assignment_1

Marco Valentino Caravella, Davide Delfino, Elena Mannino, Sara Tozzi

2024-10-09

Commands to set the directory:

```
getwd()
```

```
## [1] "C:/Users/Davide/Desktop/Alma Mater/SECOND YEAR/GIS/Assignment"
```

```
path <- "C:/Users/Davide/Desktop/Alma Mater/SECOND YEAR/GIS/Assignment"
setwd(path)
```

Those are the libraries we will need to fulfill each task:

```
library(sf)
```

```
## Linking to GEOS 3.12.1, GDAL 3.8.4, PROJ 9.3.1; sf_use_s2() is TRUE
```

```
library(spData)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(readxl)
library(dplyr)
library(viridis)
```

```
## Caricamento del pacchetto richiesto: viridisLite
```

```r
library(exactextractr)
library(cowplot)
```

```
##
## Caricamento pacchetto: 'cowplot'
##
## Il seguente oggetto è mascherato da 'package:lubridate':
##
##     stamp
```

```r
library(ggpubr)
```

```
##
## Caricamento pacchetto: 'ggpubr'
##
## Il seguente oggetto è mascherato da 'package:cowplot':
##
##     get_legend
```

```r
library(patchwork)
```

```
##
## Caricamento pacchetto: 'patchwork'
##
## Il seguente oggetto è mascherato da 'package:cowplot':
##
##     align_plots
```

```r
library(gridExtra)
```

```
##
## Caricamento pacchetto: 'gridExtra'
##
## Il seguente oggetto è mascherato da 'package:dplyr':
##
##     combine
```

```r
library(s2)
library(ggmap)
```

```
## i Google's Terms of Service: <https://mapsplatform.google.com>
##   Stadia Maps' Terms of Service: <https://stadiamaps.com/terms-of-service/>
##   OpenStreetMap's Tile Usage Policy: <https://operations.osmfoundation.org/policies/tiles/>
## i Please cite ggmap if you use it! Use `citation("ggmap")` for details.
##
## Caricamento pacchetto: 'ggmap'
##
##
## Il seguente oggetto è mascherato da 'package:cowplot':
##
##     theme_nothing
```

```r
library(broom)
library(rmapshaper)
library(ggrepel)
library("RColorBrewer")
library(parallel)
library(conflicted)
library(validate)
library(assertr)
library(fixest)
library(feather)
library(haven)
library(gmapsdistance)
library(units)
```

```
## udunits database from C:/Users/Davide/AppData/Local/R/win-library/4.4/units/share/udunits/udunits2.xm
```

```r
library(ggpubr)
library(elevatr)
```

```
## elevatr v0.99.0 NOTE: Version 0.99.0 of 'elevatr' uses 'sf' and 'terra'.  Use
## of the 'sp', 'raster', and underlying 'rgdal' packages by 'elevatr' is being
## deprecated; however, get_elev_raster continues to return a RasterLayer.  This
## will be dropped in future versions, so please plan accordingly.
```

```r
library(raster)
```

```
## Caricamento del pacchetto richiesto: sp
```

```r
conflicts_prefer(dplyr::filter) #to avoid any kind of conflict error
```

```
## [conflicted] Will prefer dplyr::filter over any other package.
```

This will install a minimal LaTeX distribution (TinyTeX), sufficient for generating PDF files.

```r
tinytex::install_tinytex(force = TRUE)
```

```
## tlmgr --repository http://www.preining.info/tlgpg/ install tlgpg
```

```
## tlmgr option repository "https://ctan.mirror.garr.it/mirrors/ctan/systems/texlive/tlnet"
```

```
## tlmgr update --list
```

BRAZIL *Road Infrastructure*

Paper: Morten, M. & Oliveira, J., 2018. *The Effects of Roads on Trade and Migration: Evidence from a Planned Capital City*

Aim: Replicate Figure 1 from the paper, which illustrates *Brazil's capital road infrastructure.*

```r
secdata <- file.path(path,"data","Brazil")
```

Import dataset on Brasil states and highways.

```r
states <- st_read(file.path(secdata,"/uf1940/uf1940_prj.shp"))
```

```
## Reading layer `uf1940_prj' from data source
##   `C:\Users\Davide\Desktop\Alma Mater\SECOND YEAR\GIS\Assignment\data\Brazil\uf1940\uf1940_prj.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 21 features and 1 field
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -73.99045 ymin: -33.75118 xmax: -28.84764 ymax: 5.271841
## Geodetic CRS:  SIRGAS 2000
```

```r
states_simple  <- rmapshaper::ms_simplify(states,keep=0.01,keep_shapes=TRUE)

year  <- 2000
file_name = paste0("/",year,"/highways_",year,"_prj.shp")
    all_highways <- st_read(file.path(secdata, "roads", file_name))
```

```
## Reading layer `highways_2000_prj' from data source
##   `C:\Users\Davide\Desktop\Alma Mater\SECOND YEAR\GIS\Assignment\data\Brazil\roads\2000\highways_200
##   using driver `ESRI Shapefile'
## Simple feature collection with 5152 features and 34 fields
## Geometry type: LINESTRING
## Dimension:     XY
## Bounding box:  xmin: -72.95826 ymin: -33.69253 xmax: -34.82812 ymax: 4.483634
## Geodetic CRS:  SIRGAS 2000
```

```r
all_highways_simple  <- rmapshaper::ms_simplify(all_highways,keep=0.01,
                                                keep_shapes=TRUE)
```

Read in Minimum Spanning Tree (MST) road network

```r
mst_pie <- st_read(file.path(secdata, "mst/mst_pie_prj.shp"))
```

```
## Reading layer `mst_pie_prj' from data source
##   `C:\Users\Davide\Desktop\Alma Mater\SECOND YEAR\GIS\Assignment\data\Brazil\mst\mst_pie_prj.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 26 features and 4 fields
## Geometry type: LINESTRING
## Dimension:     XY
## Bounding box:  xmin: -67.812 ymin: -30.04 xmax: -34.82257 ymax: 2.817999
## Geodetic CRS:  SIRGAS 2000
```

```r
mst_pie_simple  <- rmapshaper::ms_simplify(mst_pie,keep=0.01,keep_shapes=TRUE)
```

Read in cities

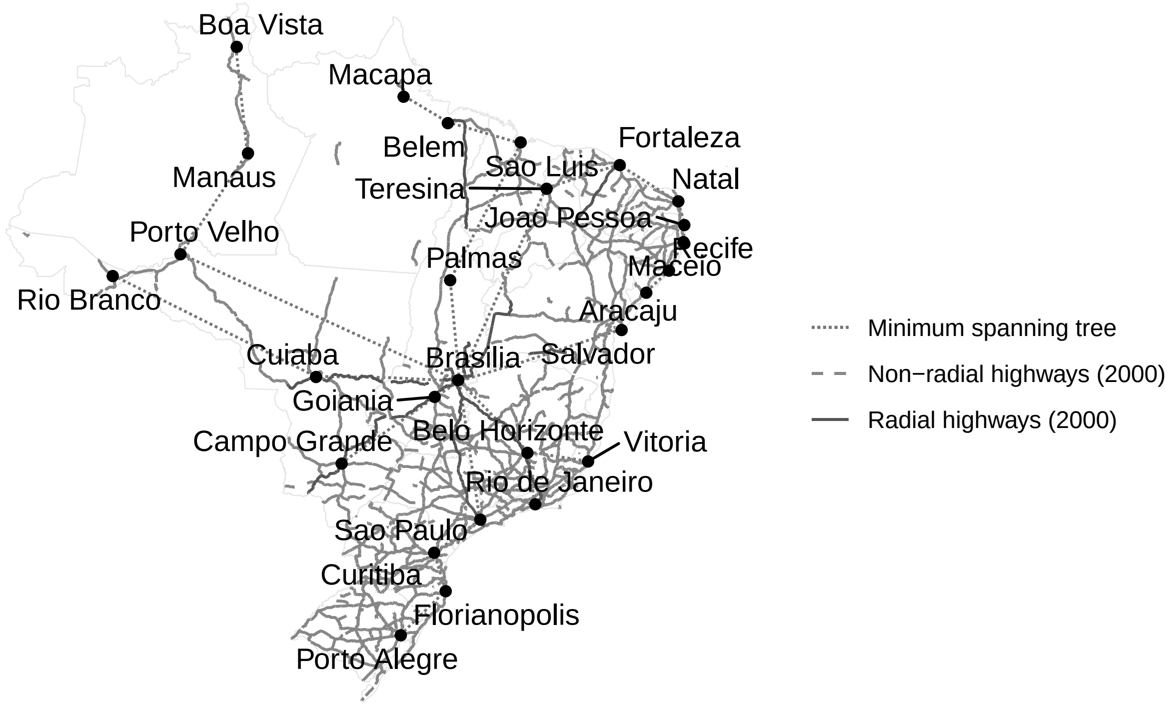
```r
capital_cities <- st_read(file.path(secdata,
                                    "/cities/brazil_capital_cities_prj.shp"))
```

```
## Reading layer 'brazil_capital_cities_prj' from data source
##   'C:\Users\Davide\Desktop\Alma Mater\SECOND YEAR\GIS\Assignment\data\Brazil\cities\brazil_capital_c
##   using driver 'ESRI Shapefile'
## Simple feature collection with 27 features and 14 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: -67.812 ymin: -30.04 xmax: -34.867 ymax: 2.817999
## Geodetic CRS:  SIRGAS 2000
```

```r
cities_xy <- cbind(capital_cities,st_coordinates(capital_cities))
```

