Part 1: Simulation Exercise Instructions

Kiattisak Chaisomboon

07/06/2020

Overview

Simulations

```
library(ggplot2)

set.seed(23)
sim <- 1000
n <- 40
lambda = 0.2

mns = NULL
vrs = NULL
for (i in 1 : sim) {
  rnd = rexp(n, lambda)
  mns = c(mns, mean(rnd))
  vrs = c(vrs, var(rnd))
}
```

Sample Mean versus Theoretical Mean

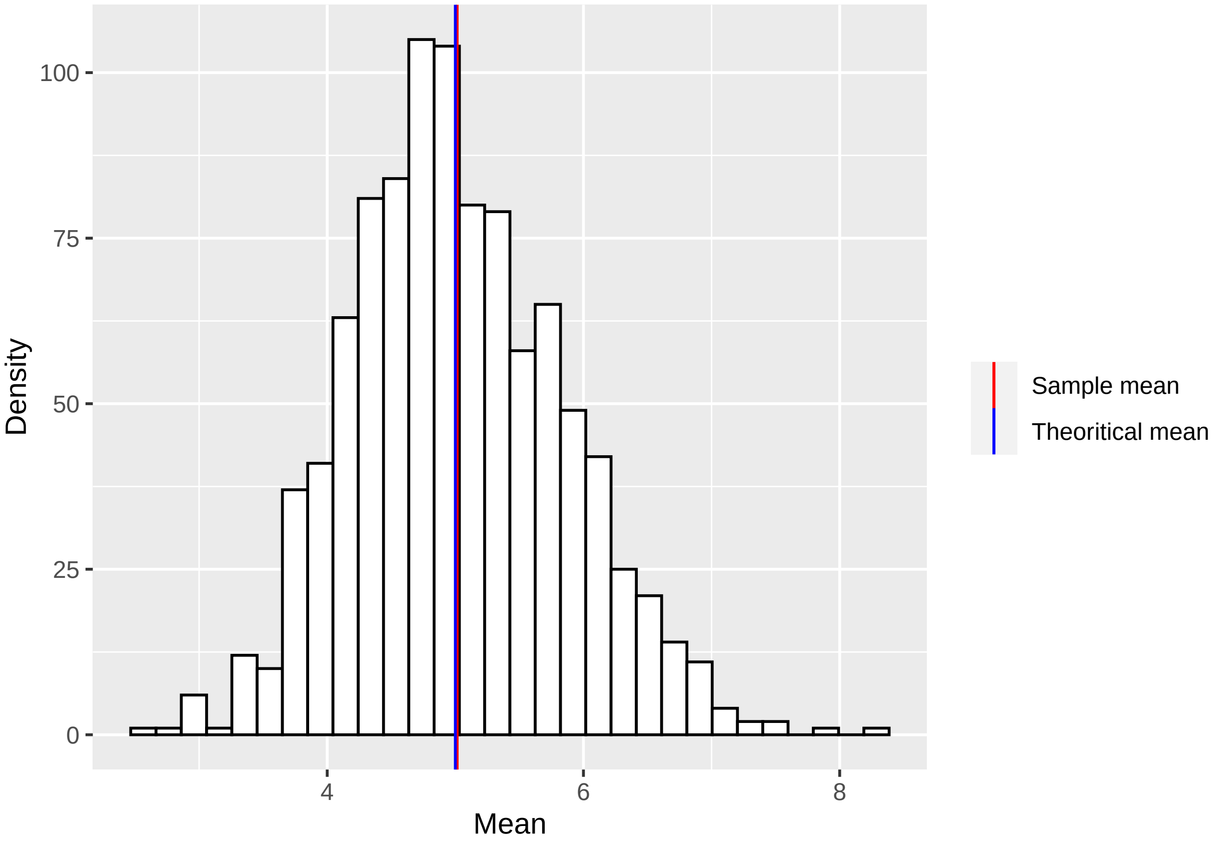
```
sample.mean <- mean(mns)
sample.mean
```

```
## [1] 5.01425
```

```
theoritical.mean <- 1/lambda
theoritical.mean
```

```
## [1] 5
```

```
dfm <- data.frame(mns)
ggplot(dfm, aes(mns)) +
  geom_histogram(bins=30, colour="black", fill="white") +
  geom_vline(aes(xintercept = sample.mean, colour='Sample mean')) +
  geom_vline(aes(xintercept = theoritical.mean, colour='Theoritical mean')) +
  scale_color_manual(name="", values = c('Sample mean' = 'red',
                                  'Theoritical mean' = 'blue')) +
  xlab("Mean") + ylab("Density")
```



