Central Limit Theorem and the Exponential Distribution

Kai Wakerman Powell

1 April 2016

Overview

The following explores the exponential distribution and the central limit theorem. It tests how close the simulated variables sit to the theoretical variables. The blow shows that the variables are extremely close. The mean, standard deviation, and variance all sit very closely with the theoretical values. Interestingly, when run with a larged simulated sample, the values decreased across the board. When comparing the distribution of samples, it is clear that they follow an approximate normal distribution.

Setup

These are some standard setup to assist reproducibility.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.4
```

```
setwd("~/Desktop/Coursera/SI/SIAss")
set.seed(1234567890)
```

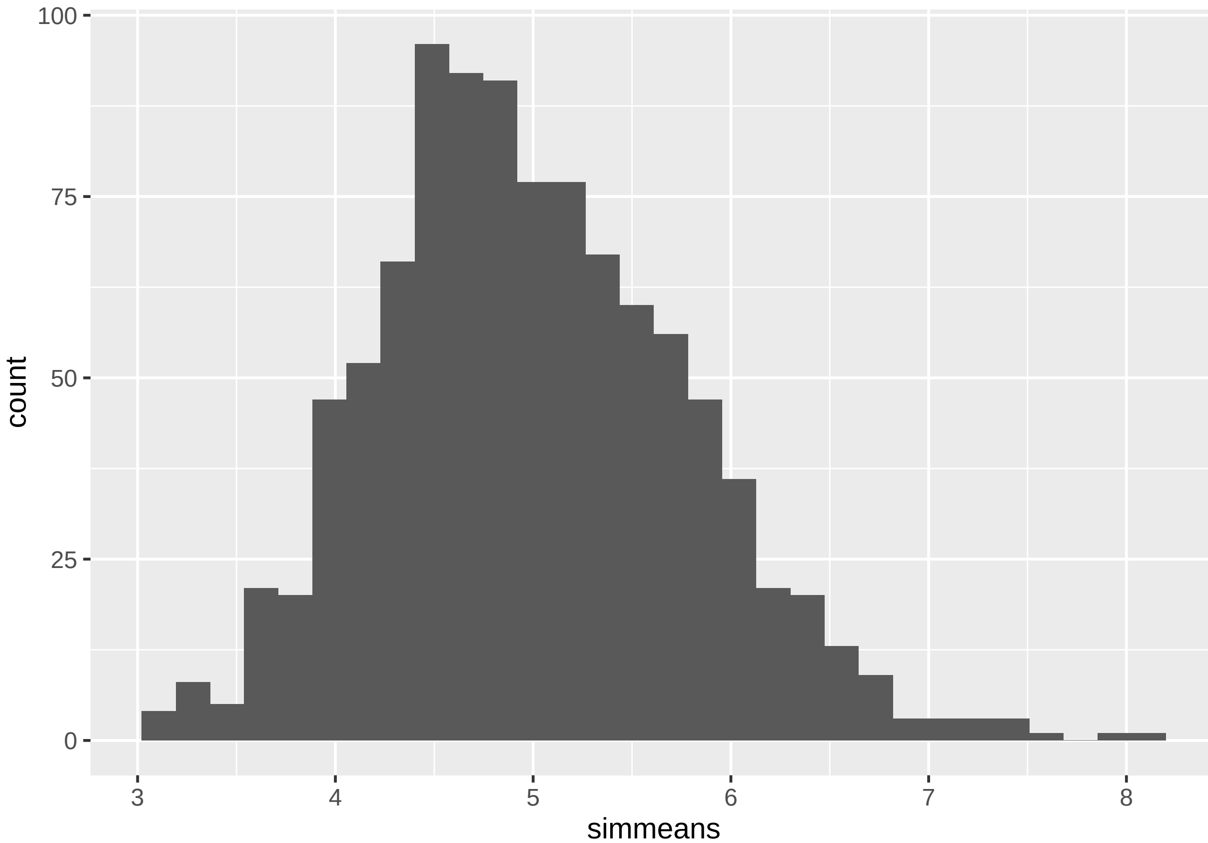
First run:

This sets up the simulation and runs for 1000 simulations as expressed in the assignment instructions:

```
nsims<-1000
exps<-40
lambda<-.2
sims<-matrix(rexp(nsims*exps,lambda),nsims)
simmeans<-data.frame(apply(sims,1,mean))
names(simmeans)<-"Simmeans"
simmeansplot<-qplot(simmeans,geom="histogram")
simmeansplot
```

```
## Don't know how to automatically pick scale for object of type data.frame. Defaulting to continuous.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



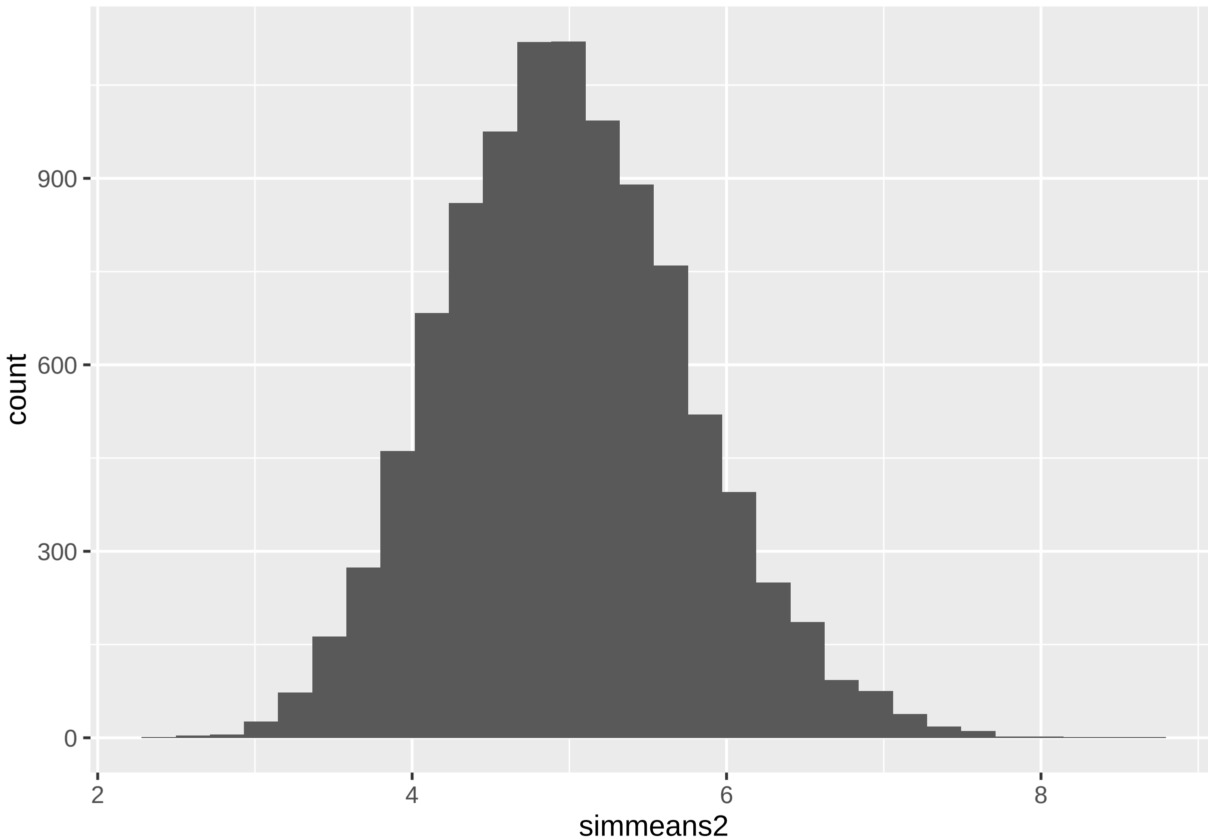
Second run:

Given that the last plot can sometimes look a bit thin, a version with 10000 runs is run below:

```
nsims2<-10000
sims2<-matrix(rexp(nsims2*exps,lambda),nsims2)
simmeans2<-data.frame(apply(sims2,1,mean))
names(simmeans2)<-"Simmeans2"
simmeansplot2<-qplot(simmeans2,geom="histogram")
simmeansplot2
```

```
## Don't know how to automatically pick scale for object of type data.frame. Defaulting to continuous.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Comparing sample means against theoretical means

The theoretical mean = 1/lambda and the theoretical standard deviation = 1/lambda.

