

Statistique Descriptive

Kossi Tonyi Wobubey ABOTSI

default

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.1      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Importation des données :

```
# install.packages("readxl")

# Load the readxl package

# Read data from the Excel file
data <- read_excel("données pour R_28 classes_ MEFG_22 24.xlsx")
#Selection des colonnes utile
data = data %>% dplyr::select(college:classe, `taille_cm`: gender, sb:pmvpa, time, CA:`CSP_P2`)

#Renommage des colonnes
colnames(data)[23:24] = c("CSP_père", "CSP_mère")

#Caste
data$weight_kg = as.double(data$weight_kg)
data$taille_cm = as.double(data$taille_cm)
data$age = as.integer(data$age)
data$time = as.double(data$time)

#Ajout de colonne des IMC
data$IMC_kg_m2 <- data$weight_kg / (data$taille_cm * 10^-2)^2

# Ajout d'une nouvelle colonne "IPS_categorie"
data$IPS_categorie <- ifelse(data$IPS < 89, "Faible",
                             ifelse(data$IPS >= 90 & data$IPS <= 114, "Moyenne", "Élevée"))

#Résolution du problème de facteur double
data$gender = ifelse(data$gender != 'M' & data$gender != 'F', 'M', data$gender)
```

```

#Suppression des valeurs manquantes
data = na.omit(data)
#data = data %>%
# filter(data$CSP_mère != 'NA' & data$CSP_père != 'NA')

# Print the first few rows of the data to verify
head(data)

## # A tibble: 6 x 26
##   college classe taille_cm weight_kg   age gender    sb    lpa    mpa    vpa psb
##   <chr>   <chr>     <dbl>     <dbl> <int> <chr>  <dbl> <dbl> <dbl> <dbl> <chr>
## 1 aigle   3.P           157        55    15 F    26.3  4.83  21.2  2.33 47,88~
## 2 aigle   3.P           178        61    14 M     14   8.33  28    4    25,45~
## 3 aigle   3.P           170        75    15 M    20.3  7.33  21.8  5    36,97~
## 4 aigle   3.P           153        68    15 F    26.2  7.33  18.7  2.83 47,58~
## 5 aigle   3.P           181        95    15 M    12.2 12.3  22.3  6.17 22,12~
## 6 aigle   3.P           164        51    15 F    20.5  6.5   20.3  4.83 37,27~
## # i 15 more variables: pla <chr>, pmpa <chr>, pvpa <chr>, mvpa <dbl>,
## #   pmvpa <chr>, time <dbl>, CA <dbl>, activites <chr>, IPS <dbl>,
## #   cat_IPS <chr>, Geographie <chr>, CSP_père <chr>, CSP_mère <chr>,
## #   IMC_kg_m2 <dbl>, IPS_categorie <chr>

```

Statistique descriptive de la Population

Calculons l'âge moyens des filles et garçons et l'âge moyen des participants.

1. Age moyen des Participants

```

age_sexe_data=data %>%
  group_by(gender) %>%
  summarise(age_total = sum(age),effectif = n())

age_moyen = sum(age_sexe_data$age_total)/sum(age_sexe_data$effectif)

age_moyen

## [1] 13.65499

```

Donc l'âge moyen des participants est **13.65**.

2. L'âge moyen des filles et garçons

```

age_sexe_data$age_moyen = age_sexe_data$age_total/age_sexe_data$effectif

age_sexe_data

```

```

## # A tibble: 2 x 4
##   gender age_total effectif age_moyen
##   <chr>     <int>     <int>     <dbl>
## 1 F         2418       177       13.7
## 2 M         2648       194       13.6

