Tutoriel_R

February 27, 2017

Luc Adjengue, Farooq Sanni et Wissem Maazoun

Table des matières

1 Ouverture des fichiers et importation des données **1**
- 1.1 Importation des fichiers à partir du site Moodle . . . 2
- 1.2 Ouverture des fichiers ".ipynb" . . . 2
- 1.3 Importation d'un fichier CSV . . . 2

2 Les graphiques dans R **4**
- 2.1 Les histogrammes . . . 4
- 2.2 Diagramme de Tukey . . . 7
- 2.3 Diagramme de probabilité normale . . . 10
- 2.4 Diagramme de dispersion ou nuage de points . . . 12

3 Description numérique et manipulation des "data.frame" dans R **14**

4 Simulation de données **16**

5 Calcul de probabilités **17**

6 Analyse de régression **18**

1 Ouverture des fichiers et importation des données

- Le but de ce tutoriel est de vous familiariser avec le fonctionnement du logiciel R.
- Le logiciel R et son interface convivial RStudio sont installés dans les laboratoires informatiques.
- Le fichier des données à utiliser dans ce tutoriel est disponible sur le site du cours sous format CSV, Notes.csv.
- Vous pouvez exécuter une à une les commandes du fichier "Tutoriel_R.pdf" directement sur la console de R, ou sous forme de script dans RStudio (fichier "Tutoriel_R.r"); vous pouvez également utiliser un notebook de jupyter (fichier "Tutoriel_R.ipynb") disponible dans l'environnement Anaconda Navogator. C'est cette dernière option qui est présentée dans ce qui suit.

1.1 Importation des fichiers à partir du site Moodle

Vous devez télécharger les fichiers “Tutoriel_R.ipynb”, “Tutoriel_R.r” et “Notes.csv”, et les mettre tous dans un même répertoire.

1.2 Ouverture des fichiers “.ipynb”

Pour utiliser ces fichiers, vous devez utiliser “jupyter” qui est diponible soit:

- dans anaconda_navigator, qui est disponible dans les programmes intallés dans les machines des laboratoires. Dans le navigateur anaconda, vous sélctionner jupyter. Une fenêtre de navigateur internet s’ouvre;
- ou bien à partir de “invite commande” de windows, vous tapez la commande

```
jupyter notebook
```

et une fenêtre de navigateur internet s’ouvre.

Dans la fenêtre ouverte vous cliquez sur les répertoires pour atteindre le dossier où vous avez télćhargé vos fichier de Moodle. Vous ouvrez le fichier ““Tutoriel_R.ipynb“.

Le fichier “Tutoriel_R.r” est un fichier script qui contient l’ensemble des commandes. Ce fichier peut-être ouvert et exécuté avec “R” ou “RStudio”.

1.3 Importation d’un fichier CSV

On peut importer des fichiers de différents formats dans R. À titre d’exemple, importer dans R le fichier CSV intitulé Notes.csv. On utilise la fonction read.csv(“NomDuFichier”, header = TRUE). Vous devez aussi spécifier si le système que vous utiliser prend “.” ou “,” pour décimale. Dans l’exemple qui suit, le fichier est lu et les données sont placées dans une base de données appelée ici “Notes”.

```
In [1]: Notes <- read.csv("Notes.csv", header = TRUE, sep = ";",dec = ".")
```

On peut afficher les données en tapant le nom du data frame qu’on a attribué (ici, on a utilisé “Notes”)

```
In [2]: Notes
```

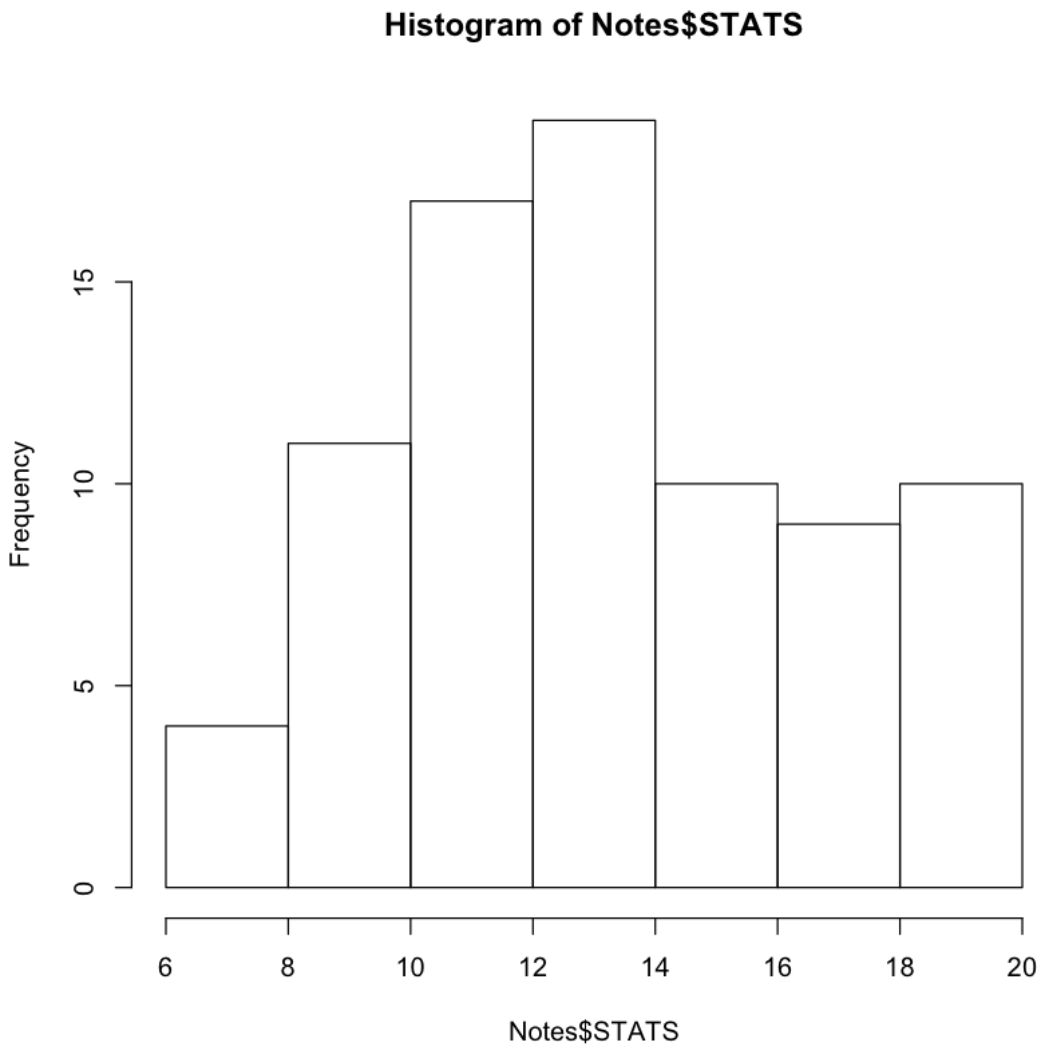
IDENT	ECONO	STATS	GROUP	SEXE
1	12.0	9.0	A	M
2	15.8	12.7	B	F
3	12.1	12.9	A	M
4	12.3	11.6	A	M
5	12.2	11.6	B	F
6	12.7	10.7	B	F
7	14.6	18.2	B	M
8	9.4	12.6	B	M
9	10.5	8.6	B	M
10	12.5	11.0	B	M
11	12.9	15.0	B	M
12	9.7	10.6	B	F
13	13.4	16.1	B	M
14	11.4	10.7	B	M
15	14.5	11.9	B	F
16	18.8	14.9	A	M
17	11.9	12.2	B	F
18	16.6	20.0	A	M
19	13.8	12.7	B	M
20	12.5	9.8	A	F
21	7.3	10.1	B	F
22	9.6	11.1	B	M
23	13.9	13.6	B	M
24	7.8	7.5	B	F
25	11.2	13.1	B	F
26	14.4	19.6	B	M
27	15.2	20.0	A	M
28	9.8	10.7	A	M
29	10.9	11.4	B	M
30	16.9	20.0	A	F
⋮	⋮	⋮	⋮	⋮
51	10.1	8.9	A	M
52	18.9	20.0	A	F
53	12.9	16.4	A	F
54	10.8	8.8	A	M
55	10.8	14.8	B	F
56	11.9	11.4	B	F
57	13.7	18.3	A	F
58	12.4	17.5	B	M
59	13.0	15.8	B	M
60	14.0	11.7	B	M
61	14.8	13.6	A	F
62	8.8	11.1	B	F
63	10.4	8.6	B	F
64	17.5	14.7	B	F
65	14.0	12.8	B	F
66	11.8	14.3	A	M
67	11.5	13.0	A	F
68	6.5	9.0	B	M
69	12.7	17.4	A	M
70	12.4	9.9	B	M
71	12.4	14.4	A	F

