AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Science and Technology

Department of Computer Science

Course Name: Introduction to Data Science

Midterm Project

Prepared by-

MRINMOY DAS	HELEN CHORA CHOWDHURY
SECTION: B	SECTION: B
ID: 20-43856-2	ID: 20-43996-2

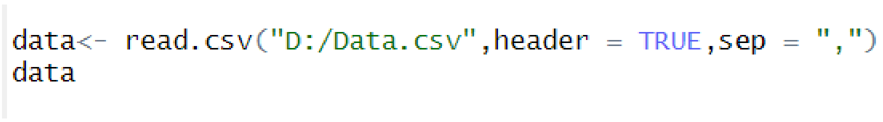
Submitted to:

TOHEDUL ISLAM
Assistant Professor
Faculty of Science & Technology
American International University-Bangladesh

About the dataset:

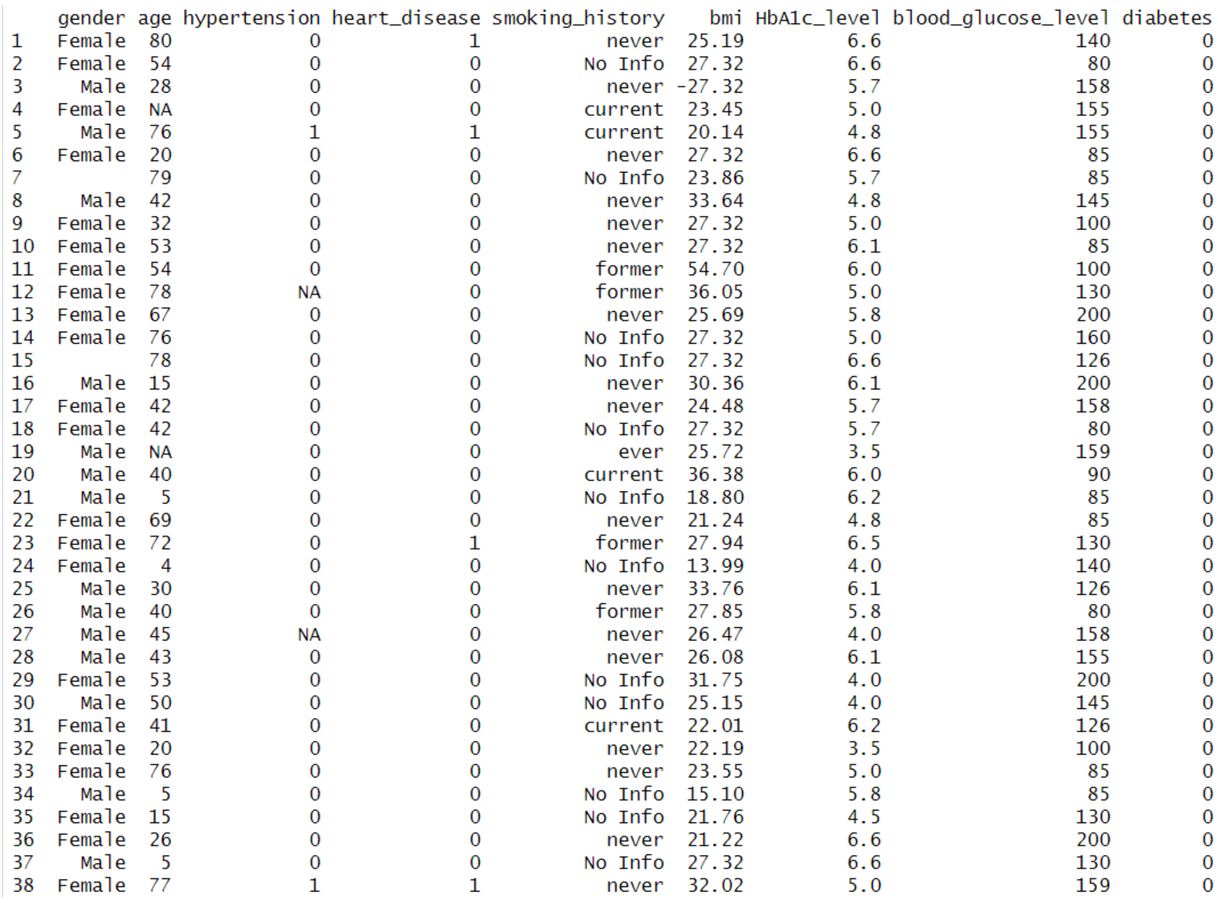
A diabetes prediction dataset contains information about individuals' health, lifestyle, and demographics, with a binary target variable indicating diabetes presence. Common features include age, gender, BMI, blood pressure, glucose and cholesterol levels, family history, physical activity, dietary habits, smoking, alcohol, and medical history. These features help identify potential risk factors and relationships that contribute to diabetes onset. Researchers use these datasets to build predictive models, perform data analysis, and identify factors influencing diabetes, aiding in prevention and management strategies.

Import the data set as csv and print the data set:



```r
data<- read.csv("D:/Data.csv",header = TRUE,sep = ",")
data
```

Output:



```
   gender age hypertension heart_disease smoking_history    bmi HbA1c_level blood_glucose_level diabetes
1  Female  80            0             1           never  25.19         6.6                 140        0
2  Female  54            0             0         No Info  27.32         6.6                  80        0
3    Male  28            0             0           never -27.32         5.7                 158        0
4  Female  NA            0             0         current  23.45         5.0                 155        0
5    Male  76            1             1         current  20.14         4.8                 155        0
6  Female  20            0             0           never  27.32         6.6                  85        0
7          79            0             0         No Info  23.86         5.7                  85        0
8    Male  42            0             0           never  33.64         4.8                 145        0
9  Female  32            0             0           never  27.32         5.0                 100        0
10 Female  53            0             0           never  27.32         6.1                  85        0
11 Female  54            0             0          former  54.70         6.0                 100        0
12 Female  78           NA             0          former  36.05         5.0                 130        0
13 Female  67            0             0           never  25.69         5.8                 200        0
14 Female  76            0             0         No Info  27.32         5.0                 160        0
15         78            0             0         No Info  27.32         6.6                 126        0
16   Male  15            0             0           never  30.36         6.1                 200        0
17 Female  42            0             0           never  24.48         5.7                 158        0
18 Female  42            0             0         No Info  27.32         5.7                  80        0
19   Male  NA            0             0            ever  25.72         3.5                 159        0
20   Male  40            0             0         current  36.38         6.0                  90        0
21   Male   5            0             0         No Info  18.80         6.2                  85        0
22 Female  69            0             0           never  21.24         4.8                  85        0
23 Female  72            0             1          former  27.94         6.5                 130        0
24 Female   4            0             0         No Info  13.99         4.0                 140        0
25   Male  30            0             0           never  33.76         6.1                 126        0
26   Male  40            0             0          former  27.85         5.8                  80        0
27   Male  45           NA             0           never  26.47         4.0                 158        0
28   Male  43            0             0           never  26.08         6.1                 155        0
29 Female  53            0             0         No Info  31.75         4.0                 200        0
30   Male  50            0             0         No Info  25.15         4.0                 145        0
31 Female  41            0             0         current  22.01         6.2                 126        0
32 Female  20            0             0           never  22.19         3.5                 100        0
33 Female  76            0             0           never  23.55         5.0                  85        0
34   Male   5            0             0         No Info  15.10         5.8                  85        0
35 Female  15            0             0         No Info  21.76         4.5                 130        0
36 Female  26            0             0           never  21.22         6.6                 200        0
37   Male   5            0             0         No Info  27.32         6.6                 130        0
38 Female  77            1             1           never  32.02         5.0                 159        0
```

Description: Here is the code of importing the dataset as csv file. It is the output of the dataset which is imported in RStudio.

Structure of Data:

Code:

```
str(data)
```

Output:

```
'data.frame':	120 obs. of  9 variables:
 $ gender             : chr  "Female" "Female" "Male" "Female" ...
 $ age                : int  80 54 28 NA 76 20 79 42 32 53 ...
 $ hypertension       : int  0 0 0 0 1 0 0 0 0 0 ...
 $ heart_disease      : int  1 0 0 0 1 0 0 0 0 0 ...
 $ smoking_history    : chr  "never" "No Info" "never" "current" ...
 $ bmi                : num  25.2 27.3 -27.3 23.4 20.1 ...
 $ HbA1c_level        : num  6.6 6.6 5.7 5 4.8 6.6 5.7 4.8 5 6.1 ...
 $ blood_glucose_level: int  140 80 158 155 155 85 85 145 100 85 ...
 $ diabetes           : int  0 0 0 0 0 0 0 0 0 0 ...
```

Description:

This dataset, with 120 observations and 9 variables, includes information on individuals' characteristics such as gender, age, hypertension, heart disease, smoking history, BMI, HbA1c level, blood glucose level, and diabetes status. The data types of variables vary between character, integer, and numeric, and some missing values (NA) are present in the "age" column.

Descriptive Statistics Using summary() Function:

Code:

```
summary(data)
```

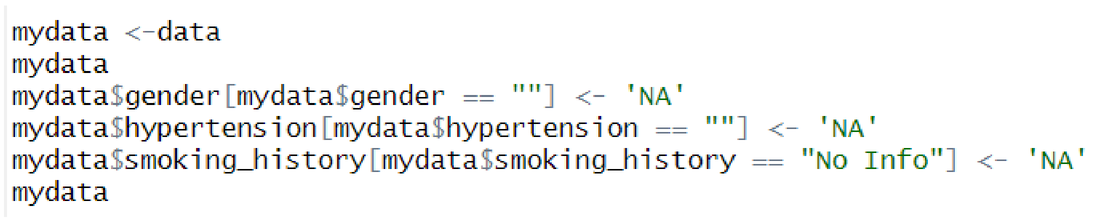
Output:

```
    gender               age          hypertension      heart_disease     smoking_history         bmi           HbA1c_level   
 Length:120         Min.   :  3.00   Min.   :0.00000   Min.   :0.00000   Length:120         Min.   :-27.32   Min.   :3.500  
 Class :character   1st Qu.: 40.00   1st Qu.:0.00000   1st Qu.:0.00000   Class :character   1st Qu.: 24.73   1st Qu.:5.700  
 Mode  :character   Median : 52.50   Median :0.00000   Median :0.00000   Mode  :character   Median : 27.32   Median :6.200  
                    Mean   : 54.17   Mean   :0.08475   Mean   :0.06667                      Mean   : 27.66   Mean   :6.275  
                    3rd Qu.: 68.25   3rd Qu.:0.00000   3rd Qu.:0.00000                      3rd Qu.: 29.53   3rd Qu.:6.650  
                    Max.   :290.00   Max.   :1.00000   Max.   :1.00000                      Max.   : 63.48   Max.   :9.000  
                    NA's   :4        NA's   :2                                                                                
 blood_glucose_level    diabetes     
 Min.   : 80.0       Min.   :0.0000  
 1st Qu.:130.0       1st Qu.:0.0000  
 Median :155.0       Median :0.0000  
 Mean   :156.8       Mean   :0.4333  
 3rd Qu.:160.0       3rd Qu.:1.0000  
 Max.   :300.0       Max.   :1.0000  
```

Description: Here using descriptive statistic, and the summary() function. In the output min, max, median, and mean are shown for every column.

