AMERICAN INTERNATIONAL UNIVERSITY, BANGLADESH
FACULTY OF SCIENCE AND TECHNOLOGY
MIDTERM PROJECT(B) FALL 23-24

PROJECT TITLE: MEDICAL HISTORY CLASSIFICATION DATASET

Submitted To:

Name	ID
Mohammad Bin Harun	21-44583-1
MD.Harun OR Rashid	21-44586-1

Submitted By:

Tohedul Islam

Project Description:

The provided dataset captures various health-related attributes, including gender, age, hypertension, heart disease, smoking history, body mass index (BMI), HbA1c level, blood glucose level, and diabetes status. It is a modified version of a diabetes prediction dataset. The dataset exhibits diversity in terms of age groups, gender, and health conditions. However, it contains missing values, outliers, and some invalid entries, requiring comprehensive data preparation and exploration. The goal of this project is to conduct univariate data exploration, handle missing values and outliers, and prepare the data for a diabetes prediction analysis in the R programming language.

Data Preparation steps:

Store the dataset into a list:

Code & Output:

```
> dataset<-read_excel("E:/AIUB/9th semester/Data Science/Mid_Project/Dataset_MIdterm_sectoin(B).xlsx")
> dataset
# A tibble: 120 x 9
   gender   age hypertension heart_disease smoking_history   bmi HbA1c_level blood_glucose_level diabetes
   <chr>  <dbl>        <dbl>         <dbl> <chr>           <dbl>       <dbl>               <dbl>    <dbl>
 1 Female    80            0             1 never            25.2         6.6                 140        0
 2 Female    54            0             0 No Info          27.3         6.6                  80        0
 3 Male      28            0             0 never           -27.3         5.7                 158        0
 4 Female    NA            0             0 current          23.4         5                   155        0
 5 Male      76            1             1 current          20.1         4.8                 155        0
 6 Female    20            0             0 never            27.3         6.6                  85        0
 7 NA        79            0             0 No Info          23.9         5.7                  85        0
 8 Male      42            0             0 never            33.6         4.8                 145        0
 9 Female    32            0             0 never            27.3         5                   100        0
10 Female    53            0             0 never            27.3         6.1                  85        0
# i 110 more rows
# i Use `print(n = ...)` to see more rows
>
```

Description:

The code utilizes the 'read_excel' function from the 'readxl' package in R to import data from an Excel file located at "E:/AIUB/9thsemester/DataScience/Mid_Project/Dataset_MIdterm_sectoin(B).xlsx". The imported dataset is stored in the 'dataset' variable.

Handle Missing value:

Code & Output:

```
> dataset$smoking_history[dataset$smoking_history=="No Info"] <- NA
> most_frequency_value<- table(dataset$smoking_history)
> most_frequency_value

    current        ever      former       never not current
         12           6          13          52           5
> sort_most_frequence_value<-sort(most_frequency_value,decreasing = TRUE)
> sort_most_frequence_value

      never      former     current        ever not current
         52          13          12           6           5
> mode_sort_most_frequency_value<-names(sort_most_frequence_value)[1]
> mode_sort_most_frequency_value
[1] "never"
> dataset$smoking_history[is.na(dataset$smoking_history)]<-mode_sort_most_frequency_value
>
```

Description:

The code segment aims to replace occurrences of "No Info" in the 'smoking_history' column of the dataset with NA. The table() function are used to calculate the total number of unique value in specified column. After counting all unique values sorting the column in descending order to decreasing using sort() function. The names() function assigns the name of the most frequent value in the smoking_history column. The NA values in 'smoking_history' are replaced with the identified mode.

Code & Output:

```
> most_frequency_value_gender<- table(dataset$gender)
> most_frequency_value_gender

Female   Male
    71     47
> sort_most_frequence_value_gender<-sort(most_frequency_value_gender,decreasing = TRUE)
> sort_most_frequence_value_gender

Female   Male
    71     47
> mode_sort_most_frequency_value_gender<-names(sort_most_frequence_value_gender)[1]
> mode_sort_most_frequency_value_gender
[1] "Female"
> dataset$gender[is.na(dataset$gender)]<-mode_sort_most_frequency_value_gender
>
```

Description:

The NA values in 'gender' are replaced with the identified mode.

Code & Output:

```
> most_frequency_value_hypertension<- table(dataset$hypertension)
> most_frequency_value_hypertension

  0   1
110  10
> sort_most_frequence_value_hypertension<-sort(most_frequency_value_hypertension,decreasing = TRUE)
> sort_most_frequence_value_hypertension

  0   1
110  10
> mode_sort_most_frequency_value_hypertension<-as.numeric(names(sort_most_frequence_value_hypertension)[1])
> mode_sort_most_frequency_value_hypertension
[1] 0
> dataset$hypertension[is.na(dataset$hypertension)]<-mode_sort_most_frequency_value_hypertension
>
```

Description:

The NA values in hypertension are replaced with the identified mode.

