Université de Technologie de Compiègne
SY09 : Analyse des données et Apprentissage automatique

SY09 : TP 02 : Analyse de données, statistique descriptive

Julien Jerphanion

Printemps 2018

Table des matières

1 Analyse exploratoire des Iris de Fisher

1.1 Remarque sur les data frames

Les lignes représentent les individus ; les colonnes les variables potentiellement de types différents.

Un dataframe `D` peut être accéder sur ces colonnes avec les syntaxe suivantes:

```
D$columnName
D[,"columnName"]
```

On peut aussi accéder à la caractéristique d'un individu de cette façon:

```
D$columnName[2]
D[2,"columnName"]
```

De manière plus générale, on peut avoir accès aux noms des lignes et des colonnes ainsi:

```
rownames(D)
colnames(D)
```

1.2 Résumés numériques et analyses graphiques.

On charge le jeu de données `iris` en mémoire ainsi:

```
data(iris)
```

On peut ensuite effectuer une analyse exploratoire des données ainsi. Ici on va effectuer une analyse globale du jeu de données:

```
class(iris)
```

```
## [1] "data.frame"
```

```
names(iris)
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"
## [5] "Species"
```

```
iris[,1]
```

```
##   [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4
##  [18] 5.1 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5
##  [35] 4.9 5.0 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0
##  [52] 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8
##  [69] 6.2 5.6 5.9 6.1 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4
##  [86] 6.0 6.7 6.3 5.6 5.5 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8
## [103] 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7
## [120] 6.0 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7
## [137] 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8 6.7 6.7 6.3 6.5 6.2 5.9
```

```
iris$Sepal.Length
```

```
##   [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4
##  [18] 5.1 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5
##  [35] 4.9 5.0 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0
##  [52] 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8
##  [69] 6.2 5.6 5.9 6.1 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4
##  [86] 6.0 6.7 6.3 5.6 5.5 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8
## [103] 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7
```

```
## [120] 6.0 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7
## [137] 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8 6.7 6.7 6.3 6.5 6.2 5.9
```

```
class(iris[,1])
```

```
## [1] "numeric"
```

```
class(iris$Species)
```

```
## [1] "factor"
```

```
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

Réalisons une analyse graphique des données

```
apply(iris[,1:4],2,mean)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##     5.843333     3.057333     3.758000     1.199333
```

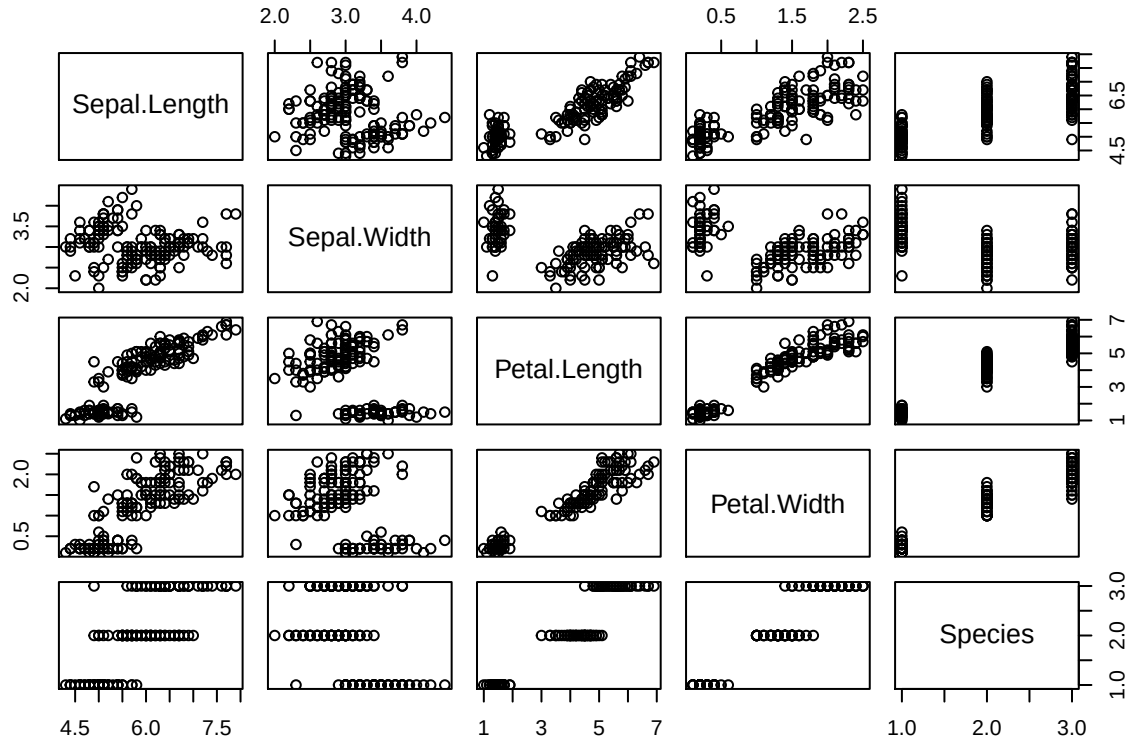
```
cor(iris[,1:4])
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
## Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
## Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
## Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

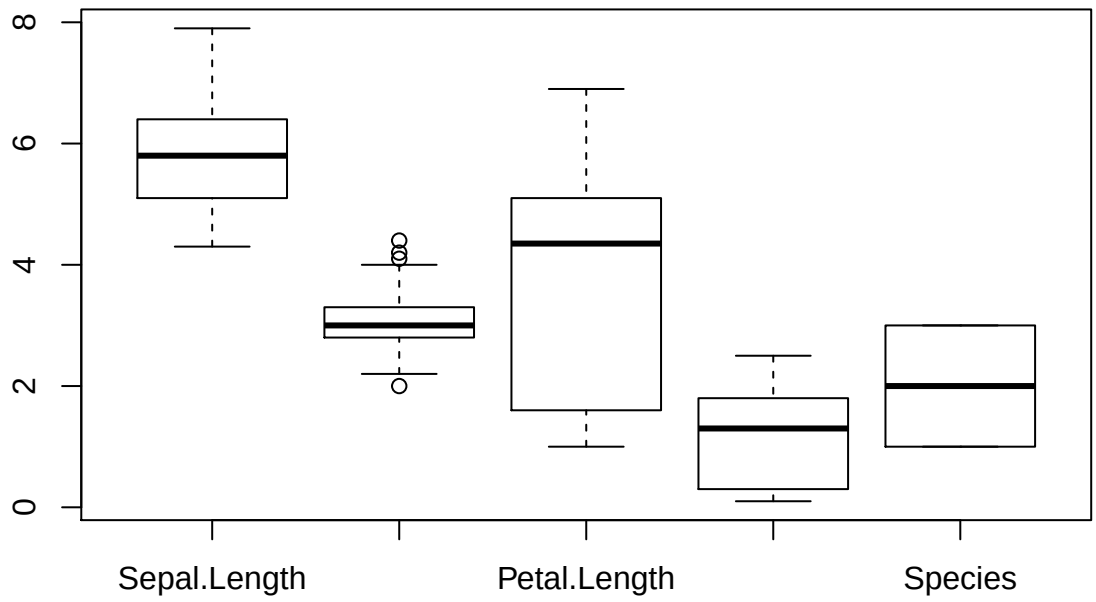
```
print(cor(iris[,1:4]),digits=3)
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length        1.000      -0.118        0.872       0.818
## Sepal.Width        -0.118       1.000       -0.428      -0.366
## Petal.Length        0.872      -0.428        1.000       0.963
## Petal.Width         0.818      -0.366        0.963       1.000
```

```
plot(iris)
```

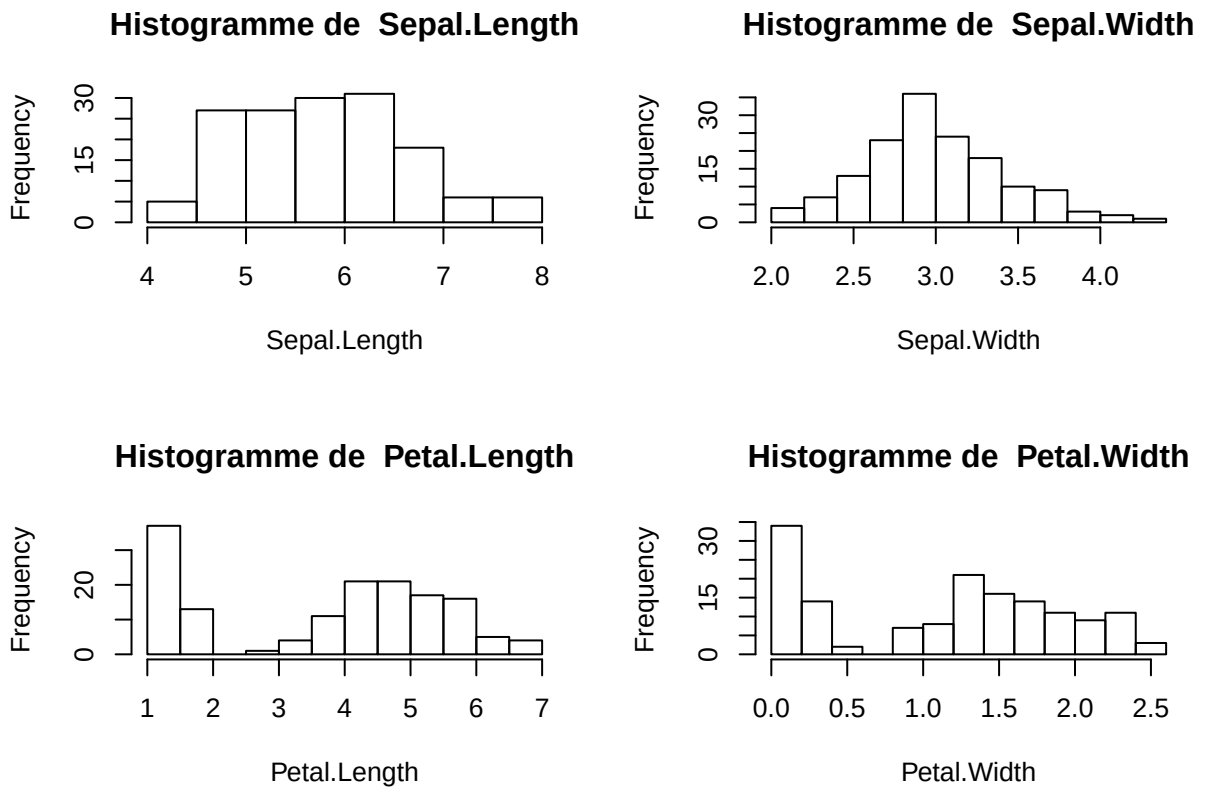



```
boxplot(iris)
```



Il est possible d'afficher plusieurs graphiques sur la même fenêtre :

