Simulations with exponential distribution

Xingmin Aaron Zhang

In this project, we will investigate the exponential distribution in R and compare it with the Central Limit Theorem. The exponential distribution can be simulated in R with rexp(n, lambda) where lambda is the rate parameter. The mean of exponential distribution is 1/lambda and the standard deviation is also 1/lambda. Set lambda = 0.2 for all of the simulations. You will investigate the distribution of averages of 40 exponentials. Note that you will need to do a thousand simulations.

Simulations

We randomly sample the exponential distribution (lambda = 0.2) for 40 numbers (n = 40) and repeated do this for 1000 simulations.

```
n = 40
lambda = 0.2
N = 1000
set.seed(31237)
expMatrix <- matrix(rexp(n * N, rate = lambda), nrow = N, byrow = TRUE )
```

We plot the numbers of all samples to make sure the simulation works as expected. The plot shows all values for each row (Figure 1). As expected, all row values have pretty similar patterns.

```
par(mar=c(2,2,1,1))
matplot(1:N, expMatrix, type = "p", pch = 19, cex = 0.1, xlab = "row", ylab = "value")
```

Theoretical means and standard deviations

For exponential distributions, both the mean and standard deviation is 1/lambda. For a sample of size n, the mean is still 1/lamba and the standard deviation of the mean will be 1/(lambda * sqrt(n)).

```
mu_t = 1/lambda
mu_t
```

```
## [1] 5
```

```
sd_t = 1/lambda
sd_t
```

```
## [1] 5
```

Sample means and standard deviations

We calculate the mean and standard deviation of these 1000 samples, and create a histograms to show their distribution(Figure 2). One can see that the means and standard deviations of samples are pretty close to the theoretical values.

