ProjetS4Scrptis

Jeros vigan

30/03/2020

#Gestionnaire de Packages

#Declaration du repertoire & Importation des fichiers

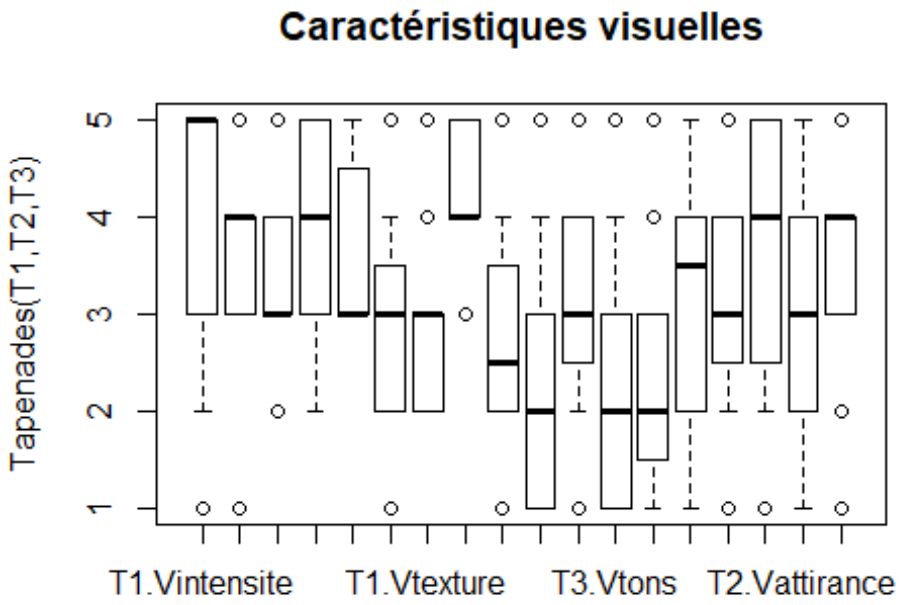
#verification des données

#Premières explorations

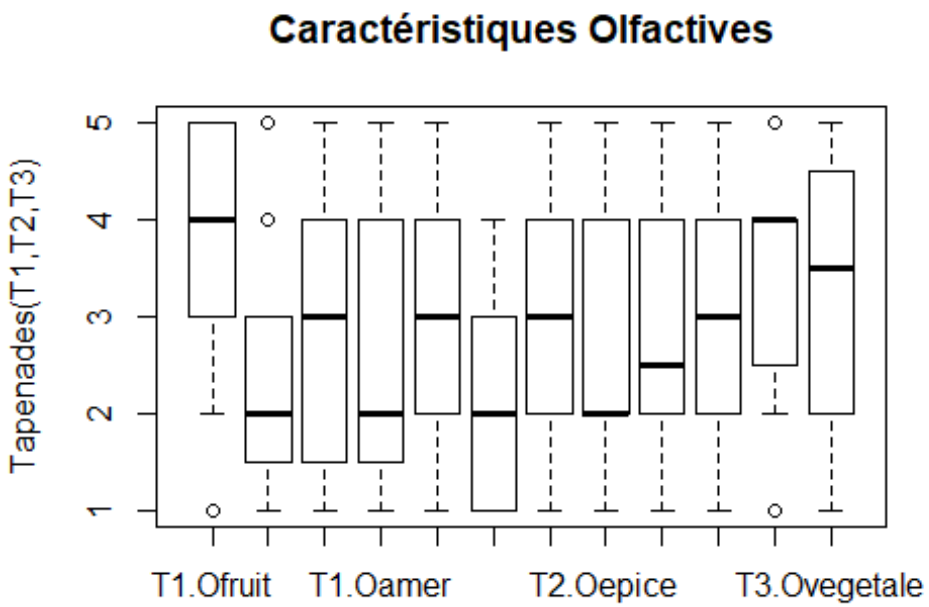
#UNIVARITE QUANTITATIVE

```
meancompCol=function(x) return(colMeans(x,na.rm=TRUE))
##variables quantitatives
#Histogramme
#Convertir tableau en data.frame
taData=data.frame(ini)

#Histogramme des variables quantitavies age
par(mfrow=c(1,2))
#boxplot(taData[,3], main ="BOXPLOT DE LA VARIABLE ÂGE", ylab = "Âge")
#hist(taData[,3], main ="HISTOGRAMME DE LA VARIABLE ÂGE", xlab = "Âge", ylab
= "Effectifs")
#legend("topright",paste(c("Min=","1Quartile=","Mediane=","Moyenne=","3Quarti
le=","Max="), summary(taData[,3],digits = 3)),pch="+", cex = 0.6)



#FONCTION HIST_bOX
#HistBoxTable=function(donneesTable){
#for(i in 1:ncol(donneesTable)) {
 # hist(donneesTable[,i],main = paste("HISTOGRAMME DE  LA
VARIABLE",str_to_upper(colnames((donneesTable)[i]))),xlab=colnames((donneesTa
ble)[i]),ylab="notes");legend("topleft",
paste(c("Min=","1Quartile=","Mediane=","Moyenne=","3Quartile=","Max="),
summary(donneesTable[,i],digits = 3)), pch="+",cex =
0.6);summary(donneesTable[,i],digits = 3);
  #boxplot(donneesTable[,i],main =paste("BOXPLOT DE LA
VARIABLE",str_to_upper(colnames((donneesTable)[i]))),ylab =paste("variable
:",colnames((donneesTable)[i])) )}}



#caractéristiques visuelles
```

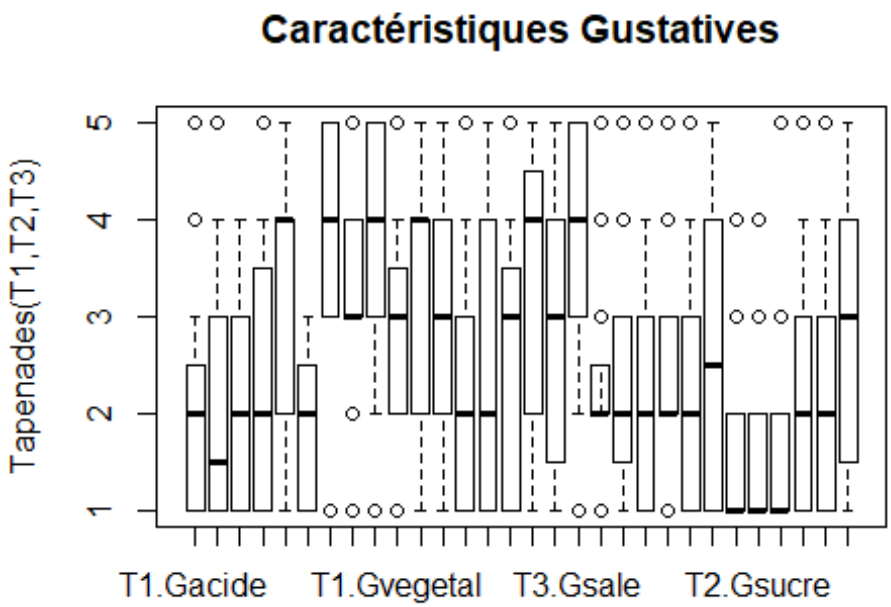
```r
par(mfrow=c(1,2))
#HistBoxTable(taData[,11:28])
par(mfrow=c(1,1))
boxplot(taData[,11:28], main="Caractéristiques visuelles",range = 0.5, ylab =
"Tapenades(T1,T2,T3)")
```



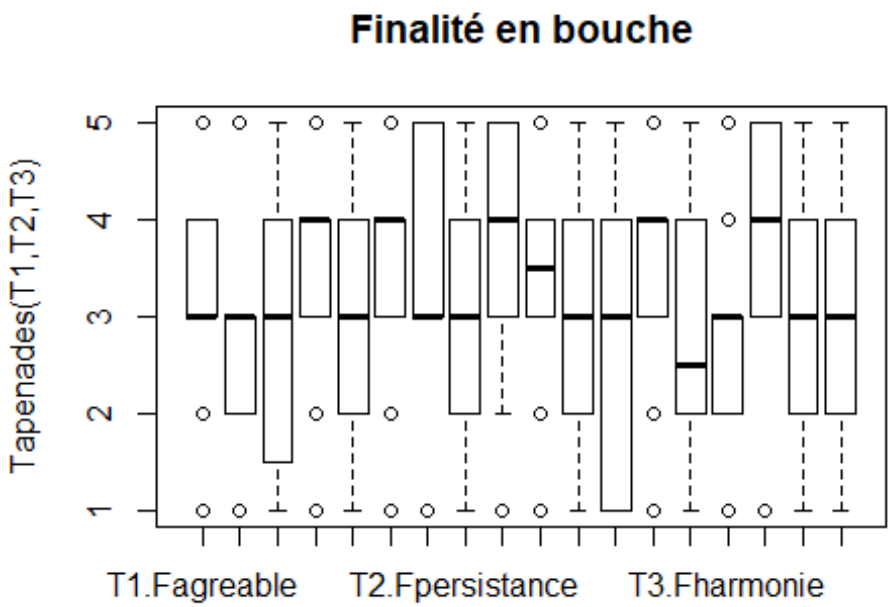

```r
#caractéristiques Olfactives
par(mfrow=c(1,2))
#HistBoxTable(taData[,29:40])
par(mfrow=c(1,1))
boxplot(taData[,29:40],main="Caractéristiques Olfactives",range = 0.5,ylab =
"Tapenades(T1,T2,T3)")
```




```
#Caractéristiques Gustatives
par(mfrow=c(1,2))
#HistBoxTable(taData[,41:70])
par(mfrow=c(1,1))
boxplot(taData[,41:70],main="Caractéristiques Gustatives",range = 0.5,ylab =
"Tapenades(T1,T2,T3)")
```




```r
#Finalité en bouche
par(mfrow=c(1,2))
#HistBoxTable(taData[,71:88])
par(mfrow=c(1,1))
boxplot(taData[,71:88],main="Finalité en bouche",range = 0.5,ylab =
"Tapenades(T1,T2,T3)")
```



#UNIVARITE

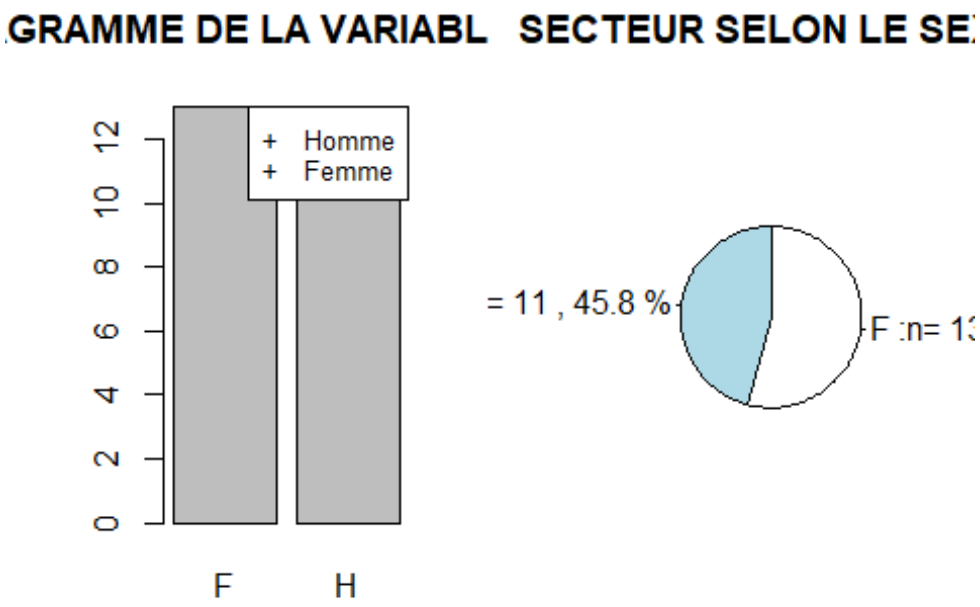
QUALITATIVE

```
##variables qualitatives
#fonction de frequence et effectifs
efreq=function(x)
{
    eff=table(x);seff=sum(eff);n=length(eff)
    effreq=cbind(rbind(eff,eff/seff),c(seff,1))

    dimnames(effreq)[[1]]=c("effectif","fréquence")
    dimnames(effreq)[[2]][n+1]="Total"

    return(round(effreq,digits=3))
}

#-------------------------Caractétistiques des Dégustateurs-----------------
--------------------
##quelle est le sexe dominant
kable(efreq(ini$sexe))
```

	F	H	Total
effectif	13.000	11.000	24
fréquence	0.542	0.458	1

```
par(mfrow=c(1,2))
barplot(table(ini$sexe),main = "DIAGRAMME DE LA VARIABLE SEXE")
legend("topright",paste(c("Homme","Femme")),pch="+", cex = 0.8)
```

```r
pieTable=function(x){
   eff=table(x)
   lbl=paste(levels(factor(x)),":n=",eff,",",round(100*eff/sum(eff),digits =
1),"%")
   pie(eff,main = "SECTEUR SELON LE SEXE",labels = lbl,init.angle =
90,clockwise = T)
}

pieTable(ini$sexe)
```

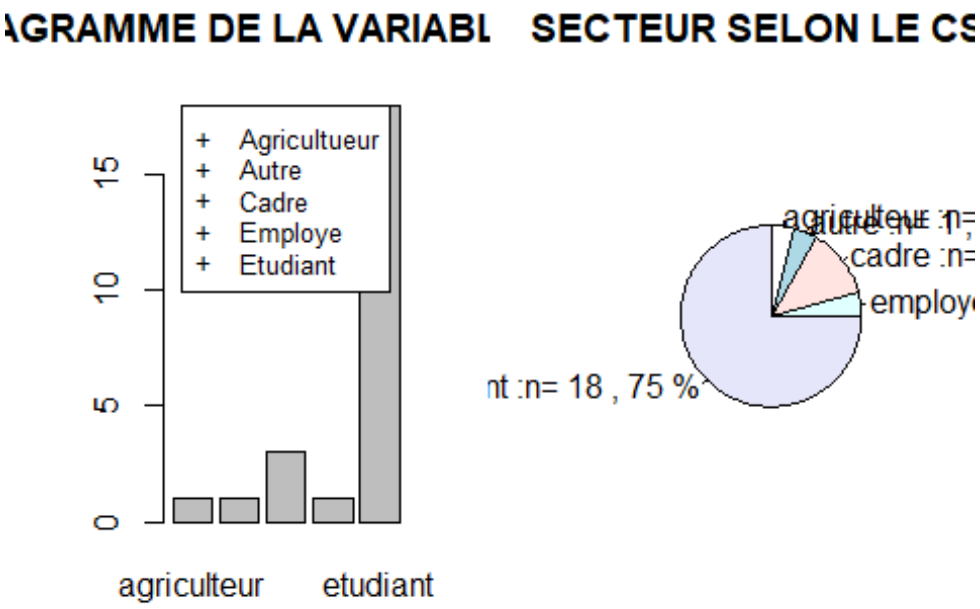



```r
#-------------------------------------------------------------------------
#Analyse : les participants sont plus les Femmes avec un pourcentage de 54,2%
#-------------------------------------------------------------------------


#-------------------------------------------------------------------------
##quelle est le CSP dominant
kable(efreq(ini$CSP))
```

	agriculteur	autre	cadre	employe	etudiant	Total
effectif	1.000	1.000	3.000	1.000	18.00	24
fréquence	0.042	0.042	0.125	0.042	0.75	1

```r
par(mfrow=c(1,2))
barplot(table(ini$CSP),main = "DIAGRAMME DE LA VARIABLE CSP")
```

```
legend("top",legend =
paste(c("Agricultueur","Autre","Cadre","Employe","Etudiant")),pch="+", cex =
0.8)

pieTable=function(x){
   eff=table(x)
   lbl=paste(levels(factor(x)),":n=",eff,",",round(100*eff/sum(eff),digits =
1),"%")
   pie(eff,main = "SECTEUR SELON LE CSP",labels = lbl,init.angle =
90,clockwise = T)
}

pieTable(ini$CSP)
```

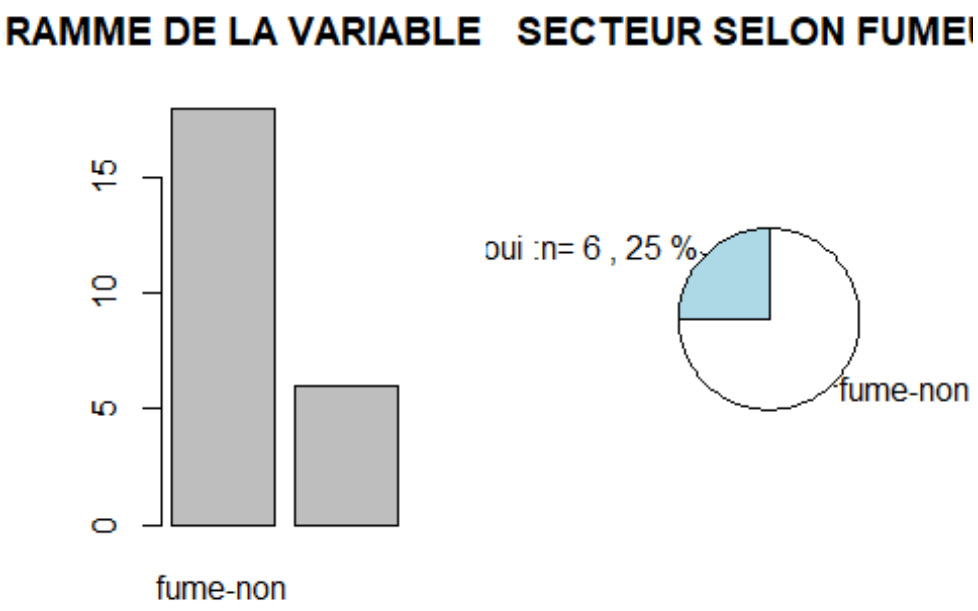



```
#-------------------------------------------------------------------------
#Analyse : les participants sont plus les etudiants avec un pourcentage de
75%
#-------------------------------------------------------------------------

#-------------------------------------------------------------------------

##quelle est fumeur oui OU non
kable(efreq(ini$fumeur))  # T1 est la plus préférée avec une part de 45,8%
```

	fume-non	fume-oui	Total

effectif	18.00	6.00	24
fréquence	0.75	0.25	1

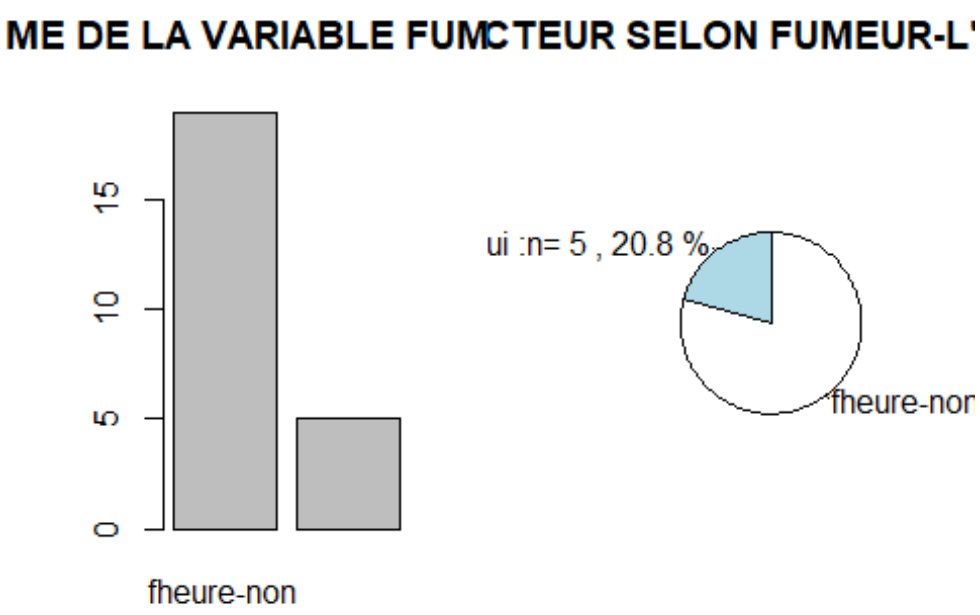
```
#diagramme
par(mfrow=c(1,2))
barplot(table(ini$fumeur),main = "DIAGRAMME DE LA VARIABLE FUMEUR")

pieTable=function(x){
   eff=table(x)
   lbl=paste(levels(factor(x)),":n=",eff,",",round(100*eff/sum(eff),digits =
1),"%")
   pie(eff,main = "SECTEUR SELON FUMEUR",labels = lbl,init.angle =
90,clockwise = T)
}

pieTable(ini$fumeur)
```




```
#-------------------------------------------------------------------------
#Analyse : la plupart des participants n'ont pas fumué avant la dégustation
#-------------------------------------------------------------------------


#-------------------------------------------------------------------------
##quelle est fumeurdeLheure oui OU non
kable(efreq(ini$fumedanslheure))  # T1 est la plus préférée avec une part de
45,8%
```

	fheure-non	fheure-oui	Total
effectif	19.000	5.000	24
fréquence	0.792	0.208	1

```
#diagramme
par(mfrow=c(1,2))
barplot(table(ini$fumedanslheure),main = "DIAGRAMME DE LA VARIABLE FUMEUR-
L'HEURE")

pieTable=function(x){
   eff=table(x)
   lbl=paste(levels(factor(x)),":n=",eff,",",round(100*eff/sum(eff),digits =
1),"%")
   pie(eff,main = "SECTEUR SELON FUMEUR-L'HEURE",labels = lbl,init.angle =
90,clockwise = T)
}

pieTable(ini$fumedanslheure)
```

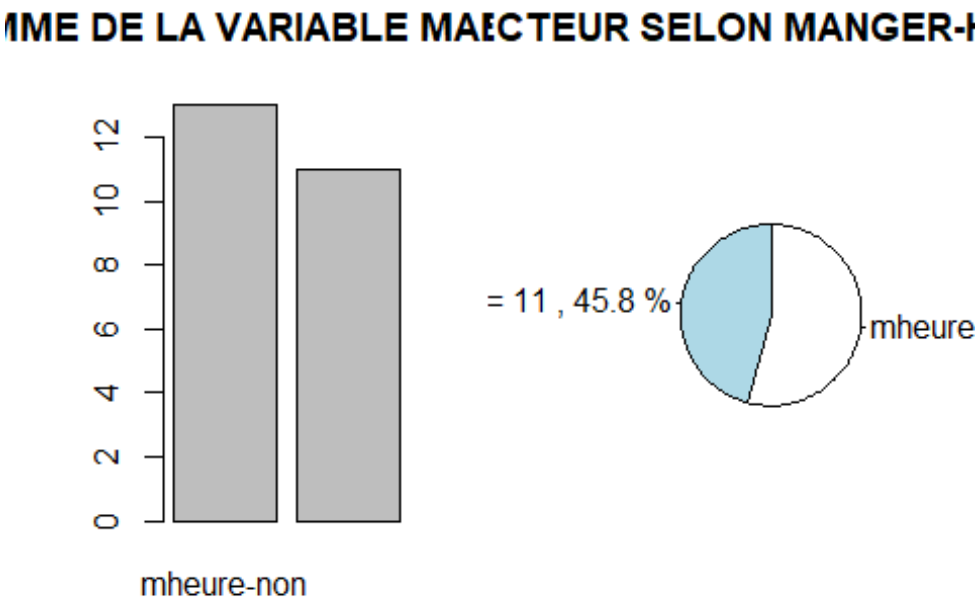



```
#-----------------------------------------------------------------------
#Analyse : la plupart des participants n'ont pas fumué avant la dégustation
#-----------------------------------------------------------------------


#-----------------------------------------------------------------------
##quelle est mange deLheure oui OU non
```

```
kable(efreq(ini$mangedanslheure))  # T1 est la plus préférée avec une part de 45,8%
```

	mheure-non	mheure-oui	Total
effectif	13.000	11.000	24
fréquence	0.542	0.458	1

```
#diagramme
par(mfrow=c(1,2))
barplot(table(ini$mangedanslheure),main = "DIAGRAMME DE LA VARIABLE MANGER-HEURE")

pieTable=function(x){
   eff=table(x)
   lbl=paste(levels(factor(x)),":n=",eff,",",round(100*eff/sum(eff),digits = 1),"%")
   pie(eff,main = "SECTEUR SELON MANGER-HEURE",labels = lbl,init.angle = 90,clockwise = T)
}

pieTable(ini$mangedanslheure)
```

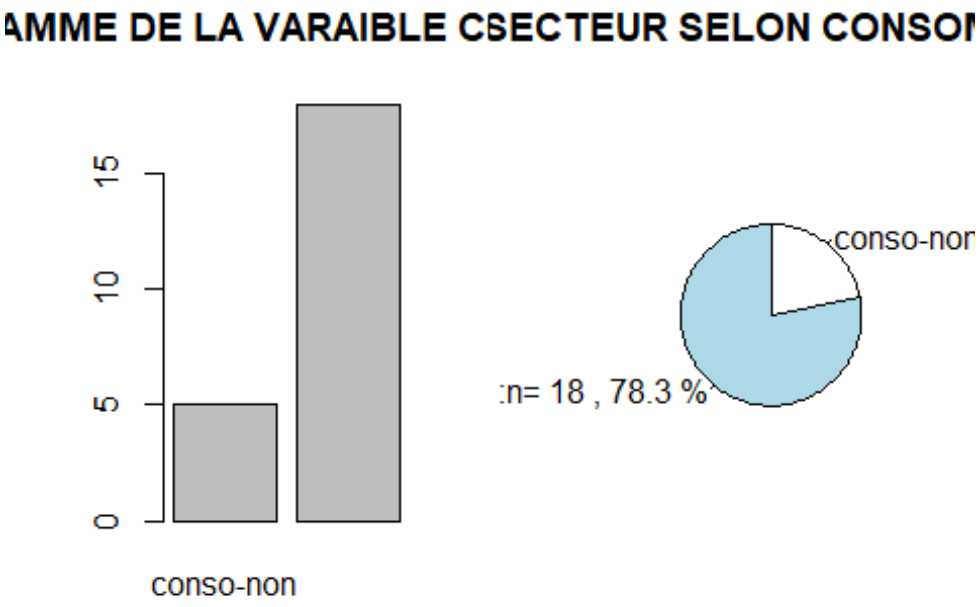



```
#-----------------------------------------------------------------------
#Analyse : la plupart des participants n'ont pas fumué avant la dégustation
#-----------------------------------------------------------------------
```

```r
#-----------------------------------------------------------------------
##quelle est consommé oui OU non
kable(efreq(ini$consomme))  # T1 est la plus préférée avec une part de 45,8%
```

	conso-non	conso-oui	Total
effectif	5.000	18.000	23
fréquence	0.217	0.783	1

```r
#diagramme
par(mfrow=c(1,2))
barplot(table(ini$consomme),main = "DIAGRAMME DE LA VARAIBLE CONSOMMÉ")

pieTable=function(x){
   eff=table(x)
   lbl=paste(levels(factor(x)),":n=",eff,",",round(100*eff/sum(eff),digits =
1),"%")
   pie(eff,main = "SECTEUR SELON CONSOMMÉ",labels = lbl,init.angle =
90,clockwise = T)
}

pieTable(ini$consomme)
```

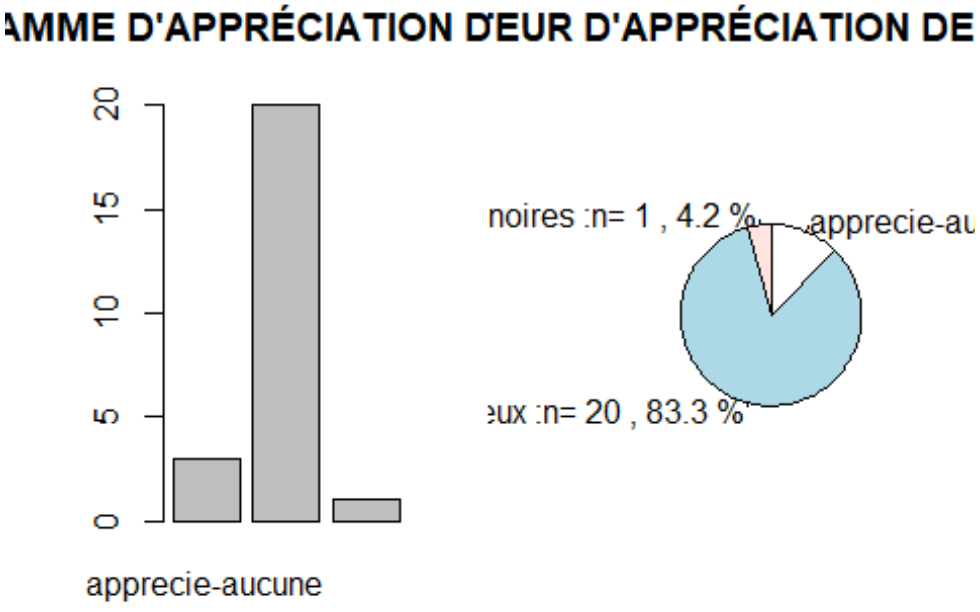



```r
#-----------------------------------------------------------------------
#Analyse : la plupart des participants n'ont pas fumué avant la dégustation
#-----------------------------------------------------------------------
```

```r
#-----------------------------------------------------------------------

##appreciation  des olives
kable(efreq(ini$apprecieOlives))
```

	apprecie-aucune	apprecie-les deux	apprecie-olives noires	Total
effectif	3.000	20.000	1.000	24
fréquence	0.125	0.833	0.042	1

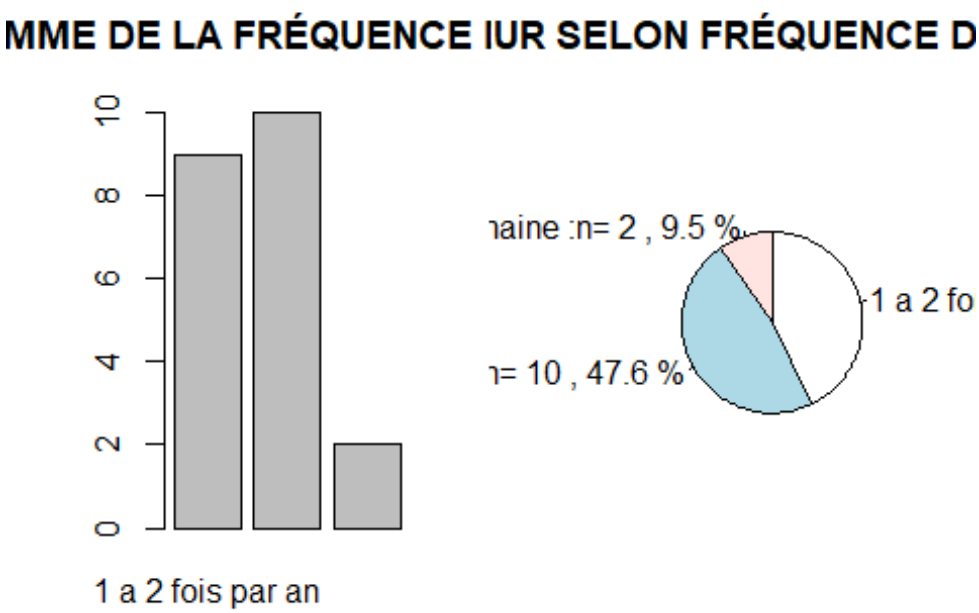
```r
#diagramme
par(mfrow=c(1,2))
barplot(table(ini$apprecieOlives),main = "DIAGRAMME D'APPRÉCIATION DES
OLIVES")

pieTable=function(x){
   eff=table(x)
   lbl=paste(levels(factor(x)),":n=",eff,",",round(100*eff/sum(eff),digits =
1),"%")
   pie(eff,main = "SECTEUR D'APPRÉCIATION DES OLIVES",labels = lbl,init.angle
= 90,clockwise = T)
}

pieTable(ini$apprecieOlives)
```



```r
#-----------------------------------------------------------------------
#Analyse : la plupart des participants apprecie les deux olives
#-----------------------------------------------------------------------



#-----------------------------------------------------------------------

##frequence des olives
kable(efreq(ini$frequenceOlives))
```

	1 a 2 fois par an	1 a 2 fois par mois	1 a 2 fois par semaine	Total
effectif	9.000	10.000	2.000	21
fréquence	0.429	0.476	0.095	1

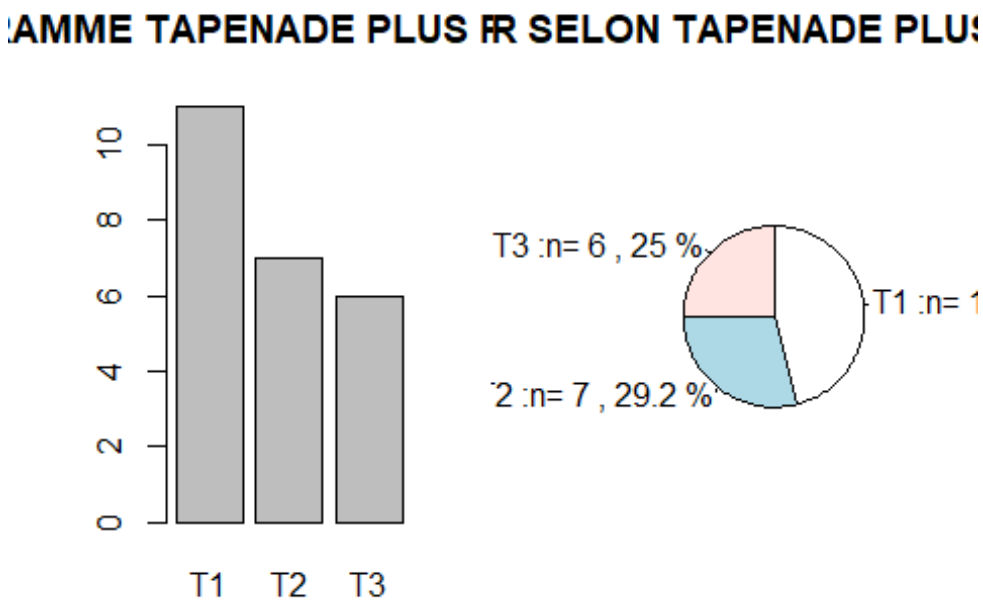
```r
#diagramme
par(mfrow=c(1,2))
barplot(table(ini$frequenceOlives),main = "DIAGRAMME DE LA FRÉQUENCE DES
OLIVES")

pieTable=function(x){
   eff=table(x)
   lbl=paste(levels(factor(x)),":n=",eff,",",round(100*eff/sum(eff),digits =
1),"%")
   pie(eff,main = "SECTEUR SELON FRÉQUENCE DES OLIVES",labels =
lbl,init.angle = 90,clockwise = T)
}

pieTable(ini$frequenceOlives)
```




```
#-------------------------------------------------------------------------
#Analyse : la plupart des participants mangent 1à 2 fois par mois les olives
#-------------------------------------------------------------------------


#-------------------------------------------------------------------------
##quelle est la tapenade préférée
kable(efreq(ini$Tap.plus))  # T1 est la plus préférée avec une part de 45,8%
```

	T1	T2	T3	Total
effectif	11.000	7.000	6.00	24
fréquence	0.458	0.292	0.25	1

```
#diagramme
par(mfrow=c(1,2))
barplot(table(ini$Tap.plus),main = "DIAGRAMME TAPENADE PLUS PRÉFÉRÉE")

pieTable=function(x){
   eff=table(x)
   lbl=paste(levels(factor(x)),":n=",eff,",",round(100*eff/sum(eff),digits =
1),"%")
   pie(eff,main = "SECTEUR SELON TAPENADE PLUS PRÉFÉRÉE",labels =
lbl,init.angle = 90,clockwise = T)
}

pieTable(ini$Tap.plus)
```




```
#-------------------------------------------------------------------------
#Analyse : T1 est la plus préférée avec un pourcentage de 54,2%
#-------------------------------------------------------------------------

#proportion de personnes qui préfèrent la Tapenade 1  et proportion de
personnes qui préfèrent la Tapenade 2 (un test d'égalité des proportions
11/24 et 7/24. )
#HO:les proportions sont égales(P1 = P2 ) contre H1 : P1 /= P2 (H1 : P1 >
P2.)

print(paste("Proportion de personnes ayant préféré la T1",11/24))

## [1] "Proportion de personnes ayant préféré la T1 0.458333333333333"

print(paste("Proportion de personnes ayant préféré la T2",7/24))

## [1] "Proportion de personnes ayant préféré la T2 0.291666666666667"

  mat<-matrix(c(11,7,13,17),2)
dimnames(mat) <- list(c("T1","T2") ,c("préférée","pas-préférée"))
mat

##    préférée pas-préférée
## T1       11           13
## T2        7           17

t1=prop.test(mat,alternative = "greater");t1$p.value<0.05
```

```
## [1] FALSE
```

```
#proportion de personnes qui préfèrent la Tapenade 1  et proportion de personnes qui préfèrent la Tapenade 3
#HO:les proportions sont égales(P1 = P3 ) contre H1 : P1 /= P3 (H1 : P1 > P3.)

print(paste("Proportion de personnes ayant préféré la T1",11/24))
```

```
## [1] "Proportion de personnes ayant préféré la T1 0.458333333333333"
```

```
print(paste("Proportion de personnes ayant préféré la T3",6/24))
```

```
## [1] "Proportion de personnes ayant préféré la T3 0.25"
```

```
mat<-matrix(c(11,6,13,18),2)
dimnames(mat) <- list(c("T1","T3") ,c("préférée","pas-préférée"))
mat
```

```
##    préférée pas-préférée
## T1       11           13
## T3        6           18
```

```
t2=prop.test(mat,alternative = "greater");t2$p.value<0.05
```

```
## [1] FALSE
```

```
#proportion de personnes qui préfèrent la Tapenade 3  et proportion de personnes qui préfèrent la Tapenade 2
#HO:les proportions sont égales(P3 = P2 ) contre H1 : P3 /= P2 (H1 : P3 > P2.)

print(paste("Proportion de personnes ayant préféré la T3",6/24))
```

```
## [1] "Proportion de personnes ayant préféré la T3 0.25"
```

```
print(paste("Proportion de personnes ayant préféré la T2",7/24))
```

```
## [1] "Proportion de personnes ayant préféré la T2 0.291666666666667"
```

```
  mat<-matrix(c(6,7,18,17),2)
dimnames(mat) <- list(c("T3","T2") ,c("préférée","pas-préférée"))
mat
```

```
##    préférée pas-préférée
## T3        6           18
## T2        7           17
```

```
t3=prop.test(mat,alternative = "greater"); t3$p.value<0.05
```

```
## [1] FALSE
```

