

# Analyse varicelle

Marc Oudart

11/04/2020

## Préparation des données

Les données de l'incidence de la varicelle sont disponibles du site Web du Réseau Sentinelles. Nous les récupérons sous forme d'un fichier en format CSV dont chaque ligne correspond à une semaine de la période demandée. Nous téléchargeons toujours le jeu de données complet, qui commence en fin 1990 et se termine avec une semaine récente. L'URL est :

```
data_url = "https://www.sentiweb.fr/datasets/incidence-PAY-7.csv"
data_local = "C:/Users/Marc/Desktop/MOOC/mooc-rr/module3/exo2/incidence-PAY-7.csv"
```

## Téléchargement

Nous allons tester si le fichier local existe avec une condition `if`. Si non il sera téléchargé depuis l'url avec la fonction `download.file`. Si oui, il retournera Fichier déjà téléchargé

```
if (file.exists(data_local) == FALSE) {
  download.file(data_url, data_local)
} else {
  print("Fichier déjà téléchargé")
}
```

```
## [1] "Fichier déjà téléchargé"
```

Importer les données dans la variable `data` en sautant la première ligne de commentaire du fichier.

```
data = read.csv(data_url, skip = 1)
head(data)
```

```
##      week indicator   inc inc_low inc_up inc100 inc100_low inc100_up geo_insee
## 1 202014          7  3881   2223   5539      6         3         9         FR
## 2 202013          7  7341   5247   9435     11         8        14         FR
## 3 202012          7  8123   5790  10456     12         8        16         FR
## 4 202011          7 10198   7568  12828     15        11        19         FR
## 5 202010          7  9011   6691  11331     14        10        18         FR
## 6 202009          7 13631  10544  16718     21        16        26         FR
##      geo_name
## 1   France
## 2   France
## 3   France
## 4   France
## 5   France
## 6   France
```

Voir la fin du document :

```
tail(data)
```

```
##      week indicator   inc inc_low inc_up inc100 inc100_low inc100_up
## 1526 199102         7 16277  11046 21508     29         20         38
## 1527 199101         7 15565  10271 20859     27         18         36
## 1528 199052         7 19375  13295 25455     34         23         45
## 1529 199051         7 19080  13807 24353     34         25         43
## 1530 199050         7 11079   6660 15498     20         12         28
## 1531 199049         7  1143     0   2610     2          0          5
##      geo_insee geo_name
## 1526         FR  France
## 1527         FR  France
## 1528         FR  France
## 1529         FR  France
## 1530         FR  France
## 1531         FR  France
```

Regarder s'il y a des données manquantes :

```
lignes_na = apply(data, 1, function(x) any(is.na(x)))
data[lignes_na,]
```

```
## [1] week      indicator inc      inc_low  inc_up   inc100
## [7] inc100_low inc100_up geo_insee geo_name
## <0 rows> (or 0-length row.names)
```

Traverser data ligne par ligne (`data, 1`) et appliquer la fonction qui traverse colonne par colonne qui retourne tout (`any`) s'il y a valeur manquante (`is.na`).

`data[lignes_na,]` pour voir le contenu de ces lignes.

Super il n'y a pas de données manquantes. On peut directement travailler sur ce jeu de donnée.

Quelles types de données ?

Pour la colonne semaine et la colonne indice :

```
class(data$week)
```

```
## [1] "integer"
```

```
class(data$inc)
```

```
## [1] "integer"
```

Tous les deux sont des entiers donc pas de transformation à faire.

Il faut que R comprenne que la 1ère colonne sont des dates.

