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 two random exponential distributions. The first distribution is a sample of 1,000 random exponential numbers and the second is 1,000 means of 40 random exponential numbers.

Simulations

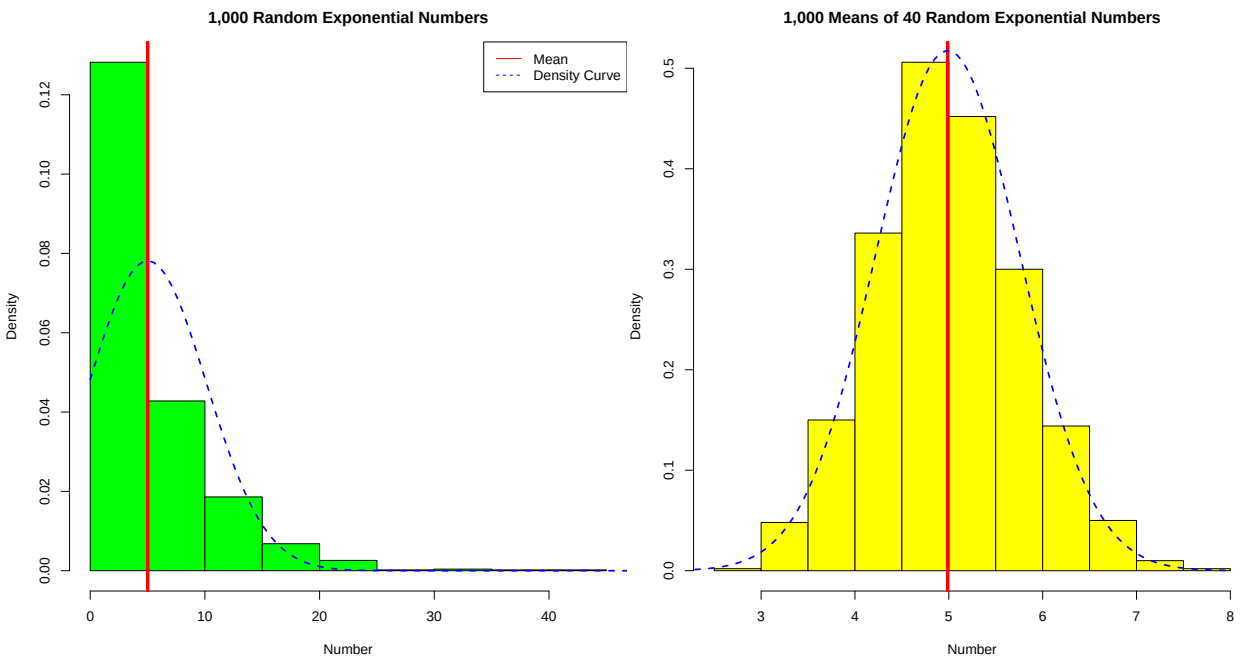
First we need to build two simulations: a sample and a theoretical. The sample simulation will simply be 1,000 random exponential numbers with lambda (the rate) set to 0.2. The theoretical simulation will also be 1,000 numbers, but they will be generated by taking the mean of 40 random exponential numbers (lambda set to 0.2 as well).

```
#simulation for 1000 random exponentials
set.seed(1000)
rnumbers <- rexp(1000, 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)))
}
```

Sample 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 for the random numbers against the random means. Included on the plot is the mean for each set and the density curve.



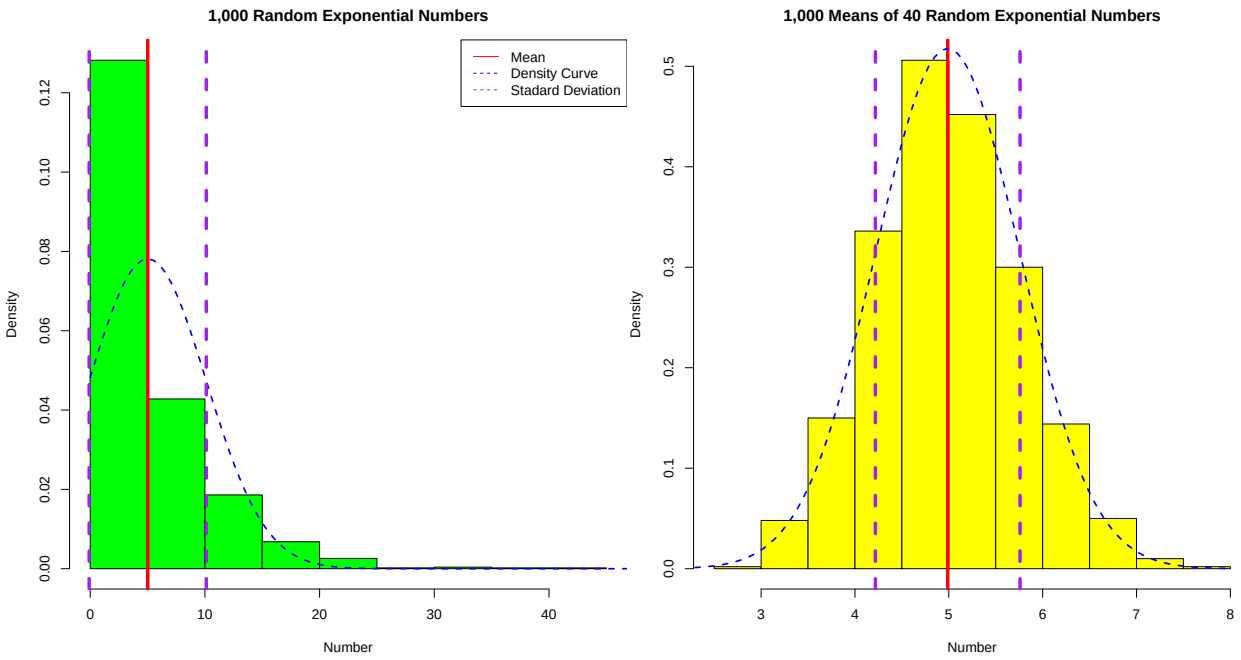
As you can see from the plot, both means are near 5. Even though the distributions look different, it does make sense for the means to be close to one another. The mean of 1,000 means of 40 random numbers is equivalent to the mean of 40,000 numbers, so it shouldn't be surprising to see the means be close to each other.

```
##         Dataset     Mean
## 1 1,000 Numbers 5.015616
## 2   1,000 Means 4.988463
```

