REPUBLIQUE DU BENIN

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université Nationale des Sciences, Technologies, Ingénierie et Mathématiques (UNSTIM)

École Nationale Supérieure de Génie Mathématique et Modélisation (ENSGMM)

THEME

Analyse Multivariée

Unité d'Enseignement : Analyse et Base de données

Filière : ENSGMM - 2

Réalisé par :

- Expéra AKAKPO

Sous la supervision de :

- Dr, Ing. Castro HOUNMENOU
- Ing. Aurel HANSINON

2025-2026

Table of contents

0.1 Exercice 1: *Base Wine*

La base **Wine** provient d'une étude sur des vins issus de trois cépages différents cultivés dans une même région. Chaque vin est décrit par 13 variables quantitatives représentant ses caractéristiques physico-chimiques (teneur en alcool, acidité, magnésium, phénols, intensité de couleur, etc.).

1. Choix de méthode multivariée approprié et justification.

   La base Wine étant composée exclusivement de variables quantitatives décrivant des caractéristiques physico-chimiques potentiellement corrélées, l'Analyse en Composantes Principales (ACP) constitue donc la méthode multivariée la plus appropriée.

2.
```r
# Importation de la base Wine
library(readxl)
wine <- read_excel("wine.xlsx")
View(wine)

# Affichage du nombre d'observations et de variables
nb_observations <- nrow(wine)
nb_variables <- ncol(wine)

cat("Nombre d'observations :", nb_observations, "\n")
```

```
Nombre d'observations : 178
```

```r
cat("Nombre de variables :", nb_variables, "\n")
```

```
Nombre de variables : 14
```

0.1.1 Présentation de la base

La base **Wine** provient d'une étude portant sur des vins issus de **trois cépages différents**, cultivés dans une même région. Chaque observation correspond à un vin, décrit à l'aide de **caractéristiques physico-chimiques**. L'objectif est d'analyser la structure des vins et de comparer les cépages à partir de ces variables quantitatives.

- **Nombre d'observations** : 178 vins
- **Nombre de variables** : 14

13 variables quantitatives (caractéristiques physico-chimiques)

1 variable qualitative (`target`) indiquant le cépage

Description des variables

Variable qualitative

`target` = Cépage du vin (3 modalités : 0, 1, 2)

Variables quantitatives

`alcohol` = Teneur en alcool

`malic_acid` = Acidité malique

`ash` = Teneur en cendres

`alcalinity_of_ash` = Alcalinité des cendres

`magnesium` = Teneur en magnésium

`total_phenols` = Phénols totaux

`flavanoids` = Flavonoïdes

`nonflavanoid_phenols` = Phénols non flavonoïdes

`proanthocyanins` = Proanthocyanines

`color_intensity` = Intensité de la couleur

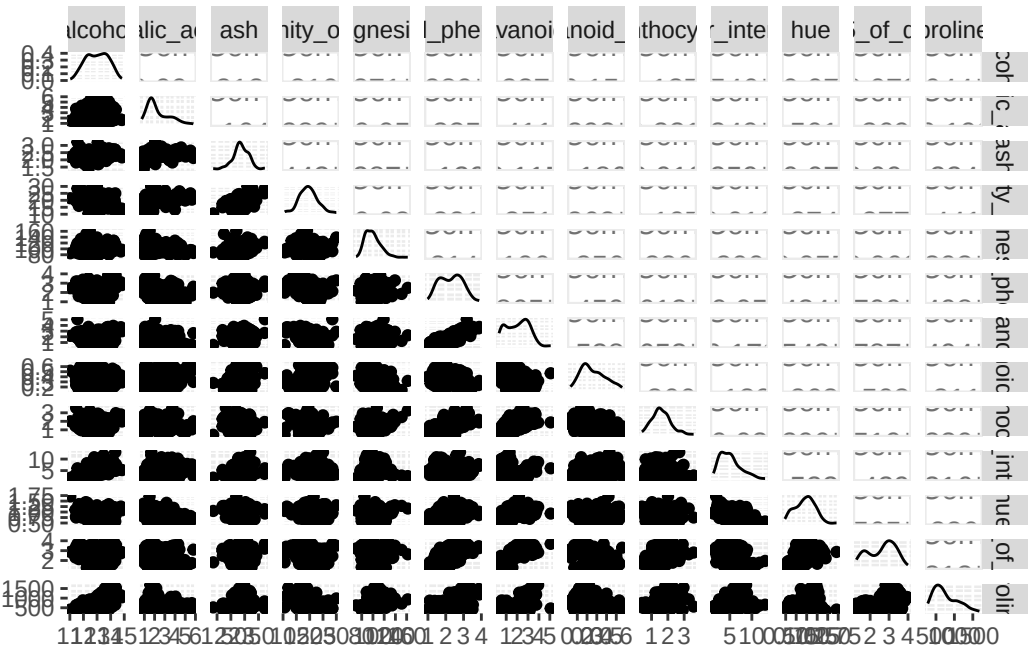
`hue` = Teinte du vin

`od280/od315_of_diluted_wines` = Rapport d'absorbance OD280 / OD315

`proline` = Teneur en proline

0.1.2 ACP sous R

```r
# Exclure la dernière colonne qui contient categories de cépage
wine_ <-  as.data.frame(wine[, -ncol(wine)])
#head(wine_, 5)
# convertir les categories de cepage en facteur
etiquettes <- as.factor(wine[[ncol(wine)]])

# Examiner la matrice de diagramme de dispersion
library(GGally)
ggpairs(wine_)
```




```
# Réaliser l'ACP
library(FactoMineR)
ResACP <- PCA(wine_, graph = FALSE)
ResACP
```

Results for the Principal Component Analysis (PCA)
The analysis was performed on 178 individuals, described by 13 variables
*The results are available in the following objects:

	name	description
1	"$eig"	"eigenvalues"
2	"$var"	"results for the variables"
3	"varcoord"	"coord. for the variables"
4	"varcor"	"correlations variables - dimensions"
5	"varcos2"	"cos2 for the variables"
6	"varcontrib"	"contributions of the variables"
7	"$ind"	"results for the individuals"
8	"indcoord"	"coord. for the individuals"
9	"indcos2"	"cos2 for the individuals"
10	"indcontrib"	"contributions of the individuals"
11	"$call"	"summary statistics"
12	"$call$centre"	"mean of the variables"
13	"$call$ecart.type"	"standard error of the variables"

```
14 "$call$row.w"      "weights for the individuals"
15 "$call$col.w"      "weights for the variables"
```

```r
# Extraire les Valeurs propres de l'ACP
print("\n Valeur propre de l'ACP")
```

```
[1] "\n Valeur propre de l'ACP"
```

```r
head(ResACP$eig, 5)
```

```
       eigenvalue percentage of variance cumulative percentage of variance
comp 1  4.7058503              36.198848                            36.19885
comp 2  2.4969737              19.207490                            55.40634
comp 3  1.4460720              11.123631                            66.52997
comp 4  0.9189739               7.069030                            73.59900
comp 5  0.8532282               6.563294                            80.16229
```

```r
# Extraire le tableau des corrélations des variables avec les
#Composantes Principales
print("\n Correlation des variables avec les composantes")
```

```
[1] "\n Correlation des variables avec les composantes"
```

```r
head(ResACP$var$cor, 5)
```

```
                      Dim.1       Dim.2      Dim.3       Dim.4       Dim.5
alcohol           0.313093350  0.76425725 -0.2493833 -0.01711761  0.24539445
malic_acid       -0.531884726  0.35543171  0.1070404  0.51467982 -0.03252695
ash              -0.004449362  0.49944611  0.7530514 -0.20531539  0.13211313
alcalinity_of_ash -0.519157081 -0.01673492  0.7360433  0.05834174 -0.06105952
magnesium         0.308022936  0.47347612  0.1572388 -0.33724320 -0.67157727
```

```r
#Extraire le tableau des coordonnee pour les catégories de cépages
print("\n Coordonnées des cépages")
```

```
[1] "\n Coordonnées des cépages"
```

```r
head(ResACP$ind$coord, 5)
```

```
     Dim.1      Dim.2      Dim.3      Dim.4      Dim.5
1 3.316751  1.4434626 -0.1657390 -0.2156312 -0.6930428
2 2.209465 -0.3333929 -2.0264574 -0.2913583  0.2576546
3 2.516740  1.0311513  0.9828187  0.7249023  0.2510331
4 3.757066  2.7563719 -0.1761918  0.5679833  0.3118416
5 1.008908  0.8698308  2.0266882 -0.4097658 -0.2984575
```

```r
#Extraire le tableau des qualités pour les catégories de cépages
print("\n Qualité")
```

```
[1] "\n Qualité"
```

```r
head(ResACP$ind$cos2, 5)
```

```
      Dim.1       Dim.2       Dim.3       Dim.4       Dim.5
1 0.6874080 0.130196702 0.001716479 0.002905441 0.030012976
2 0.4261832 0.009703642 0.358506506 0.007411002 0.005795595
3 0.5740096 0.096357876 0.087536511 0.047621317 0.005710899
4 0.5988669 0.322335688 0.001317056 0.013686863 0.004125730
5 0.1430059 0.106296731 0.577064302 0.023589662 0.012514553
```

```r
#Extraire le tableau des contributions pour les catégories de cépages
print("\n Contribution")
```

```
[1] "\n Contribution"
```

```r
head(ResACP$ind$contrib, 5)
```

```
      Dim.1      Dim.2      Dim.3      Dim.4      Dim.5
1 1.3133110 0.46878868 0.01067185 0.02842497 0.31625323
2 0.5827954 0.02500798 1.59538328 0.05189573 0.04371100
3 0.7561686 0.23922734 0.37526398 0.32124456 0.04149319
4 1.6851534 1.70939196 0.01206040 0.19721864 0.06402991
5 0.1215194 0.17022981 1.59574678 0.10264746 0.05865159
```

Les deux premiers axes expliquent **55,4 % de la variance totale**, dont **36,2 % pour l'axe 1** et **19,2 % pour l'axe 2**.

L'axe 1 oppose :

- des vins **riches en flavanoids, total_phenols, proline, hue et alcohol** (coordonnées positives),
- à des vins caractérisés par une **forte acidité malique (malic_acid)** et une **alcalinité élevée des cendres** (coordonnées négatives).

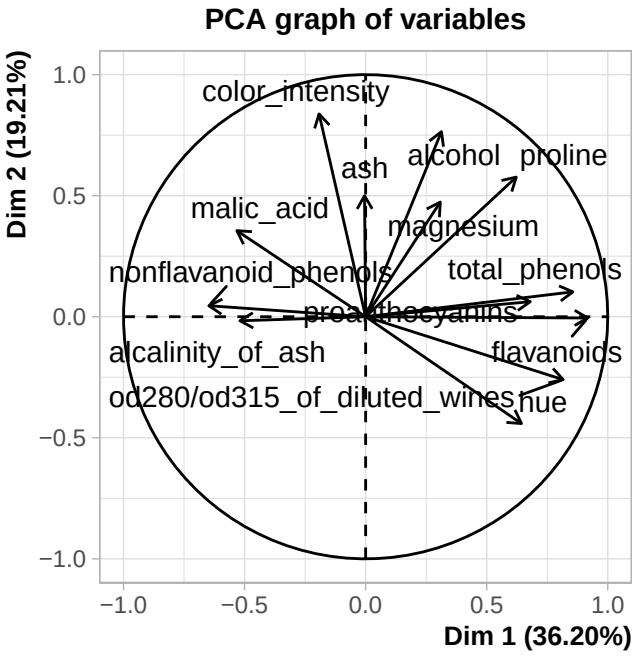
  Cet axe traduit **la richesse phénolique et la maturité du vin**

L'axe 2 est principalement associé à :

- des valeurs élevées de **alcohol, color_intensity, magnesium et ash** (coordonnées positives),
- opposées à des vins ayant des valeurs plus faibles de **hue** et du rapport **od280/od315** (coordonnées négatives).

  Cet axe reflète un **contraste entre intensité colorante et structure minérale du vin**.

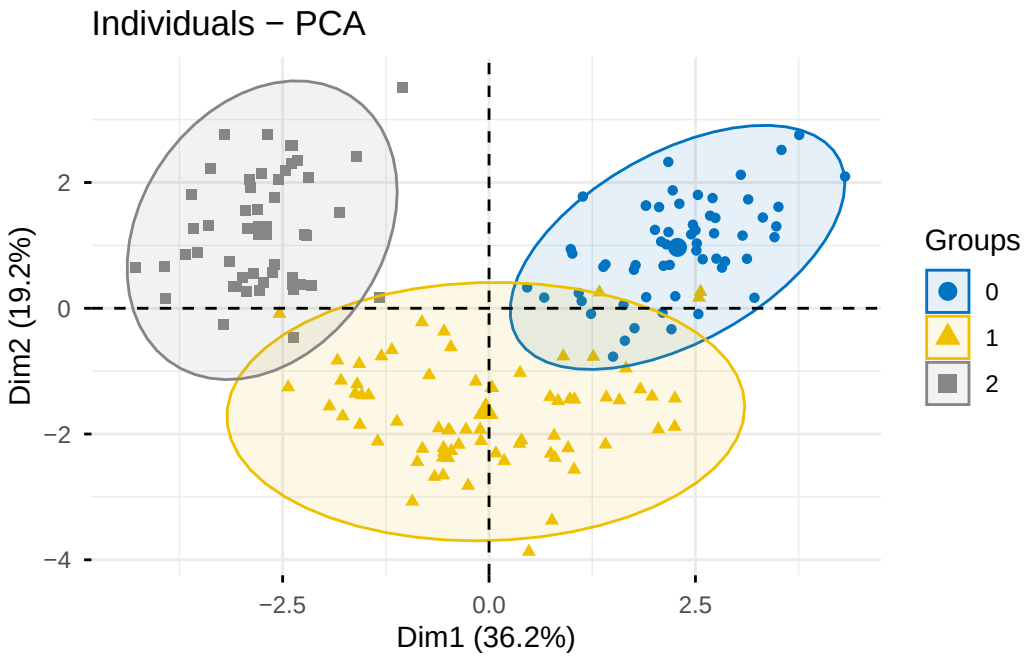
```r
# Afficher le graphe de corrélation des variables avec les Composantes
#Principales 1 & 2
# Encore appellée Cercle de corrélations
plot(ResACP, choix = "var", axes = c(1, 2))
```



Relations entre les caractéristiques physico-chimique:

- Les variables **flavanoids**, **total_phenols** et **proline** sont fortement et positivement corrélées entre elles.
- **malic_acid** et **alcalinity_of_ash** sont corrélées entre elles et opposées aux variables phénoliques.
- Les variables **alcohol** et **color_intensity** contribuent fortement à la structuration du plan factoriel.
- Les variables proches du centre du cercle contribuent faiblement aux deux axes.

```r
# Afficher le graphe de projection des categorie de cepage dans le système
#d'axes formé par les Composante Principales 1 & 2
library(factoextra)
fviz_pca_ind(ResACP,
             habillage = etiquettes,     # colorie par groupe (0,1,2)
             axes = c(1, 2),
             addEllipses = TRUE,     # ellipse autour de chaque groupe
             palette = "jco",
             repel = FALSE,
             geom = "point")
```



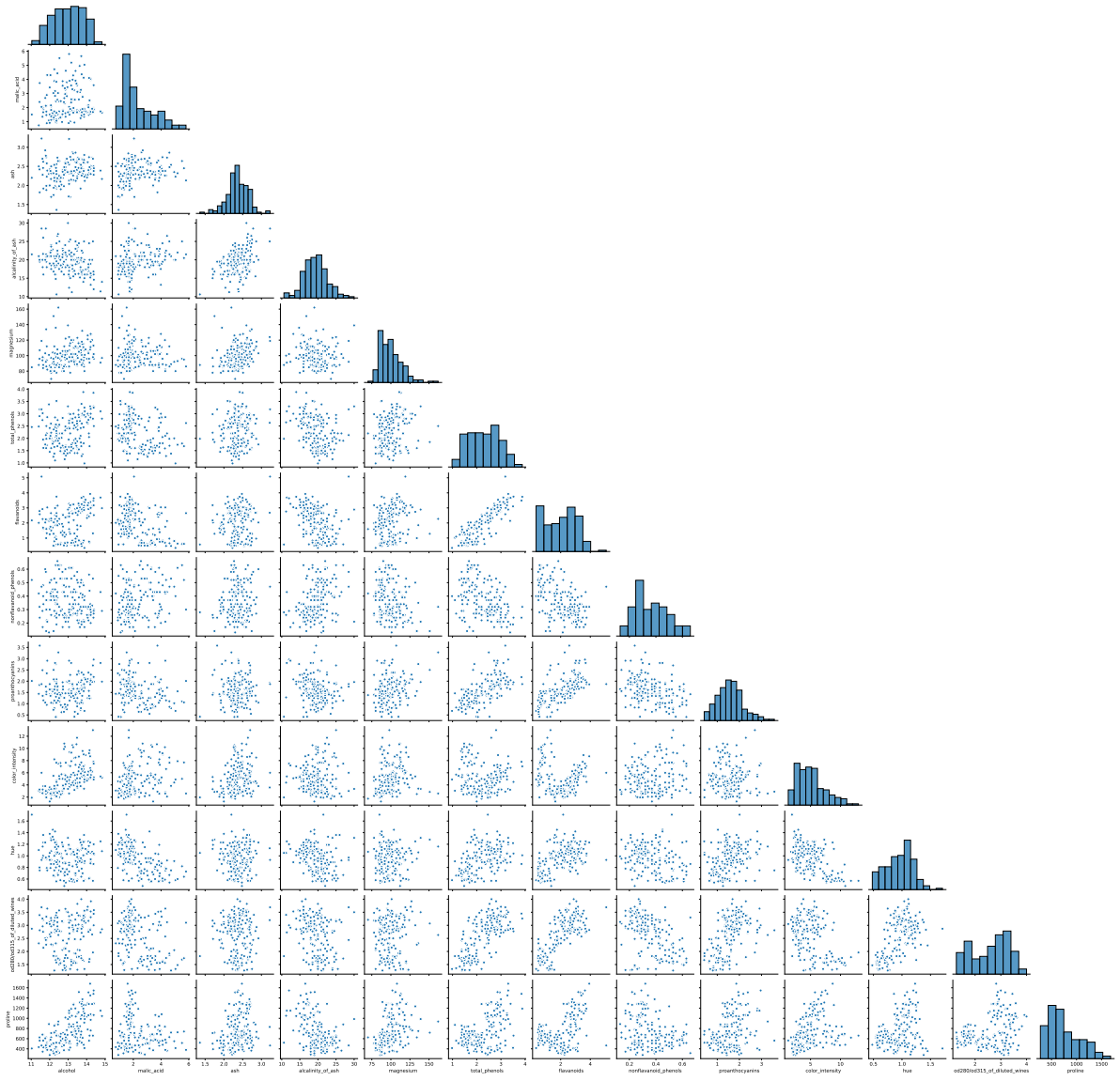
- Les vins du **cépage 0** sont principalement situés du côté positif de l'axe 1, indiquant des vins riches en composés phénoliques et en alcool.
- Les vins du **cépage 1** sont plutôt positionnés vers les valeurs négatives de l'axe 2, traduisant des caractéristiques différentes en termes de couleur et de teinte.
- Les vins du **cépage 2** se situent majoritairement du côté négatif de l'axe 1, associés à une acidité plus marquée.

