TP2 Statistiques 2

18 février 2022

L'analyse exploratoire des données: révision et extension

Air quality monitoring

Airparif exploite un système de surveillance de la qualité de l'air avec un réseau de sites dans la région de la capitale (Ile de France) sur lesquels les mesures de la qualité de l'air sont effectuées automatiquement. Ces mesures sont utilisées pour résumer les niveaux actuels de pollution atmosphérique, pour prévoir les niveaux futurs et pour fournir des données pour la recherche scientifique, contribuant à l'évaluation des risques pour la santé et des impacts environnementaux des polluants atmosphériques.

Nous examinerons *l'ozone troposphérique* (O_3). Ce polluant n'est pas émis directement dans l'atmosphère, mais est produit par des réactions chimiques entre le dioxyde d'azote (NO_2), les hydrocarbures et la lumière du soleil.

Nous nous concentrerons sur les données de deux sites de surveillance: un site urbain à Neuilly-sur-seine (**NEUIL**) et un site rural (**RUR.SE**) près de la forêt de Fontainbleu.

Les principales questions d'intérêt sont

- Comment, le cas échéant, la distribution des mesures de l'ozone varie-t-elle entre les sites urbains et ruraux?
- Comment, le cas échéant, la distribution des mesures d'ozone est-elle affectée par saison?

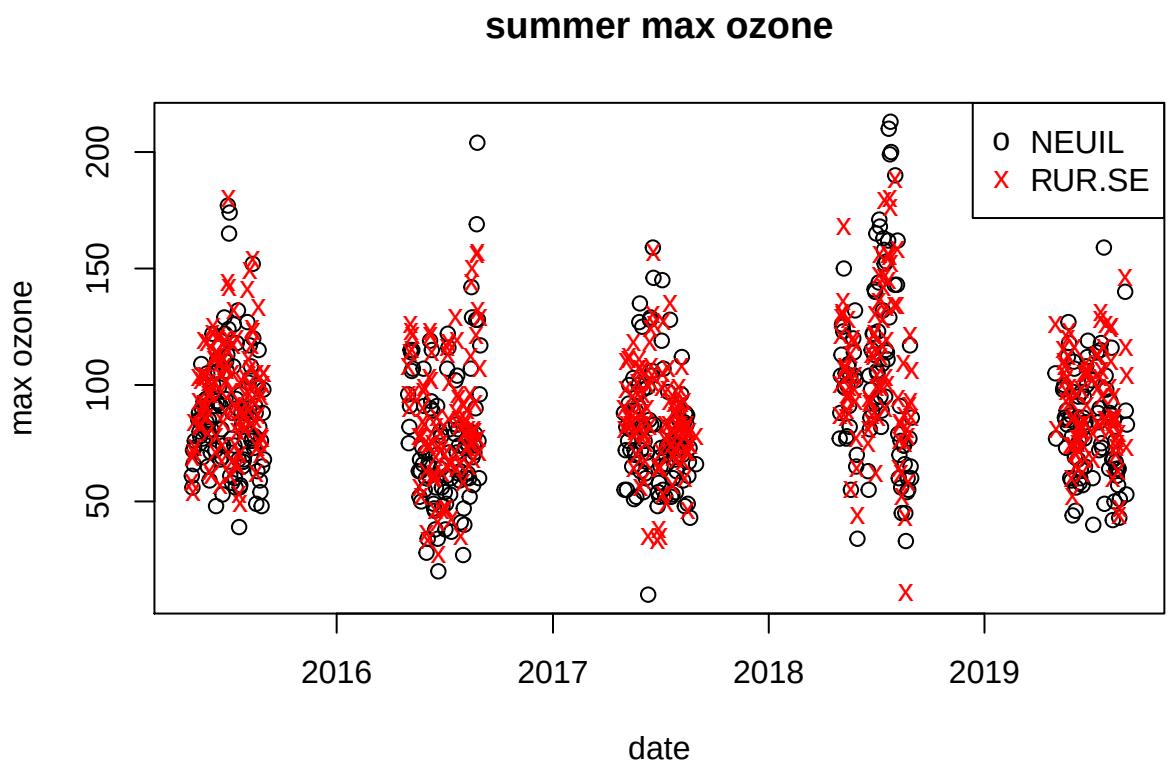
Les données de chaque site sont des mesures quotidiennes de la concentration moyenne horaire maximale de O_3 enregistrée en microgrammes par mètre cube ($\mu g/m^3$), de 2014 à 2019 inclusivement. Pour nous concentrer sur la question de la saison, nous comparons les données de *hiver* (novembre-février inclus) (*winter_ozone.csv*) et *été* (mai - août inclus) (*summer_ozone.csv*).

Import data from a .csv file

Nous importons le jeu de données d'été et effectuons la vérification initiale avec la commande **str**, **head** et **tail**. La fonction **names** imprime les noms des variables de colonne et **summary** donne les statistiques récapitulatives par colonne.

```
ozone.summer = read.csv("summer_ozone.csv")
str(ozone.summer)  # structure
names(ozone.summer) # names of the variables
head(ozone.summer)  # first few observations
tail(ozone.summer)  # last few observations
ozone.summer[1:10,] # first 10 observations
summary(ozone.summer) # summary
ozone.summer$date = as.Date(ozone.summer$date2)  # transform chr to date format
## plot
plot(ozone.summer$date, ozone.summer$NEUIL,
     xlab="date", ylab="max ozone", main="summer max ozone")
points(ozone.summer$date, ozone.summer$RUR.SE, col="red", pch = "x")
```

```
legend("topright", legend = c("NEUIL", "RUR.SE"),
       col=c("black","red"), pch=c("o","x"))
```

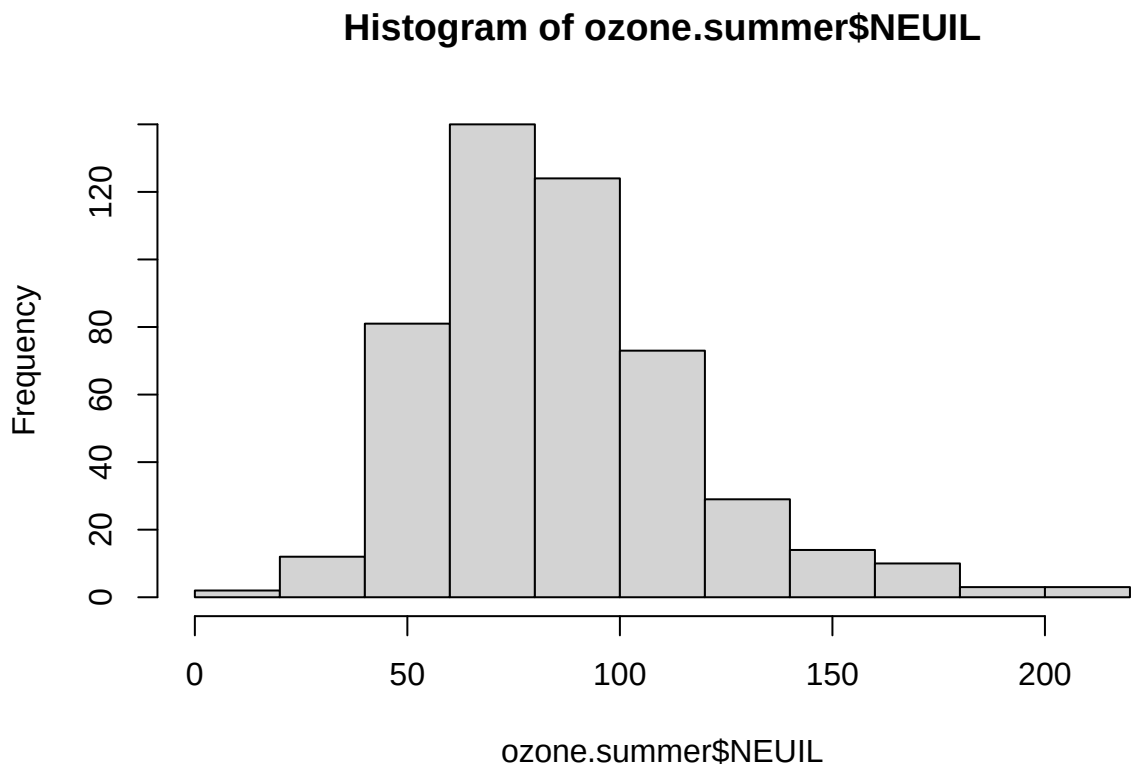


1. Quelle est la taille des données? Y a-t-il une différence entre les sites urbains et ruraux? Y a-t-il une variation annuelle?

