Simulation-Exercise.R

Mahesha

Sun Oct 15 16:48:48 2017

```
setwd("C:\\Users\\Mahesha\\Desktop\\Desktop\\Data_Science\\Courseera\\Statistic
al Inference")
getwd()
```

```
## [1] "C:/Users/Mahesha/Desktop/Desktop/Data_Science/Courseera/Statistical Inf
erence"
```

```
#Load Libraries and set Global Options.
#To suppress loading messages set *message = FALSE*.
#Set global options *echo = TRUE* so others will be able to read the code and s
et *results = hold* to hold & push output to end of chunk.
library(knitr)
opts_chunk$set(echo = TRUE, results = 'hold')
library(data.table)
library(ggplot2)
#
#Set variables as defined in the problem.
n <- 40 # number of exponentials (sample size)
lambda <- 0.2 # lambda for rexp (limiting factor) (rate)
nosim <- 1000 # number of simulations
quantile <- 1.96 # 95th % quantile to be used in Confidence Interval( 95% of th
e area under the normal distribution lies within 1.96 standard deviations of th
e mean.)
set.seed(234) # set the seed value for reproducibility

#Create a matrix with 1000 simulations each with 40 samples drawn from the expo
nential distribution.
# Use rexp() and matrix() to generate 40 samples creating a matrix with 1000 ro
ws and 40 columns.
simData <- matrix(rexp(n * nosim, rate = lambda), nosim)
#Calculate the averages across the 40 values for each of the 1000 simulations.
simMeans <- rowMeans(simData) # Matrix Mean



#Mean Comparison
#Show the sample mean and compare it to the theoretical mean of the distributio
n.
#Mean of Sample Means
#Calculate the actual mean of sample data; the average sample mean of 1000 simu
lations of 40 randomly sampled exponential distributions.
sampleMean <- mean(simMeans) # Mean of sample means
print(sampleMean)
```

```
## [1] 5.001573
```

```
#
#Theoretical Mean
#Calculate the theoretical mean; the expected mean of the exponential distribut
ion of rate = 1/lambda.
theoMean <- 1 / lambda # Theoretical Mean
print(theoMean)
```

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

```
#The distribution of the mean of the sample means is centered at 5.001573 and t
he theoretical mean is centered at 5. The mean of the sample means and the theo
retical mean (expected mean) are very close.
#
#
#Variance Comparison
#Show how variable the sample is and compare it to the theoretical variance of
the distribution.
#Sample Variance
#Calculate the Actual Variance.
sampleVar <- var(simMeans)
print(sampleVar)
```

```
## [1] 0.6631504
```

```
#Theoretical Variance
#Calculate the theoretical variance (expected variance).
theoVar  <- (1 / lambda)^2 / (n)
print(theoVar)
```

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

```
#The variance of the sample means is 0.6631504 and the thoeretical variance of
the distribution is 0.625. Both variance values are very close to each other.
#Sample Standard of Deviation
#Calculate the sample means standard of deviation.
sampleSD <- sd(simMeans)
print(sampleSD)
```

