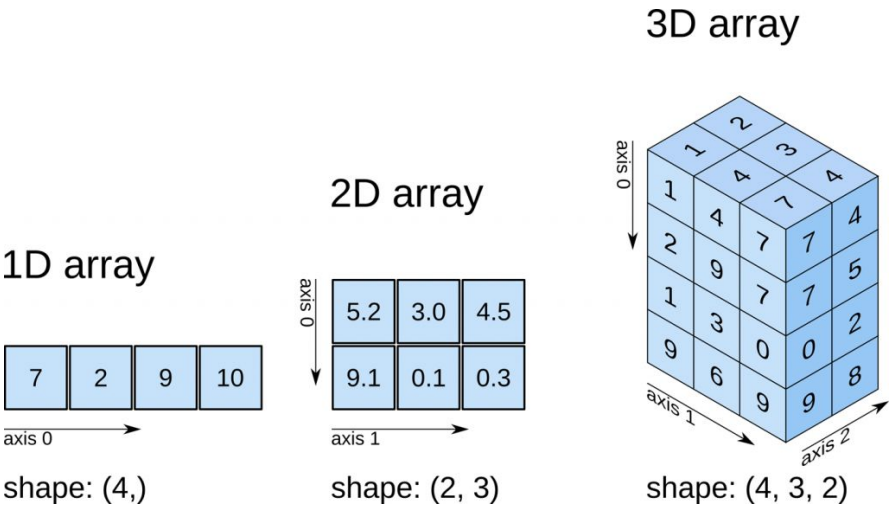
HAS Tools:

Geospatial data analysis: vector data and geopandas

October 7, 2024

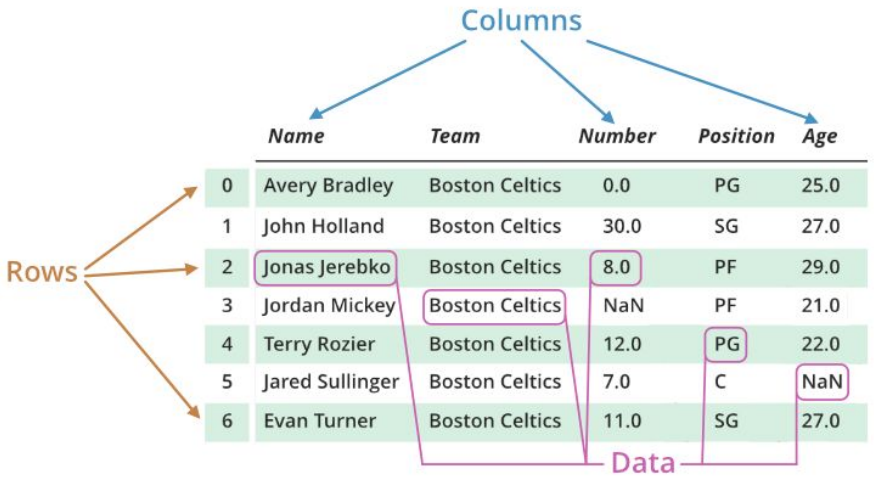
So far we've looked at arrays and data tables as data structures/representations

Array



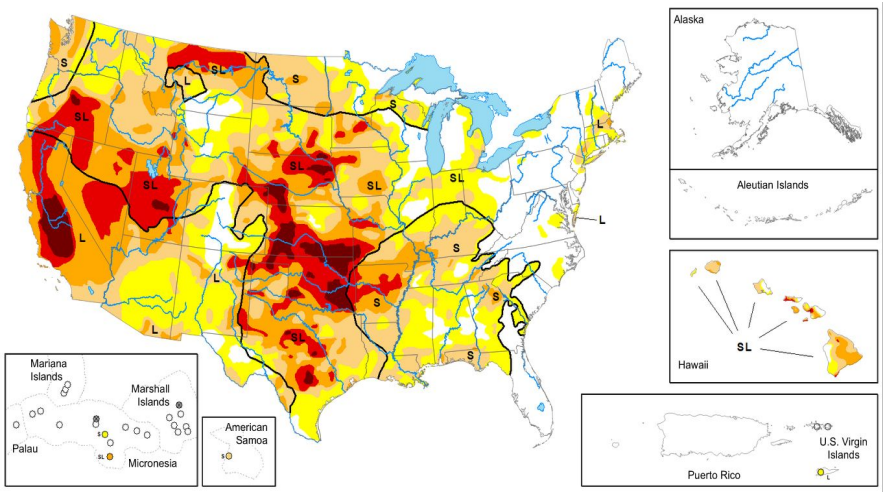
https://predictivehacks.com/tips-about-numpy-arrays/

