A simulation exercise

Sanja Stegerer

Tuesday, November 17, 2015

A note in the beginning:
As stated in the introduction for the assignment, each report should not be longer than 3 pages + 3 pages of figures and code. I therefore went for the goal to not exceed 6 pages including all figures and code.

Introduction

This document demonstrates the validity and limitations of the Central Limit Theorem. I will run a simulation of a sample draw from the population of averages of 40 exponentially distributed random variables. The Central Limit Theorem provides us with a population distribution for the simulated sample. In order to compare the sample distribution with the theoretical population distribution I will compare the mean and variance of each of the distributions seperately.

Lets start with the properties of the exponential distribuion. We know it has the properties

$$E[X] = \frac{1}{\lambda}, \quad Var(X) = \frac{1}{\lambda^2} \quad \Rightarrow \sigma = \frac{1}{\sqrt{\lambda}}.$$

For the following simulation and comparison let $\lambda = 0.2$.

Calculate the mean and variance of the distribution of 40 exponentials

Save the mean and standard deviation of the exponential distribution in variables for later use.

```
lambda = 0.2; E_exp = 1 / lambda; sd_exp = 1 / lambda
```

These have the values E_exp = 5 and sd_exp = 5.

Let $\bar{X} = \frac{\sum_{i=1}^{40} X_i}{40}$ be a random variable of the distribution of 40 exponentials, with $X_i \sim Exp(\lambda)$, $i \in \{1, \ldots, 40\}$. And then calculate the standard deviation, variance and expected value for the distribution of averages of n exponentials with the Central Limit Theorem, which provides the following formulas

$$E[\bar{X}] = \frac{1}{\lambda}, \quad \sqrt{Var(\bar{X})} = \sigma = \frac{\frac{1}{\lambda^2}}{n} = \frac{1}{\lambda^2 \cdot n}$$

Apply it to our distribution of averages of 40 exponentials and save $E[\bar{X}], Var(\bar{X})$ and $\sigma = \sqrt{Var(\bar{X})}$ in R

```
E40 <- E_exp; sd40 <- 1 / (lambda^2*40); var40 <- sd40^2
```

which results in E40 = 5, var40 = 0.391 and sd40 = 0.625.

Simulate a sample draw from the population of the mean of 40 averages.

Set the specifics to create a reproducible random sample from the exponential distribution in R. This includes the number of simulations. Also initiate the vector which will later contain the results of the simulation.

```
set.seed(1); nosim = 1000; average_exp = NULL
```

Run the simulation by appending the average of 40 (reproducibly) randomly selected values of the exponential distribution and save the data into a dataframe. Additionally create the variable mean_dat which holds the mean of the simulated sample draw from the distribution of the average of 40 exponenetials.

```
for (i in 1:1000) average_exp = c(average_exp, mean(rexp(40,0.2)))
dat = data.frame(average_exp = average_exp)
mean_dat = mean(dat$average_exp)
```

See if the distribution is normally distributed

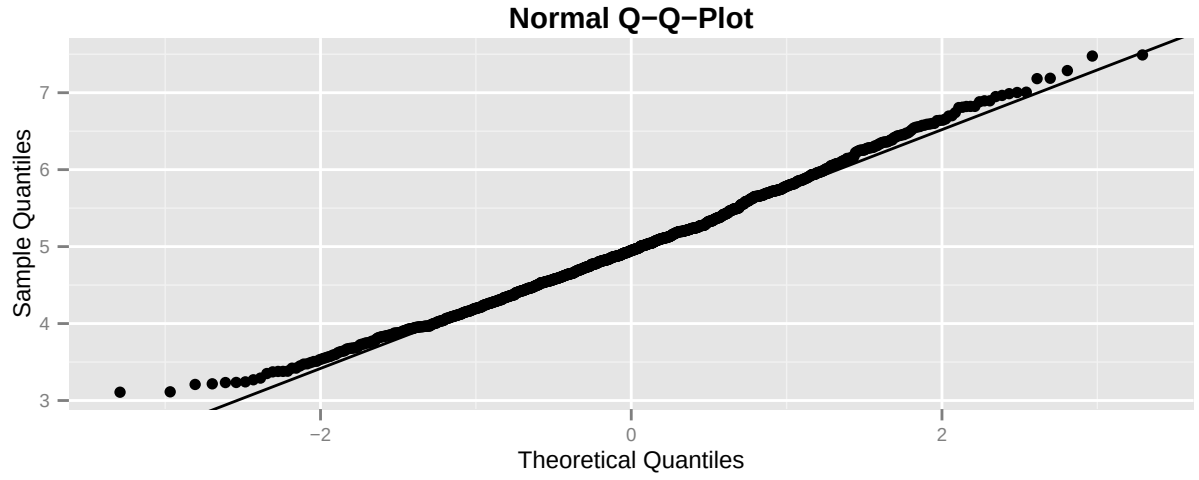
I want to start with the normality test. I decided to put this test up front, because I find it essential to know which of the known distributions can be used as an approximation for the sample distribution. Afterwards I can use this knowledge for the further discussion about the differences in their variance and mean.
To test if a set of data fits a specific underlying distribution, one can use the the so called Quantile-Quantile-Plot. It plots the quantiles of the dataset against the quantiles of the distribution with which we want to estimate our data. If the points can be fitted by a 45 degree line, we know the distribution fits the data quite well, the more the dots deviate from that line, the lesser the fit becomes.
For background information on the Q-Q plot please see here and here. To create a Quantile-Quantile Plot in ggplot2, including the qqline, which is not included in ggplot2, I used the code from this link to create the qqline

