

Projet Analyse de Donnée

FERRERE HOAREAU Anthony et CALLIS Guilhem

2025-01-02

R Markdown

Chargement du jeux de donnée: Décrivez l'ensemble du jeu de données en précisant la nature des variables

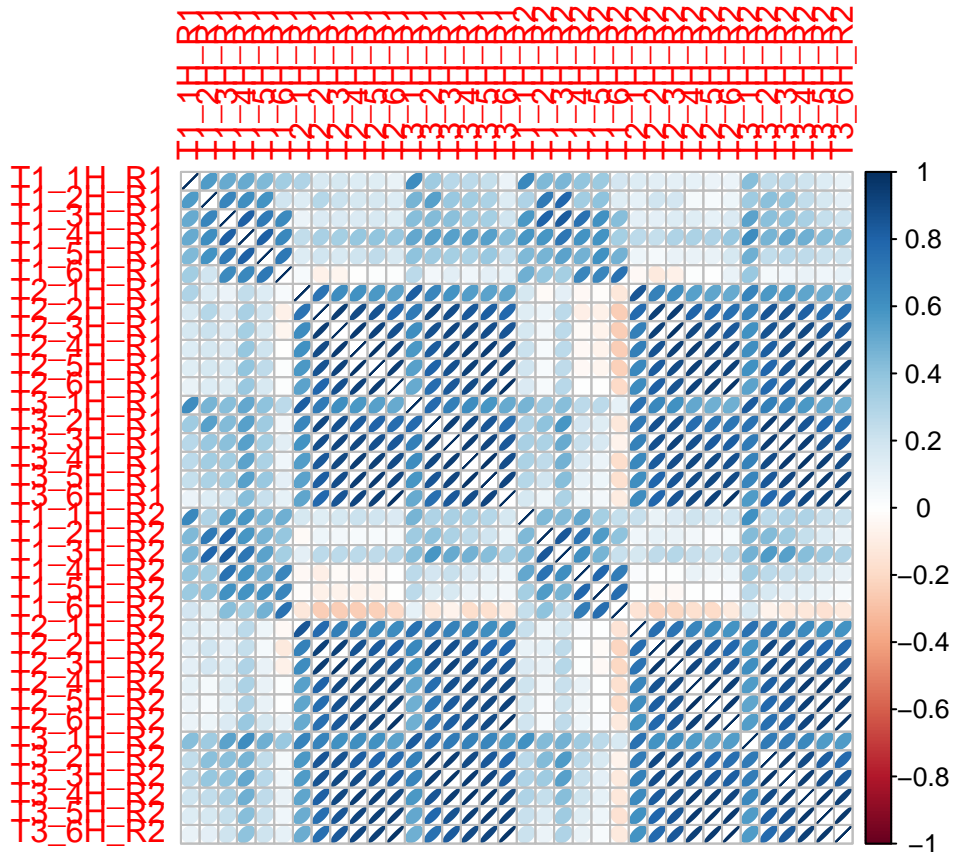
```
## Le chargement a nécessité le package : ggplot2
```

```
## 'data.frame': 542 obs. of 39 variables:
## $ T1_1H_R1: num -0.205 -0.62 0.309 0.192 0.108 ...
## $ T1_2H_R1: num -0.689 -0.856 0.817 0.148 0.288 ...
## $ T1_3H_R1: num -0.1811 -0.0211 -0.5615 0.2424 -0.1975 ...
## $ T1_4H_R1: num -0.06657 -0.14456 0.18148 0.56182 -0.00155 ...
## $ T1_5H_R1: num 0.5217 0.4934 -0.337 0.0453 -0.2274 ...
## $ T1_6H_R1: num 0.448 0.454 -0.373 -0.635 -0.571 ...
## $ T2_1H_R1: num -0.449 -0.572 -0.209 0.526 0.34 ...
## $ T2_2H_R1: num -1.5144 -1.4755 -1.29 1.4315 -0.0468 ...
## $ T2_3H_R1: num -3.815 -3.079 -2.633 1.842 -0.327 ...
## $ T2_4H_R1: num -2.5 -2.22 -2.4 1.83 -0.47 ...
## $ T2_5H_R1: num -2.9144 -2.2659 -2.4397 1.9242 0.0153 ...
## $ T2_6H_R1: num -3.57 -3.36 -2.03 2.19 2.17 ...
## $ T3_1H_R1: num -0.6645 -0.5427 -0.2709 0.4127 -0.0402 ...
## $ T3_2H_R1: num -2.522 -2.281 -1.176 1.688 0.179 ...
## $ T3_3H_R1: num -1.797 -1.597 -3.018 1.812 -0.192 ...
## $ T3_4H_R1: num -2.967 -2.635 -2.953 1.868 -0.553 ...
## $ T3_5H_R1: num -2.99182 -2.42474 -2.96356 2.14249 -0.00553 ...
## $ T3_6H_R1: num -2.84 -2.54 -2.49 2.1 2.09 ...
## $ T1_1H_R2: num -0.25 -0.527 0.303 -0.234 -0.33 ...
## $ T1_2H_R2: num -0.2376 -0.3474 0.5477 -0.2899 0.0044 ...
## $ T1_3H_R2: num -0.741 -0.64 0.589 0.725 0.171 ...
## $ T1_4H_R2: num 0.504 0.274 -0.908 -0.488 -0.53 ...
## $ T1_5H_R2: num 0.355 0.347 -1.428 -0.289 -0.481 ...
## $ T1_6H_R2: num 0.698 0.663 -0.699 -0.649 -0.714 ...
## $ T2_1H_R2: num -0.671 -0.67 0.163 0.272 -0.373 ...
## $ T2_2H_R2: num -2.489 -2.416 -2.307 1.47 -0.109 ...
## $ T2_3H_R2: num -2.4 -2.21 -2.32 1.83 0.13 ...
## $ T2_4H_R2: num -2.552 -2.198 -3.294 1.618 -0.453 ...
## $ T2_5H_R2: num -2.474 -2.238 -2.967 2.191 0.888 ...
## $ T2_6H_R2: num -3.14 -2.47 -3.18 2.19 2.18 ...
## $ T3_1H_R2: num -0.62 -0.842 -0.195 0.17 -0.446 ...
## $ T3_2H_R2: num -2.7064 -2.4478 -2.1068 1.5124 0.0384 ...
## $ T3_3H_R2: num -2.828 -2.552 -2.624 2.051 -0.111 ...
## $ T3_4H_R2: num -2.849 -2.484 -3.024 1.559 -0.222 ...
```

```
## $ T3_5H_R2: num -2.94 -2.46 -2.81 2.32 1.19 ...
## $ T3_6H_R2: num -3.39 -2.97 -2.7 2.16 1.95 ...
## $ ExpT1 : chr "Non" "Non" "Non" "Non" ...
## $ ExpT2 : chr "Sous" "Sous" "Sous" "Sur" ...
## $ ExpT3 : chr "Sous" "Sous" "Sous" "Sur" ...
```

```
## Le chargement a nécessité le package : corrplot
```

```
## corrplot 0.95 loaded
```



