Analyse des données européennes

FALL Khadim ,

2025-01-06

Contents

Le code source du projet est disponible sur GitHub :https://github.com/khadimfall2/Analysis.git

Les données analysées dans ce projet concernent différents indicateurs socio-économiques et environnementaux pour un ensemble de pays européens. Ces indicateurs offrent une vision globale de divers aspects comme la démographie, l'économie, l'énergie, l'emploi et l'environnement. Les analyses visent à explorer les structures sous-jacentes et les relations entre ces variables, ainsi qu'à regrouper les pays selon des caractéristiques communes.

Description des variables

Les données contiennent 16 variables descriptives de 30 pays européens , reflétant des dimensions clés :

Population au 1er janvier : Nombre absolu d'habitants.

Population jeune (15-29 ans) : Pourcentage de jeunes dans la population totale.

Premières demandes d'asile : Nombre absolu de demandes.

Écart de rémunération entre les sexes : Pourcentage de différence de salaire horaire brut moyen entre hommes et femmes.

Salaire minimum : Montant en euros par mois.

Décrocheurs scolaires précoces : Pourcentage de la population âgée de 18 à 24 ans quittant prématurément le système scolaire.

Taux d'inflation : Variation en pourcentage par rapport à l'année précédente.

Taux de chômage : Pourcentage de la population active âgée de 15 à 74 ans.

Taux de chômage des jeunes : Pourcentage de la population active de moins de 25 ans.

PIB par habitant : Produit intérieur brut en euros par habitant.

Dette brute du gouvernement : Pourcentage de la dette brute par rapport au PIB.

Émissions de gaz à effet de serre : Quantité moyenne en tonnes par habitant.

Énergies renouvelables : Pourcentage dans la consommation finale brute d'énergie.

Prix de l'électricité : Montant en euros par MWh, incluant les taxes.

Dépendance aux importations d'énergie : Pourcentage de dépendance à l'énergie importée.

Taux de risque de pauvreté ou d'exclusion sociale : Pourcentage de la population à risque de pauvreté ou d'exclusion sociale.

L'objectif de cette analyse est d'explorer et de réduire la dimensionnalité des données grâce à une analyse en composantes principales (ACP), puis de grouper les pays selon leurs caractéristiques à l'aide de méthodes

de classification. L'ACP permet de visualiser les similitudes entre pays et d'identifier les variables les plus remarquables . Les méthodes de classification, notamment la classification ascendante hiérarchique (CAH) et l'algorithme des centres mobiles (k-means), permettent d'interpréter les regroupements obtenus. Les résultats des classifications seront comparés afin de comprendre les proximités entre pays et leur cohérence.

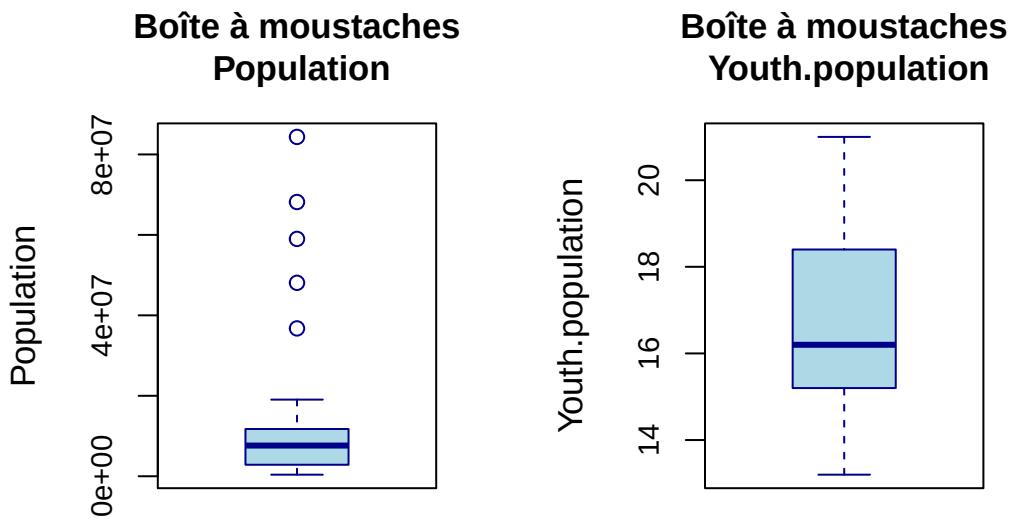
Pour la préparation des données, une normalisation a été appliquée dans certains cas pour rendre les variables comparables. L'analyse inclut la création de matrices de dissimilarité, l'utilisation de la décomposition en valeurs propres pour l'ACP, et l'application des méthodes de classification sur les données normalisées. Les regroupements obtenus seront interprétés à travers l'étude des centres de gravité, des inerties et des plans factoriels. Enfin, une attention particulière sera portée à l'analyse des proximités des pays dans des zones spécifiques de l'espace factoriel, afin de mieux comprendre les similarités entre pays.

```r
# Vérification et chargement des bibliothèques nécessaires
if (!require(ggplot2)) install.packages("ggplot2", dependencies = TRUE)
library(ggplot2)

# Chargement de TinyTeX si nécessaire (une seule fois)
if (!tinytex::is_tinytex()) {
  tinytex::install_tinytex()
}
knitr::opts_chunk$set(comment = NA)
```

```r
euro_data <- read.csv("data/euro.csv", header = TRUE, sep = ";")

# Normalisation des données (Min-Max Scaling)
euro_data_normalized <- as.data.frame(lapply(euro_data[, -1], function(x) {
  (x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE))
}))
```

Diagrammes à mmoustache

```r
# Supprimer la première colonne contenant les pays.
euro_data <- euro_data[, -1]


selected_columns <- colnames(euro_data)[1:2]

# Ajuster les marges et l'espacement
par(mfrow = c(1, 2),  # Deux graphiques côte à côte
    oma = c(2, 2, 2, 2),  # Marges extérieures : bas, gauche, haut, droite
    mar = c(5, 5, 4, 2))  # Marges intérieures : bas, gauche, haut, droite

# Boucle pour tracer les boîtes à moustaches
for (col_name in selected_columns) {
  boxplot(euro_data[[col_name]],
          main = paste("Boîte à moustaches\n", col_name),  # Titre sur deux lignes
          ylab = col_name,
          col = "lightblue",
          border = "darkblue",
          cex.main = 1.1,  # Taille du titre
          cex.lab = 1.1,  # Taille des étiquettes
          cex.axis = 0.9)  # Taille des axes
}
```

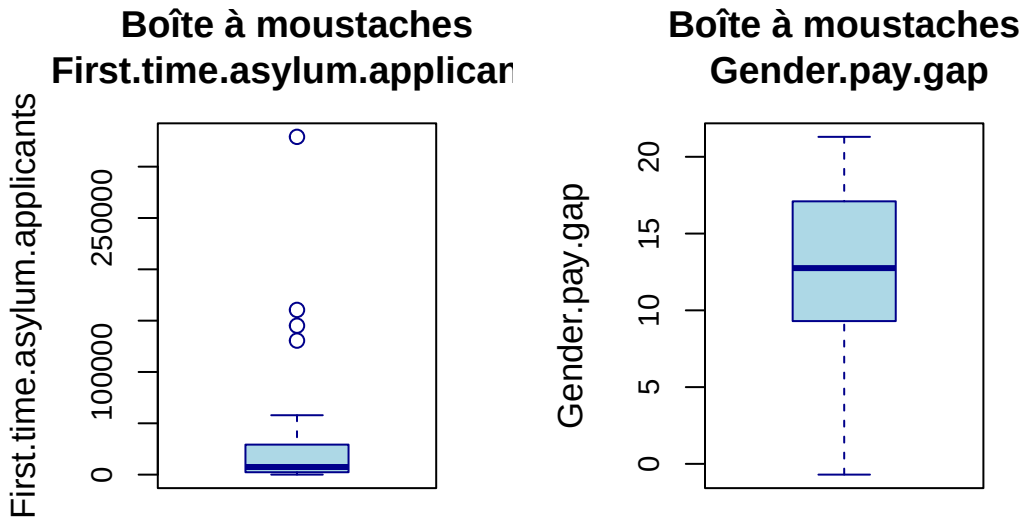


La distribution de la variable Population montre une grande variabilité entre les pays européens. Certains pays, comme l'Allemagne (84,3 millions), la France (68,1 millions) et l'Italie (58,9 millions), présentent des populations nettement supérieures à la majorité. Ces valeurs extrêmes contrastent avec des pays de petite taille comme Malte (0,54 million) et Chypre (0,92 million). La médiane, bien centrée dans la boîte du boxplot, reflète une répartition relativement équilibrée dans l'intervalle interquartile, bien que la présence de valeurs aberrantes, telles que celles de l'Allemagne et de la France, étende la distribution vers des valeurs élevées.

Quant à la variable Youth Population (pourcentage de jeunes âgés de 15 à 29 ans), elle présente une répartition beaucoup plus homogène. Les pourcentages varient de 13,2 % (Bulgarie) à 21,0 % (Islande), sans aucune valeur aberrante. Cependant, la médiane est située dans le quart inférieur de la boîte du boxplot, ce qui indique que les proportions de jeunes pour la majorité des pays se concentrent légèrement au-dessus de la médiane. On observe que les pays nordiques, comme l'Islande (21,0 %) et la Norvège (18,7 %), possèdent des proportions relativement élevées de jeunes, tandis que des pays comme la Bulgarie se situent à l'extrémité inférieure de cette répartition.

```r
selected_columns <- colnames(euro_data)[3:4]


par(mfrow = c(1, 2),
    oma = c(2, 2, 2, 2),
    mar = c(5, 5, 4, 2))


for (col_name in selected_columns) {
  boxplot(euro_data[[col_name]],
          main = paste("Boîte à moustaches\n", col_name),
```

```
          ylab = col_name,
          col = "lightblue",
          border = "darkblue",
          cex.main = 1.2,
          cex.lab = 1.1,
          cex.axis = 0.9)
}
```



La distribution de la variable First Time Asylum Applicants (premières demandes d'asile) montre une forte asymétrie, avec plusieurs valeurs extrêmes très élevées. Ces valeurs, représentées par des points au-dessus des moustaches, indiquent que quelques pays reçoivent un nombre disproportionné de premières demandes d'asile par rapport à la majorité. La médiane basse reflète que la moitié des pays ont un faible nombre de demandes d'asile, tandis que les valeurs élevées sont concentrées dans un petit nombre de pays, ce qui étend la moustache supérieure.

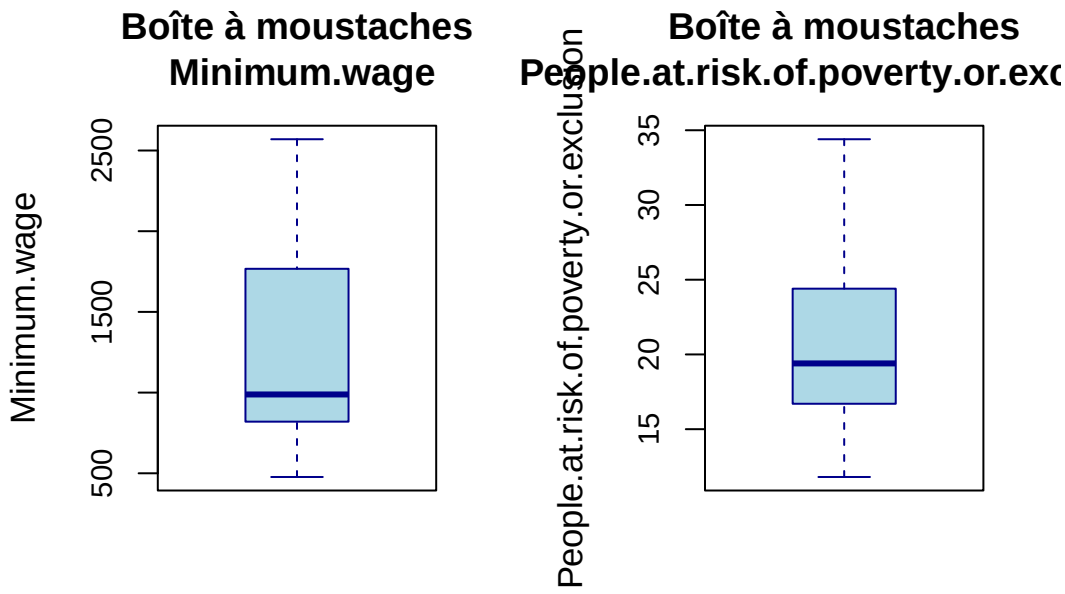
La variable Gender Pay Gap (écart de rémunération entre les sexes) présente une distribution symétrique et homogène. La médiane est bien centrée, et la boîte à moustaches indique que les valeurs sont concentrées dans une plage relativement étroite. Aucun outlier n'est visible, ce qui suggère que les écarts salariaux entre les sexes sont globalement similaires parmi les pays analysés. Interprétation

Les données brutes confirment les observations issues des diagrammes à moustaches. Concernant les First Time Asylum Applicants, on observe de fortes valeurs pour des pays comme l'Allemagne (0,32 million) et la France (0,14 million), qui reçoivent un grand nombre de demandes. À l'inverse, des pays comme la Slovaquie (0,0003 million) et la Hongrie (0,00003 million) présentent des valeurs bien en dessous de la moyenne, reflétant une disparité importante.

Pour la variable Gender Pay Gap, les données confirment l'homogénéité observée. Toutefois, on peut noter des différences significatives entre certains pays. Par exemple, l'Autriche (18,4) et la Suisse (17,9) affichent

des valeurs élevées, tandis que le Luxembourg présente une valeur négative (-0,7), indiquant une situation inverse inhabituelle.

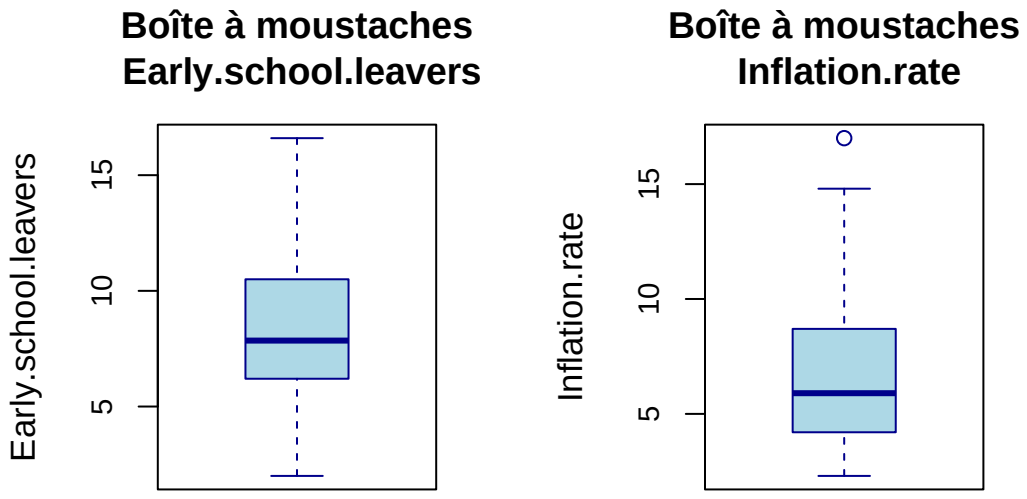
```
selected_columns <- colnames(euro_data)[5:6]

par(mfrow = c(1, 2),
    oma = c(2, 2, 2, 2),
    mar = c(5, 5, 4, 2))


for (col_name in selected_columns) {
  boxplot(euro_data[[col_name]],
          main = paste("Boîte à moustaches\n", col_name),
          ylab = col_name,
          col = "lightblue",
          border = "darkblue",
          cex.main = 1.2,
          cex.lab = 1.1,
          cex.axis = 0.9)
}
```



On remarque que les salaires varient de 500 à 2000 . La distribution de la variable Minimum Wage (salaire minimum) montre une dispersion modérée avec quelques valeurs basses qui se démarquent, visibles sous forme de moustaches allongées. La médiane, située dans la moitié inférieure de la boîte, indique que la moitié des pays ont un salaire minimum inférieur à cette valeur médiane. Aucune valeur aberrante significative n'est visible, ce qui reflète une relative homogénéité dans la répartition des salaires minimums parmi les pays étudiés.

La variable People at Risk of Poverty or Exclusion (personnes à risque de pauvreté ou d'exclusion sociale) présente une distribution relativement homogène. La médiane, légèrement orientée vers le bas de la boîte, indique que la moitié des pays ont des valeurs proches mais légèrement inférieures à la médiane. La taille modérée de la boîte et l'absence de valeurs aberrantes suggèrent que les pays étudiés affichent des niveaux de risque relativement similaires pour cet indicateur.

```
selected_columns <- colnames(euro_data)[7:8]


par(mfrow = c(1, 2),
    oma = c(2, 2, 2, 2),
    mar = c(5, 5, 4, 2))


for (col_name in selected_columns) {
  boxplot(euro_data[[col_name]],
          main = paste("Boîte à moustaches\n", col_name),
          ylab = col_name,
          col = "lightblue",
          border = "darkblue",
          cex.main = 1.2,
          cex.lab = 1.1,
          cex.axis = 0.9)
}
```

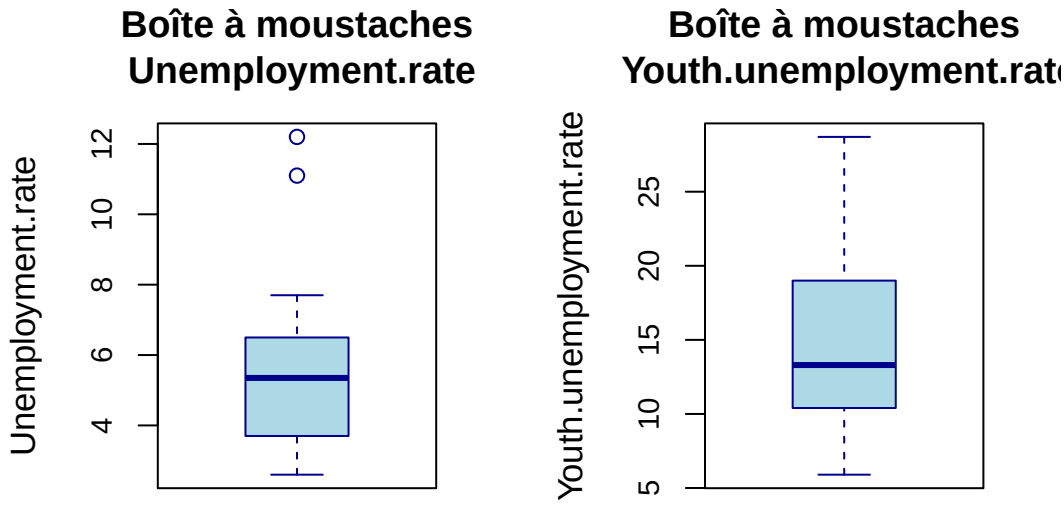


Early School Leavers (décrocheurs scolaires précoces) : La distribution de cette variable montre une répartition homogène, sans valeurs aberrantes significatives. La médiane, légèrement orientée vers le bas

de la boîte, indique que la moitié des pays ont des taux de décrochage scolaire légèrement inférieurs à la médiane. Les moustaches montrent que les taux de décrochage scolaire sont globalement compris dans une plage relativement étroite, reflétant une faible variabilité entre les pays.

Inflation Rate (taux d'inflation) : Cette variable présente une valeur aberrante notable, représentée par un point au-dessus des moustaches, ce qui indique qu'un ou quelques pays ont un taux d'inflation nettement supérieur à celui des autres. La boîte, asymétrique et légèrement orientée vers le bas, suggère que la majorité des pays ont des taux d'inflation relativement faibles, concentrés dans la moitié inférieure de la plage des données.

```
selected_columns <- colnames(euro_data)[9:10]


par(mfrow = c(1, 2),
    oma = c(2, 2, 2, 2),
    mar = c(5, 5, 4, 2))


for (col_name in selected_columns) {
  boxplot(euro_data[[col_name]],
          main = paste("Boîte à moustaches\n", col_name),
          ylab = col_name,
          col = "lightblue",
          border = "darkblue",
          cex.main = 1.2,
          cex.lab = 1.1,
          cex.axis = 0.9)
}
```

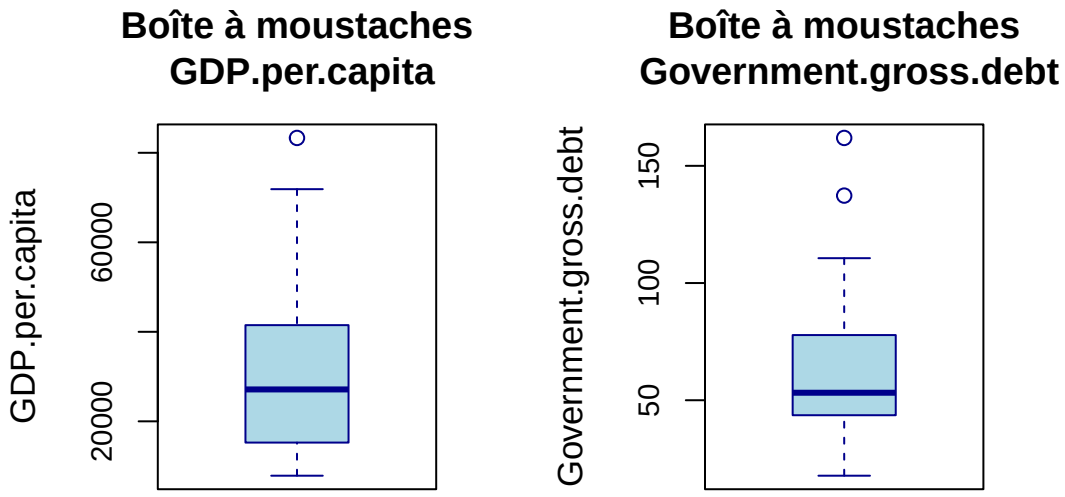


Unemployment Rate (taux de chômage) : La distribution de cette variable est globalement homogène, bien que deux valeurs aberrantes soient visibles au-dessus des moustaches. Ces valeurs traduisent des taux de chômage exceptionnellement élevés pour certains pays par rapport à la majorité. La médiane, légèrement au-dessus du centre de la boîte, indique que la moitié des pays ont des taux de chômage supérieurs ou égaux à la médiane, tandis que les 50 % inférieurs sont répartis sur une plage plus large, traduisant une légère asymétrie vers les valeurs supérieures

Pour la variable Youth Unemployment Rate (taux de chômage des jeunes), la distribution est plus dispersée, sans valeurs aberrantes. La boîte et les moustaches reflètent une variabilité importante entre les pays, traduisant des disparités régionales marquées. La médiane, située un peu en dessous du centre, indique une distribution légèrement asymétrique.

```
selected_columns <- colnames(euro_data)[11:12]


par(mfrow = c(1, 2),
    oma = c(2, 2, 2, 2),
    mar = c(5, 5, 4, 2))


for (col_name in selected_columns) {
  boxplot(euro_data[[col_name]],
          main = paste("Boîte à moustaches\n", col_name),
          ylab = col_name,
          col = "lightblue",
          border = "darkblue",
          cex.main = 1.2,
```

```
          cex.lab = 1.1,
          cex.axis = 0.9)
}
```

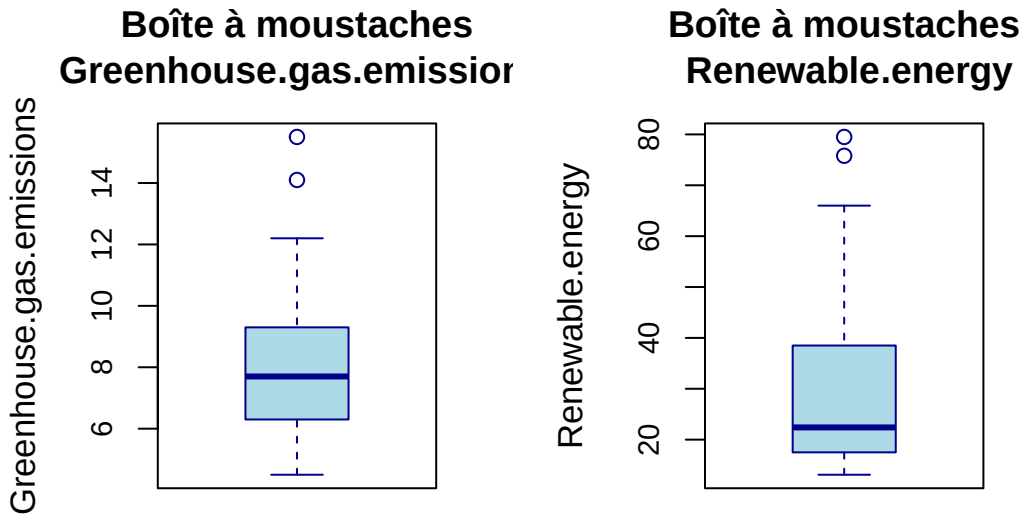


GDP per Capita (PIB par habitant) : La variable présente une distribution avec une valeur aberrante notable, correspondant à un pays dont le PIB par habitant est extrêmement élevé par rapport aux autres. La majorité des pays se situent dans une plage relativement restreinte, comme le montre la boîte. La médiane, bien centrée dans la boîte, indique une répartition relativement symétrique des valeurs autour de la médiane.

