Project #4: Fitting probabilistic models

Antoinette Lynne, Harini Shanmugam, Aanya Shrivastava

2021-10-25

```
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
```

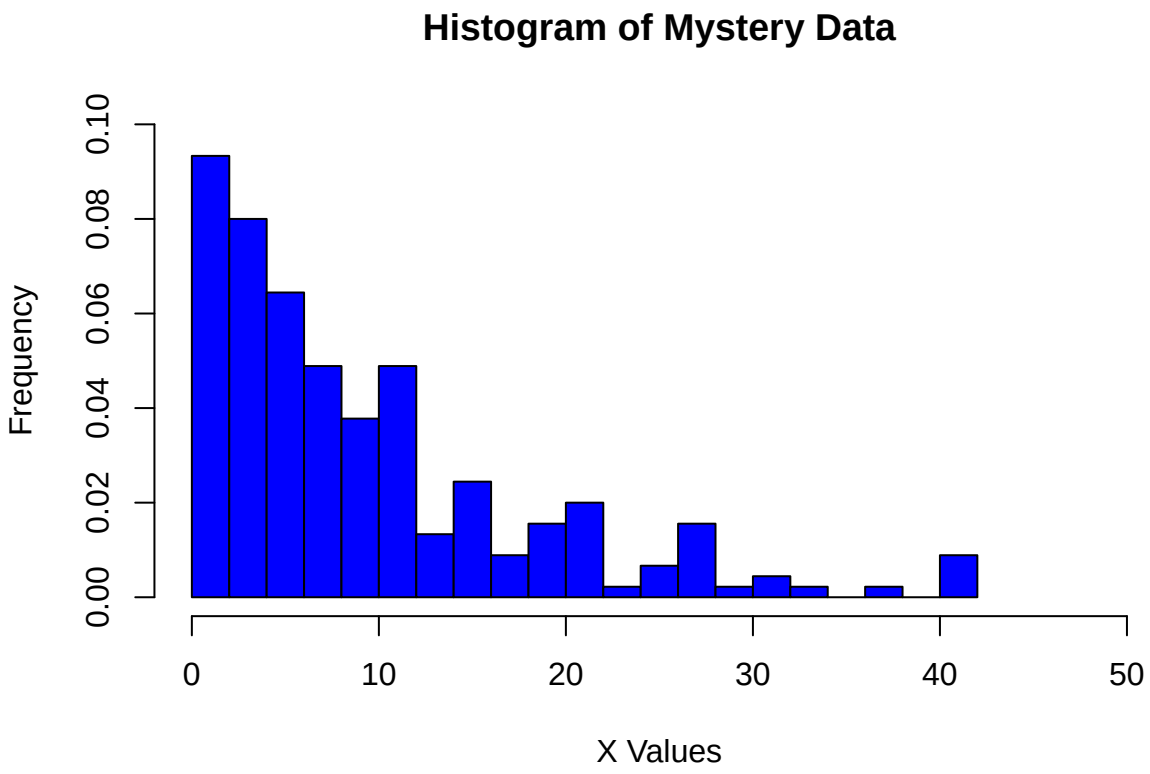
Problem #1. (10 points)

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:

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

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

```
hist(mysterydata$x,
     breaks=20,
     main="Histogram of Mystery Data",
     xlab="X Values",
     ylab="Frequency",
     xlim=c(0,50),
     ylim=c(0.00, 0.10),
     col="blue",
     prob=TRUE)
```



- **(2 points)** propose a **named parametric distribution** to fit to your data and justify your choice;

```
mean(mysterydata$x)
## [1] 9.287816
sd(mysterydata$x)
## [1] 8.813874
median(mysterydata$x)
## [1] 6.437853
```