This table shows the actual values of the means. Notice how the difference between the two is less than .03. This shows how the two datasets are really similar even if they look different.

Sample Variance vs Theoretical Variance

Now we want to see how the variances are different between the two datasets. For this, we will look at the standard deviation which is the square root of the variance, since it is easier to chart since it is in the same units as the data. We'll look at the same plots as before, only we'll add the lines to show 1 standard deviation above and below the mean for each dataset.



As you can see, there is a huge difference in the standard deviations. For the 1,000 random numbers the standard deviation is about 5, wherease for the 1,000 means of 40, the standard deviation is less than 1. Averaging 40 numbers takes out a lot of variability since "outliers" get averaged in with numbers which are closer to the mean of the entire dataset. This drop in variance also shows why it will be a lot easier to determine whether exponential numbers follow the normal distribution than just 1,000 random exponential numbers.

```
##         Dataset Standard.Deviation
## 1 1,000 Numbers           5.105883
## 2   1,000 Means           0.770832
```

The table above shows the actual values of the standard deviations of the two datasets.

Distribution

Finally we want to prove the central limit theorem applies in for exponential numbers by proving the 1,000 means form a normal distribution. As you can see in the plots above, the 1,000 means do seem to form a bell curve (looking at the density 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

```
#comparing means of simulations
par(mfrow = c(1, 2), mar = c(4, 4, 3, 0))
hist(rnumbers, col = "green", xlab = "Number", freq = FALSE,
     main = "1,000 Random Exponential Numbers")
abline(v = mean(rnumbers), col = "red", lwd = 4)
x.num <- seq(0, 50, length.out=100)
y.num <- dnorm(x.num, mean(rnumbers), sd(rnumbers))
lines(x.num, y.num, col = "blue", lwd = 2, lty = 2)
legend("topright", legend = c("Mean", "Density Curve"), col = c("red", "blue"),
       lty = 1:2)

hist(rmeans, col = "yellow", xlab = "Number", freq = FALSE,
     main = "1,000 Means of 40 Random Exponential Numbers")
abline(v = mean(rmeans), col = "red", lwd = 4)
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)

means <- data.frame("Dataset" = c("1,000 Numbers", "1,000 Means"),
                    "Mean" = c(mean(rnumbers), mean(rmeans)))
print(means)

#comparing variances
par(mfrow = c(1, 2), mar = c(4, 4, 3, 0))
hist(rnumbers, col = "green", xlab = "Number", freq = FALSE,
     main = "1,000 Random Exponential Numbers")
abline(v = mean(rnumbers), col = "red", lwd = 4)
x.num <- seq(0, 50, length.out=100)
y.num <- dnorm(x.num, mean(rnumbers), sd(rnumbers))
lines(x.num, y.num, col = "blue", lwd = 2, lty = 2)
legend("topright", legend = c("Mean", "Density Curve", "Stadard Deviation"),
       col = c("red", "blue", "purple"), lty = c(1, 2, 2))
abline(v = mean(rnumbers) - sd(rnumbers), col = "purple", lwd = 4, lty = 2)
abline(v = mean(rnumbers) + sd(rnumbers), col = "purple", lwd = 4, lty = 2)

hist(rmeans, col = "yellow", xlab = "Number", freq = FALSE,
     main = "1,000 Means of 40 Random Exponential Numbers")
abline(v = mean(rmeans), col = "red", lwd = 4)
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 = "purple", lwd = 4, lty = 2)
abline(v = mean(rmeans) + sd(rmeans), col = "purple", lwd = 4, lty = 2)

sds <- data.frame("Dataset" = c("1,000 Numbers", "1,000 Means"),
                  "Standard Deviation" = c(sd(rnumbers), sd(rmeans)))
print(sds)

#prove normal distribution
sd1low <- mean(rmeans) - sd(rmeans)
```

```r
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

```r
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
```