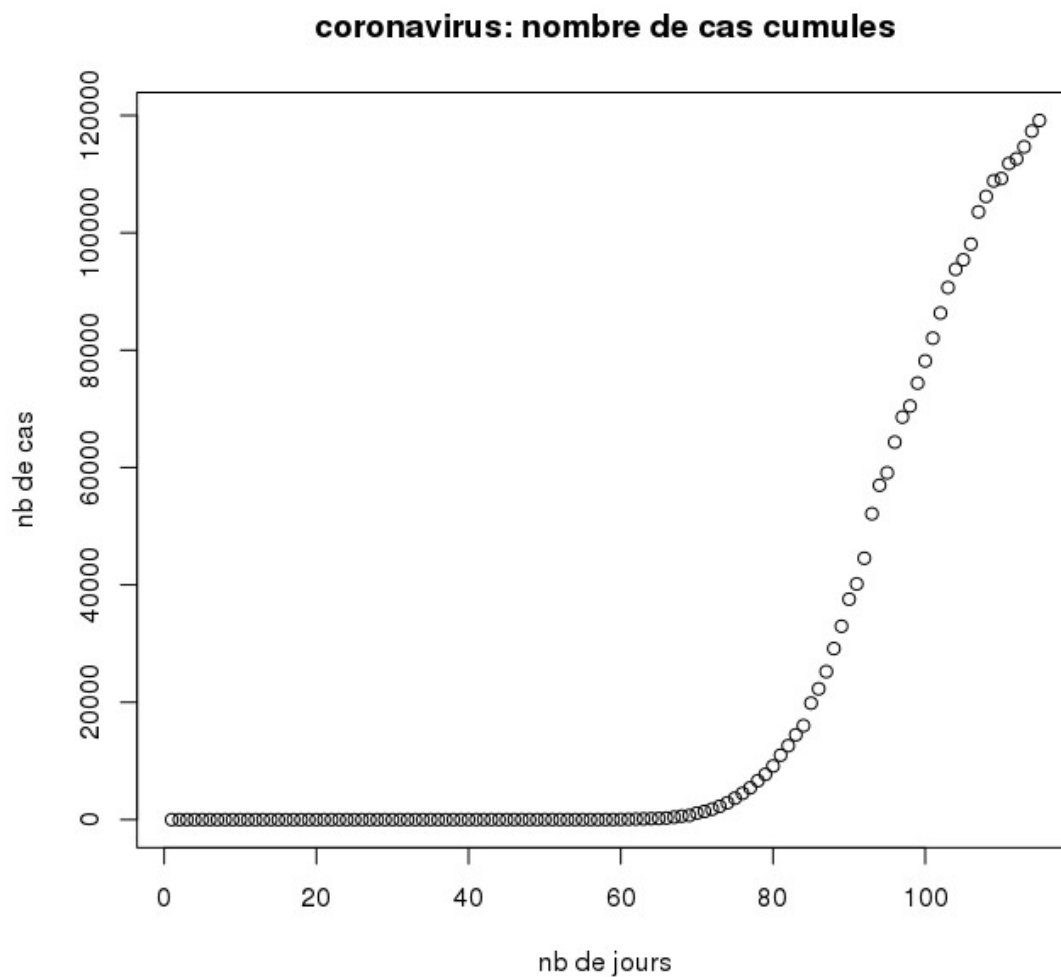


Un exemple de visualisation des données du Coronavirus avec R



Préambule

Beaucoup d'utilisateurs souhaitent suivre l'évolution des données du Coronavirus, et utilisent de ce fait des applications téléchargées qui contiennent pour beaucoup des malwares. L'objet de cette publication est de fournir aux utilisateurs de R le moyen de visualiser les données du Coronavirus avec les ressources standard de R, sans téléchargement de ressources additionnelles particulières, afin d'en permettre la visualisation.

Le présent document sera diffusé sur le site de developpez.com. Vous y trouverez également des tutoriels R en Français qui vous permettront de vous initier au langage R.

