Spatial Data Analysis

Week 2: Spatial Thinking: Distance, Projections, and Spatial Computations

Meredith Franklin

Department of Statistical Sciences and School of the Environment

September 12th, 2025

Topics Covered

Today we will cover:

- Definitions of distance
- Definitions of projections
- The `sf` package

The R code that accompanies this lecture is spatial2.Rmd

Spatial Thinking

Spatial data have two special traits, location and relation

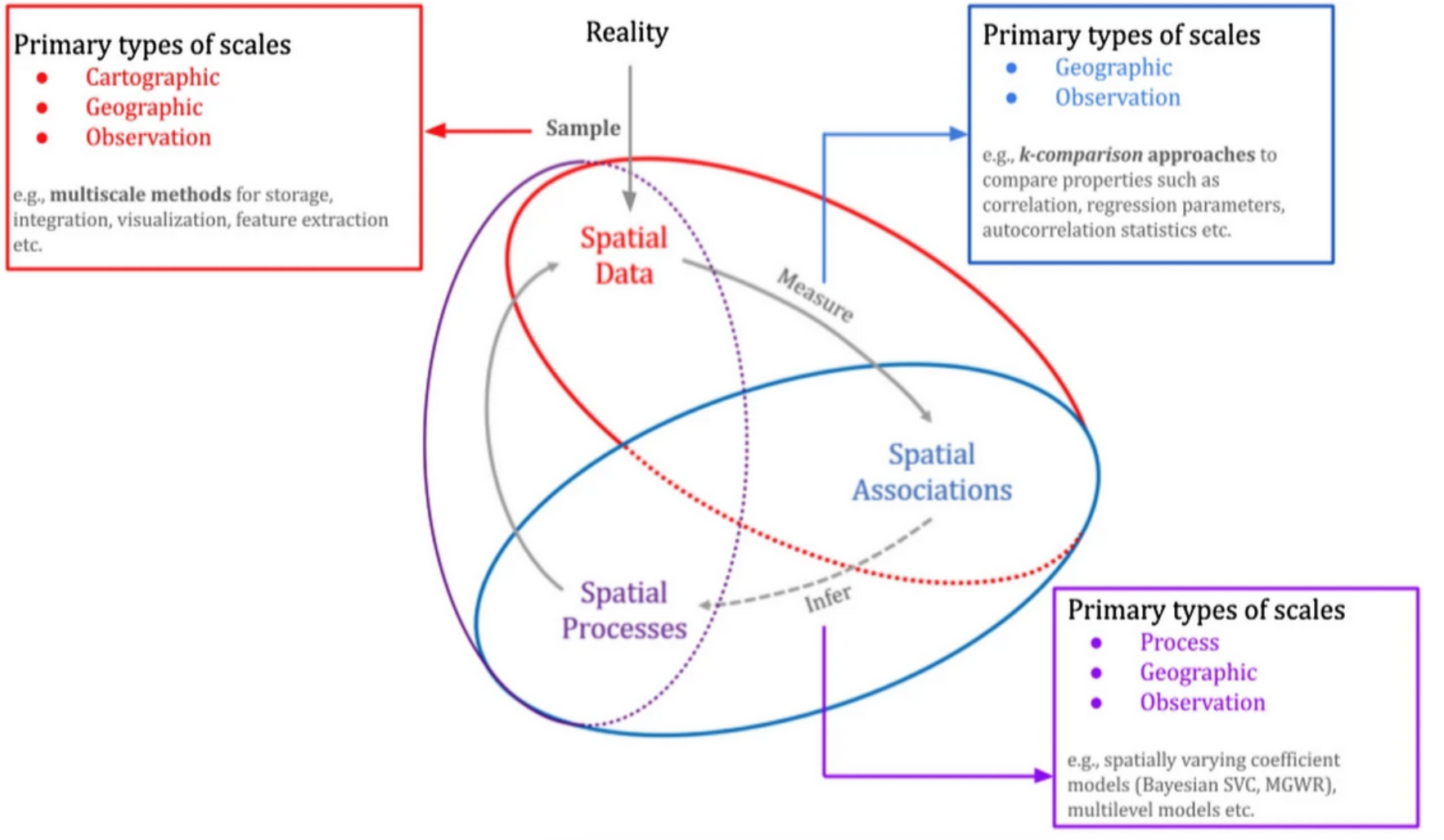
- The location in space (and time) gives the position, and a map is a representation of the locations.
- The relations between locations let us contextualize the data. We build links between locations that then allows us to build models.
- Spatial processes are represented by locations, relations, and often have a feature.
- Locations are not isolated from each other, and interactions occur among adjacent places. The spatial dimension that defines these relations is distance.

Scale

- The appropriate geographical scale should be selected prior to any geographical/spatial analysis, as it directly affects the selection of the data model, the data to be collected, the methods to be used and the way conclusions will be drawn.
- Scale is used to refer to the level at which data are collected (e.g. individuals, census tracts, counties) or the range over which spatial processes vary (e.g. local, regional, global).
- For example, we might find clustering in a feature (unemployment) at a 100 m and a 10,000 m scale, representing patterns at both a very local and city level.

Scale

From: A scoping review on the multiplicity of scale in spatial analysis



https://link.springer.com/article/10.1007/s10109-022-00384-8

Scale

- Different processes have different spatial and temporal scales at which they operate.
- Scale can refer to both "extent" and "resolution".
- Processes that operate over a larger extent (e.g., a province) can be studied at a larger resolution (counties or cities) whereas processes that operate over a smaller extent (e.g. a neighborhood) may need to be studied at the level of buildings.
- Scale affects our estimates of distance (total distance or "extent" and within feature distance or "resolution").
- Resolution affects our understanding of relationships between variables of interest. In terms of data collection this means that we want data to be at the highest spatial (and temporal) resolution possible (affordable).
- We can aggregate our data to lower resolutions, but it is often difficult or impossible to correctly disaggregate ("downscale") data to a higher resolution.

Distance

- Distance is a numerical description of how far apart things are.
- Recall Tobler's First Law of Geography "everything is related to everything else, but near things are more related than distant things".
- Distances can be regular (gridded data) but are more often irregular. Point data usually rise to few (usually one) pair of points for a given distance.
- In time, there is only one dimension, but in space there are two, so direction can matter as well as distance (this is called anisotropy, which we will cover later).

Distance

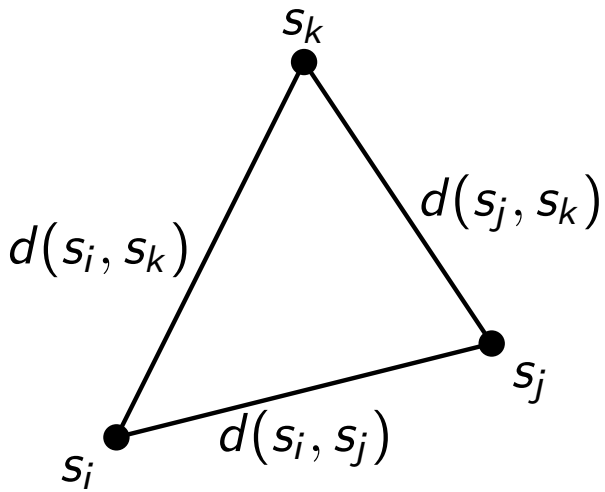
- Spatial distance can be defined in various ways depending on the context and the nature of the space involved.
- We often have to define a domain where we can compute distances (e.g. a country, city) and some times features need to be considered that "get in the way", naturally defining our borders (e.g. a mountain).
- Many times the spatial scale drives what distance metric we choose.

Distance

- Distance is often represented as d or h
- Several ways to define spatial distances but they must satisfy several technical conditions:
  1. symmetry $d(s_i, s_j) = d(s_j, s_i)$
  2. the distance between a point and itself is zero
  3. the triangle inequality $d(s_i, s_k) \leq d(s_i, s_j) + d(s_j, s_k)$

Triangle Inequality

- The direct distance between two points is never longer than going through a third point.



Triangle inequality

$$d(s_i, s_k) \leq d(s_i, s_j) + d(s_j, s_k)$$

Distance Types in Spatial Analysis

- Different applications use different notions of distance:
  - **Euclidean**: straight-line in flat maps.
  - **Geodesic**: shortest path on Earth's curved surface.
  - **Manhattan**: constrained to grid-like movement.
  - **Network**: constrained to road/path networks.
  - **Surface/Topographic**: across terrain (3D), like hiking path. Uses manifolds.

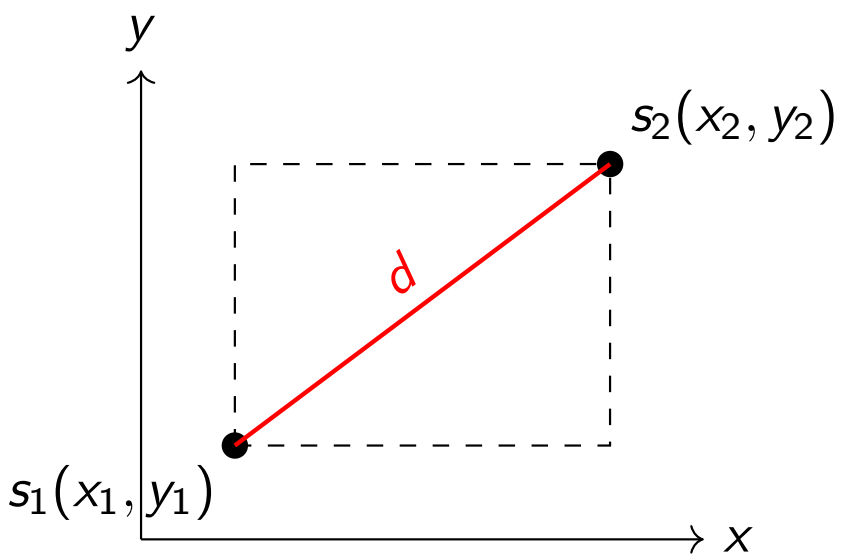
Euclidean Distance

- The most common way to define spatial distance is using the Euclidean distance, which is the "straight-line" distance between two points in a space.
- This is the traditional method of measuring the shortest path between two points.

