Probability and Statistics Homework

Prepared by: Md Zahid Hasan

Date: 2022-12-21

Firstly, we will import packages necessary for writing R script.

```
install.packages(c(
  "Sleuth2", "dplyr", "DataExplorer",
  "Hmisc", "pastecs", "UsingR",
  "ggplot2", "ggfortify", "scales",
  "plotly", "pracma", "fitdistrplus"),
  contriburl = contrib.url(
    "https://cran.r-project.org/bin/windows/contrib/4.0/R.rsp_0.44.0.zip"))
```

Next, we will import the libraries.

```
library(Sleuth2)
library(dplyr)
library(DataExplorer)
library(Hmisc)
library(pastecs)
library(UsingR)
library(ggplot2)
library(ggfortify)
library(scales)
library(plotly)
library(pracma)
library(fitdistrplus)
```

Task - 1 : (1pt) Load the data set and separate the data into the two observed parts. Provide an overview of each of them by estimating the expectation, variance and median of the corresponding distribution and briefly describing the nature of the studied problem.

Answer.

For this homework we will use case0101 of library Sleuth2.

It contains data from an experiment concerning the effects of intrinsic and extrinsic motivation on creativity. Subjects with considerable experience in creative writing were randomly assigned to one of two treatment groups.

```
Sleuth2::case0101
```

```
##    Score Treatment
## 1    5.0 Extrinsic
## 2    5.4 Extrinsic
## 3    6.1 Extrinsic
## 4   10.9 Extrinsic
## 5   11.8 Extrinsic
## 6   12.0 Extrinsic
## 7   12.3 Extrinsic
## 8   14.8 Extrinsic
## 9   15.0 Extrinsic
```

```
## 10   16.8 Extrinsic
## 11   17.2 Extrinsic
## 12   17.2 Extrinsic
## 13   17.4 Extrinsic
## 14   17.5 Extrinsic
## 15   18.5 Extrinsic
## 16   18.7 Extrinsic
## 17   18.7 Extrinsic
## 18   19.2 Extrinsic
## 19   19.5 Extrinsic
## 20   20.7 Extrinsic
## 21   21.2 Extrinsic
## 22   22.1 Extrinsic
## 23   24.0 Extrinsic
## 24   12.0 Intrinsic
## 25   12.0 Intrinsic
## 26   12.9 Intrinsic
## 27   13.6 Intrinsic
## 28   16.6 Intrinsic
## 29   17.2 Intrinsic
## 30   17.5 Intrinsic
## 31   18.2 Intrinsic
## 32   19.1 Intrinsic
## 33   19.3 Intrinsic
## 34   19.8 Intrinsic
## 35   20.3 Intrinsic
## 36   20.5 Intrinsic
## 37   20.6 Intrinsic
## 38   21.3 Intrinsic
## 39   21.6 Intrinsic
## 40   22.1 Intrinsic
## 41   22.2 Intrinsic
## 42   22.6 Intrinsic
## 43   23.1 Intrinsic
## 44   24.0 Intrinsic
## 45   24.3 Intrinsic
## 46   26.7 Intrinsic
## 47   29.7 Intrinsic
```

At first, we store the case0101 data in mc data-set.

We know, Expectation/Mean,

$$E[X] = \sum x_i P_i$$

also variance,

$$S^2 = \frac{\sum (x_i - \bar{x})^2}{n-1}$$

finally, median,

$$Med(x) = \begin{cases} X[\frac{n+1}{2}] & \text{: if n is odd} \\ \frac{X[\frac{n+1}{2}]+X[\frac{n}{2}]}{2} & \text{: if n is even} \end{cases}$$

But here we use sample mean to calculate mean value.

$$\bar{x} = \frac{\sum x_i}{n}$$

We can use mean(), var(), median() to get the mean, variance and median values for mc data-set.

We used summary(), dim(), describe(), stat.desc(), attributes() for displaying more information.

```
mc <- case0101

mean(mc$Score)
```

```
## [1] 17.85532
```

```
var(mc$Score)
```

```
## [1] 27.4347
```

```
median(mc$Score)
```

```
## [1] 18.7
```

```
summary(mc)
```

```
##      Score            Treatment
##  Min.   : 5.00   Extrinsic:23
##  1st Qu.:14.90   Intrinsic:24
##  Median :18.70
##  Mean   :17.86
##  3rd Qu.:21.25
##  Max.   :29.70
```

```
dim(mc)
```

```
## [1] 47  2
```

```
describe(mc)
```

```
## mc
##
##  2  Variables      47  Observations
## --------------------------------------------------------------------------------
## Score
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##       47        0       39    0.999    17.86     5.82     7.54    11.92
##      .25      .50      .75      .90      .95
##    14.90    18.70    21.25    23.46    24.21
##
```

```
## lowest :  5.0  5.4  6.1 10.9 11.8, highest: 23.1 24.0 24.3 26.7 29.7
## --------------------------------------------------------------------------------
## Treatment
##        n  missing distinct
##       47        0        2
##
## Value      Extrinsic Intrinsic
## Frequency         23        24
## Proportion     0.489     0.511
## --------------------------------------------------------------------------------
```

```
stat.desc(mc)
```

```
##                     Score Treatment
## nbr.val       47.0000000        NA
## nbr.null       0.0000000        NA
## nbr.na         0.0000000        NA
## min            5.0000000        NA
## max           29.7000008        NA
## range         24.7000008        NA
## sum          839.2000074        NA
## median        18.7000008        NA
## mean          17.8553193        NA
## SE.mean        0.7640138        NA
## CI.mean.0.95   1.5378799        NA
## var           27.4347004        NA
## std.dev        5.2378145        NA
## coef.var       0.2933476        NA
```

```
attributes(mc)
```

```
## $row.names
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15"
## [16] "16" "17" "18" "19" "20" "21" "22" "23" "24" "25" "26" "27" "28" "29" "30"
## [31] "31" "32" "33" "34" "35" "36" "37" "38" "39" "40" "41" "42" "43" "44" "45"
## [46] "46" "47"
##
## $names
## [1] "Score"     "Treatment"
##
## $class
## [1] "data.frame"
```

In the next step, we will filter the data-set for Intrinsic Treatment type(In) and similarly, mean, variance, median and other information for In.

```
In <- filter(mc, mc$Treatment == "Intrinsic")

mean(In$Score)
```

```
## [1] 19.88333
```

```r
var(In$Score)
```

```
## [1] 19.70928
```

```r
median(In$Score)
```

```
## [1] 20.4
```

```r
describe(In$Score)
```

```
## In$Score
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##       24        0       23        1    19.88     5.05    12.13    13.11
##      .25      .50      .75      .90      .95
##    17.43    20.40    22.30    24.21    26.34
##
## lowest : 12.0 12.9 13.6 16.6 17.2, highest: 23.1 24.0 24.3 26.7 29.7
```

```r
stat.desc(In$Score)
```

```
##      nbr.val     nbr.null       nbr.na          min          max        range
##   24.0000000    0.0000000    0.0000000   12.0000000   29.7000008   17.7000008
##          sum       median         mean      SE.mean CI.mean.0.95          var
##  477.2000027   20.3999996   19.8833334    0.9062118    1.8746420   19.7092762
##      std.dev     coef.var
##    4.4395131    0.2232781
```

```r
summary(In$Score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12.00   17.43   20.40   19.88   22.30   29.70
```

In the next step, we will filter the data-set for Extrinsic Treatment type(ex) and similarly, mean, variance, median and other information for ex.

```r
ex <- filter(mc, mc$Treatment == "Extrinsic")

mean(ex$Score)
```

```
## [1] 15.73913
```

```r
var(ex$Score)
```

```
## [1] 27.58976
```

```
median(ex$Score)
```

```
## [1] 17.2
```

```
describe(ex$Score)
```

```
## ex$Score
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##       23        0       21    0.999    15.74    5.906     5.47     7.06
##      .25      .50      .75      .90      .95
##    12.15    17.20    18.95    21.10    22.01
##
## lowest :  5.0  5.4  6.1 10.9 11.8, highest: 19.5 20.7 21.2 22.1 24.0
```

```
stat.desc(ex$Score)
```

```
##      nbr.val     nbr.null       nbr.na          min          max        range
##   23.0000000    0.0000000    0.0000000    5.0000000   24.0000000   19.0000000
##          sum       median         mean      SE.mean CI.mean.0.95          var
##  362.0000048   17.2000008   15.7391306    1.0952420    2.2713928   27.5897645
##      std.dev     coef.var
##    5.2525960    0.3337285
```

```
summary(ex$Score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00   12.15   17.20   15.74   18.95   24.00
```

Task - 2 : (1pt) For each group separately, estimate the density and distribution function of the data using the histogram and the empirical distribution function.

Answer.

We know, density or mass is,

$$_{px}(x) = P(X = x_i)$$

and cumulative distribution function is,

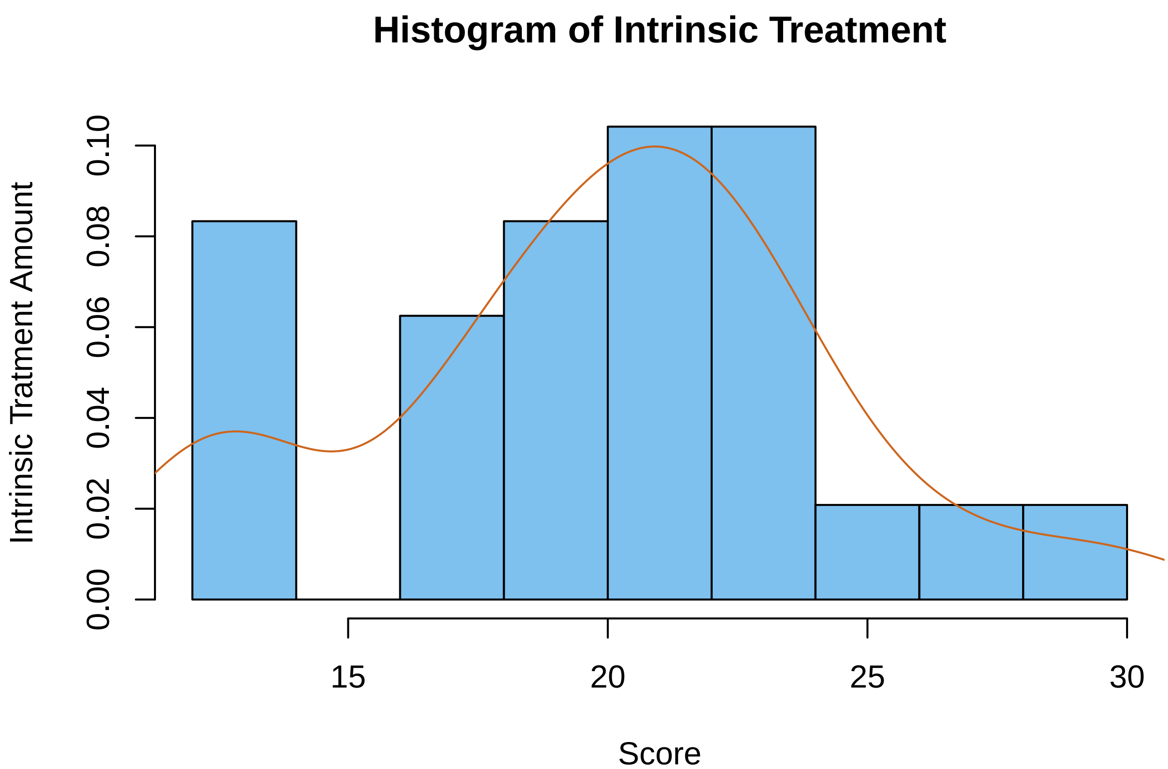
$$F(x) = \sum P(X \leq x_i)$$

In this task, we have to generate density and distribution functions for both Intrinsic => In and Extrinsic => ex data-sets. Lets start with In.