Les scripts ci-après ont été testés avec Linux. Néanmoins, R sous Windows acceptera la plupart des instructions qui suivent.

Chargement des données mondiales

L'URL retenue est la suivante :

<https://opendata.ecdc.europa.eu/covid19/casedistribution/csv>

Elle permet de télécharger un fichier au format csv qui donne des statistiques sur le nombre de cas (cases) et le nombre de décès (death) au niveau mondial.

Vous renommerez le fichier téléchargé en coronavirus_data.csv, et ce sera le nom de fichier csv qui sera utilisé dans la suite de cette étude.

Placez vous ensuite dans R, et exécutez ;

```
donnees_corona ← read.csv("coronavirus_data.csv");
```

Pour vérifier le chargement , vous aurez tendance à taper

```
donnees_corona ;
```

et il se peut que la taille du fichier téléchargé dépasse le nombre de lignes de max.print

vous pouvez modifier max.print par

```
options(max.print="200000");
```

le problème devrait être corrigé, jusqu'à la nouvelle limite.

Vous pouvez maintenant vous renseigner sur le contenu de votre téléchargement.

```
ls(donnees_corona);
```

```
typeof(donnees_corona$dateRep);
```

```
str(donnees_corona);
```

Vous pouvez constater que le format date n'est pas très satisfaisant.

On va donc forcer le type de la date au chargement en champs caractères :

```
donnees_corona ← read.csv("coronavirus_data.csv", colClasses=c("dateRep"="character"));
```

puis on va ajouter une colonne au format Date, en signalant à R que les données diffusées sont au format jj/mm/yyyy pour faciliter la conversion

```
donnees_corona$dateRep2<-as.Date(donnees_corona$dateRep, "%d/%m/%y");
```

Sélection des données pour la France

Affichage des dates pour la France

Ceci se fait avec l'instruction

```
donnees_corona$dateRep2[which(donnees_corona$countriesAndTerritories=="France")];
```

Ceci permet d'afficher les dates disponibles pour la France, parmi les données mondiales (contenues dans la variable `donnees_corona`)

Construction d'un data.frame pour l'étude du cas France

Pour cela, on va générer une séquence de nombres pour initialiser le data.frame à une dimension convenable avec la fonction `seq()` de R

```
l0<-length(donnees_corona$dateRep[which(donnees_corona$countriesAndTerritories=="France")])
```

```
donnees_france ← data.frame(seq(l0))
```

on ajoute ensuite les colonnes qui nous intéressent :

```
donnees_france$dateRep<-donnees_corona$dateRep[which(donnees_corona$countriesAndTerritories=="France")];
```

```
donnees_france$cases<-donnees_corona$cases[which(donnees_corona$countriesAndTerritories=="France")];
```

```
donnees_france$death ← donnees_corona$deaths[which(donnees_corona$countriesAndTerritories=="France")];
```

on peut visualiser le résultat obtenu

```
edit(donnees_france);
```

on va aussi générer une autre séquence, mais en ordre décroissant dans une colonne num :

```
donnees_france$num<-seq(length(donnees_france$death),by=-1);
```

```
edit(donnees_france);
```

