

Root mean square error (RMSE) summarizes the cross-validation errors in one number:

$$\text{RMSE} = \sqrt{\frac{\sum(\hat{z}_i - z_i)^2}{n}}$$

where $\hat{z}_i$ is the estimate at observation i, z_i is its observed value, and n is the number of observations. Squaring keeps positive and negative errors from canceling; the square root returns to the original units. So an RMSE of 2 mm of rainfall means something directly. **Lower is better.** RMSE is the standard way to compare methods, or settings such as the IDW power, *on the same data*.

In Figure 15.10, the radial basis function has by far the lowest RMSE, 4.2, on a surface ranging from about 33 to 118. The global polynomial has the highest, 10.6, because a single bowl can't follow two peaks and a basin. On a different surface, the ranking could change. That is the point: no method wins everywhere.

The **mean error** is worth checking too. It should be near 0. If it isn't, the surface is biased: consistently too high or too low.

15.4.3 Four questions

Before trusting any interpolated surface, ask:

1. **What type of data is it?** Categorical data call for nearest neighbor; continuous data allow the other methods.
2. **What is the error at the observations?** Exact interpolators have none there, but that doesn't make them right elsewhere.
3. **How does the surface behave between observations?** Look for overshoot, bull's-eyes, and flat spots.
4. **What does the error look like?** Errors should look random. If they are consistently large in one region, the method or its settings are wrong there.

15.5 Interpolation in ArcGIS Pro and QGIS

- **ArcGIS Pro.** The Spatial Analyst toolbox has *IDW*, *Spline* (an RBF), *Natural Neighbor*, and *Trend* (a global polynomial). The **Geostatistical Analyst** extension adds the **Geostatistical Wizard**, which runs IDW, global and local polynomials, RBFs, and kriging with full control of the search neighborhood, and reports cross-validation statistics, including RMSE and the mean error. *Create Thiessen Polygons* (Analysis toolbox) draws Thiessen polygons.
- **QGIS.** The Processing Toolbox's *Interpolation* group has *IDW interpolation* and *TIN interpolation*; GDAL's *Grid* tools add nearest neighbor, moving average (average neighbor), and inverse distance to a power; and *Voronoi polygons* draws Thiessen polygons. SAGA and GRASS add splines and kriging.

15.6 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

15.7 Summary

- **Spatial interpolation** estimates values at unsampled locations from **observations**, relying on Tobler's first law. **Extrapolation** goes beyond the observations.
- Every interpolation has five steps: choose a function, define the neighborhood, find the points, evaluate the function, and estimate the error.
- **Exact** interpolators pass through every observation; **inexact** ones smooth. **Deterministic** methods use a formula; **geostatistical** methods (kriging) model autocorrelation and uncertainty.
- **Nearest neighbor** (**Thiessen polygons**) suits categories but gives a stepped surface. **Average neighbor** smooths but is inexact. **IDW** weights by distance raised to a **power**; it is exact but forms **bull's-eyes** and can't exceed the observed range.
- **Polynomial** interpolation fits a **trend surface** whose **order** sets its bends; global polynomials show broad trends, local ones follow detail. **Radial basis functions** (splines) are exact, smooth, and can extrapolate.
- The **search neighborhood** (radius, sectors, shape) and the **sampling design** (systematic, random, clustered, adaptive) shape every result.

- **Cross validation** and **RMSE** compare methods objectively. Lower RMSE is better, and the mean error should be near 0.

15.8 Exercises

1. **IDW by hand.** Redo the Austin–San Antonio–Houston example with power 2 and power 3. How does the estimate change, and why?
2. **Thiessen polygons.** In QGIS or ArcGIS Pro, draw Thiessen polygons around the weather stations in North Carolina (from the State Climate Office of North Carolina or NOAA's station list). Which part of the state has the largest polygons?
3. **Compare methods.** Download a month of rainfall totals for North Carolina stations from NOAA's Climate Data Online. Interpolate them with IDW (powers 1, 2, and 3) and a spline. Which has the lowest cross-validation RMSE?
4. **Sampling.** You must sample soil lead in a 1 km² neighborhood with a possible hot spot near an old gas station. Which sampling design would you use, and how many samples would go near the station?
5. **Bull's-eyes.** Find a published IDW map (for example, of precipitation or groundwater). Can you spot bull's-eyes? What do they tell you about the stations?

15.9 Further reading and online resources

15.9.1 Books

- *GIS Fundamentals*, by Paul Bolstad (Bolstad 2022). The chapter on spatial estimation covers sampling, IDW, splines, and kriging.
- *Geographic Information Analysis*, by David O'Sullivan and David Unwin (O'Sullivan and Unwin 2010), has a thorough chapter on interpolation.
- *Geospatial Analysis: A Comprehensive Guide*, by de Smith, Goodchild, and Longley, free online, with a long section on deterministic interpolation: https://www.spatialanalysisonline.com.
- *Intro to GIS and Spatial Analysis*, by Manuel Gimond, free online, chapter on spatial interpolation: https://mgimond.github.io/Spatial/spatial-interpolation.html.

15.9.2 ArcGIS Pro

- Esri, "Comparing interpolation methods": https://pro.arcgis.com/en/pro-app/latest/tool-reference/3d-analyst/comparing-interpolation-methods.htm.
- Esri, "How IDW works": https://pro.arcgis.com/en/pro-app/latest/tool-reference/3d-analyst/how-idw-works.htm.
- Esri, "How Spline works": https://pro.arcgis.com/en/pro-app/latest/tool-reference/3d-analyst/how-spline-works.htm.
- Esri, "Performing cross validation and validation": https://pro.arcgis.com/en/pro-app/latest/help/analysis/geostatistical-analyst/performing-cross-validation-and-validation.htm.
- Esri, "Create Thiessen Polygons": https://pro.arcgis.com/en/pro-app/latest/tool-reference/analysis/create-thiessen-polygons.htm.

15.9.3 QGIS

- QGIS User Guide, “Interpolation” algorithms (IDW and TIN): https://docs.qgis.org/latest/en/docs/user_manual/processing_algs/qgis/interpolation.html.
- QGIS User Guide, GDAL “Raster analysis” algorithms (Grid tools): https://docs.qgis.org/latest/en/docs/user_manual/processing_algs/gdal/rasteranalysis.html.
- QGIS Training Manual, “Spatial Statistics” and interpolation lessons: https://docs.qgis.org/latest/en/docs/training_manual/processing/interpolation.html.

15.9.4 Data

- NOAA National Centers for Environmental Information, Climate Data Online: https://www.ncei.noaa.gov/cdo-web/.
- State Climate Office of North Carolina: https://climate.ncsu.edu.
- PRISM Climate Group, Oregon State University, which publishes interpolated climate maps for the United States: https://prism.oregonstate.edu.
- The GIS&T Body of Knowledge, “Spatial Interpolation” topics: https://www.ucgis.org/site/gis-t-body-of-knowledge.

16 Kriging

> **Learning objectives**
>
> By the end of this chapter you should be able to:
>
> 1. Compute semivariance for a pair of observations and for a distance bin.
> 2. Read a semivariogram, and explain its nugget, sill, and range.
> 3. Explain how kriging sets its weights, and how that differs from IDW.
> 4. Recognize anisotropy and trends, and model them.
> 5. Tell ordinary, simple, universal, and cokriging apart.
> 6. Judge a kriged surface with cross-validation statistics and a prediction standard error map.

The interpolation methods in Chapter 15 are deterministic: you choose a formula and its settings, and the surface follows. IDW, for example, weights observations by distance raised to a power you pick, whatever the data look like. **Kriging** takes a different approach. It first *measures* how values in your data are related at different distances, then uses that measurement to set the weights. Because it is built on a statistical model, it can also estimate how uncertain each prediction is.

16.1 Semivariance

16.1.1 Measuring dissimilarity with distance

Tobler's first law says near things are more related than distant things (Tobler 1970). Chapter 12 called that tendency **spatial autocorrelation**. **Semivariance** turns it into a number. For two observations i and j, it is half the squared difference between their values:

$$\gamma = \tfrac{1}{2}(z_i - z_j)^2$$

Similar values give low semivariance; different values give high semivariance. If the first law holds, pairs close together should have low semivariance, and pairs far apart high semivariance.

For many pairs, we group them into **bins** (also called lags) by the distance between them, and average within each bin. The semivariance of a bin h with n pairs is:

$$\gamma(h) = \frac{1}{2n}\sum(z_i - z_j)^2$$

In Figure 16.1, the pair of values 4 and 3 has a semivariance of ½ × 1² = 0.5. Four pairs are exactly 10 apart, with squared differences of 1, 1, 0, and 0, so $\gamma(10)$ = ½ × 2 ÷ 4 = 0.25. (The two diagonal pairs, 14.1 apart, belong to another bin.)

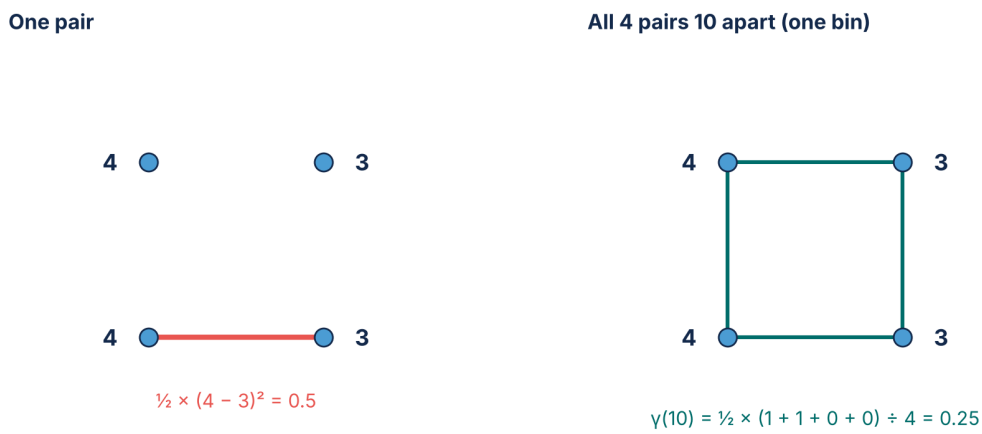


Figure 16.1: Semivariance for one pair (left) and for the bin of pairs 10 apart (right).

16.1.2 Direction matters

Consider a made-up surface where values rise steadily from west to east, from 0 to 100, and don't change at all from north to south (Figure 16.2). Plotting the semivariance of every pair against its distance gives a **semivariogram cloud**. West–east pairs have semivariance that grows quickly with distance, because the farther apart they are, the more their values differ. North–south pairs have low semivariance at *any* distance, because values don't change in that direction.

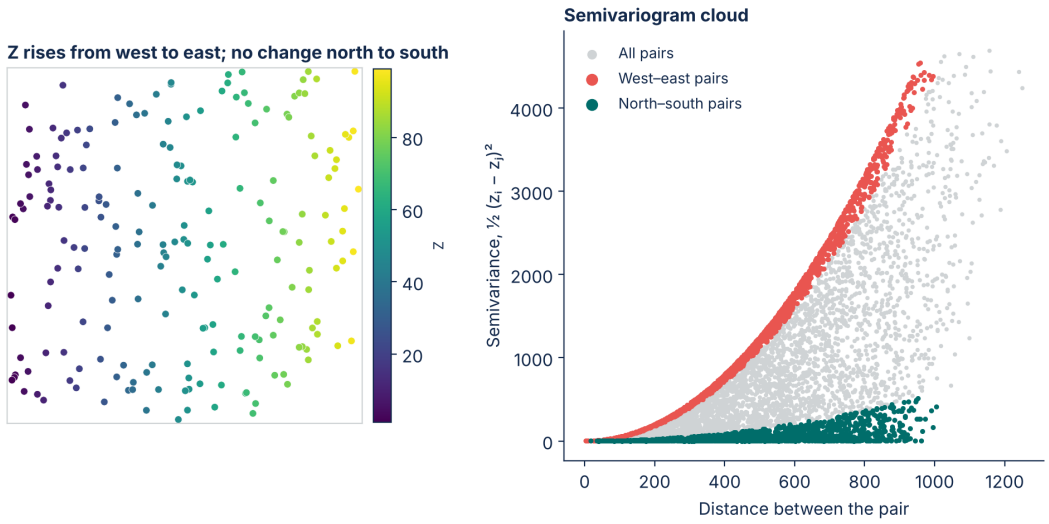


Figure 16.2: A made-up surface rising from west to east (left), and its semivariogram cloud (right). West–east pairs grow more different with distance; north–south pairs do not.

So distance alone doesn't determine semivariance; direction can matter too. We return to this as **anisotropy** (Section 16.3.3).

16.2 The semivariogram

A **semivariogram** (often just **variogram**) summarizes the cloud: it plots the average semivariance of each distance bin against the bin's average distance. Kriging then fits a smooth

mathematical model through those points. Three parameters describe the model (Figure 16.3):

- The **sill** is the semivariance where the curve levels off. Pairs farther apart than this are as different as two random observations.
- The **range** is the distance at which the sill is reached. Within the range, observations are autocorrelated; beyond it, they are effectively independent, and distance adds no information.
- The **nugget** is the semivariance at a distance of (nearly) zero. In theory, two measurements at the same place should be identical, so the nugget should be 0. In practice it reflects measurement error, and variation at scales smaller than the spacing of the samples. The sill minus the nugget is the **partial sill**.

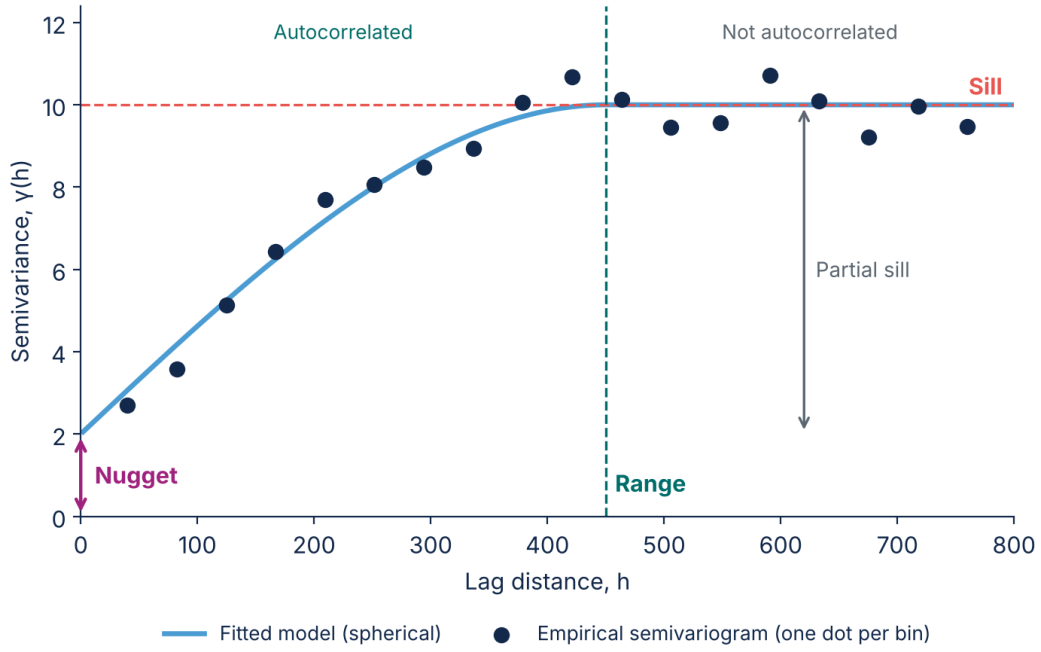


Figure 16.3: The parts of a semivariogram. Dots are bin averages from the data; the line is a fitted spherical model.

Did you know..?

Kriging is named for Danie Krige, a South African mining engineer. In the early 1950s, he was estimating how much gold lay between boreholes in the Witwatersrand gold fields, and found that ore grades in nearby samples were related in a way he could use (Krige 1951). The French mathematician Georges Matheron formalized his ideas as *geostatistics* in the 1960s, and coined the term *krigeage* in Krige's honor (Matheron 1963). The "nugget" in the semivariogram is a mining term too: a gold nugget in one sample makes it very different from a sample taken right beside it.

16.2.1 Semivariogram models

The fitted model must be one of a few allowed shapes, which guarantee that kriging gives sensible answers. Four are common (Figure 16.4):

- **Spherical**: rises almost linearly and reaches the sill exactly at the range. The most widely used.
- **Exponential**: rises quickly and approaches the sill gradually.

- **Gaussian**: flat near the origin, then rises; suits very smooth surfaces.
- **Linear**: never levels off. A semivariogram with no sill usually signals a **trend** in the data (Section 16.3.4).

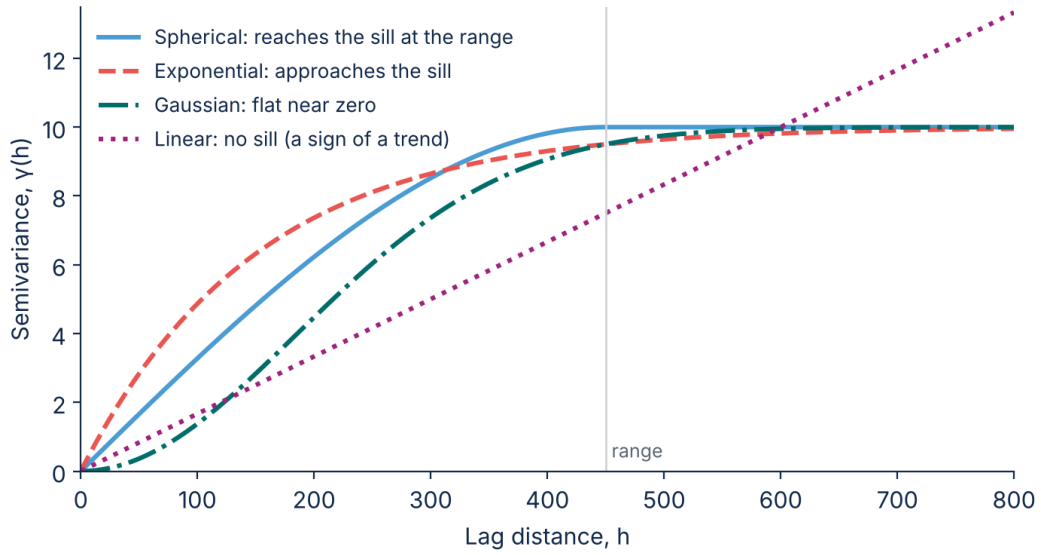


Figure 16.4: Four semivariogram models with the same range and sill.

Choose the model whose shape best matches the empirical semivariogram, especially at short distances, which matter most for prediction. Then check the choice with cross validation.

16.3 Kriging

16.3.1 A surface in three parts

Kriging treats each value as the sum of three parts (Figure 16.5):

1. A **trend**: the broad, large-scale change across the study area.
2. **Spatial autocorrelation**: local, distance-related variation around the trend. This is what the semivariogram describes.
3. **Random error**: what's left, which can't be predicted from location.

16.3.2 Weights from the semivariogram

Like IDW, kriging predicts a value at location s_0 as a weighted sum of nearby observations:

$$\hat{Z}(s_0) = \sum \lambda_i \, Z(s_i)$$

The difference is where the weights λ_i come from. IDW sets them from distance and a power you choose. Kriging *solves* for them, using the fitted semivariogram, so that the predictions are **unbiased** and have the **smallest possible error variance**. In **ordinary kriging**, the weights sum to 1.

Because kriging accounts for how the observations relate to *each other*, not only to the prediction location, a cluster of nearby observations **shares** weight instead of each counting

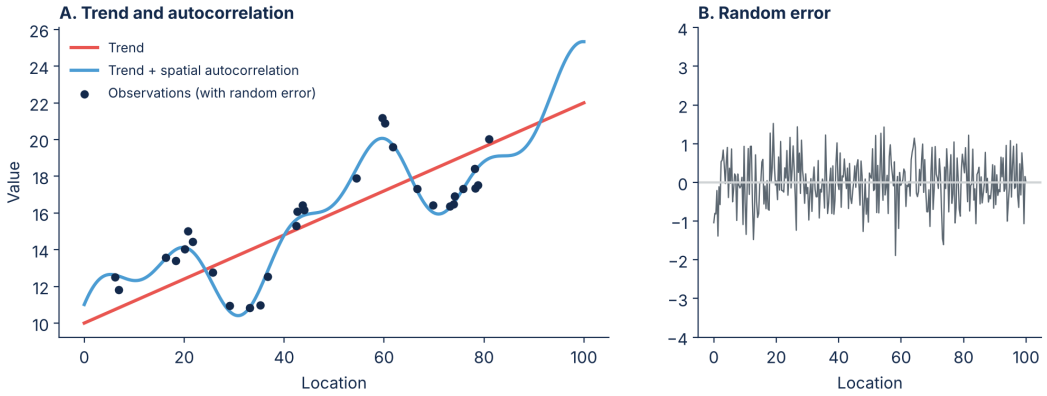


Figure 16.5: Kriging's view of a surface: a trend, autocorrelated variation around it (A), and random error (B).

fully (Figure 16.6). Four clustered observations tell us little more than one of them would. IDW gives the cluster more than half the weight; kriging gives it a third, and gives more to the isolated observations that bring new information. Kriging weights can even be slightly negative, for observations "screened" behind others.

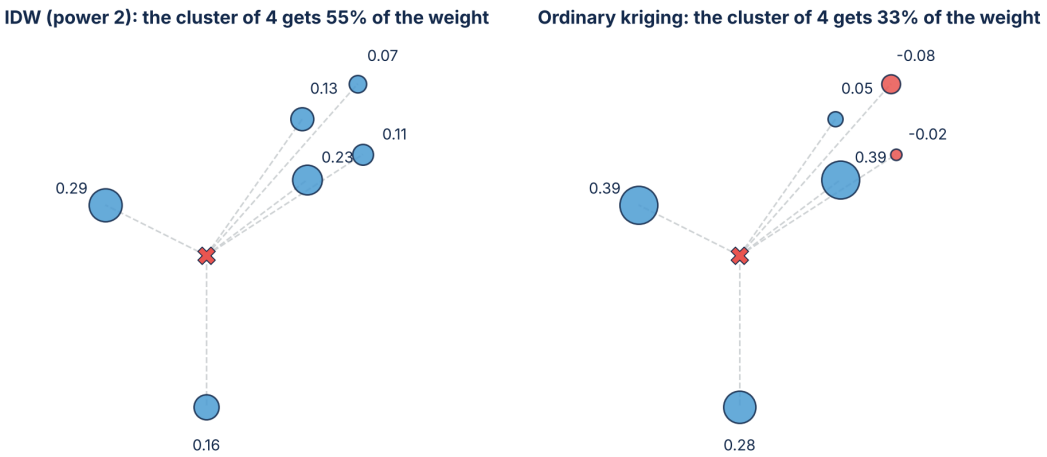


Figure 16.6: Weights for the same prediction (red X) from IDW and from ordinary kriging. Circle size shows weight; red circles are negative weights.

