TP Analyse de la variance (ANOVA) à un facteur

Mathilde Boissel

19/10/2020

Table of Contents

Étude de cas 1 : Étude du rendement en jus de 3 variétés de pommes

Pour chaque variété, 4 arbres sont échantillonnés.
On se pose la question suivante : **Existe-t-il une différence significative entre ces 3 variétés quant à la moyenne des rendements ?**

Lire les données

On peut construire le tableau de données (`data.frame`) à la main.

- Pour ce faire, recopier les commandes suivantes dans R.

```
## grouped data
pommes_by_group <- data.frame(
  Golden = c(48,46,52,50),
  Delicious = c(52,50,49,49),
  Jonagold = c(53,51,55,57)
)
## tidy data
pommes <- data.frame(
  rendement = c(48,46,52,50,52,50,49,49,53,51,55,57),
  variete = factor(rep(c("Golden","Delicious","Jonagold"), rep(4,3)))
)
```

La construction de ce `data.frame` peut passer par bien d'autres procédures.
Par exemple il existe la commande `gl`, pour la construction de facteur, qui peut être utilisé comme suit :

```
variete_facteur <- gl(n = 3, k = 4, label = c("Golden","Delicious","Jonagold"
))
```

Il y a 3 modalités pour le facteur "variete", 4 répétitions et on donne des noms aux modalités du facteur (pour plus d'informations, voir `help("gl")`).

- Noter la différence de présentation des données.

Visualisation Par Groupe

Golden	Delicious	Jonagold
48	52	53
46	50	51
52	49	55
50	49	57

Visualisation Tidy (Rangée)

rendement	variete
48	Golden
46	Golden
52	Golden
50	Golden
52	Delicious
50	Delicious
49	Delicious
49	Delicious
53	Jonagold
51	Jonagold
55	Jonagold
57	Jonagold

Les données sont souvent collectées par groupe, alors que pour les traiter dans R nous aurons besoin d'une seule observation par ligne. Il faut alors ranger les données avec chaque variable (ici "rendement" et "variete") en colonne et une ligne par observation. Pour plus d'informations sur le format "tidy", parcourir le chapitre suivant : tidyverse/12-tidyr

Pour la suite nous utiliserons le jeu de données nommé "pommes", au format **tidy**.

- S'assurer que la variable à expliquer ("rendement") est numérique et que la variable explicative "variete" est bien un facteur.

```
str(pommes)

is.data.frame(pommes)

## [1] TRUE

is.numeric(pommes$rendement)

## [1] TRUE

is.factor(pommes$variete)
```

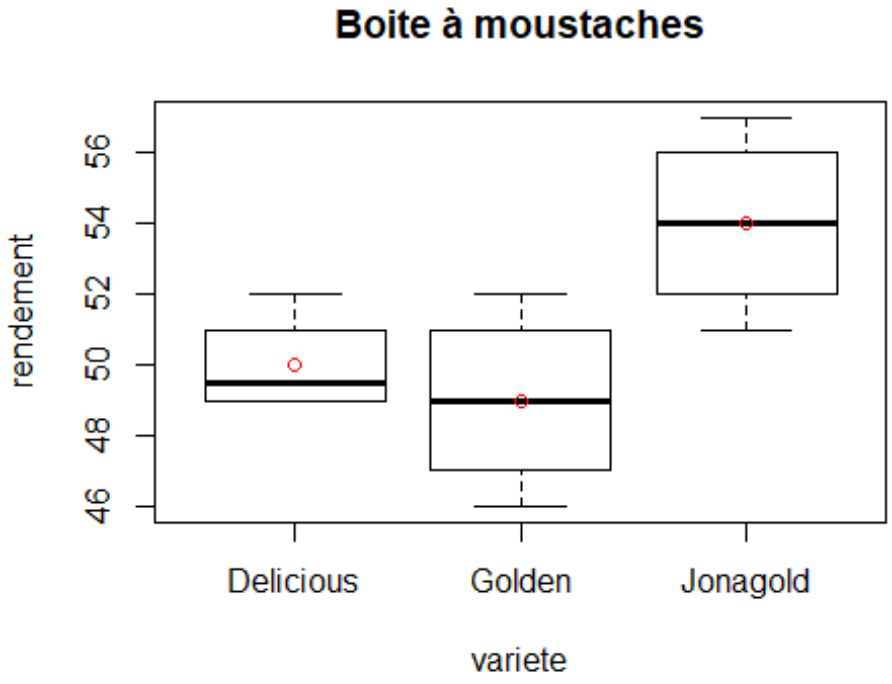
```
## [1] TRUE

class(pommes)

## [1] "data.frame"

class(pommes$rendement)

## [1] "numeric"

class(pommes$variete)

## [1] "factor"
```

`str` procède à l'affichage compact de la structure interne d'un objet R.
Les fonctions `is.[type]` testent le `[type]` d'un objet R et retourne une valeur booléenne.
La fonction `class` retourne la classe (i.e. le type) d'un objet R.