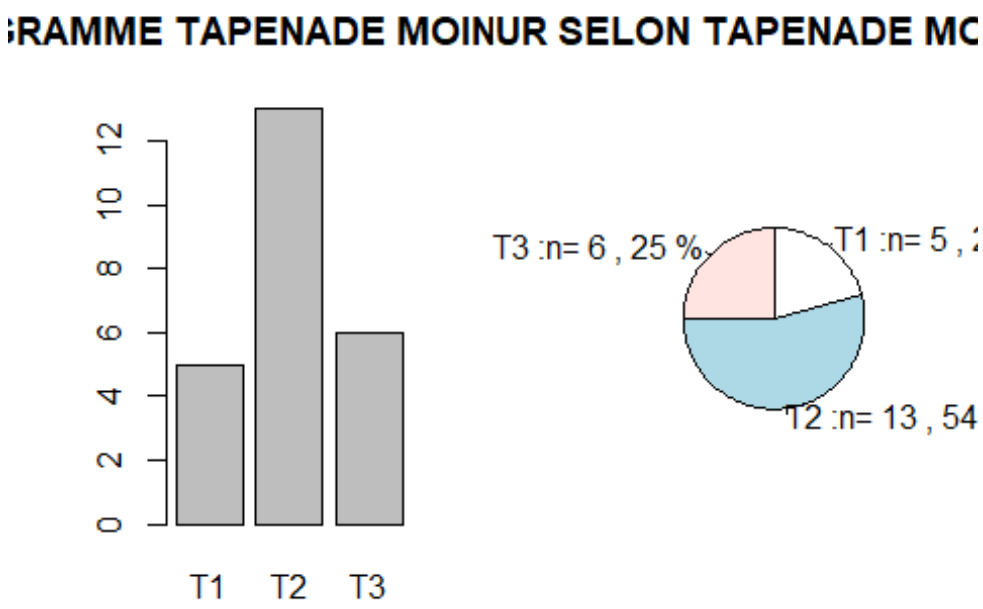
```r
#Resumer les resultats dans un tableau
tablePro =data.frame("Tapenades"=c("T1 & T2", "T1 & T3","T3 &
T2"),"Statistique de
test"=c(t1$statistic,t2$statistic,t3$statistic),"ddl"=c(t1$parameter,t2$param
eter,t3$parameter),"Pvalue"
=c(t1$p.value,t2$p.value,t3$p.value),"Significativité"=c(" non significatif à
5%", "non significatif à 5%"," non significatif à 5%") ,
"Conclusion"=c("Égalités","Égalités", "Égalités") )

kable(tablePro)
```

Tapenades	Statistique.de.test	ddl	Pvalue	Significativité	Conclusion
T1 & T2	0.800000	1	0.1855467	non significatif à 5%	Égalités
T1 & T3	1.457305	1	0.1136797	non significatif à 5%	Égalités
T3 & T2	0.000000	1	0.5000000	non significatif à 5%	Égalités

```r
#-------------------------------------------------------------------------
#quelle est la tapenade la moins aimée,
kable(efreq(ini$Tap.moins)) #T2 est la moins aimée avec une part de 54,2%
```

	T1	T2	T3	Total
effectif	5.000	13.000	6.00	24
fréquence	0.208	0.542	0.25	1

```r
# diagramme
par(mfrow=c(1,2))

barplot(table(ini$Tap.moins),main = "DIAGRAMME TAPENADE MOINS AIMÉE")

pieTable=function(x){
   eff=table(x)
   lbl=paste(levels(factor(x)),":n=",eff,",",round(100*eff/sum(eff),digits =
1),"%")
   pie(eff,main = "SECTEUR SELON TAPENADE MOINS AIMÉE",labels =
lbl,init.angle = 90,clockwise = T)
}

pieTable(ini$Tap.moins)
```




```
#-------------------------------------------------------------------------
#Analyse : T2 est la plus préférée avec un pourcentage de 54,2%
#-------------------------------------------------------------------------
#proportion de personnes qui aiment le moins la Tapenade 1  et proportion de
personnes qui aiment le moins la Tapenade 2  (un test d'égalité des
proportions 5/24 et 13/24. )
#HO:les proportions sont égales(P1 = P2 ) contre H1 : P1 /= P2 (H1 : P1 >
P2.)

print(paste("Proportion de personnes ayant moins aimée la T1",5/24))

## [1] "Proportion de personnes ayant moins aimée la T1 0.208333333333333"

print(paste("Proportion de personnes ayant moins aimée la T2",13/24))

## [1] "Proportion de personnes ayant moins aimée la T2 0.541666666666667"

  mat<-matrix(c(5,13,19,11),2)
dimnames(mat) <- list(c("T1","T2") ,c("moins aimée","pas-moins aimée"))
mat

##    moins aimée pas-moins aimée
## T1           5              19
## T2          13              11

t1=prop.test(mat,alternative = "greater");t1$p.value<0.05

## [1] FALSE
```

```r
#proportion de personnes qui aiment le moins la Tapenade 1  et proportion de personnes qui aiment le moins la Tapenade 3
#HO:les proportions sont égales(P1 = P3 ) contre H1 : P1 /= P3 (H1 : P1 > P3.)

print(paste("Proportion de personnes ayant moins aimée la T1",5/24))
```

```
## [1] "Proportion de personnes ayant moins aimée la T1 0.208333333333333"
```

```r
print(paste("Proportion de personnes ayant moins aimée la T3",6/24))
```

```
## [1] "Proportion de personnes ayant moins aimée la T3 0.25"
```

```r
  mat<-matrix(c(5,6,19,18),2)
dimnames(mat) <- list(c("T1","T3") ,c("moins aimée","pas-moins aimée"))
mat
```

```
##    moins aimée pas-moins aimée
## T1           5              19
## T3           6              18
```

```r
t2=prop.test(mat,alternative = "greater");t2$p.value<0.05
```

```
## [1] FALSE
```

```r
#proportion de personnes qui aiment le moins la Tapenade 2  et proportion de personnes qui aiment le moins la Tapenade 3
#HO:les proportions sont égales(P3 = P2 ) contre H1 : P3 /= P2 (H1 : P3 > P2.)

print(paste("Proportion de personnes ayant préféré la T3",6/24))
```

```
## [1] "Proportion de personnes ayant préféré la T3 0.25"
```

```r
print(paste("Proportion de personnes ayant préféré la T2",13/24))
```

```
## [1] "Proportion de personnes ayant préféré la T2 0.541666666666667"
```

```r
  mat<-matrix(c(6,13,18,11),2)
dimnames(mat) <- list(c("T3","T2") ,c("moins aimée","pas-moins aimée"))
mat
```

```
##    moins aimée pas-moins aimée
## T3           6              18
## T2          13              11
```

```r
t3=prop.test(mat,alternative = "greater"); t3$p.value<0.05
```

```
## [1] FALSE
```

```r
#Resumer les resultats dans un tableau
tablePro =data.frame("Tapenades"=c("T1 & T2","T1 & T3","T3 & T2"),"Statistique de
```

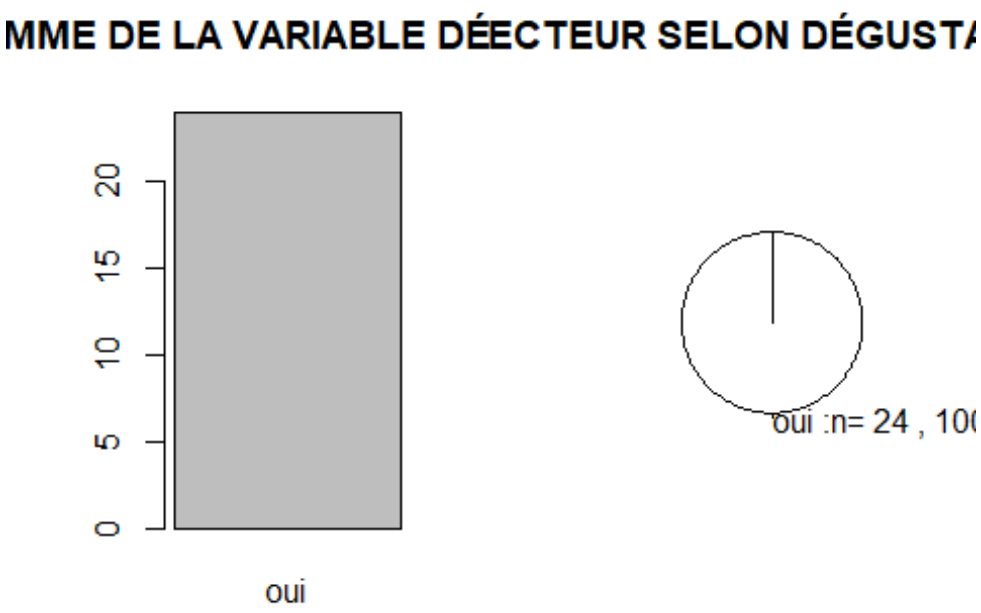
```
test"=c(t1$statistic,t2$statistic,t3$statistic),"ddl"=c(t1$parameter,t2$param
eter,t3$parameter),"Pvalue"
=c(t1$p.value,t2$p.value,t3$p.value),"Significativité"=c("non significatif à
5%", "non significatif à 5%"," non significatif à 5%") ,
"Conclusion"=c("Égalités","Égalités", "Égalités") )

kable(tablePro)
```

Tapenades	Statistique.de.test	ddl	Pvalue	Significativité	Conclusion
T1 & T2	4.355556	1	0.9815558	non significatif à 5%	Égalités
T1 & T3	0.000000	1	0.5000000	non significatif à 5%	Égalités
T3 & T2	3.136116	1	0.9617120	non significatif à 5%	Égalités

```
#-----------------------------------------------------------------------
#quelle est dégustation oui ou non ,
kable(efreq(ini[,97])) #T2 est la moins aimée avec une part de 54,2%
```

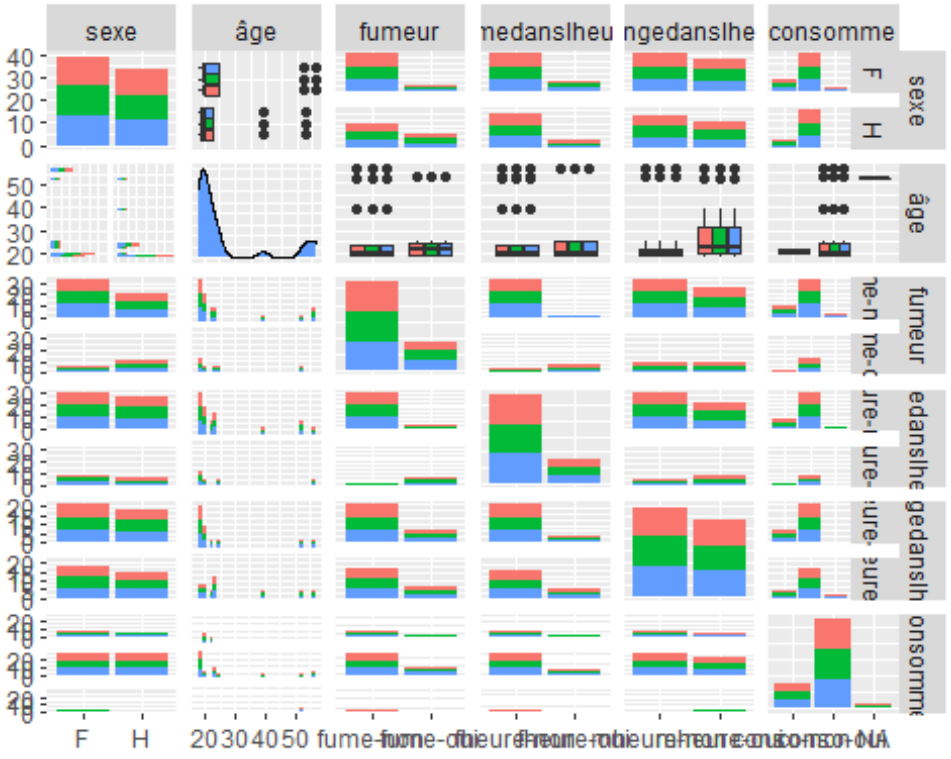
	oui	Total
effectif	24	24
fréquence	1	1

```
# diagramme
par(mfrow=c(1,2))


barplot(table(ini[,97]),main = "DIAGRAMME DE LA VARIABLE DÉGUSTATION")


pieTable=function(x){
   eff=table(x)
   lbl=paste(levels(factor(x)),":n=",eff,",",round(100*eff/sum(eff),digits =
1),"%")
   pie(eff,main = "SECTEUR SELON DÉGUSTATION",labels = lbl,init.angle =
90,clockwise = T)
}

pieTable(ini[,97])
```




```
#-------------------------------------------------------------------------
#Analyse : T2 est la plus préférée avec un pourcentage de 54,2%
#-------------------------------------------------------------------------


#Treemap
efreq1=function(x)
{
    eff=table(x);seff=sum(eff);n=length(eff)
    effreq1=cbind(rbind(eff,(eff/seff)*100))

    dimnames(effreq1)[[1]]=c("effectif","fréquence")

    return(round(effreq1,digits=))
}


#Caractéristiques générale des Dégustateurs
ggpairs(tableau[,c(2:3,5:8)], mapping = aes(color = tableau$Tapenade))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

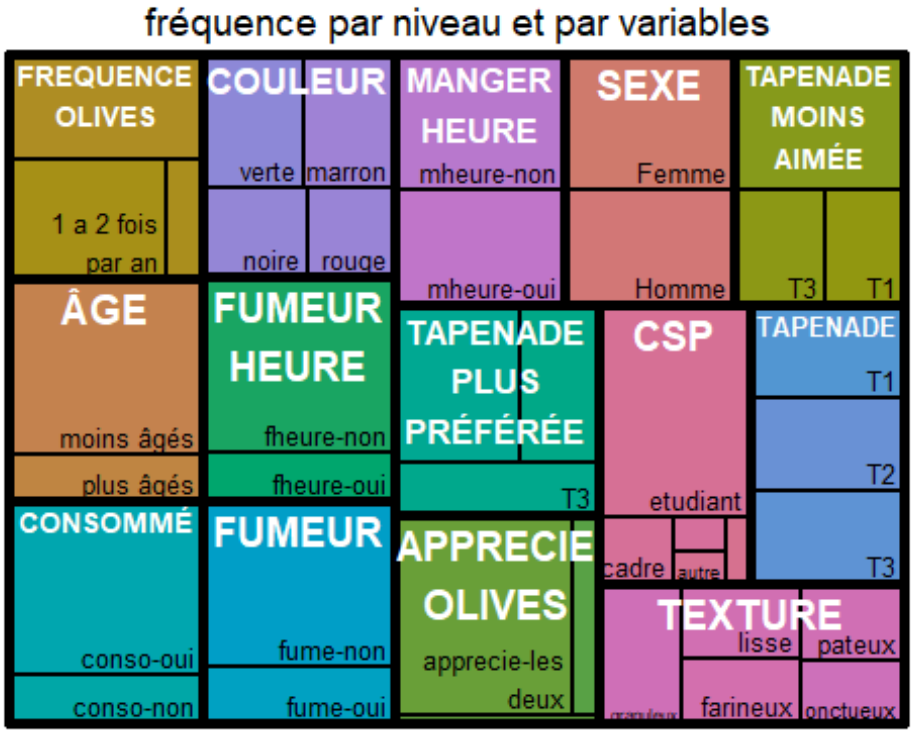



```
ggpairs(tableau[,c(4,9:10)], mapping = aes(color = tableau$Tapenade))
```


```
ggpairs(tableau[,38:41], mapping = aes(color = tableau$Tapenade))
```


```
#t=cbind(efreq1(tableau$sexe),efreq1(tableau$CSP),efreq1(tableau$fumeur),efreq1(tableau$fumedanslheure),efreq1(tableau$mangedanslheure),efreq1(tableau$consomme),efreq1(tableau$apprecieOlives),efreq1(tableau$frequenceOlives),efreq1(tableau$Vcouleur),efreq1(tableau$Gtexture),efreq1(tableau$Tap.moins),efreq1(tableau$Tap.plus),efreq1(tablo$decoupe_age),efreq1(tableau$Tapenade))

#write.table(t(t),file = "D:/Navigation/Téléchargements/Cours Distance/Projet S4/traitement/t.txt", row.names = TRUE,col.names =TRUE, sep = ";", dec = ".")

t11=read.csv("D:/Navigation/Téléchargements/Cours Distance/Projet S4/traitement/t.csv",sep = ";" , dec = ".",header = TRUE)

# treemap(t11,
#         index = c("Groupe","modalite"),
#         vSize = "effectif",
#         type = "index",
#         title="Effectif par niveau et par variables",
#         fontsize.title = 14
# )


# treemap(t11,
#         index = c("Groupe","modalite"),
#         vSize = "fréquence",
#         vColor = "modalite",
#         type = "depth",
```

```r
#          title="fréquence par niveau et par variables",
#         fontsize.title = 14
#
# )

# treemap(t11,
#         index = c("Groupe","modalite"),
#         vSize = "fréquence",
#         vColor = "modalite",
#         type = "categorical",
#          title="fréquence par niveau et par variables",
#         fontsize.title = 14
#
# )


treemap(t11,
        index = c("Groupe","modalite"),
        vSize = "fréquence",
        vColor = "modalite",
        type = "index",
         title="fréquence par niveau et par variables",
        fontsize.title = 14,
        fontsize.labels=c(15,10),
        fontcolor.labels=c("white","black"),
       fontface.labels=c(2,1),
      bg.labels=c("transparent"),
      align.labels=list(c("center", "top"), c("right", "bottom")),
    overlap.labels=0.5,
    inflate.labels=F,
    border.col=c("black","black"),
    border.lwds=c(5,2)

)
```



##BIVARIÉES #les graphiques: quanti-quali

```
# est-ce qu'il y a des différences de notes entre les tapenades (moyenne par
tapenade, boite et moustaches).

taData=data.frame(inid) #72 observations non centrées et ini pour les 24
observations

#Moyenne par groupe
meancomplet=function(x) return(mean(x,na.rm=TRUE))
meancompCol=function(x) return(colMeans(x,na.rm=TRUE))

#fonction
boxTable1=function(donneesTable){
for(i in 1:ncol(donneesTable)) {boxplot(donneesTable[,i]~taData$Tapenade,main
=paste(str_to_upper(colnames((donneesTable)[i])),"VISUELLE"),xlab
="TAPENADE",ylab =paste("VARIABLE
:",str_to_upper(colnames((donneesTable)[i]))) )}
}


#Caractéristiques visuelles
for (i in 11:28) print(paste("moyenne des valeurs
de",colnames(ini)[i],"=",meancomplet(ini[[i]])))
```

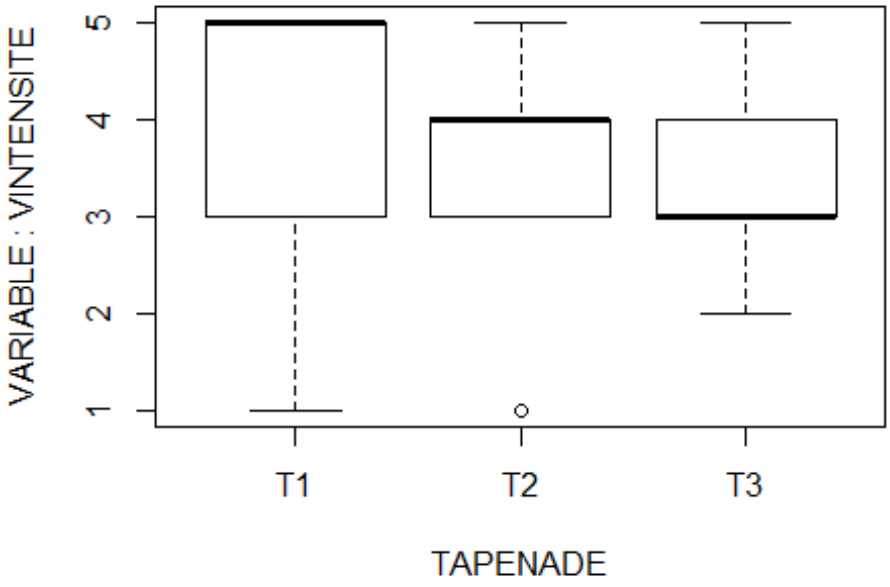
```
## [1] "moyenne des valeurs de T1.Vintensite = 4.04166666666667"
## [1] "moyenne des valeurs de T2.Vintensite = 3.68181818181818"
## [1] "moyenne des valeurs de T3.Vintensite = 3.0952380952381"
## [1] "moyenne des valeurs de T1.Vcompact = 3.83333333333333"
## [1] "moyenne des valeurs de T2.Vcompact = 3.70833333333333"
## [1] "moyenne des valeurs de T3.Vcompact = 2.875"
## [1] "moyenne des valeurs de T1.Vtexture = 3"
## [1] "moyenne des valeurs de T2.Vtexture = 4.125"
## [1] "moyenne des valeurs de T3.Vtexture = 2.79166666666667"
## [1] "moyenne des valeurs de T1.Vtons = 2.29166666666667"
## [1] "moyenne des valeurs de T2.Vtons = 3.125"
## [1] "moyenne des valeurs de T3.Vtons = 2.17391304347826"
## [1] "moyenne des valeurs de T1.Vbrillance = 2.375"
## [1] "moyenne des valeurs de T2.Vbrillance = 3.25"
## [1] "moyenne des valeurs de T3.Vbrillance = 3.25"
## [1] "moyenne des valeurs de T1.Vattirance = 3.5"
## [1] "moyenne des valeurs de T2.Vattirance = 2.95833333333333"
## [1] "moyenne des valeurs de T3.Vattirance = 3.625"
```

```
kable(data.frame("Moyenne"= meancompCol(ini[11:28])))
```

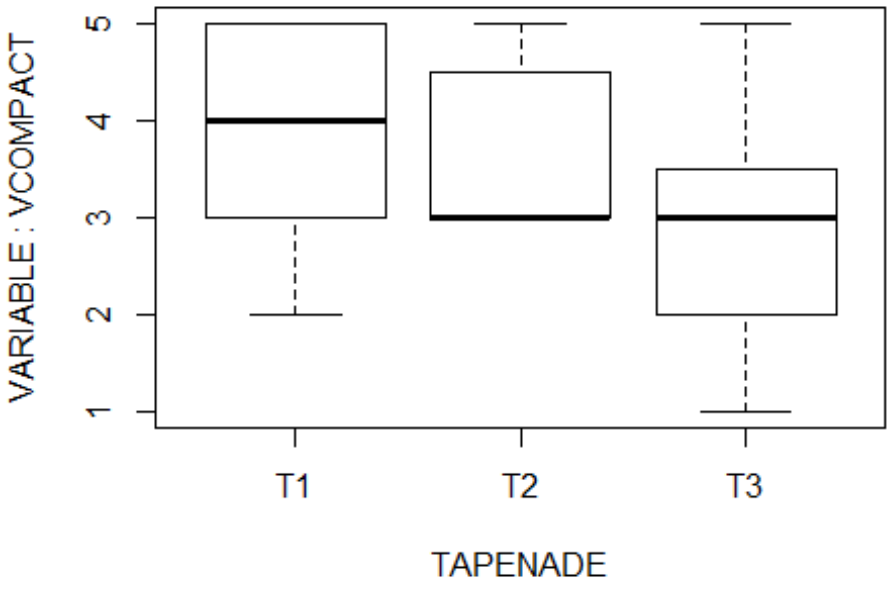
	Moyenne
T1.Vintensite	4.041667
T2.Vintensite	3.681818
T3.Vintensite	3.095238
T1.Vcompact	3.833333
T2.Vcompact	3.708333
T3.Vcompact	2.875000
T1.Vtexture	3.000000
T2.Vtexture	4.125000
T3.Vtexture	2.791667
T1.Vtons	2.291667
T2.Vtons	3.125000
T3.Vtons	2.173913
T1.Vbrillance	2.375000
T2.Vbrillance	3.250000
T3.Vbrillance	3.250000
T1.Vattirance	3.500000
T2.Vattirance	2.958333
T3.Vattirance	3.625000

```
par(mfrow=c(1,1))
boxTable1(taData[,12:17])
```

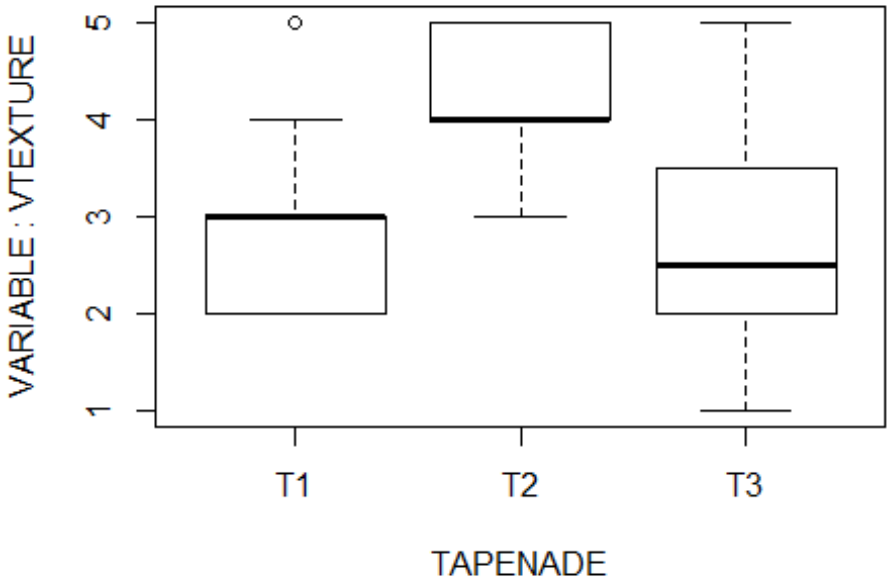
VINTENSITE VISUELLE



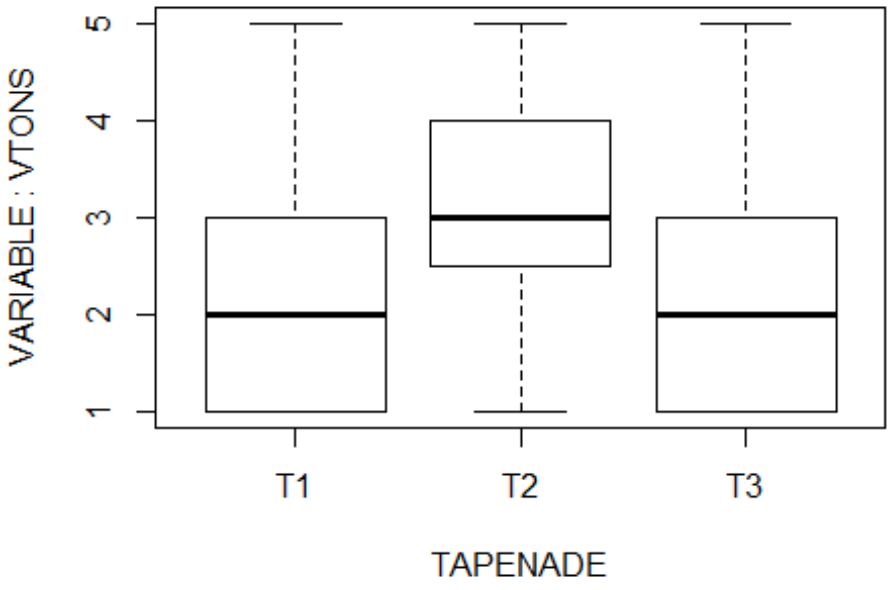
VCOMPACT VISUELLE

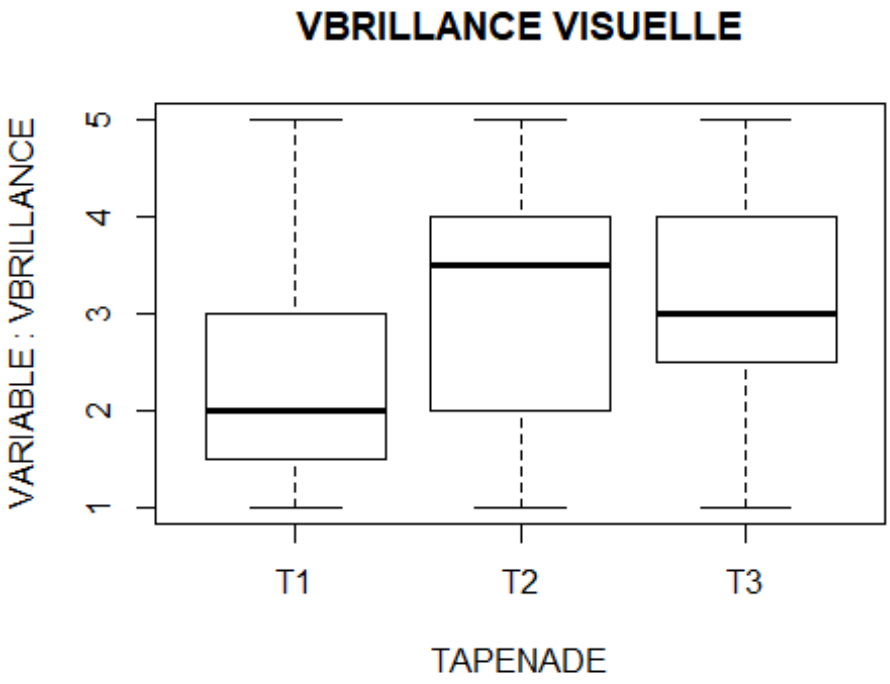


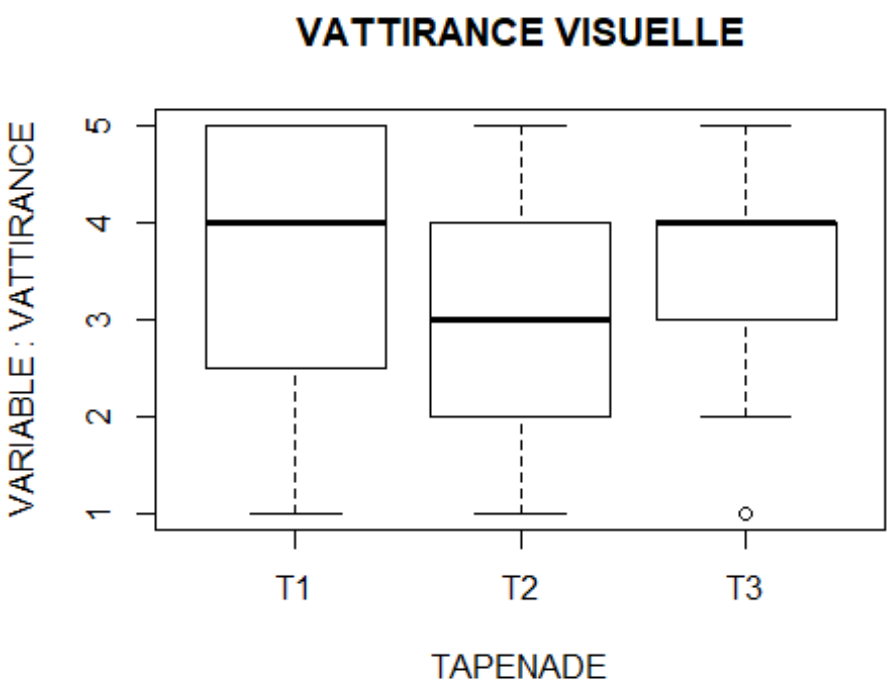
VTEXTURE VISUELLE



VTONS VISUELLE





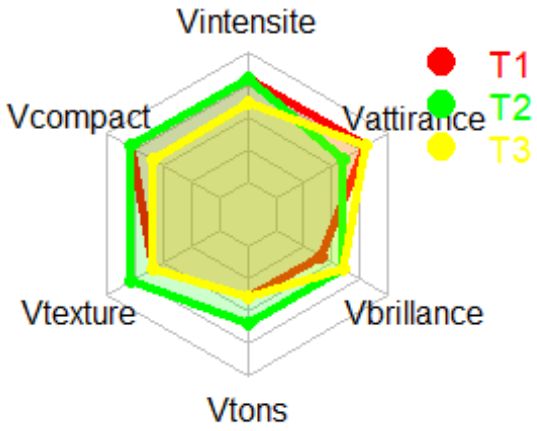



```r
#Radar
TapenadeT1=subset(taData[,12:17],taData$Tapenade=="T1")
TapenadeT2=subset(taData[,12:17],taData$Tapenade=="T2")
TapenadeT3=subset(taData[,12:17],taData$Tapenade=="T3")
```

```
clc=data.frame("T1"= meancompCol(TapenadeT1),"T2"=
meancompCol(TapenadeT2),"T3"=
meancompCol(TapenadeT3));clc=as.data.frame(t(clc));clc=round(clc,0)
clc=rbind(rep(5,6) , rep(0,6) , clc);clc
```

```
##    Vintensite Vcompact Vtexture Vtons Vbrillance Vattirance
## 1           5        5        5     5          5          5
## 2           0        0        0     0          0          0
## T1          4        4        3     2          2          4
## T2          4        4        4     3          3          3
## T3          3        3        3     2          3          4
```

```
#coul=brewer.pal(3, "BrBG"); colors_border=coul;colors_in =alpha(coul,0.5)
coul=c("red","green","yellow");colors_border=coul;colors_in =alpha(coul,0.2)

radarchart(clc,axistype=0, maxmin=T,
           pcol=colors_border , pfcol=colors_in , plwd=4 , plty=1,
           cglcol="grey", cglty=1, axislabcol="black", cglwd=0.8,
           vlcex=1,

           )

legend(x=1, y=1.2, legend = rownames(clc[-c(1,2),]), bty = "n", pch=20 ,
col=coul , text.col = coul, cex=1.1, pt.cex=3)
```



```
# #avec Plotrix
# radial.plot(clc, main="Test Radial Lines", line.col="red",lwd=3,
rad.col="lightblue")
# radial.plot(clc, rp.type="p", main="Test Polygon",
line.col="blue",labels=LETTERS[1:6])
# ggradar(clc)


#Caractéristiques Olfactives----------------------------------------
for (i in 29:40) print(paste("moyenne des valeurs
de",colnames(ini)[i],"=",meancomplet(ini[[i]])))
```

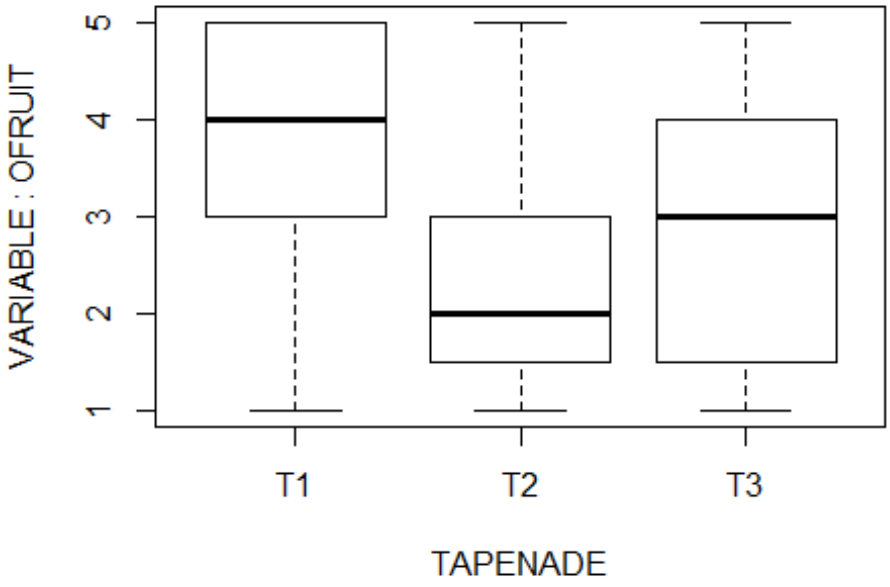
```
## [1] "moyenne des valeurs de T1.Ofruit = 3.625"
## [1] "moyenne des valeurs de T2.Ofruit = 2.375"
## [1] "moyenne des valeurs de T3.Ofruit = 2.83333333333333"
## [1] "moyenne des valeurs de T1.Oamer = 2.5"
## [1] "moyenne des valeurs de T2.Oamer = 3.08333333333333"
## [1] "moyenne des valeurs de T3.Oamer = 2"
## [1] "moyenne des valeurs de T1.Oepice = 3.125"
## [1] "moyenne des valeurs de T2.Oepice = 2.79166666666667"
## [1] "moyenne des valeurs de T3.Oepice = 2.91666666666667"
## [1] "moyenne des valeurs de T1.Ovegetale = 2.82608695652174"
## [1] "moyenne des valeurs de T2.Ovegetale = 3.33333333333333"
## [1] "moyenne des valeurs de T3.Ovegetale = 3.33333333333333"
```

```
kable(data.frame("Moyenne"= meancompCol(ini[29:40])))
```

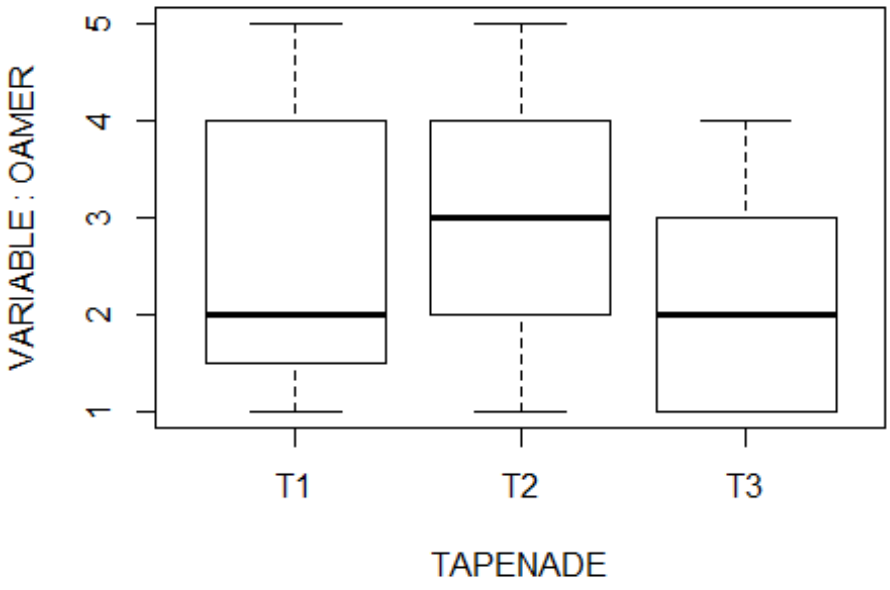
	Moyenne
T1.Ofruit	3.625000
T2.Ofruit	2.375000
T3.Ofruit	2.833333
T1.Oamer	2.500000
T2.Oamer	3.083333
T3.Oamer	2.000000
T1.Oepice	3.125000
T2.Oepice	2.791667
T3.Oepice	2.916667
T1.Ovegetale	2.826087
T2.Ovegetale	3.333333
T3.Ovegetale	3.333333

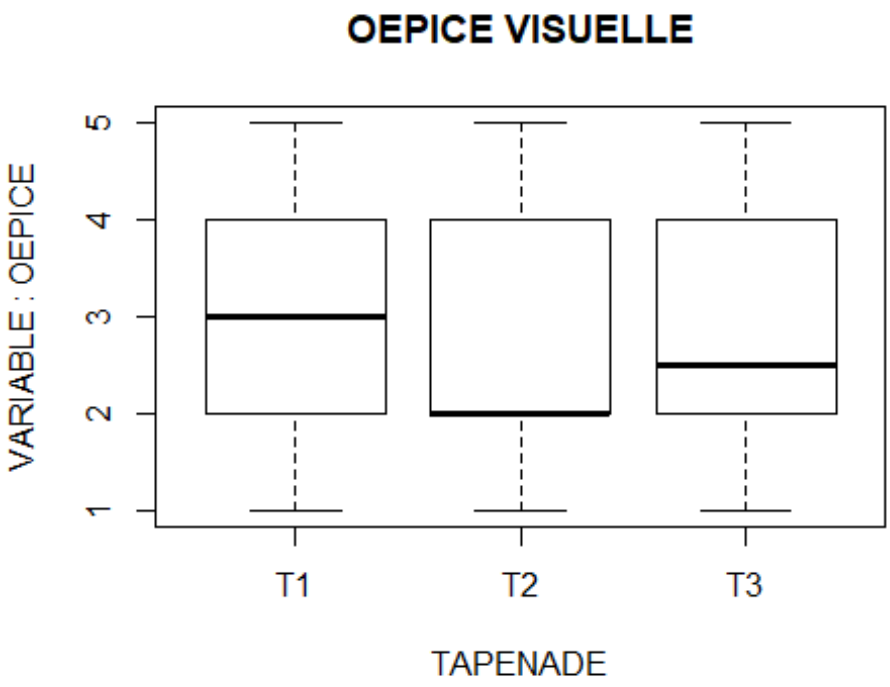
```
par(mfrow=c(1,1))
boxTable1(taData[,18:21])
```

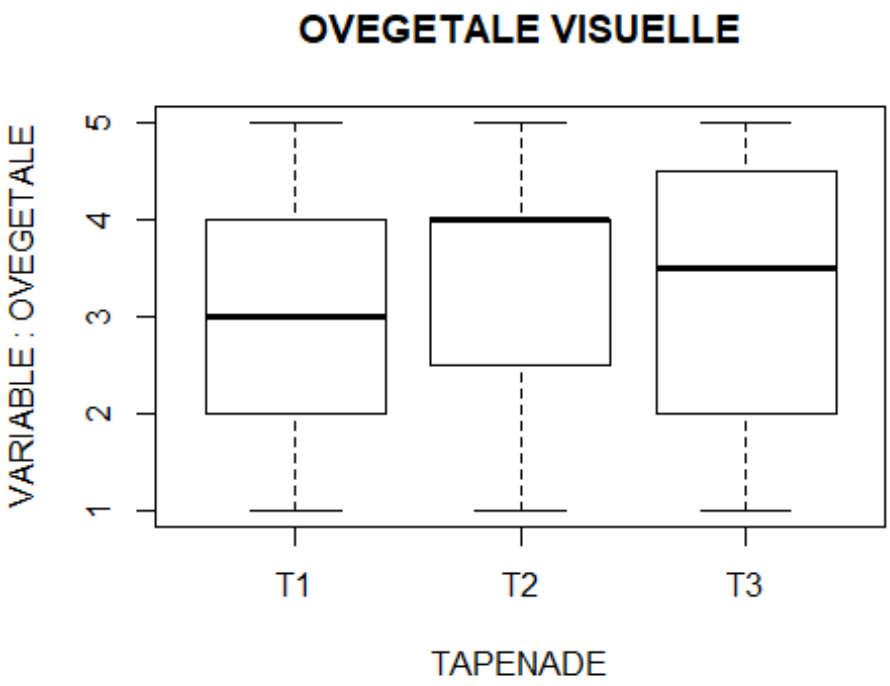
OFRUIT VISUELLE



OAMER VISUELLE





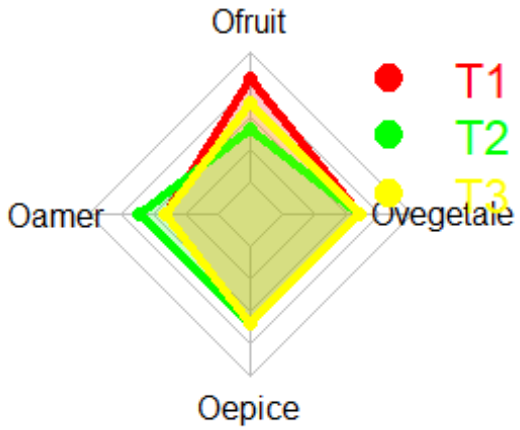



