Introduction to spatial data in R for paleontologists

Ádám T. Kocsis

Friedrich-Alexander-Universität Erlangen-Nürnberg

2025-08-22

GeoZentrum Nordbayern

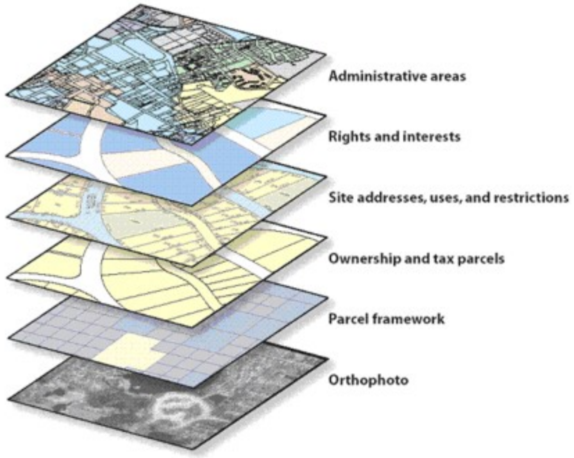
Basic Concepts

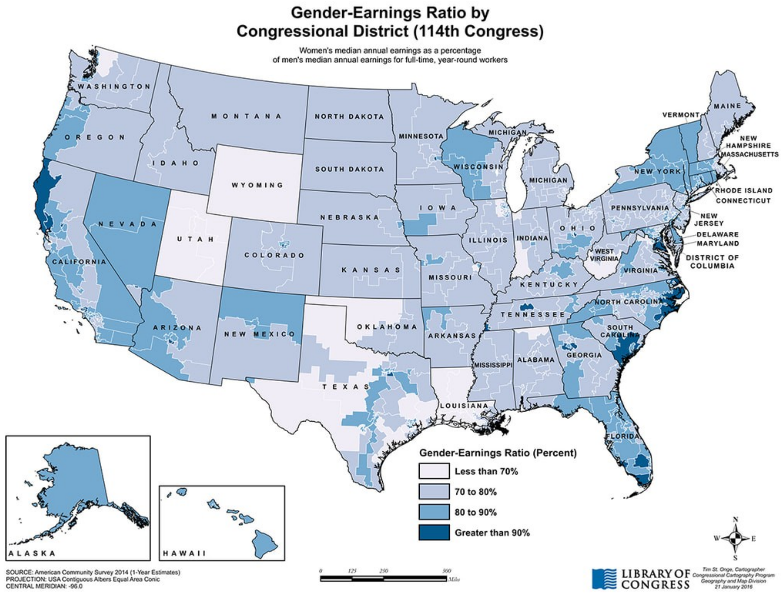
Where, what, why?

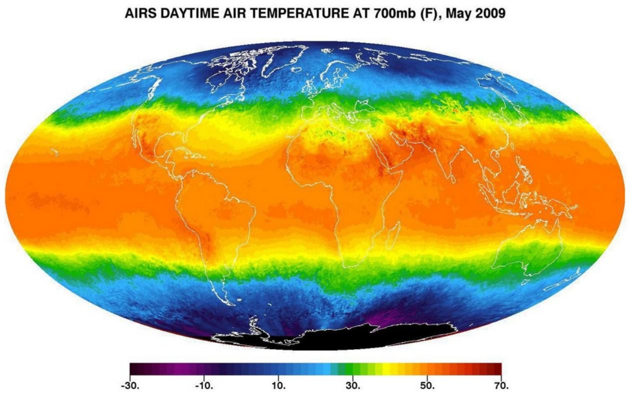
GIS basics

G(eographic) I(nformation) S(ystems/oftware) are everywhere

- Science
- Civil Engineering
- Aeronautics
- Military
- Economics
- etc.



