

BIOS6611-Homework1

Randy

9/3/2019

BIOS6611 Homework1

Exercise 1a

Normal($\mu = 125$, $\sigma = 8$)

Poisson($\lambda = 1.5$)

Binomial($n = 5$, $p = 0.15$)

```
set.seed( seed = 55 )
r.norm <- rnorm( n = 10000, mean = 125, sd = 8 ); head( r.norm )

## [1] 125.9611 110.5010 126.2127 116.0462 125.0153 134.5081

r.pois <- rpois( n = 10000, lambda = 1.5 ); head( r.pois )

## [1] 4 2 1 1 1 3

r.binom <- rbinom( n = 10000, size = 5, prob = 0.15); head( r.binom )

## [1] 0 0 1 0 1 2
```

Exercise 1b

for Normal($\mu = 125$, $\sigma = 8$), reference from R help()

the theoretical mean = 125; standard deviation = 8

```
mean( r.norm ); sd( r.norm )

## [1] 124.9802

## [1] 8.055709
```

for Poisson($\lambda = 1.5$), reference from R help()

the theoretical mean = standard deviation = 1.5

```
mean( r.pois ); sd( r.pois )

## [1] 1.4967

## [1] 1.235209
```

for Binomial(size = 5, prob = 0.15), reference from R help()

the theoretical mean = $np = 0.75$; standard deviation = $np(1-p) = 0.6375$

```
mean( r.binom ); sd( r.binom )
```

```
## [1] 0.7494
```

```
## [1] 0.7955268
```

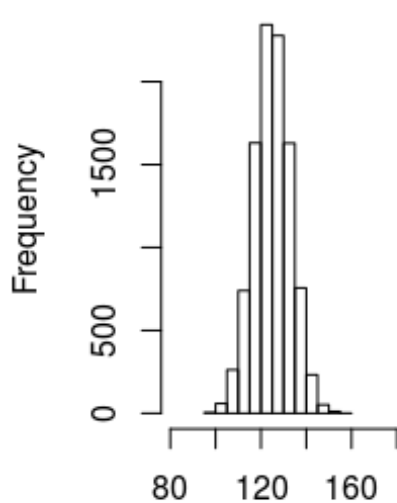
Excercise 1c

the histogram and boxplot for Normal($\mu = 125$, $\sigma = 8$)

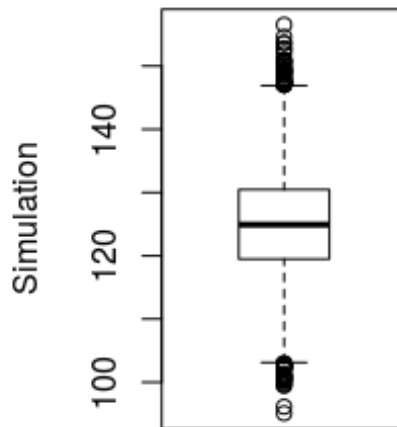
```
par(mfrow=c(1,2))
```

```
hist( r.norm, xlim = c( 80, 180 ), xlab = "Normal( mu = 125, sigma = 8 )", main = "" )
```

```
boxplot( r.norm, xlab = "Normal( mu = 125, sigma = 8 )", ylab = "Simulation" )
```



Normal(mu = 125, sigma = 8



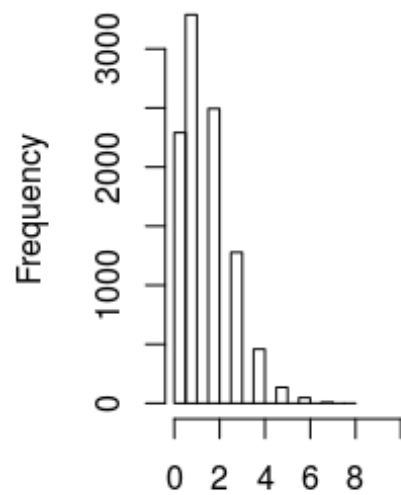
Normal(mu = 125, sigma = 8

the histogram and boxplot for Poisson($\lambda = 1.5$)