0.1.3 ACP sous Python

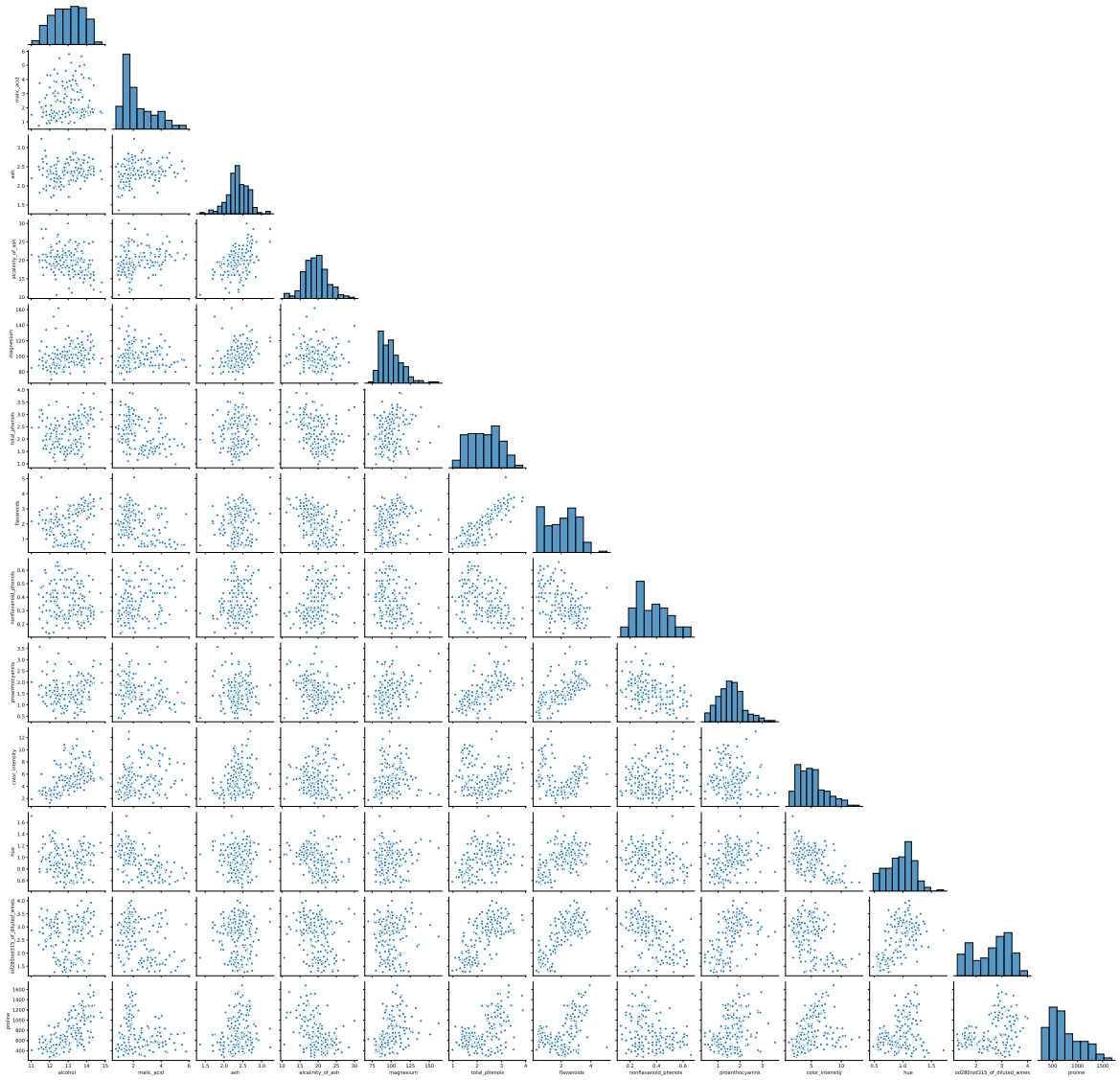
```
library(reticulate)

#virtualenv_create("
#D://Mes_Codes/R/ENSGMM2/analyse_base_donnee/Travaux_de_maison_GMM2/r-python")
use_virtualenv("D://Mes_Codes/R/ENSGMM2/analyse_base_donnee/Travaux_de_maison_GMM2/r-python"

#py_install(c("pandas", "numpy", "matplotlib", "seaborn", "scikit-learn",
#"openpyxl", "prince", "scikit-learn==1.5.2"))
#py_install(c(""))
```

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pathlib import Path
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

base_dir = Path.cwd()
```

```
# Charger la base wine dans un DataFrame
wine = pd.read_excel(base_dir / "wine.xlsx")

#Séparation des variables quantitatives et du cépage
# La dernière colonne correspond au cépage
wine_ = wine.iloc[:, :-1]      # variables quantitatives
etiquettes = wine.iloc[:, -1] # cépages (0, 1, 2)

# Matrice de diagrammes de dispersion
sns.pairplot(wine_, corner=True)
```


alcohol
malic_acid
ash
alcalinity_of_ash
magnesium
total_phenols
flavanoids
nonflavanoid_phenols
proanthocyanins
color_intensity
hue
od280/od315_of_diluted_wines
proline

```
plt.show()
```




```

#Standardisation des données
scaler = StandardScaler()
X_scaled = scaler.fit_transform(wine_)
```

```
#Réalisation de l'ACP
pca = PCA()
X_pca = pca.fit_transform(X_scaled)

```

```
# Valeurs propres et pourcentages d'inertie
eig_values = pca.explained_variance_
explained_var = pca.explained_variance_ratio_ * 100
cumulative_var = np.cumsum(explained_var)

eig_df = pd.DataFrame({
    "eigenvalue": eig_values,
    "percentage of variance": explained_var,
    "cumulative percentage of variance": cumulative_var
})

print("\nValeurs propres de l'ACP")
```

```
Valeurs propres de l'ACP
```

```
print(eig_df.head())
```

```
   eigenvalue  percentage of variance  cumulative percentage of variance
0    4.732437               36.198848                          36.198848
1    2.511081               19.207490                          55.406338
2    1.454242               11.123631                          66.529969
3    0.924166                7.069030                          73.598999
4    0.858049                6.563294                          80.162293
```

```
# Corrélations des variables avec les composantes principales
# Corrélations = loadings
loadings = pca.components_.T * np.sqrt(eig_values)

corr_df = pd.DataFrame(
    loadings,
    index=wine_.columns,
    columns=[f"Dim.{i+1}" for i in range(loadings.shape[1])]
)

print("\nCorrélation des variables avec les composantes")
```

```
Corrélation des variables avec les composantes
```

```
print(corr_df.iloc[:, :5].head())
```

```
                     Dim.1     Dim.2     Dim.3     Dim.4     Dim.5
alcohol            0.313977  0.766413 -0.250087 -0.017166 -0.246087
malic_acid        -0.533385  0.356434  0.107342  0.516132  0.032619
ash               -0.004462  0.500855  0.755176 -0.205895 -0.132486
alcalinity_of_ash -0.520622 -0.016782  0.738120  0.058506  0.061232
magnesium          0.308892  0.474812  0.157682 -0.338195  0.673472
```

```
# Coordonnées des individus
coord_df = pd.DataFrame(
    X_pca,
    columns=[f"Dim.{i+1}" for i in range(X_pca.shape[1])]
)

print("\nCoordonnées des individus")
```

```
Coordonnées des individus
```

```
print(coord_df.head())
```

```
      Dim.1     Dim.2     Dim.3  ...    Dim.11    Dim.12    Dim.13
0  3.316751  1.443463 -0.165739  ... -0.451563  0.540810 -0.066239
1  2.209465 -0.333393 -2.026457  ... -0.142657  0.388238  0.003637
2  2.516740  1.031151  0.982819  ... -0.286673  0.000584  0.021717
3  3.757066  2.756372 -0.176192  ...  0.759584 -0.242020 -0.369484
4  1.008908  0.869831  2.026688  ... -0.525945 -0.216664 -0.079364

[5 rows x 13 columns]
```

```
#Qualité de représentation (cos²)
cos2 = (coord_df ** 2).div((coord_df ** 2).sum(axis=1), axis=0)

print("\nQualité de représentation (cos²)")
```

```
Qualité de représentation (cos²)
```

```
print(cos2.iloc[:, :5].head())
```

```
      Dim.1     Dim.2     Dim.3     Dim.4     Dim.5
0  0.687408  0.130197  0.001716  0.002905  0.030013
1  0.426183  0.009704  0.358507  0.007411  0.005796
2  0.574010  0.096358  0.087537  0.047621  0.005711
3  0.598867  0.322336  0.001317  0.013687  0.004126
4  0.143006  0.106297  0.577064  0.023590  0.012515
```

```
#Contribution des individus
contrib = (coord_df ** 2) / coord_df.shape[0]

print("\nContribution des individus")
```

```
Contribution des individus
```

```
print(contrib.iloc[:, :5].head())
```

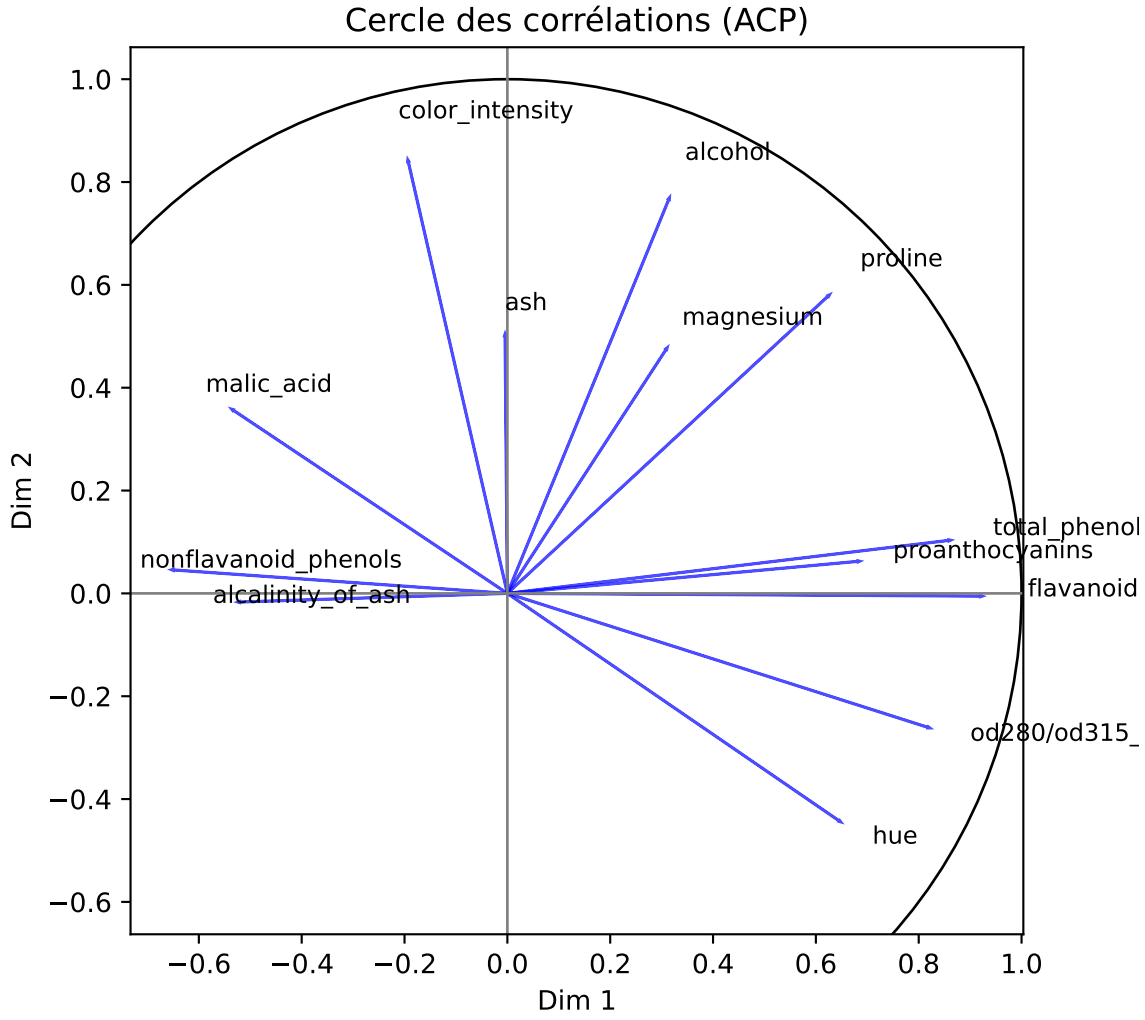
```
      Dim.1     Dim.2     Dim.3     Dim.4     Dim.5
0  0.061802  0.011706  0.000154  0.000261  0.002698
1  0.027425  0.000624  0.023070  0.000477  0.000373
2  0.035584  0.005973  0.005427  0.002952  0.000354
3  0.079301  0.042683  0.000174  0.001812  0.000546
4  0.005719  0.004251  0.023076  0.000943  0.000500
```

```
#Cercle des corrélations (axes 1 et 2)
plt.figure(figsize=(6, 6))

for i, var in enumerate(wine_.columns):
    plt.arrow(0, 0,
              corr_df.iloc[i, 0],
              corr_df.iloc[i, 1],
              color='blue', alpha=0.7)
    plt.text(corr_df.iloc[i, 0]*1.1,
             corr_df.iloc[i, 1]*1.1,
             var, fontsize=9)

circle = plt.Circle((0, 0), 1, color='black', fill=False)
```

```
plt.gca().add_artist(circle)

plt.axhline(0, color='grey', lw=1)
plt.axvline(0, color='grey', lw=1)

plt.xlabel("Dim 1")
plt.ylabel("Dim 2")
plt.title("Cercle des corrélations (ACP)")
plt.axis("equal")
```

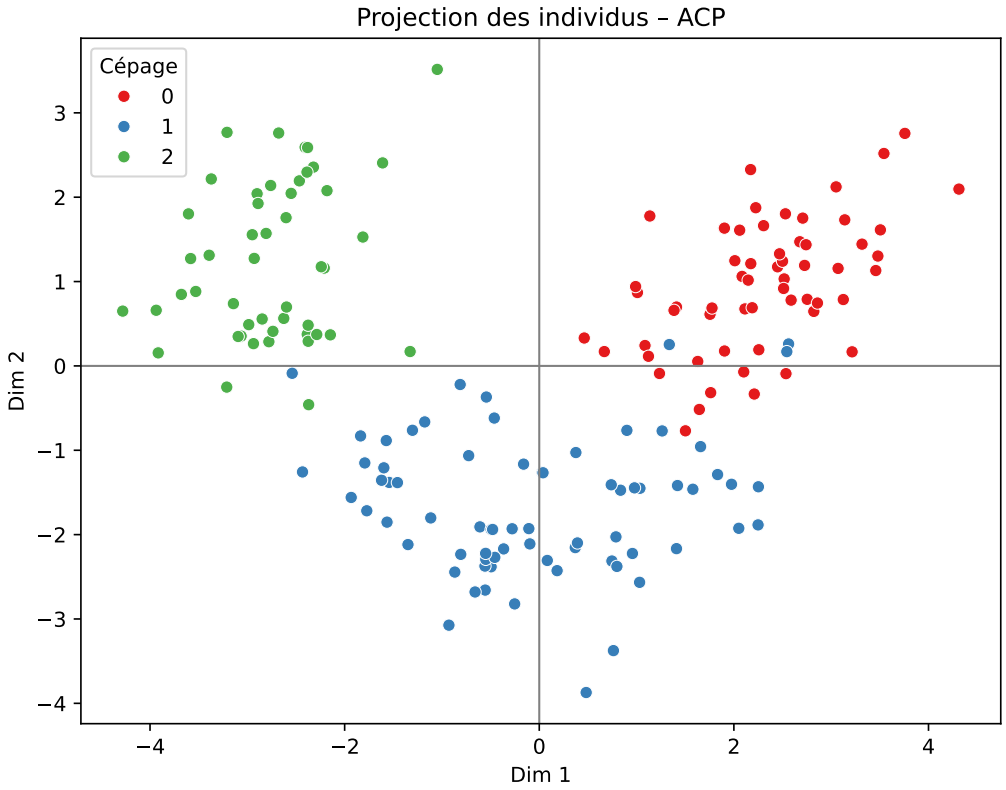
```
(np.float64(-0.732846833540742), np.float64(1.00348222299197), np.float64(-0.509494757896436
```

```
plt.show()
```




```python
# Projection des individus colorés par cépage (axes 1 et 2)
plt.figure(figsize=(8, 6))
sns.scatterplot(
    x=coord_df["Dim.1"],
    y=coord_df["Dim.2"],
    hue=etiquettes,
    palette="Set1"
)

```

```
plt.axhline(0, color='grey', lw=1)
plt.axvline(0, color='grey', lw=1)

plt.xlabel("Dim 1")
plt.ylabel("Dim 2")
plt.title("Projection des individus - ACP")
plt.legend(title="Cépage")
plt.show()
```



En conclusion, Les vins du **cépage 0** se distinguent par des **teneurs élevées en flavanoids, phénols totaux, proline et alcool**, traduisant des vins plus riches et plus structurés. Les vins du **cépage 1** sont caractérisés par une **intensité de couleur plus faible**, une **teinte différente** et des valeurs intermédiaires pour les composés phénoliques. Les vins du **cépage 2** présentent principalement une **acidité malique plus élevée** et une **alcalinité des cendres importante**, indiquant des profils plus acides et moins riches en composés phénoliques.

Ainsi, l'ACP montre que les **composés phénoliques, la teneur en alcool, l'intensité**

colorante et l'acidité sont les variables dominantes dans la discrimination des cépages.

0.2 Exercice 2 : *Base HouseTasks*

La base **HouseTasks** est issue d'une enquête portant sur la répartition des tâches ménagères au sein des couples.

3. Choix de méthode multivarier et justification

   La méthode multivariée la plus adaptée pour analyser cette base est **l'Analyse Factorielle des Correspondances (AFC)**.

   L'Analyse Factorielle des Correspondances (AFC) est la méthode multivariée appropriée car les données forment un **tableau de contingence** croisant deux variables qualitatives : les tâches ménagères et les modes de répartition.

4.