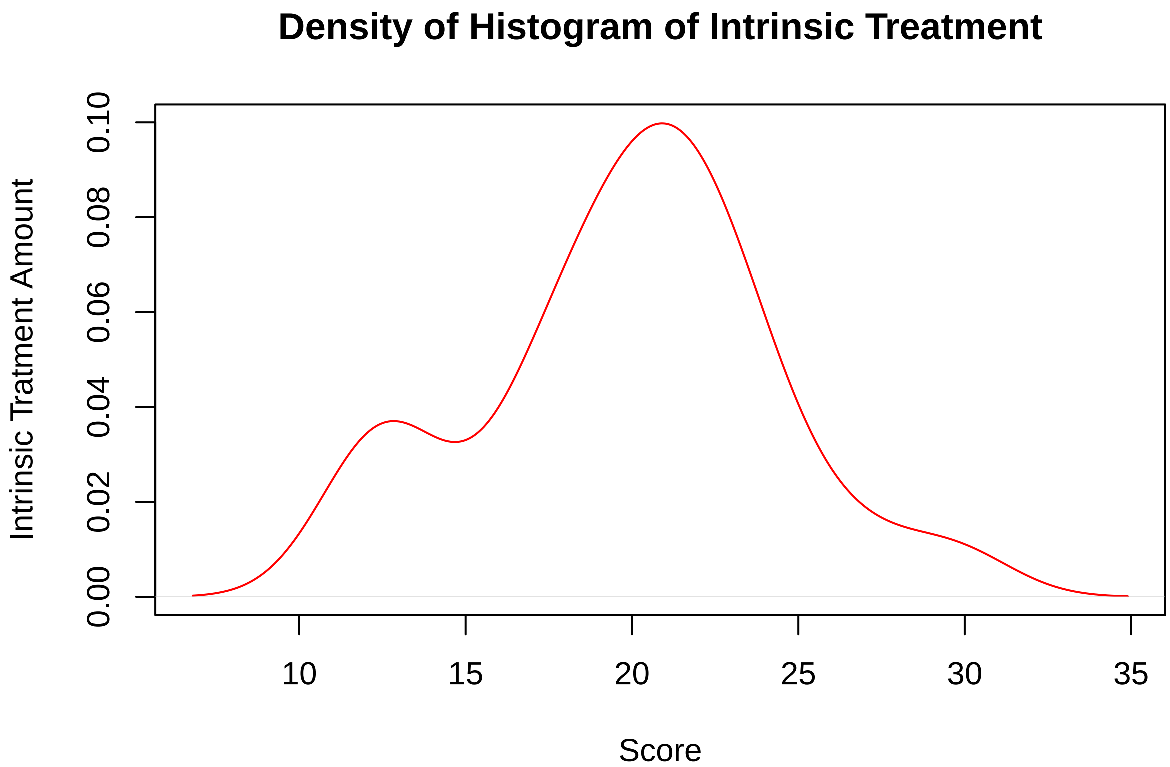
First generate histogram then the curve which shows the density of In.We use hist() function to generate histograms and lines() to show the density along histogram.We use density() function to get density of data-set.

Next generate the ecdf() function and from that plot the distribution of In. ggfortify::ggistribution() is used to show continuous distribution increase. ggplot2::labs() function is used to specify the names and other parameters of graphs.

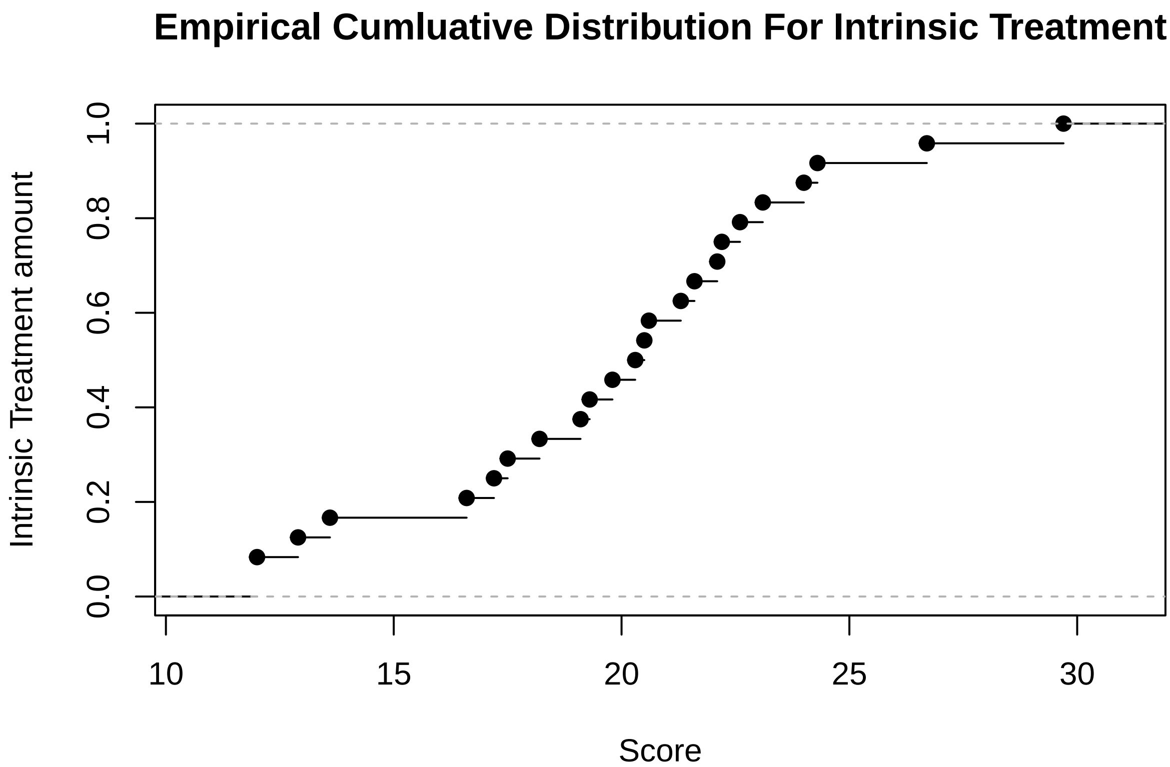
```r
hist(In$Score ,
     col = "skyblue2",
     main = "Histogram of Intrinsic Treatment",
     ylab = "Intrinsic Tratment Amount",
     xlab = "Score",
     plot = TRUE,
     breaks = 8,
     probability = T)
lines(density(In$Score),
      col = "chocolate3")
```



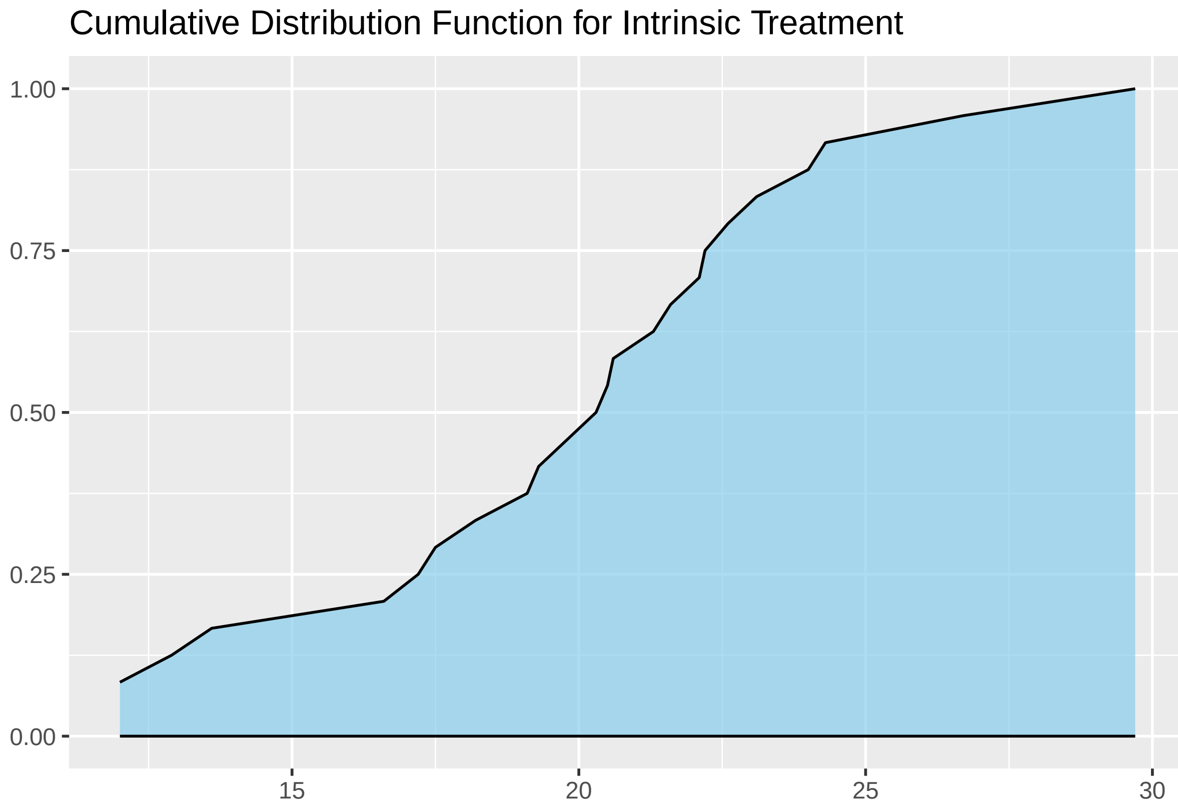

```r
plot(density(In$Score),
     frame = TRUE,
     col = "red",
     main = "Density of Histogram of Intrinsic Treatment",
     ylab = "Intrinsic Tratment Amount",
     xlab = "Score")
```




```
#ecdf graph generating
In.ecdf = ecdf(In$Score)
plot(In.ecdf,
     xlab = "Score",
     main = "Empirical Cumluative Distribution For Intrinsic Treatment",
     ylab = "Intrinsic Treatment amount")
```



