

V1_CLAUD

Statistique Descriptive avec R : Guide Complet

1. Création et Manipulation de Données

1.1 Création d'un Vecteur

```
elev <- c(138, 164, 150, 132, 144, ...)
```

- **Fonction** : `c()` (concatenate)
- **Objectif** : Créer un vecteur de données
- **Syntaxe** : `nom_variable <- c(valeur1, valeur2, ...)`
- **Exemple** : Créer un vecteur de températures

```
temperatures <- c(22, 25, 23, 20, 26)
```

1.2 Création d'un DataFrame

```
data = data.frame(cbind(c(rep("homme",4),rep("femme",5)), c(1:4,3:7)))  
names(data) = c("genre","score")  
data$score = as.numeric(data$score)
```

- **Fonctions** :
 - `data.frame()` : Créer un tableau de données
 - `cbind()` : Combiner des colonnes
 - `rep()` : Répéter des valeurs
 - `c(1:4, 3:7)` : Créer une séquence
 - `names()` : Nommer les colonnes
 - `as.numeric()` : Convertir en valeurs numériques
- **Exemple** :

```
donnees = data.frame(nom = c("Alice", "Bob"), age = c(25, 30))
```

1.3 Lecture de Fichiers

```
data = read.table(file.choose(), header=T)
```

- **Fonction** : `read.table()`
- **Paramètres** :
 - `file.choose()` : Ouvrir une fenêtre de sélection

- `header=T` : Utiliser la première ligne comme en-tête
- **Exemple :**

```
donnees_csv = read.table("mon_fichier.csv", sep=";", header=T)
```

2. Calculs Statistiques Descriptifs

2.1 Moyenne Simple

```
mean(elev)
```

- **Fonction :** `mean()`
- **Objectif :** Calculer la moyenne arithmétique
- **Exemple :**

```
notes = c(15, 16, 17, 14)
moyenne = mean(notes) # Retourne 15.5
```

2.2 Moyenne Pondérée

```
weighted.mean(x, wt)
```

- **Fonction :** `weighted.mean()`
- **Objectif :** Calculer une moyenne avec des poids
- **Exemple :**

```
valeurs = c(10, 20, 30)
poids = c(0.3, 0.4, 0.3)
moyenne_ponderee = weighted.mean(valeurs, poids)
```

3. Visualisation Graphique

3.1 Histogramme

```
hist(elev,
     breaks = seq(from = min(elev), to = max(elev), length = 6),
     col = grey(0.8),
     labels = TRUE,
     las = 1,
     xlab = "Taille (cm)",
     ylab = "Nombre d'élèves",
     main = "Avec un découpage en 5 classes")
```

- **Fonction :** `hist()`

- **Paramètres :**
 - `breaks` : Nombre ou séquence de classes
 - `col` : Couleur
 - `labels` : Afficher les effectifs
 - `las` : Orientation des étiquettes
 - `xlab, ylab, main` : Légendes
- **Exemple :**

```
hist(temperature, breaks=5, col="blue")
```

3.2 Boîte à Moustaches

```
boxplot(elev,
        horizontal=T,
        xlab = "Taille (cm)",
        main = "Taille de 40 élèves")
rug(elev, 0.1)
```

- **Fonctions :**
 - `boxplot()` : Créer une boîte à moustaches
 - `rug()` : Ajouter les points individuels
- **Paramètres :**
 - `horizontal=T` : Orientation horizontale

4. Analyses Conditionnelles

4.1 Moyenne Conditionnelle : `ave()`

```
ave(score, genre)
```

- **Fonction :** `ave()`
- **Objectif :** Calculer des moyennes par groupe
- **Exemple :**

```
notes_par_classe = ave(notes, classe)
```

4.2 Analyse Agrégée : `aggregate()`

```
aggregate(data[, 3:4], list(data$Name), mean)
aggregate(. ~ Name, data[-2], mean)
```

- **Fonction :** `aggregate()`
- **Variations :**

- Moyenne par groupe
- Calculs conditionnels (variance, somme, etc.)
- **Exemples :**

```
# Moyenne par groupe
aggregate(revenus ~ secteur, data, mean)

# Variance par groupe
aggregate(. ~ categorie, data, var)
```

