Assignment 1. Statistical Theory and Modelling

AUTHOR
Caroline Birkehammar, Pablo Paras Ochoa, Steven Hiram Rubio Vasquez

Assignment 1

Import of the libraries and data

```
[1] "es_ES.UTF-8/es_ES.UTF-8/es_ES.UTF-8/C/es_ES.UTF-8/en_US.UTF-8"
```

Part 1

Problem 1a

```
dist <- rexp(10000, 1/2)

mean(dist)
```

```
[1] 2.013022
```

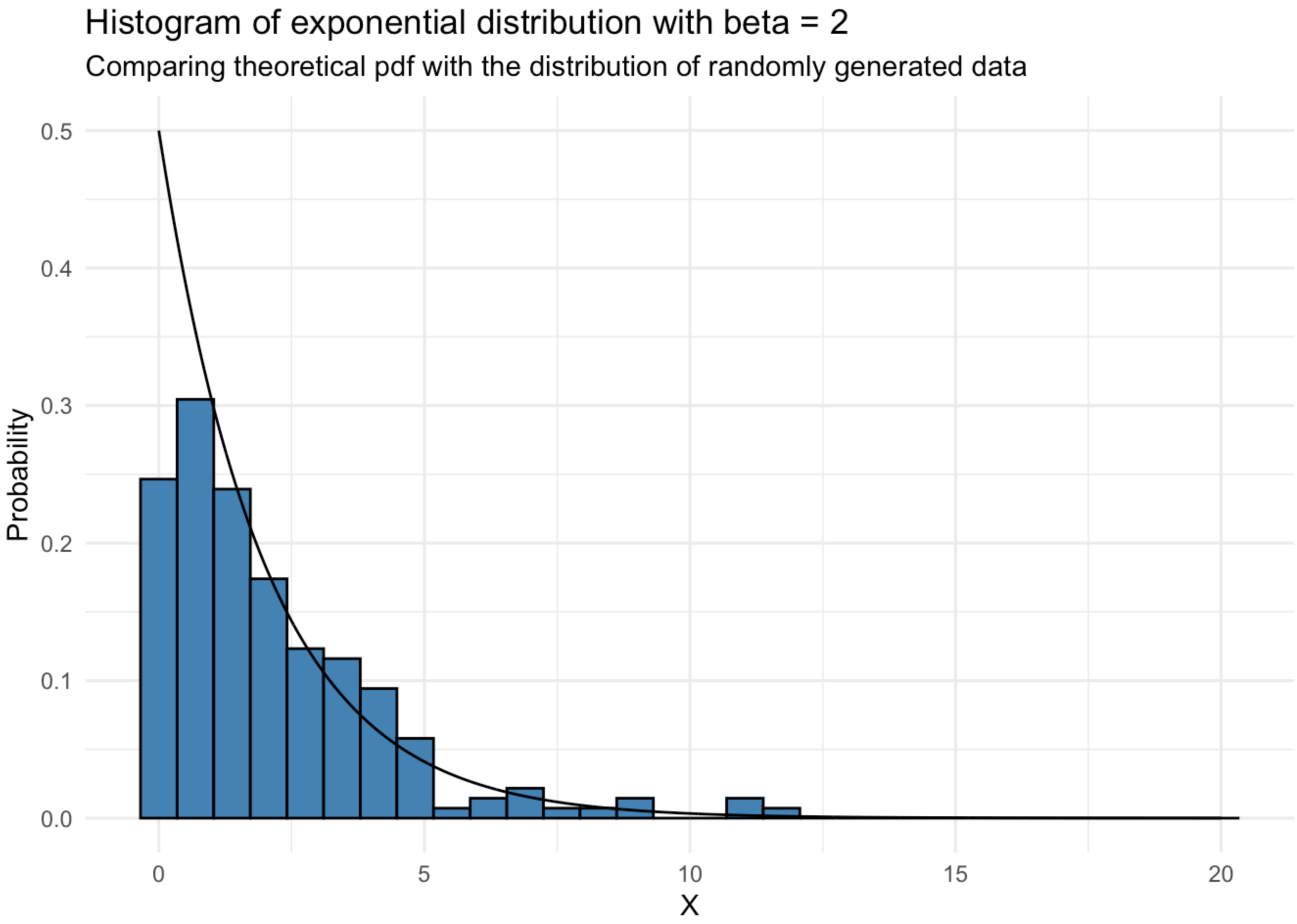
The true value for the mean given that $\lambda = 2$ or $\beta = 1/2$ is 2. When drawing a sample of 10,000 we obtain $\bar{x} = 2.013022$. Therefore, we conclude that R is using the correct rate of parameterization.

Problem 1b

```
dist <- tibble(rexp(200, 1/2)) %>% rename(X = 1)

dist_theo <- tibble(c(seq(from = 0, to = 20, by = .01)), dexp(c(seq(from = 0, to = 20, by

ggplot() +
  geom_histogram(aes(dist$X, after_stat(density)), bins = 30, fill = "steelblue", color =
  geom_line(aes(dist_theo$X, dist_theo$Y)) +
  theme_minimal() +
  labs(title = "Histogram of exponential distribution with beta = 2",
       subtitle = "Comparing theoretical pdf with the distribution of randomly generated
  xlab("X") +
  ylab("Probability")
```



