Profils d'expression génique du cancer du sein

Chahbaoui Mohammed

Introduction

L'analyse de l'expression génique joue un rôle essentiel dans la compréhension des sous-types du cancer du sein. Ce rapport présente une analyse exploratoire des données METABRIC en utilisant l'Analyse en Composantes Principales (ACP) et le clustering k-means afin d'identifier des profils distincts de patients. Enfin, une régression linéaire est mise en place pour prédire l'indice de Nottingham.

Préparation des données

Données prises de kaggle: https://www.kaggle.com/datasets/raghadalharbi/breast-cancer-gene-expression-profiles-metabric

Les données METABRIC sont prétraitées pour sélectionner les gènes les plus corrélés au score de gravité de Nottingham.

```
metabric.data <- read.csv2("C:/Users/Moha/Documents/data/METABRIC_RNA_Mutation.csv"
                           ,header=TRUE,sep=",",row.names='patient_id')
genes.data <- metabric.data[,1:519]
genes.data <- type_convert(genes.data)
corre <- c()
for (i in c(1:489)){
  corre <- c(corre,cor(genes.data[,21],genes.data[,i+30])**2)
}
selected <- order(corre,decreasing = TRUE)[1:100]
genes.data <- genes.data[,c(1:30,30+selected)]
```

Analyse en Composantes Principales (ACP) et Clustering k-means

Analyse en Composantes Principales