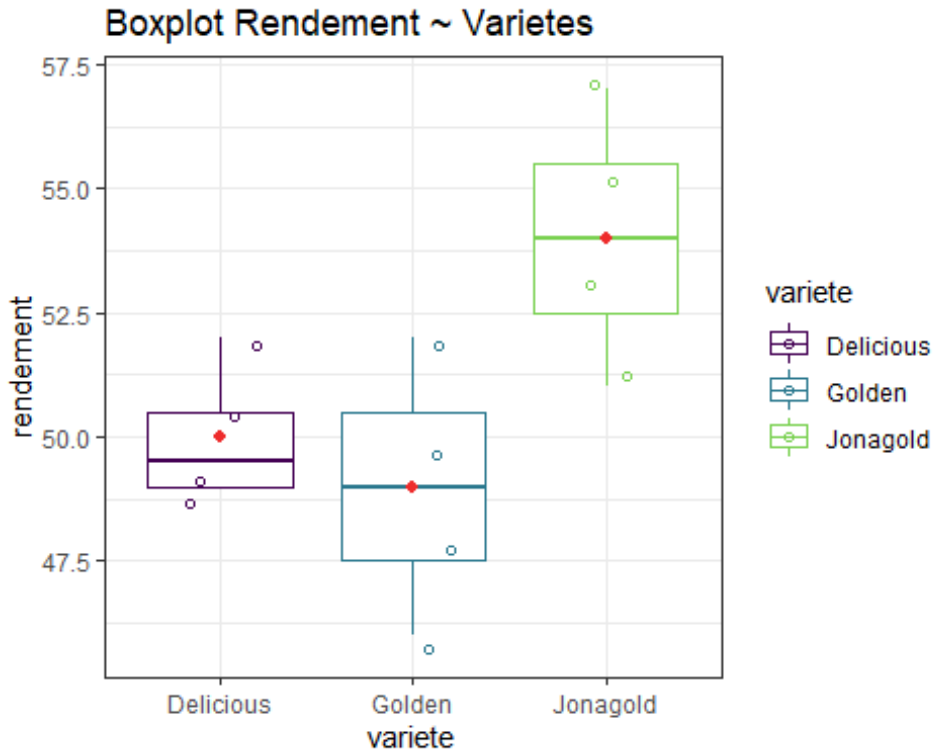
Visualiser et résumer les données

- Visualiser les données avec le graphique adéquat. `?boxplot`

```
## base / stats
means <- aggregate(formula = rendement ~ variete, data = pommes, FUN = mean)
boxplot(formula = rendement ~ variete, data = pommes, main = "Boite à moustac
hes")
points(1:3, means$rendement, col = "red")
```



```r
## Puisque les colonnes de l'objet pommes possèdent la bonne classe la fonction plot s'adapte et retourne également le graphique boxplot.
# plot(formula = rendement ~ variete, data = pommes)

## ggplot2
ggplot(
  data = pommes,
  mapping = aes(x = variete, y = rendement, color = variete)
) +
  geom_boxplot() +
  geom_jitter(shape = 21, position = position_jitter(0.2)) +
  stat_summary(fun = mean, geom = "point", shape = 20, color = "firebrick2", fill = "firebrick2", size = 3) +
  scale_color_viridis_d(end = 0.8) +
  theme_bw() +
  labs(title = "Boxplot Rendement ~ Varietes")
```



On affiche les boxplots des observations correspondant à chaque niveau (modalité) du facteur “variete”. Si les boxplots sont décalés, on peut soupçonner un effet du facteur.

- Tester les commandes suivantes pour avoir un résumé numérique des données.

```r
summary(pommes$rendement)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   46.00   49.00   50.50   51.00   52.25   57.00
```

```r
summary(pommes$variete) # summary.factor(pommes$variete)

## Delicious    Golden  Jonagold
##         4         4         4

table(pommes$variete)

##
## Delicious    Golden  Jonagold
##         4         4         4
```

summary est une fonction générique qui va invoquer la méthode `summary.[type]` adapté à l'object R d'un certain `[type]`. Ici pour "variete" qui est un facteur, la fonction `summary(pommes$variete)` applique en fait `summary.factor(pommes$variete)` sans qu'on ait besoin de faire la nuance nous-même.
Si on sait à l'avance que l'on souhaite compter des effectifs, on peut aussi directement choisir la fonction `table`.
N.B. : `table` peut aussi faire des tableaux de contingences si nous regardons plusieurs facteurs.

- Ces commandes indique-t-elle s'il y a des valeurs manquantes ?

summary précisera le nombre d'observations manquantes dans la colonne `NA's`.
`table` n'affiche le nombre d'observations manquantes que si on précise l'option `useNA` avec la valeur "ifany" ou "always".

- Résumer les données pour chaque groupe :
  effectif, moyenne, ecart-type, données manquantes.
  `?aggregate ?tapply`

```r
## base / stat
n_group <- aggregate(formula = rendement ~ variete, data = pommes, FUN = leng
th)
mean_group <- aggregate(formula = rendement ~ variete, data = pommes, FUN = m
ean)
sd_group <- aggregate(formula = rendement ~ variete, data = pommes, FUN = sd)
na_group <- aggregate(formula = rendement ~ variete, data = pommes, FUN = fun
ction(x) sum(is.na(x)))

## un exemple avec tapply
mean_tapply <- tapply(X = pommes$rendement, INDEX = pommes$variete, FUN = mea
n)

## dplyr (from tidyverse)
pommes %>%
  dplyr::group_by(variete) %>%
  dplyr::summarise(
    n = n(),
    mean = mean(rendement), # /!\ mean(, na.rm = TRUE)
    sd = sd(rendement), # /!\ sd(, na.rm = TRUE)
```

```
    NA_rendement = sum(is.na(rendement))
  ) %>%
  dplyr::ungroup()
```

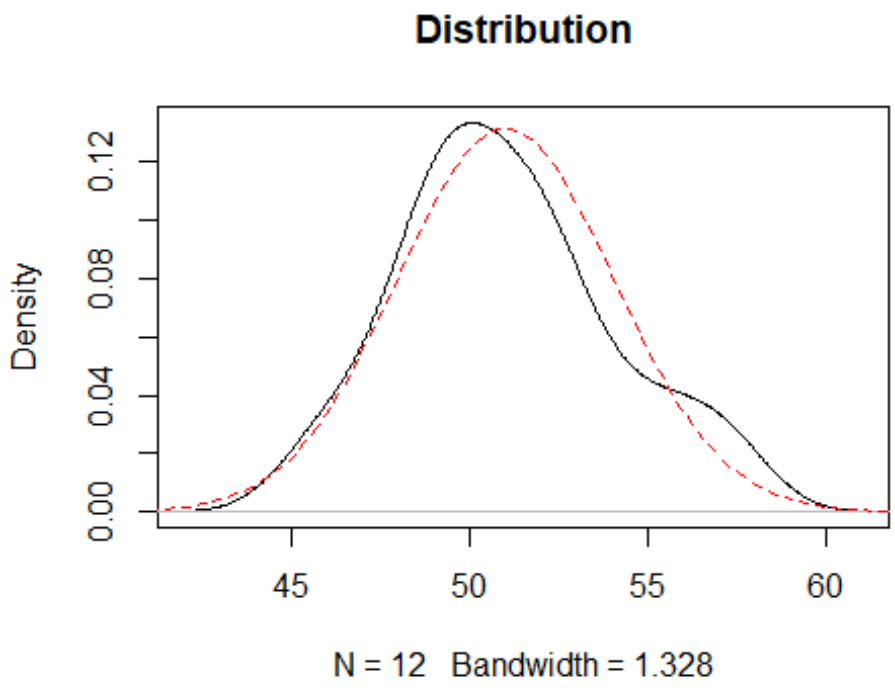
```
## # A tibble: 3 x 5
##   variete       n  mean    sd NA_rendement
##   <fct>     <int> <dbl> <dbl>        <int>
## 1 Delicious     4    50  1.41            0
## 2 Golden        4    49  2.58            0
## 3 Jonagold      4    54  2.58            0
```

```
## et encore beaucoup d'autres facon de faire
# pommes %>%
#   group_by(variete)
#   rstatix::get_summary_stats(rendement, type = "mean_sd")
```

