Ishaan Taneja

Grading:
30 marks- Complete submission of Problem 1,2,3,6
70 marks- Problem 6

1. Each of you who has an elder sibling enter into google spreadsheet (link to be provided in class on Tuesday) the following:
   (a) your gender and height (in cm)
   (b) gender and height of your oldest sibling

   Solution: 1
   The data is entered in the `Google spreadsheet`.

2. Each of you survey your parent(s) who have an elder sibling and enter into google spreadsheet (link to be provided in class on Tuesday) the following:
   (a) parent gender and height (in cm)
   (b) gender and height of parent's oldest sibling

   Solution: 2
   The data is entered in the `Google spreadsheet`.

3. Simulate 1000 samples from Normal$(0, 1)$ in `R`. Implement `R`-codes to do the following:
   (a) Assume that variance known to be 1. Find a 95% confidence interval for the mean μ.
   (b) Assume that variance unknown to be 1. Find a 95% confidence interval for the mean μ.

   Solution: 3

```
> n=1000
> sigma=1
> norm_sample=rnorm(n)
```

   (a) When variance is known to be 1.

```
> cfin=function(x){
+   alpha=0.95
+   z=qnorm((1-alpha)/2,lower.tail = FALSE)
+   sd_x=sqrt(1/length(x))
+   return(c(mean(x)-z*sd_x,mean(x)+z*sd_x))
+ }
> lower=cfin(norm_sample)[1]
> upper=cfin(norm_sample)[2]
> cat("The 95% confidence interval for the mean is (",lower,",",upper,")")
```

```
The 95% confidence interval for the mean is ( -0.06596337 , 0.05799564 )
```

(b) When variance is unknown to be 1.

```
> cifn_t=function(x){
+   alpha=0.95
+   t_Score=qt((1-alpha)/2,length(x)-1,lower.tail = FALSE)
+   sd_x=sqrt(var(x)/length(x))
+   return(c(mean(x)-t_Score*sd_x,mean(x)+t_Score*sd_x))
+ }
> lower=cifn_t(norm_sample)[1]
> upper=cifn_t(norm_sample)[2]
> cat("The 95% confidence interval for the mean is (",lower,",",upper,")")
```

```
The 95% confidence interval for the mean is ( -0.06581467 , 0.05784694 )
```

4. Implement via an R-code to perform 100 trials of question 3. Compute the following

   (a) In both cases (known and unknown variance) find the number of trials in which the intervals contain the true mean.

   (b) Plot the difference of the length of intervals observed in both cases (known and unknown variance) across trials.

Solution: 4

(a) **Case 1:** The variance is known.

```
> true_mean=0
> count=0
> length_normal=c()
> length_t=c()
> for (i in 1:100){
+   x=rnorm(1000)
+   lower=cfin(x)[1]
+   upper=cfin(x)[2]
+   length_normal=append(length_normal,upper-lower)
+   if((true_mean>lower)&(true_mean<upper)){
+     count=count+1
+   }
+ }
> cat('Number of times true mean of each trial lies within the confidence interval:'
+     ,count,'\n')
```

```
Number of times true mean of each trial lies within the confidence interval: 95
```

