Assignment5

Jack

2024-10-16

1. Exploratory data analysis

```
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
```

1.1 Q1

```
HawksTail <- Hawks[["Tail"]]
HawksTailMean <- HawksTail %>% mean(na.rm=TRUE)
HawksTailMedian <- HawksTail %>% median(na.rm=TRUE)
# Mean
HawksTailMean
```

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

```
# Median
HawksTailMedian
```

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

1.2 Q1

```
s <- Hawks %>% summarise(across(c("Wing","Weight"),list(mean=~mean(.,na.rm=TRUE),t_mean=~ quantile(.,probs=0.5,na.rm=TRUE),median=~median(.,na.rm=TRUE)),.names="{.col}_{.fn}"))
s
```

```
##   Wing_mean Wing_t_mean Wing_median Weight_mean Weight_t_mean Weight_median
## 1  315.6375         370         370    772.0802           970           970
```

1.2 Q2

```
s_group <- Hawks %>% group_by(Species) %>% summarise(across(c("Wing","Weight"),list(mean=~mean(.,na.rm=TRUE),t_mean=~ quantile(.,probs=0.5,na.rm=TRUE),median=~median(.,na.rm=TRUE)),.names="{.col}_{.fn}"))
s_group
```

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

1.3 Q1

$Mean = b + aA$

```
# verify
HawksTailMean
```

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

```
mean(HawksTail*2+3)
```

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

1.3 Q2

$sample \quad variance = a^2 \times p$

```
# HawksTail sample variance
var(HawksTail,na.rm=TRUE)
```

```
## [1] 1356.037
```

```
var(HawksTail*2+3,na.rm=TRUE)
```

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

```
4*var(HawksTail,na.rm=TRUE)
```

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

$sample \quad deviation = a\sqrt{p} = a \times q$

```
# HawksTail sample deviation
sd(HawksTail,na.rm=TRUE)
```

```
## [1] 36.8244
```

```
sd(HawksTail*2+3,na.rm=TRUE)
```

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

```
2*sd(HawksTail,na.rm=TRUE)
```

```
## [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))
}
means_vect %>% head(5)
```

```
## [1] 26.41086 26.49236 26.57367 26.65481 26.73576
```

1.4 Q1 Sample median

```
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))
}
medians_vect %>% head(5)
```

```
## [1] 29.40 29.40 29.40 29.40 29.45
```

1.4 Q2 Sample trimmed mean

```
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))
}
t_means_vect %>% head(5)
```

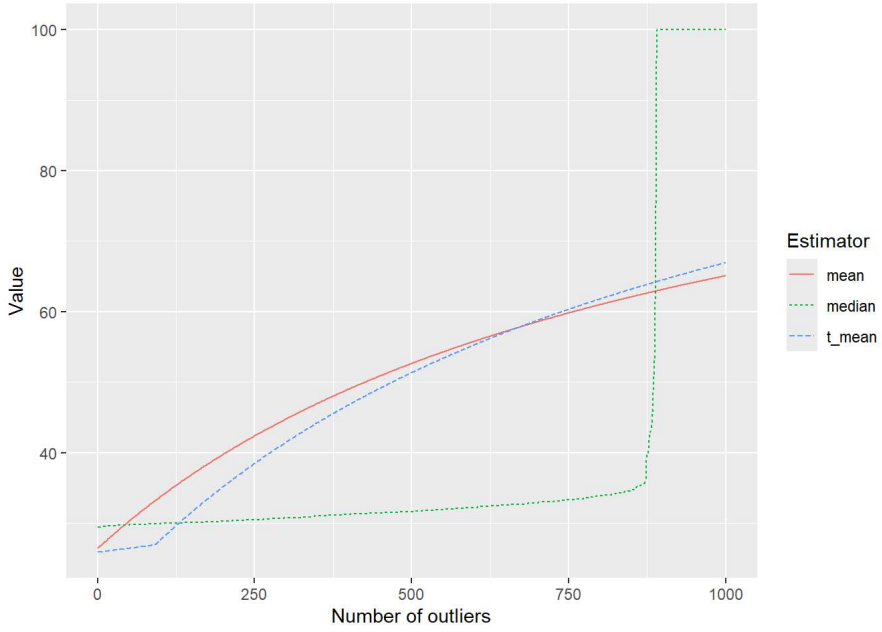
```
## [1] 25.88386 25.89371 25.90352 25.91345 25.92335
```

