

## TP analyse multivarié

### 1. Statistiques descriptives

Les moyennes du diamètre des arbres varient selon les catégories du facteur *Statut*.

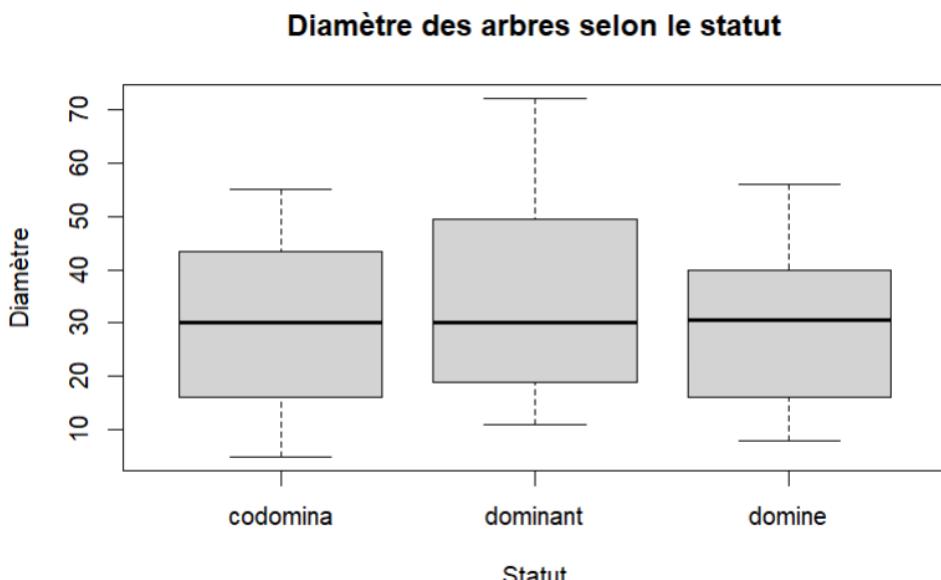
Le tableau de contingence montre également que les effectifs ne sont pas strictement équilibrés, mais restent suffisants pour une ANOVA.

Le boxplot illustre visuellement les différences potentielles de distributions entre les statuts.

```

> arbres <- read.csv("arbres.csv",
+                      header = TRUE,
+                      sep = ",",
+                      stringsAsFactors = FALSE)
> # Vérifier les premières lignes
> head(arbres)
  X8 codomina
1 15 codomina
2 22 codomina
3 20 codomina
4 17 codomina
5 21 codomina
6 45 codomina
> # Renommer les colonnes si nécessaire
> colnames(arbres) <- c("Diametre", "Statut")
> # Conversion en numérique
> arbres$Diametre <- as.numeric(as.character(arbres$Diametre))
> # Vérifier structure
> str(arbres)
'data.frame': 103 obs. of 2 variables:
 $ Diametre: num 15 22 20 17 21 45 31 5 50 32 ...
 $ Statut   : chr "codomina" "codomina" "codomina" "codomina" ...
> # Moyenne du diamètre par statut
> by(arbres$Diametre, arbres$Statut, mean)
arbres$Statut: codomina
[1] 30.10417
-----
arbres$Statut: dominant
[1] 35.13333
-----
arbres$Statut: domine
[1] 28.73438
>
> # Effectif par groupe
> table(arbres$Statut)

codomina dominant domine
      24       15       64
>
> # Boxplot
> par(mfrow = c(1,1))
> boxplot(Diametre ~ Statut, data = arbres,
+           main = "Diamètre des arbres selon le statut",
+           xlab = "Statut",
+           ylab = "Diamètre")
  
```



## 2. Comparaison méthode 1 (modèle linéaire) et méthode 2 (ANOVA)

### Méthode 1 : Modèle linéaire (lm)

Le modèle linéaire Diamètre ~ Statut estime un intercept correspondant au premier niveau du facteur, puis des coefficients *additionnels* pour les niveaux suivants (contrastes de traitement).

