Data and Climate

Session 3 - Maps with R

Jean-Baptiste Guiffard

2024-01-10

Course structure

Sessions

Sessions	Topics
Session 1	The Basics of R / Manipulating dataset with the DPLYR package
Session 2	Graphic representations with GGPLOT
Session 3	Making maps with R
Session 4	Web scraping with R
Session 5	Extracting and analyzing textual data using R
Session 6	Produce documents with Rmarkdown...

Introduction: les données cartographiques

Quelques cartes issues des rapports du GIEC

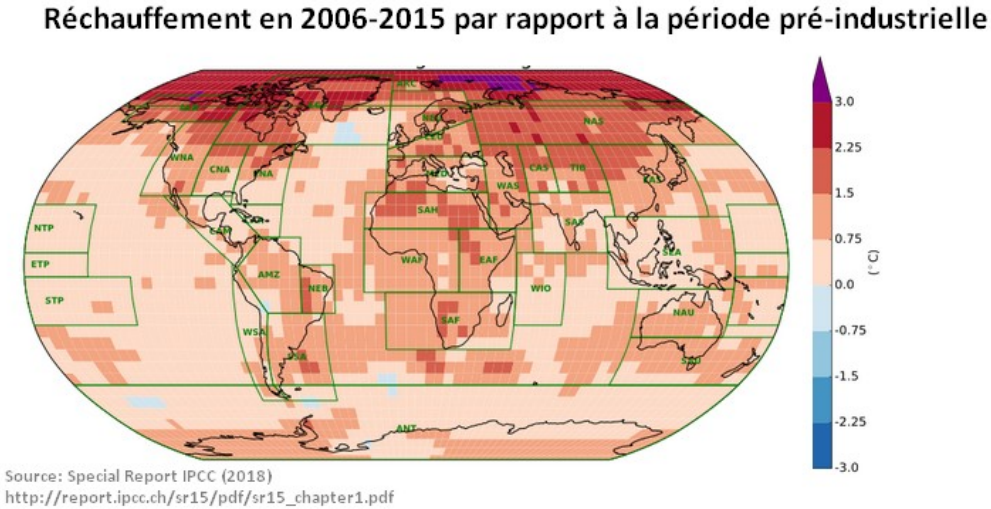


Figure 1: Augmentation de la température de surface multi-décennale moyenne par rapport aux niveaux pré-industriels

- Activités humaines → augmentation de la température moyenne de surface de la planète de 0.8 à 1.2°C depuis l'ère pré-industrielle (réchauffement de +0,2°C chaque décennie).
- Le GIEC estime qu'il est très vraisemblable que le nombre de jours et nuits froides a diminué et à l'inverse pour les journées chaudes à l'échelle mondiale. La fréquence des vagues de chaleur a augmenté en Europe, en Asie et en Australie avec un degré de confiance élevé.

Quelques cartes issues des rapports du GIEC

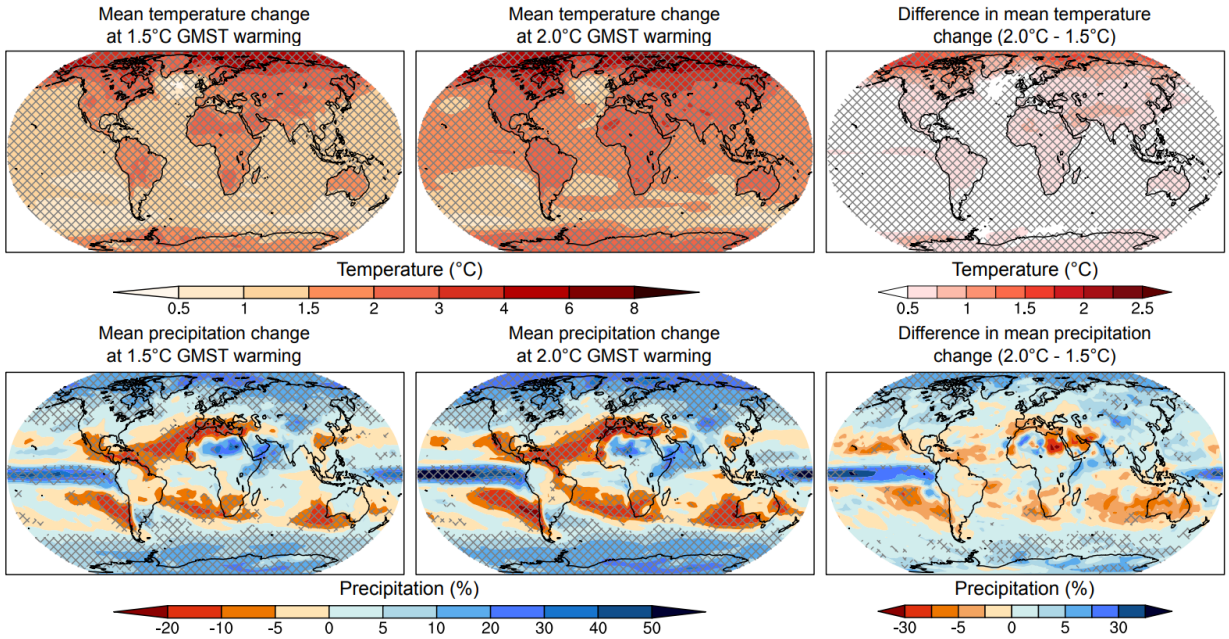


Figure 2: Changements projetés de la température moyenne (en haut) et de la réciprocité moyenne (en bas) pour un réchauffement planétaire de 1,5°C (à gauche) et de 2°C (au milieu) par rapport à la période préindustrielle (1861-1880), et différence entre 1,5°C et 2°C de réchauffement planétaire (à droite).

Dans les deux scénarios d'un réchauffement de 1.5°C ou 2°C: il est très probable que le réchauffement affectera plus les terres que les océans et le nombre de jours et nuits chaudes continuera d'augmenter. $\rightarrow$ Il est aussi probable que la fréquence des événements extrêmes de chaleur dans les régions tropicale augmentera.

Quelques cartes issues des rapports du GIEC

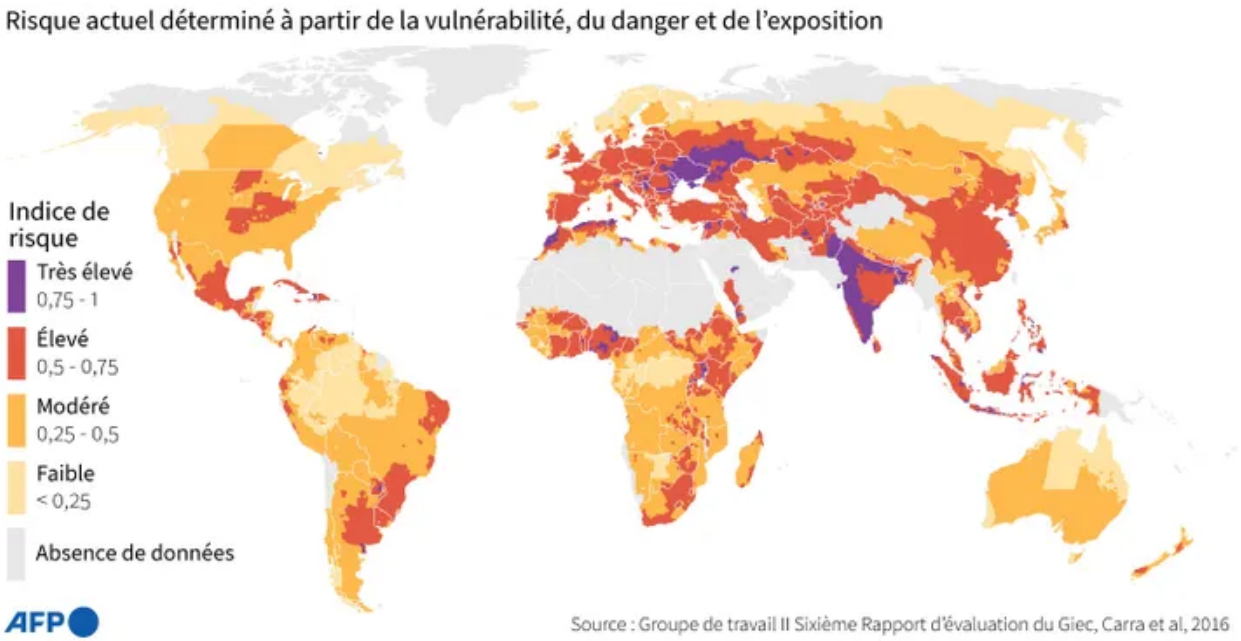


Figure 3: Risque de sécheresse dans le Monde

Les systèmes de coordonnés (Datum)

- Datum : Peut être utilisé localement (RGF93 = EPSG4171, France ou NAD 1983, USA) ou globalement (WGS84 = EPSG4326).
- EPSG = European Petroleum Survey Group, liste des systèmes de coordonnées géoéréférencées de projection.

Figure 4: Zone des datums WGS84 et RGF93 selon EPSG.io.

Projections

- Projection : Cylindrique (Pseudo-Mercator, EPSG3857) et conique (Lambert, EPSG:2154).

