Assignment 5

Keli Niu

2024-10-15

1. Exploratory data analysis

1.1 (Q1)

```
head(Hawks)
```

```
##   Month Day Year CaptureTime ReleaseTime  BandNumber Species Age Sex Wing
## 1     9  19 1992       13:30               877-76317      RT   I       385
## 2     9  22 1992       10:30               877-76318      RT   I       376
## 3     9  23 1992       12:45               877-76319      RT   I       381
## 4     9  23 1992       10:50               745-49508      CH   I   F  265
## 5     9  27 1992       11:15              1253-98801      SS   I   F  205
## 6     9  28 1992       11:25             1207-55910      RT   I       412
##   Weight Culmen Hallux Tail StandardTail Tarsus WingPitFat KeelFat Crop
## 1    920   25.7   30.1  219           NA     NA         NA      NA   NA
## 2    930     NA     NA  221           NA     NA         NA      NA   NA
## 3    990   26.7   31.3  235           NA     NA         NA      NA   NA
## 4    470   18.7   23.5  220           NA     NA         NA      NA   NA
## 5    170   12.5   14.3  157           NA     NA         NA      NA   NA
## 6   1090   28.5   32.2  230           NA     NA         NA      NA   NA
```

```
HawksTail <- Hawks[["Tail"]]
head(HawksTail)
```

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

```
mean_value <- mean(HawksTail)
median_value <- median(HawksTail)
mean_value
```

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

```
median_value
```

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

1.2 (Q1)

```
hawks_summary <- Hawks %>%
  summarise(
    Wing_mean = mean(Wing, na.rm = TRUE),
    Wing_t_mean = mean(Wing, trim = 0.5, na.rm = TRUE),
    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),
    Weight_med = median(Weight, na.rm = TRUE)
  )

hawks_summary
```

```
##   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)

```
hawks_grouped_summary <-Hawks%>%
  group_by(Species)%>%
  summarise(
    Wing_mean = mean(Wing, na.rm = TRUE),
    Wing_t_mean = mean(Wing, trim = 0.5, na.rm = TRUE),
    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),
    Weight_med = median(Weight, na.rm = TRUE)
  )

hawks_grouped_summary
```

```
## # 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)

The sample mean after the linear transformation is given by the formula:

$$\tilde{A} = aA + b$$

where $\tilde{A}$ is the transformed sample mean, A is the original sample mean, and a, b are constants.

```
a <- 2
b <- 3
HawksTail_transformed <- a * HawksTail + b

mean_transformed <- mean(HawksTail_transformed, na.rm = TRUE)

mean_theoretical <- a * mean_value + b

mean_transformed
```

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

```
mean_theoretical
```

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

1.3 (Q2)

The sample variance and standard deviation after the linear transformation are given by the formulas: - Variance:

$$\tilde{p} = a^2 p$$

- Standard deviation:

$$\tilde{q} = |a|q$$

where $\tilde{p}$ and $\tilde{q}$ are the transformed sample variance and standard deviation, respectively. p and q are the original sample variance and standard deviation, and a is a constant. The constant b does not affect the variance or standard deviation.

```
a <- 2
b <- 3
var_original <- var(HawksTail, na.rm = TRUE)
sd_original <- sd(HawksTail, na.rm = TRUE)

HawksTail_transformed <- a * HawksTail + b

var_transformed <- var(HawksTail_transformed, na.rm = TRUE)
sd_transformed <- sd(HawksTail_transformed, na.rm = TRUE)

#proven
var_theoretical <- a^2 * var_original
sd_theoretical <- abs(a) * sd_original

#variance
var_transformed
```

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

```
var_theoretical
```

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

```
#standard deviation
sd_transformed
```

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

```
sd_theoretical
```

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

1.4 (Q1)

```
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))
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))
}
```

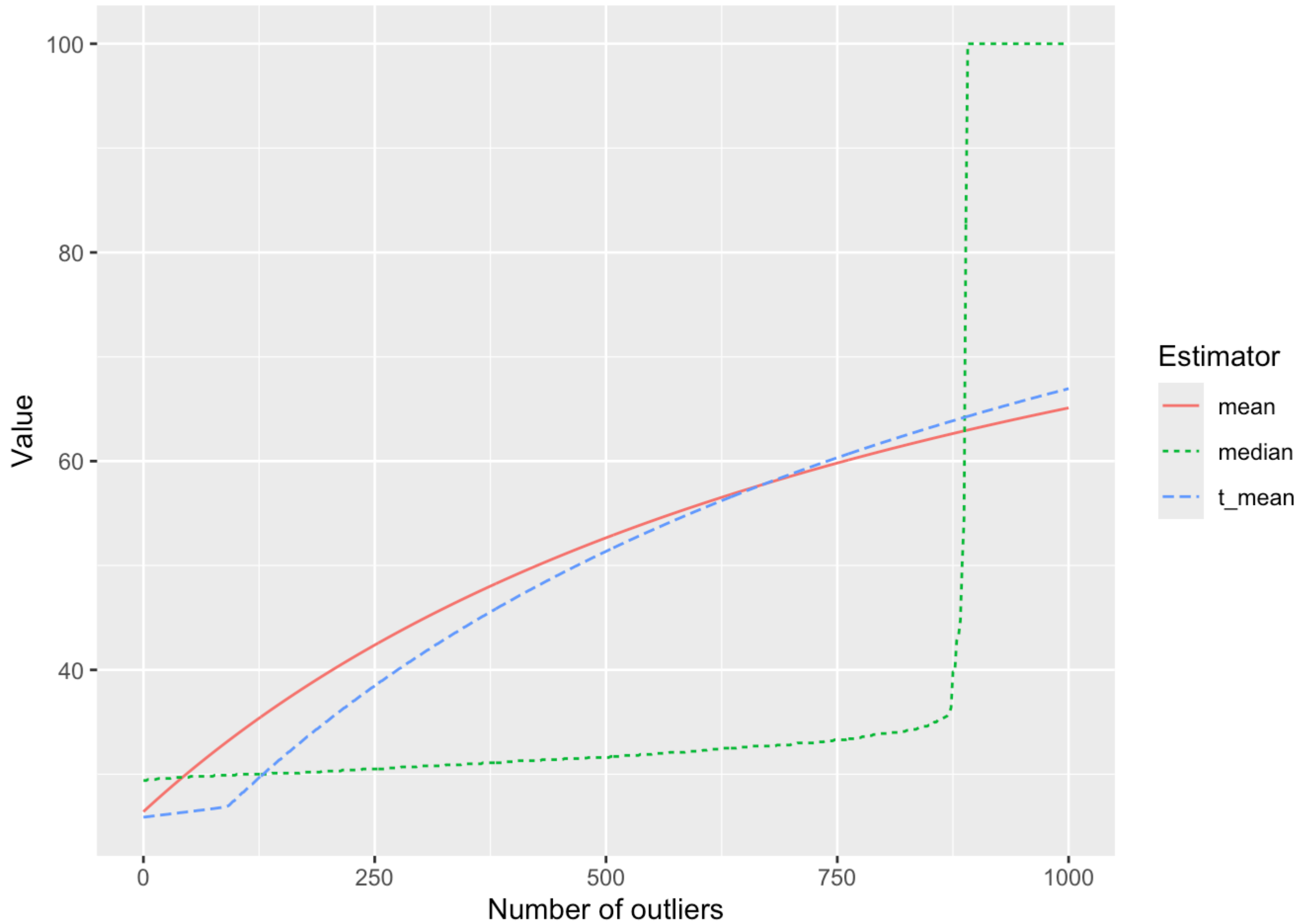
```
medians_vect <- c()
for(num_outliers in num_outliers_vect){
 corrupted_hal <- c(hal,rep(outlier_val,times=num_outliers))
 medians_vect <- c(medians_vect, median(corrupted_hal))
}
```

