Statistical Inference, part 1

Jen Becker

July 22, 2015

In this paper, I will use R's exponential distribution to illustrate the predictions of the Central Limit Theorem, namely that the distribution of sample means and variances from a population is nearly normal, even if the population distribution is not normal, and that the mean of these sample measurements approximates the population's measurements with increasing accuracy as the number of samples increases.

To simulate the exponential distribution in R, we use rexp(n, lambda) with n=40 and lambda=0.2. Data from a single run will generally resemble an exponential distribution. If we plot the data from 1,000 simulations of 40 exponentials, we see a much more accurate depiction of the exponential distribution.

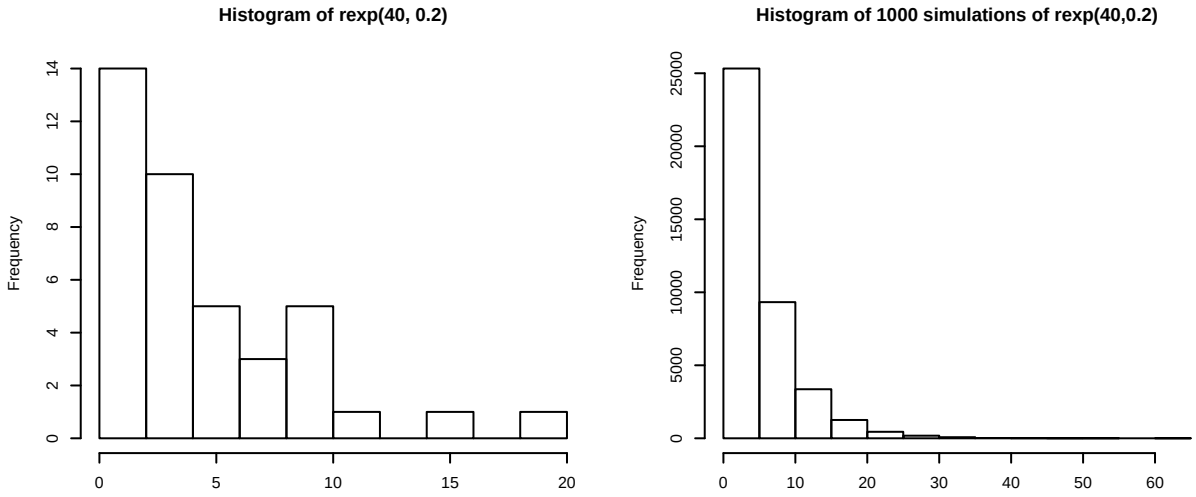

Figure 1: See appendix for supporting R code

Mean

The distribution of sample means from the simulations closely follows a normal distribution.

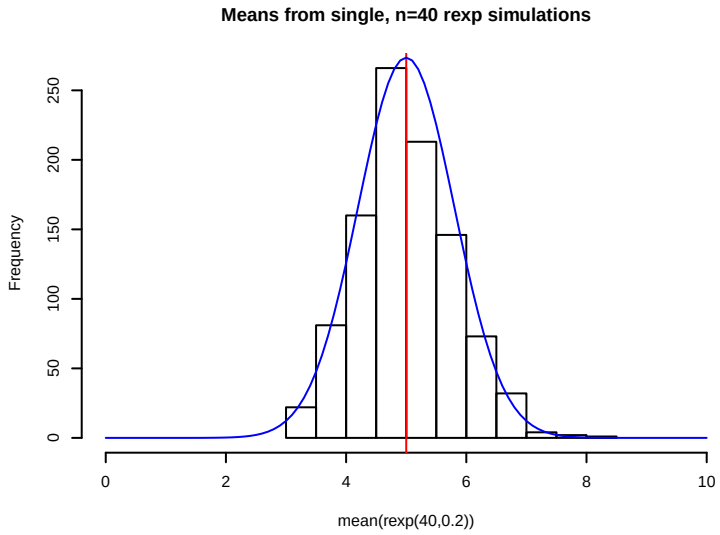

Figure 2: See appendix for supporting R code

While the mean of a single, n=40 simulation may vary from the population mean, as we gather more means and average them together, we can see the value approach the population mean.

