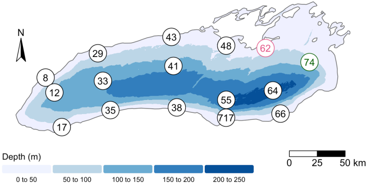
Spatial Analysis:An Introduction

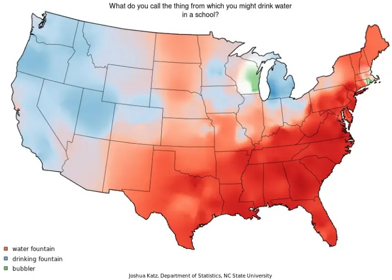
Augustus Pendleton

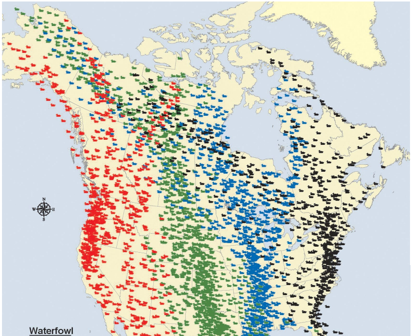
Spatial analysis is more common than you expect!

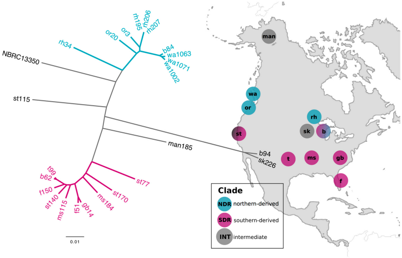
- Environmental microbiology
- Epidemiology
- Microscopy images

Spatial analysis is fun!









Spatial analysis is also *difficult*

- Irregular shapes on an uneven surface
- Historical or technical inaccuracies
- Computationally demanding

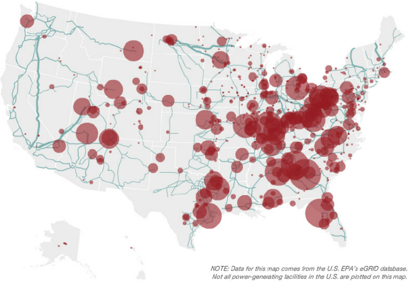
Spatial analysis is rarely reproducible

- Most popular software (ArcGIS, QGIS) are semi-reproducible
- Analyses are rarely shared
- Authors often make un-annotated transformations

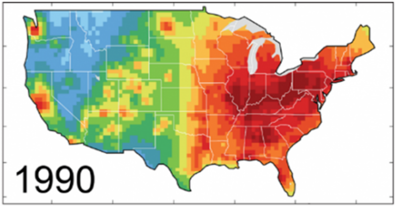
Types of Spatial Data

Vector vs. Raster Data

Vectors have discrete coordinates Example: Coal Fired Powerplants



Rasters are continuous Example: Mercury Deposition Rates



Vector Data: Points, Lines, and Polygon

- Point: Defined by single (x,y) coordinates
- Line: Defined by multiple (ordered) coordinates
- Polygon: Defined by multiple coordinates which form a **closed shape**

Vector Data: Geometries and Attributes

- A vector's points are defined by its **geometry**
- An attribute table provides additional information
  - Names or IDs (e.g. “Station 02”, “Arizona”)
  - Measurements (e.g. “Temperature”)
- *Multi*-points/lines/polygons have multiple geometries with a single attribute record
  - For instance, Michigan has a geometry for the mainland and the Upper Peninsula

Vector Data Filetypes

- Delimited files (.csvs, .txts, .xlsx, etc.)
- Shapefiles (.shp with many files in a trenchcoat)
- Geopackage (.gpkg)
- GeoJSON/JSON (.geojson, .json)
- Open Street Maps (.xml)

Raster Data: Continuous data and images

- Defined by an evenly-spaced grid of cells (or pixels)
- Each cell's "value" represents the average of that cell's area
- Spatial extent of raster defined by coordinates of grid's corners

Raster Data Filetypes

- Any image format (.img, .tif) with embedded spatial information
- Separate files for the raster and the spatial info (e.g. .tif with .tfw)
- Geopackages (.gpkg)

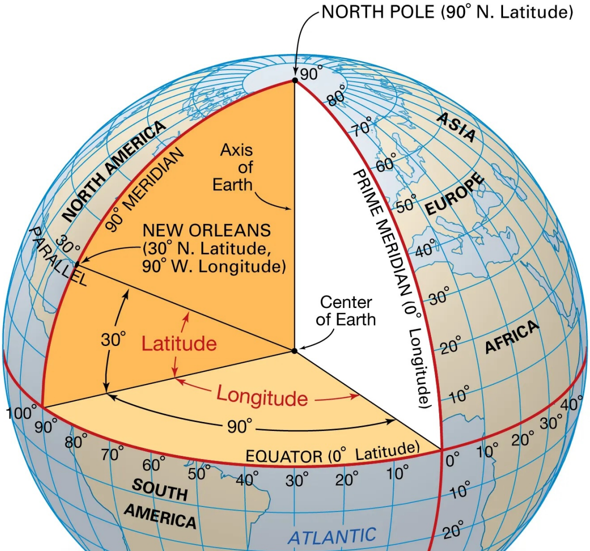
Coordinate Systems

How do we define a coordinate?

