Sri Lanka Institute of Information Technology

<IT24103527>

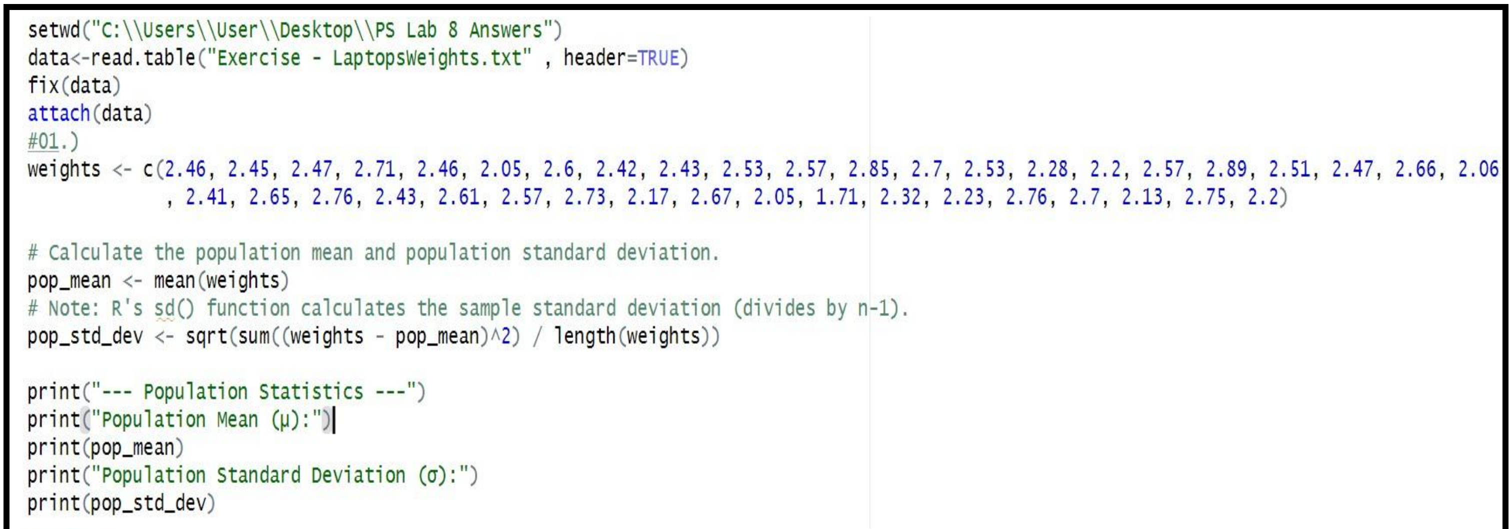
<De Silva S.N.D.D>

<LAB SHEET 08>

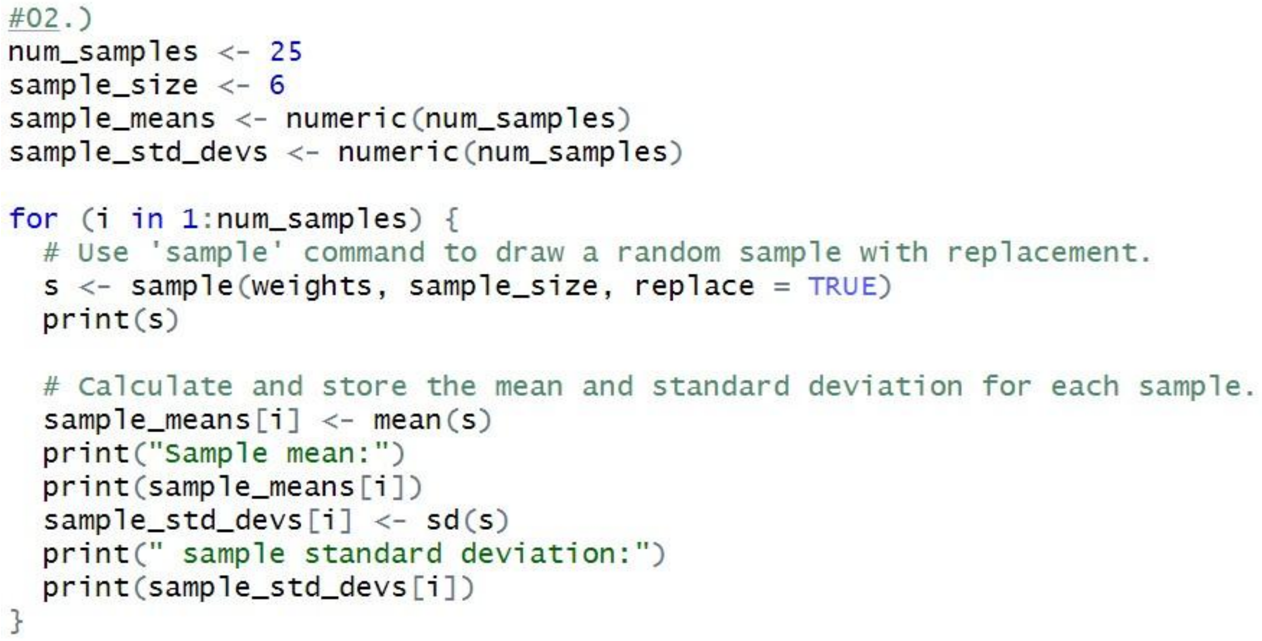
IT2120 | Probability & Statistics

Year 2 Semester 1

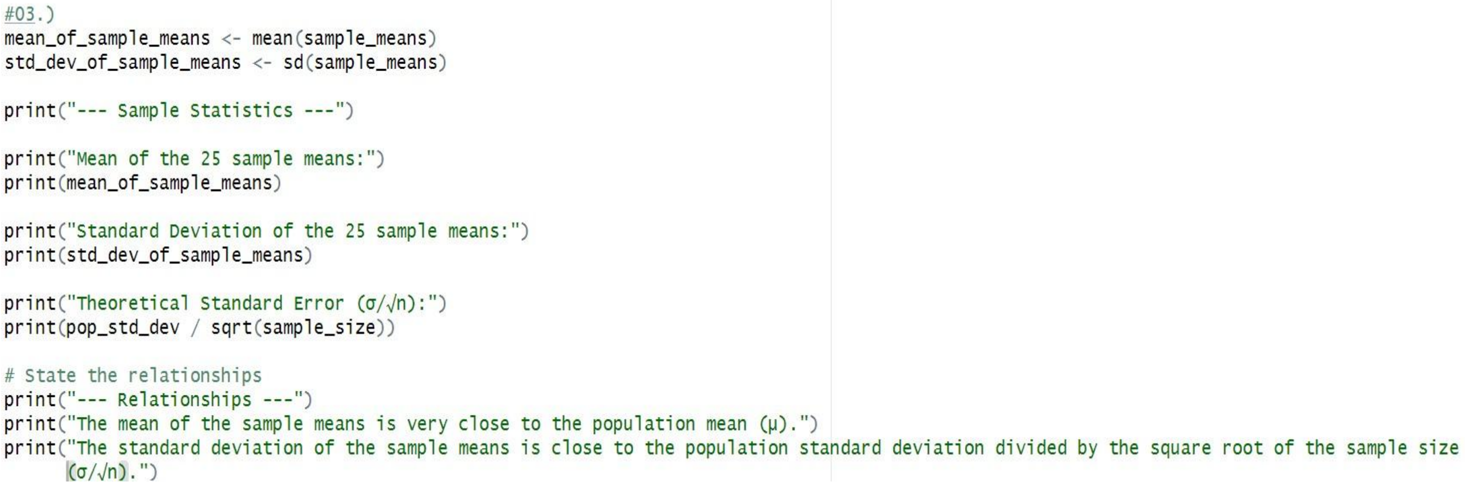
Lab 8 Answers



```
setwd("C:\\Users\\User\\Desktop\\PS Lab 8 Answers")
data<-read.table("Exercise - LaptopsWeights.txt" , header=TRUE)
fix(data)
attach(data)
#01.)
weights <- c(2.46, 2.45, 2.47, 2.71, 2.46, 2.05, 2.6, 2.42, 2.43, 2.53, 2.57, 2.85, 2.7, 2.53, 2.28, 2.2, 2.57, 2.89, 2.51, 2.47, 2.66, 2.06
             , 2.41, 2.65, 2.76, 2.43, 2.61, 2.57, 2.73, 2.17, 2.67, 2.05, 1.71, 2.32, 2.23, 2.76, 2.7, 2.13, 2.75, 2.2)

# Calculate the population mean and population standard deviation.
pop_mean <- mean(weights)
# Note: R's sd() function calculates the sample standard deviation (divides by n-1).
pop_std_dev <- sqrt(sum((weights - pop_mean)^2) / length(weights))

print("--- Population Statistics ---")
print("Population Mean (μ):")
print(pop_mean)
print("Population Standard Deviation (σ):")
print(pop_std_dev)
```

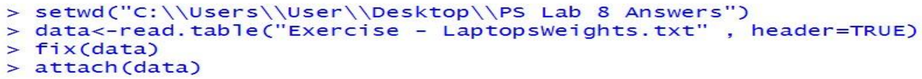


```
#02.)
num_samples <- 25
sample_size <- 6
sample_means <- numeric(num_samples)
sample_std_devs <- numeric(num_samples)

for (i in 1:num_samples) {
  # Use 'sample' command to draw a random sample with replacement.
  s <- sample(weights, sample_size, replace = TRUE)
  print(s)

  # Calculate and store the mean and standard deviation for each sample.
  sample_means[i] <- mean(s)
  print("Sample mean:")
  print(sample_means[i])
  sample_std_devs[i] <- sd(s)
  print(" sample standard deviation:")
  print(sample_std_devs[i])
}
```

