1 - Régression linéaire

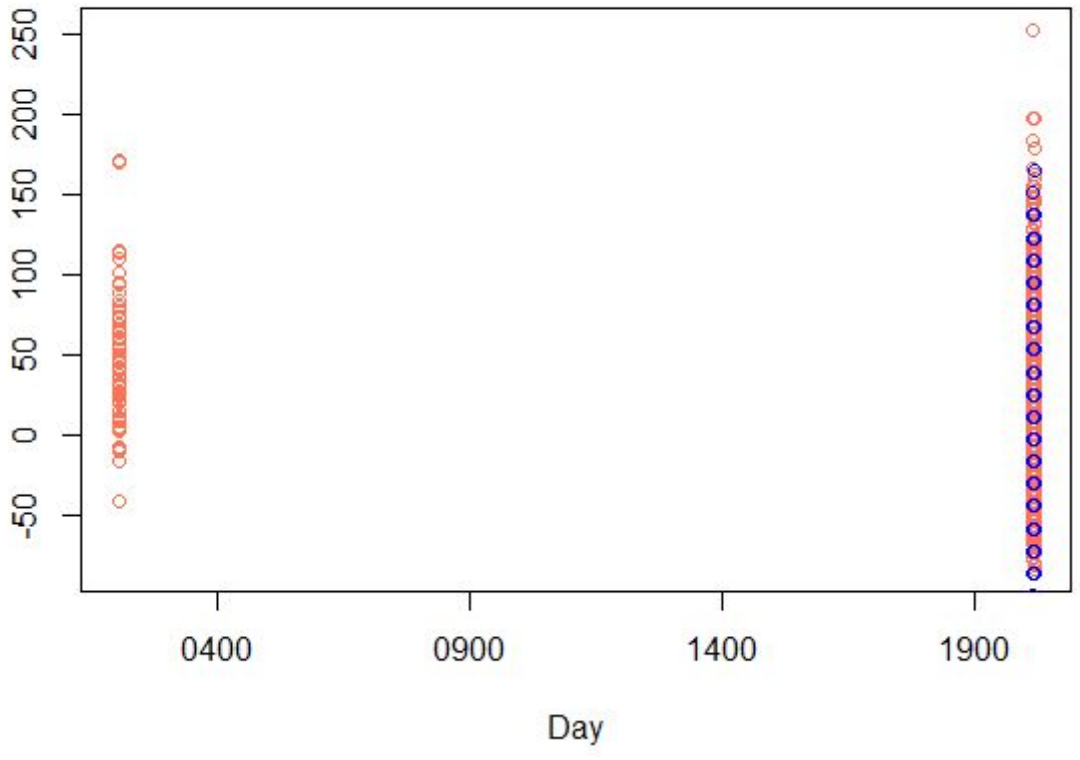
Ces group by sont nécessaires pour calculer le nombre total de patients pour chaque date et pour calculer la valeur moyenne puis pour créer un set de nombres pour comparer

2 - Arbre de décision
A - Visualisation des données

La visualisation des données est importante dans un premier temps. Après avoir fusionné les données, il s'agit de la seconde étape afin de pouvoir réaliser un arbre de décision et le clusturing des données. Tout d'abord, nous pouvons voir graphiquement s'il semble y avoir des corrélations et de plus, cela permettra de différencier des différentes variables importantes pour les analyses.

```
plot(as.Date(environmentDataNO$DAY_), environmentDataNO$PERCENT_, col="coral1", xlab = "Day", ylab = "")
points(as.Date(patientsData2$DAY_),patientsData2$DIFFERENCE,col="blue")
abline(modelNO)
```

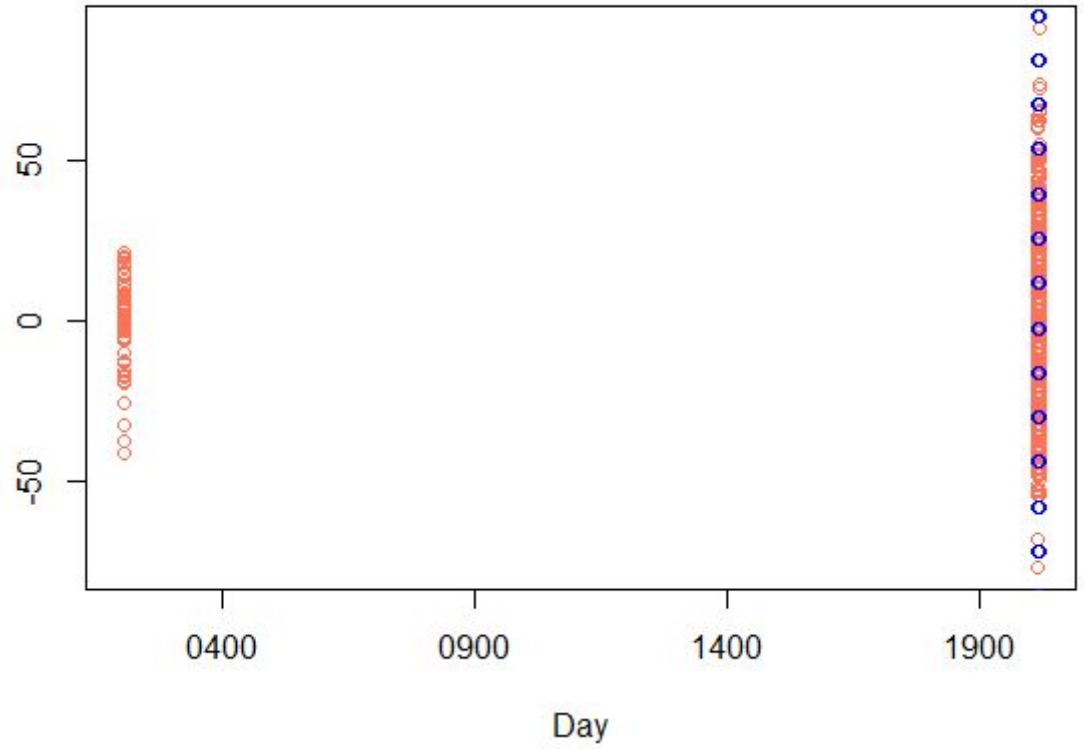
plot() permet de dessiner des points sur le graphique.
Nous dessinons donc ici le taux de pollution par jour et son pourcentage (en couleur corail)
Puis points() permet de rajouter les points correspondants aux données des patients. Ici, les données relatives au jour et à la "différence" (valeur - average / valeur) (en couleur bleue)
abline() sera utilisé par la suite pour ajouter la régression linéaire au graphique.



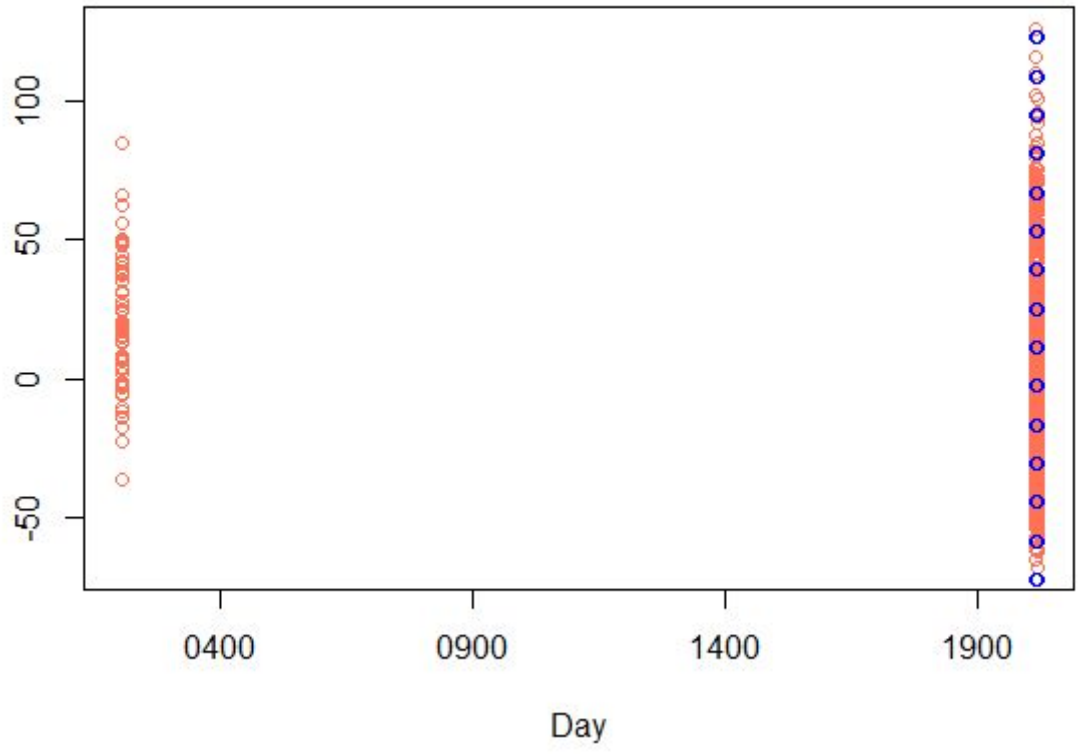
Pollutant : NO

Ici, nous voyons donc que les données des patients se sont ajoutées à un seul endroits à celles du pollutant NO. En revanche, aucune interprétation n'est possible : l'échelle semble totalement erronée
Peut-être les données sélectionnées ne sont pas pertinentes : à refaire !!

