

PROJET STATISTIQUE

Bouhouch Roua

Zoé Marinez

EXERCICE 1

A. INFORMATIONS GÉNÉRALES

B. CORRÉLATION ET RÉGRESSION LINÉAIRE

Question 1 :

TYPE DE VARIABLE ?

Question 2 :

Quel est le nombre d'heures d'observation de l'échantillon ?

Quel est le nombre d'heures où les particules ont été mesurées dans toutes les stations?

Question 3 :

```
#3
PM10Obs <- c()
AzoteObs <- c()
#colonne azote : 2 4 6 9 12 14 16
#colonne PM10 : 3 5 8 11 13 15 17

for (i in 1:length(Station.Polluant)) {
  if (is.na(Air[i, 2] & Air[i, 4] & Air[i, 6] & Air[i, 9] & Air[i, 12] & Air[i, 12] & Air[i, 14] & Air[i, 16])) {
    AzoteObs = c(AzoteObs, FALSE)
  }
  else {
    AzoteObs = c(AzoteObs, TRUE)
  }
}

for (i in 1:length(Station.Polluant)) {
  if (is.na(Air[i, 3] & Air[i, 5] & Air[i, 8] & Air[i, 11] & Air[i, 13] & Air[i, 15] & Air[i, 14] & Air[i, 17]))
  {
    PM10Obs = c(PM10Obs, FALSE)
  }
  else {
    PM10Obs = c(PM10Obs, TRUE)
  }
}

table(AzoteObs)
```

```

table(PM10Obs)
#L'azote a été observé dans toutes les stations pendant 534 heures
#Les particules ont été observées dans toutes les stations pendant 499 heures

#4
na.omit(Lyon.Centre.Ozone)

mean(Lyon.Centre.Ozone) #Moyenne empirique

varE <- function(x) {l<-length(x); var(x)*(l-1)/l}
varE(Lyon.Centre.Ozone) #Variance empirique

var(Lyon.Centre.Ozone) #Variance empirique non-biaisée

quantile(Lyon.Centre.Ozone, probs = 0.25) #Quantile à 25%

(reste de l'exercice à la page 8)

```

EXERCICE 2

Question 1:

On va tout d'abord procéder à la construction de vecteur **U** qui suit une loi uniforme sur l'intervalle [0;1] avec taille **N=1000**. Pour cela on utilise la fonction [runif](#):

```
U<-c(runif(1000,0,1))
```

On se sert ensuite de la fonction log pour construire le **vecteur x de taille N** qui suit une loi qu'on souhaite déterminer tel que $X = -\ln(U)$:

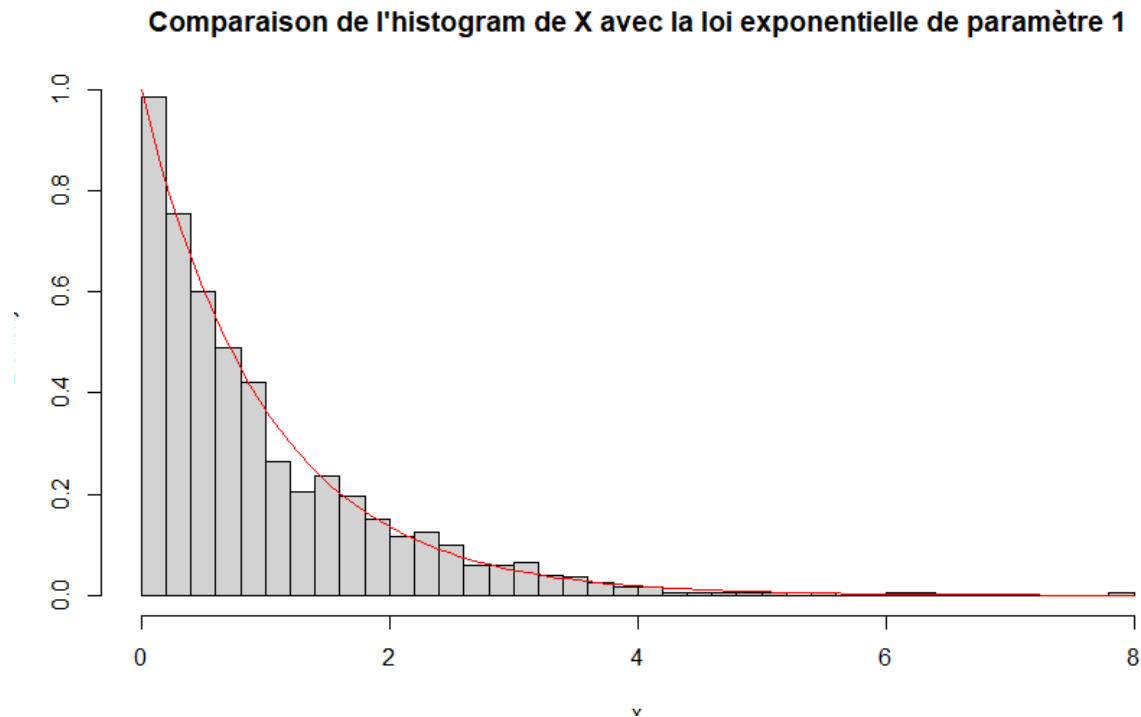
```
x<-c(-log(U))
```

Question 2:

Dans cette partie de l'analyse on va faire une comparaison qui va nous permettre de faire un hypothèse sur la loi de X. Pour cela on réalise une comparaison entre l'histogramme associé et la densité de probabilité qu'on a choisit dans ce cas là avec un paramètre de 1:

```
{hist(x, prob=T, breaks="FD",
      main="comparaison de l'histogram de X avec la loi exponentielle de paramètre 1")}
curve(dexp(x, rate = 1), add=T ,col="red")
```

on obtient les observations suivantes:



Vu les différences observées, il est vraisemblable que X suit une loi exponentielle. Les données semblent s'ajuster à la courbe de la loi exponentielle ce qui nous permet de faire un hypothèse sur la loi de X. Soit alors "**L**" une loi de probabilité exponentielle **de paramètre 1** étant possiblement *en adéquation* avec la loi inconnue de **X**. On considère les hypothèses :

H_0 : "X suit la loi L" contre H_1 : "X ne suit pas la loi L"

Pour pouvoir décider du rejet de H_0 :

- On crée le tableau de: C_1, \dots, C_k de classes et effectifs de la variable X
- On considère une variable R qui suit la loi exponentielle définie avec les paramètres estimés (dans notre cas la loi L)
- On considère une var R suivant cette loi définie avec les paramètres estimés.
- Pour tout $i \in \{1, \dots, k\}$, on calcule la probabilité :

$$p_i = P(R \in C_i)$$
- On effectue le test d'adéquation à l'aide de la fonction **chisq.test** en prenant comme p les probabilités obtenues dans la partie précédente p_1, \dots, p_k

On commence donc par la classification de la variable X. Pour trouver la bonne classification on recherche le minimum et maximum de la variable X.

min(x)
max(x)

cela nous donne:

```
> min(x)
[1] 0.0004420587
> max(x)
[1] 7.939387
> |
```

par rigueur on va donc choisir de regrouper ces valeurs dans $k=6$ classes: $C_0...C_k$
 $C=\{"]0,1["","]1,2["","]2,3["","]3,4["","]4,5["","]5,10[""]\}$ de sorte que on a une fréquence supérieure à 5 dans chacune de nos classes

Pour cela on effectue le script suivant:

```
categorieR<-cut(x,
               breaks = c(0,1,2,3,4,5,10),
               labels = c("]0,1["","]1,2["","]2,3["","]3,4["","]4,5["","]5,10["")
)
mydataR<-data.frame(x,check.names = TRUE)
mydataR<-cbind(mydataR,categorieR)
TCR<-table(categorieR)
```

