Assignment 3

Xingyu Wang

2023-10-13

1. Exploratory data analysis

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────────────── tidy
verse 2.0.0 ──
## ✔ dplyr     1.1.3     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.3     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.0
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────────────────────
────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
me errors
```

```
library(Stat2Data)
data("Hawks")
```

1.1 (Q1)

```
# Creating the HawksTail vector
HawksTail <- Hawks$Tail

# Display the first few elements of the vector
head(HawksTail)
```

```
## [1] 219 221 235 220 157 230
```

```
# Output: [1] 219 221 235 220 157 230

# Compute the sample mean and sample median
sample_mean <- mean(HawksTail, na.rm = TRUE)  # using na.rm = TRUE to handle any NA values
sample_median <- median(HawksTail, na.rm = TRUE)

# Display the sample mean and sample median
sample_mean
```

```
## [1] 198.8315
```

```
sample_median
```

```
## [1] 214
```

1.2 (Q1)

```
# Using summarise() to compute mean, trimmed mean, and median for Wing and Weight columns
result <- Hawks %>%
  summarise(
    Wing_mean = mean(Wing, na.rm = TRUE),
    Wing_t_mean = mean(Wing, trim = 0.5, na.rm = TRUE),  # trimmed mean with q=0.5
    Wing_med = median(Wing, na.rm = TRUE),
    Weight_mean = mean(Weight, na.rm = TRUE),
    Weight_t_mean = mean(Weight, trim = 0.5, na.rm = TRUE),  # trimmed mean with q=0.5
    Weight_med = median(Weight, na.rm = TRUE)
  )

# Display the result
print(result)
```

```
##   Wing_mean Wing_t_mean Wing_med Weight_mean Weight_t_mean Weight_med
## 1  315.6375         370      370    772.0802           970        970
```

1.2 (Q2)

```
# Grouping by Species and then using summarise() to compute mean, trimmed mean, and median
grouped_result <- Hawks %>%
  group_by(Species) %>%
  summarise(
    Wing_mean = mean(Wing, na.rm = TRUE),
    Wing_t_mean = mean(Wing, trim = 0.5, na.rm = TRUE),  # trimmed mean with q=0.5
    Wing_med = median(Wing, na.rm = TRUE),
    Weight_mean = mean(Weight, na.rm = TRUE),
    Weight_t_mean = mean(Weight, trim = 0.5, na.rm = TRUE),  # trimmed mean with q=0.5
    Weight_med = median(Weight, na.rm = TRUE)
  )

# Display the grouped result
print(grouped_result)
```

```
## # A tibble: 3 × 7
##   Species Wing_mean Wing_t_mean Wing_med Weight_mean Weight_t_mean Weight_med
##   <fct>       <dbl>       <dbl>    <dbl>       <dbl>         <dbl>      <dbl>
## 1 CH           244.         240      240        420.          378.       378.
## 2 RT           383.         384      384       1094.         1070       1070
## 3 SS           185.         191      191        148.          155        155
```

....

1.3 (Q1)

```
a <- 2
b <- 3

transformed_mean <- mean(HawksTail * a + b)
calculated_mean <- a * mean(HawksTail) + b

# Compare the two means:
transformed_mean
```

```
## [1] 400.663
```

```
calculated_mean
```

```
## [1] 400.663
```

1,3 (Q2)

```
transformed_variance <- var(HawksTail * a + b)
calculated_variance <- aˆ2 * var(HawksTail)

transformed_sd <- sd(HawksTail * a + b)
calculated_sd <- a * sd(HawksTail)

# Compare the variance and standard deviation:
transformed_variance
```

```
## [1] 5424.147
```

```
calculated_variance
```

```
## [1] 5424.147
```

```
transformed_sd
```

```
## [1] 73.64881
```

```
calculated_sd
```

```
## [1] 73.64881
```

….

1.4

```
hal<-Hawks$Hallux # Extract the vector of hallux lengths
hal<-hal[!is.na(hal)] # Remove any nans
outlier_val<-100
num_outliers<-10
corrupted_hal<-c(hal,rep(outlier_val,times=num_outliers))
mean(hal)
```