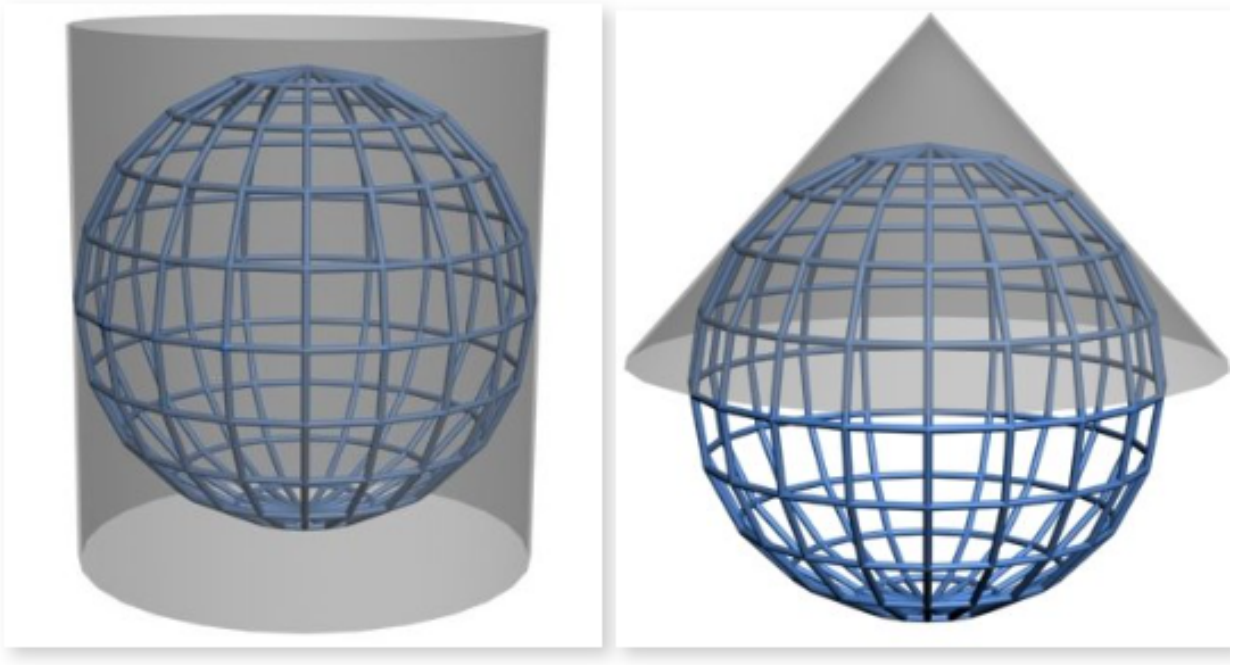
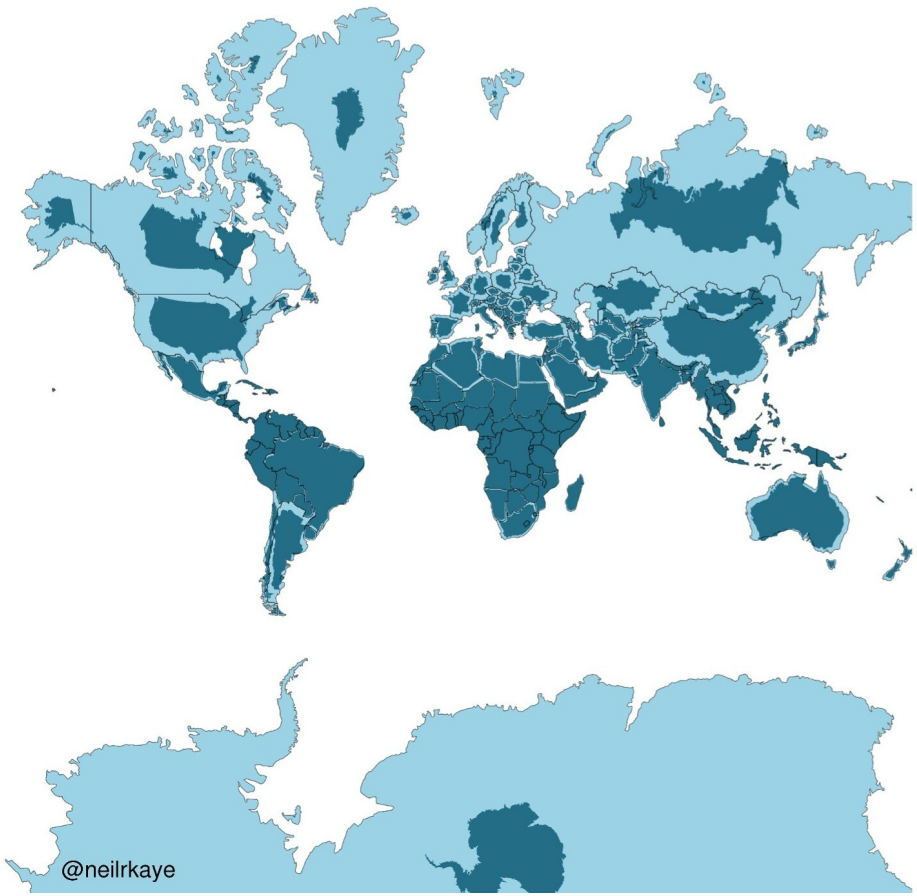


Figure 5: Représentation graphique des deux grands types de projection.

Comparaison de la projection Mercator avec la véritable taille des pays



Les systèmes de projection

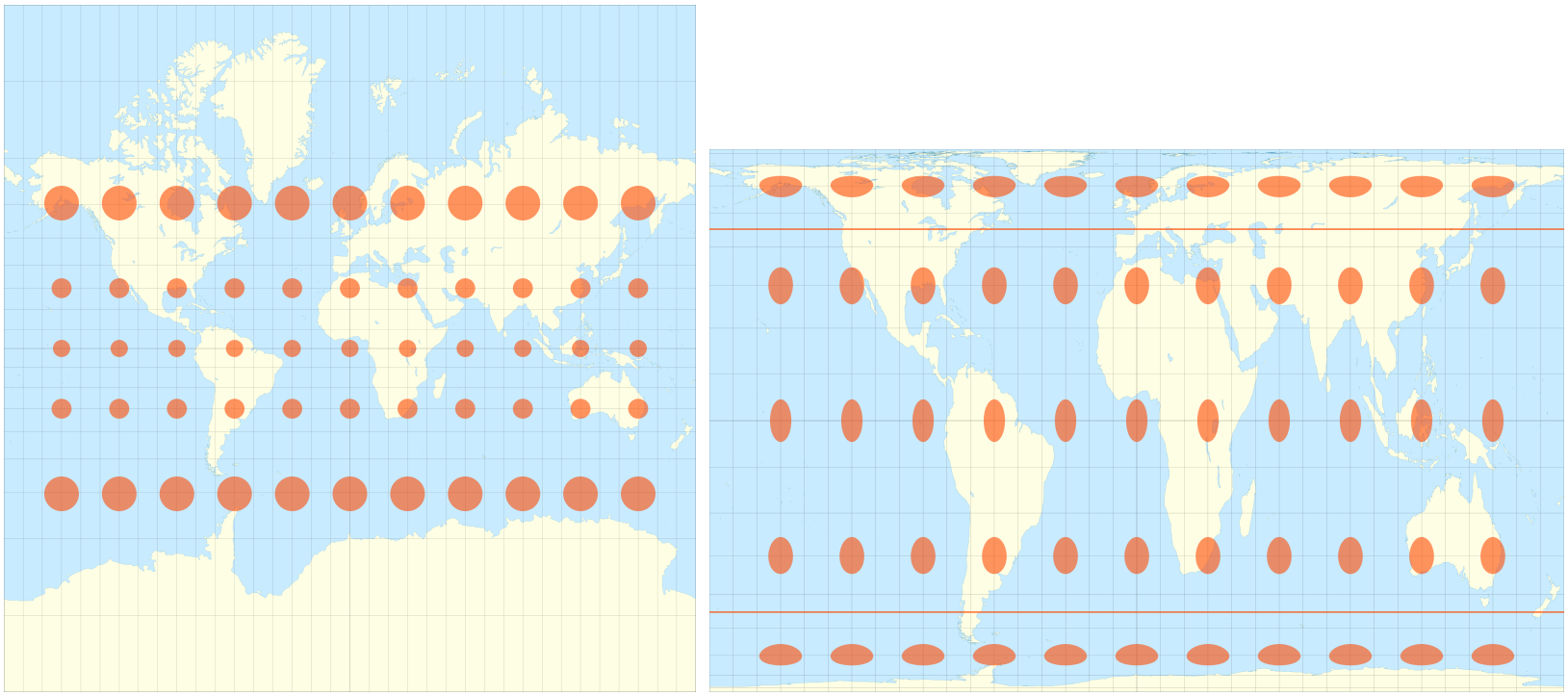


Figure 6: Projections cylindriques conforme de Mercator et équivalente de Gall-Peters.

Les projections cylindriques ou l'indicatrice de Tissot, du nom de son inventeur Nicolas Auguste Tissot, est une forme géométrique (un cercle ou une ellipse) qui permet d'évaluer le degré de déformation d'un système de projection cartographique.