Problem 1c

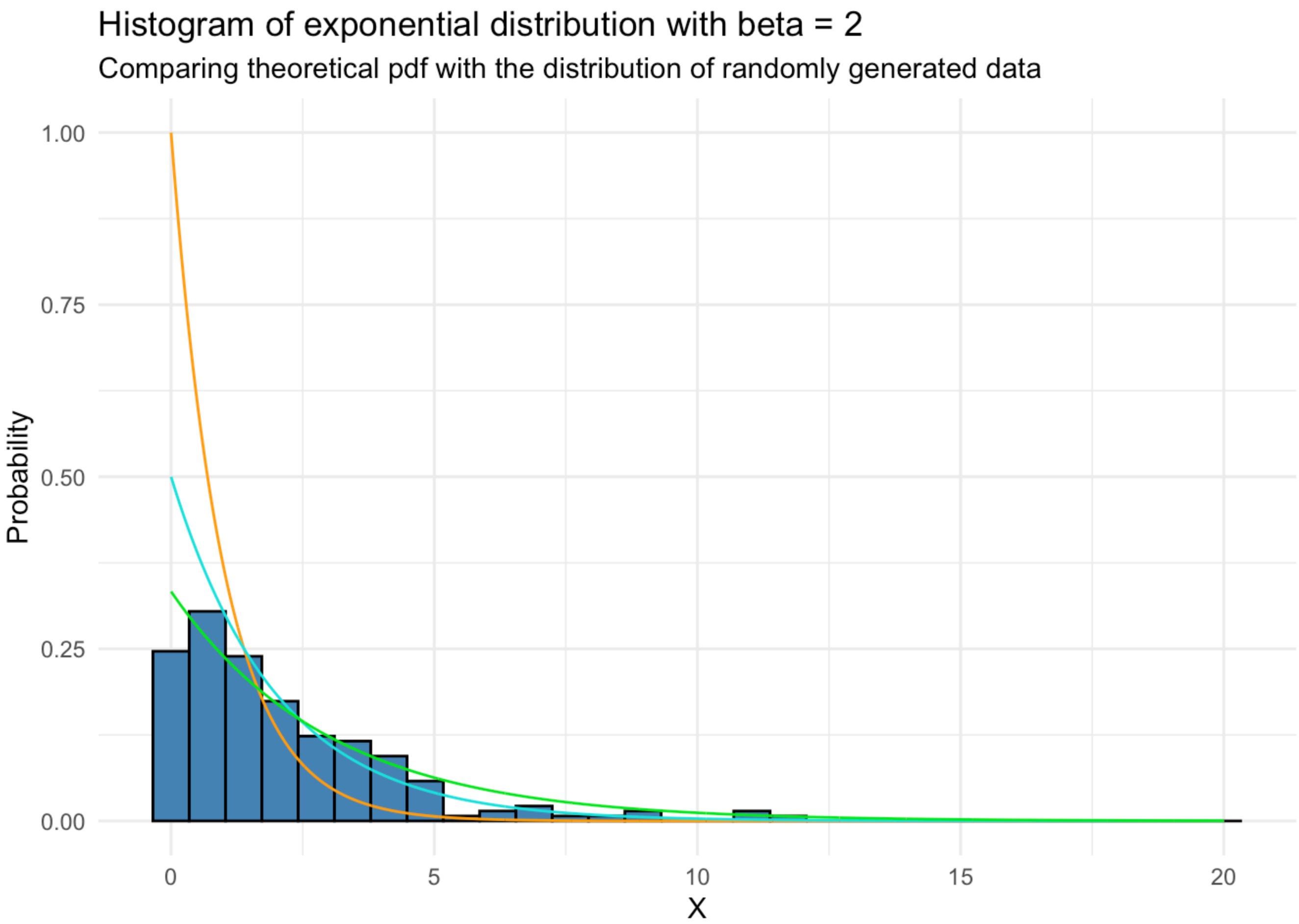
```
theo_data <- c(seq(from = 0, to = 20, by = .01))

dist_theo1 <- tibble(theo_data, dexp(theo_data, rate = 1/1)) %>% rename(X = 1, Y = 2)

dist_theo2 <- tibble(theo_data, dexp(theo_data, rate = 1/2)) %>% rename(X = 1, Y = 2)

dist_theo3 <- tibble(theo_data, dexp(theo_data, rate = 1/3)) %>% rename(X = 1, Y = 2)


ggplot() +
  geom_histogram(aes(dist$X, after_stat(density)), bins = 30, fill = "steelblue", color =
  geom_line(aes(dist_theo1$X, dist_theo1$Y), color = "#FF9E0D") +
  geom_line(aes(dist_theo2$X, dist_theo2$Y), color = "#1BE2DC") +
  geom_line(aes(dist_theo3$X, dist_theo3$Y), color = "#09E920") +
  theme_minimal() +
  labs(title = "Histogram of exponential distribution with beta = 2",
       subtitle = "Comparing theoretical pdf with the distribution of randomly generated
  xlab("X") +
  ylab("Probability")
```



Out of the three curves, it is clear that the orange one using $\beta = 1$ is the worse fit, as it underestimates the density of larger values and overestimates the density of smaller values. Out of the two other curves, using $\beta = 2$ for the blue one (which uses the same parameter value as when we generated a random sample) and $\beta = 3$ for the green one, it is not readily obvious which one is the better fit. The green pdf ($\beta = 3$) slightly overestimates large values and slightly underestimates small ones, but the blue pdf ($\beta = 2$) is also not a perfect fit.