En affichant le tableau de fréquence on obtient:

```
categorieR
 ]0,1]  ]1,2]  ]2,3]  ]3,4]  ]4,5]  ]5,10]
   623    242    88    23    18     6
> |
```

Soit R une variable suivant la loi de probabilité exponentielle " L " de paramètre $\lambda=1$.
On effectue les calculs des probabilités que R appartient à chacune des classes $C_0...C_k$. On Obtient alors les résultats suivants:

```
p(0<R<1)=0.6321
p(1<R<2)=0.2325
p(2<R<3)=0.0855
p(3<R<4)=0.0315
p(4<R<5)=0.0116
p(5<R<10)=0.0068
```

On effectue ensuite le test avec la commande **chisq.test** en prenant comme p le vecteur des probabilités de la loi L :

```
TCR2=as.data.frame(TCR, row.names = NULL)
```

```
nb=TCR2 %>% pull(Freq)
proba=c(0.6321,0.2325,0.0855,0.0315,0.0116,0.0068)
chisq.test(nb, p = proba)$p.value
```

on Obtient ainsi le résultat suivant:

```
[1] 0.05930844
> |
```

comme p-1 valeur>0.05 on ne peut donc pas rejeter l'hypothèse H_0 Et on ne peut donc pas confirmer le fait que X ne suit pas une loi exponentielle de paramètre 1.

On effectue une boucle pour compter combien de fois on garde ou on rejette l'hypothèse H_0 on utilise un compteur b qu'on incrémente à chaque fois qu'on conserve l'hypothèse H_0

On obtient alors:

```
[1] 94
```

On remarque que sur 100 simulation on garde l'hypothèse H_0 96 fois ce qui affirme cette hypothèse fortement.

Question 4:

Dans cette partie de l'exercice on va procéder avec la même logique que la partie 1. On va tout d'abord procéder à la construction de deux vecteurs **N** et **B** qui suivent une loi uniforme sur l'intervalle [0;1] avec taille **N=1000**. Pour cela on utilise la fonction [runif](#). On constitue ensuite le vecteur Y à partir de ces deux vecteurs:

```
N<-c(runif(1000,0,1))
B<-c(runif(1000,0,1))
Y<-(sqrt(-2*log(N)))*cos(2*pi*B)
```

On teste le minimum et maximum de Y et on obtient:

```
> min(Y)
[1] -3.136911
> max(Y)
[1] 3.380087
> -log(u)
```

Pour mieux observer la variable Y on construit ensuite son tableau de fréquences:

```
categorieY<-cut(Y,
  breaks = c(-4,-2,0,2,4),
  labels = c("]-4,-2]", "]-2,0]", " ]0,2]", " ]2,4]"))
)
mydataY<-data.frame(Y,check.names = TRUE)
mydataY<-cbind(mydataY,categorieY)
TCR3<-table(categorieY)
TCR3
```

Cela revoit:

```

categorieY
]-4,-2]  ]-2,0]  ]0,2]  ]2,4]
    13     504    464    19