```
#Radar
TapenadeT1=subset(taData[,18:21],taData$Tapenade=="T1")
TapenadeT2=subset(taData[,18:21],taData$Tapenade=="T2")
TapenadeT3=subset(taData[,18:21],taData$Tapenade=="T3")
```

```
clc=data.frame("T1"= meancompCol(TapenadeT1),"T2"=
meancompCol(TapenadeT2),"T3"=
meancompCol(TapenadeT3));clc=as.data.frame(t(clc));clc=round(clc,0)
clc=rbind(rep(5,4) , rep(0,4) , clc);clc
```

```
##    Ofruit Oamer Oepice Ovegetale
## 1       5     5      5         5
## 2       0     0      0         0
## T1      4     2      3         3
## T2      2     3      3         3
## T3      3     2      3         3
```

```
radarchart(clc,axistype=0, maxmin=T,
           pcol=colors_border , pfcol=colors_in , plwd=4 , plty=1,
           cglcol="grey", cglty=1, axislabcol="black", cglwd=0.8,
           vlcex=1,

           )

legend(x=0.6, y=1.2, legend = rownames(clc[-c(1,2),]), bty = "n", pch=20 ,
col=coul , text.col = coul, cex=1.5, pt.cex=3)
```




```
#Caractéristiques Gustatives----------------------------------------------
------
for (i in 41:70) print(paste("moyenne des valeurs
de",colnames(ini)[i],"=",meancomplet(ini[[i]])))
```

```
## [1] "moyenne des valeurs de T1.Gacide = 2"
## [1] "moyenne des valeurs de T2.Gacide = 2.15"
## [1] "moyenne des valeurs de T3.Gacide = 2.33333333333333"
## [1] "moyenne des valeurs de T1.Gdurete = 2.45833333333333"
## [1] "moyenne des valeurs de T2.Gdurete = 3.22727272727273"
## [1] "moyenne des valeurs de T3.Gdurete = 1.83333333333333"
## [1] "moyenne des valeurs de T1.Gintensite = 3.8695652173913"
## [1] "moyenne des valeurs de T2.Gintensite = 3.30434782608696"
## [1] "moyenne des valeurs de T3.Gintensite = 3.70833333333333"
## [1] "moyenne des valeurs de T1.Gvegetal = 2.79166666666667"
## [1] "moyenne des valeurs de T2.Gvegetal = 3.39130434782609"
## [1] "moyenne des valeurs de T3.Gvegetal = 3.125"
## [1] "moyenne des valeurs de T1.Giode = 2.08333333333333"
## [1] "moyenne des valeurs de T2.Giode = 2.52173913043478"
## [1] "moyenne des valeurs de T3.Giode = 2.66666666666667"
## [1] "moyenne des valeurs de T1.Gsale = 3.375"
## [1] "moyenne des valeurs de T2.Gsale = 2.73913043478261"
## [1] "moyenne des valeurs de T3.Gsale = 3.70833333333333"
## [1] "moyenne des valeurs de T1.Gfruite = 2.39130434782609"
## [1] "moyenne des valeurs de T2.Gfruite = 2.25"
## [1] "moyenne des valeurs de T3.Gfruite = 2.34782608695652"
## [1] "moyenne des valeurs de T1.Gepice = 2.41666666666667"
## [1] "moyenne des valeurs de T2.Gepice = 2.25"
## [1] "moyenne des valeurs de T3.Gepice = 2.66666666666667"
## [1] "moyenne des valeurs de T1.Gsucre = 1.66666666666667"
## [1] "moyenne des valeurs de T2.Gsucre = 1.54166666666667"
## [1] "moyenne des valeurs de T3.Gsucre = 1.79166666666667"
## [1] "moyenne des valeurs de T1.Gamer = 2.33333333333333"
## [1] "moyenne des valeurs de T2.Gamer = 2.20833333333333"
## [1] "moyenne des valeurs de T3.Gamer = 2.83333333333333"
```

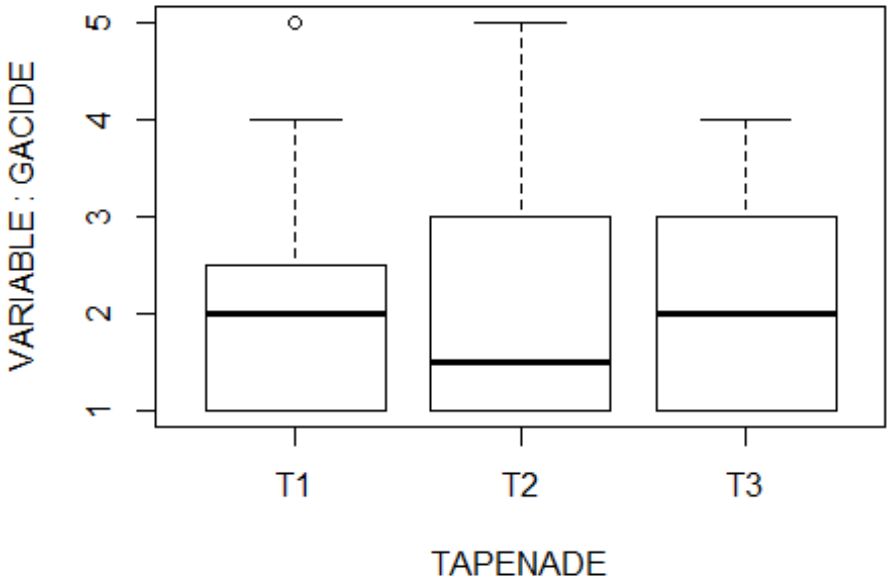
```
kable(data.frame("Moyenne"= meancompCol(ini[41:70])))
```

	Moyenne
T1.Gacide	2.000000
T2.Gacide	2.150000
T3.Gacide	2.333333
T1.Gdurete	2.458333
T2.Gdurete	3.227273
T3.Gdurete	1.833333
T1.Gintensite	3.869565
T2.Gintensite	3.304348
T3.Gintensite	3.708333
T1.Gvegetal	2.791667
T2.Gvegetal	3.391304

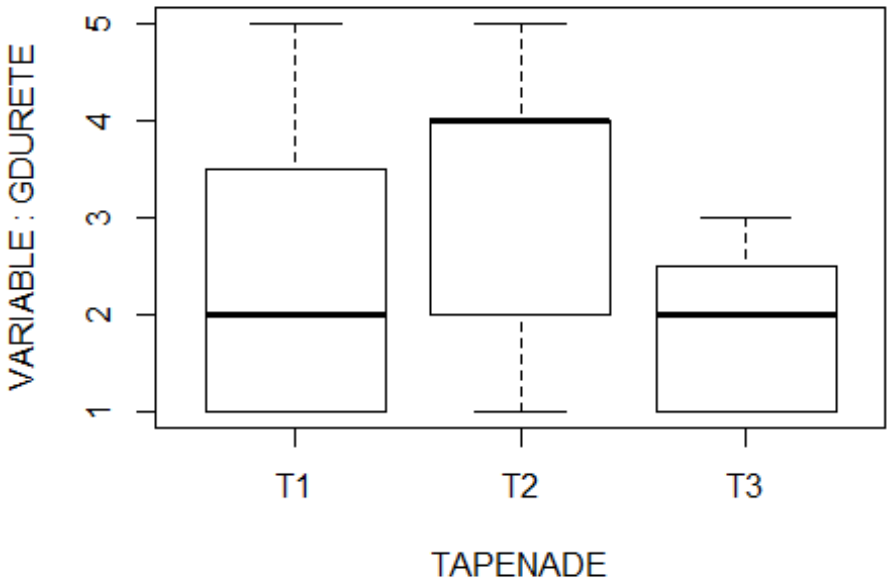
T3.Gvegetal	3.125000
T1.Giode	2.083333
T2.Giode	2.521739
T3.Giode	2.666667
T1.Gsale	3.375000
T2.Gsale	2.739130
T3.Gsale	3.708333
T1.Gfruite	2.391304
T2.Gfruite	2.250000
T3.Gfruite	2.347826
T1.Gepice	2.416667
T2.Gepice	2.250000
T3.Gepice	2.666667
T1.Gsucre	1.666667
T2.Gsucre	1.541667
T3.Gsucre	1.791667
T1.Gamer	2.333333
T2.Gamer	2.208333
T3.Gamer	2.833333

```
par(mfrow=c(1,1))
boxTable1(taData[,22:31])
```

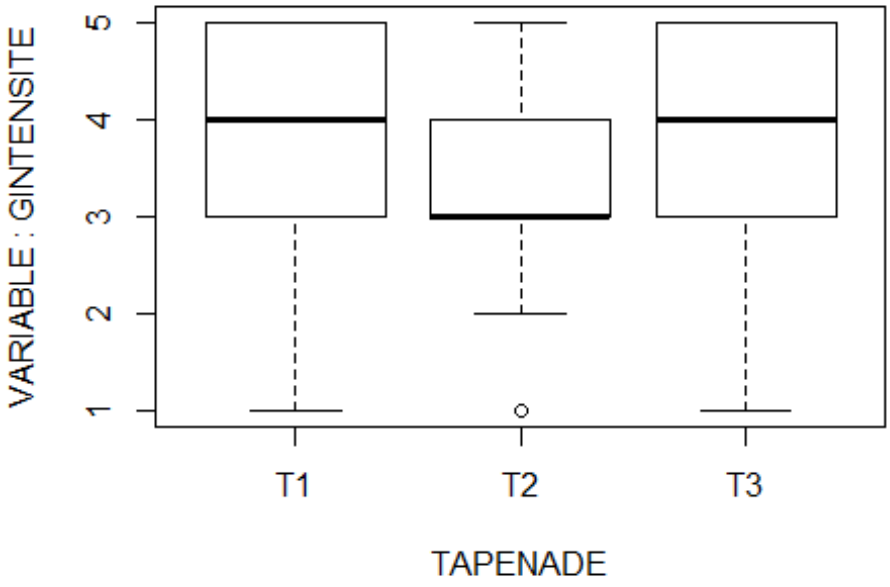
GACIDE VISUELLE



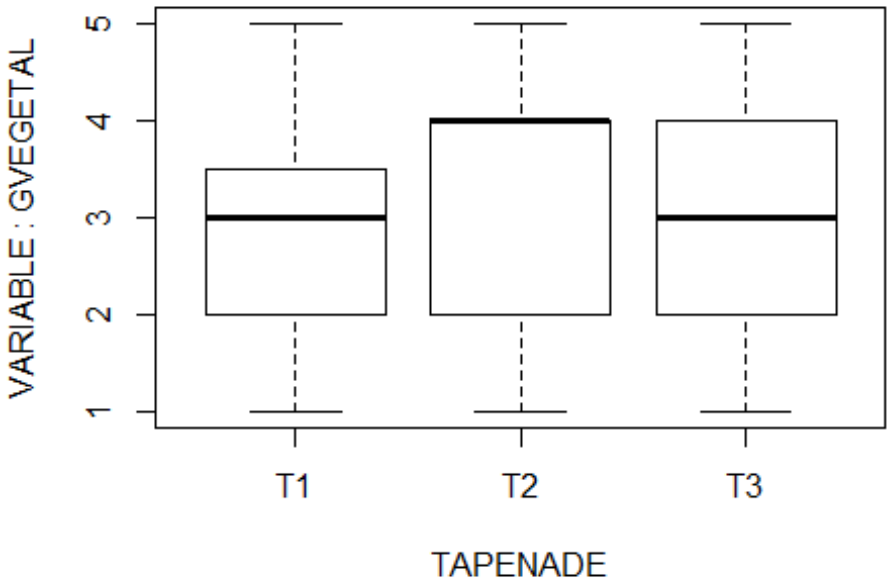
GDURETE VISUELLE



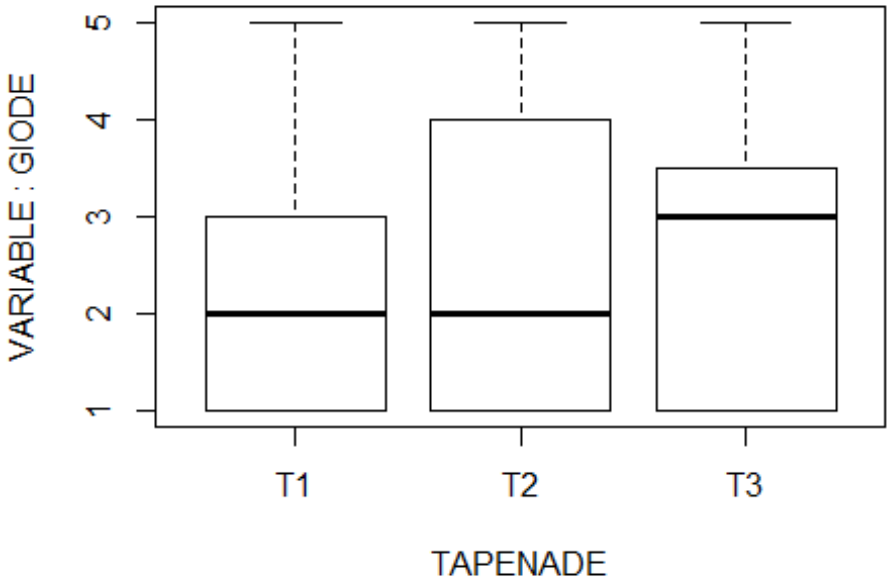
GINTENSITE VISUELLE



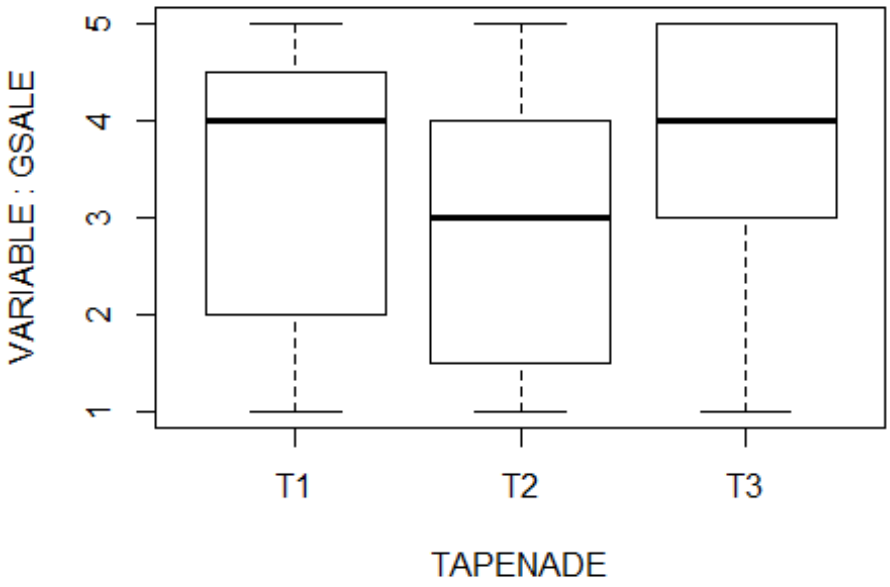
GVEGETAL VISUELLE



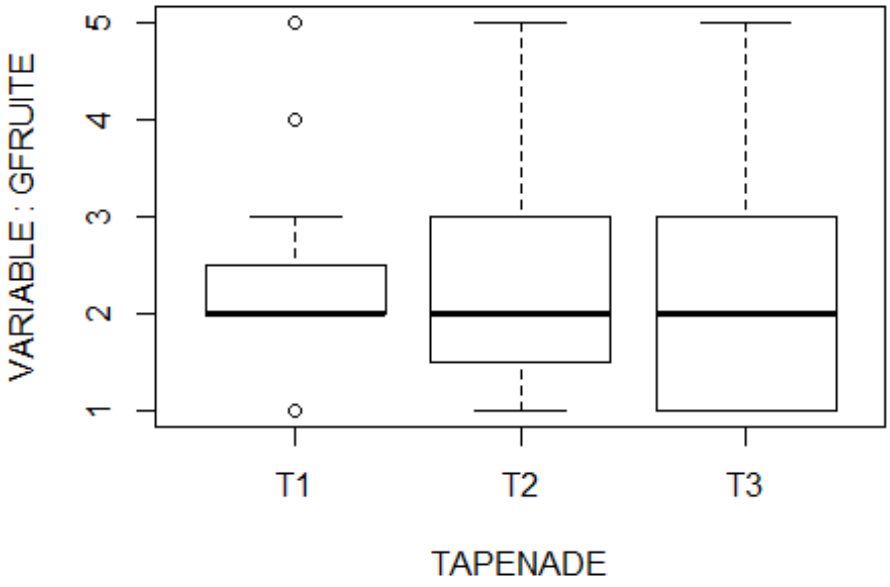
GIODE VISUELLE



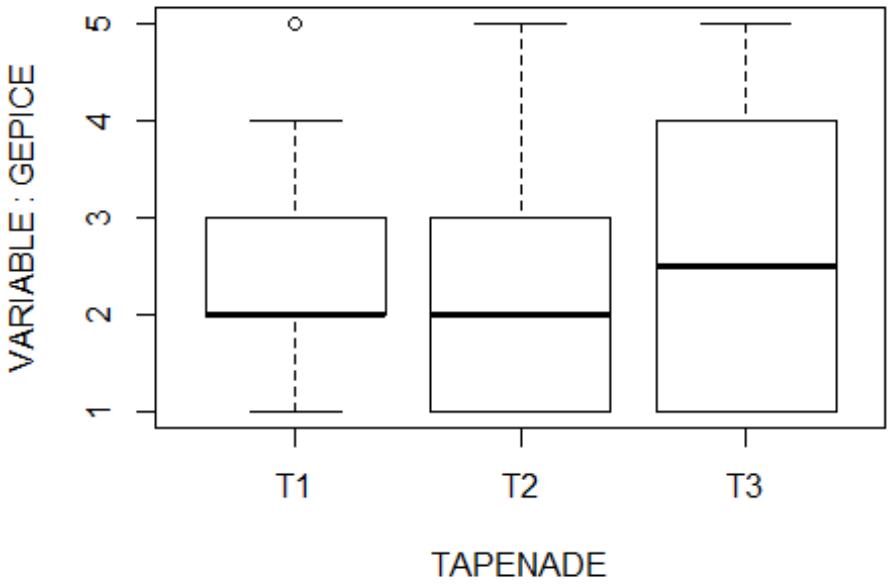
GSALE VISUELLE

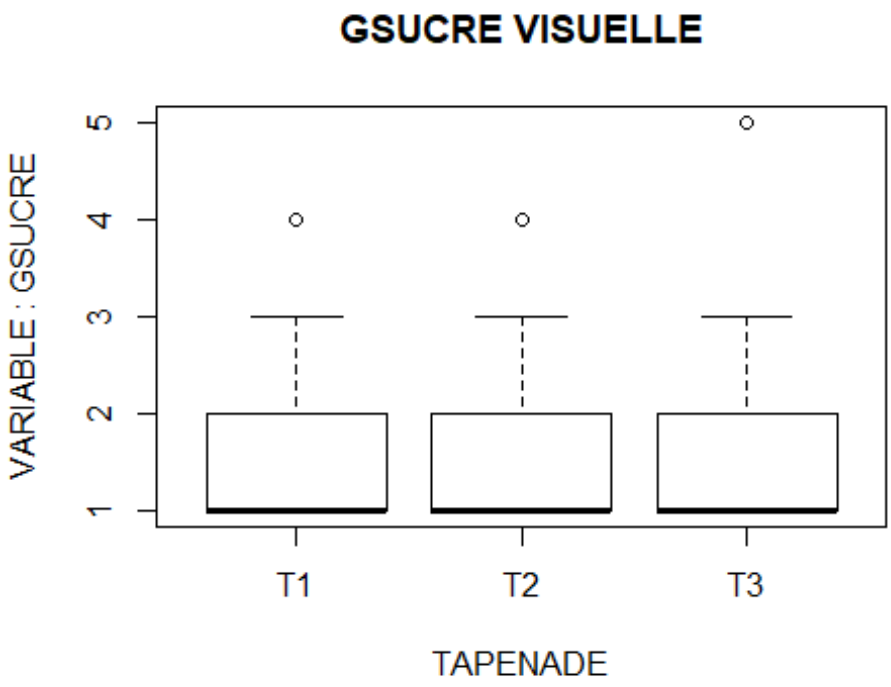


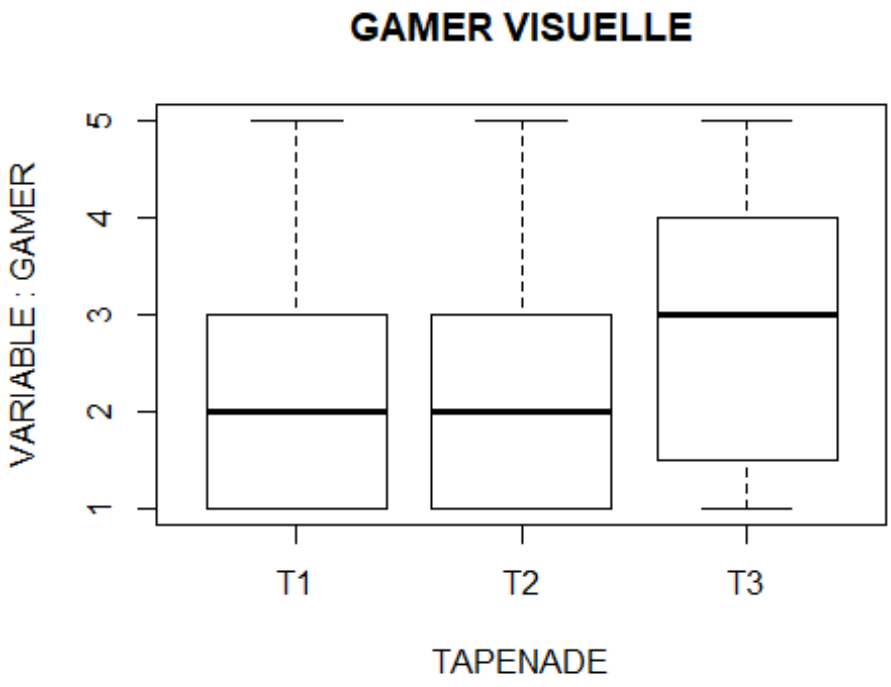
GFRUITE VISUELLE



GEPICE VISUELLE





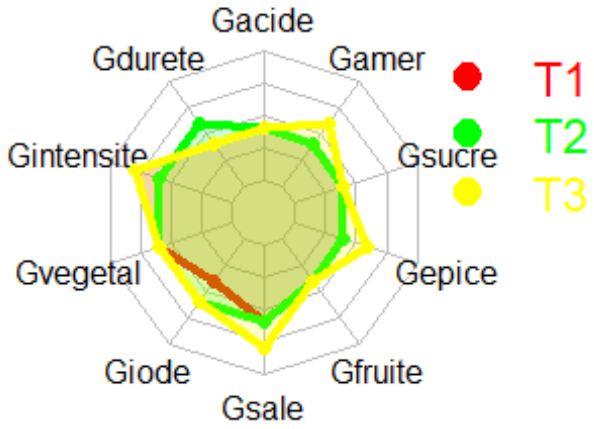



```r
#Radar
TapenadeT1=subset(taData[,22:31],taData$Tapenade=="T1")
TapenadeT2=subset(taData[,22:31],taData$Tapenade=="T2")
TapenadeT3=subset(taData[,22:31],taData$Tapenade=="T3")
```

```r
clc=data.frame("T1"= meancompCol(TapenadeT1),"T2"=
meancompCol(TapenadeT2),"T3"=
meancompCol(TapenadeT3));clc=as.data.frame(t(clc));clc=round(clc,0)
clc=rbind(rep(5,10) , rep(0,10) , clc);clc
```

```
##    Gacide Gdurete Gintensite Gvegetal Giode Gsale Gfruite Gepice Gsucre
Gamer
## 1       5       5          5        5     5     5       5      5      5
5
## 2       0       0          0        0     0     0       0      0      0
0
## T1      2       2          4        3     2     3       2      2      2
2
## T2      2       3          3        3     3     3       2      2      2
2
## T3      2       2          4        3     3     4       2      3      2
3
```

```r
radarchart(clc,axistype=0, maxmin=T,
           pcol=colors_border , pfcol=colors_in , plwd=4 , plty=1,
           cglcol="grey", cglty=1, axislabcol="black", cglwd=0.8,
           vlcex=1,

           )

legend(x=1, y=1.2, legend = rownames(clc[-c(1,2),]), bty = "n", pch=20 ,
col=coul , text.col = coul, cex=1.5, pt.cex=3)
```




```
#Finalité en bouche-------------------------------------------------------------
for (i in 71:88) print(paste("moyenne des valeurs
de",colnames(ini)[i],"=",meancomplet(ini[[i]])))
```

```
## [1] "moyenne des valeurs de T1.Fagreable = 3.17391304347826"
## [1] "moyenne des valeurs de T2.Fagreable = 2.65217391304348"
## [1] "moyenne des valeurs de T3.Fagreable = 2.95833333333333"
## [1] "moyenne des valeurs de T1.Fintensite = 3.65217391304348"
## [1] "moyenne des valeurs de T2.Fintensite = 3.21739130434783"
## [1] "moyenne des valeurs de T3.Fintensite = 3.5"
## [1] "moyenne des valeurs de T1.Fpersistance = 3.59090909090909"
## [1] "moyenne des valeurs de T2.Fpersistance = 3.17391304347826"
## [1] "moyenne des valeurs de T3.Fpersistance = 3.56521739130435"
## [1] "moyenne des valeurs de T1.Fnote = 3.58333333333333"
## [1] "moyenne des valeurs de T2.Fnote = 3.08333333333333"
## [1] "moyenne des valeurs de T3.Fnote = 2.91666666666667"
## [1] "moyenne des valeurs de T1.Fharmonie = 3.5"
## [1] "moyenne des valeurs de T2.Fharmonie = 2.91666666666667"
## [1] "moyenne des valeurs de T3.Fharmonie = 2.83333333333333"
## [1] "moyenne des valeurs de T1.Farôme = 3.70833333333333"
## [1] "moyenne des valeurs de T2.Farôme = 3"
## [1] "moyenne des valeurs de T3.Farôme = 3.20833333333333"
```

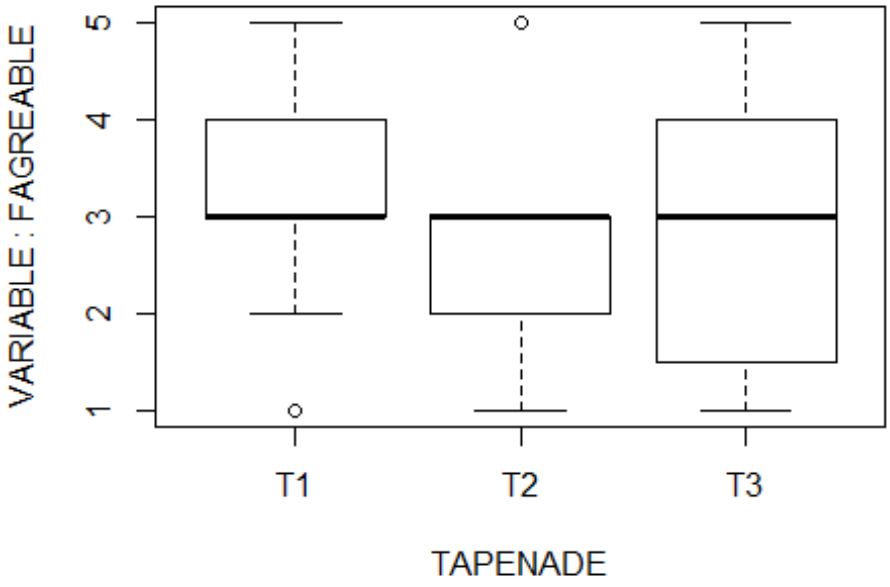
```
kable(data.frame("Moyenne"= meancompCol(ini[71:88])))
```

	Moyenne

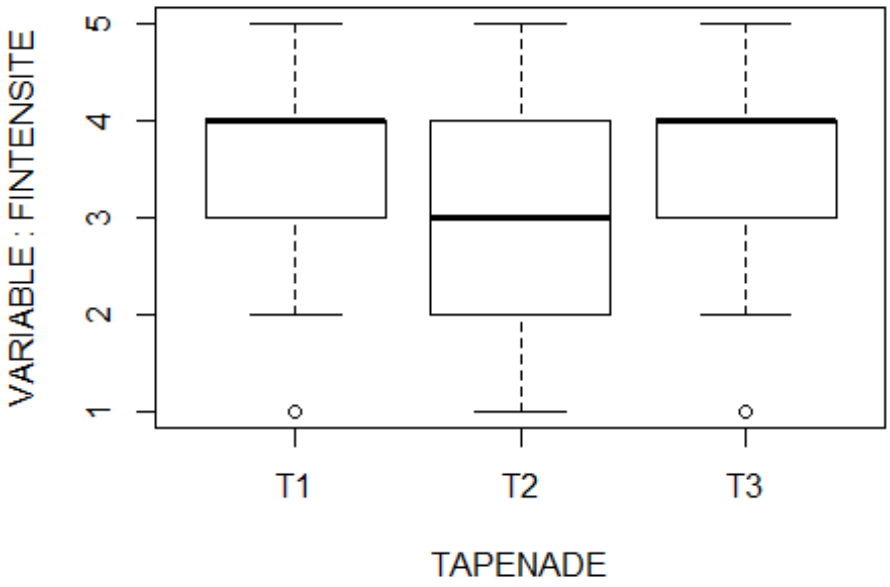
T1.Fagreable	3.173913
T2.Fagreable	2.652174
T3.Fagreable	2.958333
T1.Fintensite	3.652174
T2.Fintensite	3.217391
T3.Fintensite	3.500000
T1.Fpersistance	3.590909
T2.Fpersistance	3.173913
T3.Fpersistance	3.565217
T1.Fnote	3.583333
T2.Fnote	3.083333
T3.Fnote	2.916667
T1.Fharmonie	3.500000
T2.Fharmonie	2.916667
T3.Fharmonie	2.833333
T1.Farôme	3.708333
T2.Farôme	3.000000
T3.Farôme	3.208333

```
par(mfrow=c(1,1))
boxTable1(taData[,32:37])
```

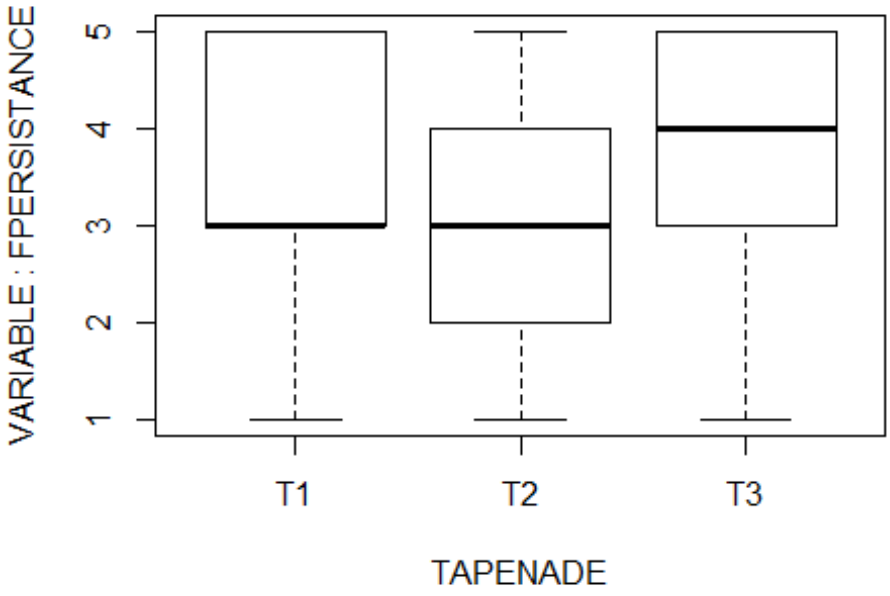
FAGREABLE VISUELLE



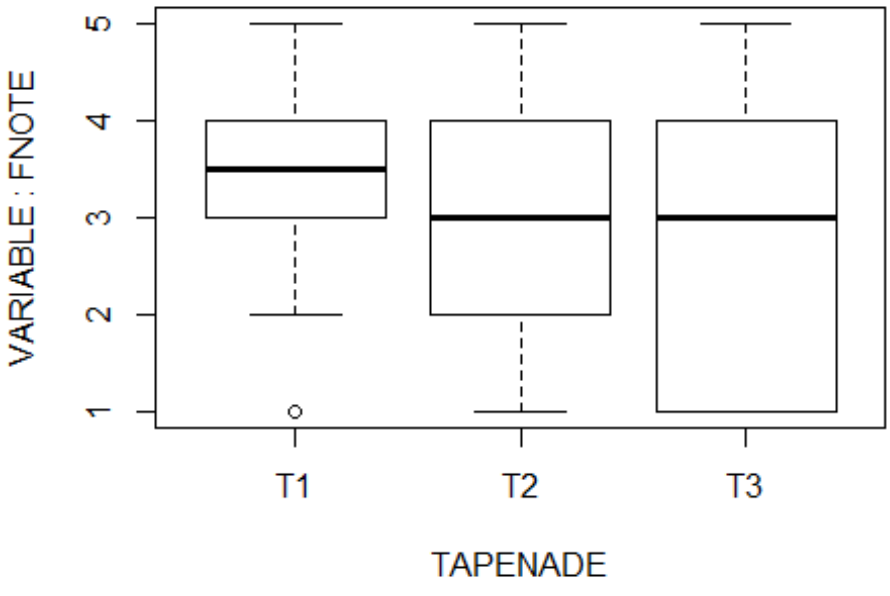
FINTENSITE VISUELLE

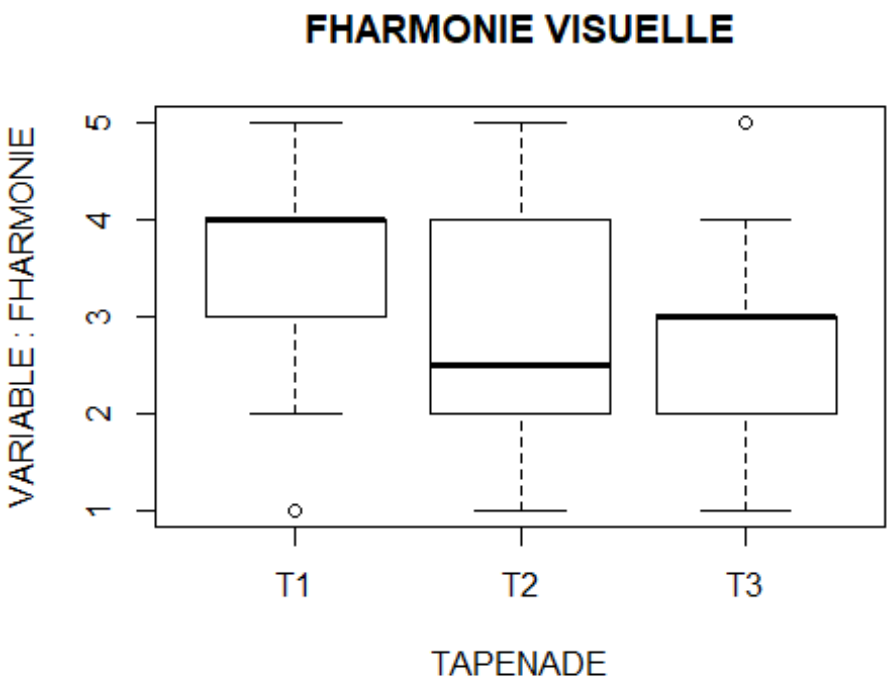


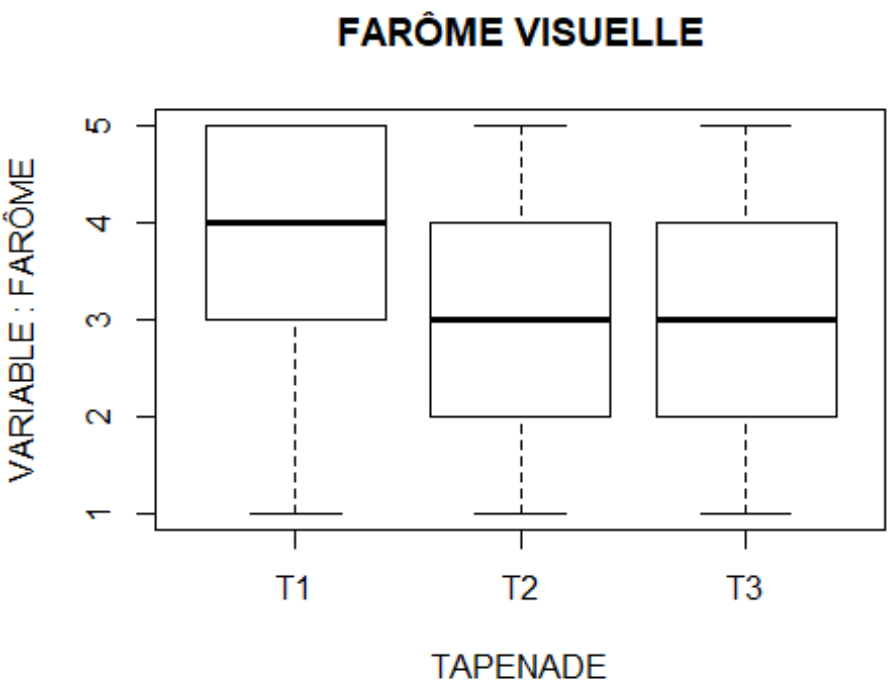
FPERSISTANCE VISUELLE



FNOTE VISUELLE





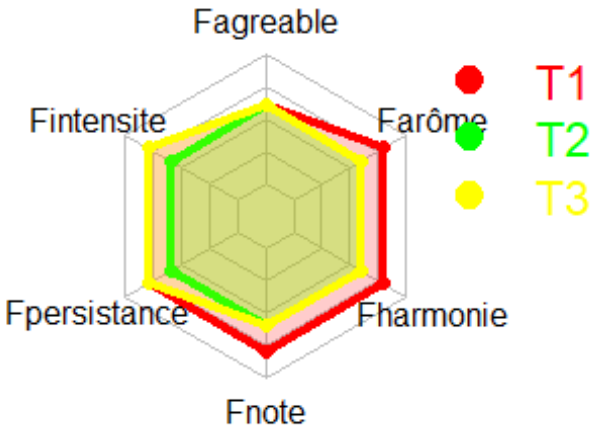