The theoretical mean of an exponential distribution is 1/lambda, or 5 in this case. The mean of the sample means from our 1,000 simulations of 40 exponentials is 4.9997019. This is only off by 0.00596%.

We can see that as more simulations are run, the mean of the sample means approaches the theoretical mean.

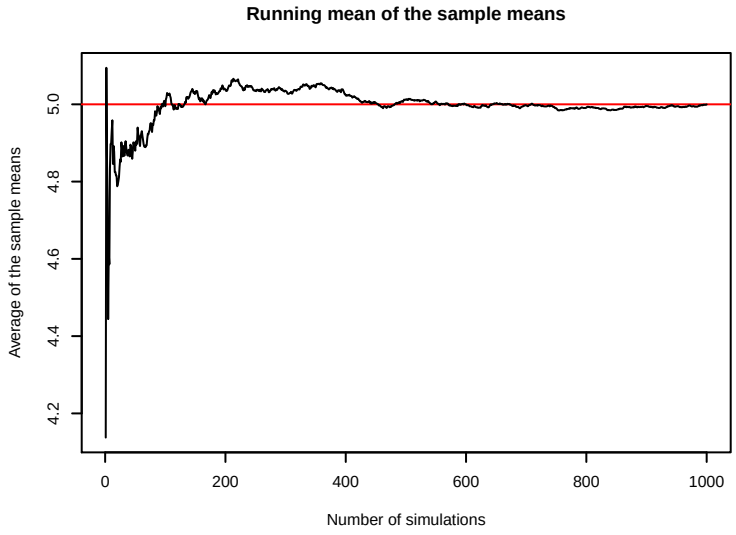

Figure 3: See appendix for supporting R code

Variance

The theoretical variance of this exponential distribution is (1/lambda)^2, or 25 in this case. The mean of the sample variations from our 1,000 simulations of 40 exponentials is 25.3728703. This is only off by -1.49%.

Again, in a single simulation, the variation may vary from the theoretical variation, but over the course of many simulations, the variances gather in a normal distribution around the population variation.

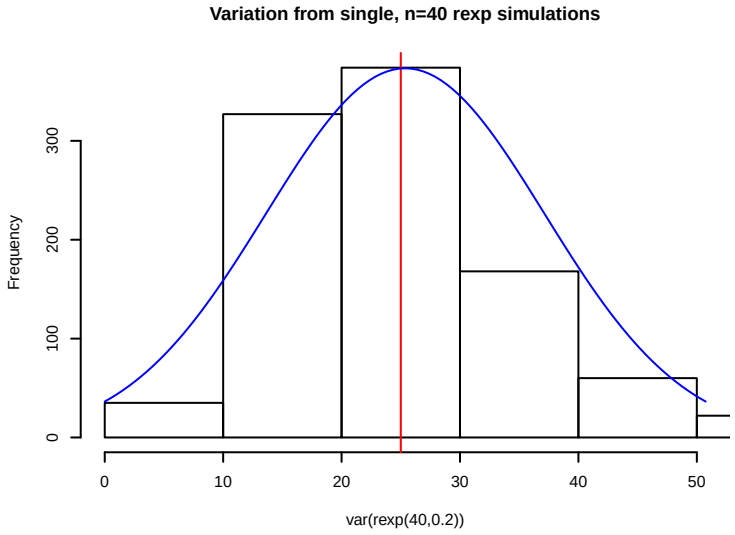

Figure 4: See appendix for supporting R code

And again, as in the mean, as more simulations are run and their variances averaged together, the mean of the sample variance approaches the population variance.

