Devoir TP 0

#Exercice 13

#la matrice des données

poids <- read.table("C/Users/ProDesK/Desktop/Poids_naissance.csv", header = TRUE)

View(poids)

#Convertion des donnés des poids des mamansen kg.

poids <- transform(poids,LWT=LWT*0.4535923)

	ID	AGE	LWT	RACE	SMOKE	PTL	HT	UI	FVT	BWT	LOW
1	85	19	182	2	0	0	0	1	0	2523	0
2	86	33	155	3	0	0	0	0	3	2551	0
3	87	20	105	1	1	0	0	0	1	2557	0
4	88	21	108	1	1	0	0	1	2	2594	0
5	89	18	107	1	1	0	0	1	0	2600	0
6	91	21	124	3	0	0	0	0	0	2622	0
7	92	22	118	1	0	0	0	0	1	2637	0
8	93	17	103	3	0	0	0	0	1	2637	0
9	94	29	123	1	1	0	0	0	1	2663	0
10	95	26	113	1	1	0	0	0	0	2665	0
11	96	19	95	3	0	0	0	0	0	2722	0

La convertion

	ID	AGE	LWT	RACE	SMOKE	PTL	HT	UI	FVT	BWT	LOW
1	85	19	82.55380	2	0	0	0	1	0	2523	0
2	86	33	70.30681	3	0	0	0	0	3	2551	0
3	87	20	47.62719	1	1	0	0	0	1	2557	0
4	88	21	48.98797	1	1	0	0	1	2	2594	0
5	89	18	48.53438	1	1	0	0	1	0	2600	0
6	91	21	56.24545	3	0	0	0	0	0	2622	0
7	92	22	53.52369	1	0	0	0	0	1	2637	0
8	93	17	46.72001	3	0	0	0	0	1	2637	0
9	94	29	55.79165	1	1	0	0	0	1	2663	0
10	95	26	51.25593	1	1	0	0	0	0	2665	0
11	96	19	43.09127	3	0	0	0	0	0	2722	0
12	97	19	68.03884	3	0	0	0	0	1	2733	0
13	98	22	43.09127	3	0	0	1	0	0	2750	0
14	99	30	48.53438	3	0	1	0	1	2	2750	0

#Exercice 14

#Saisie des donnees

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")
acteurs = data.frame(Mort.à, Années.de.carrière, Nombre.de.films, Prénom,
                     Nom, Date.du.décès)
#Modification du nom de la 1ere colonne
colnames(acteurs)[1] = "Age.du.décès"
#Extraction de la colonne Prénom
acteurs$Prénom
#Ordonner la data frame par ordre croissant suivant l'âge de la mort.
acteurs[order(acteurs$Age.du.décès), ]
```

	Mort.à	Années.de.carrière	Nombre.de.films	Prénom	Nom	Date.du.décès
1	93	66	211	Michel	Galabru	04-01-2016
2	53	25	58	André	Raimbourg	23-09-1970
3	72	48	98	Jean	Gabin	15-10-1976
4	68	37	140	Louis	de Funès	27-01-1983
5	68	31	74	Lino	Ventura	22-10-1987
6	53	32	81	Jacques	Villeret	28-01-2005

Modification du nom de la premiere colonne

	Age.du.décès	Années.de.carrière	Nombre.de.films	Prénom	Nom	Date.du.décès
1	93	66	211	Michel	Galabru	04-01-2016
2	53	25	58	André	Raimbourg	23-09-1970
3	72	48	98	Jean	Gabin	15-10-1976
4	68	37	140	Louis	de Funès	27-01-1983
5	68	31	74	Lino	Ventura	22-10-1987
6	53	32	81	Jacques	Villeret	28-01-2005

Extraction de la colonne prenom

```
[1] Michel  André   Jean    Louis   Lino    Jacques
Levels: André Jacques Jean Lino Louis Michel
```

```
  Age.du.décès Années.de.carrière Nombre.de.films  Prénom       Nom Date.du.décès
2           53                 25              58   André Raimbourg    23-09-1970
6           53                 32              81 Jacques  Villeret    28-01-2005
4           68                 37             140   Louis  de Funès    27-01-1983
5           68                 31              74    Lino   Ventura    22-10-1987
3           72                 48              98    Jean     Gabin    15-10-1976
1           93                 66             211  Michel   Galabru    04-01-2016
>
```

#Exercice 15

w <- read.table("C:/Users/ProDesK/Desktop/fromage.txt", header = TRUE)

attach(w)

X1

str(w) #caracteristiques de w

summary(w) # les paramètres statistiques élémentaires pour les variables Y , X1, X2 et X3

pairs(w)#Cela renvoie les nuages de points des variables deux à deux.

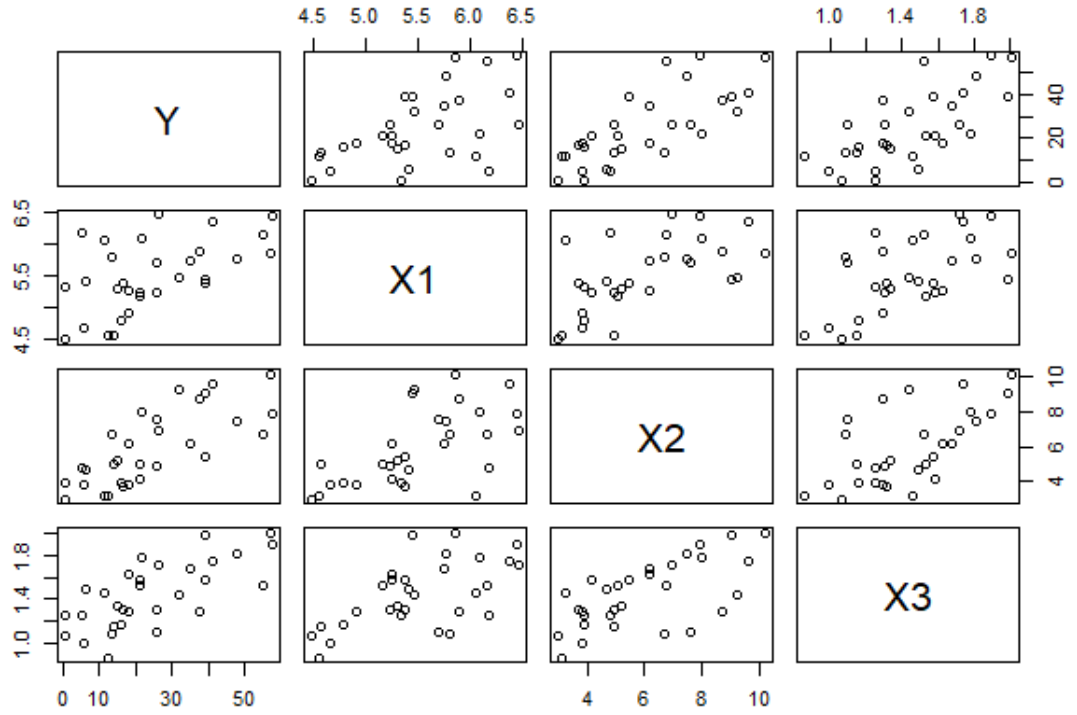
ww = w[(X1 > 5.1) & (X3 < 1.77),] #construcion du nouveau data frame

str(ww) #caracteristiques de ww

summary(ww) #donner les paramètres statistiques élémentaires pour les variables Y ,X1, X2 et X3.

```
> X1
 [1] 4.543 5.159 5.366 5.759 4.663 5.697 5.892 6.078 4.898 5.242 5.740 6.446 4.477 5.236 6.151 6.365 4.787
[18] 5.412 5.247 5.438 4.564 5.298 5.455 5.855 5.366 6.043 6.458 5.328 5.802 6.176
```

```
> str(w) #caracteristiques de w
'data.frame':   30 obs. of  4 variables:
 $ Y : num  12.3 20.9 39 47.9 5.6 25.9 37.3 21.9 18.1 21 ...
 $ X1: num  4.54 5.16 5.37 5.76 4.66 ...
 $ X2: num  3.13 5.04 5.44 7.5 3.81 ...
 $ X3: num  0.86 1.53 1.57 1.81 0.99 1.09 1.29 1.78 1.29 1.58 ...
>
```



#Exercice 16

data(airquality)

?airquality

	Ozone	Solar.R	Wind	Temp	Month	Day
1	41	190	7.4	67	5	1
2	36	118	8.0	72	5	2
3	12	149	12.6	74	5	3
4	18	313	11.5	62	5	4
5	*NA*	*NA*	14.3	56	5	5
6	28	*NA*	14.9	66	5	6
7	23	299	8.6	65	5	7
8	19	99	13.8	59	5	8
9	8	19	20.1	61	5	9
10	*NA*	194	8.6	69	5	10
11	7	*NA*	6.9	74	5	11
12	16	256	9.7	69	5	12
13	11	290	9.2	66	5	13
14	14	274	10.9	68	5	14

New York Air Quality Measurements

Description

Daily air quality measurements in New York, May to September 1973.

Usage

```
airquality
```

Format

A data frame with 153 observations on 6 variables.

`[,1] Ozone`	numeric Ozone (ppb)
`[,2] Solar.R`	numeric Solar R (lang)
`[,3] Wind`	numeric Wind (mph)
`[,4] Temp`	numeric Temperature (degrees F)
`[,5] Month`	numeric Month (1--12)
`[,6] Day`	numeric Day of month (1--31)

Details

names(airquality) #les noms des variables considérées

dim(airquality) # le nombre de lignes et de colonnes

summary(airquality) #Calcul des paramètres statistiques

boxplot(airquality$Ozone~airquality$Month) #représentation de la boîte à moustaches de la variable Ozone pour chaque mois

#Création de la variable saison

install.packages("dplyr")

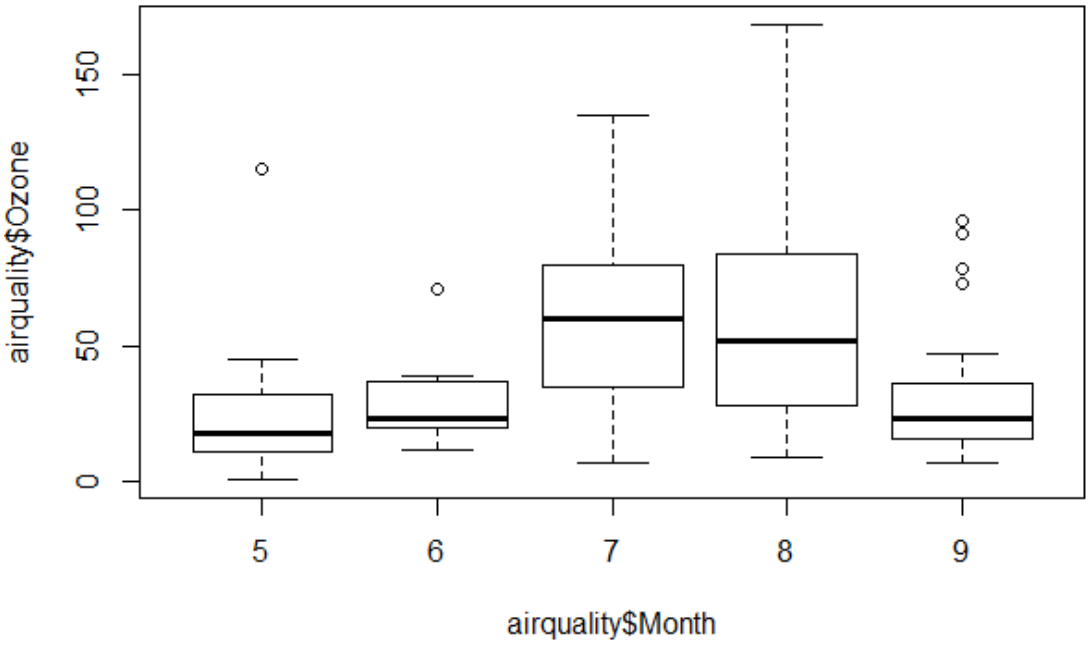
library(dplyr)