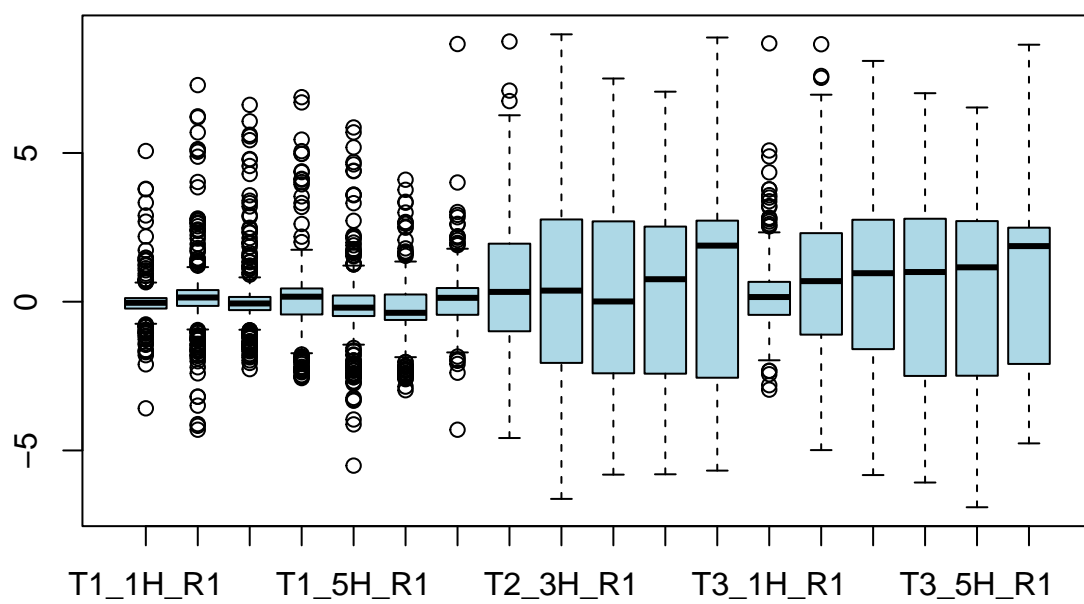
Faites une analyse uni-dimensionnelle et bi-dimensionnelle du jeu de données. Certaines variables sont-elles liées ? Une attention particulière sera portée sur le choix des représentations, et sur l'interprétation des résultats présentés. (Voilà ce que j'ai fait pour l'analyse uni-dimensionnelle. Je vais essayer de tout regrouper dans un seul graphe (je n'y arrive pas pour l'instant mais je vais continuer à essayer))

```
## Le chargement a nécessité le package : grid
```

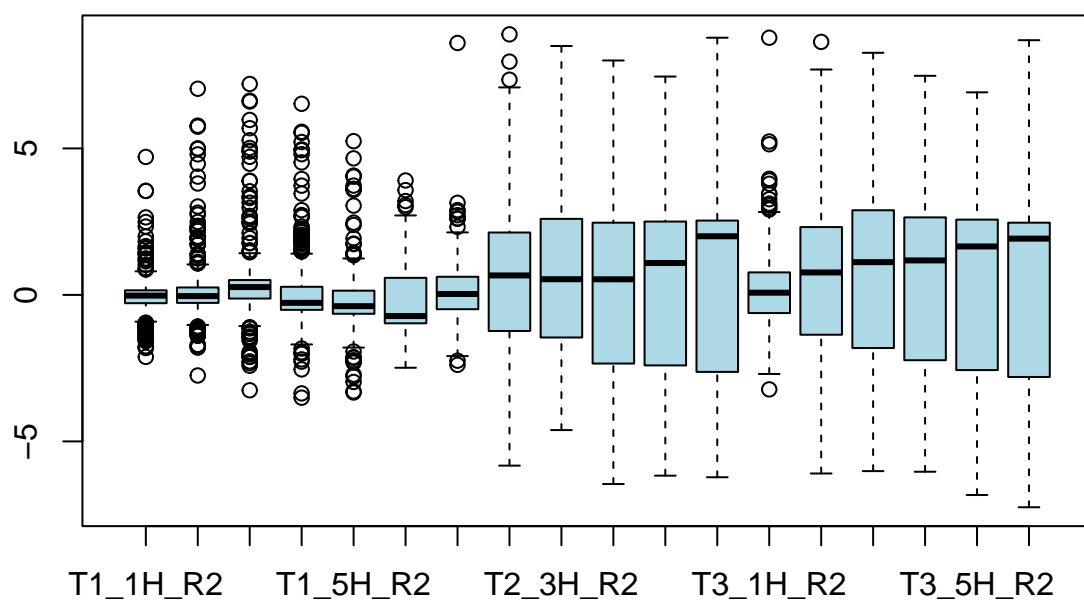
```
## Le chargement a nécessité le package : gridExtra
```

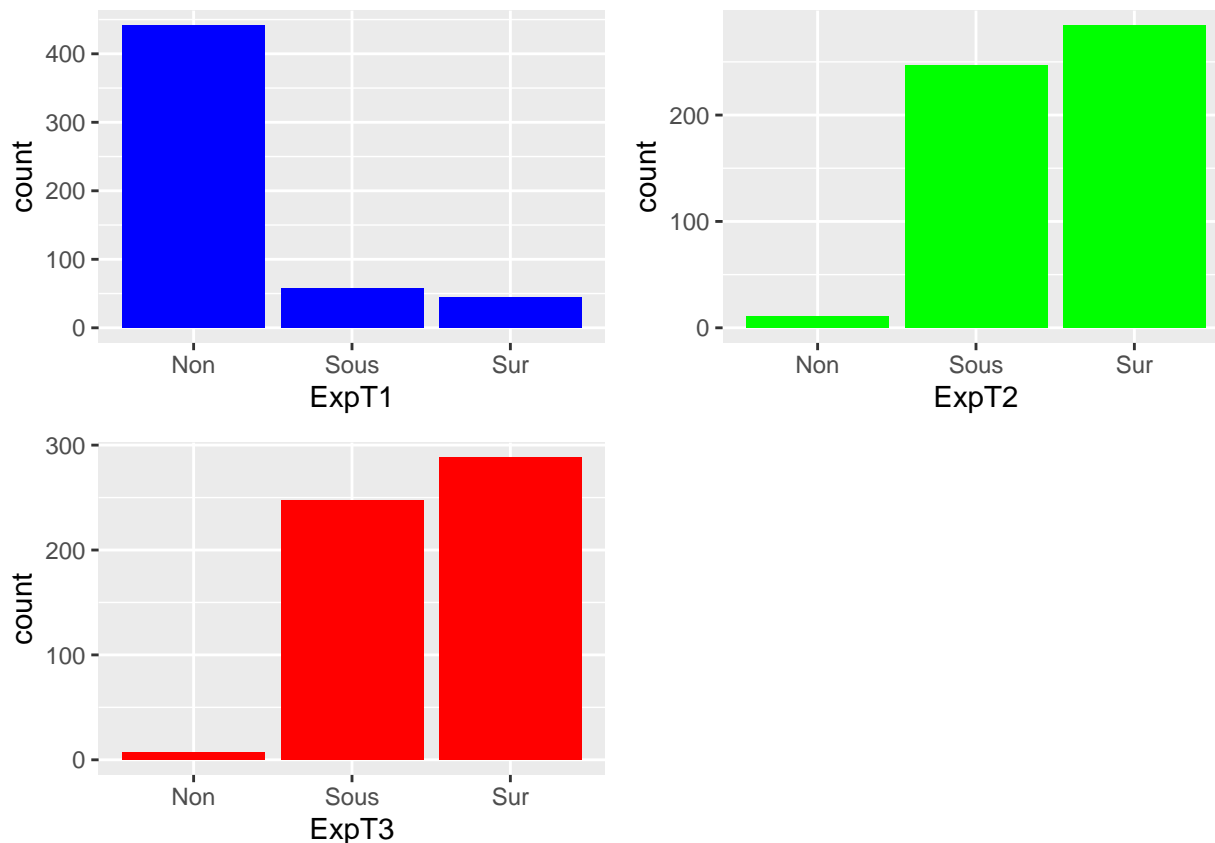
```
## Le chargement a nécessité le package : lattice
```