```

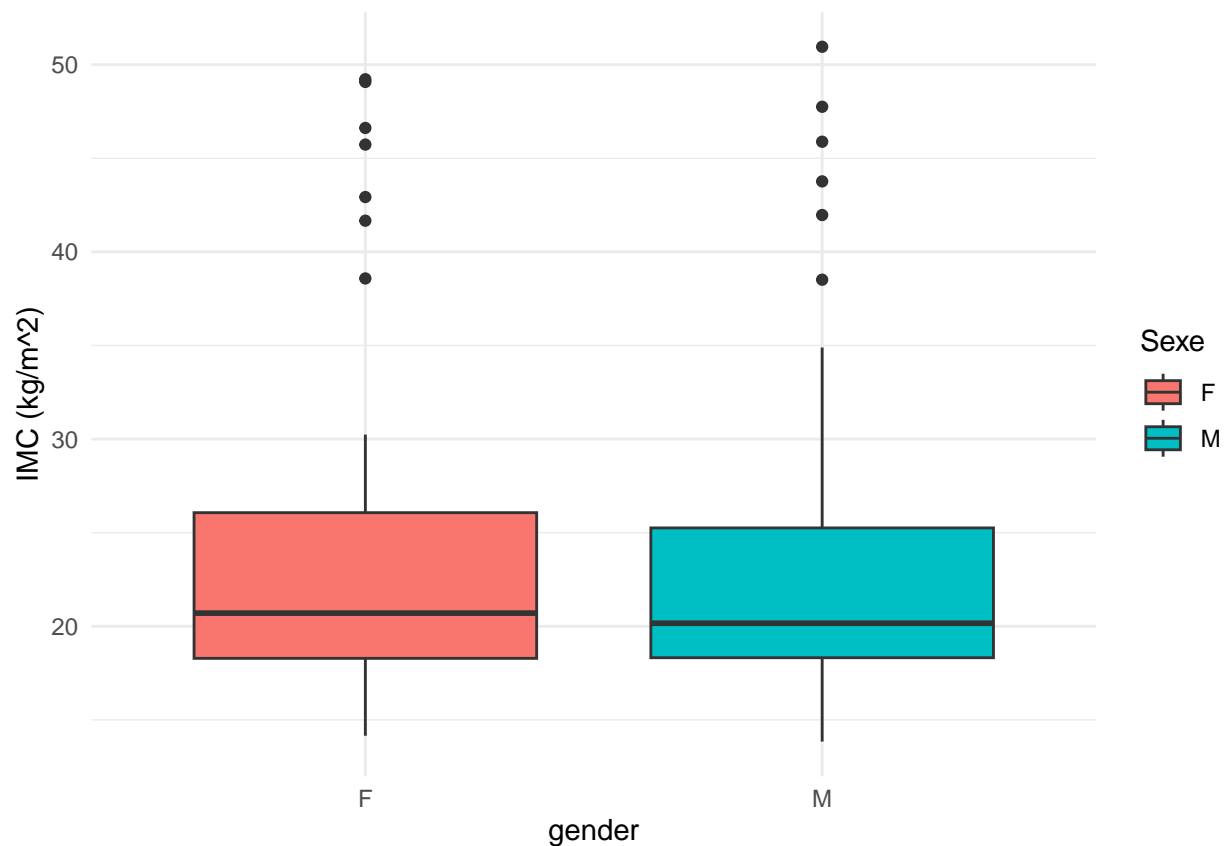
Sexe	F	M	Participant(les deux sexes)
Age Moyen	13.66	13.65	13.65

- IMC moyen selon le sexe

```
data %>%
  group_by(gender) %>%
  summarise(IMC_moyen = mean(na.omit(IMC_kg_m2)))
```

```
## # A tibble: 2 x 2
##   gender IMC_moyen
##   <chr>     <dbl>
## 1 F         22.4
## 2 M         22.2
```

```
# Créer le diagramme en boîte pour l'IMC par classe et sexe
ggplot(data, aes(x = gender, y = IMC_kg_m2, fill = gender)) +
  geom_boxplot() +
  labs(x = "gender", y = "IMC (kg/m^2)", fill = "Sexe") +
  theme_minimal()
```



En moyenne l'IMC des filles est légèrement plus grand que celui des garçons.

```
mean(data$IMC_kg_m2)
```

```
## [1] 22.29897
```

En général la moyenne des IMC est de **22.3**.

