Introduction to spatial data in R for paleontologists

Ádám T. Kocsis

Friedrich-Alexander-Universität Erlangen-Nürnberg

2024-08-21

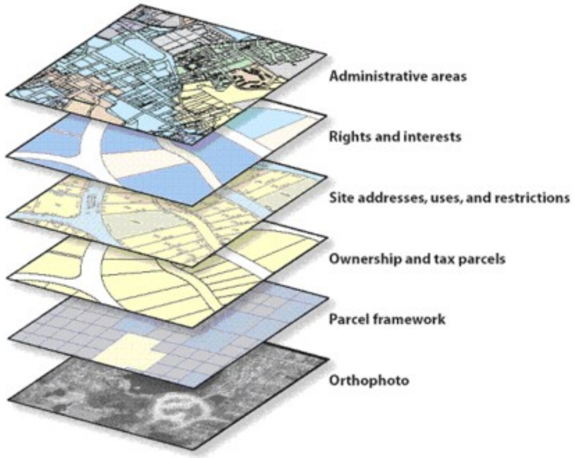
Basic Concepts

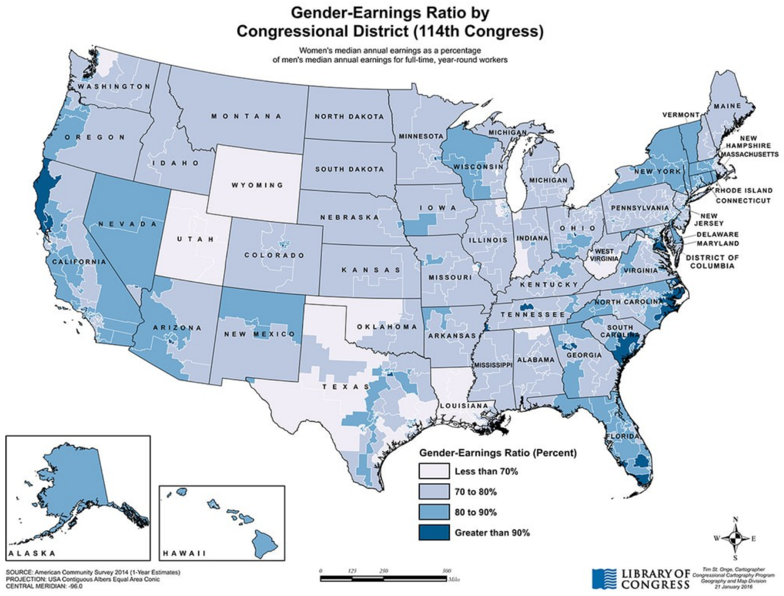
Where, what, why?

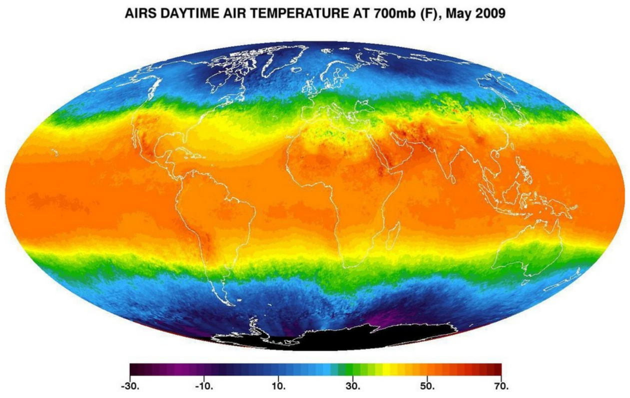
GIS basics

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

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



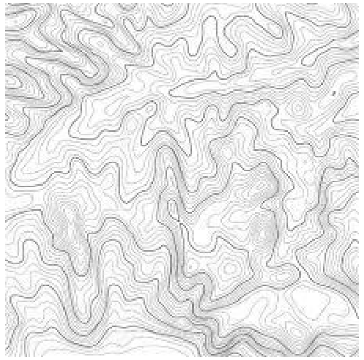




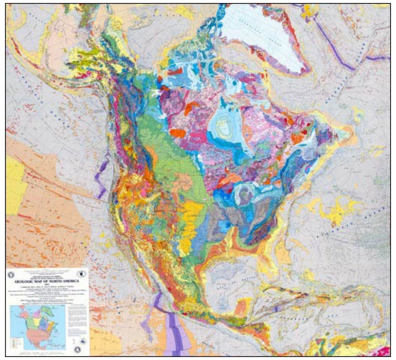
Vital in Earth Sciences

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

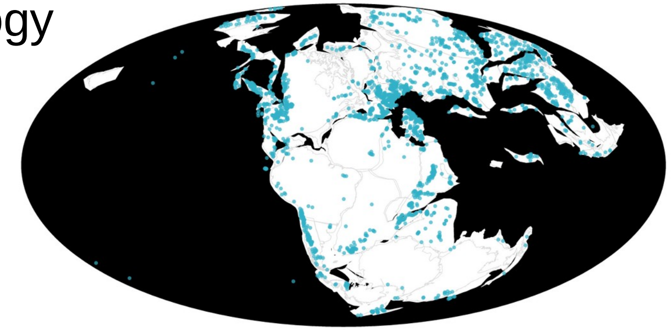
- topography



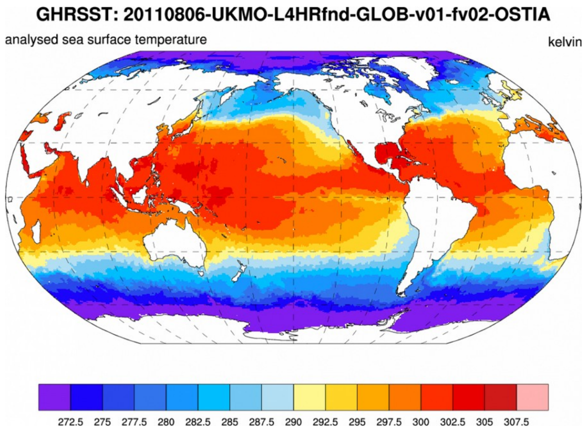
- geology



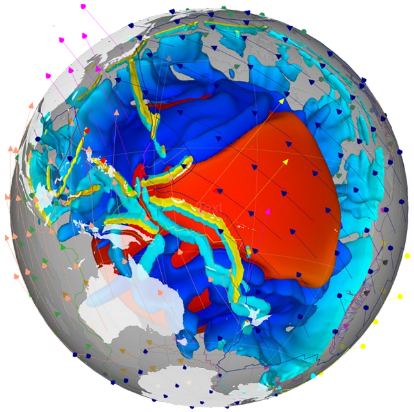
- paleontology



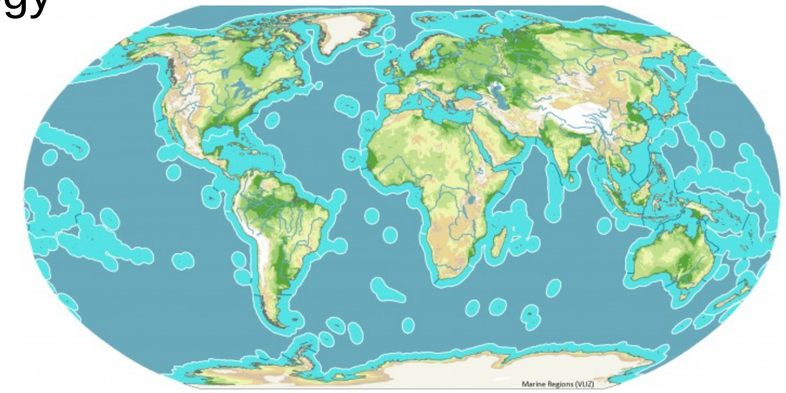
- climate



- geophysics



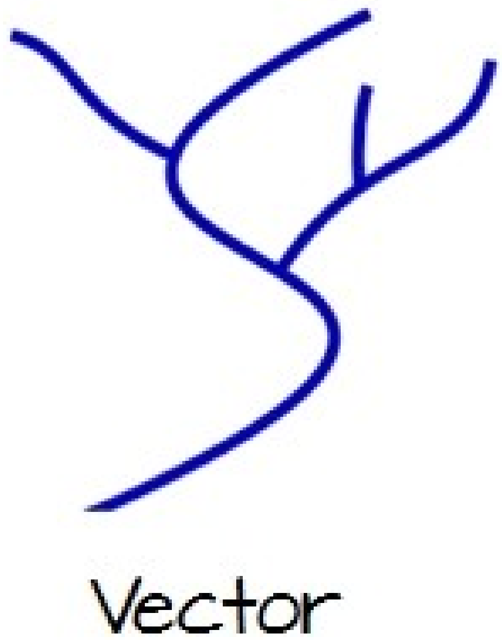
- ecology



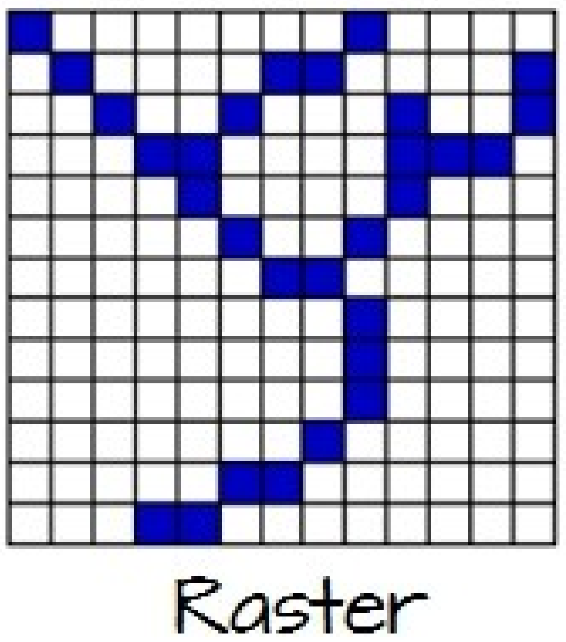
Basic data types

- Similar to computer graphics
- Sometimes in 3D

Object view

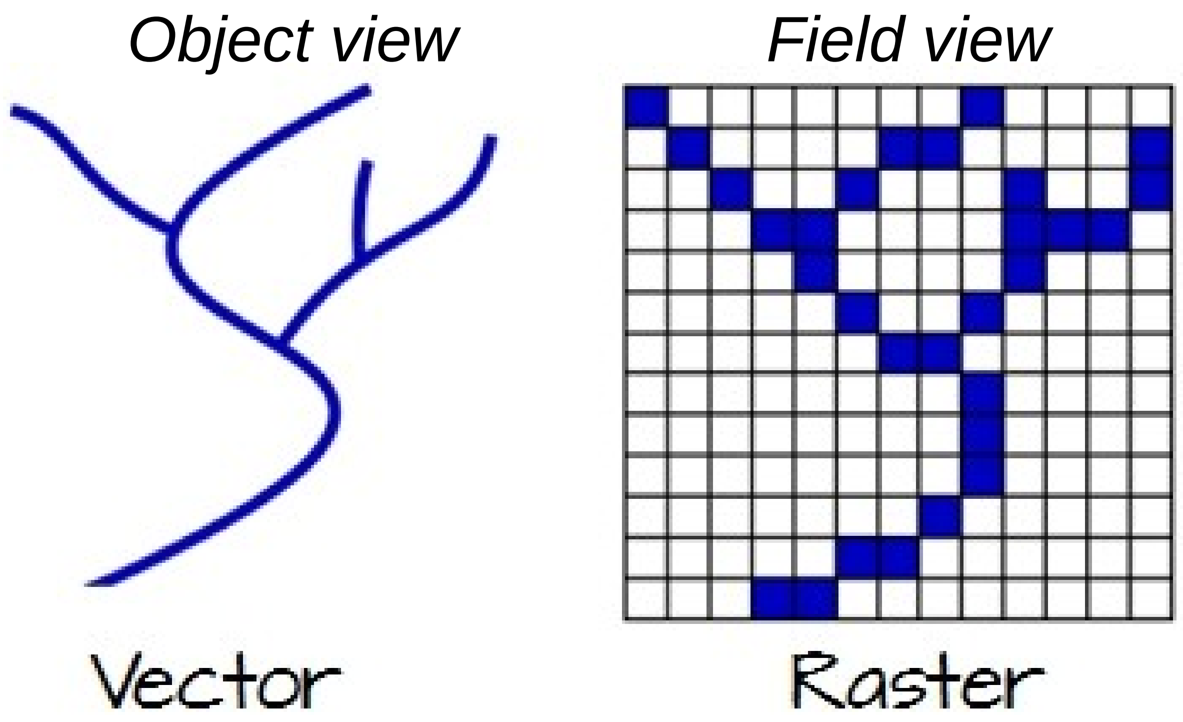


Field view



Basic data types

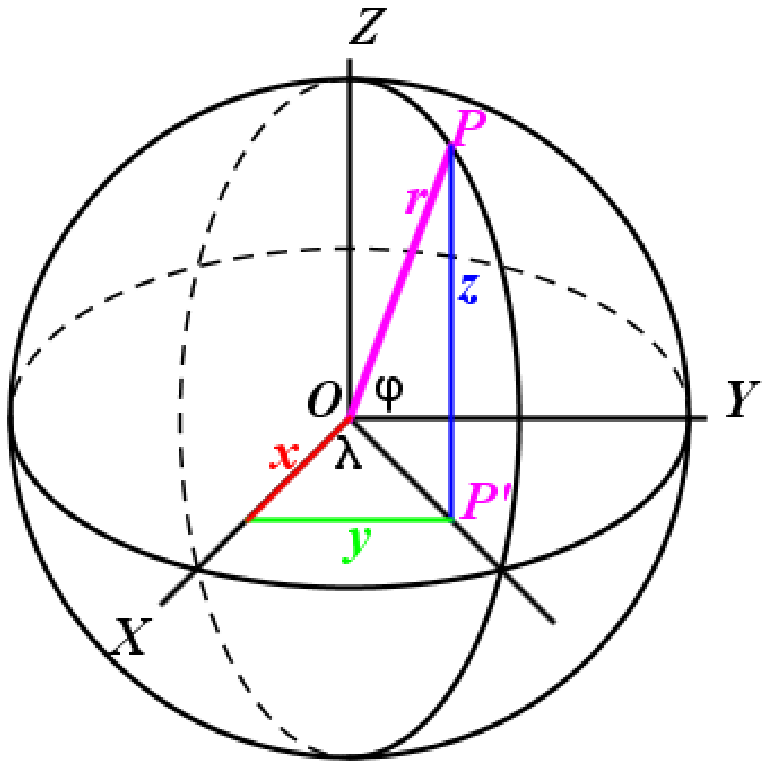
- Similar to computer graphics
- Sometimes in 3D

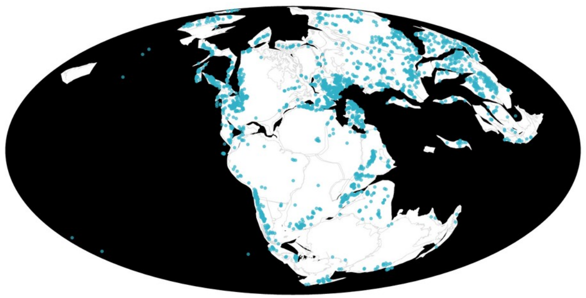


There is one twist:

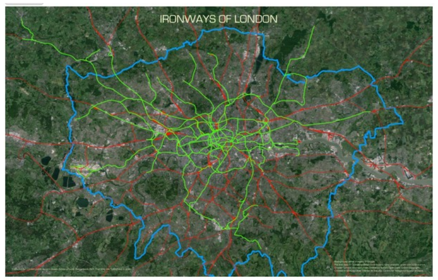
Vector data

- 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

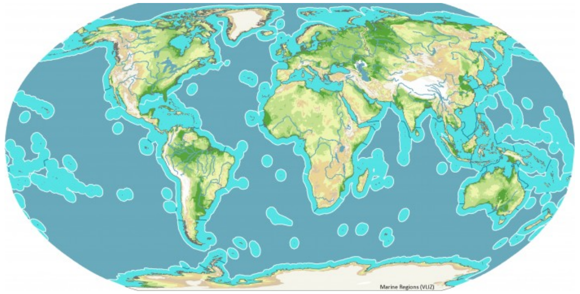




Points



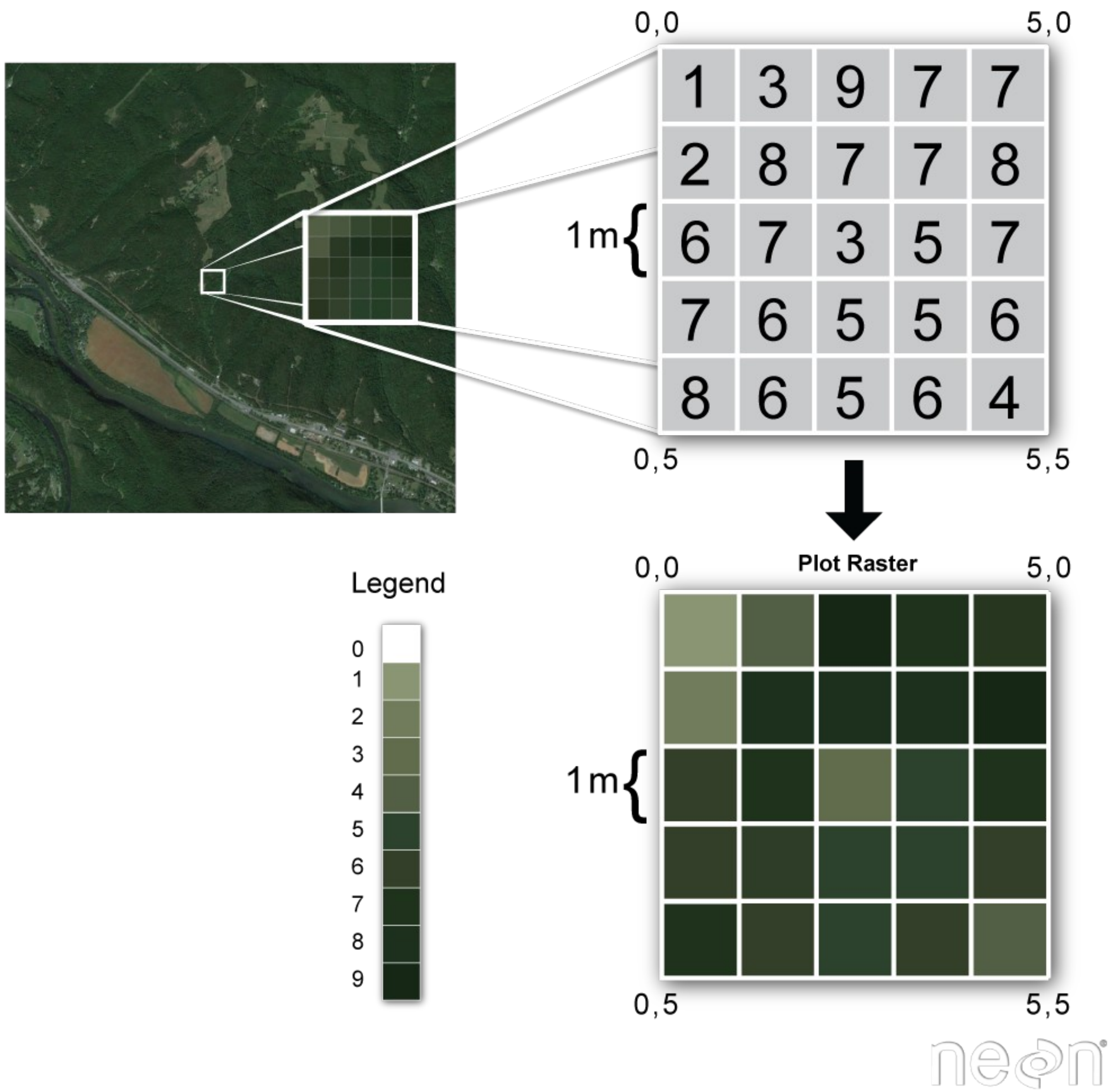
Lines



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

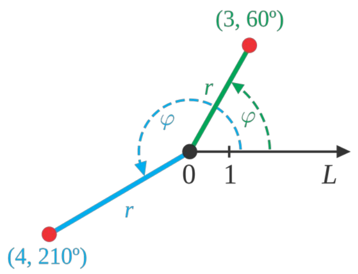


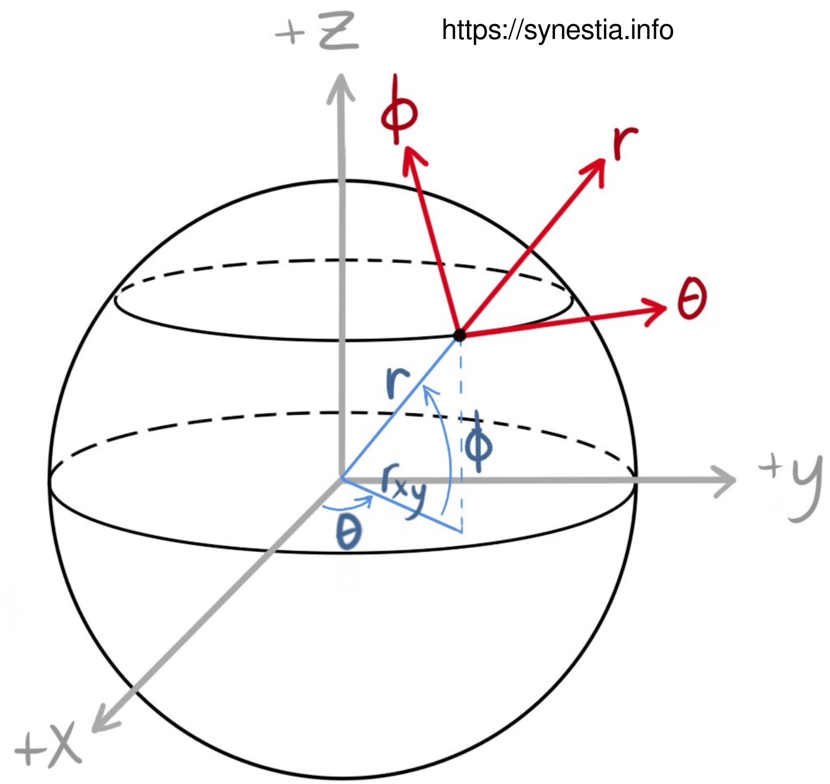
Coordinate Reference Systems

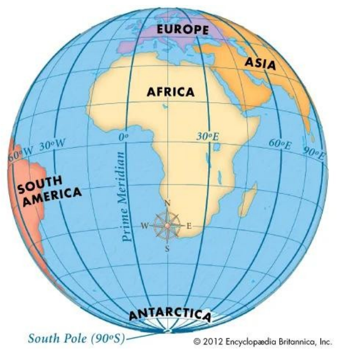
https://synestia.info

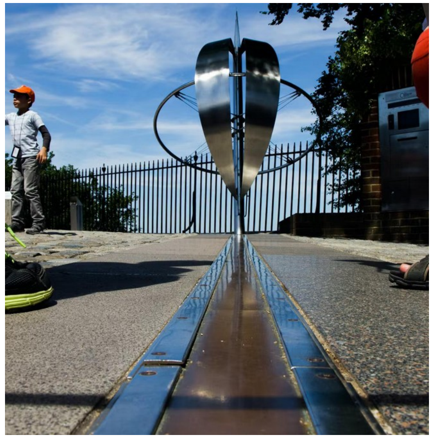
Geographic coordinates

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







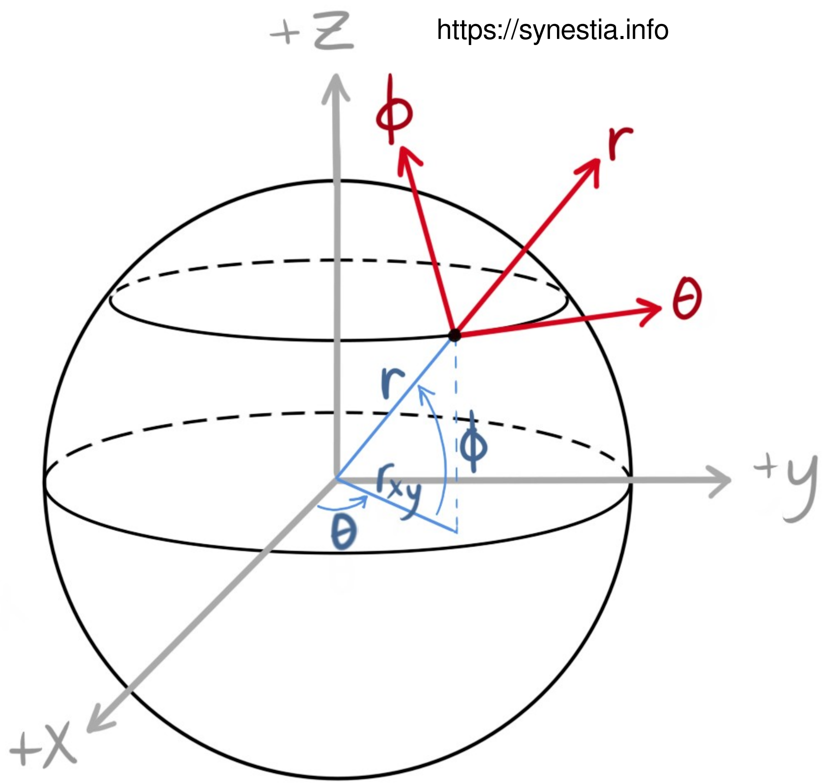


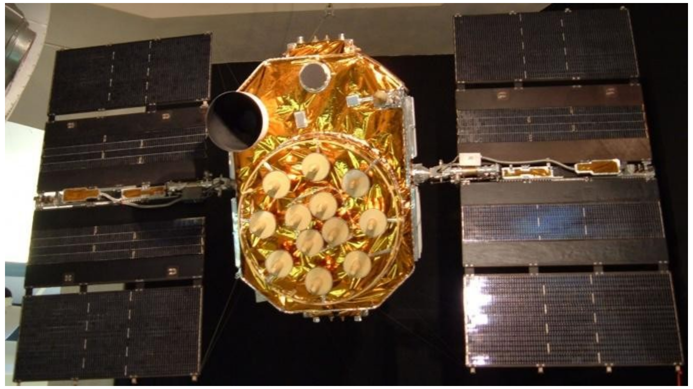
Coordinate Reference Systems

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

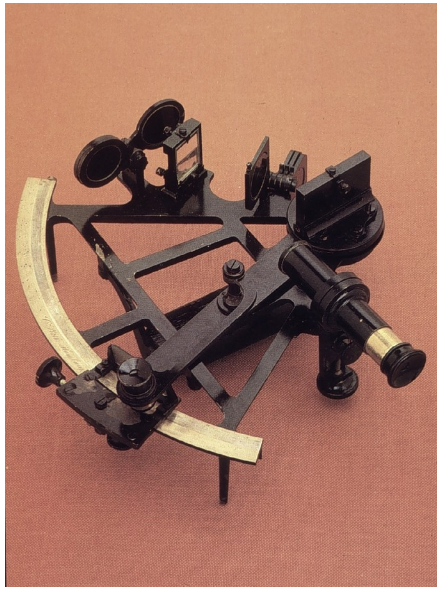


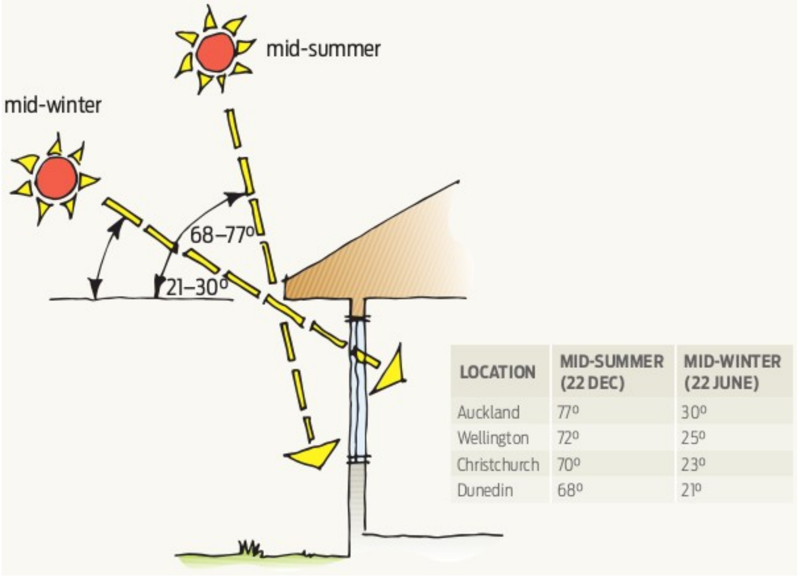


Coordinate Reference Systems

Projection based coordinate systems

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

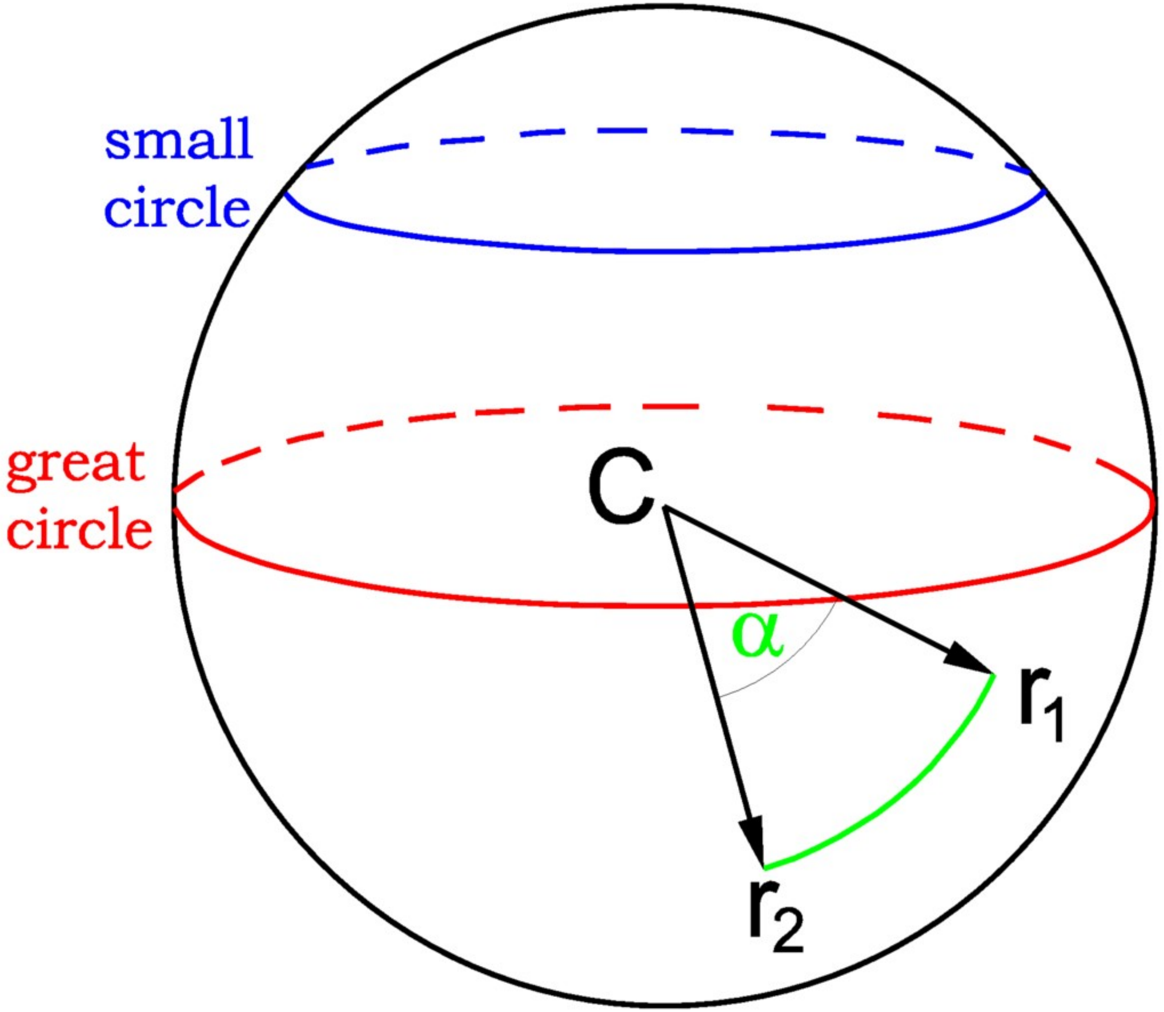




Spheres and Circles

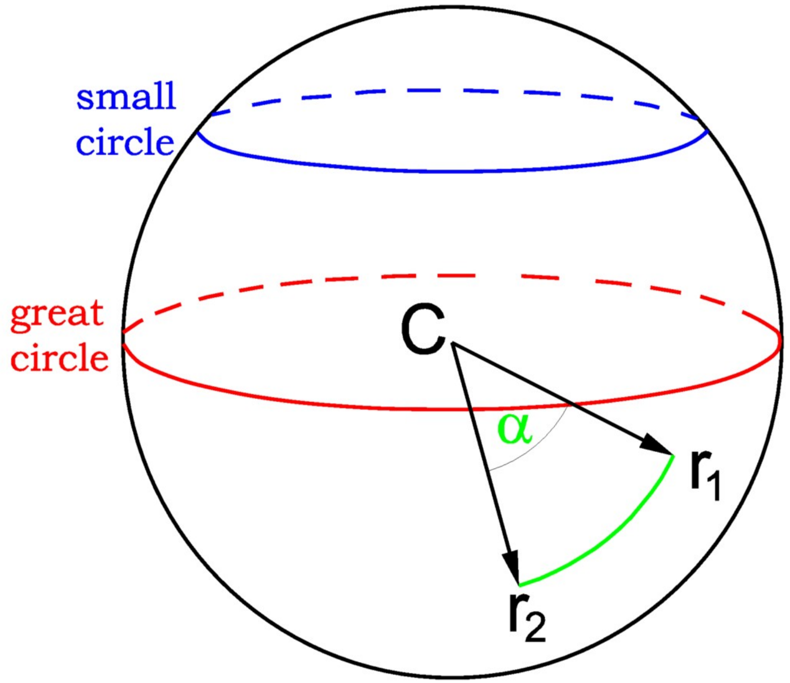
- There are no straight lines
- All points are connected by arcs, either along a great or small circles

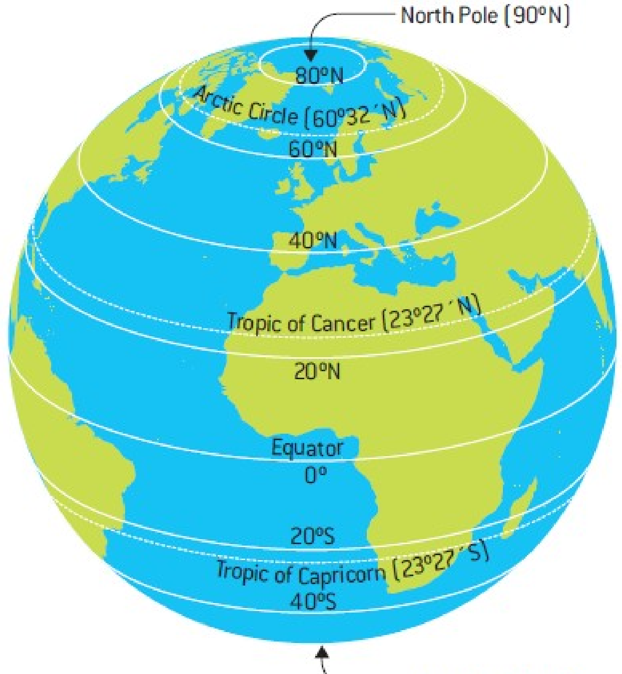
Path of the ISS



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:

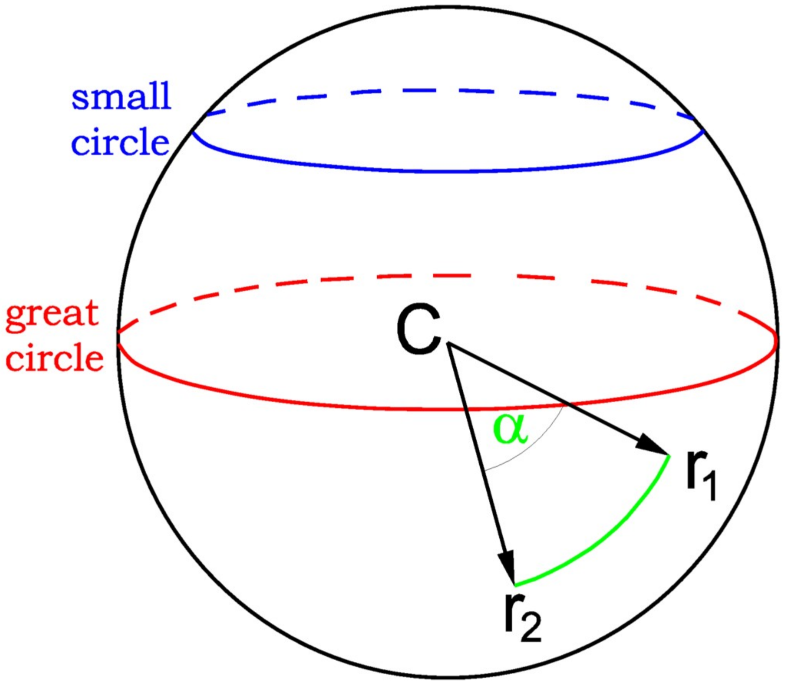




The parallels of latitude

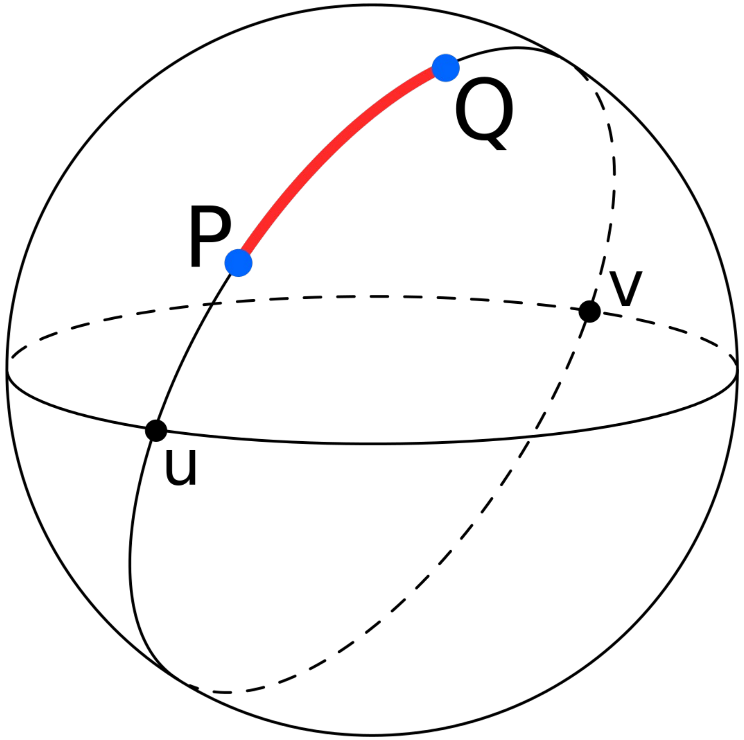
Longitude [-180-180˚]

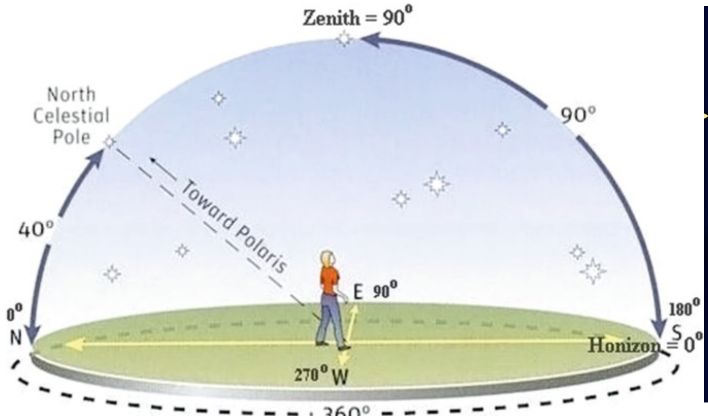
- Latitudinal circles: meridians
- All of them are great circles!
- Their distance decreases at the poles
- Direction: **meridional (v)**
- Notable circle:

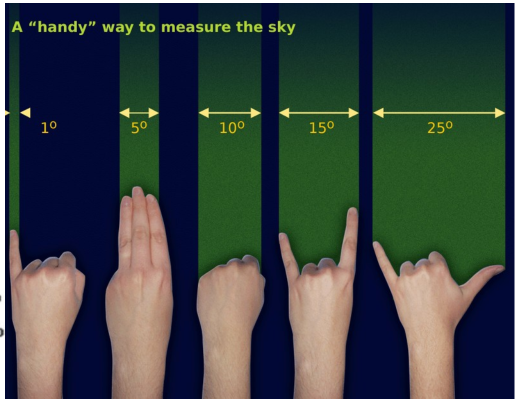


Distances

- 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

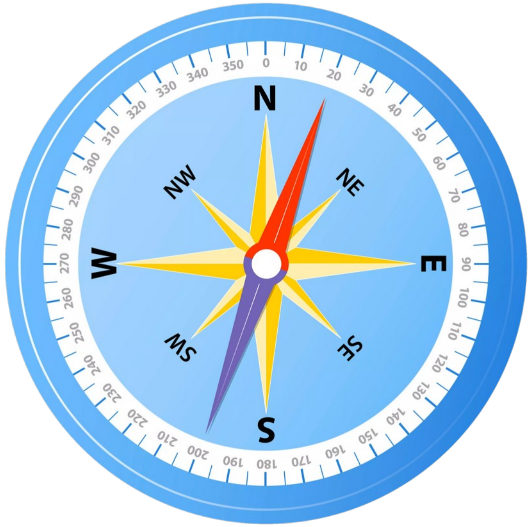






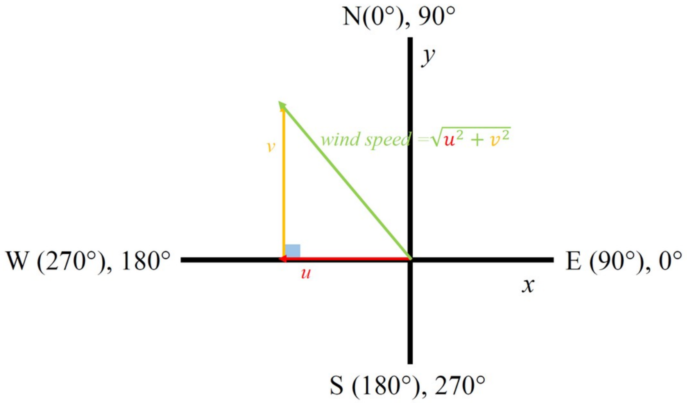
Directions

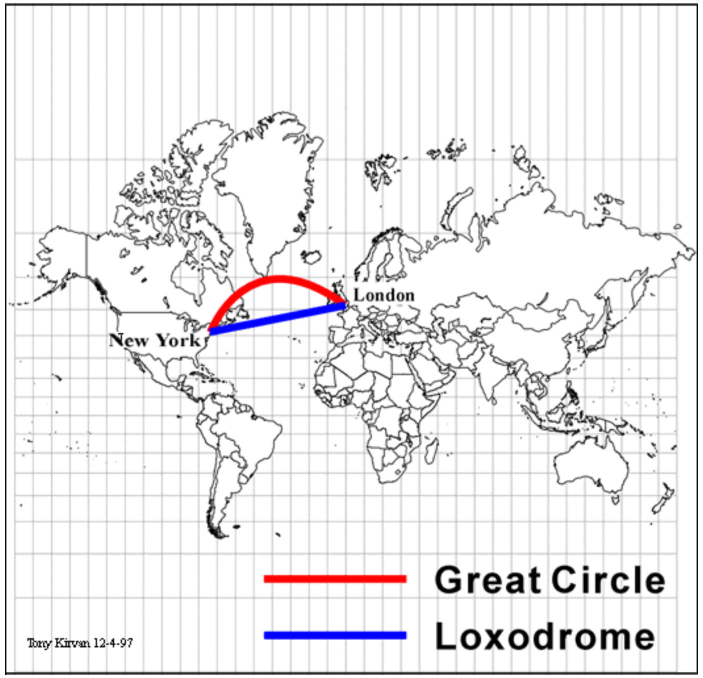
- Heading: angular distance to north
- Going straight: great circle
- Constant change of heading! (even without the true vs. magnetic difference)



Vectors

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

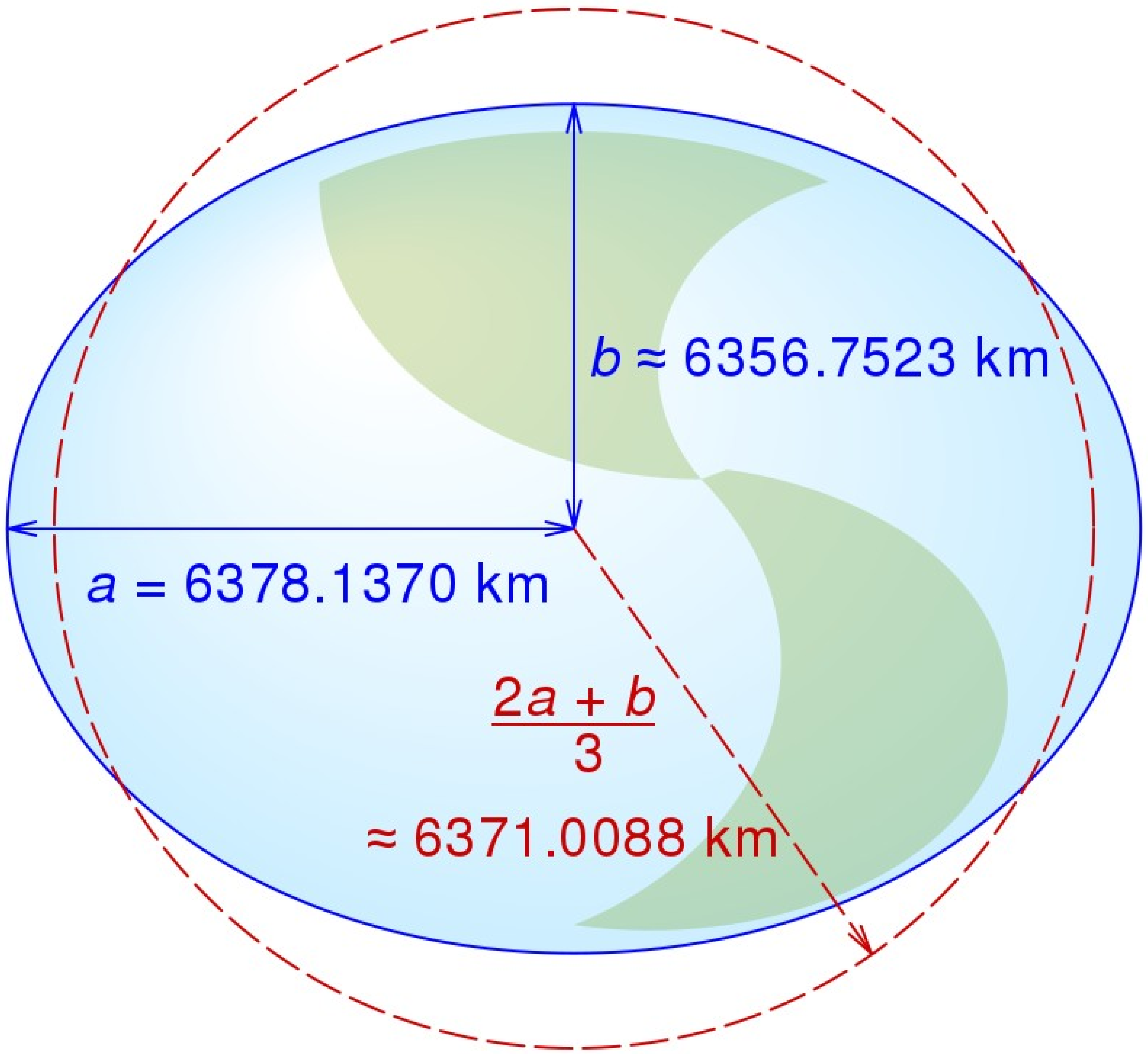




Mercator projection

The reference ellipsoid

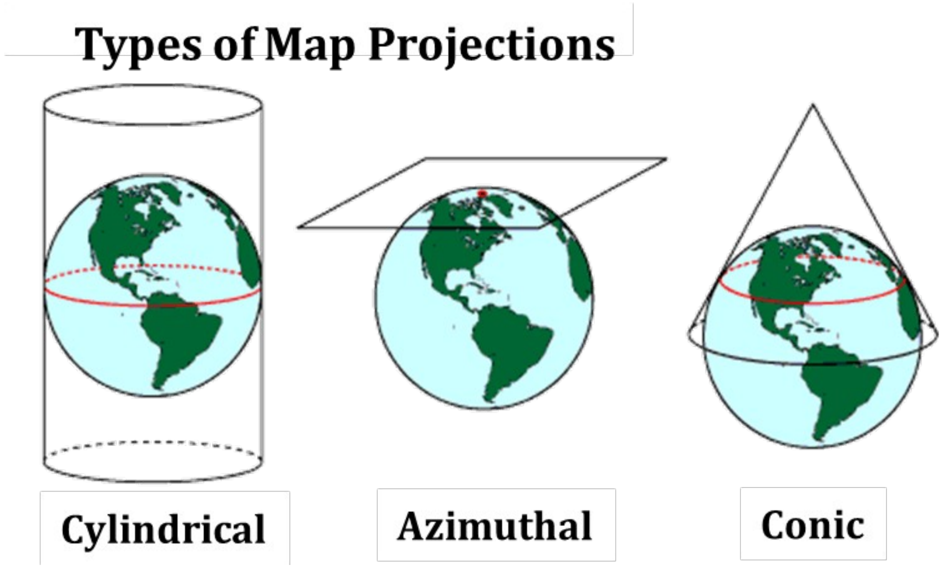
- Approximates the geoid
- Flattened rotational ellipsoid
- Used as a basis for elevation



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

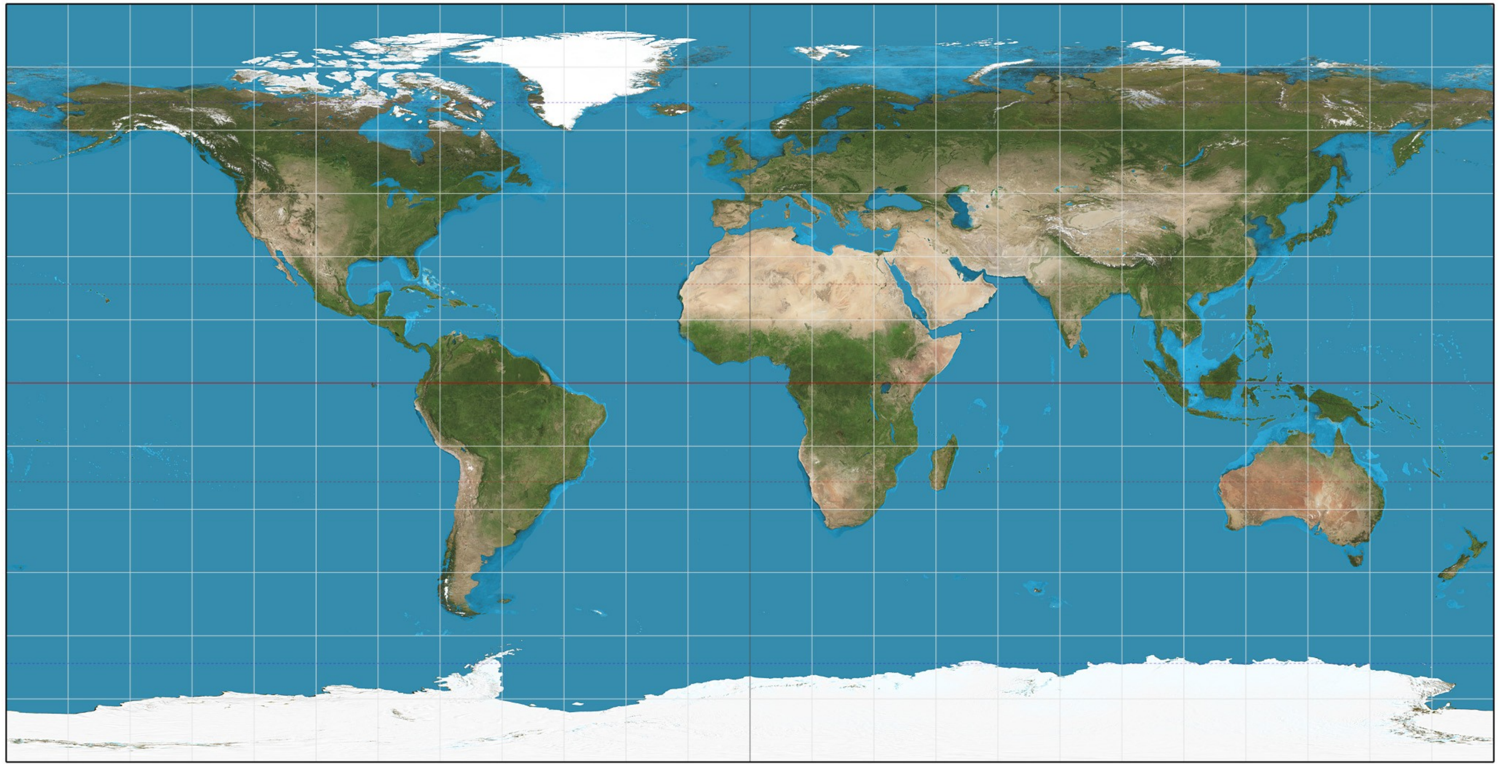


General types...