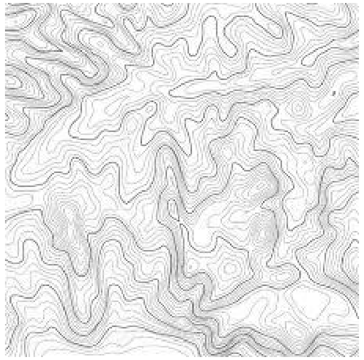




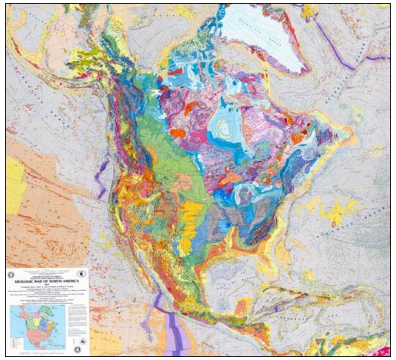
Vital in Earth Sciences

G(eographic) I(nformation) S(ystems) are everywhere

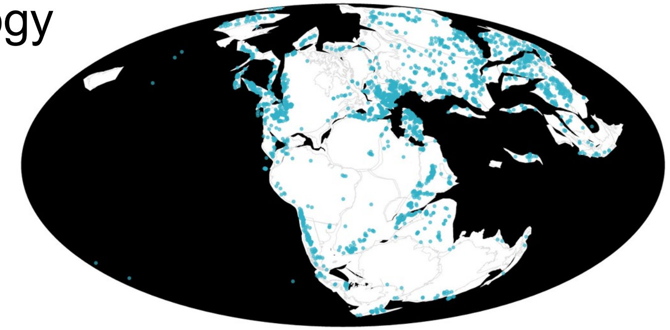
- topography



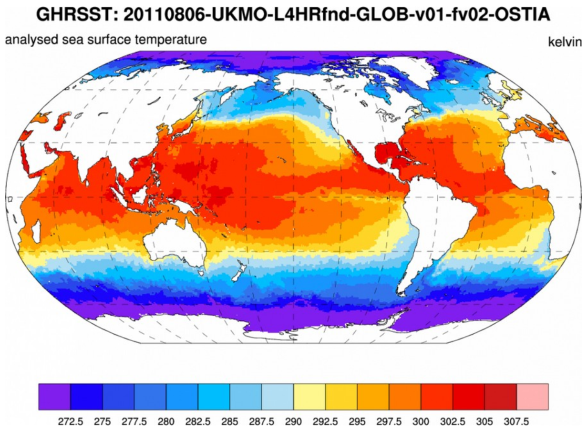
- geology



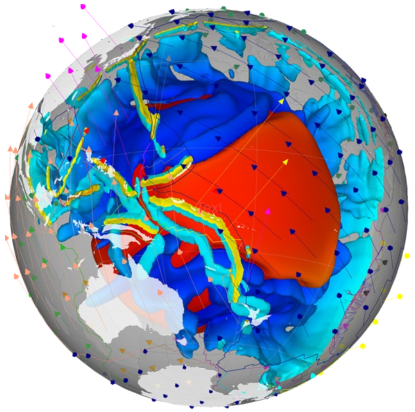
- paleontology



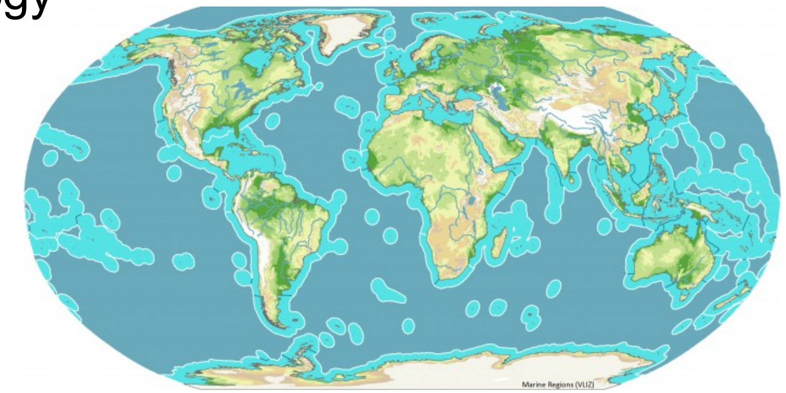
- climate



- geophysics



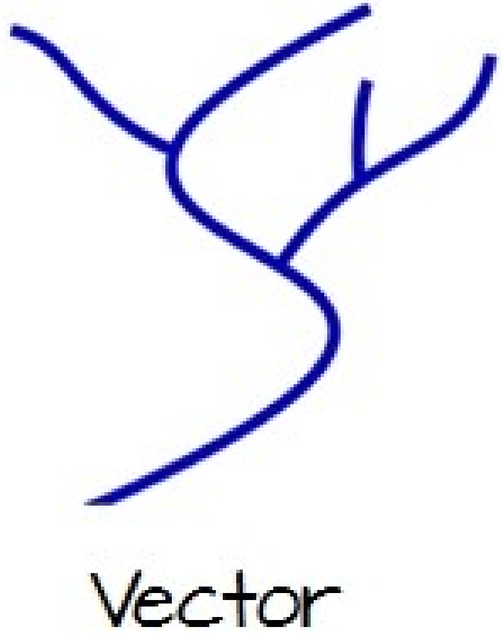
- ecology



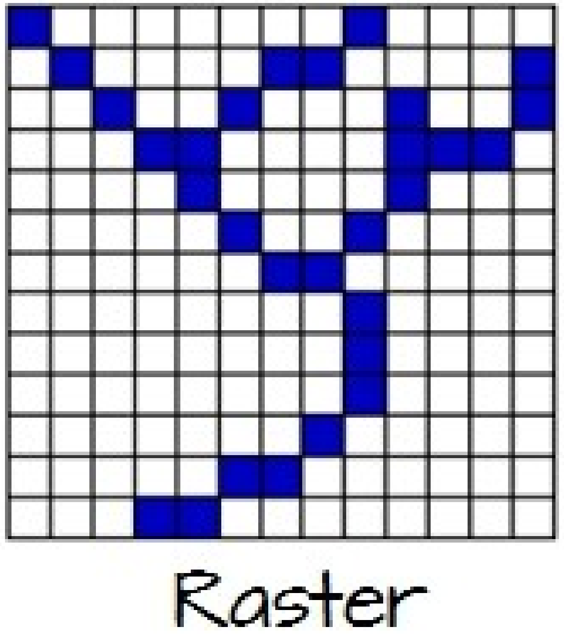
Basic data types

- Similar to computer graphics
- Sometimes in 3D

Object view



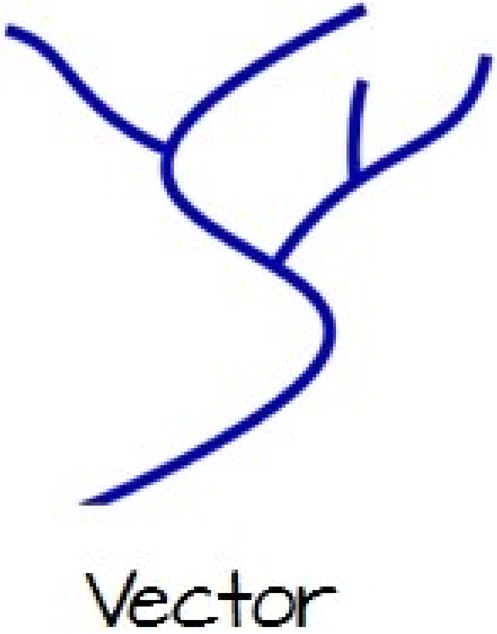
Field view



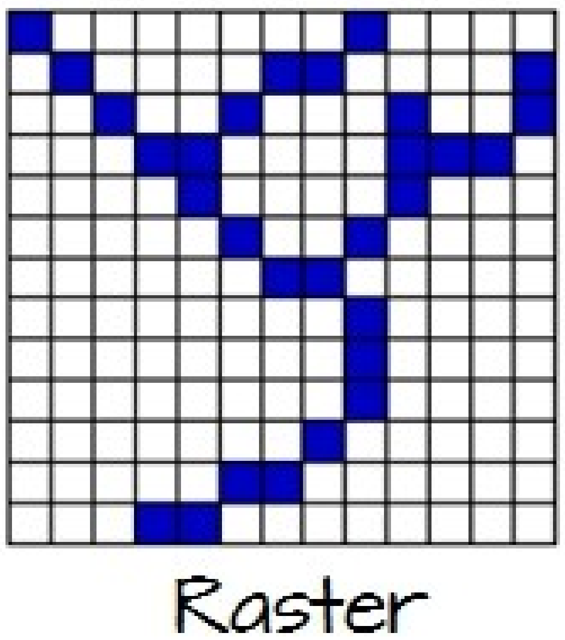
Basic data types

- Similar to computer graphics
- Sometimes in 3D

Object view



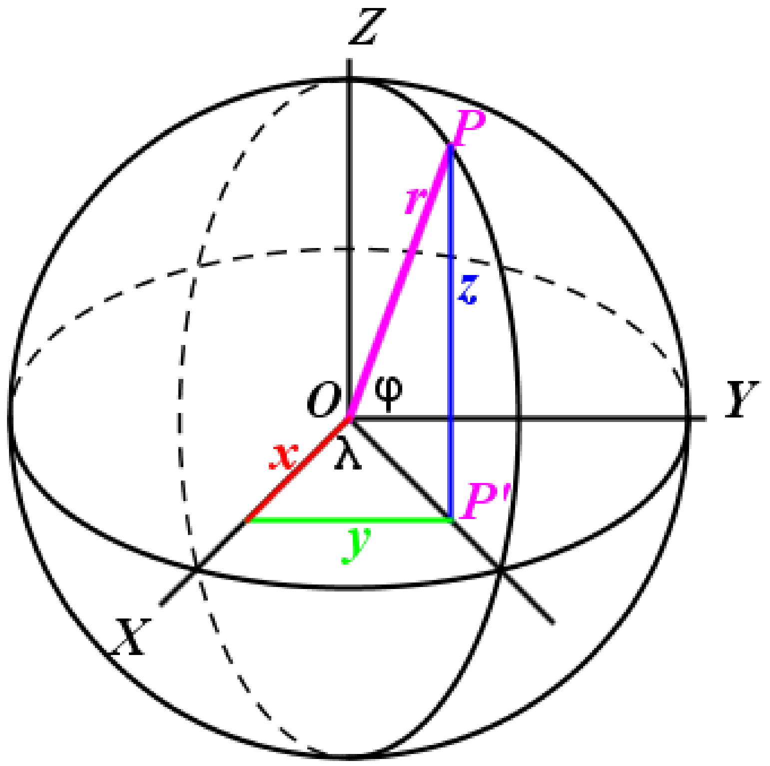
Field view

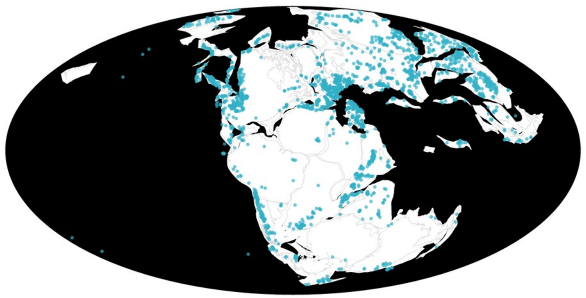


There is one twist:

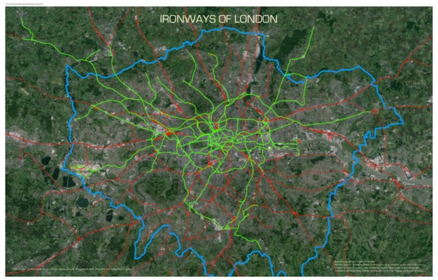
Vector data

- Structure is defined with points that have coordinates
- Spherical coordinates, Cartesian too, or projection-specific
- Additional attributes are rendered to the entities
- 3 primary kind

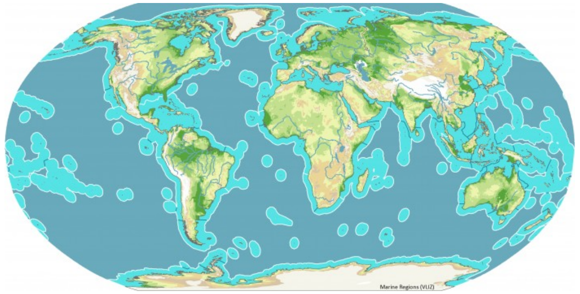




Points



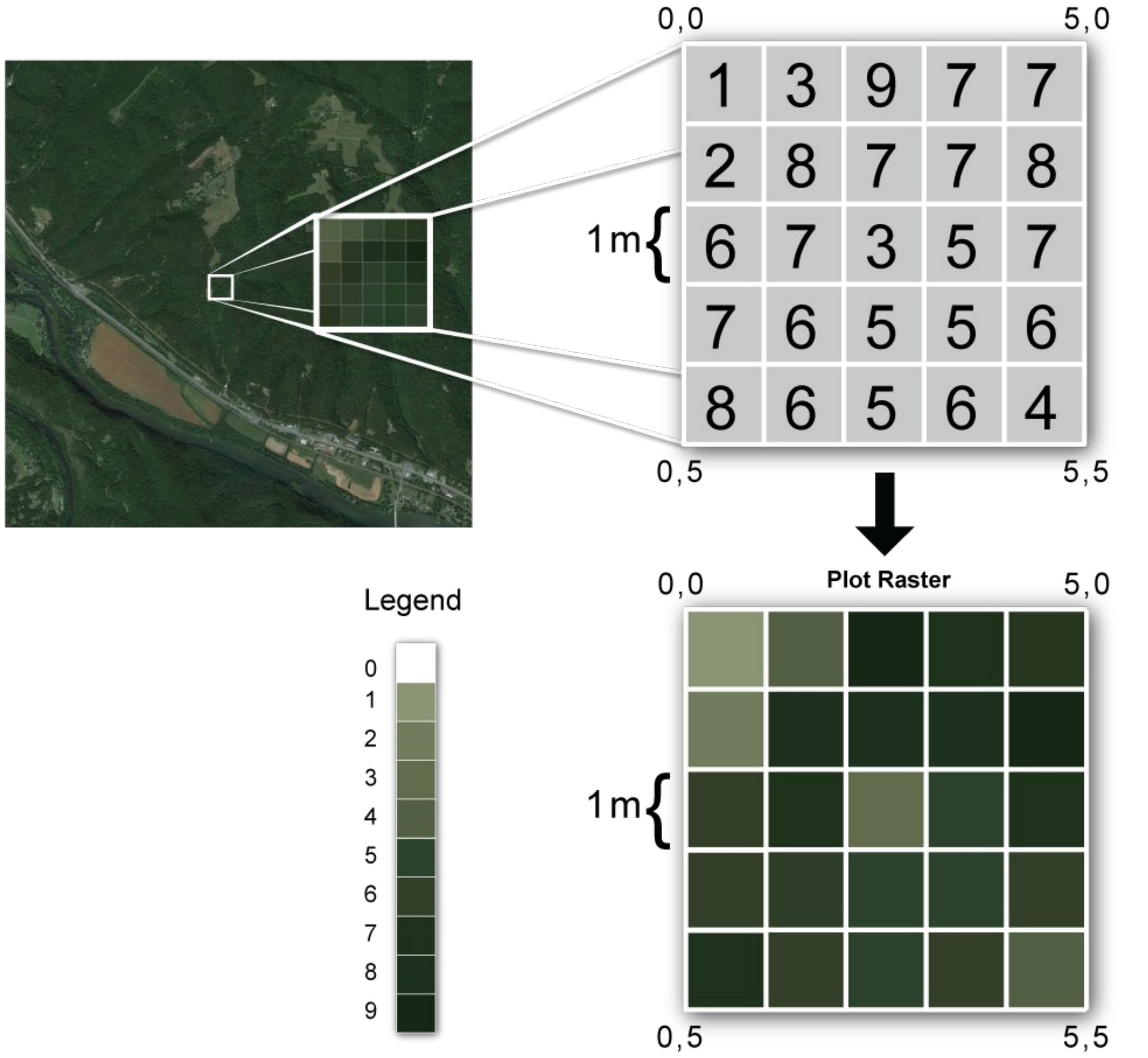
Lines



Polygons (shapes)

