Travaux Pratiques de Biométrie 3

Benoît Simon-Bouhet

2019-03-02

Table des matières

1 Préambule

Ce livre contient l'ensemble du matériel (contenus, exemples, exercices...) nécessaire à la réalisation des travaux pratiques et TEA de biométrie 3. Ces travaux pratiques ont un seul objectif principal : vous permettre de mettre en œuvre, dans `RStudio` les méthodes statistiques découvertes en cours magistral et en TD de biométrie 2 (au semestre précédent) et en biométrie 3 depuis début janvier.

Je considère qu'à ce stade, vous devez être à l'aise dans RStudio pour effectuer les tâches suivantes :

1. Importer des jeux de données dans RStudio
2. Manipuler des tableaux de données avec `tidyr` pour les mettre dans un format permettant les analyses statistiques et les représentations graphiques
3. Faire des graphiques exploratoires avec `ggplot2` pour visualiser des données
4. Filtrer des lignes, sélectionner des colonnes, trier, créer de nouvelles variables et calculer des résumés des données avec les fonction `filter()`, `select()`, `arrange()`, `mutate()`, `summarise()` et `group_by()` du package `dplyr`
5. Utiliser le pipe %>% afin d'enchaîner plusieurs commandes

6. Créer des scripts parlants contenant des commandes et des commentaires utiles
7. Spécifier/modifier votre répertoire de travail
8. Installer des packages additionnels.

Si vous pensez avoir besoin de rappels sur ces notions, je vous encourrage vivement à consulter le livre en ligne dédié aux travaux pratiques de Biométrie 2 pour vous rafraîchir la mémoire.

L'organisation des TP et TEA de biométrie 3 sera la suivante :

— Séance 1 : 1h30 de TP suivie d'une séance de 1h30 de TEA. Rappels concernant les statistiques descriptives et les visualisations graphiques utiles pour déméler la complexité de certains jeux de données. Comparaisons (paramétriques et non paramétriques) de la moyenne de 2 populations.
— Séance 2 : 1h30 de TP suivie d'une séance de 1h30 de TEA. Comparaisons (paramétriques et non paramétriques) la moyenne de plus de 2 populations : analyse de variance, hypothèses et conditions d'application.
— Séance 3 : 1h30 de TP suivie d'une séance de 1h30 de TEA. Étude de la liaison entre 2 variables. Corrélation (paramétrique et non paramétrique) et régression linéaire. Tests d'hypothèses, estimation et conditions d'application.
— Séance 4 : 1h30 de TP. Exercices d'application et corrections en guise de préparation pour l'examen.

2 Introduction

Sur votre disque dur, créez un dossier nommé "Biometrie3" (sans accent, sans espace). Au début de chaque nouvelle séance de TP, vous devrez ensuite effectuer les opérations suivantes :

1. Créez, dans votre dossier "Biometrie3", un sous-dossier nommé "TP_1", "TP_2", etc.
2. Téléchargez les fichiers utiles disponibles sur l'ENT et placez-les dans le dossier du TP correspondant.
3. Lancez RStudio.
4. Dans l'onglet "Files" de RStudio, naviguez jusqu'au sous-dossier "TP_X" que vous venez de créer et indiquez à RStudio qu'il s'agit de votre répertoire de travail. Si vous ne savez plus comment faire, consultez la section 2.2.2 du livre en ligne de Biométrie 2. Si votre répertoire de travail a été correctement spécifié, vous devriez constater qu'une commande ressemblant à ceci est apparue dans la console de RStudio :

```
setwd("C:/.../Biometrie3/TP_X")
```

5. Dans la console, tapez :

```
list.files()
```

La liste des fichiers contenus dans votre répertoire de travail (donc le nom des fichiers que vous avez téléchargé sur l'ENT) devrait apparaître dans la console. Si ce n'est pas le cas, recommencez depuis

le début. Vous pouvez également vérifier à tout moment si le répertoire de travail utilisé par RStudio est bien celui que vous pensez en tapant :

```
getwd()
```

6. Créez un nouveau script dans votre répertoire de travail et sauvegardez-le. Si vous ne savez plus comment faire, consultez la section 2.2.3 du livre en ligne de Biométrie 2.
7. Dans l'onglet "History" de RStudio, cliquez sur la commande commençant par `setwd()` puis cliquez sur le bouton "To source" (une flèche verte dirigée vers la gauche). Cela a pour effet de copier dans votre script la commande permettant de spécifier le répertoire de travail correct. Ainsi, lors de votre prochaine session de travail, vous n'aurez pas besoin de spécifier manuellement quel est votre répertoire de travail comme nous l'avons fait à l'étape 4 ci-dessus : il vous suffira d'ouvrir votre script et d'envoyer cette commande dans la console en pressant les touches `ctrl + Entrée`.
8. N'oubliez pas de sauvegarder votre script très régulièrement et d'y ajouter autant de commentaires que nécessaire avec le symbole #.