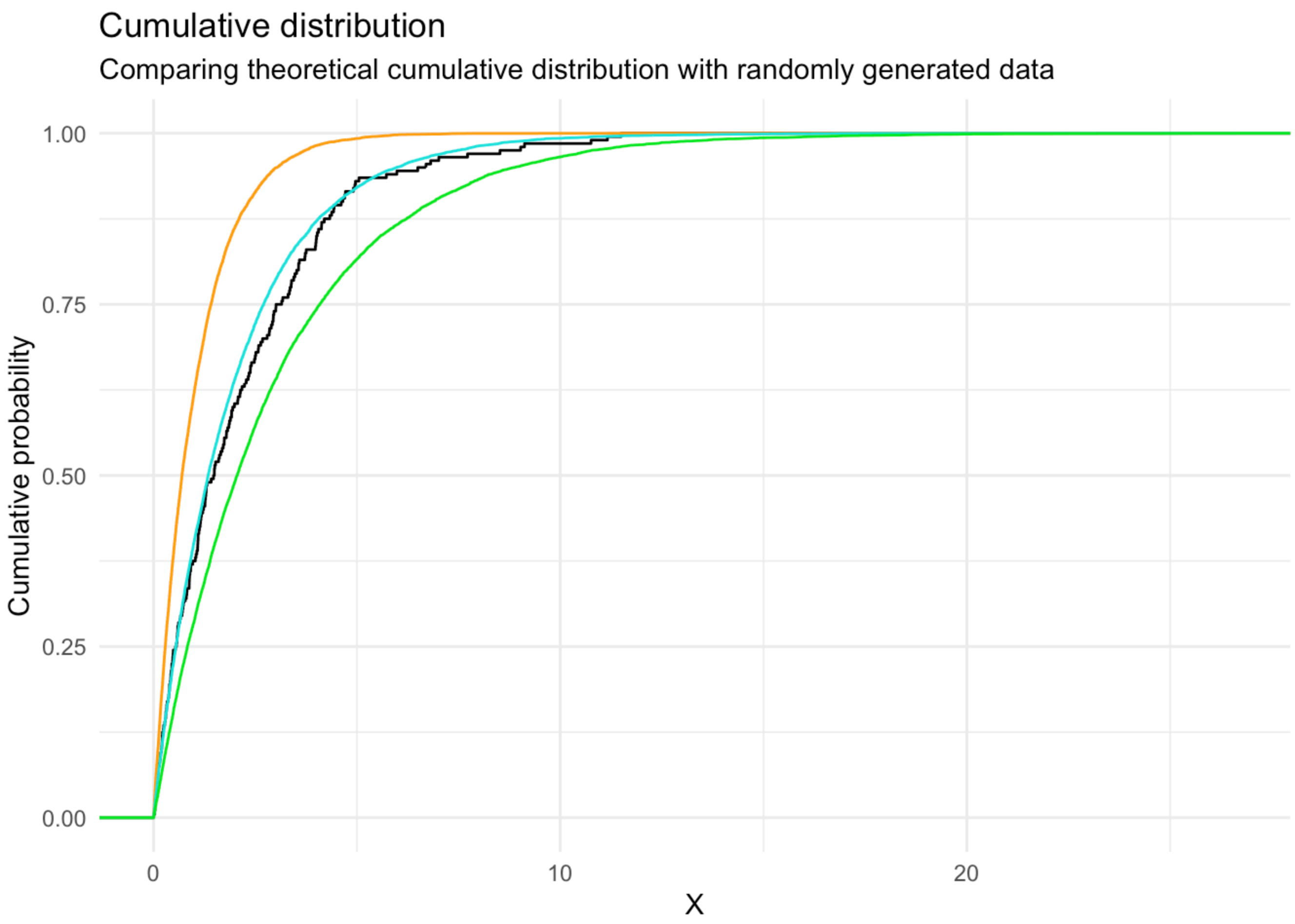
Problem 1d

```
dist_theo1 <- tibble(rexp(10000, 1/1)) %>% rename(X = 1)

dist_theo2 <- tibble(rexp(10000, 1/2)) %>% rename(X = 1)

dist_theo3 <- tibble(rexp(10000, 1/3)) %>% rename(X = 1)

ggplot() +
  stat_ecdf(aes(dist$X), color = "black") +
  stat_ecdf(aes(dist_theo1$X), color = "#FF9E0D") +
  stat_ecdf(aes(dist_theo2$X), color = "#1BE2DC") +
  stat_ecdf(aes(dist_theo3$X), color = "#09E920") +
  theme_minimal() +
  labs(title = "Cumulative distribution",
       subtitle = "Comparing theoretical cumulative distribution with randomly generated
  xlab("X") +
  ylab("Cumulative probability")
```


Cumulative distribution

Comparing theoretical cumulative distribution with randomly generated data

In this case it is more obvious which theoretical CDF is the better fit. The blue one ($\beta = 2$) is very closely aligned with the data while the orange ($\beta = 1$) and green ($\beta = 3$) are clearly different.

Problem 1e

```
obs_median   <- median(dist$X)
theo_median1 <- log(2)/1        # Theoretical median where lambda = 1
theo_median2 <- log(2)/(1/2)    # Theoretical median where lambda = 1/2
theo_median3 <- log(2)/(1/3)    # Theoretical median where lambda = 1/3

print(obs_median)
```

```
[1] 1.485096
```

```
print(theo_median1)
```

```
[1] 0.6931472
```

```
print(theo_median2)
```

```
[1] 1.386294
```

```
print(theo_median3)
```

```
[1] 2.079442
```

To obtain a sample median, one must simply order all observations by magnitude and select the value in the middle. Should there be an even number of observations, the two values in the middle can be averaged to obtain it.

For theoretical distributions we must do a little bit of math. In this case for the exponential distribution the median is defined as

$$median(x) = \frac{ln(2)}{\lambda}$$

and

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

Above, we calculated the theoretical medians for $\lambda = 1$, $\lambda = 1/2$ and $\lambda = 1/3$ by simply plugging in those values into the formula to obtain our results.