Euclidean Distance

In a 2-dimensional space, if you have points $s_1 = (x_1, y_1)$ and $s_2 = (x_2, y_2)$, the Euclidean distance d is calculated as:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



Geodesic Distance

- This is used to measure the distance between two points on the surface of a sphere (e.g., between two cities on Earth).
- The great-circle distance is an example of geodesic distance, and is the shortest distance between two points along the surface of the sphere. It takes into account the curvature of the Earth.
- The formula for great-circle distance involves trigonometry and the haversine function, particularly when using latitude and longitude coordinates.
- This method is often used for calculating distance between two very distant locations on earth.

Geodesic Distance

Haversine formula for great-circle distance

The haversine formula for great-circle distance is calculated given two locations s_1 and s_2 , with:

- Latitude and longitude of s_1: ϕ_1 (latitude), λ_1 (longitude)
- Latitude and longitude of s_2: ϕ_2 (latitude), λ_2 (longitude)

Then distance, d between the two points on a sphere with radius r (Earth's radius is approximately 6371 kilometers) is:

$$d = 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

Geodesic Distance

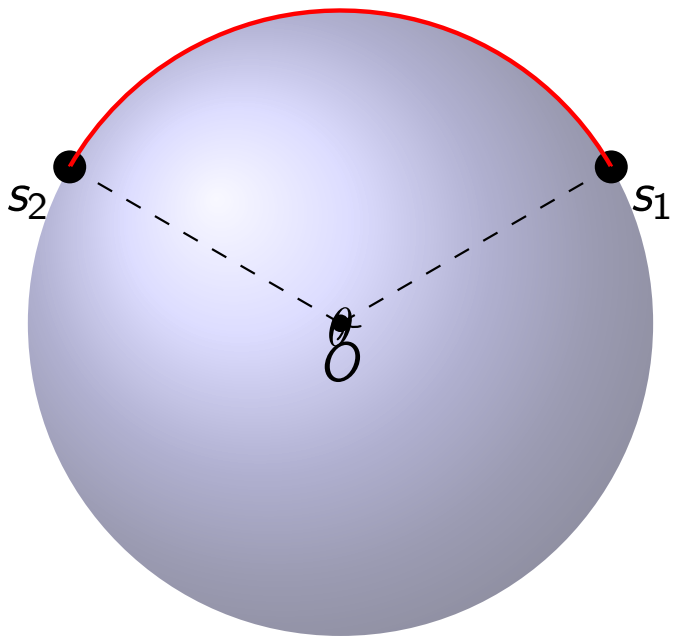
Haversine formula for great-circle distance, con't

- ϕ_1 and ϕ_2 are the latitudes of the two points (in radians).
- λ_1 and λ_2 are the longitudes of the two points (in radians).
- r is the radius of the sphere (e.g. Earth, typically 6371 km).
- The formula calculates the central angle (in radians) between the two points, then multiplies by the radius of the sphere to find the great-circle distance.

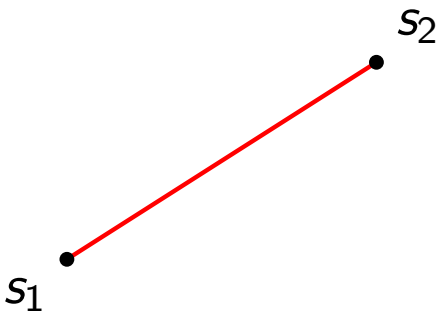
To use the formula we need to convert latitude and longitude from degrees to radians (if they are in degrees) by radians $= \frac{\pi}{180} \times$ degrees and then apply the Haversine formula to calculate the distance between the two points.

Geodesic (Great-Circle) Distance

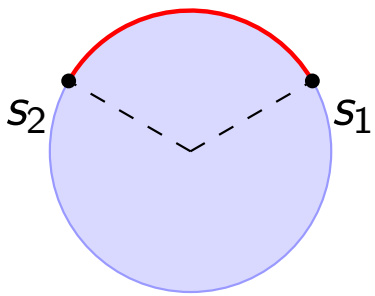
- Distance measured along the surface of a sphere (e.g., Earth).
- The shortest path is the **great-circle arc**, not the straight line through space.
- This accounts for Earth's curvature and is used in navigation and global distance calculations.



Comparing Distance Types



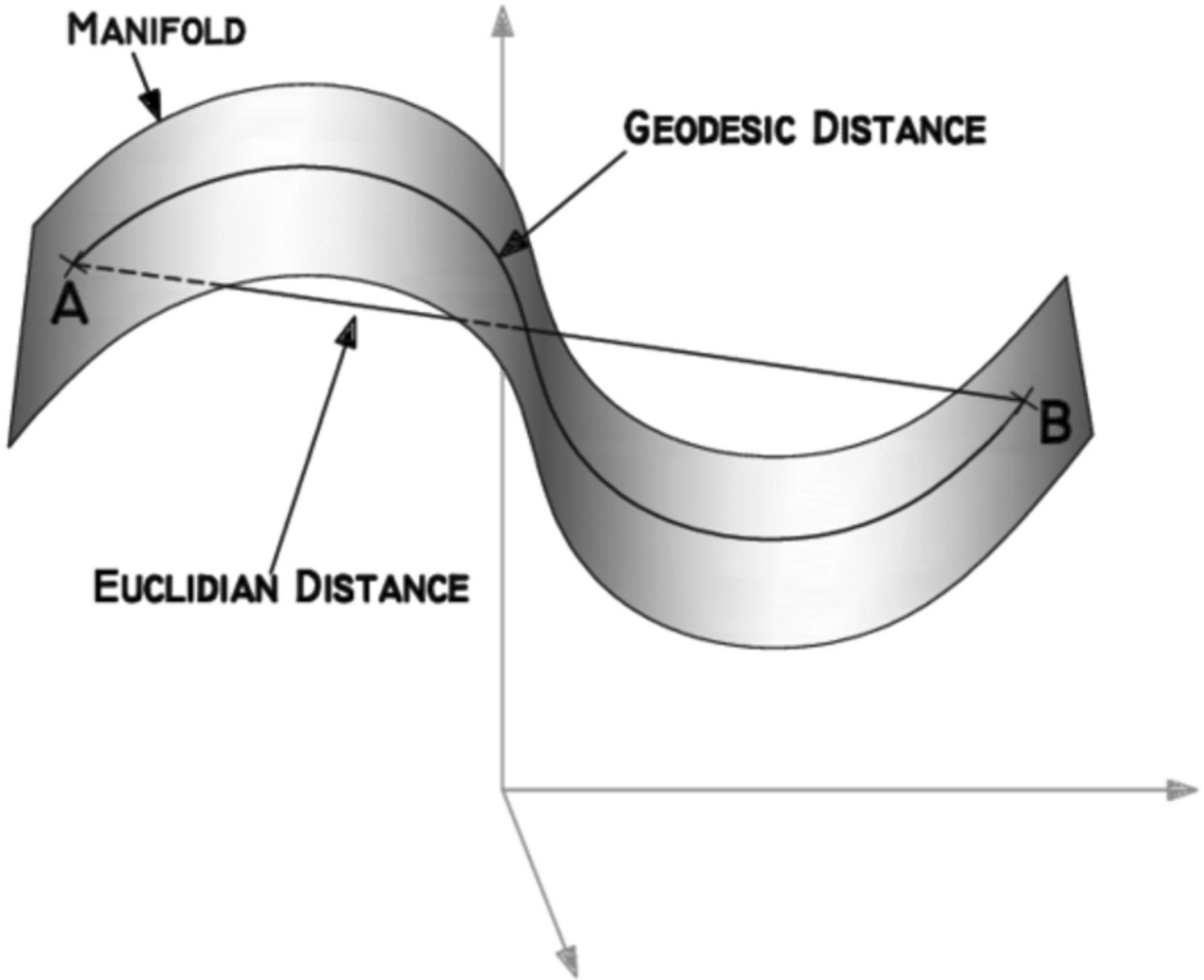
Euclidean



Geodesic (great-circle)

Manifold (curved surface)

Distances



From "Dimensionality Reduction in Surrogate Modeling: A Review of Combined Methods" in Data Science and Engineering (2022)

Manifold Distance

- A **manifold** is a curved space that looks flat locally but may be curved globally.
- **Manifold distance** = shortest path constrained to lie on the manifold (a **geodesic**).

General definition

$$d_M(p, q) = \min_{\gamma} \int_0^1 \sqrt{\langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle}\, dt$$

- γ = smooth path on the manifold with endpoints p, q
- The geodesic minimizes path length

- **Sphere (Earth)**: great-circle distance between lat/long points
- **Terrain surface**: hiking path length across hills/valleys (DEM)
- **Network manifold**: shortest path on a road/rail graph

Manhattan Distance

- Manhattan distance, also known as taxicab distance or L1 distance measures distance between two points in a grid-based system by summing the absolute differences of their coordinates.
- Manhattan distance is useful where movement is restricted to horizontal and vertical directions, such as along a regular grid (raster) or streets in cities with grid-like systems.
- Manhattan distance is the total number of units you would travel if you could only move horizontally or vertically along a grid such as only going north-south or east-west but not diagonally.
- Manhattan distance defines a valid metric on Euclidean space (it satisfies symmetry, identity, and triangle inequality).

