

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

2024-02-09

```
# LIBRAIRIES ----  
library(janitor)  
library(tidyverse)  
library(ggrepel)  
library(ggtext)  
library(rcartocolor)  
  
rm(list=ls())
```

## Origine des datas

Les données que nous utiliserons dans un premier temps sont compilées par le Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE) et sont mises à disposition sur GitHub. C'est plus particulièrement sur les données `time_series_covid19_confirmed_global.csv` (des suites chronologiques au format csv) disponibles à l'adresse : [https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse\\_covid\\_19\\_data/csse\\_covid\\_19\\_time\\_series/time\\_series\\_covid19\\_confirmed\\_global.csv](https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv), que nous allons nous concentrer.

On vérifie si il y a des NA

```
##           Province.State Country.Region nb  
## 1 Repatriated Travellers      Canada    2  
## 2                Unknown      China    2
```

Deux valeurs NA sont présentes sur 2 lignes : “Canada / Repatriated Travellers” et “China / Unknown”. Cela ne devrait pas impacter nos analyses à venir.

## Sélection des données

Nous choisissons de travailler sur les données des pays suivants : la Belgique (Belgium), la Chine - toutes les provinces sauf Hong-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).

```
selectPaysProvince <- function(pays,province){  
  data %>%  
    filter(Country.Region==pays,Province.State==province) %>%  
    select(-Province.State, -Lat, -Long)}  
  
#France  
data_fr <- selectPaysProvince("France","")  
  
#Pays-Bas  
data_nl <- selectPaysProvince("Netherlands","")
```

```

#Chine hors HK
data_ch <- data %>%
  filter(Country.Region=="China",Province.State!="Hong Kong") %>%
  select(-Province.State, -Lat, -Long)%>%
  adorn_totals("row", name="China (Hors HK)") %>%
  filter(Country.Region=="China (Hors HK)")

#Hong Kong
data_hk <- selectPaysProvince("China","Hong Kong")

data_hk <-data_hk %>%
  adorn_totals("row", name="China (Hong Kong)") %>%
  filter(Country.Region=="China (Hong Kong)")

#Autres pays
data_cum <- data %>%
  filter(Country.Region %in% c("Belgium","Germany","Iran","Italy","Japan","Korea.South","Portugal","Spa
  select(-Province.State, -Lat, -Long)

data_fin <- rbind(data_fr, data_nl, data_ch, data_hk, data_cum)

data_fin <- data_fin %>%
  pivot_longer(cols =! Country.Region, names_to = "date", values_to = "count") %>%
  mutate(date=substr(date,2,nchar(date))) %>%
  mutate(date=mdy(date))

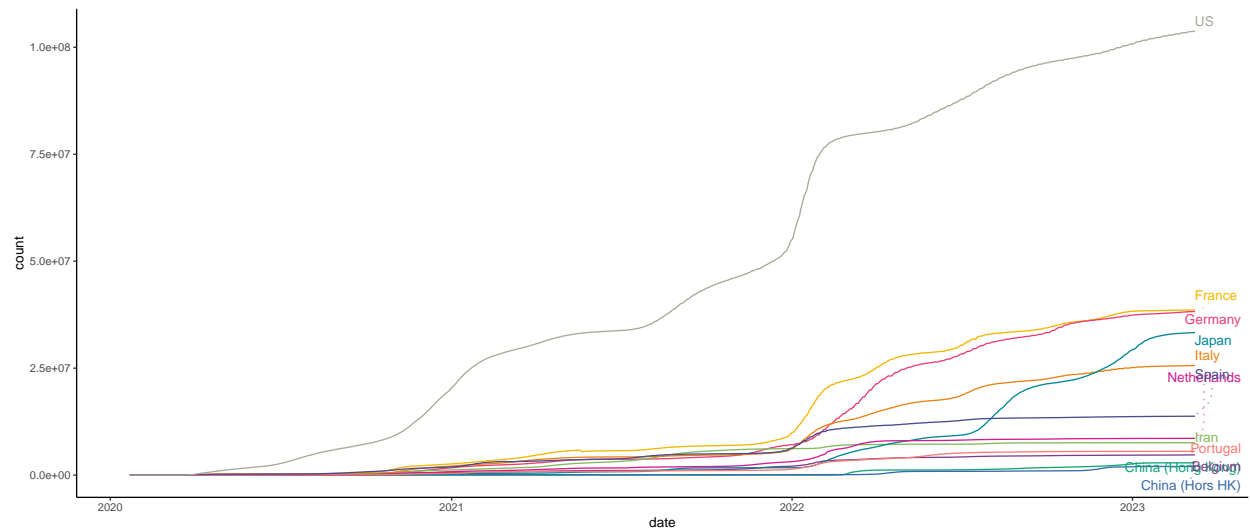
```

Graph 1 : Cumul (échelle linéaire)

```

ggplot(data_fin, aes(x=date, y=count, group=Country.Region, colour=Country.Region))+
  geom_line()+
  scale_color_manual(values = c(carto_pal(n = 12, name = "Bold")))+
  geom_text_repel(data = subset(data_fin,date == max(date)),max.overlaps = Inf,
    aes(color=Country.Region, label=Country.Region),
    direction="y",hjust = 0, segment.size = .7,
    segment.alpha = .5,segment.linetype = "dotted",
    box.padding = .4, segment.curvature = -0.1,
    segment.ncp = 3,segment.angle = 20)+
  guides(color="none")+
  theme_classic()

```



Graphe 2 : Cumul (échelle logarithmique)

```
ggplot(data_fin, aes(x=date, y=count, group=Country.Region, colour=Country.Region))+
  geom_line()+
  scale_color_manual(values = c(carto_pal(n = 12, name = "Bold")))+
  geom_text_repel(data = subset(data_fin, date == max(date)), max.overlaps = Inf,
    aes(color=Country.Region, label=Country.Region),
    direction="y", hjust = 0, segment.size = .7,
    segment.alpha = .5, segment.linetype = "dotted",
    box.padding = .4, segment.curvature = -0.1,
    segment.ncp = 3, segment.angle = 20)+
  guides(color="none")+
  scale_y_continuous(trans = "log")+
  theme_classic()
```