Kriging also gives a **prediction standard error** at every location: an estimate of how far off the prediction might be. It is smallest near observations, and largest where they are sparse (Figure 16.9, right). No deterministic method offers this.

16.3.3 Anisotropy

A surface is **isotropic** when spatial dependence is the same in every direction, and **anisotropic** when it changes with direction. Consider a made-up surface that varies in a sine wave from west to east, with a wavelength of 400, and doesn't change from north to south (Figure 16.7). A semivariogram that pools every direction blurs the pattern. One that uses only west–east pairs shows it clearly: pairs half a wavelength apart (200, 600) are most different, and pairs a full wavelength apart (400, 800) are alike again.

An **anisotropic** model fits a longer range in one direction (the **major range**) and a shorter one in the perpendicular direction (the **minor range**). Its search neighborhood becomes an ellipse stretched along the direction where values stay similar. On the sine surface, the

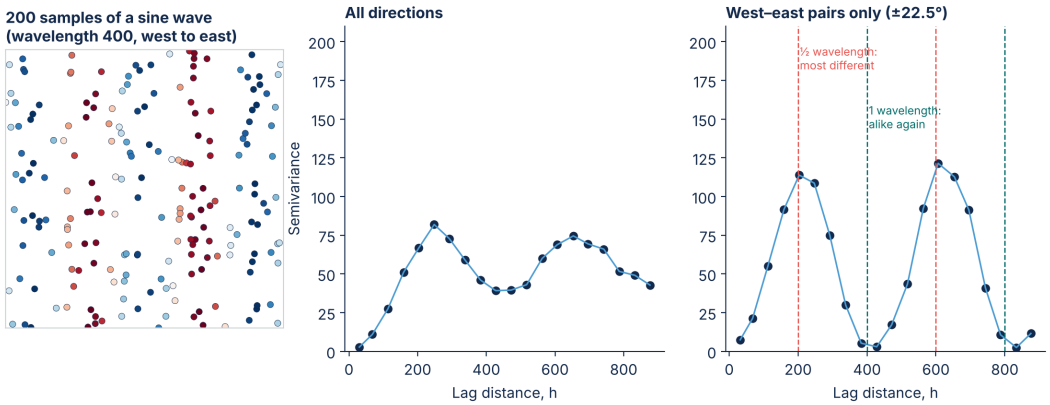


Figure 16.7: Made-up sine-wave samples (left), and their semivariograms for all directions (middle) and for west–east pairs only (right).

cross-validation RMSE falls from 2.62 for IDW to 1.61 for isotropic kriging, and to 0.38 for anisotropic kriging, which keeps the bands continuous (Figure 16.8).

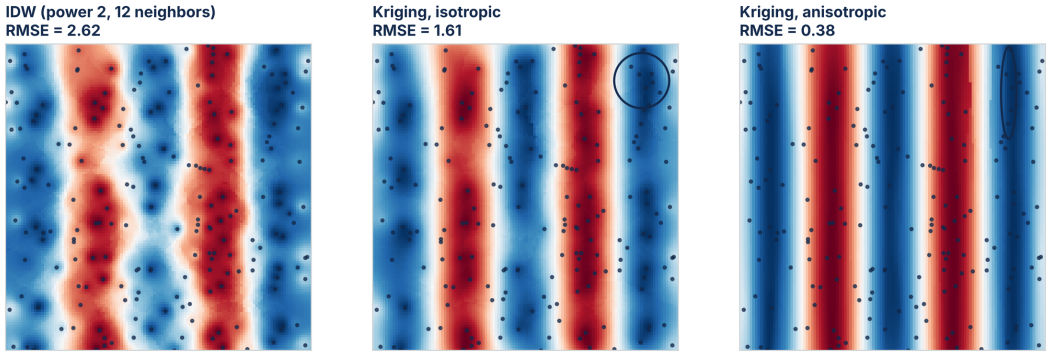


Figure 16.8: The sine samples interpolated with IDW, isotropic kriging (circular search), and anisotropic kriging (elliptical search), with cross-validation RMSE.

Think about it

Name a variable that you would expect to be anisotropic. Which direction would have the longer range? (Think of a river valley, a mountain front, the prevailing wind, or a coastline.)

16.3.4 Trends

Kriging assumes that the mean is stable across the study area. When the data have a strong trend, such as temperature falling steadily from south to north, the semivariogram keeps rising without a sill, like the linear model in Figure 16.4. The fix is to **remove the trend** first: fit a trend surface (Chapter 15), krige the **residuals** (observed minus trend), and add the trend back to the predictions.

Figure 16.9 does this for the 40 made-up samples from Chapter 15. The raw semivariogram has no sill; after a first-order trend is removed, the residuals' semivariogram levels off and fits a spherical model. The kriged surface has a cross-validation RMSE of 5.2, better than IDW (8.2) but not as good as the radial basis function (4.2) on this surface. Kriging is not guaranteed to win. Its standard error map shows where the prediction is least certain: the corners and the empty lower right.

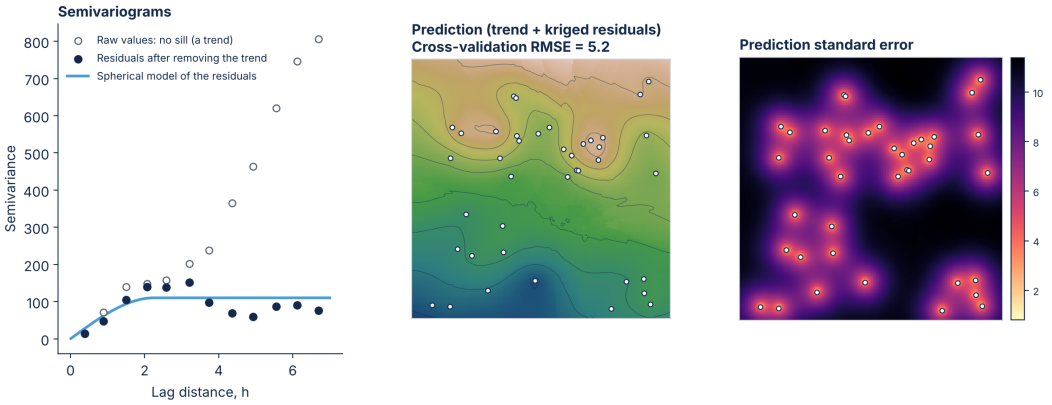


Figure 16.9: Kriging the 40 samples from Chapter 15: semivariograms of the raw values and of the residuals after removing a trend (left), the prediction (middle), and its standard error (right).

16.3.5 Types of kriging

The main types differ in how they treat the trend:

Table 16.1: Types of kriging.

Type	The trend (mean) is	Use when
Ordinary kriging	An unknown constant	The most common choice; no strong trend
Simple kriging	A known constant	The mean is known, for example from a long record
Universal kriging	A function of location, such as a polynomial	There is a clear trend, such as a west–east gradient
Cokriging	Any of the above, plus a correlated secondary variable	A second, densely sampled variable helps, such as elevation for temperature

Universal kriging is the version of Figure 16.9: the trend is modeled as a polynomial and the residuals are kriged. It suits variables like U.S. precipitation, which changes steadily from the dry West to the wet Southeast.

16.4 Judging the result

16.4.1 Cross-validation statistics

As in Chapter 15, **cross validation** removes each observation in turn, predicts it from the rest, and compares. Because kriging also predicts its own standard errors, it adds two statistics that test whether its uncertainty is realistic (Table 16.2). ArcGIS Pro's Geostatistical Wizard reports all four.

Table 16.2: Kriging cross-validation statistics.

Statistic	Target	What it tells you
Mean error	Close to 0	Predictions are unbiased
Root mean square error (RMSE)	As small as possible	How close predictions are to the observed values
Average standard error	Close to the RMSE	Whether the model's uncertainty is realistic
Root mean square standardized error	Close to 1	Above 1: uncertainty is underestimated; below 1: overestimated

16.4.2 The kriging workflow

Putting it together, a typical kriging analysis, as in ArcGIS Pro's Geostatistical Wizard:

1. **Explore the data.** Map it; check the histogram; look for outliers and trends.
2. **Remove any trend**, and choose the type of kriging.
3. **Compute the semivariogram**, in several directions to check for anisotropy.
4. **Fit a model**: choose its shape, and set the nugget, sill, and range (and major and minor ranges, if anisotropic).
5. **Set the search neighborhood**: the number of neighbors, sectors, and shape.
6. **Cross-validate**, and compare the statistics with other models and methods.
7. **Map the prediction and its standard error**.

Use cross-validation statistics, not appearance, to choose among models and methods. Every surface looks plausible.

16.5 Kriging in ArcGIS Pro and QGIS

- **ArcGIS Pro.** The **Geostatistical Wizard** (Geostatistical Analyst extension) runs ordinary, simple, universal, and cokriging, with full control of the semivariogram, anisotropy, trend removal, and search neighborhood, and reports cross-validation statistics. **Empirical Bayesian Kriging** automates semivariogram fitting by simulating many semivariograms. The Spatial Analyst *Kriging* tool offers a simpler interface.
- **QGIS.** The Processing Toolbox includes SAGA's kriging tools (ordinary and universal kriging) and GRASS's *v.surf.rst* for splines. The **Smart-Map** plugin provides kriging with semivariogram fitting and cross validation.

16.6 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

16.7 Summary

- **Semivariance** is half the squared difference between two values. Averaged by distance **bin**, it shows how dissimilarity grows with distance.
- A **semivariogram** plots semivariance against distance. Its **nugget**, **sill**, and **range** describe spatial dependence. Models include **spherical**, **exponential**, **Gaussian**, and **linear**.
- **Kriging** predicts a weighted sum of observations, with weights solved from the fitted semivariogram so predictions are **unbiased** with **minimum error variance**. Clustered observations share weight, and every prediction has a **standard error**.
- **Anisotropy** (dependence that changes with direction) is modeled with major and minor ranges and an elliptical search. **Trends** are removed before modeling, and added back.
- **Ordinary**, **simple**, **universal**, and **cokriging** differ in how they treat the trend, and whether they use a secondary variable.
- Judge a kriged surface with cross validation: mean error near 0, small RMSE, average standard error near the RMSE, and RMS standardized error near 1.

16.8 Exercises

1. **Semivariance by hand.** Five wells along a road, 100 m apart, have water-table depths of 12, 14, 13, 17, and 18 m. Compute $\gamma(100)$ and $\gamma(200)$.
2. **Read a semivariogram.** For the semivariogram in Figure 16.3, estimate the nugget, sill, partial sill, and range. Beyond what distance would you stop using observations to predict a location?
3. **Kriging vs. IDW.** Using a month of North Carolina rainfall totals from NOAA's Climate Data Online, interpolate with IDW and ordinary kriging in ArcGIS Pro's Geostatistical Wizard. Compare the cross-validation statistics. Which method wins?
4. **Trend.** Interpolate average annual temperature for the lower 48 states with ordinary kriging and with universal kriging (first-order trend). Compare the semivariograms. Which levels off?
5. **Standard error.** Map the prediction standard error from Exercise 3. Where would you add new rain gauges, and why?

16.9 Further reading and online resources

16.9.1 Books and papers

- *GIS Fundamentals*, by Paul Bolstad (Bolstad 2022). The chapter on spatial estimation introduces the semivariogram and kriging.
- *Geographic Information Analysis*, by David O'Sullivan and David Unwin (O'Sullivan and Unwin 2010), explains semivariograms and kriging clearly, with worked examples.
- *An Introduction to Applied Geostatistics*, by Edward Isaaks and R. Mohan Srivastava (Oxford University Press, 1989), a classic, readable introduction.
- Krige's original paper (Krige 1951) and Matheron's "Principles of Geostatistics" (Matheron 1963).
- *Intro to GIS and Spatial Analysis*, by Manuel Gimond, free online, includes kriging in its chapter on spatial interpolation: https://mgimond.github.io/Spatial/spatial-interpolation.html.

16.9.2 ArcGIS Pro

- Esri, "How Kriging works": https://pro.arcgis.com/en/pro-app/latest/tool-reference/3d-analyst/how-kriging-works.htm.
- Esri, "Understanding a semivariogram: The range, sill, and nugget": https://pro.arcgis.com/en/pro-app/latest/help/analysis/geostatistical-analyst/understanding-a-semivariogram-the-range-sill-and-nugget.htm.
- Esri, "What is Empirical Bayesian Kriging?": https://pro.arcgis.com/en/pro-app/latest/help/analysis/geostatistical-analyst/what-is-empirical-bayesian-kriging-.htm.
- Esri, "Performing cross validation and validation": https://pro.arcgis.com/en/pro-app/latest/help/analysis/geostatistical-analyst/performing-cross-validation-and-validation.htm.
- Esri, "Kriging in Geostatistical Analyst": https://pro.arcgis.com/en/pro-app/latest/help/analysis/geostatistical-analyst/kriging-in-geostatistical-analyst.htm.

16.9.3 QGIS and other tools

- Smart-Map QGIS plugin: https://plugins.qgis.org/plugins/Smart_Map/.
- The GIS&T Body of Knowledge, "Kriging Interpolation" and "Semivariograms": https://www.ucgis.org/site/gis-t-body-of-knowledge.



Part IV

GIS Software

17 Getting to Know QGIS

Learning objectives

By the end of this chapter you should be able to:

1. Describe what QGIS is, what "free and open source" means for it, and how it grew from a one-person project in 2002 into one of the most widely used GIS programs in the world.
2. Choose and install a QGIS release, and find your way around the QGIS window.
3. Add data to a project from files, a CSV of coordinates, and web basemaps.
4. Enter data: create a new layer, digitize features, and edit and calculate attributes.
5. Make a basic map: symbolize and label layers, then build a print layout with a title, legend, scale bar, north arrow, and source.
6. Find and run analysis tools in the Processing Toolbox, and extend QGIS with plugins.

The chapters so far have explained the ideas of GIS: data models, coordinate systems, queries, joins, and analysis. This chapter is about one of the tools you will use to put those ideas to work. QGIS is a free, open-source desktop GIS that runs on Windows, macOS, and Linux. It can do nearly everything this book describes, and because it costs nothing, you can install it on any computer and keep using it after the course ends. Chapter 18 introduces ArcGIS Pro, the other GIS program used in this course.

The screenshots in this chapter come from QGIS 3.34 on Linux, using North Carolina's 100 counties and their 2025 population estimates from Chapter 11. Newer versions, and QGIS on Windows and macOS, look much the same; menus and buttons may move a little between versions, but the workflow does not change.

17.1 What is QGIS?

QGIS is a desktop geographic information system: a program for viewing, creating, editing, analyzing, and mapping geographic data. Its name is pronounced "Q-G-I-S," letter by letter, and comes from its original name, *Quantum GIS*. Its core features are much like those of any GIS:

- **Viewing** vector and raster data in almost any format, in any coordinate system, reprojected on the fly into the project's coordinate system (Chapter 4).
- **Editing**: creating new layers, drawing features, and entering and calculating attributes.
- **Analysis**: hundreds of tools for selection, overlay, buffers, raster analysis, interpolation, and more, in the Processing Toolbox.
- **Cartography**: symbology, labels, and print layouts for making finished maps.

17.1.1 Free and open source

QGIS is **open source** software, released under the GNU General Public License (GPL). Its source code is public, anyone can read or change it, and anyone may copy and share it, as long as the copies stay under the same license. "Free" here means two things: *free as in free beer*, because it costs nothing, and *free as in freedom*, because no company controls who may use it or how. QGIS is developed by a worldwide community of volunteers and paid developers, and is funded by donations, sponsors (including governments and companies that rely on it), and paid contracts for new features.

QGIS doesn't do everything itself. Like most open-source GIS, it is built on shared libraries that many other programs use too:

- **GDAL**, the Geospatial Data Abstraction Library, reads and writes hundreds of raster and vector formats. ArcGIS Pro, Google Earth, and R's `sf` package use it as well.
- **PROJ** converts coordinates between coordinate systems.
- **GEOS** performs geometric operations, such as buffers, intersections, and unions.
- **Qt**, a toolkit for building desktop programs, draws the windows and buttons. QGIS itself is written mostly in C++, and can be extended with Python.

QGIS is a project of the Open Source Geospatial Foundation (**OSGeo**), a nonprofit that supports open-source geospatial software such as GDAL, PROJ, GEOS, PostGIS, and GRASS GIS. Organizations that use QGIS include national mapping agencies, cities, counties, and state governments, and newsrooms such as *The Economist*.

17.2 A short history

QGIS began in early 2002, when Gary Sherman started developing a program he called *Quantum GIS*. The first public release, version 0.0.1, followed in July 2002. Other developers joined, and over the next two decades the program grew into a full GIS used around the world (Figure 17.1).

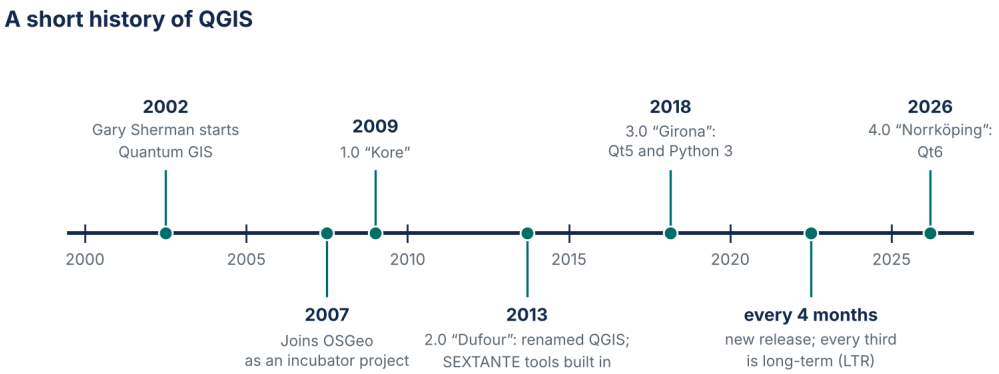


Figure 17.1: Milestones in the history of QGIS.

- **2007.** QGIS became an incubator project of the new Open Source Geospatial Foundation, which gave it a home, legal support, and visibility.
- **2009.** Version 1.0, code-named "Kore," was released in January 2009.



- **2013.** Version 2.0, “Dufour,” brought new symbology and labeling, and built in the SEXTANTE geoprocessing framework, now called **Processing**, with hundreds of analysis tools. The project officially dropped “Quantum” from its name, which had long been shortened to QGIS anyway.
- **2018.** Version 3.0, “Girona,” was a major rewrite, moving to new versions of Qt and to Python 3. GeoPackage became the default format for new layers.
- **2026.** Version 4.0, “Norrköping,” moved QGIS to Qt6, the newest version of its toolkit.

Every QGIS release has a version number and a code name, usually the name of a town or city: Kore, Dufour, Girona, Madeira, Prizren, Solothurn, Norrköping, and so on.

Did you know..?

QGIS can be extended with plugins, small add-ons that anyone can write and share. The official QGIS plugin repository lists more than 2,000 of them, contributed by the community: tools for geocoding, downloading OpenStreetMap data, making web maps, drawing elevation profiles, and almost anything else a GIS user has needed and decided to build. Each one is free to install from inside QGIS.

17.2.1 Long-term and latest releases

A new version of QGIS is released every four months. Every third release (one a year) is a **long-term release** (**LTR**), which gets bug fixes, but no new features, for about a year, until the next LTR replaces it. The others, called the *latest release* (LR), get the newest features first. Bug-fix updates to both come out monthly.

For coursework, install the **LTR** unless your instructor tells you otherwise. It is the most stable version, and the one that tutorials and lab instructions are usually written for. If you need a brand-new feature, install the latest release; you can install both side by side.

17.2.2 Installing QGIS

Download QGIS from https://qgis.org/download/. The site detects your operating system:

- **Windows**: download the standalone installer (an `.msi` file) for the LTR and run it. Installing as an administrator is easiest. Advanced users can use the OSGeo4W network installer instead, which installs QGIS alongside other open-source tools.
- **macOS**: download the `.dmg` file, open it, and drag QGIS to the Applications folder. The first time you open it, macOS may ask you to confirm that you want to open an app downloaded from the internet.
- **Linux**: install the QGIS package from your distribution, or, for the newest version, from the QGIS repositories (instructions are on the download page).

QGIS keeps your settings, plugins, and customizations in a **user profile**. If a setting or plugin breaks QGIS, you can start fresh by creating a new profile (Settings ▶ User Profiles ▶ New Profile) without reinstalling.

17.3 The QGIS window

When you open QGIS, you see a welcome page listing recent projects and templates. Choose **New Empty Project**, or open a recent one. Figure 17.2 shows the main window with a few layers added.

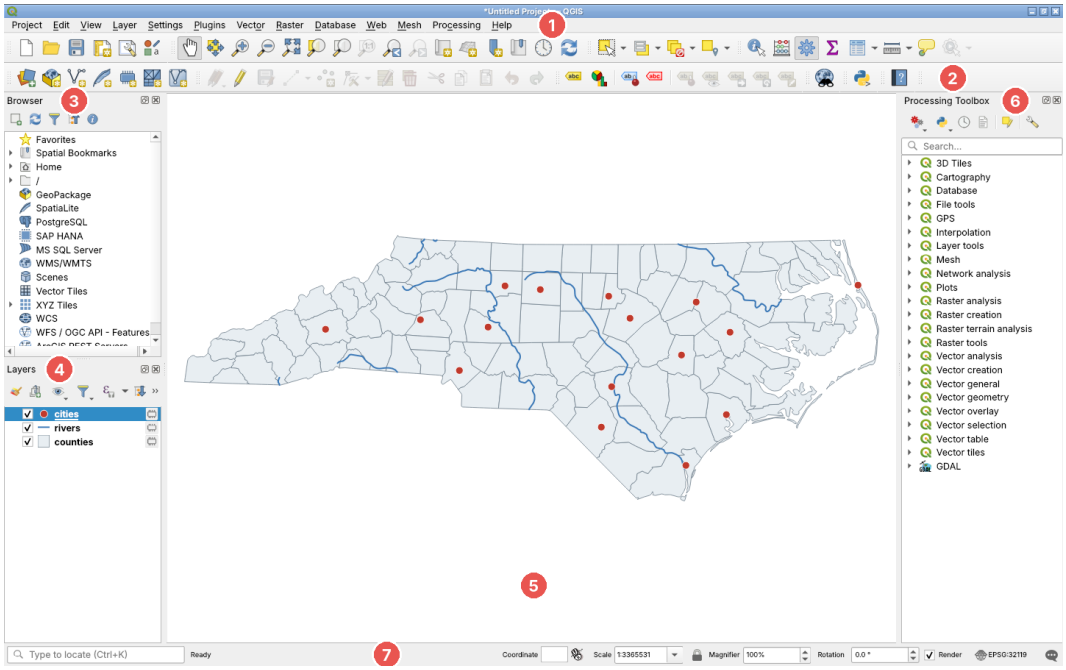


Figure 17.2: The QGIS window: (1) menu bar, (2) toolbars, (3) Browser panel, (4) Layers panel, (5) map canvas, (6) Processing Toolbox panel, and (7) status bar. Screenshot of QGIS 3.34.

