IT24102543
Bawanya S.A.S.
IT2120 - Probability and Statistics
Lab 08

Exercise

Q1)

```
> setwd("C:\\Users\\sesal\\OneDrive\\Desktop\\IT24102543_LAB08")
> getwd()
[1] "C:/Users/sesal/OneDrive/Desktop/IT24102543_LAB08"
> data <- read.table("Exercise - LaptopsWeights.txt", header = TRUE)
> # R converts "Weight(kg)" to "Weight.kg."
> laptop_weights <- data$Weight.kg.
> n_population <- length(laptop_weights)
> population_mean <- mean(laptop_weights)
> # Adjust R's sample sd to get the population sd
> population_sd <- sd(laptop_weights) * sqrt((n_population - 1) / n_population)
> print(paste("Population Mean:", population_mean))
[1] "Population Mean: 2.468"
> print(paste("Population Standard Deviation:", population_sd))
[1] "Population Standard Deviation: 0.252885349516337"
> samples <- c()
> setwd("C:\\Users\\sesal\\OneDrive\\Desktop\\IT24102543_LAB08")
> getwd()
[1] "C:/Users/sesal/OneDrive/Desktop/IT24102543_LAB08"
> data <- read.table("Exercise - LaptopsWeights.txt", header = TRUE)
> # R converts "Weight(kg)" to "Weight.kg."
> laptop_weights <- data$Weight.kg.
> n_population <- length(laptop_weights)
> population_mean <- mean(laptop_weights)
> # Adjust R's sample sd to get the population sd
> population_sd <- sd(laptop_weights) * sqrt((n_population - 1) / n_population)
> print(paste("Population Mean:", population_mean))
[1] "Population Mean: 2.468"
> print(paste("Population Standard Deviation:", population_sd))
[1] "Population Standard Deviation: 0.252885349516337"
```

Q2)

```
> #Q2
> samples <- c()
> number_of_samples <- 25
> sample_size <- 6
> for (i in 1:number_of_samples) {
+   s <- sample(laptop_weights, sample_size, replace = TRUE)
+   samples <- cbind(samples, s)
+ }
> colnames(samples) <- paste("Sample", 1:number_of_samples)
> sample_means <- apply(samples, 2, mean)
> sample_sds <- apply(samples, 2, sd)
> print(sample_means)
 Sample 1  Sample 2  Sample 3  Sample 4  Sample 5  Sample 6  Sample 7  Sample 8  Sample 9 Sample 10 Sample 11 
 2.333333  2.633333  2.516667  2.636667  2.465000  2.541667  2.286667  2.228333  2.465000  2.385000  2.378333 
Sample 12 Sample 13 Sample 14 Sample 15 Sample 16 Sample 17 Sample 18 Sample 19 Sample 20 Sample 21 Sample 22 
 2.410000  2.401667  2.310000  2.536667  2.468333  2.446667  2.373333  2.506667  2.505000  2.416667  2.566667 
Sample 23 Sample 24 Sample 25 
 2.468333  2.478333  2.481667 
> print(sample_sds)
  Sample 1   Sample 2   Sample 3   Sample 4   Sample 5   Sample 6   Sample 7   Sample 8   Sample 9  Sample 10 
0.19438793 0.09953224 0.26341349 0.11994443 0.19274335 0.17451838 0.18007406 0.32975243 0.24704251 0.18854708 
 Sample 11  Sample 12  Sample 13  Sample 14  Sample 15  Sample 16  Sample 17  Sample 18  Sample 19  Sample 20 
0.25794702 0.26145745 0.24070037 0.26548070 0.20519909 0.30596841 0.29971097 0.39988332 0.27045640 0.20801442 
 Sample 21  Sample 22  Sample 23  Sample 24  Sample 25 
0.16342174 0.17142540 0.22560290 0.23301645 0.22973173 
```

Q3)

```
> #Q3
> mean_of_sample_means <- mean(sample_means)
> sd_of_sample_means <- sd(sample_means) # This is the Standard Error
> print(paste("Mean of the 25 Sample Means:", mean_of_sample_means))
[1] "Mean of the 25 Sample Means: 2.4496"
> print(paste("Standard Deviation of the 25 Sample Means (Standard Error):", sd_of_sample_means))
[1] "Standard Deviation of the 25 Sample Means (Standard Error): 0.100188803248149"
> # Compare with population parameters
> theoretical_standard_error <- population_sd / sqrt(sample_size)
> print(paste("Population Mean:", population_mean))
[1] "Population Mean: 2.468"
> print(paste("Theoretical Standard Error (Population SD / sqrt(n)):", theoretical_standard_error))
[1] "Theoretical Standard Error (Population SD / sqrt(n)): 0.103240011623401"
> 
> 
```

Environment History Connections Tutorial

Import Dataset 146 MiB List

R Global Environment

Data	
data	40 obs. of 1 variable
samples	num [1:6, 1:25] 2.41 2.2 2.13 2.2 2.65 2.41 2.71 2.46 2.7 2.57 ...
Values	
i	25L
laptop_weights	num [1:40] 2.46 2.45 2.47 2.71 2.46 2.05 2.6 2.42 2.43 2.53 ...
mean_of_sample_means	2.4496
n_population	40L
number_of_samples	25
population_mean	2.468
population_sd	0.252885349516337
s	num [1:6] 2.32 2.57 2.46 2.13 2.76 2.65
sample_means	Named num [1:25] 2.33 2.63 2.52 2.64 2.46 ...
sample_sds	Named num [1:25] 0.1944 0.0995 0.2634 0.1199 0.1927 ...
sample_size	6
sd_of_sample_means	0.100188803248149
theoretical_standard_err...	0.103240011623401