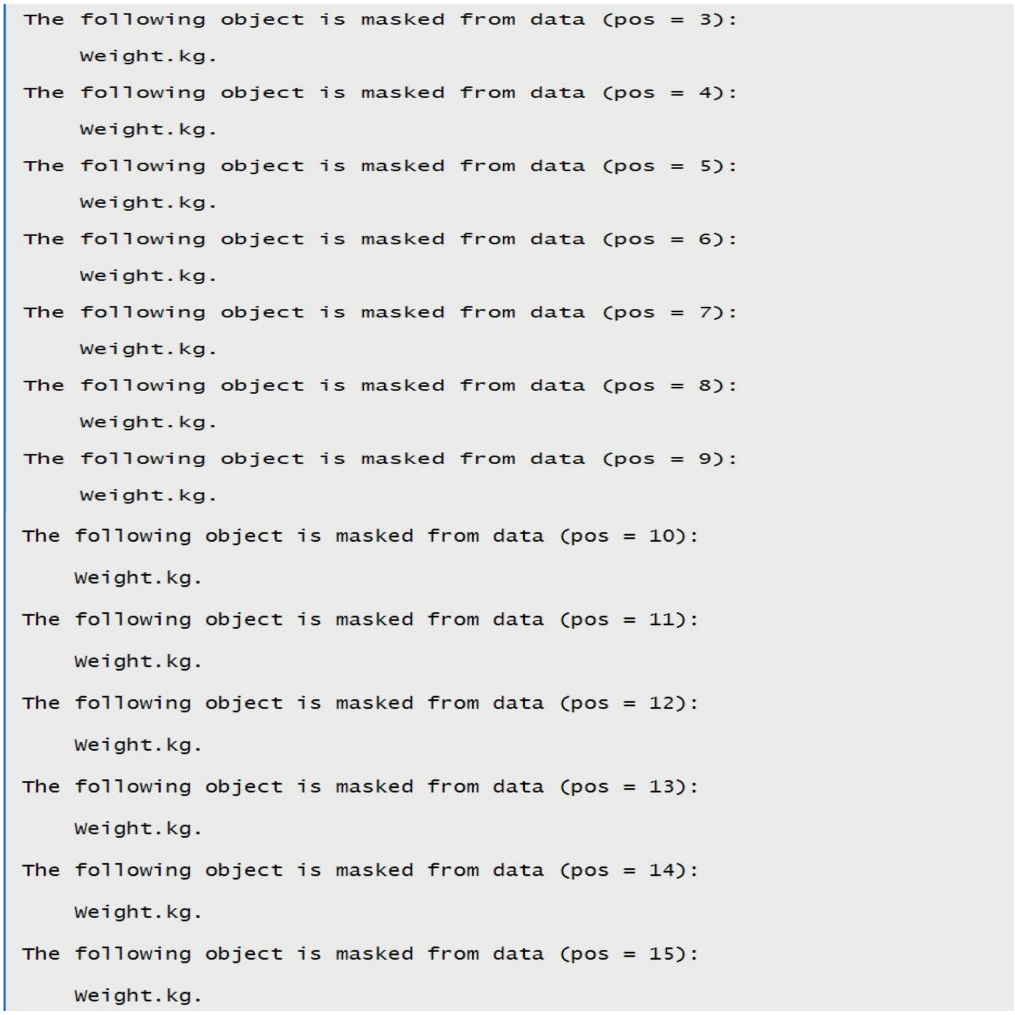


```
#03.)
mean_of_sample_means <- mean(sample_means)
std_dev_of_sample_means <- sd(sample_means)

print("--- Sample Statistics ---")

print("Mean of the 25 sample means:")
print(mean_of_sample_means)

print("Standard Deviation of the 25 sample means:")
print(std_dev_of_sample_means)

print("Theoretical Standard Error (σ/√n):")
print(pop_std_dev / sqrt(sample_size))

# State the relationships
print("--- Relationships ---")
print("The mean of the sample means is very close to the population mean (μ).")
print("The standard deviation of the sample means is close to the population standard deviation divided by the square root of the sample size
     (σ/√n).")
```

