Assignment 3

Keli Niu

2024-10-01

1. Visualisation

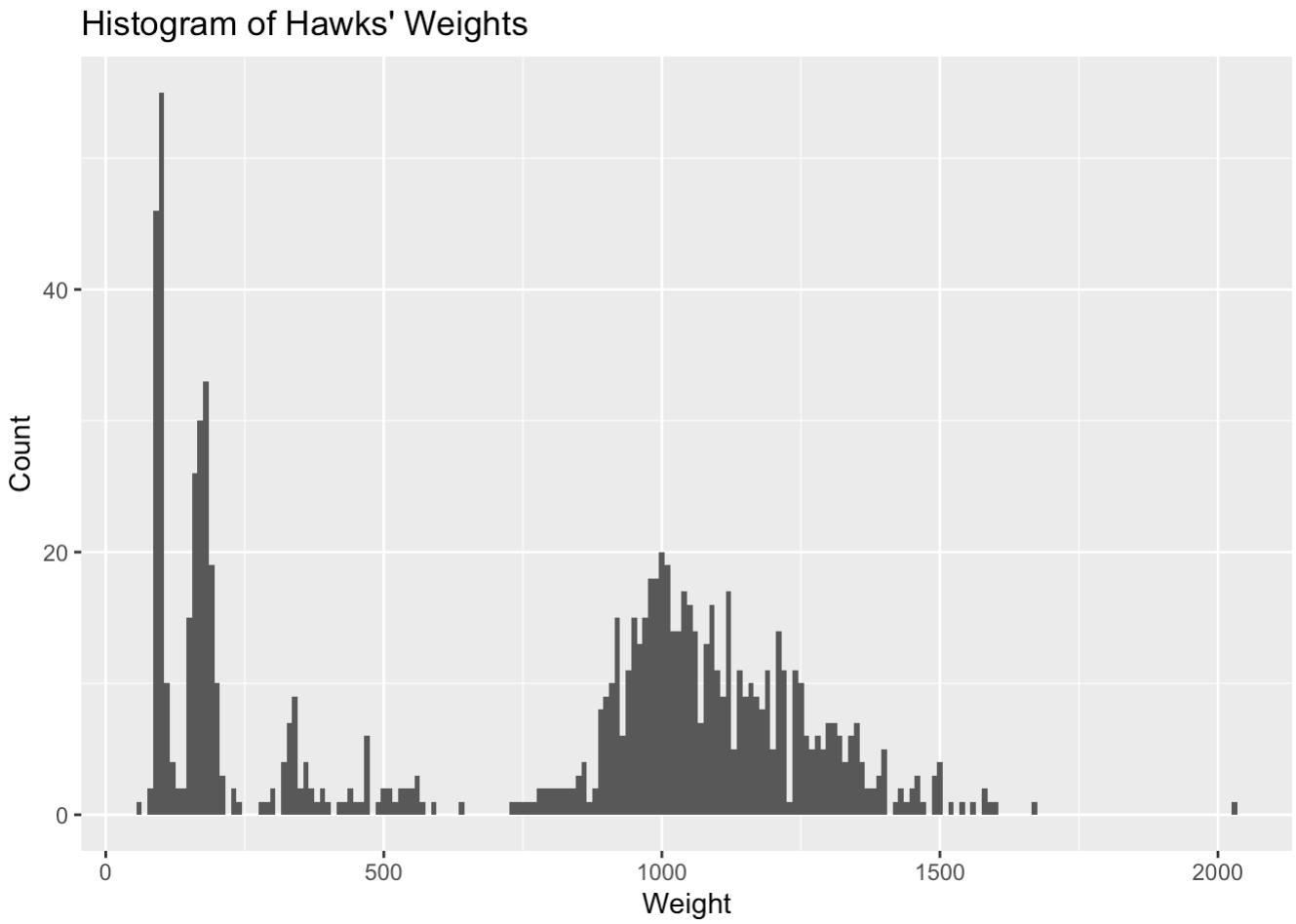
1 (Q1)

```
## [1] 897   9
```

```
##   Age Day Month Year CaptureTime Species Wing Weight Tail
## 1   I  19     9 1992       13:30      RT  385    920  219
## 2   I  22     9 1992       10:30      RT  376    930  221
## 3   I  23     9 1992       12:45      RT  381    990  235
## 4   I  23     9 1992       10:50      CH  265    470  220
## 5   I  27     9 1992       11:15      SS  205    170  157
```

