

Climate and Data

Session 3 - Maps with R

Jean-Baptiste Guiffard

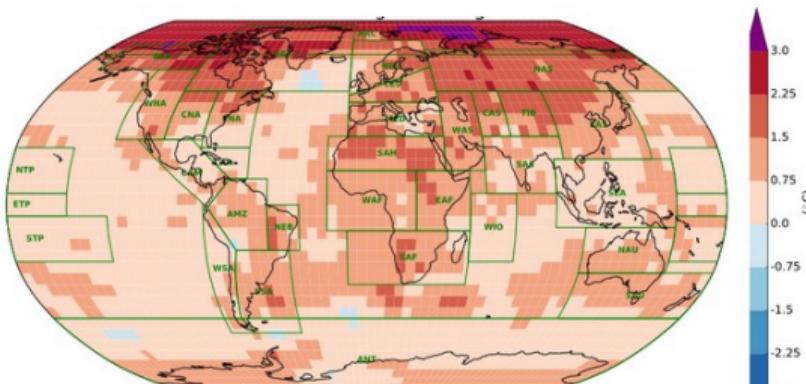
November 15, 2024

- 3.** Making maps with R
- 4.** Web scraping with R
- 5.** Extracting and analyzing textual data using R
- 6.** Advanced skills in data visualization with R

Introduction générale

Quelques cartes issues des rapports du GIEC

Réchauffement en 2006-2015 par rapport à la période pré-industrielle

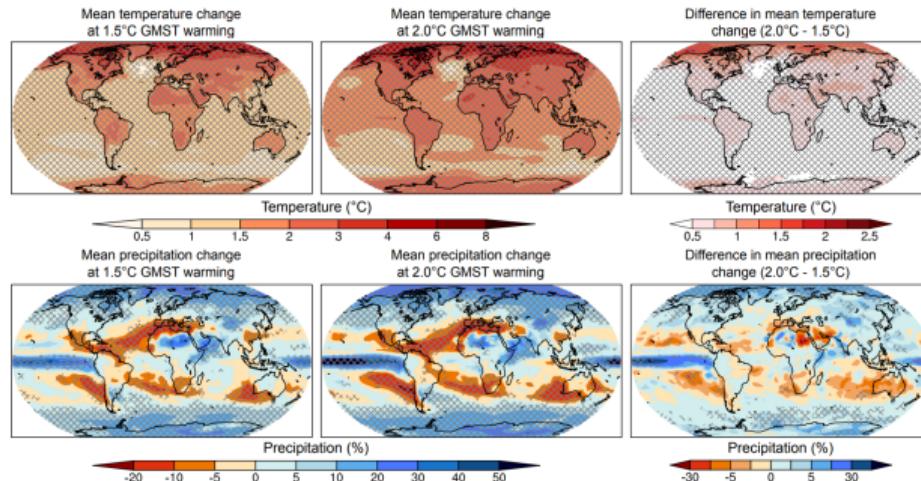


Source: Special Report IPCC (2018)
http://report.ipcc.ch/sr15/pdf/sr15_chapter1.pdf

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

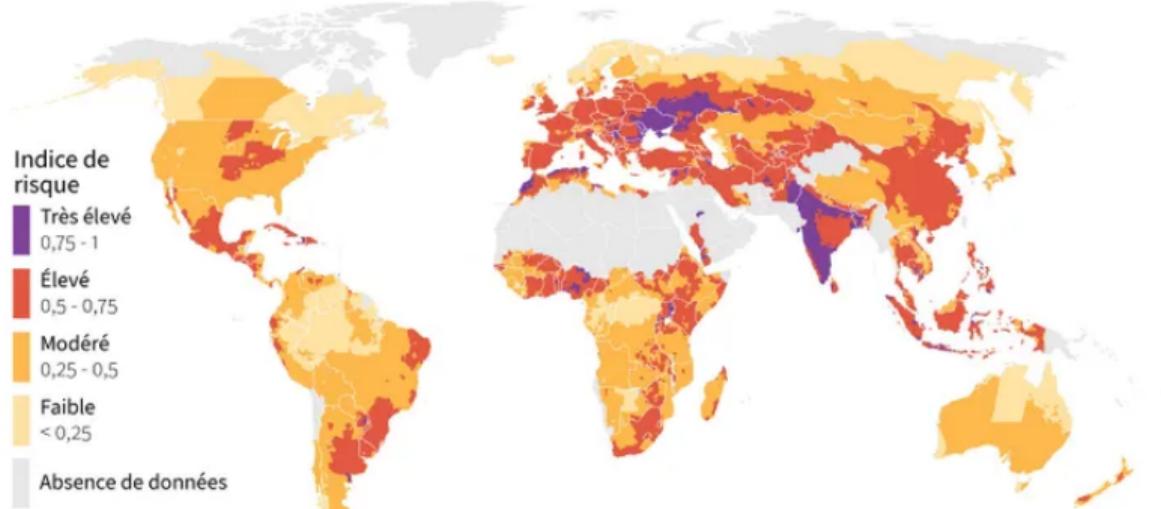


Changements projetés de la température moyenne (en haut) et de la réciprocité moyenne (en bas) pour un réchauffement planétaire de 1,5°C (à gauche) et de 2°C (au milieu) par rapport à la période préindustrielle (1861-1880), et différence entre 1,5°C et 2°C de réchauffement planétaire (à droite).

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.
 → Il est aussi probable que la fréquence des événements extrêmes de chaleur dans les régions tropicale augmentera.

