Compte-rendu de cours

Quentin KALINKA | quentin.kalinka.etu@univ-lille.fr

M1 DSS

Matrice de corrélation : Analyse et Visualisation

Une **matrice de corrélation** est utilisée pour évaluer la **dépendence** entre plusieurs variables en méme temps. Le résultat est une table contenant les **coefficients de corrélation** entre chaque variable et les autres. Il existe différentes méthodes de **test de corrélation** : **le test de corrélation de Pearson**, la **corrélation de Kendall** et de **Spearman** qui sont des tests basés sur le rang. Ces méthodes sont discutées dans les sections suivantes. La **matrice de corrélation** peut étre visualisée en utilisant un ***corrélogramme***.

L'objectif est de montrer comment calculer et visualiser une **matrice de corrélation** dans **R**.

Packages

```
library(dplyr)
library(Hmisc)
library(corrplot)
```

Pour réaliser ce travail, plusieurs outils statistiques et de visualisation ont été mobilisés à l'aide des packages R suivants : *dplyr*, *Hmisc*, ainsi que *corrplot*.

Lecture des données

La première étape consiste à importer la table de données *mtcars* disponible dans **R** afin d'en observer la structure et d'évaluer leur conformité avec les attentes initiales.

```
mtcars <- read.csv("data/mtcars.csv", header = TRUE)
head(mtcars, 9)
```

```
##        manufacturer  mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 1         Mazda RX4 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## 2     Mazda RX4 Wag 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## 3        Datsun 710 22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## 4    Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## 5 Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## 6           Valiant 18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## 7        Duster 360 14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## 8         Merc 240D 24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## 9          Merc 230 22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
```

- **Format** : un fichier de données *"mtcars"* avec 32 observations sur 11 variables : *mpg* (Miles/(US) gallon), *cyl* (Nombre de cylindres), *disp* (Cylindrée), *hp* (Puissance brute), *drat* (Rapport de pont arrière), *wt* (Poids (1000 lbs)), *qsec* (Temps au 1/4 mille), *vs* (Moteur (0 = V-shaped, 1 = straight)), *am* (Transmission (0 = automatique, 1 = manuelle)), *gear* (Nombre de vitesses), *carb* (Nombre de carburateurs).
- **Description** : les données ont été extraites du magazine Motor Trend US de 1974 et comprennent la consommation de carburant et 10 aspects de la conception et des performances de 32 automobiles (modèles 1973-1974).

Analyse de corrélation dans R

La fonction **cor()** de **R** peut être utilisée pour calculer **la matrice de corrélation**.

```
numeric_mtcars <- mtcars %>% select(where(is.numeric))
correlation_matrix <- cor(numeric_mtcars, method = "pearson")
print(correlation_matrix)
```

```
##             mpg        cyl       disp         hp        drat         wt
## mpg   1.0000000 -0.8521620 -0.8475514 -0.7761684  0.68117191 -0.8676594
## cyl  -0.8521620  1.0000000  0.9020329  0.8324475 -0.69993811  0.7824958
## disp -0.8475514  0.9020329  1.0000000  0.7909486 -0.71021393  0.8879799
## hp   -0.7761684  0.8324475  0.7909486  1.0000000 -0.44875912  0.6587479
## drat  0.6811719 -0.6999381 -0.7102139 -0.4487591  1.00000000 -0.7124406
## wt   -0.8676594  0.7824958  0.8879799  0.6587479 -0.71244065  1.0000000
## qsec  0.4186840 -0.5912421 -0.4336979 -0.7082234  0.09120476 -0.1747159
## vs    0.6640389 -0.8108118 -0.7104159 -0.7230967  0.44027846 -0.5549157
## am    0.5998324 -0.5226070 -0.5912270 -0.2432043  0.71271113 -0.6924953
## gear  0.4802848 -0.4926866 -0.5555692 -0.1257043  0.69961013 -0.5832870
## carb -0.5509251  0.5269883  0.3949769  0.7498125 -0.09078980  0.4276059
##             qsec         vs          am       gear        carb
## mpg   0.41868403  0.6640389  0.59983243  0.4802848 -0.55092507
## cyl  -0.59124207 -0.8108118 -0.52260705 -0.4926866  0.52698829
## disp -0.43369788 -0.7104159 -0.59122704 -0.5555692  0.39497686
## hp   -0.70822339 -0.7230967 -0.24320426 -0.1257043  0.74981247
## drat  0.09120476  0.4402785  0.71271113  0.6996101 -0.09078980
## wt   -0.17471588 -0.5549157 -0.69249526 -0.5832870  0.42760594
## qsec  1.00000000  0.7445354 -0.22986086 -0.2126822 -0.65624923
## vs    0.74453544  1.0000000  0.16834512  0.2060233 -0.56960714
## am   -0.22986086  0.1683451  1.00000000  0.7940588  0.05753435
## gear -0.21268223  0.2060233  0.79405876  1.0000000  0.27407284
## carb -0.65624923 -0.5696071  0.05753435  0.2740728  1.00000000
```

Ce code extrait les variables numériques du jeu de données *mtcars*, calcule leur matrice de corrélation en utilisant le coefficient de Pearson *(méthode par défaut ici)*, et affiche le résultat.

Cette méthode renvoie une **matrice de corrélation** qui contient le **coefficient de correlation de Pearson** entre chaque combinaison par paires possibles de variables. Ces coefficients indiquent la force et la direction de la relation linéaire entre deux variables.

Structure d'une matrice de corrélation

Les valeurs sur la diagonale principale sont égales à 1, car chaque variable est parfaitement corrélée avec elle-même.

Les coefficients varient entre -1 et 1 :

- +1 : corrélation positive *(quand une variable augmente, l'autre augmente proportionnellement).*
- -1 : corrélation négative *(quand une variable augmente, l'autre diminue proportionnellement).*
- 0 : pas de relation linéaire entre les variables.

Résultats obtenus (Exemples)

- Corrélation négative : *"mpg" et "wt" ont une correlation de -0.8676594 indiquant qu'une augmentation du poids du véhicule est associée à une diminution de l'efficacité énergétique.*
- Corrélation positive : *"cyl" et "disp" ont une correlation de 0.9020329 signifiant que les véhicules avec un plus grand nombre de cylindres ont tendance à avoir des moteurs de plus grande cylindrée.*
- Faible ou Absence de corrélation : *"am" et "carb" ont une correlation de 0.05753435 indiquant qu'il n'y a pratiquement aucune relation linéaire.*

Résumé statistique des coefficients de corrélation

```
summary(correlation_matrix)
```

```
##       mpg                cyl                disp                hp          
##  Min.   :-0.867659   Min.   :-0.852162   Min.   :-0.84755   Min.   :-0.77617  
##  1st Qu.:-0.811860   1st Qu.:-0.645590   1st Qu.:-0.65072   1st Qu.:-0.57849  
##  Median : 0.418684   Median :-0.492687   Median :-0.43370   Median :-0.12570  
##  Mean   :-0.004587   Mean   : 0.006774   Mean   : 0.01157   Mean   : 0.09153  
##  3rd Qu.: 0.631936   3rd Qu.: 0.807472   3rd Qu.: 0.83946   3rd Qu.: 0.77038  
##  Max.   : 1.000000   Max.   : 1.000000   Max.   : 1.00000   Max.   : 1.00000  
##       drat                wt                qsec                vs          
##  Min.   :-0.71244   Min.   :-0.86766   Min.   :-0.70822   Min.   :-0.81081  
##  1st Qu.:-0.57435   1st Qu.:-0.63789   1st Qu.:-0.51247   1st Qu.:-0.64001  
##  Median : 0.09120   Median :-0.17472   Median :-0.21268   Median : 0.16835  
##  Mean   : 0.08753   Mean   : 0.01557   Mean   :-0.06839   Mean   :-0.01324  
##  3rd Qu.: 0.69039   3rd Qu.: 0.72062   3rd Qu.: 0.25494   3rd Qu.: 0.55216  
##  Max.   : 1.00000   Max.   : 1.00000   Max.   : 1.00000   Max.   : 1.00000  
##        am                 gear              carb        
##  Min.   :-0.69250   Min.   :-0.5833   Min.   :-0.6562  
##  1st Qu.:-0.38291   1st Qu.:-0.3527   1st Qu.:-0.3209  
##  Median : 0.05753   Median : 0.2060   Median : 0.2741  
##  Mean   : 0.09574   Mean   : 0.1349   Mean   : 0.1421  
##  3rd Qu.: 0.65627   3rd Qu.: 0.5899   3rd Qu.: 0.4773  
##  Max.   : 1.00000   Max.   : 1.0000   Max.   : 1.0000
```

Pour venir compléter ce qui a été juste avant, ce code produit un résumé statistique des coefficients de corrélation, comprenant les valeurs minimales, maximales, la médiane et les quartiles, offrant ainsi une vue d'ensemble de leur distribution.

