RAPPORT STATISTIQUE DES POLLUANTS SUR LA BANLIEU LYONNAISE

Sujet n°97177

THINOT Edouard & SIONI Farès

Semestre automne 2021/2022
Licence 2 Informatique
Statistiques pour l'informatique
17/12/2021

EXERCICE 1

PARTIE A

QUESTION 1 :

STATIONS	TYPE DE LA VARIABLE
Lyon Gerland :	
- Dioxyde d'azote - Monoxyde d'azote - Ozone - Particules PM10	quantitative discrète
Villeurbanne :	
- Dioxyde d'azote - Monoxyde d'azote	quantitative discrète
- Particules PM10	quantitative continue
Aéroport Saint-Exupéry :	
- Dioxyde d'azote - Monoxyde d'azote - Ozone - Particules PM10	quantitative discrète
Lyon centre :	
- Dioxyde d'azote - Monoxyde d'azote - Ozone - Particules PM10	quantitative discrète
- Particules PM2.5	quantitative continue

Tunnel Croix-Rousse :	
- Dioxyde d'azote - Monoxyde d'azote	quantitative discrète
- Particules PM10	quantitative continue
Lyon périphérique :	
- Dioxyde d'azote - Monoxyde d'azote	quantitative discrète
- Monoxyde de carbone - Particules PM10	quantitative continue

---

QUESTION 2 :

```
j = nrow(Air)
cat("Il y a", j, "jours observés")
```

Le nombre de jours observés de l'échantillon est ***730***.

```
CountMonoxyteAzoteDays()
```

Le monoxyde d'azote a été observé dans toutes les stations ***546*** jours.

QUESTION 3 :

Les 6 observations de polluants à certaines stations ayant strictement moins de 25 jours de non-observation sont :

STATION	POLLUANT
Villeurbanne Place Grand Clément	Particules PM10

Lyon Centre	Ozone
Lyon Périphérique	Dioxyde d'azote
Lyon Périphérique	Monoxyde de carbone
Lyon Périphérique	Monoxyde d'azote
Lyon Périphérique	Particules PM10

QUESTION 4 :

Monoxyde de carbone sur le périphérique lyonnais :

CALCUL	VALEUR
Moyenne	**304,053**
Variance biaisée	**20 800,83**
Variance non-biaisée	**20 829,67**
3ème quartile	**364**

QUESTION 5 :

```
LyonPM10 = select(Air, c(5, 8, 12, 16, 20, 24))
```

LyonPM10 est un dataframe contenant toutes les colonnes mesurant le PM10.

```
PM10Moyen = round(rowMeans(LyonPM10, na.rm = TRUE), digits=3)
```

PM10Moyen est la moyenne pour chaque ligne de LyonPM10 en omettant les valeurs NA.

```
cat("Le seuil de 50mg/m cube est dépassé ", round(1-ecdf(PM10Moyen)(50),
digits=3), "% du temps sur les",j , "jours observés")

cat("Le seuil de 80mg/m cube est dépassé ", round(1-ecdf(PM10Moyen)(80),
digits=3), "% du temps sur les",j ,"jours observés")
```

Le seuil de **50mg/m** cube est dépassé ***0.022%*** du temps sur les 730 jours observés
Le seuil de **80mg/m** cube est dépassé ***0.01%*** du temps sur les 730 jours observés

---

QUESTION 6 :

```
PM10SeuilInfo = filter(PM10Moyen, PM10Moyen > 50)
PM10SeuilAlerte = filter(PM10Moyen, PM10Moyen > 80)
```

PM10SeuilInfo : Les jours où le PM10Moyen dépasse le seuil d'information :

PM10Moyen	DATE
50.167	19/01/2017
86.333	20/01/2017
103.167	21/01/2017
104.500	22/01/2017
107.000	23/01/2017
89.500	24/01/2017
57.000	25/01/2017
96.500	26/01/2017
77.000	27/01/2017
83.667	28/01/2017
54.667	29/01/2017
55.167	14/02/2017
51.833	16/02/2017
53.400	16/03/2017
51.550	03/11/2017
61.400	21/11/2017