2 Les graphiques dans R

2.1 Les histogrammes

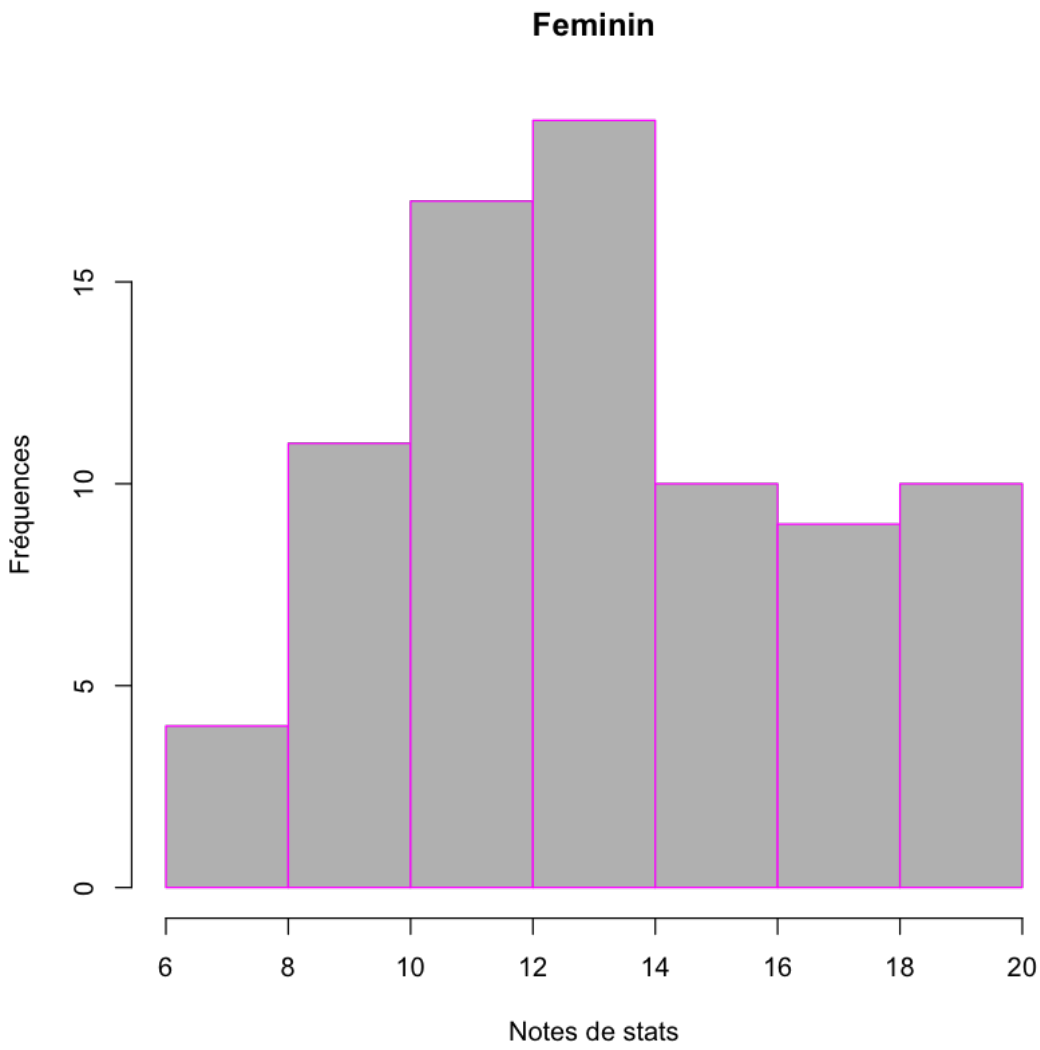
Pour faire un histogramme pour la variable STAT, on peut utiliser la commande suivante

```
In [3]: hist(Notes$STATS)
```



On peut personnaliser l'histogramme en ajoutant col = " " pour couleur border = " " pour la couleur de la bordure main = paste(" ") pour donner un titre au graphique xlab =”" et ylab = “” pour donner des noms aux axes

```
In [4]: hist(Notes$STATS, col="grey",border="magenta", main=paste("Feminin"),
            xlab="Notes de stats",ylab="Fréquences")
```

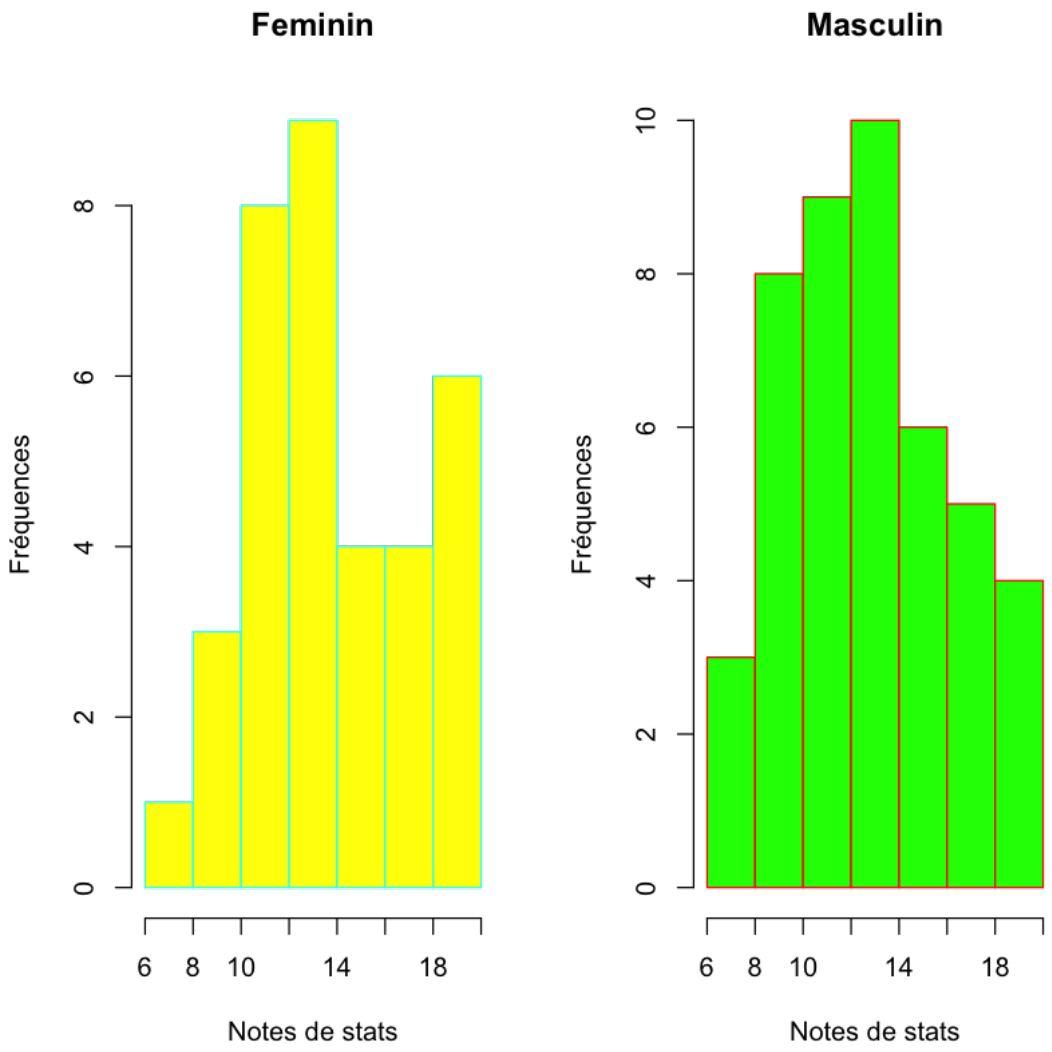


Faites un histogramme pour la variable ECONO

In [ ]:

On peut constuire deux histogrammes juxtaposés pour la variable STAT en fonction du sexe

```
In [5]: layout(matrix(1:2,1,2)) # permet de diviser la sortie graphique en deux
        hist(Notes$STATS[Notes$SEXE=="F"], col="yellow",border="cyan",
             main=paste("Feminin"),xlab="Notes de stats",ylab="Fréquences")
        hist(Notes$STATS[Notes$SEXE=="M"], col="green",border="red",
             main=paste("Masculin"),xlab="Notes de stats",ylab="Fréquences")
```



Faites un histogramme juxtaposé pour la note de stat en fonction du sexe

In [ ]:

Faites un histogramme juxtaposé pour la note de stat en fonction du groupe

In [ ]:

Faites un histogramme juxtaposé pour la note de econo en fonction du sexe

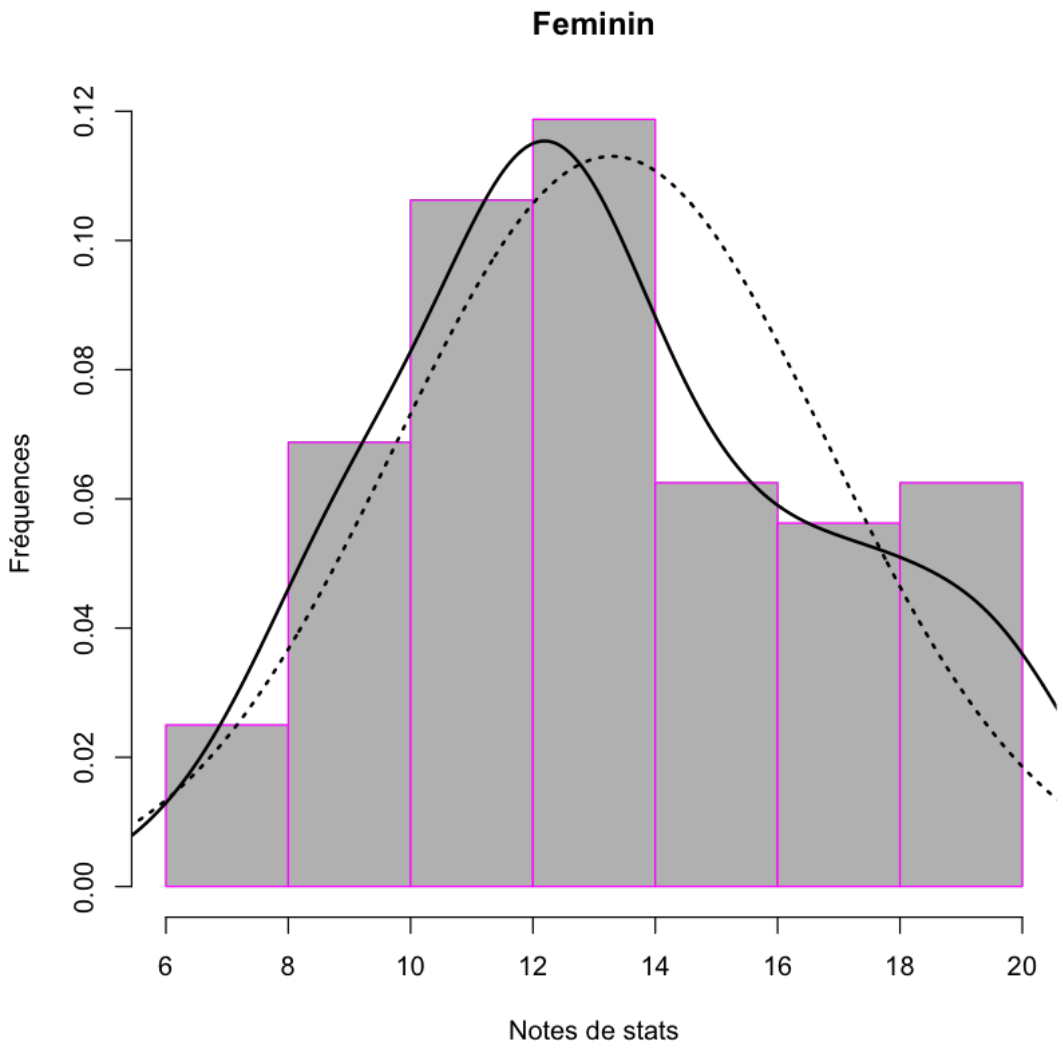
In [ ]:

Faites un histogramme juxtaposé pour la note de econo en fonction du groupe

In [ ]:

On peut rajouter à l'histogramme la courbe de densité d'une loi normale et celle estimée pour les données. Dans l'appel à la fonction hist(), il faut mettre le paramètre freq à FALSE (freq=FALSE)

```
In [6]: hist(Notes$STATS, col="grey",border="magenta", main=paste("Feminin"),
             xlab="Notes de stats",ylab="Fréquences", freq=FALSE)
        lines(density(Notes$STATS), lwd=2) # densité ajustée
        x = seq(5,21,length.out=500)
        lines(x, dnorm(x, mean(Notes$STATS), sd(Notes$STATS)), lwd=2, lty=3)
        #densité d'une loi normale
```

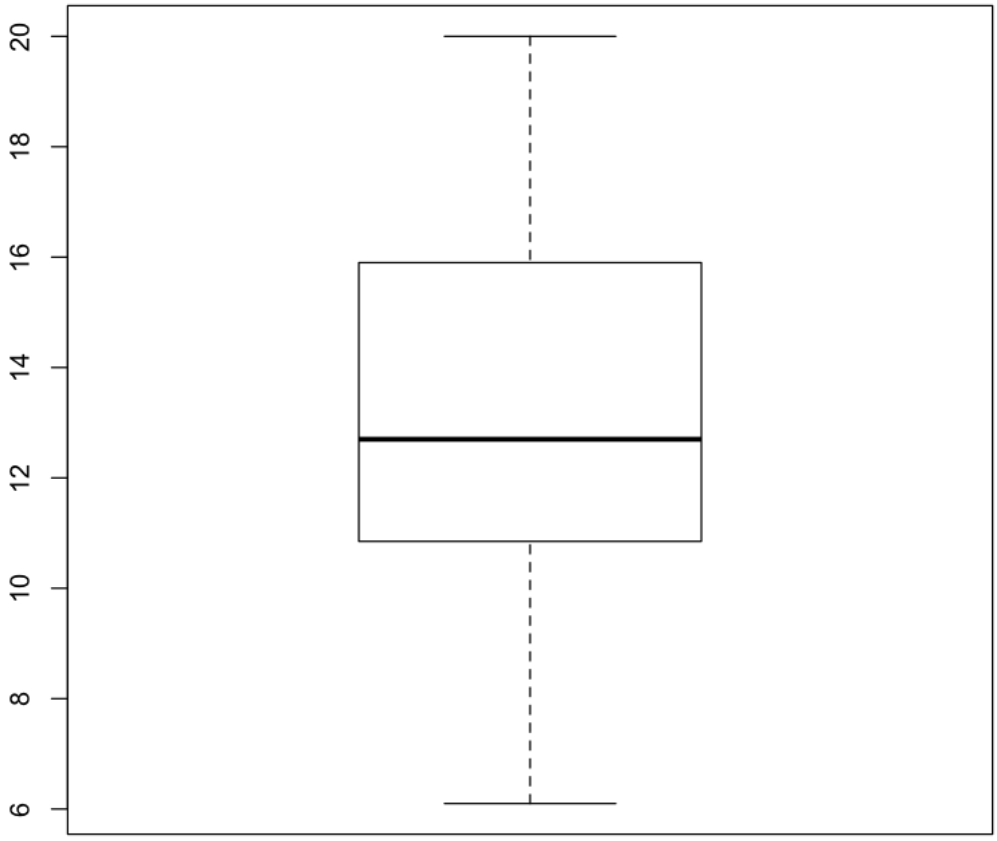


2.2 Diagramme de Tukey

La commande pour tracer un diagramme de Tukey est boxplot().

Pour tracer le diagramme de Tukey des notes de stats on fait :

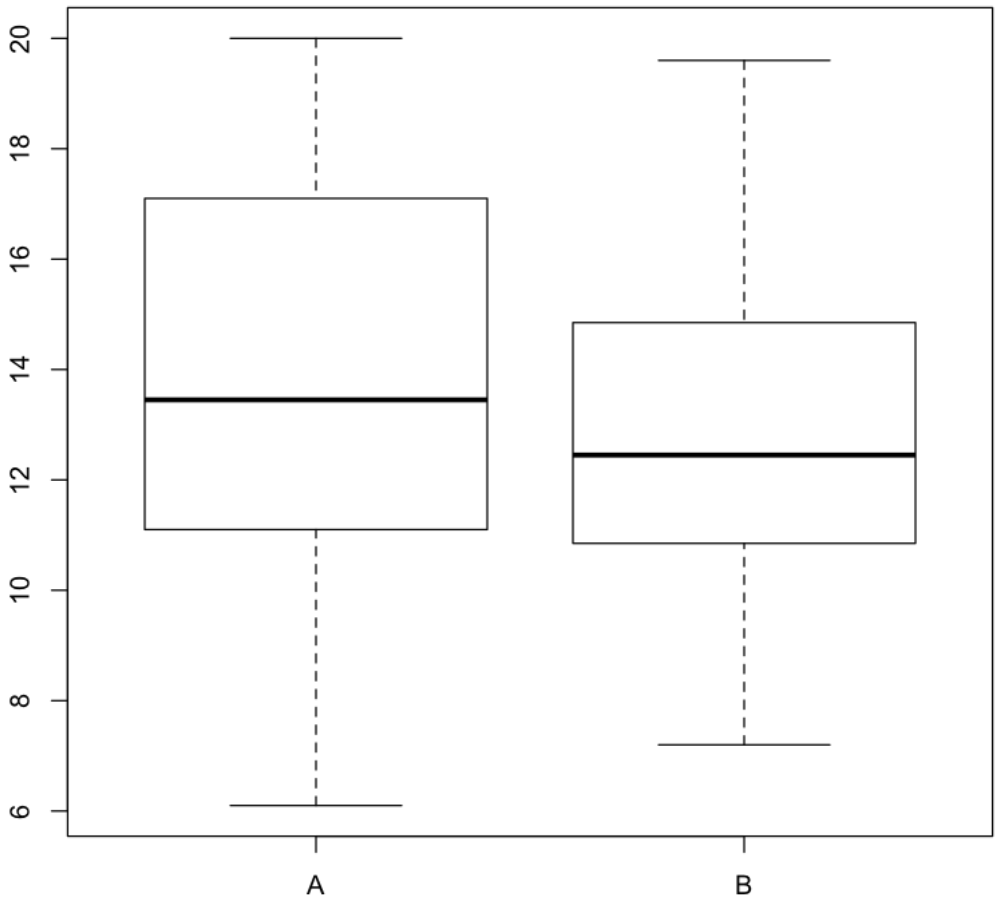
```
In [7]: boxplot(Notes$STATS)
```



Il est souvent plus simple d'invoquer directement les variables par leurs noms dans les commandes si on précise le nom de la base de données (voir l'exemple ci-dessous).

