BAZA Reem

2025-01-08

Question 1 ACP et k-means

Preparation de données

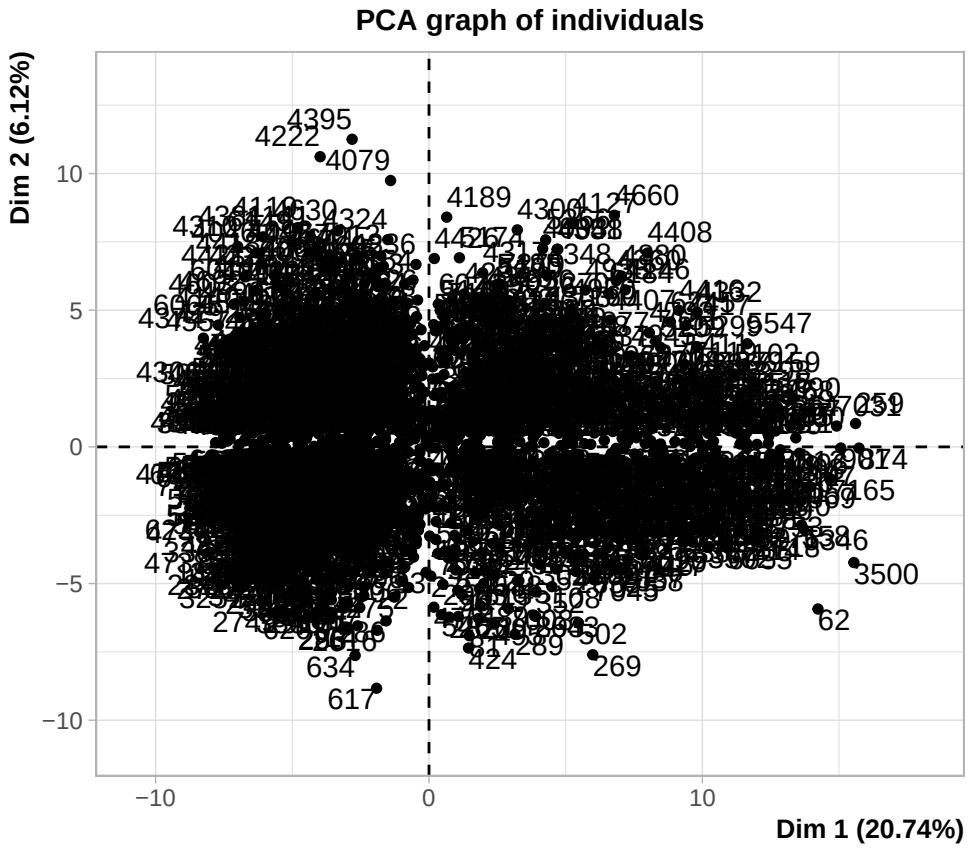
En hargeant les données et préparer l'ensemble de données en sélectionnant les 100 gènes les plus corrélés à l'indice de gravité.

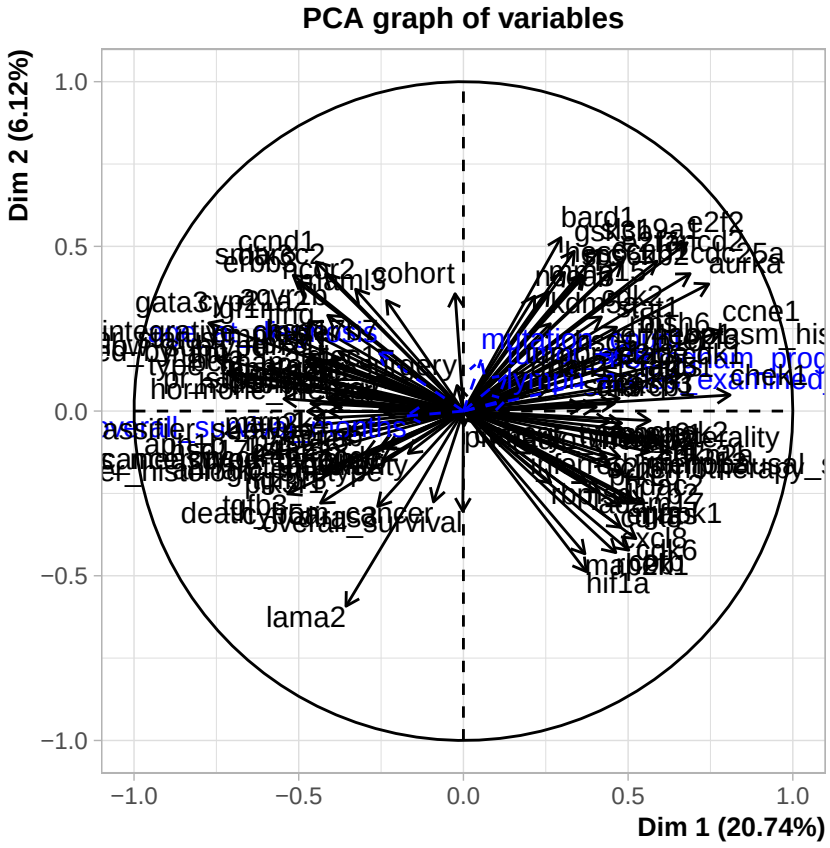
Analyse en Composantes Principales Normé

Pour réaliser une analyse en composantes principales (ACP) normalisée sur notre jeu de données, tout en conservant les variables médicales quantitatives comme variables supplémentaires pour l'interprétation. Nous avons traité les valeurs manquantes avec la méthode imputePCA, puis appliqué une ACP normalisée pour garantir une contribution équitable de toutes les variables.

