Homework 4

GONG Kuiyuan

Student ID: 39-246182

Student name: GONG Kuiyuan

Preferred name: Eddie

Answers:

Task 1: We normally use the code `install.packages()` to download the package that we want. However, it shouldn't be written into the Rscript or Quarto file because the line of code gets to run everytime when we run the entire file. The problem here is twofold. Firstly, the console will report `Error` if the package is already uploaded. Secondly, if we are downloading packages from the Github, there may be a rate limit of package download per hour. People wouldn't be able to download the packages as long as it hits the limit.

The `p_load()` is better because it helps to check the download status of the package. If it is installed and automatically installs it if it isn't, then loads it. It avoids reinstalling a package that is already loaded.

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

library(pacman)
p_load(palmerpenguins)
```

Task 2: We typically use `?` to check the documentation of a package or function. So, for this package, it includes the size measurements, clutch observations, and blood isotope ratios for adult foraging Adélie, Chinstrap, and Gentoo penguins observed on islands in the Palmer Archipelago near Palmer Station, Antarctica. Additionally, we can also find other information including the *authors* and *useful links*.

We can use `str(penguins)` to describe the data type of each variable.

```
?palmerpenguins

names(penguins)
```

```
[1] "species"           "island"            "bill_length_mm"
[4] "bill_depth_mm"     "flipper_length_mm" "body_mass_g"
[7] "sex"               "year"
```

```
str(penguins)
```

```
tibble [344 x 8] (S3: tbl_df/tbl/data.frame)
 $ species          : Factor w/ 3 levels "Adelie","Chinstrap",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ island           : Factor w/ 3 levels "Biscoe","Dream",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ bill_length_mm   : num [1:344] 39.1 39.5 40.3 NA 36.7 39.3 38.9 39.2 34.1 42 ...
 $ bill_depth_mm    : num [1:344] 18.7 17.4 18 NA 19.3 20.6 17.8 19.6 18.1 20.2 ...
 $ flipper_length_mm: int [1:344] 181 186 195 NA 193 190 181 195 193 190 ...
 $ body_mass_g      : int [1:344] 3750 3800 3250 NA 3450 3650 3625 4675 3475 4250 ...
 $ sex              : Factor w/ 2 levels "female","male": 2 1 1 NA 1 2 1 2 NA NA ...
 $ year             : int [1:344] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
```

```
#summary(penguins) This is helpful to get descriptive statistics.
```

Task 3: We can use the verb "rename" to change the names of variables. We can also combine it together with pipe to make the process clearer to follow.

```
names(penguins)
```

```
[1] "species"           "island"            "bill_length_mm"
[4] "bill_depth_mm"     "flipper_length_mm" "body_mass_g"
[7] "sex"               "year"
```

```
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag
```

```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
# Using dplyr verb 'rename'.
penguins_processed <- rename(penguins, body_mass = body_mass_g)
names(penguins_processed)
```

```
[1] "species"           "island"            "bill_length_mm"
[4] "bill_depth_mm"     "flipper_length_mm" "body_mass"
[7] "sex"               "year"
```

```r
# Using 'pipe'.
penguins_processed <- penguins_processed |>
  rename(bill_length = bill_length_mm) |>
  rename(bill_depth = bill_depth_mm) |>
  rename(flipper_length = flipper_length_mm)

names(penguins_processed)
```

```
[1] "species"        "island"         "bill_length"    "bill_depth"
[5] "flipper_length" "body_mass"      "sex"            "year"
```

Task 4:

```r
p_load(griffen, griffendata)
names(mini_cps)
```

```
[1] "race"               "years_of_education" "experience"
[4] "education_category" "wage"
```

```r
if_else(c("white", "black", "black") == "black", 1, 0)
```

```
[1] 0 1 1
```

```r
if_else(mini_cps$race == "black", 1, 0)
```

```
[1] 1 0 0 1 1
```

```
# We can also apply this to mini_cps data.
```

Task 5:

```
penguins_processed <- penguins_processed |>
  mutate (dummy_1 = if_else(body_mass > 5000, 1, 0))

penguins_processed$dummy_1
```

```
  [1]  0  0  0 NA  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 [26]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 [51]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 [76]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[101]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[126]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[151]  0  0  0  1  0  1  1  0  0  1  0  1  0  1  0  1  0  1  0  1  0  1  1  0  0
[176]  1  0  1  0  1  0  1  1  0  1  1  1  1  0  1  0  1  0  1  0  0  1  0  0  1
[201]  1  1  0  1  0  0  0  1  0  0  0  1  0  1  0  1  0  1  0  1  0  1  0  0  1
[226]  1  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0  0
[251]  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0 NA  0  1  1
[276]  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[301]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[326]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Task 6:

```
penguins_processed <- penguins_processed |>
  left(dummy_1)

names(penguins_processed)
```

```
[1] "dummy_1"        "species"        "island"         "bill_length"
[5] "bill_depth"     "flipper_length" "body_mass"      "sex"
[9] "year"
```

Task 7: Since we have made a dummy, we can easily get the percentage by calculating the mean of the dummy variable and its product with 100. The final result is 17.83626%.

```r
penguins_processed |>
  summarise(percentage_weight = mean(dummy_1) * 100)
```

```
# A tibble: 1 x 1
  percentage_weight
              <dbl>
1              17.8
```

Task 8: Yes, the result is different, where we can find that only some penguins that come from Biscoe island have a weight over 5000 grams, which is 36.5% of the total population. Penguins come from Dream island and Torgersen island are generally smaller than 5000 grams.

```r
penguins_processed |>
  group_by(island) |>
  summarise(percentage_weight = mean(dummy_1) * 100)
```

```
# A tibble: 3 x 2
  island    percentage_weight
  <fct>                 <dbl>
1 Biscoe                 36.5
2 Dream                   0
3 Torgersen               0
```

Task 9: Yes, Sexual dimorphism is true given that the average weight of females are smaller than males. On the other hand, the species of penguins have very different average weight as well. We observe that *Gentoo* penguins have the biggest size and the other two are similar.

```r
penguins_processed |>
  group_by(sex) |>
  summarise(mean_body_mass = mean(body_mass))
```

```
# A tibble: 3 x 2
  sex    mean_body_mass
  <fct>           <dbl>
1 female          3862.
2 male            4546.
3 <NA>            4006.
```

```r
penguins_processed |>
  group_by(species) |>
  summarise(mean_body_mass = mean(body_mass))
```

```
# A tibble: 3 x 2
  species   mean_body_mass
  <fct>              <dbl>
1 Adelie             3701.
2 Chinstrap          3733.
3 Gentoo             5076.
```

Task 10: Note that `dataFrame` here could be any data frame that contains `body_mass` and `sex`. This function allows us to pass data frame as an argument and get the results.

```r
regression <- function(dataFrame){
  lm (body_mass ~ sex, dataFrame)
}

regression(penguins_processed)
```

```
Call:
lm(formula = body_mass ~ sex, data = dataFrame)

Coefficients:
(Intercept)      sexmale
     3862.3        683.4
```