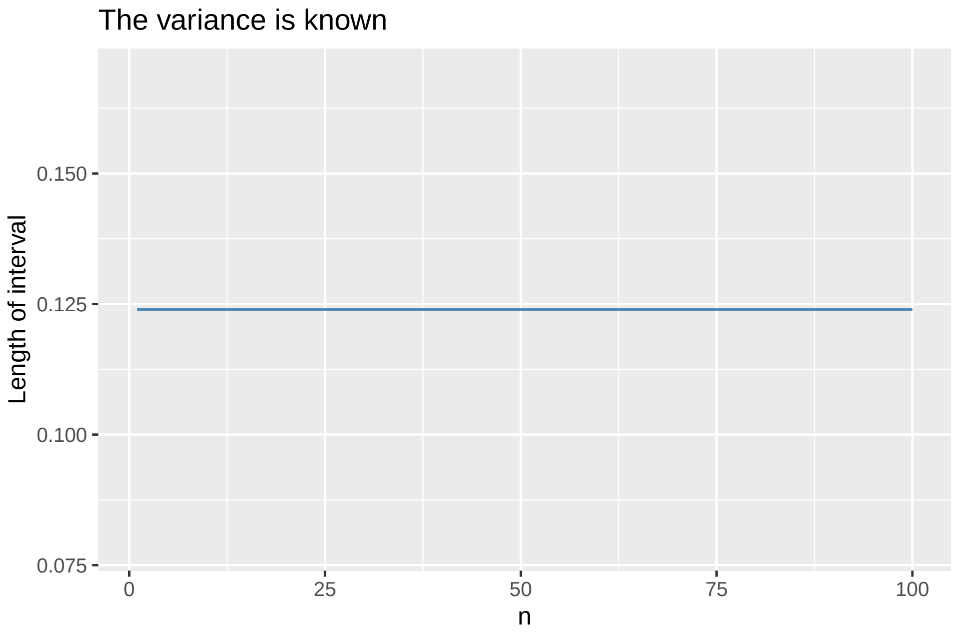
Case 2: The variance is unknown.

```
> true_mean=0
> count=0
> for (i in 1:100){
+   x=rnorm(1000)
+   lower=cifn_t(x)[1]
+   upper=cifn_t(x)[2]
+   length_t=append(length_t,upper-lower)
+   if((true_mean>lower)&(true_mean<upper)){
+     count=count+1
+   }
+ }
> cat('Number of times true mean of each trial lies within the confidence interval:'
+     ,count,'\n')
```

```
Number of times true mean of each trial lies within the confidence interval: 96
```

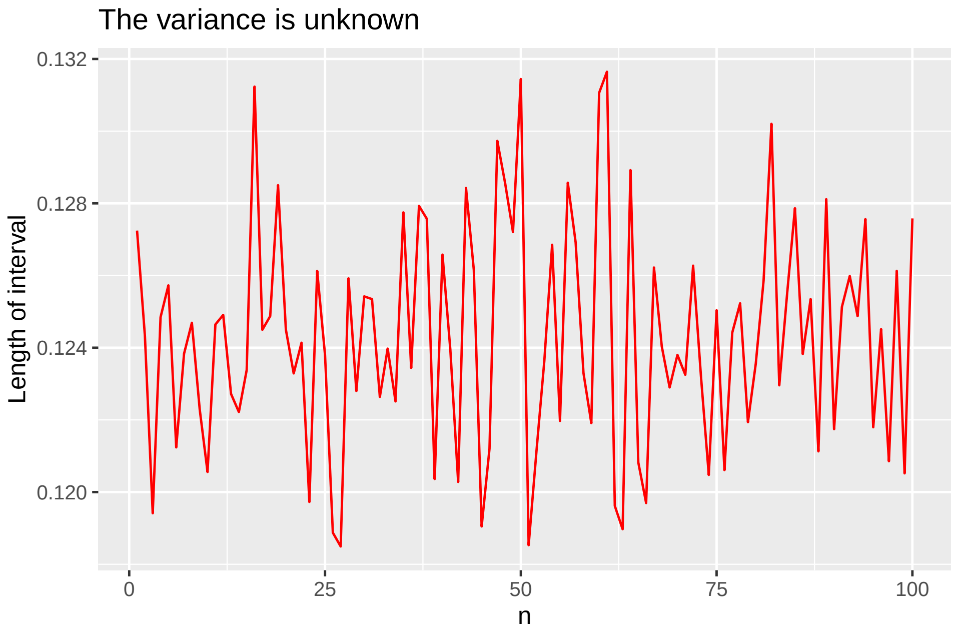
(b)

```
> n=1:100
> df=data.frame(n,length_normal,length_t)
> library(ggplot2)
> ggplot(df,aes(x=n))+geom_line(aes(y=length_normal),colour="steelblue")+
+   labs(x='n',y='Length of interval')+ggtitle("The variance is known")
```



Therefore, in case of known variance, the length of interval remains the constant for each trial.

```
> ggplot(df,aes(x=n))+geom_line(aes(y=length_t),colour="red")+
+   labs(x='n',y='Length of interval')+ggtitle("The variance is unknown")
```



Therefore, in case of unknown variance, the length of interval keep varying across the trials.

