TP Noté Fondements Statistiques - ANDREOLLI Justine M1 SDSC

Préparation du jeu de données

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.4     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
metabric.data <- read.csv2('./data/METABRIC_RNA_Mutation.csv',header=TRUE,sep=",",
                           row.names='patient_id')
# supprimer les données de mutation que nous n'allons pas utiliser
genes.data <- metabric.data[,1:519]
# Conversion des colonnes au bon type
genes.data <- type_convert(genes.data)
```

```
##
## ── Column specification ────────────────────────────────────────────────────────────────────────
──
## cols(
##   .default = col_double(),
##   type_of_breast_surgery = col_character(),
##   cancer_type = col_character(),
##   cancer_type_detailed = col_character(),
##   cellularity = col_character(),
##   pam50_._claudin.low_subtype = col_character(),
##   er_status_measured_by_ihc = col_character(),
##   er_status = col_character(),
##   her2_status_measured_by_snp6 = col_character(),
##   her2_status = col_character(),
##   tumor_other_histologic_subtype = col_character(),
##   inferred_menopausal_state = col_character(),
##   integrative_cluster = col_character(),
##   primary_tumor_laterality = col_character(),
##   oncotree_code = col_character(),
##   pr_status = col_character(),
##   X3.gene_classifier_subtype = col_character(),
##   death_from_cancer = col_character()
## )
## ℹ Use `spec()` for the full column specifications.
```