```
tmean<-1/lambda
tsd<-((1/lambda)*(1/sqrt(exps)))
tvar<-tsd^2
```

As for the sampled means and standard variations:

```
mean1000<-mean(simmeans$apply.sims..1..mean.)
```

```
## Warning in mean.default(simmeans$apply.sims..1..mean.): argument is not
## numeric or logical: returning NA
```

```
mean10000<-mean(simmeans2$apply.sims2..1..mean.)
```

```
## Warning in mean.default(simmeans2$apply.sims2..1..mean.): argument is not
## numeric or logical: returning NA
```

```
sd1000<-sd(simmeans$apply.sims..1..mean.)
sd10000<-sd(simmeans2$apply.sims2..1..mean.)
var1000<-sd1000^2
var10000<-sd10000^2
```

To compare the theoretical means and standard deviations against the sample means and standard deviations:

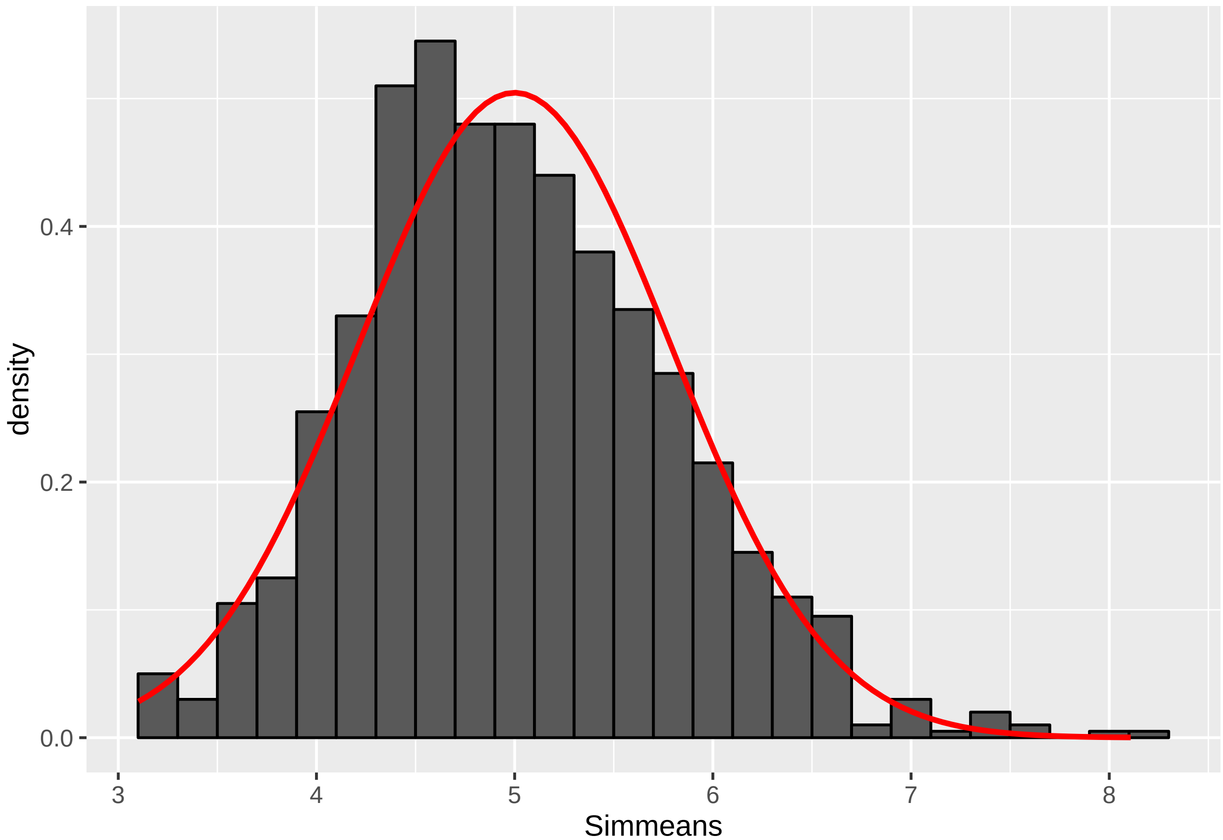
Variable	Theoretical value	1000 simulations	10000 simulations
Mean	5	NA	NA
Standard Deviation	0.7905694	NA	NA
Variance	0.625	NA	NA

This shows that the sample mean, standard deviation and variances are very close.