1.4 (Q2)

```
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)

```
df_means_medians <- data.frame(
  num_outliers = num_outliers_vect,
  mean = means_vect,
  t_mean = t_means_vect,
  median = medians_vect
)
df_means_medians %>%
 pivot_longer(!num_outliers, names_to = "Estimator", values_to =
"Value") %>%
 ggplot(aes(x=num_outliers,color=Estimator,
linetype=Estimator,y=Value)) +
 geom_line()+xlab("Number of outliers")
```

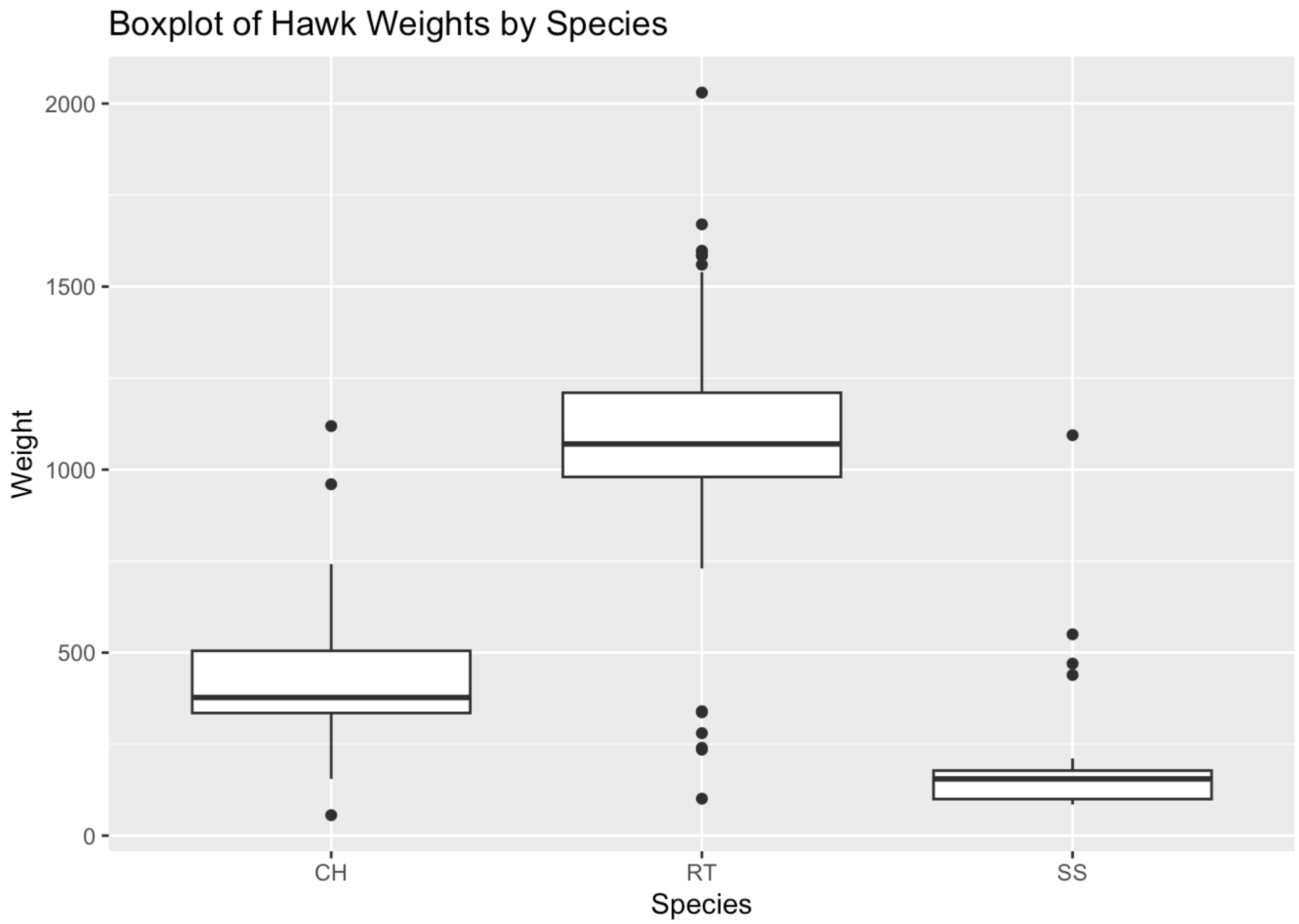


Median is the most robust estimator when the number of outliers is small.

1.5 (Q1)

```
library(ggplot2)
ggplot(Hawks, aes(x = Species, y = Weight)) +
  geom_boxplot() +
  labs(title = "Boxplot of Hawk Weights by Species", x = "Species", y = "Weight")
```

```
## Warning: Removed 10 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



1.5 (Q2)

```
quantiles <- 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)
  )

quantiles
```

```
## # 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.
```

1.5 (Q3)

```
num_outliers<-function(x){
  x <- na.omit(x)
  q25 <- quantile(x, 0.25)
  q75 <- quantile(x, 0.75)
  IQR_value <- q75 - q25
  lower_bound <- q25 - 1.5 * IQR_value
  upper_bound <- q75 + 1.5 * IQR_value
  sum(x < lower_bound | x > upper_bound)
}
num_outliers(c(0, 40, 60, 185))
```

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

1.5 (Q4)

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

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

1.6 (Q1)

```
Weight_value<-Hawks$Weight
Wing_value<-Hawks$Wing
cov_value<-cov(Weight_value,Wing_value,use = "complete.obs")
cor_value<-cor(Weight_value,Wing_value,use = "complete.obs")
cov_value
```

```
## [1] 41174.39
```

```
cor_value
```

```
## [1] 0.9348575
```

1.6 (Q2)

Covariance and Correlation under Linear Transformations Covariance Let X and Y be two variables with sample covariance $\text{Cov}(X, Y)$ and sample correlation $\text{Cor}(X, Y)$. We define linear transformations of X and Y as follows:

$$\tilde{X}_i = aX_i + b, \quad \tilde{Y}_i = cY_i + d$$

where $a, b, c, d \in \mathbb{R}$.

