Assignment-5

Thrisha Rajkumar

2024-12-29

Assignment 5

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

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

1. Exploratory data analysis

```r
head(Hawks,10)
```

```
##    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
## 7      9  28 1992       13:30                877-76320      RT   I      370
## 8      9  29 1992       11:45                877-76321      RT   A      375
## 9      9  29 1992       15:35                877-76322      RT   A      412
## 10     9  30 1992       13:45               1207-55911      RT   I      405
##    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
## 7      960    25.3    30.1   212           NA      NA          NA       NA    NA
## 8      855    27.2    30.0   243           NA      NA          NA       NA    NA
## 9     1210    29.3    31.3   210           NA      NA          NA       NA    NA
## 10    1120    26.0    30.2   238           NA      NA          NA       NA    NA
```

1.1 Location estimators (Q1)

```
HawksTail = Hawks$Tail
head(HawksTail)
```

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

```
print(mean(HawksTail))
```

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

```
print(median(HawksTail))
```

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

1.2 Combining location estimators with the summarise function

(Q1)

```
summarise(Hawks, 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))
```

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

(Q2)

```
group_by(Hawks, 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))
```

```
## # A tibble: 3 x 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 Location and dispersion estimators under linear transformations

(Q1)

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

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

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

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

(Q2)

```r
var(HawksTail)*4
```

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

```r
var(HawksTail*2+3)
```

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

```r
sd(HawksTail)*2
```

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

```r
sd(HawksTail*2+3)
```

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

1.4 Robustness of location estimators

```r
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
```

```r
mean(corrupted_hal)
```

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

```r
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))
}
```

(Q1) Sample median:

```r
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))
}
```

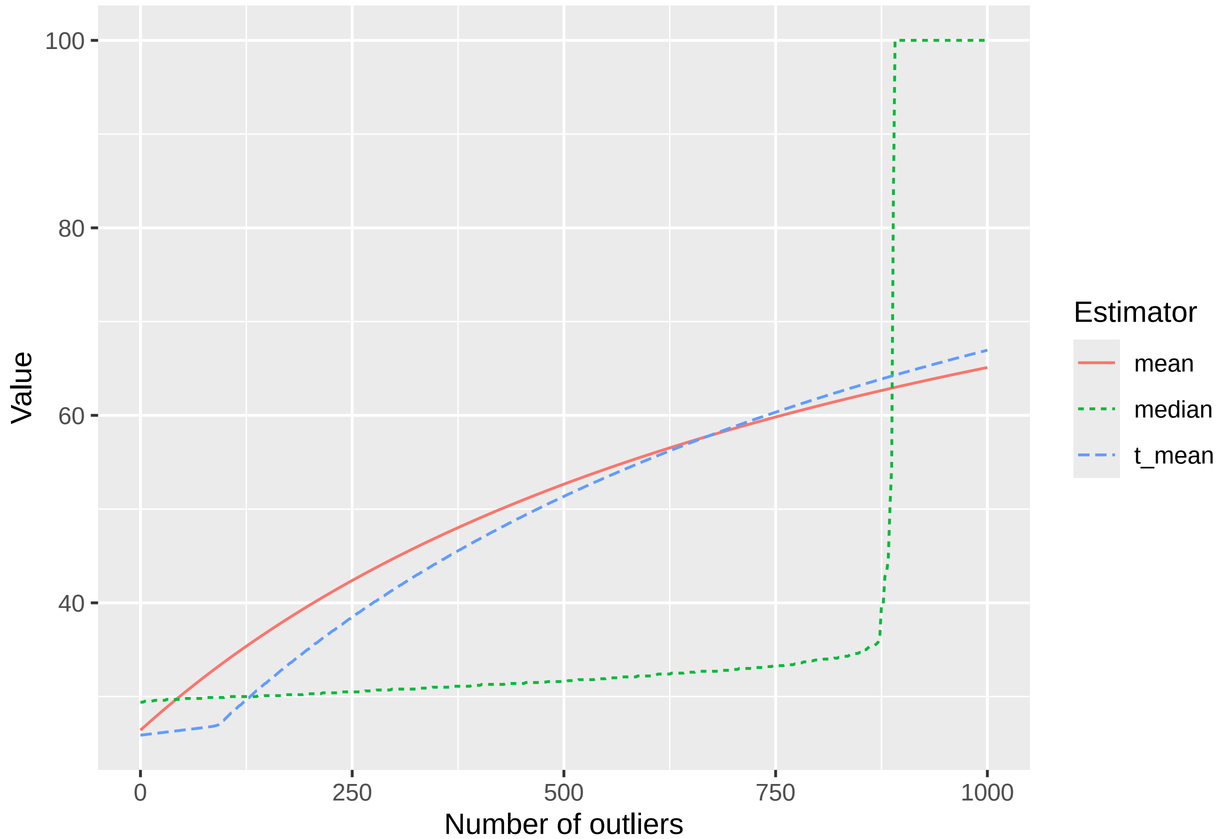
(Q2) Sample trimmed mean:

```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))
}
```

(Q3) Visualisation

```r
df_means_medians <- data.frame(num_outliers=num_outliers_vect, mean=means_vect, t_mean=t_means_vect, me
```

```r
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")
```

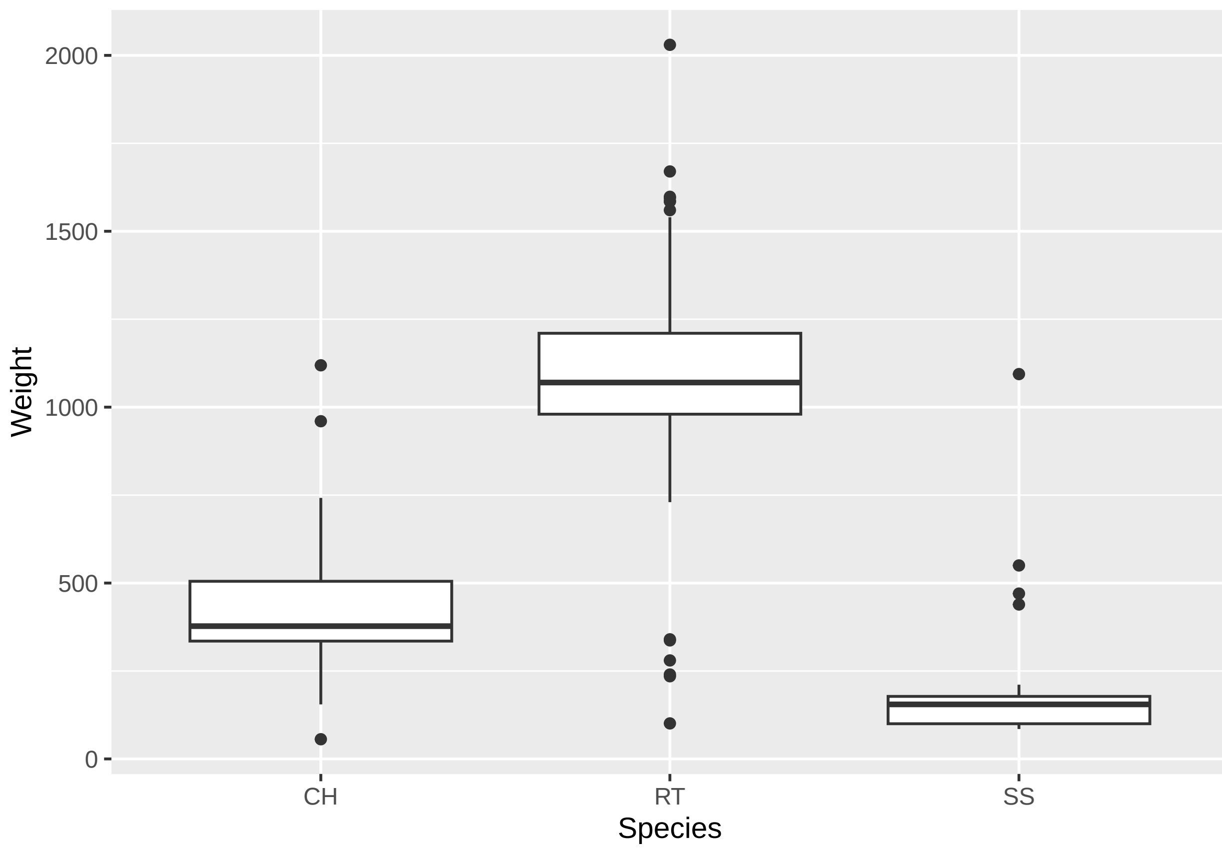


1.5 Box plots and outliers

(Q1)

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

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



(Q2) quantile and boxplots

