Sampling and parameter estimation

September 16, 2020

In this lab, you will apply the concepts from the two last classes, in addition to learning how to create analysis reports with R Markdown.

Introduction to R Markdown

Create a R Markdown document

The R Markdown format allows you to combine text, chunks of R code, and results in one document. This tutorial is a very brief introduction to R Markdown. A more complete tutorial can be found at https://rmarkdown.rstudio.com/lesson-1.html.

In RStudio, create a new R Markdown document with the menu commands *File -> New File -> R Markdown...*. Choose the Word (.docx) output format.

> The advantage of the Word format is that it is possible to edit the resulting document in Word. This is the recommended format for handing in assignments in this course. The PDF format is also useful for producing documents, while the HTML format allows you to publish the results on the web. Note that all materials in this course are produced using R Markdown!

The file created already contains sample R Markdown text. Save the file (give it the name `example.Rmd`) and press the **Knit** button to produce the .docx file.

When reading the description of the different parts of the document, compare the .Rmd file and the result in Word.

Components of a R Markdown document

Header

The header of the file contains information such as title, date, and output format. It begins and ends with a block of three dashes `---`.

R code chunks

The R code chunks have a gray background when displayed in RStudio. They start and end with three backticks "'.

The shortcut **Ctrl + Alt + I** automatically inserts a new code chunk into the document.

The first code chunk (which contains `knitr::opts_chunk$set(echo = TRUE)`) is used to specify certain parameters. You can ignore it for now.

Look at the second chunk that contains `summary(cars)`. To the right of the three backticks on the first line, you find the chunk header surrounded by braces: `{r cars}`. It starts with `r` to indicate that it is code R, while`cars` is the name of the chunk. (It is optional to name the chunks.) The green arrow on the far right is used to execute the code and display the result.

Now look at the Word file. There you will find the code chunk followed by the result.

The second code chunk `plot(pressure)` produces the graph that you see in the Word document. The chunk header contains the option `echo = FALSE` which means that the code is invisible, only the result appears in Word.

Markdown text

The rest of the document is composed of text with some Markdown language markers for the layout.

Here are some examples of layout markers and their output.

`# Level 1 Header`

Level 1 Header

`## Level 2 Header`

Level 2 Header

`### Level 3 Header`

Level 3 Header

`Text in *italic*`

Text in *italic*

`Text in **bold**`

Text in **bold**

`- List Item - List Item`

- List item
- List item

You can now replace the code and text in the example with your answers to the following exercises.

Exercises

1. Characteristics of three species of penguins

For this exercise, we will use a dataset containing measurements taken on 344 penguins from three species (Adelie, Chinstrap and Gentoo) present on the Palmer Archipelago of Antarctica.

To load a dataset from a R package, you must first load the package, then read the dataset with the `data` function. Here, we will load the `penguins` dataset from the `palmerpenguins` package.