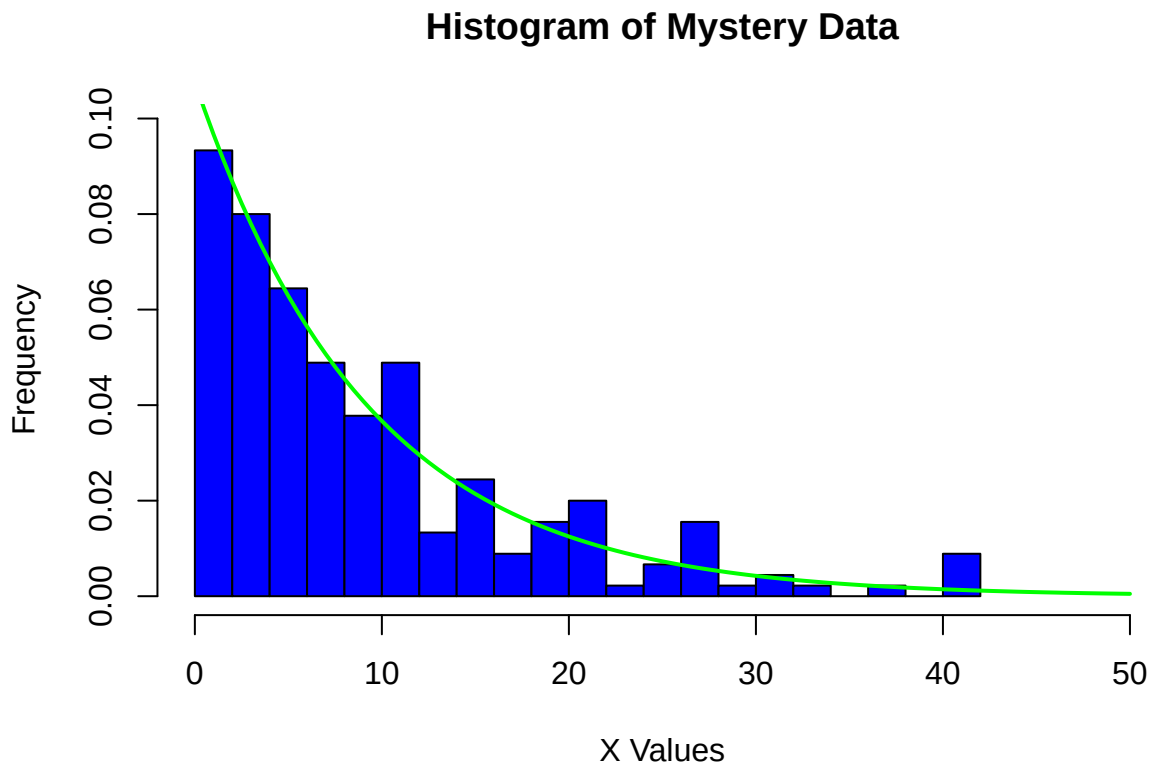
The distribution is approximately exponential. In an exponential distribution, the mean and standard deviation are equivalent. Here, the mean and standard deviation are nearly equivalent. The mean is actually a little higher, showing that the distribution is skewed right a little bit, which models an exponential distribution.

- **(2 points)** using the data, propose a point estimate for any parameters in your model;

The sample mean x is a point estimate of the population mean mu which equals mean(mysterydata$x) which equals 9.287816.