```
group_by(Hawks, Species) %>%
summarise(quantile025=quantile(Weight, probs=0.25, na.rm=TRUE),
quantile050=quantile(Weight, probs=0.5, na.rm=TRUE),
quantile075=quantile(Weight, probs=0.75, na.rm=TRUE))
```

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

(Q3) Outliers

```
num_outliers <- function(x){
q25 <- quantile(x, 0.25, na.rm=TRUE)
q75 <- quantile(x, 0.75, na.rm=TRUE)
iq_range <- q75 - q25
num <- sum( (x>q75+1.5*iq_range)|(x<q25-1.5*iq_range), na.rm=TRUE )
return (num)
```

```
}

num_outliers( c(0, 40,60,185))
```

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

(Q4) Outliers by group

```
group_by(Hawks, Species) %>%
summarise(num_outliers_weight = num_outliers(Weight))
```

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

1.6 Covariance and correlation under linear transformations

(Q1)

```
cov(Hawks$Weight, Hawks$Wing, use='complete.obs')
```

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

```
cor(Hawks$Weight, Hawks$Wing, use='complete.obs')
```

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

(Q2)

(Q1).

```
cov(Hawks$Weight, Hawks$Wing, use='complete.obs')*2.4*(-1) -
cov(Hawks$Weight*2.4+7.1, Hawks$Wing*(-1)+3, use='complete.obs')
```

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

```
cor(Hawks$Weight, Hawks$Wing, use='complete.obs')*sign(2.4*(-1)) -
cor(Hawks$Weight*2.4+7.1, Hawks$Wing*(-1)+3, use='complete.obs')
```

