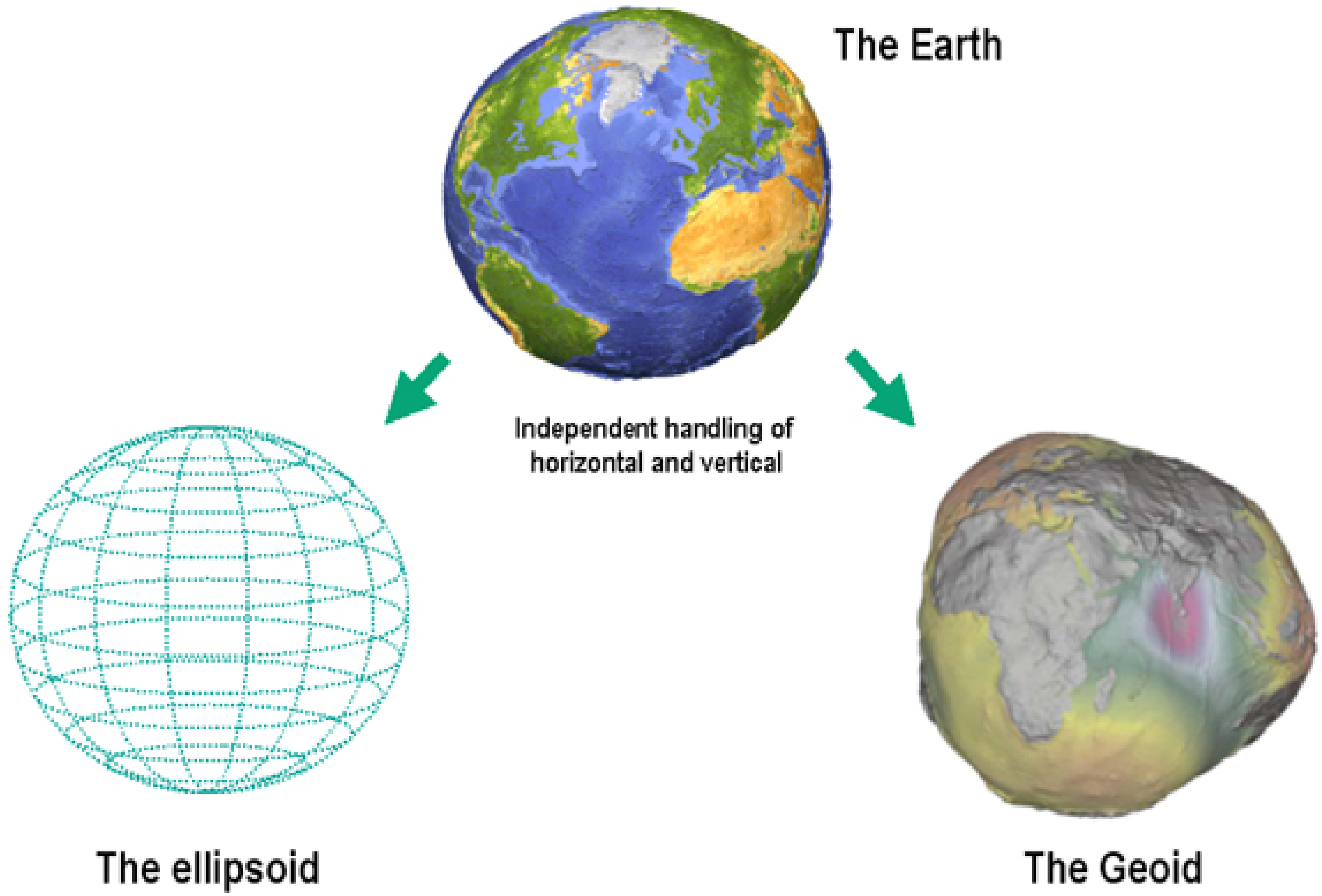
Introduction to Geospatial Vector Data with R