#### Interprétation

- Le test F dans le modèle linéaire est équivalent à celui de l'ANOVA.
- Les coefficients indiquent comment chaque groupe diffère du groupe de référence.
- Le résumé (summary(modelarbres)) donne les *p-values* pour chaque comparaison individuelle contre le groupe de base.

#### Limite de la méthode linéaire simple

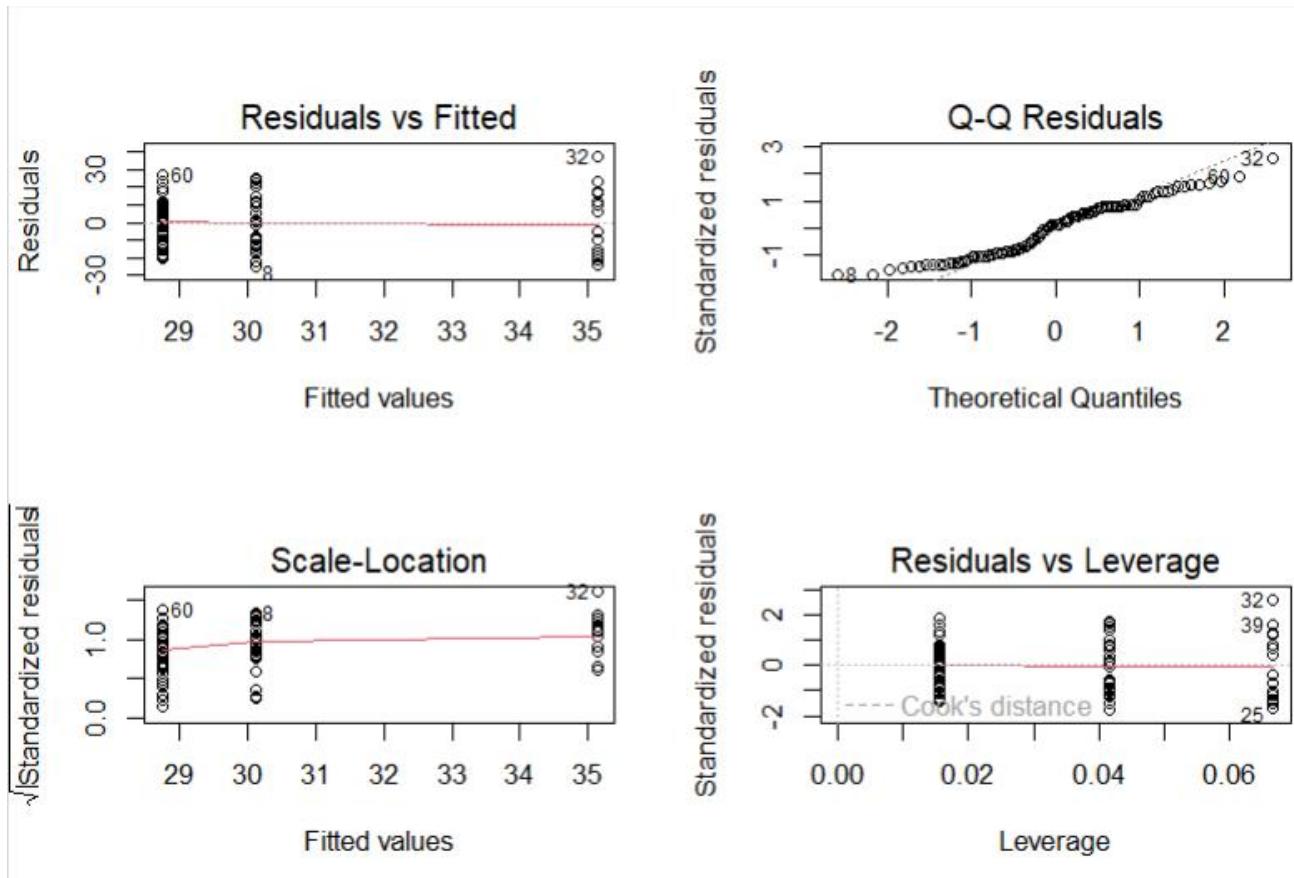
Elle ne teste que les contrastes *vs groupe de référence* → pas toutes les comparaisons deux-à-deux.

```
# -----
# 3. Méthode 1 : Modèle linéaire simple (lm)
# -----

modelarbres <- lm(Diametre ~ Statut, data = arbres)

# Diagnostics
par(mfrow = c(2,2))
plot(modelarbres)

# Résumé du modèle
summary(modelarbres)
```



```
> # Résumé du modèle
> summary(modelarbres)

Call:
lm(formula = Diametre ~ Statut, data = arbres)

Residuals:
    Min      1Q  Median      3Q     Max 
-25.104 -13.734   1.266  11.266  36.867 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 30.104     3.008 10.007 <2e-16 ***
Statutdominant 5.029     4.851  1.037  0.302    
Statutdomine -1.370     3.527 -0.388  0.699    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.74 on 100 degrees of freedom
Multiple R-squared:  0.02242, Adjusted R-squared:  0.002865 
F-statistic: 1.147 on 2 and 100 DF,  p-value: 0.3219
```

