Project #4: Fitting probabilistic models

Ben Cartwright, Heewon Huh, Sally Kyong, Harvey Lee

2021-10-23

```
#library(nimble)
```

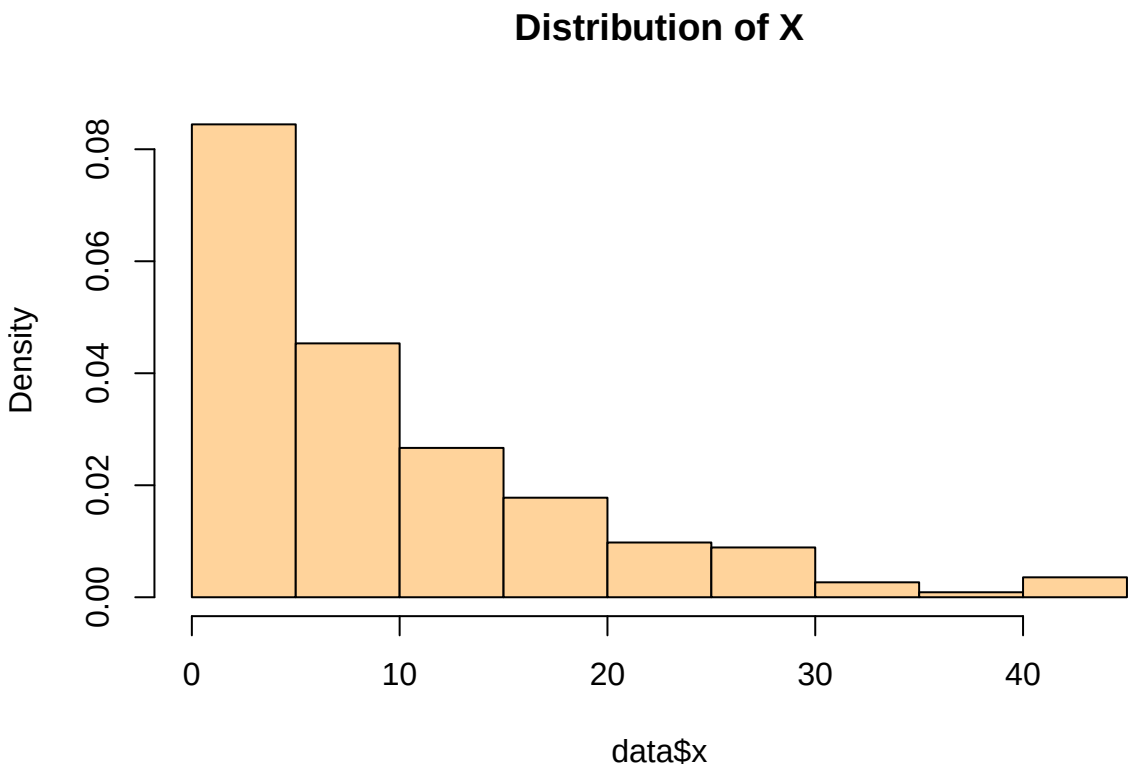
Problem #1. (10 points)

Read in the data from the "mystery-data.csv" file. Your goal is to fit a probabilistic model to the data and visually substantiate your choices. More precisely, you need to do the following:

```
data = read.csv("mystery-data.csv")
```

- **(2 points)** visually display your data;

```
hist(data$x, prob = TRUE, breaks = 6, main = "Distribution of X", col = "burlywood1")
```



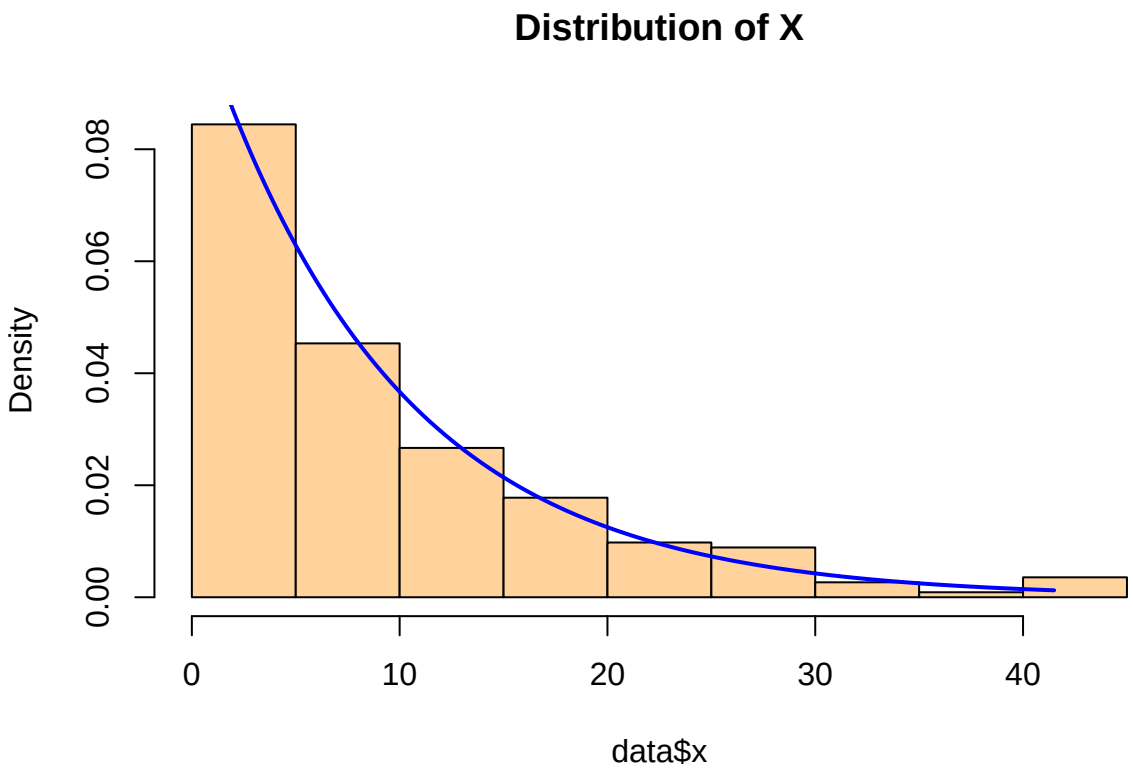
- **(2 points)** propose a **named parametric distribution** to fit to your data and justify your choice; We think the data is exponentially distributed because it starts with its peak at the left side, decreases at a fast rate and starts to level off towards the tail which is the characteristic of the exponential distribution.
- **(2 points)** using the data, propose a point estimate for any parameters in your model;

```
# The parameter of an exponential distribution, lambda, is a reciprocal of the mean of
# the distribution.
lambda<- 1/mean(data$x)
print(lambda)
## [1] 0.1076679
```

