OUTILS INFORMATIQUES DE LA STATISTIQUE

Conservatoire National des Arts & Métiers, Paris.

le cnam

RESUME

Réalisation du projet « Huiles d'olives 2 » dans le cadre de l'unité d'enseignement STA115 sous la direction de Monsieur Giorgio RUSSOLILLO, Maître de conférences au CNAM.

Guillaume CHEVRON

chevron.guillaume@gmail.com

+33 6 76 77 34 26

Le 5 Avril 2021

Table des matières

I. Objectif de l'étude

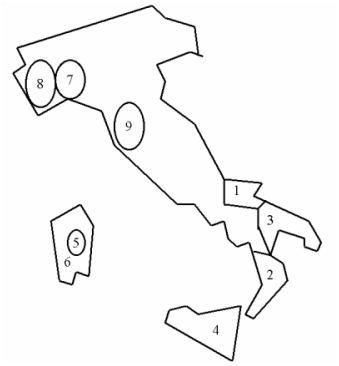
L'objet de cette analyse est l'étude de la composition en acide gras de plusieurs huiles d'olive italiennes et sa mise en relation avec leur région d'origine. En d'autres termes, est-il possible de classer ces huiles à partir de leurs acidités respectives et du lieu où elles ont été produites ?

Pour ce faire, nous disposons de 572 échantillons d'huiles d'olives caractérisées par dix variables, identifiées en amont de l'étude. Ces variables sont les suivantes :

a) Le secteur géographique (*macro_area*) est une variable catégorielle composée de trois niveaux
b) La région d'origine (*region*) est une variable catégorielle composée de neuf niveaux
c) Les mesures d'acidité pour chacun des huit acides retenus représentent huit variables continues

En agrégeant l'ensemble des échantillons recueillis par région d'origine, on obtient le jeu de données suivant :

Group	Macro Area	Region	Samples
1	South	Apulia-North	25
2	South	Calabria	56
3	South	Apulia-South	206
4	South	Sicily	36
5	Sardinia	Sardinia-Inland	65
6	Sardinia	Sandinia-Coast	33
7	Centre-North	Liguria-East	50
8	Centre-North	Liguria-West	50
9	Centre-North	Umbria	51
Total			572



II. Importation et exportation des données avec R

Dans cette première partie, on cherchera d'abord à extraire puis explorer le jeu de données. Cette étape est cruciale puisqu'elle permet de construire un fichier .txt qui servira comme base de travail au prétraitement des données dans SAS. Le jeu de données initial est téléchargeable directement depuis le package R '*pdfCluster*'. Une fois téléchargé, on peut le visualiser sous la forme d'un data.frame que l'on nommera *oliveoil*.

```
### Part 1: Importing & exporting the data
## 1) Installing standard packages for statistical analysis
# a) Downloading the required packages
packages.installed = c(
  'pdfCluster',
  'FactoMineR',
  'ggplot2',
  'ggpubr',
  'MLmetrics')

if (length(setdiff(packages.installed,
                   rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages.installed,
                           rownames(installed.packages())))}
library(pdfCluster)
library(FactoMineR)
library(ggplot2)
library(ggpubr)
library(MLmetrics)

# b) Loading and Visualizing the data.frame 'oliveoil'
data(oliveoil) # Loading 'oliveoil' from packagae 'pdfCluster'
is.data.frame(oliveoil) # Checking is 'oliveoil' is a data.frame
head(oliveoil) # Checking the first part of the data.frame 'oliveoil'

oliveoil$macro.area = as.character(oliveoil$macro.area);
oliveoil$region = as.character(oliveoil$region) # Changing both categorical columns from factor to
characters
oliveoil$macro.area = gsub(".", "-", oliveoil$macro.area, fixed = TRUE);
oliveoil$region = gsub(".", "-", oliveoil$region, fixed = TRUE) # Replacing dots by spaces in both
columns

summary(oliveoil) # Visualizing the descriptive statistics of the data.frame 'oliveoil'
```

Afin de prendre connaissance de l'ensemble du jeu de données, on procède à un travail préliminaire de validation et de vérification :

- ➔ Y a-t-il des valeurs manquantes observables ?
- ➔ Le data.frame s'affiche-t-il correctement ?
- ➔ Les statistiques descriptives permettent-elles de mieux appréhender d'éventuelles particularités des mesures d'acidité (extrema, moyennes, etc.) ?

Après ces premières investigations réalisées, on peut ensuite exporter le jeu de données sous la forme d'un fichier texte *olives.txt*, issu du data.frame *oliveoil*.

```
# c) Creating a .txt file using the data.frame 'oliveoil'
write.table(oliveoil, file = "oliveoil.txt", sep = ";",
            row.names = FALSE) # Creating a .txt file from the data.frame 'oliveoil'
oliveoil_txt = read.table('oliveoil.txt', sep=';') # Storing the command into an object
```

Ce fichier .txt sera ensuite utilisé comme base dans le prétraitement des données à partir du logiciel de statistiques SAS.

III. Prétraitement des données avec SAS

1. Importation du jeu de données

Dans cette deuxième partie, on cherchera à transformer les données collectées à l'aide d'une transformation log-ratio via le logiciel SAS. Cette transformation garantit une plus grande précision et fiabilité des résultats lors de l'utilisation de méthodes d'analyse en composantes principales ou de partitionnement en k-moyennes.