Quelques cartes issues des rapports du GIEC

Risque actuel déterminé à partir de la vulnérabilité, du danger et de l'exposition



AFP

Source : Groupe de travail II Sixième Rapport d'évaluation du Giec, Carra et al, 2016

Risque de sécheresse dans le Monde

- 1. Localiser** (montre un territoire, une localisation, une situation, une répartition) → Réfléchir
- 2. Décrire** (montre des éléments, des caractéristiques, des phénomènes, des relations) → Comprendre ou interpréter
- 3. Saisir** (montre un milieu géographique, un processus historique, une combinaison de phénomènes) → Analyser ou modéliser
- 4. Découvrir** (montre une structure complexe, une dynamique, une théorie) → Décider, agir ou anticiper.

- ▶ Passer d'une information complexe à une image graphique (faire apparaître les relations d'ordre et de ressemblance entre les données)
- ▶ La carte est un moyen de communication (elle doit être lisible, compréhensible, informative, attractive) → Elle obéit donc à des règles.
- ▶ La cartographie thématique possède une méthodologie bien définie afin d'obtenir une représentation efficace → **La sémiologie graphique.**

La sémiologie graphique

Bertin, Jacques (1967). Sémiologie graphique: les diagrammes, les réseaux, les cartes.

→ Devenu une référence pour la pratique de la cartographie.

- ▶ Jusqu'ici la “bonne représentation” devait essentiellement être précise et complète, avec Bertin, la priorité devient l’efficacité de la communication.

Avec l’efficacité définie :

Il importe donc de définir un critère précis, mesurable, à partir duquel on puisse classer les constructions, définir incontestablement la meilleure et expliquer, s'il y a lieu, pourquoi certains lecteurs préfèrent une construction et certains une autre. Nous appellerons ce critère “l'efficacité” si pour obtenir une réponse correcte et complète à une question donnée, et, toutes choses égales, une construction requiert un temps d'observation plus court qu'une autre construction, on dira qu'elle est plus efficace pour cette question. (1967, 139)

C'est donc un ensemble de règles qui permettent l'utilisation d'un système graphique de signes pour la transmission d'une information correcte et accessible à un lecteur.

La cartographie est un langage avec un alphabet (point, trait, surface), un vocabulaire (variables visuelles) et une syntaxe (définie par les règles de la perception visuelle).

Ce langage doit être visuel, universel, clair et cohérent.

→ Concilier rigueur scientifique et efficacité visuelle.

Prise en compte de 3 grands paramètres

- 1. Implantation géographique** → transcription cartographique d'un objet, d'une valeur ou d'un phénomène sur un plan.
- 2. Niveaux de représentation** → Quelles données ? Variable qualitative (variation de forme, de grain, de couleur) ou variable quantitative (variations de valeurs, de couleurs, de taille). On peut discréteriser, ce qui implique une réduction de l'information mais permet la construction d'une carte lisible (un compromis).
- 3. Variables visuelles:** celle qui permet d'exprimer visuellement l'importance de la donnée → Les symboles, les couleurs, les formes, les tailles, les textures, les orientations, les valeurs.

Variables visuelles

	Points	Lines	Areas	show
Shape		possible, but too weird to show	cartogram	qualitative differences
Size			cartogram	quantitative differences
Color Hue				qualitative differences
Color Value				quantitative differences
Color Intensity				qualitative differences
Texture				qualitative & quantitative differences

Type d'implantation	Nature des données							
	Qualitative				Quantitative			
	Nominale		Ordinal		Relative		Absolue	
Ponctuelle	Forme	Couleur	Taille	Valeur	Valeur	Couleur	Texture	Taille
Linéaire	Forme	Couleur	Taille	Valeur	Couleur	Valeur	Couleur	Taille
Zonale	Couleur	Texture	Valeur	Couleur	Valeur	Couleur	Grain	Taille

Source : Zanin C., Trémélo M.-L., Savoir faire une carte. Aide à la conception et à la réalisation d'une carte thématique univariée. Ed. Belin, Paris, 2003.

Etapes de construction d'une carte

- 1.** Identifier l'objectif de la carte
- 2.** Identifier la cible, le public et le support
- 3.** Identifier l'information à cartographier
- 4.** Identifier l'implantation
- 5.** Choisir le fond de carte
- 6.** Choisir la variable visuelle et la figuration
- 7.** Choisir les habits de la carte
- 8.** Choisir la mise en scène.

- ▶ Lambert, N., & Zanin, C. (2016). Manuel de cartographie: principes, méthodes, applications. Armand Colin.
- ▶ Béguin, M., & Pumain, D. (2017). La représentation des données géographiques-4e éd.: Statistique et cartographie. Armand Colin.

