Statistiques Descriptives en R (2)

October 17, 2018

Greta Laage, Luc Adjengue

Contents

1 Importation de données

On s'intéresse ici aux concepts de base en statistique et à la description des données. Il est possible de charger des données sous plusieurs formats (.csv, .txt) ou de les définir à la main.

1.1 Importation de données dans un fichier csv avec la fonction read.csv

Il y a plusieurs paramètres à spécifier:

- le nom du fichier
- la présence d'une première ligne avec les noms de colonne: *header*
- le type de séparation des colonnes: *sep*
- le type de caractère pour les décimales ("." ou ","): *dec*

```
In [ ]: # importer un fichier
        data <- read.csv("nom_du_fichier.csv", header = TRUE, sep = ";",dec = ".")
```

1.2 Définition des données à la main

- soit sous forme de tableau
- soit sous forme de vecteurs pour les valeurs numériques

```
In [19]: # Déclaration d'un dataframe avec des données
         donnees = data.frame(taille =c(185,194,165,175,172,150,165),
                              poids=c(82,90,55,65,68,45,64),
                              groupe=c("M","M","F","F","F","F","M"),
                              row.names=c("A","B","C","D","E","G","H"))

         #ăAffichageăduătableauădeădonnees
         donnees
```

	taille	poids	groupe
A	185	82	M
B	194	90	M
C	165	55	F
D	175	65	F
E	172	68	F
G	150	45	F
H	165	64	M

```
In [17]: # Définir des données sous forme de vecteur
         X=c(14,18,40,43,45,112)
         Y=c(280,350,470,500,560,1200)
```

La notation *c()* en R permet de définir un vecteur ie une liste de valeurs.

1.3 Conseils utiles

Pour accéder à une colonne en particulier d'un tableau de données, vous pouvez utiliser le signe $ et le nom de la colonne. Cette astuce peut être utilisée dans de nombreuses situations avec des objets de R. Elle sera beaucoup utilisée lors des calculs de régression linéaire.

```
In [21]: donnees$taille
```

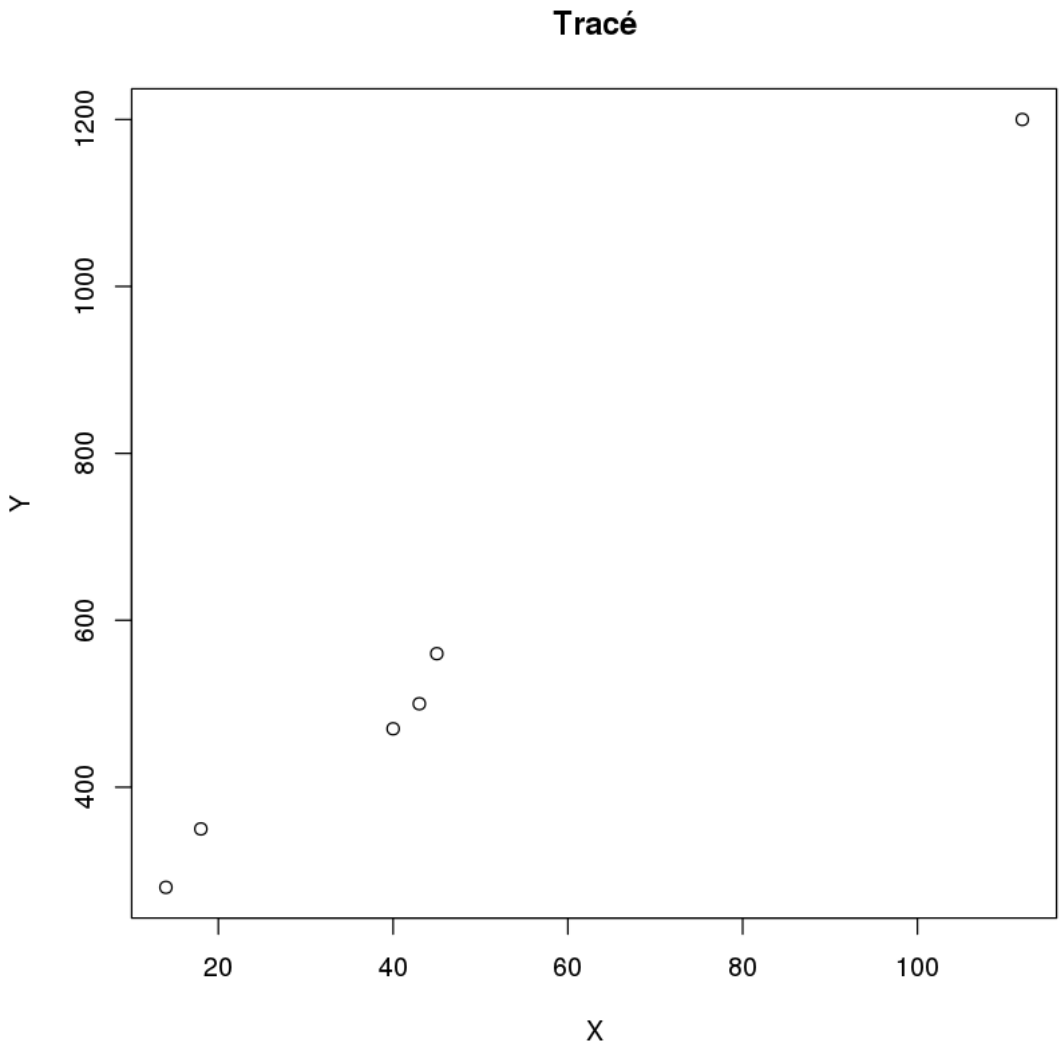
1. 185 2. 194 3. 165 4. 175 5. 172 6. 150 7. 165