```r
# Importation de la base
library(readxl)
housetasks <- read_excel("housetasks.xlsx")
View(housetasks)

# Affichage du nombre d'observations et de variables
nb_observations <- nrow(housetasks)
nb_variables <- ncol(housetasks)

cat("Nombre d'observations :", nb_observations, "\n")
```

```
Nombre d'observations : 11
```

```r
cat("Nombre de variables :", nb_variables, "\n")
```

```
Nombre de variables : 5
```

0.2.1 Présentation et description de la base HouseTasks

La base **HouseTasks** provient d'une enquête sur la répartition des tâches ménagères au sein des couples. Elle décrit la fréquence (ou le nombre de répondants) associée à chaque mode de répartition pour différentes tâches domestiques. Les modalités de répartition sont les suivantes :

- **Wife** : la tâche est principalement effectuée par la femme.
- **Alternating** : la tâche est alternée entre les deux partenaires.
- **Husband** : la tâche est principalement effectuée par l'homme.
- **Jointly** : la tâche est effectuée conjointement par les deux partenaires.

Les tâches considérées sont :

- Cuisine (Cooking)
- Lessive (Laundry)
- Ménage (Cleaning)
- Courses (Shopping)
- Repassage (Ironing)
- Gestion des finances (Finances)
- Réparations (Repairs)
- Conduite (Driving)
- Jardinage (Gardening)
- Aide aux devoirs (Homework)
- Vaisselle (Dishes)

Chaque cellule du tableau indique le **nombre de ménages** correspondant à une combinaison tâche / mode de répartition. Il s'agit donc d'un **tableau de contingence**.

0.2.2 AFC sous R

```r
# Charger le package requis
library(FactoMineR)

housetasks_tab <- housetasks[, -1]
#  Exécuter l'analyse factorielle des correspondances
CAResults1 <- CA(housetasks_tab, graph = FALSE)
#Afficher les résultats de l'analyse des correspondances.
summary(CAResults1)
```

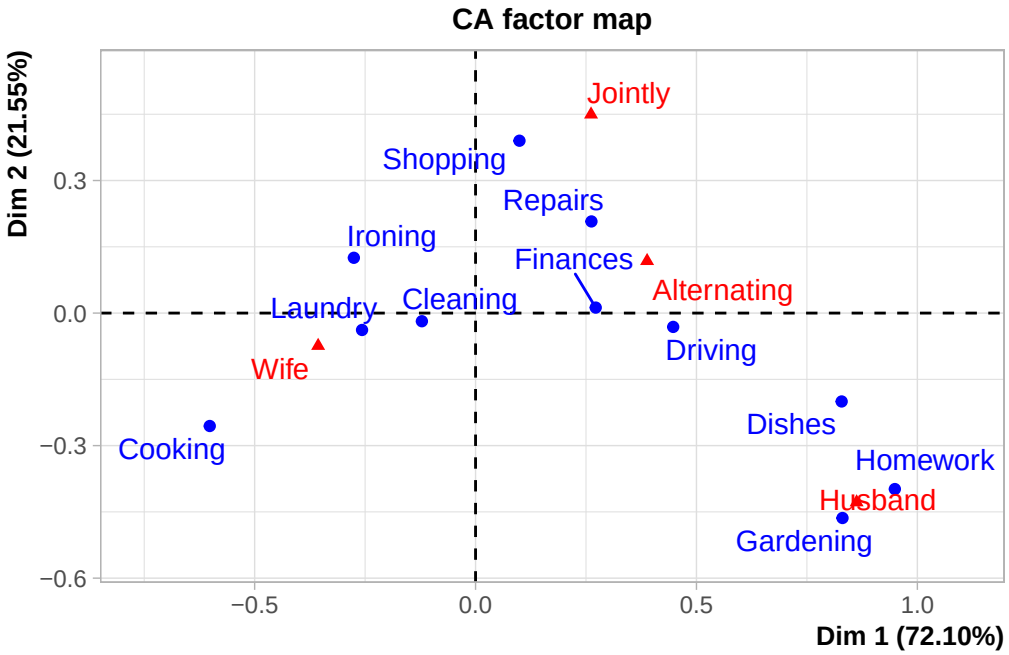
```
Call:
```

```
CA(X = housetasks_tab, graph = FALSE)

The chi square of independence between the two variables is equal to 279.268 (p-value =  3.1

Eigenvalues
                       Dim.1   Dim.2   Dim.3
Variance               0.190   0.057   0.017
% of var.             72.102  21.552   6.346
Cumulative % of var.  72.102  93.654 100.000

Rows (the 10 first)
              Iner*1000    Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3
1           |    71.748 | -0.602 31.638  0.838 | -0.256 19.108  0.151 | -0.066
2           |    11.799 | -0.257  5.867  0.945 | -0.038  0.435  0.021 |  0.048
3           |     5.456 | -0.121  0.894  0.311 | -0.018  0.069  0.007 |  0.180
4           |    28.247 |  0.099  0.888  0.060 |  0.390 45.761  0.921 |  0.057
5           |     6.828 | -0.275  2.977  0.829 |  0.125  2.054  0.171 | -0.001
6           |    12.660 |  0.272  2.756  0.414 |  0.012  0.019  0.001 | -0.323
7           |    10.644 |  0.262  2.802  0.501 |  0.207  5.860  0.313 | -0.160
8           |     6.282 |  0.447  3.280  0.993 | -0.031  0.054  0.005 | -0.022
9           |    33.051 |  0.831 12.676  0.729 | -0.464 13.226  0.227 |  0.202
10          |    36.211 |  0.949 16.104  0.846 | -0.398  9.490  0.149 | -0.076
               ctr   cos2
1            4.374  0.010 |
2            2.370  0.034 |
3           22.214  0.681 |
4            3.289  0.019 |
5            0.001  0.000 |
6           44.281  0.585 |
7           11.866  0.187 |
8            0.091  0.002 |
9            8.559  0.043 |
10           1.183  0.005 |

Columns
              Iner*1000     Dim.1     ctr    cos2     Dim.2     ctr    cos2
Wife        |    74.944 |  -0.356  37.763   0.958 |  -0.074   5.468   0.041 |
Alternating |    38.926 |   0.388  13.476   0.658 |   0.118   4.159   0.061 |
Husband     |   104.142 |   0.863  43.229   0.789 |  -0.428  35.647   0.195 |
Jointly     |    45.698 |   0.261   5.532   0.230 |   0.450  54.727   0.681 |
              Dim.3     ctr    cos2
Wife          0.008   0.207   0.000 |
Alternating  -0.254  65.368   0.281 |
```

```
Husband       0.124  10.076   0.016 |
Jointly       0.163  24.349   0.089 |
```

- Les tâches avec un $\cos^2$ (>50%) sur l'axe 1 sont : cuisine, jardinage, devoirs, vaisselle.
- Sur l'axe 2 : courses, réparations, finances.
- Le test du khi-deux confirme une **forte dépendance** entre tâches et modes de répartition ($p < 0.001$).

```r
# Modifier les noms des lignes
rownames(CAResults1$row$coord) <- housetasks$...1

# Visualiser le résultat sur une graphe avec les nouveaux noms
plot.CA(CAResults1, axes = c(1, 2))
```



- **Axe 1 (72.1%)** oppose les tâches faites par la **femme** (cuisine, lessive, repassage, menage) à celles faites par le **mari** (jardinage, aide aux devoirs, vaisselle).
- **Axe 2 (21.6%)** met en avant les tâches **faites conjointement** (shopping, réparations) ou **alternées** (réparations, finances, conduit), surtout en haut du plan.

0.2.3 AFC sous Python

```

import pandas as pd
import numpy as np
from scipy.linalg import svd
import matplotlib.pyplot as plt
from pathlib import Path

base_dir = Path.cwd()
```

```
housetasks = pd.read_excel(base_dir/"housetasks.xlsx")

# Nombre d'observation et nombre de variable
nb_observations = housetasks.shape[0]
nb_variables = housetasks.shape[1]

# Tableau de contiengence
row_labels = housetasks.iloc[:, 0].values
housetasks_tab = housetasks.iloc[:, 1:]
col_labels = housetasks_tab.columns.values
data = housetasks_tab.values.astype(float)

# Fonction de performance de l'analyse des correspondance
def perform_ca(data, row_labels, col_labels):
    # Grand total
    total = np.sum(data)

    # Proportions
    P = data / total

    # poids lignes et colonnes
    r = np.sum(P, axis=1)
    c = np.sum(P, axis=0)

    expected = np.outer(r, c)

    # Dr and Dc diagonals
    Dr_inv_sqrt = np.diag(1 / np.sqrt(r))
    Dc_inv_sqrt = np.diag(1 / np.sqrt(c))

    # Matricxe stantdardiser des residus
```

```
    S = Dr_inv_sqrt @ (P - expected) @ Dc_inv_sqrt

    # SVD
    U, sigma, Vt = svd(S, full_matrices=False)

    # Eigenvalues
    eigenvalues = sigma ** 2
    explained_inertia = (eigenvalues / np.sum(eigenvalues)) * 100

    # Principal coordinates
    row_coords = Dr_inv_sqrt @ U @ np.diag(sigma)
    col_coords = Dc_inv_sqrt @ Vt.T @ np.diag(sigma)

    if row_coords[0, 1] > 0:
        row_coords[:, 1] *= -1
        col_coords[:, 1] *= -1

    # Contributions (absolute)
    row_contrib = (r[:, np.newaxis] * row_coords ** 2) / eigenvalues[np.newaxis, :]
    col_contrib = (c[:, np.newaxis] * col_coords ** 2) / eigenvalues[np.newaxis, :]

    # Cos2 (quality of representation)
    row_cos2 = (row_coords ** 2) / np.sum(row_coords ** 2, axis=1)[:, np.newaxis]
    col_cos2 = (col_coords ** 2) / np.sum(col_coords ** 2, axis=1)[:, np.newaxis]

    return {
        'eigenvalues': eigenvalues,
        'explained_inertia': explained_inertia,
        'row_coords': row_coords,
        'col_coords': col_coords,
        'row_contrib': row_contrib,
        'col_contrib': col_contrib,
        'row_cos2': row_cos2,
        'col_cos2': col_cos2
    }

# Execute CA
ca_results = perform_ca(data, row_labels, col_labels)

# Affichage du resumer de l'analyse
print("\n Valeurs propre:")
```

Valeurs propre:

```
for i, (ev, perc) in enumerate(zip(ca_results['eigenvalues'],
ca_results['explained_inertia'])):
    print(f"Dim {i+1}: {ev:.4f} ({perc:.2f}%)")
```

```
Dim 1: 0.1901 (72.10%)
Dim 2: 0.0568 (21.55%)
Dim 3: 0.0167 (6.35%)
Dim 4: 0.0000 (0.00%)
```

```
print("\n Coordinates Ligne:")
```

Coordinates Ligne:

```
row_coord_df = pd.DataFrame(ca_results['row_coords'], index=row_labels)
print(row_coord_df)
```

```
                 0         1         2             3
Cooking   -0.601633 -0.255626 -0.066363 -1.060532e-16
Laundry   -0.256893 -0.038241  0.048436  7.991367e-17
Cleaning  -0.121441 -0.018457  0.179632 -1.041003e-16
Shopping   0.099382  0.390093  0.056747 -6.113531e-17
Ironing   -0.275454  0.125110 -0.001317  2.025177e-17
Finances   0.271998  0.012410 -0.323463 -6.154413e-17
Repairs    0.262322  0.207396 -0.160136 -5.145460e-17
Driving    0.447334 -0.031322 -0.022152  3.658624e-17
Gardening  0.830564 -0.463838  0.202466  4.165422e-18
Homework   0.949061 -0.398335 -0.076315  1.176088e-16
Dishes     0.828652 -0.200084  0.072977 -2.021032e-16
```

```
print("\nCos2 Ligne:")
```

Cos2 Ligne:

```
row_cos2_df = pd.DataFrame(ca_results['row_cos2'], index=row_labels)
print(row_cos2_df)
```

```
                 0         1         2             3
Cooking    0.838437  0.151362  0.010201  2.605277e-32
Laundry    0.945441  0.020950  0.033609  9.148948e-32
Cleaning   0.311424  0.007193  0.681383  2.288367e-31
Shopping   0.059761  0.920754  0.019485  2.261478e-32
Ironing    0.828970  0.171011  0.000019  4.480923e-33
Finances   0.413855  0.000862  0.585283  2.118796e-32
Repairs    0.500569  0.312891  0.186540  1.925937e-32
Driving    0.992699  0.004867  0.002434  6.640322e-33
Gardening  0.729234  0.227433  0.043333  1.834165e-35
Homework   0.845575  0.148957  0.005467  1.298504e-32
Dishes     0.938036  0.054689  0.007275  5.579833e-32
```

```
print("\nContributions Ligne:")
```

```
Contributions Ligne:
```

```
row_contrib_df = pd.DataFrame(ca_results['row_contrib'], index=row_labels)
print(row_contrib_df)
```

```
                 0         1         2         3
Cooking    0.316380  0.191075  0.043739  0.229977
Laundry    0.058667  0.004349  0.023697  0.132807
Cleaning   0.008936  0.000690  0.222142  0.153599
Shopping   0.008878  0.457610  0.032890  0.078594
Ironing    0.029768  0.020544  0.000008  0.003764
Finances   0.027557  0.000192  0.442805  0.033003
Repairs    0.028023  0.058600  0.118657  0.025222
Driving    0.032795  0.000538  0.000914  0.005132
Gardening  0.126760  0.132256  0.085587  0.000075
Homework   0.161036  0.094903  0.011831  0.057850
Dishes     0.201200  0.039243  0.017731  0.279977
```

```
print("\nCoordinates Colonnes:")
```

Coordinates Colonnes:

```
col_coord_df = pd.DataFrame(ca_results['col_coords'], index=col_labels)
print(col_coord_df)
```

```
                    0         1         2             3
Wife        -0.356288 -0.074121  0.007828 -9.015502e-17
Alternating  0.388267  0.117926 -0.253685 -9.015502e-17
Husband      0.862540 -0.428228  0.123538 -9.015502e-17
Jointly      0.261418  0.449538  0.162703 -9.015502e-17
```

```
print("\nCos2 colonnes:")
```

Cos2 colonnes:

```
col_cos2_df = pd.DataFrame(ca_results['col_cos2'], index=col_labels)
print(col_cos2_df)
```

```
                    0         1         2             3
Wife         0.958073  0.041465  0.000462  6.134459e-32
Alternating  0.658262  0.060723  0.281015  3.549101e-32
Husband      0.789266  0.194543  0.016191  8.622725e-33
Jointly      0.230180  0.680657  0.089164  2.737632e-32
```

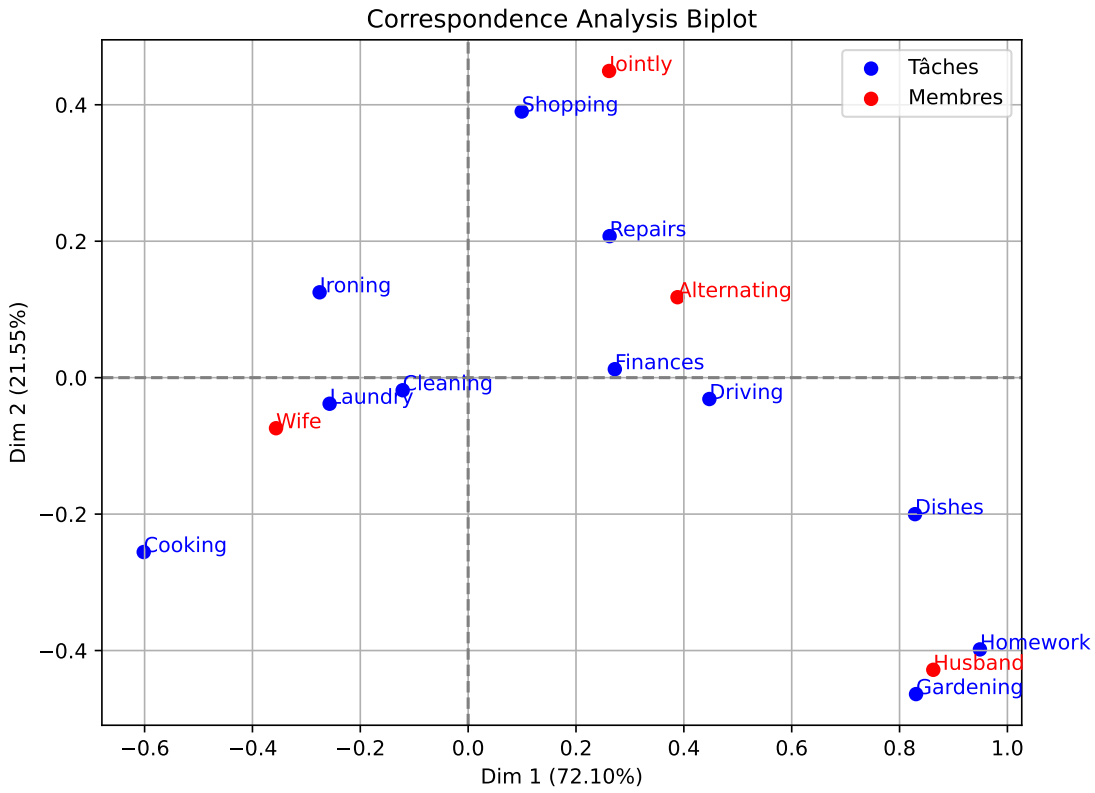
```
print("\nContributions Column:")
```

Contributions Column:

```
col_contrib_df = pd.DataFrame(ca_results['col_contrib'], index=col_labels)
print(col_contrib_df)
```

```
                    0         1         2         3
Wife         0.377625  0.054676  0.002071  0.565628
Alternating  0.134761  0.041588  0.653679  0.169972
Husband      0.432292  0.356466  0.100760  0.110482
Jointly      0.055321  0.547270  0.243490  0.153919
```

```python
# Plot the results (biplot for dimensions 1 and 2)
dim1 = 0
dim2 = 1

plt.figure(figsize=(8, 6))
plt.scatter(ca_results['row_coords'][:, dim1],
ca_results['row_coords'][:, dim2], color='blue', label='Tâches')
for i, label in enumerate(row_labels):
    plt.text(ca_results['row_coords'][i, dim1],
    ca_results['row_coords'][i, dim2], label, color='blue')

plt.scatter(ca_results['col_coords'][:, dim1],
ca_results['col_coords'][:, dim2], color='red', label='Membres')
for i, label in enumerate(col_labels):
    plt.text(ca_results['col_coords'][i, dim1],
    ca_results['col_coords'][i, dim2], label, color='red')

plt.axhline(0, color='gray', linestyle='--')
plt.axvline(0, color='gray', linestyle='--')
plt.xlabel(f"Dim {dim1+1} ({ca_results['explained_inertia'][dim1]:.2f}%)")
plt.ylabel(f"Dim {dim2+1} ({ca_results['explained_inertia'][dim2]:.2f}%)")
plt.title("Correspondence Analysis Biplot")
plt.legend()
plt.grid()
plt.show()
```



En conclusion, les tâches telles que **la cuisine, la lessive, le repassage, et le menage** sont majoritairement associées au mode **Wife**. À l'inverse, les tâches comme **le jardinage, les aides aux devoirs, la vaisselle** sont principalement liées au mode **Husband**. Les tâches telles que **shopping, réparations** se rapprochent davantage du mode **Jointly**, montrant que la **coopération au sein du couple est dominante** pour ces activités. Et enfin le mode **Alternating** pour les tâches comme **les réparations, la gestion des finances, et les courses**.

0.3 Exercice 3 : *Base Adult (Census Income)*

La base Adult provient d'un recensement socio-économique et décrit des individus à l'aide de variables telles que : le sexe, le niveau d'éducation, la situation matrimoniale, la profession, le niveau de revenu. L'objectif est d'identifier des profils d'individus et d'analyser les relations entre les différentes modalités des variables.

1. Description des variables et leur nature.

Sexe : variable qualitative nominale (modalités : Male, Female).

Éducation : variable qualitative ordinale (modalités ordonnées par niveau : HS-grad < Bachelors < Masters < PhD).

Situation matrimoniale : variable qualitative nominale (Married, Single, Divorced).

Profession : variable qualitative nominale (Tech, Service, Exec).

Revenu : variable qualitative binaire (<=50K, >50K), souvent traitée comme la variable cible.

2. Choix de méthode multivariée approprié et justification.

   Une analyse des correspondances multiples (ACM) est adaptée, car toutes les variables sont qualitatives.

3.
```r
# Importation de la base Wine
library(readxl)
adult_clean <- read_excel("adult_clean.xlsx")
View(adult_clean)