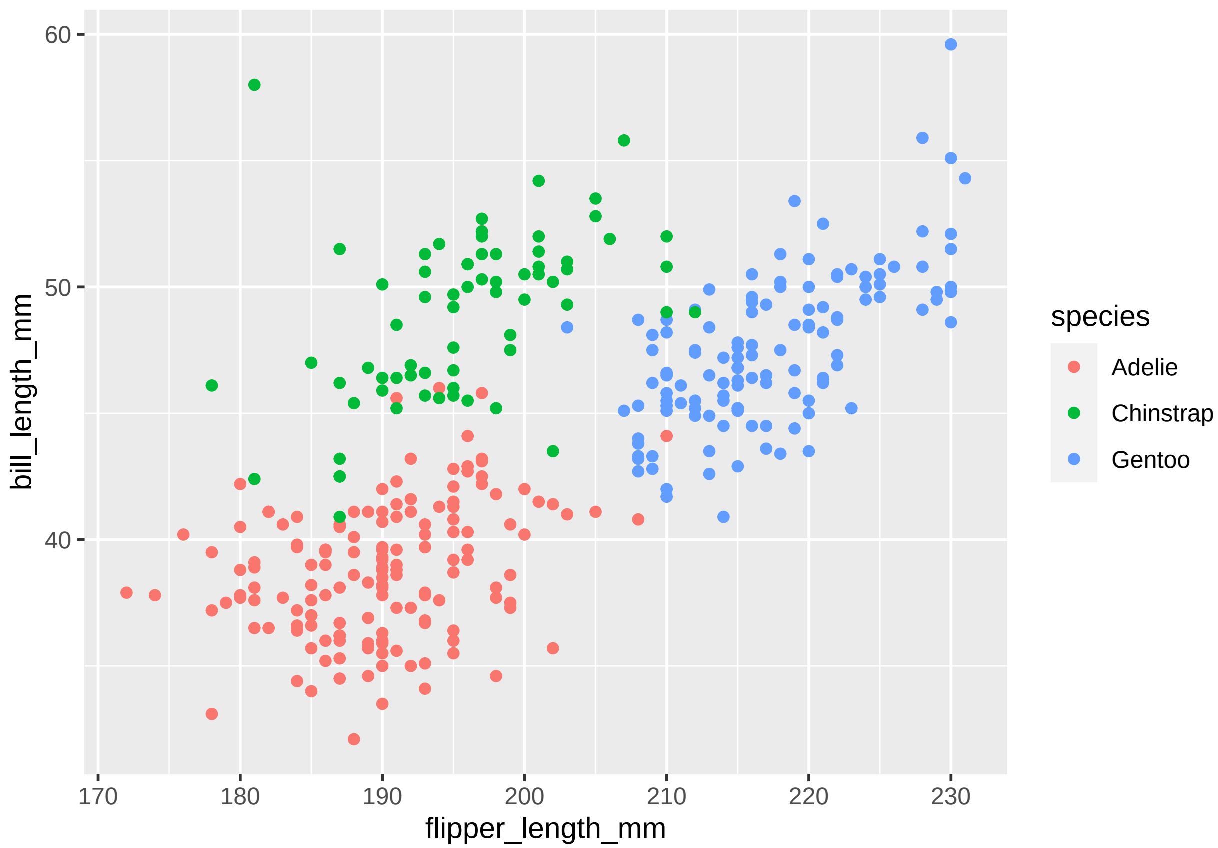
```r
library(palmerpenguins)
data(penguins)
head(penguins)
```

```
## # A tibble: 6 x 8
##   species island bill_length_mm bill_depth_mm flipper_length_~ body_mass_g sex
##   <fct>   <fct>           <dbl>         <dbl>            <int>       <int> <fct>
## 1 Adelie  Torge~           39.1          18.7              181        3750 male
## 2 Adelie  Torge~           39.5          17.4              186        3800 fema~
## 3 Adelie  Torge~           40.3          18                195        3250 fema~
## 4 Adelie  Torge~           NA            NA                 NA          NA <NA>
## 5 Adelie  Torge~           36.7          19.3              193        3450 fema~
## 6 Adelie  Torge~           39.3          20.6              190        3650 male
## # ... with 1 more variable: year <int>
```

a) Let's first visualize part of the data. With *ggplot2*, produce a scatterplot of the flipper length vs. the bill length of the penguins, using colors to differentiate the species.

Solution

```r
library(ggplot2)
library(dplyr)
ggplot(penguins, aes(x = flipper_length_mm, y = bill_length_mm, color = species)) +
    geom_point()
```



How would you calculate the mean of `flipper_length_mm` and its 95% confidence interval by species?

b) What quantities do you need for this calculation?

The mean, standard error (which depends on the standard deviation and sample size) and the probabilities of the t distribution for $p = 0.025$ and $p = 0.975$.

c) Using the *dplyr* package, calculate the mean, sample size, standard deviation and standard error of the mean of `flipper_length_mm` for each species. Save the result in a data frame `fl_stat`.

Hints

- Only keep rows where `flipper_length_mm` is not missing, using the condition `!is.na(flipper_length_mm)`.
- In `summarize`, you can use the `n()` function to count the number of rows per group, e.g.: `summarize(n = n(), ...)`. This is preferrable to using the `count` function when you want to calculate both the number of observations as well as other summary statistics.