$X \sim Exp(\lambda = 0.1076679)$

- **(4 points)** superimpose the appropriate graph for your model onto the appropriate graph of the data to convince your reader that your model is valid.

```
hist(data$x, prob = TRUE, breaks = 6, main = "Distribution of X", col = "burlywood1")
x<- seq(min(data$x), max(data$x), length = length(data$x))
y<- dexp(x, rate = lambda)
lines(x, y, col="blue", ylim=c(0,0.15), lwd = 2)
```



Problem #2 (14 points)

Here, the goal is to provide a probabilistic model for some of the data available here:

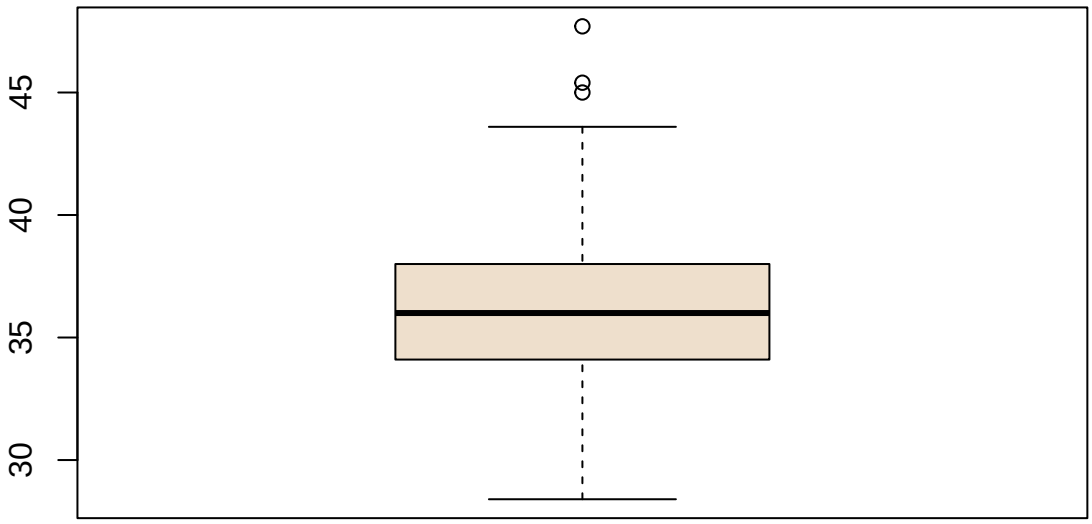
Data set associated with the textbook

First, download the data set and import it into R. Do not forget that the documentation for the data set is also available under the above link.

```
data <- read.csv("bdims.csv")
```

(2 points) Focus on the measurements of the male respondents' calves' maximum girth in centimeters. Does this set of measurements have any outliers?

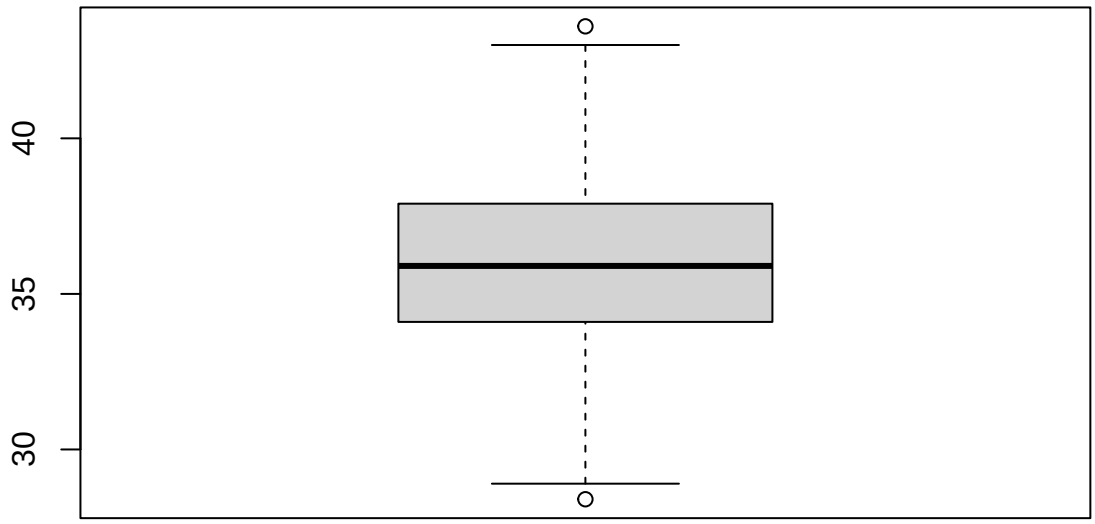
```
male_calve_data <- filter(data$cal_gi, data$sex[1])
boxplot(male_calve_data, col= "antiquewhite2")
```