Raster data

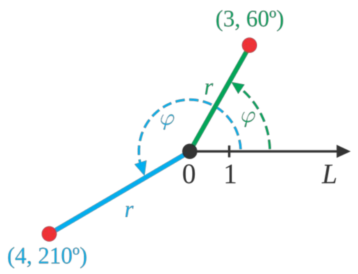
- Structure is defined by a grid, data is a lattice of values
- The field view of Earth
- Similar to raster graphics, but with georeferencing



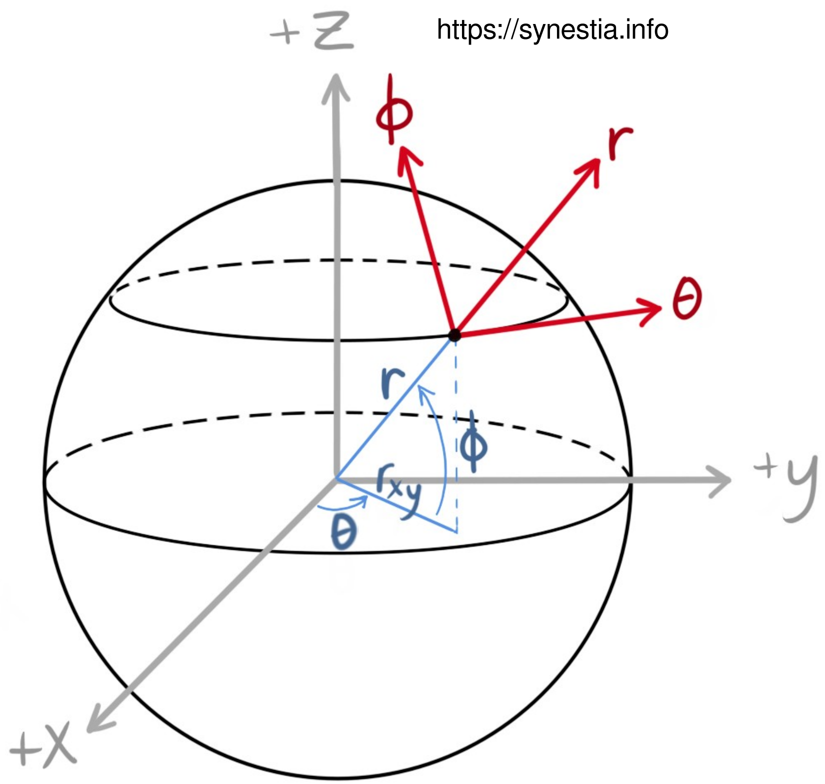
Coordinate Reference Systems

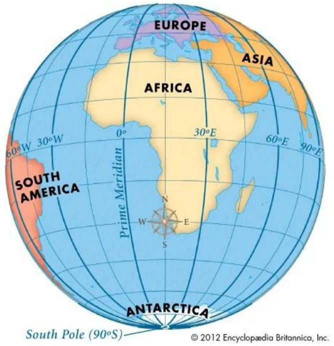
Geographic coordinates

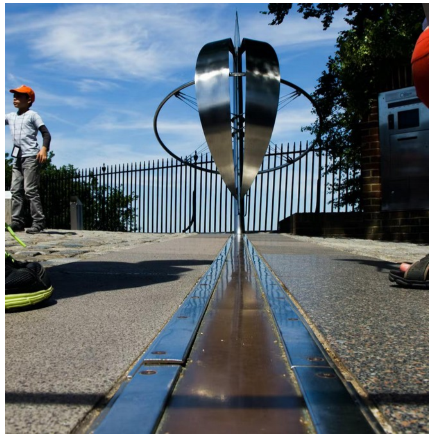
- Based on polar coordinates
- φ: latitude
  - [-90°, 90°], equator : 0°
- θ: longitude
  - [-180°, 180°], 0°: prime meridian
- r: elevation



https://synestia.info



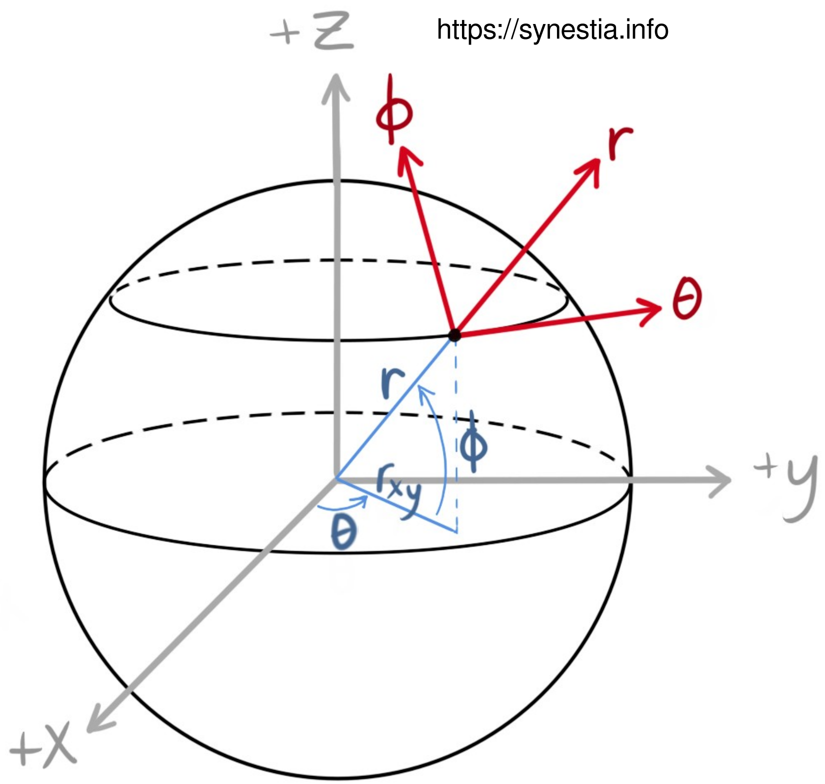




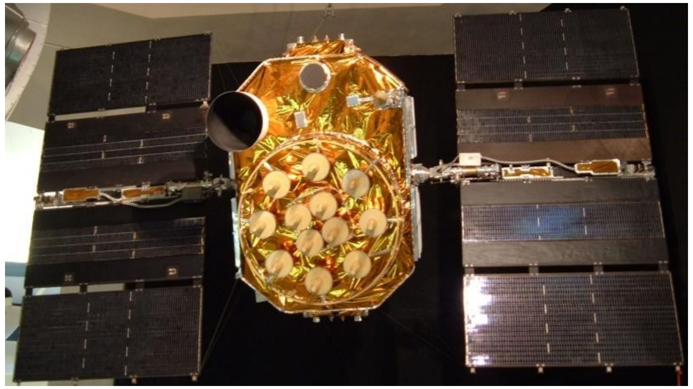
Coordinate Reference Systems

Geocentric coordinates

- x, y, z, Earth-Centered Coordinate System
- Based on center of mass, equator and PM
- Used for satellite tracking and navigation (e.g. GPS)
- WGS84 datum



