Alexander Abuabara
DAEN 489 - Spring 2026
College Station 13- and 22-Jan-2026

# Week 1 and 2

## Contents



## 1 About Me

https://abuabara.github.io/

# 2 Course Structure

We meet **once per week**, in person, on **Thursdays 12:45–3:15 PM**. Each class session integrates theoretical material with hands-on activities using spatial datasets and open-source tools. The course is cumulative; later assignments and the final project assume mastery of earlier material.

- **Weeks 1–7**: Foundations of spatial data engineering, including coordinate systems, spatial ETL, vector and raster operations, geometry, and raster–vector interaction.
- **Week 8**: In-class midterm exam covering Weeks 1–8.
- **Weeks 9–13**: Advanced topics including cartographic design, automated workflows, large-scale spatial data, web GIS, spatial statistics/modeling.
- **Weeks 14–15**: Project presentations and final project report submission.

You should be comfortable with:

- Basic data structures and statistics
- Programming in `R` (and `Python`)
  (→ check `dplyr`, `tidyverse`, `units` in `R`)

You are **expected** to:

- Maintain academic integrity.
- Complete assigned readings prior to class.
- Participate in discussions, quizzes, and in-class problem-solving.
- Submit work that is clear, reproducible, and professionally presented.

**Collaboration** is encouraged for ideas and troubleshooting:

- All assignments must be completed and submitted individually.
- The course project is the only graded group component, and must be completed in groups of two.

## Assignments and Project Milestones

| Deliverable | Due Week[1] |
|---|---|
| Assignment 1: Projection Distortion | Week 2 |
| Assignment 2: Spatial ETL Workflow | Week 4 |
| Assignment 3: Raster–Vector Interaction | Week 6 |
| Assignment 4: Data Cube Construction | Week 8 |
| Assignment 5: Engineering Decision-Making | Week 10 |
| Project Proposal | Week 11 |
| Project Presentation | Week 14 |
| Final Project Report | Week 15 |

[1] Besides in-class presentation, due date for submissions will be Fridays 5 pm.

### Assignments

Details about assignments and practice questions at the end of each presentation.

### Project

The course project focuses on applied spatial analysis in an engineering context:

- Integration of large-scale spatial datasets
- Automated and reproducible workflows
- High-quality maps and visualizations
- A short in-class presentation (approximately 5 minutes)
- A written report (approximately 12 pages)

More details will be available on each assignment instructions, stay tuned.

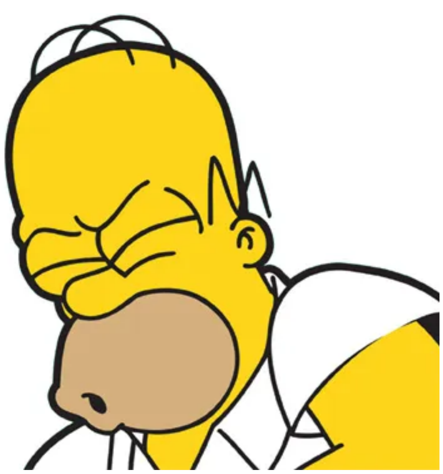
## Grading

| Component | Weight |
|---|---|
| Attendance and Participation | 10% |
| Assignments | 30% |
| Midterm Exam | 30% |
| Course Project (group of 2) | 30% |

- Regrades apply to the entire submission, must be made within 1-week.
  No changes are possible after that window.
- Use of AI-based text or code generation tools is permitted only with explicit acknowledgment and appropriate attribution, and only when such use meaningfully supports the student's comprehension of the concepts, methods, and material being assessed, and not as a substitute for independent understanding, reasoning, or analysis.

**Assessment**

| **Criterion** | **Description** | **Points** |
|---|---|---|
| Problem Understanding | Clear understanding of the problem, requirements, objectives, and constraints. | 0–2 |
| Methodology/Approach | Approach is appropriate, structured. Assumptions clearly stated. | 0–2 |
| Technical | Analysis, calculations, code are correct and properly implemented. | 0–3 |
| Results/Interpretation | Results are accurate, clearly presented, and correctly interpreted and reproducible. | 0–2 |
| Presentation | Work is well-organized, clearly written, and professionally presented. | 0–1 |
| **Total** | | **10** |



**Late Work**

- Late submissions are **not accepted** and receive a grade of zero.
- Applies to assignments, project components, and the final report.
- Excused absences apply only to attendance and in-class activities and do **not** extend assignment deadlines.
- Any alternative arrangements must be approved **prior** to the deadline.

# 3 Resources

### Textbook

- Lecture notes (posted on Canvas)
- *R for Data Science.* https://r4ds.hadley.nz
- *Essentials of GIS.* https://open.umn.edu/opentextbooks/textbooks/67
- *Spatial Data Science With Applications in R.* https://r-spatial.org/book
- *Geocomputation with R.* https://r.geocompx.org
- *Geocomputation with R in Docker* https://hub.docker.com/r/geocompr/geocompr
- *Geocomputation with Python.* https://py.geocompx.org

### Software

- QGIS https://qgis.org
- R https://cran.r-project.org → RStudio https://posit.co/rstudio-desktop
- Python https://www.python.org → Positron https://positron.posit.co
- Several libraries and plugins will be cited throughout the presentations.

# 4 Readings for Today

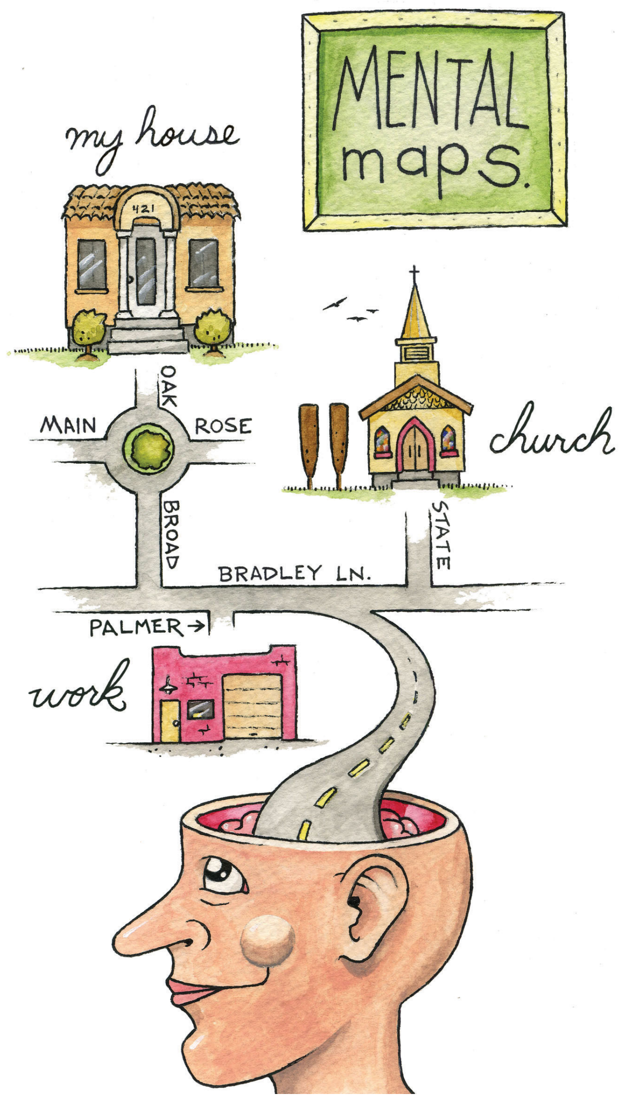
- Essentials of Geographic Information Systems, Chapters 1-2-3
  https://saylordotorg.github.io/text_essentials-of-geographic-information-systems/index.html
- Spatial Data Science, All Spatial Data https://r-spatial.org/book/part-1.html
- Map projections: A working manual
  https://www.usgs.gov/publications/map-projections-a-working-manual
- Map projections https://pubs.usgs.gov/gip/70047422/report.pdf
- Choosing the right map projection
  https://source.opennews.org/articles/choosing-right-map-projection/

# 5 Spatial Thinking

**"This Is Not a Pipe"** René Magritte


*"The famous pipe. How people reproached me for it! And yet, could you stuff my pipe? No, it's just a representation, is it not? So if I had written on my picture "This is a pipe", I'd have been lying!"* René Magritte
Wikipedia: The Treachery of Images
https://en.wikipedia.org/wiki/The_Treachery_of_Images

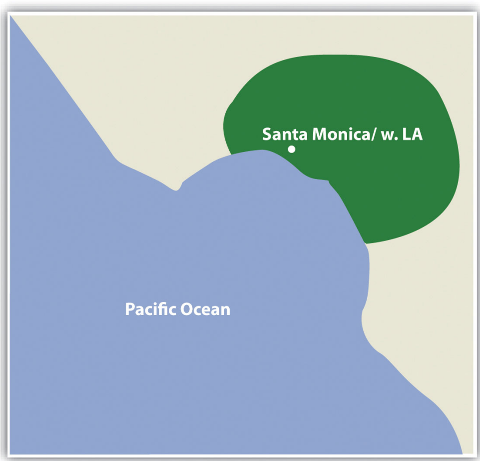
Mental or cognitive maps are psychological tools that we all use every day. As the name suggests, mental maps are maps of our environment that are stored in our brain. We rely on our mental maps to get from one place to another, to plan our daily activities, or to understand and situate events that we hear about from our friends, family, or the news. Mental maps also reflect the amount and extent of geographic knowledge and spatial awareness that we possess. To illustrate this point, pretend that a friend is visiting you from out of town for the first time. Using a blank sheet of paper, take five to ten minutes to draw a map from memory of your hometown that will help your friend get around.

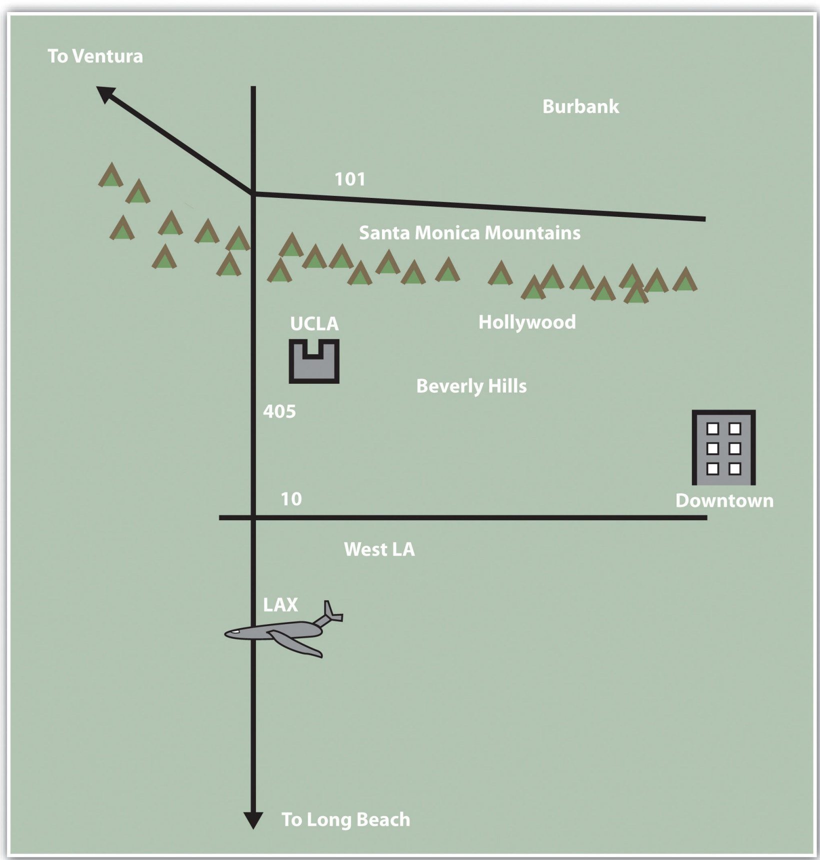


This simple exercise is instructive for several reasons:

- First, it illustrates what you know about where you live. Your simple map is a rough approximation of your local geographic knowledge and mental map.
- Second, it highlights the way in which you relate to your local environment. What you choose to include and exclude on your map provides insights about what places you think are important and how you move through your place or residence.
- Third, if we were to compare your mental map to someone else's from the same place, certain similarities emerge that shed light upon how we as humans tend to think spatially and organize geographical information in our minds.
- Fourth, this exercise reveals something about your artistic, creative, and cartographic abilities. In this respect, not only are mental maps unique, but also the way in which such maps are drawn or represented on the page is unique too.

To reinforce these points, consider the mental maps of Los Angeles. Each map is probably an imperfect representation, but we can see some similarities and differences that provide insights into how people relate to Los Angeles (and maps).

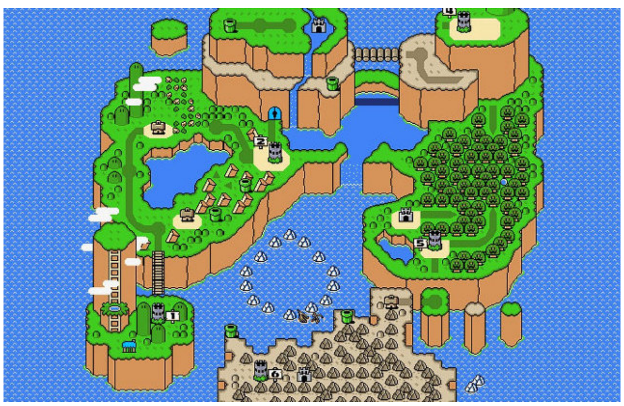


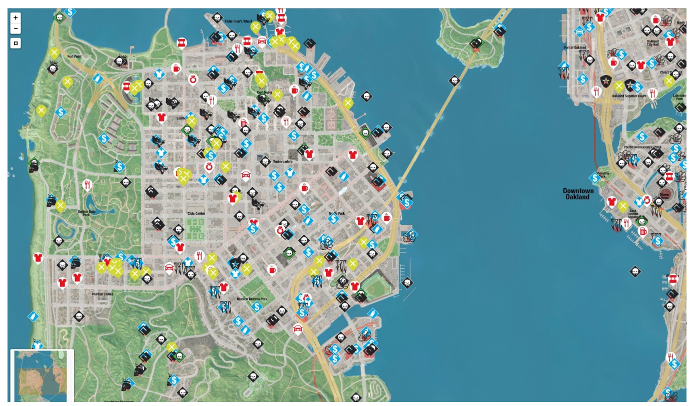



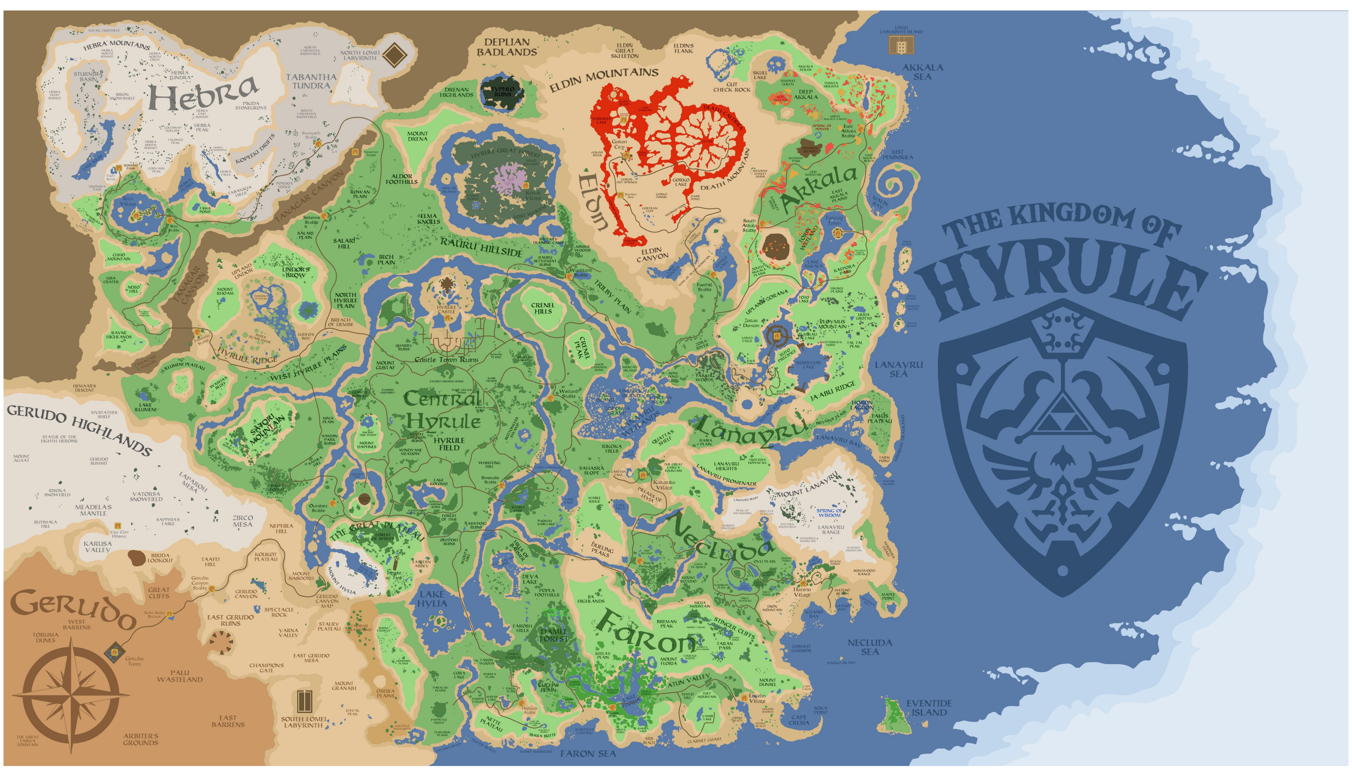
## Defining Video Game Map Types

https://www.dailygamedesigns.com/games/755-defining-video-game-map-types/

- Maps in video games are deeply interesting. In general, it's rare that a map actually functions in a coherent way. That is, in a way that makes sense for anything to function. They live by their own rules, somewhat.
- There are plenty of different types of maps, but generally they seem to fall into a few categories: a map that "reveals" itself as you explore parts of it, a detailed map that is entirely revealed for you from the start, some version of either of those but extremely simplified or obfuscated, or some version of any of these while emulating modern map technology. Below: Super Mario World (SNES, 1991), Watch Dogs 2 (PC/PS4/XBOX1, 2016), Breath of the Wild (Nintendo Switch, 2017).