A partir du travail préliminaire réalisé sous R et de la construction du fichier *oliveoil.txt*, on crée une table SAS que l'on nomme « *olives* » via une étape DATA. On aurait également pu utiliser la procédure IMPORT.

```
/* PART 1 */
/* Test for the current directory. */
DATA _NULL_ ;
 rc=dlgcdir();
 PUT rc=;
RUN;
/* Changing the directory. */
DATA _NULL_ ;
 rc=dlgcdir('D:\3. Studies\1. CNAM\STA115 - Outils Informatiques de la Statistique');
 PUT rc=;
RUN;

/* First method: creating SAS table from oliveoil.txt using DATA statement */
DATA olives;
 LENGTH macro_area $ 20 region $ 20; /* Taking into account long strings /*
 INFILE 'oliveoil.txt'
 DELIMITER=';'
 FIRSTOBS=2;
 INPUT macro_area$ region$ palmitic palmitoleic stearic oleic linoleic linolenic arachidic eicosenoic;
  macro_area = DEQUOTE(macro_area);
  region = DEQUOTE(region); /* Dequoting both categorical columns /*
 DATALINES;
/* Checking SAS table by printing it */;
PROC PRINT DATA=olives;
RUN;

/* Second method: creating SAS table from oliveoil.txt using PROC IMPORT */
PROC IMPORT
 DATAFILE= 'D:\3. Studies\1. CNAM\STA115 - Outils Informatiques de la Statistique\oliveoil.txt'
 OUT = olives
 DBMS = DLM
 REPLACE;
 DELIMITER = ';';
/* Checking SAS table by printing it */
PROC PRINT DATA=olives;
RUN;
```

2. Vérification des mesures d'acidité

Afin de vérifier le niveau de précision des relevés expérimentaux effectués en amont, on calcule la somme des huit mesures d'acidité pour chacune des huiles que l'on stocke dans une variable nommée « *total_acidity* ».

```
/* PART 2 */
/* Calculating the total acidity of each olive oil and processing to a quality check (must be equal to
10000) */
DATA olives_sum;
 SET olives;
 total_acidity = SUM( of palmitic palmitoleic stearic oleic linoleic linolenic arachidic eicosenoic );
 acidity_quality_check = 10000= SUM( of palmitic palmitoleic stearic oleic linoleic linolenic arachidic
eicosenoic );
/* Printing the total acidity and the acidity check for each olive oil */
PROC PRINT
 DATA=olives_sum;
RUN;
/* Counting the number of oils that have an acidity exactly equal to 10000 */
PROC MEANS
 DATA=olives_sum
 SUM MAXDEC=0;
 VAR acidity_quality_check;
RUN;
```

En théorie, cette somme est censée être égale à 10000 pour toutes les huiles identifiées. Toutefois, on observe grâce à une seconde variable nommée « *acidity_quality_check* » que ce n'est le cas que pour seulement 71 d'entre elles.

3. Transformation Log-ratio additive

Afin d'accroitre la précision de l'étude, on décide d'appliquer une transformation log-ratio additive sur l'ensemble des mesures d'acidité du jeu de données.

a) Positivité stricte des valeurs & calcul des ratios

Pour ce faire, on ajoute une unité à la valeur de chaque mesure d'acidité afin de s'assurer que toutes ces valeurs soient strictement positives, sans quoi la transformation log-ratio rencontrerait des erreurs.

```
/* PART 3 */
/* Adding one unit to each acidity measure to prevent errors when transforming data using logratio */
DATA olives_logratio;
 SET olives;
 ARRAY inc {*} _NUMERIC_;
   DO i = 1 TO DIM(inc);
   inc(i) = inc(i)+1;
   END;
RUN;
```

On divise ensuite les mesures résultantes de chaque huile par leur somme, puis on les multiplie par 10000.

```
/* Modifying the acidity data using log-ratio transformation */
DATA olives_logratio;
 SET olives_logratio;
 total_acidity_pluseight = SUM( of palmitic palmitoleic stearic oleic linoleic linolenic arachidic
eicosenoic );
 ARRAY inc {*} _NUMERIC_;
   DO i = 1 TO DIM(inc);
   inc(i) = (inc(i)/total_acidity_pluseight)*10000;
   END;
DATA olives_logratio;
 SET olives_logratio (DROP=i--total_acidity_pluseight);
PROC PRINT
 DATA=olives_logratio;
 TITLE 'Transformation Log-Ratio performed on olive oils acidity';
RUN;
```

En théorie, on devrait obtenir une somme des huit mesures d'acidité de chaque huile égale à exactement 10000.

L'affichage des résultats obtenus semble indiqué cela :

Obs.	macro_area	region	palmitic	palmitoleic	stearic	oleic	linoleic	linolenic	arachidic	eicosenoic	Total
1	South	Apulia-north	1075.57	75.970	226.909	7820.87	672.73	36.9852	60.976	29.9880	10000
2	South	Apulia-north	1088.56	73.970	224.910	7706.92	781.69	31.9872	61.975	29.9880	10000
3	South	Apulia-north	911.64	54.978	246.901	8110.76	549.78	31.9872	63.974	29.9880	10000
4	South	Apulia-north	966.52	57.971	240.880	7949.03	619.69	50.9745	78.961	35.9820	10000
5	South	Apulia-north	1051.58	67.973	259.896	7768.89	672.73	50.9796	80.968	46.9812	10000
6	South	Apulia-north	911.73	49.985	268.919	7922.62	678.80	51.9844	70.979	44.9865	10000
7	South	Apulia-north	922.82	66.987	264.947	7989.40	618.88	49.9900	56.989	29.9940	10000
8	South	Apulia-north	1100.56	61.975	235.906	7725.91	734.71	39.9840	64.974	35.9856	10000
9	South	Apulia-north	1082.46	60.970	239.880	7742.13	709.65	46.9765	83.958	33.9830	10000
10	South	Apulia-north	1038.21	56.011	214.043	7946.59	634.13	27.0054	53.011	31.0062	10000

Toutefois, par souci de validation, on souhaite vérifier la qualité des résultats obtenus. Pour cela, on construit deux variables de vérification permettant de s'assurer de la positivité et de l'exactitude des mesures calculées : « *logratio_quality_check* » et « *logratio_positivity_check* ».

```
/* Processing to a quality check on new data transformed */
/* 1/ Checking if the sum of the new transformed data is equal to 10000 */
/* 2/ Checking if all the new transformed data are stricly positive*/
/* 3/ Selecting the rows that verify these two conditions (hopefully none) */
DATA olives_logratio_quality_check;
 SET olives_logratio;
total_acidity_logratio=sum( of palmitic palmitoleic stearic oleic linoleic linolenic arachidic
eicosenoic );
logratio_quality_check=10000-ROUND(total_acidity_logratio, .000001);
logratio_positivity_check=total_acidity_logratio>0;
PROC PRINT
 DATA=olives_logratio_quality_check;
 VAR macro_area region total_acidity_logratio logratio_quality_check logratio_positivity_check;
 WHERE logratio_quality_check=1
 OR logratio_positivity_check=0;
RUN;
```

Cette étape de vérification valide bien la fiabilité des résultats obtenus.