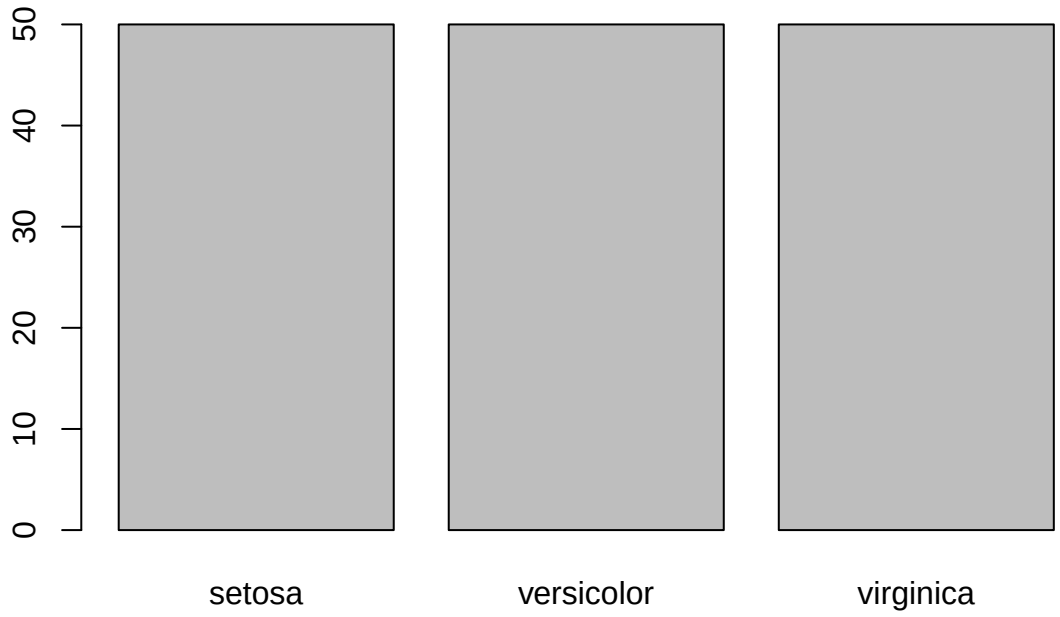
```
def.par <- par(no.readonly=T)
par(mfrow=c(2,2))
for(i in 1:4) {
  titre = colnames(iris)[i]
  hist(iris[,i],xlab=titre,main=paste("Histogramme de ",titre))
}
```




```
par(def.par)
```

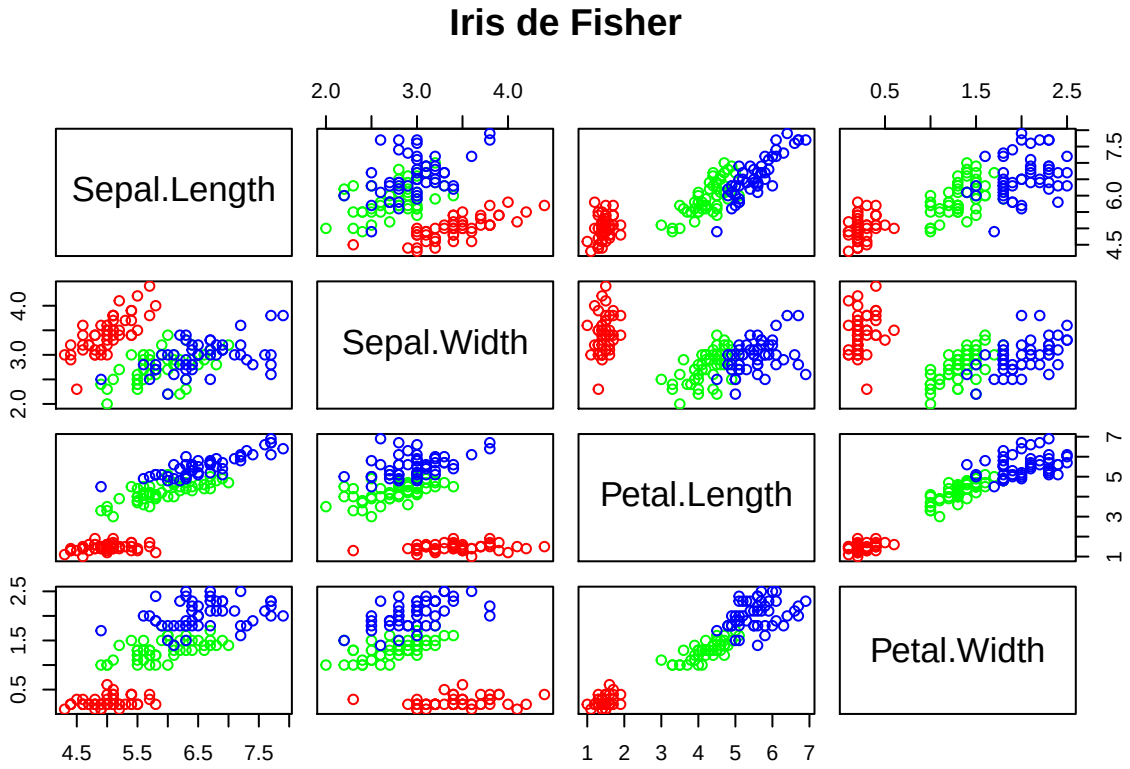
La commande suivante retourne un diagramme en bâtons des effectif des classes du jeu de données:

```
barplot(summary(iris$Species))
```

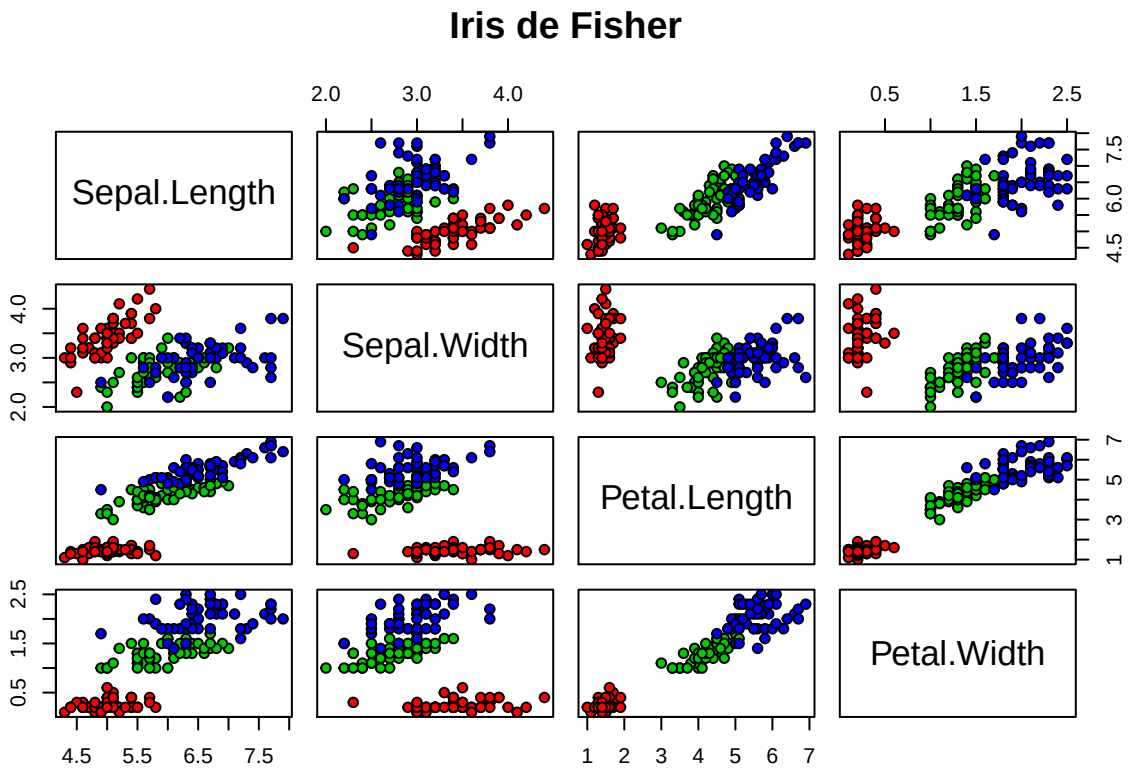


On peut aussi explorer le jeu de données ainsi:

```
plot(iris[,1:4], main="Iris de Fisher", col=c("red","green","blue")[iris$Species])
```




```
pairs(iris[,1:4], main="Iris de Fisher",pch=21,bg=c("red","green3","blue")[iris$Species])
```



Ici, les individus sont représentés par l'ensemble des projections de ceux-ci sur deux de leur modalités. Cet ensemble de graphe de dispersions permet de se faire une idée de la répartition des données et des relations sous-jacentes des variables.