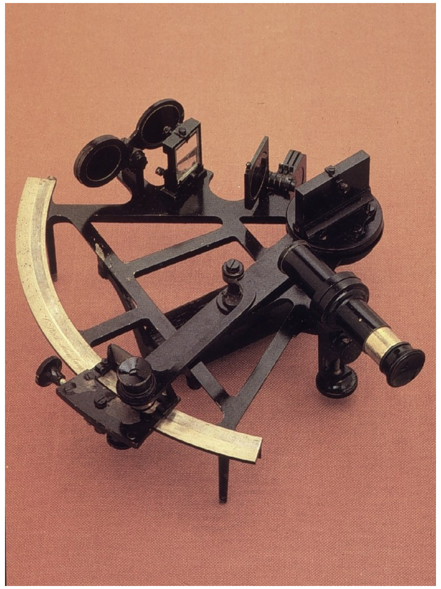
https://synestia.info

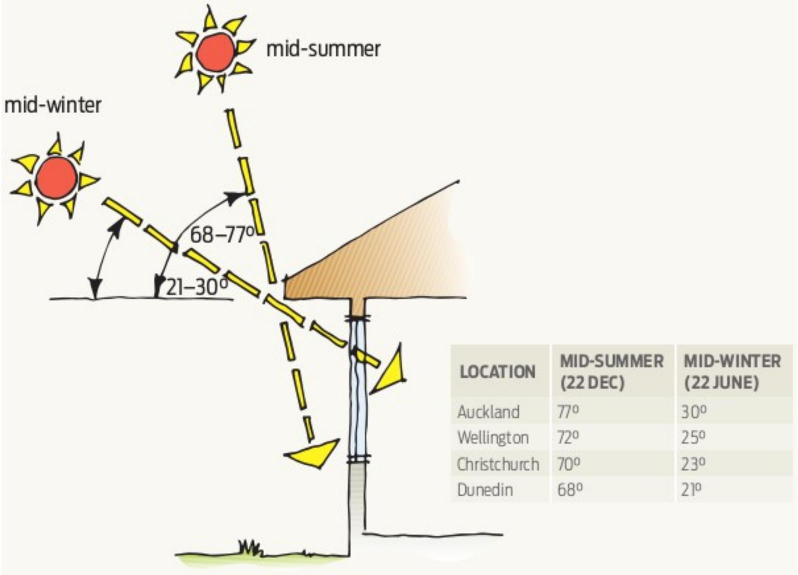


Coordinate Reference Systems

Projection based coordinate systems

- Surface: a 2-dimensional map
- Traditionally: Cartography
- Especially nautical applications

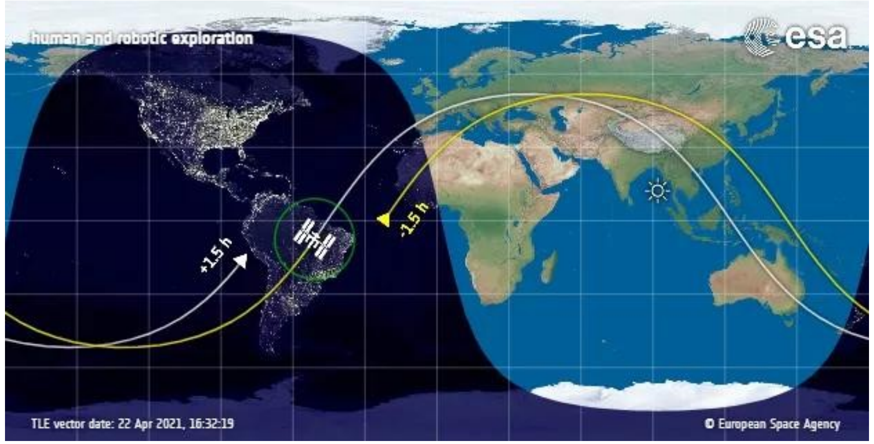




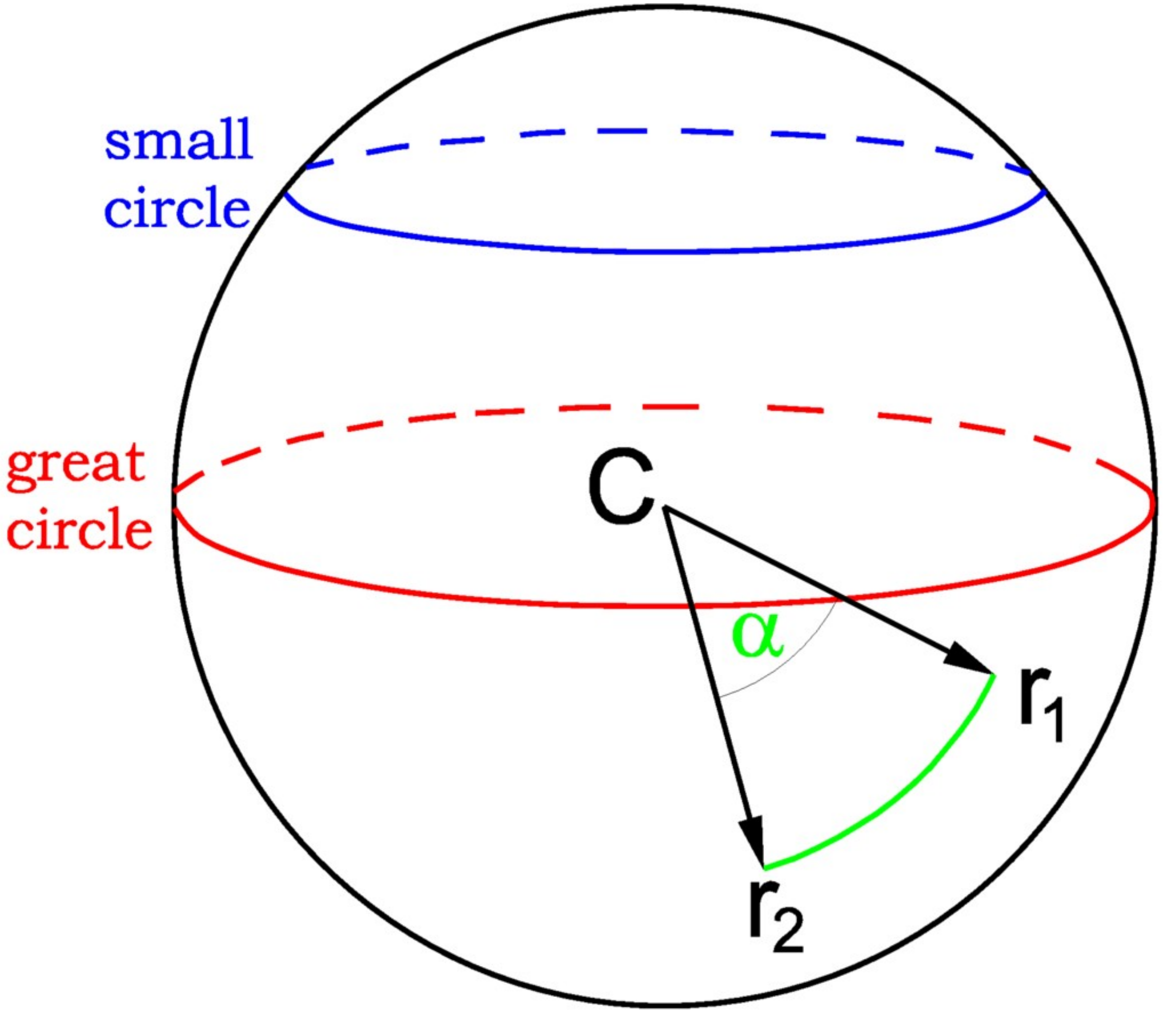
LOCATION	MID-SUMMER (22 DEC)	MID-WINTER (22 JUNE)
Auckland	77°	30°
Wellington	72°	25°
Christchurch	70°	23°
Dunedin	68°	21°

Spheres and Circles

- There are no straight lines
- All points are connected by arcs, either along a great or small circles

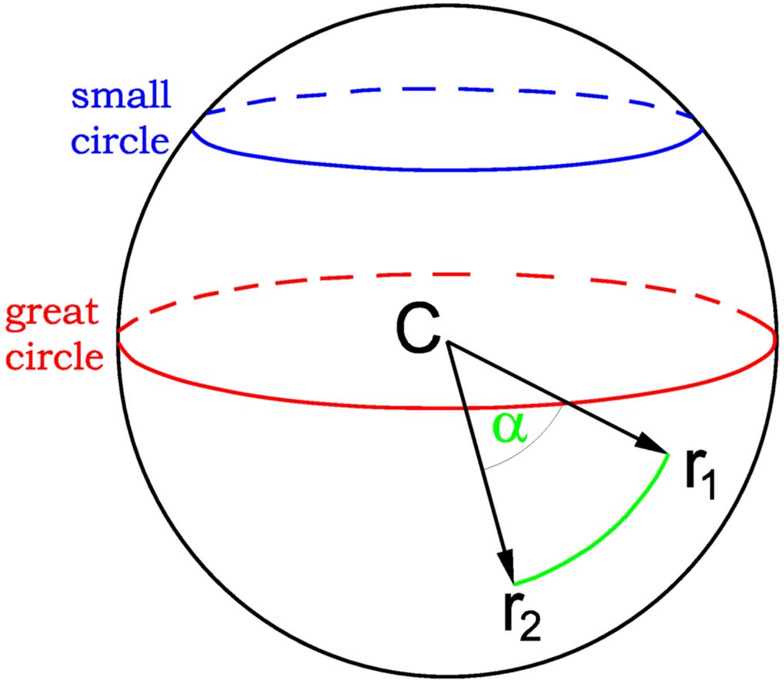


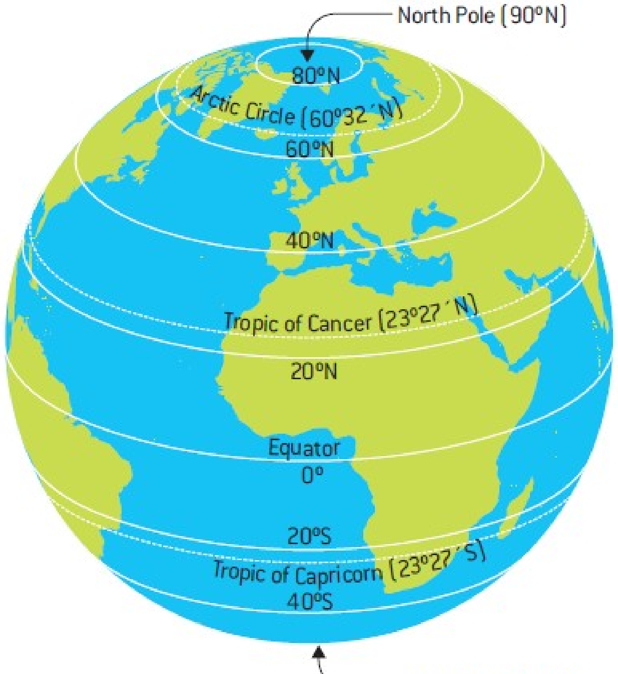
Path of the ISS



Latitude [-90,90˚]