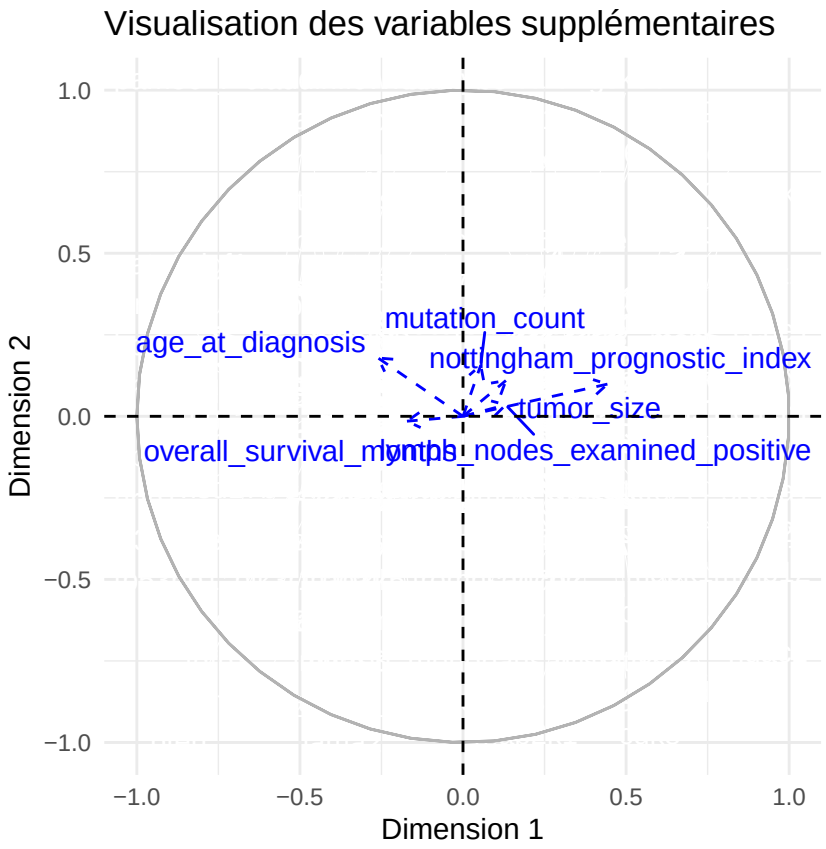
```
library(FactoMineR)
library(factoextra)
library(tidyverse)
library(missMDA)

genes.data-> data

# Indices des 30 premières colonnes
first_30_cols <- 1:30

# Les variables quantitatives
quanti_vars <- c(1, 19, 20, 21, 23, 28)

# Indices des colonnes qualitatives
quali_var <- setdiff(first_30_cols, quanti_vars)

# Extraire les colonnes qualitatives
quali_var <- data[, quali_var]

# Convertit les colonnes qualitatives en facteurs puis numériques
data <- data %>%
  mutate(across(where(is.character), as.factor)) %>%
  mutate(across(where(is.factor), as.numeric))

# Imputer les données manquantes
data <- imputePCA(data, quanti.sup = quanti_vars)$completeObs
data <- as.data.frame(data)
```

```r
# Réalisation de l'ACP
res.pca <- PCA(data, scale.unit = TRUE,
               quanti.sup = quanti_vars, # Variable supplementaire
               graph = TRUE)
```