```

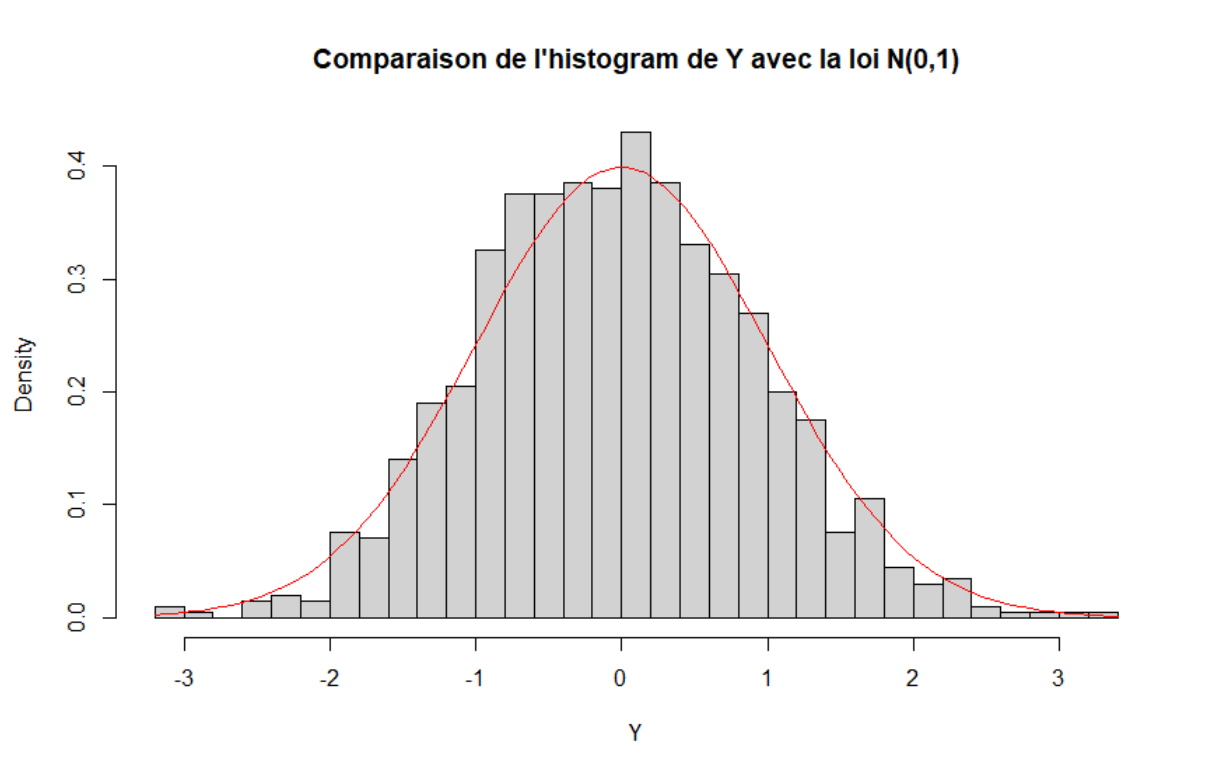
Cette distribution de données nous fait penser à une loi normale . On va réaliser une comparaison qui va nous permettre de faire un hypothèse sur la loi de Y. Pour cela on réalise une comparaison entre l'histogramme associé et la densité de probabilité qu'on va choisir , soit la loi $N(0,1)$

```

{hist(Y, prob=T, breaks="FD",
      main="Comparaison de l'histogram de X avec la loi exponentielle de paramètre 1")
 curve(dnorm(x, mean = 0, sd=1), add=T ,col="red")}

```

On obtient alors le résultat suivant:



Vu les différences observées, il est vraisemblable que Y suit une loi normale $N(0,1)$. Les données semblent s'ajuster à la courbe de la loi Normale ce qui nous permet de faire un hypothèse sur la loi de Y. Soit alors "**N**" une loi de probabilité normale **de paramètres $E=0$ et $V=1$** étant possiblement *en adéquation* avec la loi inconnue de Y. On considère les hypothèses :

H_0 : "X suit la loi N" contre H_1 : "X ne suit pas la loi N"

Soit S une variable suivant la loi de probabilité Normale N.

On effectue les calculs des probabilités que R appartient à chacune des classes $C_0 \dots C_k$.