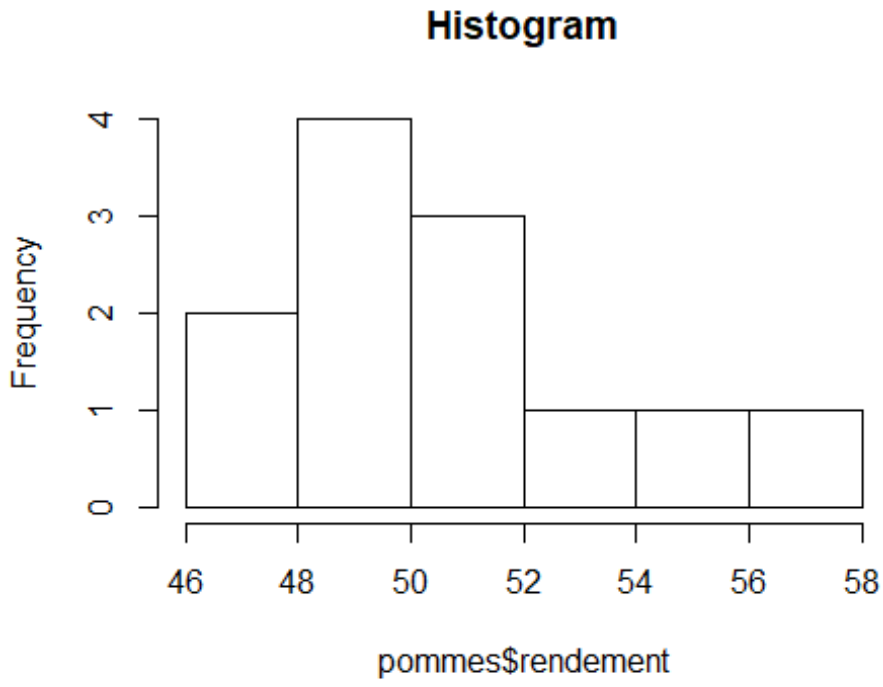
Il faut garder en tête les options par défaut de ces fonctions de base lorsqu'on les applique. Une bonne pratique pourrait être de les rendre explicite à chaque utilisation. Par exemple, faites attention à l'option `na.rm = FALSE` qui indique que les valeurs manquantes ne seront pas supprimées par défaut. Ici le calcul de la moyenne échouerait s'il y avait des valeurs manquantes.

- Visuellement, vérifier la normalité de la variable d'intérêt.

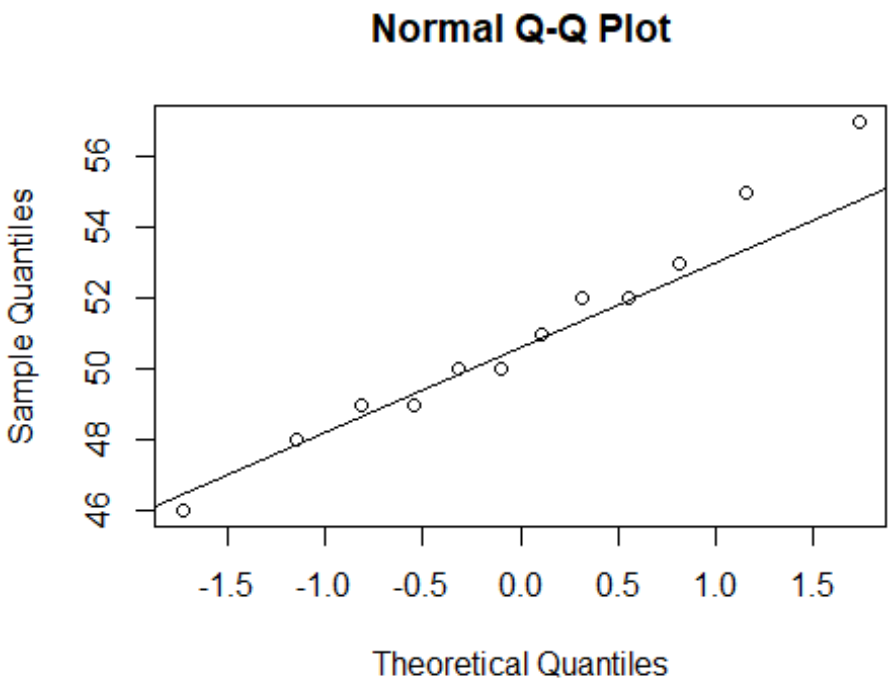
```
## With a density plot
plot(
  density(pommes$rendement),
  col = "black",
  lty = 1,
  main = "Distribution"
)
mr = mean(pommes$rendement, na.rm = TRUE)
sdr = sd(pommes$rendement, na.rm = TRUE)
x_norm = seq(-4,4,length=100) * sdr + mr
lines(
  x = x_norm,
  y = dnorm(x = x_norm, mean = mr, sd = sdr),
  col = "red", lty = 2
)
```