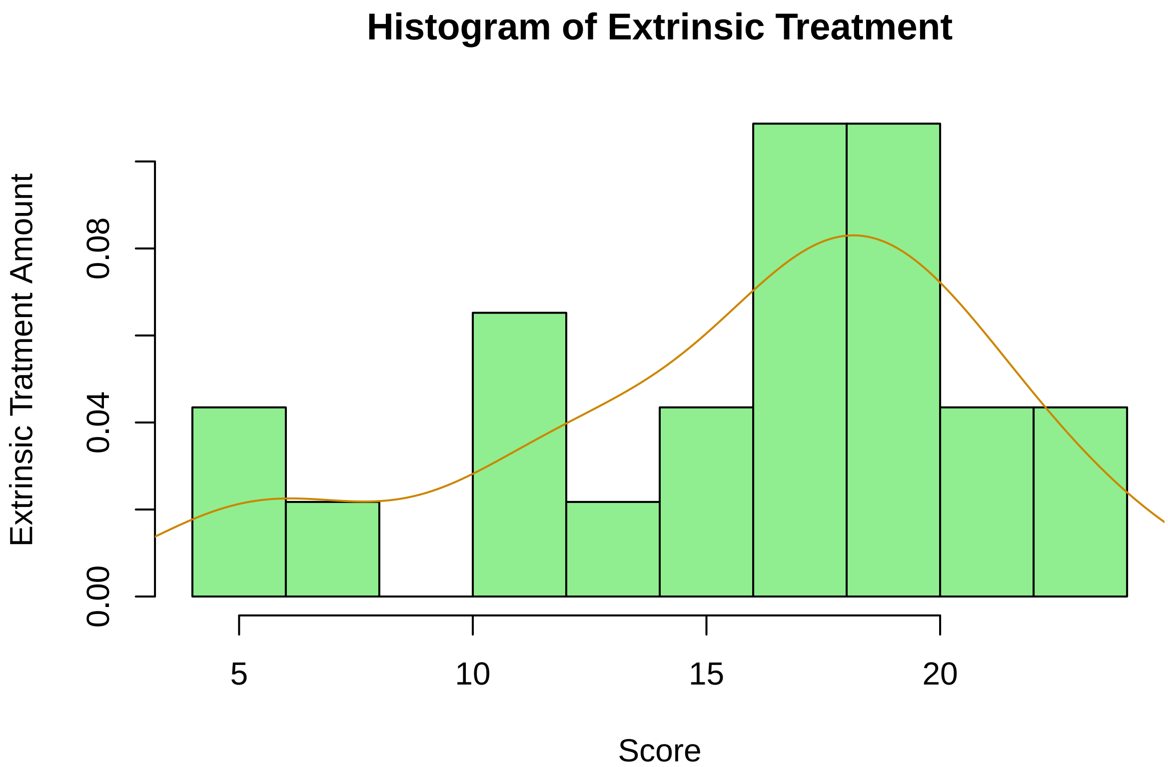

```
ggfortify::ggdistribution(In.ecdf,
                          In$Score,
                          colour = "black",
                          alpha = 0.7,
                          fill = "skyblue") +
  ggplot2::labs(
    title =
      "Cumulative Distribution Function for Intrinsic Treatment") +
  ggplot2::xlab ("Score") +
  ggplot2::ylab ("Frequency")
```



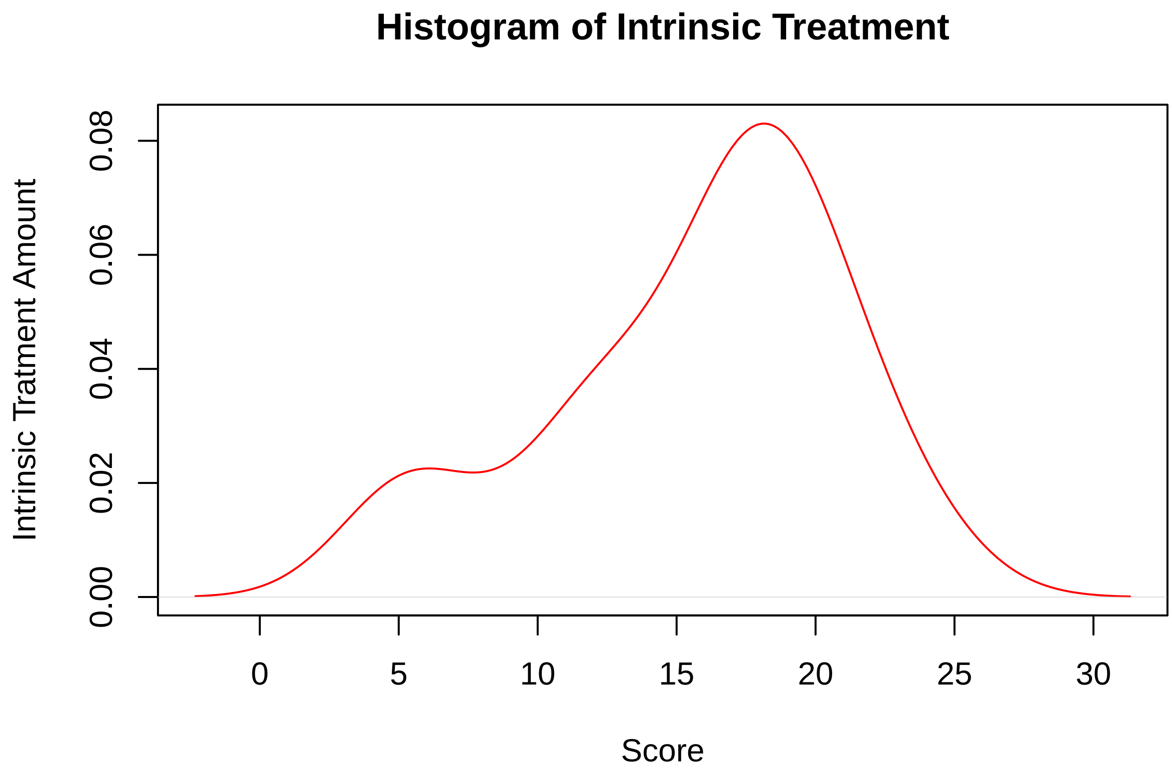
Then, generate histogram then the curve which shows the density of ex.

Next generate the ecdf function and from that plot the distribution of ex.

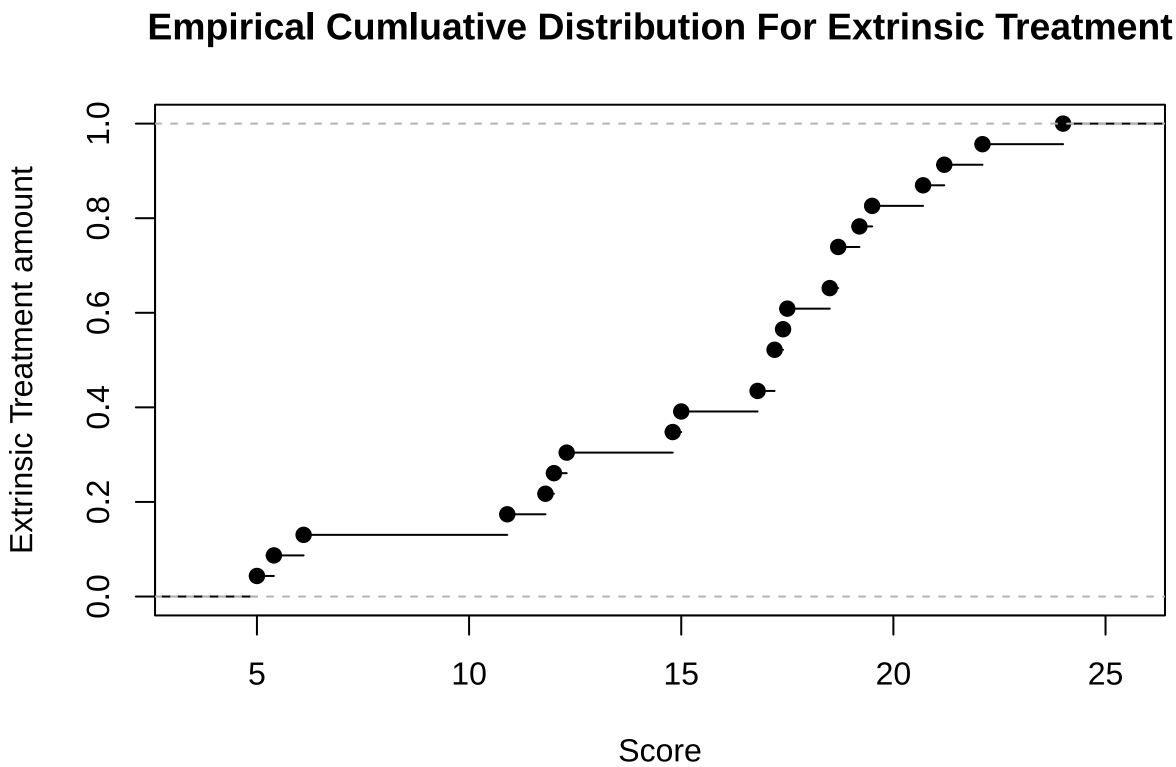
```
hist(ex$Score,
     col = "lightgreen",
     main = "Histogram of Extrinsic Treatment",
     ylab = "Extrinsic Tratment Amount",
     xlab = "Score",
     plot = TRUE,
     breaks = 8,
     probability = T)
lines(density(ex$Score),
      col = "orange3")
```