```
TapenadeT1=subset(taData[,32:37],taData$Tapenade=="T1")
TapenadeT2=subset(taData[,32:37],taData$Tapenade=="T2")
TapenadeT3=subset(taData[,32:37],taData$Tapenade=="T3")
```

```
clc=data.frame("T1"= meancompCol(TapenadeT1),"T2"=
meancompCol(TapenadeT2),"T3"=
meancompCol(TapenadeT3));clc=as.data.frame(t(clc));clc=round(clc,0)
clc=rbind(rep(5,6) , rep(0,6) , clc);clc
```

```
##    Fagreable Fintensite Fpersistance Fnote Fharmonie Farôme
## 1          5          5            5     5         5      5
## 2          0          0            0     0         0      0
## T1         3          4            4     4         4      4
## T2         3          3            3     3         3      3
## T3         3          4            4     3         3      3
```

```
radarchart(clc,axistype=0, maxmin=T,
           pcol=colors_border , pfcol=colors_in , plwd=4 , plty=1,
           cglcol="grey", cglty=1, axislabcol="black", cglwd=0.8,
           vlcex=1,

           )

legend(x=1, y=1.2, legend = rownames(clc[-c(1,2),]), bty = "n", pch=20 ,
col=coul , text.col = coul, cex=1.5, pt.cex=3)
```

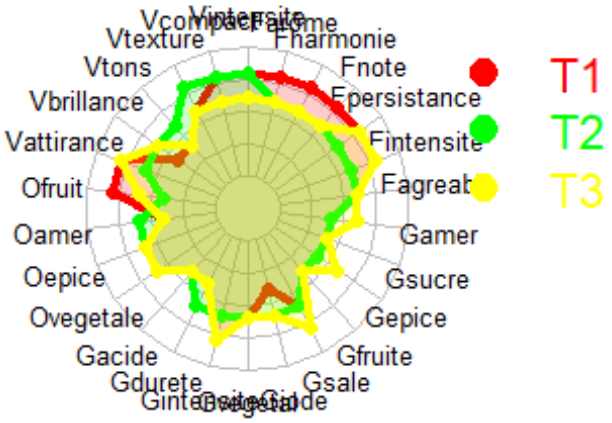



```
#les  26  variables ------------------------------------------------------
---------
TapenadeT1=subset(taData[,12:37],taData$Tapenade=="T1")
TapenadeT2=subset(taData[,12:37],taData$Tapenade=="T2")
TapenadeT3=subset(taData[,12:37],taData$Tapenade=="T3")
```

```r
clc=data.frame("T1"= meancompCol(TapenadeT1),"T2"=
meancompCol(TapenadeT2),"T3"=
meancompCol(TapenadeT3));clc=as.data.frame(t(clc));clc=round(clc,0);
clc=rbind(rep(5,26) , rep(0,26) , clc);clc
```

```
##    Vintensite Vcompact Vtexture Vtons Vbrillance Vattirance Ofruit Oamer
Oepice
## 1           5        5        5     5          5          5      5     5
5
## 2           0        0        0     0          0          0      0     0
0
## T1          4        4        3     2          2          4      4     2
3
## T2          4        4        4     3          3          3      2     3
3
## T3          3        3        3     2          3          4      3     2
3
##    Ovegetale Gacide Gdurete Gintensite Gvegetal Giode Gsale Gfruite Gepice
## 1          5      5       5          5        5     5     5       5      5
## 2          0      0       0          0        0     0     0       0      0
## T1         3      2       2          4        3     2     3       2      2
## T2         3      2       3          3        3     3     3       2      2
## T3         3      2       2          4        3     3     4       2      3
##    Gsucre Gamer Fagreable Fintensite Fpersistance Fnote Fharmonie Farôme
## 1       5     5         5          5            5     5         5      5
## 2       0     0         0          0            0     0         0      0
## T1      2     2         3          4            4     4         4      4
## T2      2     2         3          3            3     3         3      3
## T3      2     3         3          4            4     3         3      3
```

```r
#coul=brewer.pal(3, "BrBG"); colors_border=coul;colors_in =alpha(coul,0.5)
coul=c("red","green","yellow");colors_border=coul;colors_in =alpha(coul,0.2)

radarchart(clc,axistype=0, maxmin=T,
           pcol=colors_border , pfcol=colors_in , plwd=4 , plty=1,
           cglcol="grey", cglty=1, axislabcol="black", cglwd=0.8,
           vlcex=0.8,

           )

legend(x=1.2, y=1.2, legend = rownames(clc[-c(1,2),]), bty = "n", pch=20 ,
col=coul , text.col = coul, cex=1.5, pt.cex=3)
```



#quali-quali : croissement et table de contingence (Tapenade&Couleur)

```
#Pour cette étude , nous allons utilisé le tableau inid 72 observations non centrées
#  fonction : tableau de contingence
tc=function(X,Y)
{
    tc=table(X,Y); tc=cbind(tc,Total=apply(tc,1,sum));
    tc=rbind(tc,Total=apply(tc,2,sum));
    return(tc)
}

#fonction : les profils lignes
pl=function(X,Y)
{
    tc=table(X,Y); suml=apply(tc,1,sum);
    sumc=apply(tc,2,sum);
    pl=rbind(tc/suml,Ensemble=sumc/sum(sumc));
    return(pl)
}




##quelle est la couleur dominante trouvée pour chaque tapenade (tableau de contingence)
kable(table(inid$Tapenade,inid$Vcouleur))
```

	marron	noire	rouge	verte
T1	6	11	5	2
T2	1	4	1	18
T3	13	1	7	3

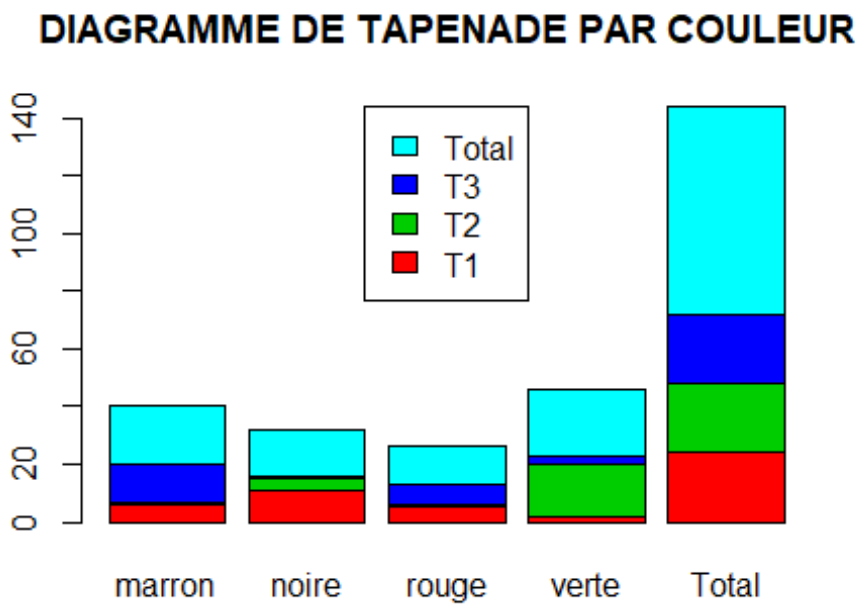
```
#Tableau contingence avec les totaux
tablTapCol=tc(inid$Tapenade,inid$Vcouleur);kable(tablTapCol)
```

	marron	noire	rouge	verte	Total
T1	6	11	5	2	24
T2	1	4	1	18	24
T3	13	1	7	3	24
Total	20	16	13	23	72

```
#profil lignes
tablTapColPr=pl(inid$Tapenade,inid$Vcouleur);kable(tablTapColPr)
```

	marron	noire	rouge	verte
T1	0.2500000	0.4583333	0.2083333	0.0833333
T2	0.0416667	0.1666667	0.0416667	0.7500000
T3	0.5416667	0.0416667	0.2916667	0.1250000
Ensemble	0.2777778	0.2222222	0.1805556	0.3194444

```
##Test du khi-deux  pour voir si la couleur est liée à la tapenade.
#Test d'indépendance
#Ho: la couleur et la tapenade sont indépendantes
t1=chisq.test(table(inid$Tapenade,inid$Vcouleur))

## Warning in chisq.test(table(inid$Tapenade, inid$Vcouleur)): Chi-squared
## approximation may be incorrect

#autres methodes
#CrossTable(inid$Tapenade,inid$Vcouleur,prop.t=F,prop.chisq=T)



#graphiques
par(mfrow=c(1,1))
barplot(tablTapCol,main ="DIAGRAMME DE TAPENADE PAR COULEUR",legend.text
=rownames(tablTapCol),col =2:5,args.legend=list(x="top"))
```




```
#-----------------------------------------------------------
#Tapenade T3 est plus lié  à la couleur marron
#Tapenade T1 est plus lié à la  couleur noire
#Tapendade T2 est plus lié  à la couleur verte
#-------------------------------------------------------------
```

#quali-quali: croissement et table de contingence (Tapenade&textuelle)

```
##quelle est Texture dominante trouvée pour chaque tapenade (tableau de
contingence)

# tableau de contingence  & profil lignes
#Tableau contingence avec les totaux
tablTapCol=tc(inid$Tapenade,inid$Gtexture);kable(tablTapCol)
```

	farineux	granuleux	lisse	onctueux	pateux	Total
T1	3	4	6	7	4	24
T2	3	12	3	3	2	23
T3	7	3	6	1	7	24
Total	13	19	15	11	13	71

```
#profil lignes
tablTapColPr=pl(inid$Tapenade,inid$Gtexture);kable(tablTapColPr)
```

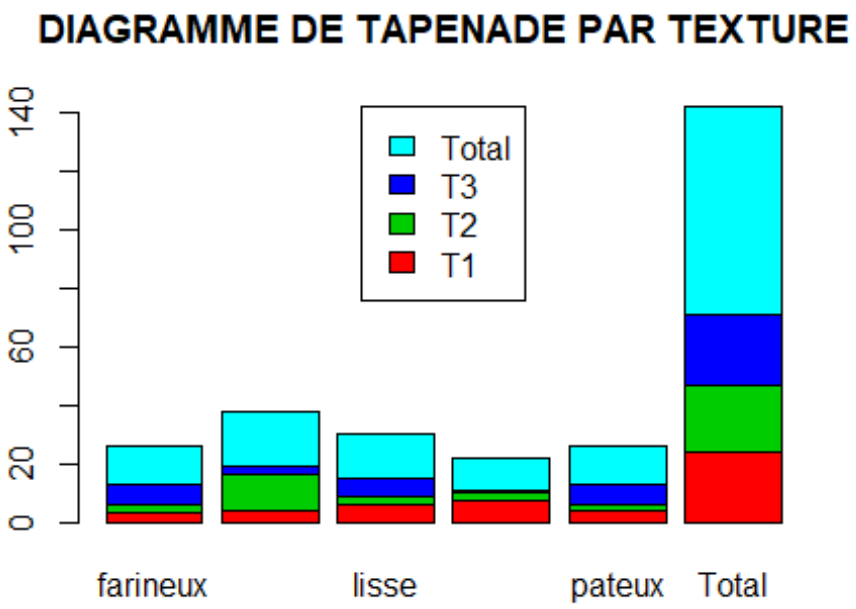
	farineux	granuleux	lisse	onctueux	pateux

T1	0.1250000	0.1666667	0.2500000	0.2916667	0.1666667
T2	0.1304348	0.5217391	0.1304348	0.1304348	0.0869565
T3	0.2916667	0.1250000	0.2500000	0.0416667	0.2916667
Ensemble	0.1830986	0.2676056	0.2112676	0.1549296	0.1830986

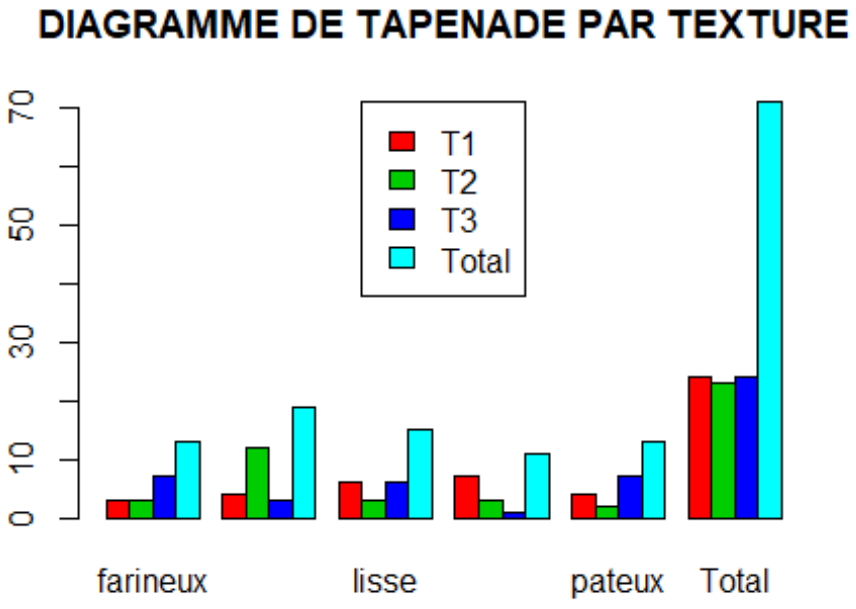
```
##Test du khi-deux  pour voir si la couleur est liée à la tapenade.
#Test d'indépendance
#Ho: la couleur et la tapenade sont indépendantes
t2=chisq.test(table(inid$Tapenade,inid$Gtexture))

## Warning in chisq.test(table(inid$Tapenade, inid$Gtexture)): Chi-squared
## approximation may be incorrect

tableIndep =data.frame("Croissements"=c("Couleur * Tapenade", "Texture *
Tapenade"),"Khi-
deux"=c(t1$statistic,t2$statistic),"ddl"=c(t1$parameter,t2$parameter),"Pvalue
" =c(t1$p.value,t2$p.value),"Significativité à 5%"=c("significative à 5%",
"significative à 5%") , "Conclusion"=c("sont liées","sont liées") );
kable(tableIndep)
```

Croissements	Khi.deux	ddl	Pvalue	Significativité.à.5.	Conclusion
Couleur * Tapenade	46.03921	6	0.0000000	significative à 5%	sont liées
Texture * Tapenade	19.39060	8	0.0129047	significative à 5%	sont liées

```
#Graphique
par(mfrow=c(1,1))
barplot(tablTapCol,main ="DIAGRAMME DE TAPENADE PAR TEXTURE",legend.text
=rownames(tablTapCol),col =2:5,args.legend=list(x="top"))
```




```
barplot(tablTapCol,beside = TRUE,main ="DIAGRAMME DE TAPENADE PAR
TEXTURE",legend.text =rownames(tablTapCol),col
=2:5,args.legend=list(x="top"))
```



```
#-------------------------------------------------
#Tapenade T3 est plus lié  texture Farineux et pateux
#Tapenade T1  et T3 est plus lié texture lisse
#Tapenade T1 est lié  texture onctueux
#Tapendade T2 est plus lié texture granuleux
#-------------------------------------------------
```

#quali * quali age* tapenadeplus

```
#Couper la serie age en deux niveau par rapport à la moyenne
taData=data.frame(inid)
mean(taData[,3]) # selon le consigne 30
```

```
## [1] 26.875
```

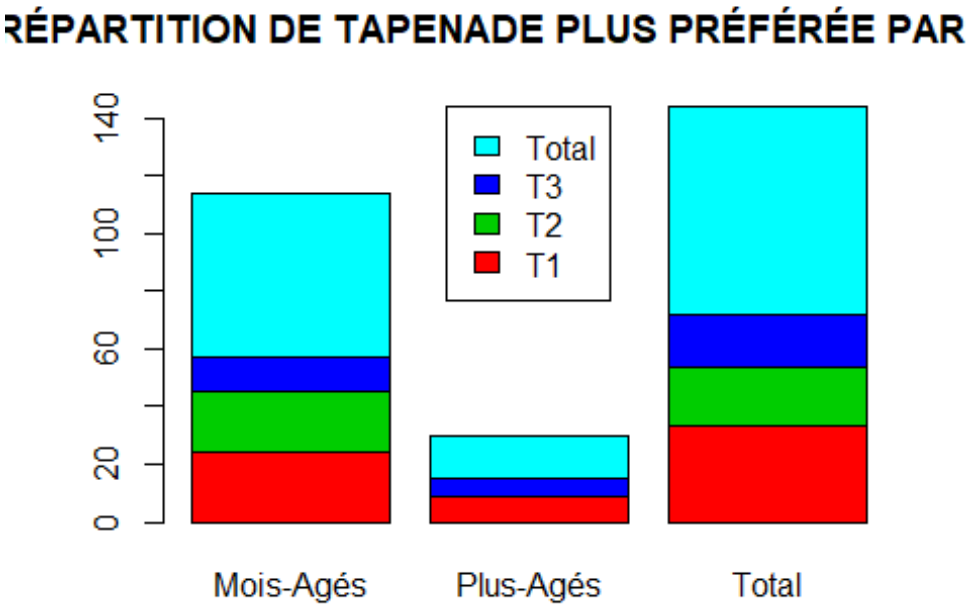
```
#couper les âges en 2 catégories selon la moyenne
taData$AgeCoupe=cut(taData[,3],breaks=c(0,30,length(taData[,3])))
levels(taData$AgeCoupe)=c("Mois-Agés","Plus-Agés")


 #tableau de contingence  &  profil lignes
tablTapCol=tc(taData$Tap.plus,taData$AgeCoupe);kable(tablTapCol)
```

	Mois-Agés	Plus-Agés	Total
T1	24	9	33
T2	21	0	21
T3	12	6	18
Total	57	15	72

```
#profil lignes
tablTapColPr=pl(taData$Tap.plus,taData$AgeCoupe);kable(tablTapColPr)
```

	Mois-Agés	Plus-Agés
T1	0.7272727	0.2727273
T2	1.0000000	0.0000000
T3	0.6666667	0.3333333
Ensemble	0.7916667	0.2083333

```
#autres methodes
#CrossTable(taData$Tap.plus,taData$AgeCoupe,prop.t=T,prop.chisq=T)


#Graphique
par(mfrow=c(1,1))
barplot(tablTapCol,main ="RÉPARTITION DE TAPENADE PLUS PRÉFÉRÉE PAR
ÂGE",legend.text =rownames(tablTapCol),col =2:5,args.legend=list(x="top"))
```




```
#-----------------------------------------------------
#Tapenade T2 est plus aimée par les jeunes (les moins âgés)
#Tapendade T1  et T3 est plus aimée par les vieux ( les plus âgés)
#-----------------------------------------------------
```

#quali * quali : age* tapenade moins

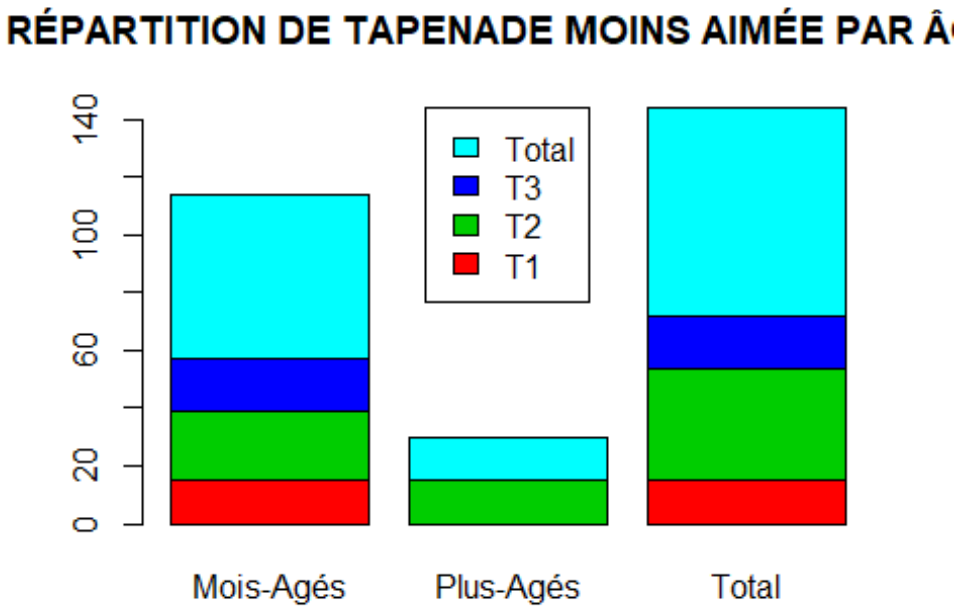
```
#tableau de contingence
tablTapCol=tc(taData$Tap.moins,taData$AgeCoupe);kable(tablTapCol)
```

	Mois-Agés	Plus-Agés	Total
T1	15	0	15
T2	24	15	39
T3	18	0	18
Total	57	15	72

```
#profil lignes
tablTapColPr=pl(taData$Tap.moins,taData$AgeCoupe);kable(tablTapColPr)
```

	Mois-Agés	Plus-Agés
T1	1.0000000	0.0000000
T2	0.6153846	0.3846154
T3	1.0000000	0.0000000
Ensemble	0.7916667	0.2083333

```r
#autres methodes
#CrossTable(taData$Tap.moins,taData$AgeCoupe,prop.t=T,prop.chisq=T)

#Graphique
par(mfrow=c(1,1))
barplot(tablTapCol,main ="RÉPARTITION DE TAPENADE MOINS AIMÉE PAR
ÂGE",legend.text =rownames(tablTapCol),col =2:5,args.legend=list(x="top"))
```




```r
#-------------------------------------------------
#Tapenade T2 est moins aimée par  les vieux ( les plus âgés)
#Tapendade T1  et T3 est moins aimée par les jeunes (les moins âgés)
#-------------------------------------------------
```

quali * quali tapenade * age

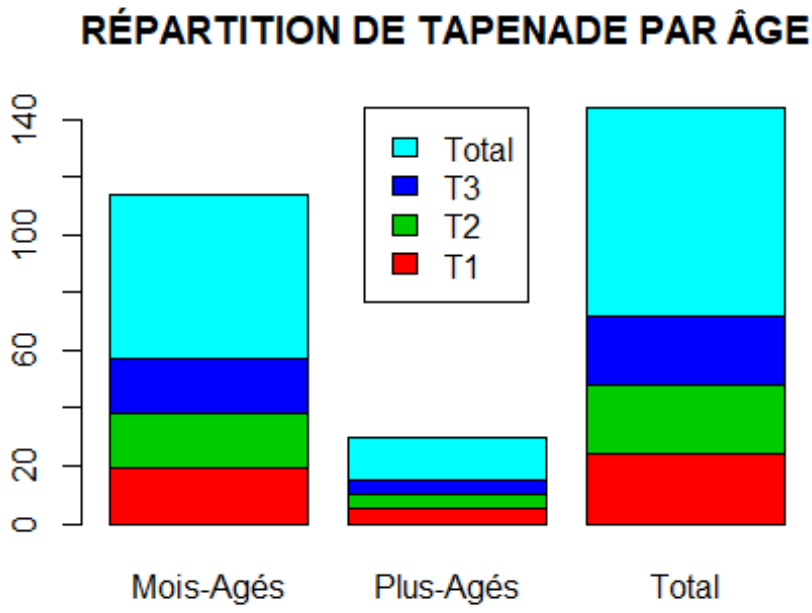
```r
#tableau de contingence
tablTapCol=tc(taData$Tapenade,taData$AgeCoupe);kable(tablTapCol)
```

	Mois-Agés	Plus-Agés	Total
T1	19	5	24
T2	19	5	24
T3	19	5	24
Total	57	15	72

```r
# profil lignes
tablTapColPr=pl(taData$Tapenade,taData$AgeCoupe);kable(tablTapColPr)
```

	Mois-Agés	Plus-Agés
T1	0.7916667	0.2083333
T2	0.7916667	0.2083333
T3	0.7916667	0.2083333
Ensemble	0.7916667	0.2083333

```r
#autres methodes
#CrossTable(taData$Tapenade,taData$AgeCoupe,prop.t=T,prop.chisq=T)

#Graphique
par(mfrow=c(1,1))
barplot(tablTapCol,main ="RÉPARTITION DE TAPENADE PAR ÂGE",legend.text
=rownames(tablTapCol),col =2:5,args.legend=list(x="top"))
```




```r
#-----------------------------------------------------
#les jeunes ont plus participés que les vieux
#-----------------------------------------------------
```

#quali*quali : CSP Tapenade

```r
##quelle est CSP dominant trouvée pour chaque tapenade (tableau de
contingence)
```

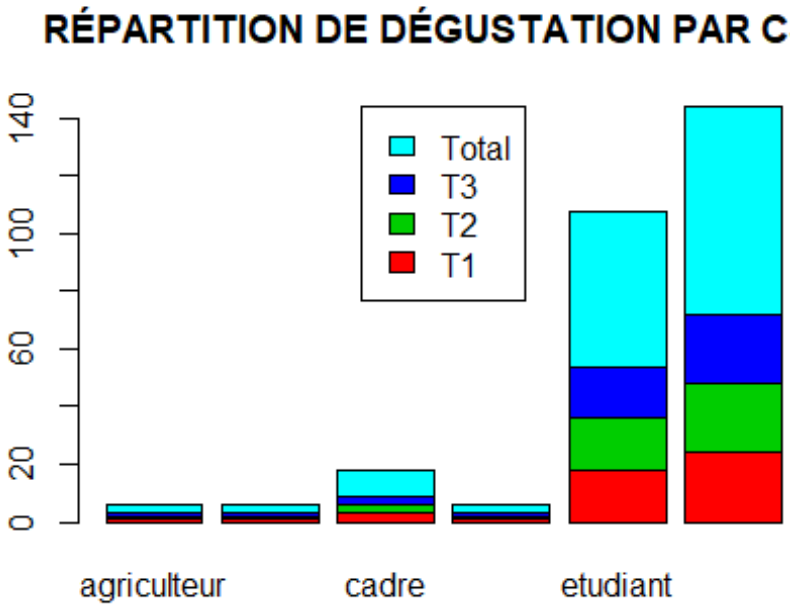
```r
# tableau de contingence
tablTapCol=tc(inid$Tapenade,inid$CSP);kable(tablTapCol)
```

	agriculteur	autre	cadre	employe	etudiant	Total
T1	1	1	3	1	18	24
T2	1	1	3	1	18	24
T3	1	1	3	1	18	24
Total	3	3	9	3	54	72

```r
#profil lignes
tablTapColPr=pl(inid$Tapenade,inid$CSP);kable(tablTapColPr)
```

	agriculteur	autre	cadre	employe	etudiant
T1	0.0416667	0.0416667	0.125	0.0416667	0.75
T2	0.0416667	0.0416667	0.125	0.0416667	0.75
T3	0.0416667	0.0416667	0.125	0.0416667	0.75
Ensemble	0.0416667	0.0416667	0.125	0.0416667	0.75

```r
#autres methodes
#CrossTable(inid$Tapenade,inid$CSP,prop.t=T,prop.chisq=T)

#Graphique
par(mfrow=c(1,1))
barplot(tablTapCol,main ="RÉPARTITION DE DÉGUSTATION PAR CSP",legend.text
=rownames(tablTapCol),col =2:5,args.legend=list(x="top"))
```




```
#-----------------------------------------------------
#Les etudiants sont les plus à participer à la degustation
#-----------------------------------------------------
```

#quali*quali :apprecieOlives

```
##quelle apprecieOlives trouvée pour chaque tapenade (tableau de contingence)

# tableau de contingence  & profil lignes
tablTapCol=tc(inid$Tapenade,inid$apprecieOlives);kable(tablTapCol)
```

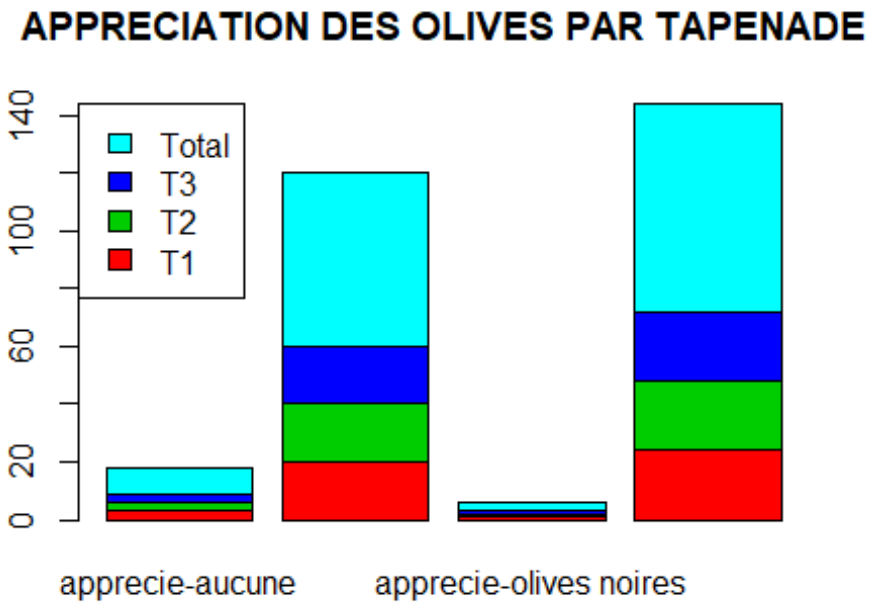
	apprecie-aucune	apprecie-les deux	apprecie-olives noires	Total
T1	3	20	1	24
T2	3	20	1	24
T3	3	20	1	24
Total	9	60	3	72

```
#profil lignes
tablTapColPr=pl(inid$Tapenade,inid$apprecieOlives);kable(tablTapColPr)
```

	apprecie-aucune	apprecie-les deux	apprecie-olives noires
T1	0.125	0.8333333	0.0416667
T2	0.125	0.8333333	0.0416667
T3	0.125	0.8333333	0.0416667

Ensemble	0.125	0.8333333	0.0416667

```
#autres methodes
#CrossTable(inid$Tapenade,inid$apprecieOlives,prop.t=T,prop.chisq=T)


#Graphique
par(mfrow=c(1,1))
barplot(tablTapCol,main ="APPRECIATION DES OLIVES PAR TAPENADE",legend.text
=rownames(tablTapCol),col =2:5,args.legend=list(x="topleft"))
```




```
#-------------------------------------------------
#les degustateurs apprecient les deux olives
#-------------------------------------------------
```

#qauli * quali frequenceOlives

```
##quelle frequenceOlives trouvée pour chaque tapenade (tableau de
contingence)

# tableau de contingence
tablTapCol=tc(inid$Tapenade,inid$frequenceOlives);kable(tablTapCol)
```

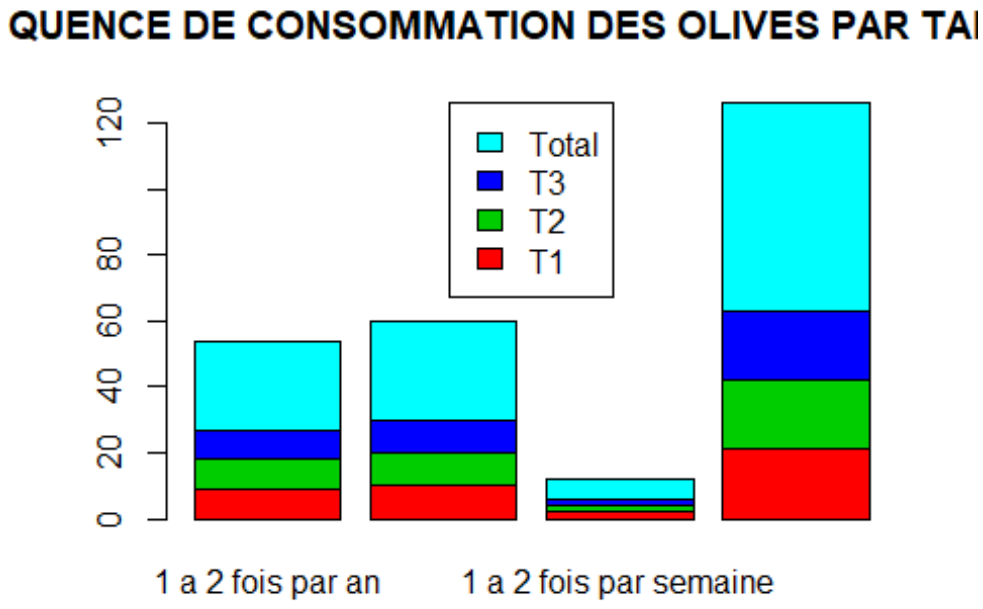
	1 a 2 fois par an	1 a 2 fois par mois	1 a 2 fois par semaine	Total
T1	9	10	2	21
T2	9	10	2	21

T3	9	10	2	21
Total	27	30	6	63

```
#profil lignes
tablTapColPr=pl(inid$Tapenade,inid$frequenceOlives);kable(tablTapColPr)
```

	1 a 2 fois par an	1 a 2 fois par mois	1 a 2 fois par semaine
T1	0.4285714	0.4761905	0.0952381
T2	0.4285714	0.4761905	0.0952381
T3	0.4285714	0.4761905	0.0952381
Ensemble	0.4285714	0.4761905	0.0952381

```
#autres methodes
#CrossTable(inid$Tapenade,inid$frequenceOlives,prop.t=T,prop.chisq=T)

#Graphique
par(mfrow=c(1,1))
barplot(tablTapCol,main ="FRÉQUENCE DE CONSOMMATION DES OLIVES PAR
TAPENADE",legend.text =rownames(tablTapCol),col
=2:5,args.legend=list(x="top"))
```




```
#-------------------------------------------------
#Tfrequence par mois souvent et par an
#-------------------------------------------------
```

#les statistiques bivariées : quanti * quanti et significativité

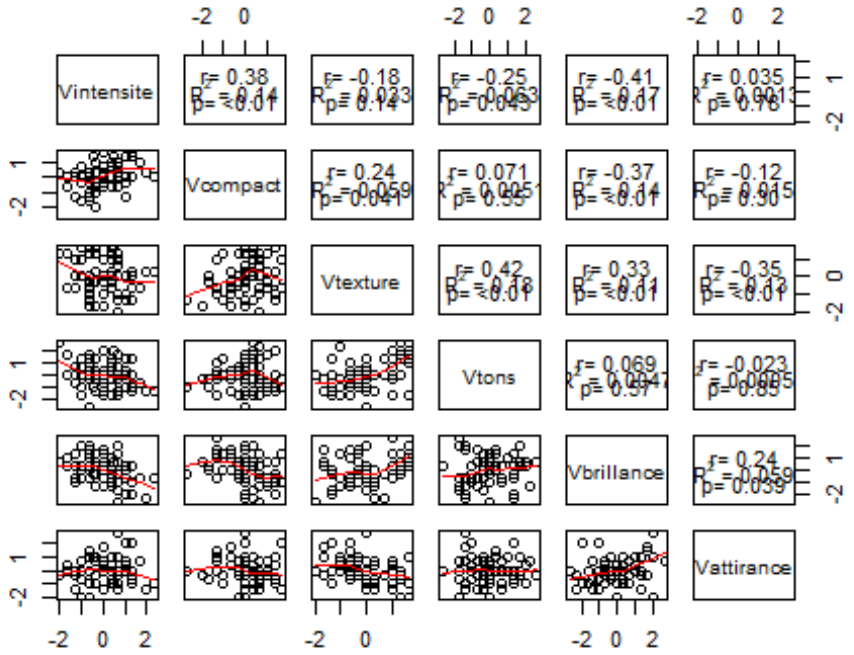
```r
# Pour cette analyse , on utilisera les variables quantitatives centrées du
Tableau
#Croisement quanti-quanti

#  fonctions matrices de p-values
cort=function(x,y) {
  present=!is.na(apply(data.frame(x,y),1,sum))

return(cor.test(as.numeric(x[[1]][present]),as.numeric(y[[1]][present]))[[3]]
)
}

#
matpvalue=function(donnees){
p=ncol(donnees)
   mat=matrix(0,nrow=p,ncol=p)
   dimnames(mat)=list(colnames(donnees),colnames(donnees))
   for (i in 1:(p-1))
     for (j in ((i+1):p))
     {
       mat[i,j]=cort(donnees[,i],donnees[,j])
       mat[j,i]=mat[i,j]
     }
   return(mat)
}

panel.cor_simple=function(x, y, digits=2, prefix="", cex.cor)
    {
        usr =par("usr"); on.exit(par(usr))
        par(usr = c(0, 1, 0, 1))
        r =cor(x, y)
        txt =format(c(r, 0.123456789), digits=digits)[1]
        txt =paste(prefix, txt, sep="")
        if(missing(cex.cor)) cex=0.8/strwidth(txt)
        test=cor.test(x,y)
        # borrowed from printCoefmat
        Signif=symnum(test$p.value, corr = FALSE, na = FALSE,
                      cutpoints = c(0, 0.001, 0.01, 0.05, 0.1, 1),
                      symbols = c("***", "**", "*", ".", " "))

        text(0.5, 0.5, txt, cex = cex * abs(r))
        text(.8, .8, Signif, cex=cex, col=2)
    }