On Obtient alors les résultats suivants:

```
p(-4<S<-2)=0.0228
p(-2<S<0)=0.4772
p(0<S<2)=0.4772
p(2<S<4)=0.0228
```

On effectue ensuite le test avec la commande **chisq.test** en prenant comme p le vecteur des probabilités de la loi **N**:

```
TCR4=as.data.frame(TCR3, row.names = NULL)
nb2=TCR4 %>% pull(Freq)
proba2=c(0.0228,0.4772 ,0.4772 ,0.0228)
chisq.test(nb2, p = proba2)$p.value
```

cela renvoie:

```
> chisq.test(nb2, p = proba2)$p.value
[1] 0.0815275
```

comme $p\text{-}1 \text{ valeur} > 0.05$ on ne peut donc pas rejeter l'hypothèse H_0 Et on ne peut donc pas confirmer le fait que Y ne suit pas la loi normale N.

On effectue une boucle pour compter combien de fois on garde ou on rejette l'hypothèse H_0 on utilise un compteur c qu'on incrémente à chaque fois qu'on conserve l'hypothèse H_0

On obtient alors:

```
[1] 97
> |
```

On conserve au total 97 fois l'hypothèse H_0 ce qui nous confirme bien que cette hypothèse est vraie.

#EXERCICE 1

#Chargement du jeu de données

```
Air<-read.delim2("http://tinyurl.com/y39an7ef/DATA28574.csv",na.strings="-")
attach(Air)
```

A - INFORMATIONS GÉNÉRALES

#1

```
typeof(Station.Polluant)
typeof(A7.Sud.lyonnais.Dioxyde.d.azote)
#Les variables sont toutes quantitatives et continues
```

#2

```
Tot <- length(Station.Polluant)
NaObs <-sum(is.na(A7.Sud.lyonnais.Dioxyde.d.azote) |
  is.na(A7.Sud.lyonnais.Particules.PM10) |
  is.na(Villeurbanne.Place.Grandclément.Dioxyde.d.azote) |
  is.na(Villeurbanne.Place.Grandclément.Particules.PM10) |
```



```

is.na(Est.lyonnais...Saint.Exupéry.Dioxyde.d.azote) |
is.na(Est.lyonnais...Saint.Exupéry.Ozone) |
is.na(Est.lyonnais...Saint.Exupéry.Particules.PM10) |
is.na(Lyon.Centre.Dioxyde.d.azote) |
is.na(Air$Lyon.Centre.Ozone) |
is.na(Lyon.Centre.Particules.PM10) |
is.na(Lyon...Tunnel.Croix.Rousse...Sortie.Rhône.Dioxyde.d.azote) |
is.na(Lyon...Tunnel.Croix.Rousse...Sortie.Rhône.Particules.PM10) |
is.na(Lyon.Périphérique.Dioxyde.d.azote) |
is.na(Lyon.Périphérique.Particules.PM10) |
is.na(Lyon.Trafic.Jaurès.Dioxyde.d.azote) |
is.na(Lyon.Trafic.Jaurès.Particules.PM10))

```

Obs <- Tot - NaObs #Total heures observées

#3

```
PM10Obs <- c()
```

```
AzoteObs <- c()
```

```

for (i in 1:length(Station.Polluant)) {
  if (is.na(Air[i, 2] & Air[i,4] & Air[i, 6] & Air[i, 9] & Air[i, 12] & Air[i, 12] & Air[i, 14] & Air[i, 16])) {
    AzoteObs = c(AzoteObs, FALSE)
  }
  else {
    AzoteObs = c(AzoteObs, TRUE)
  }
}

```

```

for (i in 1:length(Station.Polluant)) {
  if (is.na(Air[i, 3] & Air[i,5] & Air[i, 8] & Air[i, 11] & Air[i, 13] & Air[i, 15] & Air[i, 14] & Air[i, 17]))
  {
    PM10Obs = c(PM10Obs, FALSE)
  }
  else {
    PM10Obs = c(PM10Obs, TRUE)
  }
}

```

```
table(AzoteObs, PM10Obs)
```

#4

```
LyonCO <- na.omit(Lyon.Centre.Ozone)
```

```
sum(is.na(Air$Lyon.Centre.Ozone))
```

#LyonCO & Lyon.Centre.Ozone sont égaux car il n'y a pas de valeurs nulles dans cette colonne,