Solution

After using `group_by` and `summarize` to calculate the sample size, the mean and the standard deviation by species, we use `mutate` to calculate the standard error from the standard deviation and the sample size.

```
fl_stat <- penguins %>%
    filter(!is.na(flipper_length_mm)) %>%
    group_by(species) %>%
    summarize(n = n(), moy = mean(flipper_length_mm),
              ecart_type = sd(flipper_length_mm)) %>%
    mutate(err_type = ecart_type / sqrt(n))
fl_stat
```

```
## # A tibble: 3 x 5
##   species       n   moy ecart_type err_type
##   <fct>     <int> <dbl>      <dbl>    <dbl>
## 1 Adelie      151  190.       6.54    0.532
## 2 Chinstrap    68  196.       7.13    0.865
## 3 Gentoo      123  217.       6.48    0.585
```

d) During the class on statistical distributions, we saw the functions `rnorm`,`dnorm`, `pnorm` and`qnorm` which calculate values from the normal distribution. Similar functions exist for the t distribution (`rt`,`dt`, `pt`,`qt`). Let's use the function `qt(p, df)` to determine the interval corresponding to 95% of the probability. What values of p (cumulative probability) should we use? How many degrees of freedom (df) based on sample size n?

p = 0.025 and 0.975, $df = n - 1$.

e) Create two new columns in `fl_stat` containing the minimum`ic_min` and the maximum `ic_max` of the confidence interval. Define these columns as functions of the mean, standard error and sample size.

Solution

We use `mutate` to create new columns and we calculate them with the formula for the minimum and maximum of the confidence intervals. These are obtained by taking the mean and adding the values of the proper quantiles of the t distribution (calculated with `qt` and the parameters specified in d) above), multiplied by the standard error.

```
fl_stat <- fl_stat %>%
    mutate(ic_min = moy + qt(0.025, df = n - 1) * err_type,
           ic_max = moy + qt(0.975, df = n - 1) * err_type)
fl_stat
```

```
## # A tibble: 3 x 7
##   species       n   moy ecart_type err_type ic_min ic_max
##   <fct>     <int> <dbl>      <dbl>    <dbl>  <dbl>  <dbl>
## 1 Adelie      151  190.       6.54    0.532   189.   191.
```

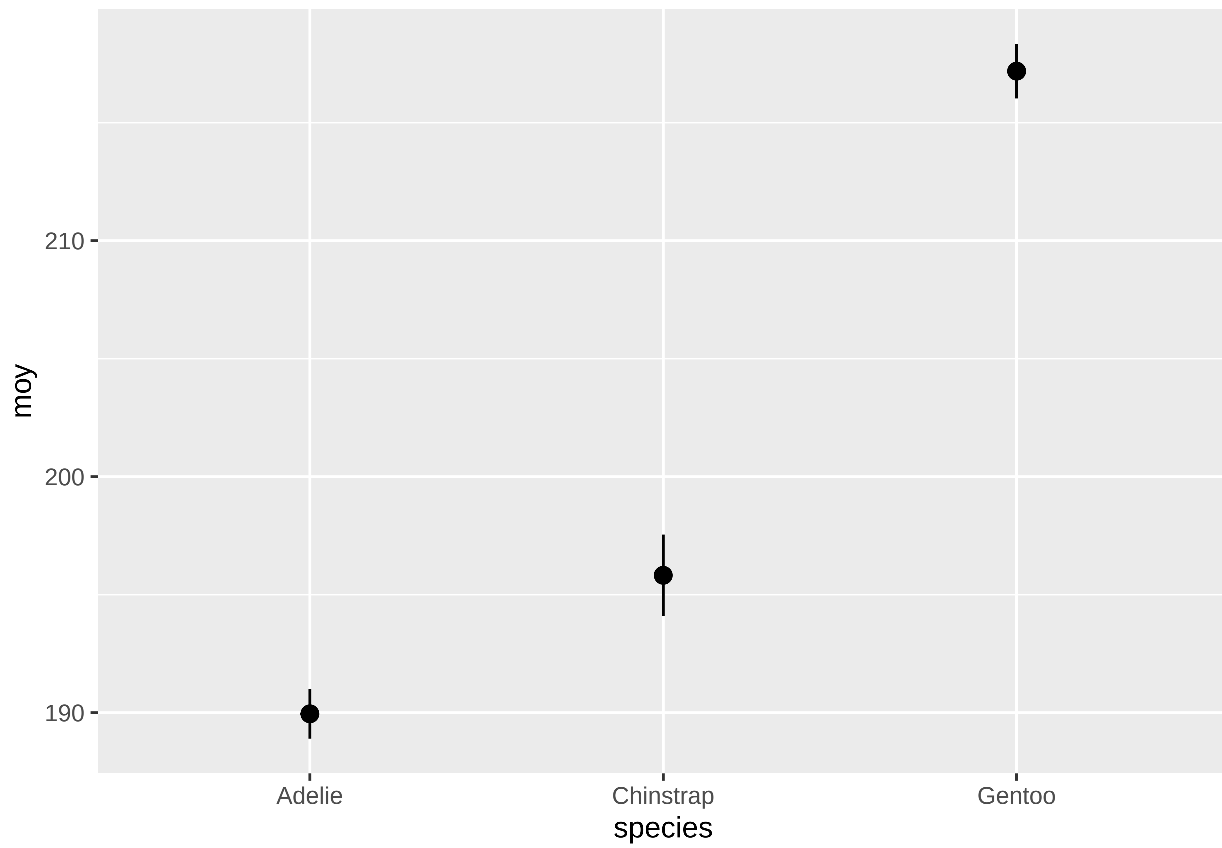
```
## 2 Chinstrap    68  196.        7.13    0.865   194.   198.
## 3 Gentoo      123  217.        6.48    0.585   216.   218.
```