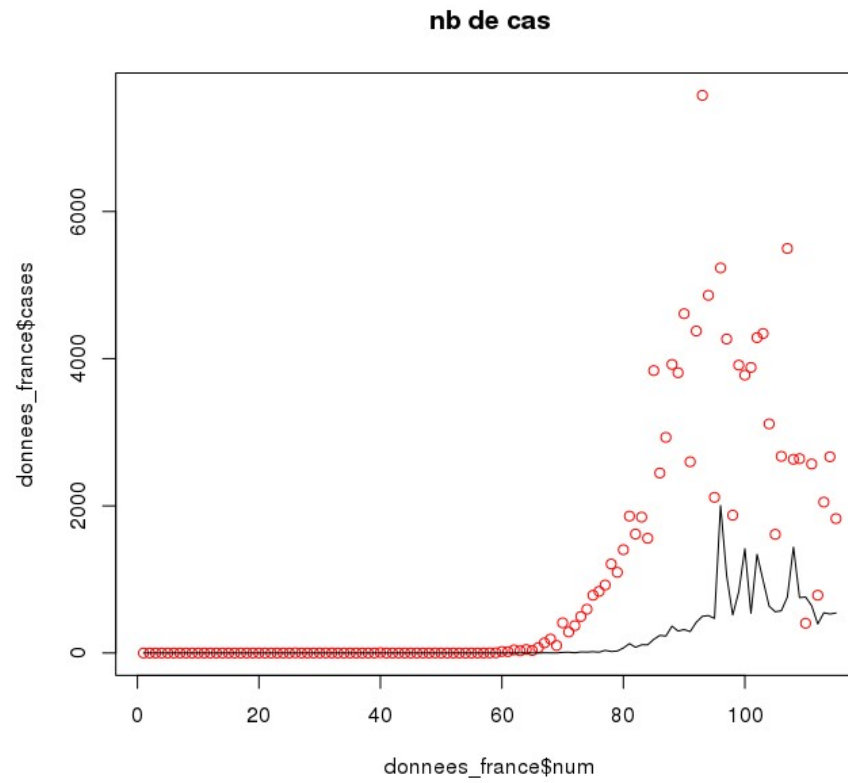
Réalisation du graphique

Par `plot.new` on provoque l'affichage d'une fenêtre graphique où seront imprimés les résultats. Ensuite, on va remplir la fenêtre avec des graphiques, à l'aide des instructions `plot` et `lines`. Le détail des fonctions graphiques disponibles peut être obtenu par `library(help = "graphics")`, et une fonction peut être interrogée individuellement (`help(plot)` par exemple)

```
plot.new();
```

```
plot(donnees_france$num,donnees_france$cases,col="red",main="nb de cas");
```

```
lines(donnees_france$num,donnees_france$death,col="black",main="nb de deces");
```