PM10SeuilAlerte : Les jours où le PM10Moyen dépasse le seuil d'alerte :

PM10Moyen	DATE
86.333	20/01/2017
103.167	21/01/2017
104.500	22/01/2017
107.000	23/01/2017
89.500	24/01/2017
96.500	26/01/2017
83.667	28/01/2017

```
OzoneSeuilInfo = filter(OzoneMoyen, OzoneMoyen > 180)
OzoneSeuilAlerte = filter(OzoneMoyen, OzoneMoyen > 240)
```

OzoneSeuilInfo et ***OzoneSeuilAlerte*** sont vides. Il n'y a aucun jour où l'ozone dépasse le seuil d'information ni le seuil d'alerte.

PARTIE B

QUESTION 1 :

Calculs des relevés de *dfg :*

	Dioxyde d'azote	Monoxyde d'azote	Ozone	Particules PM10
Dioxyde d'azote				
Covariance	193.784	187,866	-206,200	106,954
Corrélation	1,000	0.692	-0.637	0.604
Monoxyde d'azote				
Covariance	187,866	380.814	-250.854	167.278
Corrélation	0.692	1,000	-0.553	0.674
Ozone				
Covariance	-206,200	-250.854	541.208	-102.401
Corrélation	-0.637	-0.553	1,000	-0.346
Particules PM10				
Covariance	106,954	167.278	-102.401	161.976
Corrélation	0.604	0.674	-0.346	1,000

Calculs des relevés de *dfg2 :*

	Ozone
Particules PM10	
Covariance	-112.113
Corrélation	-0.358

On se demande pourquoi nos corrélations de l'Ozone et des particules PM10 dans le relevé de *dfg* sont différentes de celles du relevé de *dfg2*.
Cela s'explique lors de la création de ces deux data-frame, plus précisément dû au fait qu'on ignore les jours de non-observations.

Quand on crée un nouveau data-frame, "*na.omit*" supprime la ligne entière, donc les données des colonnes adjacentes pour garantir la même longueur sur les différentes colonnes du data-frame.
Voilà pourquoi les corrélations diffèrent, *dfg2* à moins de données "tronquées".

QUESTION 2 :

Pour effectuer un test de Pearson, il faut supposer que les deux échantillons suivent une distribution de loi normale.
Soit *y* l'échantillon des mesures des Particules PM10 à la Station Lyon Gerland et *x* l'échantillon des mesures d'Ozone à Lyon Gerland.

```
x = dfg2$Lyon.Gerland.Ozone
y = dfg2$Lyon.Gerland.Particules.PM10
cor.test(x,y)
```

La fonction *cor.test* renvoie un test de corrélation de *x* et *y*.

```
Pearson's product-moment correlation

data:  x and y
t = -9.6681, df = 636, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.4238076 -0.2883508
sample estimates:
       cor
-0.3579611
```

On constate que le coefficient de corrélation est négatif (-0.358) et que la p-value du test est très proche de 0.
On peut donc en conclure que *x* et *y* sont négativement corrélés, c'est-à-dire qu'ils tendent à augmenter lorsque les valeurs de l'autre variable diminuent.
De plus, en vue de très la petite p-value on peut rejeter l'hypothèse nulle avec une grande certitude.