```
## [1] 1.110223e-16
```

2. Random variables and discrete random variables

Random Variables and Discrete Random Variables

Expectation

The expectation $\mathbb{E}(X)$ of the random variable X is defined by

$$\mathbb{E}(X) := \sum_{x \in \mathbb{R}} x \cdot p(x).$$

Linearity of Expectation

Given random variables $X_1, X_2, \ldots, X_n$ and numbers $\alpha_1, \alpha_2, \ldots, \alpha_n$

So,

$$\mathbb{E}(\alpha X) = \alpha \mathbb{E}(X).$$

Equivalent Condition for Independent Random Variables

Let $X_1, X_2, \ldots, X_n : \Omega \to \mathbb{R}$ be a sequence of random variables. Then $X_1, X_2, \ldots, X_n$ are independent if and only if the following relationship holds for every sequence of well-behaved functions $f_1, f_2, \ldots, f_n$:

$$\mathbb{E}[f_1(X_1) \ldots f_n(X_n)] = \mathbb{E}[f_1(X_1)] \cdot \mathbb{E}[f_2(X_2)] \cdot \cdots \cdot \mathbb{E}[f_n(X_n)].$$

2.1 Expectation and Variance

(Q1) Covariance Between Independent Random Variables Suppose X and Y are independent. The covariance between X and Y is defined by

$$\operatorname{Cov}(X, Y) := \mathbb{E}[(X - \mathbb{E}[X]) \cdot (Y - \mathbb{E}[Y])].$$

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

Since X and Y are independent, we use the linearity of expectation and the equivalent condition for independent random variables:

$$\mathbb{E}[XY] = \mathbb{E}[f(X)f(Y)] = \mathbb{E}[f(X)] \cdot \mathbb{E}[f(Y)] = \mathbb{E}[X] \cdot \mathbb{E}[Y].$$

$$\operatorname{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X] \cdot \mathbb{E}[Y] = 0.$$

2.2 Distributions

$$\mathbb{P}(X = 3) = \alpha, \quad \mathbb{P}(X = 10) = \beta, \quad \mathbb{P}(X \notin \{0, 3, 10\}) = 0.$$

(Q1) Expectation and Variance of a Discrete Random Variable

1. **Probability Mass Function (PMF)**: The probability mass function $p(x)$ of X is

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

2. **Expectation of X**: The expectation of X is

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

3. **Variance of X**: The variance of X is

$$\mathrm{Var}(X) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2.$$

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

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

Now, the variance is

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

4. **Standard Deviation of X**: The standard deviation is the square root of the variance:

$$\mathrm{SD}(X) = \sqrt{\mathrm{Var}(X)}.$$

(Q2) Distribution and Distribution Function

1. **Distribution of X**: The distribution of X is

$$P(S) = (1-\alpha-\beta)\nVdash(0) + \alpha\nVdash(3) + \beta\nVdash(10).$$

2. **Distribution Function of X**: The distribution function $F(x)$ of X is

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

(Q3) Variance and Covariance The variance of Y is the sum of the variances of the independent random variables:

$$\text{Var}(Y) = n \cdot \text{Var}(X) = n \cdot (9\alpha + 100\beta - 9\alpha - 100\beta - 60\alpha\beta).$$

Q4.

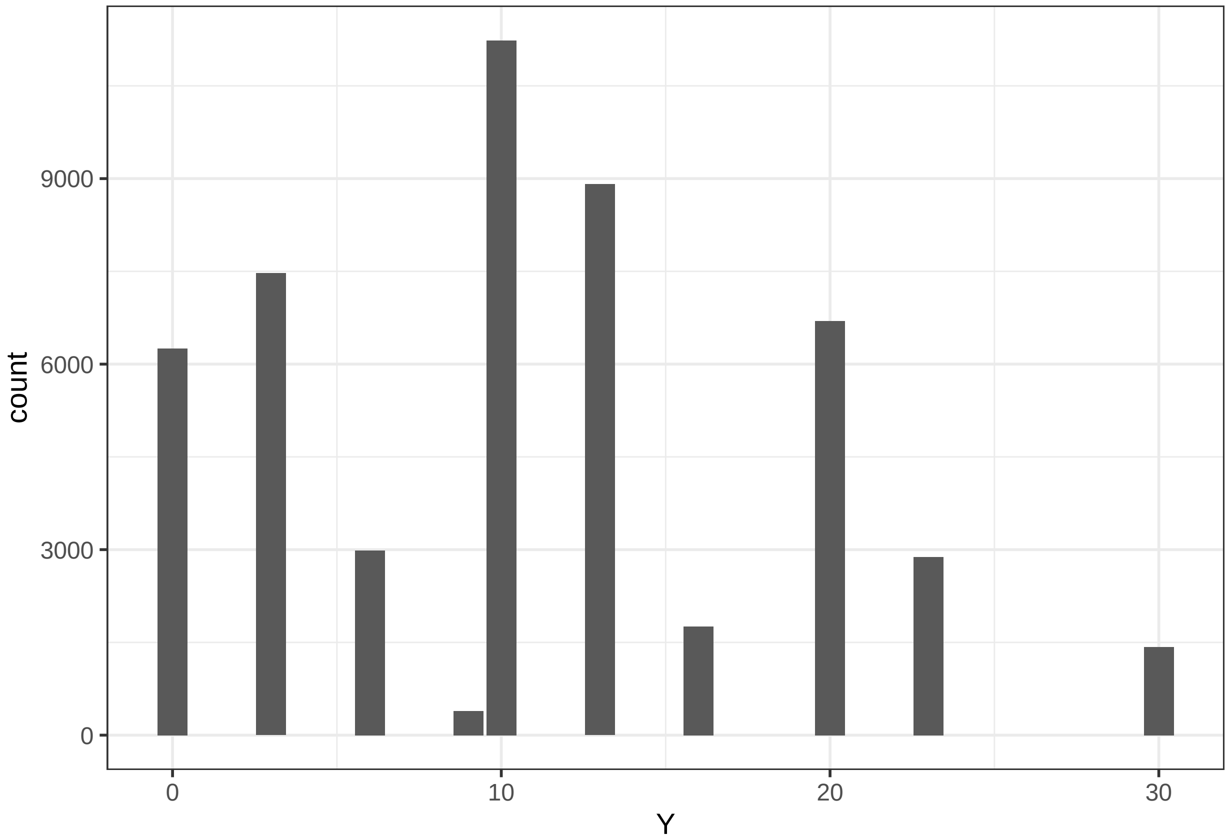
```r
Gen_X_numbers <- function(n){
Uniform <- runif(n)
X = 0*(Uniform<0.5) + 3 * ( (Uniform>=0.5)*(Uniform<0.7) ) + 10 *
(Uniform>0.7)
return (X)
}
set.seed(1002)