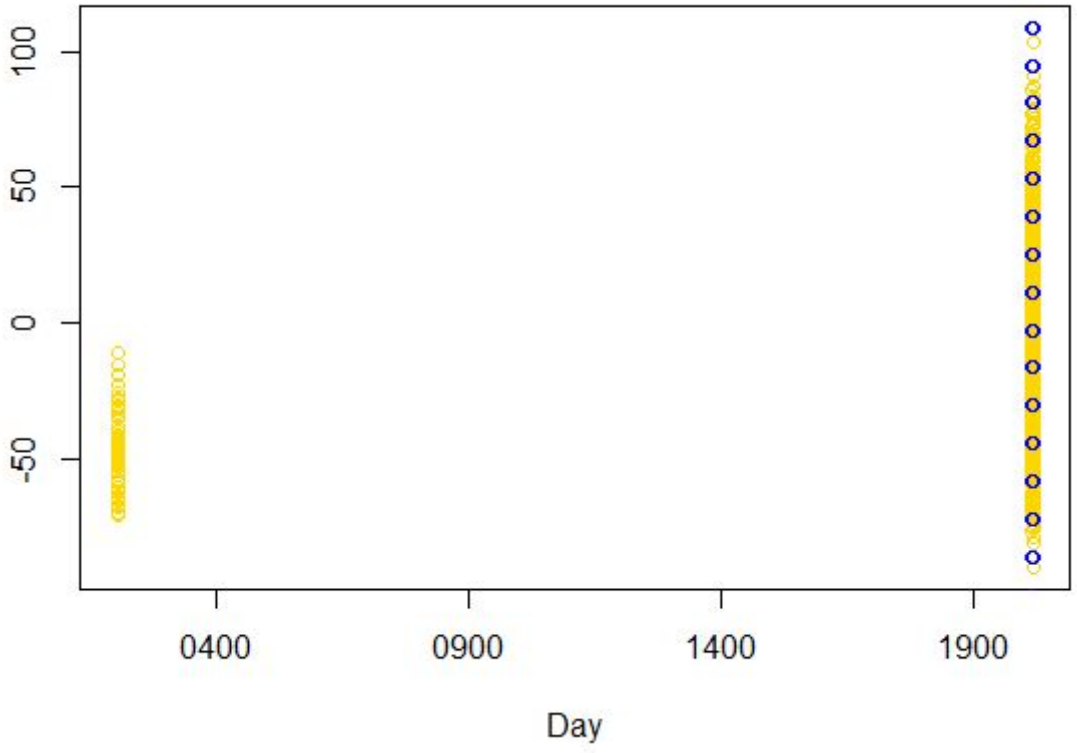
Par la suite, nous faisons les mêmes calculs pour les autres polluants



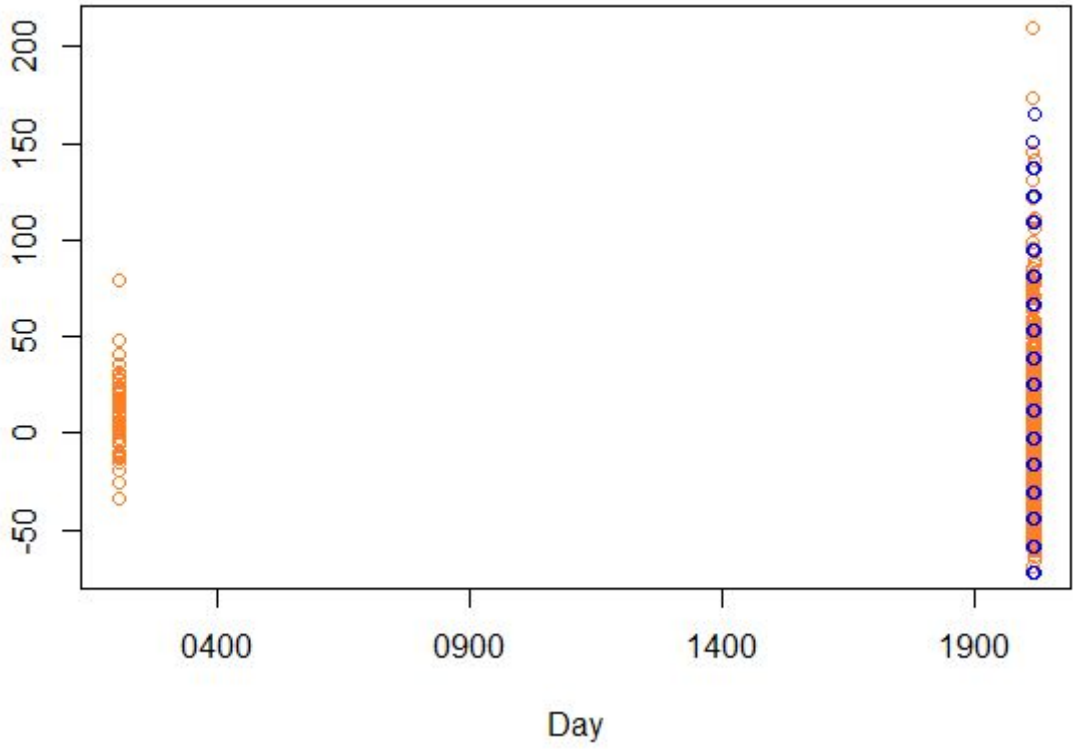
Pollutant : NO2



Pollutant NOX



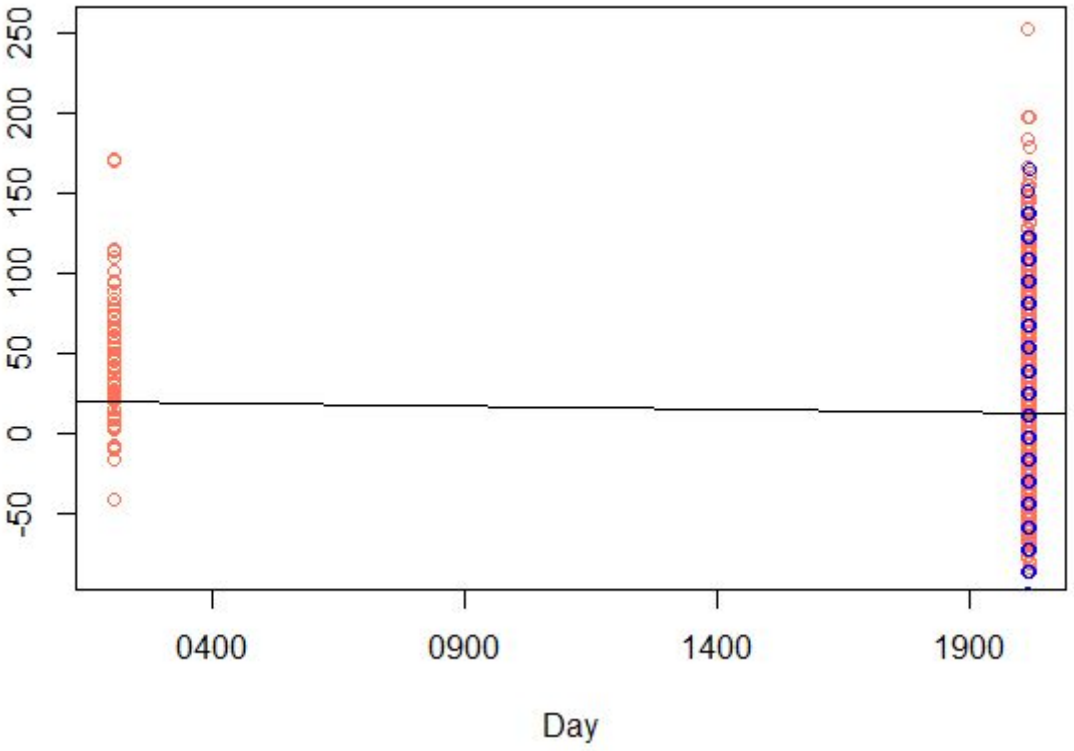
Pollutant O3



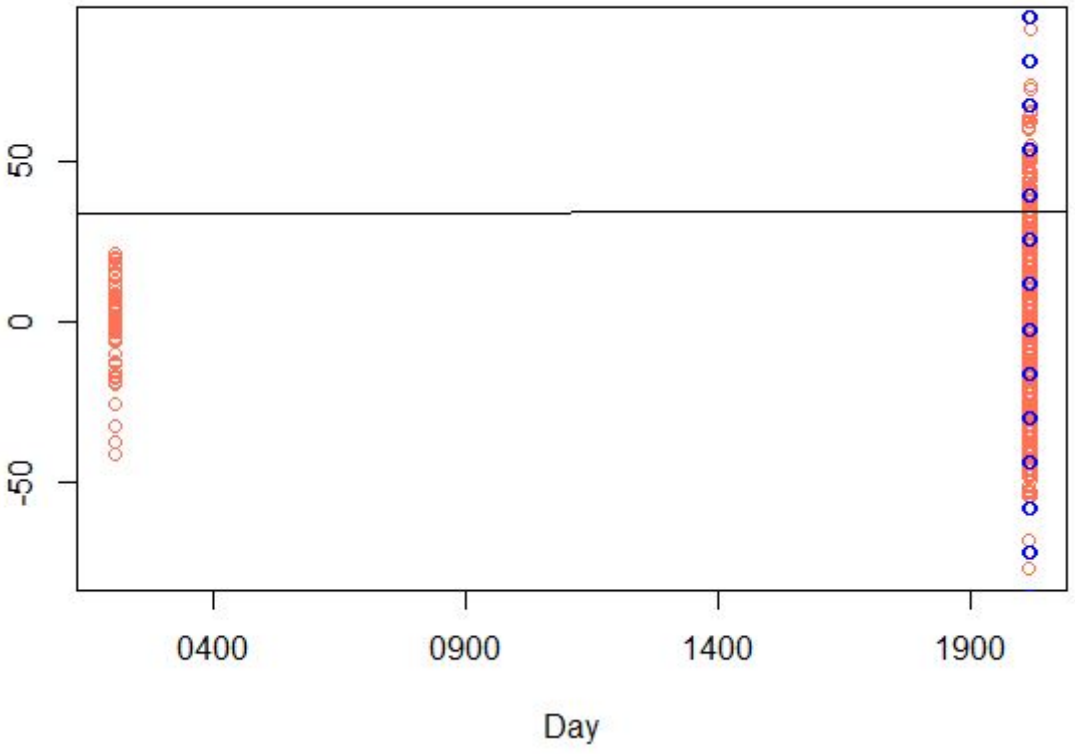
Pollutant PM10
On ajoute la régression linéaire (fonction lm)
Ici, abline() permet d'ajouter la régression linéaire.

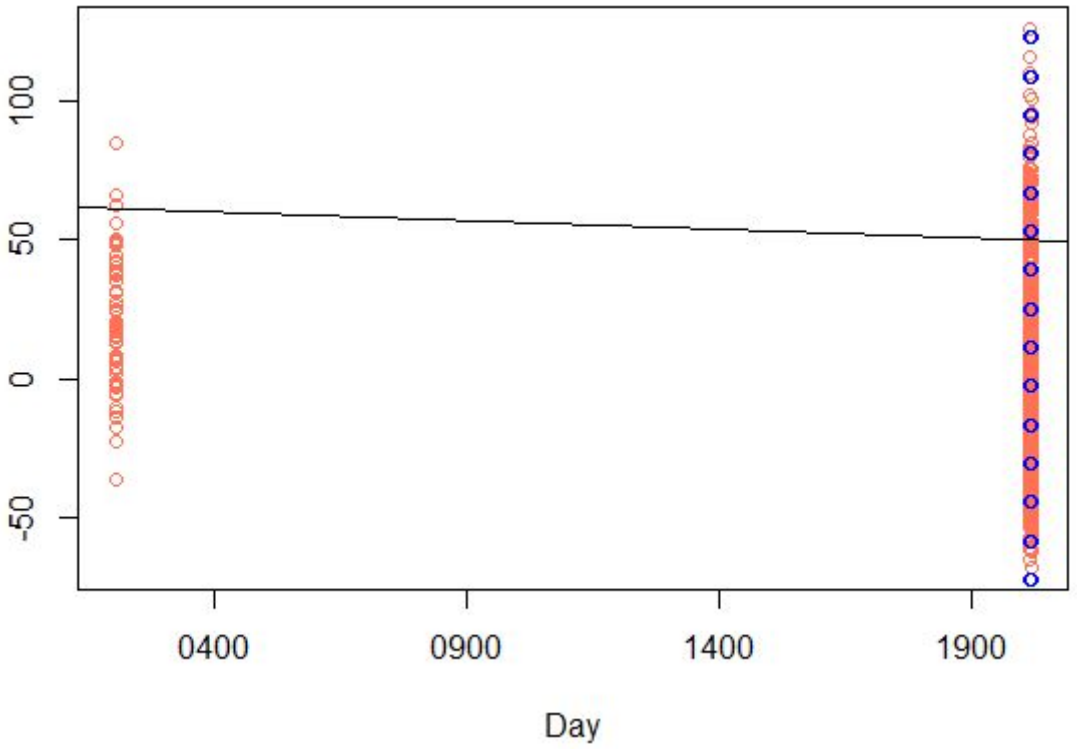
```
modelNO <- lm(SR_ZNACH ~ as.Date(DAY_), data=environmentDataNO)
```

