Devoir_2

EL_Hadrami

10/11/2020

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readxl)
library(gplots)
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

Exercice.13

```r
# Creation d'un data-frame "acteur"
Mort.à <- c(93,53,72,68,68,53)
Années.de.carrière <- c(66,25,48,37,31,32)
Nombre.de.films <- c(211,58,98,140,74,81)
Prénom <- c("Michel","André","Jean","Louis","Lino","Jacques")
Nom <- c("Galabru","Raimbourg","Gabin","De Funès","Ventura","Villeret")
Date.du.décès <- c("04-01-2016","23-09-1970","15-10-1976","27-01-1983","22-10-1987","28-01-2005")
data.acteur <- data.frame(Mort.à,Années.de.carrière,Nombre.de.films,Prénom,Nom,Date.du.décès)
 #utilisation d'un dplyer pour renommer la premiere variable
data.acteur.r <- rename(data.acteur,"Age.du.décès"=Mort.à)
# extraction de la colonne Prénom
prenom.extract <- data.acteur$Prénom
data.acteur.arrange <- arrange(data.acteur.r,Age.du.décès)
```

Exercice.14

Question 1.

```r
w <-read.delim(file="data/fromages1-TP-M1.txt")
```

Question 2.

```r
w <- rename(w,"mean.score"=Y,"c.a.a"=X1,"c.h.s"=X2,"c.a.l"=X3)
w$X1
```

```
## NULL
```

Question 3.:Les caracteristiques de w:

```r
print(w)
```

```
##    mean.score c.a.a  c.h.s c.a.l
## 1        12.3 4.543  3.135  0.86
## 2        20.9 5.159  5.043  1.53
## 3        39.0 5.366  5.438  1.57
## 4        47.9 5.759  7.496  1.81
## 5         5.6 4.663  3.807  0.99
## 6        25.9 5.697  7.601  1.09
## 7        37.3 5.892  8.726  1.29
## 8        21.9 6.078  7.966  1.78
## 9        18.1 4.898  3.850  1.29
## 10       21.0 5.242  4.174  1.58
## 11       34.9 5.740  6.142  1.68
## 12       57.2 6.446  7.908  1.90
## 13        0.7 4.477  2.996  1.06
## 14       25.9 5.236  4.942  1.30
## 15       54.9 6.151  6.752  1.52
## 16       40.9 6.365  9.588  1.74
## 17       15.9 4.787  3.912  1.16
## 18        6.4 5.412  4.700  1.49
## 19       18.0 5.247  6.174  1.63
## 20       38.9 5.438  9.064  1.99
## 21       14.0 4.564  4.949  1.15
## 22       15.2 5.298  5.220  1.33
## 23       32.0 5.455  9.242  1.44
## 24       56.7 5.855 10.199  2.01
## 25       16.8 5.366  3.664  1.31
```

```
## 26       11.6 6.043  3.219  1.46
## 27       26.5 6.458  6.962  1.72
## 28        0.7 5.328  3.912  1.25
## 29       13.4 5.802  6.685  1.08
## 30        5.5 6.176  4.787  1.25
```

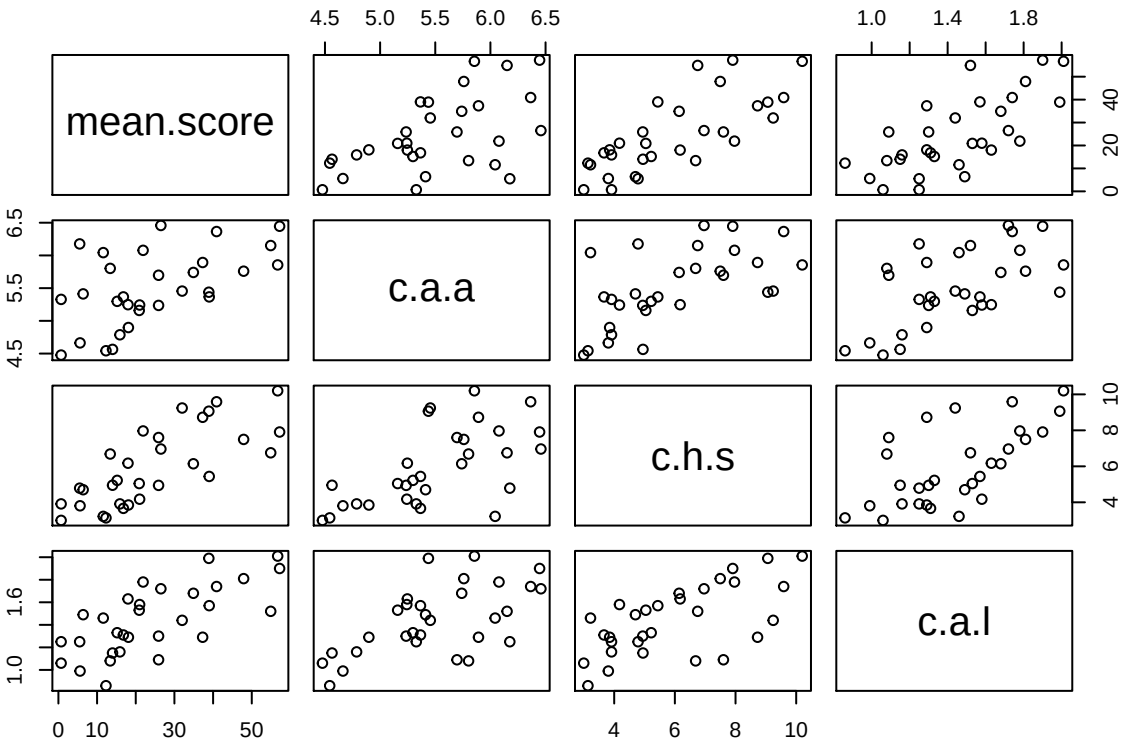
Question 4:Les parametres statistiques des variables

```
summary(w)
```

```
##    mean.score        c.a.a           c.h.s            c.a.l
##  Min.   : 0.70   Min.   :4.477   Min.   : 2.996   Min.   :0.860
##  1st Qu.:13.55   1st Qu.:5.237   1st Qu.: 3.978   1st Qu.:1.250
##  Median :20.95   Median :5.425   Median : 5.329   Median :1.450
##  Mean   :24.53   Mean   :5.498   Mean   : 5.942   Mean   :1.442
##  3rd Qu.:36.70   3rd Qu.:5.883   3rd Qu.: 7.575   3rd Qu.:1.667
##  Max.   :57.20   Max.   :6.458   Max.   :10.199   Max.   :2.010
```

Question 5.:

```
pairs(w)
```



la commande pairs permet de tracer une nuage de point pour chaque variable afin de voir les differentes correlations qui peuvent exister

Question 6.:Construction d'une nouvelle data frame

```
nv.c.a.a <- c(w$c.a.a[w$c.a.a > 5.1],rep(NA,6))
nv.c.a.l <- c(w$c.a.l[w$c.a.l < 1.77],rep(NA,5))
ww <- data.frame(w$mean.score,nv.c.a.a,w$c.h.s,nv.c.a.l)
ww <- rename(ww,"mean.score"=w.mean.score,"c.a.a"=nv.c.a.a,"c.h.s"=w.c.h.s,"c.a.l"=nv.c.a.l)
```

Question 7.: Les caracteristiques de ww

```
print(ww)
```

```
##    mean.score c.a.a  c.h.s c.a.l
## 1        12.3 5.159  3.135  0.86
## 2        20.9 5.366  5.043  1.53
## 3        39.0 5.759  5.438  1.57
## 4        47.9 5.697  7.496  0.99
## 5         5.6 5.892  3.807  1.09
## 6        25.9 6.078  7.601  1.29
## 7        37.3 5.242  8.726  1.29
## 8        21.9 5.740  7.966  1.58
## 9        18.1 6.446  3.850  1.68
## 10       21.0 5.236  4.174  1.06
## 11       34.9 6.151  6.142  1.30
## 12       57.2 6.365  7.908  1.52
## 13        0.7 5.412  2.996  1.74
## 14       25.9 5.247  4.942  1.16
## 15       54.9 5.438  6.752  1.49
## 16       40.9 5.298  9.588  1.63
## 17       15.9 5.455  3.912  1.15
## 18        6.4 5.855  4.700  1.33
## 19       18.0 5.366  6.174  1.44
## 20       38.9 6.043  9.064  1.31
## 21       14.0 6.458  4.949  1.46
## 22       15.2 5.328  5.220  1.72
## 23       32.0 5.802  9.242  1.25
## 24       56.7 6.176 10.199  1.08
## 25       16.8    NA  3.664  1.25
## 26       11.6    NA  3.219    NA
## 27       26.5    NA  6.962    NA
## 28        0.7    NA  3.912    NA
## 29       13.4    NA  6.685    NA
## 30        5.5    NA  4.787    NA
```

Question 8.: Les parametres statistiques de la variable ww

```
summary(ww)
```

```
##    mean.score        c.a.a           c.h.s           c.a.l
##  Min.   : 0.70   Min.   :5.159   Min.   : 2.996   Min.   :0.860
##  1st Qu.:13.55   1st Qu.:5.356   1st Qu.: 3.978   1st Qu.:1.160
##  Median :20.95   Median :5.718   Median : 5.329   Median :1.310
##  Mean   :24.53   Mean   :5.709   Mean   : 5.942   Mean   :1.351
##  3rd Qu.:36.70   3rd Qu.:6.052   3rd Qu.: 7.575   3rd Qu.:1.530
##  Max.   :57.20   Max.   :6.458   Max.   :10.199   Max.   :1.740
##                  NA's   :6                        NA's   :5
```

