ECON6027 2B

COORDINATE REFERENCE SYSTEMS AND PROJECTIONS

Packages you need

- sf
- spData

Executive summary

1. There are two types of CRSs: (i) geographic ('lon/lat', with units in degrees longitude and latitude) and (ii) projected (typically with units of meters).
2. Many geometry operations in sf, assume their inputs have a projected CRS.
3. In R, a CRS can be set either by an EPSG code (4/5 digit number, nickname), a WKT2 string (full name) or proj4string (short name).

```
> st_crs(world)
> st_crs(world)$epsg
> st_crs(world)$proj4string
```

Probably a good idea to remember 4326 (World Geodetic System)

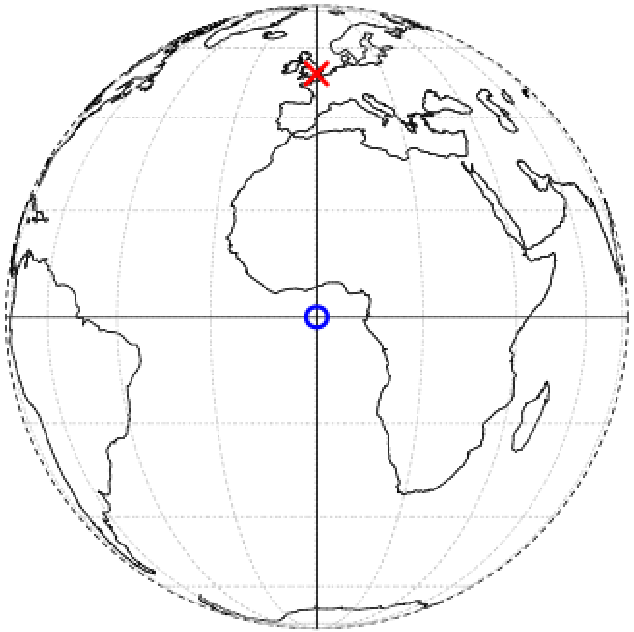
Coordinate Reference System (CRS)

A CRS defines how the spatial elements of the data relate to the surface of the Earth (or other bodies).

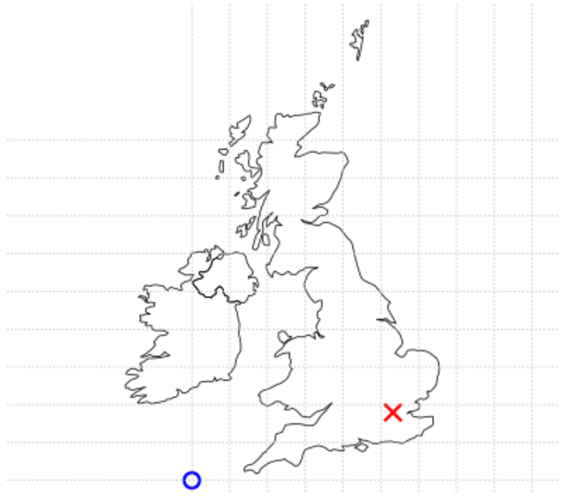
CRSs are either **geographic** or **projected**.

Example: “geographic” location of London

London, can be represented by the coordinates c(-0.1, 51.5). This means that its location is -0.1 degrees west and 51.5 degrees north of the origin. The origin in this case is at 0 degrees longitude (the Prime Meridian) and 0 degree latitude (the Equator) in a geographic (‘lon/lat’) CRS.



