Course Project for Statistical Inference Part 1 - Evaluating the Coverage of an exponential distribution

The objective is to address the following: The exponential distribution can be simulated in R with rexp(nosim, lambda) where lambda is the rate parameter. The mean of exponential distribution is 1/lambda and the standard deviation is also 1/lambda. Set lambda = 0.2 for all of the simulations. In this simulation, you will investigate the distribution of averages of 40 exponential(0.2)s.

Illustrate via simulation and associated explanatory text the properties of the distribution of the mean of 40 exponential(0.2)s. You should

1. Show where the distribution is centered at and compare it to the theoretical center of the distribution.
2. Show how variable it is and compare it to the theoretical variance of the distribution.
3. Show that the distribution is approximately normal.
4. Evaluate the coverage of the confidence interval for 1/lambda: $\bar{X} \pm 1.96Sn$.

1. Setting the environment

In this step, we will setup the environment that is needed in R. The code that loads all the necessary libraries in R are shown in appendix A.

2. Data Preparation

First, We will simulate the data required to perform the exercise. The code that is used for generating the simulated data and storing them in R is showed in appendix B.

With a Lambda of 0.2 the population mean and standard deviation would both be 5. The sample data is expected to have the same mean, and, for a sample size of 40, the sample SD is 0.79. The simulated data has a mean of 5.01 and SD of 0.79; which are close to the expected values. In addition, the Median for the sample is 4.96, again closely matching the expected value, and supporting a non-skewed data.