Nb : la fonction *describe()* peut également être utilisée pour explorer la matrice de corrélation. Ici, la fonction *summary()* a été privilégiée pour sa simplicité et parce que les informations qu'elle fournit étaient suffisantes pour les besoins de cette analyse.

Test de significativité de la corrélation (p-value)

La fonction *rcorr()* du package *Hmisc* peut étre utilisée pour calculer le **niveau de significativité** pour les **corrélations de Pearson** et **de Spearman**. En utilisant cette fonction le **coefficient de corrélation r de Pearson ou rho de Spearman** est calculer pour toutes les paires de variables possibles dans la table de donnée.

```
rcorr(correlation_matrix, type = c("pearson"))
```

```
##        mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
## mpg   1.00 -0.99 -0.99 -0.96  0.94 -0.99  0.71  0.93  0.83  0.77 -0.80
## cyl  -0.99  1.00  0.99  0.97 -0.92  0.97 -0.77 -0.96 -0.78 -0.74  0.82
## disp -0.99  0.99  1.00  0.94 -0.95  0.99 -0.69 -0.93 -0.84 -0.80  0.76
## hp   -0.96  0.97  0.94  1.00 -0.82  0.90 -0.88 -0.98 -0.63 -0.56  0.92
## drat  0.94 -0.92 -0.95 -0.82  1.00 -0.97  0.47  0.79  0.94  0.92 -0.57
## wt   -0.99  0.97  0.99  0.90 -0.97  1.00 -0.59 -0.87 -0.90 -0.85  0.70
## qsec  0.71 -0.77 -0.69 -0.88  0.47 -0.59  1.00  0.90  0.20  0.14 -0.95
## vs    0.93 -0.96 -0.93 -0.98  0.79 -0.87  0.90  1.00  0.59  0.54 -0.91
## am    0.83 -0.78 -0.84 -0.63  0.94 -0.90  0.20  0.59  1.00  0.98 -0.34
## gear  0.77 -0.74 -0.80 -0.56  0.92 -0.85  0.14  0.54  0.98  1.00 -0.24
## carb -0.80  0.82  0.76  0.92 -0.57  0.70 -0.95 -0.91 -0.34 -0.24  1.00
##
## n= 11
##
##
## P
##      mpg    cyl    disp   hp     drat   wt     qsec   vs     am     gear
## mpg         0.0000 0.0000 0.0000 0.0000 0.0000 0.0147 0.0000 0.0017 0.0058
## cyl  0.0000        0.0000 0.0000 0.0000 0.0000 0.0059 0.0000 0.0044 0.0099
## disp 0.0000 0.0000        0.0000 0.0000 0.0000 0.0183 0.0000 0.0011 0.0030
## hp   0.0000 0.0000 0.0000        0.0020 0.0002 0.0004 0.0000 0.0359 0.0704
## drat 0.0000 0.0000 0.0000 0.0020        0.0000 0.1426 0.0035 0.0000 0.0000
## wt   0.0000 0.0000 0.0000 0.0002 0.0000        0.0563 0.0005 0.0001 0.0008
## qsec 0.0147 0.0059 0.0183 0.0004 0.1426 0.0563        0.0002 0.5474 0.6815
## vs   0.0000 0.0000 0.0000 0.0000 0.0035 0.0005 0.0002        0.0539 0.0839
## am   0.0017 0.0044 0.0011 0.0359 0.0000 0.0001 0.5474 0.0539        0.0000
## gear 0.0058 0.0099 0.0030 0.0704 0.0000 0.0008 0.6815 0.0839 0.0000
## carb 0.0032 0.0020 0.0070 0.0000 0.0700 0.0171 0.0000 0.0000 0.3056 0.4849
##      carb
## mpg  0.0032
## cyl  0.0020
## disp 0.0070
## hp   0.0000
## drat 0.0700
## wt   0.0171
## qsec 0.0000
## vs   0.0000
## am   0.3056
## gear 0.4849
## carb
```

```
# Autres options : rcorr(as.matrix(numeric_mtcars[,1:11]))
```

Comme résultat, la fonction *rcorr()* renvoie une liste avec les éléments suivants :

- **r** : les valeurs affichées dans la première partie représentant les **coefficients de corrélation de Pearson**
- **n** : la matrice du nombre d'observations utilisé dans l'analyse de chaque paire de variables
- **p** : les **p-values** correspondant aux **niveaux de significativité** des **correlations** : si $p < 0.05$ alors on rejette l'hypothèse nulle d'absence de corrélation ; si $p > 0.05$ alors la corrélation n'est pas significative

Exemple 1 : Relation *mpg* et *qsec*

- Coefficient : **0.71**. Une corrélation positive indique ici que les voitures avec des temps de 1/4 de mile (*qsec*) plus longs sont associées à une consommation de carburant plus élevée (*mpg*).
- P-value : **0.0147**. Cette corrélation est significative au seuil de 5%.

Exemple 2 : Relation *carb* et *am*

- Coefficient : **-0.34**. Une faible corrélation négative indique une légère tendance indiquant que les voitures automatiques (*am*) ont moins de carburateurs (*carb*).
- P-value : **0.3056**. Cette relation n'est pas significative (p > 0.05). On ne peut pas conclure qu'il y a une relation réelle.

Corrélogramme : visualisation d'une matrice de corrélation

Plusieurs méthodes sont disponibles dans **R** pour dessiner un **corrélogramme**. Il est possible d'utiliser soit la fonction *symnum()*, la fonction *corrplot()* ou des **nuages de points** pour faire le graphique de la **matrice de corrélation**.

Utiliser la fonction *symnum*

La fonction *symnum()* de **R** remplace les **coefficients de corrélation** par des symboles en fonction de la valeur. Elle prend la **matrice de corrélation** comme argument.

```
symnum(correlation_matrix, abbr.colnames = FALSE)
```

```
##      mpg cyl disp hp drat wt qsec vs am gear carb
## mpg  1
## cyl  +   1
## disp +   *   1
## hp   ,   +   ,    1
## drat ,   ,   ,    .  1
## wt   +   ,   +    ,  ,    1
## qsec .   .   .    ,          1
## vs   ,   +   ,    ,  .    .  ,    1
## am   .   .   .       ,    ,          1
## gear .   .   .       ,    .          ,  1
## carb .   .   .    ,       .  ,    .          1
## attr(,"legend")
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
```

Comme indiqué dans la légende, les **coefficient de correlation** entre **0** et **0.3** sont remplacés par une espace(' ') ; les **coefficient de correlation** entre **0.3** et **0.6** sont remplacés par une espace(.) ; etc.

Faire un corrélogramme avec la fonction corrplot de R

Il est nécessaire d'installer le package *corrplot* qui permet de faire une visualisation graphique de la **matrice de corrélation**.