DataFrame

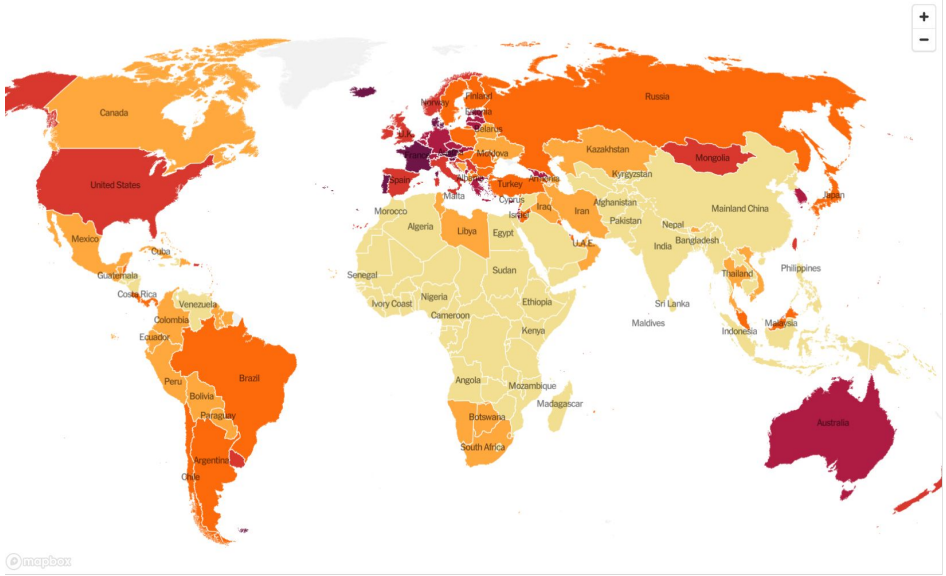


https://www.geeksforgeeks.org/dataframe-operations-in-r/

But map making and geospatial analysis is important in Earth & environmental science