Gen_X_numbers(4)
```

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

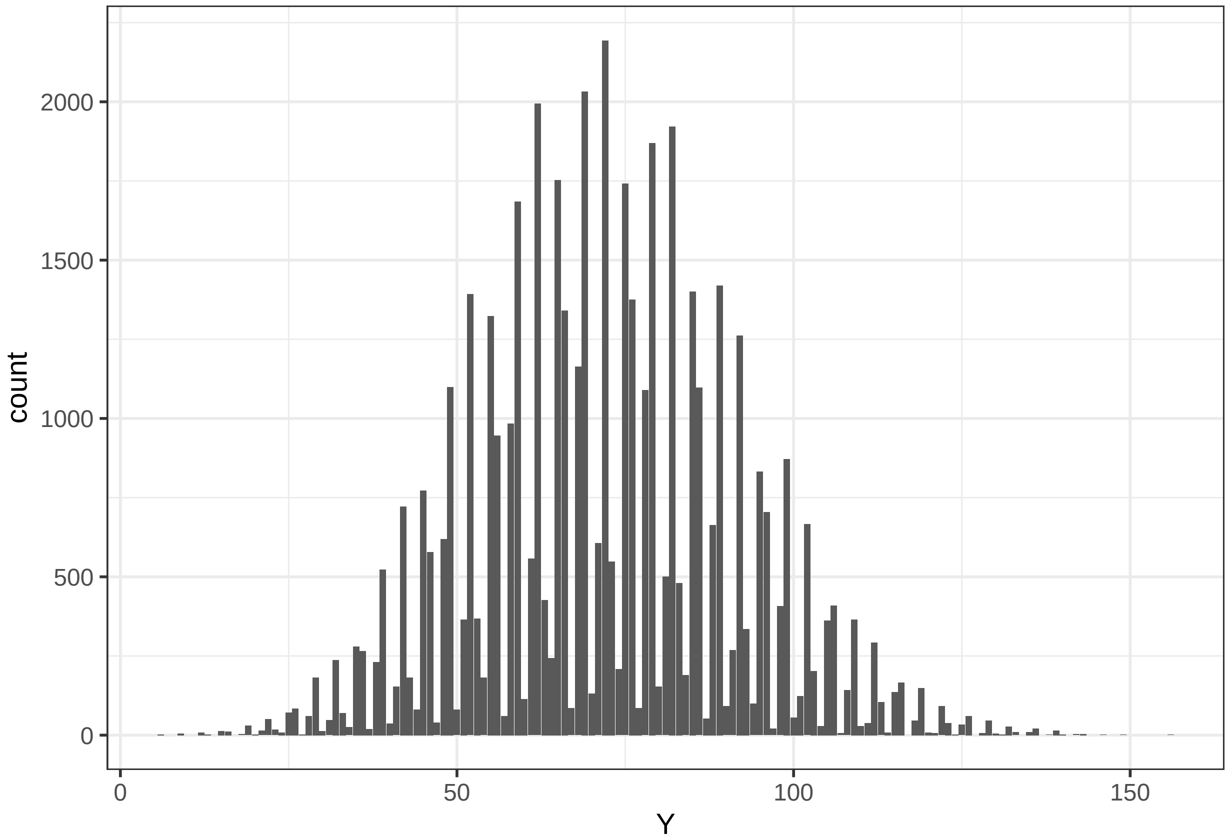
```r
Gen_Y_samples <- function(m,n){
Y_sample <- data.frame(index=seq(m)) %>%
mutate(Y=map_dbl(index, ~ sum(Gen_X_numbers(n)) ))
return (Y_sample)
}

Gen_Y_samples(5, 2)
```

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

```r
# Visualization
samples_Y <- Gen_Y_samples(50000, 3)
ggplot(samples_Y, aes(Y)) + geom_bar() + theme_bw()
```