2 La présentation graphique de données

2.1 Le nuage de points: fonction plot()

- *xlab* (*ylab*) est un paramètre optionnel qui définit un nom pour l'axe des abscisses (des ordonnées)
- *main* définit le nom du graphe

```
In [39]: # Nuage de points simple des données X et Y
         plot(X,Y,xlab = 'X',ylab = 'Y',main = 'Tracé')
```



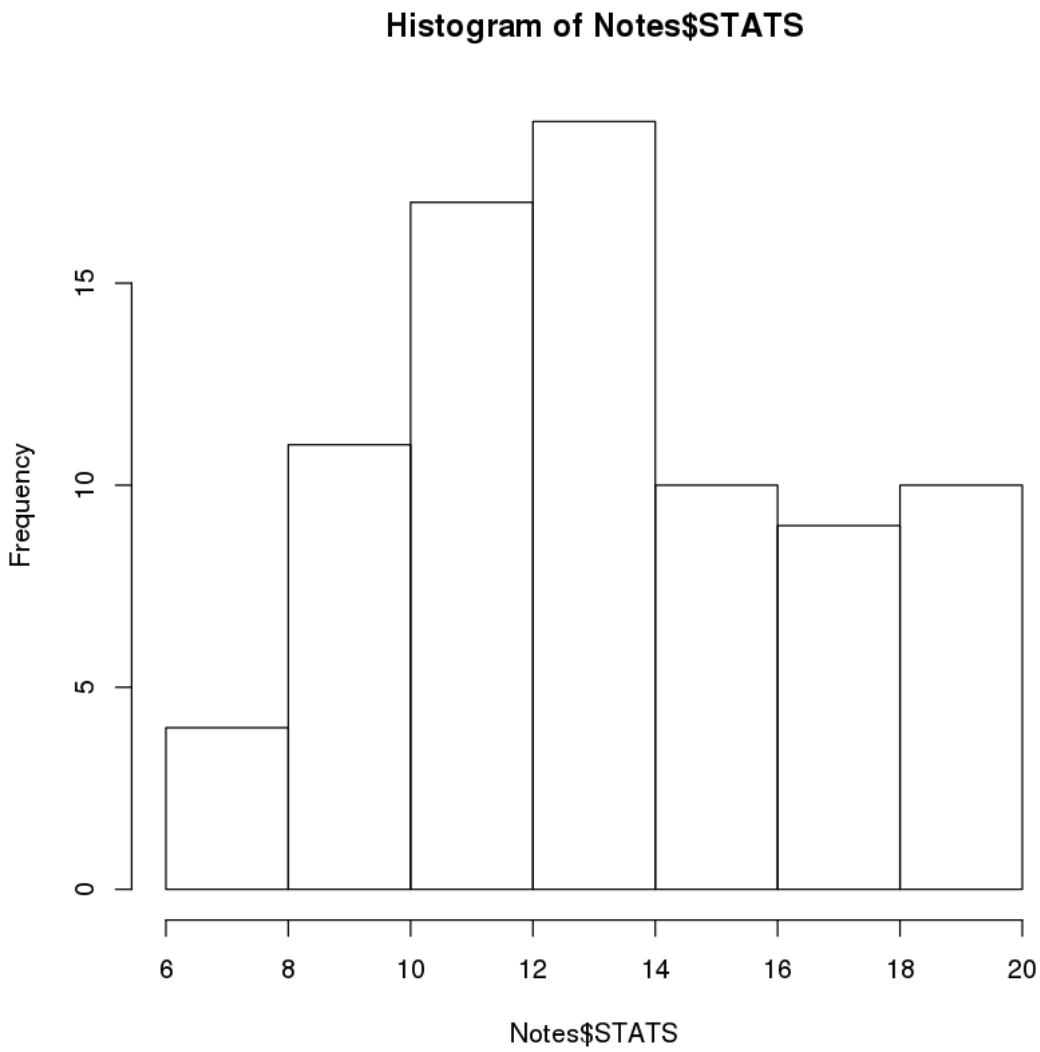
Pour illustrer les commandes vues dans ce notebook, j'ai utilisé les données 'Notes' du TD7.

```
In [25]: Notes <- read.csv("Notes.csv", header = TRUE, sep = ";",dec = ".")
```