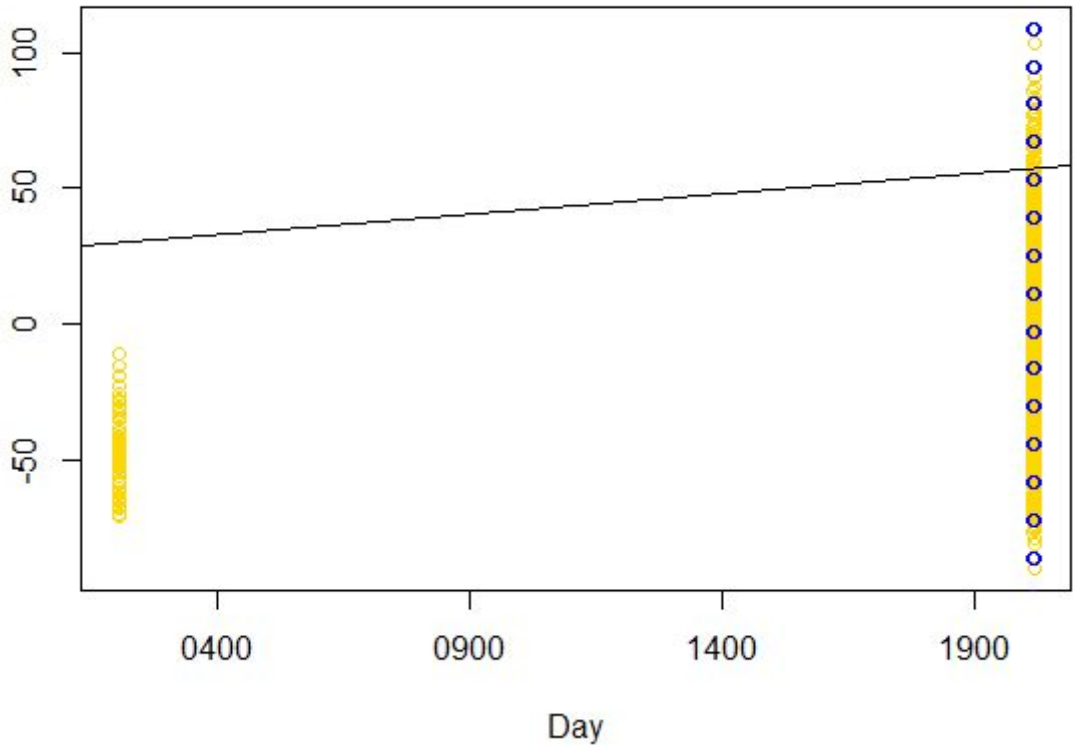
Ici, nous faison le modèle de régression linéaire pour le pollutant NO.
Nous l'ajoutons au graphique avec la fonction abline(modelNO)

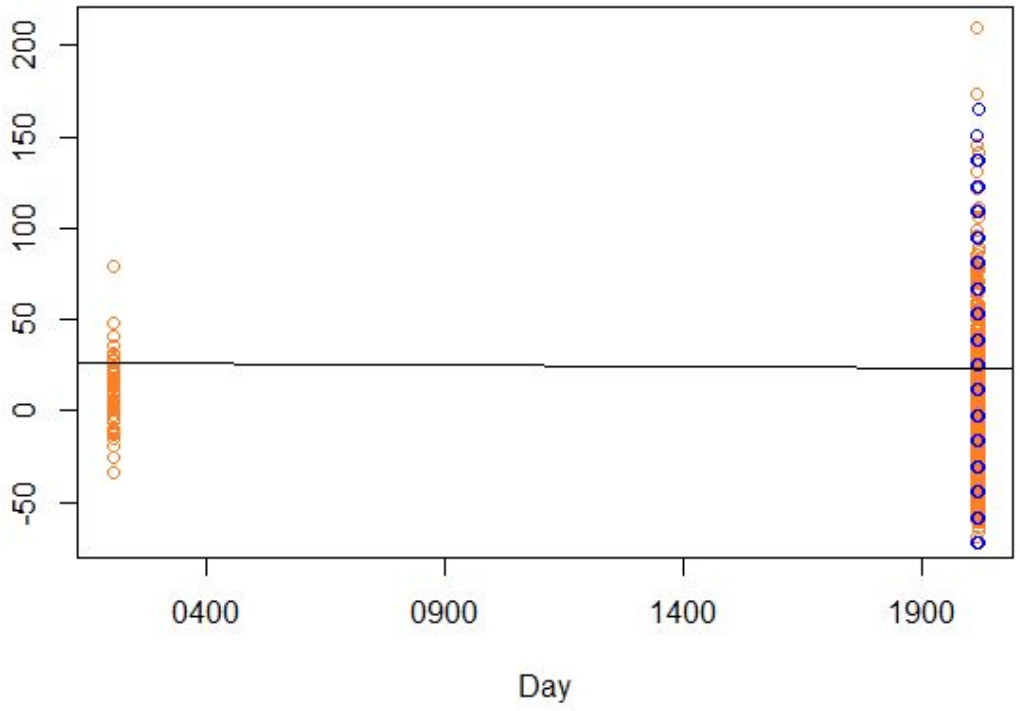


Ici, les résultats ne sont toujours pas bons pour être interprétés car il est impossible de comprendre ce à quoi cela correspond.