QUESTION 3 :

Dans cette question on cherche à créer la droite de régression linéaire et le nuage de points de *y* en fonction de *x*.
On commentera les résultats pour voir si, oui ou non, ces résultats sont significatifs.

```
lmll = lm(y~x)
summary(lmll)
```

summary(lmll) donne :

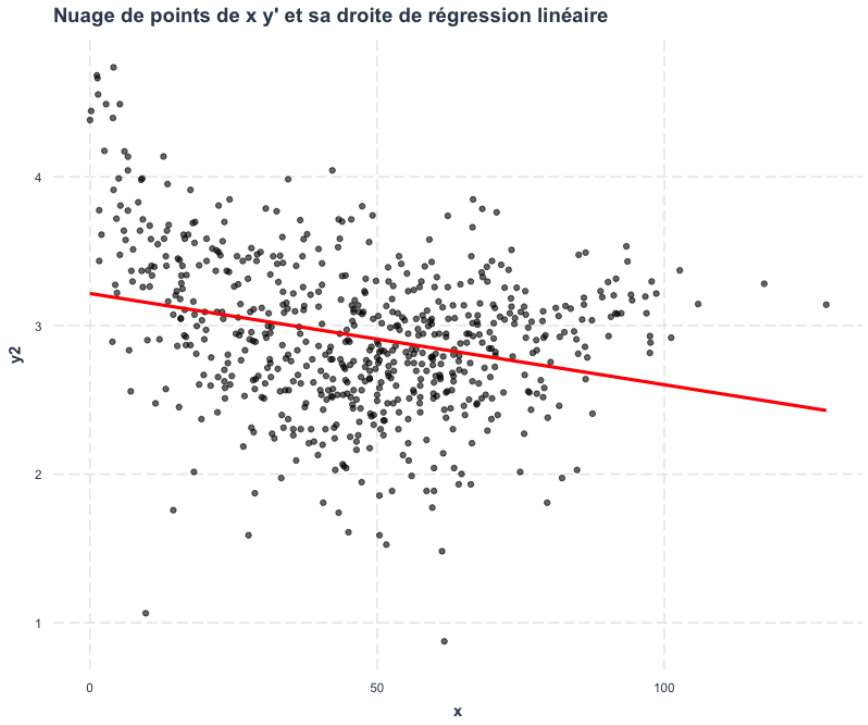
```
Residuals:
    Min      1Q  Median      3Q     Max
-26.252  -8.102  -2.555   5.858  83.698

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 31.14305    1.11036  28.048   <2e-16 ***
x           -0.20526    0.02123  -9.668   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.52 on 636 degrees of freedom
Multiple R-squared:  0.1281,  Adjusted R-squared:  0.1268
F-statistic: 93.47 on 1 and 636 DF,  p-value: < 2.2e-16
```

Cela nous renseigne sur pleins de choses, dont le “Residual standard error” qui en l'occurrence, est élevé. Il ne vaut mieux pas utiliser cette représentation linéaire.
De plus le “Multiplied R-squared” de 12.810% nous conforte dans le fait que cette représentation n’est pas très significative.

Ci-dessous, l'affichage d'un nuage de point de *x* et *y* avec la droite de régression linéaire.



QUESTION 4 :

A l'aide de la fonction gofTest de le librairy "EnvStats" nous devons réaliser un test du $\chi 2$ de normalité pour x, y et y' = ln(y).
Cela nous permettra de conclure sur la normalité des échantillons.

```
gofTest(y, test = "chisq", distribution = "norm")

gofTest(x, test = "chisq", distribution = "norm")

y2 = log(y)
gofTest(y2, test = "chisq", distribution = "norm")
```

Pour l'échantillon *y*, la p-value du test est égale à 0, on rejette donc l'hypothèse avec une grande sûreté que y suivent une loi normale.
Pour *x*, le p-value vaut 0.255, ici on ne peut pas rejeter l'hypothèse que l'échantillon *x* suive une loi normale. Donc *x* suit une loi normale.

En ce qui concerne *y2* / *y'* le p-value du test est égale à 0.498 on peut en conclure que *y2* suit une loi log normal.