La cartographie sur R

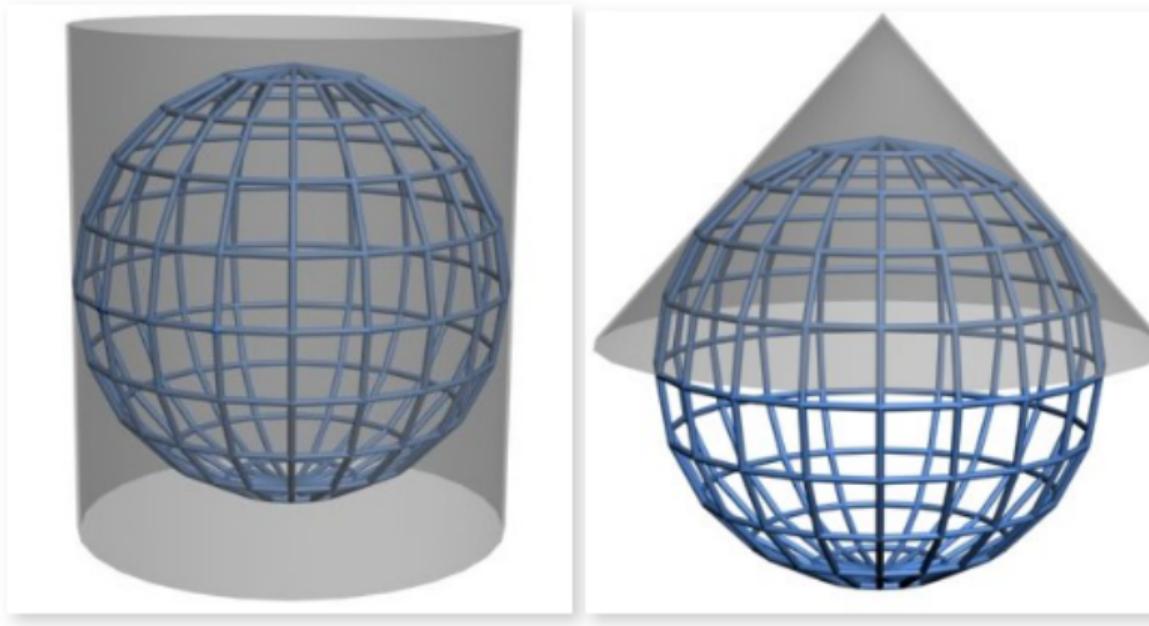
Les systèmes de coordonnées (Datum)

- ▶ 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 géoréférencées de projection.



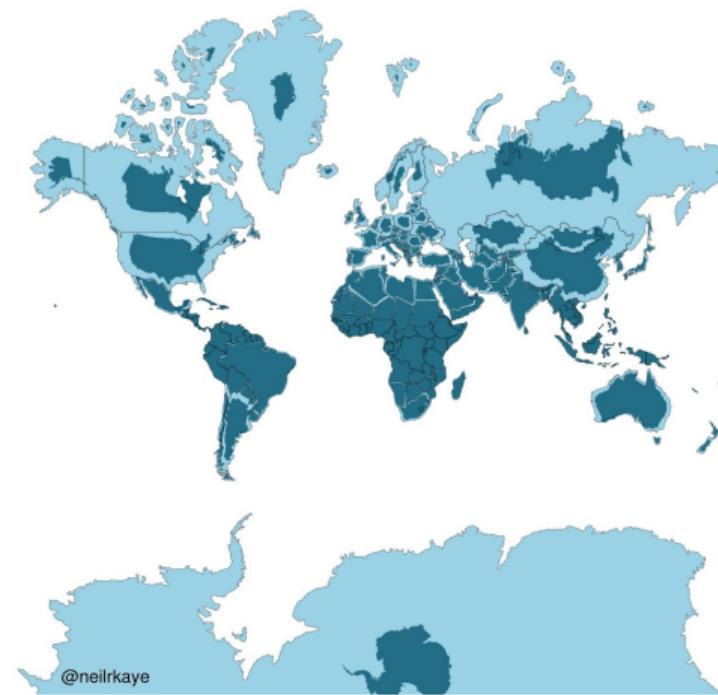
Zone des datums WGS84 et RGF93 selon EPSG.io.

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

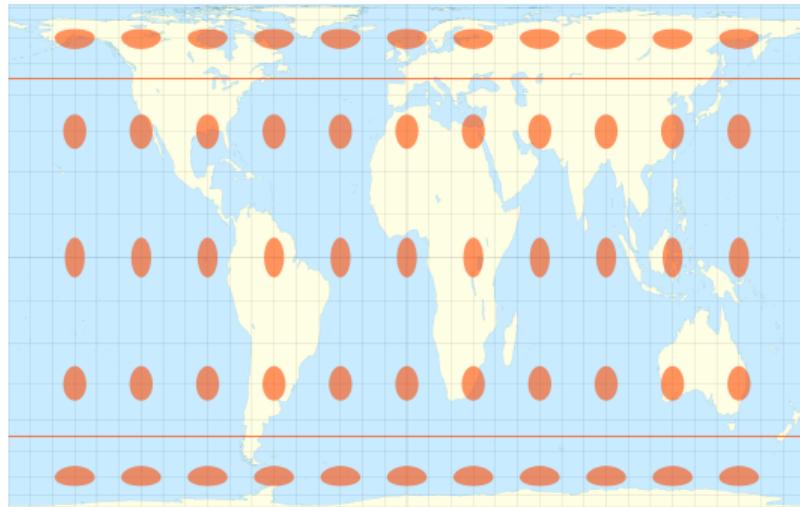
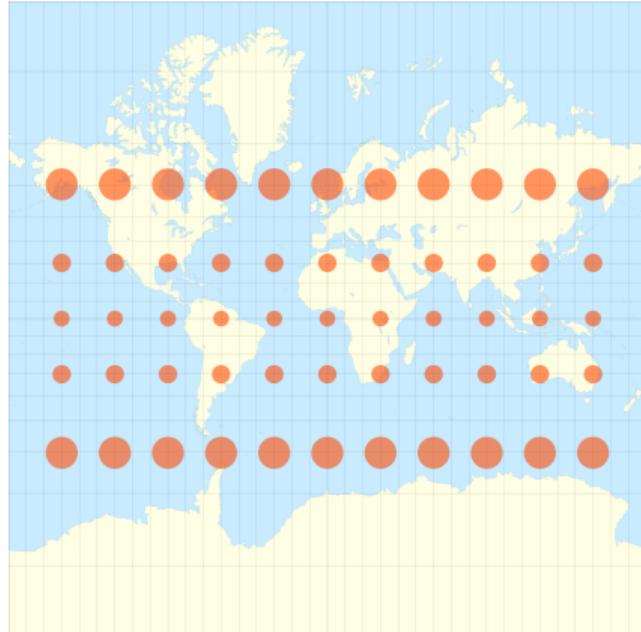


Représentation graphique des deux grands types de projection.