### **"The Map is Not the Territory"** Alfred Korzybski

Our models, perceptions, words, and beliefs (the "map") are not reality itself (the "territory"), but simplified representations that can be incomplete, misleading, or different from the actual complex world.

This core idea in general semantics highlights that representations are useful but not identical to the actual thing, urging us to test our mental maps against reality to avoid errors in judgment or action, like confusing data with the lived experience.

Do not to confuse our understanding with what truly is:

- **Map:** Mental models, language, beliefs, data, theories, any reality abstraction.
- **Territory:** Reality itself, the world as it exists, independent of our perception.

Maps are inherently simplified, selective, subject to the mapmaker's perspective, scale, and errors, while the territory is rich, complex, and constantly changing. Examples:

- A financial forecast (map) is not the actual market performance (territory).
- A written description of a flavor (map) isn't the same as tasting it (territory).

So what:

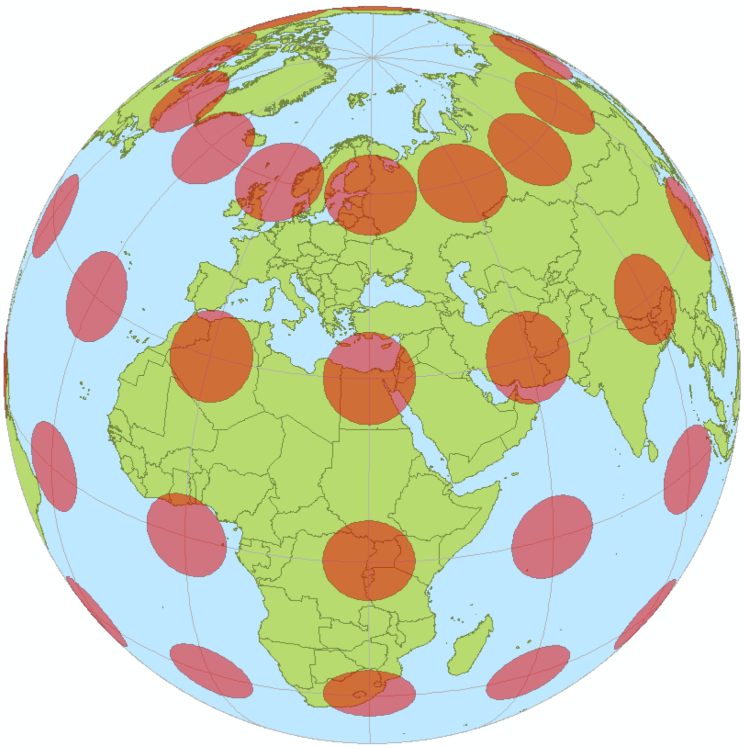
- **Avoid Dogmatism:** Treating models as absolute truth; they need updating.
- **Avoid Misunderstandings:** Acknowledge that your internal map of a situation might differ from someone else's.
- **Actionable Insight:** Test your understanding (the map) by engaging with the actual situation (the territory) to see if it holds up (e.g., Jeff Bezos verified customer service wait times instead of relying solely on data).

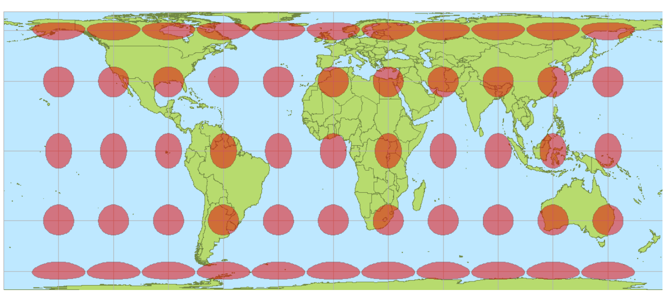
**Cartagrapher** (argument mapping) https://johnmacfarlane.net/cartargrapher.html

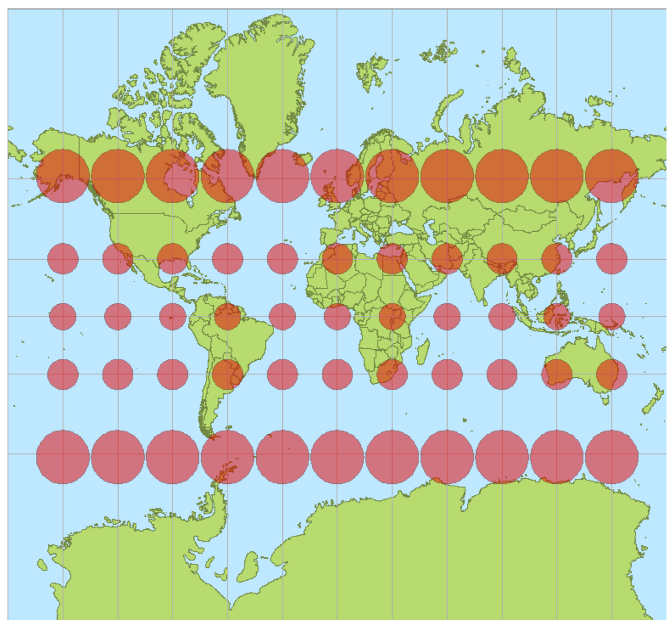
**Map–territory relation**

Tissot's indicatrices viewed on a sphere: All are identical circles.

https://en.wikipedia.org/wiki/Tissot%27s_indicatrix



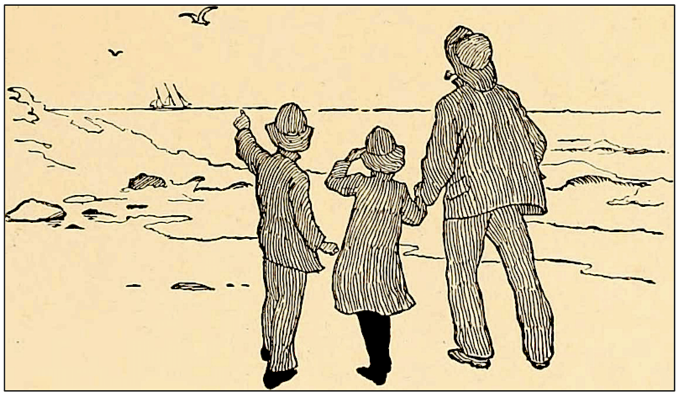




The indicatrices demonstrate the difference between the 3D world as seen from space and 2D projections of its surface. The Behrmann projection with Tissot's indicatrices (left). The Mercator projection with Tissot's indicatrices (right).

https://en.wikipedia.org/wiki/Map-territory_relation

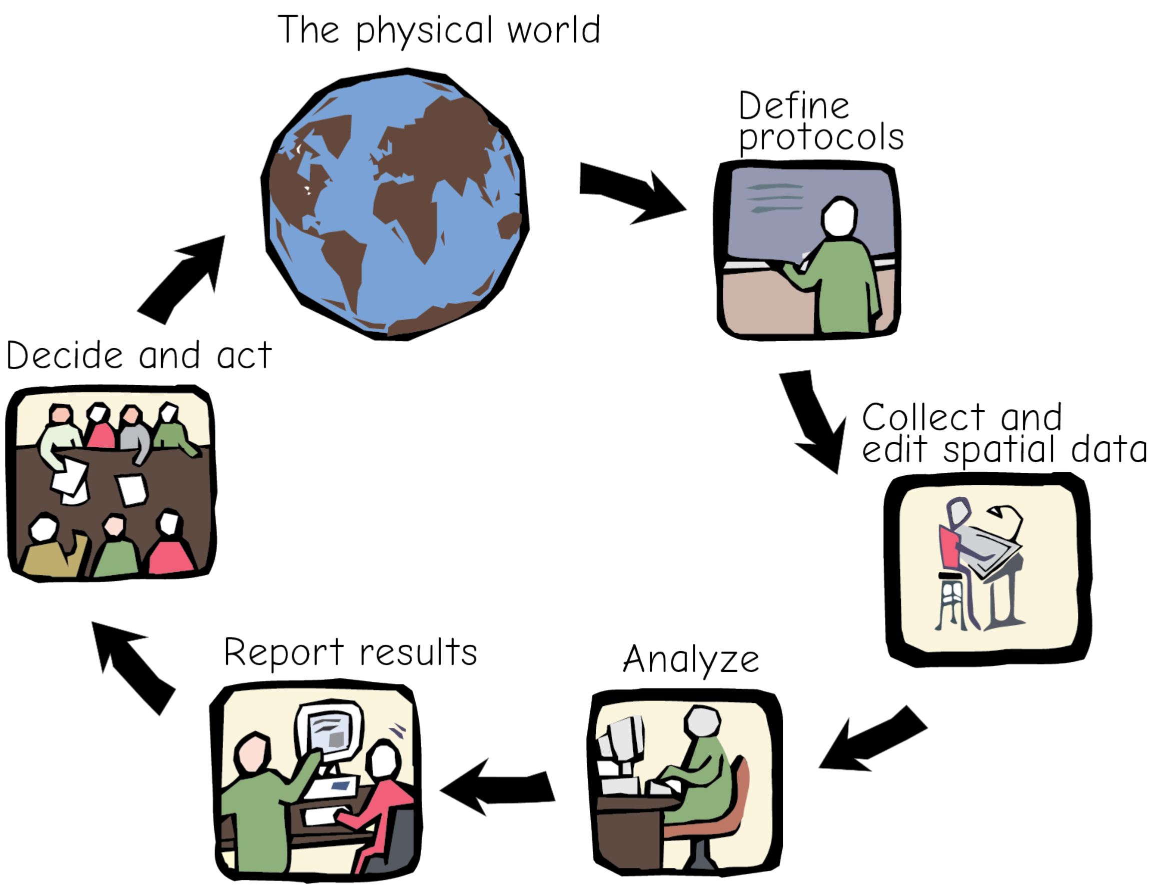
## “The Land Defines the Ocean, the Ocean Defines the Land”



*“The Seashore Book: Bob and Betty’s summer with Captain Hawes”* (1912, p.33)

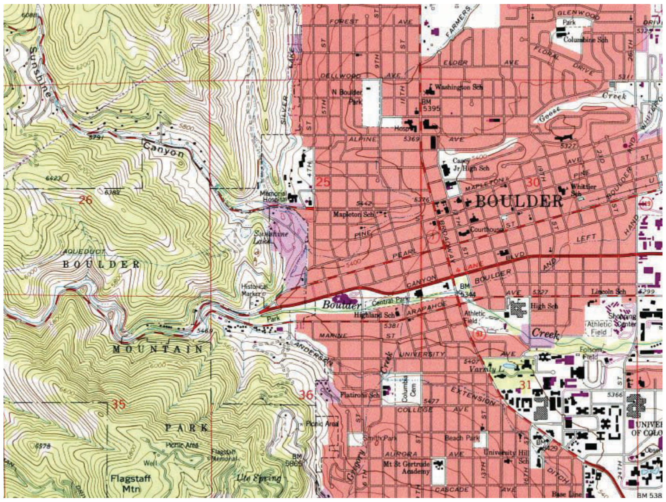
How space is encoded?

- Points, lines, polygons, and rasters as representational primitives (elements)
- Symbols, legends, and visual variables (size, color, shape, texture)
- Semantics: what spatial symbols mean vs. what they imply
- Ambiguity and misinterpretation in maps and spatial graphics

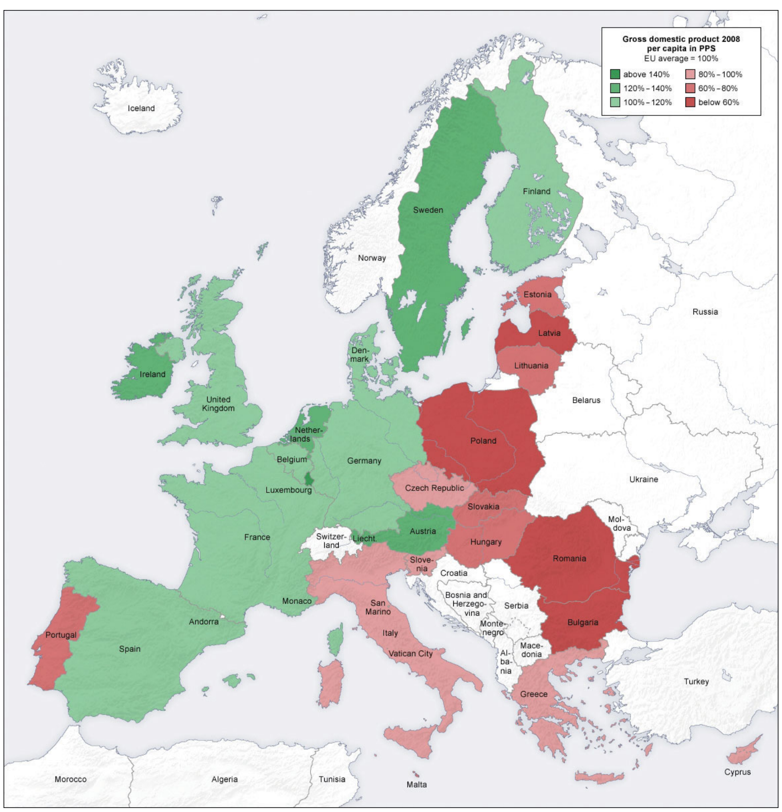


## Map Types

**1. Reference Maps:** The primary purpose of a reference map is to deliver location information to the map user. The accuracy of a given reference map is indeed critical to many users.

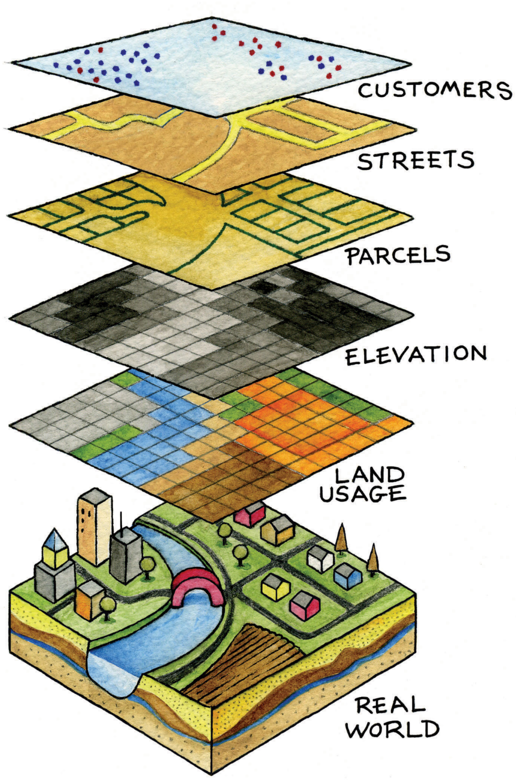



**2. Thematic Maps:** Contrasting the reference map are thematic maps. As the name suggests, thematic maps are concerned with a particular theme or topic of interest. While reference maps emphasize the location of geographic features, thematic maps are more concerned with how things are distributed across space.




Such things are often abstract concepts such as “per capita gross domestic product (GDP) in Europe”. One of the strengths of mapping, and in particular of thematic mapping, is that it can make such abstract and invisible concepts visible and comparable on a map. It is important to note that reference and thematic maps are not mutually exclusive. Thematic maps often contain geographical information, and reference maps may contain thematic information.

### Spatial Representation




In spatial information systems, space is encoded by reducing continuous geography into a small set of geometric elements that computers can store, index, and analyze:

1 **Points** represent discrete locations with no area or length—such as wells, sensors, trees, or intersections. They encode where something exists.

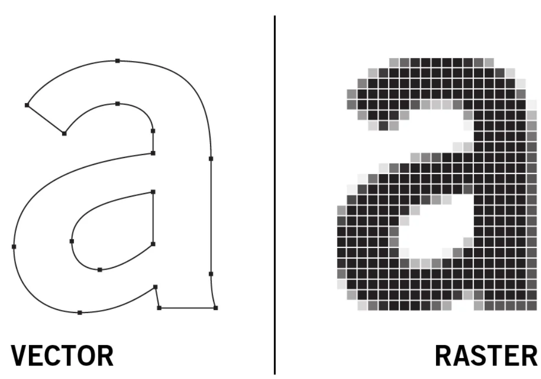
2 **Lines** represent linear features with length but negligible width—such as roads, rivers, pipelines, and boundaries. They encode how things connect and move through space.

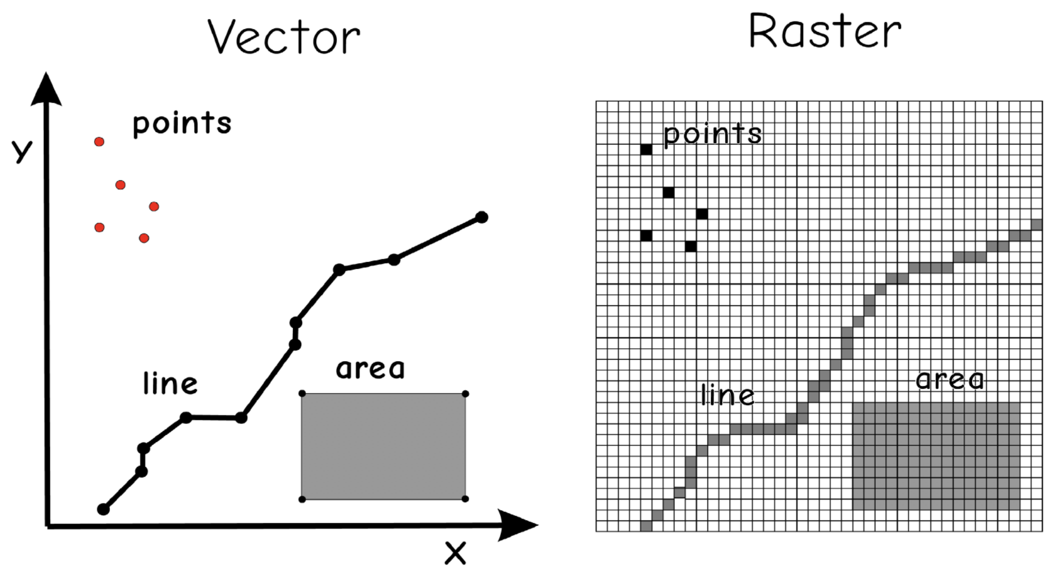
3 **Polygons** represent areal features enclosed by boundaries—such as parcels, lakes, land-use zones, and administrative regions. They encode occupied spaces.

