Introduction to spatial data in R for paleontologists

Ádám T. Kocsis

Friedrich-Alexander-Universität Erlangen-Nürnberg

2022-08-29
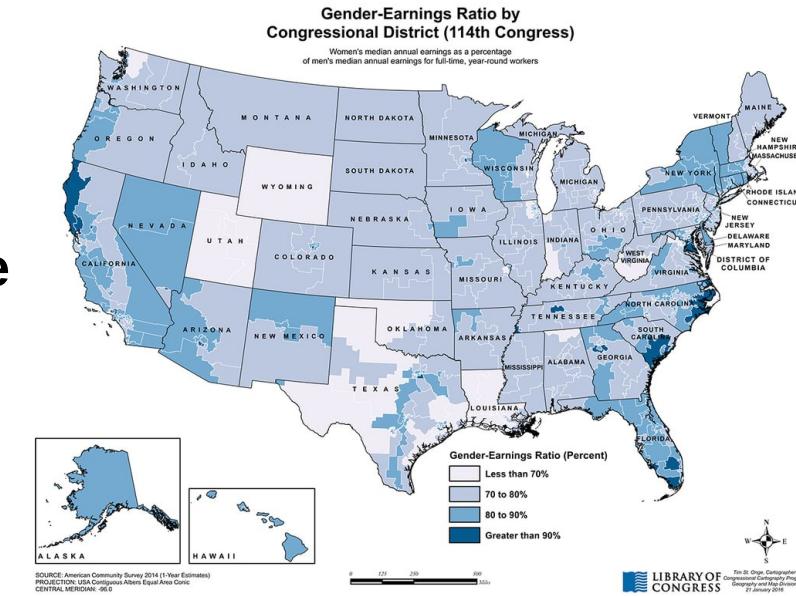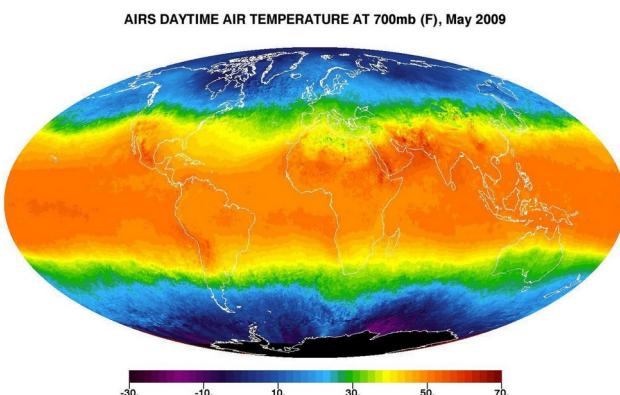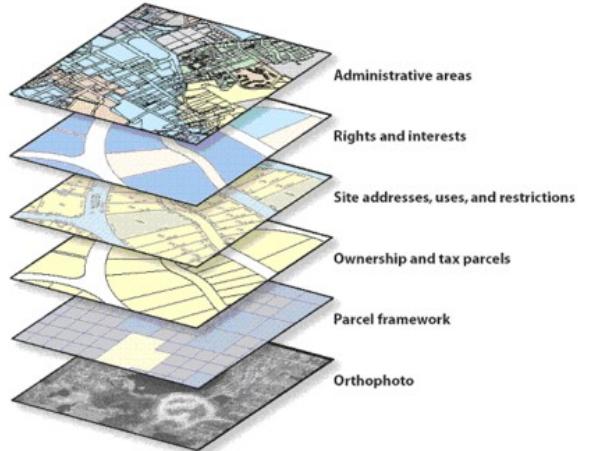
Basic Concepts

Where, what, why?

GIS basics

**G(eographic) I(nformation) S(ystems/oftware) are everywhere**
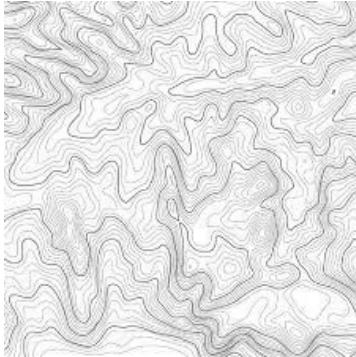
- Science
- Civil Engineering
- Aeronautics
- Military
- Economics
- etc.

Gender-Earnings Ratio by Congressional District (114th Congress)

AIRS DAYTIME AIR TEMPERATURE AT 700mb (F), May 2009

sources: FlightConnections.com / Google

Vital in Earth Sciences

**G(eographic) I(nformation) S(ystems) are everywhere**
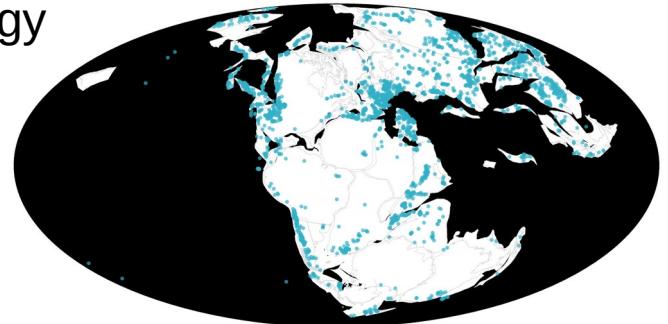
- topography
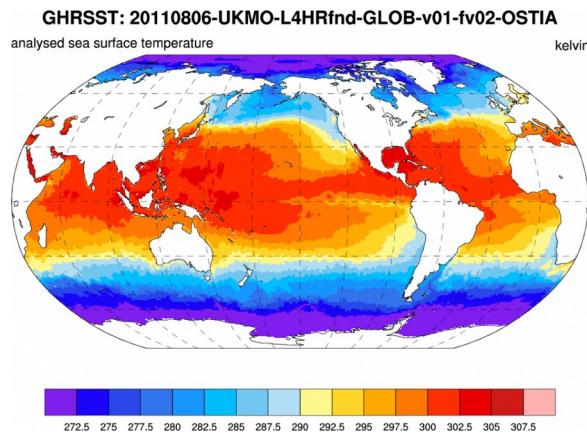
- geology
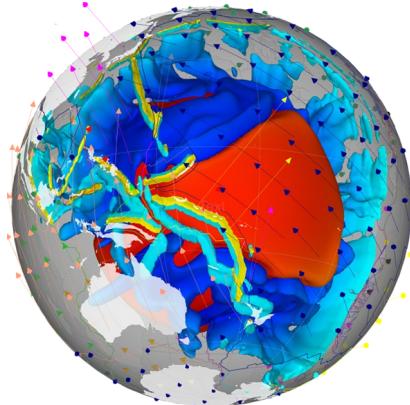
- paleontology

- climate

- geophysics
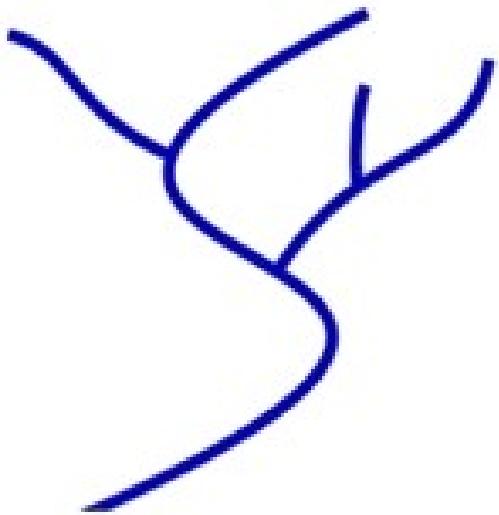
- ecology

Basic data types
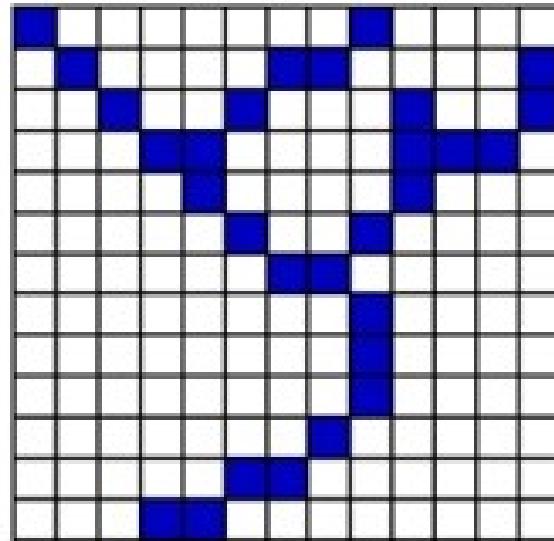
- Similar to computer graphics

- Sometimes in 3D

Object view   *Field view*
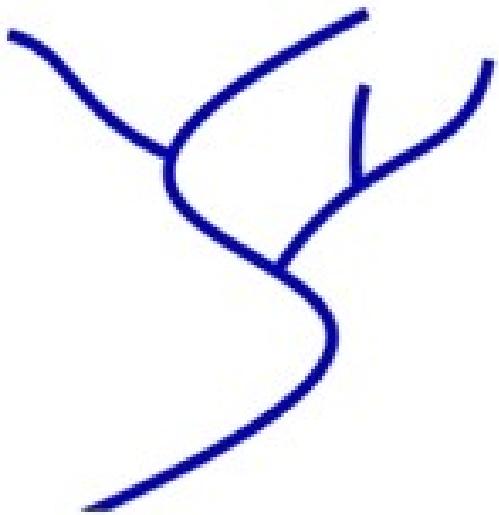
Vector   Raster

Basic data types
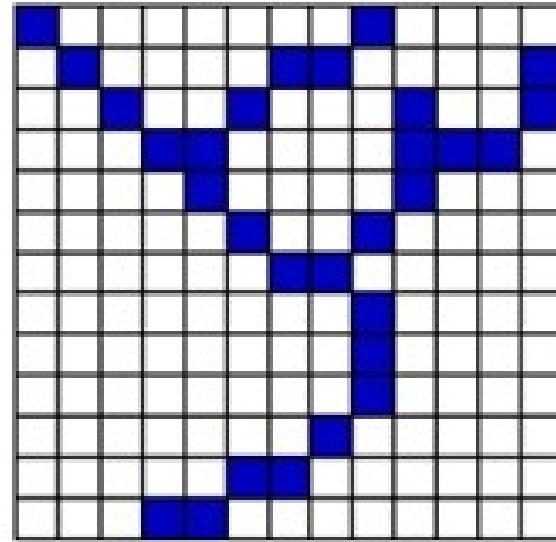
- Similar to computer graphics
- Often in 3D

Object view

Field view
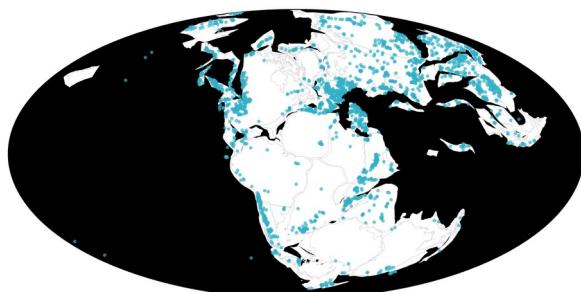
There is one twist:

Vector

Raster

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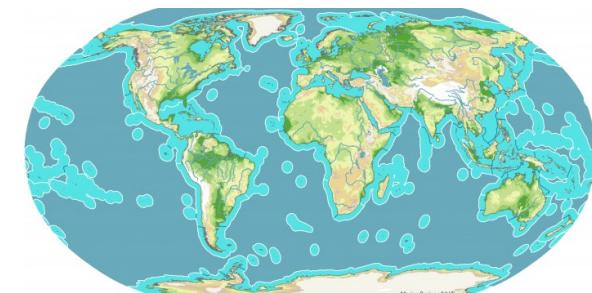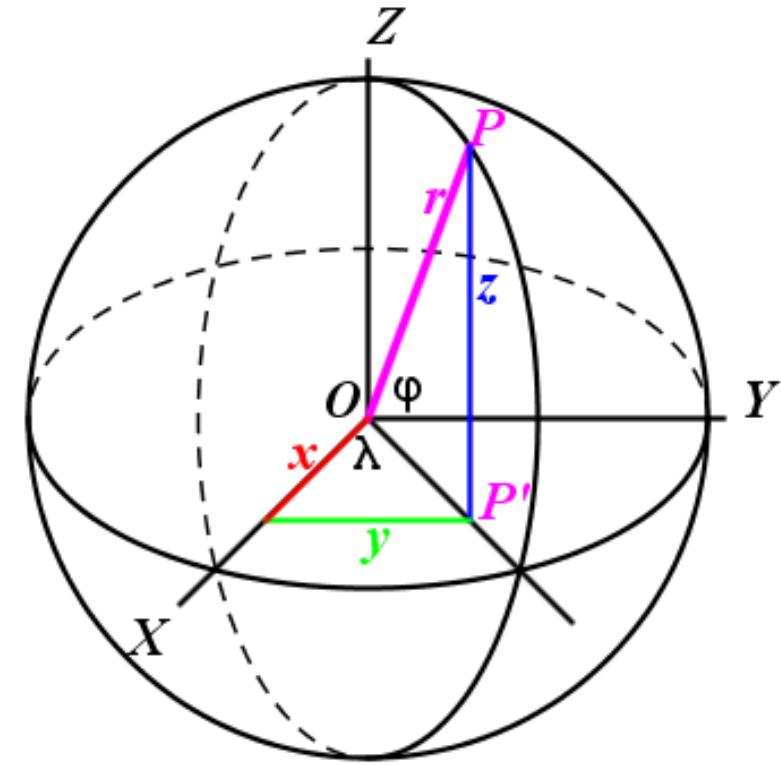
- Structure is defined with points that have coordinates
- Spherical coordinates, Cartesian too, or projection-specific
- Additional attributes are rendered to the entities
- 3 primary kind