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

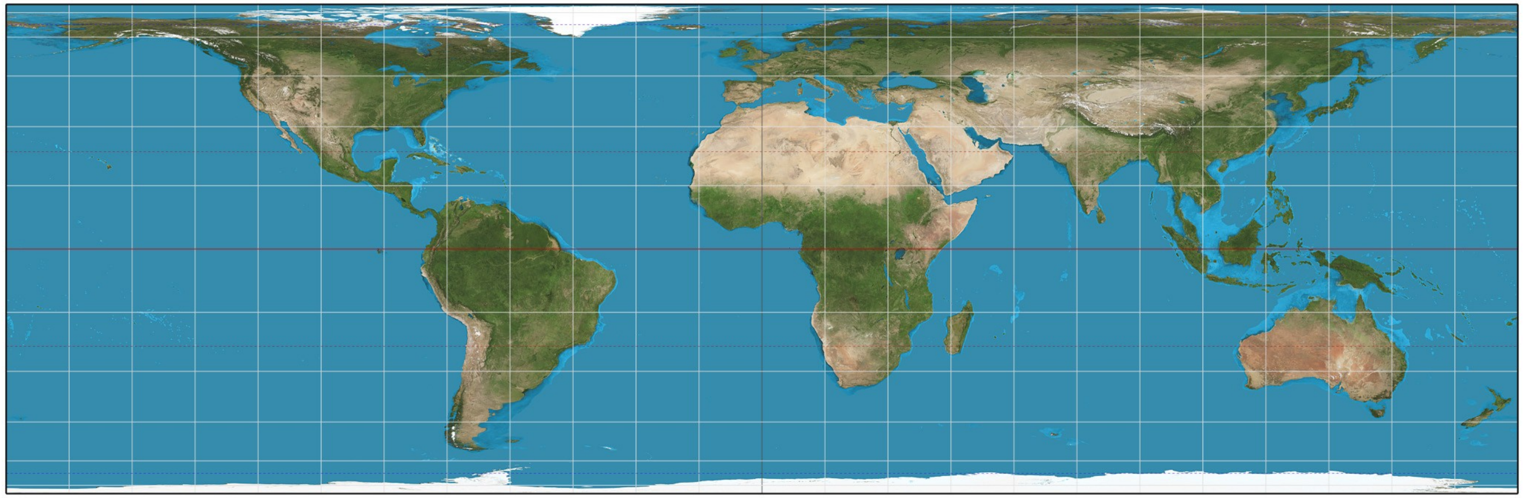
e.g. Mercator projection

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

Coordinate Reference Systems

e.g. Lambert's Cylindrical EA Projection

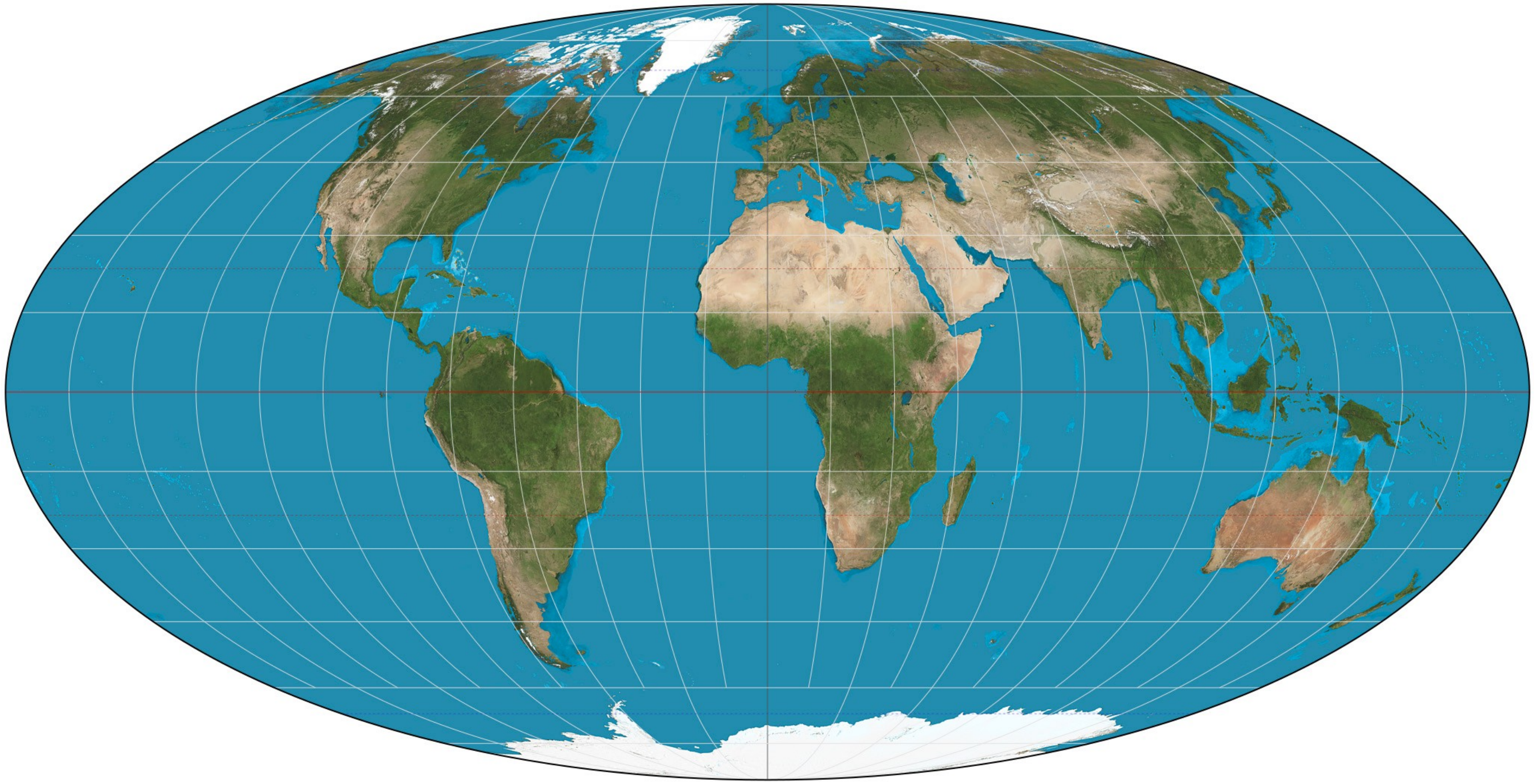
- Equal-area
- Highly distorts angles latitudinally



Coordinate Reference Systems

e.g. Mollweide Projection

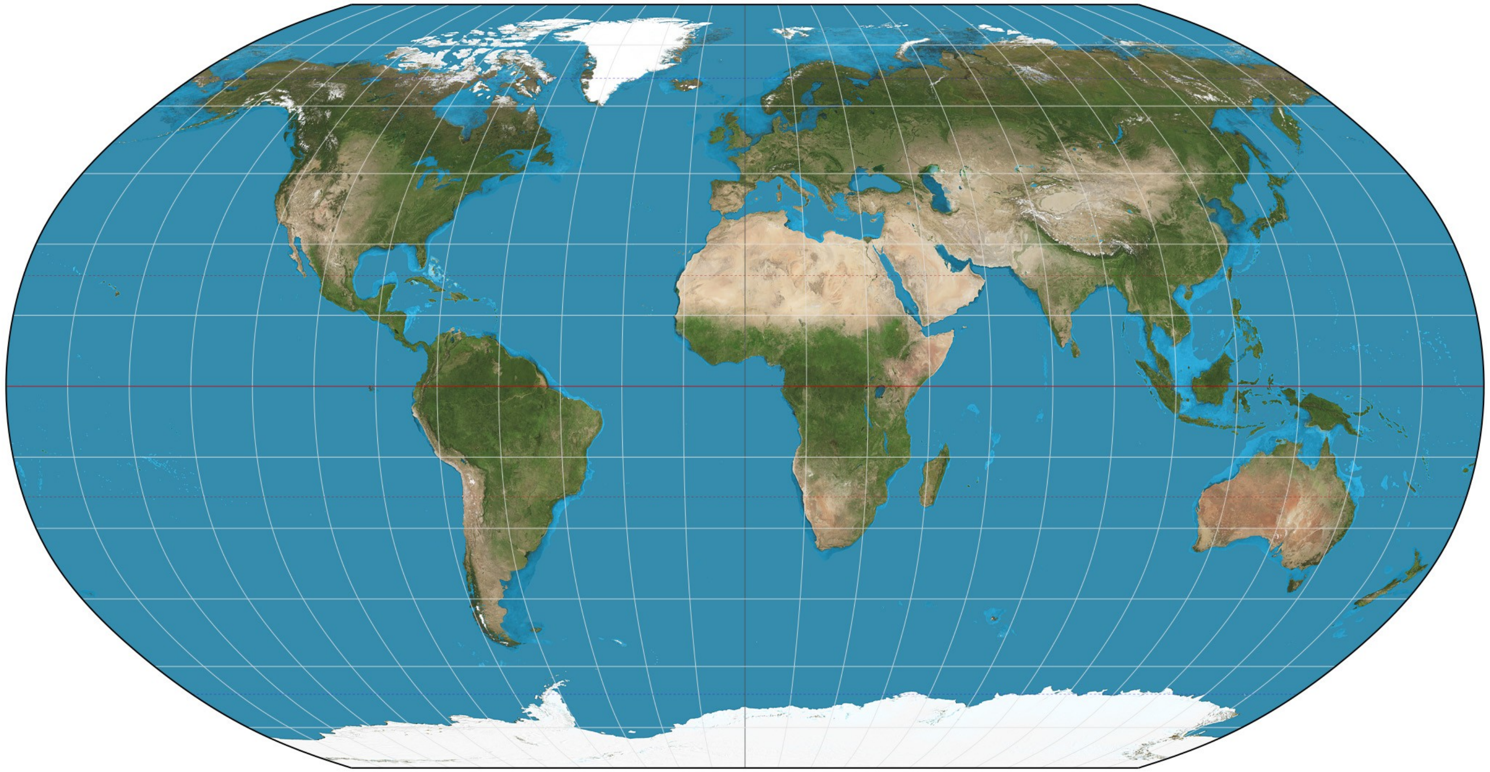
- Equal-area
- Highly distorts angles longitudinally



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

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

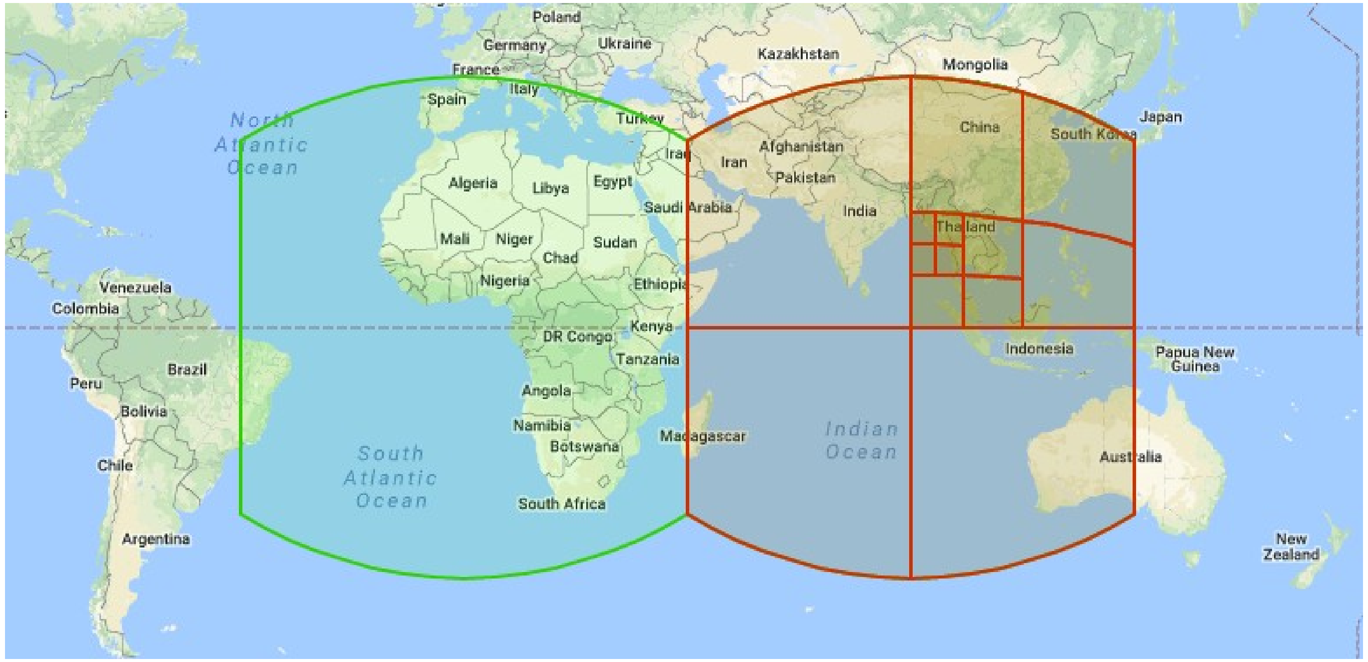
GEOS: Geometry Open Source

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

Open Source Libraries/Projects

S2 Geometry

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



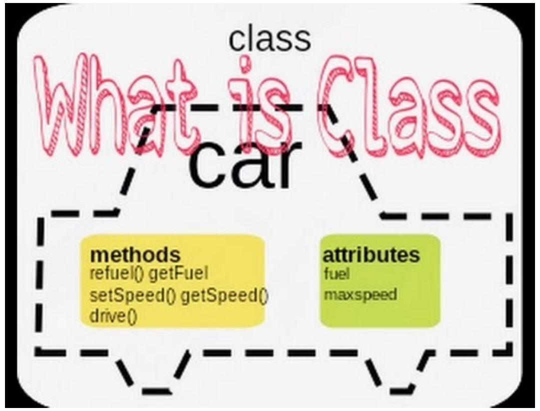
The R packages

- 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=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 list, again:

~~sp raster rgdal rgeos~~ terra sf ~~(stars)~~ geosphere

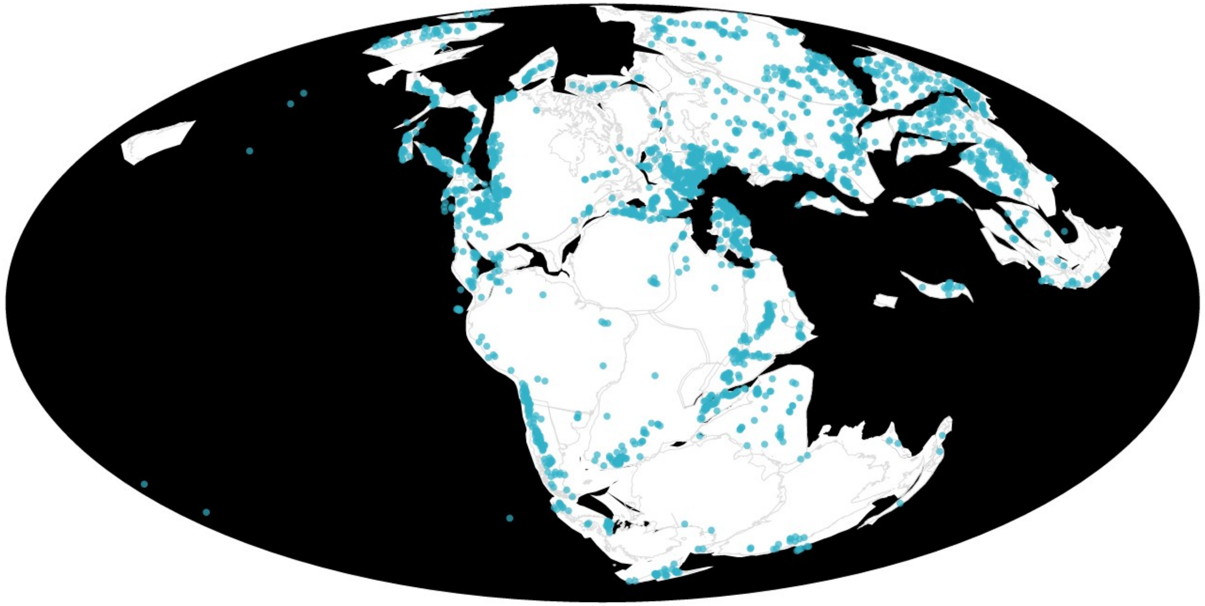
Vector data

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

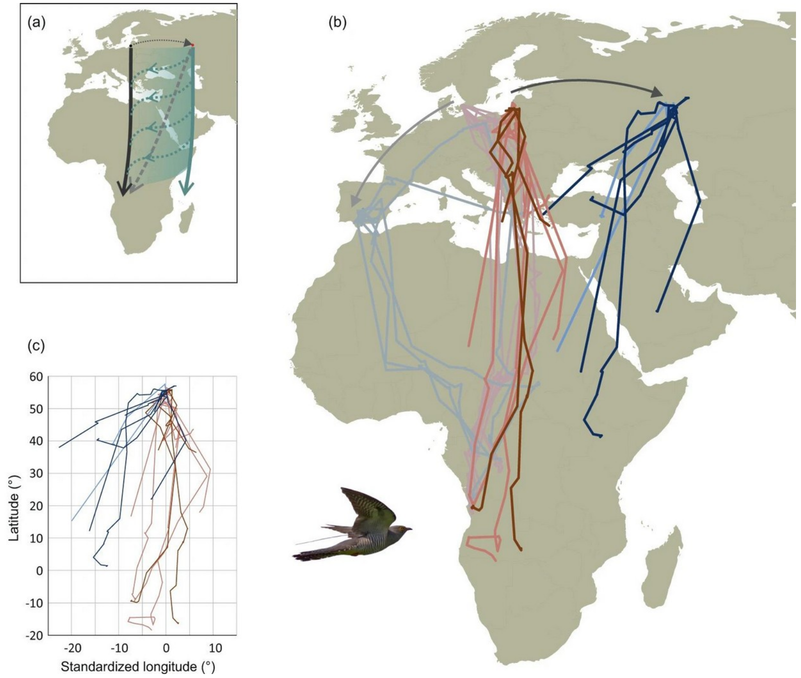


Fossil occurrences in the PBDB

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

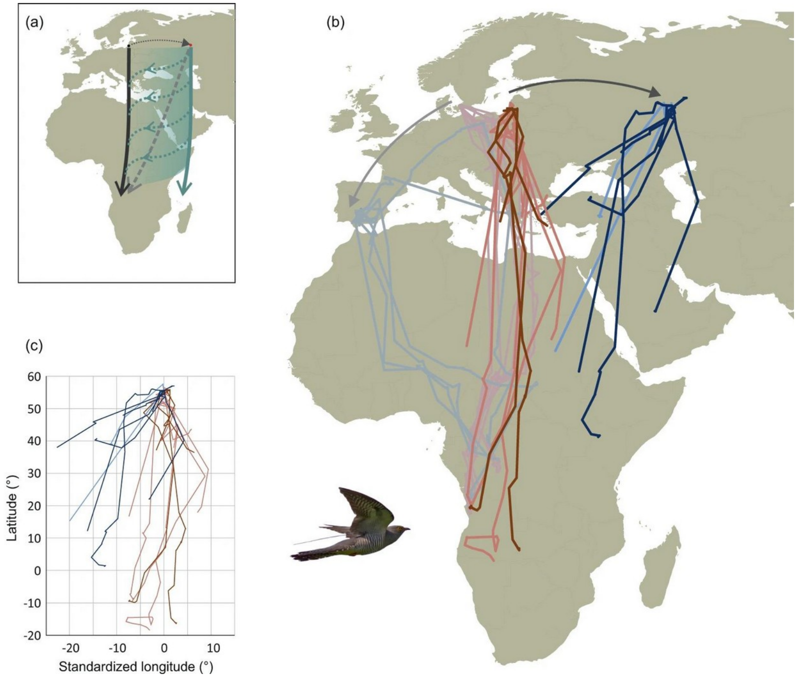


Thorup et al. 2020 Sci. Rep.