b) Passage au logarithme

Dès lors, on peut appliquer la transformation log-ratio additive pour chaque acide en remplaçant les valeurs des mesures actuelles par le rapport entre leur logarithme respectif et le logarithme de l'acide eicosénoïque qui servira de base de centrage. On s'autorise à arrondir les nouvelles valeurs obtenues des suites de cette transformation à la deuxième décimale.

```
/* PART 4 */
/* Performing the transformation logratio additive (rounded) */
DATA olives_logratio;
 SET olives_logratio;
 ARRAY inc {*} _NUMERIC_;
   DO i = 1 TO DIM(inc);
   inc(i) = ROUND(LOG(inc(i))/log(eicosenoic),.01);
   END;
DATA olives_logratio;
 SET olives_logratio (DROP=i eicosenoic total_acidity_logratio logratio_quality_check
logratio_positivity_check);
PROC PRINT
 DATA=olives_logratio;
 RUN;
```

La variable eicosénoïque ayant servie comme base de centrage est désormais une constante unitaire. Aussi, on pourra la sortir de l'étude afin de gagner en lisibilité. Pour les dix premières huiles, on obtient alors les résultats suivants :

Obs.	macro_area	region	palmitic	palmitoleic	stearic	oleic	linoleic	linolenic	arachidic
1	South	Apulia-north	2.05	1.27	1.60	2.64	1.91	1.06	1.21
2	South	Apulia-north	2.06	1.27	1.59	2.63	1.96	1.02	1.21
3	South	Apulia-north	2.00	1.18	1.62	2.65	1.86	1.02	1.22
4	South	Apulia-north	1.92	1.13	1.53	2.51	1.79	1.10	1.22
5	South	Apulia-north	1.81	1.10	1.44	2.33	1.69	1.02	1.14
6	South	Apulia-north	1.79	1.03	1.47	2.36	1.71	1.04	1.12
7	South	Apulia-north	2.01	1.24	1.64	2.64	1.89	1.15	1.19
8	South	Apulia-north	1.95	1.15	1.52	2.50	1.84	1.03	1.16
9	South	Apulia-north	1.98	1.17	1.55	2.54	1.86	1.09	1.26
10	South	Apulia-north	2.02	1.17	1.56	2.62	1.88	0.96	1.16

La transformation log-ratio ayant été appliquée avec succès, on peut désormais exporter ces nouvelles données transformées au sein d'un fichier nommé *olives.txt*.

```
/* PART 5 */
/* Creating text from 'olives_logratio' SAS table (additive logratio transformation of the initial
oliveoil dataset) */
PROC EXPORT
 DATA=olives_logratio
 OUTFILE='D:\3. Studies\1. CNAM\STA115 - Outils Informatiques de la Statistique\olives.txt'
 DBMS=dlm REPLACE;
 DELIMITER = ';';
RUN;
```

IV. Analyse statistique avec R

Dans cette troisième et dernière partie, on cherchera à construire une classification des huiles en plusieurs groupes en combinant les méthodes d'analyse en composante principale et de partitionnement des K-means. On pourra ensuite visualiser les résultats obtenus graphiquement et analytiquement.

1. Importation et visualisation du jeu de données transformé

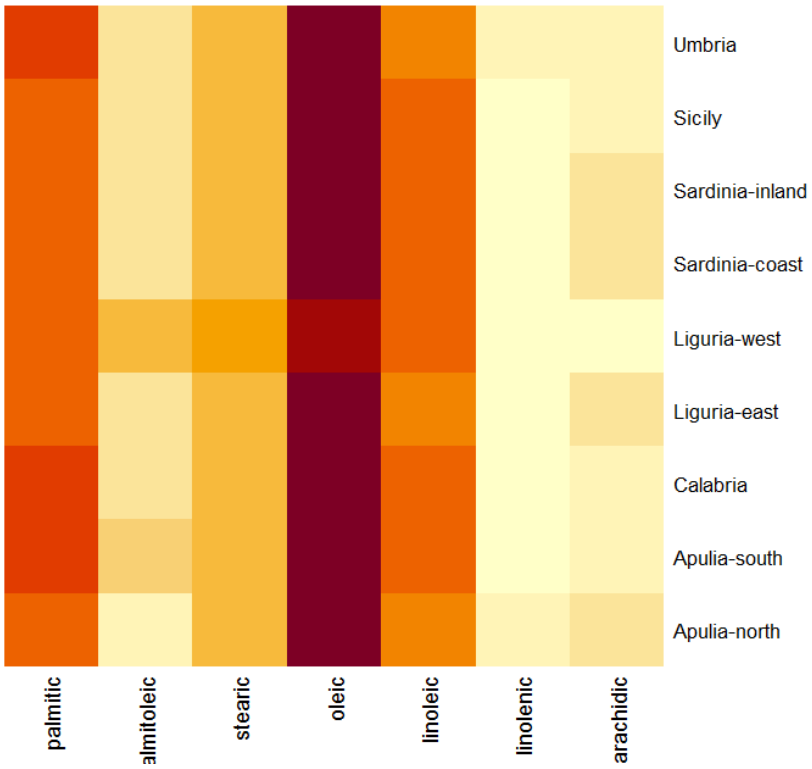
On importe les données transformées dans un data.frame, nommé « olives », à partir du fichier *olives.txt*.

```
### Part 2: Performing a statistical analysis
## 1) Creating a dataframe from the SAS table 'olives.txt'
olives_df = read.table("olives.txt", sep=";", header=TRUE) # Importing the dataset
class(olives_df) # Checking the class of the object

## 2) Building a 9x7 matrix showing the average measures for each acid and displaying them on a heatmap
M = aggregate(x=olives_df[,3:9],
          by=list(region=olives_df$region),
          FUN=mean) # Building a pivot table by aggregating the data
rownames(M) = M$region # Giving names to each row of M (row index)
M = M[,2:8] # Removing the column 'region'
heatmap(as.matrix(M),Colv = NA, Rowv = NA) # Building a heatmap to represent visually the data
```

Puis, on construit une matrice (9x7) permettant d'agréger l'ensemble des mesures par moyenne de chaque acide pour chaque région.

Afin de mieux visualiser le résultat de cette agrégation, on décide d'utiliser la fonction heatmap qui permet d'avoir un premier bon aperçu des valeurs moyennes observées.