Plotting Figure 1 : Map of straight-line instrument and radial highways

```r
ggplot() +
  geom_sf(data=states_simple, fill="white", color="grey90") +
  geom_sf(data=mst_pie_simple, size=.6, linetype = "11",
          aes(color = "Minimum spanning tree"), show.legend = "line")   +
  geom_sf(data=dplyr::filter(all_highways_simple, dm_anlys_p==1 & dm_radial==0),
          size=.3, linetype = "dashed", aes(color = "Non-radial highways (2000)"),
          show.legend = "line") +
  geom_sf(data=dplyr::filter(all_highways_simple, dm_anlys_p==1 & dm_radial==1),
          size=0.6, aes(color = "Radial highways (2000)"), show.legend = "line") +
  theme_minimal() +
  theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid = element_blank()) +
  geom_point(data=cities_xy,aes(x=X,y=Y)) +
  geom_text_repel(data=cities_xy,aes(x=X,y=Y,label=CITY_NAME)) +
  labs(color = " ") +
  scale_color_manual(values=c("#777676","#868686","#565555"),
                     guide = guide_legend(override.aes =
                                            list(linetype = c("11",
                                                              "dashed",
                                                              "solid"))))
```



Now we clean the environment for the execution of the next task.

```
rm(list=ls())
```

BRAZIL *meso-regions population density*

Paper: Pellegrina, H.S. and Sotelo, S., 2021. *Migration, Specialization, and Trade: Evidence from Brazil's March to the West*

Aim: Replicate Figure 2 from the paper, which illustrates *Population in Brazil's meso-regions (or districts) in different periods*

Import data set on Brasil roads and population. Recall to turn into sf the shape file. *Attention*:The original Excel file did have formatting issues, such as empty cells in the first row, which could affected the import process. Due to this issues some adjustment to the original dataset is been performed

```
br_mesoregiones <-"data/BR_Mesorregioes_2021.shp"
sf.br_mesoregiones <- st_read(br_mesoregiones)
```

```
## Reading layer 'BR_Mesorregioes_2021' from data source
##   'C:\Users\Davide\Desktop\Alma Mater\SECOND YEAR\GIS\Assignment\data\BR_Mesorregioes_2021.shp'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 139 features and 3 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
```

```
## Bounding box:  xmin: -73.99045 ymin: -33.75118 xmax: -28.84764 ymax: 5.271841
## Geodetic CRS:  SIRGAS 2000
```

```
pop_mesoregiones <- read_excel("data/BrazilPopulation.xlsx")
```

Before plotting the whole density we need to identify Western regions and aggregate them togheter using the function st_union(). One way is to create manually a new variable to differenciate West and East. To do that we have visualized the location of each region SIGLA in a plot using ggplot.

```
West <- c("AC", "AM", "AP", "GO", "MA", "MT", "MS", "PA", "TO","RO", "RR")
East <- c("AL", "BA", "CE", "DF", "ES", "MG", "PB", "PE", "PI", "PR", "RJ",
          "RN", "RS", "SC", "SE", "SP")

sf.br_mesoregiones <- sf.br_mesoregiones %>%
  mutate(regiones = case_when(
    SIGLA %in% West ~ "West",
    SIGLA %in% East ~ "East",
    TRUE ~ "Altro"
  ))

west_regions <- sf.br_mesoregiones[sf.br_mesoregiones$regiones == "West", ]
west_union <- st_union(west_regions)
west_union_sf <- st_sf(geometry = st_union(west_union))
```

Since we do not have data on meso-regiones population density we left_join the original dataset on Brazil with a dataset that still doesn't have any info on density, but at least collects population's data.

```
Brazil <- sf.br_mesoregiones %>%
  left_join(pop_mesoregiones,by = c("NM_MESO" =  "Brasil, Grande Região, Unidade da Federação e Mesorre
```

We try to create three plots on the evolution of Brazil's population density in 30-years period. First, we convert in factor years and then we compute the density for each year. Since in every year observations on population is missing for Lago mirim and Lago dos patos, we can exclude them from the dataset

```
# Na observations has been remuved from "1991"
Brazil <- Brazil %>%
  filter(!is.na(`1991`))  # we used the backticks because the coloumn was named by numbers
```

We compute the population density after converting our variables of interest into numerical variables.

```
Brazil$`1991` <- as.numeric(as.character(Brazil$`1991`))
Brazil$`2000` <- as.numeric(as.character(Brazil$`2000`))
Brazil$`2010` <- as.numeric(as.character(Brazil$`2010`))
```

Total population by year of each meso region over the mean

```
pop_mean_1991 <- mean(Brazil$`1991`, na.rm = TRUE)
pop_mean_2000 <- mean(Brazil$`2000`, na.rm = TRUE)
pop_mean_2010 <- mean(Brazil$`2010`, na.rm = TRUE)
```

We create a new variable for every year population share

```r
Brazil <- Brazil %>%
  mutate(
    density_1991 = `1991` / pop_mean_1991,
    density_2000 = `2000` / pop_mean_2000,
    density_2010 = `2010` / pop_mean_2010
  )
```

We create manually the categories based on the density values for each year and we assign a color to each category.

```r
Brazil <- Brazil %>%
  mutate(density_category_1991 = cut(density_1991,
                                     breaks = c(-Inf, 0.5, 1, 2, 4, Inf),
                                     labels = c("Very Low", "Low", "Medium", "High", "Very High"),
                                     right = FALSE))


Brazil <- Brazil %>%
  mutate(density_category_2000 = cut(density_2000,
                                     breaks = c(-Inf, 0.5, 1, 2, 4, Inf),
                                     labels = c("Very Low", "Low", "Medium", "High", "Very High"),
                                     right = FALSE))


Brazil <- Brazil %>%
  mutate(density_category_2010 = cut(density_2010,
                                     breaks = c(-Inf, 0.5, 1, 2, 4, Inf),
                                     labels = c("Very Low", "Low", "Medium", "High", "Very High"),
                                     right = FALSE))


colors <- c("Very Low" = "#f0f8ff",
            "Low" = "#87ceeb",
            "Medium" = "#4682b4",
            "High" = "blue",
            "Very High" = "navy")
```

Now we create one graph for each year of interest. Note that, since the data we used differ from those of the paper we refer to, the graphs are not exactly equal.

```r
legend_title <- "pop.share"

# plot for year 1991
plot_1991 <- ggplot() +
  geom_sf(data = Brazil, aes(fill = density_category_1991)) +
  geom_sf(data = west_union_sf, fill = NA, color = "red", size = 0.5) +
  scale_fill_manual(values = colors) +
  theme_minimal() +
  theme(legend.text = element_text(size = 6),
        legend.title = element_text(size = 8),
        panel.grid = element_blank(),
        legend.position = c(0, 0),
        legend.justification = c(0, 0),
```

```r
        axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank()) +
  guides(fill = guide_legend(keywidth = 0.5, keyheight = 0.5,title = legend_title)) +
  ggtitle("1991") + theme(plot.title = element_text(hjust = 0.5))
```