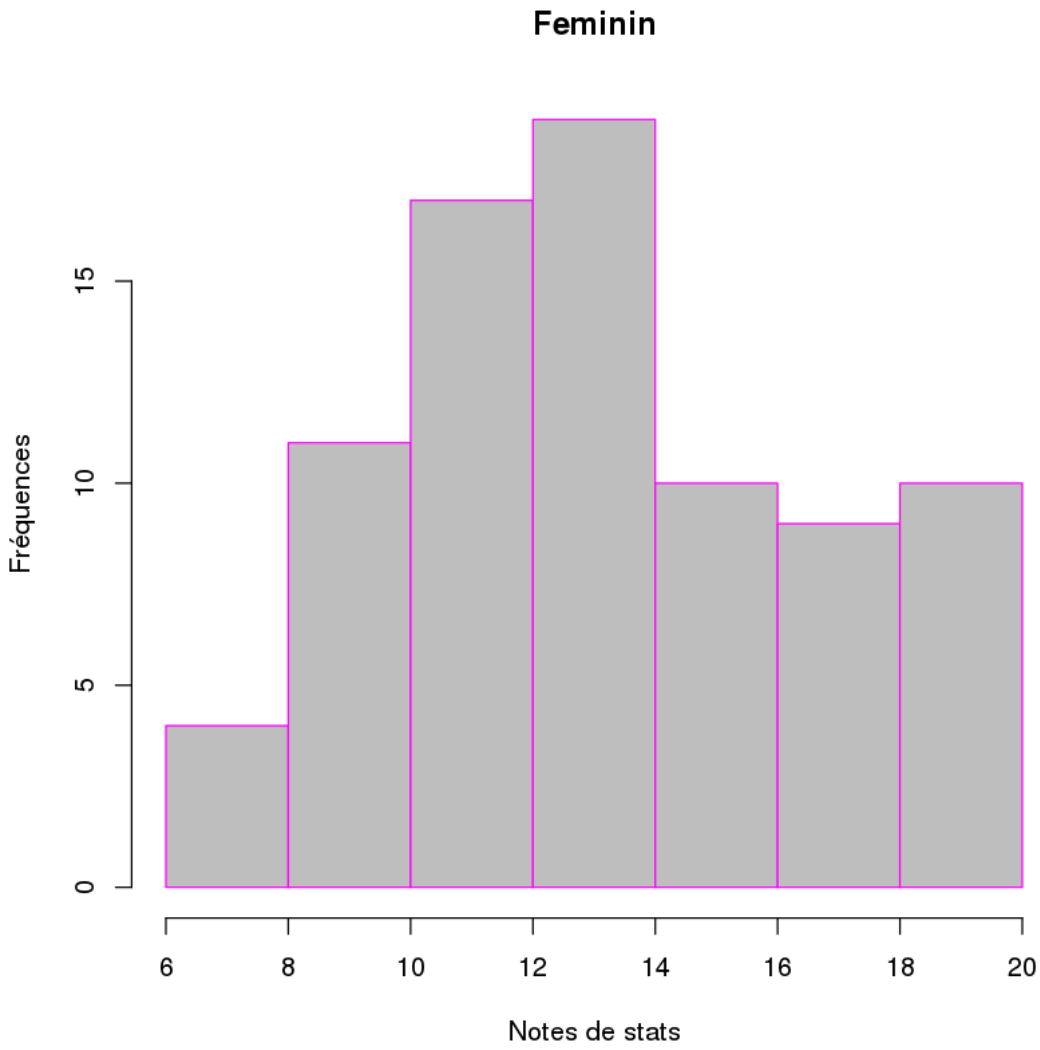
2.2 Les histogrammes: fonction *hist()*

2.2.1 Tracé d'un histogramme

```
In [26]: # histogramme simple d'une colonne du tableau de données
         hist(Notes$STATS)
```