Il semblerait que les valeurs les plus importantes soient observées pour les acides oléiques, linoléiques et palmitiques.



2. Sélection des plans factoriels par la méthode de l'ACP

Afin d'améliorer la qualité de la classification, on souhaite d'abord procéder à une analyse en composante principale sur les données transformées afin de sélectionner le ou les plans factoriels à conserver.

```
## 3) Proceeding to a Principal Component Analysis on the 7 variables included into the dataframe
'olives.txt'

# Note 1: we decided to perform the PCA on the whole 'olives.txt' dataset
# Note 2: we could have used the previous matrix M which is based on the whole 'olives.txt' dataset but
with aggregated measures (mean)
pca_olives = PCA(olives_df, quali.sup=1:2) # Performing the PCA ;
pca_olives$eig # Giving the eigenvalues, the percentage of variance and the cumulative percentage of
variance for each component
pca_olives$eig[2,3]  # Returning the cumulative percentage of variance explained for the second
component (6.22% + 92.74% = 0.9896%)
```

On observe alors que le pourcentage de variance expliqué par le premier plan factoriel, c'est-à-dire la somme des deux premières composantes principales, s'élève à 98.96%. En d'autres termes, l'utilisation de ce plan factoriel permet de recueillir près de 99% de l'information initiale. Cet excellent résultat confirme l'intérêt d'utiliser l'analyse en composante principale comme base de travail dans le partitionnement en K-moyennes. On décide donc de conserver seulement les deux premières composantes de l'ACP pour la suite de l'analyse.

En visualisant le graphe des individus (voir graphe 1.1 en annexe), on observe un premier regroupement très clair du secteur sud sur la gauche du graphe. Les secteurs de la Sardaigne et du Centre-Nord semblent quant à eux se partager trois groupes distincts. Un premier coup d'œil au graphe des variables (voir graphe 1.2 en annexe) permet également de distinguer que, en dehors des acides linoléiques et arachidiques, les acides ont une forte corrélation entre eux.

3. Visualisation des huiles à partir de la méthode de l'ACP

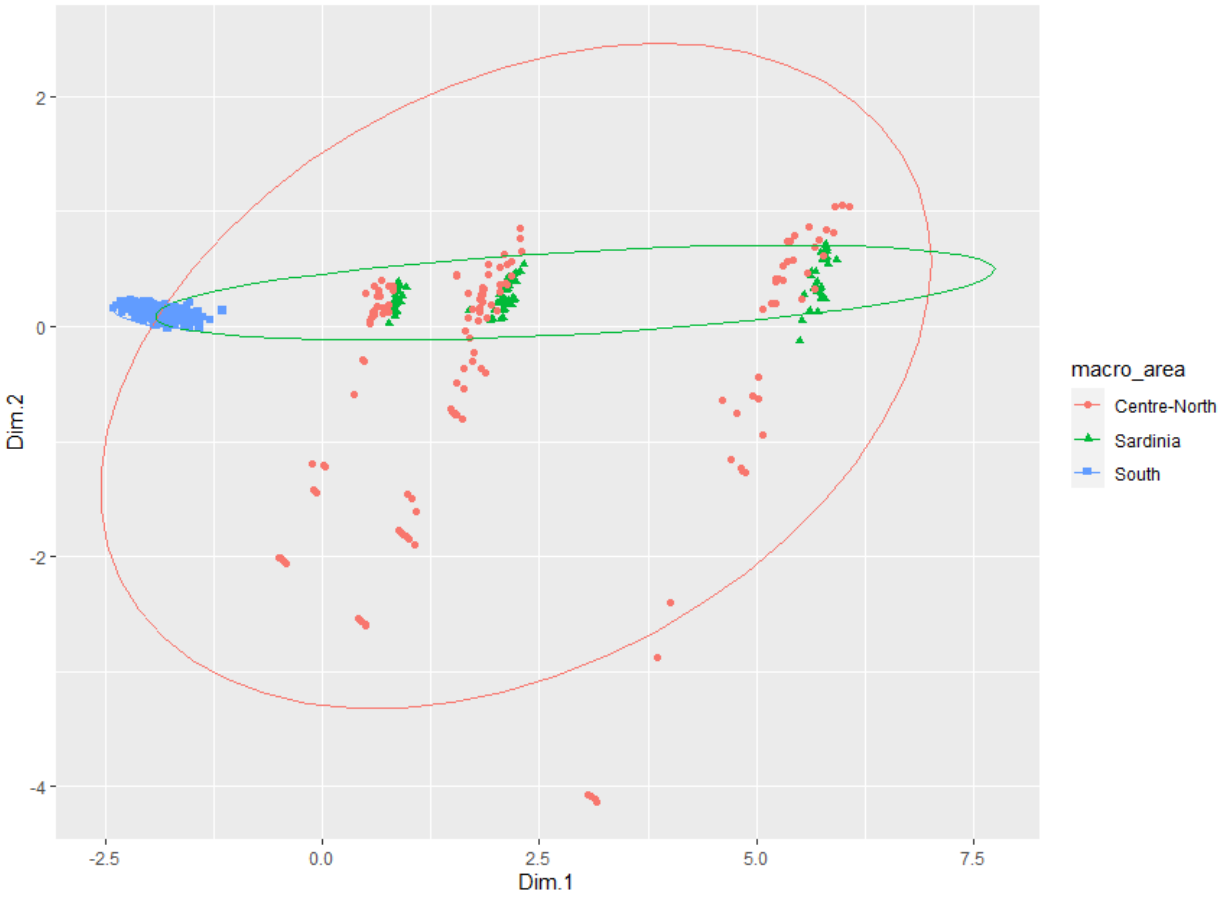
On stocke les coordonnées des huiles des deux premières composantes principales au sein d'un data.frame, nommé « Z », comportant également les deux variables catégorielles « *macro_area* » et « *region* », le data.frame *Z_ind_coord* contenant quant à lui uniquement les deux variables continues.

```
## 4) Building a data.frame with the coordinates of the first two principal components
Z_ind_coord = as.data.frame(x=pca_olives$ind$coord[,1:2]) # Storing the coordinates of the PCA's first
two axes

Z = merge(olives_df[,1:2],
          Z_ind_coord, # Adding the row labels 'macro_area' and 'region' from original dataset
          by.x = 0,
          by.y = 0)
Z$Row.names = as.numeric(Z$Row.names) # Converting 'Row.names' column into numeric values
row.names(Z) = Z$Row.names # Changing the row index values
Z = subset(Z, select = -c(Row.names) ) # Removing 'Row.names' column
Z = Z[order(as.numeric(rownames(Z))),,drop=FALSE] # Reordering dataframe using to row index
```