100
50
0
-50
0400
0900
1400
1900
Day


100
50
0
-50
0400
0900
1400
1900
Day

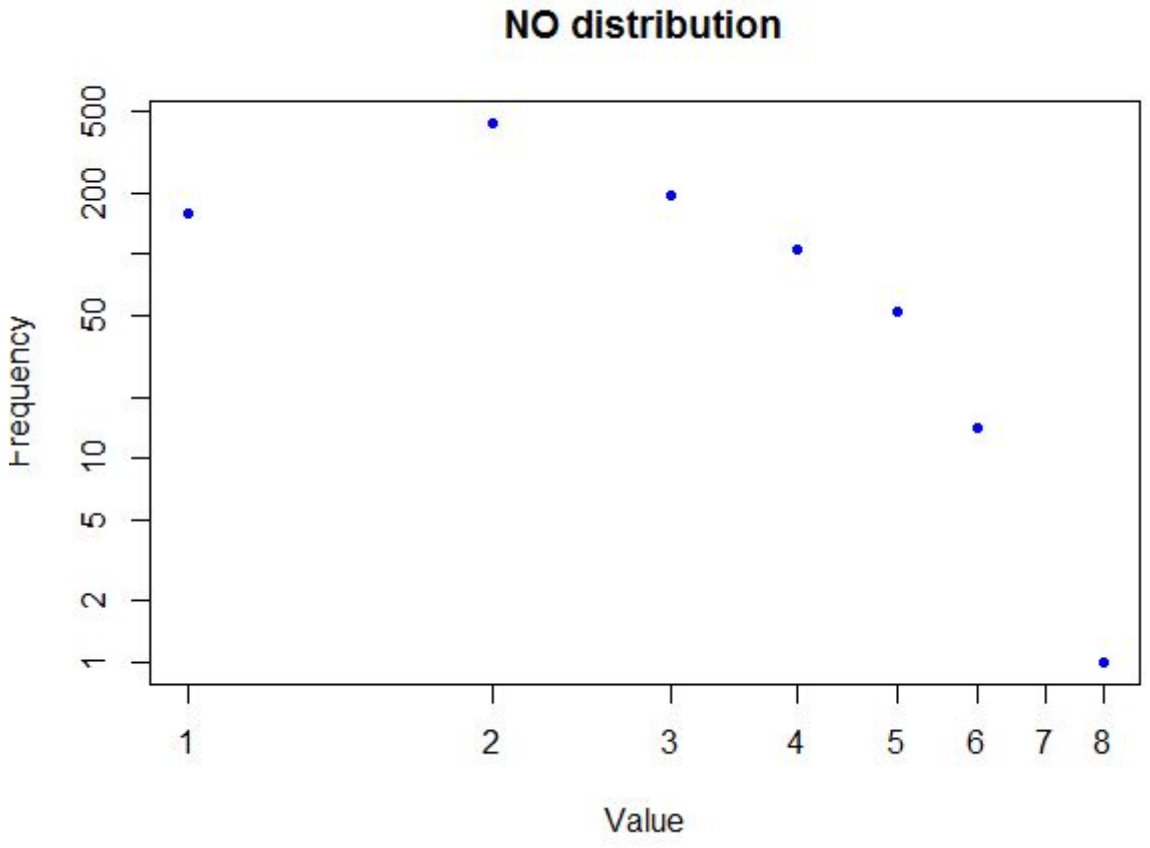


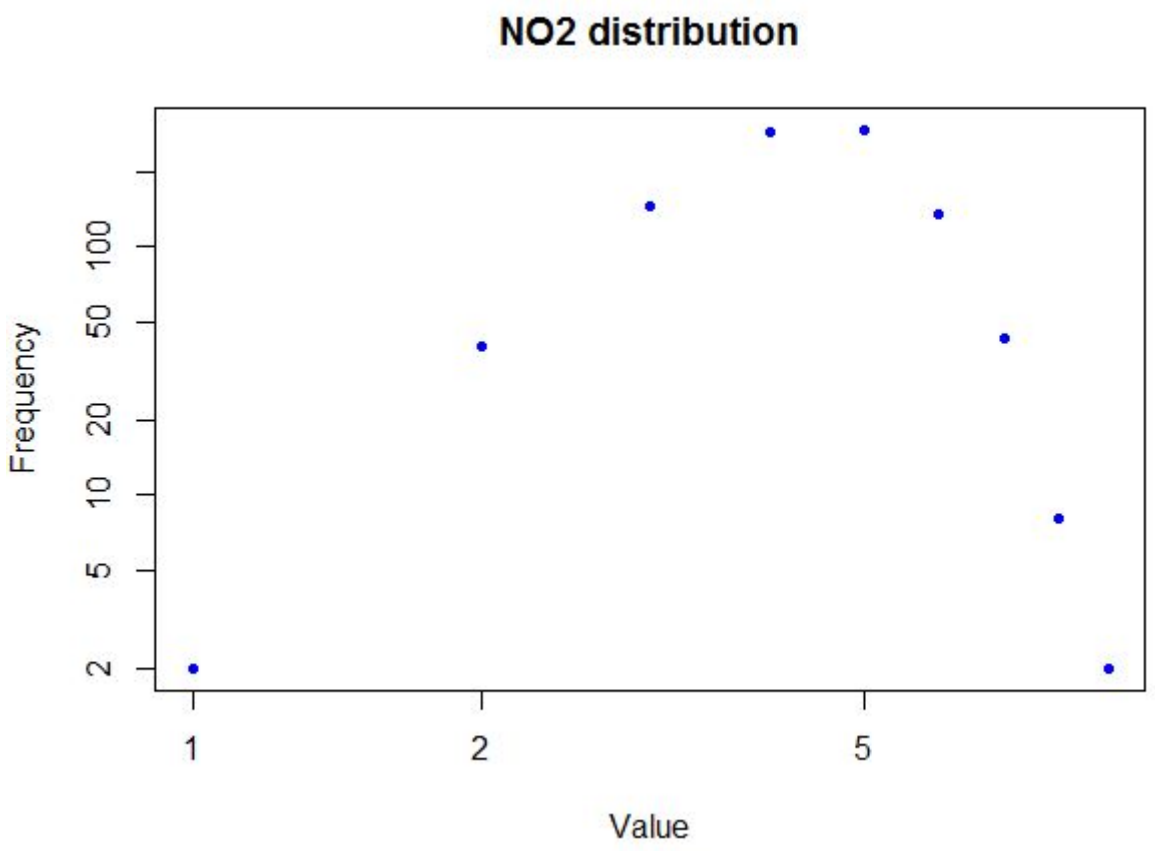
Il va falloir vérifier ces résultats. On ne sait pas à quoi correspondent les échelles

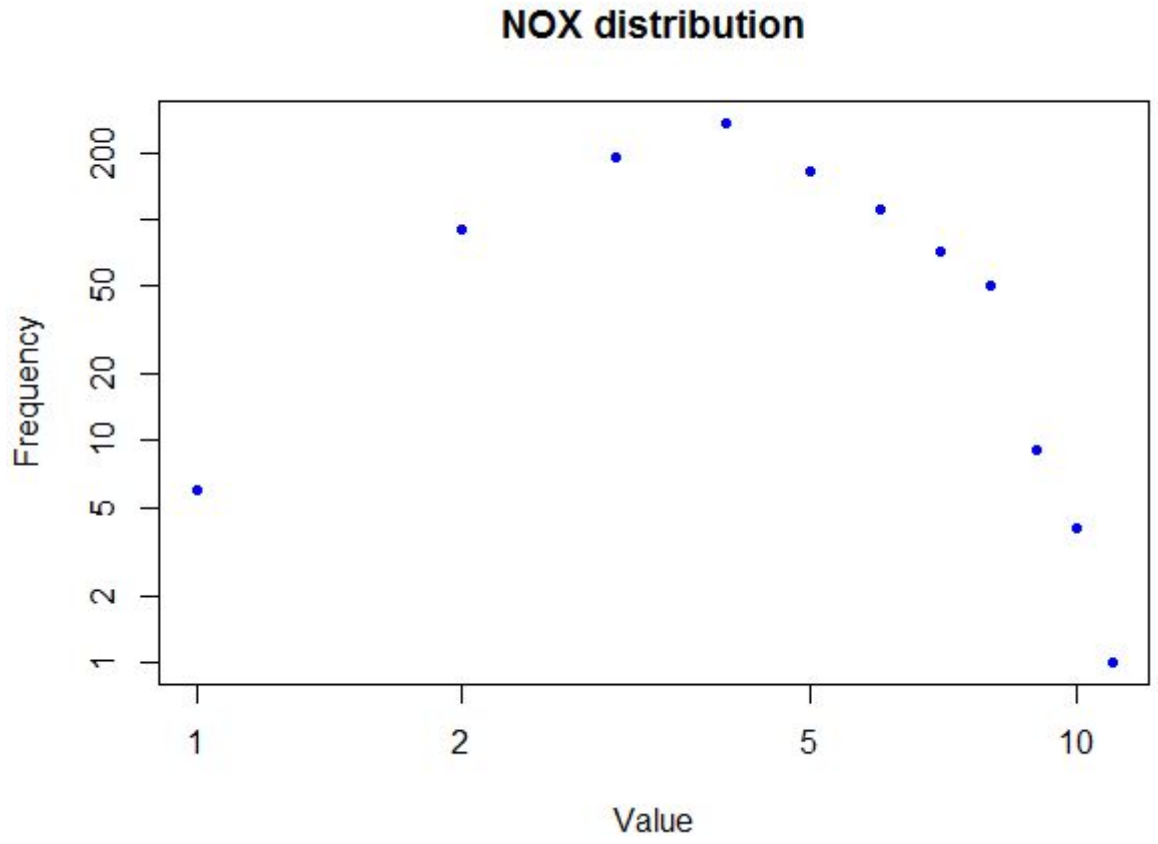
Data distribution

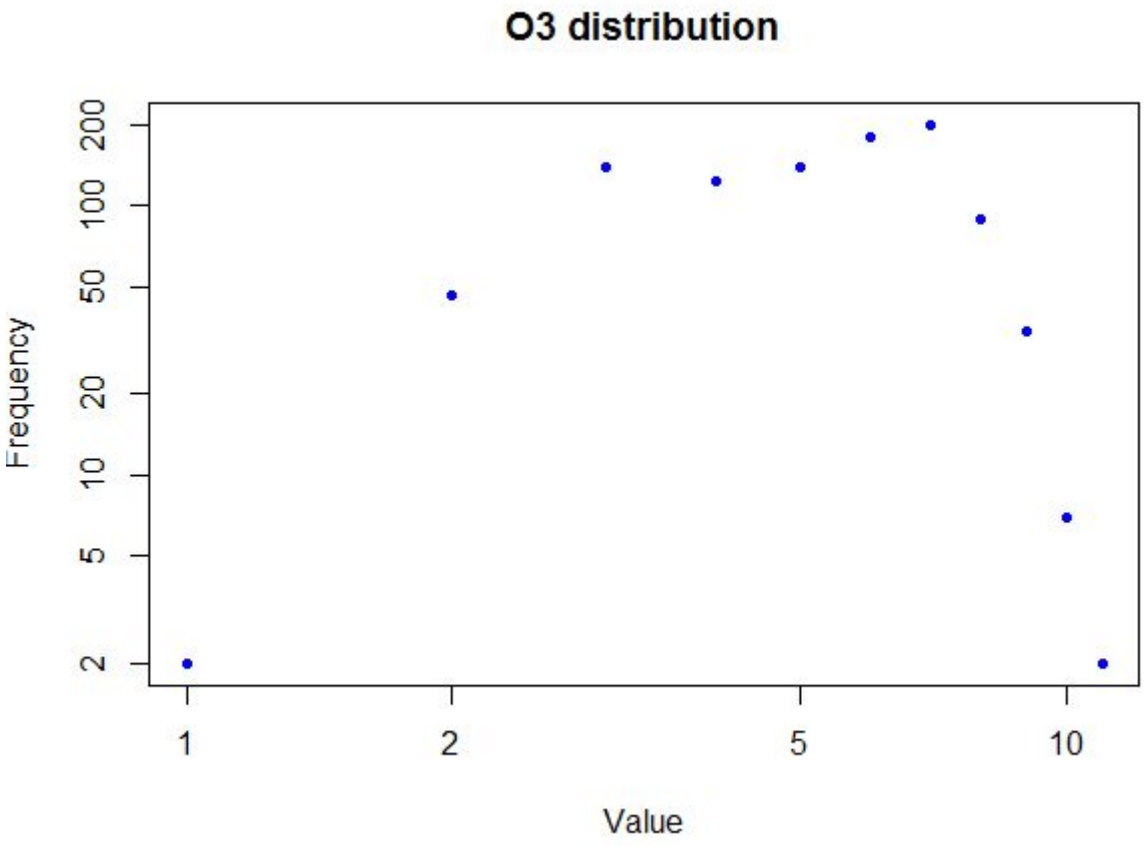
```
r <- patientsDataNo$DIFFERENCE
h <- hist(r, plot=F)
plot(h$counts, log="xy", pch=20, col="blue",
     main="patients distribution",
     xlab="Value", ylab="Frequency")
```

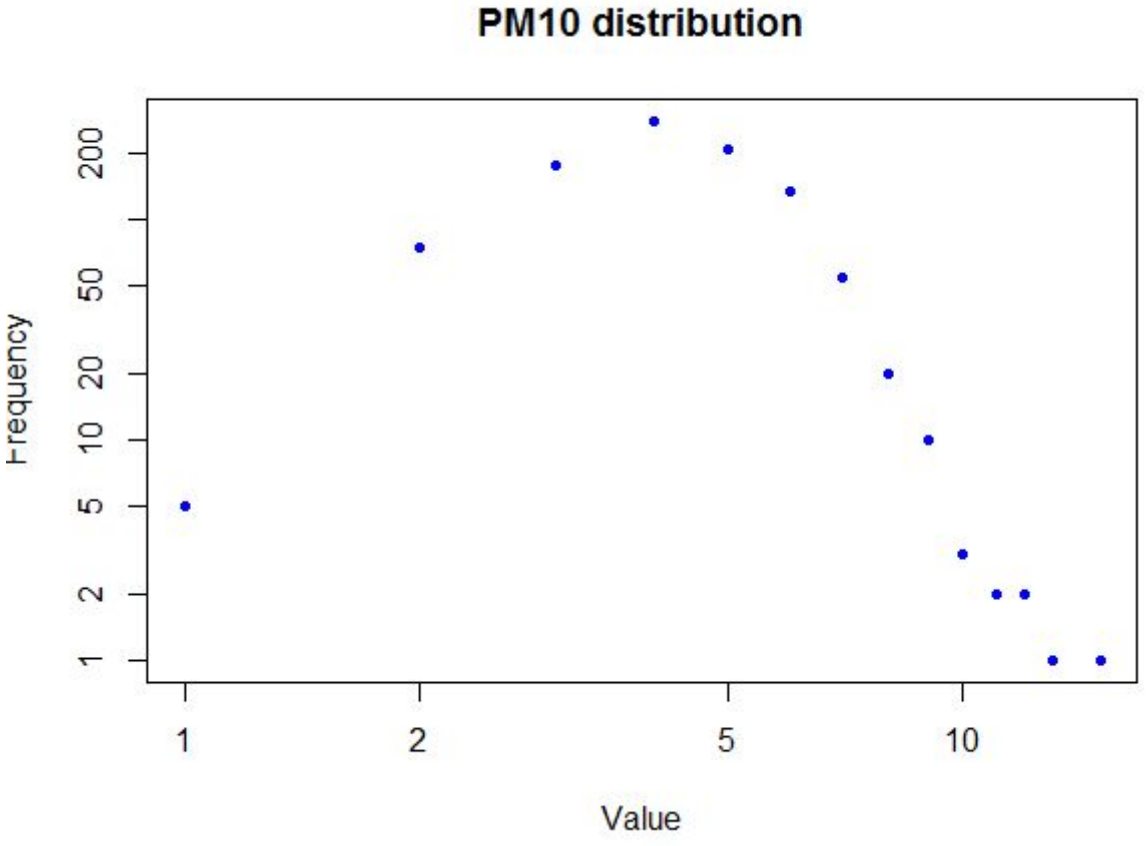
r prend la valeur difference de patientsDataNo (il s'agit d'une fusion des données de santé et de pollution comme suit : patientsDataNo <- merge(x= patientsData2,y= environmentDataNO, by= 'DAY_', all.x= T)
h prend la valeur de l'histogramme de r
plot permet de représenter graphiquement les points de h (la valeur) de couleur bleue avec un graphique ayant pour nom "patients distribution"
Le résultat de ce script est la dernière distribution de ce document.
Les autres distributions ont été réalisées de la même manière avec les valeurs "pourcentage" (cela correspond à la pollution) à la place de "difference".