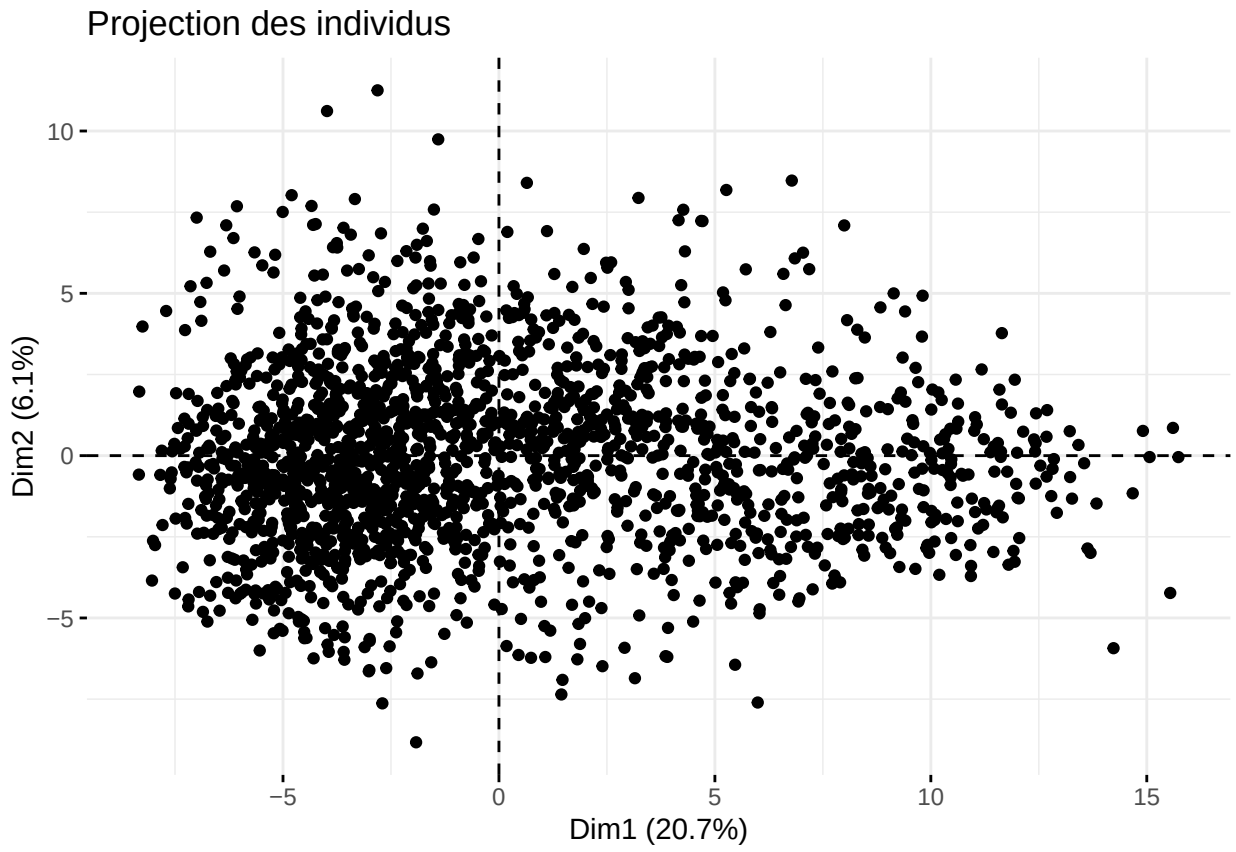




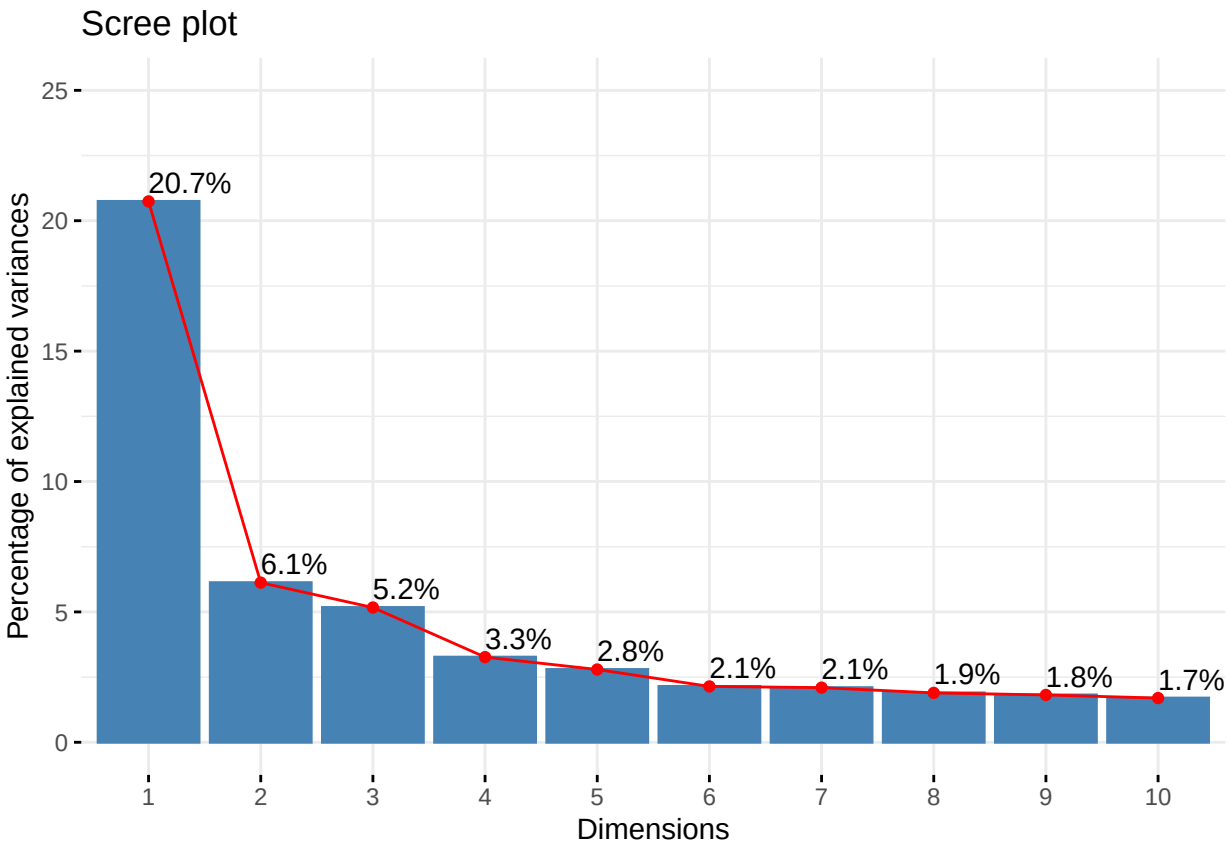

```
# Visualisation des résultats ------------------------------------------------
fviz_pca_var(res.pca,
             col.var = "transparent",                # Masquer les variables principales
             col.quanti.sup = "blue",                # Afficher les variables supplémentaires
             repel = TRUE                            # Éviter les chevauchements
) +
  labs(title = "Visualisation des variables supplémentaires",
       x = "Dimension 1", y = "Dimension 2") +
  theme_minimal()
```




```
fviz_pca_ind(res.pca,
             geom = "point",      # Affiche uniquement les points
             repel = TRUE,
             title = "Projection des individus")
```




```
#Visualisation les variables
fviz_eig(res.pca, addlabels = TRUE, ylim = c(0, 25),  linecolor = "red")
```