Expression sous R1



Expression sous R2



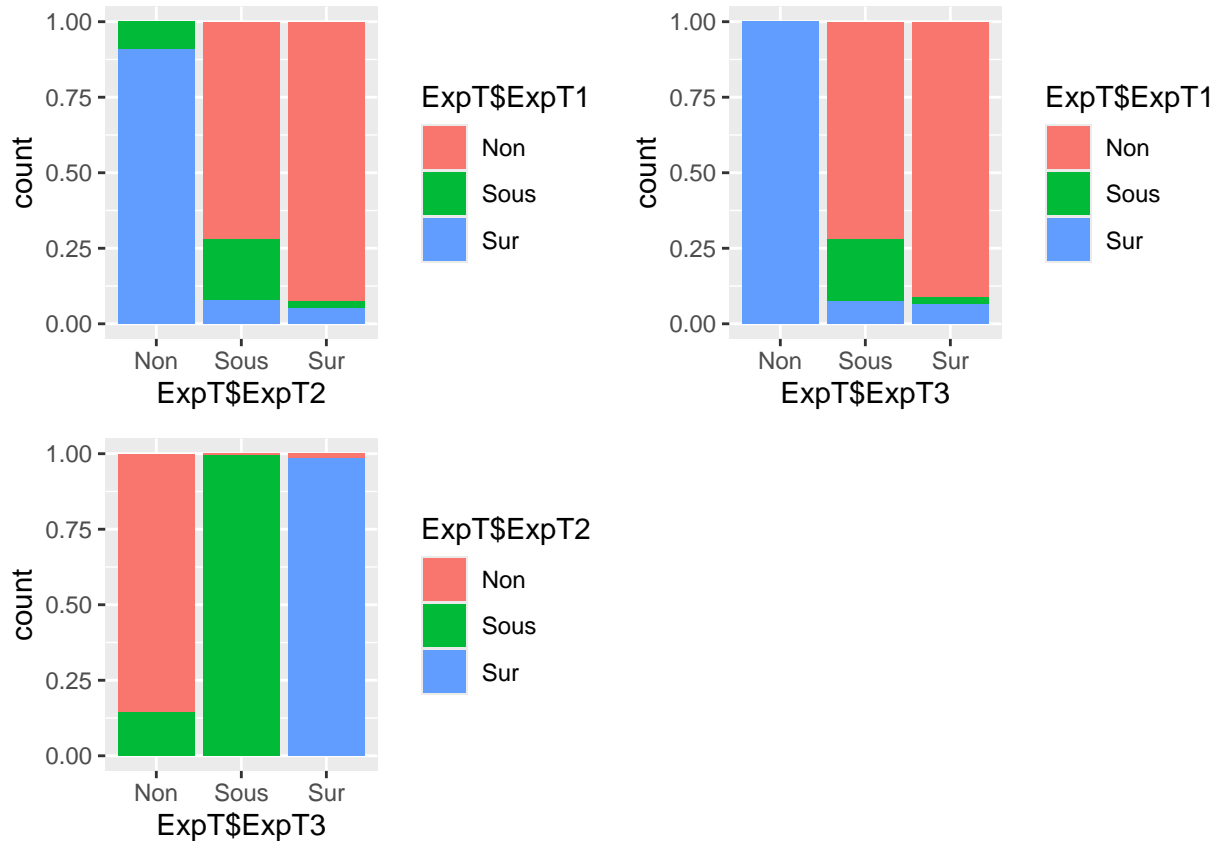


Interprétation des Résultats (analyse uni-dimensionnelle): À la suite de cette analyse, plusieurs observations et conclusions peuvent être tirées sur les relations entre les variables : (on rappelle que T3 est une combinaison de T1 et T2, indiqué dans le sujet.) En ce qui concerne l'analyse Uni-dimensionnelle sur les variables qualitatives, les fréquences des gènes classés comme surexprimés, sous-exprimés et non exprimés sont globalement similaires pour ExpT2 et ExpT3. Cela peut s'expliquer par le fait que T3 est une combinaison de T1 et T2, ce qui entraîne des distributions proches. En revanche, ExpT1 se distingue clairement des deux autres, la majorité des gènes dans ExpT1 sont non exprimés, ce qui contraste avec les répartitions plus équilibrées observées pour ExpT2 et ExpT3. Cette observation suggère que le traitement T1 induit très peu de changements dans l'expression des gènes en réponse à un traitement.

Pour les variables quantitatives, on remarque plusieurs choses. Premièrement, on remarque que les distributions des valeurs d'expression pour R1 et R2 sont remarquablement similaires. Cela indique une bonne reproductibilité biologique entre les réplicats. La cohérence entre R1 et R2 valide la qualité des données et leur fiabilité pour les analyses ultérieures. Pour les traitements, T2 et T3 sont fortement liés, leurs médianes, leurs intervalles interquartiles sont très similaires. Cela renforce l'idée que T3, étant une combinaison de T1 et T2, hérite principalement des caractéristiques de T2. Cependant, les colonnes T2_1H_R1/R2 et T3_1H_R1/R2 présentent un grand nombre de valeurs aberrantes (outliers), ce qui peut indiquer des réponses génétiques atypiques à 1 heure pour ces traitements. Les intervalles interquartiles pour T1 sont beaucoup plus petits, suggérant que les données pour ce traitement sont plus concentrées autour de la médiane. Toutefois, T1 présente également de nombreux outliers, en plus grand nombre que pour T2 ou T3, ce qui peut indiquer des comportements génétiques spécifiques ou une variabilité accrue pour certains gènes sous ce traitement.

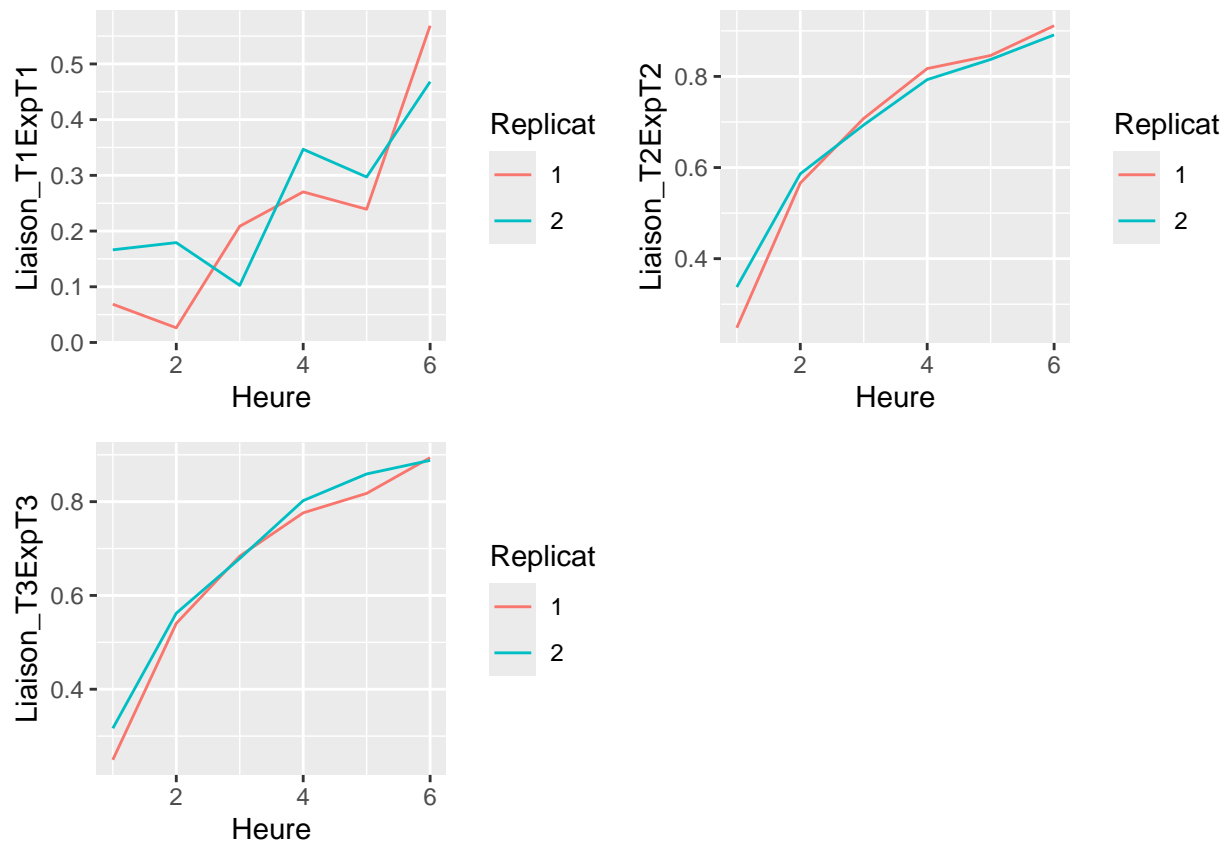
A présent, passons à l'analyse Bi-dimensionnelle. Pour celle-ci j'ai décidé afin de bien comprendre la relation qu'il y pourrait y avoir entre les variables, d'en faire 3.