Les systèmes de projection

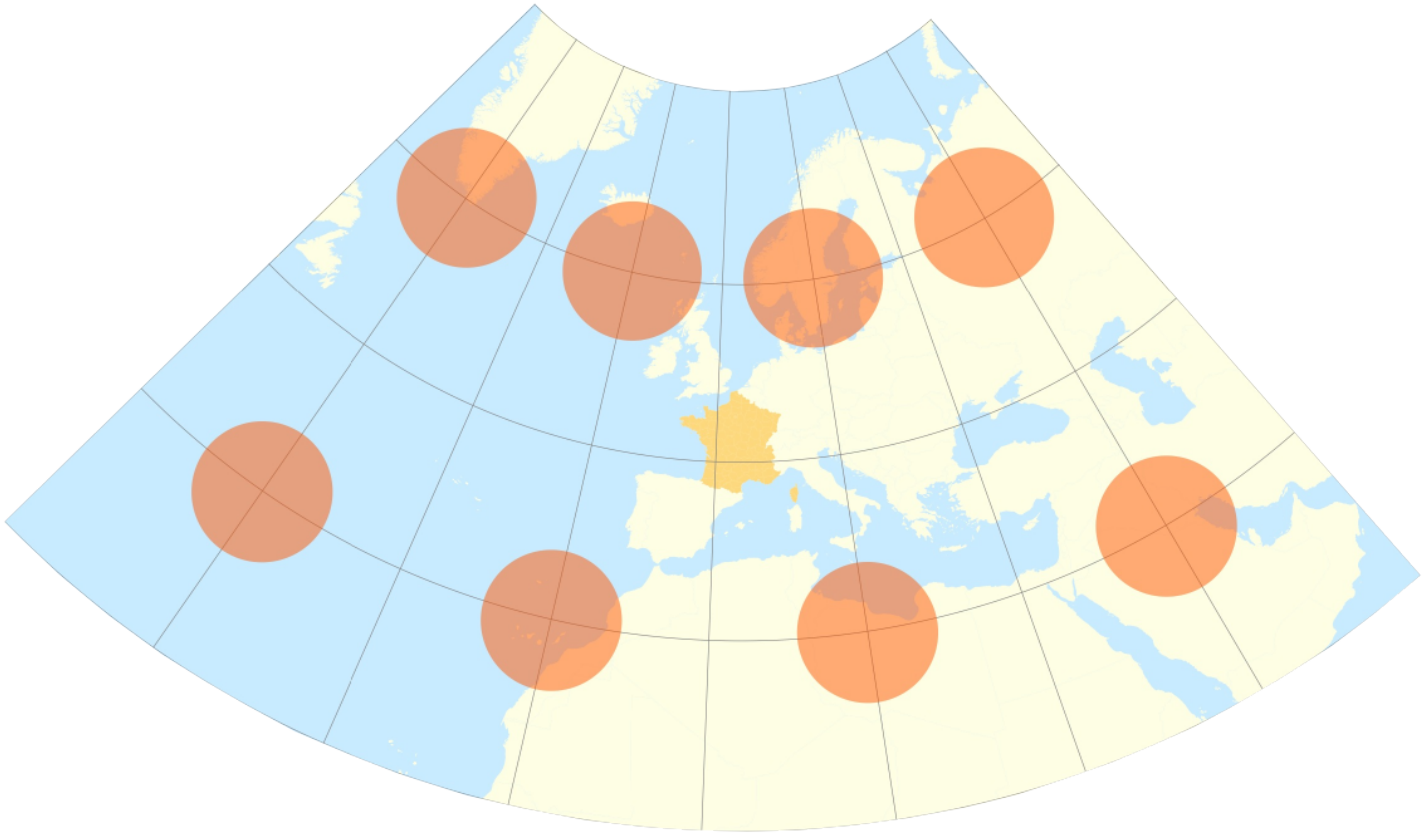


Figure 7: Projection conique conforme de Lambert centrée sur la France

Les types de données cartographiques

Les vecteurs

- Vectoriel : Shapefile (shp, dbf, shx), GeoPackage. . .

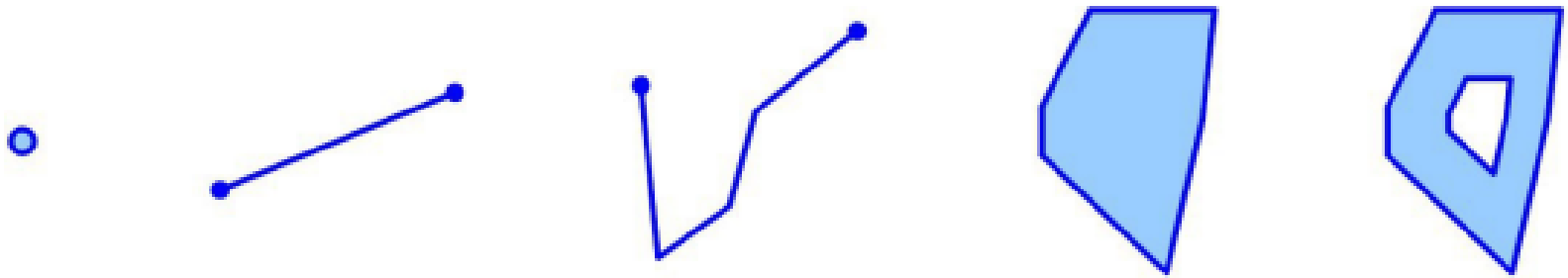


Figure 8: Forme des différents vecteurs.

Les Rasters

- Raster : GeoTIFF, GeoPackage...
- Un raster est simplement une image dont chaque pixel qui la compose représente une valeur.

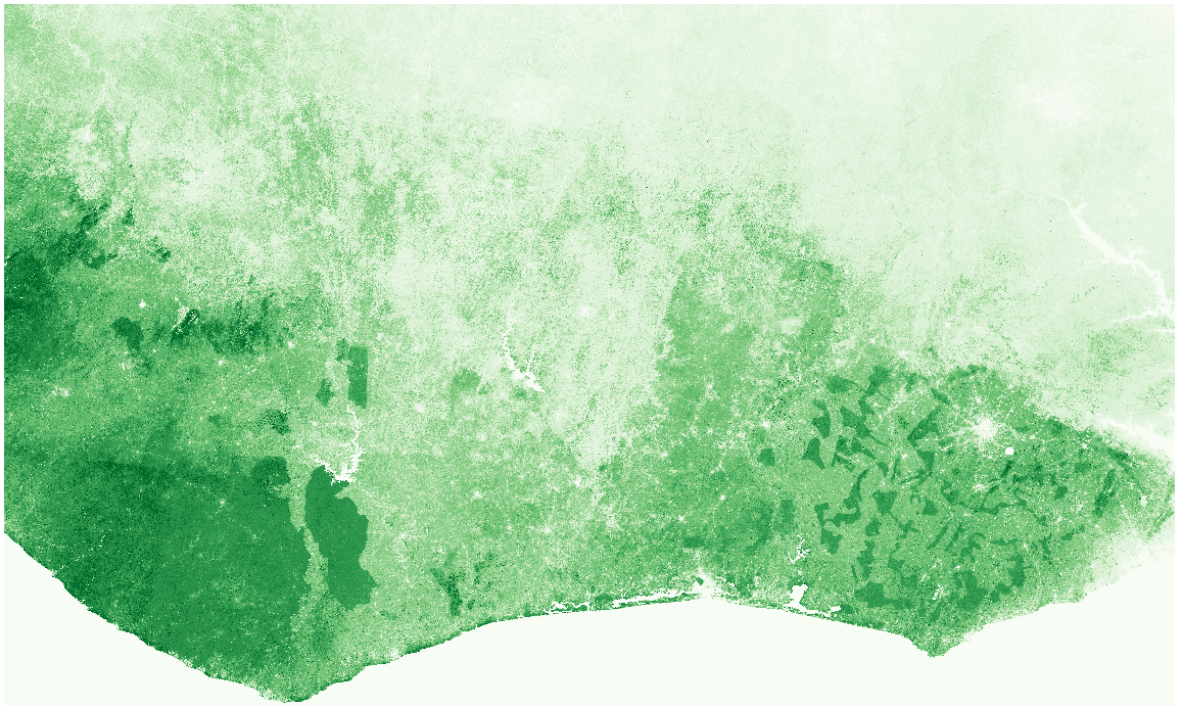


Figure 9: Exemple d'un raster

Logiciels de cartographie

- QGIS (gratuit, facile, point and click, compatible Python et R...)
- ArcGis (payant, similaire à QGIS, données exclusives...)
- R (gratuit, prise en main plus complexe, facilite le traitement des données...)
- Possibilité : combiner QGIS et R.

L'obtention de données cartographiques

- NASA, Socioeconomic data and application center : https://sedac.ciesin.columbia.edu/
- Global Administrative Areas : https://gadm.org/
- Earth Engine Data Catalog : https://developers.google.com/earth-engine/datasets
- DHS/Afrobarometer/enquêtes microéconomiques...

Traitement des données cartographiques avec le package sf

Introduction à sf

Un package qui date de 2016 publié par Edzer Pebesma qui vise à regrouper les fonctionnalités de trois packages plus anciens sur R (sp, rgeos et rgdal).

La forme d'un objet sf

C'est un data.frame avec une colonne spéciale nommée "geometry" qui contient les coordonnées d'un polygone.

YPE_1	NL_NAME_1	VARNAME_1	geometry
norate	NA	Al Aryiānah|L'Ariana|Tunis Ariana|Al Aryānah	list(list(c(10.2109699249268, 10.2109699249268, 10.2106895...
norate	NA	Bājah|Béja	list(list(c(9.08780288696289, 9.08878231048584, 9.09033107...
norate	NA	Bin `Arūs|Ben Arous|Tunis Ben Arous	list(list(c(10.2948608398438, 10.2948608398438, 10.2962512...
norate	NA	BanzartBanzart|Bensert|Binzart|Biserta|Bizerta	list(list(c(10.2334718704225, 10.2334718704225, 10.2337503...
norate	NA	Qābis|Gabās	list(list(c(9.67866897583008, 9.69477558135992, 9.69743347...