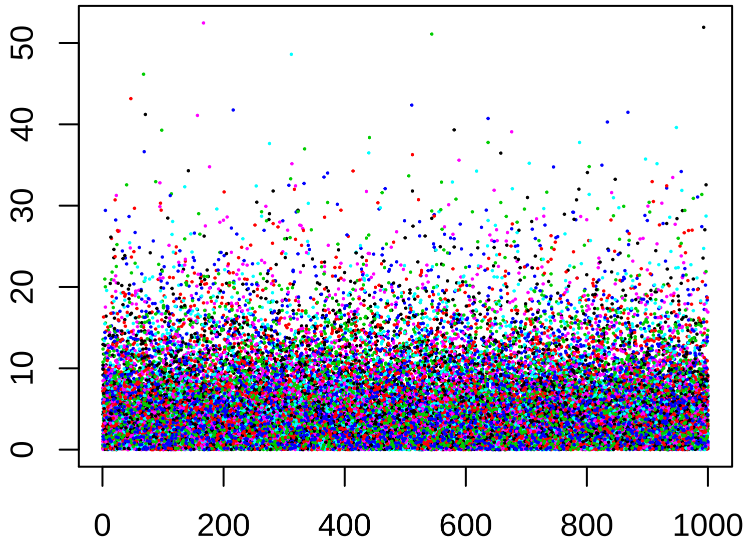

Figure 1: sampled data plotted against row index

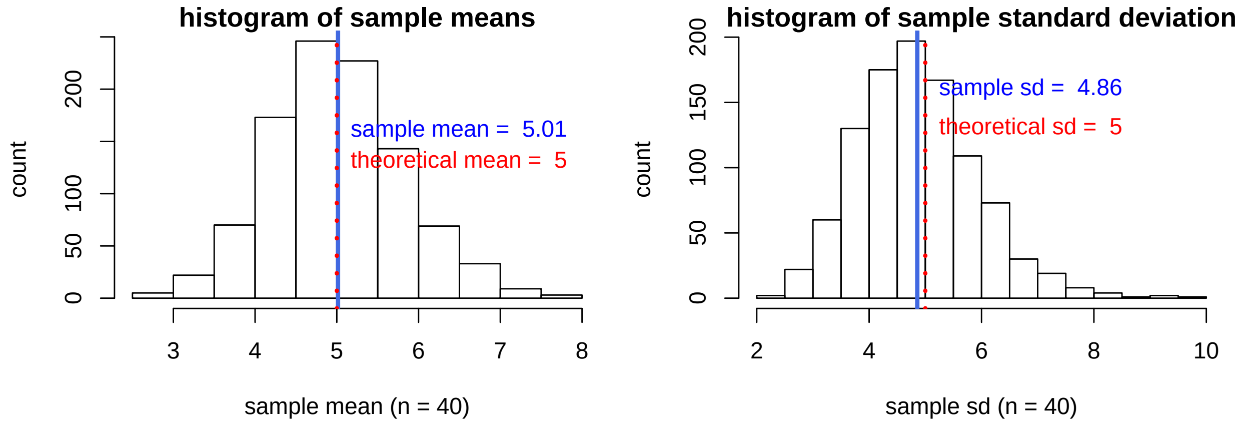

Figure 2: Distribution of means and standard deviations of 1000 samples

```r
par(mfrow = c(1,2), mar=c(4,4,1,1))
expMean <- rowMeans(expMatrix)
{ hist(expMean, xlab = "sample mean (n = 40)", ylab = "count",
      main = "histogram of sample means")
  abline(v = mean(expMean), col = "royalblue", lwd = 3)
  text(x = 5, y = 160, labels = paste("sample mean = ", round(mean(expMean), 2)), pos = 4, col = "blue"
  abline(v = mu_t, col = "red", lwd = 3, lty = 3)
  text(x = 5, y = 130, labels = paste("theoretical mean = ", mu_t), pos = 4, col = "red")}
expSD <- apply(expMatrix, 1, sd)
{ hist(expSD, xlab = "sample sd (n = 40)", ylab = "count",
      main = "histogram of sample standard deviation")
  abline(v = mean(expSD), col = "royalblue", lwd = 3)
  text(x = 5, y = 160, labels = paste("sample sd = ", round(mean(expSD), 2)), pos = 4, col = "blue")
  abline(v = sd_t, col = "red", lwd = 3, lty = 3)
  text(x = 5, y = 130, labels = paste("theoretical sd = ", sd_t), pos = 4, col = "red")}
```

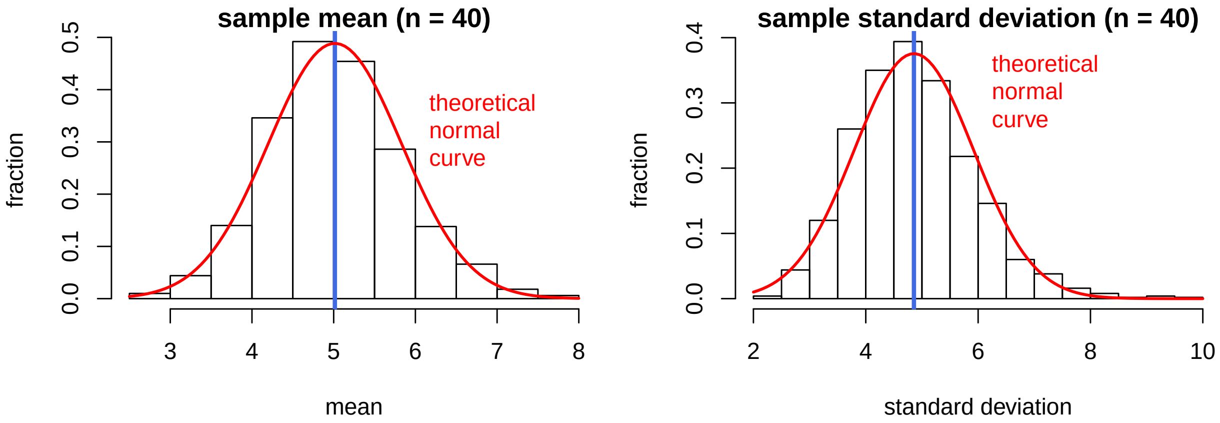

Figure 3: Distribution of sample means (left) and raw values of samples (right)

We can make inference about population mean from the simulations, the 95% confidence interval is be [3.41 ~ 6.62]. Clearly, the theoretical mean (u = 5) is within this interval.

Comparison with normal distributions

We can also fit the above histograms with a normal distribution curve and found that they match very well (Figure 3), which consists with the cental limit theorem.

```r
par(mfrow = c(1, 2), mar=c(4,4,1,1))
{ hist(expMean, xlab = "mean", ylab = "fraction",
      main = "sample mean (n = 40)", freq = FALSE)
  abline(v = mean(expMean), col = "royalblue", lwd = 3)
  text(x = 6, y = 0.3, labels = "theoretical\nnormal\ncurve", pos = 4, col = "red")
  x <- seq(2, 8, by = 0.2)
  curve(dnorm(x, mean = mean(expMean), sd = sd(expMean)),add = TRUE, col = "red", lwd = 2)
}
{ hist(expSD, xlab = "standard deviation", ylab = "fraction",
      main = "sample standard deviation (n = 40)", freq = FALSE)
  abline(v = mean(expSD), col = "royalblue", lwd = 3)
  text(x = 6, y = 0.3, labels = "theoretical\nnormal\ncurve", pos = 4, col = "red")
  x <- seq(2, 8, by = 0.2)
  curve(dnorm(x, mean = mean(expSD), sd = sd(expSD)),add = TRUE, col = "red", lwd = 2)
}
```