Question no. 2

Abhishek Chakraborty

2022-11-15

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(ggplot2)
```

1.

```
ll = function(log_al, x, si){
  log_like = sum(log(dgamma(x, shape = exp(log_al), scale = si)))
  return(-log_like)
}

MyMLE = function(x, si){
  log_al_initial_guess = log(mean(x)^2/var(x))
  estimator = optim(log_al_initial_guess,ll, x = x, si = si)
  log_al_hat = estimator$par
  return(log_al_hat)
}
```

```
estimate = function(n,al,si){
  estimates = c()
  for (i in 1:1000){
    samples = rgamma(n,shape = al,scale = si)
    estimates = append(estimates, MyMLE(x = samples, si = si))
  }
  return(estimates)
}
```

2.

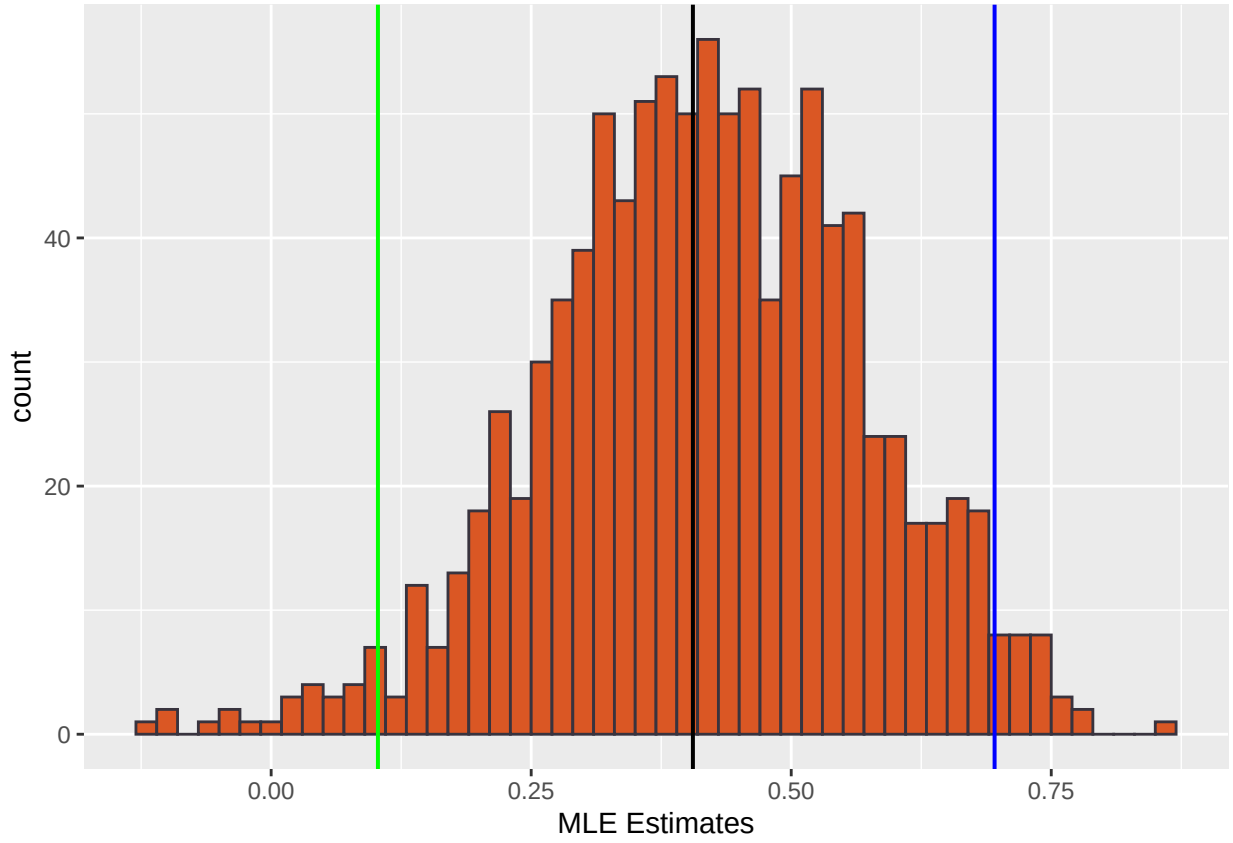
```
n = 20
al = 1.5
si = 2.2

est_MLE = tibble(estimate(n = n, al = al, si = si))
colnames(est_MLE) = c("MLE Estimates")
```

```
plt = ggplot(est_MLE,aes(`MLE Estimates`))
plt = plt+geom_histogram(fill = "#DA5724",color = "#38333E",binwidth = 0.02)
plt = plt+geom_vline(xintercept = log(al),
             size = 0.7,color = "black")


percentile_2.5 = quantile(est_MLE$`MLE Estimates`, probs = 0.025)

percentile_97.5 = quantile(est_MLE$`MLE Estimates`, probs = 0.975)
plt = plt+geom_vline(xintercept = percentile_2.5,
                     size = 0.7,color = "green")
plt = plt+geom_vline(xintercept = percentile_97.5,
                     size = 0.7,color = "blue")
plt
```