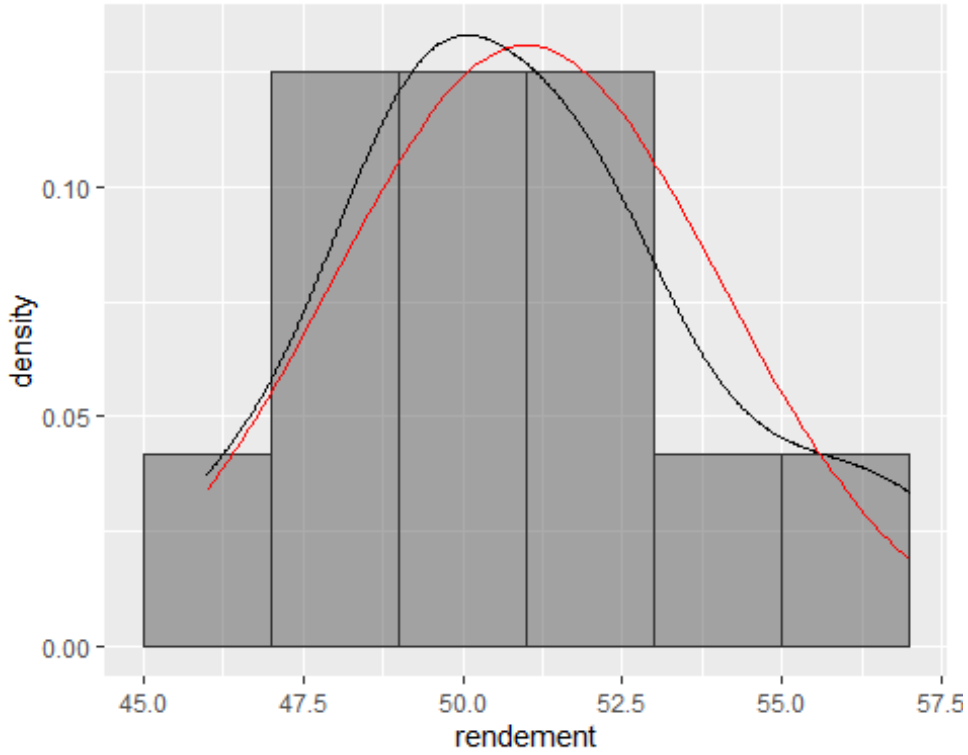

```r
## with histogram
hist(x = pommes$rendement, main = "Histogram")
```



```r
##  with a normal qqplot
qqnorm(pommes$rendement)
qqline(pommes$rendement)
```




```r
## ggplot
ggplot(data = pommes, mapping = aes(x = rendement)) +
  geom_histogram(
    mapping = aes(y=..density..),
    position="identity", binwidth = 2, alpha = 0.5, color = "grey20"
  ) +
  geom_density() +
  stat_function(
    fun = dnorm, n = 101, args = list(mean = mr, sd = sdr),
    color = "red",
    inherit.aes = FALSE
  ) +
  theme(legend.position = "none")
```



Si la validation visuelle n'est pas concluante, on peut utiliser le test de Normalité de Shapiro-Wilk comme suit :

```
shapiro.test(x = pommes$rendement)
##  Shapiro-Wilk normality test
##
## data:  pommes$rendement
## W = 0.97643, p-value = 0.9653
```

On rappelle l'hypothèse nulle $H0$: la variable est normalement distribuée. Si la p-value est inférieure à un niveau alpha choisi (par exemple 0.05), Alors l'hypothèse nulle est rejetée. Pour supposer la normalité des résidus, il est donc nécessaire d'obtenir une p-value > 0.05. (Ici p-value = 0.96, on ne rejette pas $H0$)

Test ANOVA

Formule

La valeur $rendement_{ik}$ est la valeur du rendement pour le k-ème arbre de la varieté i et μ_i est la moyenne inconnue des rendements pour la varieté i.

L'ANOVA est un modèle régression linéaire qui fait l'hypothèse d'une moyenne par modalité du facteur étudié. Son modèle peut s'écrire :

$$rendement_{ik} = \mu_i + \varepsilon_{ik}$$

$$\Leftrightarrow$$

$$rendement_{ik} = \mu + \alpha_i + \varepsilon_{ik}$$

avec

- μ moyenne générale inconnue de tous les rendements,
- α_i effet principal additif par rapport à μ dû à la i-ème modalité du facteur,
- $\varepsilon_{ik} \sim \mathcal{N}(0, \sigma^2)$ les résidus (les écarts entre les observations et les moyennes des groupes auxquels elles sont relatives),
- avec $\sum_{i=1}^{n} \alpha_i = 0$.

Dans R cette formule est écrite ainsi : `rendement ~ variete`.

Réalisation de l'analyse de la variance sous R

- Lire la documentation de la fonction `aov` dans R : `?stats::aov`
- Lire la documentation de la fonction `lm` dans R : `?stats::lm`
- Lire la documentation de la fonction `anova` dans R : `?stats::anova`
- Expliquer la différence et noter ce qui nous intéresse pour répondre à la question initiale.

`?aov` précise l'information suivante :

```
Fit an analysis of variance model by a call to lm for each stratum.
```

`?lm` précise l'information suivante :

```
lm is used to fit linear models. It can be used to carry out regression,
single stratum analysis of variance and analysis of covariance (although aov
may provide a more convenient interface for these).
```

`?anova` précise l'information suivante :

`Compute analysis of variance (or deviance) tables for one or more fitted model objects.`

Le fonction `aov` permet bien de réaliser l'analyse de variance et on voit qu'elle appelle la fonction `lm` pour chaque niveau (les modalités de notre facteur).

Du coup, `lm` convient aussi pour répondre à notre question puisque nous voulons faire une analyse de variance **à un facteur** (single stratum).
On comprend que `lm(rendement ~ variete, data = pommes)` est donc équivalent à `aov(formula = rendement~variete, data = pommes)` dans notre cas.
La sortie de la fonction `aov` est cependant plus adaptée (more convenient interface) pour répondre à la question de l'ANOVA.

La fonction `anova` permet de tester la significativité des prédicteurs donc `anova(lm(rendement ~ variete, data = pommes))` répondrait aussi à notre question.

N.B. : la fonction `anova` permet aussi de tester l'ajout d'un prédicteur entre un modèle nul et un modèle alternatif par exemple (comparison between two or more models), mais ce point ne sera pas abordé ici.

- Réaliser l'ANOVA qui permet de tester l'effet du facteur "variete" sur la mesure "rendement", et afficher le résumé statistique.

```
my_anova <- aov(formula = rendement~variete, data = pommes)
my_anova
```

```
## Call:
##    aov(formula = rendement ~ variete, data = pommes)
##
## Terms:
##                   variete Residuals
## Sum of Squares         56        46
## Deg. of Freedom         2         9
##
## Residual standard error: 2.260777
## Estimated effects may be unbalanced
```

Pour faire le parallèle avec le tableau ANOVA (cf. tableau du cours), l'objet "my_anova" affiche un résultat de la forme :

	Facteur A	Residuelle
SCE	SCE_A	SCE_R
d.d.l.	ddl_A	ddl_R

Et l'estimation de l'écart-type résiduel : $\hat{\sigma} = \sqrt{CM_R}$.