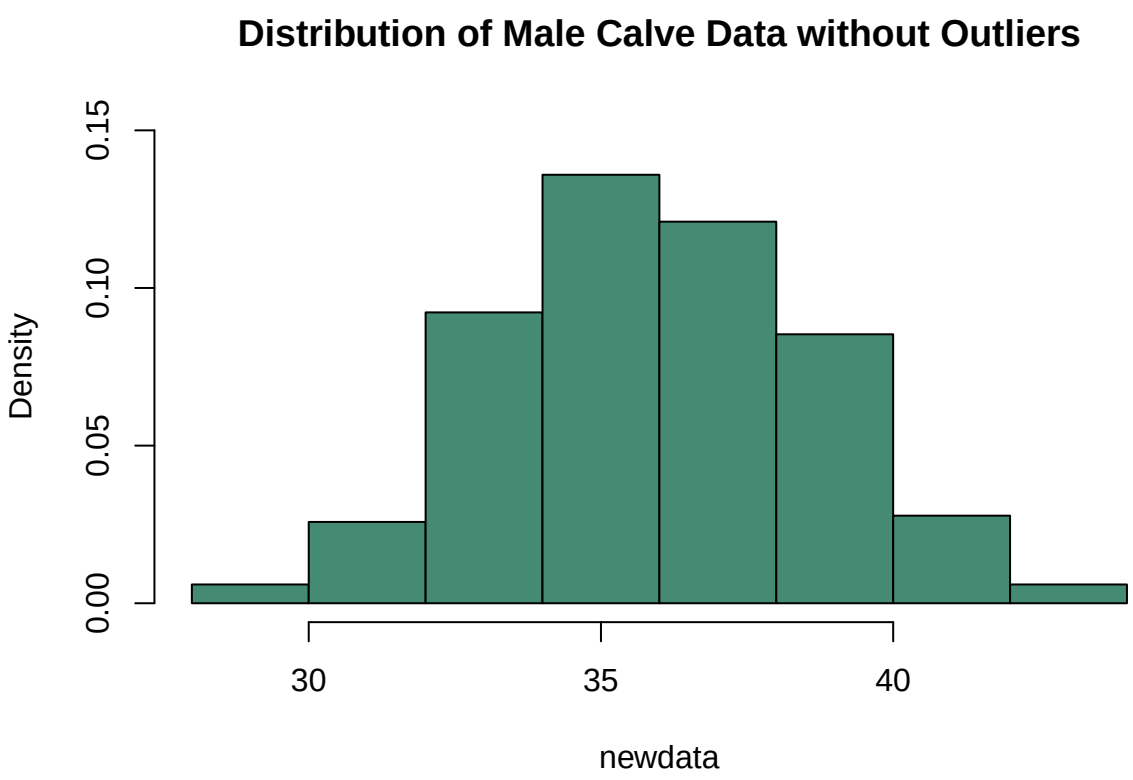
There are three outliers above the 3rd quartile.

(4 points) If there are outliers, discard them from your data set. Visualize the remaining data points appropriately.

```
outliers = boxplot(male_calve_data, plot = FALSE)$out
newdata <- male_calve_data[-which(male_calve_data %in% outliers)]
boxplot(newdata)
```




```
hist(newdata, ylim=c(0,0.15), prob = TRUE, main = "Distribution of Male Calve Data without Outliers",
     col="aquamarine4")
```