Example: “projected” location of London

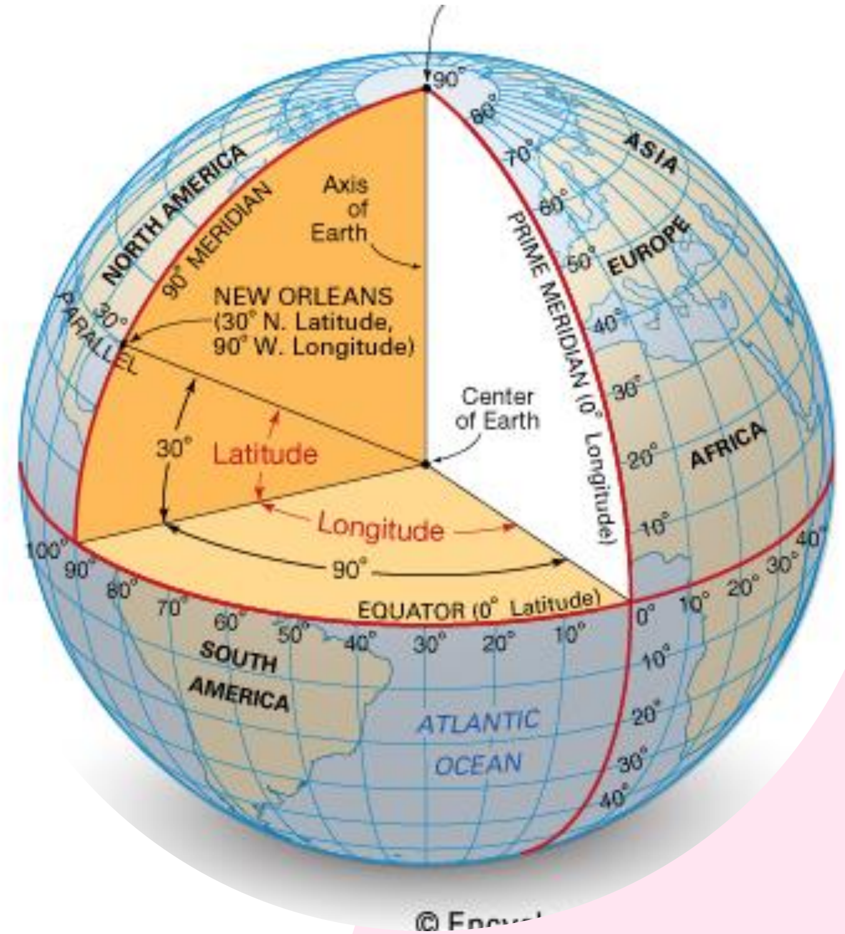


The same point could also be approximated in a **projected CRS** with ‘Easting/Northing’ values of c(530000, 180000) in the British National Grid, meaning that London is located 530 km *East* and 180 km *North* of the origin of the CRS. This can be verified visually: slightly more than 5 ‘boxes’ — square areas bounded by the grey grid lines 100 km in width — separate the point representing London from the origin (Figure right panel). The location of National Grid’s origin, in the sea beyond South West Peninsular, ensures that most locations in the UK have positive Easting and Northing values.

Singapore uses SVY21 (Singapore Transverse Mercator)

Geographic CRS

- Geographic coordinate systems identify any location on the Earth's surface using two values — longitude and latitude.
- *Longitude* is location in the East-West direction in angular distance from the Prime Meridian plane.
- *Latitude* is angular distance North or South of the equatorial plane.
- Distances in geographic CRSs are therefore **not** measured in meters but angle based on the datum (the point of origin, usually the Earth core).



TRY THIS: CHECK THE GEOGRAPHIC COORDINATES OF SINGAPORE

Exercise A

Find the geographic coordinates of Singapore and record this for future reference.

Geographic CRS

- The distance between two lines of longitude, called **meridians**, is around 111 km at the equator.
  - Run geosphere::distGeo(c(0, 0), c(1, 0)) to find the precise distance).
  - This shrinks to zero at the poles.
- Lines of latitude, by contrast, are equidistant from each other irrespective of latitude: they are always around 111 km apart, including at the equator and near the poles

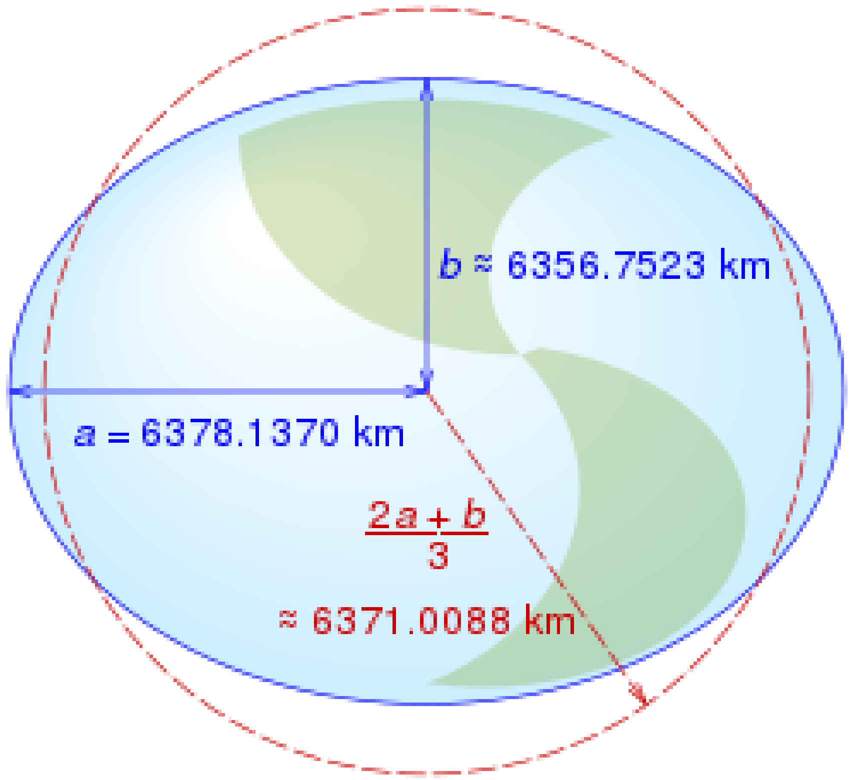
Geographic CRS

The surface of the Earth in geographic coordinate systems is represented by a **spherical** or **ellipsoidal** surface.

Spherical models assume that the Earth is a perfect sphere of a given radius. **Spherical models** have the advantage of **simplicity** but are rarely used because they are **inaccurate**: Earth is not a sphere!

Ellipsoidal models are defined by two parameters: the equatorial radius and the polar radius. These are suitable because the **Earth is compressed**: the equatorial radius is around 11.5 km longer than the polar radius.

WGS84 uses an ellipsoidal model.



Two quick ways to describe CRS in R: **epsg** code or **proj4string** definition

- An epsg code is usually shorter, and therefore easier to remember. The code also refers to only one, well-defined coordinate reference system. (epsg points to exactly one particular CRS.)
  - https://epsg.io
  - http://www.epsg-registry.org/
  - https://spatialreference.org/ref/

- A proj4string string definition allows you more flexibility when it comes to specifying different parameters such as the projection type, datum, unit, and the ellipsoid. This way you can specify many different projections, modify existing ones or even create your own!

An overview of some main parameters in a CRS: ellipsoid/spheroid, datums, towgs84 and projections

- The **ellipsoid/spheroid**: describes the generalized shape of the Earth. All mapping and coordinate systems begin with this description and is described using the ellps.