Exercice 15

1.

```
data(airquality) # charger les donées airquality
df.airquality <- data.frame(airquality)
```

2.Affichage des noms des variables

```r
names(df.airquality)
```

```
## [1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
```

3.nombre de ligne et de colonne

```r
nrow(df.airquality)
```

```
## [1] 153
```

```r
ncol(df.airquality)
```

```
## [1] 6
```

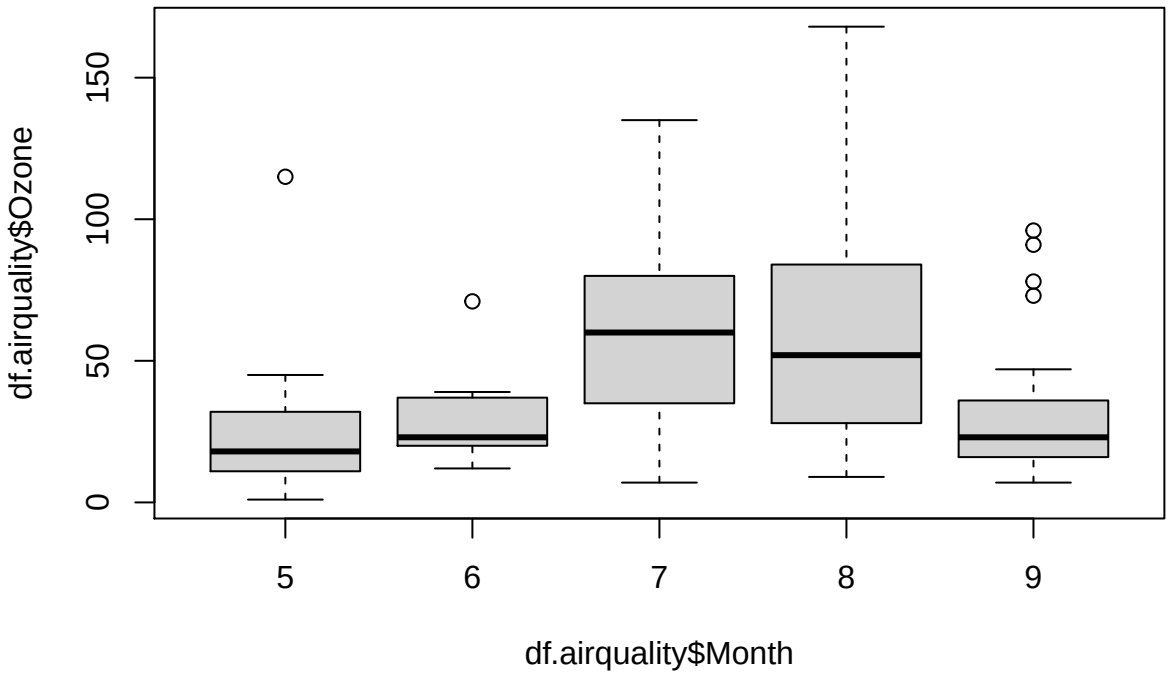
4.Les parametres statistiques

```r
summary(df.airquality)
```

```
##      Ozone           Solar.R           Wind             Temp
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00
##  Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88
##  3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00
##  NA's   :37       NA's   :7
##      Month            Day
##  Min.   :5.000   Min.   : 1.0
##  1st Qu.:6.000   1st Qu.: 8.0
##  Median :7.000   Median :16.0
##  Mean   :6.993   Mean   :15.8
##  3rd Qu.:8.000   3rd Qu.:23.0
##  Max.   :9.000   Max.   :31.0
##
```

5.representation de la boite a moustache

```r
boxplot(df.airquality$Ozone~df.airquality$Month)
```

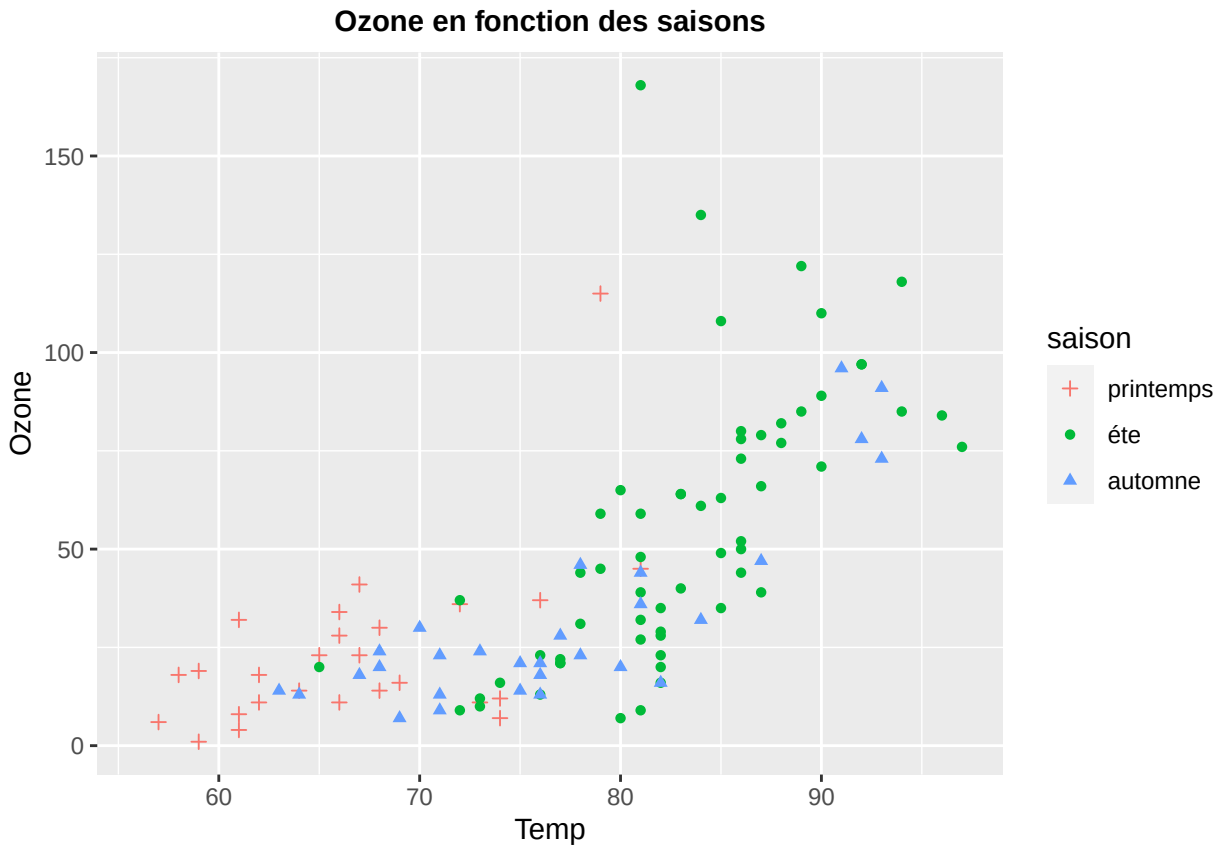


6.Creation d'une variable qualitative "saison"

```
saison <-factor(df.airquality$Month,levels=c(5:9))
levels(saison)[levels(saison)==5] <- "printemps"
levels(saison)[levels(saison)==6] <- "éte"
levels(saison)[levels(saison)==7] <- "éte"
levels(saison)[levels(saison)==8] <- "éte"
levels(saison)[levels(saison)==9] <- "automne"
df.airquality$season = saison
```

7.

```
g <- ggplot(data = df.airquality) +
  geom_point(mapping = aes(x =Temp, y = Ozone, shape = saison,color=saison)) +
  scale_shape_manual(values=c(3, 16, 17))+
  ggtitle("Ozone en fonction des saisons")
g + theme (plot.title = element_text(size=11,face="bold",hjust = 0.5))
```



Exercice 16

1.Simulation de 100 valeurs suivant une loi normale

```
n <- 100
e <- rnorm(n,0,25)
```