1. The **menu bar** holds every command, grouped by purpose: Project, Edit, View, Layer, Settings, Plugins, Vector, Raster, Database, Web, Mesh, Processing, and Help.
2. **Toolbars** give one-click access to common commands: opening and saving projects, navigating the map, selecting features, editing, and labeling. Right-click any toolbar to turn toolbars on and off.
3. The **Browser panel** shows your computer's folders, databases, and web services. It is the fastest way to find and add data: drag a layer from the Browser onto the map.
4. The **Layers panel** lists the layers in the project. Layers draw from the bottom up, so a layer at the top of the list draws on top of the ones below. Check or uncheck a layer to show or hide it, drag layers to reorder them, and right-click a layer for its options.
5. The **map canvas** displays the layers.
6. The **Processing Toolbox** holds the analysis tools (Section 17.9).
7. The **status bar** shows the map coordinates under the cursor, the map scale, and, at the far right, the project's coordinate reference system (CRS). Click the CRS button to change it.

Panels can be moved, docked, stacked as tabs, or closed. To get a closed panel back, use **View ▶ Panels**. Other useful panels include **Layer Styling** (for changing symbology as you watch), **Identify Results**, and **Log Messages**.

17.3.1 Navigating the map

Table 17.1: Navigating the map canvas.

To do this	Use
Move around the map	**Pan** tool (the hand), or hold the spacebar or mouse wheel and drag
Zoom in and out	Mouse wheel, or the **Zoom In** and **Zoom Out** tools
Show a whole layer	Right-click the layer ▶ **Zoom to Layer(s)**
Show all layers	**Zoom Full** (Ctrl+Shift+F)
Go back to the previous view	**Zoom Last**
See a feature's attributes	**Identify Features** tool, then click the feature

17.4 Projects and data

A QGIS **project** is saved as a `.qgz` file. The project stores the list of layers, their symbology and labels, the coordinate system, and print layouts, but **not the data itself**. Each layer in a project is a *link* to a file, database, or web service. If you move or rename a data file, the project can't find it, and QGIS asks you to locate it the next time you open the project.

Good habits prevent most problems:

- **Keep each project and its data in one folder**, with the data in a subfolder. QGIS saves links as relative paths by default, so you can then move or share the whole folder.
- **Use short file and folder names without spaces or special characters**, such as `nc_density` rather than `NC Density (final)!`.
- **Save often** (Ctrl+S). Unlike a word processor, QGIS doesn't save automatically.
- **Copy data from a USB drive or network share to your computer** before working with it.

17.4.1 File formats

QGIS reads almost any format, but two are most common for vector data:

- The **GeoPackage** (`.gpkg`) is the default format in QGIS. It is a single file, based on the SQLite database, that can hold many vector layers, rasters, and tables. It has no limit on field name length and no 2 GB size limit.
- The **shapefile** (`.shp`) is an older Esri format that nearly every GIS reads. One "shapefile" is really several files with the same name (`.shp`, `.shx`, `.dbf`, `.prj`, and often others) that must always be kept together. Field names are limited to 10 characters.

For new data, use GeoPackage. To convert a layer, right-click it in the Layers panel and choose **Export ▶ Save Features As…**, pick the format and coordinate system, and give it a file name.

17.5 Adding data

There are three ways to add data to a project:

- **Drag** a file or layer from the Browser panel (or from your computer's file manager) onto the map.
- Open the **Data Source Manager** (Layer ▶ Data Source Manager, or Ctrl+L). Its tabs cover every kind of source: vector files, rasters, delimited text, GeoPackages, databases, and web services.
- Use **Layer ▶ Add Layer** and choose the kind of layer.

When the first layer is added to an empty project, the project takes on that layer's coordinate system. Layers in other coordinate systems are reprojected on the fly to match. Check the project's CRS in the status bar, and change it if it isn't suitable for your map (Chapter 4).

17.5.1 Points from a table of coordinates

Spreadsheets of coordinates are one of the most common ways to get new data into a GIS: field observations, store locations, or stations with a latitude and longitude for each row. Save the table as a CSV file, then open the **Delimited Text** tab of the Data Source Manager (Figure 17.3):

1. Choose the CSV file. QGIS reads the first rows and shows a preview.
2. Under **File Format**, choose CSV.
3. Under **Geometry Definition**, choose **Point coordinates**, and set the **X field** to the longitude and the **Y field** to the latitude.
4. Set the **Geometry CRS** to the coordinate system of the coordinates. Latitude and longitude from a GPS or a web map are almost always WGS 84 (`EPSG:4326`).
5. Click **Add**.

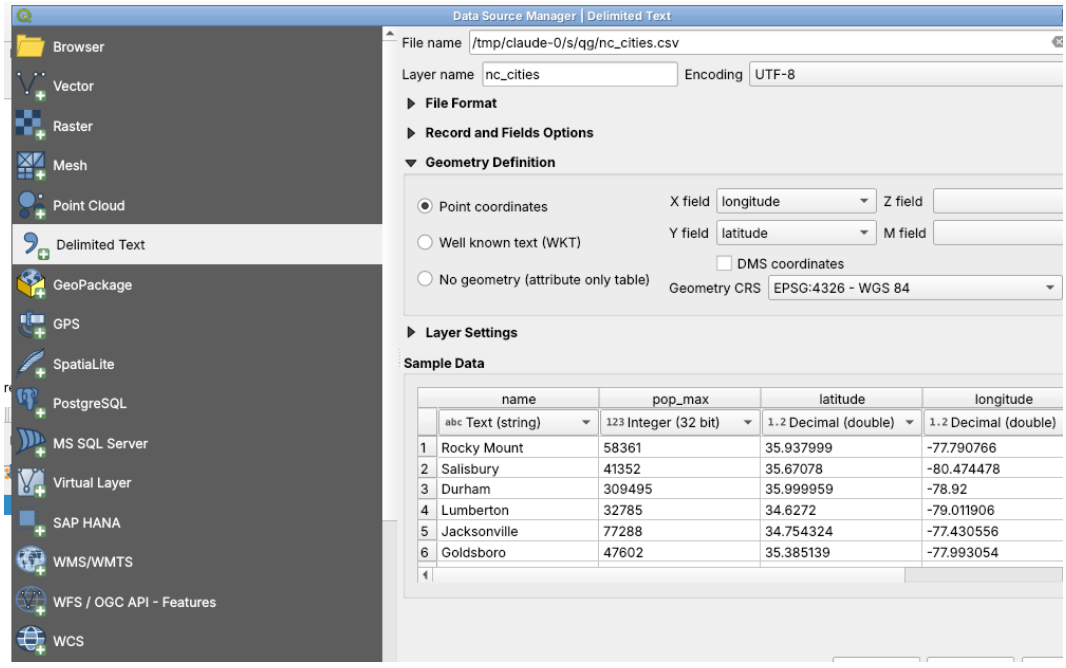


Figure 17.3: The Delimited Text tab of the Data Source Manager, adding a CSV of North Carolina cities with their longitude as the X field, latitude as the Y field, and WGS 84 as the CRS. Screenshot of QGIS 3.34.



> **Think about it**
>
> The most common mistake in adding a CSV is swapping the coordinates. Longitude is **X** (east–west) and latitude is **Y** (north–south). In North Carolina, longitudes are around −75 to −84 and latitudes around 34 to 37. If your points land in Antarctica, or off the coast of Africa near 0°, 0°, check the X and Y fields, and check that western longitudes are negative.

The new layer is a *delimited text layer*, read from the CSV each time. To edit it or use it in analysis, save it as a GeoPackage (Export ▶ Save Features As…).

17.5.2 Basemaps and web services

A **basemap** is a background map, such as streets or aerial imagery, that gives context to your own layers. In the Browser panel, expand **XYZ Tiles** and double-click **OpenStreetMap** to add OpenStreetMap as a basemap. The Browser also connects to web services in standard formats, such as WMS and WMTS (map images), WFS and OGC API – Features (vector data), and ArcGIS REST services. Many state and federal agencies, including NC OneMap, publish their data this way.

17.6 Entering data

Sometimes the data you need doesn't exist yet: field sites, a new trail, or the buildings on a campus map. You create a new layer, then draw (**digitize**) its features and type in their attributes.

17.6.1 Creating a new layer

Choose **Layer ▶ Create Layer ▶ New GeoPackage Layer** (Figure 17.4). Then:

1. **Database**: choose a new or existing `.gpkg` file. One GeoPackage can hold many layers.
2. **Table name**: the name of the new layer, such as `field_sites`.
3. **Geometry type**: Point, Line, Polygon, or one of their multi-part versions. A layer holds only one geometry type.
4. **CRS**: the coordinate system for the layer. Choose a projected coordinate system suited to your area, such as NAD83 / North Carolina (`EPSG:32119`), so that lengths and areas are in meters.
5. **New Field**: type a field name, choose its type, and click **Add to Fields List**. Repeat for each field. The common types are Text (string), Whole Number (integer), Decimal Number (real), and Date.
6. Click **OK**.

QGIS adds an `fid` field automatically. It is the feature ID, a unique number for each feature, which QGIS fills in for you.

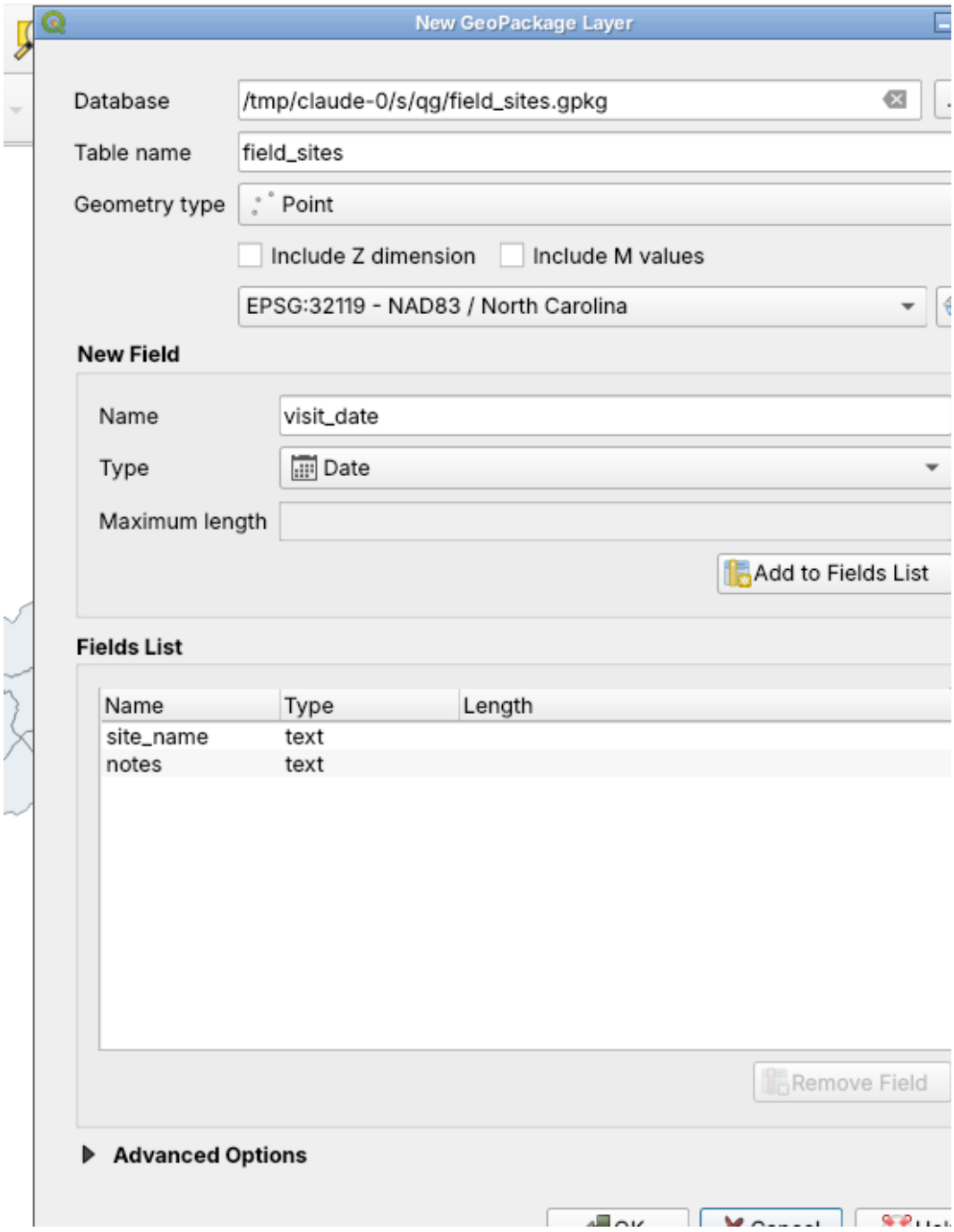


Figure 17.4: The New GeoPackage Layer dialog, creating a point layer called field_sites in the North Carolina State Plane coordinate system, with text fields site_name and notes and a date field visit_date. Screenshot of QGIS 3.34.



> **Think about it**
>
> Choose field types carefully, because they are hard to change later. A ZIP Code or county FIPS code looks like a number, but it should be stored as text: 01003 stored as a number becomes 1003, losing the leading zero, and nobody will ever add two ZIP Codes together. What type would you use for a phone number? A count of trees? A percentage?

17.6.2 Digitizing features

To add features to a layer, you must first put it in edit mode (Figure 17.5):

1. Select the layer in the Layers panel, and click **Toggle Editing** (the pencil). A pencil icon appears next to the layer's name.
2. Click **Add Point Feature** (for a line or polygon layer, **Add Line Feature** or **Add Polygon Feature**).
3. Click the map where the feature goes. For a line or polygon, click once for each vertex, and right-click to finish.
4. An attribute form opens. Type the attributes and click **OK**.

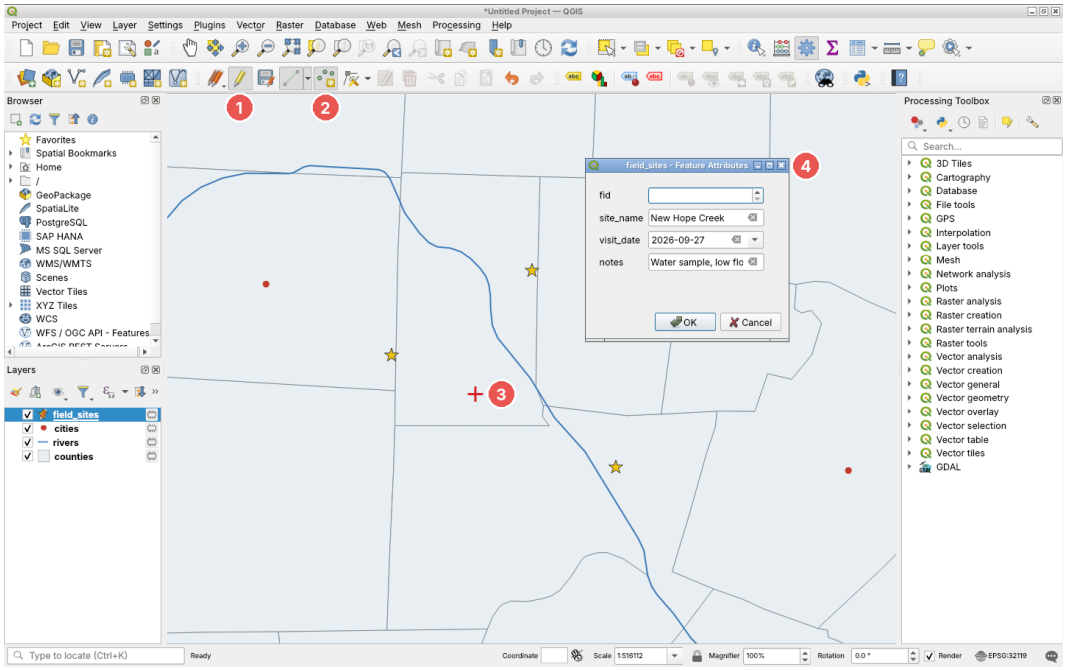


Figure 17.5: Digitizing a new field site: (1) Toggle Editing, (2) Add Point Feature, (3) the clicked location, and (4) the attribute form for the new feature. The stars are sites already entered. Screenshot of QGIS 3.34.

Edits are kept in memory until you save them. Click **Save Layer Edits** to write them to the file, and click **Toggle Editing** again when you are done; QGIS asks whether to save any unsaved edits. Until you save, **Undo** (Ctrl+Z) can take back mistakes.

A few more tools make digitizing accurate:

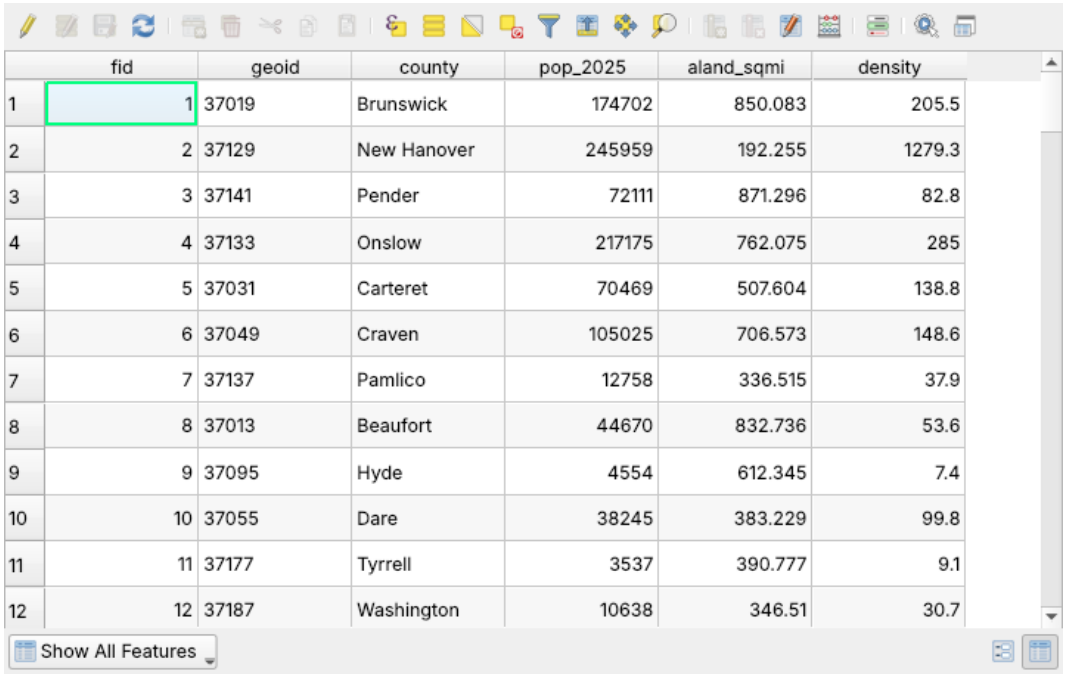
- **Snapping** (Project ▶ Snapping Options, or the magnet on the Snapping toolbar) makes a new vertex jump to a nearby vertex or segment within a tolerance, so that shared boundaries line up exactly and no slivers or gaps form (Chapter 13).
- The **Vertex Tool** moves, adds, and deletes the vertices of existing features.

- Digitizing over a **basemap** or a georeferenced image (Chapter 9) lets you trace features, such as buildings or trails, from imagery.

17.6.3 The attribute table

Every vector layer has an **attribute table**, with one row per feature and one column per field (Figure 17.6). Open it by right-clicking the layer and choosing **Open Attribute Table**, or by pressing F6. In the table you can:

- Sort by any column, by clicking its header.
- Select features, by clicking row numbers. Selected rows are highlighted on the map, too.
- Edit values, once the layer is in edit mode (the pencil at the upper left of the table).
- Add and delete fields, in edit mode.
- Show only selected features, or features matching a filter, using the menu at the lower left.



	fid	geoid	county	pop_2025	aland_sqmi	density
1	1	37019	Brunswick	174702	850.083	205.5
2	2	37129	New Hanover	245959	192.255	1279.3
3	3	37141	Pender	72111	871.296	82.8
4	4	37133	Onslow	217175	762.075	285
5	5	37031	Carteret	70469	507.604	138.8
6	6	37049	Craven	105025	706.573	148.6
7	7	37137	Pamlico	12758	336.515	37.9
8	8	37013	Beaufort	44670	832.736	53.6
9	9	37095	Hyde	4554	612.345	7.4
10	10	37055	Dare	38245	383.229	99.8
11	11	37177	Tyrrell	3537	390.777	9.1
12	12	37187	Washington	10638	346.51	30.7

Show All Features

Figure 17.6: The attribute table of the counties layer, one row for each of North Carolina's 100 counties. Screenshot of QGIS 3.34.

17.6.4 Calculating fields

The **Field Calculator** fills a field with values computed from an **expression**: a formula made of fields, numbers, operators, and functions. Open it from the attribute table's toolbar (the abacus button). You can create a new field or update an existing one (Figure 17.7).

Expressions follow a few rules:

- Field names go in **double quotes**: `"pop_2025"`.
- Text values go in **single quotes**: `"county" = 'Orange'`.
- Arithmetic uses `+ - * /`, and comparison uses `= < > <= >= <>`, as in the SQL queries of Chapter 5.