Une première sur la relation entre l'expression des gènes lors des traitements.



Interprétation des résultats: On constate ici que les relation entre l'expressions des gène sur le 2ème et 3ème traitement sont très similaire vis à vis du premier traitement. En effet, la grande majorité, si ce n'est la totalité des gène non exprimier des traitement 2 et 3 sont sur exprimer pour le traitement 1 et que les gène sur et sous exprimer des traitement 2 et 3 sont majoritairement non exprimer pour le traitement 1. On peut donc on conclure ici que le traitement 1 est quasiment l'opposé du traitement 2 en terme d'expression de gène et que grace au troisième graphique, le traitement 3 étant une combinaison des 2 autres, est quasiment identique au traitement 2 en terme d'expression des gène sur exprimer et sous exprimer bien qu'on note une différence minime pour les gène non exprimer. Ce qui est cohérent avec les observation de l'analyse uni-dimensionnel de l'expression des gènes.

Le chargement a nécessité le package : BioStatR



#Peut être faire la même analyse mais avec T1 et les gène de T2 (je l'ai regarder mais pas mis en graph j'attend ton avis)# Montre le fait que T1 bien que éloigné en terme de fonctionnement et de comportement comparé aux deux autres, à tendance au bout de 4h à se rapprocher de l'expression des gène de T2 à auteur de 40% avant de s'éloigné à nouveau (induisant un comportement différent à long terme mais des réaction à peu près similaire des gène des plante à cours terme.)

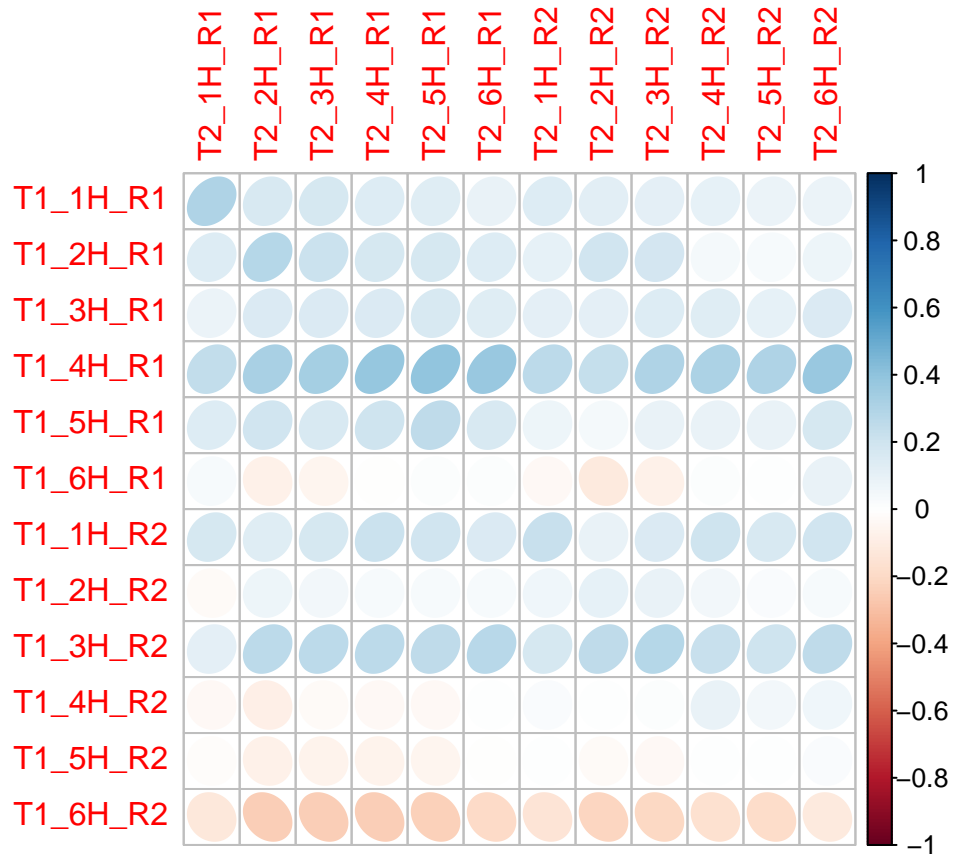
#analyse de de chaque traitement avec leur gènes# Cette analyse-ci nous montre différent graphique sur l'évolution des degré de liaison entre les traitement et l'expression de leur gène au cours du temps. Ce degré de liaison ici reflète la prximité entre l'évolution de l'expression des gènes considérer au fils du temps lors du traitement et l'expression de ces même gène au bout de 6h. On constate pour le premier traitement que malgré une évolution croissante du degré de liaison celui-ci très variant en fonction du temps que cela soit pour le premier ou le second répliat avec des diminution à 2h, 3h et 5h. De plus ce degré à un maximum de 0.56 pour le premier répliat et 0.46 pour le deuxième répliat. Indiquant une forte évolution au cours du temps des gène de la plante lors du traitement ainsi que ce traitement est plus long à agir que les deux autres puisque n'étant semblable qu'à 50% au résultat final attendu pour ce traitement au bout de 6h. Les deux autres traitement quant à eux, ayant de forte similarité comme constaté précédemment, ont une courbe d'évolution du degré de liaison très similaire, on peut donc en conclure que ces deux traitement agisse dans un même temps et bien plus rapidement que le premier traitement avec beaucoup moins de variation des l'expression des gène au cours du temps, puisque les courbe étant des courbe logarithmique, on constate qu'il à de fort changement dans l'expression des gènes au départ mais que ces changement diminue en fonction du temps. On peut donc en déduire que contrairement au premier traitement qui lui est fortement actif sur la plante tout au long de l'expérience malgré une forte présence de gène non actif, et prend beaucoup de temps à agir, les deux autres traitement quant à eux, sont principalement actif sur la plante au début de l'expérience et leur influence et a variation de l'expression des gène diminue avec le temps ils sont donc en comparaison au résultat mesuré à 6h bien plus rapide et efficace que le traitement 1. Ce qui est cohérent avec les observation réaliser jusqu'à présents.

```

corT1T2 = cor(T1, T2)
corT1T3 = cor(T1, T3)
corT2T3 = cor(T2, T3)

g1 = corrplot(corr = corT1T2, method="ellipse")

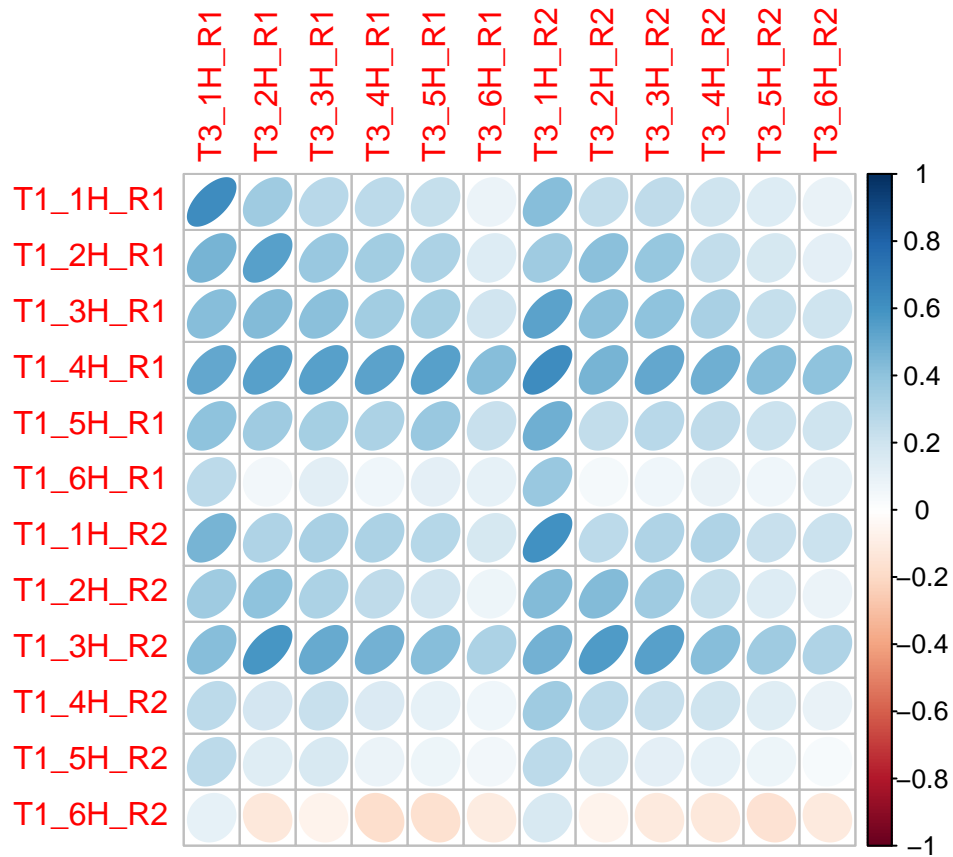
```