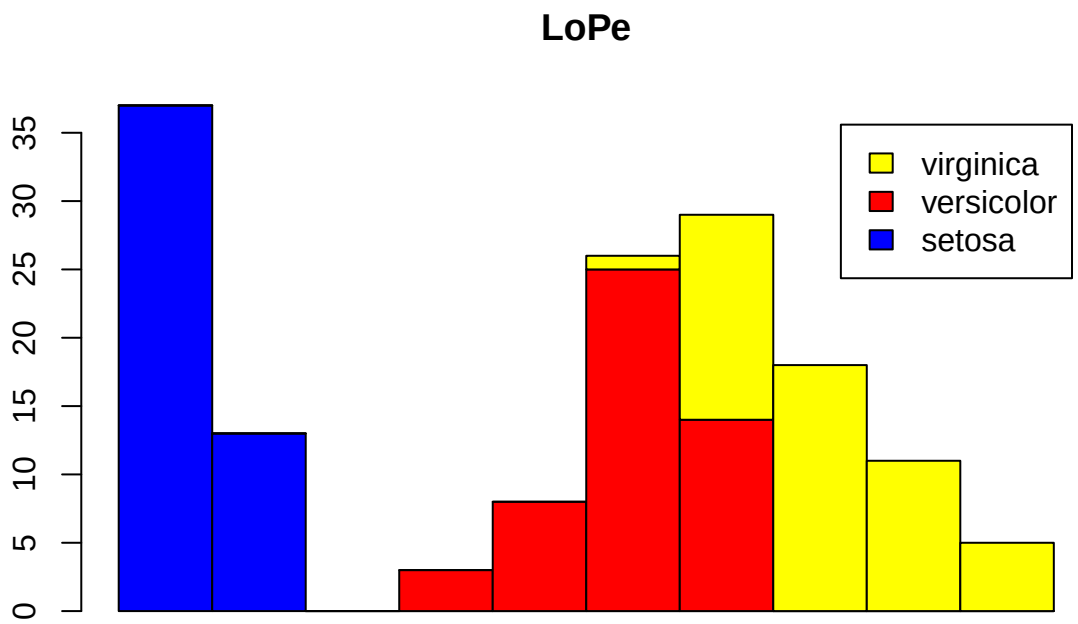
De manière plus technique, seules les quatres premières colonnes sont traitées, la dernière étant représentés en utilisant des couleurs (pour `plot` uniquement)

Si l'on souhaite travailler de manière plus directe avec le jeu de données, on peut l'*attacher* au chemin utilisé par de `R` avec la commande `attach` ; c'est ce que l'on va faire avec `iris` pour avoir un niveau d'indirection de moins.

```
attach(iris)
```

Une fois cela executé, on peut faire des histogrammes plus sympathiques ainsi:

```
inter <- seq(min(Petal.Length),max(Petal.Length),by=(max(Petal.Length)-min(Petal.Length))/10)
h1 <- hist(plot=F,Petal.Length[Species=="setosa"],breaks=inter)
h2 <- hist(plot=F,Petal.Length[Species=="versicolor"],breaks=inter)
h3 <- hist(plot=F,Petal.Length[Species=="virginica"],breaks=inter)
barplot(rbind(h1$counts,h2$counts,h3$counts),space=0,
legend=levels(Species),main="LoPe",col=c("blue","red","yellow"))
```



On a ici un histogramme de la longueur des pétales sur les différentes classes. La contribution de chaque classe sur les subdivisions apparait en couleurs.

On peut exporter le résultat en `eps` avec:

```
fileName = "iris.eps"
postscript(fileName,horizontal=F,width=12/2.5,height=12/2.5)
pairs(iris[2:5],main="Les Iris",pch=21,bg=c("red","green3","blue")[Species])
dev.off()
```

```
## pdf
##   2
```

On peut aussi exporter ça en`pdf` pour être au goût du jour:

```
fileName = "iris.pdf"
pdf(fileName)
pairs(iris[2:5],main="Les Iris",pch=21,bg=c("red","green3","blue")[Species])
dev.off()
```

```
## pdf
##   2
```

Finalement, on peut détacher le jeu de données ainsi:

```
detach(iris)
```

1.3 Exercice: fonction `hist.factor`

On peut généraliser le snippet suivant pour dresser un histogramme d'une variable quantitative à partir d'une variable qualitative avec la fonction `hist.factor` suivante:

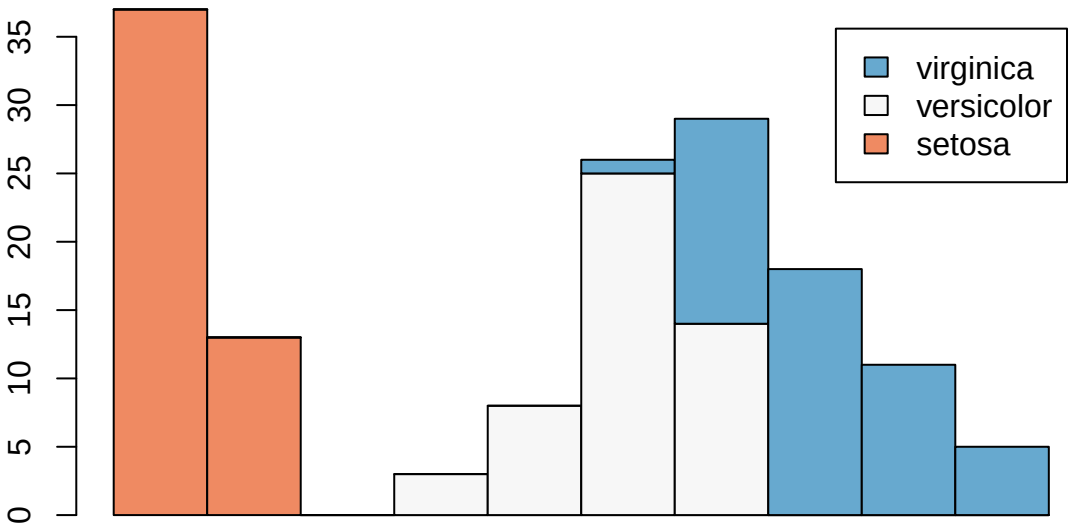
```
hist.factor = function(quantVar, qualVar) {

  minQuant = min(quantVar, na.rm=TRUE)
  maxQuant = max(quantVar, na.rm=TRUE)
  inter = seq(minQuant, maxQuant, by=(maxQuant-minQuant)/10)

  h = c()
  classes = unique(qualVar)
  n = length(classes)

  for(clazz in classes) {
    countClass = hist(plot=F,quantVar[qualVar==clazz],breaks=inter)$counts
    h = rbind(h,countClass)
  }

  barplot(h,space=0, legend=levels(qualVar),main=" ",col=brewer.pal(n, name = "RdBu"))
}
```

Ici, on utilise une bibliothèque particulière pour utiliser une palette de couleurs rococo, et ce, quelque soit le nombre de classes du jeu de données:

```
library("RColorBrewer")
```

On peut utiliser cette fonction pour retrouver le même résultat que tout à l'heure modulo cette nouvelle palette de couleurs:

```
hist.factor(iris$Petal.Length,iris$Species)
```



2 Analyse des notes du médian de SY02 (Printemps 2014)

On va s'amuserà voir si les étudiants de SY02 en P14 étaient bons. Pour cela, on charge tout d'abord le jeu de données présent dans le fichier `median-sy02-p2014.csv`. On supprime aussi les individus absents.

```
notes <- read.csv("donnees/median-sy02-p2014.csv", header=F, na.strings=c("NA","ABS"))
names(notes) <- c("branche","note")
notes <- notes[-which(is.na(notes$note)),]
```

On peut aussi utiliser pour importer le jeu de données:

```
read.table(fileName, sep=",", na.strings="ABS")
```

On peut vérifier que le jeu de données est intègre ainsi:

```
summary(notes)
```

```
##     branche         note
##  GB02   : 37   Min.   : 1.00
##  GU02   : 36   1st Qu.: 9.00
##  GM02   : 31   Median :11.00
##  GP02   : 18   Mean   :11.17
##  GS02   : 18   3rd Qu.:13.50
##  GM04   : 14   Max.   :19.50
##  (Other):121
```

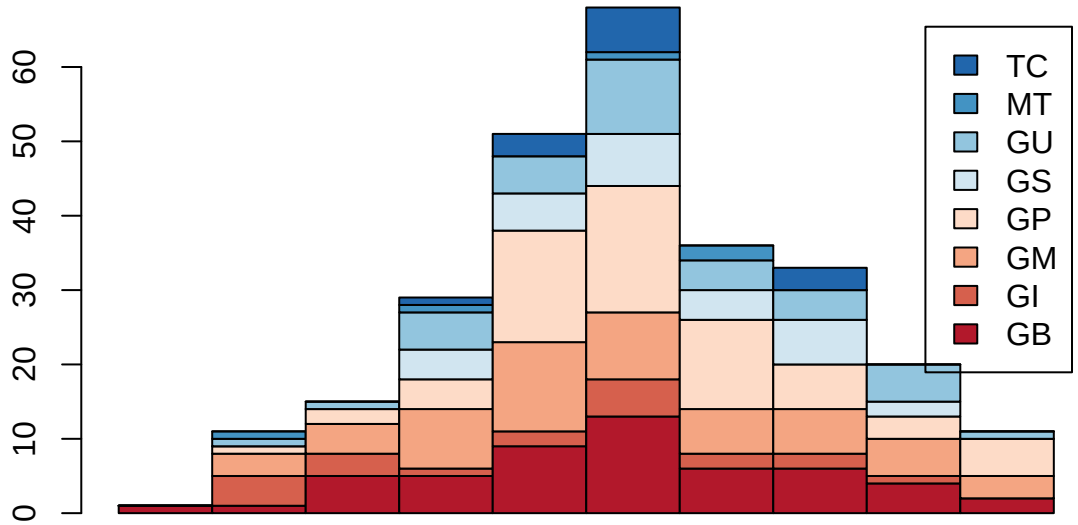
Les variables sont bien qualitatives ici : pas de problèmes de formattage.