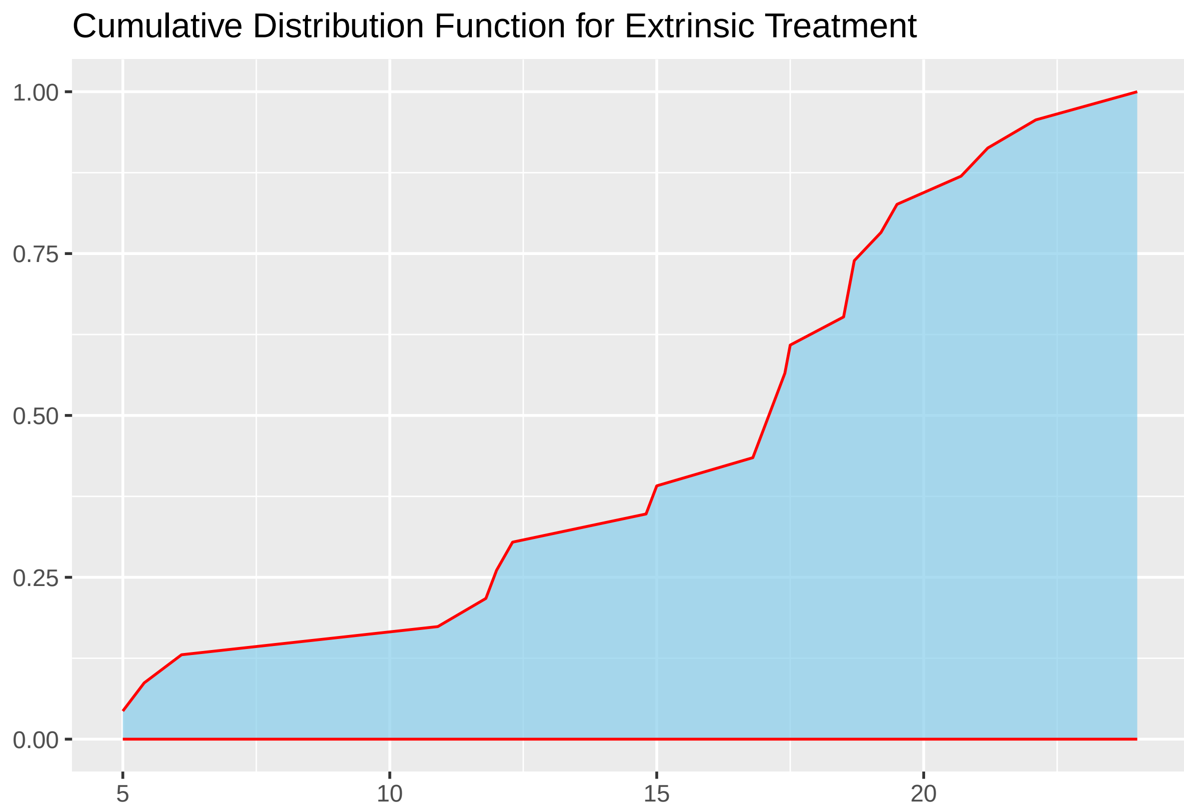

```
plot(density(ex$Score),
     frame = TRUE,
     col = "red",
     main = "Histogram of Intrinsic Treatment",
     ylab = "Intrinsic Tratment Amount",
     xlab = "Score")
```




```r
#ecdf graph generating
ex.ecdf = ecdf(ex$Score)
plot(ex.ecdf,
     xlab = "Score",
     main = "Empirical Cumluative Distribution For Extrinsic Treatment",
     ylab = "Extrinsic Treatment amount")
```




```
ggfortify::ggdistribution(ex.ecdf,
                          ex$Score,
                          colour = "red",
                          alpha = 0.7,
                          fill = "skyblue") +
  ggplot2::labs(title =
                "Cumulative Distribution Function for Extrinsic Treatment", ) +
  ggplot2::xlab ("Score") +
  ggplot2::ylab ("Frequesncy")
```



Task - 3 : (3pt) For each of the observed parts separately, find the most similar distribution: Estimate the parameters of the normal, exponential and uniform distribution. Insert the corresponding densities with estimated parameters into the plot of the histogram. Discuss which of them fits the data best.

Answer.

We know that, Formula for normal distribution,

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}.\exp^{-}\frac{1}{2}(\frac{x-\mu}{\sigma})^2)$$

Exponential distribution,

$$f(x) = \begin{cases} \frac{1}{(b-a)} & : a \leq x \leq b \\ 0 & : x < a \| x > b \end{cases}$$

Uniform distribution,

$$f(x) = \begin{cases} \lambda.\exp^{-}(\lambda.x) & : x \geq 0 \\ 0 & : x < 0 \end{cases}$$

Standard deviation,

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

Expectation,

$$E[X] = \sum x_i P_i$$

Standard deviation for uniform distribution,

$$\sigma = \sqrt{(}\frac{\sum(x_i - \mu)^2}{N}) = \frac{(b-a)^2}{12}$$

Expectation for uniform distribution,

$$E[X] = \sum x_i P_i = \frac{(a+b)}{2}$$

Now, we also need to consider maximum likelihood of three distribution For uniform distribution,

$$\hat{b}_n = max(1, ..., n)$$

For normal distribution,

$$\hat{\mu}_n = \frac{\sum x_i}{n}$$

For exponential distribution,

$$\hat{\lambda}_n = \frac{1}{\frac{\sum x_i}{n}}$$

In this problem we have to generate histogram of two datasets Intrinsic(In) and Extrinsic(ex).

At first lets start with Intrinsic data frame/dataset. First task is to calculate different parameters necessary to calculate the distributions for In. We are going to need mean value and standard deviation, a, b(for uniform distribution) for both In and ex. We get a+b(a_b) from expectation value and b - a(a___b) from standard deviation for(1st quadrant so b-a will be positive)

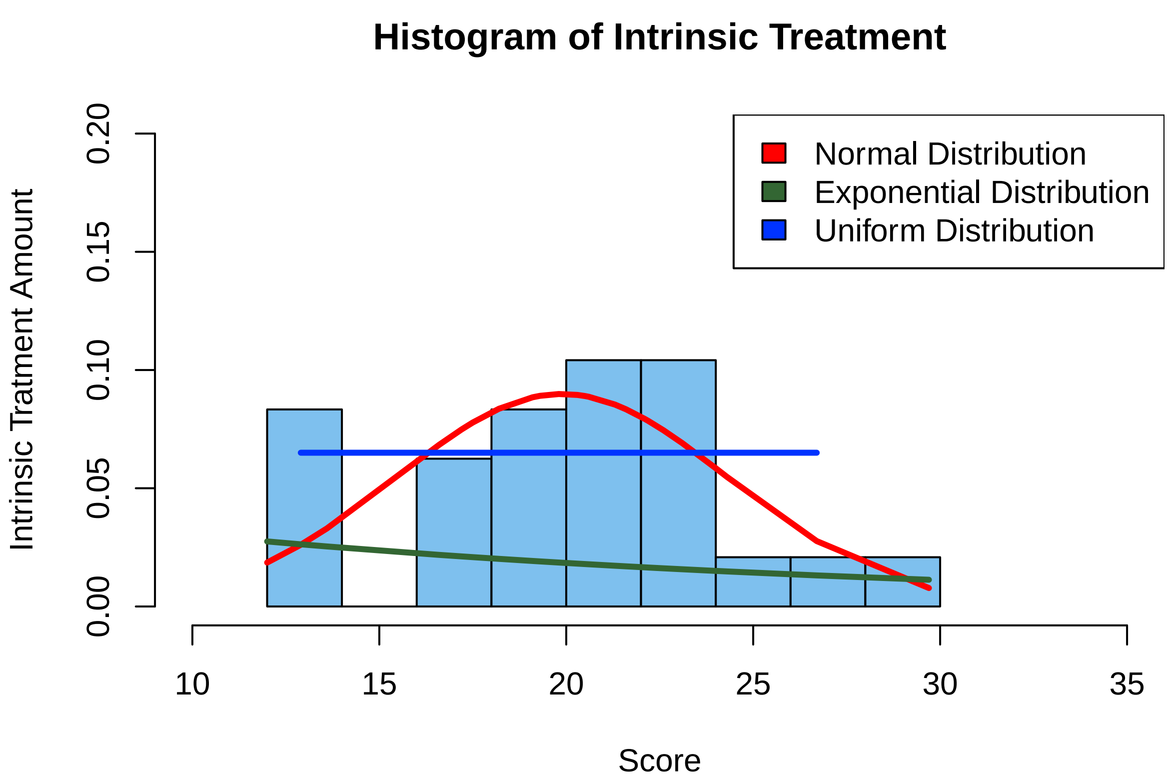
Then we use these functions, dnorm() for normal, dexp() for exponential, dunif() for uniform distribution

Then we can generate the histogram and plot the distributions.

```
mean_In <- mean(In$Score)
sd_In <- sqrt(var(In$Score))
a_b <- (mean(In$Score) * 2)#a+b
a__b <- (sqrt(var(In$Score) * 12))#b-a
a <- ((a_b - a__b) / 2)
b <- ((a_b + a__b) / 2)
In_x <- seq(min(In$Score),
            max(In$Score),
            length=1000)
In_y_normal <- dnorm(In$Score,
                     mean_In, sd_In)
In_y_expon <- dexp(In$Score,
                   rate = 1/mean_In)
In_y_uniform <- dunif(In$Score,
                      min = a,
                      max = b,
                      log = FALSE)

x = hist(In$Score ,
         col = "skyblue2",
         main = "Histogram of Intrinsic Treatment",
         ylab = "Intrinsic Tratment Amount",
```

```
         xlab = "Score",
         plot = TRUE,
         breaks = 12,
         probability = T,
         ylim = c(0, 0.2),
         xlim = c(10, 35))
legend("topright",
       seg.len = 2,
       c("Normal Distribution", "Exponential Distribution",
         "Uniform Distribution"),
       fill=c("red", "#336633", "#0033FF"))
lines(In$Score,
      In_y_normal,
      type = "l",
      col = "red",
      lwd = "3")
lines(In$Score,
      In_y_expon,
      type = "l",
      col =c("#336633", "#0000FF"),
      lwd = "3")
first <-first(which(In_y_uniform != 0))
last <- last(which(In_y_uniform != 0))
lines(In$Score[first:last],
      In_y_uniform[first:last],
      col =c("#0033FF"),
      lwd = "3")
```



```
#We used sum of the indexes for uniform distribution as they are 0 in that range
#Now we need to have maximum likelihood of estimator for Normal, Exponential and uniform distribution
ib_uniform = max(In$Score)
ib_exponen = 24/sum(In$Score)
ib_normal = sum(In$Score)/24
imin = min(abs(ib_uniform-sum(In$Score)/24), abs(ib_normal-sum(In$Score)/24), abs(ib_exponen-sum(In$S
#As ib_normal is close to the average of the data.So,
print(paste("The normal distribution fits the data best.", imin))
```

```
## [1] "The normal distribution fits the data best. 0"
```