Code & Output:

```
> colSums(is.na(dataset))
              gender                  age         hypertension        heart_disease      smoking_history
                   0                    0                    0                    0                    0
                 bmi          HbA1c_level  blood_glucose_level             diabetes
                   0                    0                    0                    0
> dataset$age[is.na(dataset$age)]<-mean(dataset$age,na.rm = TRUE)
> colSums(is.na(dataset))
              gender                  age         hypertension        heart_disease      smoking_history
                   0                    0                    0                    0                    0
                 bmi          HbA1c_level  blood_glucose_level             diabetes
                   0                    0                    0                    0
>
```

Description:

The first line calculates the sum of missing values in each column of the dataset using. The second line replaces the missing values in the 'age' column with the mean of the 'age' column.

Handle the negative value:

Code & Output:

```
> dataset$bmi<- abs(dataset$bmi)
> dataset$bmi
  [1] 25.19 27.32 27.32 23.45 20.14 27.32 23.86 33.64 27.32 27.32 54.70 36.05 25.69 27.32 27.32 30.36 24.48 27.32
 [19] 25.72 36.38 18.80 21.24 27.94 13.99 33.76 27.85 26.47 26.08 31.75 25.15 22.01 22.19 23.55 15.10 21.76 21.22
 [37] 27.32 32.02 29.30 27.32 24.93 19.95 18.03 28.27 19.27 27.32 27.32 28.12 26.10 27.32 27.32 30.22 23.11 27.32
 [55] 28.16 26.78 23.04 15.94 15.80 27.01 27.32 22.19 27.45 17.98 26.45 31.16 24.42 30.50 19.31 27.32 27.32 25.91
 [73] 27.32 37.16 63.48 27.32 32.27 27.32 27.32 27.32 31.70 22.06 36.49 30.80 39.36 31.90 26.71 27.32 27.77 27.32
 [91] 35.06 23.25 29.25 24.81 36.18 50.30 27.09 27.32 27.09 24.36 29.20 25.41 40.31 27.32 26.53 36.12 27.32 37.24
[109] 35.56 43.41 27.32 49.27 39.00 22.43 32.19 25.94 27.73 19.46 27.32 27.32
>
```

Description:

In the dataset has to the BMI negative value. Which actually does not exist for the BMI. So, using the abs() function absolute the 'bmi' column.

Handle Noisy value:

Now since outliers are an issue terms of getting accurate results from this dataset, we remove them and replace them with the mean values of those attribute columns.

Age:

From the age column we have identified the two outliers for line 52(age-290) and line 119(age-280) they have unusually high age.

Code:

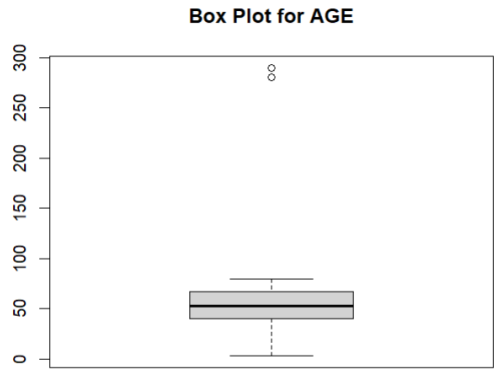
```
> boxplot(dataset$age, main='Box Plot for AGE')
>
```

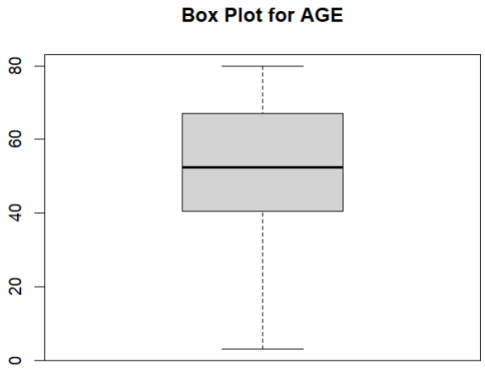
```
> q1<-quantile(dataset$age,0.25)
> q1
  25%
40.75
> q3<-quantile(dataset$age,0.75)
> q3
  75%
67.25
> iqr<-q3-q1
> iqr
 75%
26.5
> outliers_age<-dataset$age<(q1-1.5*iqr)| dataset$age>(q3+1.5*iqr)
> dataset$age<-ifelse(outliers_age,NA,dataset$age)
> dataset$age[is.na(dataset$age)]<-mean(dataset$age,na.rm = TRUE)
>
> dataset$age[52]
[1] 50.26008
> dataset$age[119]
[1] 50.26008
>
```

Univariate data Exploration:

Box plot: To Finding the outliers first using the box-plot graphical representation. After that remove the outliers.