La fonction *corrplot()* prend la **matrice de corrélation** comme premier argument. Le second argument *(type = upper)* est utilisé pour afficher seulement le triangle supérieur de la **matrice de corrélation**.

```
corrplot(correlation_matrix, type = "upper", order = "hclust", tl.col = "black", tl.srt = 45)
```

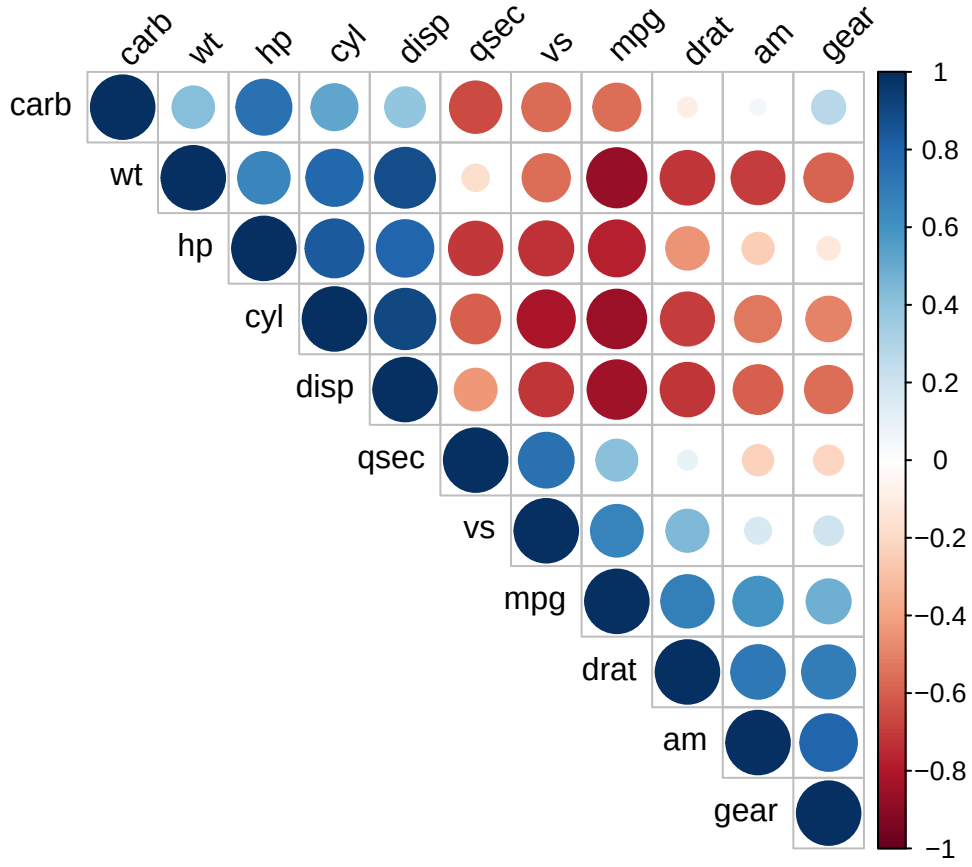


Figure 1: Corrélogramme de la matrice mtcars

Les **corrélations positives** sont affichées en bleu et les **corrélations négatives** en rouge. L'intensité de la couleur et la taille des cercles sont proportionnelles aux **coefficients de corrélation**. A droite du **corrélogramme**, la légende de couleurs montre les **coefficients de corrélation** et les couleurs correspondantes.

La matrice de corrélation est réarrangée en fonction des coefficients de corrélation en utilisant la méthode hclust. Les arguments *tl.col* (text label color) et *tl.srt* (text label string rotation) sont utilisés pour changer la couleur et la rotation des étiquettes de texte.

Ressources

Si besoin, voici une liste de ressources en ligne pour aider à l'interprétation d'une matrice de corrélation :

- https://datatab.fr/tutorial/correlation
- https://www.questionpro.com/blog/fr/matrice-de-correlation/
- https://www.sthda.com/french/wiki/matrice-de-correlation-guide-simple-pour-analyser-formater-et-visualiser
- https://psychometrie.jlroulin.fr/cours/aide_quizz.html?H31.html
- https://support.minitab.com/fr-fr/minitab/help-and-how-to/statistics/basic-statistics/how-to/correlation/interpret-the-results/all-statistics-and-graphs/
- https://www.sthda.com/french/wiki/visualiser-une-matrice-de-correlation-par-un-correlogramme#google_vignette
- https://delladata.fr/correlation-deux-a-deux-correlation-des-paires-ou-pairewise-correlations/
- https://jmeunierp8.github.io/ManuelJamovi/s12.html

Analyse en Composantes Principales (ACP) : Analyse et Visualisation

L'Analyse en Composantes Principales (ACP) est une méthode statistique multivariée utilisée pour explorer, décrire et simplifier un jeu de données quantitatives. Elle est particulièrement utile lorsque les données comportent de nombreuses variables inter-corrélées, ce qui complique l'interprétation. En identifiant des composantes principales (axes factoriels), l'ACP permet de concentrer l'information dans un nombre réduit de dimensions tout en conservant un maximum de variance.

L'ACP poursuit deux objectifs majeurs :

1. *Réduction de la dimensionnalité* : l'ACP synthétise plusieurs variables corrélées en un nombre restreint de variables synthétiques appelées composantes principales. Chaque composante principale est une combinaison linéaire des variables initiales et est ordonnée par importance selon la variance expliquée.
2. *Visualisation et Interprétation* : l'ACP permet de produire des représentations graphiques facilitant l'interprétation des relations entre individus et variables.

1. Packages

```
library(ggplot2)
library(factoextra)
library(FactoMineR)
```

Pour réaliser ce travail, plusieurs outils statistiques et de visualisation ont été mobilisés à l'aide des packages R suivants : *ggplot2*, *factoextra*, ainsi que *FactoMineR*.

Nb : l'utilisation du package R *explor()* est aussi possible pour réaliser une ACP.

2. ACP

2.1 Lecture des données

La première étape consiste à importer la table de données *mtcars* disponible dans **R** afin d'en observer la structure et d'évaluer leur conformité avec les attentes initiales.

```
mtcars <- read.csv("data/mtcars.csv", header = TRUE)
head(mtcars, 5)
```

```
##        manufacturer  mpg cyl disp  hp drat    wt  qsec vs am gear carb
## 1         Mazda RX4 21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## 2     Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## 3        Datsun 710 22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## 4    Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## 5 Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
```

- **Format** : un fichier de données *"mtcars"* avec 32 observations sur 11 variables : *mpg* (Miles/(US) gallon), *cyl* (Nombre de cylindres), *disp* (Cylindrée), *hp* (Puissance brute), *drat* (Rapport de pont arrière), *wt* (Poids (1000 lbs)), *qsec* (Temps au 1/4 mille), *vs* (Moteur (0 = V-shaped, 1 = straight)), *am* (Transmission (0 = automatique, 1 = manuelle)), *gear* (Nombre de vitesses), *carb* (Nombre de carburateurs).
- **Description** : les données ont été extraites du magazine Motor Trend US de 1974 et comprennent la consommation de carburant et 10 aspects de la conception et des performances de 32 automobiles (modèles 1973-1974).