```
# Première diminution de dimension par selection des $100$ gènes
#les plus corrélées au score de gravité de Nottingham
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)]
```

ACP et k-means

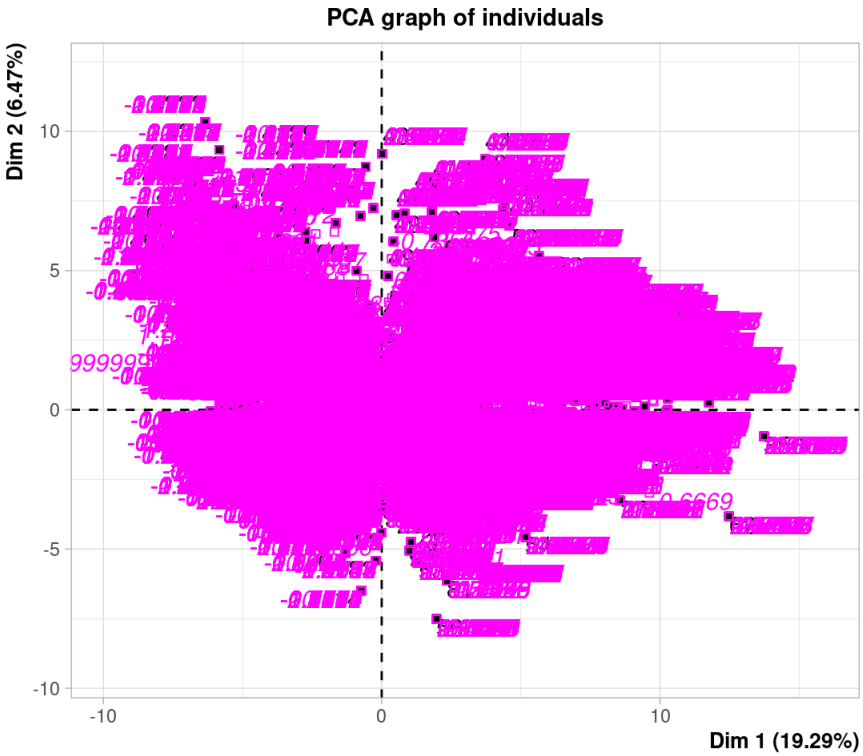
Question 1

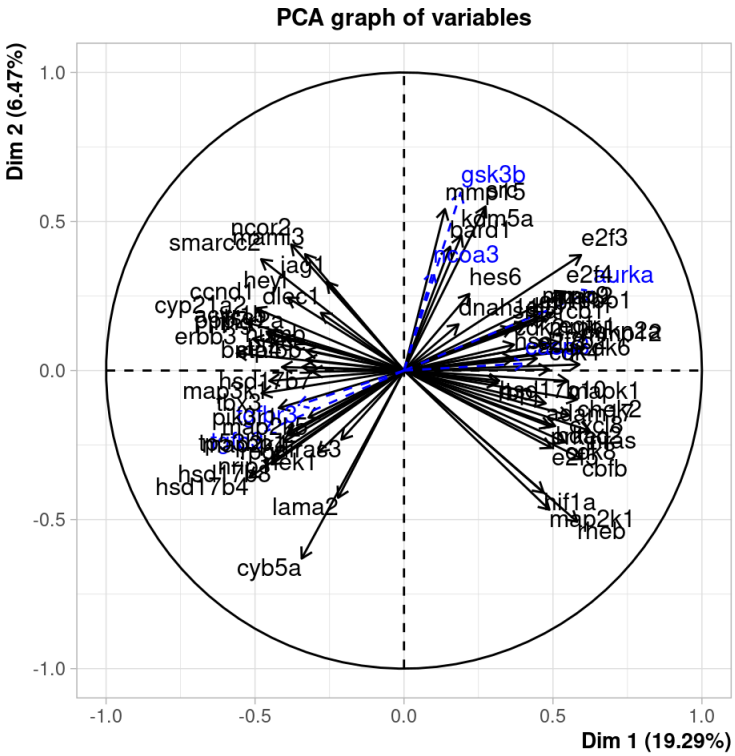
1. Réaliser une ACP normée sur les données des 100 gènes considérées en gardant les variables médicales comme variables supplémentaires d'interprétation (les variables 1, 19, 20, 21, 23 et 28 sont quantitatives, les autres sont qualitatives. Vous justifierez le nombre de composantes gardées.

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

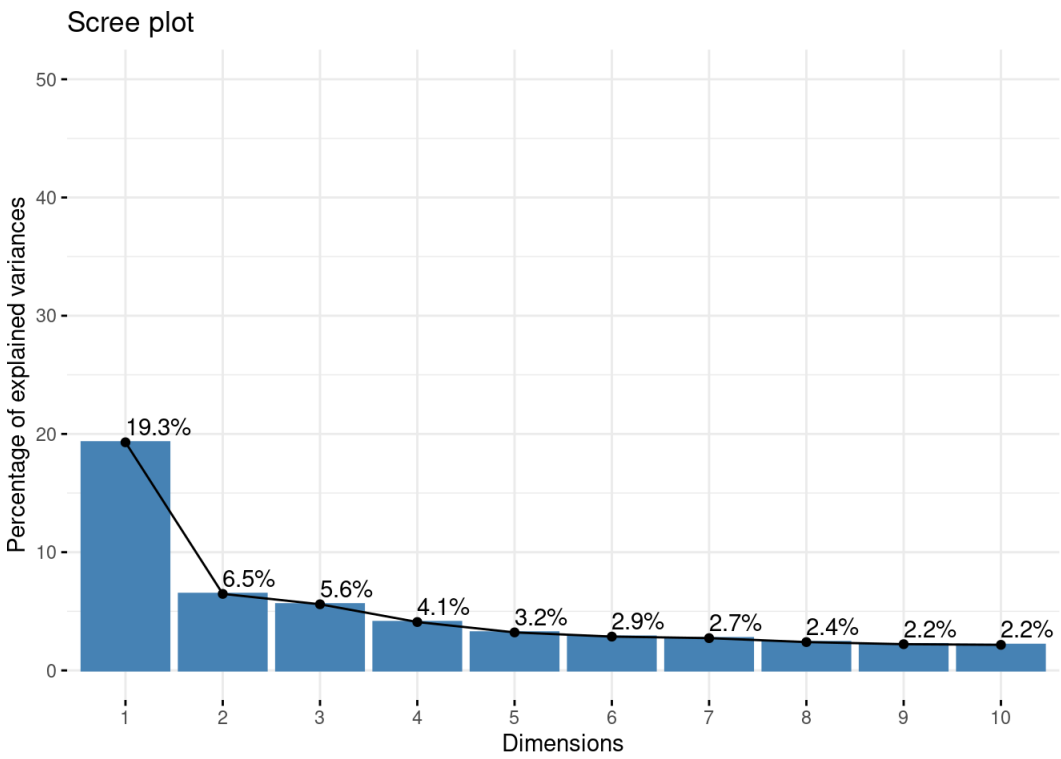
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
acp_rslt <- PCA(genes.data[, -(1:30)],
                scale.unit = TRUE,  # on normalise les données
                quanti.sup = c(1, 19, 20, 21, 23, 28),  # variables quantitatives
                quali.sup = setdiff(1:30, c(1, 19, 20, 21, 23, 28)))  # variables qualitatives
```






```
fviz_screeplot(acp_rslt, addlabels = TRUE, ylim = c(0, 50))
```




```
variance_expliquee <- acp_rslt$eig[, 2]  # variance par composante
variance_cumulee <- acp_rslt$eig[, 3]  # variance cumulée

nb_composantes <- which(variance_cumulee >= 80)[1]
cat("\nNombre de composantes guardées (en choisissant au moins 80% comme seuil) :", nb_composantes, "\n")
```