Ana Petrović Claudiu Forgaci

Introduction to Geospatial Concepts

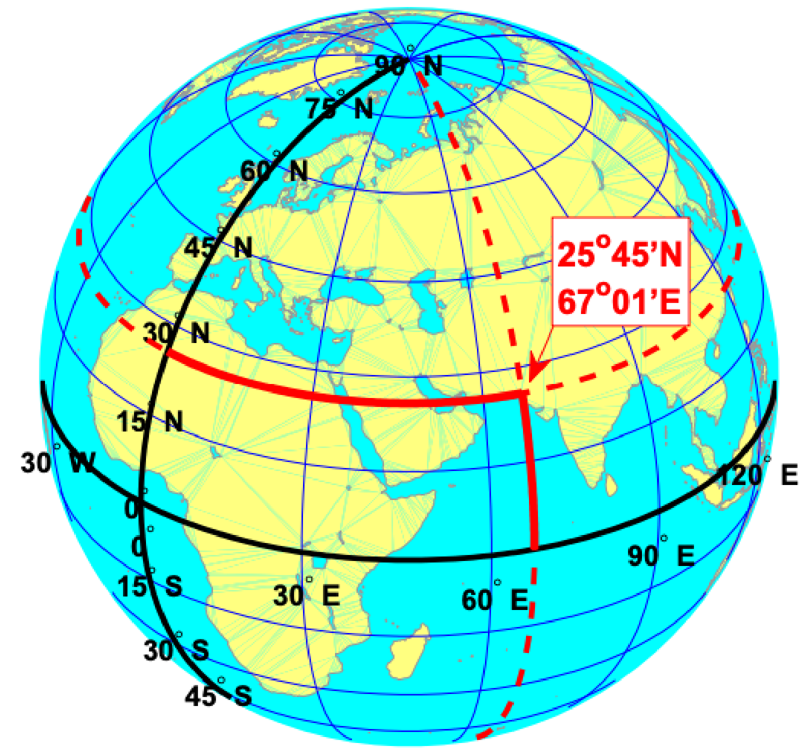
The shape of the Earth



Source: UN Statistics Division and International Cartographic Association (2012)

Geographical latitude and longitude

- **Meridians** – vertical circles with constant longitude, ***great circles***
- **Parallels** – horizontal circles with constant latitude, ***small circles***



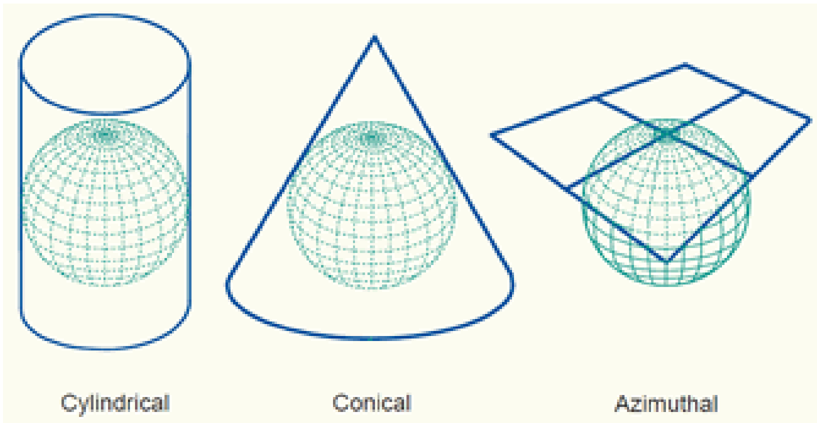
Source: van der Marel (2014)

Map projection: From the 3D Earth to a 2D map

Map projection is a systematic transformation of the latitudes and longitudes of locations on the surface of an ellipsoid into locations on a plane.

Source: Data Carpentry (2023)

Different map projections for different purposes



Source: Knippers (2009)

Each map projection introduces a **distortion** in geometrical elements – **distance**, **angle**, and **area**.

To best preserve …	use projections which are …
angles between any two curves	conformal
area or scale	equal-area (equivalent)
distances	equal-distance (conventional)

Coordinate reference systems (CRS)

A coordinate reference system (CRS) is a coordinate-based local, regional or global system for locating geographical entities, which uses a specific map projection.

Each CRS has a unique **EPSG code**.

