Séance 4_Annexe: Mesures de tendance centrale

Comment les analyses sont faites

Visseho Adjiwanou, PhD.

31 January 2023

Introduction

Ceci est un fichier .RMD que vous pouvez manipuler à votre guise. C'est lui que j'ai utilisé pour l'analyse des données de la séance 4. Dans cette séance, nous avons les données sur les revenus de 15 hommes et de 16 femmes. Ce qui veut dire que nous avons deux variables qui sont le sexe (homme, femme), une variable nominale et le revenu (ratio). Il est toujours important aussi d'avoir dans une base de données les identifiants des individus; donc, une variable allant de 1 à 31 (dans notre cas).

Pour entrer ces données dans R, c'est simple, il faut donc créer trois variables ou objets. Voici comment cela se fait:

Création des variables

- **Identifiant**

```
ident <- c(1:31)
```

- **iden** veut dire que c'est le nom de ma variable. C'est un contenant.
- **<-** est ce qu'on appelle signe d'affectation. C'est juste pour dire que je mets quelques chose dans ident.
- c(), vous en avez besoin toujours pour dire les choses que vous voulez mettre dans le contenant.
- 1:31 veut dire de 1 à 31. On aurait pu écrire c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 20, 31). Cela donnerait le même résultat. Voyez comme c'est élégant ce que j'ai écrit et comment cela vous prend moins de temps. Vous devez chercher l'efficacité quand vous "programmez".

Si vous cliquer sur la flèche verte de la ligne 17, il va vous dire ce que vous venez de faire. La commande en bas donne le même résultat. Il vous affiche le résultat.

```
ident
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30 31
```

- Question: A votre tour, ajouter deux individus à cet identifiant.

Maintenant que vous savez faire, on va créer les deux autres variables:

- **Sexe**

Il y a 15 hommes et 16 femmes. On peut donc faire comme précédemment. Mais, ici, on va répéter Homme 15 fois et Femme 16 fois. On va alors utiliser la fonction : - **rep()** qui va répéter l'information pour nous.

```
sexe <- c(rep("Homme", 15), rep("Femme", 16))
sexe
```

```
##  [1] "Homme" "Homme" "Homme" "Homme" "Homme" "Homme" "Homme" "Homme" "Homme"
## [10] "Homme" "Homme" "Homme" "Homme" "Homme" "Homme" "Femme" "Femme" "Femme"
## [19] "Femme" "Femme" "Femme" "Femme" "Femme" "Femme" "Femme" "Femme" "Femme"
## [28] "Femme" "Femme" "Femme" "Femme"
```

- Question: Ajouter un homme et une femme à sexe
- **Revenu**

```
revenu_homme <- c(2, 2.5, 1.7, 3, 5, 4.1, 8.1, 5.2, 3.1, 1.4, 7.1, 6.0, 3.3, 4.3, 6.1)

revenu_homme
```

```
##  [1] 2.0 2.5 1.7 3.0 5.0 4.1 8.1 5.2 3.1 1.4 7.1 6.0 3.3 4.3 6.1
```

- Question: Ajouter une nouvelle valeur de 2.5$ à revenu_homme

Et maintenant les revenus des femmes

```
revenu_femme <- c(3.1, 2.7, 1.2, 4.2, 5.5, 4.3, 2.0, 1.5, 0.5, 1.3, 2.9, 2.7, 5.1, 3.0, 6.3, 4.2)

revenu_femme
```

```
##  [1] 3.1 2.7 1.2 4.2 5.5 4.3 2.0 1.5 0.5 1.3 2.9 2.7 5.1 3.0 6.3 4.2
```

- Question: Ajouter une nouvelle valeur de 4.2$ à revenu_femme. Les revenus sont en millier.

Création de la base de données.