5. Autres Fonctions Utiles

5.1 Conversion de Type

```
as.numeric() # Convertir en numérique
```

5.2 Arrondissement

```
round(valeur, digits = 2) # Arrondir à 2 décimales
```

5.3 Fréquences Relatives

```
prop.table() # Calculer les fréquences
```

5.4 Configurations Graphiques

```
par() # Paramètres graphiques (similaire à subplot)
```

V2_CHAT

1. Calculs statistiques simples

- **Commande : `mean()`**

- **Description :** Calcule la moyenne des éléments d'un vecteur.
- **Syntaxe :** `mean(vecteur)`
- **Exemple :**

```
R
Copy code
elev <- c(138, 164, 150, 132, ...)
mean(elev) # Résultat : 146.8
```

- **Commande : `weighted.mean()`**

- **Description :** Calcule la moyenne pondérée.
- **Syntaxe :** `weighted.mean(valeurs, poids)`
- **Exemple :**

```
R
Copy code
moy5 <- weighted.mean(avec5$mids, avec5$counts)
moy5 # Résultat : 146.93
```

2. Regroupement et histogrammes

- **Commande : `hist()`**

- **Description :** Trace un histogramme des données.
- **Syntaxe :**

```
R
Copy code
hist(vecteur, breaks = ..., col = ..., labels = ..., las = ...,
     xlab = ..., ylab = ..., main = ...)
```

- **Exemple :**

```
R
Copy code
hist(elev, breaks = seq(from = min(elev), to = max(elev),
length = 6), col = grey(0.8), labels = TRUE, las = 1, main =
"Avec 5 classes")
```

3. Boîte à moustaches

- **Commande : `boxplot()`**

- **Description :** Affiche une boîte à moustaches.
- **Syntaxe :** `boxplot(vecteur, horizontal = TRUE, xlab = ..., main = ...)`
- **Exemple :**

```
R
Copy code
boxplot(elev, horizontal = TRUE, xlab = "Taille (cm)", main =
"Taille de 40 élèves")
rug(elev, 0.1) # Ajoute des marques sur l'axe
```

4. Moyenne conditionnée

- **Commande : `ave()`**

- **Description :** Calcule une moyenne conditionnée.
- **Syntaxe :** `ave(vecteur, condition)`
- **Exemple :**

```
R
Copy code
data <- data.frame(genre = c(rep("homme", 4), rep("femme", 5)),
score = c(1:4, 3:7))
data$MoyCondi <- ave(data$score, data$genre)
```

- **Commande : `aggregate()`**

- **Description :** Calcule des statistiques conditionnées par groupe.
- **Syntaxe :** `aggregate(. ~ variable, data, fonction)`
- **Exemple :**

```
R
Copy code
aggregate(data[, 3:4], list(data$Name), mean) # Moyenne par
groupe
aggregate(. ~ Name, data, sum) # Somme par groupe
```

5. Fonctions complémentaires

- **`prop.table()`**

- **Description :** Calcule directement les fréquences relatives.
- **Exemple :**

```
R
Copy code
prop.table(table(elev)) # Renvoie les fréquences des tailles
```

- **`round()`**

- **Description :** Arrondit les valeurs à un nombre spécifié de décimales.
- **Exemple :**

```
R
Copy code
round(3.14159, 2) # Résultat : 3.14
```

- **`par()`**

- **Description :** Configure la mise en page pour les graphiques (similaire à `subplot` en MATLAB).
- **Exemple :**

```
R
Copy code
par(mfrow = c(1, 2)) # Affiche deux graphiques côte à côte
```

6. Visualisation et analyse avancée avec *by*

- **Commande : `by()`**
 - **Description :** Applique une fonction sur des sous-ensembles définis par un facteur.
 - **Syntaxe :** `by(vecteur, facteur, fonction)`
 - **Exemple :**

```
R
Copy code
by(data$Taux1, data$Name, mean) # Moyenne par groupe
```

7. Extensions possibles avec *aggregate()*

- **Variance conditionnée :**

```
R
Copy code
aggregate(. ~ Name, data, var)
```

- **Écart-type conditionné :**

```
R
Copy code
aggregate(. ~ Name, data, sd)
```