```
## [1] 26.41086
```

```
mean(corrupted_hal)
```

```
## [1] 27.21776
```

```
num_outliers_vect <- seq(0,1000)
means_vect <- c()
for(num_outliers in num_outliers_vect){
corrupted_hal <- c(hal,rep(outlier_val,times=num_outliers))
means_vect <- c(means_vect, mean(corrupted_hal))
}
```

1.4 (Q1)

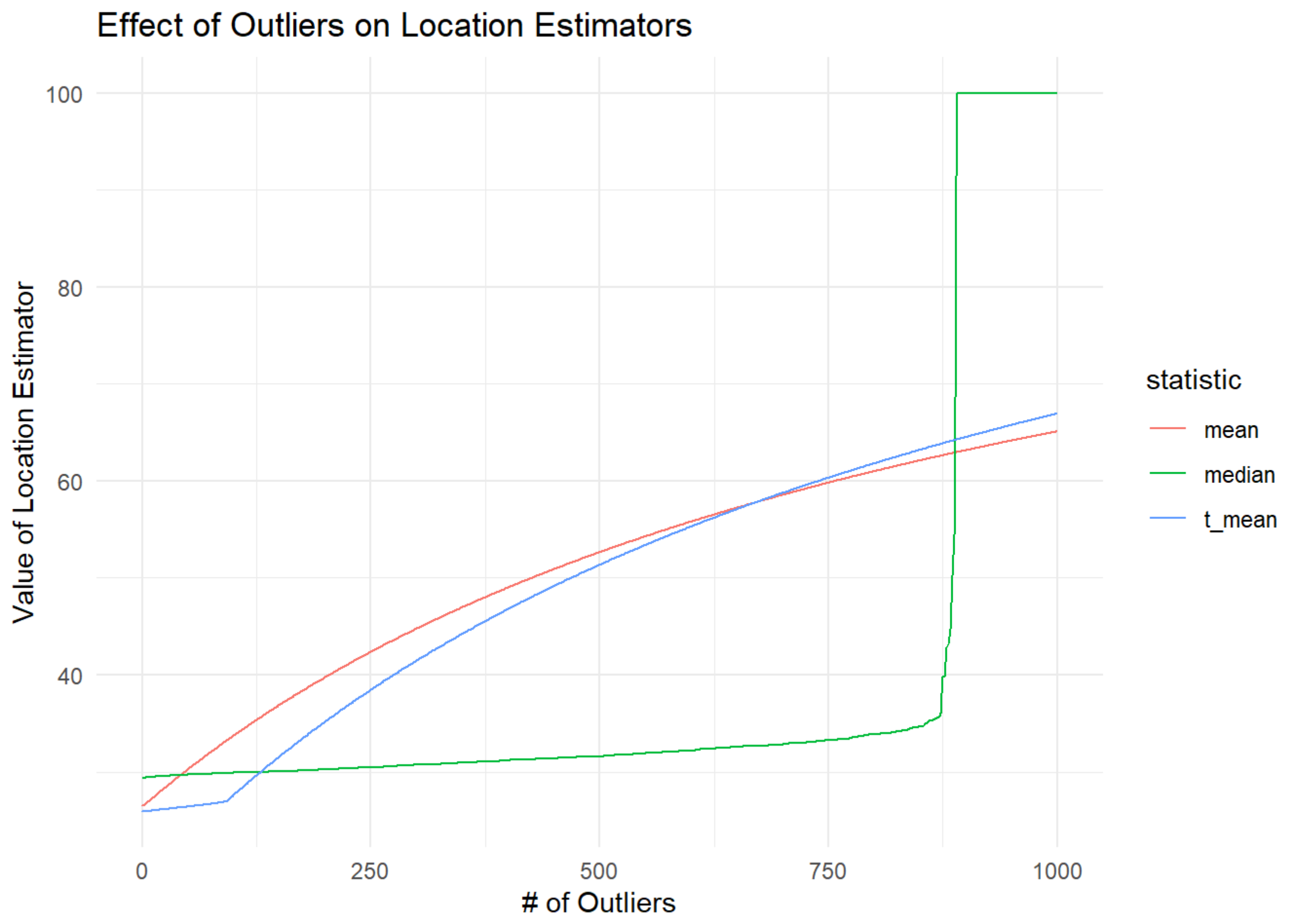
```
num_outliers_vect <- seq(0,1000)
medians_vect <- c()

for(num_outliers in num_outliers_vect){
  corrupted_hal <- c(hal, rep(outlier_val, times=num_outliers))
  medians_vect <- c(medians_vect, median(corrupted_hal))
}
```

1.4 (Q2)

```r
num_outliers_vect <- seq(0,1000)
t_means_vect <- c()