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

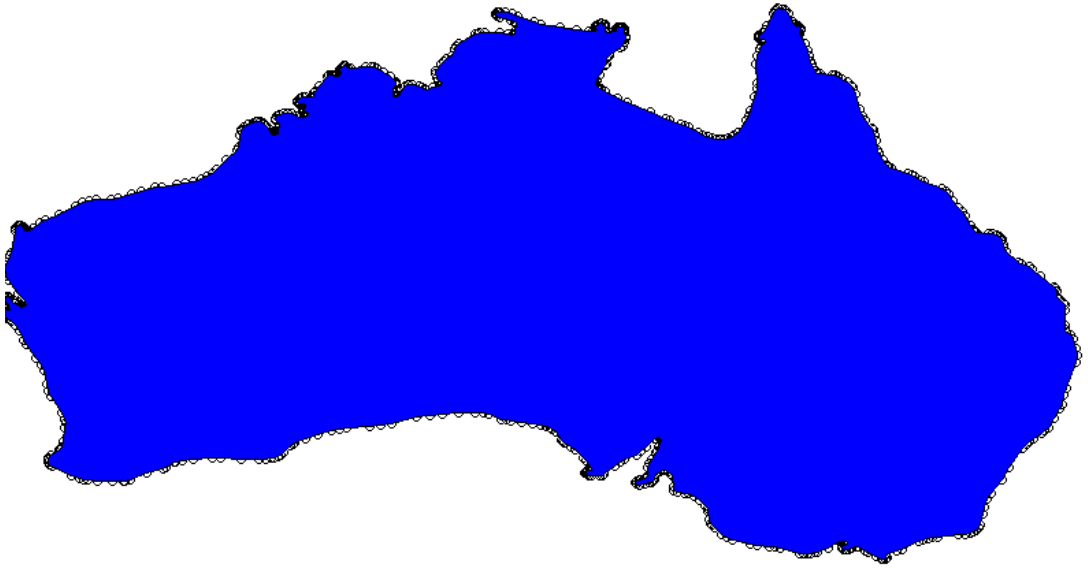


Thorup et al. 2020 Sci. Rep.

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

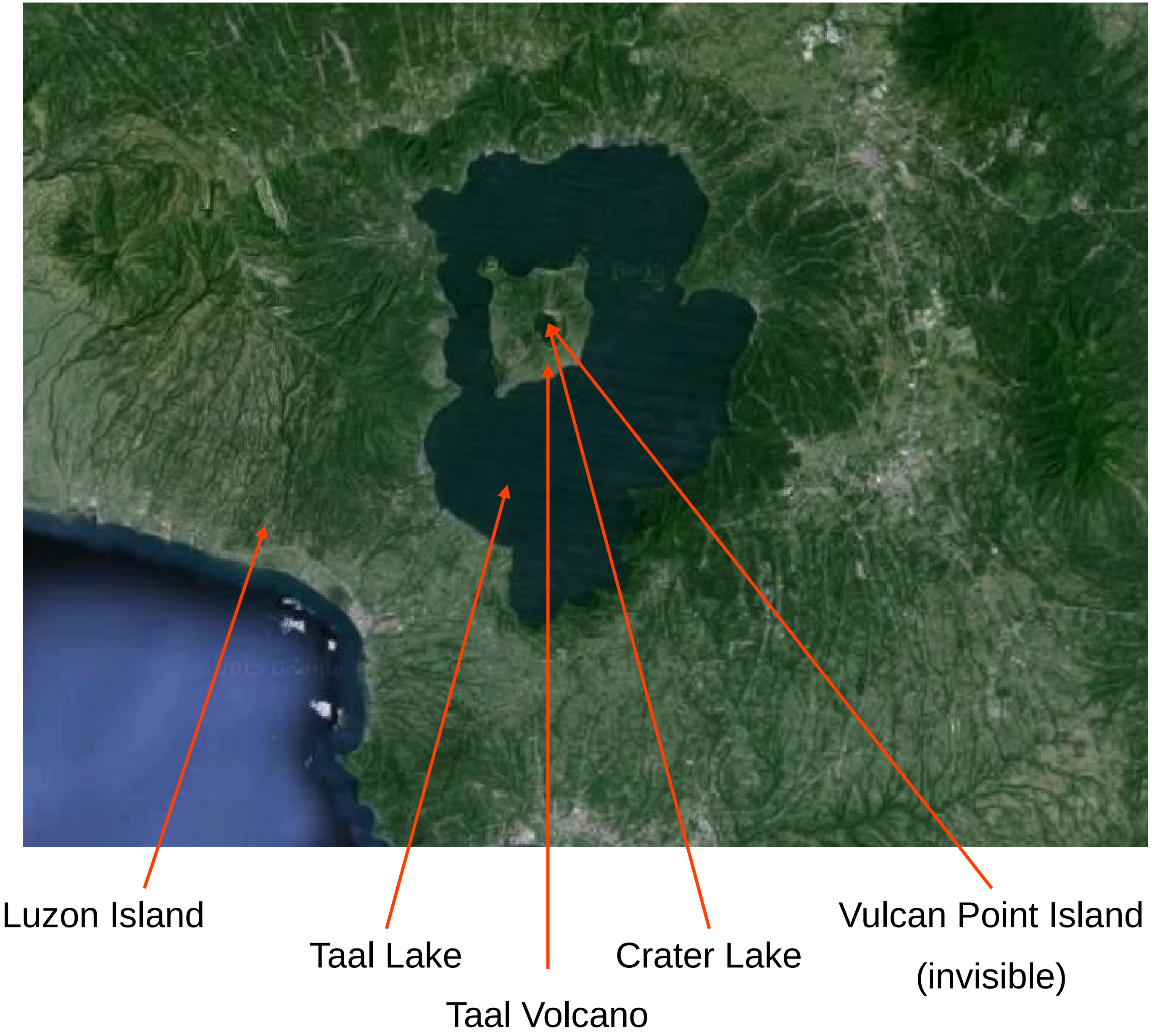
Feature types: Multiple polygons

- 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



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

adam@posidonia: ~ 80x62

```
> a <- rnorm(1:10)
> b <- rnorm(1:10)
> model <- lm(b ~ a)
> str(model)
List of 12
 $ coefficients : Named num [1:2] -0.4468 -0.0945
  ..- attr(*, "names")= chr [1:2] "(Intercept)" "a"
 $ residuals    : Named num [1:10] 0.0315 -0.0252 -0.8386 -0.3477 0.0826 ...
  ..- attr(*, "names")= chr [1:10] "1" "2" "3" "4" ...
 $ effects      : Named num [1:10] 1.434 -0.209 -0.838 -0.344 0.071 ...
  ..- attr(*, "names")= chr [1:10] "(Intercept)" "a" "" "" ...
 $ rank         : int 2
 $ fitted.values: Named num [1:10] -0.529 -0.424 -0.41 -0.384 -0.503 ...
  ..- attr(*, "names")= chr [1:10] "1" "2" "3" "4" ...
 $ assign       : int [1:2] 0 1
 $ qr           :List of 5
  ..$ qr   : num [1:10, 1:2] -3.162 0.316 0.316 0.316 0.316 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:10] "1" "2" "3" "4" ...
  .. .. ..$ : chr [1:2] "(Intercept)" "a"
  .. ..- attr(*, "assign")= int [1:2] 0 1
  ..$ qraux: num [1:2] 1.32 1.23
  ..$ pivot: int [1:2] 1 2
  ..$ tol  : num 1e-07
  ..$ rank : int 2
  ..- attr(*, "class")= chr "qr"
 $ df.residual  : int 8
 $ xlevels      : Named list()
 $ call         : language lm(formula = b ~ a)
 $ terms        :Classes 'terms', 'formula'  language b ~ a
  .. ..- attr(*, "variables")= language list(b, a)
  .. ..- attr(*, "factors")= int [1:2, 1] 0 1
  .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. ..$ : chr [1:2] "b" "a"
  .. .. .. ..$ : chr "a"
  .. ..- attr(*, "term.labels")= chr "a"
  .. ..- attr(*, "order")= int 1
  .. ..- attr(*, "intercept")= int 1
  .. ..- attr(*, "response")= int 1
  .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
  .. ..- attr(*, "predvars")= language list(b, a)
  .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
  .. .. ..- attr(*, "names")= chr [1:2] "b" "a"
 $ model        :'data.frame':  10 obs. of  2 variables:
  ..$ b: num [1:10] -0.497 -0.449 -1.249 -0.732 -0.42 ...
  ..$ a: num [1:10] 0.868 -0.246 -0.384 -0.665 0.595 ...
  ..- attr(*, "terms")=Classes 'terms', 'formula'  language b ~ a
  .. .. ..- attr(*, "variables")= language list(b, a)
  .. .. ..- attr(*, "factors")= int [1:2, 1] 0 1
  .. .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. .. ..$ : chr [1:2] "b" "a"
  .. .. .. .. ..$ : chr "a"
  .. .. ..- attr(*, "term.labels")= chr "a"
  .. .. ..- attr(*, "order")= int 1
  .. .. ..- attr(*, "intercept")= int 1
  .. .. ..- attr(*, "response")= int 1
  .. .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
  .. .. ..- attr(*, "predvars")= language list(b, a)
  .. .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
  .. .. .. ..- attr(*, "names")= chr [1:2] "b" "a"
 - attr(*, "class")= chr "lm"
>
```

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!

```
> str(spp)
Formal class 'SpatialPolygons' [package "sp"] with 4 slots
  ..@ polygons   :List of 1
  .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
  .. .. .. ..@ Polygons :List of 2
  .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
  .. .. .. .. .. .. ..@ labpt  : num [1:2] 134.4 -25.6
  .. .. .. .. .. .. ..@ area   : num 685
  .. .. .. .. .. .. ..@ hole   : logi FALSE
  .. .. .. .. .. .. ..@ ringDir: int 1
  .. .. .. .. .. .. ..@ coords : num [1:637, 1:2] 142 143 143 143 143 ...
  .. .. .. .. .. .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. .. .. .. .. .. ..$ : NULL
  .. .. .. .. .. .. .. .. ..$ : chr [1:2] "long" "lat"
  .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
  .. .. .. .. .. .. ..@ labpt  : num [1:2] 63.8 36.8
  .. .. .. .. .. .. ..@ area   : num 8883
  .. .. .. .. .. .. ..@ hole   : logi FALSE
  .. .. .. .. .. .. ..@ ringDir: int 1
  .. .. .. .. .. .. ..@ coords : num [1:7337, 1:2] 180 180 180 180 179 ...
  .. .. .. .. .. .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. .. .. .. .. .. ..$ : NULL
  .. .. .. .. .. .. .. .. ..$ : chr [1:2] "V1" "V2"
  .. .. .. ..@ plotOrder: int [1:2] 2 1
  .. .. .. ..@ labpt    : num [1:2] 63.8 36.8
  .. .. .. ..@ ID       : chr "0"
  .. .. .. ..@ area     : num 9569
  ..@ plotOrder  : int 1
  ..@ bbox       : num [1:2, 1:2] -17.4 -39 180.4 77.7
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:2] "x" "y"
  .. .. ..$ : chr [1:2] "min" "max"
  ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
  .. .. ..@ projargs: chr NA
```

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

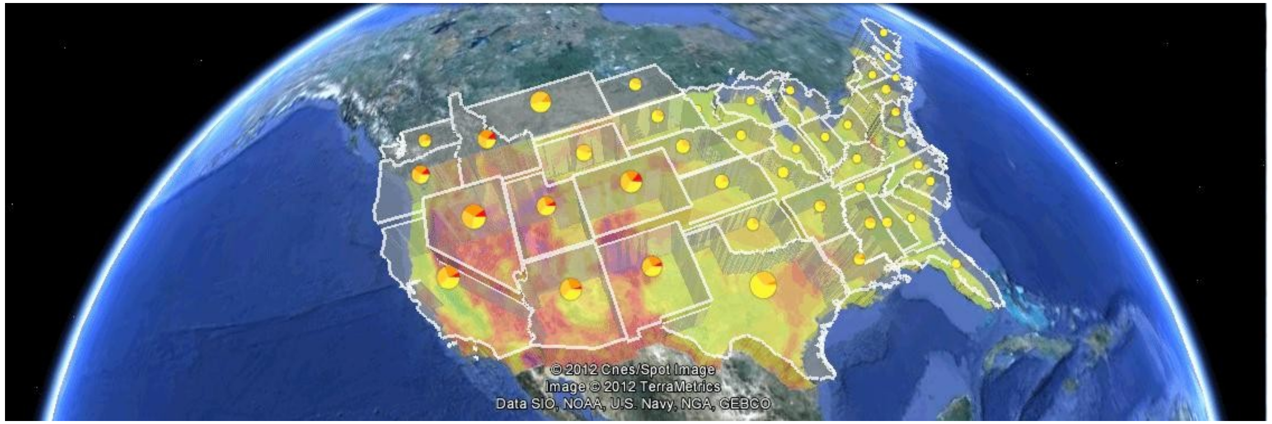
ESRI Shapefiles

- 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

Open Geospatial Consortium

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

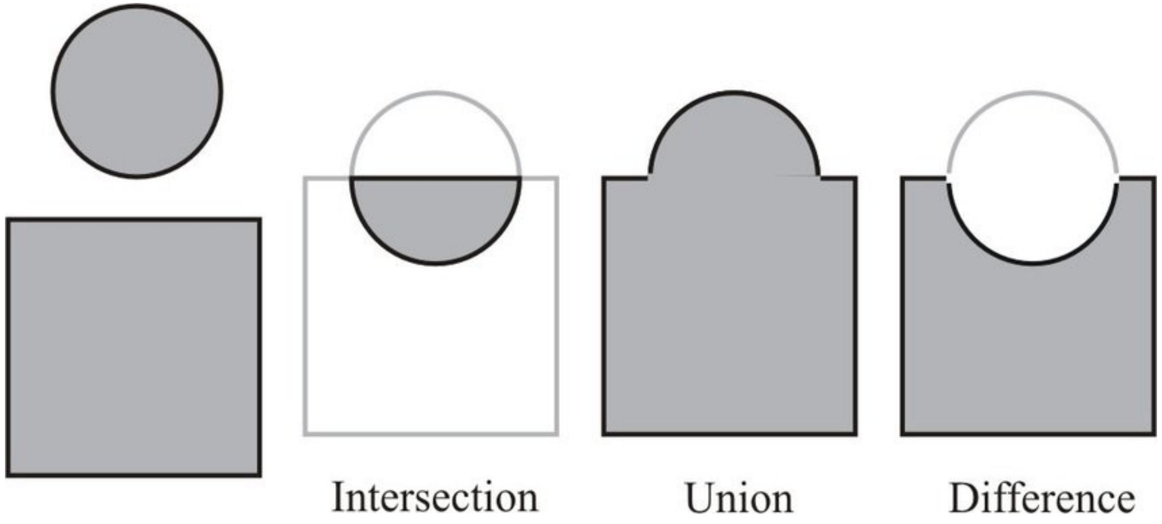
- Database-oriented (PostgreSQL)

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

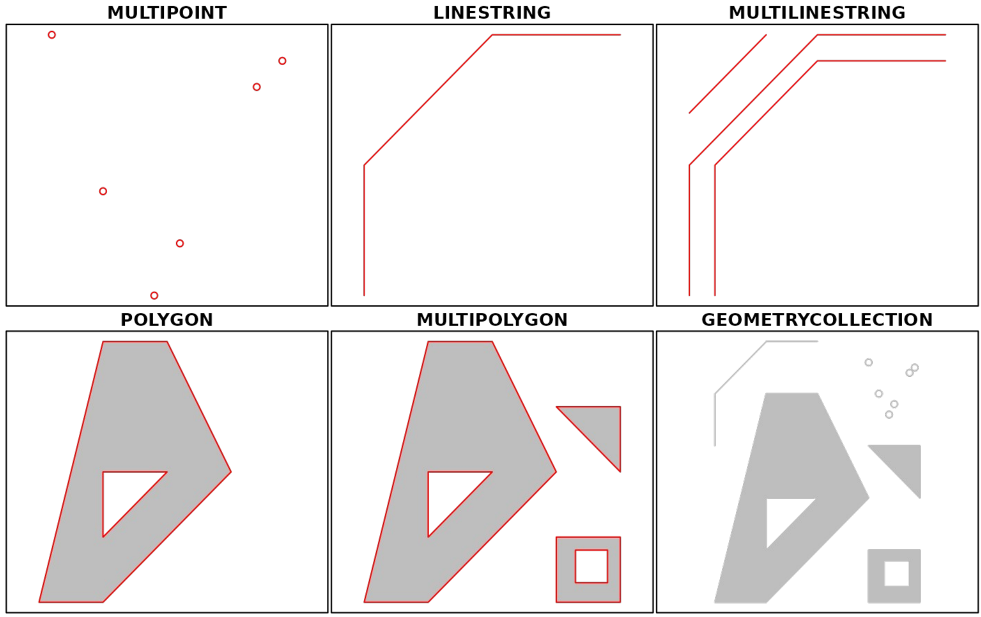
Package rgeos: Interface to Geometry Engine - Open Source ('GEOS')