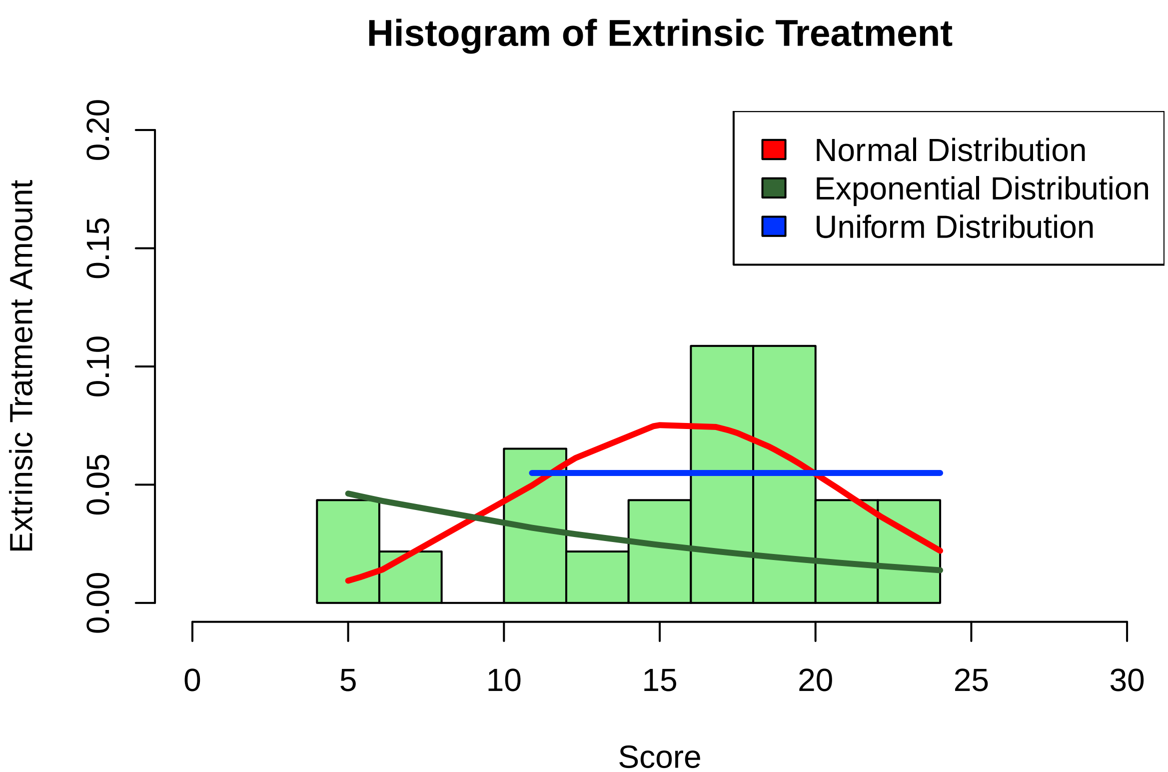
```
#We can see that only normal distribution has the minimum distance
```

Similarly, let's calculate differnet parameters necessary to calculate the distributions for ex.

```
mean_ex <- mean(ex$Score)
sd_ex <- sqrt(var(ex$Score))
a_b <- (mean(ex$Score) * 2)
a__b <- (sqrt(var(ex$Score) * 12))
a <- ((a_b - a__b) / 2)
b <- ((a_b + a__b) / 2)
ex_x <- seq(min(ex$Score),
            max(ex$Score),
            length=1000)
ex_y_normal <- dnorm(ex$Score,
                     mean_ex,
                     sd_ex)
ex_y_expon <- dexp(ex$Score,
                   rate = 1/mean_ex)
ex_y_uniform <- dunif(ex$Score,
                      min = a,
                      max = b,
                      log = FALSE)

y <- hist(ex$Score,
          col = "lightgreen",
          main = "Histogram of Extrinsic Treatment",
          ylab = "Extrinsic Tratment Amount",
          xlab = "Score",
          plot = TRUE,
          breaks = 8,
          probability = T,
          ylim = c(0, .2),
          xlim = c(0, 30))
legend("topright",
       seg.len = 1,
       c("Normal Distribution",
         "Exponential Distribution",
         "Uniform Distribution"),
       fill=c("red", "#336633", "#0033FF"))
```

```
lines(ex$Score, ex_y_normal,
      type = "l",
      col = "red",
      lwd = "3")
lines(ex$Score,
      ex_y_expon,
      type = "l",
      col =c("#336633", "#0000FF"),
      lwd = "3")
first <-first(which(ex_y_uniform != 0))
last <- last(which(ex_y_uniform != 0))
lines(ex$Score[first:last],
      ex_y_uniform[first:last],
      col =c("#0033FF"),
      lwd = "3")
```




```
eb_uniform = max(ex$Score)
eb_exponen = 23/sum(ex$Score)
eb_normal = sum(ex$Score)/23
emin = min(abs(eb_uniform-eb_normal), abs(eb_normal-eb_normal), abs(eb_exponen-eb_normal))
#As ib_normal is close to the average of the data.So,
print(paste("The normal distribution fits the data best with difference", emin))
```

```
## [1] "The normal distribution fits the data best with difference 0"
```

Task - 4 : (1pt) For each of the groups, generate a random sample of 100 observations from the distribution you have chosen in the previous part, with parameters estimated from the data. Compare the histogram of the simulated values with the original data.

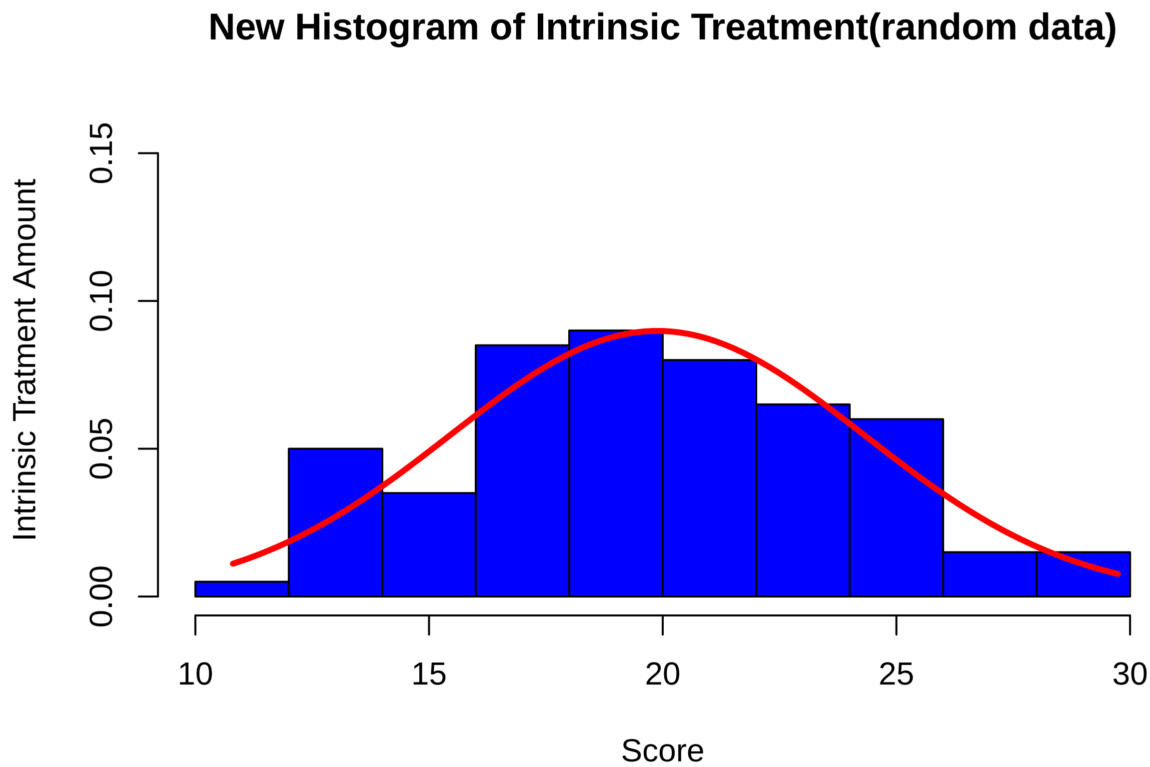
Answer.

For solving this problem, we first need to get a plot space for 4 graphs. We use par() to get the space.

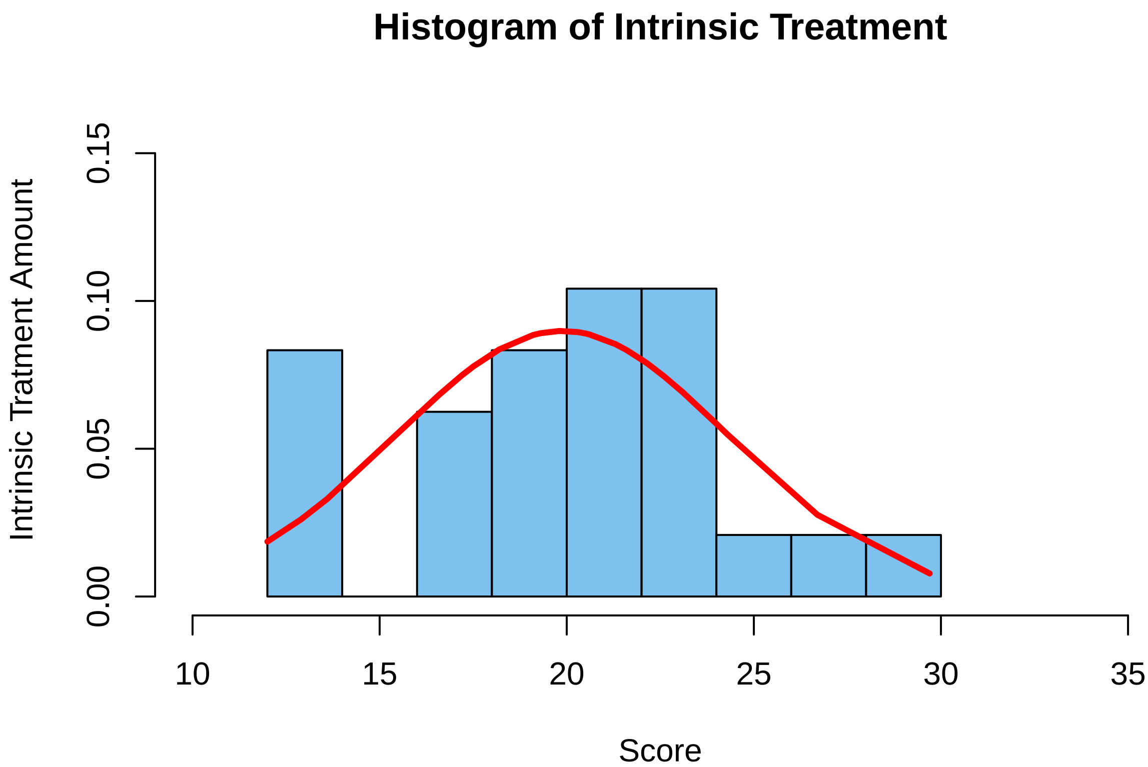
```
graph <- par(mfrow = c(2,2),
             cex = .4,
             mai = c(.3, .3, .3, .3))
```

Next, we generate new In100 vector with 100 random sample data which contains the same expected value and standard deviation as normal distribution.We will use dnorm() function for that. Next we plot both of these(new random, data from case0101) in two different histograms.

```
In100 <- rnorm(100,
               mean_In,
               sd_In)
In_100 <- seq(min(In100),
              max(In100),
              length=100)
In_y_normal100 <- dnorm(In_100,
                        mean_In,
                        sd_In)
graph[1:1] <- hist(In100 ,
                   col = "blue",
                   main = "New Histogram of Intrinsic Treatment(random data)",
                   ylab = "Intrinsic Tratment Amount",
                   xlab = "Score",
                   plot = TRUE,
                   breaks = 12,
                   probability = T,
                   ylim = c(0, 0.16))
lines(In_100,
      In_y_normal100,
      type = "l",
      col = "red",
      lwd = "3")
```