Récapitulatif dans le tableau suivant :

	Population globale	Filles	Garçons
IMC	22.3	22.45	22.16

- CSP des parents

```
data_1 <- as.data.frame(table(data$CSP_père, data$gender))
# Renommer les colonnes
names(data_1) <- c("CSP", "sexe", "Effectif_Participant")

data_2 = as.data.frame(table(data$CSP_mère, data$gender))
# Renommer les colonnes
names(data_2) = c("CSP", "sexe", "Effectif_Participant")

CSP_data <- rbind(data_1, data_2)

# Renommer la première colonne
colnames(CSP_data)[1] <- "CSP_Parent"

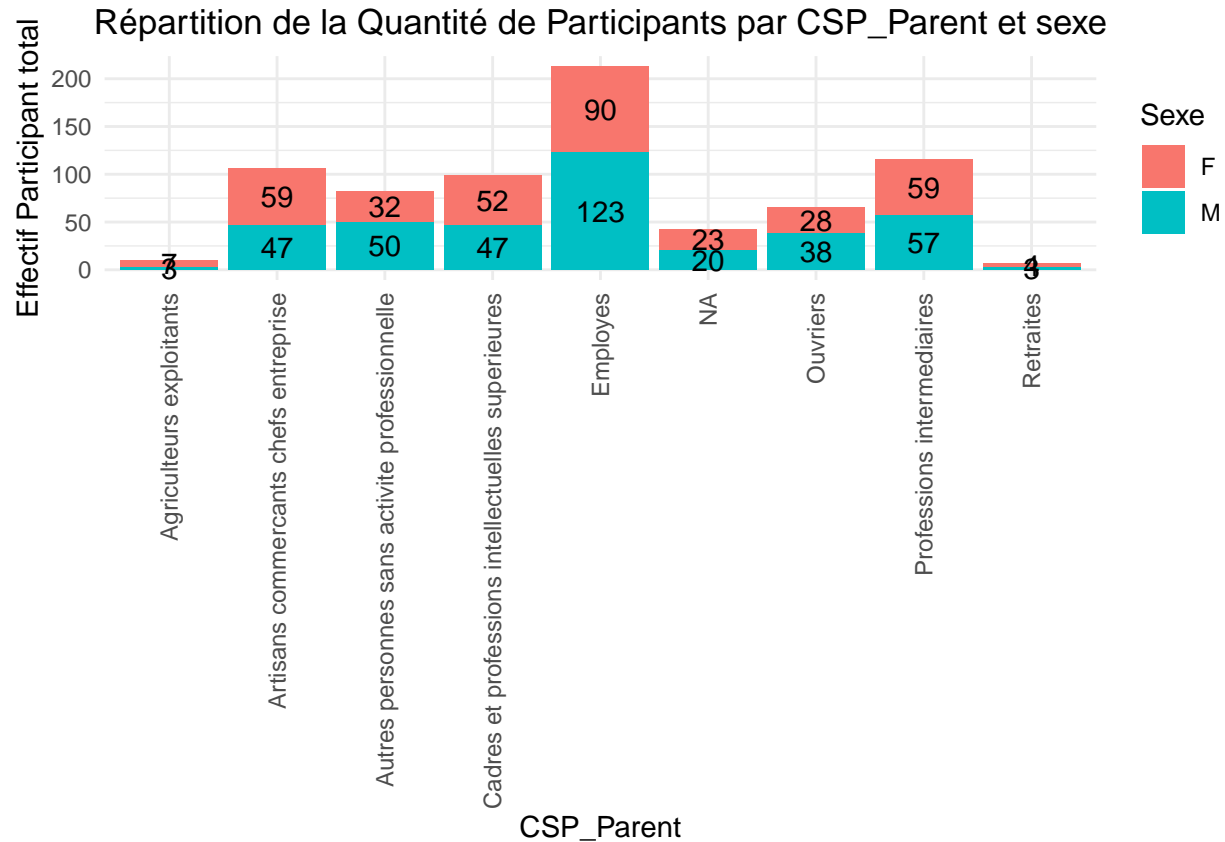
# Grouper par CSP_Parent et sexe, puis calculer les totaux
CSP_data <- CSP_data %>%
  group_by(CSP_Parent, sexe) %>%
  summarise(Effectif_Participant_total = sum(Effectif_Participant)) %>%
  ungroup()

# Afficher le dataframe
print(CSP_data)
```

```
## # A tibble: 18 x 3
##   CSP_Parent      sexe Effectif_Participant_total
##   <fct>          <fct>               <int>
## 1 Agriculteurs exploitants F               7
## 2 Agriculteurs exploitants M               3
## 3 Artisans commerçants chefs entreprise F            59
## 4 Artisans commerçants chefs entreprise M            47
## 5 Autres personnes sans activité professionnelle F            32
## 6 Autres personnes sans activité professionnelle M            50
## 7 Cadres et professions intellectuelles supérieures F            52
## 8 Cadres et professions intellectuelles supérieures M            47
## 9 Employés F            90
## 10 Employés M           123
## 11 NA F            23
## 12 NA M            20
## 13 Ouvriers F            28
## 14 Ouvriers M            38
## 15 Professions intermédiaires F            59
## 16 Professions intermédiaires M            57
## 17 Retraites F             4
## 18 Retraites M             3
## # i abbreviated name: 1: Effectif_Participant_total
```

```
# Créer le diagramme en barres empilées avec les modalités en abscisses affichées verticalement
ggplot(CSP_data, aes(x = CSP_Parent, y = Effectif_Participant_total, fill = sexe)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = Effectif_Participant_total), position = position_stack(vjust = 0.5), color = "black")
labs(x = "CSP_Parent", y = "Effectif Participant total", fill = "Sexe", title = "Répartition de la Qu")
theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5),
```

```
axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)
) # Modifier l'angle des modalités sur l'axe x
```