for(num_outliers in num_outliers_vect){
  corrupted_hal <- c(hal, rep(outlier_val, times=num_outliers))
  t_means_vect <- c(t_means_vect, mean(corrupted_hal, trim = 0.1))
}
```

1.4 (Q3)

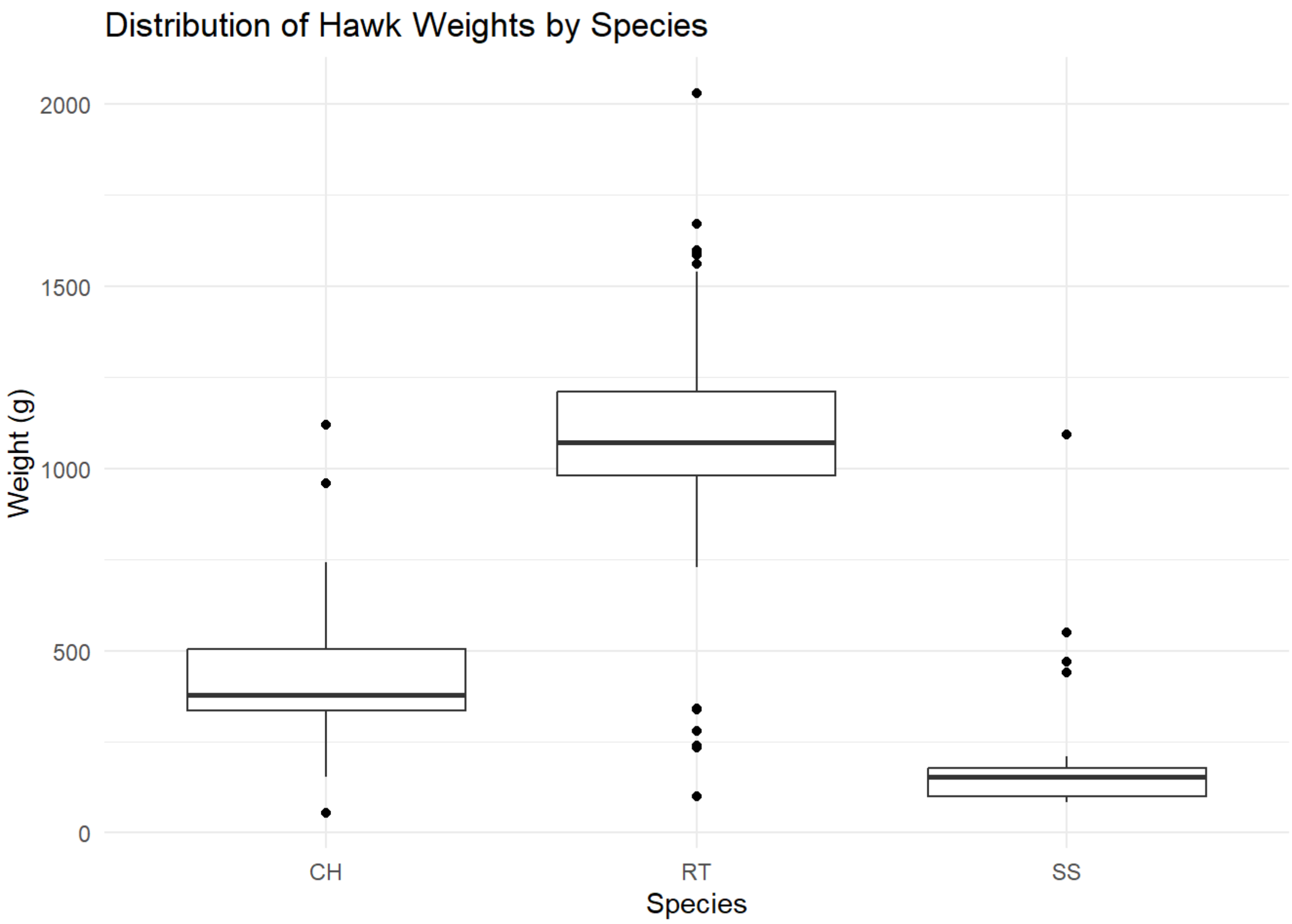
```r
# Combining into a dataframe
df_means_medians <- data.frame(
  num_outliers=num_outliers_vect,
  mean=means_vect,
  t_mean=t_means_vect,
  median=medians_vect
)

# Reshape data for plotting
df_long <- df_means_medians %>%
  pivot_longer(
    cols = -num_outliers,
    names_to = "statistic",
    values_to = "value"
  )

# Plotting
ggplot(df_long, aes(x=num_outliers, y=value, color=statistic)) +
  geom_line() +
  labs(title="Effect of Outliers on Location Estimators",
       x="# of Outliers",
       y="Value of Location Estimator") +
  theme_minimal()
```



1.5(Q1)

```
# Plotting the data
ggplot(Hawks, aes(x=Species, y=Weight)) +
  geom_boxplot(outlier.color = "black", outlier.shape = 16) +
  labs(title="Distribution of Hawk Weights by Species",
       x="Species",
       y="Weight (g)") +
  theme_minimal()
```

```
## Warning: Removed 10 rows containing non-finite values (`stat_boxplot()`).
```



## 1.5(Q2)

```
# Grouping by species and computing quantiles
quantiles_df <- Hawks %>%
  group_by(Species) %>%
  summarise(
    quantile025 = quantile(Weight, 0.25, na.rm = TRUE),
    quantile050 = quantile(Weight, 0.50, na.rm = TRUE),
    quantile075 = quantile(Weight, 0.75, na.rm = TRUE)
  )

print(quantiles_df)
```

```
## # A tibble: 3 × 4
##   Species quantile025 quantile050 quantile075
##   <fct>         <dbl>       <dbl>       <dbl>
## 1 CH              335        378.         505
## 2 RT              980       1070         1210
## 3 SS              100        155          178.
```

quantile025 corresponds to the lower hinge (the bottom of the box) of the boxplot. quantile050 corresponds to the median (the line inside the box) of the boxplot. quantile075 corresponds to the upper hinge (the top of the box) of the boxplot. ## 1.5(Q3)

```
num_outliers <- function(sample_vector) {
  # Removing NA values
  sample_vector <- sample_vector[!is.na(sample_vector)]

  # Calculating the quantiles
  q25 <- quantile(sample_vector, 0.25)
  q75 <- quantile(sample_vector, 0.75)

  # Calculating the Interquartile Range (IQR)
  IQR <- q75 - q25

  # Finding outliers based on the provided conditions
  lower_bound <- q25 - 1.5 * IQR
  upper_bound <- q75 + 1.5 * IQR

  outliers <- sample_vector[sample_vector < lower_bound | sample_vector > upper_bound]

  return(length(outliers))
}

# Testing the function
num_outliers( c(0, 40, 60, 185))
```

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

1.5(Q4)