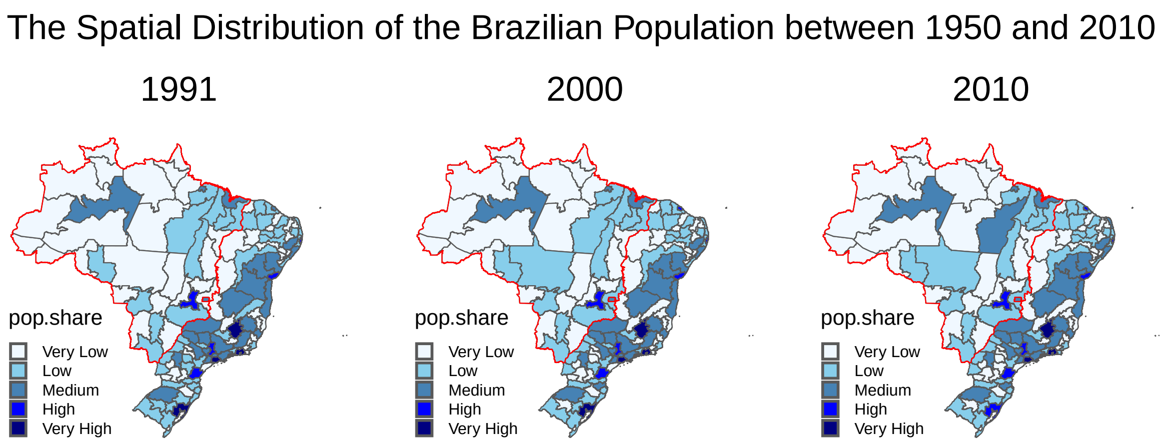
```
## Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2
## 3.5.0.
## i Please use the `legend.position.inside` argument of `theme()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
#plot for year 2000
plot_2000 <- ggplot() +
  geom_sf(data = Brazil, aes(fill = density_category_2000)) +
  geom_sf(data = west_union_sf, fill = NA, color = "red", size = 0.5) +
  scale_fill_manual(values = colors) +
  theme_minimal() +
  theme(legend.text = element_text(size = 6),
        legend.title = element_text(size = 8),
        panel.grid = element_blank(),
        legend.position = c(0, 0),
        legend.justification = c(0, 0),
        axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank()) +
  guides(fill = guide_legend(keywidth = 0.5, keyheight = 0.5,title = legend_title)) +
  ggtitle("2000") + theme(plot.title = element_text(hjust = 0.5))

# plot for year 2010
plot_2010 <- ggplot() +
  geom_sf(data = Brazil, aes(fill = density_category_2010)) +
  geom_sf(data = west_union_sf, fill = NA, color = "red", size = 0.5) +
  scale_fill_manual(values = colors) +
  theme_minimal() +
  theme(legend.text = element_text(size = 6),
        legend.title = element_text(size = 8),
        panel.grid = element_blank(),
        legend.position = c(0, 0),
        legend.justification = c(0, 0),
        axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank()) +
  guides(fill = guide_legend(keywidth = 0.5, keyheight = 0.5,title = legend_title)) +
  ggtitle("2010") + theme(plot.title = element_text(hjust = 0.5))

# combine graphs with an unique title
combined_plot <- plot_1991 + plot_2000 + plot_2010 + plot_layout(ncol = 3)
final_plot <- combined_plot +
  plot_annotation(title = "The Spatial Distribution of the Brazilian Population between 1950 and 2010",
                  theme = theme(plot.title = element_text(size = 13, hjust = 0.5)))
```

```r
print(final_plot)
```



Clean the enviorment for the execution of the next task

```r
rm(list=ls())
```

SOUTH AFRICA

Paper: Mettetal, E., 2019. *Irrigation dams, water and infant mortality: Evidence from South Africa*

Aim: Replicate Figure 2 from the paper, which illustrates *hydro dams in South Africa*

Load datasets

```r
dams.shp <- "data/mygeodata_merged.shp"
sf.dams <- st_read(dams.shp)
```

```
## Reading layer 'mygeodata_merged' from data source
##   'C:\Users\Davide\Desktop\Alma Mater\SECOND YEAR\GIS\Assignment\data\mygeodata_merged.shp'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 5717 features and 11 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 17.5755 ymin: -34.67222 xmax: 33.57778 ymax: -19.19806
## Geodetic CRS:  WGS 84
```

```r
SouthAfrica.shp <- "data/LocalMunicipalities2018_Final.shp"
sf.SouthAfrica <- st_read(SouthAfrica.shp)
```

```
## Reading layer `LocalMunicipalities2018_Final' from data source
##   `C:\Users\Davide\Desktop\Alma Mater\SECOND YEAR\GIS\Assignment\data\LocalMunicipalities2018_Final.s
##   using driver `ESRI Shapefile'
## Simple feature collection with 213 features and 11 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -534343.5 ymin: 6122919 xmax: 1091131 ymax: 7550944
## Projected CRS: WGS 84 / UTM zone 35S
```

Load river gradient raster

```r
river_gradient_raster <- raster("data/South_Africa_SRMT30meters.tif")
```

Ensure CRS alignment between shapefiles and raster

```r
sf.SouthAfrica <- st_transform(sf.SouthAfrica, crs(river_gradient_raster))
```

Add the extracted river gradient data to the shapefile

```r
river_gradients_by_region <- exact_extract(river_gradient_raster, sf.SouthAfrica, 'mean')
```

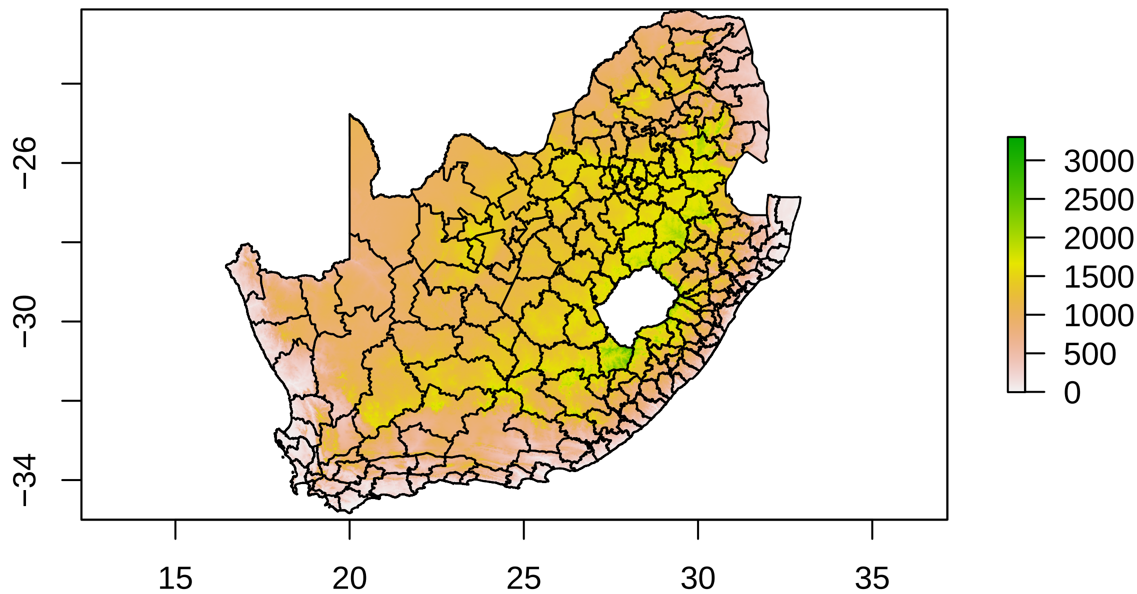
```
## Cannot preload entire working area of 2715326792 cells with max_cells_in_memory = 3e+07. Raster valu

##   |                                                                              |
```

```r
sf.SouthAfrica$avg_river_gradient <- river_gradients_by_region
st_crs(sf.SouthAfrica) == crs(river_gradient_raster)
```

```
## [1] FALSE
```

```r
plot(river_gradient_raster)
plot(sf.SouthAfrica$geometry, add=TRUE)
```