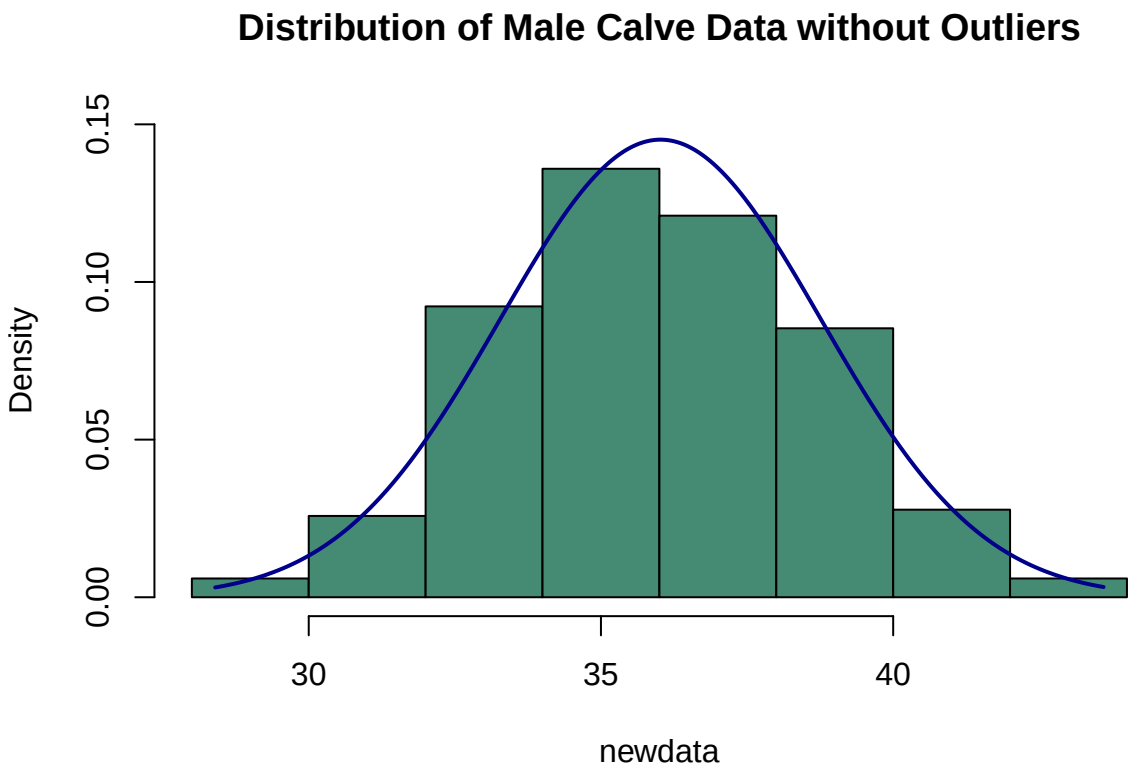
(2 points) Propose a **named parametric distribution** to fit to your data and justify your choice. We think the data is normally distributed because the center of the distribution is about the middle of the distributions and the distirbution is overall symmetrical with bell-shaped curve. **(2 points)** Using the data, propose a point estimate for any parameters in your model.

```
# The parameters of a normal distribution is the mean and the standard deviation of the data.
mean<- mean(newdata)
sd<- sd(newdata)
print(mean)
## [1] 36.01905
print(sd)
## [1] 2.748546
```

The distribution of data is $Normal(\mu = 36.01905, \sigma = 2.748546)$

(4 points) Superimpose the appropriate graph for your model onto the appropriate graph of the data to convince your reader that your model is valid.

```
hist(newdata, ylim=c(0,0.15), prob = TRUE, main = "Distribution of Male Calve Data without Outliers",
     col="aquamarine4")
x<- seq(min(newdata), max(newdata), length = length(newdata))
y<- dnorm(x, mean = mean(newdata), sd = sd(newdata))
lines(x,y, col="darkblue", ylim=c(0,0.15), lwd = 2)
```



Problem #3 (22 points)

Here, the goal is to provide a probabilistic model for some of the data available here:

Data set associated with the textbook

First, download the data set and import it into R. Do not forget that the documentation for the data set is also available under the above link.

```
quakes <- read.csv("earthquakes.csv")
```

We want to construct a probabilistic model for the number of severe earthquakes in a year based on this data set.

(6 points) Create a `table` which contains the number of times that a particular yearly number of earthquakes was recorded in the above data set.

```
#earthquake per year table
eqpy <- table(quakes$year)
count_earthquakes<-table(eqpy)
columns<- c("Earthquake per Year", "Count")
knitr::kable(count_earthquakes, col.names = columns)
```

Earthquake per Year	Count
1	34
2	25
3	8
4	1
5	1
6	1

(2 points) We must not forget that in the years not mentioned in the data set, there were no earthquakes. Add this information to the table you created above. Do not forget to include the appropriate name to the added observation.

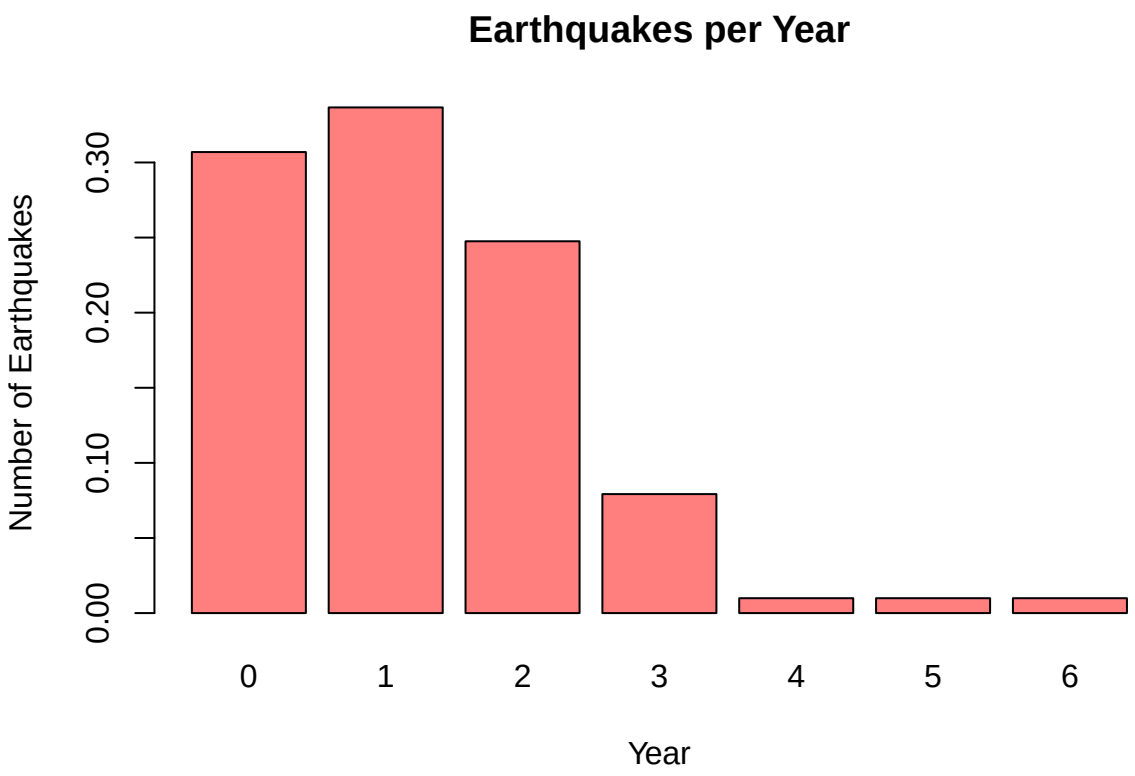
```
#Adding years with 0 earth quakes
quake_occur = quakes$year
years = seq(1900, 2000)
count_vector = c(1:101)
for(i in count_vector){
  count_vector[i] = 0
}

for( i in quake_occur){
  x = count_vector[i - 1899]
  count_vector[i - 1899] = x + 1
}
count_table = table(count_vector)
data = prop.table(count_table)
columns<- c("Earthquake per Year", "Count")
knitr::kable(count_table, col.names = columns)
```