QUESTION 5 :

Dans cette question on cherche à créer la droite de régression linéaire et le nuage de points de *y'* en fonction de *x*.
On commentera les résultats pour voir si oui ou non ces résultats sont significatifs.

```
lmll2 = lm(y2~x)
summary(lmll2)
```

summary(lmll2) donne :

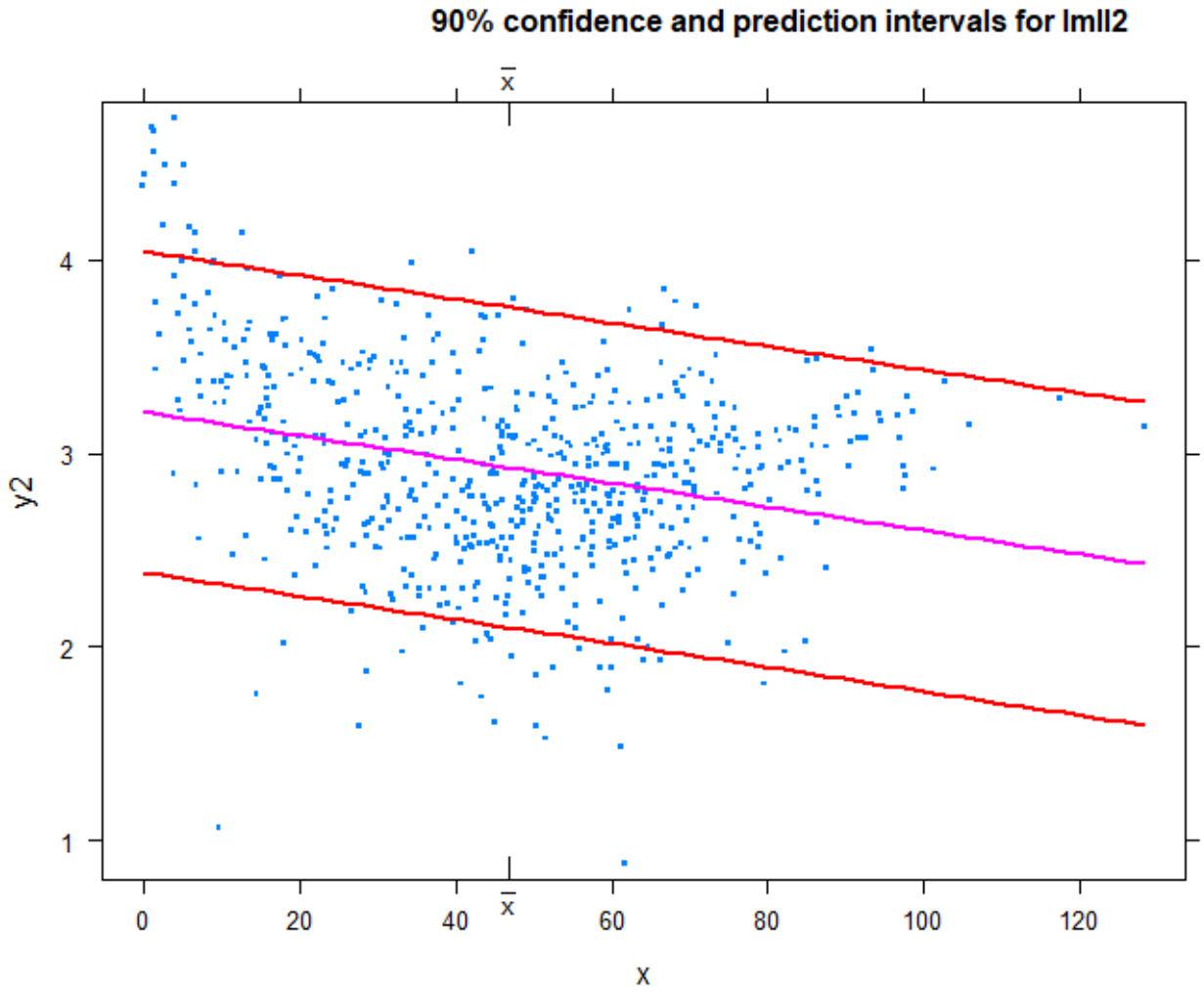
```
Residuals:
     Min       1Q   Median       3Q      Max
-2.09093 -0.31510 -0.00123  0.34084  1.54620

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.2151638  0.0445148   72.23  < 2e-16 ***
x           -0.0061367  0.0008511   -7.21 1.59e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5021 on 636 degrees of freedom
Multiple R-squared:  0.07556, Adjusted R-squared:  0.07411
F-statistic: 51.98 on 1 and 636 DF,  p-value: 1.593e-12
```