Il nous faut donc la librairie `parsedate`.

```
library("parsedate")
```

Création d'une fonction (`convert_week`) pour transformer la 1ère colonne en date.

```
convert_week = function(date){
  ws = paste(date)
  iso = paste0(substring(ws, 1, 4), "-W", substring(ws, 5, 6))
  as.character(parse_iso_8601(iso))
}
```

Appliquer la fonction à toute les lignes.

```
data$date = as.Date(apply(data$week, convert_week))
class(data$date)
```

```
## [1] "Date"
```

Trier le jeu de données par ordre chronologique.

```
data = data[order(data$date),]  
head(data)
```

```
##      week indicator   inc inc_low inc_up inc100 inc100_low inc100_up  
## 1531 199049         7  1143      0  2610      2          0          5  
## 1530 199050         7 11079    6660 15498     20         12         28  
## 1529 199051         7 19080   13807 24353     34         25         43  
## 1528 199052         7 19375   13295 25455     34         23         45  
## 1527 199101         7 15565   10271 20859     27         18         36  
## 1526 199102         7 16277   11046 21508     29         20         38  
##      geo_insee geo_name      date  
## 1531          FR  France 1990-12-03  
## 1530          FR  France 1990-12-10  
## 1529          FR  France 1990-12-17  
## 1528          FR  France 1990-12-24  
## 1527          FR  France 1990-12-31  
## 1526          FR  France 1991-01-07
```

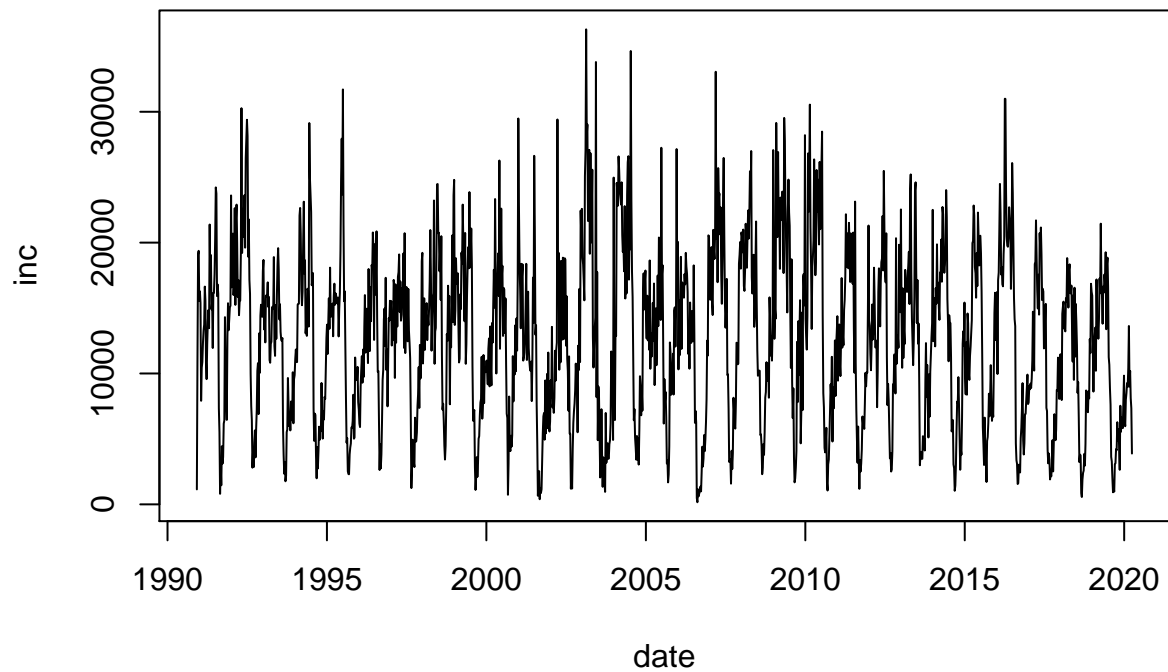
Est-ce que les lignes sont bien séparées d'1 semaine ?

```
all(diff(data$date) == 7)
```

```
## [1] TRUE
```

On peut plotter les données :