```
gap1 = as.numeric(percentile_97.5) - as.numeric(percentile_2.5)
gap1
```

```
## [1] 0.5928719
```

3.

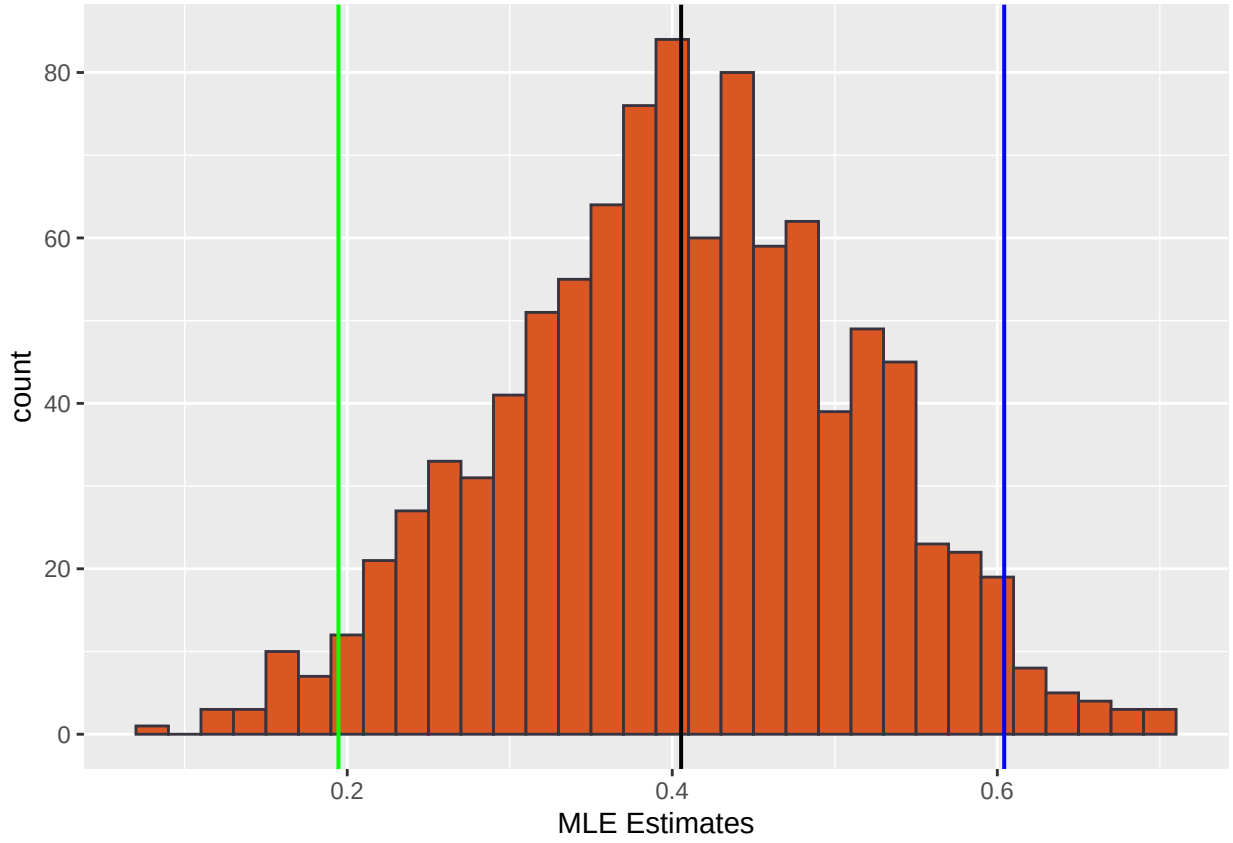
```
n = 40
al = 1.5
si = 2.2

est_MLE = tibble(estimate(n = n, al = al, si = si))
colnames(est_MLE) = c("MLE Estimates")
```

```
plt = ggplot(est_MLE,aes(`MLE Estimates`))
plt = plt+geom_histogram(fill = "#DA5724",color = "#38333E",binwidth = 0.02)
plt = plt+geom_vline(xintercept = log(al),
             size = 0.7,color = "black")


percentile_2.5 = quantile(est_MLE$`MLE Estimates`, probs = 0.025)

percentile_97.5 = quantile(est_MLE$`MLE Estimates`, probs = 0.975)
plt = plt+geom_vline(xintercept = percentile_2.5,
                     size = 0.7,color = "green")
plt = plt+geom_vline(xintercept = percentile_97.5,
                     size = 0.7,color = "blue")
plt
```