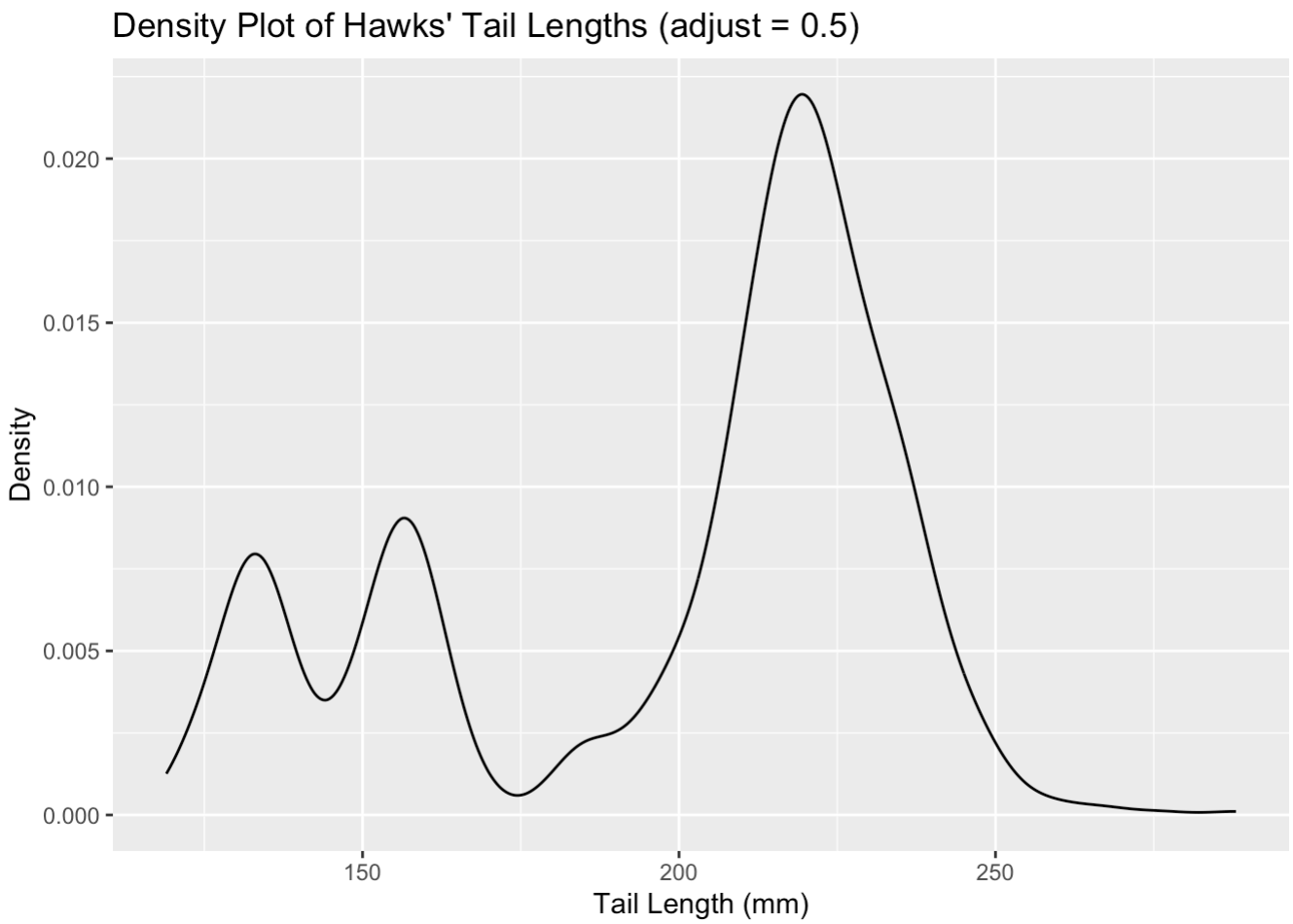
1 (Q2)

```
knitr::opts_chunk$set(echo = TRUE)

library(ggplot2)
ggplot(hawksSmall, aes(x = Weight)) +
  geom_histogram(binwidth = 10) +
  labs(title = "Histogram of Hawks' Weights", x = "Weight", y = "Count")
```

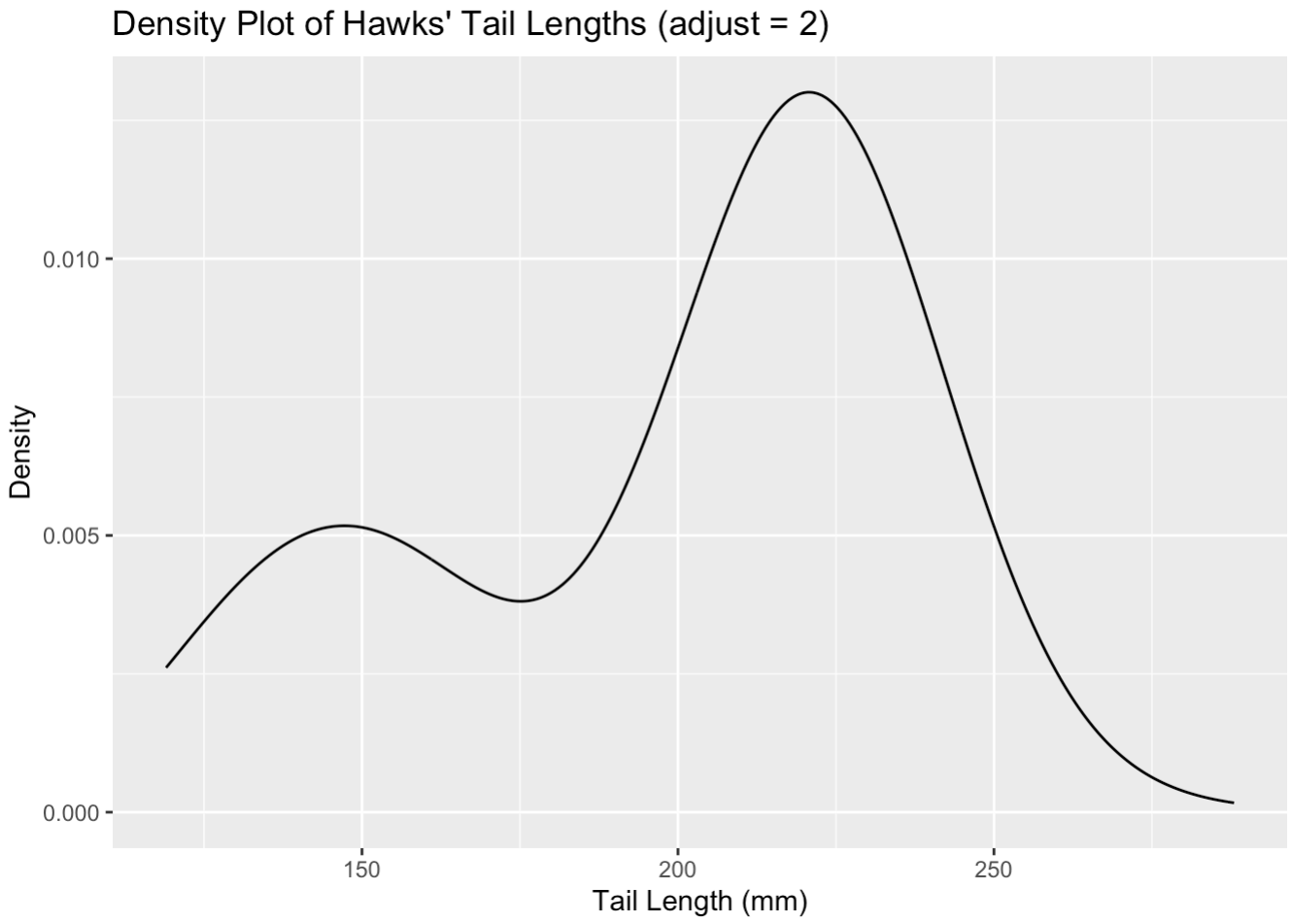


1 (Q3)

```
knitr::opts_chunk$set(echo = TRUE)
plot1 <- ggplot(hawksSmall, aes(x = Tail)) +
  geom_density(adjust = 0.5) +
  labs(title = "Density Plot of Hawks' Tail Lengths (adjust = 0.5)",
       x = "Tail Length (mm)", y = "Density")

plot2 <- ggplot(hawksSmall, aes(x = Tail)) +
  geom_density(adjust = 2) +
  labs(title = "Density Plot of Hawks' Tail Lengths (adjust = 2)",
       x = "Tail Length (mm)", y = "Density")

plot1
```