Output:



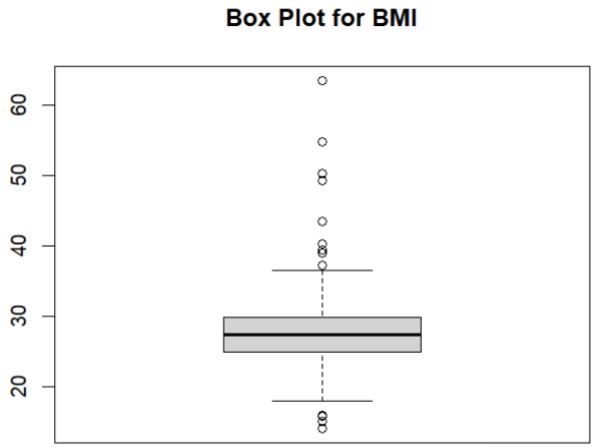


BMI:

Code:

```
> boxplot(dataset$bmi, main='Box Plot for BMI')
>
```

Output:



Description:

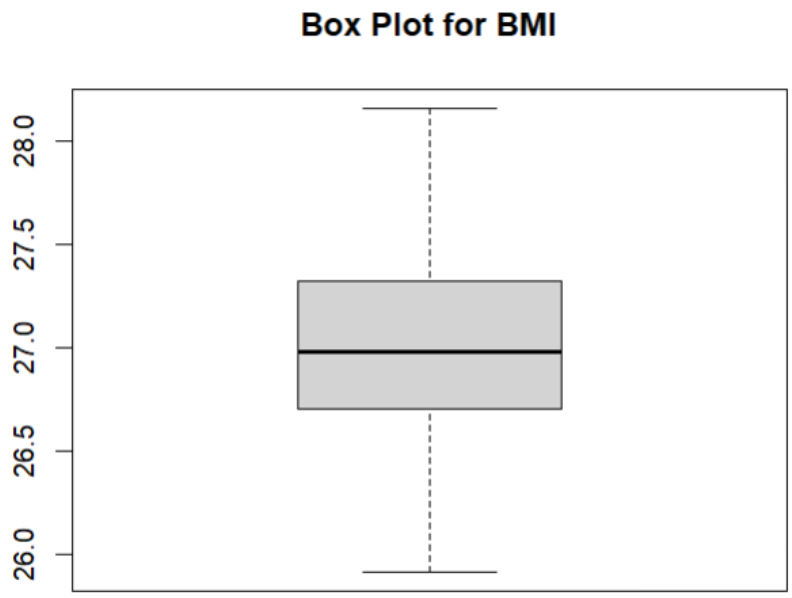
Generates a box plot for the 'bmi' column in the dataset, providing a visual representation of its distribution and identifying potential outliers.

Handle BMI column outlier:

Code:

```
> q1<-quantile(dataset$bmi,0.25)
> q1
     25%
26.70249
> q3<-quantile(dataset$bmi,0.75)
> q3
  75%
27.32
> iqr<-q3-q1
> iqr
      75%
0.6175126
> outliers_bmi<-dataset$bmi<(q1-1.5*iqr)| dataset$bmi>(q3+1.5*iqr)
> dataset$bmi<-ifelse(outliers_bmi,NA,dataset$bmi)
> dataset$bmi[is.na(dataset$bmi)]<-mean(dataset$bmi,na.rm = TRUE)
> boxplot(dataset$bmi, main='Box Plot for BMI')
>
```

Output:



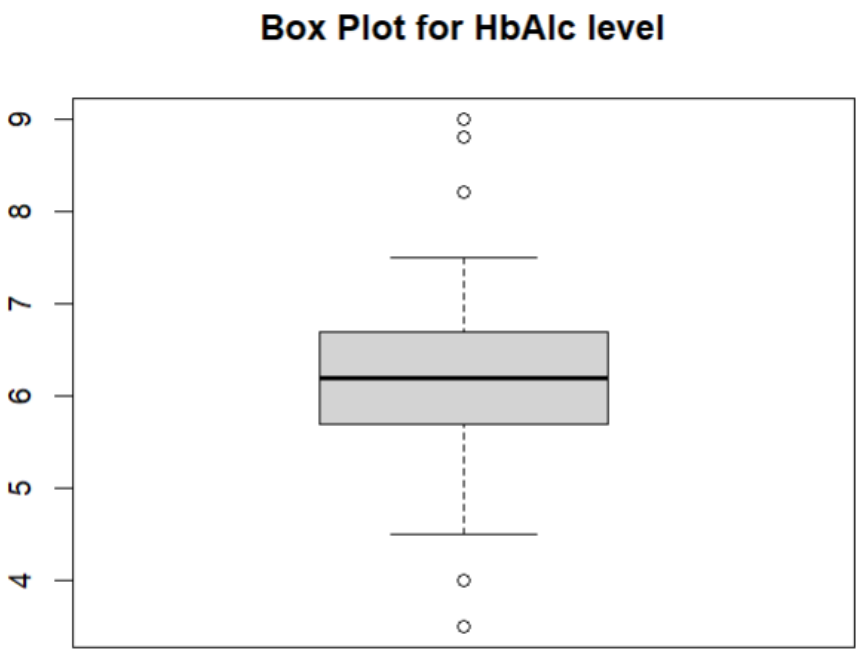
Description:
To handle the outliers, computes the first quartile (q1), third quartile (q3), and interquartile range (iqr) for 'bmi', identifies outliers based on a iqr criterion, and replaces those outliers with NA, then replace the NA values with the mean of the 'bmi' column.