1.4 Q3 Visualisation

```
df_means_medians <- data.frame(num_outliers=num_outliers_vect,mean=means_vect,t_mean=t_means_vect,median=medians_vect)
df_means_medians %>% head(5)
```

```
##   num_outliers     mean   t_mean median
## 1            0 26.41086 25.88386  29.40
## 2            1 26.49236 25.89371  29.40
## 3            2 26.57367 25.90352  29.40
## 4            3 26.65481 25.91345  29.40
## 5            4 26.73576 25.92335  29.45
```

```
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")
```



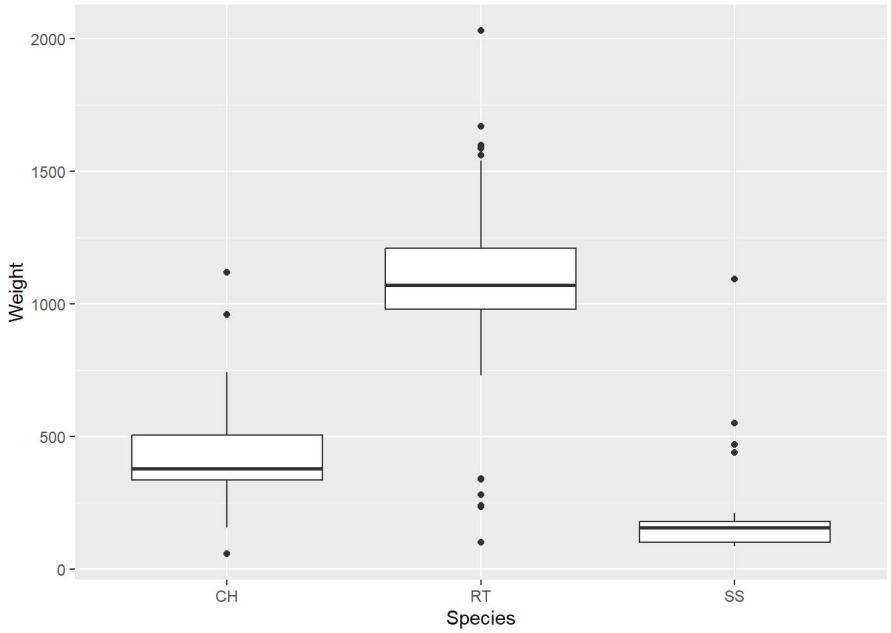

```
# I think they are all stable when the number of outliers is small
```

1.5 Box plots and outliers

1.5 Q1

```
ggplot(data=Hawks,aes(x=Species,y=Weight))+geom_boxplot()
```

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



1.5 Q2 quantile and boxplots

```
quantiles <- Hawks %>% group_by(Species) %>% summarise(across("Weight",list(quantile025=~quantile(.,probs=0.25,na.rm=TRUE),q
uantile050=~ quantile(.,probs=0.5,na.rm=TRUE),quantile075=~quantile(.,probs=0.75,na.rm=TRUE))))
quantiles
```

```
## # A tibble: 3 × 4
##   Species Weight_quantile025 Weight_quantile050 Weight_quantile075
##   <fct>                <dbl>              <dbl>              <dbl>
## 1 CH                     335               378.               505
## 2 RT                     980              1070               1210
## 3 SS                     100               155                178.
```

```
# These numbers are correspond to the box (interquartile range)
```

1.5 Q3 Outliers

```
num_outliers <- function(x){
  #remove any nans
  x <- x[!is.na(x)]
  # compute q25 and q75
  q25 <- quantile(x,0.25,na.rm=TRUE)
  q75 <- quantile(x,0.75,na.rm=TRUE)
  # create IQR
  IQR <- q75 - q25
  n <- 0
  for(num in x){
    if(num < q25-1.5*IQR | num > q75+1.5*IQR){
      n <- n + 1
    }
  }
  return (n)
}
num_outliers(c(0,40,60,185))
```