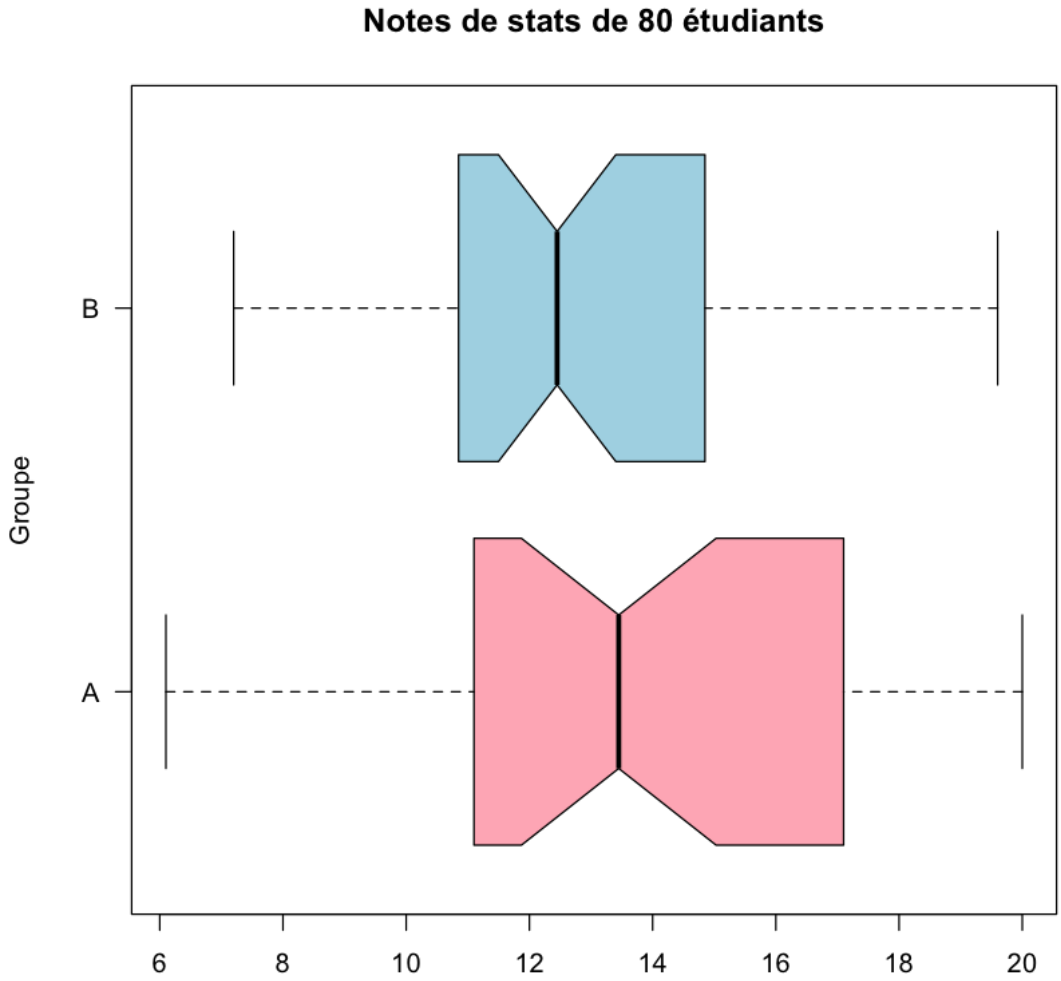
On peut obtenir deux diagrammes de Tukey des notes de stats pour chaque groupe comme suit :

```
In [8]: boxplot(STATS~GROUP, data=Notes)
```



Vous pouvez personnaliser vos diagrammes de Tukey comme suit:

```
In [9]: boxplot(Notes$STATS~Notes$GROUP,
        col=c("lightpink","lightblue"),
        horizontal=TRUE,
        notch=TRUE,
        main=paste("Notes de stats de",nrow(Notes),"étudiants"),
        ylab="Groupe",
        las=1)
```



Tracer le diagramme de Tukey des notes d'économie

```
In [ ]:
```

Tracer les diagrammes de Tukey des notes d'économie pour chaque groupe

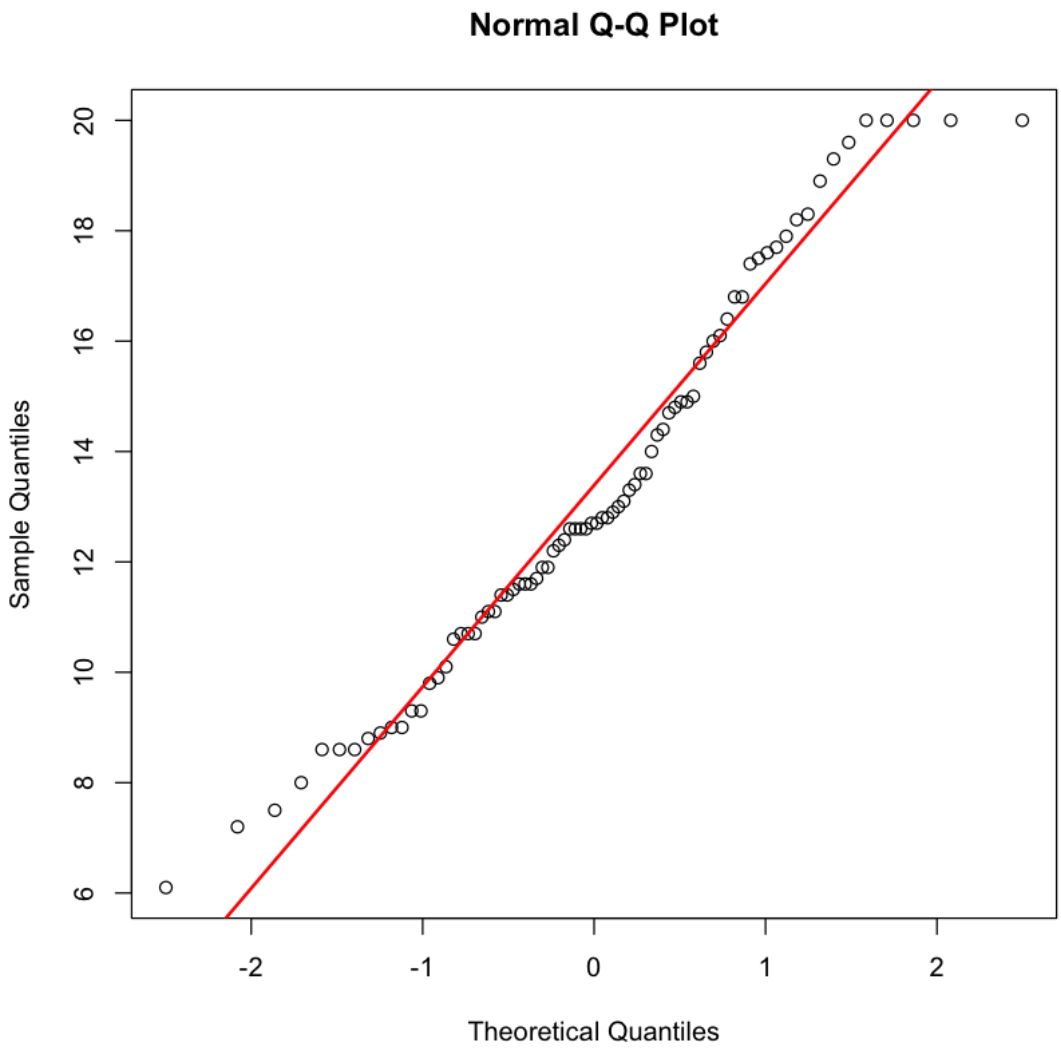
```
In [ ]:
```

2.3 Diagramme de probabilité normale

Les commandes ci-contre permettent d'obtenir le diagramme de probabilité normale.

Pour les notes d'économie on a :