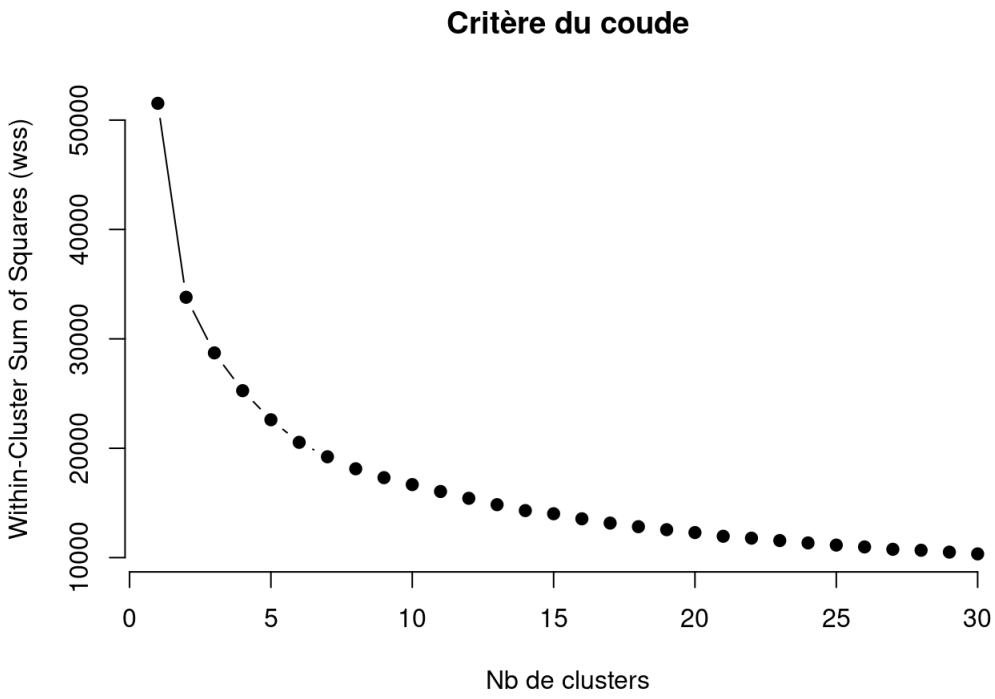
```
##
## Nombre de composantes guardées (en choisissant au moins 80% comme seuil) : 34
```

Il est donc nécessaire de guarder les 34 premières composantes pour avoir 80.1% de la variance totale des données. Cela s'explique car chaque composante a une variance individuelle relativement faible. Avec notamment la première composante qui explique seulement 19,3% de la variance puis ensuite les autres composantes ne font que contribuer de moins en moins à la variance.