The covariance between $\tilde{X}$ and $\tilde{Y}$ can be expressed as:

$$\text{Cov}(\tilde{X}, \tilde{Y}) = \frac{1}{n} \sum_{i=1}^{n} (\tilde{X}_i - \mu_{\tilde{X}})(\tilde{Y}_i - \mu_{\tilde{Y}})$$

Substituting the linear transformations and expanding the terms:

$$\text{Cov}(\tilde{X}, \tilde{Y}) = \frac{1}{n} \sum_{i=1}^{n} (a(X_i - \mu_X))(c(Y_i - \mu_Y))$$

This simplifies to:

$$\text{Cov}(\tilde{X}, \tilde{Y}) = ac \cdot \text{Cov}(X, Y)$$

Correlation

The correlation between X and Y is defined as:

$$\text{Cor}(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}$$

Under the linear transformation, the standard deviations of $\tilde{X}$ and $\tilde{Y}$ become:

$$\sigma_{\tilde{X}} = |a|\sigma_X, \quad \sigma_{\tilde{Y}} = |c|\sigma_Y$$

Therefore, the correlation between $\tilde{X}$ and $\tilde{Y}$ is:

$$\text{Cor}(\tilde{X}, \tilde{Y}) = \frac{ac \cdot \text{Cov}(X, Y)}{|a|\sigma_X \cdot |c|\sigma_Y}$$

where if $ac > 0$, the correlation remains unchanged, but if $ac < 0$, the correlation is reversed.

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

X <- Hawks$Weight
Y <- Hawks$Wing

X_transformed <- a * X + b
Y_transformed <- c * Y + d

cov_original <- cov(X, Y, use = "complete.obs")
cor_original <- cor(X, Y, use = "complete.obs")

cov_transformed <- cov(X_transformed, Y_transformed, use = "complete.obs")
cor_transformed <- cor(X_transformed, Y_transformed, use = "complete.obs")


cov_theoretical <- a * c * cov_original
cor_theoretical <- sign(a * c)*cor_original

#original
cov_original
```

```
## [1] 41174.39
```

```
#compute
cov_transformed
```

```
## [1] -98818.54
```

```
#mathematical derivation
cov_theoretical
```

```
## [1] -98818.54
```

```
#original
cor_original
```

```
## [1] 0.9348575
```

```
#compute
cor_transformed
```

```
## [1] -0.9348575
```

```
#mathematical derivation
cor_theoretical
```

```
## [1] -0.9348575
```

2. Random variables and discrete random variables

2.1 (Q1)

$$\text{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$$

Our goal is to show that if X and Y are independent, then $\text{Cov}(X, Y) = 0$.

We expand the product $(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])$ using distributive properties:

$$(X - \mathbb{E}[X])(Y - \mathbb{E}[Y]) = XY - X\mathbb{E}[Y] - \mathbb{E}[X]Y + \mathbb{E}[X]\mathbb{E}[Y]$$

Now, we take the expectation of each term separately:

$$\mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = \mathbb{E}[XY] - \mathbb{E}[X\mathbb{E}[Y]] - \mathbb{E}[\mathbb{E}[X]Y] + \mathbb{E}[\mathbb{E}[X]\mathbb{E}[Y]]$$

Since $\mathbb{E}[Y]$ and $\mathbb{E}[X]$ are constants, we can factor them out of the expectation:

$$\mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] - \mathbb{E}[X]\mathbb{E}[Y] + \mathbb{E}[X]\mathbb{E}[Y]$$

The last two terms cancel each other out, so we are left with:

$$\text{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$$

Now, we use the fact that if X and Y are independent, then $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$. Substituting this into the covariance formula:

$$\text{Cov}(X, Y) = \mathbb{E}[X]\mathbb{E}[Y] - \mathbb{E}[X]\mathbb{E}[Y] = 0$$

Therefore, we have proven that the covariance of two independent random variables is zero:

$$\text{Cov}(X, Y) = 0$$

2.2 (Q1)

1. What is the probability mass function p_X for X?

The probability mass function p_X describes the probability of each possible value of the random variable X. Based on the problem setup, the PMF can be expressed as:

$$p_X(x) = \begin{cases} 1 - \alpha - \beta & \text{if } x = 0, \\ \alpha & \text{if } x = 3, \\ \beta & \text{if } x = 10, \\ 0 & \text{otherwise.} \end{cases}$$

2. What is the expectation of X?

The expectation (mean) of a discrete random variable is calculated as the sum of each possible value of X, weighted by its probability:

$$\mathbb{E}(X) = \sum_x x \cdot \mathbb{P}(X = x)$$

Substituting the given values of X and their probabilities, we get:

$$\mathbb{E}(X) = 0 \cdot (1 - \alpha - \beta) + 3 \cdot \alpha + 10 \cdot \beta$$

Thus, the expectation is:

$$\mathbb{E}(X) = 3\alpha + 10\beta$$

3. What is the variance of X?

The variance of a discrete random variable is given by:

$$\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}(X))^2] = \mathbb{E}(X^2) - (\mathbb{E}(X))^2$$

First, we calculate $\mathbb{E}(X^2)$:

$$\mathbb{E}(X^2) = 0^2 \cdot (1 - \alpha - \beta) + 3^2 \cdot \alpha + 10^2 \cdot \beta = 9\alpha + 100\beta$$

Now, we can compute the variance:

$$\text{Var}(X) = 9\alpha + 100\beta - (3\alpha + 10\beta)^2$$

Simplifying further:

$$\text{Var}(X) = 9\alpha + 100\beta - (9\alpha^2 + 60\alpha\beta + 100\beta^2)$$

Thus, the variance is:

$$\text{Var}(X) = 9\alpha(1 - \alpha) + 100\beta(1 - \beta) - 60\alpha\beta$$

4. What is the standard deviation of X?

The standard deviation is the square root of the variance:

$$\text{SD}(X) = \sqrt{\text{Var}(X)}$$

Thus, the standard deviation of X is:

$$\text{SD}(X) = \sqrt{9\alpha(1 - \alpha) + 100\beta(1 - \beta) - 60\alpha}\beta$$

2.2 (Q2)

1. The distribution $P_X(S)$ of X

$$P_X(S) = (1 - \alpha - \beta) \cdot \mathbb{1}_S(0) + \alpha \cdot \mathbb{1}_S(3) + \beta \cdot \mathbb{1}_S(10)$$

2. Distribution Function

$$F_X(x) = \begin{cases} 0 & \text{if } x < 0, \\ 1 - \alpha - \beta & \text{if } 0 \leqslant x < 3, \\ 1 - \beta & \text{if } 3 \leqslant x < 10, \\ 1 & \text{if } x \geqslant 10. \end{cases}$$

2.2 (Q3)

We are asked to calculate the variance of $Y = X_1 + X_2 + \cdots + X_n$, where $X_1, X_2, \ldots, X_n$ are independent and identically distributed random variables. First, we know from the properties of variance that:

$$\text{Var}(Y) = \text{Var}(X_1 + X_2 + \cdots + X_n) = \text{Var}(X_1) + \text{Var}(X_2) + \cdots + \text{Var}(X_n)$$

Since $X_1, X_2, \ldots, X_n$ are independent and identically distributed, the variance of each X_i is the same. Therefore:

$$\text{Var}(Y) = n \cdot \text{Var}(X)$$

From Question 2.2 (Q1), we already know the variance of X:

$$\text{Var}(X) = 9\alpha(1-\alpha) + 100\beta(1-\beta) - 60\alpha\beta$$

Thus, the variance of Y can be expressed as:

$$\text{Var}(Y) = n \cdot [9\alpha(1-\alpha) + 100\beta(1-\beta) - 60\alpha\beta]$$

2.2 (Q4)

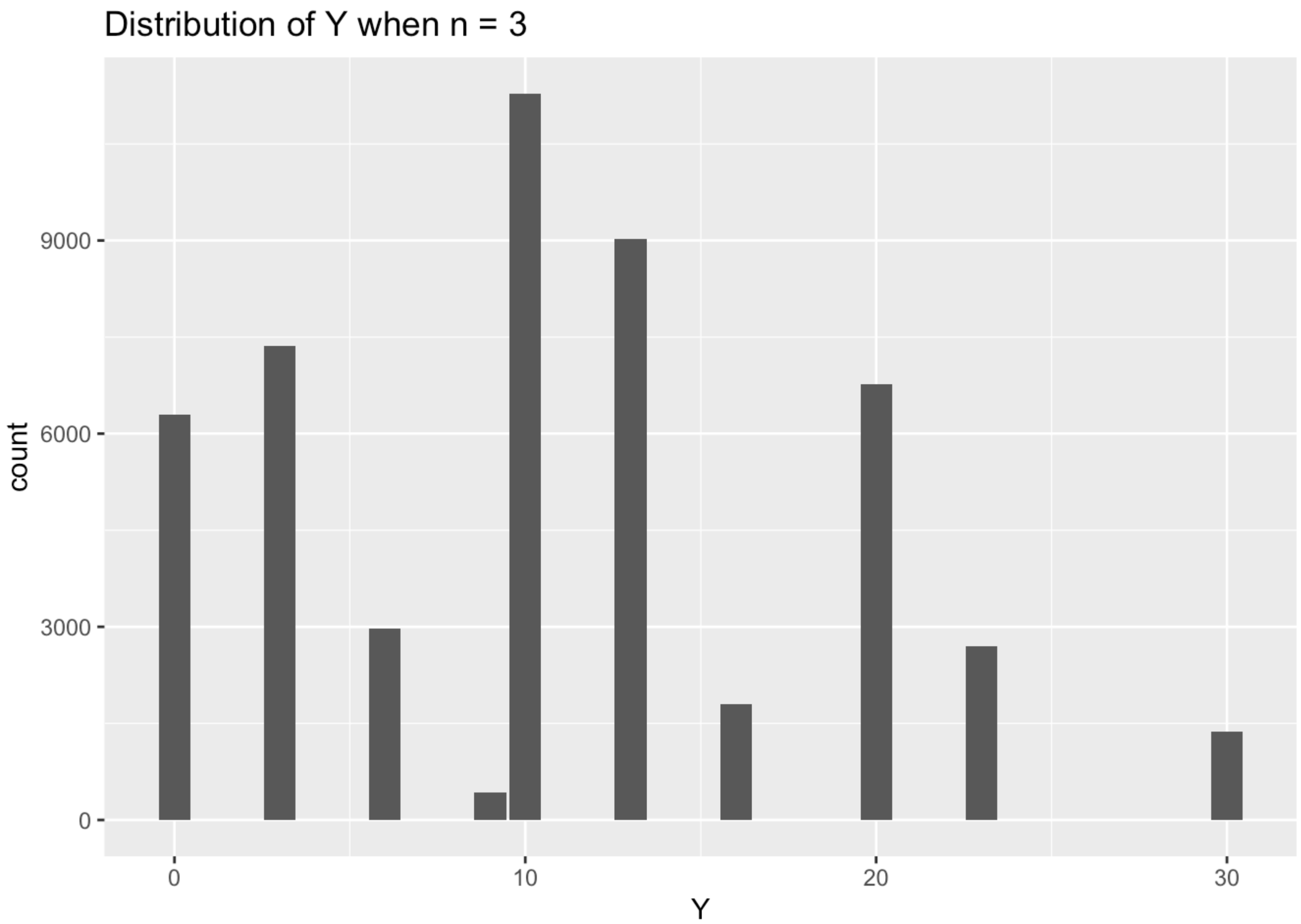
```
#Step1
Gen_X_numbers<-function(n){
  random_numbers<-runif(n)
  X_values <- c()
  for (random_number in random_numbers) {
  if (0<=random_number && random_number<0.5){
    X<-0
  }else if(0.5<=random_number&& random_number<0.7){
    X<-3
  }else{
    X<-10
  }
    X_values<-c(X_values,X)
    }
  return(X_values)