Most used CRS in the Netherlands:

- World Geodetic System 1984 (WGS84) – EPSG:4326
- European Terrestrial Reference System 1989 (ETRS89) – EPSG:4258
- Amersfoort / RD New – EPSG:28992

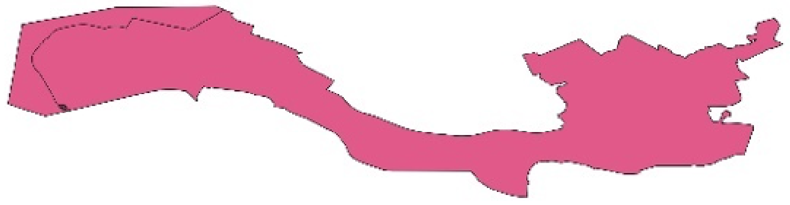
The main parameters of each CRS

- **Datum**: model of the shape of the Earth – how a coordinate system is linked to the Earth, e.g. where (0,0) is (units of measure: degrees)
- **Projection**: transformation of degrees to linear units (e.g. meters) on a flat surface
- **Additional parameters**, such as a definition of the centre of the map

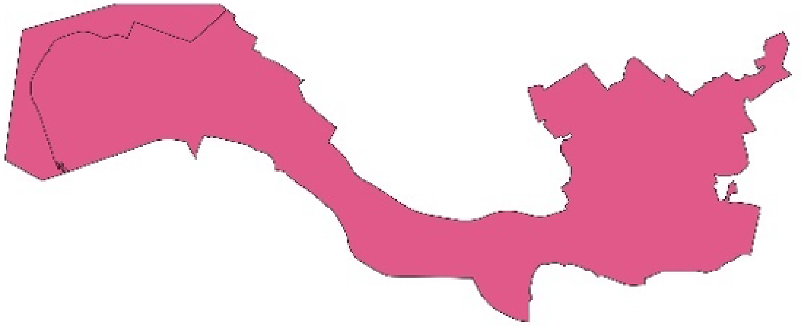
We will use these two CRS

	WGS 84 (EPSG:4326)	Amersfoort / RD New (EPSG:28992)
Definition	Dynamic (relies on a datum which is not plate-fixed)	Static (relies on a datum which is plate-fixed)
Celestial body	Earth	Earth
Ellipsoid	WGS-84	Bessel 1841
Prime meridian	International Reference Meridian	Greenwich
Datum	World Geodetic System 1984 ensemble	Amersfoort
Projection	Geographic (uses latitude and longitude for coordinates)	Projected (uses meters for coordinates)
Method	Lat/long (Geodetic alias)	Oblique Stereographic Alternative
Units	Degrees	Meters

Rotterdam in these two CRS



Rotterdam EPSG:4326



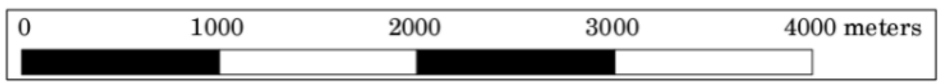
Rotterdam EPSG:28992

Map scale

Map scale measures the ratio between distance on a map and the corresponding distance on the ground.

On a 1:100 000 scale map, 1cm equals 1km (100 000 cm) on the ground.

It can be expressed as:

Verbal:	1 centimetre represents 250 meters
Fraction:	1:25000
Graphic:	