The **datum**: defines origin and orientation of the coordinate axes (as well the size/shape of Earth)

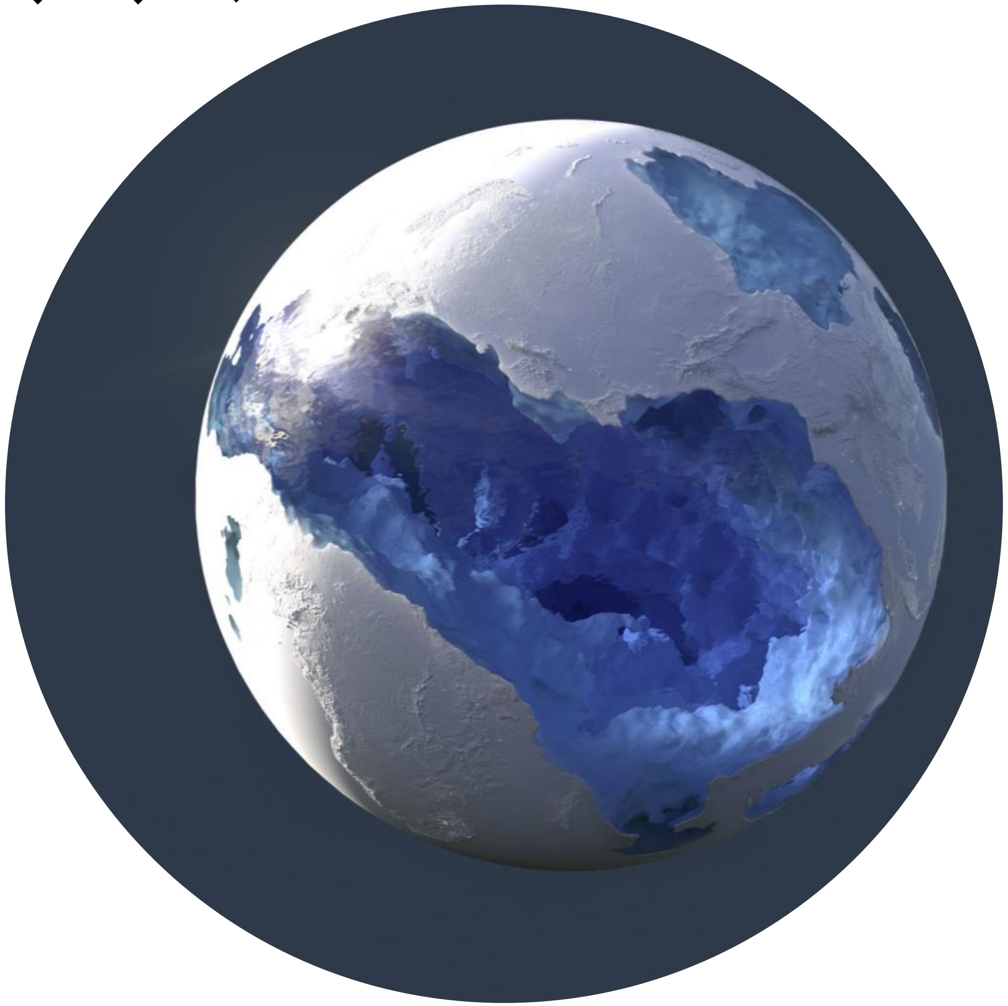
The **projection**: Project the globe onto a 2D surface using a predefined mathematical derivation

- A Globe: A 3D ellipse with Lat/Long coordinates.
- The vector of parameters used to describe the transformation from datum to WGS84 can be described using the **towgs84** parameter.

- A Map: A 2D representation of the 3D Earth with Easting/Northing coordinates

There are lots of ways to do each step, resulting in lots of coordinate reference systems.

HTTP://DOCS.OPENGEOSPATIAL.ORG/IS/18-010R7/18-010R7.HTML



Example

Geographic CRS:

- > st_crs(world)$proj4string
- [1] "+proj=longlat +datum=WGS84 +no_defs"
- longlat means it is geographic.
- Datum is the center of the earth.

Projected CRS

- > st_crs(seine)$proj4string
- [1] "+proj=lcc +lat_0=46.5 +lon_0=3 +lat_1=49 +lat_2=44 +x_0=700000 +y_0=6600000 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs"
- Projection: Lambert Conformal Conic
- Point of origin and orientation: +lat_0=46.5 +lon_0=3 +lat_1=49 +lat_2=44 +x_0=700000 +y_0=6600000
- Ellipse: GRS80
- Units: meter

Ellipses (+ellps)

- Determining the shape of the earth is the first step in developing a CRS. An ellipse is a simple model describing the basic shape of the Earth.
- All mapping and coordinate systems are based on this shape.
- The Earth is almost spherical, however there is a tiny bulge at the equator that makes it ~0.33% larger than at the poles.
- The ellipsoid is an approximation and does not fit the Earth perfectly. **There are different ellipsoids** in use, some are designed to **fit the whole Earth** (WGS84, GRS80) and some are designed to **fit a local region** (NAD27).
- Local ellipses can be more accurate for the area they were designed for, however, are not useful in other parts of the world.

Spheroids/Ellipsoids

- **+ellps** parameter contains information on what ellipsoid to use and the precise **relationship** between the Cartesian **coordinates** and **location** on the Earth's surface.
- Additional details are stored in the towgs84 argument.
- These allow local variations in Earth's surface, for example due to large mountain ranges, to be accounted for in a local CRS.
- In the WKT2 convention ellipsoid and spheroid are used interchangeably.
- Available ellipsoid definitions can be seen by executing:

```
> sf_proj_info(type = "ellps")
```

Datum

Provides the information needed to **anchor** the abstract coordinates to the Earth.

The datum defines an **origin** point of the coordinate axes and defines the **direction of the axes**.

- For example, longitudes running through true north vs. magnetic north, etc

The datum always specifies the **ellipsoid** that is used, but the ellipsoid does not specify the datum! Datums are based on specific ellipsoids and sometimes have the same name as the ellipsoid.