- Latitudinal circles: **parallels**
- The only great circle is the equator!
- Length decrease with a cosine function of latitude
- 10° wide **zone** has different area at different latitudes
- Direction: **zonal (u)**
- Notable circles:

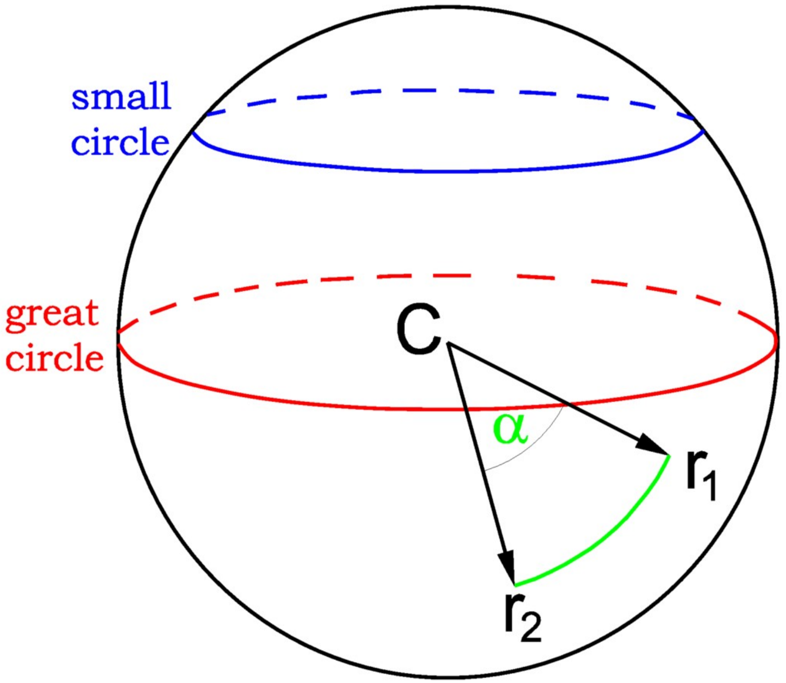




The parallels of latitude

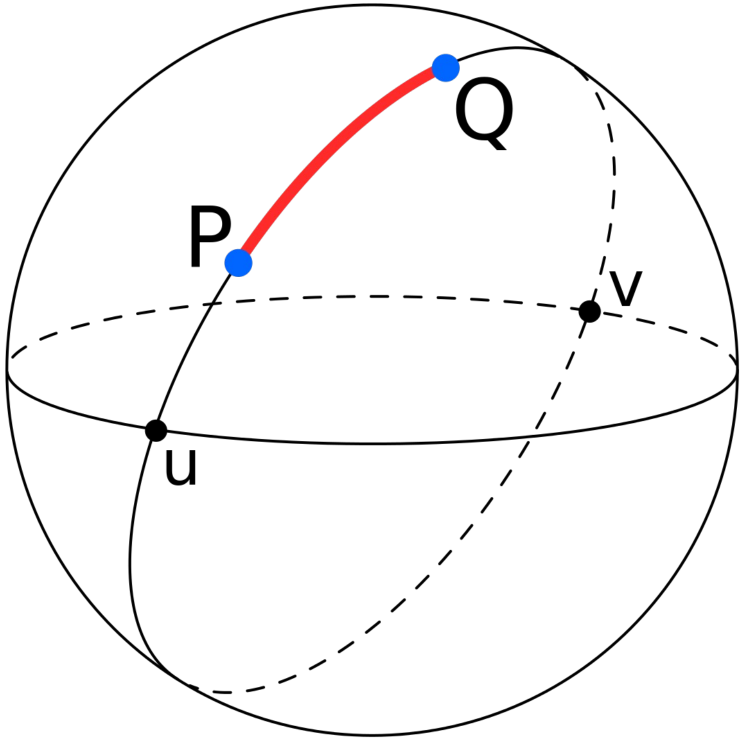
Longitude [-180,180˚]

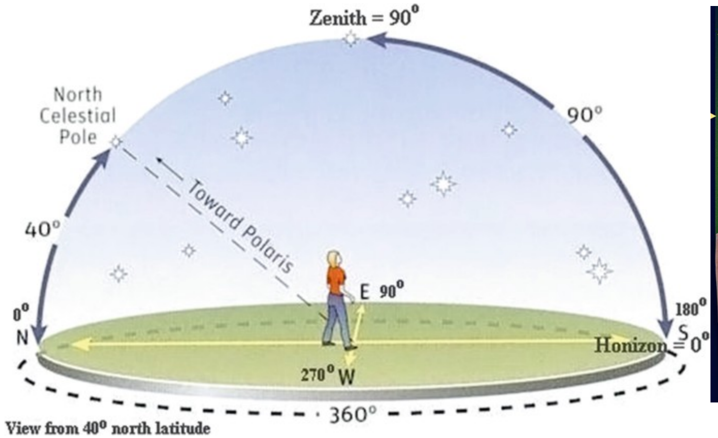
- Latitudinal circles: meridians
- All of them are great circles!
- Their distance decreases at the poles
- Direction: **meridional (v)**
- Notable circle:

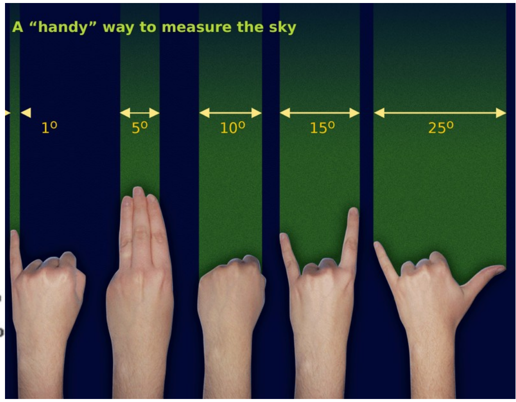


Distances

- Two points define a great circle
- Shortest distance between them is a great circle arc (Great Circle Distance)
- Distance on a surface of a sphere is simplest **in angles (degrees)**
- For practicality we give this in distance:
- 1° (longitude) ~ 111.1 km
- 1’ (longitude) = 1 nautical mile ~ 1.8km

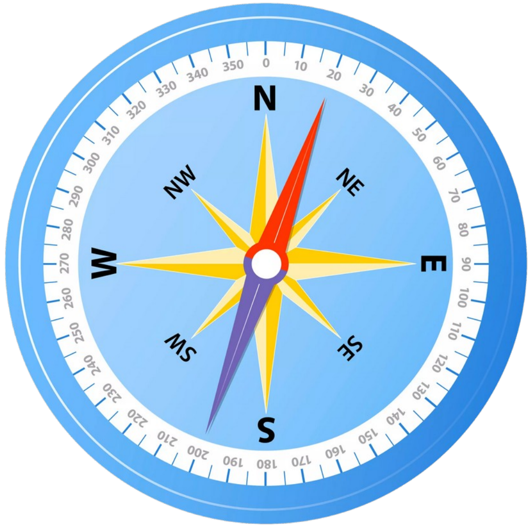






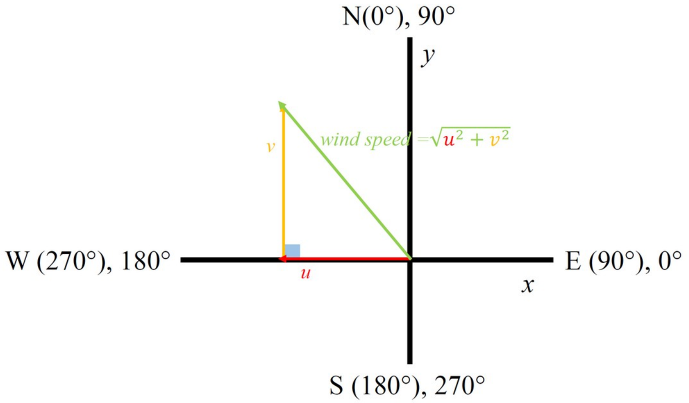
Directions

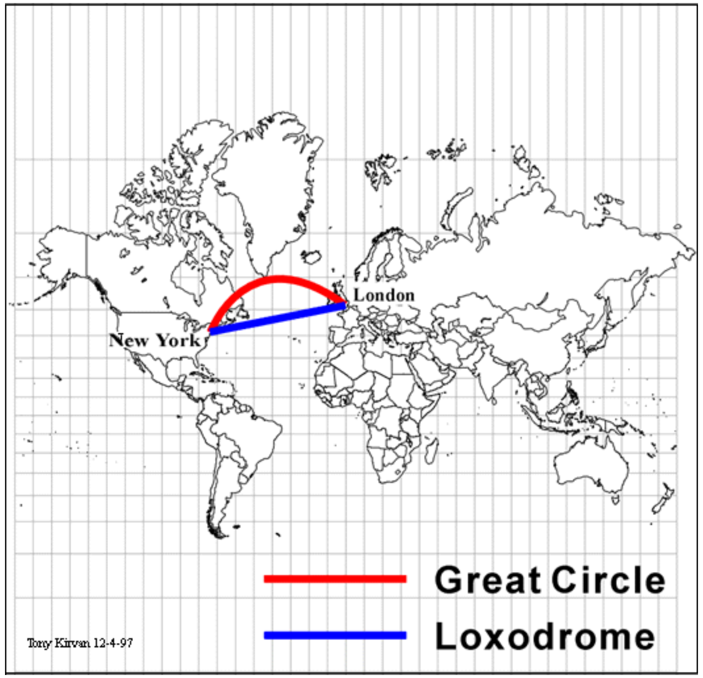
- Heading: angular distance to north (azimuth)
- Going straight: great circle
- Constant change of heading! (even without the true vs. magnetic difference)