#Caractéristiques visuelles
#pairs(tableau[,12:17])
kable(round(cor(tableau[,12:17],use="complete.obs"),digits=3))
```

	Vintensite	Vcompact	Vtexture	Vtons	Vbrillance	Vattirance
Vintensite	1.000	0.386	-0.184	-0.251	-0.419	0.034
Vcompact	0.386	1.000	0.252	-0.017	-0.383	-0.110
Vtexture	-0.184	0.252	1.000	0.491	0.342	-0.371
Vtons	-0.251	-0.017	0.491	1.000	0.221	-0.043
Vbrillance	-0.419	-0.383	0.342	0.221	1.000	0.235
Vattirance	0.034	-0.110	-0.371	-0.043	0.235	1.000

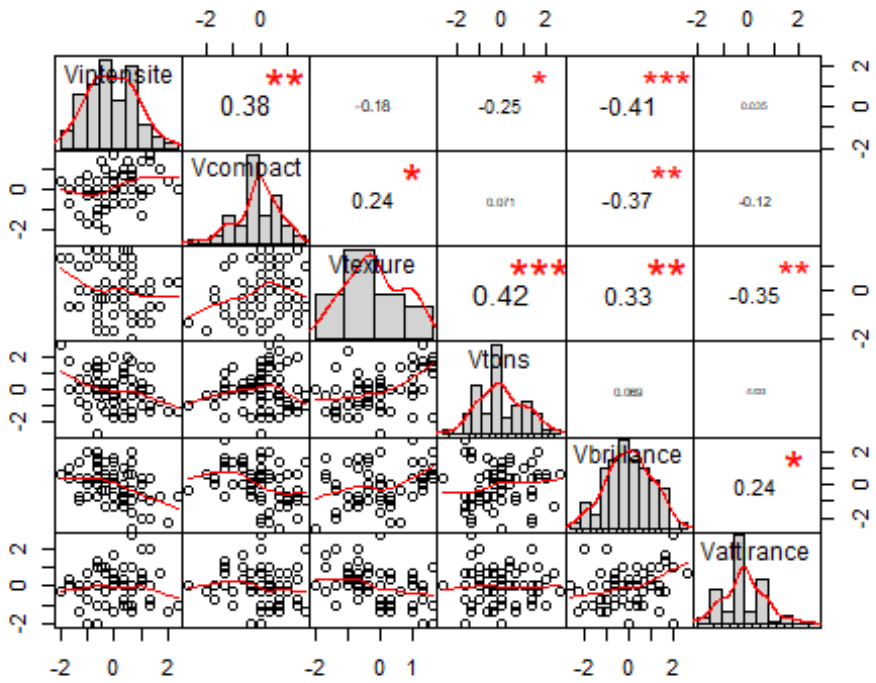
```
kable(round(matpvalue(tableau[,12:17]),digits=3))
```

	Vintensite	Vcompact	Vtexture	Vtons	Vbrillance	Vattirance
Vintensite	0.000	0.002	0.144	0.043	0.000	0.776
Vcompact	0.002	0.000	0.041	0.555	0.001	0.301
Vtexture	0.144	0.041	0.000	0.000	0.005	0.002
Vtons	0.043	0.555	0.000	0.000	0.568	0.846
Vbrillance	0.000	0.001	0.005	0.568	0.000	0.039
Vattirance	0.776	0.301	0.002	0.846	0.039	0.000

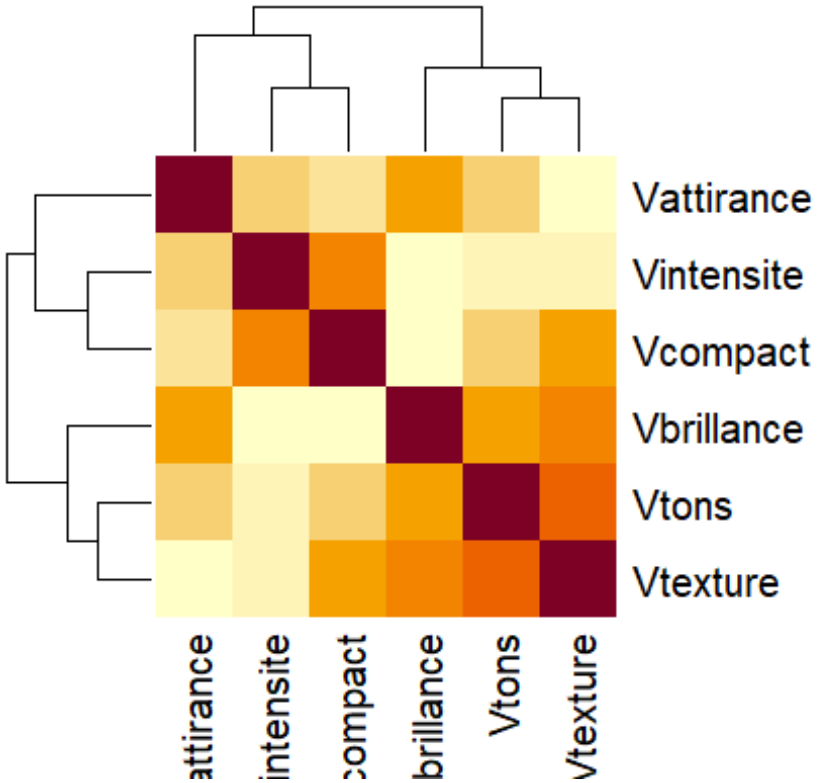
```
pairs(tableau[,12:17],lower.panel=panel.smooth, upper.panel=panel.cor)
```



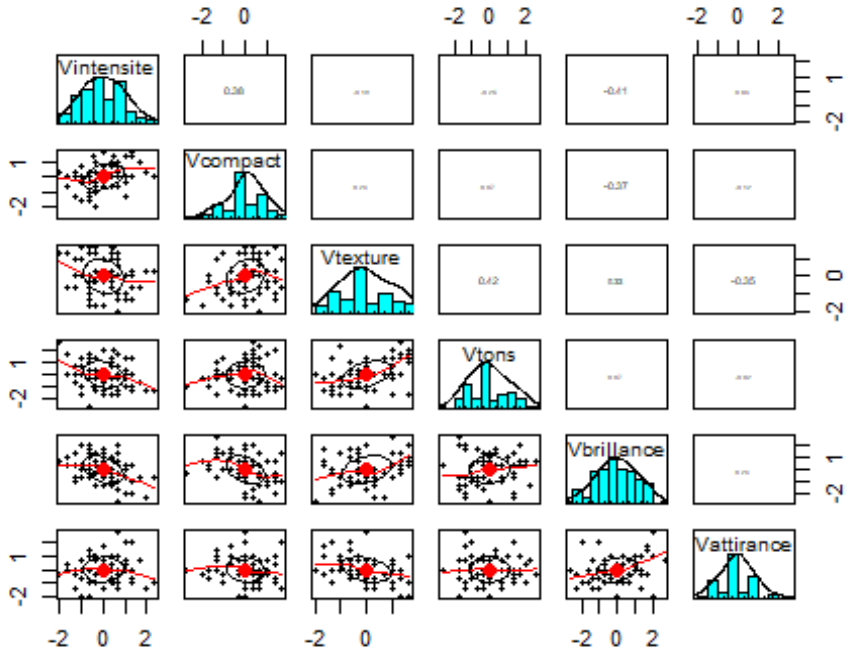

```
chart.Correlation(tableau[,12:17], histogram=TRUE, pch=19)
```




```
heatmap(x=cor(tableau[,12:17],use="complete.obs"),symm = TRUE)
```




```
pairs.panels(tableau[,12:17], scale=TRUE)
```




```
ggpairs(tableau[,12:17], mapping = aes(color = tableau$Tapenade))
```

```
## Warning: Removed 5 rows containing non-finite values (stat_density).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
"pearson", :
## Removed 5 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
"pearson", :
## Removed 6 rows containing missing values
```

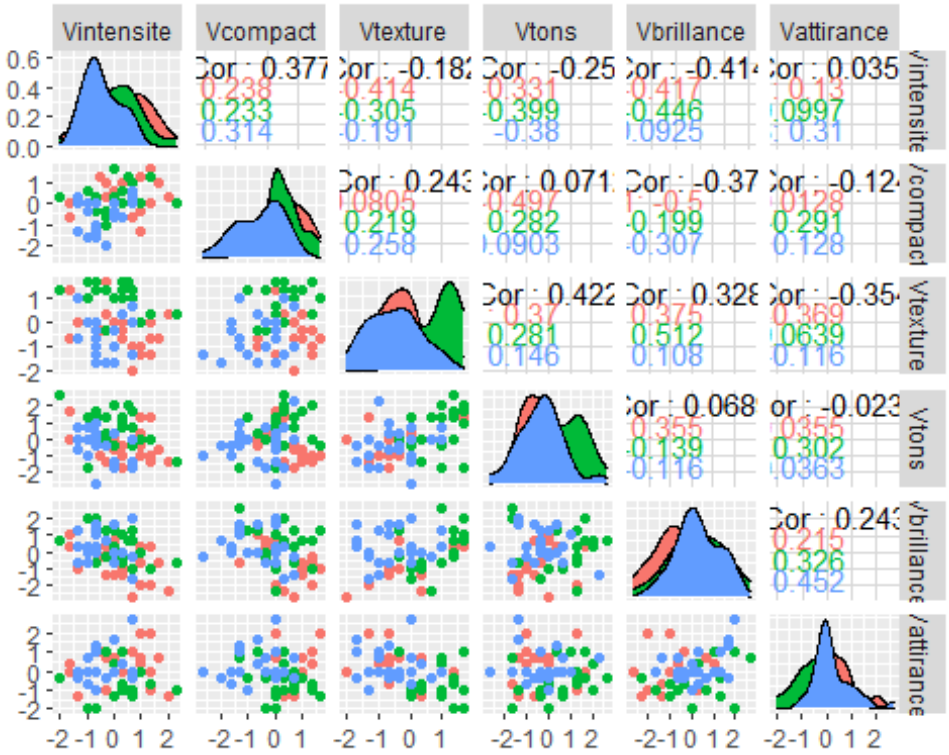
```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
"pearson", :
## Removed 6 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
"pearson", :
## Removed 5 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
"pearson", :
## Removed 5 rows containing missing values
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
"pearson", :
## Removing 1 row that contained a missing value

## Warning in (function (data, mapping, alignPercent = 0.6, method =
"pearson", :
## Removing 1 row that contained a missing value

## Warning: Removed 6 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
"pearson", :
## Removed 2 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
"pearson", :
## Removing 1 row that contained a missing value

## Warning in (function (data, mapping, alignPercent = 0.6, method =
"pearson", :
## Removing 1 row that contained a missing value

## Warning: Removed 6 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
"pearson", :
## Removing 1 row that contained a missing value

## Warning in (function (data, mapping, alignPercent = 0.6, method =
"pearson", :
## Removing 1 row that contained a missing value

## Warning: Removed 5 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 5 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_point).
```

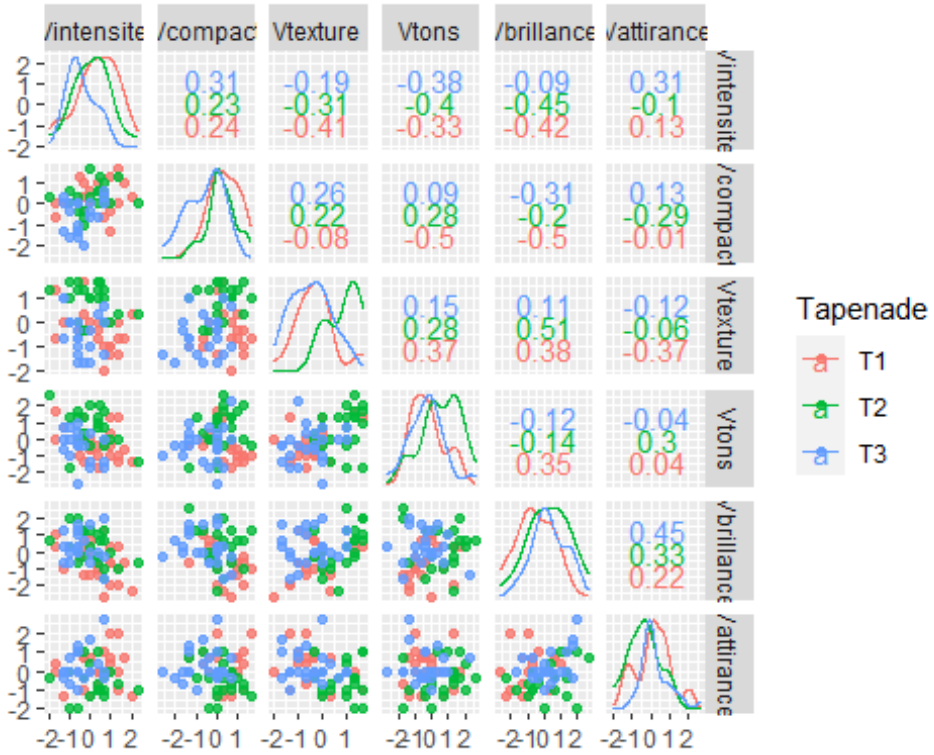



```
ggscatmat(tableau, columns = 12:17, color="Tapenade", alpha=0.8)
```

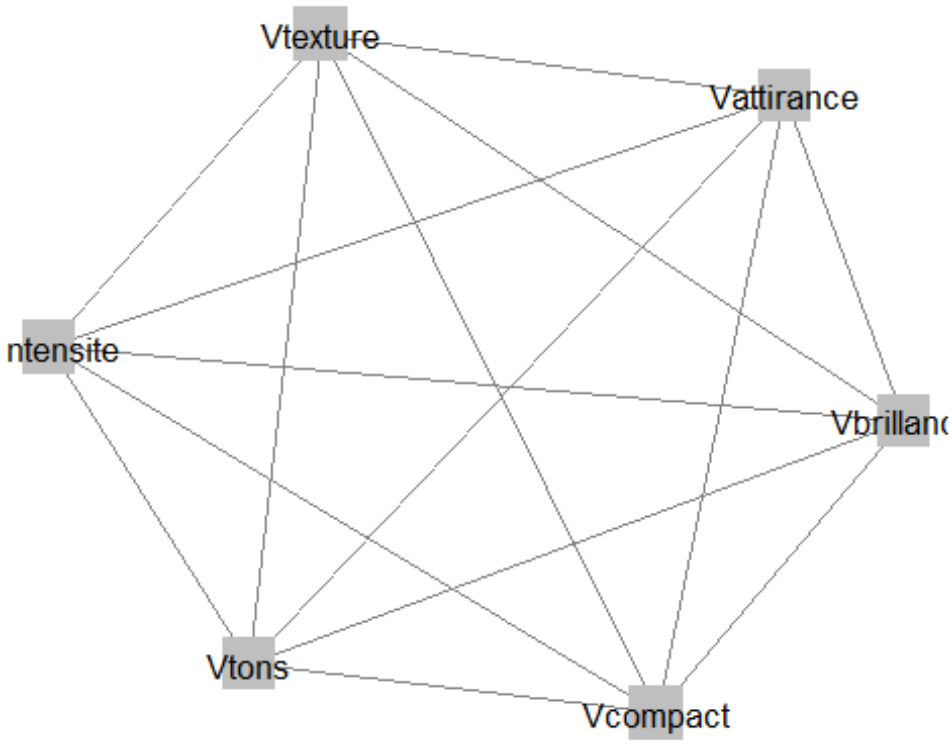
```
## Warning: Removed 5 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```



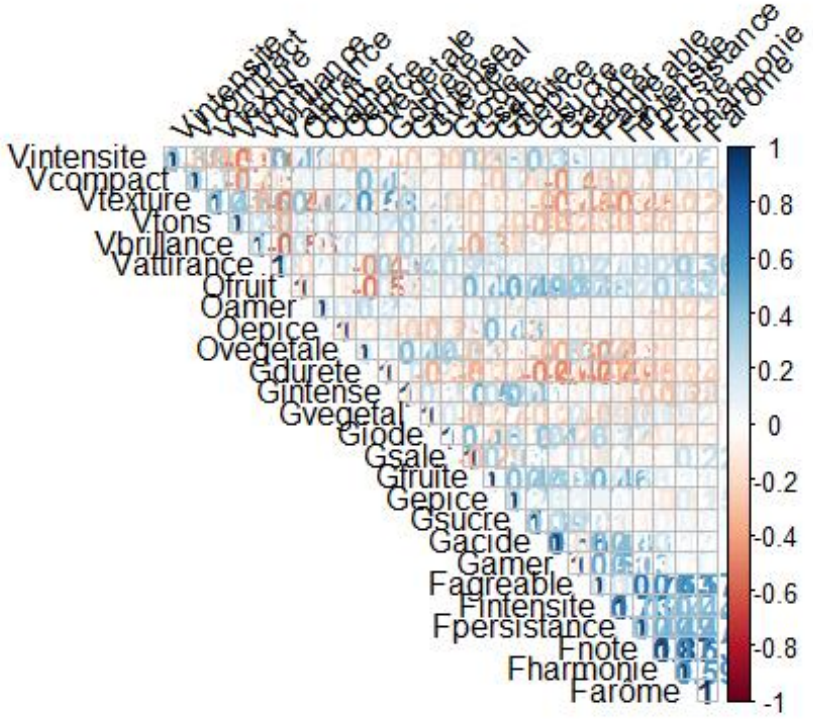

```
n=network::network(cor(tableau[,12:17],use="complete.obs"), directed =
FALSE);n

##  Network attributes:
##   vertices = 6
##   directed = FALSE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 15
##     missing edges= 0
##     non-missing edges= 15
##
##  Vertex attribute names:
##     vertex.names
##
## No edge attributes

ggnet2(n, label = TRUE, shape = 15)
```



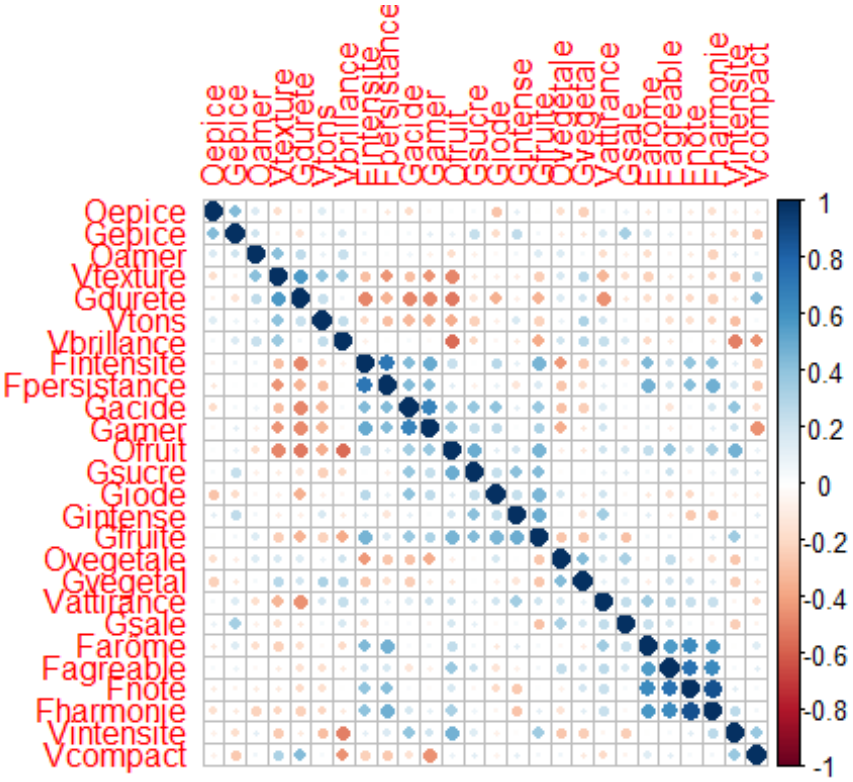

```
library(corrr)
# network_plot(correlate(cor(tableau[,12:17],use="complete.obs")),min_cor =
.4,colors = c("red", "green"), legend = TRUE )



# #Caractéristiques Olfactives
# #pairs(tableau[,18:21])
# kable(round(cor(tableau[,18:21],use="complete.obs"),digits=3))
# kable(round(matpvalue(tableau[,18:21]),digits=3))
# pairs(tableau[,18:21],lower.panel=panel.smooth, upper.panel=panel.cor)
# heatmap(x =cor(tableau[,18:21],use="complete.obs"), symm = TRUE)
# chart.Correlation(tableau[,18:21], histogram=TRUE, pch=19)
# pairs.panels(tableau[,18:21], scale=TRUE)
# ggpairs(tableau[,18:21], mapping = aes(color = tableau$Tapenade))
# ggscatmat(tableau, columns = 18:21, color="Tapenade", alpha=0.8)
# n=network::network(cor(tableau[,18:21],use="complete.obs"), directed =
FALSE);n
# ggnet2(n, label = TRUE, shape = 15)
# library(corrr)
# network_plot(correlate(cor(tableau[,18:22],use="complete.obs")),min_cor =
.4,colors = c("red", "green"), legend = TRUE )
#
#
#
# #Caractéristiques Gustatives
```

```
# #pairs(tableau[,22:31])
# kable(round(cor(tableau[,22:31],use="complete.obs"),digits=3))
# kable(round(matpvalue(tableau[,22:31]),digits=3))
# pairs(tableau[,22:31],lower.panel=panel.smooth, upper.panel=panel.cor)
# heatmap(x =cor(tableau[,22:31],use="complete.obs"), symm = TRUE)
# chart.Correlation(tableau[,22:31], histogram=TRUE, pch=19)
# pairs.panels(tableau[,22:31], scale=TRUE)
# ggpairs(tableau[,22:31], mapping = aes(color = tableau$Tapenade))
# ggscatmat(tableau, columns = 22:31, color="Tapenade", alpha=0.8)
# n=network::network(cor(tableau[,22:31],use="complete.obs"), directed =
FALSE);n
# ggnet2(n, label = TRUE, shape = 15)
# library(corrr)
# network_plot(correlate(cor(tableau[,23:31],use="complete.obs")),min_cor =
.4,colors = c("red", "green"), legend = TRUE )
#
#
# #Finalité en bouche
# #pairs(tableau[,32:37])
# kable(round(cor(tableau[,32:37],use="complete.obs"),digits=3))
# kable(round(matpvalue(tableau[,32:37]),digits=3))
# pairs(tableau[,32:37],lower.panel=panel.smooth, upper.panel=panel.cor)
# heatmap(x =cor(tableau[,32:37],use="complete.obs"), symm = TRUE)
# chart.Correlation(tableau[,32:37], histogram=TRUE, pch=19)
# pairs.panels(tableau[,32:37], scale=TRUE)
# ggpairs(tableau[,32:37], mapping = aes(color = tableau$Tapenade))
# ggscatmat(tableau, columns = 32:37, color="Tapenade", alpha=0.8)
# n=network::network(cor(tableau[,32:37],use="complete.obs"), directed =
FALSE);n
# ggnet2(n, label = TRUE, shape = 15)
# library(corrr)
# network_plot(correlate(cor(tableau[,32:37],use="complete.obs")),min_cor =
.4,colors = c("red", "green"), legend = TRUE )
#
# #26 variables
# n=network::network(cor(tableau[,12:37],use="complete.obs"), directed =
FALSE);n
# ggnet2(n, label = TRUE, shape = 15)
#
# library(corrr)
# network_plot(correlate(cor(tableau[,12:37],use="complete.obs")),min_cor =
.4,colors = c("red", "green"), legend = TRUE )

# # pairs(tableau[,12:37])
# # (round(cor(tableau[,12:37],use="complete.obs"),digits=3))
# # (round(matpvalue(tableau[,12:37]),digits=3))
# #
# #
write.table((round(cor(tableau[,12:37],use="complete.obs"),digits=3)),file =
"D:/Navigation/Téléchargements/Cours Distance/Projet
```

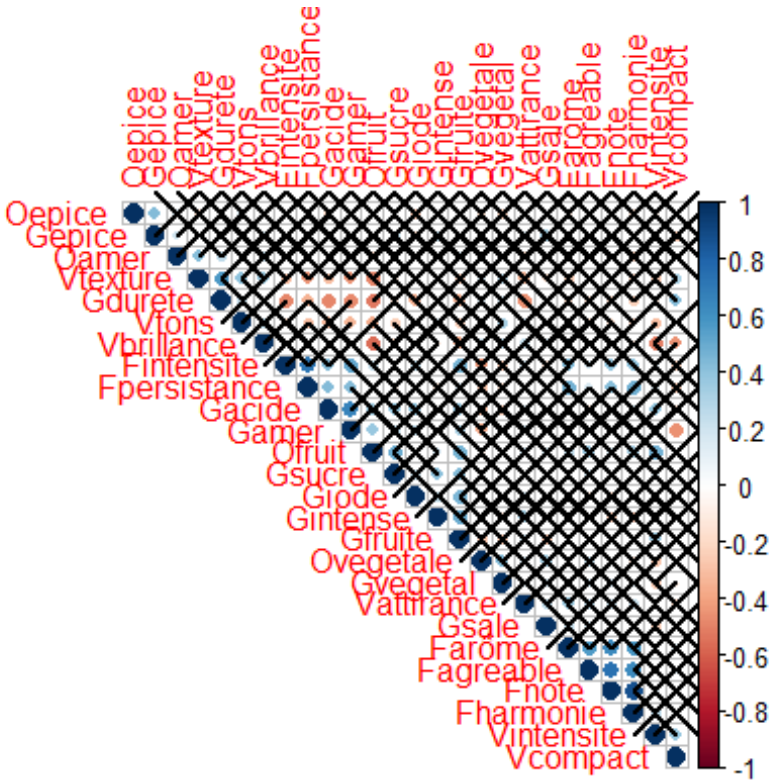
```
S4/traitement/inférientielle/Tableau_test/cor.txt", row.names =
TRUE,col.names =TRUE, sep = ";", dec = ".")
# #
# # #write.table((round(matpvalue(tableau[,12:37]),digits=3)),file =
"D:/Navigation/Téléchargements/Cours Distance/Projet
S4/traitement/inférientielle/Tableau_test/pvl.txt", row.names =
TRUE,col.names =TRUE, sep = ";", dec = ".")
# #
# #
#
corrplot(cor(tableau[,12:37],use="complete.obs"),method =
"number",tl.srt=45,tl.col="black",type="upper")
```




```
palette = colorRampPalette(c("green", "white", "red")) (20)

# corrplot.mixed(cor(tableau[,12:37],use="complete.obs"), order="hclust",
tl.col="black")
#

corrplot(cor(tableau[,12:37],use="complete.obs"), order = "hclust")
#classement
```

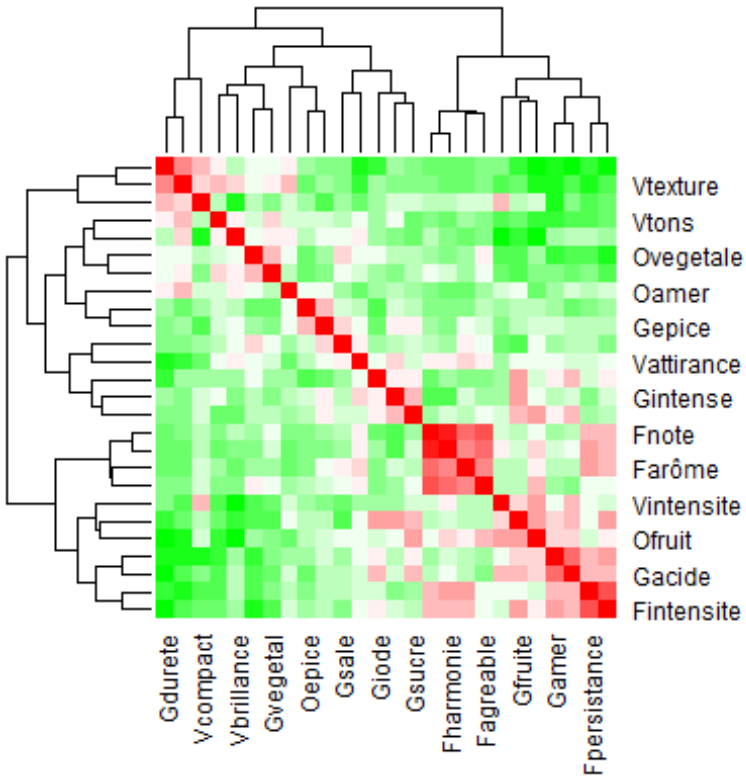



```
#

M=(round(cor(tableau[,12:37],use="complete.obs"),digits=3))
p_mat=(round(matpvalue(tableau[,12:37]),digits=3))
#
corrplot(M, type = "upper", order = "hclust", p.mat = p_mat, sig.level =
0.05)
```

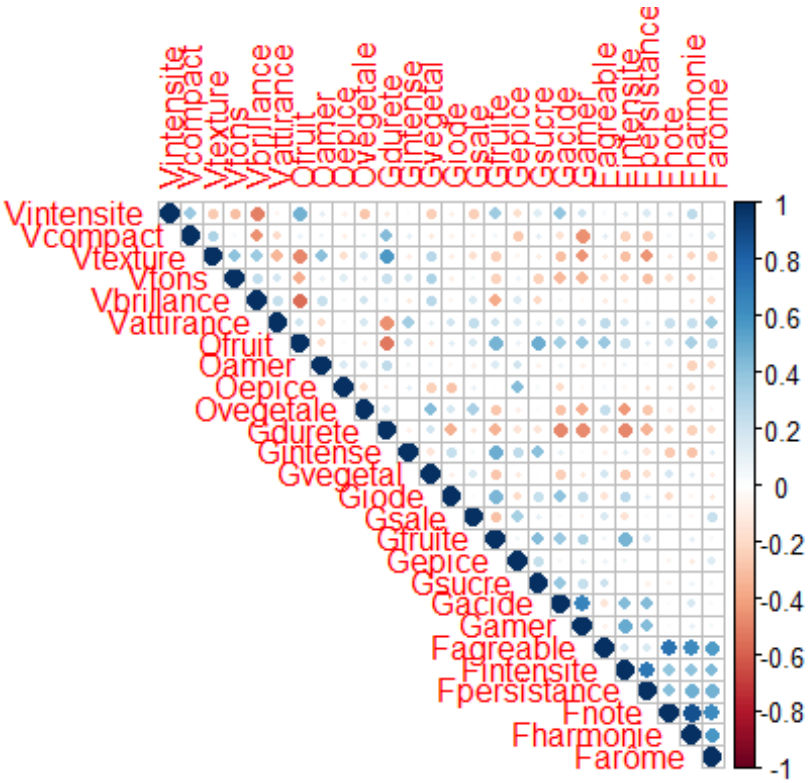



```
#
#
#   col=colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD",
"#4477AA"))
# # corrplot(M, method = "color", col = col(200),
# #          type = "upper", order = "hclust",
# #          addCoef.col = "black", # Add coefficient of correlation
# #          tl.col = "darkblue", tl.srt = 45, #Text label color and rotation
# #          # Combine with significance level
# #          p.mat = p_mat, sig.level = 0.05,
# #          # hide correlation coefficient on the principal diagonal
# #          diag = FALSE
# #          )
# #
# # col=colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD",
"#4477AA"))
# # corrplot(M, method = "color", col = col(200),
# #          order = "hclust",
# #          addCoef.col = "black", # Add coefficient of correlation
# #          tl.col = "darkblue", tl.srt = 45, #Text label color and rotation
# #          # Combine with significance level
# #          p.mat = p_mat, sig.level = 0.05,
# #          # hide correlation coefficient on the principal diagonal
# #          diag = FALSE
# #          )
# #
# #
```

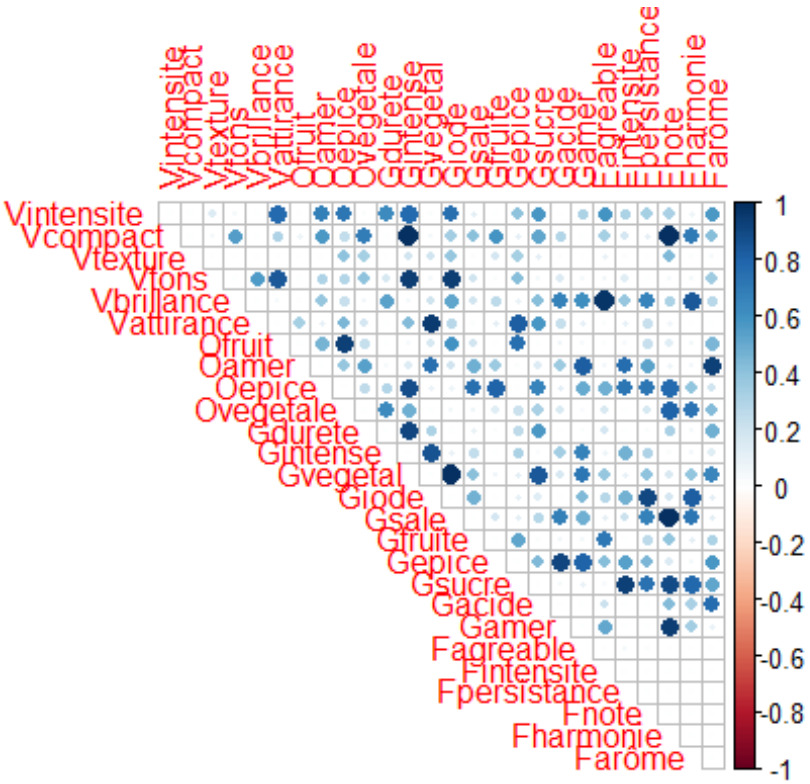
```
#
heatmap(x =cor(tableau[,12:37],use="complete.obs"), col = palette, symm =
TRUE)
```




```
#
#
corrplot(cor(tableau[,12:37],use="complete.obs"), type="upper") #correlation
des variables
```




```
#

corrplot(matpvalue(tableau[,12:37]), type="upper") # Pvalues des corrélations
```



```
#

t=symnum(cor(tableau[,12:37],use="complete.obs"), abbr.colnames=FALSE);t
#correlation

##              Vintensite Vcompact Vtexture Vtons Vbrillance Vattirance
Ofruit
## Vintensite   1
## Vcompact     .          1
## Vtexture                .        1
## Vtons        .                  .        1
## Vbrillance   .          .        .              1
## Vattirance                      .                       1
## Ofruit       .                  .        .     .                     1
## Oamer                           .
## Oepice
## Ovegetale
## Gdurete                 .       .                       .          .
## Gintense                                                .
## Gvegetal                                        .
## Giode
## Gsale
## Gfruite      .                                  .                    .
## Gepice
## Gsucre                                                               .
## Gacide       .                  .        .                          .
## Gamer                   .       .        .                          .
## Fagreable                                                            .
## Fintensite                      .
## Fpersistance                    .        .
## Fnote
## Fharmonie                                                            .
## Farôme                                                      .
##              Oamer Oepice Ovegetale Gdurete Gintense Gvegetal Giode Gsale
## Vintensite
## Vcompact
## Vtexture
## Vtons
## Vbrillance
## Vattirance
## Ofruit
## Oamer        1
## Oepice             1
## Ovegetale                1
## Gdurete                           1
## Gintense                                1
## Gvegetal                 .                      1
## Giode                             .                        1
## Gsale                    .                                      1
## Gfruite                           .       .                .
```

```
## Gepice             .                                                .
## Gsucre                                    .
## Gacide                              .                        .
## Gamer                      .         .
## Fagreable
## Fintensite                 .         .
## Fpersistance                         .
## Fnote
## Fharmonie
## Farôme
##              Gfruite Gepice Gsucre Gacide Gamer Fagreable Fintensite
## Vintensite
## Vcompact
## Vtexture
## Vtons
## Vbrillance
## Vattirance
## Ofruit
## Oamer
## Oepice
## Ovegetale
## Gdurete
## Gintense
## Gvegetal
## Giode
## Gsale
## Gfruite      1
## Gepice              1
## Gsucre       .              1
## Gacide       .              .      1
## Gamer        .                     ,      1
## Fagreable                                        1
## Fintensite   .                     .      .              1
## Fpersistance                       .      .              ,
## Fnote                                               ,         .
## Fharmonie                                           ,         .
## Farôme                                              .         .
##              Fpersistance Fnote Fharmonie Farôme
## Vintensite
## Vcompact
## Vtexture
## Vtons
## Vbrillance
## Vattirance
## Ofruit
## Oamer
## Oepice
## Ovegetale
## Gdurete
## Gintense
```

```
## Gvegetal
## Giode
## Gsale
## Gfruite
## Gepice
## Gsucre
## Gacide
## Gamer
## Fagreable
## Fintensite
## Fpersistance 1
## Fnote        .           1
## Fharmonie    .           +     1
## Farôme       .           ,     .         1
## attr(,"legend")
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
```

```
t1=symnum(matpvalue(tableau[,12:37]), abbr.colnames=FALSE);t1 #Pvalues des
corre
```

```
##              Vintensite Vcompact Vtexture Vtons Vbrillance Vattirance
Ofruit
## Vintensite
## Vcompact
## Vtexture
## Vtons                  .
## Vbrillance                                 .
## Vattirance   ,         .                   +
## Ofruit                                                   .
## Oamer        ,         .                   .     .                   .
## Oepice       ,                   .                      .            *
## Ovegetale              ,         .         .
## Gdurete      ,                                       .
## Gintense     ,         1                   *                 .
## Gvegetal                                                 B
## Giode        ,         .         .         *     .                   ,
## Gsale                  .
## Gfruite                .
## Gepice       .                   .         .                 +         ,
## Gsucre       .         .                               .         .
## Gacide                                                 ,
## Gamer        .                                         ,
## Fagreable    .         .                               B
## Fintensite   .                                         .
## Fpersistance .                                         ,
## Fnote        .         1         .                     .
## Fharmonie              ,                               +
## Farôme       .         .                   .                       .
##              Oamer Oepice Ovegetale Gdurete Gintense Gvegetal Giode Gsale
## Vintensite
```

```
## Vcompact
## Vtexture
## Vtons
## Vbrillance
## Vattirance
## Ofruit
## Oamer
## Oepice       .
## Ovegetale    .
## Gdurete                ,
## Gintense          +    .         *
## Gvegetal     ,                   .       +
## Giode                                          1
## Gsale        .     ,                           .        .
## Gfruite      .     +
## Gepice                                   .
## Gsucre            ,      .         .              +
## Gacide       .                              .                    ,
## Gamer        +     .                        ,        ,        .     .
## Fagreable         .                              .
## Fintensite   ,     ,                        .              .
## Fpersistance .     ,                        .        .        *     ,
## Fnote             ,      +         .                             1
## Fharmonie         .      ,                        .        +     ,
## Farôme       *             .         .              ,
##              Gfruite Gepice Gsucre Gacide Gamer Fagreable Fintensite
## Vintensite
## Vcompact
## Vtexture
## Vtons
## Vbrillance
## Vattirance
## Ofruit
## Oamer
## Oepice
## Ovegetale
## Gdurete
## Gintense
## Gvegetal
## Giode
## Gsale
## Gfruite
## Gepice       .
## Gsucre              .
## Gacide              *
## Gamer               +
## Fagreable    ,      .                   .
## Fintensite          .      *
## Fpersistance        .      ,
## Fnote        .              +      .      *
```

```
## Fharmonie                    ,      .      .
## Farôme       .       .      .      ,
##              Fpersistance Fnote Fharmonie Farôme
## Vintensite
## Vcompact
## Vtexture
## Vtons
## Vbrillance
## Vattirance
## Ofruit
## Oamer
## Oepice
## Ovegetale
## Gdurete
## Gintense
## Gvegetal
## Giode
## Gsale
## Gfruite
## Gepice
## Gsucre
## Gacide
## Gamer
## Fagreable
## Fintensite
## Fpersistance
## Fnote
## Fharmonie
## Farôme
## attr(,"legend")
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1

#
# # #write.table(t1,file = "D:/Navigation/Téléchargements/Cours
Distance/Projet S4/traitement/inférientielle/Tableau_test/pvlx.txt",
row.names = TRUE,col.names =TRUE, sep = ";", dec = ".")
# # #write.table(t,file = "D:/Navigation/Téléchargements/Cours
Distance/Projet S4/traitement/inférientielle/Tableau_test/corx.txt",
row.names = TRUE,col.names =TRUE, sep = ";", dec = ".")
#
#
```

