Statistical Inference project 1

Kumar Shaket

28/12/2020

Short Summary

Here we want to run a simulation consisting in generating 40 variables from an exponential function with given parameters and repeating this for 1000 times. Then we compute the mean for each simulation(1000 means in total).

Load required packages

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

Building up the data

```
set.seed(1000)
n=40
lambda=0.2
t=1000
```

Form the matrix

```
trialnumber <- replicate(t,rexp(n,lambda))
trialmatrix <- matrix(trialnumber,t,n)
head(trialmatrix)
```

```
##             [,1]      [,2]        [,3]       [,4]       [,5]      [,6]
## [1,]  5.0233101 0.7473493  0.03941261  6.1860307  0.4020792 0.6931959
## [2,]  2.5884649 2.8317688  7.21862819  6.7250282  4.3223908 1.0682927
## [3,] 12.1869221 1.7721079  1.42994496 14.0900957  5.5432357 1.9317167
## [4,] 10.8162445 3.3115762  4.27923195  0.1454799  7.8831155 9.7983511
## [5,]  2.3871486 7.3503527  5.85093695  1.3965076 15.6537778 3.7963294
## [6,]  0.8353505 9.4115464 18.22307295  5.6261530  0.2334372 4.3486865
##             [,7]      [,8]      [,9]      [,10]     [,11]     [,12]      [,13]
## [1,]  7.3171036  3.375721 1.9749825 15.3272373  1.015509 2.6182851  2.5081543
## [2,] 13.5653369 14.182342 3.4783841  0.2865757  2.625609 3.8382534 23.0067680
## [3,]  0.9274014  7.528574 0.7746315  2.5066661  5.365350 7.8066454  2.1587459
## [4,]  1.5583455  8.184750 0.7355128  9.3557438  1.927450 0.1938821  0.2508426
## [5,]  0.5601613  4.971133 3.0321103 10.0318554  3.123647 3.2102485  1.3434465
## [6,]  4.5814735  8.757947 3.5681708  8.7016978 10.469053 2.8121100  3.5402882
##            [,14]      [,15]      [,16]     [,17]     [,18]     [,19]     [,20]
## [1,]  6.0358480 2.44436750  5.5107228 6.6218460 9.0534960 0.2036547  0.330572
## [2,]  0.2360893 2.75492049  8.9577403 5.7946950 6.9155045 0.9587005 13.646423
## [3,] 13.6424379 0.07809449  0.7615727 4.5774226 4.1950357 1.6442855 14.476683
## [4,]  4.8774233 6.17832276  0.4209420 2.4890731 2.3359682 2.5299499  2.834273
## [5,]  3.0697056 4.69673057 13.3954979 2.9061218 3.3622189 2.0992831  2.736501
## [6,]  0.8251210 5.09532352  1.4458056 0.2322347 0.6202688 3.0019916  3.638751
##            [,21]     [,22]     [,23]    [,24]      [,25]     [,26]      [,27]
## [1,]  3.3598121  3.517421  6.062532 6.803458  1.7235236 13.354375  0.8851062
## [2,]  0.7107621  1.861854 30.501119 1.320943 13.4011896  4.576556  2.2342887
## [3,]  0.5049377  7.471164  4.114167 2.947997  0.1973122  6.811534  0.9979929
## [4,] 13.6563071 10.376610  6.202606 1.001129  2.0915063  2.970744 27.8628302
## [5,]  9.7250753  8.549285  1.028210 3.390930  6.2281348  6.709486  2.0103142
## [6,]  9.4243398 22.927850  4.799950 5.092919  9.1261200  1.014617 16.7048876
##           [,28]     [,29]      [,30]     [,31]     [,32]    [,33]     [,34]
## [1,] 0.2045332  5.660417  0.3027708 9.3156681 1.0503828 6.255422 2.5877771
## [2,] 0.7188482  6.262518  1.2241545 3.7567283 1.2395053 1.305814 0.7329292
## [3,] 9.9938605  2.949614  0.2195073 0.2880173 1.8807823 0.805801 5.4253651
## [4,] 0.6936611  2.027666 21.0061633 7.0698954 3.4076968 3.218387 0.4324838
## [5,] 2.3766017  7.720888  7.3278409 2.8289994 0.7989316 3.463637 4.2872219
## [6,] 3.9802013 15.867858 12.1524805 1.1658107 3.8275844 9.314825 7.1338306
##            [,35]      [,36]     [,37]      [,38]      [,39]     [,40]
## [1,]  3.2245284  1.4412888  5.747009 16.9689537  7.8060965 4.3279400
## [2,]  0.3186402 16.1771904  2.498786  3.4464721 18.4641933 8.4664099
## [3,] 11.8335406  0.7555724  1.248725  3.5843796 24.0075388 7.8937531
## [4,]  1.1690175  6.1681166 19.227905  0.4524475  0.1464874 3.8962613
## [5,]  1.1622907 16.8819268  6.983928  4.6813547  4.1859487 4.2484719
## [6,]  7.8486231  2.6653624  1.605143 45.8866585  5.7751790 0.9518475
```