- **(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(mysterydata$x,
     breaks=20,
     main="Histogram of Mystery Data",
     xlab="X Values",
     ylab="Frequency",
     xlim=c(0,50),
     ylim=c(0.00, 0.099),
     col="blue",
     prob=TRUE)
curve(dexp(x, 1/mean(mysterydata$x)),
      add=TRUE,
      col="green",
      lwd=2)
```



Problem #2 (14 points)

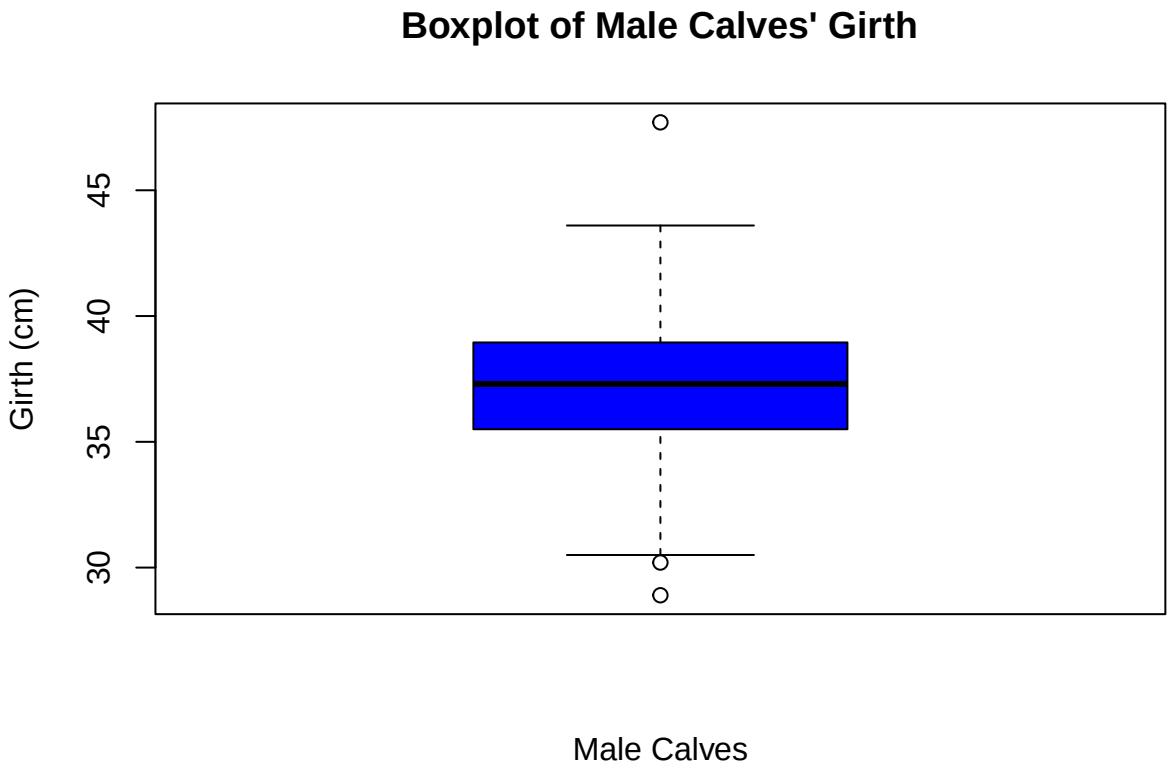
Here, the goal is to provide a probabilistic model for some of the data available here:

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.

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

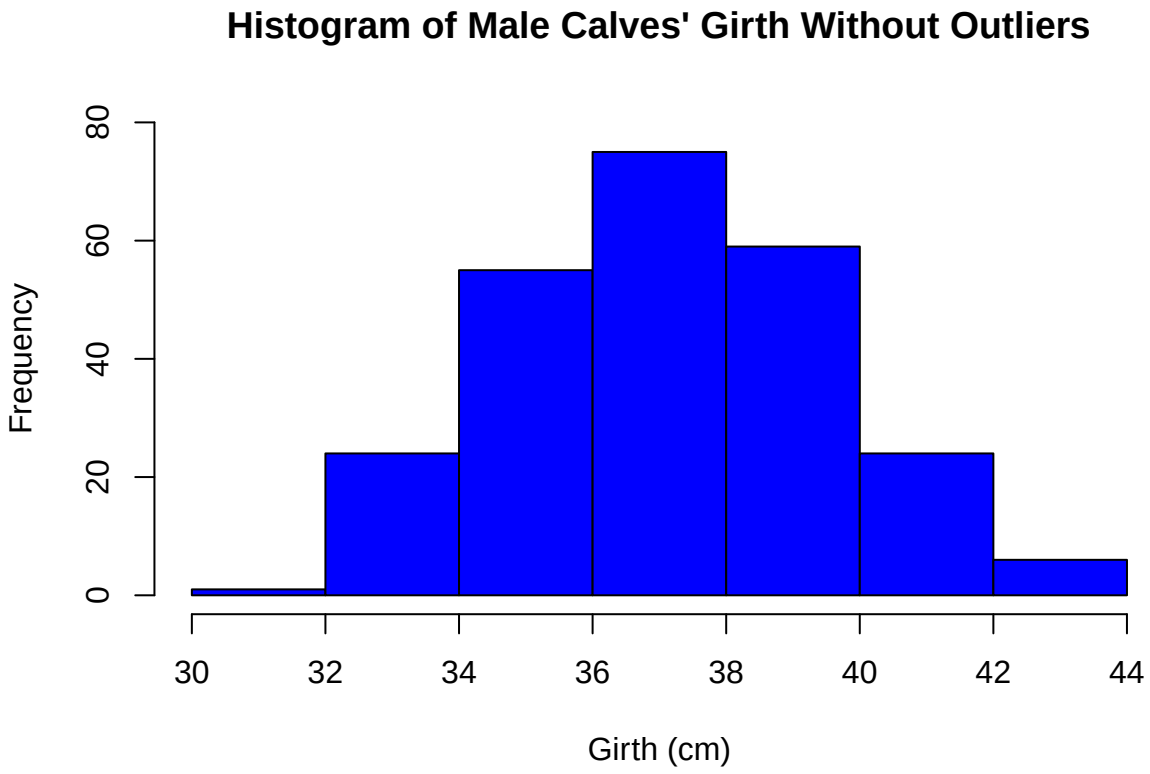
```
data_dim = read.csv("bdims.csv")
calves <- data_dim$cal_gi[data_dim$sex == 1]
boxplot(calves,
        main="Boxplot of Male Calves' Girth",
        xlab="Male Calves",
        ylab="Girth (cm)",
        col="blue")
```



Yes, there are outliers. There are 3 outliers to be specific (data points 47.7, 28.9, and 30.2)

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