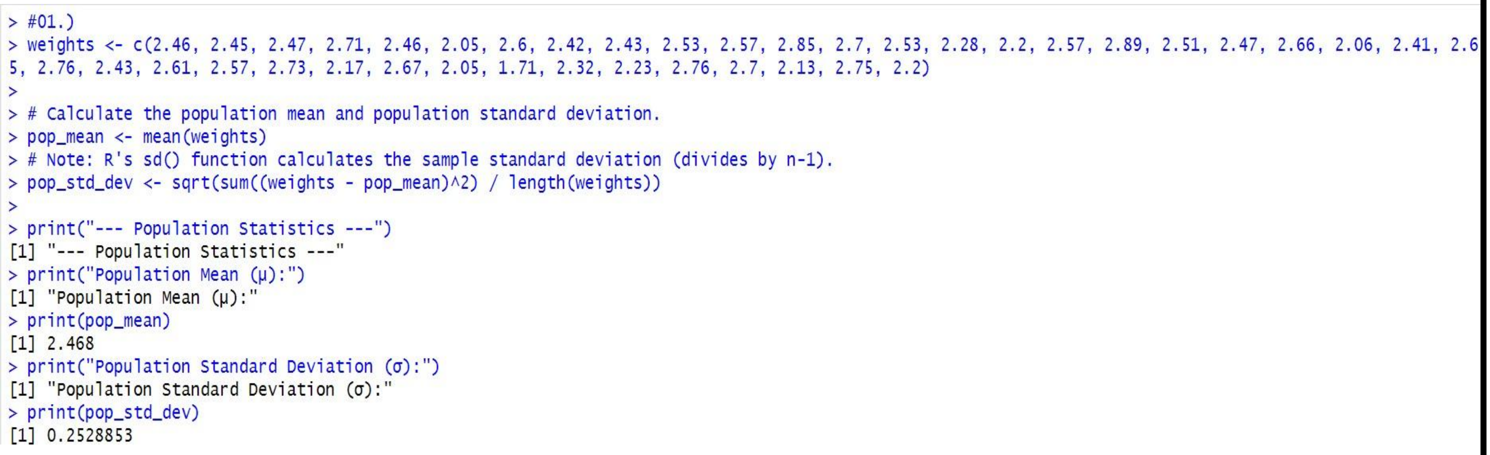
Console Output



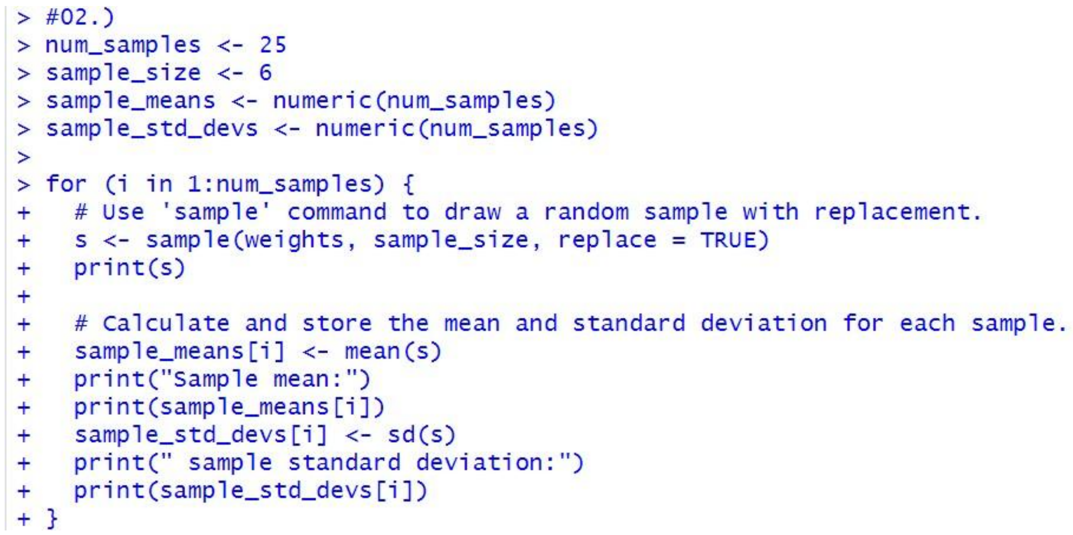
```
> setwd("C:\\Users\\User\\Desktop\\PS Lab 8 Answers")
> data<-read.table("Exercise - LaptopsWeights.txt" , header=TRUE)
> fix(data)
> attach(data)
```



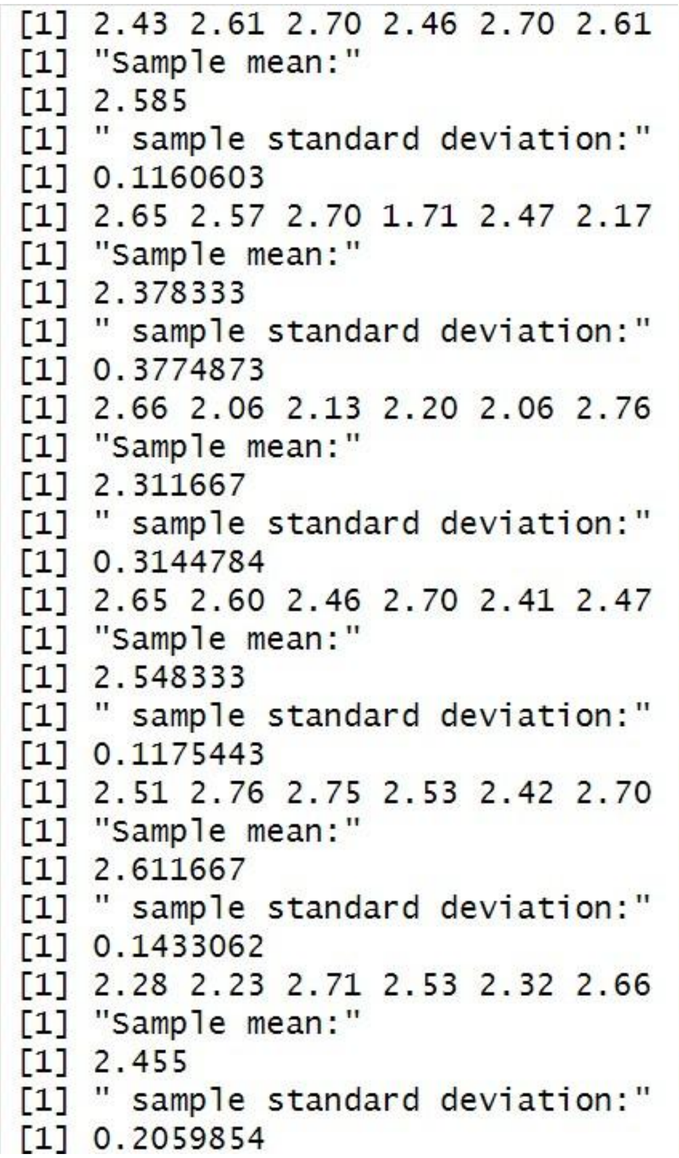
```
The following object is masked from data (pos = 3):

    Weight.kg.

The following object is masked from data (pos = 4):

    Weight.kg.

The following object is masked from data (pos = 5):

    Weight.kg.

The following object is masked from data (pos = 6):

    Weight.kg.

The following object is masked from data (pos = 7):

    Weight.kg.

The following object is masked from data (pos = 8):

    Weight.kg.

The following object is masked from data (pos = 9):

    Weight.kg.

The following object is masked from data (pos = 10):

    Weight.kg.

The following object is masked from data (pos = 11):

    Weight.kg.

The following object is masked from data (pos = 12):

    Weight.kg.

The following object is masked from data (pos = 13):

    Weight.kg.

The following object is masked from data (pos = 14):

    Weight.kg.

The following object is masked from data (pos = 15):

    Weight.kg.
```



