Simulation using the exponential distribution

Hashdevrunner

Executive Summary

One of the most important part of learning Statistical Inference subject is to understand the intricacies of distribution functions. It is interesting to verify the effects on expectancy as study begins to gather more data. Using the tools available, the exponential distribution can be simulated in R using function `rexp(n, lambda)`, where $lambda(\lambda)$ is the rate of the parameter and `n` n is number of samples.

For exponential distribution PDF, the mean and standard deviation can be derived as:

$$\mu = \frac{1}{\lambda} = \sigma$$

Simulation of values

To begin with the simulation, we need to create a 1000 simulations for 40 random values. It could be achieved with the code:

```r
library(ggplot2)
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.1.3

## Loading required package: grid
```

```r
set.seed(0)
noSim <- 1e3
lambda <- 0.2
#create a simulation of exponential distribution in a 1000x40 matrix
sim <- matrix(rexp(noSim*40, rate=lambda), noSim, 40)
```

From these simulated values, obtaining the means for each row and displaying the frequency seems to be a normal distribution.

```r
#generate data set for means of the rows
data_est <- data.frame(r = rowMeans(sim),
                       z = as.factor("Actual"))
```

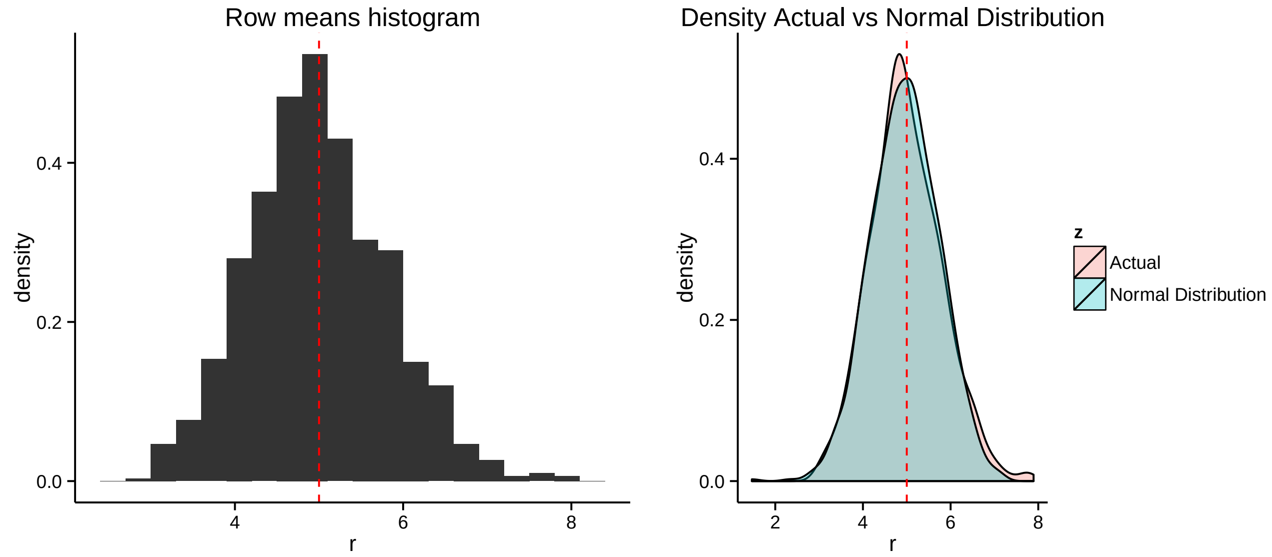
Comparing the estimated density of the distribution vs theoretical normal distribution.

```r
#show to histogram
ghist <- ggplot(data_est,aes(x=r)) +
  geom_histogram(binwidth = .3, aes(y=..density..)) +
  theme_classic() +
  ggtitle(label = "Row means histogram") +
  geom_vline(xintercept = 5, color = "red", linetype = "dashed")

#data set for theoretical normal distribution
data_th <- data.frame(
  r = rnorm(noSim,mean=1/lambda,sd=1/lambda/sqrt(40)),
  z = as.factor("Normal Distribution")
  )

#create one big data set
data_src <- rbind(data_est,data_th)

#Plot side by side
gplot_density <- ggplot(data_src, aes(x=r,fill=z)) +
  geom_density(alpha=.3) +
  theme_classic() +
  ggtitle(label = "Density Actual vs Normal Distribution") +
  geom_vline(xintercept = 5, color = "red", linetype = "dashed")
grid.arrange(ghist, gplot_density, nrow = 1)
```



CLT

From the graph, we expect from the CLT, that theoretical vs actual probability distribution are very close.

```
library(pander)
```

```
## Warning: package 'pander' was built under R version 3.1.3
```

```
tab_df <- data.frame(Mean = c(mean(rowMeans(sim)),1/lambda),
                     Variance = c(mean(apply(sim,1,var)),1/lambda^2))
rownames(tab_df) <- c("Simulated","Theoretical")
pander(tab_df, round=2)
```

