Statistical Inference: Exponential Distribution CLT Simulation

Marcio Gualtieri

Contents

Installing the Required Packages

You might need to install the following packages if you don't already have them:

```
install.packages("ggplot2")
install.packages("gridExtra")
```

Importing the Required Packages

Once the libraries are installed, they need to be loaded as follows:

```
suppressMessages(library(xtable))                          # Pretty printing dataframes
suppressMessages(library(ggplot2))                         # Plotting
suppressMessages(library(gridExtra, warn.conflicts = FALSE))
```

Exponential Distribution CLT Simulation

Overview

The CLT (Central Limit Theorem) states that if independent random variables are linearly combined (for instance, their sum or mean), the result (which is also a random variable) has a distribution which will resemble more and more closely a normal distribution as the number of variables increases. Note that no assumption is made about the original distribution of the variables combined, which does not matter.

The objective of this experiment is to show that the CTL also applies to random variables that follow an exponential distribution.

Theory

The Exponential Distribution

Exponential distributions describe Poison processes, i.e., processes that consist of counting events that occur continuously and independently at a constant average rate. One of such processes would be the distribution of the number of clicks on a given online ad over the period from 10:00 PM to 11:00 PM for instance.

The function below creates data using the probability exponential distribution's density function for a given range of values of `x` and `lambda`:

```
create_dexp_data <- function(xs, lambdas) {
  x_data <- rep(xs, length(lambdas))
  lambda_data <- rep(lambdas, length(xs))
  data.frame(x = x_data,
             lambda = as.factor(lambda_data),
             density = mapply(dexp, x_data, lambda_data))
}
```

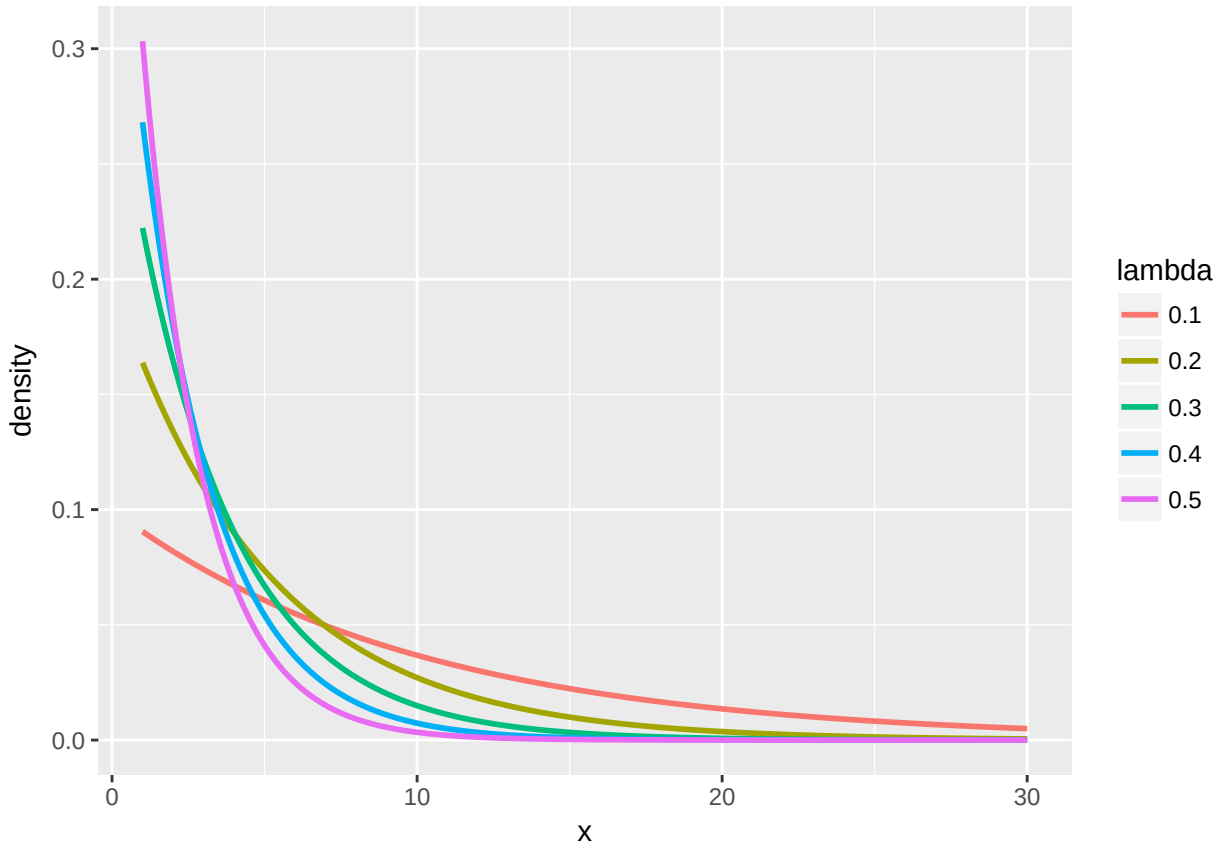
I'm using `dexp()` to compute the density. You will find a summary of the functions available in R related to the exponential distribution here.

