Statistical Inference Course Project: Part 1

Technophobe01

2015-04-25

1 Introduction

The purpose of this document is to develop and answer the Coursera Peer Assessment Part 1 of the Coursera Statistical Inference Course Project. Our goal is to investigate the exponential distribution in R and compare it with the Central Limit Theorem. This reports source code is available via **GitHub**, if you wish to reveiw the markdown formatting in more detail.

The exponential distribution can be simulated in R with rexp(n, lambda) where lambda is the rate parameter. The mean of exponential distribution is 1/lambda and the standard deviation is also 1/lambda. For this project we set lambda = 0.2 for all of the simulations. We investigate the distribution of averages of 40 exponentials, across a thousand simulations.

2 Overview:

We illustrate via simulation and associated explanatory text the properties of the distribution of the mean of 40 exponentials. Specifically, we:

1. Show the sample mean and compare it to the theoretical mean of the distribution.
2. Show how variable the sample is (via variance) and compare it to the theoretical variance of the distribution.
3. Show that the distribution is approximately normal.

In point 3, we focus on the difference between the distribution of a large collection of random exponentials and the distribution of a large collection of averages of 40 exponentials.

2.1 Simulations

2.1.1 Questions 1. Answer: Show the sample mean and compare it to the theoretical mean of the distribution.

Theoretically, we expect to see our simulation means centered on $1/\lambda$, with a variance of $(1/\lambda)^2$ / 40. If we compare the simulation population mean to theoretical population mean we see that they are close.

- **Simulation Mean:** 4.969789
- **Theoretical Mean:** 5

In conclusion, for question 1 we can see from the above numbers that the average mean of **4.969789** is very near to the theoretical mean of **5**.

2.2 Sample Variance versus Theoretical Variance

A measure of variability is perhaps the most important quantity in statistical analysis. The greater the variability in the data, the greater will be our uncertainty in the values of parameters estimated from the data, and the less will be our ability to distinguish between competing hypotheses about the data. Thus, our goal is to compare the sample variance with the theoretical variance.