Check CRS of the raster (optional)

```r
crs(river_gradient_raster)
```

```
## Coordinate Reference System:
## Deprecated Proj.4 representation: +proj=longlat +datum=WGS84 +no_defs
## WKT2 2019 representation:
## GEOGCRS["unknown",
##     DATUM["World Geodetic System 1984",
##         ELLIPSOID["WGS 84",6378137,298.257223563,
##             LENGTHUNIT["metre",1]],
##         ID["EPSG",6326]],
##     PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433],
##         ID["EPSG",8901]],
##     CS[ellipsoidal,2],
##         AXIS["longitude",east,
##             ORDER[1],
##             ANGLEUNIT["degree",0.0174532925199433,
##                 ID["EPSG",9122]]],
##         AXIS["latitude",north,
##             ORDER[2],
##             ANGLEUNIT["degree",0.0174532925199433,
##                 ID["EPSG",9122]]]]
```

Check CRS of the vector data (optional)

```r
st_crs(sf.SouthAfrica)
```

```
## Coordinate Reference System:
##   User input: GEOGCRS["unknown",
##     DATUM["World Geodetic System 1984",
##         ELLIPSOID["WGS 84",6378137,298.257223563,
##             LENGTHUNIT["metre",1]],
##         ID["EPSG",6326]],
##     PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433],
##         ID["EPSG",8901]],
##     CS[ellipsoidal,2],
##         AXIS["longitude",east,
##             ORDER[1],
##             ANGLEUNIT["degree",0.0174532925199433,
##                 ID["EPSG",9122]]],
##         AXIS["latitude",north,
##             ORDER[2],
##             ANGLEUNIT["degree",0.0174532925199433,
##                 ID["EPSG",9122]]]]
##   wkt:
## GEOGCRS["unknown",
##     DATUM["World Geodetic System 1984",
##         ELLIPSOID["WGS 84",6378137,298.257223563,
##             LENGTHUNIT["metre",1]],
##         ID["EPSG",6326]],
##     PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433],
##         ID["EPSG",8901]],
##     CS[ellipsoidal,2],
##         AXIS["longitude",east,
##             ORDER[1],
##             ANGLEUNIT["degree",0.0174532925199433,
##                 ID["EPSG",9122]]],
##         AXIS["latitude",north,
##             ORDER[2],
##             ANGLEUNIT["degree",0.0174532925199433,
##                 ID["EPSG",9122]]]]
```

Remove NA

```r
sf.SouthAfrica <- sf.SouthAfrica %>% filter(!is.na(avg_river_gradient))
```

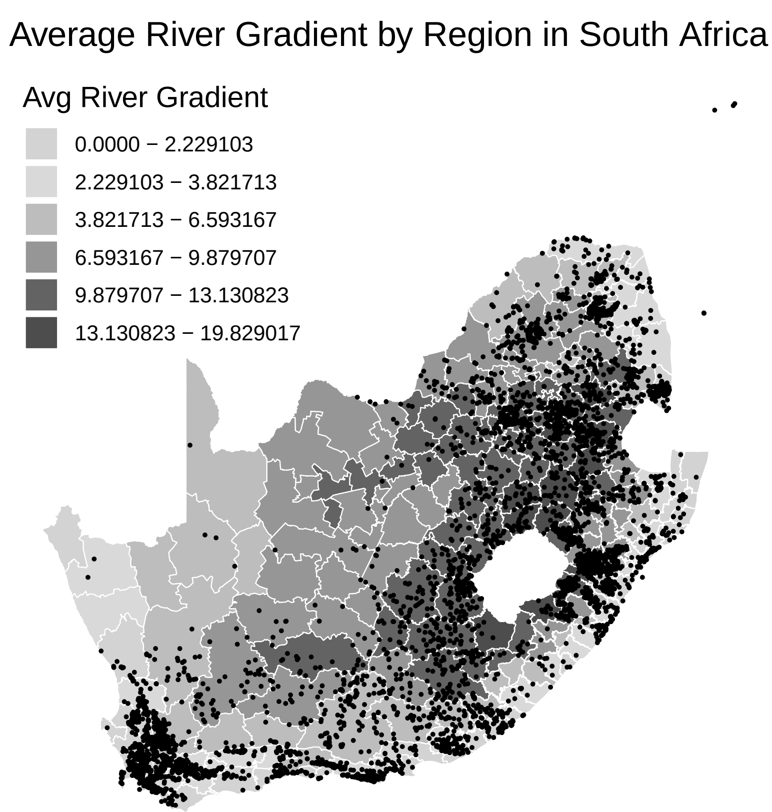
Classify the average river gradient into 6 intervals

```r
sf.SouthAfrica <- sf.SouthAfrica %>%
  mutate(gradient_class = cut(avg_river_gradient,
                              breaks = 6,
                              labels = c("0.0000 - 2.229103",
                                         "2.229103 - 3.821713",
                                         "3.821713 - 6.593167",
                                         "6.593167 - 9.879707",
```

```
                                          "9.879707 - 13.130823",
                                          "13.130823 - 19.829017"),
                               include.lowest = TRUE))
```

Create the plot

```
ggplot() +
  geom_sf(data = sf.SouthAfrica, aes(fill = gradient_class), color = "white") +
  geom_sf(data = sf.dams, size = 0.2, color = "black") +
  scale_fill_manual(values = c("lightgrey", "#d9d9d9",
                               "#bdbdbd", "#969696", "#636363", "#4d4d4d"),
                    name = "Avg River Gradient") +
  theme_minimal() +
  labs(title = "Average River Gradient by Region in South Africa",
       fill = "Avg River Gradient") +
  theme(
    legend.position = c(0, 1),
    legend.justification = c(0, 1),
    legend.direction = "vertical",
    legend.key.size = unit(0.5, "cm"), # Make the legend keys smaller
    legend.text = element_text(size = 8),
    plot.title = element_text(hjust = 0.5),
    panel.grid = element_blank(),
    axis.text = element_blank(),
    axis.ticks = element_blank()
  )
```



We get a slightly different plot with respect to the figure we want to replicate. Differences are due to the datasets. We did not find the same District boundaries and this probably caused the average river gradient to be computed over different regions in terms of dimension.

Clean the enviorment for the execution of the next task

```
rm(list=ls())
```

ETHIOPIA *distircts*

Paper: Fried, S. and Lagakos, D., 2021. *Rural electrification, migration and structural transformation: Evidence from Ethiopia*

Aim: Replicate Figure 2 from the paper, which illustrates *districts and electricity grid in Ethiopia*

Load the datasets

```
#data that ontains data for the major roads of Ethiopia
shproads<-"data/Ethiopia_Roads.shp"
sf.roads<-st_read(shproads)
```

```
## Reading layer `Ethiopia_Roads' from data source
##   `C:\Users\Davide\Desktop\Alma Mater\SECOND YEAR\GIS\Assignment\data\Ethiopia_Roads.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 76 features and 16 fields
## Geometry type: MULTILINESTRING
## Dimension:     XY, XYM
## Bounding box:  xmin: 35.58148 ymin: 5.6777 xmax: 43.5421 ymax: 14.33759
## m_range:       mmin: 0 mmax: 161.7112
## Geodetic CRS:  WGS 84
```

```
#data thatcontains the spatial data for Electricity grids across the African continent
shppwr<-"data/africa_grid_20170906final.shp"
sf.power<-st_read(shppwr)
```

```
## Reading layer `africa_grid_20170906final' from data source
##   `C:\Users\Davide\Desktop\Alma Mater\SECOND YEAR\GIS\Assignment\data\africa_grid_20170906final.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 62001 features and 8 fields
## Geometry type: LINESTRING
## Dimension:     XY
## Bounding box:  xmin: -17.50856 ymin: -34.77055 xmax: 59.52648 ymax: 37.14215
## Geodetic CRS:  WGS 84
```

```
#power plant data
plantsafr2<-read_excel("data/RePP_Petersetal.xlsx", sheet= "S3 HPPD", skip=2)


#population density data
shp3<-"data/eth_weredas.shp"
podens_district<-st_read(shp3)
```

```
## Reading layer `eth_weredas' from data source
##   `C:\Users\Davide\Desktop\Alma Mater\SECOND YEAR\GIS\Assignment\data\eth_weredas.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 466 features and 19 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: -161882.6 ymin: 376388.1 xmax: 1495283 ymax: 1645936
## CRS:           NA
```

Filter the data to focus only on Ethiopian datas

```r
sf.pwret<-sf.power %>%
  filter(country=="Ethiopia")
plantseth<-plantsafr2 %>%
  filter(country=='Ethiopia')
```

we use the function "st_zm()" to remove the dimension "Z(density)" from the geometries.

```r
sf.roads <- st_zm(sf.roads)
sf.pwret <- st_zm(sf.pwret)
```

we perform a log transformation of the population density data to allow for a better visualization.

```r
podens_district$log_POP_DENS <- log1p(podens_district$POP_DENS)
```

We assign a CRS (Coordinate Reference System) to the population density to insure compatibility between datasets. Therefore, we assigned the CRS corresponding to Ethiopia to ensure compatibility with the other components of the graph. Similarly, it is necessary to transform the power plant dataframe into an sf object and then set the appropriate CRS.