Government Gross Debt (dette brute du gouvernement en pourcentage du PIB) : La variable montre deux valeurs aberrantes au-dessus des moustaches, suggérant que certains pays ont des niveaux de dette exceptionnellement élevés. La médiane, légèrement en dessous du centre de la boîte, reflète une asymétrie vers les valeurs plus faibles, indiquant que les pays ayant des dettes inférieures ou proches de la médiane sont répartis sur une plage plus restreinte, tandis que ceux ayant des dettes supérieures sont davantage dispersés.

```
selected_columns <- colnames(euro_data)[13:14]


par(mfrow = c(1, 2),
    oma = c(2, 2, 2, 2),
    mar = c(5, 5, 4, 2))


for (col_name in selected_columns) {
  boxplot(euro_data[[col_name]],
          main = paste("Boîte à moustaches\n", col_name),
```

```
          ylab = col_name,
          col = "lightblue",
          border = "darkblue",
          cex.main = 1.2,
          cex.lab = 1.1,
          cex.axis = 0.9)
}
```

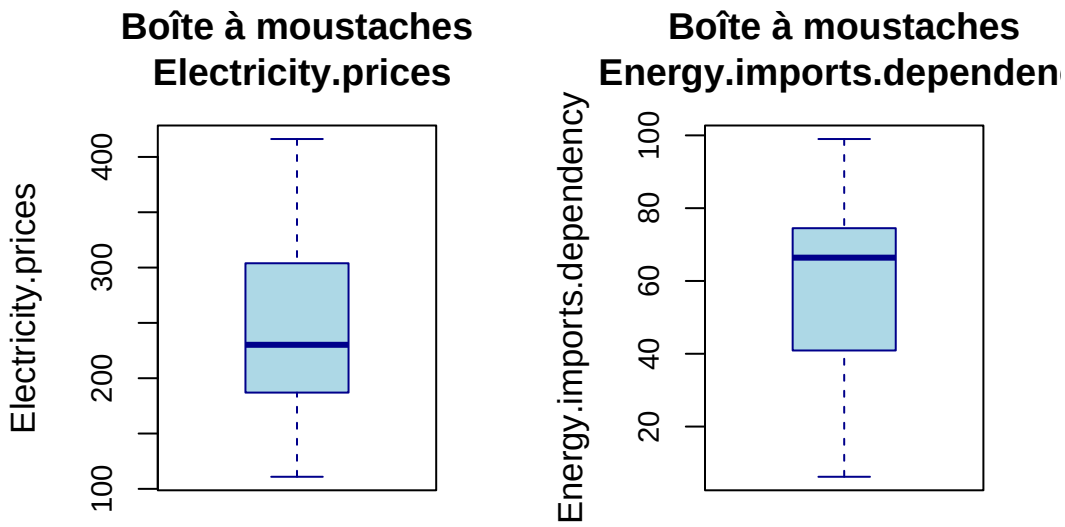


Greenhouse Gas Emissions (émissions de gaz à effet de serre) La variable présente deux valeurs aberrantes correspondant à des pays avec des émissions particulièrement élevées. La médiane, située au centre de la boîte, reflète une répartition relativement équilibrée des données. Cela suggère qu'environ la moitié des pays ont des émissions proches ou inférieures à la médiane, tandis que l'autre moitié a des émissions plus élevées.

Renewable Energy (énergies renouvelables) La variable montre également deux valeurs aberrantes pour des pays ayant une part particulièrement élevée d'énergies renouvelables. La médiane, située vers le bas de la boîte, indique qu'une proportion importante de pays a des parts d'énergies renouvelables faibles ou proches de cette valeur, tandis que l'autre moitié des pays a des parts plus élevées.

```
selected_columns <- colnames(euro_data)[15:16]


par(mfrow = c(1, 2),
    oma = c(2, 2, 2, 2),
    mar = c(5, 5, 4, 2))


for (col_name in selected_columns) {
```

```
  boxplot(euro_data[[col_name]],
          main = paste("Boîte à moustaches\n", col_name),
          ylab = col_name,
          col = "lightblue",
          border = "darkblue",
          cex.main = 1.2,
          cex.lab = 1.1,
          cex.axis = 0.9)
}
```



Electricity Prices (Prix de l'électricité) : La médiane est située dans le quart inférieur de la boîte, indiquant que la moitié des pays ont des prix de l'électricité inférieurs ou égaux à cette valeur médiane. L'absence de valeurs aberrantes reflète une distribution relativement homogène parmi les pays. La position basse de la médiane montre que les prix inférieurs à la médiane sont davantage concentrés, tandis que les prix supérieurs couvrent une plage plus étendue, traduisant une certaine variabilité.

Energy Imports Dependency (Dépendance aux importations d'énergie) : La médiane est située dans le quart supérieur de la boîte, ce qui signifie que la moitié des pays ont une dépendance énergétique inférieure ou égale à cette valeur. Contrairement à la variable précédente, cette distribution montre également une absence de valeurs aberrantes. La position élevée de la médiane reflète une plus grande dispersion des valeurs inférieures, tandis que les valeurs supérieures à la médiane sont relativement concentrées, indiquant une variabilité marquée dans la dépendance énergétique parmi les pays européens.

Pour apporter davantage de visibilité , nous allons remplacer les noms des variables des colonnes par des valeurs numériques, en nous référant à la légende suivante.

Légende des colonnes :

“[1]: Population
[2]: Youth.population
[3]: First.time.asylum.applicants
[4]: Gender.pay.gap
[5]: Minimum.wage
[6]: People.at.risk.of.poverty.or.exclusion
[7]: Early.school.leavers
[8]: Inflation.rate
[9]: Unemployment.rate
[10]: Youth.unemployment.rate
[11]: GDP.per.capita
[12]: Government.gross.debt
[13]: Greenhouse.gas.emissions
[14]: Renewable.energy
[15]: Electricity.prices
[16]: Energy.imports.dependency”

```r
# Étape 1 : Centrer les données (sans mise à l'échelle)
euro_data_centered <- scale(euro_data, center = TRUE, scale = FALSE)

# Étape 2 : Calculer la matrice de variance-covariance
n <- nrow(euro_data)
V <- (t(euro_data_centered) %*% euro_data_centered) / (n - 1)

# Étape 3 : Ajouter des indices numériques pour les colonnes et lignes
colnames(V) <- paste0("[", seq_len(ncol(V)), "]")
rownames(V) <- colnames(euro_data)

# Étape 4 : Arrondir les valeurs pour une meilleure lisibilité
V <- round(V, 4)

# Étape 5 : Ajuster les paramètres d'affichage pour éviter les retours à la ligne
options(width = 200)

# Étape 6 : Afficher la matrice en quatre parties
cat("Matrice de variance-covariance (indices entre crochets pour les colonnes) :\n")
```

```
Matrice de variance-covariance (indices entre crochets pour les colonnes) :
```

```r
cat("\nPartie 1 : Colonnes [1] à [4]\n")
```

```

Partie 1 : Colonnes [1] à [4]
```

```r
print(V[, 1:4], row.names = TRUE, right = FALSE)
```

```
                                        [1]           [2]           [3]           [4]
Population                     4.653257e+14 -6839267.0638  1.365717e+12 -6195038.4282
Youth.population              -6.839267e+06        3.2609 -1.592480e+04       -2.2307
First.time.asylum.applicants   1.365717e+12   -15924.8017  4.950971e+09    23416.4569
Gender.pay.gap                -6.195038e+06       -2.2307  2.341646e+04       25.6190
Minimum.wage                             NA            NA            NA            NA
```

```
People.at.risk.of.poverty.or.exclusion           NA            NA            NA           NA
Early.school.leavers                  1.621818e+07        1.5759  6.578850e+04       -1.1510
Inflation.rate                       -6.534739e+06       -2.6463 -5.499202e+04        5.9963
Unemployment.rate                     8.017485e+06       -0.8741  3.117897e+04       -0.8288
Youth.unemployment.rate               1.445936e+07       -2.2734  1.909011e+04       -9.0915
GDP.per.capita                       -2.857962e+10    23946.7345  3.054725e+07   -23296.9276
Government.gross.debt                 2.994623e+08       -2.1453  9.319273e+05      -30.4644
Greenhouse.gas.emissions             -8.092486e+06        1.8637 -1.295407e+04       -3.5834
Renewable.energy                     -1.066077e+08       10.1029 -2.707843e+05       23.8361
Electricity.prices                    7.486738e+08       -0.8086  2.853515e+06       -0.5714
Energy.imports.dependency             6.282654e+07        4.0745  3.598142e+05      -34.9379
```

```
cat("\nPartie 2 : Colonnes [5] à [8]\n")
```

```
Partie 2 : Colonnes [5] à [8]
```

```
print(V[, 5:8], row.names = TRUE, right = FALSE)
```

```
                                      [5] [6]           [7]           [8]
Population                             NA  NA 16218178.4517 -6534739.2138
Youth.population                       NA  NA        1.5759       -2.6463
First.time.asylum.applicants           NA  NA    65788.5000   -54992.0172
Gender.pay.gap                         NA  NA       -1.1510        5.9963
Minimum.wage                           NA  NA            NA            NA
People.at.risk.of.poverty.or.exclusion NA  NA            NA            NA
Early.school.leavers                   NA  NA       12.5517        0.7159
Inflation.rate                         NA  NA        0.7159       12.1161
Unemployment.rate                      NA  NA        0.0483       -2.5167
Youth.unemployment.rate                NA  NA        0.6197       -3.7521
GDP.per.capita                         NA  NA    -1667.7931   -40170.7103
Government.gross.debt                  NA  NA        6.2831      -21.8946
Greenhouse.gas.emissions               NA  NA        0.1055       -0.4309
Renewable.energy                       NA  NA       19.0807       -9.6309
Electricity.prices                     NA  NA      -44.9041      -99.8614
Energy.imports.dependency              NA  NA      -27.2752      -28.8008
```

```
cat("\nPartie 3 : Colonnes [9] à [12]\n")
```

```
Partie 3 : Colonnes [9] à [12]
```

```
print(V[, 9:12], row.names = TRUE, right = FALSE)
```

```
                                         [9]          [10]          [11]          [12]
Population                       8017484.9552 14459359.1397 -2.857962e+10  2.994623e+08
Youth.population                       -0.8741       -2.2734  2.394673e+04 -2.145300e+00
First.time.asylum.applicants        31178.9655    19090.1121  3.054725e+07  9.319273e+05
Gender.pay.gap                         -0.8288       -9.0915 -2.329693e+04 -3.046440e+01
Minimum.wage                                NA            NA            NA            NA
People.at.risk.of.poverty.or.exclusion      NA            NA            NA            NA
```

```
Early.school.leavers                    0.0483       0.6197 -1.667793e+03  6.283100e+00
Inflation.rate                         -2.5167      -3.7521 -4.017071e+04 -2.189460e+01
Unemployment.rate                       5.0494      11.5647 -8.682110e+03  4.358010e+01
Youth.unemployment.rate                11.5647      34.5511 -2.704445e+04  1.032097e+02
GDP.per.capita                      -8682.1103  -27044.4517  3.983704e+08 -1.581849e+05
Government.gross.debt                  43.5801     103.2097 -1.581849e+05  1.240830e+03
Greenhouse.gas.emissions               -1.9539      -3.9530  2.158401e+04 -1.137270e+01
Renewable.energy                       -0.8711     -14.7666  1.228200e+05 -1.327777e+02
Electricity.prices                     18.5154     -30.1509  2.644578e+05  6.659872e+02
Energy.imports.dependency               7.1974      15.1315  5.860834e+04  2.703910e+02
```

```
cat("\nPartie 4 : Colonnes [13] à [16]\n")
```

```
Partie 4 : Colonnes [13] à [16]
```

```
print(V[, 13:16], row.names = TRUE, right = FALSE)
```

```
                                        [13]          [14]          [15]          [16]
Population                      -8092485.6885 -1.066077e+08  7.486738e+08 62826536.5937
Youth.population                        1.8637  1.010290e+01 -8.086000e-01        4.0745
First.time.asylum.applicants       -12954.0690 -2.707843e+05  2.853515e+06   359814.2500
Gender.pay.gap                          -3.5834  2.383610e+01 -5.714000e-01      -34.9379
Minimum.wage                                 NA            NA            NA            NA
People.at.risk.of.poverty.or.exclusion       NA            NA            NA            NA
Early.school.leavers                    0.1055  1.908070e+01 -4.490410e+01      -27.2752
Inflation.rate                         -0.4309 -9.630900e+00 -9.986140e+01      -28.8008
Unemployment.rate                      -1.9539 -8.711000e-01  1.851540e+01        7.1974
Youth.unemployment.rate                -3.9530 -1.476660e+01 -3.015090e+01       15.1315
GDP.per.capita                      21584.0069  1.228200e+05  2.644578e+05    58608.3414
Government.gross.debt                 -11.3727 -1.327777e+02  6.659872e+02      270.3910
Greenhouse.gas.emissions                7.0331 -2.733100e+00  1.906030e+01       -3.6866
Renewable.energy                       -2.7331  3.593812e+02 -4.262927e+02     -290.3213
Electricity.prices                     19.0603 -4.262927e+02  6.935113e+03      741.6931
Energy.imports.dependency              -3.6866 -2.903213e+02  7.416931e+02      592.5486
```

Légende des colonnes :

“[1]: Population
[2]: Youth.population
[3]: First.time.asylum.applicants
[4]: Gender.pay.gap
[5]: Minimum.wage
[6]: People.at.risk.of.poverty.or.exclusion
[7]: Early.school.leavers
[8]: Inflation.rate
[9]: Unemployment.rate
[10]: Youth.unemployment.rate
[11]: GDP.per.capita
[12]: Government.gross.debt
[13]: Greenhouse.gas.emissions
[14]: Renewable.energy

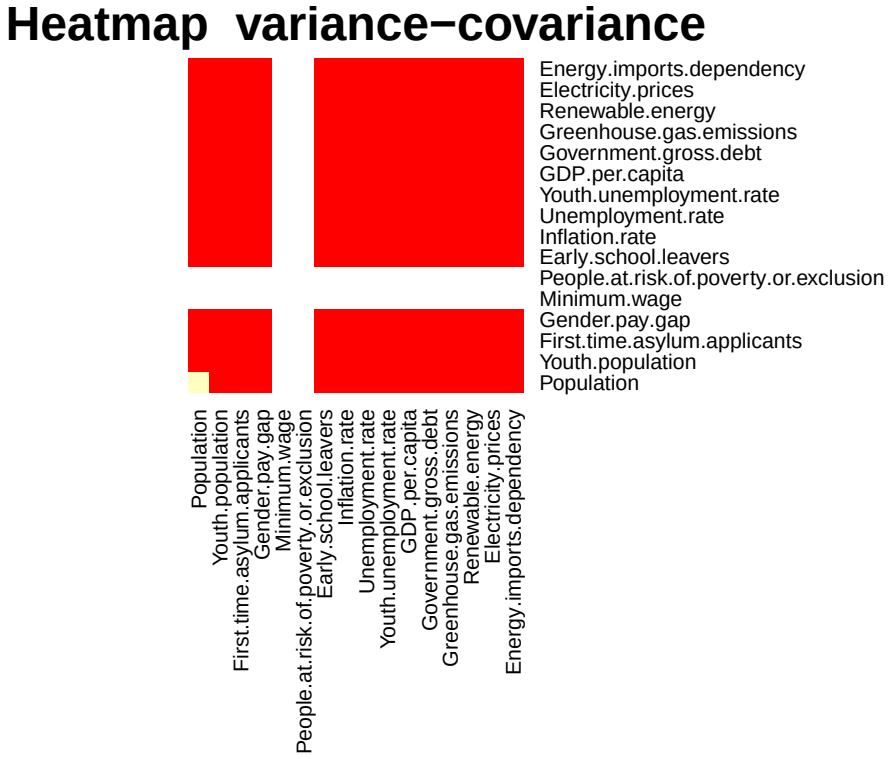
[15]: Electricity.prices
[16]: Energy.imports.dependency"

La matrice de variance-covariance V calcule la dispersion des variables ainsi que les relations linéaires entre elles. Les valeurs sur la diagonale principale représentent la variance de chaque variable. Par exemple, la variance de Population est très élevée (4.653257e+14), reflétant une grande dispersion entre les pays, comme cela a été observé dans les diagrammes à moustaches. En revanche, la variance de Gender Pay Gap est beaucoup plus faible (25.6190), confirmant l'homogénéité relative notée précédemment, bien qu'elle reste significative.

Une covariance négative entre Renewable Energy et Energy Imports Dependency (−290.3213) reflète une relation inverse plus marquée : les pays avec une part plus élevée d'énergies renouvelables, comme l'Islande et la Norvège, sont généralement moins dépendants des importations d'énergie.

La matrice V enrichit l'analyse exploratoire initiale en fournissant une quantification précise des dispersions et interactions entre variables. Ces résultats constituent une base solide pour des analyses plus avancées, comme l'analyse en composantes principales (ACP), tout en préparant la transition vers l'étude des relations normalisées via la matrice de corrélation.

```r
#3
V_rounded <- round(V, 2)
heatmap(as.matrix(V_rounded),
        main = "Heatmap  variance-covariance",
        Colv = NA, Rowv = NA,  # Désactive le clustering
        scale = "none",  # Pas de normalisation supplémentaire
        col = heat.colors(10),  # Palette de couleurs
        margins = c(15, 15),  # Ajuster les marges
        labCol = colnames(euro_data),  # Noms originaux des colonnes
        labRow = rownames(V_rounded),  # Noms des lignes
        cexCol = 0.8,  # Taille des labels des colonnes
        cexRow = 0.8)  # Taille des labels des lignes
```



Comme les données ne sont pas normalisées, chaque variable est utilisée avec son échelle propre, ce qui limite la possibilité de faire des interprétations précises quant aux variances et covariances. En effet, des échelles différentes rendent difficile la comparaison directe des intensités entre les variables.

Malgré cela, on peut remarquer que la variable Population se distingue par une coloration plus marquée de sa variance, indiquant une forte variation au sein de cette variable. Les zones blanches représentent des valeurs non définies ou nulles, tandis que pour le reste la couleur rouge ne nous permets pas de discerner de différences significatives entre les covariances des autres variables.

Bien que cette matrice de variance-covariance fournisse des informations limitées en profondeur, elle reste utile pour donner une vue d'ensemble des relations potentielles entre les variables. Cependant, pour extraire davantage d'informations et permettre des comparaisons cohérentes entre les variables, il est préférable d'utiliser la matrice de corrélation, qui repose sur des données normalisées.

```r
# 4
# Étape 1 : Calculer la matrice de corrélation
correlation_matrix <- cor(euro_data, use = "complete.obs")

# Étape 2 : Arrondir les valeurs à deux décimales pour une meilleure lisibilité
correlation_matrix_rounded <- round(correlation_matrix, 2)

# Étape 3 : Ajouter des indices numériques pour les colonnes et les lignes
colnames(correlation_matrix_rounded) <- paste0("[", seq_len(ncol(correlation_matrix_rounded)), "]")
rownames(correlation_matrix_rounded) <- paste0("[", seq_len(nrow(correlation_matrix_rounded)), "]")

# Étape 4 : Ajuster les paramètres pour l'affichage
options(width = 180)  # Ajuste la largeur pour afficher toutes les colonnes sans retour à la ligne
```

```r
cat("Matrice de corrélation (indices pour les lignes et colonnes) :\n")
```

```
Matrice de corrélation (indices pour les lignes et colonnes) :
```

```r
# Étape 5 : Afficher la matrice sous forme de tableau compact
print(correlation_matrix_rounded, row.names = TRUE, right = FALSE)
```

```
       [1]   [2]   [3]   [4]   [5]   [6]   [7]   [8]   [9]  [10]  [11]  [12]  [13]  [14]  [15]  [16]
[1]   1.00 -0.11  0.90 -0.06  0.25  0.13  0.32 -0.11  0.14  0.08  0.01  0.41 -0.13 -0.22  0.40  0.04
[2]  -0.11  1.00 -0.07 -0.25  0.63 -0.43  0.03 -0.52 -0.15 -0.15  0.71 -0.06  0.30 -0.18  0.23  0.49
[3]   0.90 -0.07  1.00  0.07  0.34  0.16  0.36 -0.25  0.19  0.03  0.10  0.38 -0.06 -0.16  0.48  0.16
[4]  -0.06 -0.25  0.07  1.00 -0.30 -0.14 -0.01  0.43 -0.05 -0.29 -0.37 -0.12 -0.24  0.43 -0.04 -0.40
[5]   0.25  0.63  0.34 -0.30  1.00 -0.35 -0.10 -0.63 -0.09 -0.19  0.90  0.03  0.51 -0.13  0.43  0.37
[6]   0.13 -0.43  0.16 -0.14 -0.35  1.00  0.43 -0.04  0.43  0.45 -0.32  0.04 -0.28  0.09 -0.18 -0.24
[7]   0.32  0.03  0.36 -0.01 -0.10  0.43  1.00  0.01  0.12  0.14 -0.13 -0.02 -0.28  0.13  0.02 -0.15
[8]  -0.11 -0.52 -0.25  0.43 -0.63 -0.04  0.01  1.00 -0.38 -0.25 -0.58 -0.29 -0.16 -0.07 -0.38 -0.42
[9]   0.14 -0.15  0.19 -0.05 -0.09  0.43  0.12 -0.38  1.00  0.87 -0.12  0.62 -0.28  0.28  0.05  0.09
[10]  0.08 -0.15  0.03 -0.29 -0.19  0.45  0.14 -0.25  0.87  1.00 -0.15  0.57 -0.20  0.06 -0.12  0.09
[11]  0.01  0.71  0.10 -0.37  0.90 -0.32 -0.13 -0.58 -0.12 -0.15  1.00 -0.12  0.61 -0.17  0.36  0.42
[12]  0.41 -0.06  0.38 -0.12  0.03  0.04 -0.02 -0.29  0.62  0.57 -0.12  1.00 -0.34 -0.11  0.27  0.37
[13] -0.13  0.30 -0.06 -0.24  0.51 -0.28 -0.28 -0.16 -0.28 -0.20  0.61 -0.34  1.00 -0.28  0.25  0.08
[14] -0.22 -0.18 -0.16  0.43 -0.13  0.09  0.13 -0.07  0.28  0.06 -0.17 -0.11 -0.28  1.00  0.00 -0.54
[15]  0.40  0.23  0.48 -0.04  0.43 -0.18  0.02 -0.38  0.05 -0.12  0.36  0.27  0.25  0.00  1.00  0.19
[16]  0.04  0.49  0.16 -0.40  0.37 -0.24 -0.15 -0.42  0.09  0.09  0.42  0.37  0.08 -0.54  0.19  1.00
```

Légende des colonnes :

"[1]: Population
[2]: Youth.population
[3]: First.time.asylum.applicants
[4]: Gender.pay.gap
[5]: Minimum.wage
[6]: People.at.risk.of.poverty.or.exclusion
[7]: Early.school.leavers
[8]: Inflation.rate
[9]: Unemployment.rate
[10]: Youth.unemployment.rate
[11]: GDP.per.capita
[12]: Government.gross.debt
[13]: Greenhouse.gas.emissions
[14]: Renewable.energy
[15]: Electricity.prices
[16]: Energy.imports.dependency"

La matrice de corrélation offre une vision normalisée des relations linéaires entre les variables, avec des coefficients compris entre -1 et 1. Une corrélation positive modérée est observée entre Population et Youth Population (r=0.87), indiquant que les pays avec une population élevée tendent à avoir une proportion importante de jeunes.