Points
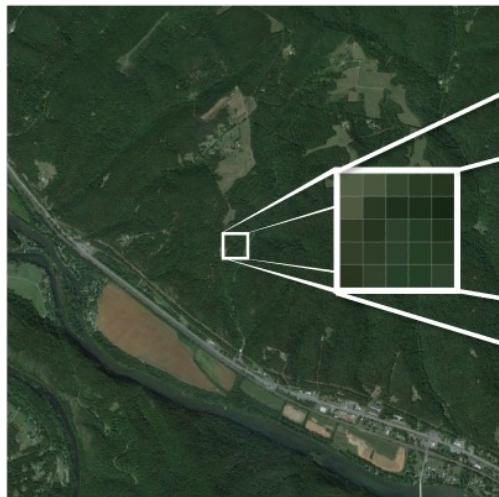
Lines
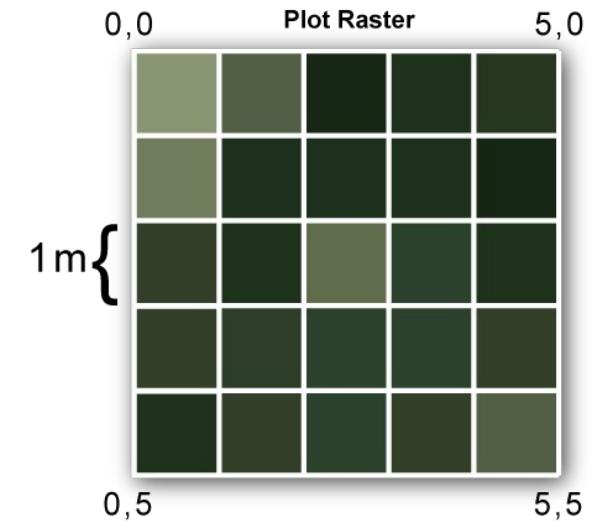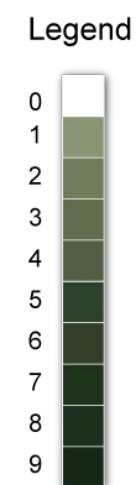
Polygons (shapes)

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
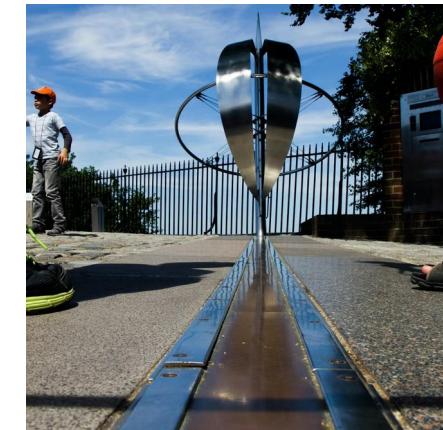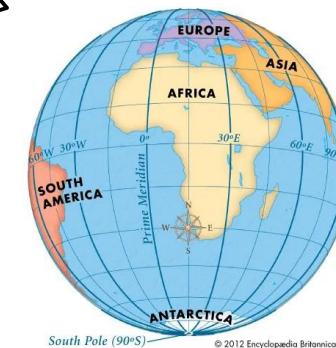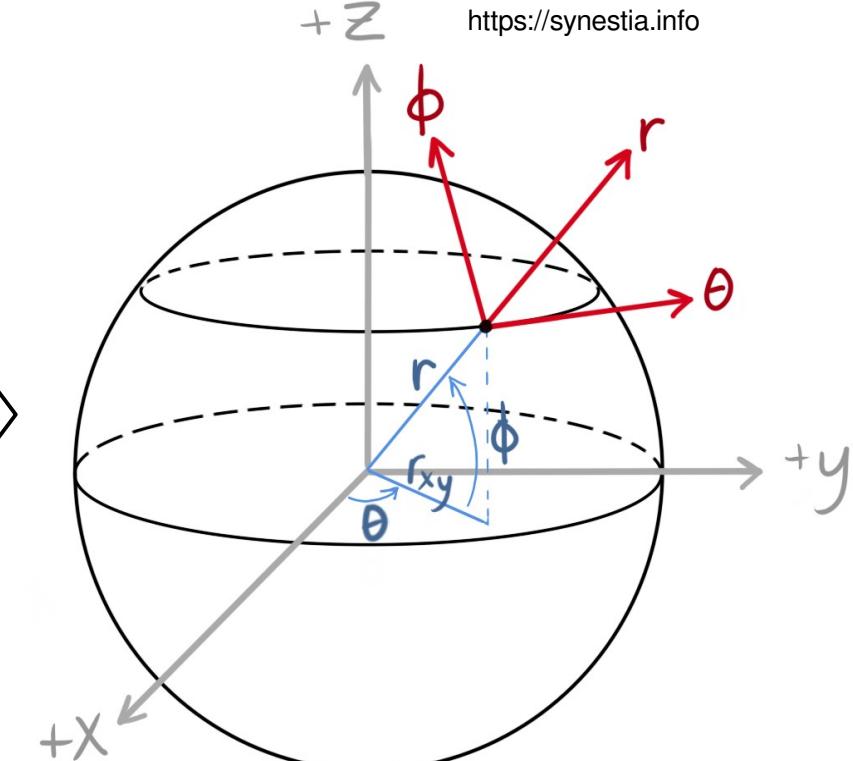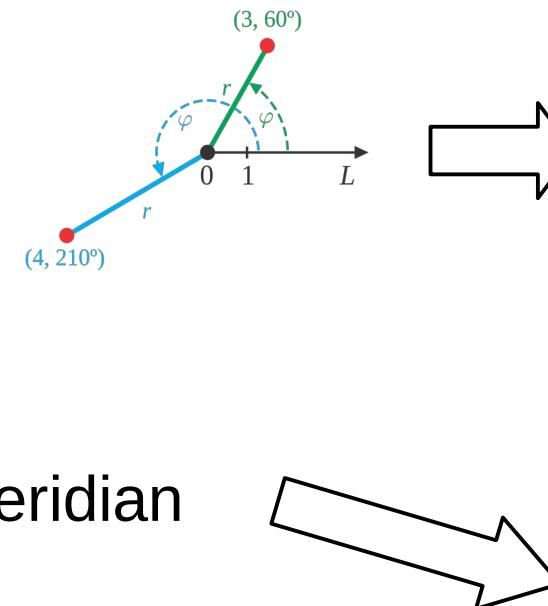
Coordinate Reference Systems

Geographic coordinates
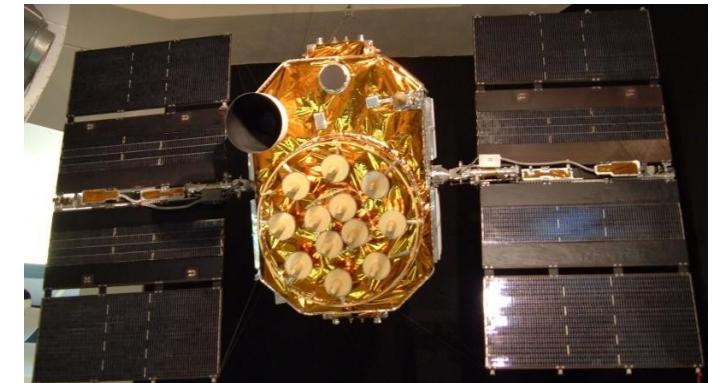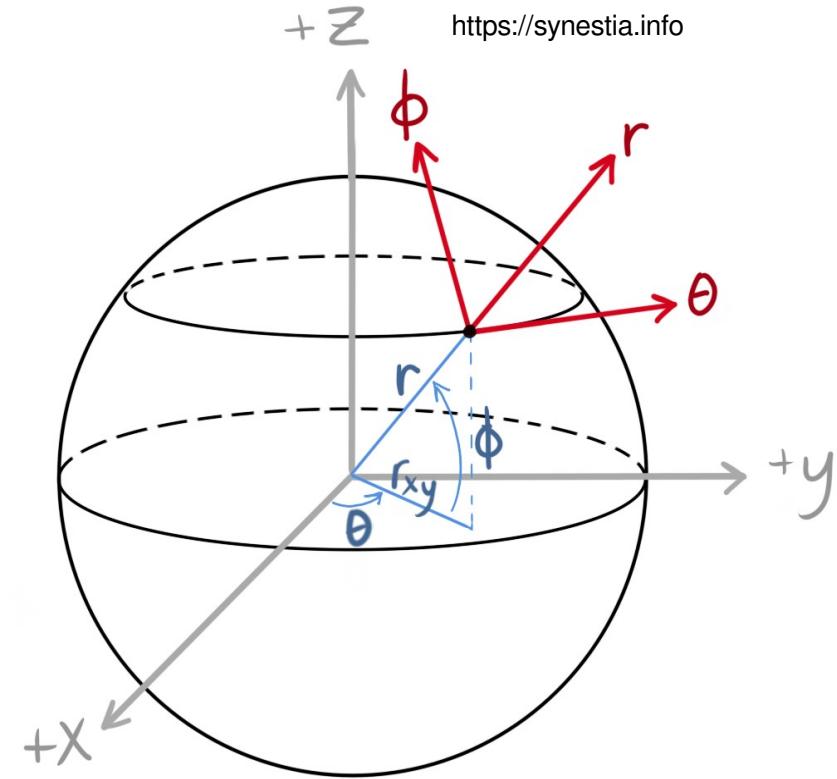
- Based on polar coordinates

- φ: latitude

  - [-90°, 90°], equator : 0°

- θ: longitude

  - [-180°, 180°], 0°: prime meridian

- r: elevation

Coordinate Reference Systems

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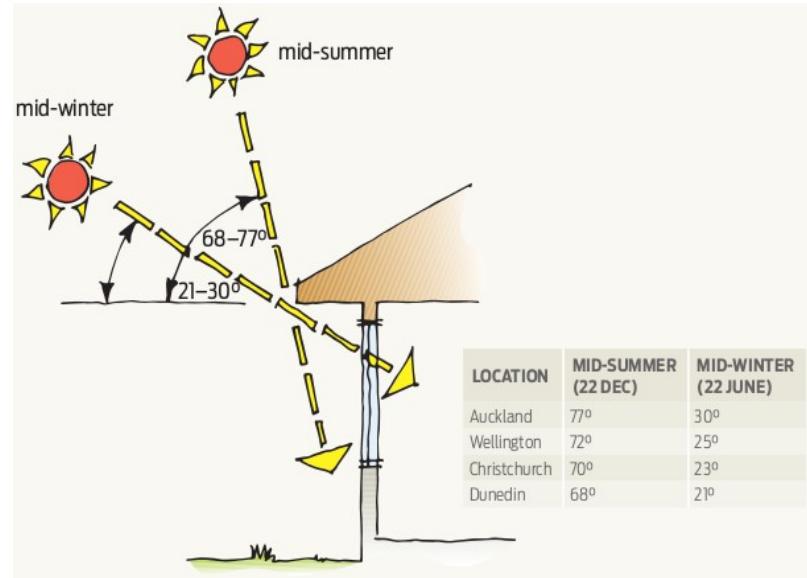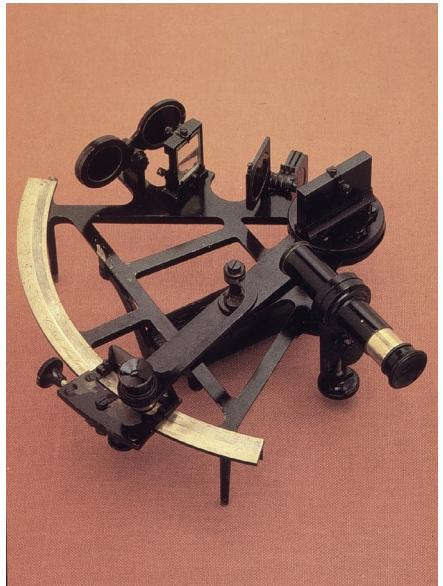
- x, y, z, Earth-Centered Coordinate System
- Based on center of mass, equator and PM
- Used for satellite tracking and navigation (e.g. GPS)
- WGS84 datum

https://synestia.info

Coordinate Reference Systems
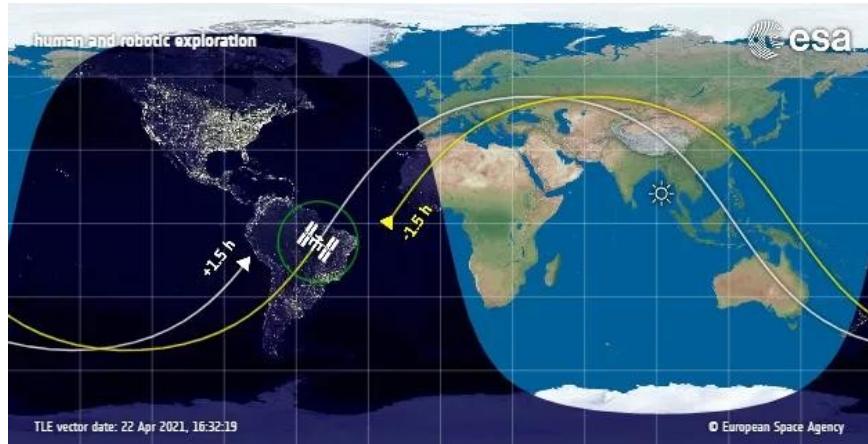
Projection based coordinate systems