Problem 1f

```
delta <- c(5, 1, .5, .25, .1, .01, .001, .0001) # 5 and 1 also, just for fun :)

for (i in delta) {
  theo_data <- c(seq(from = 0, to = 20, by = i))
  dist_theo <- tibble(theo_data, dexp(theo_data, rate = 1/2)) %>% rename(X = 1, Y = 2)
  integral = sum(dist_theo$Y*i)
  print(paste("Density obtained using delta = ", i, ": ", round(integral, digits = 5), se
}
```

```
[1] "Density obtained using delta = 5: 2.72355"
[1] "Density obtained using delta = 1: 1.27071"
[1] "Density obtained using delta = 0.5: 1.13016"
[1] "Density obtained using delta = 0.25: 1.06376"
[1] "Density obtained using delta = 0.1: 1.02516"
[1] "Density obtained using delta = 0.01: 1.00246"
[1] "Density obtained using delta = 0.001: 1.0002"
[1] "Density obtained using delta = 0.0001: 0.99998"
```

Problem 1g

```
expon_fun <- function(x) {
  y = (1/2)*exp(-x/2)
  return(y)
}

integrate(f = expon_fun, lower = 0, upper = Inf)
```

```
1 with absolute error < 0.000034
```

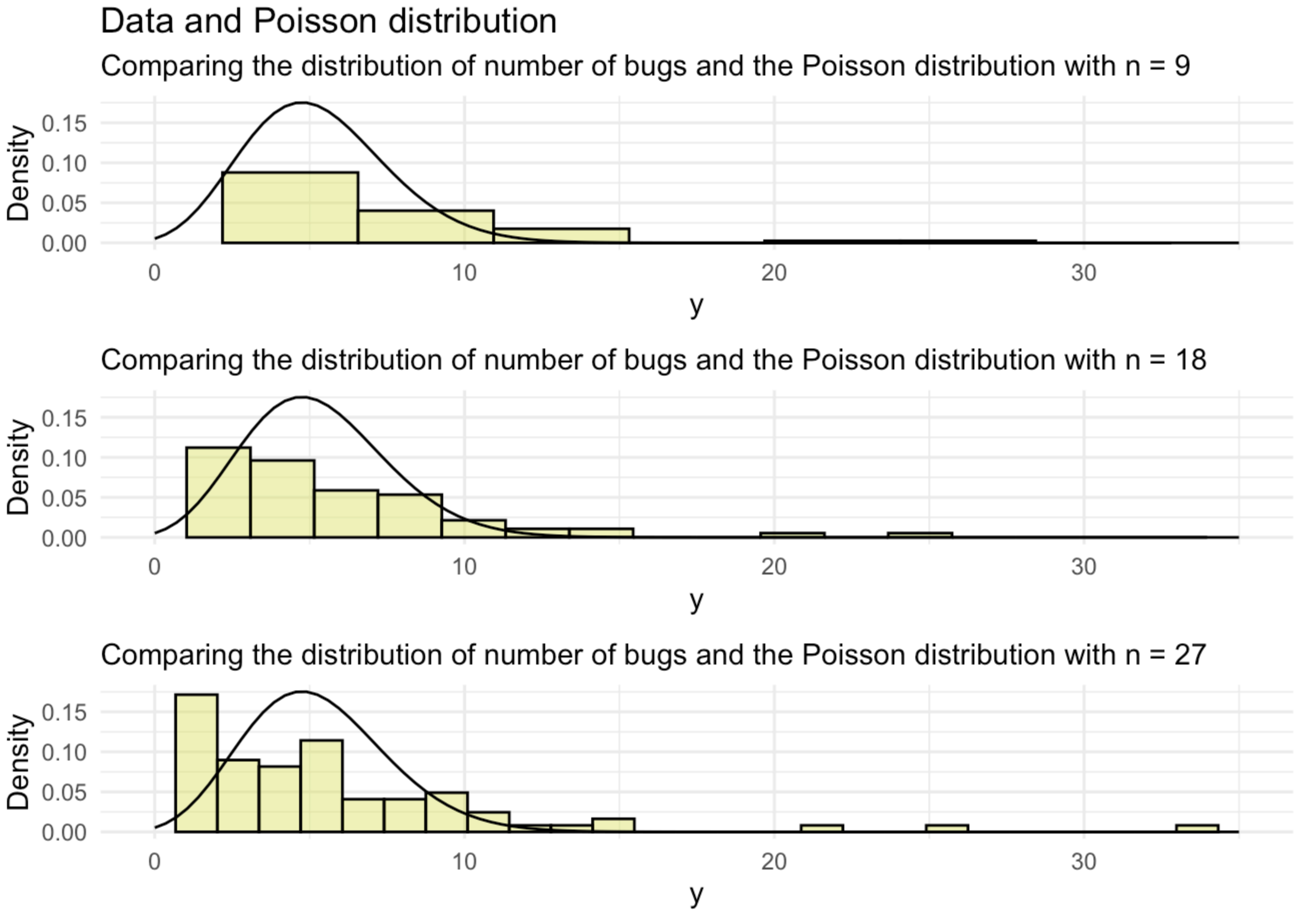
Part 2

Problem 2a

After reading the data set, we select the bugs column and calculate the expected value (λ) and fit a Poisson regression to the data. As for the number of bins, the rule ***sqrt****(number of observations)* was used as a starting point, but we tried different bin sizes.

```
y <- data$nBugs
lambda <- mean(y)
lambda
```

```
[1] 5.252747
```


Data and Poisson distribution

Comparing the distribution of number of bugs and the Poisson distribution with n = 9