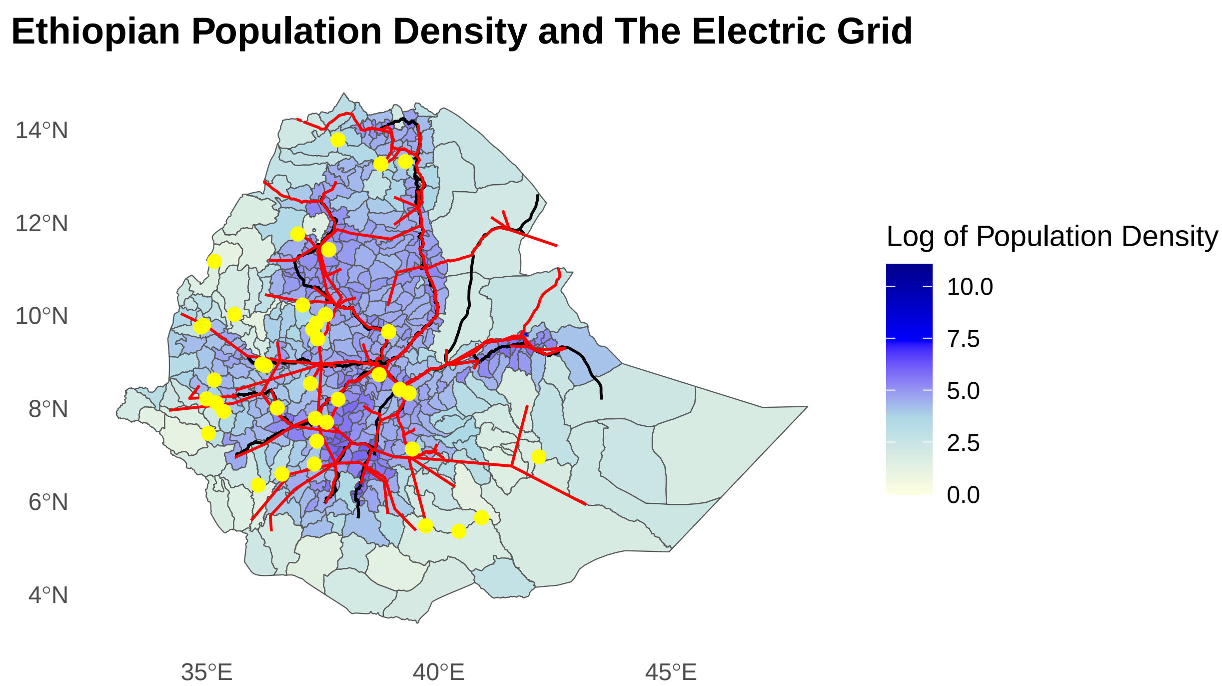
```r
podens_district <- st_set_crs(podens_district, 32637) # use the appropriate EPSG code
plantseth_sf <- st_as_sf(plantseth, coords = c("lon", "lat"), crs = 4326)
plantseth_sf <- st_transform(plantseth_sf, 32637)
st_crs(plantseth_sf)
```

```
## Coordinate Reference System:
##   User input: EPSG:32637
##   wkt:
## PROJCRS["WGS 84 / UTM zone 37N",
##     BASEGEOGCRS["WGS 84",
##         ENSEMBLE["World Geodetic System 1984 ensemble",
##             MEMBER["World Geodetic System 1984 (Transit)"],
##             MEMBER["World Geodetic System 1984 (G730)"],
##             MEMBER["World Geodetic System 1984 (G873)"],
##             MEMBER["World Geodetic System 1984 (G1150)"],
##             MEMBER["World Geodetic System 1984 (G1674)"],
##             MEMBER["World Geodetic System 1984 (G1762)"],
##             MEMBER["World Geodetic System 1984 (G2139)"],
##             ELLIPSOID["WGS 84",6378137,298.257223563,
##                 LENGTHUNIT["metre",1]],
##             ENSEMBLEACCURACY[2.0]],
##         PRIMEM["Greenwich",0,
```

```
##             ANGLEUNIT["degree",0.0174532925199433]],
##         ID["EPSG",4326]],
##     CONVERSION["UTM zone 37N",
##         METHOD["Transverse Mercator",
##             ID["EPSG",9807]],
##         PARAMETER["Latitude of natural origin",0,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8801]],
##         PARAMETER["Longitude of natural origin",39,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8802]],
##         PARAMETER["Scale factor at natural origin",0.9996,
##             SCALEUNIT["unity",1],
##             ID["EPSG",8805]],
##         PARAMETER["False easting",500000,
##             LENGTHUNIT["metre",1],
##             ID["EPSG",8806]],
##         PARAMETER["False northing",0,
##             LENGTHUNIT["metre",1],
##             ID["EPSG",8807]]],
##     CS[Cartesian,2],
##         AXIS["(E)",east,
##             ORDER[1],
##             LENGTHUNIT["metre",1]],
##         AXIS["(N)",north,
##             ORDER[2],
##             LENGTHUNIT["metre",1]],
##     USAGE[
##         SCOPE["Navigation and medium accuracy spatial referencing."],
##         AREA["Between 36°E and 42°E, northern hemisphere between equator and 84°N, onshore and offsh
##         BBOX[0,36,84,42]],
##     ID["EPSG",32637]]
```

Plot the graph. In the graph, the yellow point represents the Ethiopian power plants, the red lines indicate the high-voltage grid, and the black lines depict the major roads of Ethiopia.

```r
ggplot() +
  geom_sf(data = podens_district, aes(fill = log_POP_DENS)) +
  scale_fill_gradientn(colors = c("lightyellow", "lightblue", "blue", "darkblue"),
                       name = "Log of Population Density") +
  geom_sf(data = sf.roads, color = "black", size = 20) +
  geom_sf(data = sf.pwret, color = "red", size = 15) +
  geom_sf(data = plantseth_sf, color = 'yellow', size = 2) +
  labs(
    title = "Ethiopian Population Density and The Electric Grid",
  ) +
  theme_minimal() +
  theme(panel.grid = element_blank(),
        panel.background = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        plot.subtitle = element_text(hjust = 0.5, size = 10))
```



Clean the enviorment for the execution of the next task

```
rm(list=ls())
```

VIETNAM *Roads infrastruture*

Paper: Balboni, C.A., 2019. In harm's way? *infrastructure investments and the persistence of coastal cities*

Aim: Replicate Figure 2 from the paper, which illustrates *Vietnam's road infrastructure by road type*

attenzione= the dataset we used is not the same of the author of the paper. This justifies the missing time series Vietnam's observations of roads and also the difference the resulted graph. However, we decided to present also a replication of this paper's plot, even if it is not accurate enough.

Import data set on Vietam's roads

```
Vietnam_roads <-"data/vnm_rdsl_2015_OSM.shp"
sf.Vietnam_roads <- st_read(Vietnam_roads)
```

```
## Reading layer 'vnm_rdsl_2015_OSM' from data source
##   'C:\Users\Davide\Desktop\Alma Mater\SECOND YEAR\GIS\Assignment\data\vnm_rdsl_2015_OSM.shp'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 118138 features and 5 fields
## Geometry type: MULTILINESTRING
## Dimension:     XY
## Bounding box:  xmin: 102.1562 ymin: 8.596338 xmax: 109.4536 ymax: 23.37756
## Geodetic CRS:  WGS 84
```

Aggregate road type in 5 categories: freeway, dual carriageway, major roads, minor roads and other roads.