Une fois qu'on a créé les variables, il faut les mettre ensemble pour créer une base de données. cette fois-ci, on ne va plus utiliser le **c()** pour les mettre ensemble mais une fonction qui s'appelle data.frame.

```
donnee_revenu <- data.frame(ident, sexe, revenu = c(revenu_homme, revenu_femme))
```

- Question: votre base de données contient combien d'individus?

Vous pouvez alors visualiser votre base de données en cliquant dessus dans le panneau Environnement à droite.

Fréquences

Pour voir les fréquences de vos variables, on va utiliser des outils/packages appropriés à cela. Il existe plusieurs développeurs de ces outils. Nous allons apprendre dans le cadre de ce cours deux outils/packages un outils qui s'appelle summarytools et tidyverse. Pour ce faire, vous devez les installer sur votre machine en exécutant kes commandes suivantes:

```
#install.packages("tidyverse")
#install.packages("summarytools")
```

Vous pouvez ainsi avec la fonction **install.packages()**, installer d'autres outils. Vous devez comme tout package, les installer une seule fois. Après avoir éxécuter ces commandes, vous ne devez plus jamais les éxécuter.

Cependant, pour les utiliser, vous devez toujours les appeler. On dit les charger en éxécutant ces deux lignes de commandes:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.1
## v readr   2.1.2     v forcats 0.5.2

## Warning: package 'tibble' was built under R version 3.6.2

## Warning: package 'tidyr' was built under R version 3.6.2

## Warning: package 'readr' was built under R version 3.6.2

## Warning: package 'purrr' was built under R version 3.6.2

## Warning: package 'dplyr' was built under R version 3.6.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(summarytools)
```

```
## Warning in system2("/usr/bin/otool", c("-L", shQuote(DSO)), stdout = TRUE):
## running command ''/usr/bin/otool' -L '/Library/Frameworks/R.framework/Resources/
## library/tcltk/libs//tcltk.so'' had status 1

## For best results, restart R session and update pander using devtools:: or remotes::install_github('ra
##
## Attaching package: 'summarytools'
##
## The following object is masked from 'package:tibble':
##
##     view
```

Vous comprenez ici que pour utiliser un outil, vous devez le charger avec la fonction **library()**.

- Question: trouver un autre package en ligne qui permet de calculer les fréquences et installer-le sur votre machine.

Maintenant que tout cela est fait, on peut alors commencer par les utiliser.

Fréquences des variables

Pour dresser le tableau de fréquences, on va utiliser la fonction **freq()** de summarytools.

```
freq(donnee_revenu$sexe)
```

```
## Frequencies
## donnee_revenu$sexe
## Type: Factor
##
##               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## ----------- ------ --------- -------------- --------- --------------
##       Femme     16     51.61          51.61     51.61          51.61
##       Homme     15     48.39         100.00     48.39         100.00
##        <NA>      0                               0.00         100.00
##       Total     31    100.00         100.00    100.00         100.00
```

- donnee_revenu est le nom de votre base de données
- sexe est la variable dont vous voulez calculer la fréquence

- $ est la fonction qui vous permet d'aller chercher cette variable dans la base de données. Ainsi, chaque fois que vous voulez faire quelque chose avec une variable, vous devez suivre ces trois étapes.
- Question: donner la distribution de la variable revenu et décrivez le résultat obtenu.

Paramètres de tendances centrales

- **Moyenne**

```
mean(donnee_revenu$revenu)
```

```
## [1] 3.658065
```

La fonction mean nous permet de calculer la moyenne.

- **Médiane**

```
median(donnee_revenu$revenu)
```

```
## [1] 3.1
```

- **Mode**

Il n'y a malheureusement pas de fonction pour calculer le mode. Vous avez vu que le tableau de fréquence vous donnait quand même cela. Alors, il y a d'autres utilisateurs qui ont crée une fonction pour pouvoir calculer le mode. C'est là où réside la force de R. Une de ces fonction est obtenu en exécutant la commande suivante:

```
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

Maintenant, vous pouvez calculer votre mode avec cette fonction en faisant:

```
getmode(donnee_revenu$revenu)
```

```
## [1] 2
```

- Premier quartile

```
quantile(donnee_revenu$revenu, probs = 0.25)
```

```
##  25%
## 2.25
```

- Question: quelle est la valeur du troisième quartile?