Density Plot of Hawks' Tail Lengths (adjust = 0.5)
Density
0.020
0.015
0.010
0.005
0.000
150
200
250
Tail Length (mm)

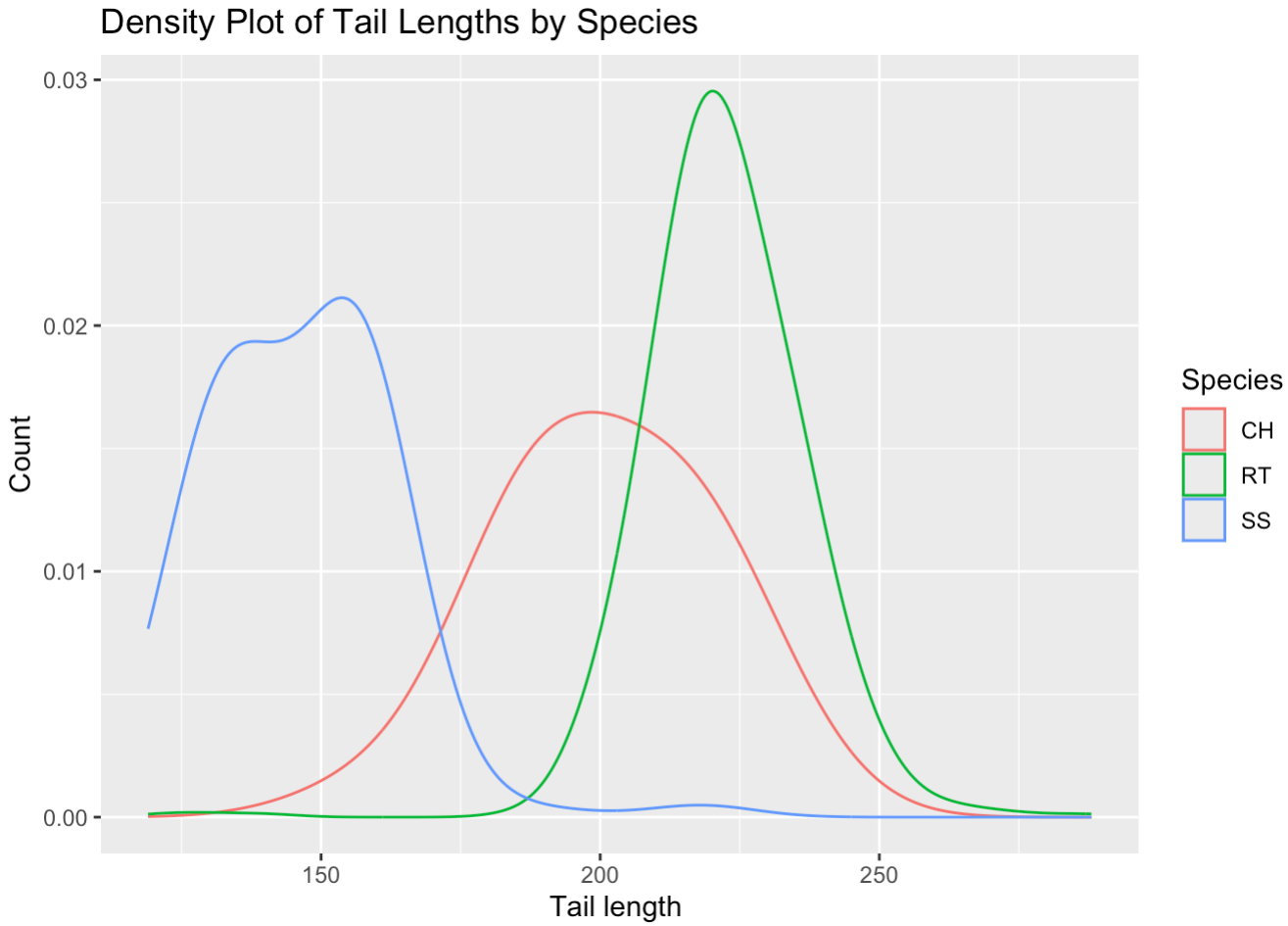
```
plot2
```


Density Plot of Hawks' Tail Lengths (adjust = 2)
Density
0.010
0.005
0.000
150
200
250
Tail Length (mm)

The main difference between the two plots is the level of smoothness. The second plot is smoother, and compared to the first plot, it loses some of the finer details. The first plot shows three distinct groupings, so it has three modes. The second plot, on the other hand, has two groupings, resulting in two modes.