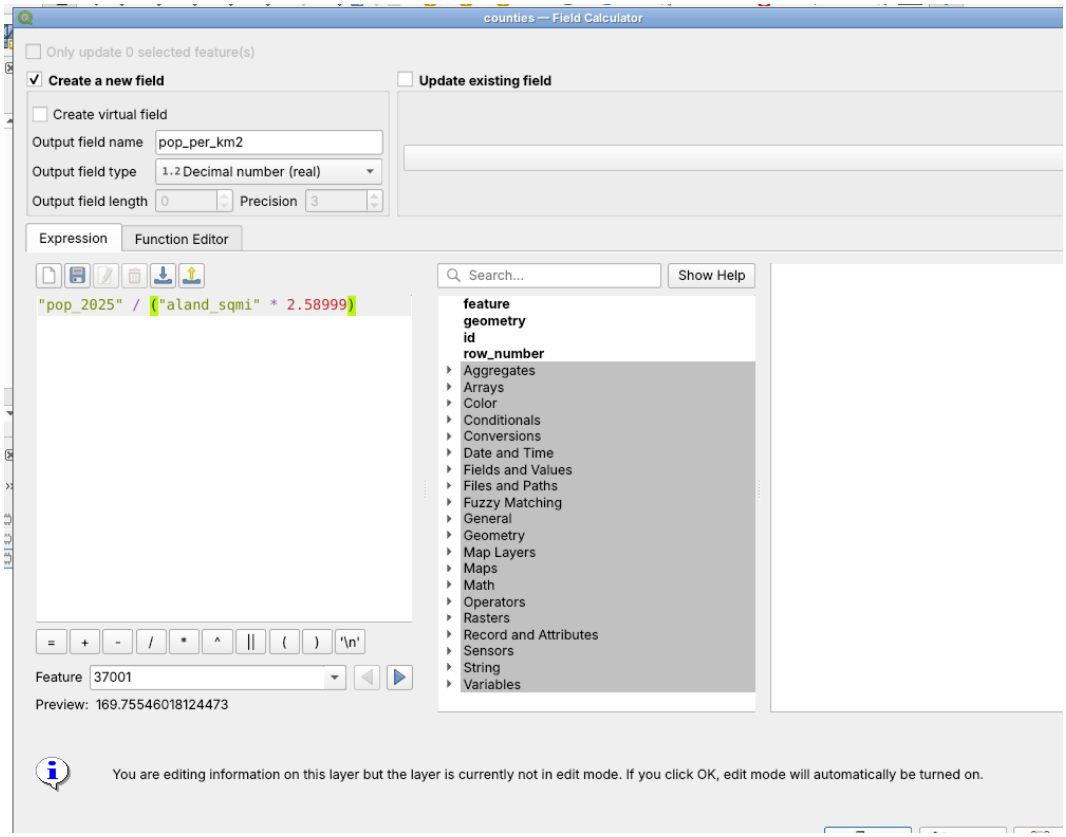


Figure 17.7: The Field Calculator, creating a decimal field, pop_per_km2, from the expression “pop_2025” / (“aland_sqmi” * 2.58999), which converts square miles to square kilometers. The preview shows the result for the first county. Screenshot of QGIS 3.34.

- Geometry functions start with $: `$area` and `$length` give each feature's area and length, in the units set under Project ▶ Properties ▶ General (by default, calculated on the ellipsoid, so they are accurate in any projection).

Expressions are used throughout QGIS, not just in the Field Calculator: to select features (Select Features by Expression), to filter a layer, to label features, and to control symbology. The *data-defined override* button next to many settings lets an expression control that setting, feature by feature.

17.6.5 Joins

To attach a table of attributes, such as census data, to a layer, use the layer's **Properties ▶ Joins** tab, and click **+** to choose the table and the matching fields (Chapter 6). For a join by location, use **Join Attributes by Location** in the Processing Toolbox (Chapter 7).

17.7 Selecting and querying

QGIS offers several ways to select features (Chapter 5):

- The **Select Features** tool on the toolbar selects features by clicking or dragging a rectangle on the map.
- **Select Features by Expression** (the ε on the Selection toolbar) selects features matching an expression, such as `"density" > 500`.
- **Select by Location** (Vector ▶ Research Tools, or the Processing Toolbox) selects features in one layer by their spatial relationship to another, such as the cities within a county.
- A **filter** (right-click the layer ▶ Filter…) hides every feature that doesn't match a query. Unlike a selection, it affects the whole project until you remove it, so be careful: a filtered layer looks like the whole dataset.

To save a selection as a new layer, right-click the layer and choose **Export ▶ Save Selected Features As…**.

17.8 Making a basic map

Making a map in QGIS follows the same steps every time (Figure 17.8): add the data, check the coordinate system, style the layers, label them, and arrange the map with its supporting elements on a page.

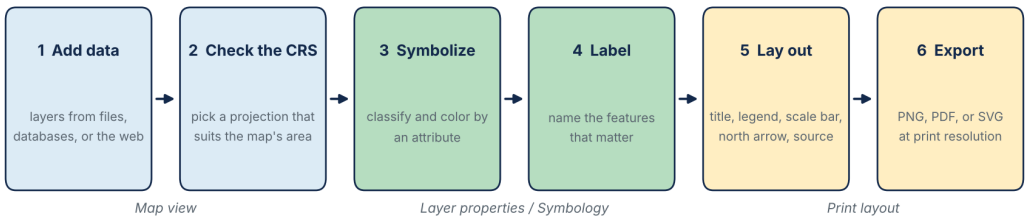


Figure 17.8: The steps of making a basic map, in QGIS or any other GIS.

17.8.1 Symbology

A layer's **symbology** controls how it is drawn. Double-click a layer (or right-click ▶ Properties) and choose the **Symbology** tab (Figure 17.9). The menu at the top chooses the kind of renderer:

- **Single Symbol** draws every feature the same way.
- **Categorized** gives each unique value of a field its own symbol, for nominal data such as land use or road type.
- **Graduated** groups a numeric field into classes and gives each class a color or size, for choropleth maps and proportional symbols.
- **Rule-based** draws features by a set of expressions you write.
- **Heatmap** (for points) draws a kernel density surface (Chapter 12).

To make the choropleth map of population density:

1. Choose **Graduated**, and set **Value** to `density`.
2. Choose a **Color ramp**, such as Greens.
3. Set the **Mode** (the classification method) to Natural Breaks (Jenks) and **Classes** to 5 (Chapter 11).
4. Click **Classify**, then **OK** or **Apply**.

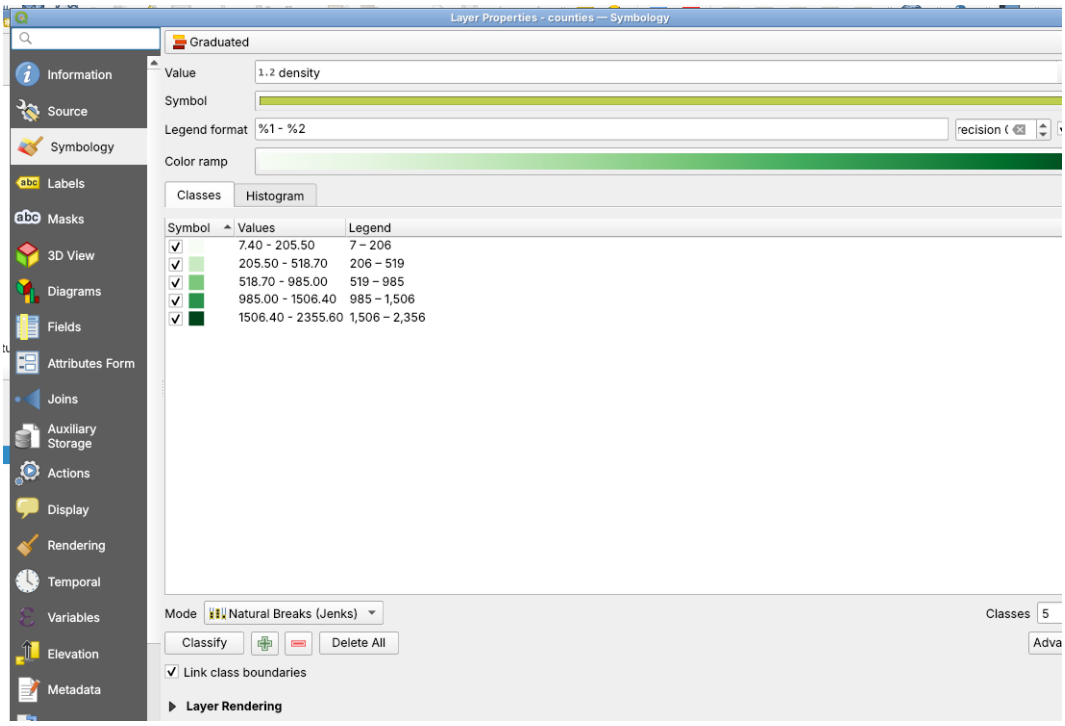


Figure 17.9: The Symbology tab of the counties layer's properties, with a graduated renderer classifying density into five natural-breaks classes on a Greens color ramp. Screenshot of QGIS 3.34.

The **Histogram** tab shows the data's distribution with the class breaks, which helps you judge a classification. You can edit a class's range or legend label by double-clicking it. The **Layer Styling** panel (F7) offers the same settings in a panel, and shows each change on the map as you make it.

17.8.2 Labels

To label features, open the layer's properties and choose the **Labels** tab (Figure 17.10). Choose **Single Labels**, set **Value** to the field to show (here, the city `name`), and set the font, size, and color. Two settings make labels much easier to read:

- A **buffer** (a thin halo of white around each letter) separates the text from the features underneath.
- **Placement** controls where labels go: around points, along lines, or inside polygons.

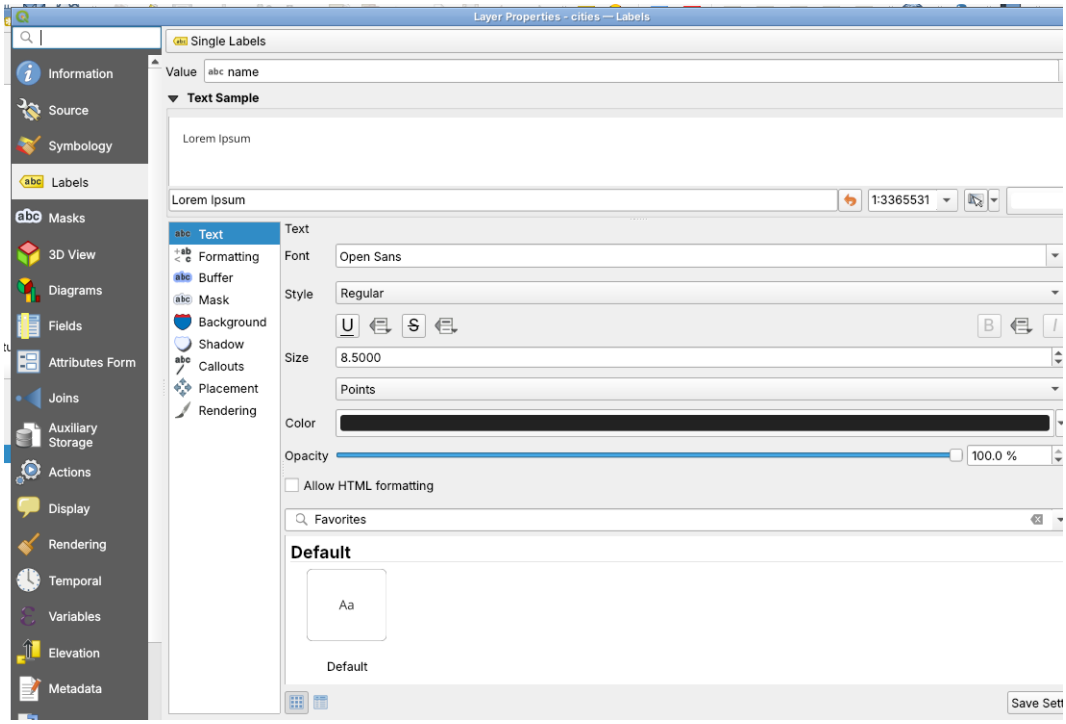


Figure 17.10: The Labels tab of the cities layer, labeling each city with its name in 8.5-point Open Sans. Screenshot of QGIS 3.34.

QGIS places labels to avoid overlaps, and leaves out labels that don't fit. Figure 17.11 shows the result.

17.8.3 The print layout

The map canvas is for working. To make a finished map, for a report, a poster, or a web page, use a **print layout**: a page on which you arrange the map and its supporting elements. Choose **Project ▶ New Print Layout**, give it a name, and the layout window opens (Figure 17.12).

Add each element from the **Add Item** menu (or the toolbar on the left), then drag on the page to place it:

1. **Add Map**: draw a rectangle for the map. It shows the current map canvas. In **Item Properties**, set its scale, or use the **Move Item Content** tool to pan the map inside its frame.
2. **Add Label**: for the title, and again for the source and author. Type the text in Item Properties, and set the font.
3. **Add Legend**: the legend lists every layer. In Item Properties, uncheck **Auto update** to remove layers that don't need explaining, rename entries, and change the title.

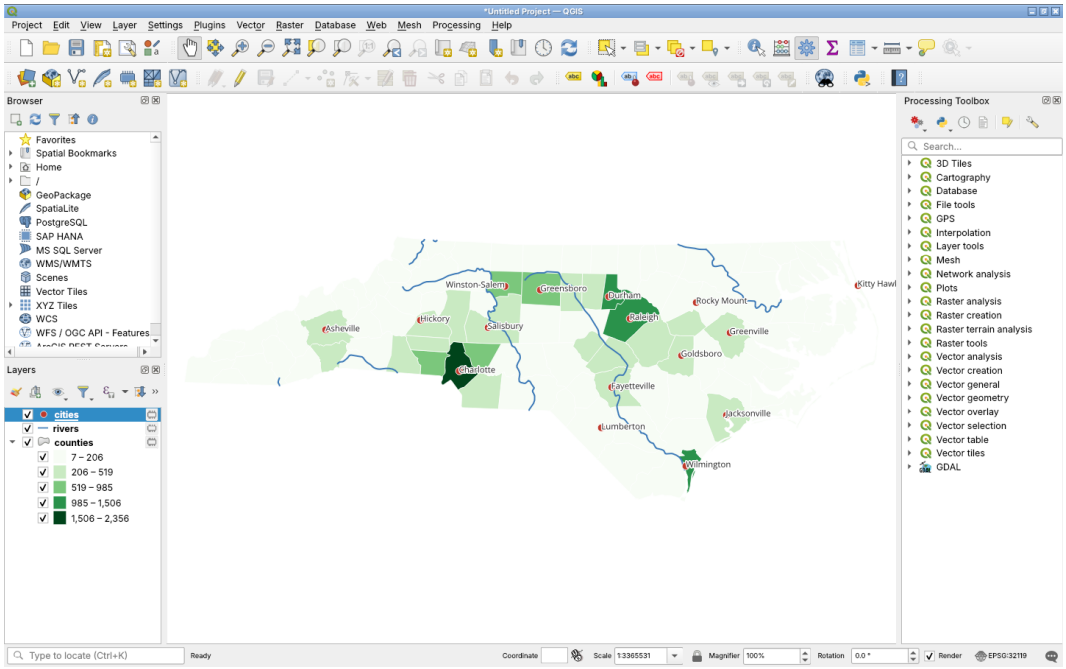


Figure 17.11: The map after styling: counties colored by population density, rivers in blue, and labeled cities. Screenshot of QGIS 3.34.

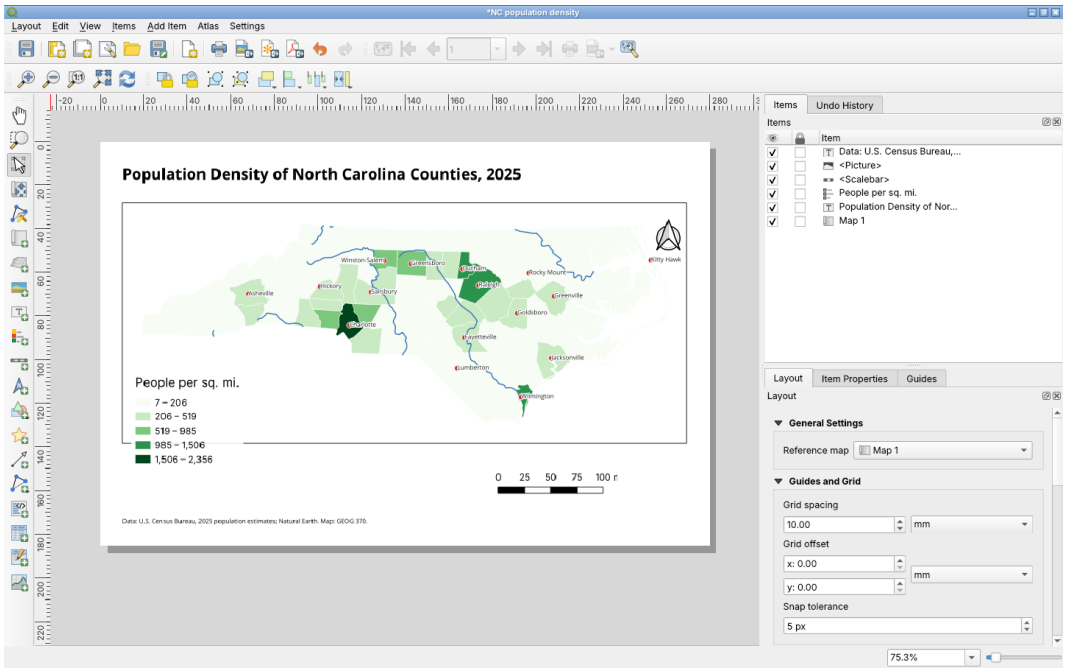


Figure 17.12: A print layout with the map, title, legend, scale bar, north arrow, and source. The Items panel at upper right lists everything on the page; the Item Properties panel below it changes the selected item. Screenshot of QGIS 3.34.

4. **Add Scale Bar**: choose the style, the units (such as miles), and the length of each segment, so that it ends at a round number.
5. **Add North Arrow**: choose an arrow from the gallery of pictures. On a small-scale map in a projection where north varies across the map, it can be left off.

Use the **Items** panel to select, hide, lock, and reorder items. Lock the map once it is placed, so that you don't move it by accident. When the layout is done, export it with **Layout ▶ Export as Image** (PNG), **Export as PDF**, or **Export as SVG** (for further editing in a drawing program). For printing, export at 300 dots per inch. Figure 17.13 is the exported map.

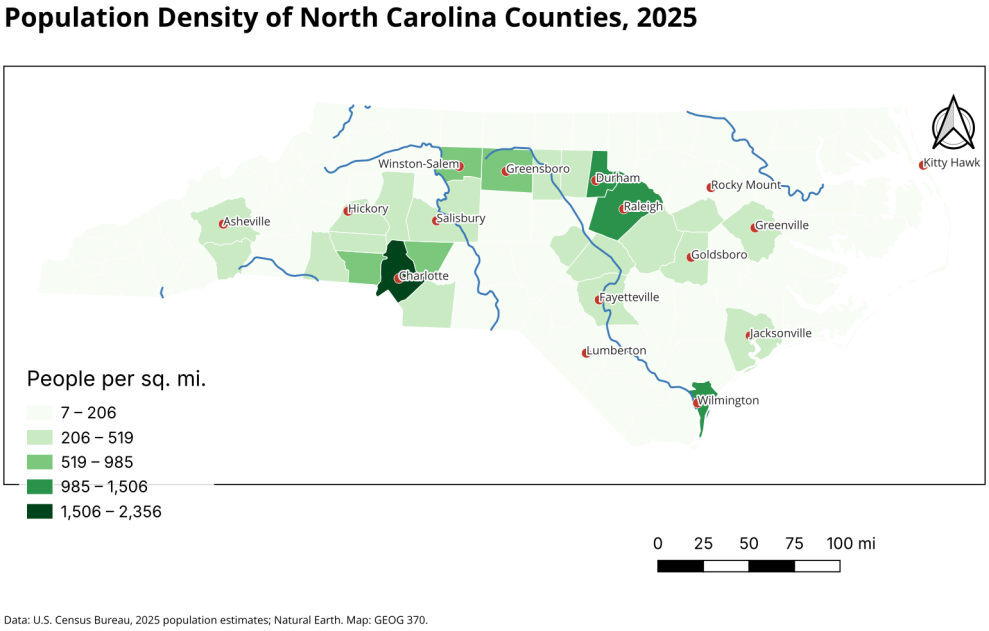


Figure 17.13: The finished map, exported from the print layout as a PNG.

A layout is saved in the project, so you can come back to it, change the data or symbology, and export again. Every element should earn its place: a map for a class or a report almost always needs a title, legend, scale bar, and source, but a decorative north arrow or a busy border can usually go.

17.9 The Processing Toolbox

Most analysis in QGIS happens in the **Processing Toolbox** (Processing ▶ Toolbox, or Ctrl+Alt+T). It holds hundreds of tools, called *algorithms*, grouped by purpose: Vector geometry, Vector overlay, Vector selection, Raster analysis, Interpolation, and more. Besides QGIS's own tools, it includes tools from GDAL and, when installed, from GRASS GIS. The quickest way to find a tool is to type its name in the search box at the top.

Double-click a tool to open its dialog (Figure 17.14):

- The **Parameters** tab asks for the inputs and settings. Each setting has a default, and the help text on the right explains them.
- The output can be a **temporary layer**, kept in memory until you close the project (marked with a chip icon in the Layers panel), or saved to a file. Save outputs you want to keep: a temporary layer disappears when QGIS closes.
- The **Log** tab shows the tool's progress, messages, and errors.

- **Run as Batch Process** runs the tool on many inputs at once.

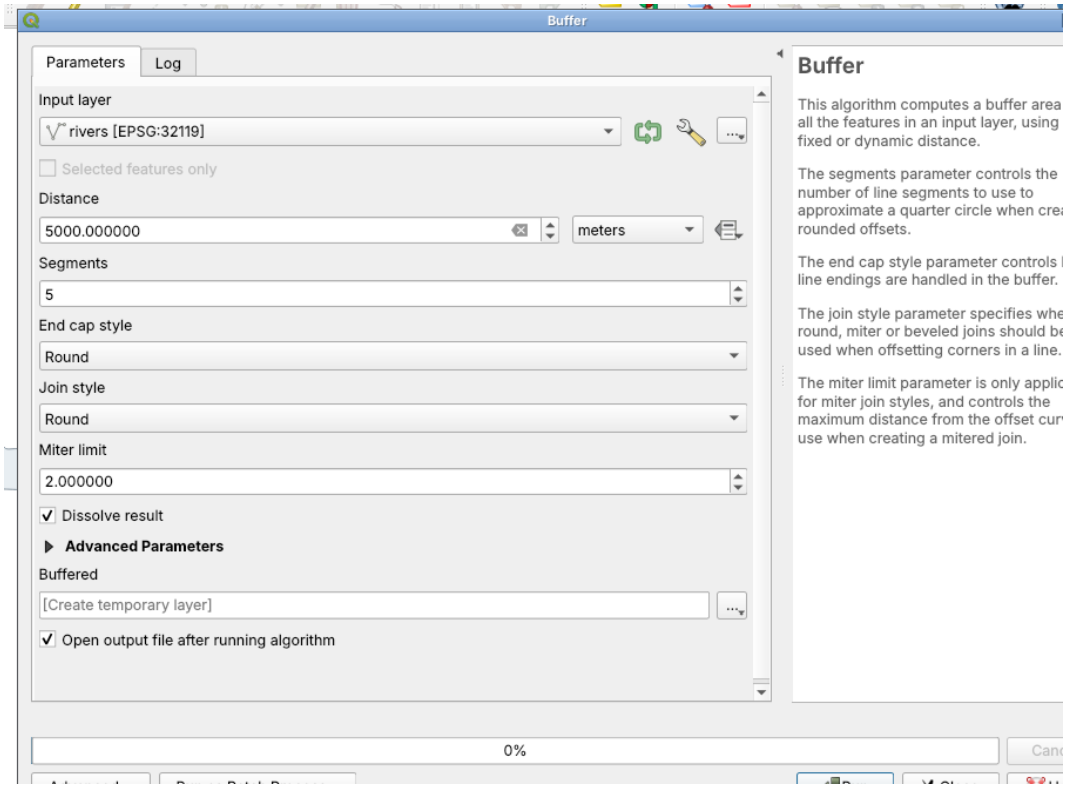


Figure 17.14: The Buffer tool, buffering the rivers by 5,000 meters and dissolving the result into one feature. Screenshot of QGIS 3.34.