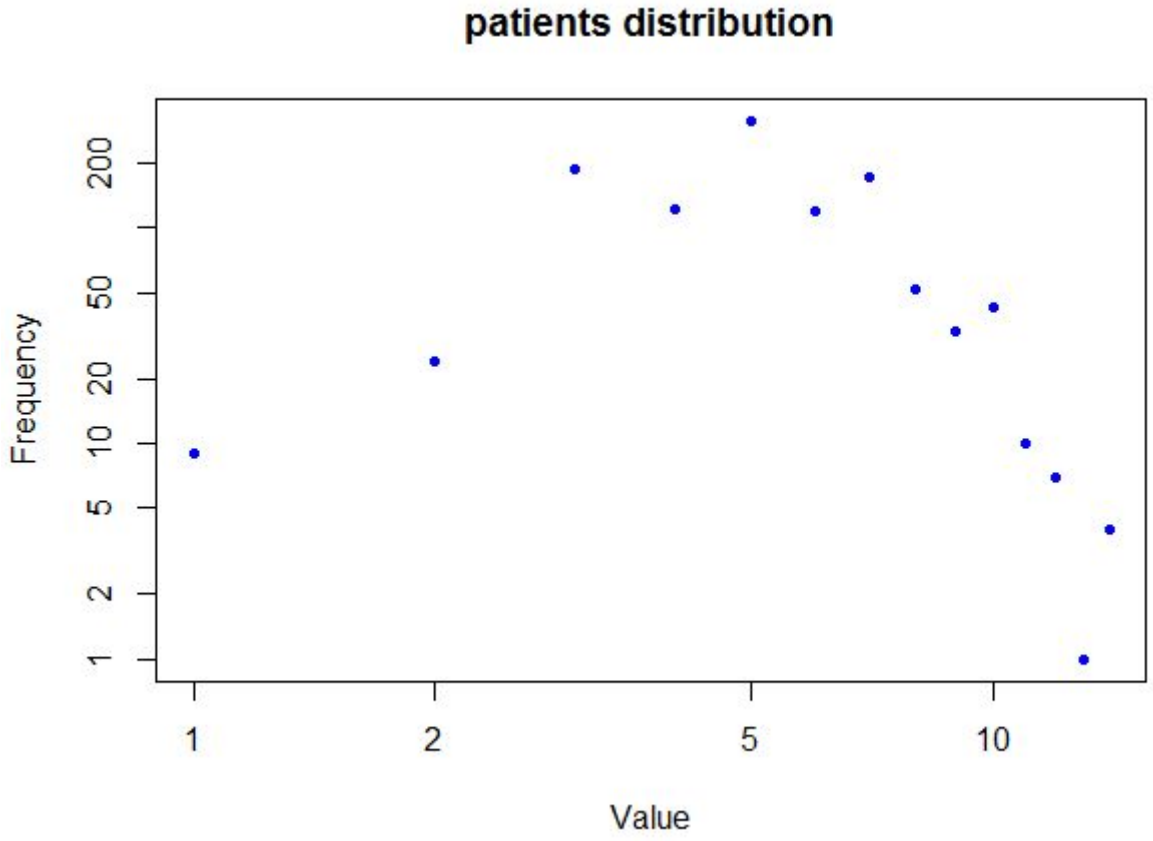



NO2 distribution
Frequency
2
5
10
20
50
100
1
2
5
Value


NOX distribution
Frequency
1
2
5
10
20
50
200
1
2
5
10
Value


O3 distribution
Frequency
2
5
10
20
50
100
200
1
2
5
10
Value


PM10 distribution
Frequency
1
2
5
10
20
50
200
1
2
5
10
Value

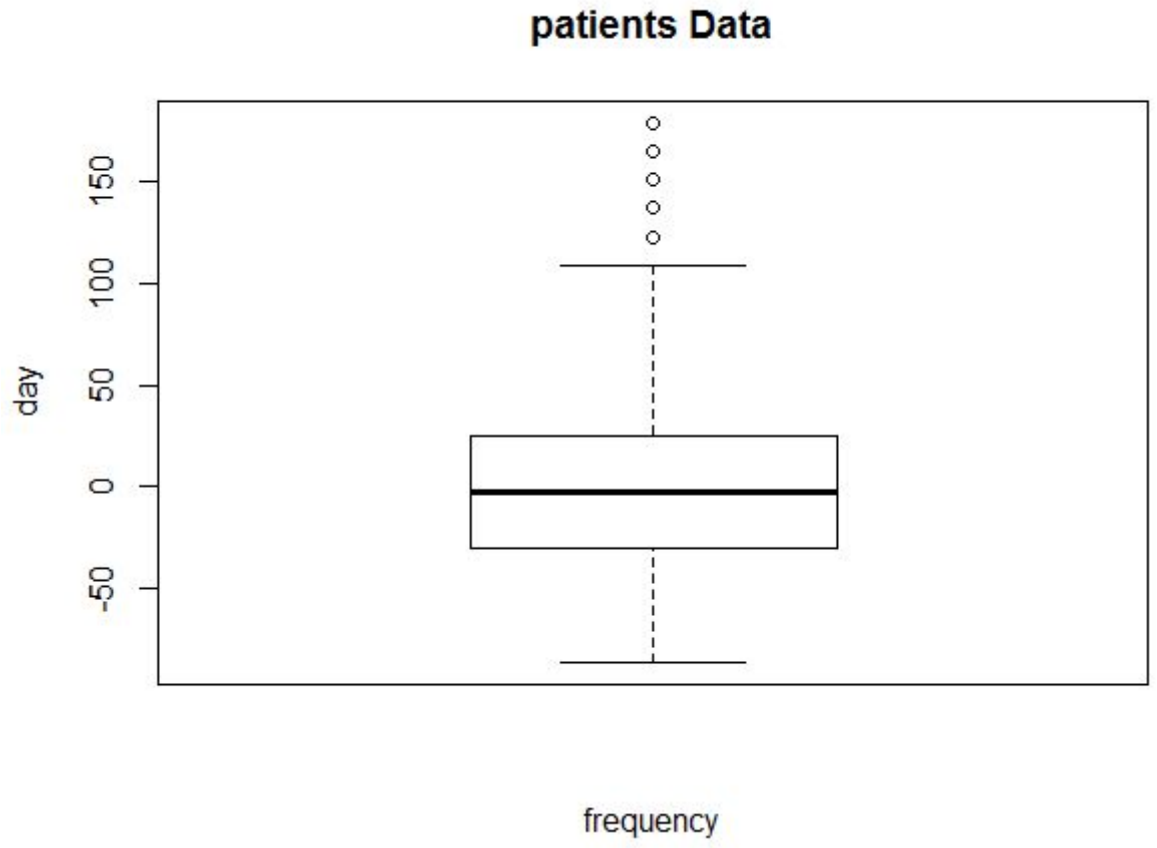


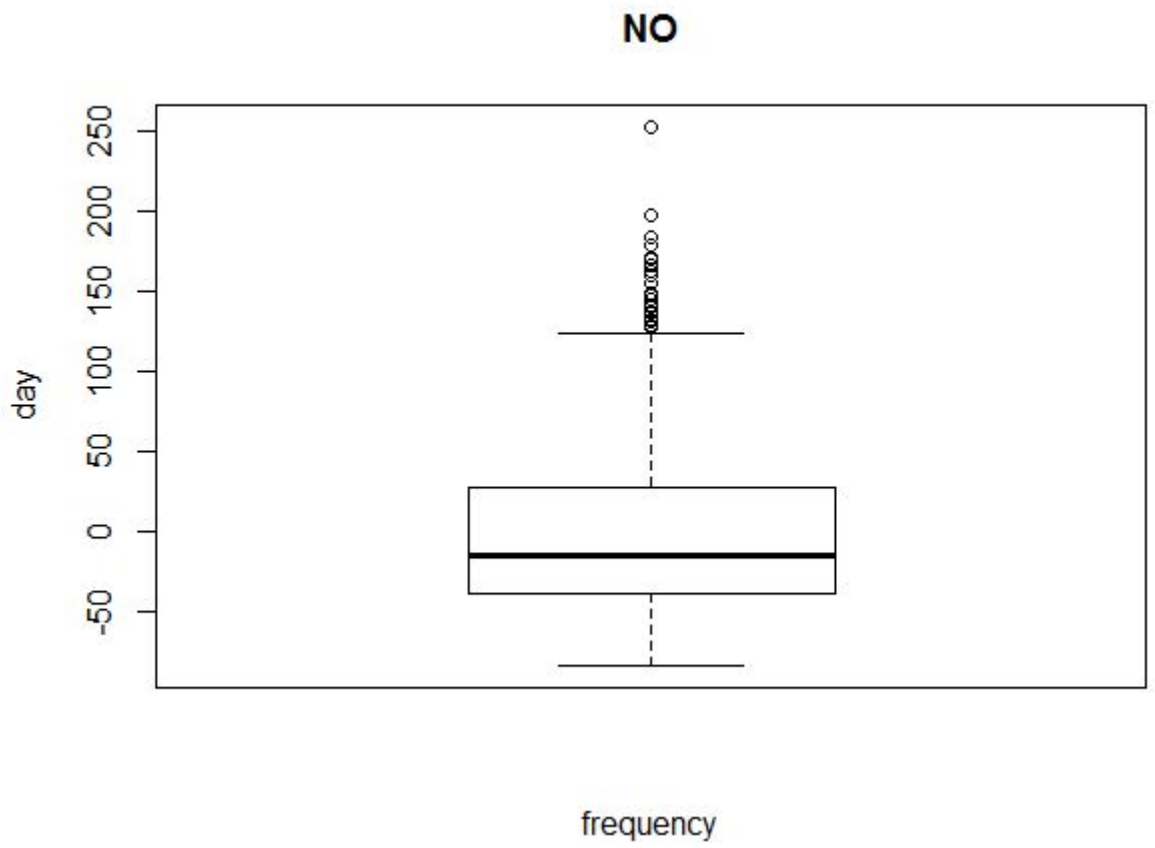
En comparant ces résultats, nous pouvons voir graphiquement qu'une corrélation a l'air d'exister.
Seulement, ici encore, l'échelle semble erronée, il est donc impossible d'établir une bonne interprétation puisque nous ne savons pas réellement à quoi cela correspond.