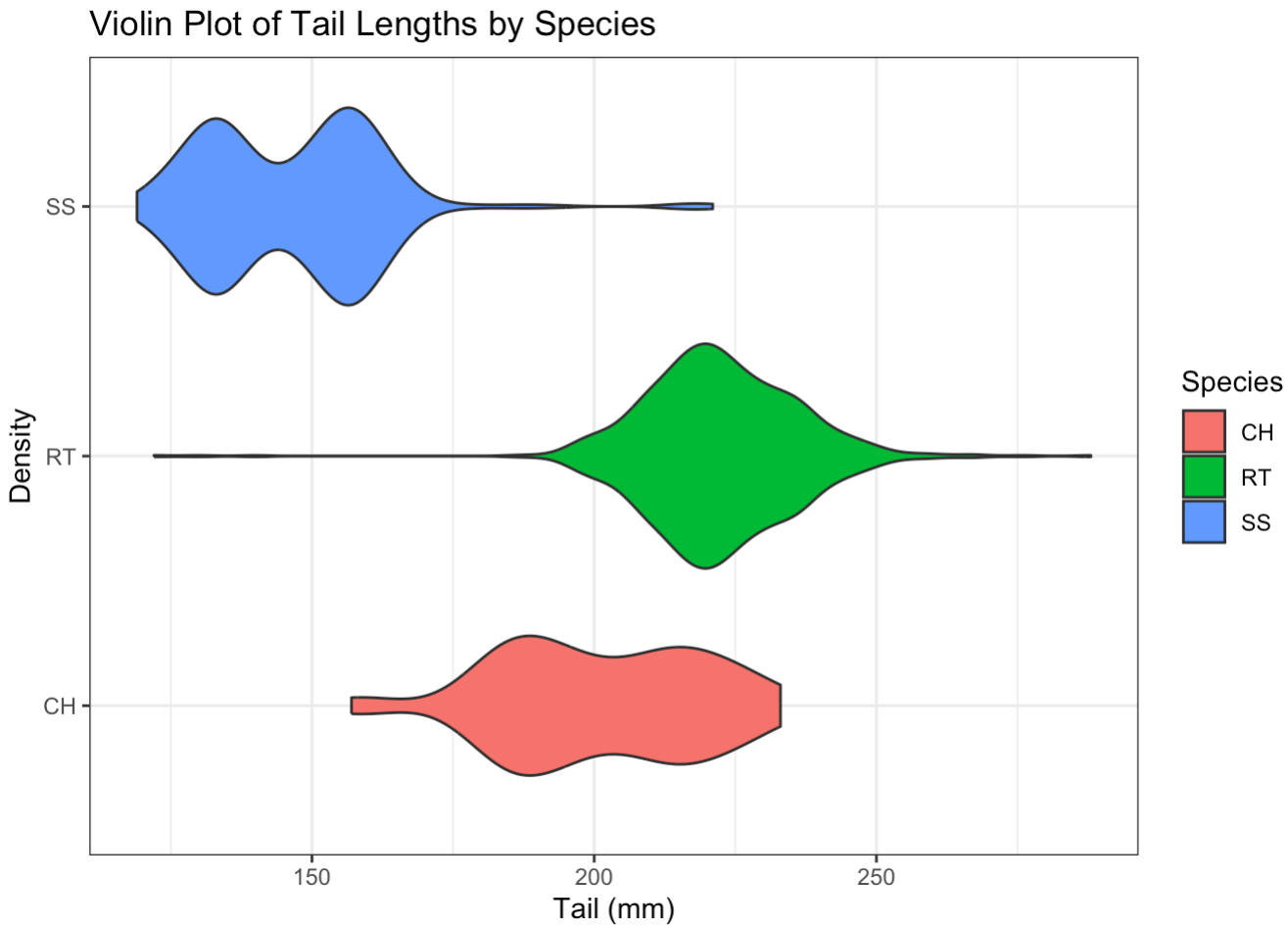
1 (Q4)

```
knitr::opts_chunk$set(echo = TRUE)
ggplot(hawksSmall, aes(x = Tail, color = Species)) +
  geom_density(adjust = 2) +
  labs(title = "Density Plot of Tail Lengths by Species",
       x = "Tail length ", y = "Count")
```



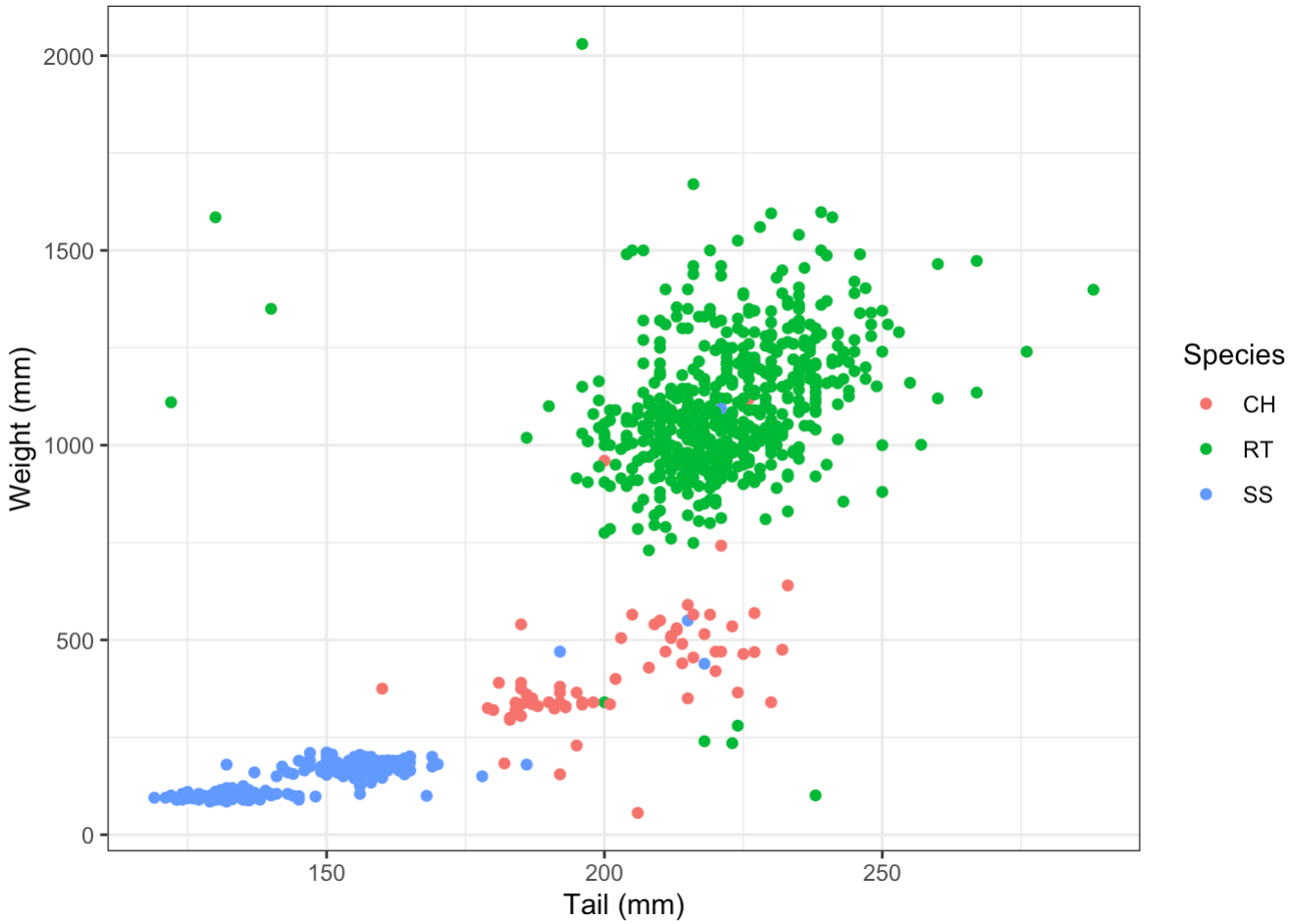
1 (Q5)

```
knitr::opts_chunk$set(echo = TRUE)
ggplot(hawksSmall, aes(x = Tail, y = Species, fill = Species)) +
  geom_violin() + theme_bw()+
  labs(title = "Violin Plot of Tail Lengths by Species",
       x = "Tail (mm)", y = "Density")
```