  }

Gen_X_numbers(4)
```

```
## [1] 10  0  0  0
```

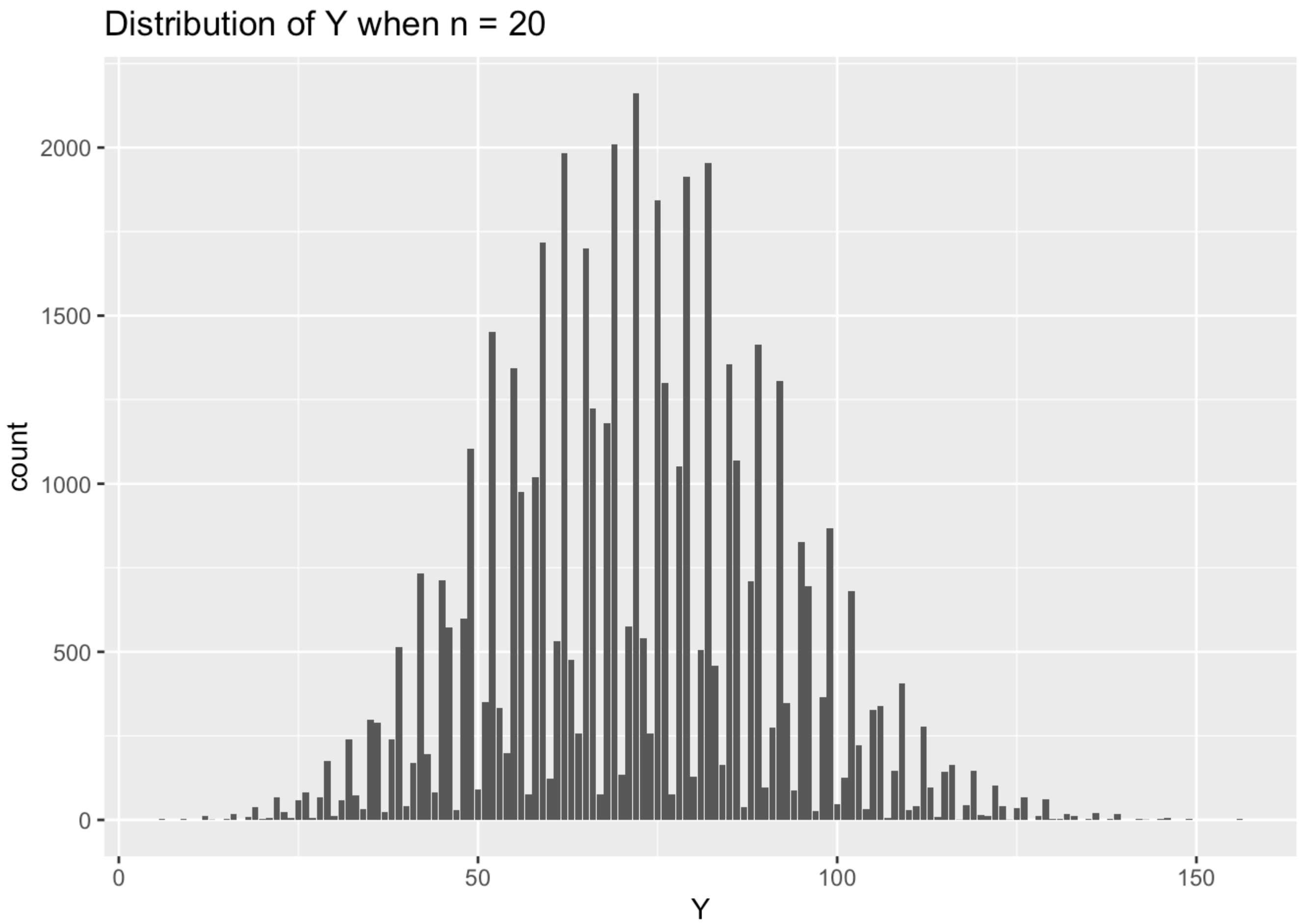
```
#Step2
Gen_Y_samples<-function(m,n){
  Y_samples <- map_dbl(1:m, ~ sum(Gen_X_numbers(n)))
  Y<-data.frame(index=1:m,Y=Y_samples)
  return(Y)

