Statistique : devoir, décembre 2020

Le sujet est composé de **5 exercices indépendants**. Vous devrez répondre aux questions sur un document Markdown avec une sortie au format **html ou pdf**. Ce document devra afficher les codes R ainsi que les sorties qui permettent de répondre aux questions. A la fin de l'épreuve vous enverrez par email le **fichier de sortie compilé correctement au format html ou pdf** ainsi que le **fichier source au format Rmd** par email à laurent.rouviere@univ-rennes2.fr. La qualité du document markdown sera prise en compte dans le barème tout comme la structure et l'élégance des codes **R**.

On utilisera les packages suivants :

```
library(tidyverse)
theme_set(theme_classic(base_size=10))
library(lubridate)
```

Exercice 1 (quelques calculs de probabilités)

1. On considère X une variable de loi binomiale $B(10, 0.4)$.
   a) Calculer les probabilités (on donnera les résultats sans utiliser de fonctions **R** mais en justifiant brièvement).
   $$P(X = -1),\ P(X \leq -1) \quad \text{et} \quad P(X \geq -1).$$
   La loi Binomiale ne peut prendre que des valeurs positives. Les deux premières probabilités sont donc nulles et la dernière vaut 1.
   b) Calculer les probabilités (on peut utiliser des fonctions **R** à partir de maintenant).
   $$P(X = 1),\ P(X = 4) \quad \text{et} \quad P(X = 10).$$

```
dbinom(c(1,4,10),size=10,prob=0.4)
[1] 0.0403107840 0.2508226560 0.0001048576
```

   c) Calculer les probabilités
   $$P(X \leq 3),\ P(X > 4),\ P(X > 3.5) \quad \text{et} \quad P(2 \leq X \leq 8).$$

```
pbinom(3,size=10,prob=0.4)
[1] 0.3822806
sum(dbinom(5:10,size=10,prob=0.4))
[1] 0.3668967
sum(dbinom(4:10,size=10,prob=0.4))
[1] 0.6177194
sum(dbinom(2:8,size=10,prob=0.4))
[1] 0.9519649
```

2. On considère ici Y une variable de loi normale d'espérance 3 et de variance 1 (notée $N(3, 1)$).
   a) Calculer les probabilités
   $$P(Y = 3) \quad \text{et} \quad P(Y = 0).$$
   Ces deux probabilités sont nulles puisque Y est une variable continue.
   b) Calculer les probabilités
   $$P(Y \leq 2),\ P(Y < 2) \quad \text{et} \quad P(Y > 2).$$

```
pnorm(2,3,1)
[1] 0.1586553
pnorm(2,3,1)
[1] 0.1586553
1-pnorm(2,3,1)
[1] 0.8413447
```

c) Calculer les probabilités

$$P(2 \leq Y \leq 4) \quad \text{et} \quad P(Y \leq 2 \text{ ou } Y \geq 3.5).$$

```
pnorm(4,3,1)-pnorm(2,3,1)
[1] 0.6826895
pnorm(2,3,1)+(1-pnorm(3.5,3,1))
[1] 0.4671928
```

Exercice 2 (intervalle de confiance)

On considère les données sur les iris de Fisher. Calculer un intervalle de confiance de niveau 95% pour les paramètres suivants :

- La longueur de Pétales moyenne

```
data(iris)
t.test(iris$Petal.Length,conf.level=0.95)$conf.int
[1] 3.473185 4.042815
attr(,"conf.level")
[1] 0.95
```

- La largeur de Sépales moyenne de l'espèce Setosa

```
sep_set <- iris %>% filter(Species=="setosa") %>% select(Sepal.Width)
t.test(sep_set,conf.level=0.95)$conf.int
[1] 3.320271 3.535729
attr(,"conf.level")
[1] 0.95
#ou
iris %>% filter(Species=="setosa") %>% select(Sepal.Width) %>%
  t.test(conf.level=0.95)

    One Sample t-test

data:  .
t = 63.946, df = 49, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 3.320271 3.535729
sample estimates:
mean of x
    3.428
```

- La longueur de Sépales moyenne pour les iris des espèces Setosa ou Virginica

```
sep_setvin <- iris %>% filter(Species=="setosa" | Species =="virginica") %>%
  select(Sepal.Length)
t.test(sep_setvin,conf.level=0.95)$conf.int
[1] 5.609428 5.984572
```

```
attr(,"conf.level")
[1] 0.95
```

Exercice 3 (IC pour un sondage)

Au cours d'une élection avec deux candidats A et B on réalise un sondage pour estimer la proportion p inconnue d'électeurs qui vont voter pour A. On interroge 1004 personnes, sur ces 1004 personnes 478 déclarent qu'elles vont voter pour A. Donner un intervalle de confiance à 90% pour le paramètre p.