#HbA1c_level outlier:

Code:

```
> boxplot(dataset$HbA1c_level, main='Box Plot for HbAlc level')
>
```

Output:



Description:

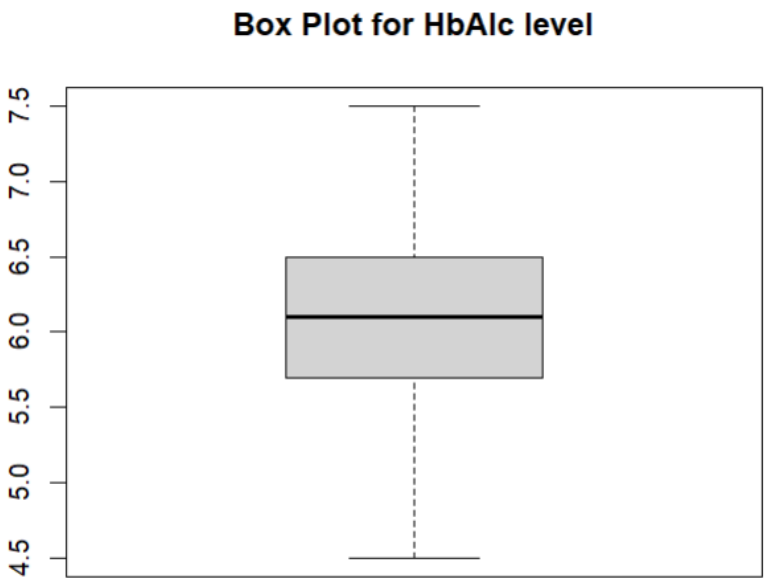
Generates a box plot for the 'HbAlc_level' column in the dataset, providing a visual representation of its distribution and identifying potential outliers.

Handle HbAlc level outlier:

Code:

```
> q1<-quantile(dataset$HbA1c_level,0.25)
> q1
25%
5.7
> q3<-quantile(dataset$HbA1c_level,0.75)
> q3
 75%
6.65
> iqr<-q3-q1
> iqr
 75%
0.95
> outliers_HbA1c_level<-dataset$HbA1c_level<(q1-1.5*iqr)| dataset$HbA1c_level>(q3+1.5*iqr)
> dataset$HbA1c_level<-ifelse(outliers_HbA1c_level,NA,dataset$HbA1c_level)
> dataset$bmi[is.na(dataset$HbA1c_level)]<-mean(dataset$HbA1c_level,na.rm = TRUE)
> |
```

Output:



Description:

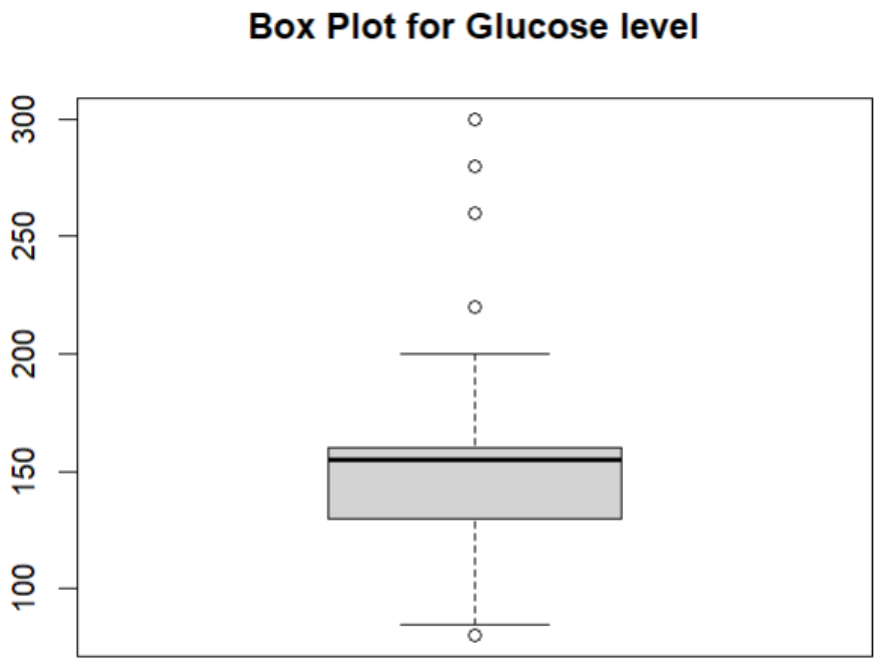
To handle the outliers, computes the first quartile (q1), third quartile (q3), and interquartile range (iqr) for 'HbAlc_level', identifies outliers based on a iqr criterion, and replaces those outliers with NA, then replace the NA values with the mean of the 'HbAlc_level' column.