Every tool you run is recorded in **Processing ▶ History**, where you can rerun it or see its settings. To chain tools into a repeatable workflow, a cartographic model (Chapter 13), use the **Model Designer** (Processing ▶ Model Designer), which lets you draw the flowchart and run it as a single tool. For more control, the **Python console** (Plugins ▶ Python Console) runs any tool, and almost anything else in QGIS, from Python code.

17.10 Plugins

Plugins add features to QGIS. Open **Plugins ▶ Manage and Install Plugins**, search for a plugin by name or keyword, and click **Install Plugin**. Its tools then appear in a menu, a toolbar, or the Processing Toolbox. A few widely used plugins are:

- **QuickMapServices**, which adds dozens of basemaps.
- **qgis2web**, which exports a project as an interactive web map.
- **MMQGIS**, a collection of tools for geocoding, combining layers, and animation.
- **Profile Tool**, which draws elevation profiles along a line.

Plugins are written by many different people. Some are maintained for years and others are abandoned, so check a plugin's description, rating, and last update before relying on it.

17.11 Getting help

- **Help buttons.** Almost every dialog has a Help button that opens the matching page of the QGIS User Manual.



- **The QGIS documentation** at https://docs.qgis.org includes the User Manual, the *Training Manual* (a full course in QGIS), and *A Gentle Introduction to GIS*.
- **Tutorials** by the community, such as Ujaval Gandhi's *QGIS Tutorials and Tips*, walk through specific tasks step by step.
- **Questions** can be searched and asked on GIS Stack Exchange, with the `qgis` tag.
- **Error messages** appear in the message bar at the top of the map, and in more detail in the Log Messages panel. Searching for the exact text of an error message usually finds someone who has had the same problem.

17.12 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

17.13 Summary

- **QGIS** is a free and **open-source** desktop GIS for Windows, macOS, and Linux, licensed under the GNU GPL and built on shared libraries: GDAL for formats, PROJ for coordinates, GEOS for geometry, and Qt for the interface.
- Gary Sherman started *Quantum GIS* in 2002. It joined OSGeo in 2007, reached 1.0 in 2009, became QGIS with version 2.0 in 2013, and moved to Qt6 with version 4.0 in 2026. A new version comes out every four months; install the **long-term release** (LTR) for stability.
- The QGIS window has a menu bar, toolbars, the **Browser** and **Layers** panels, the map canvas, the **Processing Toolbox**, and a status bar showing the coordinates, scale, and project CRS.
- A **project** (`.qgz`) stores links to data, not the data. Keep projects and data in one folder, use simple file names, and save often. Use **GeoPackage** for new data.
- Add data by dragging from the Browser or with the **Data Source Manager**. Add a CSV of coordinates as delimited text, with longitude as X and latitude as Y.
- To enter data, create a layer, **toggle editing**, **digitize** features, and fill in the attribute form. Edit and compute attributes in the **attribute table** and the **Field Calculator**.
- Make a map by setting **symbology** and **labels** in layer properties, then arranging the map, title, legend, scale bar, north arrow, and source in a **print layout**, and exporting it.
- Run analysis tools from the **Processing Toolbox**, chain them in the Model Designer, and extend QGIS with **plugins**.

17.14 Exercises

1. **Install and explore.** Install the QGIS LTR. Which version and code name did you install? Open the Help ▶ About dialog to check. Find the Browser, Layers, and Processing Toolbox panels, then close one and get it back with View ▶ Panels.
2. **A basemap and your campus.** Add the OpenStreetMap basemap, and zoom to your campus. What is the project's CRS? What is the scale when the whole campus fills the screen?



3. **Points from a spreadsheet.** Make a CSV with the names, latitudes, and longitudes of five places you have visited (look up the coordinates in a web map). Add it as a delimited text layer, check that the points land in the right places, and save it as a GeoPackage.
4. **Digitize.** Create a new GeoPackage polygon layer in NAD83 / North Carolina (`EPSG:32119`) with a text field `name`. Digitize three buildings on campus over the basemap, and add an `area_m2` field calculated with `$area`. Which building is largest?
5. **A choropleth map.** Download North Carolina county boundaries and a table of county populations (Chapter 8). Join them, calculate population density, and make a graduated map with five classes. Compare natural breaks and quantile classes (Chapter 11). Then make a print layout with a title, legend, scale bar, and source, and export it as a PDF.
6. **A plugin.** Install a plugin from the list in Section 17.10 or one you find yourself. What does it do? Who wrote it, and when was it last updated?

17.15 Further reading and online resources

17.15.1 QGIS documentation

- QGIS project home and downloads: https://qgis.org.
- QGIS User Manual (latest LTR): https://docs.qgis.org/latest/en/docs/user_manual/.
- QGIS Training Manual, a complete self-paced course: https://docs.qgis.org/latest/en/docs/training_manual/.
- *A Gentle Introduction to GIS*, from the QGIS documentation: https://docs.qgis.org/latest/en/docs/gentle_gis_introduction/.
- QGIS User Manual, "Working with Vector Data" (symbology, labels, attribute tables, editing): https://docs.qgis.org/latest/en/docs/user_manual/working_with_vector/index.html.
- QGIS User Manual, "Laying out the maps" (print layouts): https://docs.qgis.org/latest/en/docs/user_manual/print_layout/index.html.
- QGIS User Manual, "QGIS processing framework": https://docs.qgis.org/latest/en/docs/user_manual/processing/index.html.
- QGIS release schedule and roadmap: https://www.qgis.org/resources/roadmap/.
- QGIS plugin repository: https://plugins.qgis.org.
- QGIS changelogs, with illustrated lists of new features in each version: https://changelog.qgis.org.

17.15.2 Tutorials and courses

- *QGIS Tutorials and Tips*, by Ujaval Gandhi: https://www.qgistutorials.com/en/.
- Spatial Thoughts, free QGIS courses including *Introduction to QGIS*: https://courses.spatialthoughts.com.
- *Intro to GIS and Spatial Analysis*, by Manuel Gimond, free online, with QGIS examples: https://mgimond.github.io/Spatial/.

17.15.3 Community and history

- QGIS on Wikipedia, including a table of every version and its code name: https://en.wikipedia.org/wiki/QGIS.



- The Open Source Geospatial Foundation (OSGeo): https://www.osgeo.org.
- GIS Stack Exchange, questions tagged `qgis`: https://gis.stackexchange.com/questions/tagged/qgis.
- QGIS source code on GitHub: https://github.com/qgis/QGIS.
- GDAL, the format library behind QGIS: https://gdal.org.
- PROJ, the coordinate transformation library: https://proj.org.
- QField, the mobile app for collecting data with QGIS projects in the field: https://qfield.org.

17.15.4 Data used in this chapter

- North Carolina county populations, U.S. Census Bureau Population Estimates Program (see Chapter 11).
- County boundaries, cities, and rivers: Natural Earth, public domain: https://www.naturalearthdata.com.

18 Getting to Know ArcGIS Pro

> **Learning objectives**
>
> By the end of this chapter you should be able to:
>
> 1. Describe ArcGIS Pro and the ArcGIS family of software, and how Esri's desktop GIS developed from ARC/INFO to ArcMap to ArcGIS Pro.
> 2. Explain how ArcGIS Pro is licensed, and how to install it and sign in.
> 3. Find your way around the ArcGIS Pro window: the ribbon, views, and panes.
> 4. Explain what an ArcGIS Pro project contains, and organize a project and its data.
> 5. Add data, create a feature class, edit features, and calculate attributes.
> 6. Make a basic map with symbology, labels, and a layout, and run tools in the Geoprocessing pane.

ArcGIS Pro is the desktop GIS made by **Esri**, the largest GIS software company in the world. It is used by governments, utilities, businesses, and universities, so it appears in a large share of job postings for GIS analysts. Like QGIS (Chapter 17), it can do nearly everything described in this book, and most tasks follow the same steps in both. This chapter introduces ArcGIS Pro: its history, how it is licensed, how its window is organized, and how to do the common tasks of entering data and making a map.

The figures in this chapter are diagrams of the ArcGIS Pro window, not screenshots, so they show where things are rather than every button. Esri's own documentation, listed at the end of the chapter, has screenshots of the current version. Commands are written as **Tab ▶ Group ▶ Button**, following the ribbon.

18.1 ArcGIS and ArcGIS Pro

ArcGIS is not one program but a family of products that share data formats and accounts:

- **ArcGIS Pro**, the desktop application, for editing, analysis, and cartography. It is the subject of this chapter.
- **ArcGIS Online**, a cloud service for storing data, making web maps and apps, and sharing them, run by Esri.
- **ArcGIS Enterprise**, the same kinds of services, run on an organization's own servers.
- **Field and web apps**, such as ArcGIS Field Maps (for collecting data on a phone), ArcGIS Survey123 (forms), ArcGIS StoryMaps (stories built around maps), and ArcGIS Dashboards.
- **Developer tools**, such as **ArcPy**, a Python library for running ArcGIS tools from scripts, and APIs for building web and mobile apps.

ArcGIS Pro is **proprietary** software: Esri owns it, its source code is private, and using it requires a paid license. It runs only on **Windows**. On a Mac, you can run it in a Windows virtual machine or use a university's virtual computer lab.

ArcGIS Pro is a 64-bit program that works with 2D maps and 3D scenes side by side, and its tools are arranged on a **ribbon**, like Microsoft Office. It is closely tied to ArcGIS Online: you sign in with an ArcGIS account to use it, and you can share maps and layers from it to the web in a few clicks.

18.2 A short history

Esri (originally the Environmental Systems Research Institute) was founded in 1969 by Jack and Laura Dangermond in Redlands, California, as a land-use consulting firm. Jack Dangermond had studied at Harvard's Graduate School of Design, near the Harvard Laboratory for Computer Graphics and Spatial Analysis, one of the birthplaces of GIS (Chapter 1). Esri's early work was GIS projects for government clients, and the software it wrote for them became its products (Figure 18.1).

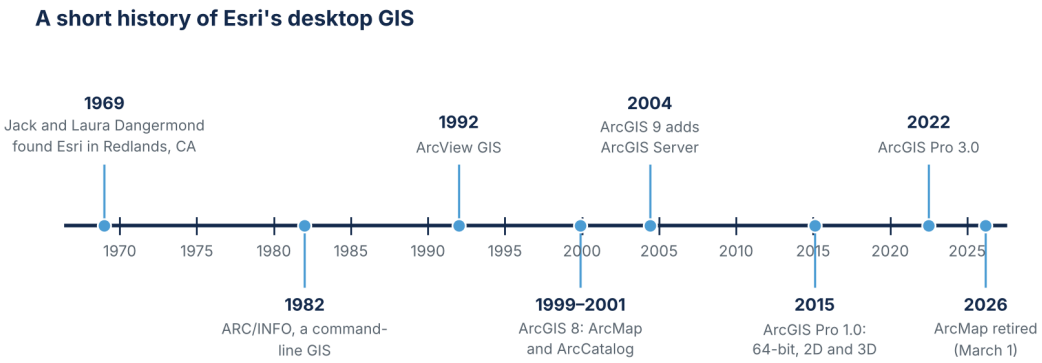


Figure 18.1: Milestones in the history of Esri's desktop GIS software.

- **1982: ARC/INFO.** Esri's first major product was a command-line GIS. Users typed commands to build and analyze *coverages*, a vector format that stored topology. ARC/INFO ran on minicomputers and workstations, and dominated professional GIS through the 1980s and 1990s.
- **1992: ArcView GIS.** A desktop program with a graphical interface, for viewing, querying, and mapping data. ArcView introduced the **shapefile**, a simple, openly documented format that became the most widely shared vector format in the world (Chapter 2).
- **1999–2001: ArcGIS 8.** Esri rebuilt its software as a single system for Windows, called **ArcGIS Desktop**. Its main applications were **ArcMap**, for mapping, editing, and analysis, and **ArcCatalog**, for managing data. It also introduced the **geodatabase**, Esri's format for storing many layers and tables together.
- **2004 onward: ArcGIS 9 and 10.** Later versions added ArcGIS Server, for publishing maps and data on the web, and ArcGIS Online, Esri's cloud platform. ArcMap 10.x became the standard desktop GIS in many offices for most of two decades.
- **2015: ArcGIS Pro 1.0.** A new, 64-bit desktop application, released alongside ArcMap, with integrated 2D and 3D, a ribbon interface, multiple maps and layouts in one project, and close ties to ArcGIS Online.
- **2022: ArcGIS Pro 3.0.** A major update. A project saved in version 3 can no longer be opened in version 2.
- **2026: ArcMap retired.** In 2020, Esri announced that ArcMap 10.8 would be its last version, and that ArcMap would be supported until March 1, 2026. ArcGIS Pro is now Esri's only desktop GIS.



You will still find ArcMap in older tutorials, books, and offices. Its ideas carry over to ArcGIS Pro, but its menus and toolbars do not, so look for tutorials written for Pro.

Did you know..?

Esri was started in 1969 with $1,100 of the Dangermonds' savings, and is still privately owned by its founders. Its first user conference, held in Redlands in 1981, had 16 attendees. Today the Esri User Conference fills the San Diego Convention Center every July, and is one of the largest GIS gatherings in the world.

18.3 Getting ArcGIS Pro

18.3.1 Licensing

Using ArcGIS Pro requires a license, which comes at one of three **license levels**. Each level includes everything in the levels below it:

Table 18.1: ArcGIS Pro license levels.

Level	Formerly	What it adds
Basic	ArcView	Viewing, mapping, editing simple data, and most analysis tools
Standard	ArcEditor	Editing of advanced geodatabase features, such as topology and versioned data shared by many editors
Advanced	ArcInfo	The full set of geoprocessing tools, including advanced overlay and generalization

Extensions, such as Spatial Analyst (raster analysis, Chapter 14), Geostatistical Analyst (kriging, Chapter 16), 3D Analyst, and Network Analyst, add more tools, and are licensed separately.

In an ArcGIS organization, the level comes with the account's *user type*: Basic with the Creator user type, Standard with Professional, and Advanced with Professional Plus. Most users today have a **named user license**: the license is assigned to your ArcGIS account by your organization (such as a university or employer), and ArcGIS Pro checks it when you sign in. Universities usually have a campus license that covers students, with the extensions included. Esri also sells personal-use licenses at low cost for non-commercial use, and offers free trials.

Some tools and services, such as geocoding many addresses with Esri's World Geocoding Service, demographic enrichment, and storing large amounts of data in ArcGIS Online, use **credits**, a currency that your organization buys and allots to its users. Tools that use credits say so in their documentation.

18.3.2 Installing and signing in

1. Check that your computer meets the system requirements: for current versions, 64-bit Windows 11, with at least 8 GB of memory (32 GB is recommended).
2. Download ArcGIS Pro from your ArcGIS account (my.esri.com, or the link your organization gives you) and run the installer.
3. Start ArcGIS Pro and **sign in** with your organization's ArcGIS account. University accounts usually sign in through the university's own login page: choose *Your ArcGIS organization's URL* and enter it.

ArcGIS Pro is updated a few times a year. The version number is in **Project ▶ About ArcGIS Pro**.

18.4 The ArcGIS Pro window

When ArcGIS Pro starts, it shows a start page for opening a recent project or creating a new one from a template. The **Map** template creates a project with one empty map, which is the usual place to start. Figure 18.2 shows the main window.



Figure 18.2: The parts of the ArcGIS Pro window: (1) the Quick Access Toolbar and title bar, (2) ribbon tabs, including contextual tabs, (3) the ribbon, with commands in groups, (4) the Contents pane, (5) the view, here a map, (6) the Catalog pane, which shares its space with other panes, and (7) the status bar under the map, with the scale, coordinates, and number of selected features. A schematic drawing, not a screenshot.

1. The **Quick Access Toolbar** at the top left has buttons for saving, undoing, and opening projects. The title bar also holds **Command Search** (Alt+Q), where you can type the name of any command or tool to find it, and your sign-in name.
2. **Ribbon tabs** organize the commands: **Project** (open, save, options, and licensing), **Map** (navigating, adding data, selecting), **Insert** (new maps, layouts, and toolboxes), **Analysis** (geoprocessing tools), **View** (panes and windows), **Edit**, **Imagery**, **Share**, and



Help. **Contextual tabs**, in a different color, appear only when something is selected. Select a layer in the Contents pane, for example, and the **Feature Layer**, **Labeling**, and **Data** tabs appear.

3. The **ribbon** shows the commands on the current tab, arranged in **groups**, such as Navigate, Layer, and Selection on the Map tab.
4. The **Contents pane** lists the layers in the active map, in drawing order: layers at the top draw on top. Check a box to show or hide a layer, drag to reorder, and right-click a layer for its options.
5. The **view** is where you work with a map, a 3D scene, a layout, or a table. Each opens as a tab, and you can have many open at once, side by side or stacked.
6. The **Catalog pane** shows everything in the project: its maps, layouts, toolboxes, databases, styles, and folder connections. Other panes, such as **Geoprocessing**, **Symbology**, **Create Features**, and **Element**, open in the same space as tabs.
7. The bottom of each map view shows the **scale**, the **coordinates** under the cursor, and the number of **selected features**.

Panes can be docked, floated, or closed. To reopen a pane, find it on the **View** tab, or search for it with Command Search.

18.4.1 Navigating the map

Table 18.2: Navigating a map in ArcGIS Pro.

To do this	Use
Pan and zoom	**Map** ▶ **Navigate** ▶ **Explore**, then drag to pan and roll the wheel to zoom
Show a whole layer	Right-click the layer ▶ **Zoom To Layer**
Show all layers	**Map** ▶ **Navigate** ▶ **Full Extent**
Go back to the previous view	**Map** ▶ **Navigate** ▶ **Previous Extent**
See a feature's attributes	Click the feature with the **Explore** tool; a pop-up shows its attributes
Measure a distance or area	**Map** ▶ **Inquiry** ▶ **Measure**

18.5 Projects

Work in ArcGIS Pro is saved as a **project**. When you create a project, ArcGIS Pro makes a **project folder** with the project file and a few default items (Figure 18.3):

- The **project file** (`.aprx`) stores the project's maps, layouts, and settings, and *links* to its data. Like a QGIS project, it does not contain the data itself.
- The **default geodatabase** (`.gdb`), named after the project, is where new data and tool outputs are saved unless you choose otherwise.
- The **default toolbox** (`.atbx`) holds models and script tools you create.
- Other folders, such as `ImportLog` and `Index`, are used by ArcGIS Pro itself.

The habits that keep QGIS projects working apply here too: keep each project and its data together in one folder, use short names without spaces, and work on a local drive rather than a USB drive or a cloud-synced folder. Save often with Ctrl+S. If a layer's data moves, a red exclamation mark appears next to it in the Contents pane; click it to locate the data again.

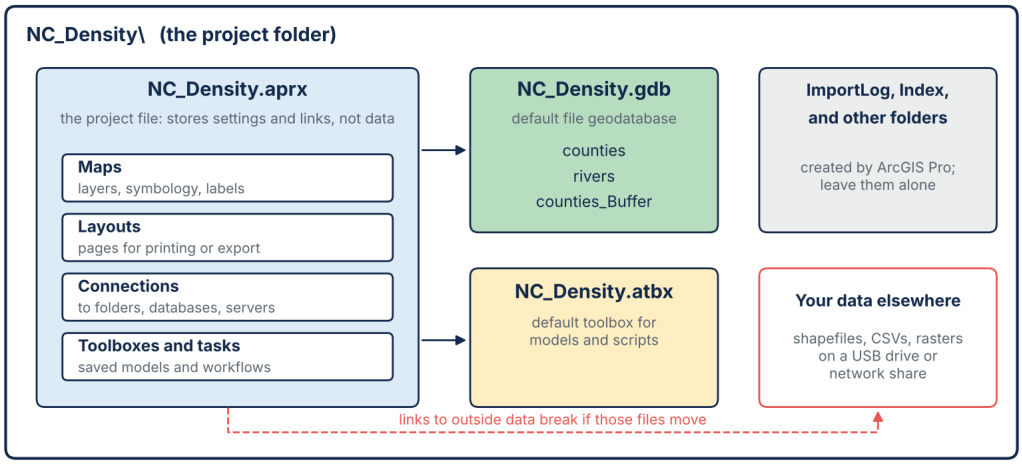


Figure 18.3: What an ArcGIS Pro project folder contains. The project file stores maps, layouts, connections, and toolboxes, and links to data in the default geodatabase and elsewhere.

18.5.1 The geodatabase

A **geodatabase** is Esri's format for storing geographic data. A **file geodatabase** is a folder ending in `.gdb` that holds any number of **feature classes** (vector layers), **tables**, and **raster datasets**. It has advantages over shapefiles: long field names, large sizes, null values, and advanced features such as domains (lists of valid values), topology, and relationship classes. Use the project's default geodatabase for your new data. Don't open, copy, or rename the files *inside* a `.gdb` folder with Windows File Explorer; manage them in the Catalog pane.

A geodatabase is not readable by every program the way a shapefile or GeoPackage is. QGIS can read file geodatabases, and newer versions can edit them, but to share data with someone who doesn't use ArcGIS, export it to a GeoPackage or shapefile.

18.6 Adding data

- **Map ▶ Layer ▶ Add Data** opens a browse dialog for files, geodatabases, and data in ArcGIS Online.
- **Drag** a feature class or file from the Catalog pane onto the map.
- To browse a folder outside the project, right-click **Folders** in the Catalog pane and choose **Add Folder Connection**.

When the first layer is added to a new map, the map takes on that layer's coordinate system, and later layers are projected on the fly to match. To change it, right-click the map's name in the Contents pane, choose **Properties**, and open the **Coordinate Systems** tab.

18.6.1 Points from a table of coordinates

To make points from a CSV or Excel table of coordinates, add the table to the map (it appears under *Standalone Tables* in the Contents pane), right-click it, and choose **Create Points From Table ▶ XY Table To Point**. The tool asks for:

- **X Field**: the longitude (or easting).



- **Y Field**: the latitude (or northing).
- **Coordinate System**: the system the coordinates are in. For latitude and longitude from a GPS or web map, this is usually GCS WGS 1984.

The output is a new point feature class in the project geodatabase. As in QGIS, the most common mistake is swapping X and Y: longitude is X.

