Central Limit Theorem and the Exponential Distribution

Derwin McGeary

September 18, 2015

Overview

We are going to demonstrate how the central limit theorem applies by using a simulation from the exponential distribution.

Simulations

So we start by setting the parameters and loading the `ggplot2` library. We also set the seed here to make the results fully reproducible. They don't need to be in this case, but it's a good habit.

```
library(ggplot2)
lambda <- 0.2
nosim <- 1000
n <- 40
set.seed(1477)
```

Now we take 1000 averages from 40 samples from `rexp` with $\lambda = 0.2$.

```
averages=NULL
for(i in 1:nosim) averages=c(averages, mean(rexp(n,lambda)))
```

Sample Mean versus Theoretical Mean

Let us take the mean of our samples.

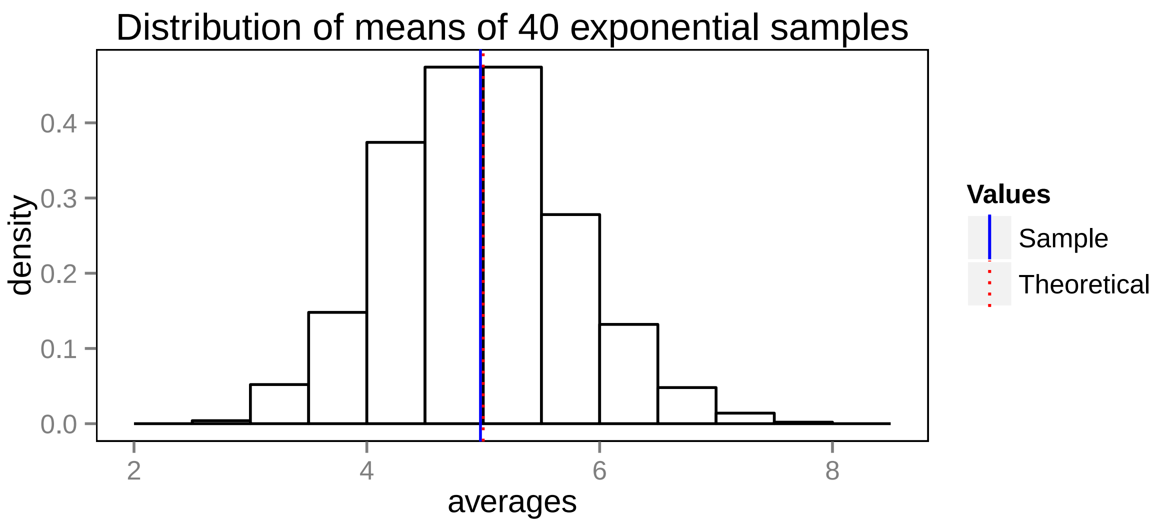
```
avmean <- mean(averages)
emean <- 1/lambda
```

The mean of the averages is **4.976**. The expected mean of the exponential distribution is $\mu = \frac{1}{\lambda} = 5$. That's a difference of 0.48%.

Let's look at that graphically

```
histo <- ggplot(as.data.frame(averages), aes(x=averages)) +
  geom_histogram(binwidth=.5, colour="black", fill="white", aes(y=..density..)) +
  scale_colour_manual(name="Values", values=c(Sample="blue",Theoretical="red")) +
  scale_linetype_manual(name="Values", values=c(Sample="solid",Theoretical="dotted")) +
  theme(legend.background=element_blank(),
        panel.grid.major=element_blank(),
        panel.grid.minor=element_blank(),
        panel.background=element_rect(colour="black", fill="white")) +
  ggtitle("Distribution of means of 40 exponential samples")
```

```r
histo +
  geom_vline(aes(xintercept=mean(averages),
                 color="Sample", linetype="Sample"), show_guide = TRUE) +
  geom_vline(aes(xintercept=1/lambda,
                 colour="Theoretical", linetype="Theoretical"), show_guide = TRUE)
```



We can see that the sample and theoretical means are practically on top of each other.