Comparaison de la projection Mercator avec la véritable taille des pays

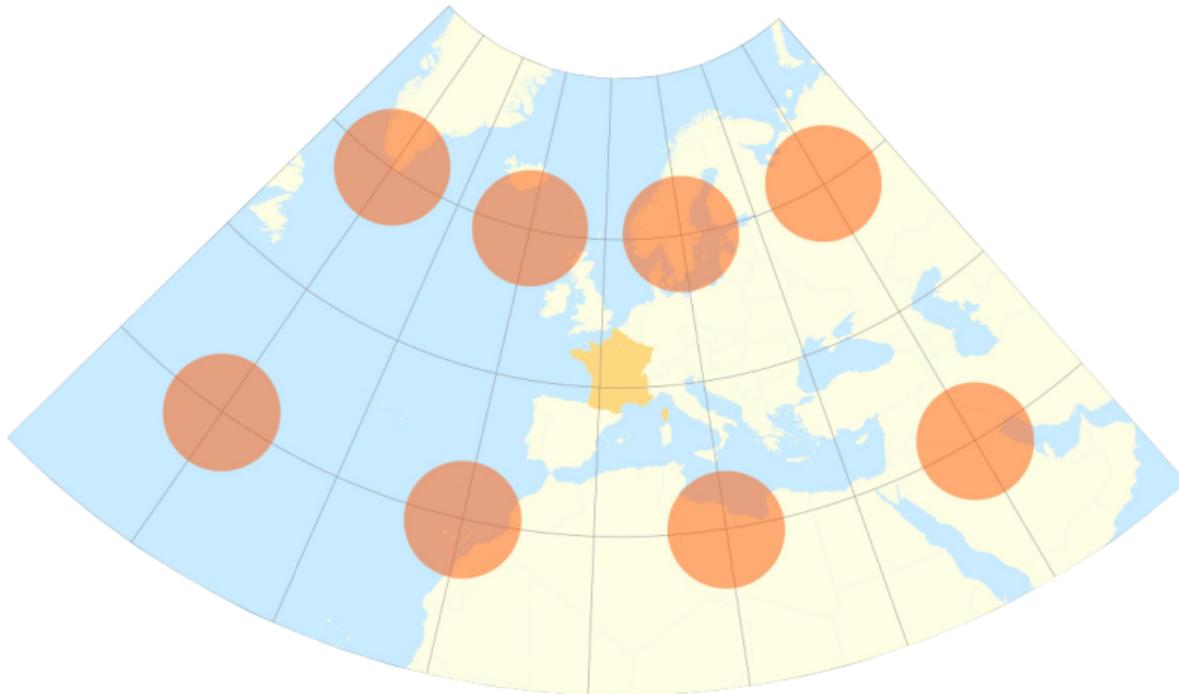


Les systèmes de projection



Projections cylindriques conforme de Mercator et équivalente de Gall-Peters.

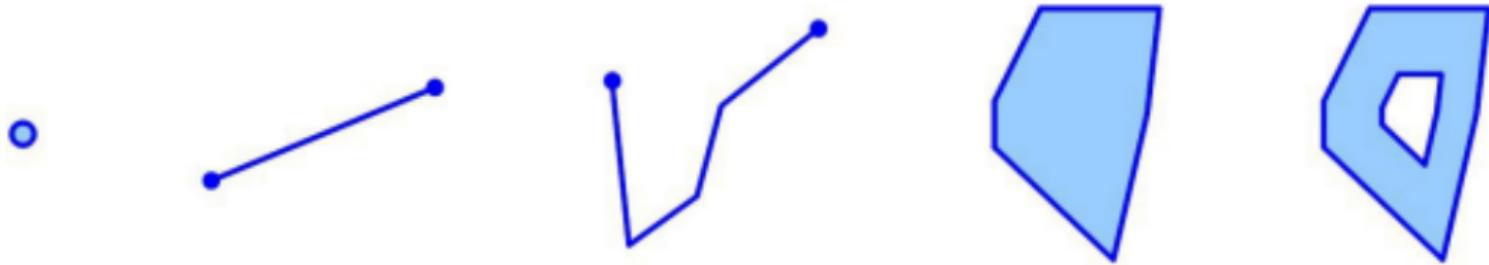
Les projections cylindriques ou l'indicatrice de Tissot, du nom de son inventeur Nicolas Auguste Tissot, est une forme géométrique (un cercle ou une ellipse) qui permet d'évaluer le degré de déformation d'un système de projection cartographique.



