Appendix of Exponential Distribution and Central Limit Theorem

CJ

Thursday, January 08, 2015

Contents

Simulation

Firstly, calculate the theoretical values of exponential mean and standard deviation. When `lambda`= 0.2, both mean and standard deviation equal to `5`.

```
lambda <-0.2

# therotical values of sd and mean
tsd <- 1/lambda; tmn <- 1/lambda

tmn;tsd
```

Function `exps` generate a vector of size `n` contains transformation of `m` exponentials. For example, here we use function `mean`, which returns a vector of means of `m` exponetials.

```
lambda <-0.2

n<-1000 # Number of simulations simulations

m<-40 # Number of  exponentials.

exps<- function(ne,ns,lambda,seed, FUN=mean){
        set.seed(seed)
        x=NULL
        for (i in 1:ns) x=c(x, FUN(rexp(ne,lambda)))
        data.frame(x)
}
mns <- exps(40,1000,lambda, 1000, mean)
vars <- exps(40,1000,lambda,1000, FUN=var)
```

Sample Mean versus Theoretical Mean

Plot the histogram of a thousand of simulated means

```
library(ggplot2)

# average of means
smn <- mean(mns$x); smn

# standard error
```

```r
sse <- sd(mns$x)/sqrt(n); sse

# plot histogram
g1 <- ggplot(mns, aes(x=x))

myhist<- function(g, bw,title) {
        hist <- g+ geom_histogram(binwidth=bw, colour="black", fill="white")+
                ggtitle(title)+
                theme(plot.title = element_text(lineheight=.8, face="bold"))+
                xlab("")+
                ylab("Count")
        hist
}

hist1 <- myhist(g1, 0.1, title="Fig1: Sample Mean versus Theoretical Mean")

# add annotation
notex <- 6; notey <- 70
hist1+ geom_vline(aes(xintercept=smn),color="red", linetype="dashed", size=1)+
        annotate("text", x = notex, y = notey,
                 label = paste("sample mean:", as.character(round(smn,2))), color="red")+
        geom_vline(aes(xintercept=tmn),color="blue", linetype="dashed", size=1)+
        annotate("text", x = notex, y = notey+5,
                 label = paste("theoretical mean:",  as.character(round(tmn,2))), color="blue")
```

As we can see from `Fig1`, the center of sample means is very close to the theoretical mean with a standard error 0.02558013.

Sample Variance versus Theoretical Variance

Plot the histogram of a thousand of simulated variances

```r
library(ggplot2)

# average of variances
mv <- mean(vars$x);mv

# standard error
sse <- sd(vars$x)/sqrt(n); sse

# plot histogram
g2 <- ggplot(vars, aes(x=x))

hist2 <- myhist(g2, 1, title="Fig2: Sample Variance versus Theoretical Variance")

# add annotation

notex <- 50; notey <- 45

hist2+ geom_vline(aes(xintercept=mv),color="red", linetype="dashed", size=1)+
        annotate("text", x = notex, y = notey,
                 label = paste("sample variance:", as.character(round(mv,2))), color="red")+
```

```
        geom_vline(aes(xintercept=tsd^2),color="blue", linetype="dashed", size=1)+
        annotate("text", x = notex, y = notey+5,
                 label = paste("theoretical variance:",  as.character(round(tsd^2,2))), color="blue")
```

Nomarlization

Here we normalize both sample mean and variance and compare with the standard normal distribution.

```
normalize <- function(x, xbar, ste){
        (x-xbar)/ste
}

Nmns <- as.data.frame(normalize(mns$x, mean(mns$x), sd(mns$x))); names(Nmns) <- "x"
Nv <- as.data.frame(normalize(vars,mean(vars$x), sd(vars$x))); names(Nmns) <- "x"
```

Plot both distributions of sample mean and variance and overlay with the standard normal distribution.

```
library(ggplot2)
require(gridExtra)

# plot histogram
g3 <- ggplot(Nmns, aes(x=x))
g4 <- ggplot(Nv, aes(x=x))

# plot density + standard normal distribution
myNormPlot <- function(fh, title){
        g<- fh+ geom_histogram(aes(y=..density..), binwidth=0.2, colour="black", fill="white")+
        ggtitle(title)+
        theme(plot.title = element_text(lineheight=.8, face="bold"))+
        xlab("")+
        ylab("")+
        stat_function(fun = dnorm, size = 1)+
        xlim(-4,4)
        g

}

g3 <- myNormPlot(g3,"(a) Normalized Sample Mean")
g4 <- myNormPlot(g4,"(b) Normalized Sample Variance")

grid.arrange(g3, g4, ncol=2,
             main=textGrob("Fig3: Comparison with Standard Normal",
                           gp = gpar(fontsize = 14,
                                     fontface = "bold"))
             )
```