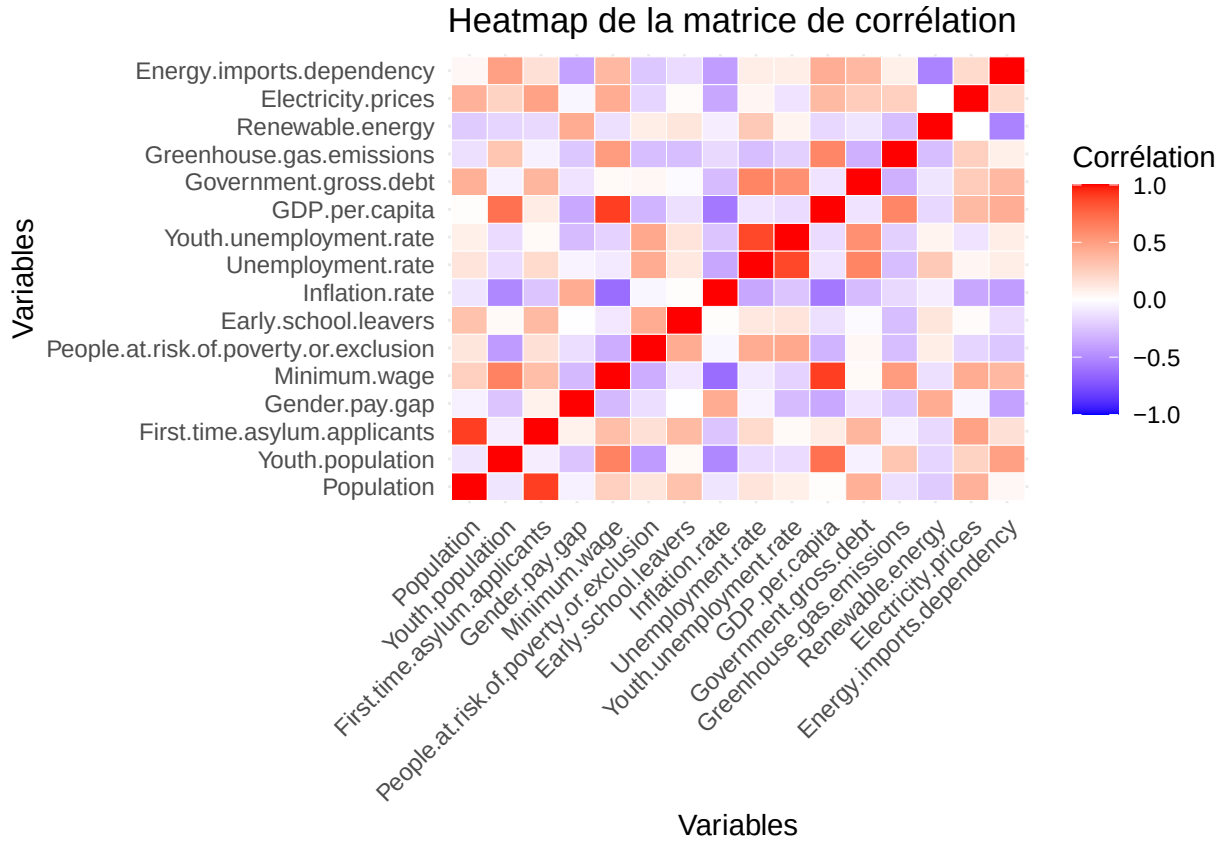
Une relation négative significative est également visible entre Renewable Energy et Energy Imports Dependency (r=−0.54), confirmant que les pays intégrant davantage d'énergies renouvelables sont généralement moins dépendants des importations énergétiques.

De plus, une corrélation très forte (r=0.87) entre Youth Unemployment Rate et Unemployment Rate montre une relation directe entre ces deux indicateurs. En revanche, certaines variables comme Gender Pay Gap et

Minimum Wage (r=−0.05) affichent une corrélation quasi nulle, suggérant une absence de relation linéaire significative. Cette matrice met ainsi en évidence les liens les plus marqués tout en soulignant les variables peu ou pas reliées entre elles.

```
# Transformer la matrice de corrélation en format long
correlation_long <- reshape2::melt(round(correlation_matrix, 2))
```

```
# Heatmap de la matrice de corrélation
ggplot(correlation_long, aes(Var1, Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1, 1),
                       name = "Corrélation") +
  theme_minimal() +
  labs(title = "Heatmap de la matrice de corrélation",
       x = "Variables",
       y = "Variables") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



La heatmap de la matrice de corrélation illustre les relations linéaires entre les différentes variables. Les couleurs rouges indiquent des corrélations positives élevées, tandis que les teintes bleues traduisent des corrélations négatives significatives. Les zones blanches ou pâles, quant à elles, reflètent des relations faibles ou inexistantes.

Par exemple, on remarque une forte corrélation positive entre Population et First Time Asylum Applicants, suggérant que les pays avec une population élevée attirent davantage de demandes d'asile. De même, une

forte corrélation existe entre GDP per Capita et Minimum Wage, montrant que les pays avec un PIB élevé ont tendance à offrir des salaires minimums plus élevés.

Une corrélation négative marquée est visible entre Renewable Energy et Energy Imports Dependency, mettant en évidence l'impact positif des énergies renouvelables sur la réduction de la dépendance énergétique. Cela reflète probablement une autonomie énergétique accrue dans les pays intégrant davantage d'énergies renouvelables.

À l'inverse, certaines variables montrent peu ou pas de corrélation, traduisant une absence de relation linéaire notable. Par exemple, Gender Pay Gap et Minimum Wage, ou encore Electricity Prices et Early School Leavers, n'affichent aucune relation significative, ce qui indique que l'évolution de l'une n'apporte aucune information sur l'autre.

Enfin, ce graphique offre une visualisation claire des relations fortes et faibles entre les variables, facilitant l'identification des interactions les plus significatives pour l'analyse. Il permet également de mettre en lumière les variables qui évoluent de manière indépendante les unes des autres.Ce qui n'était pas possible avec la matrice de variance-covariance

```
# 5

euro_data_replace_na <- apply(euro_data, 2, function(x) ifelse(is.na(x), mean(x, na.rm = TRUE), x))
```

```
# 5.
# Calculer les composantes principales
res <- prcomp(euro_data_replace_na, scale = TRUE, center = TRUE)

# Créer un fichier PDF pour les résultats
pdf("resultats_prcomp.pdf", width = 10, height = 8)

# Vérification de l'orthogonalité et des normes
cat("Produit scalaire des vecteurs propres (orthogonalité) :\n")
```

```
Produit scalaire des vecteurs propres (orthogonalité) :
```

```
orthogonality_check <- t(res$rotation) %*% res$rotation
print(round(orthogonality_check, 2))
```

```
     PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9 PC10 PC11 PC12 PC13 PC14 PC15 PC16
PC1    1   0   0   0   0   0   0   0   0    0    0    0    0    0    0    0
PC2    0   1   0   0   0   0   0   0   0    0    0    0    0    0    0    0
PC3    0   0   1   0   0   0   0   0   0    0    0    0    0    0    0    0
PC4    0   0   0   1   0   0   0   0   0    0    0    0    0    0    0    0
PC5    0   0   0   0   1   0   0   0   0    0    0    0    0    0    0    0
PC6    0   0   0   0   0   1   0   0   0    0    0    0    0    0    0    0
PC7    0   0   0   0   0   0   1   0   0    0    0    0    0    0    0    0
PC8    0   0   0   0   0   0   0   1   0    0    0    0    0    0    0    0
PC9    0   0   0   0   0   0   0   0   1    0    0    0    0    0    0    0
PC10   0   0   0   0   0   0   0   0   0    1    0    0    0    0    0    0
PC11   0   0   0   0   0   0   0   0   0    0    1    0    0    0    0    0
PC12   0   0   0   0   0   0   0   0   0    0    0    1    0    0    0    0
PC13   0   0   0   0   0   0   0   0   0    0    0    0    1    0    0    0
PC14   0   0   0   0   0   0   0   0   0    0    0    0    0    1    0    0
PC15   0   0   0   0   0   0   0   0   0    0    0    0    0    0    1    0
PC16   0   0   0   0   0   0   0   0   0    0    0    0    0    0    0    1
```

```r
cat("Normes des vecteurs propres (1 attendu) :\n")
```

```
Normes des vecteurs propres (1 attendu) :
```

```r
norms <- apply(res$rotation, 2, function(col) sqrt(sum(col^2)))
print(round(norms, 2))
```

```
 PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9 PC10 PC11 PC12 PC13 PC14 PC15 PC16
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
```

```r
# Afficher les premières coordonnées dans la nouvelle base (divisé en deux parties)
cat("\nPremières coordonnées des observations (Colonnes [1] à [8]) :\n")
```

```
Premières coordonnées des observations (Colonnes [1] à [8]) :
```

```r
print(head(res$x[, 1:8], n = 6))  # Affichage des colonnes 1 à 8
```

```
           PC1         PC2         PC3         PC4        PC5        PC6        PC7          PC8
[1,]  0.8930784 -0.05981786 -0.97098080 -0.32722747  0.9238143 -0.4927803  0.1009824  0.651241168
[2,]  1.7565034 -2.16393542 -0.07834658  1.31816585 -0.2325251 -0.1199267 -0.3036975  0.514094776
[3,] -2.6417622  1.77214093 -0.25429763 -0.23821400 -1.6690826  1.9269231 -0.1860566  0.006445367
[4,] -1.4323209  0.71295161 -0.13114739  1.72626176  0.5961859 -0.0925010 -0.1856331 -0.928931619
[5,]  1.2302770 -0.94486199  0.27801253  1.08399558 -0.3331520 -1.0172092  0.4715290  1.995736250
[6,] -0.2887514  2.35926409 -2.92285398  0.04385271 -0.1945308 -0.9634055 -1.3480963  0.346571689
```

```r
cat("\nPremières coordonnées des observations (Colonnes [9] à [16]) :\n")
```

```
Premières coordonnées des observations (Colonnes [9] à [16]) :
```

```r
print(head(res$x[, 9:16], n = 6))  # Affichage des colonnes 9 à 16
```

```
           PC9       PC10       PC11       PC12       PC13        PC14        PC15        PC16
[1,]  0.6247011 -0.2092549  0.2453571  0.5611628  0.4778485  0.29771859 -0.17539637 -0.14631681
[2,] -1.1343872 -0.1550909 -0.6500718  0.4742277  0.3592356 -0.10884273 -0.25695322  0.09789929
[3,]  0.4886804 -1.1327358 -0.1528996 -0.3188954  0.1384509 -0.30770441  0.12484341  0.06258464
[4,]  0.1125664 -0.3373321 -0.2422261 -0.2552262 -0.5752980  0.25874542 -0.35628808  0.18537500
[5,] -0.3374791  0.6185508 -0.1764931 -0.9572414 -0.4161035 -0.02523226  0.02481071  0.02486392
[6,] -0.3554556  0.7335514  0.3869734 -0.1430733  0.1021369 -0.16528975  0.15868235  0.08522987
```

```r
# Ajouter le biplot au PDF
#biplot(res, scale = 0, main = "Biplot des composantes principales")

# Fermer le fichier PDF
#dev.off()
```

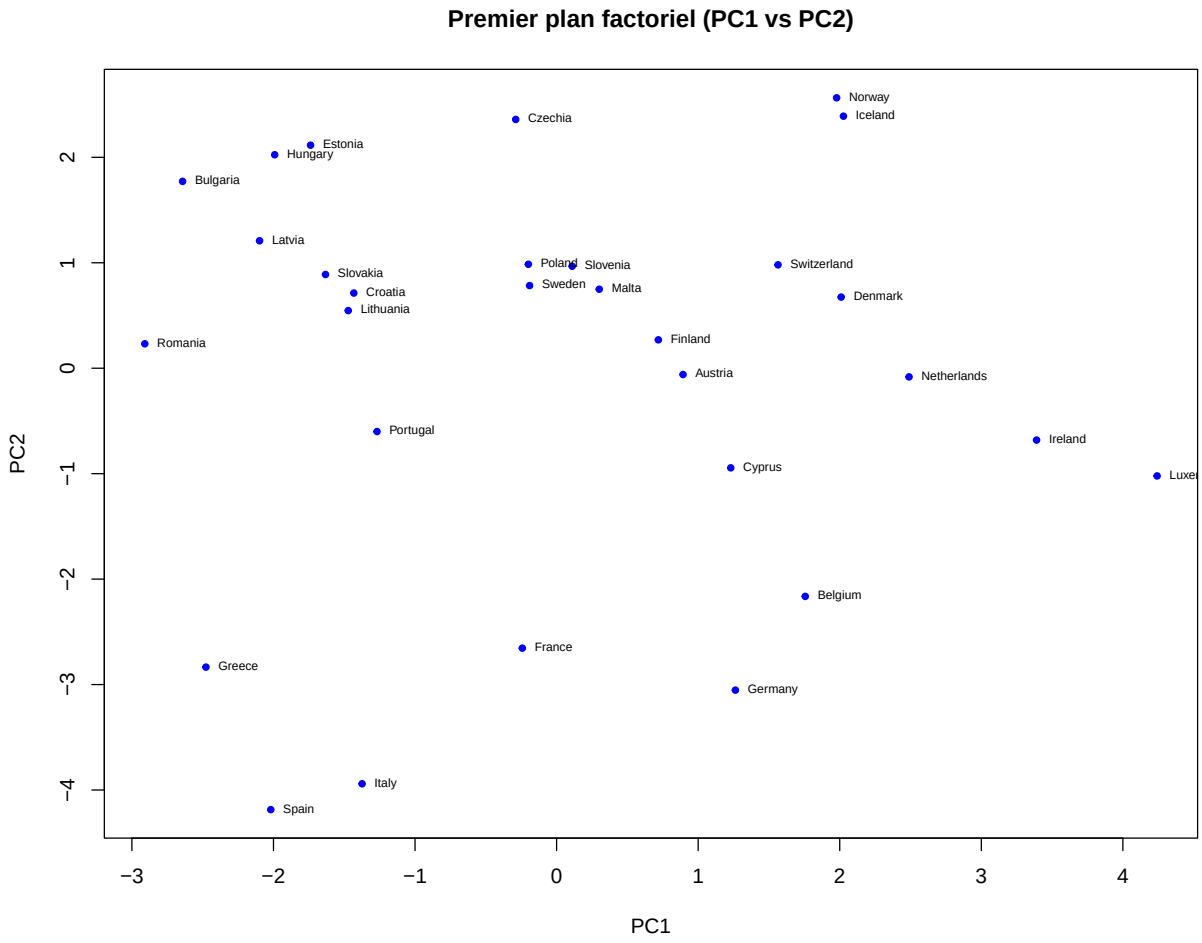
Les composantes principales forment une base orthonormée, ce qui est prouvé numériquement. Le produit scalaire des vecteurs propres (matrice de rotation transposée multipliée par elle-même) donne une matrice diagonale avec des valeurs de 1 sur la diagonale et 0 ailleurs, confirmant leur orthogonalité. Les normes des vecteurs propres, calculées comme la racine carrée de la somme des carrés des coefficients, sont toutes égales à 1, prouvant qu'ils sont normalisés.

Le paramètre center = TRUE recentre chaque variable en soustrayant sa moyenne, garantissant que les composantes principales sont calculées par rapport à un centre des données égal à zéro. Le paramètre scale = TRUE met chaque variable à l'échelle en divisant par son écart type, standardisant ainsi les variables pour qu'elles aient toutes une variance de 1, ce qui est crucial lorsque les variables ont des échelles différentes.

Sans center, les composantes principales seraient biaisées par des variables aux moyennes élevées, faussant leur interprétation. Sans scale, les variables avec des échelles ou des variances élevées domineraient les calculs, influençant de manière disproportionnée les composantes principales. En combinant center et scale, chaque variable contribue de manière équitable à la définition des composantes principales.

```r
#6
# Étape 1 : Calculer les composantes principales (si non déjà fait)
euro_data_replace_na <- apply(euro_data, 2, function(x) ifelse(is.na(x), mean(x, na.rm = TRUE), x))
res <- prcomp(euro_data_replace_na, scale = TRUE, center = TRUE)
```

```r
#6.
coord <- res$x

# Étape 3 : Ajouter les noms des pays comme noms de lignes
tmp <- read.csv("data/euro.csv", header = TRUE, sep = ";")
rownames(coord) <- tmp[, 1]

# Étape 4 : Créer le graphique du premier plan factoriel
plot(coord[, 1], coord[, 2],
     xlab = "PC1", ylab = "PC2",
     main = "Premier plan factoriel (PC1 vs PC2)",
     pch = 20, col = "blue")

# Étape 5 : Ajouter les noms des pays sur le graphique
text(coord[, 1], coord[, 2], labels = rownames(coord), pos = 4, cex = 0.6)
```



Le graphique du premier plan factoriel (PC1 vs PC2) illustre la position relative des pays selon les deux premières composantes principales, qui capturent la majorité de la variance des données. Ces composantes principales permettent de résumer l'ensemble des variables tout en conservant au mieux l'information.

On observe une forte opposition sur l'axe 1 entre des pays comme la France et la République tchèque (Czechia). Cette opposition reflète d'importantes dissimilitudes entre ces deux pays, le premier axe représentant le maximum de variance des données. De manière similaire, l'Allemagne se situe en bas à droite et contraste fortement avec des pays comme la Bulgarie et la Hongrie, positionnés en haut à gauche, traduisant des différences marquées entre ces pays.

Sur l'axe 2, le Luxembourg et la Roumanie occupent des positions opposées, indiquant également une forte dissimilarité entre eux sur les variables résumées par cette composante. Cet axe capture une partie supplémentaire de la variance non expliquée par le premier, apportant une nouvelle perspective dans l'analyse.

Des regroupements centraux sont également visibles, formant un cluster de pays comme le Portugal, la Slovénie, la Suède et Malte. Cette proximité traduit une certaine similarité entre ces pays, du point de vue de l'ensemble des variables étudiées.

Par ailleurs, certains pays se distinguent nettement par leur éloignement des autres :

La Norvège et l'Islande se démarquent dans la partie supérieure droite, probablement en raison de leur indépendance énergétique et de leur forte part d'énergies renouvelables. À l'opposé, la Grèce, l'Espagne et l'Italie apparaissent isolées dans la partie inférieure gauche, ce qui peut s'expliquer par leurs spécificités économiques ou sociales.

En somme, ce type de visualisation permet d'identifier rapidement les regroupements et les oppositions entre

pays, offrant une base précieuse pour des analyses comparatives plus approfondies.

```
# Étape 1 : Calculer les variances des composantes principales
variances <- res$sdev^2

# Étape 2 : Calculer le pourcentage de variance expliquée
pourcentages <- variances / sum(variances) * 100

# Étape 3 : Calculer la variance expliquée cumulée
cumul <- cumsum(pourcentages)

# Étape 4 : Créer une table avec les informations organisées
variance_table <- data.frame(
  Composante = paste0("PC", seq_along(variances)),  # Étiquettes des composantes
  `Valeur propre` = round(variances, 2),
  `% Variance expliquée` = round(pourcentages, 2),
  `% Variance cumulée` = round(cumul, 2)
)

# Étape 5 : Ajuster les paramètres pour l'affichage
options(width = 100)  # Largeur de l'affichage

# Étape 6 : Afficher la table sous forme bien formatée
cat("Table des variances expliquées :\n")
```

```
Table des variances expliquées :
```

```
print(variance_table, row.names = FALSE)
```

```
 Composante Valeur.propre X..Variance.expliquée X..Variance.cumulée
        PC1          3.64                 22.72                22.72
        PC2          3.52                 22.02                44.74
        PC3          2.03                 12.72                57.46
        PC4          1.86                 11.60                69.06
        PC5          1.28                  8.02                77.08
        PC6          0.87                  5.43                82.51
        PC7          0.75                  4.70                87.21
        PC8          0.56                  3.50                90.70
        PC9          0.45                  2.80                93.51
       PC10          0.35                  2.16                95.66
       PC11          0.21                  1.28                96.95
       PC12          0.17                  1.05                98.00
       PC13          0.16                  0.99                98.99
       PC14          0.09                  0.56                99.55
       PC15          0.04                  0.26                99.81
       PC16          0.03                  0.19               100.00
```

```
#7.
variances <- res$sdev^2

# Calculer le pourcentage de variance expliquée
pourcentages <- variances / sum(variances) * 100
```

```r
# Calculer la variance expliquée cumulée
cumul <- cumsum(pourcentages)

# Afficher les informations pour contrôle
cat("Valeurs propres :\n")
```

```
Valeurs propres :
```

```r
print(round(variances, 2))
```

```
 [1] 3.64 3.52 2.03 1.86 1.28 0.87 0.75 0.56 0.45 0.35 0.21 0.17 0.16 0.09 0.04 0.03
```

```r
cat("\nPourcentages de variances expliquées :\n")
```

```

Pourcentages de variances expliquées :
```

```r
print(round(pourcentages, 2))
```

```
 [1] 22.72 22.02 12.72 11.60  8.02  5.43  4.70  3.50  2.80  2.16  1.28  1.05  0.99  0.56  0.26  0.19
```

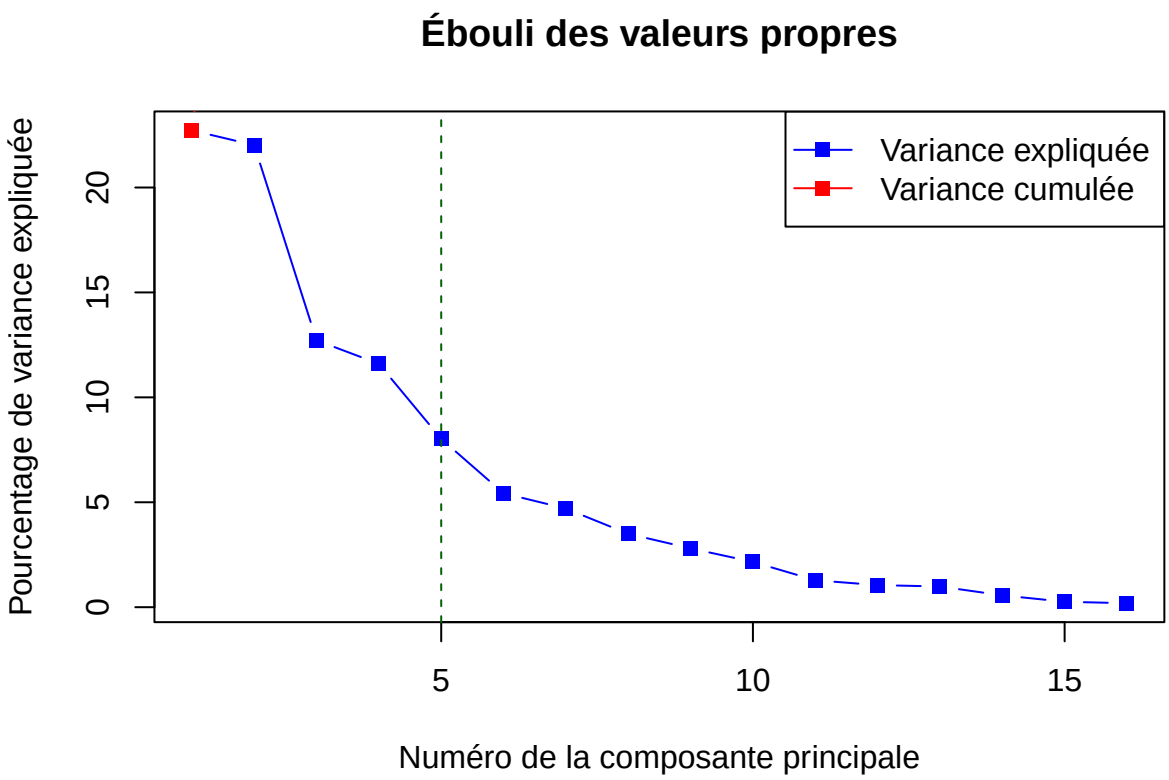
```r
cat("\nVariance expliquée cumulée :\n")
```

