

IT2120 - Probability and Statistics

Lab Sheet 08

IT24103089 – Bandara MMSD

1. Calculate the population mean and population standard deviation of the laptop bag weights.

The screenshot displays the R Studio interface with the following components:

- Script Editor:** Contains R code for loading data and calculating statistics.
- Console:** Shows the output of the R commands.
- Data Editor:** A window showing the data loaded from the file, with columns for weight and other variables.
- Environment:** A window showing the objects in the environment, including the data frame and calculated statistics.

R Code in Script Editor:

```
1 setwd("C:\\Users\\user\\Downloads")
2 getwd()
3
4 ##Importing dataset
5 data <- read.table("Exercise - Laptopsweights.txt", header = TRUE)
6 fix(data)
7 attach(data)
8
9
10 ##Q1
11 popmn <- mean(weight.kg.)
12 popsd <- sd(weight.kg.)
13
14
```

Console Output:

```
> setwd("C:\\Users\\user\\Downloads")
> getwd()
[1] "C:/Users/user/Downloads"
> ##Importing dataset
> data <- read.table("Exercise - Laptopsweights.txt", header = TRUE)
> fix(data)
The following object is masked from data (pos = 3):
  weight.kg.
> ##Q1
> popmn <- mean(weight.kg.)
> popsd <- sd(weight.kg.)
```

Data Editor Window:

	Weight.kg.	var2	var3	var4	var5	var6
1	2.46					
2	2.45					
3	2.47					
4	2.71					
5	2.46					
6	2.05					
7	2.6					
8	2.42					
9	2.43					
10	2.53					
11	2.57					
12	2.85					
13	2.7					
14	2.53					
15	2.28					
16	2.2					
17	2.57					
18	2.89					
19	2.51					

Environment Window:

Object	Value
data	40 obs. of 1 variable
popmn	2.468
popsd	0.256106948813907
samples	NULL

2. Draw 25 random samples of size 6 (with replacement) and calculate the sample mean and sample standard deviation for each sample.

```

14
15 ##Q2
16 #First null vector to store sample data sets.
17 samples <- c()
18 n <- c()
19
20
21 for(i in 1:25){
22   s <- sample(weight.kg., 6, replace = TRUE)
23   samples <- cbind(samples, s)
24   n <- c(n, paste('s', i))
25 }
26
27 #Assign column names for each sample created
28 colnames(samples) = n
29
30 # considered the second argument as "2" we can calculate either mean/variance column wise
31 s.means <- apply(samples, 2, mean)
32 s.stds <- apply(samples, 2, sd)
33

```

Variable	Value
data	40 obs. of 1 variable
samples	num [1:6, 1:25] 2.6 2.51 2.57 2...
Values	
i	25L
n	chr [1:25] "s 1" "s 2" "s 3" "s 4"...
popmn	2.468
popstd	0.256106948813907
s	num [1:6] 2.43 2.76 2.61 2.46 2.7 ...
s.means	Named num [1:25] 2.56 2.48 2.48 2...
s.stds	Named num [1:25] 0.0833 0.2194 0.1...

3. Calculate the mean and standard deviation of the 25 sample means and state the relationship of them with true mean and true standard deviation.

```

34
35 ##Q3
36 samplemean <- mean(s.means)
37 samplestd <- sd(s.means)
38
39 #compare the population mean and mean of sample means.
40 popmn
41 samplemean
42
43
44 #compare the population std and std of sample means.
45 truesd = popstd/6
46 samplestd
47:1 (Top Level)

```

Variable	Value
data	40 obs. of 1 variable
samples	num [1:6, 1:25] 2.6 2.51 2.57 2...
Values	
i	25L
n	chr [1:25] "s 1" "s 2" "s 3" "s 4"...
popmn	2.468
popstd	0.256106948813907
s	num [1:6] 2.43 2.76 2.61 2.46 2.7 ...
s.means	Named num [1:25] 2.56 2.48 2.48 2...
s.stds	Named num [1:25] 0.0833 0.2194 0.1...
samplemean	2.4484
samplestd	0.119071912600623
truesd	0.0426844914689845

```

> #Assign column names for each sample created
> colnames(samples) = n
> s.means <- apply(samples, 2, mean)
> s.stds <- apply(samples, 2, sd)
> samplemean <- mean(s.means)
> samplestd <- sd(s.means)
> popmn
[1] 2.468
> samplemean
[1] 2.4484
> truesd = popstd/6
> samplestd
[1] 0.1190719
>

```