Vous voyez que tout cela nous est crée individuellement. On peut les mettre tous ensemble avec un outil puissant de tidyverse qui s'appelle **summarise**. Voici comment on l'utilise:

```
tendance1 <-
  donnee_revenu %>%
  summarise(minimum = min(revenu))
```

Voyez ce que vous venez de faire

```
tendance1
```

```
##   minimum
## 1     0.5
```

Maintenant, on peut ajouter les autres paramètres aisément en mettant une virgule. Remarquez où j'ai mis la virgule:

```
tendance2 <-
  donnee_revenu %>%
  summarise(minimum = min(revenu),
            maximum = max(revenu))

tendance2
```

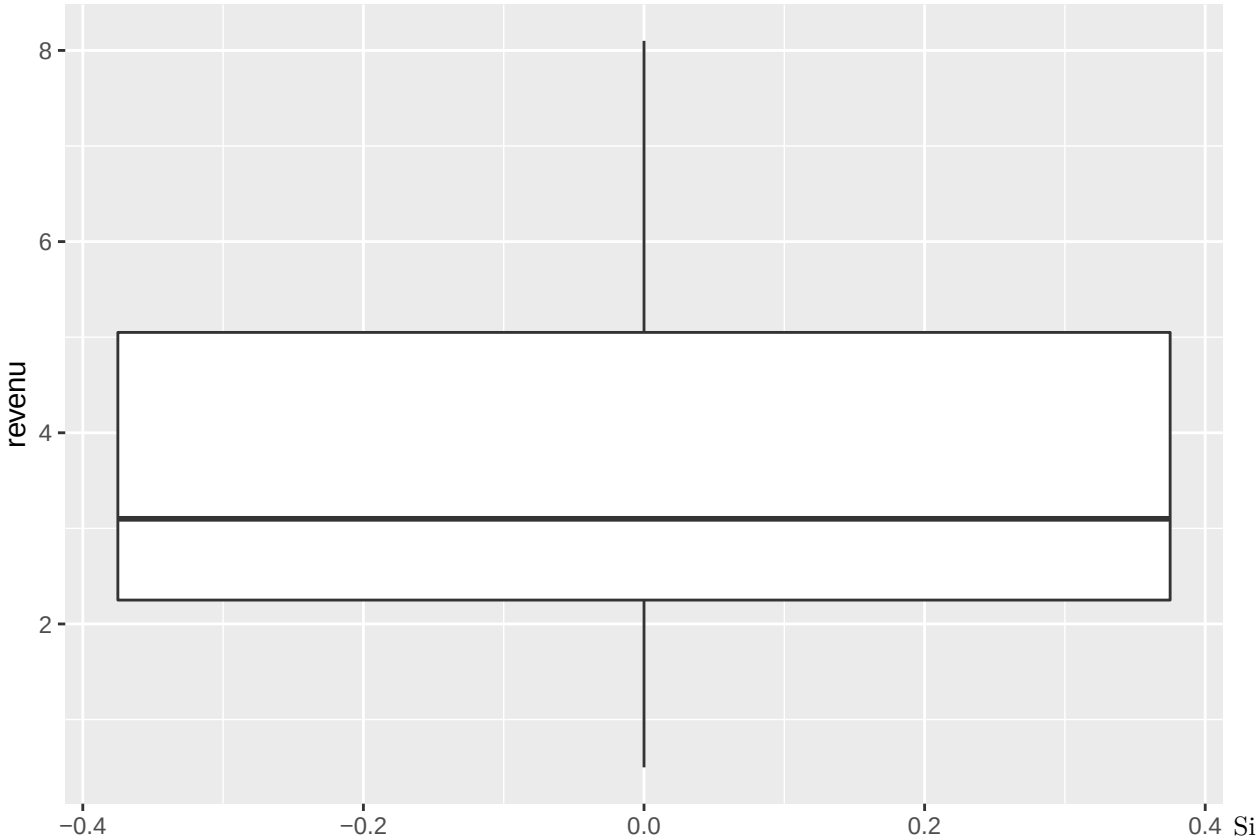
```
##   minimum maximum
## 1     0.5     8.1
```