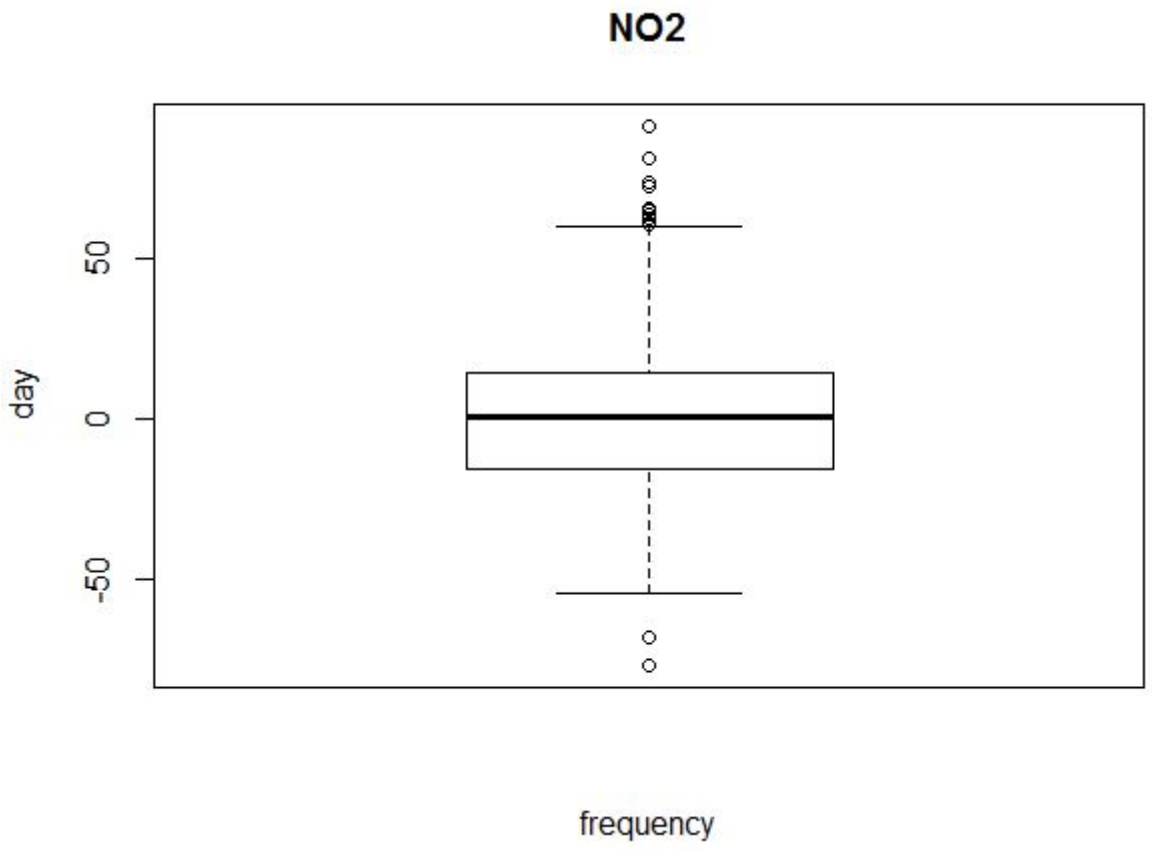
Boxplot

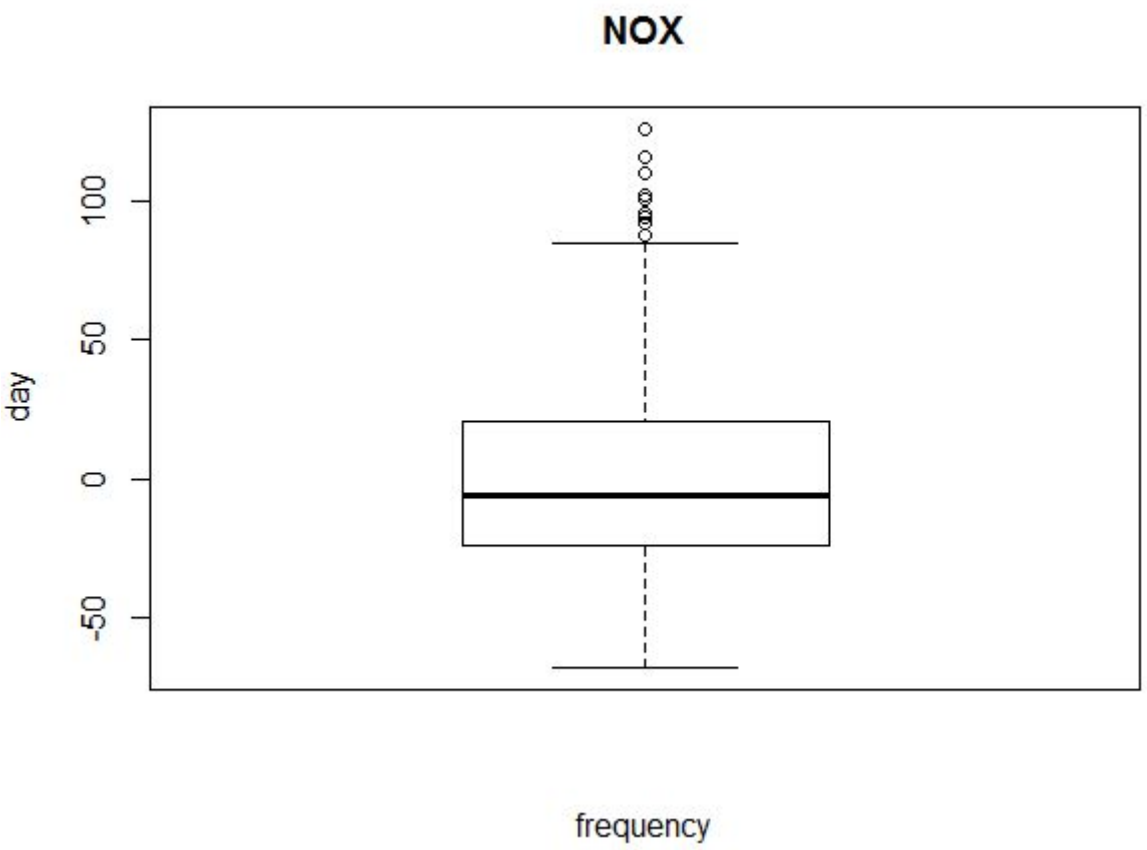
```
r <- patientsDataNo$DIFFERENCE
h <- hist(r, plot=F)
plot(h$counts, log="xy", pch=20, col="blue",
     main="patients distribution",
     xlab="Value", ylab="Frequency")
boxplot(r, xlab = "frequency", ylab = "day", main = "patients Data")
```

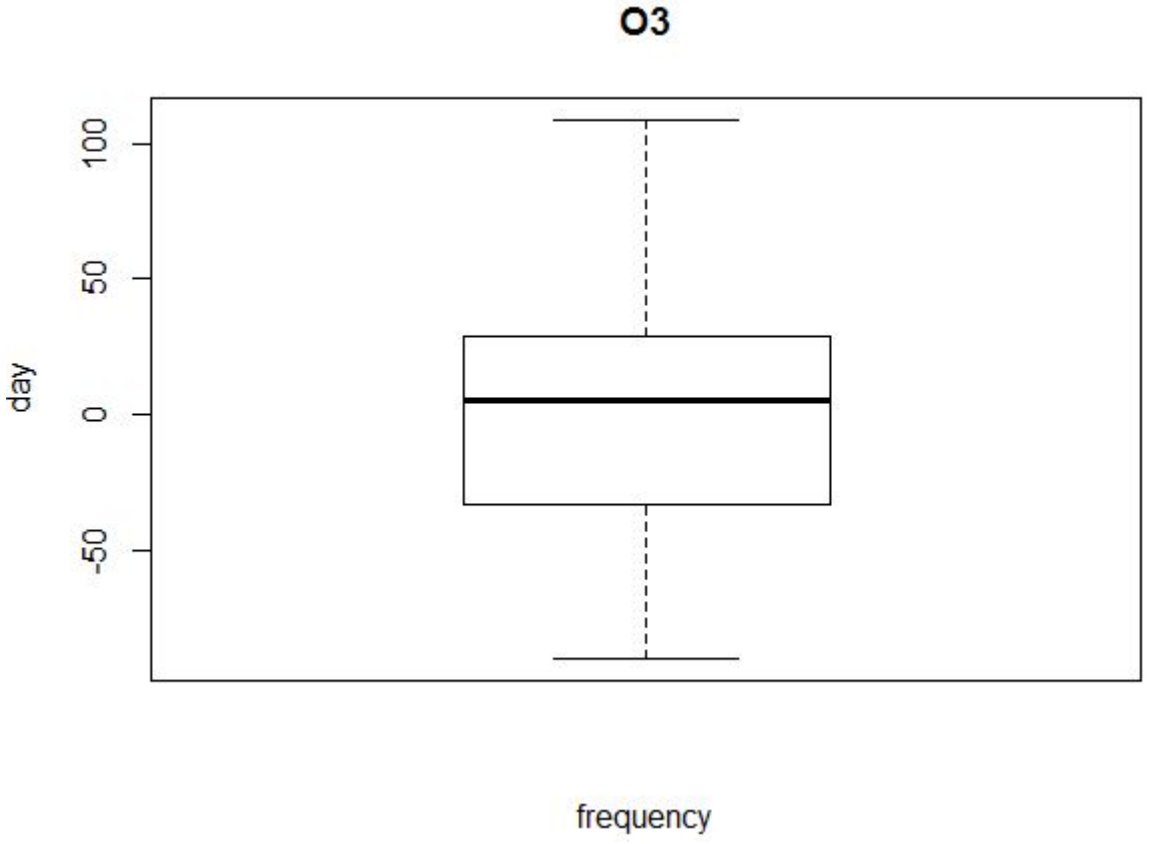
ici, au lieu d'utiliser la fonction plot() comme précédemment, il faut utiliser la fonction boxplot() qui génère une boite à moustache.
En attribut, nous mettons r qui représente les données que nous souhaitons représenter. xlab est l'axe des ordonnées et ylab l'axe des abscisses. xlab représente la fréquence des valeurs et ylab les jours. Main est le nom du graphique : ici il s'agit de "patients Data".