```

Variance expliquée cumulée :
```

```r
print(round(cumul, 2))
```

```
 [1]  22.72  44.74  57.46  69.06  77.08  82.51  87.21  90.70  93.51  95.66  96.95  98.00  98.99
[14]  99.55  99.81 100.00
```

```r
# Créer un graphique combiné
plot(pourcentages, type = "b", pch = 15, col = "blue",
     xlab = "Numéro de la composante principale",
     ylab = "Pourcentage de variance expliquée",
     main = "Ébouli des valeurs propres")
lines(cumul, type = "b", pch = 15, col = "red")  # Ajouter la variance cumulée

# Ajouter une légende
legend("topright", legend = c("Variance expliquée", "Variance cumulée"),
       col = c("blue", "red"), pch = 15, lty = 1)

# Ajouter une ligne verticale pour la sélection des composantes (facultatif)
abline(v = 5, col = "darkgreen", lty = 2)  # Par exemple, si 5 composantes sont retenues
```

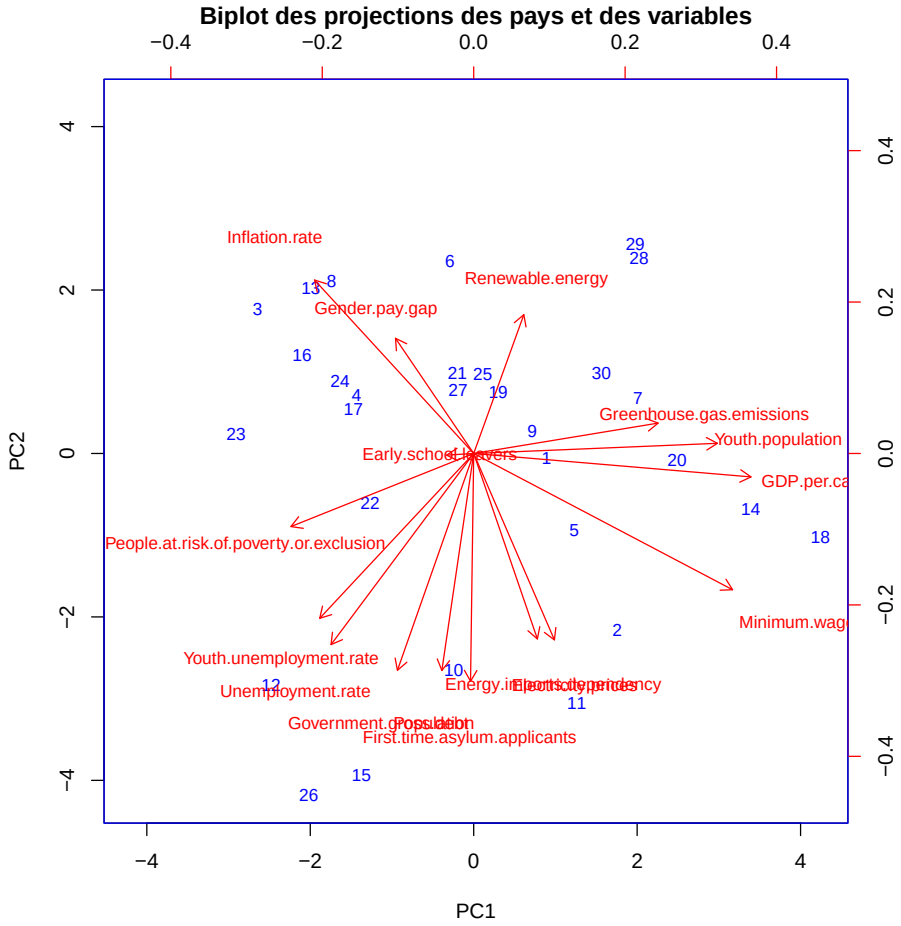


L'ébouli des valeurs propres représente la contribution de chaque composante principale à la variance totale. Dans le graphique, la ligne bleue montre les pourcentages de variances expliquées par chaque composante, tandis que la ligne rouge illustre la variance expliquée cumulée.

En analysant la variance expliquée cumulée, les cinq premières composantes principales expliquent environ 77 % de la variance totale. Cela justifie le choix de retenir les cinq premières composantes pour une analyse approfondie, car elles capturent l'essentiel de l'information tout en réduisant la dimensionnalité des données. Les composantes suivantes apportent une contribution marginale et peuvent être négligées pour simplifier l'interprétation.

Le critère de l'éboulis (coudée visible dans la ligne bleue après la cinquième composante) renforce également cette décision. Ainsi, la sélection des cinq premières composantes est justifiée à la fois par la proportion de variance expliquée et par la méthode visuelle.

```r
# 8
# Créer le biplot avec des couleurs et des tailles ajustées
biplot(res, scale = 0,
       main = "Biplot des projections des pays et des variables",
       cex = 0.8,  # Taille des points et flèches
       col = c("blue", "red"))
```



Le biplot illustre simultanément les projections des pays, identifiés par des numéros en bleu, et des variables initiales, représentées par des flèches rouges, dans le plan formé par les deux premières composantes principales (PC1 et PC2). Cette représentation permet d'observer les regroupements de pays ayant des profils similaires ainsi que l'influence des variables sur les composantes principales. Les flèches rouges indiquent la direction et l'intensité des variables : plus la flèche est longue, plus la contribution de la variable à la composante principale est importante. On remarque que Renewable.energy, Greenhouse.gas.emissions et Youth.population influencent fortement le PC1, tandis que des variables comme People.at.risk.of.poverty.or.exclusion et Youth.unemployment.rate ont un impact plus prononcé sur le PC2.

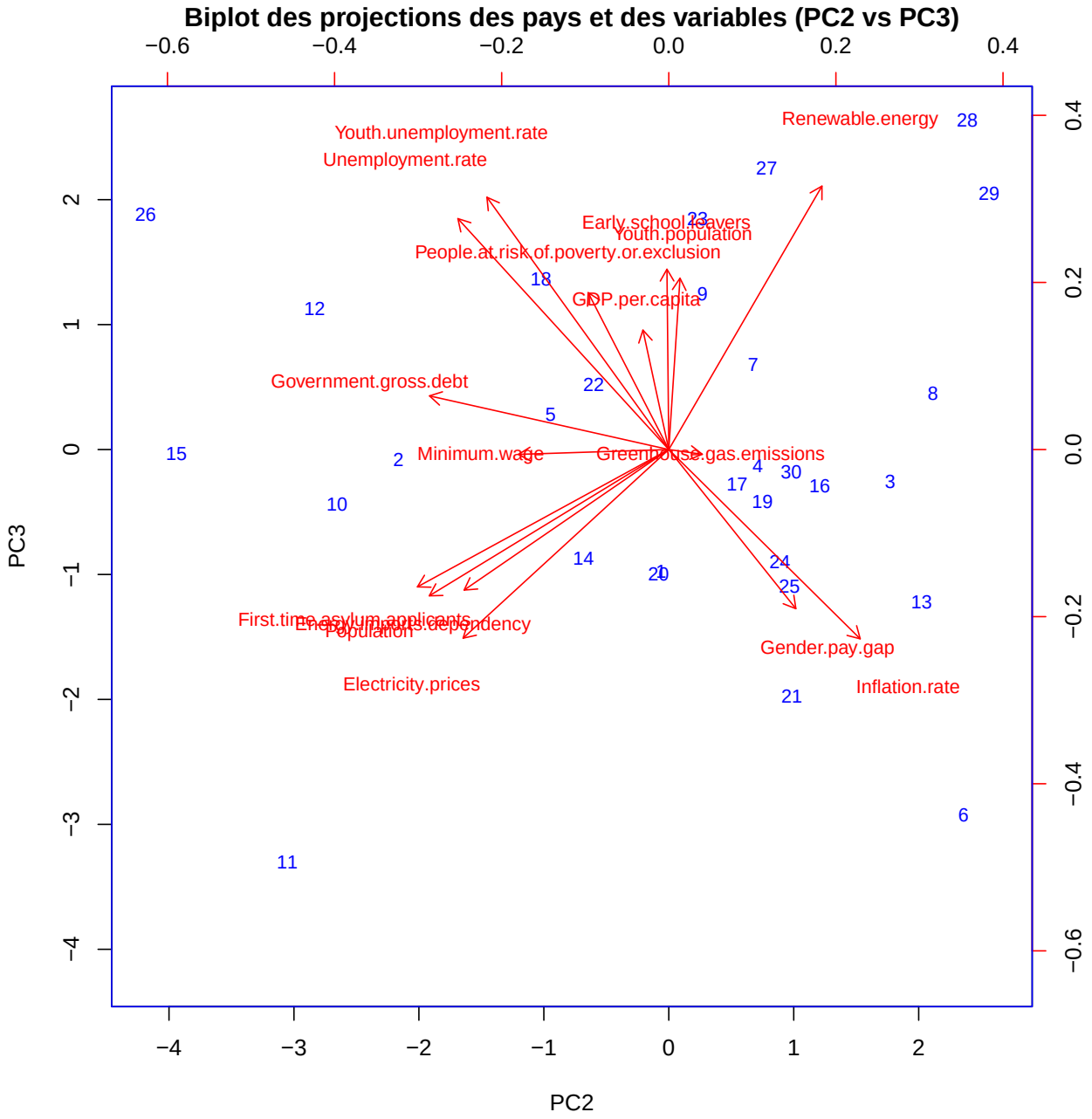
Les pays proches sur le graphique ont des caractéristiques comparables. Par exemple, la Norvège (29) et l'Islande (28), situées à l'extrémité droite, se démarquent par leurs valeurs élevées de Renewable.energy, reflétant leur forte utilisation des énergies renouvelables. En revanche, la Grèce (12) et l'Espagne (26), situées à l'opposé dans le plan factoriel, présentent des spécificités économiques différentes. On observe également un regroupement central formé par la Pologne (21), la Slovénie (25), la Suède (27) et Malte (19), suggérant des similitudes dans leurs profils globaux.

Ce biplot facilite l'analyse des similarités et divergences entre les pays et met en évidence les variables expliquant ces différences. Il constitue ainsi un outil essentiel pour interpréter les relations complexes au sein des données multidimensionnelles.

Bien que le plan formé par les composantes principales PC1 et PC2 capture la majeure partie de la variance des données, il ne représente pas nécessairement toutes les informations de manière optimale pour chaque variable ou individu. En effet, certaines variables ou certains pays pourraient être mieux expliqués dans d'autres plans factoriels. Il est donc pertinent d'explorer des plans complémentaires, tels que PC2 vs PC3,

pour obtenir une vision plus exhaustive et détaillée des relations entre les variables et les pays.

```r
#  9
# Plan PC2 vs PC3
biplot(res, choices = c(2, 3), scale = 0,
       main = "Biplot des projections des pays et des variables (PC2 vs PC3)",
       cex = 0.8,
       col = c("blue", "red"))
```



Le biplot montre simultanément les projections des pays et des variables dans le plan formé par les composantes principales PC2 et PC3. Ce plan permet d'explorer des informations supplémentaires qui n'étaient pas totalement visibles dans le premier plan (PC1 vs PC2).

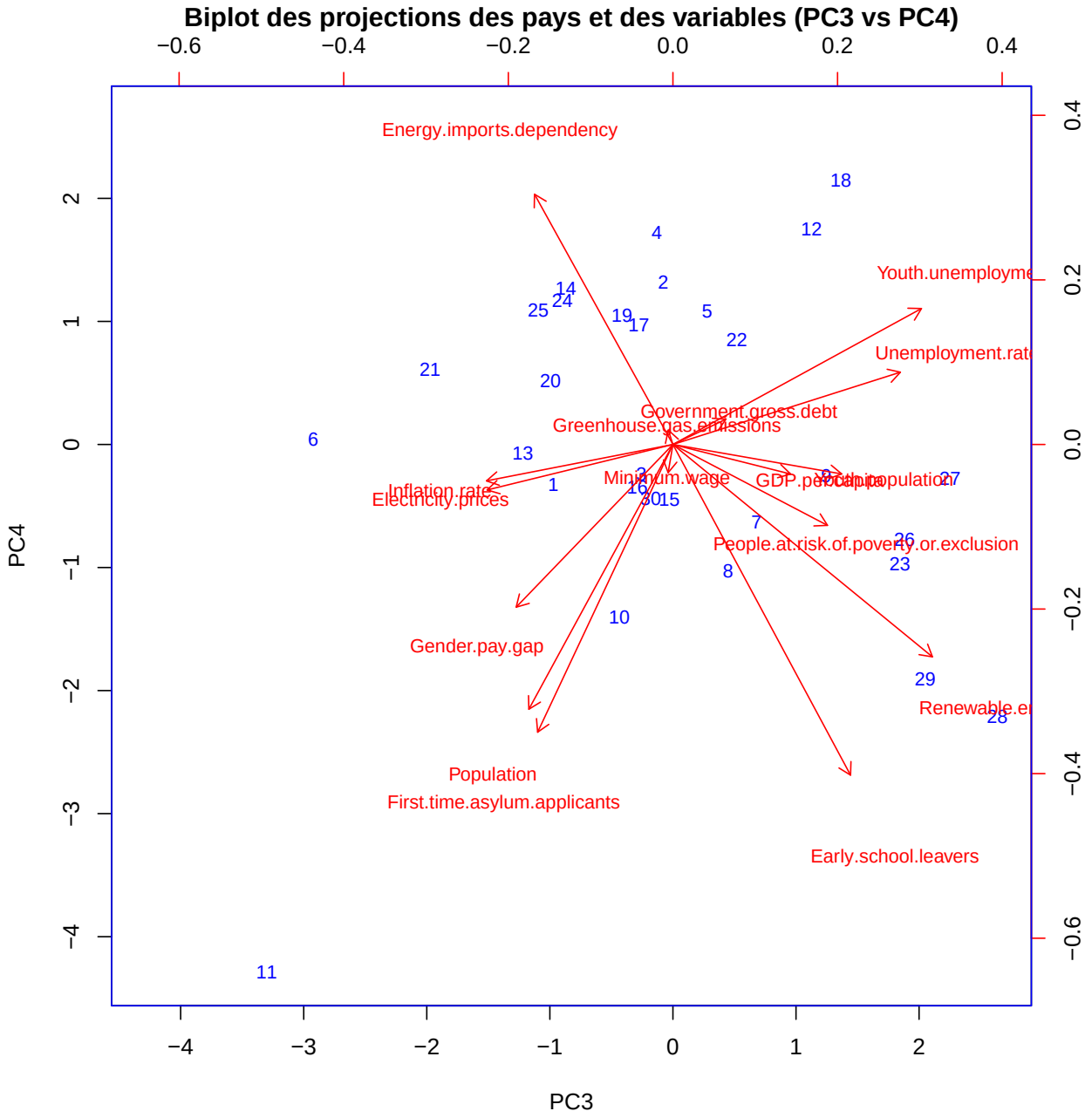
Les variables représentées par les flèches rouges indiquent leur contribution au plan PC2 vs PC3. On

remarque que certaines variables, comme Renewable Energy et Youth Population, ont une forte contribution dans la direction de PC3. De même, des variables comme Gender Pay Gap et Inflation Rate influencent davantage le PC2.

Du côté des pays (points bleus numérotés), on observe que :

Les pays 28 (Islande) et 29 (Norvège) se distinguent encore une fois dans la direction de Renewable.energy, reflétant leur forte proportion d'énergie renouvelable dans la consommation totale. Les pays 21 (Pologne) et 13 (Hongrie) sont projetés dans la direction de Inflation.rate et Gender.pay.gap, indiquant des caractéristiques liées à ces indicateurs économiques.

```r
# 9.
# Plan PC3 vs PC4
biplot(res, choices = c(3, 4), scale = 0,
       main = "Biplot des projections des pays et des variables (PC3 vs PC4)",
       cex = 0.8,  # Taille des points et flèches
       col = c("blue", "red"))
```



Sur le second plan factoriel (PC3 vs PC4), certaines variables se distinguent par leur forte influence. On observe notamment que Energy.imports.dependency et Renewable.energy présentent des projections marquées dans des directions opposées. Cela traduit une relation inverse entre ces deux variables : les pays ayant une forte dépendance aux importations d'énergie tendent à utiliser moins d'énergies renouvelables, et vice versa.

Les variables Population et First.time.asylum.applicants se projettent dans des directions similaires, indiquant une corrélation . Cela peut refléter une tendance selon laquelle les pays ayant une population élevée reçoivent davantage de premières demandes d'asile. Une relation similaire est observée entre les varibles infliation.rates et Electicité.Prices, indiquant leur compatitibilité à évoluer dans le meme sens.

Au niveau des pays, on observe que le pays 11 (Allemagne)se distingue nettement sur l'axe PC3, en bas à gauche, par son éloignement des autres. Cela reflète des caractéristiques uniques par rapport aux autres nations. Les pays 29 (Norvège) et 28 (Islande) restent toujours associés à la variable Renewable.energy sur l'axe PC3, ce qui souligne leur forte indépendance énergétique liée à une faible dépendance aux importations d'énergie.

Le plan PC2 vs PC3 et PC3 vs PC4 consolident et révèlent des informations supplémentaires sur les relations entre les pays, qui étaient moins visibles dans le plan précédent. Ce type d'analyse permet ainsi de compléter la compréhension des relations complexes entre les variables et les pays européens.

Ces analyses mettent en lumière les interactions complexes entre les variables et les pays, offrant une meilleure compréhension des dynamiques sous-jacentes et identifiant les facteurs différenciant certains pays.

```r
#  10. Déterminez quelles sont les variables les mieux représentées par le premier plan factoriel.
# Étape 1 : Extraire les loadings des composantes principales
loadings <- res$rotation

# Étape 2 : Calculer le cos² pour le plan PC1-PC2
cos2_PC1_PC2 <- rowSums(loadings[, 1:2]^2)  # Somme des carrés des charges sur PC1 et PC2

# Étape 3 : Créer une table avec les cos² et les variables associées
cos2_table <- data.frame(
  Variable = rownames(loadings),          # Noms des variables
  Cos2_PC1_PC2 = round(cos2_PC1_PC2, 2)   # Valeurs arrondies des cos²
)

# Étape 4 : Vérifier que la colonne Cos2_PC1_PC2 est bien un vecteur numérique
cos2_table$Cos2_PC1_PC2 <- as.numeric(cos2_table$Cos2_PC1_PC2)

# Étape 5 : Trier les variables par ordre décroissant de cos²
cos2_table <- cos2_table[order(cos2_table$Cos2_PC1_PC2, decreasing = TRUE), ]

# Étape 6 : Afficher la table triée
cat("Cos² des variables sur le plan PC1-PC2 (par ordre de qualité) :\n")
```

```
Cos² des variables sur le plan PC1-PC2 (par ordre de qualité) :
```

```r
print(cos2_table, row.names = FALSE)
```

```
                          Variable Cos2_PC1_PC2
                     Minimum.wage         0.23
                   GDP.per.capita         0.21
                Youth.population         0.16
              Unemployment.rate          0.16
                 Inflation.rate          0.15
      First.time.asylum.applicants       0.14
           Youth.unemployment.rate       0.14
             Government.gross.debt       0.14
                       Population        0.13
```

```
 People.at.risk.of.poverty.or.exclusion         0.11
                     Electricity.prices         0.11
                Greenhouse.gas.emissions        0.10
              Energy.imports.dependency         0.10
                       Renewable.energy         0.06
                         Gender.pay.gap         0.05
                   Early.school.leavers         0.00
```

Les cos² indiquent la qualité de la représentation des variables sur le premier plan factoriel (PC1-PC2). Plus le cos² est élevé, meilleure est la représentation de la variable sur ce plan. Les variables les mieux représentées incluent Minimum.wage, GDP.per.capita, et Youth.population, avec des cos² respectifs de 0.23, 0.21 et 0.16. Ces variables jouent donc un rôle clé dans la structuration de ce plan factoriel.

En revanche, des variables comme Gender.pay.gap et Early.school.leavers ont des cos² très faibles (0.05 et 0.00), indiquant qu'elles ne sont pas bien représentées sur ce plan. Cela signifie que leur contribution à l'inertie totale du plan PC1-PC2 est limitée, et qu'elles pourraient être mieux représentées sur d'autres plans factoriels.

Ces résultats permettent de se concentrer sur les variables dominantes pour analyser les premières composantes principales et comprendre leur impact sur la structuration des individus et des variables.