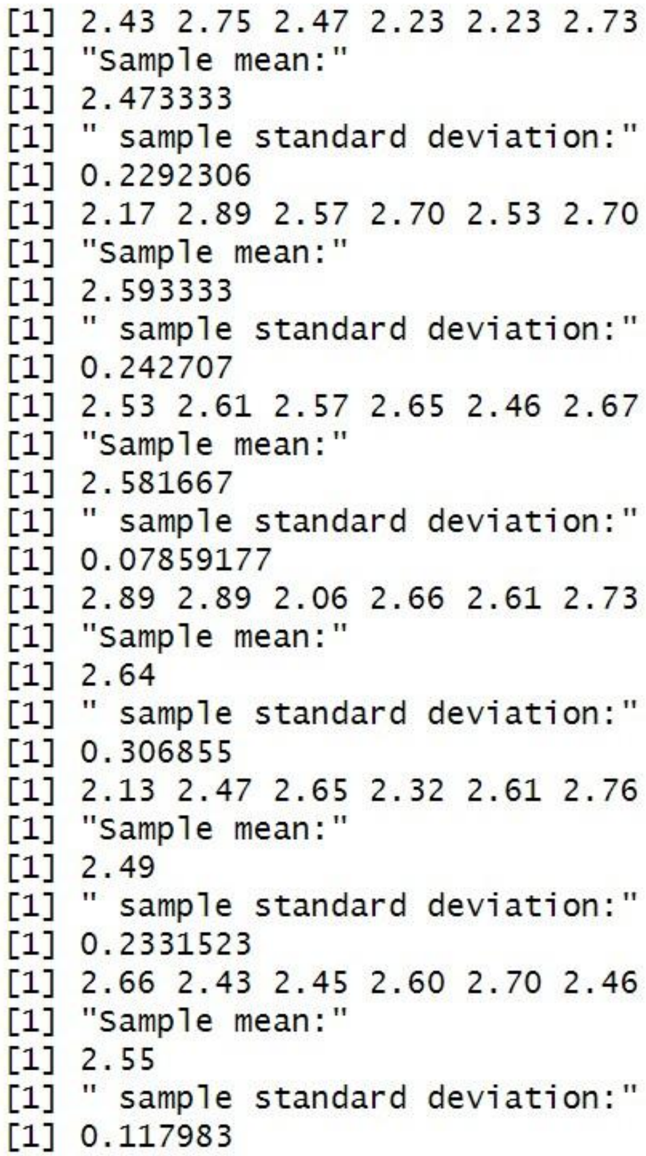
```
> #01.)
> weights <- c(2.46, 2.45, 2.47, 2.71, 2.46, 2.05, 2.6, 2.42, 2.43, 2.53, 2.57, 2.85, 2.7, 2.53, 2.28, 2.2, 2.57, 2.89, 2.51, 2.47, 2.66, 2.06, 2.41, 2.6
5, 2.76, 2.43, 2.61, 2.57, 2.73, 2.17, 2.67, 2.05, 1.71, 2.32, 2.23, 2.76, 2.7, 2.13, 2.75, 2.2)
>
> # Calculate the population mean and population standard deviation.
> pop_mean <- mean(weights)
> # Note: R's sd() function calculates the sample standard deviation (divides by n-1).
> pop_std_dev <- sqrt(sum((weights - pop_mean)^2) / length(weights))
>
> print("--- Population Statistics ---")
[1] "--- Population Statistics ---"
> print("Population Mean (μ):")
[1] "Population Mean (μ):"
> print(pop_mean)
[1] 2.468
> print("Population Standard Deviation (σ):")
[1] "Population Standard Deviation (σ):"
> print(pop_std_dev)
[1] 0.2528853
```



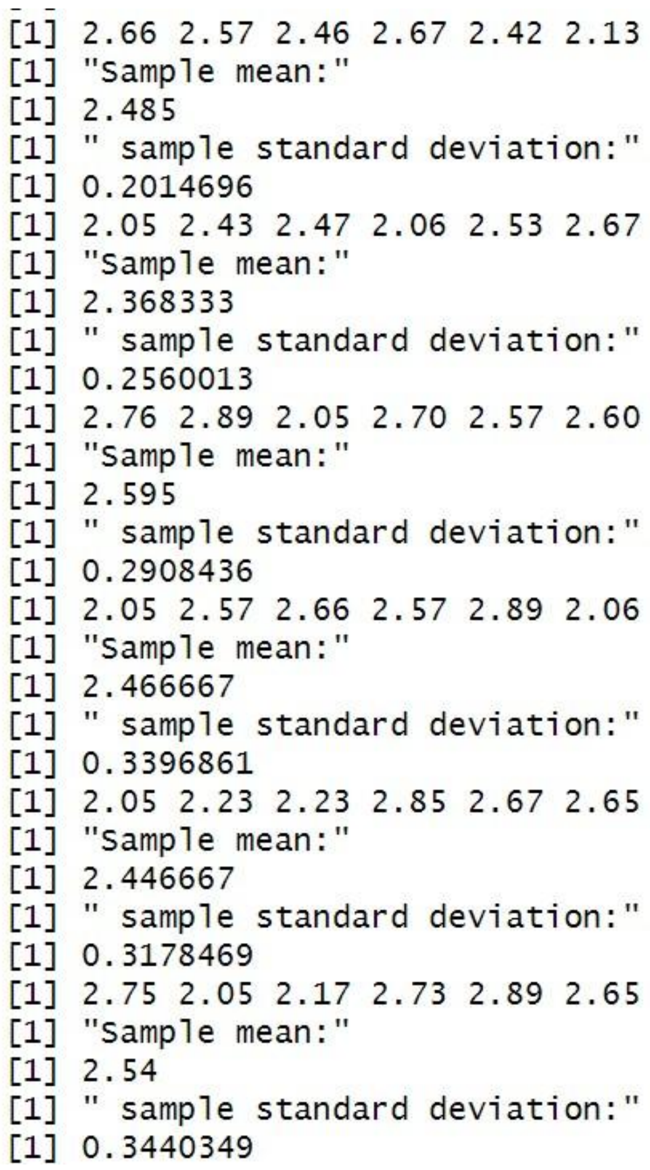
```
> #02.)
> num_samples <- 25
> sample_size <- 6
> sample_means <- numeric(num_samples)
> sample_std_devs <- numeric(num_samples)
>
> for (i in 1:num_samples) {
+   # Use 'sample' command to draw a random sample with replacement.
+   s <- sample(weights, sample_size, replace = TRUE)
+   print(s)
+
+   # Calculate and store the mean and standard deviation for each sample.
+   sample_means[i] <- mean(s)
+   print("Sample mean:")
+   print(sample_means[i])
+   sample_std_devs[i] <- sd(s)
+   print(" sample standard deviation:")
+   print(sample_std_devs[i])
+ }
```