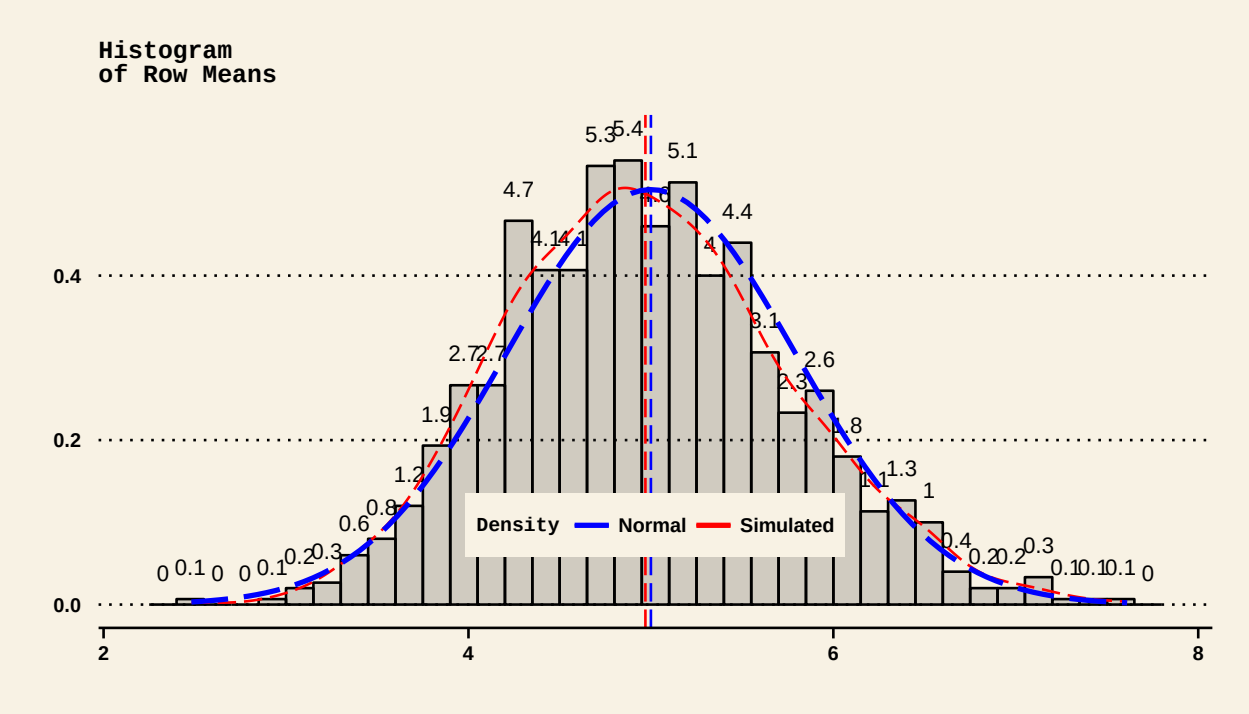

Figure 1: Histogram of Row Means and Normal Distribution comparison

2.2.1 *Answer*: Question 2. Show how variable the sample is (via variance) and compare it to the theoretical variance of the distribution.

The theoretical value for the variance of the distribution of averages is given by the variance of the original population σ^2 divided by the number of samples 40 used to compute the averages : $var(\bar{X}) = \frac{\sigma^2}{n} = \frac{1}{40\lambda^2} = 0.625$

The standard deviation of the distribution is **0.7764661** with the theoretical SD calculated as **0.7905694**. The Theoretical variance is calculated as $var(\bar{X}) = \frac{\sigma^2}{n} = \frac{1}{40\lambda^2} =$ **0.625**. The actual variance of the distribution is **0.6028996**.

Thus, we can see that both the standard deviation of the exponential distribution and the theoretical variance of the distribution are very close. ***Their is little or no variance in the data.***

2.3 Distribution

Our goal here is to show that the distribution of the simulation is approximately normal. In order to do this we want to compare the distribution of averages to the Central Limit Theorem (CLT). To do this we plot the distribution as a histogram along with a normal distribution in Figure 1.

2.3.1 *Answer*: Question 3. Show that the distribution is approximately normal.

We can see from Figure 1 that the depicted histogram shows that the simulated sample approximates the normal distribution, as predicted by the central limit theorem. The sample mean is shown as a **RED** dotted line, and the theoretical mean is the shown as a **BLUE**) dotted line. ***Thus, we can see that the simulated distribution approximates the normal distribution.***

It is worth noting that whilst histogram and density plots are useful to visualize such distributions due to the depicted bell-shape appearance. It is often easier to judge if we can get the distribution to lie in a straight line. To do that you can use quantile-quantile plots (QQ plots). In the *R Language* we have several commands available relating to QQ plots; the first of these is `qqnorm()`, which takes a vector of numeric values and plots them against a set of theoretical quantiles from a normal distribution. The upshot is that you produce a series of points that appear in a perfectly straight line if your original data are normally distributed.