There are **two types** of datum — local and geocentric.

- In a local datum such as NAD83 the ellipsoidal surface is shifted to **align with the surface at a given location**.
- In a geocentric datum such as WGS84 the **centre is the Earth's centre of gravity (core)** and the accuracy of projections is not optimized for a specific location.

Projected CRS

- Projected CRSs are based on Cartesian coordinates on an implicitly flat surface (from the curvy surface of Earth), i.e., the elliptical Earth is projected onto a flat surface (i.e., a paper map)
- They have an **origin**, x and y axes, and a **linear unit of measurement** such as meters.
- All **projected CRSs are based on a geographic CRS** and rely on map projections (a mathematical formula) to convert the three-dimensional surface of the Earth into Easting and Northing (x and y) values in a projected CRS.
- In other words, map coordinates of a point are computed from its ellipsoidal latitude and longitude by a standard formula known as a map projection
- Available projection definitions can be seen by executing:

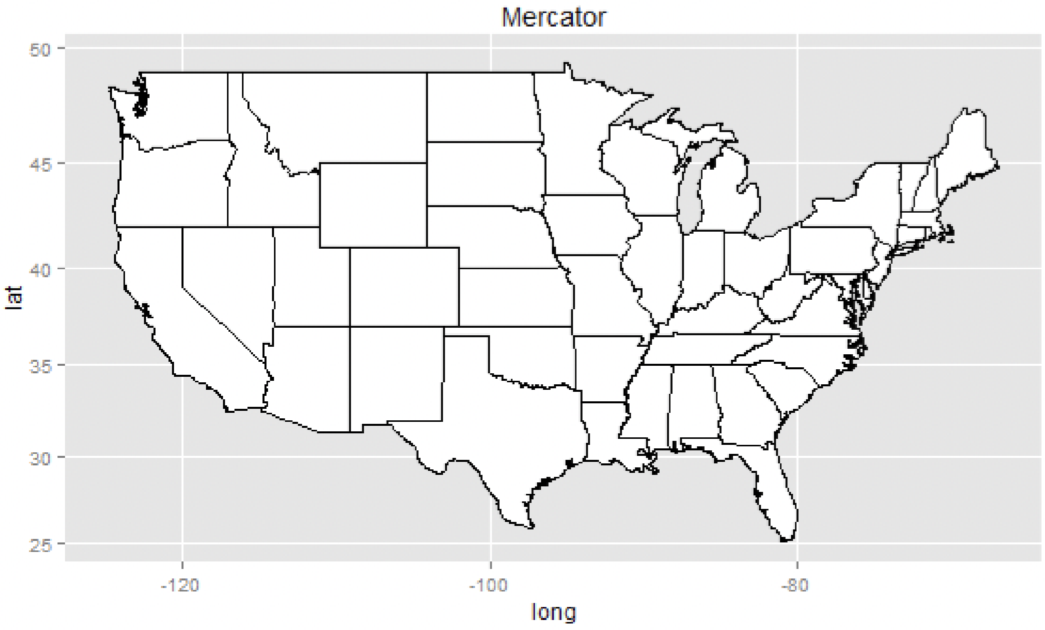
```
> sf_proj_info(type = "proj")
```

Projected CRS: Distortions

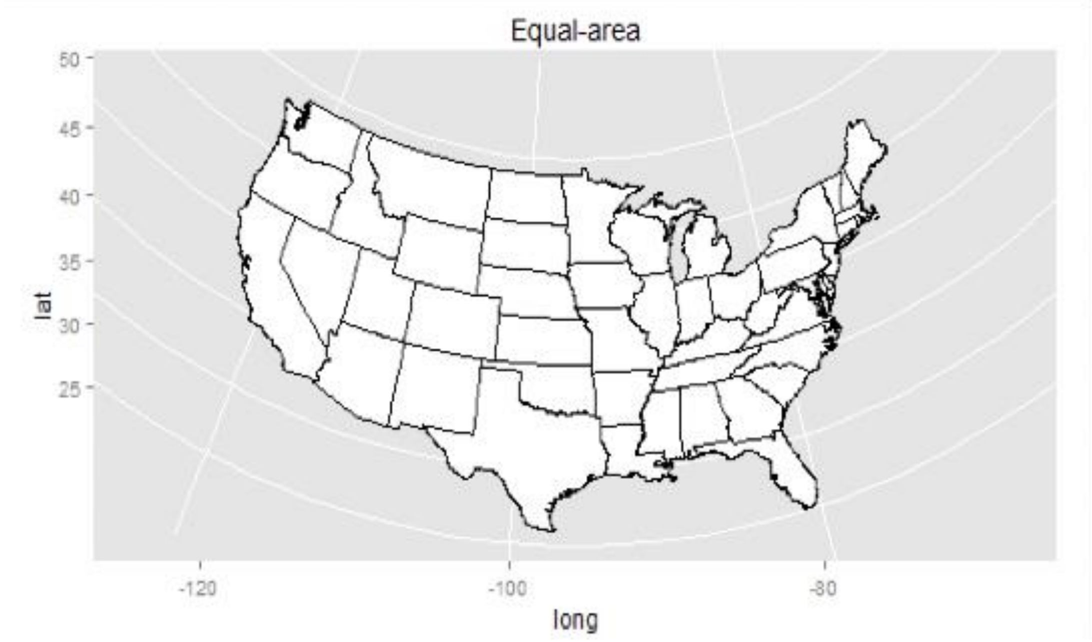
- It is impossible to flatten a round object without distortion, and this results in trade-offs between **(i) area, (ii) direction, (iii) shape, and (iv) distance**.
  - For example, there is a trade-off between distance and direction because both features cannot be simultaneously preserved.
- There is no "best" projection, but some projections are better suited to different applications.
- A projected coordinate system can preserve only one or two of those properties. Projections are often named based on a property they preserve - equal-area preserves area, azimuthal preserve direction, equidistant preserve distance, and conformal preserve local shape.