Question 2

2. Appliquer l'algorithme des k-means sur les vecteurs projetés, pour un nombre de composantes allant de 1 à 30 et regarder l'évolution de la somme des carrés intra-classes. En utilisant un critère du coude, décider du nombre d'un nombre de classes à retenir. Appliquer l'algorithme pour ce nombre de classes et en déduire une classification des patientes.

```
library(cluster)
library(factoextra)

coord_projetees <- acp_rslt$ind$coord
nb_dimensions <- ncol(coord_projetees)  # nombre de dimensions

wss <- c()  # somme des carrés intra-clusters
for (k in 1:30) {
  kmeans_rslt <- kmeans(coord_projetees[, 1:min(30, nb_dimensions)], centers = k, nstart = 10, iter.max=100)
  wss <- c(wss, kmeans_rslt$tot.withinss)
}

plot(1:30, wss, type = "b", pch = 19, frame = FALSE,
     xlab = "Nb de clusters",
     ylab = "Within-Cluster Sum of Squares (wss)",
     main = "Critère du coude")
```



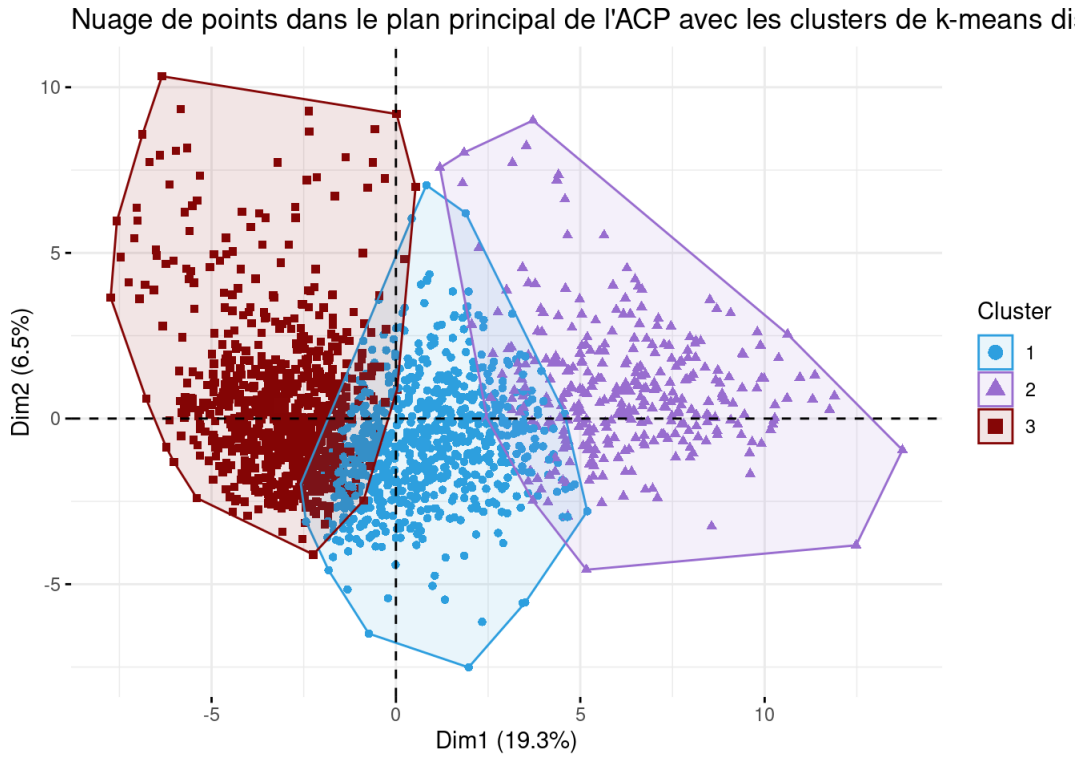

```
nombre_optimal_clusters <- 3  # On choisi 3 car c'est là qu'on repère que le wss ralenti

# On applique kmeans avec 3 comme choix de nombre de cluster
final_kmeans <- kmeans(coord_projetees[, 1:min(30, nb_dimensions)], centers = nombre_optimal_clusters, nstart = 10, iter.max=100)

genes.data$cluster <- as.factor(final_kmeans$cluster)
```

Donc avec l'algortihme Kmeans avec 3 clusters, on peut finalement classer les patientes en 3 groupes.