```
outliers_by_species <- Hawks %>%
  group_by(Species) %>%
  summarise(
    num_of_outliers = num_outliers(Weight)
  )

print(outliers_by_species)
```

```
## # A tibble: 3 × 2
##   Species num_of_outliers
##   <fct>             <int>
## 1 CH                    3
## 2 RT                   13
## 3 SS                    4
```

1.6(Q1)

```
# Compute covariance
cov_weight_wing <- cov(Hawks$Weight, Hawks$Wing, use = "complete.obs")

# Compute correlation
cor_weight_wing <- cor(Hawks$Weight, Hawks$Wing, use = "complete.obs")

cat("Covariance between Weight and Wing:", cov_weight_wing, "\n")
```

```
## Covariance between Weight and Wing: 41174.39
```

```
cat("Correlation between Weight and Wing:", cor_weight_wing, "\n")
```

```
## Correlation between Weight and Wing: 0.9348575
```

1.6(Q2)

```
# Assuming you have a dataframe Hawks with columns Weight and Wing

a <- 2.4
b <- 7.1
c <- -1
d <- 3

# Creating the transformed variables
Hawks$Weight_transformed <- a * Hawks$Weight + b
Hawks$Wing_transformed <- c * Hawks$Wing + d

# Computing covariance and correlation for transformed variables
cov_transformed <- cov(Hawks$Weight_transformed, Hawks$Wing_transformed, use = "complete.obs")
cor_transformed <- cor(Hawks$Weight_transformed, Hawks$Wing_transformed, use = "complete.obs")