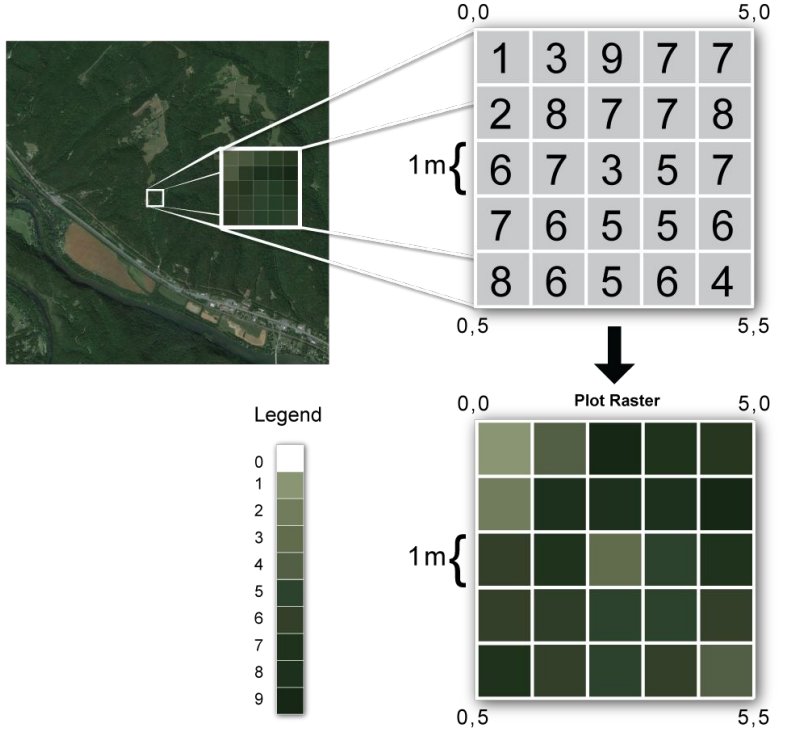
https://droughtmonitor.unl.edu/



https://www.nytimes.com/interactive/2021/world/covid-cases.html

Geographic Information Systems (GIS) have 2 main data representations

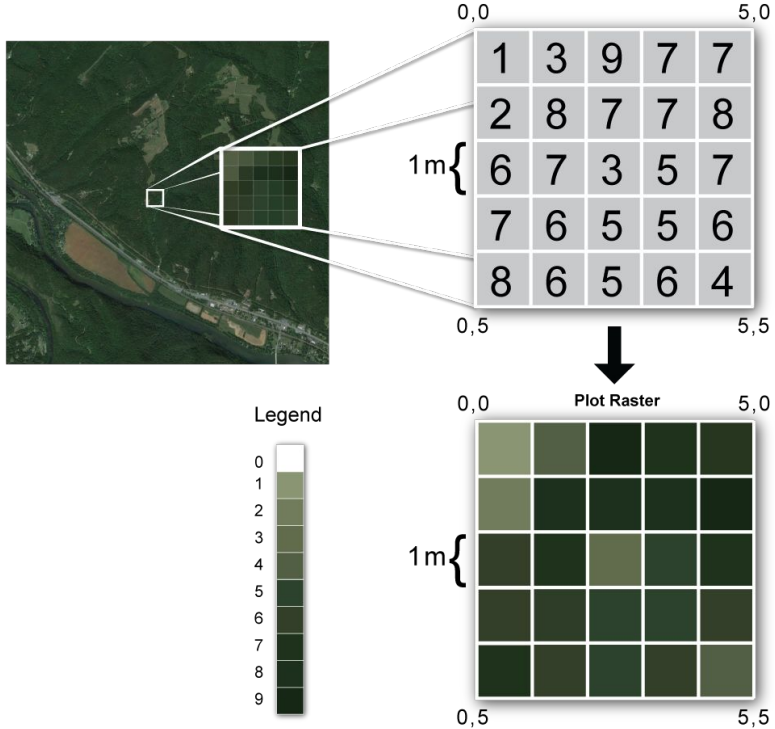
Raster data



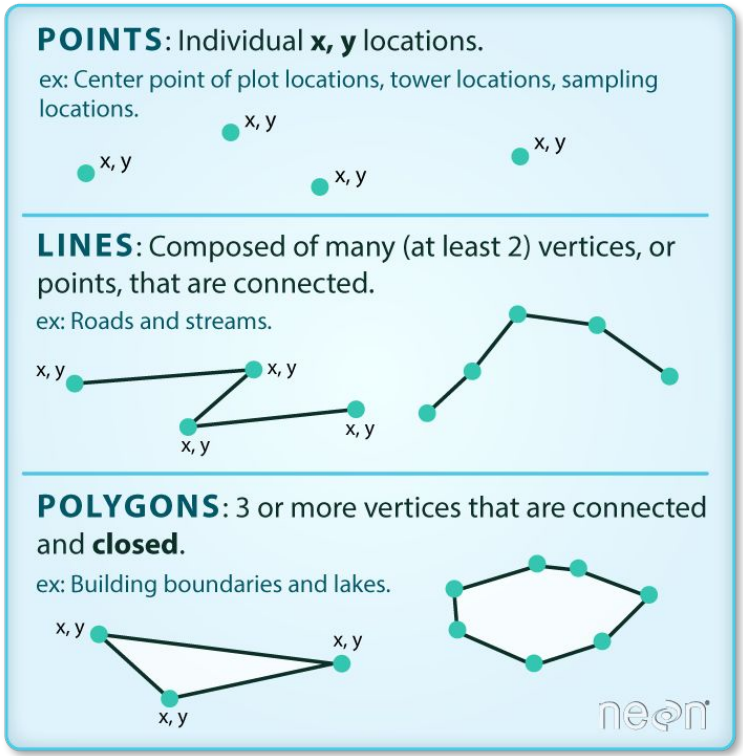
https://datacarpentry.org/organization-geospatial/