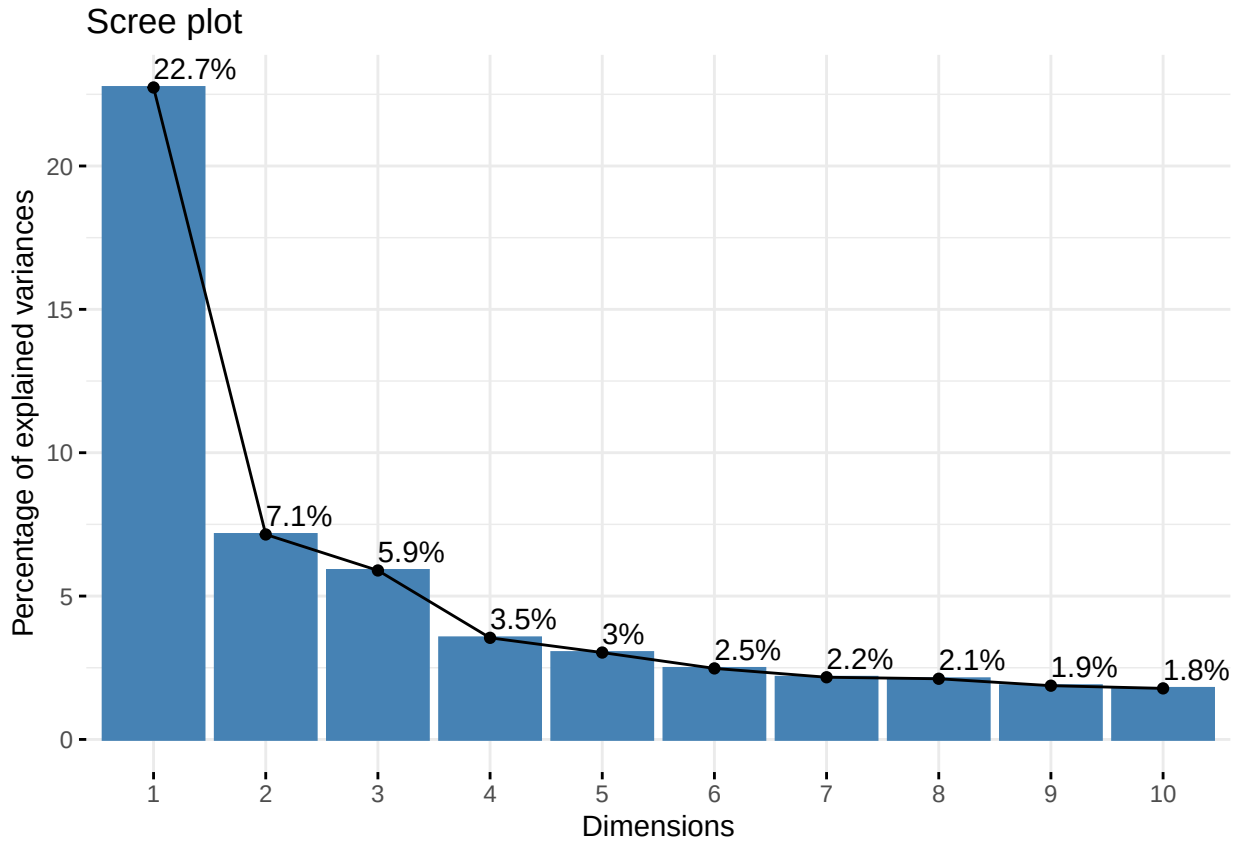
L'ACP est appliquée afin de réduire la dimensionnalité des données. Les variables médicales quantitatives et qualitatives sont ajoutées en variables supplémentaires.

```
quanti_supp <- c(1,19,20,21,23,28)
quali_supp <- setdiff(1:30, quanti_supp)
genes_pca <- PCA(genes.data, scale.unit = TRUE, ncp = 100, quanti.sup = quanti_supp, quali.sup = quali_
genes_pca$eig[1:10, 1:3]
```

```
##        eigenvalue percentage of variance cumulative percentage of variance
```

```
## comp 1   22.737987              22.737987                    22.73799
## comp 2    7.148879               7.148879                    29.88687
## comp 3    5.893729               5.893729                    35.78059
## comp 4    3.543957               3.543957                    39.32455
## comp 5    3.030587               3.030587                    42.35514
## comp 6    2.477902               2.477902                    44.83304
## comp 7    2.169085               2.169085                    47.00213
## comp 8    2.116721               2.116721                    49.11885
## comp 9    1.875241               1.875241                    50.99409
## comp 10   1.780638               1.780638                    52.77473
```

```r
fviz_eig(genes_pca, addlabels = TRUE)
```