Sample Variance versus Theoretical Variance

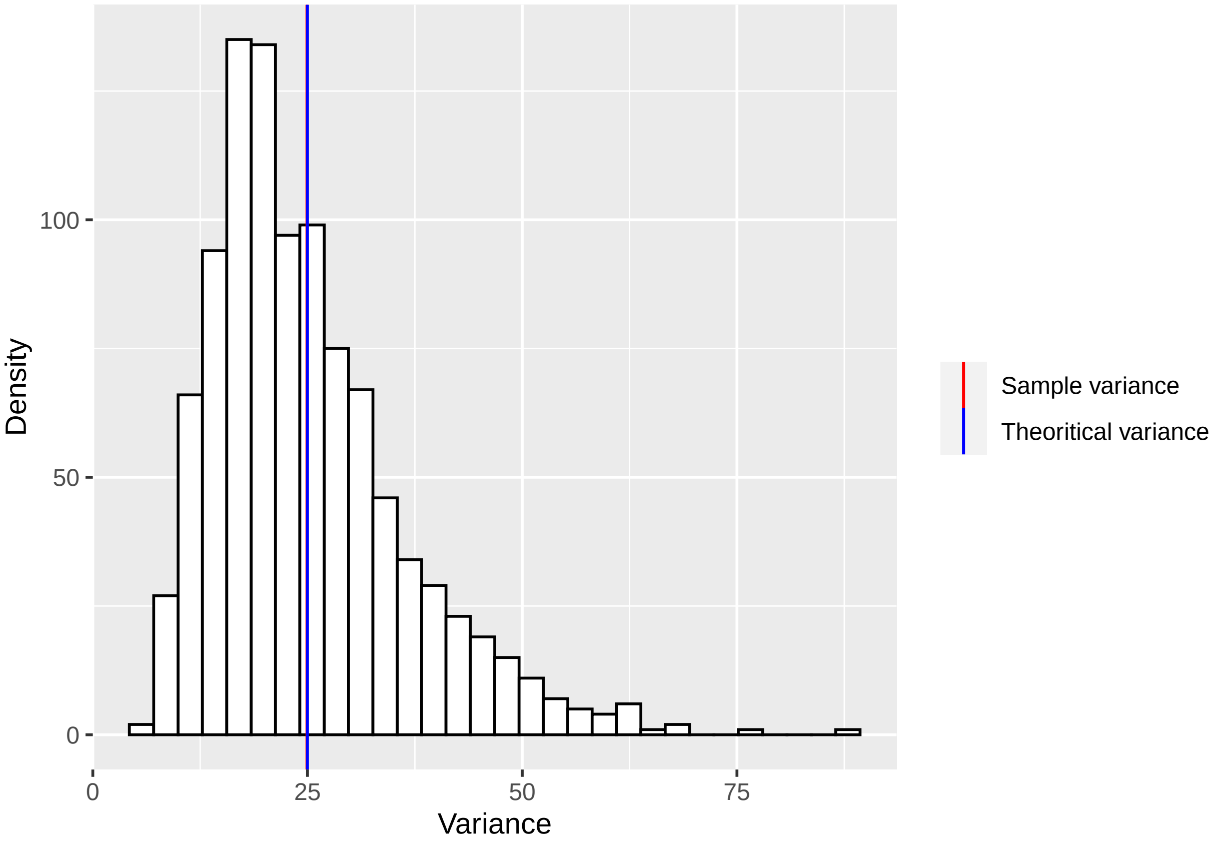
```
sample.variance <- mean(vrs)
sample.variance
```

```
## [1] 24.92895
```

```
theoritical.variance <- (1/lambda)^2
theoritical.variance
```

```
## [1] 25
```

```
dfv <- data.frame(vrs)
ggplot(dfv, aes(vrs)) +
  geom_histogram(bins=30, colour="black", fill="white") +
  geom_vline(aes(xintercept = sample.variance, colour='Sample variance')) +
  geom_vline(aes(xintercept = theoritical.variance, colour='Theoritical variance')) +
  scale_color_manual(name="", values = c('Sample variance' = 'red',
                                         'Theoritical variance' = 'blue')) +
  xlab("Variance") + ylab("Density")
```

