

Autour du SARS-CoV-2 (Covid-19)

Pauline

Préparation des données

Les données que nous utiliserons dans un premier temps sont compilées par le Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE). L'URL est:

```
data_url = read.csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/
```

Téléchargement et sélection des données

Le jeu de données prend la forme suivante :

```
data<-data_url  
head(data[,1:6])
```

##	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20
## 1		Afghanistan	33.93911	67.70995	0	0
## 2		Albania	41.15330	20.16830	0	0
## 3		Algeria	28.03390	1.65960	0	0
## 4		Andorra	42.50630	1.52180	0	0
## 5		Angola	-11.20270	17.87390	0	0
## 6	Antigua and Barbuda		17.06080	-61.79640	0	0

```
tail(data[,1:6])
```

##	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20
## 274		Venezuela	6.42380	-66.58970	0	0
## 275		Vietnam	14.05832	108.27720	0	2
## 276	West Bank and Gaza		31.95220	35.23320	0	0
## 277		Yemen	15.55273	48.51639	0	0
## 278		Zambia	-13.13390	27.84933	0	0
## 279		Zimbabwe	-19.01544	29.15486	0	0

Les pays qui nous intéressent sont la Belgique (Belgium), la Chine - toute les provinces sauf Honk-Kong (China), Hong Kong (China, Hong-Kong), la France métropolitaine (France), l'Allemagne (Germany), l'Iran (Iran), l'Italie (Italy), le Japon (Japan), la Corée du Sud (Korea, South), la Hollande sans les colonies (Netherlands), le Portugal (Portugal), l'Espagne (Spain), le Royaume-Uni sans les colonies (United Kingdom), les États-Unis (US).

```
names(data)[match("Country/Region",names(data))] <- "Country.Region"  
names(data)[match("Province/State",names(data))] <- "Province.State"
```

```
data1<-subset(data, c(Country.Region == "Belgium" | Country.Region == "Japan" | Country.Region == "Korea"  
data2<-subset(data[c(59:70,72:92),])  
data3<-subset(data[71,])  
data4<-subset(data[131,])
```

```
data5<-subset(data[270,])
data6<-subset(data[198,])
data7<-rbind(data1, data2)
data8<-rbind(data7, data3)
data9<-rbind(data8, data4)
data10<-rbind(data9, data5)
data.CR<-rbind(data10, data6)

data.CR$Country.Region[data.CR$Province.State == "Hong Kong" & data.CR$Country.Region == "China"] <- "C
data.CR$Country.Region[data.CR$Province.State != "Hong Kong" & data.CR$Country.Region == "China"] <- "C
```

Transformation

À la suite de la précédente manipulation, nous supprimons la variable Province.State qui ne nous est plus utile pour la suite de l'analyse :

```
data.CR<-data.CR[, -c(1)]
```

Les variables sur les latitudes et les longitudes ne nous intéressent pas pour la suite du traitement des données.

```
data.CR<-data.CR[, -c(2,3)]
```

Pour faciliter le traitement suivant, nous agrégeons les provinces de la Chine (hors Hong Kong) ensemble :

```
data.CR<-aggregate(. ~ Country.Region, data.CR, sum, na.rm = T)
```

Inspection

Nous regardons notre nouveau jeu de données :

```
head(data.CR[, 1:6])
```

```
##           Country.Region 1/22/20 1/23/20 1/24/20 1/25/20 1/26/20
## 1           Belgium           0         0         0         0         0
## 2 Chine (sans Hong Kong)     548       641       918      1401      2067
## 3     Chine, Hong Kong           0         2         2         5         8
## 4           France           0         0         2         3         3
## 5           Germany           0         0         0         0         0
## 6           Iran             0         0         0         0         0
```

Nous vérifions la nature de nos variables :

```
str(data.CR[, 1:6])
```

```
## 'data.frame':   13 obs. of  6 variables:
## $ Country.Region: chr  "Belgium" "Chine (sans Hong Kong)" "Chine, Hong Kong" "France" ...
## $ 1/22/20       : int  0 548 0 0 0 0 2 1 0 0 ...
## $ 1/23/20       : int  0 641 2 0 0 0 2 1 0 0 ...
## $ 1/24/20       : int  0 918 2 2 0 0 2 2 0 0 ...
## $ 1/25/20       : int  0 1401 5 3 0 0 2 2 0 0 ...
## $ 1/26/20       : int  0 2067 8 3 0 0 4 3 0 0 ...
```

Transformation

Pour réaliser un graphique, nous allons devoir mettre les colonnes en lignes successives de la manière suivante :

```
library(tidyr)
data.CR2<-data.CR %>% pivot_longer(-Country.Region)
```

Par la suite, nous allons définir le format de date de notre variable renommée name :

```
data.CR2$name<-as.POSIXct(data.CR2$name, format = "%m/%d/%Y")
View(data.CR2)
```

Graphique

Il est maintenant possible d'établir un graphique démontrant le nombre de personnes contaminées dans chaque pays étudié depuis le début de la pandémie.

```
library(ggplot2)
ggplot(data.CR2, aes(name, value, color= Country.Region)) + geom_line() + labs(x="Date (YYYY-MM)", y="N
```