Voici le tableau des effectifs des participants dont l'un de ses parents au moins exerce une des fonctions ci dessous :

	Filles	Garçons	Population Globale
Agriculteurs exploitants	7	3	10
Artisans commerçants chefs entreprise	59	47	106
Autres personnes sans activité professionnelle	32	50	82
Cadres et professions intellectuelles supérieures	52	47	99
Employés	90	123	213
Ouvriers	28	38	66
Professions intermédiaires	59	57	116
Retraites	4	3	7
NA	23	20	43

Statistique descriptive pour le lieu d'étude :

- Proportion de l'échantillon global de la population selon le CA

```
prop.table(table(data$CA))*100
```

```
##
##      1      2      3      4
## 12.12938 26.14555 10.24259 51.48248
```

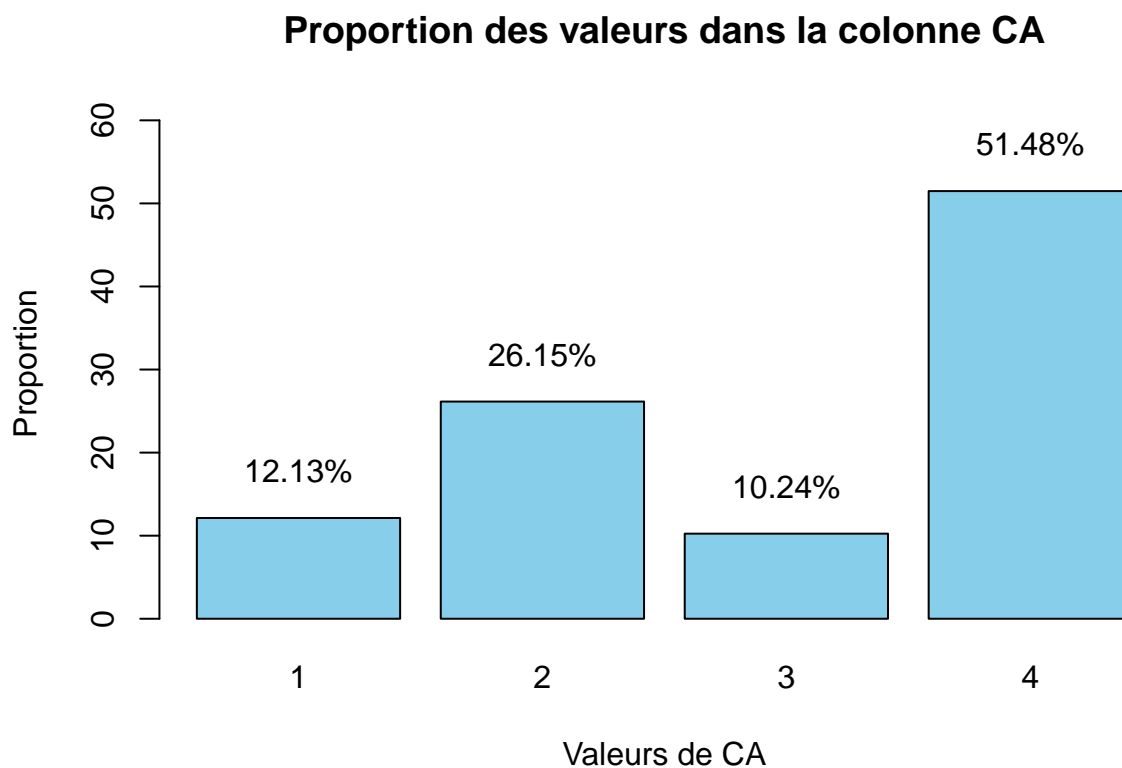
CA	1	2	3	4
Proportion(%)	12.13	26.15	10.24	51.48

Illustration avec un barplot :

```
# Calculer les proportions
proportions <- prop.table(table(data$CA)) * 100

# Créer le barplot
barplot(proportions, main = "Proportion des valeurs dans la colonne CA",
        xlab = "Valeurs de CA", ylab = "Proportion", col = "skyblue",
        ylim=c(0, max(proportions) + 10)) # Ajuster ylim pour éviter le chevauchement du texte

# Ajouter les proportions sur les barres
text(x = barplot(proportions, plot = FALSE), # Obtenir les positions en x des barres
     y = proportions+2, # Décaler légèrement le texte au-dessus des barres
     labels = sprintf("%.2f%", proportions), # Formater les proportions avec deux décimales
     pos = 3) # Positionner le texte au-dessus des barres (3 = au-dessus)
```



- Proportion de l'échantillon global de la population selon l'IPS
- IPS faible inférieur à 89
- IPS moyenne entre 90 et 114
- IPS élevé supérieur à 115

```
prop.table(table(data$IPS_categorie))*100
```

```
##
##   Élevée   Faible   Moyenne
## 30.99730 22.37197 46.63073
```