```
saison <-case_when (
 airquality$Month==5 ~ "printemps",
 airquality$Month==6 |airquality$Month==7|airquality$Month==8 ~ "été",
 airquality$Month==9 ~ "automne",
 TRUE ~ "Autre"

)
```

install.packages("ggplot2")

library(ggplot2)

qplot(Temp, Ozone, data = airquality, colour = Month)



#Exercice 17

#simulation

X=rnorm(100,0,5)

head(X)

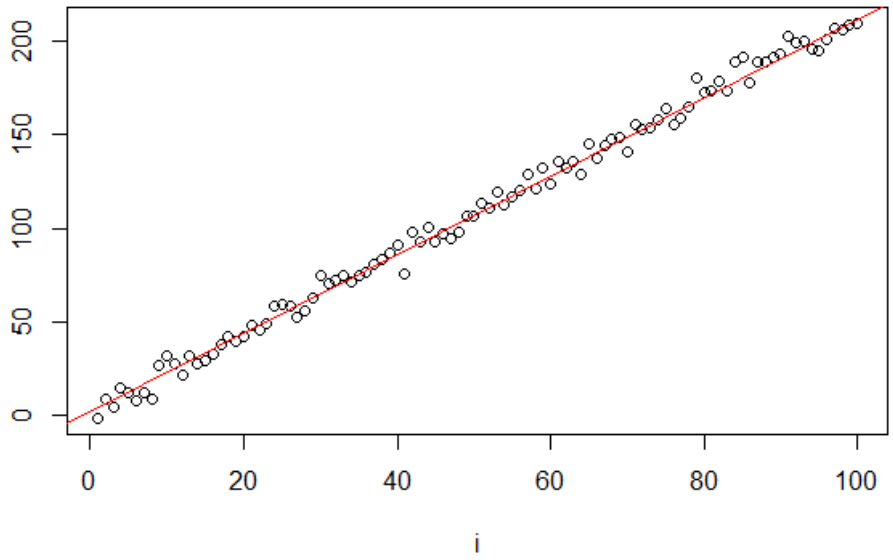
#graphique

i=c(1:100)

Y=1.7+2.1*i+X

plot(i,Y)

abline(a=1.7, b=2.1, col="red")




```
#Exercice 18
#saisi des DONNEES
brun =c(68,15,5,20)
chatin=c(119,54,29,84)
roux=c(26,14,14,17)
blond=c(7,10,16,94)
couleur=data.frame(brun,chatin,roux,blond)
View(couleur)
#Calcul de la matrice des frequences
freq <- couleur/sum(couleur)
round(freq*100,digit=2)
#distributions marginales r pour les lignes et c pour les colonnes
r <- apply(freq,1,sum)
round(r,digit=2)
c <-  apply(freq ,2,sum)
round(c,digit=2)
#Matrice des profils-lignes L (distributions conditionnelles en ligne)
L <- sweep(freq,1,STAT=r,FUN="/")
round(L,digits=2)
#Matrice des profils colonnes C (distributions conditionnelles en colonnes)
```

```
C <- sweep(freq,1,STAT=c,FUN="/")

round(L,digits=2)

sum((L[1,]-L[2,])^2/c)#carre de la distance entre marron et noisette

sum((L[1,]-L[4,])^2/c)#carre de la distance entre marron et bleu

sum((L[1,]-c)^2/c)#carre de la distance entre marron et moyenne

#Matrice des taux de liaisons :

T <- (freq-r%*%t(c))/(r%*%t(c))

round(T,digit=2)

#test de khi-deux

chisq.test(couleur)$statistic

#on remarque que le test de khi-deux=138.2898

#138.2898>23.59, donc on rejette H0 qui estime qu'il n'a pas de liaison entre la couleur des yeux et la couleurdes cheveux.
```

TP1

```
#EXO19
#ecriture du data.frame
data <- data.frame(BEPC = c(15,10,15,40), BAC = c(12,18,5,35), Licence
                = c(3,4,8,15), Total = c(30,32,28,90))
rownames(data) <- c("Plus de 50 ans", "Entre 30 et 50 ans", "Moins de 30
ans", "Total")

data
#tableau de contingence de la frequence
freq <- data/sum(data)
round(freq*100,digit=2)
r <- apply(freq,1,sum)
round(r,digit=2)

c <-  apply(freq ,2,sum)
round(c,digit=2)
#Matrice des profils-lignes L (distributions conditionnelles en ligne)

L <- sweep(freq,1,STAT=r,FUN="/")
round(L,digits=2)
#Matrice des profils colonnes C (distributions conditionnelles en colonnes)

C <- sweep(freq,1,STAT=c,FUN="/")
round(C,digits=2)



chisq.test(data)
#11.175 > 9.49, les 2 variables ne sont pas indépendantes
```

```
                    BEPC BAC Licence Total
Plus de 50 ans        15  12       3    30
Entre 30 et 50 ans    10  18       4    32
Moins de 30\nans      15   5       8    28
Total                 40  35      15    90
```

```
> #tableau de contingence de la frequence
> freq <- data/sum(data)
> round(freq*100,digit=2)
                     BEPC  BAC Licence Total
Plus de 50 ans       4.17 3.33    0.83  8.33
Entre 30 et 50 ans   2.78 5.00    1.11  8.89
Moins de 30\nans     4.17 1.39    2.22  7.78
Total               11.11 9.72    4.17 25.00
```

4- test de khi-deux : chisq.test(data)

```
> chisq.test(data)

        Pearson's Chi-squared test

data:  data
X-squared = 11.175, df = 9, p-value = 0.2639
```

#11.175 > 9.49, les 2 variables ne sont pas indépendantes

#EXO20

#création du tableau

tableau <- matrix(c(290,410,110,190), ncol=2, byrow=TRUE)

colnames(tableau) <- c("Bleu","Brun")

rownames(tableau) <- c("Celib","Marie")

tableau

tableau <- as.table(tableau)

#representation graphique du contenu du tableau

barplot(tableau)

n <- margin.table(tableau) #nombre total de personnes

m1 <- margin.table(tableau,1)# nombre total de personnes par colonnes (celib/marie)

m2 <- margin.table(tableau,2) #nombre total de personnes par ligne (bleu/brun)

prop.table(tableau) #proportions

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

tab0 <- as.table(tab0)

summary(tableau)

summary(tab0)

#test de khi-deux pour l'échantillon HairEyeColor

HairEyeColor

HairEyeNew<- margin.table(HairEyeColor, margin = c(1,2))

chisq.test(HairEyeNew)

#Le test nous indique qu'une ou plusieurs relations sont plus fréquentes que les autres dans le tableau.

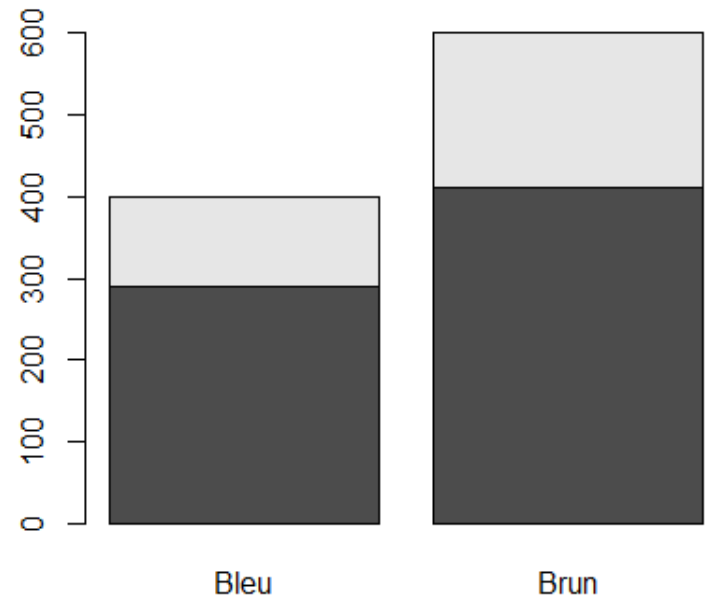
#Pour déterminer quelle relation entre la couleur des cheveux et des yeux est plus fréquente que les autres, nous allons

#calculer les proportions du tableau, comme indiqué ci-dessous.

HairEyeNew/sum(HairEyeNew)

#Comme vous pouvez le voir dans le tableau, les cheveux bruns et les yeux marron sont les plus fréquents (20%),

#suivis des cheveux blonds et des yeux bleus (15%).




```
n <- margin.table(tableau) #nombre total de personnes
m1 <- margin.table(tableau,1)# nombre total de personnes par colonnes (celib/marie)
m2 <- margin.table(tableau,2) #nombre total de personnes par ligne (bleu/brun)
prop.table(tableau) #proportions
```

```
> tab0 <- as.array(m1) %*% t(as.array(m2))/n
> tab0 <- as.table(tab0)
> summary(tableau)
Number of cases in table: 1000
Number of factors: 2
Test for independence of all factors:
        Chisq = 1.9841, df = 1, p-value = 0.159
> summary(tab0)
Number of cases in table: 1000
Number of factors: 2
Test for independence of all factors:
        Chisq = 1.154e-29, df = 1, p-value = 1
>
```

```
> HairEyeNew<- margin.table(HairEyeColor, margin = c(1,2))
> chisq.test(HairEyeNew)

        Pearson's Chi-squared test

data:  HairEyeNew
X-squared = 138.29, df = 9, p-value < 2.2e-16

> #Le test nous indique qu'une ou plusieurs relations sont plus fréquentes que les autres dans le tableau.
> #Pour déterminer quelle relation entre la couleur des cheveux et des yeux est plus fréquente que les autres, nous allons
> #calculer les proportions du tableau, comme indiqué ci-dessous.
> HairEyeNew/sum(HairEyeNew)
       Eye
Hair        Brown        Blue       Hazel       Green
  Black 0.114864865 0.033783784 0.025337838 0.008445946
  Brown 0.201013514 0.141891892 0.091216216 0.048986486
  Red   0.043918919 0.028716216 0.023648649 0.023648649
  Blond 0.011824324 0.158783784 0.016891892 0.027027027
```

#EXO21 :

data()

data(cars)

cars #Distance necessaire pour qu'une voiture s'arrete en fonction de la vitesse

names(cars)

dim(cars) #une matrice de taille 50 , 2

plot(cars) #on ne comprend pas grand chose de ce graphique

?lm

reg <- lm(dist ~ speed, data = cars)

attributes(reg)

summary(reg)

anova(reg)

names(reg)

plot(reg)

plot(cars,pch=20,col='blue')

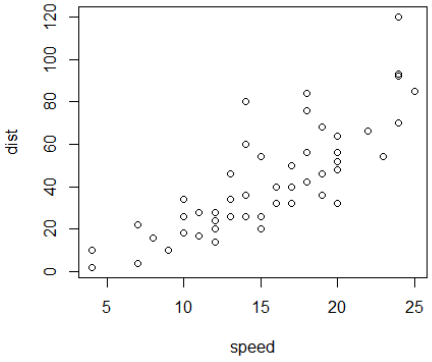
abline(reg=reg,col="red")

abline(reg$coeff,col="yellow")

hat<-predict(reg)

hat

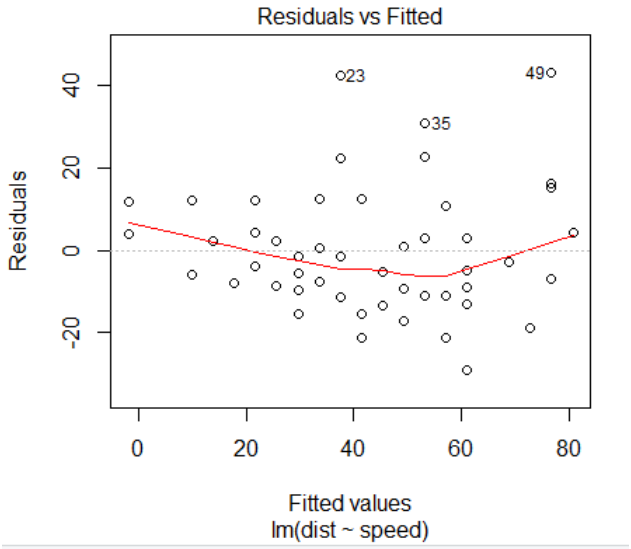
predict(reg , data.frame(speed = 20) , interval = "prediction")



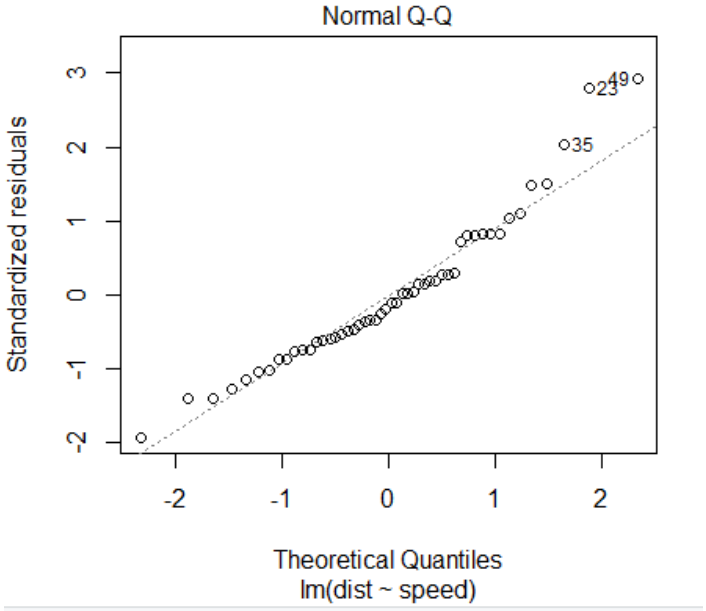
Les fonctions summary et anova