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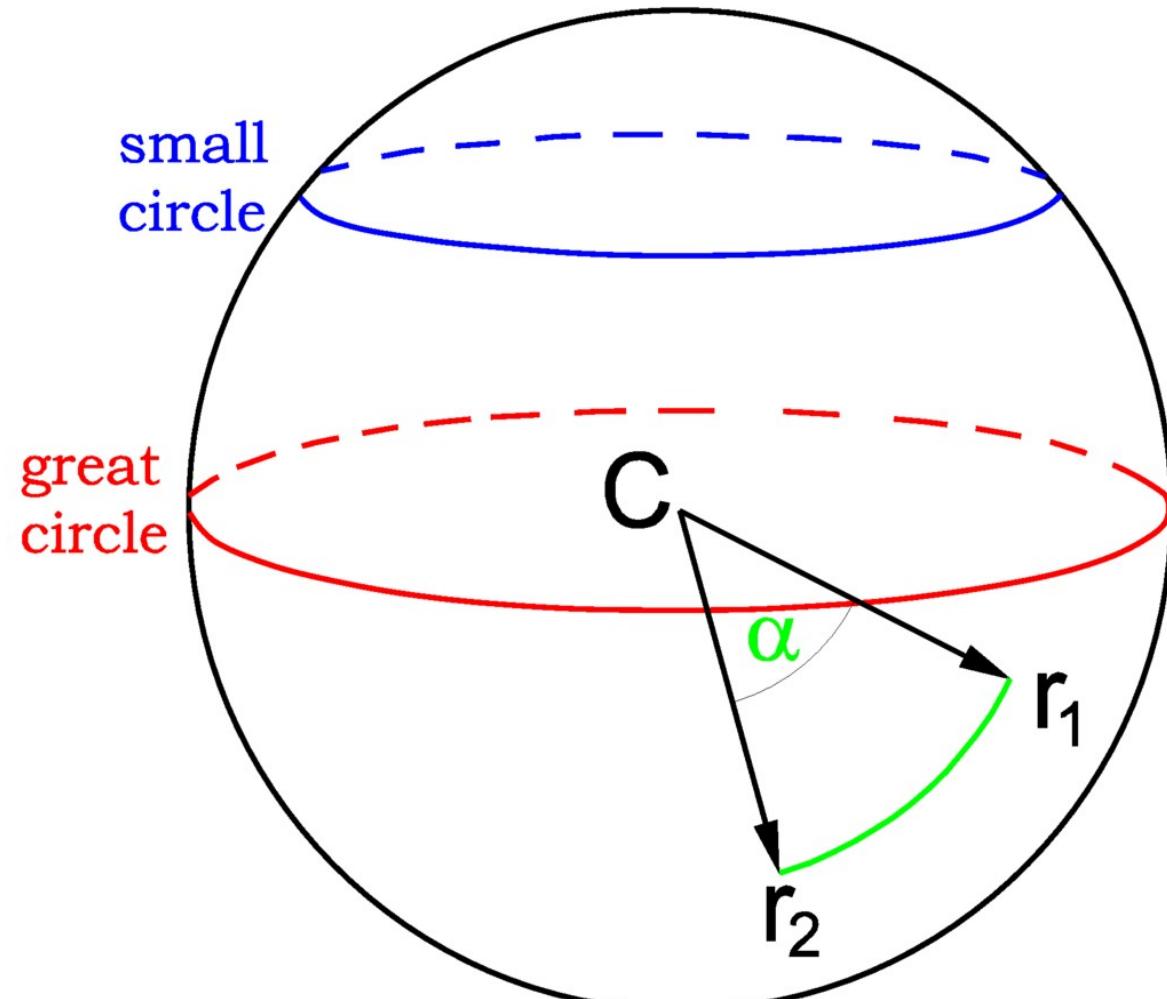
Spheres and Circles

- There are no straight lines
- All points are connected by arcs, either along a great or a small circle
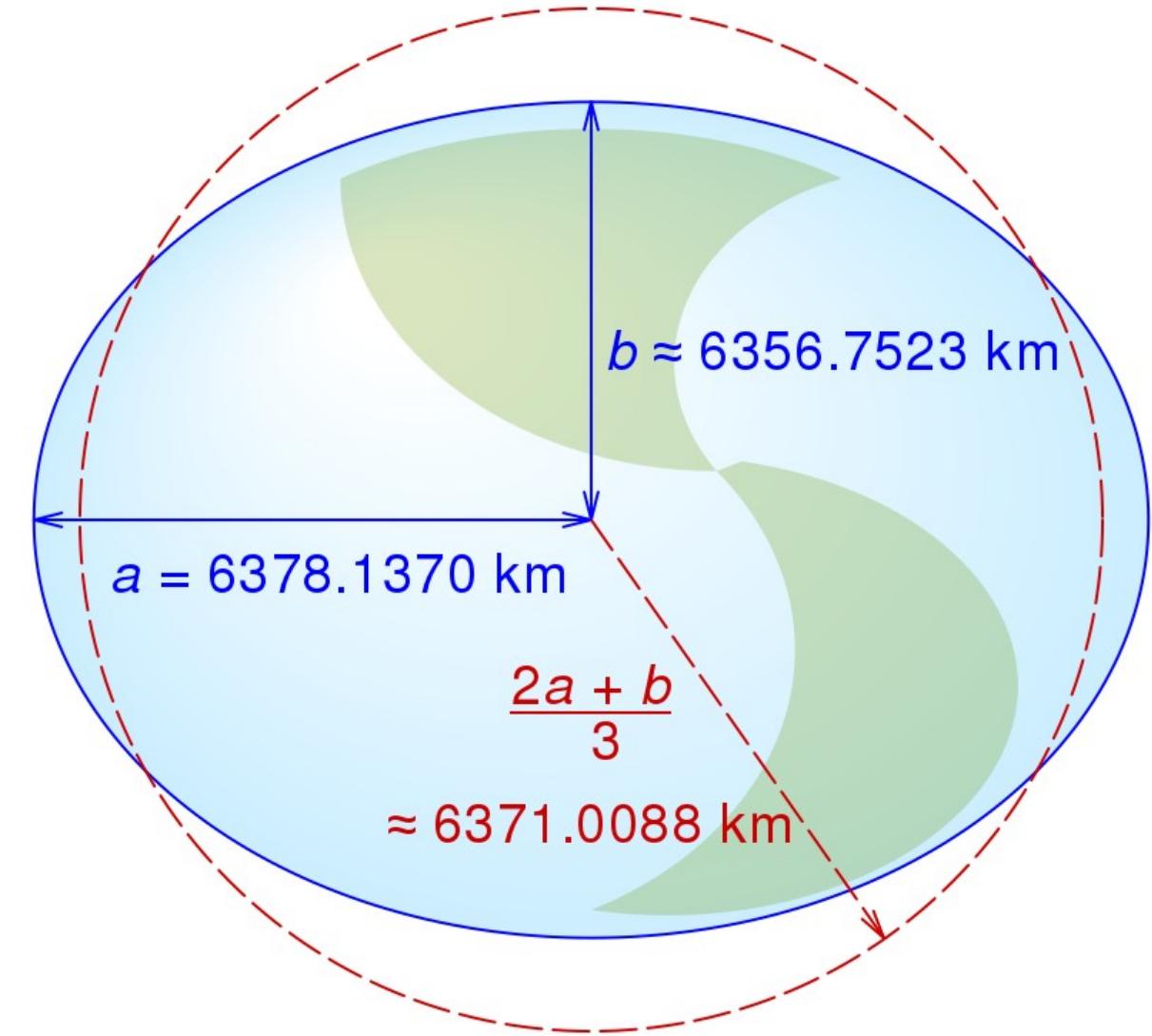
Path of the ISS

The reference ellipsoid

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

$b \approx 6356.7523$ km

$a = 6378.1370$ km

$$\frac{2a + b}{3}$$

≈ 6371.0088 km

Coordinate Reference Systems

Types of Map Projections

Cylindrical   Azimuthal   Conic

General types...

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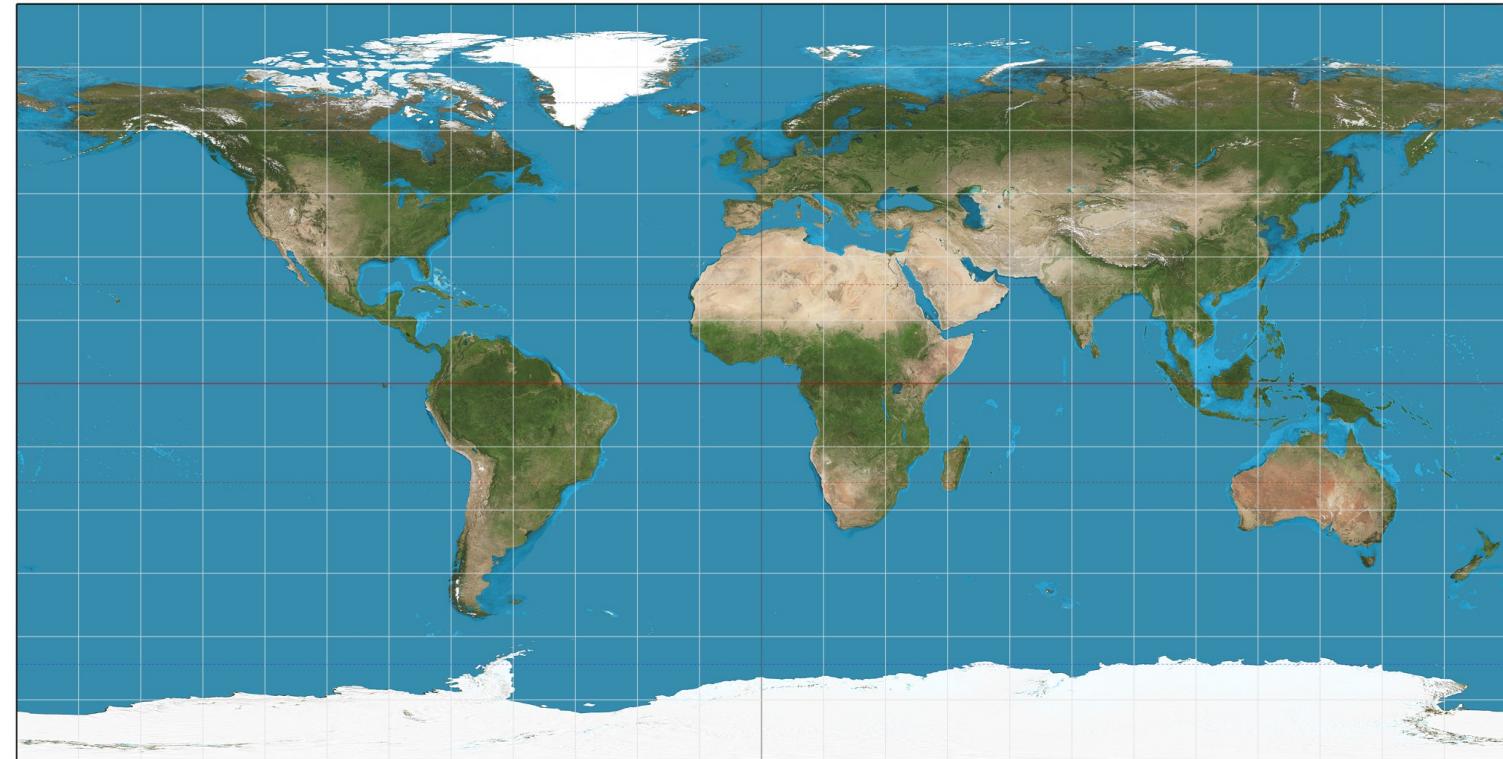
The True size of.. :
https://www.thetruesize.com