# Affichage du nombre d'observations et de variables
nb_observations <- nrow(adult_clean)
nb_variables <- ncol(adult_clean)

cat("Nombre d'observations :", nb_observations, "\n")
```

```
Nombre d'observations : 8
```

```r
cat("Nombre de variables :", nb_variables, "\n")
```

```
Nombre de variables : 5
```

0.3.1 ACM sous R

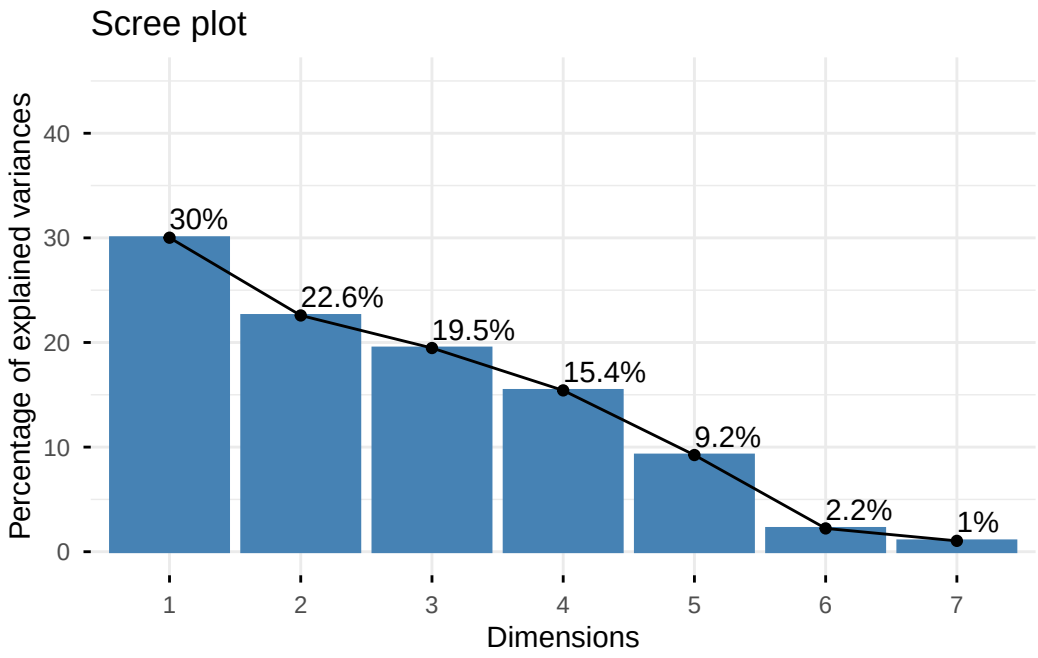
```r
library(FactoMineR)
library(factoextra)

#the MCA is performed only on the active
#'individuals/variables :
res.mca <- MCA(adult_clean, graph = FALSE)
print(res.mca)
```

```
**Results of the Multiple Correspondence Analysis (MCA)**
The analysis was performed on 8 individuals, described by 5 variables
*The results are available in the following objects:

   name              description
1  "$eig"            "eigenvalues"
2  "$var"            "results for the variables"
3  "$var$coord"      "coord. of the categories"
4  "$var$cos2"       "cos2 for the categories"
5  "$var$contrib"    "contributions of the categories"
6  "$var$v.test"     "v-test for the categories"
7  "$var$eta2"       "coord. of variables"
8  "$ind"            "results for the individuals"
9  "$ind$coord"      "coord. for the individuals"
10 "$ind$cos2"       "cos2 for the individuals"
11 "$ind$contrib"    "contributions of the individuals"
12 "$call"           "intermediate results"
13 "$call$marge.col" "weights of columns"
14 "$call$marge.li"  "weights of rows"
```

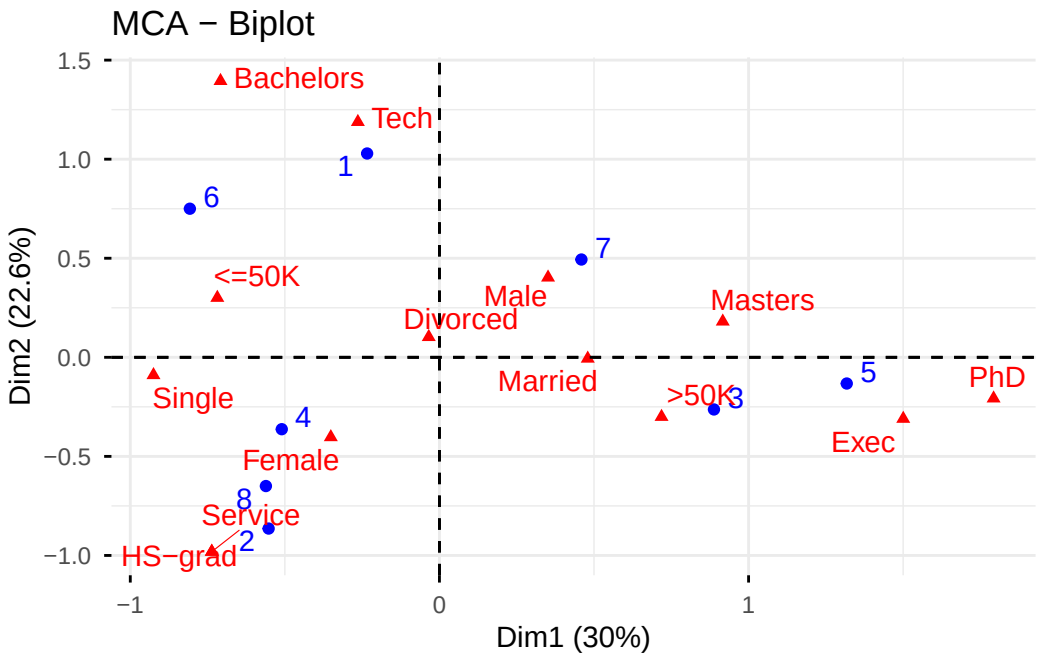
```r
#'Eigenvalues / Variances
#'The proportion of variances retained by the different dimensions (axes)
#'can be extracted using the function get_eigenvalue() [factoextra package]
#'as follow:
eig.val <- get_eigenvalue(res.mca)
head(eig.val)
```

```
      eigenvalue variance.percent cumulative.variance.percent
Dim.1 0.54028394        30.015774                    30.01577
Dim.2 0.40651835        22.584353                    52.60013
Dim.3 0.35047865        19.471036                    72.07116
Dim.4 0.27755515        15.419730                    87.49089
Dim.5 0.16634916         9.241620                    96.73251
Dim.6 0.04010754         2.228197                    98.96071
```

```r
#'To visualize the percentages of inertia explained by each
#'MCA dimensions, use the function fviz_eig() or fviz_screeplot()
fviz_screeplot(res.mca, addlabels = TRUE, ylim = c(0, 45))
```




```
#'Biplot
#'The function fviz_mca_biplot() [factoextra package] is used to draw
#'the biplot of individuals and variable categories:
fviz_mca_biplot(res.mca,
                repel = TRUE, # Avoid text overlapping (slow if many point)
                ggtheme = theme_minimal())
```




```r
#'Graph of variables
#'Results: The function get_mca_var() [in factoextra] is used
#'to extract the results for variable categories.
#'This function returns a list containing the coordinates,
#'the cos2 and the contribution of variable categories:

var <- get_mca_var(res.mca)
var
```

```
Multiple Correspondence Analysis Results for variables
 ===================================================
  Name       Description
1 "$coord"   "Coordinates for categories"
2 "$cos2"    "Cos2 for categories"
3 "$contrib" "contributions of categories"
```

```r
summary(res.mca)
```

```
Call:
MCA(X = adult_clean, graph = FALSE)
```

```
Eigenvalues
                       Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
Variance               0.540   0.407   0.350   0.278   0.166   0.040   0.019
% of var.             30.016  22.584  19.471  15.420   9.242   2.228   1.039
Cumulative % of var.  30.016  52.600  72.071  87.491  96.733  98.961 100.000

Individuals
             Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr   cos2
1          | -0.234  1.265  0.036 |  1.029 32.545  0.690 |  0.096  0.329  0.006
2          | -0.553  7.063  0.183 | -0.864 22.971  0.448 |  0.312  3.483  0.059
3          |  0.888 18.244  0.438 | -0.263  2.131  0.038 |  0.422  6.351  0.099
4          | -0.510  6.018  0.156 | -0.363  4.048  0.079 | -1.004 35.940  0.605
5          |  1.318 40.165  0.668 | -0.132  0.538  0.007 |  0.296  3.118  0.034
6          | -0.807 15.076  0.337 |  0.750 17.295  0.291 |  0.696 17.274  0.251
7          |  0.459  4.880  0.109 |  0.494  7.492  0.126 | -0.959 32.800  0.476
8          | -0.561  7.289  0.249 | -0.650 12.981  0.333 |  0.141  0.705  0.016

1          |
2          |
3          |
4          |
5          |
6          |
7          |
8          |

Categories (the 10 first)
             Dim.1    ctr   cos2 v.test    Dim.2    ctr   cos2 v.test    Dim.3
Female     | -0.351  2.285  0.123 -0.930 | -0.403  3.991  0.162 -1.066 |  0.663
Male       |  0.351  2.285  0.123  0.930 |  0.403  3.991  0.162  1.066 | -0.663
Bachelors  | -0.708  4.641  0.167 -1.082 |  1.395 23.932  0.649  2.131 |  0.669
HS-grad    | -0.736  7.527  0.325 -1.509 | -0.981 17.764  0.578 -2.011 | -0.310
Masters    |  0.916  7.773  0.280  1.400 |  0.181  0.402  0.011  0.276 | -0.454
PhD        |  1.793 14.868  0.459  1.793 | -0.207  0.265  0.006 -0.207 |  0.499
Divorced   | -0.035  0.011  0.000 -0.053 |  0.103  0.129  0.004  0.157 | -1.658
Married    |  0.480  4.260  0.230  1.269 | -0.006  0.001  0.000 -0.017 |  0.403
Single     | -0.925  7.918  0.285 -1.413 | -0.090  0.099  0.003 -0.137 |  0.852
Exec       |  1.500 20.831  0.750  2.292 | -0.310  1.183  0.032 -0.474 |  0.606
              ctr   cos2 v.test
Female     12.558  0.440  1.755 |
Male       12.558  0.440 -1.755 |
Bachelors   6.384  0.149  1.022 |
```

```
HS-grad    2.058  0.058 -0.635 |
Masters    2.935  0.069 -0.693 |
PhD        1.779  0.036  0.499 |
Divorced  39.206  0.916 -2.532 |
Married    4.635  0.162  1.066 |
Single    10.349  0.242  1.301 |
Exec       5.241  0.122  0.926 |

Categorical variables (eta2)
                Dim.1 Dim.2 Dim.3
Sex           | 0.123 0.162 0.440 |
Education     | 0.940 0.861 0.231 |
Marital_status | 0.329 0.005 0.950 |
Occupation    | 0.792 0.914 0.131 |
Income        | 0.516 0.090 0.001 |
```

```r
#'The different components can be accessed as follow:
# Coordinates
print('\ncoordonnées variable\n')
```

```
[1] "\ncoordonnées variable\n"
```

```r
var$coord
```

```
               Dim 1        Dim 2       Dim 3      Dim 4       Dim 5
Female    -0.3513499 -0.402813934  0.66342712  0.4423430 -0.21838309
Male       0.3513499  0.402813934 -0.66342712 -0.4423430  0.21838309
Bachelors -0.7081745  1.394907207  0.66894214 -0.2197648  0.02104255
HS-grad   -0.7363786 -0.981241402 -0.31008869 -0.2305834 -0.01362251
Masters    0.9164736  0.180678124 -0.45353738  1.2553485 -0.58339886
PhD        1.7925378 -0.207446455  0.49945655 -1.3794171  1.16558016
Divorced  -0.0345076  0.102572172 -1.65775383  0.2763768  0.18228915
Married    0.4797310 -0.006448353  0.40302832 -0.4660863 -0.58211716
Single    -0.9249545 -0.089675466  0.85169718  0.6557957  0.98194516
Exec       1.5003278 -0.310148655  0.60612790 -0.1113982 -0.14837006
Service   -0.7363786 -0.981241402 -0.31008869 -0.2305834 -0.01362251
Tech      -0.2638399  1.188007172 -0.09399657  0.3048489  0.11253589
<=50K     -0.7184505  0.300424426 -0.03005863 -0.4849230 -0.31599039
>50K       0.7184505 -0.300424426  0.03005863  0.4849230  0.31599039
```

```r
# Cos2: quality on the factore map
print("Qualité Variable\n")
```

```
[1] "Qualité Variable\n"
```

```r
var$cos2
```

```
                Dim 1        Dim 2        Dim 3       Dim 4        Dim 5
Female     0.1234467417 1.622591e-01 0.4401355394 0.195667367 0.0476911732
Male       0.1234467417 1.622591e-01 0.4401355394 0.195667367 0.0476911732
Bachelors  0.1671703940 6.485887e-01 0.1491611971 0.016098862 0.0001475963
HS-grad    0.3253520891 5.777008e-01 0.0576929988 0.031901222 0.0001113437
Masters    0.2799745946 1.088153e-02 0.0685653847 0.525299928 0.1134514101
PhD        0.4590274188 6.147719e-03 0.0356366926 0.271827350 0.1940824448
Divorced   0.0003969248 3.507017e-03 0.9160492547 0.025461384 0.0110764447
Married    0.2301418586 4.158126e-05 0.1624318303 0.217236409 0.3388603837
Single     0.2851802482 2.680563e-03 0.2417960299 0.143356001 0.3214054345
Exec       0.7503278161 3.206406e-02 0.1224636776 0.004136519 0.0073378917
Service    0.3253520891 5.777008e-01 0.0576929988 0.031901222 0.0001113437
Tech       0.0417668936 8.468166e-01 0.0053012135 0.055759697 0.0075985957
<=50K      0.5161711314 9.025484e-02 0.0009035211 0.235150364 0.0998499261
>50K       0.5161711314 9.025484e-02 0.0009035211 0.235150364 0.0998499261
```

```r
# Contributions to the principal components
print('Contribution variable\n')
```

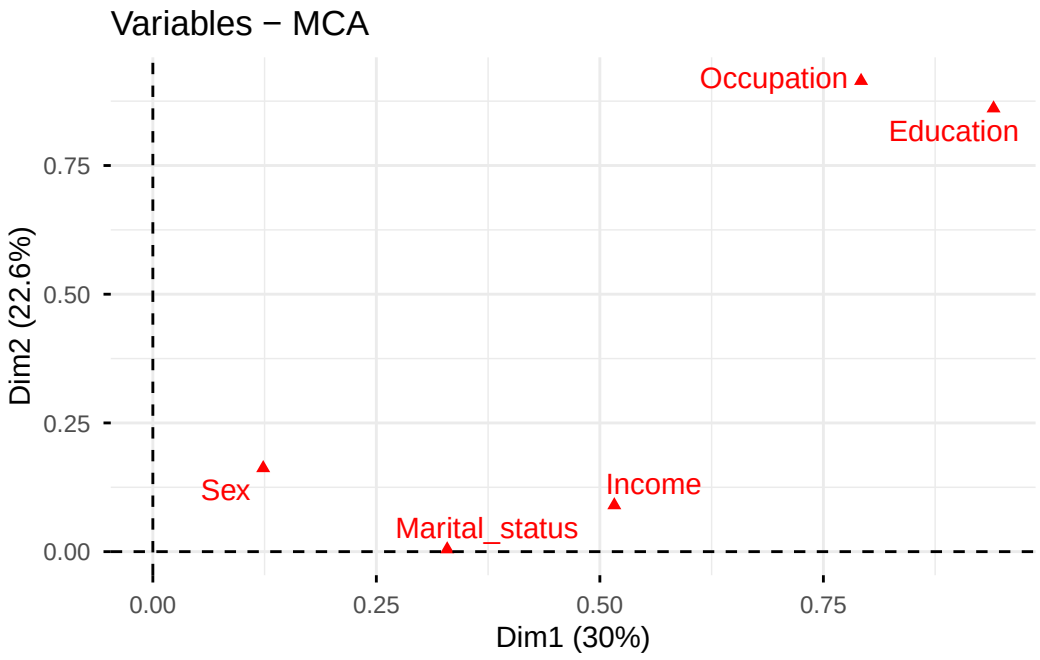
```
[1] "Contribution variable\n"
```

```r
var$contrib
```

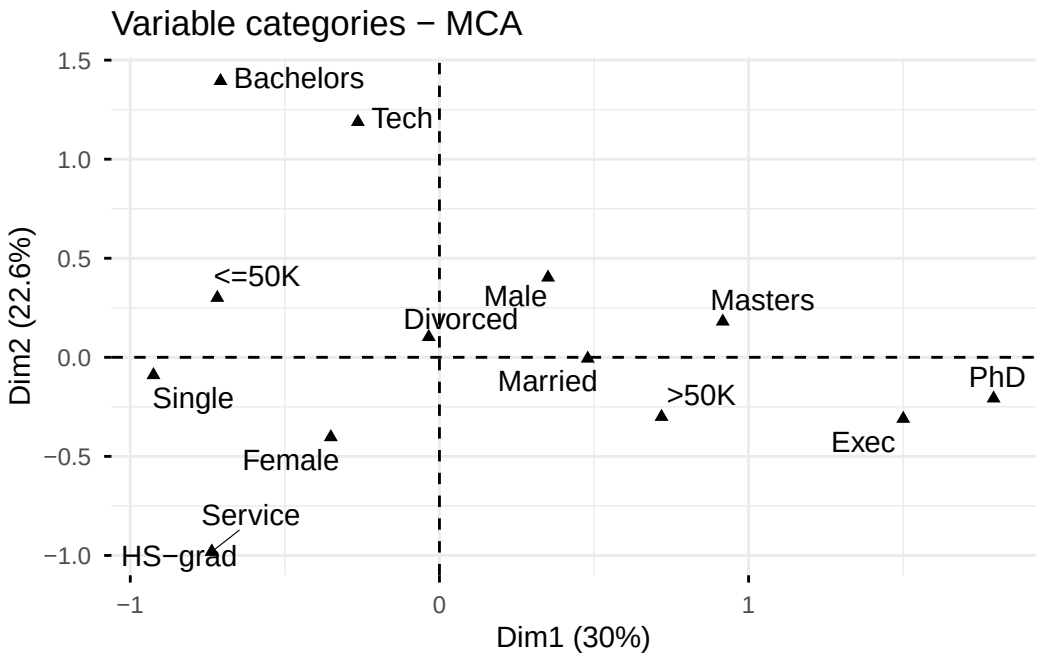
```
                Dim 1        Dim 2       Dim 3      Dim 4        Dim 5
Female      2.28484937  3.991432732 12.55812701  7.0496754  2.866931960
Male        2.28484937  3.991432732 12.55812701  7.0496754  2.866931960
Bachelors   4.64118166 23.932082112  6.38389232  0.8700358  0.013309023
HS-grad     7.52734038 17.763675692  2.05765026  1.4367065  0.008366719
Masters     7.77298493  0.401514275  2.93450335 28.3889492 10.230115498
PhD        14.86807074  0.264649993  1.77940117 17.1388594 20.417552585
Divorced    0.01101989  0.129404373 39.20563734  1.3760176  0.998782724
Married     4.25964649  0.001022863  4.63457134  7.8267837 20.370429132
```