Sample Variance versus Theoretical Variance

The theoretical variance of the exponential function is λ^{-2} which is 25 in this case. The expected variance *of the sample mean* is $\frac{\lambda^{-2}}{N}$ which is **0.625**. In reality, our samples give us

```r
esampvar <- (lambda^-2)/n # this is the expected sample variance
print(avvar <- var(averages)) # this is our actual sample variance
```

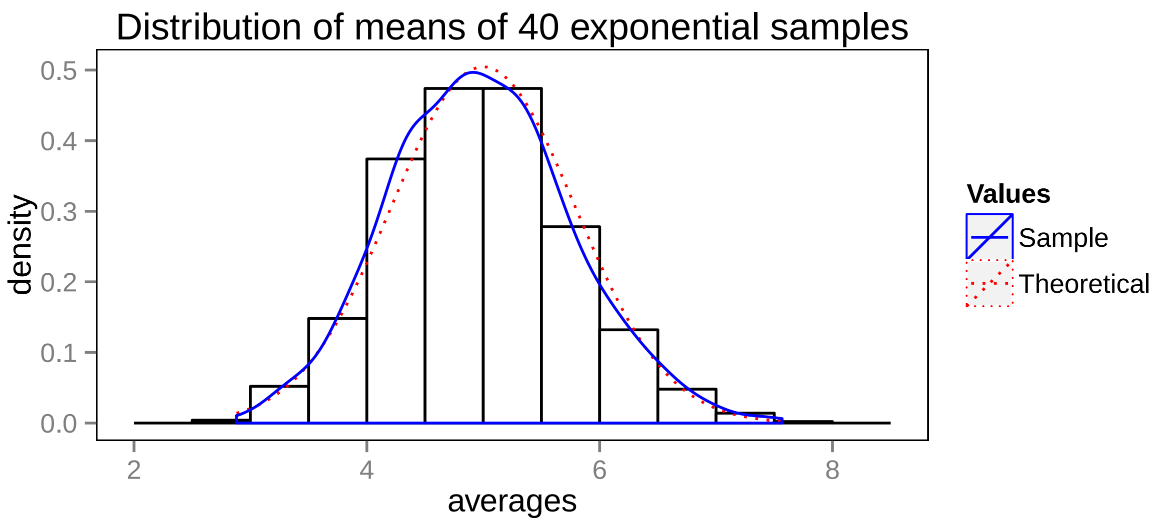
```
## [1] 0.6043701
```

So the variance is **0.604** and the difference from our expected variance is 0.021, or 3.3%.

Distribution

The following chart displays the density function for our samples and superimposes the normal curve.

```r
histo +
  stat_function(fun = dnorm, args=list(mean=1/lambda, sd=sqrt(esampvar)),
                aes(colour="Theoretical", linetype="Theoretical")) +
  geom_density(alpha=.2, aes(colour="Sample",linetype="Sample"))
```



So it can be seen that our distribution is indeed approximately normal.

Appendix

This was done with the following setup, on Ubuntu running on a Samsung Chromebook.

```
sessionInfo()
```

```
## R version 3.2.1 (2015-06-18)
## Platform: arm-unknown-linux-gnueabihf (32-bit)
## Running under: Ubuntu 14.10
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] ggplot2_1.0.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.1      digest_0.6.4     MASS_7.3-33      grid_3.2.1
##  [5] plyr_1.8.1       gtable_0.1.2     formatR_1.2      evaluate_0.7
##  [9] scales_0.2.4     reshape2_1.4     rmarkdown_0.7.3  labeling_0.2
## [13] proto_0.3-10     tools_3.2.1      stringr_0.6.2    munsell_0.4.2
## [17] yaml_2.1.13      colorspace_1.2-4 htmltools_0.2.6  knitr_1.10.5
```