On peut ensuite tracer le nuage de points représentant les huiles sur le plan défini par les coordonnées retenues. En outre, on peut également tenir compte du secteur d'origine (*macro_area*) dans le tracé du nuage de point afin de mettre en évidence d'éventuels motifs ou regroupements.

```
## 5) Plotting the coordinates of olives dataset's first two principal components
# a) Identifying 'macro_area' with different colors and shapes
Z_plot_macro_area = ggplot(Z, aes(x=Dim.1, y=Dim.2,
                                  shape=macro_area, color=macro_area)) + geom_point()

Z_plot_macro_area + stat_ellipse(type = "norm") # Standardizing the values and plotting ellipses again

# b) # Identifying 'region' with different colors
Z_plot_region = ggplot(Z, aes(x=Dim.1, y=Dim.2,
                              color=region))

Z_plot_region + geom_point() + stat_ellipse()
```

On obtient alors la figure suivante :



On observe alors quatre regroupements distinctifs, correspondant plus ou moins aux secteurs d'origine des huiles. En effet, seul le secteur du Sud semble pleinement défini. Les secteurs de la Sardaigne et du Centre-Nord semblent être répartis au sein des trois autres grappes selon le même motif géométrique.

4. Partitionnement en k-moyennes

Une fois le travail de sélection du plan factoriel et de recueil des coordonnées des différents individus effectués, on peux procéder à la classification des huiles en neuf partitions avec la méthode des K-means à partir du data.frame *Z*. Pour ce faire, l'algorithme est calibré pour choisir le meilleur des 25 premiers centroïdes calculés aléatoirement.

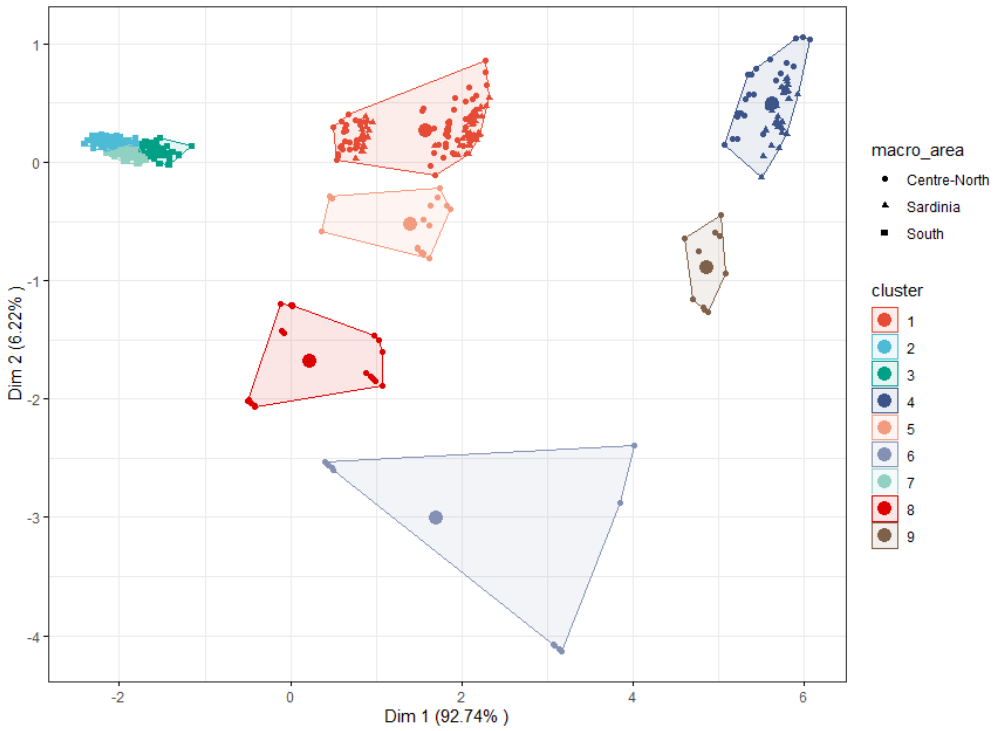
```
## 6) Performing a K-Means clustering analysis on the new data.frame 'Z'
# a) Running kmeans() package with the right parameters
Z_kmeans = kmeans(scale(Z[, 3:4]),
                  9, # Setting the number of clusters to 9
                  nstart = 25) # Generating 25 initial random centroids and choose the best one

# b) Adding to the initial dataset (including the individuals coordinates from PCA) the k-mean clusters
Z_ind_coord$cluster = as.factor(Z_kmeans$cluster)
Z_ind_coord$macro_area = olives_df$macro_area
Z_ind_coord$region = olives_df$region # Adding the 'olive_df' labels (macro_area and 'region')
```

5. Visualisation des huiles à partir de la méthode des K-moyennes

Après avoir ajouté les variables catégorielles aux coordonnées calculées à partir de la méthode des k-means, on peut ensuite visualiser le nuage de points ainsi obtenu pour les individus.

```
## 7) Visualizing the scatterplot of all individuals using the coordinates
ggscatter(Z_ind_coord,
          x = 'Dim.1', y = 'Dim.2', # Giving names to both labels
          color = 'cluster', shape = 'macro_area', # Displaying individuals according to their k-mean
cluster (color) and macro_area (shape)
  palette = "npg", ellipse = TRUE, ellipse.type = 'convex',
  size = 1.5,  legend = "right", ggtheme = theme_bw(),
  xlab = paste0('Dim 1 (', round(pca_olives$eig[1,2],2), '% )' ), # Giving the percentage of variance
explained for the first component
  ylab = paste0('Dim 2 (', round(pca_olives$eig[2,2],2), '% )' ) # Giving the percentage of variance
explained for the second component
) + stat_mean(aes(color = cluster), size = 4)
```

Les groupes d'individus sont désormais clairement identifiés et identifiables. On distingue très nettement neuf classes à part entière :