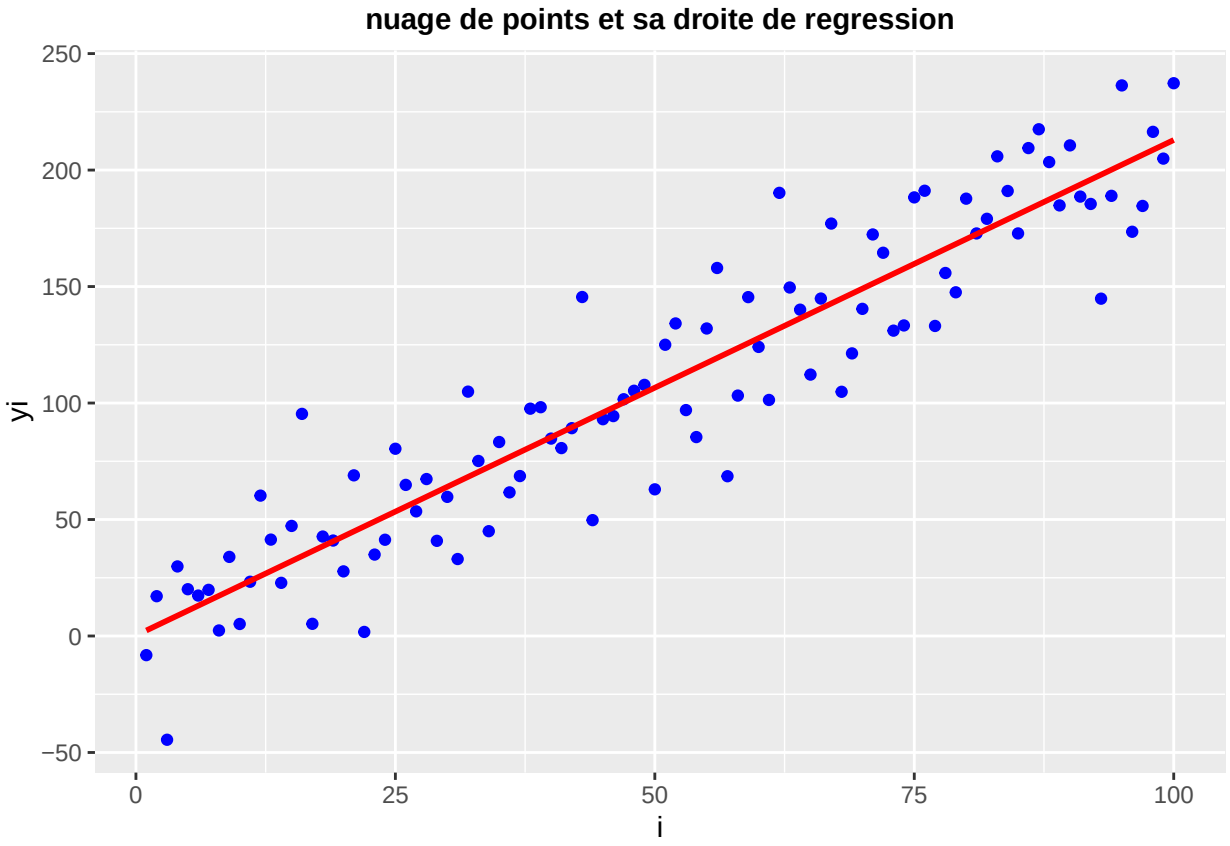
2. Pour tout $i \in 1, \ldots, 100$,on pose $y_i = 1.7 + 2.1i + e_i$

```
i <- c(1:100)
yi <- 1.7 + 2.1 * i + e[i]
```

2.a et 2.b: representation d'un nuage de point et une droite (i, y_i)

```
data.f <- data.frame(i,yi)
g <- ggplot(data = data.f) +
geom_point(mapping = aes(x = i, y = yi),colour="blue") +
geom_smooth(mapping = aes(x = i, y = yi),se = FALSE,colour="red",fill="red",method = lm) +
ggtitle("nuage de points et sa droite de regression")
g + theme (plot.title = element_text(size=11,face="bold",hjust = 0.5))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Exercice 17

```
# Creation d'une matrice
ligne1 <- c(68,119,26,7)
ligne2 <- c(15,54,14,10)
ligne3 <- c(5,29,14,16)
ligne4 <- c(20,84,17,94)
mat <- matrix(c(ligne1,ligne2,ligne3,ligne4),nrow = 4,ncol=4,byrow = T,dimnames = list(c("marron","nois
```

2. Calculer la matrice des fréquences (arrondit au 100ème près)

```
matfreq <- mat / sum(mat)
round(matfreq*100,2)
```

```
##           brun chatin roux blond
## marron   11.49  20.10 4.39  1.18
## noisette  2.53   9.12 2.36  1.69
## vert      0.84   4.90 2.36  2.70
## bleu      3.38  14.19 2.87 15.88
```

3. les lois marginales (nommer c pour le vecteur colonne et r pour le vecteur ligne)

```
l <- round(apply(matfreq,1,sum),2)
c <- round(apply(matfreq,2,sum),2)
```

4.Profils lignes

```
L <- round(sweep(mat,1,rowSums(mat),'/'),2)
```

5.Profils colonnes

```r
C <- round(sweep(mat,2,colSums(mat),'/'),2)
```

6. La distance de chi-deux entre les profils lignes

```r
d.chi <- 0
distancechideux <- function(L){
  for(i in 1:nrow(L)-1){

    d.chi <- d.chi + sum(((L[i,] - L[i+1,]) ^ 2) /c)
  }
  return (d.chi)
}
```

7. la matrice des taux de liaison

```r
t <- round((matfreq - (1%*%t(c))) / (1%*%t(c)),2)
```

Exercice 18

1.

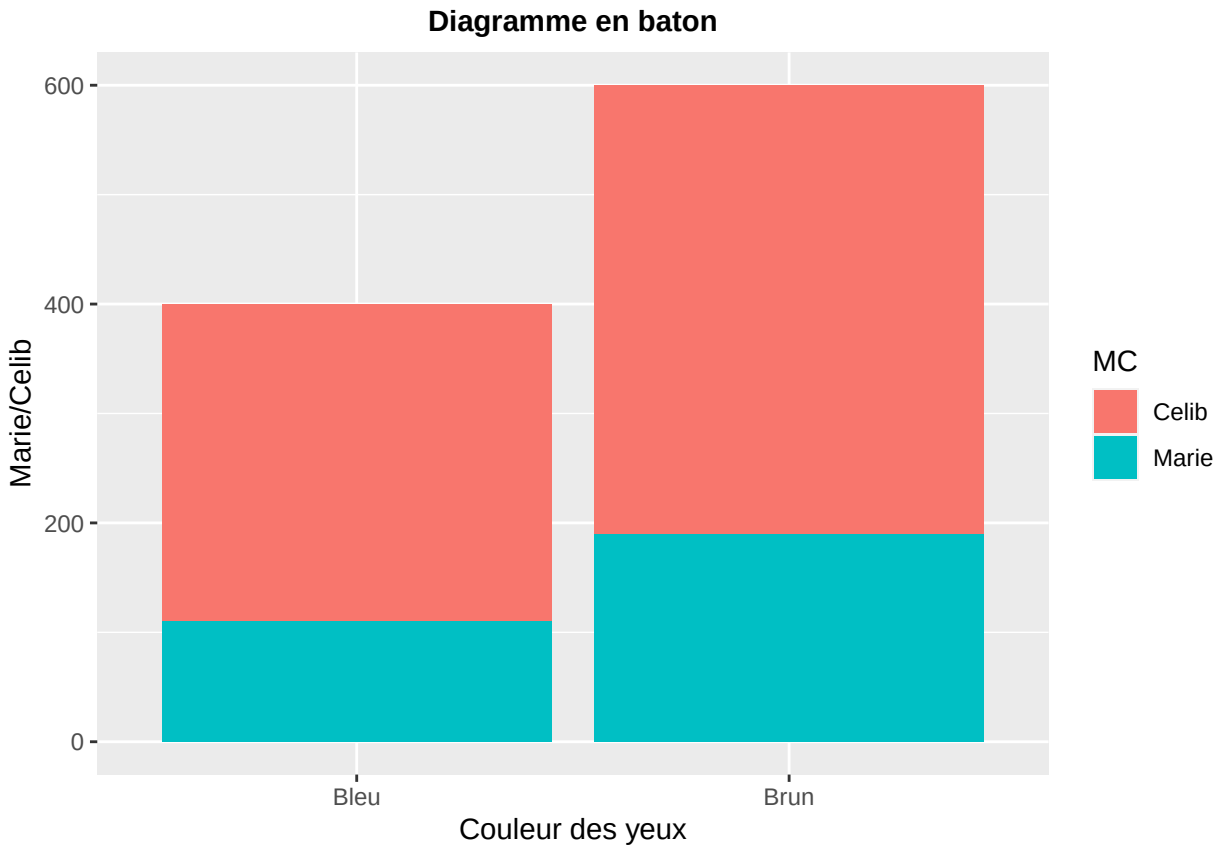
```r
tableau <- matrix(c(290,410,110,190), ncol=2, byrow=TRUE)
colnames(tableau) <- c("Bleu","Brun")
rownames(tableau) <- c("Celib","Marie")
tableau <- as.table(tableau)
```

2.

```r
tableau
```

```
##       Bleu Brun
## Celib  290  410
## Marie  110  190
```

```r
tabdf <-data.frame(tableau)
MC <- tabdf$Var1
g <- ggplot(tabdf) + geom_bar(stat="identity",mapping=aes(x = Var2,y = Freq,fill = MC)) +
  xlab("Couleur des yeux") + ylab("Marie/Celib") + ggtitle("Diagramme en baton")
g + theme (plot.title = element_text(size=11,face="bold",hjust = 0.5))
```



3.