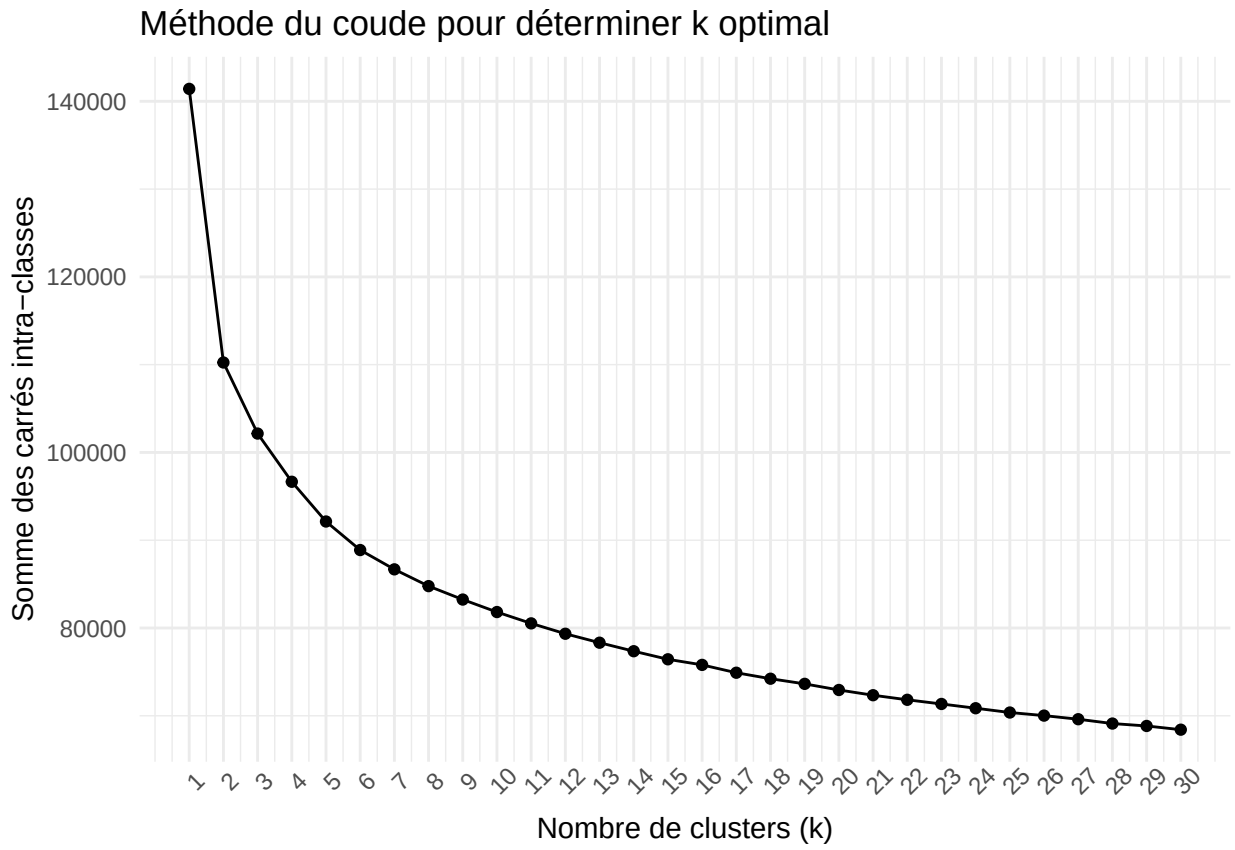
Le choix du nombre optimal de composantes principales est basé sur le pourcentage de variance cumulée, avec 31 composantes expliquant 74.27% de la variance.

```r
composantes <- 31
```