#On utilisera donc Air\$Lyon.Centre.Ozone pour la suite de cette question

```

mean(Air$Lyon.Centre.Ozone) #Moyenne empirique

varE <- function(x) {l<-length(x); var(x)*(l-1)/l}
varE(Air$Lyon.Centre.Ozone) #Variance empirique

var(Air$Lyon.Centre.Ozone) #Variance empirique non-biaisée

quantile(Air$Lyon.Centre.Ozone, probs = 0.25) #Quantile à 25%

max(Air$Lyon.Centre.Ozone)

#5
moyA7DA <- c()
moyA7DANa <- c()
moyA7PM10 <- c()
moyA7PM10Na <- c()

moyVGCDa <- c()
moyVGCDANa <- c()
moyVGCPM10 <- c()
moyVGCPM10Na <- c()

moyEstExDA <- c()
moyEstExDANa <- c()
moyEstExO <- c()
moyEstExONa <- c()
moyEstExPM10 <- c()
moyEstExPM10Na <- c()

moyLCDA <- c()
moyLCDANa <- c()
moyLCO <- c()
moyLCONa <- c()
moyLCPM10 <- c()
moyLCPM10Na <- c()

moyITCRDA <- c()
moyITCRDANa <- c()
moyITCRPM10 <- c()
moyITCRPM10Na <- c()

moyLPeriphDA <- c()
moyLPeriphDANa <- c()
moyLPeriphPM10 <- c()
moyLPeriphPM10Na <- c()

moyLJJDA <- c()

```

```

moyLJJDA Na <- c()
moyLJJPM10 <- c()
moyLJJPM10Na <- c()

for(i in 0:123) {
  debut <- 1+6*i
  fin <- 6+6*i

  moyA7DA <- c(moyA7DA, mean(A7.Sud.lyonnais.Dioxyde.d.azote[debut:fin], na.rm =
TRUE))
  moyA7DANA <- c(moyA7DANA, sum(is.na(A7.Sud.lyonnais.Dioxyde.d.azote[debut:fin])))
  moyA7PM10 <- c(moyA7PM10, mean(A7.Sud.lyonnais.Particules.PM10[debut:fin], na.rm
= TRUE))
  moyA7PM10Na <- c(moyA7PM10Na,
sum(is.na(A7.Sud.lyonnais.Particules.PM10[debut:fin])))

  moyVGCDNA <- c(moyVGCDNA,
mean(Villeurbanne.Place.Grandclément.Dioxyde.d.azote[debut:fin], na.rm = TRUE))
  moyVGCDANA <- c(moyVGCDANA,
sum(is.na(Villeurbanne.Place.Grandclément.Dioxyde.d.azote[debut:fin])))
  moyVGCPM10 <- c(moyVGCPM10,
mean(Villeurbanne.Place.Grandclément.Particules.PM10[debut:fin], na.rm = TRUE))
  moyVGCPM10Na <- c(moyVGCPM10Na,
sum(is.na(Villeurbanne.Place.Grandclément.Particules.PM10[debut:fin])))

  moyEstExDA <- c(moyEstExDA,
mean(Est.lyonnais...Saint.Exupéry.Dioxyde.d.azote[debut:fin], na.rm = TRUE))
  moyEstExDANA <- c(moyEstExDANA,
sum(is.na(Est.lyonnais...Saint.Exupéry.Dioxyde.d.azote[debut:fin])))
  moyEstExO <- c(moyEstExO, mean(Est.lyonnais...Saint.Exupéry.Ozone[debut:fin], na.rm
= TRUE))
  moyEstExONa <- c(moyEstExONa,
sum(is.na(Est.lyonnais...Saint.Exupéry.Ozone[debut:fin])))
  moyEstExPM10 <- c(moyEstExPM10,
mean(Est.lyonnais...Saint.Exupéry.Particules.PM10[debut:fin], na.rm = TRUE))
  moyEstExPM10Na <- c(moyEstExPM10Na,
sum(is.na(Est.lyonnais...Saint.Exupéry.Particules.PM10[debut:fin])))

  moyLCDA <- c(moyLCDA, mean(Lyon.Centre.Dioxyde.d.azote[debut:fin], na.rm = TRUE))
  moyLCDANA <- c(moyLCDANA, sum(is.na(Lyon.Centre.Dioxyde.d.azote[debut:fin])))
  moyLCO <- c(moyLCO, mean(Lyon.Centre.Ozone[debut:fin], na.rm = TRUE))
  moyLCONa <- c(moyLCONa, sum(is.na(Lyon.Centre.Ozone[debut:fin])))
  moyLCPM10 <- c(moyLCPM10, mean(Lyon.Centre.Particules.PM10[debut:fin], na.rm =
TRUE))
  moyLCPM10Na <- c(moyLCPM10Na,
sum(is.na(Lyon.Centre.Particules.PM10[debut:fin])))