}
Gen_Y_samples(5,2)
```

```
##   index  Y
## 1     1 10
## 2     2 10
## 3     3 20
## 4     4 10
## 5     5  6
```

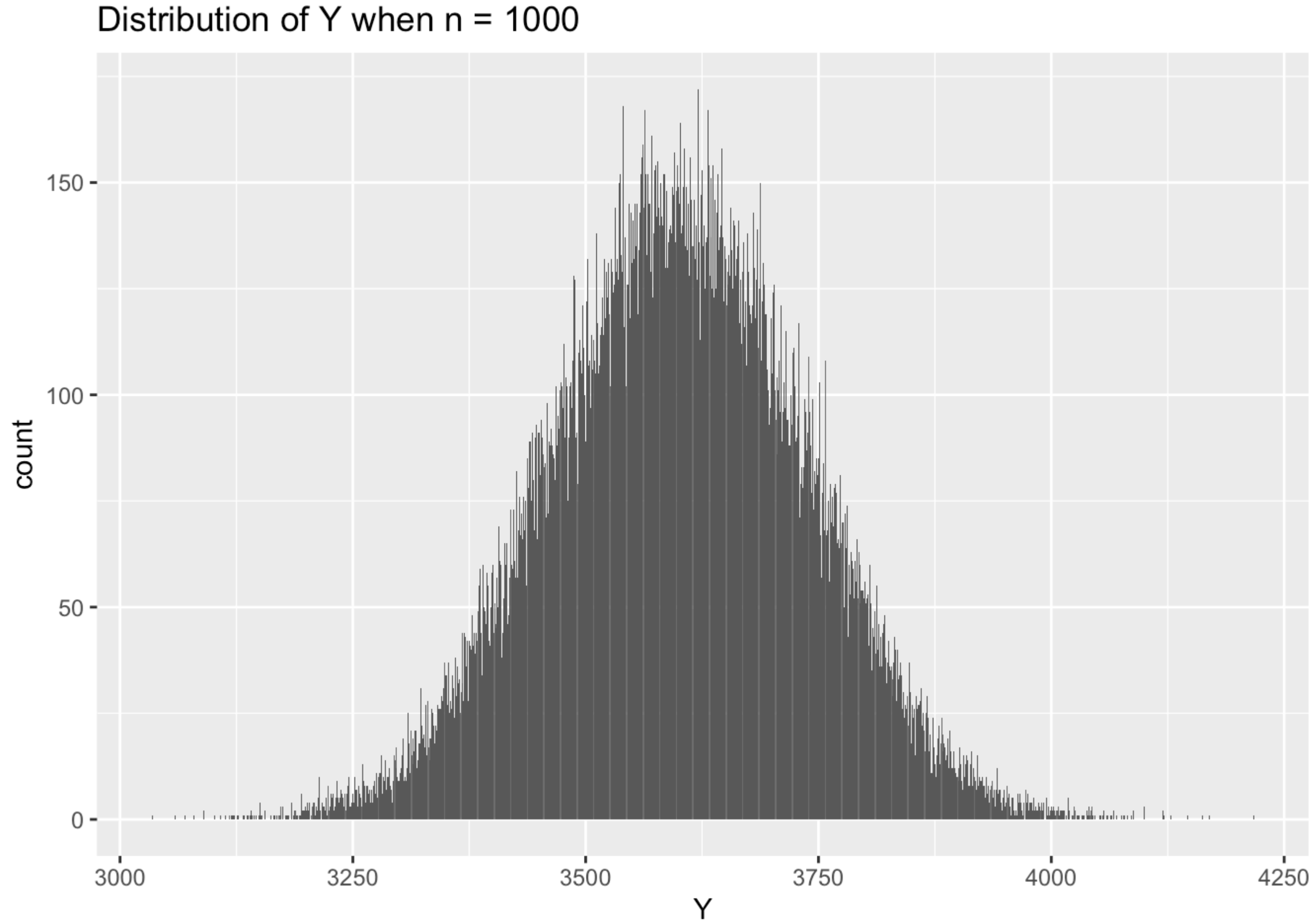
```
#Step3
n<-3
m<-50000
Y<- Gen_Y_samples(m, n)

ggplot(Y, aes(x = Y)) +
  geom_bar() +
  labs(title = "Distribution of Y when n = 3", x = "Y", y = "count")
```




```
#Step4
n_20<-20
Y_20<-Gen_Y_samples(m,n_20)
ggplot(Y_20, aes(x = Y)) +
  geom_bar() +
  labs(title = "Distribution of Y when n = 20", x = "Y", y = "count")
```




```
#Step5
n_1000 <- 1000
Y_1000 <- Gen_Y_samples(m, n_1000)
ggplot(Y_1000, aes(x = Y)) +
  geom_bar() +
  labs(title = "Distribution of Y when n = 1000", x = "Y", y = "count")
```



3. Probability theory

3.1 (Q1)

We want to calculate the probability that U falls within the interval $[a, b]$, where $0 \leqslant a \leqslant b \leqslant 1$. This probability can be expressed as:

$$\mathbb{P}(U \in [a, b]) = \int_a^b p_U(x)\, dx$$

Substituting the value of $p_U(x) = 1$ for $x \in [0, 1]$:

$$\mathbb{P}(U \in [a, b]) = \int_a^b 1\, dx = b - a$$

Thus, we have shown that for any $a, b \in [0, 1]$, the probability that $U \in [a, b]$ is given by:

$$\mathbb{P}(U \in [a, b]) = b - a$$

3.1 (Q2)

```
set.seed(0)
n <- 1000
sample_X <- data.frame(U=runif(n)) %>%
 mutate(X=case_when(
 (0<=U)&(U<0.25)~3,
 (0.25<=U)&(U<0.5)~10,
 (0.5<=U)&(U<=1)~0)) %>%
 pull(X)

head(sample_X)
```

```
## [1]  0 10 10  0  0  3
```

```
table(sample_X) / n
```

```
## sample_X
##     0     3    10
## 0.481 0.244 0.275
```

Since each random variable X_i is generated as an independent uniform sample using and then mapped according to the proportions using the function.

3.1 (Q3)