```
n <- margin.table(tableau)
m1 <- margin.table(tableau,1)
m2 <- margin.table(tableau,2)
prop.table(tableau)
```

```
##       Bleu Brun
## Celib 0.29 0.41
## Marie 0.11 0.19
```

4. Le test du chi-deux

```
chisq.test(tableau)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tableau
## X-squared = 1.7907, df = 1, p-value = 0.1808
```

4.a

```
tab0 <- as.array(m1) %*% t(as.array(m2))/n
tab0 <- as.table(tab0)
```

4.b

5.b

```r
tab1 <- as.matrix(tableau)
tab1[2,1] <- tab1[1,1] + tab1[2,1]
tab1[1,1] <- 0
tab1[1,2] <- tab1[1,2] + tab1[2,2]
tab1[2,2] <- 0
tab1tab <- as.matrix(tab1)
chisq.test(tab1tab)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tab1tab
## X-squared = 995.84, df = 1, p-value < 2.2e-16
```

6. Test de khi-deux sur quelques echantillons de R

```r
data(HairEyeColor)
dfH <- as.data.frame(HairEyeColor)
tH <- xtabs(Freq~Hair+Eye,dfH)
chisq.test(tH)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tH
## X-squared = 138.29, df = 9, p-value < 2.2e-16
```

```r
data(Titanic)
dfT <- as.data.frame(Titanic)
dfTtab <- xtabs(Freq~Sex+Survived,dfT)
chisq.test(dfTtab)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  dfTtab
## X-squared = 454.5, df = 1, p-value < 2.2e-16
```

```r
data(UCBAdmissions)
dfU <- as.data.frame(UCBAdmissions)
dfUtab <- xtabs(Freq~Admit+Gender,dfU)
chisq.test(as.table(dfUtab))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  as.table(dfUtab)
## X-squared = 91.61, df = 1, p-value < 2.2e-16
```

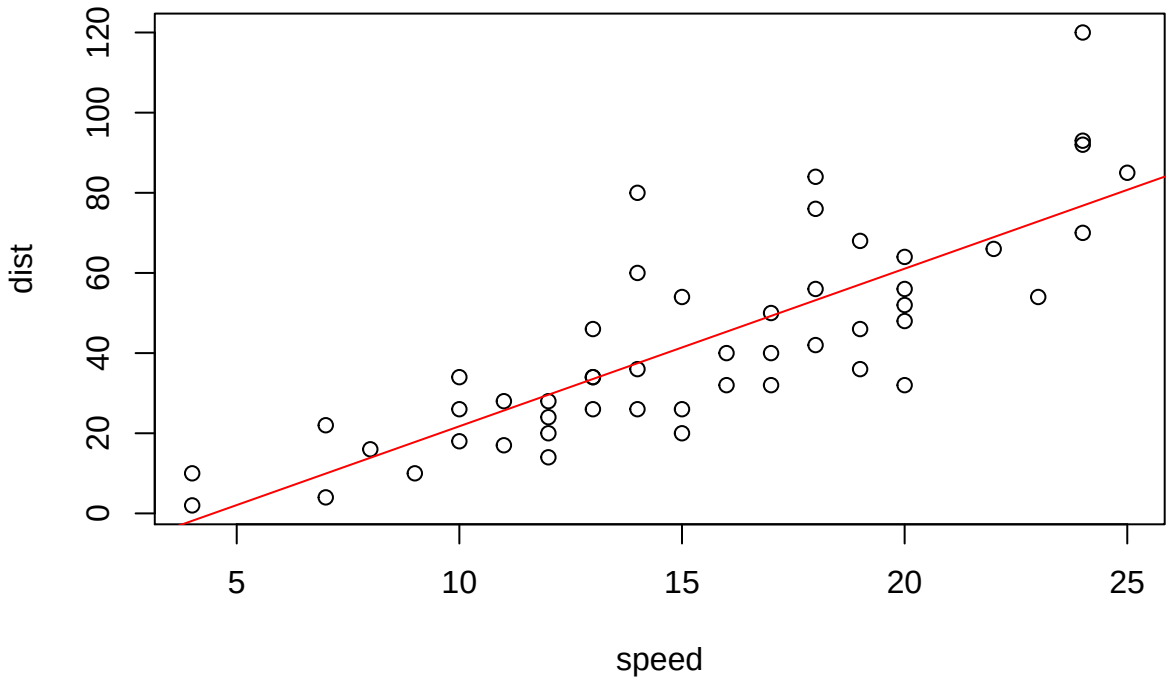
Exercice 19

```r
data(cars)
# regression lineaire
reg <- lm(dist~speed,cars)
summary(reg)
```

```
##
## Call:
```

```
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

```
plot(cars)
abline(reg,col="red")
```



19.a: La valeur prédite pour une vitesse de 20

```
predict(reg,newdata = data.frame(speed=20))
```

```
##        1
## 61.06908
```

Jeux de données cpus

```
data(cpus)
head(cpus)
```

```
##             name syct mmin  mmax cach chmin chmax perf estperf
## 1  ADVISOR 32/60  125  256  6000  256    16   128  198     199
## 2  AMDAHL 470V/7   29 8000 32000   32     8    32  269     253
```

```
## 3  AMDAHL 470/7A   29 8000 32000   32     8    32  220     253
## 4 AMDAHL 470V/7B   29 8000 32000   32     8    32  172     253
## 5 AMDAHL 470V/7C   29 8000 16000   32     8    16  132     132
## 6  AMDAHL 470V/8   26 8000 32000   64     8    32  318     290
```

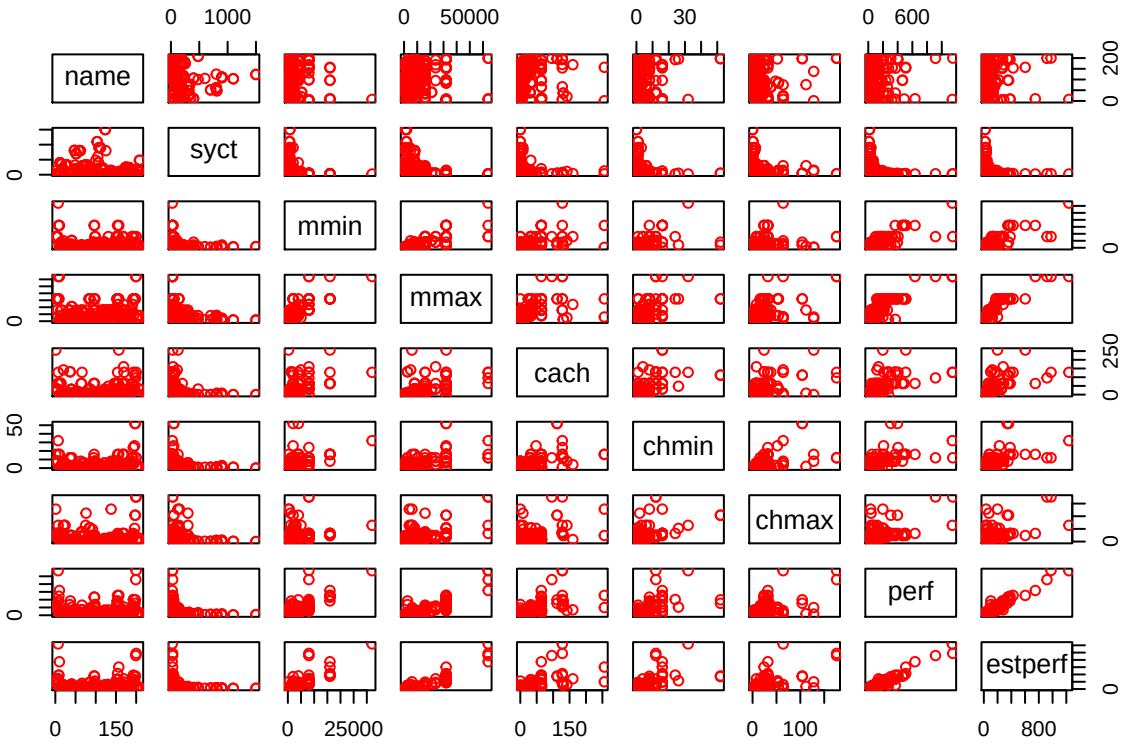
```
summary(cpus)
```

```
##              name          syct             mmin            mmax
##  ADVISOR 32/60 :  1   Min.   :  17.0   Min.   :   64   Min.   :   64
##  AMDAHL 470/7A :  1   1st Qu.:  50.0   1st Qu.:  768   1st Qu.: 4000
##  AMDAHL 470V/7 :  1   Median : 110.0   Median : 2000   Median : 8000
##  AMDAHL 470V/7B:  1   Mean   : 203.8   Mean   : 2868   Mean   :11796
##  AMDAHL 470V/7C:  1   3rd Qu.: 225.0   3rd Qu.: 4000   3rd Qu.:16000
##  AMDAHL 470V/8 :  1   Max.   :1500.0   Max.   :32000   Max.   :64000
##  (Other)       :203
##       cach            chmin            chmax             perf
##  Min.   :  0.00   Min.   : 0.000   Min.   :  0.00   Min.   :   6.0
##  1st Qu.:  0.00   1st Qu.: 1.000   1st Qu.:  5.00   1st Qu.:  27.0
##  Median :  8.00   Median : 2.000   Median :  8.00   Median :  50.0
##  Mean   : 25.21   Mean   : 4.699   Mean   : 18.27   Mean   : 105.6
##  3rd Qu.: 32.00   3rd Qu.: 6.000   3rd Qu.: 24.00   3rd Qu.: 113.0
##  Max.   :256.00   Max.   :52.000   Max.   :176.00   Max.   :1150.0
##
##     estperf
##  Min.   :  15.00
##  1st Qu.:  28.00
##  Median :  45.00
##  Mean   :  99.33
##  3rd Qu.: 101.00
##  Max.   :1238.00
##
```

comportement des variables les unes par rapport aux autres

On utilise la commande plot

```
plot(cpus,col="red")
```



Regression lineaire multiples

```
regm <- lm(perf~syct+mmin+mmax+cach+chmin+chmax+estperf,cpus)
names(regm)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
summary(regm)
```

```
##
## Call:
## lm(formula = perf ~ syct + mmin + mmax + cach + chmin + chmax +
##     estperf, data = cpus)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -160.572  -15.224   -2.224    7.556  234.589
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.9069391  6.7801517   1.019   0.3096
## syct        -0.0134520  0.0125081  -1.075   0.2835
## mmin         0.0017772  0.0015114   1.176   0.2410
## mmax        -0.0006548  0.0005910  -1.108   0.2692
## cach         0.1740674  0.0990531   1.757   0.0804 .
## chmin       -0.1072525  0.5786821  -0.185   0.8531
## chmax        0.3479115  0.1657820   2.099   0.0371 *
## estperf      0.9447315  0.0608743  15.519   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 40.56 on 201 degrees of freedom
## Multiple R-squared:  0.9385, Adjusted R-squared:  0.9364
## F-statistic: 438.4 on 7 and 201 DF,  p-value: < 2.2e-16
```