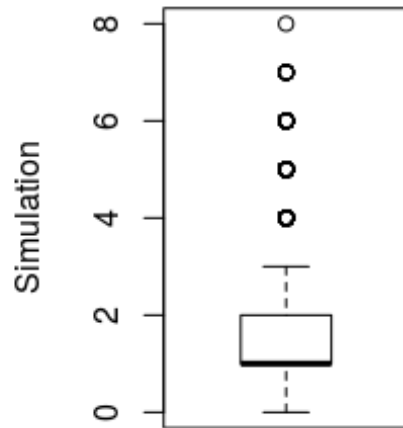
```
par(mfrow=c(1,2))
```

```
hist( r.pois, xlim = c( 0, 10 ), xlab = "Poisson( lambda = 1.5 )", main = "" )
```

```
boxplot( r.pois, xlab = "Poisson( lambda = 1.5 )", ylab = "Simulation" )
```



Poisson(lambda = 1.5)



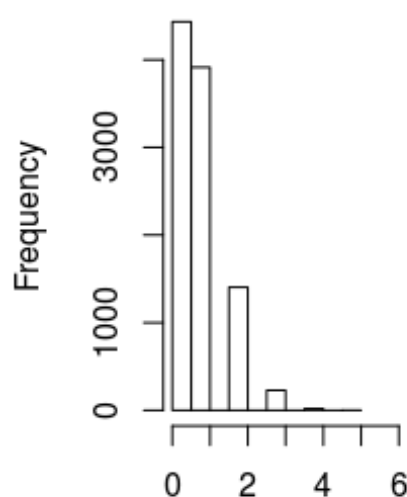
Poisson(lambda = 1.5)

the histogram and boxplot for Binomial($n = 5$, $p = 0.15$)

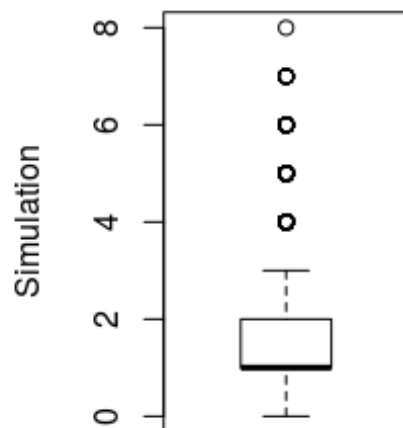
```
par(mfrow = c(1,2))
```

```
hist( r.binom, xlim = c( 0, 6 ), xlab = "Binomial( size = 5, prob = 0.15 )", main = "" )
```

```
boxplot( r.pois, xlab = "Binomial( size = 5, prob = 0.15 )", ylab = "Simulation" )
```



Binomial(size = 5, prob = 0.15



Binomial(size = 5, prob = 0.15

Exercise 2a

```
set.seed( seed = 555 )
r.norm.2 <- rnorm( n = 1000, mean = 40, sd = 10)
mean( r.norm.2 ); median( r.norm.2 ); sd( r.norm.2 )

## [1] 39.76918
## [1] 39.62885
## [1] 9.805364

number.sims <- 1000
sample.size <- 10
vector.sample.means <- rep( NA, number.sims)
vector.sample.medians <- rep( NA, number.sims)
vector.sample.sd <- rep( NA, number.sims)
for (i in 1:number.sims) {
  vector.sample.means[i] <- mean( rnorm(n = sample.size, mean = 40, sd = 10) )
  vector.sample.medians[i] <- median( rnorm(n = sample.size, mean = 40, sd = 10) )
  vector.sample.sd[i] <- sd( rnorm(n = sample.size, mean = 40, sd = 10) )
}
head( vector.sample.means)

## [1] 39.95319 38.67707 39.01104 41.44201 36.75610 41.25785

head( vector.sample.medians)

## [1] 35.31841 41.04394 35.89209 43.16669 42.26291 40.98771
```

