APPLIED STATISTICAL ANALYSIS LAB

NAME: ADITI KULKARNI

ROLL NO: 55

YEAR: SY

DIVISION: E (E3)

SRN NO: 202201893

ASSIGNMENT 2

STATEMENT: *Run frequencies to explore distribution of several variables using pre-existing data file.*

THEORY:

- ***Histogram Definition****: A histogram is a graphical representation of the distribution of a dataset. It displays the frequency or count of data points falling into different intervals, called "bins."*
- ***Data Binning****: The first step in creating a histogram is to divide the data range into equal-width bins. This is usually done by specifying the number of bins or the bin width.*
- ***Frequency Calculation****: Once the bins are defined, the histogram calculates the frequency or count of data points that fall into each bin.*
- ***Plotting Bars****: Histograms consist of vertical bars, where the height of each bar represents the frequency of data points in the corresponding bin.*
- ***Density Scaling****: To compare distributions with different sample sizes, histograms can be scaled to represent a density distribution (normalized to have area equal to 1).*
- ***Main Components****: A basic histogram plot includes the x-axis (data values), y-axis (frequency or density), and bars representing each bin.*
- ***Bin Width Calculation:***
  Bin Width = (Max Value - Min Value) / Number of Bins
- ***Plotting Histogram:***

*Use the **hist()** function in R to create a histogram.*
*Syntax: **hist(data, breaks = k, col = "color", main = "Title", xlab = "X Label", ylab = "Y Label")***

- ***Custom Binning:***
  *You can specify your own breaks using the **breaks** parameter in the **hist()** function.*
- ***Density Histogram:***
  A density histogram normalizes the counts by the bin width, giving you a relative frequency.
- ***Adding Lines:***
  *Use **abline(v = value, col = "color")** to add vertical lines at specific values on the histogram.*
- ***Histogram Shapes:***
  Histograms can have different shapes: symmetric (bell-shaped), skewed (long tail on one side), or uniform.

SOURCE CODE:

FOR HEIGHT -

```
data<-read.csv(file.choose()) ## Read the dataset
View(data) ## View the data-frame
dim(data) ## find rows and columns
str(data) ## find attribute types
var=data$HT ## select a column data
mx=max(var) ## find max value in that column
mx ## display max
mn=min(var) ## find min value in that column
mn ## display max
width=(mx-mn)/5 ## divide the range into 5 bins
bins=seq(mn,mx,width) ## create list of class boundaries
scores=cut(var,bins) ## group the data into bin
table(scores) ## display frequency table
transform(table(scores))
```

```
freq_table=transform(table(scores))
freq_table
transform(freq_table,Rel_Freq=prop.table(Freq),Cum_Freq=cumsum(Freq))
#### Histogram
hist(var)
hist(var,breaks = bins, main= "Histogram For Weight", xlab= "Height", col="light yellow”)
```

FOR WEIGHT –

```
data<-read.csv(file.choose()) ## Read the dataset
View(data) ## View the data-frame
dim(data) ## find rows and columns
str(data) ## find attribute types
var=data$WT ## select a column data
mx=max(var) ## find max value in that column
mx ## display max
mn=min(var) ## find min value in that column
mn ## display max
width=(mx-mn)/5 ## divide the range into 5 bins
bins=seq(mn,mx,width) ## create list of class boundaries
scores=cut(var,bins) ## group the data into bin
table(scores) ## display frequency table
transform(table(scores))
freq_table=transform(table(scores))
freq_table
transform(freq_table,Rel_Freq=prop.table(Freq),Cum_Freq=cumsum(Freq))
#### Histogram
hist(var)
```