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

1.5 Q4 Outliers by group

```
Hawks %>% group_by(Species) %>% summarise(across("Weight",list(num_outliers_weight=~num_outliers(.))))
```

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

1.6 Covariance and correlation under linear transformations

1.6 Q1 Compute the covariance and correlation

```
cov(Hawks[["Weight"]],Hawks[["Wing"]],use="complete.obs")
```

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

```
cor(Hawks[["Weight"]],Hawks[["Wing"]],use="complete.obs")
```

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

1.6 Q2

$Cov(\tilde{X},\tilde{Y}) = a \times c \times Cov(X,Y) = a \times c \times S$

```
# Verify Cov
cov(Hawks[["Weight"]]*2.4+7.1,Hawks[["Wing"]]*(-1)+3,use="complete.obs")
```

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

```
cov(Hawks[["Weight"]],Hawks[["Wing"]],use="complete.obs")*2.4*(-1)
```

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

$Cor(\tilde{X},\tilde{Y}) = R$

```
# Verify Cor
cor(Hawks[["Weight"]]*2.4+7.1,Hawks[["Wing"]]*(-1)+3,use="complete.obs")
```

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

```
cor(Hawks[["Weight"]],Hawks[["Wing"]],use="complete.obs")
```

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

2. Random variables and discrete random variables

2.1 Q1 Expectation and variance

$E[(X-\overline{X})\cdot(Y-\overline{Y})] = E(XY) - E(X\overline{Y}) - E(\overline{X}Y) + E(\overline{X}\overline{Y}) = E(XY) - E(XY) - E(XY) + E(XY) = 0$

2.2 Distribution

2.2 Q1

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

$$p_X(x) = P(X=x) = \begin{cases} \alpha, X=3 \\ \beta, X=10 \\ 1-\alpha-\beta, X=0 \\ 0, otherwise \end{cases}$$

2. What is the expectation of X?

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

3. What is the variance of X?

$Var(X) = E[(X-E(X))^2] = \sum_x (X-E(X))^2 \cdot p_X(x) = (3-3\alpha-10\beta)^2\times\alpha + (10-3\alpha-10\beta)^2\times\beta + (0-3\alpha-10\beta)^2\times(1-\alpha-\beta)$

4. What is the standard deviation of X?

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

2.2 Q2 Distribution and distribution function

1.

$P(S) = \alpha \times 1_{(3)}(x) + \beta \times 1_{(10)}(x) + (1-\alpha-\beta) \times \ 1_{(0)}(x)$

2. Write down the distribution function F of X

$$F_X(x) = \begin{cases} 0, & if \quad x < 0 \\ 1-\alpha-\beta, & if \quad 0 \leq x < 3 \\ 1-\beta, & if \quad 3 \leq x < 10 \\ 1, & if \quad x \geq 10 \end{cases}$$

2.2 Q3 Variance and Covariance

$Var(Y) = n \cdot Var(X) = n \cdot (9\alpha + 100\beta - 9\alpha^2 - 60\alpha\beta - 100\beta^2)$