```
samples_Y <- Gen_Y_samples(50000, 20)
ggplot(samples_Y, aes(Y)) + geom_bar() + theme_bw()
```



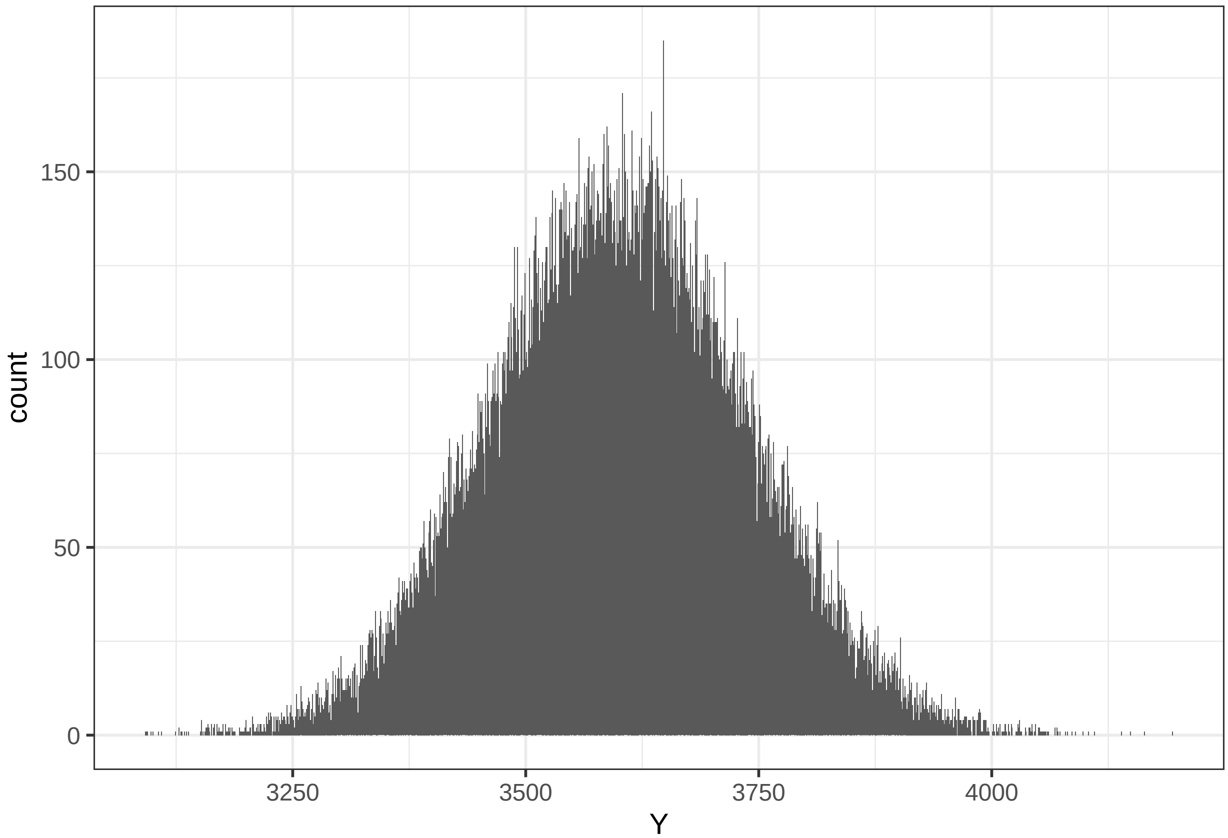

```
print(range(samples_Y))
```

```
## [1]     1 50000
```

```
print(diff(range(samples_Y)))
```

```
## [1] 49999
```

```
samples_Y <- Gen_Y_samples(50000, 1000)
ggplot(samples_Y, aes(Y)) + geom_bar() + theme_bw()
```



3. Continuous random variables and limit laws

3.1 Simulating data with the uniform distribution

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)
```

Q3.

```
sample_X_0310 <- function(alpha, beta, n){
sample_X <- data.frame(U=runif(n)) %>%
mutate(X=case_when(
(0<=U)&(U<alpha)~3,
(alpha<=U)&(U<alpha+beta)~10,
```

```
(alpha+beta<=U)&(U<=1)~0)) %>%
pull(X)
return (sample_X)
}
```

Q4.

```
n <- 10000
alpha <- 1/2
beta <- 1/10
sample_X <- sample_X_0310(alpha,beta,n)
mean(sample_X)
```

```
## [1] 2.5135
```