```
> reg <- lm(dist ~ speed, data = cars)
> attributes(reg)
$names
 [1] "coefficients"  "residuals"     "effects"       "rank"          "fitted.values" "assign"        "qr"
 [8] "df.residual"   "xlevels"       "call"          "terms"         "model"

$class
[1] "lm"

> 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

> anova(reg)
Analysis of Variance Table

Response: dist
          Df Sum Sq Mean Sq F value    Pr(>F)
speed      1  21186 21185.5  89.567 1.49e-12 ***
Residuals 48  11354   236.5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

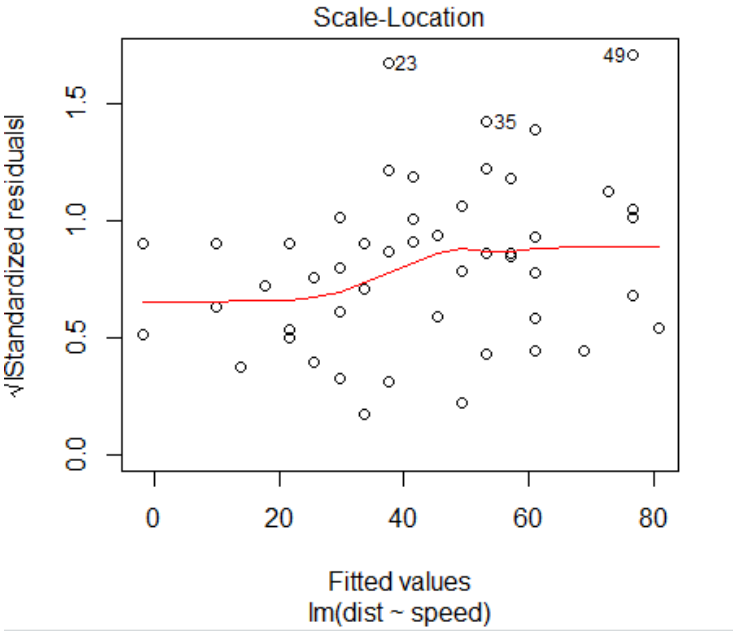
Le graphique 1 :



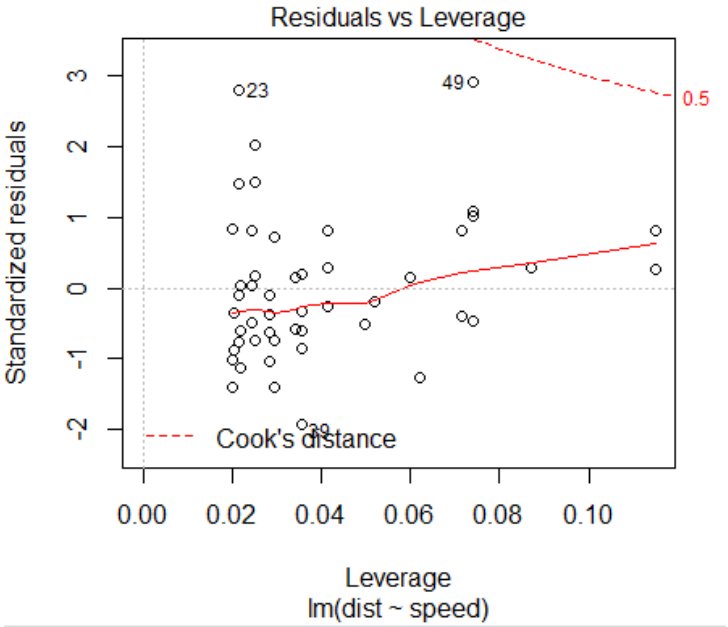
Le graphique 2 (QQ-plot) :



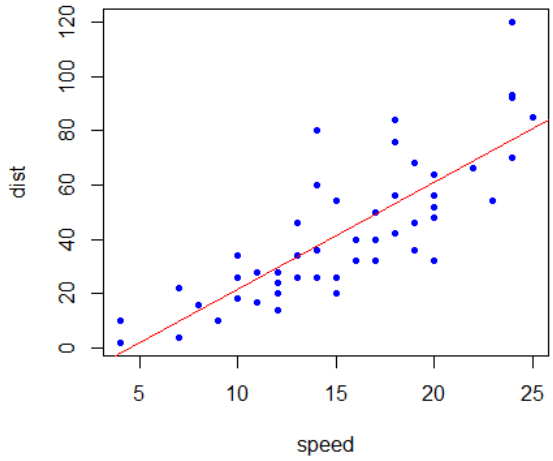
Le graphique 3 :



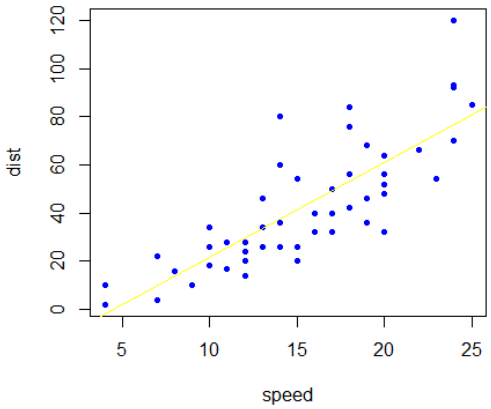
Le graphique 4 (Cook’s D) :



plot(cars,pch=20,col='blue')
abline(reg=reg,col='red')



avec abline(reg$coeff,col='yellow')



a)valeur prédite pour une vitesse de 20 :

```
> hat<-predict(reg)
> hat
        1         2         3         4         5         6         7         8         9        10        11        12        13
-1.849460 -1.849460  9.947766  9.947766 13.880175 17.812584 21.744993 21.744993 21.744993 25.677401 25.677401 29.609810 29.609810
       14        15        16        17        18        19        20        21        22        23        24        25        26
29.609810 29.609810 33.542219 33.542219 33.542219 33.542219 37.474628 37.474628 37.474628 37.474628 41.407036 41.407036 41.407036
       27        28        29        30        31        32        33        34        35        36        37        38        39
45.339445 45.339445 49.271854 49.271854 49.271854 53.204263 53.204263 53.204263 53.204263 57.136672 57.136672 57.136672 61.069080
       40        41        42        43        44        45        46        47        48        49        50
61.069080 61.069080 61.069080 61.069080 68.933898 72.866307 76.798715 76.798715 76.798715 76.798715 80.731124
```

⇒37.474628

b)

```
> predict(reg , data.frame(speed = 20) , interval = "prediction")
       fit      lwr      upr
1 61.06908 29.60309 92.53507
```

Devoir TP2

```
#Exercice 23

#Question 1 :

library("FactoMineR")
library("factoextra")

dataC<-data.frame(
  Indidvidus = c("Z1","Z2"),
  n1 = c(1.00 ,5.00),
  n2 = c(2.00 ,10.00),
  n3 = c(3.00 ,8.00),
  n4 = c(4.00 ,8.00),
  n5 = c(9.00 ,12.00)
)
dim(dataC)
#construire la matrice de correlation
pairs(dataC[,2:6])
Matricecorl<-cor(dataC[,2:6])

#observons si c'est une matrice identitaire

det(Matricecorl)

#faire l'APC
```

res.pca<-PCA(Matricecorl,scale.unit = TRUE,graph=TRUE)

#Interpretation : On voit que les individus n1 et n5 sont les individus qui contribuent le plus à droite de l'axe.

fviz_eig(res.pca, addlabels = TRUE, ylim = c(0, 50))

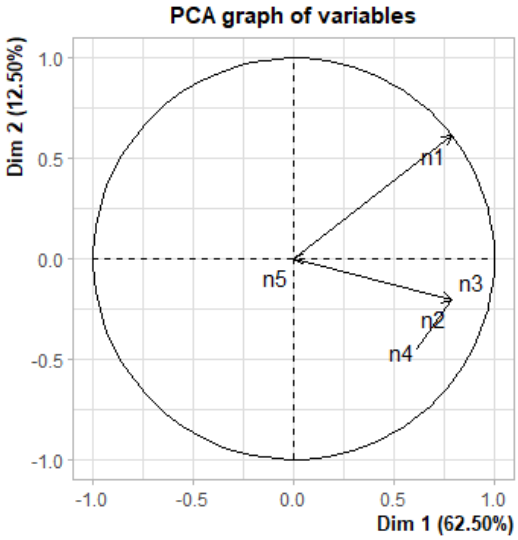
#Question 2

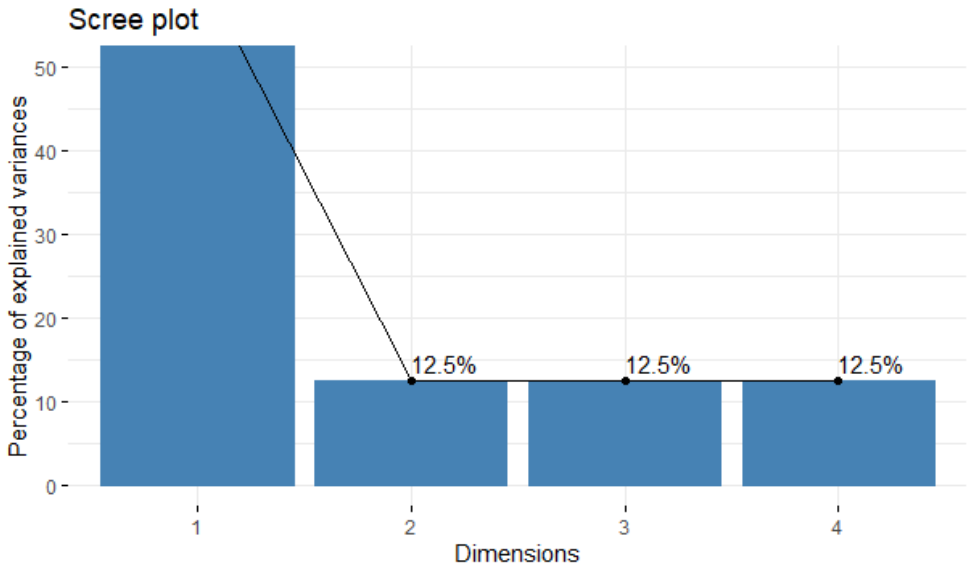
#en utilisant primcomp et prcomp

X<-princomp(Matricecorl,scale=FALSE)