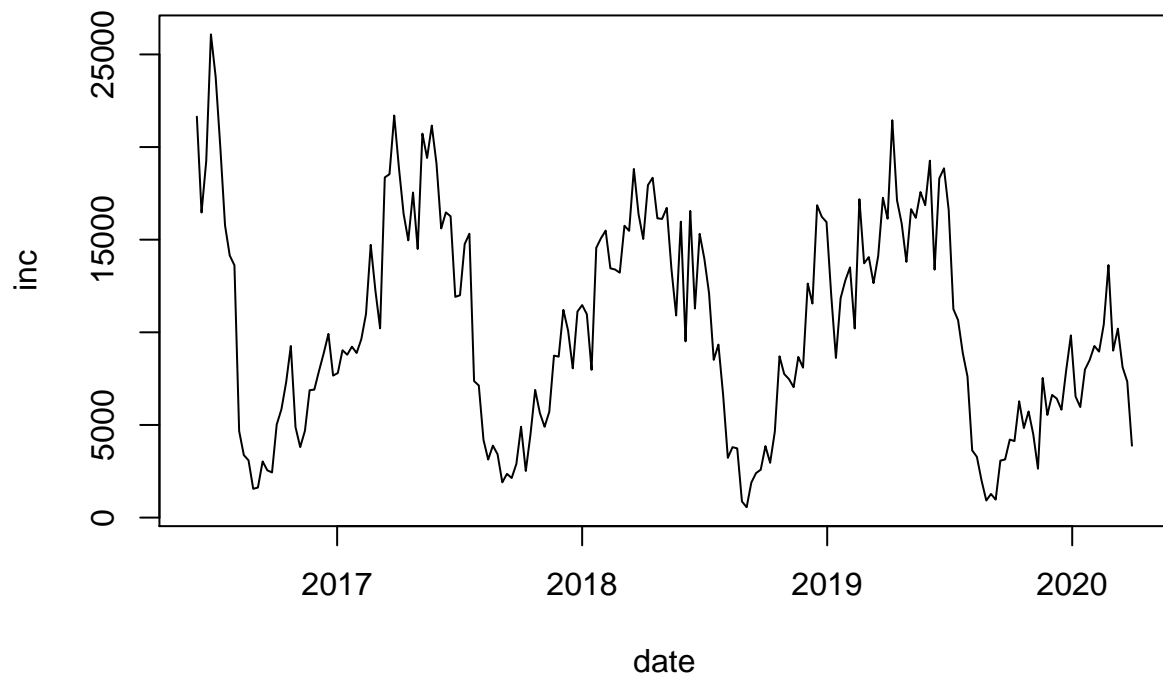
```
with(data, plot(date, inc, type = "l"))
```



Le `with` spécifie juste que c'est de la variable `data` qu'il faut prendre (au lieu d'utiliser `data$date`).

Appliquer seulement aux 200 derniers points :

```
with(tail(data, 200), plot(date, inc, type = "l"))
```



L'incidence de la varicelle semble suivre un cycle dont la largeur des pics est vraiment plus grand que ceux de la grippe. C'est pas uniquement en hiver. Il y a cependant un creux aux 2/3 de l'année vers septembre. Ce mois semble tout indiqué pour séparer nos années.

## L'analyse

On va analyser d'années en années et non pas de semaine en semaine.

On va faire des années du 1er septembre au 1er septembre de chaque année pour ne pas tomber au milieu d'un pic.

On va faire une fonction pour ça :

```
pic_annuel = function(annee){
  debut = paste0(annee-1, "-09-01")
  fin = paste0(annee, "-09-01")
  semaines = data$date > debut & data$date <= fin
  sum(data$inc[semaines], na.rm = TRUE)
}
```

L'année 1990 commence en fin d'année donc dans le pic donc il vaut mieux prendre des années à partir de 1991. 2020 n'est pas encore arrivé à septembre donc on va ignorer cette année là aussi.