Earthquake per Year	Count
0	31
1	34
2	25
3	8
4	1
5	1
6	1

(2 points) Choose an appropriate plot and visualize the data in the table you obtained above.

```
#Displaying the Data
barplot(data, pch = 19, main = "Earthquakes per Year", ylab = "Number of Earthquakes",
        xlab = "Year", col=rgb(red=1,green= 0, blue = 0, alpha = 0.5))
```

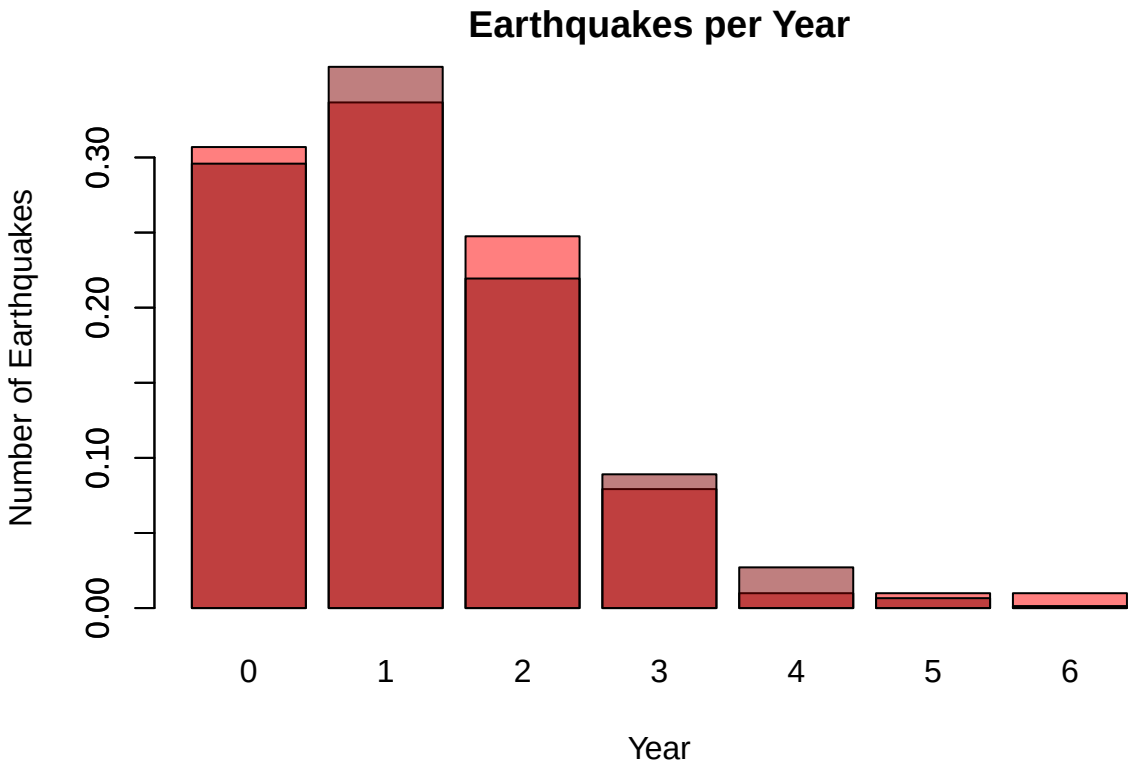


(6 points) Now, we want to construct a probabilistic model for these data. When it comes to counts of rare events, the **Poisson distribution** is oftentimes used. Before your proceed, review the Poisson distribution (Section 4.5 in the textbook and/or your *M362K Probability* notes). Then, familiarize yourself with the built-in commands related to the Poisson distribution in `R`. You can easily access the `help` by typing `dpois` in the console in RStudio. Now, fit the Poisson distribution to the data by estimating its parameter.

```
#Poisson Model
# The parameter of a Poisson distribution, lambda, is equal to the mean of the distribution
lambda = mean(count_vector)
x = dpois(c(0:6), lambda = lambda, log = FALSE)
```