Apres avoir fait le summary on remarque que les variables les pertinentes sont ceux qui sont les plus significatives au seuil $\alpha = 0.05$ donc ces variables sont :

- estperf
- chmax
- cach

Les commandes stepAIC,addterm et dropterm peuvent etre utiliser aussi pour trouver les variables pertinentes.

```
stepAIC(regm)
```

```
## Start:  AIC=1555.64
## perf ~ syct + mmin + mmax + cach + chmin + chmax + estperf
##
##           Df Sum of Sq    RSS    AIC
## - chmin    1        57 330773 1553.7
## - syct     1      1903 332619 1554.8
## - mmax     1      2020 332736 1554.9
## - mmin     1      2275 332991 1555.1
## <none>                 330716 1555.6
## - cach     1      5081 335797 1556.8
## - chmax    1      7246 337963 1558.2
## - estperf  1    396286 727002 1718.3
##
## Step:  AIC=1553.67
## perf ~ syct + mmin + mmax + cach + chmax + estperf
##
##           Df Sum of Sq    RSS    AIC
## - syct     1      1881 332654 1552.9
## - mmax     1      2057 332830 1553.0
## - mmin     1      2219 332992 1553.1
## <none>                 330773 1553.7
## - cach     1      5147 335920 1554.9
## - chmax    1      7545 338318 1556.4
## - estperf  1    396587 727360 1716.4
##
## Step:  AIC=1552.86
## perf ~ mmin + mmax + cach + chmax + estperf
##
##           Df Sum of Sq    RSS    AIC
## - mmax     1      1131 333785 1551.6
## <none>                 332654 1552.9
## - mmin     1      3456 336110 1553.0
## - cach     1      6747 339400 1555.0
## - chmax    1      9281 341935 1556.6
## - estperf  1    423060 755713 1722.3
##
## Step:  AIC=1551.57
## perf ~ mmin + cach + chmax + estperf
##
##           Df Sum of Sq     RSS    AIC
```

```
## - mmin     1      3115  336900 1551.5
## <none>                  333785 1551.6
## - cach     1      7771  341555 1554.4
## - chmax    1      8975  342760 1555.1
## - estperf  1    674856 1008640 1780.7
##
## Step:  AIC=1551.51
## perf ~ cach + chmax + estperf
##
##           Df Sum of Sq     RSS    AIC
## <none>                  336900 1551.5
## - chmax    1      5998  342897 1553.2
## - cach     1      8884  345783 1555.0
## - estperf  1   2119750 2456650 1964.7
```

```
##
## Call:
## lm(formula = perf ~ cach + chmax + estperf, data = cpus)
##
## Coefficients:
## (Intercept)         cach        chmax      estperf
##      1.8910       0.2144       0.2601       0.9420
```

Exercice 20

declaration des données

```
x <- c(3,6,9,12,15,18,21,24)
y <- c(20,50,40,70,40,60,50,80)
```

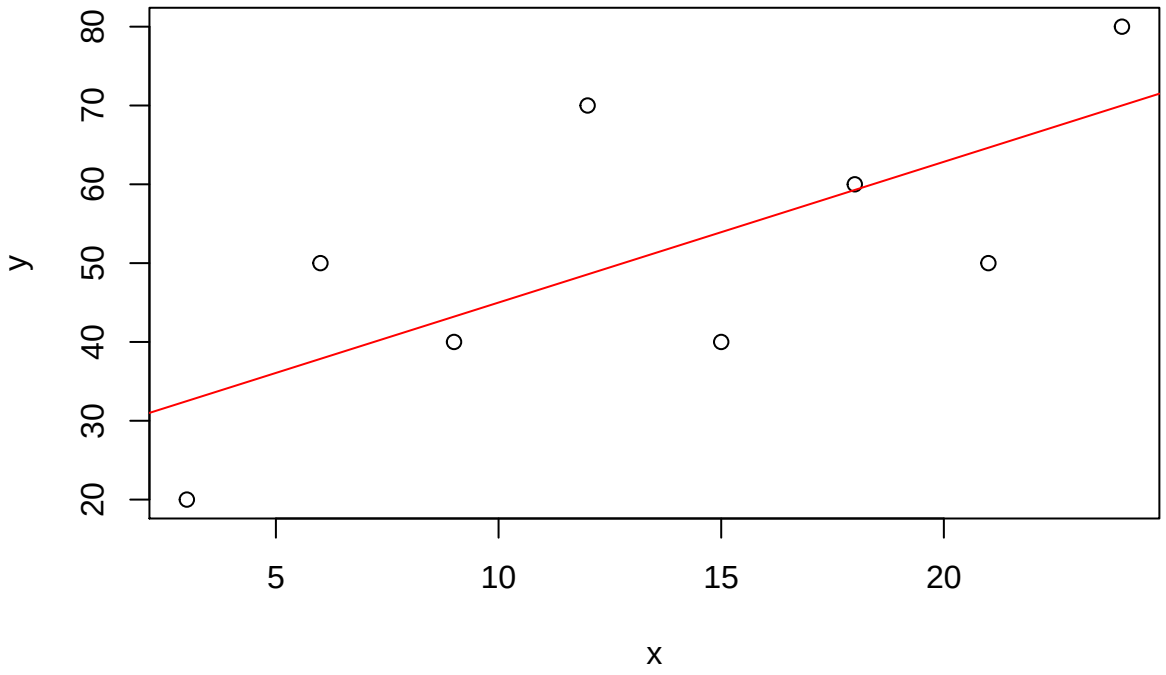
```
# Coefficient de correlation
cor(x,y)
```

```
## [1] 0.6961075
```

```
# Methode de fisher
fisher.test(x,y)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  x and y
## p-value = 1
## alternative hypothesis: two.sided
```

```
reg <- lm(y~x)
plot(x,y)
abline(reg,col="red")
```



Exercice 21

```
y <- c(85,70,100,140,115,105)
x1 <- c(3,5,9,12,14,17)
x2 <- c(11,14,15,16,19,23)
regm <- lm(y~x1+x2)
anova(regm)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## x1         1 1284.03 1284.03  31.821 0.01102 *
## x2         1 1532.42 1532.42  37.977 0.00860 **
## Residuals  3  121.06   40.35
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Exercice 22

```
# load data
data <- read_excel("data/Donnees-TP2-M1-MIASHS.xls")
summary(data)
```

```
##      Sexe                Age          EtatCivil            Nbenfant
##  Length:168         Min.   :25.00   Length:168         Min.   :0.00
##  Class :character   1st Qu.:37.00   Class :character   1st Qu.:1.00
##  Mode  :character   Median :41.00   Mode  :character   Median :2.00
##                     Mean   :41.99                      Mean   :1.72
##                     3rd Qu.:49.25                      3rd Qu.:2.00
##                     Max.   :57.00                      Max.   :5.00
##    Diplome            Anciennete       Salaire      Satisfaction
##  Length:168         Min.   : 1.00   Min.   :1200   Min.   : 3.85
##  Class :character   1st Qu.:10.00   1st Qu.:1650   1st Qu.:13.84
##  Mode  :character   Median :15.00   Median :1720   Median :19.17
```

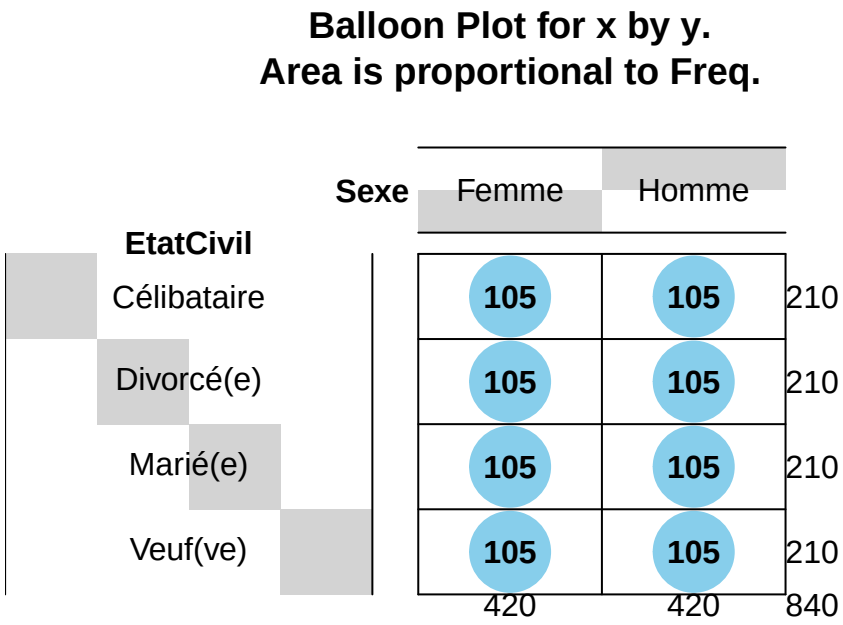
```
##                     Mean   :16.55   Mean   :1778   Mean   :20.43
##                     3rd Qu.:24.25   3rd Qu.:1908   3rd Qu.:28.31
##                     Max.   :34.00   Max.   :2200   Max.   :38.45
##      Stress         EstimeSoi      AvisReforme
##  Min.   : 3.70   Min.   : 3.54   Length:168
##  1st Qu.:15.19   1st Qu.:14.03   Class :character
##  Median :18.19   Median :19.68   Mode  :character
##  Mean   :18.20   Mean   :21.08
##  3rd Qu.:21.11   3rd Qu.:29.84
##  Max.   :31.84   Max.   :42.15
```

Tableau de contigence

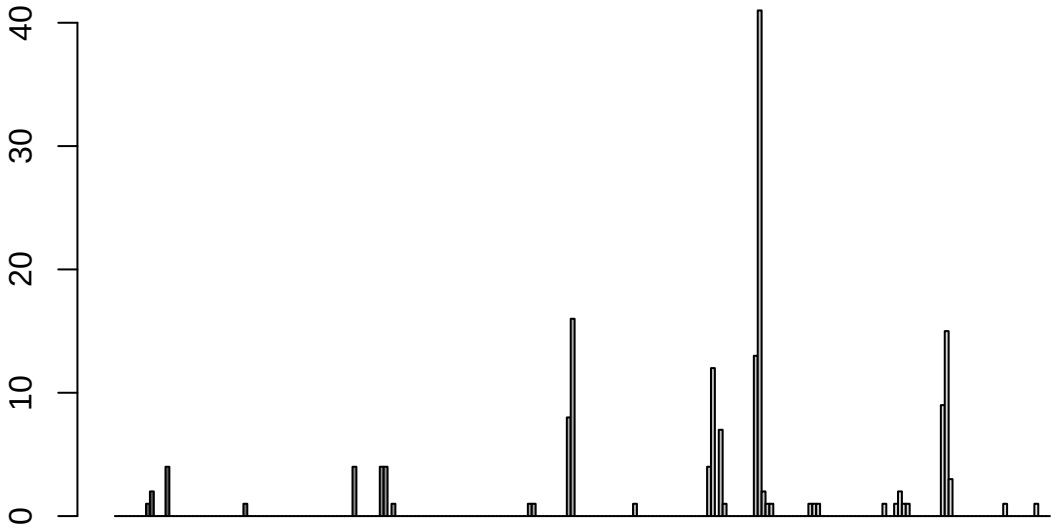
```
datavq <- subset(data,select=c(Sexe,EtatCivil,Diplome,AvisReforme))
tabc1 <- table(datavq)
tabc2 <- round(tabc1/sum(tabc1),2)
tabc3 <- tabc2*100
```

Representation graphique

