

TP R 2 - Visualisation et Tests Statistiques

L. Ziad

EST-Essaouira

Partie 1 : Visualisation avancee avec ggplot2

Exercice 1 : Prise en main de ggplot2

```
1  # Chargement du package
2  library(ggplot2)
3
4  # Donnees d'exemple
5  data(mtcars)
6  data(iris)
7
8  # Graphique simple
9  ggplot(mtcars, aes(x = wt, y = mpg)) +
10 geom_point() +
11 labs(title = "Poids vs Consommation",
12 x = "Poids (tonnes)",
13 y = "Miles par gallon")
14
```

Exercice 2 : Types de graphiques

```
1  # Histogramme avec courbe de densite
2  ggplot(mtcars, aes(x = mpg)) +
3  geom_histogram(aes(y = ..density..), bins = 15, fill = "lightblue") +
4  geom_density(color = "red") +
5  theme_minimal()
6
7  # Boxplot par groupe
8  ggplot(iris, aes(x = Species, y = Sepal.Length, fill = Species)) +
9  geom_boxplot() +
10 labs(title = "Longueur des sepales par espece")
11
12 # Barplot avec valeurs
13 donnees_agregees <- aggregate(Sepal.Length ~ Species, iris, mean)
14 ggplot(donnees_agregees, aes(x = Species, y = Sepal.Length, fill =
15 Species)) +
16 geom_col() +
17 geom_text(aes(label = round(Sepal.Length, 2)), vjust = -0.5)
```

Partie 2 : Tests statistiques

Exercice 3 : Tests parametriques

```
1 notes <- c(12,14,11,15,13,16,10,12,14,13)
2 t.test(notes, mu = 12)
3
4 groupe_A <- c(12,14,11,15,13)
5 groupe_B <- c(16,18,15,17,16)
6 t.test(groupe_A, groupe_B)
7
8 shapiro.test(notes)
9 qqnorm(notes)
10 qqline(notes)
11
```

Exercice 4 : ANOVA

```
1 # ANOVA
2 resultat_anova <- aov(Sepal.Length ~ Species, data = iris)
3 summary(resultat_anova)
4
5 # Post-hoc
6 TukeyHSD(resultat_anova)
7
8 # Homogeneite variances
9 bartlett.test(Sepal.Length ~ Species, data = iris)
10
```

Partie 3 : Analyses avancees

Exercice 5 : Tests non parametriques

```
1 wilcox.test(groupe_A, groupe_B)
2
3 kruskal.test(Sepal.Length ~ Species, data = iris)
4
5 tableau_contingence <- matrix(c(30,20,25,35), nrow=2)
6 chisq.test(tableau_contingence)
7
```

Exercice 6 : Analyse de puissance

```
1 library(pwr)
2
3 pwr.t.test(n=30, d=0.5, sig.level=0.05)
4
5 pwr.t.test(power=0.8, d=0.5, sig.level=0.05)
6
```

Partie 4 : Rapport automatique

Exercice 7 : Fonction d'analyse

```
1 analyse_complete <- function(x, y) {  
2   cat("=== ANALYSE STATISTIQUE ===\n")  
3   cat("Test de correlation:\n")  
4   print(cor.test(x, y))  
5  
6   cat("\nRegression lineaire:\n")  
7   modele <- lm(y ~ x)  
8   print(summary(modele))  
9  
10  cat("\nDiagnostics graphiques:\n")  
11  par(mfrow=c(2,2))  
12  plot(modele)  
13 }  
14  
15 set.seed(789)  
16 x_test <- rnorm(100)  
17 y_test <- 2*x_test + rnorm(100,0,0.5)  
18 analyse_complete(x_test, y_test)  
19
```