Glucose level:

Code:

```
> boxplot(dataset$blood_glucose_level, main='Box Plot for Glucose level')
> 
```

Output:



Description:

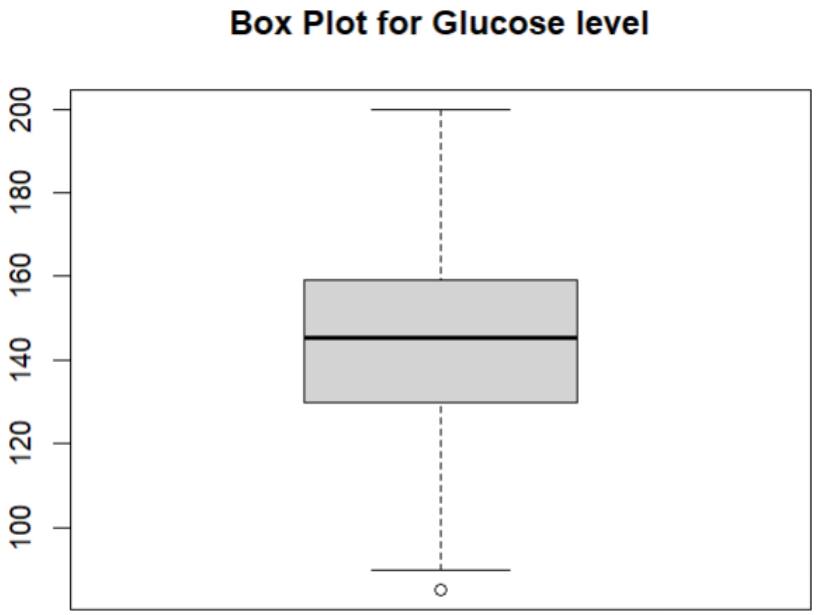
Generates a box plot for the 'blood_glucose_level' column in the dataset, providing a visual representation of its distribution and identifying potential outliers.

Handle Glucose level outlier:

Code:

```
> q1<-quantile(dataset$blood_glucose_level,0.25)
> q1
25%
130
> q3<-quantile(dataset$blood_glucose_level,0.75)
> q3
75%
160
> iqr<-q3-q1
> iqr
75%
 30
> outliers_blood_glucose_level<-dataset$blood_glucose_level<(q1-1.5*iqr)| dataset$blood_glucose_level>(q3+1.5*iqr)
> dataset$blood_glucose_level<-ifelse(outliers,NA,dataset$blood_glucose_level)
> dataset$blood_glucose_level[is.na(dataset$blood_glucose_level)]<-mean(dataset$blood_glucose_level,na.rm = TRUE)
> boxplot(dataset$blood_glucose_level, main='Box Plot for Glucose level')
>
```

Output:



Description:

To handle the outliers, computes the first quartile (q1), third quartile (q3), and interquartile range (iqr) for 'blood_glucose_level', identifies outliers based on a iqr criterion, and replaces those outliers with NA, then replace the NA values with the mean of the 'blood_glucose_level' column.