Manhattan Distance

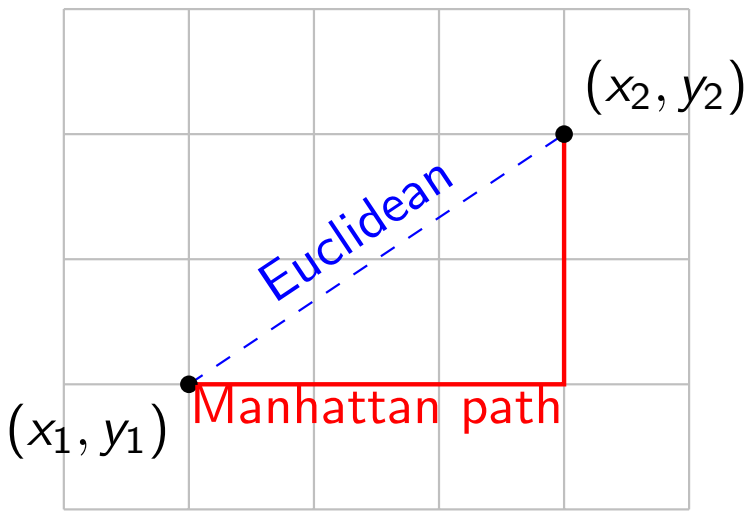
For locations $s_1 = (x_1, y_1)$ and $s_2 = (x_2, y_2)$ in 2-dimensions, Manhattan distance d is given by:

$$d = |x_2 - x_1| + |y_2 - y_1|$$

For example, $s_1 = (1, 2)$ and $s_2 = (4, 6)$, the Manhattan distance is $d = |4 - 1| + |6 - 2| = 3 + 4 = 7$ meaning that in a grid-like path, the shortest route between s_1 and s_2 would involve moving 7 units.

Manhattan Distance

- Manhattan (taxicab) distance measures distance as the sum of horizontal and vertical moves.
- Example: two points (x_1, y_1) and (x_2, y_2).
- Distance: $d = |x_2 - x_1| + |y_2 - y_1|$.



Minkowski Distance

- Minkowski Distance is the generalized form of the Euclidean and Manhattan distances.
- It is defined by a parameter p, which determines the form of the distance calculation.

Minkowski Distance

For two locations $s_1 = (x_1, y_1)$ and $s_2 = (x_2, y_2)$ in 2-dimensional space, the Minkowski distance d is given by:

$$d_p(s_1, s_2) = \left(|x_2 - x_1|^p + |y_2 - y_1|^p\right)^{1/p}$$

where p is the order of the Minkowski distance (a parameter that defines the type of distance).

- When $p = 1$ (Manhattan distance): $d = |x_2 - x_1| + |y_2 - y_1|$
- When $p = 2$ (Euclidean distance): $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- When $p \to \infty$ (Chebyshev distance): $d = \max\left(|x_2 - x_1|, |y_2 - y_1|\right)$

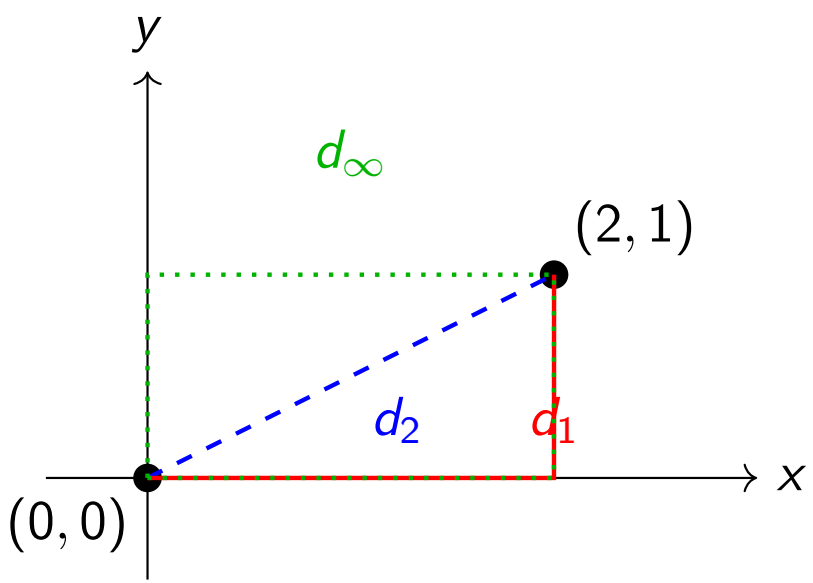
Minkowski Distance Example

- General formula:

$$d_p(s_1, s_2) = (|x_2 - x_1|^p + |y_2 - y_1|^p)^{1/p}$$

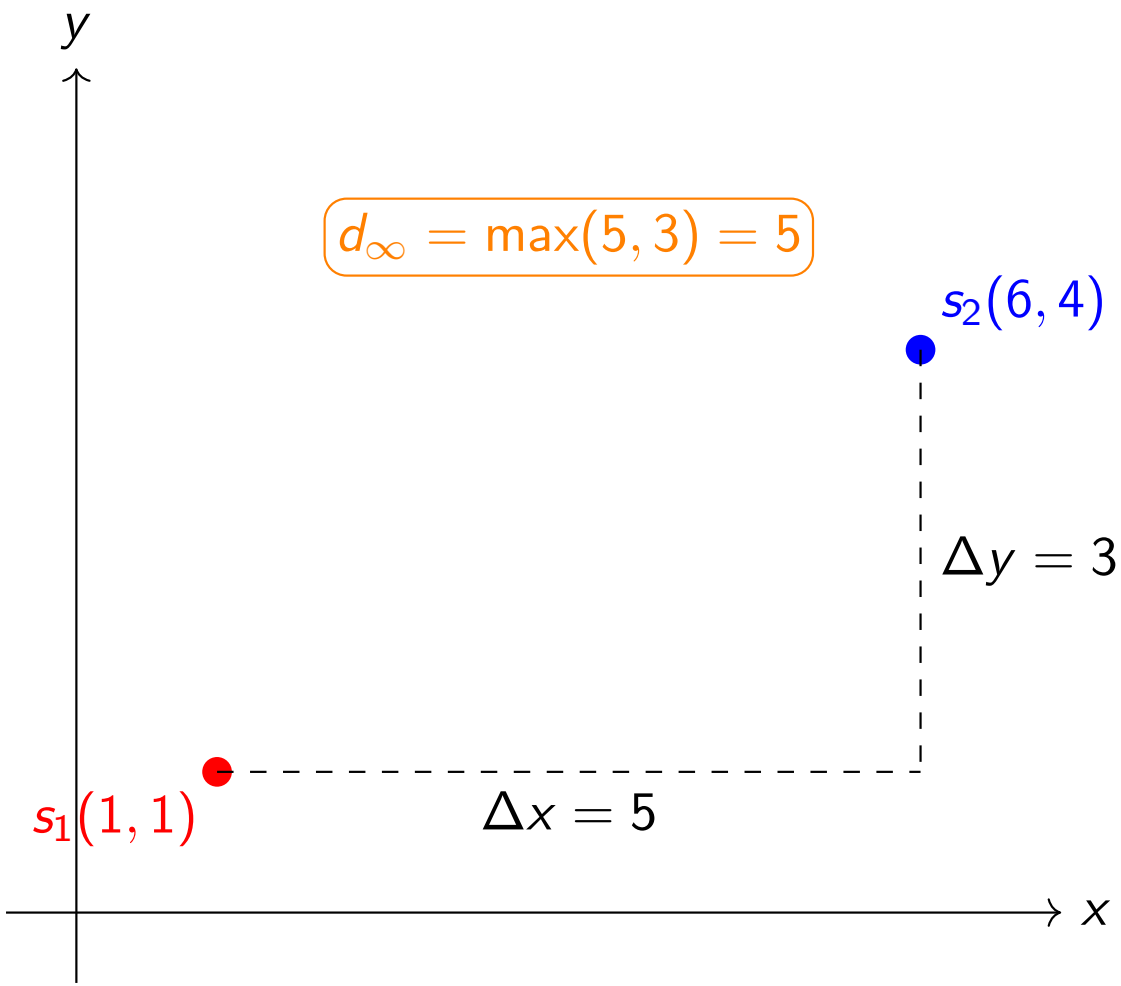
- For $s_1 = (0, 0)$ and $s_2 = (2, 1)$:

$$d_1 = 3 \quad \text{(Manhattan)}$$
$$d_2 = \sqrt{5} \approx 2.24 \quad \text{(Euclidean)}$$
$$d_\infty = 2 \quad \text{(Chebyshev)}$$



Chebyshev Distance as Maximum Difference

- Defined: $d_\infty(s_1, s_2) = \max(|x_2 - x_1|, |y_2 - y_1|)$.
- Intuition: When diagonal moves are allowed, the distance is limited by the larger coordinate gap.