Geospatial Vector Data in R

The sf package

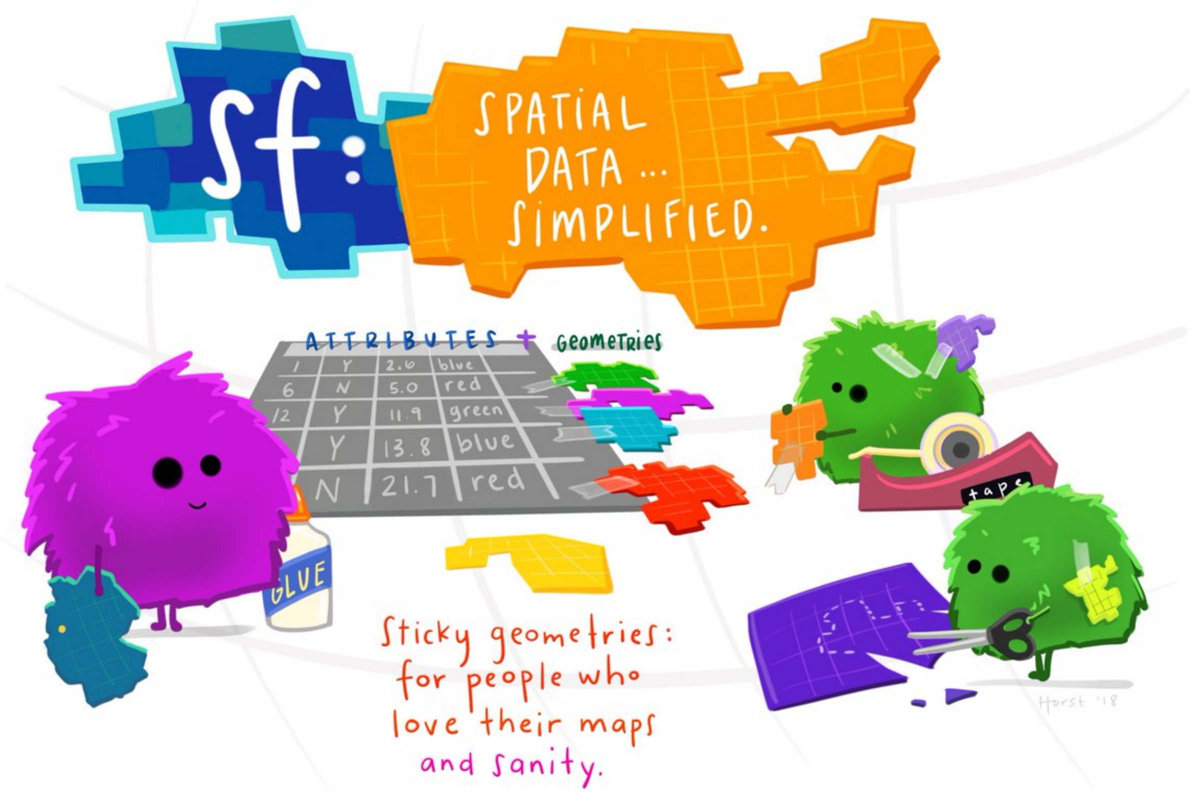


Illustration (c) 2018 by Allison Horst

The `sf` package

- `sf` provides simple features access to R
- represents simple features as records in a data frame with a `geometry` column
- function names start with a prefix `st_` ("st" stands for spatial type), which is also handy for auto-completion in RStudio

Open and Plot Shapefiles

Challenge 1: 5 mins

Read in `delft-streets.shp` and `delft-leisure.shp` and call them `lines_Delft` and `point_Delft` respectively.

Answer the following questions:

1. What type of R spatial object is created when you import each layer?
2. What is the CRS and extent for each object?
3. Do the files contain points, lines, or polygons?
4. How many spatial objects are in each file?

```r
lines_Delft <- st_read(here("episodes", "data", "delft-streets.shp"))
point_Delft <- st_read(here("episodes", "data", "delft-leisure.shp"))

st_geometry_type(lines_Delft)
st_geometry_type(point_Delft)

st_crs(lines_Delft)
st_crs(point_Delft)

st_bbox(lines_Delft)
st_bbox(point_Delft)
```

03:00

Explore and plot by vector layer attributes

Challenge 2: 3 mins

Explore the attributes associated with the `point_Delft` and `boundary_Delft` spatial objects.

1. How many attributes does each have?
2. What types of leisure points do the points represent? Give three examples.
3. Which of the following is NOT an attribute of the point_Delft data object?

A) location B) leisure C) osm_id

```
ncol(point_Delft)
ncol(boundary_Delft)

head(point_Delft)
head(point_Delft, 10)

point_Delft

names(point_Delft)
```

03:00

Challenge 3: 5 mins

1. Create a new object that only contains the motorways in Delft.
2. How many features does the new object have?
3. What is the total length of motorways?
4. Plot the motorways.
5. Extra: follow the same steps with pedestrian streets.

```r
levels(factor(lines_Delft$highway))

motorway_Delft <- lines_Delft %>%
  filter(highway == "motorway")

motorway_Delft %>%
  mutate(length = st_length(.)) %>%
  select(everything(), geometry) %>%
  summarise(total_length = sum(length))

nrow(motorway_Delft)

ggplot(data = motorway_Delft) +
  geom_sf(size = 1.5) +
  ggtitle("Mobility network of Delft", subtitle = "Motorways") +
  coord_sf()
```

05:00

Challenge 4: 3 mins

In the previous example, we set the line widths to be 1, 0.75, 0.5, and 0.25. In our case line thicknesses are consistent with the hierarchy of the selected road types, but in some cases we might want to show a different hierarchy.