First, load the packages ggplot2 to use the ggplot plotting system.

```
library(ggplot2);library(gridExtra);library(grid)
```

```
vec <- average_exp
y <- quantile(vec, c(0.25, 0.75)); x <- qnorm(c(0.25, 0.75))
slope <- diff(y)/diff(x); int <- y[1L] - slope * x[1L]

d <- data.frame(resids = vec)
ggplot(d, aes(sample = resids)) + stat_qq() + geom_abline(slope = slope, intercept = int) +
    ylab("Sample Quantiles") + xlab("Theoretical Quantiles") +ggtitle("Normal Q-Q-Plot") +
    theme(text = element_text(size=9), plot.title=element_text(face='bold'))
```



The points are for most part on a straight line. They diverge up on both tails, which indicate, that the distribution has slightly thicker tails than the normal distribution, but this divergence is that minor, that we can savely assume a normal distribution for the simulation data.

Compare the sample mean with the theoretical distribution mean

Create a basis for the sample and population distribution

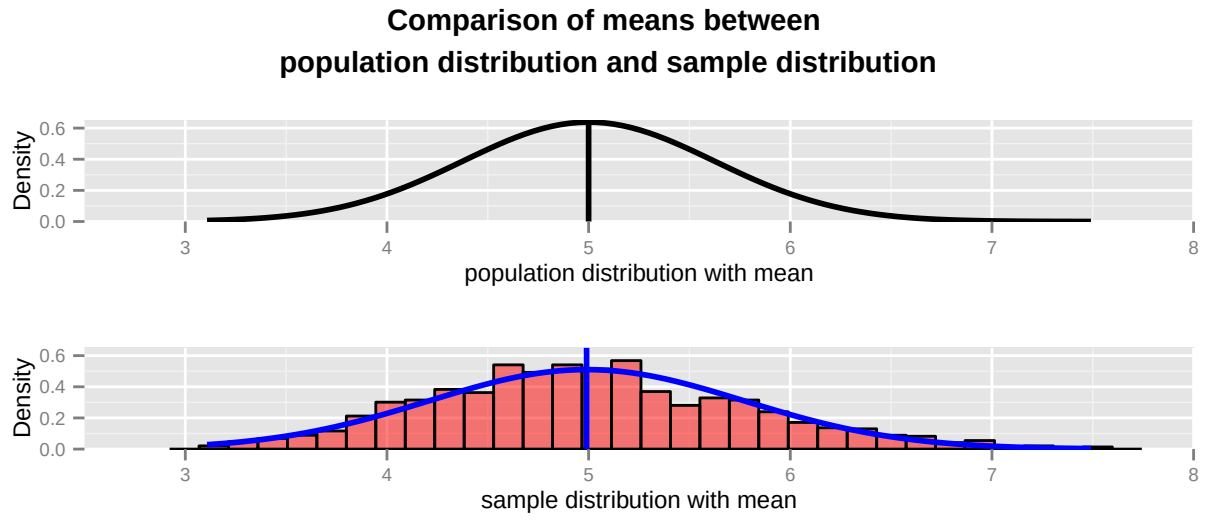
```
g = ggplot(dat, aes(x=average_exp)) + coord_cartesian(ylim = c(0, 0.65), xlim = c(2.5,8)) +
    theme(text = element_text(size=9))
```