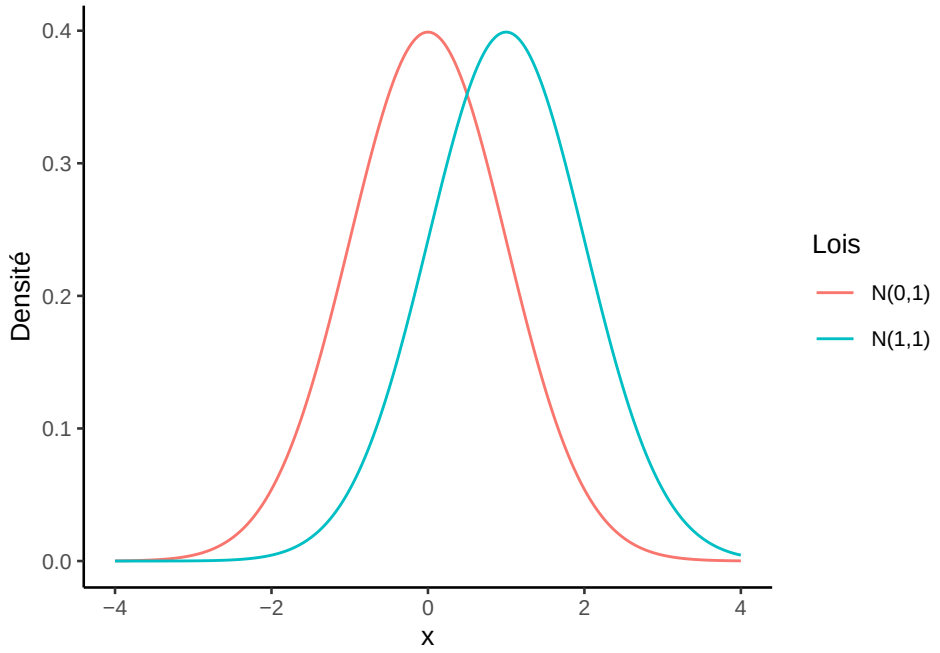
```
prop.test(478,1004,conf.level=0.90)$conf.int
[1] 0.4497733 0.5025488
attr(,"conf.level")
[1] 0.9
```

Exercice 4 (Graphes ggplot)

On utilisera les fonctions du package **ggplot** pour faire les graphes demandés.

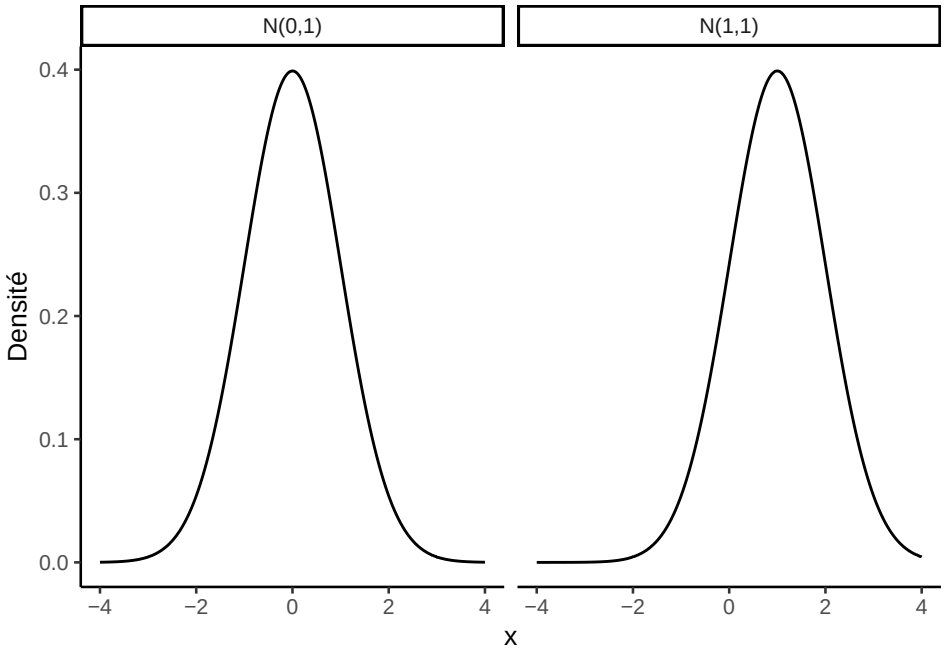
1. Tracer les densités gaussiennes des lois $N(0,1)$ et $N(1,1)$ sur un même graphe avec deux couleurs différentes (et une légende qui permet d'identifier les densités).

```
df <- tibble(x=seq(-4,4,by=0.001)) %>% mutate(`N(0,1)`=dnorm(x),
                                              `N(1,1)`=dnorm(x,1,1)) %>%
  pivot_longer(-x,names_to="Lois",values_to="Densité")
ggplot(df)+aes(x=x,y=Densité,color=Lois)+geom_line()
```