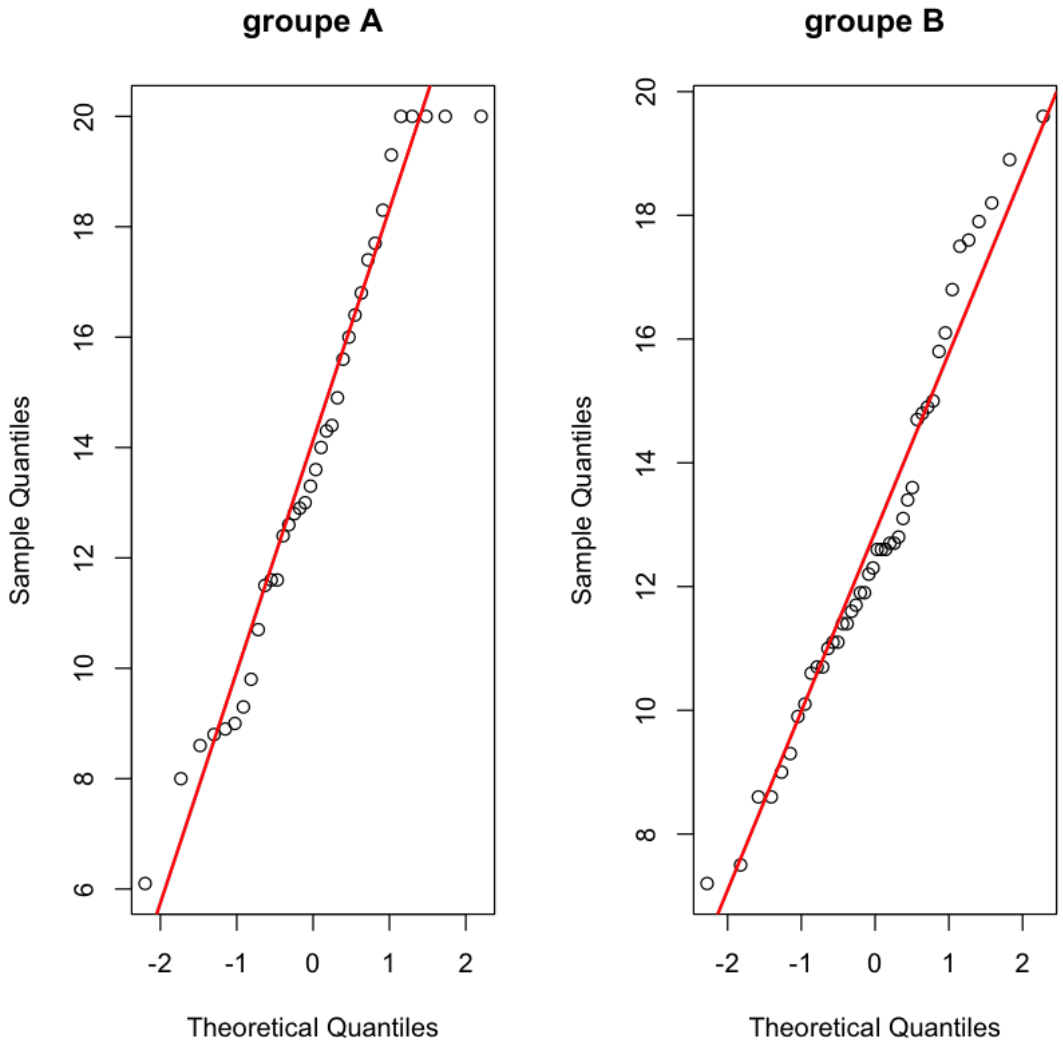
```
In [10]: qqnorm(Notes$STATS)
         qqline(Notes$STATS, col="red", lwd=2)
```



Pour tracer le diagramme de probabilité normale des notes d'économie en fonction du groupe, on exécute les commandes suivantes :

```
In [11]: layout(matrix(1:2,1,2)) #divise la sortie graphique en deux
         # groupe A
         qqnorm(Notes$STATS[Notes$GROUP=="A"], main="groupe A")
         qqline(Notes$STATS[Notes$GROUP=="A"], col="red", lwd=2)

         # groupe B
         qqnorm(Notes$STATS[Notes$GROUP=="B"], main="groupe B")
         qqline(Notes$STATS[Notes$GROUP=="B"], col="red", lwd=2)
```



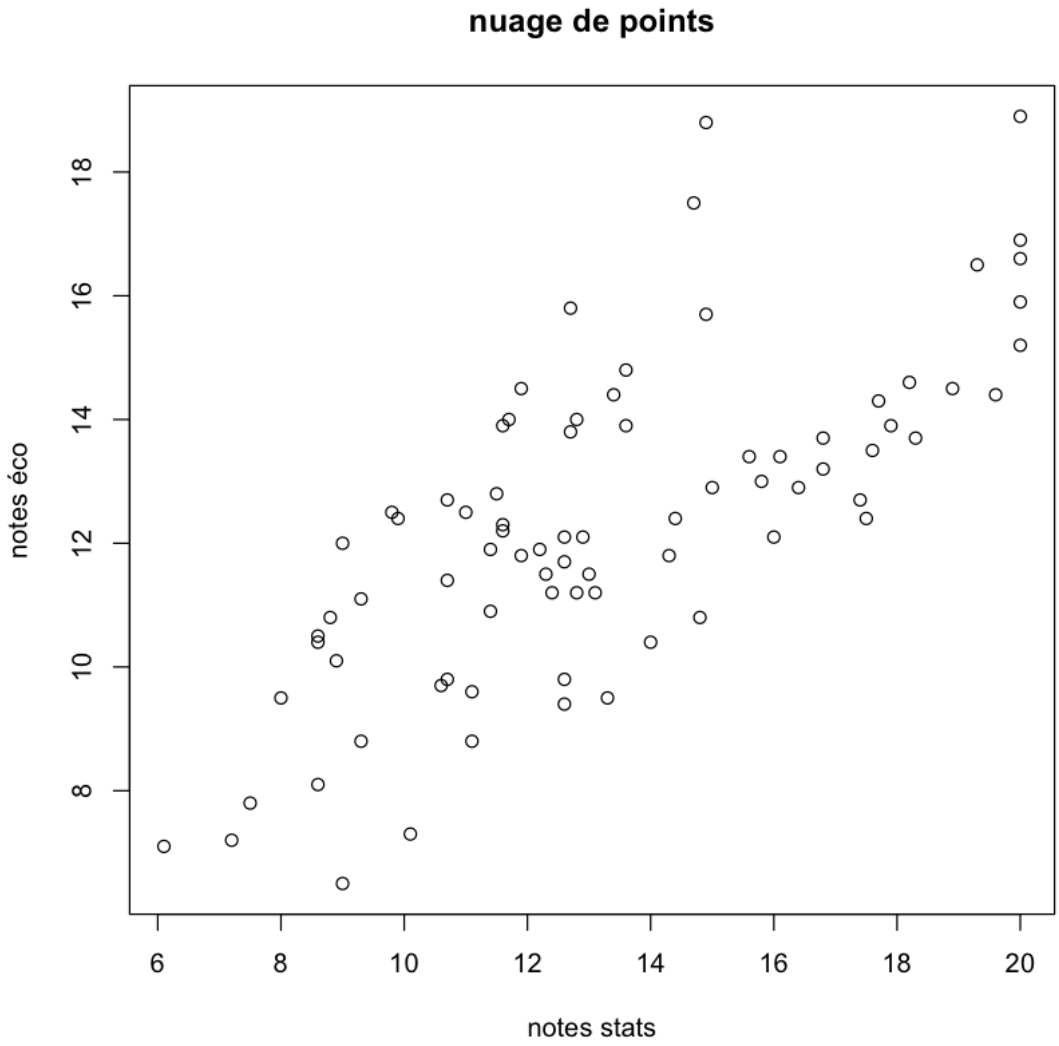
Obtenir un graphique similaire pour les notes d'économie en fonction du genre

2.4 Diagramme de dispersion ou nuage de points

Un nuage de points s'obtient à l'aide de la fonction plot().

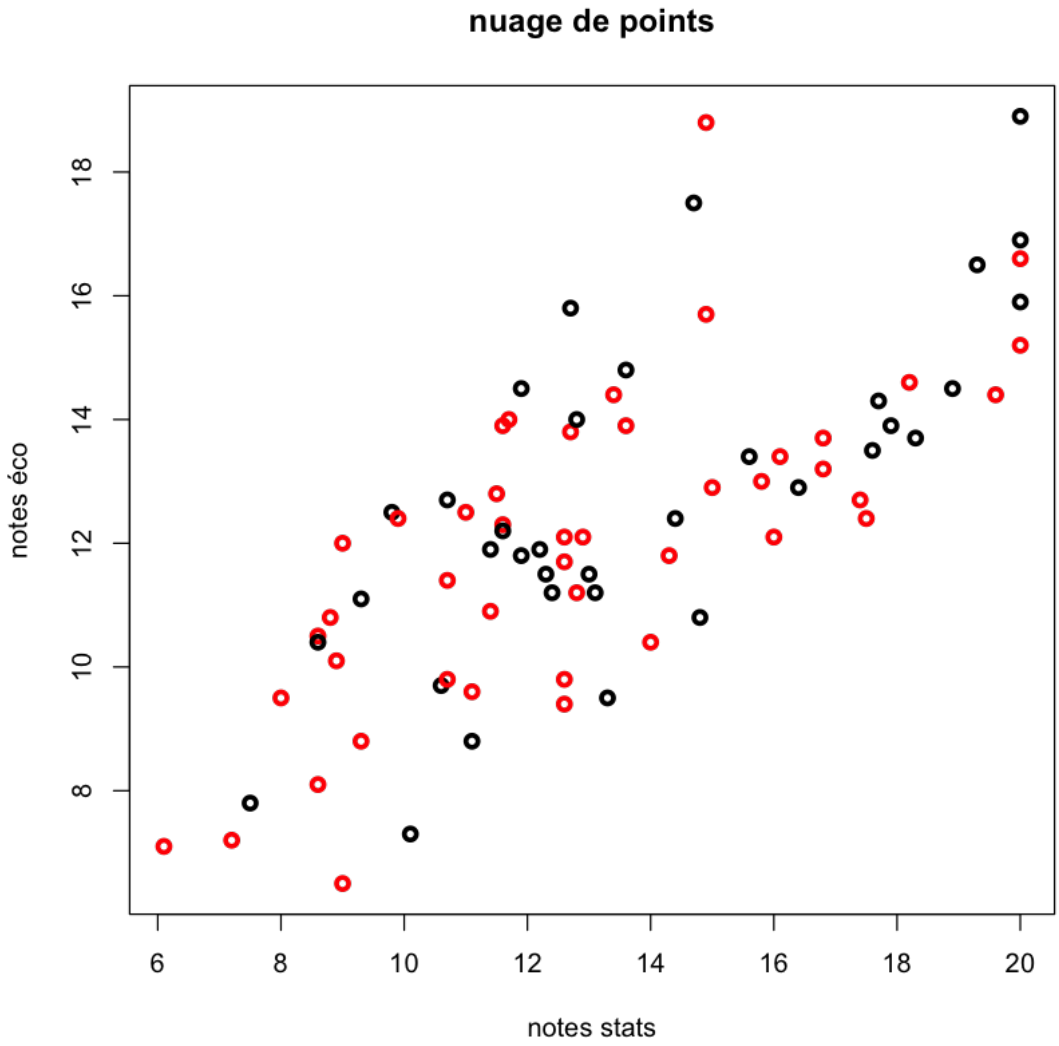
La commande suivante permet d'obtenir le nuage de points des notes d'économie en fonction de celles de statistiques

```
In [12]: plot(Notes$STATS, Notes$ECONO, xlab = "notes stats",
              ylab = "notes éco", main="nuage de points")
```