5. Cracker-Free-rang-dal wants to understand the noise level of firecracker 10000 strip. Measuring the noise level of a random sample of 12 crackers, it gets the following data (in decibels).

   94.0, 98.6, 96.8, 95.5, 93.8, 95.6, 99.3, 95.8, 93.9, 90.2, 91.0, 93.9

   Find a 95% confidence interval for the average noise level of such crackers. Do not round the final answer. Enter the data with 1 decimal place.
   Solution: 5

```
> noise_level=c(94.0,98.6,96.8,95.5,93.8,95.6,99.3,95.8,93.9,90.2,91.0,93.9)
> S=sd(noise_level)
> t_score=qt((1-0.95)/2,df=length(noise_level)-1,lower.tail = FALSE)
> lower=mean(noise_level)-t_score*S/sqrt(length(noise_level))
> upper=mean(noise_level)+t_score*S/sqrt(length(noise_level))
> cat("The 95% confidence interval for the average noise level of crackers is
+     (",lower,",",upper,")")

The 95% confidence interval for the average noise level of crackers is
    ( 93.1572 , 96.57613 )
```

6. Use the inbuilt `iris` data set in `R`. For each of the species `setosa, versicolor, virginica` find a 95% confidence interval for: `Sepal.Length`, `Sepal.Length`, `Sepal.Width`, `Petal.Length`, and `Petal.Width`
   Solution: 6

```
> cifn=function(x){
+   alpha=0.95
+   t=qt((1-alpha)/2,df=length(x)-1,lower.tail = FALSE)
+   sd_x=sd(x)*sqrt(1/length(x))
+   paste(mean(x)-t*sd_x,mean(x)+t*sd_x)
+ }
> data("iris")
> setosa=iris[iris$Species=="setosa",]
> versicolor=iris[iris$Species=="versicolor",]
> virginica=iris[iris$Species=="virginica",]
> species=c("Sepal_Length","Sepal_Width","Petal_Length","Petal_Width")
> print("Confidence intervals for Setosa:")

[1] "Confidence intervals for Setosa:"

> for(i in 1:4){
+   print(paste(species[i],":",(cifn(setosa[,i]))))
+ }

[1] "Sepal_Length : 4.90582353929926 5.10617646070074"
[1] "Sepal_Width : 3.32027109827236 3.53572890172764"
[1] "Petal_Length : 1.41264523828751 1.51135476171249"
[1] "Petal_Width : 0.216049746883627 0.275950253116373"

> print("Confidence intervals for Versicolor:")

[1] "Confidence intervals for Versicolor:"

> for(i in 1:4){
+   print(paste(species[i],":",(cifn(versicolor[,i]))))
+ }

[1] "Sepal_Length : 5.78930578310683 6.08269421689317"
[1] "Sepal_Width : 2.68081950323602 2.85918049676398"
[1] "Petal_Length : 4.12645277790548 4.39354722209452"
[1] "Petal_Width : 1.26979931017686 1.38220068982314"

> print("Confidence intervals for Virginica:")

[1] "Confidence intervals for Virginica:"

> for(i in 1:4){
+   print(paste(species[i],":",(cifn(virginica[,i]))))
+ }
```

```
[1] "Sepal_Length : 6.4072850191175 6.7687149808825"
[1] "Sepal_Width : 2.88234746952339 3.06565253047661"
[1] "Petal_Length : 5.39515326292752 5.70884673707248"
[1] "Petal_Width : 1.94794531782633 2.10405468217367"
```