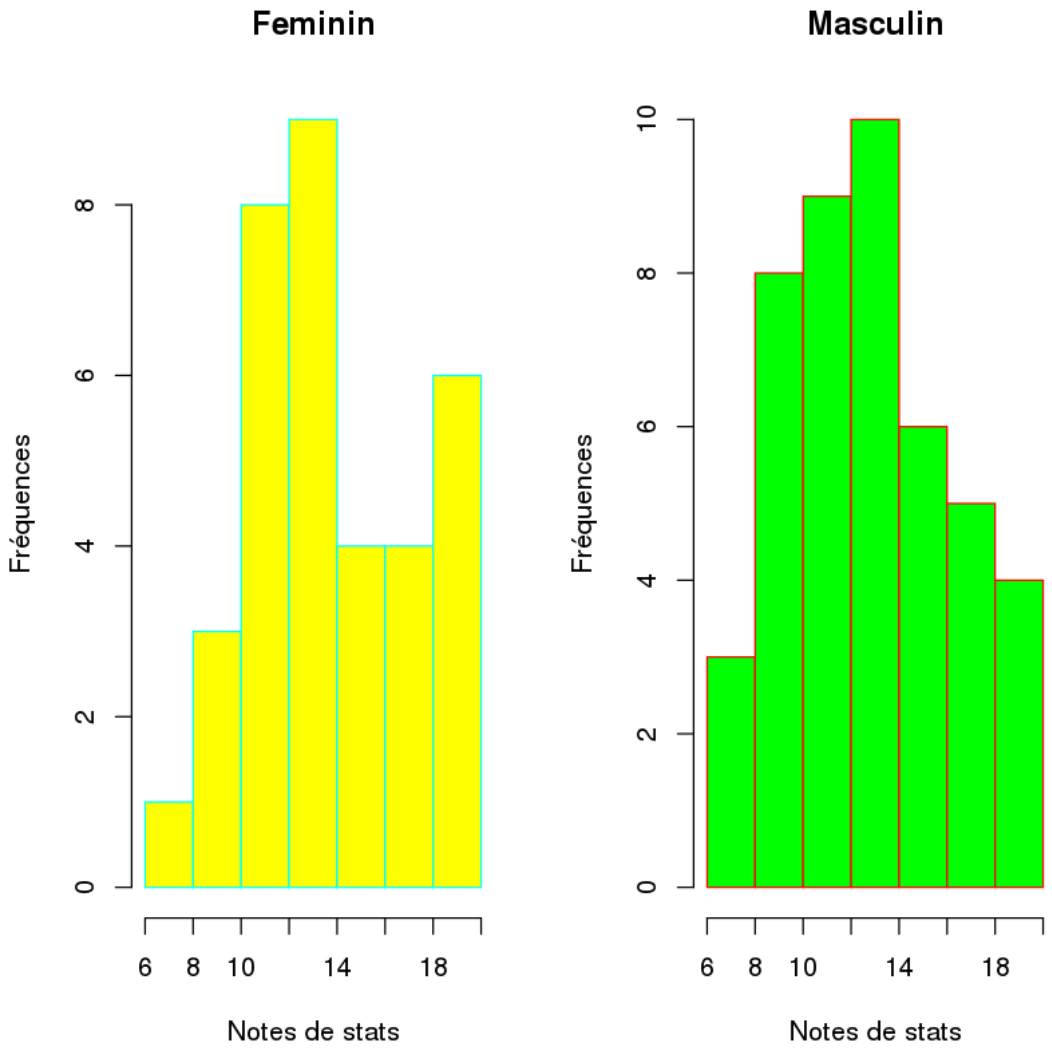


In [27]:
```
# Histogramme des mêmes données que ci-dessus
# avec des paramètres supplémentaires d'affichage
hist(Notes$STATS, col="grey",border="magenta", main=paste("Feminin"),
     xlab="Notes de stats",ylab="Fréquences")
```



2.2.2 Tracé de deux histogrammes juxtaposés

```
In [28]: 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")
```

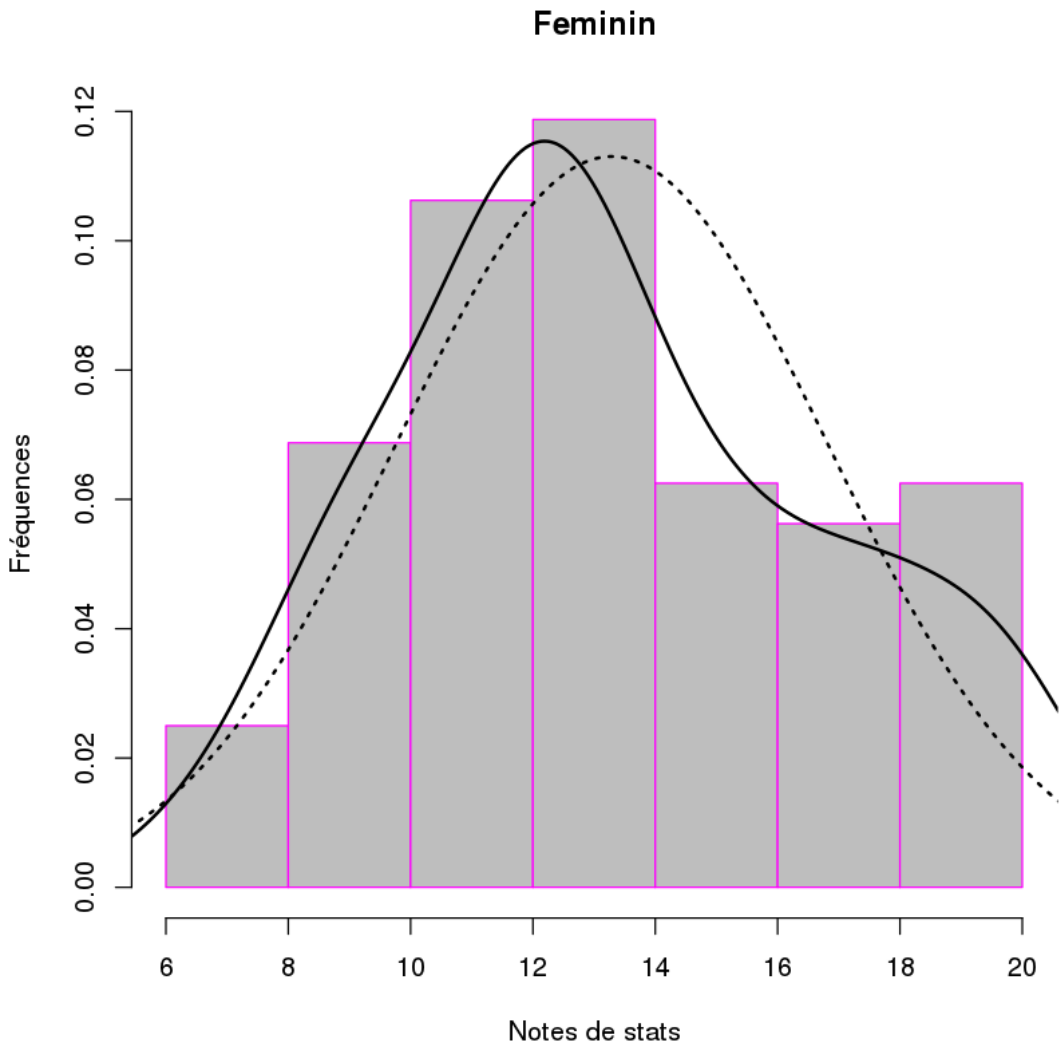


On peut rajouter à l'histogramme la courbe de densité d'une loi normale et celle estimée pour les données. Dans l'appel à la fonction hist(), il faut mettre le paramètre freq à FALSE (freq=FALSE)