Vectors

- Given with a) initial heading and magnitude or b) as zonal and meridional components

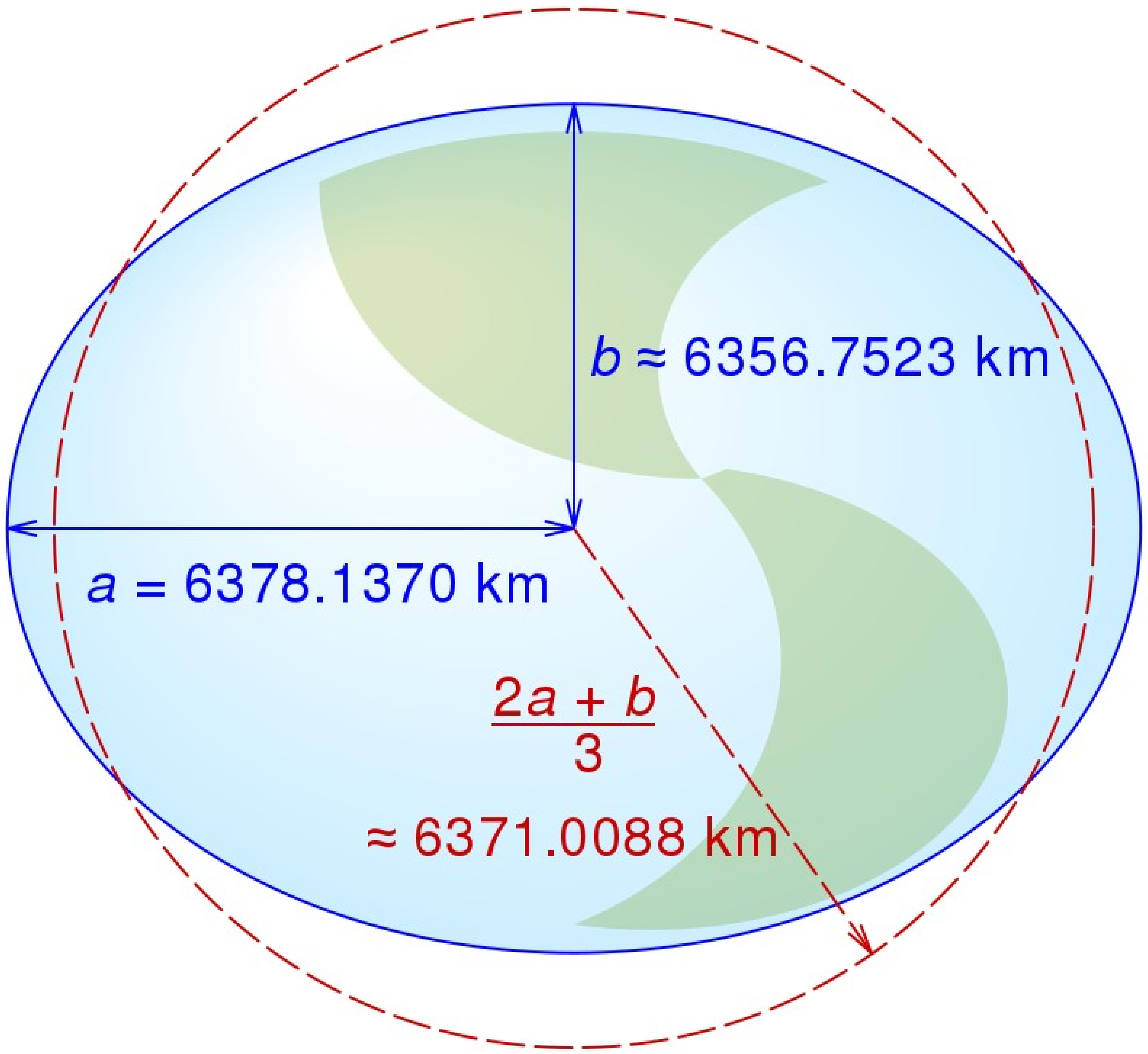




Mercator projection

The reference ellipsoid

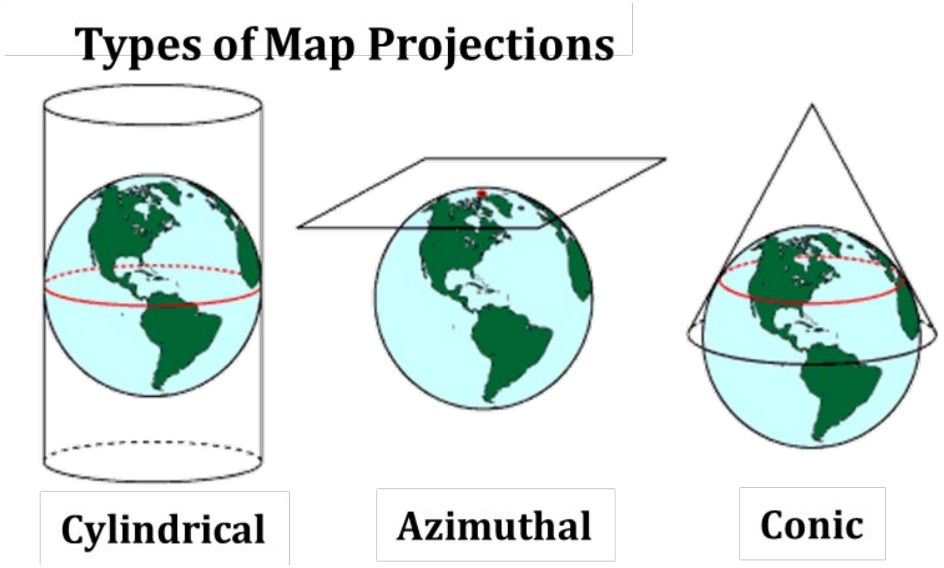
- Approximates the geoid
- Flattened rotational ellipsoid
- Used as a basis for elevation



Coordinate Reference Systems

Map projections

- There is dozens of them
- Complex transformations of spherical coordinates
- They all distort the globe in some sense:
  - conserve area and distort angles *(equal-area projection)*
  - distort area and conserve angles *(conformal projection)*
  - distort both area and angles



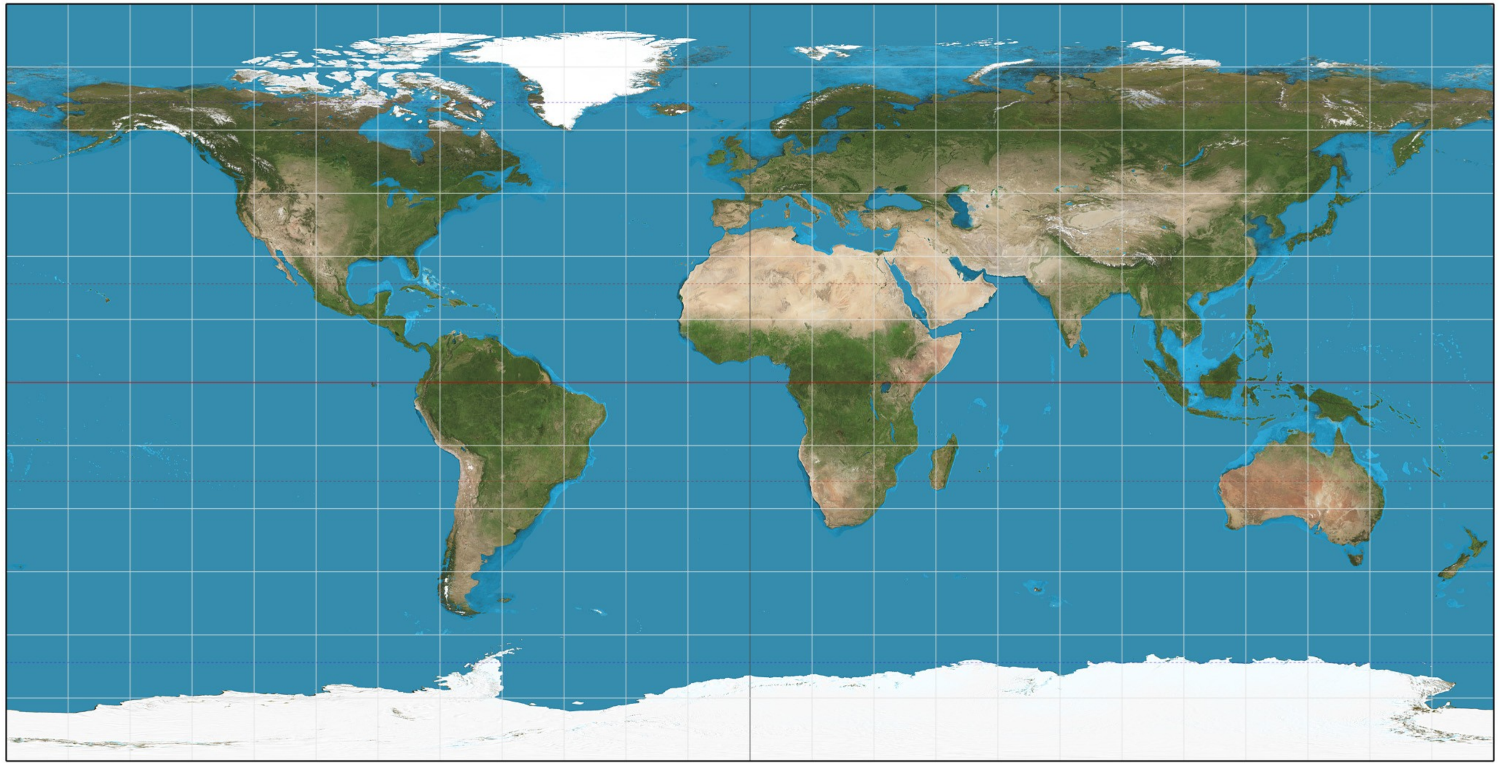
General types...