Q5.

```
var(sample_X)
```

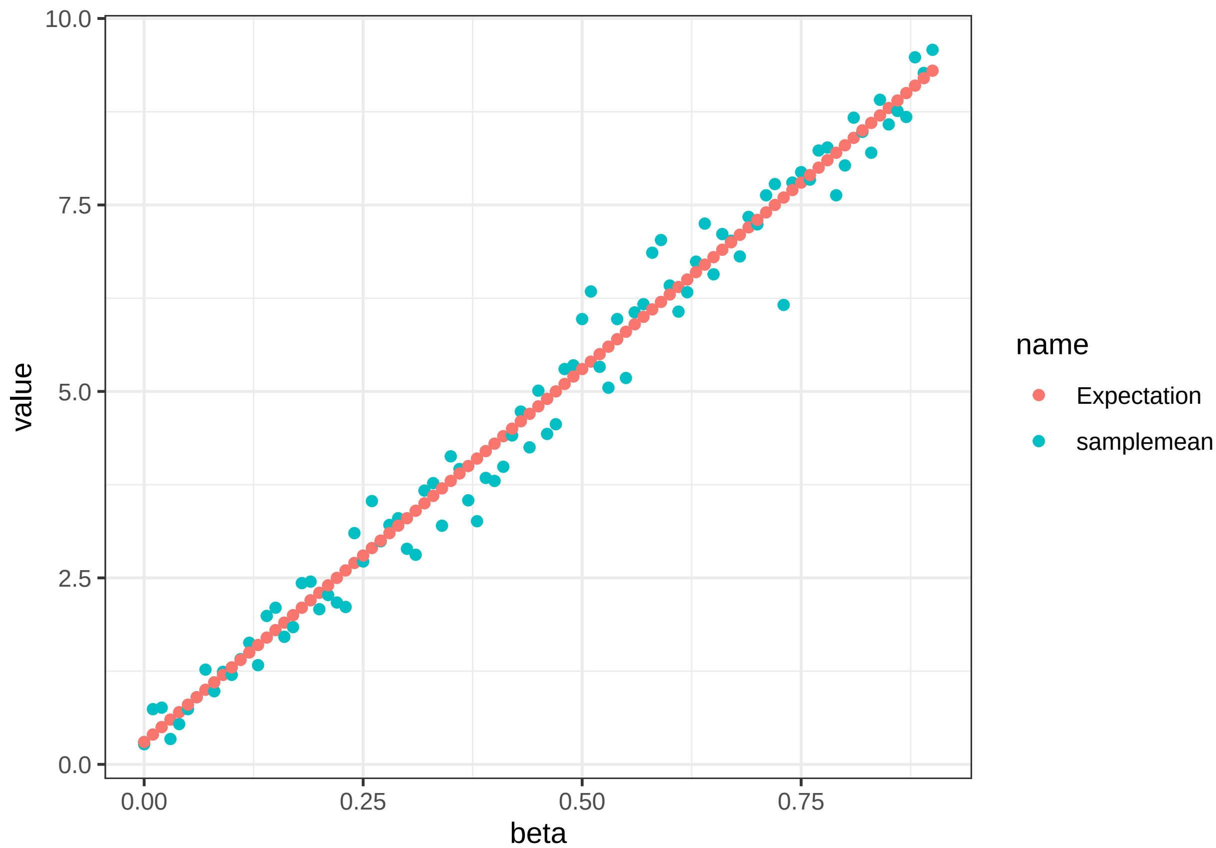
```
## [1] 8.328651
```

Q6.

```
n = 100
alpha = 1/10
samples <- data.frame(beta = seq(0, 9/10, 0.01)) %>%
mutate( sample_X =  map(beta, ~sample_X_0310(alpha,.x,n) )) %>%
mutate( samplemean = map_dbl(sample_X, mean) ) %>%
mutate( Expectation = 3*alpha + 10*beta)
```

Q7.

```
df <- samples %>% pivot_longer(cols=c(samplemean, Expectation),
names_to = 'name', values_to = 'value')
ggplot(df, aes(x=beta, y=value, color=name)) +
geom_point()  + theme_bw()
```



3.2 Exponential distribution

Q2.

```
my_cdf_exp <- function(x, lambda){
if (x<0) {return (0)}
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
```

Q3.

```
my_quantile_exp <- function(p, lambda){
  if (p<=0) return (-Inf)
  return (log(1-p)/(-lambda))
  }
test_inputs <- seq(0.01, 0.99, 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
```

3.3 The Binomial distribution and the central limit theorem

Q2.

```
binom_df <- data.frame(x = seq(0,50)) %>%
mutate(pmf = map_dbl(x, ~dbinom(.x, size=50, prob=0.7)) )
head(binom_df, 3)
```

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

Q3.

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

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

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", 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 range
## (‘geom_line()‘).

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