Clustering k-means

Le clustering k-means est appliqué sur les coordonnées des individus dans l'espace réduit de l'ACP. La méthode du coude est utilisée pour déterminer le nombre optimal de clusters.

```r
coord_pca <- genes_pca$ind$coord[, 1:composantes]
inertia <- c()
for (k in 1:30) {
  kmeans_res <- kmeans(coord_pca, centers = k, nstart = 25, iter.max = 100)
  inertia[k] <- kmeans_res$tot.withinss
}

elbow_data <- data.frame(k = 1:30, inertia = inertia)
ggplot(elbow_data, aes(x = k, y = inertia)) +
  geom_line() +
  geom_point() +
  labs(title = "Méthode du coude pour déterminer k optimal", x = "Nombre de clusters (k)", y = "Somme d
  theme_minimal() +
  scale_x_continuous(breaks = 1:30) +
  theme(axis.text.x = element_text(angle = 45))
```



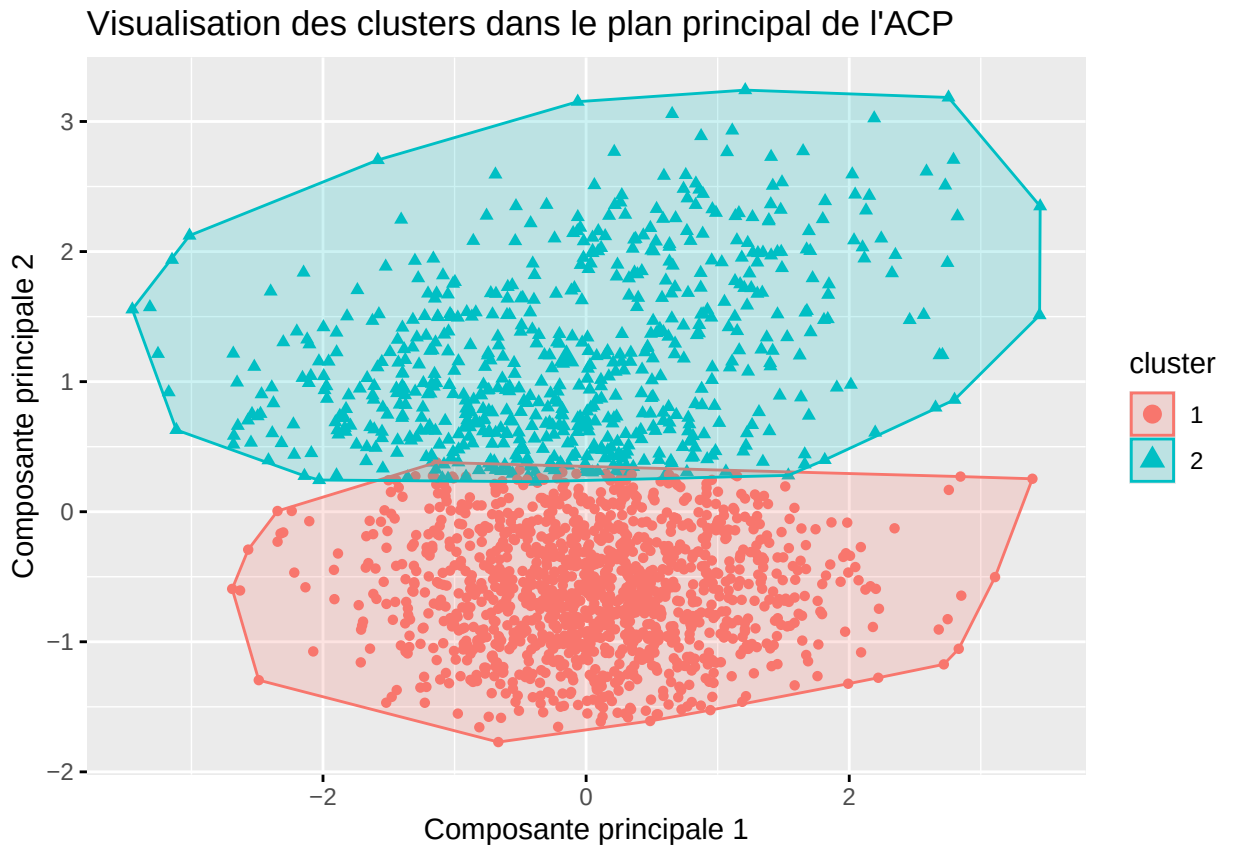
Le choix optimal du nombre de clusters est k = 2.

```r
k_optimal <- 2
kmeans_final <- kmeans(coord_pca, centers = k_optimal, nstart = 25)
cluster <- kmeans_final$cluster
genes.data$cluster <- as.factor(cluster)
```