- Question: A votre tour d'ajouter
  - la moyenne
  - la médiane
  - le mode
  - le premier quartile
  - le troisième quartile
  - le troisième décile

Visualisation

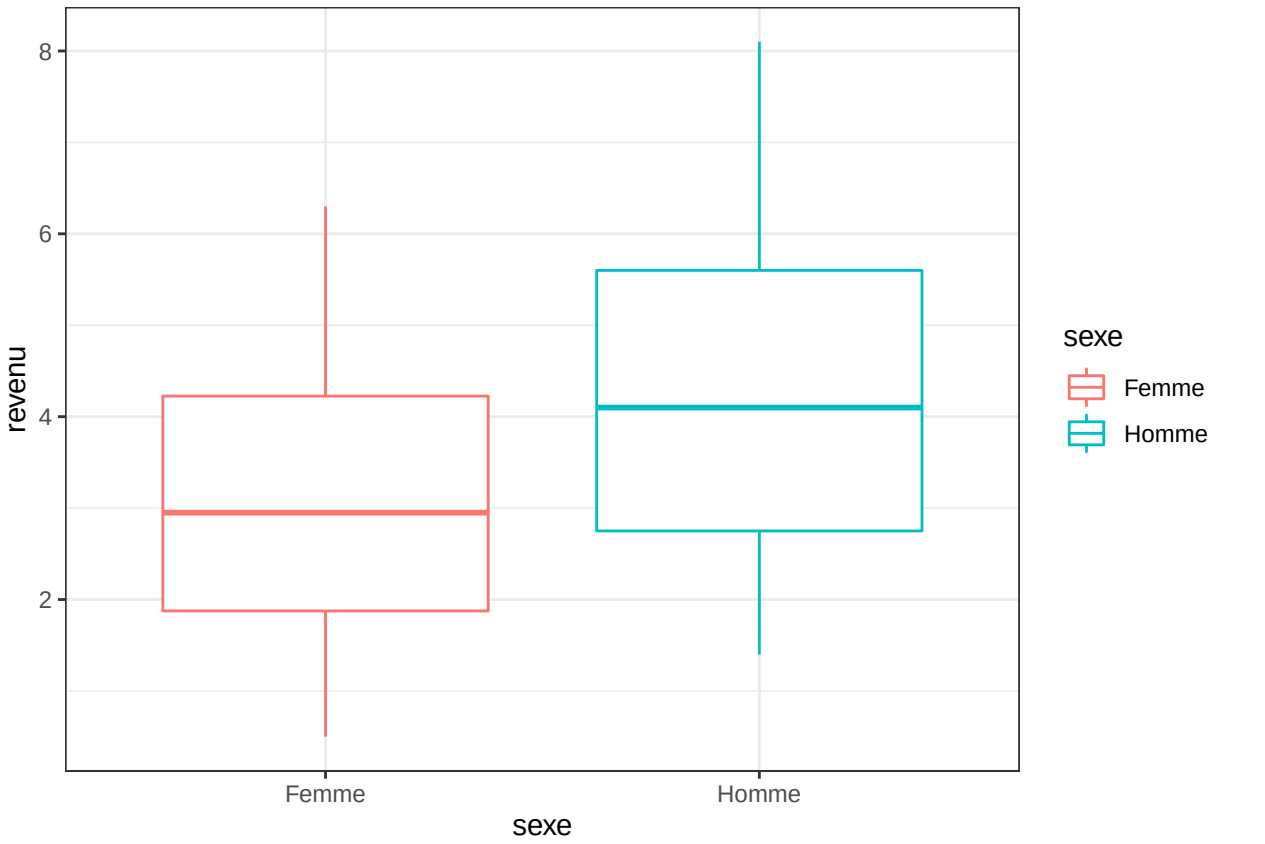
Je ne vais pas encore élaborer sur ceci, mais voyez comment on crée le graphique qu'on appelle boite de moustâche (boxplot ou diagramme de quartile).

```
ggplot(donnee_revenu) +
  geom_boxplot(aes(y = revenu))
```



