Using R to Compare the Exponential Distribution with the Central Limit Theorem

Alexander Pyle, apyle@github.com (mailto:apyle@github.com)

December 27, 2015

Overview

The Central Limit Theorem (CLT) posits that the distribution of averages of independent and identically distributed (iid) variables randomly drawn from a population becomes that of a population normal as the sample size increases. This analysis tests this theorem and determines that it holds true for mean and the variance. That is to say, the sample mean will converge to the population mean as the number of simulations of iid varables averaged together increases, and that the variance of these averages will converge to the population variance.

Simulations

To demonstrate the Central Limit Theorem (CLT) this analysis will draw 40 independent and identically distributed (iid) sample values from the exponential distribution. These values will be averaged together. By collecting a large number of these averages we can observe the mean of the averages is close to the theoretical mean of an exponential distribution population.

The theoretical mean for the exponential distribution is $\frac{1}{\lambda}$ while the standard deviation is also $\frac{1}{\lambda}$. Since λ is 0.2, the mean for our population would be $\frac{1}{0.2}$ or 5.

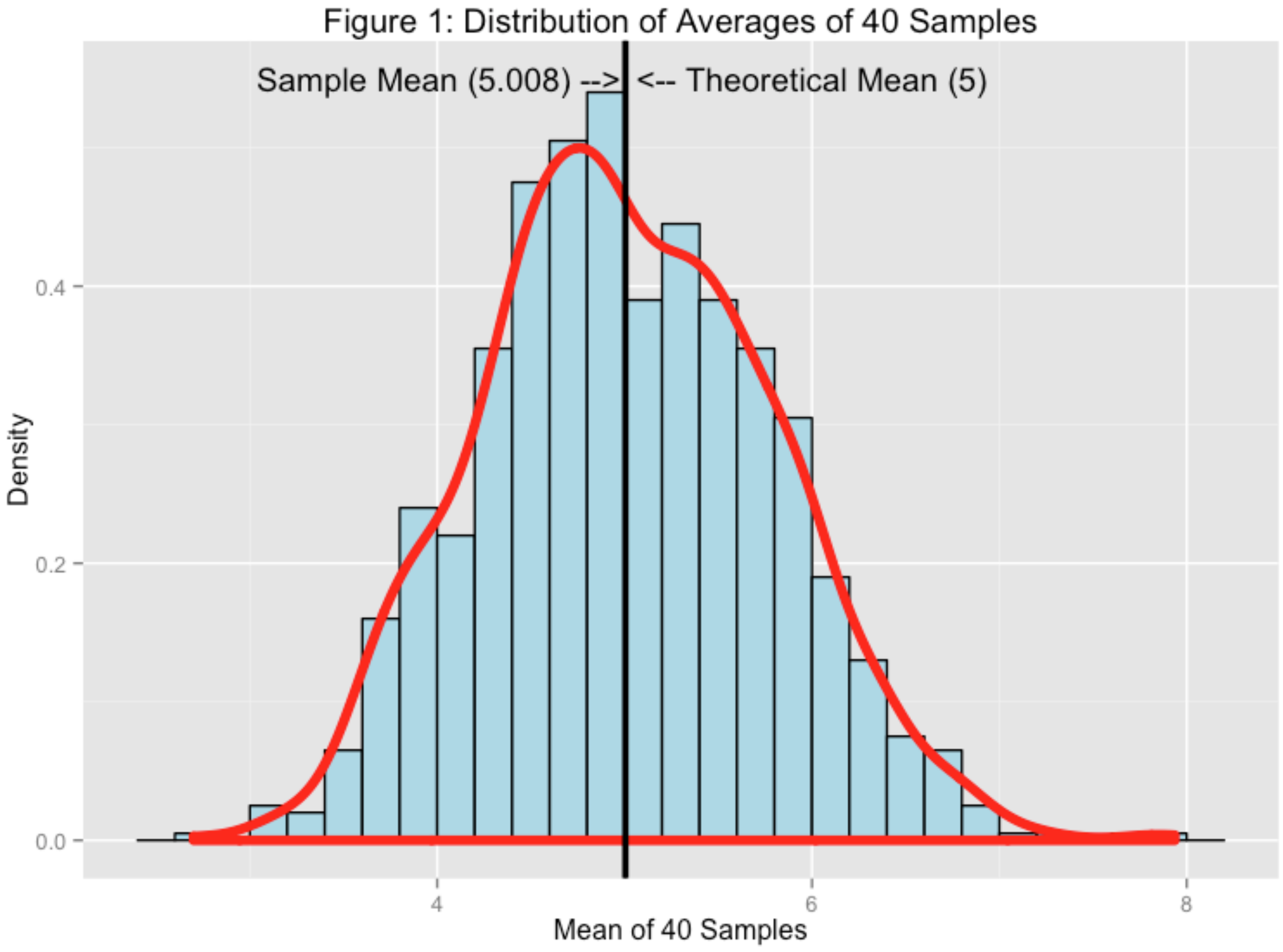
The following code generates 1000 simulations of 40 variables and stores the means of each simulation in `meanexp`. This set of simulations are then plotted with the `data.frame meanDF` variable.

```
set.seed(90125) # seed the random number generator for reproducible results
lambda <- 0.2   # given by assignment
samples <- 40   # number of samples to use for the mean
simulations <- 1000     # number of simulations to run
popmean <- 1/lambda     # theoretical mean of the population
popvariance <- (1/lambda)^2 / samples # theoretical variance of the population

meanexp <- NULL
for (i in 1 : simulations)
        meanexp <- c(meanexp, mean(rexp(samples, lambda)))

samplemean <- mean(meanexp)
meanDF <- as.data.frame(meanexp)
```

```
labYPos = 0.55  # standardize location for labels
g <- ggplot(data = meanDF, aes(x = meanexp))
g <- g + geom_histogram(aes(y = ..density..), fill = "lightblue", binwidth = 0.2, colour = "black")
g <- g + geom_density(size = 2, colour = "red")
g <- g + labs(title = paste("Figure 1: Distribution of Averages of", samples, "Samples"),
              x = paste("Mean of", samples, "Samples"), y = "Density")
g <- g + geom_vline(x = samplemean, size = 1, colour = "black")
g <- g + annotate("text", x = samplemean - 1, y = labYPos,
                  label = paste("Sample Mean (", round(samplemean, 3), ") -->", sep = ""))
g <- g + geom_vline(x = popmean, size = 1, colour = "black")
g <- g + annotate("text", x = popmean + 1, y = labYPos,
                  label = paste("<-- Theoretical Mean (", round(popmean, 3), ")", sep = ""))
g
```