Question 3

3. Tracer le nuage de points dans le plan principal de l'ACP avec des couleurs correspondant aux classes de k-means. Que constatez-vous?

```
fviz_pca_ind(acp_rslt,
             geom = "point",
             col.ind = genes.data$cluster,
             palette = c("#2E9FDF", "#996DCF", "#850606"),
             addEllipses = TRUE,
             ellipse.type = "convex",
             legend.title = "Cluster",
             repel = TRUE) + ggtitle("Nuage de points dans le plan principal de l'ACP avec les clusters de k-means distingués")
```



On utilise donc les deux premières composantes pour les dimensions. On remarque qu'il y a 3 clusters. Le cluster 1 et 2 se confondent légèrement dans certaines zones ce qui pourrait signifier que ces deux groupes de clusters pourraient avoir bénéfécié d'autres composantes pour mieux séparer ceux-ci. La dimension 1 est plus discriminante que la dimension 2 avec 19.3% de variance expliquée pour la première contre seulement 6.5% pour la deuxième.

Interprétation des résultats

Question 4

4. A l'aide de tests du chi-deux, trouver les variables médicales qualitatives auxquelles les classes trouvées sont les plus liées (ici, on comprendra lié comme éloigné de l'indépendance). Comparer les premières d'entre elles (par exemple les trois premières) avec la classification établie, et utilisant la fonction table qui permet d'établir des tables de contingence. Commenter les résultats en terme d'interprétation de la classification établie.

```
variables_qualitatives <- setdiff(1:30, c(1, 19, 20, 21, 23, 28))

rslt_chi2 <- data.frame(Variable = character(), p_value = numeric())

# On fait le test du Chi2 pour chaque variable qualitative
for (var in variables_qualitatives) {
  table_contingence <- table(genes.data[[var]], genes.data$cluster)

  test_chi2 <- chisq.test(table_contingence)

  rslt_chi2 <- rbind(rslt_chi2,
                     data.frame(Variable = colnames(genes.data)[var],
                                p_value = test_chi2$p.value))
}
```

```
## Warning in chisq.test(table_contingence): L'approximation du Chi-2 est
## peut-être incorrecte
## Warning in chisq.test(table_contingence): L'approximation du Chi-2 est
## peut-être incorrecte
## Warning in chisq.test(table_contingence): L'approximation du Chi-2 est
## peut-être incorrecte
## Warning in chisq.test(table_contingence): L'approximation du Chi-2 est
## peut-être incorrecte
## Warning in chisq.test(table_contingence): L'approximation du Chi-2 est
## peut-être incorrecte
## Warning in chisq.test(table_contingence): L'approximation du Chi-2 est
## peut-être incorrecte
## Warning in chisq.test(table_contingence): L'approximation du Chi-2 est
## peut-être incorrecte
```

```
rslt_chi2 <- rslt_chi2[order(rslt_chi2$p_value), ]
print(rslt_chi2)
```

```
##                         Variable       p_value
## 16           integrative_cluster 5.916586e-302
## 6     pam50_._claudin.low_subtype 6.023344e-287
## 22     X3.gene_classifier_subtype 2.261457e-274
## 9                    er_status 5.308525e-215
## 8      er_status_measured_by_ihc 6.533070e-190
## 20                   pr_status  1.173955e-90
## 10     neoplasm_histologic_grade  4.038172e-89
## 5                 chemotherapy  1.378208e-73
## 7                    cohort  1.514405e-45
## 14              hormone_therapy  8.112072e-42
## 11   her2_status_measured_by_snp6  6.404208e-24
## 13 tumor_other_histologic_subtype  9.455608e-23
## 12                 her2_status  5.620920e-22
## 3          cancer_type_detailed  5.859788e-19
## 18                oncotree_code  5.859788e-19
## 15     inferred_menopausal_state  7.774012e-16
## 24            death_from_cancer  5.355371e-13
## 21                radio_therapy  2.466406e-08
## 4                  cellularity  2.516478e-08
## 23                 tumor_stage  9.372699e-05
## 19             overall_survival  2.034667e-02
## 2                  cancer_type  8.562204e-02
## 1         type_of_breast_surgery  1.360954e-01
## 17       primary_tumor_laterality  6.841762e-01
```

```
# Comparer les trois premières variables qualitatives
for (var in rslt_chi2$Variable[1:3]) {
  cat("\nTable de contingence de la variable :", var, "\n")
  print(table(genes.data[[var]], genes.data$cluster))
}
```

```
##
## Table de contingence de la variable : integrative_cluster
##
##         1   2   3
##   1    86  12  34
##   10    18 201   0
##   2    39   1  32
##   3    74   1 207
##   4ER-  24  36  14
##   4ER+ 103  11 130
##   5   133  33  18
##   6    61   2  21
##   7    57   1 124
##   8    75   1 213
##   9    90  23  29
##
## Table de contingence de la variable : pam50_._claudin.low_subtype
##
##                1   2   3
##   Basal       24 170   5
##   claudin-low  76 100  23
##   Her2        141  40  39
##   LumA        179   0 500
##   LumB        293   9 159
##   NC            2   0   4
##   Normal       45   3  92
##
## Table de contingence de la variable : X3.gene_classifier_subtype
##
##                          1   2   3
##   ER-/HER2-              55 221  14
##   ER+/HER2- High Prolif 362  13 228
##   ER+/HER2- Low Prolif  152   1 466
##   HER2+                 129  40  19
```