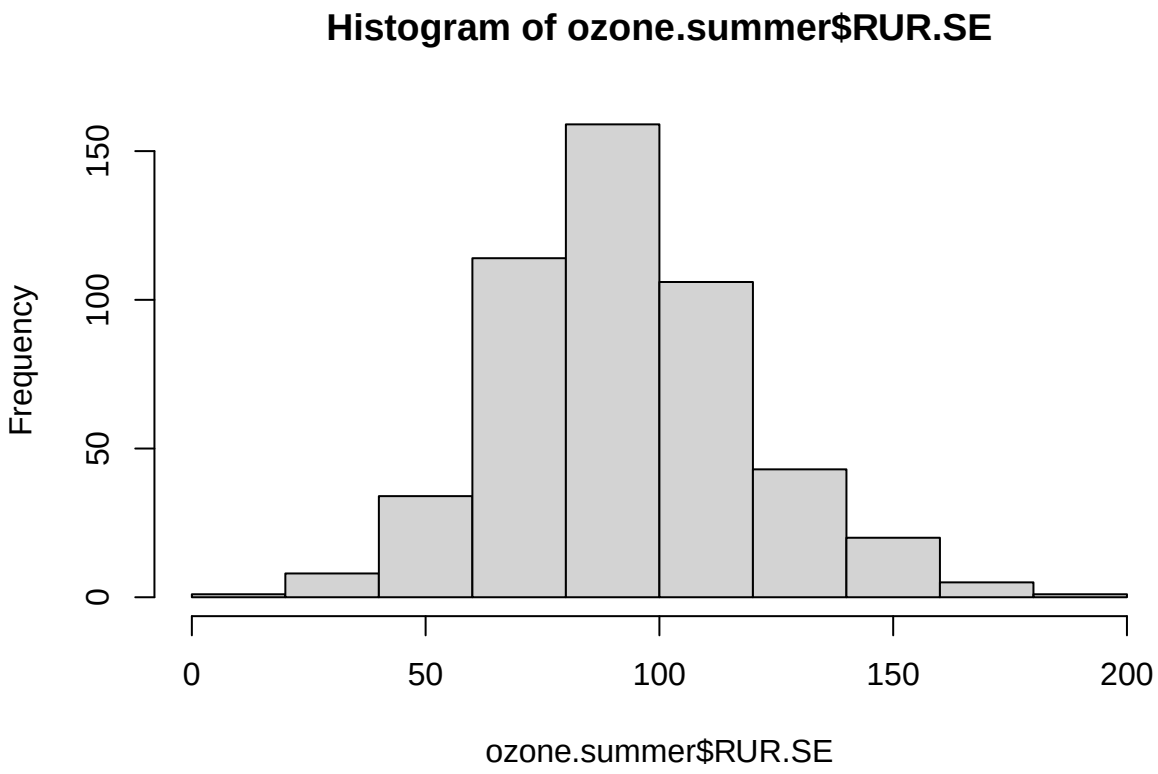
REPONSE Q1: On a n=491. Il n'y a pas de grandes différences entre milieux urbains et ruraux , si ce n'est que les pics d'ozones ont lieu en milieu urbain (données plus étalées). Il ya une petite variation annuelle entre +-50 µ-grammes/m3.

2. Faites les histogrammes des données d'été sur l'ozone pour les deux sites et comparez-les. Pour les rendre comparables sur l'échelle de densité, vous pouvez utiliser l'option *prob = TRUE*.

```
## histogram
hist(ozone.summer$NEUIL)
```



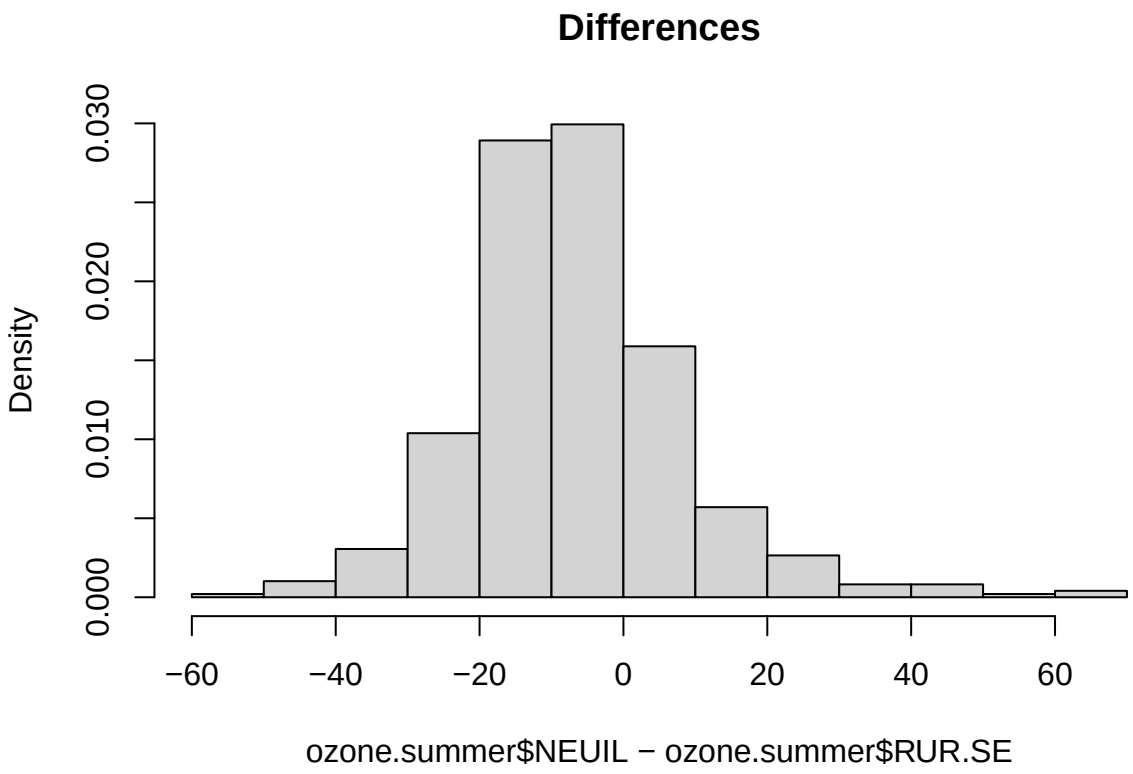

```
hist(ozone.summer$RUR.SE)
```



Une partie de la variabilité de chacune des distributions est due aux conditions climatiques, qui seront similaires puisqu'elles sont relativement proches les unes des autres. Comme nous nous intéressons à la différence des distributions, nous n'avons pas nécessairement à regarder la distribution séparée elle-même.