The following function creates line plots for the data:

```
categorical_line_plot <- function(data, x_column, y_column, group_column)
ggplot(data=data, aes_string(x = x_column, y = y_column, color = group_column)) +
geom_line(size=1)
```

Here's the plot for different values of lambda:

```
categorical_line_plot(create_dexp_data(seq(1, 30, 0.01), seq(0.1 ,0.5, 0.1)), "x", "density", "lambda")
```



Nothing very special here: These are negative exponential functions, whose shape you might already be familiar with (through physics, for instance, the radiactive decay of unstable atomic nucleous).

Distribution's Theoretical Center and Variance

From probability theory we expect this distribution to possess the following properties:

Property	Value
Density	$f(x) = \lambda \times e^{-\lambda \times x}$
μ	$\frac{1}{lambda}$
σ	$\frac{1}{lambda}$

Experiment

Experiment's Configuration

We are going to perform one thousand simulations, each one consisting of drawing a sample of size 40 from the exponential distribution:

```
number_of_experiments <- 1000
sample_size <- 40
```

Each sample consists of values drawn from independent 40 random variables, each one following an exponential distribution.

Once we drawn each sample (1000 of them), we are going to take the mean over each one of them and

analyze the its distribution, which should resemble a normal one, as you might remember from what we have discussed about the CLT in the overview section.

Distribution's Predicted Center and Variance

We are going to set lambda to 0.2:

```
lambda = 0.2
```

Which results in $\mu = 5$ and $\sigma = 5$:

```
mu <- 1 / lambda
sigma <- 1 / lambda
```

Simulation

The code below performs the simulations:

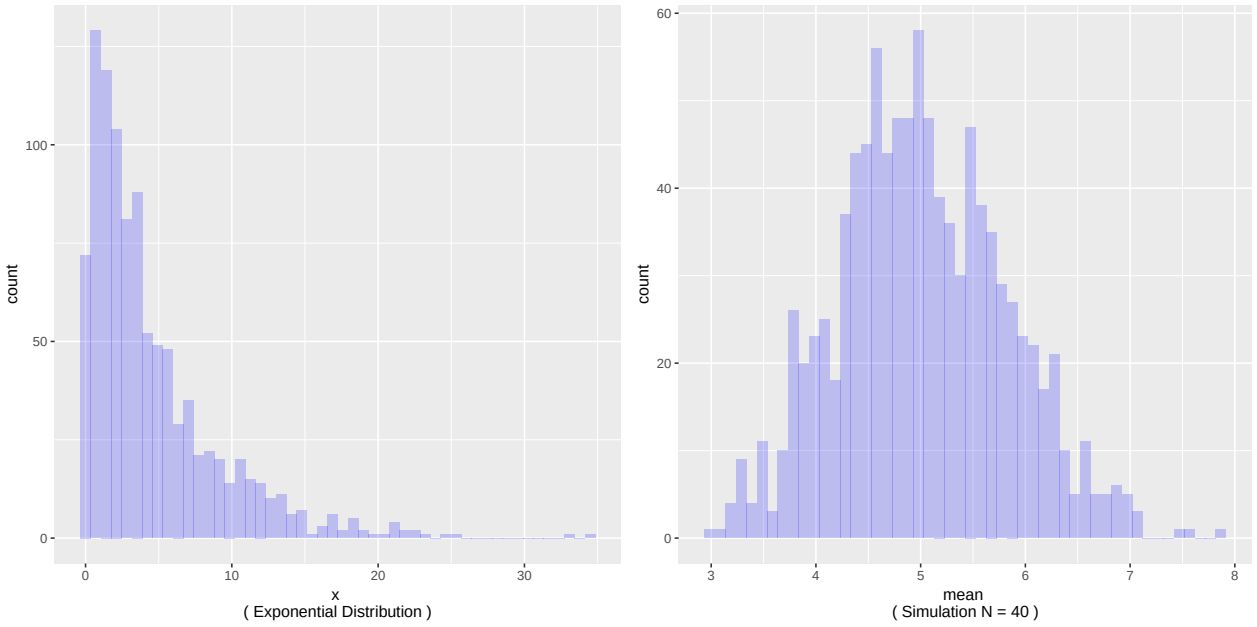
```
simulations <- matrix(rexp(sample_size * number_of_experiments, lambda), number_of_experiments, sample_s
simulations_means <- apply(simulations, 1, mean)
```

Here's a histogram for the simulations' means:

```
histogram <- function(data, column, bins, title) {
  ggplot(data=data, aes_string(x = column)) +
    geom_histogram(bins = bins, na.rm = TRUE, fill = "blue", alpha = 0.2) +
    xlab(paste(column, "\n(", title, ")"))
}

histogram_exponential_distribution <- histogram(data.frame(x = rexp(number_of_experiments, lambda)), "x
histogram_simulation_N_40 <- histogram(data.frame(mean = simulations_means), "mean", 50, "Simulation N =

grid.arrange(histogram_exponential_distribution, histogram_simulation_N_40, nrow=1)
```



Note that the exponential distribution histogram's shape doesn't resemble a normal distribution in any way (but a exponential function, as expected).