Y<-prcomp(Matricecorl,scale=FALSE)

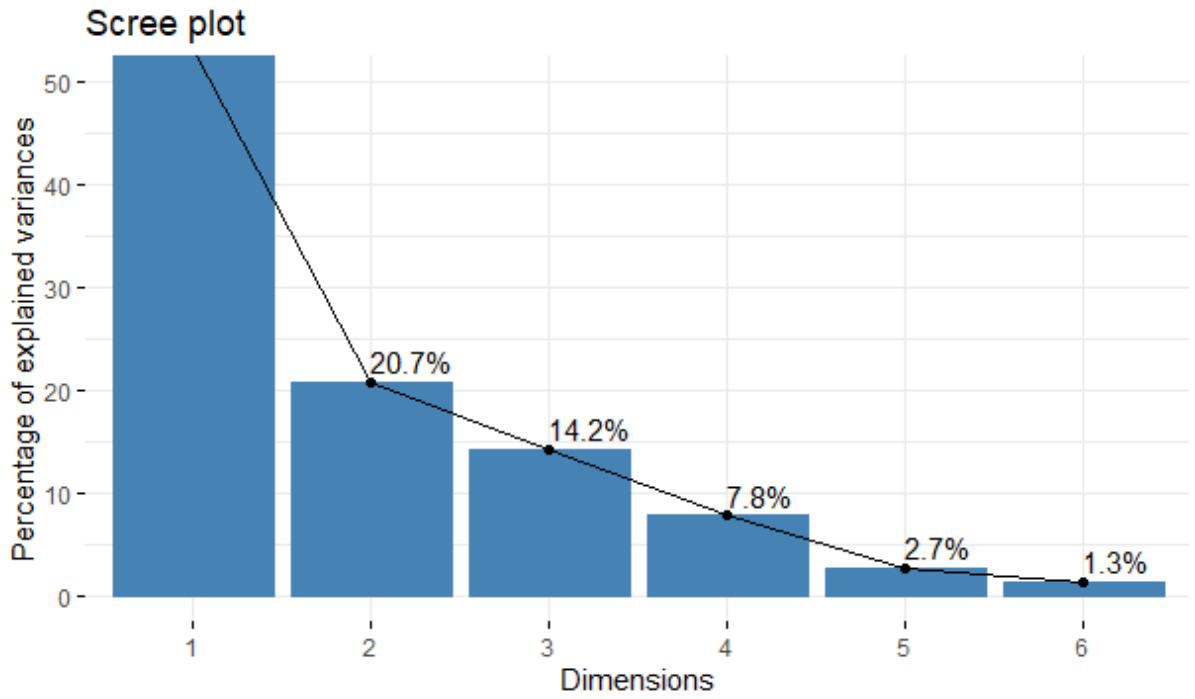
#Graphes




Scree plot
Percentage of explained variances
50
40
30
20
10
0
12.5%
12.5%
12.5%
1
2
3
4
Dimensions

Devoir TP2 station du ski

```
#Exercice 24 station du ski

stations <- read.csv("C:/Users/ProDesK/Downloads/post-197353-stations.txt", sep="")

res.pca1 <- PCA(stations,quali.sup =1, graph = FALSE)

#valeurs propres

valp<-- get_eigenvalue(res.pca1)

# La règle de Kaiser. Elle consiste à retenir les axes pour lesquels les valeurs propres sontsupérieures à 1 . Il est à noter qu'on peut aussi avoir des résultats d'ACP dont la somme des valeurs propres

#n'est pas égale à p . Dans ce cas, il faut adapter cette règle deKaiser et retenir les valeurs propres

#supérieures à la moyenne des valeurs propres, et non plus à 1.

#l'interprétation.

fviz_eig(res.pca1, addlabels = TRUE, ylim = c(0, 50))
```



pour l'axe 2, le calcul de la variabilité est : 20,789%=1,247/6 et le % cumulé est

73,998=20,789+53,209. Cela

#signifie que le deuxième axe comporte 20,789 de la variance (ou variabilité, ou inertie)

totale du nuage, et que le plan (1,2)

#totalise 73,998% de cette variance totale.

#interpréter les 2 axes.

var <- get_pca_var(res.pca1)

var

Coordonnées

head(var$coord)

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
prixforf	0.93031706	0.09513297	-0.08572117	0.1251448	-0.31055554
altmin	-0.07336694	0.82270492	0.48871130	0.2792394	0.02904032
altmax	0.65006226	0.53099234	-0.03832048	-0.5398817	0.04967488
pistes	0.95404437	-0.06226765	-0.11082956	0.1446174	0.05250905
kmfond	0.36193326	-0.50154750	0.76829658	-0.1613207	-0.03250806

remontee 0.92973674 -0.14239486 -0.03422684 0.1886935 0.23708189

Coordonnées

head(var$coord)

Dim.1 Dim.2 Dim.3 Dim.4 Dim.5

prixforf 0.865489833 0.009050282 0.007348118 0.01566121 0.0964447462

altmin 0.005382708 0.676843387 0.238838735 0.07797465 0.0008433401

altmax 0.422580938 0.281952868 0.001468459 0.29147230 0.0024675934

pistes 0.910200666 0.003877261 0.012283191 0.02091420 0.0027572005

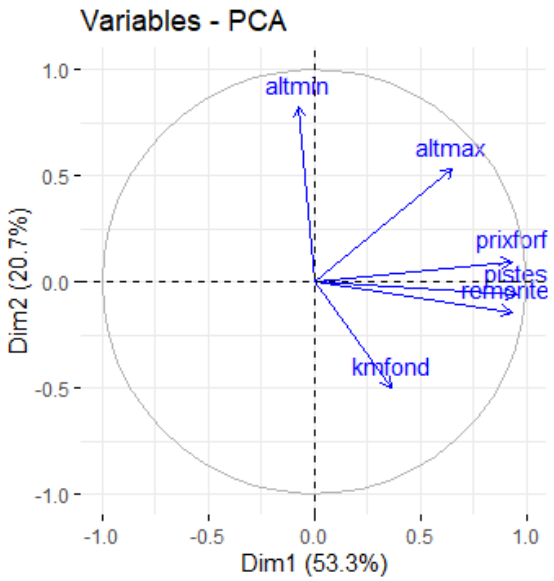
kmfond 0.130995683 0.251549890 0.590279627 0.02602438 0.0010567740

remontee 0.864410398 0.020276296 0.001171477 0.03560523 0.0562078214

Contributions aux composantes principales

head(var$contrib)

fviz_pca_var(res.pca1, col.var = "blue")



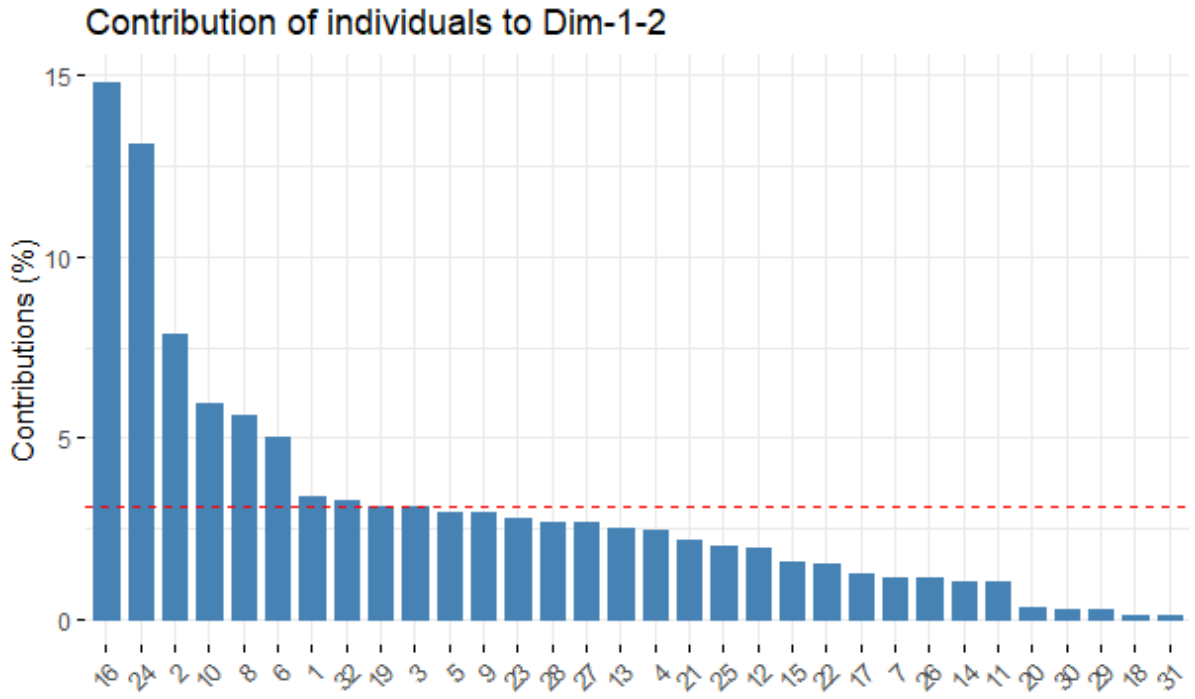
library("corrplot")

corrplot(var$cos2, is.corr=FALSE)

fviz_contrib(res.pca1, choice = "ind", axes = 1 :2)

ind <- get_pca_ind(res.pca1)

ind



Coordonnées des individus

head(ind$coord)

Dim.1 Dim.2 Dim.3 Dim.4 Dim.5

1 -0.9538555 -1.9658361 0.30654862 -0.1467118 0.07898986

2 3.2622418 -0.7109340 -1.25776337 -0.6421885 -0.20595549

3 -0.9670028 -1.8619314 -0.03410107 -0.9136117 -0.30583446

4 -1.3737068 -1.2479434 -1.75191973 -1.7603420 0.09790165

5 -2.0038380 -0.4415058 2.92715126 -0.3557049 0.15456409

6 -1.4108447 2.2642552 0.03318371 -0.3860225 0.26058662

Qualité des individus

head(ind$cos2)