Figure 10: Exemple extrait objet sf

Les commandes les plus basiques

```
mydata.shp <- st_read('NOM_DATA_MAP.shp')
head(mydata.shp)
plot(mydata.shp)
plot(mydata.shp['var1'])

merge(mydata.shp, other_df, by="ID")
```

D'autres commandes intéressantes (I)

Connaître la projection et la modifier

```
st_crs(mydata.shp)
mydata.shp_reproj <- st_transform(mydata.shp, 2154) #Lambert-93
```

Obtenir les centroïdes des polygones

```
mydata_centroids.shp <- st_centroid(mydata.shp)
```

Calculer des distances

```
matrice_distance_centroids <- st_distance(x = mydata_centroids.shp,
                                          y = mydata_centroids.shp)
```

D'autres commandes intéressantes (II)

Unifier/agréger des polygones

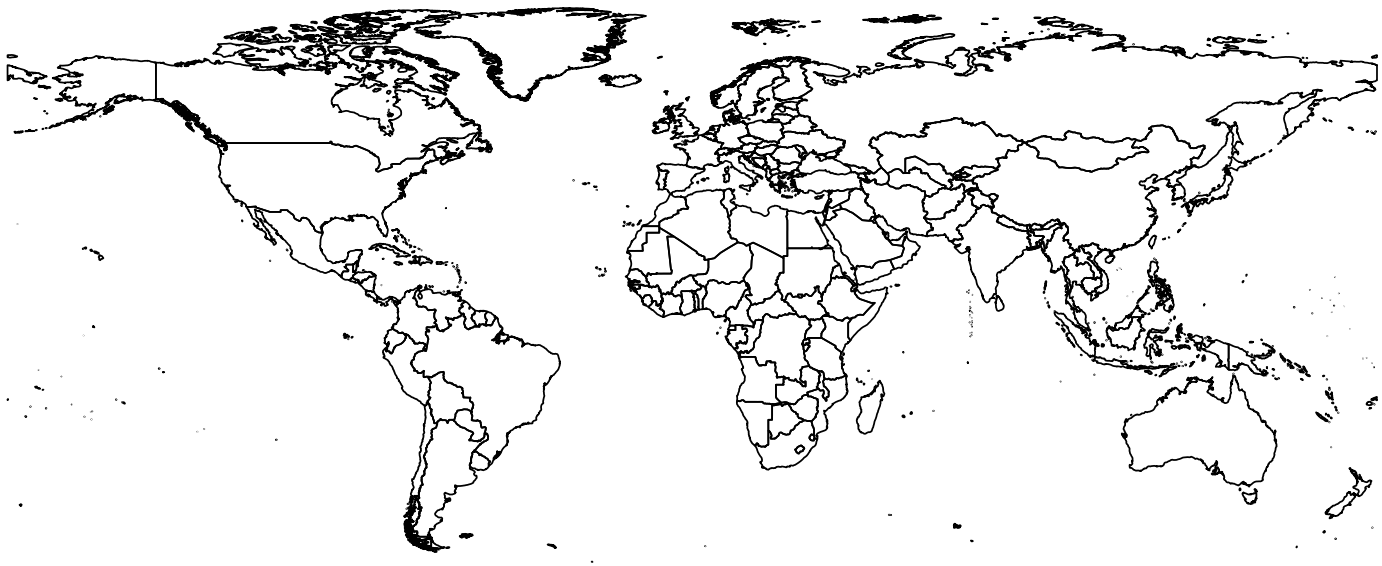
```
mydata_union.shp <- st_union(mydata.shp)
```

Créer un buffer ou zone tampon

```
mydata_buffer.shp <- st_buffer(x = mydata_union.shp,
                               dist = 1000)
```

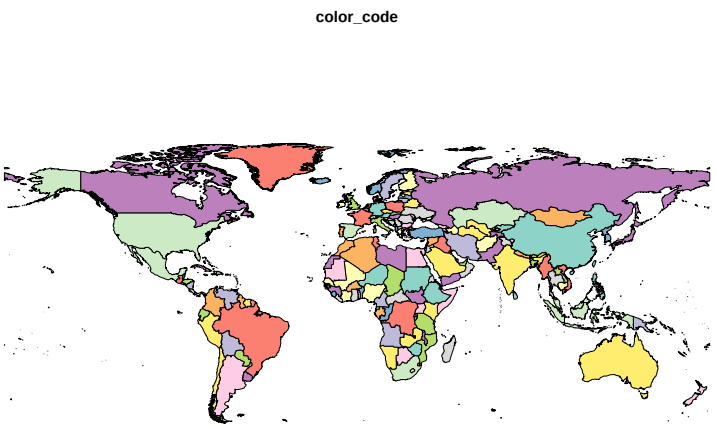
Quelques manipulations de données avec un .shp du Monde

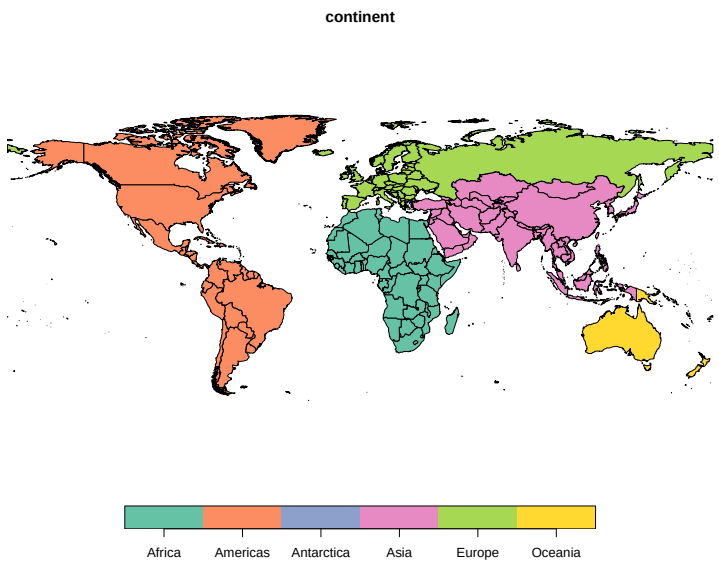
```
plot(st_geometry(world_map))
```



Représentation des cartes “pays” et “continents”

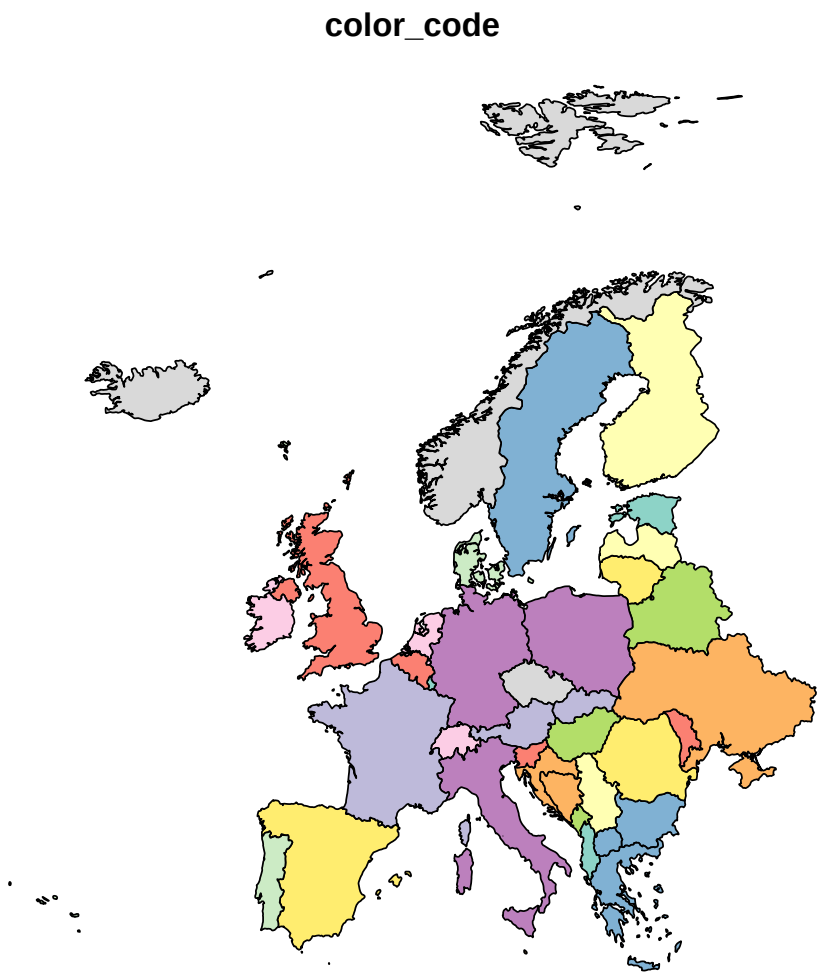
```
par(mar = c(4, 4, .1, .1))
plot(world_map['color_code'])
plot(world_map['continent'])
```



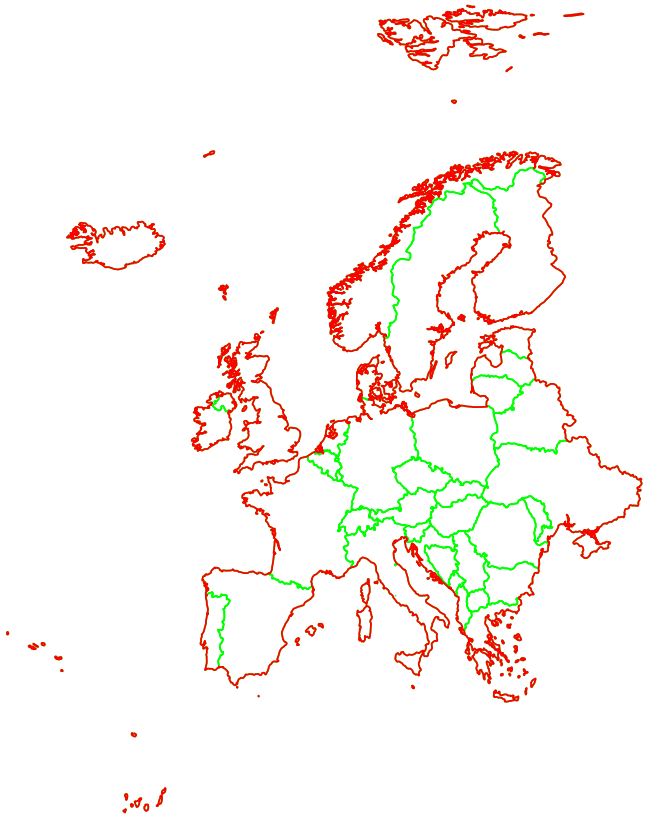


Sélection de quelques polygones

```
Europe_map <- subset(world_map, continent=="Europe" & color_code != "RUS" )
plot(Europe_map['color_code'])
```