On souhaite maintenant regarder les résultat des UVs entre branches plus qu'entre semestres d'études. Pour cela on va enlever le numéro du semestre:

```
notes$branche <- as.character(notes$branche)
notes$branche <- substr(notes$branche,1,2)
notes$branche <- as.factor(notes$branche)
```

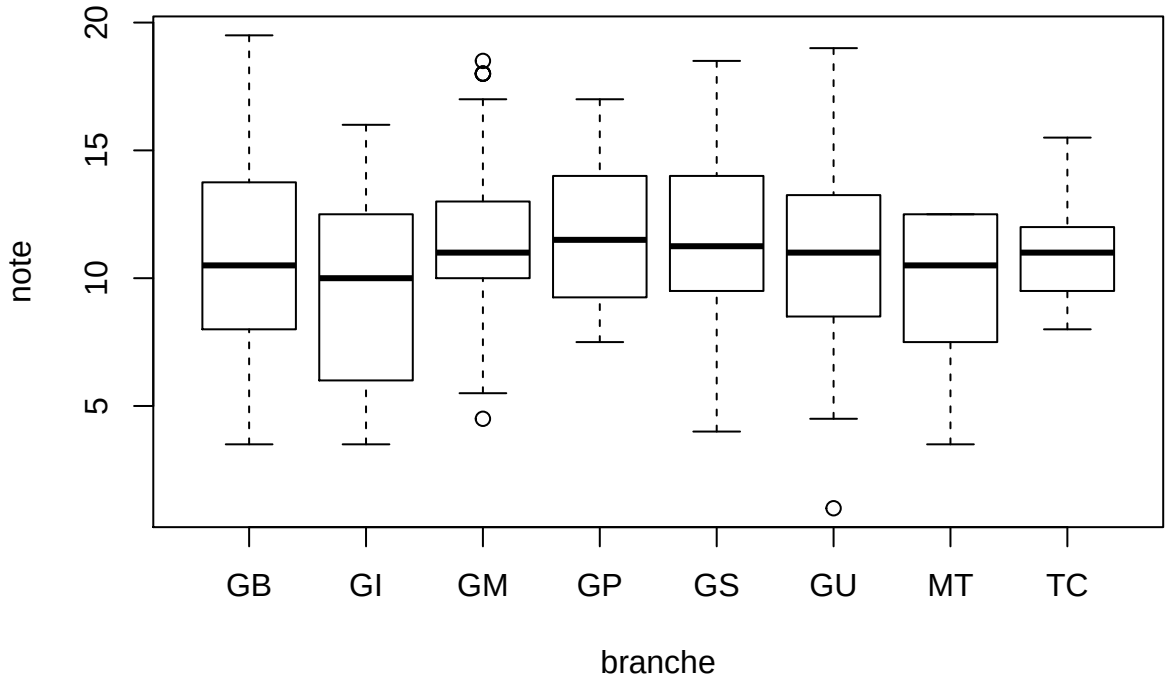
On peut utiliser notre fonction `hist.factor` pour visualiser les résultats de notes en fonction de la branche.

```
hist.factor(notes$note,notes$branche)
```



On peut aussi `boxploter` tout simplement:

```
plot(notes)
```



Aussi, il est intéressant d'effectuer un test d'indépendance du χ^2 pour voir s'il y a une différence significative inter-branche. Pour cela on crée déjà une table de contingence et on effectue le dît test:

```
# Table de contingence
tbl = table(notes$note, notes$branche)

# Test du chi-deux
chisq.test(tbl)
```

```
## Warning in chisq.test(tbl): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 213.37, df = 224, p-value = 0.6839
```

De manière identique, on pourrait se demander s'il existe une indépendance entre les étudiants en Génie Informatique et les étudiants en Génie des Procédes.

On peut *projeter* les données sur ces classes et effectuer un test de la même trempe.

```
filterGI = notes$branche == "GI"
filterGP = notes$branche == "GP"
filter = filterGI | filterGP
projOnGIetGP = notes[filter,]

# On enlève les niveaux qui ne sont pas présents ici:
projOnGIetGP$branche = factor(projOnGIetGP$branche)
```

```r
# Test du chi-deux
tbl = table(projOnGIetGP$note, projOnGIetGP$branche)
chisq.test(tbl)
```

```
## Warning in chisq.test(tbl): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 30.309, df = 22, p-value = 0.1112
```

3 Analyse des données `babies`

Analysons maintenant un autre jeu de données, `babies23.data` constitué de 1236 bébés décrits par 8 variables ici.

```r
babies = read.table("donnees/babies23.txt",header=T)
babies = babies[c(7,5,8,10,12,13,21,11)]
names(babies) = c("bwt","gestation","parity","age","height","weight","smoke","education")
```

On remplace les codes des données non disponibles par `NA` (“*not available*”) :

```r
babies[babies$bwt == 999, 1] <- NA
babies[babies$gestation == 999, 2] <- NA
babies[babies$age == 99, 4] <- NA
babies[babies$height == 99, 5] <- NA
babies[babies$weight == 999, 6] <- NA
babies[babies$smoke == 9, 7] <- NA
babies[babies$education == 9, 8] <- NA
```

Enfin, on déclare les variables qualitatives comme facteurs :

```r
babies$smoke<-factor(c("NonSmoking","Smoking","NonSmoking","NonSmoking")[babies$smoke+1])
babies$education<-factor(babies$education,ordered=T)
```

Les variables disponibles deviennent alors :

1. `bwt` : le poids de naissance (birth weight) en onces,
2. `gestation` : la durée de la gestation en jours,
3. `parity` : le nombre de grossesses précédentes,
4. `age` : l'âge de la mère à la fin de la grossesse,
5. `height` : la taille de la mère en pouces,
6. `weight` : le poids de la mère en livres,
7. `smoke` : la mère a-t-elle fumé pendant la grossesse ?,
8. `ed` : le niveau d'éducation de la mère :
   — 0 : less than 8th grade ;
   — 1 : 8th to 12th grade - did not graduate ;
   — 2 : High School graduate, no other schooling
   — 3 : High School + trade ;
   — 4 : High School + some college ;
   — 5 : College graduate ;
   — 6 and 7 : Trade school;
   — HS unclear.

3.1 Analyse exploratoire des données

On peut tout d'abord affichrun extrait et un résumé pour se faire une meilleure idée:

```
head(babies)
```

```
##   bwt gestation parity age height weight      smoke education
## 1 120       284      1  27     62    100 NonSmoking         5
## 2 113       282      2  33     64    135 NonSmoking         5
## 3 128       279      1  28     64    115    Smoking         2
## 4 123        NA      2  36     69    190 NonSmoking         5
## 5 108       282      1  23     67    125    Smoking         5
## 6 136       286      4  25     62     93 NonSmoking         2
```

`bwt`, `gestation`, `age`, `parity`, `height` et `weight` sont des variables quantitatives. `education` est une variable qualitative ordinal et `smoke` est une variable binaire.

```
summary(babies)
```

```
##       bwt          gestation         parity            age
##  Min.   : 55.0   Min.   :148.0   Min.   : 0.000   Min.   :15.00
##  1st Qu.:108.8   1st Qu.:272.0   1st Qu.: 0.000   1st Qu.:23.00
##  Median :120.0   Median :280.0   Median : 1.000   Median :26.00
##  Mean   :119.6   Mean   :279.3   Mean   : 1.932   Mean   :27.26
##  3rd Qu.:131.0   3rd Qu.:288.0   3rd Qu.: 3.000   3rd Qu.:31.00
##  Max.   :176.0   Max.   :353.0   Max.   :13.000   Max.   :45.00
##                  NA's   :13                       NA's   :2
##      height          weight             smoke       education
##  Min.   :53.00   Min.   : 87.0   NonSmoking:742   2      :444
##  1st Qu.:62.00   1st Qu.:114.8   Smoking   :484   4      :298
##  Median :64.00   Median :125.0   NA's      : 10   5      :219
##  Mean   :64.05   Mean   :128.6                    1      :183
##  3rd Qu.:66.00   3rd Qu.:139.0                    3      : 65
##  Max.   :72.00   Max.   :250.0                    (Other): 26
##  NA's   :22      NA's   :36                       NA's   :  1
```

Concentrons-nous sur les données manquantes:

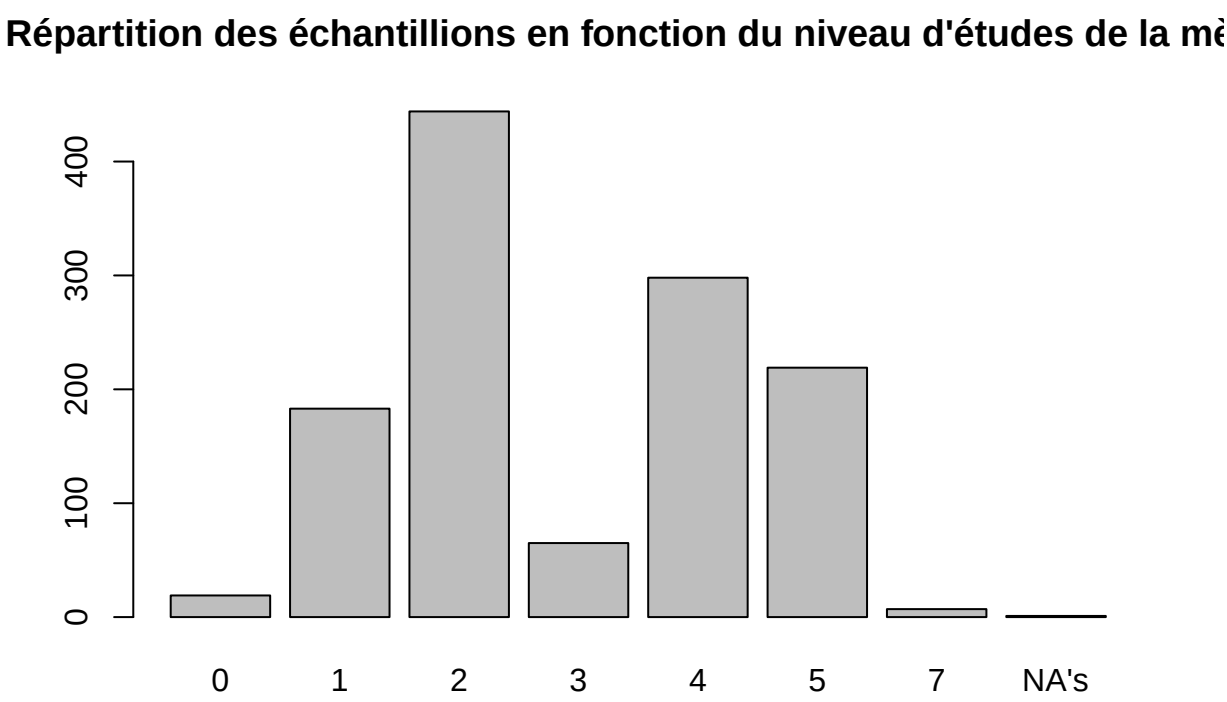
```
naCount <-sapply(babies, function(y) sum(length(which(is.na(y)))) / 1236 * 100)
data.frame(naCount)
```

```
##              naCount
## bwt       0.00000000
## gestation 1.05177994
## parity    0.00000000
## age       0.16181230
## height    1.77993528
## weight    2.91262136
## smoke     0.80906149
## education 0.08090615
```