```
sample_X_0310 <- function(alpha, beta, n){
  random_numbers<-runif(n)
  X_values<-c()
  for (random_number in random_numbers) {
  if (0<=random_number && random_number<alpha){
    X<-3
  }else if(alpha<=random_number&& random_number<alpha+beta){
    X<-10
  }else{
    X<-0
  }
    X_values<-c(X_values,X)
    }
  return(X_values)
}
table(sample_X_0310(0.25, 0.25, 100))/n
```

```
##
##     0     3    10
## 0.050 0.025 0.025
```

3.1 (Q4)

```
alpha <- 1/2
beta <- 1/10
n <- 10000

X_samples <- sample_X_0310(alpha, beta, n)


sample_mean <- mean(X_samples)

print(paste("The value of sample mean is:", sample_mean))
```

```
## [1] "The value of sample mean is: 2.5104"
```

Theoretical Expectation:

The theoretical expectation $E(X)$ is obtained by the weighted average of the values. According to the formula derived earlier:

$$E(X) = 3 \cdot \alpha + 10 \cdot \beta + 0 \cdot (1 - \alpha - \beta)$$

Substituting $\alpha = \frac{1}{2}$ and $\beta = \frac{1}{10}$, we get:

$$E(X) = 3 \cdot \frac{1}{2} + 10 \cdot \frac{1}{10} = 1.5 + 1 = 2.5$$

According to the Law of Large Numbers, when n is sufficiently large, the sample mean will approach the theoretical expectation $E(X)$. This is because the sample average of independent and identically distributed random variables converges to their expected value.

3.1 (Q5)

```
sample_variance <- var(X_samples)

print(paste("The value of sample_variance is:", sample_variance))
```

```
## [1] "The value of sample_variance is: 8.2859204320432"
```

The formula for the theoretical variance $\text{Var}(X)$ is:

$$\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}(X))^2] = \mathbb{E}(X^2) - (\mathbb{E}(X))^2$$

First, let's compute $E(X^2)$:

$$E(X^2) = 3^2 \cdot \alpha + 10^2 \cdot \beta + 0^2 \cdot (1 - \alpha - \beta)$$

Substituting $\alpha = \frac{1}{2}$ and $\beta = \frac{1}{10}$, we get:

$$E(X^2) = 9 \cdot \frac{1}{2} + 100 \cdot \frac{1}{10} = 4.5 + 10 = 14.5$$

Thus, the theoretical variance is:

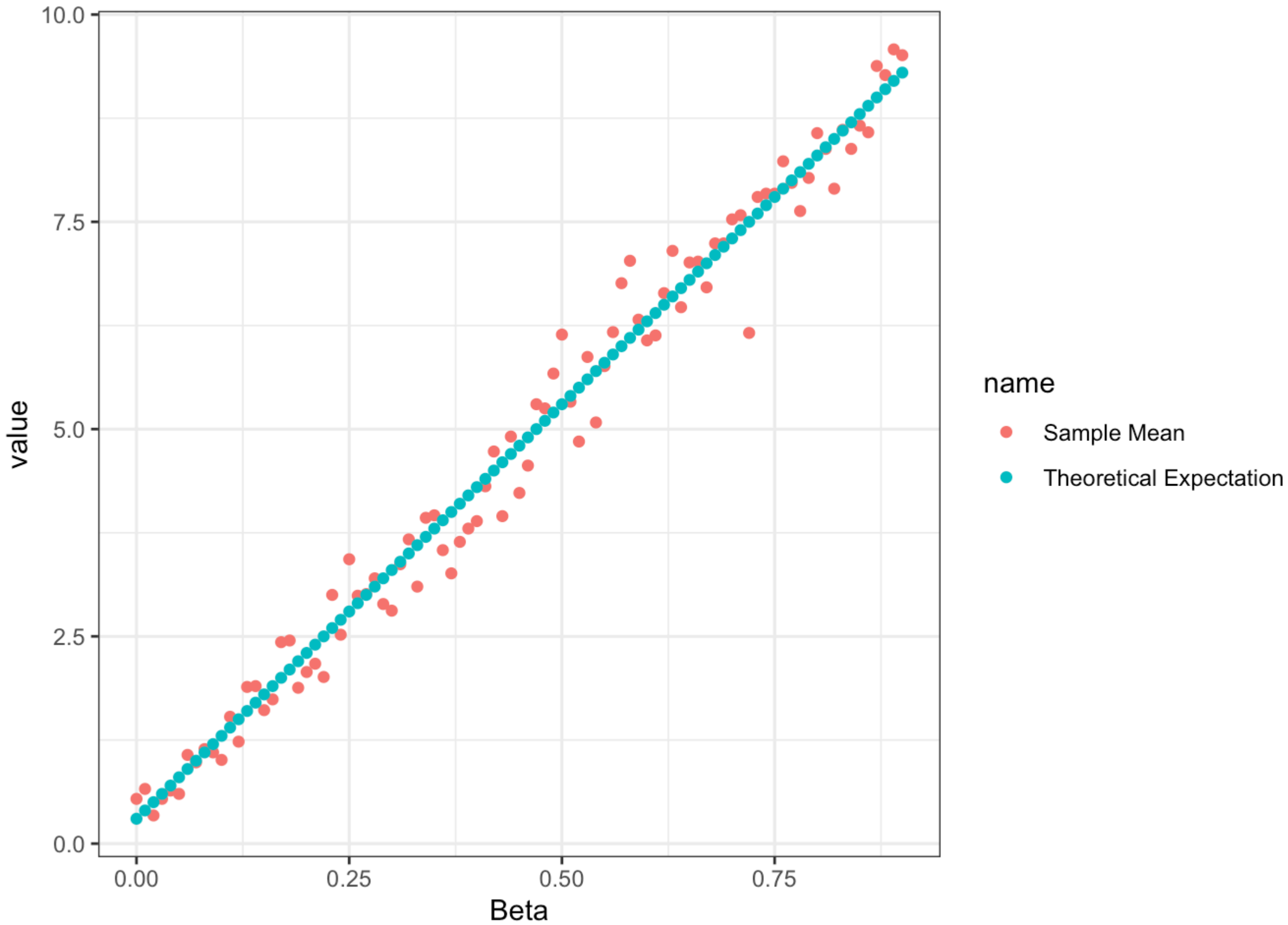
$$\mathrm{Var}(X) = 14.5 - (2.5)^2 = 14.5 - 6.25 = 8.25$$

3.1 (Q6)

```
alpha <- 1/10
n <- 100
beta_values <- seq(0, 0.9, by = 0.01)

Table<-data.frame(beta=beta_values)%>%
  mutate(
    sample_X=map(beta,~sample_X_0310(alpha, .x, n)),
    samplemean = map_dbl(sample_X, mean),
    Expectation = 3 * alpha + 10 * beta
  )
```

3.1 (Q7)