```
Single      7.91751043  0.098909300 10.34853466  7.7474335 28.981699869
Exec       20.83148585  1.183122330  5.24127552  0.2235512  0.661670747
Service     7.52734038 17.763675692  2.05765026  1.4367065  0.008366719
Tech        0.96631814 26.038696038  0.18907049  2.5111990  0.570982413
<=50K       9.55370119  2.220190934  0.02577963  8.4722034  6.002430325
>50K        9.55370119  2.220190934  0.02577963  8.4722034  6.002430325
```

```r
#'Correlation between variables and principal dimensions
#'To visualize the correlation between variables
#'and MCA principal dimensions, type this:
fviz_mca_var(res.mca, choice = "mca.cor",
             repel = TRUE, # Avoid text overlapping (slow)
             ggtheme = theme_minimal())
```




```r
#'Use the function fviz_mca_var() [in factoextra]
#'to visualize only variable categories:
fviz_mca_var(res.mca, col.var="black",
             repel = TRUE)
```



Axe 1

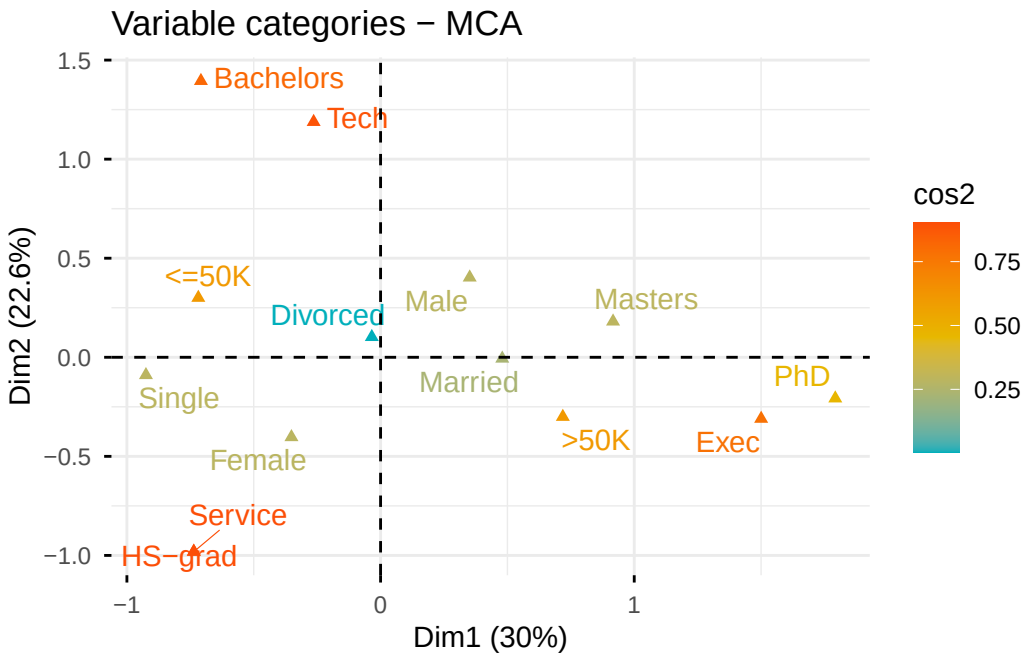
- Sens positif : Les catégories les plus proches sont >50K, PhD, Masters, Exec, male et Married. Cela signifie que les personnes de sexe masculin avec un doctorat ou un master, dans des postes d'exécutif, mariées, ont tendance à avoir un revenu supérieur à 50K.
- Sens négatif : Les catégories clés sont <=50K, divorced, female et Single. Cela signifie que les personnes de sexe féminin celibataires, divorcées , dans des métiers du service, on tendance à gagner moins de 50K.

Axe 2

- Sens positif : On y trouve male, tech, divorced et bachelors. Cela suggère une association entre être un homme ayant divorcé, avoir un bachelor et travailler dans le secteur technique.
- Sens négatif : Les catégories female, service et hs-grad sont regroupées ici, indiquant une concentration de femmes avec un diplôme d'études secondaires dans les emplois de service.

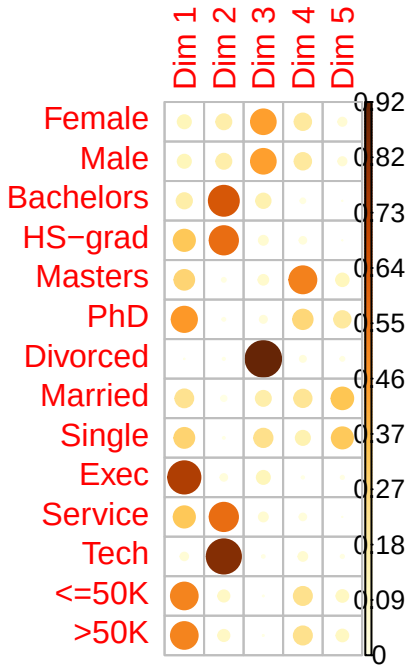
```
#'For instance, gradient.cols = c("white", "blue", "red") means that:
#'variable categories with low cos2 values will be colored in "white"
#'variable categories with mid cos2 values will be colored in "blue"
#'variable categories with high cos2 values will be colored in "red"
# Color by cos2 values: quality on the factor map
fviz_mca_var(res.mca, col.var = "cos2",
```

```
            gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
            repel = TRUE, # Avoid text overlapping
            ggtheme = theme_minimal())
```

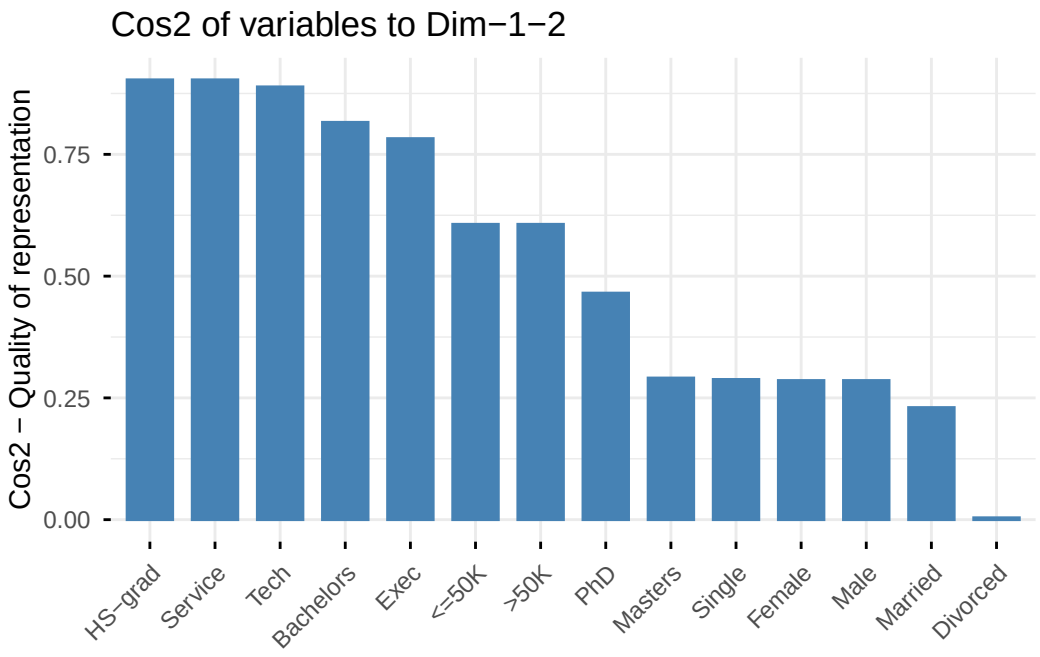


- Plusieurs catégories sont bien représentées sur les deux axes (cos2 $>$ 0.5), notamment PhD, Exec, HS-grad, Service, >50K et <=50K.
- D'autres, comme Divorced ou Single, ont un cos2 faible sur l'axe 1, ce qui signifie qu'elles sont mieux expliquées par les axes suivants.

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




```r
# Cos2 of variable categories on Dim.1 and Dim.2
fviz_cos2(res.mca, choice = "var", axes = 1:2)
```



0.3.2 ACM sous Python

```

import pandas as pd
import prince
import matplotlib.pyplot as plt
import seaborn as sns
from pathlib import Path

base_dir = Path.cwd()
```

```
# Chargement des données
adult_clean = pd.read_excel(base_dir / "adult_clean.xlsx")

# Vérifie que toutes les colonnes sont bien catégorielles
adult_clean = adult_clean.astype('category')

# Analyse des Correspondances Multiples (ACM / MCA)
mca = prince.MCA(
    n_components=6,          # nombre d'axes à calculer
    random_state=42
)

# Ajustement de l'ACM
mca = mca.fit(adult_clean)

# ---- Valeurs propres (inertie / variance) ----
eig = pd.DataFrame({
    'eigenvalue': mca.eigenvalues_,
    'variance_percent': mca.eigenvalues_ / mca.eigenvalues_.sum() * 100,
})
eig['cumulative_variance_percent'] = eig['variance_percent'].cumsum()
print("\nValeurs propres :")
```

```
Valeurs propres :
```

```
print(eig.round(3))
```

```
   eigenvalue  variance_percent  cumulative_variance_percent
```

```
0       0.540            30.331                    30.331
1       0.407            22.822                    53.153
2       0.350            19.676                    72.828
3       0.278            15.582                    88.410
4       0.166             9.339                    97.748
5       0.040             2.252                   100.000
```

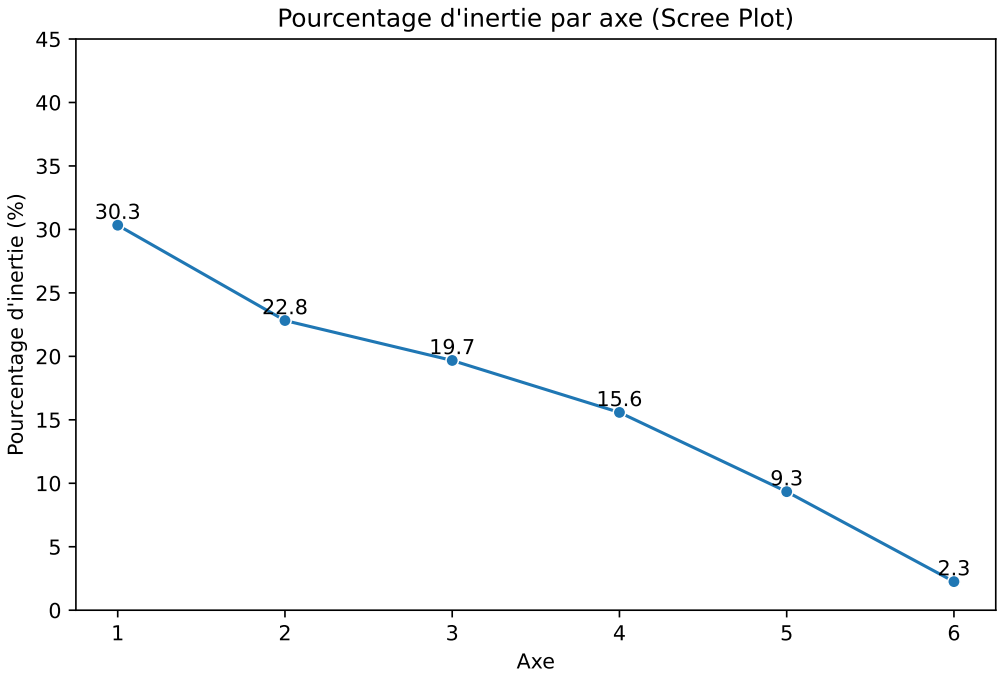
```
# Scree plot (inertie par axe)
plt.figure(figsize=(8, 5))
sns.lineplot(x=range(1, len(eig)+1), y=eig['variance_percent'], marker='o')
plt.title("Pourcentage d'inertie par axe (Scree Plot)")
plt.xlabel("Axe")
plt.ylabel("Pourcentage d'inertie (%)")
plt.ylim(0, 45)
```

```
(0.0, 45.0)
```

```
plt.xticks(range(1, len(eig)+1))
```

```
([<matplotlib.axis.XTick object at 0x00000142E7DA9400>, <matplotlib.axis.XTick object at 0x0
```

```
for i, txt in enumerate(eig['variance_percent']):
    plt.text(i+1, txt + 0.5, f"{txt:.1f}", ha='center')
plt.show()
```




```
# ----  Coordonnées des catégories ----
coords = mca.column_coordinates(adult_clean)
print("\nCoordonnées des catégories (10 premières) :")
```

```
Coordonnées des catégories (10 premières) :
```

```
print(coords.head(10).round(3))
```

```
                              0      1      2      3      4      5
Sex__Female              -0.351 -0.403 -0.663 -0.442  0.218  0.112
Sex__Male                 0.351  0.403  0.663  0.442 -0.218 -0.112
Education__Bachelors     -0.708  1.395 -0.669  0.220 -0.021  0.028
Education__HS-grad       -0.736 -0.981  0.310  0.231  0.014 -0.106
Education__Masters        0.916  0.181  0.454 -1.255  0.583  0.050
Education__PhD            1.793 -0.207 -0.499  1.379 -1.166  0.161
Marital_status__Divorced -0.035  0.103  1.658 -0.276 -0.182  0.358
Marital_status__Married   0.480 -0.006 -0.403  0.466  0.582 -0.226
Marital_status__Single   -0.925 -0.090 -0.852 -0.656 -0.982  0.093
Occupation__Exec          1.500 -0.310 -0.606  0.111  0.148  0.441
```

```python
# Cos2 (qualité de représentation)
cos2 = mca.column_cosine_similarities(adult_clean)
print("\nCos2 des catégories (10 premières) :")
```

```
Cos2 des catégories (10 premières) :
```

```python
print(cos2.head(10).round(3))
```

```
                              0      1      2      3      4      5
Sex__Female               0.123  0.162  0.440  0.196  0.048  0.012
Sex__Male                 0.123  0.162  0.440  0.196  0.048  0.012
Education__Bachelors      0.167  0.649  0.149  0.016  0.000  0.000
Education__HS-grad        0.325  0.578  0.058  0.032  0.000  0.007
Education__Masters        0.280  0.011  0.069  0.525  0.113  0.001
Education__PhD            0.459  0.006  0.036  0.272  0.194  0.004
Marital_status__Divorced  0.000  0.004  0.916  0.025  0.011  0.043
Marital_status__Married   0.230  0.000  0.162  0.217  0.339  0.051
Marital_status__Single    0.285  0.003  0.242  0.143  0.321  0.003
Occupation__Exec          0.750  0.032  0.122  0.004  0.007  0.065
```

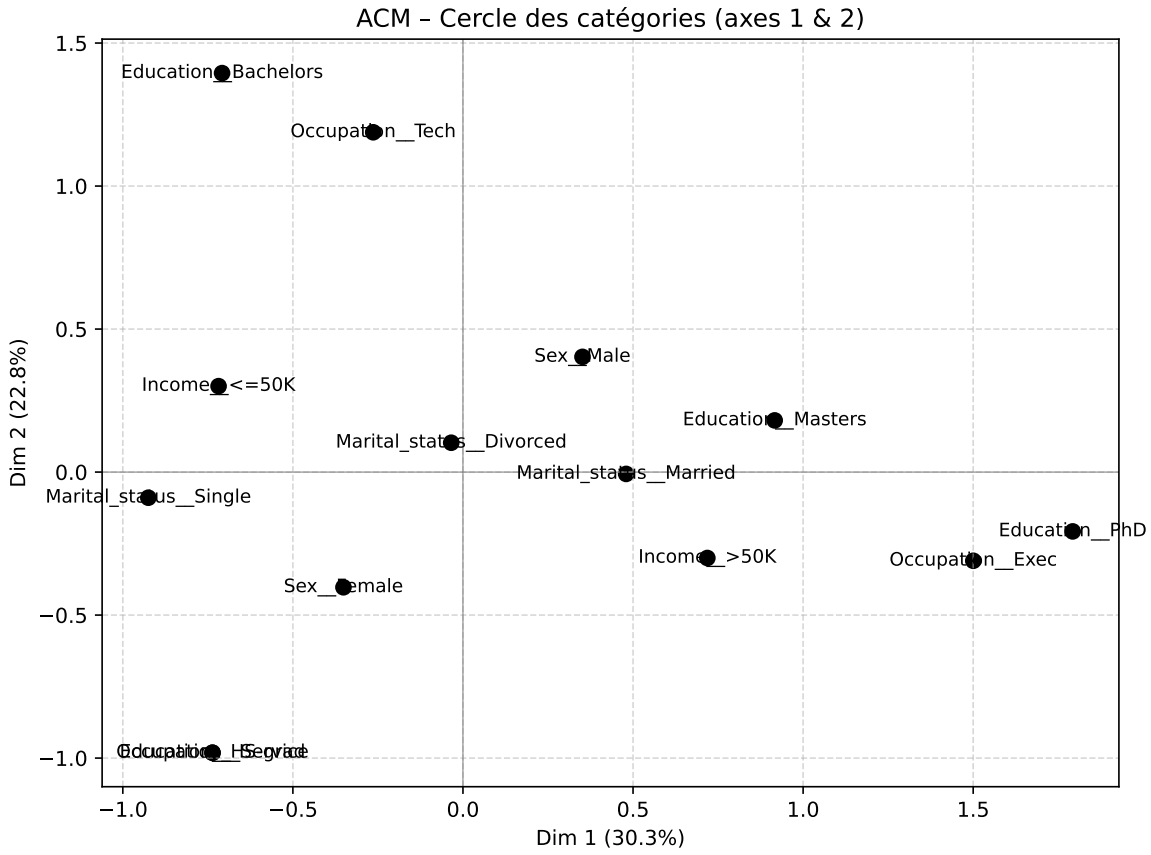
```python
# Contributions
contrib = mca.column_contributions_ * 100  # en %
print("\nContributions des catégories (en %) :")
```