2.2 Q4

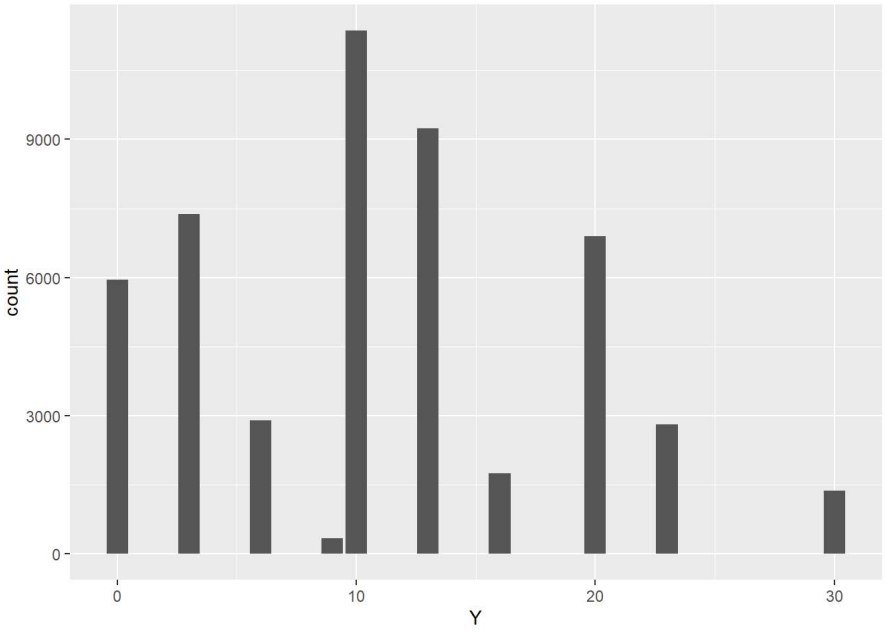
```
# create function to generate a sample then return a vector
Gen_X_numbers <- function(n){
  numbers <- c()
  nums <- sample(seq(0,1,by=0.01),n)
  for(num in nums){
    if(num < 0.5){
      numbers <- c(numbers,0)
    }else if(num < 0.7){
      numbers <- c(numbers,3)
    }else{
      numbers <- c(numbers,10)
    }
  }
  return (numbers)
}
Gen_X_numbers(4)
```

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

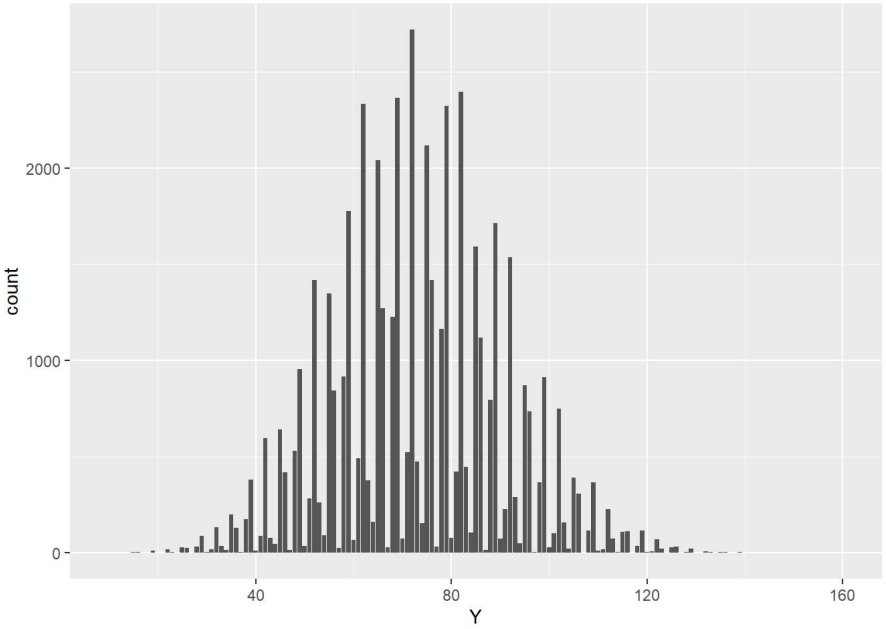
```
# create function to generate a sample of Y
Gen_Y_samples <- function(m,n){
  iters <- seq(1,m)
  Y <- map_dbl(iters,~sum(Gen_X_numbers(n)))
  result <- data.frame(index=iters,Y=Y)
  return (result)
}
Gen_Y_samples(5,2)
```

```
##   index  Y
## 1     1 20
## 2     2  3
## 3     3  0
## 4     4 13
## 5     5 10
```

```
# Let n=3, m=50000
Gen_Y_samples(50000,3) %>% ggplot(aes(x=Y)) + geom_bar() + ylab("count")
```




```
# increase the values of n to 20
Gen_Y_samples(50000,20) %>% ggplot(aes(x=Y)) + geom_bar() + ylab("count")
```




```
# increase n to 1000
# It is too large
#Gen_Y_samples(50000,1000) %>% ggplot(aes(x=Y)) + geom_bar() + ylab("count")
```

3. Continuous random variables and limit laws

3.1 Simulating data with the uniform distribution

3.1 Q1

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

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)