```

```

moyITCRDA <- c(moyITCRDA,
mean(Lyon...Tunnel.Croix.Rousse...Sortie.Rhône.Dioxyde.d.azote[debut:fin], na.rm =
TRUE))
moyITCRDANa <- c(moyITCRDANa,
sum(is.na(Lyon...Tunnel.Croix.Rousse...Sortie.Rhône.Dioxyde.d.azote[debut:fin])))
moyITCRPM10 <- c(moyITCRPM10,
mean(Lyon...Tunnel.Croix.Rousse...Sortie.Rhône.Particules.PM10[debut:fin], na.rm =
TRUE))
moyITCRPM10Na <- c(moyITCRPM10Na,
sum(is.na(Lyon...Tunnel.Croix.Rousse...Sortie.Rhône.Particules.PM10[debut:fin])))

moyLPeriphDA <- c(moyLPeriphDA, mean(Lyon.Périphérique.Dioxyde.d.azote[debut:fin],
na.rm = TRUE))
moyLPeriphDANa <- c(moyLPeriphDANa,
sum(is.na(Lyon.Périphérique.Dioxyde.d.azote[debut:fin])))
moyLPeriphPM10 <- c(moyLPeriphPM10,
mean(Lyon.Périphérique.Particules.PM10[debut:fin], na.rm = TRUE))
moyLPeriphPM10Na <- c(moyLPeriphPM10Na,
sum(is.na(Lyon.Périphérique.Particules.PM10[debut:fin])))

moyLJJDA <- c(moyLJJDA, mean(Lyon.Trafic.Jaurès.Dioxyde.d.azote[debut:fin], na.rm =
TRUE))
moyLJJDANa <- c(moyLJJDANa,
sum(is.na(Lyon.Trafic.Jaurès.Dioxyde.d.azote[debut:fin])))
moyLJJPM10 <- c(moyLJJPM10, mean(Lyon.Trafic.Jaurès.Particules.PM10[debut:fin],
na.rm = TRUE))
moyLJJPM10Na <- c(moyLJJPM10Na,
sum(is.na(Lyon.Trafic.Jaurès.Particules.PM10[debut:fin])))
}

dates <- rep(c("nuit", "matin", "après-midi", "soir"), 31)
dates

dft = data.frame(date = dates,
  A7Sud.Dioxide.Azote = moyA7DA,
  A7Sud.PM10 = moyA7PM10,
  Villeurbanne.Dioxyde.Azote = moyVGCDa,
  Villeurbanne.PM10 = moyVGCPM10,
  #Est.St.Exupéry.Dioxyde.Azote = moyEstExDA,
  Est.St.Exupéry.Ozone = moyEstExO,
  Est.St.exupéry.PM10 = moyEstExPM10,
  Lyon.Centre.Dioxyde.Azote = moyLCDA,
  Lyon.Centre.Ozone = moyLCO,
  #Lyon.Centre.PM10 = moyLCPM10,
  Tunnel.Croix.Rousse.Dioxyde.Azote = moyITCRDA,
  Tunnel.Croix.Rousse.PM10 = moyITCRPM10,
  Périphérique.Dioxyde.Azote = moyLPeriphDA,
  #Périphérique.PM10 = moyLPeriphPM10,