```
Contributions des catégories (en %) :
```

```python
print(contrib.head(10).round(3))
```

```
                              0       1       2       3       4       5
Sex__Female               2.285   3.991  12.558   7.050   2.867   3.102
Sex__Male                 2.285   3.991  12.558   7.050   2.867   3.102
Education__Bachelors      4.641  23.932   6.384   0.870   0.013   0.095
Education__HS-grad        7.527  17.764   2.058   1.437   0.008   2.083
Education__Masters        7.773   0.402   2.935  28.389  10.230   0.315
Education__PhD           14.868   0.265   1.779  17.139  20.418   1.617
Marital_status__Divorced  0.011   0.129  39.206   1.376   0.999  15.975
Marital_status__Married   4.260   0.001   4.635   7.827  20.370  12.683
Marital_status__Single    7.918   0.099  10.349   7.747  28.982   1.081
Occupation__Exec         20.831   1.183   5.241   0.224   0.662  24.241
```

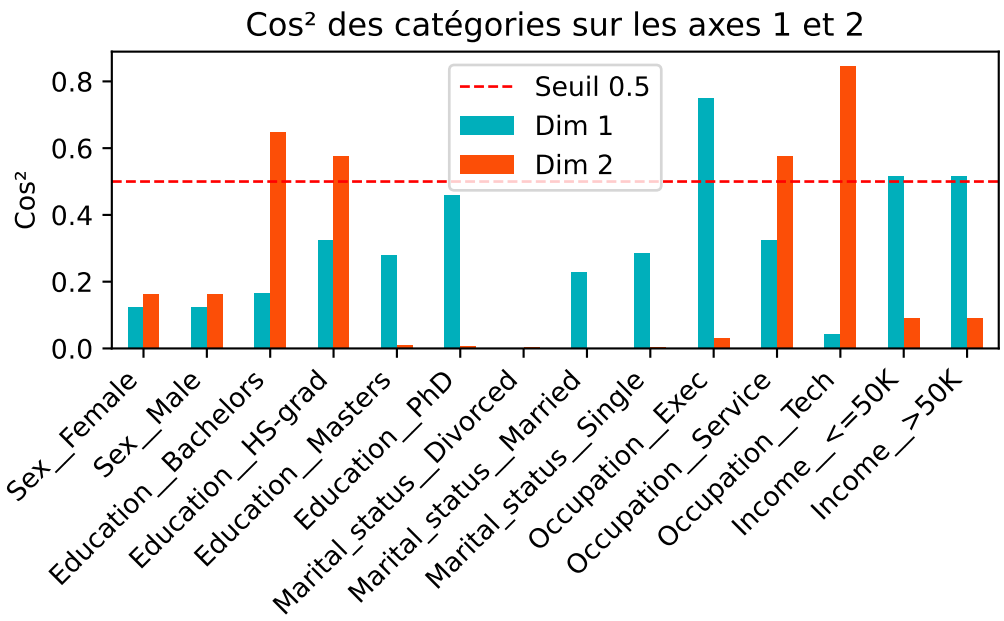
```python
# ----  Visualisations ----

# Cercle des catégories (axes 1 et 2)
coords_cat = mca.column_coordinates(adult_clean)

plt.figure(figsize=(8, 6))
ax = plt.gca()

# Tracer les points des catégories
for i, (cat, row) in enumerate(coords_cat.iterrows()):
    x, y = row[0], row[1]
    ax.scatter(x, y, color='black', s=50)
    ax.text(x, y, cat, fontsize=9, ha='center', va='center')

# Ajouter les axes et la grille
ax.axhline(0, color='grey', linewidth=0.5)
ax.axvline(0, color='grey', linewidth=0.5)
ax.grid(True, linestyle='--', alpha=0.5)

# Titre et labels
ax.set_xlabel(f"Dim 1 ({eig.loc[0, 'variance_percent']:.1f}%)")
ax.set_ylabel(f"Dim 2 ({eig.loc[1, 'variance_percent']:.1f}%)")
ax.set_title("ACM - Cercle des catégories (axes 1 & 2)")
plt.tight_layout()
plt.show()
```




```python
# Qualité de représentation (cos2) sur les deux premiers axes
plt.figure(figsize=(8, 6))
cos2_toplot = cos2[[0, 1]].copy()
cos2_toplot.columns = ['Dim 1', 'Dim 2']
cos2_toplot.plot(kind='bar', stacked=False, color=['#00AFBB', '#FC4E07'])
plt.title("Cos² des catégories sur les axes 1 et 2")
plt.ylabel("Cos²")
plt.axhline(0.5, color='red', linestyle='--', linewidth=1, label='Seuil 0.5')
plt.legend()
plt.xticks(rotation=45, ha='right')
```

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13]), [Text(0, 0, 'Sex__Female')
```

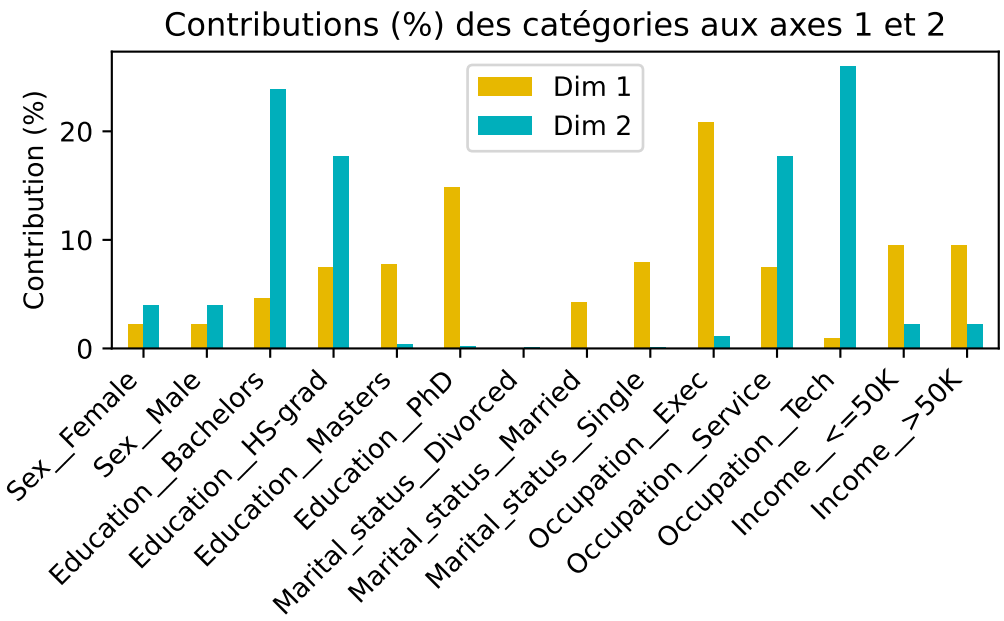
```python
plt.tight_layout()
plt.show()
```


Cos² des catégories sur les axes 1 et 2
Seuil 0.5
Dim 1
Dim 2
Cos²
0.0
0.2
0.4
0.6
0.8
Sex_Female
Sex_Male
Education_Bachelors
Education_HS-grad
Education_Masters
Education_PhD
Marital_status_Divorced
Marital_status_Married
Marital_status_Single
Occupation_Exec
Occupation_Service
Occupation_Tech
Income_<=50K
Income_>50K

```
#  Contributions des catégories aux axes 1 et 2
plt.figure(figsize=(8, 6))
contrib_toplot = contrib[[0, 1]].copy()
contrib_toplot.columns = ['Dim 1', 'Dim 2']
contrib_toplot.plot(kind='bar', stacked=False, color=['#E7B800', '#00AFBB'])
plt.title("Contributions (%) des catégories aux axes 1 et 2")
plt.ylabel("Contribution (%)")
plt.xticks(rotation=45, ha='right')
```

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13]), [Text(0, 0, 'Sex__Female')
```

```
plt.tight_layout()
plt.show()
```


Contributions (%) des catégories aux axes 1 et 2
Dim 1
Dim 2
Contribution (%)
0
10
20
Sex__Female
Sex__Male
Education__Bachelors
Education__HS-grad
Education__Masters
Education__PhD
Education__Divorced
Marital_status__Married
Marital_status__Single
Occupation__Exec
Occupation__Service
Occupation__Tech
Income__<=50K
Income__>50K

En conclusion, le revenu est fortement associé au niveau d'éducation, à la profession aux sexe et à la situation matrimoniale : les individus de sexe masculin avec un diplôme élevé (Master, PhD), exerçant une profession de type exécutive (Exec) et mariés ont une forte probabilité de gagner plus de 50K. À l'inverse, ceux qui sont de sexe feminin avec un diplôme secondaire (HS-grad), travaillant dans le secteur du service (Service) et célibataires sont plus souvent dans la catégorie de revenu inférieur (50K). En outre, des différences de répartition selon le sexe apparaissent clairement : les hommes sont plus présents dans les métiers techniques (Tech) avec un niveau de Bachelor, tandis que les femmes sont plus concentrées dans les emplois de service, souvent avec un niveau d'éducation plus modeste.

0.4 Exercice 4 : *Base Bank Marketing*

La base Bank Marketing provient d'une campagne marketing menée par une banque. Elle décrit des clients à l'aide des variables présentes dans la base. L'objectif est de comprendre la structure globale de la clientèle.

1. Choix de méthode multivarié et justification.

La méthode multivariée la plus appropriée pour analyser cette base est l'analyse factorielle des données mixtes (AFDM),

La base contient à la fois des variables quantitatives (comme *age* et *balance*) et des variables qualitatives (comme *job*, *marital*, *education*, *housing*, *loan*). L'ACP classique ne convient qu'aux variables quantitatives, tandis que l'analyse des correspondances multiples (ACM) s'applique uniquement aux qualitatives. La AFMD permet de traiter simultanément les deux types de variables.

2.

```r
#' Packages
library(PCAmixdata)
library(ggplot2)
library(factoextra)
# Importation de la base bank
library(readxl)
bank <- read_excel("bank.xlsx")
View(bank)

# Affichage du nombre d'observations et de variables
nb_observations <- nrow(bank)
nb_variables <- ncol(bank)

cat("Nombre d'observations :", nb_observations, "\n")
```

```
Nombre d'observations : 8
```

```r
cat("Nombre de variables :", nb_variables, "\n")
```

```
Nombre de variables : 7
```

0.4.1 Présentation et description de la base bank

La base **Bank Marketing** décrit un échantillon de clients à l'aide de **7 variables** :

- **2 variables quantitatives** : *age* (âge en années) et *balance* (solde bancaire).
- **5 variables qualitatives** : *job* (catégorie professionnelle), *marital* (état civil), *education* (niveau d'études), *housing* (possède un prêt immobilier : oui/non), *loan* (possède un prêt personnel : oui/non).

0.4.2 AFMD sous R

```r
# Convertir en data.frame standard
bank_df <- as.data.frame(bank)

#Ajout d'une colonne label client
bank_df$Label <- paste0("client_", 1:nrow(bank_df))

# Labels pour les individus (à utiliser dans les graphiques)
ind_labels <- bank_df$Label


#  the number quantitatives variables and the number of qualitatives
X.quanti <- bank_df[,c(1, 5)]
X.quali <- bank_df[,c(2, 3, 4, 6, 7)]
# Convertir les variables qualitatives en facteurs si besoin
X.quali <- as.data.frame(lapply(X.quali, as.factor))

#'Perform the analyses
m1 <- PCAmix(X.quanti,X.quali, graph=FALSE, rename.level = TRUE)
summary(m1)
```

```
Call:
PCAmix(X.quanti = X.quanti, X.quali = X.quali, rename.level = TRUE,     graph = FALSE)

Method = Factor Analysis of mixed data (FAmix)

Data:
   number of observations:  8
   number of  variables:  7
        number of numerical variables:  2
        number of categorical variables:  5

Squared loadings :
          dim 1 dim 2 dim 3 dim 4 dim 5
age        0.57  0.32  0.03  0.00  0.03
balance    0.71  0.11  0.03  0.06  0.07
job        0.97  0.84  0.99  0.83  0.98
marital    0.64  0.50  0.05  0.44  0.06
education  0.34  0.36  0.19  0.07  0.01
```

```
housing    0.50  0.11  0.29  0.07  0.00
loan       0.37  0.19  0.00  0.08  0.05
```

```r
#'Answers for The Eigenvalue. This answer allows to identify the Principal
# Components we will consider
m1$eig
```

```
      Eigenvalue Proportion Cumulative
dim 1  4.1020144  34.183454   34.18345
dim 2  2.4319737  20.266448   54.44990
dim 3  1.5923791  13.269825   67.71973
dim 4  1.5428815  12.857346   80.57707
dim 5  1.1962832   9.969027   90.54610
dim 6  0.7190934   5.992445   96.53854
dim 7  0.4153747   3.461456  100.00000
```

```r
#Results for quantitatives variables
m1$quanti
```

```
$coord
            dim 1     dim 2      dim 3       dim 4      dim 5
age     0.7575282 0.5623344  0.1807516 -0.03663649  0.1669880
balance 0.8447799 0.3363411 -0.1809565  0.23911528 -0.2738226

$contrib
           dim 1     dim 2      dim 3       dim 4      dim 5
age     0.573849 0.3162200 0.03267114 0.001342232 0.02788500
balance 0.713653 0.1131254 0.03274525 0.057176115 0.07497883

$contrib.pct
           dim 1     dim 2    dim 3      dim 4    dim 5
age     13.98944 13.002607 2.051719 0.08699516 2.330970
balance 17.39762  4.651587 2.056373 3.70580085 6.267649

$cos2
           dim 1     dim 2      dim 3       dim 4      dim 5
age     0.573849 0.3162200 0.03267114 0.001342232 0.02788500
balance 0.713653 0.1131254 0.03274525 0.057176115 0.07497883
```

```r
#Results for qualitatives  variables
m1$levels # for the modalities
```

```
$coord
                          dim 1       dim 2        dim 3       dim 4
job=admin           -0.9738810  0.57502892 -0.016603411  0.89631905
job=management       1.7688954  0.07352619 -0.490391777  1.13526738
job=retired          1.1324376  0.89551537  1.878156655 -0.73716677
job=services        -0.7137637  0.13301191 -0.076641722 -1.04493371
job=student         -0.1537355 -2.28221697  0.706556329  0.64715007
job=technician       0.6276920 -0.10290626 -1.907830942 -0.74802136
marital=divorced    -1.0459076  1.37400327 -0.137190890  1.65882304
marital=married      0.7912762  0.24784041 -0.189340985 -0.43459884
marital=single      -0.7063991 -0.78845497  0.298184943  0.02652411
education=secondary -0.4485704  0.46231941  0.338333278 -0.20687922
education=tertiary   0.7476173 -0.77053235 -0.563888797  0.34479870
housing=no           0.9158658 -0.43772514  0.698107069  0.34841689
housing=yes         -0.5495195  0.26263508 -0.418864241 -0.20905014
loan=no              0.3515858 -0.25246679  0.008862743 -0.15957171
loan=yes            -1.0547573  0.75740038 -0.026588229  0.47871512
                           dim 5
job=admin            0.458146367
job=management      -1.474021691
job=retired          0.869397774
job=services        -1.074851620
job=student          0.426878592
job=technician       1.411155831
marital=divorced     0.591010274
marital=married     -0.002827588
marital=single      -0.193233308
education=secondary -0.072802546
education=tertiary   0.121337577
housing=no          -0.059248442
housing=yes          0.035549065
loan=no              0.123475117
loan=yes            -0.370425351

$contrib
                           dim 1        dim 2        dim 3        dim 4
job=admin           0.237111073 0.0826645652 6.891831e-05 0.2008469598
job=management      0.391123865 0.0006757626 3.006051e-02 0.1611040027
job=retired         0.160301862 0.1002434727 4.409341e-01 0.0679268550
```

```
job=services        0.127364651 0.0044230422 1.468488e-03 0.2729716151
job=student         0.002954327 0.6510642885 6.240273e-02 0.0523504012
job=technician      0.049249660 0.0013237122 4.549774e-01 0.0699419941
marital=divorced    0.136740331 0.2359856237 2.352668e-03 0.3439617337
marital=married     0.313059004 0.0307124349 1.792500e-02 0.0944380766
marital=single      0.187124863 0.2331229666 3.334285e-02 0.0002638232
education=secondary 0.125759614 0.1335870217 7.154338e-02 0.0267493817
education=tertiary  0.209599356 0.2226450362 1.192390e-01 0.0445823028
housing=no          0.314553822 0.0718512356 1.827576e-01 0.0455228744
housing=yes         0.188732293 0.0431107413 1.096545e-01 0.0273137247
loan=no             0.092709422 0.0478046108 5.891116e-05 0.0190973471
loan=yes            0.278128266 0.1434138325 1.767335e-04 0.0572920412
                           dim 5
job=admin           5.247452e-02
job=management      2.715925e-01
job=retired         9.448156e-02
job=services        2.888265e-01
job=student         2.277817e-02
job=technician      2.489201e-01
marital=divorced    4.366164e-02
marital=married     3.997626e-06
marital=single      1.400217e-02
education=secondary 3.312632e-03
education=tertiary  5.521053e-03
housing=no          1.316392e-03
housing=yes         7.898350e-04
loan=no             1.143458e-02
loan=yes            3.430374e-02

$contrib.pct
                         dim 1       dim 2        dim 3       dim 4
job=admin           5.78035686  3.39907316  0.004328009 13.01765316
job=management      9.53492171  0.02778659  1.887773636 10.44176139
job=retired         3.90788148  4.12189789 27.690269632  4.40259708
job=services        3.10492937  0.18187048  0.092219775 17.69232560
job=student         0.07202137 26.77102504  3.918836457  3.39302804
job=technician      1.20062132  0.05442955 28.572177125  4.53320589
marital=divorced    3.33349219  9.70346117  0.147745447 22.29346441
marital=married     7.63183575  1.26286048  1.125674458  6.12088990
marital=single      4.56177974  9.58575196  2.093901423  0.01709938
education=secondary 3.06580135  5.49294681  4.492861128  1.73372887
education=tertiary  5.10966891  9.15491134  7.488101881  2.88954812
housing=no          7.66827682  2.95444131 11.477013268  2.95051013
```

```
housing=yes          4.60096609  1.77266478  6.886207961  1.77030608
loan=no              2.26009497  1.96567137  0.003699569  1.23777149
loan=yes             6.78028492  5.89701412  0.011098707  3.71331447
                            dim 5
job=admin           4.386463e+00
job=management      2.270303e+01
job=retired         7.897926e+00
job=services        2.414366e+01
job=student         1.904078e+00
job=technician      2.080779e+01
marital=divorced    3.649775e+00
marital=married     3.341705e-04
marital=single      1.170473e+00
education=secondary 2.769103e-01
education=tertiary  4.615172e-01
housing=no          1.100401e-01
housing=yes         6.602408e-02
loan=no             9.558421e-01
loan=yes            2.867526e+00

$cos2
                           dim 1        dim 2        dim 3       dim 4
job=admin           0.316148097 0.1102194203 9.189108e-05 0.267795946
job=management      0.446998702 0.0007723001 3.435487e-02 0.184118860
job=retired         0.183202128 0.1145639688 5.039246e-01 0.077630691
job=services        0.169819535 0.0058973895 1.957984e-03 0.363962153
job=student         0.003376374 0.7440734725 7.131741e-02 0.059829030
job=technician      0.056285325 0.0015128140 5.199741e-01 0.079933708
marital=divorced    0.156274664 0.2696978557 2.688763e-03 0.393099124
marital=married     0.626118008 0.0614248697 3.585001e-02 0.188876153
marital=single      0.299399781 0.3729967466 5.334856e-02 0.000422117
education=secondary 0.335358970 0.3562320579 1.907823e-01 0.071331684
education=tertiary  0.335358970 0.3562320579 1.907823e-01 0.071331684
housing=no          0.503286115 0.1149619769 2.924121e-01 0.072836599
housing=yes         0.503286115 0.1149619769 2.924121e-01 0.072836599
loan=no             0.370837688 0.1912184433 2.356446e-04 0.076389388
loan=yes            0.370837688 0.1912184433 2.356446e-04 0.076389388
                           dim 5
job=admin           6.996603e-02
job=management      3.103914e-01
job=retired         1.079789e-01
job=services        3.851020e-01
job=student         2.603219e-02
```