Projection conique conforme de Lambert centrée sur la France

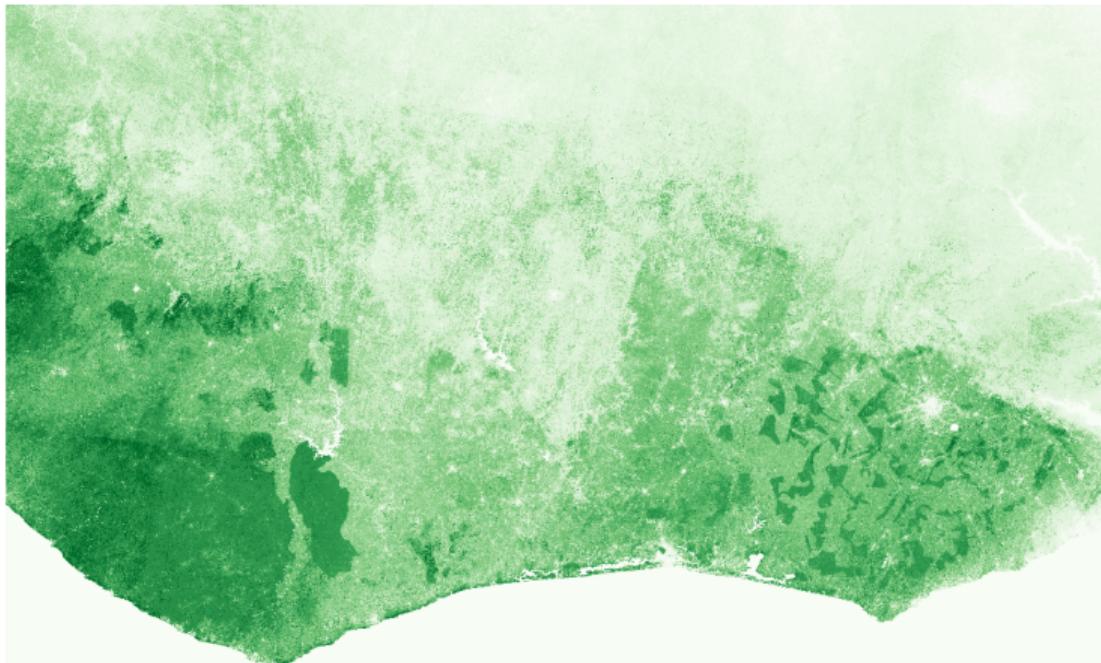
Les types de données cartographiques

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



Forme des différents vecteurs.

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



Exemple d'un raster

- ▶ 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.

- ▶ 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

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.

TYPE_1	NL_NAME_1	VARNAME_1	geometry
norate	NA	Al Aryānah 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	Banzart Banzart Bensert Binzart Biserta Bizerta	list(list(c(10.2334718704225, 10.2334718704225, 10.2337503...))
norate	NA	Qābis Gabâs	list(list(c(9.67866897583008, 9.69477558135992, 9.69743347...))

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

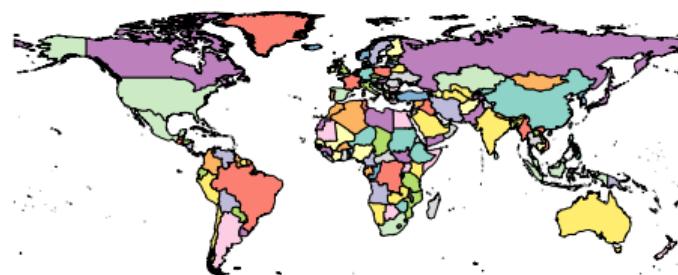
```
plot(st_geometry(world_map))
```



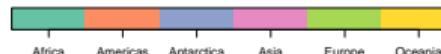
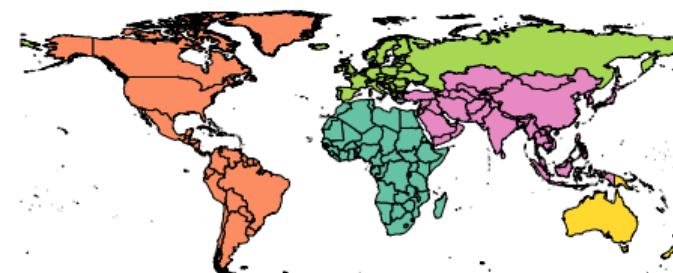
Représentation des cartes “pays” et “continents”