18.6.2 Basemaps

Map ▶ Layer ▶ Basemap opens a gallery of basemaps from ArcGIS Online, such as Topographic, Imagery, Streets, and Light Gray Canvas. New maps start with a basemap, usually World Topographic Map, which appears at the bottom of the Contents pane. Basemaps require an internet connection and sign-in.

18.7 Entering data

18.7.1 Creating a feature class

To create a new, empty layer, right-click the project geodatabase in the Catalog pane and choose **New ▶ Feature Class**. A wizard walks through the steps:

1. **Name** the feature class (such as `field_sites`) and choose its **feature class type**: point, line, polygon, or another type.
2. **Fields**: add fields, giving each a name and a **data type**, such as Text, Short or Long (integers), Float or Double (decimal numbers), or Date. ArcGIS adds an `OBJECTID` field (the feature ID) and, for lines and polygons, `Shape_Length` and `Shape_Area` fields, which it keeps up to date.
3. **Spatial Reference**: choose the coordinate system, such as NAD 1983 StatePlane North Carolina FIPS 3200 (Meters).
4. Click **Finish**. The new, empty layer is added to the map.

The **Create Feature Class** tool in the Geoprocessing pane does the same thing.

18.7.2 Editing features

Editing in ArcGIS Pro is done from the **Edit** tab:

1. Click **Edit ▶ Features ▶ Create**. The **Create Features** pane lists a *template* for each editable layer.
2. Click the template for the layer you want, such as `field_sites`, and choose a construction tool (for a point layer, Point).
3. Click the map to place a point. For a line or polygon, click each vertex, and double-click (or press F2) to finish.
4. Fill in the new feature's attributes in the **Attributes** pane (**Edit ▶ Selection ▶ Attributes**).
5. Click **Edit ▶ Manage Edits ▶ Save** to save the edits, or **Discard** to throw them away.

Other tools on the Edit tab move, reshape, split, and merge features. **Snapping** (**Edit ▶ Snapping**) makes new vertices jump to existing vertices and edges, so that shared boundaries line up.

18.7.3 The attribute table

Right-click a layer and choose **Attribute Table** (or select the layer and press Ctrl+T) to open its table as a view below the map. Click a row's gray box at the left to select the feature; double-click a cell to edit it. The table's menu buttons add fields, calculate fields, select by attributes, and switch between all and selected records.

18.7.4 Calculating fields

Calculate Field (right-click a column heading in the table, or find the tool in the Geoprocessing pane) fills a field with an expression. You can write the expression in **Python** or in **Arcade**, Esri's own expression language, which also works in ArcGIS Online:

- In Python, field names are wrapped in exclamation marks: `!pop_2025! / (!aland_sqmi! * 2.58999)`.
- In Arcade, fields are properties of `$feature`: `$feature.pop_2025 / ($feature.aland_sqmi * 2.58999)`.

To add a field first, use **Add Field** on the table's menu, or **Data ▶ Design ▶ Fields**, which shows all the layer's fields and their types. **Calculate Geometry** (right-click a column heading) fills a field with each feature's area, length, or coordinates, in the units and coordinate system you choose.

18.7.5 Joins

To join a table to a layer by a common field (Chapter 6), right-click the layer and choose **Joins and Relates ▶ Add Join**. For a spatial join (Chapter 7), use the **Spatial Join** tool or **Add Spatial Join**.

18.8 Selecting and querying

The **Map ▶ Selection** group holds the selection tools (Chapter 5):

- **Select** selects features by clicking or dragging on the map.
- **Select By Attributes** selects features with a query built from clauses, such as `density is greater than 500`, or written in SQL.
- **Select By Location** selects features by their spatial relationship to another layer.
- **Clear** clears the selection.

The number of selected features is shown at the bottom of the map and of the table. Most tools run on the selected features only, when a layer has a selection, so clear selections you no longer need. To keep a selection as a new layer, right-click the layer and choose **Data ▶ Export Features**.

A **definition query** (layer **Properties ▶ Definition Query**) hides every feature that doesn't match a query. Like a QGIS filter, it affects everything you do with the layer.

18.9 Making a basic map

The steps of making a map are the same as in QGIS (Figure 18.4): add data, check the coordinate system, symbolize, label, lay out, and export.

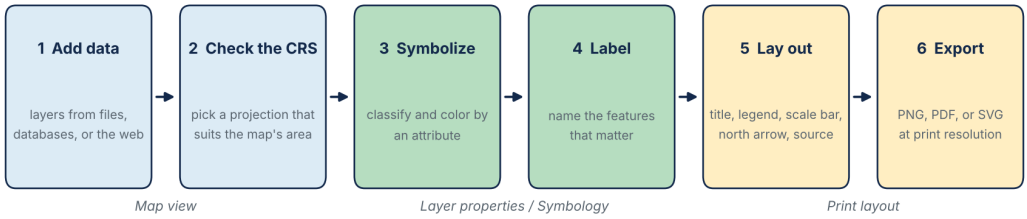


Figure 18.4: The steps of making a basic map, in ArcGIS Pro or any other GIS.

18.9.1 Symbology

Right-click a layer and choose **Symbology** to open the **Symbology pane**. The **Primary symbology** menu chooses how features are drawn:

- **Single Symbol**: every feature the same.
- **Unique Values**: a symbol for each value of a field, for categories.
- **Graduated Colors**: classes of a numeric field shown in a color scheme, for choropleth maps.
- **Graduated Symbols** and **Proportional Symbols**: symbol sizes that vary with a numeric field.
- **Dot Density**, **Charts**, **Heat Map**, and others.

For the map of population density, choose **Graduated Colors**, set **Field** to `density`, set **Method** to Natural Breaks (Jenks) and **Classes** to 5, and pick a **Color scheme**. The **Histogram** tab shows the class breaks on the data's distribution (Chapter 11). Changes appear on the map as you make them.

18.9.2 Labels

Select the layer in the Contents pane, and click the **Labeling** contextual tab. Click **Label** to turn labels on, choose the **Field** (such as `name`), and set the font, size, and color in the **Text Symbol** group. A halo (Esri's name for a buffer around the text) makes labels readable over busy backgrounds. To add one, click the small launcher arrow at the corner of the **Text Symbol** group, which opens the **Label Class** pane, and set the **Halo** under the symbol's appearance settings.

18.9.3 Layouts

A **layout** is a page for printing or exporting a finished map.

1. **Insert ▶ Project ▶ New Layout**, and choose a page size and orientation, such as Letter, landscape. The layout opens as a new view.



2. **Insert ▶ Map Frames ▶ Map Frame**, choose the map, and drag a rectangle on the page. To change what the frame shows, use **Layout ▶ Map ▶ Activate**, navigate the map, then close the activated view.
3. **Insert ▶ Graphics and Text ▶ Text** (or **Dynamic Text ▶ Title**) for the title, source, and author.
4. **Insert ▶ Map Surrounds ▶ Legend**, **Scale Bar**, and **North Arrow**. Each is linked to the map frame and updates when the map changes.
5. Select any element and use the **Format** pane (or the **Element** pane) to change its properties: legend items, scale bar units and divisions, fonts, and sizes.
6. **Share ▶ Output ▶ Export Layout** to export the layout as a PDF, PNG, JPEG, SVG, or other format, at the resolution you choose. **Share ▶ Output ▶ Print Layout** prints it.

A project can hold many maps and many layouts, all listed in the Catalog pane.

18.10 Geoprocessing

Geoprocessing is Esri's term for running tools that create or change data: buffers, overlays, conversions, and analysis of every kind (Chapters 13–16). The tools are organized in **toolboxes**, such as the Analysis, Data Management, Conversion, and Spatial Analyst toolboxes.

Open the **Geoprocessing pane** with **Analysis ▶ Geoprocessing ▶ Tools**. Type a tool's name in the search box, and click it to open it (Figure 18.5):

- **Required parameters** are marked with a red asterisk. A tool can't run until they are filled in.
- The **output** is saved by default to the project geodatabase, with a name built from the input and the tool.
- **Environments** set rules for every tool, such as the output coordinate system, the processing extent, and the cell size for raster outputs.
- **Run** starts the tool. A green check means it succeeded, and **View Details** shows its messages; a red X means it failed, and the messages explain why.

Every tool you run is recorded in the project's **History** (**Analysis ▶ Geoprocessing ▶ History**), where you can open it again with the same settings. To chain tools into a repeatable workflow, use **ModelBuilder** (**Analysis ▶ Geoprocessing ▶ ModelBuilder**), which draws the workflow as a flowchart (Chapter 13). For scripting, the **Python window** (**Analysis ▶ Geoprocessing ▶ Python**) and **ArcGIS Notebooks** run tools with **ArcPy**. Every tool's help page shows the Python code to run it.

> **Think about it**
>
> ArcGIS Pro has both a **Buffer** and a **Pairwise Buffer** tool, and both a **Dissolve** and a **Pairwise Dissolve**. The "pairwise" tools are newer versions that are faster on large datasets and use all of your computer's processor cores. Why might Esri keep both versions, instead of replacing the old tools?

18.11 Sharing to the web

Because ArcGIS Pro is tied to ArcGIS Online, you can publish your work to the web. **Share ▶ Share As ▶ Web Map** shares a map, with its layers and symbology, as a web map that

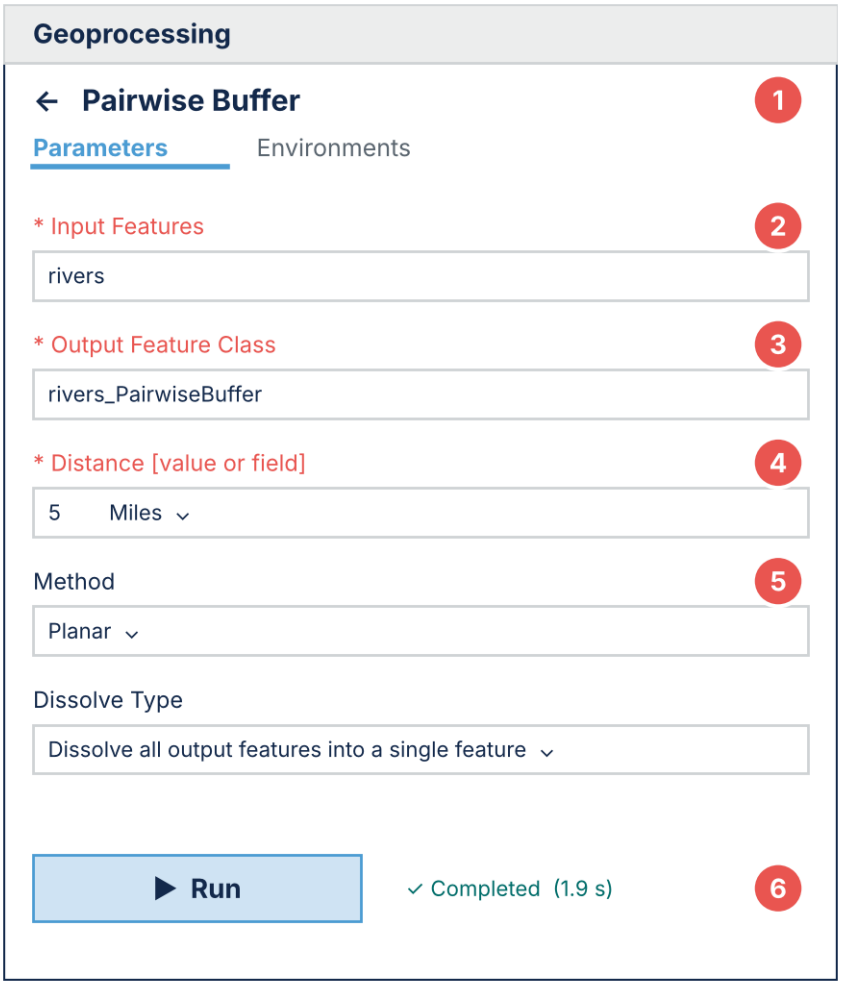


Figure 18.5: The Geoprocessing pane with the Pairwise Buffer tool: (1) the tool's name, (2) the input layer, (3) the output feature class, (4) the buffer distance, (5) optional parameters, and (6) the Run button and status. A schematic drawing, not a screenshot.



anyone you choose can open in a browser; **Web Layer** shares a single layer. Web maps can then be used in ArcGIS StoryMaps, Dashboards, and apps. You choose who can see each item: just you, groups, your organization, or everyone.

18.12 QGIS and ArcGIS Pro compared

Table 18.3: QGIS and ArcGIS Pro side by side.

	QGIS	ArcGIS Pro
Cost	Free	Paid license (usually through an employer or university)
License	Open source (GNU GPL)	Proprietary
Operating systems	Windows, macOS, Linux	Windows only
Main vector format	GeoPackage	File geodatabase
Project file	`.qgz`	`.aprx`
Analysis tools	Processing Toolbox (QGIS, GDAL, GRASS)	Geoprocessing toolboxes and extensions
Workflow models	Model Designer	ModelBuilder
Scripting	Python (PyQGIS)	Python (ArcPy), Arcade
Map output	Print layout	Layout
Web sharing	Plugins and other software, such as qgis2web and QGIS Server	Built in, through ArcGIS Online or Enterprise
Extending	Thousands of free community plugins	Esri extensions, add-ins, and partner products

The two programs share far more than they differ. Both read the same common formats (shapefiles, GeoPackages, GeoTIFFs, CSVs), both use GDAL and PROJ-style coordinate systems identified by EPSG codes, and the concepts in this book apply equally to both. Many GIS professionals use both: ArcGIS Pro where their organization runs on Esri software, and QGIS on their own computers, for formats or tools that ArcGIS lacks, or when sharing work with people without a license.

18.13 Getting help

- **Help buttons**: the **?** on any tool or pane, or F1, opens the matching page of the ArcGIS Pro documentation.
- **The ArcGIS Pro documentation** at https://pro.arcgis.com includes a "Get started" section, a tool reference for every geoprocessing tool, and the ArcPy reference.
- **Learn ArcGIS** (https://learn.arcgis.com/en/) has free, step-by-step lessons with data, from beginner to advanced.
- **Esri Training** (https://www.esri.com/training/) has courses, many of them free, including *Getting Started with ArcGIS Pro*.
- **Esri Community** (https://community.esri.com) and **GIS Stack Exchange** (tag `arcgis-pro`) answer questions.

18.14 Check your understanding

The interactive self-check questions for this chapter are in the online version of the book.

18.15 Summary

- **ArcGIS Pro** is Esri's proprietary desktop GIS for Windows, part of the ArcGIS family that includes ArcGIS Online, ArcGIS Enterprise, field and web apps, and ArcPy.
- **Esri** was founded in 1969 by Jack and Laura Dangermond. Its desktop GIS went from **ARC/INFO** (1982) to **ArcView** (1992), **ArcMap** (1999), and **ArcGIS Pro** (2015). ArcMap was retired in March 2026.
- ArcGIS Pro needs a license, usually a **named user license** assigned to your ArcGIS account, at the **Basic**, **Standard**, or **Advanced** level, with **extensions** for specialized tools. Some services use **credits**.
- The window has a **ribbon** of tabs and groups, with **contextual tabs** that appear as needed; **views** for maps, layouts, and tables; and **panes**, such as Contents, Catalog, Geoprocessing, and Symbology. **Command Search** finds any command.
- A **project** (`.aprx`) links to data and holds maps and layouts. New data goes in the project's default **file geodatabase**.
- Add data with **Add Data** or from the Catalog pane, and points from coordinates with **XY Table To Point**. Create **feature classes** in the Catalog pane, digitize with **Create Features**, and save edits from the **Edit** tab.
- Make a map with the **Symbology** pane, the **Labeling** tab, and a **layout** with a map frame, title, legend, scale bar, and north arrow, then export it.
- Run tools in the **Geoprocessing pane**, chain them in **ModelBuilder**, and script them with **ArcPy**.

18.16 Exercises

1. **Sign in and explore.** Start ArcGIS Pro, sign in, and create a project from the Map template. What version are you using? What license level and extensions do you have (**Project ▶ Licensing**)? Find the Contents pane, the Catalog pane, and Command Search.
2. **The project folder.** Open the project folder in Windows File Explorer. List what ArcGIS Pro created. Which items hold data, and which hold settings?
3. **Points from a spreadsheet.** Make a CSV of five places with their latitudes and longitudes, and use XY Table To Point to map them. Add a basemap and check that they land in the right places.
4. **Digitize.** Create a polygon feature class in the project geodatabase, with a Text field `name`, in a State Plane coordinate system. Digitize three buildings on campus over the Imagery basemap. What are their areas in the `Shape_Area` field, and in what units?
5. **A choropleth map.** Map population density for North Carolina's counties with Graduated Colors and five natural-breaks classes. Build a layout with a title, legend, scale bar, north arrow, and source, and export it as a PDF. Then make the same map in QGIS. Which steps were the same, and which were different?
6. **Geoprocessing history.** Buffer a layer with the Pairwise Buffer tool, then open the History. Copy the tool's Python command (right-click ▶ Copy Python Command), and paste it into the Python window. What does each part of the command mean?



18.17 Further reading and online resources

18.17.1 ArcGIS Pro documentation

- ArcGIS Pro documentation home: https://pro.arcgis.com/en/pro-app/latest/get-started/get-started.htm.
- "What's new in ArcGIS Pro," for the current version: https://pro.arcgis.com/en/pro-app/latest/get-started/whats-new-in-arcgis-pro.htm.
- "ArcGIS Pro system requirements": https://pro.arcgis.com/en/pro-app/latest/get-started/arcgis-pro-system-requirements.htm.
- "Layouts in ArcGIS Pro": https://pro.arcgis.com/en/pro-app/latest/help/layouts/layouts-in-arcgis-pro.htm.
- "What is geoprocessing?": https://pro.arcgis.com/en/pro-app/latest/help/analysis/geoprocessing/basics/what-is-geoprocessing-.htm.
- "What is ArcPy?": https://pro.arcgis.com/en/pro-app/latest/arcpy/get-started/what-is-arcpy-.htm.
- "License levels" for ArcGIS Pro: https://pro.arcgis.com/en/pro-app/latest/get-started/license-levels.htm.
- ArcGIS Arcade documentation: https://developers.arcgis.com/arcade/.

18.17.2 Lessons and courses

- Learn ArcGIS, free lessons: https://learn.arcgis.com/en/.
- Esri Training catalog: https://www.esri.com/training/.
- *Getting to Know ArcGIS Pro*, by Michael Law and Amy Collins (Esri Press), a step-by-step workbook updated for each major version.
- *Understanding GIS: An ArcGIS Pro Project Workbook*, by Christian Harder and Clint Brown (Esri Press).

18.17.3 History and community

- Esri on Wikipedia: https://en.wikipedia.org/wiki/Esri.
- ArcGIS on Wikipedia, with a table of versions: https://en.wikipedia.org/wiki/ArcGIS.
- Esri Community: https://community.esri.com.
- GIS Stack Exchange, questions tagged `arcgis-pro`: https://gis.stackexchange.com/questions/tagged/arcgis-pro.
- Esri User Conference: https://www.esri.com/en-us/about/events/uc/overview.

Glossary

Key terms used in this book. In the chapters, a term with a dotted underline shows its definition when you hover over it, tap it, or reach it with the Tab key.

A missing value: nothing was recorded. NULL fails every comparison, so test for it with IS NULL.

.csvt file A one-line file next to a CSV that sets the data type of each column for QGIS and GDAL, such as "String","Integer".

absolute location A location given in a coordinate system that means the same thing to everyone, such as 38° 53′ 52″ N, 77° 02′ 11″ W.

accessibility How easily people can reach a place or service, often measured as distance or travel time.

ACS 1-year estimates ACS estimates from 12 months of data, published only for areas with 65,000 or more people.

ACS 5-year estimates ACS estimates from 60 months of data, published for every area down to block groups.

adaptive sampling Sampling in passes: a sparse first pass, then more samples where values change quickly.

address matching Geocoding an address by comparing it with a reference file whose features already carry address information.

address point A point for one address, often placed on the rooftop or the building entrance.

address range The lowest and highest house numbers on one side of a street segment.

affine transformation A six-coefficient transformation that corrects translation, scale, skew, and rotation. Needs at least 3 GCPs.

aggregation Rescaling a raster to coarser cells by combining blocks of cells into one, using a statistic such as the mean, sum, or majority.

Albers Equal-Area Conic An equal-area conic projection, the standard for measuring areas across the contiguous United States (EPSG:5070).

American Community Survey (ACS) The Census Bureau's yearly sample survey of about 3.5 million addresses, the main source of social, economic, and housing data.

anisotropy Spatial dependence that changes with direction. Its opposite is isotropy, the same dependence in every direction.

annotation Text placed on a map, such as place names, street names, and labels.

apportionment Dividing the 435 seats in the U.S. House of Representatives among the states, by population.

arc A line stored as an ordered list of coordinates. Polygons can be built from arcs that share their end points.

ARC/INFO Esri's command-line GIS, first released in 1982.

Arcade Esri's expression language, used for calculating fields, labels, and pop-ups in ArcGIS Pro and ArcGIS Online.

ArcGIS Online Esri's system for making and sharing maps and layers on the web.

ArcGIS Pro Esri's commercial desktop GIS, for Windows.

ArcMap Esri's main desktop GIS application from 1999 until ArcGIS Pro replaced it. Esri ended support for it on March 1, 2026.

ArcPy Esri's Python library for running ArcGIS tools and automating work in ArcGIS Pro.

arithmetic progression A classification method in which each class is wider than the one before by a constant amount.

asserted data Data contributed by the public, such as OpenStreetMap. Their accuracy is undocumented and uneven.

attribute A non-spatial fact about a feature, such as a building's name or the year it was built. Attributes are stored in a table with one row per feature.

attribute accuracy How correct the non-spatial values in a dataset are, such as a feature's type or name.

authoritative data Data from official agencies, collected under documented standards and published with metadata.

average neighbor An interpolation method that estimates each location as the mean of the observations within a search radius.

azimuthal A projection onto a plane that keeps true directions from its center point.

bag A collection that can contain duplicates. A table is a bag unless a primary key forbids repeated rows.

bandwidth The radius of a kernel. Small bandwidths show local hot spots; large ones show regional patterns.

basemap A background map, such as streets, topography, or aerial imagery, that gives context to the layers drawn on top of it.

batch geocoding Geocoding a whole table of addresses in one run.

Big Data Datasets too large, too fast, or too varied for traditional tools, often described by three Vs: volume, velocity, and variety.

bilinear interpolation Resampling that averages the four nearest input cells, weighted by distance.

binary Base-2 numbers, written with only 0 and 1, in which each column is worth twice the one to its right. Computers store all data in binary.

bit A binary digit, 0 or 1. With n bits you can write 2^n different values.

block group A cluster of census blocks within a tract, usually with 600 to 3,000 people. The smallest area with ACS estimates.