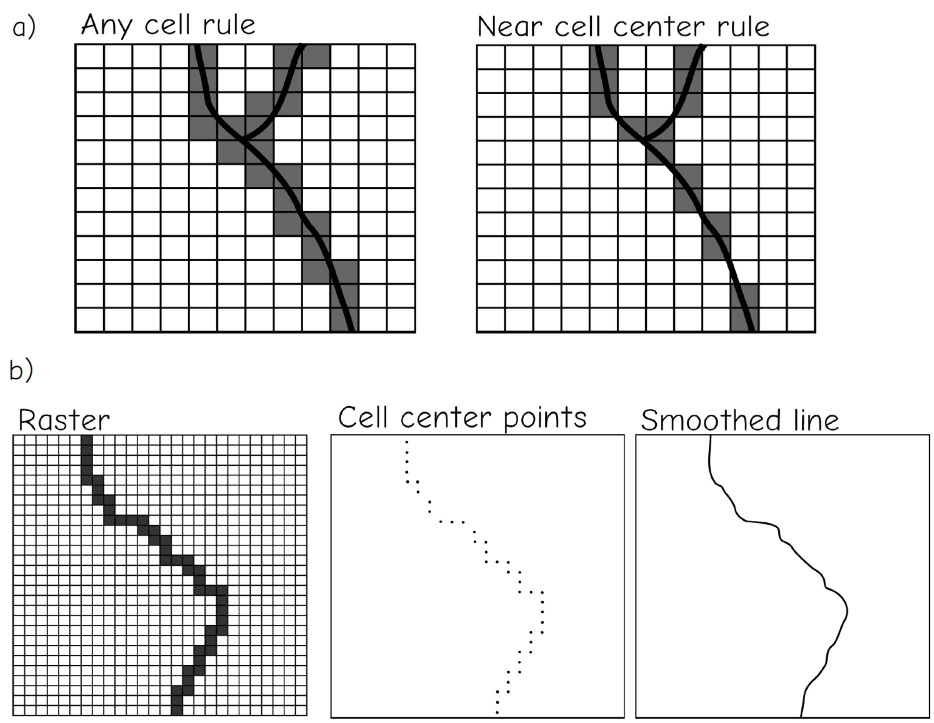
4 **Rasters** represent space as a continuous grid of cells (pixels), each storing a value—such as elevation, temperature, rainfall, or satellite reflectance. They encode how properties vary continuously across space.

Together, these primitives provide two complementary views of geography:

- **Vector Models** (points, lines, polygons) grab discrete objects and topology,
- **Raster Models** (grid of equally sized cells) capture continuous fields to be expressed in computable form.






### Visual Encoding

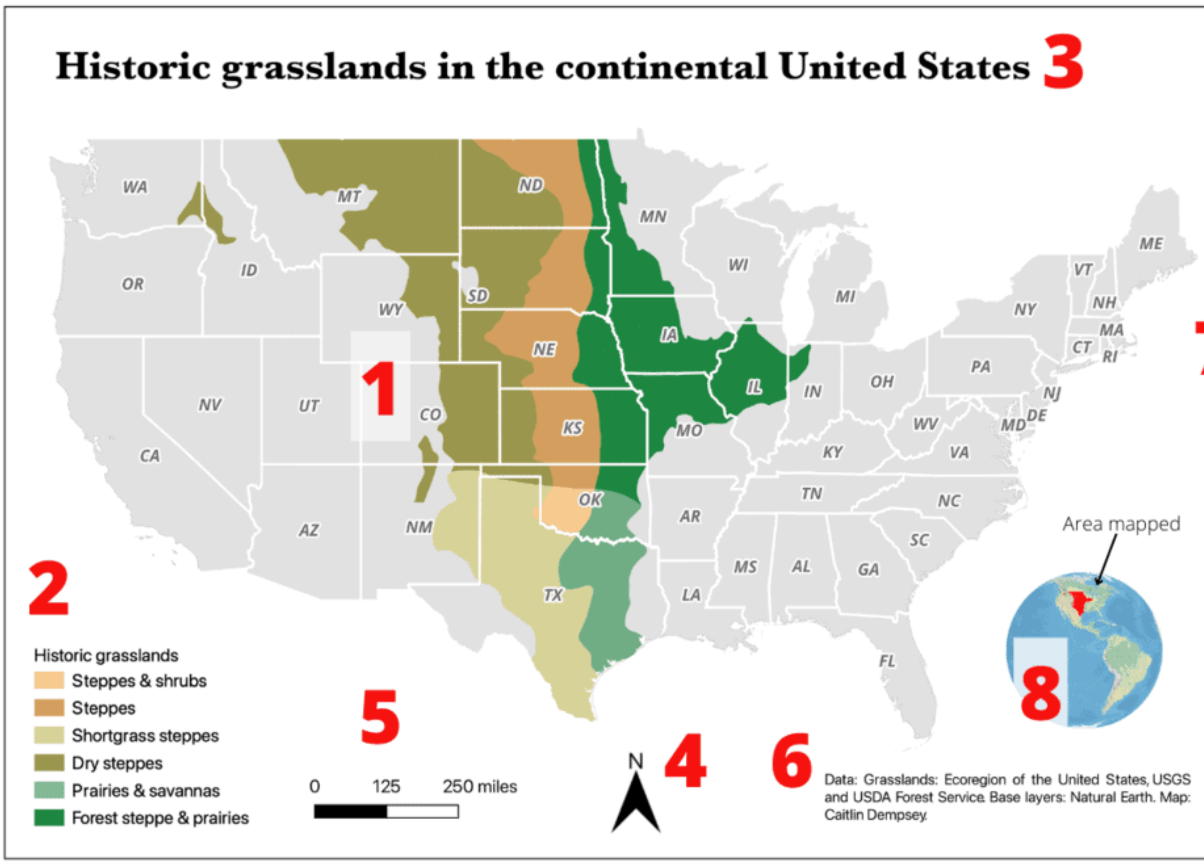
Space is encoded cartographically by translating geographic phenomena into symbols whose meanings are defined through legends and visually differentiated using visual variables.

- Symbols serve as graphical surrogates for spatial features and phenomena.
- Legends formalize the mapping between symbol forms and their semantic meaning, providing the interpretive key that makes a map readable.

- Visual variables (including size, color, shape, and texture) are used to structure how information is perceived: size conveys magnitude, color differentiates categories or intensities, shape distinguishes feature types, and texture suggests density or pattern.

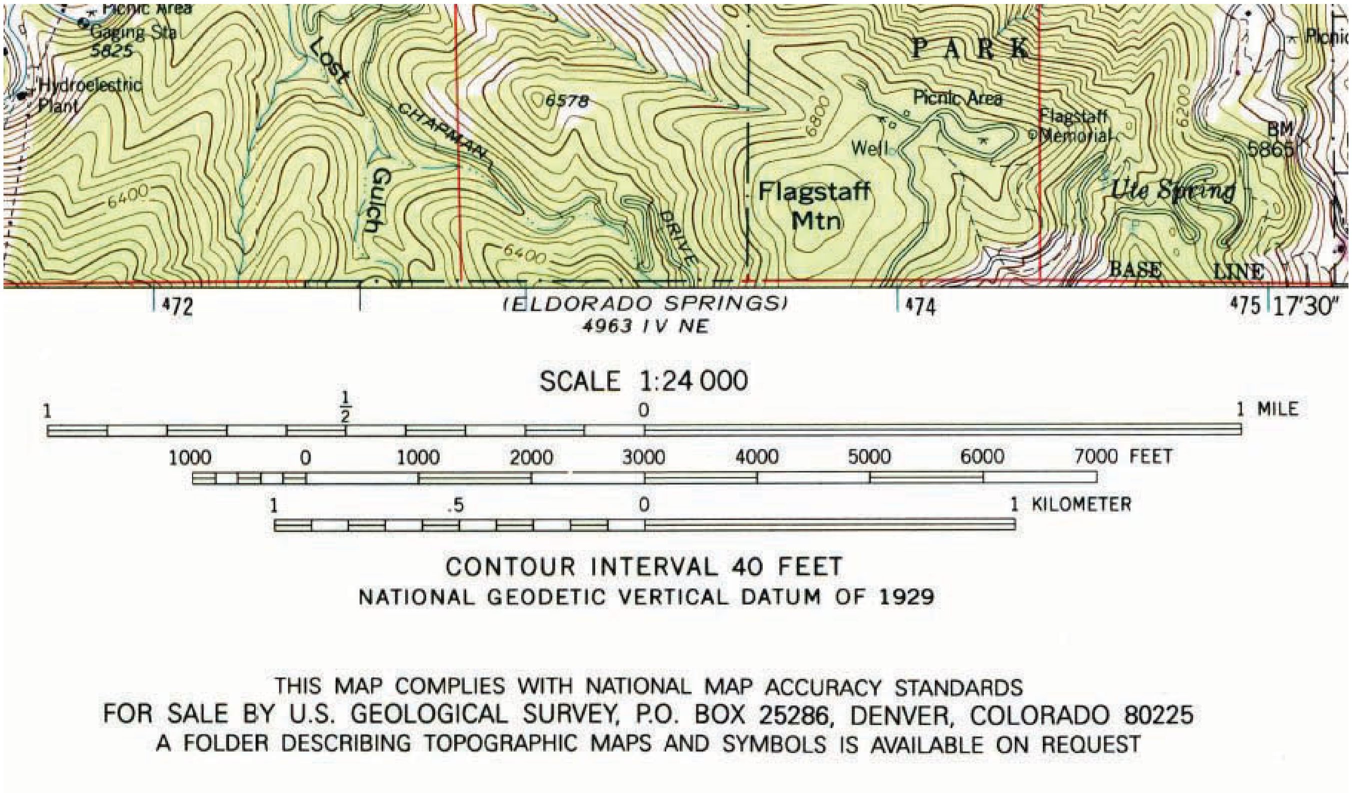
Together, these elements transform geographic space into an interpretable visual language, enabling complex spatial information to be efficiently communicated and cognitively processed.

What are the parts of a map?

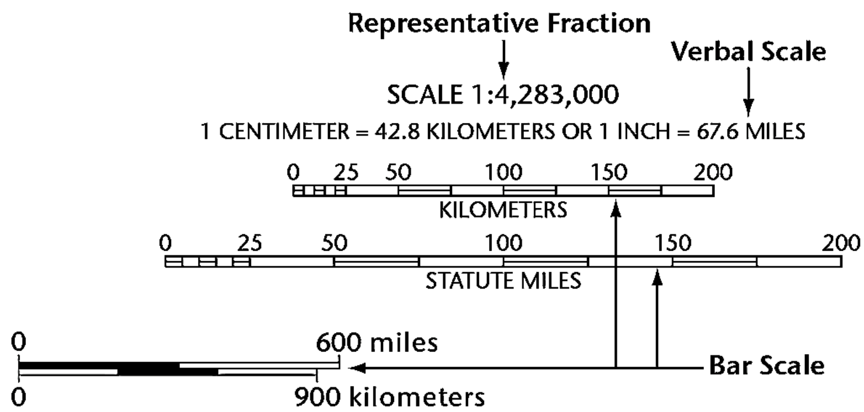


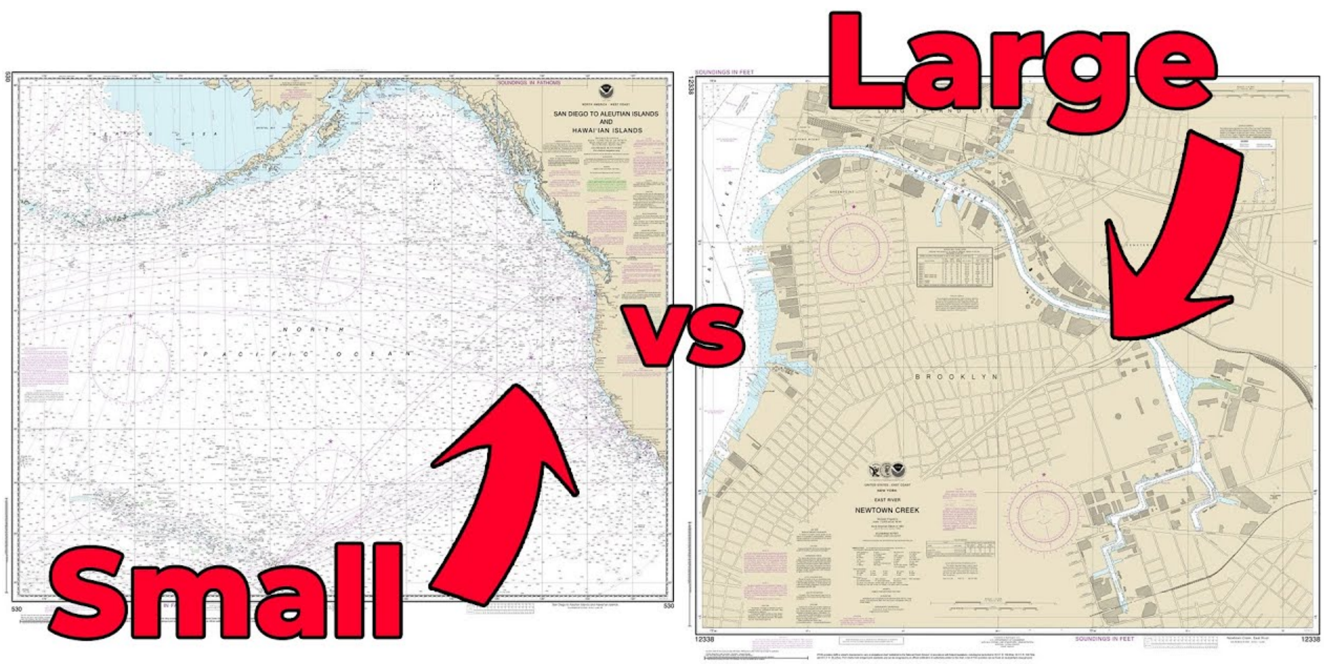

1. **Data Frame (Map):** Primary area, geographic features and data are displayed.
2. **Legend:** Explains the symbols, colors, and patterns used to represent features.
3. **Title:** States the subject and purpose of the map.
4. **North Arrow:** Indicates the orientation of the map relative to north.
5. **Scale Bar:** Relationship between distances on the map and real-world.
6. **Metadata:** Information about data sources, projection, authorship, and date.
7. **Border (Neatline):** A frame that visually contains the map and its elements.
8. **Inset Map (Locator):** A secondary map, provides geographic context.

## Map Scale




- Every map has a scale, which relates the size of the objects shown on the map to their sizes in real life.
- Scales are usually expressed in the form of two numbers, also called a ratio. For example, where a scale is 1 cm = 1km, that could be expressed as: 1:100,000. This means that anything that measures 1 unit on the map needs to be multiplied by 100,000 to work out how big it is in real life. A scale where 1 cm = 100,000 cm is the same as 1 cm = 1km.
  - The smaller the second number in the scale (i.e., 1:12), the larger the items on the map are, and the less that can be seen on the map. These are called **large scale** maps. These maps will be very detailed, for example a map of a city block, or part of a small country town.
  - The larger the second number in the scale (i.e., 1:500,000), the items are shown as smaller on the map, and more can be seen on the map. These are called **small scale** maps. These maps are not very detailed, for example, a map of the country.