Dim.1 Dim.2 Dim.3 Dim.4 Dim.5

1 0.1796712 0.76314664 0.0185571843 0.004250536 0.0012321285

2 0.8030306 0.03813804 0.1193706912 0.031118942 0.0032007117

3 0.1751260 0.64926578 0.0002177866 0.156321426 0.0175173337

4 0.1961070 0.16184332 0.3189579045 0.322032045 0.0009960588

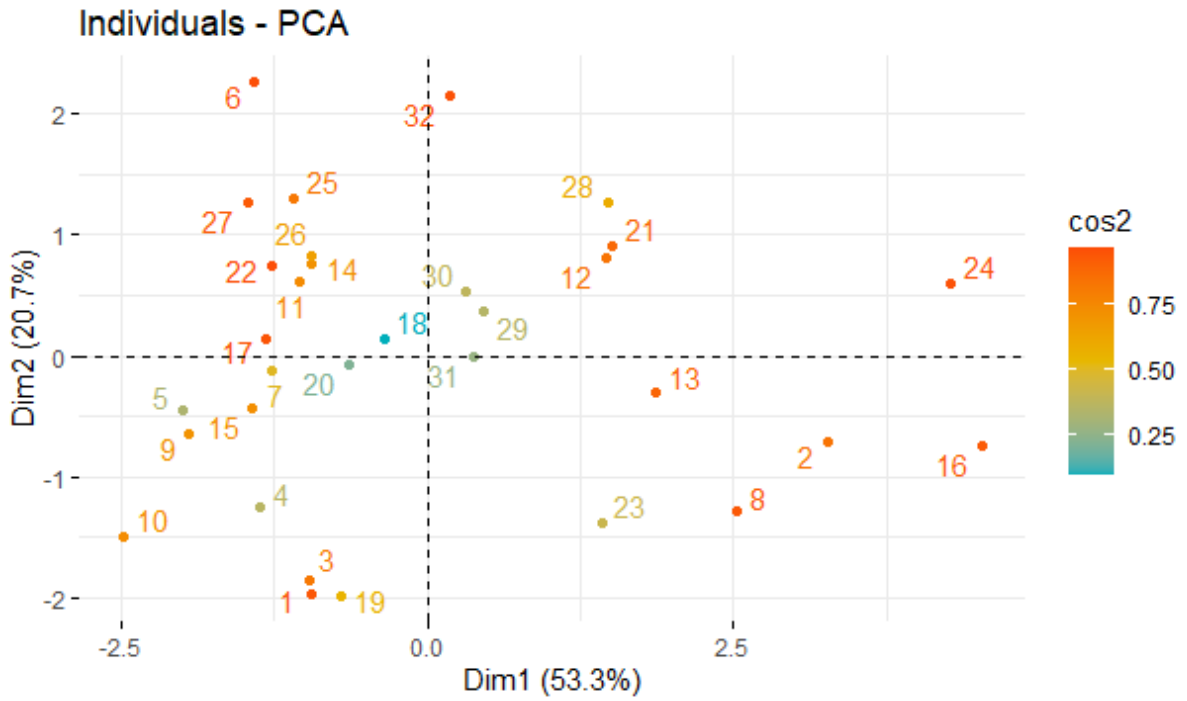
5 0.3100140 0.01504974 0.6615252023 0.009768678 0.0018444772

6 0.2712227 0.69858345 0.0001500436 0.020304517 0.0092527696

```
# Contributions des individus
head(ind$contrib)
```

```
  Dim.1      Dim.2       Dim.3      Dim.4     Dim.5
1  0.8887770  9.7113902  0.344921603  0.1438327 0.1220330
2 10.3958474  1.2701217  5.806568582  2.7558288 0.8296238
3  0.9134463  8.7119250  0.004268327  5.5776415 1.8293943
4  1.8433836  3.9136014 11.265489894 20.7071996 0.1874625
5  3.9224083  0.4898461 31.449374115  0.8454871 0.4672526
6  1.9444019 12.8836087  0.004041770  0.9957547 1.3281242
```

```
fviz_pca_ind (res.pca1)
fviz_pca_ind (res.pca1, col.ind = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE )
```


Individuals - PCA
Dim2 (20.7%)
Dim1 (53.3%)
cos2
0.75
0.50
0.25
2
1
0
-1
-2
-2.5
0.0
2.5
6
32
25
28
27
26
21
22
14
30
12
24
11
18
29
17
31
13
5
7
20
15
9
2
16
4
8
10
23
3
1
19

Devoir TP3

#Exercice 31

#1)

```
data("USArrests")

head(USArrests)
```

```
class(USArrests)
```

	Murder	Assault	UrbanPop	Rape
Alabama	13.2	236	58	21.2
Alaska	10.0	263	48	44.5
Arizona	8.1	294	80	31.0
Arkansas	8.8	190	50	19.5
California	9.0	276	91	40.6
Colorado	7.9	204	78	38.7

#2)

```
head(princomp(USArrests,cor=TRUE)$scores)

head(prcomp(USArrests,scale=TRUE)$x)
```

la fonction princomp :

	Comp.1	Comp.2	Comp.3	Comp.4
Alabama	0.9855659	1.1333924	0.44426879	0.156267145
Alaska	1.9501378	1.0732133	-2.04000333	-0.438583440
Arizona	1.7631635	-0.7459568	-0.05478082	-0.834652924
Arkansas	-0.1414203	1.1197968	-0.11457369	-0.182810896
California	2.5239801	-1.5429340	-0.59855680	-0.341996478
Colorado	1.5145629	-0.9875551	-1.09500699	0.001464887

la fonction prcomp :

	PC1	PC2	PC3	PC4
Alabama	-0.9756604	1.1220012	-0.43980366	0.154696581
Alaska	-1.9305379	1.0624269	2.01950027	-0.434175454
Arizona	-1.7454429	-0.7384595	0.05423025	-0.826264240
Arkansas	0.1399989	1.1085423	0.11342217	-0.180973554
California	-2.4986128	-1.5274267	0.59254100	-0.338559240
Colorado	-1.4993407	-0.9776297	1.08400162	0.001450164

```
#3)
#a)
Z <- scale(USArrests)

#b)

# fonction SVD generalisee avec metriques diagonales
gsvd <- function(Z,r,c) {

  N=diag(r)

  M=diag(c)

k <- qr(Z)$rank
colnames<-colnames(Z)
rownames<-rownames(Z)
Z <- as.matrix(Z)
Ztilde <- diag(sqrt(r)) %*% Z %*% diag(sqrt(c))
e <- svd(Ztilde)
U <-diag(1/sqrt(r))%*%e$u[,1 :k]
V <-diag(1/sqrt(c))%*%e$v[,1 :k]
d <- e$d[1 :k]
rownames(U) <- rownames
rownames(V) <- colnames
if (length(d)>1)
  colnames(U) <- colnames (V) <- paste("dim", 1 :k, sep = "")
return(list(U=U,V=V,d=d))
}
r <- rep(1/nrow(Z),nrow(Z))
```

```
c <- rep(1,ncol(Z))

U <- gsvd(Z,r,c)$U

d <- gsvd(Z,r,c)$d

Psi <- U %*% diag(d)

head(Psi)
```

```
               [,1]       [,2]        [,3]         [,4]
Alabama    -0.9756604  1.1220012 -0.43980366  0.154696581
Alaska     -1.9305379  1.0624269  2.01950027 -0.434175454
Arizona    -1.7454429 -0.7384595  0.05423025 -0.826264240
Arkansas    0.1399989  1.1085423  0.11342217 -0.180973554
California -2.4986128 -1.5274267  0.59254100 -0.338559240
Colorado   -1.4993407 -0.9776297  1.08400162  0.001450164
```

```
#d)

install.packages(c("FactoMineR"))

library("FactoMineR")

head(PCA(USArrests,graph=FALSE)$ind$coord)
```

```
                Dim.1      Dim.2       Dim.3        Dim.4
Alabama     0.9855659 -1.1333924  0.44426879  0.156267145
Alaska      1.9501378 -1.0732133 -2.04000333 -0.438583440
Arizona     1.7631635  0.7459568 -0.05478082 -0.834652924
Arkansas   -0.1414203 -1.1197968 -0.11457369 -0.182810896
California  2.5239801  1.5429340 -0.59855680 -0.341996478
Colorado    1.5145629  0.9875551 -1.09500699  0.001464887
```

```
#Exercice 32

data <- data.frame(CAMP = c(239,1003,682,2594), HOTEL = c(155,1556,1944,1124),
                   LOCA = c(129,1821,967,2176), RESI = c(0,1521,1333,1038))

rownames(data) <- c("AGRI","CADR","INAC","OUVR")

#1)
```

```
chisq <- chisq.test (data)
chisq
Pearson's Chi-squared test

data:  data
X-squared = 2067.9, df = 9, p-value < 2.2e-16

#2)
#tableau de contingence des frequences
sansT=data

cont=prop.table(sansT)
round(cont,digit=2)

CAMP HOTEL LOCA RESI
AGRI 0.01  0.01 0.01 0.00
CADR 0.05  0.09 0.10 0.08
INAC 0.04  0.11 0.05 0.07
OUVR 0.14  0.06 0.12 0.06

#vecteur ligne
r1 <- apply(cont,1,sum)
round(r1, digit=2)

AGRI CADR INAC OUVR
0.03 0.32 0.27 0.38

#vecteur colonne
c1 <- apply(cont,2,sum)
```

```
round(c1, digit=2)
```

```
CAMP HOTEL  LOCA  RESI
 0.25  0.26  0.28  0.21
```

```
#ajout des totaux
cont2<-cbind(cont,r1)
cont2<-rbind(cont2,c1)
cont2[5,5]=sum(c1)
colnames(cont2) <- c("CAMP","HOTEL","LOCA","RESI","TOTAL")
rownames(cont2) <- c("AGRI","CADR","INAC","OUVR","TOTAL")
round(cont2,digit=2)
```

```
  CAMP HOTEL LOCA RESI TOTAL
AGRI  0.01  0.01 0.01 0.00  0.03
CADR  0.05  0.09 0.10 0.08  0.32
INAC  0.04  0.11 0.05 0.07  0.27
OUVR  0.14  0.06 0.12 0.06  0.38
TOTAL 0.25  0.26 0.28 0.21  1.00
```

```
#matrice des profils lignes
L1 <- sweep(cont2[-5,],1,STAT=r1,FUN="/")
round(L1,digits=2)
```

```
CAMP HOTEL LOCA RESI TOTAL
AGRI 0.46  0.30 0.25 0.00     1
CADR 0.17  0.26 0.31 0.26     1
INAC 0.14  0.39 0.20 0.27     1
OUVR 0.37  0.16 0.31 0.15     1
```

#matrice des profils colonnes

C1 <- sweep(cont2[,-5],2,STAT=c1,FUN="/")

round(C1,digits=2)

CAMP HOTEL LOCA RESI

AGRI 0.05 0.03 0.03 0.00

CADR 0.22 0.33 0.36 0.39

INAC 0.15 0.41 0.19 0.34

OUVR 0.57 0.24 0.43 0.27

TOTAL 1.00 1.00 1.00 1.00

#3) AFC

res.ca <- CA (data, graph = FALSE)

print(res.ca)

Results of the Correspondence Analysis (CA)

The row variable has 4 categories; the column variable has 4 categories

The chi square of independence between the two variables is equal to 2067.911 (p-value = 0).