On peut personnaliser le nuage de points en modifiant par exemple la grosseur des points ou en modifiant leur couleur selon une catégorie.

```
In [13]: plot(Notes$STATS, Notes$ECONO, xlab = "notes stats",
              ylab = "notes éco", main="nuage de points",lwd=3, col=Notes$SEXE)
```



Tracer les notes de statistiques en fonction de celles d'économie

```
In [ ]:
```

La commande cor() permet de calculer le coefficient de corrélation entre deux vecteurs de données. Ainsi nous obtenons la correlation entre les notes d'économie et de statistiques :

```
In [14]: cor(Notes$STATS, Notes$ECONO)
```

0.730563270676972

3 Description numérique et manipulation des "data.frame" dans R

Nous allons obtenir quelques mesures sur nos deux variables écono et stats

Nous allons d'abord créer un tableau (data.frame) pour stocker nos résultats. Il s'agit uniquement d'une question d'esthétique.

```
In [15]: mesures = data.frame(notes=c("ECONO", "STATS"),
                              moyenne=NA, s=NA, q1=NA, mediane=NA,
                              q3=NA, r=NA, iqr=NA, min=NA, max=NA,
                              kurt=NA, skew=NA)
         mesures
```

notes	moyenne	s	q1	mediane	q3	r	iqr	min	max	kurt	skew
ECONO	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
STATS	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Remplissons notre tableau maintenant

```
In [16]: #moyenne
         mesures$moyenne = sapply(2:3, function(i) mean(Notes[,i]))
         mesures
```

notes	moyenne	s	q1	mediane	q3	r	iqr	min	max	kurt	skew
ECONO	12.33375	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
STATS	13.29125	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

```
In [17]: # écart-type
         mesures$s = sapply(2:3, function(i) sd(Notes[,i]))
         # mediane
         mesures$mediane = sapply(2:3, function(i) sd(Notes[,i]))
         # min
         mesures$min = sapply(2:3, function(i) min(Notes[,i]))
         # max
         mesures$max = sapply(2:3, function(i) max(Notes[,i]))
         # quantiles q1 et q3
         mesures[1, c("q1", "q3")] = quantile(Notes$ECONO, probs = c(0.25,0.75))
         mesures[2, c("q1", "q3")] = quantile(Notes$STATS, probs = c(0.25,0.75))
         # étendue
         mesures$r = mesures$max - mesures$min
         mesures$iqr = mesures$q3 - mesures$q1
         options(digits=4) # Pour limiter le nombre de décimales et
         mesures
```

notes	moyenne	s	q1	mediane	q3	r	iqr	min	max	kurt	skew
ECONO	12.33	2.591	10.80	2.591	13.90	12.4	3.100	6.5	18.9	NA	NA
STATS	13.29	3.530	10.93	3.530	15.85	13.9	4.925	6.1	20.0	NA	NA

Pour les coefficients d'aplatissement et de symétrie, nous avons besoin de le package "moments".

Elle se charge à l'aide de la commande library(). Si le package n'est pas installé, on procède comme suit :

```
In [18]: #install.packages("moments")
```

```
In [19]: library(moments)
         # si la commande échoue, on installe alors le package
         #install.packages("moments") dans la ligne de commande qui précède
         # enlever la dièse pour l'exécution!
         # et on réexécute la commande library

In [20]: # asymétrie (skewness)
         mesures$skew = sapply(2:3, function(i) skewness(Notes[,i]))

         # aplatissment (kurtosis)
         mesures$kurt = sapply(2:3, function(i) kurtosis(Notes[,i])) - 3
         # On fait -3 pour être conforme à la formule du cours
         options(digits=4)
         mesures
```

notes	moyenne	s	q1	mediane	q3	r	iqr	min	max	kurt	skew
ECONO	12.33	2.591	10.80	2.591	13.90	12.4	3.100	6.5	18.9	0.09207	0.1003
STATS	13.29	3.530	10.93	3.530	15.85	13.9	4.925	6.1	20.0	-0.71186	0.3004

Obtenir les mêmes mesures pour l'une des deux notes en fonction du groupe ou du genre. Essayer si possible de présenter les résultats sous la forme d'un tableau

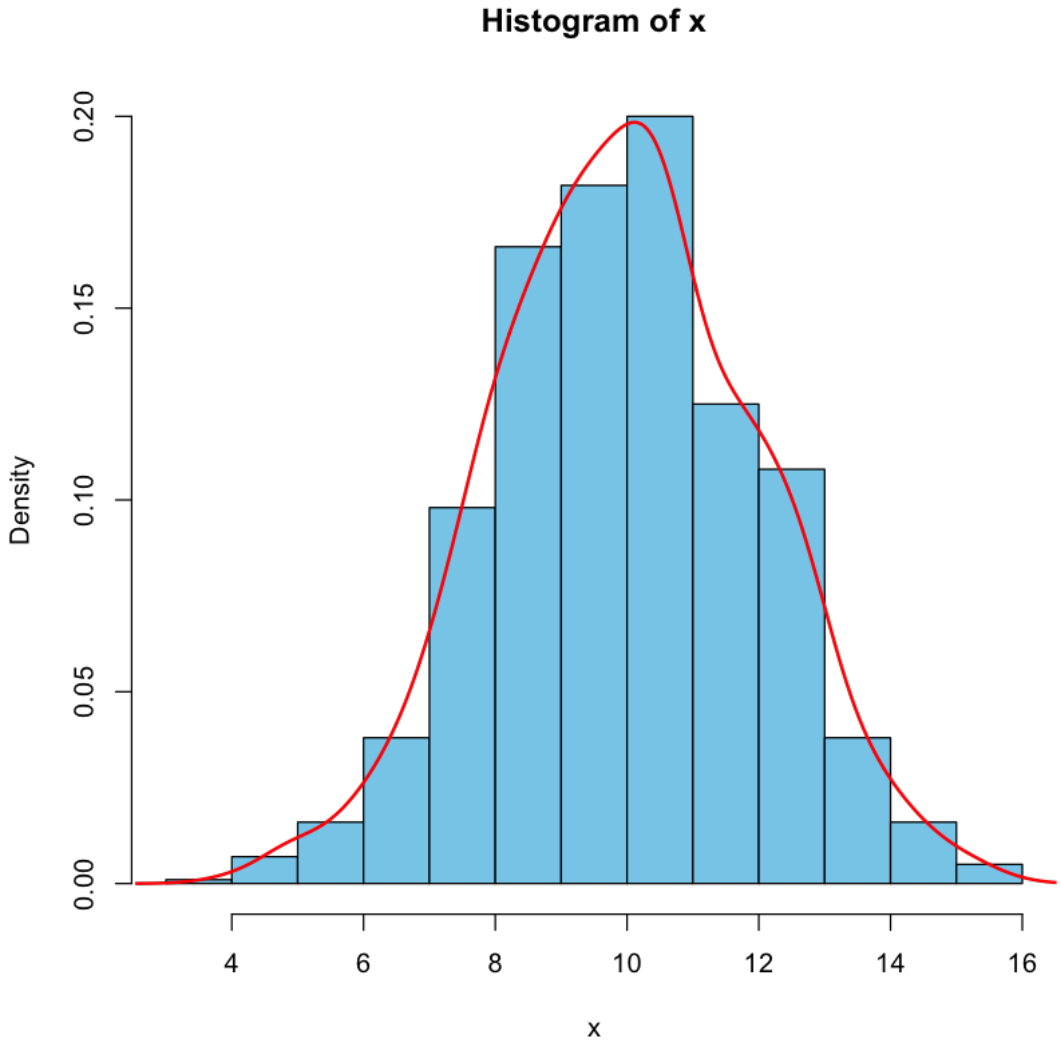
```
In [ ]:
```

4 Simulation de données

R nous permet de simuler des données selon les lois usuelles notamment celles que nous avons étudié en classe. Les commandes R se présentent par "r" suivi d'une troncature du nom de la loi. Ainsi les commandes rnorm(), rbinom() et rpois(), etc. permettent respectivement de simuler des données selon une loi normale, une loi de Poisson et une loi binomiale, etc.