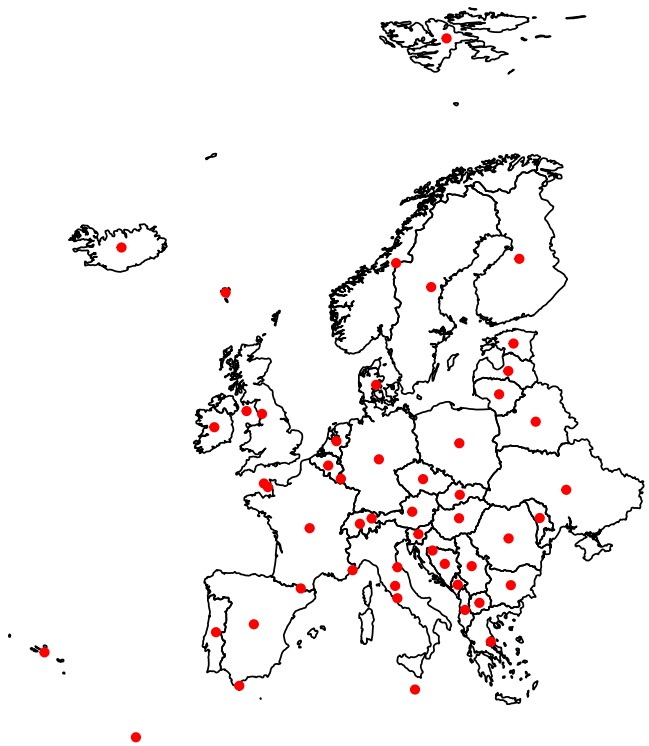


Obtenir les contours de l'Europe



Projection des centroids

```
Europe_centroids <- st_centroid(Europe_map)
plot(st_geometry(Europe_map))
plot(st_geometry(Europe_centroids), add=TRUE, cex=1, col="red", pch=20)
```



Matrice des distances entre les centroids

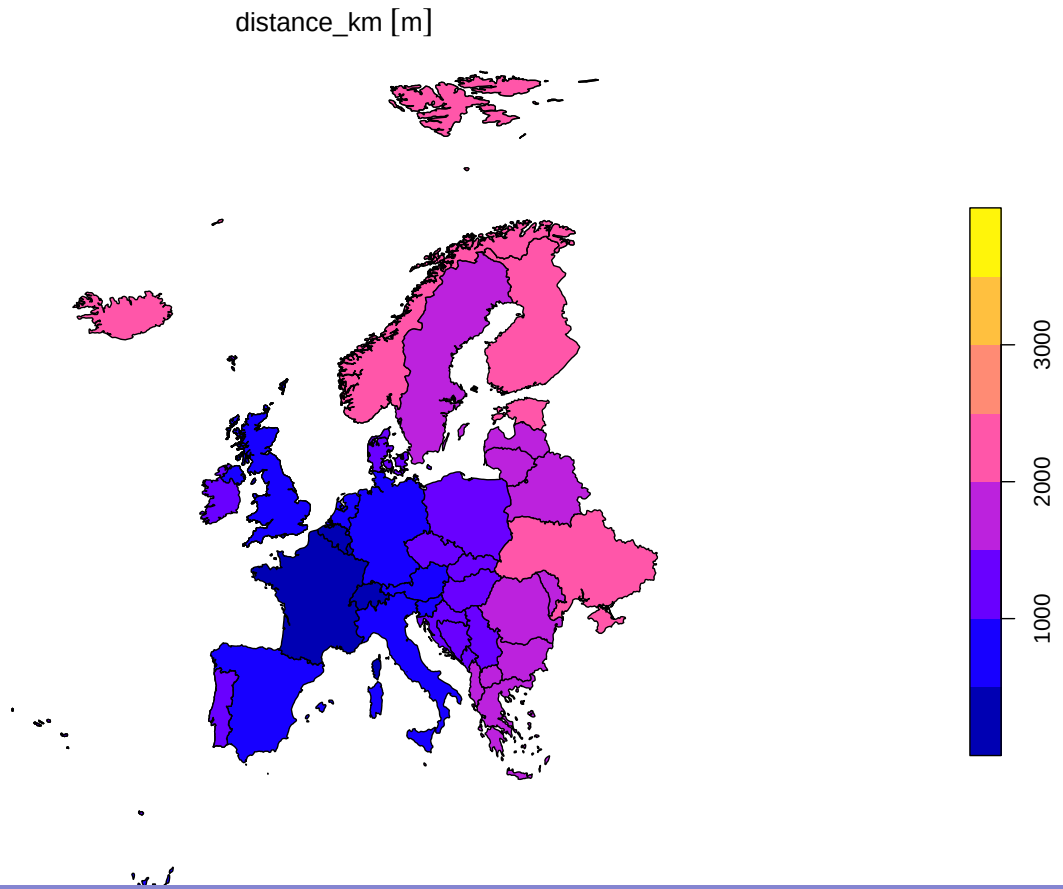
```
#install.packages("reshape2")
library(reshape2)
matrix_distance <- st_distance(Europe_centroids)
rownames(matrix_distance) <- Europe_centroids$color_code
colnames(matrix_distance) <- Europe_centroids$color_code

df_distance <- as.data.frame(as.table(matrix_distance))
df_distance$distance_km <- df_distance$Freq/1000
```

Carte des distances

```
df_distance_france <- subset(df_distance, Var1=='FRA')
Europe_map <- merge(Europe_map,df_distance_france, by.x="color_code", by.y="Var
```

```
plot(Europe_map['distance_km'])
```



Projeter des points sur une carte

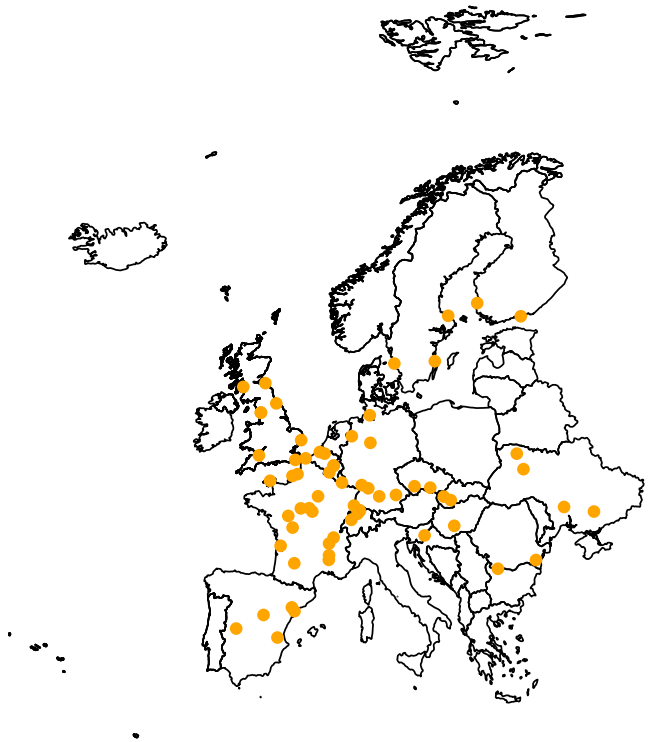
```
power_plants_points <- st_read('Global_Power_Plants/Power_Plants.shp')
```

```
## Reading layer ‘Power_Plants’ from data source
##   ‘C:\Users\jbguiffard\OneDrive - UniversitÃ© Paris 1 PanthÃ©on-Sorbonne\COU
##   using driver ‘ESRI Shapefile’
## Simple feature collection with 28664 features and 23 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: -179.9777 ymin: -77.847 xmax: 179.3887 ymax: 71.292
## Geodetic CRS:  WGS 84
```

```
power_plants_points_europe <- subset(power_plants_points,
                                     country %in% unique(Europe_map$iso3))
nuclear_pw_plants <- subset(power_plants_points_europe, fuel1 == "Nuclear")
nuclear_pw_plants.pts <- st_as_sf(nuclear_pw_plants,
                                  coords = c("longitude", "latitude"),
                                  crs=st_crs(Europe_map))
```