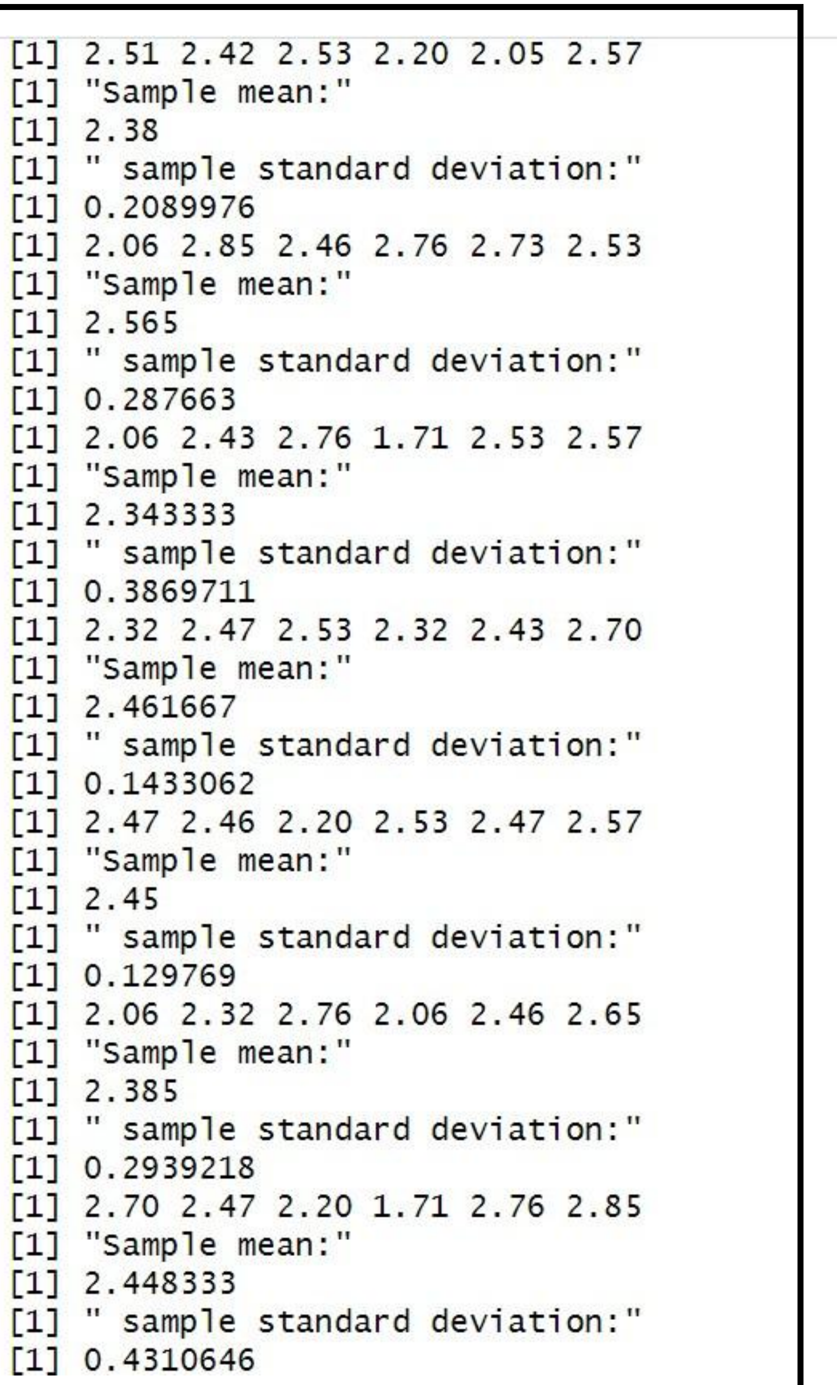
```
[1] 2.43 2.61 2.70 2.46 2.70 2.61
[1] "Sample mean:"
[1] 2.585
[1] " sample standard deviation:"
[1] 0.1160603
[1] 2.65 2.57 2.70 1.71 2.47 2.17
[1] "Sample mean:"
[1] 2.378333
[1] " sample standard deviation:"
[1] 0.3774873
[1] 2.66 2.06 2.13 2.20 2.06 2.76
[1] "Sample mean:"
[1] 2.311667
[1] " sample standard deviation:"
[1] 0.3144784
[1] 2.65 2.60 2.46 2.70 2.41 2.47
[1] "Sample mean:"
[1] 2.548333
[1] " sample standard deviation:"
[1] 0.1175443
[1] 2.51 2.76 2.75 2.53 2.42 2.70
[1] "Sample mean:"
[1] 2.611667
[1] " sample standard deviation:"
[1] 0.1433062
[1] 2.28 2.23 2.71 2.53 2.32 2.66
[1] "Sample mean:"
[1] 2.455
[1] " sample standard deviation:"
[1] 0.2059854
```



