Central Limit Theorem simulation and a basic inferential analysis of the ToothGrowth dataset in R

Andrew Pratt

Last updated: 2017-08-31

For a final project in the Inferential Statistics course of the Data Science Specialization on Coursera, we were tasked with simulating the distribution of sample means of random variables drawn from an exponential distribution. The purpose of this was to see the Central Limit Theorem at work. We were then tasked with completing a basic analysis of the ToothGrowth dataset in R using any of the inferential statisical techniques we learned in the course. The Appendix at the end of the report contains the code to reproduce the plots that appear in the report.

Part I: Simulation

For our simulation, we'll be drawing random variables from an exponential distribution. Exponential distributions have one parameter, λ. The mean and standard deviation of exponential distributions are both $\frac{1}{\lambda}$.

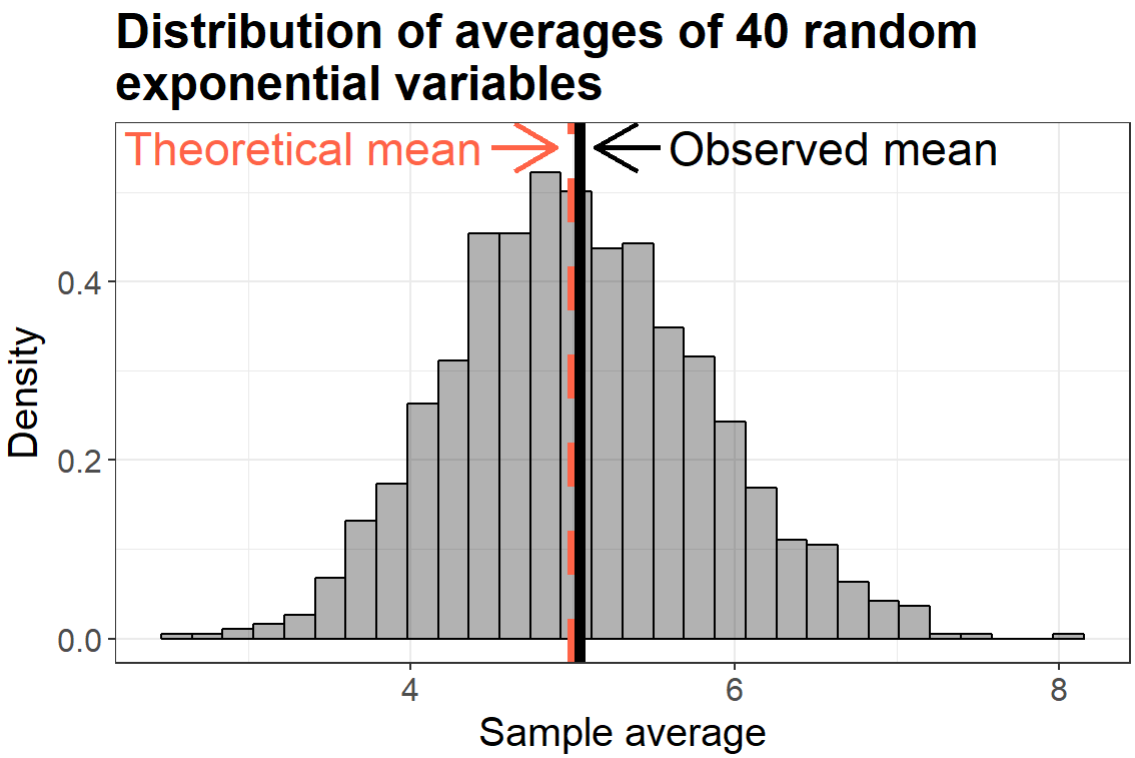
Ok! First, let's simulate 1000 samples of 40 random exponential variables and calculate the mean of each sample:

```
lambda <- .2
n <- 40
nsims <- 1000

set.seed(333)
# exp.sim() is a function that generates means of samples of
# n random exponentials
exp.sim <- function(nsims, n, lambda){
        means <- numeric()
        for(i in 1:nsims){
                means[i] <- mean(rexp(n, lambda))
        }
        means
}

means <- exp.sim(nsims, n, lambda) # generate distribution of sample means
```