```
balloonplot(tabc1)
```




```
barplot(as.matrix(tabc1),beside = TRUE)
```



Croisement qualitatif vs qualitatif

```
# distribution marginale
matfreq <- tabc1 / sum(tabc1)
dl <- round(apply(matfreq,1,sum),2)
dl
```

```
## Femme Homme
##  0.32  0.68
```

```
dc <- round(apply(matfreq,2,sum),2)
dc
```

```
## Célibataire  Divorcé(e)    Marié(e)    Veuf(ve)
##        0.14        0.08        0.74        0.04
```

```
# distribution conditionnelle
DC <- sweep(tabc1,2,colSums(tabc1),"/")
```

```
## Warning in sweep(tabc1, 2, colSums(tabc1), "/"): STATS is longer than the extent
## of 'dim(x)[MARGIN]'
```

```
# test de khi-deux
dfse <- subset(data,select = c(Sexe,EtatCivil))
tabdfse <- table(dfse)
chisq.test(tabdfse)
```

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

```
##
##  Pearson's Chi-squared test
##
## data:  tabdfse
## X-squared = 5.6972, df = 3, p-value = 0.1273
```

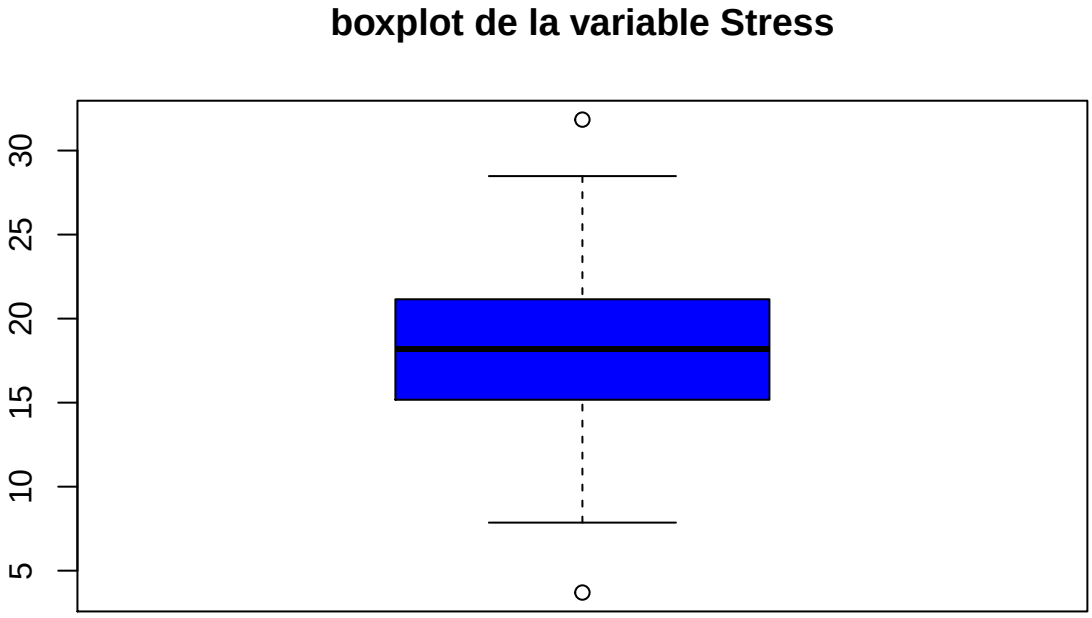
```
# p-value > 0.05 donc pas de dependance significative entre les deux variables
```

Croisement qualitatif vs quantitatif

```
# Resumé de la variable stress
s <- data$Stress
summary(s)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3.70   15.19   18.19   18.20   21.11   31.84
```

```
boxplot(s,col="blue",main="boxplot de la variable Stress")
```




```
datf <- subset(data,select = c(Stress,EtatCivil))
tabSE <- table(datf) # effectif
head(tabSE / sum(tabSE),6)
```

```
##                   EtatCivil
## Stress             Célibataire  Divorcé(e)    Marié(e)    Veuf(ve)
##   3.7              0.000000000 0.000000000 0.005952381 0.000000000
##   7.86             0.005952381 0.000000000 0.000000000 0.000000000
##   8                0.000000000 0.005952381 0.000000000 0.000000000
##   8.34             0.000000000 0.000000000 0.005952381 0.000000000
##   9.12             0.000000000 0.000000000 0.000000000 0.005952381
##   9.18000000000001 0.005952381 0.000000000 0.000000000 0.000000000
```

```
head(round(tabSE / sum(tabSE),2) * 100)
```

```
##                   EtatCivil
## Stress             Célibataire Divorcé(e) Marié(e) Veuf(ve)
##   3.7                        0          0        1        0
##   7.86                       1          0        0        0
##   8                          0          1        0        0
##   8.34                       0          0        1        0
##   9.12                       0          0        0        1
##   9.18000000000001           1          0        0        0
```

```
chisq.test(tabSE)
```

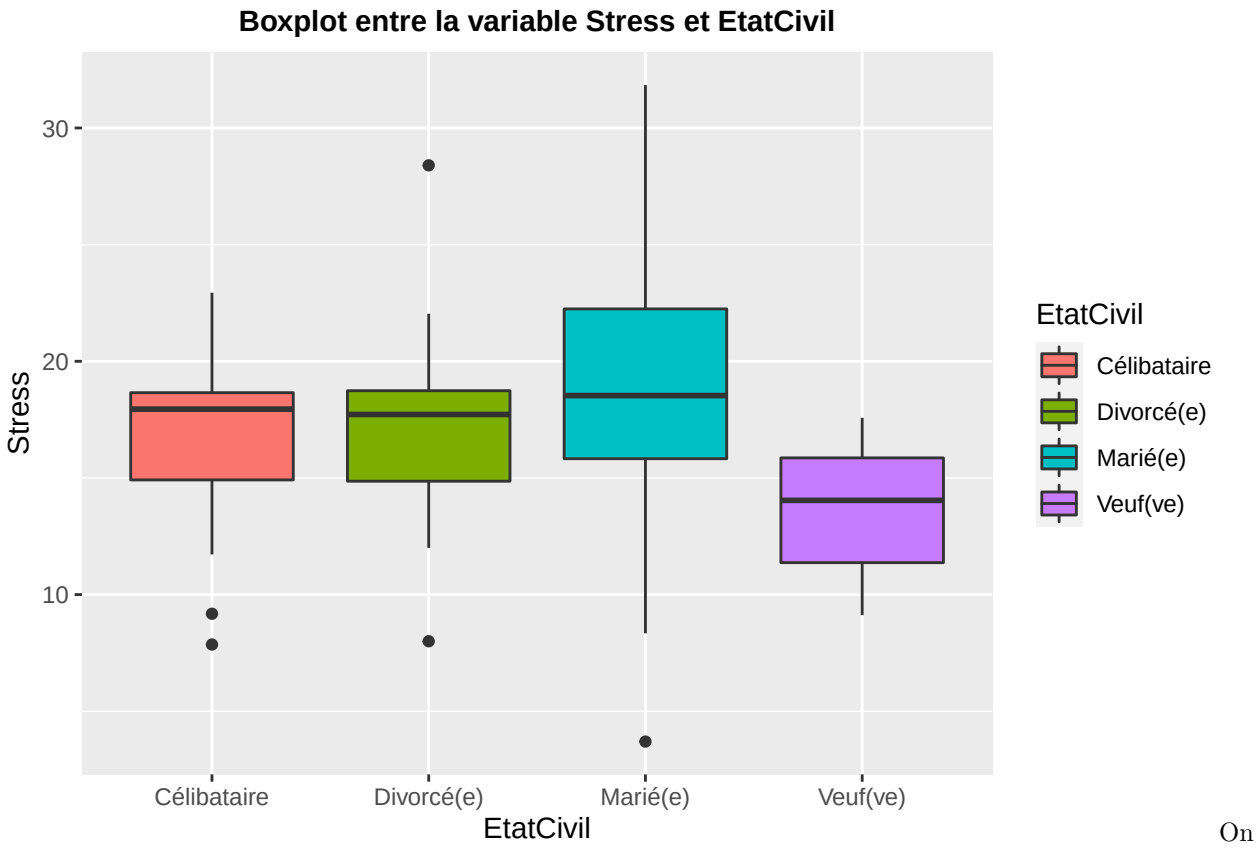
```
## Warning in chisq.test(tabSE): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  tabSE
```

```
## X-squared = 471.47, df = 465, p-value = 0.4078
```

e.