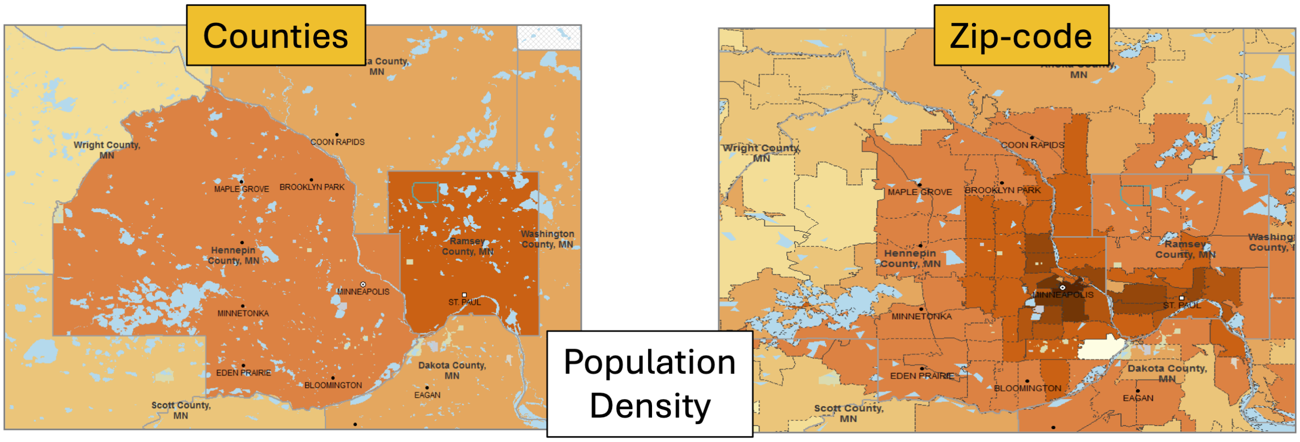

### Resolution vs. Accuracy vs. Precision

Resolution = your measuring stick

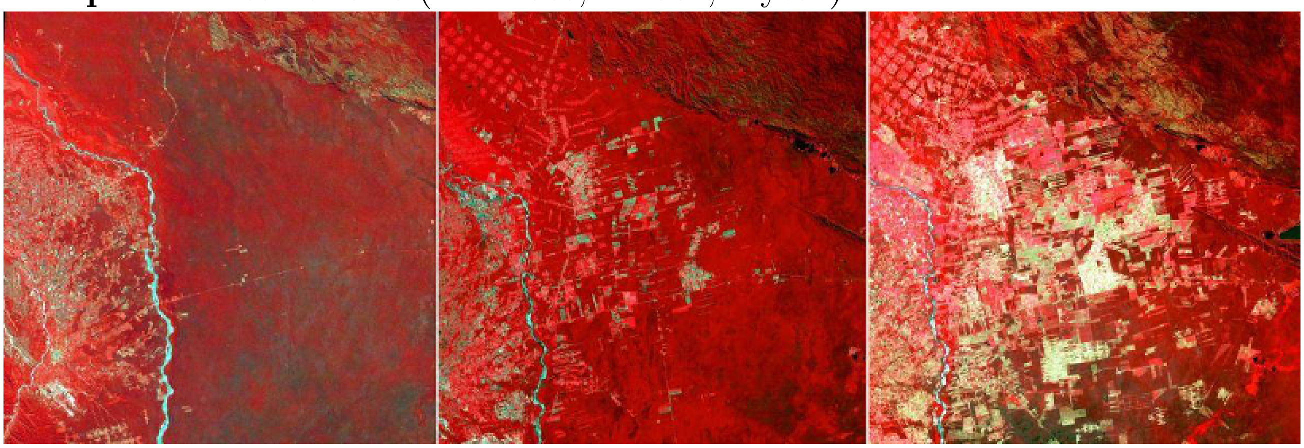
- Resolution determines the smallest feature or unit that can be accurately represented in a spatial dataset.
- It acts as the scale or granularity at which geographic phenomena are observed and analyzed.
- A higher resolution (e.g., 1 m spatial resolution) means a smaller measuring stick, capturing finer details.
- A lower resolution (e,g., 30 m spatial resolution) means a larger measuring stick, generalizing features and losing small details.

Types of resolution:

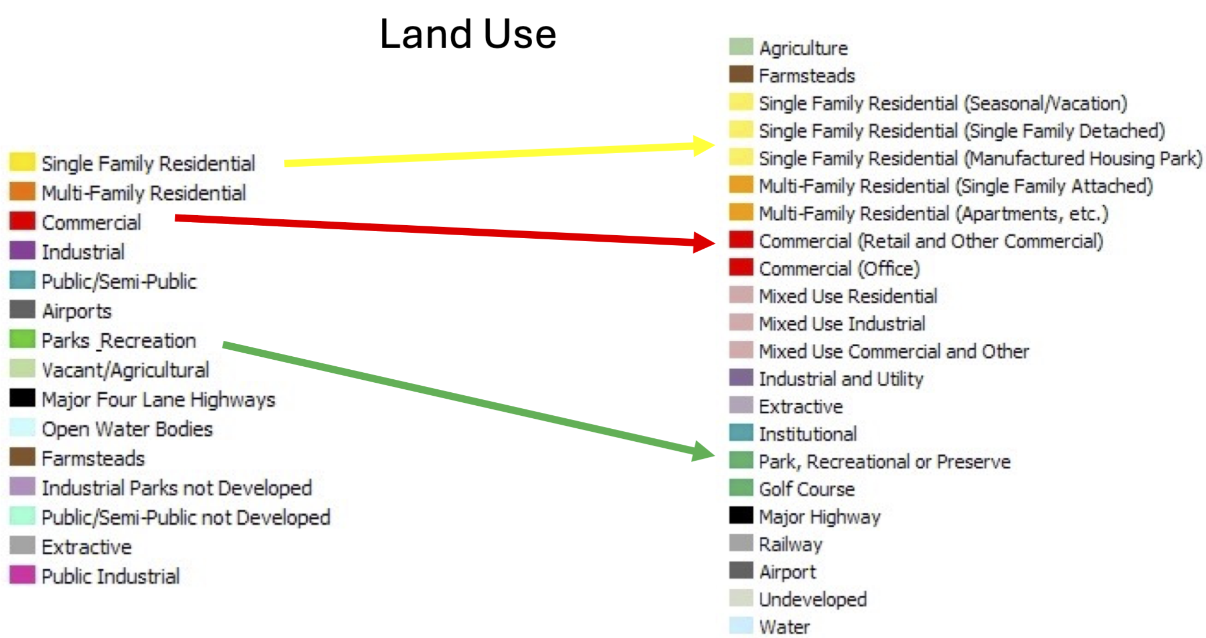
- **Spatial:** the grain of measurement (1 meter, 1 km, county, zip-code)




- **Temporal:** interval size (1 minute, 1 hour, 1 year)

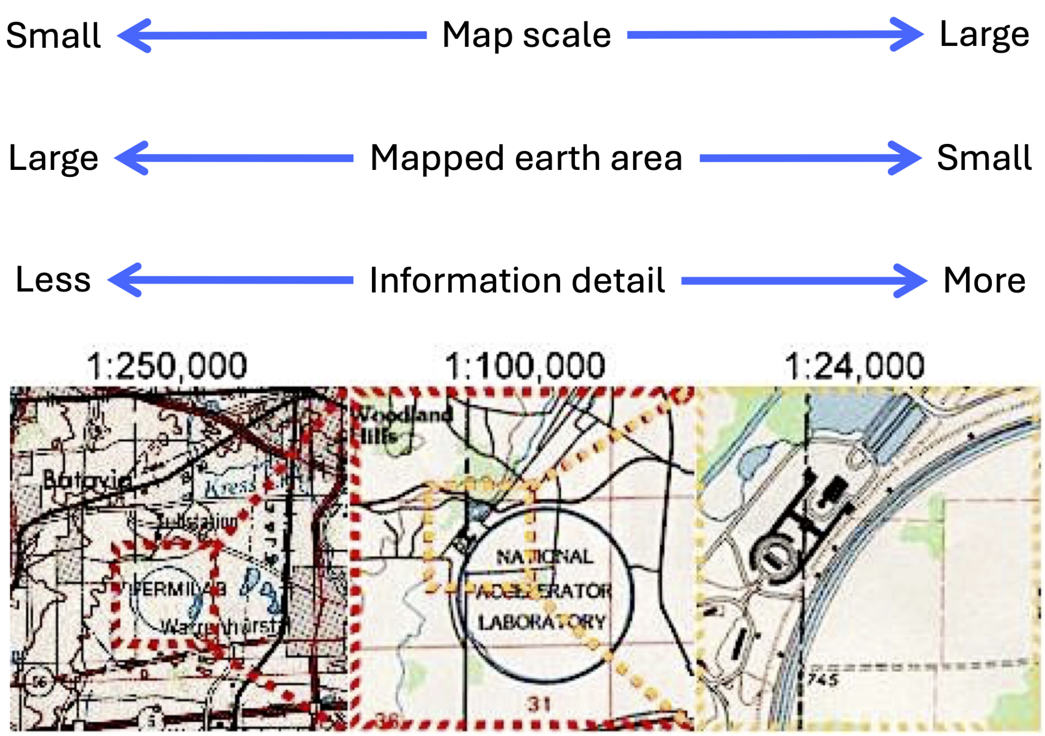


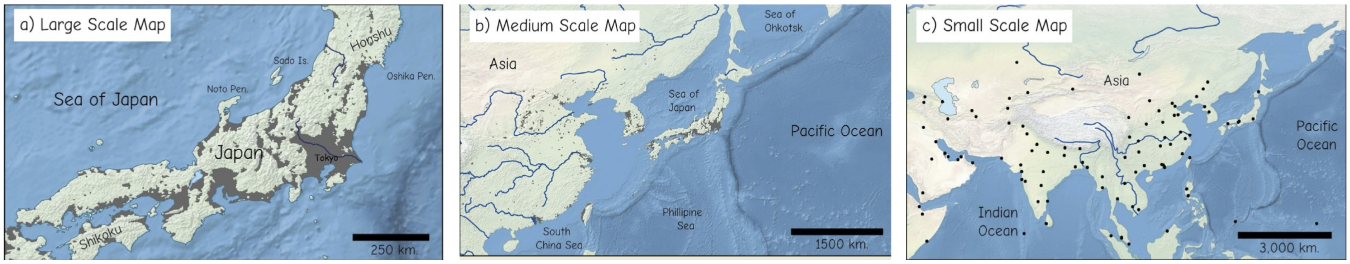
- **Attribute:** how detailed (urban vs. urban/suburban)



$$\text{Scale} = \text{Resolution} + \text{Extent}$$

- Resolution (level of detail)
  - the smallest unit of measurement in your dataset (your “measuring stick”).
  - Higher resolution = More detail (e.g., 1 m satellite imagery).
  - Lower resolution = Less detail (e.g., 30 m satellite imagery).
- Extent (geographic coverage)
  - The total area your dataset covers.
  - A large extent covers a big area (e.g., an entire country).
  - A small extent covers a localized area (e.g., a city block).
- Scale (combination of both)
  - The balance between resolution and extent.
  - If you have high resolution, you often have a smaller extent (because detailed data takes up more resources).
  - If you have low resolution, you can cover a larger extent (since less detail requires less resources).



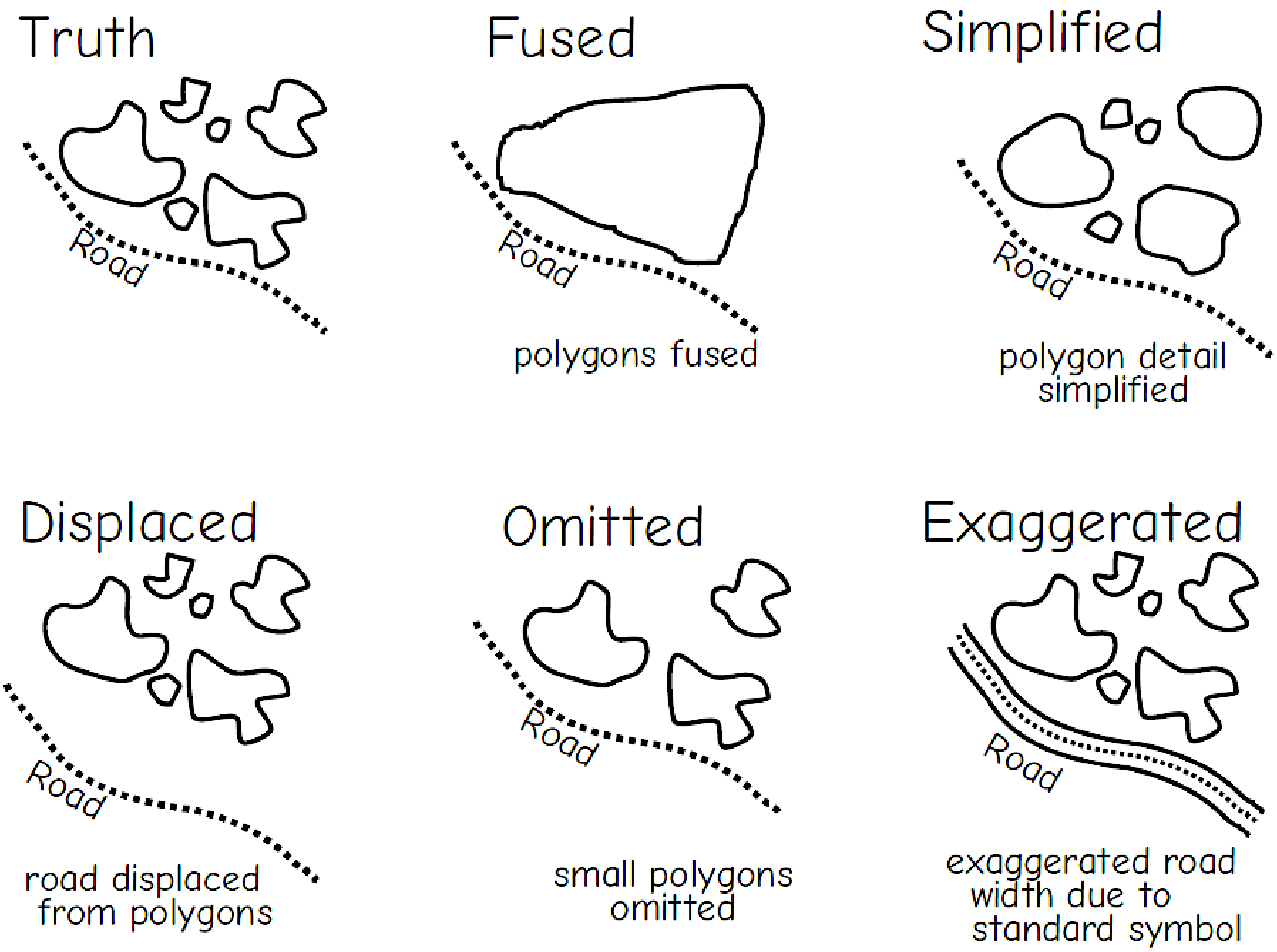



**Scale issue: ERROR!** The surface error caused by a 1-millimeter digitizing or map error will change as scale changes. Note the larger error at smaller scales.

- Resolution is the smallest spatial unit mapped.
- Accuracy is the difference between encoded and actual value.
- The surface error caused by a 1-millimeter digitizing or map error will change as scale changes. Note the larger error at smaller scales:

| Scale | Error (m) | Error (ft) |
|---|---|---|
| 1:24,000 | 24 | 79 |
| 1:50,000 | 50 | 164 |
| 1:62,500 | 63 | 205 |
| 1:100,000 | 100 | 328 |
| 1:250,000 | 250 | 820 |
| 1:1,000,000 | 1,000 | 3,281 |

**Cartographic Generalization (Simplification, Aggregation, Smoothing)**



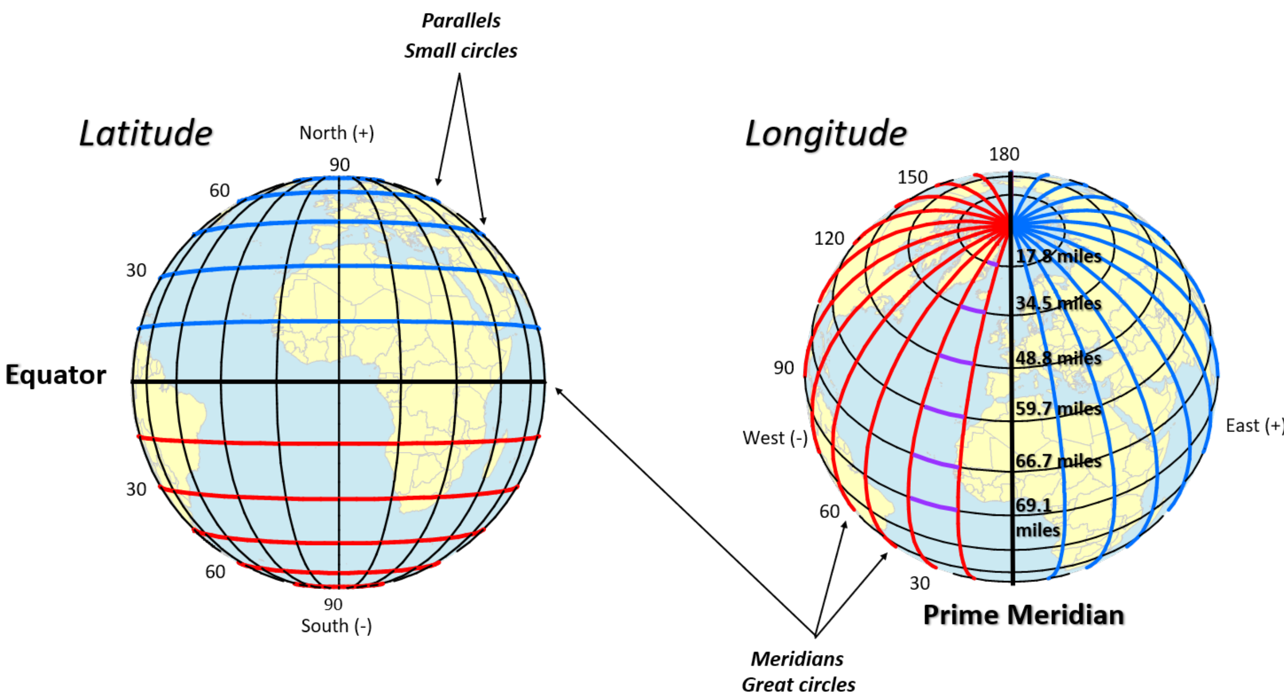

**Coordinate Systems**

Coordinate systems are frameworks that are used to define unique positions. For instance, in geometry we use x (horizontal) and y (vertical) coordinates to define points on a two-dimensional plane.

The coordinate system that is most commonly used to define locations on the three-dimensional earth is called the geographic coordinate system (GCS), and it is based on a sphere or spheroid.