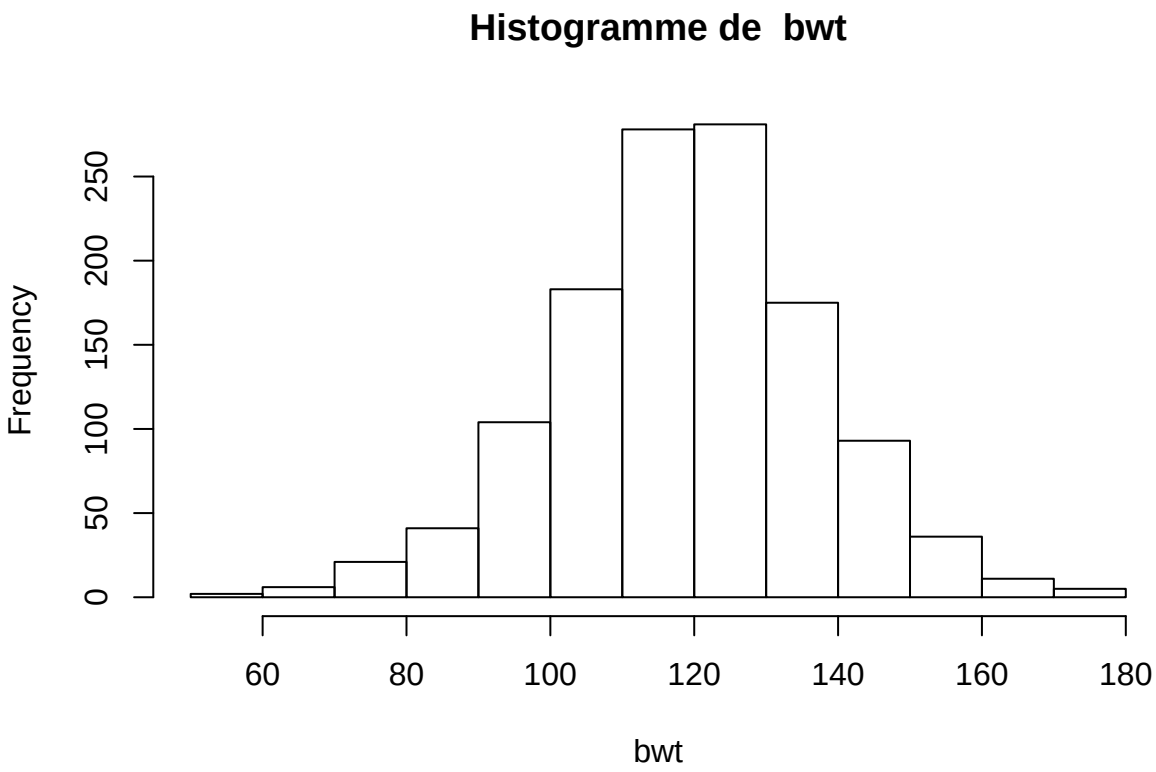
Il n'existe que peu de données non renseignées, dans le pire des cas un peu moins de 3% de valeurs sont manquantes (voir `weight`).

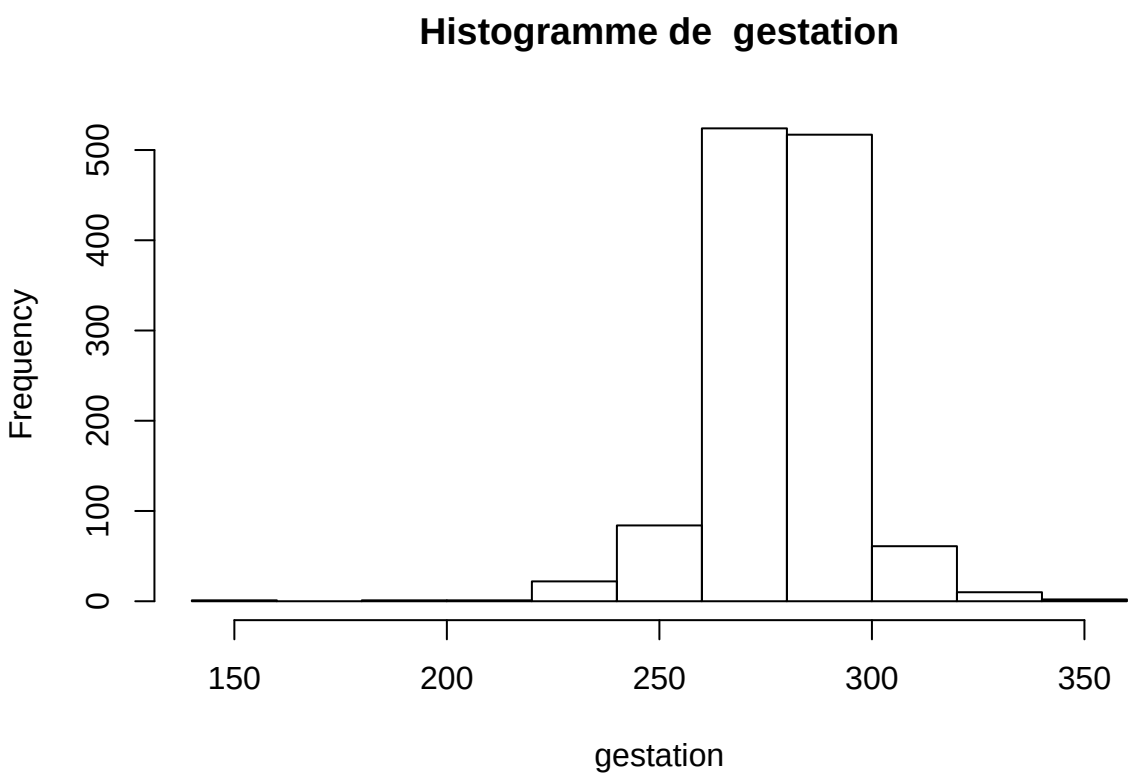
On peut essayer de regarder la répartition des échantillons en fonction

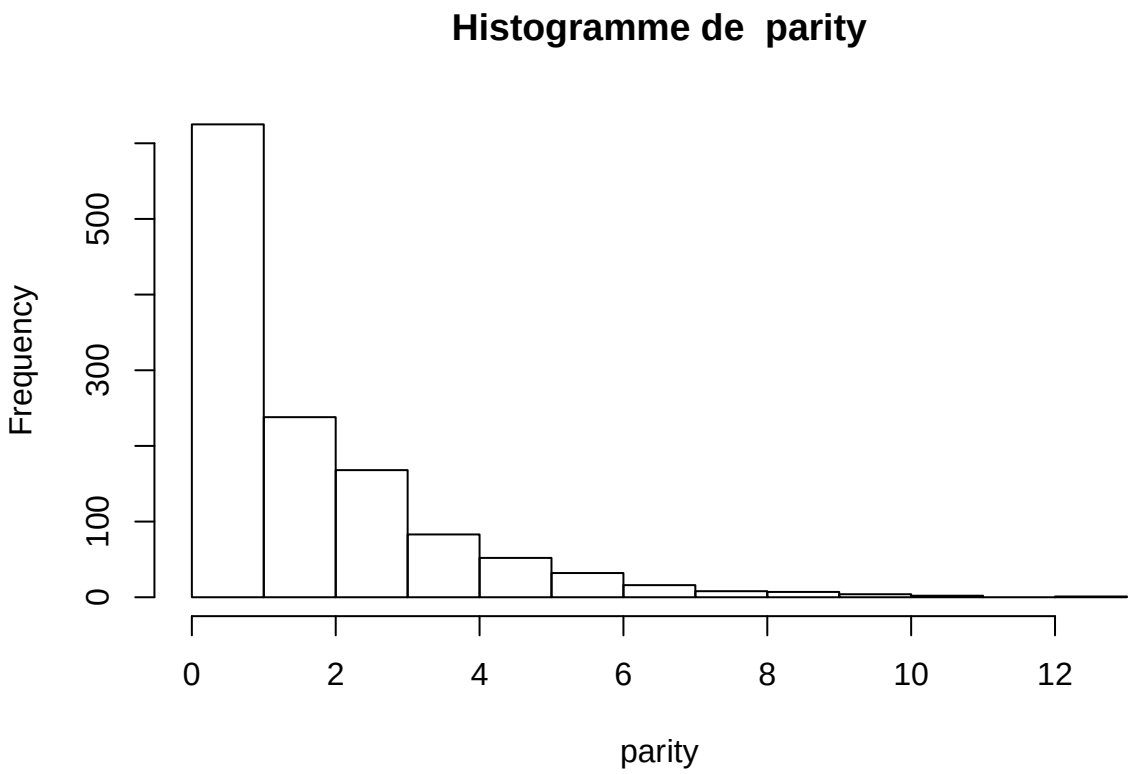
```
barplot(summary(babies$education),
        main = "Répartition des échantillions en fonction du niveau d'études de la mère")
```



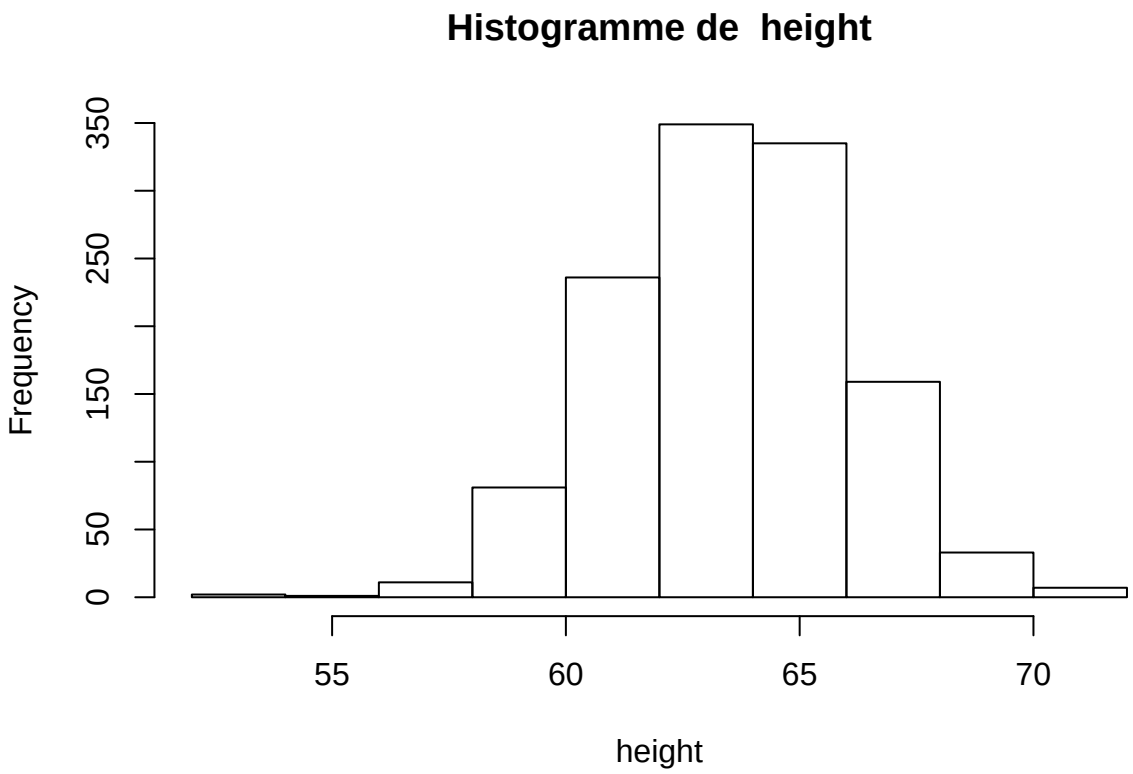

```
for(i in 1:6) {
  titre = colnames(babies)[i]
  hist(babies[,i], xlab=titre, main=paste("Histogramme de ",titre))
}
```