```
# Realisation d'un boxplot
g <- ggplot(datf) + geom_boxplot(aes(x = EtatCivil, y = Stress, fill = EtatCivil)) +
  ggtitle("Boxplot entre la variable Stress et EtatCivil")
g + theme (plot.title = element_text(size=11,face="bold",hjust = 0.5))
```



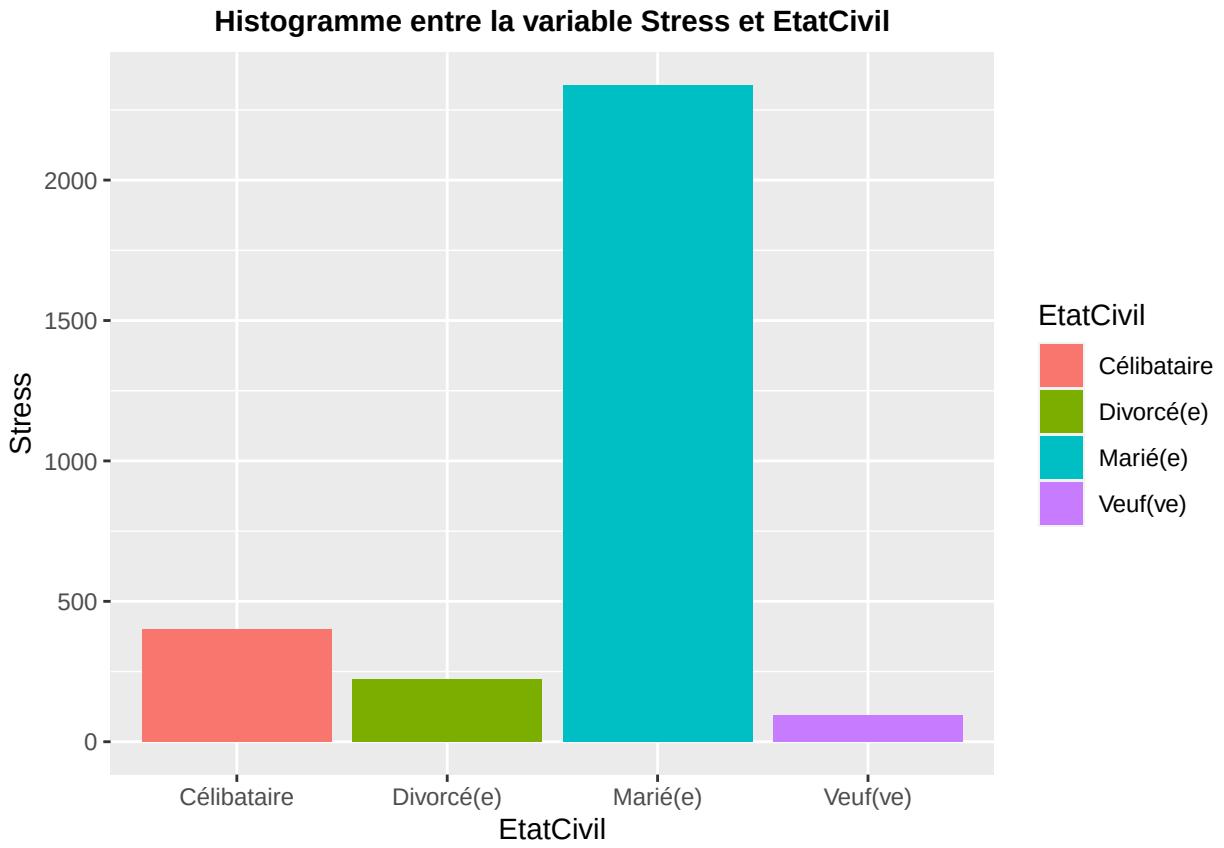
On remarque qu'il y a plus d'individus mariés car ils ont la moyenne la plus elevés et autant d'individus celibataires que divorcés car leurs moyennes sont identiques et moins d'individus Veufs.

f.

```
g <- ggplot(datf) + geom_histogram(stat="identity",aes(x = EtatCivil, y = Stress, fill = EtatCivil)) +
  ggtitle("Histogramme entre la variable Stress et EtatCivil")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
g + theme (plot.title = element_text(size=11,face="bold",hjust = 0.5))
```



g.

```
head(datf)
```

```
## # A tibble: 6 x 2
##   Stress EtatCivil
##    <dbl> <chr>
## 1   15.7 Célibataire
## 2   18.9 Célibataire
## 3   21.4 Célibataire
## 4   13.9 Marié(e)
## 5   17.9 Marié(e)
## 6   18.8 Marié(e)
```

6.Croisement quantitatif vs quantitatif

```
dfas <- subset(data,select = c(Age,Satisfaction))
summary(dfas)
```

```
##       Age         Satisfaction
##  Min.   :25.00   Min.   : 3.85
##  1st Qu.:37.00   1st Qu.:13.84
##  Median :41.00   Median :19.17
##  Mean   :41.99   Mean   :20.43
##  3rd Qu.:49.25   3rd Qu.:28.31
##  Max.   :57.00   Max.   :38.45
```

```
boxplot(dfas$Satisfaction,xlab="Satisfaction",main="Boxplot pour la variable Satisfaction")
```

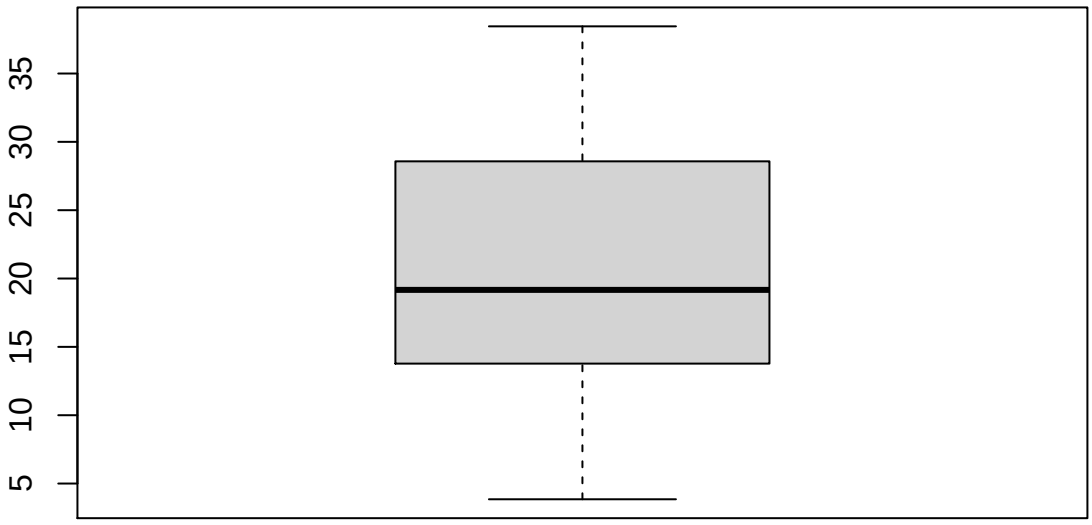
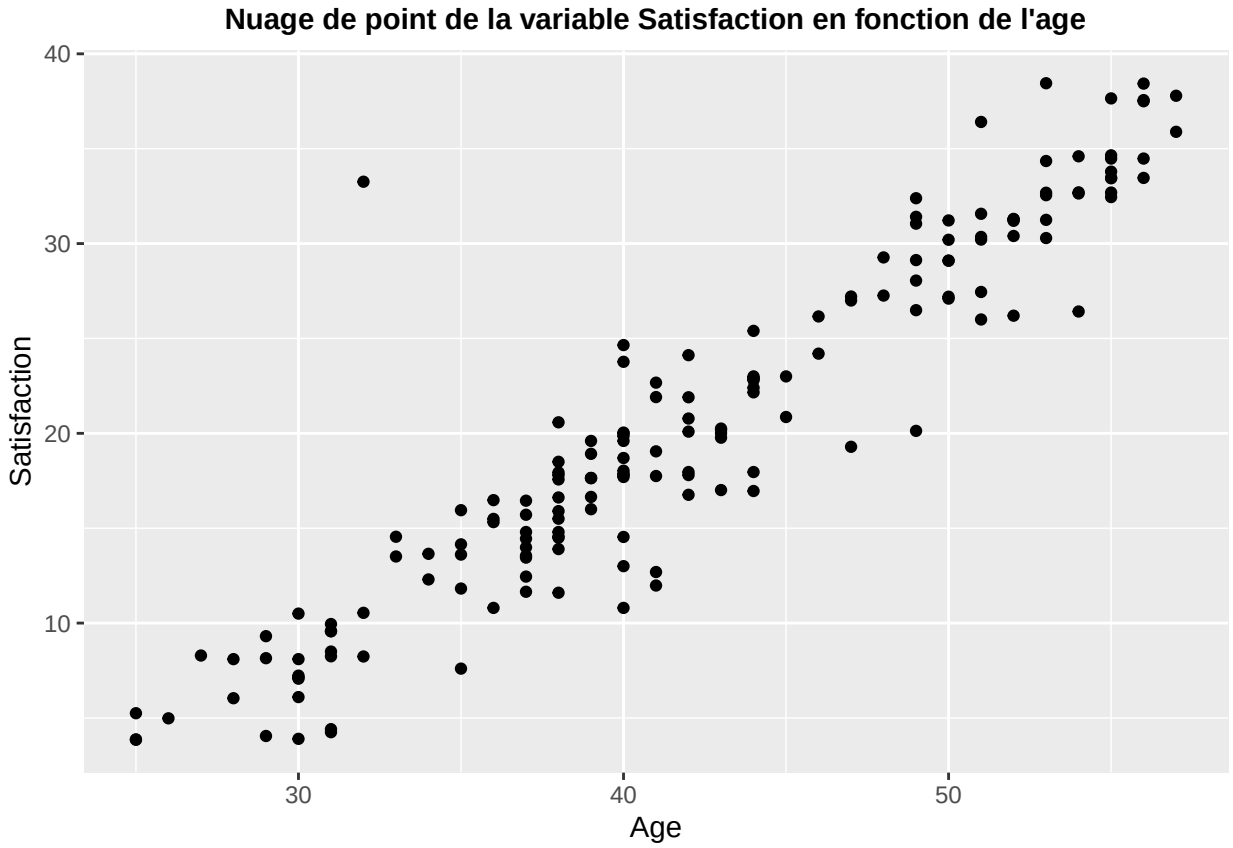


Tableau de contingence