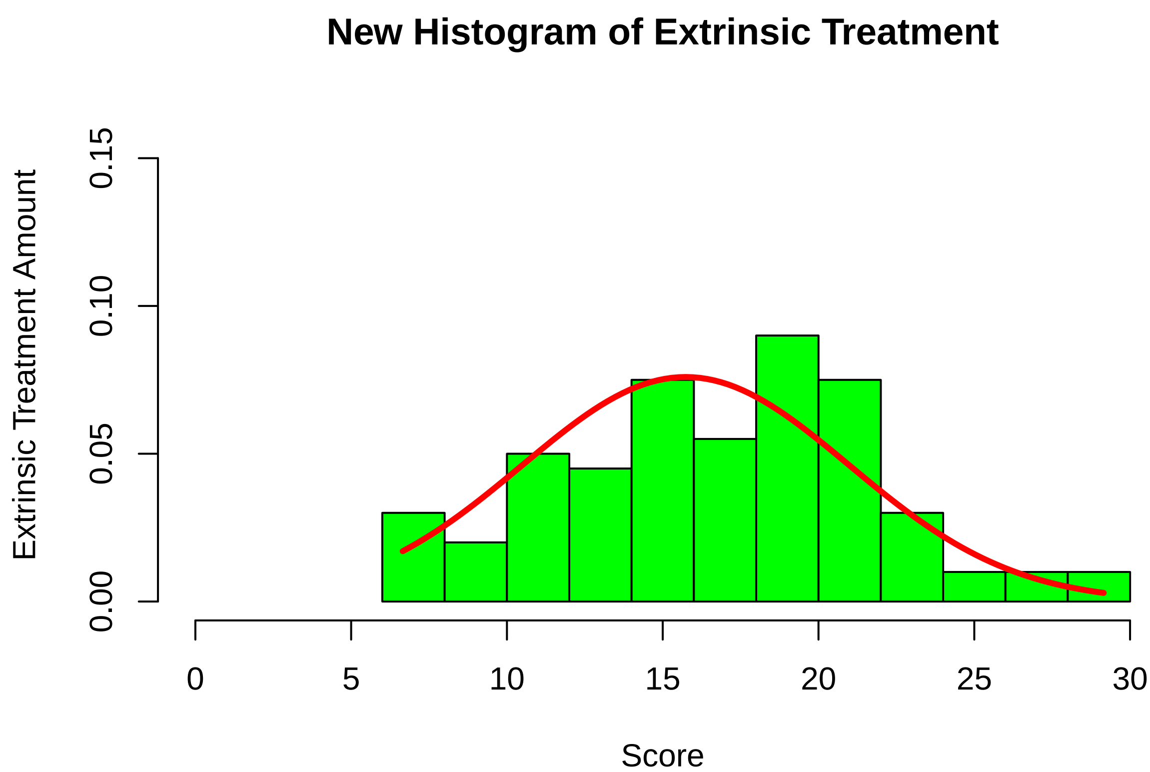



```
graph[1:2] <- hist(In$Score ,
                   col = "skyblue2",
                   main = "Histogram of Intrinsic Treatment",
                   ylab = "Intrinsic Tratment Amount",
                   xlab = "Score",
                   plot = TRUE,
                   breaks = 12,
                   probability = T,
                   ylim = c(0, 0.16),
                   xlim = c(10, 35))
lines(In$Score,
      In_y_normal,
      type = "l",
      col = "red",
      lwd = "3")
```

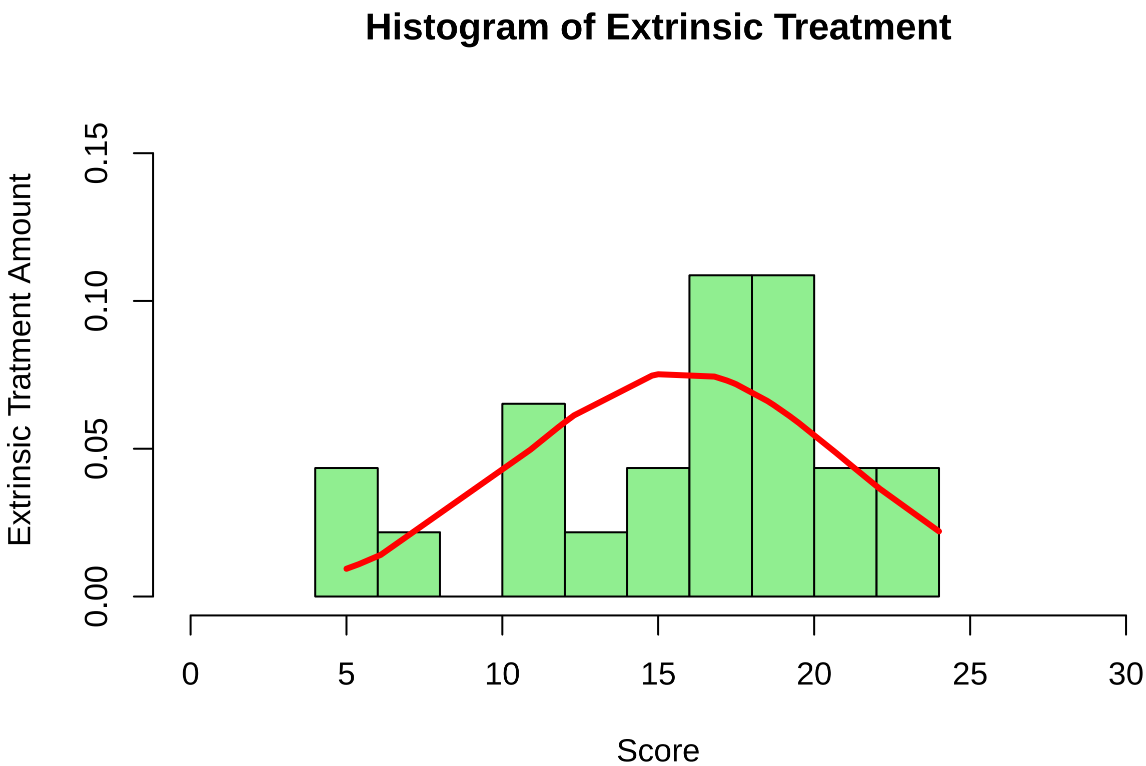


Same for ex100.

```
ex100 <- rnorm(100,
               mean_ex,
               sd_ex)
ex_100 <- seq(min(ex100),
              max(ex100),
              length=100)
ex_y_normal100 <- dnorm(ex_100,
                        mean_ex,
                        sd_ex)
graph[2:1] <- hist(ex100,
                   col = "green",
                   main = "New Histogram of Extrinsic Treatment",
                   ylab = "Extrinsic Treatment Amount",
                   xlab = "Score",
                   plot = TRUE,
                   breaks = 12,
                   probability = T,
                   ylim = c(0, .16),
                   xlim = c(0, 30))
lines(ex_100,
      ex_y_normal100,
      type = "l",
      col = "red",
      lwd = "3")
```




```
graph[2:2] <- hist(ex$Score,
                   col = "lightgreen",
                   main = "Histogram of Extrinsic Treatment",
                   ylab = "Extrinsic Tratment Amount",
                   xlab = "Score",
                   plot = TRUE,
                   breaks = 12,
                   probability = T,
                   ylim = c(0, .16),
                   xlim = c(0, 30))
lines(ex$Score,
      ex_y_normal,
      type = "l",
      col = "red",
      lwd = "3")
```



Our par() function generates the following graphs.

Thus we get 4 graphs for all 4 possible combination. Now, by comparing Intrinsic and Extrinsic random generated and actual data, **we see that in random generated, the normal distribution has the tails from and till the beginning and end of data.**

It also proves that our calculation comes from the the normal distribution as both of them has same kurtosis and skewdness.

Task - 5 : (1pt) For both parts separately, compute the two-sided

confidence interval for the expected value with confidence level 95%.

Answer.

We know that the formula for calculating confidence level is,

$$\left\langle X_n - \frac{t_{n-1}.\frac{\alpha}{2}.sd}{\sqrt{n}}, X_n + \frac{t_{n-1}.\frac{\alpha}{2}.sd}{\sqrt{n}} \right\rangle$$

$$\alpha = \frac{100 - confidence}{100}$$
$$n = \text{length of data-set}$$

.

So, first we start with Intrinsic treatments. We have mean and standard deviations from previous tasks.