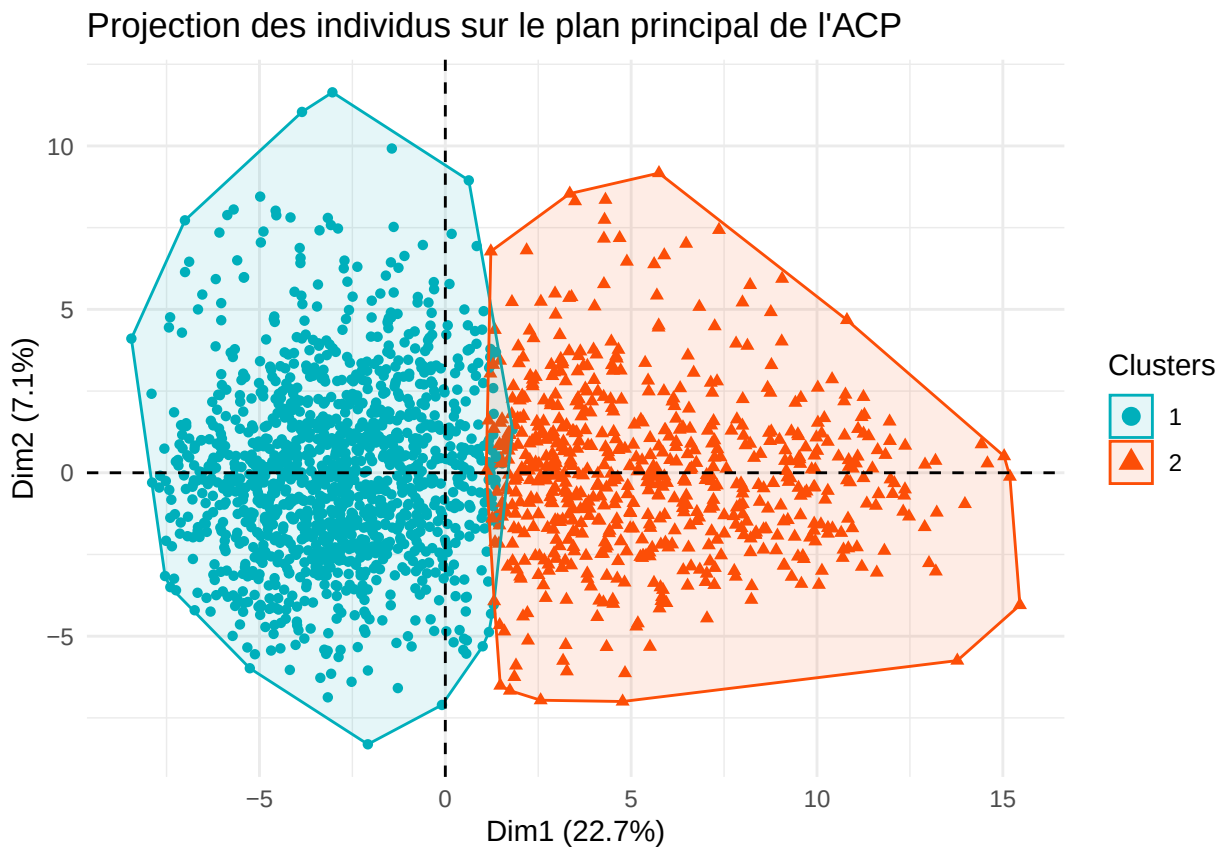
Visualisation des clusters

Les clusters sont représentés graphiquement sur les deux premières composantes principales.

```
fviz_cluster(kmeans_final, data = coord_pca, geom = "point", ellipse.type = "convex") +
  labs(title = "Visualisation des clusters dans le plan principal de l'ACP", x = "Composante principale
```




```
fviz_pca_ind(genes_pca, geom = "point", col.ind = genes.data$cluster, palette = c("#00AFBB", "#FC4E07")
```



Les clusters identifiés semblent correspondre aux principaux sous-types moléculaires du cancer du sein :

- **Cluster 1** : cancers luminal (ER+/PR+)
- **Cluster 2** : cancers basal-like ou triple négatif (ER-/PR-)

Interprétation des résultats

Analyse des associations avec les variables médicales

L'impact des variables qualitatives médicales sur la répartition des clusters est évalué via un test du chi-deux.

```
chi_square_results <- data.frame(variable = character(), chi_square = numeric(), p_value = numeric(), s
for(i in quali_supp) {
  cont_table <- table(genes.data[,i], cluster)
  chi_test <- chisq.test(cont_table)
  chi_square_results <- rbind(chi_square_results, data.frame(variable = names(genes.data)[i], chi_squar
}

chi_square_results <- chi_square_results[order(-chi_square_results$chi_square),]
print(head(chi_square_results, 3))
```

```
##                                variable chi_square       p_value
## X-squared15          integrative_cluster  1031.7187 2.744257e-215
## X-squared5  pam50_._claudin.low_subtype   942.9134 1.979987e-200
## X-squared21  X3.gene_classifier_subtype   898.4478 1.921835e-194
```

```r
for(var in head(chi_square_results$variable, 3)) {
  cat("\nTable de contingence pour", var, ":\n")
  print(table(cluster, genes.data[[var]]))
}
```

```
##
## Table de contingence pour integrative_cluster :
##
## cluster   1  10   2   3 4ER- 4ER+   5   6   7   8   9
##       1  73   3  63 271   21  207  35  54 174 284  69
##       2  59 216   9  11   53   37 149  30   8   5  73
##
## Table de contingence pour pam50_._claudin.low_subtype :
##
## cluster Basal claudin-low Her2 LumA LumB  NC Normal
##       1     7          74   52  664  325   6    126
##       2   192         125  168   15  136   0     14
##
## Table de contingence pour X3.gene_classifier_subtype :
##
## cluster ER-/HER2- ER+/HER2- High Prolif ER+/HER2- Low Prolif HER2+
##       1        36                   444                  616    33
##       2       254                   159                    3   155
```