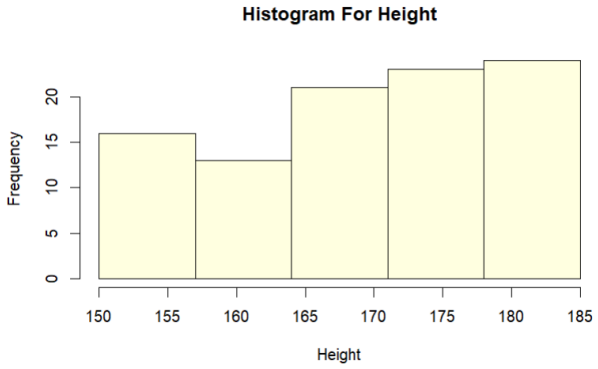
```
hist(var,breaks = bins, main= "Histogram For Weight", xlab= "Weight", col="light pink")
```

FOR AGE –

```
data<-read.csv(file.choose()) ## Read the dataset
View(data) ## View the data-frame
dim(data) ## find rows and columns
str(data) ## find attribute types
var=data$AGE ## select a column data
mx=max(var) ## find max value in that column
mx ## display max
mn=min(var) ## find min value in that column
mn ## display max
width=(mx-mn)/5 ## divide the range into 5 bins
bins=seq(mn,mx,width) ## create list of class boundaries
scores=cut(var,bins) ## group the data into bin
table(scores) ## display frequency table
transform(table(scores))
freq_table=transform(table(scores))
freq_table
transform(freq_table,Rel_Freq=prop.table(Freq),Cum_Freq=cumsum(Freq))
#### Histogram
hist(var)
hist(var,breaks = bins, main= "Histogram For Age ", xlab= "AGE", col="light blue")
```

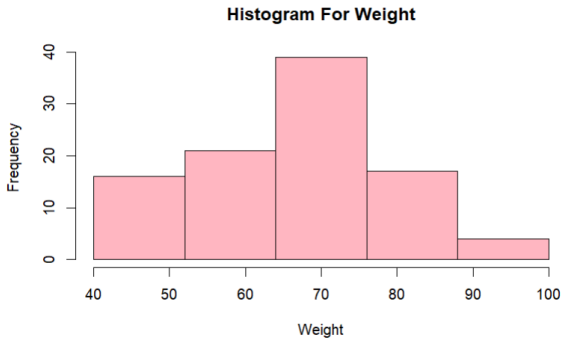
OUTPUT:

FOR HEIGHT-



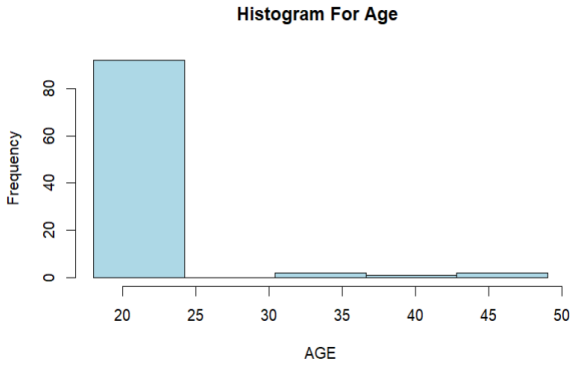

```
> var=dataSHT ## select a column data
> mx=max(var) ## find max value in that column
> mx ## display max
[1] 185
> mn=min(var) ## find min value in that column
> mn ## display max
[1] 150
> width=(mx-mn)/5 ## divide the range into 5 bins
> bins=seq(mn,mx,width) ## create list of class boundaries
> scores=cut(var,bins) ## group the data into bin
> table(scores) ## display frequency table
scores
(150,157] (157,164] (164,171] (171,178] (178,185]
       12        13        21        23        24
> transform(table(scores))
     scores Freq
1 (150,157]   12
2 (157,164]   13
3 (164,171]   21
4 (171,178]   23
5 (178,185]   24
> freq_table=transform(table(scores))
> freq_table
     scores Freq
1 (150,157]   12
2 (157,164]   13
3 (164,171]   21
4 (171,178]   23
5 (178,185]   24
> transform(freq_table,Rel_Freq=prop.table(Freq),Cum_Freq=cumsum(Freq))
     scores Freq  Rel_Freq Cum_Freq
1 (150,157]   12 0.1290323       12
2 (157,164]   13 0.1397849       25
3 (164,171]   21 0.2258065       46
4 (171,178]   23 0.2473118       69
5 (178,185]   24 0.2580645       93
```