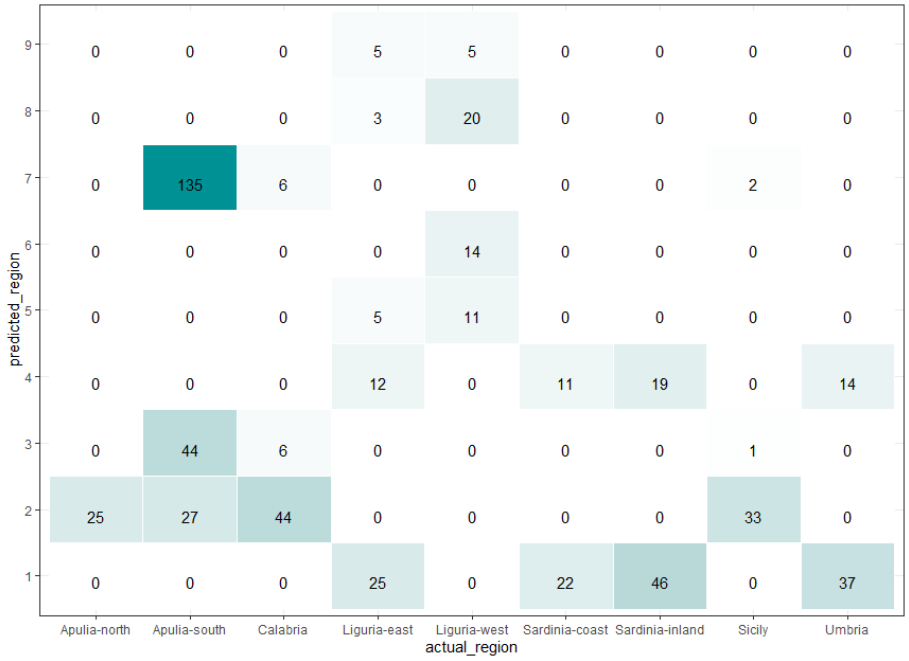


6. Comparaison des groupes statistiquement calculés aux groupes initialement mesurés

Il s'agit désormais de comparer ces neuf groupes, calculés statistiquement, aux prélèvements expérimentaux effectués en amont. L'objectif est de mesurer la fiabilité de l'analyse en comparant les résultats obtenus aux mesures initiales. Pour ce faire, on construit une matrice de confusion.

```
## 8) Creating a confusion matrix from the k-means analysis using the clusters as predictors
# a) Integrating all the required variables into a data.frame
z_conf_matrix_df = Z_ind_coord[,c(3,5)]
colnames(z_conf_matrix_df) = c('predicted_region','actual_region')
actual_region = factor(z_conf_matrix_df$actual_region)
predicted_region = factor(z_conf_matrix_df$predicted_region)

# b) Building the confusion matrix using frequencies
z_conf_matrix = ConfusionDF(predicted_region, actual_region)
z_conf_count_df = as.data.frame(z_conf_matrix) # Storing the object into a data.frame
colnames(z_conf_count_df) = c('actual_region','predicted_region','count')

# c) Visualizing the confusion matrix using frequencies
ggplot(data =  z_conf_count_df, mapping = aes(x = actual_region, y = predicted_region)) +
  geom_tile(aes(fill = count), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", count)), vjust = 1) +
  scale_fill_gradient(low = "white", high = "#009194") +
  theme_bw() + theme(legend.position = "none")
```

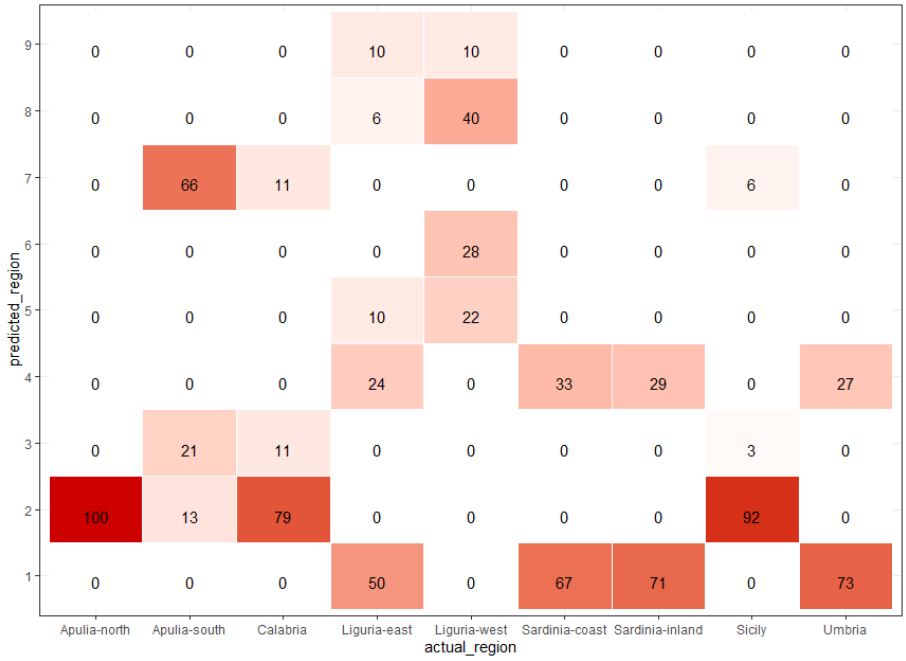
On obtient le tableau comparatif suivant :