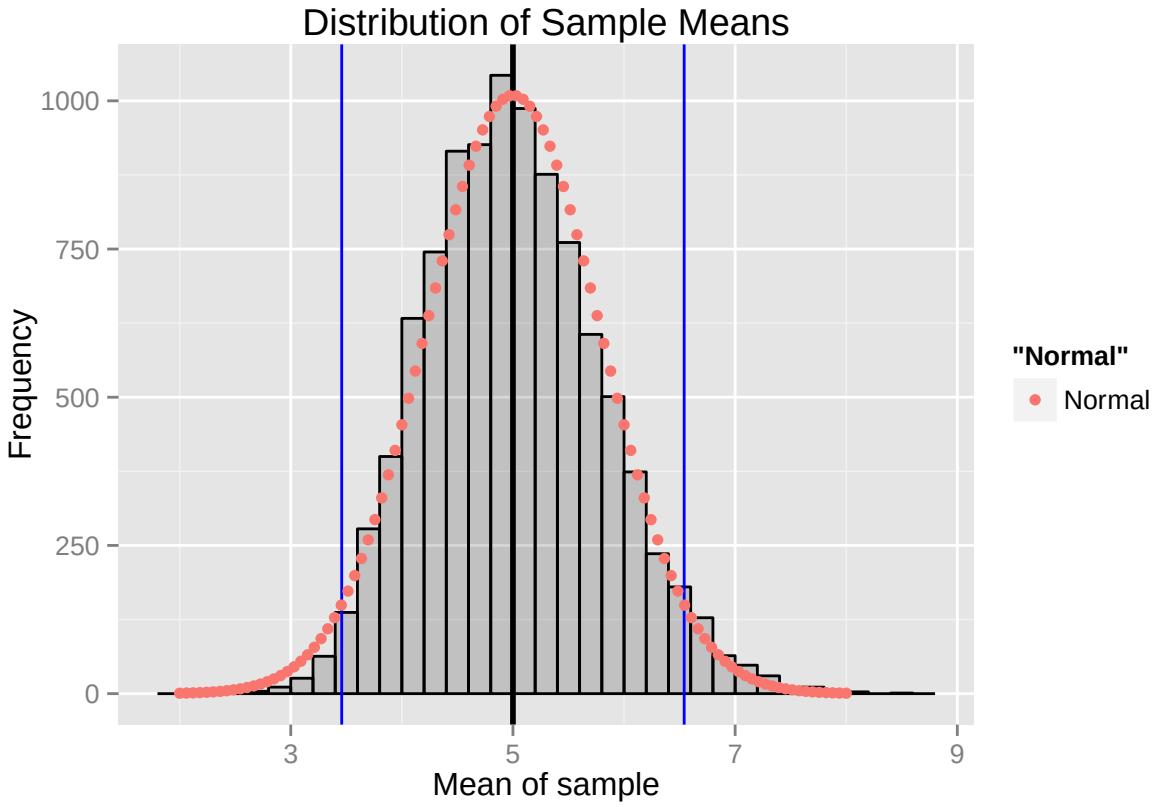
3. Data processing

The next step is to prepre the data for a normal curve with the mean and the standard deviation of the simulated data.Thr R code which performs this is shown in appendix C

The following plot displays a histogram of the simulated sample data, with an overlay for

1. A plot of a normal distribution for the mean 5 and expected sample SD 0.79
2. A Black Vertical line for the population mean, at x = 5
3. Blue vertical lines for the two SD distances away from the mean; indicating the 95% coverage area, at x = 6.54 & x = 3.46

This demonstrates that the simulated data gives a good approximation to the normal curve. The R code for producing the plot is shown in appendix D



4. Results

With this level of sample size, we can assess how well the dataset matches a 95% confidence level to the population. Taking the sample mean, the standard error and the SD value for 95% confidence, we can then determine the number of times this result includes the population mean. This will be expected to be approximately 95% of the time. Also we have to note that by increasing the sample size, The coverage should improve. Whereas reducing the sample size is likely to lead to a lesser coverage. The R code for finding the coverage is shown in appendix E

From the sample data, the proportion of sample means that are within a 95% interval of the population mean is 95.44%

5. Appendices

5.1 Appendix A

```
library(ggplot2)
```

5.2 Appendix B

```
lLambda <- 0.2
nSize <- 40 # the size of the sample
noSim <- 10000 # the number of simulations

# The mean of the rexp is 1/lambda and so is standard deviation.
# Lambda is 0.2

lMean <- 1 / lLambda
lSd <- 1 / lLambda

## The creator of the 1000 means of 40 values

simData <- replicate(noSim, mean(rexp(nSize, lLambda)))
simDataMean <- mean(simData)
simDataMed <- median(simData)
simDataSd <- sd(simData)

# Sample variance/ sd is population variance /n

simMeanSd <- lSd / sqrt(nSize)
```

5.3 Appendix C

```
sampleData <- seq(2, 8, length = 100)
sampleNormData <- dnorm(sampleData, mean = lMean, sd = simMeanSd)

# scaling the data

sampleNormData <- noSim * sampleNormData * .2
sampleDataFrame <- data.frame(cbind(sampleData, sampleNormData))
```

5.4 Appendix D

```
simData2 <- data.frame(simData)
simDataHist <- ggplot(simData2, aes(x = simData)) +
                      geom_histogram(alpha = .20, binwidth = .2,
                                     colour = "black")
simDataHist <- simDataHist + xlab("Mean of sample") + ylab("Frequency") +
                             ggtitle("Distribution of Sample Means")
```

```
simDataHist <- simDataHist + geom_vline(xintercept = lMean, size = 1)
simDataHist <- simDataHist + geom_vline(xintercept = lMean + 1.96 * simDataSd,
                                        size = 0.5, color = "Blue")
simDataHist <- simDataHist + geom_vline(xintercept = lMean - 1.96 * simDataSd,
                                        size = 0.5, color = "Blue")
simDataHist <- simDataHist + geom_point(data = sampleDataFrame,
                    aes(x = sampleData, y = sampleNormData,
                        colour = 'Normal'))
simDataHist
```

5.5 Appendix E

```
# simData is the set of means, With the absolute difference of the means,
# we simply subtract the Zvalue times the stderror

coverageData <- abs(simData - lMean) - 1.96 * simMeanSd

# Fin the percentage of time the simulated mean is in the range from 0 to 95

coveragePerc <- 100 * length(coverageData[coverageData <= 0]) / noSim
```