1 (Q6)

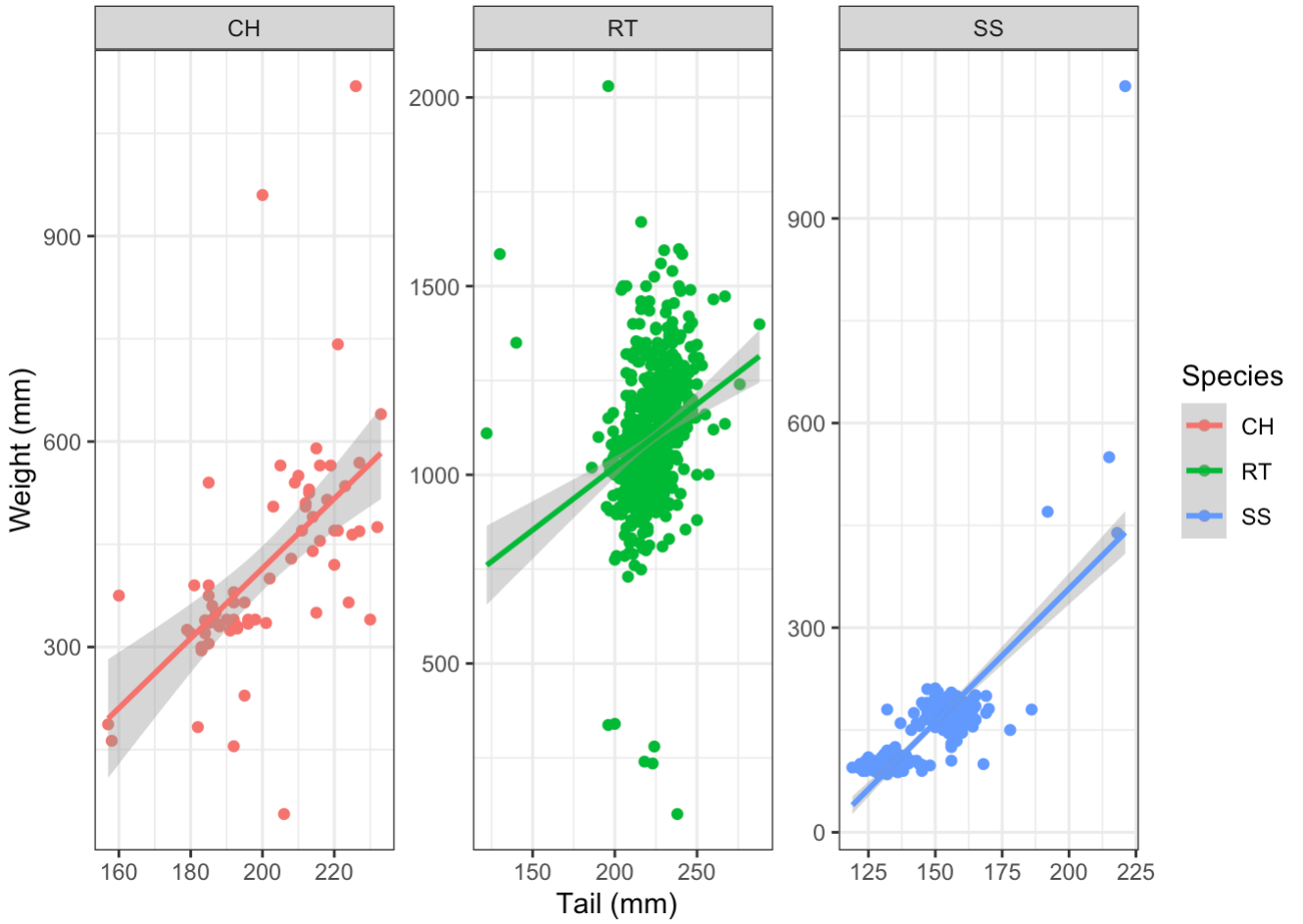
```
knitr::opts_chunk$set(echo = TRUE)
ggplot(hawksSmall, aes(x = Tail, y = Weight, colour = Species)) +
  geom_point() + theme_bw()+
  labs(x = "Tail (mm)", y = "Weight (mm)")
```



1. There are three aesthetics: Tail(x-axis), Weight(y-axis), Species(color)
2. The glyphs within this plot are points, which repersent the relationship between the Tail and the Weight.
3. position, colour

1 (Q7)

```
knitr::opts_chunk$set(echo = TRUE)
ggplot(hawksSmall, aes(x = Tail, y = Weight, colour = Species)) +
  geom_point() +
  geom_smooth(method=lm)+
  facet_wrap(~Species, scales = "free")+
  theme_bw()+
  labs(x = "Tail (mm)", y = "Weight (mm)")
```