Now that we've generated 1000 averages of 40 random exponential variables, let's see how the mean of our distribution of sample means compares to its expected value. The expected, theoretical value of the mean of sample means is just the mean of the original exponential distribution itself. Is that what we see? You bet it is (NOTE: the code to reproduce the plots can be found in the Appendix at the end of the report):



See how the theoretical mean and the observed mean are almost identical? The exact values are:

```
sim.mean <- round(mean(means), 2) # calculate mean of sample means
c("Observed mean" = sim.mean, "Theoretical mean" =  1/lambda)
```

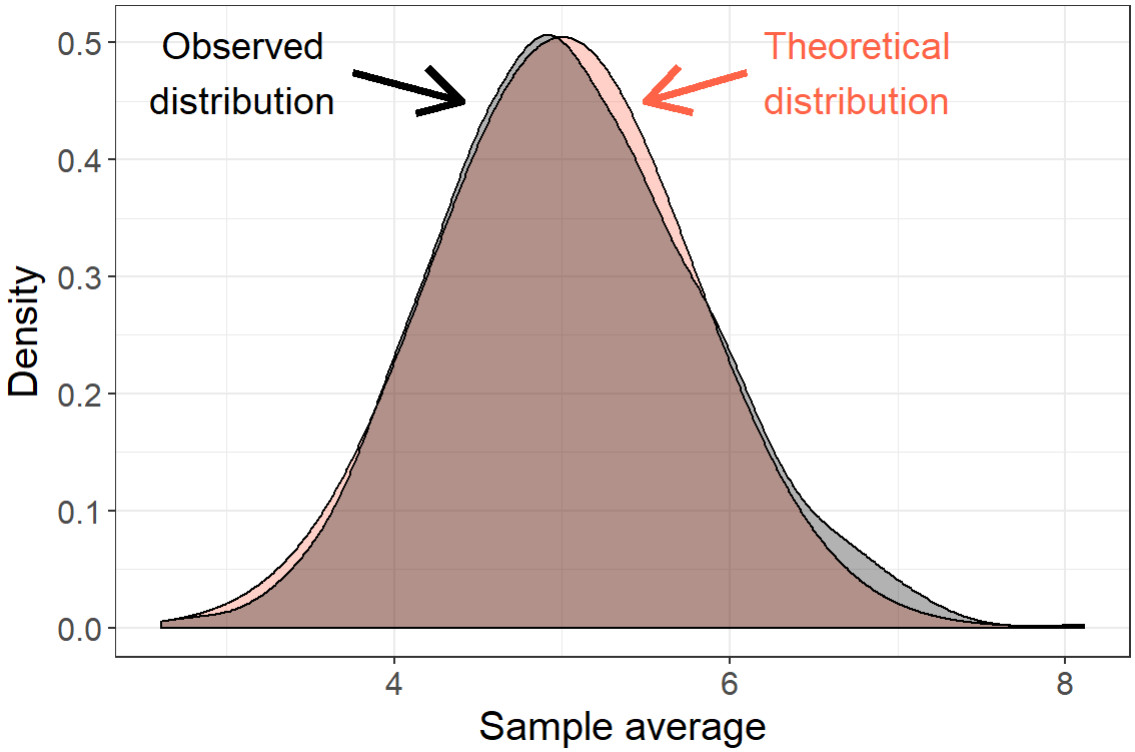
```
##    Observed mean Theoretical mean
##             5.05             5.00
```

That's a difference of only 0.05! Now how about the variance? The theoretical variance of the sample means is the variance of the original distribution divided by the sample size. Since the standard deviation of exponential distributions is $\frac{1}{\lambda}$, their variance must be $\frac{1}{\lambda^2}$. In our simulation, the sample size is 40. Therefore, the theoretical variance of the sample means is $\frac{1}{40\lambda^2}$. How close is that to our simulation?