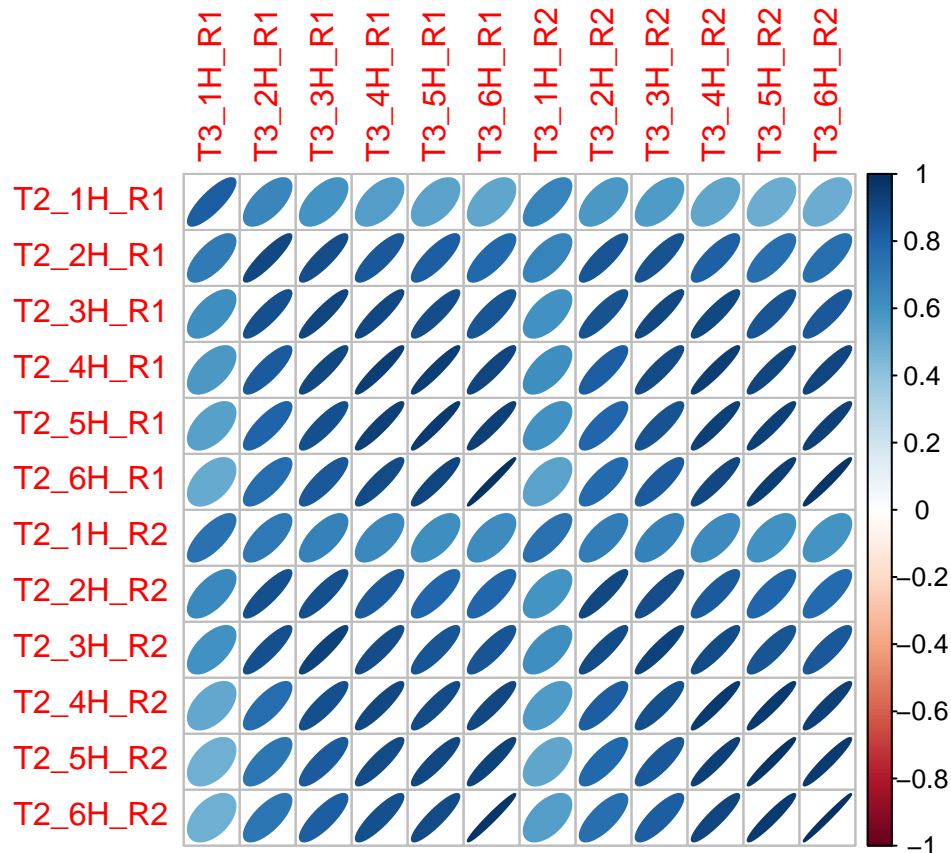
```

g2 = corrplot(corr = corT1T3, method="ellipse")

```

```
g3 = corrplot(corr = corT2T3, method="ellipse")
```



Ici en comparant l'expression des gènes au cours du temps des différents traitements, on remarque que le traitement 1 est très peu corrélé avec le traitement 2 voire même corrélé négativement pour ce qui est de la 6ème heure du réplicat 2, un même constat que l'on peut faire vis-à-vis du traitement 3 (bien que moins visible, puisqu'il s'agit d'une combinaison du traitement 1 et 2), appuyant sur le fait que le traitement 1 se comporte différemment des deux autres. Les traitements T2 et T3 quant à eux sont similaires dans leur variation, en effet ils sont très corrélés positivement quelque soit l'heure à laquelle on a réalisé les mesures, ils se comportent donc de la même manière et sont semblables en termes d'évolution de l'expression des gènes, confirmant le fait que le traitement T3 hérite principalement des caractéristiques du traitement T2.

De manière générale en dehors du premier traitement peu corrélé positivement voire négativement à certaines heures des deux autres, les variables sont fortement corrélées positivement, ce qui implique qu'une forte réduction des dimensions est attendue pour l'ACP.

Menez une analyse en composantes principales où les Tt_sH_Rr sont les individus d'écrits par les gènes.

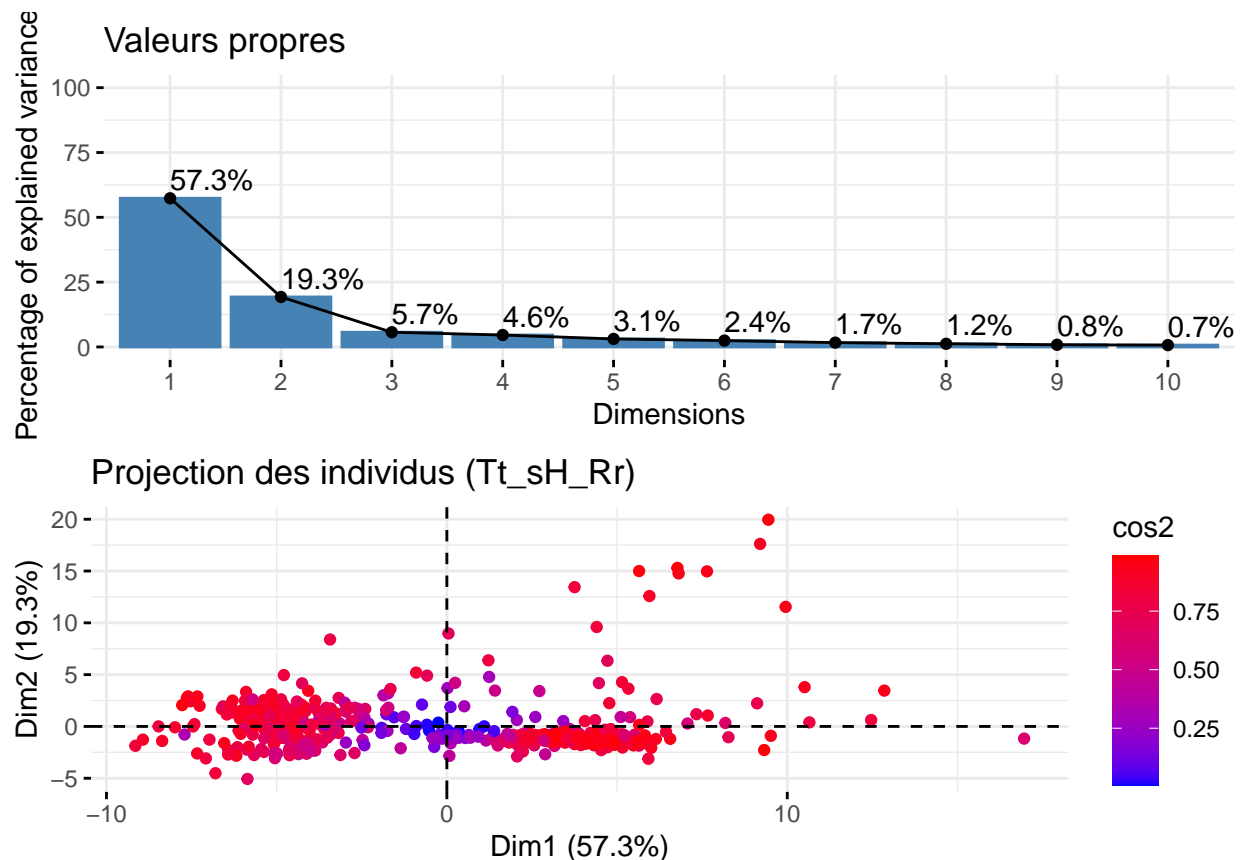
```
## Le chargement a nécessité le package : FactoMineR
```

```
## Le chargement a nécessité le package : factoextra
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3Wba
```