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
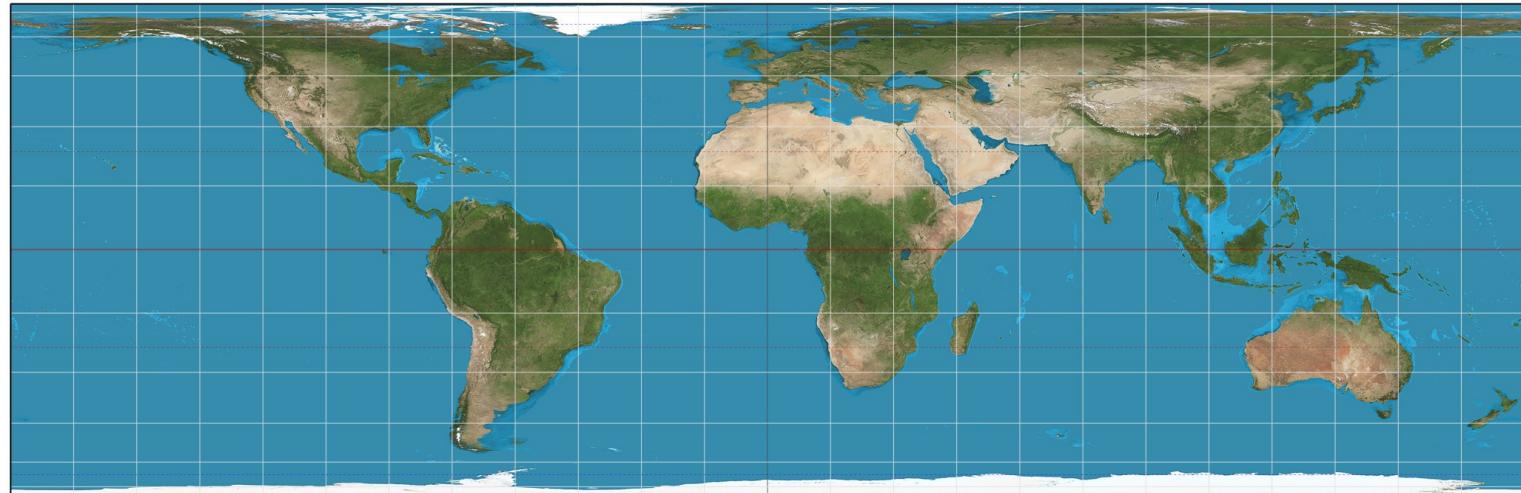
- Conformal

- (nautical use)

- Highly distorts areas

- Cannot represent poles

- Variant: Google Earth

Coordinate Reference Systems

e.g. Lambert's Cylindrical EA Projection
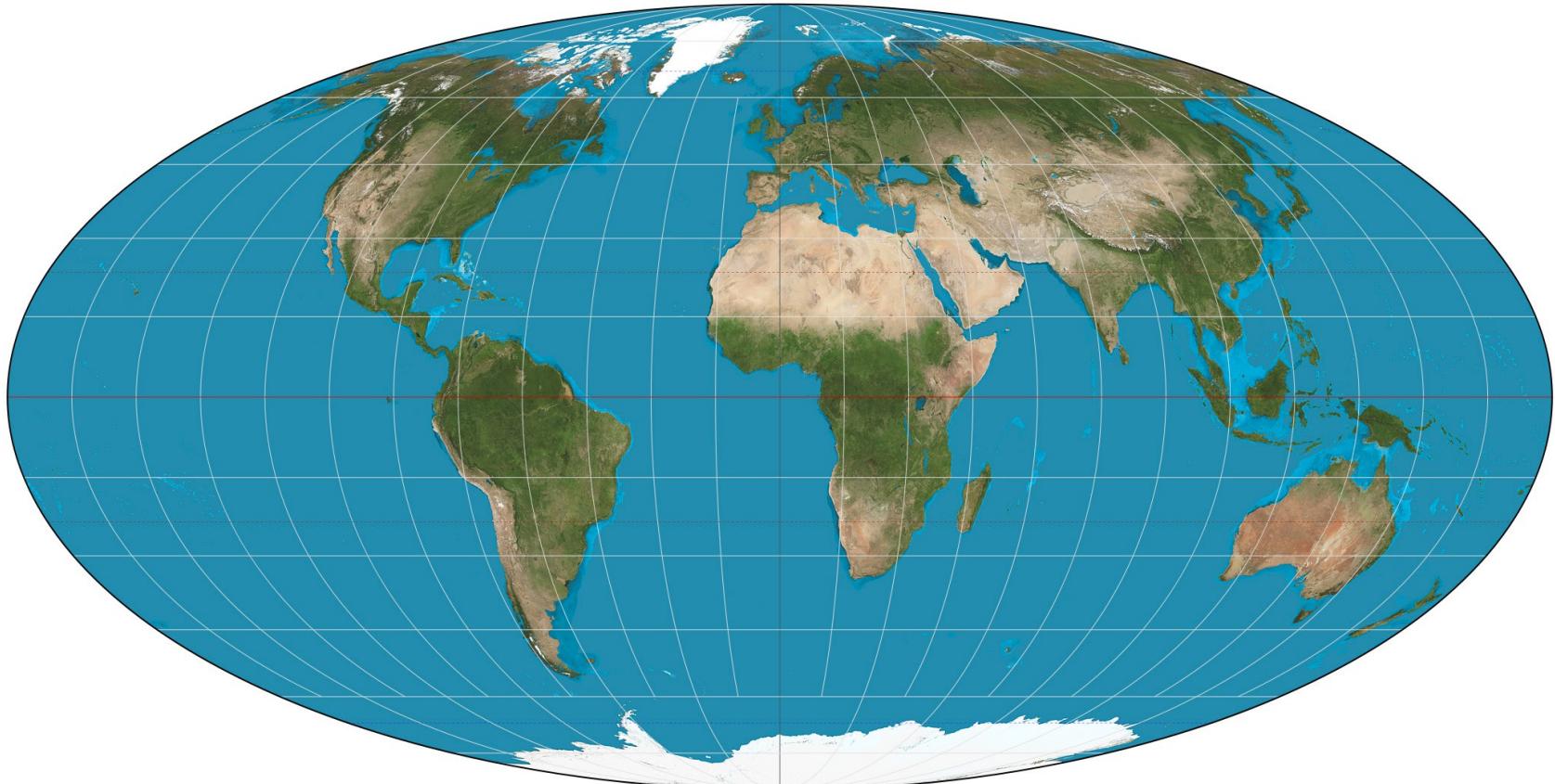
- Equal-area

- Highly distorts angles latitudinally

Coordinate Reference Systems

**e.g. Mollweide Projection**
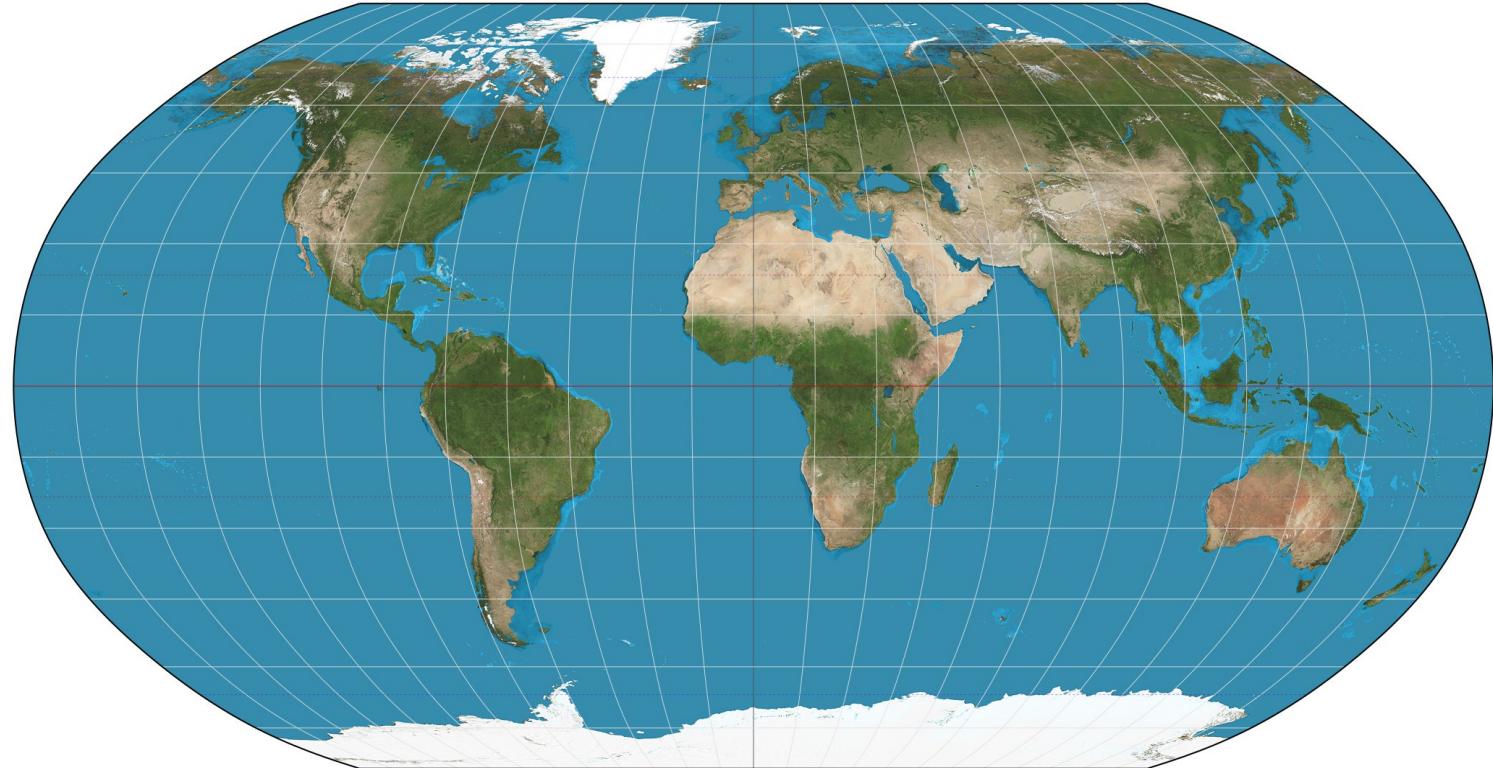
- Equal-area

- 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

Implementation
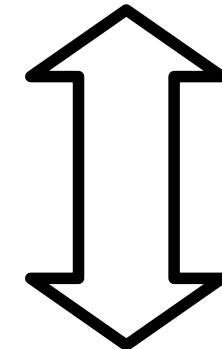
Where, what, why?

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 9.0.0 (March 2022)

- R packages rely on older versions (PROJ4 or PROJ6)

P R Ø J

Open Source Libraries/Projects
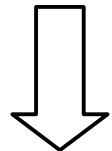
GEOS: Geometry Open Source

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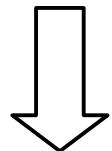
The R packages

- 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...
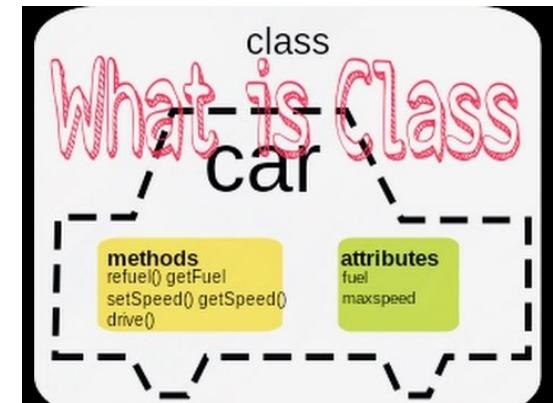
Lot of dependencies, slow development

The current mess

What is a class?

https://www.youtube.com/watch?v=XqrkcO42Dl8

Current issues with R spatial

PROJ4 to PROJ6-based

- This is the worst time to learn spatial calculations in R
- Old packages have been around for 10+ years

Packages

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

+ others: e.g. `geosphere`

What to learn?

Focus on the new but...

- Lot's of older methods rely on the older packages
- The objects can typically be translated from one to another, without problems

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 list, again:

```
sp raster rgdal rgeos terra sf (stars) geosphere
```
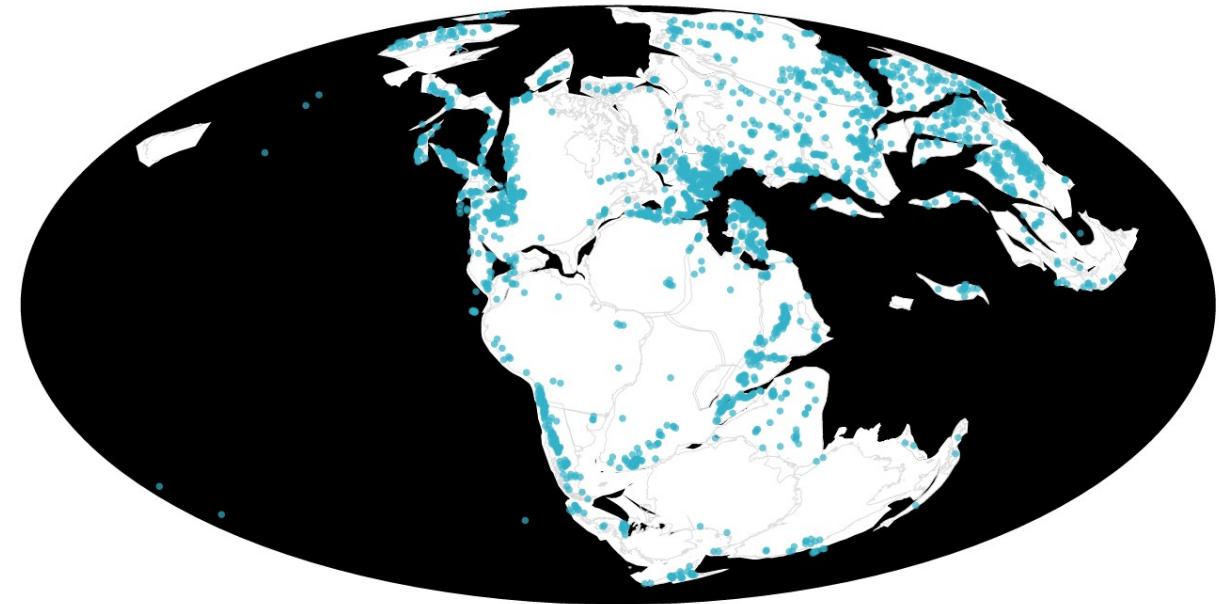
Vector data

with sf (and sp, rgdal, and rgeos)

Data concepts

Feature types: points

- A feature is a thing – everything is a feature, that can be combined to form sets.