Nous désignons les données sur l'ozone du site urbain par $x_1, \ldots, x_n$ et le site rural par $y_1, \ldots, y_n$, l'indice indiquant les n jours différents pour lesquels nous avons des mesures. L'histogramme ci-dessous montre la différence $d_i = x_i - y_i$ pour $i = 1, \ldots, n$ pour les jours d'été.

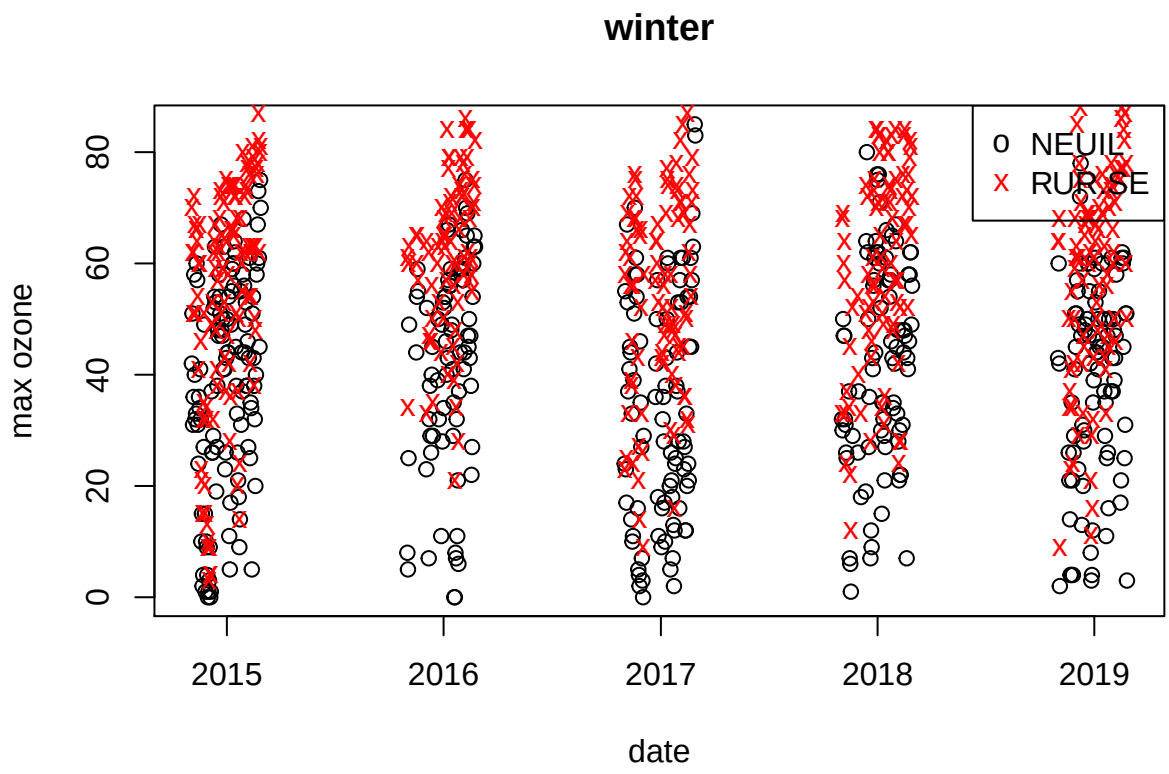
```
hist(ozone.summer$NEUIL - ozone.summer$RUR.SE, prob=TRUE, main = "Differences")
```



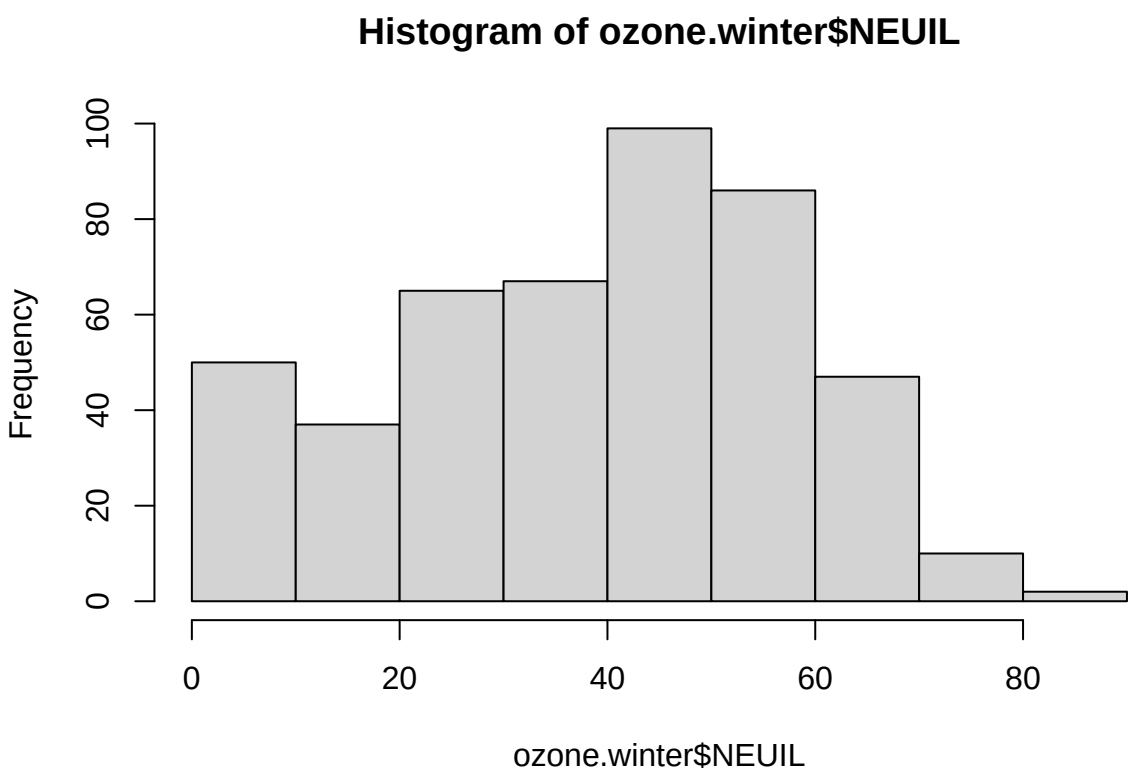
La variabilité de ces données différenciées est considérablement moindre que la variabilité des mesures effectuées sur les sites séparés. Cela indique que des facteurs communs affectant les deux sites influencent la variation des valeurs d'ozone. La plupart des différences sont négatives, ce qui suggère qu'en général les mesures rurales sont plus grandes que les mesures urbaines, ce qui coïncide avec les attentes scientifiques.