```r
# 11
# Étape 1 : Extraire les scores des individus
scores <- res$x  # Les coordonnées des individus dans le nouvel espace

# Étape 2 : Extraire les variances des composantes principales (valeurs propres)
variances <- res$sdev^2  # Variances associées à chaque composante principale

# Étape 3 : Calculer les contributions des individus sur chaque composante
# Contribution = (scores^2) / (variance de la composante)
contributions <- sweep(scores^2, 2, variances, "/")

# Étape 4 : Ajuster les paramètres pour l'affichage
options(width = 150)  # Ajuste la largeur pour éviter les retours à la ligne

# Étape 5 : Afficher les contributions des individus sous forme de tableau
contributions_table <- data.frame(Individu = seq_len(nrow(contributions)),
                                  round(contributions, 2))  # Arrondir les valeurs pour lisibilité

cat("Contributions des individus sur chaque composante principale :\n")
```

```
Contributions des individus sur chaque composante principale :
```

```r
print(contributions_table, row.names = FALSE)
```

```
 Individu  PC1  PC2  PC3  PC4  PC5  PC6   PC7  PC8  PC9 PC10 PC11 PC12 PC13 PC14 PC15 PC16
        1 0.22 0.00 0.46 0.06 0.66 0.28  0.01 0.76 0.87 0.13 0.29 1.87 1.45 0.98 0.74 0.71
        2 0.85 1.33 0.00 0.94 0.04 0.02  0.12 0.47 2.87 0.07 2.06 1.34 0.82 0.13 1.59 0.32
        3 1.92 0.89 0.03 0.03 2.17 4.27  0.05 0.00 0.53 3.72 0.11 0.60 0.12 1.05 0.37 0.13
        4 0.56 0.14 0.01 1.61 0.28 0.01  0.05 1.54 0.03 0.33 0.29 0.39 2.10 0.74 3.05 1.14
        5 0.42 0.25 0.04 0.63 0.09 1.19  0.30 7.12 0.25 1.11 0.15 5.45 1.10 0.01 0.01 0.02
        6 0.02 1.58 4.20 0.00 0.03 1.07  2.42 0.21 0.28 1.56 0.73 0.12 0.07 0.30 0.61 0.24
        7 1.11 0.13 0.23 0.21 0.66 0.22  0.26 0.70 0.28 1.96 1.71 0.03 0.01 0.72 1.16 0.32
        8 0.83 1.27 0.10 0.57 0.04 0.93  4.24 1.01 1.45 0.49 1.85 0.45 0.01 0.90 0.25 0.18
```

9	0.14	0.02	0.76	0.04	1.05	0.38	0.33	0.04	0.32	0.00	1.20	0.02	5.32	0.07	0.22	2.69
10	0.02	2.00	0.09	1.06	0.29	0.54	0.00	3.02	0.24	0.01	3.33	1.48	2.41	0.43	1.72	0.18
11	0.44	2.65	5.35	9.89	0.06	0.43	0.01	0.01	0.33	0.12	0.06	0.95	0.03	0.47	3.44	0.80
12	1.69	2.28	0.62	1.65	1.23	0.62	1.67	0.51	1.02	4.76	0.18	0.16	0.71	0.71	0.01	1.17
13	1.09	1.16	0.73	0.00	0.60	3.79	1.43	0.07	2.56	0.79	1.29	3.53	0.00	0.01	0.10	0.09
14	3.16	0.13	0.37	0.87	0.08	1.16	0.85	0.67	0.04	0.17	0.27	4.07	1.54	0.23	0.30	1.16
15	0.52	4.41	0.00	0.11	0.45	0.15	0.01	0.06	3.52	0.00	3.39	0.12	0.01	0.63	0.24	0.81
16	1.21	0.41	0.04	0.07	0.78	1.52	0.09	1.56	0.83	0.00	0.01	1.46	0.00	1.69	0.68	0.07
17	0.60	0.08	0.04	0.51	0.00	1.08	0.10	0.15	0.00	0.02	0.23	0.01	0.02	7.92	1.73	0.26
18	4.95	0.30	0.92	2.48	3.90	1.30	0.75	1.28	2.35	0.30	1.92	0.15	0.35	0.13	0.18	0.25
19	0.02	0.16	0.08	0.60	0.35	0.03	11.17	0.11	0.11	0.57	0.29	0.45	0.11	0.18	0.16	0.49
20	1.70	0.00	0.49	0.14	0.00	0.00	0.81	0.23	1.43	0.36	3.32	0.01	0.01	0.02	0.10	1.09
21	0.01	0.28	1.91	0.20	0.99	0.10	1.09	2.97	2.49	0.00	0.31	1.05	1.60	0.06	0.81	3.21
22	0.44	0.10	0.13	0.39	0.49	0.26	0.25	0.03	0.50	0.19	0.13	0.10	0.64	0.76	1.95	4.43
23	2.33	0.02	1.67	0.51	5.34	1.24	1.12	0.07	0.85	1.12	0.21	2.12	0.00	0.42	1.82	0.17
24	0.73	0.22	0.40	0.74	0.26	0.57	0.01	0.03	1.18	1.74	0.27	0.52	0.01	0.47	2.12	0.14
25	0.00	0.27	0.59	0.64	0.00	0.11	0.01	1.65	2.51	0.76	0.07	0.25	7.63	0.03	0.12	4.48
26	1.12	4.97	1.74	0.32	0.01	0.01	0.01	0.04	1.04	1.37	0.12	2.03	0.94	0.65	3.52	0.52
27	0.01	0.17	2.49	0.04	2.13	0.21	0.01	3.00	0.49	3.21	0.01	0.00	0.66	1.73	1.56	1.31
28	1.13	1.62	3.41	2.64	1.77	4.17	0.50	0.66	0.34	3.40	0.01	0.01	0.81	2.98	0.06	0.02
29	1.08	1.87	2.07	1.96	0.30	0.23	0.00	1.02	0.00	0.15	1.78	0.03	0.00	4.37	0.36	0.89
30	0.67	0.27	0.02	0.10	4.97	3.10	1.32	0.01	0.30	0.60	3.43	0.23	0.53	0.24	0.02	1.67

L'analyse des contributions des individus aux composantes principales a permis de mettre en évidence plusieurs aspects intéressants. Tout d'abord, les composantes principales les plus importantes, notamment PC1, PC2, PC3 et PC4, capturent la majeure partie de la variance totale des données. Les contributions élevées sur ces composantes montrent les individus qui influencent fortement la structure globale de l'espace factoriel. Par exemple, l'individu 11 (Allemagne) se distingue particulièrement sur PC3 (5.35), PC5 (9.89) et PC15 (3.44), tandis que l'individu 18 (Luxembourg) se démarque sur PC1 (4.95) et sur PC5 (3.90), de même que l'individu 19 (Malte) avec une valeur remarquable sur PC7 (11.17).

L'individu 28 (Islande) se démarque sur PC3 (3.41) et sur PC10 (3.40). L'individu 25 (Slovénie) se distingue également sur PC13 (7.63).

Les contributions totales révèlent que certains individus, comme 11 (Allemagne), 18 (Luxembourg) et 28 (Islande), participent de manière significative à la variance expliquée sur plusieurs axes, ce qui explique leur rôle important dans l'analyse. D'autres individus, bien que moins marquants sur les composantes principales initiales, se révèlent essentiels sur des axes secondaires. Par exemple, l'individu 23 (Roumanie) a une forte contribution sur PC5, et l'individu 19 (Malte) se distingue clairement sur PC7. Ainsi, si PC7 était utilisé dans l'un des plans d'analyse, il serait tout à fait pertinent de refaire cette analyse sans l'individu 19 (Malte), en raison de sa forte contribution (11.17), c'est-à-dire en le supprimant temporairement de l'analyse.

La suppression d'un individu dans une analyse en ACP peut se justifier si cet individu a une contribution anormalement élevée sur une ou plusieurs composantes principales, ce qui peut indiquer qu'il exerce une influence disproportionnée sur les résultats globaux (outlier). Cela peut biaiser l'interprétation des axes.

Cette répartition des contributions montre que tous les individus jouent un rôle dans la construction des composantes principales, aucun ne pouvant être considéré comme négligeable ou trop influent sur les axes utilisés dans les plans (PC1, PC2, PC3, PC4).

Cela signifie qu'il n'est pas nécessaire de retirer des individus de l'analyse de manière définitive, car chacun contribue de manière significative à l'explication de la variance totale.

Les individus ayant des contributions élevées sur certaines composantes principales indiquent des particularités ou des caractéristiques spécifiques qui les différencient des autres, tandis que ceux ayant une contribution plus homogène sur l'ensemble des composantes montrent une participation équilibrée dans l'analyse.

Avant de passer à l'ordre des contributions, cette analyse globale offre déjà une vue d'ensemble des rôles joués par chaque individu dans la construction des composantes principales et dans l'explication de la variance totale des données.

```
# Étape 5 : Calcul des contributions totales par individu (somme sur toutes les composantes)
contributions_totales <- rowSums(contributions)

# Étape 6 : Trier les contributions totales de façon décroissante
contributions_totales_table <- data.frame(Individu = seq_len(nrow(scores)),
                                          Contribution_Totale = round(contributions_totales, 2))
contributions_totales_table <- contributions_totales_table[order(-contributions_totales_table$Contributi

# Étape 7 : Afficher les contributions totales triées sous forme de tableau
cat("\nContributions totales par individu (classées par ordre décroissant) :\n")
```

```
Contributions totales par individu (classées par ordre décroissant) :
```

```
print(contributions_totales_table, row.names = FALSE)
```

```
 Individu Contribution_Totale
       11               25.03
       28               23.55
       18               21.50
       25               19.12
       23               18.99
       12               18.96
       26               18.42
        5               18.13
       30               17.47
       13               17.24
       21               17.08
       27               17.04
       10               16.81
       29               16.11
        3               16.01
       14               15.08
       19               14.88
        8               14.57
       15               14.42
        6               13.44
        2               12.96
       17               12.74
        9               12.61
        4               12.26
       22               10.81
       16               10.42
       20                9.73
        7                9.72
        1                9.50
       24                9.42
```

```r
#11..

seuil <- 1 / nrow(scores)
individus_faibles <- contributions_totales_table$Individu[contributions_totales < seuil]

# Étape 8 : Afficher les individus ayant une contribution faible
if (length(individus_faibles) == 0) {
 #  cat("\nAucun individu n'a une contribution faible. Il n'est pas nécessaire d'éliminer des individus
} else {
  cat("\nIndividus ayant une contribution faible (en dessous du seuil) :\n")
  print(individus_faibles)
}
```

```
NULL
```

L'analyse globale des contributions montre que les individus 11, 28 et 18 se démarquent par des contributions totales particulièrement élevées, respectivement 25.03 %, 23.55 % et 21.50 %. Cela signifie qu'ils jouent un rôle majeur dans l'explication de la variance globale des données sur l'ensemble des composantes principales.

Les individus 25, 23, 12 et 26 suivent de près, avec des contributions allant de 18.42 % à 19.12 %. Ces valeurs confirment leur importance relative dans l'analyse, même s'ils ne dominent pas autant que les trois premiers. Ces individus influencent fortement plusieurs axes, ce qui renforce leur pertinence dans l'espace factoriel.

En revanche, les individus 1, 7 et 24 affichent les contributions les plus faibles, respectivement 9.50 %, 9.72 % et 9.42 %. Bien que ces valeurs soient les plus basses, elles restent significativement supérieures au seuil minimal de 3.33 %, ce qui confirme que tous les individus participent de manière notable à l'explication de la variance totale.

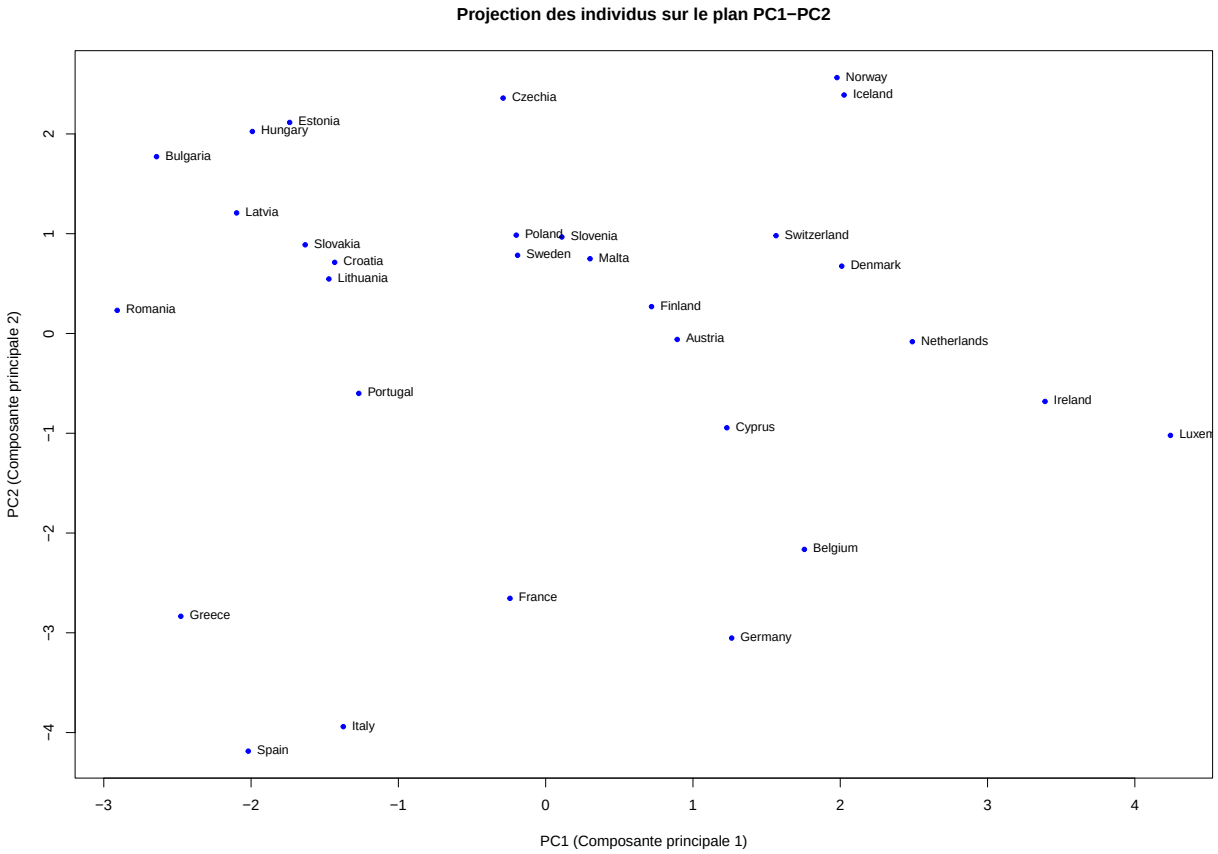
Cette hiérarchisation des individus permet d'identifier ceux qui contribuent le plus à l'explication des données et ceux dont l'influence est plus modérée. Elle met en lumière les individus clés dans l'interprétation des résultats de l'analyse en composantes principales (ACP) et justifie l'inclusion de tous les individus dans l'étude.

La suppression d'un individu reste une décision contextuelle : elle dépend du domaine d'application et de l'objectif de l'analyse.

En définitive, aucun individu ne présente une valeur qui s'éloigne significativement de la moyenne, surtout sur les axes utilisés pour les plans de l'analyse (PC1, PC2, PC3, PC4), d'où le choix de ne supprimer aucun individu.

```r
# 12

# Étape 1 : Extraire les scores
coord <- res$x  # Scores des individus

# Étape 2 : Ajouter les noms des pays
tmp <- read.csv("data/euro.csv", header = TRUE, sep = ";")  # Charger les noms
rownames(coord) <- tmp[, 1]

# Étape 3 : Créer le graphique
plot(coord[, 1], coord[, 2],
     xlab = "PC1 (Composante principale 1)",
     ylab = "PC2 (Composante principale 2)",
     main = "Projection des individus sur le plan PC1-PC2",
     pch = 20, col = "blue")
```

```r
# Ajouter les noms des individus
text(coord[, 1], coord[, 2], labels = rownames(coord), pos = 4, cex = 0.8)
```



La projection des individus sur les composantes principales (PC1 et PC2) montre des regroupements qui correspondent globalement aux similarités socio-économiques attendues :

On peut remarquer un regroupement de certains pays du nord, situés en haut à droite, comme l'Islande, la Norvège, le Danemark et l'Irlande, qui contrastent avec des pays du sud comme l'Espagne, l'Italie et la Grèce en bas à gauche. Ces proximités et éloignements, ou encore ces similarités et dissimilarités, s'expliquent du point de vue des variables socio-économiques considérées. Bien qu'il puisse exister une certaine corrélation entre la proximité des pays et leur situation géographique, cette relation n'est pas systématique : on observe, par exemple, la Suisse parmi les pays du nord alors qu'elle ne fait pas partie de cette région. De même, certains pays du nord se rapprochent de pays appartenant à d'autres zones géographiques.

Au centre du graphique, on observe un regroupement de pays comme la Pologne, la Suède, Malte et la Slovénie, qui partagent des caractéristiques socio-économiques similaires.

En haut à gauche, on trouve particulièrement des pays de l'Est comme la Bulgarie, la Hongrie et la Roumanie, qui sont diamétralement opposés à des pays tels que l'Allemagne et la Belgique, situés en bas à droite.

Conclusion Cette étude en analyse en composantes principales (ACP) a permis de réduire la dimensionnalité des données tout en conservant l'essentiel de l'information. Les résultats obtenus, tels que les projections des pays sur les composantes principales et l'éboulis des valeurs propres, ont révélé des regroupements significatifs et des similarités entre les pays en fonction de leurs caractéristiques socio-économiques. Ces visualisations offrent une vue claire des relations entre les variables et des différences structurelles entre les pays, facilitant ainsi une interprétation plus approfondie. Cette analyse constitue une base solide pour explorer des méthodes complémentaires, telles que le partitionnement, afin de segmenter davantage les pays en groupes homogènes.

```r
# TP2 ****************
```

```r
# Question 1

# Matrice de dissimilarité Euclidienne (non normalisée)
dissimilarity_euclidean <- as.matrix(dist(euro_data, method = "euclidean"))

# Normaliser les données
euro_data_normalized <- scale(euro_data)

# Matrice de dissimilarité Euclidienne (normalisée)
dissimilarity_reduced <- as.matrix(dist(euro_data_normalized, method = "euclidean"))

# Afficher les matrices
cat("Matrice de dissimilarité Euclidienne (non normalisée) :\n")
```

```
Matrice de dissimilarité Euclidienne (non normalisée) :
```

```r
print(round(dissimilarity_euclidean, 2))
```

```
            1          2          3          4          5          6          7          8          9
1         0.0  2638160.9  2657441.6  5254209.8  8184197.7  1723739.7  3172609.6  7739095.7  3541179.23 5
2   2638160.9        0.0  5295172.6  7891982.7 10822113.6   915893.4  5810224.2 10376966.3  6178875.82 5
3   2657441.6  5295172.6        0.0  2596908.1  5527055.2  4379883.5   517377.9  5081864.8   884403.26 6
4   5254209.8  7891982.7  2596908.1        0.0  2930238.9  6976636.0  2082102.7  2485011.2  1713222.94 6
5   8184197.7 10822113.6  5527055.2  2930238.9        0.0  9906837.9  5012023.0   445423.9  4643283.92 6
6   1723739.7   915893.4  4379883.5  6976636.0  9906837.9        0.0  4894993.5  9461646.0  5263592.70 5
7   3172609.6  5810224.2   517377.9  2082102.7  5012023.0  4894993.5        0.0  4566922.4   369016.54 6
8   7739095.7 10376966.3  5081864.8  2485011.2   445423.9  9461646.0  4566922.4        0.0  4198142.43 6
9   3541179.2  6178875.8   884403.3  1713222.9  4643283.9  5263592.7   369016.5  4198142.4        0.00 6
10 59068272.1 56430300.0 61725394.4 64322245.8 67252408.7 57345630.8 62240489.5 66807244.6 62609165.06
11 75254567.8 72616667.8 77911743.7 80508619.6 83438748.0 73532049.3 78426873.1 82993600.2 78795543.54 1
12  1309339.5  1329246.9  3966447.0  6563330.6  9493397.5   417425.3  4481796.4  9048259.5  4850339.35 5
13   498677.6  2143374.4  3152120.1  5748850.2  8679061.1  1227792.4  3667289.2  8233861.0  4035839.96 5
14  3833770.1  6471512.3  1178083.5  1421690.0  4350916.7  5556402.2   661626.8  3905930.1   294757.49 6
15 49892485.3 47254514.4 52549606.4 55146459.4 58076621.7 48169847.0 53064707.3 57631457.5 53433380.51
16  7222010.5  9859856.2  4564752.4  1967886.6   962468.6  8944522.6  4049836.8   517133.4  3681040.00 6
17  6247781.6  8885592.0  3590504.5   993615.7  1936652.9  7970250.9  3075606.3  1491399.1  2706784.64 6
18  8444257.2 11082114.6  5787427.3  3190822.5   265931.4 10166927.0  5271935.1   708356.7  4903380.29 6
19  8562908.8 11200787.2  5905729.2  3308864.2   378816.7 10285481.2  5390666.0   823917.6  5021931.53 6
20  8706540.1  6068506.0 11363651.2 13960476.9 16890619.4  6983909.5 11878694.2 16445468.8 12247370.11 5
21 27649015.7 25010959.5 30306030.3 32902842.6 35833037.6 25926208.1 30821105.6 35387852.2 31189774.11 3
22  1412975.2  1226591.2  4068976.5  6665728.9  9595927.6   310914.0  4584084.3  9150738.2  4952681.53 5
23  9949920.8  7311827.9 12606844.4 15203656.9 18133855.5  8227027.8 13121964.2 17688665.7 13490605.64 4
24  3676462.9  6314104.5  1019192.6  1577899.5  4508119.1  5398737.4   505152.5  4062909.8   136788.50 6
25  6987988.3  9625861.3  4330788.3  1733946.6  1196292.5  8710559.9  3815806.0   751126.5  3447031.16 6
26 38980730.5 36342803.8 41637883.5 44234753.4 47164894.8 37258173.3 42153012.4 46719740.1 42521678.83 1
27  1464078.3  1261488.4  4207644.3  6889503.2  9915736.2   317260.5  4739409.5  9455987.4  5120180.96 5
28  9319046.8 12139076.8  6478490.5  3702359.4   569987.2 11160612.2  5927755.2  1046036.9  5533605.49 7
29  3734910.3  6459056.0   992533.1  1692844.2  4718327.5  5513902.8   458665.5  4258721.0    85416.58 6
30   301607.7  3023539.1  2446003.6  5127617.3  8153693.3  2078822.6  2977400.6  7694009.0  3358237.98 6
           14         15         16         17         18         19         20         21         22       1
```

1	3833770.1	49892485	7222010.5	6247781.6	8444257.2	8562908.8	8706540	27649016	1412975.22	99499	
2	6471512.3	47254514	9859856.2	8885592.0	11082114.6	11200787.2	6068506	25010959	1226591.23	73118	
3	1178083.5	52549606	4564752.4	3590504.5	5787427.3	5905729.2	11363651	30306030	4068976.47	126068	
4	1421690.0	55146459	1967886.6	993615.7	3190822.5	3308864.2	13960477	32902843	6665728.92	152036	
5	4350916.7	58076622	962468.6	1936652.9	265931.4	378816.7	16890619	35833038	9595927.61	181338	
6	5556402.2	48169847	8944522.6	7970250.9	10166927.0	10285481.2	6983909	25926208	310914.02	82270	
7	661626.8	53064707	4049836.8	3075606.3	5271935.1	5390666.0	11878694	30821106	4584084.33	131219	
8	3905930.1	57631458	517133.4	1491399.1	708356.7	823917.6	16445469	35387852	9150738.20	176886	
9	294757.5	53433381	3681040.0	2706784.6	4903380.3	5021931.5	12247370	31189774	4952681.53	134906	
10	62901731.7	9175789	66290127.4	65315860.8	67512336.5	67631081.0	50361800	31419547	57656533.80	491186	
11	79088088.5	25362422	82476490.1	81502231.0	83698685.7	83817438.5	66548190	47606198	73842947.40	653050	
12	5143049.7	48583274	8531161.6	7556922.1	9753540.9	9872100.7	7397378	26339802	116587.45	86407	
13	4328749.0	49397632	7716736.2	6742465.0	8939201.5	9057701.2	8211692	27153993	916896.26	94548	
14	0.0	53725952	3388911.8	2414819.3	4610612.9	4729577.0	12539951	31482393	5245494.44	137832	
15	53725951.5	0	57114340.6	56140074.3	58336558.1	58455294.8	41186016	22243808	48480749.68	399428	
16	3388911.8	57114341	0.0	974273.0	1224209.5	1341023.6	15928356	34870729	8633615.51	171715	
17	2414819.3	56140074	974273.0	0.0	2197538.9	2315257.5	14954089	33896458	7659343.84	161972	
18	4610612.9	58336558	1224209.5	2197538.9	0.0	131665.5	17150563	36092993	9856017.34	183938	
19	4729577.0	58455295	1341023.6	2315257.5	131665.5	0.0	17269291	36211688	9974572.55	185125	
20	12539950.7	41186016	15928356.0	14954089.2	17150563.2	17269290.8	0	18942493	7294799.49	12440	
21	31482393.0	22243808	34870728.6	33896457.8	36092992.5	36211687.6	18942493	0	26237115.97	176991	
22	5245494.4	48480750	8633615.5	7659343.8	9856017.3	9974572.6	7294799	26237116	0.00	85379	
23	13783290.5	39942840	17171542.2	16197272.4	18393885.7	18512506.5	1244054	17699189	8537935.36		
24	167299.3	53568569	3545785.7	2571513.5	4768452.2	4886751.3	12382589	31324945	5087830.52	136257	
25	3154818.4	56880363	234200.3	740373.1	1457455.8	1574941.4	15694366	34636765	8399650.60	169375	
26	42814244.4	10911881	46202627.6	45228366.0	47424850.3	47543579.1	30274323	11332659	37569072.06	290312	
27	5422417.0	50065591	8921916.1	7915695.8	10184216.3	10306809.1	7528867	27092497	27013.79	88129	
28	5220936.5	62656277	1598777.4	2640185.5	295313.7	165765.1	18626576	38876876	10828235.15	199556	
29	224880.3	55263219	3724730.3	2718655.6	4986532.4	5109387.3	12726502	32290151	5192801.16	140106	
30	3660253.9	51827680	7159963.7	6153770.5	8422071.9	8544791.2	9290960	28854655	1757788.42	105751	

	28	29	30
1	9319046.8	3734910.33	301607.7
2	12139076.8	6459056.01	3023539.1
3	6478490.5	992533.11	2446003.6
4	3702359.4	1692844.17	5127617.3
5	569987.2	4718327.55	8153693.3
6	11160612.2	5513902.84	2078822.6
7	5927755.2	458665.48	2977400.6
8	1046036.9	4258721.03	7694009.0
9	5533605.5	85416.58	3358238.0
10	72465604.4	64739922.94	61304386.8
11	89769540.2	81457137.37	78021579.4
12	10718665.1	5087099.33	1651970.6
13	9848070.9	4245993.26	812140.9
14	5220936.5	224880.27	3660253.9
15	62656277.5	55263218.86	51827680.0
16	1598777.4	3724730.34	7159963.7
17	2640185.5	2718655.56	6153770.5
18	295313.7	4986532.40	8422071.9
19	165765.1	5109387.27	8544791.2
20	18626576.4	12726501.58	9290960.2
21	38876876.5	32290151.02	28854655.5
22	10828235.2	5192801.16	1757788.4
23	19955666.8	14010599.65	10575119.5