  - **Geometry**: It has coordinates and a CRS

  - **Attributes**: every point has data such as age, name, id etc..

- Features can be combined into sets, e.g. Multipoint
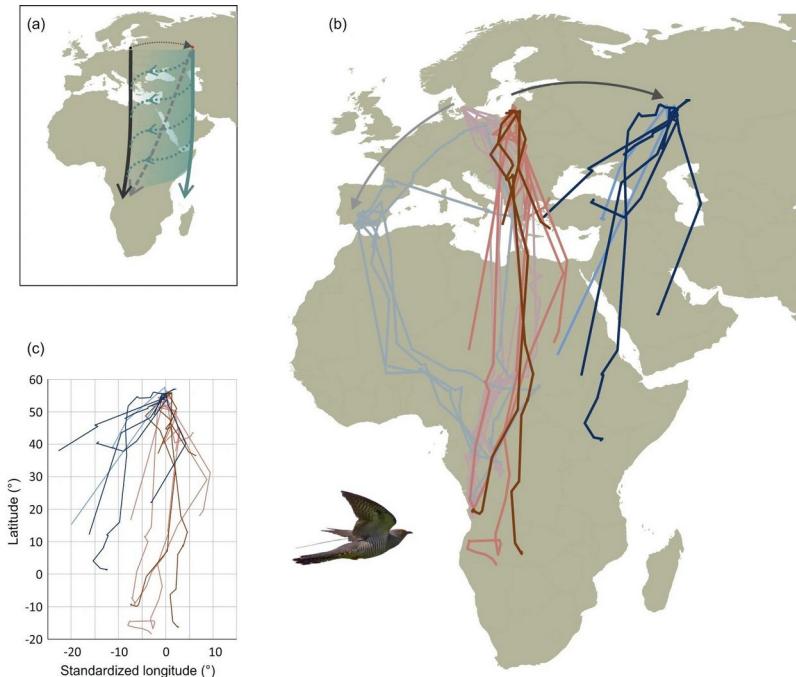
Fossil occurrences in the PBDB

Data concepts

Feature types: line

- Made up of sequence of points
  - One line: path, GPS track
  - Multiline: multiple paths combined that represent one thing
- Both a single line and a multiline can have attributes
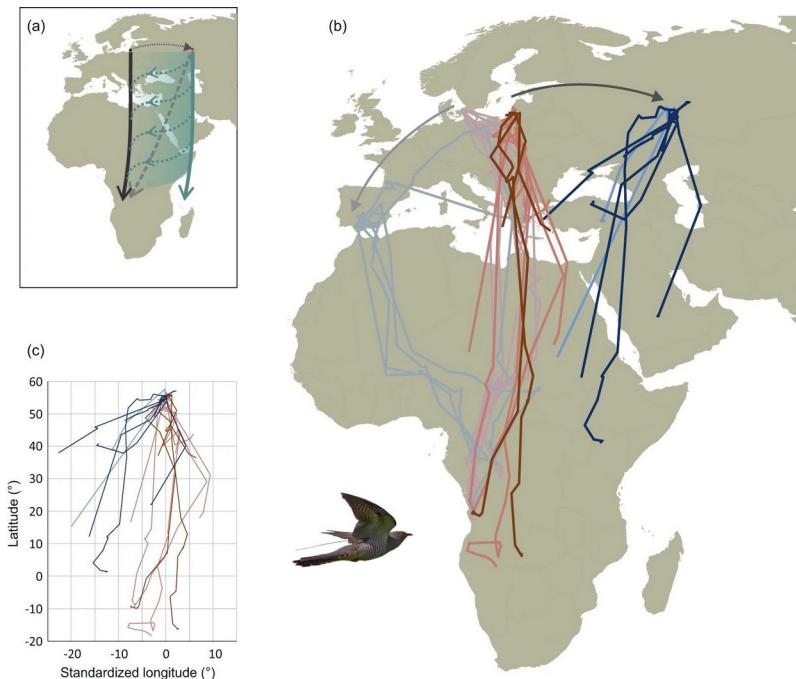
Thorup et al. 2020 Sci. Rep.

Data concepts

Feature types: line

- Made up of sequence of points
  - One line: path, GPS track
  - Multiline: multiple paths combined that represent one thing
- Both a single line and a multiline can have attributes
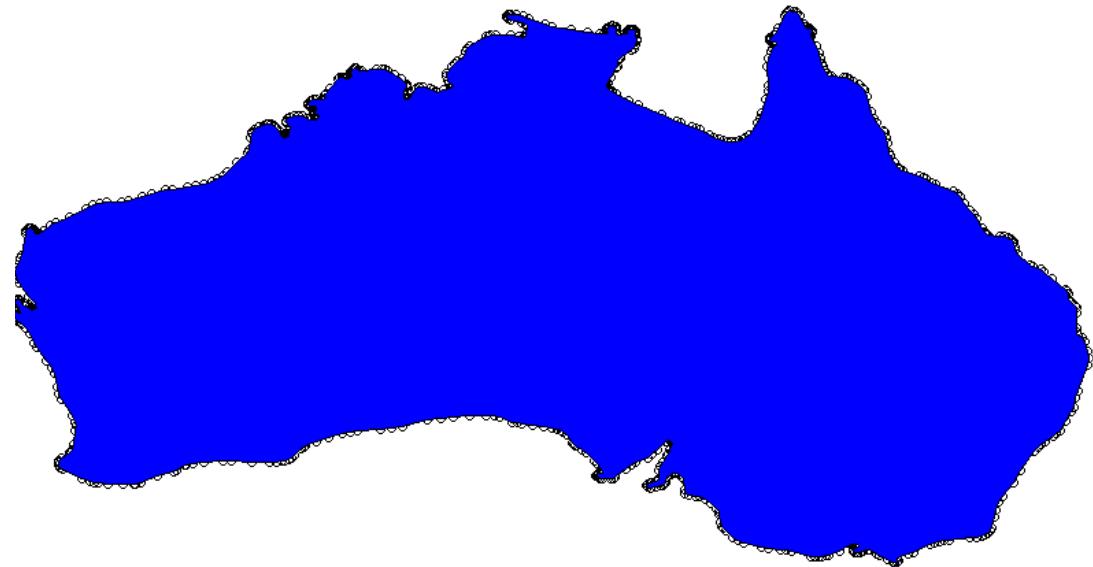
Thorup et al. 2020 Sci. Rep.

Data concepts

Feature types: Single Polygons

- Made up of sequence of points
- Closed to separate inside from outside
- Every polygon has attributes
- E.g. every island/landmass is by definition one polygon

Data concepts

Feature types: Multiple polygons
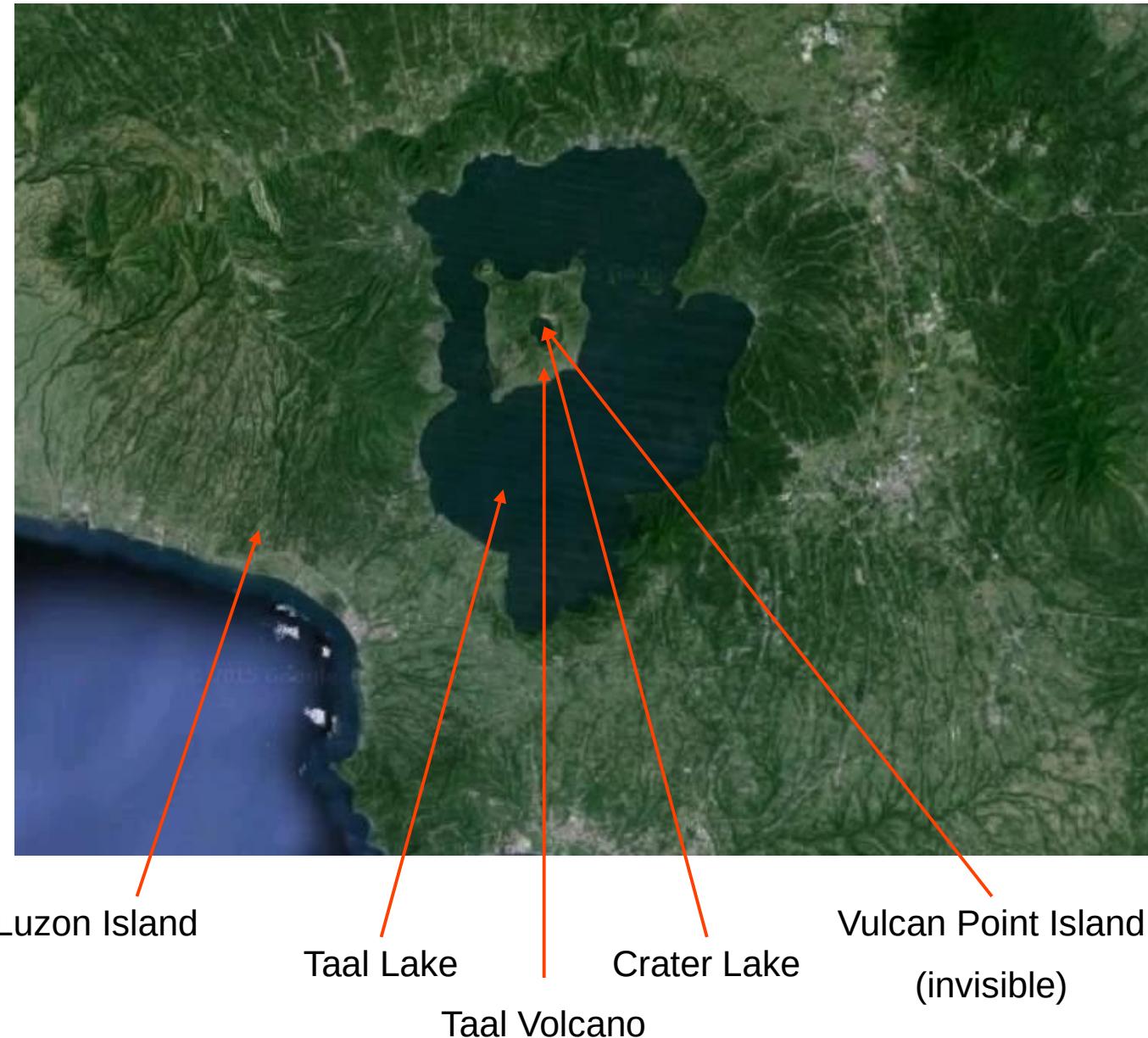
- Polygons can be combined to represent single entities
- One thing is represented by multiple polygons, e.g. a country

Data concepts

Feature types: Multiple polygons

- Polygons can have holes

- Holes are also polygons that are inside other polygons

- The direction of points defines whether something is a hole or not

Luzon Island

Taal Lake

Taal Volcano

Crater Lake

Vulcan Point Island
(invisible)

Registering complex data in R

- For GIS, 2 column matrices can do the job for points, lines and polygons

- You can create lists of Polygons and lists of that, but it quickly becomes a mess…

- S3 and S4 classes

	long	lat
1	142.3752	-10.99448
2	142.5119	-11.01241
3	142.6009	-11.05538
4	142.6376	-11.08747
5	142.7468	-11.26374
6	142.9131	-11.68280
7	142.9775	-11.80399
8	143.0637	-11.90817
9	143.0800	-11.90902
10	143.0876	-11.88224
11	143.1104	-11.88547
12	143.1691	-11.92764
13	143.2208	-11.98907
14	143.2265	-12.05601
15	143.1904	-12.18825
16	143.1972	-12.27063
17	143.3962	-12.80673
18	143.4831	-13.10462
19	143.5455	-13.38480
20	143.5754	-13.67029
21	143.6182	-13.85430
22	143.6887	-14.04426

S3 Objects

- Informal classes, which is based on a label: the `class` attribute

- They are often just lists: access elements with `$` or `[[]]`

- Example the "`lm`" class produced by `lm()`

- Methods dispatch works based on function names:
  - `predict.lm()` is called when `predict()` is used on an object that comes from this class

- There is no constraint on the object structure!

- Example in spatial: sf

S4 Objects

- Formal classes, with a pre-defined structure
- The object has slots, that can have only specific types and they must have them!
- Slots can be accessed with the @ operator
- Methods are written more explicitly for the argument classes (with `setMethod()`
- Most spatial objects use this framework!