Why $d_\infty \leq d_2 \leq d_1$

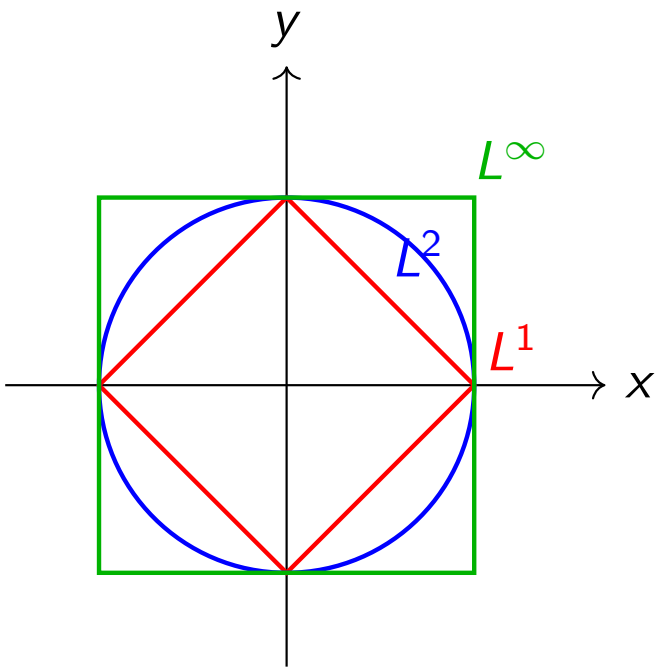
- The "unit ball" (all points with distance ≤ 1 from the origin) depends on p:
  - L^1: $|x| + |y| \leq 1 \rightarrow$ diamond
  - L^2: $x^2 + y^2 \leq 1 \rightarrow$ circle
  - L^∞: $\max(|x|, |y|) \leq 1 \rightarrow$ square
- Geometric nesting:

$$\text{square} \subset \text{circle} \subset \text{diamond}.$$

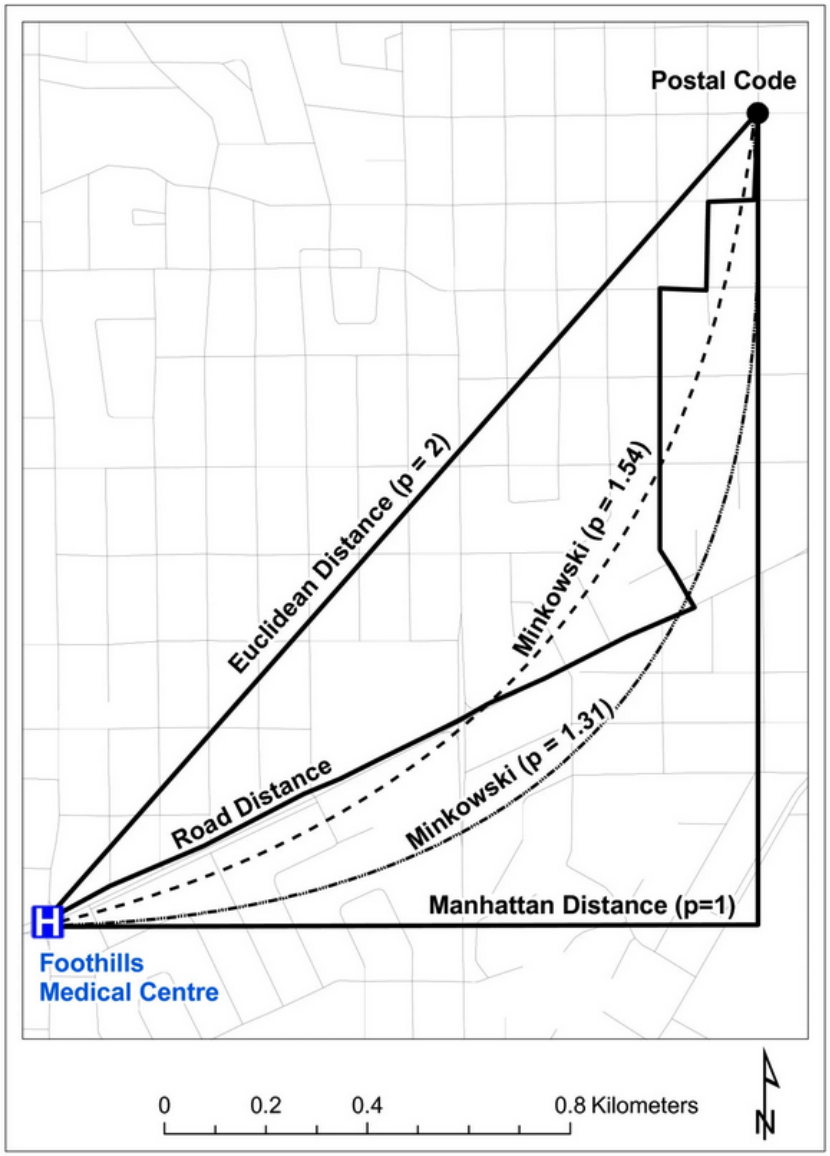
- Therefore, for any two points,

$$d_\infty \leq d_2 \leq d_1$$

- L^2 is curved because $x^2 + y^2 \leq 1$ defines a circle, while L^1 and L^∞ are linear inequalities, giving straight edges.



Application of Distance Calculations



From "Comparison of distance measures in spatial analytical modeling for health service planning" in BMC Health Services Research (2009)

- Evaluated Minkowski p from 1 to 2
- Compared against:
  - Road network distances
  - Travel times
- Best-fit p values:
  - $p \approx 1.54 \rightarrow$ road distance
  - $p \approx 1.31 \rightarrow$ travel time
- Shows that real travel lies "between" Euclidean and Manhattan
- Travel time also a good predictor of road distance, thus providing the best single model of travel. The Minkowski method produces more reliable results than the traditional Euclidean metric.

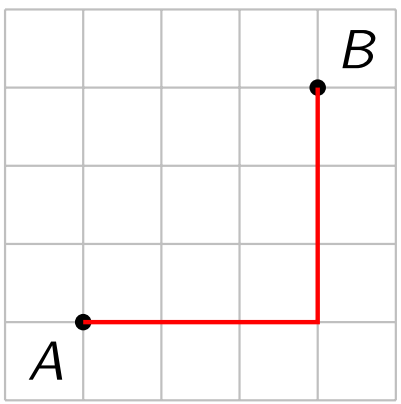
Key Findings

- Euclidean systematically underestimates real travel
- Manhattan systematically overestimates
- Optimized Minkowski p gives a better approximation
- Choice of distance metric affects:
  - Spatial weights matrices
  - Regression model fit
  - Policy implications for access to care
- Practical: Minkowski can be tuned when road/travel-time data are unavailable

Manhattan vs. Network Distance

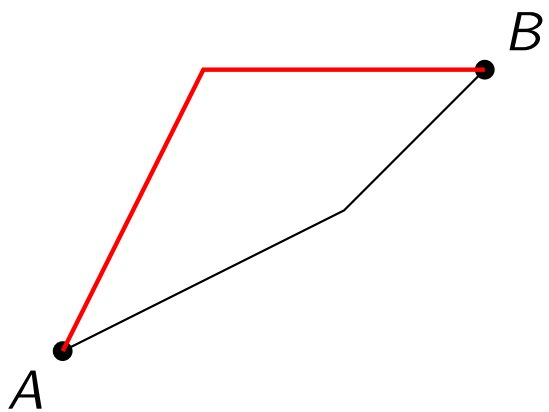
Manhattan (grid-based)

- Sum of horizontal + vertical steps
- Models urban grids



Network (roads/paths)

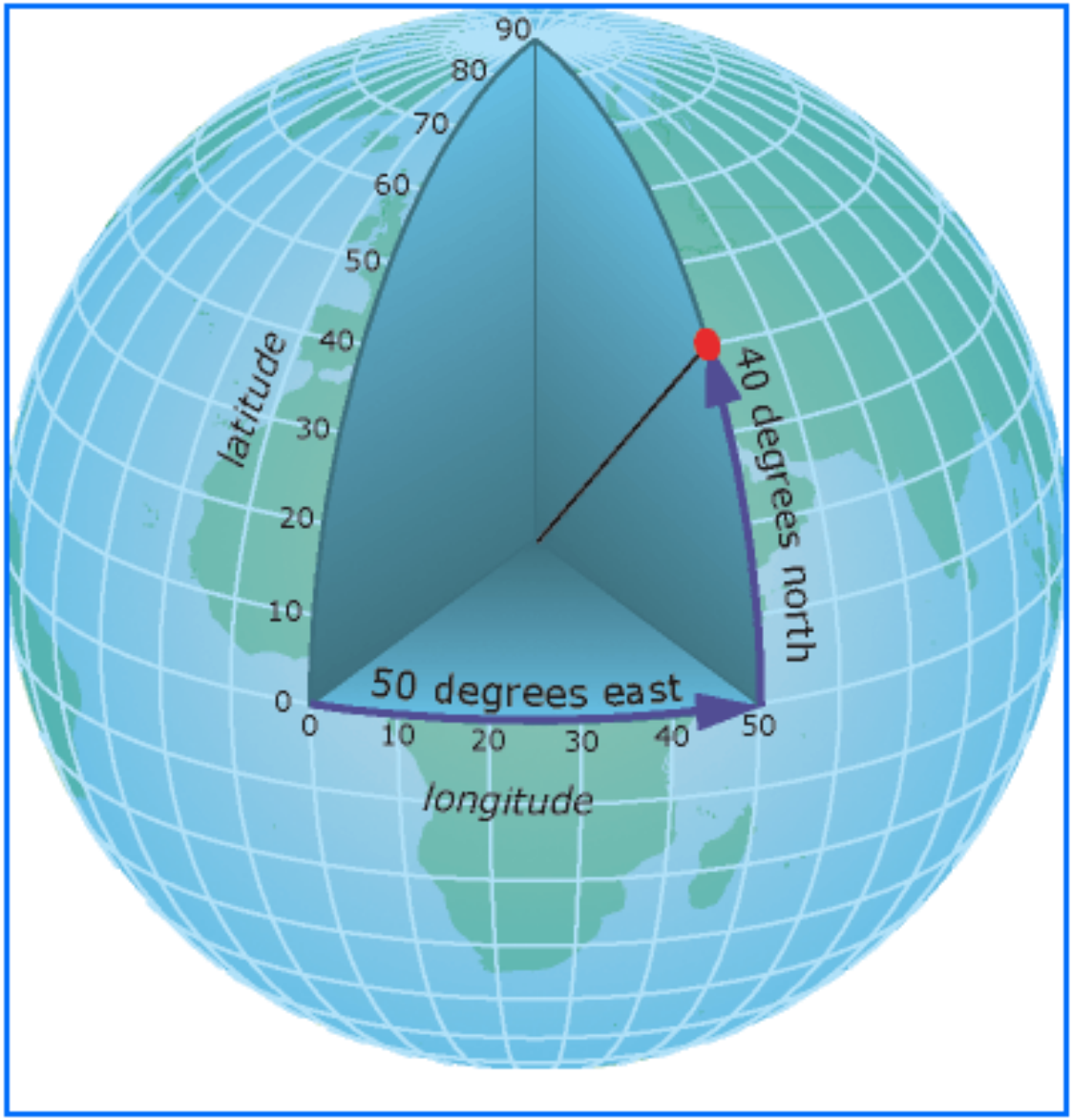
- Distance along network
- Used in routing/logistics