Choix du Nombre de Composantes

Le graphique montre les composantes principales et leur variance expliquée, utilisées pour identifier le nombre approprié de composantes à retenir. D'après le graphique, nous choisissons de conserver les 2 premières composantes principales, qui expliquent respectivement 20,7% et 6.1%, soit un total de 26,8% de la variance cumulée.

Ce choix est justifié par la méthode du coude, qui montre un ralentissement marqué dans la décroissance de la variance expliquée après la 2 composante. Ces deux composantes sont donc suffisantes pour une analyse simplifiée ou une visualisation. Au-delà de la troisième composante, la contribution à l'explication de la variance diminue considérablement, n'apportant qu'une information marginale.

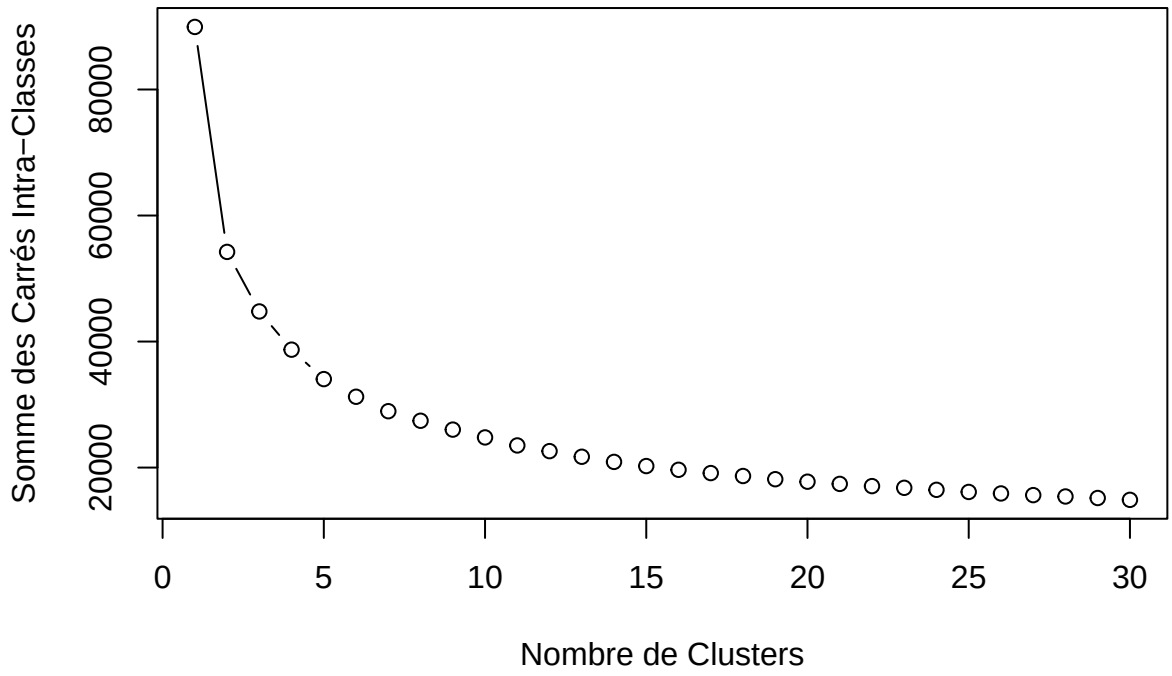
Question 2

Appliquer l'Algorithme des K-means

An appliquant l'algorithme k-means sur les vecteurs projetés obtenus de l'ACP pour différents nombres de clusters allant de 1 à 30. Nous observerons ensuite l'évolution de la somme des carrés intra-classes pour déterminer le nombre optimal de clusters.

```
library(cluster)

# Utiliser les coordonnées des individus
```

```r
pca_ind_coords  <- get_pca_ind(res.pca)$coord

# Application de k-means pour différents nombres de clusters
wcss <- numeric(30)
for (k in 1:30) {
  set.seed(123)  # Pour la reproductibilité
  clusters <- kmeans(pca_ind_coords , centers = k, nstart = 25)
  wcss[k] <- clusters$tot.withinss
}

# Tracé du critère du coude
plot(1:30, wcss, type = 'b', xlab = 'Nombre de Clusters', ylab = 'Somme des Carrés Intra-Classes')
```



Détermination du Nombre Optimal de Clusters

D'après le graphique du critère du coude, la somme des carrés intra-classes diminue rapidement au début, puis ralentit après un certain nombre de clusters. nous déterminerons le nombre optimal de clusters pour notre modèle k-means qui semble etre 5 classe dans notre cas.