On peut constater d'abord que le "Residual standard error" est faible (0.502) de plus le "Multiple R-squared" à un faible pourcentage (7.556%) on peut donc dire que ses résultats statistiques sont significatifs.

Ci-dessous, l'affichage d'un nuage de point de *x* et *y*' avec la droite de régression linéaire bornée par l'intervalle de prédiction au niveau 90% .



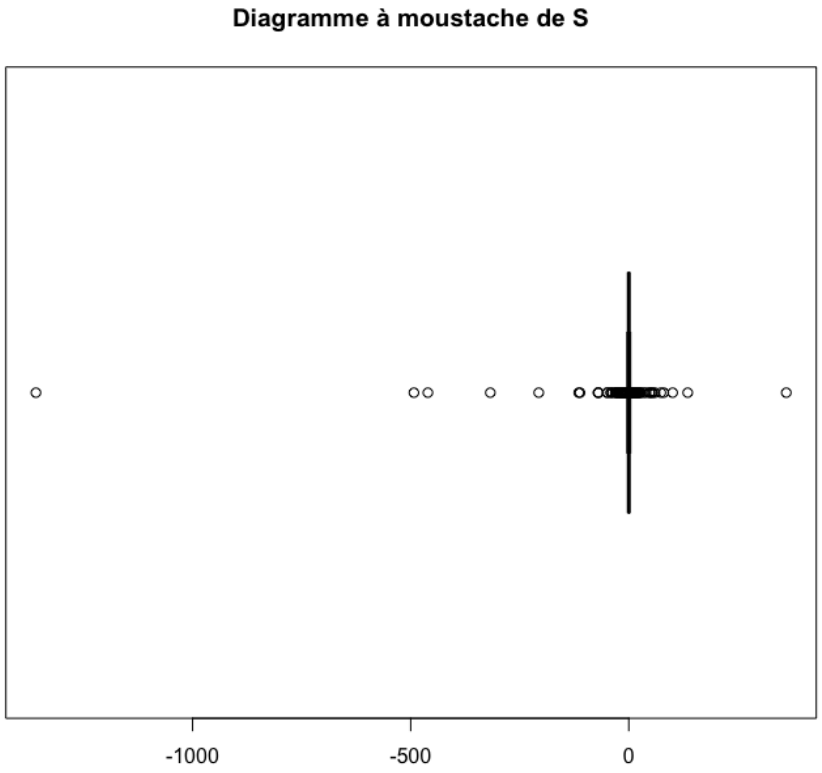
EXERCICE 2

QUESTION 1 :

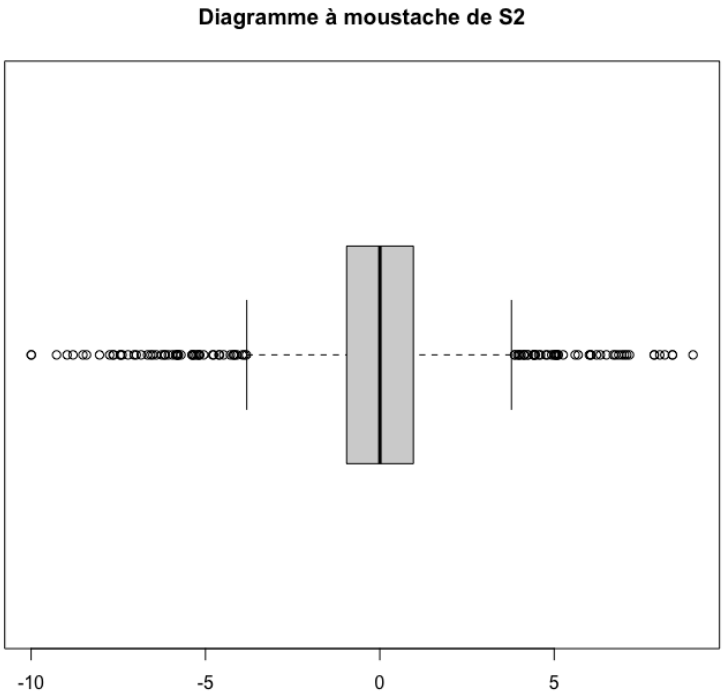
```
U<- runif(1000,min=0,max=1)
S <- tan(pi*U)
```