3. Répétez l'analyse pour les données d'hiver sur l'ozone. Résumez vos découvertes.

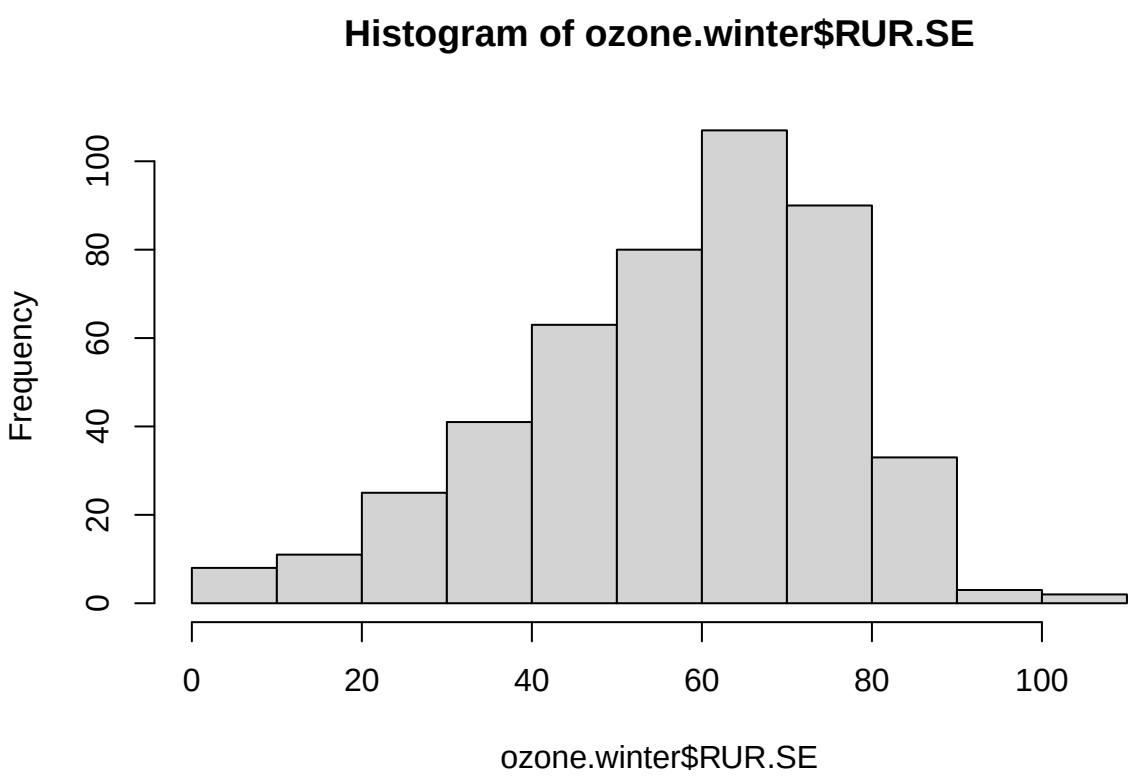
```
ozone.winter = read.csv("winter_ozone.csv")
str(ozone.winter)
ozone.winter$date = as.Date(ozone.winter$date2)

plot(ozone.winter$date, ozone.winter$NEUIL, xlab="date", ylab="max ozone", main="winter")
points(ozone.winter$date, ozone.winter$RUR.SE, col="red", pch = "x")
legend("topright", legend = c("NEUIL", "RUR.SE"), col=c("black","red"), pch=c("o","x"))
```




```
## histogram
hist(ozone.winter$NEUIL)
```




```
hist(ozone.winter$RUR.SE)
```



REPONSE Q3 : On constate 3 différences entre les statistiques d'été et d'hiver
-en moyenne,il y a 2 fois moins de pollution en hiver (45 µ-grammes/m3 en milieu urbain et 65 en milieu rural) qu'en été (45 µ-grammes/m3 en milieu urbain et 90 en milieu rural)
-les données sont moins étalées en hiver
-contrairement aux périodes estivales dans lesquelles on observe des pics en milieux urbains, on observe surtout des maxima en milieu rural

Empirical distribution function

La fonction de distribution empirique est donnée par

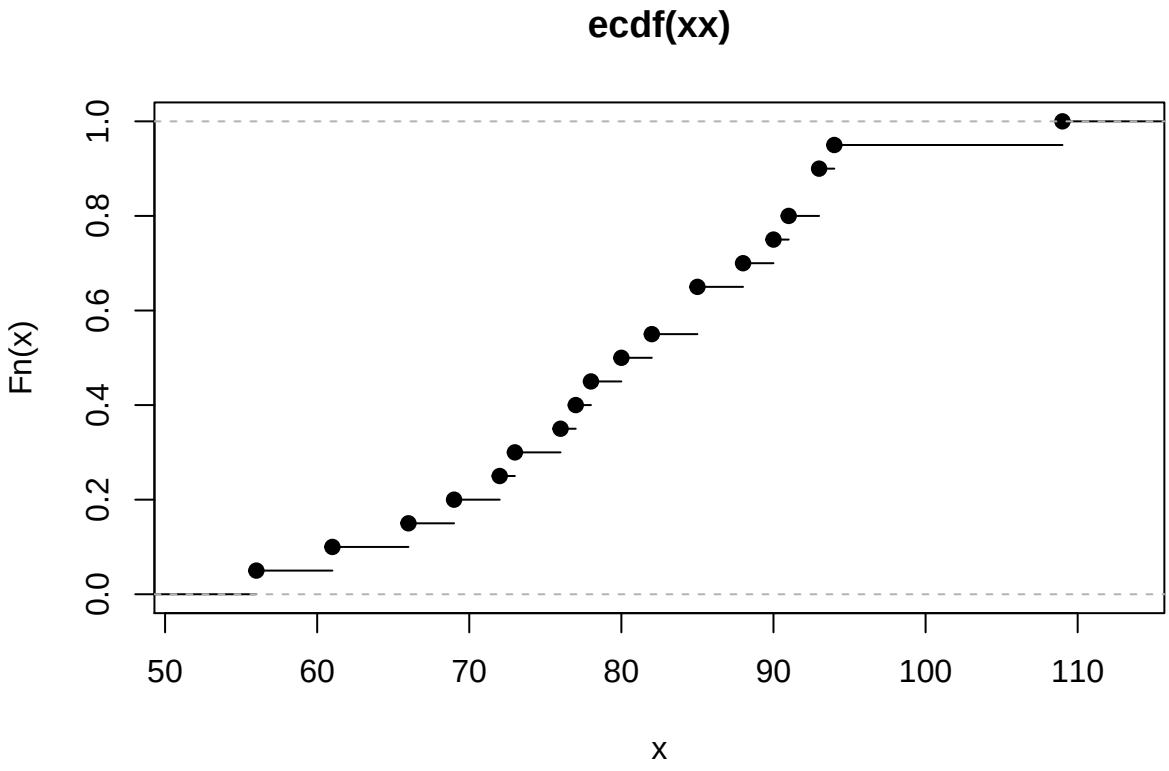
$$F_n(x_{(i)}) = \frac{i}{n}$$

aux points de données ordonnés $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(n)}$. Pour les valeurs de x entre les points de données, nous avons