f) Finally, use the `geom_pointrange` graph type of *ggplot2* to visualize the confidence interval for each species. This type of graph requires the specification of `y` (central point), `ymin` (minimum of range) and `ymax` (maximum of range) in the `aes` function.

We need a graph that shows the means and the confidence intervals with limits `ic_min` and `ic_max`.

Solution

```
ggplot(fl_stat, aes(x = species, y = moy, ymin = ic_min, ymax = ic_max)) +
    geom_pointrange()
```



2. Mean DBH of species sampled at Kejimkujik National Park

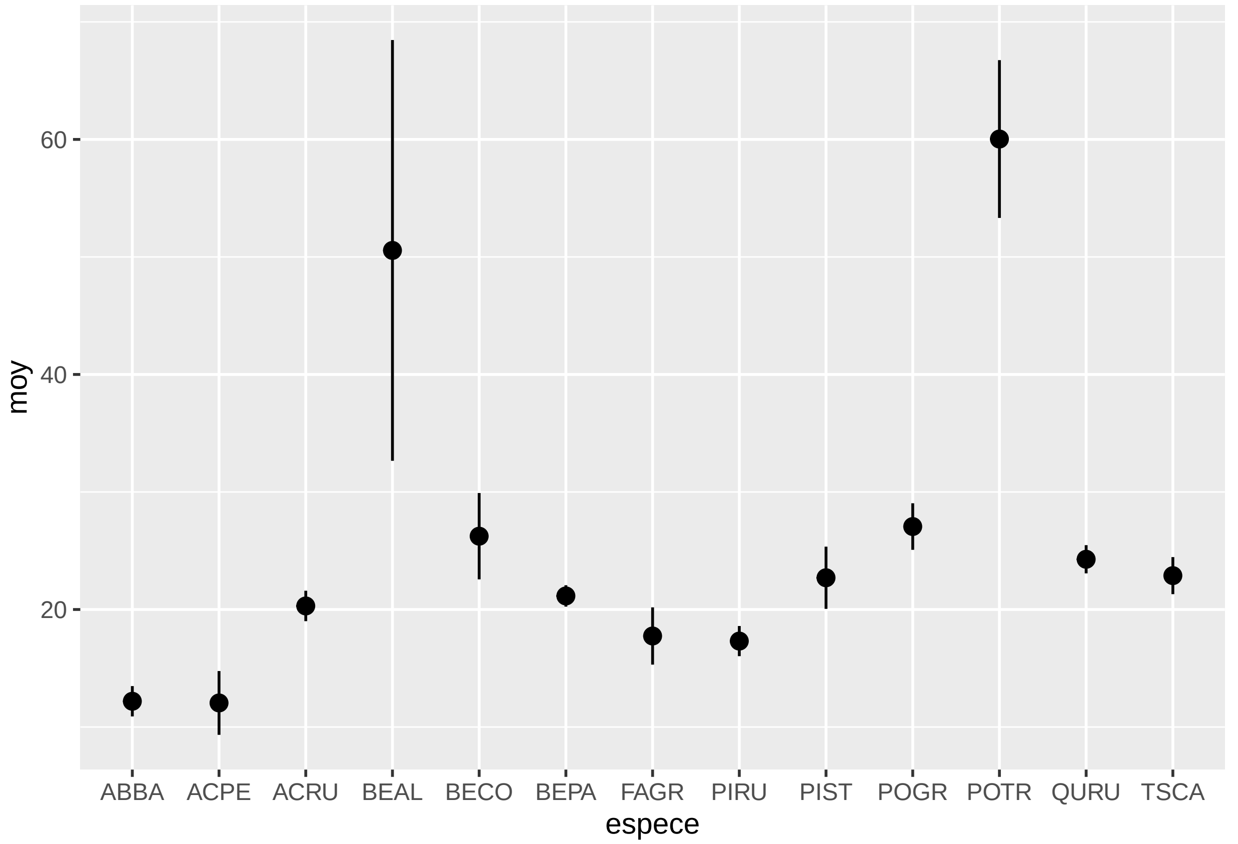
Based on the code written in the last exercise, produce a graph of the mean DBH with the 95% confidence interval for all species in the data frame from the file `cours1_kejimkujik.csv`.