2. Même question mais sur deux graphes séparés (côte à côte).

```
ggplot(df)+aes(x=x,y=Densité)+geom_line()+facet_wrap(~Lois)
```



Exercice 5 (Données sur le covid)

Le jeu de données `data_covid_2020.csv` contient des informations sur les nombres de cas confirmés et de décès entre le 1er mars 2020 et le 21 novembre 2020 (quelques journées d'observations peuvent être manquantes pour certains pays). Il contient 7 colonnes :

- `date` : identifiant de la journée de l'observation
- `country` : pays
- `population` : la population totale du pays
- `confirmed` : nombre de cas positifs dans le pays `country` confirmés depuis le début de l'épidémie jusqu'à la date `date`
- `deaths` : nombre de décès dans le pays `country` depuis le début de l'épidémie jusqu'à la date `date`
- `deaths.day` : nombre de décès observé le jour `date` dans le pays `country`
- `confirmed.day` : nombre de cas positifs observés le jour `date` dans le pays `country`.

On souhaite calculer différents indicateurs et visualiser ces données. On utilisera les verbes **dplyr** et **ggplot** pour répondre aux questions suivantes.

1. Importer les données et afficher un résumé (fonction **summary**).

```
df1 <- read.csv("data_covid_2020.csv")
```

2. Convertir la première colonne du jeu de données (colonne **date**) en objet `date` à l'aide de la fonction `as_date` du package `lubridate`.

```
df1$date <- as_date(df1$date)
```

3. Combien de pays sont représentés dans le tableau ? On pourra utiliser le verbe **n_distinct**.

```
df1 %>% summarise(nb_pays=n_distinct(country))
  nb_pays
1      10
```

4. Afficher les pays présents dans l'étude.

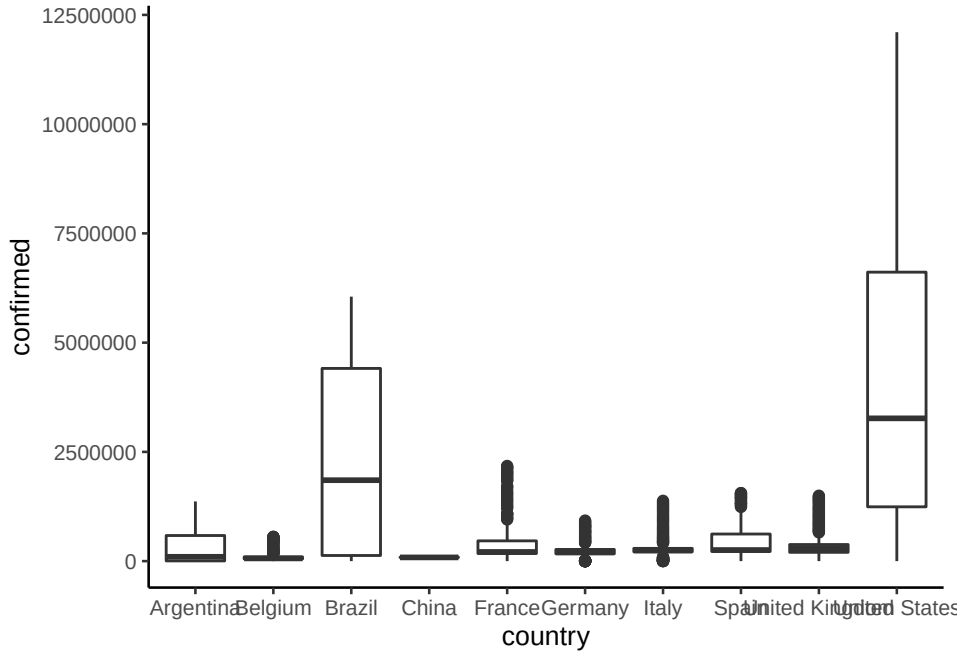
```
df1 %>% distinct(country)
           country
1        Argentina
2          Belgium
3           Brazil
4            China
5          Germany
6            Spain
7           France
8   United Kingdom
9            Italy
10  United States
```

