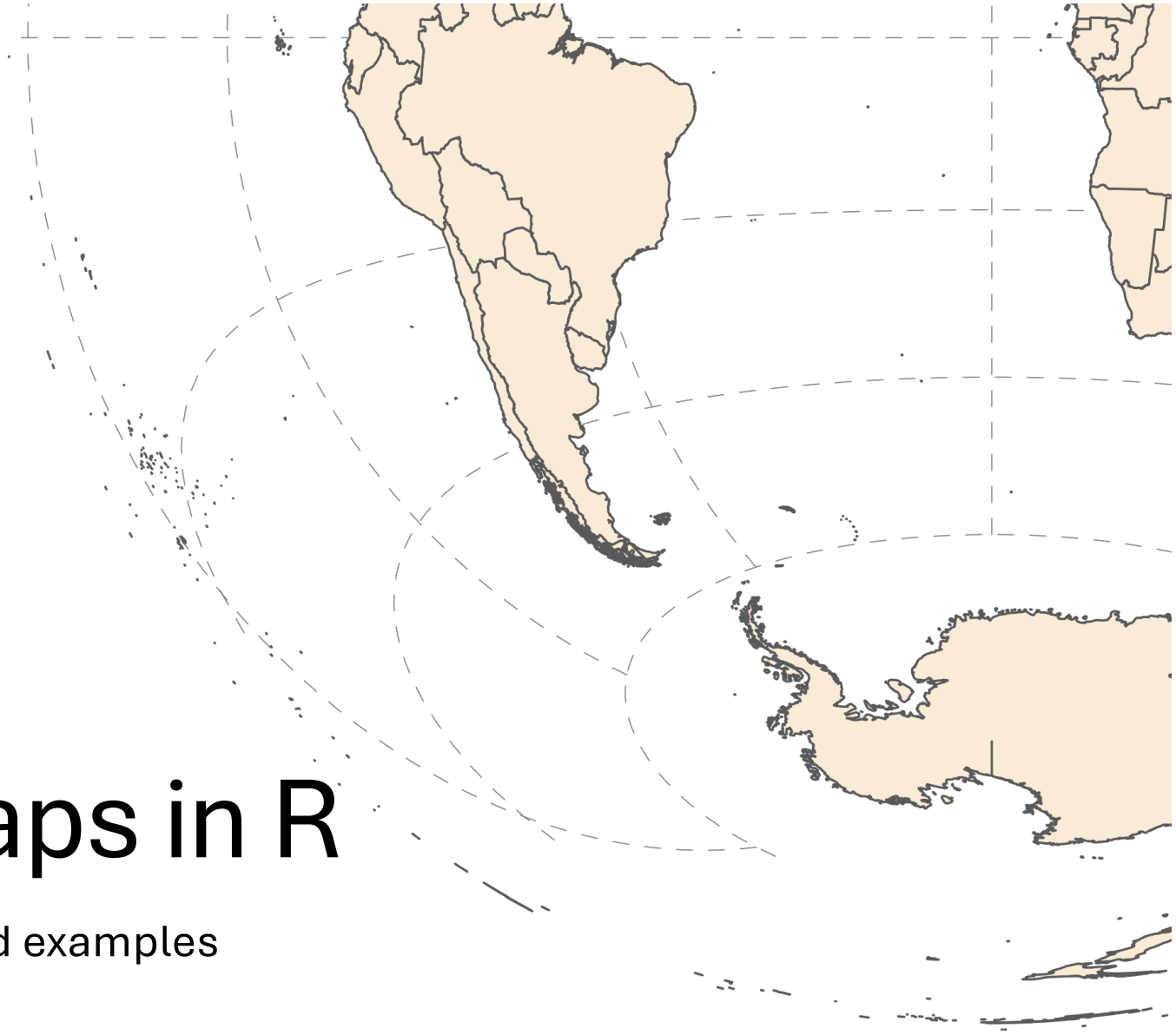
Creating maps in R

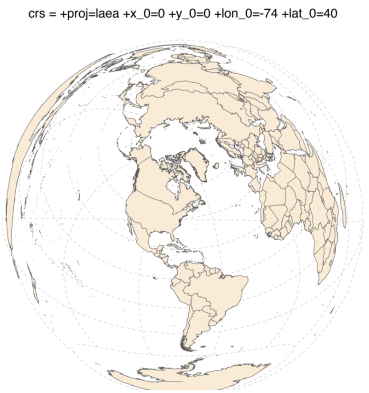
Brief introduction and examples

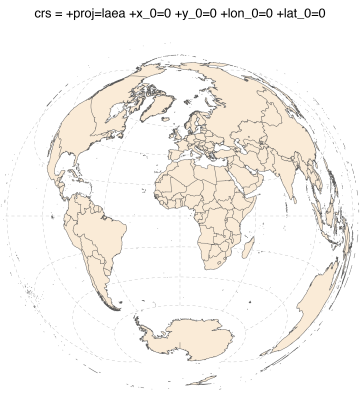


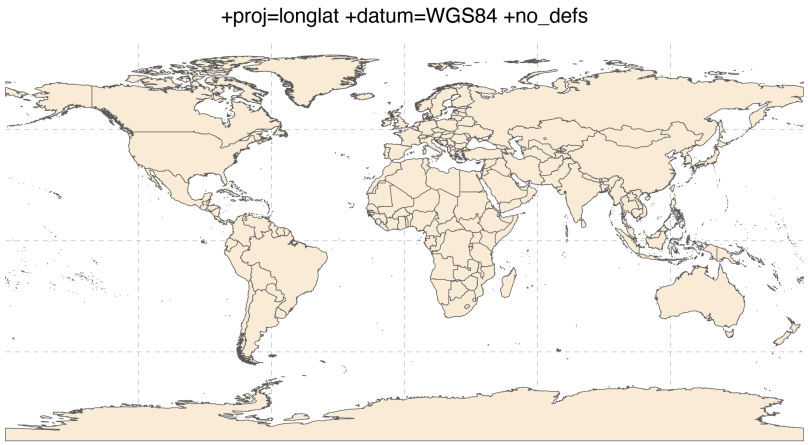
Spatial data

Geographic coordinate systems and projections

- Standardized way of describing locations to describe geographic data
- Defines the location of a point on a plane or sphere



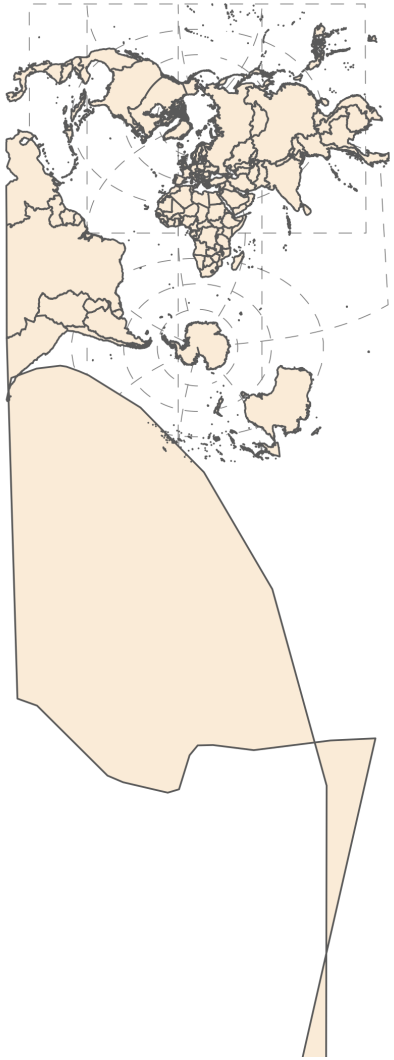




Overview: https://www.nceas.ucsb.edu/sites/default/files/2020-04/OverviewCoordinateReferenceSystems.pdf
ESPG-codes: https://spatialreference.org

SWEREF99

Projections and transformations



Geographic coordinates

- Latitude/Longitude for referencing location on the ellipsoid Earth (unprojected)
- Three-dimensional

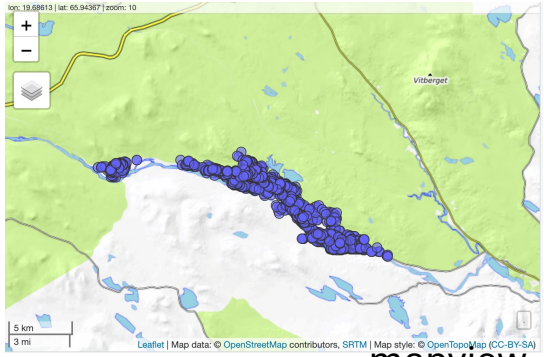
Projected coordinates

- CRS
- Easting/Northing for referencing location on 2D representations of Earth (the creation of maps) To see the options in R: projInfo(type = "proj")
- Two-dimensional
- Used for national or local grids

Packages in R

Many packages for spatial data:
e.g., ”terra”-, “raster”-, and “sf”