The simulation's histogram on the other hand shows the familiar "bell shape" from normal distributions. You might have also noticed that the mean seems to be roughly centered on 5 on the plot.

Comparing Predicted and Experimental Results

Let's calculate this data's mean and standard error:

```
estimate_mean <- mean(simulations_means)
estimate_sigma <- sd(simulations_means)
```

Property	Value	Comment
$\mu_{estimate}$	5.0146601	Close to $\mu = 5$, the prediction for the mean.
$\sigma_{estimate}$	0.8013103	Close to $\frac{\sigma}{sqrt(N)} = 0.7905694$, the prediction for the standard error

Where: $N = 40$ (sample size) and $\sigma = 5$. Our experiment's results are consistent with the predictions from the theory.

Graphically Witnessing the CLT in Action

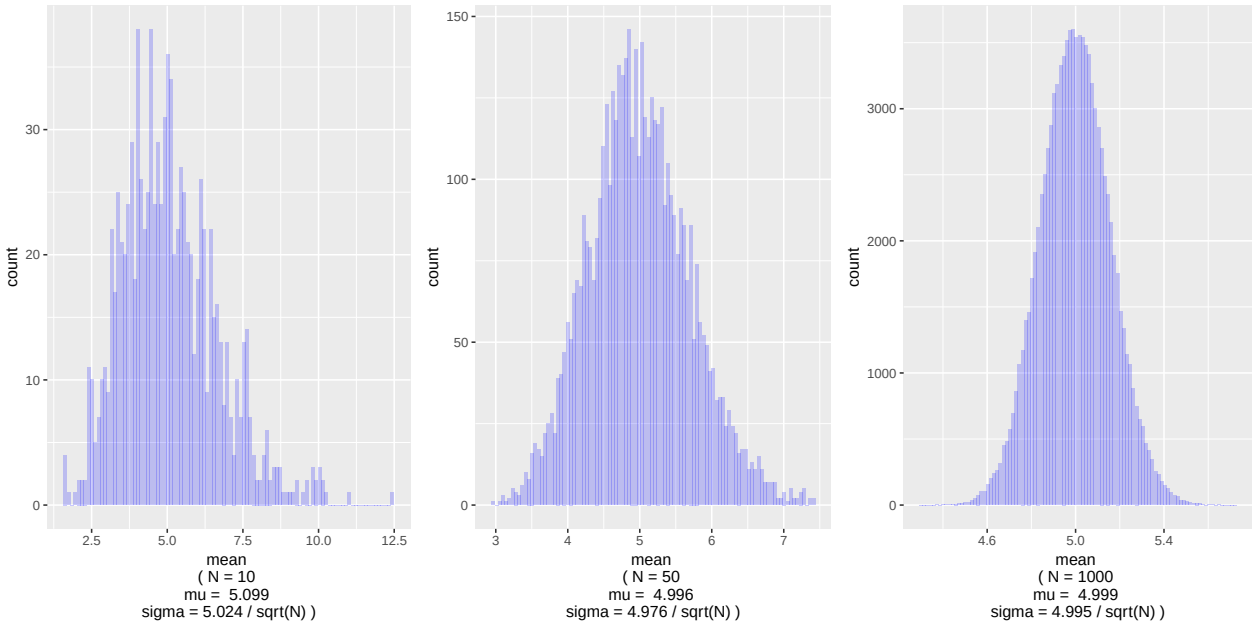
Here I created a function to perform all steps for the experiment we've just performed for $N = 40$, including plotting its respective histogram:

```
experiment_histogram <- function(sample_size, number_of_experiments, bins) {
simulations <- matrix(rexp(sample_size * number_of_experiments, lambda), number_of_experiments, sample_s
simulations_means <- apply(simulations, 1, mean)
mu = mean(simulations_means)
sigma = sd(simulations_means) * sqrt(sample_size)
title <- paste("N =", sample_size, "\nmu = ", round(mu, 3), "\nsigma =", round(sigma, 3), "/ sqrt(N)")
histogram(data.frame(mean = simulations_means), "mean", bins, title)
}
```

Let's get a few histograms for different sample sizes:

```
histogram_N_10 <- experiment_histogram(10, 1000, 100)
histogram_N_50 <- experiment_histogram(50, 5000, 100)
histogram_N_1000 <- experiment_histogram(1000, 100000, 100)

grid.arrange(histogram_N_10, histogram_N_50, histogram_N_1000, ncol=3, nrow=1)
```



Note that the histogram's shape resembles more closely the "bell shape" from the normal distribution as the size of the sample increases and that μ and σ also get closer and closer to the values predicted by theory.