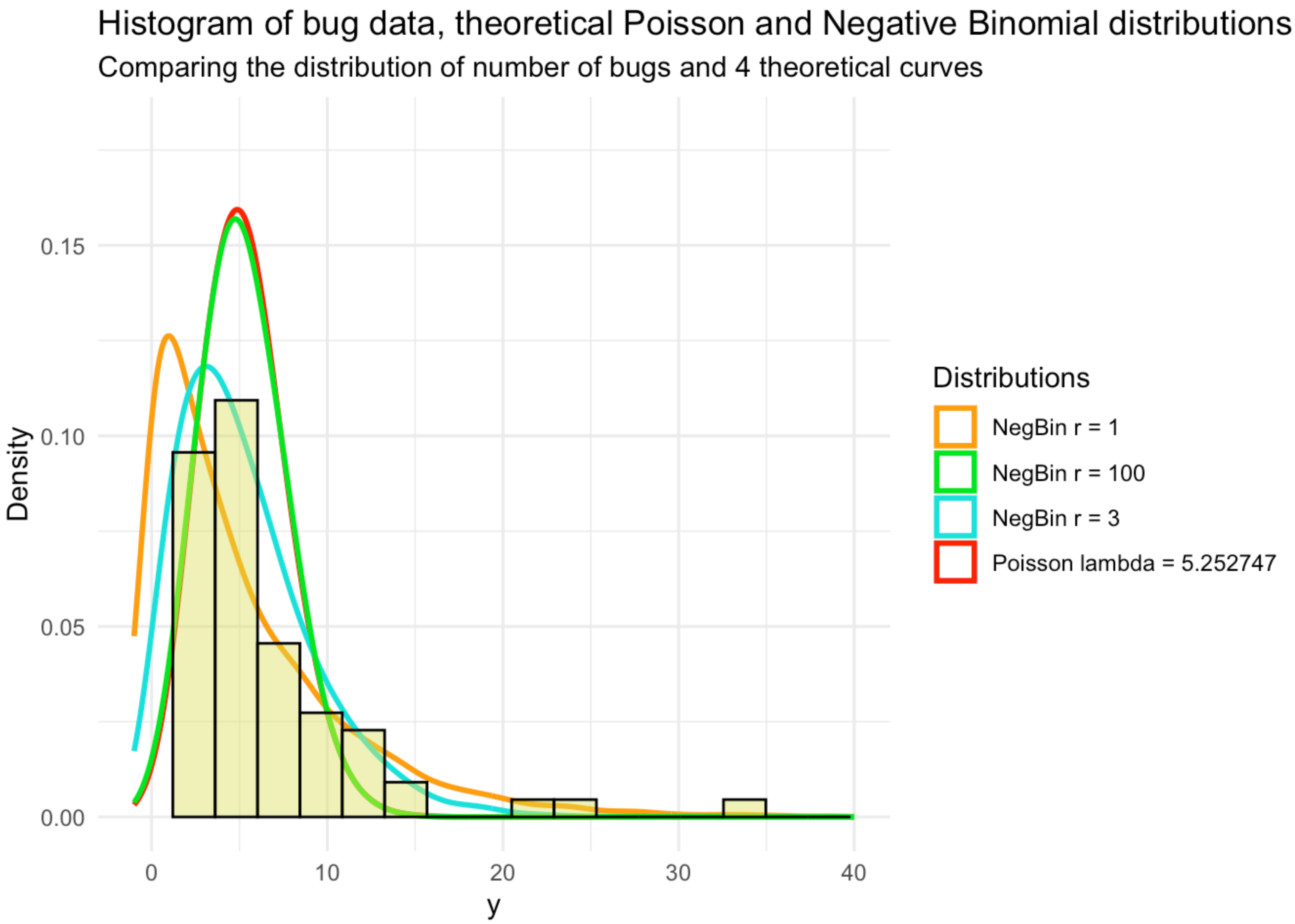
Comparing the distribution of number of bugs and the Poisson distribution with n = 18

Comparing the distribution of number of bugs and the Poisson distribution with n = 27

The distribution does not seem to follow the Poisson distribution, based on how the histograms look compared to the theoretical probability distribution.

Problem 2b

We fit a Negative Binomial distribution to the data and compare it with the previous result.