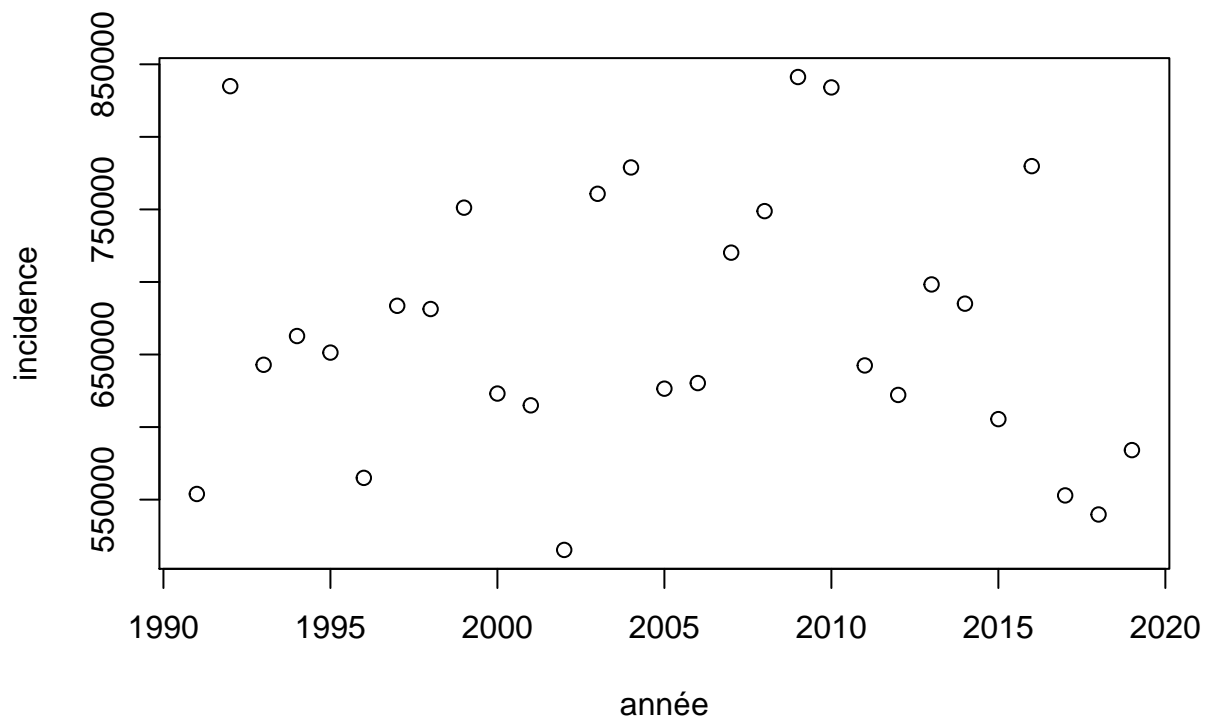
```
annees = 1991:2019
```

On va créer un nouveau tableau des incidences par année.

```
incidence_annuelle = data.frame(année = annees, incidence = sapply(annees, pic_annuel))
head(incidence_annuelle)
```

```
##   année incidence
## 1  1991   553895
## 2  1992   834935
## 3  1993   642921
## 4  1994   662750
## 5  1995   651333
## 6  1996   564994
```

```
plot(incidence_annuelle, type="p")
```



Voir les pics les plus importants en faisant un tri par l'incidence.

```
head(incidence_annuelle[order(-incidence_annuelle$incidence),])
```

```
##   année incidence
## 19  2009   841233
## 2   1992   834935
## 20  2010   834077
## 26  2016   779816
## 14  2004   778914
## 13  2003   760765
```

L'année avec le plus fort pic est l'année 2009.

Voir les pics les moins importants en faisant un tri inverse par l'incidence.

```
head(incidence_annuelle[order(incidence_annuelle$incidence),])
```

```
##   année incidence
```

```
## 12 2002 515343
## 28 2018 539765
## 27 2017 552906
## 1 1991 553895
## 6 1996 564994
## 29 2019 584116
```

L'année 2002 a eu le moins de cas.

Histogramme de la fréquence des incidences en 10 catégories.

```
hist(incidence_annuelle$incidence, breaks=10)
```