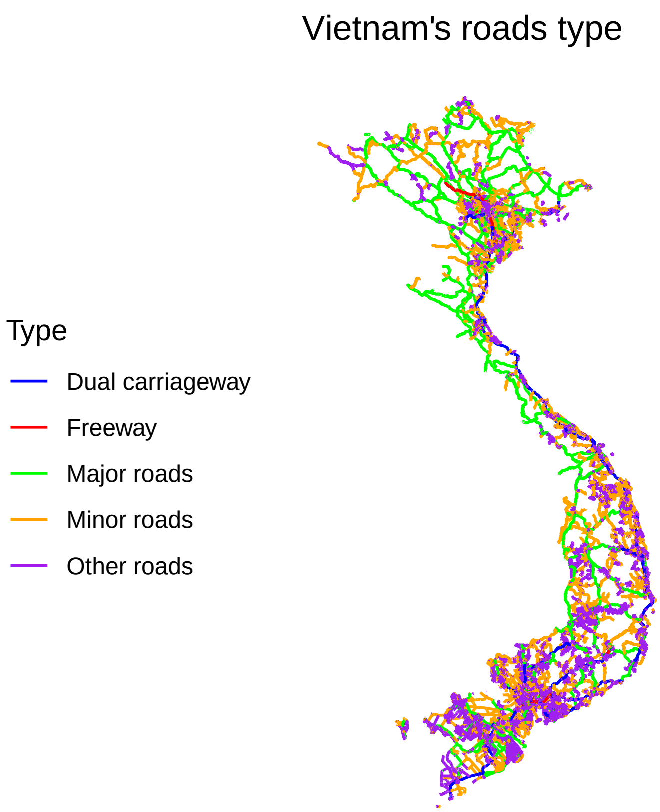
```r
sf.Vietnam_roads <- sf.Vietnam_roads %>%
  mutate(road_category = case_when(
    type %in% c("motorway", "motorway_link") ~ "Freeway",
    type %in% c("trunk", "trunk_link") ~ "Dual carriageway",
    type %in% c("primary", "primary_link") ~ "Major roads",
    type %in% c("secondary", "secondary_link", "tertiary", "tertiary_link") ~ "Minor roads",
    TRUE ~ "Other roads"
  ))
```

Plot the graph

```r
sf.Vietnam_roads$road_category <- as.factor(sf.Vietnam_roads$road_category)
aes(color = road_category, size = ifelse(road_category == "Freeway", 2, 1))
```

```
## Aesthetic mapping:
## * `colour` -> `road_category`
## * `size`   -> `ifelse(road_category == "Freeway", 2, 1)`
```

```r
ggplot() +
  geom_sf(data = sf.Vietnam_roads, aes(color = road_category)) +
  scale_color_manual(values = c("Freeway" = "red",
                                "Dual carriageway" = "blue",
                                "Major roads" = "green",
                                "Minor roads" = "orange",
                                "Other roads" = "purple")) +
  labs(title = "Vietnam's roads type",
       color = "Type",
       size = "Relevance") +
  theme_minimal() +
  theme(
    panel.grid = element_blank(),
    legend.position = "left",
    axis.title = element_blank(),
    axis.text = element_blank(),
    axis.ticks = element_blank()
  )
```



clean the enviorment

```
rm(list=ls())
```

OVERLAPPING *overlap between Mexico and Europe*

Aim: create a plot that overlays the Mexican shapefile onto the European shapefile (excluding Russia and Overseas France if necessary)

import "world" dataset

```
world<- world
```

The data set propose a division for Frence: 'France' e 'French Southern and Antarctic Lands'. Acceding to the geom of Frence we aknowledge that French Polynesia is considered in this observation.

```
sf.France <- world %>%
  filter(name_long %in% 'France')
```

To exclude this overses land we implemented the following procedure: Isolate Frence, extract the list in geometry involving polygons of Polynesia and eliminate it. Could be reasonable to move it toward French Southern and Antarctic Lands.

```r
France <- world[world$name_long == "France",]

geom_France <- st_geometry(France)[[1]]
geom_France <- geom_France[-1]

st_geometry(France) <- st_sfc(st_multipolygon(geom_France), crs = st_crs(world))
world[world$name_long == "France", ] <- France  # add to the original dataset
```

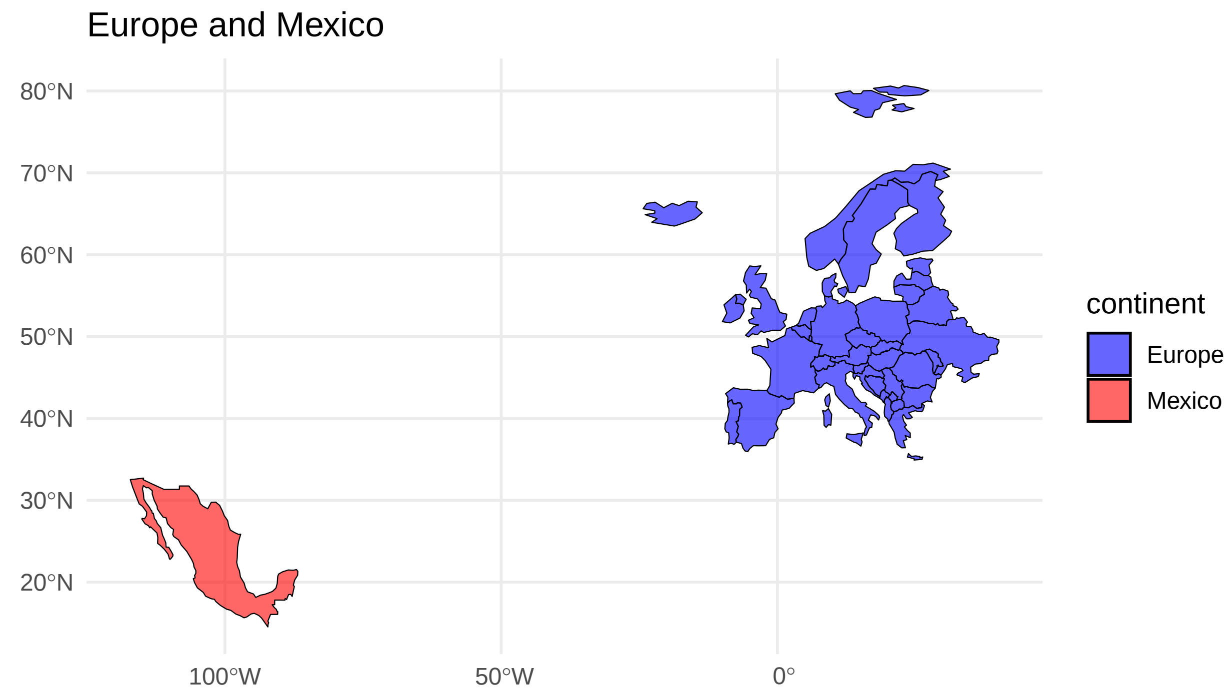
We are now left overlapping Mexico and Europe. The first step is to select just observations we are interested to excluding Russian Federation and Polynesia from Europe.

```r
sf.EU <- world %>%
  filter(continent == 'Europe' & name_long != 'Russian Federation')

sf.Mexico <- world %>%
  filter(name_long == 'Mexico')
```

Graph with no manual manipulations.

```r
# Creazione del grafico
ggplot() +
  geom_sf(data = sf.EU, aes(fill = continent), color = "black", alpha = 0.6) +
  geom_sf(data = sf.Mexico, aes(fill = "Mexico"), color = "black",
          inherit.aes = FALSE, alpha = 0.6) +
  scale_fill_manual(values = c("Europe" = "blue"  , "Mexico" = "red")) +
  theme_minimal() +
  labs(title = "Europe and Mexico")
```