Projections

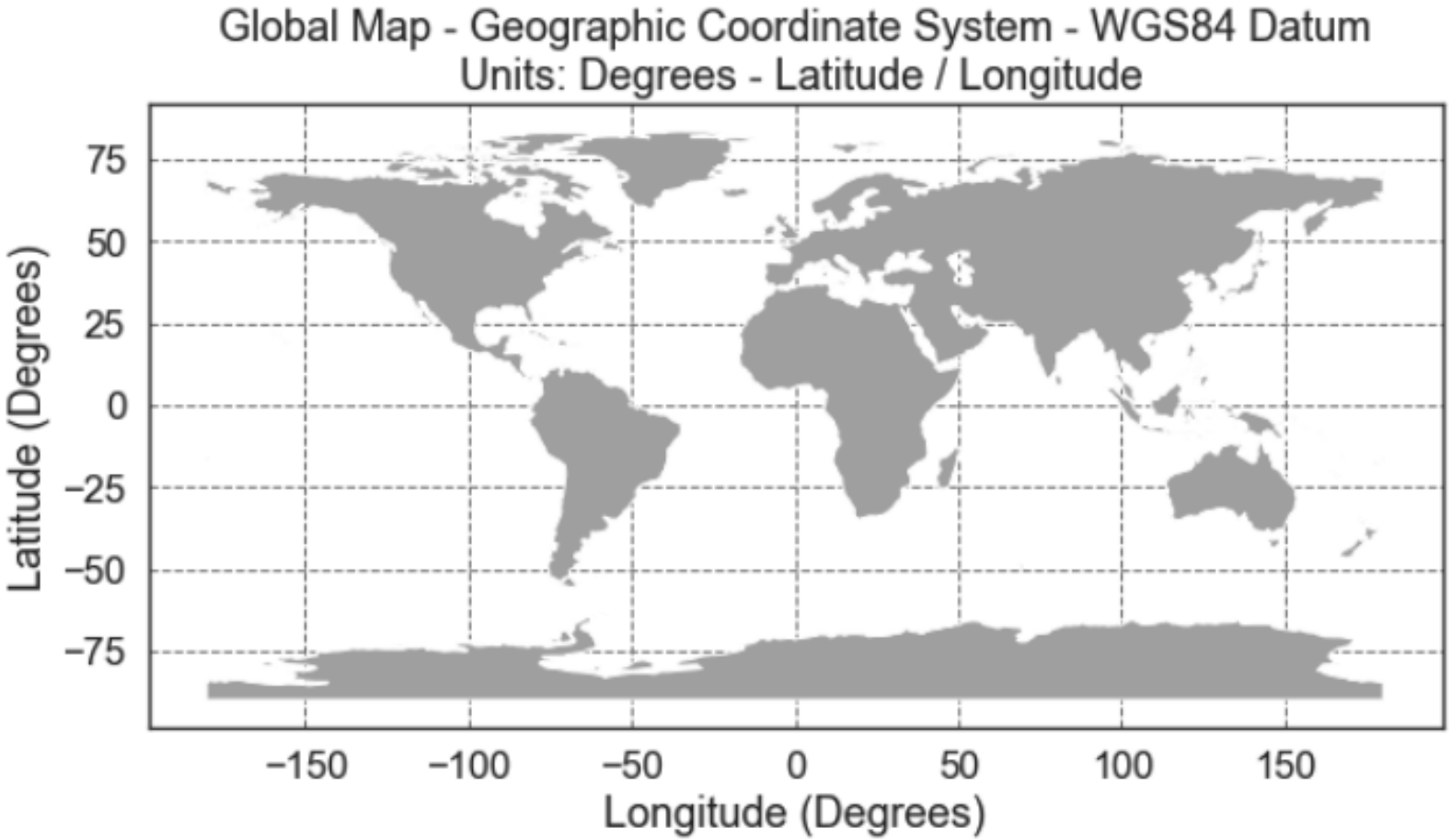
- To represent (Euclidean) distances between spatial locations meaningfully, we need cartographic projections.
- The purpose of a projection is to convert geographic coordinates from the sphere (latitude, longitude) to a plane (x,y).
- Since the sphere cannot be directly flattened to a plane without distortion, and intermediate step is taken using a "developable surface" including a cone or cylinder.
- The projections themselves are complex differential equations, but thankfully all of the difficult mathematics has been done for us. In practice we choose a particular projection predefined for the local area or region of study.
- A local projection preserves distances. Examples: the UTM zone system, state plane system, Albers equal area (conic projection).

Geographic Coordinate Systems



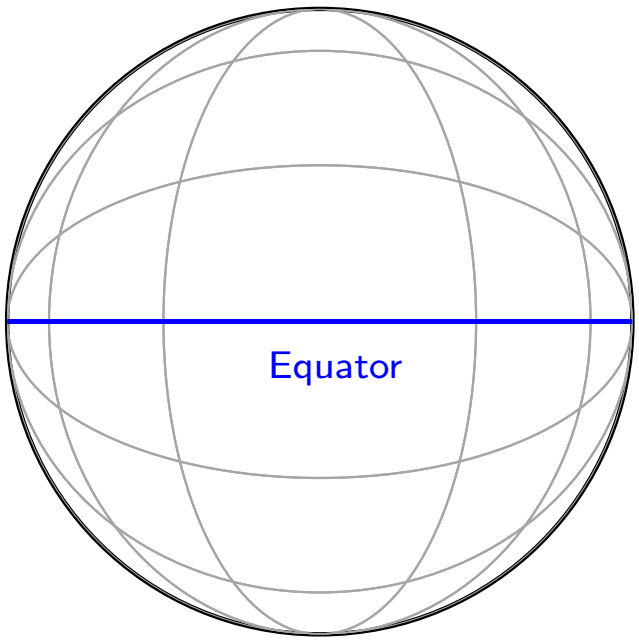
A geographic coordinate system locates latitude and longitude location using angles. Thus the spacing of each line of latitude moving north and south is not uniform. Source: ESRI

Geographic Coordinate Systems

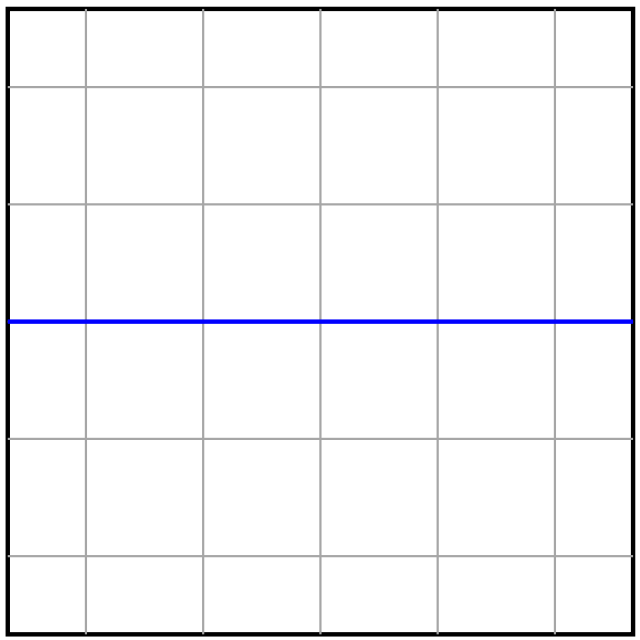


The distance between the 2 degrees of longitude at the equator (0○) is $\sim$ 111 km. The distance between 2 degrees of longitude at 40○N (or S) is only 85 km. This difference in actual distance relative to "distance" between the actual parallels and meridians demonstrates how distance calculations will be less accurate when using geographic CRSs

