p8105_hw1_WL3011

Weiqi Liang

2024-09-20

1. Problem 1

Firstly, the **penguins** dataset and the **palmerpenguins** package are loaded.

```
# Load the penguins dataset
library(palmerpenguins)
library(ggplot2)
library(dplyr)
data("penguins", package = "palmerpenguins")
```

1.1 Description of Penguins Dataset

The **penguins** dataset consists of **344** observations about 3 species of penguins: **Adelie**, **Gentoo**, and **Chinstrap**.

It includes following **8** important variables: species, island, bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g, sex, year.

Besides, it has 344 rows and 8 columns. All penguins' average flipper length is 200.9152047 mm.

```
names(penguins)
## [1] "species"           "island"            "bill_length_mm"
## [4] "bill_depth_mm"     "flipper_length_mm" "body_mass_g"
## [7] "sex"               "year"
nrow(penguins)
## [1] 344
ncol(penguins)
## [1] 8
mean(pull(penguins,flipper_length_mm), na.rm = TRUE)
## [1] 200.9152
```

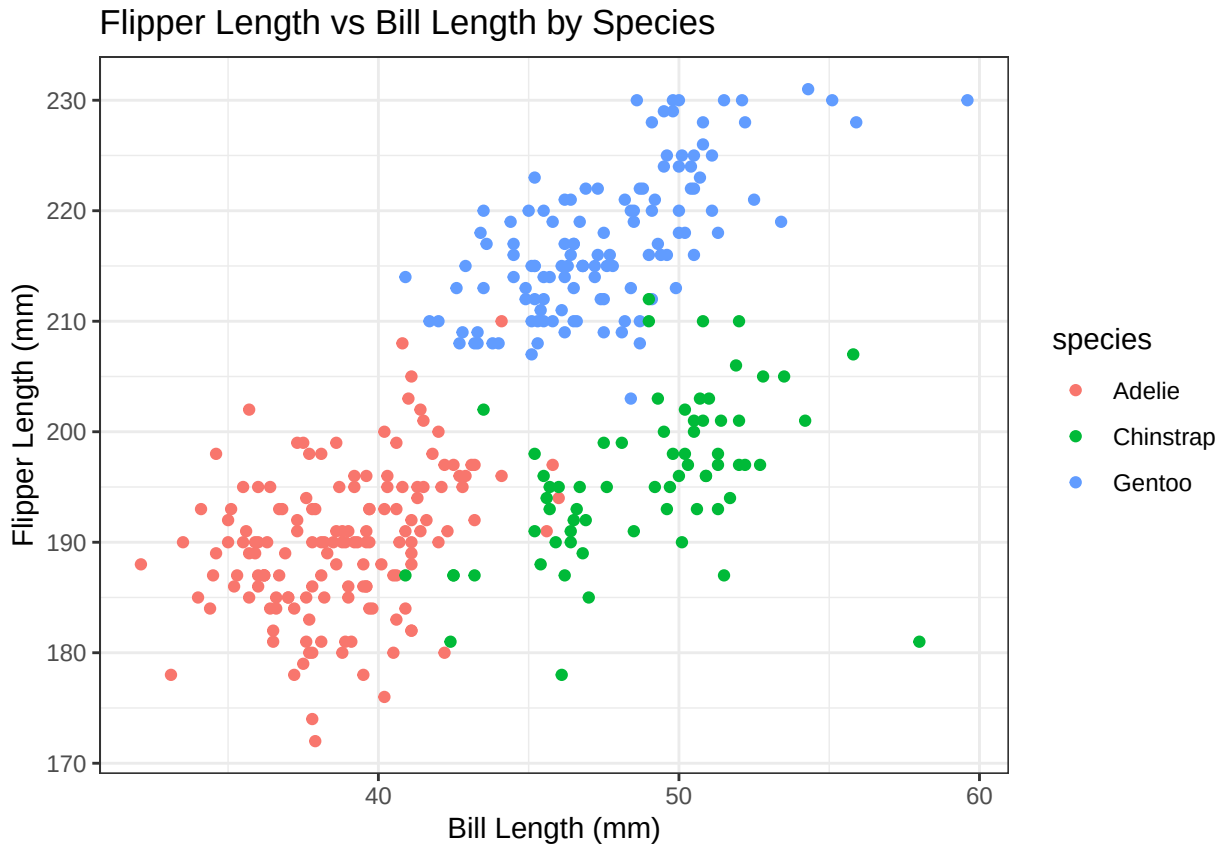
1.2 Flipper Length vs Bill Length

Using the following code to create a scatterplot of **flipper_length** vs **bill_ length** by **speicies**.

```
# Create scatterplot
p <- ggplot(penguins, aes(x = bill_length_mm,
                          y = flipper_length_mm,
                          color = species)) +
```

```
  geom_point(na.rm = TRUE) +
  labs(title = "Flipper Length vs Bill Length by Species",
       x = "Bill Length (mm)",
       y = "Flipper Length (mm)") +
  theme_bw()

print(p)
```



Next, use **ggsave** to export this scatterplot to the project directory.

```
# save as png
ggsave("penguins_scatterplot.png", plot = p)
```

```
## Saving 6.5 x 4.5 in image
```

2. Problem 2

2.1 Create Data Frame

Firstly, create a data frame as follows:

```
library(tidyverse)
set.seed(3011)
```

```r
samp_df = tibble(
  norm_samp = rnorm(10),
  norm_samp_flag = norm_samp > 0,
  char_vector = c("A", "B", "C", "D", "E", "F", "G", "H", "I", "J"),
  factor_vector = factor(rep(c("Level5", "Level2", "Level3"),
                             length.out = 10))
)
```

```r
print(samp_df)
```

```