As we can observe from the plot above, the theoretical mean of 5 is closely simulated with the sample mean of 5.008. By simply using 1000 simulations of 40 we have quickly demonstrated the value of using the sample mean as a good approximation of the population mean the iid values have been drawn from.

Variance

The variance for the exponential distribution is $\frac{\sigma^2}{n}$ where n is given as 40. The variance then works out to be

$\frac{(\frac{1}{\lambda})^2}{n} = \frac{(\frac{1}{0.2})^2}{40} = \frac{5^2}{40} = \frac{25}{40} = 0.625.$

To demonstrate the CLT in action we'll generate 100, 1,000, 10,000, and 100,000 simulations and compare the theoreticl population variance of each of these. Since `meanexp` already has 1,000 simulations we will reuse it for the variance.

```
mean100 <- NULL
for (i in 1: 100)
        mean100 <- c(mean100, mean(rexp(samples, lambda)))
mean10K <- NULL
for (i in 1:10000)
        mean10K <- c(mean10K, mean(rexp(samples, lambda)))
mean100K <- NULL
for (i in 1:100000)
        mean100K <- c(mean100K, mean(rexp(samples, lambda)))

df <- data.frame(Samples = c("100", "1,000", "10,000", "100,000"),
                Variance = c(round(c(var(mean100), var(meanexp), var(mean10K), var(mean100K)),4)),
                PopVariance = popvariance,
                Difference = round(c(var(mean100) - popvariance, var(meanexp) - popvariance,
                                     var(mean10K) - popvariance, var(mean100K) - popvariance),3))

vartable <- xtable(df, "Table 1: Variance for Samples of 40 Draws", digits = 3)
vartable <- print.xtable(vartable, type = "html", include.rownames = FALSE, print.results = FALSE)
```

The results are are tabulated below. As we can see, as the number of simulations increases, the sample variance gets closer to the population variance. This is predicted by the CLT since as we increase the number of samples the sample variance will converge to the population variance.