2.2. Normaliser les données

Il est nécessaire d'abord de normaliser les données pour qu'elles aient une moyenne nulle et une variance unitaire

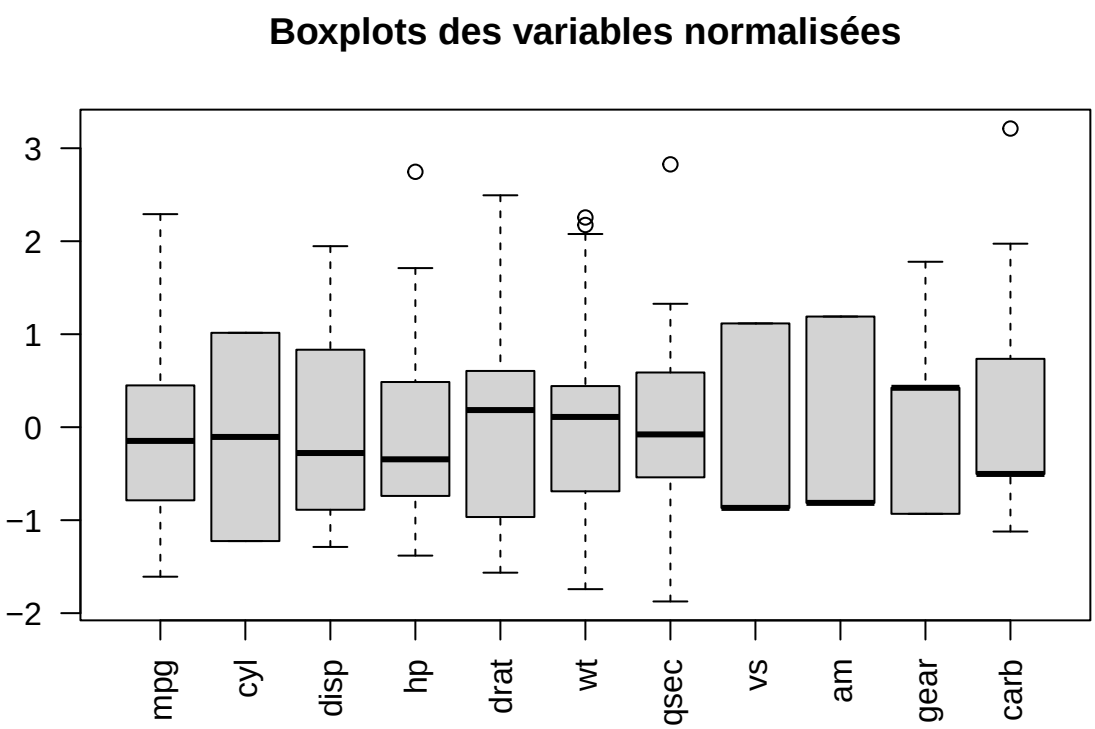
$$Z_{ij} = \frac{X_{ij} - \bar{X}_j}{s_j}$$

où : Z est la matrice standardisée, X est la matrice originale, $\bar{X}_j$ est la moyenne de la variable j, et s_j est l'écart-type de la variable j.

```
mtcars_scaled <- scale(numeric_mtcars)
```

Différents graphiques peuvent être utilisés pour visualiser les données. Ici, le graphique présente des boxplots pour chaque variable normalisée, ce qui permet d'évaluer visuellement leur distribution, leu rdispersion, et la présence éventuelle de valeurs aberrantes.

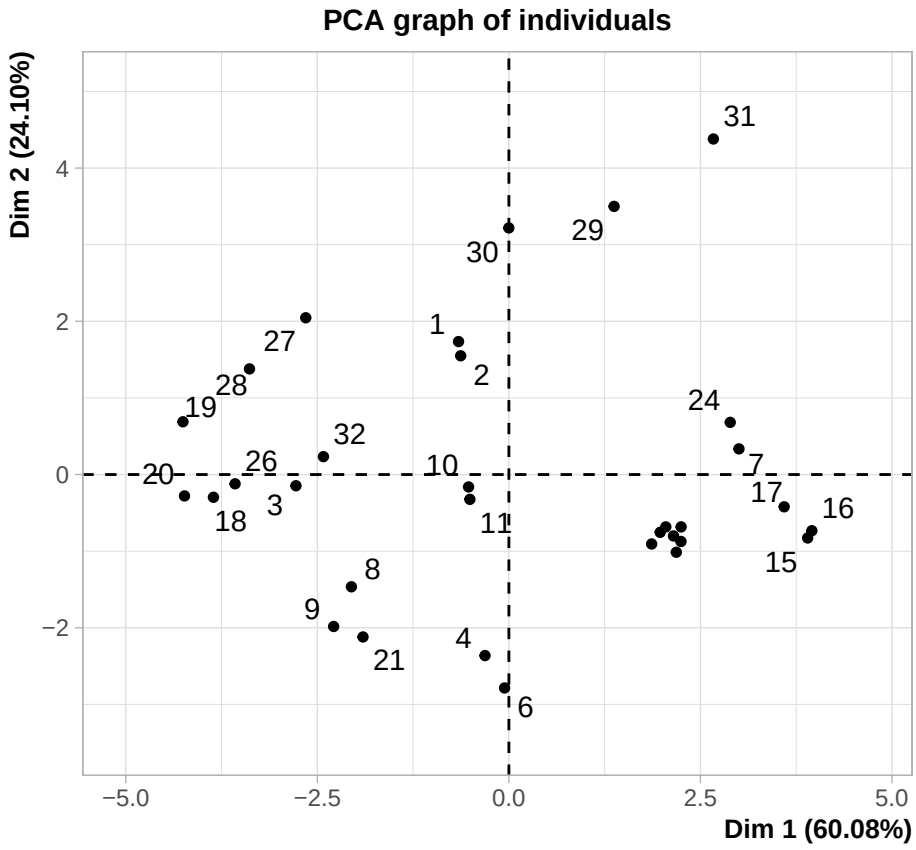
```
boxplot(mtcars_scaled, main = "Boxplots des variables normalisées", las = 2)
```