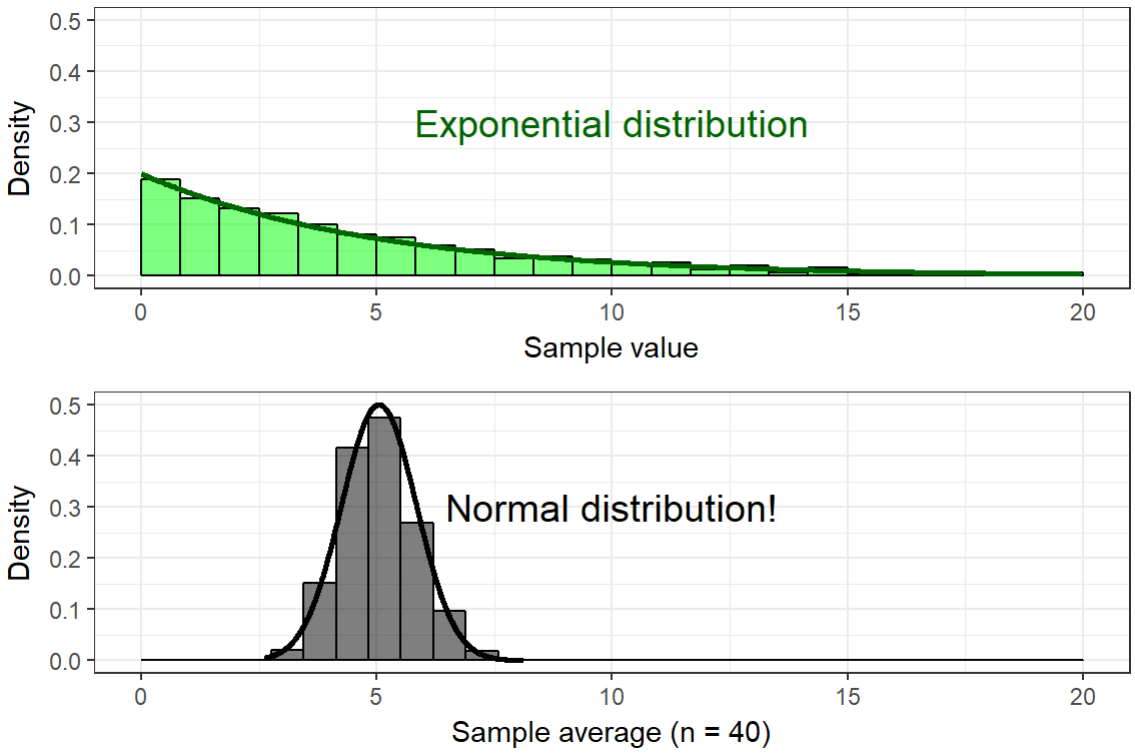
```
sim.sd <- round(sd(means), 2) # sd of sample means
c("Observed variance" = sim.sd^2, "Theoretical variance" = 1/lambda^2/n)
```

```
##    Observed variance Theoretical variance
##                0.640                0.625
```

Again, the difference is very small – only 0.02. To get an idea of just how closely the observed and the theoretical distribution match, take a look at the following plot:



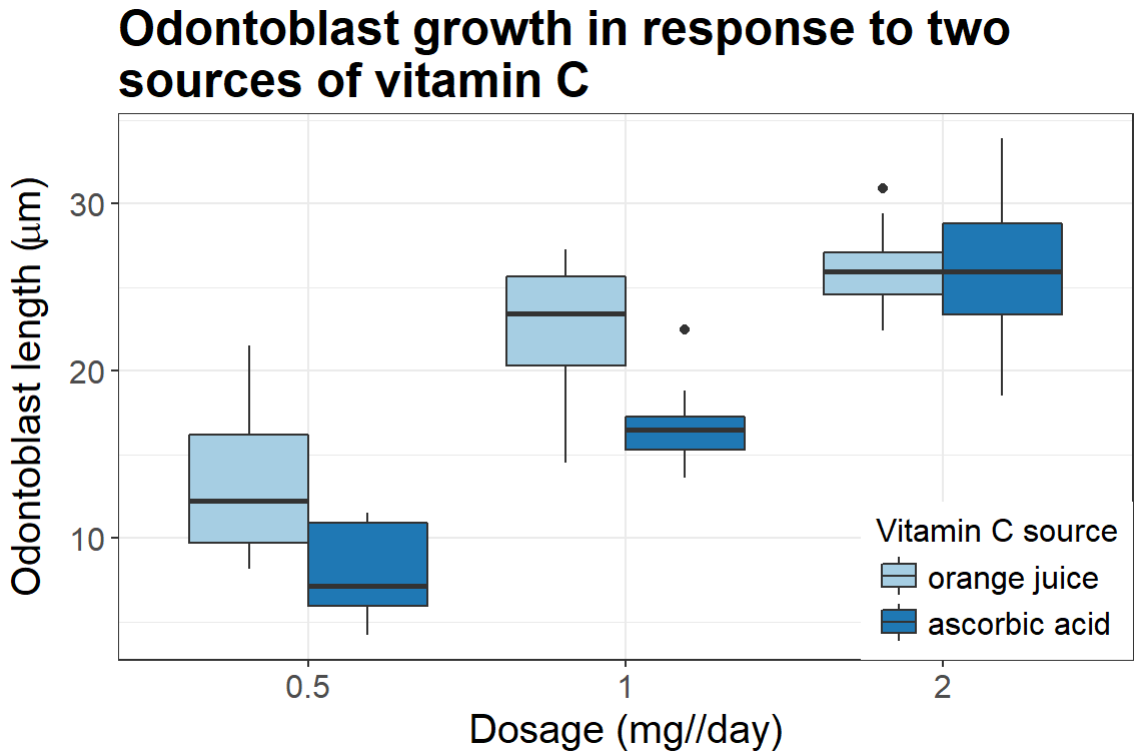
The theoretical distribution is shown in red and the actual observed distribution is shown in black. They overlap almost entirely! It should also have occurred to you by now that the distribution of sample means certainly does not look like the exponential distribution that the samples were drawn from. This is because of the Central Limit Theorem! According to the Central Limit Theorem, for large sample sizes where samples are drawn from ANY distribution with mean μ and variance σ^2, the distribution of the sample means is approximately normal with mean μ and variance $\frac{\sigma^2}{n}$. For example, look at the following figure that shows the difference between the density plots of one large sample of 1000 random exponential (where $\lambda = .2$) and 1000 means of 40 random variables sampled from the same exponential distribution:



See how even though the original distribution is exponential, the distribution of sample means looks like a normal distribution? That's the Central Limit Theorem at work!

Part II: ToothGrowth dataset analysis

The ToothGrowth dataset in R contains data from a study of the effect of vitamin C on tooth growth in guinea pigs. Each of sixty guinea pigs received one of three vitamin C doses (0.5, 1, or 2 mg/day) by one of two vitamin C sources (orange juice or ascorbic acid) and the length of odontoblasts, dentin-producing cells responsible for tooth growth, was measured. Here is a summary of the data:



As the dosage level increases, the length of odontoblasts also increases. Furthermore, it appears that, at least at 0.5 and 1 mg/day of vitamin C, orange juice causes longer odontoblast length than ascorbic acid. Although there are more appropriate ways to analyze the data, we'll simply look at the difference in odontoblast length between the orange juice supplement and the ascoribc acid supplement. We'll perform three t-tests assuming unequal variance and adjust the critical values for the tests using the Benjamini-Hochberg method:

```
library(dplyr)
doses <- c(.5, 1, 2); crit.value <- rep(.05, 3)*1:3/3
t.stat <- rep(NA, 3); p.value <- rep(NA, 3); mean.diff <- rep(NA, 3)
for(i in 1:3){
        subset <- ToothGrowth %>% filter(dose == doses[i])
        test <- with(subset, t.test(len[supp == "OJ"], len[supp == "VC"]))
        t.stat[i] <- test$statistic; p.value[i] <- test$p.value
        mean.diff[i] <- with(test, estimate[1] - estimate[2])
}
p.values <- data.frame(doses, t.stat, p.value, crit.value, mean.diff) %>% arrange(p.value)
p.values <- p.values %>% mutate(sig = ifelse(p.value <= crit.value, "REJECT", "ACCEPT"))
p.values
```

```
##   doses     t.stat     p.value crit.value mean.diff    sig
## 1   1.0  4.0327696 0.001038376 0.03333333      5.93 REJECT
## 2   0.5  3.1697328 0.006358607 0.01666667      5.25 REJECT
## 3   2.0 -0.0461361 0.963851589 0.05000000     -0.08 ACCEPT
```

The printed results show the t-statistic, p-value, adjusted critical value, difference between the mean odontoblast length in the two vitamin C supplements (orange juice - ascorbic acid), and whether or not the null hypothesis (that there is no difference between the means) is rejected. In the lower doses (0.5 and 1 mg/day of vitamin C), odontoblasts in guinea pigs that received orange juice were significantly longer that those that received ascorbic acid. However, by the time the dosage reached 2 mg/day, there was no longer a difference between the two vitamin C supplements.

It is important to always be aware of the assumptions that are made when drawing conclusions with statistical tests. In this case, the assumptions that we've made are the following:

1. The data are independent and identically distributed. This means that none of the data points are in any way dependent on any other data point and that all the data are drawn from the same distribution.
2. The data are normally distributed. If this assumption does not hold, than the calculated t-statistic isn't guaranteed to follow a t-distribution with n-1 degrees of freedom! (Note, however, that this assumption is not as important for large sample sizes because of the central limit theorem.)

Appendix

Plot 1 code

```
# Calculate mean, sd, and range of sample means
sim.mean <- mean(means); sim.sd <- sd(means);sim.range <- range(means)
# Create vector of normal density values to plot on top of histogram
sequence <- seq(sim.range[1], sim.range[2], length = 1000)
norm.curve <- dnorm(sequence, sim.mean, sim.sd)
# (ggplot annotation information)
xblack <- c(sim.mean + .05, sim.mean + .5, sim.mean + .55)
xred <- c(1/lambda - .05, 1/lambda - .5, 1/lambda - .55)
yblack <- yred <- max(norm.curve) + .05
# Plot a histogram of the sample means with observed and theoretical means.
library(ggplot2)
ggplot() + geom_histogram(aes(means, ..density..), color = "black", fill = "black",
                          alpha = .3, boundary = 0) +
        geom_vline(xintercept = 1/lambda, size = 2, color = "tomato", linetype = "dashed") +
        geom_vline(xintercept = sim.mean, color = "black", size = 2) +
        labs(x = "Sample average", y = "Density",
             title = "Distribution of averages of 40 random \nexponential variables") +
        annotate("segment", x = xblack[2], xend = xblack[1], y = yblack, yend = yblack,
                 color = "black", arrow = arrow(), size = 1) +
        annotate("text", x = xblack[3], y = yblack, label = "Observed mean",
                 color = "black", hjust = 0, size = 6) +
        annotate("segment", x = xred[2], xend = xred[1], y = yred, yend = yred,
                 arrow = arrow(), color = "tomato", size = 1) +
        annotate("text", x = xred[3], y = yred,
                 label = "Theoretical mean", color = "tomato", hjust = 1, size = 6) +
        theme_bw() + theme(plot.title = element_text(size = 18, face = "bold"),
                axis.text = element_text(size = 12), axis.title = element_text(size = 15))
```