Samples	Variance	PopVariance	Difference
100	0.710	0.625	0.085
1,000	0.598	0.625	-0.027
10,000	0.634	0.625	0.009
100,000	0.629	0.625	0.004

Table 1: Variance for Samples of 40 Draws

Distribution

Refering back to Figure 1 above, we can observe the distribution density in the red line which smooths out the distribution given by the boxes in the plot. The density function closely resembles a normal distribution which we would expect from taking the average of a draw of iid values. It does not completely match the normal Gaussian distribution because we did not draw an infinite number of values, but it already come very close simply with 1000 simulations.

Conclusion

By running simulations of averages of iid variables, we can show show that the Central Limit Theorem holds for the sample mean closely simulating the population mean, the sample variance simulating the population variance, and the distribution of averages closely resembles a normal distribuion with only 1000 simulations.

Appendix

This analysis was run with the following configuration. Running 100,000 simulations took a long time and is not recommended for normal processing without a compelling reason.

```
library(devtools)
devtools::session_info() # display environment the script was create and run in.
```

```
## Session info ---------------------------------------------------------------
```

```
##  setting  value
##  version  R version 3.1.2 (2014-10-31)
##  system   x86_64, darwin10.8.0
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  tz       America/Denver
```

```
## Packages -------------------------------------------------------------------
##  package    * version  date       source
##  colorspace   1.2-4    2013-09-30 CRAN (R 3.1.0)
##  data.table * 1.9.2    2014-02-27 CRAN (R 3.1.0)
##  devtools   * 1.8.0    2015-05-09 CRAN (R 3.1.3)
##  digest       0.6.4    2013-12-03 CRAN (R 3.1.0)
##  evaluate     0.5.5    2014-04-29 CRAN (R 3.1.0)
##  formatR      1.0      2014-08-25 CRAN (R 3.1.1)
##  ggplot2    * 1.0.0    2014-05-21 CRAN (R 3.1.0)
##  git2r        0.10.1   2015-05-07 CRAN (R 3.1.3)
##  gtable       0.1.2    2012-12-05 CRAN (R 3.1.0)
##  htmltools    0.2.6    2014-09-08 CRAN (R 3.1.1)
##  knitr        1.8      2014-11-11 CRAN (R 3.1.2)
##  labeling     0.3      2014-08-23 CRAN (R 3.1.1)
##  MASS         7.3-35   2014-09-30 CRAN (R 3.1.2)
##  memoise      0.2.1    2014-04-22 CRAN (R 3.1.0)
##  munsell      0.4.2    2013-07-11 CRAN (R 3.1.0)
##  plyr         1.8.1    2014-02-26 CRAN (R 3.1.0)
##  proto        0.3-10   2012-12-22 CRAN (R 3.1.0)
##  Rcpp         0.11.3   2014-09-29 CRAN (R 3.1.1)
##  RCurl        1.95-4.3 2014-07-29 CRAN (R 3.1.1)
##  reshape2     1.4.1    2014-12-06 CRAN (R 3.1.2)
##  rmarkdown    0.3.10   2015-01-18 Github (rstudio/rmarkdown@b96214b)
##  rversions    1.0.0    2015-04-22 CRAN (R 3.1.3)
##  scales       0.2.4    2014-04-22 CRAN (R 3.1.0)
##  stringr      0.6.2    2012-12-06 CRAN (R 3.1.0)
##  XML          3.98-1.1 2013-06-20 CRAN (R 3.1.0)
##  xtable     * 1.7-4    2014-09-12 CRAN (R 3.1.1)
##  yaml         2.1.13   2014-06-12 CRAN (R 3.1.0)
```