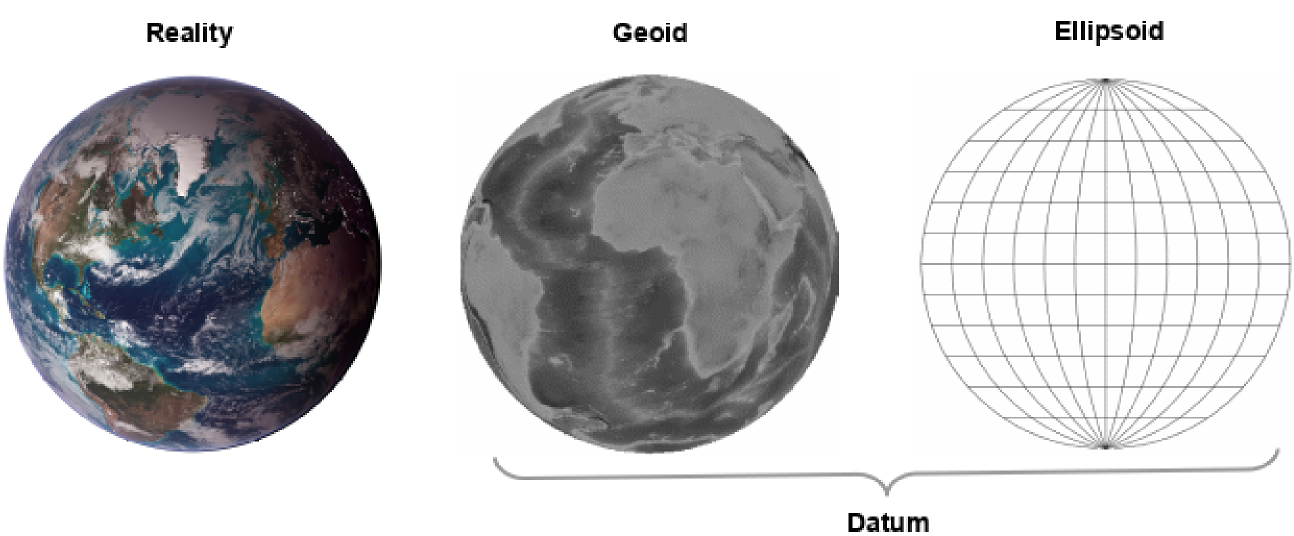
A spheroid (a.k.a. ellipsoid) is simply a sphere that is slightly wider than it is tall and approximates more closely the true shape of the earth.

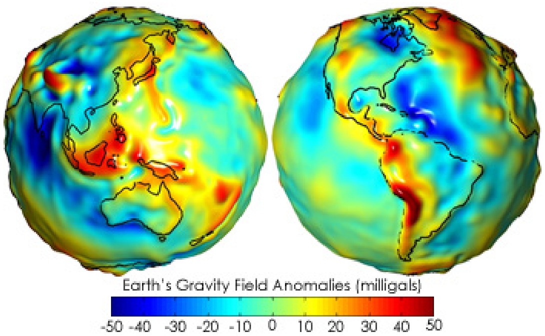
Spheres are commonly used as models of the earth for simplicity.

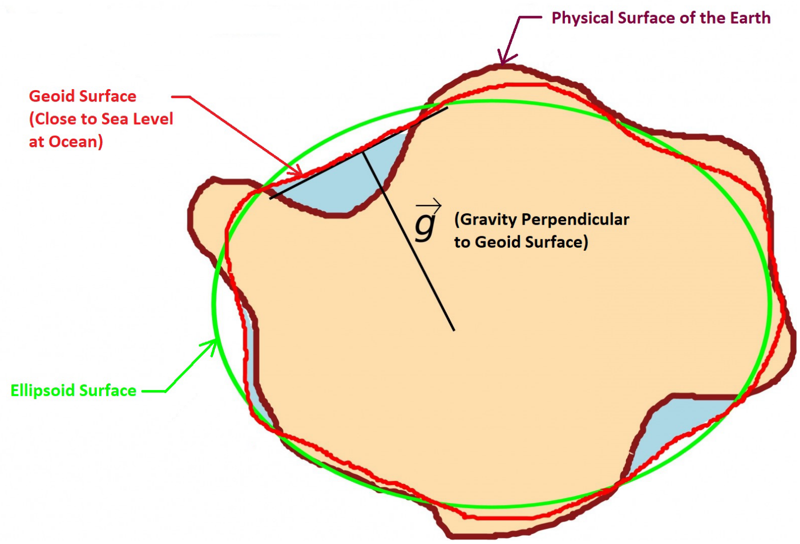


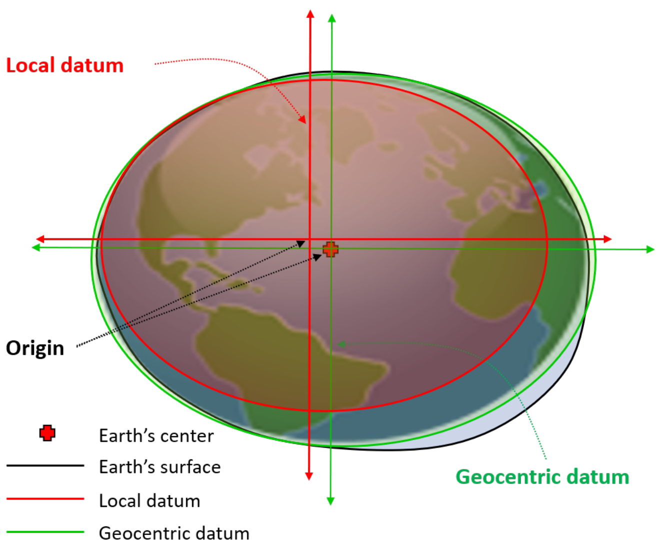

The unit of measure in the GCS is degrees, and locations are defined by their respective latitude and longitude within the GCS. Latitude is measured relative to the equator at zero degrees, with maxima of either ninety degrees north at the North Pole or ninety degrees south at the South Pole. Longitude is measured relative to the prime meridian at zero degrees, with maxima of 180 degrees west or 180 degrees east.

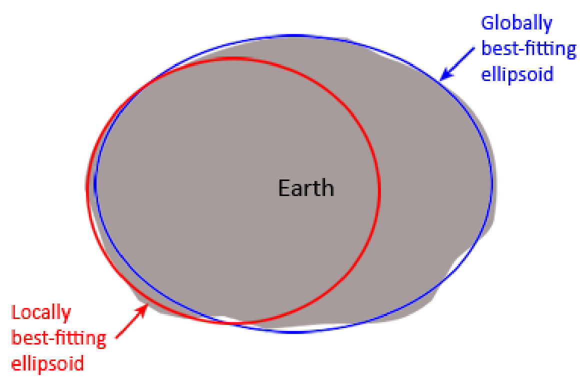
**Datums (Reference Frames)**: A geodetic datum is a mathematical model of the Earth that provides a fixed origin, scale, and orientation for a coordinate system, enabling all latitude, longitude, height, and map coordinates to be defined.





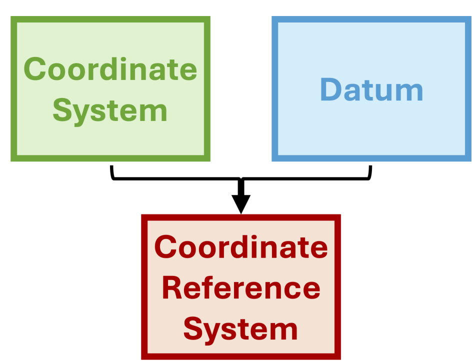



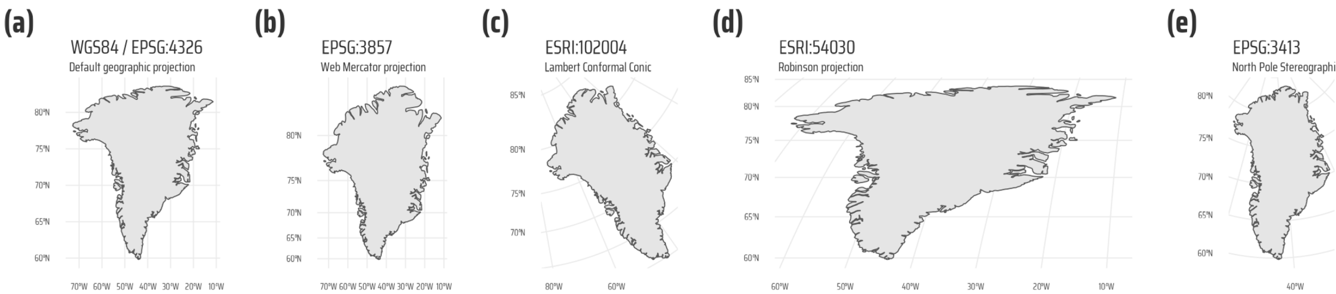

The two types of Datums are:

- Geocentric datum (e.g., WGS84): Centered at Earth's center of gravity, providing global consistency but less local accuracy.
- Local datum (e.g., NAD83): Adjusted for specific regions to better align with the Earth's surface, accounting for local geographic variations.
- Coordinate Reference Systems (CRSs) are essential for spatial data, defining how spatial elements correspond to the Earth's surface.



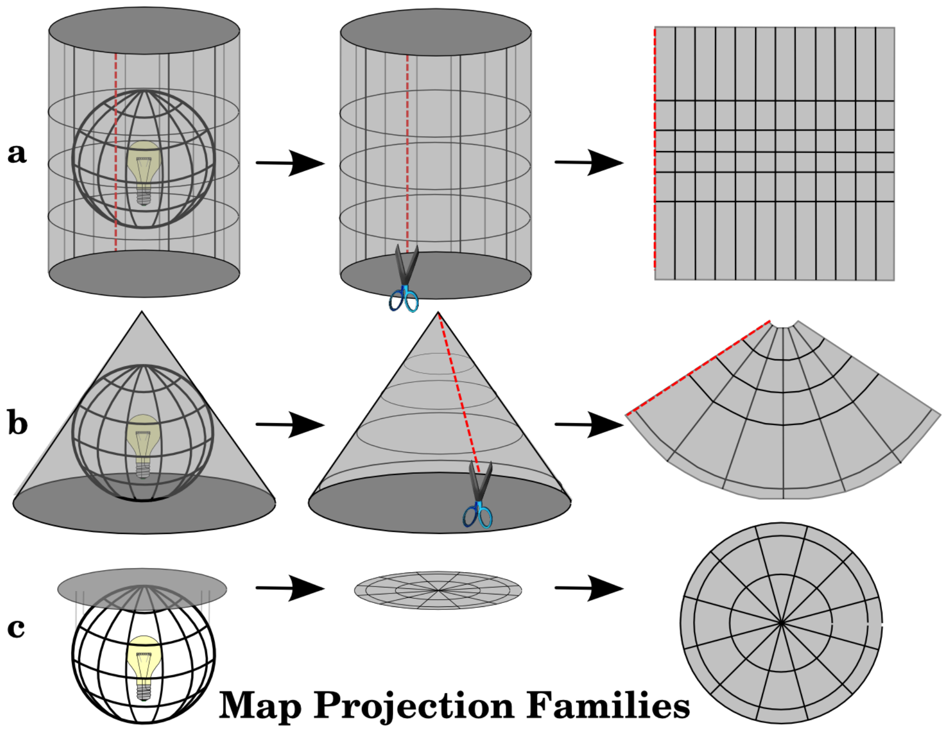
## Map Projections

Greenland is good for CRS projections examples:




**Map Projection Explorer:** → https://www.geo-projections.com

**Spatial Coordinate Reference Systems:** → https://spatialreference.org



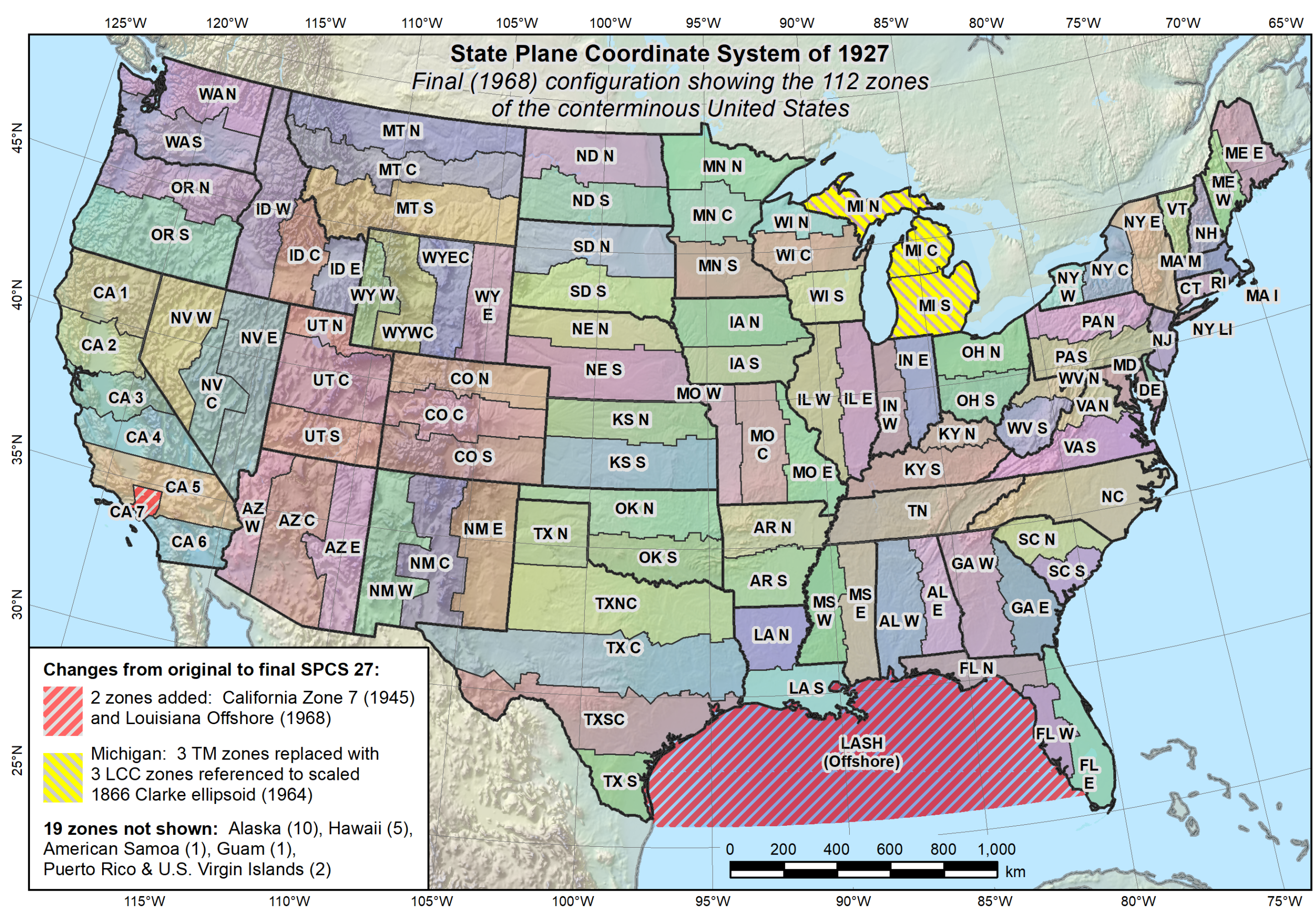

**Map Projection Families**

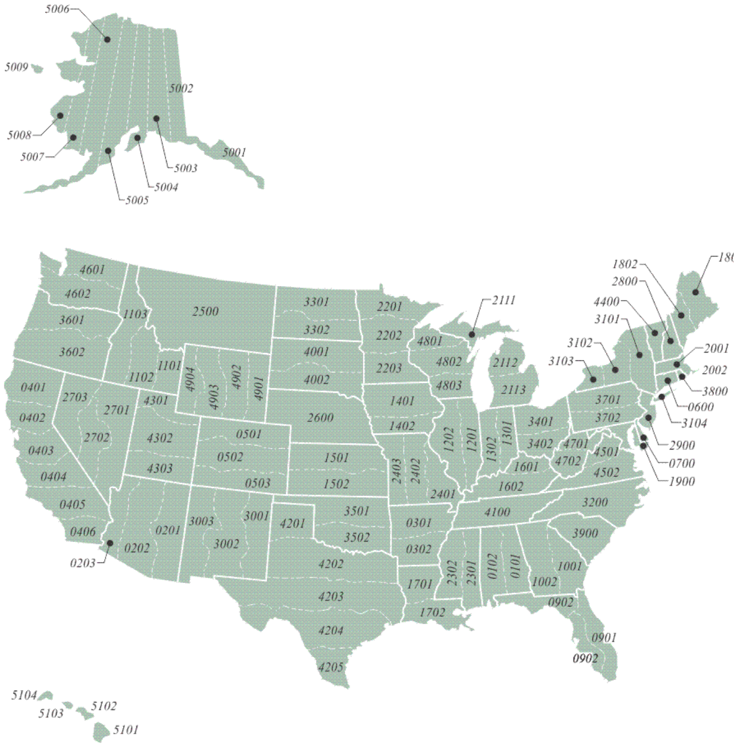
| **Type of Projection** | **Description** | **Properties Preserved** | **Best Used For** |
|---|---|---|---|
| Cylindrical (a) | Project the surface onto a cylinder. | Direction, shape | World maps |
| Conic (b) | Project the surface onto a cone. | Area, shape | Maps of mid-lat regions |
| Planar/Azimuthal (c) | Projects onto a flat surface at a point or line. | Distance, direction | Polar region maps |

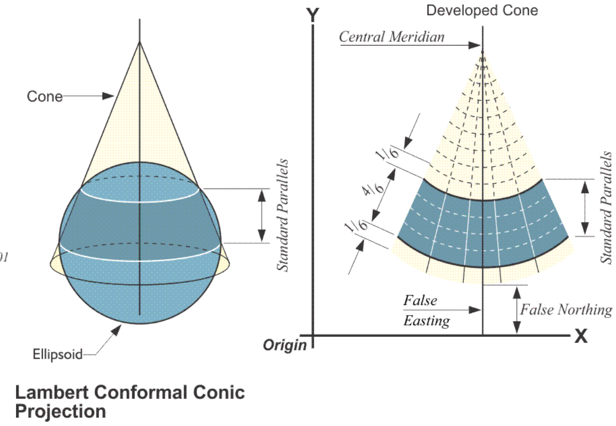
## Distortion of Distance, Area, Shape, and Direction

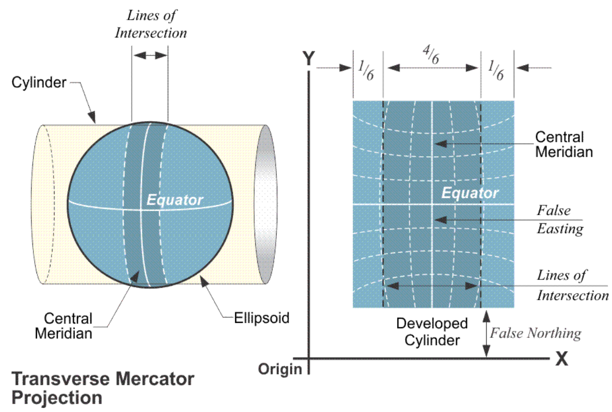
| Property | Definition | Projection Type That Preserves It |
|---|---|---|
| Area | The relative size of regions is maintained. | Equal-area projections (e.g., Albers) |
| Direction | Bearings from the center are accurate. | Azimuthal projections (e.g., Lambert) |
| Distance | Correct distances are preserved along specific lines or points. | Equidistant projections (e.g., Equirectangular) |
| Shape | Local angles and shapes are maintained, though areas are distorted. | Conformal projections (e.g., Mercator) |

The **State Plane Coordinate System** (SPCS) is a set of geographic zones or coordinate systems designed for specific regions of the United States. Each U.S. state contains one or more state plane zones, the boundaries of which usually follow county lines. The SPCS is widely used for geographic data by state and local governments.