Geographic Information Systems (GIS) have 2 main data representations

Raster data

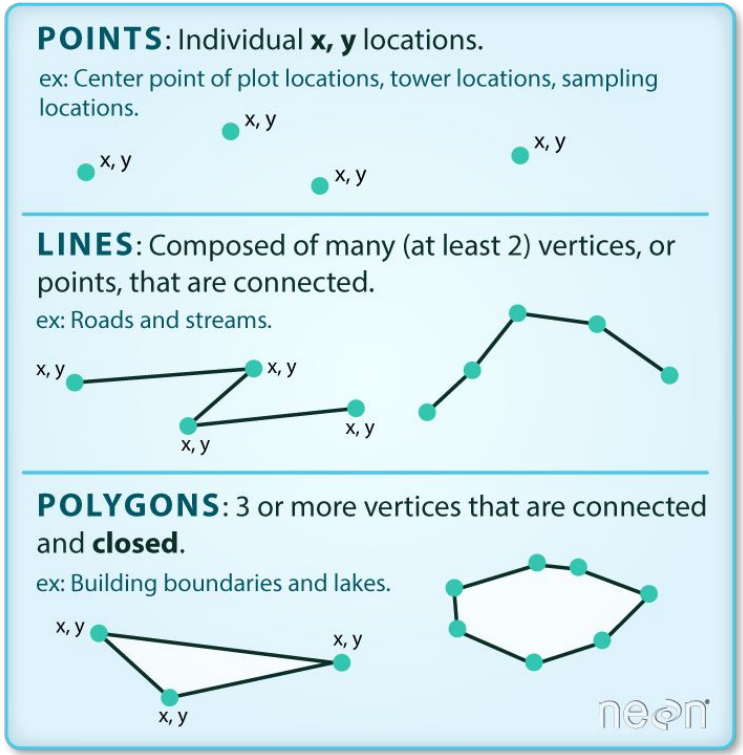


Vector data

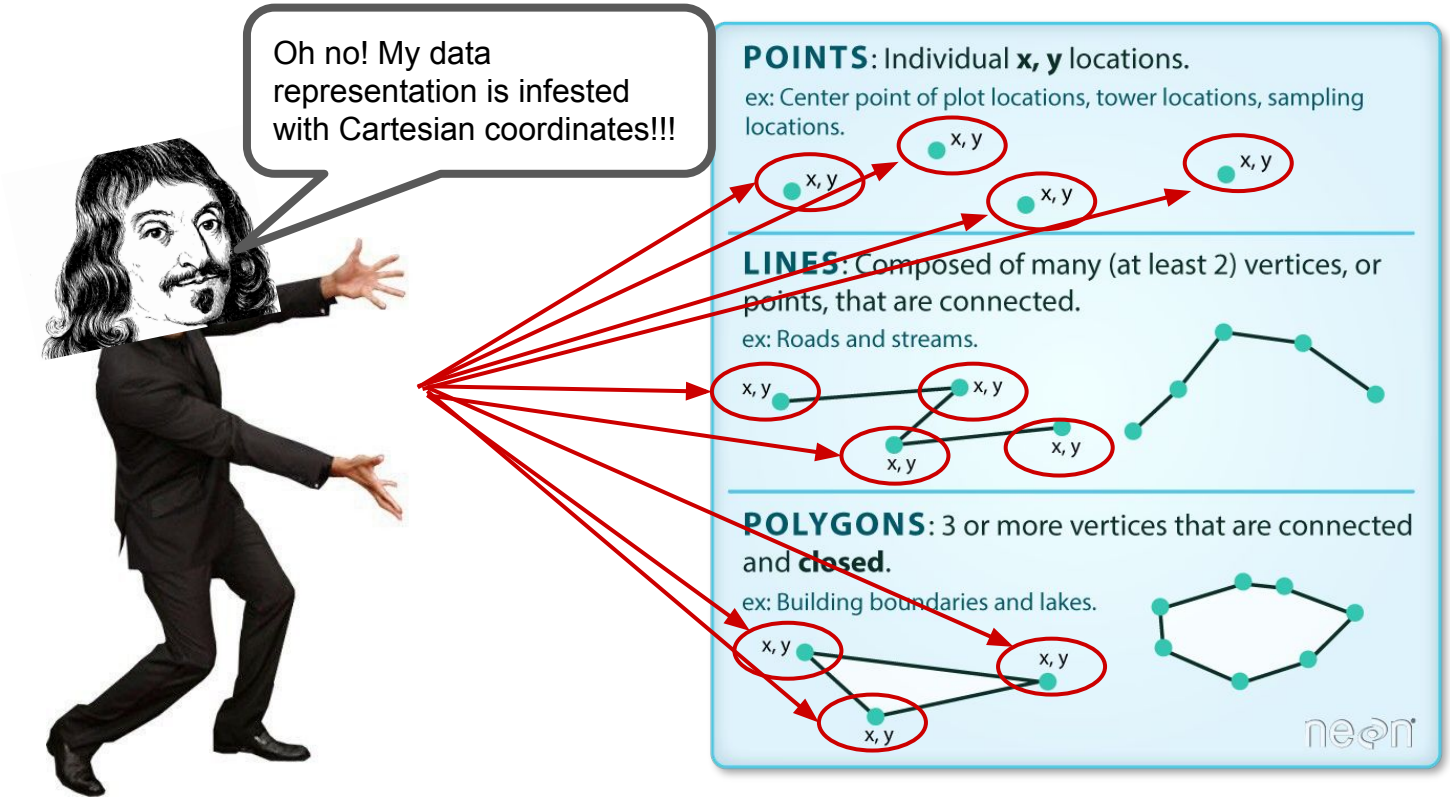