5. Quel est le nombre de jours considéré dans ce tableau (nombre de jours entre la première observation et la dernière ?

```
df1 %>% summarize(nb_jours=n_distinct(date))
  nb_jours
1      266
```

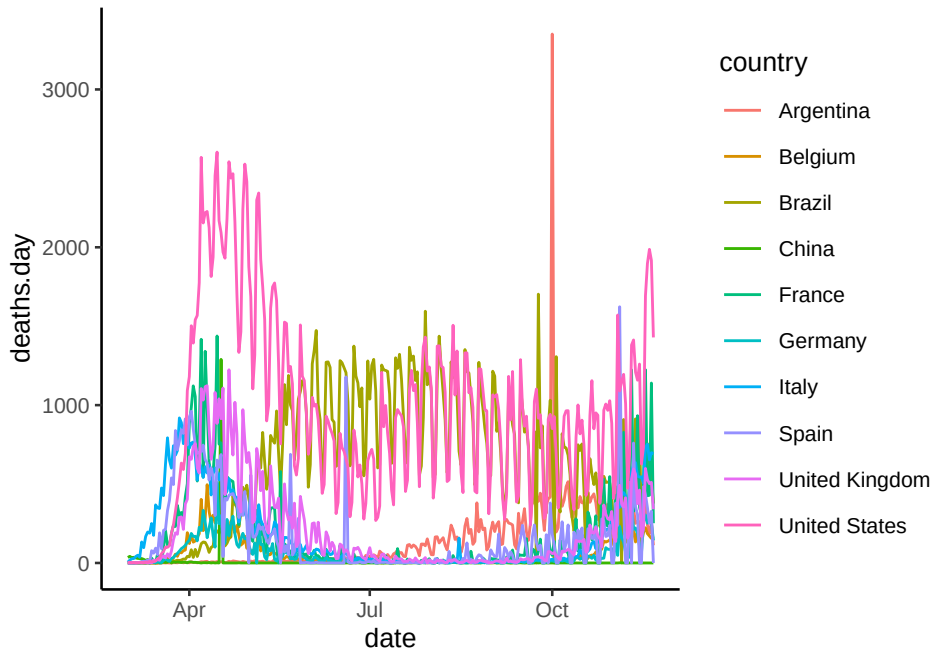
6. Comparer la distribution de la variable `confirmed` de chaque pays à l'aide d'un boxplot.

```
ggplot(df1)+aes(x=country,y=confirmed)+geom_boxplot()
```