```
job=technician      2.844801e-01
marital=divorced    4.989902e-02
marital=married     7.995251e-06
marital=single      2.240347e-02
education=secondary 8.833685e-03
education=tertiary  8.833685e-03
housing=no          2.106227e-03
housing=yes         2.106227e-03
loan=no             4.573831e-02
loan=yes            4.573831e-02
```

```
m1$quali  # for the variables
```

```
$contrib
               dim 1     dim 2        dim 3      dim 4       dim 5
job       0.9681054 0.8403948 0.9899120651 0.82514183 0.979073343
marital   0.6369242 0.4998210 0.0536205194 0.43866363 0.057667807
education 0.3353590 0.3562321 0.1907823450 0.07133168 0.008833685
housing   0.5032861 0.1149620 0.2924120879 0.07283660 0.002106227
loan      0.3708377 0.1912184 0.0002356446 0.07638939 0.045738314

$contrib.pct
             dim 1     dim 2       dim 3     dim 4      dim 5
job       23.60073 34.556083 62.16560463 53.480571 81.8429396
marital   15.52711 20.552074  3.36732133 28.431454  4.8205815
education  8.17547 14.647858 11.98096301  4.623277  0.7384275
housing   12.26924  4.727106 18.36322123  4.720816  0.1760642
loan       9.04038  7.862685  0.01479828  4.951086  3.8233684
```

```
#Results for individuals
m1$ind
```

```
$coord
       dim 1      dim 2      dim 3      dim 4      dim 5
1 -1.8265648 -0.3492381  0.1312170  0.1662154  0.3557768
2  1.2712917 -0.1604801 -2.4074824 -0.9291381  1.5434479
3  3.5826200  0.1146625 -0.6188229  1.4101472 -1.6122073
4 -2.1541693  0.2195707  0.1060174 -0.8712195 -1.4567194
5 -0.3113672 -3.5590687  0.8915999  0.8038431  0.4668973
6  2.2935746  1.3965371  2.3700365 -0.9156553  0.9509015
7 -2.1183216  2.1427288 -0.1731205  2.0604702  0.6464159
```

```
8 -0.7370633  0.1952878 -0.2994450 -1.7246629 -0.8945128

$contrib
       dim 1       dim 2       dim 3       dim 4      dim 5
1 0.41704236 0.015245908 0.002152237 0.003453445 0.01582214
2 0.20202281 0.003219233 0.724496421 0.107912207 0.29777893
3 1.60439574 0.001643437 0.047867729 0.248564382 0.32490154
4 0.58005567 0.006026410 0.001404961 0.094877927 0.26525392
5 0.01211869 1.583371227 0.099368801 0.080770464 0.02724914
6 0.65756055 0.243789489 0.702134148 0.104803087 0.11302670
7 0.56091081 0.573910831 0.003746339 0.530692189 0.05223169
8 0.06790780 0.004767166 0.011208414 0.371807779 0.10001914

$contrib.pct
       dim 1       dim 2       dim 3      dim 4     dim 5
1 10.1667697  0.62689443  0.13515858  0.2238309  1.322608
2  4.9249660  0.13237122 45.49773629  6.9941994 24.892010
3 39.1123865  0.06757626  3.00605119 16.1104003 27.159249
4 14.1407515  0.24779913  0.08823031  6.1493982 22.173171
5  0.2954327 65.10642885  6.24027308  5.2350401  2.277817
6 16.0301862 10.02434727 44.09340528  6.7926855  9.448156
7 13.6740331 23.59856237  0.23526676 34.3961734  4.366164
8  1.6554744  0.19602046  0.70387852 24.0982722  8.360825

$cos2
        dim 1        dim 2       dim 3       dim 4      dim 5
1 0.432108278 0.0157966757 0.002229988 0.003578203 0.01639373
2 0.148129300 0.0023604403 0.531222911 0.079124527 0.21834061
3 0.715791783 0.0007332098 0.021355908 0.110895547 0.14495292
4 0.489351494 0.0050840510 0.001185265 0.080041723 0.22377577
5 0.006532797 0.8535445169 0.053566526 0.043540760 0.01468913
6 0.359926224 0.1334420534 0.384324291 0.057365636 0.06186696
7 0.314653432 0.3219460374 0.002101579 0.297701730 0.02930035
8 0.087101634 0.0061145842 0.014376422 0.476897589 0.12828910
```

Axe 1

- Sens positives : Ce sont les clients âgés, avec un solde bancaire élevé, souvent cadres ou retraités, mariés, sans prêt personnel (loan=no) et sans prêt immobilier (housing=no). Leur niveau d'études est souvent tertiaire.
- Sens négatives : Ce sont les clients plus jeunes, avec un solde bancaire faible, souvent étudiants ou dans des emplois de services, célibataires, et qui ont tendance à avoir un

prêt personnel (loan=yes).

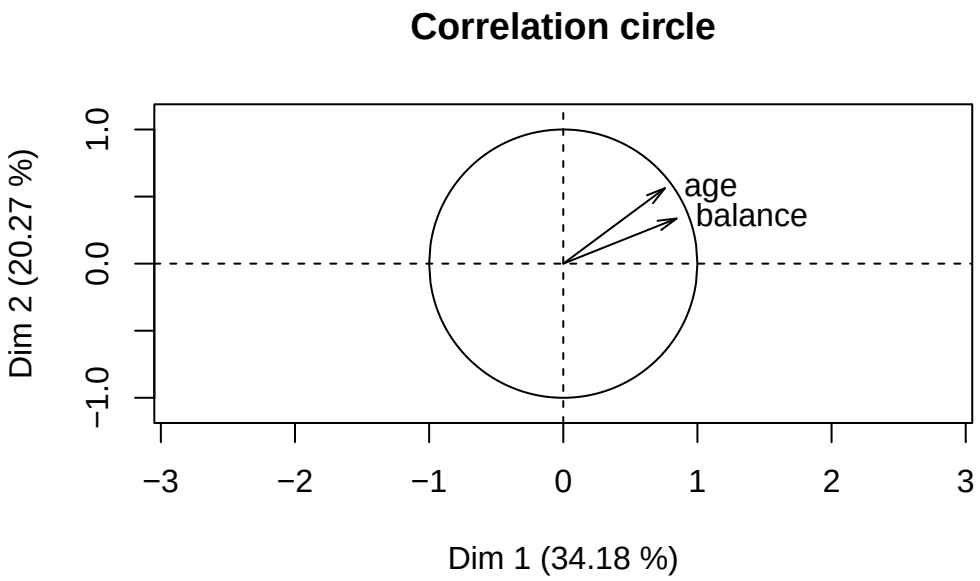
Le métier (job) est la variable la plus importante sur cet axe, suivi par l'âge et le solde.

Axe 2

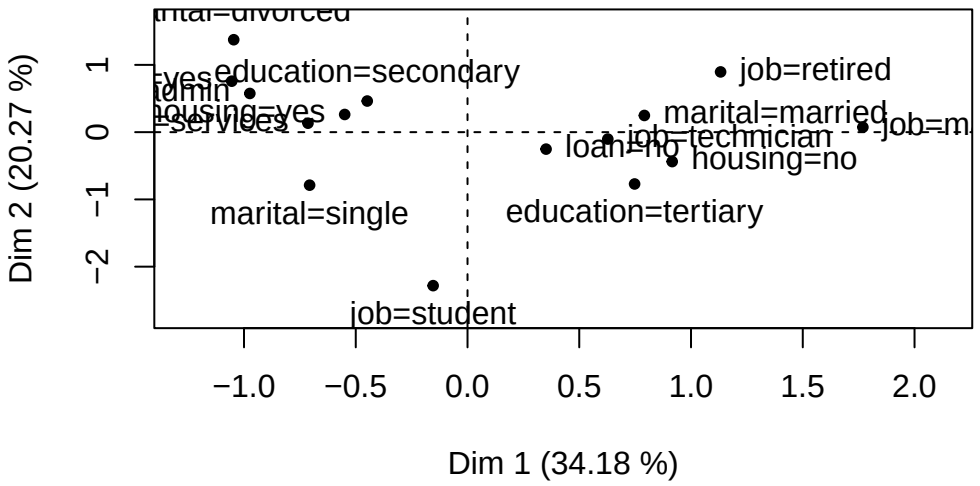
- Sens positives : Ce sont les clients divorcés ou parfois marié, qui ont souvent un prêt personnel (loan=yes) et/ou un prêt immobilier (housing=yes).
- Sens négatives : Ce sont les clients célibataire, sans prêts personnels (loan=no), et souvent avec un solde bancaire faible. On y trouve aussi les étudiants, qui sont jeunes et ont un solde faible.

  La situation matrimoniale (marital) et la possession d'un prêt personnel (loan) sont les variables les plus importantes sur cet axe.

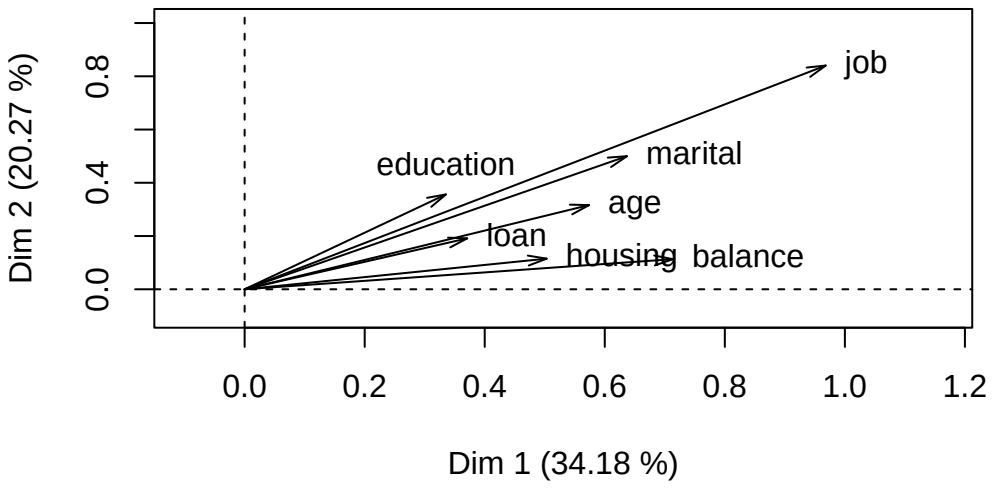
```
# Ajouter les labels aux résultats
rownames(m1$ind$coord) <- ind_labels
rownames(m1$ind$contrib) <- ind_labels
rownames(m1$ind$cos2) <- ind_labels

#' Visualisation the results
#The correlation circle for the quantitatives variables
plot(m1, axes = c(1, 2), choice = "cor", label = TRUE)
```



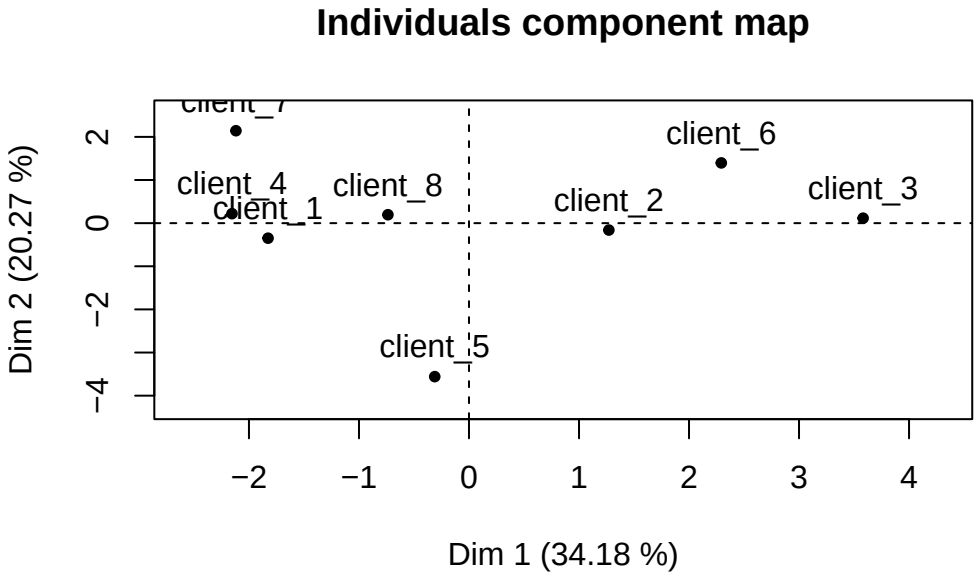

```
#The graph for modalities of qualitatives variables
plot(m1, axes = c(1, 2), choice = "levels", label = TRUE)
```




```r
#The graph of squared loadings for mixtes variables
plot(m1, axes = c(1, 2), choice = "sqload", label = TRUE)
```




```r
#The graph of individuals
plot(m1, axes = c(1, 2), choice = "ind", label = TRUE)
```



0.4.3 AFMD sous Python

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Circle
from pathlib import Path

base_dir = Path.cwd()
```

```
# Importer la base de données bank
bank = pd.read_excel(base_dir / "bank.xlsx")

# Nombre d'observations et de variables
nb_observations = bank.shape[0]
nb_variables = bank.shape[1]

# Ajout d'une colonne Label pour identifier les clients
bank['Label'] = [f"client_{i+1}" for i in range(nb_observations)]

# Variables quantitatives
X_quanti = bank.iloc[:, [0, 4]]
```

```

# Variables qualitatives
X_quali = bank.iloc[:, [1, 2, 3, 5, 6]]

# Labels des individus pour les graphiques
ind_labels = bank['Label']

# Standardisation des variables quantitatives
Z1 = (X_quanti - X_quanti.mean()) / X_quanti.std(ddof=1)

# Création de la matrice indicatrice pour les variables qualitatives
G = pd.get_dummies(X_quali, prefix_sep='.', dtype=float)

# Centrage de la matrice indicatrice
Z2 = G - G.mean()

# Concaténation pour former la matrice Z
Z = pd.concat([Z1, Z2], axis=1)

# Dimensions
n = Z.shape[0]      # nombre d'individus
p1 = Z1.shape[1]    # nombre de variables quantitatives
m = Z2.shape[1]     # nombre de modalités
p2 = X_quali.shape[1]  # nombre de variables qualitatives

# Inertie totale (formule de l'AFM)
total_inertia = p1 + m - p2

# Poids des colonnes (matrice diagonale M)
M_diag_quanti = np.ones(p1)         # poids 1 pour les quanti
p_levels = G.mean().values          # proportions des modalités
M_diag_quali = 1 / p_levels         # poids 1/p_j pour les modalités
M_diag = np.concatenate((M_diag_quanti, M_diag_quali))
M = np.diag(M_diag)

# Préparation pour la décomposition GSVD
sqrt_M = np.diag(np.sqrt(M_diag))
sqrt_M_inv = np.diag(1 / np.sqrt(M_diag))

# Matrice L = (1/√n) * Z * √M
L = np.sqrt(1 / n) * Z.values @ sqrt_M

```

```python
# Décomposition en valeurs singulières de L
P, S, Vt = np.linalg.svd(L, full_matrices=False)
Q = Vt.T

# Composantes de la GSVD
U = np.sqrt(n) * P          # coordonnées des lignes (individus)
V = sqrt_M_inv @ Q          # coordonnées des colonnes normalisées

# Valeurs propres (carrés des valeurs singulières)
eig = S ** 2

# Pourcentage et cumul de variance expliquée
percentage = 100 * eig / total_inertia
cum_percentage = np.cumsum(percentage)

# Affichage des valeurs propres
eig_df = pd.DataFrame({
    'valeur propre': eig,
    'pourcentage de variance': percentage,
    'pourcentage cumulé': cum_percentage
})
print("Valeurs propres :")
```

```
Valeurs propres :
```

```python
print(eig_df)
```

```
   valeur propre  pourcentage de variance  pourcentage cumulé
0   3.946516e+00             3.288763e+01           32.887630
1   2.373576e+00             1.977980e+01           52.667432
2   1.585160e+00             1.320967e+01           65.877102
3   1.534819e+00             1.279016e+01           78.667261
4   1.182929e+00             9.857744e+00           88.525006
5   7.153894e-01             5.961579e+00           94.486584
6   4.116099e-01             3.430082e+00           97.916667
7   2.595191e-32             2.162659e-31           97.916667
```

```python
# Coordonnées des individus (F)
F = Z.values @ M @ V