Let's create another plot where we show the different line types with the following thicknesses:

- motorways size = 0.25
- primary size = 0.75
- secondary size = 0.5
- cycleway size = 1

```
levels(factor(lines_Delft$highway))

line_widths <- c(0.25, 0.75, 0.5, 1)

ggplot(data = lines_Delft_selection) +
  geom_sf(aes(size = highway)) +
  scale_size_manual(values = line_widths) +
  labs(size = "Road Size") +
  ggtitle("Mobility network of Delft", subtitle = "Roads & Cycleways - Line width varies") +
  coord_sf()
```

03:00

Challenge 5: 5 mins

Create a plot that emphasizes only roads where bicycles are allowed, as follows:

- Make the lines where bicycles are not allowed THINNER than the roads where bicycles are allowed.
- Be sure to add a title and legend to your map.
- You might consider a color palette that has all bike-friendly roads displayed in a bright color. All other lines can be black.

```r
levels(factor(lines_Delft_selection$highway))

lines_Delft_bicycle <- lines_Delft %>%
  filter(highway == "cycleway")

ggplot() +
  geom_sf(data = lines_Delft) +
  geom_sf(data = lines_Delft_bicycle, color = "magenta", size = 2) +
  ggtitle("Mobility network of Delft", subtitle = "Roads dedicated to bikes") +
  coord_sf()
```

05:00

Challenge 6: 3 mins

Create a map of the municipal boundaries in the Netherlands, as follows:

- Use the data located in your data folder: `nl-gemeenten.shp`.
- Apply a line color to each state using its region value.
- Add a legend.

```r
municipal_boundaries_NL <- st_read(here("episodes", "data", "nl-gemeenten.shp"))
str(municipal_boundaries_NL)
levels(factor(municipal_boundaries_NL$ligtInPr_1))

ggplot(data = municipal_boundaries_NL) +
  geom_sf(aes(color = ligtInPr_1), size = 1) +
  ggtitle("Contiguous NL Municipal Boundaries") +
  coord_sf()
```

03:00

Plot multiple shapefiles

Challenge 7: 5 mins

Create a map of leisure locations only including `playground` and `picnic_table`:

- Color each point by the leisure type.
- Overlay this layer on top of the `lines_Delft` layer (the streets).
- Create a custom legend that applies line symbols to lines and point symbols to the points.
- Extra: Modify the previous plot. Tell R to plot each point, using a different symbol of shape value.

```r
leisure_locations_selection <- st_read(here("episodes", "data", "delft-leisure.shp")) %>%
  filter(leisure %in% c("playground", "picnic_table"))

blue_orange <- c("cornflowerblue", "darkorange")

p <- ggplot() +
  geom_sf(data = lines_Delft_selection, aes(color = highway)) +
  scale_color_manual(name = "Line Type", values = road_colors) +
  ggtitle("Road network and leisure")

p +
  geom_sf(data = leisure_locations_selection, aes(fill = leisure), shape = 21) +
  scale_fill_manual(name = "Leisure Type", values = blue_orange)

p +
  geom_sf(data = leisure_locations_selection, aes(fill = leisure, shape = leisure), size = 3) +
  scale_fill_manual(name = "Leisure Type", values = blue_orange) +
  scale_shape_manual(name = "Leisure Type", values = c(21, 22))
```

05:00

Handling spatial projections

Challenge 8: 3 mins

Create a map of the South Holland, as follows:

1. Import `nl-gemeenten.shp` and filter only the municipalities in South Holland.
2. Plot it and adjust line width as necessary.
3. Layer the boundary of Delft onto the plot.
4. Add a title.
5. Add a legend that shows both the province boundaries (as a line) and the boundary of Delft (as a filled polygon).

```r
boundary_ZH <- municipal_boundary_NL %>%
  filter(ligtInPr_1 == "Zuid-Holland")

ggplot() +
  geom_sf(data = boundary_ZH, aes(color ="color"), show.legend = "line") +
  scale_color_manual(name = "", labels = "Municipal Boundaries", values = c("color" = "gray18")) +
  geom_sf(data = boundary_Delft, aes(shape = "shape"), color = "purple", fill = "purple") +
  scale_shape_manual(name = "", labels = "Municipality of Delft", values = c("shape" = 19)) +
  ggtitle("Delft location in South Holland") +
  theme(legend.background = element_rect(color = NA)) +
  coord_sf()
```

03:00