https://datacarpentry.org/organization-geospatial/

Today we begin to investigate vector data



One problem though… Earth isn't flat

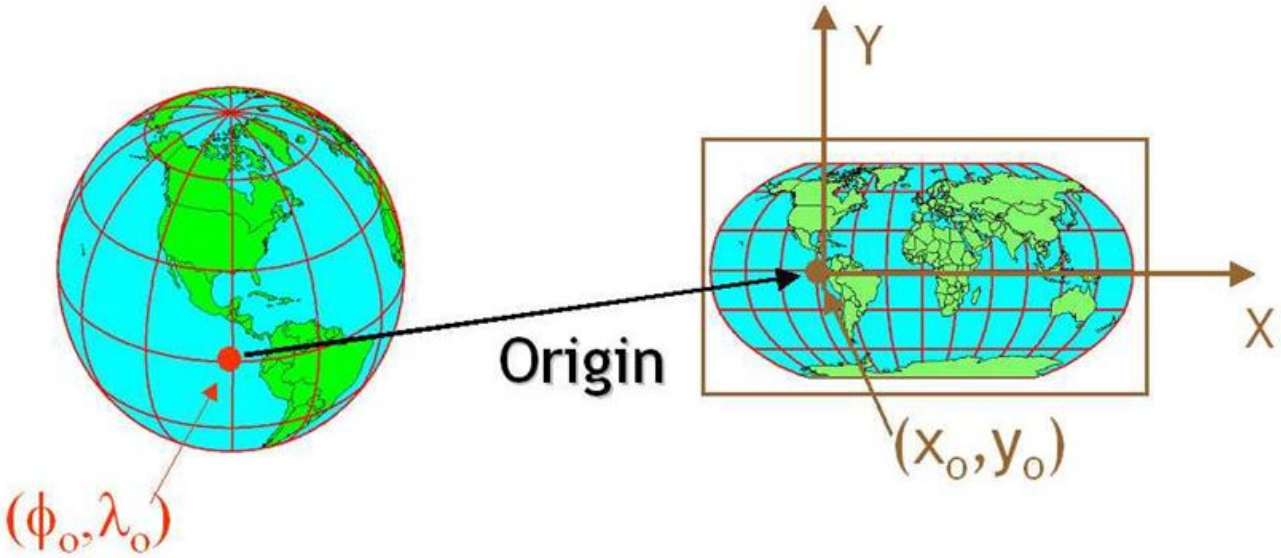



OR IS IT???