# Displaying the results
cat("Covariance of transformed variables:", cov_transformed, "\n")
```

```
## Covariance of transformed variables: -98818.54
```

```
cat("Correlation of transformed variables:", cor_transformed, "\n")
```

```
## Correlation of transformed variables: -0.9348575
```

2.1(Q1)

1. Random Experiment: A random experiment is an experiment or a process for which the outcome cannot be predicted with certainty.
2. Sample Space: The sample space, often denoted as S or Ω, refers to the set of all possible outcomes of a random experiment. It encompasses every conceivable result for the given experiment

3. Event: An event is any subset of the sample space. It represents a specific set of outcomes of a random experiment that we might be interested in.

2.1 (Q2)

2.2 (Q1)

2.2 (Q2)

2.2 (Q2)

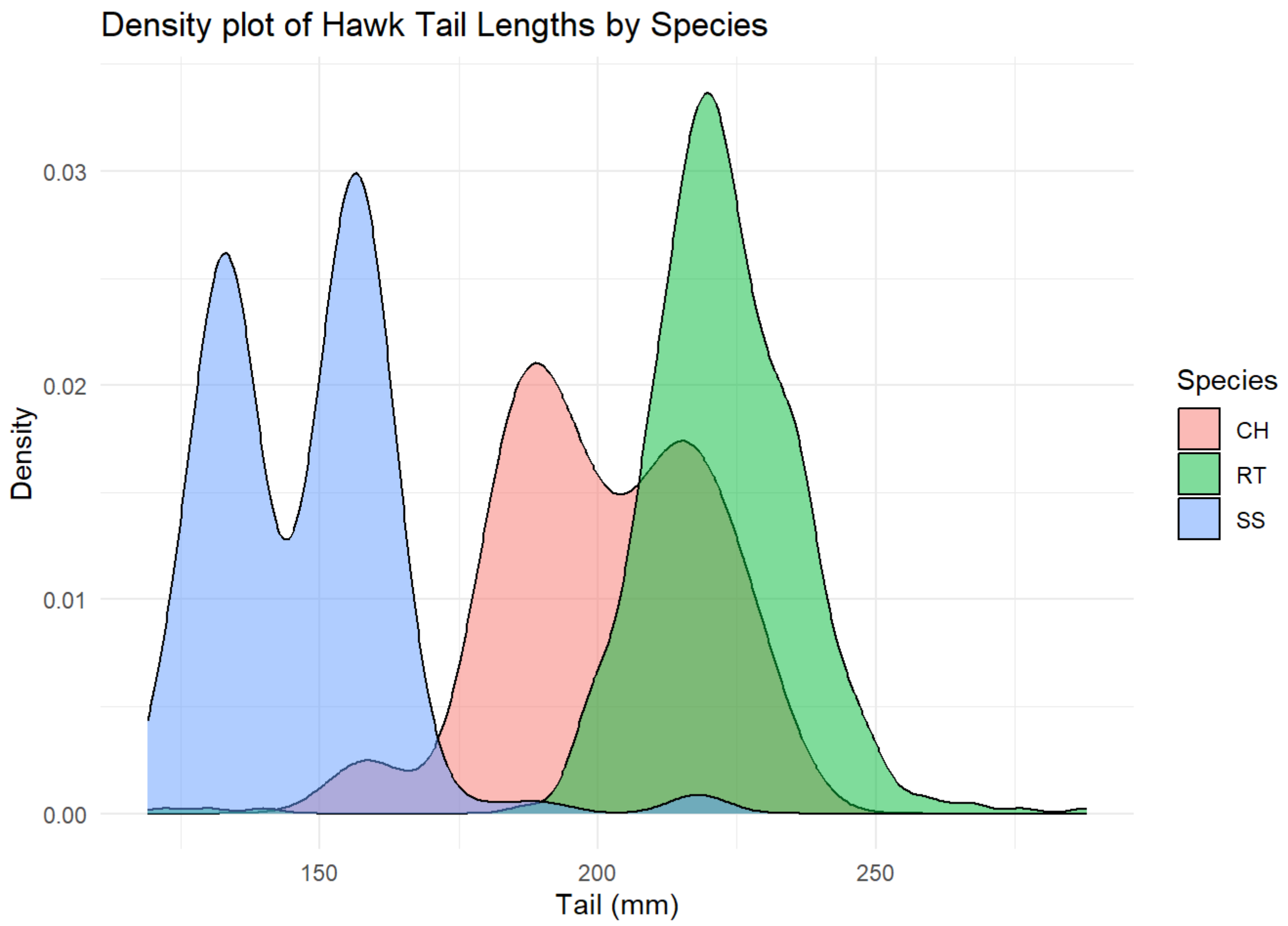
2.2 (Q3)

2.2 (Q4)

2.2 (Q5)

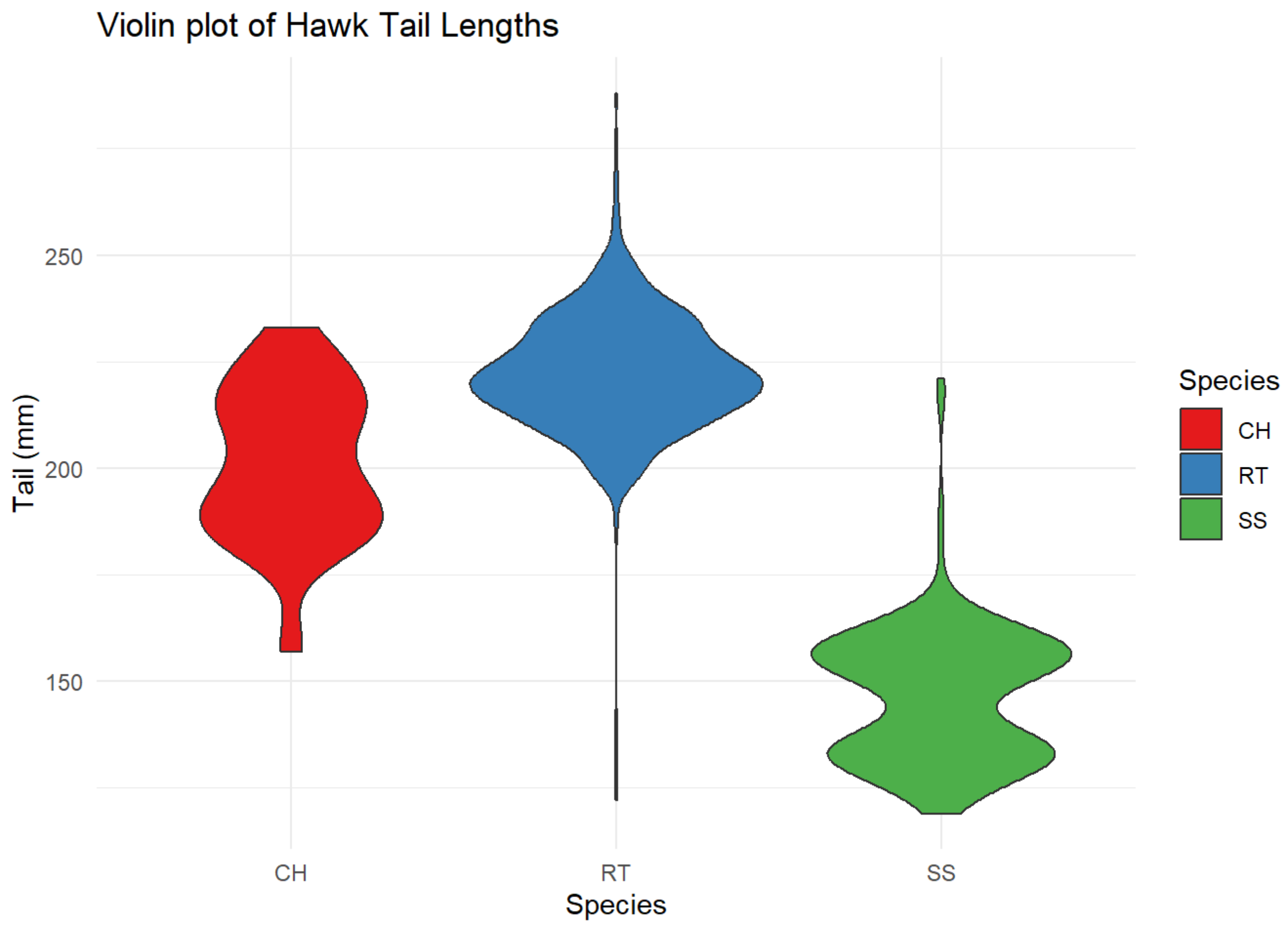
3 (Q1)

```
# Load the required libraries
library(ggplot2)