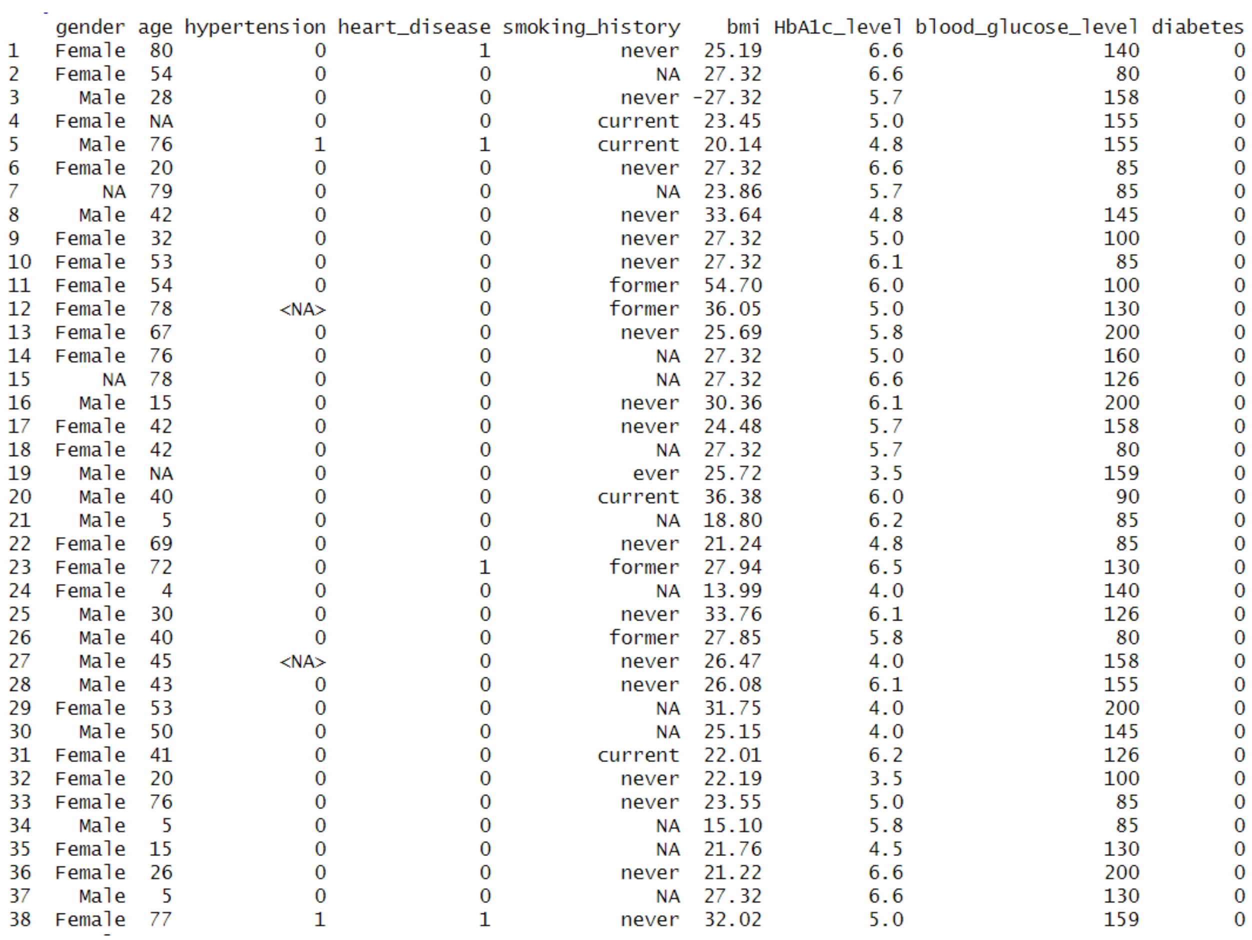
Outlier Detection as a missing value:

Code:



```
mydata <-data
mydata
mydata$gender[mydata$gender == ""] <- 'NA'
mydata$hypertension[mydata$hypertension == ""] <- 'NA'
mydata$smoking_history[mydata$smoking_history == "No Info"] <- 'NA'
mydata
```

Output:



```
   gender age hypertension heart_disease smoking_history    bmi HbA1c_level blood_glucose_level diabetes
1  Female  80            0             1           never  25.19         6.6                 140        0
2  Female  54            0             0              NA  27.32         6.6                  80        0
3    Male  28            0             0           never -27.32         5.7                 158        0
4  Female  NA            0             0         current  23.45         5.0                 155        0
5    Male  76            1             1         current  20.14         4.8                 155        0
6  Female  20            0             0           never  27.32         6.6                  85        0
7      NA  79            0             0              NA  23.86         5.7                  85        0
8    Male  42            0             0           never  33.64         4.8                 145        0
9  Female  32            0             0           never  27.32         5.0                 100        0
10 Female  53            0             0           never  27.32         6.1                  85        0
11 Female  54            0             0          former  54.70         6.0                 100        0
12 Female  78         <NA>             0          former  36.05         5.0                 130        0
13 Female  67            0             0           never  25.69         5.8                 200        0
14 Female  76            0             0              NA  27.32         5.0                 160        0
15     NA  78            0             0              NA  27.32         6.6                 126        0
16   Male  15            0             0           never  30.36         6.1                 200        0
17 Female  42            0             0           never  24.48         5.7                 158        0
18 Female  42            0             0              NA  27.32         5.7                  80        0
19   Male  NA            0             0            ever  25.72         3.5                 159        0
20   Male  40            0             0         current  36.38         6.0                  90        0
21   Male   5            0             0              NA  18.80         6.2                  85        0
22 Female  69            0             0           never  21.24         4.8                  85        0
23 Female  72            0             1          former  27.94         6.5                 130        0
24 Female   4            0             0              NA  13.99         4.0                 140        0
25   Male  30            0             0           never  33.76         6.1                 126        0
26   Male  40            0             0          former  27.85         5.8                  80        0
27   Male  45         <NA>             0           never  26.47         4.0                 158        0
28   Male  43            0             0           never  26.08         6.1                 155        0
29 Female  53            0             0              NA  31.75         4.0                 200        0
30   Male  50            0             0              NA  25.15         4.0                 145        0
31 Female  41            0             0         current  22.01         6.2                 126        0
32 Female  20            0             0           never  22.19         3.5                 100        0
33 Female  76            0             0           never  23.55         5.0                  85        0
34   Male   5            0             0              NA  15.10         5.8                  85        0
35 Female  15            0             0              NA  21.76         4.5                 130        0
36 Female  26            0             0           never  21.22         6.6                 200        0
37   Male   5            0             0              NA  27.32         6.6                 130        0
38 Female  77            1             1           never  32.02         5.0                 159        0
```

Description:

In this code, missing or empty values in the "gender," "hypertension," and "smoking_history" columns of the "mydata" dataset have been replaced with 'NA'. This is a common data preprocessing step to handle missing or unknown values.

Annotate Male as 1, Female as 0 from "gender" attribute:

Code:

```
mydata$gender <- factor(mydata$gender,
                        levels = c("Female","Male"),
                        labels = c(0,1))
mydata
```

Output:

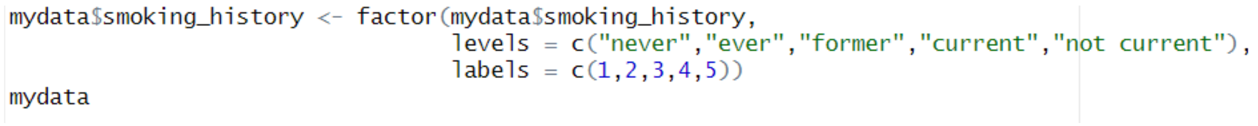
```
   gender age hypertension heart_disease smoking_history   bmi HbA1c_level blood_glucose_level diabetes
1       0  80            0             1           never  25.19         6.6                 140        0
2       0  54            0             0         No Info  27.32         6.6                  80        0
3       1  28            0             0           never -27.32         5.7                 158        0
4       0  NA            0             0         current  23.45         5.0                 155        0
5       1  76            1             1         current  20.14         4.8                 155        0
6       0  20            0             0           never  27.32         6.6                  85        0
7    <NA>  79            0             0         No Info  23.86         5.7                  85        0
8       1  42            0             0           never  33.64         4.8                 145        0
9       0  32            0             0           never  27.32         5.0                 100        0
10      0  53            0             0           never  27.32         6.1                  85        0
11      0  54            0             0          former  54.70         6.0                 100        0
12      0  78           NA             0          former  36.05         5.0                 130        0
13      0  67            0             0           never  25.69         5.8                 200        0
14      0  76            0             0         No Info  27.32         5.0                 160        0
15   <NA>  78            0             0         No Info  27.32         6.6                 126        0
16      1  15            0             0           never  30.36         6.1                 200        0
17      0  42            0             0           never  24.48         5.7                 158        0
18      0  42            0             0         No Info  27.32         5.7                  80        0
19      1  NA            0             0            ever  25.72         3.5                 159        0
20      1  40            0             0         current  36.38         6.0                  90        0
21      1   5            0             0         No Info  18.80         6.2                  85        0
22      0  69            0             0           never  21.24         4.8                  85        0
23      0  72            0             1          former  27.94         6.5                 130        0
24      0   4            0             0         No Info  13.99         4.0                 140        0
25      1  30            0             0           never  33.76         6.1                 126        0
26      1  40            0             0          former  27.85         5.8                  80        0
27      1  45           NA             0           never  26.47         4.0                 158        0
28      1  43            0             0           never  26.08         6.1                 155        0
29      0  53            0             0         No Info  31.75         4.0                 200        0
30      1  50            0             0         No Info  25.15         4.0                 145        0
31      0  41            0             0         current  22.01         6.2                 126        0
32      0  20            0             0           never  22.19         3.5                 100        0
33      0  76            0             0           never  23.55         5.0                  85        0
34      1   5            0             0         No Info  15.10         5.8                  85        0
35      0  15            0             0         No Info  21.76         4.5                 130        0
36      0  26            0             0           never  21.22         6.6                 200        0
37      1   5            0             0         No Info  27.32         6.6                 130        0
38      0  77            1             1           never  32.02         5.0                 159        0
```

Description:

The "gender" column in the "mydata" dataset has been converted into a categorical variable of type "factor." The levels "Female" and "Male" have been assigned numerical labels, where "Female" is labeled as 0 and "Male" is labeled as 1.