Les variables les plus significatives et donc avec la p-value la plus faible sont integrative_cluster, pam50_._claudin.low_subtype et X3.gene_classifier_subtype.

Pour la table de contigence integrative_cluster, on remarque que la modalité 10 est un bon indicateur du cluster 3. Pour le tableau de contingence pam50_._claudin.low_subtype, on remarque que la modalité Basal est très lié au cluster 3 et LumA au cluster 2. Pour la dernière table de contingence X3.gene_classifier_subtype, la modalité ER-/HER2- est ptincipalement associée au cluster 3 et ER+/HER2- Low Prolif au cluster 2.

Prédiction de l'indice de Nottingham

Question 5

5. Mettre en place une prédiction de l'indice de Nottingham à l'aide d'un modèle linéaire. Vous justifierez les choix faits.

```
y <- genes.data$nottingham_prognostic_index

X <- genes.data[, c("cluster", colnames(genes.data)[31:130])]
X$cluster <- as.factor(X$cluster)

set.seed(48)
train_index <- sample(1:nrow(X), 0.8 * nrow(X))  # 80% réservé pour l'entraînement

X_train <- X[train_index, ]
y_train <- y[train_index]
X_test <- X[-train_index, ]
y_test <- y[-train_index]

# Application du modèle linéaire sur les données d'entraînements
model <- lm(y_train ~ ., data = X_train)

summary(model)
```

```
##
## Call:
## lm(formula = y_train ~ ., data = X_train)
```

```
## lm(formula = y_train ~ ., data = x_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4274 -0.6496 -0.0251  0.6217  2.9545
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.0994439  0.0512334  80.015  < 2e-16 ***
## cluster2    -0.0076915  0.1505414  -0.051 0.959259
## cluster3    -0.1410548  0.0939531  -1.501 0.133492
## aurka        0.0076382  0.0596245   0.128 0.898083
## chek1       -0.0376150  0.0562617  -0.669 0.503878
## aph1b       -0.0507105  0.0483125  -1.050 0.294065
## cdk1         0.0586510  0.0629702   0.931 0.351801
## bcl2         0.0124662  0.0476701   0.262 0.793737
## e2f2        -0.0514066  0.0589307  -0.872 0.383179
## ccne1       -0.0543104  0.0541276  -1.003 0.315850
## cdc25a       0.0777278  0.0500897   1.552 0.120940
## mapt        -0.0918140  0.0465576  -1.972 0.048797 *
## igf1r       -0.0962308  0.0419585  -2.293 0.021966 *
## gata3       -0.0152015  0.0639990  -0.238 0.812282
## fancd2       0.0251464  0.0434710   0.578 0.563044
## ccnb1        0.1320912  0.0511464   2.583 0.009905 **
## slc19a1      0.0788973  0.0363985   2.168 0.030355 *
## runx1       -0.0047356  0.0417187  -0.114 0.909641
## ahnak        0.0512304  0.0432193   1.185 0.236073
## dtx3        -0.0598838  0.0451344  -1.327 0.184792
## stat1        0.0048952  0.0418386   0.117 0.906876
## gsk3b        0.1512756  0.0449827   3.363 0.000792 ***
## ncoa3        0.1454845  0.0409342   3.554 0.000392 ***
## tgfbr3      -0.0271268  0.0337503  -0.804 0.421676
## rad51       -0.0097200  0.0409415  -0.237 0.812372
## casp3        0.0041076  0.0356796   0.115 0.908362
## msh6         0.0302790  0.0413913   0.732 0.464576
## e2f7        -0.0353870  0.0423194  -0.836 0.403190
## bmpr1b      -0.0542851  0.0370490  -1.465 0.143081
## rps6kb2      0.0388045  0.0368781   1.052 0.292870
## tgfb3       -0.1122772  0.0469661  -2.391 0.016951 *
## srd5a1      -0.0595796  0.0386346  -1.542 0.123265
## pten         0.0243396  0.0369041   0.660 0.509659
## stat5b      -0.0646596  0.0356626  -1.813 0.070029 .
## map2k4      -0.0542684  0.0379777  -1.429 0.153237
## e2f8         0.0302797  0.0323820   0.935 0.349906
## cyp21a2     -0.0335657  0.0384304  -0.873 0.382585
## tsc1        -0.0722703  0.0407394  -1.774 0.076283 .
## rbl1         0.0128420  0.0360131   0.357 0.721449
## diras3      -0.1006905  0.0313804  -3.209 0.001363 **
## hes6         0.0147504  0.0365015   0.404 0.686197
## chek2       -0.0585307  0.0415286  -1.409 0.158934
## eif4ebp1     0.0253351  0.0338837   0.748 0.454761
## e2f3        -0.0867830  0.0464566  -1.868 0.061962 .
## mapk1       -0.0240386  0.0419715  -0.573 0.566913
## ppp2r2a     -0.0269022  0.0333155  -0.807 0.419515
## rpgr        -0.0414240  0.0379599  -1.091 0.275345
## e2f4         0.0154450  0.0440819   0.350 0.726112
## hsd17b10    -0.0060326  0.0343549  -0.176 0.860636
## cdk2         0.0113326  0.0444769   0.255 0.798916
## nras        -0.0661143  0.0382623  -1.728 0.084220 .