# Create the density plot
ggplot(data = Hawks, aes(x = Tail, fill = Species)) +
  geom_density(alpha = 0.5) +
  labs(title = ″Density plot of Hawk Tail Lengths by Species″,
       x = ″Tail (mm)″,
       y = ″Density″) +
  theme_minimal()
```



## 3 (Q2)

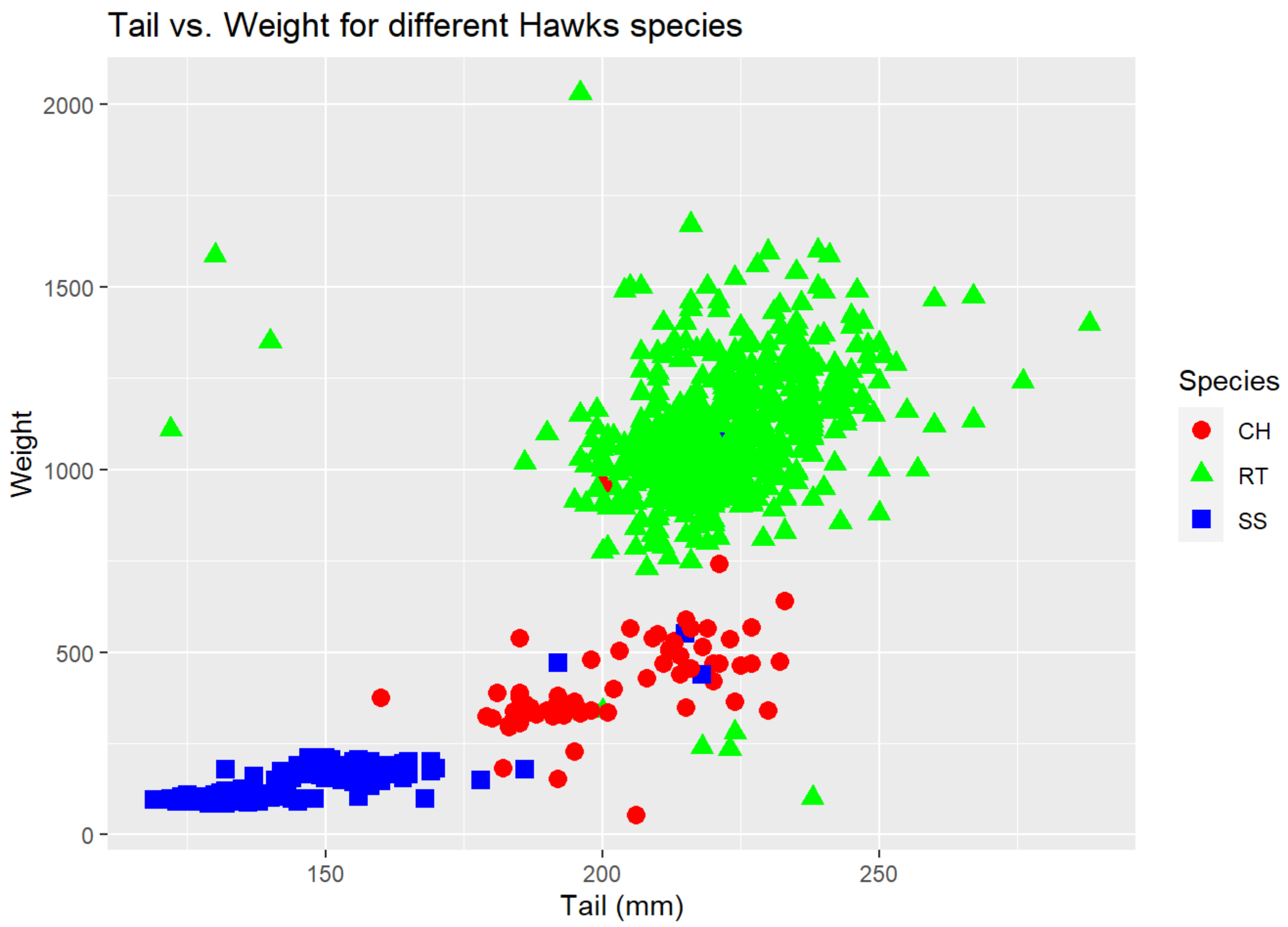
```
# Load the required libraries
library(ggplot2)

# Create the violin plot
ggplot(data = Hawks, aes(x = Species, y = Tail, fill = Species)) +
  geom_violin() +
  labs(title = ″Violin plot of Hawk Tail Lengths″,
       x = ″Species″,
       y = ″Tail (mm)″) +
  theme_minimal() +
  scale_fill_brewer(palette = ″Set1″)
```



## 3 (Q3)

```
# Define the custom shapes and colors for species
hawks_shapes <- c(CH = 16, RT = 17, SS = 15)  # 16: circle, 17: triangle, 15: square
hawks_colors <- c(CH = "red", RT = "green", SS = "blue")

# Plot
ggplot(Hawks, aes(x = Tail, y = Weight, shape = Species, color = Species)) +
  geom_point(aes(shape = Species, color = Species), size = 3) +
  scale_shape_manual(values = hawks_shapes) +
  scale_color_manual(values = hawks_colors) +
  labs(title = "Tail vs. Weight for different Hawks species",
       x = "Tail (mm)",
       y = "Weight",
       shape = "Species",
       color = "Species")
```