Prédiction de l'indice de Nottingham

Un modèle linéaire est construit pour prédire l'indice de Nottingham en utilisant les composantes principales sélectionnées.

```r
# Limiter les données aux 61 premières colonnes (30 variables médicales + 31 composantes principales)
data <- genes.data[, 1:61]

# Suppression de la variable 'cancer_type' si elle est constante
if ("cancer_type" %in% colnames(data)) {
  data <- subset(data, select = -cancer_type)
}

# Séparer les données en ensembles d'entraînement (85%) et de test (15%)
set.seed(123)  # Pour assurer la reproductibilité
n <- nrow(data)
train_indices <- sample(1:n, size = 0.85 * n)
train_data <- na.omit(data[train_indices, ])
test_data <- na.omit(data[-train_indices, ])

# Construire un modèle linéaire pour prédire l'indice de Nottingham
model <- lm(nottingham_prognostic_index ~ ., data = train_data)

# Faire des prédictions sur l'ensemble de test
test_predictions <- predict(model, test_data)

# Calculer les métriques d'évaluation
```

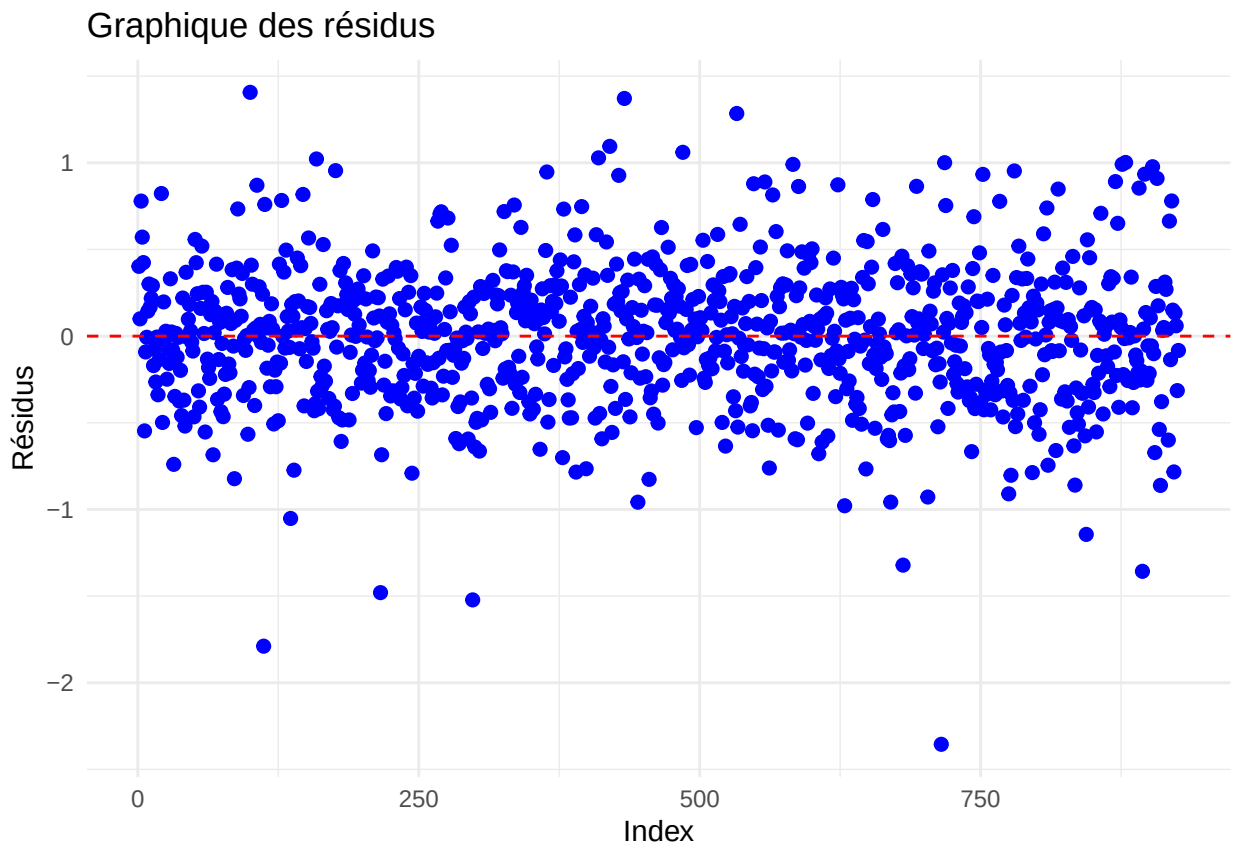
```r
test_rmse <- sqrt(mean((test_predictions - test_data$nottingham_prognostic_index)^2))
r2 <- cor(test_data$nottingham_prognostic_index, test_predictions)^2

# Afficher les résultats
cat("Test RMSE:", round(test_rmse, 3), "\n")
```

```
## Test RMSE: 0.382
```