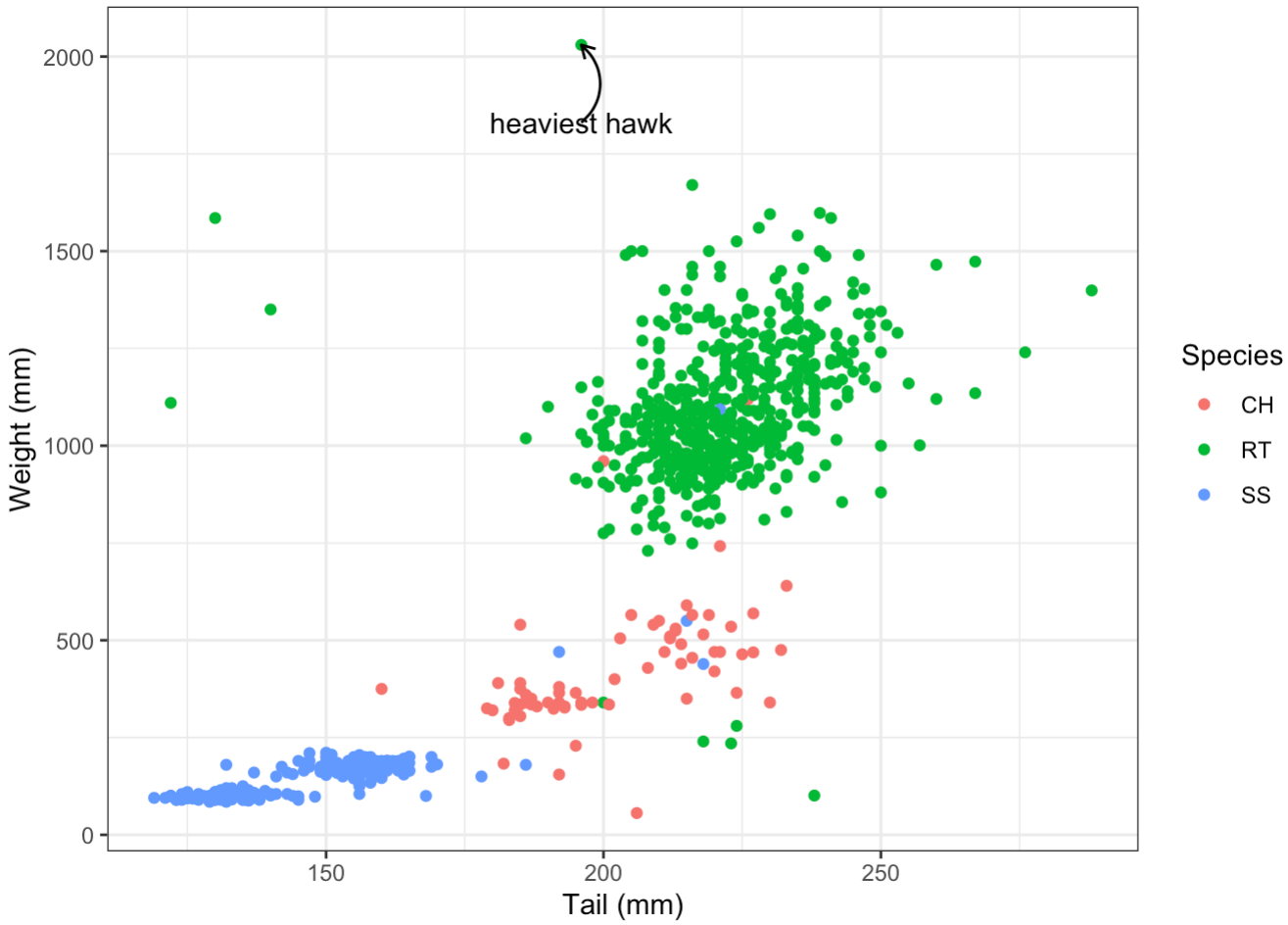
1. Shape, colour, position
2. There is a positive correlation between the weight and tail length of hawks in all three species, meaning that the longer the tail, the heavier the hawk. Among them, the CH species has a more dispersed distribution, with a weaker correlation, indicating that other factors may be influencing the weight. Additionally, both RT and SS species show a stronger correlation, with RT having a steeper slope, indicating that tail length has a greater impact on weight.

1 (Q8)

```
knitr::opts_chunk$set(echo = TRUE)
hawkHeaviest<-hawksSmall%>%
  filter(Weight==max(Weight))%>%
  select(Weight, Tail, Species)

hawkHeaviest
```

```
##   Weight Tail Species
## 1   2030  196      RT
```

```
ggplot(hawksSmall, aes(x = Tail, y = Weight, colour = Species)) +
  geom_point() + theme_bw()+
  labs(x = "Tail (mm)", y = "Weight (mm)")+

annotate("text",x = hawkHeaviest$Tail, y = hawkHeaviest$Weight-200, label = "heaviest
hawk")+
annotate("curve", x = hawkHeaviest$Tail, y = hawkHeaviest$Weight-200 ,xend = hawkHeav
iest$Tail, yend = hawkHeaviest$Weight,curvature = 0.5, arrow = arrow(length = unit(0.
2, "cm")))
```



2. Finite probability spaces

2.1 (Q1)

$$P(Z = z) = \binom{22}{z} \cdot \left(\frac{3}{10}\right)^{z} \cdot \left(\frac{7}{10}\right)^{22-z}$$

2.1 (Q2)

```
knitr::opts_chunk$set(echo = TRUE)
prob_red_spheres<-function(z){
  n<-22
  p<-3/10

  probability<-choose(n,z)*(p^z)*((1-p)^(n-z))
  return(probability)

}

prob_red_spheres(10)
```

```
## [1] 0.05285129
```