```
## Warning: Removed 10 rows containing missing values (`geom_point()`).
```



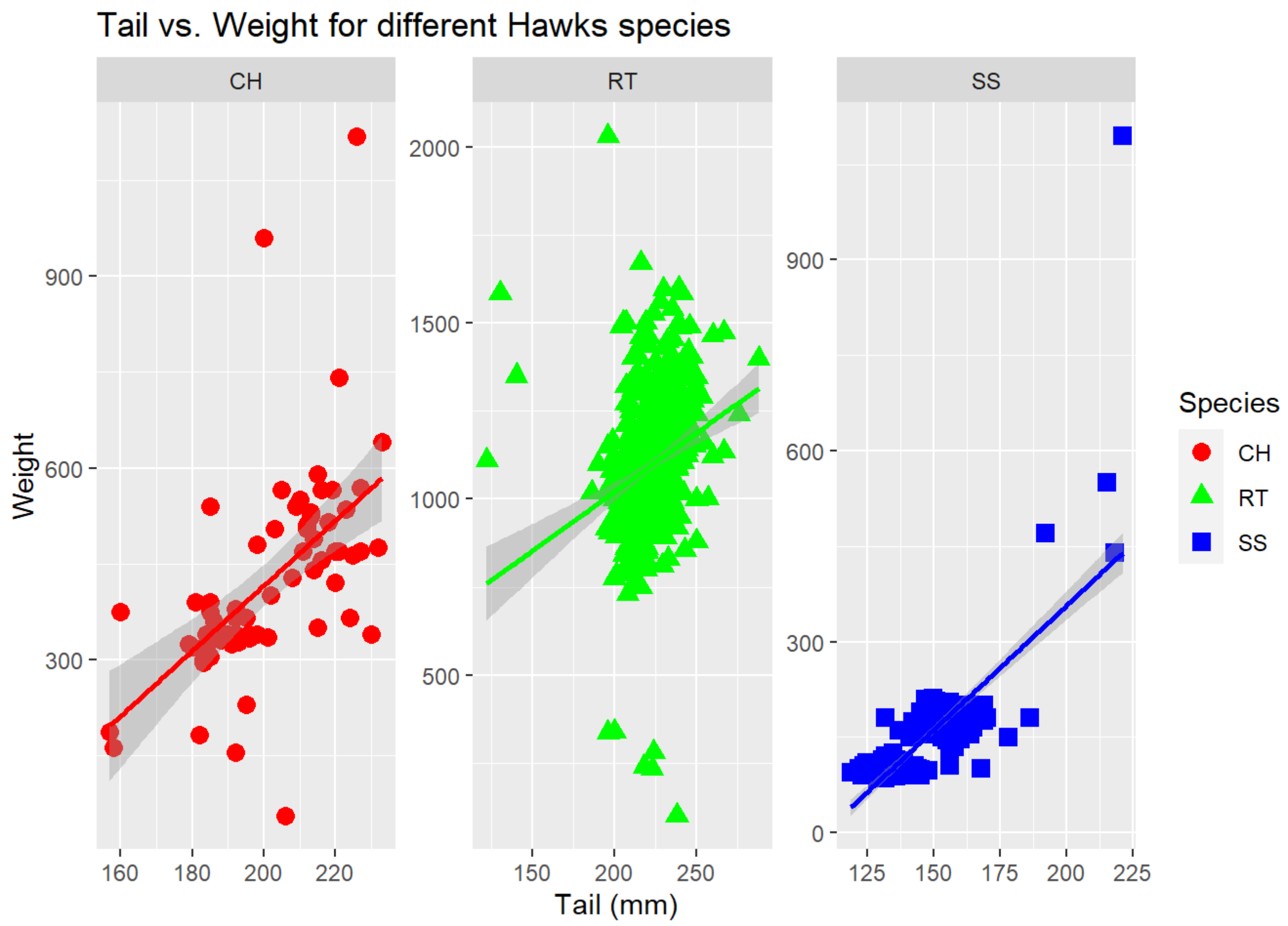
## 3 (Q4)

```
# Define the custom shapes and colors for species
hawks_shapes <- c(CH = 16, RT = 17, SS = 15)  # 16: circle, 17: triangle, 15: square
hawks_colors <- c(CH = "red", RT = "green", SS = "blue")