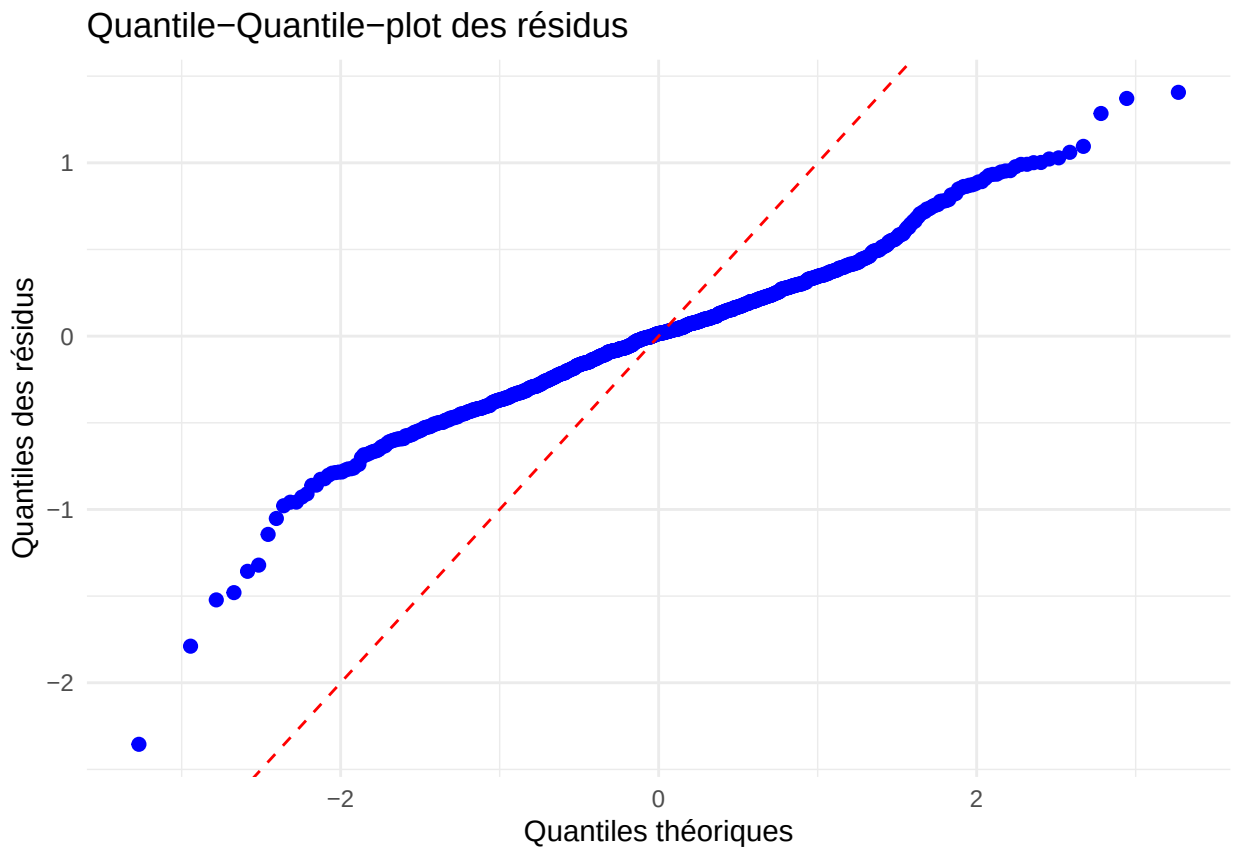
```r
cat("R²:", round(r2, 3), "\n")
```

```
## R²: 0.862
```

