Assignment 4 Question 1 c)

Sheen Thusoo

30/11/2021

i. Calculate the two attributes of interest using the given sample.

```
data <- read.csv('EconomicMobility.csv')
```

```
CommunitiesSample = c(265, 596, 270, 334, 653, 273,  93,  58, 113, 668,
                      235, 243, 703, 672, 411, 231, 723, 127, 640, 217,
                      626, 279, 482, 395, 410, 162,   7, 603,  28, 100,
                      68, 141, 593, 564, 557, 604, 443, 202, 480, 285,
                      210, 585, 199, 224, 577, 551, 464, 611, 292, 649,
                      80, 180,   3, 463, 479,  77, 453, 241, 548, 488,
                      447, 396, 124, 552, 340, 615,  63, 380, 599, 590,
                      386,  99, 374, 225, 116, 610, 215, 651,  55, 563,
                      562, 122, 476, 355,  36, 293, 534, 652,  53, 571,
                      398, 353, 383, 627, 352, 377, 537, 151, 392,  51)
```

```
VarIQR <- function(pop, N) {
  variance <- var(pop) * (N-1)/N
  iqr <- IQR(pop)
  c(variance, iqr)
}
```

```
communities.sample <- data$Mobility[CommunitiesSample]
var_iqr <- VarIQR(communities.sample, length(communities.sample))
```

```
cat("Variance of Sample: ", var_iqr[1])
```

```
## Variance of Sample:  0.003185652
```

```
cat("IQR of Sample: ", var_iqr[2])
```

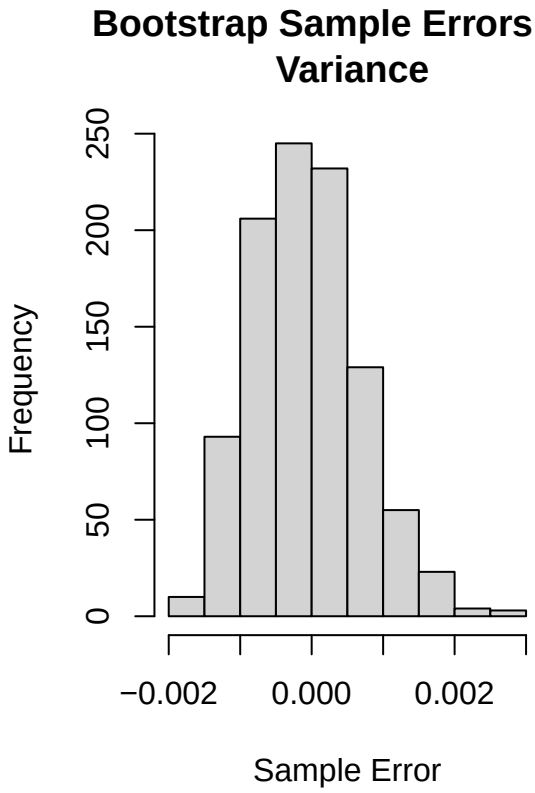
```
## IQR of Sample:  0.05140013
```

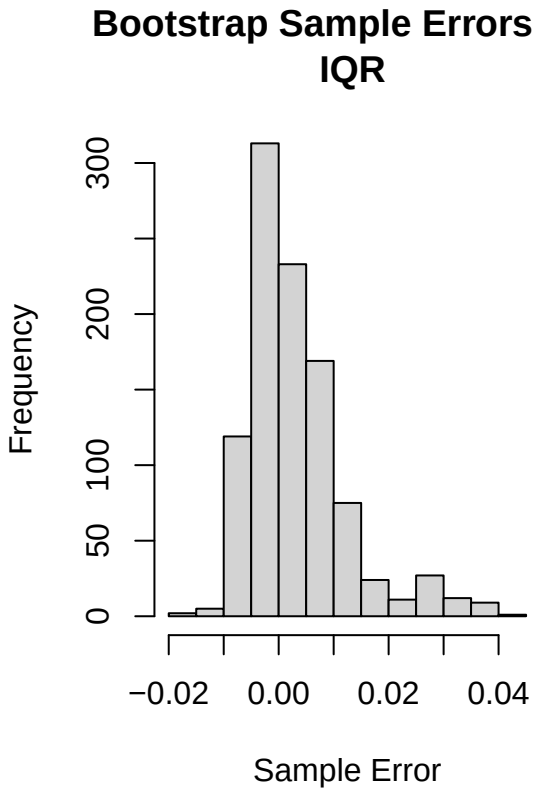
ii. By re-sampling the sample $\mathcal{S}$ with replacement, construct $B = 1000$ bootstrap samples $S_1^\star, S_2^\star, \ldots, S_{1000}^\star$ and calculate the two attributes of interest on each bootstrap sample. Then construct two histograms (in a single row) of the bootstrap sample error for each attribute. Make sure you label your histograms clearly.

```
popSize <- function(pop) {
  if (is.vector(pop))
  {if (is.logical(pop))
    ## then assume TRUE values identify units
    sum(pop) else length(pop)}
  else nrow(pop)
}

getSample <- function(pop, size, replace=FALSE) {
  N <- popSize(pop)
  pop[sample(1:N, size, replace = replace)]
}
```

```
S <- communities.sample
B <- 1000
n <- 100
set.seed(341)
bootstraps <- sapply(1:B, FUN = function(b) getSample(S, n, replace=T))
boot.attributes <- apply(bootstraps, 2, function(x) VarIQR(x, length(x) ))
```

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

boot_error_var <- apply(boot.attributes, MARGIN=2, FUN=function(s) {
  s[1] - var_iqr[1]
})
boot_error_iqr <- apply(boot.attributes, MARGIN=2, FUN=function(s) {
  s[2] - var_iqr[2]
})

hist(boot_error_var, main="Bootstrap Sample Errors for \n Variance", xlab="Sample Error")
hist(boot_error_iqr, main="Bootstrap Sample Errors for \n IQR", xlab="Sample Error")
```





iii. Calculate standard errors for each sample estimate and then construct a 95% confidence interval for the population quantity using the percentile method.

```
SE <- function(estimate, B) {
  sqrt( sum((estimate - mean(estimate))^2) / B )
}

var_SE <- SE(boot.attributes[1,], B)
cat('Standard Error for Variance: ', var_SE)
```

```
## Standard Error for Variance:  0.0007493775
```

```
iqr_SE <- var_SE <- SE(boot.attributes[2,], B)
cat('Standard Error for IQR: ', iqr_SE)
```

```
## Standard Error for IQR:  0.009040104
```

```
var_interval <- quantile(boot.attributes[1,], prob=c(0.025, 0.975))
paste0("The 95% confidence interval for variance is [",var_interval[1], ", ", var_interval[2], "]")
```

```
## [1] "The 95% confidence interval for variance is [0.00188994389925433, 0.00472602330018961]"
```

```r
iqr_interval <- quantile(boot.attributes[2,], prob=c(0.025, 0.975))
paste0("The 95% confidence interval for IQR is [",iqr_interval[1], ", ", iqr_interval[2], "]")
```

```
## [1] "The 95% confidence interval for IQR is [0.042726155105625, 0.08065661597875]"
```