#l'influence de la tapenade sur la correlation entre deux variables

```
#le nuage de point en plotant chaque tapenande d'une couleur/logo different
avec la droite d'ajustement (les 3 tapenades apparaissent dans le meme plot)

#le nuage de point pour chaque tapenade et voir si les correlations sont
differentes d'une tapenade à une autre la droite d'ajustement (3 plots
differents)
```

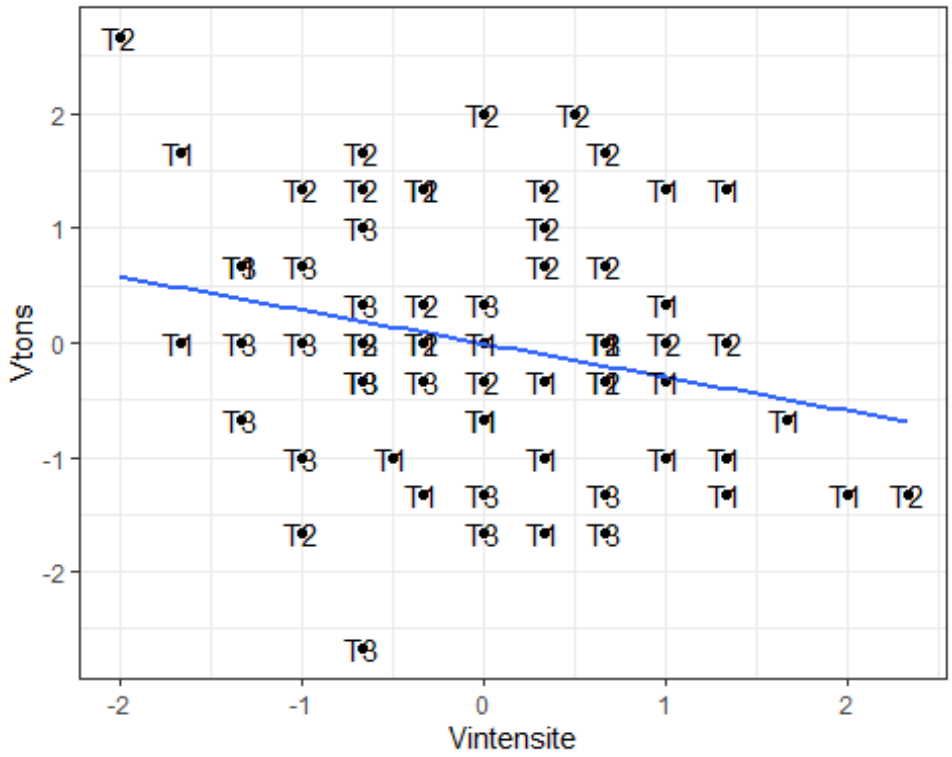
```r
#couple (Vintensite et Vtons) dans les caractéristiques visuelles


ggplot(tableau, aes(x=Vintensite, y=Vtons)) +
  geom_point()+
   geom_smooth(method=lm,se=FALSE)+
  geom_text(label=as.factor(tableau$Tapenade))+
   theme_bw()
```

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

```
## Warning: Removed 6 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```

```
## Warning: Removed 6 rows containing missing values (geom_text).
```

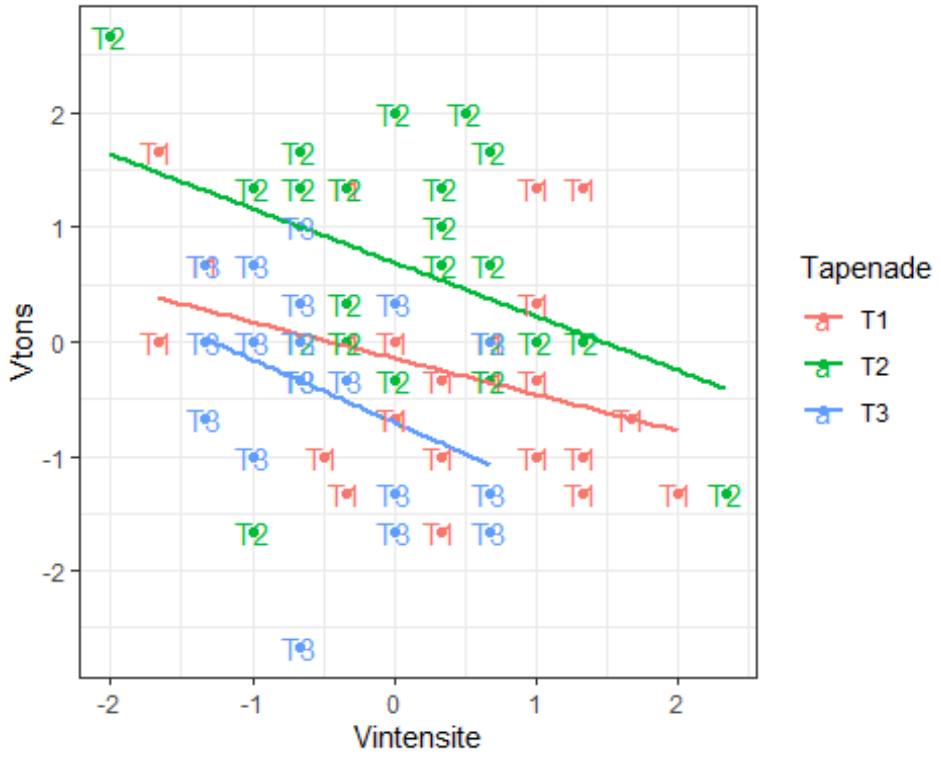



```r
ggplot(tableau, aes(x=Vintensite, y=Vtons, color=Tapenade)) +
  geom_point()+
   geom_smooth(method=lm,se=FALSE)+
  geom_text(label=as.factor(tableau$Tapenade))+
   theme_bw()
```

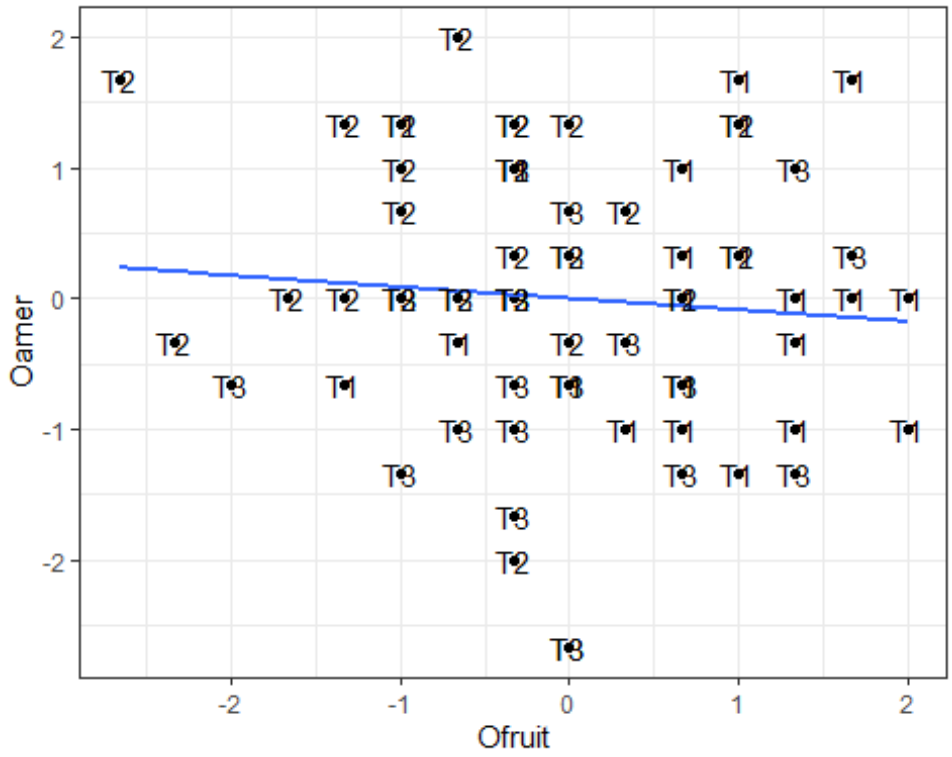
```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 6 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 6 rows containing missing values (geom_point).

## Warning: Removed 6 rows containing missing values (geom_text).
```

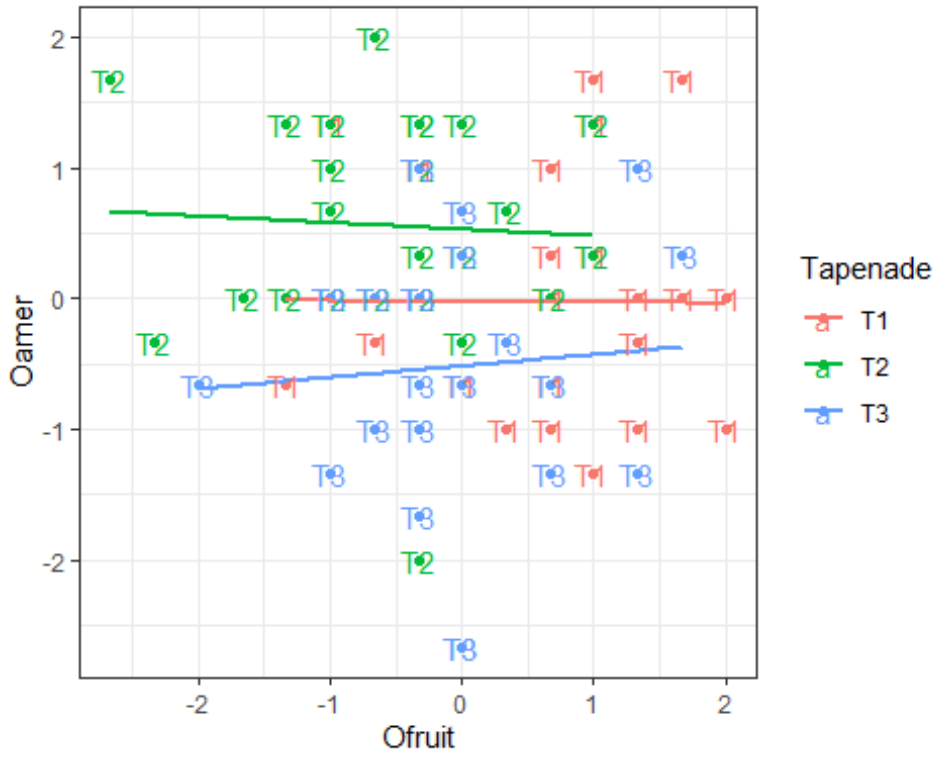



```
#couple (Gdurete et Gsale) dans les caractéristiques oflactives

ggplot(tableau, aes(x=Ofruit, y=Oamer)) +
  geom_point()+
   geom_smooth(method=lm,se=FALSE)+
  geom_text(label=as.factor(tableau$Tapenade))+
   theme_bw()

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

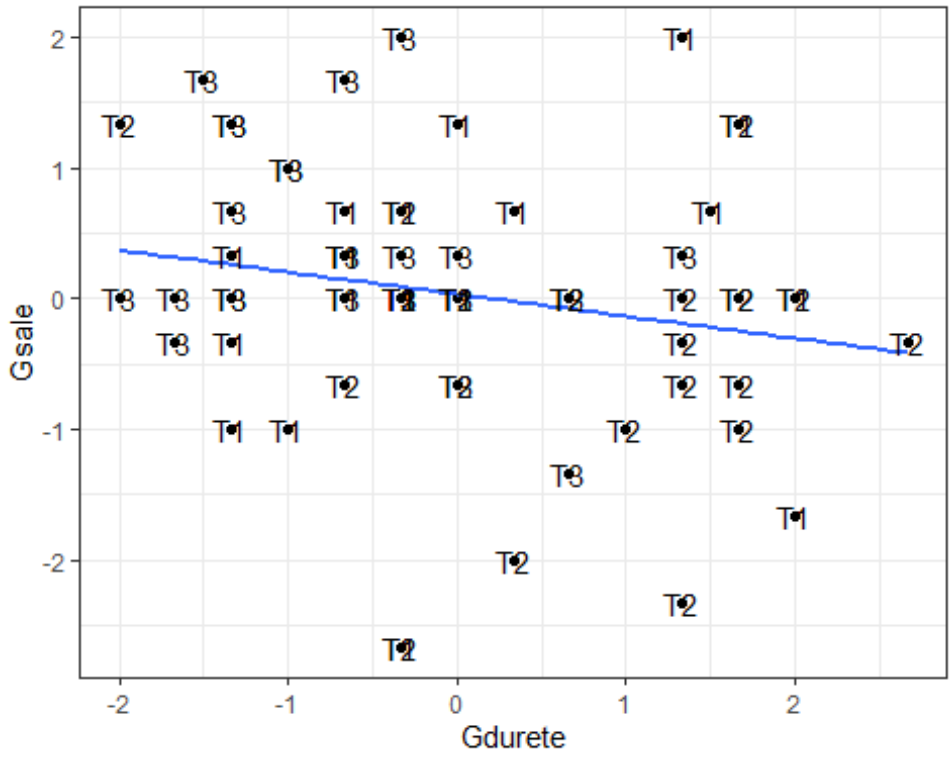



```
ggplot(tableau, aes(x=Ofruit, y=Oamer, color=Tapenade)) +
  geom_point()+
   geom_smooth(method=lm,se=FALSE)+
  geom_text(label=as.factor(tableau$Tapenade))+
   theme_bw()
```

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




```
#couple (Gdurete et Gsale) dans les caractéristiques gustatives

ggplot(tableau, aes(x=Gdurete, y=Gsale)) +
  geom_point()+
   geom_smooth(method=lm,se=FALSE)+
  geom_text(label=as.factor(tableau$Tapenade))+
   theme_bw()

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

## Warning: Removed 2 rows containing non-finite values (stat_smooth).

## Warning: Removed 2 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_text).
```



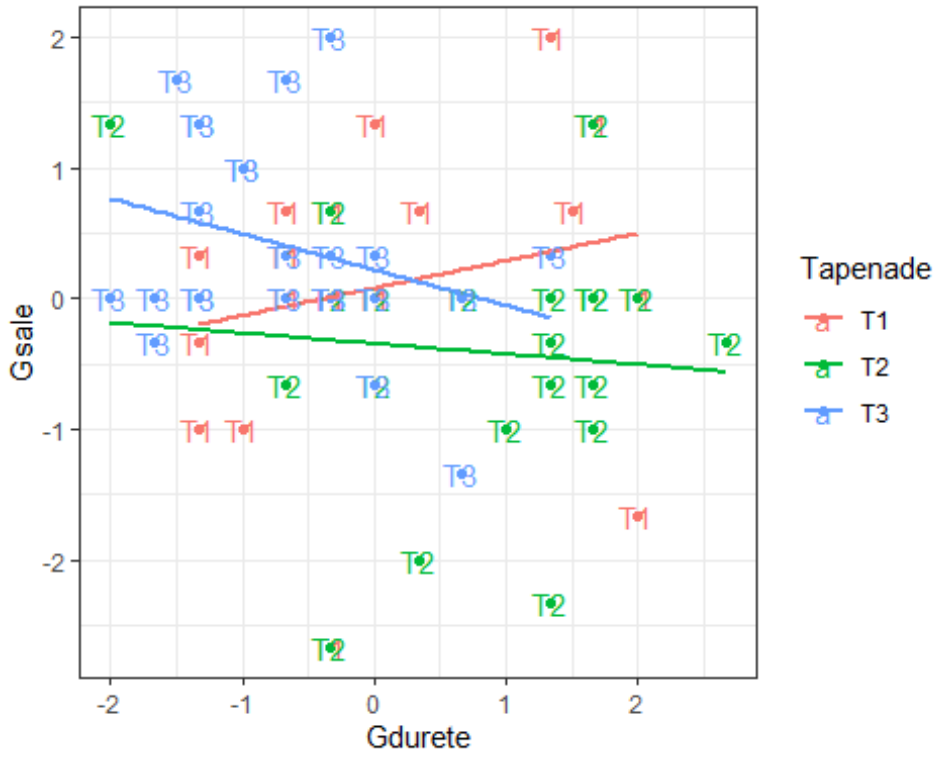

```
ggplot(tableau, aes(x=Gdurete, y=Gsale, color=Tapenade)) +
  geom_point()+
   geom_smooth(method=lm,se=FALSE)+
  geom_text(label=as.factor(tableau$Tapenade))+
  theme_bw()
```

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

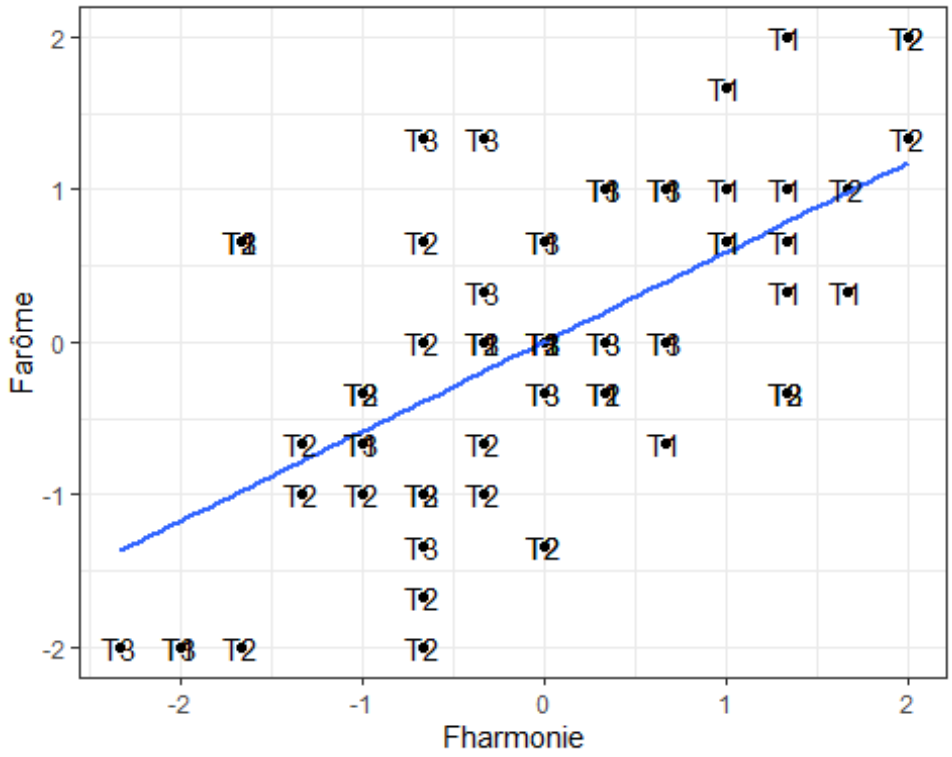
```
## Warning: Removed 2 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```

```
## Warning: Removed 2 rows containing missing values (geom_text).
```

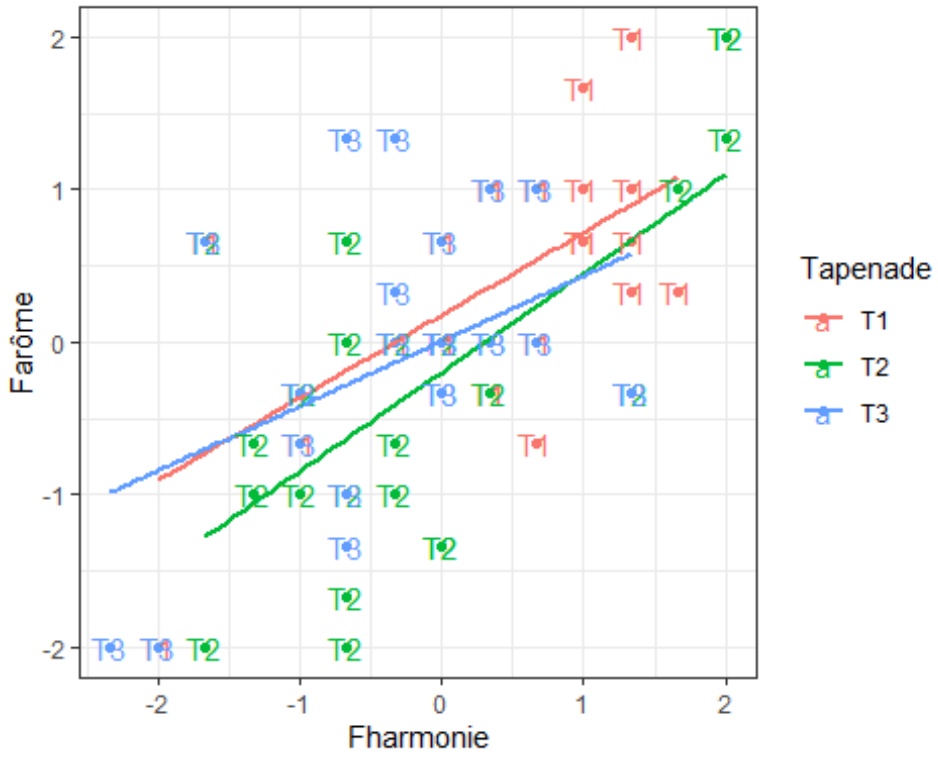



```
#couple (Gdurete et Gsale) dans les caractéristiques finalités en bouche
ggplot(tableau, aes(x=Fharmonie, y=Farôme)) +
  geom_point()+
   geom_smooth(method=lm,se=FALSE)+
  geom_text(label=as.factor(tableau$Tapenade))+
  theme_bw()

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




```
ggplot(tableau, aes(x=Fharmonie, y=Farôme, color=Tapenade)) +
  geom_point()+
   geom_smooth(method=lm,se=FALSE)+
  geom_text(label=as.factor(tableau$Tapenade))+
    theme_bw()
```

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




```
#Conclure sur l'influence de la tapenade sur la correlation entre deux
variables
#la corrélation entre deux variables varie sur l'effet de la tapenade.
```

#personl

#Prépration du tableau et Réalisation de l'ACP

```
#Transformation
tab=data.frame(Place=tableau$"No
place",sexe=tableau$sexe,age=tableau$âge,CSP=tableau$CSP,
fumeur=tableau$fumeur,fumedanslheure=tableau$fumedanslheure,

mangedanslheure=tableau$mangedanslheure,consomme=tableau$consomme,

apprecieOlives=tableau$apprecieOlives,frequenceOlives=tableau$frequenceOlives
,
                Tapenade=tableau$Tapenade,

Vintensite=tableau$Vintensite,Vcompact=tableau$Vcompact,Vtexture=tableau$Vtex
ture,

Vtons=tableau$Vtons,Vbrillance=tableau$Vbrillance,Vattirance=tableau$Vattiran
ce,

Ofruit=tableau$Ofruit,Oamer=tableau$Oamer,Oepice=tableau$Oepice,
```

```
Ovegetal=tableau$Ovegetale,Gacide=tableau$Gacide,Gdurete=tableau$Gdurete,

Gintense=tableau$Gintense,Gvegetal=tableau$Gvegetal,Giode=tableau$Giode,

Gsale=tableau$Gsale,Gfruit=tableau$Gfruite,Gepice=tableau$Gepice,

Gsucre=tableau$Gsucre,Gamer=tableau$Gamer,Fagreable=tableau$Fagreable,
               Fintense=tableau$Fintensite,Fpersistance=tableau$Fpersistance,

Fnote=tableau$Fnote,Fharmonie=tableau$Fharmonie,Farome=tableau$Farôme,

Vcouleur=tableau$Vcouleur,Gtexture=tableau$Gtexture,Tapmoins=tableau$Tap.moin
s,
            Tapplus=tableau$Tap.plus,Ressenti=tableau$Ressenti.general)


mean(tab[[12]],na.rm=TRUE)

## [1] 6.625932e-17

meancomplet=function(x) return(mean(x,na.rm=TRUE))

for (i in 12:37) {print(paste("la moyenne des valeurs de",
colnames(tab)[i],"=",meancomplet(tab[[i]])))}

## [1] "la moyenne des valeurs de Vintensite = 6.6259316648644e-17"
## [1] "la moyenne des valeurs de Vcompact = 3.07161335386296e-17"
## [1] "la moyenne des valeurs de Vtexture = 0"
## [1] "la moyenne des valeurs de Vtons = -9.30270627978713e-18"
## [1] "la moyenne des valeurs de Vbrillance = -1.85738639537549e-17"
## [1] "la moyenne des valeurs de Vattirance = 6.2257967342879e-18"
## [1] "la moyenne des valeurs de Ofruit = -2.15515298508507e-17"
## [1] "la moyenne des valeurs de Oamer = -3.02371939215402e-18"
## [1] "la moyenne des valeurs de Oepice = 3.39400455078483e-17"
## [1] "la moyenne des valeurs de Ovegetal = -3.44102992913282e-17"
## [1] "la moyenne des valeurs de Gacide = -3.80592348824222e-18"
## [1] "la moyenne des valeurs de Gdurete = -5.08397887280917e-17"
## [1] "la moyenne des valeurs de Gintense = -1.27207449505979e-17"
## [1] "la moyenne des valeurs de Gvegetal = -4.06133401694799e-17"
## [1] "la moyenne des valeurs de Giode = 6.35832633114414e-18"
## [1] "la moyenne des valeurs de Gsale = 1.25891443786227e-17"
## [1] "la moyenne des valeurs de Gfruit = -2.53764377708466e-17"
## [1] "la moyenne des valeurs de Gepice = 6.19534537697427e-18"
## [1] "la moyenne des valeurs de Gsucre = -9.23127877508376e-18"
## [1] "la moyenne des valeurs de Gamer = -9.22274570242993e-18"
## [1] "la moyenne des valeurs de Fagreable = -5.07654434362641e-17"
## [1] "la moyenne des valeurs de Fintense = -1.27176472301051e-17"
## [1] "la moyenne des valeurs de Fpersistance = 4.57092507947128e-17"
## [1] "la moyenne des valeurs de Fnote = -1.24479543640617e-17"
```

```
## [1] "la moyenne des valeurs de Fharmonie = 3.69674040045165e-17"
## [1] "la moyenne des valeurs de Farome = -1.5357355679927e-17"

quanti=tab[,12:37];quanti[is.na(quanti[,1]),1];meancomplet(quanti[,1])

## [1] NA NA NA NA NA

## [1] 6.625932e-17

#affcihage des NA

 for (j in 1:26) {
  mj=meancomplet(quanti[1:24,j])
  print(paste("moyenne colonne",j,"=",mj))
  for (i in 1:24)
    if (is.na(quanti[i,j])) (quanti[i,j]=mj)
 }

## [1] "moyenne colonne 1 = 0.326388888888889"
## [1] "moyenne colonne 2 = 0.361111111111111"
## [1] "moyenne colonne 3 = -0.347826086956522"
## [1] "moyenne colonne 4 = -0.25"
## [1] "moyenne colonne 5 = -0.583333333333333"
## [1] "moyenne colonne 6 = 0.138888888888889"
## [1] "moyenne colonne 7 = 0.680555555555556"
## [1] "moyenne colonne 8 = -0.0277777777777778"
## [1] "moyenne colonne 9 = 0.180555555555556"
## [1] "moyenne colonne 10 = -0.318840579710145"
## [1] "moyenne colonne 11 = -0.0833333333333333"
## [1] "moyenne colonne 12 = -0.0486111111111112"
## [1] "moyenne colonne 13 = 0.231884057971014"
## [1] "moyenne colonne 14 = -0.277777777777778"
## [1] "moyenne colonne 15 = -0.319444444444444"
## [1] "moyenne colonne 16 = 0.0694444444444445"
## [1] "moyenne colonne 17 = 0.072463768115942"
## [1] "moyenne colonne 18 = -0.0277777777777778"
## [1] "moyenne colonne 19 = -9.24658811131561e-18"
## [1] "moyenne colonne 20 = -0.125"
## [1] "moyenne colonne 21 = 0.231884057971014"
## [1] "moyenne colonne 22 = 0.188405797101449"
## [1] "moyenne colonne 23 = 0.196969696969697"
## [1] "moyenne colonne 24 = 0.388888888888889"
## [1] "moyenne colonne 25 = 0.416666666666667"
## [1] "moyenne colonne 26 = 0.402777777777778"

 for (j in 1:26) {
  mj=meancomplet(quanti[25:48,j])
  print(paste("moyenne colonne",j,"=",mj))
  for (i in 25:48)
    if (is.na(quanti[i,j])) (quanti[i,j]=mj)
 }
```

```
## [1] "moyenne colonne 1 = 0.0833333333333334"
## [1] "moyenne colonne 2 = 0.236111111111111"
## [1] "moyenne colonne 3 = 0.833333333333333"
## [1] "moyenne colonne 4 = 0.583333333333333"
## [1] "moyenne colonne 5 = 0.291666666666667"
## [1] "moyenne colonne 6 = -0.402777777777778"
## [1] "moyenne colonne 7 = -0.569444444444444"
## [1] "moyenne colonne 8 = 0.555555555555556"
## [1] "moyenne colonne 9 = -0.152777777777778"
## [1] "moyenne colonne 10 = 0.152777777777778"
## [1] "moyenne colonne 11 = -0.133333333333333"
## [1] "moyenne colonne 12 = 0.787878787878788"
## [1] "moyenne colonne 13 = -0.311594202898551"
## [1] "moyenne colonne 14 = 0.231884057971014"
## [1] "moyenne colonne 15 = 0.0579710144927536"
## [1] "moyenne colonne 16 = -0.492753623188406"
## [1] "moyenne colonne 17 = -0.0763888888888889"
## [1] "moyenne colonne 18 = -0.194444444444444"
## [1] "moyenne colonne 19 = -0.125"
## [1] "moyenne colonne 20 = -0.25"
## [1] "moyenne colonne 21 = -0.289855072463768"
## [1] "moyenne colonne 22 = -0.246376811594203"
## [1] "moyenne colonne 23 = -0.289855072463768"
## [1] "moyenne colonne 24 = -0.111111111111111"
## [1] "moyenne colonne 25 = -0.166666666666667"
## [1] "moyenne colonne 26 = -0.305555555555556"
```

```
 for (j in 1:26) {
  mj=meancomplet(quanti[49:72,j])
  print(paste("moyenne colonne",j,"=",mj))
  for (i in 49:72)
    if (is.na(quanti[i,j])) (quanti[i,j]=mj)
 }
```

```
## [1] "moyenne colonne 1 = -0.46031746031746"
## [1] "moyenne colonne 2 = -0.597222222222222"
## [1] "moyenne colonne 3 = -0.5"
## [1] "moyenne colonne 4 = -0.347826086956522"
## [1] "moyenne colonne 5 = 0.291666666666667"
## [1] "moyenne colonne 6 = 0.263888888888889"
## [1] "moyenne colonne 7 = -0.111111111111111"
## [1] "moyenne colonne 8 = -0.527777777777778"
## [1] "moyenne colonne 9 = -0.0277777777777777"
## [1] "moyenne colonne 10 = 0.152777777777778"
## [1] "moyenne colonne 11 = 0.240740740740741"
## [1] "moyenne colonne 12 = -0.673611111111111"
## [1] "moyenne colonne 13 = 0.0763888888888889"
## [1] "moyenne colonne 14 = 0.0555555555555555"
## [1] "moyenne colonne 15 = 0.263888888888889"
## [1] "moyenne colonne 16 = 0.402777777777778"
```

```
## [1] "moyenne colonne 17 = 0.00724637681159418"
## [1] "moyenne colonne 18 = 0.222222222222222"
## [1] "moyenne colonne 19 = 0.125"
## [1] "moyenne colonne 20 = 0.375"
## [1] "moyenne colonne 21 = 0.0555555555555555"
## [1] "moyenne colonne 22 = 0.0555555555555555"
## [1] "moyenne colonne 23 = 0.101449275362319"
## [1] "moyenne colonne 24 = -0.277777777777778"
## [1] "moyenne colonne 25 = -0.25"
## [1] "moyenne colonne 26 = -0.0972222222222222"
```

```
#Tableau d'ACP
rownames(quanti)=paste(tab$Place,tab$Tapenade,sep="")

#Tableau d'analyse Discriminante
quantii=quanti; quantii$Tapenade=tab$Tapenade
```

#DataViz

#Nuage de mots de la colonne ressentie

```
b=sample(seq(0,1,0.01) , length(tableau$Ressenti.general) , replace=TRUE)
par(bg="black")
wordcloud(tableau$Ressenti.general, b ,
col=terrain.colors(length(tableau$Ressenti.general) , alpha=0.9) ,
rot.per=0.3 )
```

```
## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : b trop le gout d'olive
could
## not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : Faire la degustation
avec un
## bon eclairage could not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : tb à faire sans pain
could
## not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : Interessant de
s'attarder sur
## les saveurs could not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : tb à faire sans pain
could
## not be fit on page. It will not be plotted.
```

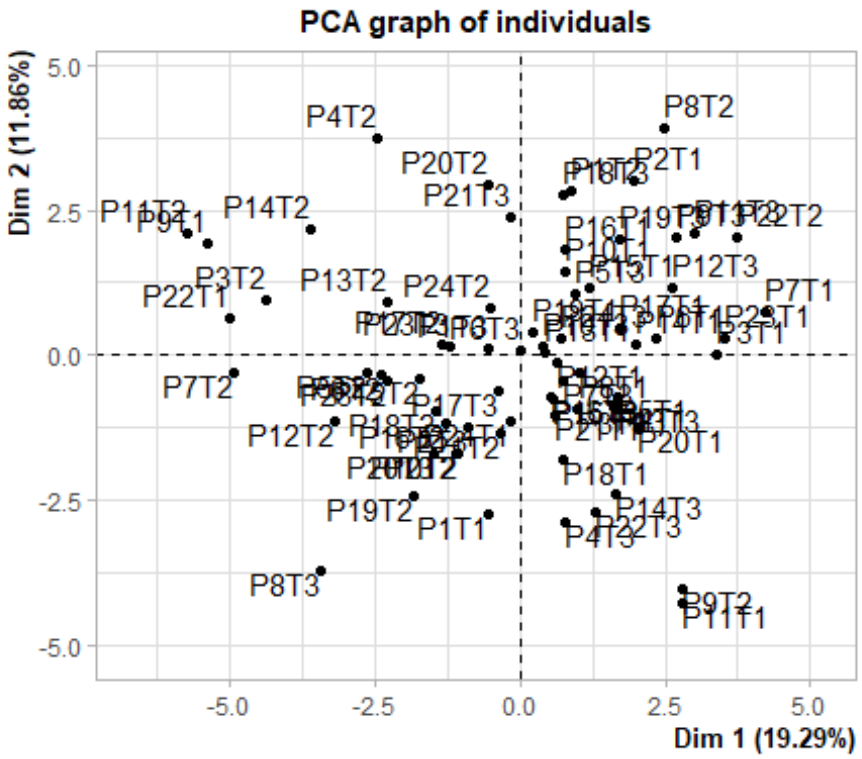
```
## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : bonne degustati could
not be
## fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : bon travail could not
be fit
## on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : a l'odeur, je n'aimais
pas le
## B, au final j'ai apprecie could not be fit on page. It will not be
plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : Interessant de
s'attarder sur
## les saveurs could not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : bonne degustati could
not be
## fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : nice as tu des bonbons
could
## not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : Faire la degustation
avec un
## bon eclairage could not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : a l'odeur, je n'aimais
pas le
## B, au final j'ai apprecie could not be fit on page. It will not be
plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : Faire la degustation
avec un
## bon eclairage could not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : b trop le gout d'olive
could
## not be fit on page. It will not be plotted.
```

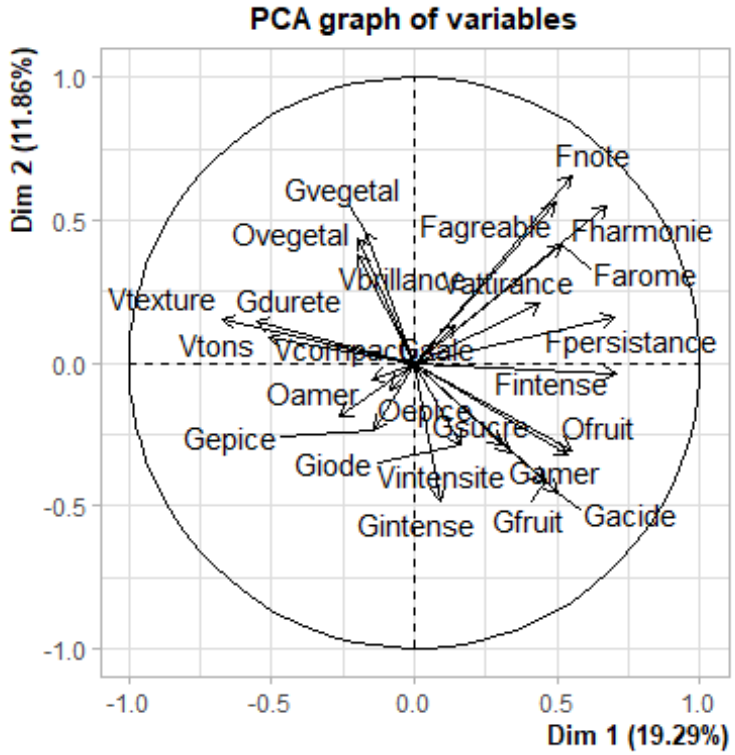
```
## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : super! could not be fit
on
## page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : très bonne experience,
## victoire tapenade C! could not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : mettre les questions
par
## tapenade could not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : attention organisation
des
## questions could not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : la A n'a pas trop de
goût
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : mettre les questions
par
## tapenade could not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : b trop le gout d'olive
could
## not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : J'ai bien aime could
not be
## fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : la A n'a pas trop de
goût
## could not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : très bonne experience,
## victoire tapenade C! could not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : bon travail could not
be fit
## on page. It will not be plotted.
```

```
## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : bon travail could not
be fit
## on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : tb à faire sans pain
could
## not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : attention organisation
des
## questions could not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : bon travail could not
be fit
## on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : très bien could not be
fit on
## page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : a l'odeur, je n'aimais
pas le
## B, au final j'ai apprecie could not be fit on page. It will not be
plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : nice as tu des bonbons
could
## not be fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : bonne degustati could
not be
## fit on page. It will not be plotted.

## Warning in wordcloud(tableau$Ressenti.general, b, col =
## terrain.colors(length(tableau$Ressenti.general), : attention organisation
des
## questions could not be fit on page. It will not be plotted.
```