```
24  5389159.8    84618.29  3498101.7
25  1848726.3  3482978.53  6918252.8
26 50991159.3 43993667.87 40558110.9
27 10833487.5  5197695.75  1762330.4
28        0.0  5453536.80  9009575.6
29  5453536.8        0.00  3435576.1
30  9009575.6  3435576.09        0.0
```

```
cat("\nMatrice de dissimilarité Euclidienne (normalisée) :\n")
```

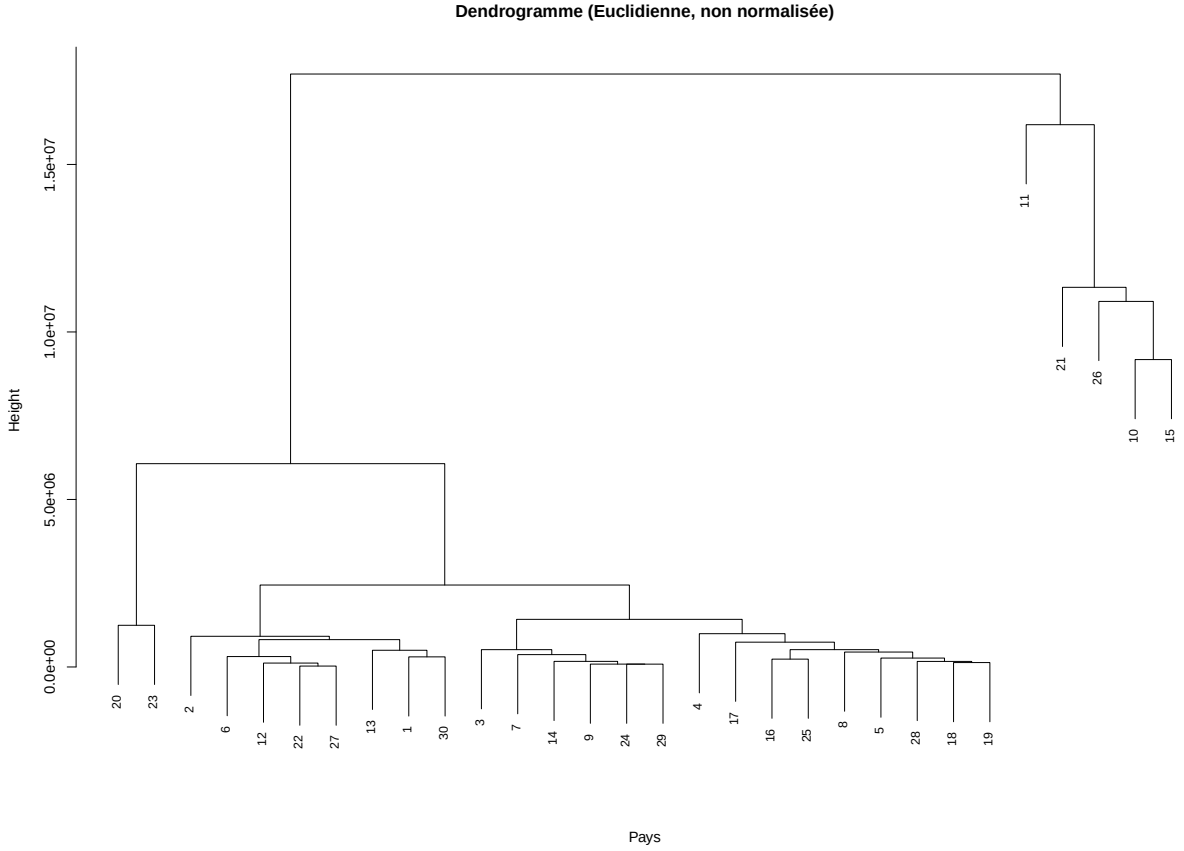
```
Matrice de dissimilarité Euclidienne (normalisée) :
```

```
print(round(dissimilarity_reduced, 2))
```

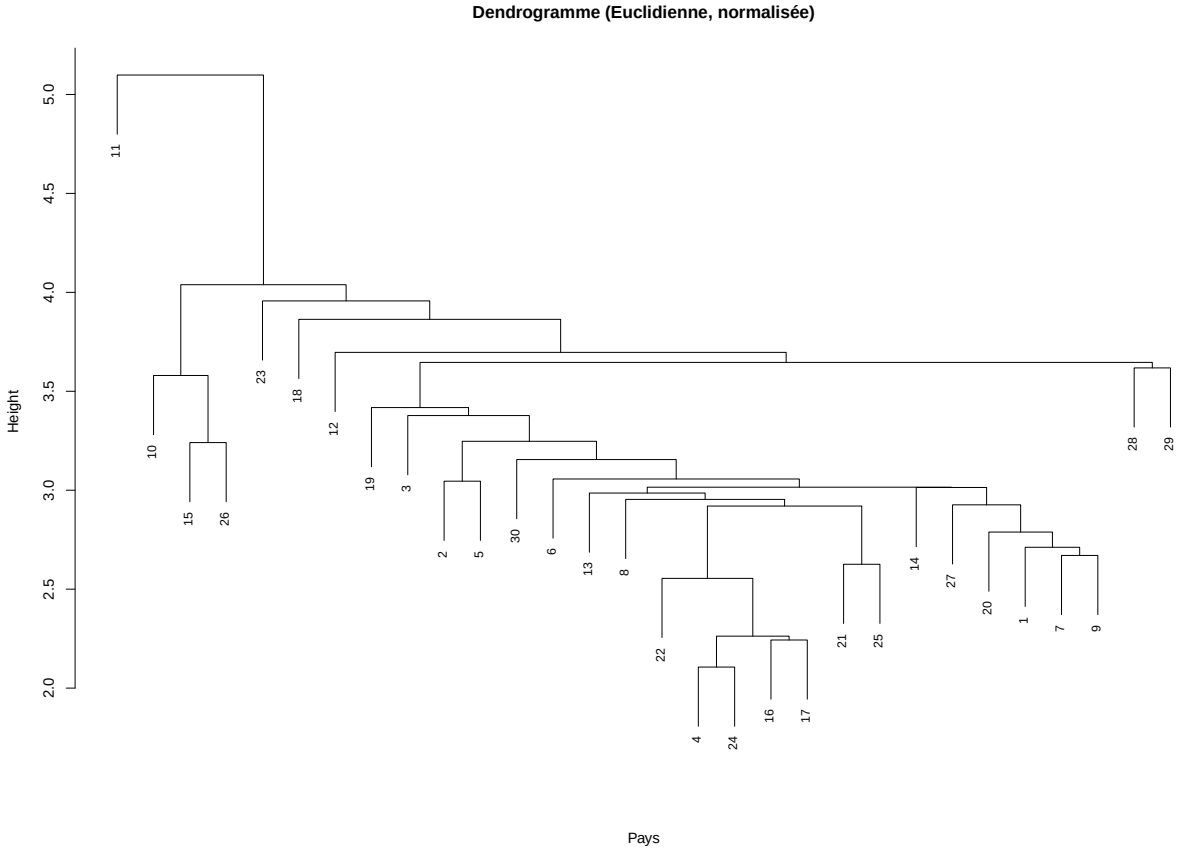
```
      1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20
1  0.00 3.80 5.55 3.97 3.61 4.15 3.04 4.70 2.71 4.04 5.87 5.60 4.48 3.95 5.40 3.91 3.49 6.32 3.92 2.79
2  3.80 0.00 6.87 4.91 3.05 6.20 4.09 6.63 4.12 4.43 6.90 5.34 6.83 3.25 4.39 5.73 4.76 4.96 4.87 3.57
3  5.55 6.87 0.00 4.26 6.58 5.39 6.13 3.53 5.61 6.52 8.25 6.66 4.71 7.13 6.77 3.58 3.38 8.16 5.17 6.17
4  3.97 4.91 4.26 0.00 4.79 4.65 4.87 4.29 3.81 4.94 8.36 4.59 4.18 5.66 5.74 3.47 2.26 6.98 4.07 4.55
5  3.61 3.05 6.58 4.79 0.00 5.64 3.65 5.96 4.01 5.21 7.36 5.63 5.87 4.10 5.24 5.39 4.42 5.60 4.00 3.80
6  4.15 6.20 5.39 4.65 5.64 0.00 5.36 4.59 5.30 6.46 7.50 7.65 4.00 5.79 7.31 4.51 4.55 8.09 5.59 5.20
7  3.04 4.09 6.13 4.87 3.65 5.36 0.00 4.84 2.67 5.01 6.95 6.94 5.88 3.89 6.41 4.54 4.40 5.79 4.24 3.15
8  4.70 6.63 3.53 4.29 5.96 4.59 4.84 0.00 4.14 6.11 8.09 6.43 4.87 6.66 7.09 2.95 3.82 8.15 6.04 5.98
9  2.71 4.12 5.61 3.81 4.01 5.30 2.67 4.14 0.00 4.28 7.33 5.32 5.22 4.86 5.73 4.15 3.85 6.06 4.63 3.70
10 4.04 4.43 6.52 4.94 5.21 6.46 5.01 6.11 4.28 0.00 5.10 5.01 5.98 5.82 3.58 5.47 5.04 7.37 5.61 4.62
11 5.87 6.90 8.25 8.36 7.36 7.50 6.95 8.09 7.33 5.10 0.00 8.79 8.22 7.12 6.18 7.62 7.63 9.10 7.80 6.42
12 5.60 5.34 6.66 4.59 5.63 7.65 6.94 6.43 5.32 5.01 8.79 0.00 6.90 7.16 4.51 5.60 4.96 8.26 6.71 6.81
13 4.48 6.83 4.71 4.18 5.87 4.00 5.88 4.87 5.22 5.98 8.22 6.90 0.00 7.13 6.86 4.51 4.08 8.53 4.31 5.62
14 3.95 3.25 7.13 5.66 4.10 5.79 3.89 6.66 4.86 5.82 7.12 7.16 7.13 0.00 6.54 6.25 5.32 3.86 5.34 3.01
15 5.40 4.39 6.77 5.74 5.24 7.31 6.41 7.09 5.73 3.58 6.18 4.51 6.86 6.54 0.00 5.94 5.32 7.64 6.34 6.32
16 3.91 5.73 3.58 3.47 5.39 4.51 4.54 2.95 4.15 5.47 7.62 5.60 4.51 6.25 5.94 0.00 2.24 8.37 4.97 5.58
17 3.49 4.76 3.38 2.26 4.42 4.55 4.40 3.82 3.85 5.04 7.63 4.96 4.08 5.32 5.32 2.24 0.00 6.90 3.66 4.48
18 6.32 4.96 8.16 6.98 5.60 8.09 5.79 8.15 6.06 7.37 9.10 8.26 8.53 3.86 7.64 8.37 6.90 0.00 6.52 4.82
19 3.92 4.87 5.17 4.07 4.00 5.59 4.24 6.04 4.63 5.61 7.80 6.71 4.31 5.34 6.34 4.97 3.66 6.52 0.00 3.42
20 2.79 3.57 6.17 4.55 3.80 5.20 3.15 5.98 3.70 4.62 6.42 6.81 5.62 3.01 6.32 5.58 4.48 4.82 3.42 0.00
21 4.26 4.89 4.65 3.46 5.12 3.06 5.07 4.96 4.83 5.16 7.14 6.74 4.46 5.11 5.86 4.56 3.76 6.73 4.71 4.26
22 3.28 3.90 4.84 2.55 3.73 5.30 4.22 4.64 3.02 4.07 7.53 3.70 4.55 5.57 4.18 3.34 2.59 6.97 3.85 4.53
23 6.37 6.72 3.96 5.49 6.36 7.12 6.26 5.05 5.90 6.23 8.59 6.76 5.37 7.83 5.67 4.75 4.56 8.30 5.61 7.07
24 3.47 5.31 4.47 2.11 4.54 3.71 4.81 4.00 3.91 5.04 7.90 5.09 2.99 5.85 5.81 3.38 2.55 7.45 4.00 4.66
25 3.37 4.17 4.78 2.92 4.59 3.75 4.19 5.03 3.65 5.16 7.51 6.13 4.42 4.90 5.82 4.05 3.18 6.46 3.69 3.66
26 6.17 5.76 7.12 6.23 5.89 8.60 6.76 6.98 5.66 4.18 7.29 4.31 7.44 7.68 3.24 6.44 5.81 8.06 6.98 7.08
27 4.68 5.57 5.89 3.95 5.55 6.49 3.66 4.65 2.93 5.19 8.63 6.26 5.96 6.14 6.52 4.25 4.23 6.83 5.47 5.25
28 6.24 7.31 7.09 7.46 6.48 7.17 5.44 6.42 5.21 7.79 9.45 9.25 7.11 7.27 8.84 7.22 7.28 7.37 7.00 6.42
29 4.96 6.67 6.61 6.09 6.24 6.39 3.65 5.37 3.74 6.63 8.53 8.53 6.30 6.18 8.27 5.92 6.12 6.70 5.88 5.15
30 3.63 5.44 6.33 5.08 5.48 6.00 3.15 5.72 4.06 5.89 7.31 7.40 6.66 4.76 7.31 4.51 4.60 6.86 4.88 3.98
     30
1  3.63
2  5.44
3  6.33
4  5.08
5  5.48
6  6.00
```

```
7  3.15
8  5.72
9  4.06
10 5.89
11 7.31
12 7.40
13 6.66
14 4.76
15 7.31
16 4.51
17 4.60
18 6.86
19 4.88
20 3.98
21 5.81
22 4.75
23 7.54
24 5.25
25 4.71
26 7.90
27 4.34
28 7.28
29 4.63
30 0.00
```

```r
# Question 1 suite
# CAH avec la matrice non normalisée
cah_single_euclidean <- hclust(dist(euro_data, method = "euclidean"), method = "single")

# CAH avec la matrice normalisée
cah_single_reduced <- hclust(dist(euro_data_normalized, method = "euclidean"), method = "single")

# Dendrogramme pour les données non normalisées
plot(cah_single_euclidean,
     main = "Dendrogramme (Euclidienne, non normalisée)",
     xlab = "Pays", sub = "", cex = 0.8)
```




```
# Dendrogramme pour les données normalisées
plot(cah_single_reduced,
     main = "Dendrogramme (Euclidienne, normalisée)",
     xlab = "Pays", sub = "", cex = 0.8)
```

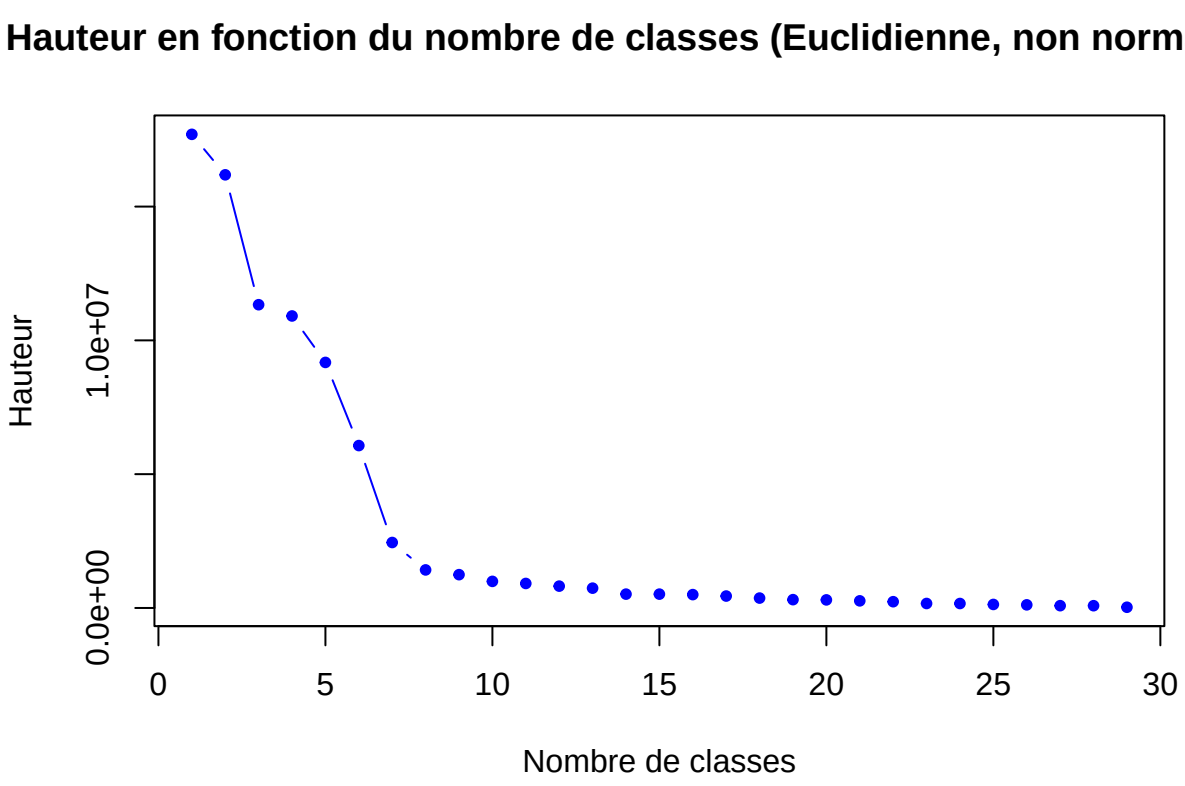


Les dendrogrammes obtenus à partir des matrices de dissimilarité Euclidienne normalisée et non normalisée mettent en évidence des différences dans les regroupements des pays. Avec la matrice normalisée, les échelles des variables sont équilibrées, ce qui permet une contribution équitable de chaque caractéristique aux regroupements.

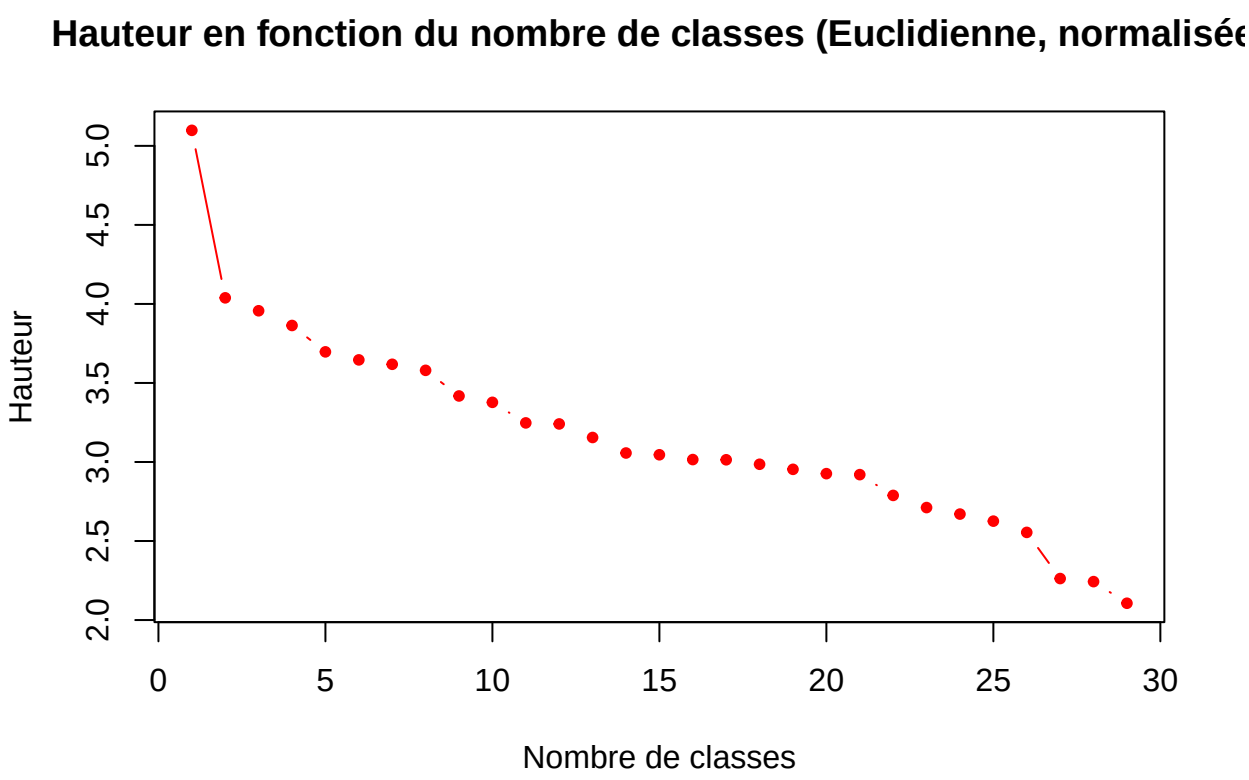
Cela se traduit par une structure de classification où les similarités relatives entre les pays sont mieux prises en compte. En revanche, la matrice non normalisée amplifie l'influence des variables à grande échelle, ce qui peut biaiser les regroupements et privilégier certaines caractéristiques au détriment d'autres.

Les fusions à des hauteurs plus élevées dans le dendrogramme non normalisé indiquent des dissimilarités globales plus marquées. Cette comparaison souligne l'importance de la normalisation pour éviter des biais et obtenir des regroupements reflétant plus fidèlement les similarités relatives entre les pays.