*The results are available in the following objects:

	name	description
1	"$eig"	"eigenvalues"
2	"$col"	"results for the columns"
3	"colcoord"	"coord. for the columns"
4	"colcos2"	"cos2 for the columns"
5	"colcontrib"	"contributions of the columns"
6	"$row"	"results for the rows"
7	"rowcoord"	"coord. for the rows"
8	"rowcos2"	"cos2 for the rows"
9	"rowcontrib"	"contributions of the rows"

10 "$call" "summary called parameters"

11 "$call$marge.col" "weights of the columns"

12 "$call$marge.row" "weights of the rows"

#Exercice 33

#1)

```
install.packages(c("ca"))
library(ca)
data(smoke)
smoke
```

```
none light medium heavy
SM    4     2      3     2
JM    4     3      7     4
SE   25    10     12     4
JE   18    24     33    13
SC   10     6      7     2
```

#2)a)

```
F <- smoke/sum(smoke)
r<-apply(F,1,sum)
r
```

```
c<-apply(F,2,sum)
c
```

```
Z <- (F-r%*%t(c))/r%*%t(c)
Z
```

```
none      light     medium      heavy
```

SM 0.1505216 -0.22020202 -0.1510264 0.4036364

JM -0.2969035 -0.28518519 0.2105735 0.7155556

SE 0.5509482 -0.15904139 -0.2675522 -0.3945098

JE -0.3528316 0.16969697 0.1673387 0.1404545

SC 0.2655738 0.02933333 -0.1283871 -0.3824000

#b)

U<-gsvd(Z,r,c)$U

V<-gsvd(Z,r,c)$V

d<-gsvd(Z,r,c)$d

X <- sweep(U,2,STAT=d,FUN="*") #coordonnees factorielles des profil-lignes

X

Y <- sweep(V,2,STAT=d,FUN="*") #coordonnees factorielles des profild-colonne

Y

Matrice des coordonnées factorielles des profils lignes AFC

	dim1	dim2	dim3
SM	-0.06576838	-0.19373700	0.070981028
JM	0.25895842	-0.24330457	-0.033705190
SE	-0.38059489	-0.01065991	-0.005155757
JE	0.23295191	0.05774391	0.003305371
SC	-0.20108912	0.07891123	-0.008081076

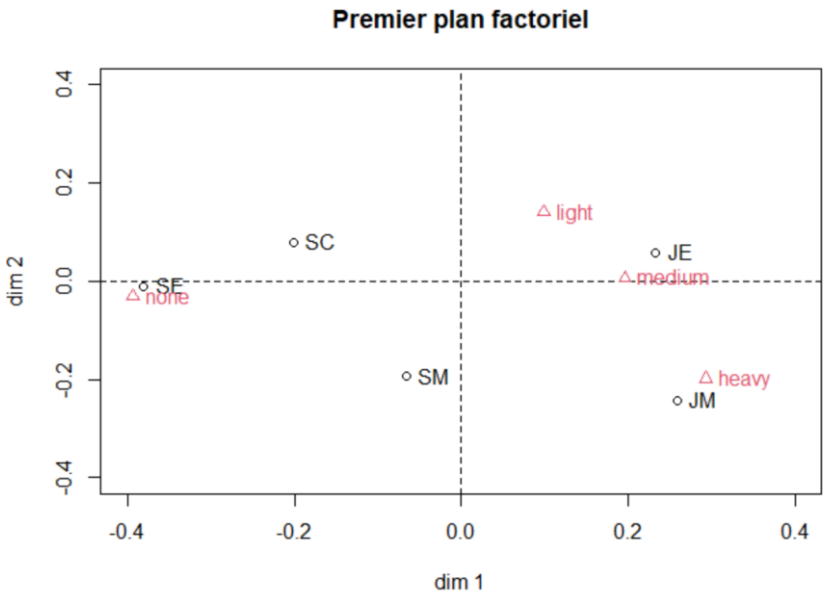
Matrice des coordonnées factorielles des profils colonnes AFC

	dim1	dim2	dim3
none	-0.39330845	-0.030492071	-0.0008904827
light	0.09945592	0.141064289	0.0219980349
medium	0.19632096	0.007359109	-0.0256590867
heavy	0.29377599	-0.197765656	0.0262108499

#c)

plot(X[,1:2],xlab="dim 1",ylab="dim 2",xlim=c(-0.4,0.4),ylim=c(-0.4,0.4),main="Premier plan factoriel")

```
abline(v = 0, lty = 2)
abline(h = 0, lty = 2)
text(X[,1:2],rownames(smoke),pos=4)
points(Y[,1:2],pch=2,col=2)
text(Y[,1:2],colnames(smoke),pos=4,col=2)
```




```
#d)le pourcentage

T <- sum(d^2)
d[1:2]^2/T*100

sum(d[1:2]^2/T)*100 #le pourcentage d'inertie du plan

#3)

?CA
res <- CA(smoke,graph=FALSE)
res$eig
```

head(resrowcoord)

head(rescolcoord)

?plot.CA

plot(res)

```
       eigenvalue percentage of variance cumulative percentage of variance
dim 1 0.0747591059             87.7558731                           87.75587
dim 2 0.0100171805             11.7586535                           99.51453
dim 3 0.0004135741              0.4854734                          100.00000
```

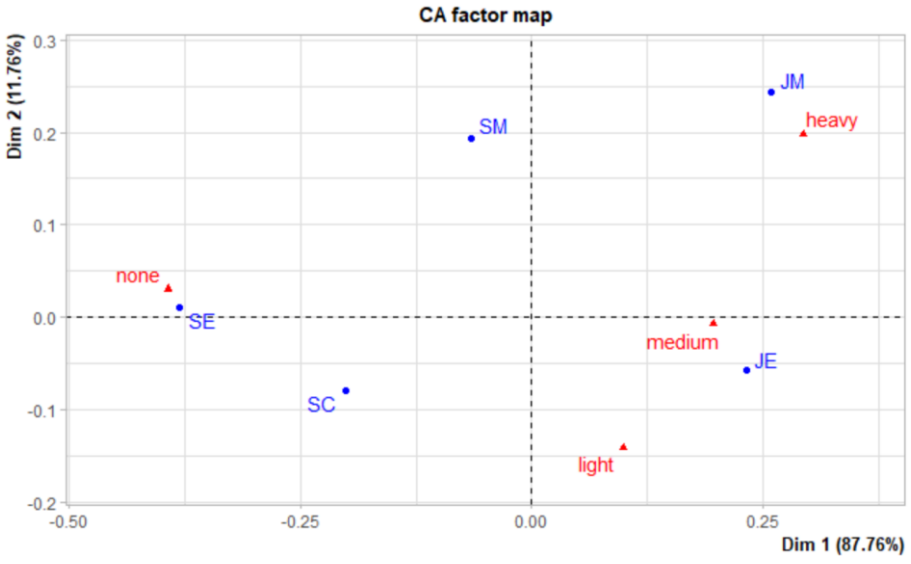
LA matrice X

```
          Dim 1       Dim 2        Dim 3
SM -0.06576838  0.19373700  0.070981028
JM  0.25895842  0.24330457 -0.033705190
SE -0.38059489  0.01065991 -0.005155757
JE  0.23295191 -0.05774391  0.003305371
SC -0.20108912 -0.07891123 -0.008081076
```

Matrice Y :

```
             Dim 1       Dim 2         Dim 3
none   -0.39330845  0.030492071 -0.0008904827
light   0.09945592 -0.141064289  0.0219980349
medium  0.19632096 -0.007359109 -0.0256590867
heavy   0.29377599  0.197765656  0.0262108499
```

Profil-lignes et les profil-colonnes sur le premier plan factoriel de l'AFC



Exercice 34

#1)

```
library(ca)
data <- read.csv(file="writers.csv", header = TRUE,row.names = 1)
data
```

```
CAMP HOTEL LOCA RESI
AGRI  239   155  129    0
CADR 1003  1556 1821 1521
INAC  682  1944  967 1333
OUVR 2594  1124 2176 1038
```

#2)

```
K <- data[1:15,]
chisq.test(K)
```

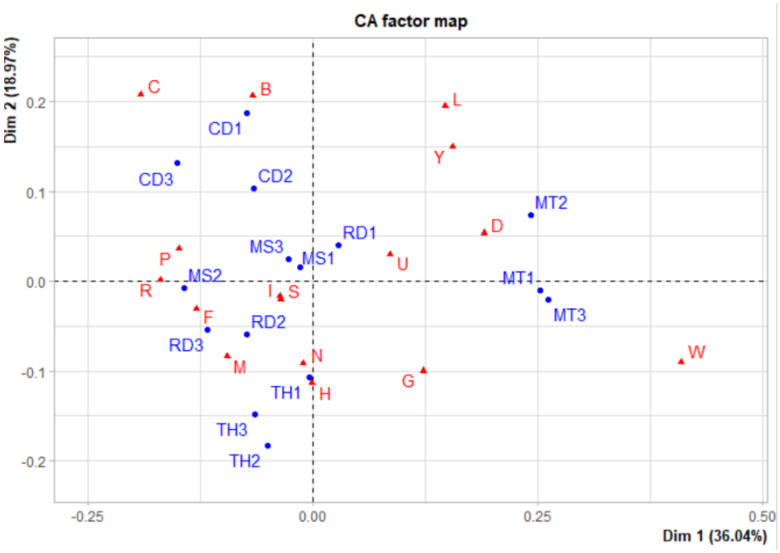
```
        Pearson's Chi-squared test

data:  K
X-squared = 455.18, df = 210, p-value < 2.2e-16
```

#3)

```
res <- CA(K)
```

```
res$eig[1:8,]
```

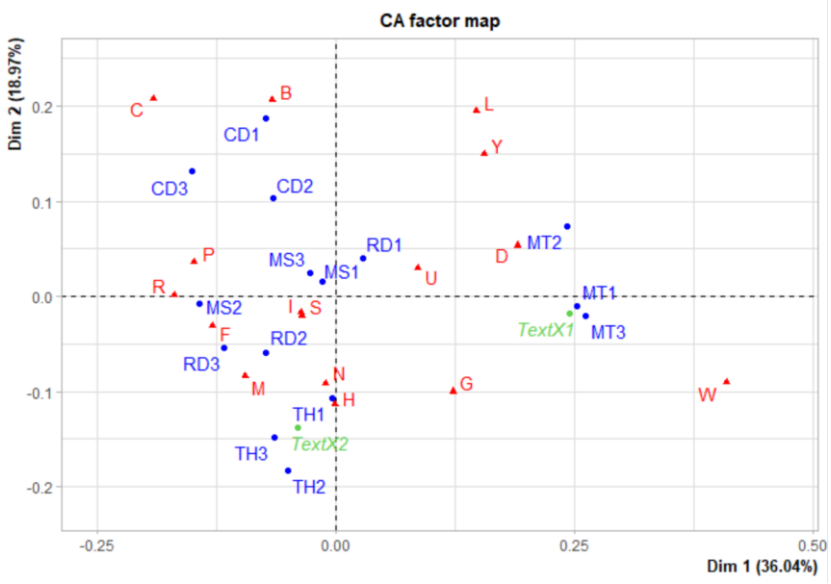
```
plot(res,axes=c(3,4))
```



#4)

res <- CA(data,row.sup=c(16,17),graph=FALSE)

plot(res,col.row.sup=3)



#5)

X <- rbind(resrowcoord[,1:4],res$row.sup$coord[,1:4])

D <- dist(X)

#CAH

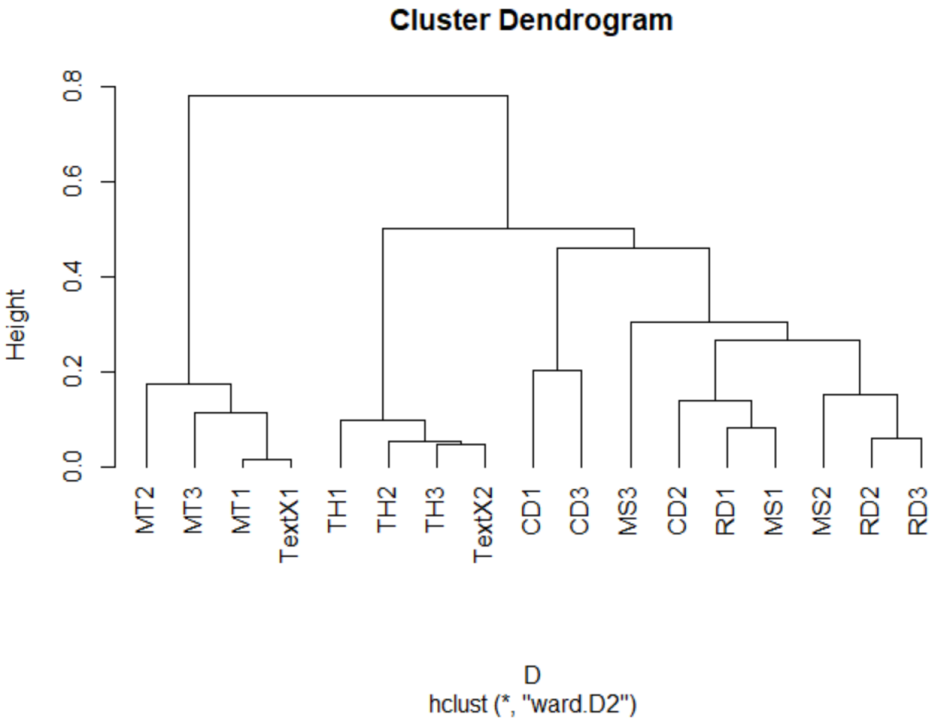
tree <- hclust(D,method="ward.D2")

