Sujet 7 : Autour du SARS-CoV-2 (Covid-19)

EL BOUAZZATI Mohamed

16/06/2021

Préparation des données

Les données que j'ai utilisé sont rassemblé par Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE) mises à disposition sur Github. sous forme d'un fichier en format CSV dont chaque ligne correspond une province par pays. L'URL est:

Téléchargement des données sur github

```
library(readr)
data = read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/cs
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   `Province/State` = col_character(),
##   `Country/Region` = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

Sélection des données pour les pays desirés

les données concernant uniquement les 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-Unis sans les colonies (United Kingdom), les États-Unis (US).

```
DataDesired = subset(data, data$`Country/Region`=='Belgium'|
               data$`Country/Region`=='China'|
               data$`Country/Region`=='France'|
               data$`Country/Region`=='Germany'|
               data$`Country/Region`=='Iran'|
               data$`Country/Region`=='Italy'|
               data$`Country/Region`=='Japan'|
               data$`Country/Region`=='Korea, South'|
               data$`Country/Region`=='Netherlands'|
               data$`Country/Region`=='Portugal'|
               data$`Country/Region`=='Spain'|
               data$`Country/Region`=='US'|
               data$`Country/Region`=='United Kingdom'
              )
```

```
less = c(14,36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 53, 54, 55,56, 61, 62, 63, 64, 65, 66, 67, 68,69
DataSelected = DataDesired[-less,]
```

###Selection des colonnes concernés et formatage des dates

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(reshape2)
table_pays = melt(DataSelected, id.vars = 'Country/Region',
                          measure.vars = colnames(select(DataSelected,
                                                         ends_with("/20"))))
table_pays = table_pays[!is.na(table_pays$value),]
table_pays$Pays_Date = paste(table_pays$'Country/Region',table_pays$'variable',sep="_")

table_pays2 = melt(DataSelected, id.vars = 'Country/Region',
                          measure.vars = colnames(select(DataSelected,
                                                         ends_with("/21"))))

table_pays2$Pays_Date = paste(table_pays2$'Country/Region',table_pays2$'variable',sep="_")
table_pays2 = table_pays2[!is.na(table_pays2$value),]
table_pays3 = bind_rows(table_pays, table_pays2)
```

```
## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

Fusion des nombres par pays et par date

```
nb_cas_pays = aggregate(value ~ Pays_Date, data= table_pays3, sum)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:reshape2':
##
##     smiths
```

```
nb_cas_pays = separate(nb_cas_pays, "Pays_Date", c("Pays", "Date"), sep = "_")
nb_cas_pays$Pays_Date = paste(nb_cas_pays$'Pays',nb_cas_pays$'Date',sep="_")
```

Verification des classes des colonnes value et Pays.

```
class(nb_cas_pays$'value')
```

```
## [1] "numeric"
```

```
class(nb_cas_pays$'Pays')
```

```
## [1] "character"
```

Verification des données manquants dans le tableau

```
na_records = apply(nb_cas_pays, 1, function (x) any(is.na(x)))
nb_cas_pays[na_records,]
```

```
## [1] Pays      Date      value     Pays_Date
## <0 rows> (or 0-length row.names)
```

Conversion des numéros de semaine

```
library("lubridate")
```

```
## Warning: package 'lubridate' was built under R version 3.6.3
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:dplyr':
##
##     intersect, setdiff, union
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library("magrittr")
```

```
##
## Attaching package: 'magrittr'

## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(parsedate)
```

```
## Warning: package 'parsedate' was built under R version 3.6.3

##
## Attaching package: 'parsedate'

## The following object is masked from 'package:readr':
##
##     parse_date
```

###Modification du format de la date dans la colonne 'convert_date'

```
nb_cas_pays$convert_date = mdy(nb_cas_pays$Date)
```

###Les points sont dans l'ordre chronologique inverse, il est donc utile de les trier:

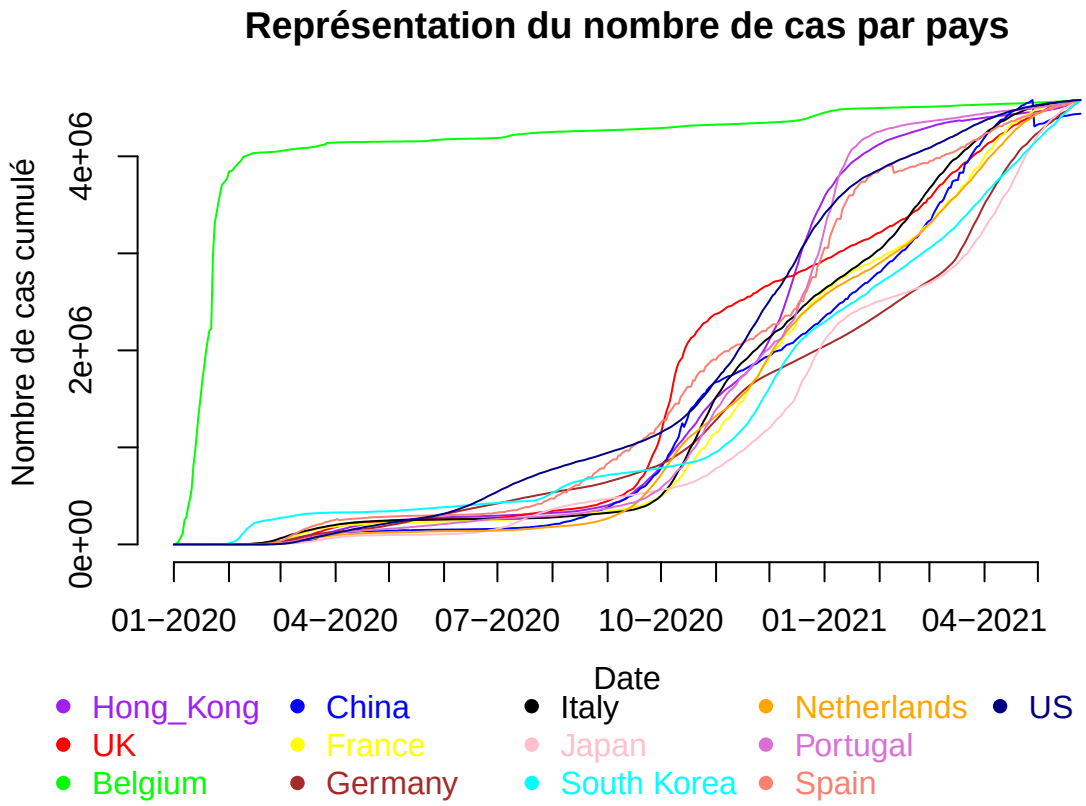
```
nb_cas_pays = nb_cas_pays[order(nb_cas_pays$convert_date),]
```

###Le jeu de données est divisé en petit tableau par différents pays.