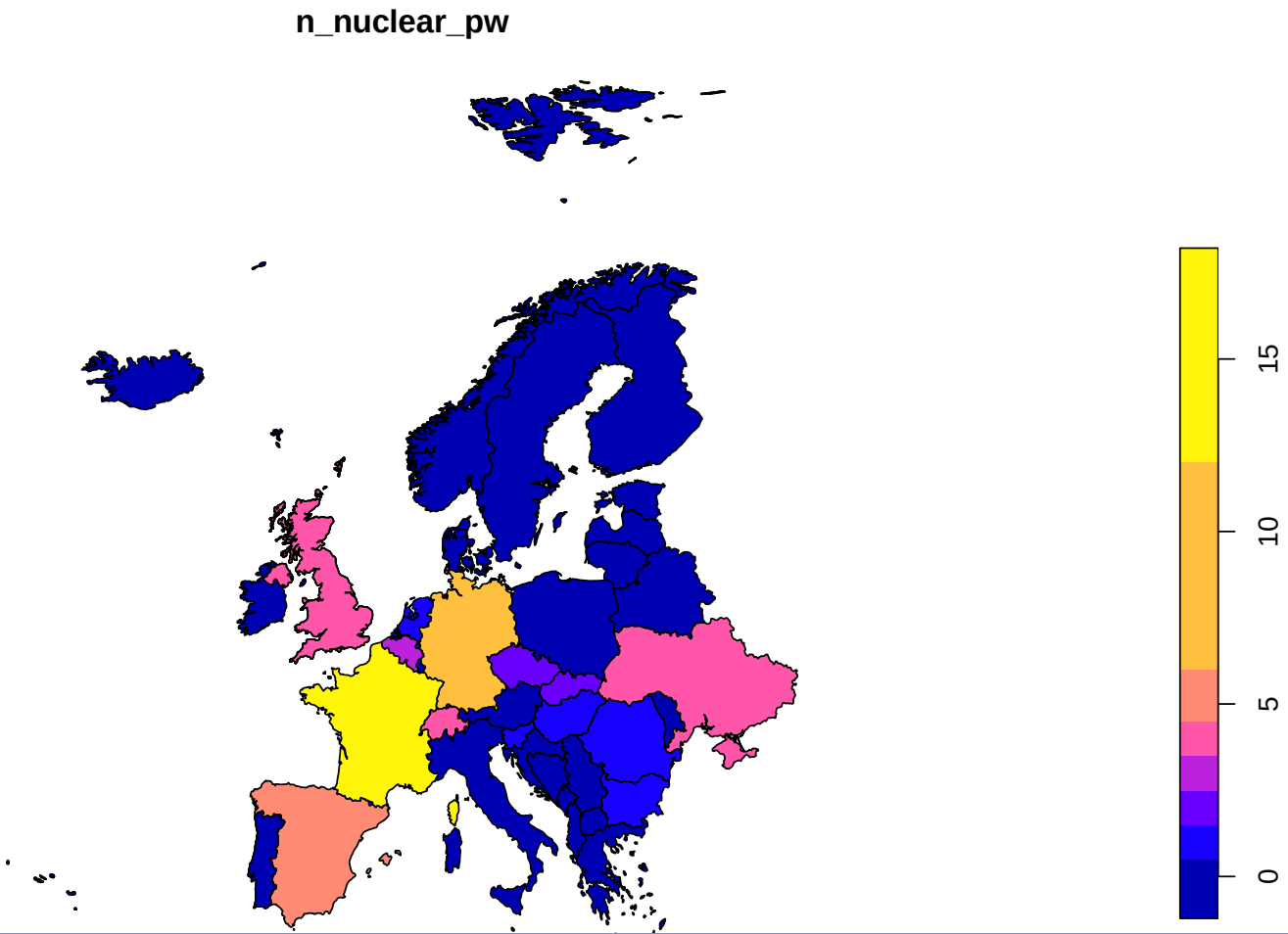
Localisation des centrales nucléaires en Europe.

```
plot(st_geometry(Europe_map))
plot(nuclear_pw_plants.pts,col="orange",cex=1,pch=16,add=T)
```

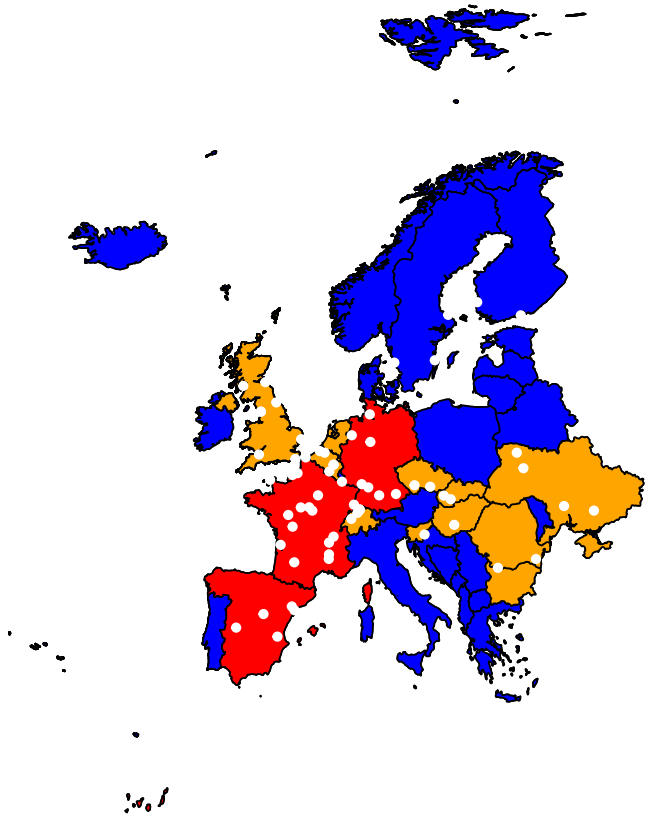


Calcul de la densité de points par aire géographique

```
n_pw_plts_country <- st_intersects(Europe_map, nuclear_pw_plants.pts)
Europe_map$n_nuclear_pw <- sapply(X = n_pw_plts_country, FUN = length)
plot(Europe_map['n_nuclear_pw'])
```



Calcul de la densité de points par aire géographique



Un package pour les cartes interactives (leaflet)

```
#install.packages('leaflet')
library(leaflet)
m <- leaflet() %>% addTiles()

m1 <- leaflet(data = power_plants_points_europe) %>% addTiles() %>%
  addMarkers(~longitude, ~latitude, popup = ~as.character(fuel1), label = ~as.ch
```

Notre première carte avec ggplot

Chargement des données

```
data_pollution <- read.csv2('DATA/owid-co2-data.csv', sep=",")
Metadata_Country <- read.csv2('DATA/Metadata_Country.csv', sep=",") %>%
  rename("Country_code" = "ï..Country.Code")
  #rename("Country_code" = "Country.Code")

data_pollution_num <- data_pollution %>%
  select(-c(country,iso_code))%>%
  mutate_if(is.character, as.numeric) %>%
  cbind(data_pollution[,c("country","iso_code")])


join_pollution_wb_data <- data_pollution_num %>%
  dplyr::inner_join(Metadata_Country, by = c("iso_code" = "Country_code"))

join_pollution_wb_data <- join_pollution_wb_data %>%
  filter(country != "") %>%
  filter(IncomeGroup !="")
```

Première carte avec ggplot2

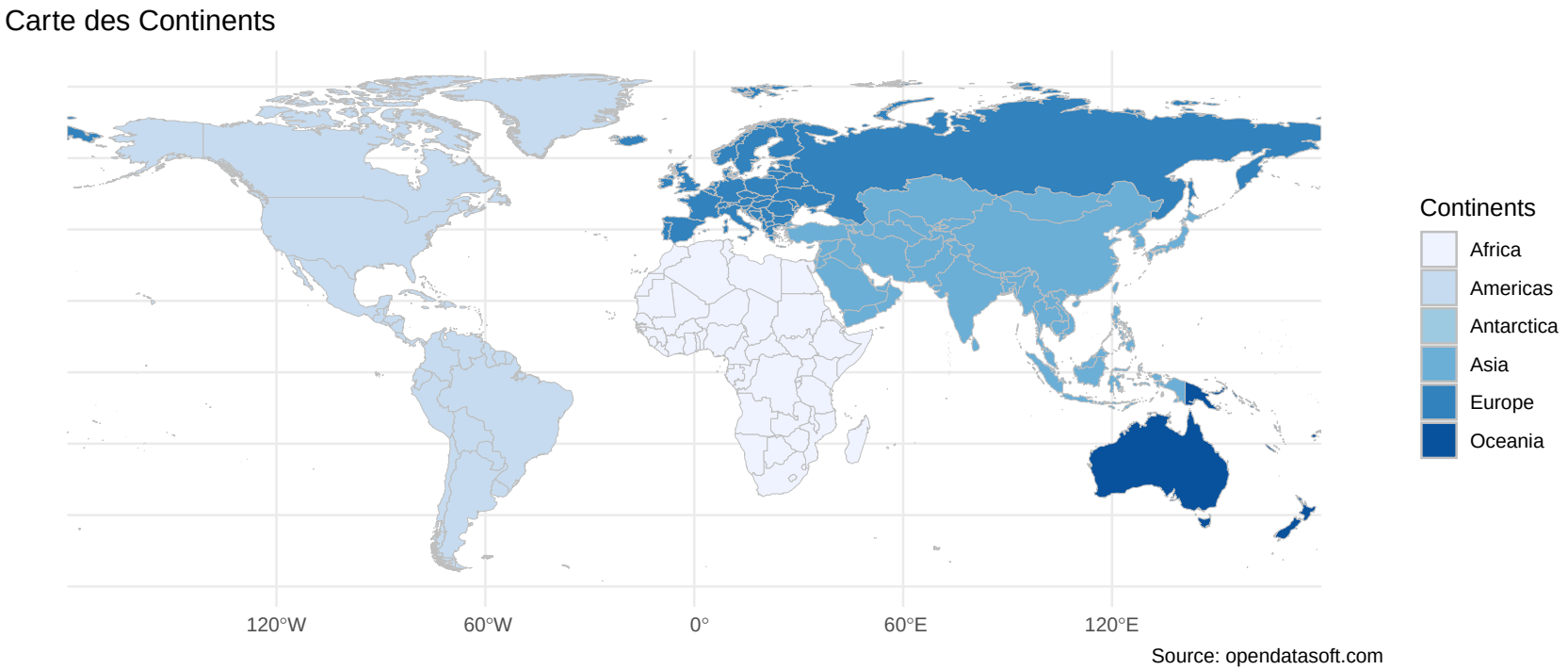
```
library(ggplot2)
```

```
## Warning: le package 'ggplot2' a été compilé avec la version R 4.1.3
```

```
map1 <- ggplot() +
  geom_sf(data = world_map, col="grey", aes(fill=continent), show.legend = TRUE
  theme_minimal() +
  theme(legend.position = "right")+
  scale_fill_brewer(name = "Continents", na.value = "grey") +
  labs(x = NULL,
       y = NULL,
       title = "Carte des Continents",
       caption = "Source: opendatasoft.com")
```