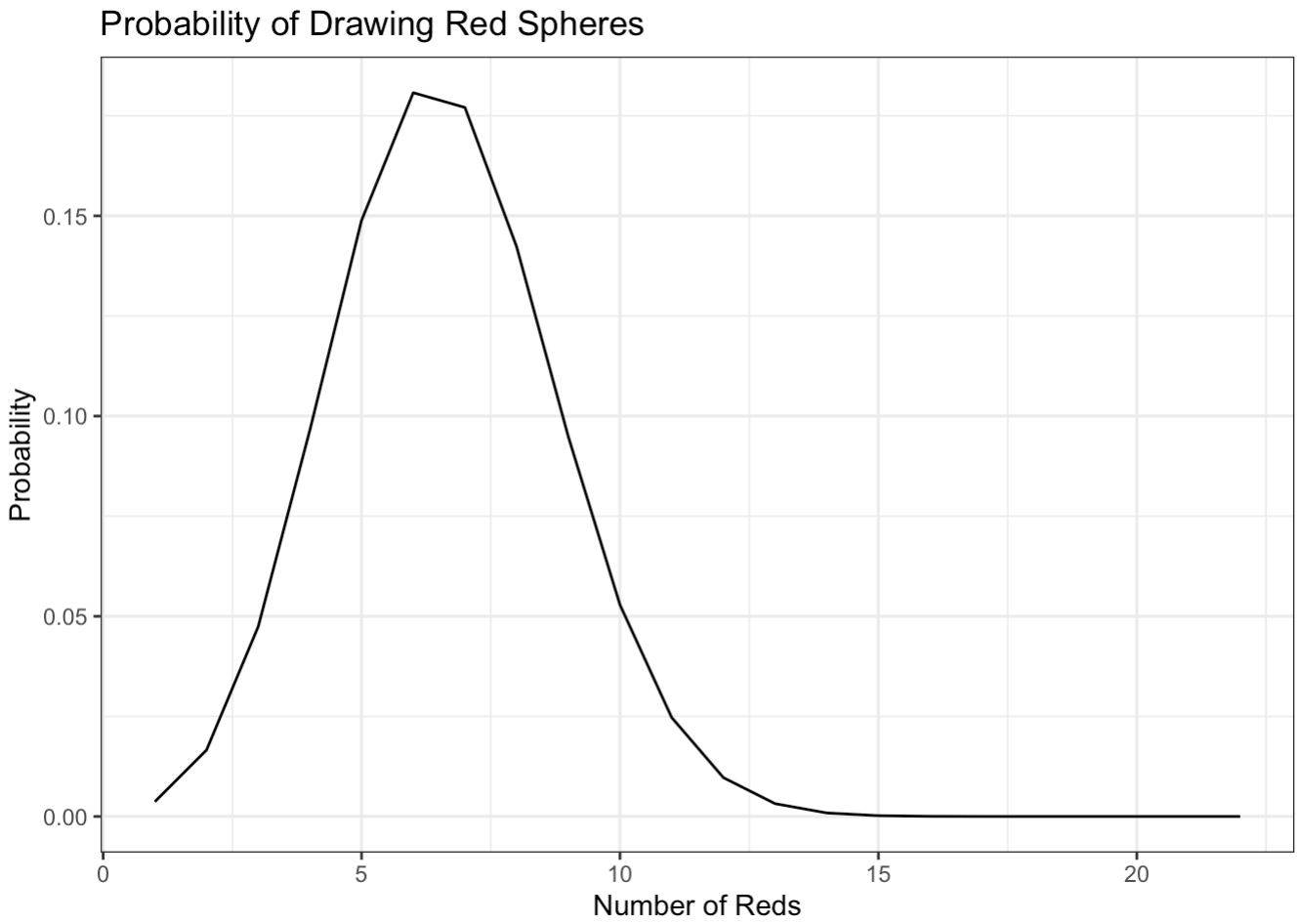
2.1 (Q3)

```
knitr::opts_chunk$set(echo = TRUE)

num_reds <- 1:22
prob <- numeric(length(num_reds))
for (i in num_reds) {
  prob[i] <- prob_red_spheres(i)
}

prob_by_num_reds<-data.frame(num_reds,prob)

head(prob_by_num_reds,3)
```

```
##   num_reds        prob
## 1        1 0.003686403
## 2        2 0.016588812
## 3        3 0.047396606
```

2.1 (Q4)

```
knitr::opts_chunk$set(echo = TRUE)

ggplot(prob_by_num_reds, aes(x = num_reds, y = prob)) +
  geom_line() +
  labs(title = "Probability of Drawing Red Spheres",
       x = "Number of Reds",
       y = "Probability") + theme_bw()
```



2.1 (Q5)

```
knitr::opts_chunk$set(echo = TRUE)

sample(10, 22, replace=TRUE)
```

```
##  [1]  3  6  3  7  8  3  1  4  7  1  6  4  4 10  5 10  2  7  5  4  7  1
```

```
## case 1: Setting the random seed just once
set.seed(0)
for(i in 1:5){
 print(sample(100,5,replace=FALSE))
 # The result may well differ every time
}
```

```
## [1] 14 68 39  1 34
## [1] 87 43 14 82 59
## [1] 51 97 85 21 54
## [1] 74  7 73 79 85
## [1]  37  89 100  34  99
```

```
## case 2: Resetting the random seed every time
set.seed(1)
print(sample(100,5,replace=FALSE))
```

```
## [1] 68 39  1 34 87
```

```
set.seed(1)
print(sample(100,5,replace=FALSE))
```

```
## [1] 68 39  1 34 87
```

```
set.seed(1)
print(sample(100,5,replace=FALSE))
```

```
## [1] 68 39  1 34 87
```

```
# The result should not change
## case 3: reproducing case 1 if we set a random seed at the beginning.
set.seed(0)
for(i in 1:5){
 print(sample(100,5,replace=FALSE))
} # The result will be 5 samples exactly the same as in case 1 (why?).
```

```
## [1] 14 68 39  1 34
## [1] 87 43 14 82 59
## [1] 51 97 85 21 54
## [1] 74  7 73 79 85
## [1]  37  89 100  34  99
```

```
itermap <- function(.x, .f) {
 result <- list()
 for (item in .x) {
 result <- c(result, list(.f(item)))
 }
 return(result)
}
itermap( c(1,2,3), function(x){ return(c(x,x^2)) } )
```

```
## [[1]]
## [1] 1 1
##
## [[2]]
## [1] 2 4
##
## [[3]]
## [1] 3 9
```

```
itermap_dbl <- function(.x, .f) {
 result <- numeric(length(.x))
 for (i in 1:length(.x)) {
 result[i] <- .f(.x[[i]])
 }
 return(result)
}
itermap_dbl( c(1,2,3), function(x){ return(x^3) } )
```

```
## [1]  1  8 27
```

```
num_trials<-1000 # set the number of trials
set.seed(0) # set the random seed
sampling_with_replacement_simulation<-data.frame(trial=1:num_trials)%>%
  mutate(sample_balls = itermap(.x=trial, function(x){sample(10,22,replace = TRUE)}))
# generate collection of num_trials simulations

sampling_with_replacement_simulation <- sampling_with_replacement_simulation%>%
  mutate(num_reds = sapply(sample_balls, function(x) sum(x <= 3)))


head(sampling_with_replacement_simulation)
```

```
##   trial                                                        sample_balls
## 1     1   9, 4, 7, 1, 2, 7, 2, 3, 1, 5, 5, 10, 6, 10, 7, 9, 5, 5, 9, 9, 5, 5
## 2     2 2, 10, 9, 1, 4, 3, 6, 10, 10, 6, 4, 4, 10, 9, 7, 6, 9, 8, 9, 7, 8, 6
## 3     3   10, 7, 3, 10, 6, 8, 2, 2, 6, 6, 1, 3, 3, 8, 6, 7, 6, 8, 7, 1, 4, 8
## 4     4  9, 9, 7, 4, 7, 6, 1, 5, 6, 1, 9, 7, 7, 3, 6, 2, 10, 10, 7, 3, 2, 10
## 5     5 1, 10, 10, 8, 10, 5, 7, 8, 5, 6, 8, 1, 3, 10, 3, 1, 6, 6, 4, 9, 5, 1
## 6     6     3, 6, 3, 7, 3, 3, 1, 9, 2, 8, 6, 1, 2, 7, 7, 4, 9, 8, 3, 5, 3, 4
##   num_reds
## 1        5
## 2        3
## 3        7
## 4        6
## 5        6
## 6       10
```