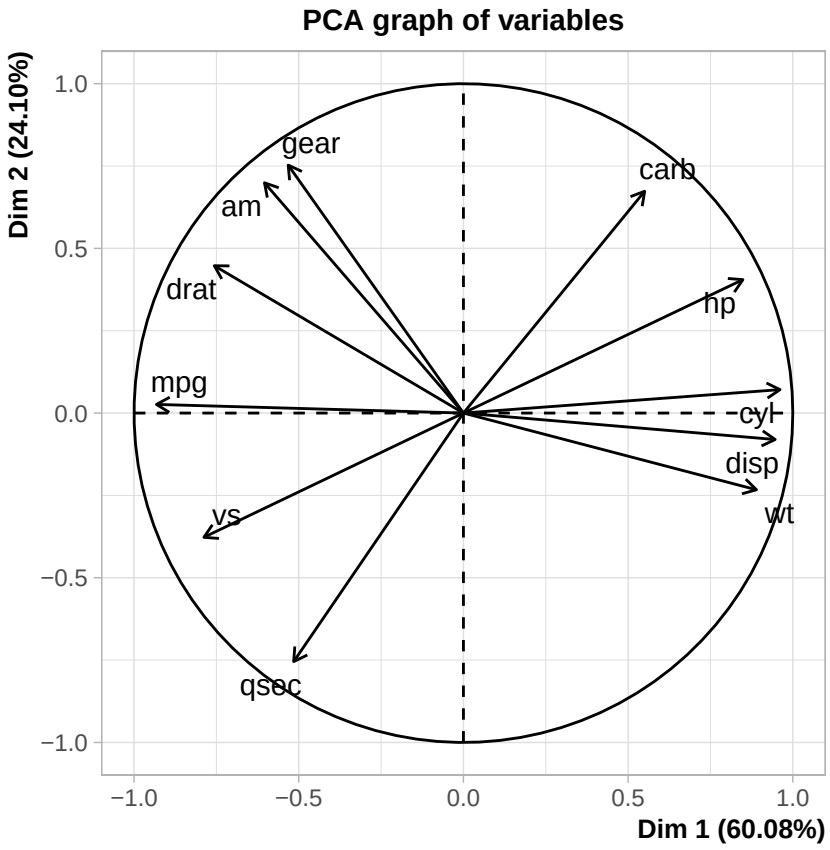


2.3. Réalisation de l'ACP

Le code suivant a été utilisé pour réaliser une ACP sur le jeu de données *mtcars* :

```
ACP_res <- PCA(mtcars_scaled[,1:11])
```






```
print(summary(ACP_res, ncp = 2))
```

```
##
## Call:
## PCA(X = mtcars_scaled[, 1:11])
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
## Variance               6.608   2.650   0.627   0.270   0.223   0.212   0.135
## % of var.             60.076  24.095   5.702   2.451   2.031   1.924   1.230
## Cumulative % of var.  60.076  84.172  89.873  92.324  94.356  96.279  97.509
##                        Dim.8   Dim.9  Dim.10  Dim.11
## Variance               0.123   0.077   0.052   0.022
## % of var.              1.117   0.700   0.473   0.200
## Cumulative % of var.  98.626  99.327  99.800 100.000
##
## Individuals (the 10 first)
##          Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2
## 1     |  2.234 | -0.657  0.204  0.087 |  1.735  3.551  0.604 |
## 2     |  2.081 | -0.629  0.187  0.091 |  1.550  2.833  0.555 |
## 3     |  2.987 | -2.779  3.653  0.866 | -0.146  0.025  0.002 |
## 4     |  2.521 | -0.312  0.046  0.015 | -2.363  6.584  0.879 |
## 5     |  2.456 |  1.974  1.844  0.646 | -0.754  0.671  0.094 |
## 6     |  3.014 | -0.056  0.001  0.000 | -2.786  9.151  0.855 |
```

```
## 7    |  3.187 |  3.003  4.264  0.888 |  0.335  0.132  0.011 |
## 8    |  2.841 | -2.055  1.998  0.523 | -1.465  2.531  0.266 |
## 9    |  3.733 | -2.287  2.474  0.375 | -1.984  4.639  0.282 |
## 10   |  1.907 | -0.526  0.131  0.076 | -0.162  0.031  0.007 |
##
## Variables (the 10 first)
##         Dim.1    ctr   cos2    Dim.2    ctr   cos2
## mpg  | -0.932 13.143  0.869 |  0.026  0.026  0.001 |
## cyl  |  0.961 13.981  0.924 |  0.071  0.191  0.005 |
## disp |  0.946 13.556  0.896 | -0.080  0.243  0.006 |
## hp   |  0.848 10.894  0.720 |  0.405  6.189  0.164 |
## drat | -0.756  8.653  0.572 |  0.447  7.546  0.200 |
## wt   |  0.890 11.979  0.792 | -0.233  2.046  0.054 |
## qsec | -0.515  4.018  0.266 | -0.754 21.472  0.569 |
## vs   | -0.788  9.395  0.621 | -0.377  5.366  0.142 |
## am   | -0.604  5.520  0.365 |  0.699 18.440  0.489 |
## gear | -0.532  4.281  0.283 |  0.753 21.377  0.567 |
## NULL
```

Ici, il a été spécifié ncp=2 afin d'afficher uniquement les résultats des deux premières dimensions principales. Cela permet de limiter la taille des tableaux et de faciliter l'interprétation.

2.3.1. Critères de sélection du nombre de composantes principales

La détermination du nombre optimal de dimensions à retenir peut se faire en utilisant plusieurs critères statistiques :

1. **la Règle de kaiser ou de la valeur propre moyenne** : Cette règle propose de conserver les composantes principales ayant une valeur propre supérieure à la valeur moyenne des valeurs propres. Si la valeur propre dépasse cette moyenne, la composante est considérée comme significative.
2. **la Règle des 80% d'information cumulée** : Il s'agit de sélectionner le nombre de dimensions nécessaires pour expliquer au moins 80% de la variance totale des données. Cela permet de conserver une proportion importante de l'information tout en réduisant la complexité du modèle.

Dans le cas présenté ici, lorsque l'on observe les pourcentages cumulés des valeurs propres (variances), on peut remarquer que la somme des deux premières dimensions est égale à 84.172%, permettant de se focaliser sur ces deux premières.

2.3.2. Interprétation des composantes principales (dimension) retenues

- **Dimension 1 (60.08%)**

Elément - : (MPG : contrib = 13.14%) ; (VS : contrib = 9.39%) | ***Elément +*** : (CYL : contrib = 13.98%) ; (DISP : contrib = 13.58%) ; (WT : contrib = 11.98%) ; (HP : contrib = 10.89%)

Contribution moyenne : 100% / 11 = 9.09% (11 est le nb de variable quantitatives de départ utilisés pour les calculs)

Somme contrib = 72.94% (CYL : contrib = 13.98% ; cos2 = 0.924 ; +) ; (DISP : contrib = 13.58% ; cos2 = 0.896 ; +) ; (MPG : contrib = 13.14% ; cos2 = 0.869 ; -) ; (WT : contrib = 11.98% ; cos2 = 0.892 ; +) ; (HP : contrib = 10.89% ; cos2 = 0.720 ; +) ; (VS : contrib = 9.39% ; cos2 = 0.621 ; -)

Les premières composantes (tendance principale de 60.08%) issue de l'ACP normale est expliquée à hauteur de 72.94% par les variables CYL, DISP, MPG, WT qui se projetent en positif, et par les variables MPG et VS qui se projettent en négatif. Du fait de cette projection, cette tendance oppose [CYL, DISP, WT, HP] au groupe [MPG, VS].

- **Dimension 2 (24.10%)**

Elément - : (QSEC : contrib = 21.47%) | ***Elément +*** : (AM : contrib = 18.44%) ; (GEAR : contrib = 21.38%)

Contribution moyenne : 100% / 11 = 9.09% (11 est le nb de variable quantitatives de départ utilisés pour les calculs)

Somme contrib = 61.29% (QSEC : contrib = 21.47%; cos2 = 0.569 ; 6) ; (GEAR : contrib = 21.38% ; cos2 = 0.567; +) ; (AM : contrib = 18.44% ; cos2 = 0.489 ; +)

Les premières composantes (tendance principale de 24.10%) issue de l'ACP normale est expliquée à hauteur de 61.29% par les variables AM, GEAR qui se projetent en positif, et par la variable QSEC qui se projette en négatif. Du fait de cette projection, cette tendance oppose [QSEC] au groupe [AM, GEAR].

2.3.3. Coordonnées des points projetés dans l"espace des composantes principales

```r
ACP <- prcomp(mtcars_scaled, scale. = TRUE)
print(ACP$x)
```