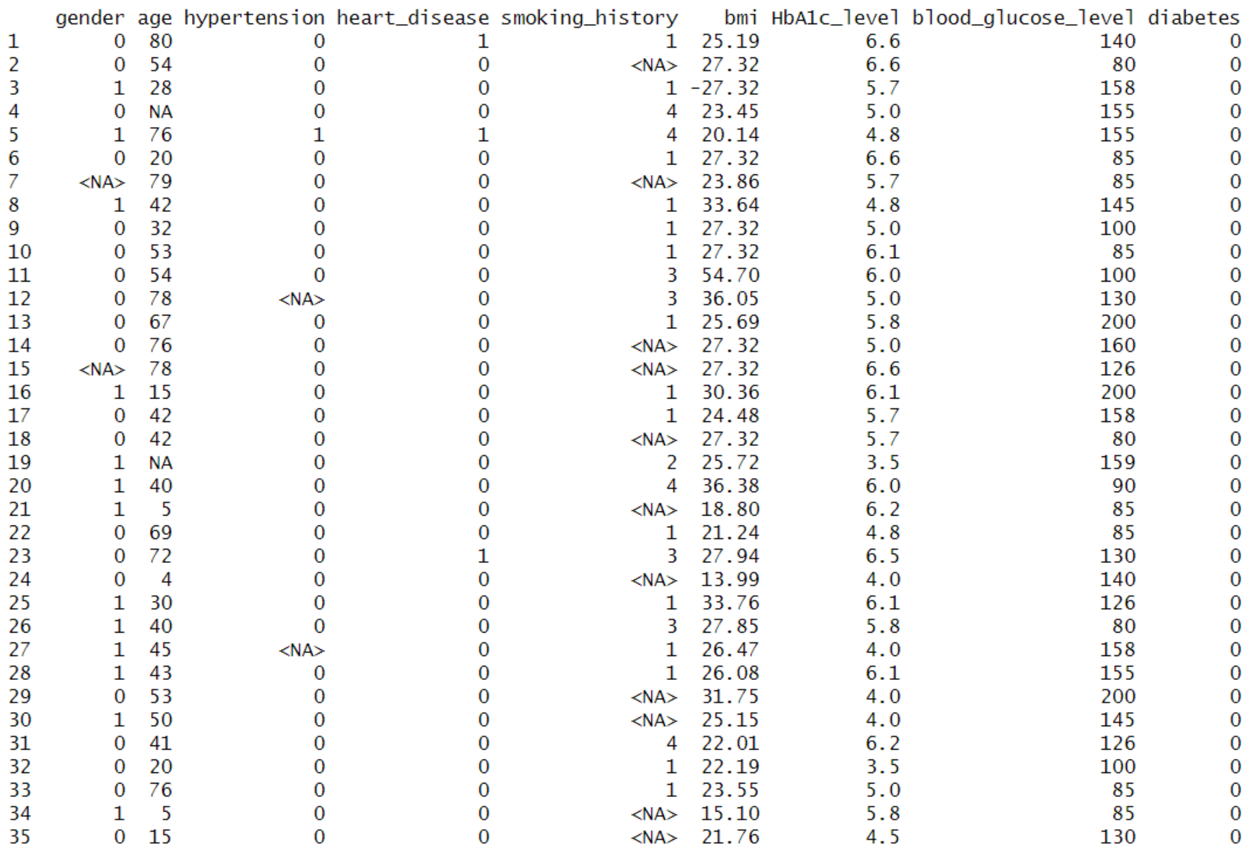
Annotate never as 1, ever as 2, former as 3, current as 4 ,not current as 5 from "smoking_history” attribute:

Code:



```
mydata$smoking_history <- factor(mydata$smoking_history,
                                 levels = c("never","ever","former","current","not current"),
                                 labels = c(1,2,3,4,5))
mydata
```

Output:



```
   gender age hypertension heart_disease smoking_history    bmi HbA1c_level blood_glucose_level diabetes
1       0  80            0             1               1  25.19         6.6                 140        0
2       0  54            0             0            <NA>  27.32         6.6                  80        0
3       1  28            0             0               1 -27.32         5.7                 158        0
4       0  NA            0             0               4  23.45         5.0                 155        0
5       1  76            1             1               4  20.14         4.8                 155        0
6       0  20            0             0               1  27.32         6.6                  85        0
7    <NA>  79            0             0            <NA>  23.86         5.7                  85        0
8       1  42            0             0               1  33.64         4.8                 145        0
9       0  32            0             0               1  27.32         5.0                 100        0
10      0  53            0             0               1  27.32         6.1                  85        0
11      0  54            0             0               3  54.70         6.0                 100        0
12      0  78         <NA>             0               3  36.05         5.0                 130        0
13      0  67            0             0               1  25.69         5.8                 200        0
14      0  76            0             0            <NA>  27.32         5.0                 160        0
15   <NA>  78            0             0            <NA>  27.32         6.6                 126        0
16      1  15            0             0               1  30.36         6.1                 200        0
17      0  42            0             0               1  24.48         5.7                 158        0
18      0  42            0             0            <NA>  27.32         5.7                  80        0
19      1  NA            0             0               2  25.72         3.5                 159        0
20      1  40            0             0               4  36.38         6.0                  90        0
21      1   5            0             0            <NA>  18.80         6.2                  85        0
22      0  69            0             0               1  21.24         4.8                  85        0
23      0  72            0             1               3  27.94         6.5                 130        0
24      0   4            0             0            <NA>  13.99         4.0                 140        0
25      1  30            0             0               1  33.76         6.1                 126        0
26      1  40            0             0               3  27.85         5.8                  80        0
27      1  45         <NA>             0               1  26.47         4.0                 158        0
28      1  43            0             0               1  26.08         6.1                 155        0
29      0  53            0             0            <NA>  31.75         4.0                 200        0
30      1  50            0             0            <NA>  25.15         4.0                 145        0
31      0  41            0             0               4  22.01         6.2                 126        0
32      0  20            0             0               1  22.19         3.5                 100        0
33      0  76            0             0               1  23.55         5.0                  85        0
34      1   5            0             0            <NA>  15.10         5.8                  85        0
35      0  15            0             0            <NA>  21.76         4.5                 130        0
```

Description:

The "smoking_history" column in the "mydata" dataset has been transformed into a categorical variable with numeric labels, representing different smoking categories: "never" (1), "ever" (2), "former" (3), "current" (4), and "not current" (5).

Finding missing values

Code:

```
colSums(is.na(mydata))
```

Output:

```
        gender             age    hypertension   heart_disease smoking_history             bmi 
             2               4               2               0              32               0 
   HbA1c_level blood_glucose_level        diabetes 
             0               0               0 
```

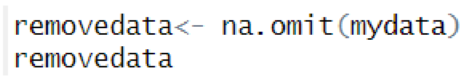
Description:

The code provides the count of missing values (NA) in each column of the "mydata" dataset. Notably, there are 2 missing values in the "gender" column, 4 in the "age" column, 2 in "hypertension," and 32 in "smoking_history." All other columns have no missing values. We will handle these missing values before performing further analysis on the dataset.

HANDLING MISSING VALUES

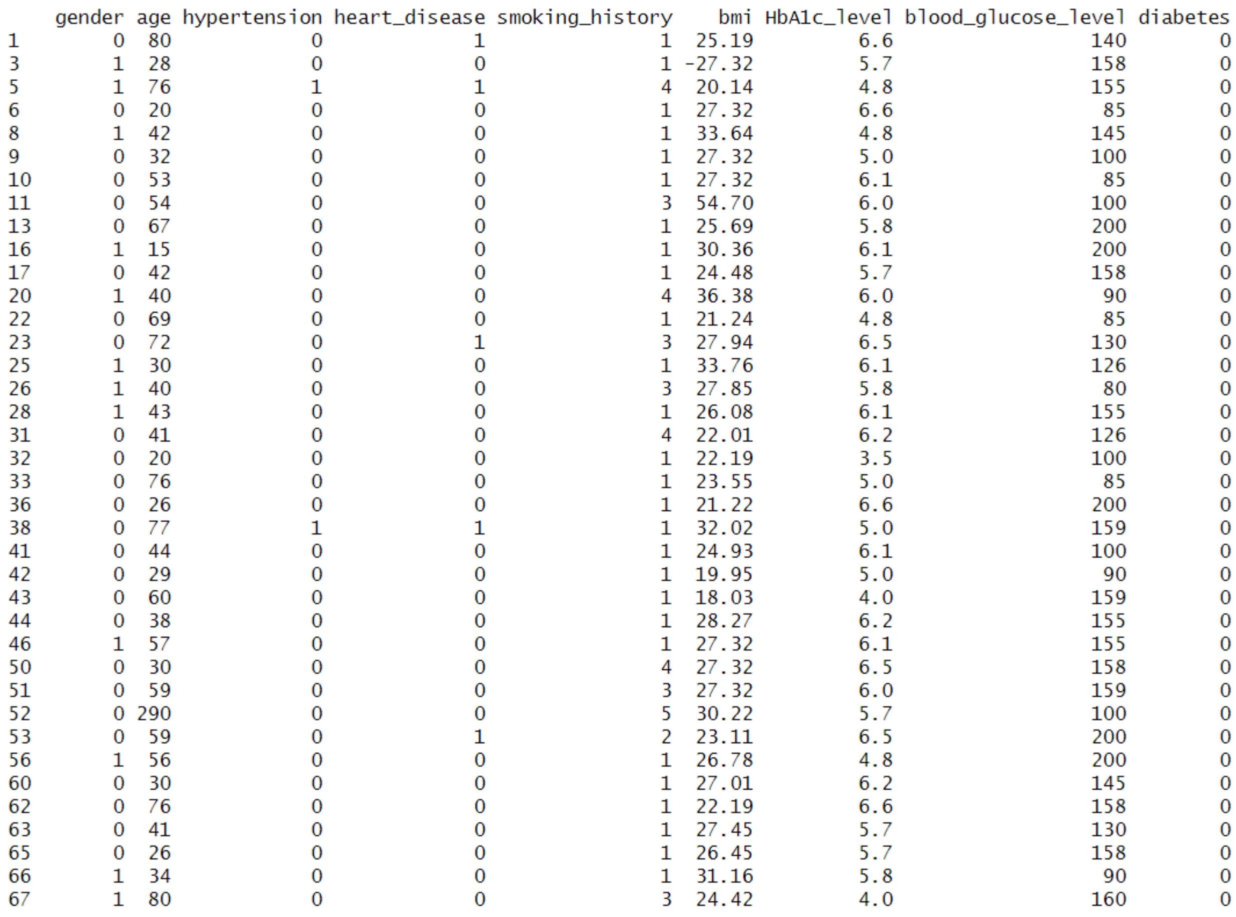
Delete the rows with missing values:

Code:



```
removedata<- na.omit(mydata)
removedata
```

Output:



```
    gender age hypertension heart_disease smoking_history    bmi HbA1c_level blood_glucose_level diabetes
1        0  80            0             1               1  25.19         6.6                 140        0
3        1  28            0             0               1 -27.32         5.7                 158        0
5        1  76            1             1               4  20.14         4.8                 155        0
6        0  20            0             0               1  27.32         6.6                  85        0
8        1  42            0             0               1  33.64         4.8                 145        0
9        0  32            0             0               1  27.32         5.0                 100        0
10       0  53            0             0               1  27.32         6.1                  85        0
11       0  54            0             0               3  54.70         6.0                 100        0
13       0  67            0             0               1  25.69         5.8                 200        0
16       1  15            0             0               1  30.36         6.1                 200        0
17       0  42            0             0               1  24.48         5.7                 158        0
20       1  40            0             0               4  36.38         6.0                  90        0
22       0  69            0             0               1  21.24         4.8                  85        0
23       0  72            0             1               3  27.94         6.5                 130        0
25       1  30            0             0               1  33.76         6.1                 126        0
26       1  40            0             0               3  27.85         5.8                  80        0
28       1  43            0             0               1  26.08         6.1                 155        0
31       0  41            0             0               4  22.01         6.2                 126        0
32       0  20            0             0               1  22.19         3.5                 100        0
33       0  76            0             0               1  23.55         5.0                  85        0
36       0  26            0             0               1  21.22         6.6                 200        0
38       0  77            1             1               1  32.02         5.0                 159        0
41       0  44            0             0               1  24.93         6.1                 100        0
42       0  29            0             0               1  19.95         5.0                  90        0
43       0  60            0             0               1  18.03         4.0                 159        0
44       0  38            0             0               1  28.27         6.2                 155        0
46       1  57            0             0               1  27.32         6.1                 155        0
50       0  30            0             0               4  27.32         6.5                 158        0
51       0  59            0             0               3  27.32         6.0                 159        0
52       0 290            0             0               5  30.22         5.7                 100        0
53       0  59            0             1               2  23.11         6.5                 200        0
56       1  56            0             0               1  26.78         4.8                 200        0
60       0  30            0             0               1  27.01         6.2                 145        0
62       0  76            0             0               1  22.19         6.6                 158        0
63       0  41            0             0               1  27.45         5.7                 130        0
65       0  26            0             0               1  26.45         5.7                 158        0
66       1  34            0             0               1  31.16         5.8                  90        0
67       1  80            0             0               3  24.42         4.0                 160        0
```

Description:

The "removedata" dataset is removing rows with missing values (NA) from the "mydata" dataset. This operation effectively eliminates rows containing incomplete data, resulting in a dataset with complete observations for further analysis or modeling.

'mydata' Dataset as 'meanData'

In meanData set we will replace all the missing values with mean value.

```
meanData <-mydata
meanData
```

Recovering Missing Values with Mean

Code:

```
meanData$gender=as.numeric(meanData$gender)
Gendermean=mean(meanData$gender , na.rm = TRUE)
missingGenderMean= meanData$gender[is.na(meanData$gender)] <- Gendermean
meanData

Agemean=mean(meanData$age , na.rm = TRUE)
missingAgeMean= meanData$age[is.na(meanData$age)] <- Agemean
meanData

meanData$hypertension=as.numeric(meanData$hypertension)
Hypertensionmean=mean(meanData$hypertension , na.rm = TRUE)
missingHypertensionMean= meanData$hypertension[is.na(meanData$hypertension)] <- Hypertensionmean
meanData

meanData$smoking_history=as.numeric(meanData$smoking_history)
Smokinghistorymean=mean(meanData$smoking_history , na.rm = TRUE)
SmokinghistoryMean= meanData$smoking_history[is.na(meanData$smoking_history)] <- Smokinghistorymean
meanData
```

Output:

```
     gender      age hypertension heart_disease smoking_history   bmi HbA1c_level blood_glucose_level diabetes
1  1.000000 80.00000   0.00000000             1               1  25.19         6.6                 140        0
2  1.000000 54.00000   0.00000000             0               2  27.32         6.6                  80        0
3  2.000000 28.00000   0.00000000             0               1 -27.32         5.7                 158        0
4  1.000000 54.17241   0.00000000             0               4  23.45         5.0                 155        0
5  2.000000 76.00000   1.00000000             1               4  20.14         4.8                 155        0
6  1.000000 20.00000   0.00000000             0               1  27.32         6.6                  85        0
7  1.398305 79.00000   0.00000000             0               2  23.86         5.7                  85        0
8  2.000000 42.00000   0.00000000             0               1  33.64         4.8                 145        0
9  1.000000 32.00000   0.00000000             0               1  27.32         5.0                 100        0
10 1.000000 53.00000   0.00000000             0               1  27.32         6.1                  85        0
11 1.000000 54.00000   0.00000000             0               3  54.70         6.0                 100        0
12 1.000000 78.00000   0.08474576             0               3  36.05         5.0                 130        0
13 1.000000 67.00000   0.00000000             0               1  25.69         5.8                 200        0
14 1.000000 76.00000   0.00000000             0               2  27.32         5.0                 160        0
15 1.398305 78.00000   0.00000000             0               2  27.32         6.6                 126        0
16 2.000000 15.00000   0.00000000             0               1  30.36         6.1                 200        0
17 1.000000 42.00000   0.00000000             0               1  24.48         5.7                 158        0
18 1.000000 42.00000   0.00000000             0               2  27.32         5.7                  80        0
19 2.000000 54.17241   0.00000000             0               2  25.72         3.5                 159        0
20 2.000000 40.00000   0.00000000             0               4  36.38         6.0                  90        0
21 2.000000  5.00000   0.00000000             0               2  18.80         6.2                  85        0
22 1.000000 69.00000   0.00000000             0               1  21.24         4.8                  85        0
23 1.000000 72.00000   0.00000000             1               3  27.94         6.5                 130        0
24 1.000000  4.00000   0.00000000             0               2  13.99         4.0                 140        0
25 2.000000 30.00000   0.00000000             0               1  33.76         6.1                 126        0
```

Description:

In this code, missing values in the "gender," "age," "hypertension," and "smoking_history" columns of the "meanData" dataset are imputed using their respective column means. The dataset is first converted to numeric values, and then missing values are replaced with the calculated means. This process ensures that the dataset now contains imputed values for these columns, making it suitable for analysis.

'mydata' Dataset as 'medianData'

In medianData set we will replace all the missing values with median value.

```
medianData <-mydata
medianData
```

Recovering Missing Values with Median

Code:

```
medianData$gender=as.numeric(medianData$gender)
Gendermedian=median(medianData$gender , na.rm = TRUE)
missingGenderMedian= medianData$gender[is.na(medianData$gender)] <- Gendermedian
medianData

Agemedian=median(medianData$age , na.rm = TRUE)
missingAgeMedian= medianData$age[is.na(medianData$age)] <- Agemedian
medianData

medianData$hypertension=as.numeric(medianData$hypertension)
Hypertensionmedian=median(medianData$hypertension , na.rm = TRUE)
missingHypertensionMedian= medianData$hypertension[is.na(medianData$hypertension)] <- Hypertensionmedian
medianData

medianData$smoking_history=as.numeric(medianData$smoking_history)
Smokinghistorymedian=median(medianData$smoking_history , na.rm = TRUE)
SmokinghistoryMedian= medianData$smoking_history[is.na(medianData$smoking_history)] <- Smokinghistorymedian
medianData
```

Output:

```
   gender  age hypertension heart_disease smoking_history    bmi HbA1c_level blood_glucose_level diabetes
1       1 80.0            0             1               1  25.19         6.6                 140        0
2       1 54.0            0             0               1  27.32         6.6                  80        0
3       2 28.0            0             0               1 -27.32         5.7                 158        0
4       1 52.5            0             0               4  23.45         5.0                 155        0
5       2 76.0            1             1               4  20.14         4.8                 155        0
6       1 20.0            0             0               1  27.32         6.6                  85        0
7       1 79.0            0             0               1  23.86         5.7                  85        0
8       2 42.0            0             0               1  33.64         4.8                 145        0
9       1 32.0            0             0               1  27.32         5.0                 100        0
10      1 53.0            0             0               1  27.32         6.1                  85        0
11      1 54.0            0             0               3  54.70         6.0                 100        0
12      1 78.0            0             0               3  36.05         5.0                 130        0
13      1 67.0            0             0               1  25.69         5.8                 200        0
14      1 76.0            0             0               1  27.32         5.0                 160        0
15      1 78.0            0             0               1  27.32         6.6                 126        0
16      2 15.0            0             0               1  30.36         6.1                 200        0
17      1 42.0            0             0               1  24.48         5.7                 158        0
18      1 42.0            0             0               1  27.32         5.7                  80        0
19      2 52.5            0             0               2  25.72         3.5                 159        0
20      2 40.0            0             0               4  36.38         6.0                  90        0
21      2  5.0            0             0               1  18.80         6.2                  85        0
22      1 69.0            0             0               1  21.24         4.8                  85        0
23      1 72.0            0             1               3  27.94         6.5                 130        0
24      1  4.0            0             0               1  13.99         4.0                 140        0
25      2 30.0            0             0               1  33.76         6.1                 126        0
```

Description: The "medianData" dataset is being processed for imputing missing values in the "gender" "age," "hypertension," and "smoking_history" columns using their respective column medians. First, the column is converted to numeric values using `as.numeric()`. Then, the variable is assigned the median value of the specific column, with missing values (NA) omitted during the calculation. This step is part of the process to impute missing values in the column using the median value.

'mydata' Dataset as 'modeData'

In modeData set we will replace all the missing values with mode value.

```
modeData <-mydata
modeData
```

Recovering missing values with Mode

Code:

```
mode <- function(x){
  unique_x <- unique(x)
  tabulate_x <-tabulate(match(x,unique_x))
  unique_x[tabulate_x == max(tabulate_x)]
}
modeData$gender=as.numeric(modeData$gender)
genderMode = mode(modeData$gender)
missingGenderMode= modeData$gender[is.na(modeData$gender)] <- genderMode
modeData

ageMode = mode(modeData$age)
missingAgeMode= modeData$age[is.na(modeData$age)] <- ageMode
modeData

modeData$hypertension=as.numeric(modeData$hypertension)
hypertensionMode = mode(modeData$hypertension)
missingHypertensionMode= modeData$hypertension[is.na(modeData$hypertension)] <- hypertensionMode
modeData

smoking_historyMode = mode(modeData$smoking_history)
missingSmoking_historyMode= modeData$smoking_history[is.na(modeData$smoking_history)] <- smoking_historyMode
modeData
```

Output:

```
    gender age hypertension heart_disease smoking_history    bmi HbA1c_level blood_glucose_level diabetes
1        1  80            0             1               1  25.19         6.6                 140        0
2        1  54            0             0               1  27.32         6.6                  80        0
3        2  28            0             0               1 -27.32         5.7                 158        0
4        1  43            0             0               4  23.45         5.0                 155        0
5        2  76            1             1               4  20.14         4.8                 155        0
6        1  20            0             0               1  27.32         6.6                  85        0
7        1  79            0             0               1  23.86         5.7                  85        0
8        2  42            0             0               1  33.64         4.8                 145        0
9        1  32            0             0               1  27.32         5.0                 100        0
10       1  53            0             0               1  27.32         6.1                  85        0
11       1  54            0             0               3  54.70         6.0                 100        0
12       1  78            0             0               3  36.05         5.0                 130        0
13       1  67            0             0               1  25.69         5.8                 200        0
14       1  76            0             0               1  27.32         5.0                 160        0
15       1  78            0             0               1  27.32         6.6                 126        0
16       2  15            0             0               1  30.36         6.1                 200        0
17       1  42            0             0               1  24.48         5.7                 158        0
18       1  42            0             0               1  27.32         5.7                  80        0
19       2  43            0             0               2  25.72         3.5                 159        0
20       2  40            0             0               4  36.38         6.0                  90        0
21       2   5            0             0               1  18.80         6.2                  85        0
22       1  69            0             0               1  21.24         4.8                  85        0
23       1  72            0             1               3  27.94         6.5                 130        0
24       1   4            0             0               1  13.99         4.0                 140        0
25       2  30            0             0               1  33.76         6.1                 126        0
```

Description: In this code, a custom "mode" function is defined to calculate the mode of a given vector. The code proceeds to apply this function to impute missing values in the "gender," "age," "hypertension," and "smoking_history" columns within the "modeData" dataset. Mode represents the most frequently occurring value in each column. The code replaces missing values with the calculated modes, ensuring that the dataset is completer and more suitable for analysis.