```
dimensions = calves[calves != 47.7 & calves != 28.9 & calves != 30.2]

hist(dimensions,
     main="Histogram of Male Calves' Girth Without Outliers",
     xlab="Girth (cm)",
     ylab="Frequency",
     xlim=c(30,44),
     ylim=c(0,80),
     col="blue")
```



(2 points) Propose a **named parametric distribution** to fit to your data and justify your choice.

This data fits a normal distribution because the distribution is symmetrical. Moreover, the mean, median, and mode are all equal. So, it is evenly distributed above and below the mean.

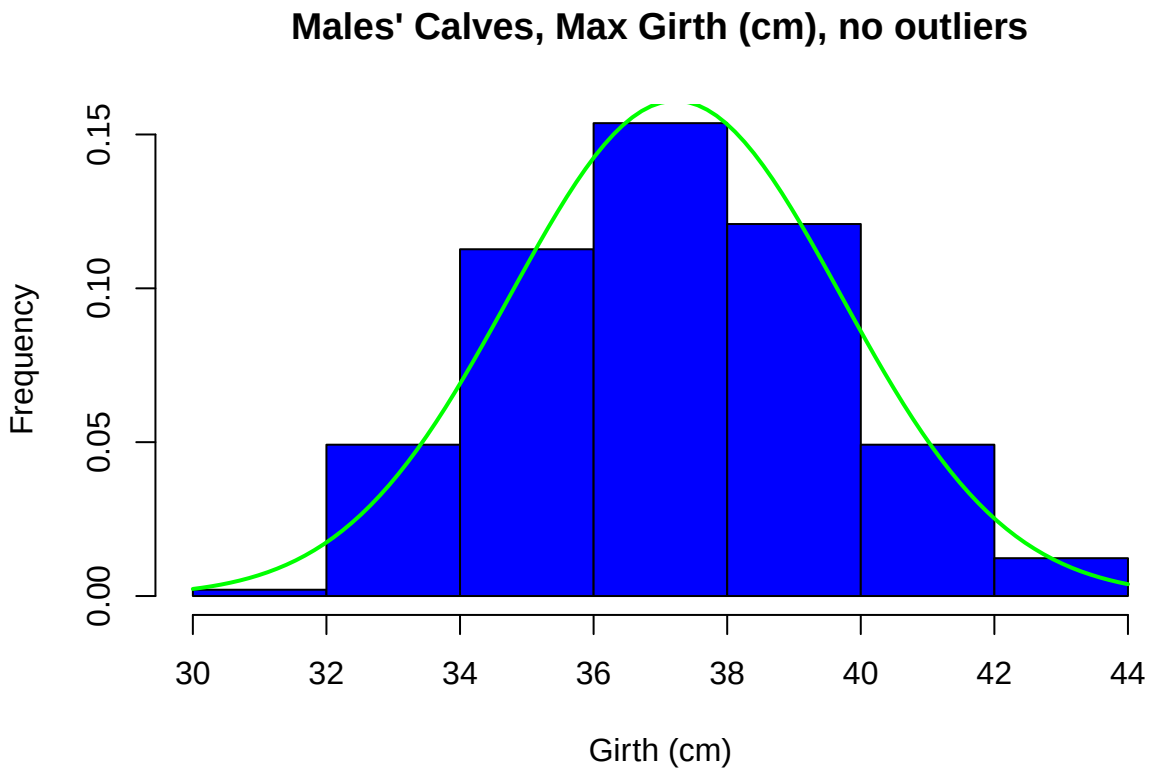
(2 points) Using the data, propose a point estimate for any parameters in your model.

```
mean_calves= mean(dimensions)
mean_calves
## [1] 37.22664
sd_calves = sd(dimensions)
sd_calves
## [1] 2.47863
```

The sample mean x is a point estimate of the population mean mu which equals mean_calves which equals 37.22664. The sample standard deviation s is a point estimate of the population standard deviation sigma which equals sd_calves which equals 2.47863.

(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.

```
x <- seq(30,44)

hist(dimensions,
     main = "Males' Calves, Max Girth (cm), no outliers",
     xlab = "Girth (cm)",
     ylab= "Frequency",
     col="blue",
     prob = TRUE)
curve(dnorm(x, mean_calves, sd_calves),
      add = TRUE,
      lwd = 2,
      col = "green")
```



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.

```
data = 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.