Pour les courbes ci-dessus, on a ajouté les paramètres:

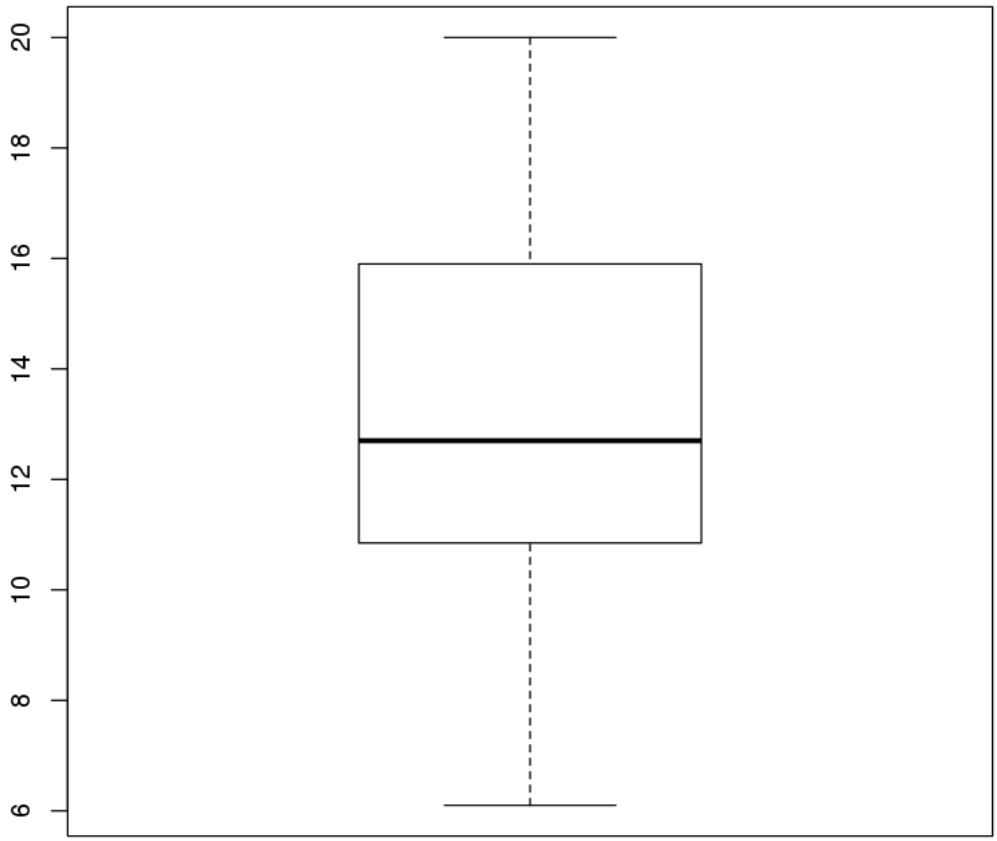
- *lwd* qui fait référence à l'épaisseur de la courbe
- *lty* qui fait référence au type de courbe: pointillée, trait,..

```
In [33]: 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)
         # densité d'une loi normale
         lines(x, dnorm(x, mean(Notes$STATS), sd(Notes$STATS)), lwd=2, lty=3)
```



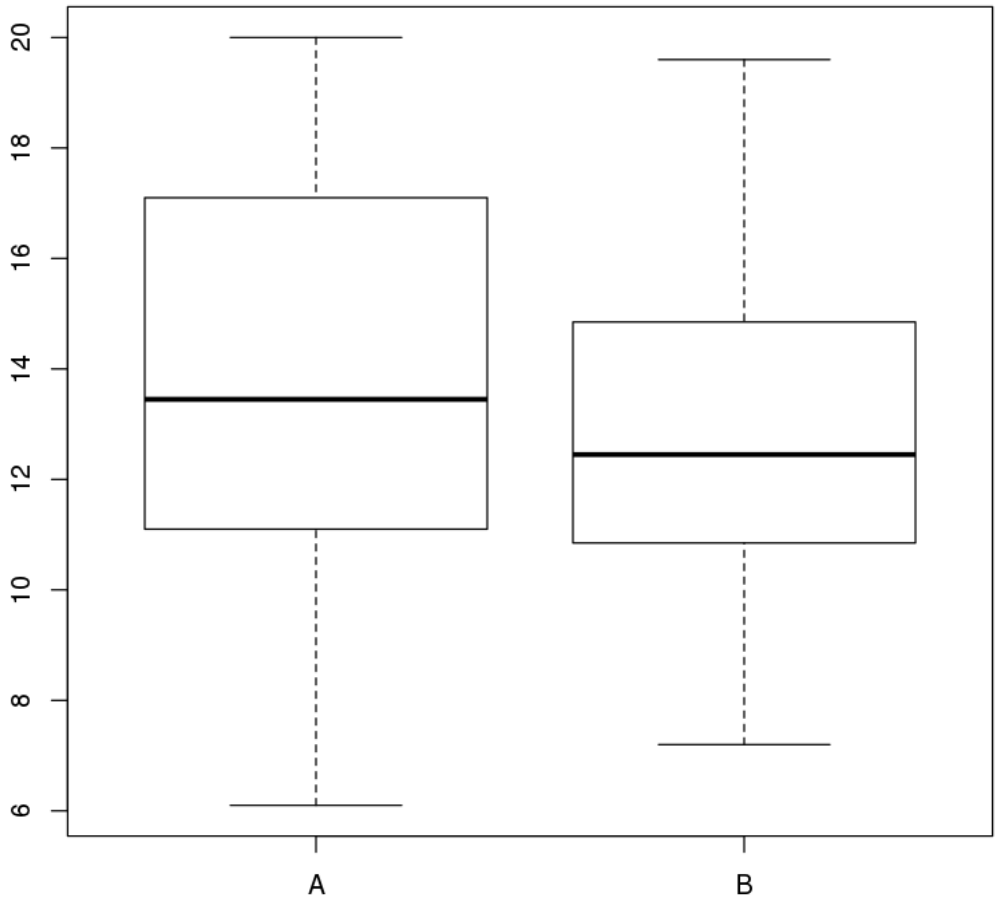
2.3 Diagramme de Tukey: la fonction *boxplot()*

```
In [34]: #Diagramme de Tukey des notes de stats:
         boxplot(Notes$STATS)
```