Application du K-means pour le Nombre Optimal de Clusters

Apres avoir identifié le nombre de clustter, nous appliquons l'algorithme k-means à nos données pour ce nombre spécifique de clusters et classifions les patientes.

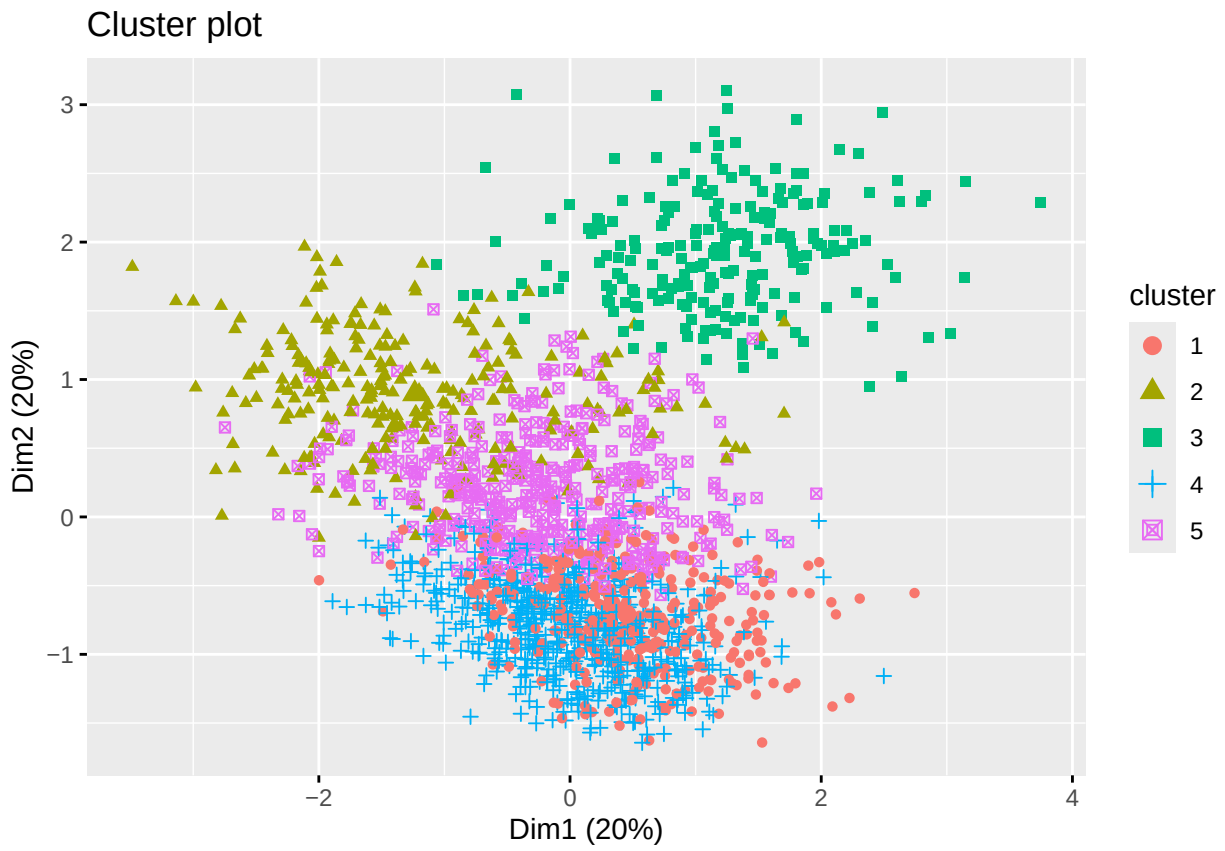
```r
k_optimal <- 5
set.seed(123)

# Appliquer k-means avec le nombre optimal de clusters
final_clusters <- kmeans(pca_ind_coords , centers = k_optimal, nstart = 25)
```

