Statistical Inference, Coursera

Course Project Part 1: Simulation exercises

Chris Rupley, October 24, 2014

Part 1: Simulation of the Exponential Distribution

We would like to explore the properties of the exponential distribution by simulating it in the R programming language. The exponential distribution has a probability density function of the form:

$$p(x) = \lambda e^{-\lambda x}$$

This function results in a mean of $1/\lambda$ and a standard deviation that is also $1/\lambda$. In order to test this, we can use the function `rexp` in R which generates a series of numbers distributed according to the exponential distribution. I am going to use this function to create a vector of 40 numbers which are exponentially distributed with a lambda value of 0.2. If we want a good statistical representation of the distribution, this process should be repeated several times (I will choose 1000) and we can then compare the results. This can be achieved with the following R code:

```r
# Create 1000 vectors of 40 exponentially distributed numbers
# with a lambda value of 0.2
lambda <- 0.2
n <- 40

set.seed(88)
s <- matrix(n, 1, 1000)
dat <- apply(s, 2, rexp, lambda)
```

This leaves us with a 40 x 1000 matrix of values with each of the columns being a separate vector of exponentially distributed values.

First, I want to compare the mean of these randomly generated values to the theoretical mean of an exponential distribution. I first compute the mean of each column and then take the average of these means which gives the following result.

```
##     Avg Sim Mean Theoretical Mean
##            4.991            5.000
```

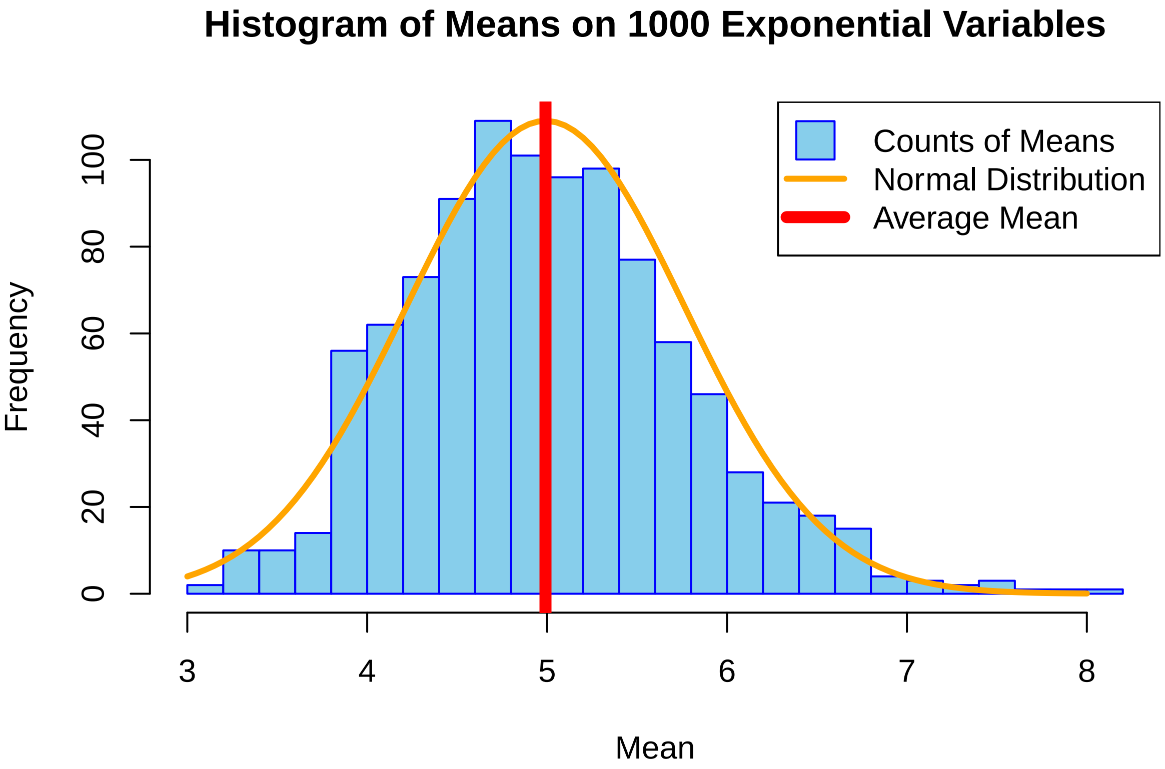
The average mean of these simulated distributions compares quite favorably with the theoretical mean of the exponential distribution, $1/\lambda$, with an error of only 0.18 percent.