En réalité, “my_anova” est une liste avec 13 composantes dont on trouve les noms en faisant `names(my_anova)`.
Astuce : pour accéder aux éléments d’une liste, on peut commencer à écrire `my_anova$` dans R et utiliser l’auto-complétion avec la touche “TAB” (tabulation) du clavier, ainsi R nous propose les noms connus dans la liste.

Pour aller plus loin dans le tableau d’analyse de la variance, il suffit d’afficher le résumé statistique comme suit :

```
summary(my_anova)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## variete      2     56  28.000   5.478 0.0278 *
## Residuals    9     46   5.111
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ici les informations sont sous la forme suivante :

	d.d.l	SCE	CM	Statistique F	p.value
Facteur A	ddl_A	SCE_A	CM_A	F_{obs}	P(F > F_{obs})
Résiduelle	ddl_R	SCE_R	CM_R		

avec bien sûr $F \sim \mathcal{F}(ddl_A, ddl_R)$

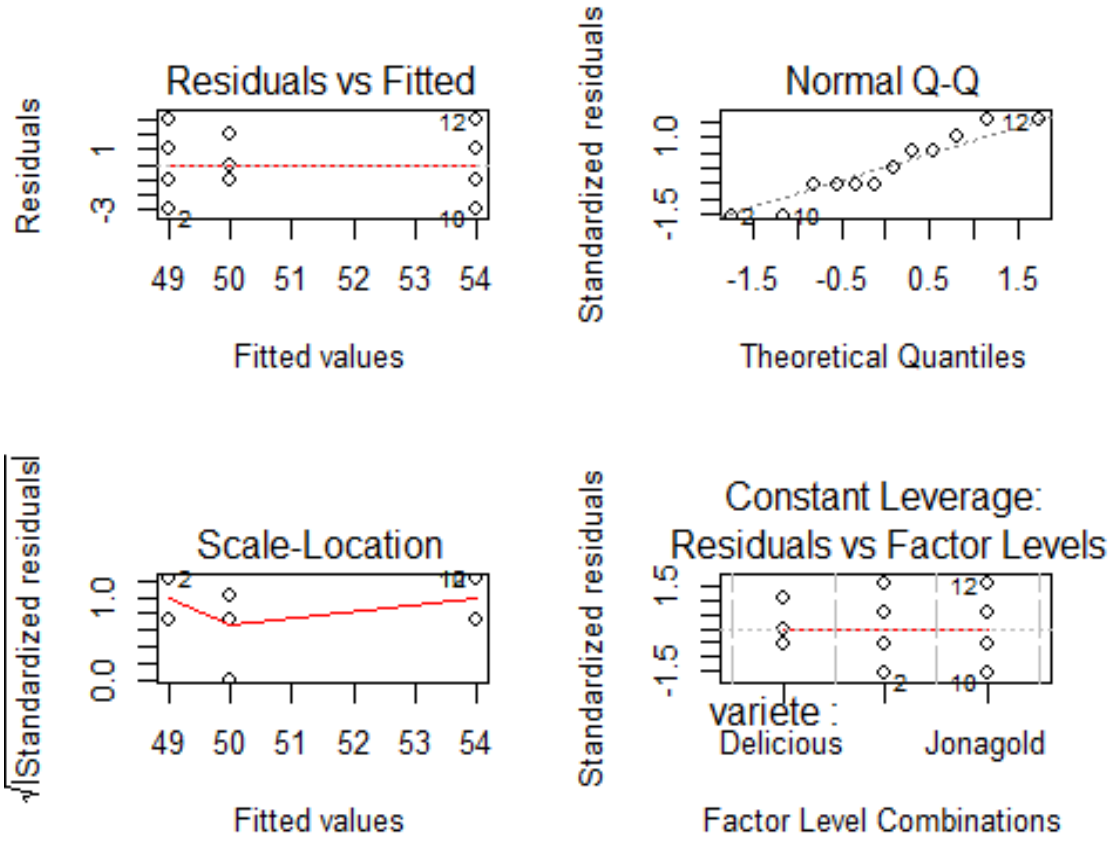
Lorsqu’on s’est fixé un risque α a priori,
on rejette $H0$: “Absence d’effet du facteur” si la $p - valeur < \alpha$.

Donc, en prenant $\alpha = 0.05$, puisque p.value = 0,02778 < 0:05, on rejette l’hypothèse nulle $H0$: $(m1 = m2 = m3)$. Les variétés ont des rendements en jus différents et en disant ceci on n’a que 5 chances sur 100 de se tromper. On conclut en disant qu’il y a un effet variété significatif.
N.B. : Cependant si on prend $\alpha = 0.01$, on ne rejette pas H0. L’effet n’est pas *très* signifiatif.

Diagnostique

- Réaliser 4 graphiques diagnostiques de l’anova.