An object from the old sp package

Package sp: Classes and Methods for Spatial Data

- Primarily by Edzer Pebesma and Roger Bivand

- From 2005

- The structures to represent the data in R

- S4 classes and how to interact with them

- Primarily used for vector data also some raster

  - `SpatialPoints(DataFrame)`

  - `SpatialLines(DataFrame)`

  - `SpatialPolygons(DataFrame)`

Vector File formats

ESRI Shapefiles

- Multiple files represent the data
  - `.shx`: Index file, where is what
  - `.shp`: Main file, geometries
  - `.dbf`: Attribute information (dBASE)
  - `.prj, .sbn`, etc..
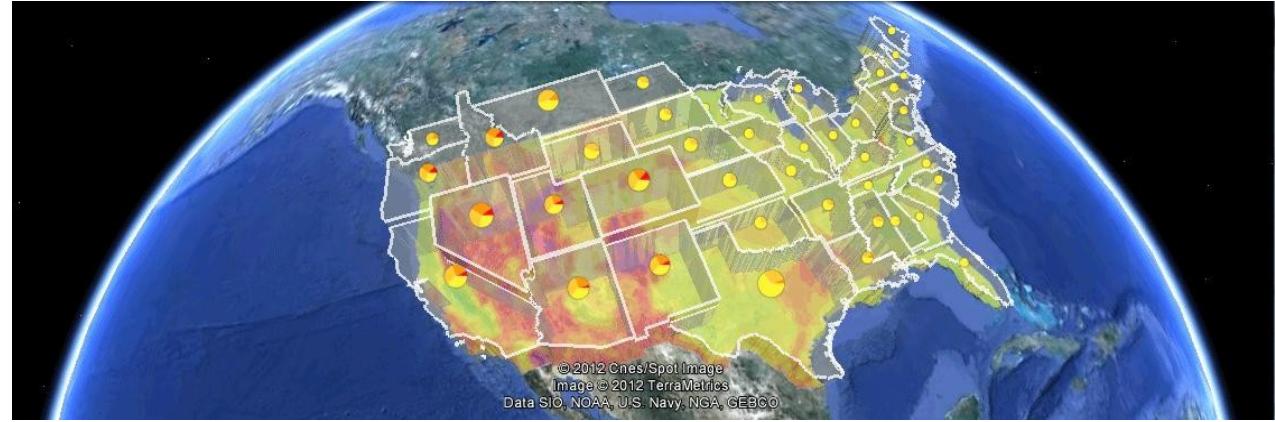
Vector File formats