Distortions

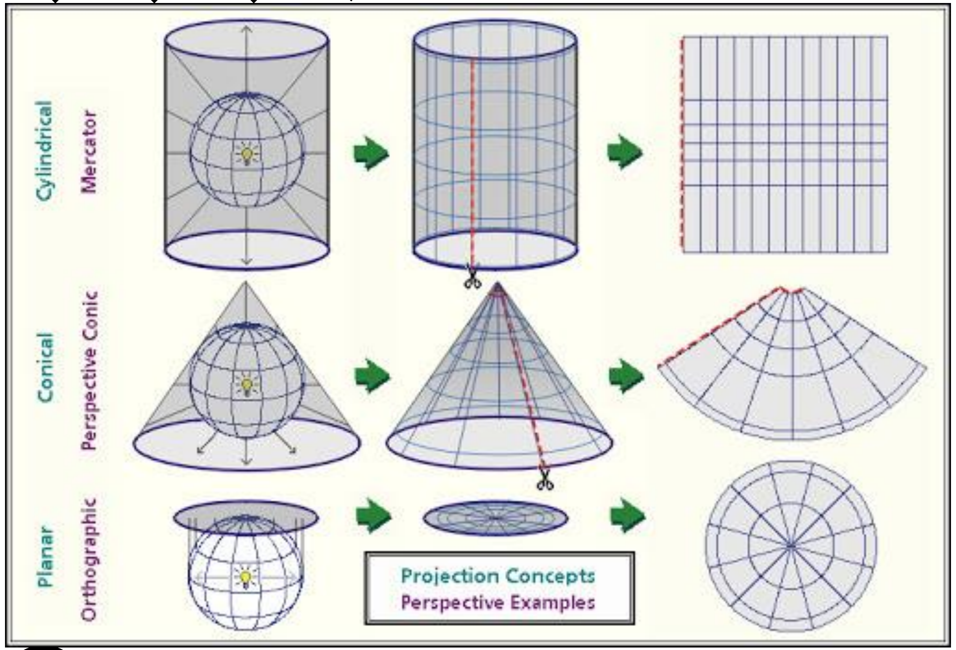
Mercator preserves direction and is useful for navigation. However, distances and areas are distorted, *especially near the polar regions*



Azimuthal Equal Area preserves area, but not direction.



Three main groups of projection types: conic, cylindrical, and planar

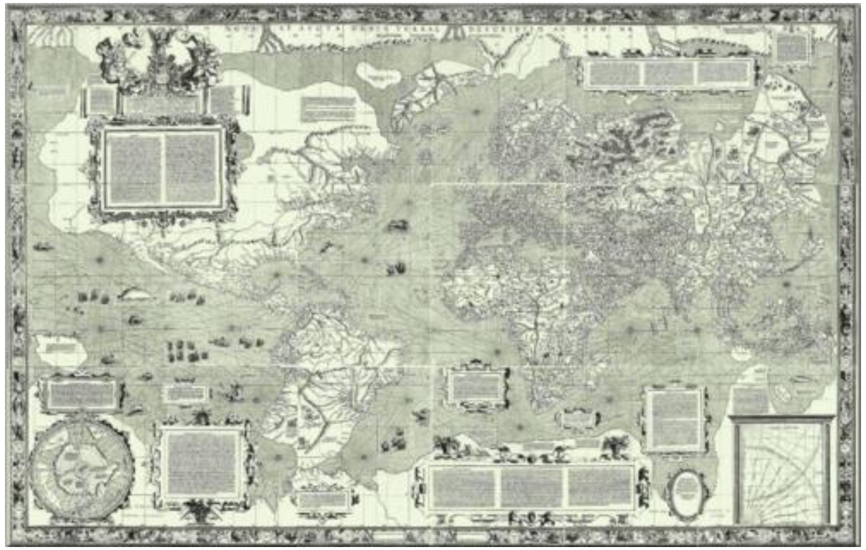


- A **cylindrical/Mercator projection** maps the surface onto a cylinder. This projection could also be created by touching the Earth's surface along a single line of tangency or two lines of tangency. Cylindrical projections are used most often when mapping the entire world. Mercator projection is extensively used in web mapping platforms including Google maps. If we flip the cylinder 90 degrees, we get the **transverse Mercator**.
- In a **conic projection**, the Earth's surface is projected onto a cone along a single line of tangency or two lines of tangency. Distortions are minimized along the tangency lines and rise with the distance from those lines in this projection. Therefore, it is the best suited for maps of mid-latitude areas.
- A **planar projection** projects data onto a flat surface touching the globe at a point or along a line of tangency. It is typically used in mapping polar regions.