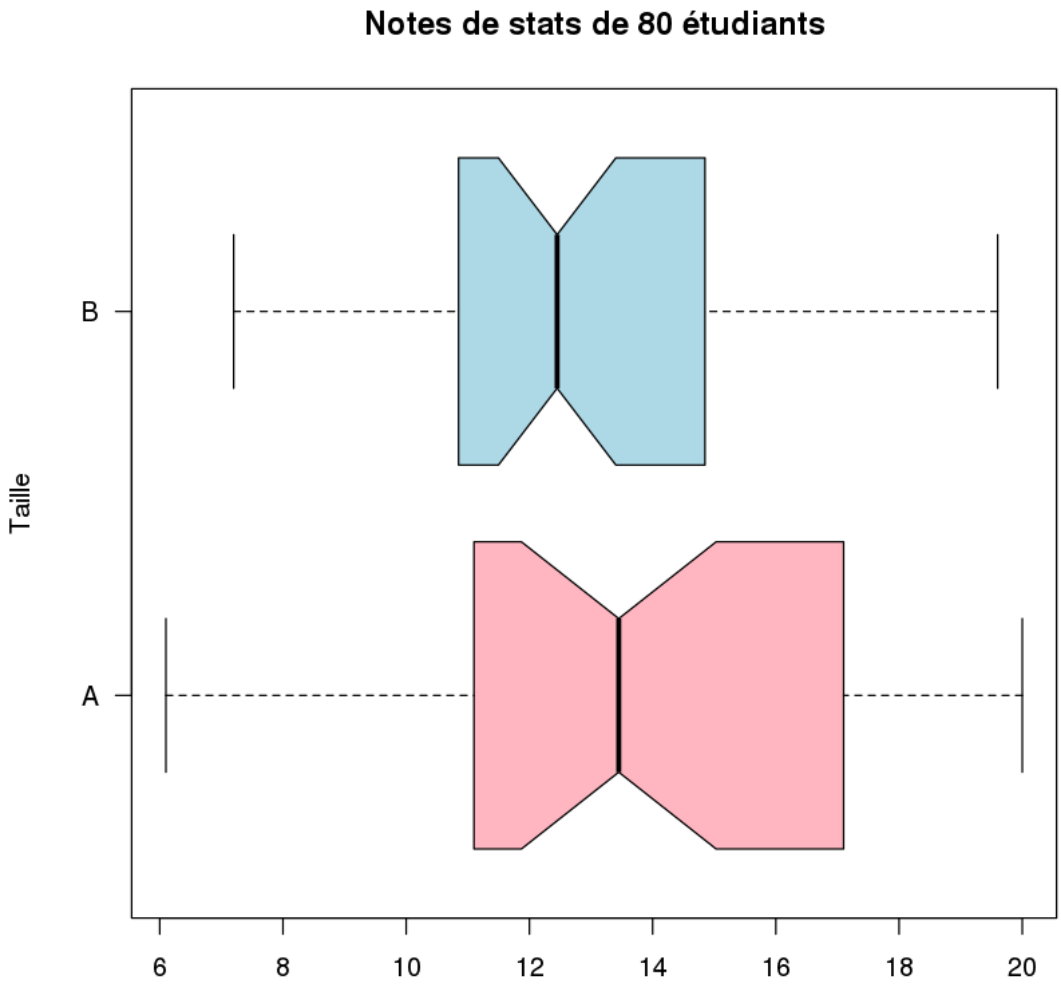


Si les données peuvent être partitionnées selon une colonne, comme c'est le cas ici par groupe (A ou B) ou par sexe (M ou F), alors on peut tracer le diagramme de Tukey pour chacune des valeurs de la partition.

```
In [35]: # Diagramme de Tukey pour deux différents groupes dans les données
         boxplot(STATS~GROUP, data=Notes)
```