- Primarily by Roger Bivand
- Interface to GEOS
- Geometric calculations in the same CRS (e.g. Area)
- Boolean operations with shapes



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



NEW

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

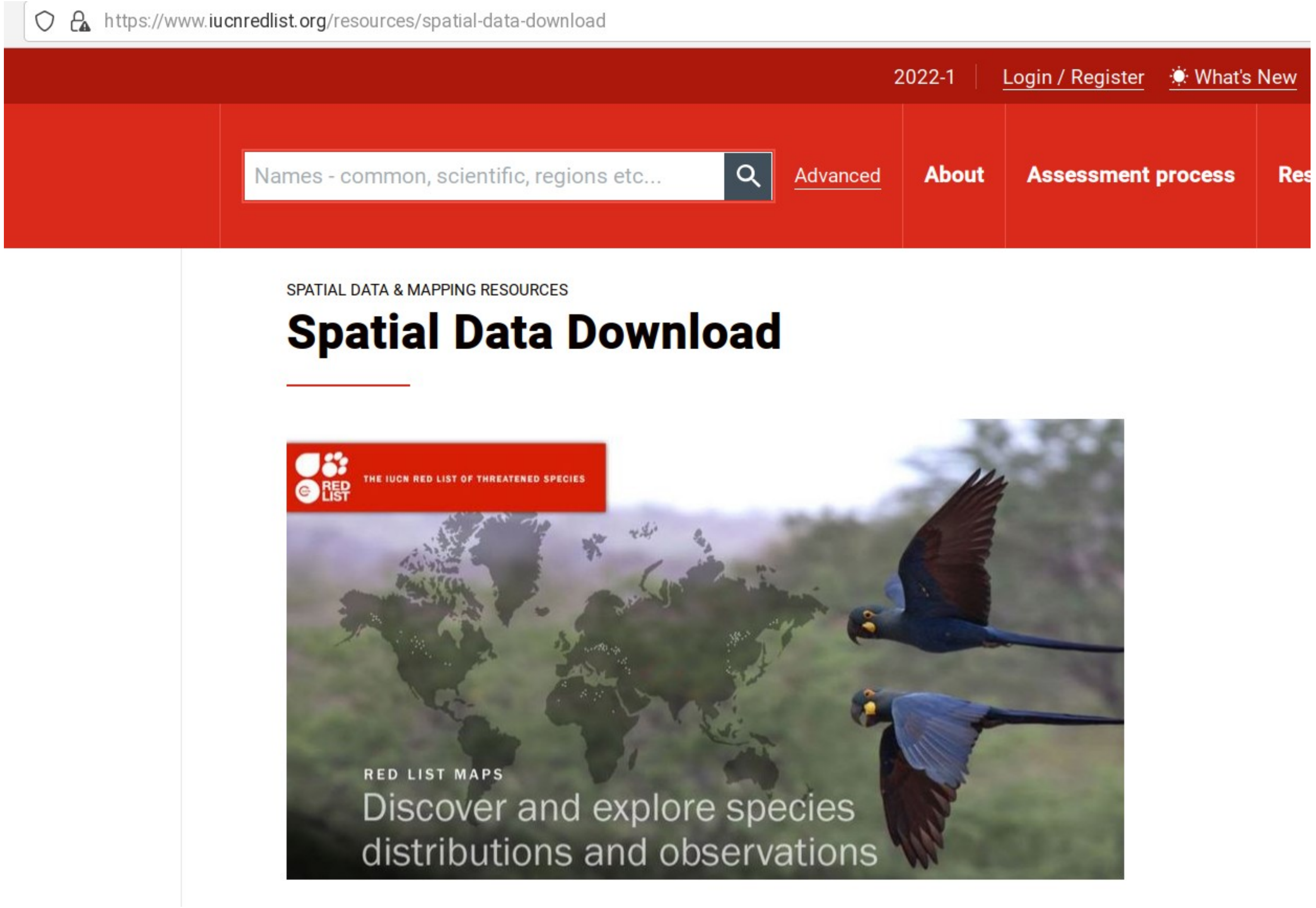
Information | Table | Map | Analyze | Export | API

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

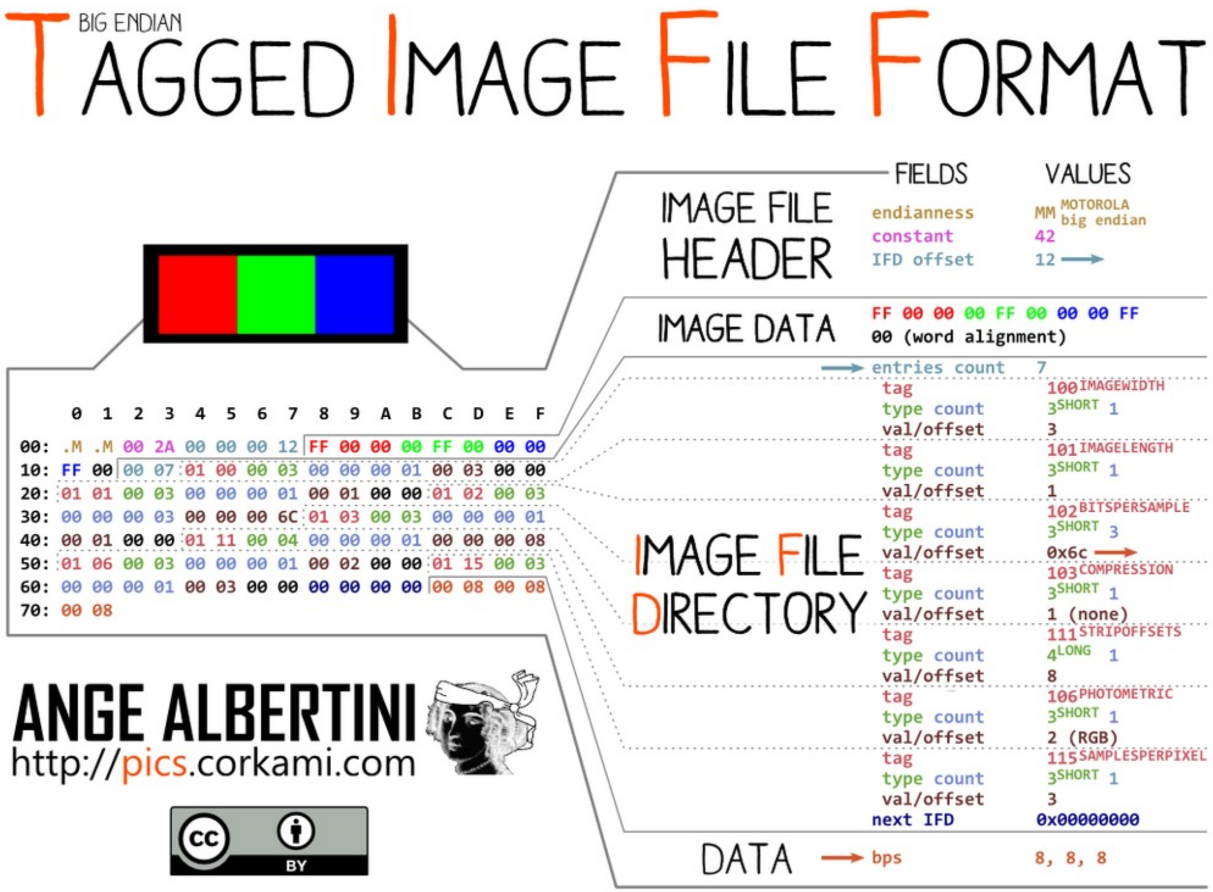
- Single uncompressed layer, a table of values
- Not properly georeferenced
- Column and row names might help

```
102,102,103,103,103,103,103,102,102,102,103,103,104,104,104,104,105,106,105,104,104,105,106,107,108,110,111,113,1
103,103,104,104,104,104,104,103,103,103,103,104,104,104,105,105,106,107,106,106,106,107,108,110,111,114,117,118,1
104,104,105,105,105,105,105,104,104,103,104,104,105,105,105,106,107,108,108,108,109,110,112,114,115,118,121,122,1
105,105,105,106,106,106,106,105,105,104,104,105,105,106,106,107,109,110,110,112,113,115,116,118,119,121,124,126,1
105,106,106,107,107,107,107,106,106,105,105,106,106,107,108,109,111,113,114,116,118,120,121,122,123,125,127,129,1
106,107,107,108,108,108,108,107,107,106,106,107,108,108,110,113,115,117,118,120,122,124,125,127,128,129,131,134,1
107,108,108,109,109,109,109,108,108,107,108,108,110,111,113,116,118,120,123,125,127,129,130,132,134,135,137,139,1
,108,109,109,110,110,110,110,109,109,108,110,110,113,116,118,120,122,125,127,129,133,136,138,140,141,142,148,150,
,109,110,110,111,111,111,111,110,110,110,112,114,118,121,123,125,127,129,133,137,141,143,145,146,148,150,154,156,
,110,110,111,113,112,111,113,112,112,114,116,119,121,124,127,129,133,138,143,146,149,149,151,153,154,157,159,160,
,110,111,113,115,114,113,114,114,115,117,119,121,124,126,129,133,140,145,150,154,155,155,157,159,161,162,164,165,
,111,113,115,117,116,115,116,117,117,119,121,124,126,128,132,137,143,151,156,161,161,162,163,165,166,167,168,170,
,114,115,117,117,117,118,119,119,120,121,124,126,128,131,137,143,150,156,160,163,165,168,170,171,172,173,174,175,
,116,118,118,118,120,121,121,122,122,123,125,128,130,134,141,147,152,156,160,165,168,170,174,176,179,180,181,181,
,118,120,120,121,122,123,124,124,125,126,127,129,132,135,142,149,153,157,161,166,170,174,178,180,182,183,184,184,
,120,121,122,123,124,125,126,127,127,128,130,132,134,137,142,151,155,158,162,169,172,176,181,183,184,186,187,188,
,120,122,125,126,126,127,128,129,130,130,132,134,136,139,145,152,157,160,167,172,175,178,181,185,186,188,190,191,
,121,124,126,128,129,129,130,131,132,133,135,137,139,143,150,154,159,164,170,173,176,179,184,186,189,190,191,192,
,122,125,127,130,130,131,133,134,135,136,137,140,143,147,154,158,162,166,171,174,177,181,186,189,190,190,191,192,
,122,125,128,130,132,133,135,136,137,139,140,143,147,152,157,161,164,168,172,175,179,182,186,190,190,190,190,189,
,123,126,129,131,133,135,137,138,139,141,143,147,150,156,161,164,167,170,173,177,181,184,187,188,190,189,187,185,
,124,127,130,132,135,137,138,140,142,144,147,149,154,157,161,165,168,171,175,178,181,184,186,187,187,184,184,181,
,123,128,131,133,136,138,140,142,144,146,149,151,154,157,160,164,168,172,175,178,181,183,184,184,185,183,180,177,
,123,127,131,134,136,138,140,142,144,147,149,151,154,157,160,164,168,171,174,178,180,181,181,182,183,181,178,173,
,120,124,128,131,134,137,139,142,144,146,149,151,153,156,160,163,167,171,174,178,180,180,180,180,180,180,175,171,
,118,121,125,129,132,134,137,140,142,145,147,149,151,155,159,163,166,169,173,177,179,180,180,180,180,179,174,169,
,117,120,121,125,129,132,135,138,140,143,145,147,149,153,157,160,163,166,171,174,177,179,180,180,180,179,172,168,
,115,118,120,122,126,130,133,136,138,141,143,145,148,151,154,157,160,163,168,171,174,177,179,179,179,176,171,167,
,114,116,118,120,122,127,131,133,136,138,141,143,146,148,151,154,157,160,164,168,171,174,178,178,179,177,173,169,
,115,114,116,118,120,122,127,129,132,136,139,141,143,146,148,151,153,156,160,164,167,172,174,176,177,176,173,170,
```

Raster File formats

GeoTIFF

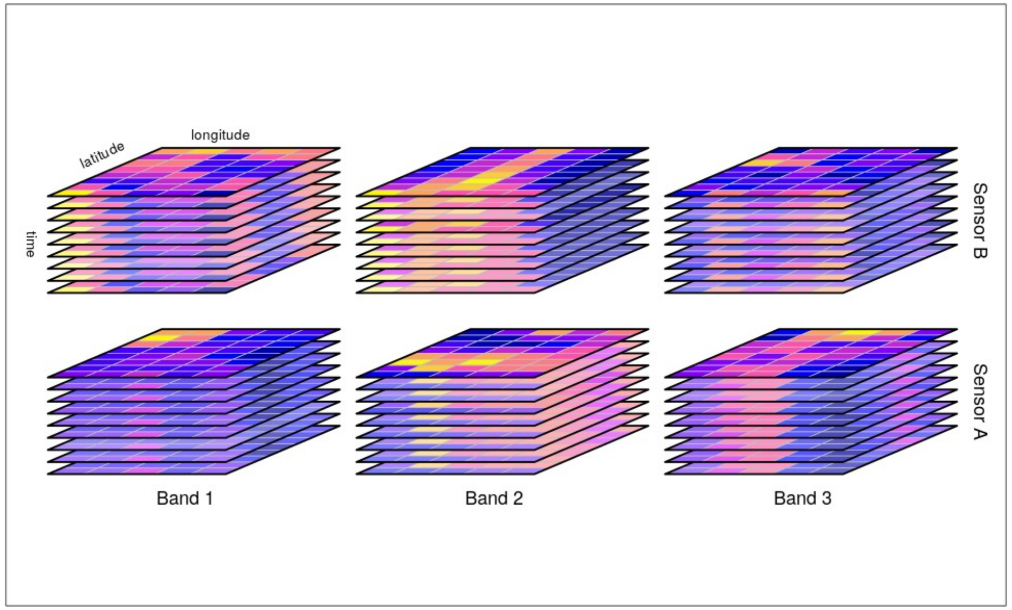
- Single uncompressed layer
- Completely open source
- TIFF image + georeferencing



Raster File formats

NetCDF

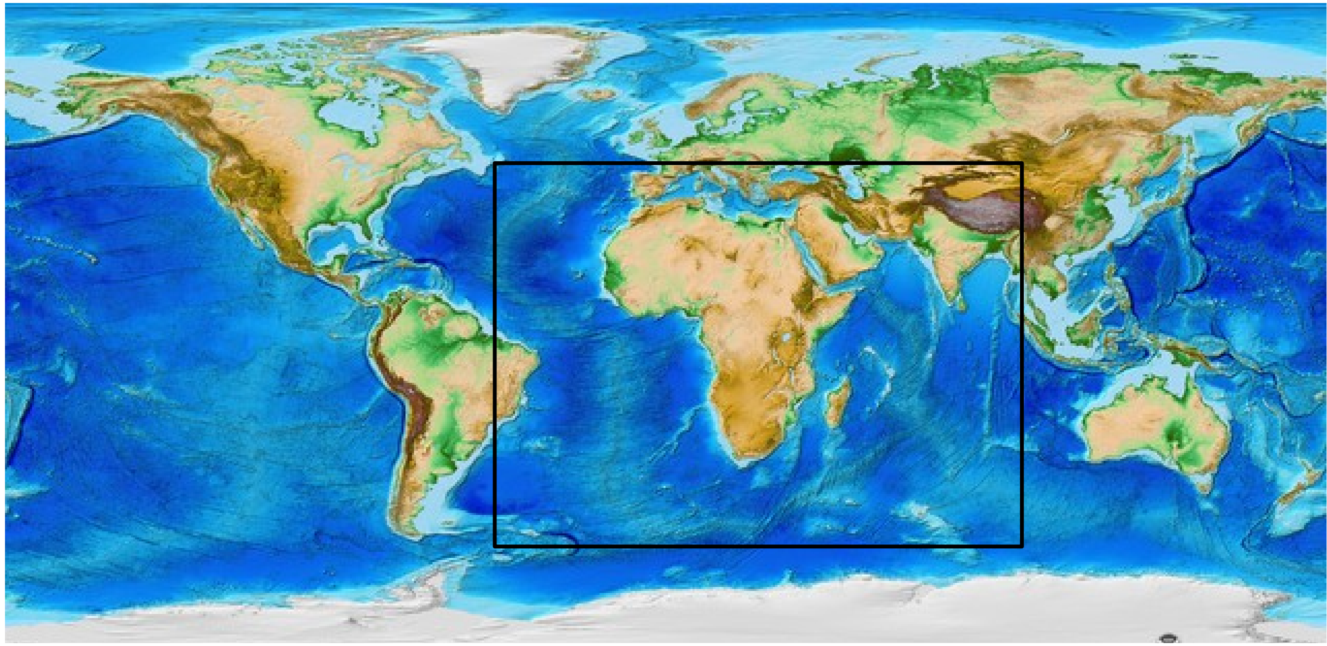
- Binary data cubes, lots of dimensions can be included
- Data registered in bands
- Requires the presence of ncdf4 in R!