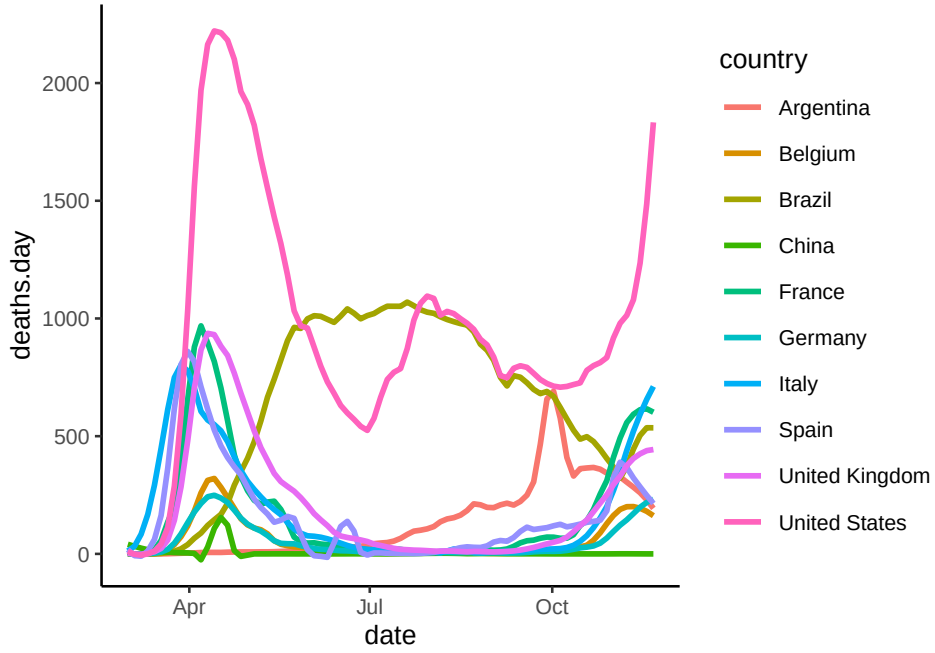
7. Tracer, pour chaque pays, les courbes représentant le nombre de morts par jours avec une couleur différente. Il s'agit de représenter la variable `deaths.day` sur l'axe des y en fonction de la `date` sur l'axe des x.

```
ggplot(df1)+aes(x=date,y=deaths.day,color=country)+geom_line()
```



8. Proposer une version plus lisse de ces courbes à l'aide de la fonction `geom_smooth`. On pourra utiliser l'option `span` de `geom_smooth` avec des petites valeurs si les courbes sont trop lissées.

```
ggplot(df1)+aes(x=date,y=deaths.day,color=country)+geom_smooth(span=0.1,se=FALSE)
```



9. On s'intéresse uniquement à la France. Calculer le nombre de cas confirmés moyen par jour (moyenne de la variable `confirmed.day`) ainsi que le nombre moyen de morts par jour pour toute la période observée. Les résultats devront être présentés sur un tableau à 1 ligne et deux colonnes.

```
df1 %>% filter(country=="France") %>% summarise(conf=mean(confirmed.day),deaths=mean(deaths.day))
      conf   deaths
1 8461.337 184.4419
```

10. Même question pour chaque pays. Les résultats devront cette fois être présentés sur un tableau à **p**

lignes et 3 colonnes (**p** représentant le nombre de pays).

```
df1 %>% group_by(country) %>% summarise(conf=mean(confirmed.day),deaths=mean(deaths.day))
# A tibble: 10 x 3
   country           conf deaths
   <chr>            <dbl>  <dbl>
 1 Argentina       5136.  139.
 2 Belgium         2099.   59.0
 3 Brazil         22755.  635.
 4 China             47.9   7.19
 5 France          8461.  184.
 6 Germany         3502.   53.4
 7 Italy           5223.  186.
 8 Spain           5946.  168.
 9 United Kingdom  5628.  206.
10 United States  45504.  962.
```

11. Ordonner les pays en fonction du nombre total de morts au 21 novembre 2020. On affichera sur une colonne le pays et sur l'autre le nombre total de morts. On pourra créer la date du 21 novembre à l'aide de

```
t1 <- ymd("2020-11-21")
```

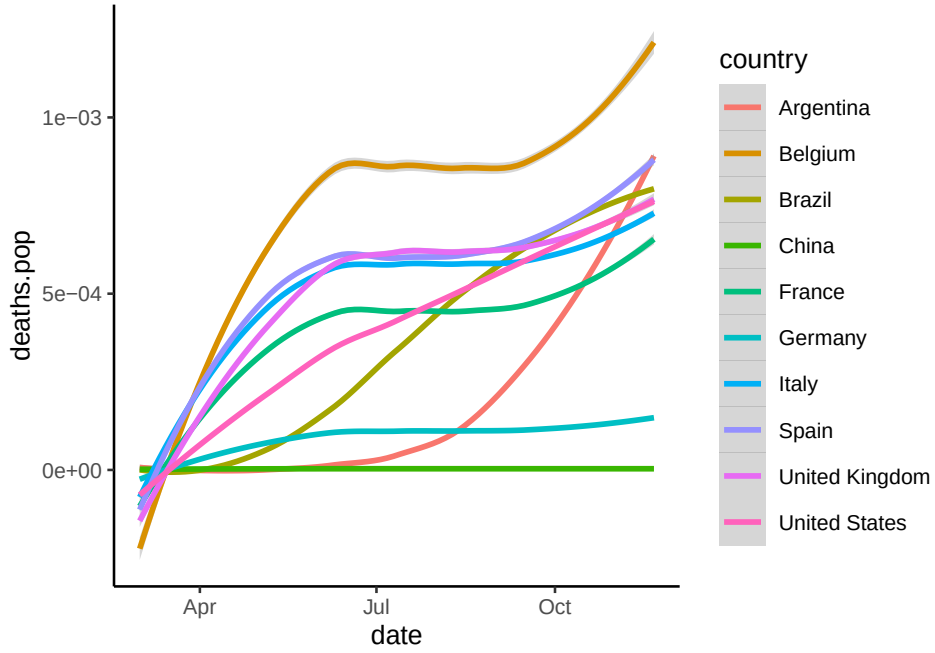
```
df1 %>% filter(date==t1) %>% arrange(desc(deaths)) %>% select(country,deaths)
          country deaths
1   United States 255946
2          Brazil 168989
3  United Kingdom  54721
4           Italy  49261
5          France  48593
6           Spain  42619
7       Argentina  36902
8         Belgium  15522
9         Germany  14061
10          China   4742
```