Première carte avec ggplot2

```
plot(map1)
```



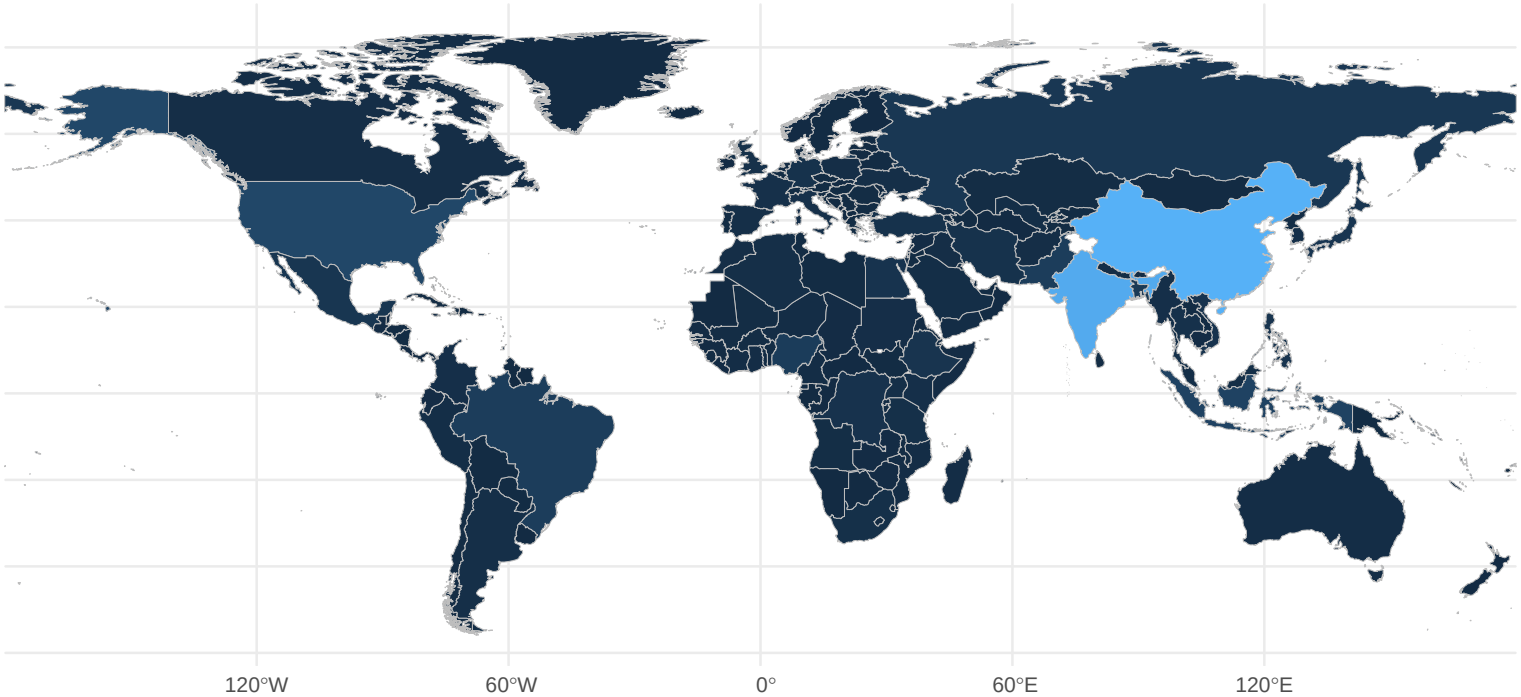
Merger avec la base “émissions de co2”

```
join_pollution_wb_data_2019 <- subset(join_pollution_wb_data, year==2019)
world_map_co2 <- world_map %>%
  right_join(join_pollution_wb_data_2019, by=c('iso3'='iso_code'))

map2 <- ggplot() +
  geom_sf(data = world_map_co2, col="grey", aes(fill=population), show.legend =
  theme_minimal() +
  theme(legend.position = "right")+
  #scale_fill_gradientn(colours = terrain.colors(100))+
  labs(x = NULL,
       y = NULL,
       title = "Carte de la Population mondiale",
       caption = "Source: World in Data")
```

La répartition de la population mondiale en 2019

Carte de la Population mondiale



Source: World in Data

Représentation de la carte des émissions moyennes de CO2 par tête (période 1990-2020)

```
join_pollution_wb_data <- join_pollution_wb_data %>%
  mutate(gdp_per_capita = gdp/population,
         co2_per_capita_en_kg = co2/population*1000000000)

data_pollution_region_mean <- join_pollution_wb_data %>%
  filter(year >= 1990 & year <= 2020) %>%
  group_by(country, iso_code,Region) %>%
  summarise(mean_gdp_per_capita = mean(gdp_per_capita, na.rm=T),
            mean_co2_per_capita = mean(co2_per_capita_en_kg, na.rm=T),
            mean_co2 = mean(co2, na.rm=T))
```

```
## `summarise()` has grouped output by 'country', 'iso_code'. You can override
## using the `.groups` argument.
```

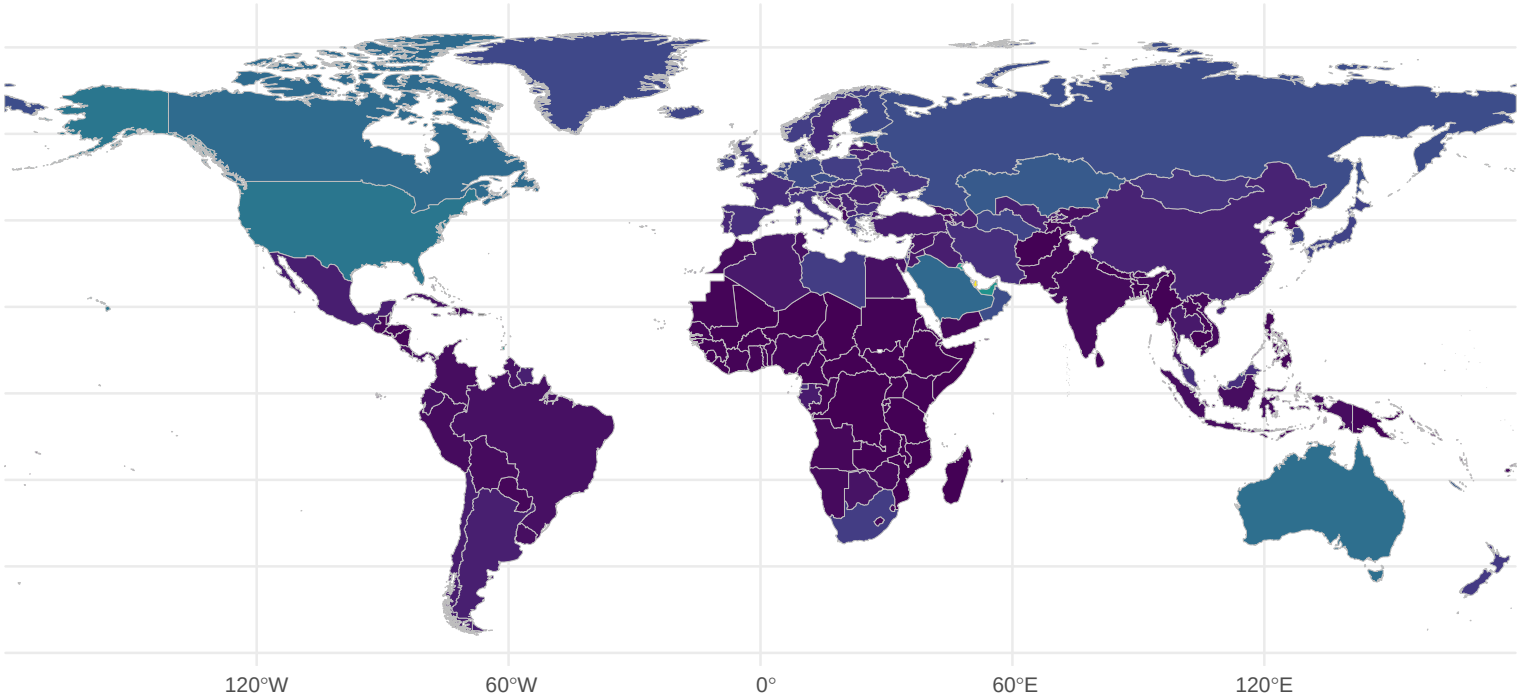
```
world_map_co3 <- world_map %>%
  right_join(data_pollution_region_mean, by=c('iso3'='iso_code'))
```

```
## Warning: le package 'viridis' a été compilé avec la version R 4.1.3

## Le chargement a nécessité le package : viridisLite
```