Si on veut voir cela selon le sexe, on va éxécuter la commande suivante:

```r
ggplot(donnee_revenu) +
  geom_boxplot(aes(x = sexe, y = revenu, color = sexe)) +
  theme_bw()
```



Remarque importante: N'éxécuter pas juste les codes, essayer de les saisir vous-mêmes, vous verez comment cela se fait.

Solution aux questions que je vous ai posées tout au long du labo

Remarquez que quand je suis dans la zone de commandes (partie grisée, commençans par `{r} et finissant par`), je ne peux pas écrire du texte. Pour ce faire, je dois le préceder de # (qui signifie que c'est un commentaire).

```r
# Ajouter deux individues
ident <- c(1:33)

# Ajouter un homme et une femme
sexe <- c(rep("Homme", 16), rep("Femme", 17))

# Ajouter 2.5 à revenu_homme et 4.2 à revenu femme
revenu_homme <- c(2, 2.5, 1.7, 3, 5, 4.1, 8.1, 5.2, 3.1, 1.4, 7.1, 6.0, 3.3, 4.3, 6.1, 2.5)

revenu_femme <- c(3.1, 2.7, 1.2, 4.2, 5.5, 4.3, 2.0, 1.5, 0.5, 1.3, 2.9, 2.7, 5.1, 3.0, 6.3, 4.2, 4.2)

# Créer la base de données

donnee_revenu <- data.frame(ident, sexe, revenu = c(revenu_homme, revenu_femme))
```