##Calculation Compute sample mean

```r
trialmean <- rowMeans(trialmatrix)
```

Q1 : Show the sample mean and compare it to the theoretical mean of the distribution.

sample mean

```r
smean <- mean(trialmean)
smean
```

```
## [1] 4.986963
```

theorical mean

```r
thmean <- 1/lambda
thmean
```

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

Difference between Sample and Theortical Mean

```r
thmean-smean
```

```
## [1] 0.01303661
```

Therefore we conclude mean difference between sample and theortical is very small

Q2: Show how variable the sample is (via variance) and compare it to the theoretical variance of the distribution

sample variance

```r
sv <- var(trialmean)
sv
```

```
## [1] 0.6583551
```

theortical variance

```r
tv <- (1/lambda)^2 /n
tv
```

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

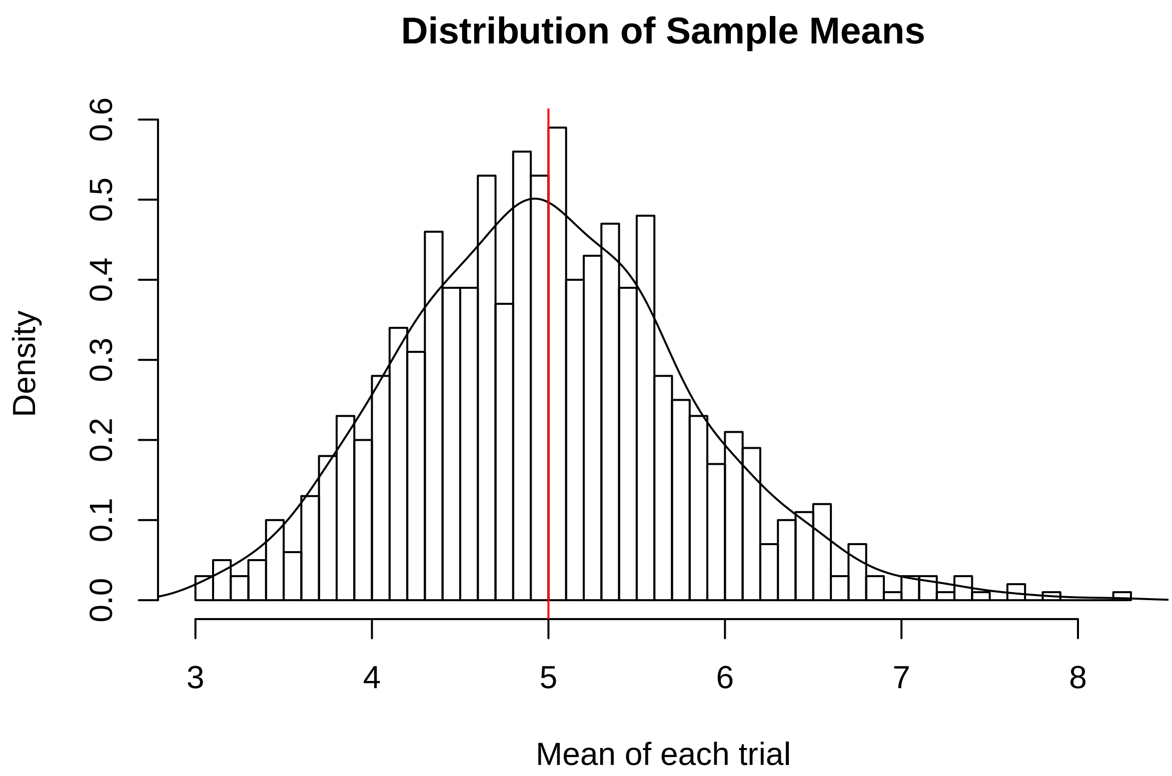
Difference in the variance

```r
tv-sv
```

```
## [1] -0.03335507
```

We conclude variance difference between sample and theortical is very small ## Q3: Show that the distribution is approximately normal.

```r
hist(trialmean,xlab = "Mean of each trial",ylab = "Density",main = "Distribution of Sample Means",breaks
lines(density(trialmean))
abline(v=1/lambda,col='red')
```



approximately normal

The next step is to proof if the distribution is normally distributed, with the confidence interval In this situation, we can compare the confidence interval between theorical and sample. If they have a small difference, that means the means, variance and also distribution tends to be normally distributed

```r
#Compute theortical standard deviation
tsd <- 1/lambda/sqrt(n)
tsd
```

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

```r
#Compute CI in theortical
tci<- thmean +c(-1,1)*1.96 * tsd/n
tci
```

```
## [1] 4.961262 5.038738
```

```r
#Compute actual standard deviation
ssd <- sd(trialmean)
ssd
```

```
## [1] 0.8113908
```

```
#Compute CI in actual
sci<- smean +c(-1,1)*1.96 * ssd/sqrt(n)
sci
```

```
## [1] 4.735511 5.238416
```

From the result, we can see that the distribution is not exactly normally distributied. However, with larger sample size then according to central limit theorem, the data is more likely to be normally distributed.