Operations on Rasters

Cropping

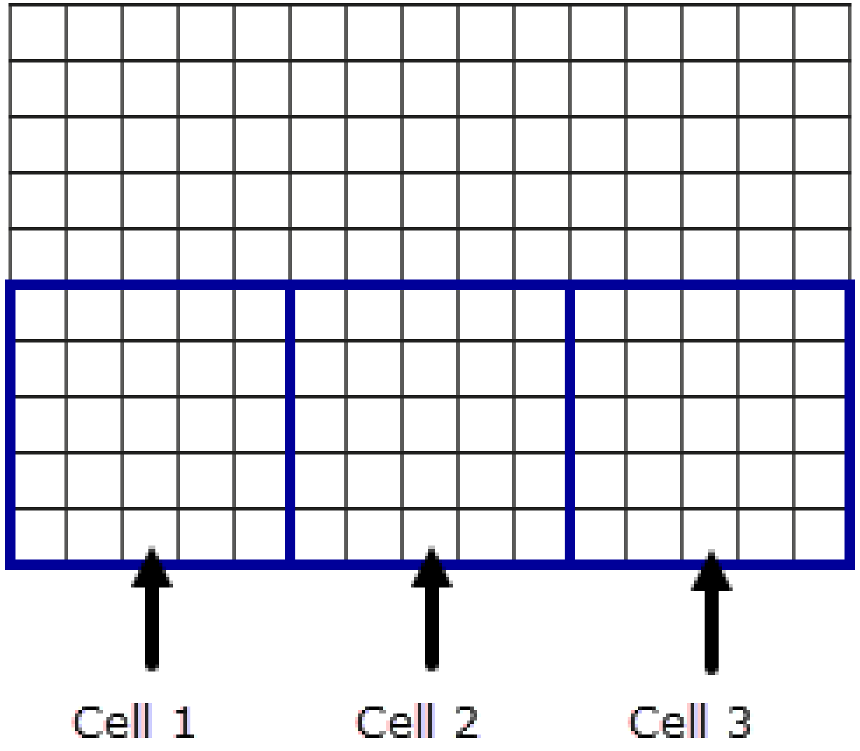
- Changes the extent of a raster



Operations on Rasters

Aggregation / Disaggregation

- Change resolution without changing cell boundaries
- Iteration of a function for data that belongs in a new cell

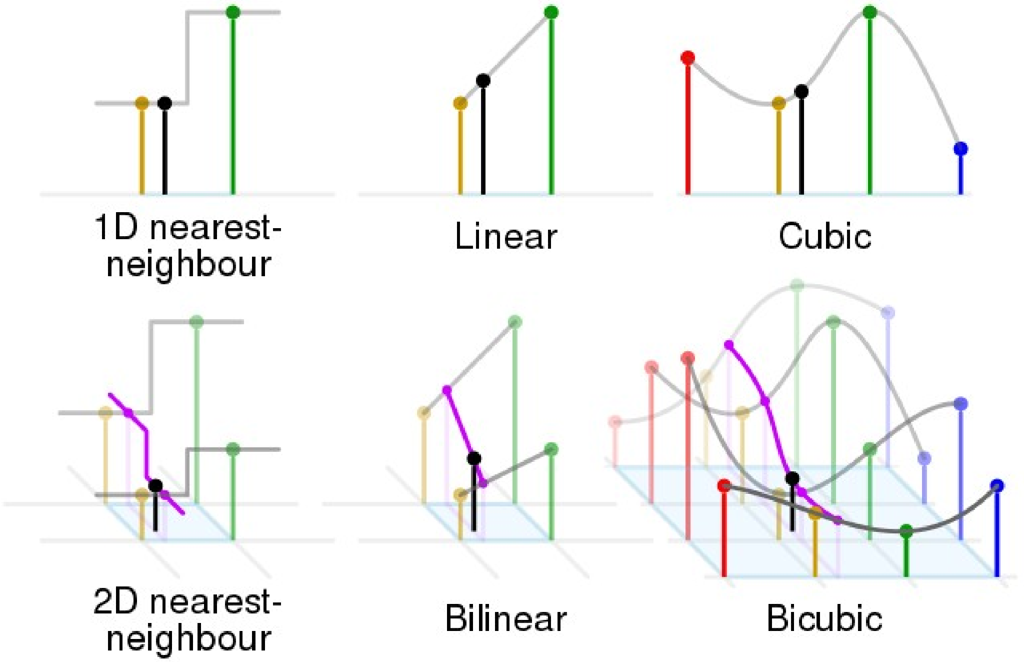


https://desktop.arcgis.com/en/arcmap/latest/tools/spatial-analyst-toolbox/how-aggregate-works.htm

Operations on Rasters

Resampling

- Change resolution to any resolution
- Often relies on interpolation

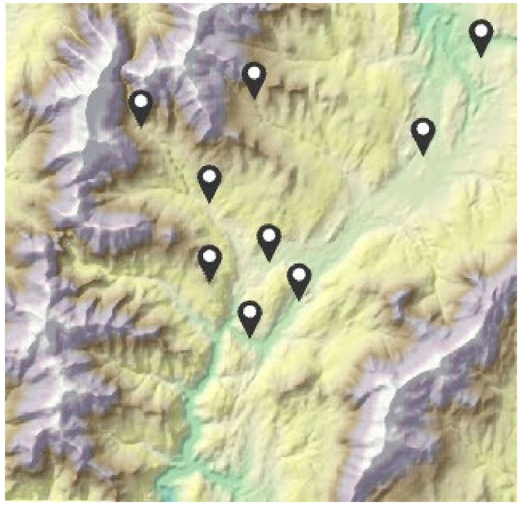


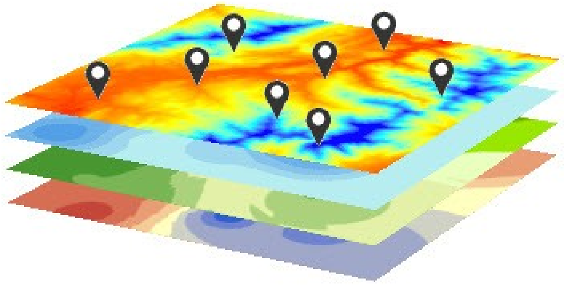
https://www.wikiwand.com/en/Bicubic_interpolation

Operations on Rasters

Extraction

- Getting data out of a raster at given coordinates/locations
- Basic operation for species distribution modelling

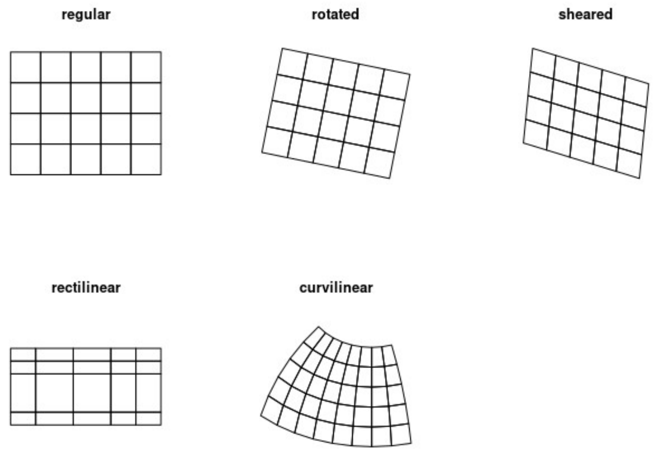




esri.com

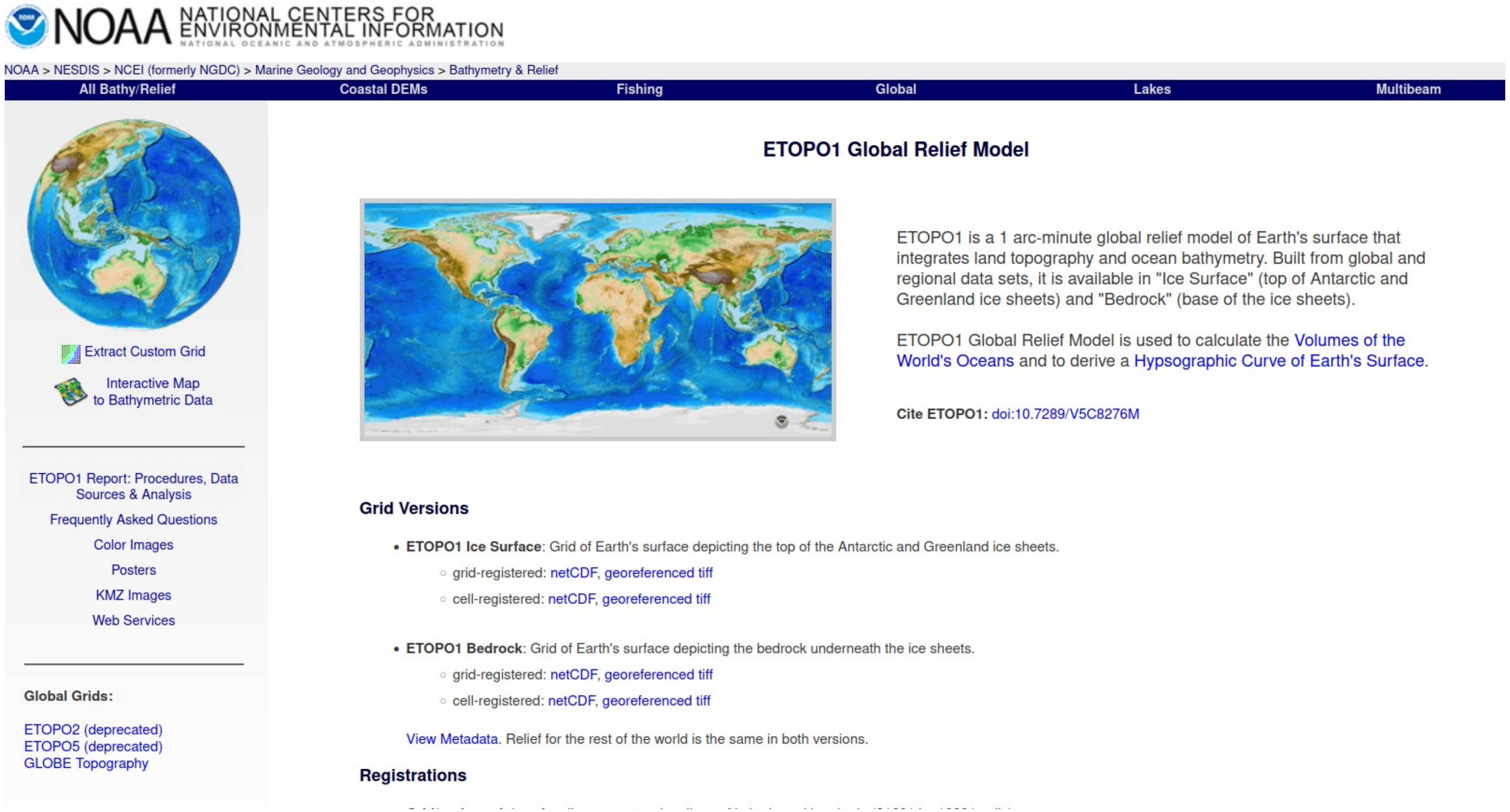
Package stars: Spatiotemporal Arrays, Raster and Vector Data Cubes

- Primarily by Edzer Pebesma
- Somewhat more complex than terra



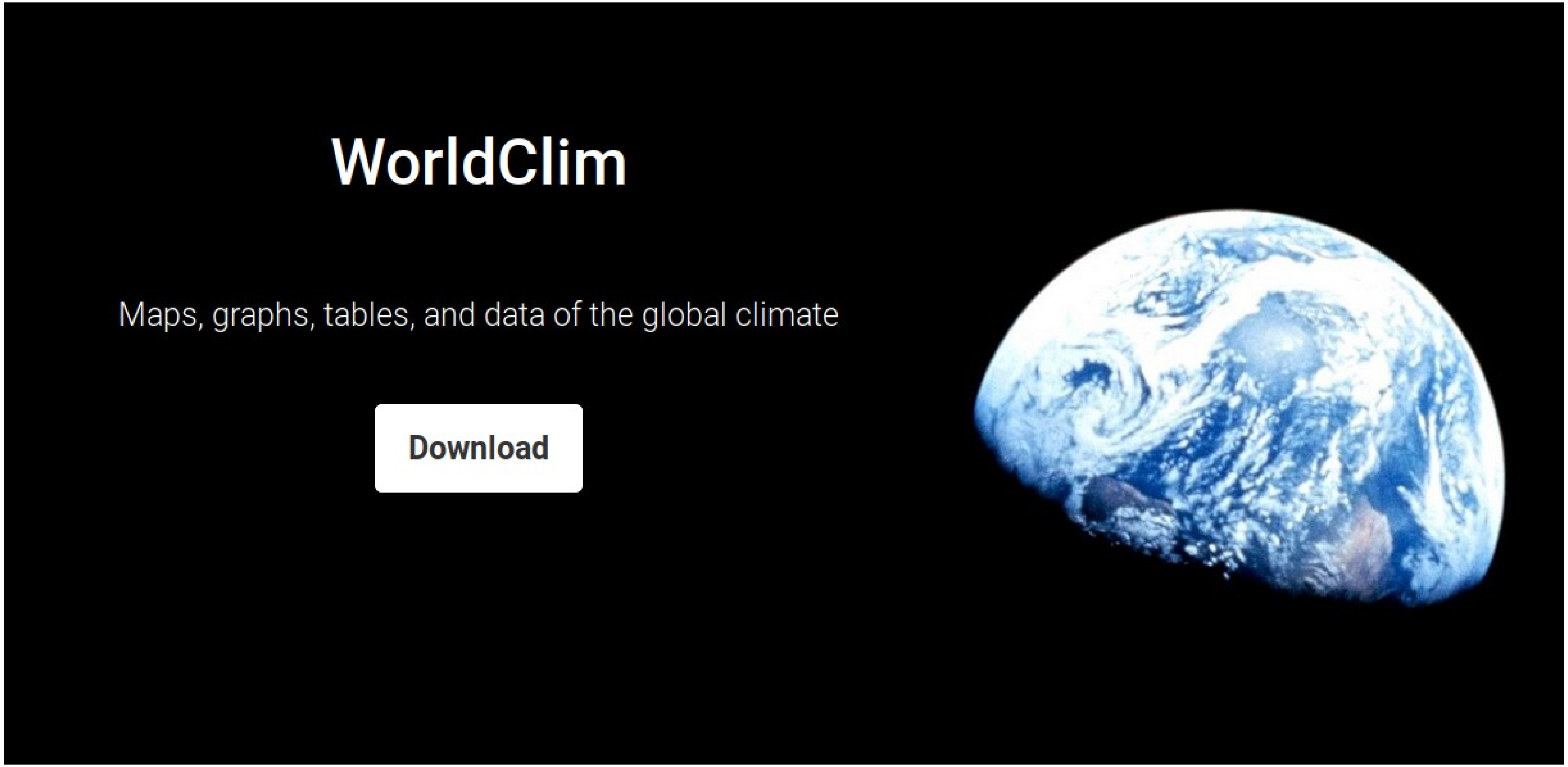
Example 1

- The ETOPO1 Topography (1° by 1°)