Exécution des commandes de l'énoncé

```
boxplot(S,horizontal=TRUE,qtype=1,main="Diagramme à moustache de S")
```

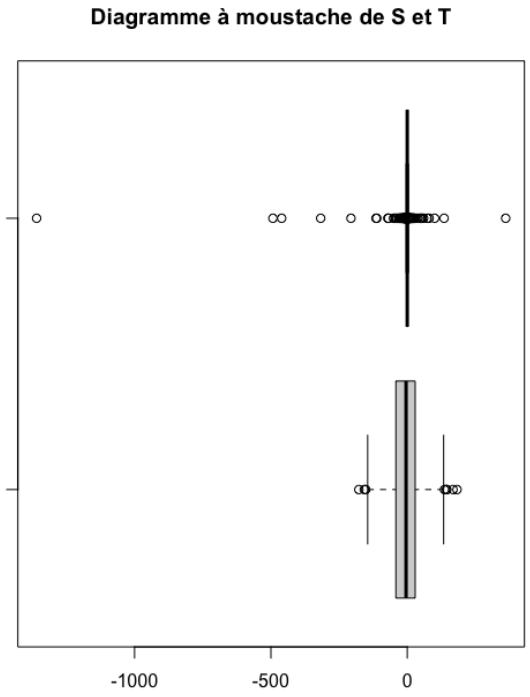


```
S2 = S[S<10 & S>-10] # rédution de l'intervalle
boxplot(S2,horizontal=TRUE,qtype=1,main="Diagramme à moustache de S2")
```

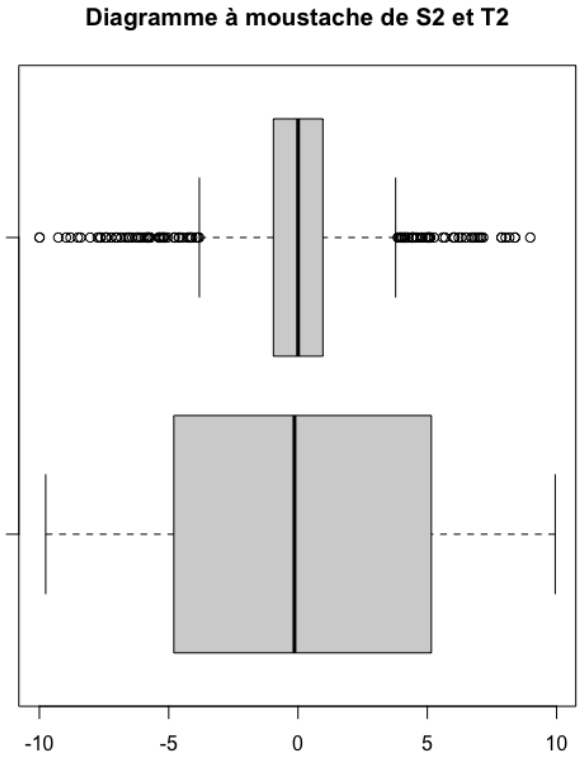


QUESTION 2 :

```
MS = mean(S)
ETS = sqrt(var(S))
T = rnorm(1000, mean = MS, sd = ETS) # loi normale de paramètre (MS, ETS)
boxplot(T,S,horizontal=TRUE,qtype=1,main="Diagramme à moustache de S et T")
# compare T & S
```




```
T2 = T[T<10 & T>-10] # sous échantillon de T
boxplot(T2,S2,horizontal=TRUE,qtype=1,main="Diagramme à moustache de S2 et T2") # compare T2 & S2 sur le même échantillon (-10,10).
```



