

Guide Détaillé des Commandes R pour la Régression Simple

1. La Commande `plot()`

Fonction : Créer une représentation graphique bivariable (nuage de points) **Syntaxe complète :**

```
plot(formule, data, xlab, ylab, main, col, pch, ...)
```

Paramètres détaillés :

- `formule` : $Y \sim X$ (variable à expliquer ~ variable explicative)
- `data` : jeu de données
- `xlab` : titre de l'axe X (optionnel)
- `ylab` : titre de l'axe Y (optionnel)
- `main` : titre du graphique (optionnel)
- `col` : couleur des points (optionnel)
- `pch` : type de points (optionnel)

Exemple détaillé :

```
# Données d'exemple
donnees <- data.frame(
  temperature = c(20, 22, 25, 27, 30),
  ventes = c(100, 120, 140, 150, 180)
)

# Création du graphique
plot(ventes ~ temperature,
     data = donnees,
     xlab = "Température (°C)",
     ylab = "Ventes (unités)",
     main = "Relation Température-Ventes",
     col = "blue",
     pch = 16)
```

2. La Commande `lm()` (Linear Model)

Fonction : Effectuer une régression linéaire **Syntaxe complète :**

```
lm(formule, data, subset, weights, na.action, ...)
```

Paramètres détaillés :

- `formule` : $Y \sim X$ (relation à modéliser)
- `data` : jeu de données
- `subset` : sous-ensemble de données (optionnel)
- `weights` : pondérations (optionnel)
- `na.action` : gestion des valeurs manquantes (optionnel)

Exemple détaillé :

```
# Modèle simple
modele1 <- lm(ventes ~ temperature, data = donnees)

# Modèle avec sous-ensemble
modele2 <- lm(ventes ~ temperature,
              data = donnees,
              subset = temperature > 22)
```

3. La Commande `summary()`

Fonction : Produire un résumé statistique complet du modèle **Syntaxe** :

```
summary(object, ...)
```

Éléments retournés :

- Coefficients de régression
- Erreurs standards
- Statistiques t
- P-values
- R^2 et R^2 ajusté
- Statistique F
- Analyse des résidus

Exemple détaillé :

```
# Création et analyse du modèle
modele <- lm(ventes ~ temperature, data = donnees)
resultat <- summary(modele)

# Accès aux différents éléments
coefficients <- resultat$coefficients
r_carre <- resultat$r.squared
r_carre_ajuste <- resultat$adj.r.squared
```

4. La Commande `abline()`

Fonction : Ajouter une droite à un graphique existant **Syntaxe complète :**

```
abline(a, b, h, v, reg, coef, col, lwd, lty, ...)
```

Paramètres détaillés :

- `reg` ou `coef` : modèle de régression ou coefficients
- `col` : couleur de la droite
- `lwd` : épaisseur de la ligne
- `lty` : type de ligne

Exemple détaillé :

```
# Création du graphique et ajout de la droite
plot(ventes ~ temperature, data = donnees)
modele <- lm(ventes ~ temperature, data = donnees)
abline(modele,
       col = "red",
       lwd = 2,
       lty = 1)
```

5. La Commande `predict()`

Fonction : Effectuer des prédictions à partir du modèle **Syntaxe complète :**

```
predict(object, newdata, interval, level, ...)
```

Paramètres détaillés :

- `object` : modèle de régression
- `newdata` : nouvelles données pour la prédiction
- `interval` : type d'intervalle ("none", "confidence", "prediction")
- `level` : niveau de confiance

Exemple détaillé :

```
# Création des nouvelles données
nouvelles_temperatures <- data.frame(temperature = c(23, 26, 29))

# Prédiction simple
predictions <- predict(modele, nouvelles_temperatures)

# Prédiction avec intervalle de confiance
predictions_ic <- predict(modele,
                          nouvelles_temperatures,
                          interval = "confidence",
                          level = 0.95)
```

6. La Commande `round()`

Fonction : Arrondir des valeurs numériques **Syntaxe complète** :

```
round(x, digits = 0)
```

Paramètres détaillés :

- `x` : valeur(s) à arrondir
- `digits` : nombre de décimales

Exemple détaillé :

```
# Arrondir une valeur
x <- 3.14159
round(x, digits = 2) # Retourne 3.14

# Arrondir un vecteur
valeurs <- c(3.14159, 2.71828, 1.41421)
round(valeurs, digits = 3) # Retourne c(3.142, 2.718, 1.414)
```

7. La Commande `data.frame()`

Fonction : Créer un tableau de données **Syntaxe complète** :

```
data.frame(..., row.names = NULL, check.rows = FALSE,
            check.names = TRUE, fix.empty.names = TRUE,
            stringsAsFactors = FALSE)
```

Exemple détaillé :

```
# Création d'un dataframe pour les prédictions
donnees_prediction <- data.frame(
  temperature = c(24, 26, 28),
  stringsAsFactors = FALSE
)
```