Keyhole Markup Language

- `.kml` or `.kmz` file (z for zipped)
- Standard for the web

Google Earth uses .kml

Vector File formats

GeoJSON

- `.js` `.json` also for the web
- Human readable

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [125.6, 10.1]
  },
  "properties": {
    "name": "Dinagat Islands"
  }
}
```

PostGIS

- Database-oriented (PostgreSQL)
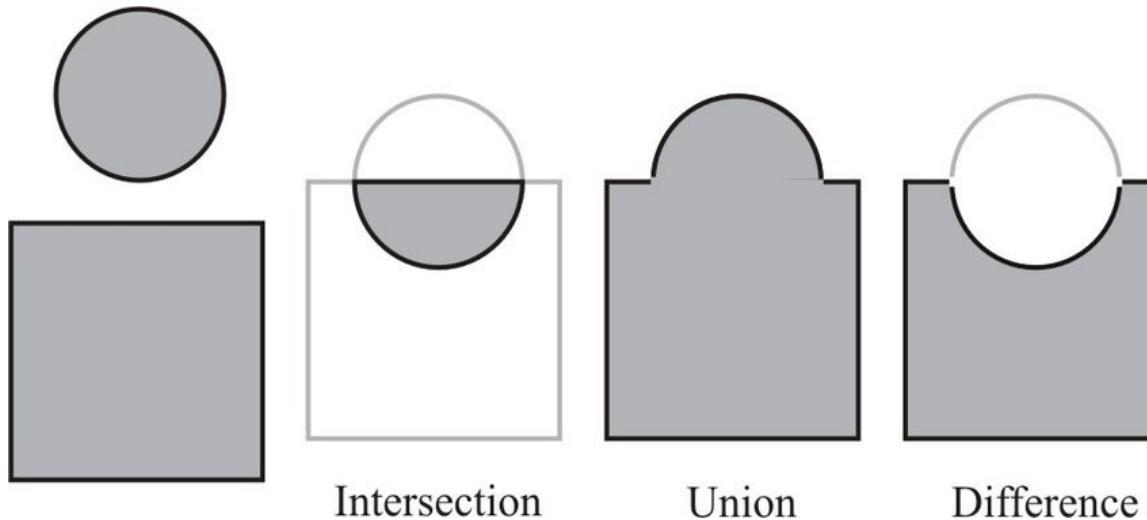
Package rgdal: Bindings for the 'Geospatial' Data Abstraction Library

- Primarily by Roger Bivand

- Interface to GDAL and PROJ<6

- Accessing data (I/O) and executing projections

OLD
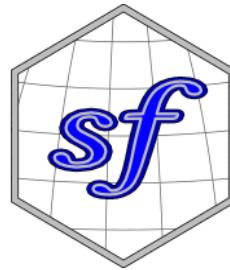
Package rgeos: Interface to Geometry Engine - Open Source ('GEOS')

- Primarily by Roger Bivand

- Interface to GEOS

- Geometric calculations in the same CRS (e.g. Area)

- Boolean operations with shapes

Intersection   Union   Difference
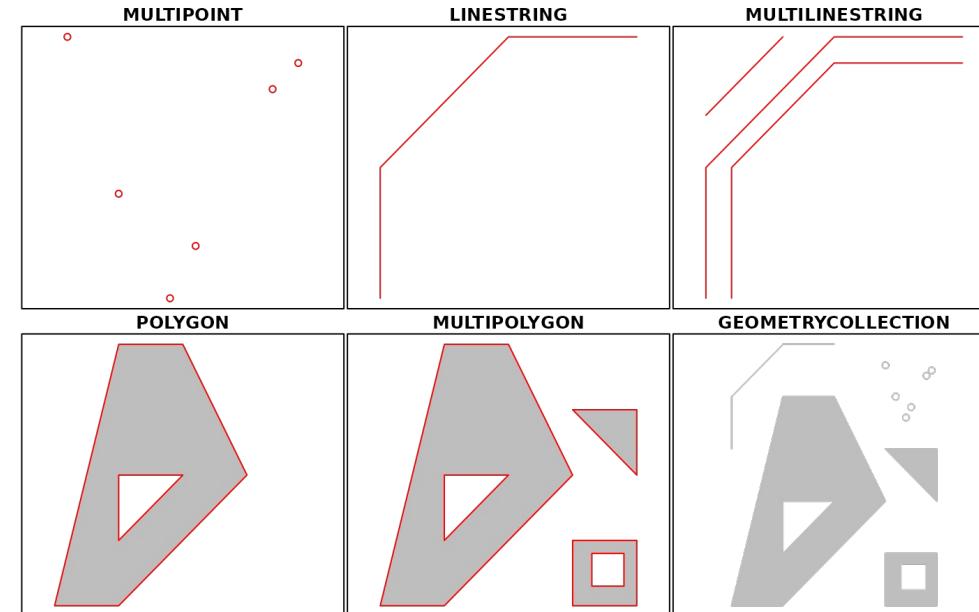
Package sf: Simple Features for R

- Primarily by Edzer Pebesma

- From 2017

- Integrates with GDAL, GEOS and PROJ

- Uses the **simple feature** standard

- Good tidyverse integration

Example 1

- Coastlines Natural Earth
- Plotting a world map!

https://www.naturalearthdata.com/downloads/110m-physical-vectors/

Example 2

- World Administrative Boundaries

World Administrative Boundaries - Countries and Territories

ⓘ Information　⊞ Table　🌐 Map　📊 Analyze　⤓ Export　⚙ API

This dataset displays level 0 world administrative boundaries. It contains countries as well as non-sovereign territories (like, for instance, French overseas).

Dataset Identifier	world-administrative-boundaries
Downloads	33,342
Themes	Administration, Government, Public finances, Citizenship
Keywords	United Nation, ISO-3 code, Countries, Territories, Shape, Boundaries
License	Open Government Licence v3.0
Language	English
Modified	April 26, 2019 11:09 AM
Publisher	World Food Programme (UN agency)
Reference	https://geonode.wfp.org/layers/geonode%3Awld_bnd_adm0_wfp
Territory	📍 world
Last processing	May 15, 2019 9:49 AM (metadata)
	May 15, 2019 9:49 AM (data)

Example 3

- Range data from the IUCN

- Freely available for academic use

Raster data

With terra (and raster)

Package raster: Geographic Data Analysis and Modeling

- Primarily by Robert Hijmans

- Since 2010

- Standard for raster processing for a long time

- Most ecological packages depend on this

- Integration of WorldClim data

Package terra: Spatial Data Analysis

- Primarily by Robert Hijmans

- Interface to GEOS

- Almost the same as raster, this is the easier transition.

Raster File formats

csv
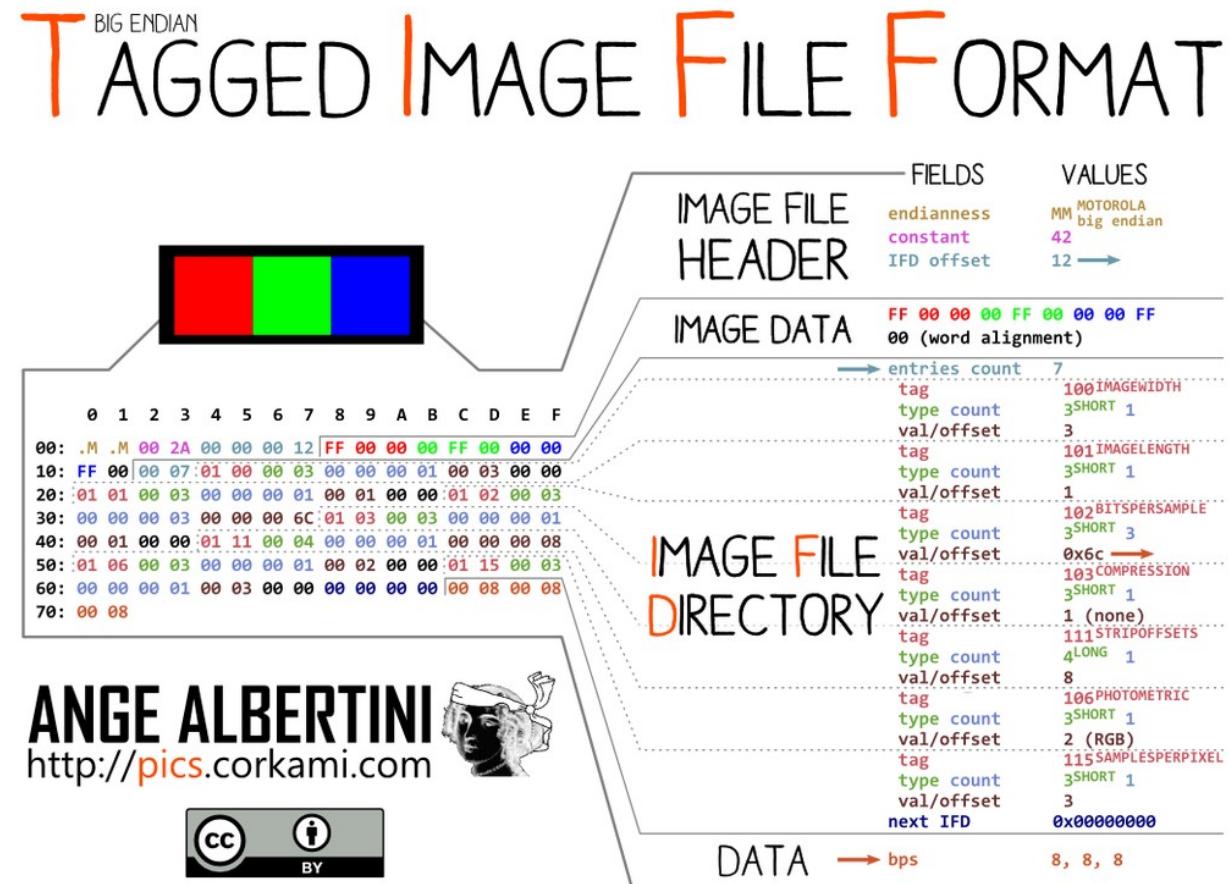
- Single uncompressed layer, a table of values

- Not properly georeferenced

- Column and row names might help

Raster File formats

GeoTIFF

- Single uncompressed layer
- Completely open source
- TIFF image + georeferencing
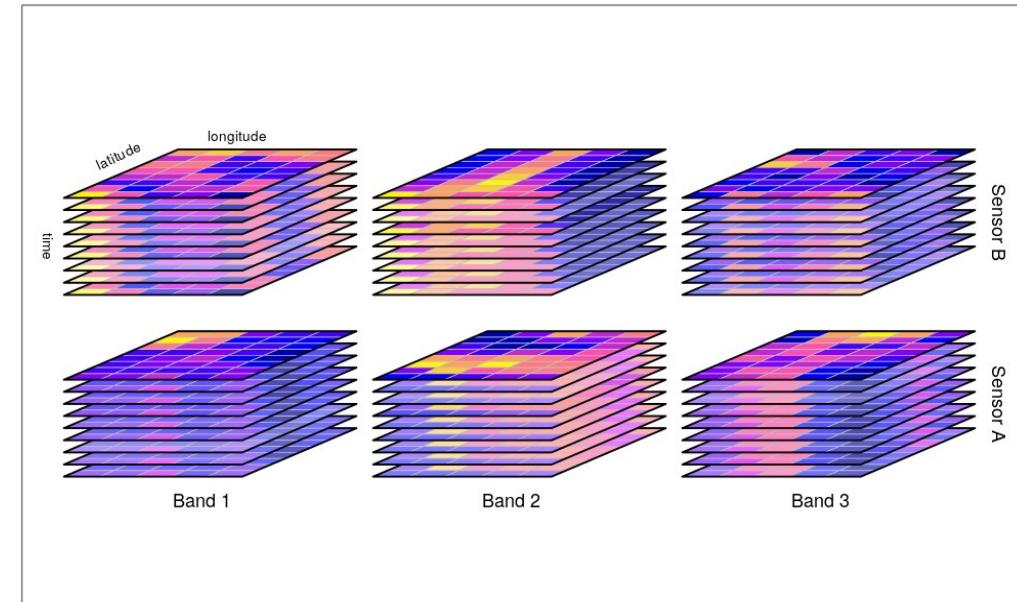
Raster File formats

NetCDF