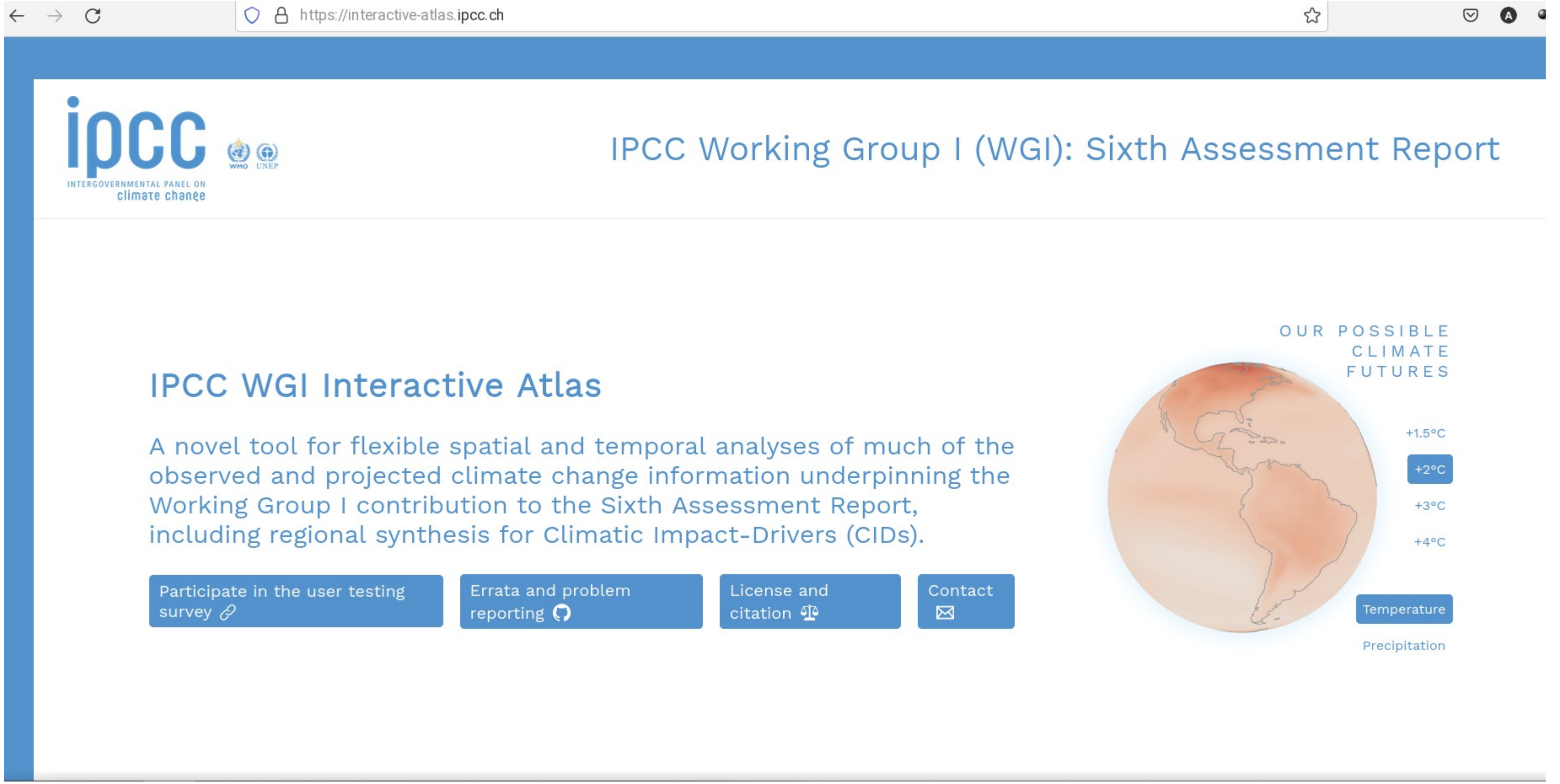
Example 2

- WorldClim



Example +

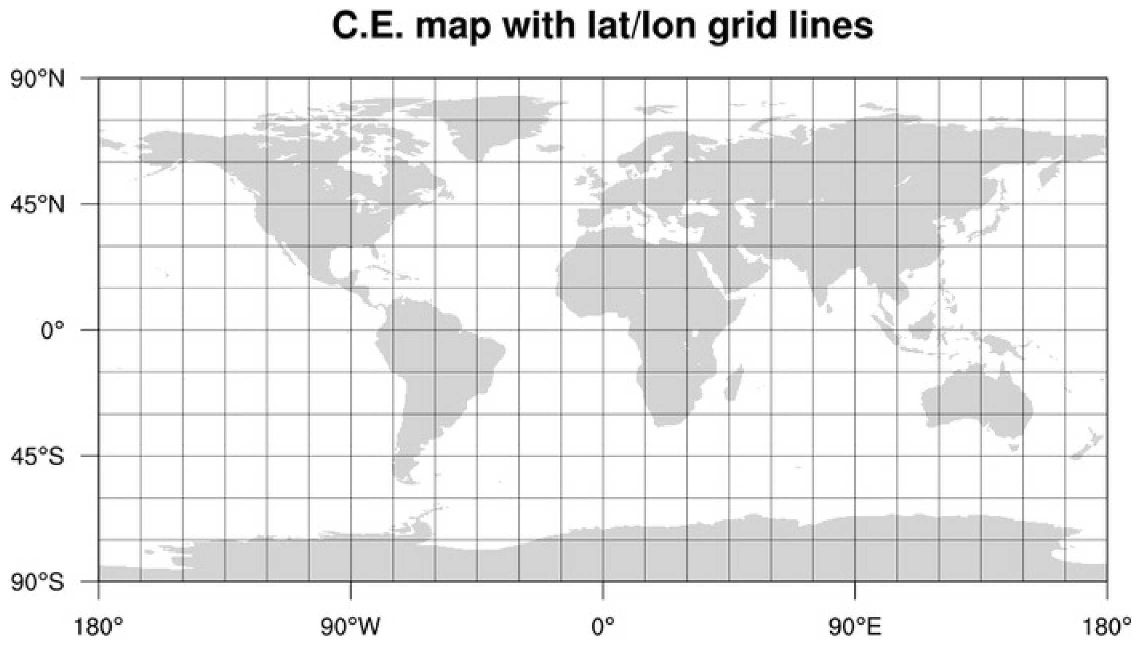
- IPCC AR6 Atlas



Non-rectangular raster grids

The advantage of rectangles

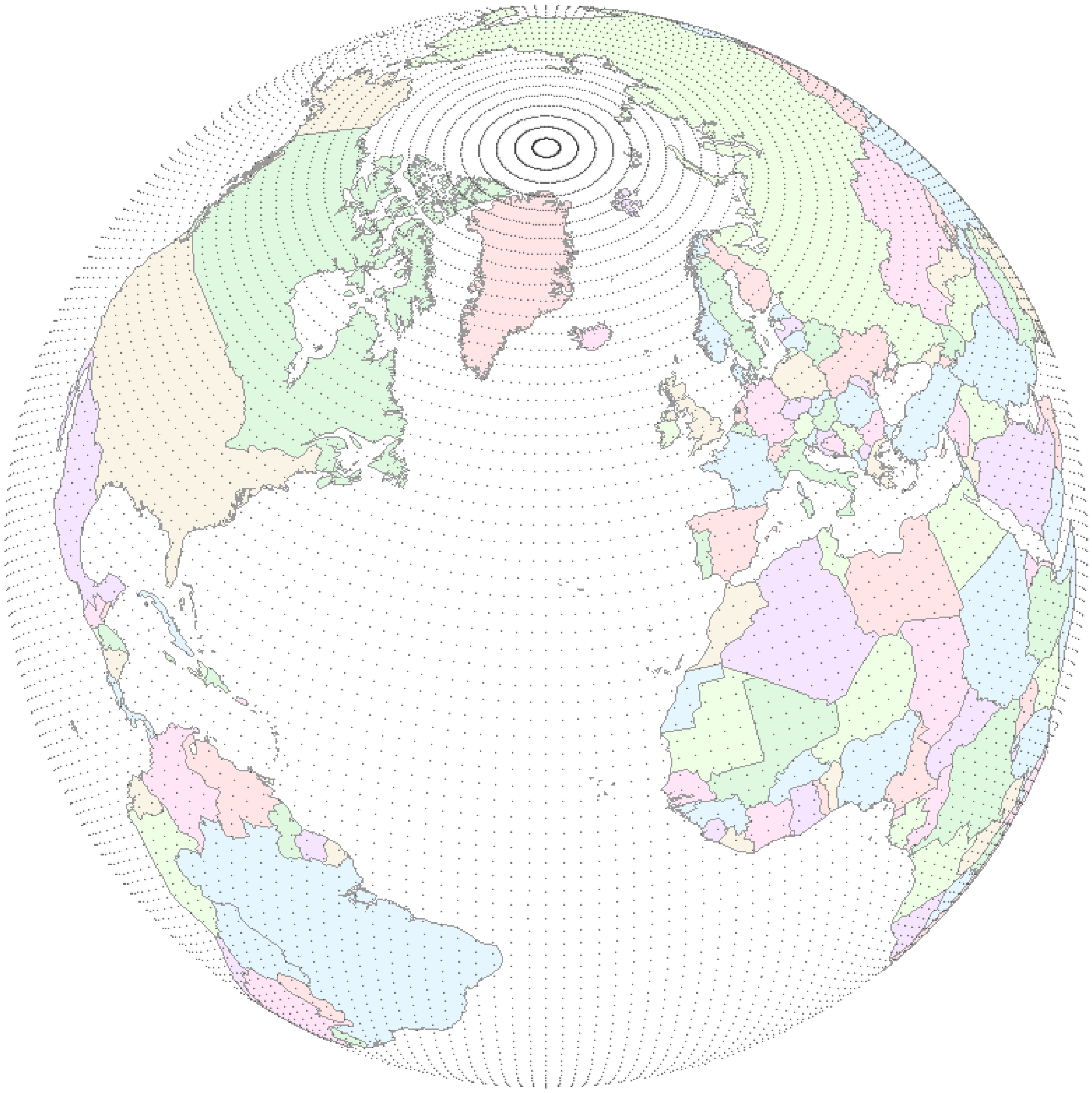
- Structure is easy to define
- Very easy to store in the memory
- Look natural in projections



Non-rectangular raster grids

The disadvantage

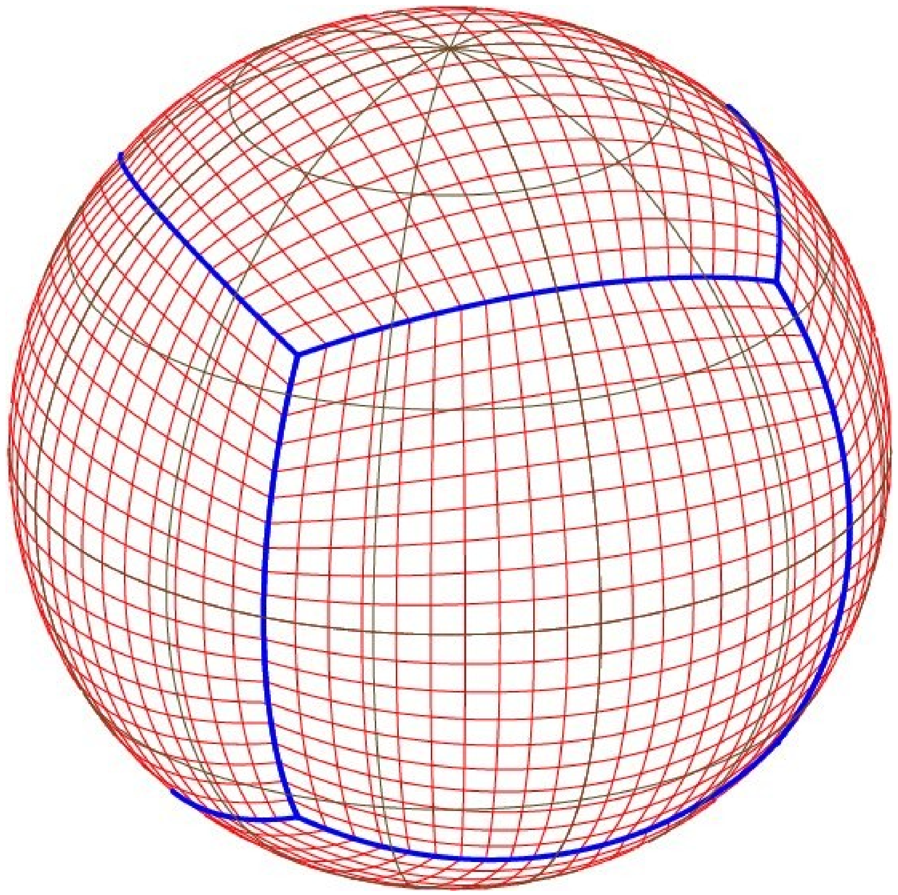
- long-lat grids are Gaussian
- Non-uniform on a sphere
  - Increased density towards the poles
  - Cell sizes decrease towards the poles
- Spatial binning will be biased!



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

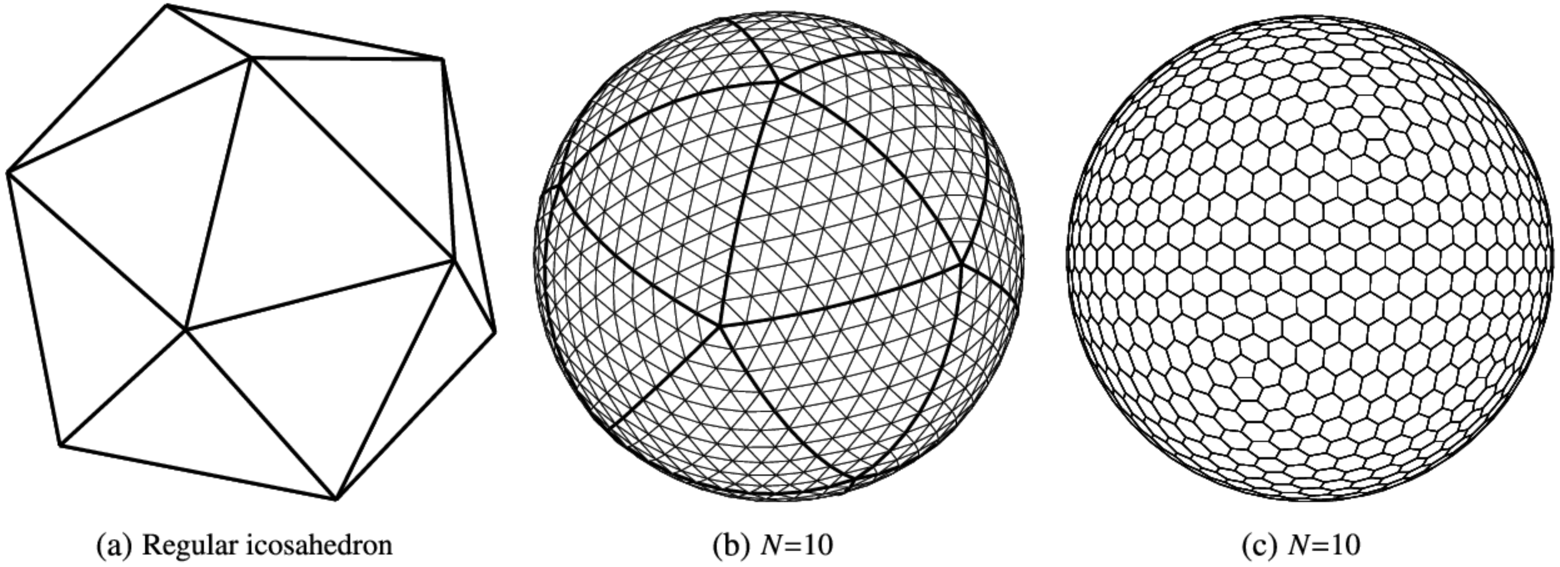
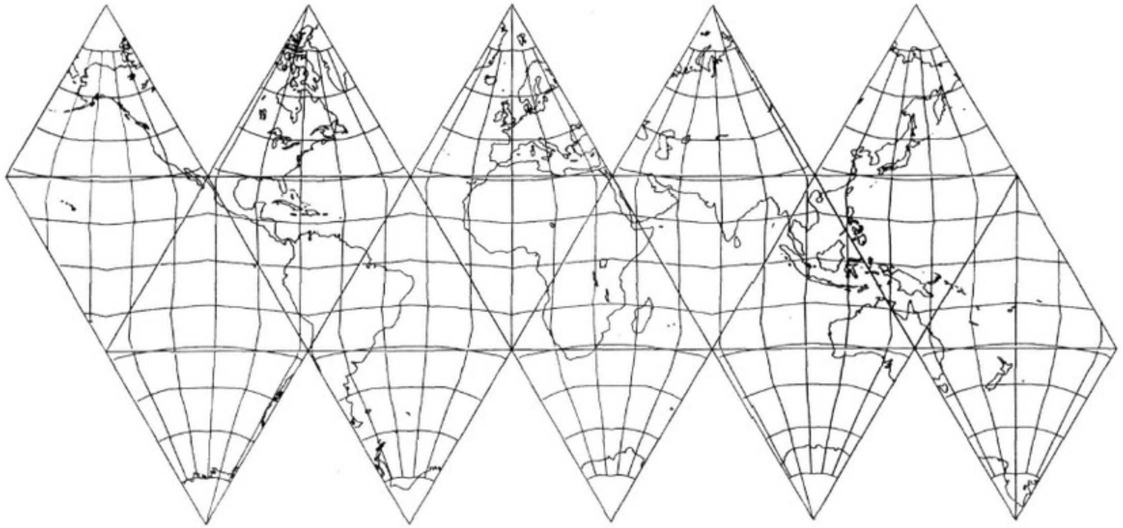


Fig. 7. The gradual refinement of the icosahedral geodesic grids.

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

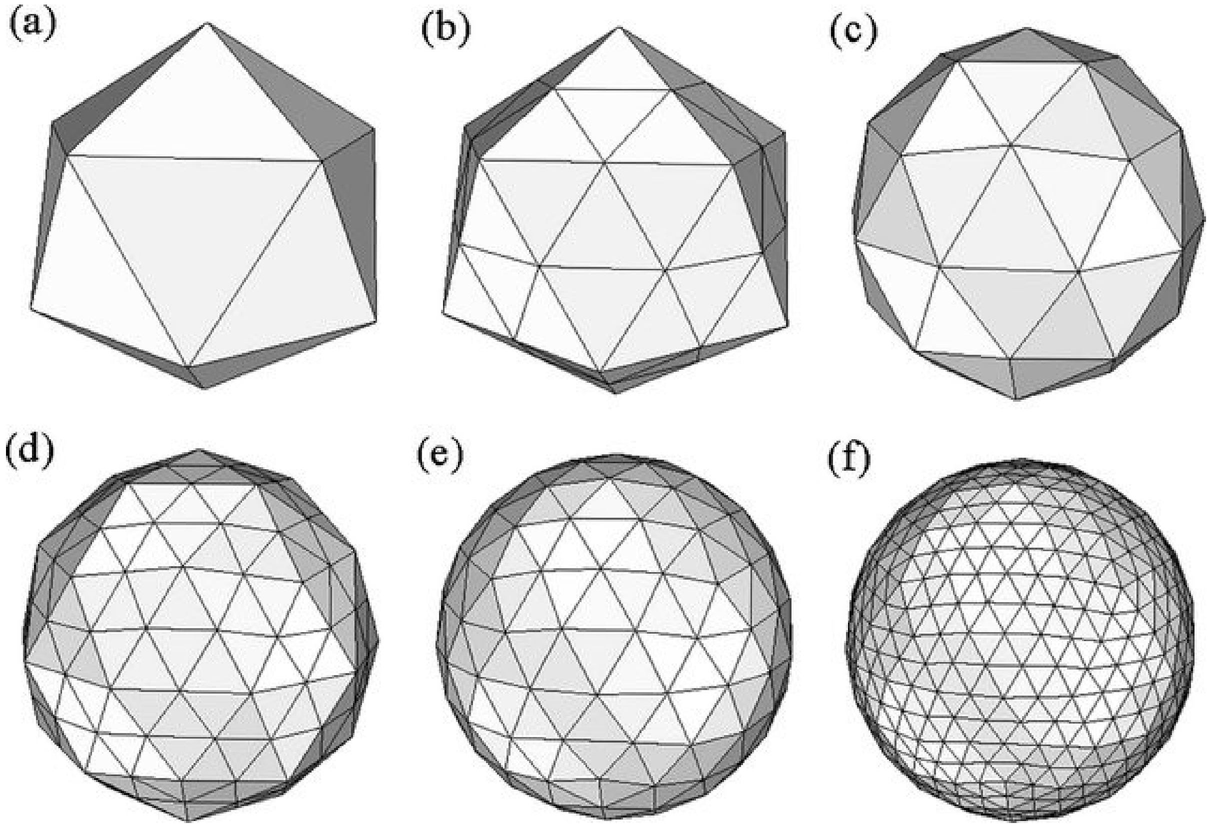


H3: Uber's gridding

- Similar system
- C, Javascript libraries
- In R: h3jsr, h3r
- Similar exponential resolution

Package icosa: Coarse grids based on tessellated icosahedra

- I wrote this, still experimental
- More scalable for global scale
- Interface to sf (will be changed)
- 3D model
- The mathematical properties are unexplored



Icosahedron tessellation (Lipscomb and Ringler, 2005)