Statistical Inference - Simulation Exercise

Abdul Rasheed Narejo

18/09/2018

```r
# load libraries
library(ggplot2) # for visualization
library(e1071) # for statistical analysis
```

A simulation exercise

In this project you will investigate the exponential distribution in R and compare it with the Central Limit Theorem. 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. Set lambda = 0.2 for all of the simulations. You will investigate the distribution of averages of 40 exponentials. Note that you will need to do a thousand simulations.

What is exponential distribution ?

According to wikipedia… In probability theory and statistics, the exponential distribution (also known as the negative exponential distribution) is the probability distribution that describes the time between events in a Poisson point process, i.e., a process in which events occur continuously and independently at a constant average rate.

The exponential distribution may be useful to model events such as

- The time between goals scored in a World Cup soccer match
- The duration of a phone call to a help center
- The time between meteors greater than 1 meter diameter striking earth
- The time between successive failures of a machine
- The time from diagnosis until death in patients with metastatic cancer
- The distance between successive breaks in a pipeline

Simulation Exerpcise

The code for generating random exponential distribution in R is rexp(n,lamda) where n refers to the sample size and lambda is the rate parameter. The mean of exponential distribution is 1/lambda and the standard deviation is also 1/lambda. In our exercise, lambda is set to 0.2 for all the simulations.

set the variables

```r
# define the variables
n = 40
lambda = 0.2
simulations = 1000
```

set seed and generate the data

```
set.seed(1234)
data <- matrix(rexp(n = n * simulations, rate = lambda), nrow = simulations)
```

calculate mean of simulations

```
# calculate mean of each simulation
mean.data <- rowMeans(data)
```

Comparing simulation statistics with theoretical

Calculate mean, variance and standard deviation for simulation

```
# simulation mean
sim.mean <- round(mean(mean.data), 5)
sim.mean
```

```
## [1] 4.97424
```

```
# simulation variance
sim.var <- round(var(mean.data), 5)
sim.var
```

```
## [1] 0.59497
```

```
# simulation standard deviation
sim.sd <- sim.var^0.5
sim.sd
```

```
## [1] 0.771343
```

Calculate theoretical mean, variance and standard deviation

```
# theoretical mean
theo.mean <- 1 / lambda
theo.mean
```

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

```
# theoretical variance
theo.var <- (1/lambda)^2/n
theo.var
```

```
## [1] 0.625
```

```
# theoretical standard deviation
theo.sd <- theo.var ^ 0.5
theo.sd
```

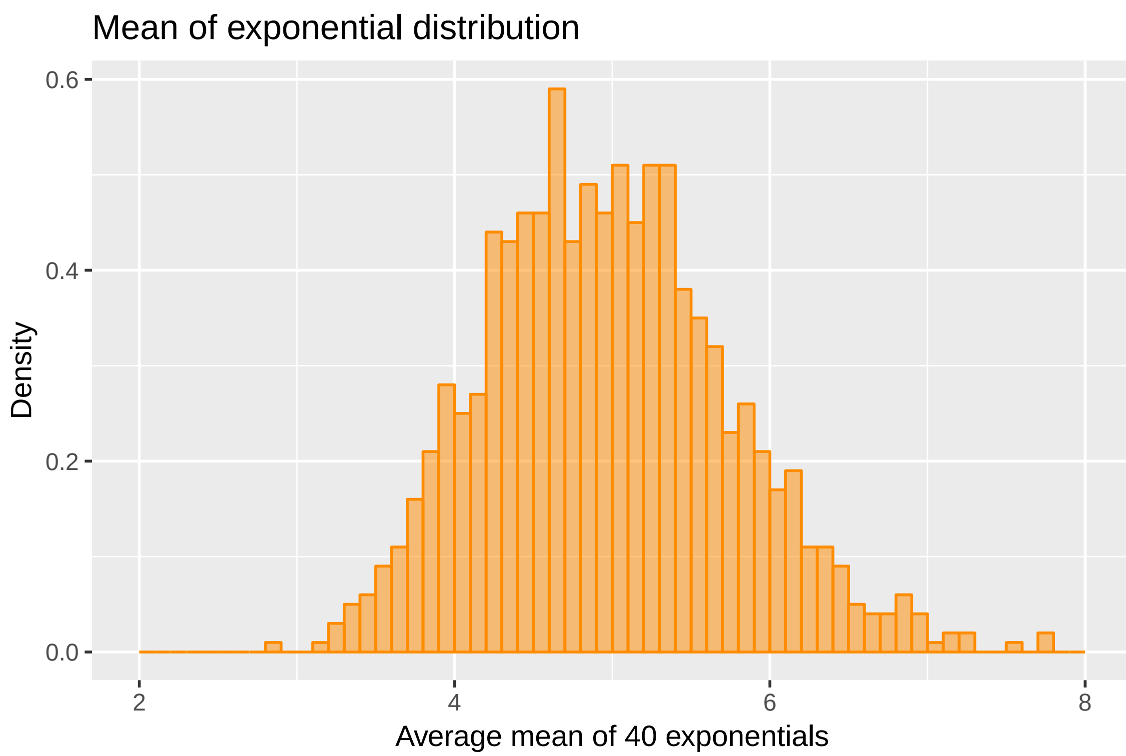
```
## [1] 0.7905694
```

Comparison of sample statistics with theoretical

- Simulation mean is 4.97424 compared to theoretical mean of 5, indicating a negligble difference of -0.52%.
- Simulation variance is 0.59497 compared to theoretical mean of 0.625, indicating a negligble difference of -4.8%.
- Simulation standard deviation is 0.771343 compared to theoretical standard deviation of 0.7905694, indicating a negligble difference of -2.43%.

```
mean.data <- data.frame(mean.data)
names(mean.data) <- c("mean")
```

```
ggplot(data = mean.data, aes(mean, ..density..)) +
    geom_histogram(breaks=seq(2,8, by = .1), col = "dark orange", fill = "dark orange",
                   alpha = .5) +
    labs(title = "Mean of exponential distribution",
         x = "Average mean of 40 exponentials", y="Density")
```



Conclusion

As shown above, the mean and variance of 40 randomly generated sample values of expoential distribution are very close to the theoretical mean and variance. The distribution is approximately normal as proven by,

Appendix

Additional statistical tests to check normal distribution

```
shapiro.test(mean.data$mean)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mean.data$mean
## W = 0.99282, p-value = 9.228e-05
```

```
skewness(mean.data$mean)
```

```
## [1] 0.3369804
```

```
kurtosis(mean.data$mean)
```

```
## [1] 0.0686228
```

```
qqnorm(mean.data$mean)
qqline(mean.data$mean, col = 3)
```