On peut tout d'abord constater que les deux médianes sont toutes les deux proches de 0. Les moustaches de T2 sont presque au même niveau que les valeurs extrêmes les plus élevées de S2. Les moustaches de S2 sont presque au même niveau que les quartiles de T2.

S semble être un échantillon de loi normale.

QUESTION 3 :

```
gofTest(S, test = "chisq", distribution = "norm")
```

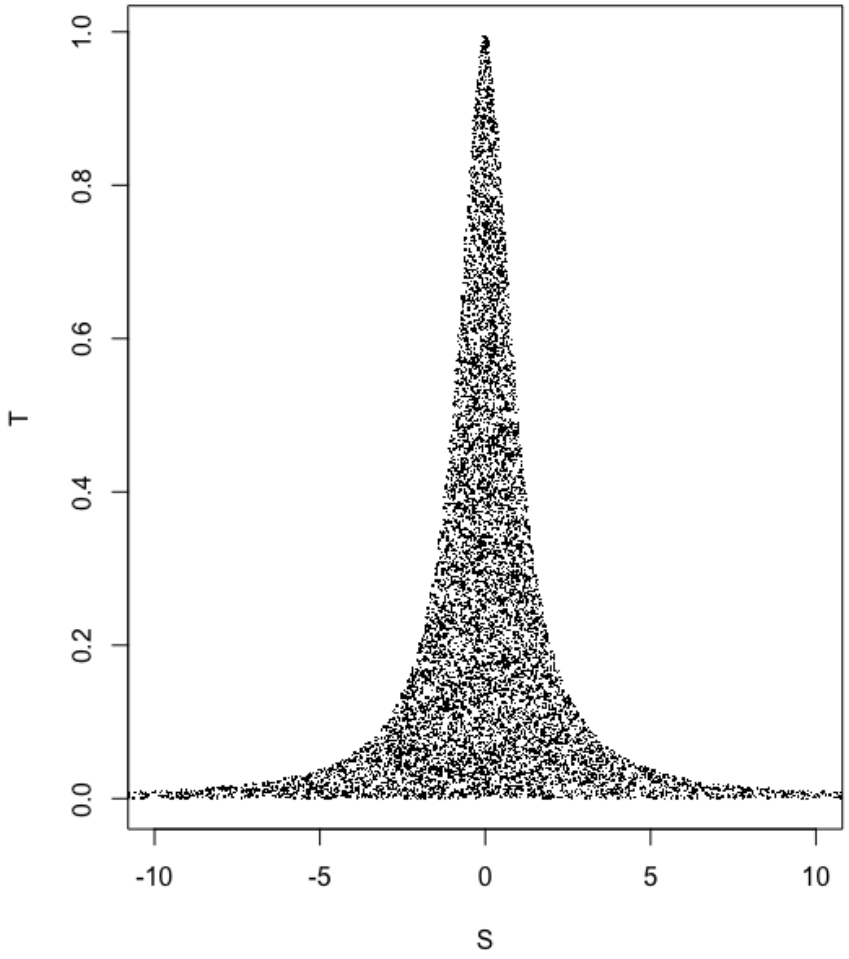
La p-valeur du test est 0. L'hypothèse H0 étant : *'S suit une loi normale'* et qu'elle n'est pas vérifiée, on peut dire avec une grande certitude que S ne suit pas une loi normale.

QUESTION 4 :

```
U = runif(10000 ,min=0,max=1)
V = runif(10000 ,min=0,max=1)
S = tan(pi*U)
T = V/(1+S^2)
```

On exécute les commandes de l'énoncé.