The True size of.. :
https://www.thetruesize.com

Coordinate Reference Systems

e.g. Equirectangular Projection

- Longitude (x) – Latitude (y)
- Centered around (0, 0)
- Plate Carée
- Most frequently used for visualization of global-scale data
- Distorts everything



Coordinate Reference Systems

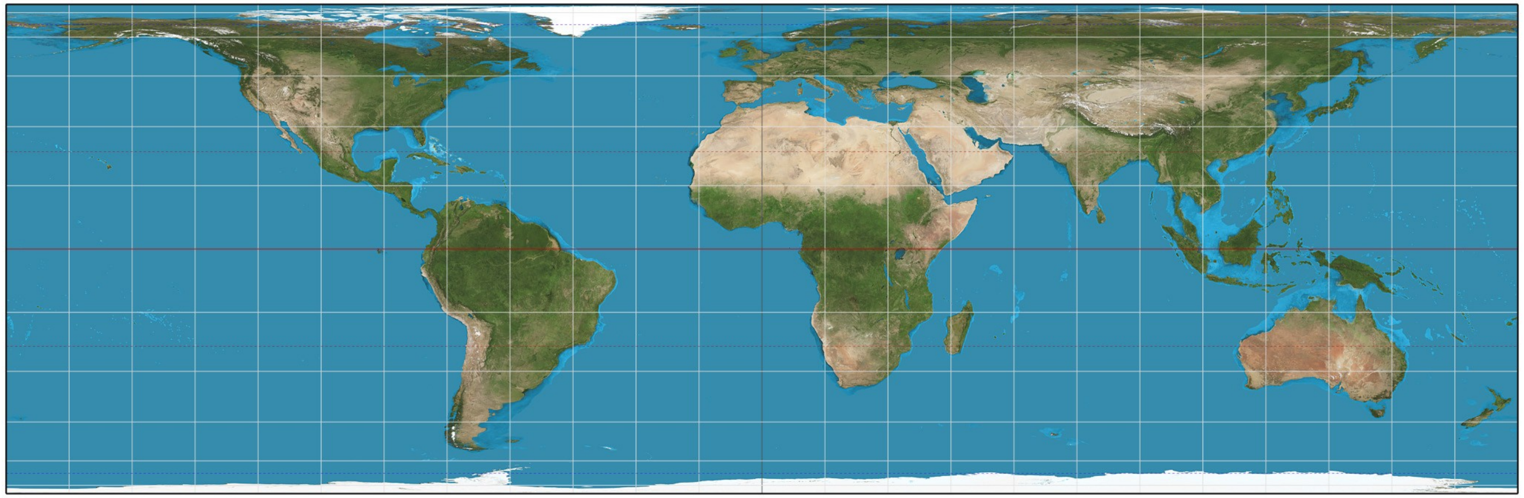
e.g. Mercator projection

- Conformal
- (nautical use)
- Highly distorts areas
- Cannot represent poles
- Variant: Google Earth

Coordinate Reference Systems

e.g. Lambert's Cylindrical EA Projection

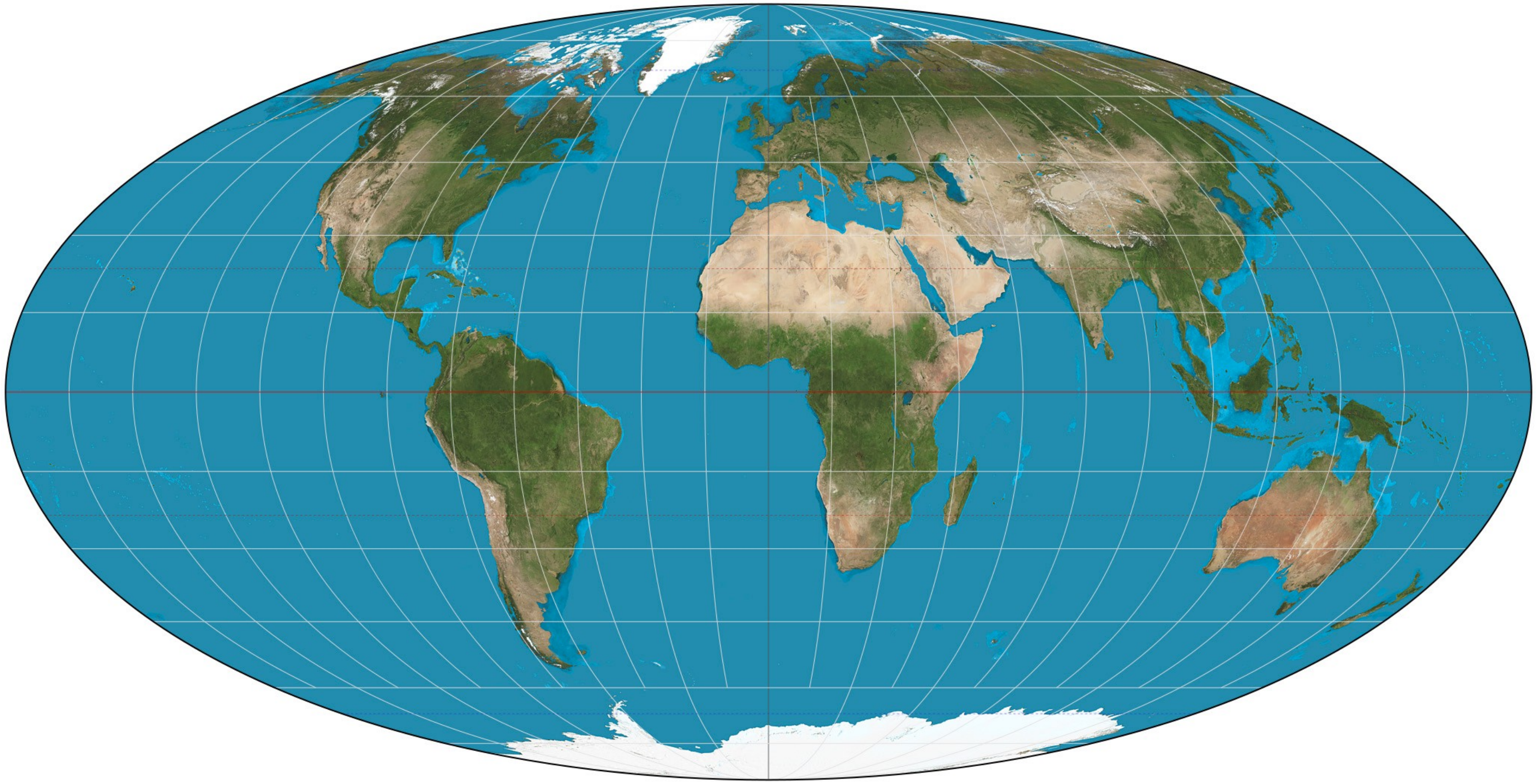
- Equal-area
- Highly distorts angles latitudinally