Distribution

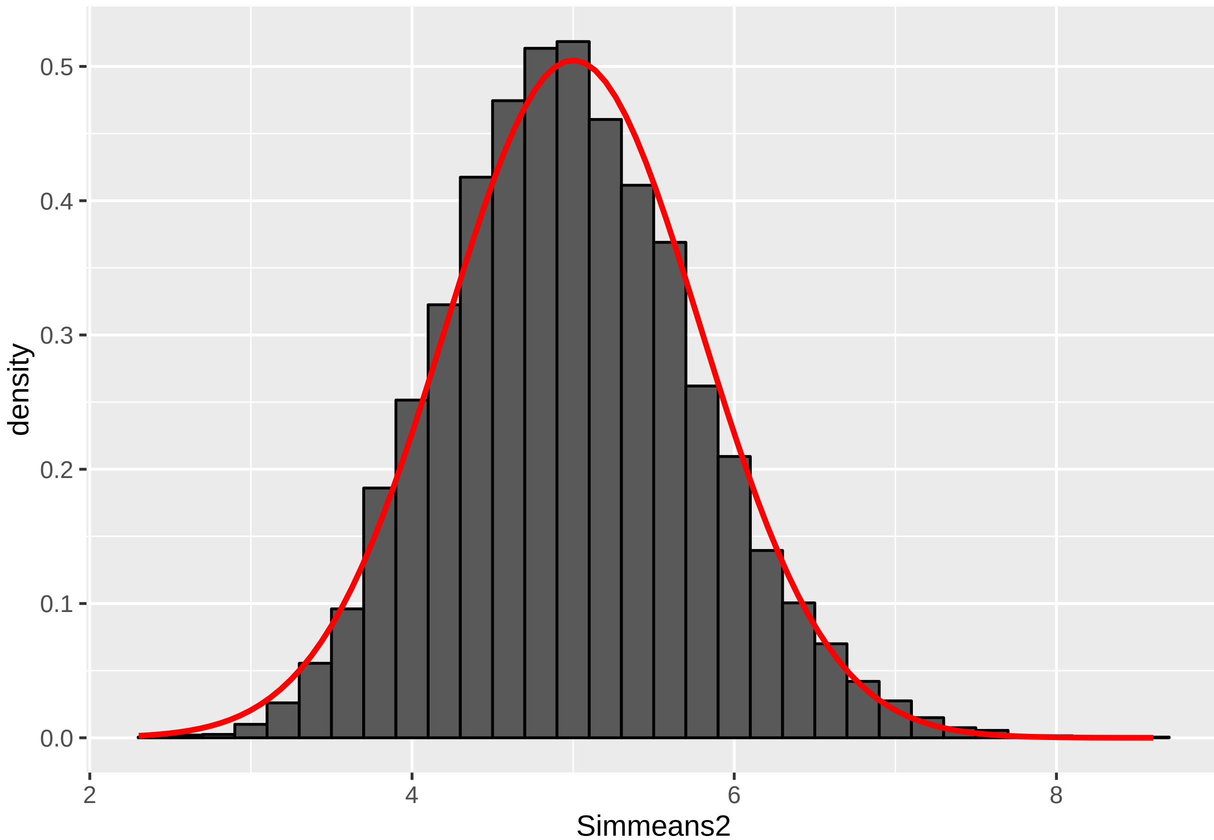
The following overlays the normal distribution, and the sampled distributions for the 1000 and 10000 simulated runs. Looking at the 1000 run simulations it appears that the simulated distributions are approximately normal:

```
normplot<-ggplot(simmeans,aes(x=Simmeans))+geom_histogram(binwidth = lambda,color="black",aes(y = ..dens
normplot<-normplot+stat_function(fun=dnorm,args=list(mean=tmean,sd=tsd),color="red",size=1.0)
normplot
```



When run at 10000 runs, the same is confirmed. However it appears more clear here that there might be some bias in the estimator:

```
normplot2<-ggplot(simmeans2,aes(x=Simmeans2))+geom_histogram(binwidth = lambda,color="black",aes(y = ..d
normplot2<-normplot2+stat_function(fun=dnorm,args=list(mean=tmean,sd=tsd),color="red",size=1.0)
normplot2
```


0.5
0.4
0.3
0.2
0.1
0.0
density
2
4
6
8
Simmeans2