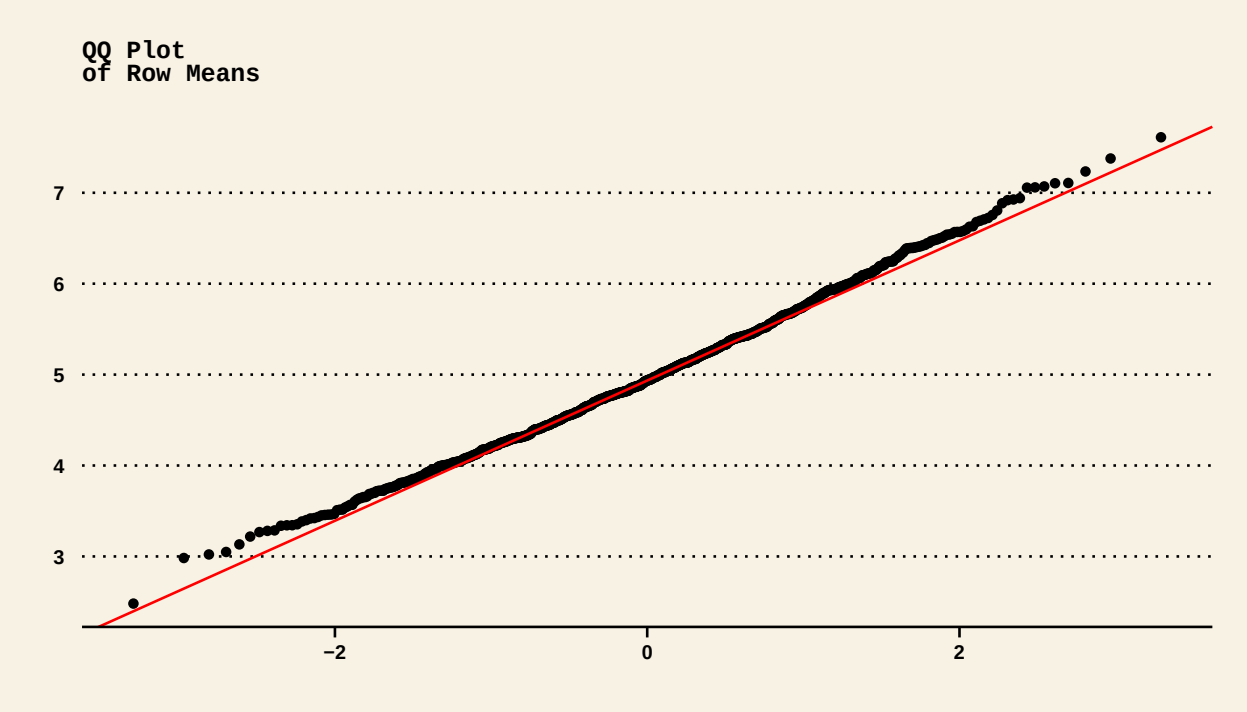

Figure 2: QQ Plot of Row Means

The above ggplot2 QQ plot shows a distribution of averages of **40** exponential random variables. The straight **RED** line is the standard normal distribution reference. ***Thus, we can deduce the population is normally distributed (because the data is distributed along the line)***.

3 Conclusions

In conclusion we can make the following observations:

1. We can deduce from the data presented that the distribution of the sample means will be normal in shape, regardless of the shape of the parent population, provided the sample size is large enough. A sample size of n = **40** is a large enough distribution of sample means to create a normal distribution shape, even though the samples were drawn from an exponential distribution.
2. The mean of the exponential distribution of sample means is identical to the mean of the parent population, the population from which the samples are drawn.
3. The higher number of the simulation conducted, the "narrower" will is the spread of the distribution of sample means.

4 Appendix A: Code and Environment

4.1 Supporting Code

4.1.1 1. Show the sample mean and compare it to the theoretical mean of the distribution.

```
set.seed(777)           # Define random number seed
lambda <- 0.2           # Define the rate parameter
simulationCount <- 1000 # Define the number of simulations to run
sampleSize <- 40        # Define the sample size

# Run Simulation
simulationData <- matrix(rexp(simulationCount*sampleSize, lambda), simulationCount, sampleSize)

# Calculate the row means...
simulationData <- data.frame(simulationData) %>% mutate(row.Means = rowMeans(.[]))
head(simulationData$row.Means)

# Calculate Theoretical Mean
theoreticalMean <- 1/lambda
theoreticalSigma <- 1/lambda /sqrt(sampleSize)
```

4.1.2 2. Show how variable the sample is (via variance) and compare it to the theoretical variance of the distribution.

```
# Calculate the simulated standard deviation from the sample data
standardDeviationDistribution <- sd(simulationData$row.Means)
standardDeviationDistribution

# Calculate the theoretical standard deviation
theoreticalStandardDeviationDistribution <- (1/lambda)/sqrt(sampleSize)
theoreticalStandardDeviationDistribution

# Calculate the distribution varience
distributionVariance <- (standardDeviationDistribution^2)
distributionVariance

# Calculate the theoretical variance
variance_theory <- (((1/lambda)*(1/sqrt(sampleSize)))^2)
variance_theory
```