#Réalisation de L'ACP

```
#Réalisation
acp=PCA(quanti);
```






```
#Factoshiny(acp)
summary(acp, ncp=4)
```

```
##
## Call:
## PCA(X = quanti)
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6
Dim.7
## Variance               5.016   3.084   2.494   2.069   1.859   1.541
1.276
## % of var.             19.291  11.863   9.594   7.958   7.148   5.926
4.908
## Cumulative % of var.  19.291  31.154  40.748  48.706  55.855  61.781
66.689
##                        Dim.8   Dim.9  Dim.10  Dim.11  Dim.12  Dim.13
Dim.14
## Variance               1.134   1.037   0.921   0.878   0.807   0.569
0.547
## % of var.              4.361   3.990   3.542   3.376   3.104   2.188
2.105
## Cumulative % of var.  71.050  75.040  78.582  81.958  85.062  87.249
89.354
##                       Dim.15  Dim.16  Dim.17  Dim.18  Dim.19  Dim.20
Dim.21
## Variance               0.499   0.395   0.376   0.323   0.266   0.240
0.199
## % of var.              1.920   1.520   1.447   1.241   1.025   0.924
0.764
## Cumulative % of var.  91.274  92.795  94.242  95.483  96.508  97.432
98.196
##                       Dim.22  Dim.23  Dim.24  Dim.25  Dim.26
## Variance               0.144   0.118   0.091   0.068   0.047
## % of var.              0.555   0.455   0.350   0.263   0.181
## Cumulative % of var.  98.750  99.206  99.556  99.819 100.000
##
## Individuals (the 10 first)
##                   Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2
Dim.3
## P1T1         |  5.955 | -0.532  0.078  0.008 | -2.736  3.370  0.211 | -
2.389
## P2T1         |  5.346 |  1.942  1.044  0.132 |  3.003  4.060  0.315 | -
0.159
## P3T1         |  4.974 |  3.388  3.179  0.464 |  0.001  0.000  0.000 | -
2.356
## P4T1         |  4.400 |  1.665  0.768  0.143 | -0.846  0.322  0.037 | -
1.836
## P5T1         |  4.725 |  1.667  0.770  0.125 | -0.716  0.231  0.023 | -
2.254
## P6T1         |  4.555 |  2.354  1.534  0.267 |  0.298  0.040  0.004 | -
0.306
```

```
## P7T1          |  6.488 |  4.213  4.916  0.422 |  0.729  0.239  0.013 | -
0.972
## P8T1          |  3.441 |  1.020  0.288  0.088 | -0.296  0.039  0.007 |
1.249
## P9T1          |  7.191 | -5.374  7.998  0.559 |  1.918  1.657  0.071 |
1.059
## P10T1         |  3.000 |  0.778  0.168  0.067 |  1.440  0.933  0.230 | -
0.155
##                 ctr   cos2    Dim.4    ctr   cos2
## P1T1          3.177  0.161 |  0.565  0.214  0.009 |
## P2T1          0.014  0.001 |  0.250  0.042  0.002 |
## P3T1          3.091  0.224 |  0.216  0.031  0.002 |
## P4T1          1.877  0.174 |  1.519  1.548  0.119 |
## P5T1          2.828  0.227 |  0.000  0.000  0.000 |
## P6T1          0.052  0.005 |  0.896  0.539  0.039 |
## P7T1          0.527  0.022 |  0.932  0.583  0.021 |
## P8T1          0.869  0.132 |  1.077  0.779  0.098 |
## P9T1          0.624  0.022 | -0.733  0.361  0.010 |
## P10T1         0.013  0.003 | -0.595  0.237  0.039 |
##
## Variables (the 10 first)
##                 Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr
## Vintensite    |  0.335  2.243  0.112 | -0.315  3.208  0.099 | -0.562 12.673
## Vcompact      | -0.150  0.447  0.022 | -0.058  0.109  0.003 | -0.742 22.083
## Vtexture      | -0.672  8.992  0.451 |  0.152  0.748  0.023 | -0.158  0.995
## Vtons         | -0.510  5.191  0.260 |  0.086  0.241  0.007 |  0.118  0.559
## Vbrillance    | -0.197  0.777  0.039 |  0.380  4.683  0.144 |  0.561 12.602
## Vattirance    |  0.440  3.863  0.194 |  0.207  1.384  0.043 |  0.351  4.938
## Ofruit        |  0.552  6.079  0.305 | -0.304  3.005  0.093 | -0.287  3.299
## Oamer         | -0.260  1.350  0.068 | -0.190  1.172  0.036 |  0.081  0.264
## Oepice        | -0.085  0.143  0.007 | -0.092  0.277  0.009 |  0.154  0.957
## Ovegetal      | -0.195  0.762  0.038 |  0.436  6.168  0.190 |  0.307  3.789
##                cos2    Dim.4    ctr   cos2
## Vintensite    0.316 | -0.037  0.068  0.001 |
## Vcompact      0.551 |  0.273  3.612  0.075 |
## Vtexture      0.025 | -0.215  2.241  0.046 |
## Vtons         0.014 | -0.061  0.183  0.004 |
## Vbrillance    0.314 | -0.251  3.043  0.063 |
## Vattirance    0.123 |  0.240  2.790  0.058 |
## Ofruit        0.082 |  0.307  4.563  0.094 |
## Oamer         0.007 | -0.409  8.076  0.167 |
## Oepice        0.024 | -0.309  4.628  0.096 |
## Ovegetal      0.095 |  0.461 10.290  0.213 |
```

```
kable(summary(acp, ncp=2) )
```

```
##
## Call:
## PCA(X = quanti)
##
```

```
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6
Dim.7
## Variance               5.016   3.084   2.494   2.069   1.859   1.541
1.276
## % of var.             19.291  11.863   9.594   7.958   7.148   5.926
4.908
## Cumulative % of var.  19.291  31.154  40.748  48.706  55.855  61.781
66.689
##                        Dim.8   Dim.9  Dim.10  Dim.11  Dim.12  Dim.13
Dim.14
## Variance               1.134   1.037   0.921   0.878   0.807   0.569
0.547
## % of var.              4.361   3.990   3.542   3.376   3.104   2.188
2.105
## Cumulative % of var.  71.050  75.040  78.582  81.958  85.062  87.249
89.354
##                       Dim.15  Dim.16  Dim.17  Dim.18  Dim.19  Dim.20
Dim.21
## Variance               0.499   0.395   0.376   0.323   0.266   0.240
0.199
## % of var.              1.920   1.520   1.447   1.241   1.025   0.924
0.764
## Cumulative % of var.  91.274  92.795  94.242  95.483  96.508  97.432
98.196
##                       Dim.22  Dim.23  Dim.24  Dim.25  Dim.26
## Variance               0.144   0.118   0.091   0.068   0.047
## % of var.              0.555   0.455   0.350   0.263   0.181
## Cumulative % of var.  98.750  99.206  99.556  99.819 100.000
##
## Individuals (the 10 first)
##                  Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2
## P1T1         |  5.955 | -0.532  0.078  0.008 | -2.736  3.370  0.211 |
## P2T1         |  5.346 |  1.942  1.044  0.132 |  3.003  4.060  0.315 |
## P3T1         |  4.974 |  3.388  3.179  0.464 |  0.001  0.000  0.000 |
## P4T1         |  4.400 |  1.665  0.768  0.143 | -0.846  0.322  0.037 |
## P5T1         |  4.725 |  1.667  0.770  0.125 | -0.716  0.231  0.023 |
## P6T1         |  4.555 |  2.354  1.534  0.267 |  0.298  0.040  0.004 |
## P7T1         |  6.488 |  4.213  4.916  0.422 |  0.729  0.239  0.013 |
## P8T1         |  3.441 |  1.020  0.288  0.088 | -0.296  0.039  0.007 |
## P9T1         |  7.191 | -5.374  7.998  0.559 |  1.918  1.657  0.071 |
## P10T1        |  3.000 |  0.778  0.168  0.067 |  1.440  0.933  0.230 |
##
## Variables (the 10 first)
##                 Dim.1    ctr   cos2    Dim.2    ctr   cos2
## Vintensite   |  0.335  2.243  0.112 | -0.315  3.208  0.099 |
## Vcompact     | -0.150  0.447  0.022 | -0.058  0.109  0.003 |
## Vtexture     | -0.672  8.992  0.451 |  0.152  0.748  0.023 |
## Vtons        | -0.510  5.191  0.260 |  0.086  0.241  0.007 |
```

```
## Vbrillance    | -0.197  0.777  0.039 |  0.380  4.683  0.144 |
## Vattirance    |  0.440  3.863  0.194 |  0.207  1.384  0.043 |
## Ofruit        |  0.552  6.079  0.305 | -0.304  3.005  0.093 |
## Oamer         | -0.260  1.350  0.068 | -0.190  1.172  0.036 |
## Oepice        | -0.085  0.143  0.007 | -0.092  0.277  0.009 |
## Ovegetal      | -0.195  0.762  0.038 |  0.436  6.168  0.190 |

## Warning in kable_markdown(x, padding = padding, ...): The table should
have a
## header (column names)
```

|| || || ||

```
#plot(acp, selectRow="cos2 0.03846154",selectCol="cos2 0.03846154")

#plot(acp, selectRow="contrib 0.01388889")

#individus
#t=data.frame(acp$var$coord[,1]>0&acp$var$cos2[,1]> 0.04)
#colnames(t)="tt"
#subset(t,t$tt=="TRUE")
#vARIABLES
#t=data.frame(acp$var$coord[,3]<0&acp$var$cos2[,3]> 0.04)
#colnames(t)="tt"
#subset(t,t$tt=="TRUE")


axe1=data.frame(cos2=acp$var$cos2[,1]>0.04,signe=round(acp$var$coord[,1],digi
ts=3));kable(axe1)
```

	cos2	signe
Vintensite	TRUE	0.335
Vcompact	FALSE	-0.150
Vtexture	TRUE	-0.672
Vtons	TRUE	-0.510
Vbrillance	FALSE	-0.197
Vattirance	TRUE	0.440
Ofruit	TRUE	0.552
Oamer	TRUE	-0.260
Oepice	FALSE	-0.085
Ovegetal	FALSE	-0.195
Gacide	TRUE	0.505
Gdurete	TRUE	-0.555
Gintense	FALSE	0.090

Gvegetal	FALSE	-0.167
Giode	FALSE	0.164
Gsale	FALSE	0.144
Gfruit	TRUE	0.456
Gepice	FALSE	-0.139
Gsucre	TRUE	0.308
Gamer	TRUE	0.538
Fagreable	TRUE	0.492
Fintense	TRUE	0.709
Fpersistance	TRUE	0.701
Fnote	TRUE	0.549
Fharmonie	TRUE	0.675
Farome	TRUE	0.517

```
axe2=data.frame(cos2=acp$var$cos2[,2]>0.04,signe=round(acp$var$coord[,2],digi
ts=3));kable(axe2)
```

	cos2	signe
Vintensite	TRUE	-0.315
Vcompact	FALSE	-0.058
Vtexture	FALSE	0.152
Vtons	FALSE	0.086
Vbrillance	TRUE	0.380
Vattirance	TRUE	0.207
Ofruit	TRUE	-0.304
Oamer	FALSE	-0.190
Oepice	FALSE	-0.092
Ovegetal	TRUE	0.436
Gacide	TRUE	-0.455
Gdurete	FALSE	0.143
Gintense	TRUE	-0.486
Gvegetal	TRUE	0.460
Giode	TRUE	-0.289
Gsale	FALSE	0.129
Gfruit	TRUE	-0.416
Gepice	TRUE	-0.236
Gsucre	TRUE	-0.288
Gamer	TRUE	-0.325

Fagreable	TRUE	0.561
Fintense	FALSE	-0.039
Fpersistance	FALSE	0.158
Fnote	TRUE	0.657
Fharmonie	TRUE	0.547
Farome	TRUE	0.413

```
axe3=data.frame(cos2=acp$var$cos2[,3]>0.04,signe=round(acp$var$coord[,3],digi
ts=3));kable(axe3)
```

	cos2	signe
Vintensite	TRUE	-0.562
Vcompact	TRUE	-0.742
Vtexture	FALSE	-0.158
Vtons	FALSE	0.118
Vbrillance	TRUE	0.561
Vattirance	TRUE	0.351
Ofruit	TRUE	-0.287
Oamer	FALSE	0.081
Oepice	FALSE	0.154
Ovegetal	TRUE	0.307
Gacide	TRUE	0.260
Gdurete	TRUE	-0.482
Gintense	FALSE	0.113
Gvegetal	TRUE	0.239
Giode	TRUE	0.299
Gsale	TRUE	0.398
Gfruit	FALSE	-0.028
Gepice	TRUE	0.344
Gsucre	FALSE	-0.007
Gamer	TRUE	0.274
Fagreable	TRUE	-0.206
Fintense	FALSE	-0.010
Fpersistance	FALSE	0.053
Fnote	TRUE	-0.241
Fharmonie	FALSE	-0.165
Farome	FALSE	-0.009

#Classification pour obtenir des profils

```r
#classification Hiérarchique
cp=acp$ind$coord[,1:3]
summary(cp)
```

```
##      Dim.1              Dim.2              Dim.3
##  Min.   :-5.7212   Min.   :-4.28066   Min.   :-2.7920
##  1st Qu.:-1.3016   1st Qu.:-1.05624   1st Qu.:-1.2266
##  Median : 0.5488   Median : 0.02351   Median : 0.0871
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 1.6380   3rd Qu.: 1.07508   3rd Qu.: 0.9753
##  Max.   : 4.2134   Max.   : 3.91782   Max.   : 3.8037
```

```r
n=nrow(cp)
apply(cp,2,mean)
```

```
##         Dim.1         Dim.2         Dim.3
## -1.485929e-16 -1.428361e-16  1.628411e-17
```

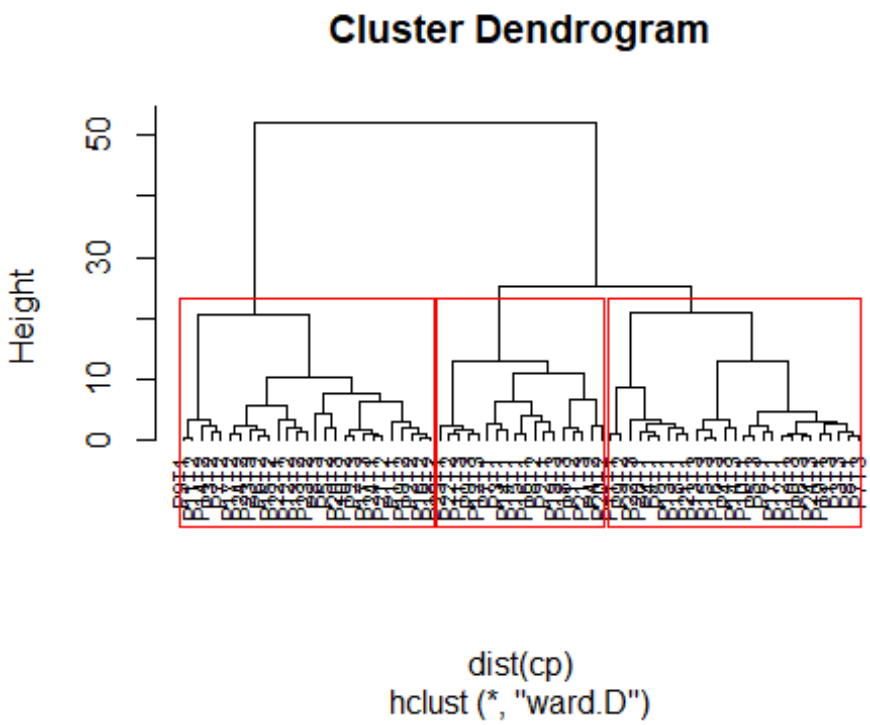
```r
apply(cp,2,var)
```

```
##    Dim.1    Dim.2    Dim.3
## 5.086293 3.127831 2.529627
```

```r
apply(cp,2,var)*(n-1)/n
```

```
##    Dim.1    Dim.2    Dim.3
## 5.015650 3.084388 2.494493
```

```r
#Méthode Ward
hcp=hclust(dist(cp),method="ward.D")
plot(hcp, hang=-1,cex=0.7) # choix de 3  classes (k=3)
cl3=rect.hclust(hcp,k=3)
rect.hclust(hcp,k=3,border = "blue")


#contenu
cl3a=cutree(hcp,3)
cl3b=rect.hclust(hcp,k=3)
```




```r
class(cl3a) # pour les statistiques
```

```
## [1] "integer"
```

```r
class(cl3b) # pour les analyses avec les noms des tapenades
```

```
## [1] "list"
```

```r
#table des classes
class(cl3) # donc on va transformer de liste en facteur
```

```
## [1] "list"
```

```r
cl3=as.factor(cutree(hcp,4)) #cutree pour couper en nombre donnée
table(cl3a)
```

```
## cl3a
##  1  2  3
## 27 18 27
```

```r
#classification non hiérachique
centres=rbind(tapply(cp[,1],cl3a,mean),tapply(cp[,2],cl3a,mean),tapply(cp[,3]
,cl3a,mean))
km3=kmeans(cp,centers=centres)
print(km3$cluster)
```

```
##  P1T1  P2T1  P3T1  P4T1  P5T1  P6T1  P7T1  P8T1  P9T1 P10T1 P11T1 P12T1
P13T1
##     3     2     2     3     3     2     2     3     1     2     3     3
```

```
3
## P14T1 P15T1 P16T1 P17T1 P18T1 P19T1 P20T1 P21T1 P22T1 P23T1 P24T1  P1T2
P2T2
##     2     2     2     2     3     3     3     3     1     2     3     2
3
##  P3T2  P4T2  P5T2  P6T2  P7T2  P8T2  P9T2 P10T2 P11T2 P12T2 P13T2 P14T2
P15T2
##     1     1     1     1     1     2     3     3     1     1     1     1
1
## P16T2 P17T2 P18T2 P19T2 P20T2 P21T2 P22T2 P23T2 P24T2  P1T3  P2T3  P3T3
P4T3
##     3     1     3     3     2     3     2     1     3     3     3     3
3
##  P5T3  P6T3  P7T3  P8T3  P9T3 P10T3 P11T3 P12T3 P13T3 P14T3 P15T3 P16T3
P17T3
##     2     3     3     1     2     3     2     2     3     3     3     3
3
## P18T3 P19T3 P20T3 P21T3 P22T3 P23T3 P24T3
##     2     2     3     2     3     3     3
```

```r
table(km3$cluster)
```

```
##
##  1  2  3
## 15 21 36
```

```r
#effectifs
tc=function(X,Y)
{
    tc=table(X,Y); tc=cbind(tc,Total=apply(tc,1,sum));
    tc=rbind(tc,Total=apply(tc,2,sum));
    return(tc)
}

#fonction : les profils lignes
pl=function(X,Y)
{
    tc=table(X,Y); suml=apply(tc,1,sum);
    sumc=apply(tc,2,sum);
    pl=rbind(tc/suml,Ensemble=sumc/sum(sumc));
    return(pl)
}

#Description des classes par partition
table(cl3a,km3$cluster)
```

```
##
## cl3a  1  2  3
##    1 14  0 13
##    2  1 17  0
##    3  0  4 23
```

```r
tc(cl3a,km3$cluster)
```

```
##        1  2  3 Total
## 1     14  0 13    27
## 2      1 17  0    18
## 3      0  4 23    27
## Total 15 21 36    72
```

##choix de la meilleure partition

```r
 ##Varience interne
varinter=function(x, groupe)
{
  moyennes=tapply(x, groupe, mean)
  effectifs=tapply(x, groupe, length)
  res=(sum(effectifs*(moyennes - mean(x))^2))/sum(effectifs)
  return(res)
}

 ##Varience Totale, inter et intra
q=3 # nombre de composante principale
vt=0

#Fonction qui calcule la variance totale
vartot=function(x)
  {
res=mean((x-mean(x))^2)
return(res)
}

varp=function(x) return(var(x)*(length(x)-1)/length(x))

for (i in 1:q) vt=vt+varp(cp[,i])
print(paste("Inertie totale :",vt))
```

```
## [1] "Inertie totale : 10.5945317746886"
```

```r
vi=0
for (i in 1:q) vi=vi+varinter(cp[,i],cl3a)
print(paste("variance inter pour la partition hiérarchique :",vi))
```

```
## [1] "variance inter pour la partition hiérarchique : 4.46521939899937"
```

```r
print(paste("variance intra pour la partition hiérarchique :",vt-vi))
```

```
## [1] "variance intra pour la partition hiérarchique : 6.12931237568925"
```

```r
vi=0
for (i in 1:q) vi=vi+varinter(cp[,i],km3$cluster)
print(paste("variance inter pour la partition non hiérarchique :",vi))
```

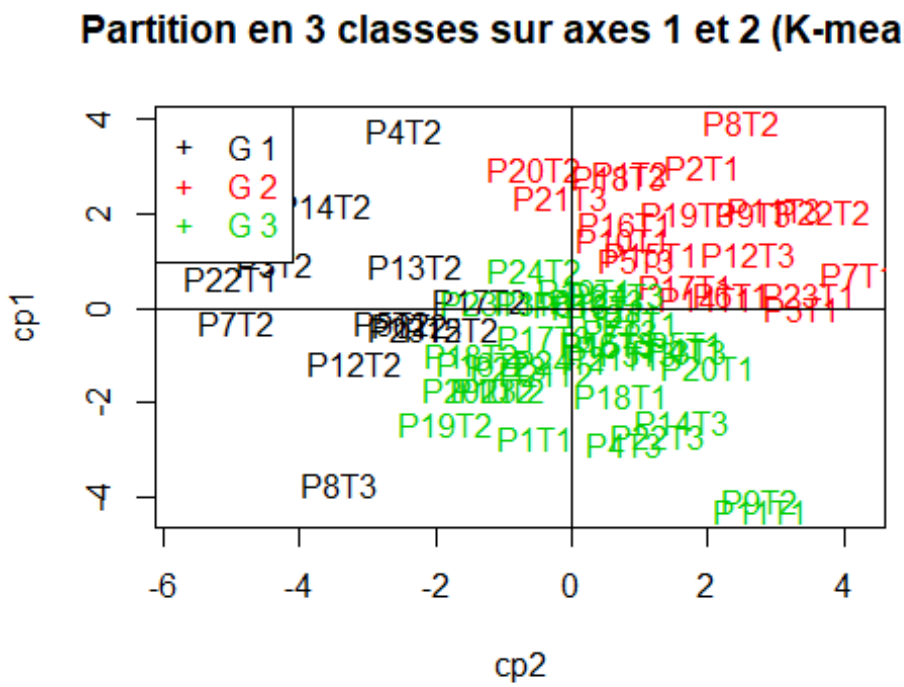
```
## [1] "variance inter pour la partition non hiérarchique : 4.99894035429444"
```

```
print(paste("variance intra pour la partition non hiérarchique :",vt-vi))
```

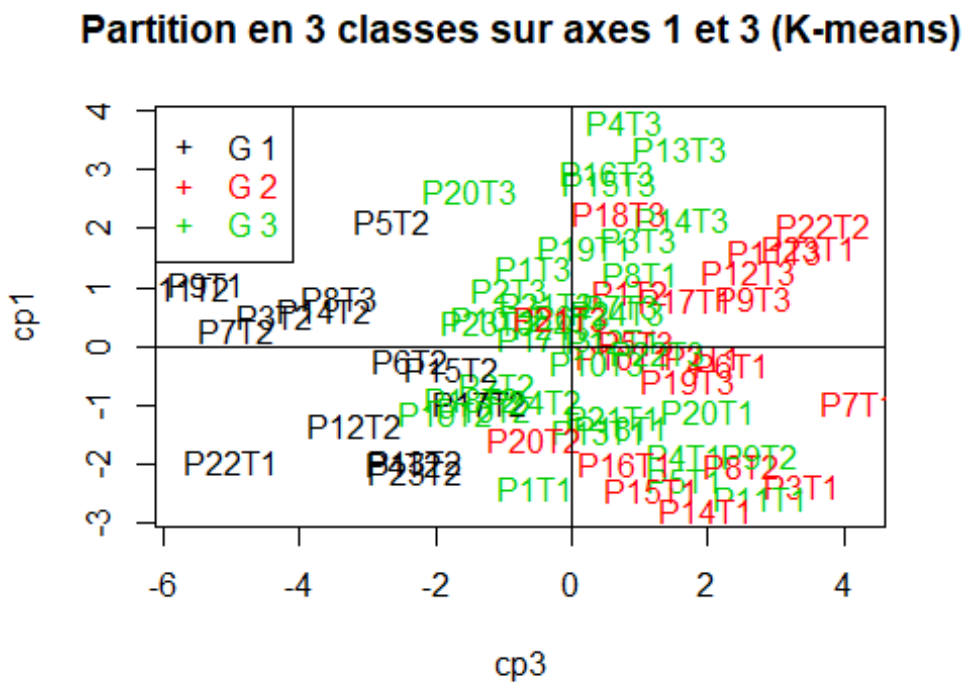
```
## [1] "variance intra pour la partition non hiérarchique : 5.59559142039417"
```

#Partition des centres mobiles , kms choisi

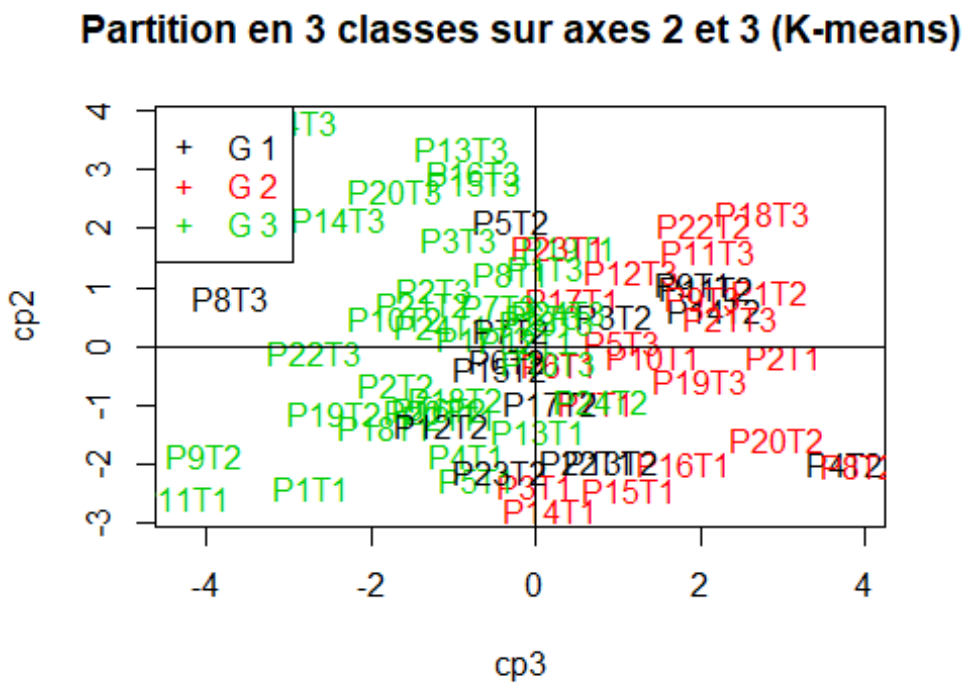
```
#Interpretation de la partition Kms
classes=as.vector(km3$cluster)
plot(cp[,1],cp[,2], main="Partition en 3 classes sur axes 1 et 2 (K-
means)",type="n", ylab="cp1",xlab="cp2")
text(cp[,1],cp[,2],dimnames(cp)[[1]],col=classes)
legend("topleft",paste("G",1 :3),pch="+", col=1:3,text.col=1:3)
abline(h=0,v=0)
```



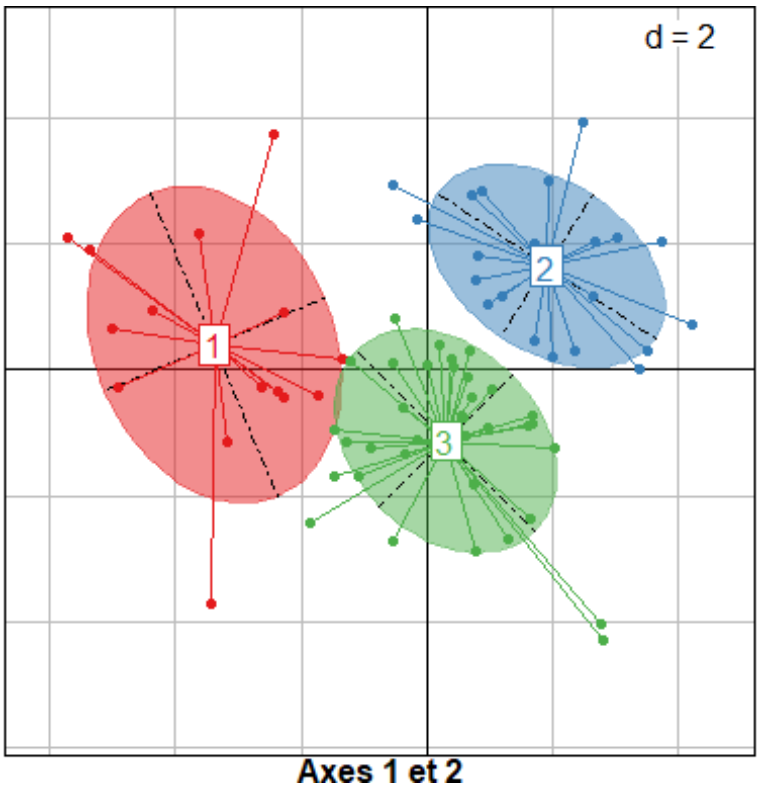

```
classes=as.vector(km3$cluster)
plot(cp[,1],cp[,3], main="Partition en 3 classes sur axes 1 et 3 (K-
means)",type="n", ylab="cp1",xlab="cp3")
text(cp[,1],cp[,3],dimnames(cp)[[1]],col=classes)
legend("topleft",paste("G",1:3),pch="+", col=1:3,text.col=1:3)
abline(h=0,v=0)
```



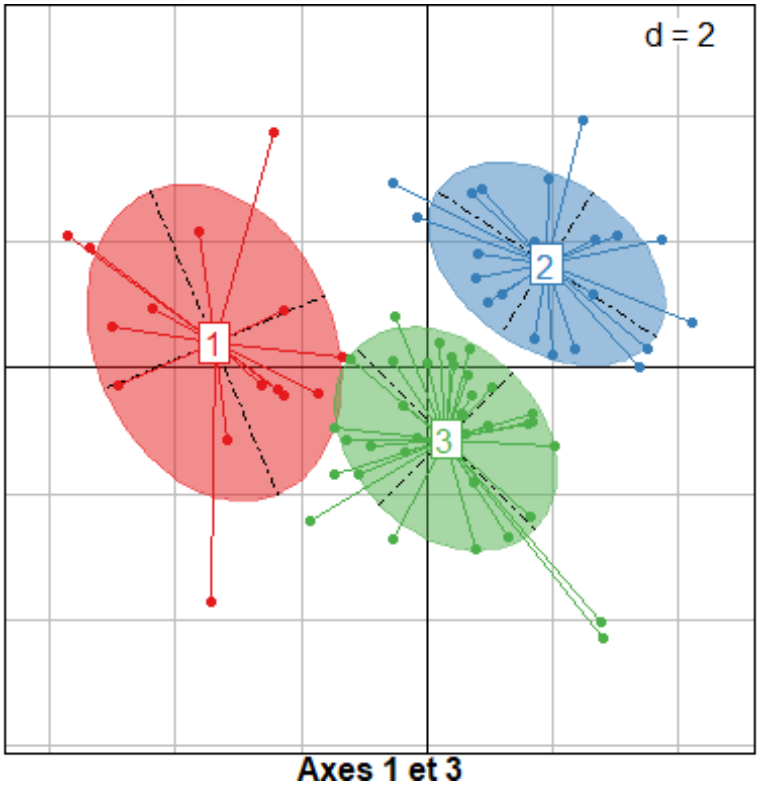

```
plot(cp[,2],cp[,3], main="Partition en 3 classes sur axes 2 et 3 (K-
means)",type="n", ylab="cp2",xlab="cp3")
text(cp[,2],cp[,3],dimnames(cp)[[1]],col=classes)
legend("topleft",paste("G",1:3),pch="+", col=1:3,text.col=1:3)
abline(h=0,v=0)
```



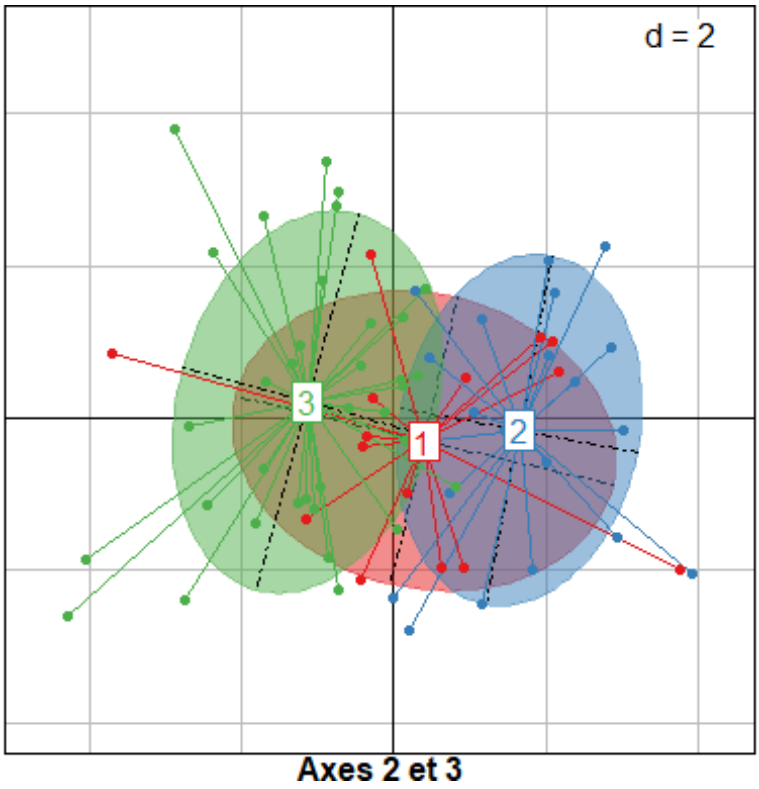

```r
#Modelisation des classes
s.class(cp[,1:2], as.factor(km3$cluster), col = brewer.pal(5, "Set1"), sub =
"Axes 1 et 2")
```



```r
s.class(cp[,1:3], as.factor(km3$cluster), col = brewer.pal(5, "Set1"), sub =
"Axes 1 et 3")
```




```r
s.class(cp[,2:3], as.factor(km3$cluster), col = brewer.pal(5, "Set1"), sub =
"Axes 2 et 3")
```




```
#Description
p=ncol(quanti)
k=3
tabmean=matrix(0,nrow=p,ncol=k+1)
dimnames(tabmean)=list(colnames(quanti),c(paste("Cl",1:k),"Total"))
tabmean[,k+1]=sapply(quanti,mean)
for (j in (1:p)) tabmean[j,1:k]=tapply(quanti[,j],km3$cluster,mean)
round(tabmean,digits=3)
```

```
##                Cl 1   Cl 2   Cl 3  Total
## Vintensite   -0.233  0.194 -0.050 -0.017
## Vcompact      0.311 -0.063 -0.093  0.000
## Vtexture      1.178 -0.381 -0.278 -0.005
## Vtons         0.889 -0.540 -0.065 -0.005
## Vbrillance    0.444  0.159 -0.278  0.000
## Vattirance   -0.600  0.444 -0.009  0.000
## Ofruit       -0.689  0.190  0.176  0.000
## Oamer         0.511 -0.286 -0.046  0.000
## Oepice        0.178 -0.333  0.120  0.000
## Ovegetal      0.489  0.382 -0.435 -0.004
## Gacide       -0.688 -0.001  0.304  0.008
## Gdurete       0.941 -0.169 -0.250  0.022
## Gintense     -0.344 -0.285  0.307 -0.001
## Gvegetal      0.289  0.186 -0.222  0.003
## Giode        -0.200 -0.029  0.102  0.001
## Gsale        -0.311  0.453 -0.148 -0.007
## Gfruit       -0.578  0.024  0.229  0.001
```

```
## Gepice         0.156 -0.333  0.130  0.000
## Gsucre        -0.289 -0.095  0.176  0.000
## Gamer         -1.089  0.127  0.380  0.000
## Fagreable     -0.556  1.002 -0.355 -0.001
## Fintense      -1.044  0.480  0.153 -0.001
## Fpersistance -0.911  0.716 -0.033  0.003
## Fnote         -0.556  1.238 -0.491  0.000
## Fharmonie     -0.822  1.111 -0.306  0.000
## Farome        -0.933  0.857 -0.111  0.000
```

```
kable(round(tabmean,digits=3))
```

	Cl 1	Cl 2	Cl 3	Total
Vintensite	-0.233	0.194	-0.050	-0.017
Vcompact	0.311	-0.063	-0.093	0.000
Vtexture	1.178	-0.381	-0.278	-0.005
Vtons	0.889	-0.540	-0.065	-0.005
Vbrillance	0.444	0.159	-0.278	0.000
Vattirance	-0.600	0.444	-0.009	0.000
Ofruit	-0.689	0.190	0.176	0.000
Oamer	0.511	-0.286	-0.046	0.000
Oepice	0.178	-0.333	0.120	0.000
Ovegetal	0.489	0.382	-0.435	-0.004
Gacide	-0.688	-0.001	0.304	0.008
Gdurete	0.941	-0.169	-0.250	0.022
Gintense	-0.344	-0.285	0.307	-0.001
Gvegetal	0.289	0.186	-0.222	0.003
Giode	-0.200	-0.029	0.102	0.001
Gsale	-0.311	0.453	-0.148	-0.007
Gfruit	-0.578	0.024	0.229	0.001
Gepice	0.156	-0.333	0.130	0.000
Gsucre	-0.289	-0.095	0.176	0.000
Gamer	-1.089	0.127	0.380	0.000
Fagreable	-0.556	1.002	-0.355	-0.001
Fintense	-1.044	0.480	0.153	-0.001
Fpersistance	-0.911	0.716	-0.033	0.003
Fnote	-0.556	1.238	-0.491	0.000
Fharmonie	-0.822	1.111	-0.306	0.000
Farome	-0.933	0.857	-0.111	0.000

```r
#table de contingence kms$cluster * tab$tapenade
kable(tc(km3$cluster,tab$Tapenade))
```

	T1	T2	T3	Total
1	2	12	1	15
2	10	4	7	21
3	12	8	16	36
Total	24	24	24	72

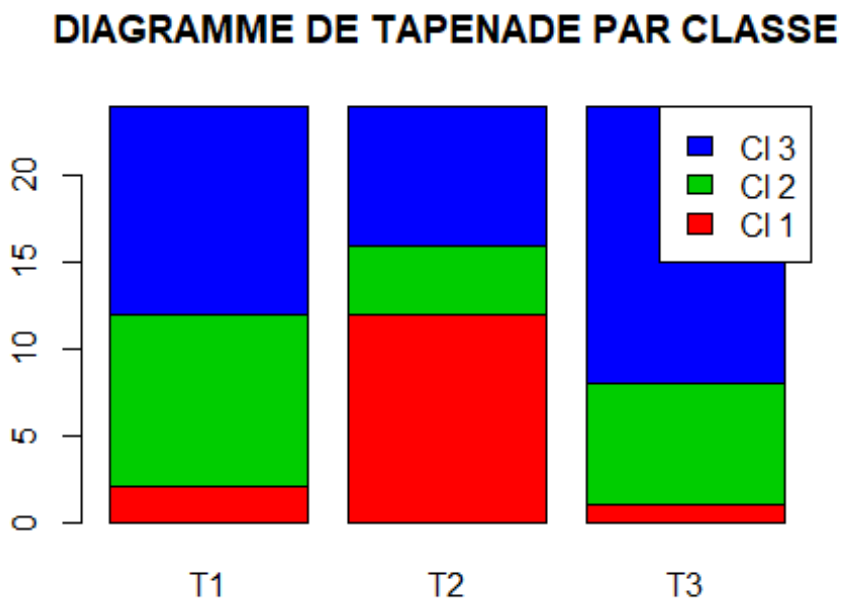
```r
kable(pl(km3$cluster,tab$Tapenade))
```