Histogramme de bwt
Frequency
0
50
100
150
200
250
60
80
100
120
140
160
180
bwt

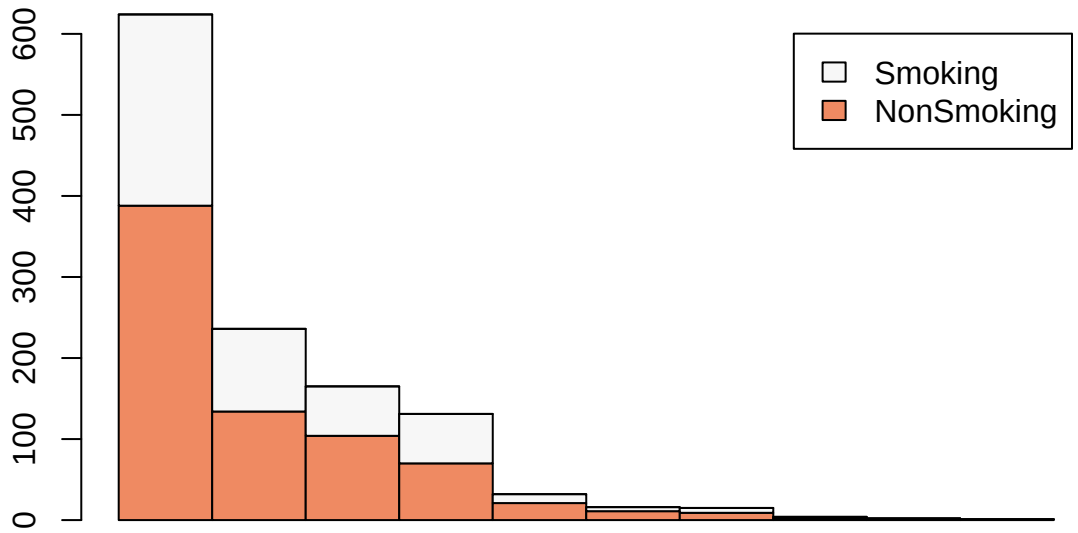

Histogramme de gestation
Frequency
0
100
200
300
400
500
150
200
250
300
350
gestation


Histogramme de parity
Frequency
0
100
300
500
0
2
4
6
8
10
12
parity


Histogramme de age
Frequency
0
50
100
150
15
20
25
30
35
40
45
age

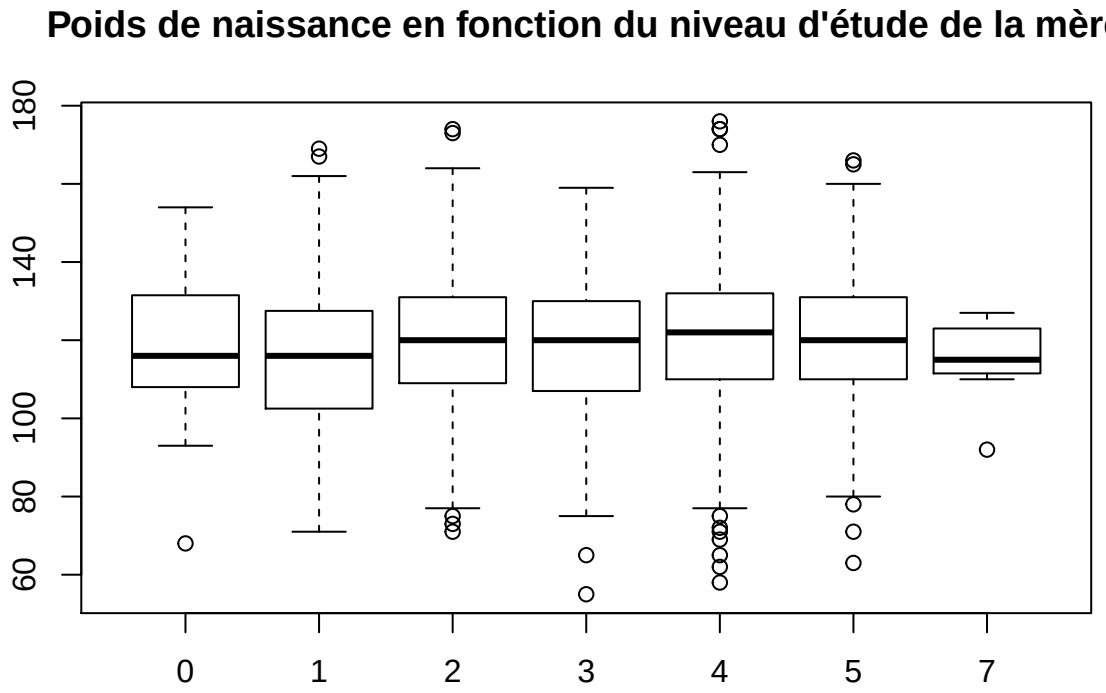

Histogramme de height
Frequency
0
50
150
250
350
55
60
65
70
height


```
hist.factor(babies$parity, babies$smoke)
```

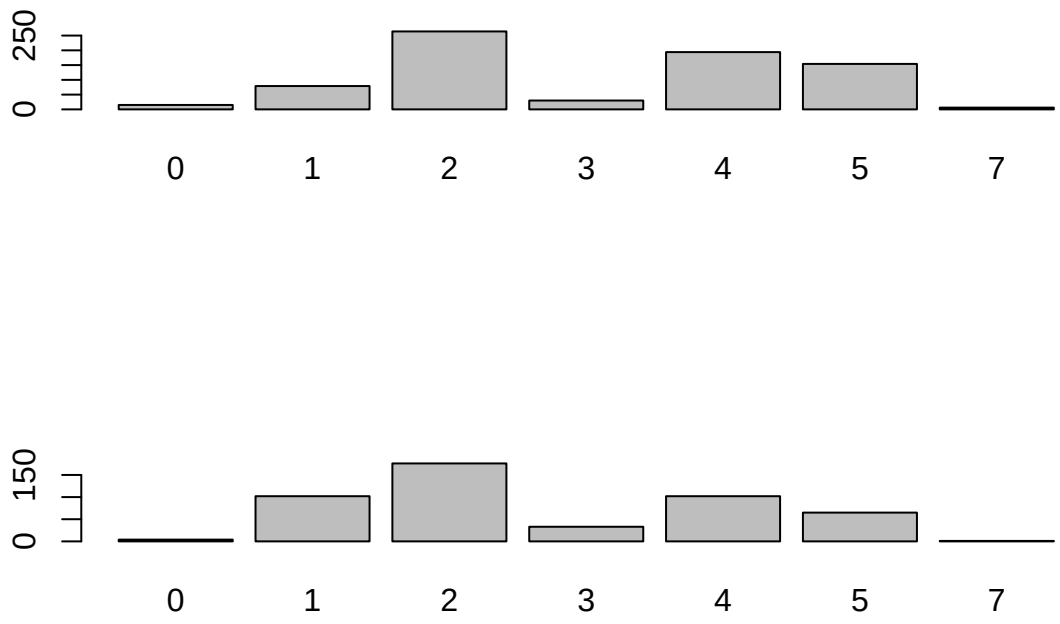


3.2 Lien tabagisme et niveau d'étude

```
boxplot(babies$bwt~babies$education,
        main="Poids de naissance en fonction du niveau d'étude de la mère")
```




```
def.par <- par(no.readonly=T)
par(mfrow=c(2,1))
plot(babies[babies$smoke == "NonSmoking",]$education)
plot(babies[babies$smoke == "Smoking",]$education)
```




```
par(def.par)
```

```
tbl = table(babies$smoke,babies$education)

chisq.test(tbl)
```

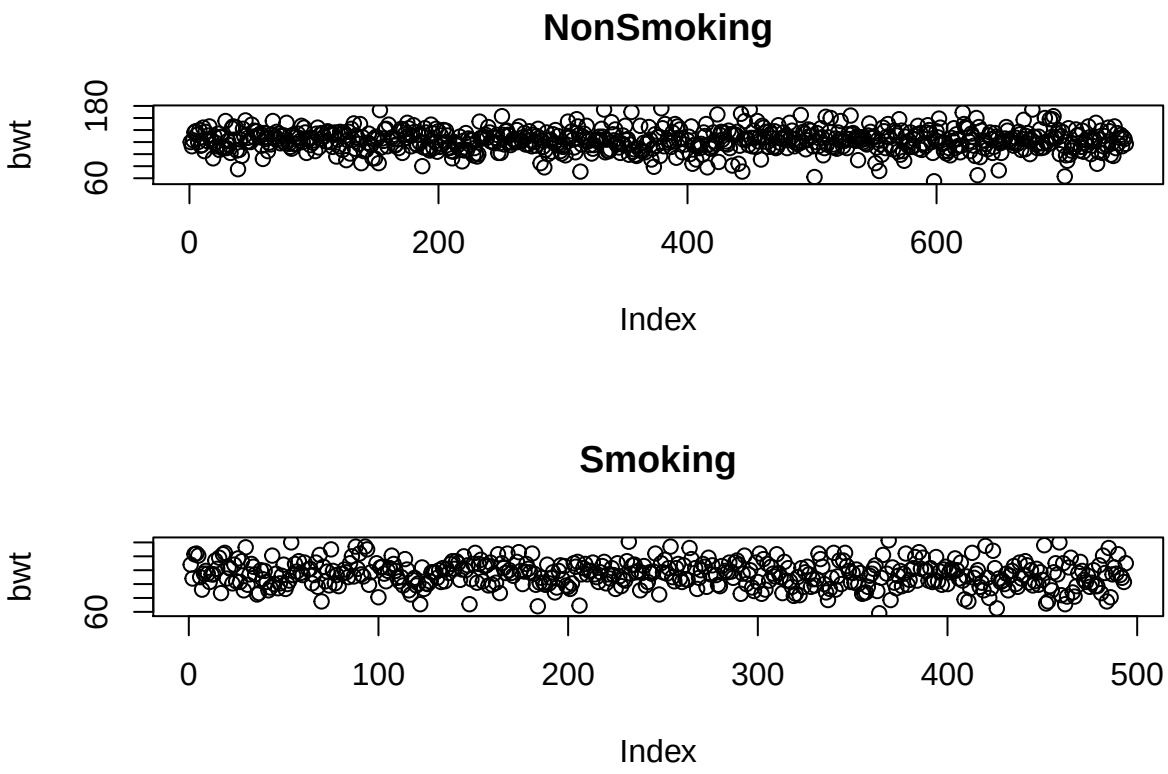
```
## Warning in chisq.test(tbl): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 42.509, df = 6, p-value = 1.459e-07
```