OR IS IT???
NO

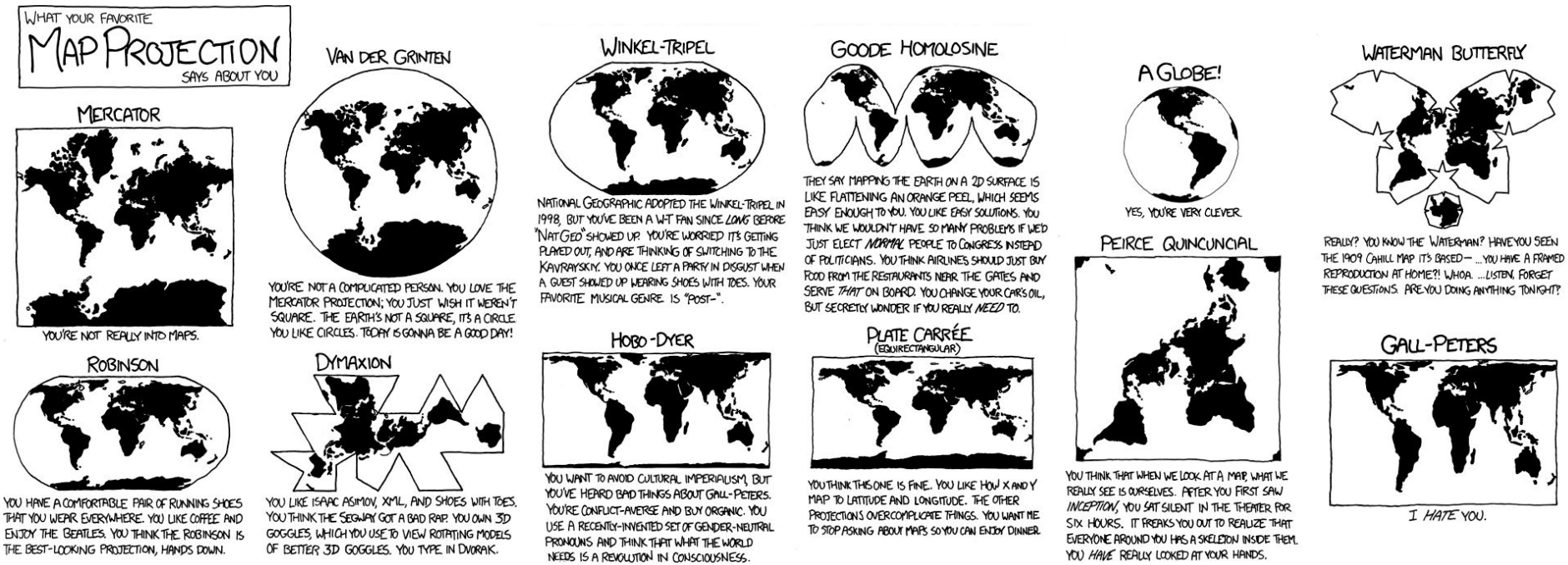
Fun fact! It is mathematically impossible to map a sphere onto a plane with no distortion.