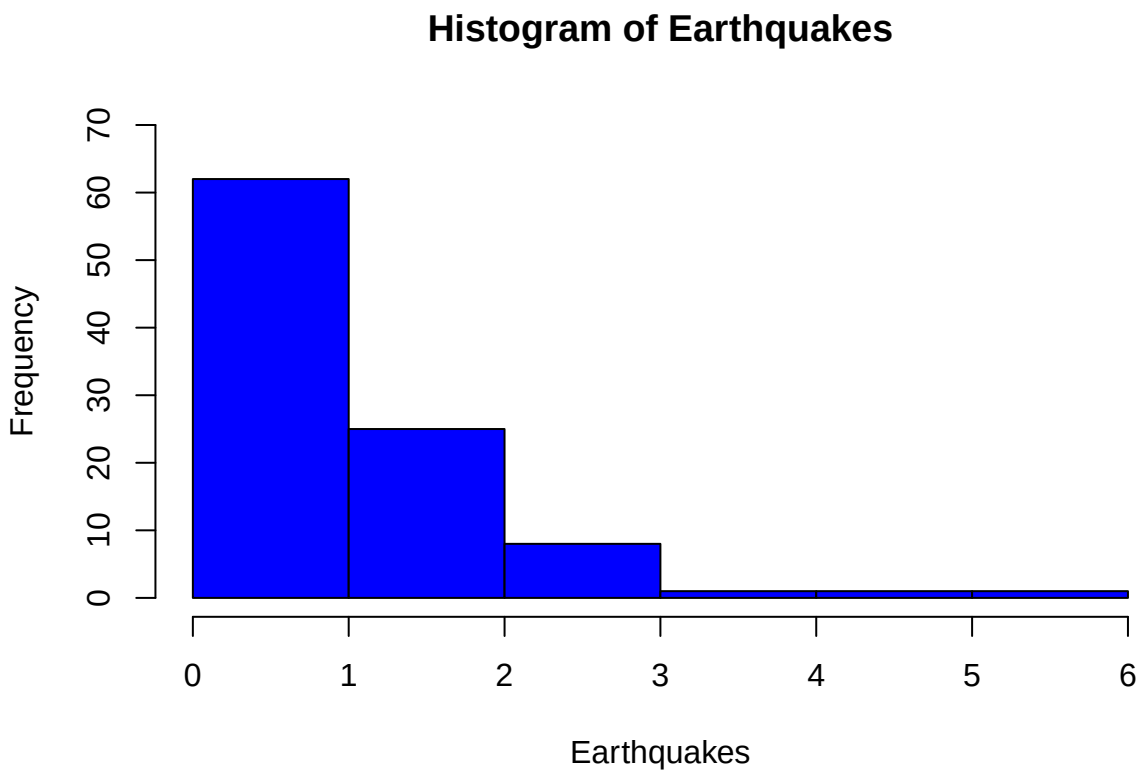
```
earthquakes = table(data$year)
earthquakes
##
## 1902 1903 1905 1906 1907 1908 1909 1912 1914 1915 1917 1918 1920 1923 1925 1927
##    2    2    1    4    1    1    1    1    1    1    1    1    1    3    1    2
## 1929 1930 1931 1933 1934 1935 1939 1940 1942 1943 1944 1945 1946 1948 1949 1950
##    1    2    3    3    1    2    2    1    1    2    2    2    3    2    2    1
## 1951 1953 1954 1956 1957 1960 1962 1963 1964 1966 1968 1969 1970 1971 1972 1974
##    1    1    1    1    2    2    1    1    1    3    1    1    3    2    2    2
## 1975 1976 1977 1978 1980 1981 1982 1983 1985 1986 1987 1988 1989 1990 1991 1992
##    2    6    1    1    2    2    1    1    1    1    1    2    1    2    2    5
## 1993 1994 1995 1997 1998 1999
##    1    2    2    2    3    3
```

(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.

```
for (i in 1902:1999){
  if (is.na(earthquakes[toString(i)])){
    earthquakes[toString(i)] = 0
  }
}

earthquakes
## 1902 1903 1905 1906 1907 1908 1909 1912 1914 1915 1917 1918 1920 1923 1925 1927
##    2    2    1    4    1    1    1    1    1    1    1    1    1    3    1    2
## 1929 1930 1931 1933 1934 1935 1939 1940 1942 1943 1944 1945 1946 1948 1949 1950
##    1    2    3    3    1    2    2    1    1    2    2    2    3    2    2    1
## 1951 1953 1954 1956 1957 1960 1962 1963 1964 1966 1968 1969 1970 1971 1972 1974
##    1    1    1    1    2    2    1    1    1    3    1    1    3    2    2    2
## 1975 1976 1977 1978 1980 1981 1982 1983 1985 1986 1987 1988 1989 1990 1991 1992
##    2    6    1    1    2    2    1    1    1    1    1    2    1    2    2    5
## 1993 1994 1995 1997 1998 1999 1904 1910 1911 1913 1916 1919 1921 1922 1924 1926
##    1    2    2    2    3    3    0    0    0    0    0    0    0    0    0    0
## 1928 1932 1936 1937 1938 1941 1947 1952 1955 1958 1959 1961 1965 1967 1973 1979
##    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
## 1984 1996
##    0    0
```

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

```
hist(earthquakes,
     main="Histogram of Earthquakes",
     xlab="Earthquakes",
     ylab="Frequency",
     ylim=c(0,70),
     col="blue")
```