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

color_code



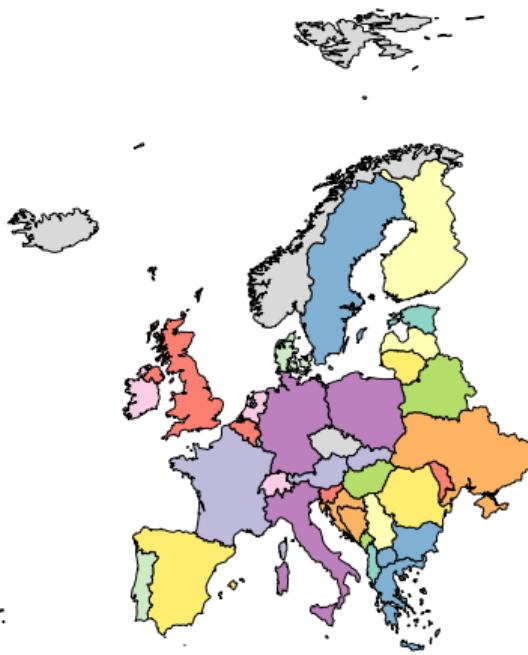
continent



Sélection de quelques polygones

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

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

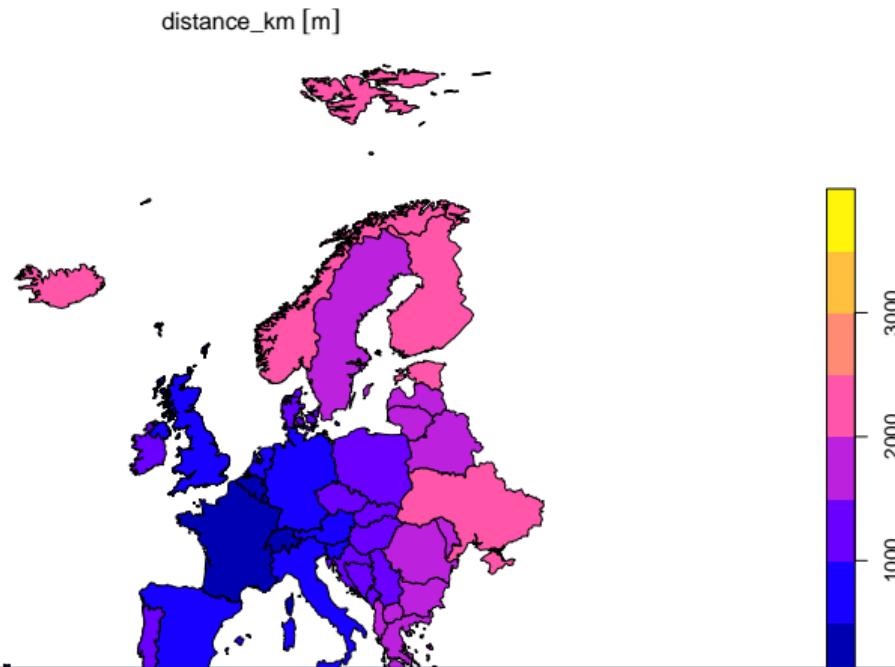
```
#install.packages("reshape2")
library(reshape2)
matrix_distance <- st_distance(Europe_centroids)
rownames(matrix_distance) <- Europe_centroids$color_code
colnames(matrix_distance) <- Europe_centroids$color_code

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="Var2" )

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



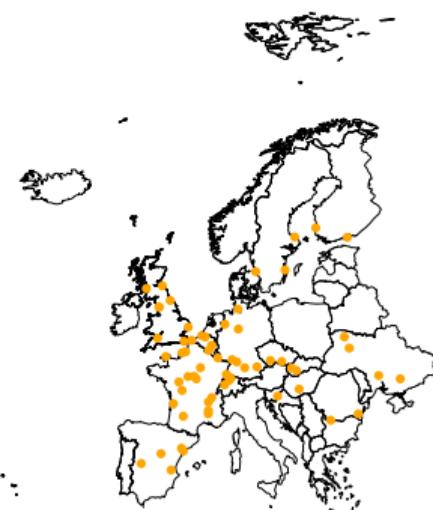
```
power_plants_points <- st_read('Global_Power_Plants/Power_Plants.shp')

## Reading layer 'Power_Plants' from data source
##   'C:\Users\Giffarrd\OneDrive - Université Paris 1 Panthéon-Sorbonne\Documents\intro_r\'
## 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))
```

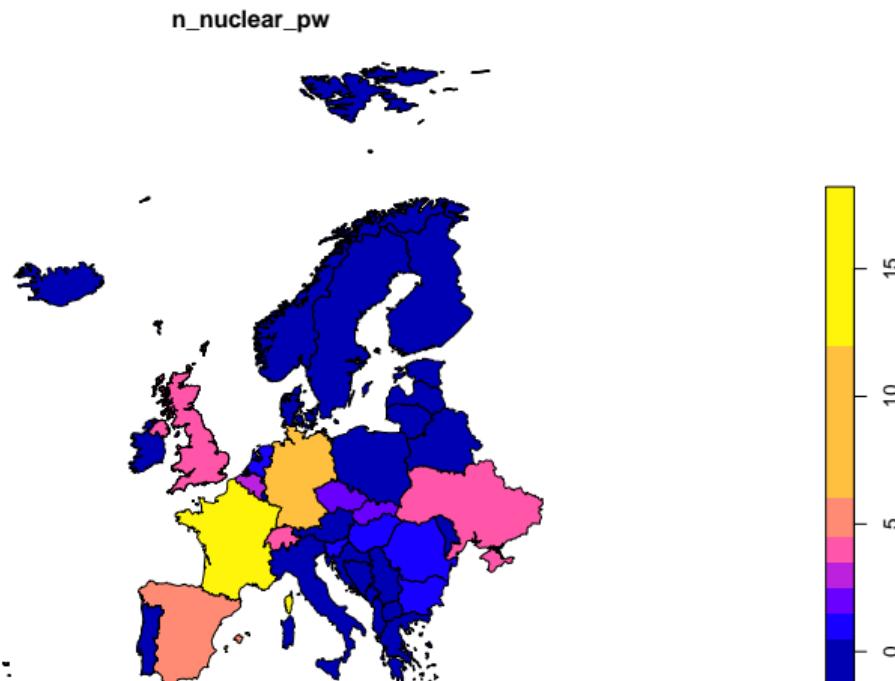
Localisation des centrales nucléaires en Europe.

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

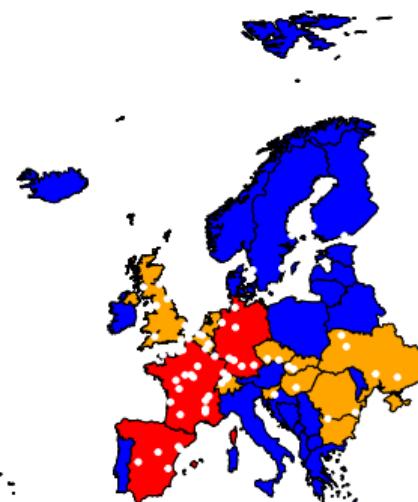


Calcul de la densité de points par aire géographique