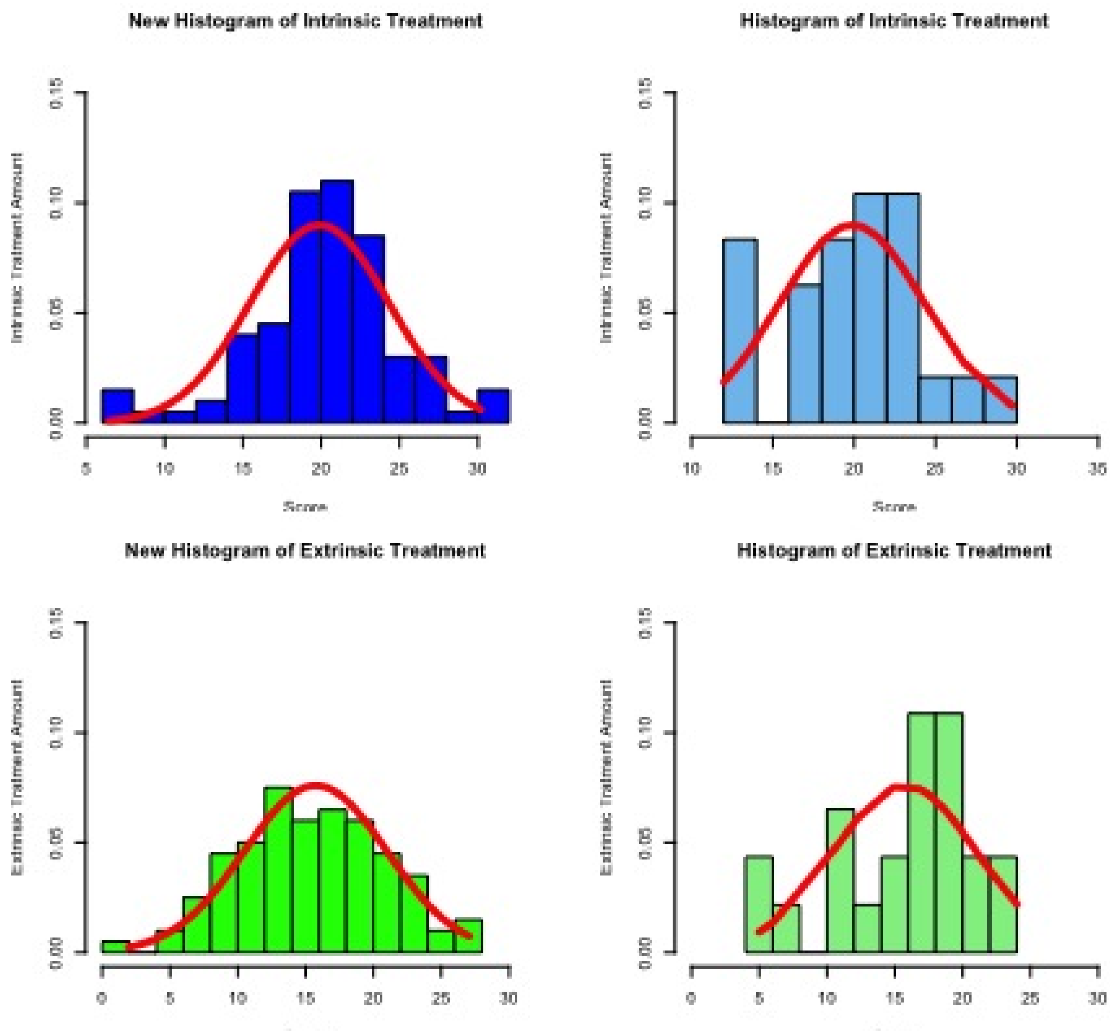
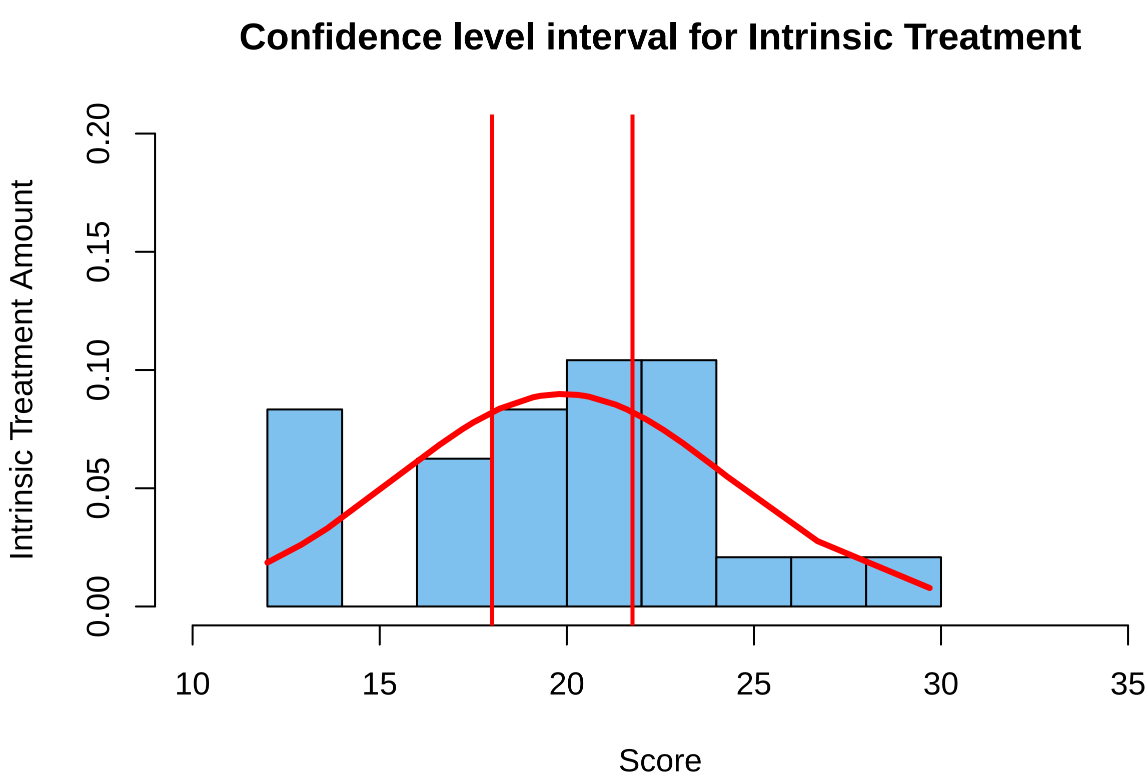


Figure 1: 4Graphs

```
In_lefttail = mean_In+
  (qt(.05/2, 23) *
     sd_In / sqrt(24))
In_righttail = mean_In-
  (qt(.05/2, 23) *
     sd_In / sqrt(24))
hist(In$Score ,
     col = "skyblue2",
     main = "Confidence level interval for Intrinsic Treatment",
     ylab = "Intrinsic Treatment Amount",
     xlab = "Score",
     plot = TRUE,
     breaks = 7,
     probability = T,
     ylim = c(0, 0.2),
     xlim = c(10, 35))
lines(In$Score,
      In_y_normal,
      type = "l",
      col = "red",
      lwd = "3")
abline(v = In_lefttail,
       col = "red",
       lwd = "2")
abline(v = In_righttail,
       col = "red",
       lwd = "2")
```



We can also view the values as,

```
In_lefttail
```

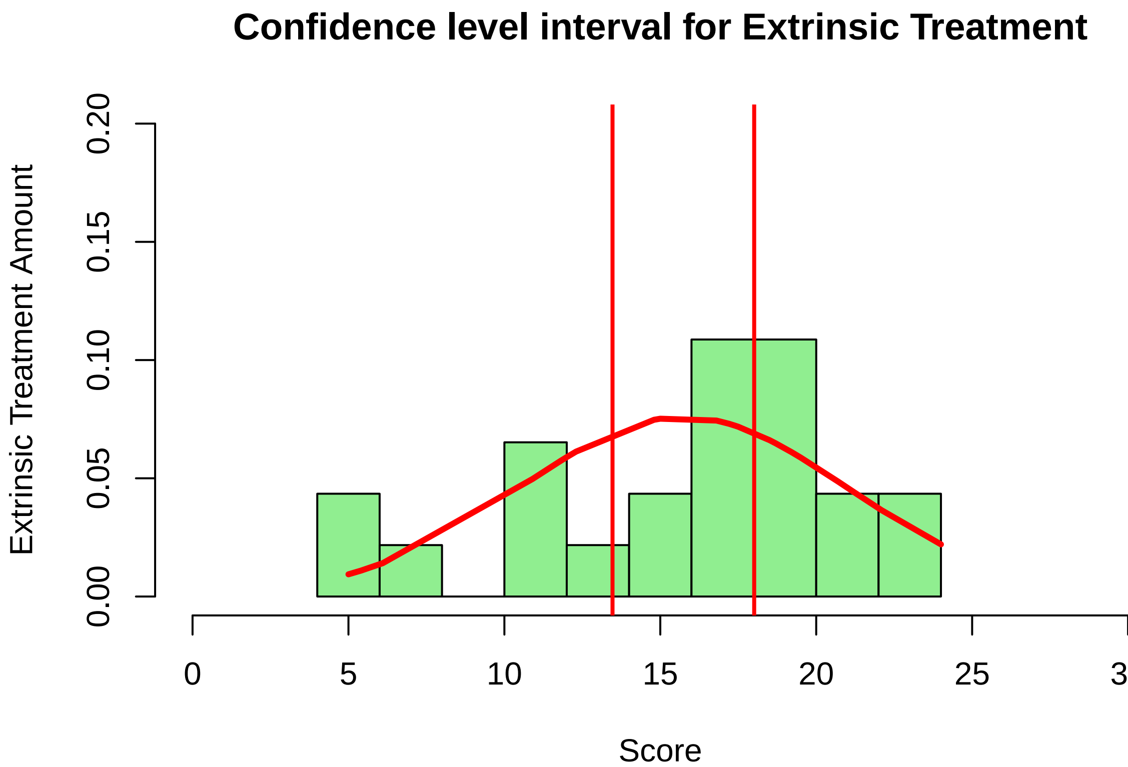
```
## [1] 18.00869
```

```
In_righttail
```

```
## [1] 21.75798
```

Next, we have to calculate similarly for extrinsic treatment.

```
ex_lefttail = mean_ex+
  (qt(.05/2, 22) *
     sd_ex / sqrt(23))
ex_righttail = mean_ex-
  (qt(.05/2, 22) *
     sd_ex / sqrt(23))
hist(ex$Score,
     col = "lightgreen",
     main = "Confidence level interval for Extrinsic Treatment",
     ylab = "Extrinsic Treatment Amount",
     xlab = "Score",
     plot = TRUE,
     breaks = 7,
     probability = T,
     ylim = c(0, .2),
     xlim = c(0, 30))
lines(ex$Score,
      ex_y_normal,
      type = "l",
      col = "red",
      lwd = "3")
abline(v = ex_lefttail,
       col = "red",
       lwd = "2")
abline(v = ex_righttail,
       col = "red",
       lwd = "2")
```



The values are

```
ex_lefttail
```

```
## [1] 13.46774
```

```
ex_righttail
```

```
## [1] 18.01052
```

Task - 6 : (1pt) Perform a test of the hypothesis, whether the expectation of either of the parts of the data set is equal to K (assignment parameter) against the two-sided alternative, on level of significance 5%. You can use either the previous result or an in-built function.

Answer.

Here in this problem, let's established the null hypothesis and its alternative for both groups with K = 15 as,

$$H_0 : \mu_l = 15, H_A : \mu_l \neq 15$$

Our

$$\mu_l$$

is treated as z in picture.

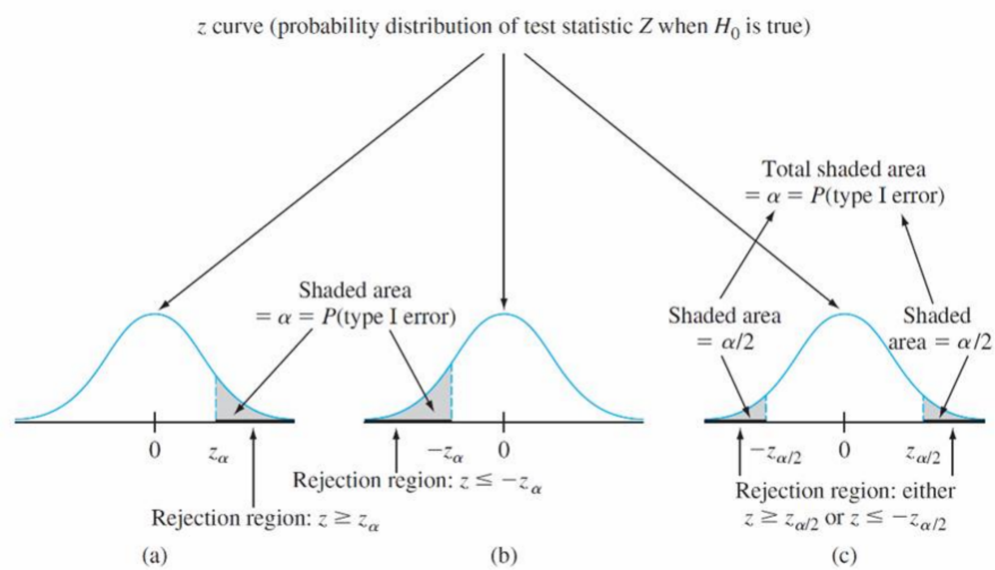


Figure 2: Hypothesis testing basis

We have to verify whether the null hypothesis is true based on the results from the previous task. That is, whether the value belongs to the interval calculated in In_lefttail and In_righttail variables, which was created in the previous task.