Si vous suivez rigoureusement ces étapes, vous devriez être dans la situation idéale pour commencer à travailler efficacement dans RStudio. Avec un minimum d'habitude, mettre tout ça en place ne devrait pas vous demander plus de 2 ou 3 minutes. À partir de maintenant, toutes vos analyses et commentaires doivent figurer dans vos scripts.

3 Séance 1 : statistiques descriptives et tests d'hypothèses

3.1 Packages et données

Pour chacune des 4 séances de travaux pratiques (et TEA) qui viennent, vous aurez besoin d'utiliser des packages spécifiques et d'importer des données depuis des fichiers externes disponibles sur l'ENT.

Les packages dont vous aurez besoin pour cette séance, et que vous devez donc charger en mémoire, sont les suivants :

```
library(tidyverse)
library(readr)
library(skimr)
```

Si ces commandes (que vous devez taper dans vos scripts avant de les exécuter dans la console de RStudio) renvoient des messages d'erreur, c'est que les packages que vous essayez de charger en mémoire ne sont pas installés sur votre ordinateur. Il vous faudra alors installer les packages manquants avec la fonction :

```
install.packages("nom_du_package")
```

Comme d'habitude, si tout ça est un peu flou pour vous, relisez la section 2.3 du livre de biométrie 2 disponible en ligne.

Vous aurez également besoin des jeux de données suivants :

- `Temperature.csv`
- `Testosterone.csv`
- `HornedLizards.csv`
- `HommesFemmes.txt`

3.2 Exploration préalable des données

Avant de se lancer dans les tests d'hypothèses, il est **toujours indispensable** d'examiner les données dont on dispose à l'aide, d'une part de statistiques descriptives numériques, et d'autres part, de graphiques exploratoires. Nous allons voir dans cette section quels indices statistiques il peut être utile de calculer et quelles représentations graphiques il peut être utile de réaliser afin de pouvoir se lancer dans des tests d'hypothèses risquer de grossières erreurs.

3.2.1 Un seul échantillon

3.2.1.1 Importation et examen visuel

Commencez par importer les données contenues dans le fichier `Temperature.csv`. Pour cela, utilisez l'assistant d'importation de RStudio. Si vous ne savez plus comment faire, consultez la section 5.3 du livre en ligne de Biométrie 2.

Vous stockerez les données dans un objet que vous nommerez `Temperature`. Après l'importation, taper son nom dans la console de RStudio doit produire le résultat suivant :

```
Temperature
```

```
# A tibble: 25 x 2
   individual temperature
        <dbl>       <dbl>
 1          1        98.4
 2          2        98.6
 3          3        97.8
 4          4        98.8
 5          5        97.9
 6          6        99
 7          7        98.2
 8          8        98.8
 9          9        98.8
10         10        99
```

```
# ... with 15 more rows
```

Ce tableau contient les températures corporelles de 25 adultes en bonne santé choisis au hasard parmi la population américaine.

La première chose à faire quand on travaille avec des données inconnues, c'est d'examiner les données brutes. Ici, les données sont importées au format `tibble`, donc seules les premières lignes sont visibles. Pour visualiser l'ensemble du tableau, utilisez la fonction `View()` :

```
View(Temperature)
```

Cette commande doit ouvrir un nouvel onglet présentant les données dans un tableur simplifié, en lecture seule.