```r
# Question 2
#2. Représentez le dendrogramme ainsi que la "hauteur" (attribut height) en fonction du nombre de class
#représente la "hauteur" ici ? Ou couperiez-vous le dendrogramme ?
# Récupérer les hauteurs des fusions
heights_euclidean <- sort(cah_single_euclidean$height, decreasing = TRUE)
heights_reduced <- sort(cah_single_reduced$height, decreasing = TRUE)

# Courbe des hauteurs pour la matrice non normalisée
plot(1:length(heights_euclidean), heights_euclidean,
     type = "b", col = "blue", pch = 20,
     xlab = "Nombre de classes",
     ylab = "Hauteur",
     main = "Hauteur en fonction du nombre de classes (Euclidienne, non normalisée))")
```




```r
# Courbe des hauteurs pour la matrice normalisée
plot(1:length(heights_reduced), heights_reduced,
     type = "b", col = "red", pch = 20,
     xlab = "Nombre de classes",
     ylab = "Hauteur",
     main = "Hauteur en fonction du nombre de classes (Euclidienne, normalisée)")
```



Les deux graphiques montrent la relation entre le nombre de classes et la hauteur des fusions dans le dendrogramme pour des matrices de dissimilarité euclidiennes, normalisées et non normalisées.

Représentation des dendrogrammes et des hauteurs : Les dendrogrammes illustrent les regroupements hiérarchiques des observations en fonction des dissimilarités. La hauteur des branches représente la distance ou la dissimilarité entre les clusters fusionnés. Plus la hauteur est élevée, plus les groupes fusionnés sont dissemblables.

Interprétation de la hauteur : La hauteur dans ces graphiques correspond à la dissimilarité mesurée entre les clusters au moment de leur fusion. Dans le cas de la matrice normalisée, les contributions des différentes variables sont équilibrées, tandis que pour la matrice non normalisée, certaines variables peuvent dominer les regroupements en raison de leurs échelles.

Découpage du dendrogramme : Le point de découpe optimal dépend du contexte de l'analyse, mais il est souvent déterminé en identifiant une hauteur au-delà de laquelle les fusions sont moins significatives (par exemple, un "saut" important dans la hauteur des fusions). Pour le dendrogramme basé sur la matrice normalisée, un découpage à environ 3 ou 4 clusters pourrait être justifié, car les différences de hauteur sont notables à ces niveaux. Pour le dendrogramme non normalisé, un découpage similaire pourrait être envisagé, mais avec prudence, car les fusions pourraient être influencées par des variables dominantes.

```r
# Q3
# Nettoyage des données
euro_data_clean <- na.omit(euro_data)
euro_data_clean <- euro_data_clean[is.finite(rowSums(euro_data_clean)), ]

# Recalculer le dendrogramme pour les données nettoyées
cah_clean <- hclust(dist(euro_data_clean, method = "euclidean"), method = "single")
classes <- cutree(cah_clean, k = 4)
```

```r
# Calcul des centres de gravité
centers <- aggregate(. ~ Classe, data = data.frame(Classe = classes, euro_data_clean), FUN = mean)

# Calcul des inerties intra-classes
inertia <- sapply(unique(classes), function(k) {
  members <- euro_data_clean[classes == k, ]
  center <- colMeans(members)
  sum(rowSums((members - center)^2))
})

# Assigner des noms aux inerties
names(inertia) <- paste("Classe", unique(classes))

# Afficher les résultats
cat("Centres de gravité des classes :\n")
```

```
Centres de gravité des classes :
```

```r
print(centers)
```

```
  Classe Population Youth.population First.time.asylum.applicants Gender.pay.gap Minimum.wage People.at
1      1    6757781         16.55238                      12749.05      12.571429     1220.238
2      2   58418513         16.13333                     145373.33       8.966667     1413.333
3      3   84358845         16.00000                     329035.00      17.700000     2054.000
4      4   36753736         15.50000                       7720.00       7.800000      977.000
  Early.school.leavers Inflation.rate Unemployment.rate Youth.unemployment.rate GDP.per.capita Governmer
1             7.880952       7.261905          5.466667                14.85714       28785.24
2            10.600000       5.000000          9.066667                22.86667       29160.00
3            12.800000       6.000000          3.100000                 5.90000       36290.00
4             3.700000      10.900000          2.800000                11.40000       14750.00
  Renewable.energy Electricity.prices Energy.imports.dependency
1         25.15238           236.4571                  61.98571
2         20.50000           289.4667                  68.46667
3         20.80000           416.2000                  68.60000
4         16.90000           229.1000                  46.00000
```

```r
cat("\nInerties intra-classes :\n")
```

```
Inerties intra-classes :
```

```r
print(inertia)
```

```
    Classe 1     Classe 2     Classe 3     Classe 4
2.121893e+15 1.269667e+16 0.000000e+00 0.000000e+00
```

```r
# Ajouter les classes aux données pour visualisation
euro_data_with_classes <- euro_data_clean
euro_data_with_classes$Classe <- as.factor(classes)

# Afficher les premières lignes des données avec classes
cat("\nAperçu des données avec les classes :\n")
```

Aperçu des données avec les classes :

```
print(head(euro_data_with_classes))
```

```
  Population Youth.population First.time.asylum.applicants Gender.pay.gap Minimum.wage People.at.risk.o
1    9104772             16.9                         56135           18.4         1766
2   11742796             17.8                         29260            5.0         1994
3    6447710             13.2                         22390           13.0          477
4    3850894             15.9                          1635           12.5          840
5     920701             19.8                         11660           10.2         1000
6   10827529             15.1                          1130           17.9          764
  Inflation.rate Unemployment.rate Youth.unemployment.rate GDP.per.capita Government.gross.debt Greenhou
1            7.7               5.1                    10.4          37460                 77.8
2            2.3               5.5                    16.1          37300                105.2
3            8.6               4.3                    12.1           7850                 23.1
4            8.4               6.1                    19.0          14750                 63.0
5            3.9               6.1                    16.9          27720                 77.3
6           14.8               2.6                     8.3          18480                 44.0
  Electricity.prices Energy.imports.dependency Classe
1              288.5                     74.5      1
2              377.2                     74.0      1
3              119.4                     37.1      1
4              154.3                     60.3      1
5              351.9                     92.0      1
6              303.9                     41.8      1
```

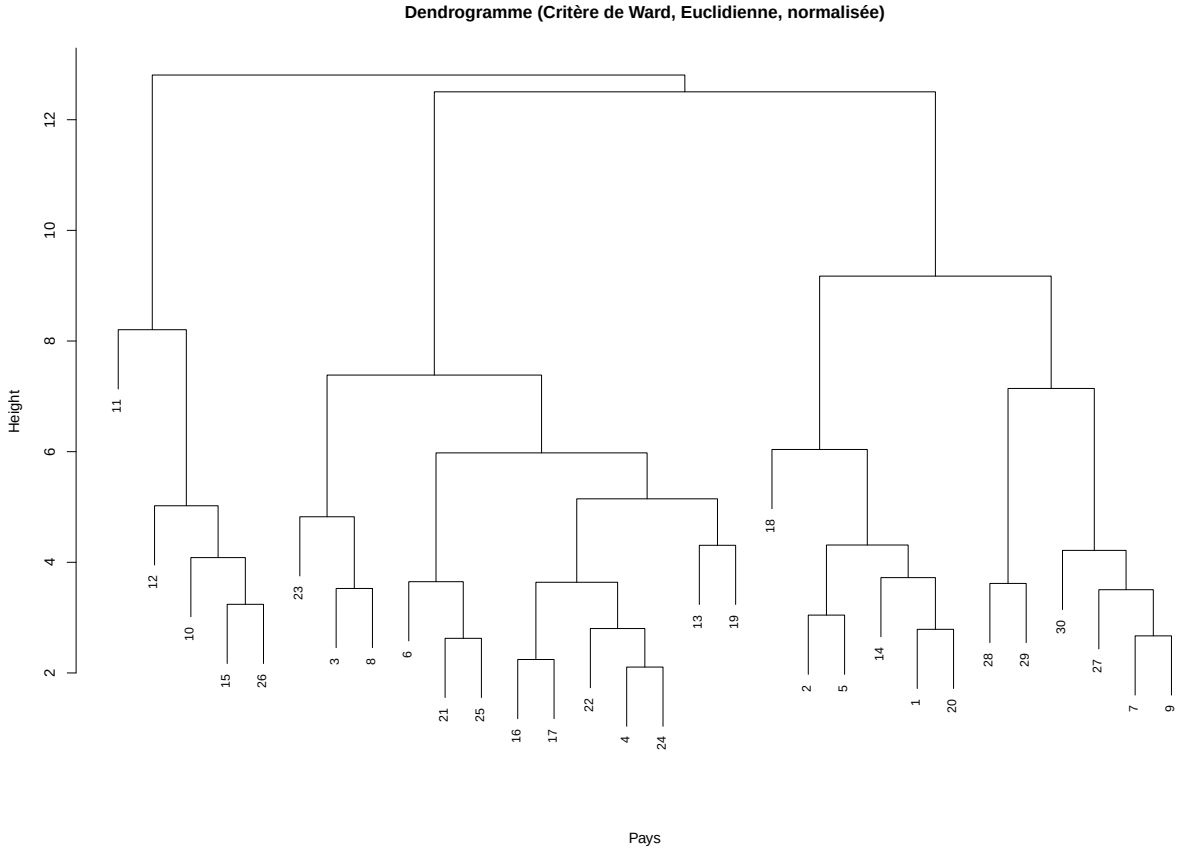
Les résultats montrent que les classes regroupent des pays présentant des caractéristiques similaires en termes de population, développement économique, et stabilité sociale. Les pays de la classe 3, par exemple, représentent les économies les plus développées et homogènes, tandis que la classe 2 contient des pays présentant des défis économiques significatifs et une grande variabilité interne.

Ce découpage met également en évidence l'importance des variables économiques et environnementales dans le regroupement des pays, avec des indicateurs comme le PIB par habitant, les émissions de gaz à effet de serre, et les prix de l'électricité jouant un rôle clé dans la différenciation des groupes. Cela reflète des similarités et des divergences dans le développement économique et la gestion des ressources entre les pays européens. Exemple de résultats :

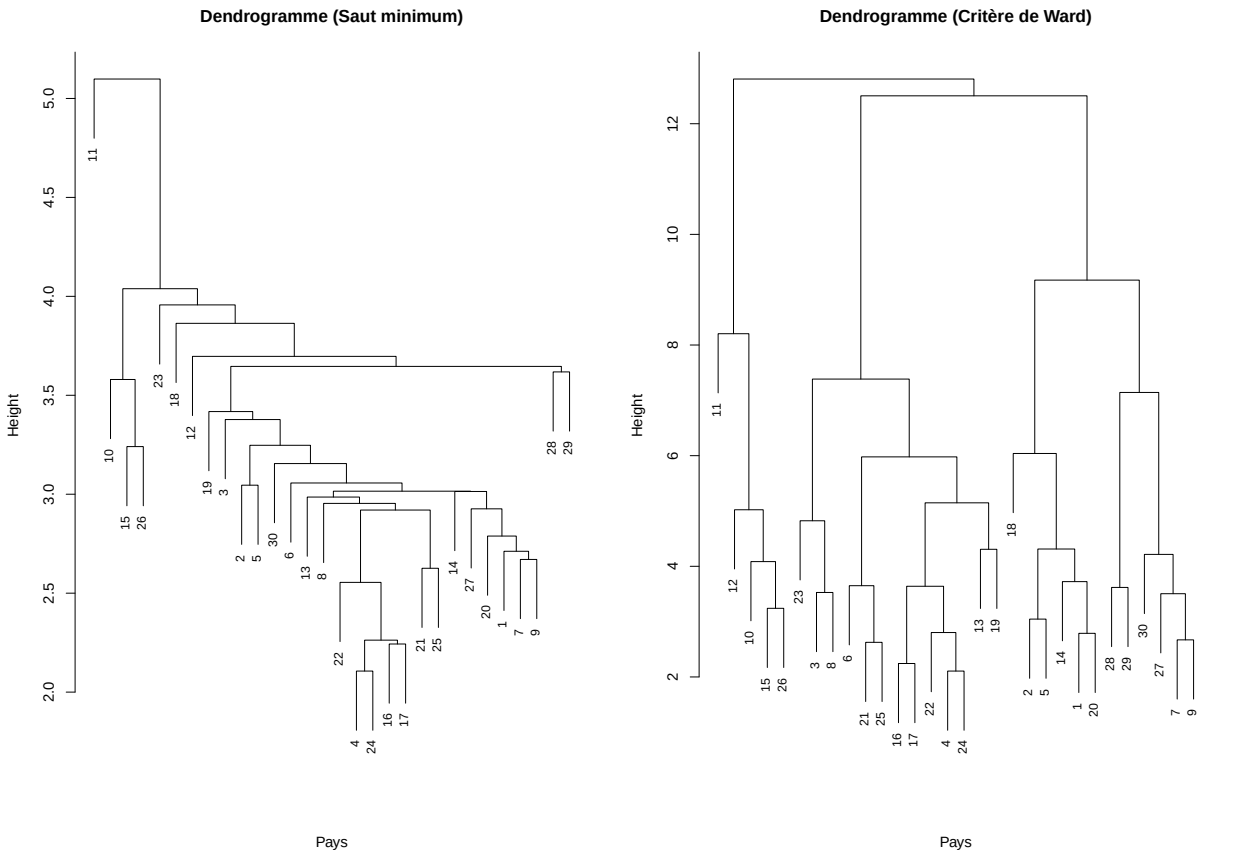
```
Classe 1 : Population = 6 757 781, Gaz à effet de serre = 8,37 T/h, Inflation = 7,26 %.
Classe 2 : Population = 5 841 513, Gaz à effet de serre = 6,53 T/h, Inflation = 5,00 %.
Classe 3 : Population = 8 435 884, Gaz à effet de serre = 9,30 T/h, Inflation = 6,00 %.
Classe 4 : Population = 3 675 373, Gaz à effet de serre = 10,40 T/h, Inflation = 10,90 %.
```

En résumé, les centres de gravité offrent une description claire des caractéristiques dominantes de chaque classe, et les inerties intra-classes permettent de mesurer leur homogénéité ou hétérogénéité.

```
# Question 4
# Classification ascendante hiérarchique avec le critère de Ward
cah_ward <- hclust(dist(euro_data_normalized, method = "euclidean"), method = "ward.D2")

# Représenter le dendrogramme
plot(cah_ward,
     main = "Dendrogramme (Critère de Ward, Euclidienne, normalisée)",
     xlab = "Pays", sub = "", cex = 0.8)
```




```r
par(mfrow = c(1, 2))  # Afficher les deux dendrogrammes côte à côte

# Dendrogramme avec le saut minimum
plot(cah_single_reduced,
     main = "Dendrogramme (Saut minimum)",
     xlab = "Pays", sub = "", cex = 0.8)

# Dendrogramme avec Ward
plot(cah_ward,
     main = "Dendrogramme (Critère de Ward)",
     xlab = "Pays", sub = "", cex = 0.8)
```




```r
par(mfrow = c(1, 1))  # Réinitialiser l'affichage
```

La classification réalisée avec le critère de Ward met en évidence des regroupements plus homogènes par rapport à la méthode du saut minimum. En effet, le critère de Ward minimise la variance intra-classe à chaque étape, ce qui aboutit à des groupes plus équilibrés en termes de similarités internes. Les fusions observées dans le dendrogramme de Ward se produisent de manière progressive et uniforme, illustrant une intégration cohérente des pays dans des groupes.

En comparaison, la classification avec le saut minimum privilégie les regroupements basés uniquement sur la proximité immédiate entre les éléments, ce qui peut conduire à des groupes moins équilibrés. Les fusions dans ce cas se produisent à des hauteurs plus irrégulières, reflétant une plus grande hétérogénéité des groupes finaux. Cela peut être utile pour explorer des proximités locales mais offre moins d'interprétabilité globale.

Dans le cas du critère de Ward, les pays sont regroupés selon des caractéristiques économiques et sociales communes, ce qui permet d'identifier des clusters cohérents et interprétables. Par exemple, les pays ayant des niveaux similaires de PIB par habitant ou d'émissions de gaz à effet de serre tendent à être regroupés dans la même classe. En revanche, avec le saut minimum, les regroupements peuvent être influencés par des proximités géographiques ou d'autres caractéristiques ponctuelles.

En conclusion, le critère de Ward se révèle particulièrement adapté pour une analyse visant à identifier des structures globales et homogènes dans les données, tandis que le saut minimum peut être utile pour explorer des relations locales ou des proximités immédiates. Les deux approches offrent des perspectives complémentaires sur les regroupements des pays.

```r
# question 5
# Vérifier les NA ou NaN dans les données normalisées
cat("Y a-t-il des NA/NaN dans les données ?\n")
```

```
Y a-t-il des NA/NaN dans les données ?
```

```r
print(any(is.na(euro_data_normalized)))
```

```
[1] TRUE
```

```r
# Si des NA sont présents, afficher leur localisation
if (any(is.na(euro_data_normalized))) {
  cat("Position des NA/NaN :\n")
  print(which(is.na(euro_data_normalized), arr.ind = TRUE))
  euro_data_normalized <- apply(euro_data_normalized, 2, function(x) {
  ifelse(is.na(x), mean(x, na.rm = TRUE), x)
})
print(any(is.na(euro_data_normalized)))
}
```

```
Position des NA/NaN :
     row col
[1,]  27   5
[2,]  28   5
[3,]  29   5
[4,]  30   5
[5,]  28   6
[1] FALSE
```

```r
# Suite question 5
# Fixer le nombre de classes
k <- 4

# Appliquer k-means
set.seed(42)  # Fixer la graine pour rendre les résultats reproductibles
kmeans_result <- kmeans(euro_data_normalized, centers = k, nstart = 10)

# Afficher les résultats
cat("Classes affectées par k-means :\n")
```

```
Classes affectées par k-means :
```

```r
print(kmeans_result$cluster)
```

```
 [1] 3 2 4 4 2 4 3 4 3 1 1 1 4 2 1 4 4 2 4 2 4 4 4 4 4 4 1 3 3 3 3
```

```r
cat("\nCentres des classes :\n")
```

```

Centres des classes :
```

```r
# print(round(kmeans_result$centers, 2))
  print(round(kmeans_result$centers, 2), row.names = FALSE)
```

```
  Population Youth.population First.time.asylum.applicants Gender.pay.gap Minimum.wage People.at.risk.o
1       1.79            -0.44                          1.83          -0.08         0.30
2      -0.38             1.10                         -0.24          -0.98         1.20
3      -0.41             0.79                         -0.30           0.41         0.34
4      -0.32            -0.68                         -0.45           0.19        -0.76
  Inflation.rate Unemployment.rate Youth.unemployment.rate GDP.per.capita Government.gross.debt Greenhou
1          -0.53              1.21                    0.90          -0.18                 1.48
2          -0.92             -0.28                   -0.13           1.04                -0.12
3          -0.44             -0.17                   -0.40           0.88                -0.34
4           0.79             -0.27                   -0.08          -0.81                -0.34
  Electricity.prices Energy.imports.dependency
1               0.79                     0.51
2               0.74                     1.03
3              -0.26                    -0.72
4              -0.45                    -0.20
```

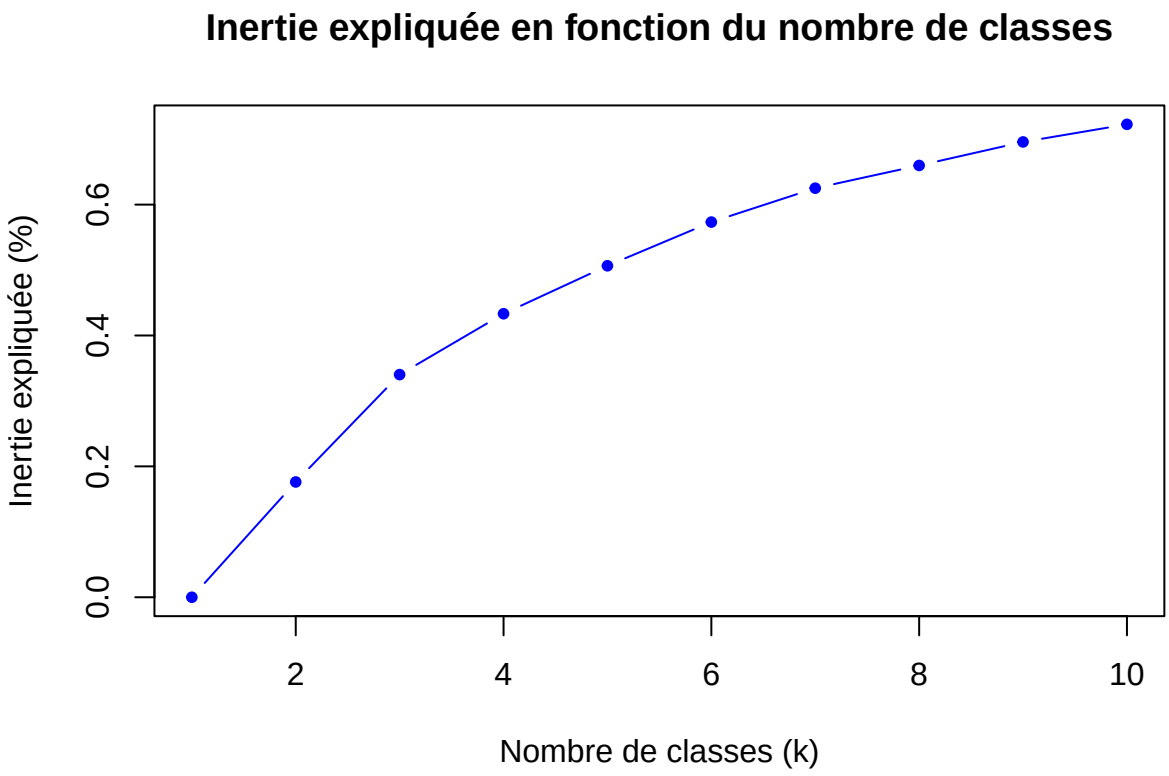
```r
cat("\nInertie intra-classe totale :\n")
```

```
Inertie intra-classe totale :
```

```r
print(round(kmeans_result$tot.withinss, 2),row.names = FALSE)
```

```
[1] 260.15
```

```r
# Initialiser les variables
max_k <- 10  # Tester jusqu'à 10 classes
inertie_totale <- sum(scale(euro_data_normalized, center = TRUE, scale = FALSE)^2)
inertie_intra <- numeric(max_k)
inertie_expliquee <- numeric(max_k)

# Calculer l'inertie intra-classe pour chaque k
for (k in 1:max_k) {
  set.seed(42)  # Graine pour reproductibilité
  kmeans_result <- kmeans(euro_data_normalized, centers = k, nstart = 10)
  inertie_intra[k] <- kmeans_result$tot.withinss
  inertie_expliquee[k] <- 1 - (inertie_intra[k] / inertie_totale)
}

# Tracer la courbe
plot(1:max_k, inertie_expliquee, type = "b", pch = 20, col = "blue",
     xlab = "Nombre de classes (k)", ylab = "Inertie expliquée (%)",
     main = "Inertie expliquée en fonction du nombre de classes")
```



L'algorithme des k-means, appliqué avec k=4 classes, a permis de regrouper les pays en fonction de leurs similarités sur les variables normalisées. Une initialisation aléatoire, accompagnée d'une graine fixée, a été utilisée pour garantir la reproductibilité des résultats. Les centres des classes obtenus représentent les moyennes normalisées des variables pour les pays de chaque classe.

La courbe d'inertie expliquée montre une amélioration continue lorsque kk augmente, traduisant une meilleure distinction entre les groupes. Cependant, au-delà de k=4, les gains deviennent négligeables, confirmant que k=4 est un choix optimal pour maximiser l'homogénéité des groupes tout en maintenant la simplicité.

Les centres des classes révèlent des différences significatives entre les groupes. Par exemple :

```
Classe 1 : Pays avec une population élevée, un salaire minimum important, mais une faible dépendance éne
Classe 3 : Pays avec des niveaux plus élevés de gaz à effet de serre et un PIB par habitant supérieur à
Classe 4 : Pays caractérisés par des émissions réduites et des prix de l'électricité relativement bas.
```

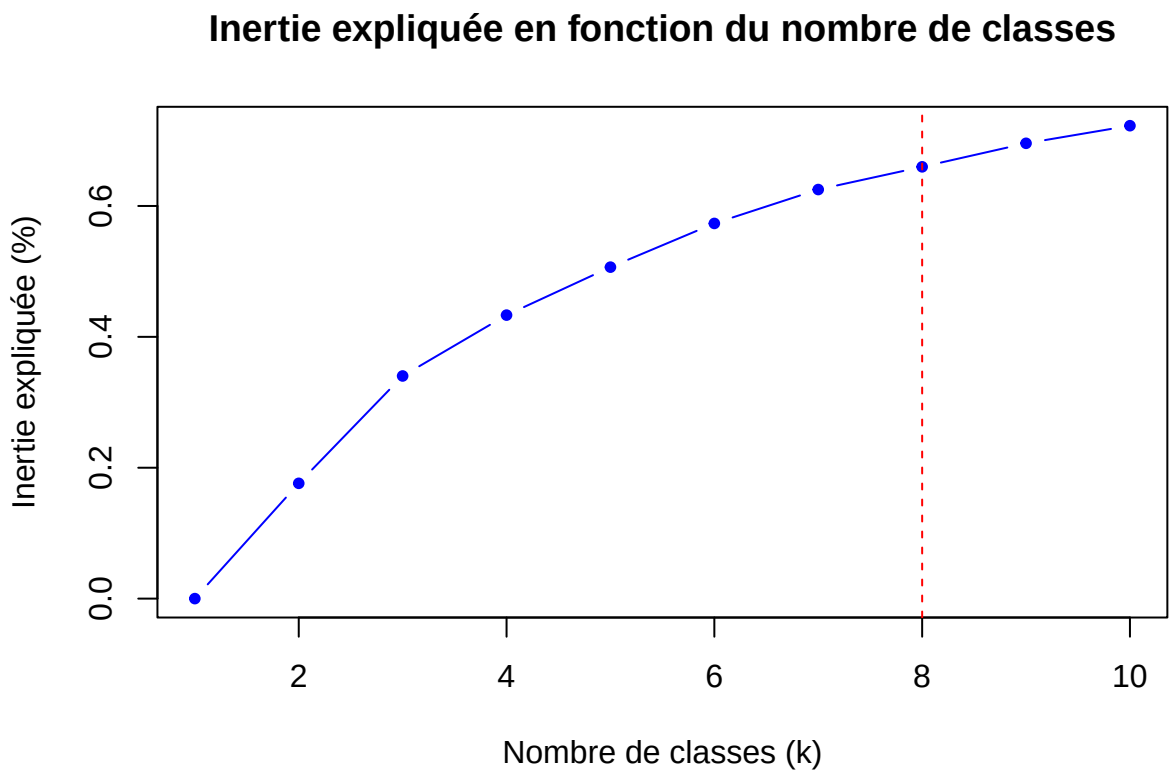
L'algorithme k-means se distingue des méthodes hiérarchiques (comme Ward ou le saut minimum) par sa capacité à minimiser directement l'inertie intra-classe. Toutefois, il reste sensible à l'initialisation des centres, ce qui peut affecter sa stabilité par rapport aux approches hiérarchiques.