Replacing categorical missing data by mode:

(Since we can use mode both for numerical and categorical value at first, we replaced the missing values with numerical value. Now we will replace it with categorical value for gender and smoking_history attribute).

Code:

```
data3<- read.csv("D:/Data.csv",header = TRUE,sep = ",")
data3
colSums(is.na(data3))
data3$gender[data3$gender == ""] <- 'NA'
data3
data3$gender <- factor(data3$gender,
                       levels = c("Male","Female"),
                       labels = c(1,2))
data3
mode <- function(v) {
  uniquevalue <- na.omit(unique(v))
  uniquevalue[which.max(tabulate(match(v, uniquevalue)))] }
Gendermode = mode(data3$gender)
missinggenderMode= data3$gender[is.na(data3$gender)] <- Gendermode
data3
data3$gender <- factor(data3$gender,
                       levels = c(1,2),
                       labels = c("Male","Female"))
data3
```

Output:

```
   gender age hypertension heart_disease smoking_history   bmi HbA1c_level blood_glucose_level diabetes
1  Female  80            0             1           never  25.19         6.6                 140        0
2  Female  54            0             0         No Info  27.32         6.6                  80        0
3    Male  28            0             0           never -27.32         5.7                 158        0
4  Female  NA            0             0         current  23.45         5.0                 155        0
5    Male  76            1             1         current  20.14         4.8                 155        0
6  Female  20            0             0           never  27.32         6.6                  85        0
7  Female  79            0             0         No Info  23.86         5.7                  85        0
8    Male  42            0             0           never  33.64         4.8                 145        0
9  Female  32            0             0           never  27.32         5.0                 100        0
10 Female  53            0             0           never  27.32         6.1                  85        0
11 Female  54            0             0          former  54.70         6.0                 100        0
12 Female  78           NA             0          former  36.05         5.0                 130        0
13 Female  67            0             0           never  25.69         5.8                 200        0
14 Female  76            0             0         No Info  27.32         5.0                 160        0
15 Female  78            0             0         No Info  27.32         6.6                 126        0
16   Male  15            0             0           never  30.36         6.1                 200        0
17 Female  42            0             0           never  24.48         5.7                 158        0
18 Female  42            0             0         No Info  27.32         5.7                  80        0
19   Male  NA            0             0            ever  25.72         3.5                 159        0
20   Male  40            0             0         current  36.38         6.0                  90        0
```

Description: The missing values in the "gender" column of the "data3" dataset are first identified and replaced with 'NA'. Then, the "gender" column is converted into a factor variable with numeric values (1 for Male, 2 for Female). The mode function is defined to calculate the mode for the "gender" column. The missing values in the "gender" column are imputed in the calculated mode, ensuring that the column is complete. Finally, the "gender" column is converted back to a factor with appropriate labels (Male and Female). This code prepares the "gender" column for analysis by handling missing values and converting it to a factor variable with meaningful labels.

Correcting Invalid Values

Code:

```
options(max.print = 10000)
data1 <- read.csv("D:/Data.csv", header = TRUE, sep = ",")
data1
data1$age[data1$age == "290"] <- "29"
data1$age[data1$age == "280"] <- "28"
data1$bmi[data1$bmi == "-27.32"] <- "27.32"
data1
```

Output:

```
    gender  age hypertension heart_disease smoking_history   bmi HbA1c_level blood_glucose_level diabetes

52   Female   29            0             0     not current 30.22         5.7                 100        0

119 Female    28            0             0         No Info 27.32         8.8                 159        1

3      Male   28            0             0           never 27.32         5.7                 158        0
```

Description: The `max.print` option is set to display up to 10,000 rows. The "data1" dataset is loaded from "Data.csv." Subsequently, data cleaning is performed by replacing incorrect values in the "age" and "bmi" columns to ensure data consistency and accuracy. The first line replaces the value "290" in the "age" column with "29,". The second line replaces "280" in the "age" column with "28," ensuring data consistency. And the third line replaces "-27.32" in the "bmi" column with "27.32," likely rectifying a data entry error.

Deleting Invalid Values

Code:

```
data2<- read.csv("D:/Data.csv",header = TRUE,sep = ",")
delete_row <-data2[c(1,2,4:51,53:118,120),]
delete_row
updateData <-delete_row
updateData
```

Output:

```
    gender  age hypertension heart_disease smoking_history   bmi  HbA1c_level blood_glucose_level diabetes
1   Female  80            0             1           never 25.19         6.6                 140        0
2   Female  54            0             0         No Info 27.32         6.6                  80        0
4   Female  NA            0             0         current 23.45         5.0                 155        0
5     Male  76            1             1         current 20.14         4.8                 155        0

50  Female  30            0             0         current 27.32         6.5                 158        0
51  Female  59            0             0          former 27.32         6.0                 159        0
53  Female  59            0             1            ever 23.11         6.5                 200        0
54  Female  19            0             0                 27.32         5.7                 145        0

118   Male  43            0             0            ever 19.46         9.0                 130        1
120 Female  43            0             0         No Info 27.32         5.8                 159        1
```

Description: This code performs data manipulation by removing specific rows from the dataset and creating a new variable ("updateData") with the modified data.

Measure of Central Tendency

Calculate the mean for each attribute:

Code:

```
means <- colMeans(meanData)
print(means)
```

Output:

```
       gender               age      hypertension     heart_disease   smoking_history               bmi
   1.39830508       54.17241379        0.08474576        0.06666667        2.00000000       27.65925000
  HbA1c_level blood_glucose_level        diabetes
   6.27500000      156.75000000        0.43333333
```

Description: The "means" variable contains the column meaning of the "meanData" dataset. It provides the average values for each column, making it useful for summarizing the central tendencies of the dataset.

Calculate the median for each attribute using apply

Code:

```
medians <- apply(medianData, 2, median)
print(medians)
```

Output:

```
             gender                 age        hypertension       heart_disease     smoking_history                 bmi         HbA1c_level
               1.00               52.50                0.00                0.00                1.00               27.32                6.20
blood_glucose_level            diabetes
             155.00                0.00
```

Description: The "medians" variable holds the median values calculated for each column of the "medianData" dataset. The "medians" variable provides the median values for each column in the "medianData" dataset. For instance, the median "age" is 52.50, and the median "bmi" is 27.32. This summary helps understand the central tendencies of the data, with medians representing the middle values in each column.

Calculate the mode for each attribute using apply

Code:

```
modes <- apply(modeData, 2, mode)
print(modes)
```

Output:

```
          gender             age    hypertension   heart_disease smoking_history             bmi     HbA1c_level 
             "1"           " 43"             "0"             "0"             "1"         " 27.32"           "6.6" 
blood_glucose_level       diabetes 
           "159"             "0" 
```

Description: In the "modes" variable, the most frequently occurring values for each column of the "modeData" dataset have been calculated. The result shows the modes for the columns, such as the mode "gender" being "1" (likely representing a category), and "age" being "43." This provides insight into the most common values within each column, which can be useful for understanding the data's central tendencies and frequent occurrences.

Measure of Spread

'medianData' Dataset as 'newData'

```
newData <-medianData
newData
```

Calculating the range

Code:

```
numeric_data <- newData[, sapply(newData, is.numeric)]
ranges <- sapply(numeric_data, range)
print(ranges)
```

Output:

```
     gender age hypertension heart_disease smoking_history    bmi HbA1c_level blood_glucose_level diabetes
[1,]      1   3            0             0               1 -27.32         3.5                  80        0
[2,]      2 290            1             1               5  63.48         9.0                 300        1
```

Description: In this code, a new dataset "numeric_data" is created by selecting only the numeric columns from the "newData" dataset. The "Range" values for various numeric columns are calculated using the `range()` function. The output shows the minimum and maximum values for each column, providing insights into the data's range or spread: For instance, in the "age" column, the range is from 3 to 290. In the "bmi" column, the range is from -27.32 to 63.48. The "blood_glucose_level" column has a range from 80 to 30048. This information is useful for understanding the variability and bounds of the data in each column.

Calculating the variance

Code:

```
variances <- sapply(numeric_data, var)
print(variances)
```

Output:

```
      gender                 age        hypertension       heart_disease     smoking_history                 bmi
2.402661e-01        1.337138e+03        7.703081e-02        6.274510e-02        1.524930e+00        7.796880e+01
 HbA1c_level blood_glucose_level            diabetes
1.910294e+00        2.573853e+03        2.476190e-01
```

Description: In this code, the variance values for various numeric columns are calculated using the `var()` function and displayed. Variance quantifies the degree of data spread or variability in each column. For instance, the variance for "age" is approximately 1,341.103, suggesting significant variability in age values, while the variance for "heart_disease" is around 0.0627, indicating relatively lower variability in that column.

Calculating the standard deviation

Code:

```
std_devs <- sapply(numeric_data, sd)
print(std_devs)
```

Output:

```
     gender                 age        hypertension       heart_disease     smoking_history                 bmi
  0.4901695          36.5668912           0.2775443           0.2504897           1.2348805           8.8299945
HbA1c_level blood_glucose_level            diabetes
  1.3821339          50.7331543           0.4976134
```

Description: In this line of code, the standard deviations (std_devs) are calculated for each numeric column within the "numeric_data" dataset. This step helps quantify the dispersion or variability of data in each column. These standard deviation values represent the degree of variability in each respective numeric column. Larger standard deviations typically indicate greater data variability, while smaller values suggest less variability or more consistent data.