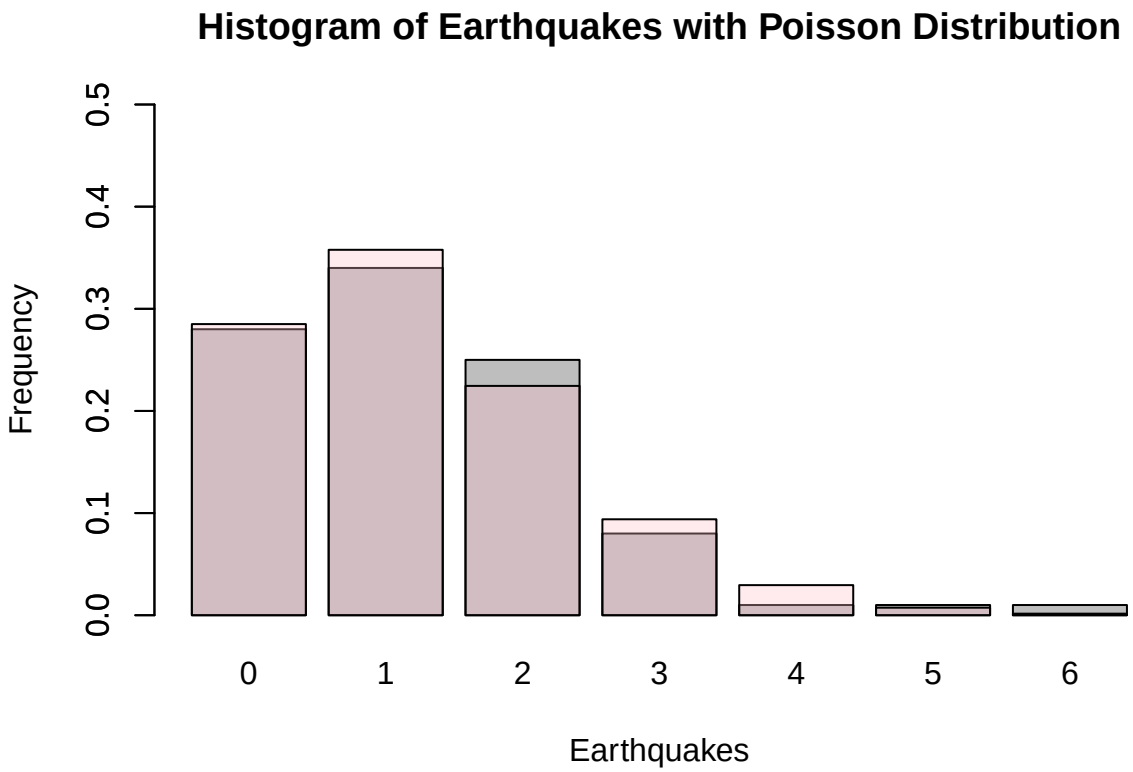


(6 points) Now, we want to construct a probabilistic model for these data. When it comes to counts of rare events, the **Poisson distribution** is oftentimes used. Before your proceed, review the Poisson distribution (Section 4.5 in the textbook and/or your *M362K Probability* notes). Then, familiarize 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.

```
table_zero = table(earthquakes)
table_zero
## earthquakes
##  0  1  2  3  4  5  6
## 28 34 25  8  1  1  1
lamda = mean(earthquakes)
lamda
## [1] 1.255102
```

(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.

```
c2 <- rgb(255,192,203, max = 255, alpha = 80, names = "lt.pink")

barplot(table_zero/100,
        main="Histogram of Earthquakes with Poisson Distribution",
        xlab="Earthquakes",
        ylab="Frequency",
        ylim = c(0,0.5))

poisson.dist <- dpois(0:6, lamda)
barplot(poisson.dist,
        add = TRUE,
        col = c2)
```



Problem #4 (14 points)