```
par(mfrow=c(2, 2)); plot(my_anova)
```



- L'absence de corrélation des résidus : Les résidus ne semblent pas corrélés entre eux (Résiduals vs Fitted) puisque la ligne rouge est bien horizontal et sur Y = 0. La valeur des résidus ne semble pas dépendre du traitement puisqu'ils sont tous globalement centrés sur Y = 0 (Residuals vs Factor Levels).
- Pour vérifier la normalité des résidus on regarde le Normal QQplot. Ici les points sont bien répartis le long de la ligne, cela signifie que les résidus sont distribués selon une loi normale. Le fait que les points soient centrés sur 0 (sur l'axe des y), montre que leur moyenne est égale à 0.
- L'hypothèse d'homogénéité des variances, c'est-à-dire l'hypothèse que les résidus ont une variance constante, peut s'évaluer via le graphique "Scale-Location". La méthode graphique consiste à représenter les résidus standardisés en fonction des valeurs prédites (les moyennes des différents facteurs). On voit ici que les dispersions des résidus (leurs écartements verticaux) relatives à chaque modalité de traitement sont globalement identiques, l'hypothèse d'homogénéité des résidus est acceptée.

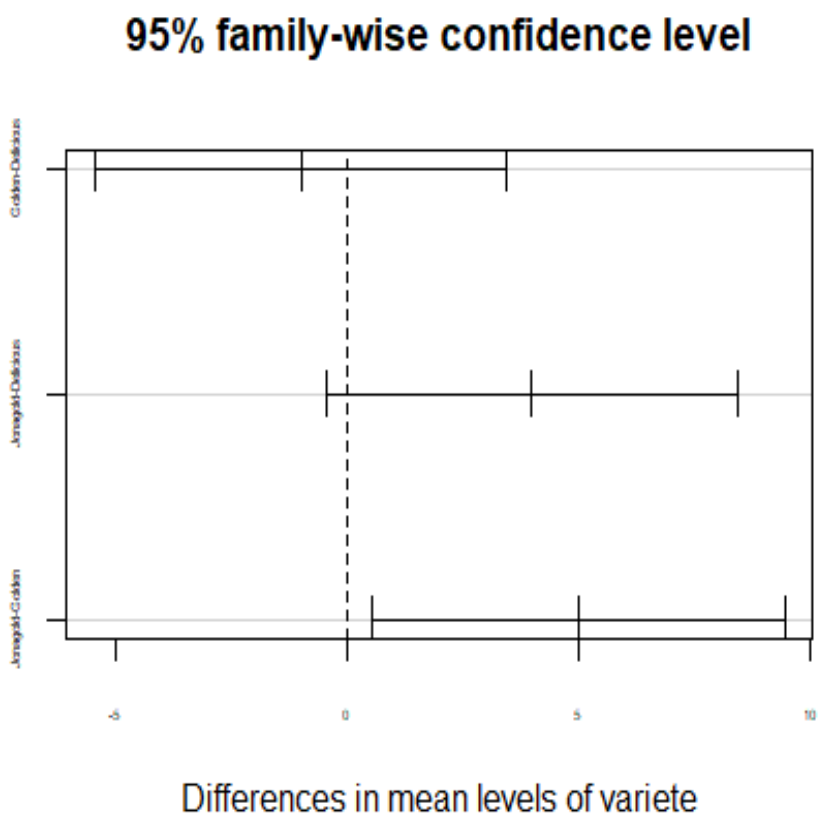
Visuellement, ces graphiques semblent être dans la norme. Les hypothèses stochastiques sont ainsi acceptables et nous pouvons donc "faire confiance" aux résultats du test ANOVA.

Autre solution : On peut aussi accéder aux résidus du modèle ainsi `my_anova$residuals` puis tester leur normalité, leur autocorrélation etc…

Test post-hoc

- Lire la documentation du test HSD de Tukey dans R avec `?TukeyHSD`.
- Réaliser ce test post-hoc.

```
TukeyHSD(x = my_anova)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = rendement ~ variete, data = pommes)
##
## $variete
##                     diff        lwr     upr     p adj
## Golden-Delicious     -1 -5.4633295 3.46333 0.8101561
## Jonagold-Delicious    4 -0.4633295 8.46333 0.0784642
## Jonagold-Golden       5  0.5366705 9.46333 0.0296317

par(cex.axis=0.4)
plot(TukeyHSD(my_anova))
```



- Quelle est la particularité des p-valeurs retournées ?

Ce test nous fournit des p-valeurs ajustées dans la colonne “p adj”. Il y a d’ores et déjà prise en compte de la correction de tests multiples.

- Quelle est la conclusion du test post-hoc ?

C’est la variété “Jonagold” qui possède, en moyenne, un rendement différent des autres.

- Pour l’exercice, tester ces autres commandes.

```r
# exemple
pairwise.t.test(x = pommes$rendement, g = pommes$variete, p.adj = "bonf")
pairwise.t.test(x = pommes$rendement, g = pommes$variete, p.adj = "holm")
```

N.B. : IMPORTANT

Dans la pratique, UNE SEULE méthode, et UNE SEULE correction, serait à choisir A PRIORI (de même que le seuil alpha). Attention faire une multitude de tests, et choisir A POSTERIORI le meilleur (celui qui nous arrange) serait du “p-hacking” (comprendre manipuler les données ou les résultats pour avoir une bonne p-valeur).

Étude de cas 2 : Étude de l'IMC

Pour plusieurs pays, des centres hospitaliers ont recrutés des patients pour étudier leur poids, la prise en charge de leur diabète et leur profil génétique.
Afin de réaliser une étude au niveau européen, les données sont réunies dans une cohorte pour être mise en commun.
Pour réaliser une étude équilibrée, on souhaite vérifier si les critères de recrutement des patients soient homogènes entre les pays. On se pose la question suivante :
Existe-t-il une différence significative entre ces pays quant à la moyenne des IMC ?

- Lire les données. `?read.csv()`

```
cohort <- read.csv(
  file = "./data/cohort.csv",
  header = TRUE
)
```

- Formater les données (avec le bon type) et faire en sorte que la modalité "France" soit la référence.

```
str(cohort)
```

```
## 'data.frame':    147 obs. of  5 variables:
##  $ MUTATION : chr  "NON" "OUI" "NON" "OUI" ...
##  $ id_centre: int  45 45 38 45 23 5 23 14 16 17 ...
##  $ sex      : chr  "FEMININ" "FEMININ" "FEMININ" "FEMININ" ...
##  $ bmi      : num  26.1 24.4 23.5 24.4 26.1 ...
##  $ Pays     : chr  "Pays-Bas" "Pays-Bas" "Italie" "Pays-Bas" ...
```

```
cohort$Pays <- as.factor(cohort$Pays) ## force la conversion en facteur, pour toutes les modalités présentes (bonne ou mauvaise...).
cohort$Pays <- relevel(x = cohort$Pays, ref = "France")
```

Quand un modèle utilise une variable factorielle, il est important de connaitre la référence. Par défaut ce sera la première modalité (ordre alphanumérique).

N.B. : Quand on réalise une analyse "cas vs control", si on a un phénotype codé 1/0, tout va bien, les contrôles (0) seront bien pris en référence. Mais on voit le problème si les phénotypes étaient indiqués avec le code "cas" et "control", ici l'ordre alphabétique ne donnera pas la modalité "control" en référence.

- Résumer les données pour en voir un aperçu numérique.