Calcul de données cumulées

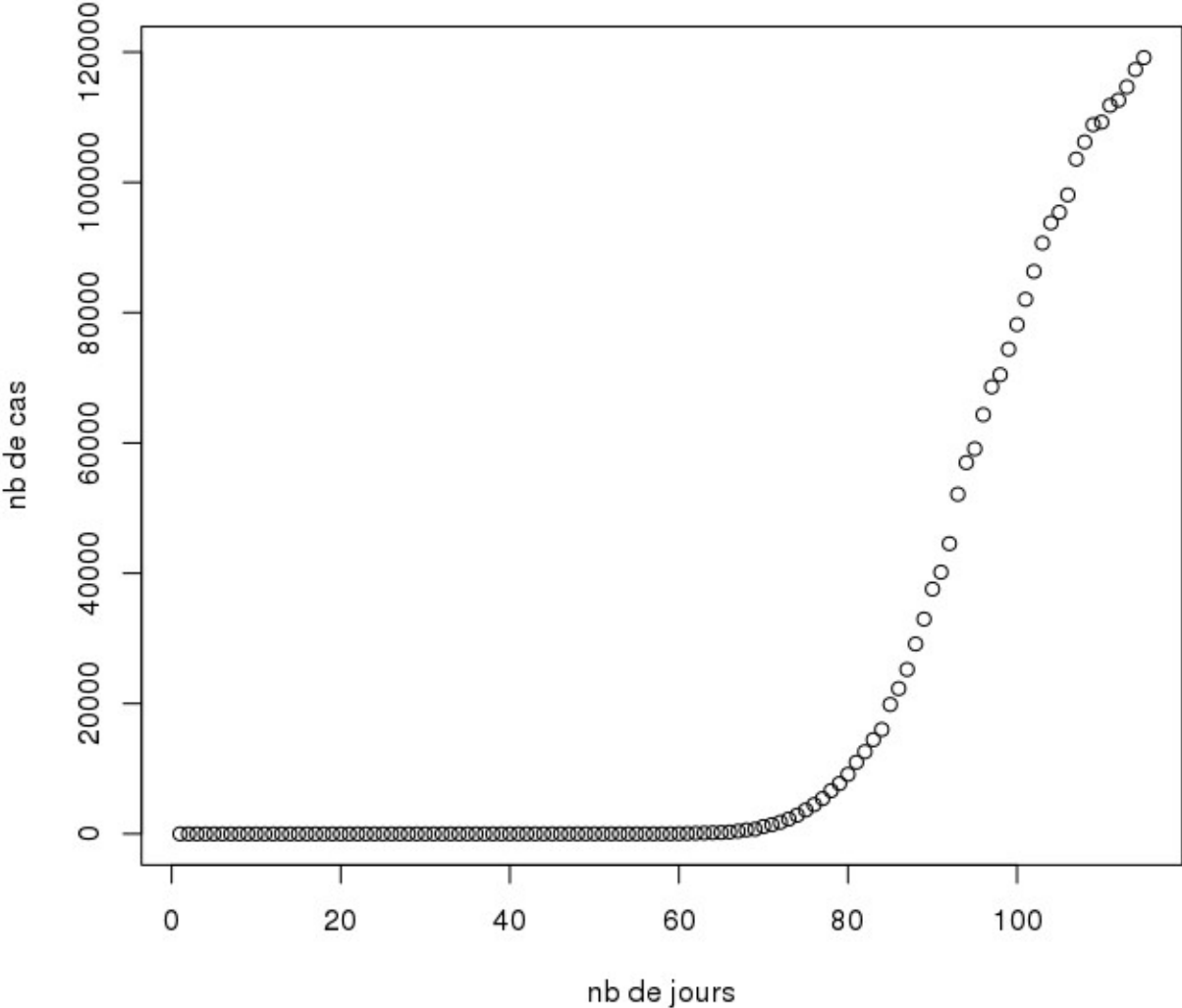
Pour cela, on peut s'aider de deux fonctions (faire un copier coller de la totalité de la fonction pour ne pas générer d'erreurs). `gl_reverse` permet d'afficher une colonne de `data.frame` en ordre inverse, et `gl_cumul` permet de calculer une colonne cumulée

```
# representation graphique de donnees cumulees
```

```
gl_reverse <-function(X) {  
  # X colonne,  
  j=length(X)  
  gl_r<-c(0)  
  for(i in 1:j)  
    {gl_r[j-i+1]<-X[i]}  
  gl_r  
};
```

```
gl_cumul <-function(X) {  
# X colonne,  
j=length(X)  
gl_r<-c(0)  
gl_r[1]=0  
for(i in 2:j)  
{gl_r[i]<-gl_r[i-1]+X[i]}  
gl_r  
};  
  
cumul_cas_france<-gl_cumul(gl_reverse(donnees_france$cas))  
cumul_cas_france  
  
on va ensuite faire un graphique  
plot(cumul_cas_france,  
      main="coronavirus: nombre de cas cumules",  
      xlab="nb de jours",  
      ylab="nb de cas"  
);
```


coronavirus: nombre de cas cumules



Conclusion

R permet la visualisation assez simple de données téléchargées au format csv.

La visualisation directe des données permet de se forger une opinion à partir des observations – et ici de conclure à un succès provisoire et précaire de la politique de confinement. Certains statisticiens espèrent à partir des données cumulées évaluer le nombre final de cas de contaminés, en utilisant la fonction logistique, ainsi que les dates de fin d'épidémies (le 15 mai pour l'Espagne?) Je vous laisse donc vous pencher sur cet exercice de style.

Le nom de l'auteur restera partiellement masqué. C'est très certainement le meilleur moyen d'éviter les polémiques, mais vous pouvez me retrouver sur developpez.com, pseudo gerard093. Bon surf !!!