	T1	T2	T3
1	0.1333333	0.8000000	0.0666667
2	0.4761905	0.1904762	0.3333333
3	0.3333333	0.2222222	0.4444444
Ensemble	0.3333333	0.3333333	0.3333333

```r
#CrossTable(km3$cluster,tab$Tapenade,prop.t=T,prop.chisq=T)


#Test du khi-deux  pour voir si la couleur est liée à la tapenade.
#Test d'indépendance
#Ho: la couleur et la tapenade sont indépendantes
t1=chisq.test(table(km3$cluster,tab$Tapenade))


tableIndep =data.frame("Croissements"=c("Classe * Tapenade"),"Khi-
deux"=c(t1$statistic),"ddl"=c(t1$parameter),"Pvalue"
=c(t1$p.value),"Significativité à 5%"=c("significative à 5%") ,
"Conclusion"=c("sont liées") ); kable(tableIndep)
```

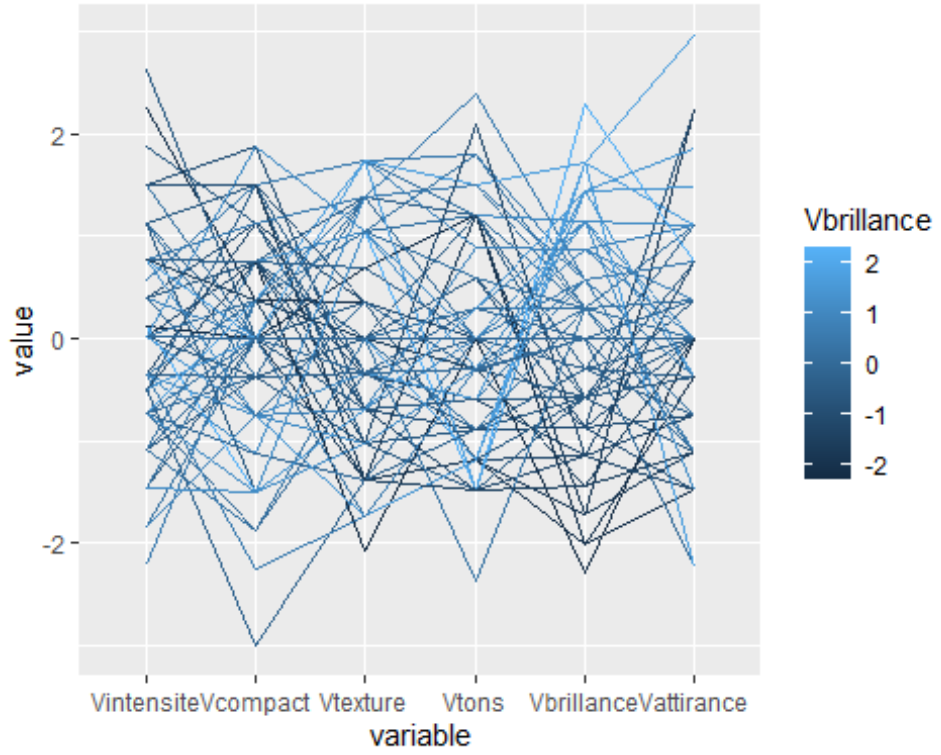
	Croissements	Khi.deux	ddl	Pvalue	Significativité.à.5.	Conclusion
X-squared	Classe * Tapenade	20.0381	4	0.0004908	significative à 5%	sont liées

```r
#Graphique
barplot(table(km3$cluster,tab$Tapenade),main ="DIAGRAMME DE TAPENADE PAR
CLASSE",legend.text
=paste("Cl",rownames(table(km3$cluster,tab$Tapenade))),col
=2:5,args.legend=list(x="topright"))
```



#Analyse discriminante

```
#Parallele
#palette=brewer.pal(3, "Set1")
#colors=palette[as.numeric(quantii$Tapenade)]
#parcoord(quantii[,1:26] , col=colors)
ggparcoord(quantii[,1:26],columns = 1:6, groupColumn = 5)
```




```
#analyse discriminante
ad=lda(x=quantii[,1:26],grouping=quantii$Tapenade,CV=F) #Prend en compte les
CP

# ebouli des valeurs propres (distribution del'information selon les axes)
ad$svd/sum(ad$svd)

## [1] 0.6127487 0.3872513

# proportion de trace
ad$svd^2/sum(ad$svd^2)

## [1] 0.7145859 0.2854141

#prediction
pred.lda=predict(ad, newdata = quantii[,1:26])
ad1=lda(x=quantii[,1:26],grouping=quantii$Tapenade,CV=T) # prend en compte
les observations


#Gestion des couleurs
color=as.vector(quantii$Tapenade)
color[color=="T1"]="black"
color[color=="T2"]="red"
color[color=="T3"]="green"

# X11()
```

```r
# #fonction de  representation des individus
# lda_visual=function(data.frame,categories,col=1) {
#   require("MASS")
#  mylda=lda(x=data.frame,data=data.frame,grouping=categories)
#  D1= mylda$scaling[,1] # vecteur 1
#  D2 =mylda$scaling[,2] # vecteur 2
#  # Calcul de projeté des individus sur D1 et D2
#  xy=as.matrix(data.frame)%*%as.matrix(cbind(D1,D2))# produit scalaire
#  plot(xy[,1],xy[,2],col=col,xlab="LD1(71,4 %)",ylab="LD2(28,5 %)",type="n")
#  legend("topleft",paste("T",1:3),pch="+", col=1:3,text.col=1:3,cex = 0.6)
#  text(xy[,1],xy[,2],labels=rownames(quantii),col=col,cex =1)
#  abline(h=0,v=0)
#  }
#
# lda_visual(quantii[,1:26],quantii$Tapenade,col=color)


#-------------------------------------------------------------------------
#Analyse discriminante AFD

X=quantii[,1:26];y=quantii$Tapenade
source("AFD_procedures.R")
afd=AFD(X,y,type="FR")
head(afd)
```

```
## $eig
## [1] 0.8008991 0.6163681
##
## $U
##                         u1          u2
## Vintensite   -0.244865340  0.12039681
## Vcompact     -0.522009335 -0.14767596
## Vtexture      0.182429643 -0.06884955
## Vtons        -0.350389861  0.12610722
## Vbrillance   -0.251779818  0.30788609
## Vattirance    0.255448329 -0.38200286
## Ofruit       -0.001387271 -0.51503806
## Oamer        -0.638976636 -0.17146433
## Oepice       -0.028434129 -0.25304881
## Ovegetal      0.213711962  0.17876326
## Gacide        0.305734308  0.14409297
## Gdurete      -0.142912471  0.05263729
## Gintense      0.398701651 -0.37427230
## Gvegetal     -0.084228293  0.33169271
## Giode        -0.336487499  0.08374530
## Gsale         0.123946594  0.07928728
## Gfruit       -0.079694253  0.36026917
## Gepice        0.201409584  0.03395500
## Gsucre       -0.180068106  0.13469408
## Gamer        -0.106233019  0.20269301
```

```
## Fagreable     0.213613330  0.40390285
## Fintense      0.300695021  0.27635940
## Fpersistance  0.045965915 -0.24226214
## Fnote        -0.262099085 -0.32804057
## Fharmonie    -0.090179574 -0.32133403
## Farome       -0.249757648 -0.07337184
##
## $S
##                  s1          s2
## P1T1  -0.239341581 -1.10413401
## P2T1  -0.714876442 -1.06447443
## P3T1  -0.124423545 -2.30760516
## P4T1   0.219358819 -1.65819039
## P5T1   0.454880663 -1.97665315
## P6T1  -0.003549864 -0.93309878
## P7T1   0.303259071 -1.62285159
## P8T1   0.824299427 -0.14951721
## P9T1   0.023272024 -0.70257889
## P10T1 -0.417440607 -0.02438880
## P11T1 -0.722287984 -0.38704012
## P12T1  0.192656253 -0.80135756
## P13T1 -0.286895176 -1.55647866
## P14T1 -0.227330316 -2.25455879
## P15T1 -0.112508513 -1.20490209
## P16T1 -0.276736182 -0.68249677
## P17T1  0.355781944 -0.50527529
## P18T1  0.447967623 -1.81571246
## P19T1  0.363898342 -0.33077949
## P20T1  0.579411023 -1.23590531
## P21T1 -0.580631135 -1.12411299
## P22T1 -0.273773112 -0.58619368
## P23T1  0.369347297 -0.51948433
## P24T1 -0.851565985 -2.09206581
## P1T2  -0.733946757  0.54340868
## P2T2  -0.482209941  0.27207028
## P3T2  -1.358193763  1.36545030
## P4T2  -1.363617532  0.06362570
## P5T2  -1.758048662  0.66783974
## P6T2  -0.987196171  0.80889155
## P7T2  -1.177728279  1.88529913
## P8T2  -1.150878029 -0.06976971
## P9T2  -0.992824369 -0.16492213
## P10T2 -1.498081456  0.38706227
## P11T2 -0.560141090 -0.18281747
## P12T2 -0.996735204  0.69859455
## P13T2 -1.127855854  1.10715444
## P14T2 -1.552106401  1.61521681
## P15T2 -1.635453022  0.97386402
## P16T2 -1.262114452  0.60186950
## P17T2 -0.646861480  0.28530697
```

```
## P18T2 -1.595322390  0.27272112
## P19T2 -0.303045654  0.58039515
## P20T2 -1.752022118  0.12168501
## P21T2 -0.677136132  0.50159879
## P22T2 -0.423460307 -0.26698242
## P23T2 -1.216900010 -0.35013476
## P24T2 -0.698027862  2.13220843
## P1T3   0.632427587  0.70043835
## P2T3   1.286044432  0.74888686
## P3T3   1.453787151  0.93201751
## P4T3   1.115428555  1.58442734
## P5T3   1.259051126  1.29147142
## P6T3   0.946629162  0.10686523
## P7T3   0.871116910 -0.26057714
## P8T3   0.396829006  0.25584637
## P9T3   1.039802750  0.90406047
## P10T3  1.985772468 -0.32611402
## P11T3  1.352679479  0.60641704
## P12T3  0.874329356  0.13932246
## P13T3  1.478491516  0.33845015
## P14T3  1.776084418  0.64121238
## P15T3  1.744609236  0.23290847
## P16T3  1.467358289  0.02550069
## P17T3  0.246962663  0.20262632
## P18T3  1.144002469  1.54486175
## P19T3  0.057092253 -0.28899799
## P20T3  1.169258796  1.11609070
## P21T3  1.257693915  0.60277276
## P22T3  0.693881120  0.85504649
## P23T3  0.844200415  0.87148948
## P24T3  1.553601818 -0.03480331
##
## $mat.corr
##                         s1           s2
## Vintensite   -0.285330856 -0.339694511
## Vcompact     -0.438361493 -0.357413287
## Vtexture     -0.631243110  0.340614424
## Vtons        -0.378414533  0.208806003
## Vbrillance    0.009194252  0.456468844
## Vattirance    0.337893085 -0.149063320
## Ofruit        0.196684084 -0.618264120
## Oamer        -0.489052637  0.037596278
## Oepice        0.055021450 -0.168004943
## Ovegetal      0.005698402  0.282909675
## Gacide        0.221338116  0.099963291
## Gdurete      -0.591605561  0.071964020
## Gintense      0.216121892 -0.268308690
## Gvegetal     -0.104524743  0.354047574
## Giode         0.096634281  0.277060790
## Gsale         0.412101147 -0.080370198
```

```
## Gfruit        0.038149750 -0.067554362
## Gepice        0.189652383  0.019946451
## Gsucre        0.206260447 -0.005398323
## Gamer         0.269216036  0.098477364
## Fagreable     0.138563396 -0.193096732
## Fintense      0.153026339 -0.198282095
## Fpersistance  0.195747920 -0.200806314
## Fnote        -0.075133270 -0.314939116
## Fharmonie    -0.044346631 -0.364204883
## Farome        0.088951412 -0.370340172
##
## $nk
## T1 T2 T3
## 24 24 24
##
## $K
## [1] 3
```

```
afd$U #matrice des facteurs discriminants
```

```
##                        u1          u2
## Vintensite   -0.244865340  0.12039681
## Vcompact     -0.522009335 -0.14767596
## Vtexture      0.182429643 -0.06884955
## Vtons        -0.350389861  0.12610722
## Vbrillance   -0.251779818  0.30788609
## Vattirance    0.255448329 -0.38200286
## Ofruit       -0.001387271 -0.51503806
## Oamer        -0.638976636 -0.17146433
## Oepice       -0.028434129 -0.25304881
## Ovegetal      0.213711962  0.17876326
## Gacide        0.305734308  0.14409297
## Gdurete      -0.142912471  0.05263729
## Gintense      0.398701651 -0.37427230
## Gvegetal     -0.084228293  0.33169271
## Giode        -0.336487499  0.08374530
## Gsale         0.123946594  0.07928728
## Gfruit       -0.079694253  0.36026917
## Gepice        0.201409584  0.03395500
## Gsucre       -0.180068106  0.13469408
## Gamer        -0.106233019  0.20269301
## Fagreable     0.213613330  0.40390285
## Fintense      0.300695021  0.27635940
## Fpersistance  0.045965915 -0.24226214
## Fnote        -0.262099085 -0.32804057
## Fharmonie    -0.090179574 -0.32133403
## Farome       -0.249757648 -0.07337184
```

```
afd$S #matrices des variables discriminantes
```

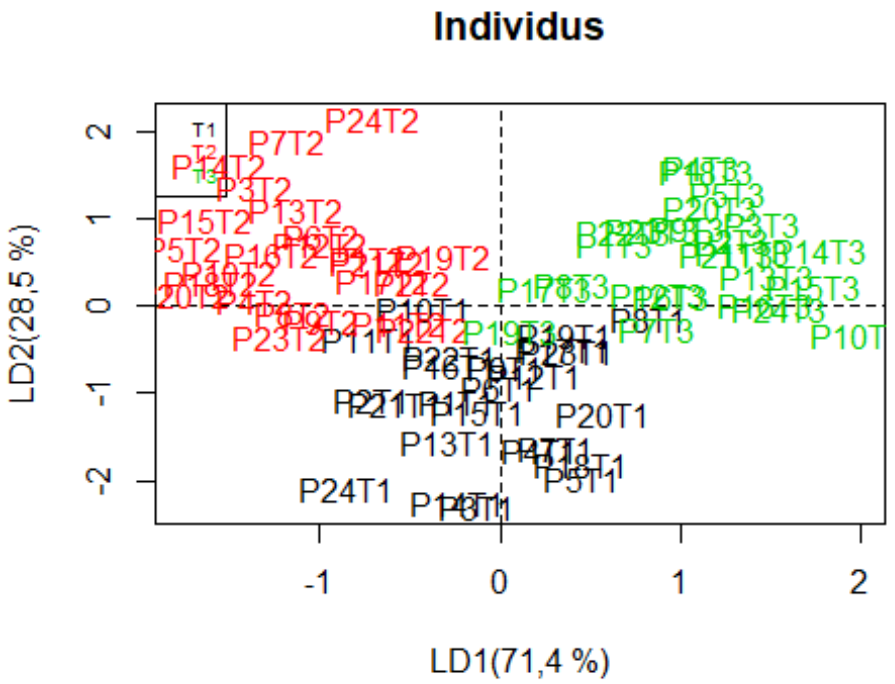
```
##                 s1          s2
## P1T1  -0.239341581 -1.10413401
## P2T1  -0.714876442 -1.06447443
## P3T1  -0.124423545 -2.30760516
## P4T1   0.219358819 -1.65819039
## P5T1   0.454880663 -1.97665315
## P6T1  -0.003549864 -0.93309878
## P7T1   0.303259071 -1.62285159
## P8T1   0.824299427 -0.14951721
## P9T1   0.023272024 -0.70257889
## P10T1 -0.417440607 -0.02438880
## P11T1 -0.722287984 -0.38704012
## P12T1  0.192656253 -0.80135756
## P13T1 -0.286895176 -1.55647866
## P14T1 -0.227330316 -2.25455879
## P15T1 -0.112508513 -1.20490209
## P16T1 -0.276736182 -0.68249677
## P17T1  0.355781944 -0.50527529
## P18T1  0.447967623 -1.81571246
## P19T1  0.363898342 -0.33077949
## P20T1  0.579411023 -1.23590531
## P21T1 -0.580631135 -1.12411299
## P22T1 -0.273773112 -0.58619368
## P23T1  0.369347297 -0.51948433
## P24T1 -0.851565985 -2.09206581
## P1T2  -0.733946757  0.54340868
## P2T2  -0.482209941  0.27207028
## P3T2  -1.358193763  1.36545030
## P4T2  -1.363617532  0.06362570
## P5T2  -1.758048662  0.66783974
## P6T2  -0.987196171  0.80889155
## P7T2  -1.177728279  1.88529913
## P8T2  -1.150878029 -0.06976971
## P9T2  -0.992824369 -0.16492213
## P10T2 -1.498081456  0.38706227
## P11T2 -0.560141090 -0.18281747
## P12T2 -0.996735204  0.69859455
## P13T2 -1.127855854  1.10715444
## P14T2 -1.552106401  1.61521681
## P15T2 -1.635453022  0.97386402
## P16T2 -1.262114452  0.60186950
## P17T2 -0.646861480  0.28530697
## P18T2 -1.595322390  0.27272112
## P19T2 -0.303045654  0.58039515
## P20T2 -1.752022118  0.12168501
## P21T2 -0.677136132  0.50159879
## P22T2 -0.423460307 -0.26698242
## P23T2 -1.216900010 -0.35013476
## P24T2 -0.698027862  2.13220843
## P1T3   0.632427587  0.70043835
```

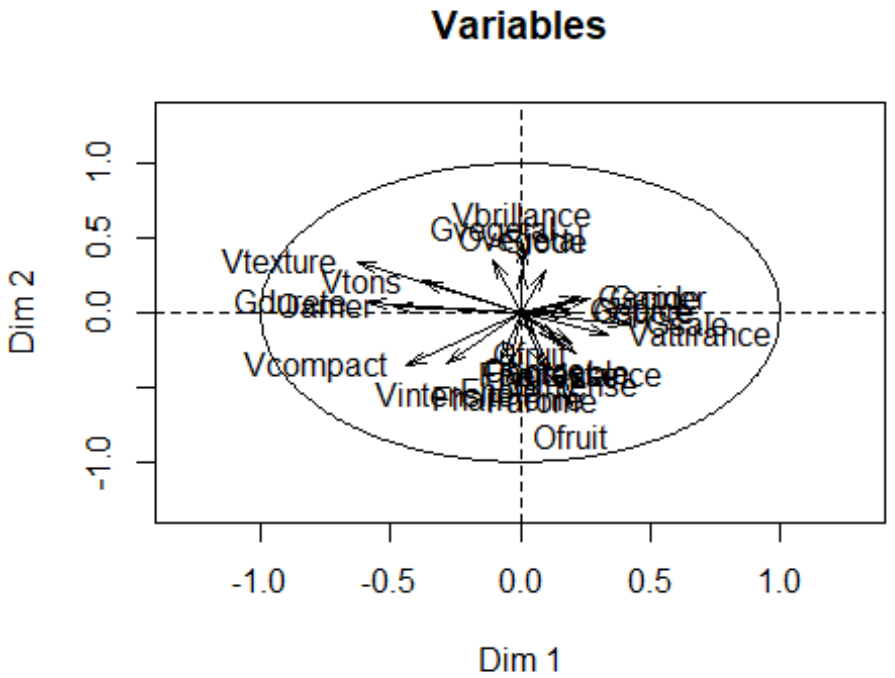
```
## P2T3   1.286044432  0.74888686
## P3T3   1.453787151  0.93201751
## P4T3   1.115428555  1.58442734
## P5T3   1.259051126  1.29147142
## P6T3   0.946629162  0.10686523
## P7T3   0.871116910 -0.26057714
## P8T3   0.396829006  0.25584637
## P9T3   1.039802750  0.90406047
## P10T3  1.985772468 -0.32611402
## P11T3  1.352679479  0.60641704
## P12T3  0.874329356  0.13932246
## P13T3  1.478491516  0.33845015
## P14T3  1.776084418  0.64121238
## P15T3  1.744609236  0.23290847
## P16T3  1.467358289  0.02550069
## P17T3  0.246962663  0.20262632
## P18T3  1.144002469  1.54486175
## P19T3  0.057092253 -0.28899799
## P20T3  1.169258796  1.11609070
## P21T3  1.257693915  0.60277276
## P22T3  0.693881120  0.85504649
## P23T3  0.844200415  0.87148948
## P24T3  1.553601818 -0.03480331
```

```
afd$mat.corr #matrices des correlations entre les variables initiales et les
variables discriminantes
```

```
##                       s1           s2
## Vintensite   -0.285330856 -0.339694511
## Vcompact     -0.438361493 -0.357413287
## Vtexture     -0.631243110  0.340614424
## Vtons        -0.378414533  0.208806003
## Vbrillance    0.009194252  0.456468844
## Vattirance    0.337893085 -0.149063320
## Ofruit        0.196684084 -0.618264120
## Oamer        -0.489052637  0.037596278
## Oepice        0.055021450 -0.168004943
## Ovegetal      0.005698402  0.282909675
## Gacide        0.221338116  0.099963291
## Gdurete      -0.591605561  0.071964020
## Gintense      0.216121892 -0.268308690
## Gvegetal     -0.104524743  0.354047574
## Giode         0.096634281  0.277060790
## Gsale         0.412101147 -0.080370198
## Gfruit        0.038149750 -0.067554362
## Gepice        0.189652383  0.019946451
## Gsucre        0.206260447 -0.005398323
## Gamer         0.269216036  0.098477364
## Fagreable     0.138563396 -0.193096732
## Fintense      0.153026339 -0.198282095
```

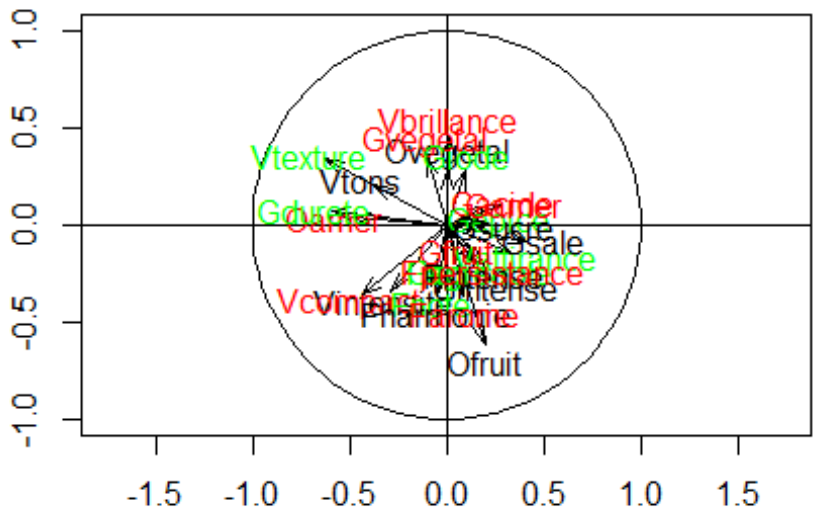
```
## Fpersistance  0.195747920 -0.200806314
## Fnote        -0.075133270 -0.314939116
## Fharmonie    -0.044346631 -0.364204883
## Farome        0.088951412 -0.370340172

plotAFD(afd) #Cercle de corrélation des variables et plot des individus
```





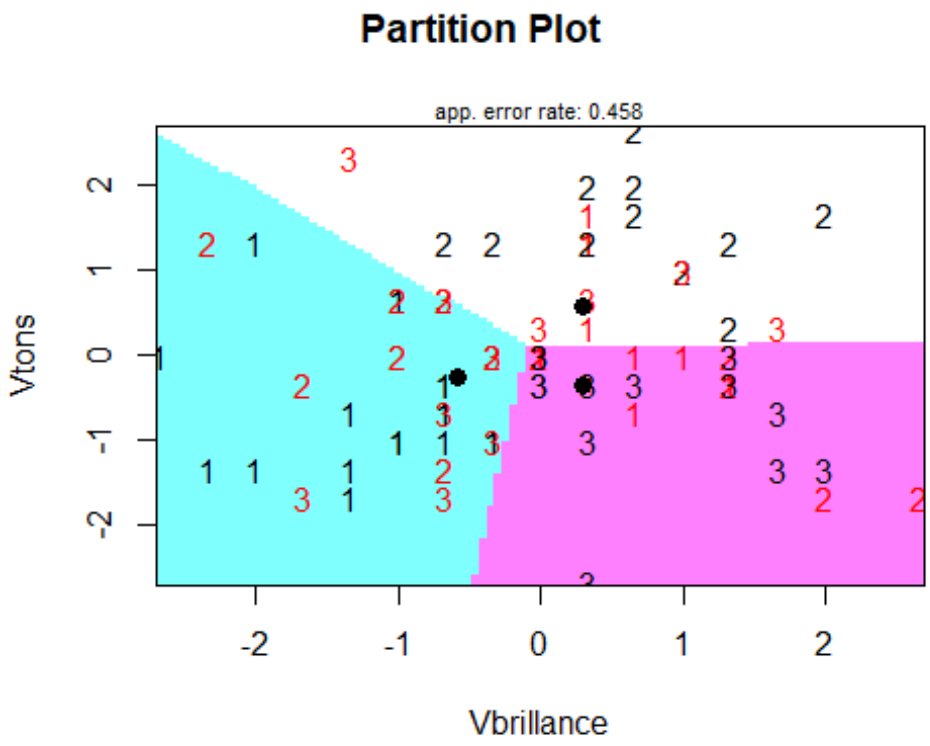

```r
#Cercle de corrélation des variables personnalisé
plot(c(-1,1),c(-1,1),type="n",xlab="",ylab="",asp=1,axes=T)
abline(h=0,v=0)
# Tracer le cercle
```

```r
draw.circle(0,0,1,lwd=1)
max(afd$mat.corr[,1]);max(afd$mat.corr[,2])

## [1] 0.4121011

## [1] 0.4564688

# Tracer les flèches
arrows(0,0,afd$mat.corr[,1],afd$mat.corr[,2],length=0.1,angle=17,lwd=1)
# Calculer la position du nom de chaque vecteur // aux pointes des flèches
x_bin=afd$mat.corr[,1] ; x_bin[x_bin<0]= -1 ;  x_bin[x_bin>0]=1
y_bin=afd$mat.corr[,2] ; y_bin[y_bin<0]= -1 ;  y_bin[y_bin>0]=1

x=afd$mat.corr[,1]^2 ; y=afd$mat.corr[,2]^2; z=x + y

x=x / z * x_bin *0.09  ; y=y /z * y_bin *0.09

# Tracer les noms des vecteurs
text(afd$mat.corr[,1]+x,afd$mat.corr[,2]+y,rownames(afd$mat.corr),cex=1,col=c
("black","red","green"))
```

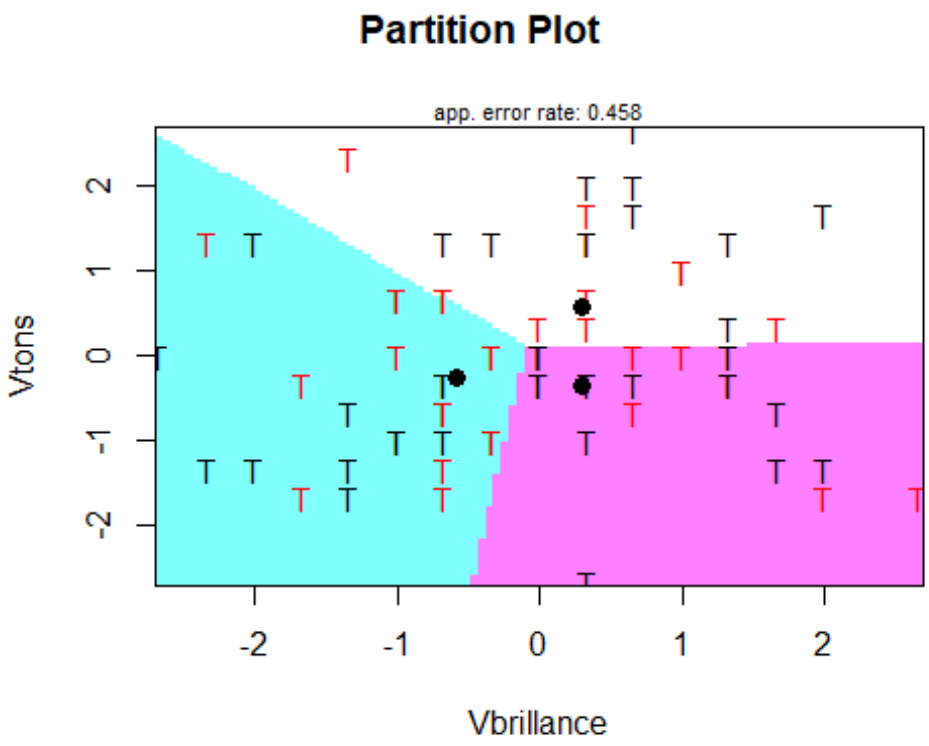



```r
#Interpretation des  caractéristiques----------------------------------------
---------

#autre image d'interpretation
quantii$Name=NA;quantii$Name=substr(quantii$Tapenade,2,2)
```

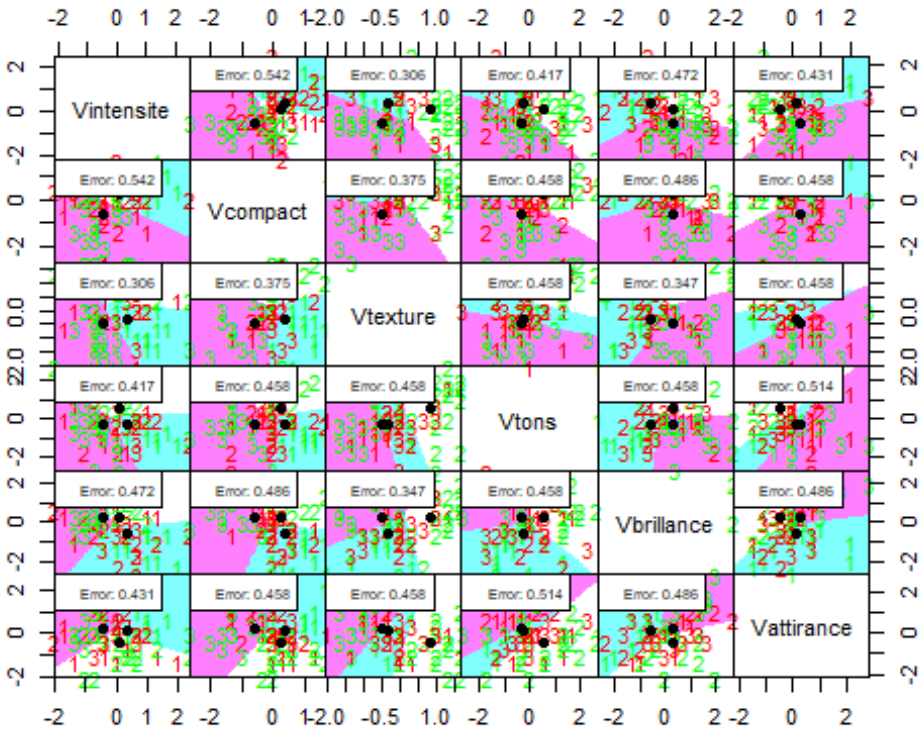
```
partimat(x=quantii[,4:5],grouping=as.factor(quantii$Name),method="lda", main
= "Partition Plot")
```



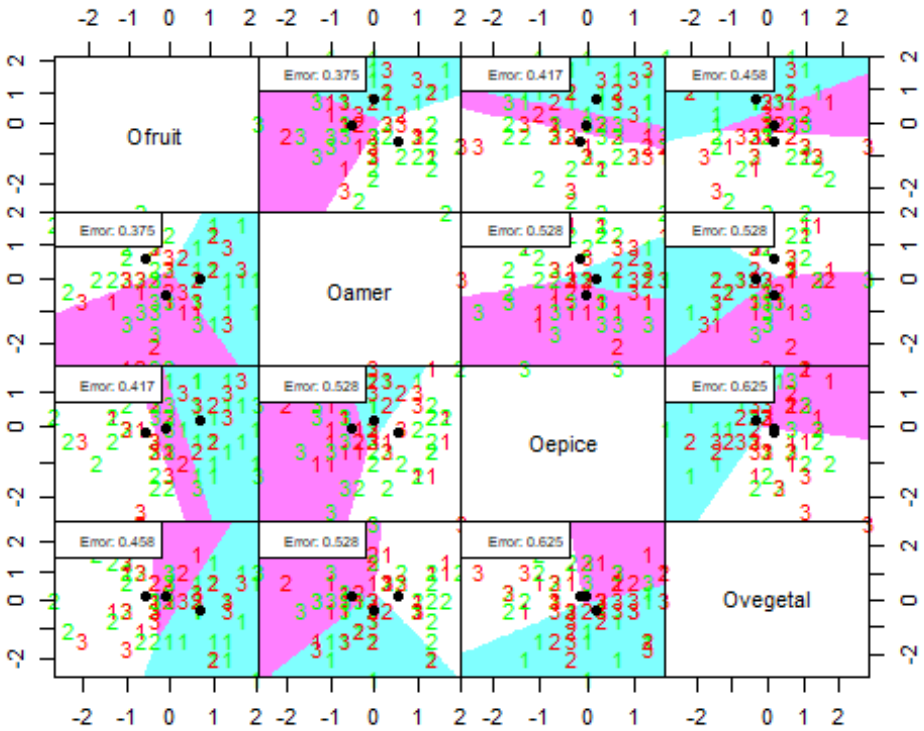

```
partimat(x=quantii[,4:5],grouping=quantii$Tapenade,method="lda", main =
"Partition Plot",prec=100)
```



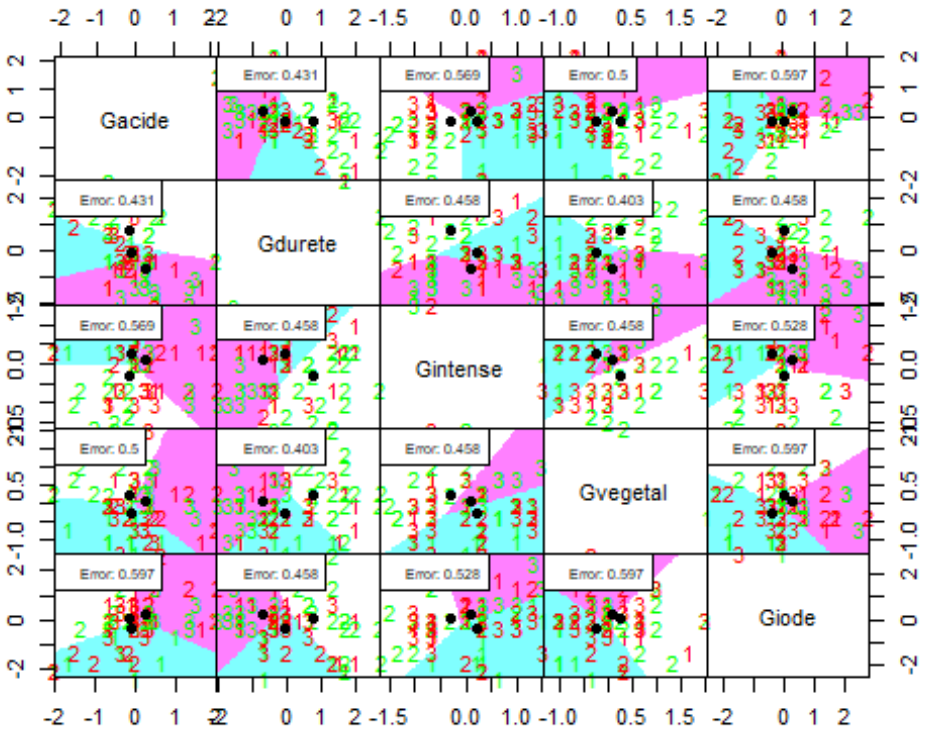

```r
#Caractéristiques Visuelles
partimat(as.factor(quantii$Name)~.,data=quantii[,1:6],method="lda",plot.matri
x = TRUE, imageplot = T, col.correct='green', col.wrong='red', cex=1,main =
"caractéristiques visuelles")
```



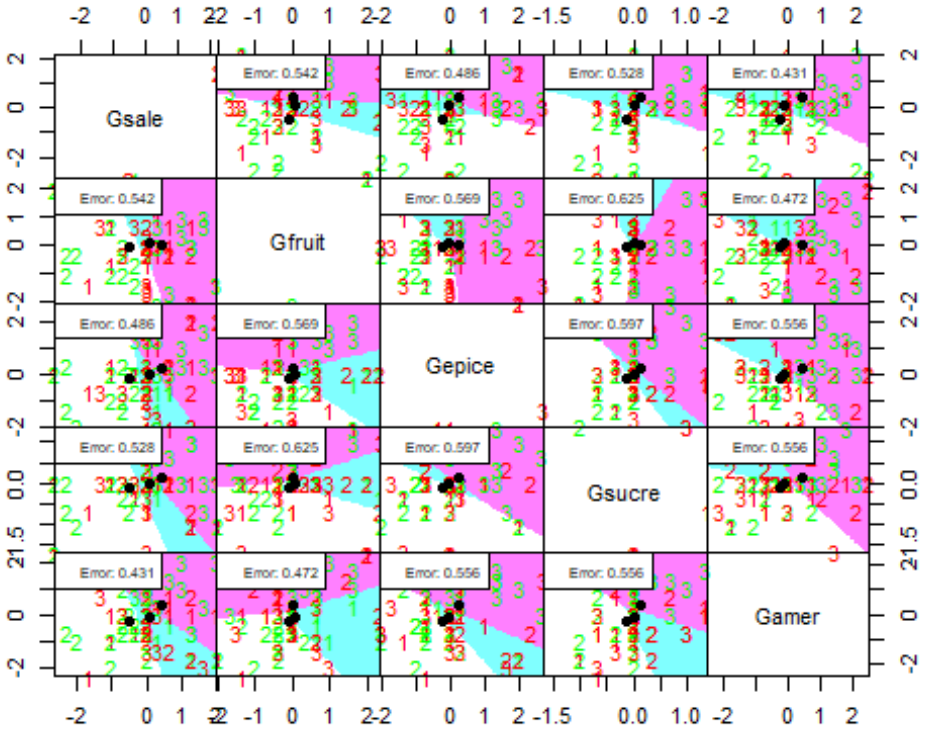

```
#Caractéristiques Olfactives
partimat(as.factor(quantii$Name)~.,data=quantii[,7:10],method="lda",plot.matrix = TRUE, imageplot = T, col.correct='green', col.wrong='red', cex=1,main = "caractéristiques olfactives")
```




```
#Caractéristiques Gustatvies

partimat(as.factor(quantii$Name)~.,data=quantii[,11:15],method="lda",plot.mat
rix = TRUE, imageplot = T, col.correct='green', col.wrong='red', cex=1,main
="caractéristiques gustatives 1")
```




```
partimat(as.factor(quantii$Name)~.,data=quantii[,16:20],method="lda",plot.matrix = TRUE, imageplot = T, col.correct='green', col.wrong='red', cex=1,main="caractéristiques gustatives 2")
```



```r
#Caractéristiques en bouche
partimat(as.factor(quantii$Name)~.,data=quantii[,21:26],method="lda",plot.matrix = TRUE, imageplot = T, col.correct='green', col.wrong='red', cex=1,main="finalités en bouche")
```