```

```
# Ajout des labels
ind_coord = pd.DataFrame(F, index=ind_labels,
      columns=[f"Dim.{i+1}" for i in range(len(S))])

# Résultats pour les individus
print("\nRésultats pour les individus (coordonnées) :")
```

```
Résultats pour les individus (coordonnées) :
```

```
print(ind_coord.head())
```

```
             Dim.1     Dim.2     Dim.3  ...     Dim.6     Dim.7         Dim.8
Label                                   ...
client_1 -1.758889 -0.414172 -0.151115  ... -1.946958 -0.534230  7.546047e-17
client_2  1.270966 -0.081766  2.539593  ...  0.376184 -0.296665 -6.427150e-16
client_3  3.466147  0.254005  0.397427  ... -0.337509 -0.089604  5.212844e-16
client_4 -2.150341  0.105680 -0.026487  ...  0.966836 -0.954331  9.974660e-17
client_5 -0.053944 -3.541437 -0.943380  ...  0.571885  0.297455 -1.170938e-16

[5 rows x 8 columns]
```

```
# Résultats pour les variables quantitatives
V_quanti = V[:p1, :]
A_quanti = V_quanti @ np.diag(S)  # coordonnées (corrélations)
quanti_coord = pd.DataFrame(A_quanti, index=X_quanti.columns,
columns=[f"Dim.{i+1}" for i in range(len(S))])
print("\nRésultats pour les variables quantitatives :")
```

```
Résultats pour les variables quantitatives :
```

```
print(quanti_coord)
```

```
            Dim.1     Dim.2     Dim.3  ...     Dim.6     Dim.7         Dim.8
age      0.677645  0.562363 -0.163717  ...  0.158621 -0.147847 -4.804083e-17
balance  0.767505  0.356860  0.132770  ... -0.043233 -0.080940  1.203905e-16

[2 rows x 8 columns]
```

```python
# Résultats pour les modalités (niveaux des variables qualitatives)
V_levels = V[p1:, :]
A_levels = V_levels @ np.diag(S)
Astar_levels = np.diag(M_diag_quali) @ A_levels  # coordonnées pondérées
#des modalités
levels_coord = pd.DataFrame(Astar_levels, index=G.columns,
columns=[f"Dim.{i+1}" for i in range(len(S))])
print("\nRésultats pour les modalités :")
```

```
Résultats pour les modalités :
```

```python
print(levels_coord)
```

```
                       Dim.1     Dim.2  ...     Dim.7         Dim.8
job.admin          -1.008285  0.514782  ... -0.062713  8.798939e-18
job.management      1.744778  0.164870  ... -0.139664 -2.522182e-16
job.retired         1.090375  0.950945  ... -0.267092  3.171888e-17
job.services       -0.715602  0.103186  ...  0.265475  1.188259e-17
job.student        -0.027154 -2.298677  ...  0.463637  1.015504e-17
job.technician      0.639775 -0.053073  ... -0.462405 -5.952488e-17
marital.divorced   -1.131186  1.298395  ...  0.707268 -4.098057e-18
marital.married     0.781539  0.300129  ...  0.287321 -8.915909e-18
marital.single     -0.664990 -0.832971  ... -0.618851  1.135013e-17
education.secondary -0.471480  0.437376  ...  0.027687 -1.038870e-17
education.tertiary  0.785799 -0.728960  ... -0.046144  2.765840e-17
housing.no          0.935999 -0.394288  ...  0.018960 -2.969904e-18
housing.yes        -0.561600  0.236573  ... -0.011376  9.820140e-18
loan.no             0.368936 -0.227832  ...  0.130038 -8.095730e-18
loan.yes           -1.106809  0.683495  ... -0.390114 -4.447496e-18

[15 rows x 8 columns]
```

```python
# Résultats pour les variables qualitatives (eta²)
quali_coord = np.zeros((p2, len(S)))
for k, var in enumerate(X_quali.columns):
    # Sélection des modalités de la variable
    levels_var = [col for col in G.columns if col.startswith(var + '.')]
    idx = [list(G.columns).index(col) for col in levels_var]
    a_star_var = Astar_levels[idx, :]
```

```
    m_diag_var = M_diag_quali[idx]
    eta2 = np.sum(m_diag_var[:, np.newaxis] * a_star_var ** 2, axis=0)
    # Signe basé sur la modalité la plus contributive
    max_idx = np.argmax(np.abs(a_star_var), axis=0)
    signs = np.sign(a_star_var[max_idx, np.arange(len(S))])
    quali_coord[k, :] = np.sqrt(eta2) * signs

quali_coord_df = pd.DataFrame(quali_coord, index=X_quali.columns,
columns=[f"Dim.{i+1}" for i in range(len(S))])
print("\nRésultats pour les variables qualitatives :")
```

```
Résultats pour les variables qualitatives :
```

```
print(quali_coord_df)
```

```
               Dim.1     Dim.2     Dim.3  ...     Dim.6     Dim.7         Dim.8
job         6.577280 -7.130798  7.929321  ... -3.190146  2.110597 -7.395985e-16
marital    -3.554925  3.939168 -0.610243  ...  1.801642  2.277761  2.523636e-17
education   1.415021 -1.312668  0.950474  ...  0.433298 -0.083094  4.703877e-17
housing     1.685492 -0.710010 -1.318785  ...  0.280836  0.034143  1.333481e-17
loan       -2.254238  1.392075 -0.078881  ...  1.805526 -0.794546 -1.290382e-17

[5 rows x 8 columns]
```

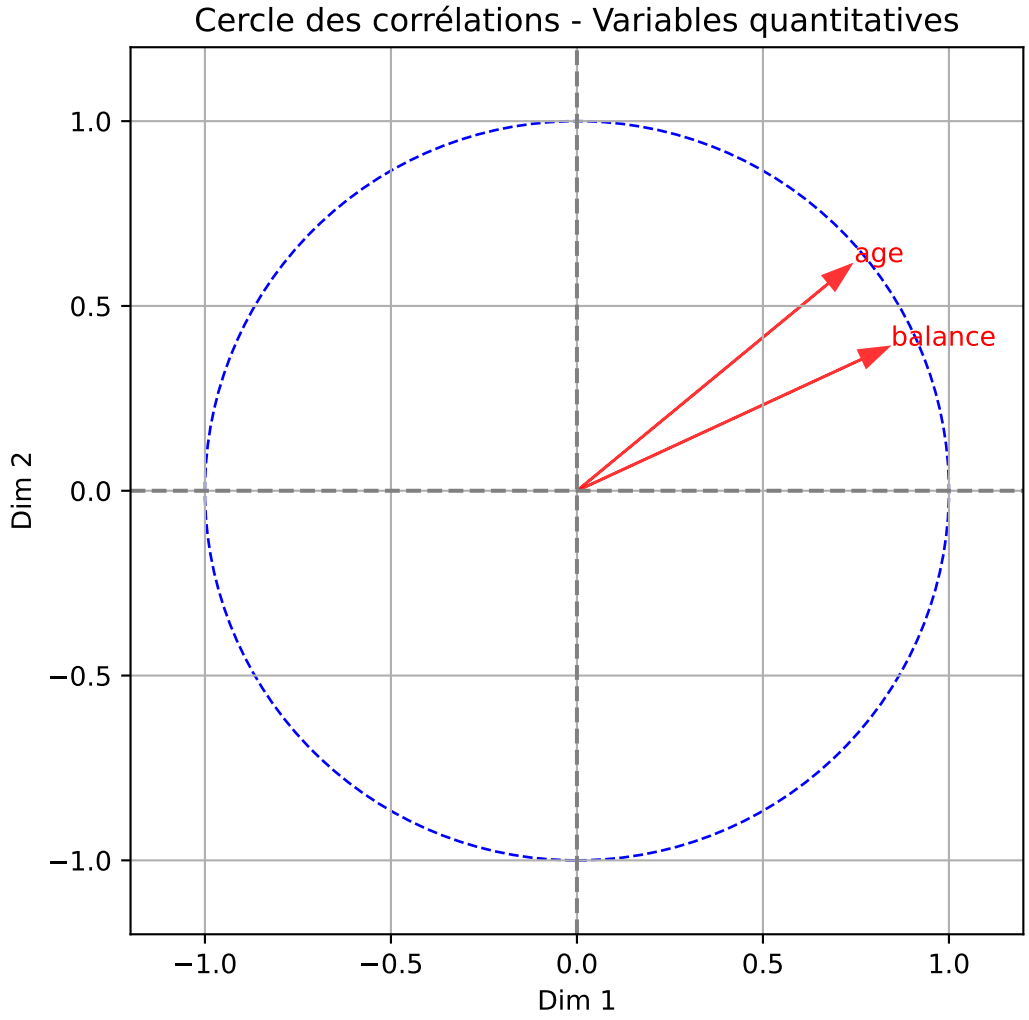
```
# Cercle des corrélations pour les variables quantitatives
fig, ax = plt.subplots(figsize=(6, 6))
for i in range(p1):
    x = A_quanti[i, 0]
    y = A_quanti[i, 1] if len(S) > 1 else 0
    ax.arrow(0, 0, x, y, head_width=0.05, color='red', alpha=0.8)
    ax.text(x * 1.1, y * 1.1, X_quanti.columns[i], color='red')

# Cercle unité
circle = Circle((0, 0), 1, color='blue', fill=False, linestyle='--')
ax.add_artist(circle)
ax.set_xlim(-1.2, 1.2)
```

```
(-1.2, 1.2)
```

```
ax.set_ylim(-1.2, 1.2)
```

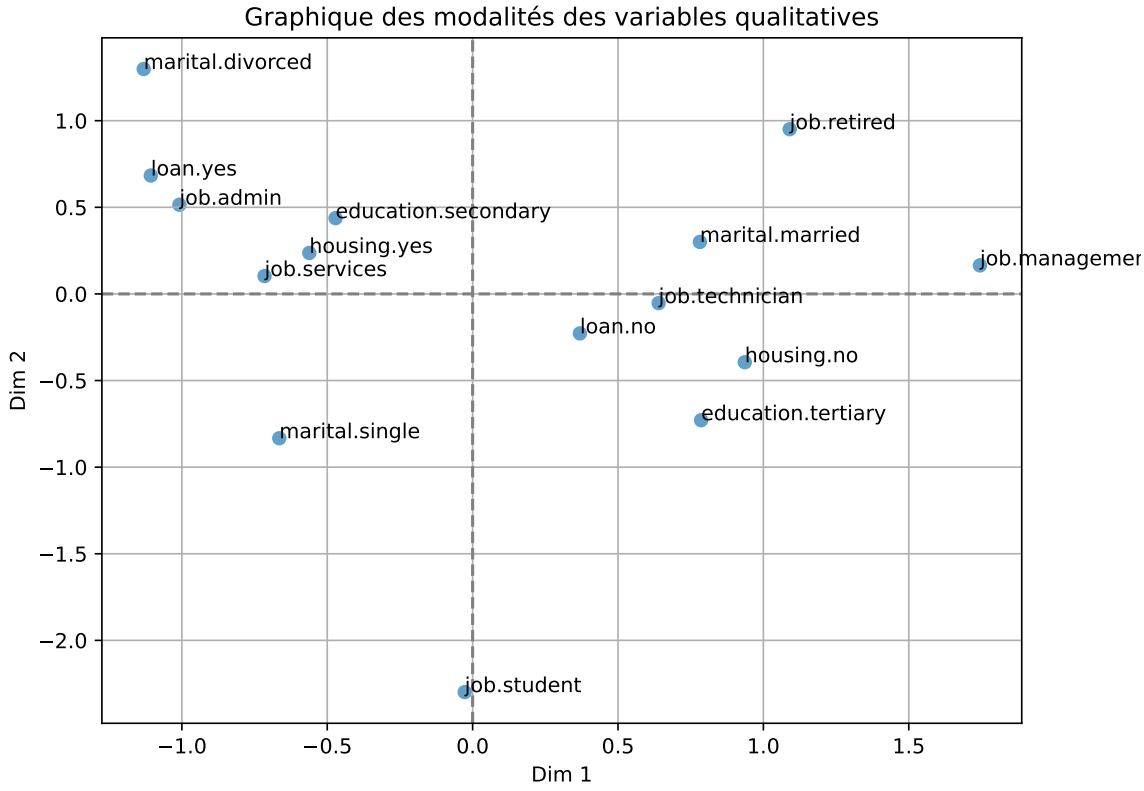
```
(-1.2, 1.2)
```

```
ax.axhline(0, color='gray', linestyle='--')
ax.axvline(0, color='gray', linestyle='--')
ax.set_title("Cercle des corrélations - Variables quantitatives")
ax.set_xlabel("Dim 1")
ax.set_ylabel("Dim 2")
ax.grid(True)
plt.show()
```




```python
# Graphique des modalités des variables qualitatives
fig, ax = plt.subplots(figsize=(8, 6))
x = Astar_levels[:, 0]
y = Astar_levels[:, 1] if len(S) > 1 else np.zeros(m)
ax.scatter(x, y, alpha=0.7)
for i, label in enumerate(G.columns):
    ax.text(x[i], y[i], label, fontsize=10)

ax.axhline(0, color='gray', linestyle='--')
ax.axvline(0, color='gray', linestyle='--')
```

```
ax.set_title("Graphique des modalités des variables qualitatives")
ax.set_xlabel("Dim 1")
ax.set_ylabel("Dim 2")
ax.grid(True)
plt.show()
```

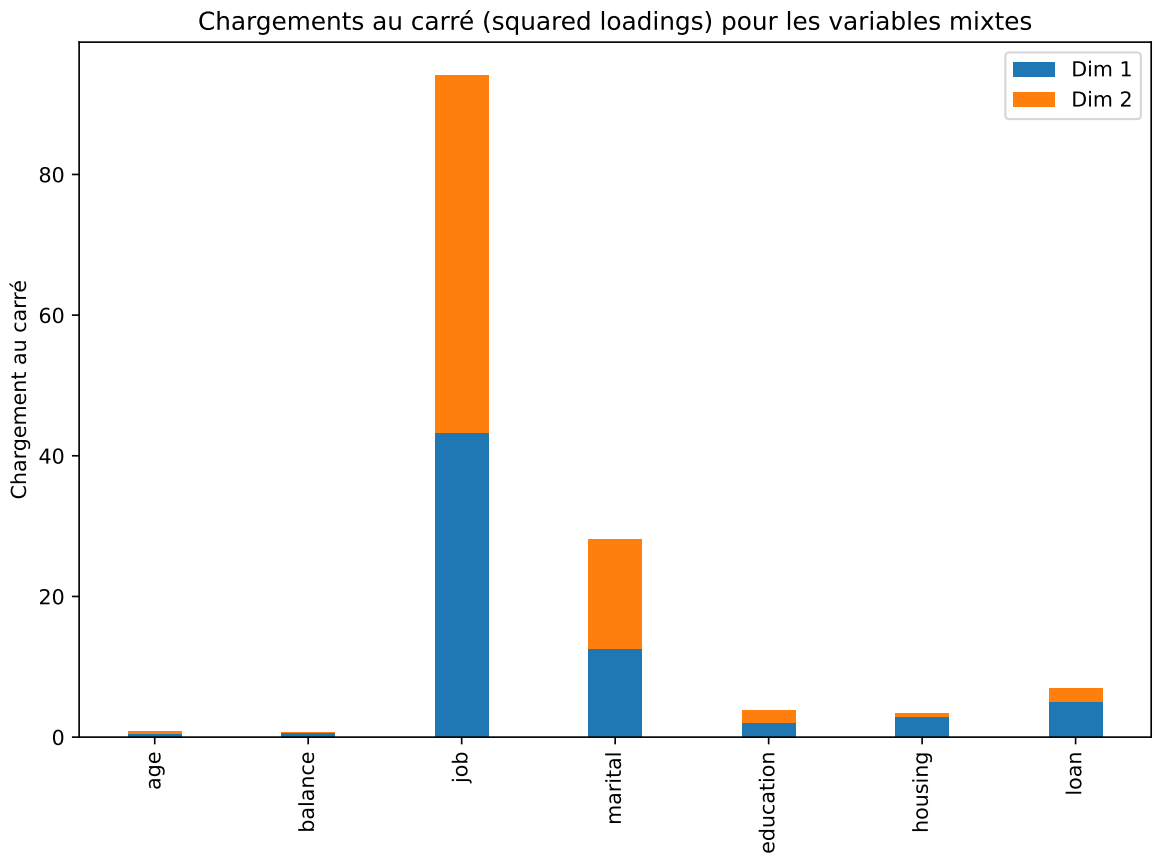



```
# Graphique des chargements au carré pour les variables mixtes
sqload = np.zeros((p1 + p2, len(S)))
sqload[:p1, :] = A_quanti ** 2  # pour les quantitatives
sqload_idx = p1
for k, var in enumerate(X_quali.columns):
    levels_var = [col for col in G.columns if col.startswith(var + '.')]
    idx = [list(G.columns).index(col) for col in levels_var]
    a_star_var = Astar_levels[idx, :]
    m_diag_var = M_diag_quali[idx]
    eta2 = np.sum(m_diag_var[:, np.newaxis] * a_star_var ** 2, axis=0)
```

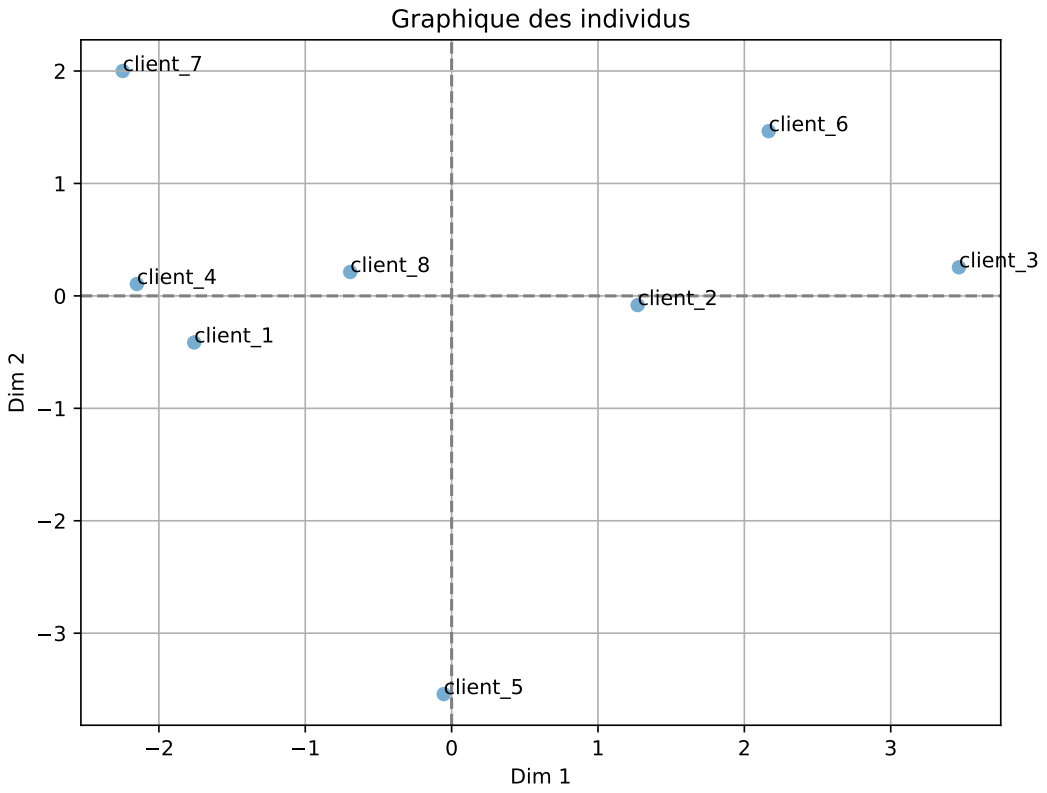
```
    sqload[sqload_idx, :] = eta2
    sqload_idx += 1

variables = list(X_quanti.columns) + list(X_quali.columns)
bar_width = 0.35
index = np.arange(len(variables))

plt.figure(figsize=(8, 6))
plt.bar(index, sqload[:, 0], bar_width, label='Dim 1')
if len(S) > 1:
    plt.bar(index, sqload[:, 1], bar_width, bottom=sqload[:, 0], label='Dim 2')
plt.xticks(index, variables, rotation=90)
```

```
([<matplotlib.axis.XTick object at 0x00000142E94EA8D0>, <matplotlib.axis.XTick object at 0x0
```

```
plt.title("Chargements au carré (squared loadings) pour les variables mixtes")
plt.ylabel("Chargement au carré")
plt.legend()
plt.tight_layout()
plt.show()
```




```python
# Graphique des individus
fig, ax = plt.subplots(figsize=(8, 6))
x = F[:, 0]
y = F[:, 1] if len(S) > 1 else np.zeros(n)
ax.scatter(x, y, alpha=0.6)
for i, label in enumerate(ind_labels):
    ax.text(x[i], y[i], label, fontsize=10)

ax.axhline(0, color='gray', linestyle='--')
ax.axvline(0, color='gray', linestyle='--')
ax.set_title("Graphique des individus")
ax.set_xlabel("Dim 1")
ax.set_ylabel("Dim 2")
ax.grid(True)
plt.show()
```



En résumé, le profil client « idéal » (stable et solvable) se caractérise par : marié, cadre ou retraité, âge mûr, solde élevé, sans prêt personnel. À l'inverse, le profil à risque ou en phase de constitution se traduit par : jeune, célibataire ou divorcé, étudiant ou en services, faible solde, souvent endetté.