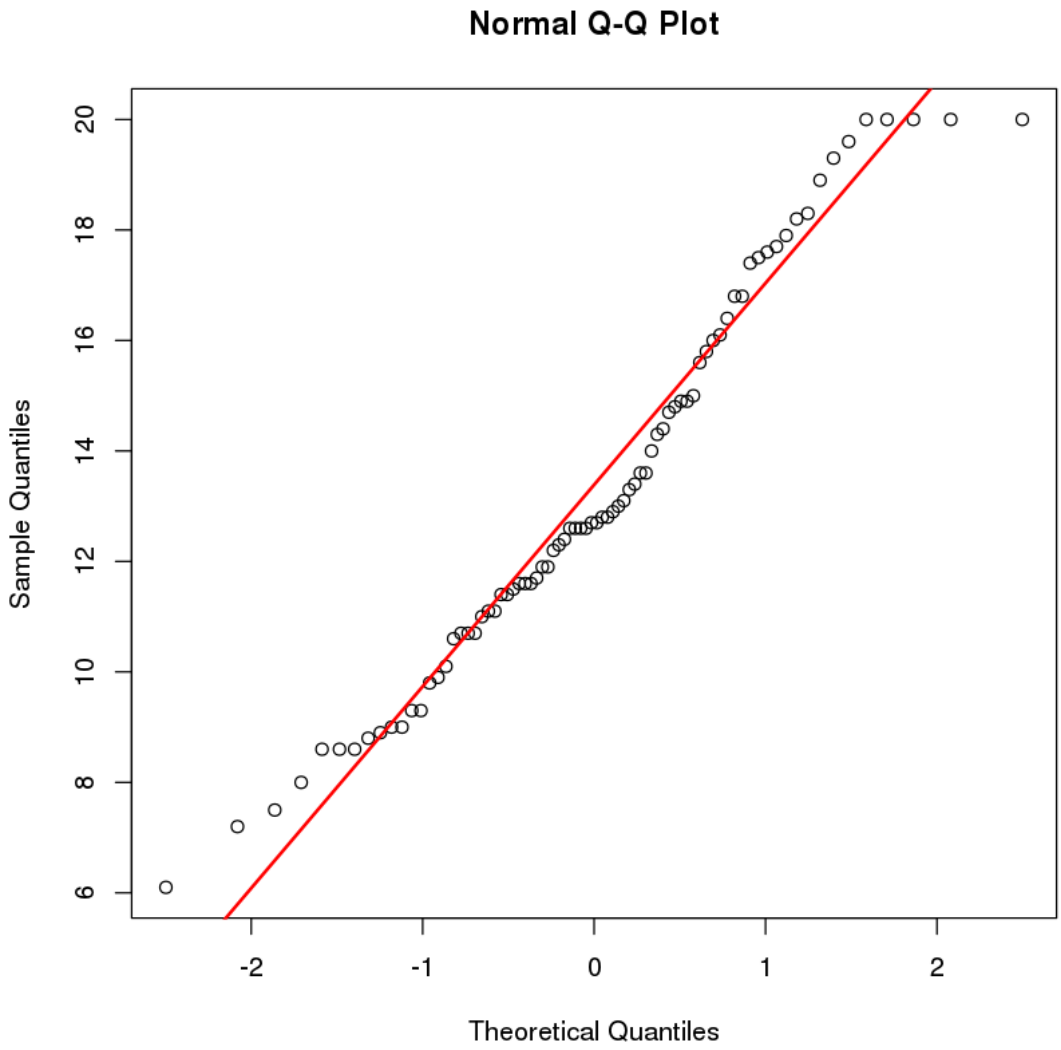
In [36]:
```
# On peut modifier l'affichage des diagrammes pour ajouter de la couleur, un titre, .
boxplot(Notes$STATS~Notes$GROUP,
col=c("lightpink","lightblue"),
horizontal=TRUE,
notch=TRUE,
main=paste("Notes de stats de",nrow(Notes),"étudiants"),
ylab="Taille",
las=1)
```