Browser panel The QGIS panel that shows folders, databases, and web services, from which data can be dragged onto the map.

buffer The region within a specified distance of a point, line, or polygon.

bull's-eye A ring of values around a single observation that differs from its neighbors; a sign of IDW interpolation.

CAD Computer-aided design: software for drawing buildings, parts, and other objects. It is spatial but not geographic, because its drawings aren't tied to places on Earth.

Canada Geographic Information System (CGIS) The first large GIS, built in the 1960s under Roger Tomlinson to inventory land across Canada.

cardinality How many records in one table match each record in another: one-to-one, many-to-one, one-to-many, or many-to-many.

cartographic boundary files Simplified census boundaries clipped to the shoreline, for small-scale maps.

cartographic modeling Planning and running a chain of spatial operations to solve a problem, where each operation's output becomes the next one's input. The term is Dana Tomlin's.

cartography The art and science of making maps.

Catalog pane The ArcGIS Pro pane that shows a project's maps, layouts, toolboxes, databases, and folder connections.

cell One square in a raster grid. Each cell holds one value. In an image, a cell is called a pixel (picture element).

cell size The real-world length of one side of a raster cell, such as 30 meters. Smaller cells give finer resolution and larger files.

census block The smallest census area, bounded by streets, streams, and other visible

features. Decennial counts only.

census geography The hierarchy of areas for which census data are published, from the nation down to the block.

census tract A relatively permanent subdivision of a county with 1,200 to 8,000 people, about 4,000 on average.

census-designated place An unincorporated community that the Census Bureau defines so that it can publish statistics for it.

central meridian The meridian at the center of a projection or zone, such as the line along which a Transverse Mercator cylinder touches the Earth.

centroid The center point of a polygon.

change detection Comparing images or maps of the same place from different dates to find what changed.

changeset A group of edits to OpenStreetMap saved together, with the editor's username and a comment.

check point A point with known coordinates that is not used to fit a transformation, kept back to test its accuracy.

choropleth map A map that shades each area, such as a county, by its value, usually in a few classes from light to dark.

class break A value that divides one class from the next on a classed map.

classed map A choropleth map in which values are grouped into a few classes, each shown with one shade.

client/server A design in which client programs send requests to a server, which does the work and sends back the answer.

clip An overlay that keeps only the parts of a data layer inside a bounding layer, keeping only the data layer's attributes.

closed way In OpenStreetMap, a way (an ordered list of nodes) whose first and last nodes are the same, forming an area such as a building.

cluster A group of events or features that are closer together than you would expect by chance, such as cases of a disease around one place.

cluster tolerance A distance within which vertices are treated as the same point during overlay, to prevent slivers.

coefficient of variation (CV) The standard error as a percentage of the estimate. Over about 40 percent, an ACS estimate is usually treated as unreliable.

cokriging Kriging that also uses one or more correlated secondary variables, such as elevation when predicting temperature.

ColorBrewer A free online tool with tested color schemes for maps, for 3 to 9 classes.

complement Every record outside a set. NOT selects the complement.

completeness An element of data quality: what is missing from a dataset, and what is there that shouldn't be.

compromise projection A projection that preserves no property exactly but keeps every distortion moderate, such as Robinson or Winkel Tripel.

concurrency Many users reading and editing a database at the same time, without overwriting each other's changes.

conditional function The raster version of if-then-else (CON in ArcGIS Pro): where a condition is true, take one value or raster; where false, another.

confidence level How often a range built the same way would contain the true value, if the survey were repeated many times.

conformal A projection that keeps local angles true, so small shapes look right. Mercator and Lambert Conformal Conic are conformal.

Contents pane The ArcGIS Pro pane that lists the layers in the active map or the elements of the active layout.

contextual tab A ribbon tab in ArcGIS Pro that appears only when a certain kind of item, such as a feature layer, is selected.

contiguity Touching. Contiguous polygons share a border (rook's case) or a border or corner (queen's case).

continuous data A quantity that has a value at every location and changes gradually from place to place, such as elevation, temperature, or rainfall.

Continuously Operating Reference Stations (CORS) A network of permanent GPS receivers, run by the National Geodetic Survey and its partners, that keeps NAD83 up to date.

control point A point on the ground, often marked with a bronze disk, whose coordinates have been measured precisely. Control points tie a datum to the Earth.

coordinate reference system (CRS) The full description of how coordinates relate to the Earth: a datum, plus a projection with its parameters, plus units.

coordinate system A system of numbers, such as latitude and longitude, that gives every location a unique position so layers line up with each other.

cost distance Distance measured as an accumulated cost, such as travel time or effort, across a cost raster, rather than in meters.

credits The currency an ArcGIS organization uses to pay for some services, such as batch geocoding and storing data in ArcGIS Online.

crisis mapping Mapping by volunteers to support the response to a disaster or other crisis.

cross validation Testing an interpolation by removing each observation in turn, estimating it from the rest, and recording the error.

CSV A plain-text table in which values are separated by commas, one record per line.

cubic convolution Resampling that uses the 16 nearest input cells.

dasymetric map A map that redistributes values within each area onto the land where they occur, such as placing people only where there are homes.

data abstraction The chain of steps that turns the real world into something a computer can store: real world, data model, data structure, and machine code.

data model The way a GIS represents the real world as data, for example as points, lines, and polygons (vector) or as a grid of cells (raster).

Data Source Manager The QGIS dialog for adding layers of every kind: files, delimited text, databases, and web services.

data structure How a data model is laid out in a computer's memory or files, for example as lists of coordinate pairs linked by pointers.

data type How a field is stored, such as text, integer, double, or date. It limits what the field can hold.

database An organized collection of tables, usually stored and managed by a database management system.

database management system (DBMS) Software that stores data once and lets many programs and people query and edit it, while it controls who may change what.

datum shift The difference in where the same latitude and longitude land on the ground in two different datums. Between NAD27 and NAD83 it can exceed 100 m.

datum transformation Equations or a grid of measured shifts that convert coordinates from one datum to another.

De Morgan's law NOT (A AND B) equals (NOT A) OR (NOT B), and NOT (A OR B) equals (NOT A) AND (NOT B).

decennial census The full count of every U.S. resident, taken every ten years as the Constitution requires.

decimal degrees (DD) Latitude or longitude written as one signed number, such as −79.0558. Negative numbers are south latitudes and west longitudes.

definition query In ArcGIS Pro, a WHERE clause that hides the features that fail it, until you remove it. QGIS calls this a filter.

degrees, minutes, and seconds (DMS) Latitude or longitude written in degrees, minutes (1/60 of a degree), and seconds (1/3,600 of a degree), with a hemisphere letter, such as 79° 03′ 21″ W.

demographic data Data on who lives in a place: counts of people and households, and their age, sex, race and ethnicity, and household type.

density A count divided by an area, such as people or crimes per square mile.

deprivation index A single score that combines several socioeconomic variables to measure how disadvantaged an area is.

destination table In a join, the table that receives the new fields. All of its records are kept.

deterministic interpolation Interpolation in which the surface is fully determined by a formula and the chosen parameters, with no statistical model of error.

developable surface A shape that can be unrolled flat without stretching: a plane, a cylinder, or a cone.

differential privacy A method that adds calibrated statistical noise to published tables, so that individuals cannot be identified.

digital elevation model A raster whose cell values are elevations of the ground surface.

digitize To create vector features by drawing them, clicking each point or vertex on a map or image.

discrete object Something with clear boundaries that is either present or absent at a location, such as a road, a house, or a state.

dissolve An operation that merges features sharing a value in a field, removing the boundaries between them.

distance join A spatial join that matches each feature to the closest feature, or to every feature within a set distance, and can record the distance.

diverging color scheme Two hues that meet at a light neutral color, for data with a meaningful midpoint such as a mean or zero.

Double A 64-bit data type for decimal numbers, with about 15 significant digits. Float is the 32-bit version, with about 7.

easting A distance east from a starting line: the x coordinate in a projected coordinate system.

ecological fallacy Assuming that every person or place in an area shares the area's average value.

Economic Census The Census Bureau's survey of businesses, taken every five years in years ending in 2 and 7.

edge effect Locations near the boundary of a study area find fewer neighbors than they should, because their neighbors outside it were not observed.

ellipsoid A smooth mathematical model of the Earth's shape: an ellipse rotated about its polar axis, wider at the equator than from pole to pole. Examples are Clarke 1866, GRS 80, and WGS 84.

ellipsoid height Height above the ellipsoid, written h. This is the height a GPS receiver measures.

entity A real-world object we want to store as data, such as a tree, a building, or a lake. On a map, an entity is shown as a feature.

enumeration unit An area for which data are collected and reported, such as a county, census tract, or state.

epidemiology The study of how diseases are distributed and spread in populations. John Snow's 1854 cholera map is a founding example.

EPSG code A number that identifies a coordinate reference system in the EPSG registry, such as 4326 for WGS84 or 32119 for North Carolina State Plane.

equal interval A classification method that divides the range of the data into classes of equal width.

equal-area A projection that keeps relative area true. Use one to measure or compare areas, or to map density.

equator The parallel halfway between the poles, at 0° latitude. It is the only parallel that is a great circle.

equidistant A projection that keeps true distance from one or two points, or along certain lines.

Esri The company that makes ArcGIS, founded in 1969 by Jack and Laura Dangermond in Redlands, California.

Euclidean distance Straight-line distance. As a raster function, each cell's distance to the nearest source cell.

exact interpolator An interpolation method whose surface passes through every observed value.

expression A formula of fields, values, operators, and functions that computes a value for each feature, used to calculate fields, select features, and control labels and symbology.

extent How far something reaches: its size, boundary, and dimensions. A layer's extent is the rectangle that just encloses all its features.

extrapolation Estimating values beyond the observations: outside the area they cover, or above or below the observed range.

extrinsic quality assessment Measuring the quality of a dataset by comparing it with an authoritative reference dataset.

FAIR principles Guidelines saying that good data should be Findable, Accessible, Interoperable, and Reusable (Wilkinson et al. 2016).

false easting A value added to every x coordinate so that none are negative, such as 500,000 m in UTM.

feature How a map or a GIS represents an entity, usually as a point, a line, or a polygon, with attributes that describe it.

feature class Esri's term for a vector layer stored in a geodatabase: a set of features with the same geometry type and fields.

field A column in an attribute table: one attribute, measured the same way for every feature. Also called an attribute or item.

field calculator A tool that computes values and writes them into a field, for every record or for the selected records only.

file coordinates Positions in a raster given as pixel columns and rows.

FIPS code A numeric code the U.S. government assigns to each state, county, and other area, such as 37 for North Carolina and 37135 for Orange County.

first-order neighbor A polygon that touches the polygon in question directly. Its own neighbors are second-order neighbors.

flattening How squashed an ellipsoid is: $f = (a - b) / a$. For the Earth, about 1/298.

focal function A map algebra function that computes each cell's output from a moving window of nearby cells, such as a 3×3 mean.

foreign key A field that holds values of another table's primary key. Its values can repeat.

full outer join A join that keeps every record of both tables.

GDAL The Geospatial Data Abstraction Library, open-source software that reads and writes hundreds of raster and vector formats. QGIS, ArcGIS Pro, and many other programs use it.

GeoAI Artificial intelligence applied to geographic data, for example to find buildings or roads in satellite images automatically.

geocoding Turning an address or place name into coordinates, so it can be put on a map.

geodatabase Esri's format for storing feature classes, tables, and rasters together. A file geodatabase is a folder ending in .gdb.

geodesic distance Distance measured along the curved surface of the ellipsoid, rather than on a flat projected map.

geodesy The science of measuring the Earth's size, shape, orientation in space, and gravity field.

geodetic datum A model that fixes an ellipsoid's size, shape, and position relative to the Earth, realized by a network of control points, so latitude and longitude point to real



places. Examples are NAD27, NAD83, and WGS84.

geographic coordinate system (GCS) A coordinate system that gives positions as latitude and longitude, in degrees, on a datum. EPSG:4326 (WGS84) and 4269 (NAD83) are examples.

geographic information science (GIScience) The theory and methods behind GIS: research on analysis, data representation, visualization, and GIS and society. Michael Goodchild named the field in 1992.

geographic information system (GIS) "A computer-based system to aid in the collection, maintenance, storage, analysis, output, and distribution of spatial data and information" (Bolstad). Its five components are data, hardware, software, people, and procedures.

geographic problem A problem that happens in specific places at specific times. It can be described by its scale, time, space, and purpose.

GEOID The Census Bureau's identifier for a geographic area, made by stringing FIPS codes together. A tract GEOID has 11 characters.

geoid The surface of equal gravity that best matches global mean sea level. It is lumpy, up to about 100 m above or below the ellipsoid, and elevations are measured from it.

geoid height The distance from the ellipsoid to the geoid at a place, written *N*. In the lower 48 United States it is negative, about −30 m.

geomasking Moving geocoded points by a small random amount to protect the privacy of the people they represent.

geometric progression A classification method in which each class is wider than the one before by a constant ratio, suited to strongly skewed data.

geometry The shape of a feature: a point, a line, or a polygon, stored as coordinates.

GeoPackage An open file format (.gpkg) from the Open Geospatial Consortium that stores many vector and raster layers in a single file. It is QGIS's default format.

geoprocessing Running a GIS tool on spatial data to make new data, such as buffering, clipping, or overlaying layers.

Geoprocessing pane The ArcGIS Pro pane for finding, filling in, and running geoprocessing tools.

georeferencing Transforming a raster, such as a scanned map or image, from file coordinates to map coordinates.

geospatial A common synonym for *geographic*, used especially in industry, the military, and government.

geostatistics Statistics for spatially continuous data, which model spatial autocorrelation to predict values and their uncertainty. Kriging is its best-known method.

GIS question A question whose answer depends on location, such as where something clusters, how places differ, what factors explain a pattern, how a place has changed, or how accessible a service is.

GIS&T Body of Knowledge A free collection of short, peer-reviewed articles on geographic information science and technology topics, from the University Consortium for Geographic Information Science (UCGIS).

GISP Geographic Information Systems Professional: a certification from the GIS Certification Institute for experienced GIS practitioners.

gnomonic projection An azimuthal projection on which every great circle is a straight line.

goodness of variance fit A measure from 0 to 1 of how well classes fit the data: 1 minus the squared deviations from the class means divided by those from the overall mean.

GPS The Global Positioning System: U.S. satellites that let a receiver, such as your phone, work out its location. GNSS is the general name for GPS and similar systems from other countries.

graticule The network of parallels and meridians drawn on a map.

great circle A circle on a sphere whose plane passes through the center. The shortest route between two points follows a great circle.

ground control point (GCP) A feature that can be found on both an image and a reference layer, used to work out the transformation between them.

heat map A map of a density surface, usually from kernel density estimation.

histogram A bar chart of how many values fall in each range, used to see the shape of a distribution.

Humanitarian OpenStreetMap Team An organization that coordinates volunteer OpenStreetMap mapping for disaster response and development.

iD OpenStreetMap's web-based editor, which runs in a browser with no installation.

image rectification Correcting an image's geometry so that it fits a map coordinate system.

inexact interpolator An interpolation method that smooths the data, so its surface may not pass through the observed values.

inner join A join that keeps only the records that match in both tables.

inside join A spatial join that matches features that contain, lie within, or intersect each other, such as points inside polygons.

integer A data type for whole numbers. Its range depends on its bits: up to 32,767 for 16 bits, about 2.1 billion for 32 bits.

intersect An overlay that keeps only the areas covered by both input layers, with attributes from both. The overlay version of AND.

intersection The records in both of two sets. AND selects the intersection.

interval Values with equal steps but an arbitrary zero, such as temperature in °C or a calendar year. Differences make sense; ratios don't.

intrinsic quality assessment Measuring the quality of OpenStreetMap from its own editing history, such as the number of editors and versions in an area.

inverse distance weighted An interpolation method that estimates each location as a weighted average of nearby observations, with weights of 1 / distance raised to a power.

ISNULL A raster function that outputs 1 where a cell is NoData and 0 elsewhere. It never outputs NoData itself.

ISO 19157 The international standard for describing and measuring the quality of geographic data.

Join_Count The field ArcGIS Pro's Spatial Join adds to hold the number of join features that matched each target feature.

JOSM The Java OpenStreetMap editor, a desktop editor for advanced work.

k nearest neighbors A neighborhood made of the k closest locations, whatever their distance.

kernel In density estimation, a moving window that weights nearby points by their distance from the cell at its center.

kernel density estimation A method that turns points into a continuous density surface by sliding a weighted window (a kernel) across a grid.

key A field whose values link the records of one table to the records of another.

kriging A geostatistical interpolation method that weights observations using a model of spatial autocorrelation (the semivariogram), and estimates the uncertainty of each prediction.

Lambert Conformal Conic A conformal conic projection with two standard parallels. It suits mid-latitude regions that run east to west, and is used for North Carolina State Plane.

Landsat A series of U.S. Earth-observing satellites, run by NASA and the USGS, that have imaged the whole planet since 1972, mostly with 30-meter cells.

latitude The angle north or south of the equator, from 0° at the equator to 90° at the poles.

layer A set of spatial data about one theme, such as roads, soils, or elevation. Layers for the same area stack on top of each other to make a map.

Layers panel The QGIS panel that lists the project's layers in drawing order.

least squares Fitting a model by choosing the coefficients that make the sum of the



squared errors as small as possible.

left join A join that keeps every record of the left (destination) table, with NULL where nothing matched. The default in QGIS and ArcGIS Pro.

legend The part of a map that explains its symbols.

level of measurement What kind of values an attribute holds, which decides what questions make sense: nominal, ordinal, interval, or ratio.

license level One of the three tiers of an ArcGIS Pro license, Basic, Standard, and Advanced, each adding tools to the one below.

line A vector feature with length but no area, stored as an ordered list of coordinates, such as a road or a stream.

line segment The straight piece of a line between two vertices or nodes.

lineage The history of a dataset: where the data came from, and how they have been collected, processed, and edited since.

linear interpolation Estimating a position between two known points in proportion to a value between their values, such as a house number within a range.

linking table A table with one row for each pair of related records, used to handle a many-to-many relationship.

Linus's Law "Given enough eyeballs, all bugs are shallow." Applied to maps, areas edited by more contributors tend to be more accurate.

logical consistency Whether data agree with their own rules, for example polygons that close and roads that meet at intersections.

long-term release A QGIS version, released once a year, that gets bug fixes but no new features for about a year. It is the most stable choice.

longitude The angle east or west of the prime meridian, from 0° at Greenwich to 180° east or west.

majority filter A neighborhood function that gives each cell the value shared by most of its neighbors, used to clean up classified images.

manual intervals Class breaks set by the analyst at meaningful values, such as a poverty line, or the same breaks for a series of maps.

many-to-many Records on each side match many records on the other. Needs a linking table.

many-to-one Many destination records match the same source record, but each matches only one. A simple join works.

map algebra Math on rasters, cell by cell, such as adding two grids or finding every cell where slope is under 10 percent.

map generalization Deciding what detail to keep on a map, by simplifying, classifying, and enhancing.

map projection A mathematical method for flattening the round Earth onto a flat map. Every projection distorts area, shape, distance, or direction in some way.

mapathon An event where volunteers map together for a few hours, often for a humanitarian project.

MapRoulette A tool that turns detected OpenStreetMap errors into small tasks that volunteers can fix.

margin of error (MOE) The likely range of sampling error around an estimate. Census Bureau margins of error use a 90 percent confidence level.

match rate The share of addresses in a table that were successfully geocoded.

match score How closely an address agrees with a candidate in the reference file, often on a scale from 0 to 100.

mean error The average of the cross-validation errors. Near 0 means the surface is unbiased.

mean sea level The average height of the sea surface over many years, which the geoid approximates.

Mercator A conformal cylindrical projection from 1569, made for navigation: lines of



constant compass bearing are straight. It greatly enlarges areas toward the poles.

merge rule In ArcGIS Pro, how the values of several matching features are combined into one, such as first, sum, or mean.

meridian A line of equal longitude. Meridians run from pole to pole and meet at the poles.

metadata Data about data: who made a dataset, when, how, in what coordinate system, and how accurate it is.

Missing Maps A project founded in 2014 that maps vulnerable places before disasters happen.

ModelBuilder ArcGIS Pro's visual tool for drawing and running a cartographic model as a flowchart. QGIS's equivalent is the Model Designer.

moving window A window of cells, usually 3 × 3, that slides across a raster one cell at a time while a neighborhood function runs.

multiband A raster with several values per cell, one per band, such as the red, green, blue, and near-infrared bands of a satellite image.

NAD27 The North American Datum of 1927. It used the Clarke 1866 ellipsoid and an origin at Meades Ranch, Kansas.

NAD83 The North American Datum of 1983, an earth-centered datum on the GRS 80 ellipsoid, fixed to the North American plate. The datum of most U.S. government GIS data.

named user license An ArcGIS Pro license assigned to a person's ArcGIS account, and checked when they sign in.

NATRF2022 The North American Terrestrial Reference Frame of 2022, the new datum that will replace NAD83 in the United States.

natural breaks A classification method (Jenks) that groups similar values by minimizing the variance within classes. The default in ArcGIS Pro.

nearest neighbor The feature closest to a given feature.

neighborhood In spatial analysis, a zone drawn around a location, defined by distance, by contiguity, or by a number of nearest neighbors.

neighborhood effect The tendency of a place's value to depend on the values around it.

neighborhood function A raster function that computes a statistic from a window of cells around each cell and writes it to the center cell.

network Connected lines and intersections, such as roads, rails, rivers, or pipes, with a cost such as length or travel time on each segment.

network analysis Analysis of movement along a network, such as roads, to find routes and measure travel distance or time.

NoData A code meaning a raster cell has no value. Any map algebra operation that touches NoData outputs NoData, and statistics skip it.

node The start or end point of a line. In a network, lines meet at nodes. In OpenStreetMap, a node is any single point with a latitude and longitude, standing alone or as a vertex of a way.

nominal Categories with no order, such as land use or a name. You can test equality and count, but not average.

non-spatial data Data with no location of their own, such as a list of grades. When non-spatial data describe a spatial feature, they are its attributes.

normalize To divide a count by an area, a population, or another total, so that areas of different sizes can be compared fairly.

north arrow A symbol showing the direction of north on a map.

northing A distance north from a starting line: the y coordinate in a projected coordinate system.

nugget The semivariance at (nearly) zero distance, from measurement error or variation smaller than the sample spacing.

on-the-fly reprojection Reprojecting layers as they are drawn, so layers in different CRSs line up on the map without changing their stored coordinates.

one-to-many One destination record matches many source records. Summarize the source first, or use a relate.

one-to-one Each destination record matches exactly one source record, and the other way around.

Open Database License The license for OpenStreetMap data. Anyone may use and share the data if they credit OpenStreetMap contributors and share derived databases under the same license.

open source Software whose source code anyone may read, change, and share, often free of charge. QGIS and GRASS GIS are open source.