Packages for map visualisation:
e.g., “leaflet”, tmap” (interactive mpas, animations), “rasterVis”, “maptools”, “mapview” (interactive maps)



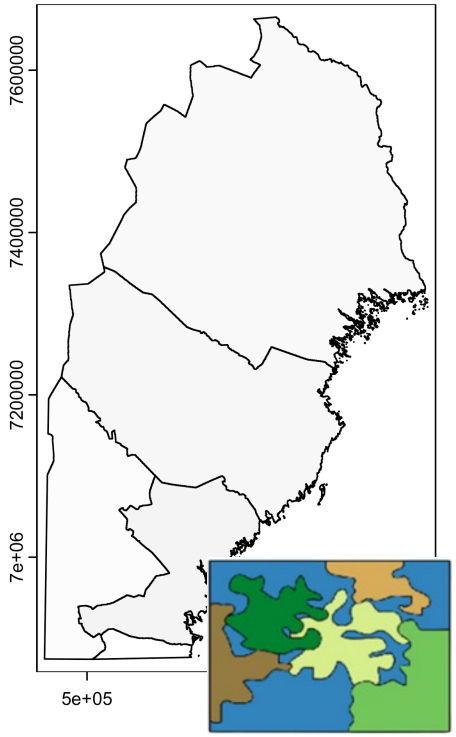
mapview

Literature e.g.,
Spatial Data Science: With Applications in R (available online)
https://r-spatial.org/book/

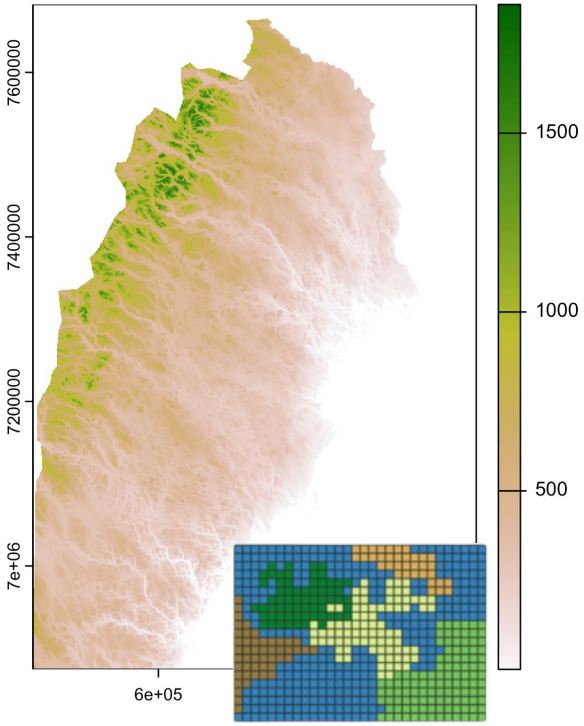
Vignettes e.g.,
https://r-spatial.github.io/sf/
https://r-spatial.github.io/mapview/
https://rstudio.github.io/leaflet/

Spatial data types

(SPATIAL) VECTOR



RASTER



Some features of spatial objects...

`coordinates` – the coordinates of the spatial objects

`bbox (or extent)` – bounding box, contains the areal extent of the spatial object

`proj4string` – the CRS object (the espg definition)

`resolution` – pixel size (of raster objects)