FOR WEIGHT-




```
> mx=max(var) ## find max value in that column
> mx ## display max
[1] 100
> mn=min(var) ## find min value in that column
> mn ## display max
[1] 40
> width=(mx-mn)/5 ## divide the range into 5 bins
> bins=seq(mn,mx,width) ## create list of class boundaries
> scores=cut(var,bins) ## group the data into bin
> table(scores) ## display frequency table
scores
 (40,52]  (52,64]  (64,76]  (76,88] (88,100]
      14       21       39       17        4
> transform(table(scores))
    scores Freq
1  (40,52]   14
2  (52,64]   21
3  (64,76]   39
4  (76,88]   17
5 (88,100]    4
> freq_table=transform(table(scores))
> freq_table
    scores Freq
1  (40,52]   14
2  (52,64]   21
3  (64,76]   39
4  (76,88]   17
5 (88,100]    4
> transform(freq_table,Rel_Freq=prop.table(Freq),Cum_Freq=cumsum(Freq))
    scores Freq   Rel_Freq Cum_Freq
1  (40,52]   14 0.14736842       14
2  (52,64]   21 0.22105263       35
3  (64,76]   39 0.41052632       74
4  (76,88]   17 0.17894737       91
5 (88,100]    4 0.04210526       95
```

FOR AGE-




```
> mx=max(var) ## find max value in that column
> mx ## display max
[1] 49
> mn=min(var) ## find min value in that column
> mn ## display max
[1] 18
> width=(mx-mn)/5 ## divide the range into 5 bins
> bins=seq(mn,mx,width) ## create list of class boundaries
> scores=cut(var,bins) ## group the data into bin
> table(scores) ## display frequency table
scores
  (18,24.2] (24.2,30.4] (30.4,36.6] (36.6,42.8]   (42.8,49]
         90           0           2           1           2
> transform(table(scores))
     scores Freq
1   (18,24.2]   90
2 (24.2,30.4]    0
3 (30.4,36.6]    2
4 (36.6,42.8]    1
5   (42.8,49]    2
> freq_table=transform(table(scores))
> freq_table
     scores Freq
1   (18,24.2]   90
2 (24.2,30.4]    0
3 (30.4,36.6]    2
4 (36.6,42.8]    1
5   (42.8,49]    2
> transform(freq_table,Rel_Freq=prop.table(Freq),Cum_Freq=cumsum(Freq))
     scores Freq   Rel_Freq Cum_Freq
1   (18,24.2]   90 0.9473684       90
2 (24.2,30.4]    0 0.0000000       90
3 (30.4,36.6]    2 0.0210526       92
4 (36.6,42.8]    1 0.0105263       93
5   (42.8,49]    2 0.0210526       95
```

CONCLUSION:

The code reads a dataset, explores its characteristics, and conducts an analysis on three attributes: Height, Weight, and Age. It follows these steps for each attribute:

- *Calculate the range and bin width for the attribute values.*
- *Group the data into bins based on the calculated boundaries.*
- *Generate a frequency table for the grouped data, along with relative and cumulative frequencies.*
- *Create a histogram to visually represent the distribution of the attribute values.*
-

The code provides a comprehensive approach to understanding the distribution of Height, Weight, and Age attributes within the dataset. By calculating frequency tables and creating histograms, it offers a visual and numerical insight into the spread and concentration of data points for each attribute. This analysis aids in identifying patterns, outliers, and general trends within the dataset. Users can leverage the code as a template for exploring and visualizing other attributes or datasets, enhancing their data exploration and decision-making processes.