Simulation of the exponential distribution and comparison with the theoretical values

Daniel Fabian

June 16, 2015

Overview

We are going to analyse the distribution of averages of 40 samples from an exponentially distributed random variable with $\lambda = 0.2$. We will run 1000 simulations at a time and want to verify whether or not the distribution is approximately normal.

Simulation

First, we define a helper function that will return n samples of averages of k exponentials with parameter λ.

```
rexpmean <- function(n, k, lambda) {
  sapply(1:n, function(i){mean(rexp(k, lambda))})
}
```

this helper function creates a vector of the length n. And in each element of that vector, it puts `mean(rexp(k, lambda))` or with other words, the average of k samples from the exponentially distributed random variable with parameter λ.

To simulate the 1000 means of averages of 40 exponentials with $\lambda = 0.2$, we just need to call

```
samples <- rexpmean(1000, 40, 0.2)
```

Sample mean vs. theoretical mean

The theoretical mean for averages of k random numbers from $X \sim \mathrm{Exp}(\lambda)$ is

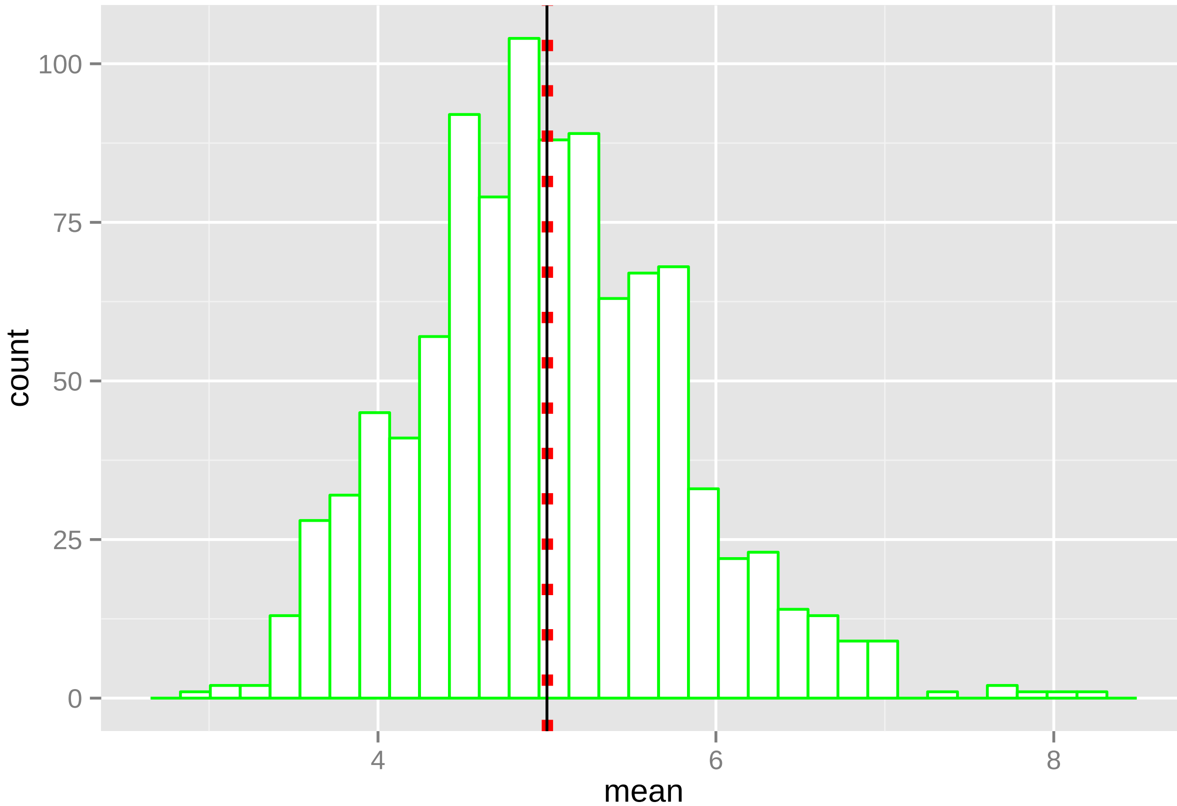
$$\mathrm{E}(\bar{X}_k) = \mu = \frac{1}{\lambda} = 5$$

```
sampleMean <- mean(samples)
sampleMean
```

```
## [1] 5.002534
```

in the histogram, we see the actual mean in a dotted line, and the theoretical mean the full line:

```
library(ggplot2)
g <- ggplot(data.frame(mean=samples), aes(x = mean))
g <- g + geom_histogram(colour="green", fill="white")
g <- g + geom_vline(colour="red", size=2, linetype="dotted", xintercept=sampleMean)
g <- g + geom_vline(xintercept=5)
g
```



as we see, they are very close, which is to be expected with $40 \cdot 1000$ samples.