```
gap2 = as.numeric(percentile_97.5) - as.numeric(percentile_2.5)
gap2
```

```
## [1] 0.4096707
```

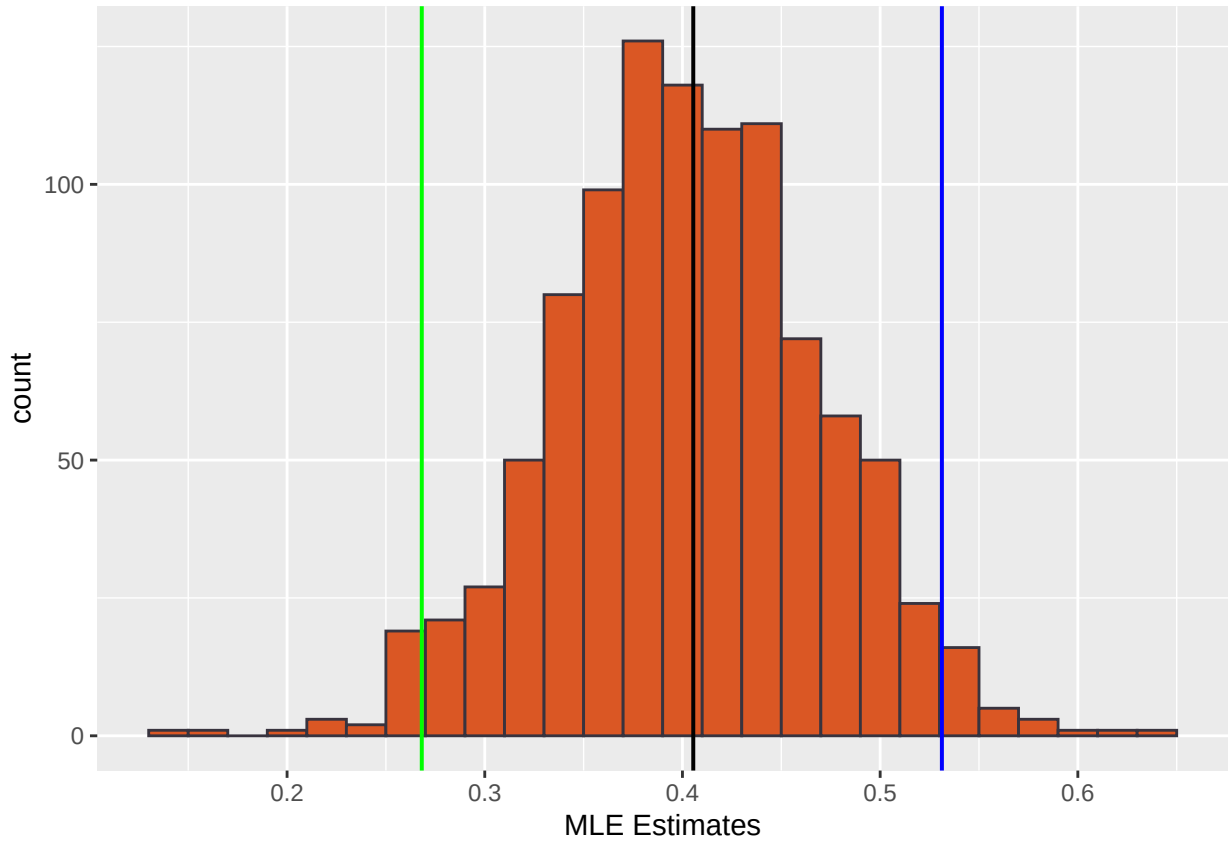
```
n = 100
al = 1.5
si = 2.2

est_MLE = tibble(estimate(n = n, al = al, si = si))
colnames(est_MLE) = c("MLE Estimates")
```

```
plt = ggplot(est_MLE,aes(`MLE Estimates`))
plt = plt+geom_histogram(fill = "#DA5724",color = "#38333E",binwidth = 0.02)
plt = plt+geom_vline(xintercept = log(al),
              size = 0.7,color = "black")


percentile_2.5 = quantile(est_MLE$`MLE Estimates`, probs = 0.025)

percentile_97.5 = quantile(est_MLE$`MLE Estimates`, probs = 0.975)
plt = plt+geom_vline(xintercept = percentile_2.5,
                     size = 0.7,color = "green")
plt = plt+geom_vline(xintercept = percentile_97.5,
                     size = 0.7,color = "blue")
plt
```




```
gap3 = as.numeric(percentile_97.5) - as.numeric(percentile_2.5)
gap3
```

```
## [1] 0.263092
```

5.

```
gap1
```

```
## [1] 0.5928719
```

```
gap2
```

```
## [1] 0.4096707
```

```
gap3
```

```
## [1] 0.263092
```

As it is evident from both the graphs and the numeric results, the gap between 97.5 percentile and 2.5 percentile decreases as the sample size increases.