Go to Google historical data

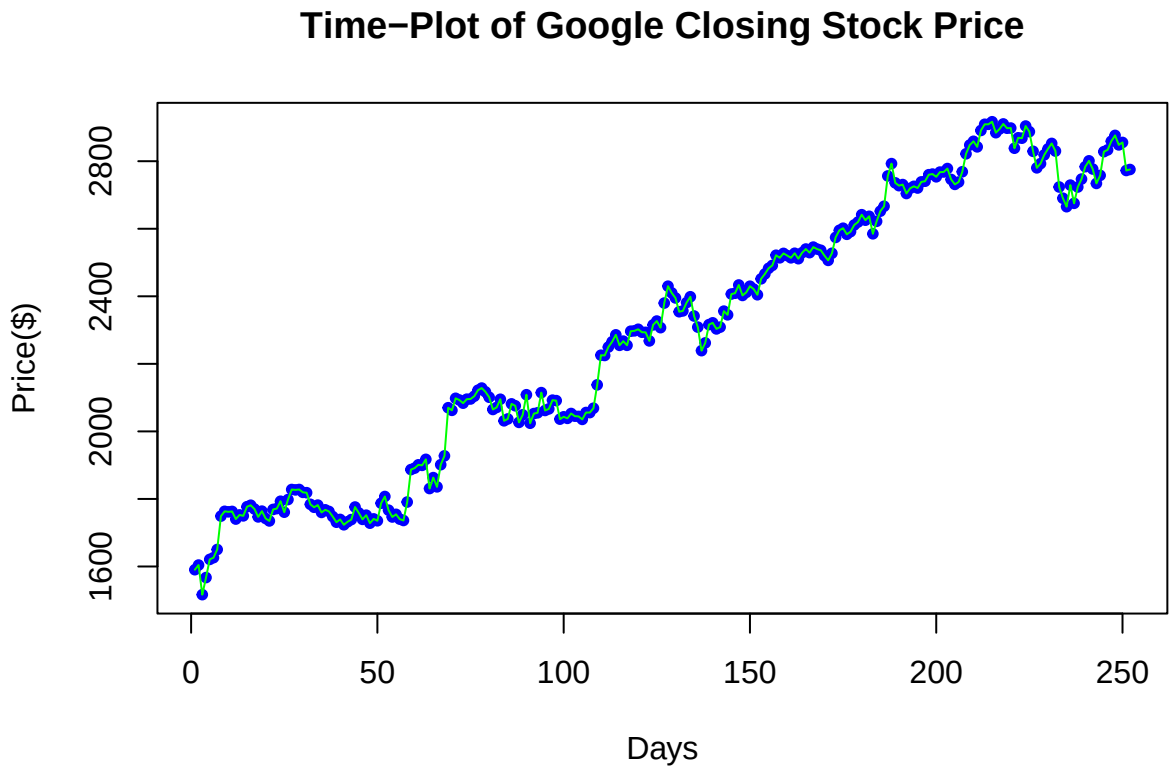
Then, download the data for the last 252 (or so) trading days, i.e., for the last year.

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

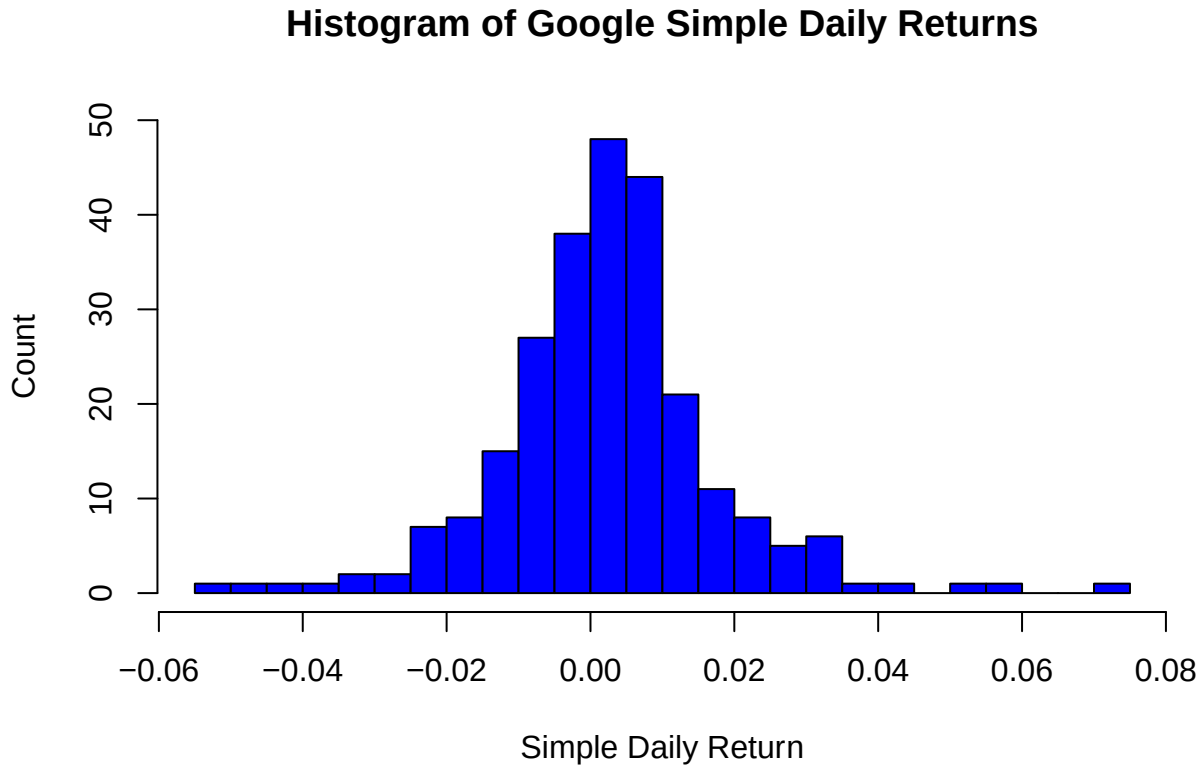
$$\frac{\text{price at end of day } t - \text{price at end of day } (t-1)}{\text{price at end of day } (t-1)}$$

```
closingprice = googdata$Close
plot(closingprice,
     main="Time-Plot of Google Closing Stock Price",
     xlab="Days",
     ylab="Price($)",
     pch=20,
     col="blue")
lines(closingprice,
      col="green")
```



(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?

