Gian Balsamo

September 23, 2015

STATISTICAL INFERENCE COURSE PROJECT - PART ONE

The Exponential Distribution in the Light of the Central Limit Theorem

Overview

This study is aimed at demonstrating, via inductive simulation, the correctness of the Central Limit Theorem. According to this theorem, the distribution of averages of independent and identically distributed samples of observations becomes that of a standard normal as the number of observations (i.e., the size of the sample) increases. The mean of a random distribution of this kind tends to its virtual distribution's mean and the variance tends to its virtual distribution's variance divided by the size of the individual samples.
In short and above all, the CLT states that the mean of iid variables tends to its expected value.
The present study verifies this principle vis-a-vis an exponential distribution with lambda = 0.2, whose expected mean and standard deviation are therefore both equal to 5. According to the CLT, we ought to expect thereby that the mean of a suitably large number of iid samples, say 1000, each sample counting a suitably large number of observations,say 40, drawn from our exponential distribution, tends to its expected value, the variance of its dispersion tending to its expected variance divided by 40.

Data Sets

Our first step consists in generating a suitable collection of observations to be used in Part 1 and Part 2 of the present exercise. This first dataset consists of the means of 1000 random samples of 40 values from our exponential distribution. The number of iid variables is 1000, and each one is based on samples of size 40. Our second dataset is to be used in Part2; it consists of a 10,000 x 2dataframe, whose first column includes our first dataset followed by 1000, 4000, and 10000 random exponentials with lambda=0.2, and whose second column includes the factor numbers 1 though 4, each one indexing one of these four random sequences.

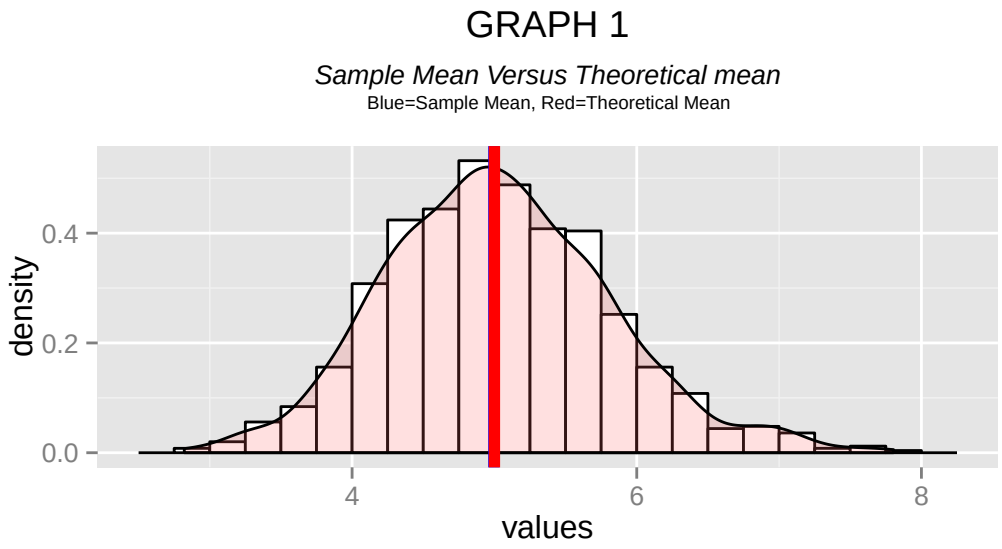
```
library(knitr)
library(ggplot2)
opts_chunk$set(echo=TRUE, results='asis',fig.align='center',dev='pdf')
```

```
#library(ggplot2)
myData<-as.data.frame(replicate(1000,mean(rexp(40,0.2))))
colnames(myData)<-"values"
virtualMean<-5
virtualSd<-5
normalMean<-virtualMean
normalSd<-virtualSd/sqrt(40)
myMean<-mean(replicate(1000,mean(rexp(40,0.2))))
mySd<-sd(replicate(1000,mean(rexp(40,0.2))))
twoMeans<-as.data.frame(c(myMean,normalMean))
colnames(twoMeans)<-"means"
twoSds<-as.data.frame(c(mySd,normalSd))
colnames(twoSds)<-"standard_deviations"
myData2<-data.frame(x=c(replicate(1000,mean(rexp(40,0.2))), rexp(1000,.2),rexp(4000,.2),rexp(10000,.2))
y=as.factor(rep(1:4, c(1000,1000,4000,10000))))
```

Part 1

Graph 1 shows how close the sample mean (blue line)is to the theoretical mean (red line).
The sample mean is equal to 4.9963007.
The theoretical mean is equal to 5.
A transparent pink density curve overlays the 1000 observations' histogram. For ease of visualization, the y-axis is scaled according to percentual densities rather than to frequency counts.
It should be noted that the the blue line of the sample mean crosses the apex of the density curve (it may be partially overshadowd by the red line, though), while it does not necessarly cross (depending of course on the random nature of the dataset under study) the middle top of the histogram's highest bar.

```
ggplot(myData, aes(x=values)) +
geom_histogram(aes(y=..density..),
binwidth=.25,
colour="black", fill="white") +
geom_density(alpha=.2, fill="#FF6666") +
geom_vline(xintercept=twoMeans[1,1],size=2,color="blue")+
geom_vline(xintercept=twoMeans[2,1],size=2, color="red")+
ggtitle(expression(atop("GRAPH 1", atop(italic("Sample Mean Versus Theoretical mean"),atop("Blue=Sample
```