On constate ici 2 choses que nous allons modifier :

1. la première colonne, intitulé `inidividual`, n'est pas véritablement une variable. Cette colonne ne contient qu'un identifiant qui est en fait identique au numéro de ligne. Nous allons donc supprimer cette colonne
2. les températures sont exprimées en degrés Fahrenheit, ce qui rend leur lecture difficile pour nous qui sommes habitués à utiliser le système métrique et les degrés Celsius. Nous allons donc convertir les températures en degrés Celsius grâce à la formule suivante :

$$C = \frac{F - 32}{1.8}$$

```
Temp_clean <- Temperature %>%
  select(-individual) %>%      # Suppression de la première colonne
  mutate(                      # Transformation des tempértures en Celsius
    temperature = (temperature - 32) / 1.8
    )

Temp_clean
```

```
# A tibble: 25 x 1
   temperature
         <dbl>
 1        36.9
 2        37.0
 3        36.6
 4        37.1
 5        36.6
 6        37.2
 7        36.8
 8        37.1
```

```
 9        37.1
10        37.2
# ... with 15 more rows
```

Il nous est maintenant possible d'examiner à nouveau les données avec la fonction `View()`. Avec des valeurs de températures comprises entre 36.3 °C et 37.8 °C, il n'y a visiblement pas de données aberrantes.

C'est toujours la première chose à faire : regarder les données brutes pour repérer : - La nature des variables présentes. - Les variables inutiles qui pourront être supprimées ou négligées. - Les unités des variables utiles, afin de pouvoir les convertir si nécessaire. - Les valeurs manquantes ou aberrantes qui demanderont toujours un soin particulier.

Une fois l'examen préliminaire des données réalisé, on peut passer au calcul des statistiques descriptives.

3.2.1.2 Statistiques descriptives

On s'intéresse ici au calcul de grandeurs statistiques nous apportant des renseignement sur la distribution des valeurs de l'échantillon. Les questions auxquelles on tente de répondre à ce stade sont les suivantes :

- Quelle est la tendance moyenne
- Quelle est la dispersion des données autour de la moyenne

Pour répondre à ces questions, on peut faire appel à de multiples fonctions. J'en évoquerai ici seulement 3 qui permettent d'obtenir la plupart des informations dont nous avons besoin très simplement :

```
summary(Temp_clean)
```

```
  temperature
 Min.   :36.33
 1st Qu.:36.67
 Median :37.00
 Mean   :36.96
 3rd Qu.:37.22
 Max.   :37.78
```

Comme son nom l'indique, la fonction `summary()` renvoie un résumé des données :

- les valeurs extrêmes (minimum et maximum)
- les valeurs "centrales" (moyenne et médiane)
- les valeurs des quartiles (premier et troisième quartiles)

Ces valeurs seront presques toutes reprises sur le graphique de type "boîte à moustaches" que nous verrons plus bas.

On constate ici que la moyenne et la médiane sont très proches. La distribution des température doit donc être à peut près symmétrique, avec à peu près autant de valeurs au dessus que de valeurs en dessous de la moyenne.

La seconde fonction utile est la fonction `IQR()`, comme "Inter Quartile Range" (ou intervalle inter-quartile). Cette fonction renvoie l'étendue de l'intervalle inter-quartile, c'est à dire la valeur du troisième quartile moins la valeur de premier quartile. Attention, cette fonction a besoin d'un vecteur en guise d'argument, or nos données sont stockées sous forme de `tibble`. Nous allons donc utiliser la fonction `pull()` du package `dplyr` afin de transformer (momentanément) la colonne `temperature` du tableau `Temp_clean` en vecteur :

```
Temp_clean %>% pull(temperature) %>% IQR()
```

```
[1] 0.5555556
```

On constate ici que l'intervalle inter quartile a une largeur de 0.55 degrés Celsius. Cela signifie que les 50% des températures les plus centrales sont situées dans un intervalle d'environ un demi-degré celsius.

Enfin, un eautre façon d'obtenir des informations rapidement consiste à utiliser la fonction `skim()` du package `skimr` :

```
skim(Temp_clean)
```

```
Skim summary statistics
 n obs: 25
 n variables: 1

-- Variable type:numeric -----
    variable missing complete  n  mean   sd    p0   p25 p50   p75
 temperature       0       25 25 36.96 0.38 36.33 36.67  37 37.22
  p100     hist
 37.78
```

(Attention : si vous lisez ce document au format pdf, vous ne pourrez pas visualiser correctement la totalité des résultats produits par cette fonction. Consultez la version html de ce document, ou tapez la commande dans RStudio).

Toute comme `summary()`, la fonction `skim()` renvoie les valeurs minimales et maximales, les premiers et troisièmes quartiles ainsi que la moyenne et la médiane. Elle nous indique en outre la valeur de l'écart-type de l'échantillon, ainsi que le nombre d'observation et le nombre de données manquantes. Enfin, elle fournit un histogramme très schématique et sans échelle. Cet histogramme nous permet de nous faire une première idée de la distribution des données.