```
summary(cohort$bmi)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   17.63   22.87   25.83   25.75   28.38   37.72       1
```

```
table(cohort$Pays, useNA = "always")
```

```
##
##    France           Allemagne  Belgique   Espagne    Italie  Pays-Bas  Por
tugal
##        18         1        15        11        41        14        33
13
##      <NA>
##         1
```

Les données `bmi` semble être cohérente (intervalle de valeurs probables) et il y a une donnée manquante qui sera à exclure. Les données `Pays` ont bien comme référence “France” (première modalité), mais il a une modalité vide "". Il semble y avoir un problème…

- D’où vient le problème avec les données renseignées dans la colonne “Pays”?

Il y a une observation manquante correctement lu “ <NA> ” et une autre qui été renseignée ainsi "“, ce qui a été pris pour une valeur. L’option ”na.strings" de la fonction `read.csv` que nous avons utilisé plus haut précise bien : pour une colonne de caractères (“strings”), une valeur NA sera interprétée manquante. Donc ici un champ vide n’est pas interprété comme une valeur manquante à la lecture.

- Tester les commandes suivantes et expliquer les sorties.

```
sum(is.na(cohort$Pays))
```

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

```
sum(cohort$Pays=="")
```

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

```
sum(cohort$Pays%in%"")
```

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

Dans les 3 cas, `sum` va sommer le vecteur booléen retourné par un test. Les valeurs TRUE, c’est à dire “VRAI”, valent 1 et les FALSE valent 0.

`is.na()` test la présence de valeurs manquantes.
`==` réalise un test d’égalité de façon vectorielle avec la valeur "". Mais on voit que la présence de `NA` pose problème pour faire la somme.
`%in%` test une inclusion et sait gérer les valeurs NA.

- Traiter les valeurs manquantes

```
# filter NA Pays
cohort <- cohort[!is.na(cohort$Pays), ]
cohort <- cohort[!cohort$Pays %in% "", ]
# filter NA bmi
cohort <- cohort[!is.na(cohort$bmi), ]
```

```r
## dplyr
cohort <- cohort %>%
  filter(!is.na(Pays)) %>%
  filter(!Pays %in% "") %>%
  filter(!is.na(bmi))

# mise à jour des niveaux
levels(cohort$Pays)
```

```
## [1] "France"    ""          "Allemagne" "Belgique"  "Espagne"   "Italie"
## [7] "Pays-Bas"  "Portugal"
```

```r
cohort$Pays <- factor(
  x = cohort$Pays,
  levels = c("France", "Allemagne", "Belgique", "Espagne", "Italie", "Pays-Ba
s", "Portugal")
)
levels(cohort$Pays)
```

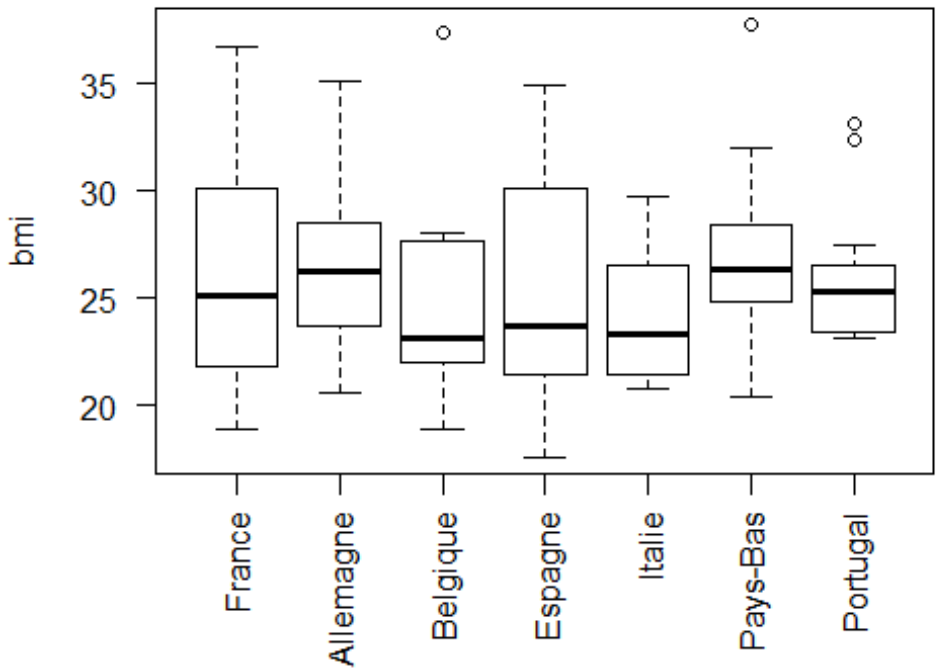
```
## [1] "France"    "Allemagne" "Belgique"  "Espagne"   "Italie"    "Pays-Bas"
## [7] "Portugal"
```

Noter que le fait d'enlever des observations n'impacte pas les niveaux de facteurs définis au moment du formatage. On peut définir explicitement les niveaux à considérer avec le paramètre "levels" de la fonction "factor" (ainsi que la façon de les afficher avec "labels").

N.B. : Notre choix ici est de retirer les individus ayant des données manquantes. Remarquer qu'une autre possibilité pourrait être d'imputer ces données (l'imputation est par exemple souvent utilisée pour les données génotypiques par exemple).

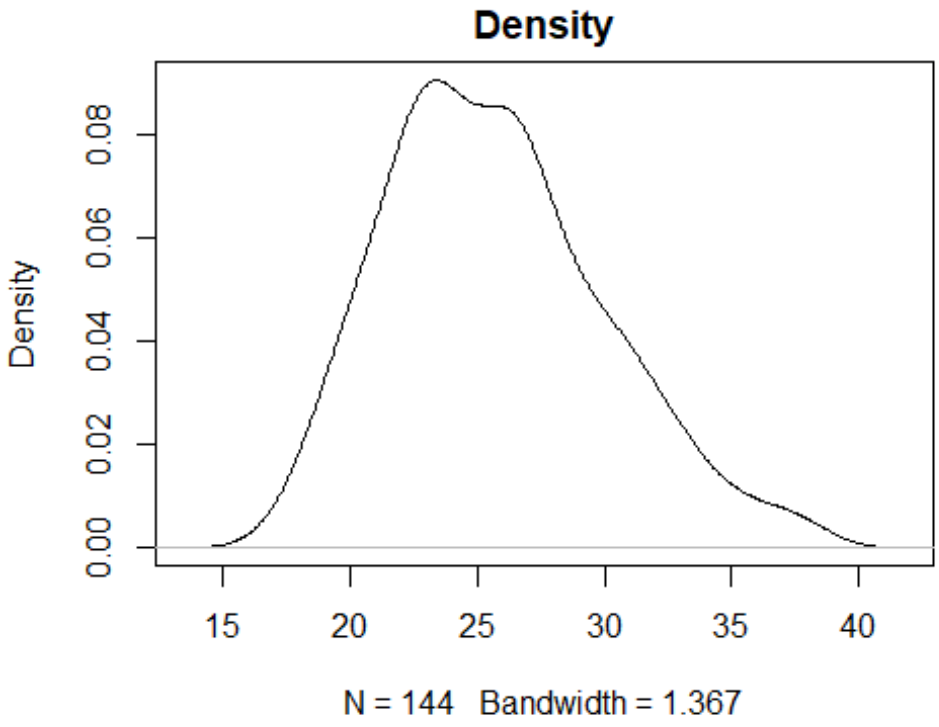
- Visualiser les données (avec tous les noms de pays lisible)