Question 3

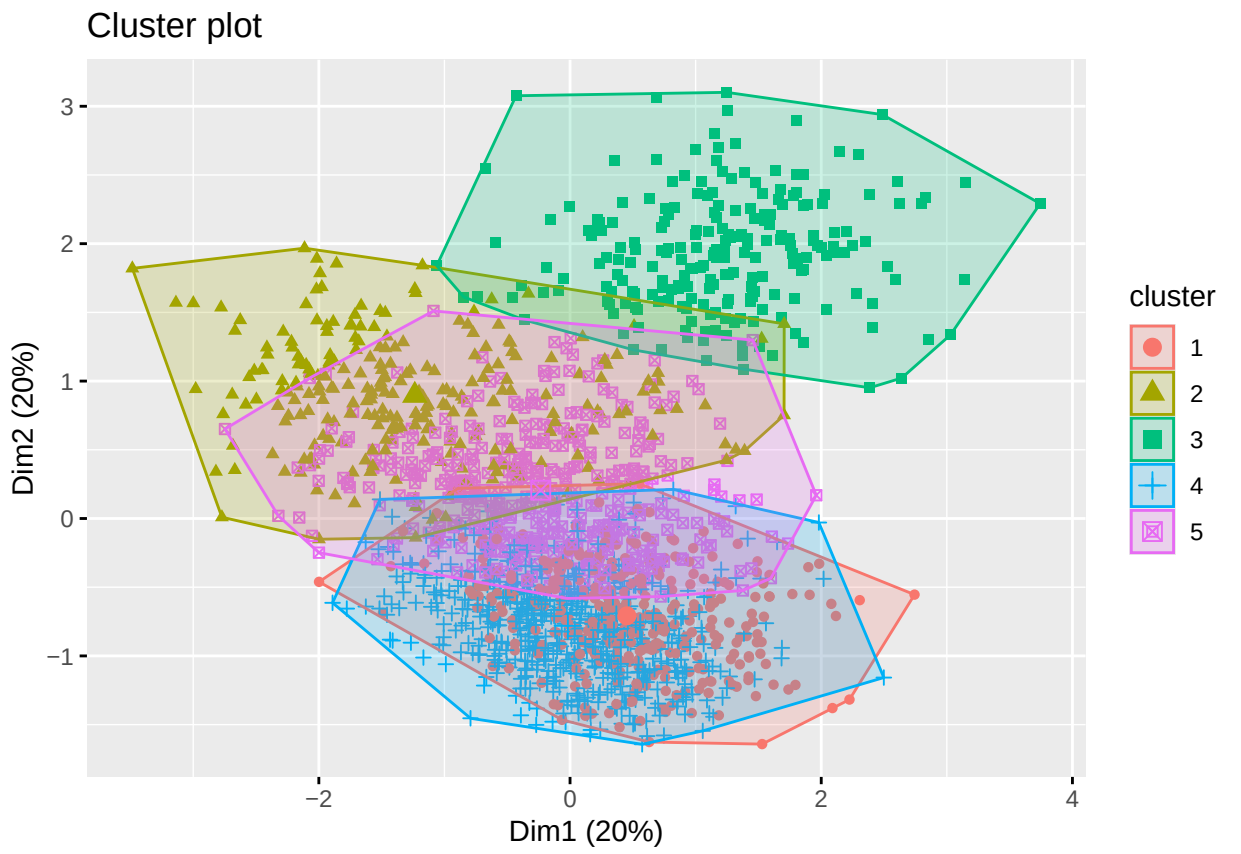
Tracé du Nuage de Points des Clusters dans le Plan Principal de l'ACP

Après avoir identifié les clusters, nous visualisons ces clusters dans le plan principal de l'ACP. Chaque cluster sera représenté par une couleur différente.

```r
# Ajouter les clusters aux données projetées
pca_scores_df <- as.data.frame(pca_ind_coords)
pca_scores_df$cluster <- as.factor(final_clusters$cluster)

# Affiche les clusters sur les données projetées
fviz_cluster(final_clusters, data = pca_scores_df[, -ncol(pca_scores_df)], geom = "point", ellipse = FA
```




```r
fviz_cluster(final_clusters, data = pca_scores_df[, -ncol(pca_scores_df)], geom = "point",ellipse.type =
```



Observation et Interprétation

Nous pouvons observer que Les clusters sont bien répartis dans l'espace défini par(Dim1 et Dim2), notamment les clusters (3 et 4), qui sont relativement bien séparés. Cependant, certains clusters, comme 2 et 5, présentent des chevauchements notables.

Cluster 1 : influencé par des valeurs négatives de Dim1. Ce cluster est influencé par des gènes comme MAPK1, Il est associé à des individus avec un faible overall_survival_months et un faible age_at_diagnosis. Ce groupe pourrait représenter des tumeurs diagnostiquées plus tôt dans la vie et ayant une progression rapide, se traduisant par une survie plus courte.

Cluster 2 : Ce cluster est influencé par des gènes proches du centre (faible contribution) ainsi que par des flèches de Dim2 positives, comme GATA3 et CCND1. Associé à des vriables comme age_at_diagnosis et mutation_count Ce groupe semble intermédiaire, sans caractéristiques marquées pour les variables supplémentaires.

Cluster 3 : Aligné avec les valeurs positives de Dim1 et Dim2 Il est influencé par des gènes tels que aurka et ccne1 Les variables supplémentaires tumor_size et lymph_nodes_examined_positive sont corrélées positivement. Ce groupe regroupe des patients présentant des caractéristiques tumorales plus agressives, avec des indices de Nottingham élevés, des tumeurs de grande taille et une forte implication ganglionnaire.

Cluster 4 : Aligné avec des valeurs négatives sur Dim1 et Dim2 Des gènes tels que lama2 ayant une contribution négative influencent ce cluster. Corrélation positivement avec overall_survival_months et negativement avec age_at_diagnosis . Ce cluster représente probablement des patients avec un bon pronostic.

Cluster 5: Situé autour du centre ce qui indique une influence équilibrée des gènes et des variables sur Dim1 et Dim2. Influence modérée par tumor_size et mutation_count. Ce groupe pourrait représenter

des individus avec des caractéristiques intermédiaires, ni fortement agressifs ni faiblement affectés par les variables supplémentaires.