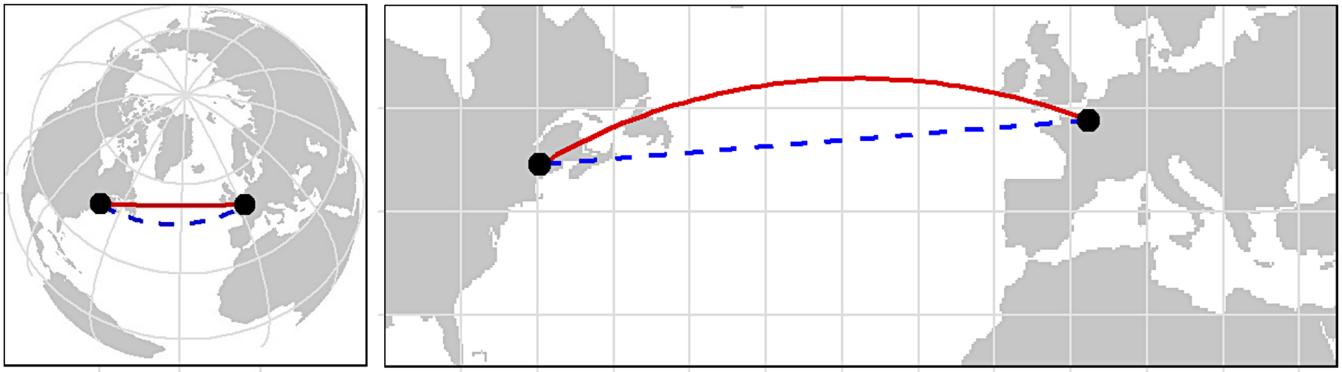



Texas is divided into five SPCS zones due to its large size. Each zone uses a Lambert Conformal Conic projection, which is ideal for east-west-oriented regions like Texas. Each zone is designed to minimize distortion and provide high accuracy for surveying, mapping, and engineering applications. The most common State Plane Coordinate Reference Systems (CRS) for Texas are:

- NAD83 Texas North (FIPS 4201)
- NAD83 Texas North Central (FIPS 4202)
- NAD83 Texas Central (FIPS 4203)
- NAD83 Texas South Central (FIPS 4204)
- NAD83 Texas South (FIPS 4205)

Treating a human head the way a globe is treated by the Mercator projection:



**Ethical Implications of Misrepresentation**

The fundamental issue is that maps and spatial analyses carry authority and influence how people understand the world, making accuracy and transparency essential ethical obligations.

- **Undermining Decision-Making:** When maps or spatial data are inaccurate or deliberately distorted, they can lead to poor decisions, misallocated resources, or inappropriate planning. Decision-makers who rely on flawed GIS outputs may implement solutions that don't address real problems or that create new ones.
- **Perpetuating Inequality:** Misrepresented data can reinforce existing inequalities. For example, understating pollution levels in low-income neighborhoods, gerrymandering electoral districts, or misrepresenting crime stats can perpetuate discrimination and deny communities the resources or protections they need.
- **Eroding Public Trust:** When people discover that maps or spatial analyses have been manipulated or are inaccurate, it damages trust in professionals, institutions, and scientific evidence more broadly. This can make it harder to communicate legitimate findings in the future.
- **Harm to Affected Communities:** Misrepresentation can directly harm people by obscuring environmental hazards, misidentifying vulnerabilities, or incorrectly representing land boundaries. This can lead to displacement and losses.
- **Professional Responsibility:** Obligation to represent spatial data accurately and transparently. Misrepresentation violates professional codes of ethics and can constitute scientific misconduct, potentially leading to legal consequences.
- **Loss of Objectivity:** Even unintentional misrepresentation (through poor methodology, biased data collection, or inappropriate visualization choices) can compromise the objectivity that GIS analysis should provide, turning what should be a neutral tool into an instrument of advocacy or manipulation.

# 6 Data Types

## Software

| Attribute | QGIS (Desktop GIS) | R/Python |
|---|---|---|
| Disciplines | Geography | Computing, Statistics |
| Focus | Graphical User Interface | Command line |
| Reproducibility | Minimal | Maximal |

## QGIS

Prepare QGIS for next week:

- Download QGIS https://qgis.org/download/
  You may use either QGIS 3.44.6 Solothurn or QGIS 3.40.14 Bratislava.
  - QGIS LTR (Long Term Release): Optimized for maximum stability and reliability, receives bug fixes and security updates only (no new features are introduced during its life cycle).
  - QGIS current (Latest Release): Includes the newest features, tools, and improvements. More frequent updates, with rapid evolution of functionality.
- Documentation for QGIS 3.40: https://docs.qgis.org/3.40/en/docs/

## R

Download and Install R and RStudio:

- R: https://cran.rstudio.com
- RStudio (IDE): https://posit.co/download/rstudio-desktop/

Recommended documentation:

- R for Data Science: https://r4ds.hadley.nz
- Hands-On Programming with R: https://rstudio-education.github.io/hopr/
  - Installing R, RStudio: https://rstudio-education.github.io/hopr/starting.html

### Install packages/libraries

```r
install.packages(tidyverse)
install.packages(sf)
```

Check out vignettes: https://cran.r-project.org/web/packages/sf/vignettes/sf1.html

## Vector Data Models

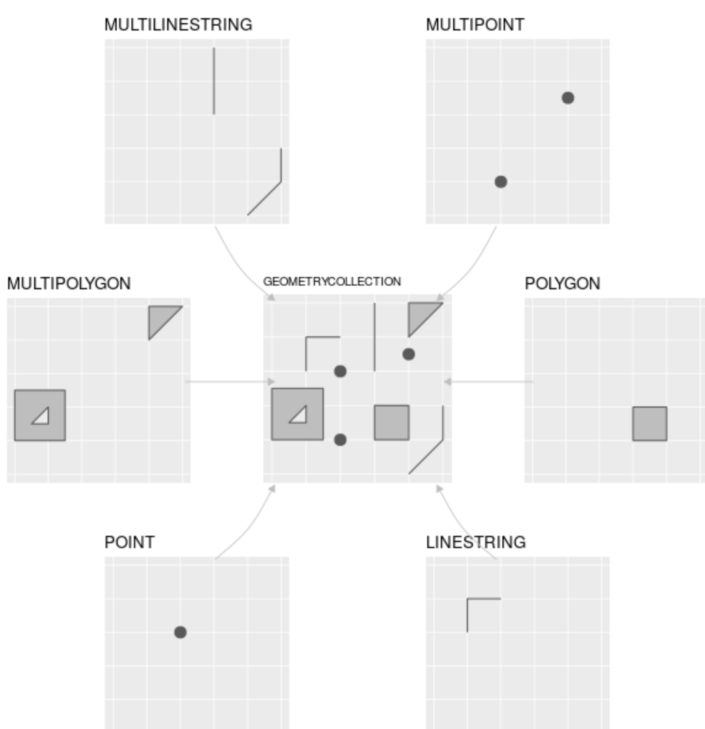
Take care when using the word 'vector', as it can have two meanings: (*i*) geographic vector data and (*ii*) the vector class in R. The former is a data model, the latter is an R class just like `data.frame` and matrix.

There is a link between the two: the spatial coordinates which are at the heart of the geographic vector data model can be represented in `R` using vector objects. The geographic vector data model is based on points located within a coordinate reference system (CRS).

Points can represent self-standing features (e.g., the location of a bus stop) or they can be linked together to form more complex geometries such as lines and polygons. Most point geometries contain only 2 dimensions (less common geometries contain an additional z-value, typically representing height above sea level).

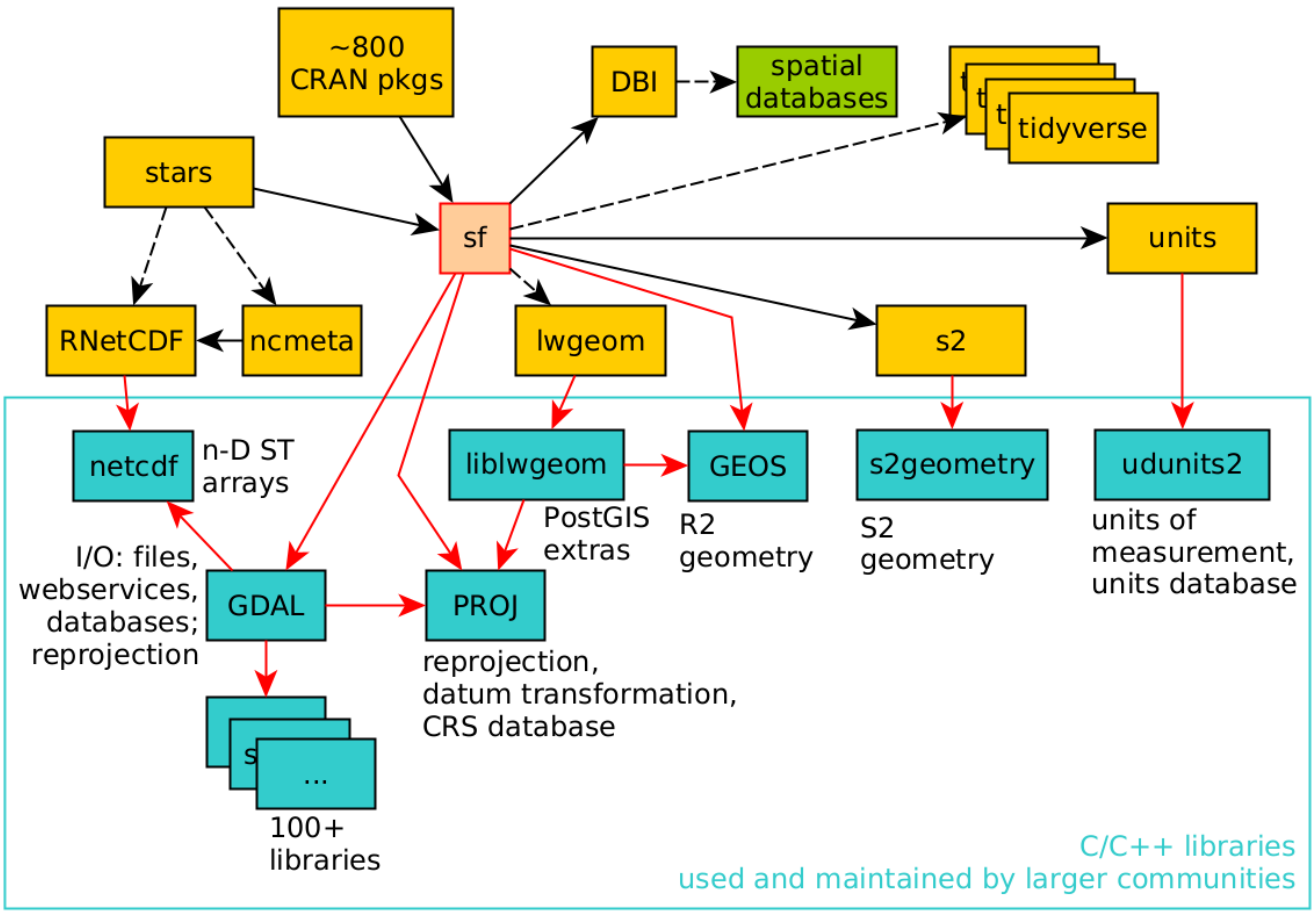
## Introduction to simple features (`sf`)

simple features (`sf`) is an open standard developed and endorsed by the Open Geospatial Consortium. `sf` is a hierarchical data model that represents a wide range of geometry types. Of 18 geometry types supported by the specification, only seven are used in the vast majority of geographic research.

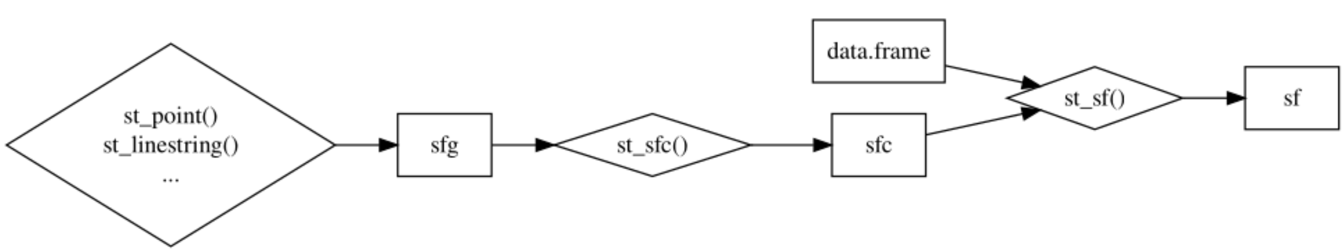


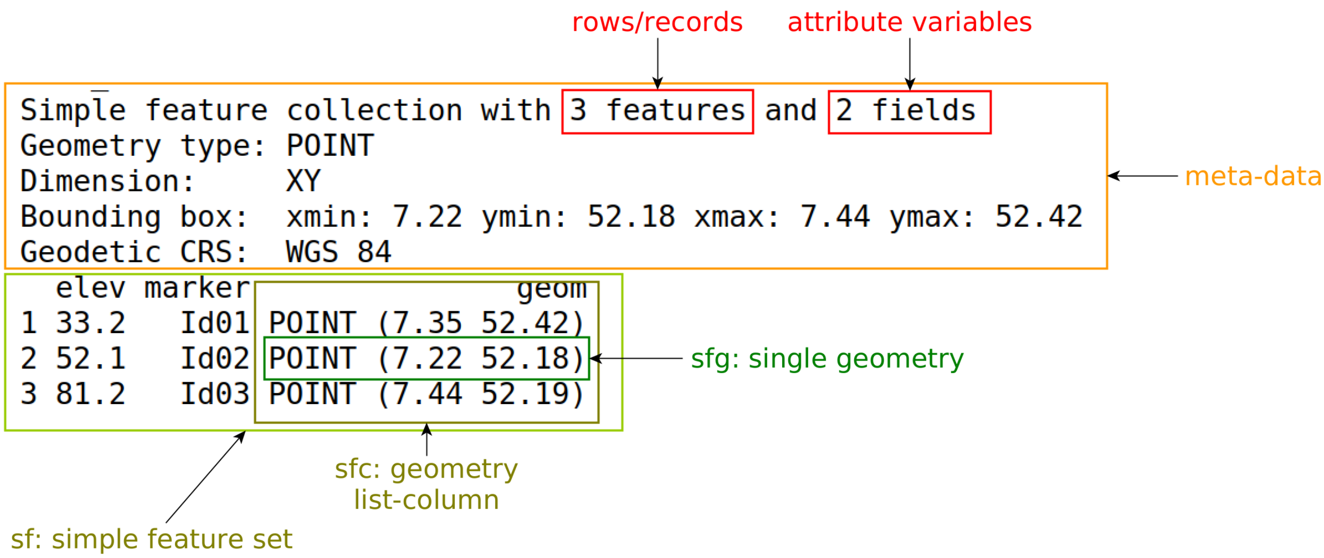
The **sf** package provides classes for geographic vector data and a consistent command line interface to important low-level libraries for geocomputation:

- **GDAL**: For reading, writing, manipulating a wide range of geo-data formats.
- **PROJ**: A powerful library for coordinate system transformations.
- **GEOS**: A planar geometry engine for operations such as calculating buffers and centroids on data with a projected CRS.
- **S2**: A spherical geometry engine written in C++ developed by Google.