Outre ces 3 fonctions (`summary()`, `IQR()`, et `skim()`), il est bien sûr possible de calculer toutes ces valeurs manuellement si besoin :

— `mean()` permet de calculer la moyenne

— `median()` permet de calculer la médiane
— `min()` et `max()` permettent de calculer les valeurs minimales et maximales respectivement
— `quantile()` permet de calculer les quartiles
— `sd()` permet de calculer l'écart-type
— `var()` permet de calculer la variance

Toutes ces fonctions prennent seulement un vecteur en guise d'argument. Il faut donc procéder comme avec `IQR()` pour les utiliser. Par exemple, pour calculer la variance, on peut taper :

```
Temp_clean %>% pull(temperature) %>% var()
```

```
[1] 0.1417901
```

ou :

```
var(Temp_clean$temperature)
```

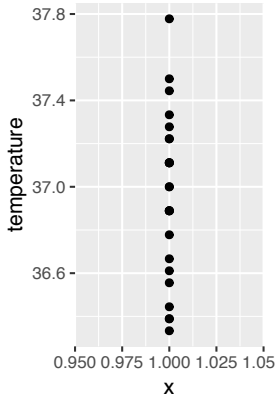
```
[1] 0.1417901
```

3.2.1.3 Exploration graphique

Ici, il s'agit d'examiner la distribution des données. Pour cela, 3 types de graphiques sont généralement utilisés.

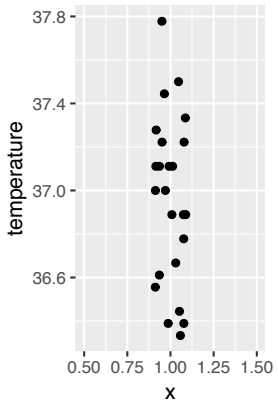
1. Les nuages de points ou stripcharts :

```
Temp_clean %>%
  ggplot(aes(x = 1, y = temperature)) +
  geom_point()
```

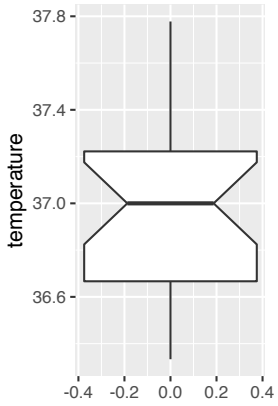


Dans la mesure ou souvent, plusieurs observations ont la même valeur, il faut tenir compte de l'over-plotting. Si vous ne vous rappelez plus de quoi il s'agit, consultez la section 4.3.4 du livre en ligne de Biométrie 2. Globalement, pour visualiser correctement les données, on va jouer soit sur la transparence des points, soit sur l'ajout d'un bruit aléatoire horizontal qui permettra de distinguer plsu facilement les points,m et de repérer les zones où les points sont abondants ou rares :

```
Temp_clean %>%
  ggplot(aes(x = 1, y = temperature)) +
  geom_jitter(height = 0, width = 0.1) +
  xlim(0.5, 1.5)
```

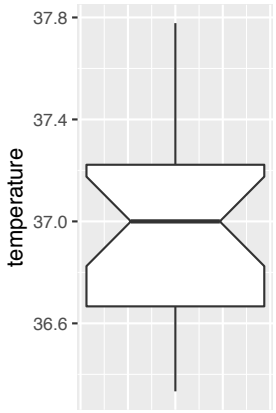



```
Temp_clean %>%
  ggplot(aes(y = temperature)) +
  geom_boxplot(notch = TRUE)
```



Ici, l'axe des abscisses n'a aucun sens. Nous n'avons qu'une unique série de données, l'axe des x est donc inutile. On peut le retirer :

```
Temp_clean %>%
  ggplot(aes(y = temperature)) +
  geom_boxplot(notch = TRUE) +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank())
```



3.2.2 Plusieurs échantillons

3.3 Comparaison de la moyenne d'une population à une valeur théorique

3.3.1 Le test paramétrique

3.3.1.1 Conditions d'application

3.3.2 Le test non paramétrique

3.4 Comparaison de la moyenne de 2 populations : données appariées

3.4.1 Le test paramétrique

3.4.1.1 Conditions d'application

3.4.2 Le test non paramétrique

3.5 Comparaison de la moyenne de 2 populations : données indépendantes

3.5.1 Le test paramétrique

3.5.1.1 Conditions d'application