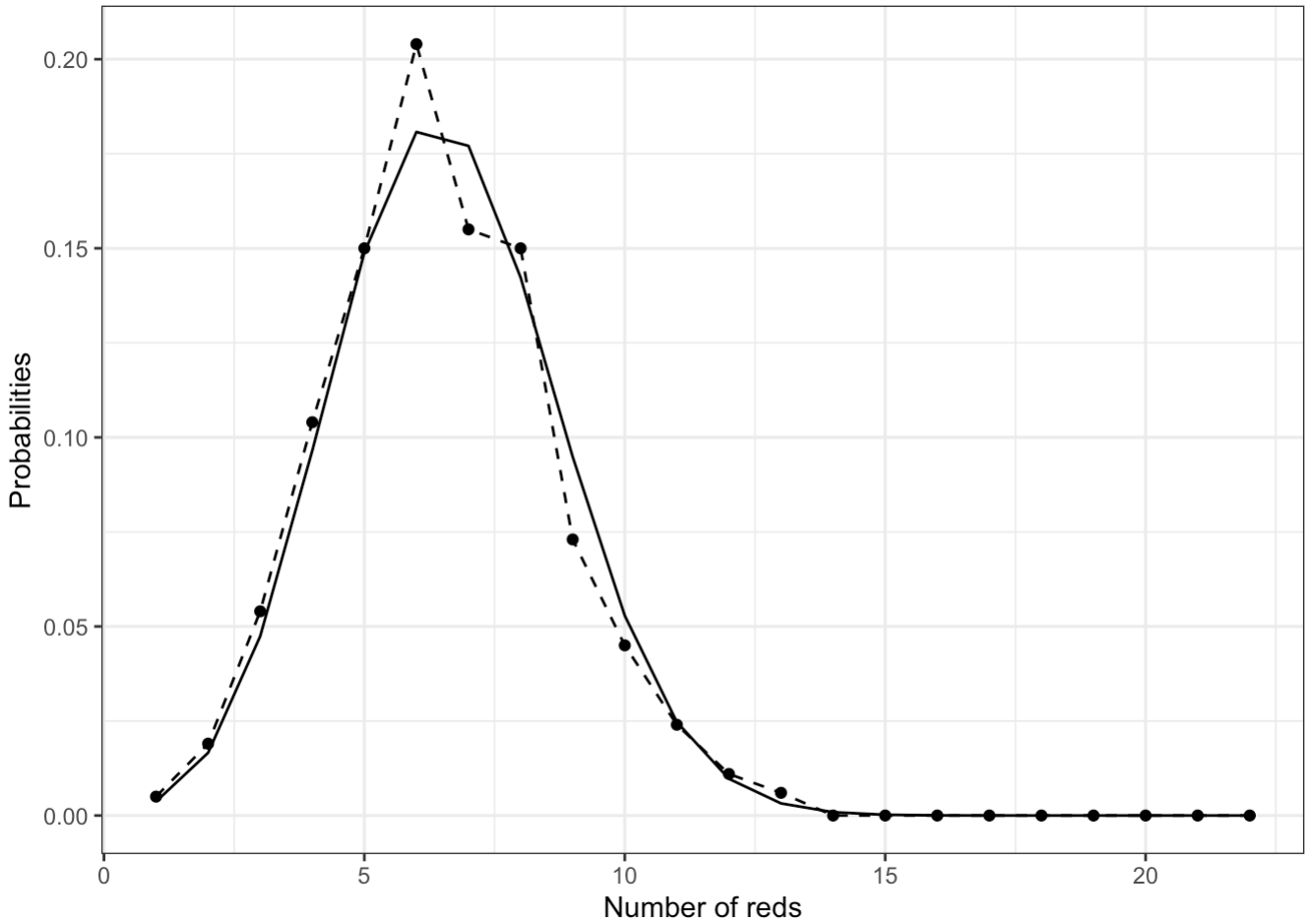
2.1 (Q6)

```
knitr::opts_chunk$set(echo = TRUE)

 num_reds_in_simulation<-sampling_with_replacement_simulation %>%
 pull(num_reds)
# we extract a vector corresponding to the number of reds in each trial
prob_by_num_reds<-prob_by_num_reds %>%
 mutate(predicted_prob=itermap_dbl(.x=num_reds, function(.x)
sum(num_reds_in_simulation==.x))/num_trials)
# add a column which gives the number of trials with a given number of reds
head(prob_by_num_reds)
```

```
##   num_reds        prob predicted_prob
## 1        1 0.003686403          0.005
## 2        2 0.016588812          0.019
## 3        3 0.047396606          0.054
## 4        4 0.096485948          0.104
## 5        5 0.148864035          0.150
## 6        6 0.180763470          0.204
```

2.1 (Q7)

```
knitr::opts_chunk$set(echo = TRUE)

prob_by_num_reds %>%
 rename(TheoreticalProbability=prob,
EstimatedProbability=predicted_prob) %>%
 ggplot() + geom_line(aes(x=num_reds, y=TheoreticalProbability)) +
 geom_line(aes(x=num_reds, y=EstimatedProbability), linetype='dashed')+
 geom_point(aes(x=num_reds, y=EstimatedProbability)) +
 theme_bw() + xlab("Number of reds") + ylab("Probabilities")
```



2.2 (Q1)

```
knitr::opts_chunk$set(echo = TRUE)


set.seed(0)


num_trials <- 1000

sample_size <- 10
num_balls <- 100

balls <- c(rep("red", 20), rep("blue", 20), rep("green", 60))

itermap_dbl <- function(.x, .f) {
  result <- numeric(length(.x))
  for (i in 1:length(.x)) {
    result[i] <- .f(.x[[i]])
  }
  return(result)
}

sampling_without_replacement_simulation <- data.frame(trial = 1:num_trials) %>%
  mutate(sample_balls = itermap(trial, function(x) sample(balls, sample_size, replace
= FALSE))) %>%
  mutate(num_reds = itermap_dbl(sample_balls, function(x) sum(x == "red")),
         num_blues = itermap_dbl(sample_balls, function(x) sum(x == "blue")),
         num_greens = itermap_dbl(sample_balls, function(x) sum(x == "green"))) %>%
  mutate(missing_color = pmin(num_reds, num_blues, num_greens) == 0)

prob_missing_color <- mean(sampling_without_replacement_simulation$missing_color)

prob_missing_color
```

```
## [1] 0.187
```