Exploring Variances of the Exponential Distribution

ww44ss

Mexico City, October 2014

Intro

This analysis shows how the mean and standard deviation of a finite random sample of a uniform distribution converge to population mean with increasing sample size.

```
##create data for analysis
        ##Set seed to make results reproducible
        set.seed(8675309)

        ##makes sample sizes variable
        rowsofsamples <- 40
        columnsofdistributions <- 1001

        ##set lambda
        lambda <- .2

        ##create data frame using slightly different method than in class
        ##c rows is the sample index and column indexes the distribution.
        rdata <- replicate(columnsofdistributions, rexp(rowsofsamples, lambda))
```

This analysis looks at a set of 1001 distributions of 40 samples.

Question1: Show where the distribution is centered and compare it to the theoretical center of the distribution.

```
##Take means of columns and plot histrogram with mean and theoretical mean

        ##Compute Column means
        expmeans <- colMeans(rdata)
        ##Calculate "Mean of Means"
        meanofdistmeans<-mean(expmeans)
```

The mean of the distribution 5.0123 differs by 0.2453% from the theoretical population mean of 5.

Question 2: Show how variable it is and compare it to the theoretical variance of the distribution.

```
        ## Calculate the Variance
        varofdistmeans = var(expmeans)
        ##expected variance
        expectedvariance = 1/(lambda^2*rowsofsamples)
```

The population variance of the distribution is lambda = 0.625 versus the standard deviation the data 0.6463, which differs by 3.41% .

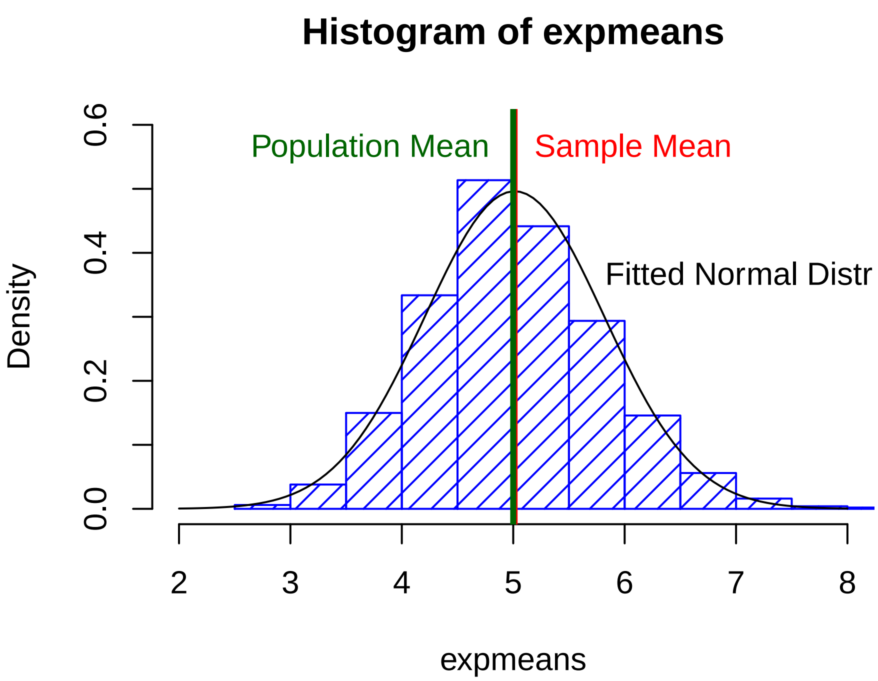
Question3:Show that the distribution is approximately normal.

This is most easily analyzed with a graph.

```
##Generate Histogram
hist(expmeans, prob=TRUE, density=12, angle=45, xlim=c(2,8), ylim=c(0,0.6),col="blue")
curve(dnorm(x, mean=mean(expmeans), sd=sd(expmeans)), add=TRUE)

##Draw lines at mean and theoretical mean = 5
abline(v=meanofdistmeans, lwd=3, col="red")
abline(v=5, lwd=3, col="darkgreen")

##Label means on graph
text(meanofdistmeans,0.55, "Sample Mean", col="red", adj=c(-0.1,0))
text(5,0.55, "Population Mean", col="darkgreen", adj=c(1.10,0))
text(5.5,0.35, "Fitted Normal Distribution", col="black", adj=c(-.10,0))
```



The histogram sample mean, shown in the graph by the red vertical line, is 5.0123 which is barely distinguishable from the popultion mean at 5. The fit of the normal distribution, from the sample mean and standard deviation 0.8039 follows the data well, but not perfectly.

Question4: Evaluate the coverage of the confidence interval

We can analyze how well the normal distribution fits the above sample by looking at the coverage of the confidence interval. We do this by creating samples as above and testing what fraction are in the confidence interval itself for different sample sizes.

The analysis follows that done in the class. I've chosen to calculate the coverage for several sample sizes and plot the results against N. The curve follows a form characteristic of 1/sqrt(N). The coverage is above 95% for a sample size of about 125. For a sample size of 40 coverage is about 80%.

```r
require(ggplot2)
        ##set up calc
        lambda<-.2
        noofreps<-2000
        ##do a large number of sample sizes
        samples<-  floor(runif(200, 10, 200))
        #calculate some parameters
                pmean=1/lambda
                psd=1/lambda
                ssd=psd/sqrt(samples)

##coverage function
coverage <- sapply(samples, function(samples) {
        rdata <- replicate(noofreps, rexp(samples, lambda))
        ##normalize the data by moving center of distribution and then scaling by std dev
        ##this essentially makes the distribution dimensionless
        ##note that sqrt(n) in contained in teh ssd term
        expmeans <- (colMeans(rdata)-pmean)/ssd

        ##calculate number outside limits
        ll <- expmeans - qnorm(0.975)
        ul <- expmeans + qnorm(0.975)
        ##count
        mean(ll < 0 & ul > 0)
})

#plot it
ggplot(data.frame(samples, coverage), aes(x = samples, y = coverage)) + geom_point(size = 2) + geom_hlin
```