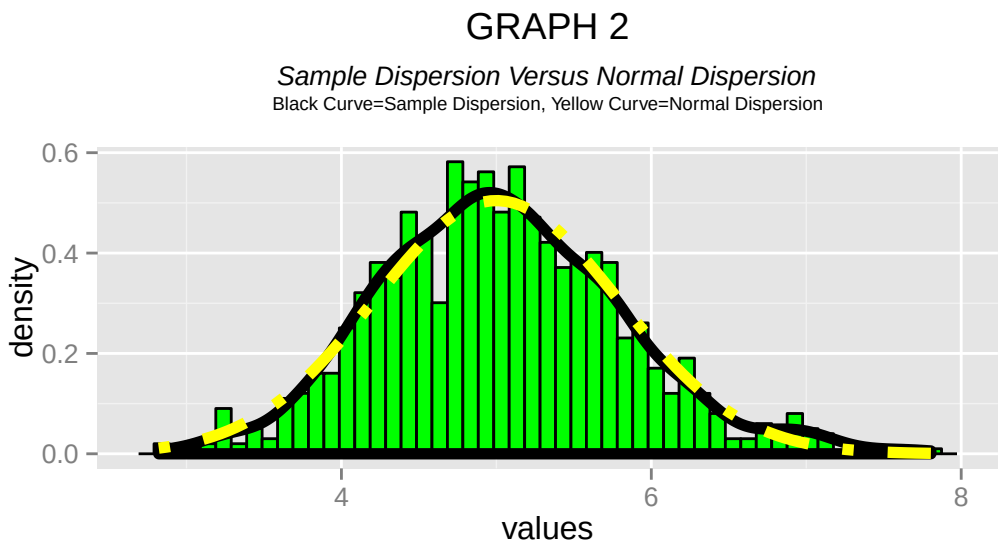


Part 2

Graph 2 shows how close the sample mean's dispersion is to the dispersion of a standard normal curve with the same theoretical mean and theoretical variance, and ranging over the same support.
It wouldn't be difficult to show that the sample mean's dispersion grows thinner with respect to the standard normal as the individual samples become larger in size.
The standard deviation of our dataset is 0.7657613, while the standard deviation of the standard normal curve is 0.7905694.

```
ggplot(myData, aes(x=values)) +
geom_histogram(aes(y=..density..),
binwidth=(max(myData[,1])-min(myData[,1]))/50,
colour="black", fill="green") +
geom_density(aes(x=values), size=2,stat = "density")+
stat_function(fun = dnorm, args = list(mean = normalMean, sd = normalSd),
```

```
colour = "yellow",linetype="dotdash",size=2)+
ggtitle(expression(atop("GRAPH 2", atop(italic("Sample Dispersion Versus Normal Dispersion"),
atop("Black Curve=Sample Dispersion, Yellow Curve=Normal Dispersion"),""))))
```



Part 3

Graph 3 shows - again, but in a different way - that the distribution of the means of 1000 random samples of 40 values from our exponential distribution is approximately normal. To show this, Graph 3 shows that not only does their distribution look normal, but it is radically different from the distribution of, respectively, 1000, 4000, and 10000 random exponentials with lambda=0.2. The sample mean distribution belongs indeed to a distinct, quasi-normal distribution, and not to an exponential distribution. Its apex is close tothe normal's expected mean, while the apex of the three exponential distributions is close to the lambda value. In Graph 3, these three exponential distributions are identified by the colors green, blue and magenta respectively; red is the sample mean's color. For the sake of visual clarity, the graph shows only a partial range of the distributions' x-axis (which explains the following list of "warnings").

```
ggplot(myData2, aes(x=x,color=y)) +
geom_density()+
xlim(0,20)+
ggtitle(expression(atop("GRAPH 3",
atop(italic("Sample Dispersion Versus Three Exponential
Dispersions"),""))))
```

```
## Warning: Removed 16 rows containing non-finite values (stat_density).

## Warning: Removed 82 rows containing non-finite values (stat_density).

## Warning: Removed 167 rows containing non-finite values (stat_density).

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa
```

```
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font metrics unknown for character 0xa

## Warning in grid.Call.graphics(L_text, as.graphicsAnnot(x$label), x$x, x
## $y, : font metrics unknown for character 0xa

## Warning in grid.Call.graphics(L_text, as.graphicsAnnot(x$label), x$x, x
## $y, : font metrics unknown for character 0xa

## Warning in grid.Call.graphics(L_text, as.graphicsAnnot(x$label), x$x, x
## $y, : font metrics unknown for character 0xa
```

```
## Warning in grid.Call.graphics(L_text, as.graphicsAnnot(x$label), x$x, x
## $y, : font metrics unknown for character 0xa

## Warning in grid.Call.graphics(L_text, as.graphicsAnnot(x$label), x$x, x
## $y, : font metrics unknown for character 0xa

## Warning in grid.Call.graphics(L_text, as.graphicsAnnot(x$label), x$x, x
## $y, : font metrics unknown for character 0xa

## Warning in grid.Call.graphics(L_text, as.graphicsAnnot(x$label), x$x, x
## $y, : font metrics unknown for character 0xa

## Warning in grid.Call.graphics(L_text, as.graphicsAnnot(x$label), x$x, x
## $y, : font metrics unknown for character 0xa
```

GRAPH 3

Sample Dispersion Versus Three Exponential Dispersions