```
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.character(fue
```

Notre première carte avec ggplot

```
data_pollution <- read.csv2('DATA/co2_clean.csv', sep=";")  
Metadata_Country <- read.csv2('DATA/Metadata_Country.csv', sep=",")  
#rename("Country_code" = "Country.Code")  
  
join_pollution_wb_data <- data_pollution %>%  
  dplyr::inner_join(Metadata_Country, by = c("iso_code" = "Country.Code"))  
  
join_pollution_wb_data <- join_pollution_wb_data %>%  
  filter(country != "") %>%  
  filter(IncomeGroup != "")
```

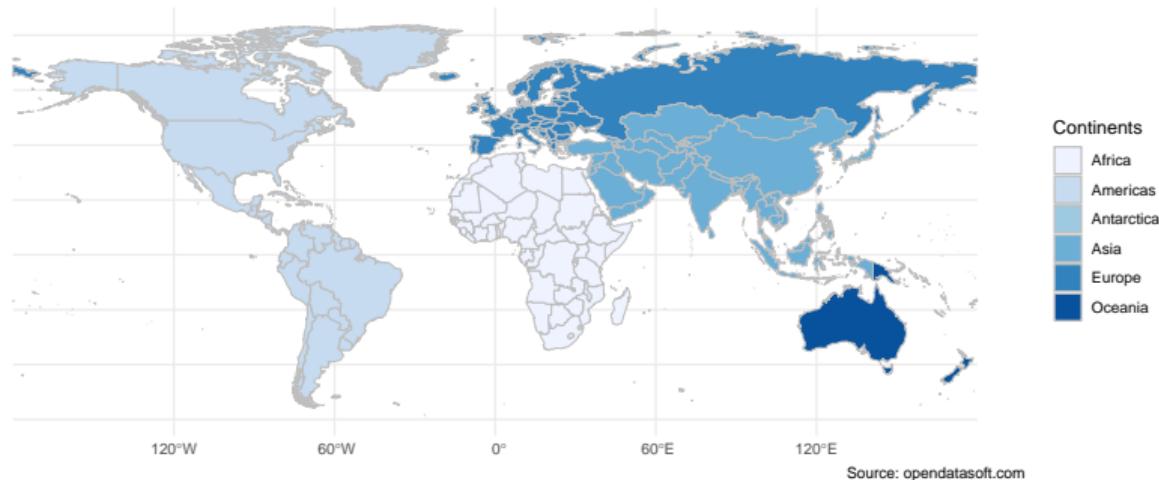
Première carte avec ggplot2

```
library(ggplot2)
map1 <- ggplot() +
  geom_sf(data = world_map, col="grey", aes(fill=continent), show.legend = TRUE, size=0.1) +
  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)
```

Carte des Continents



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 = FALSE, size=0.5) +
  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))

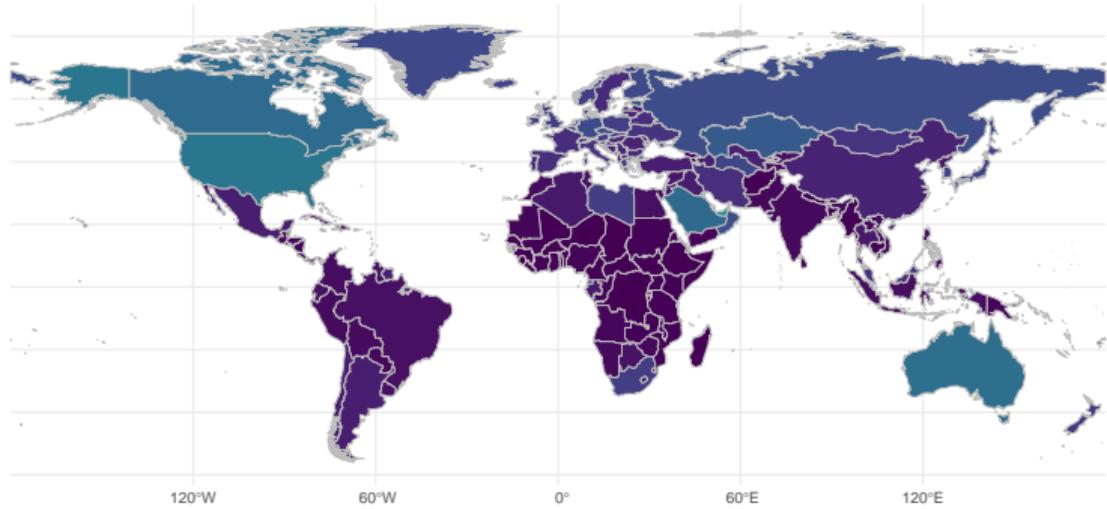
## `summarise()` has grouped output by 'country', 'iso_code'. You can override
## using the '.groups' argument.

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

## 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

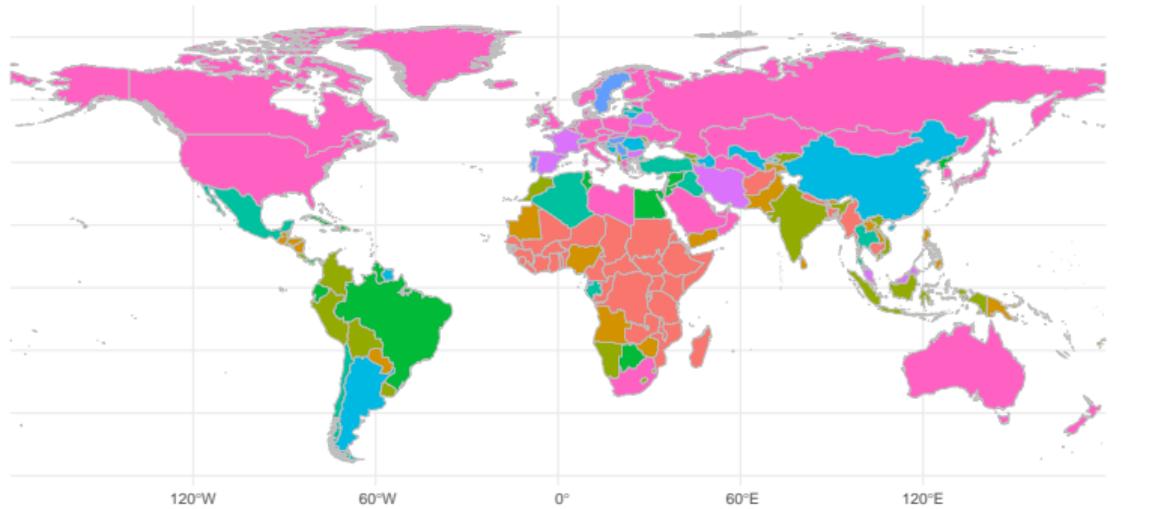
```
summary(world_map_co3$mean_co2_per_capita)