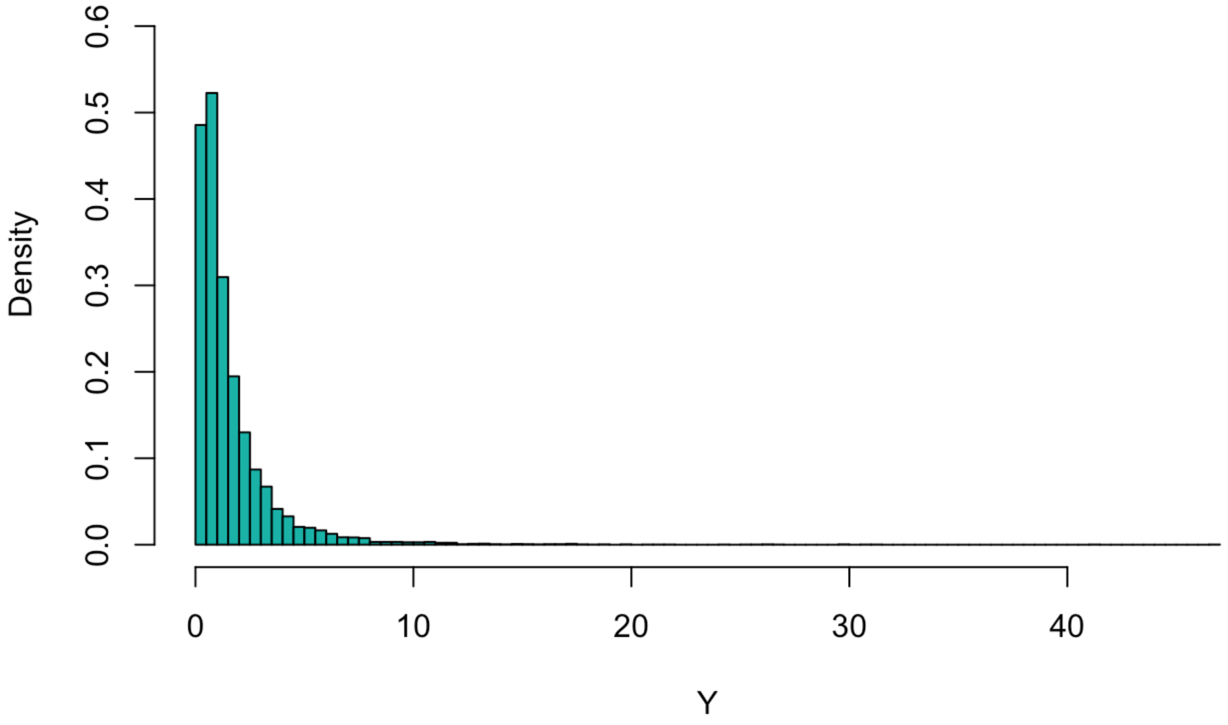


We observe that the curve that follows the distribution of the data best is the Negative Binomial distribution with r = 3 (the blue pdf). The curve that follows a Poisson distribution best is the Negative Binomial distribution with r = 100 (the green pdf). This is because the Negative Binomial distribution converges in distribution to a Poisson distribution as r increases.

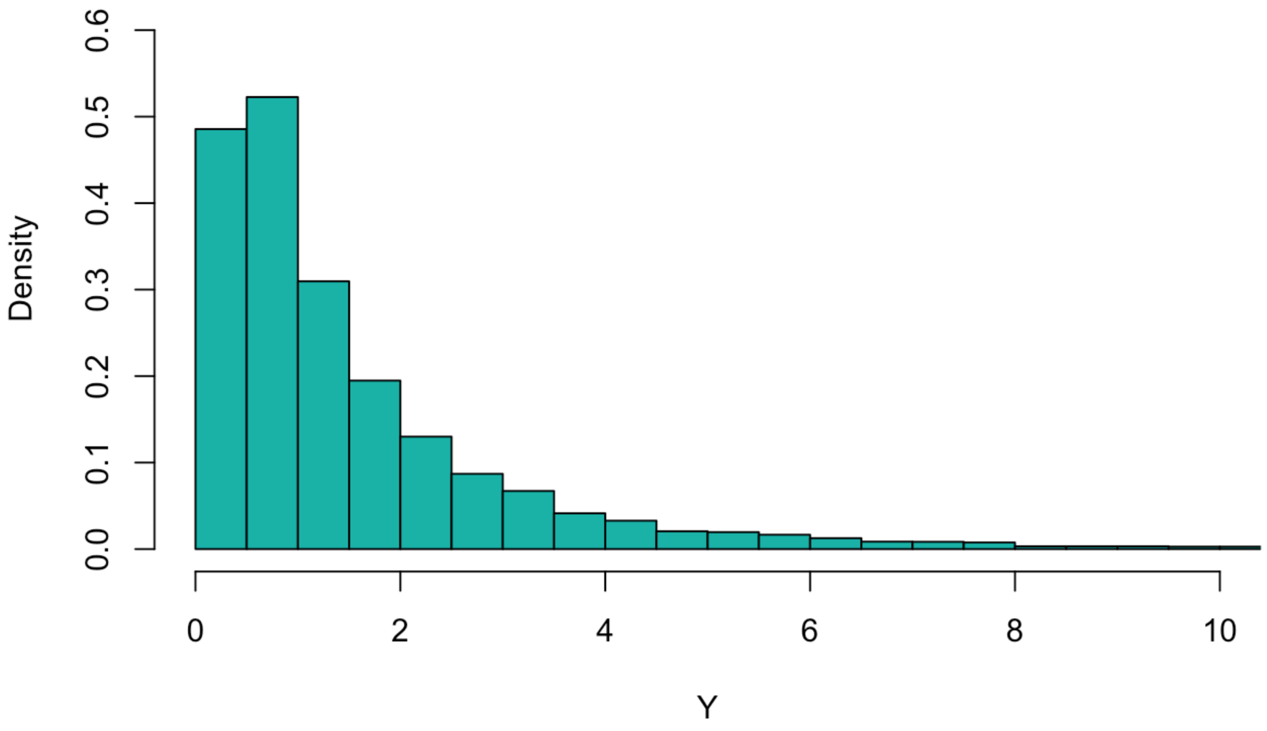
Part 3

Problem 3a

Distribution of Y in simulated data



Distribution of Y in simulated data, zoomed in



Problem 3b

We know that for any function $X = f(x)$ and $Y = g(X)$, where $g(X)$ is an invertible, differentiable and monotonically increasing or decreasing variable, the probability density function of Y is given by

$$f_Y(y) = f_X\left(g^{-1}(y)\right)\left|\frac{d}{dy}g^{-1}(y)\right|,$$

where $g^{-1}(y)$ is the inverse function of Y.

In our case where $X \sim \mathrm{Normal}(\mu = 0, \sigma^2 = 1)$, the probability density function of X is given by