3.3 Lien tabagisme et poids du nouveau né

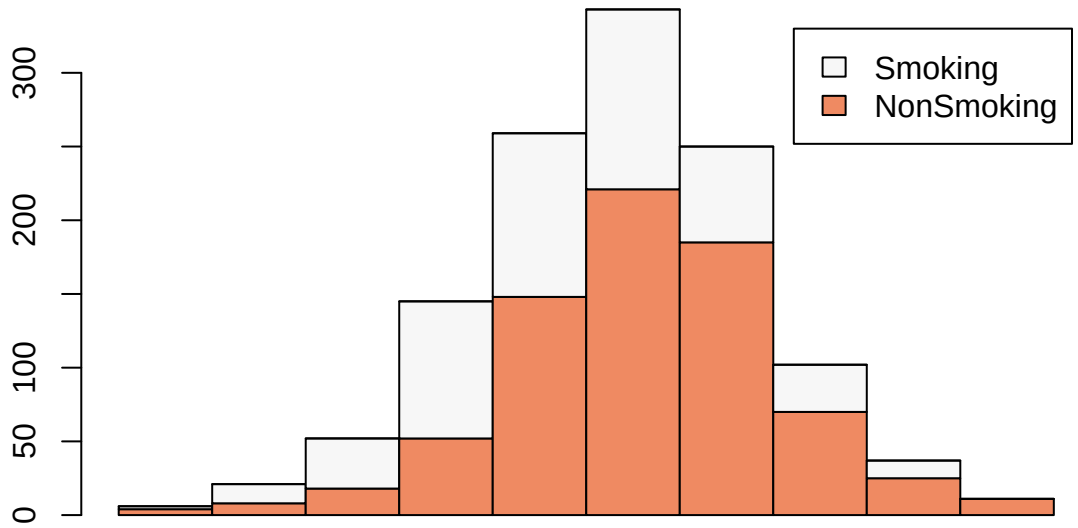
Explorons les types de graphes de dispersions:

```
def.par <- par(no.readonly=T)
par(mfrow=c(2,1))
plot(babies[babies$smoke == "NonSmoking",]$bwt, main = "NonSmoking",ylab = "bwt")
plot(babies[babies$smoke == "Smoking",]$bwt, main = "Smoking",ylab = "bwt")
```

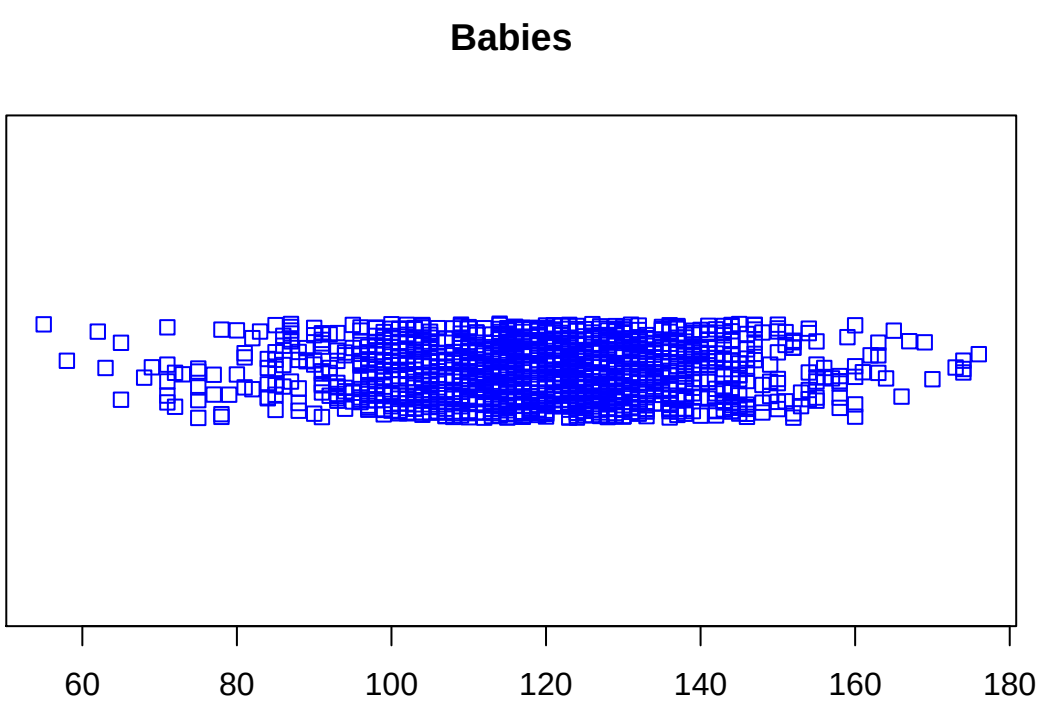



```
par(def.par)
```

```
hist.factor(babies$bwt,babies$smoke)
```




```r
stripchart(babies$bwt, main="Babies", col=c("blue","green")[babies$smoke], method = "jitter" )
```



On peut se donner aussi le résumé pour les deux valeurs binaires:

```
summary(babies[babies$smoke == "NonSmoking",]$bwt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##      55     113     123     123     134     176      10
```

```
summary(babies[babies$smoke == "Smoking",]$bwt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    58.0   102.0   115.0   114.1   126.0   163.0      10
```

Un petit test du χ^2:

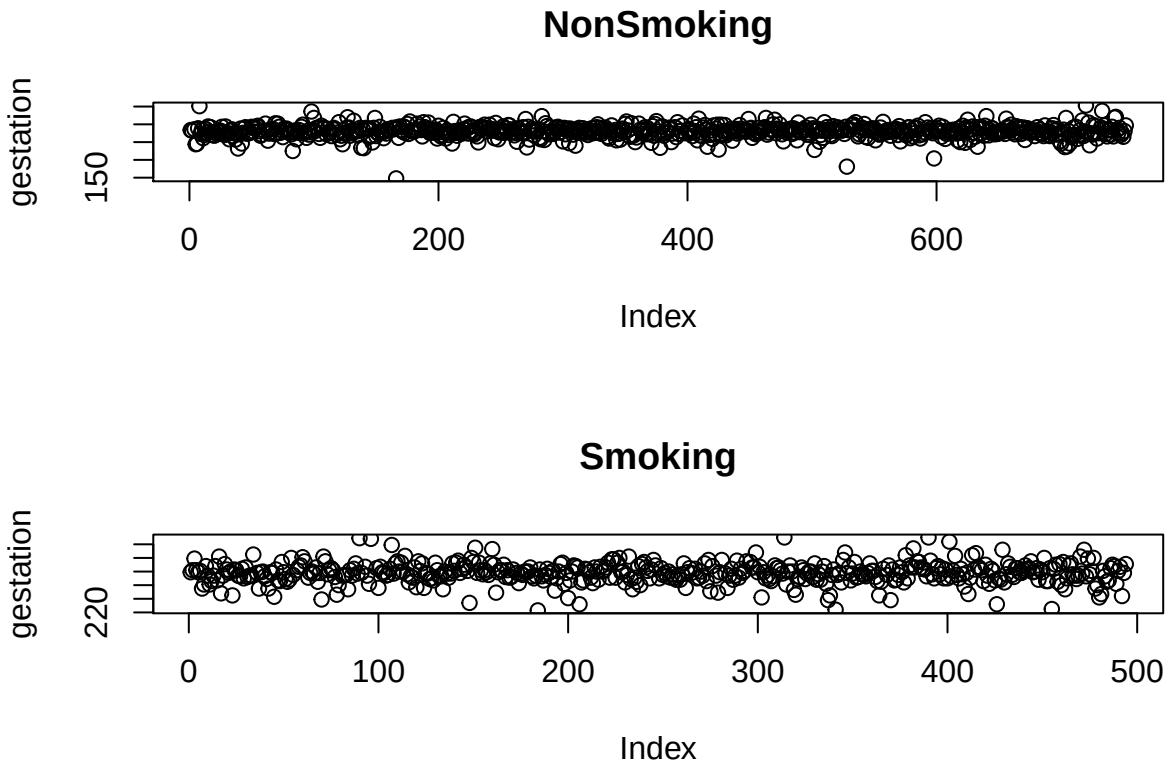
```
table = table(babies$smoke, babies$bwt)

chisq.test(table)
```

```
## Warning in chisq.test(table): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table
## X-squared = 170, df = 106, p-value = 7.959e-05
```

3.4 Lien tabagisme et temps de gestation

```
def.par <- par(no.readonly=T)
par(mfrow=c(2,1))
plot(babies[babies$smoke == "NonSmoking",]$gestation, main = "NonSmoking",ylab = "gestation")
plot(babies[babies$smoke == "Smoking",]$gestation, main = "Smoking",ylab = "gestation")
```




```
par(def.par)
```

