hw1

Zitao Zhang

2024-09-21

Probelm 1

```
# Load dataset
data("penguins", package = "palmerpenguins")
```

The penguins dataset contains data on penguin species observed in the Palmer Archipelago, Antarctica. It includes measurements of various physical characteristics and other information for three different penguin species.

Important variables in the dataset include:

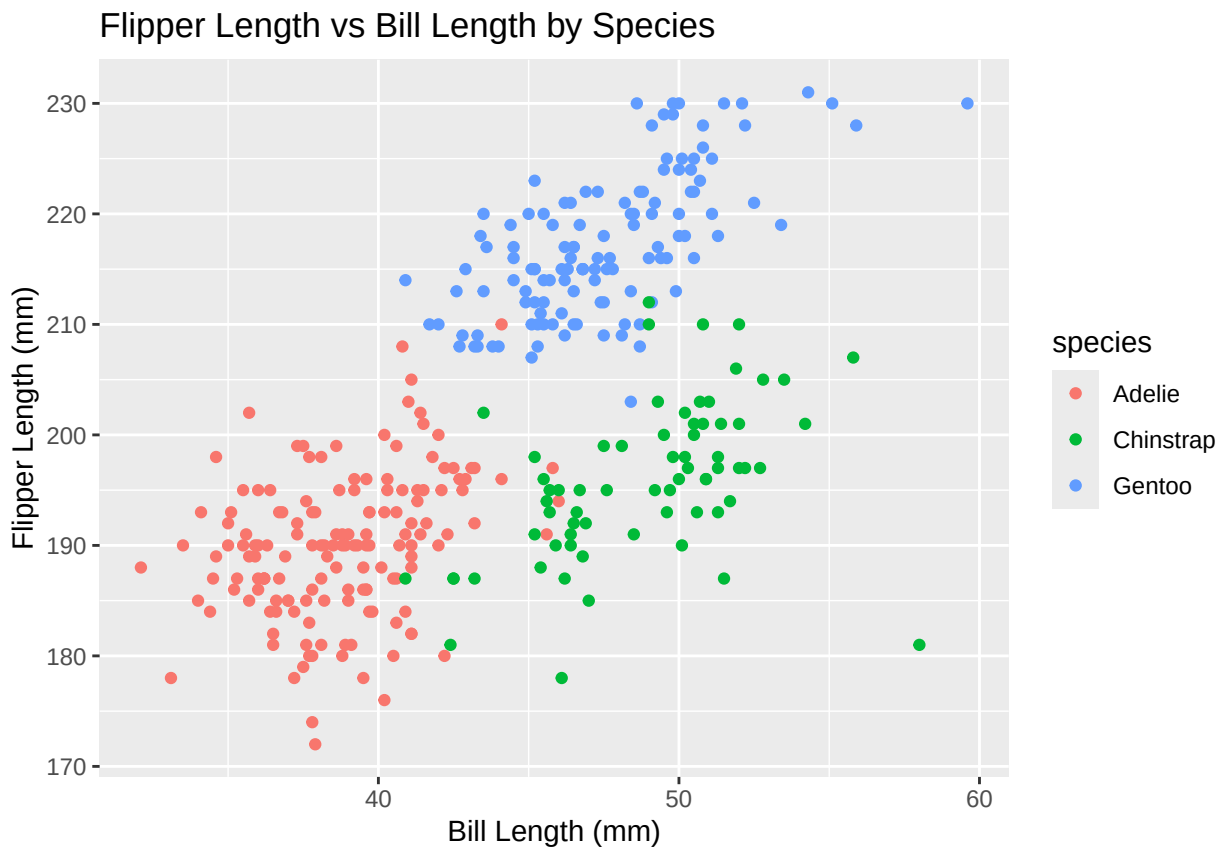
- **species**: The penguin species (`Adelie, Gentoo, Chinstrap`).
- **island**: The island where the penguin was observed (`Torgersen, Biscoe, Dream`).
- **bill_length_mm**: Length of the penguin's bill in millimeters.
- **bill_depth_mm**: Depth of the penguin's bill in millimeters.
- **flipper_length_mm**: Length of the penguin's flipper in millimeters.
- **body_mass_g**: Body mass of the penguin in grams.
- **sex**: Sex of the penguin (male, female).
- **year**: Year of observation (2007, 2008, 2009).