#Dendrogramme

plot(tree,hang=-1)

#partition en 4 classes

cutree(tree,k=4)



Repartition en 4 classes :

```
CD1  CD2  CD3  RD1  RD2  RD3  TH1  TH2  TH3  MS1  MS2  MS3  MT1
  1    2    1    2    2    2    3    3    3    2    2    2    4
MT2  MT3 TextX1 TextX2
  4    4      4      3
```

TP 4 et 5

TP4 :

#1)

load("post-198636-chiens.rda")

head(chiens)

class(chiens)

	taille	poids	velocite	intellig	affect	agress	fonction
beauceron	T++	P+	V++	I+	Af+	Ag+	Utilite
basset	T-	P-	V-	I-	Af-	Ag+	Chasse
ber_allem	T++	P+	V++	I++	Af+	Ag+	Utilite
boxer	T+	P+	V+	I+	Af+	Ag+	Compagnie
bull-dog	T-	P-	V-	I+	Af+	Ag-	Compagnie
bull-mass	T++	P++	V-	I++	Af-	Ag+	Utilite

#2)

H <- subset(chiens,select=-fonction)

H

#3)

#a)

#tableau disjonctif complet

K <- tab.disjonctif(H)

K

#matrice des fréquences

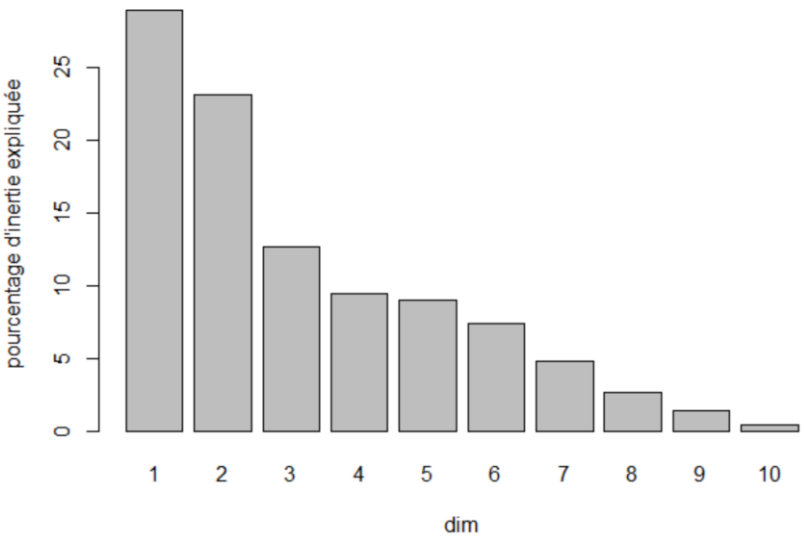
Freq <- K/sum(K)

#poids des lignes (r) et colonnes (c)

r<-apply(Freq,1,sum)

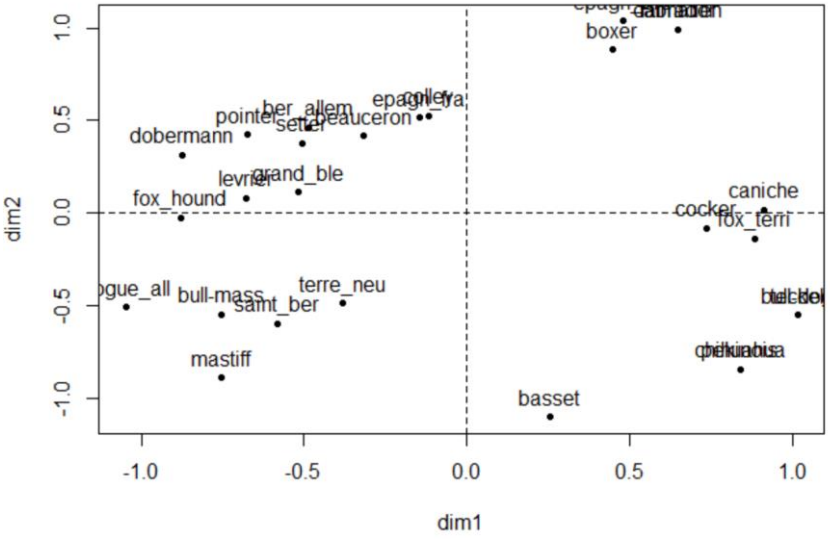
c<-apply(Freq,2,sum)

```
Z<-(Freq-r%*%t(c))/r%*%t(c)

U<-gsvd(Z,r,c)$U
V<-gsvd(Z,r,c)$V
d<-gsvd(Z,r,c)$d

X <- sweep(U,2,STAT=d,FUN="*")
Y <- sweep(V,2,STAT=d,FUN="*")

#b)

#inertie totale
p <- ncol(H)
m <- ncol(K)
m/p-1
#somme des valeurs singulières
sum(d^2)
#c)
#nb dim
length(round(d^2,digit=3))
n<-27
min(n-1,m-p)
#d)
barplot(d^2/sum(d^2)*100,
      names.arg=1:length(d),
      xlab="dim",
      ylab="pourcentage d'inertie expliquée")
```



#e)

```
X <- data.frame(U[,1:3]%*%diag(d[1:3]))
rownames(X) <- rownames(H)
colnames(X) <- paste("dim", 1:3, sep = "")
round(X,digit=2)
Y <- data.frame(V[,1:3]%*%diag(d[1:3]))
rownames(Y) <- colnames(K)
colnames(Y) <- paste("dim", 1:3, sep = "")
round(Y,digit=2)
```



```
#f)

#Plan factoriel des individus
plot(X[,c(1,2)],pch=20)
abline(v = 0, lty = 2)
abline(h = 0, lty = 2)
text(X[,c(1,2)],labels=rownames(X),pos=3)

#Plan factoriel des modalités
plot(Y[,c(1,2)],pch=17)
abline(v = 0, lty = 2)
abline(h = 0, lty = 2)
text(Y[,c(1,2)],labels=rownames(Y),pos=3)

#g)
which(K[,1]==1)
moy <- apply(X[which(K[,1]==1),],2,mean)
moy*(1/d[1:3])
Y[1,]
```

```
 basset  bull-dog   caniche chihuahua fox_terri  pekinois    teckel
      2         5         7         8        17        22        26
```

```
#h)
eta2 <- function(x, gpe) {
  moyennes <- tapply(x, gpe, mean)
  effectifs <- tapply(x, gpe, length)
  varinter <- (sum(effectifs * (moyennes - mean(x)) ^ 2))
  vartot <- (var(x) * (length(x) - 1))
  res <- varinter / vartot
  return(res)
```

}

eta2(X$dim1,chiens$taille)

eta2(X$dim2,chiens$taille)

rapport de corrélation entre la variable taille avec la première composante principale :

0.8870733

rapport de corrélation entre la variable taille avec la deuxième composante principale :

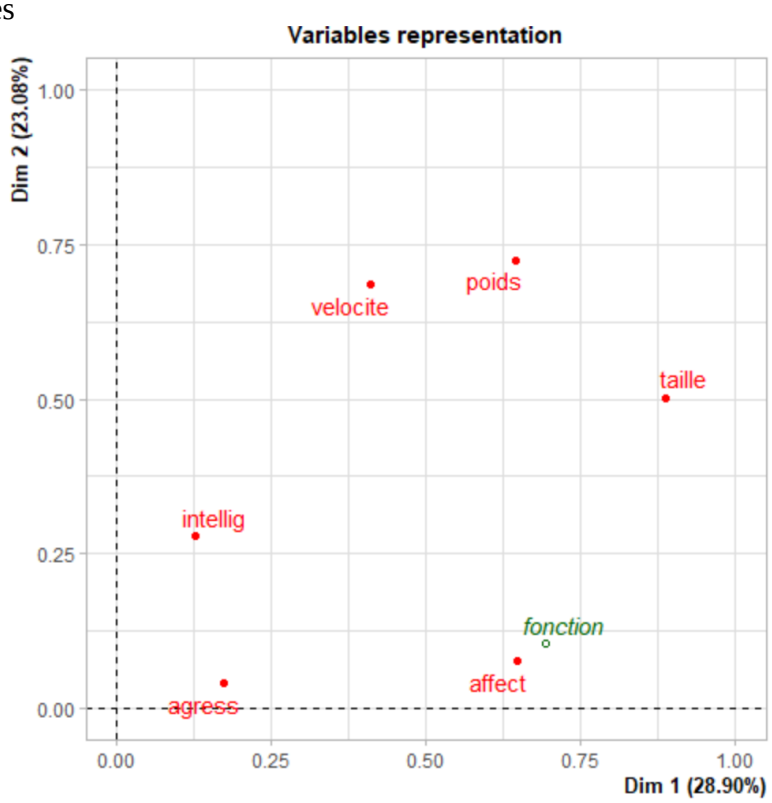
0.5024857

#4)

#a)

res <- MCA(chiens, quali.sup = 7)

res



#b)

head(X)

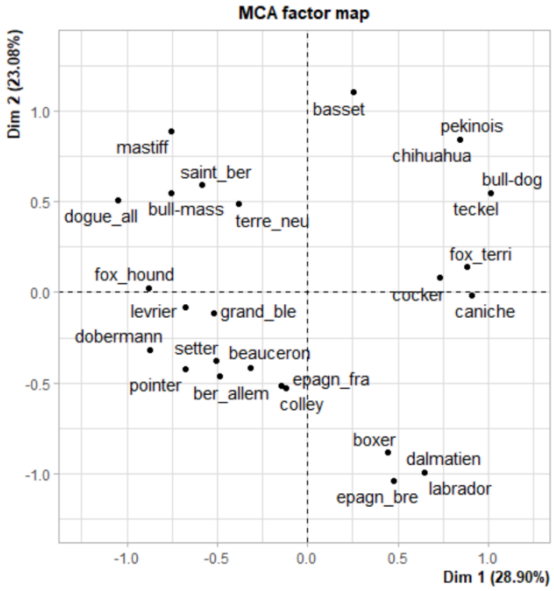
head(resindcoord)[,1:3]

head(resvarcoord)[,1:3]

plot(res,choix="ind",invisible=c("var","quali.sup"))