HTTP://WWW.RADICALCARTOGRAPHY.NET/?PROJECTIONREF

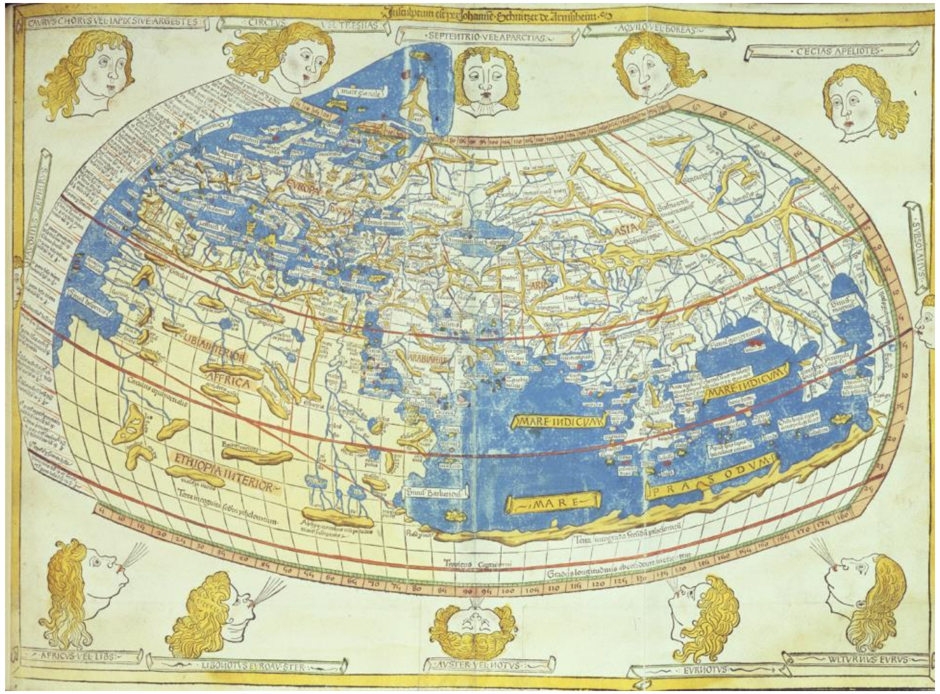
Mercator 1569 World Map (The Medieval Google Map!)

- Mercator titled the map *Nova et Aucta Orbis Terrae Descriptio ad Usum Navigantium Emendata*: "A new and augmented description of Earth corrected for the use of sailors".
- However, it was much ahead of its time, since the old navigational and surveying techniques were not compatible with its use in navigation.
- Only in the middle of the 18th century, after the marine chronometer was invented could the Mercator projection be fully adopted by navigators.
- Criticisms for the Mercator projection meant that its use declined over the 19th and 20th centuries until the advent of Web mapping which has revived its use in the present day. Refer: https://en.wikipedia.org/wiki/Web_Mercator_projection



INTERESTING FACT

PTOLEMY’S WORLD MAP IS THE EARLIEST KNOWN WORLD MAP THAT USES CONIC PROJECTION.



TIME TO GET TO WORK…

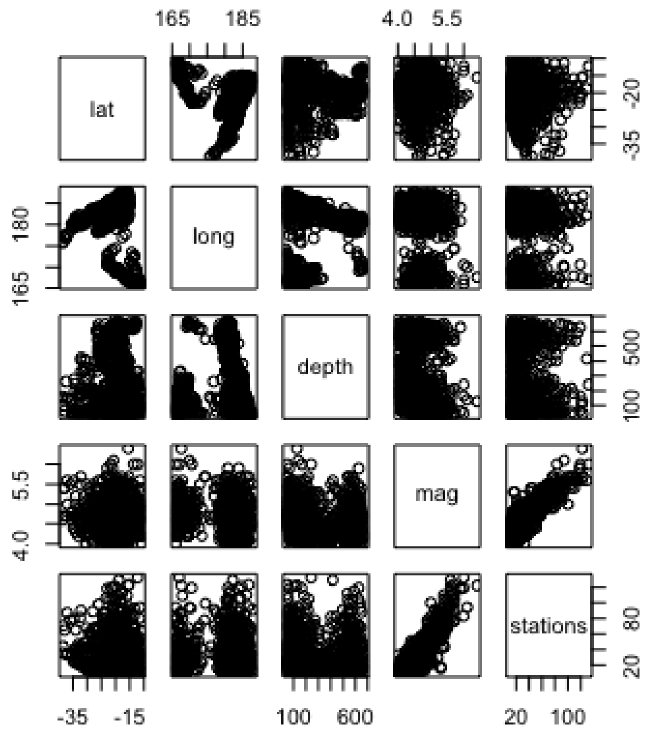
Locations of earthquakes off Fiji

- The data set give the locations of 1000 seismic events of MB > 4.0. The events occurred in a cube near Fiji since 1964.
- This is a basic data frame that treats the positional data in “lat” and “lon” as another set of attributes.
- Let us give this dataset “spatial awareness”

```
> fiji = datasets::quakes
> head(fiji)
     lat   long depth mag stations
1 -20.42 181.62   562 4.8       41
2 -20.62 181.03   650 4.2       15
3 -26.00 184.10    42 5.4       43
4 -17.97 181.66   626 4.1       19
5 -20.42 181.96   649 4.0       11
6 -19.68 184.31   195 4.0       12
> class(fiji)
[1] "data.frame"
```