To get around this we can “project” the sphere onto the plane while trying to preserve certain properties…



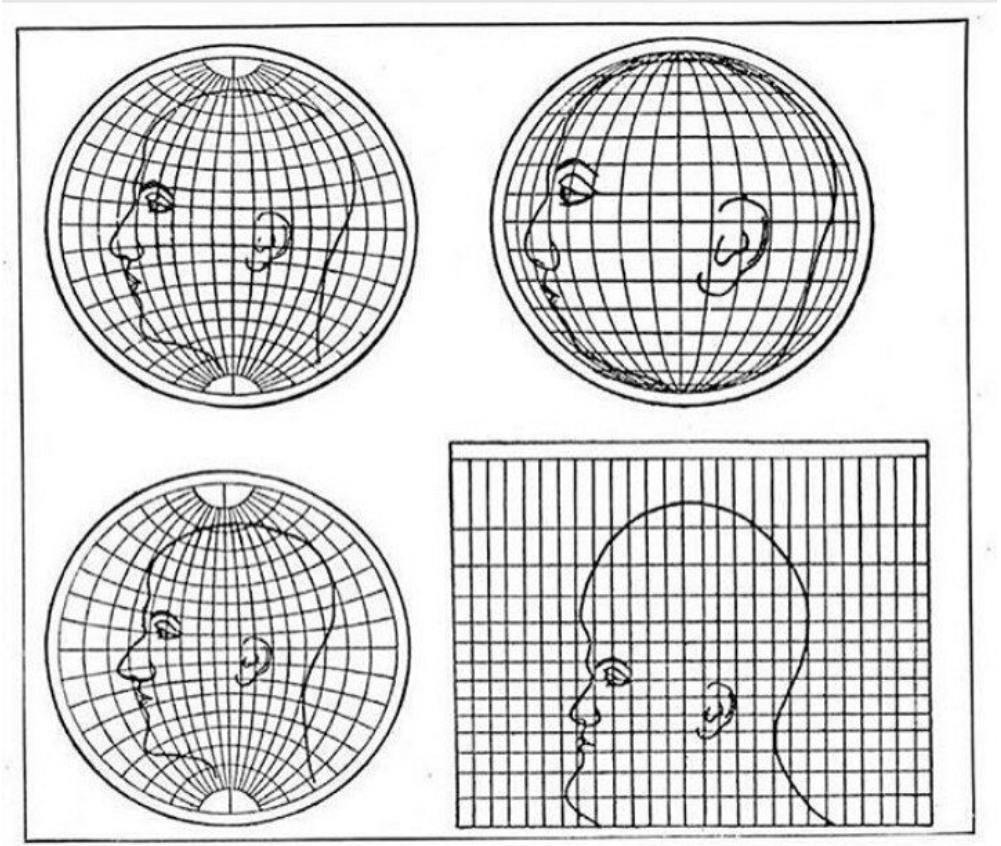
https://www.earthdatascience.org/courses/earth-analytics/spatial-data-r/intro-to-coordinate-reference-systems/

Many ways to do this, and of course, there is an XKCD for it.



