Statistical Inference, Part 1

Aditya Salapaka

Friday 20 March 2015

A Basic Analysis of the Exponential Function in R

Overview

An exponential distribution with lambda = 0.2 and 40 observations was simulated 1000 times, and its mean and standard deviation were calculated. These were compared with the theoretical mean and standard deviation. A histogram of the simulated dataset was also created, and a normal distribution curve was superimposed on it to prove that the simulated dataset was approximately normal.

The Exponential Distribution

The exponential function is defined as:

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

where λ is the rate parameter.

The mean of this distribution is $1/\lambda$.

Simulations

The exponential distribution can be generated in R by the `rexp()` function. I generated an exponential distribution with 40 observations and λ = 0.2 and took its mean. I ran this simulation 1000 times and created a data frame with the averages of all these distributions. The seed for the pseudo-random generator was set at 850.

```
lambda <- 0.2
n <- 40
sims <- 1:1000

set.seed(850)

mns <- data.frame(x = sapply(sims, function(x){
        mean(rexp(n, lambda))
}))
```

Sample Mean vs Theoretical Mean

The distribution is centered at the sample mean 4.9915253, which is close to the theoretical mean 5.

Sample variance vs Theoretical Variance

The theoretical variance of this distribution is calculated by applying the central limit theorem. The variance is $\frac{1/\lambda}{\sqrt{n}}$ The sample variance is 0.6250624, which is close to the theoretical variance 0.625.

Distribution

A histogram of the sample data superimposed with a normal distribution curve is shown. It is clear that the distribution is approximately normal.

```
require("ggplot2")
```

```
## Loading required package: ggplot2
```

```
g <- ggplot(mns, aes(x = mns$x))
p <- g + geom_histogram(aes(y = ..density..), binwidth = 0.4, fill = "white", c
olor = "black") +
        stat_function(fun = dnorm, arg = list(mean = 5, sd = sd(mns$x)), colour
= "red") +
        xlab("Means") + ylab("Density")
print(p)
```