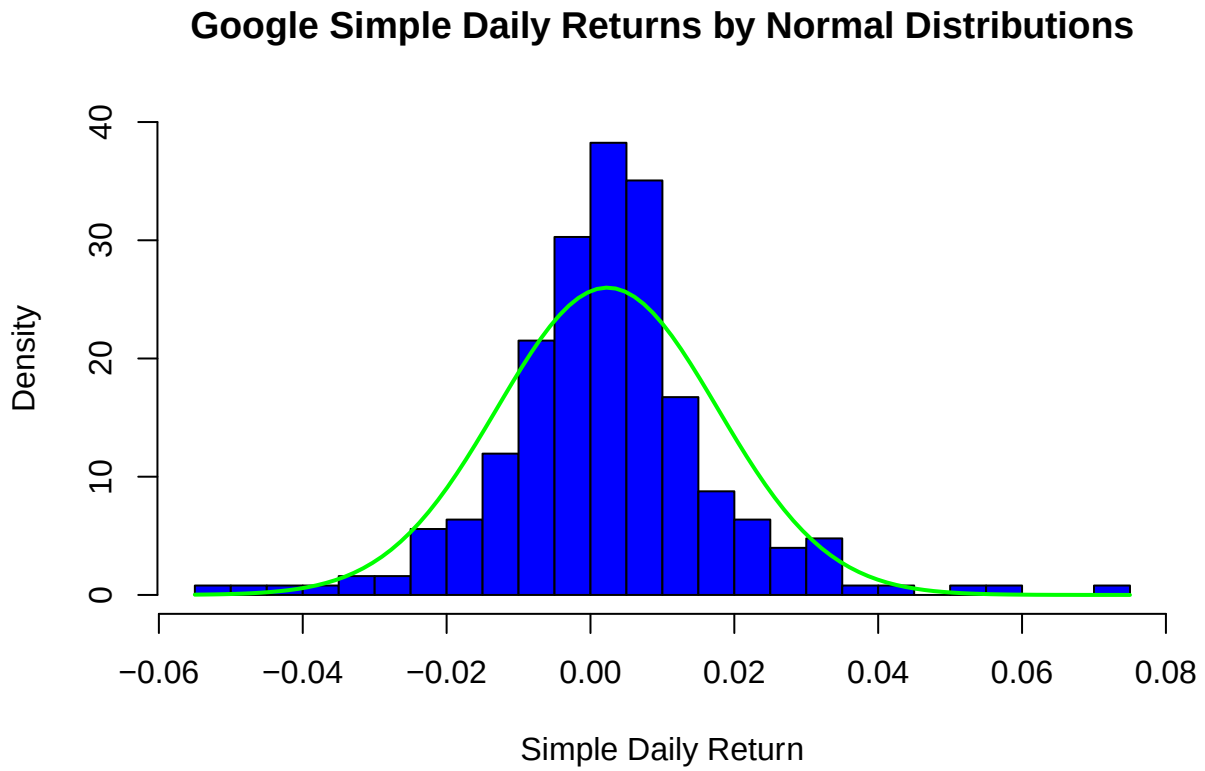
```
totaldailyreturns = c()
for (i in 2:252) {
  dailyreturns = (closingprice[i] - closingprice[i-1]) / closingprice[i-1]
  totaldailyreturns = append(totaldailyreturns, dailyreturns)
  }

hist(totaldailyreturns,
     breaks=20,
     main="Histogram of Google Simple Daily Returns",
     xlab="Simple Daily Return",
     ylab="Count",
     ylim=c(0,50),
     pch=20,
     col="blue")
```



The distribution is approximately normal. Based on the plot, there is a unimodal, bell curve with the peak at the center/ median/ mean.

(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.

```
hist(totaldailyreturns,
     breaks=20,
     main="Google Simple Daily Returns by Normal Distributions",
     xlab="Simple Daily Return",
     ylab="Density",
     ylim=c(0,40),
     col="blue",
     prob=TRUE)
curve(dnorm(x, mean(totaldailyreturns), sd(totaldailyreturns)),
      add=TRUE,
      col="green",
      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.

```
hist(totaldailyreturns,
     breaks=25,
     main="Google Simple Daily Returns by Laplace Distribution",
     xlab="Daily Returns",
     ylab="Density",
     ylim=c(0,50),
     col="blue",
     prob=TRUE)
curve(ddexp(x, mean(totaldailyreturns), sqrt((sd(totaldailyreturns)^2)/2)),
      add=TRUE,
      col="green",
      lwd=2)
```