##	eigenvalue	percentage of variance	cumulative percentage of variance
## comp 1	20.643727183	57.34368662	57.34369
## comp 2	6.938281435	19.27300398	76.61669
## comp 3	2.041274447	5.67020680	82.28690
## comp 4	1.656135505	4.60037640	86.88727
## comp 5	1.109348959	3.08152489	89.96880

## comp 6	0.881791877	2.44942188	92.41822
## comp 7	0.599431245	1.66508679	94.08331
## comp 8	0.436409980	1.21224994	95.29556
## comp 9	0.300326182	0.83423939	96.12980
## comp 10	0.254557652	0.70710459	96.83690
## comp 11	0.195646575	0.54346271	97.38036
## comp 12	0.114623243	0.31839790	97.69876
## comp 13	0.113438274	0.31510632	98.01387
## comp 14	0.098934866	0.27481907	98.28869
## comp 15	0.084015374	0.23337604	98.52206
## comp 16	0.075403512	0.20945420	98.73152
## comp 17	0.068914101	0.19142806	98.92295
## comp 18	0.060357899	0.16766083	99.09061
## comp 19	0.056624240	0.15728956	99.24790
## comp 20	0.045538131	0.12649481	99.37439
## comp 21	0.037080890	0.10300247	99.47739
## comp 22	0.032099053	0.08916404	99.56656
## comp 23	0.026356683	0.07321301	99.63977
## comp 24	0.021809019	0.06058061	99.70035
## comp 25	0.015239265	0.04233129	99.74268
## comp 26	0.014299054	0.03971959	99.78240
## comp 27	0.011721521	0.03255978	99.81496
## comp 28	0.010954844	0.03043012	99.84539
## comp 29	0.009729805	0.02702724	99.87242
## comp 30	0.009492364	0.02636768	99.89879
## comp 31	0.008089909	0.02247197	99.92126
## comp 32	0.006904142	0.01917817	99.94044
## comp 33	0.006571646	0.01825457	99.95869
## comp 34	0.005669160	0.01574767	99.97444
## comp 35	0.004995223	0.01387562	99.98831
## comp 36	0.004206745	0.01168540	100.00000



Interprétation des Résultats :

En premier lieu, lorsqu'on regarde notre graphe des valeurs propres, on remarque bien que les 2 premières dimensions représentent environ 80% de la variance, ces deux dimensions sont alors suffisantes pour résumer les données. Il ne faut pas oublier pour la suite que la dimension 1 (57,3%) a beaucoup plus "d'information" que la dimension 2 (19,3%). Chaque point sur notre ACP correspond à une colonne Tt_sH_Rr et donc aux variables.

Nous avons pris la décision de ne pas analyser le graphe des variables car trop de gènes pour l'analyser et tirer des conclusions correctes. Il est alors compliqué d'étudier la contribution des gènes. On ne peut pas résumer le nuage de points à l'aide de deux méta-variables dans ce cas précis.

Cependant, il reste possible d'étudier la contribution relative des individus. Les individus les plus "intéressants" sont ceux qui sont les plus éloignés de l'individu moyen et ceux dont la contribution à la dispersion est la plus importante. Ici, on remarque que la variable la plus intéressante est T1_6H_R2. Elle est éloignée du centre, indiquant une réponse atypique ou spécifique par rapport aux autres combinaisons traitement-heure-réplikat et sa position dans l'espace principal suggère qu'elle joue un rôle important dans la variation observée dans les données.

Faites une classification non supervisée (clustering) de ces données afin de regrouper les Tt sH Rr. en plusieurs classes homogènes.

```
if (!require("forcats")) install.packages("forcats")
if (!require("ggplot2")) install.packages("ggplot2")
if (!require("corrplot")) install.packages("corrplot")
if (!require("FactoMineR")) install.packages("FactoMineR")
if (!require("factoextra")) install.packages("factoextra")
if (!require("mclust")) install.packages("mclust")
if (!require("cluster")) install.packages("cluster")
```

```

if (!require("ppclust")) install.packages("ppclust")
if (!require("circlize")) install.packages("circlize")
if (!require("ggalluvial")) install.packages("ggalluvial")
library(forcats)
library(ggplot2)
library(corrplot)
library(reshape2)

library(FactoMineR)
library(factoextra)

library(mclust)
library(cluster)
library(ppclust)

library(circlize)
library(ggalluvial)

# Maintenant que l'ACP a été effectuée, on fait un clustering des classes à l'aide de la méthode K-means.

# Avant de débiter le clustering avec la méthode K-means, il faut déterminer le nombre de classes.

Kmax<-20
reskmeanscl<-matrix(0,nrow=nrow(DataBio),ncol=Kmax-1)
lintra<-NULL
for (k in 1:Kmax){
  resaux<-kmeans(DataBioCR,centers=k)
  reskmeanscl<-resaux$cluster #pourquoi le [,k-1] ?
  lintra<-c(lintra,resaux$tot.withinss) # tot.withinss correspond à la somme des composantes au carré d
}

df<-data.frame(K=1:20,lintra=lintra)
g1 = ggplot(df,aes(x=K,y=lintra))+
  geom_line()+
  geom_point()+
  xlab("Nombre de classes")+
  ylab("Inertie intraclasses")

# Avec cette méthode, on dirait que le coude correspond lorsque le nombre de classes est de 2.
# On va alors utiliser 2 classes pour la méthode des K-means.

reskmeans<-kmeans(DataBioCR,centers = 2)
g2 = fviz_cluster(reskmeans,data=Tt_sH_Rr,
  ellipse.type="norm",labelsize=8,
  geom=c("point"))+ggtitle("")
#fviz_pca_ind(resacp,col.ind=as.factor(reskmeans$cluster),geom = c("point"),axes=c(1,2))

# A présent, on va essayer une autre méthode, la méthode hiérarchique.

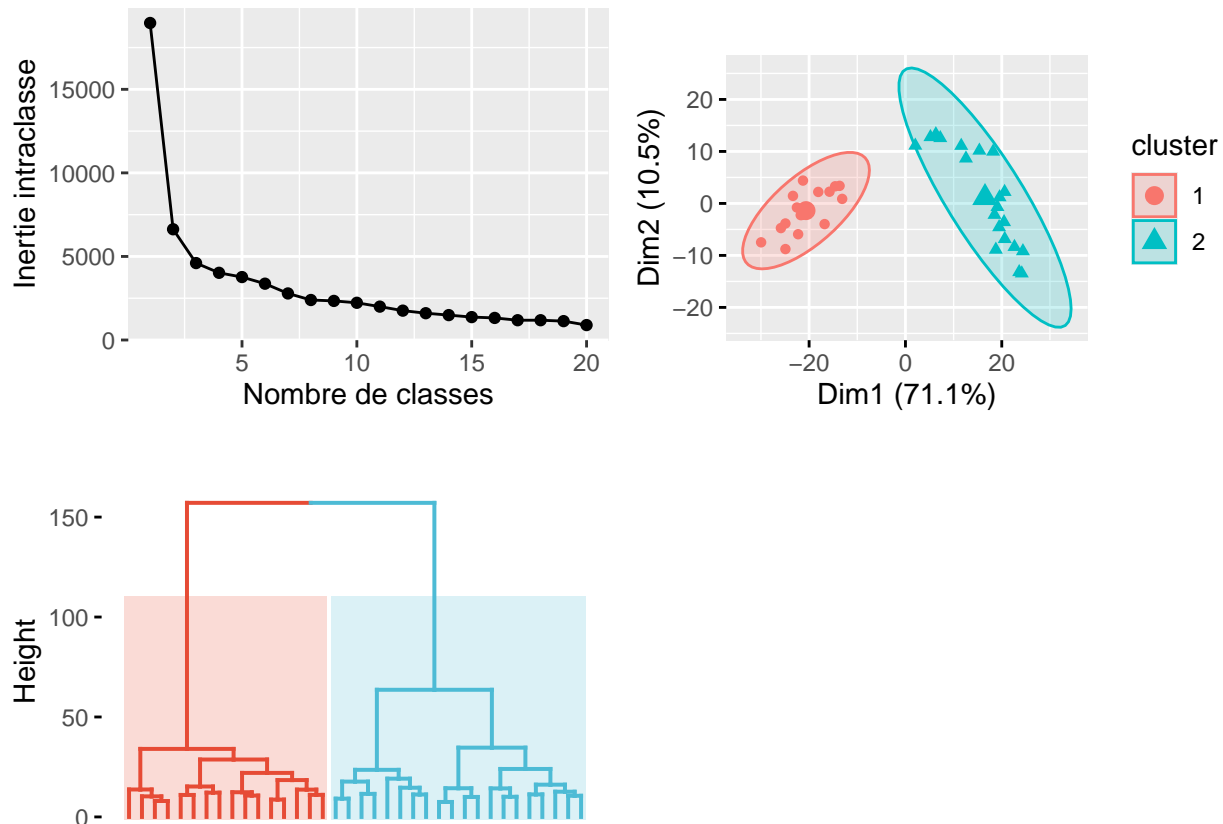
# D'une part, on fait le calcul de la matrice de distances
dist_matrix <- dist(DataBioCR, method = "euclidean")
# Clustering hiérarchique avec la méthode de liaison "ward.D2", on peut aussi faire avec "single", "com
hc <- hclust(dist_matrix, method = "ward.D2")