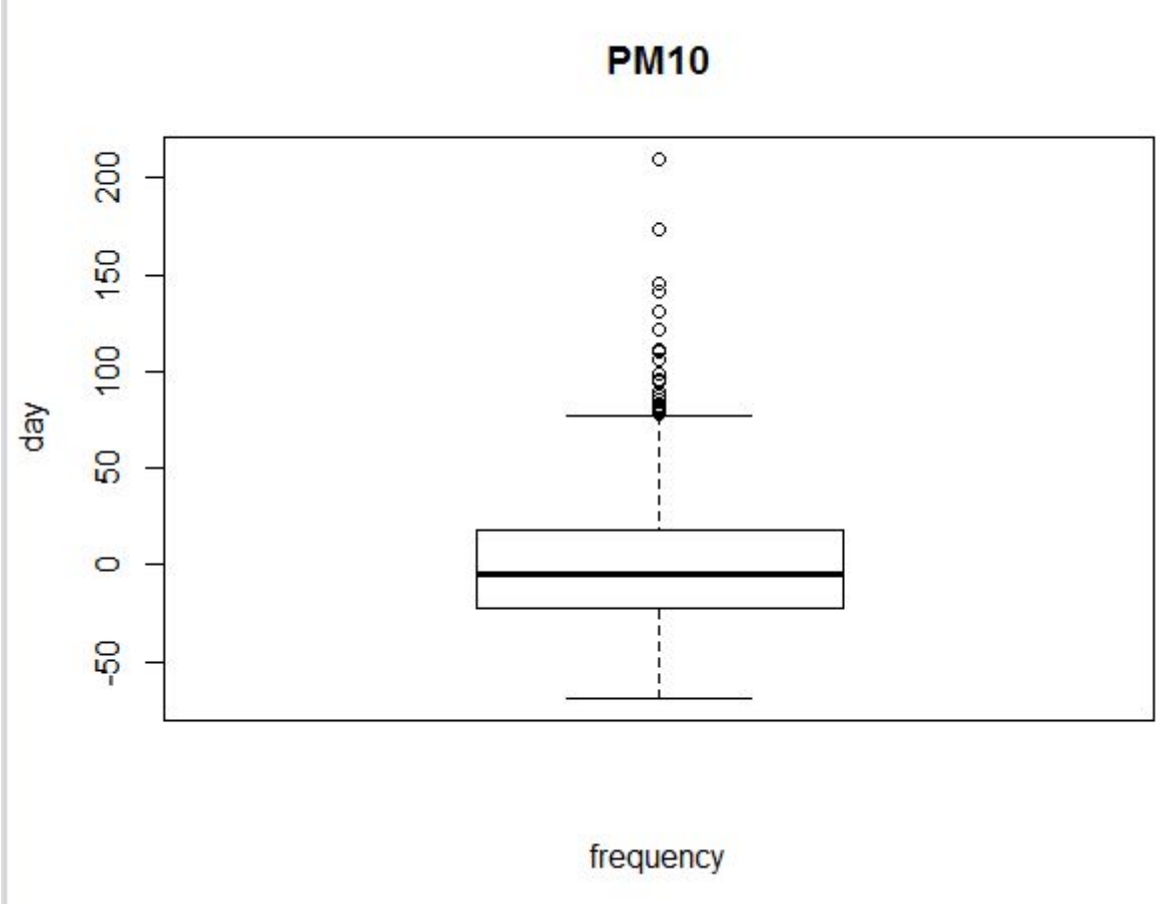


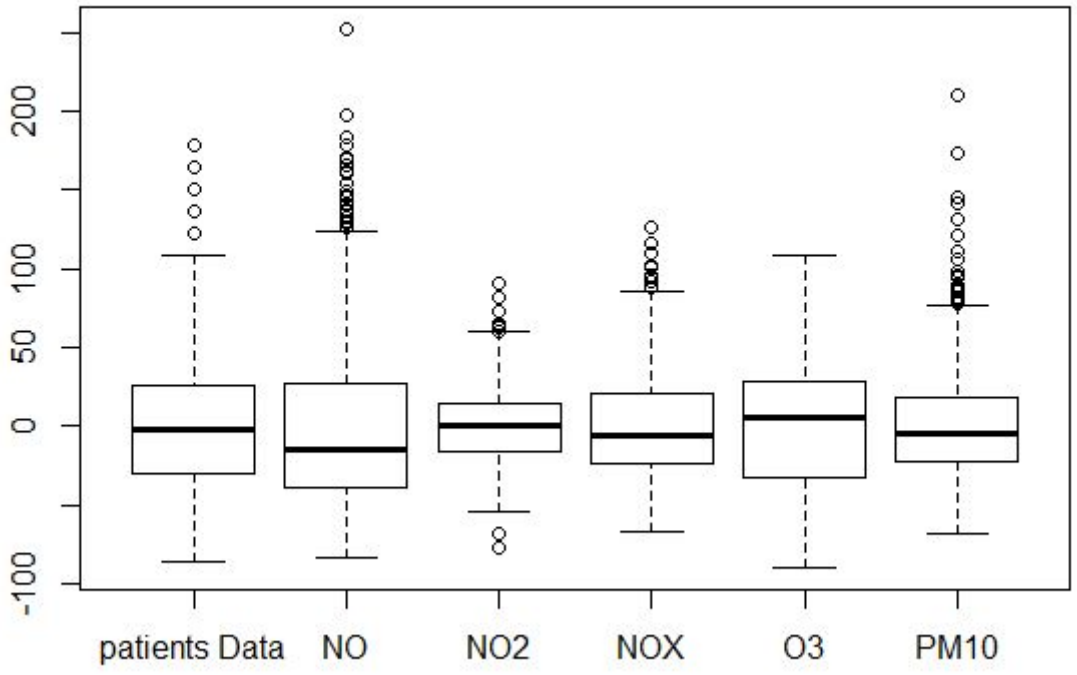

NO
day
250
200
150
100
50
0
-50
frequency


NO2
day
50
0
-50
frequency


NOX
day
100
50
0
-50
frequency

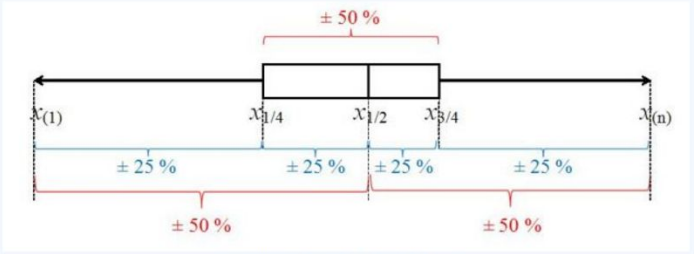

O3
day
100
50
0
-50
frequency


PM10
day
200
150
100
50
0
-50
frequency



Boîte à moustache permet d'une façon simple et visuelle d'avoir quelques traits marquants de la série observée.
Nous pouvons observer la médiane, l'intervalle interquartile, la valeur maximale et minimale de la distribution.
Tendance centrale des valeurs de chaque boîte en observant la position de la médiane. Si elle n'est pas au centre : symétrie de la distribution.
Par la longueur de la boîte : estimer variabilité des valeurs.
Longueur des moustaches : idée de la taille de la queue de la distribution.
Ici, pour la plupart des sous groupes, la médiane est dans le bas de la boîte, ce qui suppose une distribution asymétrique vers les valeurs basses.
De plus, la médiane correspondante aux données de patients se rapproche des médianes des différents polluants ce qui peut caractériser une certaine corrélation.
Enfin, le graphique montre la présence de beaucoup de valeurs extrêmes. Il est possible qu'il y ait eu des erreurs au niveau des données ou que l'échantillon analysé soit trop gros car plus nous avons d'échantillon important, plus il y a de valeurs extrêmes.
La présence de données extrêmes est assez gênante car beaucoup de traitements statistiques y sont très sensibles.

http://www.itse.be/statistique2010/co/233_Cours_boxplot.html
http://zoonek2.free.fr/UNIX/48_R_2004/04.html
http://ceal.fing.uncu.edu.ar/data_mining/Libros/RJournal_2009-2_Williams.pdf



Informations fournies par la version de base

- la médiane nous renseigne sur le milieu de la série ;
- les largeurs des deux parties de la boîte rendent compte de la dispersion des valeurs situées au centre de la série (la boîte contient 50% (environ) de l'ensemble des observations : 25% à gauche de la médiane et 25% à sa droite) ;
- la longueur des moustaches renseigne sur la dispersion des valeurs situées au début de la série ordonnée (les valeurs les plus petites correspondant à 25% des observations) ou à la fin de celle-ci (les valeurs les plus grandes correspondant aussi à 25% des observations) ;
- de façon générale, la boîte et les moustaches seront d'autant plus étendues que la dispersion de la série statistique est grande.

http://www.itse.be/statistique2010/co/233_Cours_boxplot.html

Rattle

Rattle est une interface intéressante pour représenter l'arbre de décision.
Seulement une erreur persiste. J'essaierai de générer un arbre de décision sur cette interface afin de le comparer aux autres établis lorsque l'erreur sera résolue.

https://rstudio-pubs-static.s3.amazonaws.com/47698_6effe42ff227451ab364b393d0bcf495.html

https://public.opendatasoft.com/explore/dataset/code-insee-postaux-geoflar/?refine.nom_region=PROVENCE-ALPES-COTE+D%27AZUR

https://datanova.laposte.fr/explore/dataset/code-postal-code-insee-2015/table/?refine.nom_dept=ALPES-MARITIMES

Pollution mapping

Ici, le but est de “mapper” une carte de Nice et d’y représenter les différents niveau de pollution avec R.
Ensuite, il serait intéressant de faire une carte dynamique qui évoluerait au cours du temps afin de voir l’évolution de la pollution dans Nice.

