Statistical Inference using ToothGrowth Dataset Part 1

Jatin chawda

2/25/2020

Overview

Part - 1 : Simulation

The project consists of two parts: 1. Simulation Exercise to explore inferenced 2. Basic inferential analysis using the ToothGrowth data in the R datasets package.

In this project we 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. We will investigate the distribution of averages of 40 exponentials. Illustrate via simulation and associated explanatory text the properties of the distribution of the mean of 40 exponentials.

What We have to do

1. Show the sample mean and compare it to the theoretical mean of the distribution.
2. Show how variable the sample is (via variance) and compare it to the theoretical variance of the distribution.
3. Show that the distribution is approximately normal.

In point 3, focus on the difference between the distribution of a large collection of random exponentials and the distribution of a large collection of averages of 40 exponentials.

Load Libraries and set Global Options.

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.6.2
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```
library(datasets)
```

Set variables

```
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
set.seed(234) # set the seed value for reproducibility
```

Create a matrix with 1000 simulations each with 40 samples Use rexp() and matrix() to generate 40 samples creating a matrix with 1000 rows and 40 columns.

```
simData <- matrix(rexp(n * nosim, rate = lambda), nosim)
```

Calculate the averages across the 40 values

```
simMeans <- rowMeans(simData) # Matrix Mean
```

Mean Comparison

Show the sample mean and compare it to the theoretical mean

Mean of Sample Means

Calculate the actual mean of sample data

```
sampleMean <- mean(simMeans) # Mean of sample means
sampleMean
```

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

Theoretical Mean

Calculate the theoretical mean

```
theoMean <- 1 / lambda # Theoretical Mean
theoMean
```

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

The distribution of the mean of the sample means is centered at 5.001573 and the theoretical mean is centered at 5. The mean of the sample means and the theoretical 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) sampleVar

Theoretical Variance

```
theoVar  <- (1 / lambda)^2 / (n)
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)
sampleSD
```

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

Theoretical Standard of Deviation

Calculate the theoretical standard of deviation.

```
theoSD <- 1/(lambda * sqrt(n))
theoSD
```

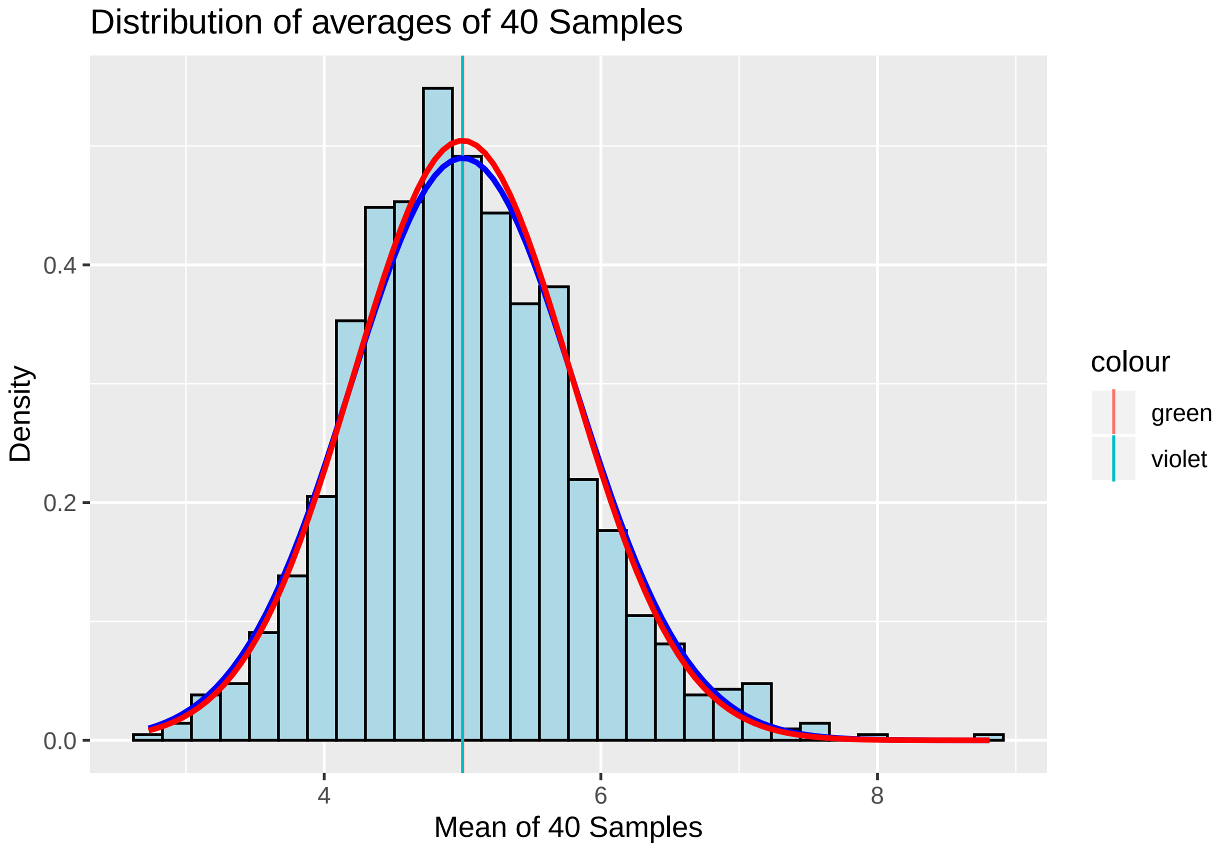
```
## [1] 0.7905694
```

Result

Show that the distribution is approximately normal. Display the results to visually compare the actual versus the theoretical values.

```
plotdata <- data.frame(simMeans)
m <- ggplot(plotdata, aes(x =simMeans))
m <- m + geom_histogram(aes(y=..density..), colour="black",
                        fill = "lightblue")
m <- m + labs(title = "Distribution of averages of 40 Samples", x = "Mean of 40 Samples", y = "Density"
m <- m + geom_vline(aes(xintercept = sampleMean, colour = "green"))
m <- m + geom_vline(aes(xintercept = theoMean, colour = "violet"))
m <- m + stat_function(fun = dnorm, args = list(mean = sampleMean, sd = sampleSD), color = "blue", size
m <- m + stat_function(fun = dnorm, args = list(mean = theoMean, sd = theoSD), colour = "red", size = 1
m
```

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



The density of the actual data is shown by the light blue bars. The theoretical mean and the sample mean are so close that they overlap. The "red" line shows the normal curve formed by the the theoretical mean and standard deviation. The "royal blue" line shows the curve formed by the sample mean and standard deviation. As you can see from the graph, the distribution of averages of 40 exponential distributions is close to the normal distribution with the expected theoretical 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 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).

```
sampleConfInterval <- round (mean(simMeans) + c(-1,1)*1.96*sd(simMeans)/sqrt(n),3)
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).

```r
theoConfInterval <- theoMean + c(-1,1) * 1.96 * sqrt(theoVar)/sqrt(n)
theoConfInterval
```

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

The sample confidence interval is 4.749 5.254 and the theoretical confidence level is 4.755 5.245. The confidence levels also match closely. Again, proving the distribution is approximately normal

Conclusion

It is determined that the distribution does indeed demonstrate the Central Limit Theorem; a bell curve. After graphing all the values above and comparing the confidence intervals the distribution is approximately normal.