$$F_n(x) = \frac{i}{n}\,, \text{où } x_{(i)} \leq x < x_{(i+1)}$$

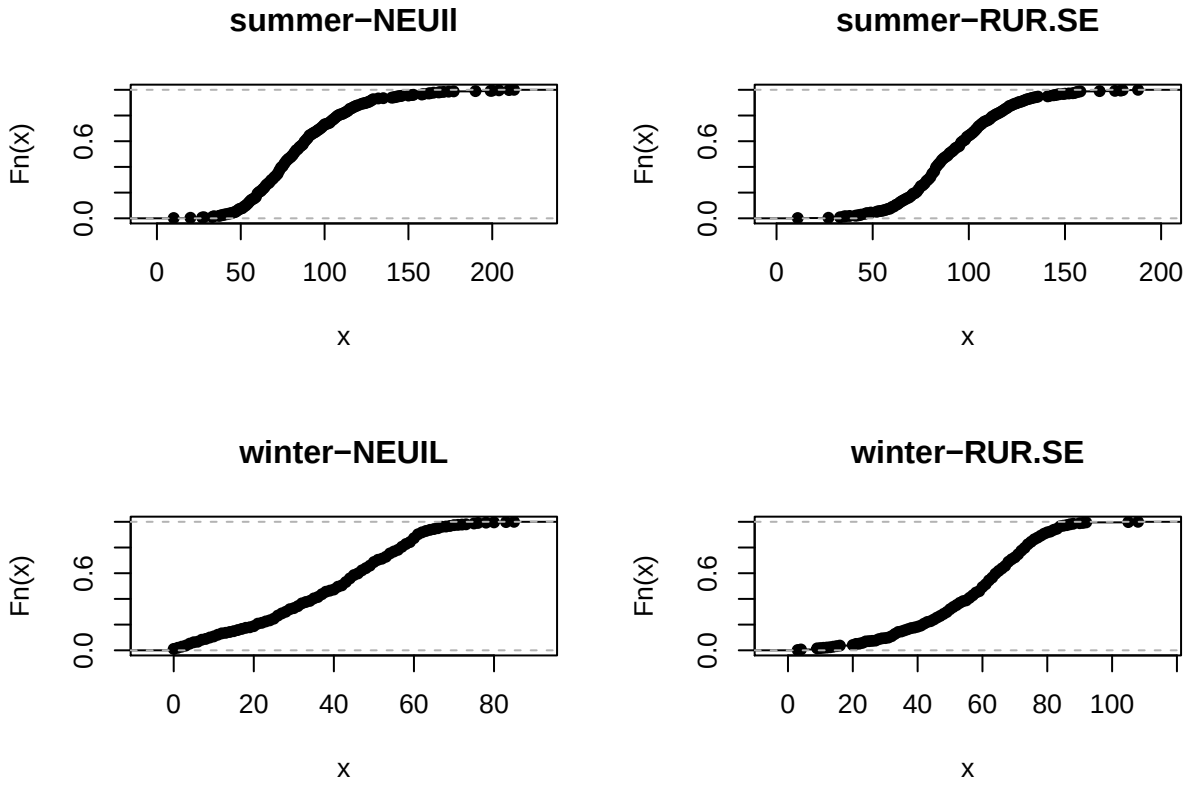
Par exemple, les 20 premières observations des mesures d'ozone en milieu urbain sont

```
xx = ozone.summer$NEUIL[1:20]
sort(xx)
plot(ecdf(xx))
```



4. Faites le c.d.f empirique des jeux de données complets pour chaque site. Expliquez comment utiliser les graphiques pour estimer la médiane.

```
par(mfrow=c(2,2))
plot(ecdf(ozone.summer$NEUIL), main="summer-NEUIl")
plot(ecdf(ozone.summer$RUR.SE), main="summer-RUR.SE")
plot(ecdf(ozone.winter$NEUIL), main="winter-NEUIL")
plot(ecdf(ozone.winter$RUR.SE), main="winter-RUR.SE")
```



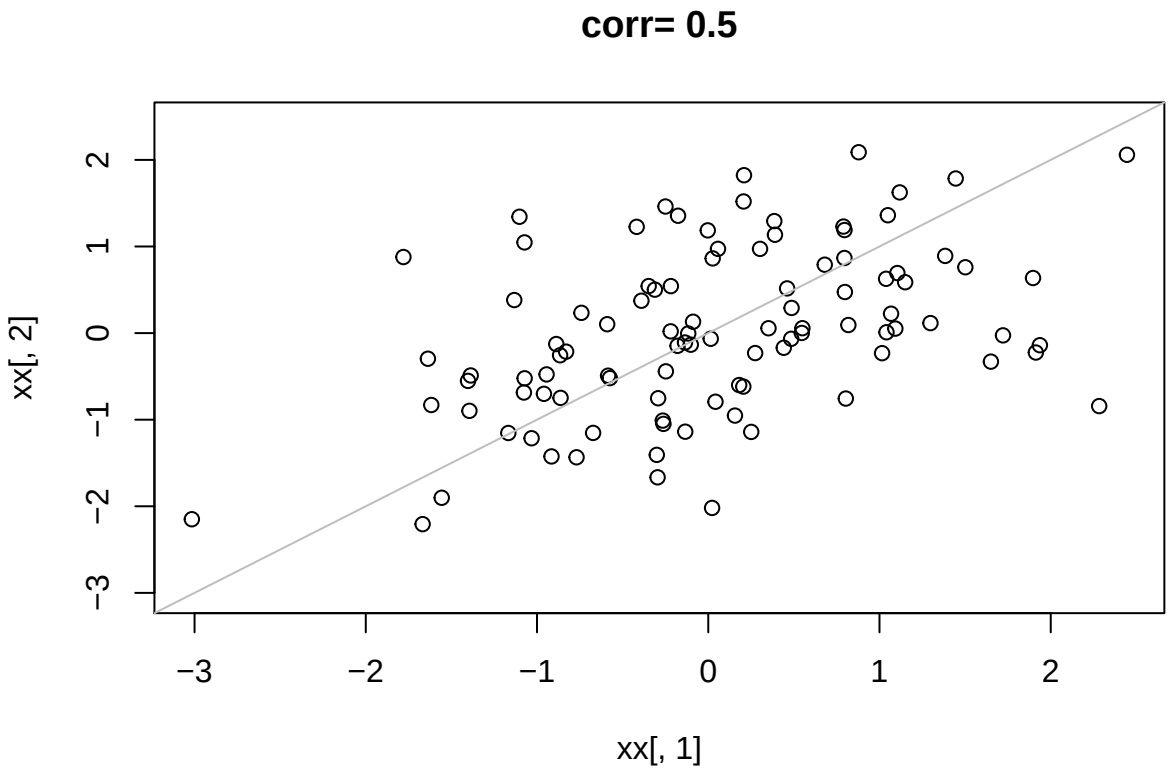
REPONSE Q4 :
Pour trouver la médiane dans chacun des 4 graphiques il suffit de regarder l'abscisse x correspondant à Fn(x)= 0.5
Sample covariance and sample correlation

La **corrélation** entre deux variables aléatoires X et Y est

$$\rho = \mathrm{Corr}(X, Y) = \frac{E[(X - E[X])(Y - E[Y])]}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}$$