Sphere (lat/lon grid) projected to a plane

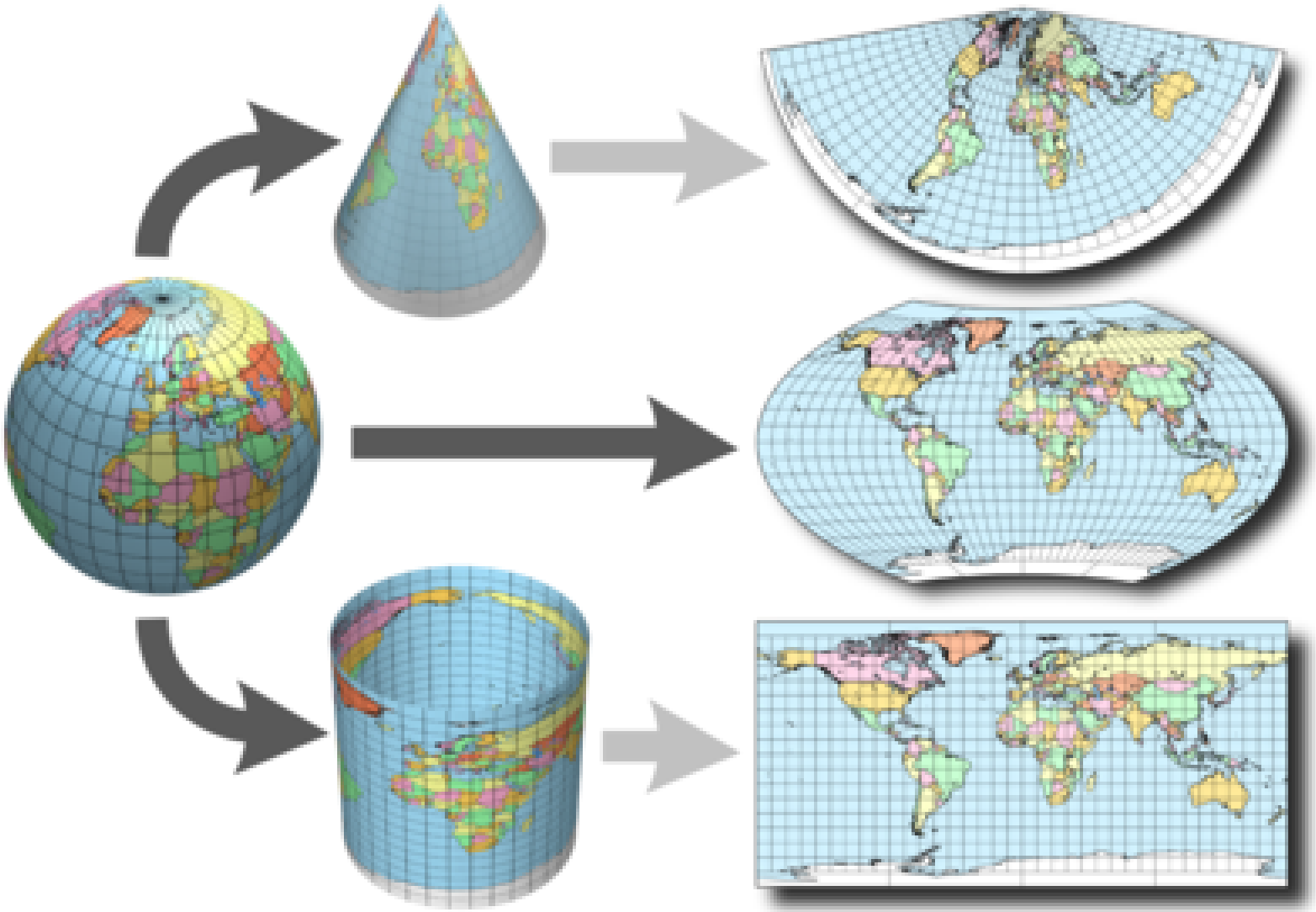


Sphere with lat/lon grid

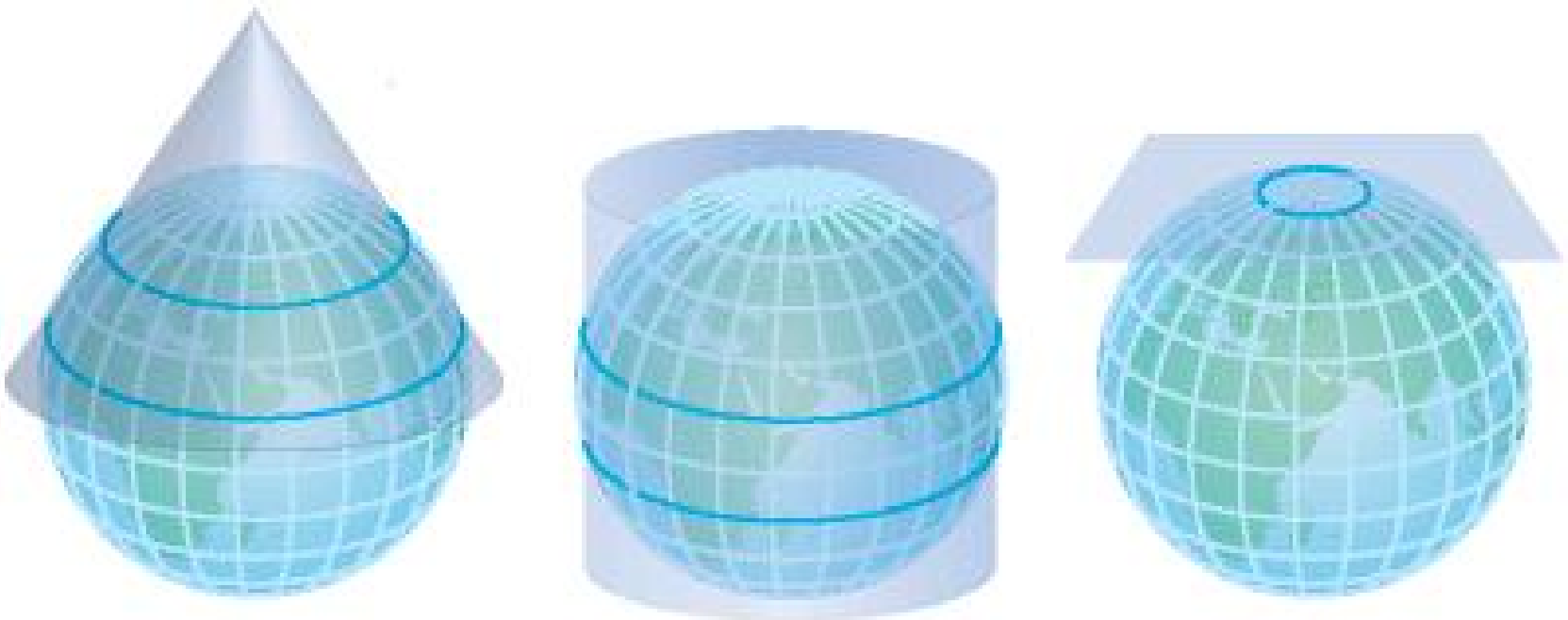


Projection plane (map grid)

Projection Types



Projection Types



Projection types: Conic, Cylindrical, Planar

Projection Types

- **Conic projections**
  - Project sphere onto a cone tangent or secant to Earth.
  - Parallels are arcs, meridians are straight lines converging at a point.
  - Useful for mid-latitude regions (e.g., Albers, Lambert Conformal Conic).
- **Cylindrical projections**
  - Project sphere onto a cylinder (e.g., Mercator).
  - Meridians and parallels appear as straight, perpendicular lines.
  - Distortion increases toward the poles.
- **Planar (Azimuthal) projections**
  - Project sphere onto a flat plane (tangent at a point).
  - Meridians are straight lines radiating from the center, parallels are circles.
  - Good for polar maps (e.g., stereographic, gnomonic).

Projection Properties

There are four properties to consider when converting from three to two dimensions: (1) shape, (2) area, (3) distance, and (4) direction. Common projections that we use are:

- Albers equal area (conic): must center at the correct location, preserves area, distorts near poles and equator.
- Mercator such as UTM (cylindrical): must pick the correct zone and hemisphere, preserves direction and shape in small areas, distorts near poles.
- Planar (orthographic or azimuthal): each state/province has its own system, very localized to a central point, distorts area and shape as you move farther from the center.

Projections - How Maps Distort

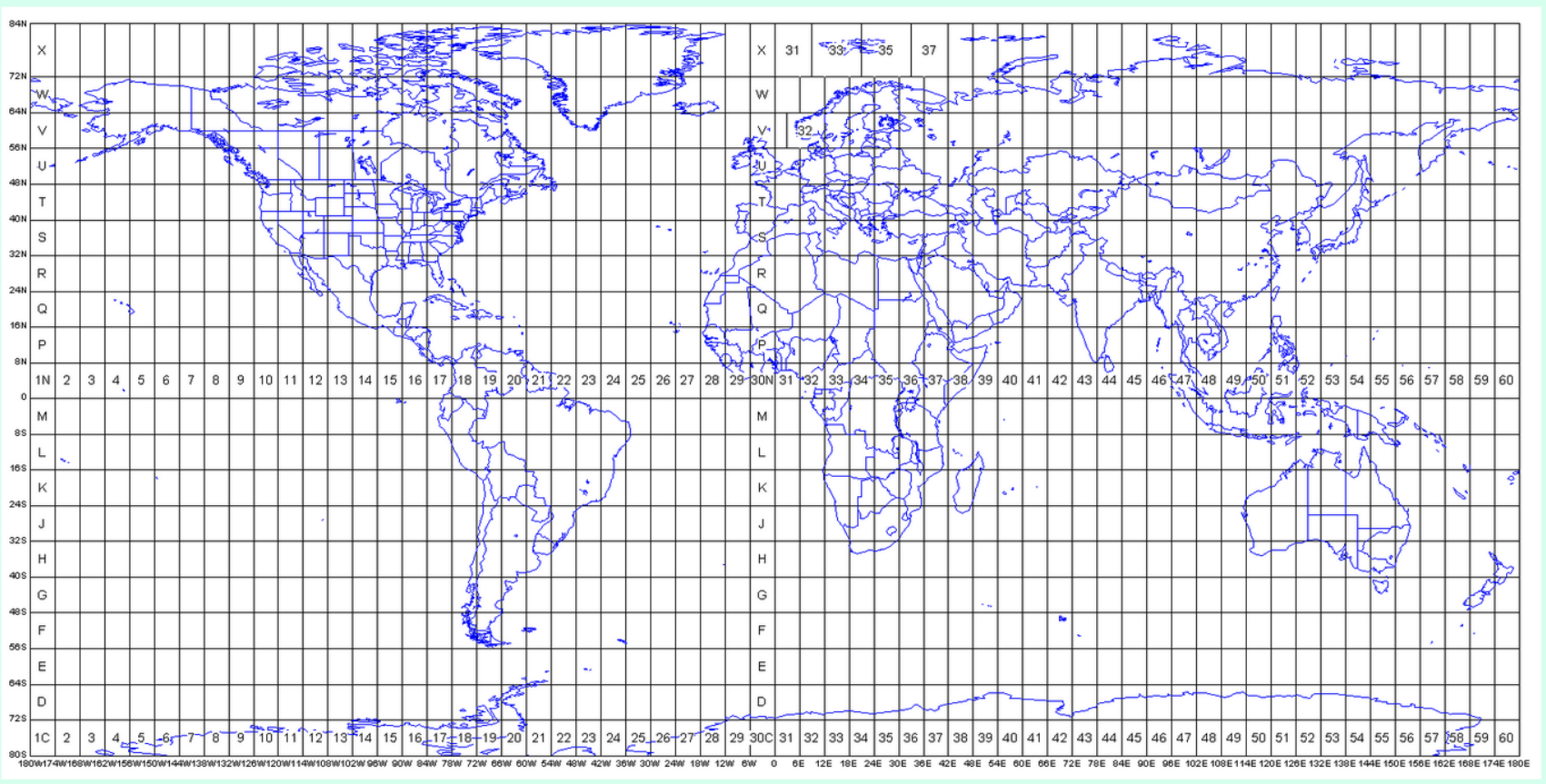
The Mercator projection exaggerates the size of the countries as you move away from the Equator. Check out `https://www.thetruesize.com`