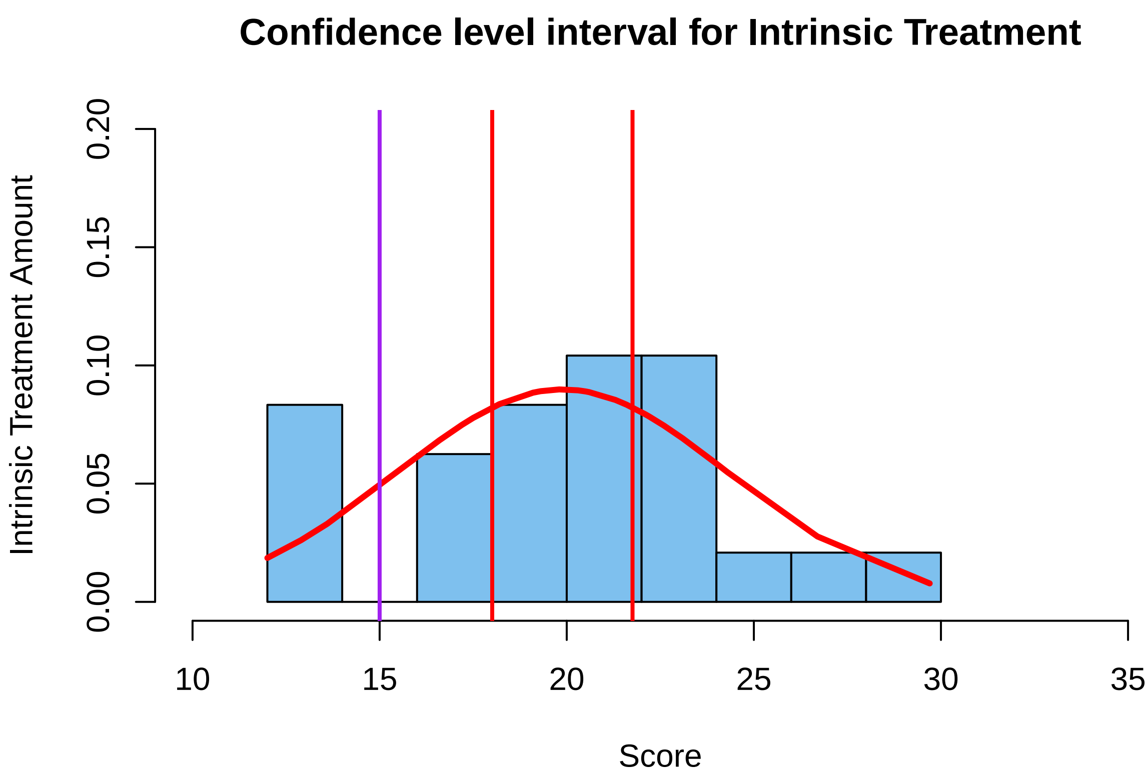
k = 15

```r
k = 15
if(k >= In_lefttail && k <= In_righttail){
  print('Is not rejected for Intrinsic')
}else{
  print('Is rejected for Intrinsic')
}
```

```
## [1] "Is rejected for Intrinsic"
```

```r
hist(In$Score ,
     col = "skyblue2",
     main = "Confidence level interval for Intrinsic Treatment",
     ylab = "Intrinsic Treatment Amount",
     xlab = "Score",
     plot = TRUE,
     breaks = 7,
     probability = T,
     ylim = c(0, 0.2),
     xlim = c(10, 35))
lines(In$Score,
      In_y_normal,
      type = "l",
```

```
      col = "red",
      lwd = "3")
abline(v = In_lefttail,
       col = "red",
       lwd = "2")
abline(v = In_righttail,
       col = "red",
       lwd = "2")
abline(v = 15,
       col = "purple",
       lwd = "2")
```



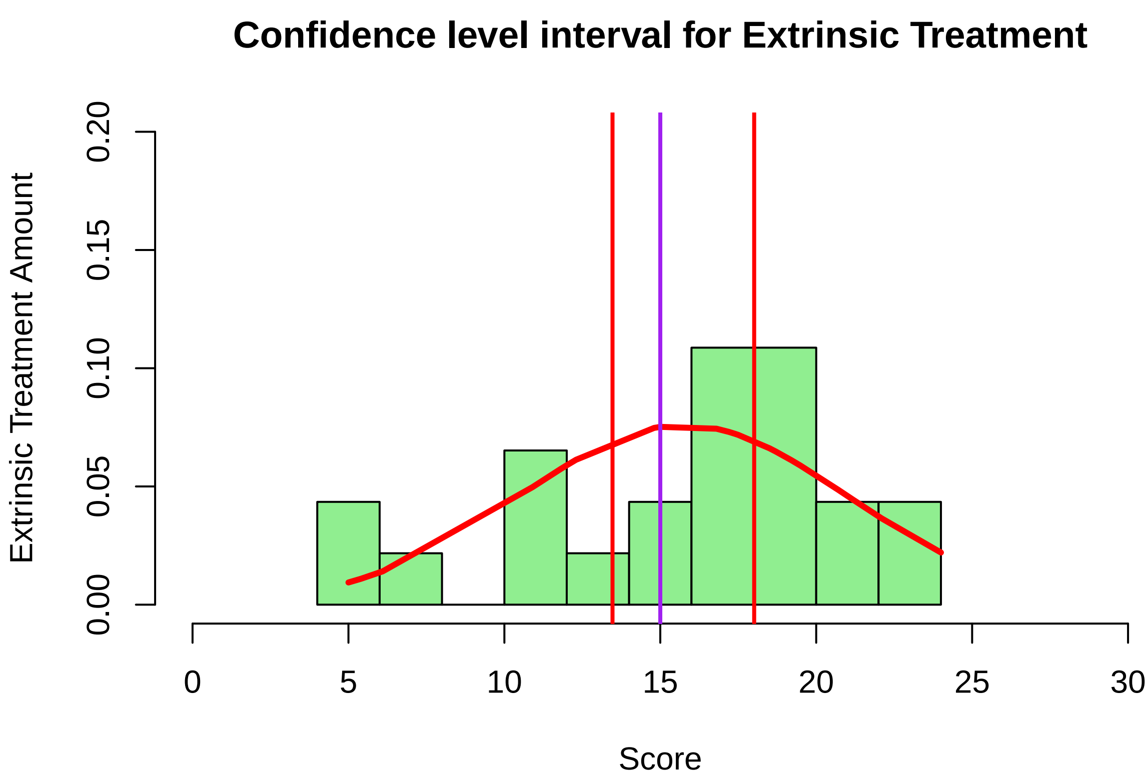
Similarly for Extrinsic amount we get that the testing is not rejected.

```
if(k >= ex_lefttail &&
   k <= ex_righttail){
  print('Is not rejected for Extrinsic')
}else{
  print('Is rejected for Extrinsic')
}
```

```
## [1] "Is not rejected for Extrinsic"
```

```
hist(ex$Score,
     col = "lightgreen",
     main = "Confidence level interval for Extrinsic Treatment",
```

```
     ylab = "Extrinsic Treatment Amount",
     xlab = "Score",
     plot = TRUE,
     breaks = 7,
     probability = T,
     ylim = c(0, .2),
     xlim = c(0, 30))
lines(ex$Score,
      ex_y_normal,
      type = "l",
      col = "red",
      lwd = "3")
abline(v = ex_lefttail,
       col = "red",
       lwd = "2")
abline(v = ex_righttail,
       col = "red",
       lwd = "2")
abline(v = 15,
       col = "purple",
       lwd = "2")
```



Task - 7 : (2pt) Perform a test of the hypothesis, whether the expectations of both observed parts are equal. Use level of significance 5%. Choose the type of test and the alternative hypothesis in a way which corresponds with the examined problem best.

Answer.

We test whether the expected heights are equal, against the alternative that they are not, on

$$\alpha = 5\%$$

.

Here in this problem, let's established the null hypothesis and its alternative for both groups as,

$$H_0 : \mu_l = \mu_x, H_A : \mu_l \neq \mu_x$$

First we deal with equal or unequal variances. Now, we perform var.test() to test equality of variances.

```
var.test(In$Score, ex$Score)
```

```
##
##  F test to compare two variances
##
## data:  In$Score and ex$Score
## F = 0.71437, num df = 23, denom df = 22, p-value = 0.4289
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.3047427 1.6612045
## sample estimates:
## ratio of variances
##          0.7143691
```

As the p-value in var.test() is greater than significant level we can use t.test() for the hypothesis testing.

Tests for the equality of expectations under $\sigma_1^2 = \sigma_2^2$:

H_0	H_A	test statistic T	critical region W_α
$\mu_1 = \mu_2$	$\mu_1 \neq \mu_2$	$T = \frac{\bar{X}_{n_1} - \bar{Y}_{n_2}}{s_{12}} \sqrt{\frac{n_1 n_2}{n_1 + n_2}}$	$\lvert T\rvert > t_{\alpha/2, n_1+n_2-2}$
$\mu_1 \leq \mu_2$	$\mu_1 > \mu_2$		$T > t_{\alpha, n_1+n_2-2}$
$\mu_1 \geq \mu_2$	$\mu_1 < \mu_2$		$T < -t_{\alpha, n_1+n_2-2}$

Figure 3: Hypothesis testing basis

Now we can use t.test() function to compare the expectations for both dataset expectations. Here in the result, df in result refers to

$$S_{12}$$

and t refers to the T value in picture.

```r
res <- t.test(In$Score, ex$Score, paired =  F, conf.level = .95, alternative = "two.sided", var.equal
t.test(In$Score, ex$Score, paired =  F, conf.level = .95, alternative = "two.sided", var.equal = T)
```

```
##
##  Two Sample t-test
##
## data:  In$Score and ex$Score
## t = 2.9259, df = 45, p-value = 0.005366
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.291432 6.996973
## sample estimates:
## mean of x mean of y
##  19.88333  15.73913
```

Here in the t.test result we can see that p value is less than .05, significant level. Also t value is greater than calculated |T| value(from picture). Therefore we can conclude that expectation of both observed parts are not equal.

```r
if(res$p.value > .05){
  print('We do not reject the null hypothesis of equality')
}else{
  print('We reject the null hypothesis of equality')
}
```

```
## [1] "We reject the null hypothesis of equality"
```

Now if we check for variances not equal, we get same result for different confidence level.

```r
t.test(In$Score, ex$Score, paired =  F, conf.level = .95, alternative = "two.sided", var.equal = F)
```

```
##
##  Welch Two Sample t-test
##
## data:  In$Score and ex$Score
## t = 2.9153, df = 43.108, p-value = 0.005618
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.277603 7.010803
## sample estimates:
## mean of x mean of y
##  19.88333  15.73913
```