Now we verify CRS for Europe and Mexico to make sure that it is not missing. Then, we traslate the Mexico in order to overlap it wit Europe: we managed to do so by setting manually the shift towards East (+120) and North (+20) that needs to occur in order to overlap them.

```
st_crs(sf.EU)
```

```
## Coordinate Reference System:
##   User input: EPSG:4326
##   wkt:
## GEOGCRS["WGS 84",
##     DATUM["World Geodetic System 1984",
##         ELLIPSOID["WGS 84",6378137,298.257223563,
##             LENGTHUNIT["metre",1]]],
##     PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433]],
##     CS[ellipsoidal,2],
##         AXIS["geodetic latitude (Lat)",north,
##             ORDER[1],
##             ANGLEUNIT["degree",0.0174532925199433]],
##         AXIS["geodetic longitude (Lon)",east,
##             ORDER[2],
##             ANGLEUNIT["degree",0.0174532925199433]],
##     USAGE[
##         SCOPE["Horizontal component of 3D system."],
##         AREA["World."],
##         BBOX[-90,-180,90,180]],
##     ID["EPSG",4326]]
```

```r
st_crs(sf.Mexico)
```

```
## Coordinate Reference System:
##   User input: EPSG:4326
##   wkt:
## GEOGCRS["WGS 84",
##     DATUM["World Geodetic System 1984",
##         ELLIPSOID["WGS 84",6378137,298.257223563,
##             LENGTHUNIT["metre",1]]],
##     PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433]],
##     CS[ellipsoidal,2],
##         AXIS["geodetic latitude (Lat)",north,
##             ORDER[1],
##             ANGLEUNIT["degree",0.0174532925199433]],
##         AXIS["geodetic longitude (Lon)",east,
##             ORDER[2],
##             ANGLEUNIT["degree",0.0174532925199433]],
##     USAGE[
##         SCOPE["Horizontal component of 3D system."],
##         AREA["World."],
##         BBOX[-90,-180,90,180]],
##     ID["EPSG",4326]]
```

```r
if (is.na(st_crs(sf.EU))) {
  st_crs(sf.EU) <- 4326
}

if (is.na(st_crs(sf.Mexico))) {
  st_crs(sf.Mexico) <- 4326
}

sf.EU <- st_transform(sf.EU, crs = 4326)
sf.Mexico <- st_transform(sf.Mexico, crs = 4326)
shift_geom <- function(geom, x_shift, y_shift) {
  st_geometry(geom) <- st_geometry(geom) + c(x_shift, y_shift)
  return(geom)
}
sf.Mexico_shifted <- shift_geom(sf.Mexico, x_shift = 120, y_shift = 20)
```

Mexico will now be shifted. Hence, we need to append to this new geometry the correct coordinate reference system compatible with Europe.

```r
st_crs(sf.Mexico_shifted)
```

```
## Coordinate Reference System: NA
```

```r
if (is.na(st_crs(sf.Mexico_shifted))) {
  st_crs(sf.Mexico_shifted) <- 4326
}
st_crs(sf.Mexico_shifted)
```

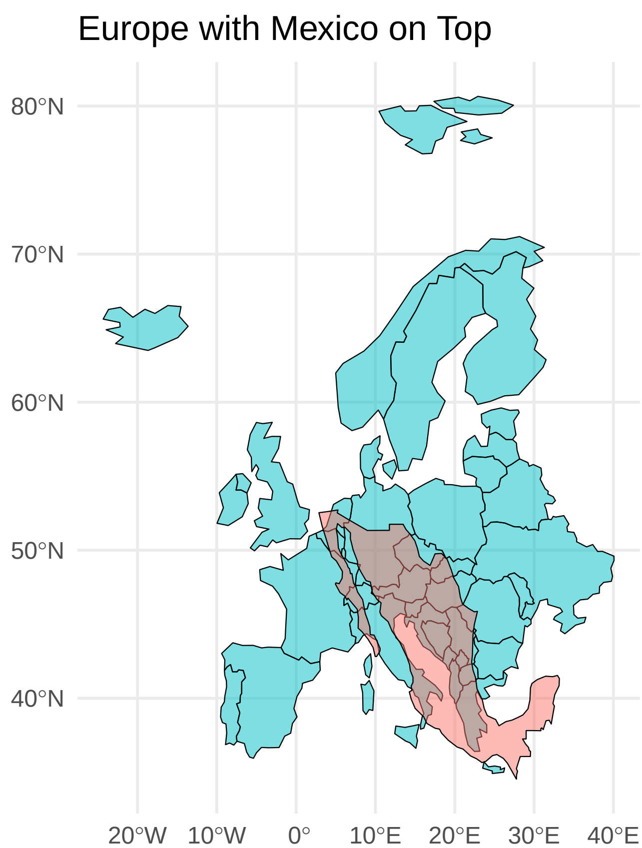
```
## Coordinate Reference System:
##   User input: EPSG:4326
##   wkt:
## GEOGCRS["WGS 84",
##     ENSEMBLE["World Geodetic System 1984 ensemble",
##         MEMBER["World Geodetic System 1984 (Transit)"],
##         MEMBER["World Geodetic System 1984 (G730)"],
##         MEMBER["World Geodetic System 1984 (G873)"],
##         MEMBER["World Geodetic System 1984 (G1150)"],
##         MEMBER["World Geodetic System 1984 (G1674)"],
##         MEMBER["World Geodetic System 1984 (G1762)"],
##         MEMBER["World Geodetic System 1984 (G2139)"],
##         ELLIPSOID["WGS 84",6378137,298.257223563,
##             LENGTHUNIT["metre",1]],
##         ENSEMBLEACCURACY[2.0]],
##     PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433]],
##     CS[ellipsoidal,2],
##         AXIS["geodetic latitude (Lat)",north,
##             ORDER[1],
##             ANGLEUNIT["degree",0.0174532925199433]],
##         AXIS["geodetic longitude (Lon)",east,
##             ORDER[2],
##             ANGLEUNIT["degree",0.0174532925199433]],
##     USAGE[
##         SCOPE["Horizontal component of 3D system."],
##         AREA["World."],
##         BBOX[-90,-180,90,180]],
##     ID["EPSG",4326]]
```

Now overlapping the 2 figures is possible according to the given coordinate reference system.

```
sf.Mexico_shifted
```

```
## Simple feature collection with 1 feature and 10 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 2.87224 ymin: 34.53883 xmax: 33.18802 ymax: 52.72083
## Geodetic CRS:  WGS 84
## # A tibble: 1 x 11
##   iso_a2 name_long continent   region_un subregion type  area_km2    pop lifeExp
## * <chr>  <chr>     <chr>       <chr>     <chr>     <chr>    <dbl>  <dbl>   <dbl>
## 1 MX     Mexico    North Amer~ Americas  Central ~ Sove~ 1969480. 1.24e8    76.8
## # i 2 more variables: gdpPercap <dbl>, geom <MULTIPOLYGON [°]>
```

```
ggplot() +
  geom_sf(data = sf.EU, aes(fill = region_un), color = "black", alpha = 0.5) +
  geom_sf(data = sf.Mexico_shifted, aes(fill = region_un), color = "black", alpha = 0.5) +
  theme_minimal() +
  labs(title = "Europe with Mexico on Top", fill = "Region") +
  theme(legend.position = "none") # hide the legend
```



Playing with coordinate refrence systems. (1st trial)