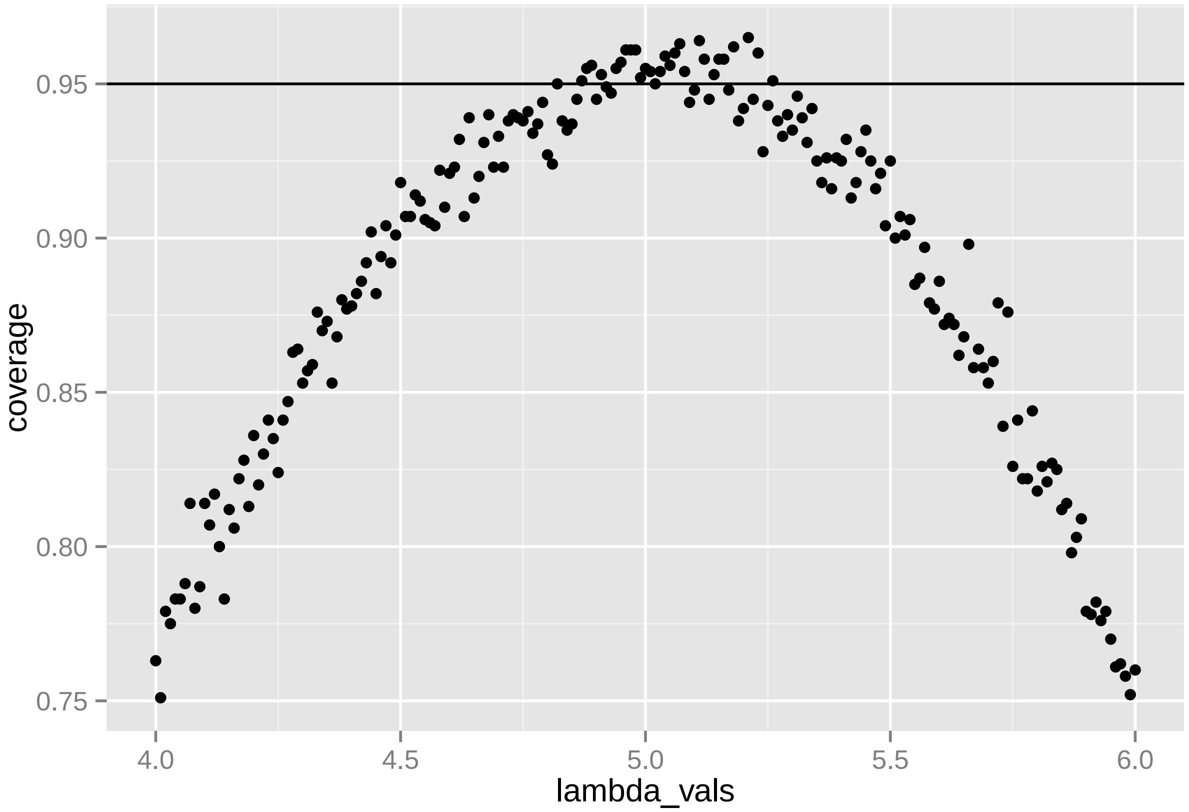
	Mean	Variance
Simulated	4.99	25.24
Theoretical	5	25

Evaluation of coverage of CI

Finally, let's evaluate the coverage of the confidence interval for $1/\lambda = \bar{X} \pm 1.96\frac{S}{\sqrt{n}}$

```
set.seed(0)
lambda_vals <- seq(4, 6, by=0.01)
coverage <- sapply(lambda_vals, function(lamb) {
    mu_hats <- rowMeans(matrix(rexp(40*noSim, rate=0.2),
                               noSim, 40))
    ll <- mu_hats - qnorm(0.975) * sqrt(1/lambda**2/40)
    ul <- mu_hats + qnorm(0.975) * sqrt(1/lambda**2/40)
    mean(ll < lamb & ul > lamb)
})

qplot(lambda_vals, coverage) + geom_hline(yintercept=0.95)
```



The 95% confidence intervals for the rate parameter (λ) to be estimated ($\hat{\lambda}$) are $\hat{\lambda}low = \hat{\lambda}(1 - \frac{1.96}{\sqrt{n}})$ and $\hat{\lambda}upp = \hat{\lambda}(1 + \frac{1.96}{\sqrt{n}})$. As can be seen from the plot above, for selection of $\hat{\lambda}$ around 5, the average of the sample mean falls within the confidence interval at least 95% of the time.

```
mean(lambda_vals)
```

```
## [1] 5
```

Note that the true rate, λ is 5.

Reproducible code: github