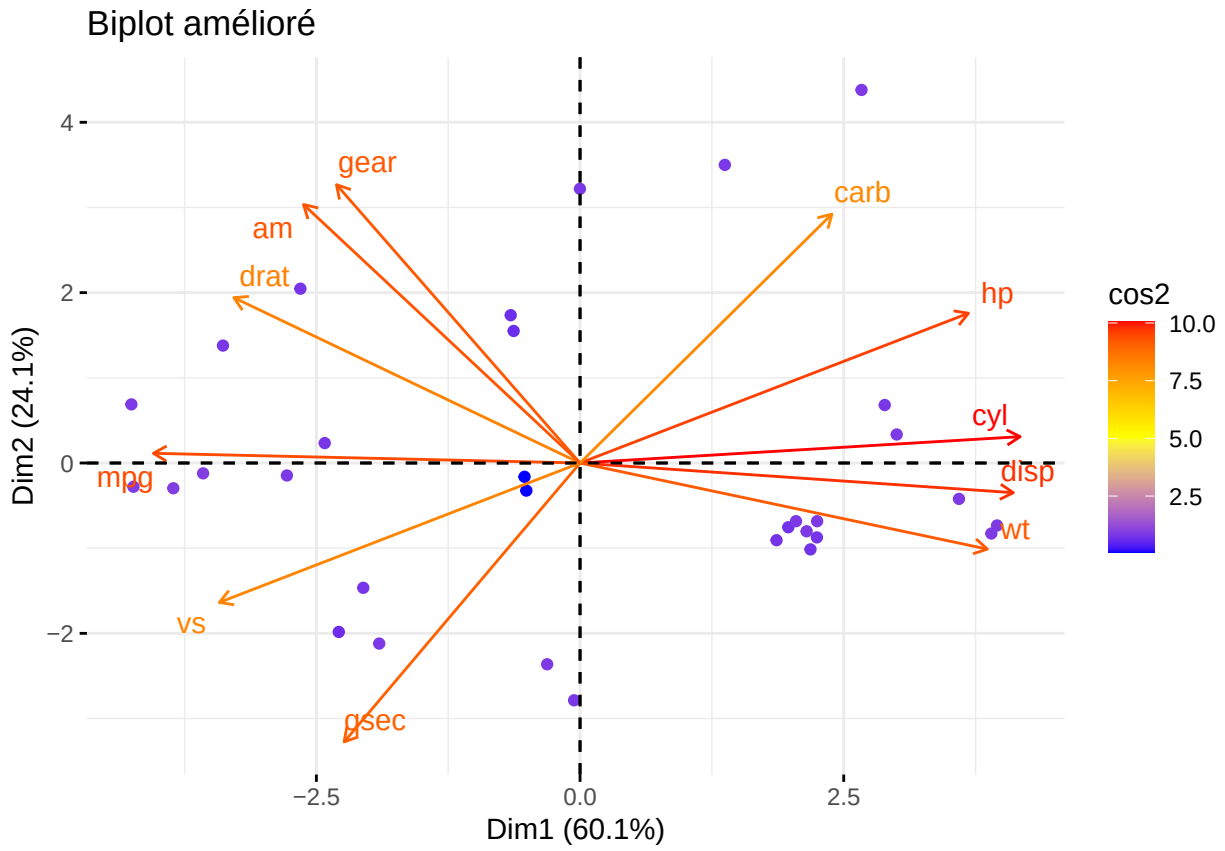
```
##                 PC1        PC2        PC3          PC4          PC5
##  [1,] -0.6468627420  1.7081142 -0.5917309  0.113702214  0.945523363
##  [2,] -0.6194831460  1.5256219 -0.3763013  0.199121210  1.016680740
##  [3,] -2.7356242748 -0.1441501 -0.2374391 -0.245215450 -0.398762288
##  [4,] -0.3068606268 -2.3258038 -0.1336213 -0.503800355 -0.549208936
##  [5,]  1.9433926844 -0.7425211 -1.1165366  0.074461963 -0.207515698
##  [6,] -0.0552534228 -2.7421229  0.1612456 -0.975167425 -0.211665375
##  [7,]  2.9553851233  0.3296133 -0.3570461 -0.051529216 -0.343847875
##  [8,] -2.0229593244 -1.4421056  0.9290295 -0.142129082  0.316651386
##  [9,] -2.2513839535 -1.9522879  1.7689364  0.287210957  0.333682355
## [10,] -0.5180912217 -0.1594610  1.4692603  0.066263362  0.069624161
## [11,] -0.5011860079 -0.3187934  1.6570701  0.094357222  0.148803650
## [12,]  2.2124096339 -0.6727099 -0.3694707 -0.129797905  0.378611141
## [13,]  2.0155715693 -0.6724606 -0.4768341 -0.210991001  0.355611763
## [14,]  2.1147047372 -0.7891129 -0.2904620 -0.175332868  0.432140303
## [15,]  3.8383725118 -0.8149087  0.6370972  0.290505877  0.048245223
## [16,]  3.8918495626 -0.7218314  0.7092612  0.405336898 -0.003899176
## [17,]  3.5363862158 -0.4145024  0.5402468  0.665665306 -0.208027112
## [18,] -3.7955510831 -0.2920783 -0.4161681  0.055191058 -0.219981109
## [19,] -4.1870356784  0.6775721 -0.2035831  1.167526096 -0.097674091
## [20,] -4.1675359344 -0.2748890 -0.4589124  0.183313028 -0.222152228
## [21,] -1.8741790870 -2.0864529  0.1543265  0.050514126 -0.039299002
## [22,]  2.1504414942 -0.9982442 -1.1503639 -0.584982249  0.226237802
## [23,]  1.8340369797 -0.8921886 -0.9472872  0.005694071  0.252565496
## [24,]  2.8434957523  0.6701037 -0.1605593  0.814340105 -0.389118986
## [25,]  2.2105479148 -0.8600504 -1.0279577  0.146420497 -0.299261925
## [26,] -3.5176818134 -0.1192950 -0.4464716 -0.013427353 -0.206753365
## [27,] -2.6095003965  2.0141425 -0.8172519  0.568564789  0.597313744
## [28,] -3.3323844512  1.3568877 -0.4467167 -1.153197531 -0.694667640
## [29,]  1.3513346957  3.4448780 -0.1343943  0.590098358 -1.101648091
## [30,] -0.0009743305  3.1683750  0.3957610 -0.938933017  0.848833976
## [31,]  2.6270897605  4.3107016  1.3315940 -0.877332804 -0.455265189
## [32,] -2.3824711412  0.2299603  0.4052798  0.223549117 -0.321777017
##                 PC6         PC7          PC8         PC9         PC10
##  [1,] -0.0169873733 -0.42648652  0.009631217 -0.14642303  0.06670350
##  [2,] -0.2417246434 -0.41620046  0.084520213 -0.07452829  0.12692766
##  [3,] -0.3487678138 -0.60884146 -0.585255765  0.13122859 -0.04573787
##  [4,]  0.0192969984 -0.04036075  0.049583029 -0.22021812  0.06039981
##  [5,]  0.1491927606  0.38350816  0.160297757  0.02117623  0.05983003
##  [6,] -0.2438358546 -0.29464160 -0.256612420  0.03222907  0.20165466
##  [7,]  0.7126920868 -0.13607792  0.171103449  0.17844547 -0.36086641
##  [8,] -0.0009889391  0.63946214 -0.163156195 -0.37698418 -0.29086529
##  [9,] -0.3338703384  0.62201034  0.105779936  0.86455356  0.11597058
## [10,]  0.8165308365  0.16117090 -0.099983313 -0.54092449  0.22093750
## [11,]  0.7308383757  0.09254430 -0.197306566 -0.30876072  0.34417564
## [12,]  0.1317014762 -0.01645498  0.194092435  0.05614966  0.06531727
## [13,]  0.2400263805  0.05123623  0.329669990  0.20501055  0.10761308
## [14,]  0.1801997325 -0.06675316  0.119252582  0.38704169  0.21191036
```

```
## [15,] -0.8844735483 -0.16615296 -0.138398783 -0.19333387 -0.06184979
## [16,] -0.8625868981 -0.19250873 -0.129305868 -0.19523562 -0.12094849
## [17,] -0.6536447300  0.03449804  0.391104141 -0.27447514 -0.27588169
## [18,] -0.4675796343 -0.03749941  0.625278746 -0.10550311  0.02717077
## [19,]  0.5180554279 -0.25316291  0.395045565 -0.23711675  0.15433928
## [20,] -0.3171521124  0.06617540  0.853947085  0.11313627  0.12606845
## [21,]  0.7236992559 -0.28027808 -0.207237627  0.44646972 -0.51147635
## [22,]  0.1062181942  0.09489585 -0.316055390 -0.10435633  0.13641143
## [23,]  0.2888101997  0.08161916 -0.321900593  0.12237636  0.29628634
## [24,]  0.9468795171 -0.21157976 -0.038657331  0.05282991 -0.32624525
## [25,] -0.1983310387  0.47269865  0.234144182 -0.20849043 -0.01547674
## [26,] -0.1449905641 -0.35850305 -0.089109764  0.02228967  0.08414018
## [27,] -0.3394265065  0.82032965 -0.634987241  0.12999660 -0.34968156
## [28,]  0.0165037718  0.51018011 -0.004140777 -0.29680350 -0.23980308
## [29,] -0.1746156635  0.41358868 -0.609167214  0.23280792  0.50262890
## [30,] -0.0097569921  0.02967883 -0.014187801 -0.09813571 -0.14491815
## [31,] -0.0156094416 -0.18813730  0.558646792  0.34081133 -0.04706368
## [32,] -0.3263029217 -0.77995741 -0.476634473  0.04473670 -0.11767108
##               PC11
##  [1,]  0.179693570
##  [2,]  0.088644265
##  [3,] -0.094632914
##  [4,]  0.147611269
##  [5,]  0.146406899
##  [6,]  0.019545064
##  [7,]  0.171863162
##  [8,] -0.019090358
##  [9,]  0.159688512
## [10,] -0.124486227
## [11,] -0.034578568
## [12,] -0.396445135
## [13,] -0.197616838
## [14,] -0.142498830
## [15,]  0.262886205
## [16,]  0.039191100
## [17,] -0.224420191
## [18,] -0.208865888
## [19,]  0.246835364
## [20,] -0.031747839
## [21,]  0.063679725
## [22,]  0.049594456
## [23,]  0.045293027
## [24,] -0.099386307
## [25,]  0.122593248
## [26,] -0.005746448
## [27,] -0.111596656
## [28,]  0.030015592
## [29,] -0.042242570
## [30,]  0.043006835
## [31,]  0.062135486
## [32,] -0.145329008
```