```
# tableau des effectifs
tabcas <- table(dfas)
# tableau de frequence
tabfreq <- tabcas / sum(tabcas)
tabfreqp <- round(tabfreq * 100,2)
```

Nuage de point

```
g <- ggplot(dfas) + geom_point(stat="identity",aes(x = Age, y = Satisfaction)) +
  ggtitle("Nuage de point de la variable Satisfaction en fonction de l'age")
g + theme (plot.title = element_text(size=11,face="bold",hjust = 0.5))
```



m.

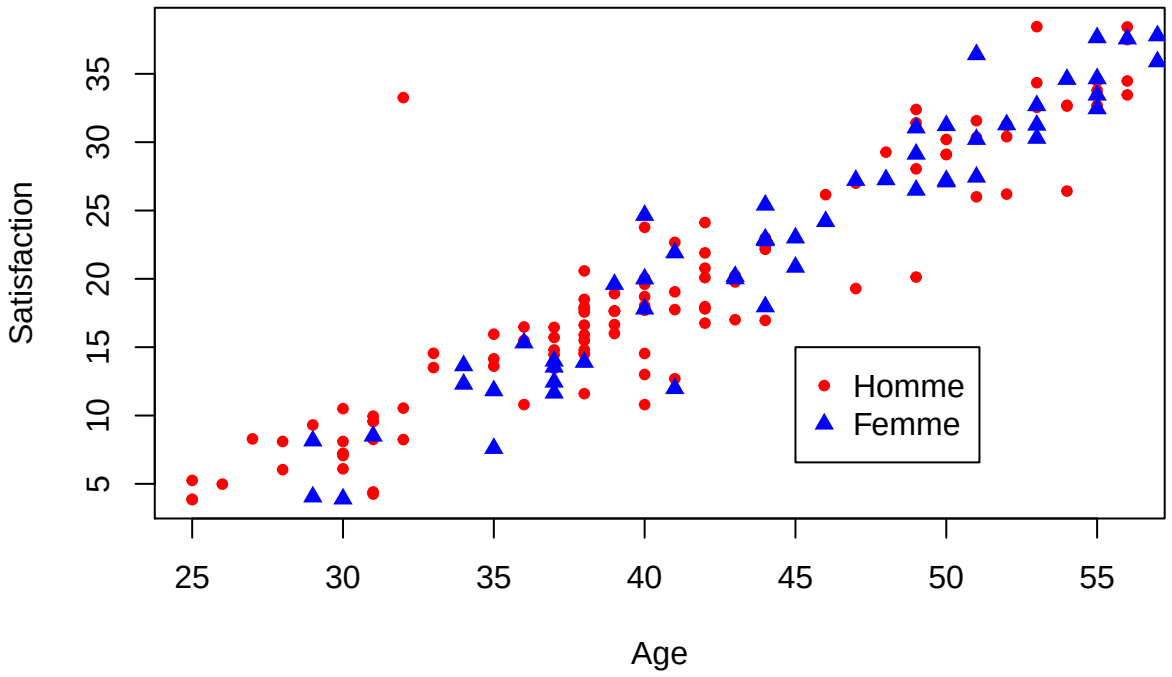
```
spl <- split(dfas,data$Sexe)
dim(spl$Homme)
```

```
## [1] 115   2
```

```
dim(spl$Femme)
```

```
## [1] 53  2
```

```
plot(spl$Homme,col="red",pch=20)
points(spl$Femme,col="blue",pch=17)
legend(45,15,c("Homme","Femme"),col=c("red","blue"),pch = c(20,17))
```



n

```
# taille des echantillons
n <- length(dfas$Age)
X <- dfas$Age
Y <- dfas$Satisfaction
covobs <- sum((X - mean(X)) * (Y - mean(Y))) / (n - 1)
covobs
```

```
## [1] 72.21181
```

```
cov(X,Y)
```

```
## [1] 72.21181
```

```
# Correlation
num <-  (1 / (n - 1)) * sum((X - mean(X)) * (Y - mean(Y)))
denom <- sqrt(((1 / (n - 1)) * sum((X - mean(X))^2)) * ((1 / (n - 1)) * sum((Y - mean(Y))^2)))
corobs <- num / denom
corobs
```

```
## [1] 0.9380404
```

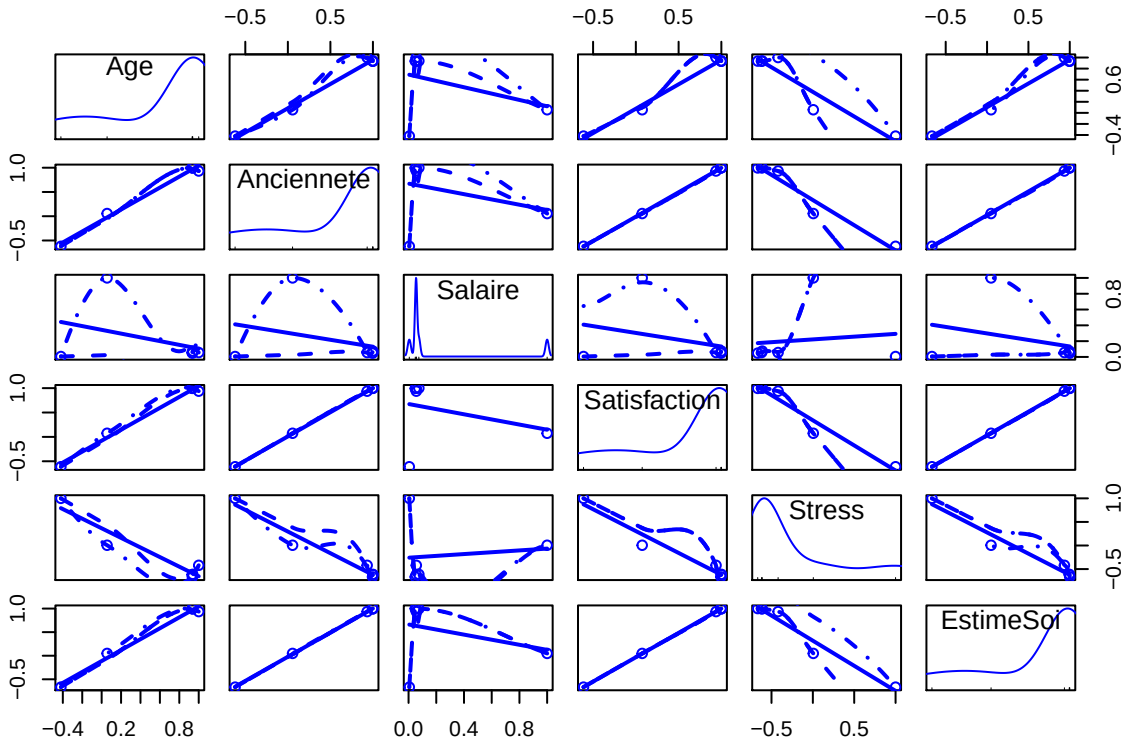
```
cor(X,Y)
```

```
## [1] 0.9380404
```

```
# Matrice de correlation
dfcorr <- subset(data,select = c(Age,Anciennete,Salaire,Satisfaction,Stress,EstimeSoi))
matcorr <- cor(dfcorr)
scatterplotMatrix(matcorr)
```

```
## Warning in smoother(x[subs], y[subs], col = smoother.args$col[i], log.x =
## FALSE, : could not fit smooth
```

```
## Warning in smoother(x[subs], y[subs], col = smoother.args$col[i], log.x =
## FALSE, : could not fit negative part of the spread
```



Le graphe montre les correlations possibles sur les differentes variablees, par exemple les variables Age et Anciennete sont correlés positivement.