## Méthode 2 : ANOVA + tests post-hoc

L'ANOVA (`aov(Diametre ~ Statut)`) teste l'hypothèse globale :

H0 : les moyennes des diamètres sont égales pour tous les niveaux du Statut

- Si la p-value de l'ANOVA est significative, cela signifie qu'au moins un groupe diffère.
- Ensuite, les tests post-hoc identifient où sont ces différences.

Tests post-hoc :

- pairwise.t.test (sans correction) → risque d'erreur de type I élevé.
- Bonferroni → très strict, réduit la probabilité de faux positifs.
- Holm → compromis optimal (plus puissant que Bonferroni).
- TukeyHSD → recommandé quand les groupes ont des tailles égales ou proches.

```

# -----
# 4. Méthode 2 : ANOVA + tests post-hoc
# -----


# ANOVA
ins.aov <- aov(Diametre ~ Statut, data = arbres)
summary(ins.aov)

# Tests post-hoc sans correction
comp.statut <- pairwise.t.test(arbres$Diametre, arbres$Statut,
                                p.adjust.method = "none")
comp.statut

# Avec correction Bonferroni
pairwise.t.test(arbres$Diametre, arbres$Statut,
                p.adjust.method = "bonferroni")

# Test de Holm (à faire)
pairwise.t.test(arbres$Diametre, arbres$Statut,
                p.adjust.method = "holm")

> # ANOVA
> ins.aov <- aov(Diametre ~ Statut, data = arbres)
> summary(ins.aov)
      Df Sum Sq Mean Sq F value Pr(>F)
Statut      2    498   249.0   1.147  0.322
Residuals 100  21718   217.2
>
> # Tests post-hoc sans correction
> comp.statut <- pairwise.t.test(arbres$Diametre, arbres$Statut,
+                                 p.adjust.method = "none")
> comp.statut

  Pairwise comparisons using t tests with pooled SD

data: arbres$Diametre and arbres$Statut

codomina dominant
dominant 0.30      -
domine   0.70      0.13

P value adjustment method: none
>
> # Avec correction Bonferroni
> pairwise.t.test(arbres$Diametre, arbres$Statut,
+                  p.adjust.method = "bonferroni")

  Pairwise comparisons using t tests with pooled SD

data: arbres$Diametre and arbres$Statut

codomina dominant
dominant 0.91      -
domine   1.00      0.40

P value adjustment method: bonferroni