## # A tibble: 10 x 4
##    norm_samp norm_samp_flag char_vector factor_vector
##        <dbl> <lgl>          <chr>       <fct>
##  1    -0.664 FALSE          A           Level5
##  2    -2.16  FALSE          B           Level2
##  3    -0.265 FALSE          C           Level3
##  4     2.14  TRUE           D           Level5
##  5    -1.24  FALSE          E           Level2
##  6    -0.206 FALSE          F           Level3
##  7    -0.591 FALSE          G           Level5
##  8     1.29  TRUE           H           Level2
##  9     0.757 TRUE           I           Level3
## 10    -1.49  FALSE          J           Level5
```

2.2 Take the Mean of Each Variable

The mean of numeric variables (`norm_samp`) and logical variables (`norm_samp_flag`) will work.

```r
mean(pull(samp_df, norm_samp))
## [1] -0.2426275
mean(pull(samp_df, norm_samp_flag))
## [1] 0.3
```

While the character (`char_vector`) and factor (`factor_vector`) variables cannot have a mean calculated since they are strings instead of numeric or logical variables.

```r
mean(pull(samp_df, char_vector))
mean(pull(samp_df, factor_vector))
```

2.3 Convert Variables

- Logical values **TRUE** and **FALSE** are treated as **1** and **0**, respectively. Therefore the mean of logical variables can work.
- The conversion of character variables will result in **NA** values and a warning message like "NAs introduced by coercion".
- The conversion of factor variable works, but it may not represent meaningful numeric values. In this case, **Level5**, **Level2**, and **Level3** are converted to numeric values **3**, **1**, and **2** respectively. This means that it converts the ascending order of Level to numeric values.

```
as.numeric(pull(samp_df,norm_samp_flag))
##  [1] 0 0 0 1 0 0 0 1 1 0
as.numeric(pull(samp_df,char_vector))
##  [1] NA NA NA NA NA NA NA NA NA NA
as.numeric(pull(samp_df,factor_vector))
##  [1] 3 1 2 3 1 2 3 1 2 3
```

This helps explain why:

- Logical variables can be converted into numeric values and their means can be calculated.
- Character variables cannot be converted to numeric. Therefore the mean of character variables cannot work.
- Factor variables can be converted into numeric values, however not in a meaningful way. After a forced conversion of `as.numeric`, the mean can be calculated.