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right).$$

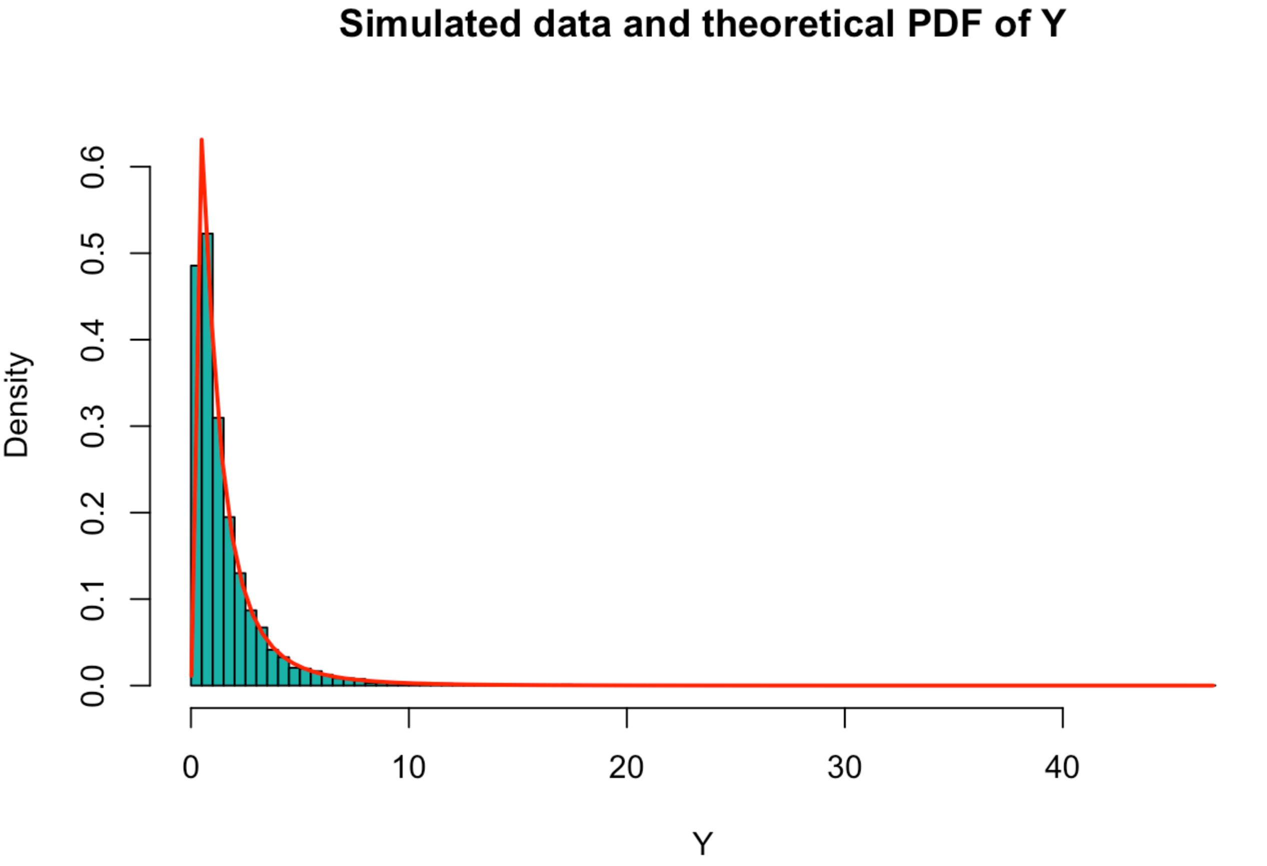
Y is a function of X, $Y = \exp(X)$, and has the inverse function $X = log(Y)$. The derivative of the inverse function is given by

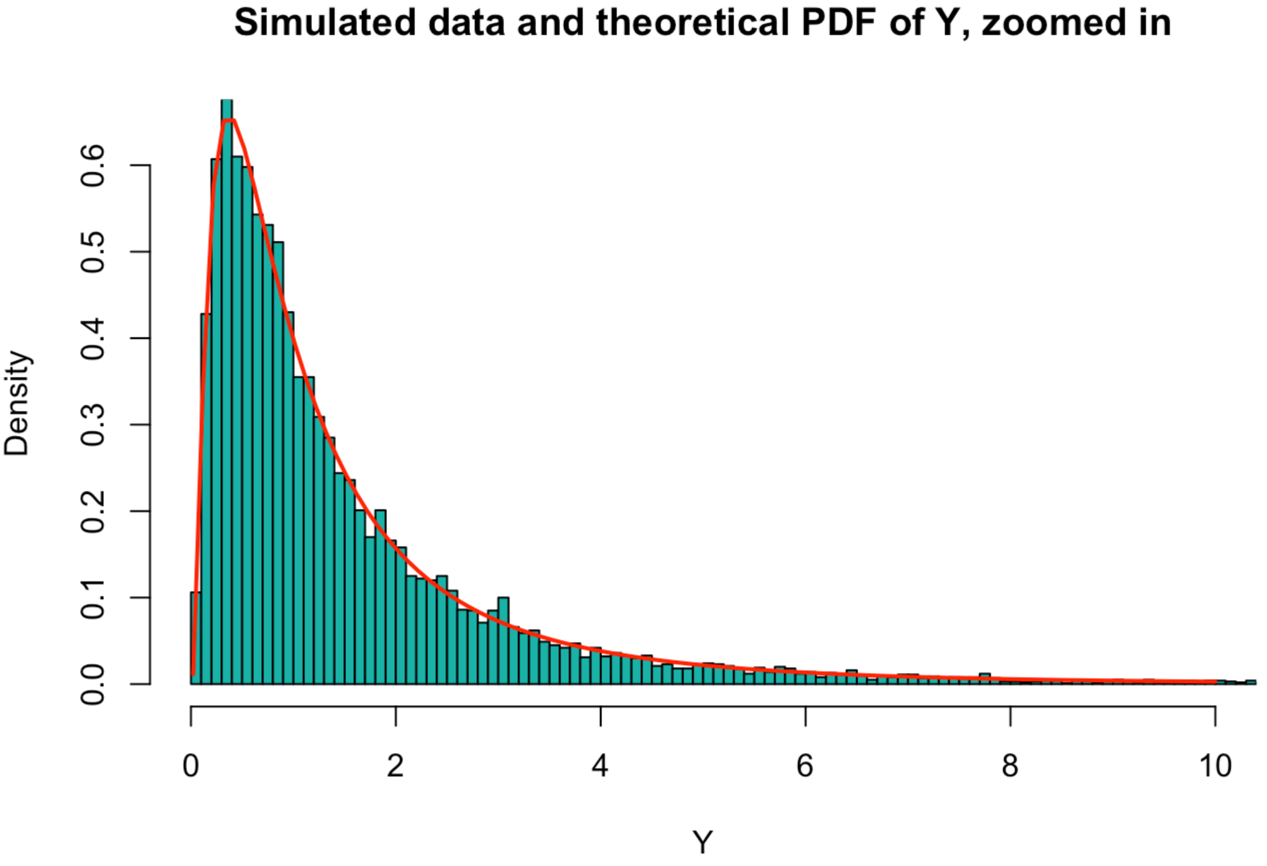
$$\frac{d}{dy}g^{-1}(y) = \frac{d}{dy}\log(y) = \frac{1}{y}.$$