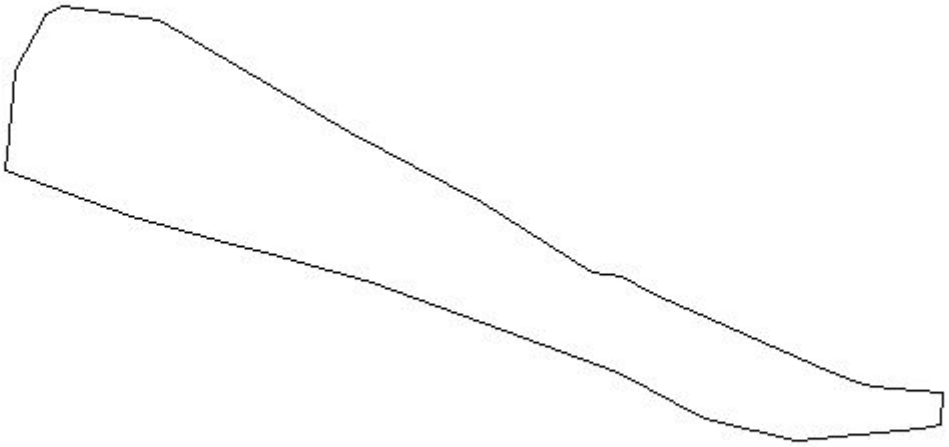
GEOFLA :
Après avoir récupéré les données géographique de la France sur GEOFLA, voilà les cartes que j’ai réussi à établir.

```
pathToShp <- "./GEOFLA/GEOFLA/GEOFLA/1_DONNEES_LIVRAISON_2016-06-00236/shp/COMMUNE"
ogrInfo(dsn = pathToShp,layer="COMMUNE")
# Import via la fonction readOGR de rgdal
comm <- readOGR(dsn = pathToShp, layer="COMMUNE", stringsAsFactors=FALSE)
# Description de la structure globale
str(comm, 2)
# Description de la structure des données associées
str(comm@data, 2)
# Description de la structure des données vectorielles
head(comm@polygons, n=1)
# On représente les contours des communes de France avec la fonction plot
plot(comm)
```

pathToShp prend en mémoire le fichier commune.shp qui a été téléchargé depuis GEOFLA.
comm prend les valeurs des communes avec la fonction rgdal() qui permet de lire le fichier shp.
srt(comm,2) permet de décrire la structure
plot(comm) permet de représenter, grâce à la fonction plot() vue précédemment, les contours des communes de France.

```
# On plot la commune de Nice (CODE_COMM égal à 88)
plot(comm[comm$ID_GEOFLA=="COMMUNE00000000000011063",],)
```

Ici, nous souhaitons seulement représenter la commune de Nice.
Après des recherches sur internet, j'ai trouvé que l'ID_GEOFLA de Nice était "COMMUNE00000000000011063".
J'ai donc fait une sélection de la commune de Nice avec cet identifiant et l'ai représenté avec la fonction plot().



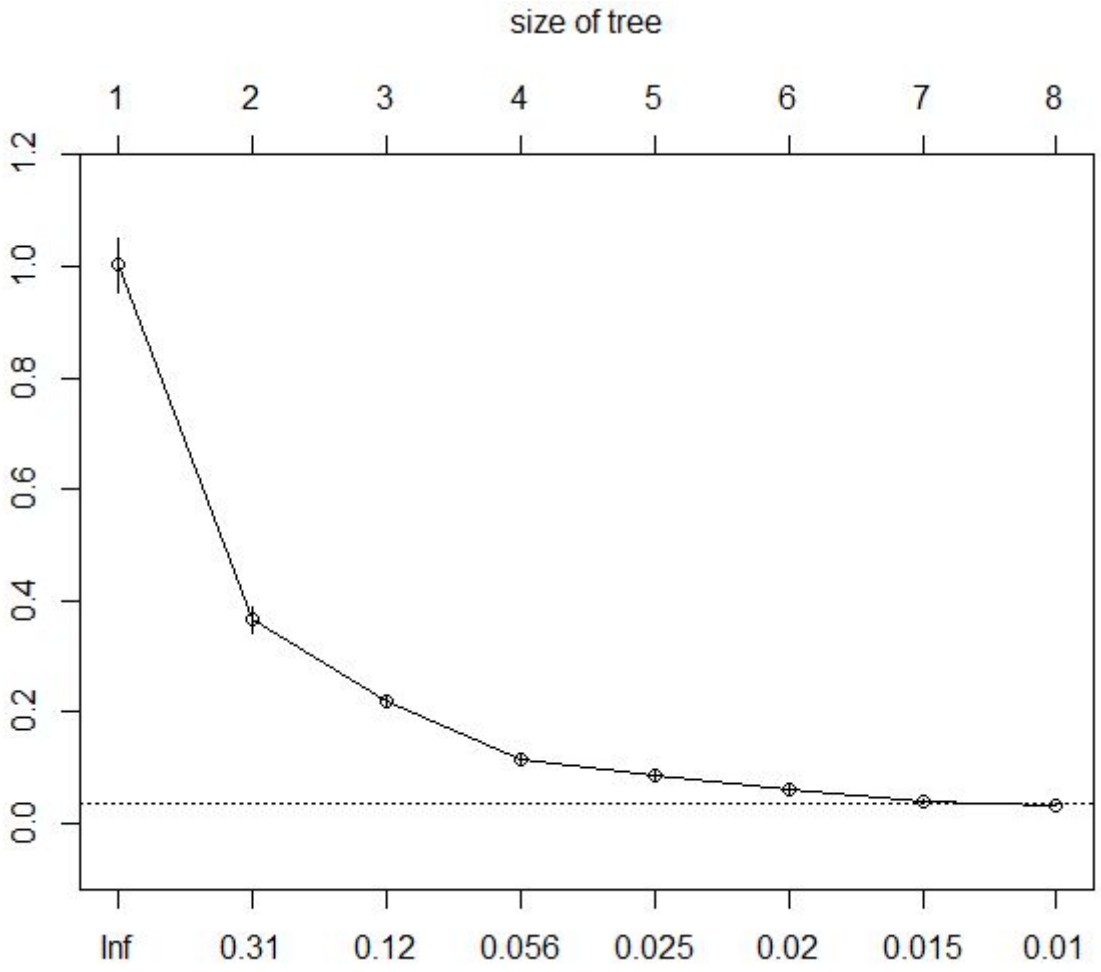
Comparaison des méthodes d'analyse R
(Références : cours Nicolas Pasquier)

Méthode	Avantages	Inconvénients
Random Forest	Efficace pour les données de grande dimension Bonnes propriétés	Sur-apprentissage dans le cas de données bruitées

Decision tree - Test 1

```
> patientsNO.rpart1 = rpart(NUMBEROFPATIENTS ~  DIFFERENCE + SR_ZNACH + DAY_ + PERCENT_,
+                           data = patientsDataNo)
> plotcp(patientsNO.rpart1)
```

La fonction rpart() permet de générer un arbre de décision. Elle prend en paramètre le nombre de patients, la différence, la valeur moyenne (SR_ZNACH), le jour et le pourcentage.
plotcp() permet de dessiner graphiquement la taille de l'arbre de décision.



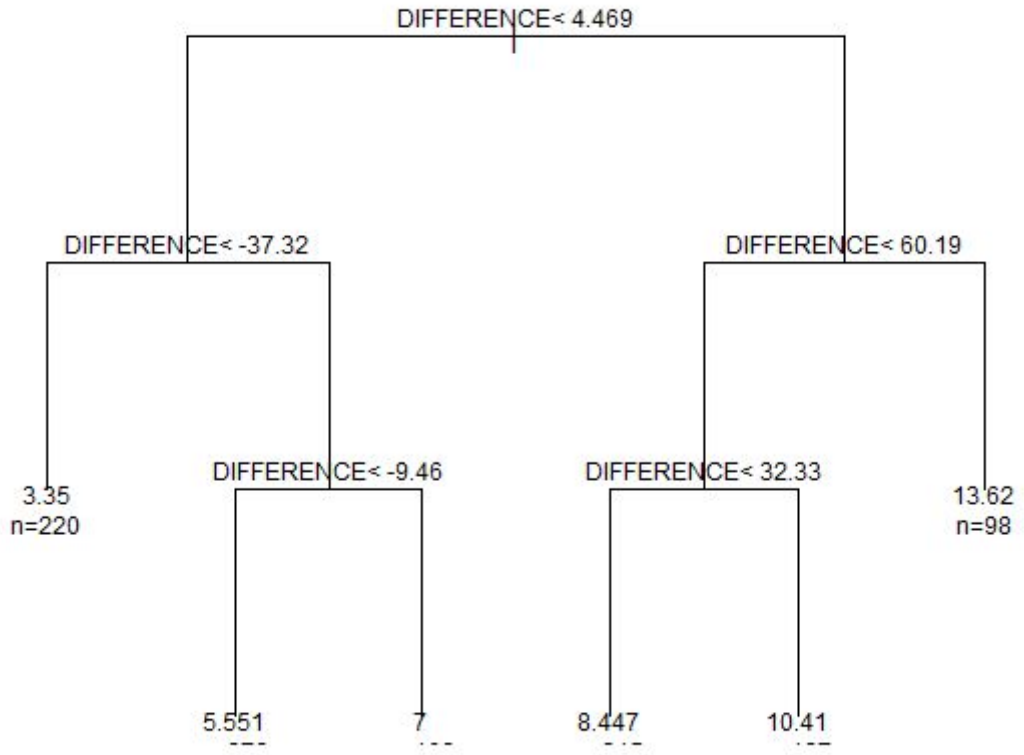
https://www.r-bloggers.com/classification-trees-using-the-rpart-function/

printcp() donne des indications sur l'arbre de décision.
printcp(patientsNO.rpart1)

```
Regression tree:
rpart(formula = NUMBEROFPATIENTS ~ DIFFERENCE + SR_ZNACH + DAY_ +
    PERCENT_, data = patientsDataNo)

Variables actually used in tree construction:
[1] DIFFERENCE

Root node error: 10309/1094 = 9.4231

n= 1094

        CP nsplit rel error   xerror      xstd
1 0.636910      0  1.000000 1.001761 0.0496269
2 0.146125      1  0.363090 0.366612 0.0223721
3 0.106490      2  0.216965 0.218703 0.0099526
4 0.029841      3  0.110474 0.111926 0.0073919
5 0.020665      4  0.080633 0.081841 0.0072216
6 0.019998      5  0.059968 0.059591 0.0070890
7 0.010623      6  0.039970 0.041329 0.0027867
8 0.010000      7  0.029347 0.033776 0.0025422
```

```
> patientsNO.rpart2 = prune(patientsNO.rpart1, cp = 0.02)
> plot(patientsNO.rpart2, uniform = TRUE)
> text(patientsNO.rpart2, use.n = TRUE, cex = 0.75)
```

plot() permet ici de dessiner les bases de l'arbre de décision.
text() permet d'ajouter le text à l'arbre de décision.



Ceci est un premier test afin de comprendre comment marche l'arbre de décision. Il dépend de la variable "différence" et établi ainsi les comparaisons nécessaires.
Actuellement, je suis entrain d'en faire un avec toutes les variables importantes, à savoir l'âge et le genre ainsi que le taux de pollution.

Clusturing - Test 1

```
d <- dist(as.matrix(patientsDataNo))
hc <- hclust(d)
plot(hc)
```

```
summary(d)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 1.08  200.11  363.78  418.96  605.97 1578.61
```

La fonction dist() calcule et renvoie la matrice de distance calculée en utilisant la mesure de distance spécifiée pour calculer les distances entre les lignes d'une matrice de données.Ici, la matrice est celle de patientsDataNo

hclust(d) permet d’appliquer un regroupement hiérarchique.
plot(hc) dessine le cluster en arbre