We can also compare the variance of the means of these distributions with the theoretical variance, $(1/\lambda)^2/n$, where n is the number of trials (40 in our case).

```
##         Sim Variance Theoretical Variance
##               0.5992               0.6250
```

Which again compares favorably with a difference of 4.1 percent.

The means of these distributions should approximately follow a normal distribution. We can examine this with the help of the following figure.



distribution.pdf

We can see from the plot that the means follow the normal distribution quite nicely.

Lastly, let us look at the spread in values of the means of our simulated distributions. If the means are indeed normally distributed, then 95% of the values should fall within a range of $\bar{X} \pm 1.96 * \sigma/\sqrt{n}$, where $\bar{X}$ is the average mean, σ is the standard deviation ($1/\lambda$ for the exponential distribution), and n is the number of samples (40 in our case). This is our confidence interval where we can say that a mean will fall into this range 95% of the time. From our simulation,

```
##             n          mean   lower limit   upper limit # in interval
##         40.00          5.00          3.45          6.55        951.00
```

And so this gives our confidence interval an actual coverage of 95.1 percent of simulated distributions; very much in line with our expected coverage of 95%.

Appendix: Code

The following is the R code used to generate the data and figures in this report.

```r
# Create 1000 vectors of 40 exponentially distributed numbers
# with a lambda value of 0.2
lambda <- 0.2
n <- 40

set.seed(88)
s <- matrix(n, 1, 1000)
dat <- apply(s, 2, rexp, lambda)

### Part 1: Mean
# Compute the mean and standard deviation of the simulated distributions
colmeans <- apply(dat, 2, mean)
distmeans <- mean(colmeans)
distsd <- sd(colmeans)
theomean <- 1/lambda
theovar <- (1/lambda)^2/n

# Compare the simulated mean to theoretical
meancomp <- c(distmeans, theomean)
names(meancomp) <- c("Avg Sim Mean", "Theoretical Mean")
print(meancomp)

#inline simulation mean deviation from theoretical
round(abs(theomean - distmeans)/theomean * 100, 2)

### Part 2: Variance
# Compare the variance of the simulated means to the theoretical
varcomp <- c(distsd^2, theovar)
names(varcomp) <- c("Sim Variance", "Theoretical Variance")
print(varcomp)

#inline simulation variance deviation from theoretical
round(abs(theovar - distsd^2)/theovar * 100, 1)


### Part 3: Comparison with Normal
histvar <- hist(colmeans, 20, plot = FALSE)
plot(histvar, 20, col = "skyblue", border = "blue",
     xlab = "Mean",
     main = paste("Histogram of Means on 1000 Exponential Variables"))

amp <- max(histvar$counts)*sqrt(2*pi)*distsd
curve(amp*dnorm(x, distmeans, distsd), 3, 8, add = TRUE,
      col = "orange", lwd = 3)
abline(v = distmeans, col = "red", lwd = 6)

legend("topright",
       c("Counts of Means", "Normal Distribution", "Average Mean"),
       pch = c(22,NA,NA),
       col = c("blue", "orange","red"),
       pt.bg = c("skyblue", NA, NA),
       lty = c("blank", "solid", "solid"),
       lwd = c(NA, 3, 6),
       pt.cex = c(3, NA, NA))

### Part 4: Confidence Intervals
# Confidence interval coverage
intlower <- theomean - 1.96 * (1/lambda)/sqrt(n)
intupper <- theomean + 1.96 * (1/lambda)/sqrt(n)

coverage <- colmeans[colmeans > intlower & colmeans < intupper]

output <- c("n" = round(n,0), "mean" = round(theomean,0),
            "lower limit" = round(intlower,2),
            "upper limit" = round(intupper,2),
            "# in interval" = round(length(coverage),0))

print(output)

# inline calculation of coverage of confidence interval
length(coverage) / length(colmeans) * 100
```