Plot

```
> plot(fiji)
```



Give it spatial powers…

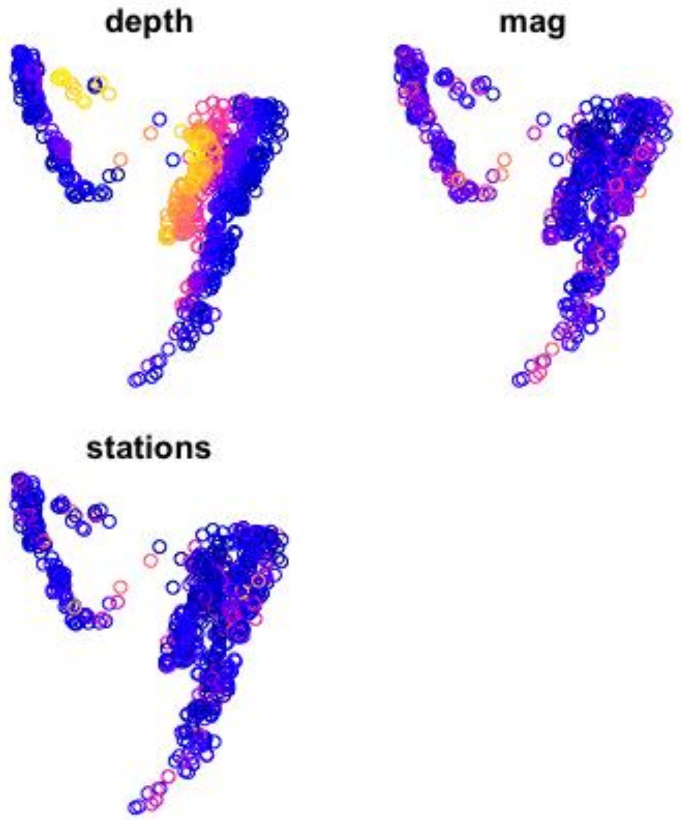
```
> summary(fiji.sf)
     depth            mag           stations           geometry
 Min.   : 40.0   Min.   :4.00   Min.   : 10.00   POINT  :1000
 1st Qu.: 99.0   1st Qu.:4.30   1st Qu.: 18.00   epsg:NA:    0
 Median :247.0   Median :4.60   Median : 27.00
 Mean   :311.4   Mean   :4.62   Mean   : 33.42
 3rd Qu.:543.0   3rd Qu.:4.90   3rd Qu.: 42.00
 Max.   :680.0   Max.   :6.40   Max.   :132.00
```

```
> fiji.sf = st_as_sf(fiji, coords=c(”long",”lat"))
> class(fiji.sf)
[1] "sf" "data.frame"
> summary(region.sf)
```

We have an sf object that is aware of location…

Plot

```
> plot(fiji.sf)
```



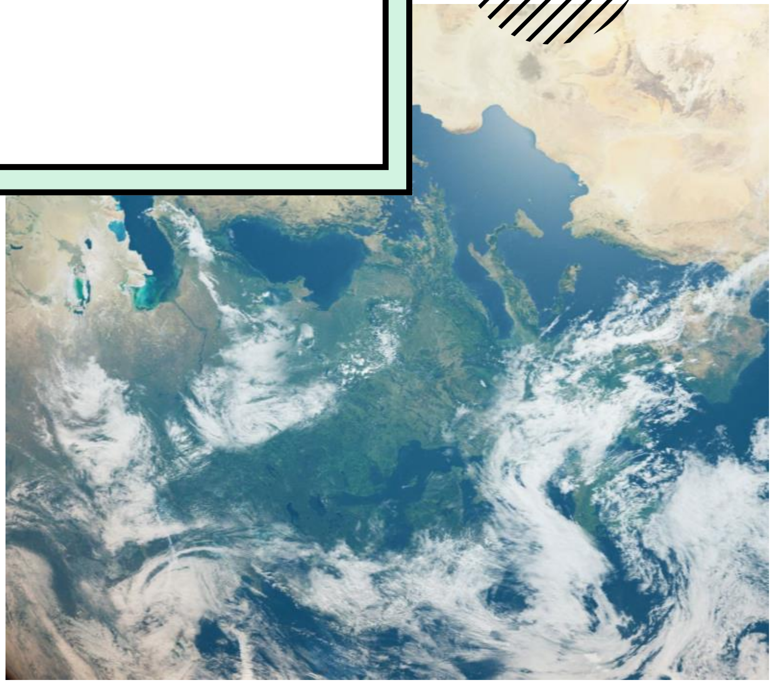
What about CRS?

```
> st_is_longlat(fiji.sf)
[1] NA


> st_crs(fiji.sf)
Coordinate Reference System:
NA
```

- However, no CRS.
- That means we know the numbers refer to "locations", but we have no idea where on "Earth" exactly these locations are.
- In general, geographic datasets without a CRS can be of limited use unless we are conducting an isolated study of some sort.
- It is ok not to know a CRS too. Only earth locations can have a CRS, but spatial data analysis is much larger. A "space" can be defined in many non-geographic forms including social networks, parts of your body, areas of a building, etc that. Even geographic locations that we study may not conform to the standard Earth CRS definitions (like sunspots or lunar craters).
- Due to this reason if no CRS is specified, sf is still able to perform all the calculations.

Set geographic CRS



The dataset gives latitudes and longitudes near Fiji, we know they come from EPSG4326.

```
> region.geo = st_set_crs(region.sf, 4326)
```

EPSG:4326 - WGS 84, is latitude/longitude coordinate system based on the Earth's centre of mass, used by the GPS system among others.

```
> st_is_longlat(fiji.geo)
 [1] TRUE
> st_crs(fiji.geo)
> st_crs(fiji.geo)$proj4string
[1] "+proj=longlat +datum=WGS84 +no_defs
"
```

Re-project

When do we project/re-project?

- When we perform metric distance-based analyses
- When we want to compare more than one dataset with different CRS definitions
- Online publishing of data may require certain conventions

Re-project…

to Mollweide projection ("+proj=moll")

```
> fiji.proj = st_transform(fiji.geo, crs = "+proj=moll")

> st_crs(fiji.proj)
```