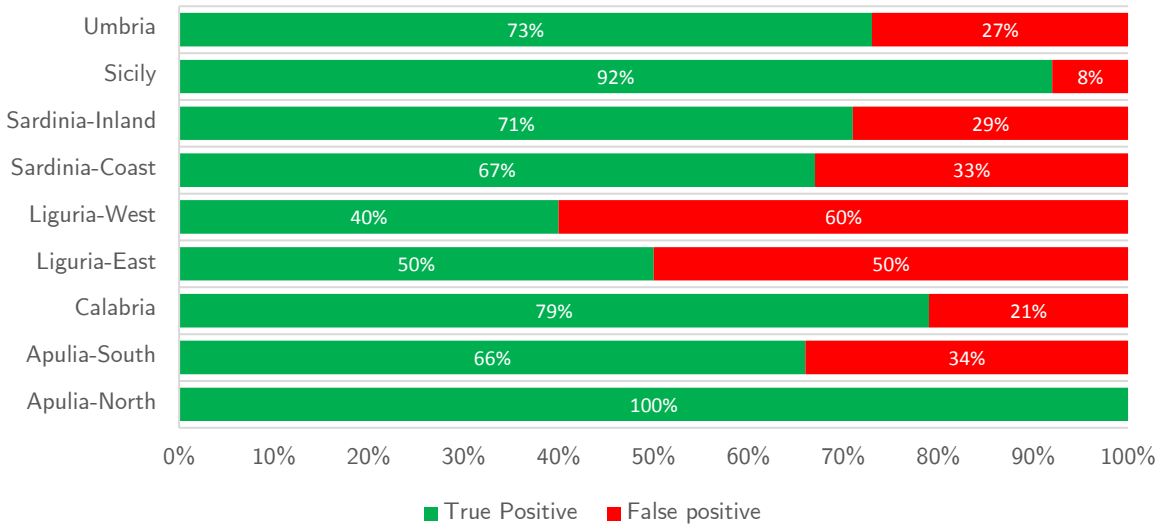
L'analyse des fréquences permet de faire une première idée de la fiabilité des résultats obtenus. Toutefois, le calcul des pourcentages permet de mieux distinguer le taux d'huiles bien classées grâce à la méthode des K-means.

```
# d) Building the confusion matrix using percentage
z_conf_percent_df = z_conf_count_df
#z_conf_percent_df$percent = tapply(z_conf_percent_df$count, z_conf_percent_df$actual_region, FUN=sum)/
z_conf_percent_df = transform(z_conf_percent_df, percent = ave(count, actual_region, FUN = prop.table))
z_conf_percent_df = z_conf_percent_df[,-3]
z_conf_percent_df$percent = round(z_conf_percent_df$percent*100,2)

# e) Visualizing the confusion matrix using percentage
ggplot(data =  z_conf_percent_df, mapping = aes(x = actual_region, y = predicted_region)) +
  geom_tile(aes(fill = percent), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", percent)), vjust = 1) +
  scale_fill_gradient(low = "white", high = "red3") +
  theme_bw() + theme(legend.position = "none")
```

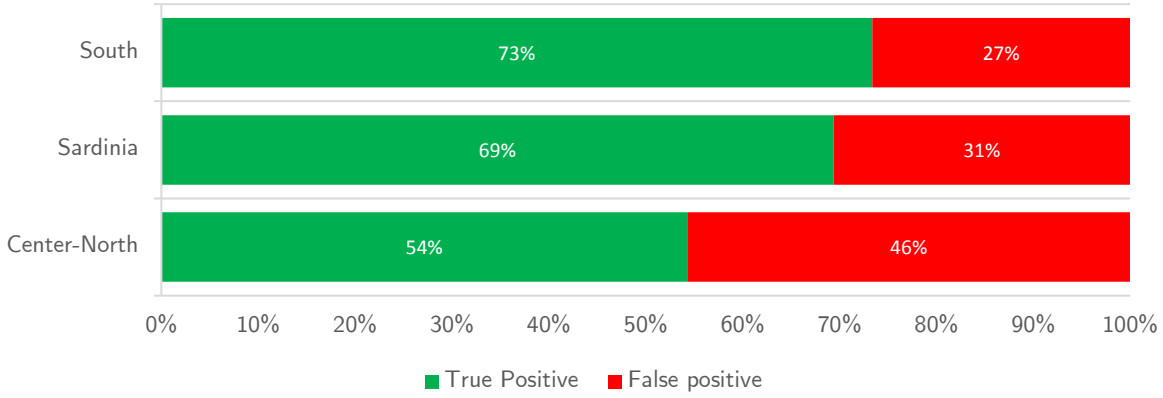
Les résultats obtenus au sein de ce second tableau comparatif ci-dessous sont indiqués en pourcentage :



Si l'on observe le taux de bien classés par région, on peut conclure que la région du Nord des Pouilles (Apulia-North) est la mieux notée avec 100% de vrais-positifs, bien qu'elle ne comporte que 25 huiles différentes. A l'inverse, la région ouest de Ligurie (Liguria-West) est la moins bien notée avec seulement 40% de vrais-positifs pour un échantillon de 50 huiles.



En pondérant les huiles par région, on observe alors que la région la mieux prédite correspond à la région Sud, suivi de près par la Sardaigne. La région Centre-Nord ferme la marche avec des résultats mitigés, près d'une huile sur deux ayant été mal identifiée.