(6 points) Superimpose the appropriate graph for your model onto the appropriate graph of the data to convince your reader that your model is valid.

```
#Fitting the Data
barplot(data, pch = 19, main = "Earthquakes per Year", ylab = "Number of Earthquakes",
        xlab = "Year", col=rgb(red=1,green= 0, blue = 0, alpha = 0.5))
barplot(x, add = TRUE, pch = 19, col=rgb(red=0.5,green= 0, blue = 0, alpha = 0.5) )
```



Problem #4 (14 points)

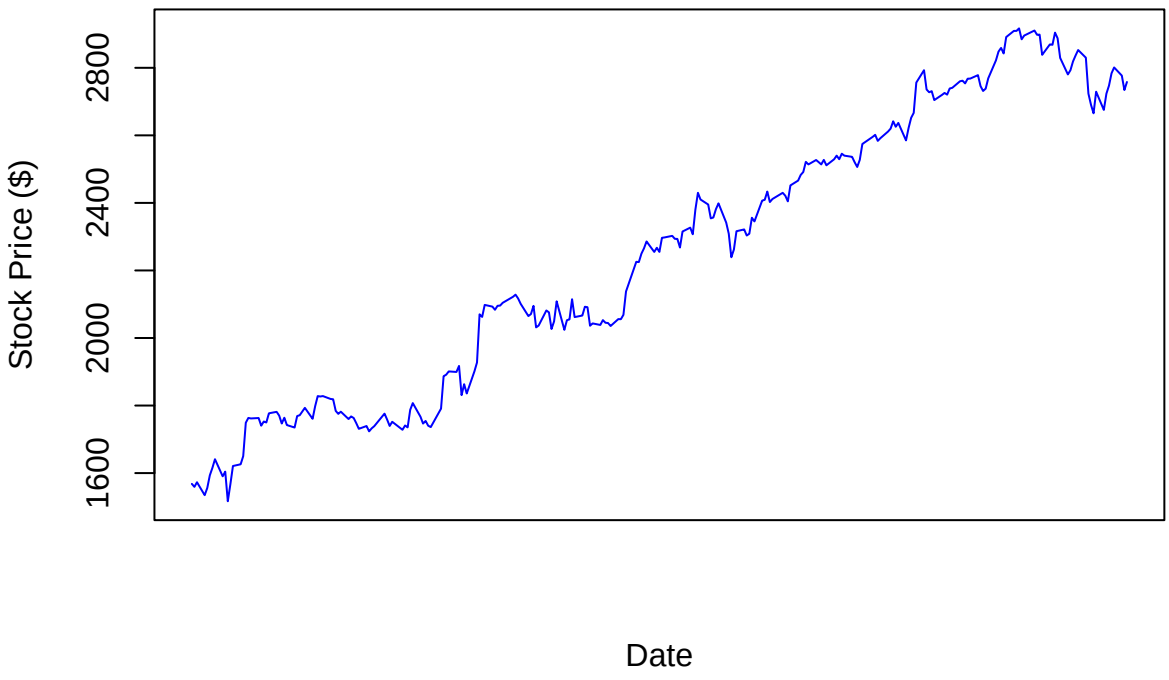
Go to Google historical data

Then, download the data for the last 252 (or so) trading days, i.e., for the last year.

```
stock <- read.csv("GOOG.csv")
```

(2 points) Draw the time-plot of the evolution of the closing stock price (not the adjusted). You do **not** need to put the calendar days on the horizontal axis, but you **do** need to label your axes and give your time-plot a title indicating the dates.

```
plot(as.Date(stock$Date), stock$Close, pch = 16, type = "l",
     main = "Daily Closing Stock Price of Google from 10/14/2020 to 10/13/2021",
     xaxt = "n", xlab = "Date", ylab = "Stock Price ($)", col = "blue")
```



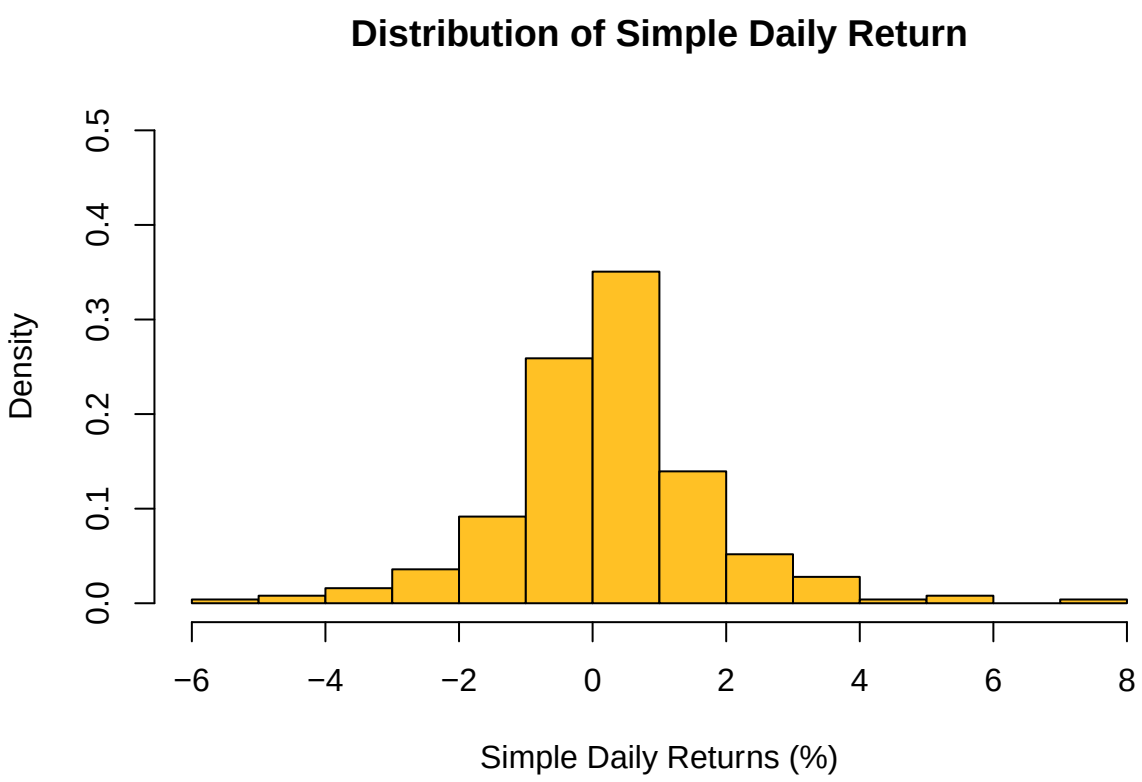
The **simple daily return** of the stock over a day indexed by t is defined as