12. Ajouter aux données une variable `deaths.pop` égale au rapport du nombre de morts cumulé (`deaths`) divisé par la population (`population`).

```
df2 <- df1 %>% mutate(deaths.pop=deaths/population)
```

13. Représenter, pour chaque pays, les courbes représentant la variable `deaths.pop` avec une couleur différente.

```
ggplot(df2)+aes(x=date,y=deaths.pop,color=country)+geom_smooth()
```



14. Ordonner les pays en fonction de la variable `deaths.pop` au 21 novembre 2020. On affichera sur une colonne le pays et sur l'autre la variable `deaths.pop`.

```
df3 <- df2 %>% filter(date==t1) %>% arrange(desc(deaths.pop)) %>% select(country,deaths.pop)
df3
```

	country	deaths.pop
1	Belgium	1.351613e-03
2	Spain	9.053083e-04
3	Argentina	8.211628e-04
4	United Kingdom	8.187550e-04
5	Italy	8.169673e-04
6	Brazil	8.007078e-04
7	United States	7.797538e-04
8	France	7.246210e-04
9	Germany	1.691390e-04
10	China	3.392680e-06

15. On s'intéresse uniquement à la France. Calculer le nombre de morts par mois. On pourra créer une variable qui permet d'identifier le mois d'une observation à l'aide du verbe **separate**

```
df2 %>% separate(date,into=c("Year","Months","Days")) %>%
  filter(country=="France") %>%
  group_by(Months) %>% summarize(nb_morts=sum(deaths.day))
```

A tibble: 9 x 2

Months	nb_morts
<chr>	<int>
03	3530
04	19876
05	4487
06	934
07	434
08	372
09	1346
10	4840

```
9 11        11767
```

16. Même question pour tous les pays. On pourra visualiser les résultats dans un tableau à double entrée avec en ligne le pays et en colonne le mois de l'étude.

```
df2 %>% separate(date,into=c("Year","Months","Days")) %>%
  group_by(Months,country) %>% summarize(nb_morts=sum(deaths.day)) %>%
  pivot_wider(names_from=Months,values_from=nb_morts)
# A tibble: 10 x 10
   country         `03`  `04`  `05`  `06`  `07`  `08`  `09`  `10`  `11`
   <chr>          <int> <int> <int> <int> <int> <int> <int> <int> <int>
 1 Argentina         27   191   321   768  2236  5117  8277 14065  5900
 2 Belgium          705  6889  1873   280    94   171   121  1609  3897
 3 Brazil           201  5805 23308 30280 32881 28906 22571 15932  9105
 4 China            472  1328     1     3    20    62    16     0     3
 5 France          3530 19876  4487   934   434   372  1346  4840 11767
 6 Germany          775  5879  1917   450   158   156   192   988  3578
 7 Italy          12399 15539  5448  1336   374   342   411  2724 10643
 8 Spain           8464 15712  4502  1228    90   651  2697  4087  6741
 9 United Kingdom  2457 24297 10773  2952   795   315   644  4412  8076
10 United States   5271 60699 41703 20113 26306 29591 23515 23928 24819
```

17. **Question ouverte** : proposer des indicateurs numériques ou graphiques qui permettent de visualiser et/ou comparer les deux vagues dasn les pays européens.