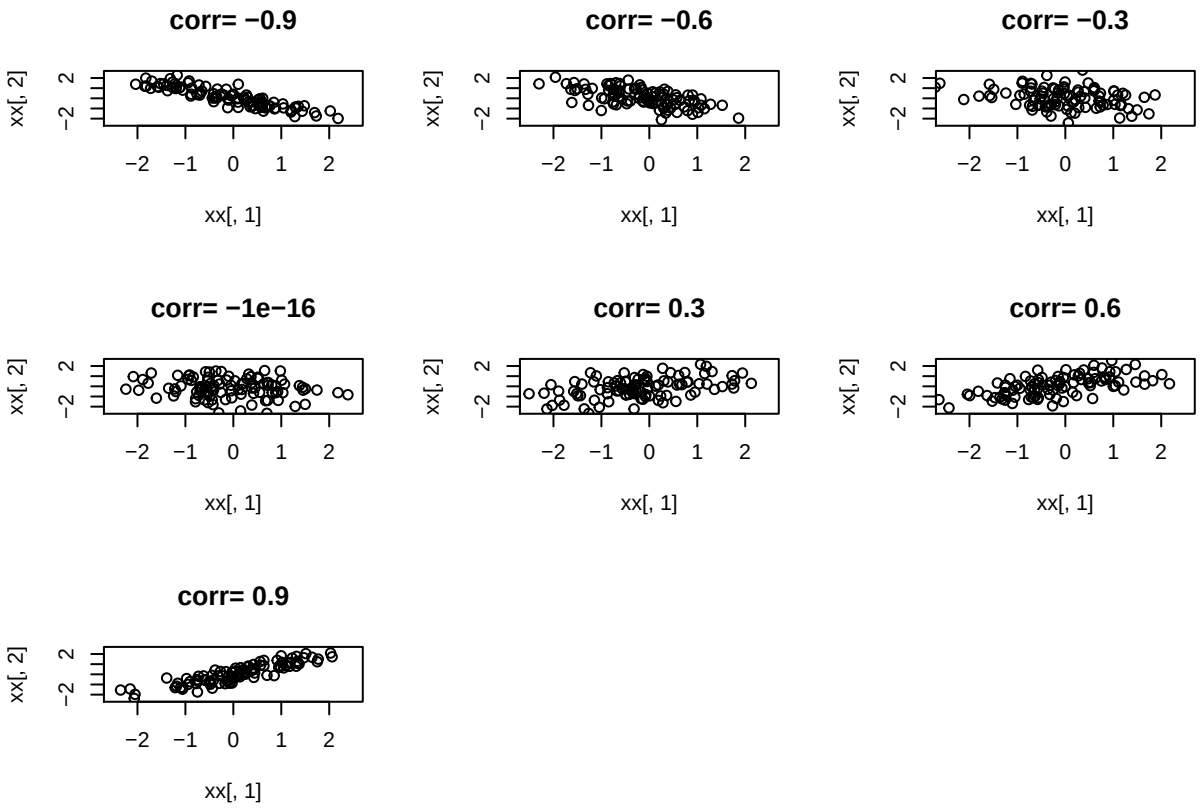
Le code suivant simule à partir d'une distribution normale bivariée avec une moyenne nulle, une variance unitaire et une corrélation ρ, $-1 \leq \rho \leq 1$:

```
#install.packages("MASS") # if not installed already
library(MASS)
?mvrnorm  # help
rho = 0.5
xx = mvrnorm(n=100, mu=c(0,0), Sigma = matrix(c(1,rho, rho, 1), ncol=2))
xx[1:10,]
xxlim = c(min(xx), max(xx))
plot(xx[,1], xx[,2], main = paste("corr=", rho), xlim = xxlim, yli=xxlim)
abline(a=0, b=1, col='gray') # add line y=x
```



5. Expérimentez avec une plage de valeurs de ρ et comparez les nuages de points. Décrivez l'effet de ρ.

```
vrho = seq(-0.9, 0.9, by=0.3)
k = length(vrho)
xxlim = c(-2.5, 2.5)
par(mfrow=c(3,3))
for (ik in 1:k){
  rho = vrho[ik]
  xx = mvrnorm(n=100, mu=c(0,0), Sigma = matrix(c(1,rho, rho, 1), ncol=2))
  plot(xx[,1], xx[,2], main = paste("corr=", signif(rho,1)), xlim = xxlim, yli=xxlim)
}
```



Le **coefficient de corrélation d'échantillon** de n paires d'observations $(x_1, y_1), \ldots, (x_n, y_n)$ est noté r et est donné par

$$r = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right)$$

où $\bar{x}$ et $\bar{y}$ sont les moyennes de l'échantillon et s_x et s_y sont les écarts types de l'échantillon.

La corrélation mesure une dépendance linéaire d'une manière indépendante de l'échelle. La **covariance** entre deux variables aléatoires X et Y est définie de manière similaire:

La **covariance** entre les variables aléatoires X et Y est

$$\mathrm{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])]$$

La **covariance d'échantillon** de n observations appariées $(x_1, y_1), \ldots, (x_n, y_n)$ est donnée par

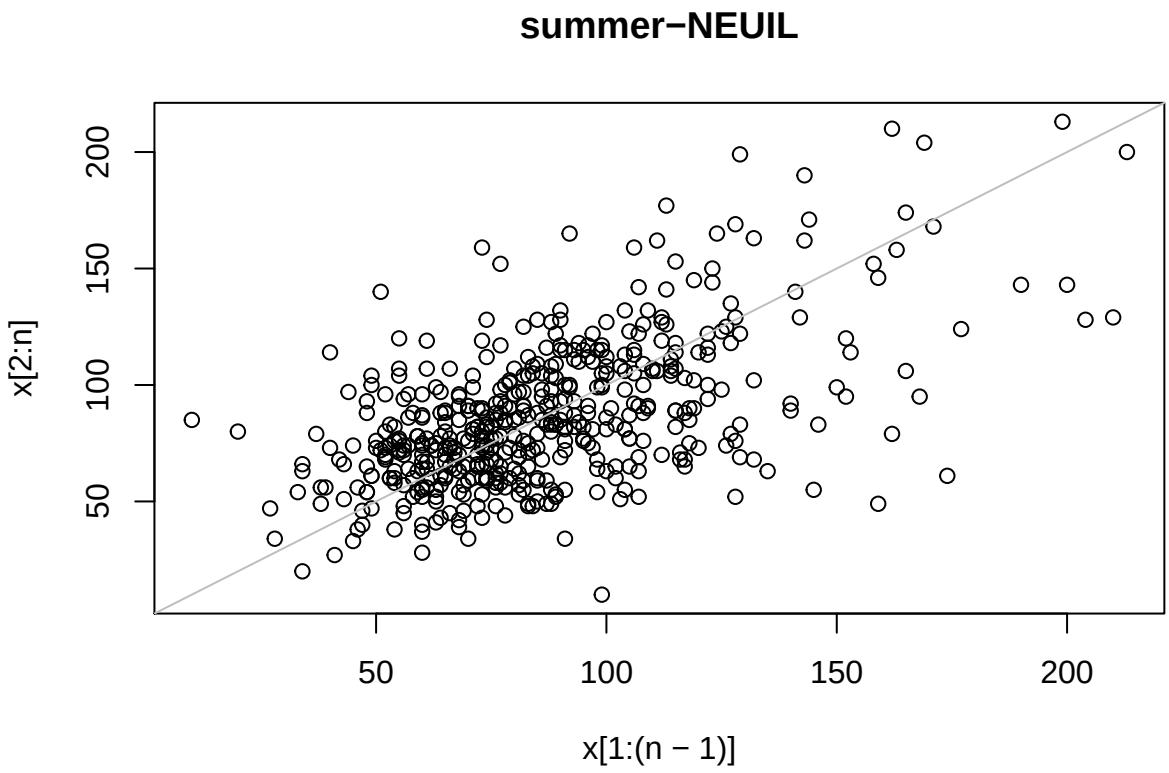
$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) = r s_x s_y$$

REPONSE Q5:
Selon certaines valeurs de la corrélation , il ya des droites et des nuages difformes de points. Plus la corrélation tend vers 1 en valeur absolue , plus les données de la loi binômiale s'alignent en une droite.