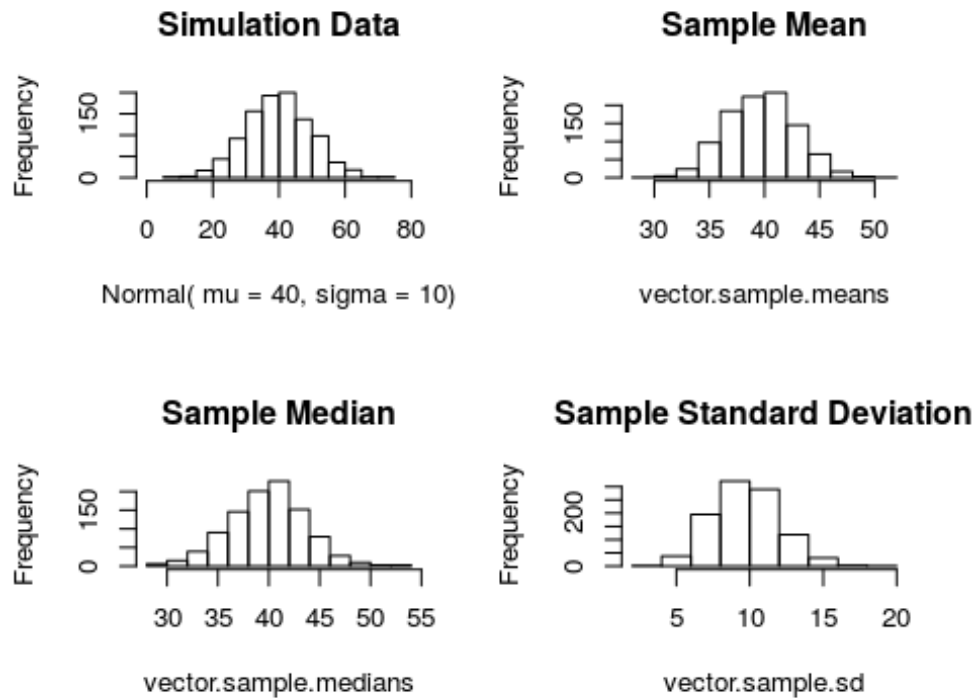
```

head( vector.sample.sd)

## [1] 6.874344 10.260156 13.990310 9.553949 8.011802 11.994217

par(mfrow = c( 2, 2 ))
hist( r.norm.2, xlim = c( 0, 80 ), xlab = "Normal( mu = 40, sigma = 10)", main = "Simulation Data" )
hist( vector.sample.means, main = "Sample Mean" )
hist( vector.sample.medians, main = "Sample Median" )
hist( vector.sample.sd, main = "Sample Standard Deviation" )

```



Exercise 2b.

All of them are normal distributed, for Mean of the sample $\sim \text{Normal}(40, 10)$ as $(\mu = X \cdot 10, \sigma^2 = \sigma^2/10)$.

```

mean( vector.sample.means )

## [1] 39.68895

sd( vector.sample.means )

## [1] 3.12022

median( vector.sample.means )

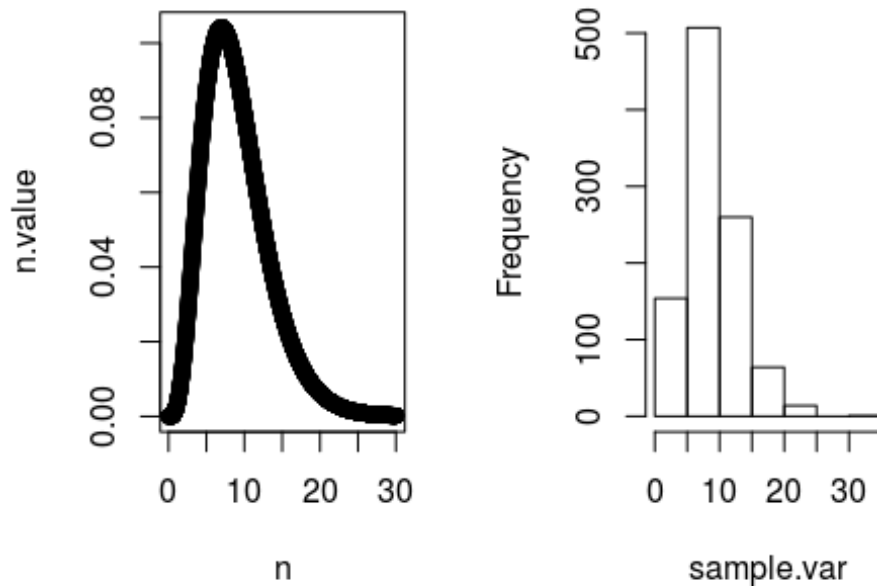
## [1] 39.65875