## hdac2        0.0019929  0.0457795   0.044 0.965283
## prkcq       -0.0103826  0.0392498  -0.265 0.791413
## cdk4         0.0142535  0.0393512   0.362 0.717246
## cdkn2a       0.0200241  0.0392229   0.511 0.609766
## nrip1       -0.0134485  0.0369649  -0.364 0.716047
## e2f5        -0.0142586  0.0392683  -0.363 0.716578
## cbfb         0.0406432  0.0483130   0.841 0.400351
## tbx3        -0.0068202  0.0365223  -0.187 0.851890
## cdk8         0.0179106  0.0351257   0.510 0.610201
## hsd17b4      0.0573288  0.0368533   1.556 0.120027
## dlec1        0.0080009  0.0316828   0.253 0.800667
## bard1       -0.0231234  0.0387134  -0.597 0.550405
## rheb         0.1271148  0.0496366   2.561 0.010543 *
## jag1        -0.0188396  0.0383807  -0.491 0.623601
## tp53bp1      0.0904476  0.0364909   2.479 0.013304 *
## heyl        -0.0092561  0.0422082  -0.219 0.826451
## pik3r1       0.0235501  0.0374816   0.628 0.529901
## ncor2       -0.1070048  0.0423990  -2.524 0.011719 *
## src          0.0447521  0.0389136   1.150 0.250322
## ccnd1        0.0106402  0.0406210   0.262 0.793407
```

```
## map2k5      -0.0174560  0.0339544  -0.514 0.607261
## erbb3        0.0003379  0.0475569   0.007 0.994332
## egfr        -0.0267967  0.0389210  -0.688 0.491258
## lama2        0.0971157  0.0499072   1.946 0.051861 .
## map3k1      -0.0388862  0.0411641  -0.945 0.344992
## map2         0.0640965  0.0367515   1.744 0.081366 .
## map2k1      -0.0204687  0.0470493  -0.435 0.663593
## mmp12       -0.1065159  0.0362590  -2.938 0.003361 **
## nek1        -0.0212839  0.0381600  -0.558 0.577099
## lfng        -0.0059437  0.0366778  -0.162 0.871289
## cdk6        -0.0177022  0.0432464  -0.409 0.682356
## smarcc2      0.0742518  0.0461936   1.607 0.108188
## numb         0.0511475  0.0359312   1.423 0.154816
## mmp9        -0.0051857  0.0358398  -0.145 0.884974
## smarcb1      0.0496935  0.0358488   1.386 0.165904
## hsd17b2     -0.0044093  0.0334760  -0.132 0.895227
## bmp4        -0.0151545  0.0322741  -0.470 0.638745
## dll3        -0.0023012  0.0316725  -0.073 0.942091
## hsd17b7      0.0050676  0.0376409   0.135 0.892923
## mmp1        -0.0516069  0.0338170  -1.526 0.127217
## cyb5a        0.0011470  0.0422749   0.027 0.978359
## maml3       -0.0577535  0.0325036  -1.777 0.075810 .
## dnah11       0.0894545  0.0299530   2.986 0.002870 **
## adam17      -0.0007287  0.0359369  -0.020 0.983824
## prkce        0.0421757  0.0322346   1.308 0.190951
## cxcl8       -0.0182313  0.0383097  -0.476 0.634224
## mmp15       -0.0256279  0.0388086  -0.660 0.509126
## hif1a       -0.0561213  0.0385085  -1.457 0.145235
## rbpj         0.0456905  0.0354062   1.290 0.197099
## acvr1b      -0.0678863  0.0391937  -1.732 0.083478 .
## hsd17b8     -0.0053303  0.0371962  -0.143 0.886071
## kdm5a        0.0293854  0.0362135   0.811 0.417244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9945 on 1420 degrees of freedom
## Multiple R-squared:  0.3006, Adjusted R-squared:  0.2503
## F-statistic: 5.983 on 102 and 1420 DF,  p-value: < 2.2e-16
```

```
y_pred <- predict(model, newdata = X_test)

# Calculer le RMSE
rmse <- sqrt(mean((y_test - y_pred)^2))
cat("RMSE sur l'ensemble de test :", rmse, "\n")
```

```
## RMSE sur l'ensemble de test : 0.99825
```

Puisque l'indice de Nottingham est une variable continue, on utilise la régression linéaire. On inclu les clusters de Kmeans car ils regroupent les patientes selon des caractéristiques similaires ce qui peut permettre d'expliquer les variations de l'indice de Nottingham.

Environ 25 % de la variance a été expliquée avec une erreur quadratique moyenne de 0.998. Les clusters de K-means et les gènes sélectionnés ont permis de repérer plusieurs variables importantes comme gsk3b, ncoa3, mmp12, et tp53bp1. Toutefois, on pourrait utiliser des modèles non linéaires par exemple Random Forest pour améliorer le modèle car il explique uniquement 1/4 de la variance totale de l'indice de Nottingham.