```r
par(mar=c(7,4,1,1))
## c(bottom, left, top, right) ## by default  c(5, 4, 4, 2) + 0.1
boxplot(bmi~Pays, data = cohort, las = 2, xlab = "")
```



- Contrôler la normalité de la variable d'intérêt

```
par(mar=c(5,4,2,1))
plot(density(cohort$bmi), main = "Density")
```



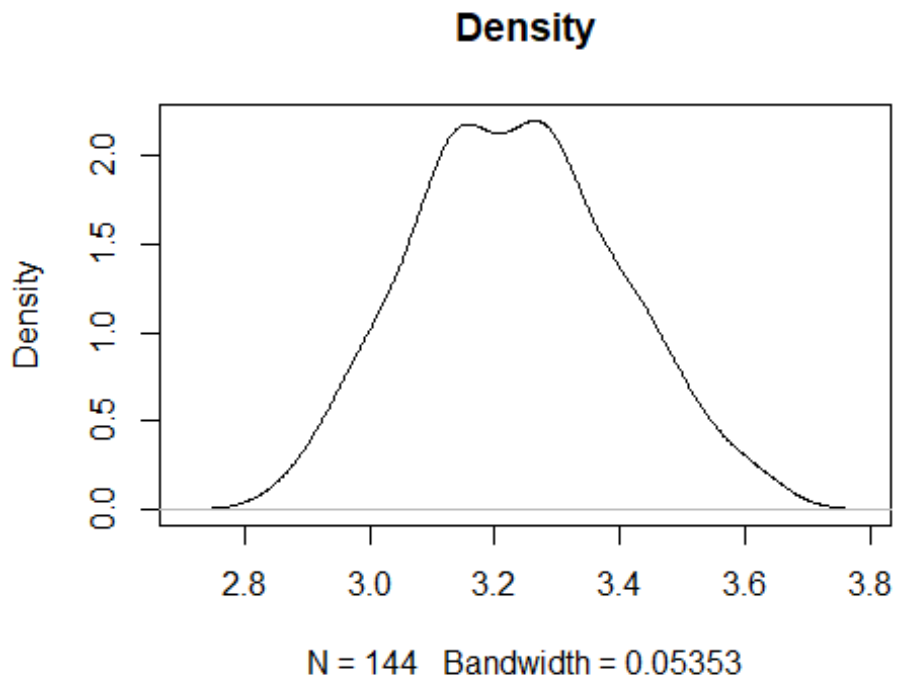
```r
shapiro.test(cohort$bmi)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  cohort$bmi
## W = 0.97322, p-value = 0.006356
```

- Que faire si notre variable d'intérêt n'est pas normalement distribuée ?

Appliquer transformation va résoudre notre problème : ici on va étudier log(bmi).

```r
cohort$log_bmi <- log(cohort$bmi)
plot(density(cohort$log_bmi), main = "Density")
```




```r
shapiro.test(cohort$log_bmi)
```

- Réaliser l'ANOVA

On rappelle les hypothèses
$H0$: L'égalité des moyennes entre tous les pays.
contre
$H1$: Au moins un des pays à une moyenne différente des autres.

```
my_anova_pays <- aov(formula = log_bmi ~ Pays, data = cohort)
summary(my_anova_pays)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Pays          6  0.190 0.03174   1.204  0.308
## Residuals   137  3.612 0.02637
```

- Donner une réponse à la question initiale.

Dans le résumé, on peut lire la p-valeur dans la colonne "Pr(>F)" valant 0.308. Elle est supérieur à $\alpha = 0.05$ donc **on ne peut pas conclure au rejet** de $H0$.

- Comment se comporte l'ANOVA, dans R, si notre jeu de données possède des données manquantes ?

`na.action` est une option pour définir comment le modèle doit se comporter avec les données manquantes. Par défaut les observations avec une valeur manquante auraient été supprimées de l'analyse. Mais il est toujours mieux de les traiter soi-même pour avoir connaissance de ses données et des effectifs.

Pour la preuve de concept, affecter à NA une observation du jeu de données "cohort" et retester l'anova.

```
cohort$log_bmi[1] <- NA
aov(formula = log_bmi ~ Pays, data = cohort)
```

```
## Call:
##    aov(formula = log_bmi ~ Pays, data = cohort)
##
## Terms:
##                     Pays Residuals
## Sum of Squares  0.190219  3.612017
## Deg. of Freedom        6       136
##
## Residual standard error: 0.1629692
## Estimated effects may be unbalanced
## 1 observation deleted due to missingness
```

Sources

- Le contenu de ce TP s'est basé sur un extrait du support écrit par Christophe Chesneau.
- le livre R Cookbook, 2nd Edition, James (JD) Long, Paul Teetor, 2019-09-26
- les pages suivantes :

https://statistique-et-logiciel-r.com/anova-a-un-facteur-partie-1/

https://statistique-et-logiciel-r.com/anova-a-un-facteur-partie-2-la-pratique/

Pour aller plus loin :

- Pas utilisé ici, mais il existe aussi la fonction `Anova` provenant du package car : https://www.rdocumentation.org/packages/car/versions/3.0-10/topics/Anova
- Pas utilisé ici, mais il existe aussi ce package `DescTools` qui propose la fonction `PostHocTest` pour réaliser les tests post-hoc : https://www.rdocumentation.org/packages/DescTools/versions/0.99.38/topics/PostHocTest