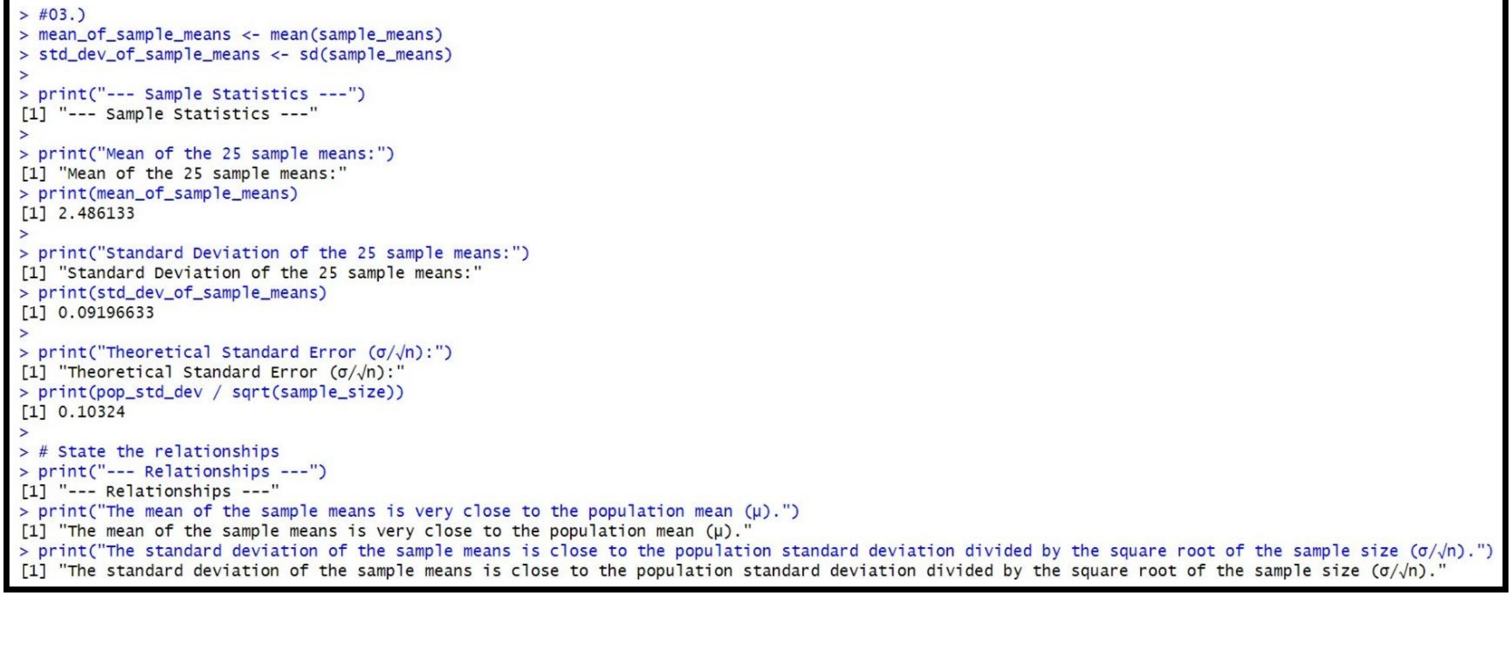
```
[1] 2.43 2.75 2.47 2.23 2.23 2.73
[1] "Sample mean:"
[1] 2.473333
[1] " sample standard deviation:"
[1] 0.2292306
[1] 2.17 2.89 2.57 2.70 2.53 2.70
[1] "Sample mean:"
[1] 2.593333
[1] " sample standard deviation:"
[1] 0.242707
[1] 2.53 2.61 2.57 2.65 2.46 2.67
[1] "Sample mean:"
[1] 2.581667
[1] " sample standard deviation:"
[1] 0.07859177
[1] 2.89 2.89 2.06 2.66 2.61 2.73
[1] "Sample mean:"
[1] 2.64
[1] " sample standard deviation:"
[1] 0.306855
[1] 2.13 2.47 2.65 2.32 2.61 2.76
[1] "Sample mean:"
[1] 2.49
[1] " sample standard deviation:"
[1] 0.2331523
[1] 2.66 2.43 2.45 2.60 2.70 2.46
[1] "Sample mean:"
[1] 2.55
[1] " sample standard deviation:"
[1] 0.117983
```



```
[1] 2.66 2.57 2.46 2.67 2.42 2.13
[1] "Sample mean:"
[1] 2.485
[1] " sample standard deviation:"
[1] 0.2014696
[1] 2.05 2.43 2.47 2.06 2.53 2.67
[1] "Sample mean:"
[1] 2.368333
[1] " sample standard deviation:"
[1] 0.2560013
[1] 2.76 2.89 2.05 2.70 2.57 2.60
[1] "Sample mean:"
[1] 2.595
[1] " sample standard deviation:"
[1] 0.2908436
[1] 2.05 2.57 2.66 2.57 2.89 2.06
[1] "Sample mean:"
[1] 2.466667
[1] " sample standard deviation:"
[1] 0.3396861
[1] 2.05 2.23 2.23 2.85 2.67 2.65
[1] "Sample mean:"
[1] 2.446667
[1] " sample standard deviation:"
[1] 0.3178469
[1] 2.75 2.05 2.17 2.73 2.89 2.65
[1] "Sample mean:"
[1] 2.54
[1] " sample standard deviation:"
[1] 0.3440349
```