Data availability

R-packages for open data, e.g.,:
rnaturalearth

ggOceanMaps

Osmdata

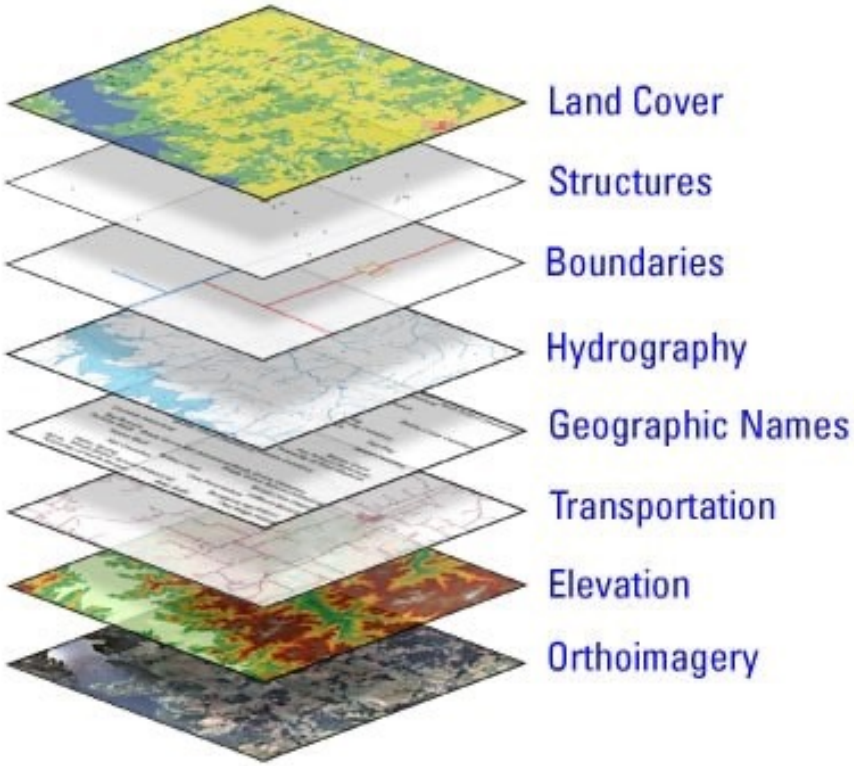
Geodata

Maptools

...

SLU server (local for Sweden):
gis.slu.se/gisdata, and the GET download service at http://maps.slu.se

See GIS support at SLU for more information



Assign known CRS or reproject (transform coords)

Set by ESPG-code or proj4string for SWEREF99 (example)

ESPG-code: 3006
Proj4string: "+proj=utm +zone=33 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs”

Assign CRS

proj4string(vector_object) <- CRS("+proj=utm ...")
crs(raster_object) <- "+proj=utm ...”

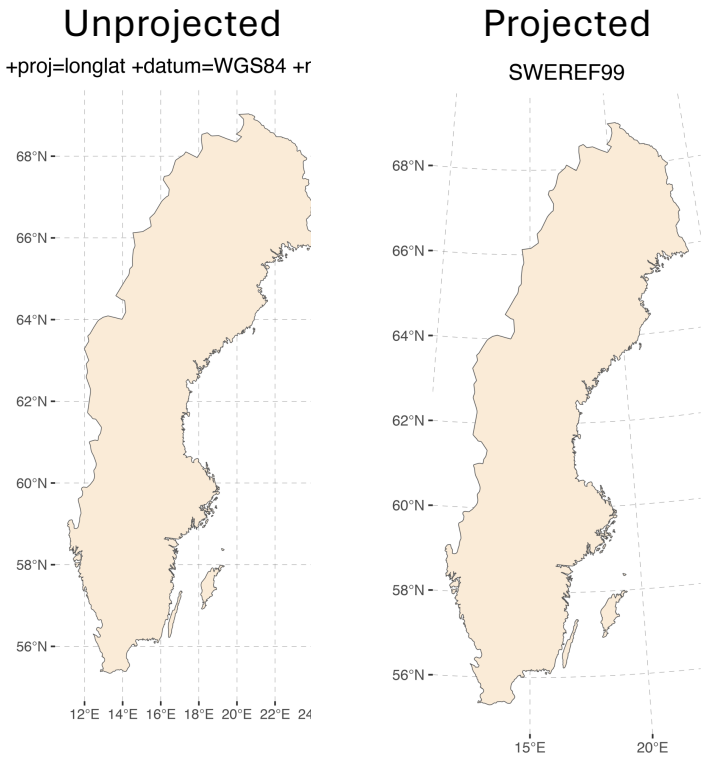
Transform CRS

Vector data

```
sf::st_transform() # sf objects
terra::project()   # spatVector objects
```

Raster data

```
raster::projectRaster() # raster objects
terra::project()        # spatRast objects
```



Maps in ggplot

VECTOR (converted to data frame)

```
ggplot2:: geom_sf()                    # sf objects
ggspatial:: geom_spatial_point()
```

RASTER (converted to data frame)

```
ggplot2::geom_raster()                 # raster objects
tidyterra::geom_spatraster()           # spatRast objects
```

Example: Lon/lat map of roads in Uppsala

```
library(sf)             # vector data
library(geodata)        # vector of roads
library(rnaturalearth) # shape of swe
library(ggplot2)
library(ggspatial)      # plot spatial in ggplot
library(dplyr)
```