```
ggplot(Table, aes(x = beta)) +
  geom_point(aes(y = samplemean, color = "Sample Mean")) +
  geom_point(aes(y = Expectation, color = "Theoretical Expectation")) +
  labs(x = "Beta", y = "value",color="name") +
  theme_bw()
```



3.2 (Q1)

(a) Verifying that p_λ is a well-defined PDF

To verify that p_λ is a well-defined probability density function (PDF), it must satisfy two conditions:

1. $p(x) \geqslant 0$ for all x.
2. $\int_{-\infty}^{\infty} p(x)\, dx = 1$.

For the probability density function of the exponential distribution $p_\lambda(x)$:

$$p_\lambda(x) = \begin{cases} 0 & \text{if } x < 0, \\ \lambda e^{-\lambda x} & \text{if } x \geqslant 0. \end{cases}$$

For $x \geqslant 0$, $\lambda e^{-\lambda x} \geqslant 0$ because $\lambda > 0$ and $e^{-\lambda x} > 0$, so the non-negativity condition is satisfied.

For $x < 0$, $p_\lambda(x) = 0$, which is clearly non-negative as well.

Now we verify that the total probability is 1:

$$\int_{-\infty}^{\infty} p_\lambda(x)\, dx = \int_{0}^{\infty} \lambda e^{-\lambda x}\, dx$$

By integrating, we get:

$$\int_{0}^{\infty} \lambda e^{-\lambda x}\, dx = \left[-e^{-\lambda x}\right]_0^\infty = 0 - (-1) = 1$$

Thus, $p_\lambda(x)$ is a valid probability density function.

(b) Cumulative Distribution Function (CDF)

The cumulative distribution function $F_X(x)$ is defined as the probability that $X \leqslant x$:

$$F_X(x) = P(X \leqslant x) = \int_{-\infty}^{x} p_\lambda(t)\, dt$$

For $x < 0$, since $p_\lambda(x) = 0$:

$$F_X(x) = 0 \quad \text{if } x < 0$$

For $x \geqslant 0$, we have:

$$F_X(x) = \int_{0}^{x} \lambda e^{-\lambda t}\, dt = 1 - e^{-\lambda x}$$

Thus, the cumulative distribution function is:

$$F_X(x) = \begin{cases} 0 & \text{if } x < 0, \\ 1 - e^{-\lambda x} & \text{if } x \geqslant 0. \end{cases}$$

(c) Quantile Function

The quantile function is the inverse of the CDF. Given a probability p, we need to find x such that $F_X(x) = p$.

Starting from the equation $F_X(x) = 1 - e^{-\lambda x} = p$, solve for x:

$$e^{-\lambda x} = 1 - p$$

Taking the natural logarithm on both sides:

$$-\lambda x = \ln(1 - p)$$

Thus, the quantile function is:

$$x = -\frac{1}{\lambda}\ln(1 - p)$$

3.2 (Q2)

```
my_cdf_exp <- function(x, lambda) {
  if (x < 0) {
    return(0)
  } else {
    return(1 - exp(-lambda * x))
  }
}

lambda <- 1/2
map_dbl(.x=seq(-1,4), .f=~my_cdf_exp(x=.x,lambda=lambda) )
```

```
## [1] 0.0000000 0.0000000 0.3934693 0.6321206 0.7768698 0.8646647
```

```
test_inputs <- seq(-1,10,0.1)
my_cdf_output <- map_dbl(.x=test_inputs,
.f=~my_cdf_exp(x=.x,lambda=lambda))
inbuilt_cdf_output <-
map_dbl(.x=test_inputs,.f=~pexp(q=.x,rate=lambda))
all.equal(my_cdf_output,inbuilt_cdf_output)
```

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

3.2 (Q3)

```
my_quantile_exp <- function(p, lambda){
  if(p>=0 && p<=1){
    return(-log(1 - p) / lambda)
  }else
    print("p must be in the range (0, 1)")
}

lambda <- 1/2
test_inputs <- seq(0.01, 0.99, by = 0.01)
my_quantile_output <- map_dbl(.x = test_inputs, .f = ~my_quantile_exp(p = .x, lambda
= lambda))

inbuilt_quantile_output <- map_dbl(.x = test_inputs, .f = ~qexp(p = .x, rate = lambd
a))

all.equal(my_quantile_output, inbuilt_quantile_output)
```

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

3.2 (Q4)

(a) The mean and expectation of the function are equivalent.

So the mean is $E(X) = \int_{-\infty}^{\infty} x\, p_\lambda(x)\, dx$

Because, if $x<0$, $p_\lambda(x)=0$. So $E(X) = \int_0^\infty x\lambda e^{-\lambda x} dx$

We can calculate this integral using the partial integration method.

The methods is expressed as:

$$\int u\,dv = uv - \int v\,du \Rightarrow \int uv'dx = uv - \int u'v\,dx$$

let $u=x$, and $v'=\lambda e^{-\lambda x}$, then $u'=1$ and $v=-e^{-\lambda x}$

So, $E(X) = [x\cdot(-e^{-\lambda x})]_0^\infty - \int_0^\infty 1\cdot(-e^{-\lambda x})dx$

$$= [x\cdot(-e^{-\lambda x})]_0^\infty + \int_0^\infty e^{-\lambda x} dx$$

$$= [x\cdot(-e^{-\lambda x})]_0^\infty + [(-\tfrac{1}{\lambda}e^{-\lambda x})]_0^\infty$$

$$= 0-0+0-(-\tfrac{1}{\lambda})$$

$$= \frac{1}{\lambda}$$

So the mean is $\frac{1}{\lambda}$.

(b) The variance is $Var(X) = E[(X-E(X))^2] = E(X^2) - [E(X)]^2$

First, we calculate the $E(X^2)$

$$E(X^2) = \int_0^\infty x^2\lambda e^{-\lambda x^2} dx$$

let $u=x^2$ and $v'=\lambda e^{-\lambda x}$, we use $\int uv'dx = uv - \int u'v\,dx$

$u'=2x \quad v=-e^{-\lambda x}$

So $E(X^2) = [-x^2e^{-\lambda x}]_0^\infty + \int_0^\infty 2xe^{-\lambda x}dx$

$$= [-x^2e^{-\lambda x}]_0^\infty + [-\tfrac{2x}{\lambda}e^{-\lambda x}]_0^\infty + \tfrac{2}{\lambda}\int_0^\infty e^{-\lambda x}dx$$

$$= [-x^2e^{-\lambda x}]_0^\infty + [-\tfrac{2x}{\lambda}e^{-\lambda x}]_0^\infty + [(-\tfrac{2}{\lambda^2}e^{-\lambda x})]_0^\infty$$

$$= 0-0+0-0+0-(\tfrac{-2}{\lambda^2})$$

$$= \frac{2}{\lambda^2}$$

So $Var(X) = E(X^2) - [E(X)]^2 = \frac{2}{\lambda^2} - (\frac{1}{\lambda})^2 = \frac{1}{\lambda^2}$

So the variance is $\frac{1}{\lambda^2}$

solution

3.3 (Q1)

Let $Z = X_1 + X_2 + \cdots + X_n$, where each $X_i \sim \text{Bernoulli}(p)$. Therefore, the expectation of Z is:

$$E(Z) = E(X_1 + X_2 + \cdots + X_n) = E(X_1) + E(X_2) + \cdots + E(X_n)$$

Since each X_i is independent and follows a Bernoulli distribution with parameter p, we have:

$$E(Z) = n \cdot p$$

The variance of Z is given by:

$$\text{Var}(Z) = \text{Var}(X_1 + X_2 + \cdots + X_n) = \text{Var}(X_1) + \text{Var}(X_2) + \cdots + \text{Var}(X_n)$$

For $X_i \sim \text{Bernoulli}(p)$, the variance is:

$$\text{Var}(X_i) = p(1 - p)$$

Thus, the variance of Z is:

$$\text{Var}(Z) = n \cdot p \cdot (1 - p)$$

3.3 (Q2)