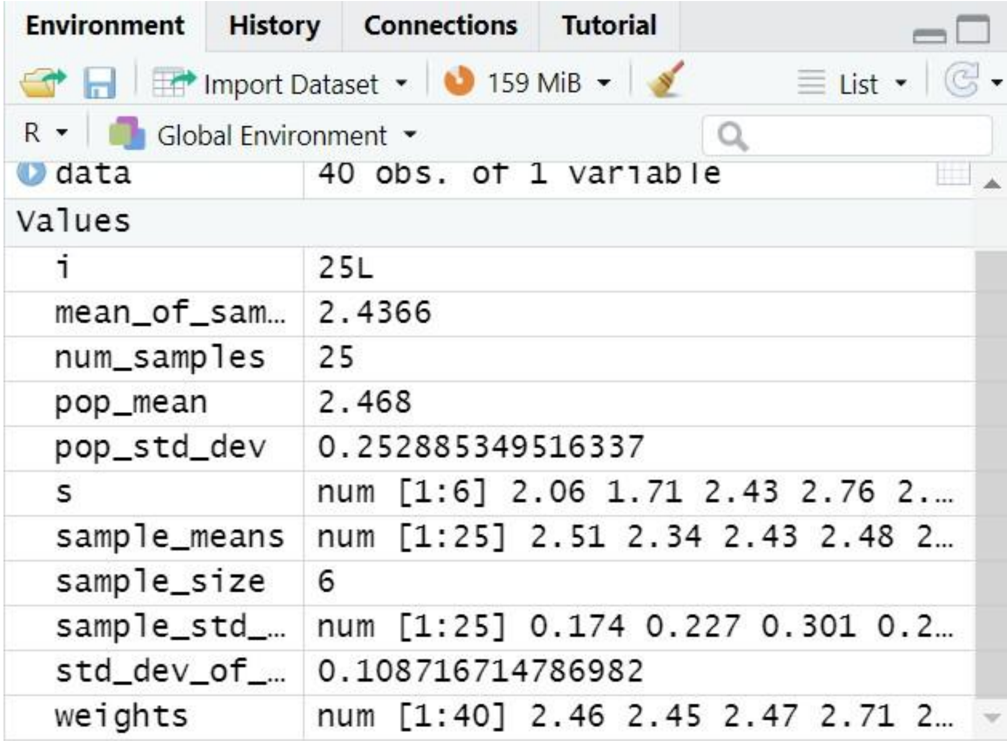


```
[1] 2.51 2.42 2.53 2.20 2.05 2.57
[1] "Sample mean:"
[1] 2.38
[1] " sample standard deviation:"
[1] 0.2089976
[1] 2.06 2.85 2.46 2.76 2.73 2.53
[1] "Sample mean:"
[1] 2.565
[1] " sample standard deviation:"
[1] 0.287663
[1] 2.06 2.43 2.76 1.71 2.53 2.57
[1] "Sample mean:"
[1] 2.343333
[1] " sample standard deviation:"
[1] 0.3869711
[1] 2.32 2.47 2.53 2.32 2.43 2.70
[1] "Sample mean:"
[1] 2.461667
[1] " sample standard deviation:"
[1] 0.1433062
[1] 2.47 2.46 2.20 2.53 2.47 2.57
[1] "Sample mean:"
[1] 2.45
[1] " sample standard deviation:"
[1] 0.129769
[1] 2.06 2.32 2.76 2.06 2.46 2.65
[1] "Sample mean:"
[1] 2.385
[1] " sample standard deviation:"
[1] 0.2939218
[1] 2.70 2.47 2.20 1.71 2.76 2.85
[1] "Sample mean:"
[1] 2.448333
[1] " sample standard deviation:"
[1] 0.4310646
```



```
> #03.)
> mean_of_sample_means <- mean(sample_means)
> std_dev_of_sample_means <- sd(sample_means)
>
> print("--- Sample Statistics ---")
[1] "--- Sample Statistics ---"
>
> print("Mean of the 25 sample means:")
[1] "Mean of the 25 sample means:"
> print(mean_of_sample_means)
[1] 2.486133
>
> print("Standard Deviation of the 25 sample means:")
[1] "Standard Deviation of the 25 sample means:"
> print(std_dev_of_sample_means)
[1] 0.09196633
>
> print("Theoretical Standard Error (σ/√n):")
[1] "Theoretical Standard Error (σ/√n):"
> print(pop_std_dev / sqrt(sample_size))
[1] 0.10324
>
> # State the relationships
> print("--- Relationships ---")
[1] "--- Relationships ---"
> print("The mean of the sample means is very close to the population mean (μ).")
[1] "The mean of the sample means is very close to the population mean (μ)."
> print("The standard deviation of the sample means is close to the population standard deviation divided by the square root of the sample size (σ/√n).")
[1] "The standard deviation of the sample means is close to the population standard deviation divided by the square root of the sample size (σ/√n)."
```

Global Environment



data	40 obs. of 1 variable
Values	
i	25L
mean_of_sam…	2.4366
num_samples	25
pop_mean	2.468
pop_std_dev	0.252885349516337
s	num [1:6] 2.06 1.71 2.43 2.76 2.…
sample_means	num [1:25] 2.51 2.34 2.43 2.48 2…
sample_size	6
sample_std_…	num [1:25] 0.174 0.227 0.301 0.2…
std_dev_of_…	0.108716714786982
weights	num [1:40] 2.46 2.45 2.47 2.71 2…