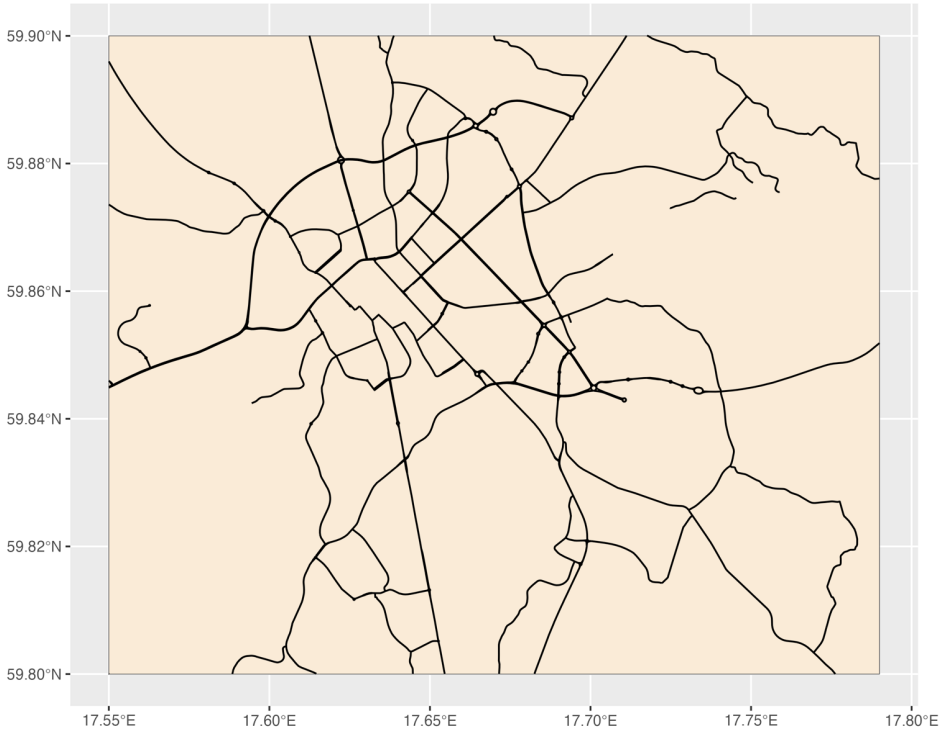
```
swe <- rnaturalearth::ne_countries(country="sweden",
                                   scale = 10) %>%
               st_as_sf(.) %>%                          #  has no be sf object
               st_crop(., xmin = 17.55, xmax= 17.79,
                         ymin=59.9, ymax=59.8)          # bbox of Uppsala (crop)
```

```
roads <- geodata::osm(country="sweden","highways",path=tempdir()) %>%
  st_as_sf(.) %>%
  st_crop(., xmin = 17.55, xmax= 17.79, ymin=59.9, ymax=59.8)
```

Plot using ggplot2

```
ggplot() +

geom_sf(data = swe,
          aes(geometry = geometry),
          fill= "antiquewhite")+
  geom_sf(data = roads,
          aes(geometry = geometry),
          fill= "darkgrey")
```



Included in a map

- reference grid lines (parallels, meridians) may be required
- axes tics usually show little, but some information
- custom elements necessary to for interpretation (arrow, scale bar, multi-type legend)
- label placement more challenging (in R)

Import/export spatial data into R

Import

Vector data (.shp)
sf::st_read() # simple feature

sf::read_sf()

Terra::vect() #SpatVector

Raster data (e.g., .tif)

raster::raster() #RasterLayer

terra::rast() # SpatRast

raster::stack() # several rasters

Export

st_write()

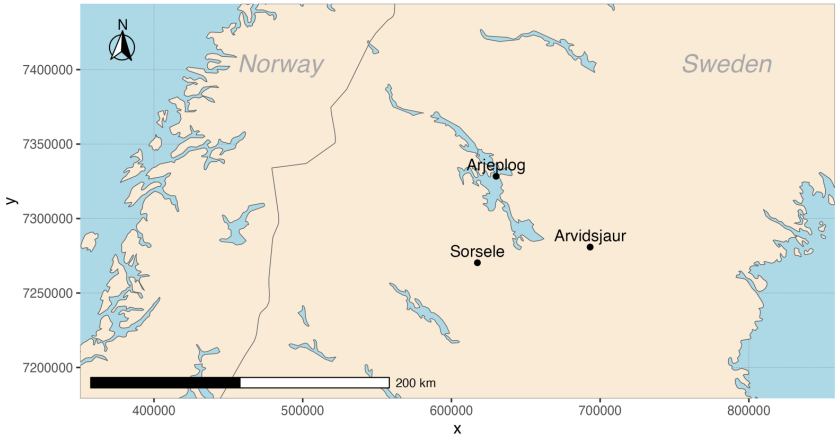
writeRaster() # SpatRast, RasterLayer

More examples

Available at:

https://github.com/HRautiainen/create-maps

Exercise 1: map using vector data



Exercise 2 (optional): combining vector and raster data