Sri Lanka Institute of Information Technology

Lab Submission
<Lab sheet 08>

<IT24102228>

<Welgama W. N. Y>

Probability and Statistics | IT2120

B.Sc. (Hons) in Information Technology

Exercise

1)

```
R 4.2.2 · C:/Users/it24102228/Desktop/IT24102228_Lab_08/
> laptopbagweights <- read.table("Exercise - LaptopsWeight
s.txt", header = TRUE)
> fix(laptopbagweights)
> attach(laptopbagweights)
The following object is masked from laptopbagweights (pos =
3):

    weight.kg.
```

```
> # Question 01
> #Calculate the population mean & population variance
> popmn <- mean(Nicotine) # population mean find
> popmn # population mean value
[1] 1.77425
> popvar <- var(Nicotine) # population variance find
> popvar # population variance value
[1] 0.1524558
>
```

2)

```
> # Question 02
> samp <- c()
> n <- c()
> for(i in 1:25){
+   s <- sample(Nicotine,6,replace = TRUE)
+   samp <- cbind(samp,s)
+   n <- c(n,paste("S",i))
+ }
> # Assign column names for each sample created. Names have
stored earlier under "n" variable.
> colnames(samp) = n
> # Using "apply" commands we can ask to calculate any func
```

```
> colnames(samp) = n
> # Using "apply" commands we can ask to calculate any func
tion such as mean, variance, etc. row wise or
> # column wise in a matrix.
> #Here, considering the second argument as "2" we can calc
ulate either mean/variance column wise
> #which stored earlier in "samples" variable which is a ma
trix
```

```
> s.means <- apply(samp,2,mean)
> s.means
     S 1      S 2      S 3      S 4      S 5      S 6
1.676667 1.685000 1.855000 1.470000 1.680000 1.593333
     S 7      S 8      S 9     S 10     S 11     S 12
1.738333 1.498333 1.845000 1.628333 1.965000 1.733333
    S 13     S 14     S 15     S 16     S 17     S 18
1.806667 2.035000 1.745000 2.076667 1.446667 1.701667
```

```
    S 19     S 20     S 21     S 22     S 23     S 24
1.828333 1.861667 1.780000 1.493333 1.845000 1.738333
    S 25
1.576667
> s.vars <- apply(samp,2, var)
> s.vars
       S 1        S 2        S 3        S 4        S 5
0.11626667 0.25883000 0.08423000 0.33804000 0.12568000
```

```
       S 6        S 7        S 8        S 9       S 10
0.11878667 0.12341667 0.19101667 0.12995000 0.33917667
      S 11       S 12       S 13       S 14       S 15
0.11147000 0.25382667 0.01402667 0.12471000 0.33323000
      S 16       S 17       S 18       S 19       S 20
0.14270667 0.44738667 0.15449667 0.04681667 0.05645667
      S 21       S 22       S 23       S 24       S 25
```

```
0.16060000 0.19158667 0.03543000 0.08645667 0.12234667
> s.sd <- apply(samp,2,sd)
> s.sd
      S 1       S 2       S 3       S 4       S 5
0.3409790 0.5087534 0.2902241 0.5814121 0.3545138
      S 6       S 7       S 8       S 9      S 10
0.3446544 0.3513071 0.4370545 0.3604858 0.5823888
     S 11      S 12      S 13      S 14      S 15
```

```
0.3338712 0.5038121 0.1184342 0.3531430 0.5772608
     S 16      S 17      S 18      S 19      S 20
0.3777654 0.6688697 0.3930606 0.2163716 0.2376061
     S 21      S 22      S 23      S 24      S 25
0.4007493 0.4377061 0.1882286 0.2940351 0.3497809
>
> #Following commands will calculate mean and variance of s
```

```
> #Following commands will calculate mean and variance of s
amples means stored in "s.means" variable.
> sampmean <- mean(s.means)
> sampmean # mean of the sample
[1] 1.732133
> sampvars <- var(s.means)

> sampvars # Variance of the sample
[1] 0.02768345
> sampsd <- sd(s.sd)
> sampsd # Standard derivation of the sample
[1] 0.1320739
```

3)

```
> # Question 03
> truevar = popvar / 6
> truevar
[1] 0.02540931
> sampvars
[1] 0.02768345
> truesd = sd(Nicotine) / 6
> truesd
[1] 0.06507599
> sampsd
[1] 0.1320739
>
```