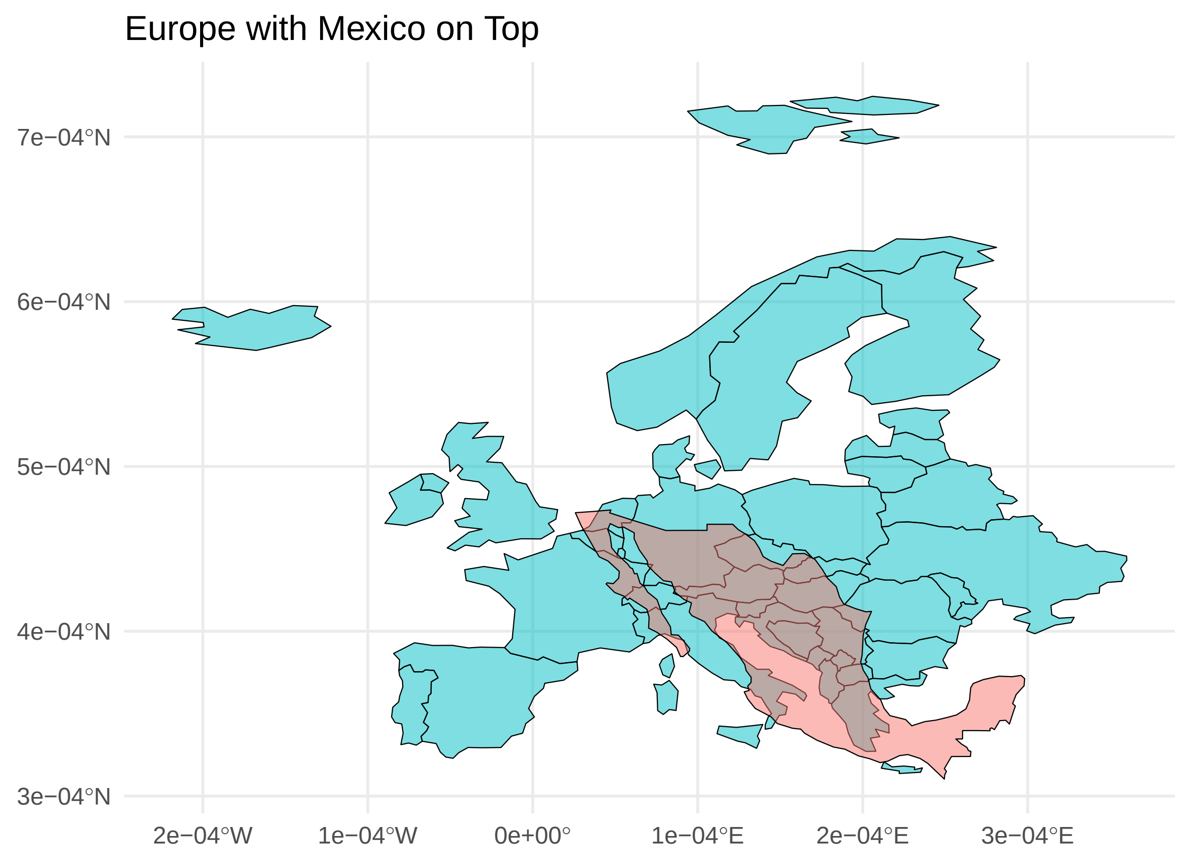
```r
st_crs(sf.Mexico_shifted)<- 3857
```

```
## Warning: st_crs<- : replacing crs does not reproject data; use st_transform for
## that
```

```r
st_crs(sf.EU)<- 3857
```

```
## Warning: st_crs<- : replacing crs does not reproject data; use st_transform for
## that
```

```r
ggplot() +
  geom_sf(data = sf.EU, aes(fill = region_un), color = "black", alpha = 0.5) +
  geom_sf(data = sf.Mexico_shifted, aes(fill = region_un),
          color = "black", alpha = 0.5) +
  theme_minimal() +
  labs(title = "Europe with Mexico on Top", fill = "Region") +
  theme(legend.position = "none")
```



Now overlapping the 2 figures is possible. (2nd trial)

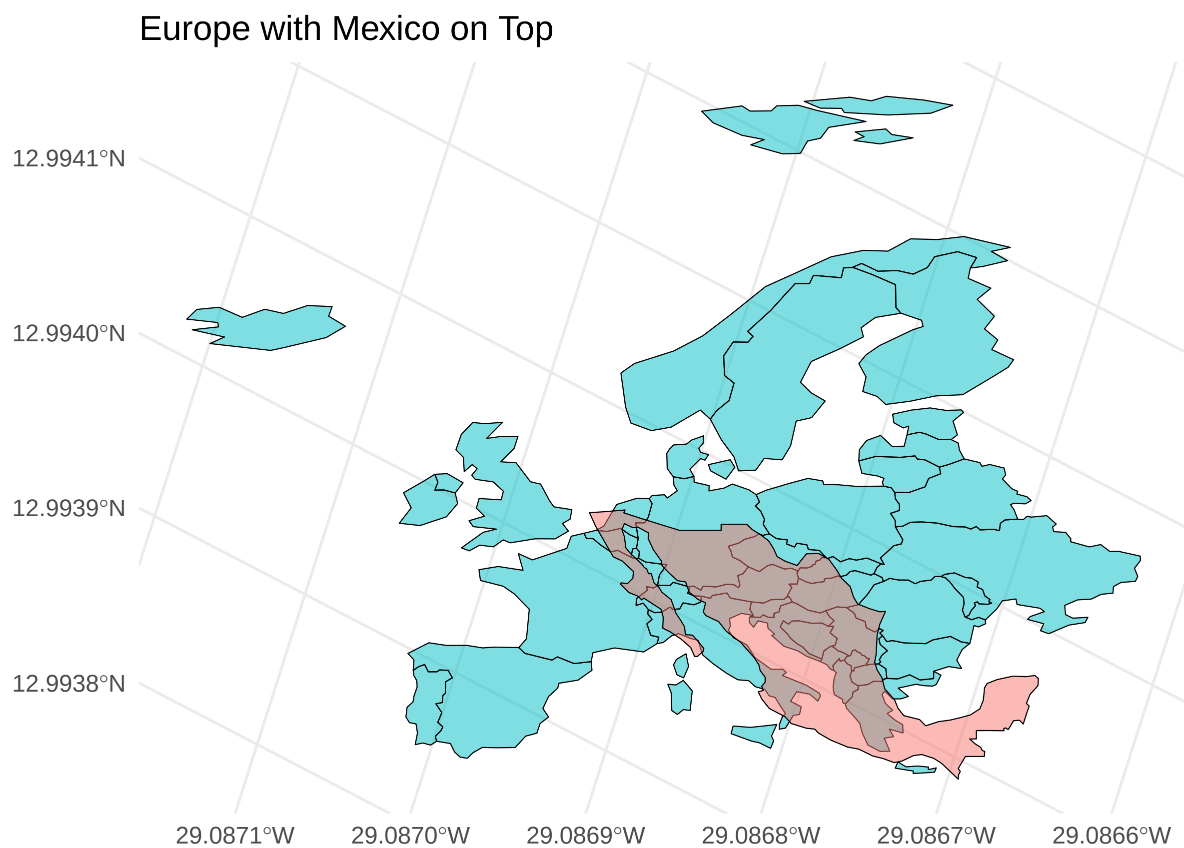
```
st_crs(sf.Mexico_shifted)<- 3035
```

```
## Warning: st_crs<- : replacing crs does not reproject data; use st_transform for
## that
```

```
st_crs(sf.EU)<- 3035
```

```
## Warning: st_crs<- : replacing crs does not reproject data; use st_transform for
## that
```

```
ggplot() +
  geom_sf(data = sf.EU, aes(fill = region_un), color = "black", alpha = 0.5) +
  geom_sf(data = sf.Mexico_shifted, aes(fill = region_un),
          color = "black", alpha = 0.5) +
  theme_minimal() +
  labs(title = "Europe with Mexico on Top", fill = "Region") +
  theme(legend.position = "none")
```



FINAL COMPARISON: it is clear that the maps we plotted with different levels of CRS do not differ significantly. Actuallt, when we plot Global Scale Representation, that means visualizing data at a global level, some projections may distort areas, though the effect may not be noticeable if the geometries are similar. Below we show the differences between the already seen CRS. CRS 4326 uses a geographic coordinate system (degrees) suitable for global datasets, while CRS 3857 uses a projected coordinate system that's ideal for web mapping but can distort large-scale maps, especially at high latitudes.

```
st_crs(sf.Mexico_shifted) <- 4326
```

```
## Warning: st_crs<- : replacing crs does not reproject data; use st_transform for
## that
```

```
st_crs(sf.EU) <- 4326
```

```
## Warning: st_crs<- : replacing crs does not reproject data; use st_transform for
## that
```

```
plot_4326 <- ggplot() +
  geom_sf(data = sf.EU, aes(fill = region_un), color = "black", alpha = 0.5) +
  geom_sf(data = sf.Mexico_shifted, aes(fill = region_un), color = "black", alpha = 0.5) +
  theme_minimal() +
  labs(subtitle = "CRS: 4326") +
  theme(legend.position = "none") +
  coord_sf(expand = FALSE)

st_crs(sf.Mexico_shifted) <- 3857
```

```
## Warning: st_crs<- : replacing crs does not reproject data; use st_transform for
## that
```

```r
st_crs(sf.EU) <- 3857
```

```
## Warning: st_crs<- : replacing crs does not reproject data; use st_transform for
## that
```

```r
plot_3857 <- ggplot() +
  geom_sf(data = sf.EU, aes(fill = region_un), color = "black", alpha = 0.5) +
  geom_sf(data = sf.Mexico_shifted, aes(fill = region_un), color = "black", alpha = 0.5) +
  theme_minimal() +
  labs(subtitle = "CRS: 3857") +
  theme(legend.position = "none") +
  coord_sf(expand = FALSE)

# Give a single title
final_plot <- annotate_figure(grid.arrange(plot_4326, plot_3857, ncol = 2),
                              top = text_grob("Comparison between different projections",
                                              face = "bold", size = 14))
plot(final_plot)
```