Solution

```
kejim <- read.csv("cours1_kejimkujik.csv")

kejim_stats <- group_by(kejim, espece) %>%
    summarize(moy = mean(dhp), ec_type = sd(dhp), n = n()) %>%
    mutate(err_type = ec_type / sqrt(n),
           ic_min = moy + qt(0.025, df = n - 1) * err_type,
           ic_max = moy + qt(0.975, df = n - 1) * err_type)
```

```r
ggplot(kejim_stats, aes(x = espece, y = moy, ymin = ic_min, ymax = ic_max)) +
    geom_pointrange()
```



Then, from the data and graph, answer the following questions.

a) What is the confidence interval of the mean DBH for white pine (*PIST*)? Assuming that the sampled individuals are representative of the population on this site, how do you interpret this interval?

CI: 20.1 to 25.3 cm, by looking at the row for species PIST in the `kejim_stats` data frame. If we repeatedly sampled this many white pines (131) in this population, then in 95% of the cases, the confidence interval around the sample mean would include the population mean of DBH, in the other 5% of the cases the interval would not include the population mean.

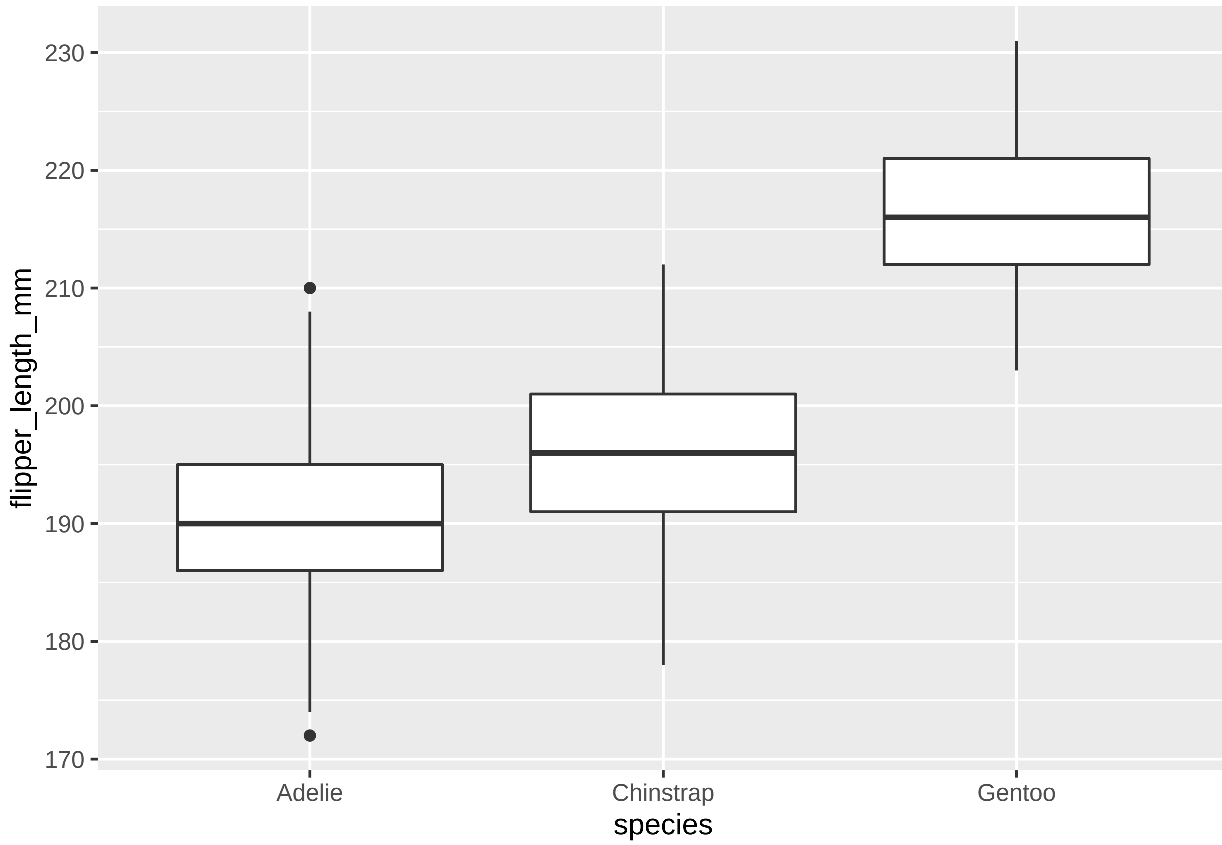
b) If the sample is representative of each species, can it be easily determined which species has the smallest mean DBH, or which has the highest mean DBH, in this population?