$$\frac{\text{price at end of day } t - \text{price at end of day } (t-1)}{\text{price at end of day } (t-1)}$$

(4 points) Construct the vector of simple daily returns over the last year. Provide a visualization of the returns. What can you say about the characteristics of the distribution based on the above plot?

```
n<- NROW(stock$Close)
sdr<- c()
for (i in 1:(n-1)){
  daily.returns <- ((stock$Close[i+1]-stock$Close[i])/stock$Close[i])*100
  sdr<- c(sdr,daily.returns)
}

hist(sdr, breaks = 12, prob = TRUE, main = "Distribution of Simple Daily Return",
     xlab = "Simple Daily Returns (%)", ylim = c(0,0.5), col = "goldenrod1")
```

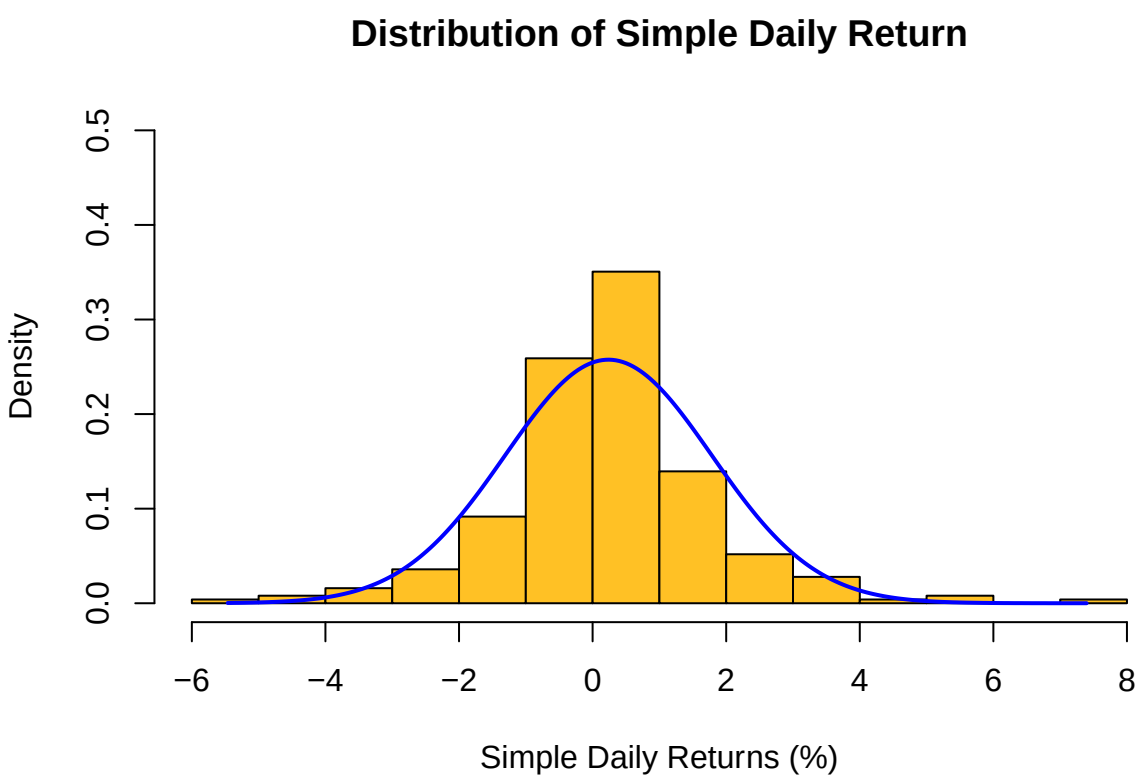


(3 points) Fit the normal distribution to the above returns. Superimpose the appropriate graph for your model onto the appropriate graph of the data to convince your reader that your model is valid.

```
# To fit the normal distribution to simple daily return,
# find the mean and the standard deviation of simple daily return.
mean(sdr)
## [1] 0.2371014
sd(sdr)
## [1] 1.549162
```

Model simple daily return as $Normal(\mu = 0.2371014, \sigma = 1.549162)$

```
hist(sdr, breaks = 12, prob = TRUE, main = "Distribution of Simple Daily Return",
     xlab = "Simple Daily Returns (%)", ylim=c(0,0.5), col = "goldenrod1")
x<- seq(min(sdr), max(sdr), length = length(sdr))
y<- dnorm(x, mean = mean(sdr),sd = sd(sdr))
lines(x, y, col="blue", ylim=c(0,0.5), lwd = 2)
```



(5 points) You will have to install a package to solve this part of your project. First, run the following in your console:

`install.packages('nimble')`

When that is finished, you need to run this in your uncomment

`library(nimble)`

from the first chunk in this document.