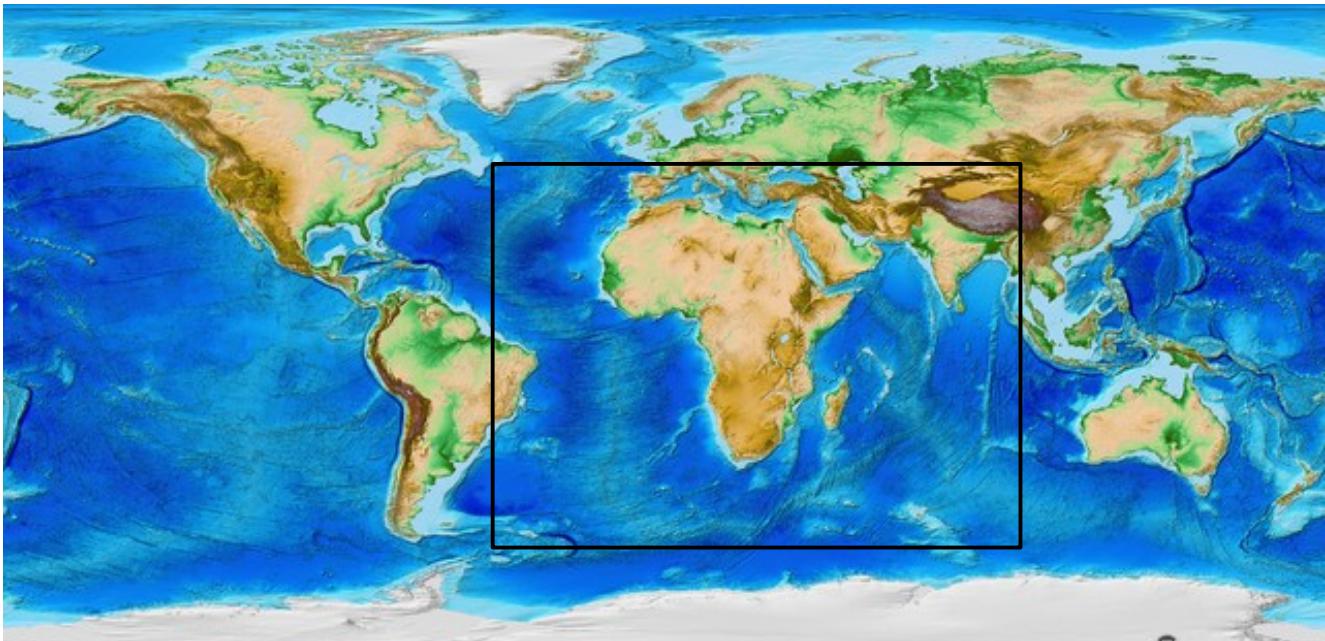
- Binary data cubes, lots of dimensions can be included

- Data registered in bands

- Requires the presence of `ncdf4` in R!

Operations on Rasters

Cropping
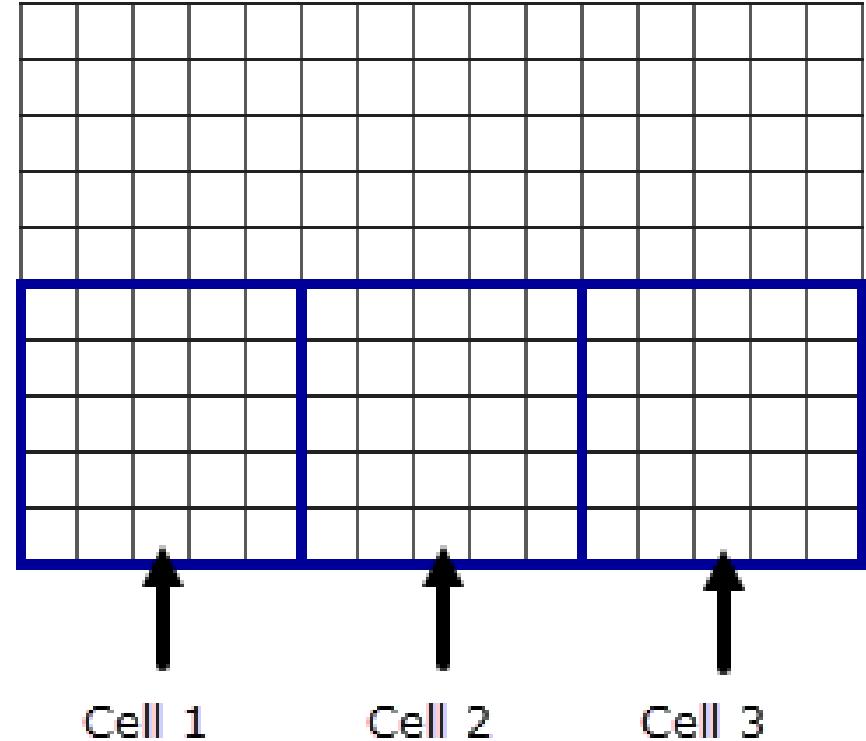
- Changes the extent of a raster

Operations on Rasters

Aggregation / Disaggregation

- Change resolution without changing cell boundaries

- Iteration of a function for data that belongs in a new cell

Cell 1   Cell 2   Cell 3
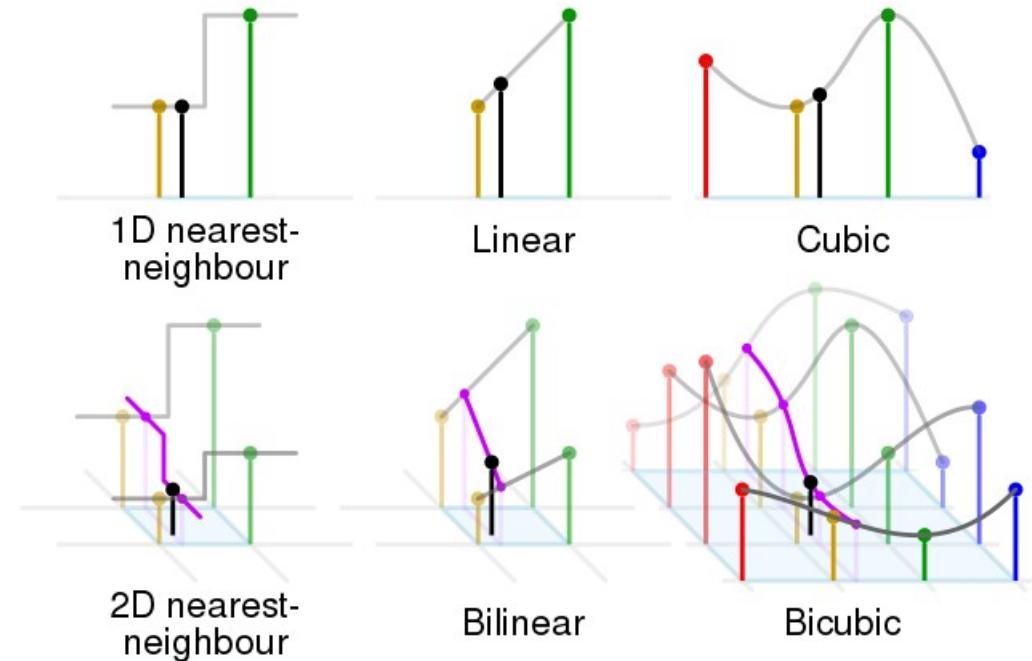
https://desktop.arcgis.com/en/arcmap/latest/tools/spatial-analyst-toolbox/how-aggregate-works.htm

Operations on Rasters

Resampling

- Change resolution to any resolution
- Often relies on interpolation
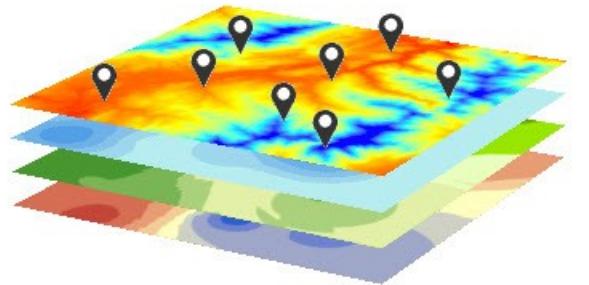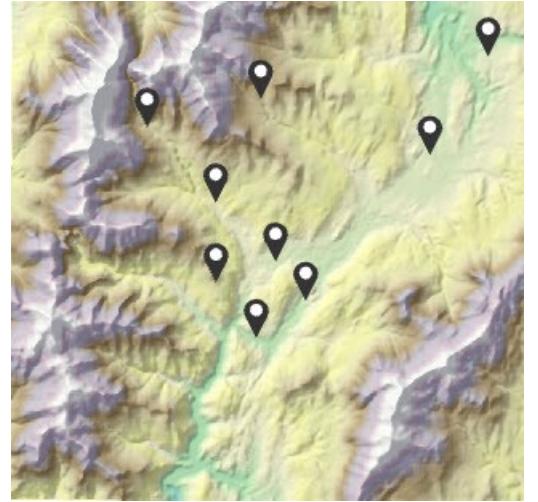
https://www.wikiwand.com/en/Bicubic_interpolation

Operations on Rasters

Extraction
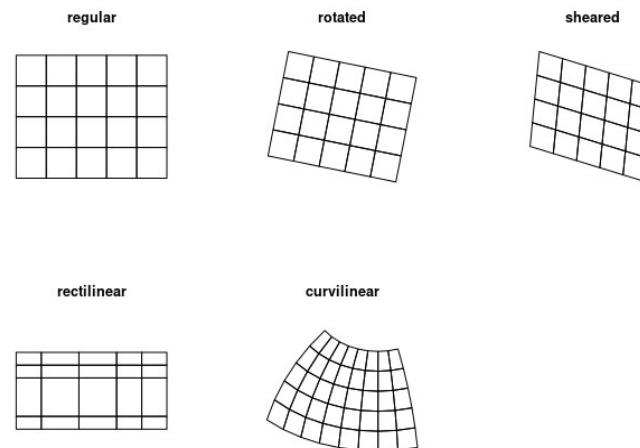
- Getting data out of a raster at given coordinates/locations

- Basic operation for species distribution modelling

esri.com

Package stars: Spatiotemporal Arrays, Raster and Vector Data Cubes

- Primarily by Edzer Pebesma
- Somewhat more complex than terra

Example 1

- The ETOPO1 Topography (1° by 1°)

Example 2

- WorldClim

Example +

- IPCC AR6 Atlas

Non-rectangular raster grids

The advantage of rectangles
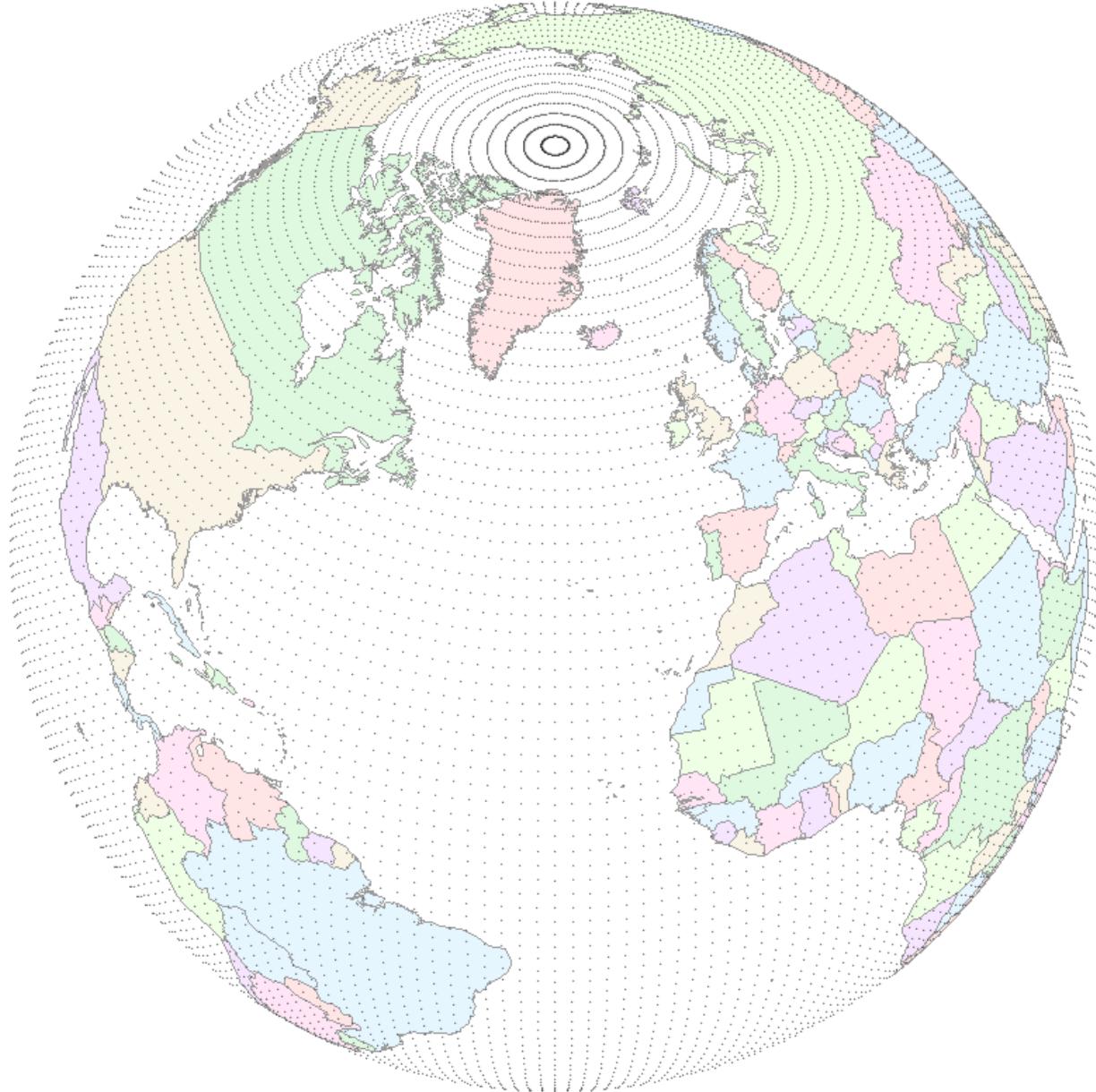
- Structure is easy to define
- Very easy to store in the memory
- Look natural in projections

C.E. map with lat/lon grid lines

# Non-rectangular raster grids

The disadvantage
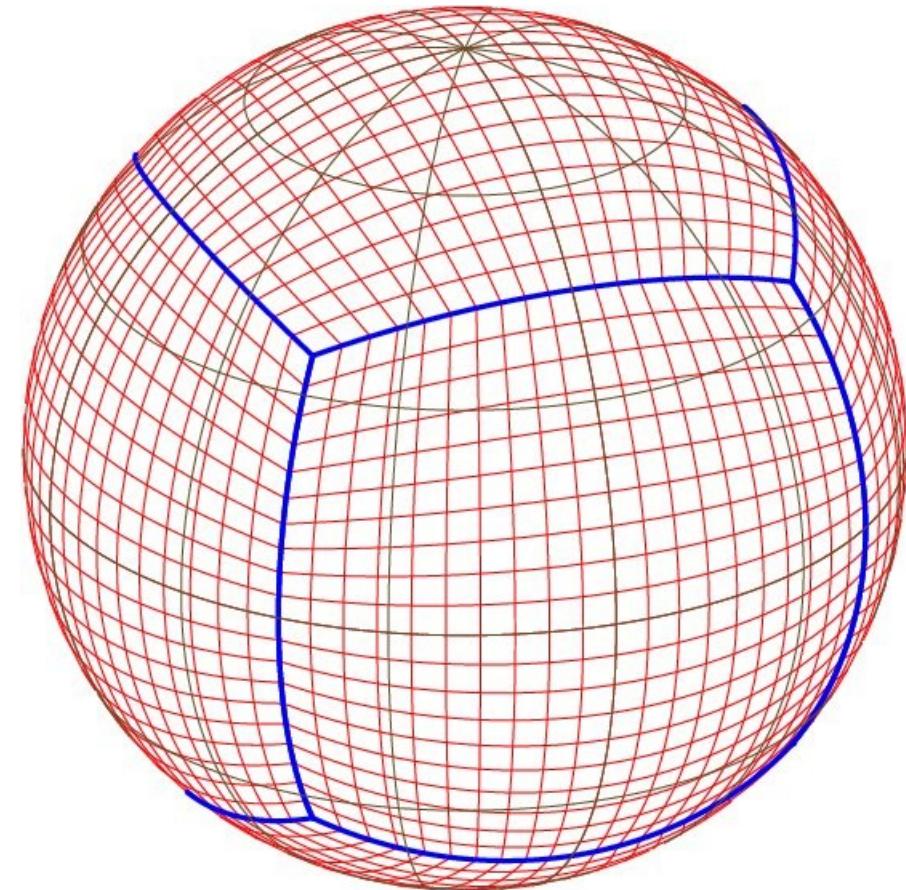
- long-lat grids are Gaussian
- Non-uniform on a sphere
  - Increased density towards the poles
  - Cell sizes decrease towards the poles
- Spatial binning will be biased!

Non-rectangular raster grids

Comon solution: cubed-sphere grid

- Still quadrilateral
- Some climate models rely on this
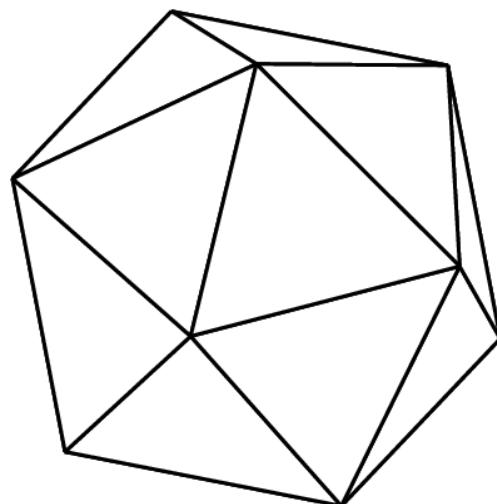
Purser and Tong, 2017
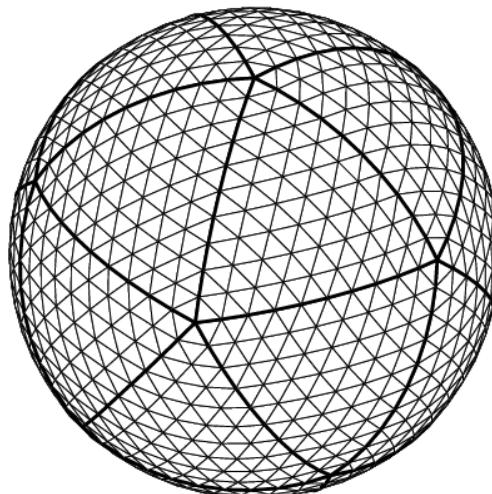
Non-rectangular raster grids

Most common solution: polyhedra
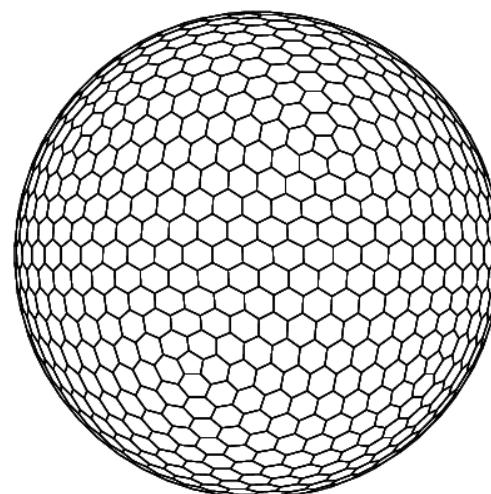
- Trianglular or Penta-Hexagonal grids
- Also in some climate models

(a) Regular icosahedron   (b) N=10   (c) N=10
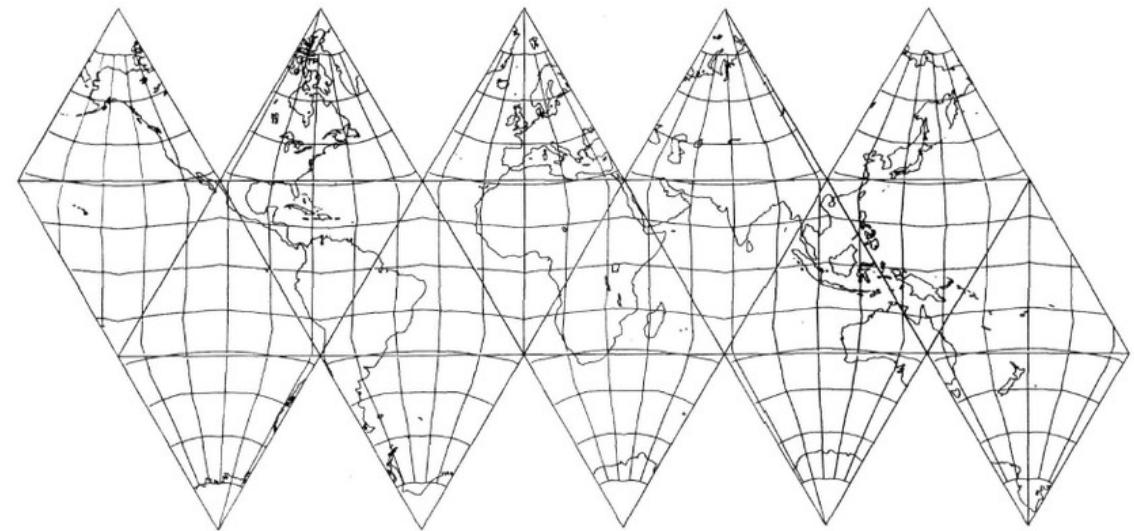
**Fig. 7.** The gradual refinement of the icosahedral geodesic grids.

Chen et al. 2014
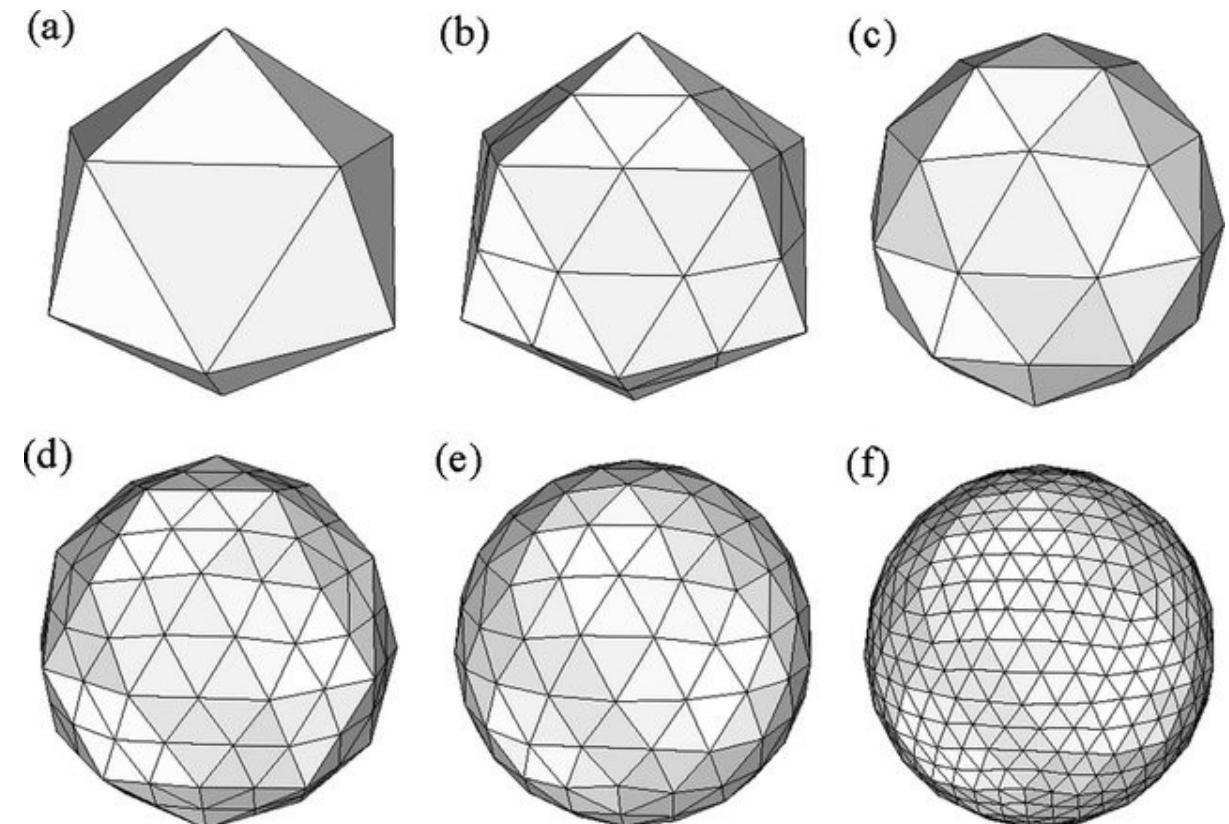
Package ddgridR: Discrete Global Grids for R

- by Richard Barnes

- Based on an program by Kevin Sahr

- Not on CRAN! https://github.com/r-barnes/dggridR

- Based on a Snyder projection

- Very high resolution, and well-known mathematical properties

Package icosa: Coarse grids based on tessellated icosahedra

- I wrote this, still experimental

- More scalable for global scale

- Interface to sp (will be changed)

- 3D model

- The mathematical properties are unexplored

Icosahedron tessellation (Lipscomb and Ringler, 2005)