Plot 2 code

```
# Draw the theoretical distribution
theory.curve <- dnorm(sequence, 1/lambda, 1/lambda/sqrt(40))
# annotate() information for plot
xblack2 <- c(3.75, 4.4); yblack2 <- c(.475, .45)
xred2 <- c(6.1, 5.5); yred2 <- c(.475, .45)
# Plot the theoretical and observed distributions with vertical lines
# at the means
ggplot() + geom_area(aes(sequence, theory.curve), color = "black", fill = "tomato", alpha = .3)
+
        geom_density(aes(means), fill = "black", alpha = .3) +
        labs(x = "Sample average",
             y = "Density") +
        annotate("segment", x = xblack2[1], xend = xblack2[2], y = yblack2[1], yend =
yblack2[2],
                 arrow = arrow(), size = 1.5) +
        annotate("text", label = "Observed \ndistribution", x = xblack2[1] - .1, y = yblack2[1],
                 hjust = 1, size = 5) +
        annotate("segment", x = xred2[1], xend = xred2[2], y = yred2[1], yend = yred2[2],
                 arrow = arrow(), size = 1.5, color = "tomato") +
        annotate("text", label = "Theoretical \ndistribution", x = xred2[1] + .1, y =
yblack2[1],
                 hjust = 0, size = 5, color = "tomato") +
        theme_bw() + theme(axis.title = element_text(size = 15),
                           axis.text = element_text(size = 12))
```

Plot 3 code

```
# Plot the distribution of 1000 random exponentials and compare it
# to the distribution of 1000 means of 40 random exponentials.
library(ggplot2)
library(gridExtra)
set.seed(222)
sample <- rexp(nsims, lambda)
density.x <- seq(0, range(sample)[2], length = 1000)
density.y <- dexp(density.x, .2)
plot1 <- ggplot() + geom_histogram(aes(sample, ..density..), bins = 25, color = "black",
                           fill = "green", alpha = .5, boundary = 0) +
        geom_line(aes(density.x, density.y), color = "darkgreen", size = 1) +
        xlim(0, 20) + ylim(0, .5) +
        annotate("text", label = "Exponential distribution",
                 x = 10, y = .3, size = 5, color = "darkgreen") +
        labs(x = "Sample value (n = 1)", y = "Density") + theme_bw()
plot2 <- ggplot() + geom_histogram(aes(means, ..density..),
                                   color = "black", fill = "black", alpha = .5, boundary = 0) +
        geom_line(aes(sequence, norm.curve), size = 1) +
        xlim(0, 20) + labs(x = "Sample average (n = 40)", y = "Density") +
        annotate("text", label = "Normal distribution!",
                 x = 10, y = .3, size = 5) + theme_bw()
grid.arrange(plot1, plot2, nrow = 2)
```

Plot 4 code

```
library(datasets)
data("ToothGrowth")
ToothGrowth$dose <- factor(ToothGrowth$dose)

# theme information for the plot
theme <- theme(axis.text = element_text(size = 12), axis.title = element_text(size = 15),
               plot.title = element_text(size = 18, face = "bold"),
               legend.title = element_text(size = 12), legend.text = element_text(size = 12),
               legend.position = c(1,0), legend.justification = c(1,0))

library(RColorBrewer)
library(ggplot2)
ggplot(ToothGrowth, aes(dose, len)) +
        geom_boxplot(aes(fill = supp)) +
        scale_fill_brewer(palette = "Paired", name = "Vitamin C source",
                          labels = c("orange juice", "ascorbic acid")) +
        labs(x = "Dosage (mg//day)", y = expression("Odontoblast length ("*mu*"m)"),
             title = "Odontoblast growth in response to two \nsources of vitamin C") +
        theme_bw() + theme
```