##      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(0,500,1000,2000,3000,4000,5000,6000,7000,48810),
                           labels=c("[0;500[", "[500;1000[", "[1000;2000[", "[2000;3000["))

map4 <- ggplot() +
  geom_sf(data = world_map_co3, col="grey", aes(fill=brks), show.legend = FALSE, size=0.1) +
  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)

## Le chargement a nécessité le package : sp

##
## Attachement du package : 'raster'

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

library(ggplot2)
tmax_data <- getData(name = "worldclim", var = "tmax", res = 10)

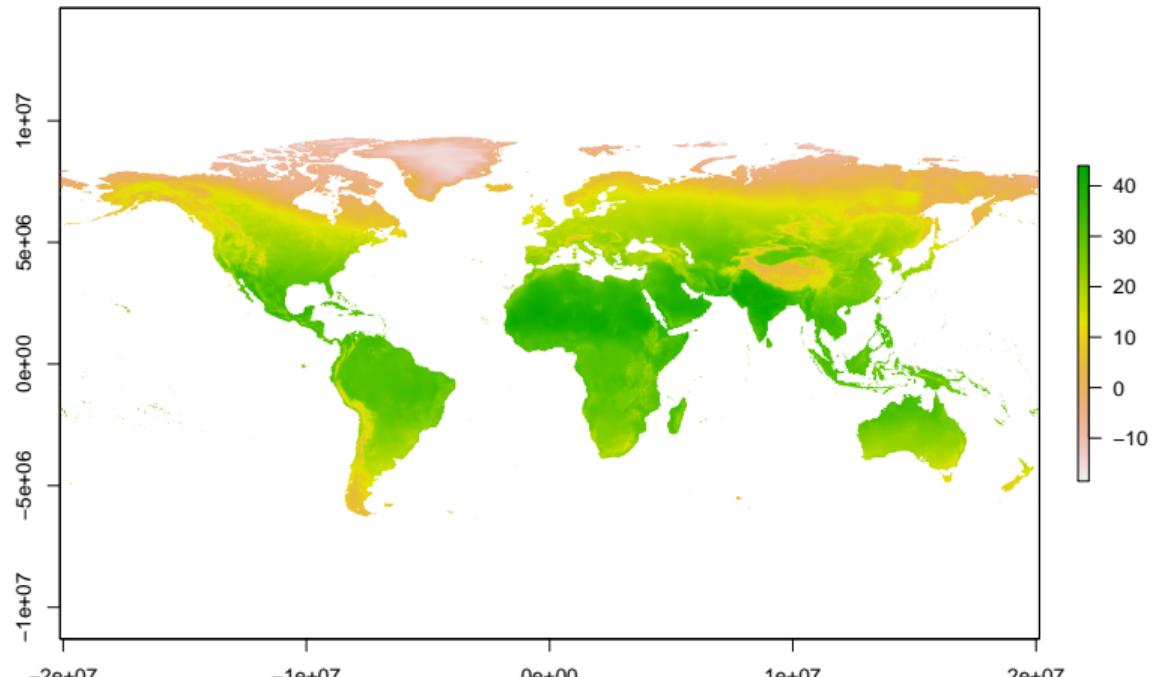
## Warning in getData(name = "worldclim", var = "tmax", res = 10): getData will be removed
## . Please use the geodata package instead

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

## class      : RasterLayer
## dimensions : 900, 2160, 1944000 (nrow, ncol, ncell)
```

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.

```
# 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.

Maximum temperature in May

For the years 1970–2000