```

```
# Afficher le dendrogramme
g3 = fviz_dend(hc,k=2,show_labels = FALSE,
rect = TRUE, rect_fill = TRUE,palette = "npg",
rect_border = "npg",
labels_track_height = 0.8)+ggtitle("")

# (Le temps de chargement est plutôt long, C'est NORMAL)

grid.arrange(g1, g2, g3, nrow=2)
```



Interprétation du Clustering : En premier lieu, on remarque que nos 2 Clustering (avec méthode K-means et méthode hiérarchique) nous donnent les mêmes résultats, on obtient les mêmes groupes avec le même nombre d'individus pour les 2 clustering. Pour le Clustering avec la méthode des K-means, on remarque que les clusters sont bien séparés, on peut en déduire que les groupes d'individus sont nettement différents.

Préliminairement, construisez un jeu de données DataExpMoy contenant la moyenne des expressions sur les réplicats de chaque gène, pour chaque traitement et chaque heure. DataExpMoy est donc une matrice de taille $G \times 18$. Vous pourrez utiliser les variables ExpT1, ExpT2 et ExpT3 pour commenter vos résultats des questions suivantes.

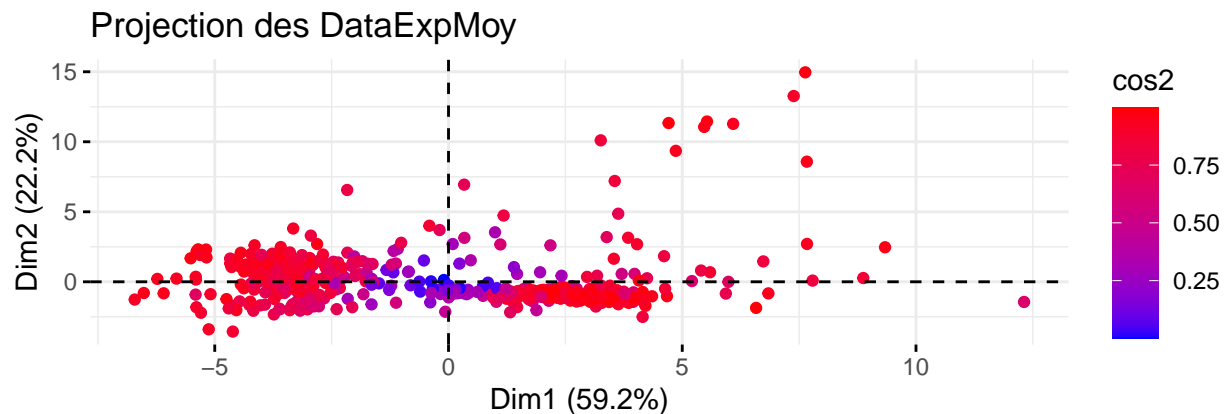
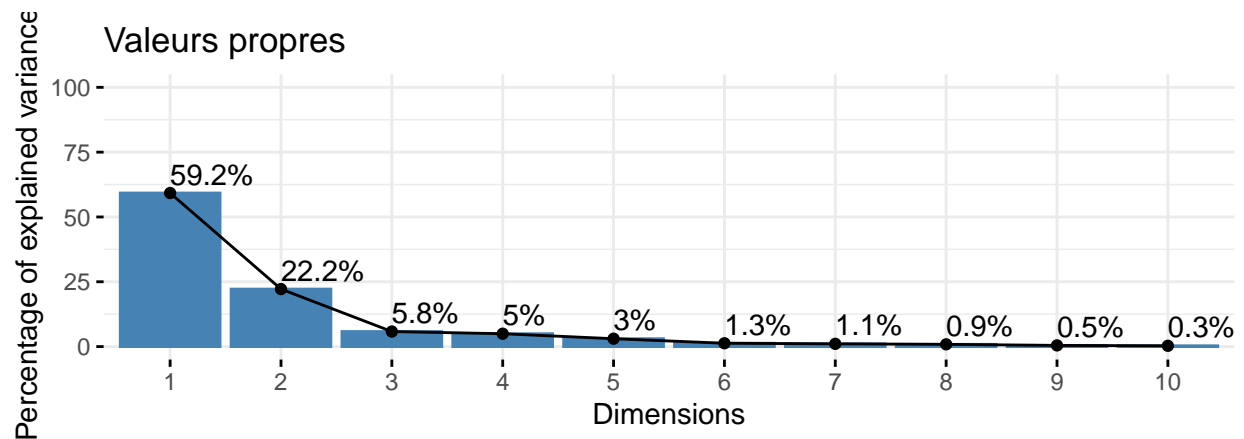
Menez une analyse en composantes principales pour les gènes à partir du jeu de données DataExpMoy.

```
## [1] 542 18
```

```
## Warning in mean.default(DataExpMoy_full[DataExpMoy_full$ExpT1 == "Sur", :
## l'argument n'est ni numérique, ni logique : renvoi de NA
```

```
## [1] NA
```

##		eigenvalue	percentage of variance	cumulative percentage of variance
##	comp 1	10.661647702	59.23137612	59.23138
##	comp 2	3.992763404	22.18201891	81.41340
##	comp 3	1.044273066	5.80151703	87.21491
##	comp 4	0.892464409	4.95813561	92.17305
##	comp 5	0.546150817	3.03417120	95.20722
##	comp 6	0.231196720	1.28442622	96.49165
##	comp 7	0.190280147	1.05711193	97.54876
##	comp 8	0.156124760	0.86735978	98.41612
##	comp 9	0.082022025	0.45567791	98.87179
##	comp 10	0.053967172	0.29981762	99.17161
##	comp 11	0.051951876	0.28862154	99.46023
##	comp 12	0.044218807	0.24566004	99.70589
##	comp 13	0.020421154	0.11345086	99.81934
##	comp 14	0.011863903	0.06591057	99.88526
##	comp 15	0.007317432	0.04065240	99.92591
##	comp 16	0.005444702	0.03024834	99.95616
##	comp 17	0.004849689	0.02694272	99.98310
##	comp 18	0.003042215	0.01690120	100.00000



Interprétation des Résultats :

En premier lieu, lorsqu'on regarde notre graphe des valeurs propres, on remarque bien que les 2 premières dimensions représentent environ 81-82% de la variance, ces deux dimensions sont alors suffisantes pour résumer les données.