Nous simulons 1000 observations d'une loi normale $\mathcal{N}(10, 4)$.Nous traçons ensuite l'histogramme des données auquel nous ajoutons la fonction de densité estimée.

```
In [21]: x = rnorm(n = 1000, mean = 10, sd = 2)
         hist(x, freq=F, col="skyblue")
         lines(density(x), col="red", lwd=2)
```



Tracer l'histogramme de 1000 observations simulées selon une loi exponentielle de paramètre $\lambda = 2$ et auquel vous ajouter la fonction de densité estimée.

```
In [ ]:
```

5 Calcul de probabilités

Nous pouvons calculer les probabilités ainsi que les quantiles à l'aide de R. Le préfixe "d" associé au nom tronqué d'une loi donne la fonction de masse dans le cas d'une loi discrète (par exemple dbinom() permet de calculer $p(x)$ pour la loi binomiale) et la fonction de densité pour une loi continue (dexp() pour la loi exponentielle).

De même avec "p", on a la fonction de répartition (ppois()) et "q", le quantile (qnorm()).

Pour une variable X de loi normale $\mathcal{N}(10, 9)$, calculons $P(X > 12{,}5)$

```
In [22]: 1- pnorm(12.5, 10, 3)
         # ou
         pnorm(12.5, 10, 3, lower.tail = F)
```

0.202328380963643
0.202328380963643
Trouvons $x_{0,05}$ qui est tel que $P(X > x_{0,05} = 0{,}05$

```
In [23]: # possibilité 1 : P(X>x_0.05) = 0.05 ==> P(X <= x_0.05) = 0.95
         qnorm(0.95, 10, 3)

         # possibilité 2 : lower.tail = F
         qnorm(0.05, 10, 3, lower.tail = F)
```

14.9345608808544
14.9345608808544
Déterminer p et x_p dans le cas où $X \sim \chi^2_{15}$ c'est à dire une loi de chi-deux à 15 degré de liberté. (indice : sous R chi-deux se tronque "chisq")

```
In [ ]:
```

6 Analyse de régression

On ajuste un modèle de régression avec la fonction "lm()". L'exemple qui suit présente l'ajustement d'un modèle de régression linéaire simple de la note de Stats en fonction de la note d'Écono :

```
Comme dans la plupart des procédures dans R, on donne un nom à l'objet créé pour po
```

Ici on a donné le nom "reg"

```
In [24]: reg<-lm(Notes$STATS~Notes$ECONO)
```

Pour obtenir le résumé de la régression, on utilise la commande "summary" comme suit

```
In [25]: summary(reg)
```

```
Call:
lm(formula = Notes$STATS ~ Notes$ECONO)

Residuals:
   Min     1Q Median     3Q    Max
 -4.83  -2.07  -0.07   2.15   4.25

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.018      1.327    0.77     0.45
```

```
Notes$ECONO    0.995      0.105    9.45  1.5e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.43 on 78 degrees of freedom
Multiple R-squared:  0.534,Adjusted R-squared:  0.528
F-statistic: 89.3 on 1 and 78 DF,  p-value: 1.46e-14
```

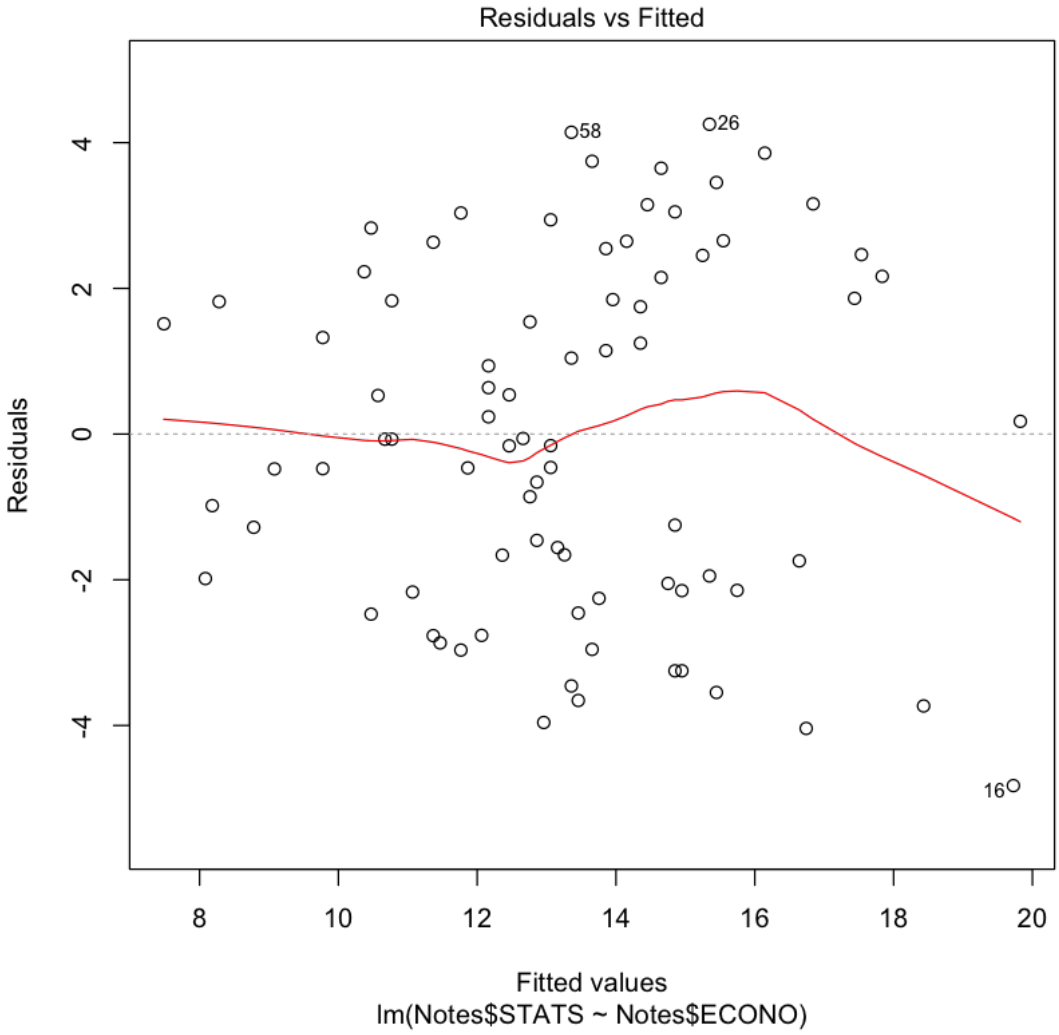
On obtient le tablea d'analyse de la variance comme suit :