Histogram

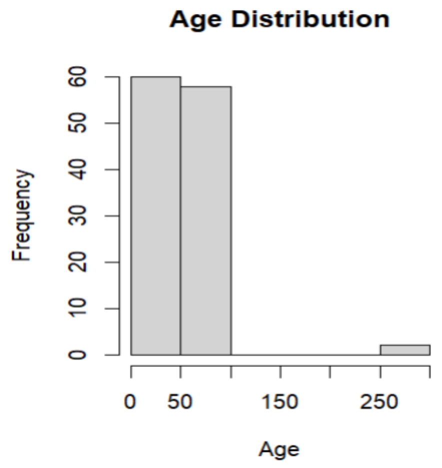
Histogram is suitable for continuous or numerical data. It displays the distribution and frequency of data values within specific bins or intervals. Here we used a dataset(modeData) where the missing values replaced with mode values.

AGE:

Code:

```
hist(modeData$age, main = "Age Distribution", xlab = "Age", ylab = "Frequency")
```

Output:



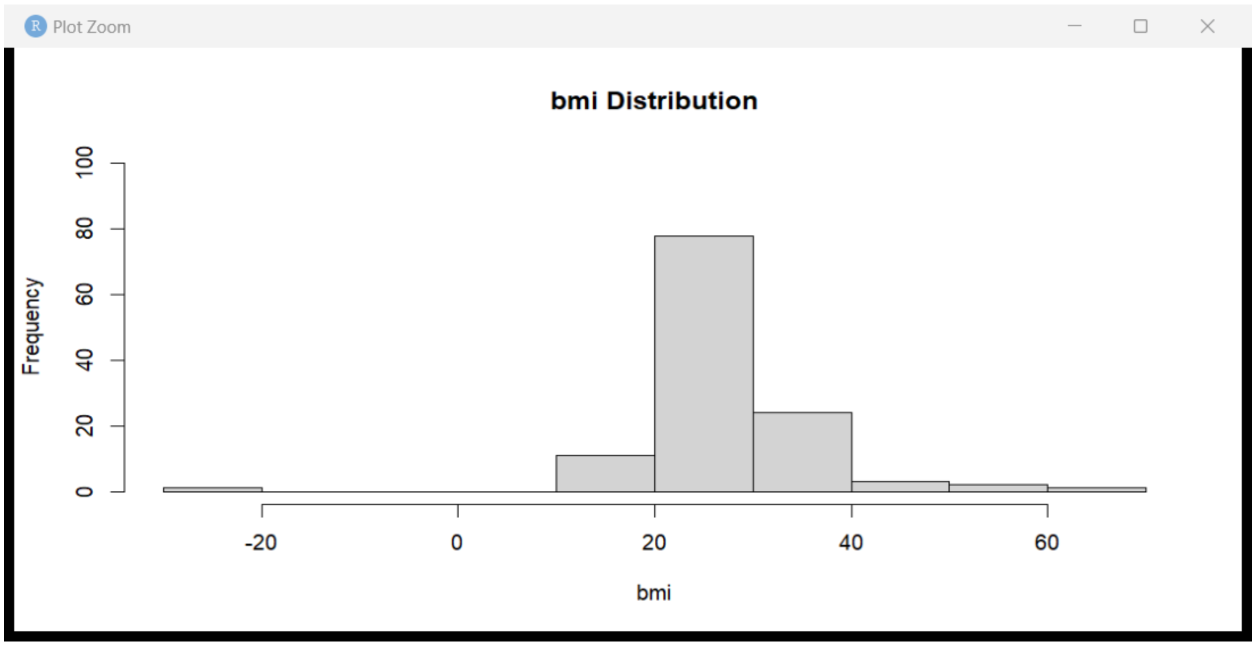
Description: In this code, a histogram is created to visualize the distribution of the "age" attribute in the "modeData" dataset. The title of the histogram is set as "Age Distribution," with "Age" displayed on the x-axis and "Frequency" on the y-axis. This histogram provides a visual representation of the frequency of different age values in the dataset. From visualization we can see the age values in dataset are between 0 to 100.Aroud 60 age values are between 0 to 50 and rest of them are between 50 to 100. There are two/three values that are between 250 to 300.

BMI:

Code:

```
hist(modeData$bmi, main = "bmi Distribution", xlab = "bmi", ylab = "Frequency" , ylim = c(0,100))
```

Output:



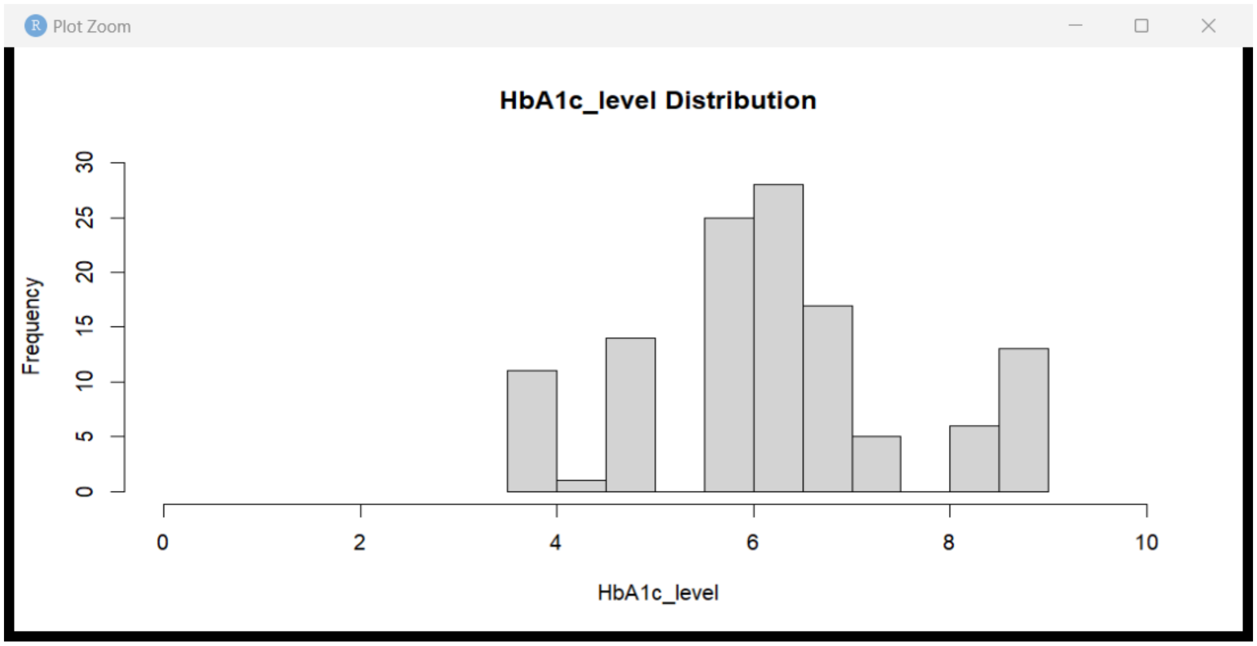
Description: Here we visualized bmi distribution data. We used ylim to set the frequency range. From data distribution we can see most of the data values are between in 20 to 30 range. The range between 30 to 40 is the second highest and there is a negative value in our data.

HbA1c Level:

Code:

```
hist(modeData$ HbA1c_level, main = " HbA1c_level Distribution", xlab = " HbA1c_level", ylab = "Frequency", xlim = c(0,10), ylim = c(0,30))
```

Output:



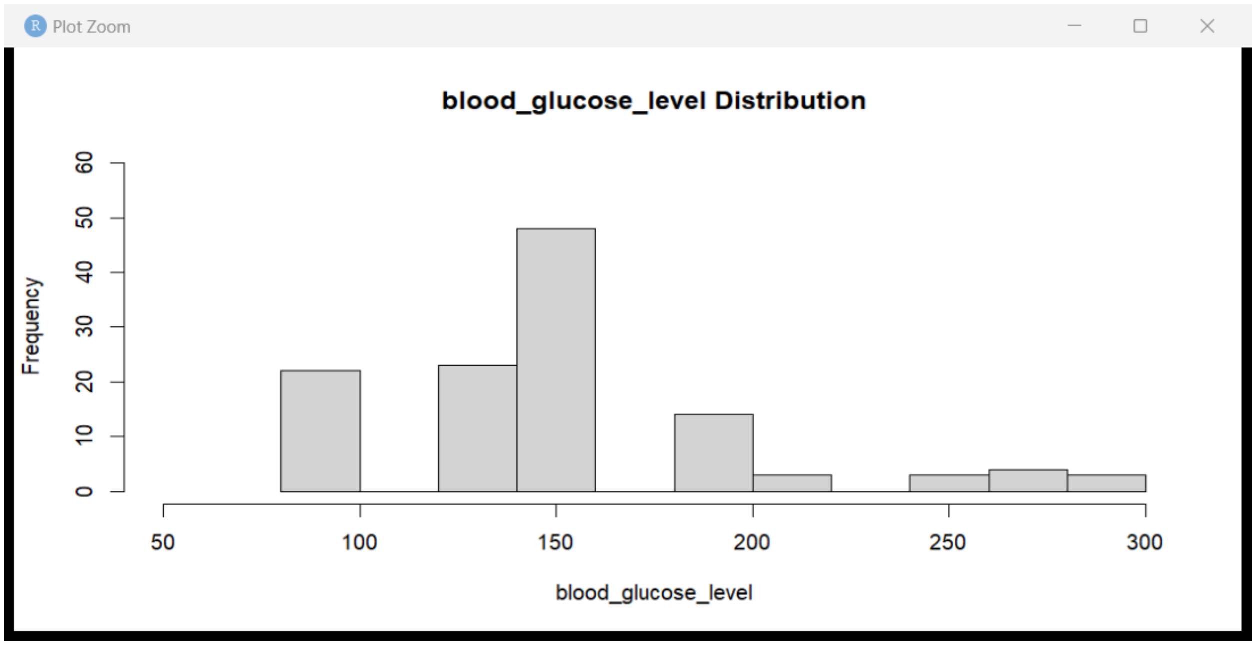
Description: HbA1c_level distribution started from range 3.5 and ended at 9. In between 5 to 5.5 and 7.5 to 8 range there were no data found.

BLOOD GLUCOSE LEVEL:

Code:

```
hist(modeData$blood_glucose_level, main = "blood_glucose_level Distribution", xlab = "blood_glucose_level", ylab = "Frequency", xlim = c(50,300), ylim = c(0,60))
```

Output:



Description was we plotted blood glucose level data. We used xlim(50,300) to set the values range between 50 to 300 and ylim(0,60) to set the frequency range between 0 to 60.

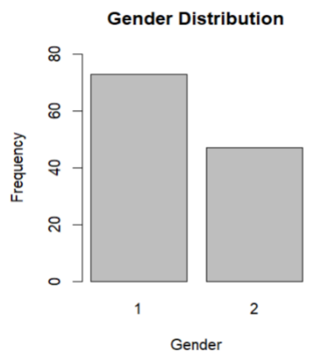
Bar Graph: Bar graph is suitable for categorical or nominal data. It represents the frequency or count different categories or groups. Mostly used for showing relationships between different categories. Here we used medianData set to replace all the null values with median value.