Question 4: Interprétation des résultats

Analyse des Variables Médicales Qualitatives

En utilisant des tests du chi-deux, on peut identifier les variables médicales qualitatives les plus fortement associées aux classes déterminées par l'algorithme K-means.

```
# Effectuer le test du chi-deux pour chaque variable qualitative
  results_chi_sq <- lapply(quali_var, function(var) {
    chisq.test(table(final_clusters$cluster, var))
  })

  # Extraction des valeurs p pour chaque test
  p_values <- sapply(results_chi_sq, function(test) test$p.value)

  # Tri des variables par valeur de p
  sorted_vars <- names(sort(p_values))

  # Sélection des trois premières variables (celles avec les p-values les plus faibles)
  top_vars <- sorted_vars[1:3]

  # Création des tables de contingence pour ces variables
  tables_de_contingence <- lapply(top_vars, function(var) {
    table(final_clusters$cluster, quali_var[[var]])
  })

  # Affichage des tables de contingence
  tables_de_contingence
```

```
## [[1]]
##
##     Basal claudin-low Her2 LumA LumB  NC Normal
##   1     3           0   29  189  109   4     25
##   2    33          57  140    5    8   0     15
##   3   152          67    2    0    2   0      0
##   4     2          65    6  415   62   1     84
##   5     9          10   43   70  280   1     16
##
## [[2]]
##
##       1  10   2   3 4ER- 4ER+   5   6   7   8   9
##   1  19   0  19  51    1   30   6  14  71 125  23
##   2  10  16   3   7   54   23 126   4   2   0  13
##   3   6 189   1   0    8    2   1   0   1   1  14
##   4   9   0  15 203   11  158   6  18  76 129  10
##   5  88  14  34  21    0   31  45  48  32  34  82
##
## [[3]]
##
```

```
##     ER-/HER2- ER+/HER2- High Prolif ER+/HER2- Low Prolif HER2+
## 1           0                  175                  114     6
## 2          77                   24                    6   126
## 3         182                    2                    0     2
## 4          27                   68                  494     8
## 5           4                  334                    5    46
```

Interprétation des Tables de Contingence

L'analyse des tables de contingence révèle des informations significatives sur la relation entre les clusters identifiés et certaines variables médicales clés.

Table 1 (sous-groupes tumoraux): Cluster 1: Majorité dans les sous-types LumA (189) et LumB (109) Les individus du cluster 1 sont principalement associés à des tumeurs de types LumaA et LumaB,qui sont souvent des sous-types tumoraux hormonodépendants, Cluster 2: Ce cluster regroupe des tumeurs associées à des sous-types agressifs Cluster 3: Ce cluster est fortement lié aux tumeurs HER2-positives, et qui nécessitent des traitements spécifiques Cluster 4: Majorité écrasante dans le sous-type LumA (415). souvent considérées comme les tumeurs les plus favorables sur le plan pronostique en raison de leur sensibilité aux traitements hormonaux.

Table 2 (Statuts hormonaux): Cluster 1: Ce cluster regroupe des tumeurs avec un statut hormonal mixte, probablement des tumeurs ER+/PR+ cohérent avec des types comme LumA et LumB. Cluster 2: Les individus de ce cluster présentent des caractéristiques hormonales plus diversifiées, Cluster 3: Ce cluster est fortement lié aux tumeurs HER2-positives (ER-/HER2+), reflétant la nature spécifique de ces tumeurs. Cluster 5: Ce cluster inclut également des tumeurs luminales, mais probablement avec une certaine diversité dans leurs statuts hormonaux.

Table 3 (Prolifération tumorale et caractéristiques): Cluster 1: Ce cluster inclut des tumeurs luminales avec une forte prolifération(175- 114), ce qui peut indiquer un sous-type agressif malgré une sensibilité hormonale. Cluster 2: Ce cluster regroupe des tumeurs triple-négatives et HER2-positives, souvent associées à un mauvais pronostic ou à des traitements spécifiques (comme pour HER2). Cluster 3 :Cluster spécifique aux tumeurs HER2-positives, qui nécessitent des thérapies ciblées. Cluster 4: Ce cluster représente des tumeurs luminales avec une prolifération faible(494), souvent associées à un pronostic favorable. Cluster 5: Ce cluster regroupe des tumeurs luminales avec une forte prolifération(334), indiquant des caractéristiques intermédiaires ou plus agressives.