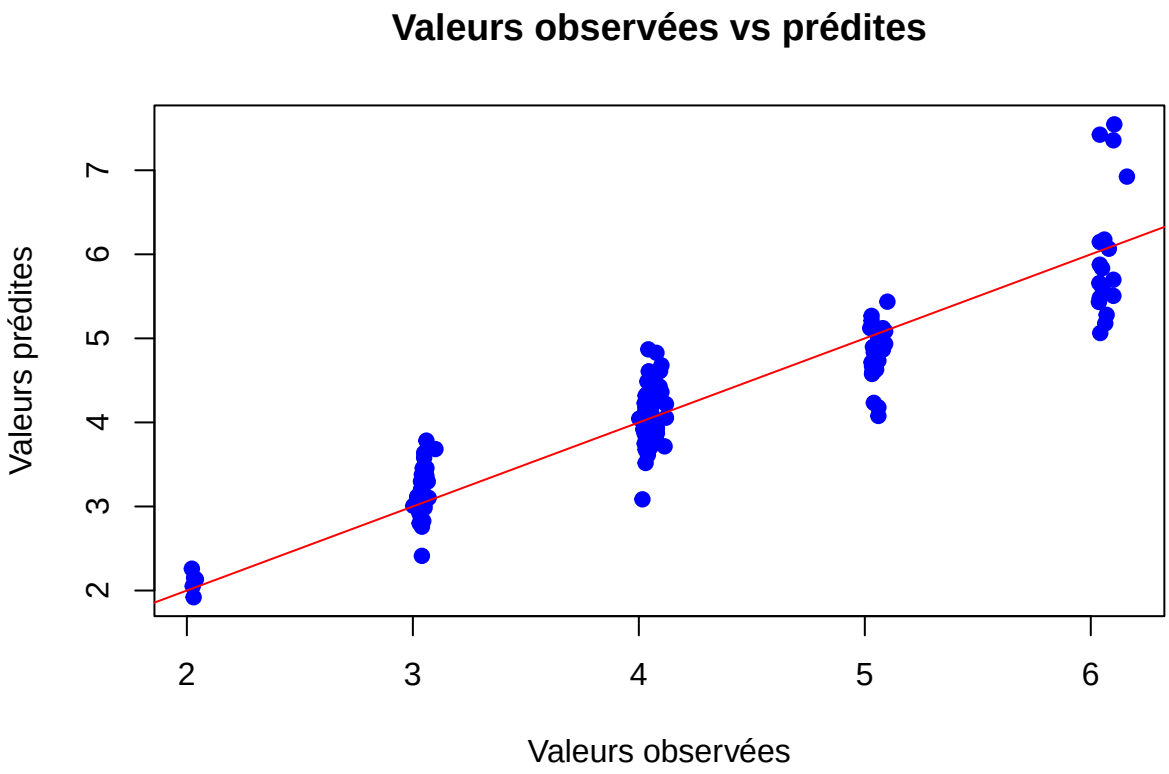
```r
# Tracer les résidus et la QQ-plot pour vérifier les diagnostics du modèle

# Résidus vs index
residuals_df <- data.frame(
  Index = 1:length(resid(model)),
  Residuals = resid(model)
)

ggplot(residuals_df, aes(x = Index, y = Residuals)) +
  geom_point(color = "blue", size = 2) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(
    title = "Graphique des résidus",
    x = "Index",
    y = "Résidus"
  ) +
  theme_minimal()
```




```r
# QQ-plot pour les résidus
qqplot_data <- data.frame(
  Theoretical = qqnorm(resid(model), plot.it = FALSE)$x,
  Sample = qqnorm(resid(model), plot.it = FALSE)$y
)

ggplot(qqplot_data, aes(x = Theoretical, y = Sample)) +
  geom_point(color = "blue", size = 2) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  labs(
    title = "Quantile-Quantile-plot des résidus",
    x = "Quantiles théoriques",
    y = "Quantiles des résidus"
  ) +
  theme_minimal()
```




```r
# Comparer les valeurs prédites et observées
plot(test_data$nottingham_prognostic_index, test_predictions,
     main = "Valeurs observées vs prédites",
     xlab = "Valeurs observées", ylab = "Valeurs prédites",
     pch = 19, col = "blue")
abline(0, 1, col = "red")
```



Les résultats indiquent une corrélation modérée entre les variables explicatives et l'indice de Nottingham.

Conclusion

Cette analyse a permis d'identifier deux clusters distincts de patients atteints de cancer du sein et de prédire l'indice de Nottingham avec une précision modérée.