GENDER:

Code:

```
barplot(table(medianData$gender), main = "Gender Distribution", xlab = "Gender", ylab = "Frequency", ylim = c(0, 80))
```

Output:



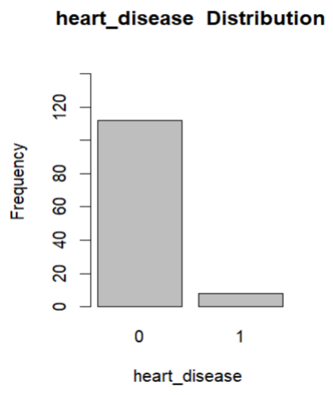
Description: barplot(….) function is used to create a bar plot.table(medianData$gender) creates a table that counts the frequency of each unique value in the mentioned variable. We annoted female by 1 and male by 2

HEART DIASEASE:

Code:

```
barplot(table(medianData$heart_disease ), main = "heart_disease  Distribution", xlab = "heart_disease ", ylab = "Frequency",ylim = c(0, 150))
```

Output:



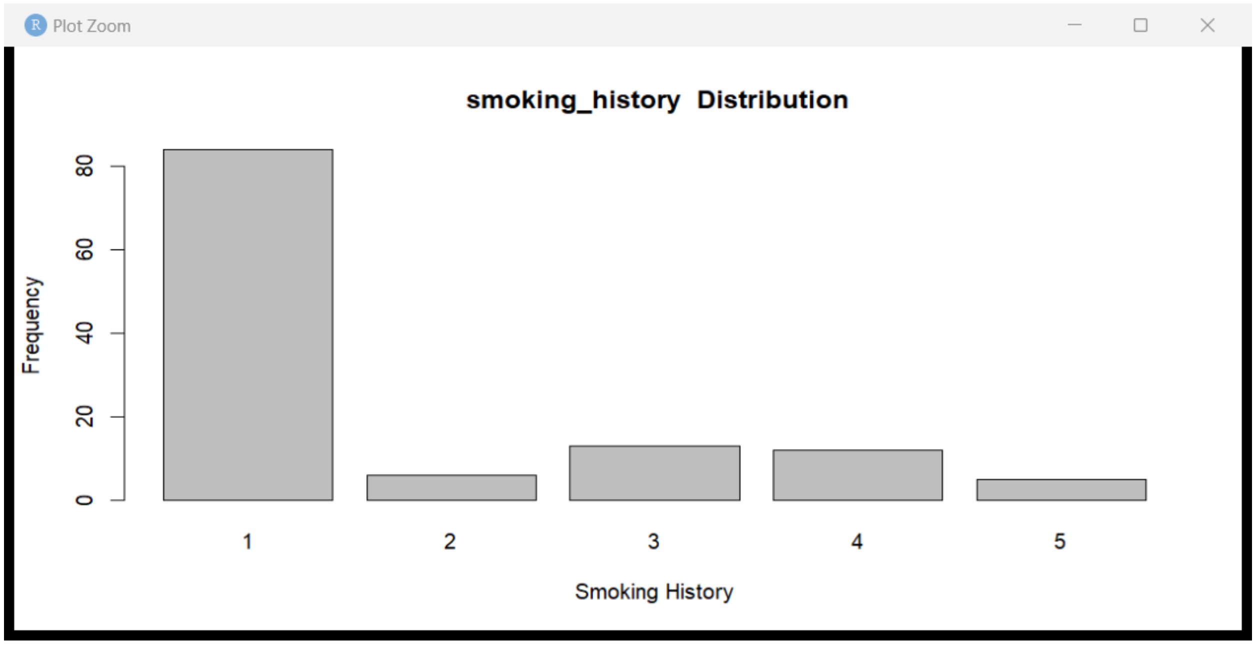
Description: barplot(….) function is used to create a bar plot.table(medianData$heart_disease) creates a table that counts the frequency of each unique value in the mentioned variable.

SMOKING HISTROY:

Code:

```
barplot(table(medianData$smoking_history ), main = " smoking_history  Distribution", xlab = "Smoking History", ylab = "Frequency")
```

Output:



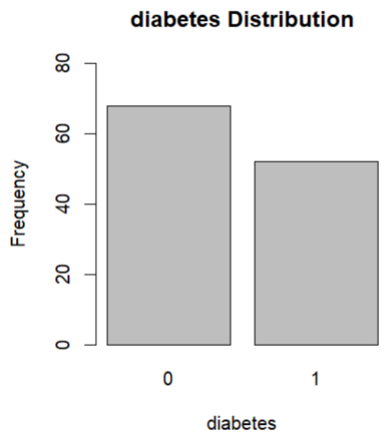
Description: Here we have visualized data using bar graph. At first, we removed all the null values from the dataset. Then used barplot. Barplot(….) function is used to create a bar plot.table(data4$smoking_history) creates a table that counts the frequency of each unique value in the mentioned variable. Here we have Annotated never as 1, ever as 2, former as 3, current as 4, not current as 5 from "smoking_history" attribute.

DIABETES:

Code:

```
barplot(table(medianData$diabetes), main = "diabetes Distribution", xlab = "diabetes", ylab = "Frequency",ylim = c(0, 80))
```

Output:



Description: Barplot(….) function is used to create a bar plot.table(medianData$diabetes) creates a table that counts the frequency of each unique value in the mentioned variable.

Visualizing outliers via boxplot

Boxplot is used for detecting outliers for numerical values in dataset. Here at first we applied boxplot for each data attribute and then removed outliers later where necessary.

' medianData ' Dataset as ' plotData '(we are replacing missing values with median data and stored it into plotdata)

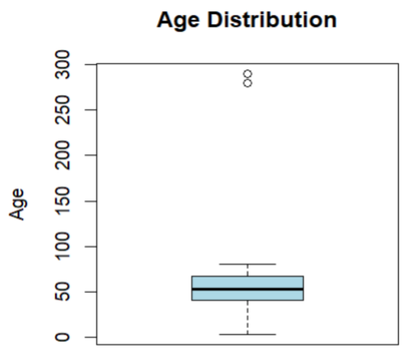
```
plotData<-medianData
plotData
```

AGE:

Code:

```
boxplot(plotData$age,
        main = "Age Distribution",
        ylab = "Age",
        col = "lightblue")
```

Output:



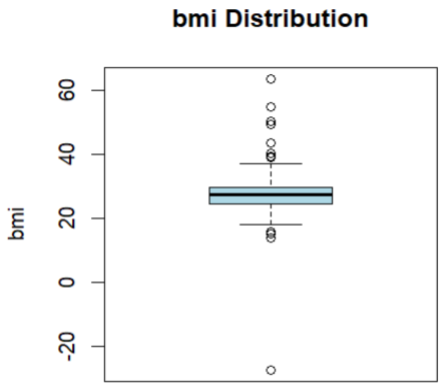
Description: plotData is used to create a box plot. plotData$age specifies the numeric variable age in the dataset. Main sets the title. Ylab sets the y-axis label.col=lightblue sets the color of the boxplot to light blue. We can see outliers here. There are two values which are out of the maximum range of boxplot

BMI:

Code:

```
boxplot(plotData$bmi,
        main = "bmi Distribution",
        ylab = "bmi",
        col = "lightblue")
```

Output:



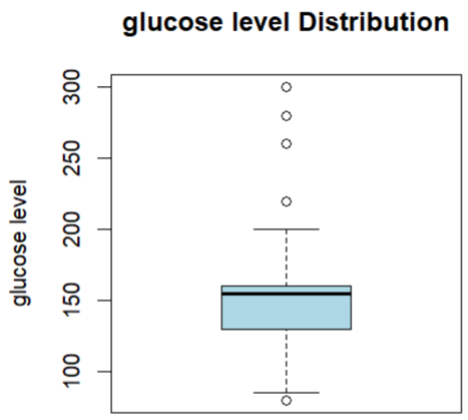
Description: In bmi distribution there are lot of outliers. Most of the outliers are out of max range of boxplot. However there are also some values which are out of min range of box plot.

BLOOD GLUCOSE LEVEL:

Code:

```
boxplot(plotData$blood_glucose_level,
        main = "glucose level Distribution",
        ylab = "glucose level",
        col = "lightblue")
```

Output:



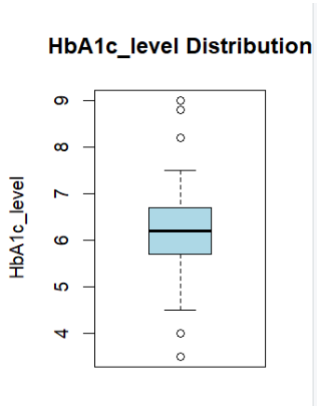
Description: In glucose distribution there are lot of outliers. Most of the outliers are out of max range of boxplot. On the other hand, there is only one value which is out of min range of box plot.

HbA1c_level

Code:

```
boxplot(plotData$HbA1c_level,
        main = "HbA1c_level Distribution",
        ylab = "HbA1c_level",
        col = "lightblue")
```

Output:



Description: In HbA1c_level distribution there are five outliers. Three outliers are out of max range of boxplot. On the other hand, two value which is out of min range of box plot.

Removing outliers

AGE:

Code:

```
data <- plotData$age
length(data)
boxplot(data, plot = FALSE)$out
outliers <- boxplot(data, plot = FALSE)$out
data_no_outlier <- data[-which(data %in% outliers)]
boxplot(data_no_outlier, plot = FALSE)$out
length(data_no_outlier)
```

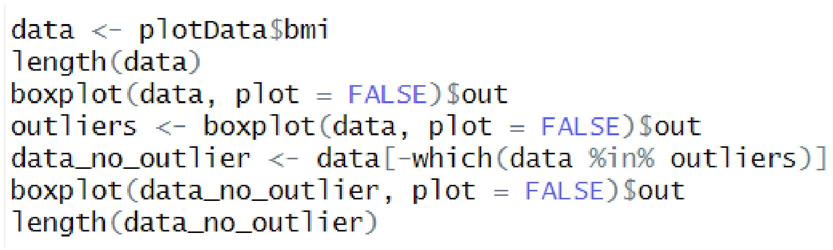
Output:

```
> data <- plotData$age
> length(data)
[1] 120
> boxplot(data, plot = FALSE)$out
[1] 290 280
> outliers <- boxplot(data, plot = FALSE)$out
> data_no_outlier <- data[-which(data %in% outliers)]
> boxplot(data_no_outlier, plot = FALSE)$out
numeric(0)
> length(data_no_outlier)
[1] 118
```

Description: The dataset "data" contains 120 values. By using a boxplot, in age column two outliers with values 290 and 280 are identified and stored in the "outliers" variable. These outliers are then removed from the "data" dataset to create a new dataset called "data_no_outlier." A second boxplot for "data_no_outlier" confirms the absence of outliers, and the length of the cleaned dataset is determined to be 118, indicating that the outliers have been successfully removed. This code showcases the process of outlier detection and data cleaning.