This has no EPSG, however.

```
> st_crs(fiji.proj)$epsg
[1] NA
```

Remember…

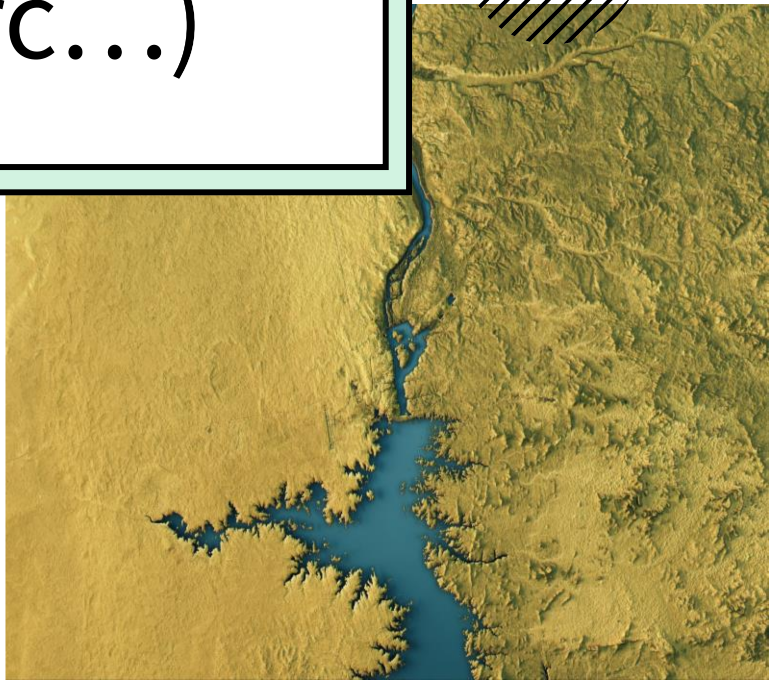
To set CRS → st_set_crs()

To change CRS → st_transform()

Which CRS?

- It depends…
- WGS84 is the most common CRS in the world, so it is worth knowing its EPSG code: 4326. However, it is un-projected.
- All projections involve a distortion when far from the centre of the specified frame. Choice of CRS should depend on
  - the properties that are most important to preserve (equal-area, equidistant, conformal or a combination of compromises) in the subsequent maps and analysis.
  - whether the analysis is local or global. A global projection may not work if the data are concentrated on a small local region.
  - Try this: `> plot(fiji.proj)`
  - *Clearly, we need an alternative projection*

Common Projections (other than merc…)



- A Lambert azimuthal equal-area projection for a custom local projection (set lon_0 and lat_0 to the center of the study area), which is an equal-area projection at all locations but distorts shapes beyond thousands of kilometres. **(+proj=laea)**
- Azimuthal equidistant projections for a specifically accurate straight-line distance between a point and the centre point of the local projection **(+proj=aeqd)**
- Web/Pseudo Mercator projection: **EPSG 3857**
- Transverse Mercator **(+proj=tmerc)**
- Robinson **(+proj=robin)**
- Universal Transverse Mercator **(UTM)**
- For available projections, visit: https://proj.org/operations/projections/aeqd.html

Universal Transverse Mercator (UTM)

A commonly used default is Universal Transverse Mercator (UTM): A set of CRSs that divides the Earth into 60 longitudinal wedges and 20 latitudinal segments.

The UTM projection is commonly used in research because it tends to be more locally accurate, and furthermore, it has attributes that make the estimating distance easy and accurate. Positions are described using Easting and Northing coordinates.

Almost every place on Earth has a UTM code

All UTM projections have the same datum (WGS84) and their EPSG codes run sequentially from 32601 to 32660 for northern hemisphere locations and from 32701 to 32760 for southern hemisphere locations.

UTM

Load function

```
> source("LonLat2UTM.R")
```

This function `lonlat2UTM()`, can accept longitudes (X) from (-180º to 180º and latitudes (Y) from -90º to 90º to give out the EPSG code related to UTM projection.

```
> sg.coords = c(103.82,1.35)
> (utm.sg = lonlat2UTM(sg.coords))
 [1] 32648

> st_crs(utm.sg)
```

Re-visit Fiji

```
> fiji.coords = c(178.07, 17.71)
> (utm.fiji = lonlat2UTM(fiji.coords))
[1] 32660
> st_crs(utm.fiji)$proj4string
[1] "+proj=utm +zone=60 +datum=WGS84 +units=m
+no_defs"
> fiji.proj3 = st_transform(fiji.geo,
crs=utm.fiji)
> plot(fiji.proj3)
```

Exercise B

Identify the UTM EPSG code and the proj4string associate with,

1. Tokyo, Japan
2. Frankfurt, Germany
3. New York, USA
4. Rio de Geneiro, Brazil
5. Johannesburg, SA
6. Beijing, China

Exercise C

Using the “world” dataset in spData, try different projections to see the map output. Some examples for you to try,

- Geodetic , EPSG4326
- Web Mercator, EPSG3857
- Robinson (crs="+proj=robin”)
- Mollweide projection, (crs="+proj=moll”)
- Lambert Azimuthal equal area projection centered at WGS84 (crs="`+proj=laea +x_0=0 +y_0=0 +lon_0=0 +lat_0=0`")
- Lambert Azimuthal equal area projection centered at Singapore

Exercise D

1. Read the Singapore planning area boundary shapefile and identify the CRS.
    a. Is it projected?
    b. What is the proj4string (note this down for future reference)
    c. What is the EPSG

Note this CRS for future reference.

Exercise E

1. Create your own proj4string. It should have
   a. the Lambert Azimuthal Equal Area (laea) projection,
   b. the WGS84 ellipsoid, the longitude of projection center of 95 degrees west, the latitude of projection center of 60 degrees north, and
   c. its units should be in meters.
2. Subset Canada from the world object and transform it into the new projection. Plot and compare a map before and after the transformation.

Take home points…

- CRSs: geographic vs projected
- Types of projections
- Describing a CRS: epsg code or proj4string definition
- UTM

Important R functions

- st_as_sf()
- st_is_longlat()
- st_set_crs()
- st_crs()
- st_transform()

References

https://geocompr.robinlovelace.net/reproj-geo-data.html