According to the graph, there are two candidate species for smallest mean DBH (ABBA and ACPE), and two candidates for largest mean DBH (BEAL and POTR). However, the confidence intervals overlap, showing we would need more data to identify the smallest or largest mean.

3. Stratified sampling simulation

For this exercise, we will compare simple and stratified sampling using simulated samples from the `penguins` dataset. Here is the distribution of flipper lengths for each species in the original data frame. Note that we have created a new `fl` table that contains only those individuals where flipper length was measured.

```r
fl <- filter(penguins, !is.na(flipper_length_mm))
ggplot(fl, aes(x = species, y = flipper_length_mm)) +
    geom_boxplot()
```



In *dplyr*, the `sample_n(df, n)` function returns a data frame containing `n` randomly selected observations from the data frame `df`. It can also be used with `group_by` to choose `n` observations by group.

a) Create two functions `fl_alea` and `fl_strat`. The first function chooses 30 random observations of `fl`, then returns the mean of `flipper_length_mm` for these observations. The second chooses 10 random observations from each of the three species, then returns the mean of `flipper_length_mm` (overall mean, not by species). Make sure that each of the two functions returns a vector of length 1.

Note: You can write these functions without arguments (empty parentheses after `function`), as in the example below.

```r
fl_alea <- function() {
    # Insert function code here
}
```

Solution

```r
fl_alea <- function() {
    samp <- sample_n(fl, 30)
    mean(samp$flipper_length_mm)
}

fl_strat <- function() {
    samp <- group_by(fl, species) %>%
```

```r
        sample_n(10)
    mean(samp$flipper_length_mm)
}

fl_alea()
```

```
## [1] 202.3667
```

```r
fl_strat()
```

```
## [1] 201.3
```

b) Generate a vector of 1000 results of each function with `replicate`, as follows:

```r
rep_alea <- replicate(1000, fl_alea())
rep_strat <- replicate(1000, fl_strat())
```

c) Calculate the standard error of each mean (from the standard deviation of `rep_alea` and `rep_strat`). Before running the calculation, can you predict which method will be more precise and why?

Solution

```r
sd(rep_alea)
```

```
## [1] 2.438234
```

```r
sd(rep_strat)
```

```
## [1] 1.155974
```

The standard error for stratified sampling (standard deviation of rep_strat) is about 2 times smaller than that of simple random sampling (standard deviation of rep_alea), i.e. 1.2 vs. 2.4. This is because flipper length is more variable between species than within each species.