6. La production d'ozone peut persister pendant plusieurs jours. Le code suivant crée le nuage de points de x_t contre x_{t-1} pour toutes les valeurs de t pour l'ozone urbain en été.

```
x = ozone.summer$NEUIL
n = length(x)
```

```r
plot(x[1:(n-1)], x[2:n], main="summer-NEUIL")
abline(a=0, b=1, col="gray")
```



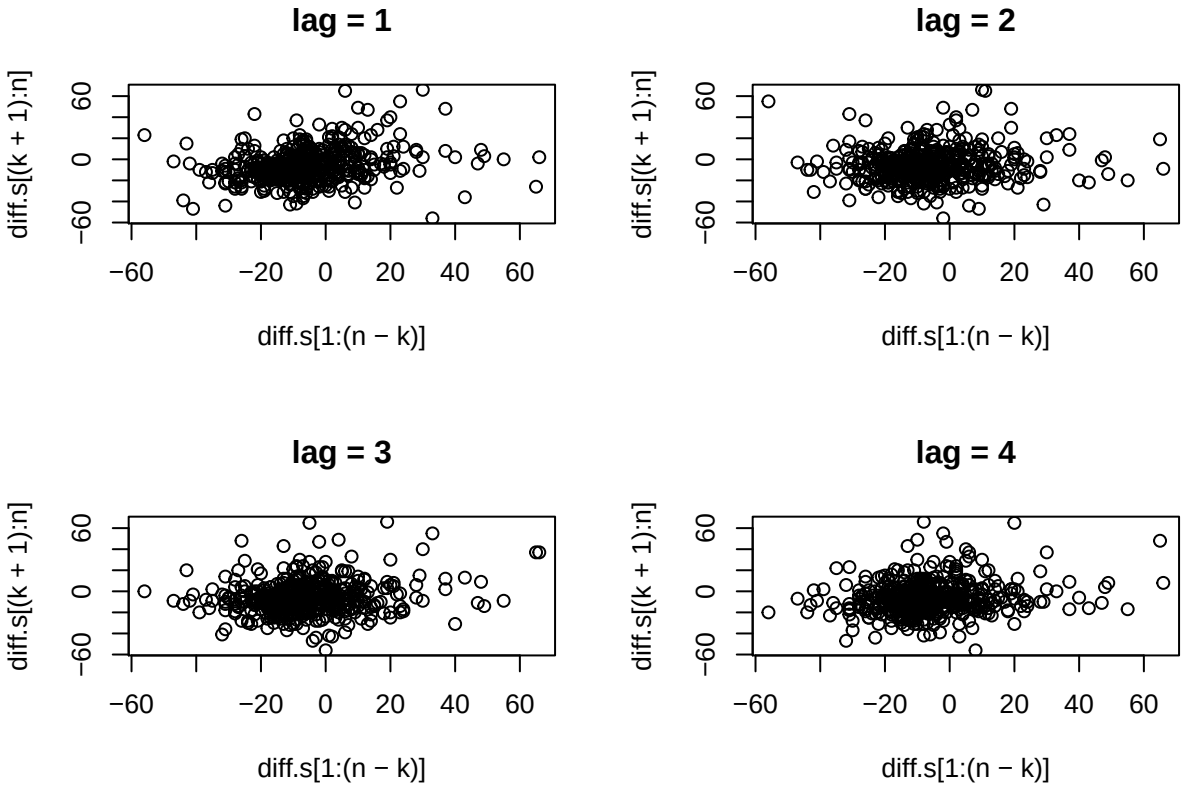
Qu'observez-vous? Estimez le coefficient de corrélation.
REPONSE Q6:
On observe une forte concentration entre 50 et 100 , cette concentration s'aligne bien sur une droite et représente donc une forte corrélation pour ces données (contrairement aux autres points dispersés, comme vu dans les premières questions).

```r
cor(x[1:n-1],x[2:n])
```

On estime le coefficient de corrélation entre chaque point et son suivant à 0.571106.

7. Explorez la dépendance pour la série différenciée $d_i, 1 \leq i \leq n$. Qu'observez-vous?

```r
diff.s = ozone.summer$NEUIL - ozone.summer$RUR.SE
n = length(diff.s)
par(mfrow=c(2,2))
for (k in 1:4){
  plot(diff.s[1:(n-k)], diff.s[(k+1):n], main = paste("lag =", k))
}
```




```
# no strong dependence
```

On observe peu de différences entre les 4 graphiques, la dépendance pour la série différenciée est très faible , il n'y a donc pas de dépendance.

Moyenne et phénomène de concentration

Nous allons montrer que la moyenne d'une variable aléatoire est un résumé déterministe d'une v.a., dont la qualité est contrôlée par la variance.

8. Rappeler l'inégalité de Bienaymé-Chebychef dans les cas Gaussien et Poisson.

   REPONSE Q8: Loi gaussienne

   $$X \sim \mathcal{N}(\mu, \sigma^2) : \forall a > 0, P\left(|X - \mu| \geq a\right) \leq \frac{\sigma^2}{a^2}$$

   Loi de poisson

   $$X \sim \mathcal{P}(\lambda) : \forall a > 0, P\left(|X - \lambda| \geq a\right) \leq \frac{\lambda}{a^2}$$

9. Estimer par Monte Carlo les probabilités de déviation d'une variable aléatoire de sa moyenne.
   (a) Exprimer $P\left(|X - \mu| \geq \delta\right)$ comme l'espérance d'une certaine variable aléatoire Z.
   (b) Simuler un échantillon de taille N $Z_1, Z_2, \ldots, Z_N$ de même loi que Z (dans le cas Gaussien, Pareto et Poisson) - on prendra N grand. Déterminer une estimation de $P\left(|X - \mu| \geq \delta\right)$.

      Pouvez vous déterminer la précision de cette estimation?

(c) Comparer avec les bornes obtenues par Bienaymé Chebychev pour plusieurs δ. Faites varier σ.

(d) Dans le cas Gaussien et Poisson, comparer les estimations Monte-Carlo de $P(X - \mu \geq \delta)$ avec les bornes données par les inégalités Chernoff pour plusieurs δ et σ (cf. cours).

REPONSE Q9:
9a)

$$P\left(|X-\mu| \geq \delta\right) = E\left(1_{|X-\mu|\geq\delta}\right) = E(Z)$$

(Chernoff:

$$P\left(X \geq a\right) = e^{-ta} E\left(e^{tX}\right)$$

)

9b)c)d) REGARDER LE RMD POUR LES RESULTATS (conversion en latex problématique)

On obtient des probabilités qui tendent vers 0 à mesure que les paramètres augmentent.

```r
library(Pareto)
library(rmutil)
```

```
##
## Attaching package: 'rmutil'
```

```
## The following object is masked from 'package:stats':
##
##     nobs
```

```
## The following objects are masked from 'package:base':
##
##     as.data.frame, units
```

```r
estimatemc<- function(N, delta, mu, sigma, a, alpha, lambda) {
  XN <- list("Gauss" = rnorm(N, mu, sigma), "Pareto" = rpareto(N, a, alpha), "Poisson" = rpois(N, l
  XNmoy <- list("Gauss" = mu, "Pareto" = alpha*a/(alpha - 1), "Poisson" = lambda)
  ZN <- list()
  ZNmoy <- list()
  for (distrib in names(XN)) {
    XNi <- XN[[distrib]]
    XNimoy <- XNmoy[[distrib]]
    ZN[[distrib]] <- unlist(lapply(XNi, function(xi) { if (abs(xi-XNimoy) >= delta) { return(1) } e
    ZNmoy[[distrib]] <- (ZN[[distrib]])
  }
  return (list("XN" = XN, "XN_moy" = XNmoy, "ZN" = ZN, "ZN_moy" = ZNmoy))
}
N <- 1e5
estimation <- estimatemc(N, delta=1, mu=0, sigma=1, a=1.0, alpha=2.5, lambda=1)
```

```
markovsup <- function(Z, N, eps) {
  return (var(Z) / (eps * N));
}
eps <- 1e-4

XN <- estimation[["XN"]]
ZN <- estimation[["ZN"]]
for (distrib in names(XN)) {
  print(paste(distrib, ":", markovsup(ZN[[distrib]], N, eps)))
}

for (delta in c(1, 5, 10)) {
  for (sigma in c(1, 10, 100)) {
    estimation <- estimatemc(N, delta, mu=0, sigma, a=1.0, alpha=2.5, lambda=1)
    XN <- estimation[["XN"]]
    ZN <- estimation[["ZN"]]
    print("------------------------------")
    print(paste("Pour delta=", delta, " et sigma=", sigma, sep=""))
    for (distrib in names(XN)) {
      print(paste(distrib, ":", markovsup(ZN[[distrib]], N, eps)))
    }
  }
}
```