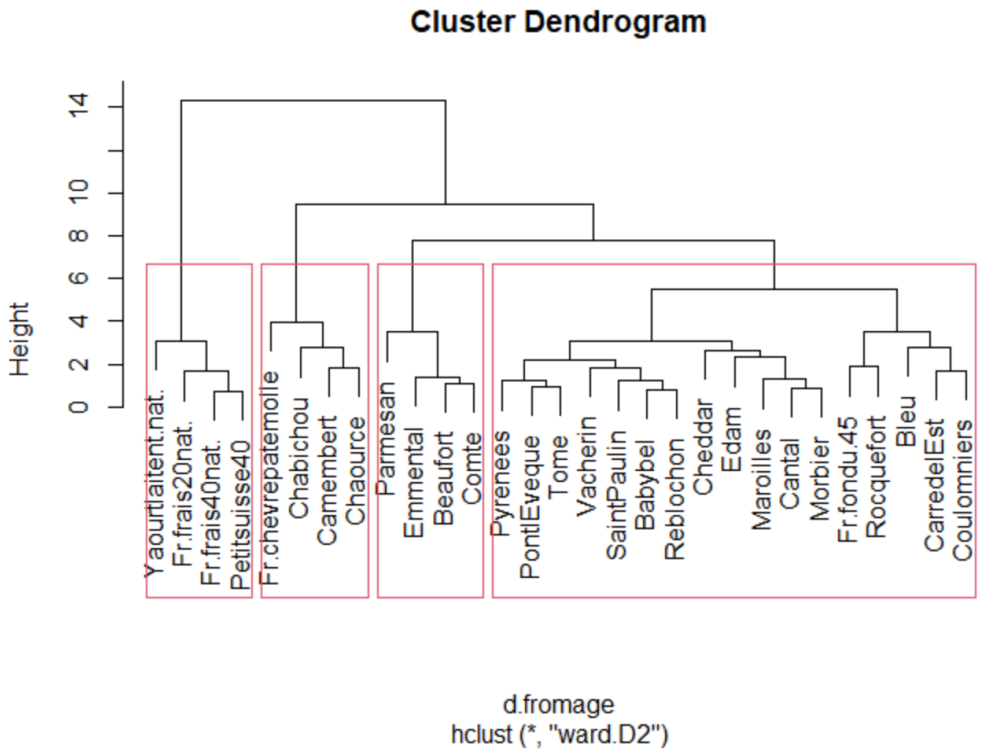
plot(res,choix="ind",invisible="ind")

```
                Dim 1      Dim 2      Dim 3
beauceron -0.3172001 -0.4177013 -0.1014677
basset     0.2541098  1.1012270 -0.1907010
ber_allem -0.4863955 -0.4644496 -0.4981339
boxer      0.4473649 -0.8817779  0.6920158
bull-dog   1.0133522  0.5498795 -0.1634232
bull-mass -0.7525745  0.5469118  0.4975731
```



TP5

```
#TP5

#Exercice 29


fromage2 <- read.delim("~/Downloads/post-198637-fromage2.txt", row.names=1)


fromage.cr <- scale(fromage2,center=T,scale=T)

d.fromage <- dist(fromage.cr)


#CAH

cah.ward <- hclust(d.fromage,method="ward.D2")

plot(cah.ward)

rect.hclust(cah.ward,k=4)

cahF <- cutree(cah.ward,k=4)

sort(groupes.cah)


#k-means

kmeansF <- kmeans(fromage.cr,centers=4,nstart=5)

kmeansF$cluster


#correspondance

table(cahF,kmeansF$cluster)
```



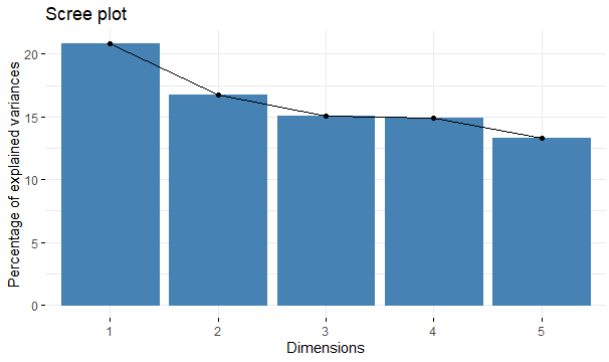
Après avoir réalisé le dendrogramme, on distingue un découpage en 4 groupes.

```
       CarredelEst            Babybel           Beaufort              Bleu
                 2                  1                  4                 1
         Camembert             Cantal          Chabichou           Chaource
                 2                  1                  2                 2
           Cheddar             Comte         Coulomniers             Edam
                 1                  4                  2                 4
         Emmental Fr.chevrepatemolle       Fr.fondu.45     Fr.frais20nat.
                 4                  2                  1                 3
    Fr.frais40nat.          Maroilles           Morbier          Parmesan
                 3                  1                  1                 4
     Petitsuisse40        PontlEveque          Pyrenees         Reblochon
                 3                  1                  1                 1
        Rocquefort        SaintPaulin              Tome          Vacherin
                 1                  1                  1                 1
Yaourtlaitent.nat.
                 3
```

Travail personnel

#Travail personnel couleur des cheveux et yeux :

```
library(factoextra)
library(FactoMineR)

dataTable<-read.csv("C:/Users/ProDesK/Desktop/yeux-cheveux-sexes data.csv")

str(dataTable)
data.frame':      592 obs. of  3 variables:
 $ cheveux: chr  "Noir" "Blond" "Noir" "Marron" ...
 $ yeux   : chr  "Marron" "Bleu" "Bleu" "Marron" ...
 $ sexe   : chr  "Male" "Femelle" "Male" "Femelle" ...

res.famd <- FAMD(dataTable, graph = FALSE)
get_famd_var(res.famd)
FAMD results for variables
 ===================================================
  Name       Description
1 "$coord"   "Coordinates"
2 "$cos2"    "Cos2, quality of representation"
3 "$contrib" "Contributions"

fviz_screeplot(res.famd)
```



#Test d'independance entre les deux caracteres .

test<-chisq.test(dataTable$cheveux,dataTable$yeux)

test	list [9] (S3: htest)	List of length 9
statistic	double [1]	138.2898
parameter	integer [1]	9
p.value	double [1]	2.325287e-25
method	character [1]	'Pearson\'s Chi-squared test'
data.name	character [1]	'dataTable$cheveux and dataTable$yeux'
observed	integer [4 x 4] (S3: table)	94 84 20 17 7 119 68 26 10 54 15 14 16 29 5 14 ...
expected	double [4 x 4]	46.12 103.87 39.22 25.79 47.20 106.28 40.14 26.39 19.95 44.93 16....
residuals	double [4 x 4] (S3: table)	7.050 -1.949 -3.069 -1.730 -5.851 1.233 4.398 -0.075 -2.228 1.353 ...
stdres	double [4 x 4] (S3: table)	9.968 -3.398 -4.254 -2.311 -8.328 2.164 6.137 -0.101 -2.738 2.050 ...

#Analyse factorielle

res.mca <- MCA (dataTable, graph = FALSE)

print(res.mca)

fviz_mca_biplot(res.mca)

Results of the Multiple Correspondence Analysis (MCA)

The analysis was performed on 592 individuals, described by 3 variables

*The results are available in the following objects:

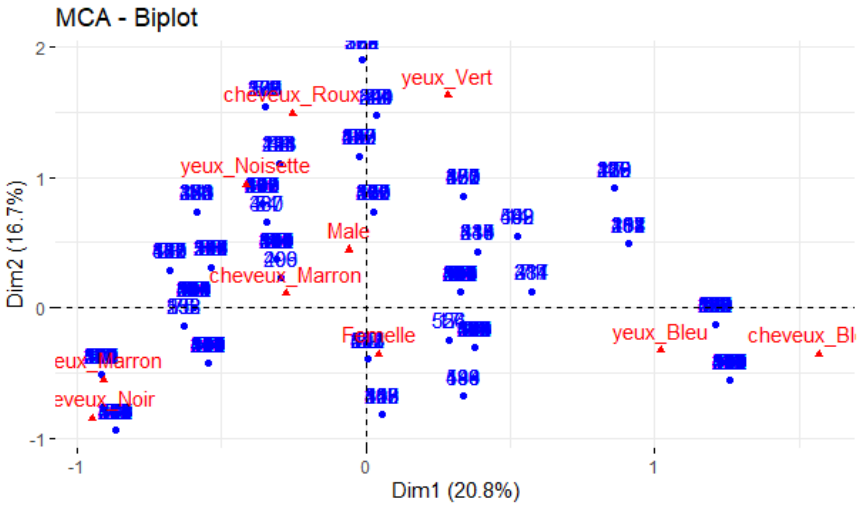
	name	description
1	"$eig"	"eigenvalues"
2	"$var"	"results for the variables"
3	"varcoord"	"coord. of the categories"
4	"varcos2"	"cos2 for the categories"
5	"varcontrib"	"contributions of the categories"
6	"varv.test"	"v-test for the categories"
7	"$ind"	"results for the individuals"
8	"indcoord"	"coord. for the individuals"
9	"indcos2"	"cos2 for the individuals"
10	"indcontrib"	"contributions of the individuals"

11 "$call" "intermediate results"

12 "$call$marge.col" "weights of columns"

13 "$call$marge.li" "weights of rows"

fviz_mca_biplot(res.mca)

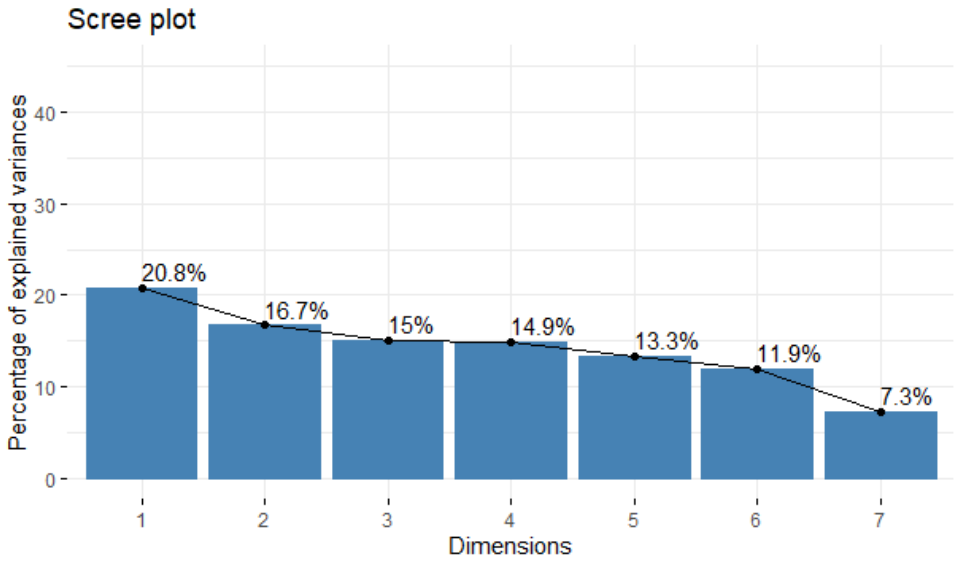


#valeurs propres

eig.val <- get_eigenvalue(res.mca)

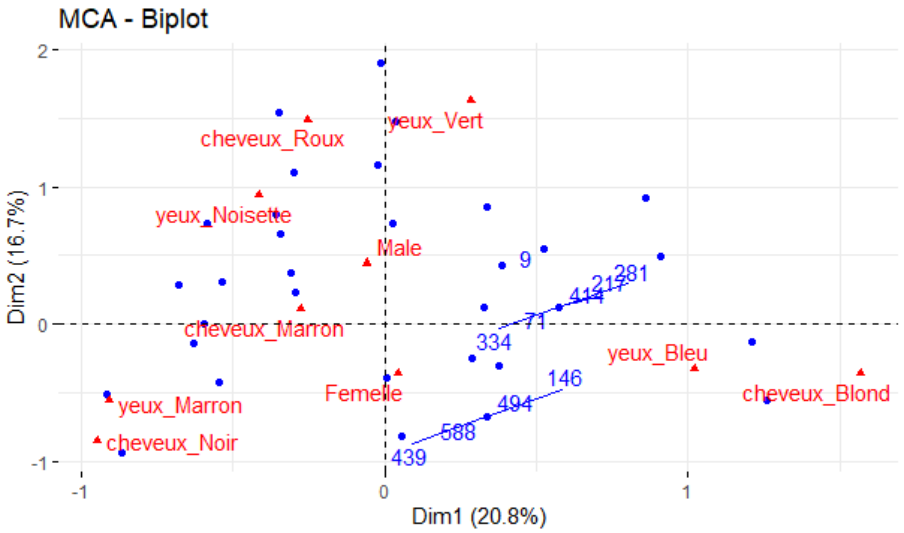
#visualisation de chaque pourcentage de variance

fviz_screeplot (res.mca, addlabels = TRUE, ylim = c (0, 45))



#visualisation du biplot des individus et des variables

fviz_mca_biplot (res.mca, repel = TRUE, ggtheme = theme_minimal())



On a utilisé L'ACM car on avait beaucoup de données qualitatives comme ca on peut bien faire notre analyse sur les données .