Carte assez peu lisible

Carte des émissions de CO2 par tête sur la période 1990−2020



Source: World in Data

Créer une variable catégorielle pour déterminer les couleurs des polygones

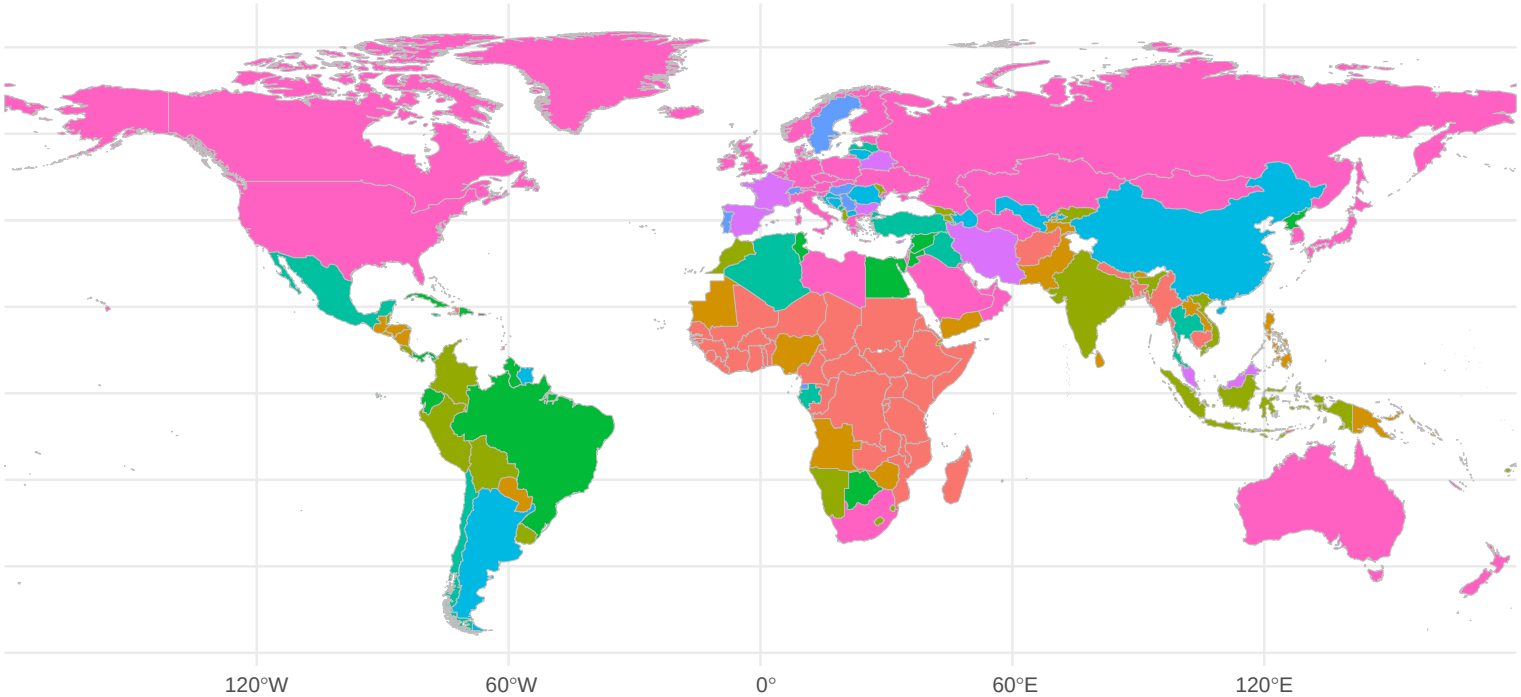
```
summary(world_map_co3$mean_co2_per_capita)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
##     37.78    632.58   2367.27   4918.59   6746.41  48809.21         1
```

```
world_map_co3$brks <-  cut(world_map_co3$mean_co2_per_capita,
                           breaks=c(0,500,1000,2000,3000,4000,5000,6000,7000
                           labels=c("[0;500[", "[500;1000[", "[1000;2000[",
```

```
map4 <- ggplot() +
  geom_sf(data = world_map_co3, col="grey", aes(fill=brks), show.legend = FALSE
  theme_minimal() +
  theme(legend.position = "right")+
  labs(x = NULL,
       y = NULL,
       title = "Carte des émissions de CO2 par tête sur la période 1990-2020",
       caption = "Source: World in Data")
```

Carte des émissions de CO2 par tête sur la période 1990−2020



Source: World in Data

Exercice: Représenter avec ggplot la carte de densité des installations électriques “charbon”

Travailler avec un autre grand type de données cartographiques: les données raster

Téléchargement d'un fichier raster

```
library(raster)
```

```
## Warning: le package 'raster' a été compilé avec la version R 4.1.3
```

```
## Le chargement a nécessité le package : sp
```

```
## Warning: le package 'sp' a été compilé avec la version R 4.1.3
```

```
##
## Attachement du package : 'raster'
```

```
## L'objet suivant est masqué depuis 'package:dplyr':
##
##     select
```

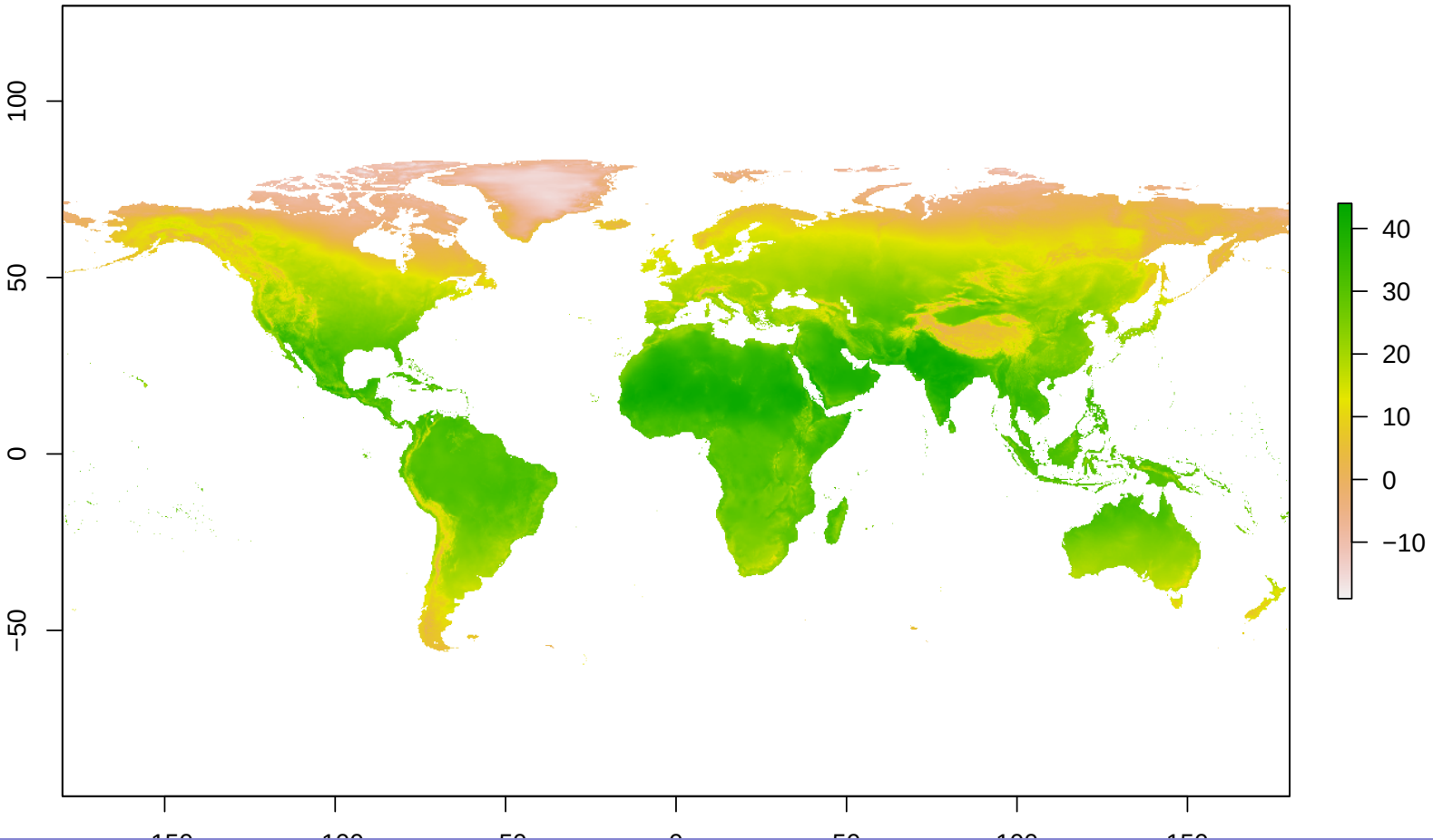
```
tmax_data <- getData(name = "worldclim", var = "tmax", res = 10)
```

```
## Warning in getData(name = "worldclim", var = "tmax", res = 10): getData will
## . Please use the geodata package instead
```

```
gain(tmax_data) <- 0.1 #must be multiplied by 0.1 to convert back to degrees Cel
tmax_data$tmax5
```

Représentation

```
plot(tmax_data$tmax5) # température maximum en mai
```



Représentation d'une jolie carte raster pour les températures maximales moyennes en mai entre 1970 et 2020.

```r
# Converting the raster object into a dataframe
tmax_data_may_df <- as.data.frame(tmax_data$tmax5, xy = TRUE, na.rm = TRUE)
rownames(tmax_data_may_df) <- c()

map5 <- ggplot(
  data = tmax_data_may_df,
  aes(x = x, y = y)
) +
  geom_raster(aes(fill = tmax5)) +
  labs(
    title = "Maximum temperature in May",
    subtitle = "For the years 1970-2000"
  ) +
  xlab("Longitude") +
  ylab("Latitude") +
  scale_fill_gradientn(
    name = "Temperature (°C)",
    colours = c("#0094D1", "#68C1E6", "#FEED99", "#AF3301"),
    breaks = c(-20, 0, 20, 40)
  )
```

Représentation d'une jolie carte raster pour les températures maximales moyennes en mai entre 1970 et 2020.