X3 <- 0
X10 <- 0
X0 <- 0
for(n in sample_X){
  if (n == 3){
    X3 <- X3+1
  }else if(n == 10){
    X10 <- X10+1
  }else{
    X0 <- X0+1
  }
}
X3/1000
```

```
## [1] 0.244
```

```
X10/1000
```

```
## [1] 0.275
```

```
X0/1000
```

```
## [1] 0.481
```

3.1 Q3

```
sample_X_0310 <- function(alpha,beta,n){
  probability <- c(alpha,beta,1-alpha-beta)
  return (sample(c(3,10,0),n,replace=TRUE,prob=probability))
}
sample_X_0310(0.25,0.25,10)
```

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

3.1 Q4

```
X <- sample_X_0310(1/2,1/10,10000)
X_mean <- mean(X,na.rm=TRUE)
X_mean
```

```
## [1] 2.5308
```

```
X_E <- 3*(1/2) + 10*(1/10) + 0*(1-1/2-1/10)
X_E
```

```
## [1] 2.5
```

```
# They are closed, if the sample is larger, then they will be closer
```

3.1 Q5

```
X_Var <- var(X,na.rm=TRUE)
X_Var
```

```
## [1] 8.412293
```

```
3*3*(1/2) + 10*10*(1/10) + 0*0*(1-1/2-1/10) - (3*(1/2) + 10*(1/10) + 0*(1-1/2-1/10))^2
```

```
## [1] 8.25
```

3.1 Q6

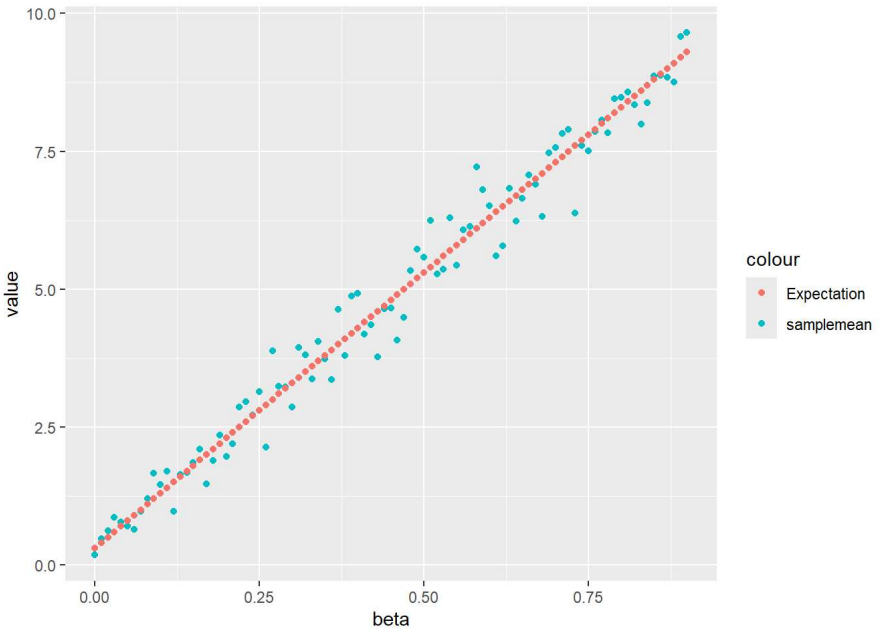
```
# 1. "beta" column
beta <- seq(0,9/10,by=0.01)

# 2. "sample_X" column
# 3. "samplemean" column
sample_list <- list()
sample_mean <- c()
for(value in beta){
  sample_X <- sample_X_0310(1/10,value,100)
  sample_mean <- c(sample_mean,mean(sample_X,na.rm=TRUE))
  sample_list <- append(sample_list, list(paste(sample_X, collapse=", ")))
}
result <- data.frame(beta=beta, sample_X=unlist(sample_list), samplemean=sample_mean)

result <- result %>% mutate(Expectation=3*(1/10)+10*beta)
```

3.1 Q7

```
result %>% ggplot(aes(x=beta)) + geom_point(aes(y=samplemean, color="samplemean")) + geom_point(aes(y=Expectation,color="Exp
ectation")) + xlab("beta") + ylab("value")
```



3.2 Exponential distribution

3.2 Q1

$\int_{-\infty}^{\infty} p(x)dx = 1 \int_{-\infty}^{\infty} p(x)dx = \int_{-\infty}^{\infty} 0dx + \int_{-\infty}^{\infty} \lambda e^{-\lambda x} dx \; Then, \quad \int_{-\infty}^{\infty} \lambda e^{-\lambda x} dx = [-e^{-\lambda x}]_0^{\infty} = (0 - (-1)) = 1$

3.2 Q2

