

Is an Exponential Distribution a Normal Distribution?

Lucas Qualmann

9/23/2019

Overview

This report will show how exponential distributions fall under the normal distribution category. To show this, we will look at the means, variance, and distribution of 1,000 means of 40 random exponential numbers and compare them to the expected values.

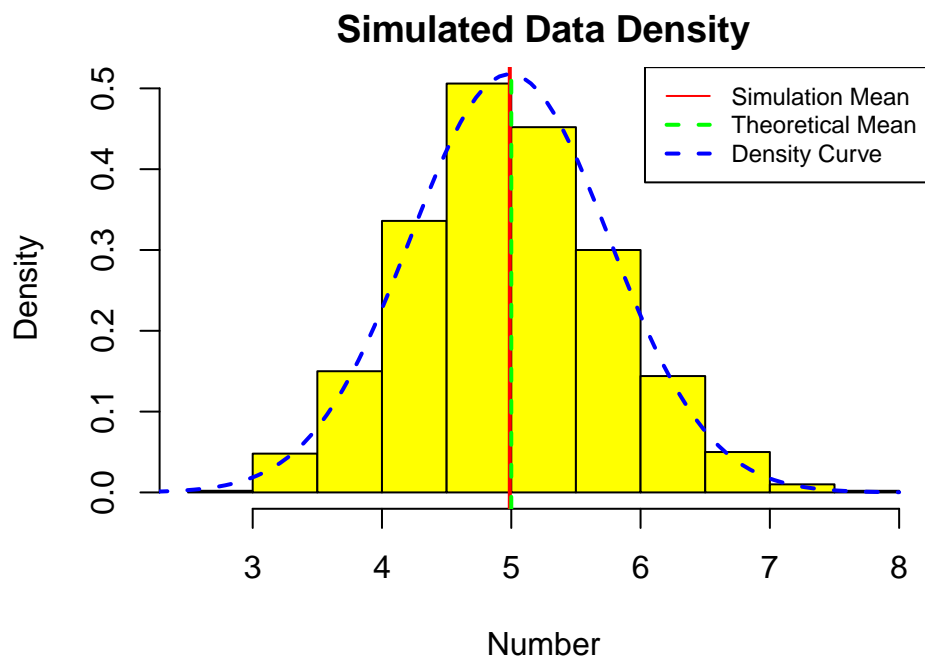
Simulations

First we need to run a simulation. The simulation will be 1,000 numbers, but they will be generated by taking the mean of 40 random exponential numbers with lambda (the rate) set to 0.2.

```
#simulation for 1000 averages of 40 random exponentials
set.seed(4000)
rmeans = NULL
for(i in 1:1000) {
  rmeans <- c(rmeans, mean(rexp(40, 0.2)))
}
```

Simulation Mean vs Theoretical Mean

Now we would like to compare the sample mean to the theoretical mean. The plot below shows a histogram showing the density of observations of the simulation. We can figure out the simulation mean by simply taking the mean of the simulation data. The theoretical mean is simply $1/\lambda$ ($1/0.2$) Included on the plot is both the simulation and theoretical mean and the density curve.



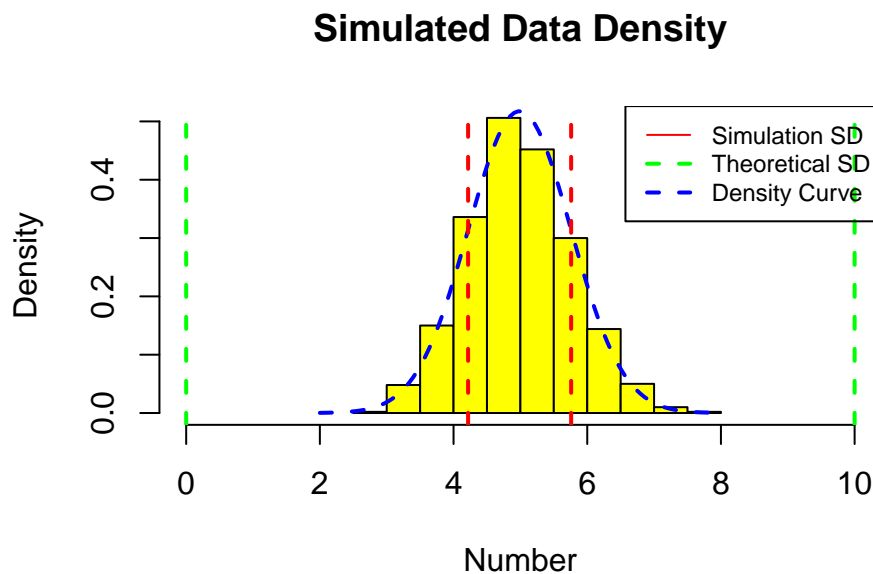
As you can see from the plot, both means are near 5. This shows that the simulation mean is very close to what we'd expect.

```
##           Statistic      Mean
## 1 Theoretical Mean 5.000000
## 2 Simulation Mean 4.988463
```

This table shows the actual values of the means. Notice how the difference between the two is less than .02. This shows how close the means are to each other

Simulation Variance vs Theoretical Variance

Now we want to see how the variances are different between the simulation and theoretical. For this, we will look at the standard deviation which is the square root of the variance, since it is easier to chart. We'll look at the same plot as before, only we'll show 1 standard deviation above and below the mean for the simulation and the theoretical.