V. Annexes

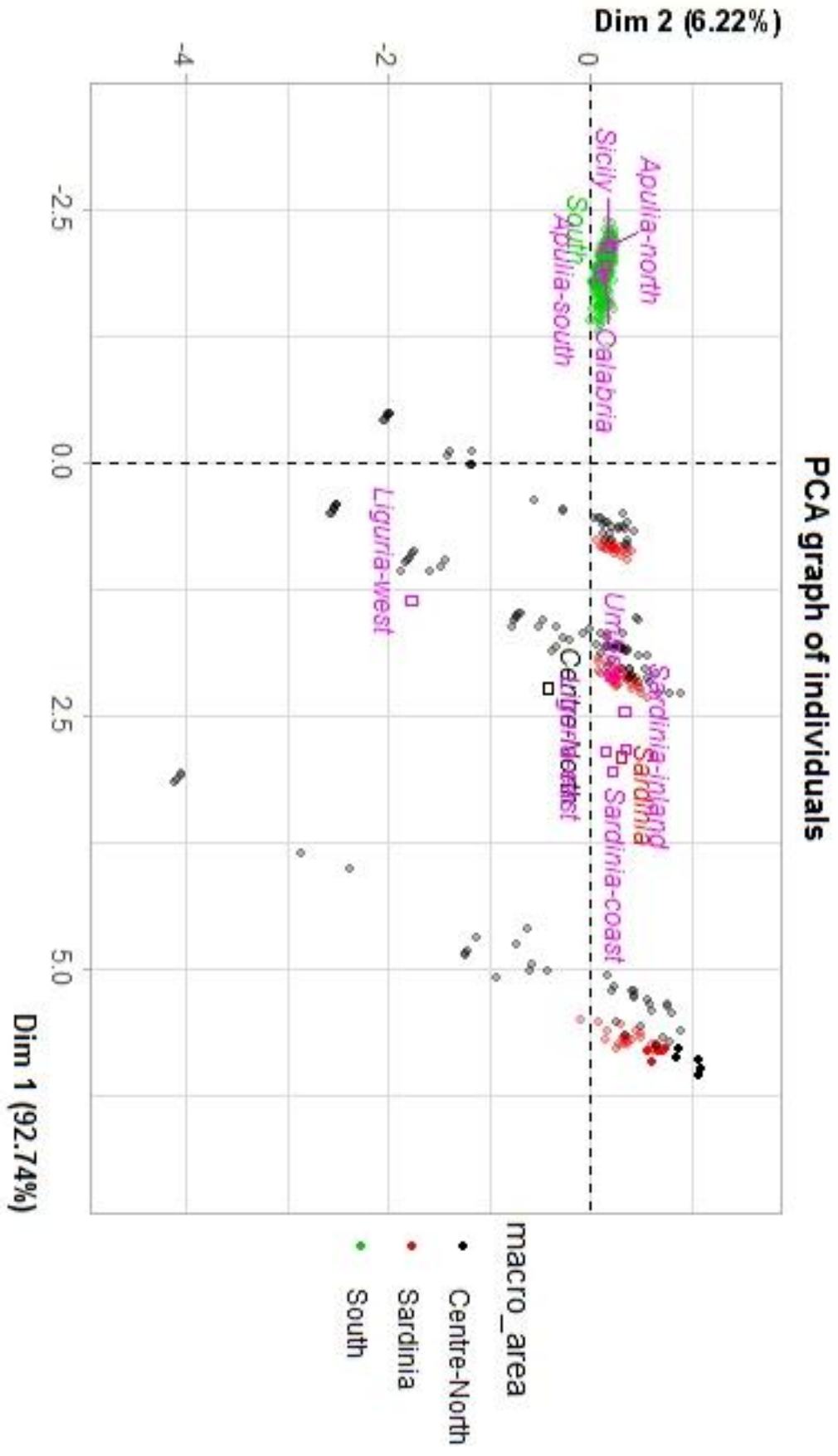


Figure 1.1 – Graphe des individus pour l'ACP

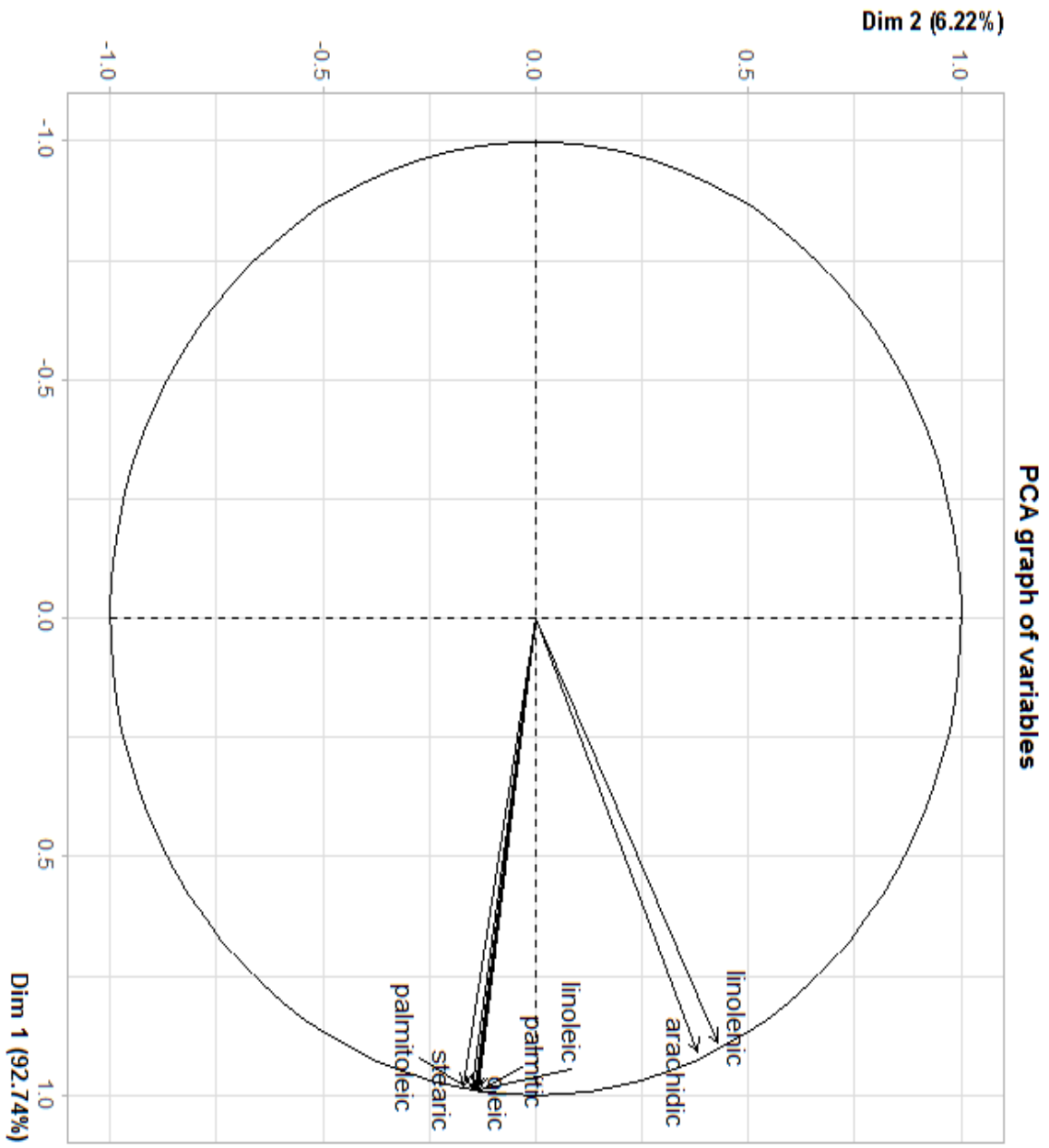


Figure 1.2 – Graphe des variables pour l'ACP