Replacing into the equation for the PDF of Y gives:

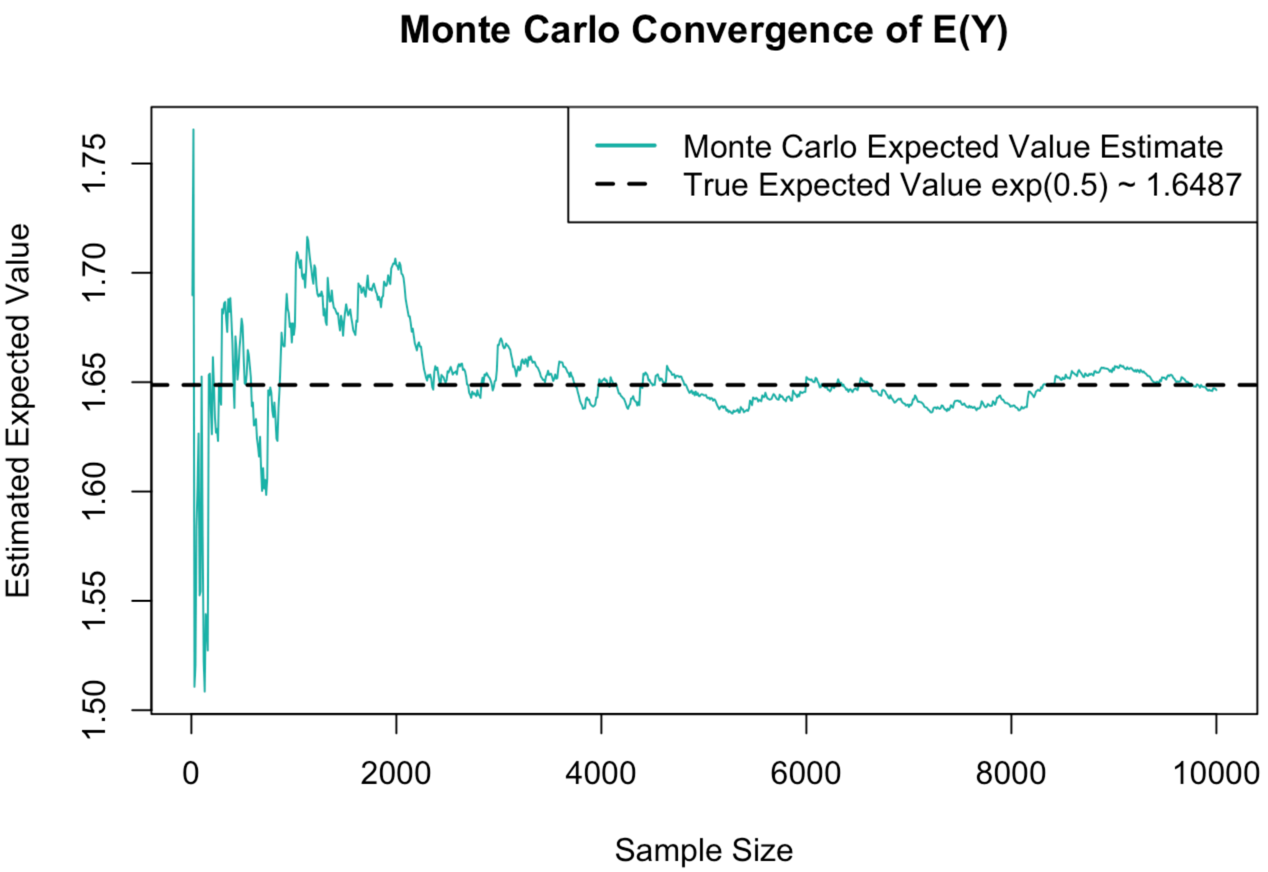
$$f_Y(y) = f_X\left(g^{-1}(y)\right)\left|\frac{d}{dy}g^{-1}(y)\right| = f_X(\log(y)) \cdot \frac{1}{y} = \frac{1}{y\sqrt{2\pi}}\exp\left(-\frac{\log(y)^2}{2}\right)$$

Below are two versions of the histograms from problem 3a with overlaid theoretical probability distributions; the first one is in full scale and with 100 bins, and the second one is zoomed in and has more bins to better show the distribution of the simulated data.


Simulated data and theoretical PDF of Y
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
10
20
30
40
Y


Simulated data and theoretical PDF of Y, zoomed in
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
2
4
6
8
10
Y

Problem 3c



Monte Carlo Simulation relies on repeated random sampling to obtain some estimated value, in this case the expected value of $Y = \exp(X)$ where $X \sim \mathrm{Normal}(\mu = 0, \sigma^2 = 1)$. The plot above shows how the average mean value of Y converges to the true mean (which is approximately 1.6587) as sample size increases. There are only 1,000 data points in the plot since each data point includes 10 observations and we have generated random data containing 10,000. After about the 200th data point (which averages 2,000 observations) the Monte Carlo estimate of the expected value stabilizes and converges around the true mean.