As you can see, there is a huge difference in the standard deviations. Theoretically, the standard deviation should be 5, whereas the actual standard deviation is less than 1 as the table below shows.

```
##           Statistic Standard.Deviation
## 1 Theoretical Mean           5.000000
## 2 Simulation Mean           0.770832
```

The actual standard deviation is 0.77. Why is that? This occurred due to the fact we are looking at the mean of 40 random exponential numbers and not actual exponential numbers. If we compare the standard deviation of the simulated data to the standard error ($sd/\sqrt{40}$), we end up with a value of 0.79, which is really close to the standard deviation of the simulated data.

```
print(1/0.2/sqrt(40))
```

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

Distribution

Finally we want to prove the central limit theorem applies for exponential numbers by showing our simulation data forms a normal distribution. As you can see in the plots above, the density curve forms a bell curve, so that would imply a normal distribution. To take it one final step, let's look at the percentage of values that

fall in 1-3 standard deviations for the 1,000 means dataset and compare it to the percentage that should fall into those levels in a regular normal distribution.

##	Standard_Deviations	Normal_Distribution	Simulation_Percents
## 1	1	0.68	0.685
## 2	2	0.95	0.950
## 3	3	0.99	0.997

Just about perfect! 68.5% of the simulation values fall within 1 standard deviation of the mean which is right in line with the 68% that we would expect if it was a normal distribution. The same holds true for both 2 and 3 standard deviations from the mean. Both the eyetest and the actual range of values compared to the standard deviation both show how exponential numbers fall under a normal distribution.

Appendix

R Code

```
#simulation for 1000 averages of 40 random exponentials
set.seed(4000)
rmeans = NULL
for(i in 1:1000) {
  rmeans <- c(rmeans, mean(rexp(40, 0.2)))
}

#comparing means of simulations
par(mar = c(4,4,2,1))
hist(rmeans, col = "yellow", xlab = "Number", freq = FALSE,
     main = "Simulated Data Density")
abline(v = mean(rmeans), col = "red", lwd = 2)
abline(v = 1/0.2, col = "green", lwd = 2, lty = 2)
x.mean <- seq(2, 8, length.out=100)
y.mean <- dnorm(x.mean, mean(rmeans), sd(rmeans))
lines(x.mean, y.mean, col = "blue", lwd = 2, lty = 2)
legend("topright", legend = c("Simulation Mean", "Theoretical Mean",
                             "Density Curve"), cex = 0.75,
      col = c("red", "green", "blue"), lwd = c(1, 2, 2), lty = c(1, 2, 2))

means <- data.frame("Statistic" = c("Theoretical Mean", "Simulation Mean"),
                   "Mean" = c(1/0.2, mean(rmeans)))
print(means)

#comparing variances
hist(rmeans, col = "yellow", xlab = "Number", freq = FALSE, xlim = c(0, 10),
     main = "Simulated Data Density")
x.mean <- seq(2, 8, length.out=100)
y.mean <- dnorm(x.mean, mean(rmeans), sd(rmeans))
lines(x.mean, y.mean, col = "blue", lwd = 2, lty = 2)
abline(v = mean(rmeans) - sd(rmeans), col = "red", lwd = 2, lty = 2)
abline(v = mean(rmeans) + sd(rmeans), col = "red", lwd = 2, lty = 2)
abline(v = 1/0.2 - 1/0.2, col = "green", lwd = 2, lty = 2)
abline(v = 1/0.2 + 1/0.2, col = "green", lwd = 2, lty = 2)
legend("topright", legend = c("Simulation SD", "Theoretical SD",
                             "Density Curve"), cex = 0.75,
      col = c("red", "green", "blue"), lwd = c(1, 2, 2), lty = c(1, 2, 2))
```

```

sds <- data.frame("Statistic" = c("Theoretical Mean", "Simulation Mean"),
                  "Standard Deviation" = c(1/0.2, sd(rmeans)))
print(sds)

print(1/0.2/sqrt(40))

#prove normal distribution
sd1low <- mean(rmeans) - sd(rmeans)
sd2low <- mean(rmeans) - 2*sd(rmeans)
sd3low <- mean(rmeans) - 3*sd(rmeans)
sd1high <- mean(rmeans) + sd(rmeans)
sd2high <- mean(rmeans) + 2*sd(rmeans)
sd3high <- mean(rmeans) + 3*sd(rmeans)
sd1percent <- length(which(rmeans <= sd1high & rmeans >=sd1low))/length(rmeans)
sd2percent <- length(which(rmeans <= sd2high & rmeans >=sd2low))/length(rmeans)
sd3percent <- length(which(rmeans <= sd3high & rmeans >=sd3low))/length(rmeans)
dist <- data.frame("Standard_Deviations" = c(1, 2, 3),
                  "Normal_Distribution" = c(.68, .95, .99),
                  "Simulation_Percents" = c(sd1percent, sd2percent, sd3percent))
print(dist)

```

R Session Info

```

print(sessionInfo())

## R version 3.6.1 (2019-07-05)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18362)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## loaded via a namespace (and not attached):
## [1] compiler_3.6.1  magrittr_1.5    tools_3.6.1    htmltools_0.3.6
## [5] yaml_2.2.0      Rcpp_1.0.2      stringi_1.4.3  rmarkdown_1.15
## [9] knitr_1.24      stringr_1.4.0   xfun_0.9       digest_0.6.20
## [13] evaluate_0.14

```