IPS	Faible	Moyenne	Elevé
Proportion(%)	31	22.37	46.63

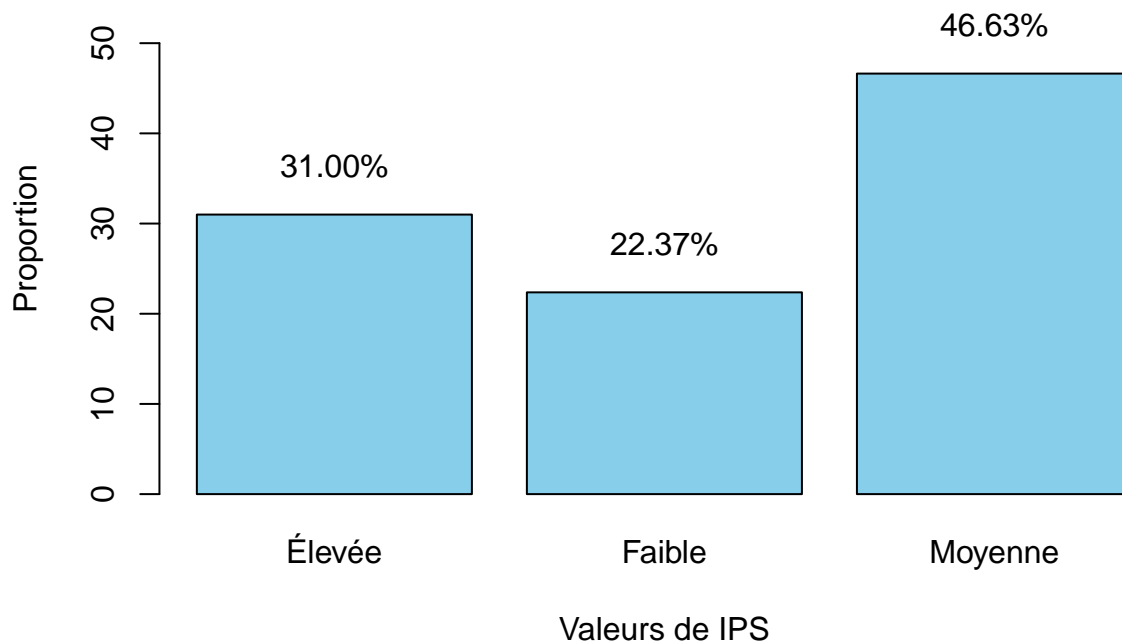
Illustration avec un barplot

```
# Calculer les proportions pour la colonne IPS_categorie
proportions_ips = prop.table(table(data$IPS_categorie)) * 100

# Créer le barplot pour IPS_categorie
bp <- barplot(proportions_ips, main = "Proportion des valeurs dans la colonne IPS",
              xlab = "Valeurs de IPS", ylab = "Proportion", col = "skyblue",
              ylim = c(0, max(proportions_ips) + 10)) # Ajuster ylim pour éviter le chevauchement du t

# Ajouter les proportions sur les barres
text(x = bp, # Positions en x des barres, retournées par barplot
     y = proportions_ips + 2, # Ajouter un petit espace au-dessus de chaque barre pour le texte
     labels = sprintf("%.2f%%", proportions_ips), # Formater les proportions avec deux décimales
     pos = 3) # Poser le texte au-dessus des barres
```

Proportion des valeurs dans la colonne IPS



- Proportion de l'échantillon global de la population selon le milieu géographique

```
prop.table(table(data$Géographie))*100
```

```
##  
##      rural      urbain  
## 35.84906 64.15094
```

Milieu géographique	urbain	rural
Proportion(%)	64.15	35.85

Illustration avec un barplot

```
# Calculer les proportions pour la colonne Géographie  
proportions_geo = prop.table(table(data$Géographie)) * 100  
  
# Créer le barplot pour Géographie  
bp_geo <- barplot(proportions_geo, main = "Proportion des valeurs dans la colonne Géographie",  
                  xlab = "Milieu géographique", ylab = "Proportion", col = "skyblue",  
                  ylim = c(0, max(proportions_geo) + 10)) # Ajuster ylim pour éviter le chevauchement  
  
# Ajouter les proportions sur les barres  
text(x = bp_geo, # Positions en x des barres, retournées par barplot  
     y = proportions_geo + 0.25, # Ajouter un petit espace au-dessus de chaque barre pour le texte  
     labels = sprintf("%.2f%", proportions_geo), # Formater les proportions avec deux décimales  
     pos = 3) # Poser le texte au-dessus des barres
```

Proportion des valeurs dans la colonne Géographie