```

Exercise 2c.

```
n <- seq( from = 0.1, to = 30, by = 0.01 )
n.value <- dchisq( n, df = 9 )
sample.var <- vector.sample.sd^2 * (9/100)
par(mfrow = c( 1, 2 ))
plot( n, n.value, main = "The variance of theoretical distribution")
hist( sample.var , main = "The variance of sampling distribution" )
```

variance of theoretical dis variance of sampling dist



Exercise 3a.

```
set.seed( seed = 555 )
r.norm.3 <- rnorm( n = 100, mean = 70, sd = 15 )
head( r.norm.3 )

## [1] 65.05210 77.55470 75.61554 98.32930 43.30146 83.28447

mean( r.norm.3 )

## [1] 70.38257

median( r.norm.3 )

## [1] 70.83197

mu = 70
bias = median( r.norm.3 ) - mu; bias

## [1] 0.831974
```

Exercise 3b.

according to the plots, the bias due to the sampling median comparing to the expected value μ (population mean) decreases with the increase of the sample size. the median estimator suppose to be an unbiased estimator comparing with population mean with extremely large sample size.

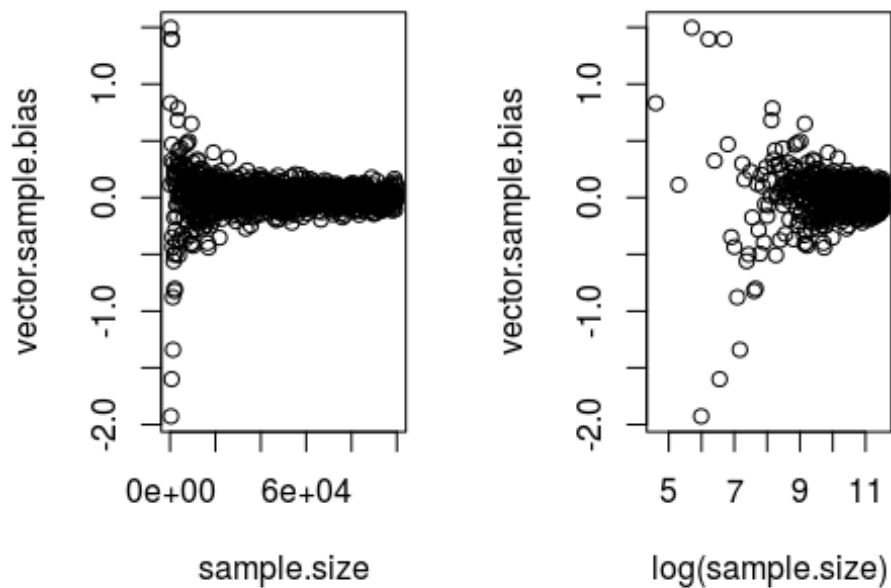
```
set.seed( seed = 555 )
mu <- 70
number.sims <- 1000
sample.size <- seq( 100, 100000, by = 100)
vector.sample.medians <- rep( NA, number.sims)
vector.sample.bias <- rep( NA, number.sims)
vector.sample.var <- rep( NA, number.sims)
for (i in 1:number.sims) {
  vector.sample.medians[i] <-
    median( rnorm(n = sample.size[i], mean = 70, sd = 15) )
  vector.sample.bias[i] <- vector.sample.medians[i] - mu
  vector.sample.var[i] <- vector.sample.bias[i]^2/sample.size[i]
}
head( vector.sample.medians )

## [1] 70.83197 70.11278 71.49597 68.07316 71.39608 70.32446

head( vector.sample.bias )

## [1] 0.8319740 0.1127752 1.4959678 -1.9268443 1.3960762 0.3244622

par(mfrow = c( 1, 2 ))
plot( sample.size, vector.sample.bias, xlim = c(100, 100000))
plot( log(sample.size), vector.sample.bias)
```