Question 5 Prédiction de l'indice de Nottingham

Preparation de données

En conservant uniquement les variables médicales, tout en excluant la colonne cible (qui correspond ici à l'Indice de Nottingham), et sans intégrer les gènes dans le modèle, l'objectif est de se concentrer sur les données cliniques et démographiques pertinentes pour construire une prédiction robuste et interprétable.

```r
library(caret)  # Pour la division train/test
library(dplyr)  # Pour manipuler les données

#choisir les variable medicale
data <- data[,1:30]
# Étape 1 : Identification des variables numériques et catégorielles
numeric_vars <- sapply(data, is.numeric)  # Colonnes numériques
categorical_vars <- sapply(data, is.factor) | sapply(data, is.character)  # Colonnes catégorielles
```

```r
# Étape 2 : Encodage des variables catégorielles (dummy encoding)
data_encoded <- data
data_encoded[, categorical_vars] <- lapply(data[, categorical_vars], as.factor)  # Convertir en facteurs
data_encoded <- model.matrix(~ . - 1, data = data_encoded)  # Dummy encoding, supprime intercept

# Étape 3 : Gestion des données manquantes
data_encoded <- as.data.frame(data_encoded)  # Conversion en data frame après encodage
data_encoded[is.na(data_encoded)] <- 0  # Remplacement des NA (imputation simple)

# Étape 4 : Division en train/test
set.seed(123)
trainIndex <- createDataPartition(data$nottingham_prognostic_index, p = 0.8, list = FALSE)
train_data <- data_encoded[trainIndex, ]  # 80% pour l'entraînement
test_data <- data_encoded[-trainIndex, ]  # 20% pour le test

# Séparer X (variables explicatives) et y (variable cible)
y_train <- data$nottingham_prognostic_index[trainIndex]
X_train <- train_data

y_test <- data$nottingham_prognostic_index[-trainIndex]
X_test <- test_data
```

Entraîner le modèle de régression linéaire

```r
# Étape 5 : Ajuster le modèle linéaire
train_data_model <- cbind(X_train, nottingham_prognostic_index = y_train)  # Fusionner X et y
model <- lm(nottingham_prognostic_index ~ ., data = train_data_model)

# Résumé du modèle
summary(model)
```

```
##
## Call:
## lm(formula = nottingham_prognostic_index ~ ., data = train_data_model)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.00791 -0.23925  0.06485  0.30938  1.45976
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    3.579e+00  6.520e-01   5.490 4.73e-08 ***
## age_at_diagnosis              -1.085e-05  1.941e-03  -0.006 0.995543
## type_of_breast_surgery         1.205e-01  3.803e-02   3.169 0.001559 **
## cancer_type                   -2.816e+00  5.747e-01  -4.901 1.06e-06 ***
## cancer_type_detailed          -5.257e-01  1.542e-01  -3.408 0.000672 ***
## cellularity                   -3.771e-02  1.615e-02  -2.335 0.019682 *
## chemotherapy                   2.669e-01  5.162e-02   5.171 2.64e-07 ***
## pam50_._claudin.low_subtype   -9.471e-03  1.258e-02  -0.753 0.451779
## cohort                         2.588e-02  1.326e-02   1.952 0.051154 .
## er_status_measured_by_ihc      6.537e-04  6.728e-02   0.010 0.992249
```

```
## er_status                          6.513e-03  6.905e-02   0.094 0.924859
## neoplasm_histologic_grade          9.792e-01  2.726e-02  35.921  < 2e-16 ***
## her2_status_measured_by_snp6      -1.323e-03  2.396e-02  -0.055 0.955962
## her2_status                       -1.026e-01  7.397e-02  -1.387 0.165653
## tumor_other_histologic_subtype    -9.044e-03  1.471e-02  -0.615 0.538782
## hormone_therapy                    2.620e-01  3.630e-02   7.220 8.27e-13 ***
## inferred_menopausal_state          5.063e-02  5.463e-02   0.927 0.354113
## integrative_cluster                3.195e-03  5.038e-03   0.634 0.526033
## primary_tumor_laterality           2.309e-02  3.022e-02   0.764 0.444898
## lymph_nodes_examined_positive      9.819e-02  4.242e-03  23.148  < 2e-16 ***
## mutation_count                     2.870e-03  3.833e-03   0.749 0.454134
## oncotree_code                      4.161e-01  1.195e-01   3.483 0.000511 ***
## overall_survival_months           -9.279e-05  2.347e-04  -0.395 0.692589
## overall_survival                   2.002e-01  7.504e-02   2.668 0.007709 **
## pr_status                         -9.972e-02  3.619e-02  -2.756 0.005929 **
## radio_therapy                      1.614e-01  3.843e-02   4.200 2.83e-05 ***
## X3.gene_classifier_subtype         4.353e-02  2.429e-02   1.792 0.073285 .
## tumor_size                        -2.209e-03  1.225e-03  -1.804 0.071450 .
## tumor_stage                        4.309e-01  3.371e-02  12.781  < 2e-16 ***
## death_from_cancer                 -1.249e-01  4.374e-02  -2.856 0.004349 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5691 on 1494 degrees of freedom
## Multiple R-squared:  0.7581, Adjusted R-squared:  0.7534
## F-statistic: 161.5 on 29 and 1494 DF,  p-value: < 2.2e-16
```

```
# Étape 6 : Validation sur le jeu de test
predictions <- predict(model, newdata = X_test)

# Calcul des métriques de performance
rmse <- sqrt(mean((predictions - y_test)^2))  # Root Mean Squared Error
mae <- mean(abs(predictions - y_test))       # Mean Absolute Error
cat("RMSE :", rmse, "\nMAE :", mae, "\n")
```

```
## RMSE : 0.5645604
## MAE : 0.3873166
```

Le modèle montre une bonne performance globale ($R^2 = 75{,}81\%$, RMSE = 0.5646). Cependant, certaines variables qualitatives ne sont pas significatives. En revanche, les variables age_at_diagnosis, cancer_type, hormone_therapy et tumor_stage jouent un rôle important.