```
library(dplyr)
table_China = subset(nb_cas_pays, nb_cas_pays$Pays == "China")
table_Belgium = subset(nb_cas_pays, nb_cas_pays$Pays == "Belgium")
table_Hong_Kong = subset(DataDesired, DataDesired$`Province/State` == "Hong Kong")
table_France = subset(nb_cas_pays, nb_cas_pays$Pays == "France")
table_Germany = subset(nb_cas_pays, nb_cas_pays$Pays == "Germany")
table_Iran = subset(nb_cas_pays, nb_cas_pays$Pays == "Iran")
table_Italy = subset(nb_cas_pays, nb_cas_pays$Pays == "Italy")
table_Japan = subset(nb_cas_pays, nb_cas_pays$Pays == "Japan")
table_Korea = subset(nb_cas_pays, nb_cas_pays$Pays == "Korea, South")

table_Netherlands = subset(nb_cas_pays, nb_cas_pays$Pays == "Netherlands")
table_Portugal = subset(nb_cas_pays, nb_cas_pays$Pays == "Portugal")
table_Spain = subset(nb_cas_pays, nb_cas_pays$Pays == "Spain")
table_UK = subset(nb_cas_pays, nb_cas_pays$Pays == "United Kingdom")
table_US = subset(nb_cas_pays, nb_cas_pays$Pays == "US")
```

Représentation graphique des nombres de cas par pays

```
plot.new()
par(mar=c(7,4,3,3))
plot_UK = plot(x=table_UK$convert_date, y=table_UK$value, type="l", axes=F, xlab="", ylab="", col="purpl
axis(side=2, at=seq(0e+00, 4e+06, by=1e+06))
mtext("Nombre de cas cumulé", side=2, line=2.5)
par(new=T)
plot_Belgium = plot(x=table_Belgium$convert_date, y=table_Belgium$value, type="l", axes=F, xlab="", ylab
par(new=T)
plot_China = plot(x=table_China$convert_date, y=table_China$value, type="l", axes=F, xlab="", ylab="",
par(new=T)
plot_France = plot(x=table_France$convert_date, y=table_France$value, type="l", axes=F, xlab="", ylab="
par(new=T)
plot_Germany = plot(x=table_Germany$convert_date, y=table_Germany$value, type="l", axes=F, xlab="", ylab
par(new=T)
plot_Iran = plot(x=table_Iran$convert_date, y=table_Iran$value, type="l", axes=F, xlab="", ylab="", col=
par(new=T)
plot_Italy = plot(x=table_Italy$convert_date, y=table_Italy$value, type="l", axes=F, xlab="", ylab="",
par(new=T)
plot_Japan = plot(x=table_Japan$convert_date, y=table_Japan$value, type="l", axes=F, xlab="", ylab="",
par(new=T)

plot_Netherlands = plot(x=table_Netherlands$convert_date, y=table_Netherlands$value, type="l", axes=F,
par(new=T)
plot_Portugal = plot(x=table_Portugal$convert_date, y=table_Portugal$value, type="l", axes=F, xlab="",
par(new=T)
plot_Spain = plot(x=table_Spain$convert_date, y=table_Spain$value, type="l", axes=F, xlab="", ylab="",
par(new=T)

plot_Korea = plot(x=table_Korea$convert_date, y=table_Korea$value, type="l", axes=F, xlab="", ylab="",
par(new=T)
plot_US = plot(x=table_US$convert_date, y=table_US$value, type="l", axes=F, xlab="", ylab="", col="navy
axis.Date(side=1, at=seq(min(table_UK$convert_date), max(table_UK$convert_date), by="months"), format="
mtext("Date", side=1, line=2.5)
legend(x="top",legend = c("Hong_Kong","UK", "Belgium", "China", "France", "Germany", "Italy", "Japan",
title(main = "Représentation du nombre de cas par pays")
```


Représentation du nombre de cas par pays
Nombre de cas cumulé
0e+00
2e+06
4e+06
01−2020
04−2020
07−2020
10−2020
01−2021
04−2021
Date
Hong_Kong
China
Italy
Netherlands
US
UK
France
Japan
Portugal
Belgium
Germany
South Korea
Spain