The dataset has 344 rows and 8 columns.

The mean flipper length is 200.92 mm.

```
# Make the scatterplot
p <- ggplot(data = penguins, aes(x = bill_length_mm, y = flipper_length_mm, color = species)) +
  geom_point() +
  labs(title = "Flipper Length vs Bill Length by Species",
       x = "Bill Length (mm)",
       y = "Flipper Length (mm)")

# Print the plot
print(p)
```




```
ggsave("penguins.png", plot = p)
```

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

Problem 2

```
set.seed(8105)

# A random sample of size 10 from a standard Normal distribution
numeric_var <- rnorm(10)

# A logical vector indicating whether elements of the sample are greater than 0
logical_var <- numeric_var > 0

# A character vector of length 10
character_var <- sample(letters, 10)

# A factor vector of length 10, with 3 different factor "levels"
factor_levels <- c("Level1", "Level2", "Level3")
factor_var <- factor(sample(factor_levels, 10, replace = TRUE))

# Create the data frame
df <- data.frame(numeric_var, logical_var, character_var, factor_var)
```

Try to take mean to see what works and what doesn't

```
# Take mean
mean_numeric <- mean(pull(df, numeric_var))
mean_logical <- mean(pull(df, logical_var))
mean_character <- try(mean(pull(df, character_var)), silent = TRUE)
```

```
## Warning in mean.default(pull(df, character_var)): argument is not numeric or
## logical: returning NA
```

```
mean_factor <- try(mean(pull(df, factor_var)), silent = TRUE)
```

```
## Warning in mean.default(pull(df, factor_var)): argument is not numeric or
## logical: returning NA
```

```
# Print result
mean_numeric
```

```
## [1] 0.5520184
```

```
mean_logical
```

```
## [1] 0.7
```

```
mean_character
```

```
## [1] NA
```

```
mean_factor
```

```
## [1] NA
```

- The mean of the numeric variable numeric_var works as expected and returns a numeric value.
- The mean of the logical variable logical_var works because R makes TRUE to 1 and FALSE to 0.
- The mean of the character variable character_var does not work; it results in an error because characters cannot be averaged.
- The mean of the factor variable factor_var does not work directly; it results in an error because factors need to be converted to numeric values representing their levels before calculating the mean.

Apply as.numeric function

```
# Convert variables to numeric
numeric_logical <- as.numeric(df$logical_var)
numeric_character <- as.numeric(df$character_var)
```

```
## Warning: NAs introduced by coercion
```

```r
numeric_factor <- as.numeric(df$factor_var)

# Print results
numeric_logical
```

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

```r
numeric_character
```

```
##  [1] NA NA NA NA NA NA NA NA NA NA
```

```r
numeric_factor
```

```
##  [1] 1 2 2 2 3 2 1 3 3 2
```

Logical Variable Conversion: Converting logical_var to numeric results in a numeric vector where TRUE is converted to 1 and FALSE to 0. This explains why taking the mean of a logical vector works, as it effectively calculates the proportion of TRUE values.

Character Variable Conversion: Converting character_var to numeric results in NA values and a warning message: NAs introduced by coercion. This is because character strings cannot be directly converted to numeric values unless they represent numbers.

Factor Variable Conversion: Converting factor_var to numeric results in the underlying integer codes that represent the factor levels. This can be misleading if not properly understood. For example, if the levels are "Level1", "Level2", "Level3", they are internally represented as 1, 2, 3. Calculating the mean of these numeric codes may not be meaningful in the context of the data.