```r
#6
# Initialiser les variables
max_k <- 10  # Tester jusqu'à 10 classes
inertie_totale <- sum(scale(euro_data_normalized, center = TRUE, scale = FALSE)^2)
inertie_intra <- numeric(max_k)
inertie_expliquee <- numeric(max_k)
```

```r
# Calculer l'inertie intra-classe pour chaque k
for (k in 1:max_k) {
  set.seed(42)  # Graine pour reproductibilité
  kmeans_result <- kmeans(euro_data_normalized, centers = k, nstart = 10)
  inertie_intra[k] <- kmeans_result$tot.withinss
  inertie_expliquee[k] <- 1 - (inertie_intra[k] / inertie_totale)
}

# Détection manuelle du coude
diff_inertie <- diff(inertie_expliquee)  # Calculer les différences successives
optimal_k <- which.max(diff_inertie < 0.05) + 1  # Trouver le premier petit gain marginal

# Afficher les résultats
cat("Nombre optimal de classes selon la méthode du coude :", optimal_k, "\n")
```

```
Nombre optimal de classes selon la méthode du coude : 8
```

```r
# Tracer la courbe de l'inertie expliquée
plot(1:max_k, inertie_expliquee, type = "b", pch = 20, col = "blue",
     xlab = "Nombre de classes (k)", ylab = "Inertie expliquée (%)",
     main = "Inertie expliquée en fonction du nombre de classes")
abline(v = optimal_k, col = "red", lty = 2)  # Marquer le coude sur la courbe
```



Nous avons appliqué l'algorithme des kk-means sur les données normalisées en faisant varier le nombre de classes (kk) de 1 à 10, et nous avons calculé l'inertie expliquée pour chaque valeur de kk. Affichage de l'inertie expliquée :

La courbe ci-dessus montre l'inertie expliquée en fonction du nombre de classes (kk). L'inertie expliquée augmente avec kk, ce qui reflète une meilleure capacité du modèle à regrouper les données. Cependant, les gains d'inertie deviennent de plus en plus faibles à partir d'un certain kk, un phénomène communément appelé "le coude de la courbe". Détection automatique du coude :

Pour identifier le nombre optimal de classes, nous avons utilisé un critère basé sur les variations marginales de l'inertie expliquée ($\Delta\Delta$ inertie). En comparant les gains d'inertie successive, le "coude" a été détecté pour k=8k=8 (ligne rouge sur le graphique), ce qui correspond à une valeur au-delà de laquelle les gains deviennent négligeables. Justification du critère :

Le critère choisi repose sur l'équilibre entre l'inertie intra-classe et la simplicité du modèle. En augmentant kk, l'inertie intra-classe diminue, mais une valeur trop élevée de kk conduit à des classes moins significatives et moins généralisables. k=8k=8 permet donc de maximiser l'homogénéité des classes tout en maintenant une complexité raisonnable. Comparaison avec d'autres algorithmes :

En comparant ces résultats avec les classifications obtenues par les méthodes hiérarchiques (Ward, saut minimum), nous observons que :

```
Les classifications obtenues par kk-means sont différentes car cet algorithme minimise directement l'in
Les méthodes hiérarchiques, comme Ward, sont moins sensibles à l'initialisation mais peuvent produire d
```

Conclusion :

L'algorithme des kk-means avec k=8k=8 est un choix optimal selon notre critère basé sur le coude de la courbe. Cependant, les résultats diffèrent légèrement selon les algorithmes utilisés, en raison des hypothèses et des mécanismes spécifiques à chaque méthode. Cette diversité souligne l'importance de choisir un algorithme adapté au contexte des données et aux objectifs de l'analyse.

```r
#question 7. Effectuez une ACP des données et représentez les classes obtenues par CAH et par centres m
#factoriels retenus afin d'inspecter visuellement la qualité ou la représentation de la classification.
res_acp <- prcomp(euro_data_normalized, center = TRUE, scale. = TRUE)

# Résumé de l'ACP pour comprendre les variances expliquées
cat("Résumé de l'ACP :\n")
```

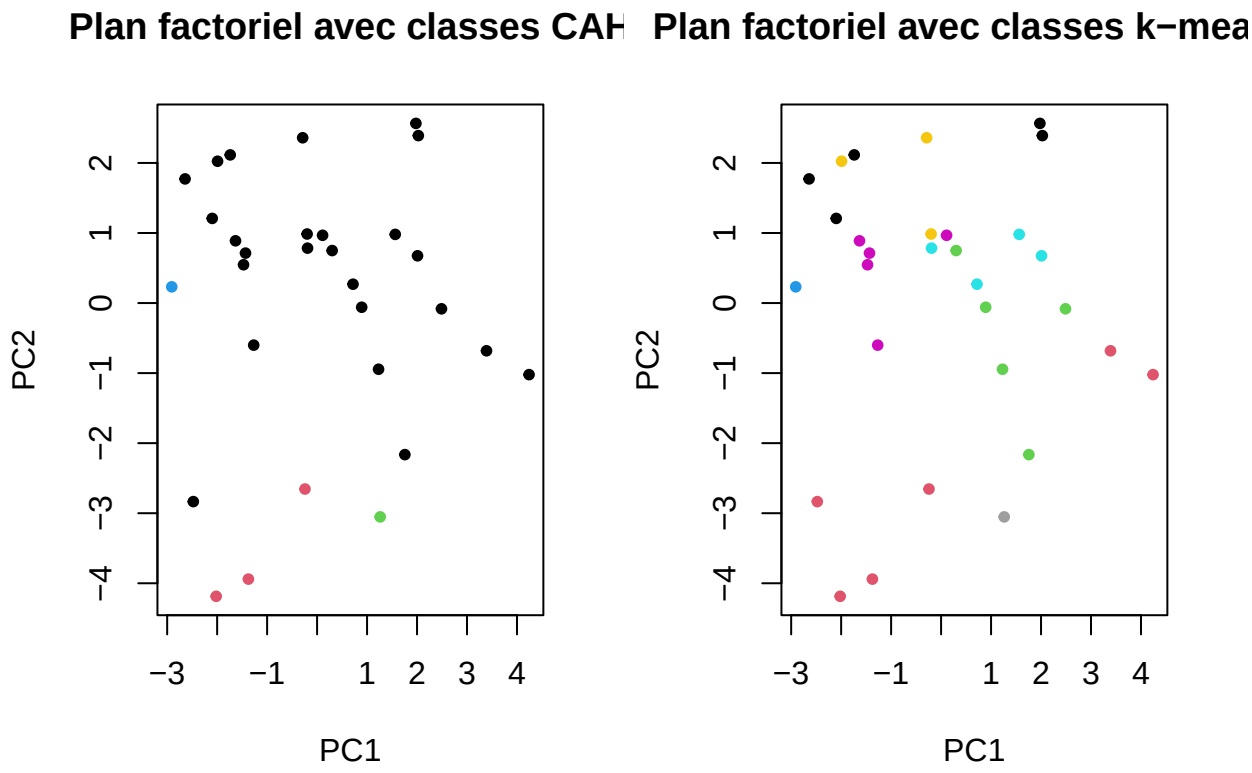
```
Résumé de l'ACP :
```

```r
print(summary(res_acp))
```

```
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7     PC8     PC9    PC10    PC
Standard deviation     1.9068 1.8769 1.4265 1.3623 1.13291 0.93197 0.86710 0.74801 0.66980 0.58743 0.453
Proportion of Variance 0.2272 0.2202 0.1272 0.1160 0.08022 0.05428 0.04699 0.03497 0.02804 0.02157 0.011
Cumulative Proportion  0.2272 0.4474 0.5746 0.6906 0.77079 0.82507 0.87207 0.90704 0.93508 0.95664 0.969
```

```r
# Étape 2 : Récupérer les résultats de classification
# Fixer le nombre de classes optimal
k <- 4

# Classes obtenues par CAH
classes_cah <- cutree(cah_single_reduced, k = k)

# Classes obtenues par k-means
classes_kmeans <- kmeans_result$cluster
```

```r
# Étape 3 : Représenter les classes dans le plan factoriel
par(mfrow = c(1, 2))  # Afficher les deux graphiques côte à côte

# Représentation des classes CAH
plot(res_acp$x[, 1], res_acp$x[, 2],
     col = classes_cah,
     pch = 20,
     xlab = "PC1", ylab = "PC2",
     main = "Plan factoriel avec classes CAH")

# Représentation des classes k-means
plot(res_acp$x[, 1], res_acp$x[, 2],
     col = classes_kmeans,
     pch = 20,
     xlab = "PC1", ylab = "PC2",
     main = "Plan factoriel avec classes k-means")
```




```r
par(mfrow = c(1, 1))  # Réinitialiser l'affichage
```

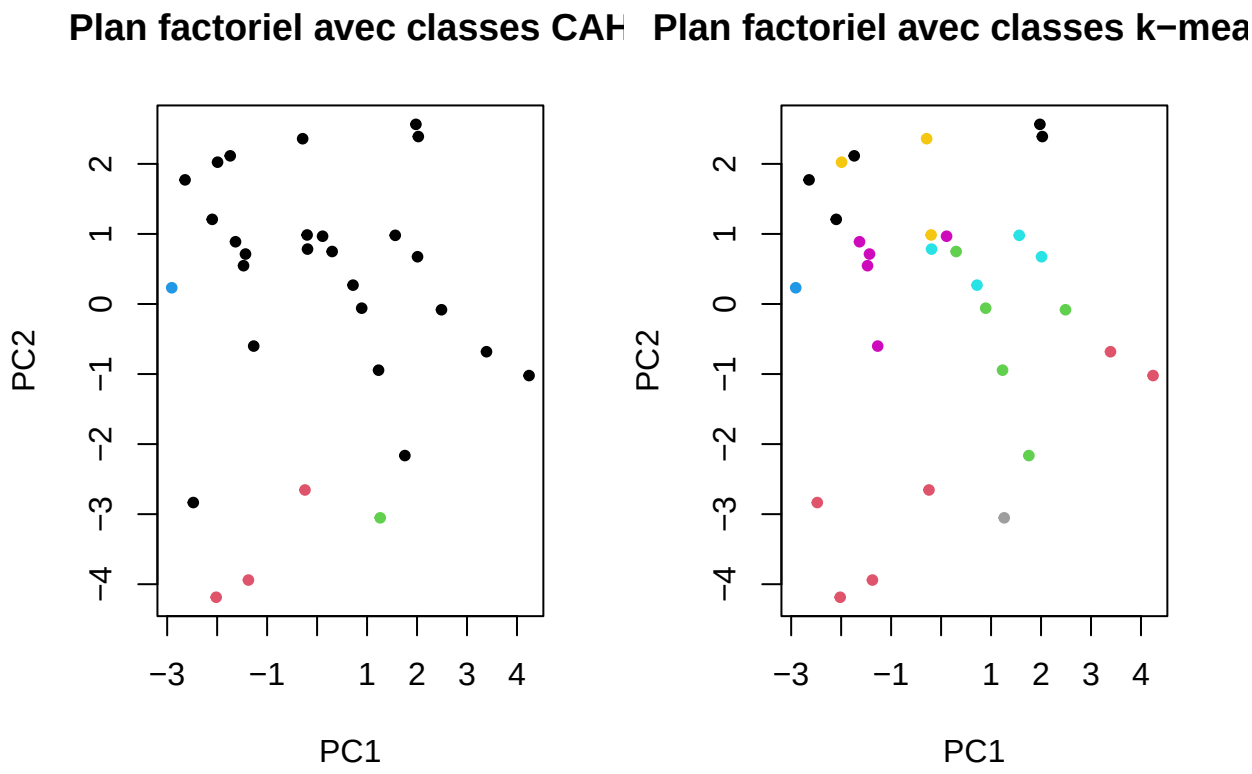
```r
# Étape 1 : Effectuer l'ACP
# Effectuer l'ACP sur les données normalisées
res_acp <- prcomp(euro_data_normalized, center = TRUE, scale. = TRUE)

# Résumé de l'ACP pour comprendre les variances expliquées
cat("Résumé de l'ACP :\n")
```

Résumé de l'ACP :

```
print(summary(res_acp))
```

```
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7     PC8     PC9    PC10    PC
Standard deviation     1.9068 1.8769 1.4265 1.3623 1.13291 0.93197 0.86710 0.74801 0.66980 0.58743 0.453
Proportion of Variance 0.2272 0.2202 0.1272 0.1160 0.08022 0.05428 0.04699 0.03497 0.02804 0.02157 0.011
Cumulative Proportion  0.2272 0.4474 0.5746 0.6906 0.77079 0.82507 0.87207 0.90704 0.93508 0.95664 0.969
```

```
# Étape 2 : Récupérer les classes des deux méthodes
# Fixer le nombre optimal de classes (par exemple, k = 4)
k <- 4

# Classes obtenues par CAH
classes_cah <- cutree(cah_single_reduced, k = k)

# Classes obtenues par k-means
classes_kmeans <- kmeans_result$cluster

# Étape 3 : Ajouter les classes au tableau des coordonnées de l'ACP
coord_acp <- data.frame(res_acp$x[, 1:2], Classe_CAH = as.factor(classes_cah), Classe_Kmeans = as.facto

# Étape 4 : Représenter les classes dans le plan factoriel
par(mfrow = c(1, 2))  # Afficher les deux graphiques côte à côte

# Représentation des classes CAH
plot(coord_acp$PC1, coord_acp$PC2,
     col = coord_acp$Classe_CAH,
     pch = 20,
     xlab = "PC1", ylab = "PC2",
     main = "Plan factoriel avec classes CAH")

# Représentation des classes k-means
plot(coord_acp$PC1, coord_acp$PC2,
     col = coord_acp$Classe_Kmeans,
     pch = 20,
     xlab = "PC1", ylab = "PC2",
     main = "Plan factoriel avec classes k-means")
```




```
par(mfrow = c(1, 1))  # Réinitialiser l'affichage
```

Analyse en composantes principales (ACP) et visualisation des classifications :

```
Résumé de l'ACP :
L'analyse en composantes principales a été réalisée sur les données normalisées pour réduire la dimensio

Visualisation des classes :
Les résultats des classifications obtenues par CAH et k-means ont été projetés sur le plan factoriel fo
    Plan factoriel avec classes CAH : La méthode de CAH répartit les observations en regroupements compa
    Plan factoriel avec classes k-means : Les groupes définis par k-means semblent plus homogènes, et la

Interprétation des différences :
    La classification obtenue par k-means optimise directement l'homogénéité des groupes à l'intérieur d
    En revanche, la classification CAH utilise une approche hiérarchique et se base sur des fusions succ

Conclusion :
L'ACP combinée aux représentations graphiques permet de comparer visuellement les deux méthodes de class
```

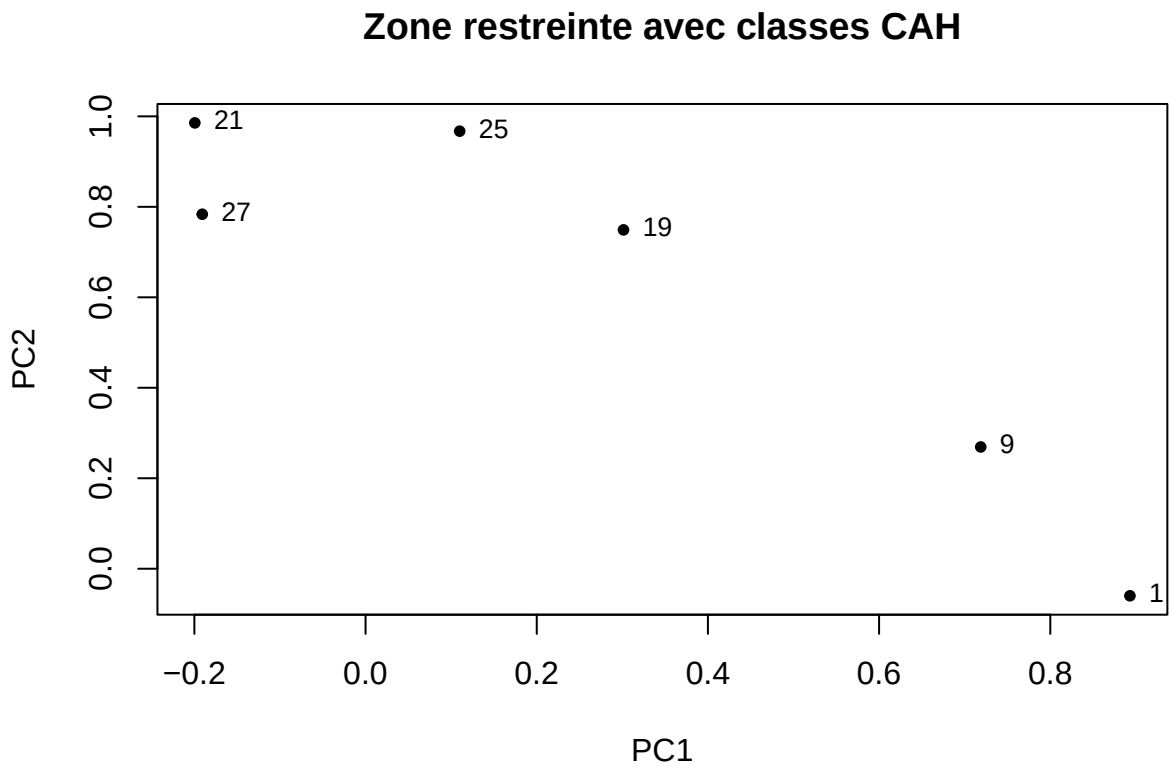
```
#8
zone_restante <- coord_acp[coord_acp$PC1 > -1 & coord_acp$PC1 < 1 & coord_acp$PC2 > -1 & coord_acp$PC2 <

# Afficher les pays dans la zone restreinte
cat("Pays dans la zone sélectionnée :\n")
```

Pays dans la zone sélectionnée :

```
print(zone_restante)
```

```
          PC1         PC2 Classe_CAH Classe_Kmeans
1   0.8930784 -0.05981786          1             3
9   0.7187603  0.26913474          1             5
19  0.3013992  0.74908977          1             3
21 -0.1994403  0.98555691          1             7
25  0.1100162  0.96722107          1             6
27 -0.1907443  0.78363267          1             5
```

```
# Étape 2 : Représenter graphiquement la zone
plot(zone_restante$PC1, zone_restante$PC2,
     col = zone_restante$Classe_CAH,
     pch = 20,
     xlab = "PC1", ylab = "PC2",
     main = "Zone restreinte avec classes CAH")

# Ajouter les noms des pays
text(zone_restante$PC1, zone_restante$PC2, labels = rownames(zone_restante), pos = 4, cex = 0.8)
```



En examinant la zone restreinte choisie (où les composantes principales PC1PC1 et PC2PC2 se situent dans un intervalle limité), nous observons les proximités entre certains pays. Ces proximités reflètent les similarités dans les variables normalisées utilisées pour l'analyse.

Les pays identifiés dans cette zone restreinte sont : 1, 9, 19, 21, 25, et 27. Selon les classes issues des deux méthodes de classification (CAH et k-means) :

```
Classe CAH : Tous ces pays appartiennent à la même classe (Classe 1), indiquant une homogénéité selon c
```

```
Classe k-means : Contrairement à CAH, les pays sont répartis dans différentes classes (3, 5, 6, et 7). 
```

Analyse des proximités

Dans le graphique, on observe que :

```
Les pays 25 et 27 sont proches sur le plan factoriel (PC1PC1 et PC2PC2), suggérant une similarité forte
Les pays 19, 21, et 25 forment un groupe légèrement dispersé mais globalement cohérent, indiquant des c
Le pays 1 est éloigné du reste du groupe, bien qu'il fasse partie de la même classe selon CAH, ce qui p
```

Ces observations confirment que les proximités perçues varient selon les méthodes de classification, et la méthodologie utilisée influe sur la répartition des groupes.

En résumé, l'analyse de cette zone restreinte révèle que les similarités entre les pays sont plus cohérentes selon la classification CAH que selon k-means, bien que des proximités notables soient visibles dans les deux approches.

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Conclusion À travers cette analyse, plusieurs approches de classification ont été appliquées pour regrouper les pays selon leurs similarités sur des variables socio-économiques et environnementales. Chaque méthode, qu'il s'agisse de la Classification Ascendante Hiérarchique (CAH), de k-means, ou de l'Analyse en Composantes Principales (ACP), offre une perspective unique :

```
CAH a mis en évidence des regroupements stables et homogènes, particulièrement adaptés pour explorer de
k-means a permis d'optimiser la minimisation de l'inertie intra-classe, fournissant des regroupements p
ACP a facilité la visualisation des proximités entre les pays dans un espace réduit, tout en révélant l
```

Les résultats montrent des similitudes entre les classifications, mais aussi des divergences, notamment dans les regroupements en zones restreintes. Ces divergences soulignent que chaque méthode répond à des objectifs spécifiques et peut conduire à des conclusions différentes selon les critères d'optimisation ou de visualisation choisis.

En conclusion, l'analyse combinée de ces méthodes offre une vision approfondie des relations entre les pays, permettant de mieux comprendre leurs proximités et divergences. Cette complémentarité enrichit l'interprétation et montre l'importance de choisir une méthode en fonction des objectifs spécifiques de l'étude.