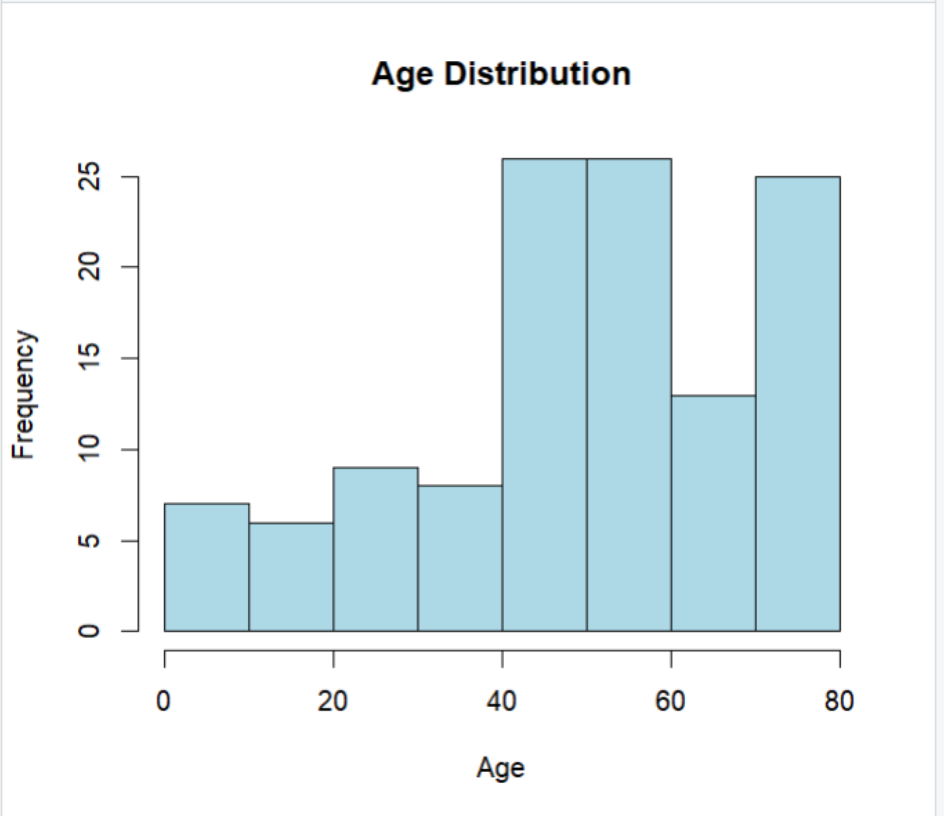
Histogram plot:

The frequencies of values of a variable bucketed into ranges are represented by a histogram. R uses the hist() function to produce a histogram. This function plots histograms using additional parameters after receiving a vector as input.

Code:

```
> hist(dataset$age,main = "Age Distribution",xlab = 'Age',col = 'lightblue')
>
```

Output:



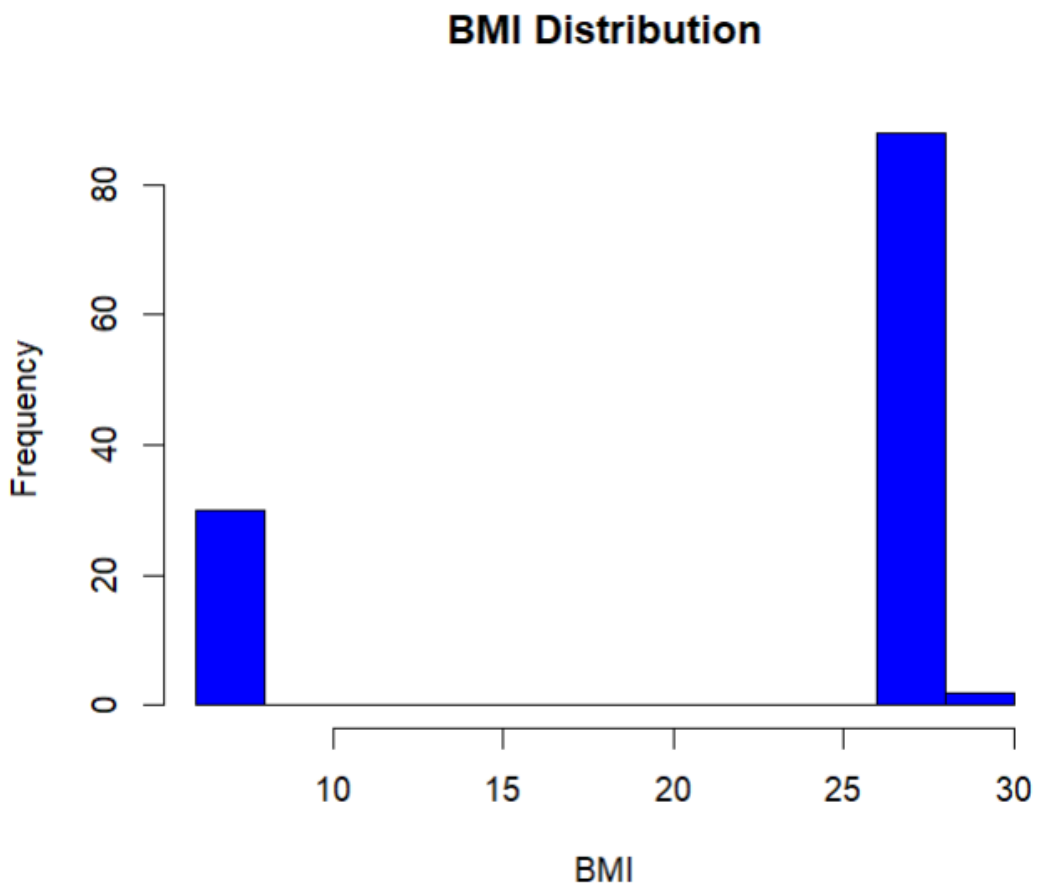
Description:
Above the graph lightblue represents the age of histogram.

