

## TP 3. Data frames et statistiques univariées quantitatives

### Objectifs du TP :

- découvrir la classe **data.frame** ;
- réaliser des graphiques simples ;
- soigner la présentation de graphiques ;
- déterminer les paramètres d'une série statistique.

### Exercice 1 :

La création d'un **data frame** peut s'effectuer en combinant des vecteurs.

1. Créer le **data frame** concernant 4 individus contenant leur **sexe**, leur **taille** en *cm* et leur **poids** en *pounds* avec les commandes suivantes :

```
> s <- c("M", "F", "F", "F")
> t <- c(182, 165, 159, 171)
> p <- c(164, 115, 140, 147)
> df <- data.frame(sexe = s, taille = t, poids = p)
> fix(df)
```

2. Taper les commandes suivantes :

```
> df
> str(df)
> dim(df)
> nrow(df)
> ncol(df)
> colnames(df)
> df[2:3]
> df$sexe
```

Que remarquer vous ?

### Exercice 2 :

On va travailler sur l'objet **InsectSprays** déjà existant dans **R**. Il contient une variable **quantitative discrète** que nous allons étudier.

1. Avant de manipuler cet objet, observons différentes façons d'appeler l'aide et utilisons l'aide pour comprendre le contenu de l'objet **InsectSprays**.

```
> help(insectspray)
> ??insectspray
> help (InsectSprays)
> ?InsectSprays
```

## 2. Observer l'objet `InsectSprays` avec les commandes suivantes:

```
> df <- InsectSprays
> str(df)
> head(df)
```

Quelle est l'utilité de la commande `head` ?

## 3. Effectuer les commandes suivantes :

```
> count
> df$count
```

## 4. Tableaux des effectifs, des fréquences et des fréquences cumulées de la variable quantitative discrète `count`.

- Réaliser le tableau des effectifs de la variable `count` avec la commande suivante.

```
> table(df$count)
```

- Réaliser le tableau des fréquences en consultant l'aide de la fonction `prop.table`. Pour plus de lisibilité, arrondir à 3 chiffres après la virgule.
- Construire le tableau des fréquences cumulées.

## 5. Représentations graphiques.

- Réaliser un premier graphique grâce à la commande suivante.

```
> plot(df$count)
```

Essayer la commande suivante. Que permet l'argument `pch` ?

```
> plot(df$count, pch = 4)
```

- Représenter graphiquement les effectifs par modalité avec la commande suivante.

```
> plot(table(df$count), main = "Effectifs", xlab = "Nombre d'insectes sur
chaque parcelle", ylab = "Nombre de parcelles")
```

- Représenter la courbe des fréquences cumulées. Utiliser l'argument `type` afin d'obtenir le graphique approprié. Faire attention aux **abscisses**.

- On peut ajouter des droites et du texte sur un graphique. Essayer les commandes suivantes.

```
> plot(df$count, pch = 4, main = "Nombre d'insectes sur chaque
parcette", xlab = "Numéro de parcette", ylab = "Effectif")
> abline(h = 10, col = "red")
> text(x = 30, y = 10, "seuil")
```

- On peut tracer plusieurs droites d'un coup avec la commande `abline`.

### Exercice 3 :

Dans cet exercice, nous allons travailler avec le tableau de données `trees` concernant des cerisiers noirs. Il contient les informations suivantes : **Girth (circonférence)**, **Height (hauteur)** et **Volume**. Les unités sont anglo-saxonnes, mais cela n'a aucune incidence pour la suite.

1. Commencer par affecter `trees` dans un nouvel objet nommé `data` et observer son contenu à l'aide de commandes déjà rencontrées.
2. Effectuer les commandes ci-dessous et les décrire en commentaire.

```
> summary(data)
> mean(data)
> mean(data$Height)
> quantile(data$Height,0.25)
> IQR(data$Girth)
```

3. Trouver une commande permettant d'obtenir d'un coup tous les quartiles, puis une autre permettant d'obtenir tous les déciles.
4. Réaliser un histogramme avec la commande suivante.

```
> hist(data$Height)
```

Mettre un titre principal et modifier les titres des axes.

5. Observer les différentes informations contenues dans l'objet `Histo`.

```
> Histo<-hist( data$Height, plot=FALSE)
> Histo
```

Que contiennent `Histo$breaks` et `Histo$counts` ?

6. Effectuer les commandes ci-dessous et ajouter le bon titre pour l'axe des ordonnées.

```
> bornes<-c(60,70,75,80,90)
> hist(data$Height,main="Répartition des arbres \n en fonction de leur hauteur",
xlab="Hauteur",ylab="",breaks=bornes)
```

### Exercice 4:

Enregistrer chaque commande utilisée dans un script nommé `VotreNom_TP2-CR.R`. Les données sont situées dans la librairie `datasets`, dans le `dataframe` `iris`. Ce `data`

*frame* contient des observations concernant une population d'iris (fleurs) pour lesquelles on a répertorié :

- **Species** : l'espèce de l'iris ;
  - **Sepal.Length** : la longueur des sépales en centimètres ;
  - **Sepal.Width** : la largeur des sépales en centimètres ;
  - **Petal.Length** : la longueur des pétales en centimètres ;
  - **Petal.Width** : la largeur des pétales en centimètres.
1. Effectuer la commande `library(datasets)` pour charger le package et affecter les données `iris` dans un **data frame** intitulé `data`.
  2. Calculer la largeur moyenne des pétales en millimètres arrondi à l'unité.
  3. Donner le tableau des effectifs de la largeur des sépales (en centimètre).
  4. Représenter graphiquement et de façon soignée et adaptée la courbe des fréquences cumulées de la largeur des sépales.
  5. Ajouter sur le graphique précédent des droites horizontales au niveau des proportions : **0.25**, **0.5** et **0.75** ; ainsi que des droites verticales au niveau des quantiles en **0.25**, **0.5** et **0.75**.
  6. Représenter le nuage des points de la longueur des pétales en fonction de la longueur des sépales.
  7. Effectuer un histogramme de la répartition de longueur des pétales soigné (titres, couleurs).