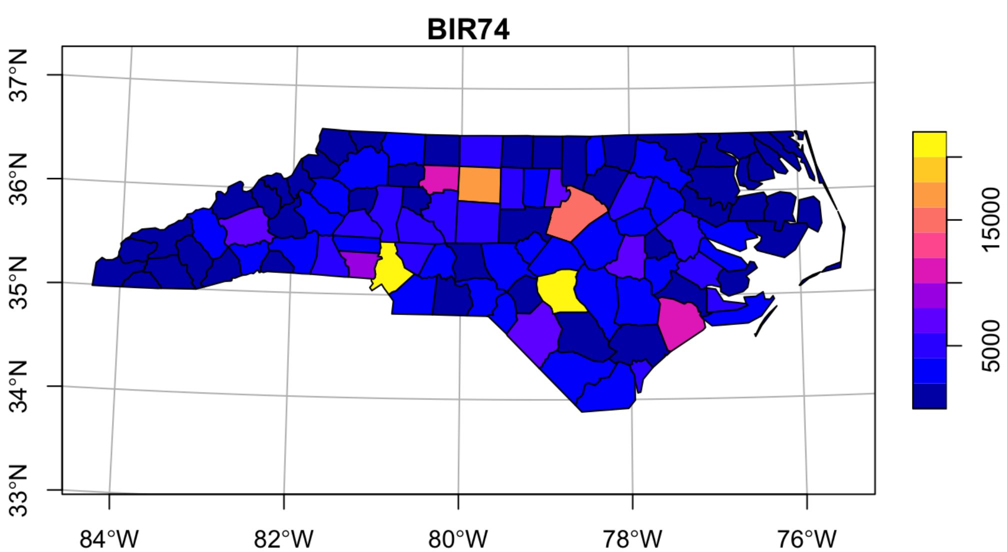

**sf** consist of two main parts: geometries and non-geographic attributes.






## A First Map

```r
library(tidyverse)
library(sf)
nc <- system.file("gpkg/nc.gpkg", package="sf") |> read_sf()
# system.file("gpkg/nc.gpkg", package="sf") |> read_sf() -> nc
glimpse(nc)
nc.32119 <- st_transform(nc, 'EPSG:32119')
nc.32119 |> select(BIR74) |> plot(graticule = TRUE, axes = TRUE)
```

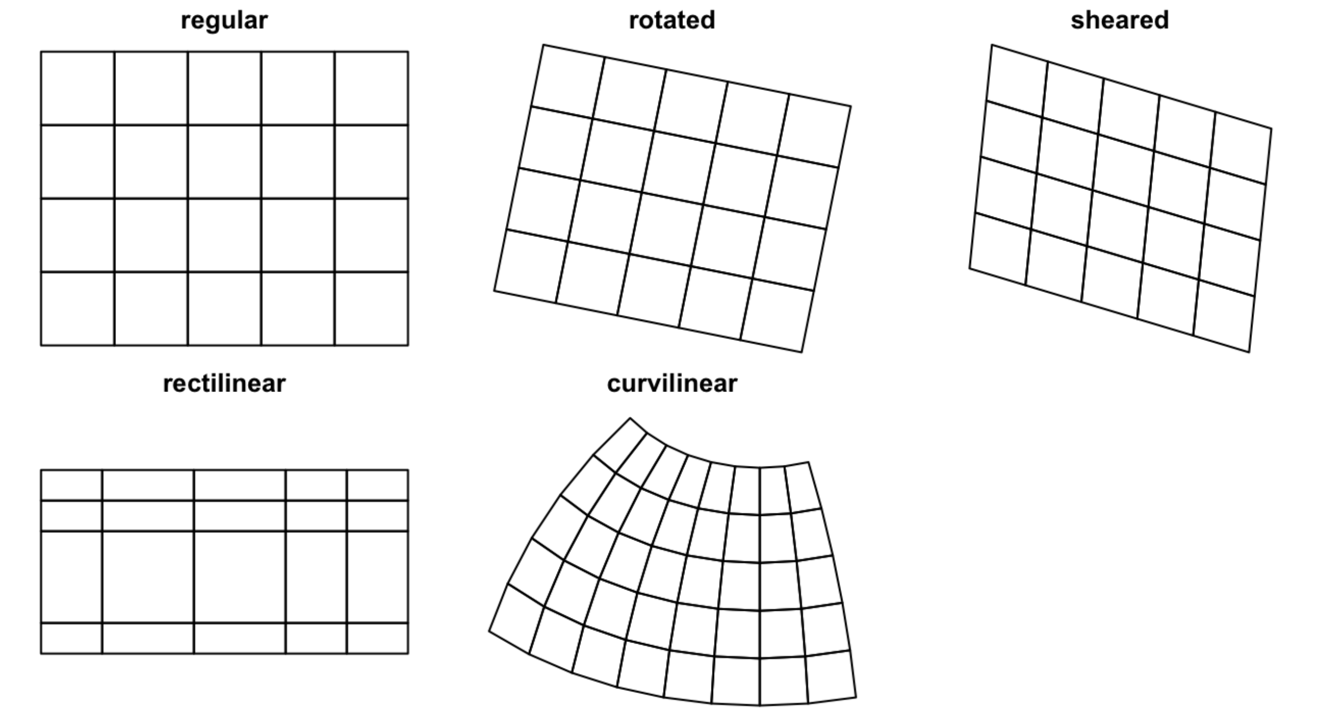


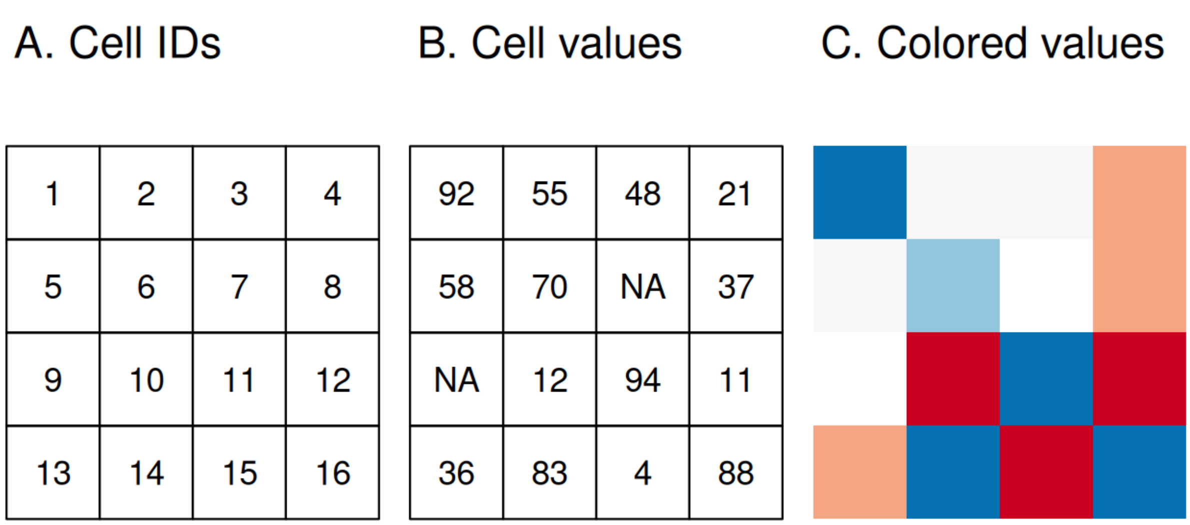

Try a more interactive map. Soon we will explore `tmap` further.

```r
library(tmap)
tmap_mode("view")
qtm(nc.32119, fill = "BIR74", fill_alpha = .5)
```

## Raster Data Models

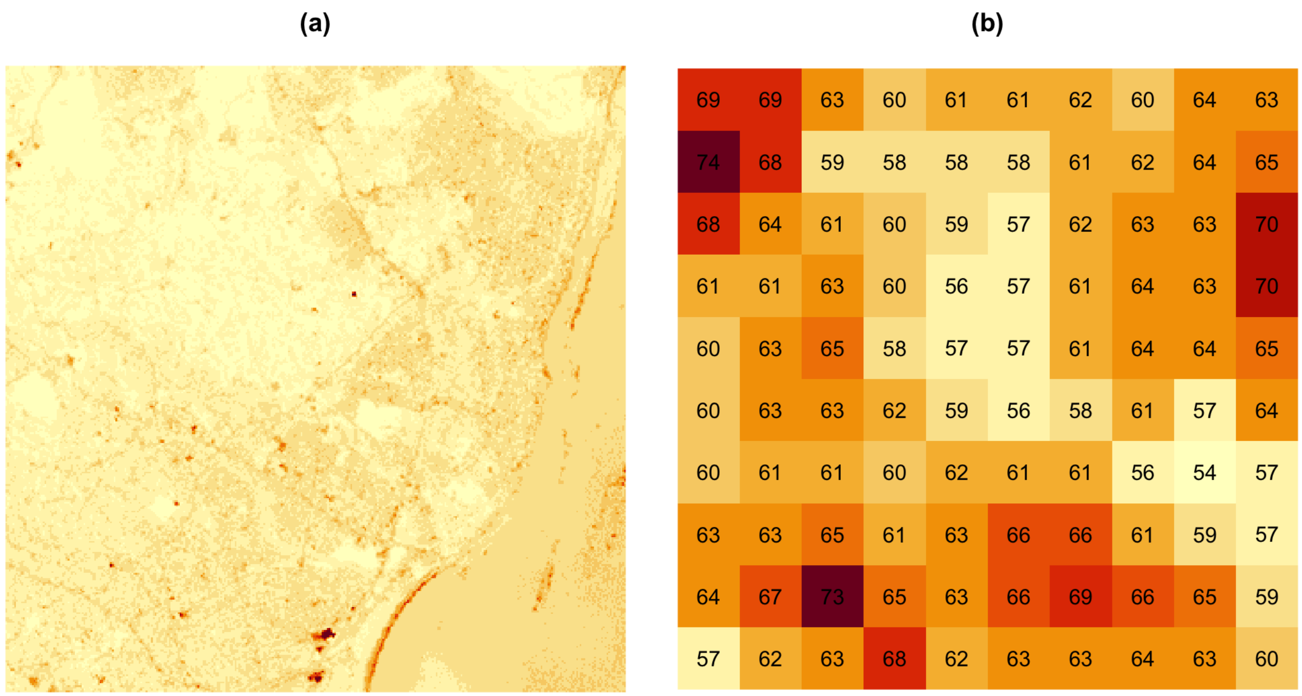
- The spatial raster data model represents the world with the continuous grid of cells (often also called pixels).
- This data model often refers to so-called regular grids, in which each cell has the same, constant size – and we will focus on the regular grids only.
- However, several other types of grids exist, including rotated, sheared, rectilinear, and curvilinear grids.



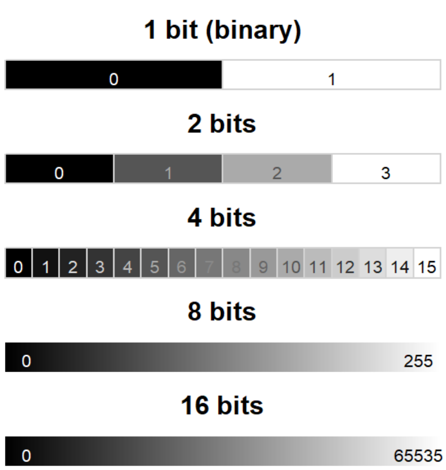



```r
install.packages(stars)
```

```r
library(stars)
par(mfrow = c(1, 2))
tif <- system.file("tif/L7_ETMs.tif", package = "stars")
x <- read_stars(tif)[,,,1]
image(x, main = "(a)")
image(x[,1:10,1:10], text_values = TRUE, border = 'grey', main = "(b)")
```




In contrast to vector data, the cell of one raster layer can only hold a single value. The value might be continuous or categorical.



### When to Use It

Advantages of the raster models

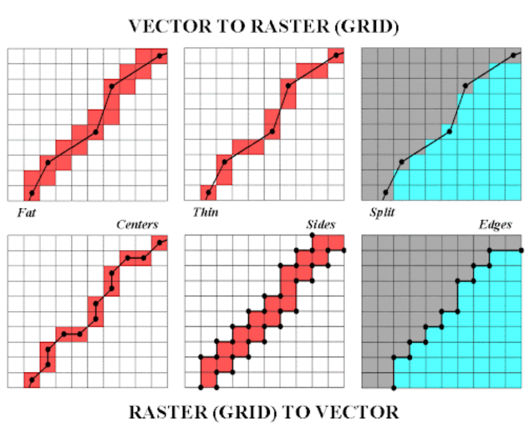
- **Simple Data Structure:** Grid-based, easy to understand, process cell-by-cell.
- **Efficient Analysis:** Fast and easy overlay analysis (map algebra) for complex calculations.
- **Ideal for Continuous Data:** Naturally represents phenomena like elevation, temperature, or satellite imagery.
- **Cost-Effective:** Often uses inexpensive, widely available technology.
- **Natural for Remote Sensing:** Good for scanned or remotely sensed data.

Disadvantages of the raster models

- **Large File Sizes:** High resolution means more cells, leading to huge files.
- **Pixelated Appearance:** Zooming in reveals blocky pixels, reducing aesthetic quality for precise features.
- **Poor for Linear Features:** Difficult to represent precise points, lines, and boundaries (e.g., roads, property lines).
- **Limited Attribute Data:** Each cell holds one value, limiting complex feature attributes compared to vector.
- **Geometric Transformation Issues:** Reprojecting maps can cause problems and data loss.

Use raster for modeling surfaces (elevation, slope, temperature), imagery analysis, and quantitative studies.

Vector models are better for representing discrete features like roads, buildings, and property boundaries.




Later we will learn how to rasterize and vectorize, but we will go further by examining their implications for accuracy, storage, and computational efficiency. We will also explore common data quality issues, including topology errors, gaps, overlaps, slivers, and related inconsistencies.

# 7 Data Sources

## Finding Data

### Primary Data Capture

- Primary data capture is a direct data acquisition methodology that is usually associated with some type of in-the-field effort.
- In the case of vector data, directly captured data commonly comes from a global positioning system (GPS) or other types of surveying equipment such as a total station (GPS Unit (left) and Total Station (right)).

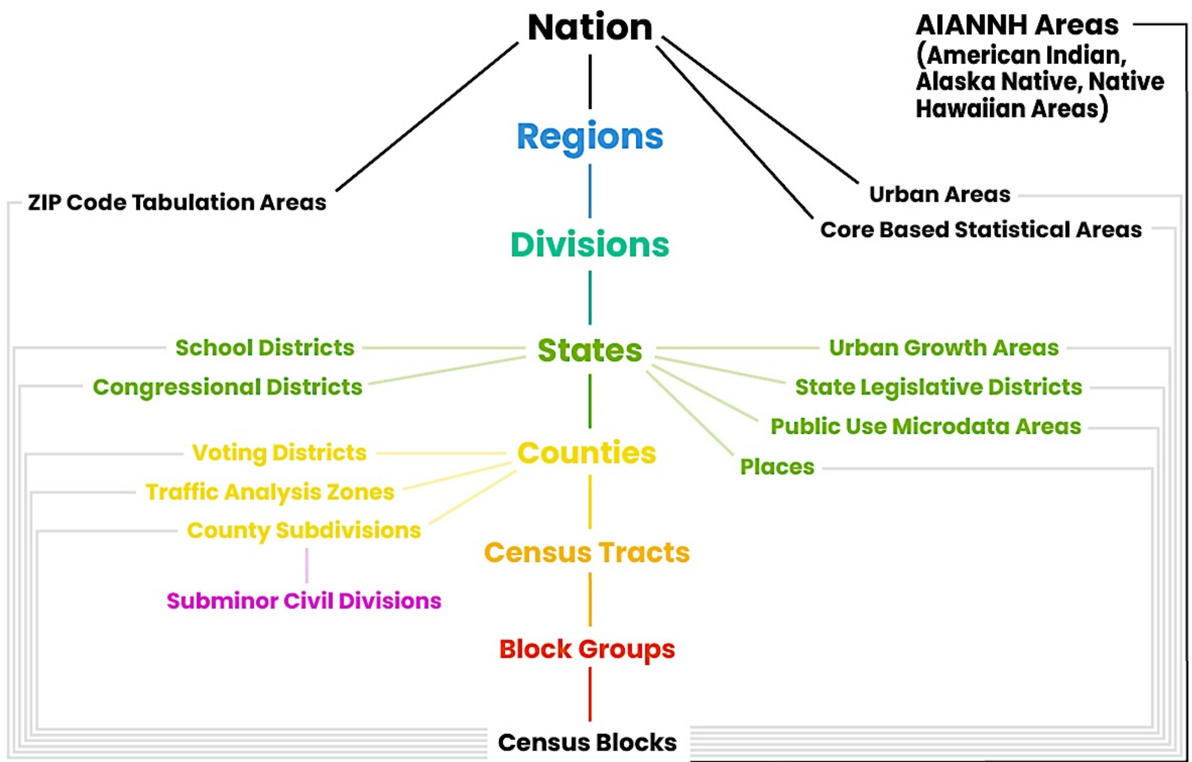
- Total stations are specialized, primary data capture instruments that combine a theodolite (or transit), which measures horizontal and vertical angles, with a tool to measure the slope distance from the unit to an observed point. Use of a total station allows field crews to quickly and accurately derive the topography for a particular landscape.

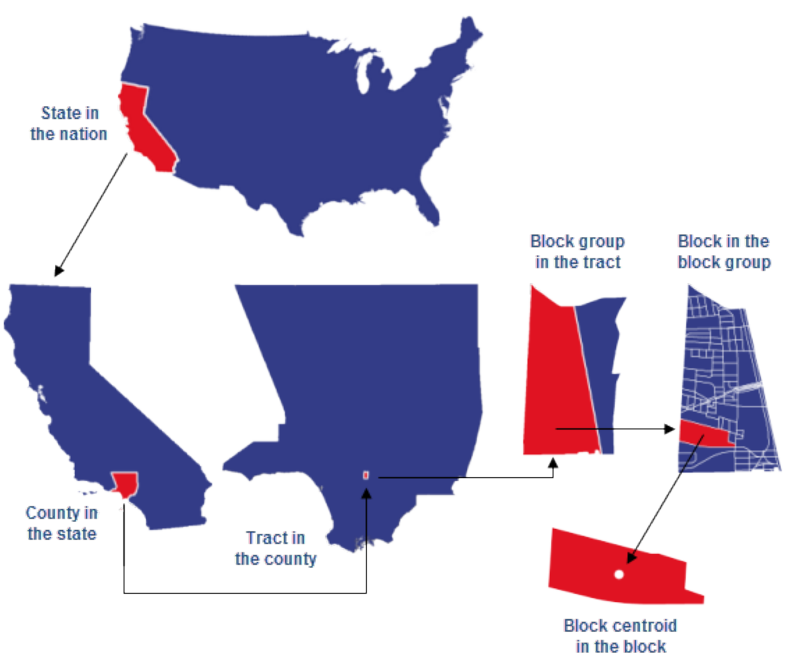
### Secondary Data Capture

- Secondary data capture is an indirect methodology that utilizes the vast amount of existing geospatial data available in both digital and hard-copy formats.
- Prior to initiating any GIS effort, it is always wise to mine online resources for existing GIS data that may fulfill your mapping needs without the potentially intensive step of creating the data from scratch.
- Such GIS data are available from a variety of sources including international agencies (CGIAR, CIESIN, United Nations, World Bank, etc.); federal governments (USGS, USDA, NOAA, USFWS, NASA, EPA, US Census, etc.); state

governments (CDFG, Teale Data Center, INGIS, MARIS, NH GIS Resources, etc.); local governments (SANDAG, RCLIS, etc.); university websites (UCLA, Duke, Stanford, University of Chicago, Indiana Spatial Data Portal, etc.); and commercial websites (ESRI, GeoEye, Geocomm, etc.).