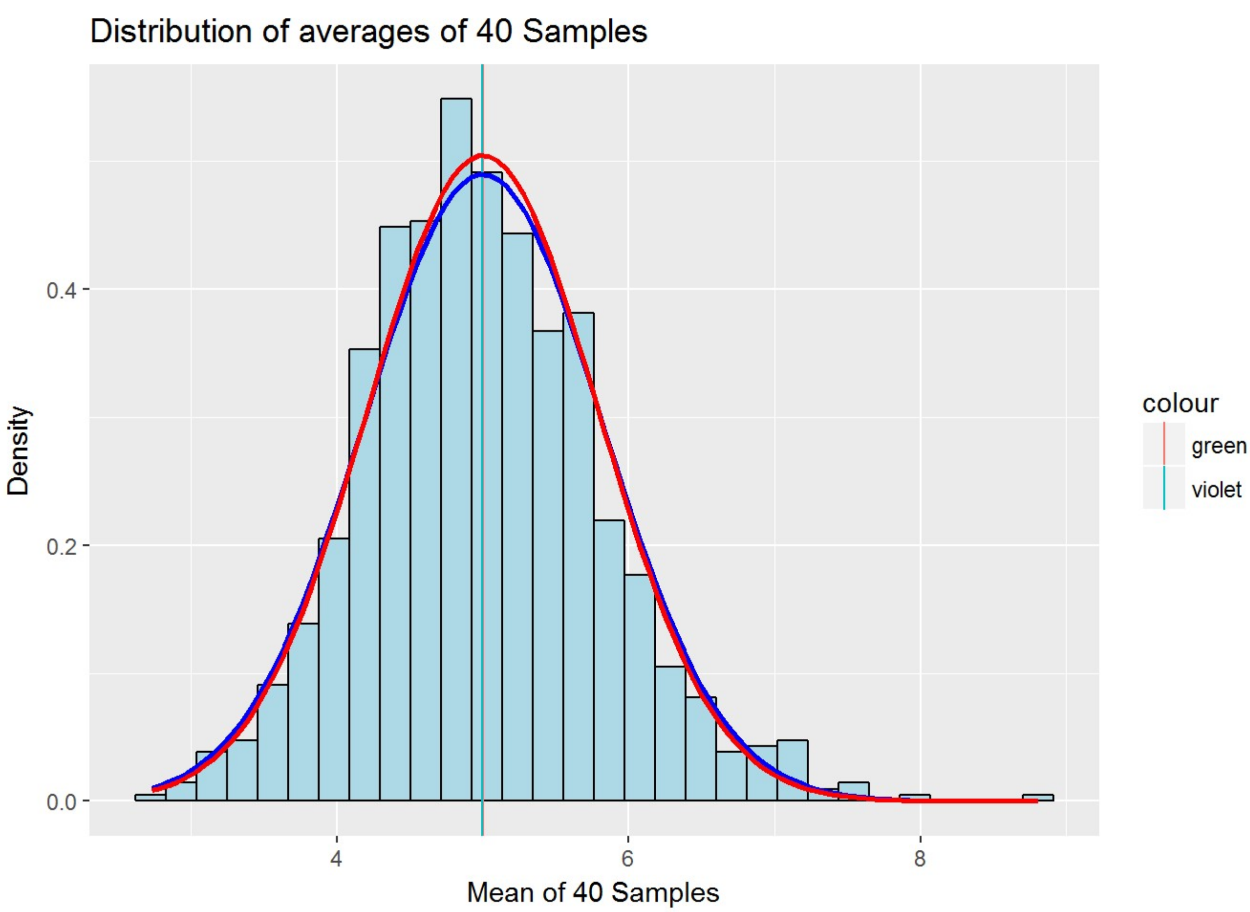
```
## [1] 0.8143405
```

```
#Theoretical Standard of Deviation
#Calculate the theoretical standard of deviation.
theoSD <- 1/(lambda * sqrt(n))
print(theoSD)
```

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

```
#The sample means standard of deviation is 0.8143405 and the thoeretical means
of standard deviation is 0.7905694. Again, the values are close.
#
#RESULTS
#Show that the distribution is approximately normal.
#Display the results to visually compare the actual (sample) values versus the
theoretical values.
plotdata <- data.frame(simMeans)
p <- ggplot(plotdata, aes(x =simMeans))
p <- p + geom_histogram(aes(y=..density..), colour="black",
                        fill = "lightblue")
p <- p + labs(title = "Distribution of averages of 40 Samples", x = "Mean of 4
0 Samples", y = "Density")
p <- p + geom_vline(aes(xintercept = sampleMean, colour = "green"))
p <- p + geom_vline(aes(xintercept = theoMean, colour = "violet"))
p <- p + stat_function(fun = dnorm, args = list(mean = sampleMean, sd = sampleS
D), color = "blue", size = 1.0)
p <- p + stat_function(fun = dnorm, args = list(mean = theoMean, sd = theoSD),
colour = "red", size = 1.0)
p
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.

#The density of the actual data is shown by the light blue bars. The theoretica
l mean and the sample mean are so close that they overlap. The "red" line show
s the normal curve formed by the the theoretical mean and standard deviation. T
he "royal blue" line shows the curve formed by the sample mean and standard dev
iation.
#As you can see from the graph, the distribution of averages of 40 exponential
distributions is close to the normal distribution with the expected theoretica
l values based on the given lambda.

#
#Confidence Interval Comparison
#Check the confidence interval levels to see how they compare.
#Sample CI
#Calculate the sample confidence interval; sampleCI = mean of x plus or minus t
he .975th normal quantile times the standard error of the mean standard deviati
on of x divided by the square root of n (the length of the vector x).
sampleConfInterval <- round (mean(simMeans) + c(-1,1)*1.96*sd(simMeans)/sqrt
(n),3)
print(sampleConfInterval)
```

```
## [1] 4.749 5.254
```

```
#Theoretical CI
#Calculate the theoretical confidence interval; theoCI = theoMean of x plus or
minus the .975th normal quantile times the standard error of the mean standard
deviation of x divided by the square root of n (the length of the vector x).
theoConfInterval <- theoMean + c(-1,1) * 1.96 * sqrt(theoVar)/sqrt(n)
print(theoConfInterval)
```

```
## [1] 4.755 5.245
```

```
#The sample confidence interval is 4.749 5.254 and the theoretical confidence l
evel is 4.755 5.245. The confidence levels also match closely. Again, proving t
he distribution is approximately normal.
#
#
#Conclusion
#It is determined that the distribution does indeed demonstrate the Central Lim
it Theorem; a bell curve. After graphing all the values above and comparing th
e confidence intervals the distribution is approximately normal.
```