```

```

> # Test de Holm (à faire)
> pairwise.t.test(arbres$Diametre, arbres$Statut,
+                  p.adjust.method = "holm")
Pairwise comparisons using t tests with pooled SD

data: arbres$Diametre and arbres$Statut

codomina dominant
dominant 0.6      -
domine   0.7      0.4

P value adjustment method: holm

# -----
# 5. Tests plus avancés (car, multcomp, DescTools)
# ----

# install.packages("car")
# install.packages("multcomp")
# install.packages("DescTools")

# Test de Tukey (plus robuste)
TukeyHSD(ins.aov)

> # -----
> # 5. Tests plus avancés (car, multcomp, DescTools)
> # -----
>
> # install.packages("car")
> # install.packages("multcomp")
> # install.packages("DescTools")
>
> # Test de Tukey (plus robuste)
> TukeyHSD(ins.aov)
  Tukey multiple comparisons of means
  95% family-wise confidence level

Fit: aov(formula = Diametre ~ Statut, data = arbres)

$Statut
            diff      lwr      upr      p adj
dominant-codomina 5.029167 -6.510888 16.569222 0.5555109
domine-codomina  -1.369792 -9.761929  7.022345 0.9203116
domine-dominant   -6.398958 -16.456799  3.658883 0.2888982

```

## Comparaison des résultats entre les deux méthodes

Méthode	Test global	Tests de comparaison	Forces	Limites
1. Modèle linéaire	Oui (équivalent ANOVA)	Seulement contre le groupe de référence	Simple, interprétable	Ne compare pas tous les groupes

Méthode	Test global	Tests de comparaison	Forces	Limites
2. ANOVA + post-hoc	Oui	Compare toutes les paires	Offre une vision complète, contrôle de l'erreur	Peut être plus complexe

Dans la pratique,

- lm répond : "Est-ce que chaque groupe diffère du groupe de référence ?"
- ANOVA + Tukey/Holm répond : "Quels groupes diffèrent entre eux ?"

→ Les conclusions finales doivent s'appuyer sur les tests post-hoc corrigés (Holm ou Tukey).

### 3. forcer une contrainte sur les coefficients du modèle linéaire :

par exemple :

$$\alpha_1 + \alpha_2 = 0$$

Dans R, le modèle lm() standard avec un facteur utilise par défaut un encodage "contr.treatment", où le premier niveau = 0, et les autres coefficients sont par rapport à ce niveau. C'est pourquoi par défaut on a  $\alpha_1 = 0$  pour le niveau de référence.

Pour imposer  $\alpha_1 + \alpha_2 = 0$ , il faut changer le contraste du facteur. On peut utiliser contr.sum() (contrastes "sum-to-zero") :

```
#-----
#travail a faire : alpha1+alpha2=0
#-----
# Forcer sum-to-zero pour Statut
arbres$Statut <- factor(arbres$Statut) # s'assurer que c'est un facteur
contrasts(arbres$Statut) <- contr.sum(length(levels(arbres$Statut)))

# Vérifier les contrastes
contrasts(arbres$Statut)

# Modèle linéaire avec contrainte alpha1 + alpha2 + ... + alphaK = 0
modelarbres_sum <- lm(Diametre ~ Statut, data = arbres)

summary(modelarbres_sum)
```

```

> #-----
> #travail a faire : alpha1+alpha2=0
> #
> # Forcer sum-to-zero pour Statut
> arbres$Statut <- factor(arbres$Statut) # s'assurer que c'est un facteur
> contrasts(arbres$Statut) <- contr.sum(length(levels(arbres$Statut)))
>
> # Vérifier les contrastes
> contrasts(arbres$Statut)
 [,1] [,2]
codomina   1    0
dominant    0    1
domine     -1   -1
>
> # Modèle linéaire avec contrainte alpha1 + alpha2 + ... + alphaK = 0
> modelarbres_sum <- lm(Diametre ~ Statut, data = arbres)
>
> summary(modelarbres_sum)

Call:
lm(formula = Diametre ~ Statut, data = arbres)

Residuals:
    Min      1Q  Median      3Q     Max 
-25.104 -13.734   1.266  11.266  36.867 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 31.324     1.730   18.111 <2e-16 ***
Statut1     -1.220     2.451   -0.498   0.620    
Statut2      3.809     2.796   1.362   0.176    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.74 on 100 degrees of freedom
Multiple R-squared:  0.02242, Adjusted R-squared:  0.002865 
F-statistic: 1.147 on 2 and 100 DF,  p-value: 0.3219

```