**Standard Hierarchy of Census Geographic Entities**



### tigris

https://cran.r-project.org/web/packages/tigris/refman/tigris.html

tigris is an R package that allows users to directly download and use TIGER/Line shapefiles from the US Census Bureau. https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html

```
install.packages(tigris)
```

```
library(tidyverse)
library(tigris)

us <- states(cb = TRUE, resolution = "20m", year = 2022)

glimpse(us)

us |>
  select(NAME) |>
  plot(graticule = TRUE, axes = TRUE)

us_shifted <- shift_geometry(us)

us_shifted |>
  select(NAME) |>
  plot(graticule = TRUE, axes = TRUE)

us_shifted |>
  select(NAME) |>
  filter(NAME == "Texas") |>
  plot(graticule = TRUE, axes = TRUE)

texas <- counties(state = "TX", cb = FALSE, year = 2022)

brazos <- tracts(county = "Brazos", state = "TX", cb = FALSE, year = 2022)
```

### tidycensus

For next week:

```
install.packages(tidycensus)
```

**tidycensus** is an R package that allows users to interface with a select number of the US Census Bureau's data APIs and return tidyverse-ready data frames, optionally with simple feature geometry included.

https://walker-data.com/tidycensus/

Request a US Census Data API Key: https://api.census.gov/data/key_signup.html

Reference: https://walker-data.com/census-r/index.html

## Hands-on

**Let us work through a few practice to prepare for the homework:**

```
library(dplyr)
library(sf)
sf_use_s2(FALSE) # spherical geometry

library(tmap)
tmap_mode("plot") # static maps
tmap_options(frame = FALSE)

library(units)

# Define the grid extent (4 wide, 4 tall)
xmin <- 0
xmax <- 40
ymin <- 0
ymax <- 40

# Create the grid
grid <-
  st_make_grid(
    st_bbox(c(xmin = xmin, ymin = ymin, xmax = xmax, ymax = ymax)),
    cellsize = 10, # 10 units cells
    square = TRUE,
    what = "polygons"
  )
print(grid, n=20)

plot(grid)
```

```
plot(grid[1], col = "red", add = TRUE)
plot(grid[2], col = "green", add = TRUE)
plot(grid[3], col = "blue", add = TRUE)
plot(grid[6], col = "yellow", add = TRUE)
plot(grid[16], col = "brown", add = TRUE)
axis(1, at = seq(0, 40, 10))
axis(2, at = seq(0, 40, 10))
grid()

# Convert to sf object
grid_sf      <- st_sf(geometry = grid)
grid_sf$id   <- 1:nrow(grid_sf)  # Add IDs
grid_sf$area <- st_area(grid_sf) # Area
grid_sf
plot(grid_sf["id"], border = "red", col = "white", main = "")

tm_shape(grid_sf) +
  tm_borders(col = "red", lwd = 1)

st_crs(grid_sf)

st_crs(4326)
st_set_crs(grid_sf, 4326)

st_crs(32610) # example: UTM Zone 10N
st_crs(32614) # example: UTM Zone 14N
st_set_crs(grid_sf, "+proj=cart +ellps=sphere") # sphere
st_set_crs(grid_sf, "+proj=cart +ellps=WGS84") # ellipse

st_crs("+proj=utm +zone=33 +datum=WGS84 +units=m +no_defs")
st_set_crs(grid_sf, "+proj=utm +zone=33 +datum=WGS84 +units=m +no_defs")

grid_sf_projected <- st_set_crs(grid_sf, 4326)

# These coordinates are longitude/latitude on the WGS-84 Earth model.
# Technically: Longitude: -180° to +180°
#              Latitude: -90° to +90°
# EPSG code: 4326

# | Task                              | CRS you should use                      |
# | --------------------------------- | --------------------------------------- |
```

```
# | Mapping / web display          | WGS84 (EPSG:4326)                 |
# | Distance, area, buffers, grids | Projected CRS (UTM, Albers, etc.)     |
# | Texas                          | EPSG:3083 (Texas Albers) or UTM 14/15 |

tm_shape(grid_sf_projected) +
  tm_borders(col = "red", lwd = 1) +
  tm_crs(crs = "+proj=robin")

data(World)

st_crs(World)

tm_shape(World) +
  tm_fill(fill = "grey") +
  tm_borders(col = "blue", lwd = .3) +
  tm_shape(grid_sf_projected) +
  tm_borders(col = "red", lwd = 2) +
  tm_scalebar(position = c("left", "bottom"), text.size = .9) +
  tm_crs(crs = "+proj=robin") +
  # tm_crs(crs = 4326) +
  # tm_crs(crs = 32610) +
  # tm_crs(crs = 32614) +
  tm_graticules(col = "gray70", lwd = 0.8, lty = "dotted")

grid_sf_projected$area_4326 <- st_area(grid_sf_projected)
grid_sf_projected

grid_sf_projected$area_4326 <- set_units(grid_sf_projected$area_4326, km^2)
grid_sf_projected

grid_sf_projected$area_4326 <- set_units(grid_sf_projected$area_4326, mi^2)
grid_sf_projected

grid_sf_projected$area_32610 <- set_units(st_area(st_transform(grid_sf_projected, crs = 32610)), mi^2)
grid_sf_projected |> st_drop_geometry()

# Four projection systems -----------------
albers_crs   <- "+proj=aea  +lat_0=23         +lat_1=29.5       +lat_2=45.5 +lon_0=-96   +datum=WGS84 +units=m"
```

```r
equidist_crs <- "+proj=eqdc +lat_0=31.1666667 +lat_1=27.4166667 +lat_2=34.9166667  +lon_0=-100  +datum=WGS84 +units=m"
lcc_crs      <- "+proj=lcc  +lat_0=31.1666667 +lat_1=27.4166667 +lat_2=34.9166667  +lon_0=-100  +datum=WGS84 +units=m"
aeqd_crs     <- "+proj=aeqd +lat_0=30.7 +lon_0=-96.3 +datum=WGS84 +units=m"

grid_sf_projected$area_albers <- set_units(st_area(st_transform(grid_sf_projected, crs = albers_crs)), mi^2)
grid_sf_projected$area_equidist <- set_units(st_area(st_transform(grid_sf_projected, crs = equidist_crs)), mi^2)
grid_sf_projected$area_lcc <- set_units(st_area(st_transform(grid_sf_projected, crs = lcc_crs)), mi^2)
grid_sf_projected$area_aeqd <- set_units(st_area(st_transform(grid_sf_projected, crs = aeqd_crs)), mi^2)
grid_sf_projected |> st_drop_geometry()

grid_sf_projected$area_true <- set_units(st_area(st_transform(grid_sf_projected, 102022)), mi^2)

africa_aea <- "+proj=aea +lat_1=20 +lat_2=-23 +lat_0=0 +lon_0=25 +datum=WGS84 +units=m +no_defs"
# Error: crs not found

grid_sf_projected$area_true <- set_units(st_area(st_transform(grid_sf_projected, africa_aea)), mi^2)
grid_sf_projected |> st_drop_geometry()

tm_shape(World) +
  tm_fill(fill = "grey") +
  tm_borders(col = "blue", lwd = .3) +
  tm_shape(grid_sf_projected) +
  tm_borders(col = "red", lwd = 2) +
  tm_scalebar(position = c("left", "bottom"), text.size = .9) +
  tm_crs(crs = africa_aea) +
  tm_graticules(col = "gray70", lwd = 0.8, lty = "dotted")

# Census ----------------------------------
library(tigris)
us      <- states(cb = FALSE, year = 2022) # %>% filter(NAME == "Texas")
```

```
texas  <- counties(state = "TX", cb = FALSE, year = 2022) # %>% filter(NAME ==
↪  "Texas")
brazos <- tracts(county = "Brazos", state = "TX", cb = FALSE, year = 2022)
st_crs(brazos)
# 4269

brazos$area_calc <- st_area(brazos)
brazos$diff <- brazos$ALAND + brazos$AWATER - as.numeric(brazos$area_calc)
sum(brazos$diff)
# -71.27524

brazos$area_4203 <- st_area(st_transform(brazos, 4203))
brazos$diff <- brazos$ALAND + brazos$AWATER - as.numeric(brazos$area_4203)
sum(brazos$diff)
# -10994.88

# NAD83 / Texas Central (FIPS 4203) --> EPSG: 32139
# Purpose  Engineering / cadastral mapping
# Preserves  shape and angles
# Does not preserve  area: Lambert Conformal Conic preserves angles, not area.
# Area bias across Brazos County runs about 0.15-0.30% - small, but real and
↪  systematic.
# For scientific, legal, or statistical area work, this is considered
↪  unacceptable.
brazos$area_32139 <- st_area(st_transform(brazos, 32139))
brazos$diff <- brazos$ALAND + brazos$AWATER - as.numeric(brazos$area_32139)
sum(brazos$diff)
# 298551.4

# sum(brazos$diff) / sum(brazos$ALAND + brazos$AWATER) * 100
# 0.0195

tm_shape(brazos) +
  tm_borders(col = "red", lwd = 2) +
  tm_crs(crs = 32139)

brazos$area_3083 <- st_area(st_transform(brazos, 3083))
brazos$diff <- brazos$ALAND + brazos$AWATER - as.numeric(brazos$area_3083)
sum(brazos$diff)
# -71.90936

```

```
brazos |> st_drop_geometry()

tmap_arrange(tm_shape(brazos) +
               tm_borders(col = "red", lwd = 2) +
               tm_crs(crs = 4269) +
               tm_graticules(col = "gray70", lwd = 0.8, lty = "dotted") +
               tm_title("Original"),

             tm_shape(brazos) +
               tm_borders(col = "red", lwd = 2) +
               tm_crs(crs = 32139) +
               tm_graticules(col = "gray70", lwd = 0.8, lty = "dotted") +
               tm_title("32139 for Shape"),

             tm_shape(brazos) +
               tm_borders(col = "red", lwd = 2) +
               tm_crs(crs = 3083) +
               tm_graticules(col = "gray70", lwd = 0.8, lty = "dotted") +
               tm_title("3083 for Area"),

             ncol = 3, sync = TRUE)

brazos <- texas %>% filter(NAME == "Brazos")

# two versions with their coordinate systems "baked in"
brazos_1 <- brazos["geometry"] %>% st_transform(4269) %>% st_set_crs(NA)
brazos_2 <- brazos["geometry"] %>% st_transform(32139) %>% st_set_crs(NA)

plot(brazos_1, border = "red", col = NA, main = "", axes = TRUE)
plot(brazos_2, border = "blue", col = NA, main = "", axes = TRUE)

normalize_geom <- function(g) {
  bb <- st_bbox(g)
  sx <- 1 / (bb["xmax"] - bb["xmin"])
  sy <- 1 / (bb["ymax"] - bb["ymin"])

  g2 <- g
  g2$geometry <- lapply(st_geometry(g), function(geom) {
    coords <- st_coordinates(geom)
    coords[,1] <- (coords[,1] - bb["xmin"]) * sx
    coords[,2] <- (coords[,2] - bb["ymin"]) * sy
```

```
    st_polygon(list(coords[,1:2]))
  })
  st_sf(geometry = st_sfc(g2$geometry))
}

b1n <- normalize_geom(brazos_1)
b2n <- normalize_geom(brazos_2)

plot(b1n, border="red", col=NA)
plot(b2n, border="blue", col=NA, add=TRUE)
```

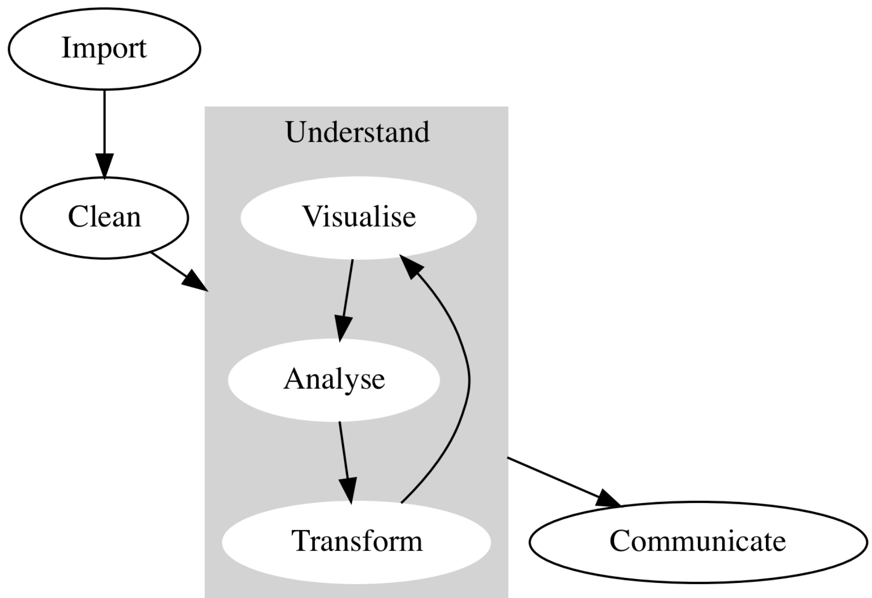
## Questions to Practice

The following questions are for practice and self-assessment. These questions will not be graded, you do not need to submit solutions. However work through them carefully as questions of similar format and difficulty may appear on the midterm exam. Discuss with classmates or during office hours. You should be able to clearly justify and explain your reasoning without notes.

1. What is an ellipsoid? How does an ellipse differ from a sphere?
2. How does magnetic north differ from the geographic North Pole?
3. Define a datum. Briefly describe how datums are developed.
4. Why are there multiple datums even for the same place on Earth?
5. Describe the State Plane coordinate system. What type of projections are used in a State Plane coordinate system?
6. Which is the larger-scale map:
    a) 1:5,000 or 1:15,000
    b) 1:5,286 or 1 inch to a mile
    c) 1:1.000.000 or 1 cm to 1 km
    d) 1:50.000 or 0.00025
    e) 5:1 or 1:1?
7. Describe three types of generalization.
8. What is the “common feature problem” when digitizing and how might it be overcome?
9. Define and describe metadata. Why are metadata important?

# 8 Homework Assignment

Due: Jan 30, 2026 [EDITED]




## Problem Description

Spatial data are inherently linked to the Earth's curved surface, yet most analyses and visualizations occur in two-dimensional Cartesian space. Map projections provide a bridge between these representations, but at the cost of distortion. This assignment examines how different coordinate reference systems distort area, distance, shape, and direction, and how those distortions affect analytical conclusions. Implement your analysis in R or Python, using proper data and multiple map projections, and quantitatively assess the implications of projection choice.

### Objectives

1. Explain the principles of geographic vs. projected coordinate systems.
2. Implement CRS transformations in R (or Python).
3. Quantify and visualize projection-induced distortions.
4. Evaluate the appropriateness of projections for specific analytical tasks.
5. Communicate findings clearly using reproducible code and visualizations.

## Data

1. Use a Census polygon that includes multiple geographic regions:
   US states, Texas counties, or Census tracts of a specific county.
2. The data must use a geographic CRS (e.g., WGS84/EPSG:4326).

## Tasks

### Part 1: Conceptual Background

Provide concise written responses to the following questions:

1. What is the difference between a *geographic coordinate reference system* (CRS) and a *projected coordinate reference system*?
2. Define the following types of projection distortion:
   Area, Distance, Shape, Direction.
3. Explain why no single map projection can preserve all four properties simultaneously.

### Part 2: Projection Selection and Transformation

Search/select **at least three** map projections representing different preservation goals (e.g., equal-area, conformal, equidistant, or compromise projections).

For each selected projection:

1. Transform the dataset from a geographic CRS into the target projected CRS.
2. Generate a map visualization of the transformed data.
3. Briefly describe the primary analytical purpose of the projection and the properties it is designed to preserve.

### Part 3: Quantifying Projection Distortion

Using a consistent set of spatial features across all projections, complete the following analyses:

1. **Area Distortion**
    - Compute polygon areas under each selected projection.
    - Compare projected areas to a reference equal-area projection.
2. **Distance Distortion**
    - Select at least three meaningful point-to-point distances.
    - Compute distances under each projection and compare results.

3. **Shape Distortion (Qualitative)**
    - Visually assess deformation of known regions or geometric features.
    - Comment on angular distortion and shape preservation where relevant.

Summarize quantitative results using tables (and figures, or both).

**Part 4: Visualization and Comparison**

Create a comparative visualization that displays the same spatial dataset under each selected projection. Ensure that:

- Visual styling, symbology are consistent across projections where possible.
- Each map includes a clear title and CRS label.

Explain how observed visual differences correspond to the quantitative distortion measures computed in Part 3.

**Part 5: Interpretation and Recommendation**

Provide written responses to the following:

1. Which projection would you recommend for:
    - Global area comparison?
    - Navigation or bearing analysis?
    - Regional distance measurement?
2. What types of analytical errors can result from using an inappropriate projection?
3. How should projection choices be documented in professional or reproducible geospatial data workflows?

## Files to be Submitted

You can submit the homework in two ways:

1. Original Jupyter Notebook that contains the output (no PDF or other format)
2. `R` script (or `Python`) + output file(s)

The basic *requirement* is that the I can run your submission and compare your submitted results against mine.