Coordinate Reference Systems

e.g. Mollweide Projection

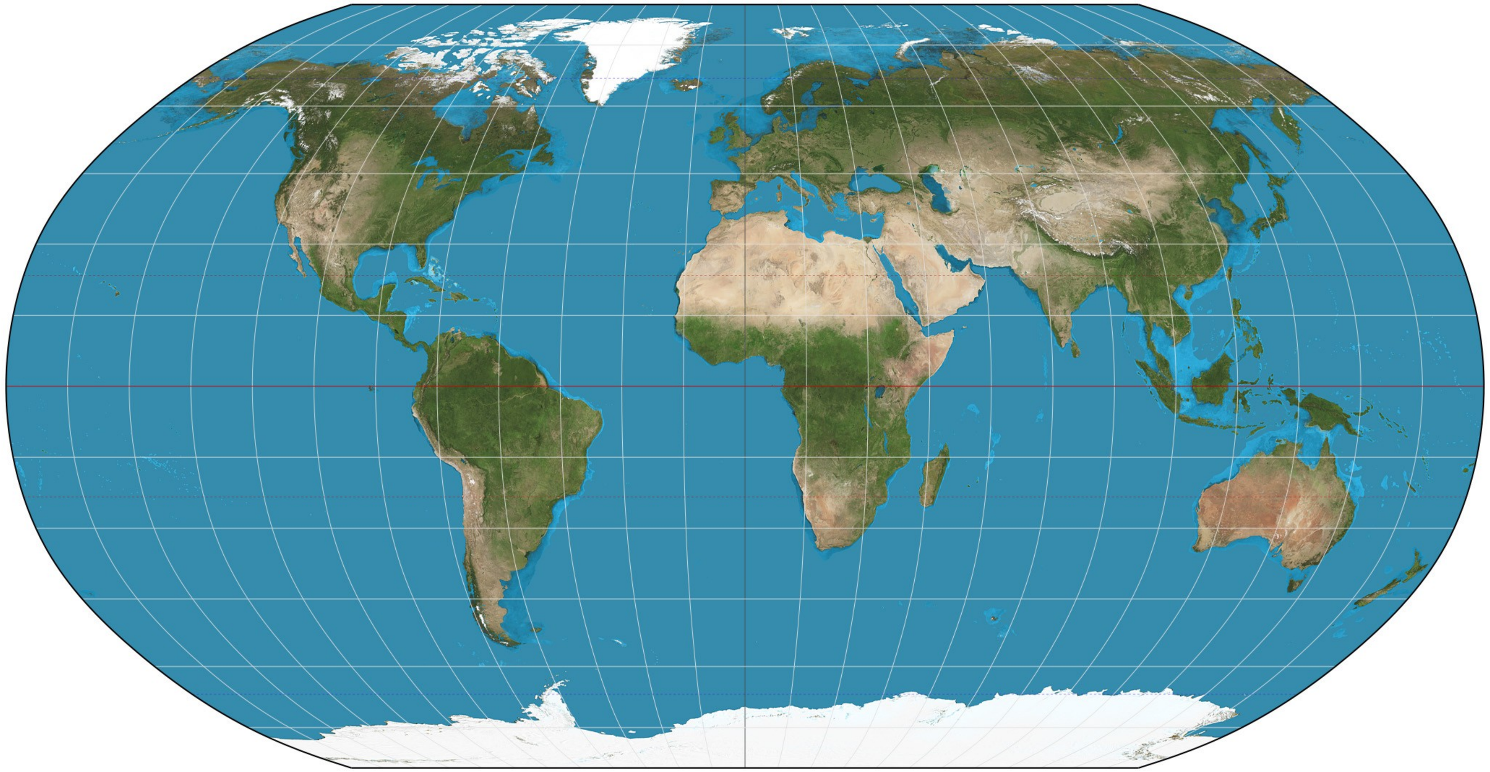
- Equal-area
- Highly distorts angles longitudinally



Coordinate Reference Systems

e.g. Robinson Projection

- Distorts both angles and areas
- Relatively representative
- Promoted by NatGeo and now used by IPCC



Coordinate Reference Systems

Representation

- Older: PROJ.4 string: “+proj=moll”
- Well known text WKT2 string

```
PROJCS["World_Mollweide",
    GEOGCS["WGS 84",
        DATUM["WGS_1984",
            SPHEROID["WGS 84",6378137,298.257223563]],
        PRIMEM["Greenwich",0],
        UNIT["Degree",0.0174532925199433]],
    PROJECTION["Mollweide"],
    PARAMETER["central_meridian",0],
    PARAMETER["false_easting",0],
    PARAMETER["false_northing",0],
    UNIT["metre",1],
    AXIS["Easting",EAST],
    AXIS["Northing",NORTH],
    AUTHORITY["ESRI","54009"]]
```

- EPSG registry ID (epsg.io) : ESRI:54009

EPSG

GEODETIC PARAMETER DATASET

Managed by IOGP's Geomatics Committee

Implementation

Where, what, why?

GIS tools: Why R?

- Automatized manipulation and extraction of data
- Larger quantity of work then with a GUI
- Python is much more powerful
- Integration with other analyses
- Statistical analyses

The Spatial ecosystem

- Multiple R extension packages form the backbone
- The heart: Open Source Geospatial Foundation
- Recipe:
    - **Open source libraries for calculations + R interface**

- R packages hook on the APIs of these libraries
- + extra calculations in compiled code (C, C++, Fortran)

Open Source Libraries/Projects

GDAL: Geodetic Data Abstraction Library

- Definitions of data formats
- Translation between data formats
- https://gdal.org/

Open Source Libraries/Projects

PROJ

- Coordinate transformation
- Definition of Coordinate Reference Systems (CRS) and translation
- Current version is 9.2.1 (June 2023)
- R packages rely on older versions (PROJ4 or PROJ6)

Open Source Libraries/Projects

GEOS: Geometry Open Source

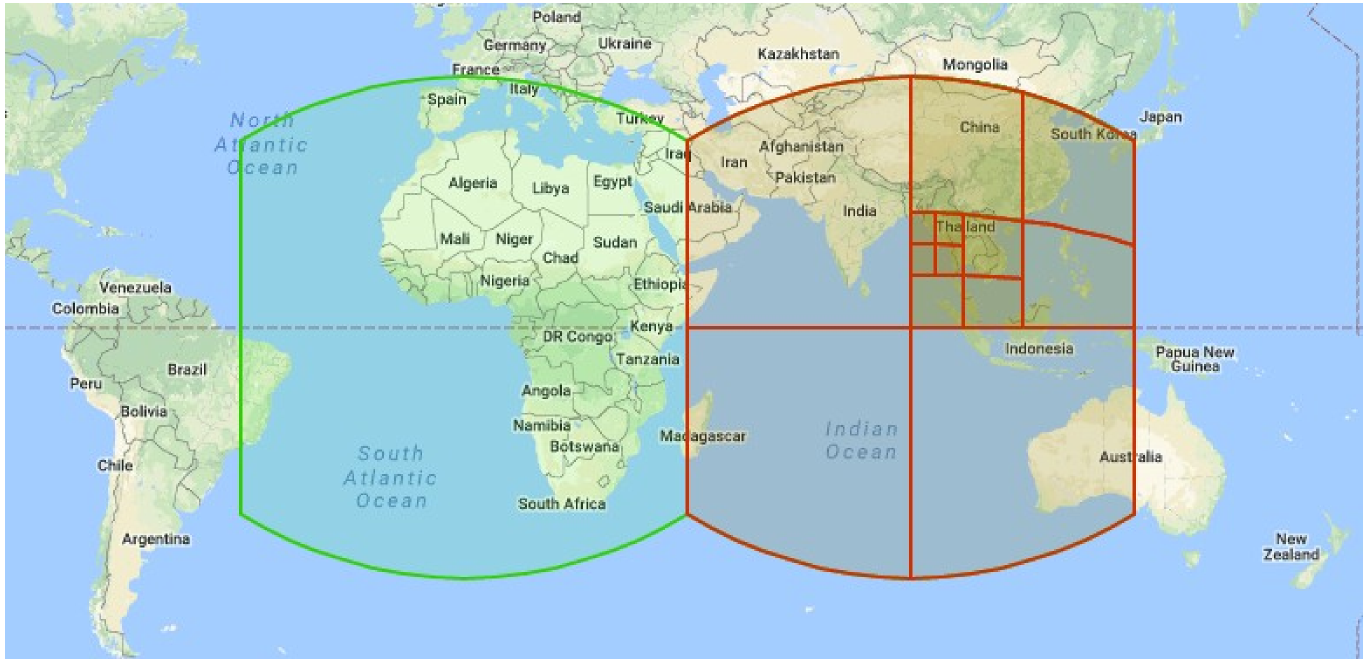
- Computational geometry especially towards the manipulation of spatial data
- http://libgeos.org/

GEOS

Open Source Libraries/Projects

S2 Geometry

- Spherical computations (Google)
- http://s2geometry.io/



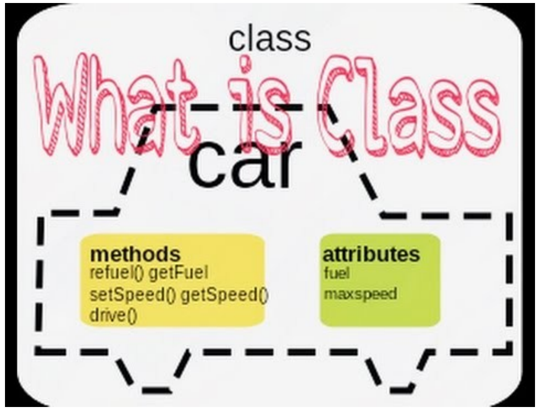
The R packages (two generations)

- Only some are essential, they are on the CRAN
- They are using the libraries
- Class definitions and basic methods
- Lots of packages build on these...

Lot of dependencies, slow development

Generation shift

What is a class?



https://www.youtube.com/watch?v=XqrkcO42DI8

Packages

	old	new
vector	sp, rgdal, rgeos	**sf, terra**
raster	raster	**terra, stars**

+ others: e.g. geosphere

Installing all of them

Focus on the new but...

- On Windows this is trivial, you can get all from the CRAN with install.packages()
- On Mac (and Linux), it is recommended to install them one at a time. They will a) either tell you what libraries need to be installed so they can function properly, 2) only indicate errors and then we have to google :)
- The most important ones:

```
install.packages(c("terra", "sf", "geosphere"))
```