Sample variance vs. theoretical variance

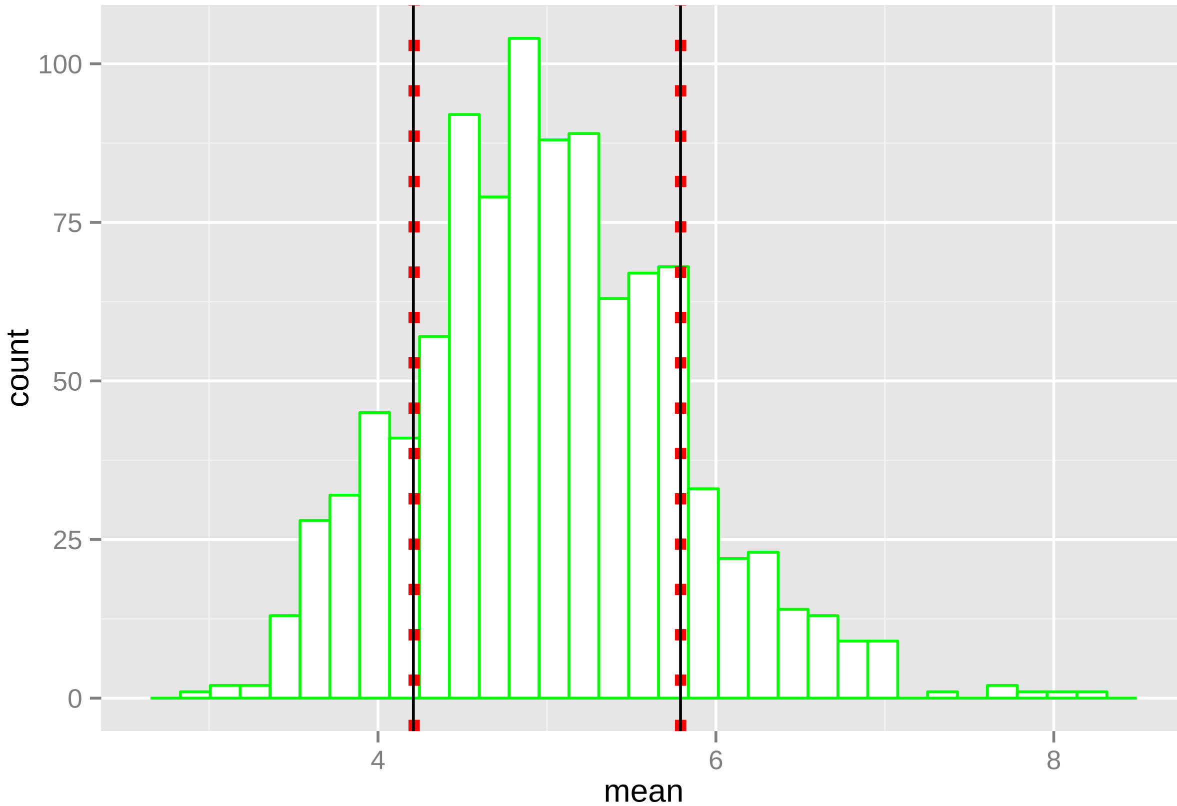
$$\mathrm{Var}(\bar{X}_k) = \frac{\sigma^2}{k} = \frac{\lambda^{-2}}{k} = \frac{25}{40} = \frac{5}{8} = 0.625$$

```
sampleVariance <- var(samples)
sampleVariance
```

```
## [1] 0.6221105
```

in the histogram, we see the actual variance standard deviation as dotted lines, and the theoretical standard deviation as full lines:

```
library(ggplot2)
g <- ggplot(data.frame(mean=samples), aes(x = mean))
g <- g + geom_histogram(colour="green", fill="white")
g <- g + geom_vline(colour="red", size=2, linetype="dotted",
                    xintercept=sampleMean + c(-1, 1)*sqrt(sampleVariance))
g <- g + geom_vline(xintercept=5 + c(-1,1)*sqrt(0.625))
g
```



as we see, they are very close, which is to be expected with $40 \cdot 1000$ samples. They describe the $\mu \pm \sigma$ interval.

Distribution

In order to see, that the distribution looks more and more like a normal distribution, we can plot it for different values of k:

first, we create a data frame with 1000 samples each, for different k values.

```
kValues <- c(1, 2, 3, 5, 8, 13, 21, 34, 40)
samples <- lapply(kValues, function(k) { cbind(rep.int(k, 1000), rexpmean(1000, k, 0.2)) })
samples <- do.call(rbind, samples)
samples <- data.frame(k = samples[, 1], mean = samples[, 2])
```

next, we calculate a density of the normal distribution and one of the Γ distribution for each k. The $\Gamma(k, \lambda)$ distribution is the density of the means of k exponential distributions $\mathrm{Exp}(\lambda)$

```
density <- lapply(kValues, function(k) {
    t(sapply(seq(0, 10, 0.1), function(x) { c(k, x, dnorm(x, 5, sqrt(25 / k)), dgamma(x, k, rate = k*0.
  })
density <- do.call(rbind, density)
density <- data.frame(k = density[, 1], x = density[, 2], norm = density[, 3], gamma = density[, 4])
```

now we can plot a histogram. The red curve is the normal distribution, and we are trying to show, that the histogram looks more and more like it. The dashed blue curve, is the theoretically expected curve for the histogram. And the green histogram is the actually measured density.

```
g <- ggplot(samples, aes(x = mean, y = ..density..))
g <- g + geom_histogram(colour="green", fill="white", binwidth=0.5)
g <- g + geom_line(data = density, aes(x = x, y = norm), colour="red")
g <- g + geom_line(data = density, aes(x = x, y = gamma), colour="blue",
                   linetype="dashed")
g <- g + facet_wrap( ~ k, nrow = 3, scales = "free_y")
g <- g + coord_cartesian(xlim=c(0, 10))
g
```