4.1.3 3. Show that the distribution is approximately normal.

```
## HISTORGRAM CODE

gp <- ggplot(as.vector(simulationData), aes(x=row.Means))
gp <- gp + theme_wsj()  # Use the wall Street Journal Theme for the plot
gp <- gp + geom_histogram(aes(y = ..density..), alpha = 0.2,
```

```
                          binwidth = .15, colour="black", width=.2 )
# One thing we want to do is show the bin count above each bar of the histogram.
# We do this by using the stat_bin() function, where we set the label to the
# count of the bin as defined by the binwidth...
gp <- gp + stat_bin(binwidth = .15,
                    aes( y = (..density..),
                         label = round(..density.. * 10, digits=1),
                         ymax = max(..density..) * 1.05 ),
                    geom  = "text",
                    size  = 3,
                    vjust = -1.5)

# Draw the simulated mean as a vertical line (red)
gp <- gp + geom_vline(xintercept = mean(simulationData$row.Means),
                      colour = 'red',size=.5, linetype = "longdash")
# Draw the theoretical mean as a vertical line (red)
gp <- gp + geom_vline(xintercept = mean(theoreticalMean),
                      colour = 'Blue',size=.5, linetype = "longdash")

# Now we create two lines to ploy the simulated distribution (red), and the
# normal distribution (blue)
gp <- gp + geom_line(aes(y = ..density.., colour = 'Simulated'),
                     stat = 'Density', linetype = "longdash")
gp <- gp + stat_function(geom = "line", fun = dnorm,
                         arg = list(mean = theoreticalMean, sd = theoreticalSigma),
                         size = 1, aes(colour = 'Normal'), fill = NA,
                         linetype = "longdash")
gp <- gp + scale_colour_manual(name = 'Density', values = c('Blue', 'Red'))
# Here we create a plot title across two lines...
title <- paste("Histogram")
title2 <- paste("of Row Means")
gp <- gp + ggtitle(paste0(title,"\n", title2, "\n"))
gp <- gp + theme(legend.position=c(.5, .2),
                 plot.title = element_text(size=10,
                                           lineheight=.8, face="bold"))
gp <- gp + theme(legend.title = element_text(size = 8, face = 'bold'))
gp <- gp + theme(legend.text = element_text(size = 8, face = 'bold'))
gp <- gp + theme(axis.text = element_text(size = 8, face = 'bold'))
# Now display the plot...
print(gp)
```

```
# QQPLOT CODE

y <- quantile(simulationData$row.Means[!is.na(simulationData$row.Means)], c(0.25, 0.75))
x <- qnorm(c(0.25, 0.75))
slope <- diff(y)/diff(x)
int <- y[1L] - slope * x[1L]

d <- data.frame(resids = simulationData$row.Means)

gp <- ggplot(d, aes(sample = resids))
gp <- gp + theme_wsj()  # Use the wall Street Journal Theme for the plot
gp <- gp + stat_qq()
```

```
gp <- gp + geom_abline(slope = slope, intercept = int, color='red')
title <- paste("QQ Plot")
title2 <- paste("of Row Means")
gp <- gp + ggtitle(paste0(title,"\n", title2, "\n"))
gp <- gp + theme(legend.position=c(.75, 0.85),
                 plot.title = element_text(size=10,
                                           lineheight=.8, face="bold"))
gp <- gp + theme(legend.title = element_text(size = 8, face = 'bold'))
gp <- gp + theme(legend.text = element_text(size = 8, face = 'bold'))
gp <- gp + theme(axis.text = element_text(size = 8, face = 'bold'))
print(gp)
```

4.2 References

- *R in Action*
  - By: Robert Kabacoff Publisher: Manning Publications Pub. Date: August 24, 2011, ISBN-10: 1-935182-39-0
- *Mathematical Statistics with Resampling and R*
  - By: Laura Chihara; Tim Hesterberg Publisher: John Wiley & Sons Pub. Date: September 6, 2011 Print ISBN: 978-1-11-02985-5
- *Think Stats, 2nd Edition*
  - By: Allen B. Downey Publisher: O'Reilly Media, Inc. Pub. Date: October 28, 2014 Print ISBN-13: 978-1-4919-0733-7

4.3 Environment

```
# Display R version info
R.version
```

```
##                _
## platform       x86_64-apple-darwin13.4.0
## arch           x86_64
## os             darwin13.4.0
## system         x86_64, darwin13.4.0
## status
## major          3
## minor          2.0
## year           2015
## month          04
## day            16
## svn rev        68180
## language       R
## version.string R version 3.2.0 (2015-04-16)
## nickname       Full of Ingredients
```