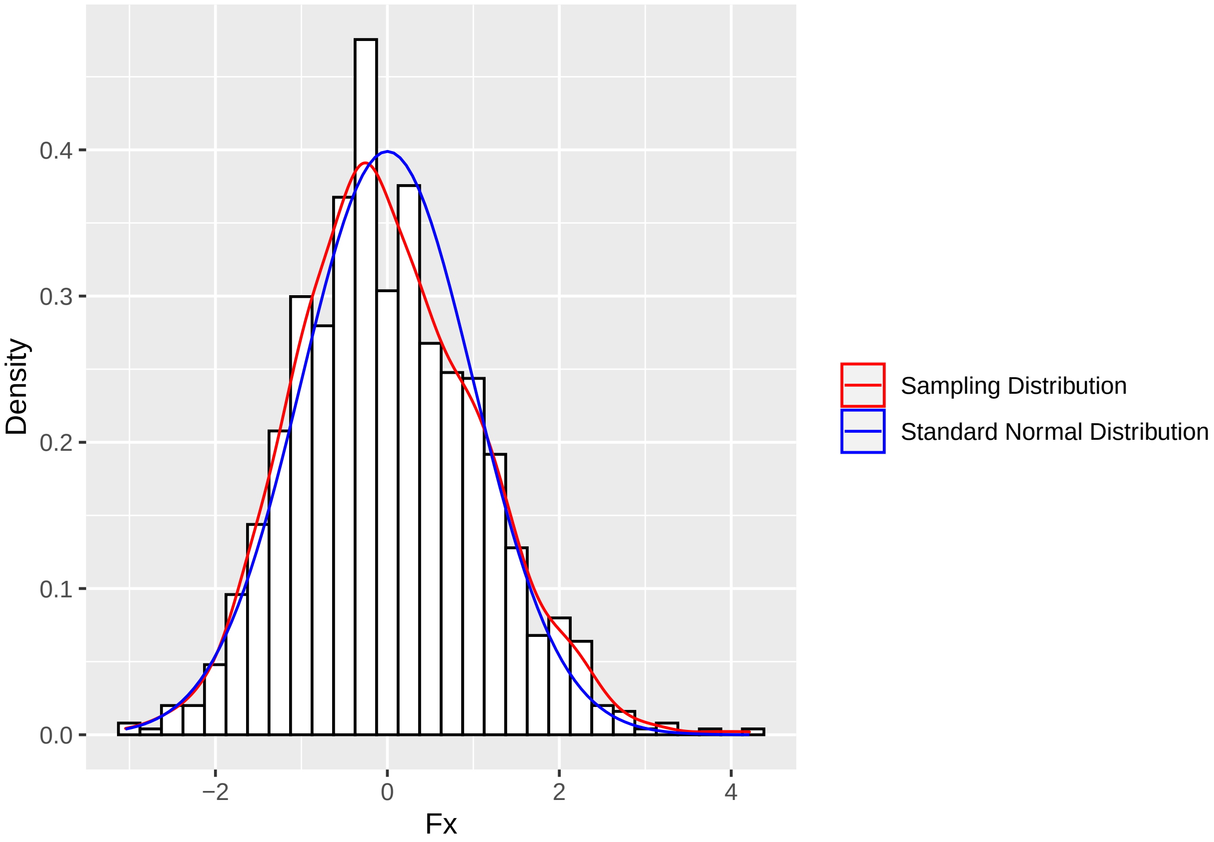


Distribution

define Fx = $\frac{\overline{X_n}-\mu}{\sigma/\sqrt{n}}$ where n is large

```
sampling.dist <- ((mns - sample.mean)/(sample.variance / sqrt(sim)))

df <- data.frame(sampling.dist)
ggplot(df, aes(x=sampling.dist)) +
  geom_histogram(aes(y=..density..), bins=30, colour="black", fill="white") +
  geom_density(aes(colour='Sampling Distribution')) +
  geom_line(stat = "function", fun = "dnorm", args = list(mean = 0, sd = 1),
            aes(colour='Standard Normal Distribution')) +
  scale_color_manual(name="", values = c('Sampling Distribution' = 'red',
                                         'Standard Normal Distribution' = 'blue')) +
  xlab("Fx") + ylab("Density")
```


0.4
0.3
0.2
0.1
0.0
Density
−2
0
2
4
Fx
Sampling Distribution
Standard Normal Distribution