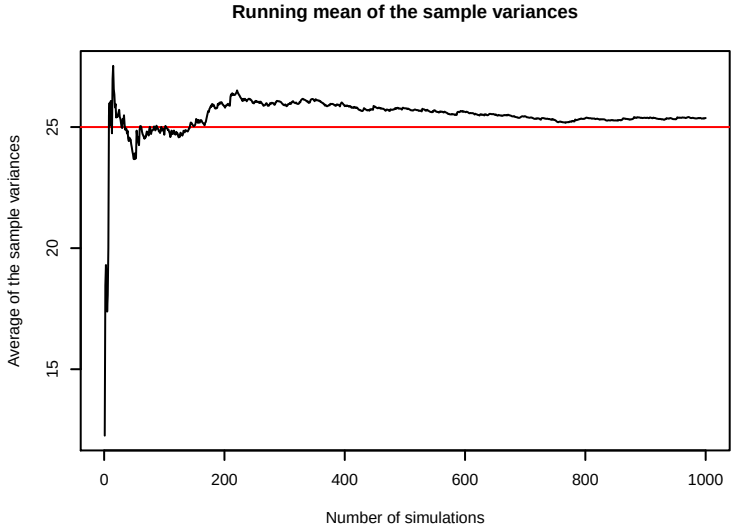

Figure 5: See appendix for supporting R code

Appendix

Figure 1:

```
## Histogram of 40 exponential values
hist(rexp(40,0.2), xlab="")
```

```
## Run 1,000 simulations with n=40
set.seed(100)
alldata <- c()
for (i in 1:1000) {
    thisrun <- rexp(40, 0.2)
    alldata <- c(alldata, thisrun)
}

## Histogram of 1,000 * 40 exponential values.
hist(alldata, main="Histogram of 1000 simulations of rexp(40,0.2)", xlab="")
```

Figure 2:

```
## Run 1,000 simulations with n=40.
## Save the mean for each run.
set.seed(100)
simedmeans <- c()
for (i in 1:1000) {
    thisrun <- rexp(40, 0.2)
    simedmeans <- c(simedmeans, mean(thisrun))
}

## Graph the means for each run with a normal distribution curve
hist(simedmeans, main="Means from single, n=40 rexp simulations",
     xlab="mean(rexp(40,0.2))", xlim=c(0, 2*mean(simedmeans)))
abline(v=1/0.2, col="red")
curve(dnorm(x, mean=mean(simedmeans), sd=sd(simedmeans))*550, add=TRUE, col="blue")
```

Figure 3:

```
## Run 1,000 simulations with n=40.
## Keep a running mean of the means of the simulations.
set.seed(100)
simedmeans <- c()
runningmean <- c()
for (i in 1:1000) {
    thisrun <- rexp(40, 0.2)
    simedmeans <- c(simedmeans, mean(thisrun))
    runningmean <- c(runningmean, mean(simedmeans))
}

## Graph the mean of the sample means as it changes as more simulations are run
## with a line representing the theoretical mean.
plot(x=1:length(runningmean), y=runningmean, pch=20, type="n",
     xlab="Number of simulations", ylab="Average of the sample means",
     main="Running mean of the sample means")
```

```
abline(h=1/0.2, col="red")
lines(x=1:length(runningmean), y=runningmean)
```

Figure 4:

```
## Run 1,000 simulations with n=40.
## Save the variation for each run.
set.seed(100)
simedvars <- c()
for (i in 1:1000) {
    thisrun <- rexp(40, 0.2)
    simedvars <- c(simedvars, var(thisrun))
}

## Graph the variations for each run with a normal distribution curve
hist(simedvars, main="Variation from single, n=40 rexp simulations",
     xlab="var(rexp(40,0.2))", xlim=c(0, 2*mean(simedvars)))
abline(v=(1/0.2)^2, col="red")
curve(dnorm(x, mean=mean(simedvars), sd=sd(simedvars))*11000, add=TRUE, col="blue")
```

Figure 5:

```
## Run 1,000 simulations with n=40.
## Keep a running mean of the variances of the simulations.
set.seed(100)
simedvars <- c()
runningvarmean <- c()
for (i in 1:1000) {
    thisrun <- rexp(40, 0.2)
    simedvars <- c(simedvars, var(thisrun))
    runningvarmean <- c(runningvarmean, mean(simedvars))
}

## Graph the mean of the sample variances as it changes as more simulations are run
## with a line representing the theoretical variance.
plot(x=1:length(runningvarmean), y=runningvarmean, pch=20, type="n",
     xlab="Number of simulations", ylab="Average of the sample variances",
     main="Running mean of the sample variances")
abline(h=(1/0.2)^2, col="red")
lines(x=1:length(runningvarmean), y=runningvarmean)
```