- The Earth is spherical
- To define global position we use *angular coordinates*
- To make maps, we use *planar coordinates*

Angular/Geographic Coordinates

- Latitude (angle from equator)
- Longitude (angle from primer meridian)



Reporting Angular Coordinates

- Degrees, minutes, and seconds
  - 40° 31' 21" North by 105° 5' 39" West
- Decimal degrees
  - 40.866389°, -124.082778°
- North and East are positive, South and West or negative
- If you can, please record your data in decimal degrees

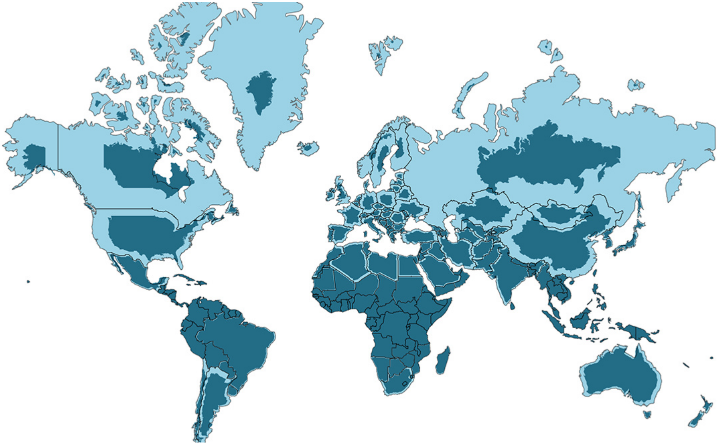
Datums

- The earth isn't perfectly symmetrical
- A datum is a 3D model of the Earth we use for angular coordinates
- WGS84 is most popular (most GPS will give you coordinates based on this datum)
- You may also see NAD83 for older data
- There are other, more accurate datums for specific regions

Projected Coordinate Systems

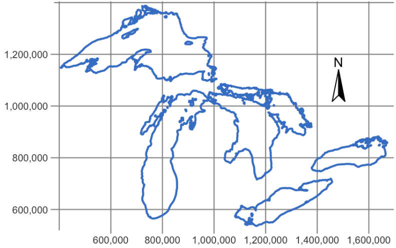
- Maps are 2D; the Earth is 3D
- We need to *project* angular coordinates to planar (cartesian) coordinates
- Project coordinate systems are in x,y coordinates from a defined origin
- Coordinates are in linear units (e.g. meters or miles)
- Projections **always** cause distortions

Projection **always** causes distortion

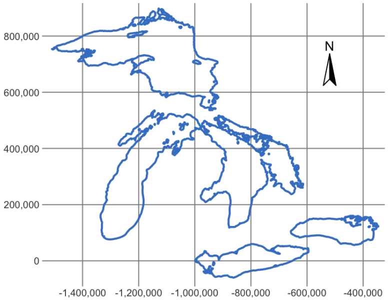


Bad projection makes bad maps!

Good projection



Inaccurate projection



Defining a Coordinate System

- Many spatial datatypes specify their coordinate systems
- If you're collecting data, you **must** report what CRS/datum you use
  - Often, this is WGS84
- Historical and governmental data may used specific CRS and datums!
- Sometime you'll need to tell your software *which* CRS the data is in
- Sometime you'll need to project your data to a new CRS

Spatial Analysis Software

Popular GIS Systems

- ArcGIS
  - Most powerful, popular, and well supported spatial software
  - $$$$
  - Closed source and often slow
- QGIS
  - Open-source version of ArcGIS - **great option**
  - Fast, but crashes a lot
- Both "point and click" but automation and scripting possible

The R Spatial Environment

Why R for spatial analysis?

- Open-source and free
- Fully reproducible
- Easy to slot into existing data analysis workflows
- Can use tools like RMarkdown/Quarto
- Builds on familiar grammar like the tidyverse

`sf`

- We'll use this package to work with vector data
- Combines geometries and attributes into a familiar-looking dataframe
- Uses tidy grammar
- Replaces the `sp` package

`terra`

- We'll use this package to work with raster data
- Replaces the `raster` package

`tmap`

- We'll use this package to create maps
  - `ggmap` also an option
  - `tmap` has more functionality, but also more complicated
  - Both in active development, so will be interested to see how they develop over the next two years!

External Software

- May also need to install separate software, including:
    - GEOS
    - PROJ
    - GDAL
- R packages (and QGIS) reference these programs to analyze spatial data

Installation Goals:

1. R $>$ 4.0
2. Install from CRAN:
   2.1 install.packages("tidyverse")
   2.2 install.packages("sf")
   2.3 install.packages("terra")
   2.4 install.packages("remotes")
3. Install from repository
   3.1 remotes::install_github("r-tmap/tmap")
   3.2 remotes::install_github("mtennekes/tmaptools")
4. Check loading:
   4.1 library([each package])
5. If necessary, we'll install GEOS, PROJ, or GDAL (depending) - Gus will help