```
n <- 50
p <- 7/10
x_values <-c(seq(0,n,by=1))
pmf_values <- dbinom(x = x_values, size = n, prob = p)
binom_df <- data.frame(x = x_values, pmf = pmf_values)

head(binom_df, 3)
```

```
##   x          pmf
## 1 0 7.178980e-27
## 2 1 8.375477e-25
## 3 2 4.787981e-23
```

3.3 (Q3)

```
mu <- 50 * 0.7
sigma <- sqrt(50 * 0.7 * (1 - 0.7))
x_values <- seq(0, 50, by = 0.01)
pdf_values <- dnorm(x = x_values, mean = mu, sd = sigma)
gaussian_df <- data.frame(x = x_values, pdf = pdf_values)
head(gaussian_df, 3)
```

```
##      x          pdf
## 1 0.00 5.707825e-27
## 2 0.01 5.901264e-27
## 3 0.02 6.101201e-27
```

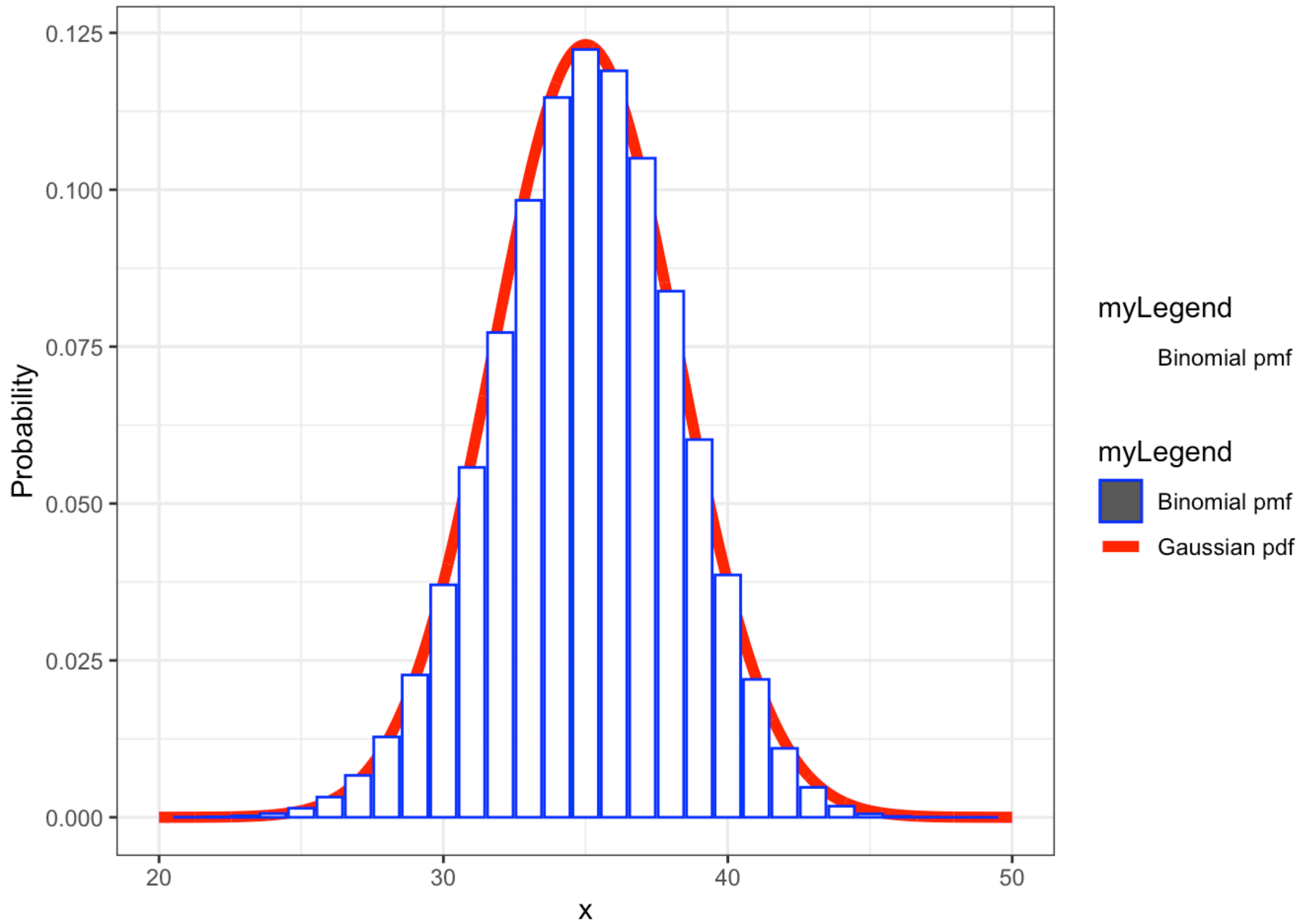
3.3 (Q4)

```
colors<-c("Gaussian pdf"="red", "Binomial pmf"="blue")
fill<-c("Gaussian pdf"="white", "Binomial pmf"="white")

ggplot() + labs(x="x",y="Probability") + theme_bw() +
 # create plot of Gaussian density
 geom_line(data=gaussian_df, aes(x,y=pdf,color="Gaussian pdf"),size=2) +
 # create a bar chart from PMF of Binomial distribution
 geom_col(data=binom_df, aes(x=x,y=pmf, color="Binomial pmf",fill="Binomial pmf")) +
 # set color
 scale_color_manual(name = "myLegend", values=colors) +
 scale_fill_manual(name = "myLegend", values=fill) +
 xlim(c(20,50))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Removed 2000 rows containing missing values or values outside the scale r
ange
## (`geom_line()`).
```

```
## Warning: Removed 22 rows containing missing values or values outside the scale ran
ge
## (`geom_col()`).
```



This phenomenon is a result of the Central Limit Theorem. The Central Limit Theorem states that as the number of independent and identically distributed random variables n increases, their mean tends to follow a normal distribution (also known as a Gaussian distribution).

In this case, the binomial distribution $Z \sim \text{Binom}(n, p)$ is made up of $n = 50$ independent Bernoulli trials. Since n is large enough, the shape of the binomial distribution closely resembles that of the normal distribution.

The comparison in the graph clearly shows that the shape of the binomial distribution's PMF is similar to the Gaussian distribution's PDF. This indicates that as n increases, the binomial distribution can be approximated by the normal distribution over a wide range. This is a manifestation of the Central Limit Theorem.