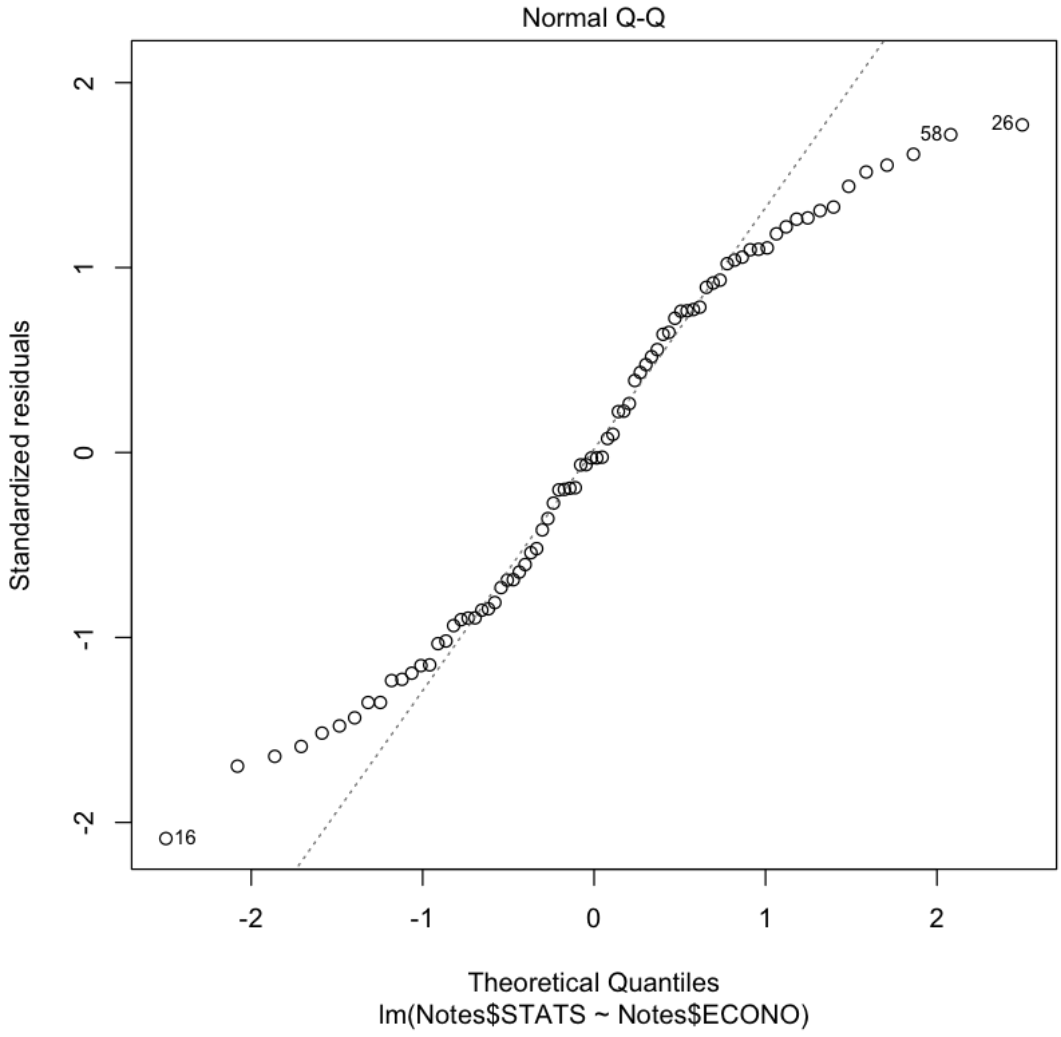
```
In [26]: anova(reg)
```

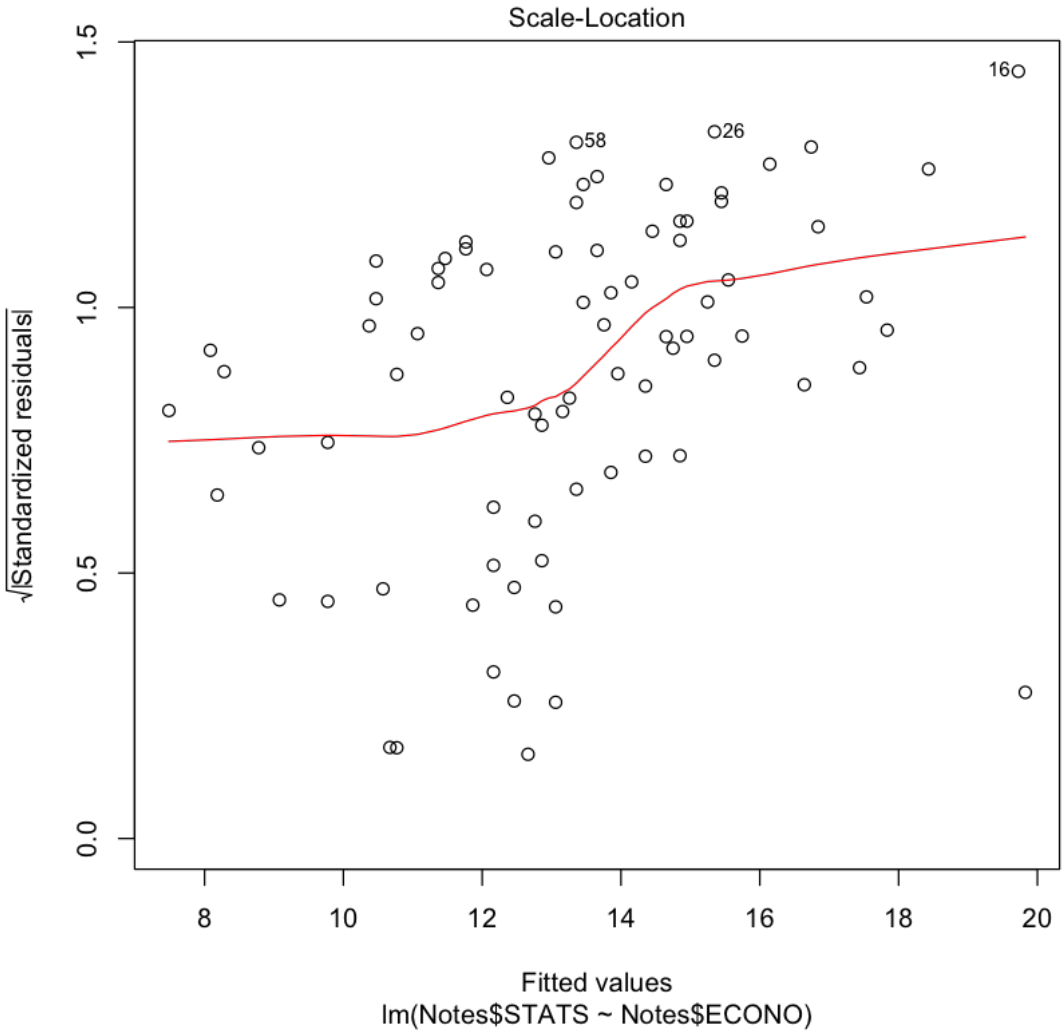
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Notes$ECONO	1	525.3	525.281	89.28	1.457e-14
Residuals	78	458.9	5.883	NA	NA

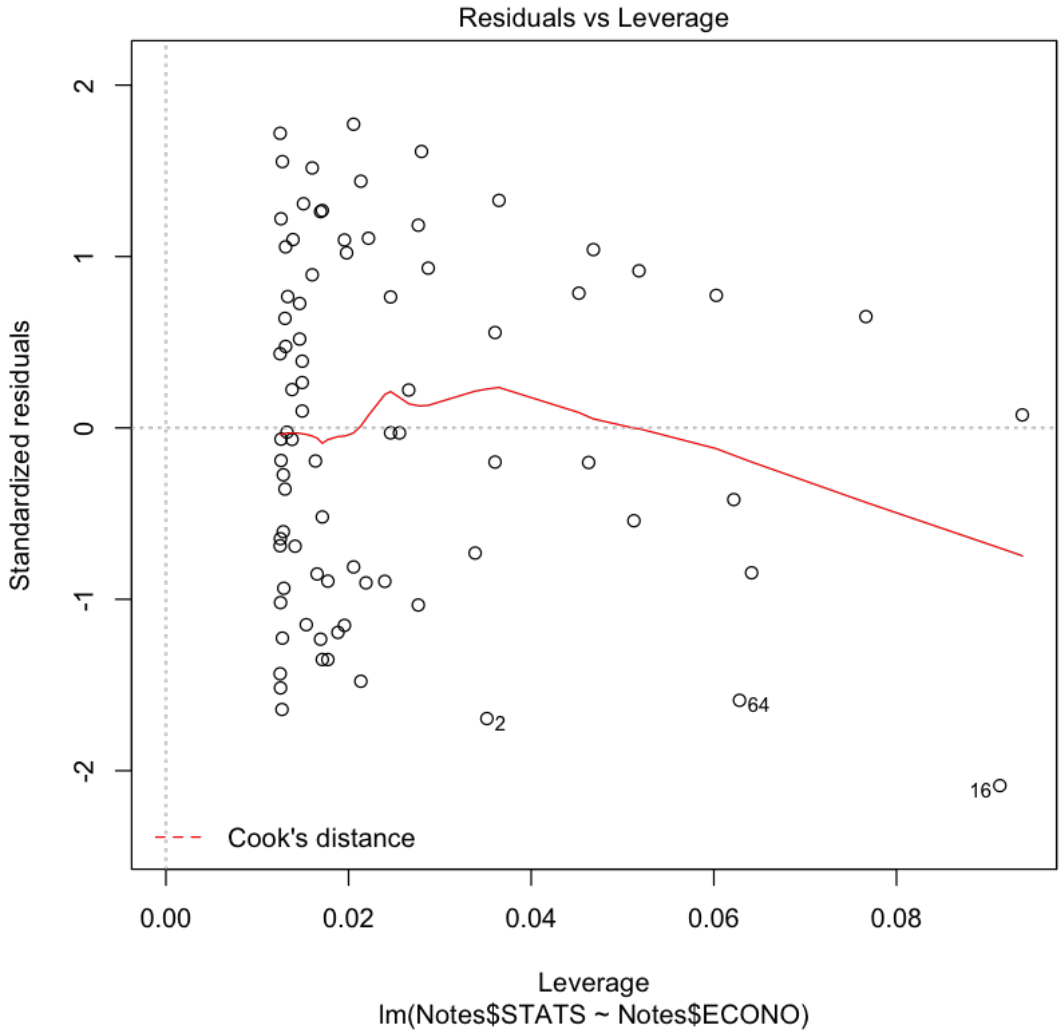
Pour obtenir quelques graphiques sur les résidus, on utilise "plot(objet)". Dans l'exemple ci-dessus "plot(reg)" permet d'obtenir, entre autres, le graphique des résidus en fonction des valeurs prédites et le graphique de probabilté normale des résidus.

```
In [27]: plot(reg)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Notes$STATS ~ Notes$ECONO)
58
26
16
4
2
0
-2
-4
8
10
12
14
16
18
20


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Notes$STATS ~ Notes$ECONO)
16
58
26
-2
-1
0
1
2


Scale-Location
√|Standardized residuals|
1.5
1.0
0.5
0.0
16
58
26
8
10
12
14
16
18
20
Fitted values
lm(Notes$STATS ~ Notes$ECONO)



In [ ]:

Tel que mentionné plu haut, on peut simplement utliser les noms des variables dans une commande en y ajoutant le nom du fichier de données. On peut vérifier que l'objet "reg2" ci-dessous contient les mêmes résultats que "reg" créé précédemment.

```
In [28]: reg2<-lm(STATS~ECONO, data=Notes)
```

In [ ]:

R possède un grand nombre de fonctions. Si vous voulez savoir plus sur une fonction donnée, faites ?fun( ), où fun est le nom de la fonction en question. Par exemple, pour la fonction plot, on a :

```
In [31]: ?plot()
```