10. Simuler un échantillon de taille $n = 20$ pour les lois de Gauss et de Poisson *(choisir σ, λ approprié)*

    (a) Calculer les bornes de Chernoff dans le cas échantillon pour $\bar{X}_n$. Faites varier $n = 20, 100, 1000$.

    (b) En déduire un estimateur de μ et λ respectivement.

    REPONSE Q10 :

    REGARDER LE RMD

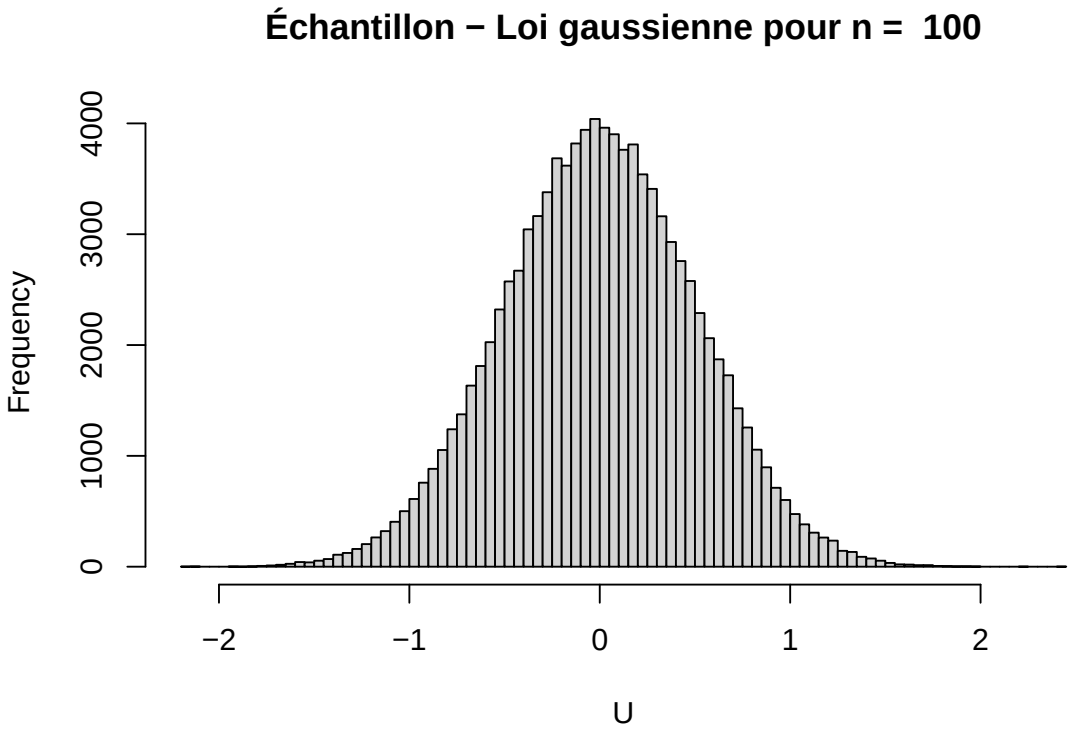
    “‘r mu <- 1 sigma <- 2 N <- 1000 ns <- c(100)

    moyempirique <- function(echantillons) ret <- apply(echantillons, 1, mean) return(ret)

    varempirique <- function(xs) n <- length(xs) moy <- mean(xs) varemp <- 0

    for (i in 1:n) varemp <- varemp + (xs[i] - moy)**2 varemp <- varemp / (n - 1) return(varemp)

    for (n in ns) g <- matrix(rnorm(N * n, mu, sigma), N, n) moyemp <- moyempirique(g) varemp <- apply(g, 1, varempirique) U <- (g - moyemp) / varemp hist(U, breaks = n, main = paste("Échantillon - Loi gaussienne pour n = ", n)) “‘



"'r range <- c(20,100,1000)

borneno <- c() for (i in 1:3) law <- rnorm(range[i],mean=0, sd = 1) varian <- varempirique(law) borneno[i] <- 10/(2*varian) borneno[i] = exp(-borneno[i])

bornepo <- c() for (i in 1:3) law <- rpois(range[i],16) varian <- varempirique(law) bornepo[i] <- 10/(2*varian) bornepo[i] = exp(-bornepo[i]) tableau <- data.frame(range, borneno, bornepo) colnames(tableau) = c("Taille de l'échantillon", "N(0,1)", "P(16)") tableau "'

Pour la loi normales, les valeurs tendent vers 0; vers 0.74 pour la loi de poisson.

11. Simuler un échantillon de taille $n = 20$ d'une loi de Cauchy $\mathcal{C}(\theta)$ de densité? $f(x, \theta) = \frac{1}{\pi} \frac{1}{1+(x-\theta)^2}$.

    (a) Calculer la moyenne empirique $\bar{X}_n$. Faites varier la taille de l'échantillon $n = 20, 100, 1000$ et 10000. Qu'en déduire ?

    (b) Expliquer ce comportement. On se rappellera notamment que la fonction caractéristique s'écrit $\phi_\theta(t) = \exp(i\theta t - |t|)$.

    (c) Quelle est la médiane d'une loi de Cauchy $\mathcal{C}(\theta)$? En déduire un estimateur de θ pour $n = 20, 100, 1000$.

    REPONSE Q11:

    REGARDER LE RMD POUR LES RESULTATS (conversion en latex problématique)

    a) Pour la loi de Cauchy, on obtient les résultats suivants :

```r
theta <- 0
for (n in c(20, 100, 1000, 10000)) {
  cauchy <- rcauchy(n, location=theta, scale=1)
```

```
  m      <- mean(cauchy)
  print(paste("n=", n, " ; la moyenne empirique calculée est: ", m, sep=""))
}
```

La moyenne empirique donne des valeurs très différentes selon n, et ne semble pas converger.

b) Une variable aléatoire X suivant une loi de Cauchy n'admet pas d'espérance. Donc le théorème centrale limite ne s'applique pas : il n'y a pas d'espérance, donc la moyenne empirique ne converge pas. Ceci s'explique par le fait que la probabilité d'obtenir une valeur éloigné de la médiane est trop elévé pour que la moyenne converge.

c)

Pour la loi de Cauchy on observe que la médiane tend vers θ -> VOIR LE RMD :

“‘r theta <- 0 for (theta in c(-1, 0, 1)) print("——") print(paste("theta=", theta, sep="")) for (n in c(20, 100, 1000, 10000)) cauchy <- rcauchy(n, location=theta, scale=1) sorted <- sort(cauchy) print(paste("la médiane de l'échantillon n=", n, " vaut:", sorted[n / 2 + 1], sep="")) “‘