Chaque point sur notre ACP correspond à un gène.

Dans ce cas, il est très pertinent d'analyser le graphe des variables car le graphe semble plus lisible et semble apporter des informations pertinentes. On remarque quelque chose de très intéressant notamment à l'aide de la deuxième meta-variable. D'après le graphe, on a l'impression que cet axe mesure l'appartenance d'un gène à un traitement. On remarque que les valeurs positives correspondent au traitement 1, les valeurs négatives au traitement 2 et les valeurs positives proches ou égal à 0 sont les gènes du traitement 3. (pas terminé)

Faites une classification non supervisée (clustering) des gènes à partir de leur expression (DataExpMoy) afin d'obtenir des classes de gènes homogènes (ayant la même évolution d'expression).

```
if (!require("forcats")) install.packages("forcats")
if (!require("ggplot2")) install.packages("ggplot2")
if (!require("corrplot")) install.packages("corrplot")
if (!require("FactoMineR")) install.packages("FactoMineR")
if (!require("factoextra")) install.packages("factoextra")
if (!require("mclust")) install.packages("mclust")
if (!require("cluster")) install.packages("cluster")
if (!require("ppclust")) install.packages("ppclust")
if (!require("circlize")) install.packages("circlize")
if (!require("ggalluvial")) install.packages("ggalluvial")
library(forcats)
library(ggplot2)
library(corrplot)
library(reshape2)

library(FactoMineR)
library(factoextra)

library(mclust)
library(cluster)
library(ppclust)

library(circlize)
library(ggalluvial)

# Maintenant que l'ACP a été effectuée, on fait un clustering des classes à l'aide de la méthode K-mean.

# Avant de débiter le clustering avec la méthode K-means, il faut déterminer le nombre de classes.

Kmax<-15
reskmeanscl<-matrix(0,nrow=nrow(DataExpMoyCR),ncol=Kmax-1)
Iintra<-NULL
for (k in 1:Kmax){
  resaux<-kmeans(DataExpMoyCR,centers=k)
  reskmeanscl[,k-1]<-resaux$cluster
  Iintra<-c(Iintra,resaux$tot.withinss) # tot.withinss correspond à la somme des composantes au carré d
}

df<-data.frame(K=1:15,Iintra=Iintra)
g1 = ggplot(df,aes(x=K,y=Iintra))+
  geom_line()+
  geom_point()+
  xlab("Nombre de classes")+
  ylab("Inertie intraclasses")
```



```

# Avec cette méthode, on dirait que le coude correspond lorsque le nombre de classes est de 2.
# On va alors utiliser 2 classes pour la méthode des K-means.

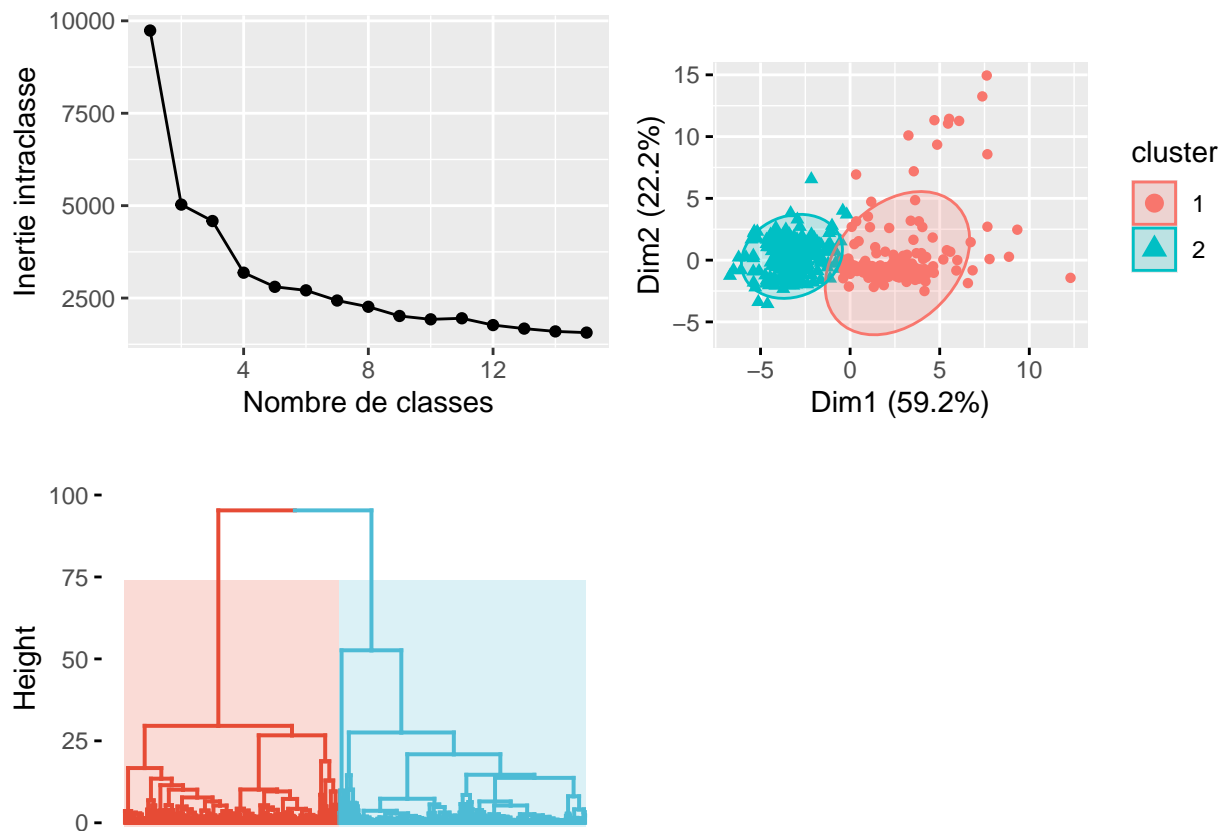
ExpMoykmeans<-kmeans(DataExpMoyCR,centers = 2)
g2 = fviz_cluster(ExpMoykmeans,data=DataExpMoyCR,ellipse.type="norm",labelsize=8,geom=c("point"))+ggtitle("K-means")
#fviz_pca_ind(resacp,col.ind=as.factor(ExpMoykmean$cluster),geom = c("point"),axes=c(1,2))

# A présent, on va essayer une autre méthode, la méthode hiérarchique.

# D'une part, on fait le calcul de la matrice de distances
dist_matrix_ExpMoy <- dist(DataExpMoyCR, method = "euclidean")
# Clustering hiérarchique avec la méthode de liaison "ward.D2", on peut aussi faire avec "single", "complete"
hc_ExpMoy <- hclust(dist_matrix_ExpMoy, method = "ward.D2")
# Afficher le dendrogramme
g3 = fviz_dend(hc_ExpMoy,k=2,show_labels = FALSE,
rect = TRUE, rect_fill = TRUE,palette = "npg",
rect_border = "npg",
labels_track_height = 0.8)+ggtitle("Dendrogramme")
# (Le temps de chargement est plutôt long)

grid.arrange(g1, g2, g3, nrow=2)

```



Faites une classification non supervisée (clustering) des gènes à partir des variables ExpT1, ExpT2 et ExpT3. Comparez avec les résultats de la question précédente.