Next, you should learn more about the **Laplace (double exponential)** distribution. This is easily done by visiting:

Wikipedia: The Laplace distribution

Now, you are equipped to fit the Laplace (double exponential) distribution to the above returns. To learn about the parametrization of the Laplace distribution in `R`, type `?ddexp` into the console in RStudio.

After you have completed the fit, superimpose the appropriate graph for your model onto the appropriate graph of the data.

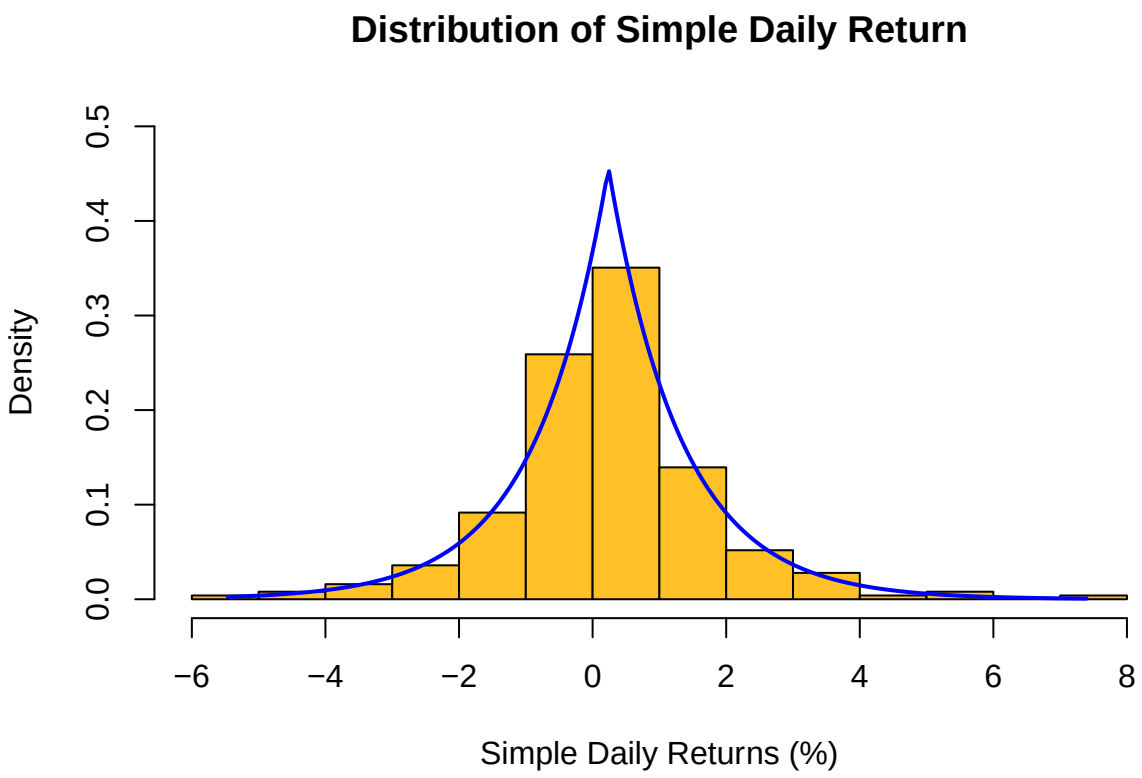
```
# To fit the Laplace distribution to simple daily return,
# find the location parameter and the scale parameter of simple daily return.
# The location parameter is same as the mean of simple daily return.
mean(sdr)
## [1] 0.2371014

# The scale parameter is the standard deviation divided by the square root of 2
```

```
sd(sdr)/sqrt(2)
## [1] 1.095423
```

Model simple daily return as $Laplace(\mu = 0.2371014, b = 1.095423)$

```
hist(sdr, breaks = 12, prob = TRUE, main = "Distribution of Simple Daily Return",
     xlab = "Simple Daily Returns (%)", ylim=c(0,0.5), col = "goldenrod1")
mu<-mean(sdr)
b<- sd(sdr)/sqrt(2)
install.packages('nimble', repos = "http://cran.us.r-project.org")
## Installing package into 'C:/Users/heewo/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)
## package 'nimble' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\heewo\AppData\Local\Temp\Rtmp4ApnJb\downloaded_packages
library(nimble)
## nimble version 0.12.1 is loaded.
## For more information on NIMBLE and a User Manual,
## please visit https://R-nimble.org.
##
## Attaching package: 'nimble'
## The following object is masked from 'package:stats':
##
##     simulate
x<- seq(min(sdr), max(sdr), length = length(sdr))
y<- ddexp(x, location = mu, scale = b)
lines(x, y, col="blue", ylim=c(0,0.5), lwd = 2)
```


Distribution of Simple Daily Return
Density
0.0
0.1
0.2
0.3
0.4
0.5
−6
−4
−2
0
2
4
6
8
Simple Daily Returns (%)