# Plot
ggplot(Hawks, aes(x = Tail, y = Weight, shape = Species, color = Species)) +
  geom_point(aes(shape = Species, color = Species), size = 3) +
  geom_smooth(method = "lm", se = TRUE, aes(color = Species), show.legend = FALSE) +
  scale_shape_manual(values = hawks_shapes) +
  scale_color_manual(values = hawks_colors) +
  facet_wrap(~ Species, scales = "free") +
  labs(title = "Tail vs. Weight for different Hawks species",
       x = "Tail (mm)",
       y = "Weight",
       shape = "Species",
       color = "Species")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 10 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 10 rows containing missing values (`geom_point()`).
```



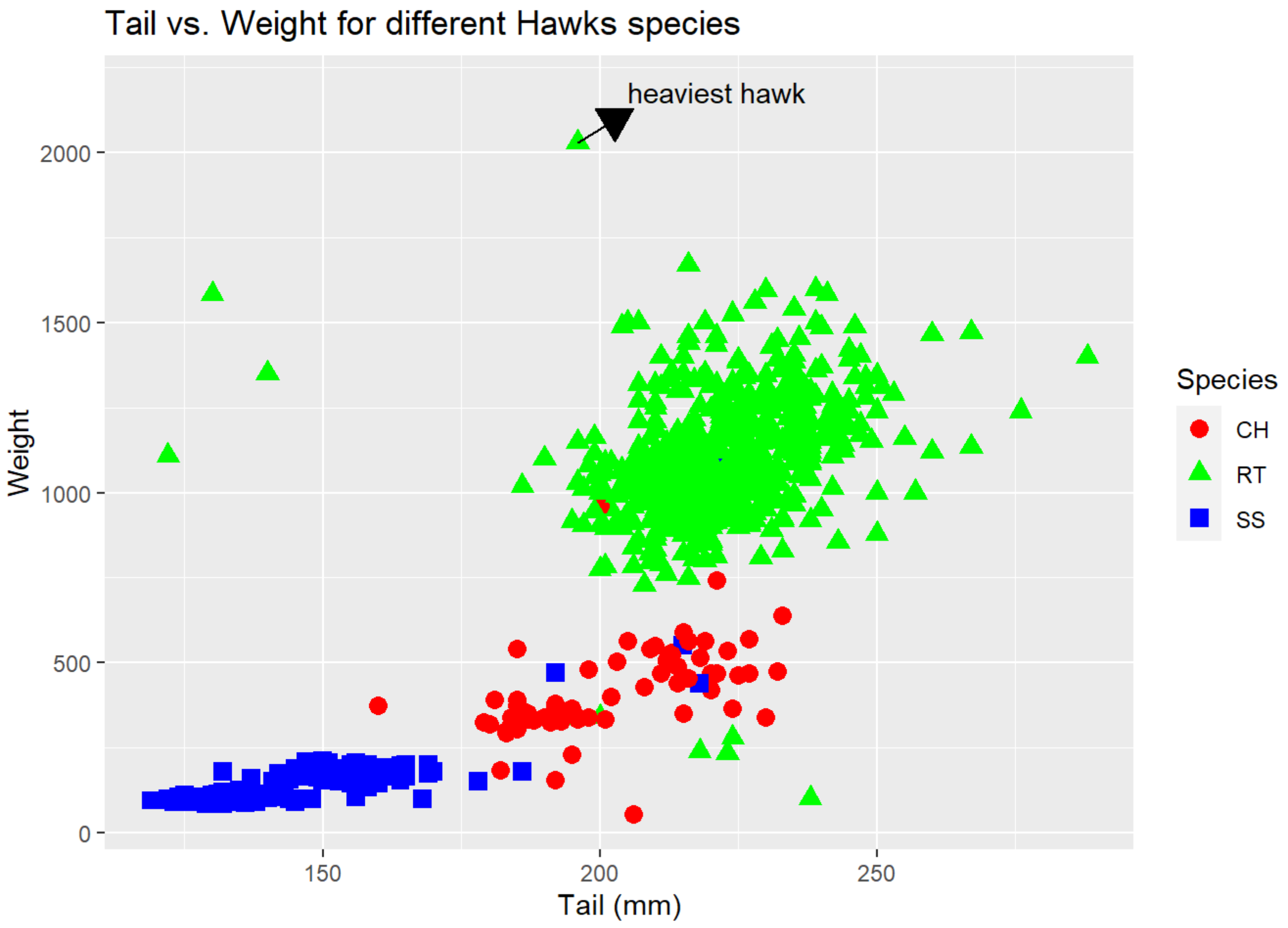
….. ## 3 (Q5)

```
# Find the heaviest hawk
heaviest_hawk <- Hawks %>%
  filter(Weight == max(Weight, na.rm = TRUE)) %>%
  select(Tail, Weight, Species) %>%
  top_n(1)
```

```
## Selecting by Species
```

```
tail_heaviest <- heaviest_hawk$Tail
weight_heaviest <- heaviest_hawk$Weight

# Define the custom shapes and colors for species
hawks_shapes <- c(CH = 16, RT = 17, SS = 15)  # 16: circle, 17: triangle, 15: square
hawks_colors <- c(CH = "red", RT = "green", SS = "blue")

# Plot
ggplot(Hawks, aes(x = Tail, y = Weight, shape = Species, color = Species)) +
  geom_point(aes(shape = Species, color = Species), size = 3) +
  scale_shape_manual(values = hawks_shapes) +
  scale_color_manual(values = hawks_colors) +
  labs(title = "Tail vs. Weight for different Hawks species",
       x = "Tail (mm)",
       y = "Weight",
       shape = "Species",
       color = "Species") +
  geom_segment(aes(x = tail_heaviest, xend = tail_heaviest + 10,
                   y = weight_heaviest, yend = weight_heaviest + 100),
               arrow = arrow(type = "closed", length = unit(0.2, "inches")), color = "black") +
  annotate("text", x = tail_heaviest + 25, y = weight_heaviest + 150,
           label = "heaviest hawk", color = "black")
```

```
## Warning: Removed 10 rows containing missing values (`geom_point()`).
```


Tail vs. Weight for different Hawks species
heaviest hawk
Weight
Tail (mm)
Species
CH
RT
SS