Universal Transverse Mercator (UTM) Projections

- Transverse Mercator is a cylindrical projection
  - Cylinder is wrapped around the Earth **longitudinally** (north–south axis).
  - Unlike standard Mercator, the central meridian is the line of tangency.
- **Global system:**
  - Earth is divided into **60 zones**, each spanning 6^o of longitude.
  - Zones are numbered 1 - 60 starting at 180^oW.
  - Each zone projected separately to minimize distortion.
- **Properties:**
  - Conformal: preserves shape and angles locally.
  - Distortion minimized within each zone (scale factor at central meridian $\approx$ 0.9996).
  - Units in meters, using a Cartesian grid (Easting, Northing).

UTM Projections

UTM Zones in the World



https://www.dmap.co.uk/utmworld.htm

UTM Projections

UTM Zones in Canada

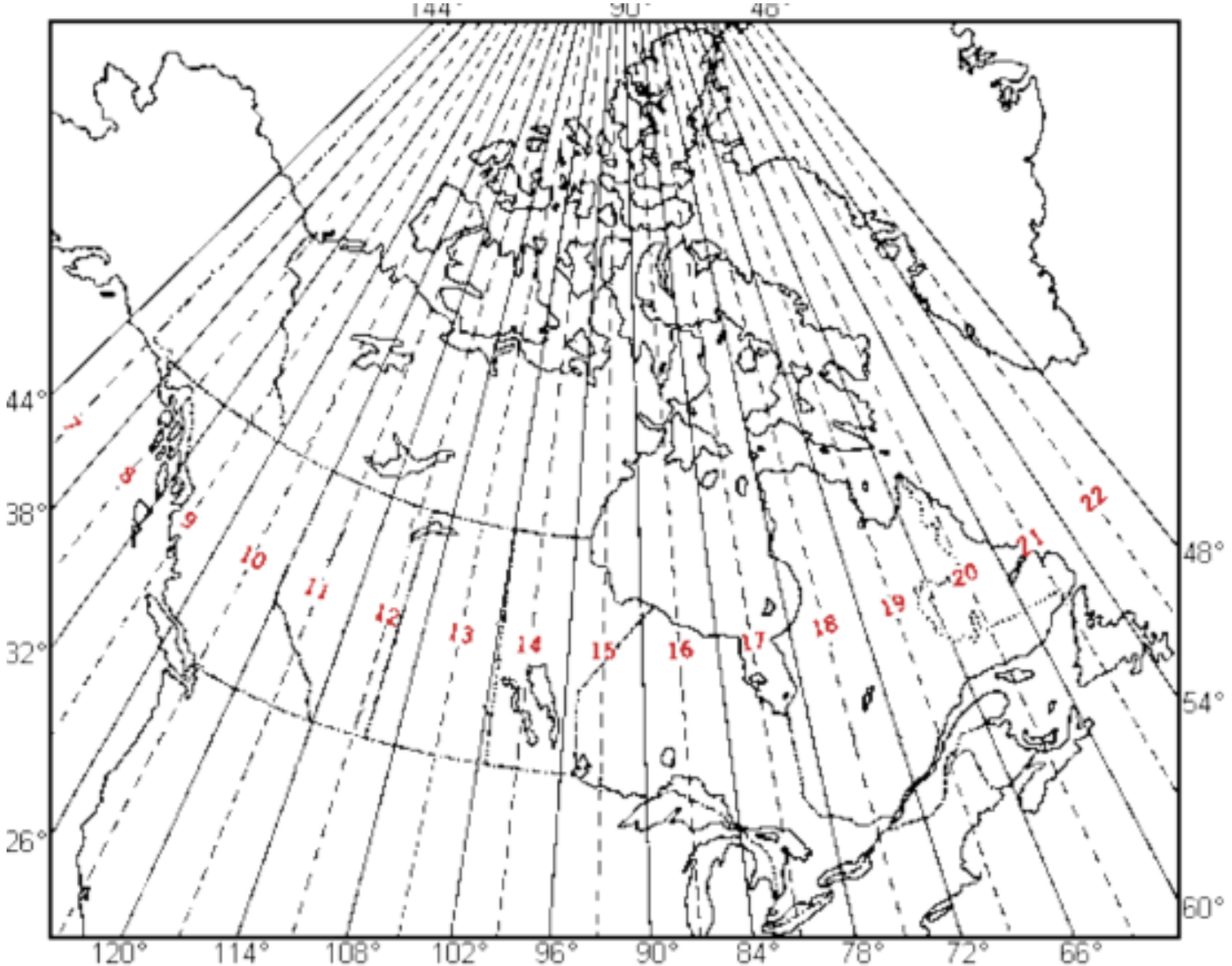
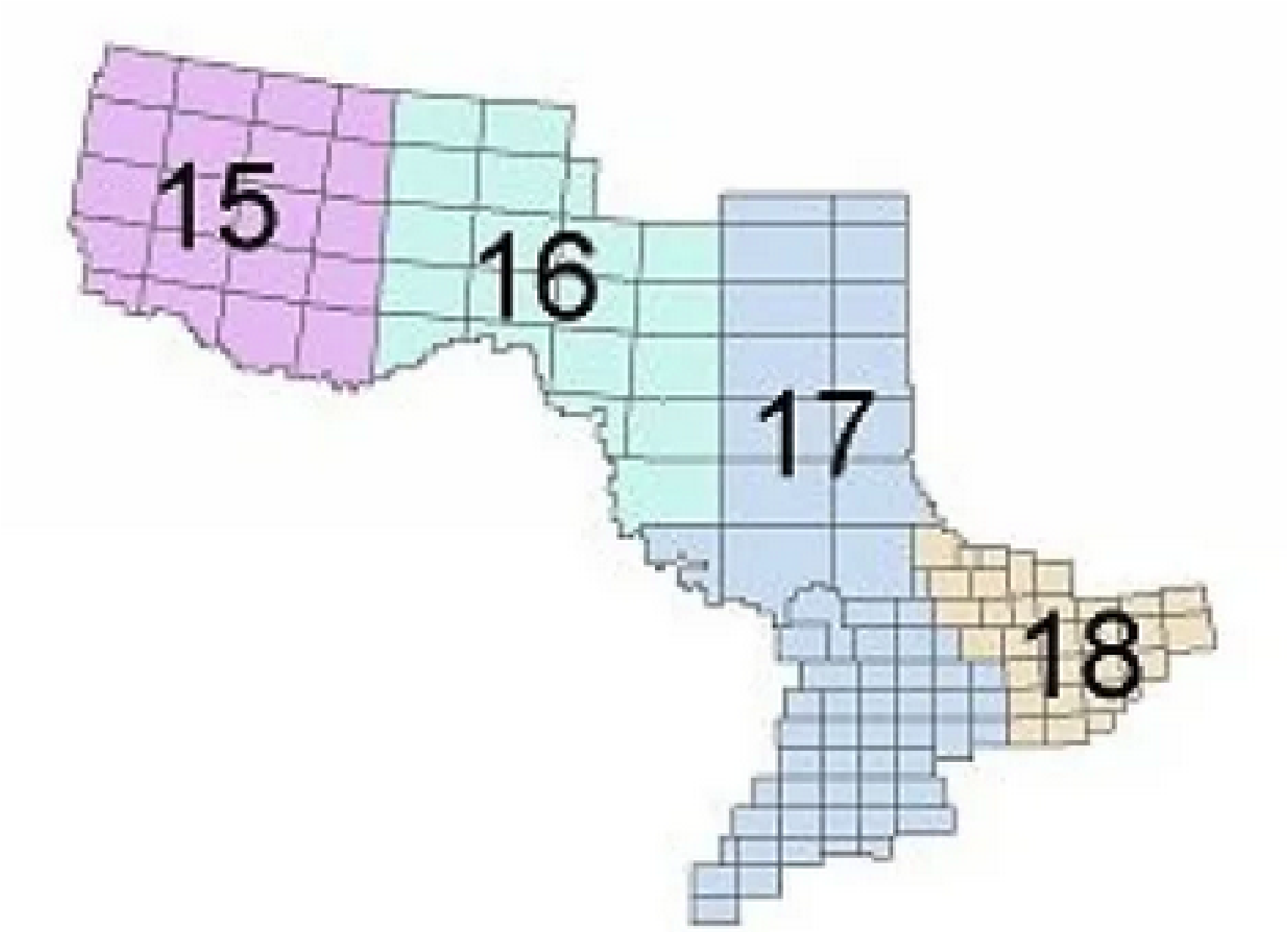


Figure 2 - UTM Zones and Central Meridians for Canada

UTM Projections

The UTM zones for Ontario



Steps for Projecting Latitude and Longitude

Concepts for projecting spatial data:

1. **Geographic Coordinates:** Latitude and longitude (in degrees) locate positions on the curved Earth. They are angular, not linear, so they cannot be used directly for flat distance/area measurements.
2. **Datum:** A mathematical model of the Earth's shape and size, providing a reference surface. Different datums (e.g., WGS84, NAD83) shift positions slightly but are critical for GPS and GIS accuracy.
3. **Coordinate Reference System (CRS):** Defines how the 3D Earth is represented in 2D. A CRS = *datum* + *projection*.
    - Example: EPSG:4326 (WGS84, geographic – lat/lon).
    - Example: EPSG:3857 (Web Mercator, projected – meters).
    - EPSG:32633 — WGS84 / UTM Zone 33N (meters, Northern Hemisphere).
    - EPSG:32733 — WGS84 / UTM Zone 33S (meters, Southern Hemisphere).