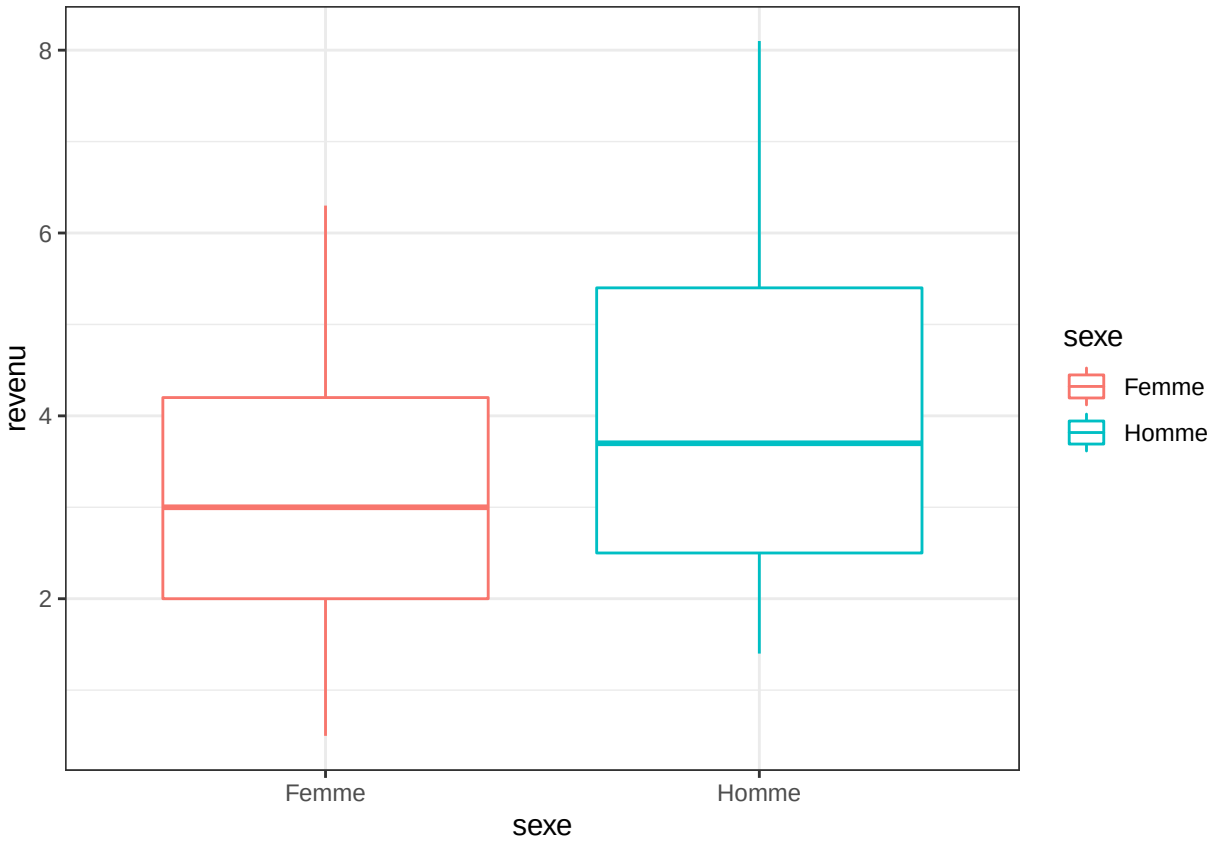
```r
# Paramètres de tendances centrales

getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}


par_tendance <-
  donnee_revenu %>%
  summarise(minimum = min(revenu),
            maximum = max(revenu),
            moyenne = mean(revenu),
            mediane = median(revenu),
            mode = getmode(revenu),
            q1 = quantile(revenu, probs = 0.25),
            q3 = quantile(revenu, probs = 0.75))

par_tendance
```

```
##   minimum maximum  moyenne mediane mode  q1 q3
## 1     0.5     8.1 3.639394     3.1  4.2 2.5  5
```

```r
# Cérise sur le gateau, je peux le faire par sexe (voir l'ajout de la ligne 305)

par_tendance_sexe <-
  donnee_revenu %>%
  group_by(sexe) %>%
  summarise(minimum = min(revenu),
            maximum = max(revenu),
            moyenne = mean(revenu),
            mediane = median(revenu),
            mode = getmode(revenu),
            q1 = quantile(revenu, probs = 0.25),
            q3 = quantile(revenu, probs = 0.75))

par_tendance_sexe
```

```
## # A tibble: 2 x 8
##   sexe  minimum maximum moyenne mediane  mode    q1    q3
##   <fct>   <dbl>   <dbl>   <dbl>   <dbl> <dbl> <dbl> <dbl>
## 1 Femme     0.5     6.3    3.22     3     4.2   2     4.2
## 2 Homme     1.4     8.1    4.09     3.7   2.5   2.5   5.4
```

```r
# Représentation

ggplot(donnee_revenu) +
  geom_boxplot(aes(x = sexe, y = revenu, color = sexe)) +
  theme_bw()
```



- Dernière questions
- Changer certaines valeurs du revenu en introduisant de grosses valeurs pour voir comment les paramètres changes. Que concluez-vous? Je peux vous assurer que si vous refaites tout ceci et que vous comprenez, vous avez acquis la moitié des notions du cours. Vous pouvez en apprendre davantage en allant ici:
  - https://oraprdnt.uqtr.uquebec.ca/pls/public/gscw031?owa_no_site=6738 https://juba.github.io/tidyverse/01-presentation.html

Donnez-moi des nouvelles.