2.4 Le diagramme quantile-quantile : fonction qqnorm()

Diagramme quantile-quantile des notes de stats

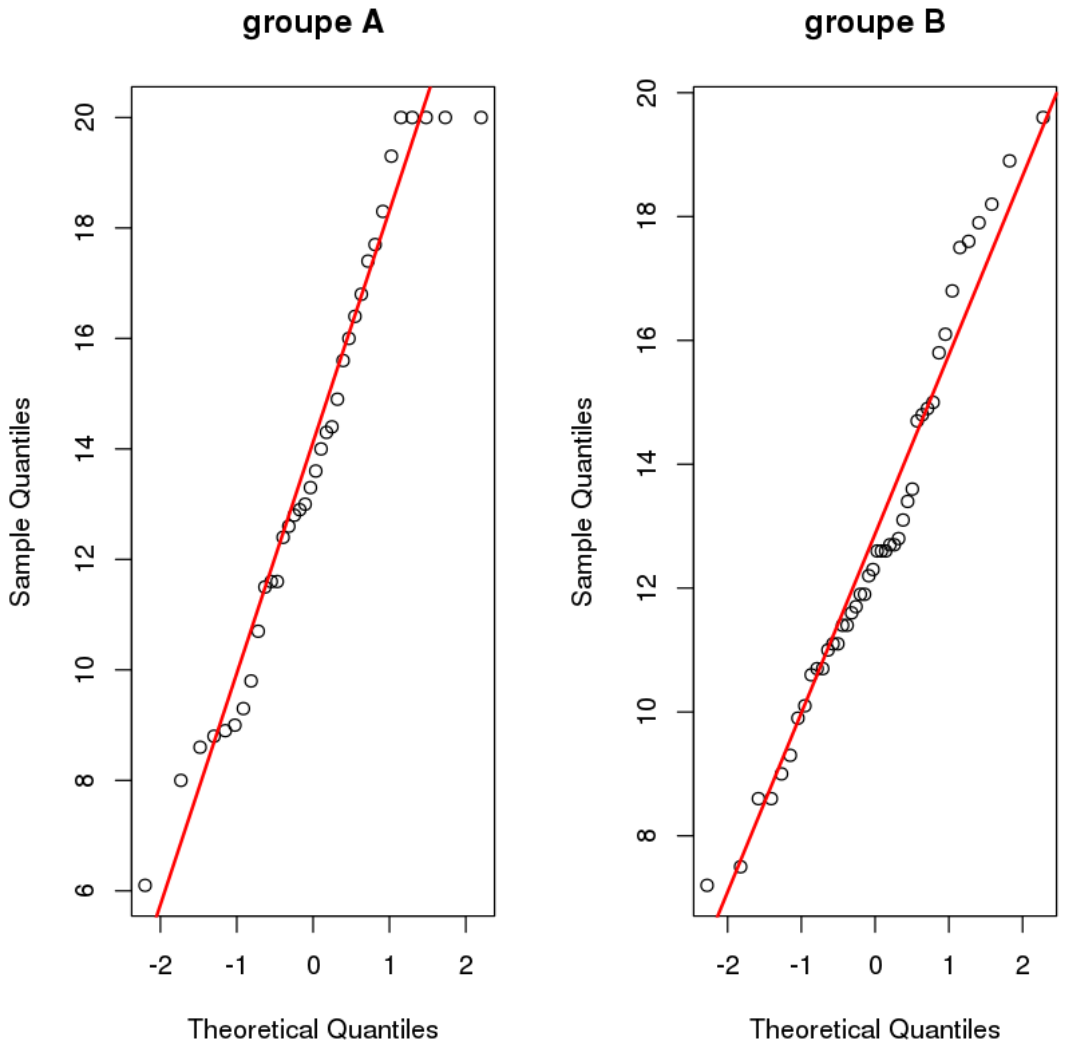
```
In [42]: qqnorm(Notes$STATS)
         qqline(Notes$STATS, col="red", lwd=2) # droite des quantiles de la loi normale
```



Lorsqu'on peut faire une partition des données sur les valeurs d'une colonne:

```
In [41]: 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)
```



3 Mesures de position

Lorsqu'on souhaite appliquer le même fonction f à plusieurs ensembles de données différents et avoir le résultat sous forme de liste, on peut utiliser la fonction **sapply()** dont les paramètres sont:

- l'index qui varie pour indiquer les différents ensembles à considérer
- le paramètre à faire varier dans la fonction souhaitée f
- la fonction f

```
In [46]: # Par exemple, calcul de la moyenne pour les colonnes 2 à 3
         # du tableau de données "Notes"
         sapply(2:3, function(i) mean(Notes[,i]))
```

1. 12.33375 2. 13.29125

3.1 Calcul de la moyenne

```
In [48]: # la moyenne des notes de stats
         mean(Notes$STATS)

         # la moyenne des notes de stats du groupe A
         mean(Notes$STATS[Notes$GROUP=="A"])
```

13.29125
13.8777777777778

3.2 Calcul de la médiane

```
In [50]: # la moyenne des notes de stats
         median(Notes$STATS)

         # la moyenne des notes de stats du groupe A
         median(Notes$STATS[Notes$GROUP=="A"])
```

12.7
13.45

4 Mesures de dispersion

4.1 Calcul des quartiles

```
In [53]: # Calcul des quartiles des notes de stats
         quantile(Notes$STATS)
```

0\% 6.1 **25\%** 10.925 **50\%** 12.7 **75\%** 15.85 **100\%** 20

4.2 Calcul de l'écart-type et de la variance

```
In [54]: # l'écart-type des notes de stats
         sd(Notes$STATS)

         # la variance des notes de stats
         var(Notes$STATS)
```

3.52959257339671
12.4580237341772

5 Coefficient d'asymétrie et d'aplatissement

Besoin d'une nouvelle librairie ("moments") pour faire le calcul du coefficient d'asymétrie et d'aplatissement.

```
In [56]: install.packages("moments")
         library(moments)
```

```
Installing package into /home/laagre/R/x86_64-pc-linux-gnu-library/3.4
(as lib is unspecified)
```

```
In [57]: # asymétrie (skewness) des notes de stats
         skewness(Notes$STATS)

         # aplatissment (kurtosis) des notes de stats selon la formule du cours,
         # d'où le -3 ci-dessous
         kurtosis(Notes$STATS) -3
```

0.300407083885415
-0.711856247025102

6 Coefficient de corrélation

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 [59]: #Calcul de la covariance entre les notes de stats et les notes d'éco
         cov(Notes$STATS, Notes$ECONO)

         # Calcul du coefficient de corrélation de Pearson
         # entre les notes de stats et les notes d'éco
         cor(Notes$STATS, Notes$ECONO)
```

6.68207120253165
0.730563270676972