https://xkcd.com/977/

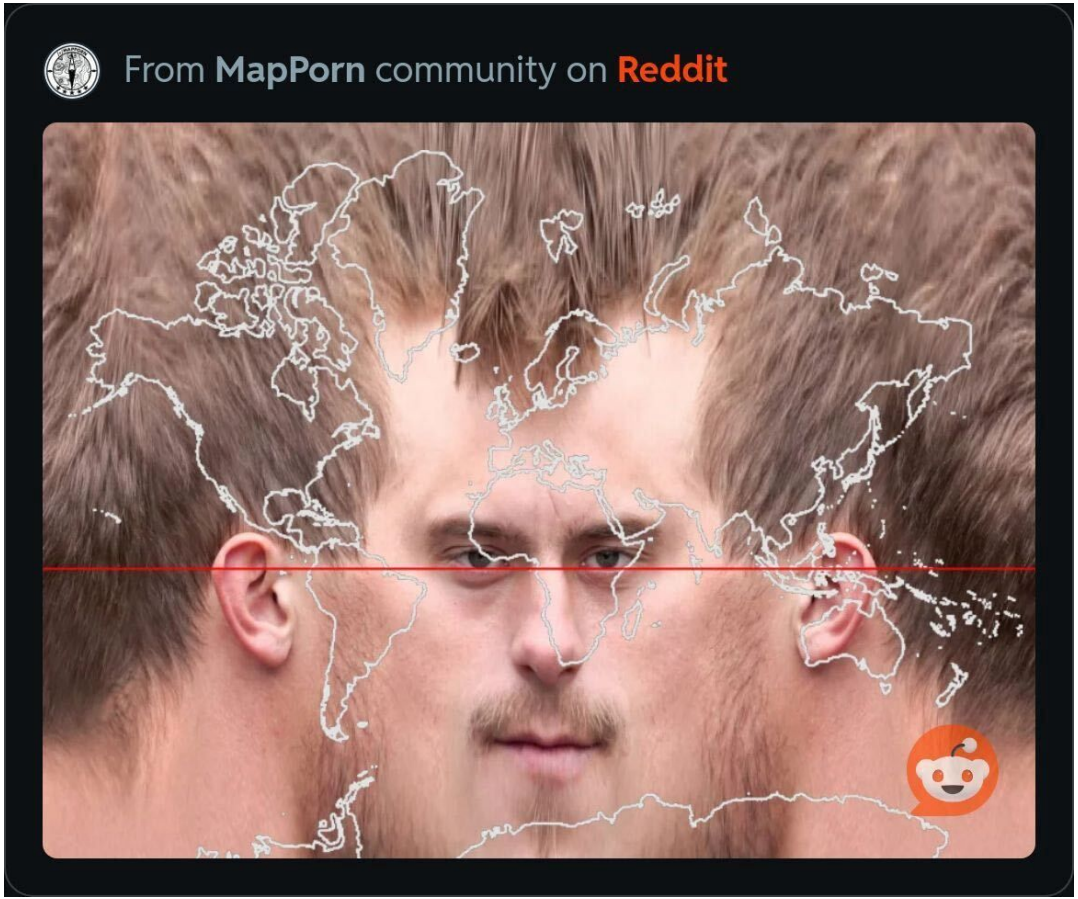
There’s also this gem, although I couldn’t find the original citation, unfortunately.



Upper left: Globular. *Upper right:* Orthographic. *Lower left:* Stereographic. *Lower right:* Mercator

What four commonly used projections do, as shown on a human head

Kudos to Aldo for also providing this effective illustration of… something?



Some things you should be aware of

GDAL / OGR

Geospatial Data Abstraction Library.

- The swiss army knife for geospatial.
- Read and write Raster (GDAL) and Vector (OGR) datasets
- More than 200 (mainly) geospatial formats and protocols.

Slide from "GDAL 2.2 What's new?" by Even Rouault (CC BY-SA)

https://archive.fosdem.org/2018/schedule/event/geopandas/attachments/slides/2487/export/events/attachments/geopandas/slides/2487/slides.pdf

Some things you should be aware of

GEOS

GEOS Geometry Engine Open Source

Geometry Engine Open Source

- C/C++ port of a subset of Java Topology Suite (JTS)
- Most widely used geospatial C++ geometry library
- Implements geometry objects (simple features), spatial predicate functions and spatial operations

Used under the hood by many applications (QGIS, PostGIS, MapServer, GRASS, GeoDjango, ...)

geos.osgeo.org

https://archive.fosdem.org/2018/schedule/event/geopandas/attachments/slides/2487/export/events/attachments/geopandas/slides/2487/slides.pdf

Geospatial packages for python

As always, this space is large and we're only covering a snippet

- Raw bindings for GDAL/OGR: https://pypi.org/project/osgeo/
- Better bindings for GDAL/OGR:
  - Rasterio for GDAL: https://rasterio.readthedocs.io/en/latest/
  - Fiona for OGR: https://fiona.readthedocs.io/en/latest/
- Higher level support for vector/geometric analysis: https://shapely.readthedocs.io/en/stable/manual.html
- High level support for merging data and geometries: https://geopandas.org/en/stable/
- The best way to work with general raster (and other labeled multi-dimensional arrays): https://docs.xarray.dev/en/stable/

GeoPandas

Next time we will explore how this all works with GeoPandas, but for now, let’s play around with projections