```
my_cdf_exp <- function(x,lambda){
  Fx <- 1-exp(-lambda*x)
  if (Fx < 0){
    Fx <- 0
  }
  return (Fx)
}
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){
  return (-log(1-p)/lambda)
}

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=lambda))
all.equal(my_quantile_output,inbuilt_quantile_output)
```

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

3.2 Q4

$\mathbb{E}(X) = \int_{-\infty}^{\infty} x p_\lambda(x) dx = \int_0^{\infty} x\lambda e^{-\lambda x} dx = 1/\lambda$

$Var(X) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2 = 1/\lambda^2$

3.3 The Binomial distribution and the central limit theorem

3.3 Q1

$E(Z) = np, \quad Var(Z) = np(1-p)$

3.3 Q2

```
x <- seq(0,50)
pmf <- dbinom(x,50,7/10)
binom_df <- data.frame(x=x,pmf=pmf)
binom_df %>% head(3)
```

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

3.3 Q3

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

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

3.3 Q4

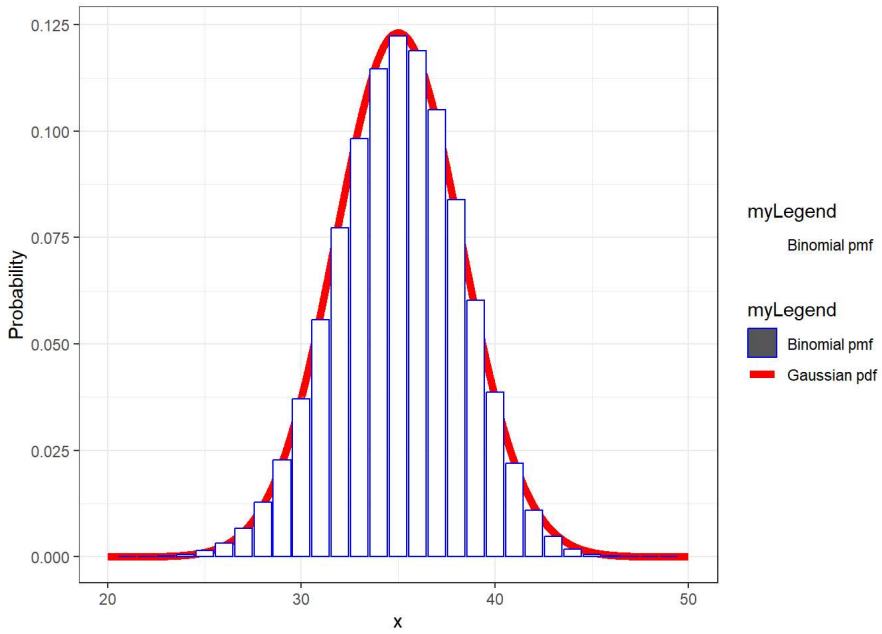
```
colors <- c("Gaussian pdf"="red","Binomial pmf"="blue")
fill <- c("Gaussian pdf"="white","Binomial pmf"="white")

ggplot() + labs(x="x",y="Probability") + theme_bw() + geom_line(data=gaussian_df, aes(x,y=pdf,color="Gaussian pdf"), size=2)
+ geom_col(data=binom_df, aes(x=x,y=pmf,color="Binomial pmf", fill="Binomial pmf")) + scale_color_manual(name="myLegend", va
lues=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 range
## (`geom_line()`).
```

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


0.125
0.100
0.075
0.050
0.025
0.000
Probability
20
30
40
50
x
myLegend
Binomial pmf
myLegend
Binomial pmf
Gaussian pdf