Datum First, Projection Second

- **Datum:** Defines the Earth's shape and reference ellipsoid.
  - Examples: WGS84, NAD83, NAD27.
  - Shifts positions depending on which model is used.
- **Projection:** Flattens the curved surface (defined by the datum) onto a 2D map.
  - Examples: Transverse Mercator, Lambert Conformal Conic, Web Mercator.
  - Requires the datum as input.
- **Order matters:**
  - First set the datum (what Earth model?).
  - Then apply the projection (how to flatten it?). Example: EPSG:4326 (WGS84 geographic) $\rightarrow$ EPSG:32617 (WGS84 UTM Zone 17N projected).
  - Wrong datum + correct projection = positional error.

Picking Projections

Picking a projection:

- The choice of projection systems depends on the objectives and scale of the map or analysis.
- For mapping small regions use a projection that minimizes distortion in a local area (e.g., Lambert Conformal Conic).
- For distance calculations use something that matches with the scale of your problem. Equidistant (azimuthal) projections are good, so are specific ones for a region.
- For area-based analyses use equal-area projections like Albers.

Projection Resources

- **PROJ**: The open-source library used in R (`sf`, `terra`, `proj4`), Python (`pyproj`), GDAL, QGIS, and PostGIS to handle coordinate transformations.
- PROJ documentation — authoritative list of supported projection methods.
- SpatialReference.org — community resource to search coordinate systems and export definitions.
- EPSG.io — search EPSG codes (official registry for coordinate reference systems).

Common EPSG Codes

- **Geographic (lat/lon, degrees)**
  - EPSG:4326 — WGS84 (global standard for GPS, lat/lon in degrees)
- **Web Mapping (projected, meters)**
  - EPSG:3857 — Web Mercator (used by Google Maps, OpenStreetMap, etc.)
- **UTM Zones (WGS84, meters)**
  - EPSG:326xx — UTM zone `xx`, Northern Hemisphere (e.g., EPSG:32617 = WGS84 / UTM Zone 17N)
  - EPSG:327xx — UTM zone `xx`, Southern Hemisphere (e.g., EPSG:32733 = WGS84 / UTM Zone 33S)
- **Regional Datums**
  - EPSG:4269 — NAD83 (North America, geographic lat/lon)
  - EPSG:26917 — NAD83 / UTM Zone 17N (North America, projected)

Spatial Data in R: `sf`

- Last week we saw how spatial data can be represented as coordinates with attributes (e.g., point-referenced or geostatistical data, areal data).
- In R, the `sf` package (**s**imple **f**eatures) allows us to:
  - Store and manage different types of spatial data (points, lines, polygons).
  - Perform spatial computations (distances, overlays, joins).
  - Apply spatial analyses more easily in a standardized framework.
- **Simple feature** = *geometry* (location on Earth) + *attributes* (properties). Example: Point(–79.4, 43.7) with attribute `name = "Toronto"`.
- **The Open Geospatial Consortium** provides the **OGC standard:** A simple feature has both spatial and non-spatial attributes. Geometry is 2D and defined by vertices with linear interpolation.
  `https://www.ogc.org/standard/sfa/`

Spatial Data in R: `sf`

- The `sf` package supports the 7 main simple feature types (OGC standard).

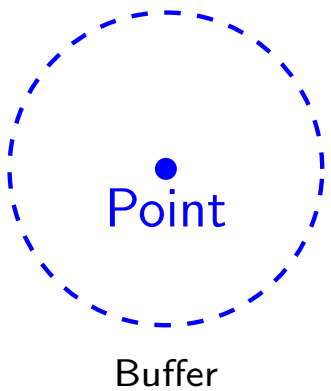
Type	**Example / Description**
`POINT`	Single coordinate (e.g., city location)
`MULTIPOINT`	Multiple coordinates (e.g., bus stops)
`LINESTRING`	Connected points (e.g., road segment)
`MULTILINESTRING`	Set of lines (e.g., road network)
`POLYGON`	Closed shape with area (e.g., park boundary)
`MULTIPOLYGON`	Multiple polygons (e.g., islands, districts)
`GEOMETRYCOLLECTION`	Mixed types (rare in practice)

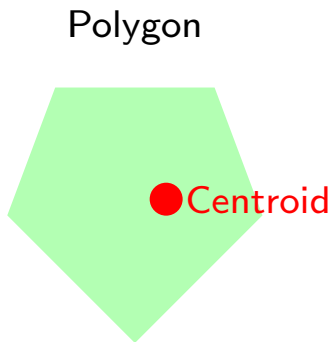
Spatial Computations

- Spatial functions are fundamental for working with geographic data.
- They allow you to:
  - Measure distances
  - Analyze spatial relationships
  - Transform and manipulate geometries
- These operations enable complex queries and data-driven decisions in GIS and spatial analysis.

Spatial Computations: Proximity and Measurement

- `st_buffer()` — Create buffers around geometries.
- `st_distance()` — Measure distance between geometries.
- `st_centroid()` — Compute centroid of a polygon.

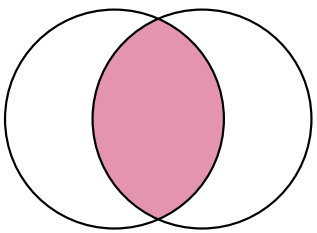




Spatial Computations: Overlay Operations

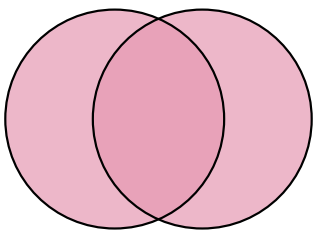
Intersection

`st_intersection()` Finds overlapping area between geometries.



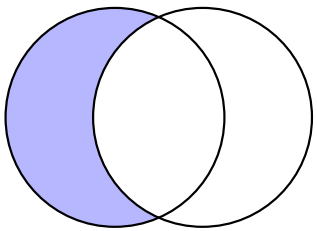
Union

`st_union()` Merges geometries into a single combined footprint.



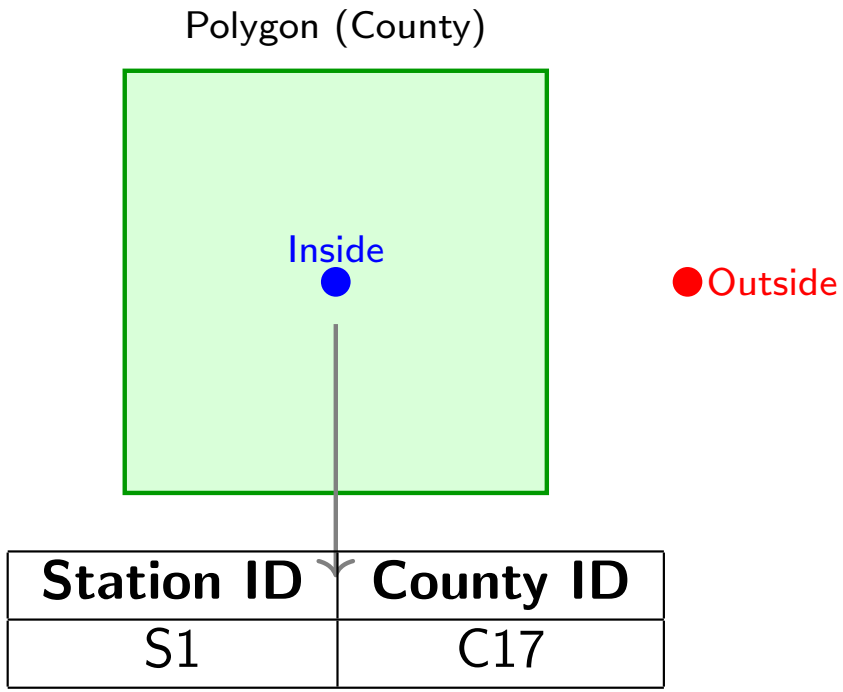
Difference

`st_difference()` Subtracts one geometry from another.



Spatial Computations: Relationships and Joins

- `st_contains()` / `st_within()`
  - Test whether one geometry is inside another.
  - Example: A county polygon contains a city point.
  - Useful for spatial queries and filtering.
- `st_join()`
  - Joins attribute tables based on spatial relationships.
  - Example: Assigning each monitoring station (points) the county it falls within (polygons).
  - Works with containment, intersection, and other spatial predicates.



Station ID	County ID
S1	C17

Spatial Computations

Let's go through spatial2.Rmd to use the `sf` package to look at different spatial types, projections using `st_transform`, and do various spatial computations.