```

```

#Jean.Jaurès.Dioxyde.Azote = moyLJJDA,
Jean.Jaurès.PM10 = moyLJJPM10,
stringsAsFactors = FALSE)

attach(dft)

#           A - CORRÉLATIONS ET RÉGRESSION LINÉAIRES

#1
cov(dft[,2:12])
cor(dft[,2:12])

#2
y <- Tunnel.Croix.Rousse.Dioxyde.Azote
x <- A7Sud.Dioxide.Azote

cor.test(x, y, method="pearson")

#3
lmll=lm(y~x)
lmll
summary(lmll)
plot(y~x, main="Régression linéaire des mesures de moyenne par tranches de 6H du
Dioxyde d'Azote à la Croix-Rousse en fonction de celles de Dioxyde d'Azote sur l'A7")
abline(lmll, col="orange") #droite de régression

#Call:
# lm(formula = y ~ x)

#Residuals:
# Min      1Q  Median      3Q      Max
#-35.345 -11.891 -0.629  8.686 69.283

#Coefficients:
# Estimate Std. Error t value Pr(>|t|)
#(Intercept) 34.8808    4.3262  8.063 5.83e-13 ***
# x          0.7372    0.0680 10.841 < 2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#Residual standard error: 18.59 on 122 degrees of freedom
#Multiple R-squared:  0.4907,    Adjusted R-squared:  0.4865
#F-statistic: 117.5 on 1 and 122 DF, p-value: < 2.2e-16

#4
n <- length(y)
moyenne <- mean(y)
sigma <- sqrt(var(y))

```

```

delta95 = (qnorm(0.975)*sigma)/sqrt(n)

#Intervale de confiance associé
IC95 = c(moyenne-delta95, moyenne+delta95) #Un vecteur [moyenne-delta95,
moyenne+delta95]
cat("IC - 95% : [", IC95[1], ", ", IC95[2], "]")

#5
gofTest(lmll$residuals, distribution = "norm", test = "chisq")$p.value

z <- A7Sud.PM10
z1 <- log(z)
lmllZ=lm(z1~z)
gofTest(lmllZ$residuals, distribution = "norm", test = "chisq")$p.value

#6
lmll6=lm(x~z1)
lmll6
summary(lmll6)

#Call:
# lm(formula = x ~ z1)

#Residuals:
# Min      1Q  Median      3Q      Max
#-52.235 -12.767  -0.444  13.050  57.333

#Coefficients:
# Estimate Std. Error t value Pr(>|t|)
#(Intercept) -30.818    10.860  -2.838  0.00532 **
# z1          27.524     3.294   8.355  1.22e-13 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#Residual standard error: 19.74 on 122 degrees of freedom
#Multiple R-squared:  0.3639,    Adjusted R-squared:  0.3587
#F-statistic: 69.8 on 1 and 122 DF, p-value: 1.218e-13

plot(x~z1, main="Régression linéaire des mesures de moyenne par tranches de 6H du
Dioxyde d'Azote sur l'A7
en fonction des logarithmes de celles de particules PM10 sur l'A7")
abline(lmll6, col="orange")

n2 <- length(z1)
moyenne2 <- mean(z1)
sigma2 <- sqrt(var(z1))
delta802 = (qnorm(0.9)*sigma2)/sqrt(n2)

```

```
#Intervale de confiance associé
IC80 = c(moyenne2-delta802, moyenne2+delta802) #Un vecteur [moyenne-delta95,
moyenne+delta95]
cat("IC - 80% : [", IC80[1], ", ", IC80[2], "]")
abline(v=c(IC80[1], IC80[2]), col="blue")
```