As a final step for comparison I create two plots with an overlying normal distribution. The first will be with the calculated expected value and standard deviation E40 and sd40 from the population distribution using the Central Limit Theorem

```
g1 = g + stat_function(fun = dnorm, size = 1, arg = list(sd=sd40, mean=E40)) +
    geom_vline(aes(xintercept=E40),colour='black',size=1) +
    xlab('population distribution with mean') +
    ylab('Density')
```

Create a discrete probability density distribution of the sample draw. We use a disrcirete probability density distribution instead of a frequency distribution (link to a great article about their difference and usage). Create a normal distribution from the mean and standard deviation form the sample and plot them on top of each other.

```
g2 = g + geom_histogram(colour='black', fill='red',aes(y = ..density..),
                    alpha=0.5,bin=(max(average_exp)-min(average_exp))/30) +
    stat_function(fun = dnorm, size = 1,
                  arg = list(sd=sd(dat$average_exp), mean=mean(dat$average_exp)),
                  color='blue') +
    geom_vline(aes(xintercept=mean(average_exp)),colour='blue',size=1) +
    xlab('sample distribution with mean') + ylab('Density')
grid.arrange(g1, g2, nrow=2,
             top=textGrob('Comparison of means between
population distribution and sample distribution',
                          gp=gpar(fontsize=11,fontface='bold')))
```



Just from looking at it, we can already tell that the sample mean and the population mean are very close to each other. To back up this observation, lets take a look at their values:
CLT mean = 5, Sample mean = 4.99

Compare the sample variance with the theoretical distribution variance

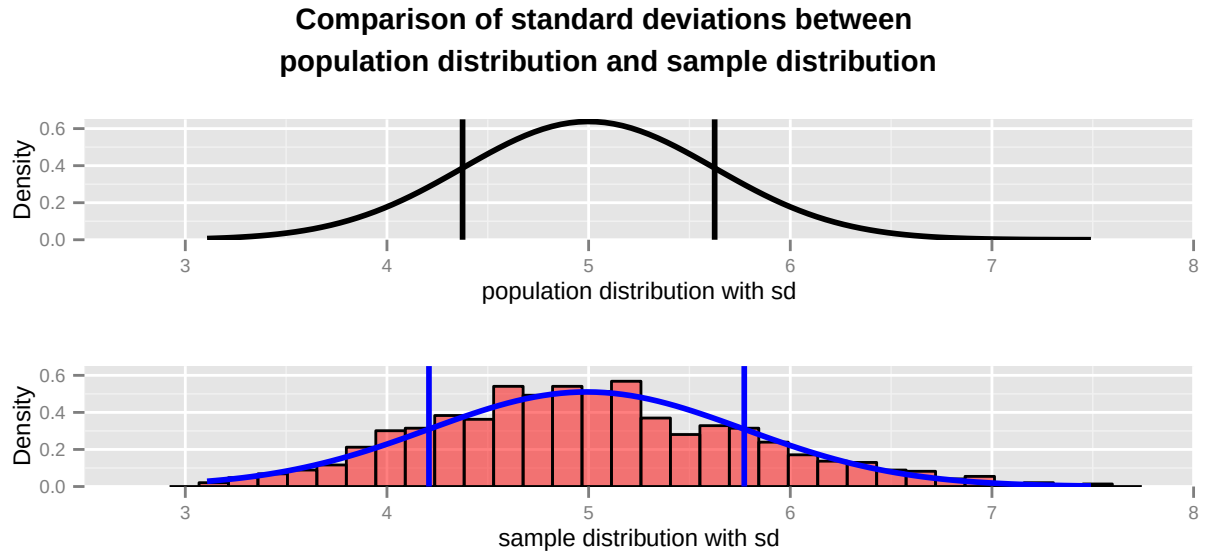
Conducting the same procedure with the sd as a measure of the variance, only exchange

```
    geom_vline(aes(xintercept=MEAN),colour='blue',size=1)
```

with the two lines

```
    geom_vline(aes(xintercept=MEAN-SD),colour='blue',size=1) +
    geom_vline(aes(xintercept=MEAN+SD),colour='blue',size=1)
```

in the plots g1 and g2 and replace the MEAN and SD respectively, yields



These plots clearly communicate the difference between the sample and the population. To back up this result with numbers, we get

Sample standard deviation = 0.78 ⇒ Sample variance = 0.61,
Population standard deviation = 0.62 ⇒ Population variance = 0.39. which yields in a quite huge difference between the two variances.