OpenStreetMap A free, editable map of the world built by volunteers, founded in 2004.

ordinal Ranked categories whose steps are of unknown size, such as A, B, C, D or a hurricane's category.

ordinary kriging Kriging that assumes an unknown but constant mean. The most common type.

orthometric height Height above the geoid, written H: the elevation, or "height above sea level," printed on maps. $h = H + N$.

OSGeo The Open Source Geospatial Foundation, a nonprofit that supports open-source geospatial software such as QGIS, GDAL, PROJ, and PostGIS.

OSMCha OSM Changeset Analyzer, a tool for reviewing recent OpenStreetMap edits to catch mistakes and vandalism.

Osmose A tool that runs automated quality checks on OpenStreetMap data.

overlay analysis Stacking layers of data for the same area to find places that meet several conditions at once, such as flat land near a road and outside a floodplain.

Overture Maps Foundation A group founded in 2022 by Amazon, Meta, Microsoft, and TomTom to build open map data, partly from OpenStreetMap.

parallel A line of equal latitude. Parallels run east and west and get shorter toward the poles.

parcel A polygon for one piece of property, usually with its owner and address.

partial sill The sill minus the nugget: the part of the variation that is spatially autocorrelated.

Plate Carrée The projection that plots longitude as x and latitude as y on an even grid. It stretches the map east to west toward the poles.

plugin An add-on that extends QGIS with new tools, installed from inside QGIS with Plugins ▶ Manage and Install Plugins.

point A vector feature with a location but no length or area, stored as a single x, y coordinate pair, such as a well or a school.

pointer In a data structure, a stored address that tells the computer where the next piece of data, such as the next arc, is found.

polygon A vector feature that encloses an area: a closed line whose first and last points are the same, such as a lake, a park, or a county.

polynomial transformation A transformation that can bend an image. Second order needs at least 6 GCPs; third order needs 10.

population density The number of people per unit of area, such as people per square mile.

positional accuracy How close the coordinates in a dataset are to the true positions of the features.

Post-Enumeration Survey A survey taken after the census to measure how many people it missed or counted twice.

PostGIS A free extension that adds geometry types and spatial functions to the PostgreSQL database.

precision For a numeric field, the total number of digits it can hold. Scale is how many of them fall after the decimal point.

prediction standard error Kriging's estimate of how far off each prediction might be. It is largest where observations are sparse.

primary key A field whose values identify each row of a table uniquely. No value may repeat or be missing.

prime meridian The meridian at 0° longitude, through Greenwich, England, chosen in 1884. Longitude is measured east and west from it.

print layout In QGIS, a page on which a map is arranged with its title, legend, scale bar, and other elements, for printing or export. ArcGIS Pro calls it a layout.

Processing Toolbox The QGIS panel that holds its analysis tools (algorithms), from QGIS itself, GDAL, and GRASS GIS.

projected coordinate system A coordinate system that gives positions as *x, y* in meters or feet, on a map projection built on a datum, such as UTM or State Plane.

proportional symbol map A map that shows counts with symbols, such as circles, sized in proportion to each value.

QGIS A free and open-source desktop GIS that runs on Windows, Mac, and Linux.

qualitative color scheme Distinct hues of similar lightness, for categories rather than quantities.

quantile A classification method that puts the same number of areas in each class. Five quantile classes are quintiles.

quartic kernel A hump-shaped kernel that weights points most at the center and falls to zero at the bandwidth. The default in ArcGIS Pro and QGIS.

queen's case Contiguity in which polygons are neighbors if they share an edge or a corner.

query A written condition that the software tests against every record, selecting those that pass.

radar A remote sensing method in which a satellite or aircraft sends out its own microwave signal and records what bounces back. It sees through clouds and smoke, and works at night.

radial basis function An exact interpolation method that fits a smooth, minimum-curvature surface through every observation, like a flexible sheet pinned at each point.

range In a semivariogram, the distance at which the sill is reached. Beyond it, observations are not autocorrelated.

raster A data model that stores the world as a regular grid of equal-sized cells, each holding one value. Best for continuous data and imagery.

rasterize To convert vector data into a raster.

ratio Values with equal steps and a true zero, such as population or area. Every arithmetic operation makes sense.

raw count A total that hasn't been normalized, such as a number of people or crimes. Raw counts grow with the size of an area.

ray casting A point-in-polygon test that counts how often a line from the point crosses the polygon's edge. An odd count means inside.

reclassification Grouping the values of a layer into fewer, more meaningful classes, such as elevations into low, medium, and high.

record A row in an attribute table: everything known about one feature.

redistricting Redrawing the boundaries of congressional and legislative districts after each census.

redlining Grading neighborhoods as risky for lending, as the federal Home Owners' Loan Corporation did on maps in the 1930s, which outlined many Black and immigrant neighborhoods in red and denied them credit.

reference file In geocoding, the layer that addresses are matched against, such as parcels, street centerlines, or ZIP Code areas.

relate A link between two tables that lets you see the matching records without copying fields into the layer. QGIS calls it a relation.

relational database A database that stores data in tables and links them through key fields that hold the same values.

relative location A description of where something is in relation to a landmark, such as "ten paces north of the oak." It only works if you know the landmark.

remote sensing Collecting information about Earth without touching it, from sensors on satellites, aircraft, or drones.

reproject To convert a layer's coordinates from one CRS to another and save the result. Defining a projection, by contrast, only labels what the coordinates already are.

resampling Giving each cell of a transformed raster a value from the input cells: nearest neighbor, bilinear, or cubic convolution.

residual The difference between an observed value and the value a model predicts; in georeferencing, the distance between a GCP's true and transformed positions.

resolution The level of detail in data. For a raster, it is set by the cell size: the smaller the cells, the finer the resolution.

rhumb line A line of constant compass bearing. It is straight on a Mercator map.

ribbon The band of tabs and grouped commands across the top of the ArcGIS Pro window.

right join A join that keeps every record of the right (source) table.

rook's case Contiguity in which polygons are neighbors only if they share an edge.

root mean square error (RMSE) The square root of the average squared residual: one number that sums up how well a model fits.

route optimization Finding the best path through a network, such as the fastest or shortest route, or a delivery route that visits many stops.

sampling design The pattern of observation locations: systematic, random, clustered, or adaptive.

scale bar A line marked with distances that shows the map's scale graphically. It stays correct when the map is enlarged or shrunk.

scale factor The ratio of scale at a place on a map to the map's stated scale: exactly 1 on a standard line, below 1 where the map shrinks the Earth, above 1 where it stretches it.

search neighborhood The area around an unknown location, set by a radius, sectors, and a shape, from which observations are taken to estimate it.

secant A projection surface that cuts through the Earth along two lines (or a circle), spreading the error more evenly.

selection The records, and their features, that pass a query. A selection is temporary until you export it or calculate it into a field.

semi-major axis The equatorial radius of an ellipsoid, written a. For the Earth, about 6,378 km.

semi-minor axis The polar radius of an ellipsoid, written b. For the Earth, about 6,357 km.

semivariance Half the squared difference between two observed values. Averaged over pairs in a distance bin, it measures dissimilarity at that distance.

semivariogram A plot of semivariance against the distance between pairs of observations, described by its nugget, sill, and range.

sequential color scheme Light-to-dark steps of one or a few related hues, for data that run from low to high.

shapefile A common vector format from Esri that stores one layer in several files with the same name: .shp (geometry), .shx (index), and .dbf (attributes) are required, and .prj holds the coordinate system.

sill The semivariance at which a semivariogram levels off.

simple join A join that copies the attributes of one source feature to each destination feature. Works for one-to-one and many-to-one relationships.

simple kriging Kriging that assumes a known, constant mean.

site selection Using GIS to choose the best location for something, such as a new store, clinic, or fire station.

skewed Describes a distribution with a long tail on one side, such as many small values and a few very large ones.

sliver A thin, tiny polygon created by overlay where two versions of the same boundary

don’t match exactly.

snap raster A raster whose grid sets where output cell edges fall, so that the outputs of raster tools line up exactly with it.

snapping An editing setting that makes a new vertex jump to a nearby vertex or edge, so that shared boundaries line up exactly.

socioeconomic data Data on economic and social conditions, such as income, education, employment, poverty, and housing costs.

source table In a join, the table whose fields are appended to the destination.

spatial analysis Methods that use the locations of things, and the relationships among them, to find patterns, explain them, or make predictions.

spatial autocorrelation The degree to which values at nearby locations are more alike (positive) or more different (negative) than chance would predict.

spatial data Data tied to a location on or near Earth, such as the address of a house, the outline of a county, or the elevation at a point.

spatial data quality How well a geographic dataset fits its intended use, described by elements such as lineage, accuracy, consistency, and completeness.

spatial interpolation Estimating the value of an attribute at unsampled locations from observations at sampled ones.

spatial join A join that matches the features of two layers by location instead of by a shared key, then appends attributes from one to the other.

spatial operation A GIS function that takes one or more input layers and produces an output layer, such as a buffer, dissolve, or overlay.

spatial predicate A rule that tests a spatial relationship between two geometries, such as intersects, contains, within, or touches.

spatial scope The input locations used to compute the value at an output location: local (the same location), neighborhood (nearby locations), or global (every location).

spatial thinking Using ideas of space, such as location, distance, pattern, and scale, to understand and solve problems.

spatial weights matrix A table recording which locations are neighbors of which, and how much each neighbor counts.

spine The main line of census geography (nation, regions, divisions, states, counties, tracts, block groups, blocks), in which each level nests in the one above.

SQL Structured Query Language, the standard language for asking databases questions. In a GIS you usually write only its WHERE clause.

SQLite A small database that lives in a single file. GeoPackages are SQLite databases.

standard deviation classification A classification method that sets class breaks at fixed numbers of standard deviations above and below the mean.

standard line A line where the projection surface meets the Earth, so the scale is exact. Standard lines that follow latitude are standard parallels.

State Plane Coordinate System (SPCS) A U.S. system of about 120 zones following state and county lines, built for surveyors, with error held to 1 part in 10,000. North Carolina has one zone (EPSG:32119 in meters, 2264 in feet).

street centerline A line along the middle of a street. Centerline files used for geocoding store the address range on each side of each segment.

Sturges’ rule A starting point for the number of classes: about 1 + 3.3 log10(n), where n is the number of values.

summarized join A join that counts and summarizes all the source features matching each destination feature (count, sum, mean, minimum, maximum).

symbology The rules that control how a layer is drawn: its symbols, colors, sizes, and classes.

table join Appending the fields of a source table to a destination table by matching the values of a key field.

tabular data Data stored in tables of rows and columns, such as the attributes of features.

tag In OpenStreetMap, a key=value pair that describes an element, such as highway=residential.

tangent A projection surface that touches the Earth along one line or at one point.

Tasking Manager HOT's tool that divides a mapping project into small squares, so that volunteers don't overlap and each square can be validated.

temporal quality How current a dataset is, and how accurate its dates are.

temporary layer A layer kept only in memory, such as a tool output not saved to a file. It is lost when QGIS closes.

Thiessen polygons Polygons that surround each point with all the area closer to it than to any other point.

TIGER/Line The Census Bureau's boundary files for states, counties, tracts, roads, and other features.

TIN Triangulated irregular network: a surface model that joins sampled elevation points into triangles, using more triangles where the ground is rough.

Tissot's indicatrix Identical small circles drawn on the globe and then projected. Their size and shape on the map show the projection's distortion.

Tobler's first law of geography "Everything is related to everything else, but near things are more related than distant things" (Tobler 1970).

topology The recorded connections between features, such as which polygons share an edge and which lines meet at a node. It keeps neighbors from having gaps or overlaps.

transformation Equations that convert image coordinates into map coordinates, with coefficients solved from ground control points.

Transverse Mercator A Mercator projection turned on its side, so the cylinder touches the Earth along a central meridian. It suits regions that run north to south.

trend surface A smooth polynomial surface fit to all the observations, usually by least squares. Its order sets how many bends it can have.

triangulation An old surveying method that measures the angles of a chain of triangles between stations to work out their positions across a large area.

unary operation A raster operation with one input raster, such as A × 2. A binary operation has two, such as A + B.

unclassed map A choropleth map in which each area gets its own shade, in proportion to its value, instead of being grouped into classes.

uniform kernel A kernel that gives every point inside the bandwidth the same weight.

union The records in either of two sets. OR selects the union.

universal kriging Kriging that models the trend as a function of location, such as a polynomial, and krigs the residuals.

Universal Transverse Mercator (UTM) A system of 60 zones, each 6° of longitude wide, each a secant Transverse Mercator projection in meters. North Carolina spans zones 17N and 18N.

validation In the HOT Tasking Manager, a check of a finished square by a second, experienced mapper.

vector A data model that stores the world as points, lines, and polygons, each defined by x, y coordinates and linked to its attributes. Best for discrete objects.

vectorize To convert a raster into vector features.

Venn diagram A drawing of sets as overlapping circles, useful for checking what an AND or OR query will select.

veracity Whether data can be trusted; sometimes called the fourth V of Big Data.

vertex An intermediate point where a line or polygon boundary changes direction.

volunteered geographic information Geographic data created by individuals, most of them not trained in mapping, and shared on the internet (Goodchild 2007). OpenStreetMap is the best-known example.

Web Mercator The Mercator variant used by nearly every web map (EPSG:3857). Good for display, never for measuring distance or area.



WGS84 The World Geodetic System of 1984, the global, earth-centered datum used by GPS. It differs from NAD83 by about 1 to 2 meters in North America.

WHERE clause The condition part of an SQL query, such as `"Area" > 20`. It is the part you write in Select by Expression or Select By Attributes.

ZIP Code Tabulation Area (ZCTA) An approximation of a ZIP Code's area built from census blocks.

ZIP+4 A nine-digit ZIP Code whose last four digits identify a short segment of a mail route, such as one side of a block.

zonal function A map algebra function that summarizes the cells within each zone, such as the mean elevation of each watershed.

zonal statistics A statistic, such as the mean or sum, computed for all the raster cells within each zone of a zone layer.

About the author



I'm **Dr. David Parr**, a Teaching Assistant Professor in the Department of Geography and the Environment at the University of North Carolina at Chapel Hill. I teach classes in spatial data science, GIS, programming, web mapping, spatial databases, cartography, spatial modeling and analysis, and data acquisition and integration.

I see my disciplines, Geographic Information Science (GIS) and spatial data science, as the intersections of Geography and Computer Science, the fields in which I have earned my degrees. I want students to be able to *think geographically*: to critically consider the dynamic spatial and temporal issues of a problem, and to consider how scale and distance affect our human and physical environments. I also want them to *think computationally*: to break problems into smaller components in a logical sequence, and to consider how a representation in a computer is related to the object in the real world. These skills, understanding spatial relations and logical problem solving, are critical in our discipline and in the wider world.

My teaching and research interests include volunteered geographic information, space-time spatial analysis and modeling, GIS and geography education, and GIS and society, including digital places and web GIS.

- **Email:** daveparr@unc.edu
- **Website:** daveparr.com
- **Office:** 317B Carolina Hall, UNC-Chapel Hill

About this book

This book was developed and written from several years of lectures for GEOG 370, Geographic Information Science. It is a Quarto book.

References

The books, articles, and documentation cited in this book.

Beyer, Mark A., and Douglas Laney. 2012. *The Importance of "Big Data": A Definition*. Gartner.

Bolstad, Paul. 2022. *GIS Fundamentals: A First Text on Geographic Information Systems*. 7th ed. XanEdu.

Brewer, Cynthia A. 2016. *Designing Better Maps: A Guide for GIS Users*. 2nd ed. Esri Press.

Brewer, Cynthia A., and Linda Pickle. 2002. "Evaluation of Methods for Classifying Epidemiological Data on Choropleth Maps in Series." *Annals of the Association of American Geographers* 92 (4): 662–81. https://doi.org/10.1111/1467-8306.00310.

Codd, E. F. 1970. "A Relational Model of Data for Large Shared Data Banks." *Communications of the ACM* 13 (6): 377–87. https://doi.org/10.1145/362384.362685.

Environmental Systems Research Institute. 1998. *ESRI Shapefile Technical Description*. An {ESRI} White Paper. Environmental Systems Research Institute.

Fan, Hongchao, Alexander Zipf, Qing Fu, and Pascal Neis. 2014. "Quality Assessment for Building Footprints Data on OpenStreetMap." *International Journal of Geographical Information Science* 28 (4): 700–719. https://doi.org/10.1080/13658816.2013.867495.

Goldberg, Daniel W. 2008. *A Geocoding Best Practices Guide*. North American Association of Central Cancer Registries.

Goodchild, Michael F. 1992. "Geographical Information Science." *International Journal of Geographical Information Systems* 6 (1): 31–45. https://doi.org/10.1080/02693799208901893.

Goodchild, Michael F. 2007. "Citizens as Sensors: The World of Volunteered Geography." *GeoJournal* 69 (4): 211–21. https://doi.org/10.1007/s10708-007-9111-y.

Haklay, Mordechai. 2010. "How Good Is Volunteered Geographical Information? A Comparative Study of OpenStreetMap and Ordnance Survey Datasets." *Environment and Planning B: Planning and Design* 37 (4): 682–703. https://doi.org/10.1068/b35097.

Haklay, Mordechai, Sofia Basiouka, Vyron Antoniou, and Aamer Ather. 2010. "How Many Volunteers Does It Take to Map an Area Well? The Validity of Linus' Law to Volunteered Geographic Information." *The Cartographic Journal* 47 (4): 315–22. https://doi.org/10.1179/000870410X12911304958827.

Harrower, Mark, and Cynthia A. Brewer. 2003. "ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps." *The Cartographic Journal* 40 (1): 27–37. https://doi.org/10.1179/000870403235002042.

Holloway, Paul. 2023. *Understanding GIS Through Sustainable Development Goals: Case Studies with ArcGIS Pro and QGIS*. Routledge.

Jenks, George F. 1977. *Optimal Data Classification for Choropleth Maps*. Occasional Paper No. 2. Department of Geography, University of Kansas.

Jensen, John R. 2015. *Introductory Digital Image Processing: A Remote Sensing Perspective*. 4th ed. Pearson.

Krige, Danie G. 1951. "A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand." *Journal of the Chemical, Metallurgical and Mining Society of South Africa* 52 (6): 119–39.

Malys, Stephen, John H. Seago, Nikolaos K. Pavlis, P. Kenneth Seidelmann, and George H. Kaplan. 2015. "Why the Greenwich Meridian Moved." *Journal of Geodesy* 89 (12): 1263–72. https://doi.org/10.1007/s00190-015-0844-y.

Matheron, Georges. 1963. "Principles of Geostatistics." *Economic Geology* 58 (8): 1246–66. https://doi.org/10.2113/gsecongeo.58.8.1246.

McHarg, Ian L. 1969. *Design with Nature*. Natural History Press.

Mitchell, Andy. 2020. *The Esri Guide to GIS Analysis, Volume 1: Geographic Patterns and Relationships*. 2nd ed. Esri Press.

Monmonier, Mark. 2018. *How to Lie with Maps*. 3rd ed. University of Chicago Press.

National Research Council. 2006. *Learning to Think Spatially: GIS as a Support System in the K–12 Curriculum*. The National Academies Press. https://doi.org/10.17226/11019.

O'Reilly, Tim. 2005. *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. O'Reilly Media, September 30.

O'Sullivan, David, and David J. Unwin. 2010. *Geographic Information Analysis*. 2nd ed. Wiley.

Oort, Pepijn A. J. van. 2006. "Spatial Data Quality: From Description to Application." PhD thesis, Wageningen University.

Parr, David. 2015. "Crisis Mapping and the Nepal Earthquake: The Impact of New Contributors." *Kartographische Nachrichten*.

Shepard, Donald. 1968. "A Two-Dimensional Interpolation Function for Irregularly-Spaced Data." *Proceedings of the 1968 23rd ACM National Conference*, 517–24. https://doi.org/10.1145/800186.810616.

Silverman, Bernard W. 1986. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall.

Slocum, Terry A., Robert B. McMaster, Fritz C. Kessler, and Hugh H. Howard. 2022. *Thematic Cartography and Geovisualization*. 4th ed. CRC Press.

Snow, John. 1855. *On the Mode of Communication of Cholera*. 2nd ed. John Churchill.

Snyder, John P. 1987. *Map Projections: A Working Manual*. U.s. Geological Survey Professional Paper 1395. U.S. Government Printing Office. https://doi.org/10.3133/pp1395.

SQLite Project. 2024. *Most Widely Deployed and Used Database Engine*. Https://www.sqlite.org/mostdeployed.html.

Stevens, S. S. 1946. "On the Theory of Scales of Measurement." *Science* 103 (2684): 677–80. https://doi.org/10.1126/science.103.2684.677.

Sturges, Herbert A. 1926. "The Choice of a Class Interval." *Journal of the American Statistical Association* 21 (153): 65–66. https://doi.org/10.1080/01621459.1926.10502161.

Thiessen, Alfred H. 1911. "Precipitation Averages for Large Areas." *Monthly Weather Review* 39 (7): 1082–89.

Tobler, Waldo R. 1970. "A Computer Movie Simulating Urban Growth in the Detroit Region." *Economic Geography* 46 (sup1): 234–40. https://doi.org/10.2307/143141.

Tobler, Waldo R. 1973. "Choropleth Maps Without Class Intervals?" *Geographical Analysis* 5 (3): 262–65. https://doi.org/10.1111/j.1538-4632.1973.tb01012.x.

Tomlin, C. Dana. 1990. *Geographic Information Systems and Cartographic Modeling*. Prentice Hall.

U.S. Census Bureau. 2022. *Census Bureau Releases Estimates of Undercount and Overcount in the 2020 Census*. Press release, March 10. https://www.census.gov/newsroom/press-releases/2022/2020-census-estimates-of-undercount-and-overcount.html.

U.S. Census Bureau. 2023. *Uses of Decennial Census Programs Data in Federal Funds Distribution: Fiscal Year 2021*. U.S. Census Bureau.

Wilkinson, Mark D. et al. 2016. "The FAIR Guiding Principles for Scientific Data Management and Stewardship." *Scientific Data* 3: 160018. https://doi.org/10.1038/sdata.2016.18.

Wroclawski, Serge. 2014. *Why the World Needs OpenStreetMap*. The Guardian, January 14.

Ziemann, Mark, Yotam Eren, and Assam El-Osta. 2016. "Gene Name Errors Are Widespread in the Scientific Literature." *Genome Biology* 17: 177. https://doi.org/10.1186/s13059-016-1044-7.

Zook, Matthew, Mark Graham, Taylor Shelton, and Sean Gorman. 2010. "Volunteered Geographic Information and Crowdsourcing Disaster Relief: A Case Study of the Haitian Earthquake." *World Medical & Health Policy* 2 (2): 7–33. https://doi.org/10.2202/1948-4682.1069.