Exercise 3c.

the variance of the data wrt the median estimator should decrease with the sample size increases. the variance goes down with the decrease of bias. according to the definition of variance, $\text{variance} = \text{bias}^2/n$

```
head( vector.sample.var )
```

```
## [1] 6.921807e-03 6.359117e-05 7.459732e-03 9.281823e-03 3.898057e-03
```

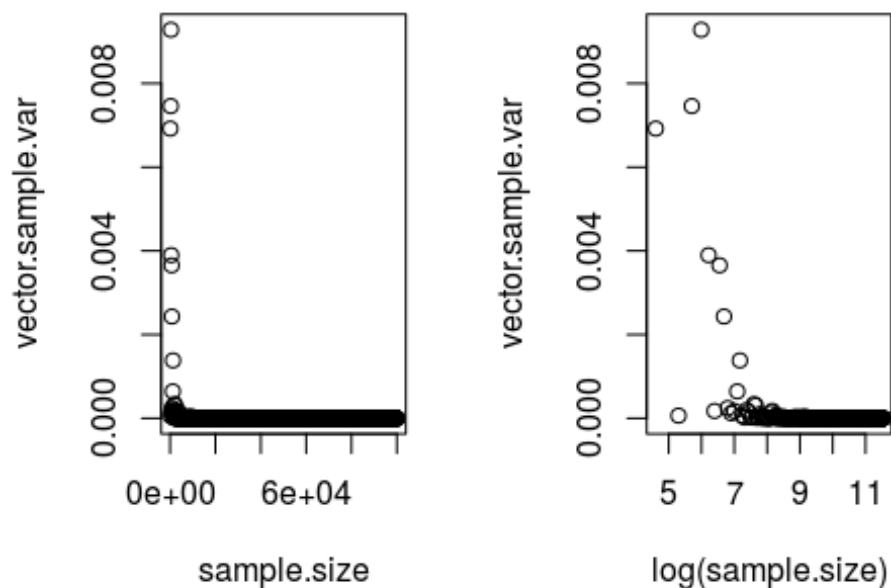
```
## [6] 1.754596e-04
```

```
par(mfrow=c( 1, 2 ))
```

```
plot( sample.size, vector.sample.var, xlim=c(100, 100000))
```

```
plot( log(sample.size), vector.sample.var )
```





Exercise 3e.

To be honest, I have no idea. it seems this method could make the biased estimator unbiased. Here are something from Wikipedia.

In estimation theory and statistics, the Cramér–Rao bound (CRB), Cramér–Rao lower bound (CRLB), Cramér–Rao inequality, Fréchet–Darmois–Cramér–Rao inequality, or information inequality expresses a lower bound on the variance of unbiased estimators of a deterministic (fixed, though unknown) parameter. In its simplest form, the bound states that the variance of any unbiased estimator is at least as high as the inverse of the Fisher information. An unbiased estimator which achieves this lower bound is said to be (fully) efficient. Such a solution achieves the lowest possible mean squared error among all unbiased methods, and is therefore the minimum variance unbiased (MVU) estimator. However, in some cases, no unbiased technique exists which achieves the bound. This may occur either if for any unbiased estimator, there exists another with a strictly smaller variance, or if an MVU estimator exists, but its variance is strictly greater than the inverse of the Fisher information.

Exercise 5a,b.

I do not know how to put out data from the internet yet, but I am still working on it...

```
##{r} ## read.csv ## NAWS <- read.csv("BIOS6611-Alexander Kaizer/NAWS2014.csv") ## hist(
NAWS$A09, xlab = "the number of years", ylab = "the number of farmers", main = "" ) ##
```