2.4. Visualisation

2.4.a. Biplot classique

```
fviz_pca_biplot(ACP_res, geom.ind = "point", col.ind = "cos2",
                gradient.cols = c("blue", "yellow", "red"),
                col.var = "contrib", repel = TRUE,
                title = "Biplot amélioré")
```

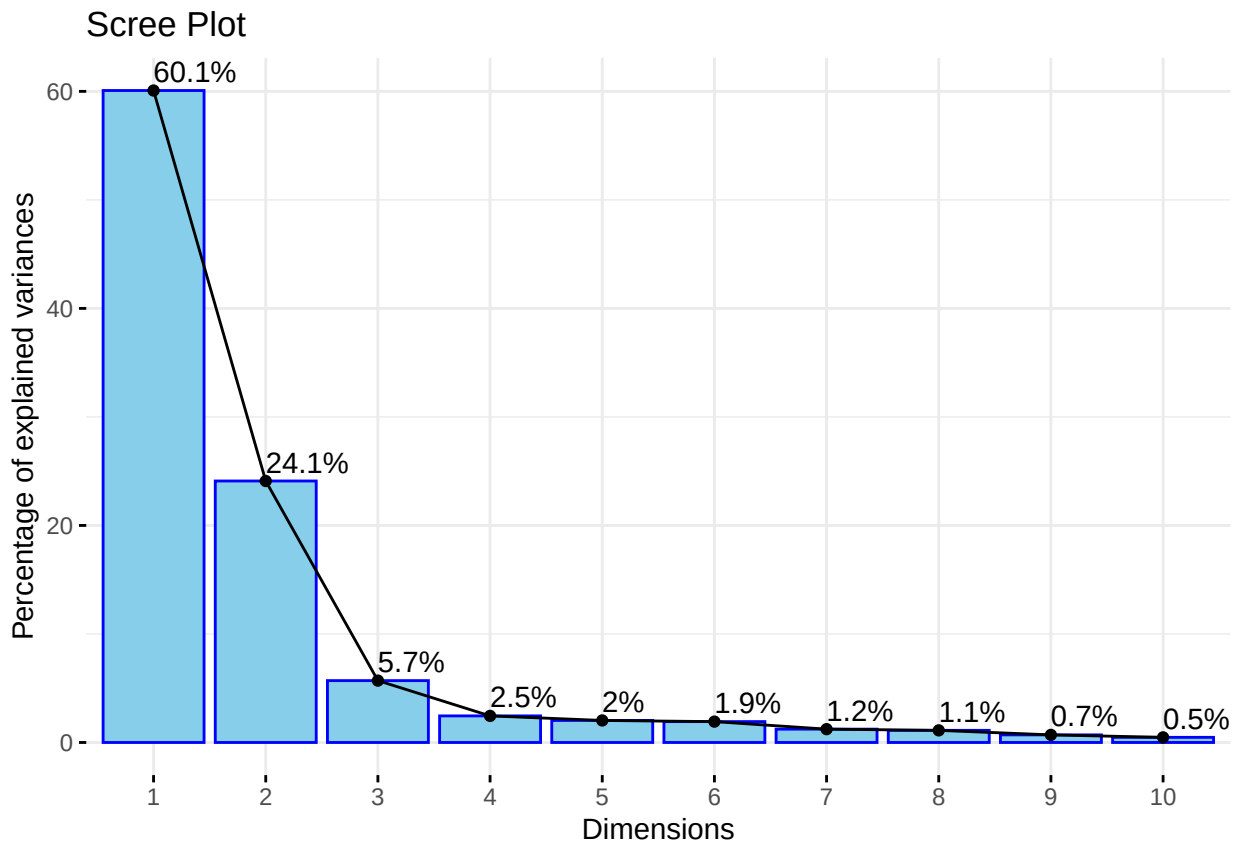


Interprétation des résultats :

- **Corrélation forte** : Si deux variables sont proches sur le cercle des corrélations, elles sont fortement liées.
- **Opposition** : Des variables opposées sur le plan sont négativement corrélées.
- **Proximité des individus** : Des individus proches dans le nuage des points sont similaires sur les variables mesurées.

2.4.b. Scree Plot

```
fviz_screeplot(ACP_res, addlabels = TRUE, barfill = "skyblue", barcolor = "blue",
    title = "Scree Plot")
```



Classification Ascendante Hierarchique (CAH) et Clustering

Il existe de nombreuses techniques statistiques visant à partinionner une population en différentes classes ou sous-groupes. La **Classification Ascendante Hiérarchique (CAH)** est l'une d'entre elles. On cherche à ce que les individus regroupés au sein d'une même classe *(homogénéité intra-classe)* soient le plus semblables possibles tandis que les classes soient le plus dissemblables *(hétérogénéité inter-classe)*.

Packages

```
library(ggplot2)
library(FactoMineR)
library(factoextra)
library(cluster)
library(corrplot)
library(dplyr)
```

Pour réaliser ce travail, plusieurs outils statistiques et de visualisation ont été mobilisés à l'aide des packages R suivants : *ggplot2*, *FactoMineR*, *factoextra*, *cluster*, *corrplot*, ainsi que *dplyr*.

Lecture des données

La première étape consiste à importer la table de données *mtcars* disponible dans **R** afin d'en observer la structure et d'évaluer leur conformité avec les attentes initiales.

```
mtcars <- read.csv("data/mtcars.csv", header = TRUE)
head(mtcars, 9)
```

```
##        manufacturer  mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 1         Mazda RX4 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## 2     Mazda RX4 Wag 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## 3        Datsun 710 22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## 4    Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## 5 Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## 6           Valiant 18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## 7        Duster 360 14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## 8         Merc 240D 24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## 9          Merc 230 22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
```

- **Format** : un fichier de données *"mtcars"* avec 32 observations sur 11 variables : *mpg* (Miles/(US) gallon), *cyl* (Nombre de cylindres), *disp* (Cylindrée), *hp* (Puissance brute), *drat* (Rapport de pont arrière), *wt* (Poids (1000 lbs)), *qsec* (Temps au 1/4 mille), *vs* (Moteur (0 = V-shaped, 1 = straight)), *am* (Transmission (0 = automatique, 1 = manuelle)), *gear* (Nombre de vitesses), *carb* (Nombre de carburateurs).
- **Description** : les données ont été extraites du magazine Motor Trend US de 1974 et comprennent la consommation de carburant et 10 aspects de la conception et des performances de 32 automobiles (modèles 1973-1974).