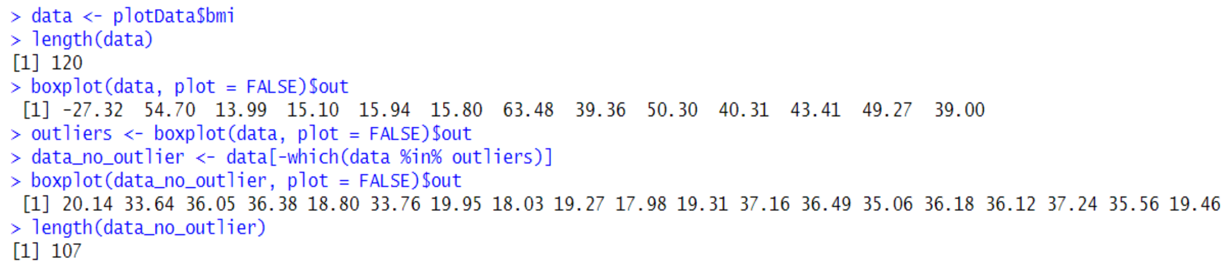
BMI:

Code:



```
data <- plotData$bmi
length(data)
boxplot(data, plot = FALSE)$out
outliers <- boxplot(data, plot = FALSE)$out
data_no_outlier <- data[-which(data %in% outliers)]
boxplot(data_no_outlier, plot = FALSE)$out
length(data_no_outlier)
```

Output:

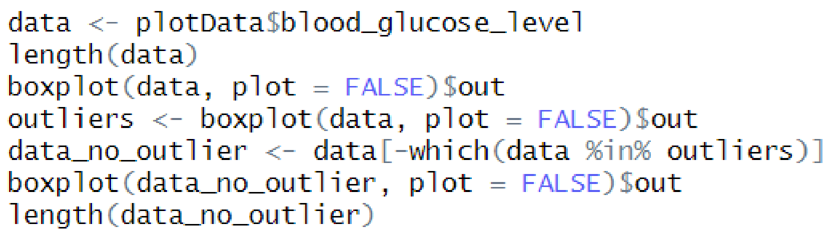


```
> data <- plotData$bmi
> length(data)
[1] 120
> boxplot(data, plot = FALSE)$out
 [1] -27.32  54.70  13.99  15.10  15.94  15.80  63.48  39.36  50.30  40.31  43.41  49.27  39.00
> outliers <- boxplot(data, plot = FALSE)$out
> data_no_outlier <- data[-which(data %in% outliers)]
> boxplot(data_no_outlier, plot = FALSE)$out
 [1] 20.14 33.64 36.05 36.38 18.80 33.76 19.95 18.03 19.27 17.98 19.31 37.16 36.49 35.06 36.18 36.12 37.24 35.56 19.46
> length(data_no_outlier)
[1] 107
```

Description: This code identifies and removes outliers from a dataset containing "bmi" values. It shows that 13 outliers were initially detected, but after their removal, the cleaned dataset contains 107 values, indicating that the outliers have been successfully removed. The code exemplifies the process of handling outliers in a "bmi" dataset.

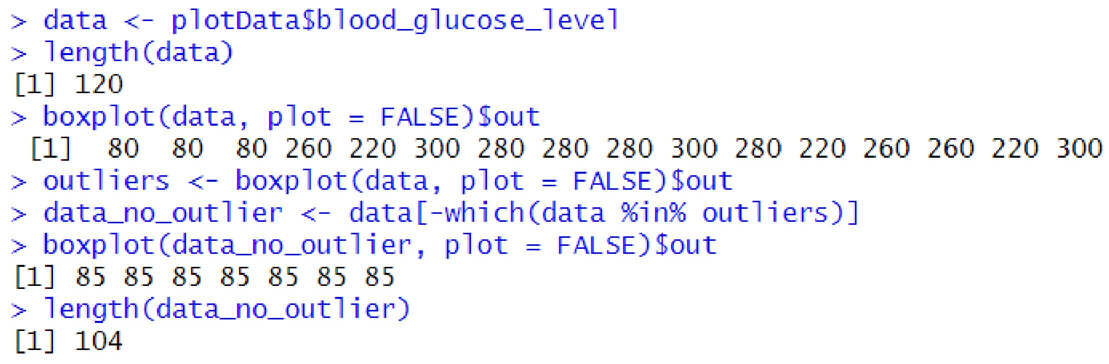
BLOOD GLUCOSE LEVEL:

Code:



```
data <- plotData$blood_glucose_level
length(data)
boxplot(data, plot = FALSE)$out
outliers <- boxplot(data, plot = FALSE)$out
data_no_outlier <- data[-which(data %in% outliers)]
boxplot(data_no_outlier, plot = FALSE)$out
length(data_no_outlier)
```

Output:

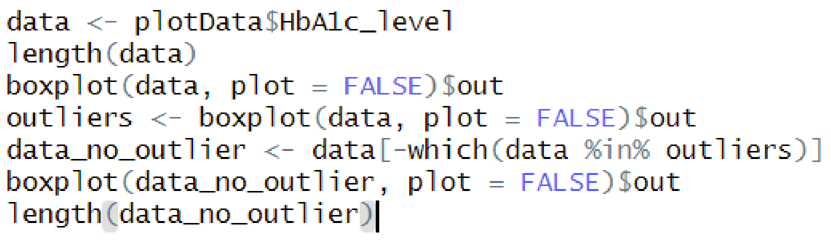


```
> data <- plotData$blood_glucose_level
> length(data)
[1] 120
> boxplot(data, plot = FALSE)$out
 [1]  80  80  80 260 220 300 280 280 280 300 280 220 260 260 220 300
> outliers <- boxplot(data, plot = FALSE)$out
> data_no_outlier <- data[-which(data %in% outliers)]
> boxplot(data_no_outlier, plot = FALSE)$out
[1] 85 85 85 85 85 85 85
> length(data_no_outlier)
[1] 104
```

Description: This code identifies and removes outliers from a dataset containing "blood_glucose_level" values. Initially, it detected 16 outliers, but after their removal, the cleaned dataset contains 104 values. The code showcases the process of handling outliers in the "blood_glucose_level" dataset

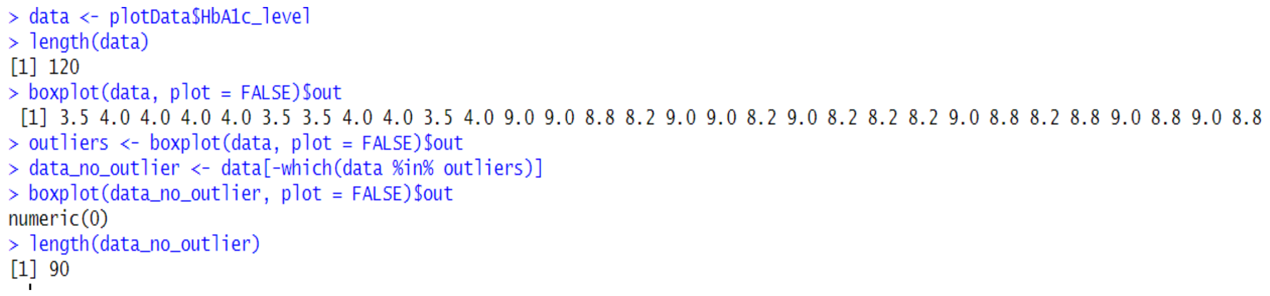
HbA1c_level

Code:



```
data <- plotData$HbA1c_level
length(data)
boxplot(data, plot = FALSE)$out
outliers <- boxplot(data, plot = FALSE)$out
data_no_outlier <- data[-which(data %in% outliers)]
boxplot(data_no_outlier, plot = FALSE)$out
length(data_no_outlier)
```

Output:

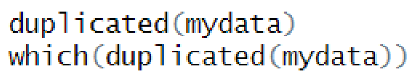


```
> data <- plotData$HbA1c_level
> length(data)
[1] 120
> boxplot(data, plot = FALSE)$out
 [1] 3.5 4.0 4.0 4.0 4.0 3.5 3.5 4.0 4.0 3.5 4.0 9.0 9.0 8.8 8.2 9.0 9.0 8.2 9.0 8.2 8.2 8.2 9.0 8.8 8.2 8.8 9.0 8.8 9.0 8.8
> outliers <- boxplot(data, plot = FALSE)$out
> data_no_outlier <- data[-which(data %in% outliers)]
> boxplot(data_no_outlier, plot = FALSE)$out
numeric(0)
> length(data_no_outlier)
[1] 90
```

Description: The original dataset contained 120 data points. The boxplot identified 5 outliers values. The identified outliers are [3.5,4.0,9.0,8.8,8.2]. The boxplot of the modified shows no outliers(`numeric(0)`) indicates that all the outliers were removed successfully. After removing all the outliers, the attribute contains 90 data points as there were duplicate data.

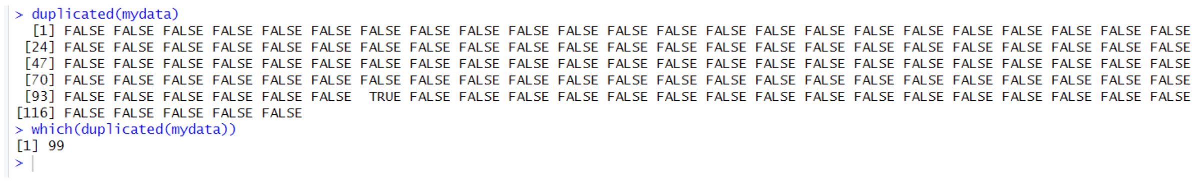
Handling Duplicate Values

Code:



```
duplicated(mydata)
which(duplicated(mydata))
```

Output:

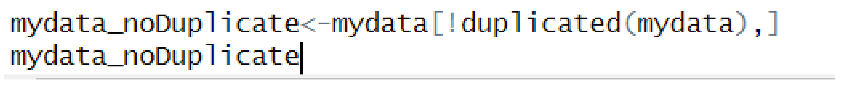


```
> duplicated(mydata)
  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [24] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [47] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [70] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [93] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[116] FALSE FALSE FALSE FALSE FALSE
> which(duplicated(mydata))
[1] 99
> 
```

Description:

The duplicated() function is used to identify duplicate row in a dataset. The function returns a logical vector indicating whether each element is a duplicate of a previous element. The TRUE indicates that the corresponding row is duplicate. Which() function is used to find the corresponding row number. In our dataset we have a row which has corresponding duplicate row. Row no 99 has a corresponding duplicate row.

Code:



```
mydata_noDuplicate<-mydata[!duplicated(mydata),]
mydata_noDuplicate
```

Output:

```
98      0  43        0        0        1  27.32     6.6          130      1
100     1  80        0        1        3  24.36     7.5          280      1
```

Description: Duplicated(mydata) identifies the duplicate data in dataset. The "!" simply removes the duplicate row. In output we can the duplicated row no 99 has been removed.