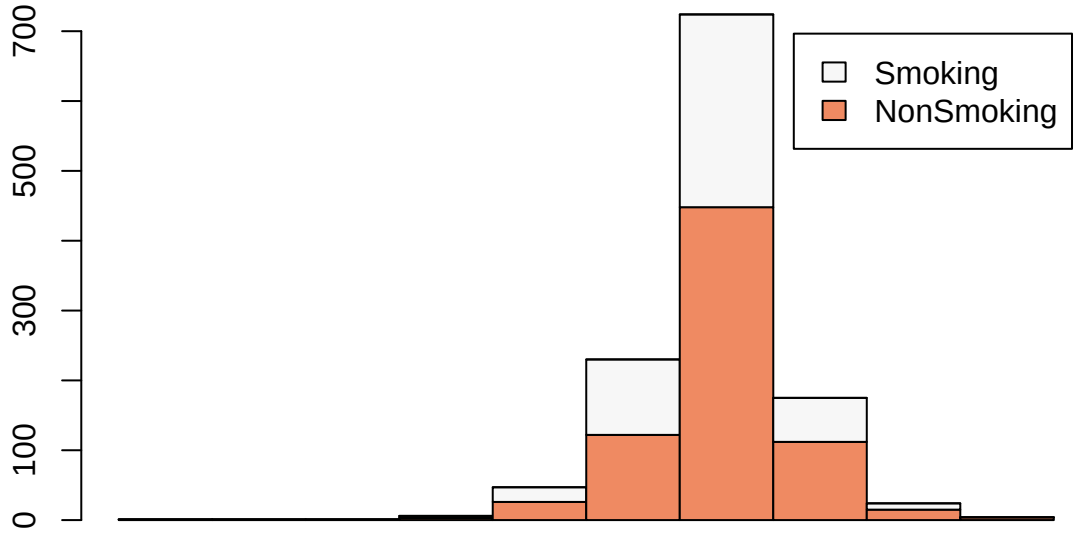
```
summary(babies[babies$smoke == "NonSmoking",]$gestation)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   148.0   273.0   281.0   280.2   289.0   353.0      19
```

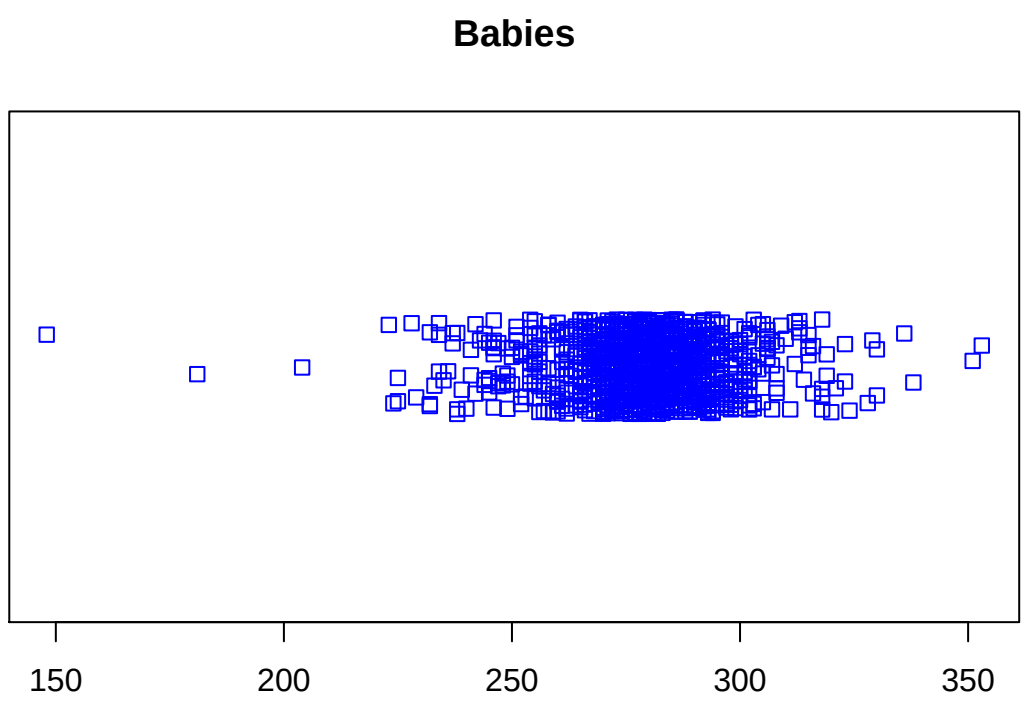
```
summary(babies[babies$smoke == "Smoking",]$gestation)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     223     271     279     278     286     330      14
```

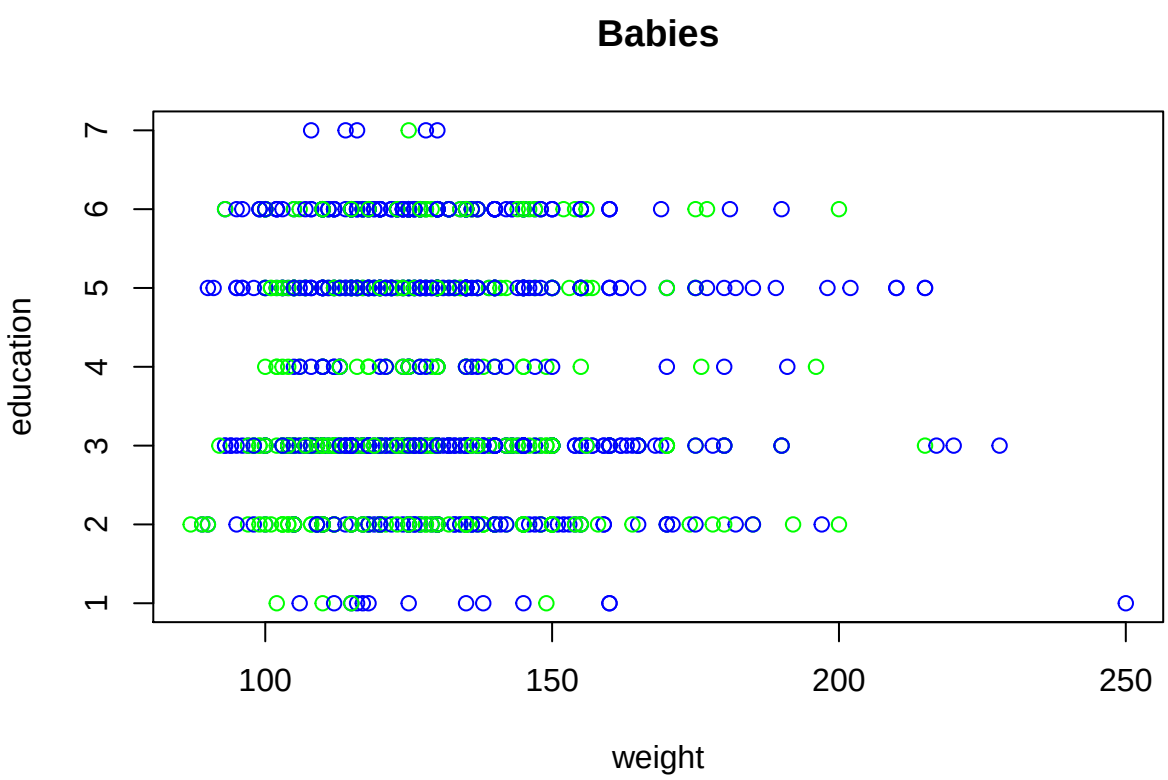
```
hist.factor(babies$gestation,babies$smoke)
```




```
stripchart(babies$gestation, main="Babies", col=c("blue","green")[babies$smoke], method = "jitter" )
```




```
plot(babies[,c(6,8)], main="Babies", col=c("blue","green")[babies$smoke])
```



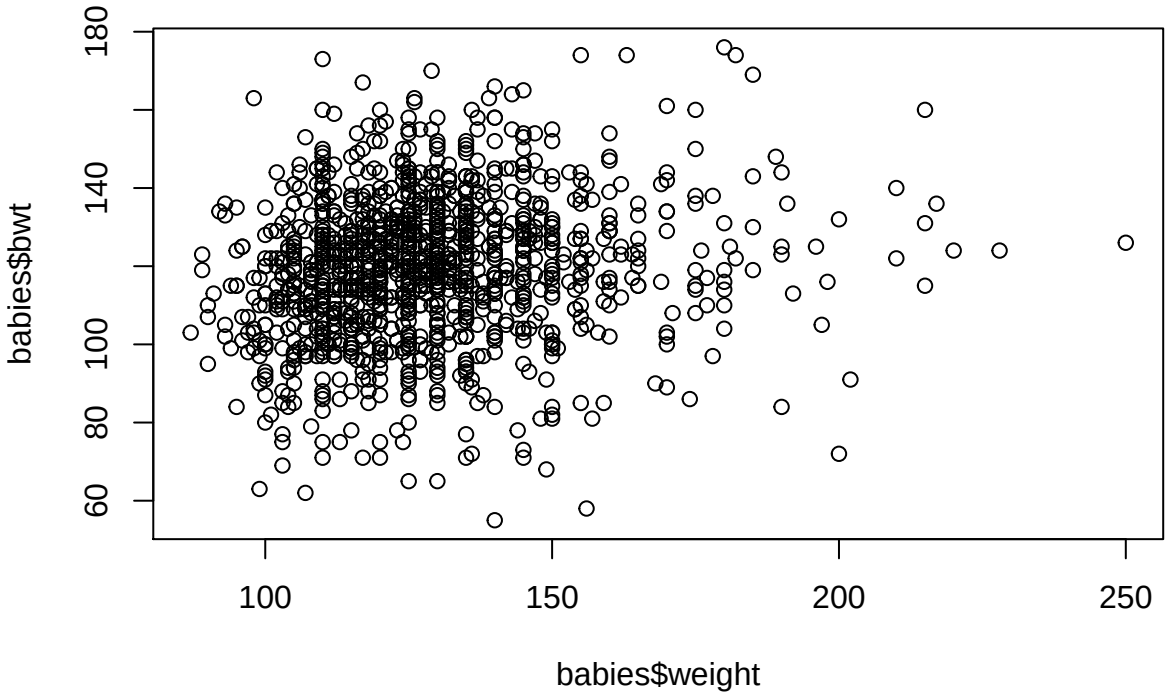

```
table = table(babies$smoke, babies$gestation)

chisq.test(table)
```

```
## Warning in chisq.test(table): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table
## X-squared = 103.12, df = 105, p-value = 0.5336
```

3.5 Lien poids de la mère et poids du nouveau né

```
plot(babies$bwt~babies$weight)
```




```
chisq.test(table(babies$bwt,babies$weight))
```

```
## Warning in chisq.test(table(babies$bwt, babies$weight)): Chi-squared
## approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(babies$bwt, babies$weight)
## X-squared = 11843, df = 11024, p-value = 3.591e-08
```