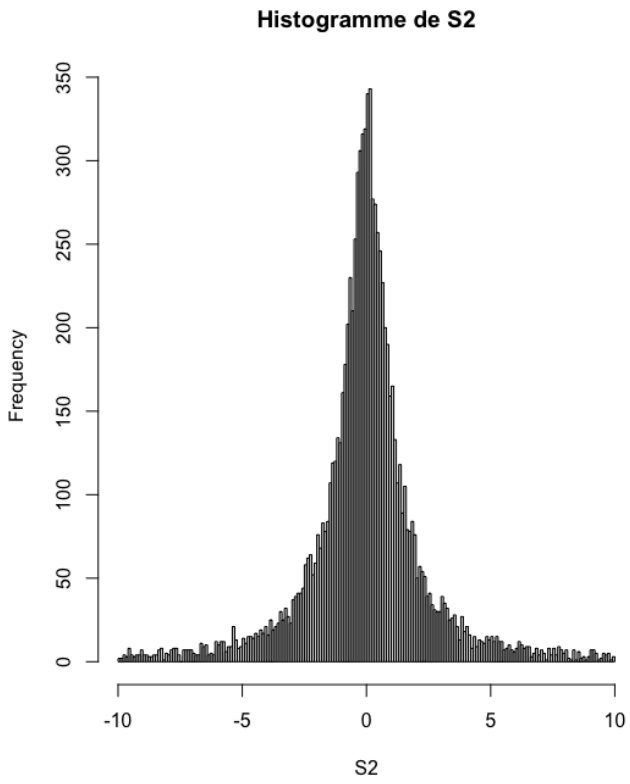
```
plot(S,T, pch=".", xlim=c(-10,10))
```



(S, T) semble uniformément distribué sur **[-10,10]** et **[0,1]**

QUESTION 5 :

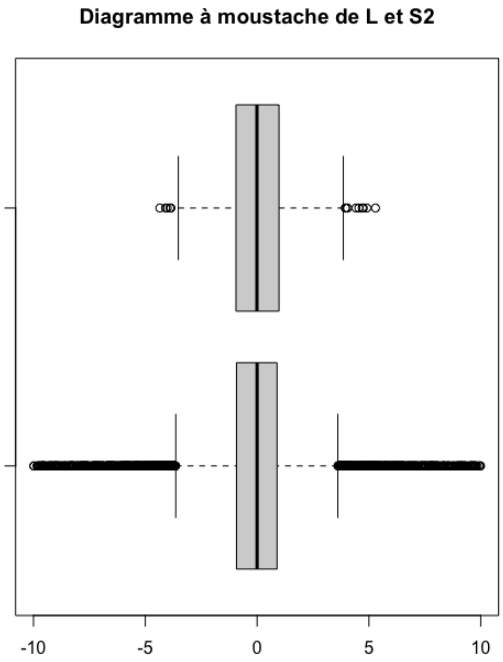
```
S2 = S[S<10 & S>-10] # Réduction de S à [-10, 10]
hist(S2,breaks=200, main = "Histogramme de S2")
```



Visuellement, S2 semble suivre une loi normale.

```
L = rnorm(1000, 0, 1.5)
boxplot(S2,L,horizontal=TRUE,qtype=1,main="Diagramme à moustache de L et
S2",ylim=c(-10,10))
```

Après quelques ajustements, on trouve L qui suit une loi normale (0, 1,5) ayant un diagramme à moustache très similaire à celui S2, hormis pour les valeurs extrêmes. On peut donc supposer que S2 suit la même loi que L avec à peu près les mêmes paramètres




```
gofTest(S2, test = "chisq", distribution = "norm")
```

La p-valeur du test est 0. L'hypothèse H0 étant : *'S2 suit une loi normale'* et qu'elle n'est pas vérifiée, on peut dire avec certitude que S2 ne suit pas une loi normale selon ce test.