Classification Ascendante Hiérarchique (CAH)

Matrice de distances

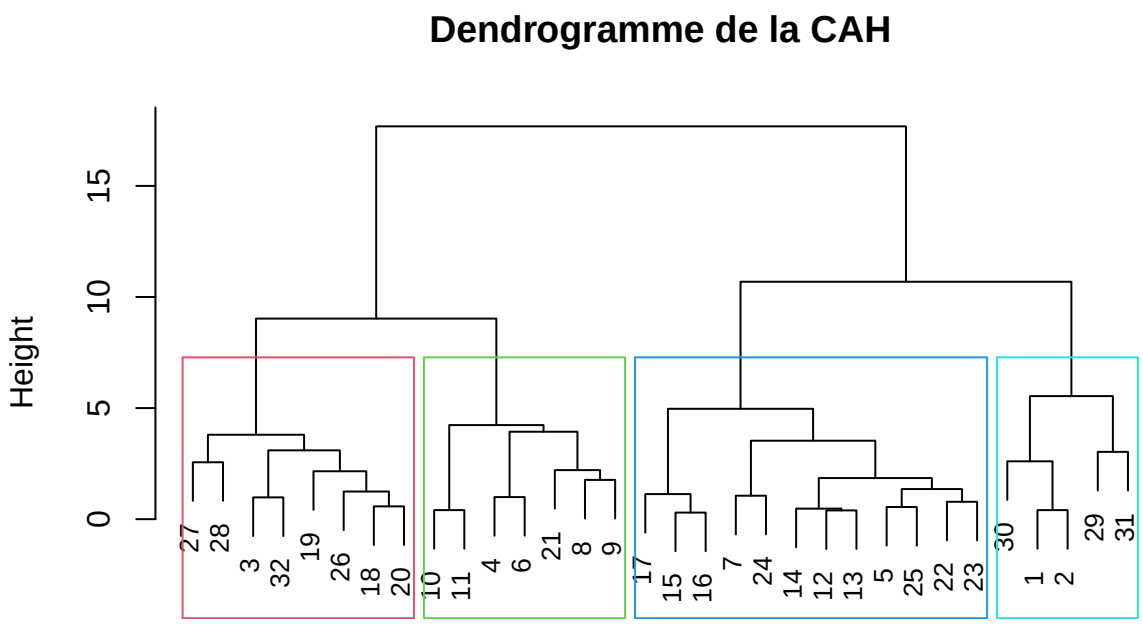
Le principe de la CAH est de rassembler des individus selon un critère de ressemblance défini au préalable qui s'exprimera sous la forme d'une matrice de distances, exprimant la distance existant entre chaque individu pris deux à deux. Deux observations identiques auront une distance nulle. Plus les deux observations seront dissemblables, plus la distance sera importante.

```
d <- dist(mtcars_scaled, method = "euclidean") # Matrice de distance
hc <- hclust(d, method = "ward.D2") # Regroupement hiérarchique
```

Dendrogramme

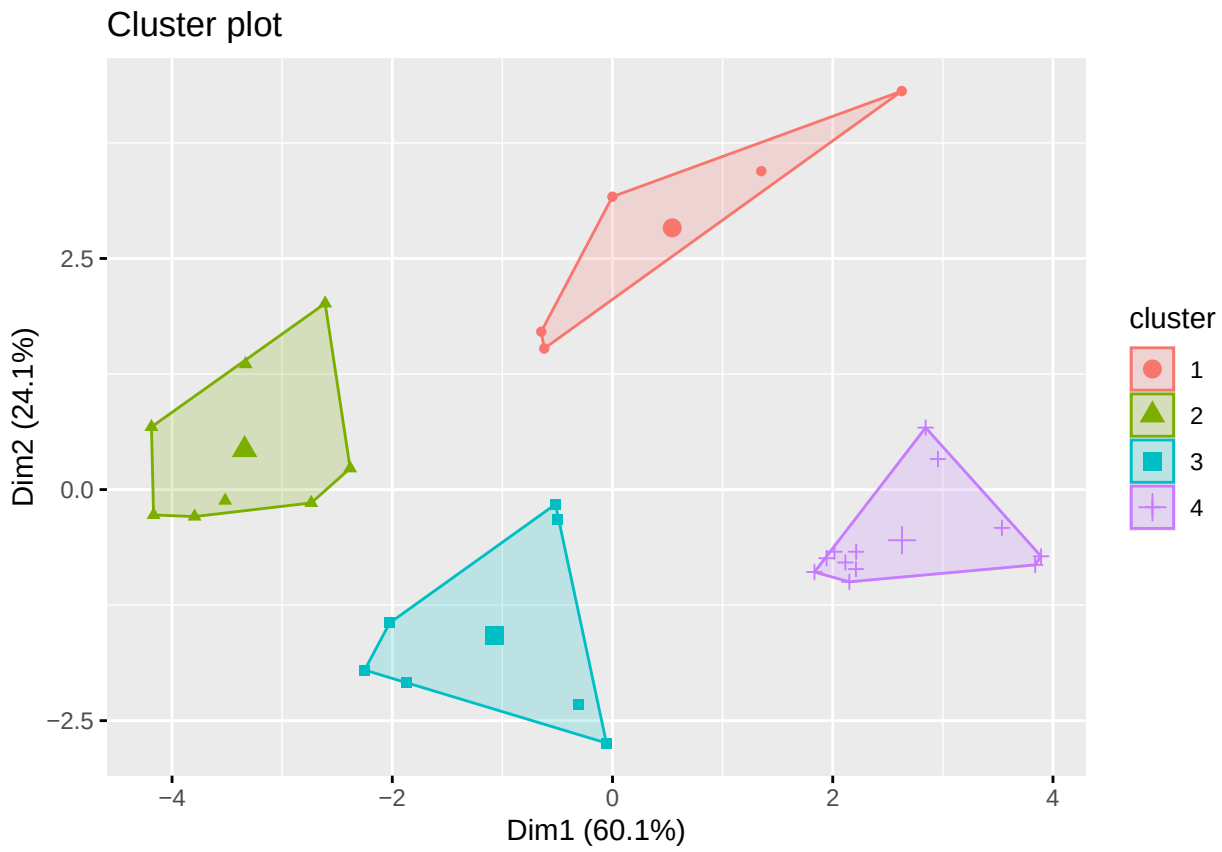
La CAH va ensuite rassembler les individus de manière itérative afin de produire un dendrogramme ou arbre de classification. La classification est ascendante car elle part des observations individuelles ; elle est hiérarchique car elle produit des classes ou groupes de plus en plus vastes, incluant des sous-groupes en leur sein. En découpant cet arbre à une certaine hauteur choisie, on produira la partition désirée.

```
# Dendrogramme
plot(hc, main = "Dendrogramme de la CAH", xlab = "", sub = "", cex = 0.8)

# Découpage en clusters (par exemple, k = 4)
k <- 4
rect.hclust(hc, k = k, border = 2:5)
```



Clustering

```r
# Ajouter les clusters aux données
clusters <- cutree(hc, k)
mtcars_clustered <- mtcars %>%
  mutate(Cluster = as.factor(clusters))

# Visualisation des clusters avec ggplot2 (ACP)
fviz_cluster(list(data = mtcars_scaled, cluster = clusters), geom = "point", stand = FALSE)
```



Ressources

Si besoin, voici une liste de ressources en ligne pour aider à l'interprétation d'une analyse en composantes principales :

- https://www.xlstat.com/fr/solutions/fonctionnalites/analyse-en-composantes-principales-acp
- https://www.soft-concept.com/surveymag/comment-lire-une-acp.html
- https://openclassrooms.com/fr/courses/4525281-realisez-une-analyse-exploratoire-de-donnees/5278723-interpretez-le-cercle-des-correlations
- https://marie-chavent.perso.math.cnrs.fr/wp-content/uploads/2013/10/Exemple_interpret_ACP.pdf
- https://laeq.github.io/LivreMethoQuantBolR/sect122.html
- https://r.qcbs.ca/workshop09/book-fr/analyse-en-composantes-principales.html
- https://sayl-85.webself.net/file/si1454787/Exercices%20de%20synth%C3%A8se%20corrig%C3%A9s%20AFC-fi22261017.pdf
- https://louernos-nature.fr/analyse-composantes-principales-logiciel-r/
- https://agritrop.cirad.fr/604013/1/RTBfoods_Manuel_Analyses%20Statistiques%20pour%20Visualiser%20les%20Donn%C3%A9es%20Sensorielles%20et%20les%20Relier%20aux%20Donn%C3%A9es%20Instrumentales.pdf

Voici également une liste de ressources en ligne pour aider à l'interprétation d'une classification ascendante hiérarchique :

- https://www.ceremade.dauphine.fr/~roche/CAH.pdf
- https://larmarange.github.io/guide-R/analyses_avancees/classification-ascendante-hierarchique.html
- https://gailloty.net/public-library-survey/classification-ascendante-hi%C3%A9rarchique-cah.html
- https://eric.univ-lyon2.fr/ricco/cours/didacticiels/R/cah_kmeans_avec_r.pdf

Infos

Cette analyse a été réalisée avec R (ver. 4.3.3).