BMi:

Code:

```
> hist(dataset$bmi,main = "BMI Distribution",xlab = 'BMI',col = 'blue')
>
```

Output:



Description:

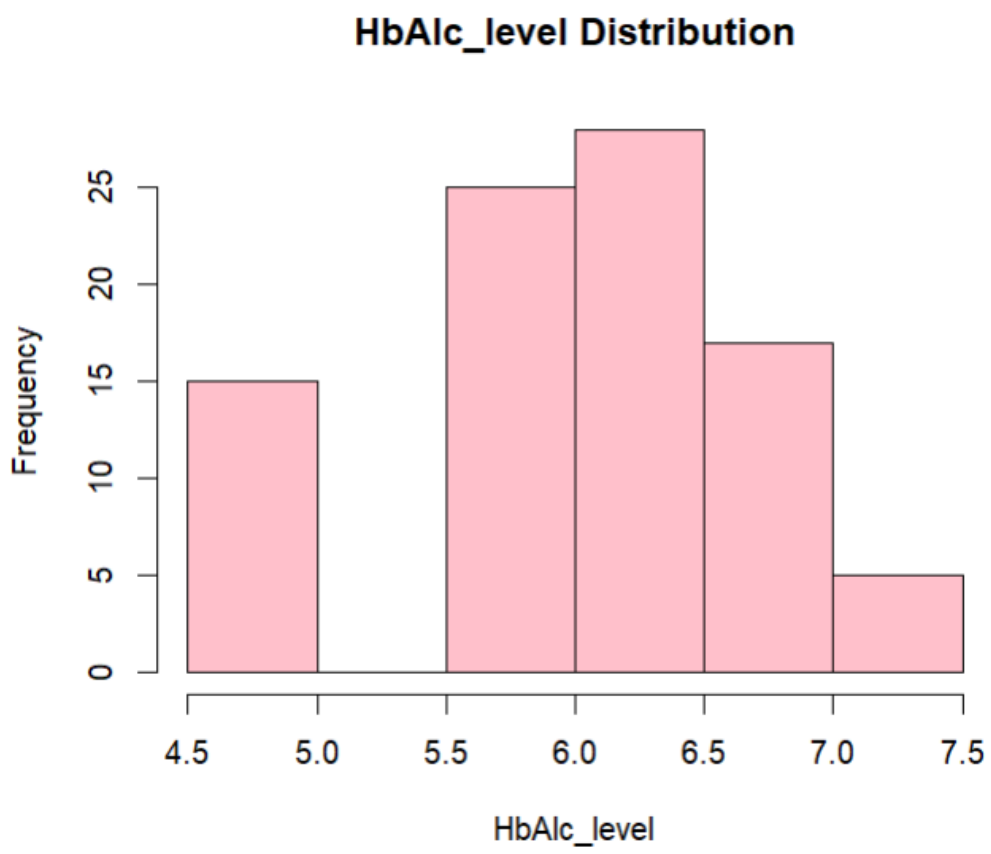
Above the graph blue represents the bmi of histogram.

HbAlc level:

Code:

```
> hist(dataset$HbA1c_level,main = "HbAlc_level Distribution",xlab = 'HbAlc_level',col = 'pink')
>
```

Output:



Description:

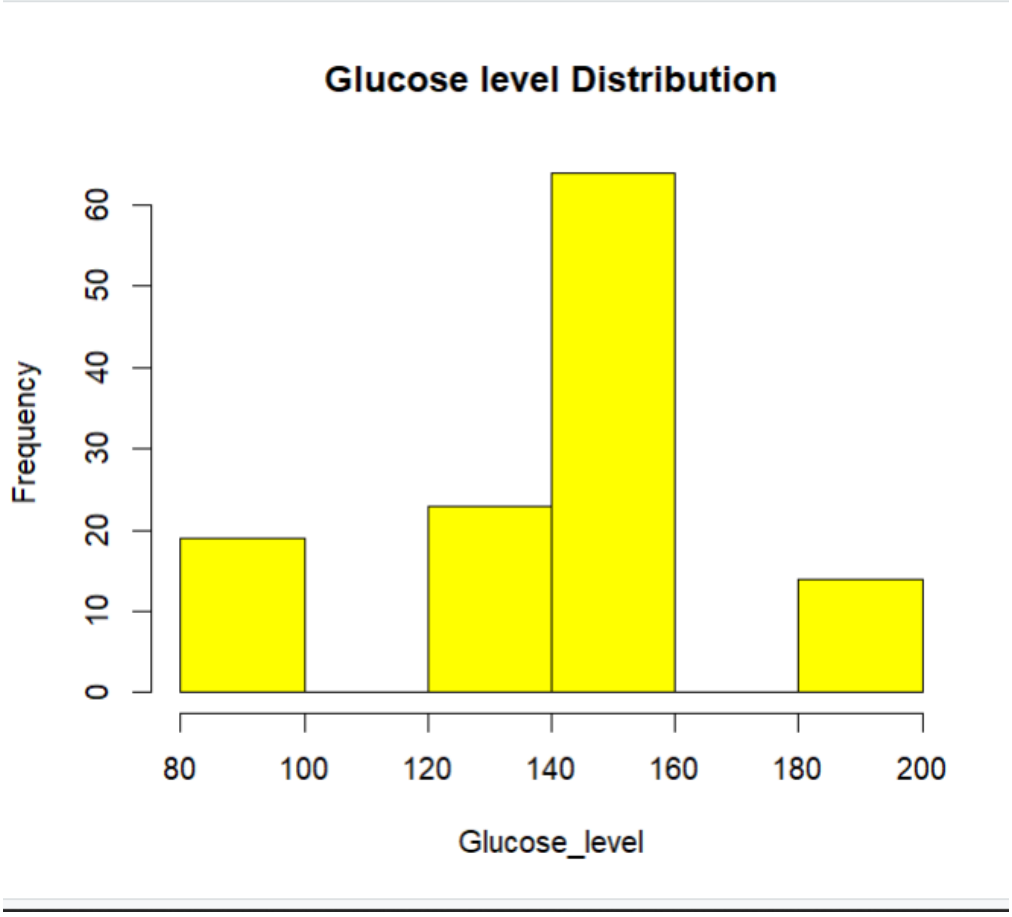
Above the graph pink color represents the HbAlc_level of histogram.

Glucose level:

Code:

```
> hist(dataset$blood_glucose_level,main = "Glucose level Distribution",xlab = 'Glucose_level',col = 'yellow')
>
```

Output:



Description:

Above the graph yellow color represents the blood_glucose_level of histogram.

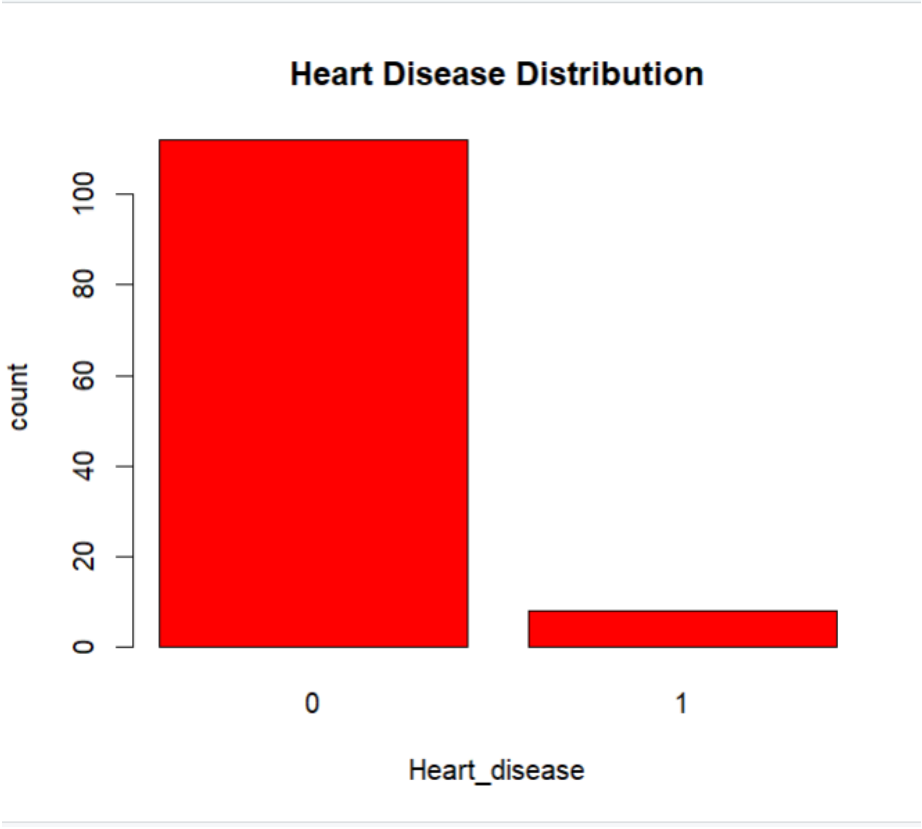
Bar plot:

Bar plots in R are graphical representations that display the distribution of categorical data. R uses, bar plots are created for variables such as gender, hypertension, heart disease, smoking history, and diabetes status, offering a quick and informative overview of the distribution of these categorical variables in the dataset.

Code:

```
> barplot(table(dataset$heart_disease),main = "Heart Disease Distribution",xlab = "Heart_disease",ylab = 'count',col =
"red")
>
```

Output:



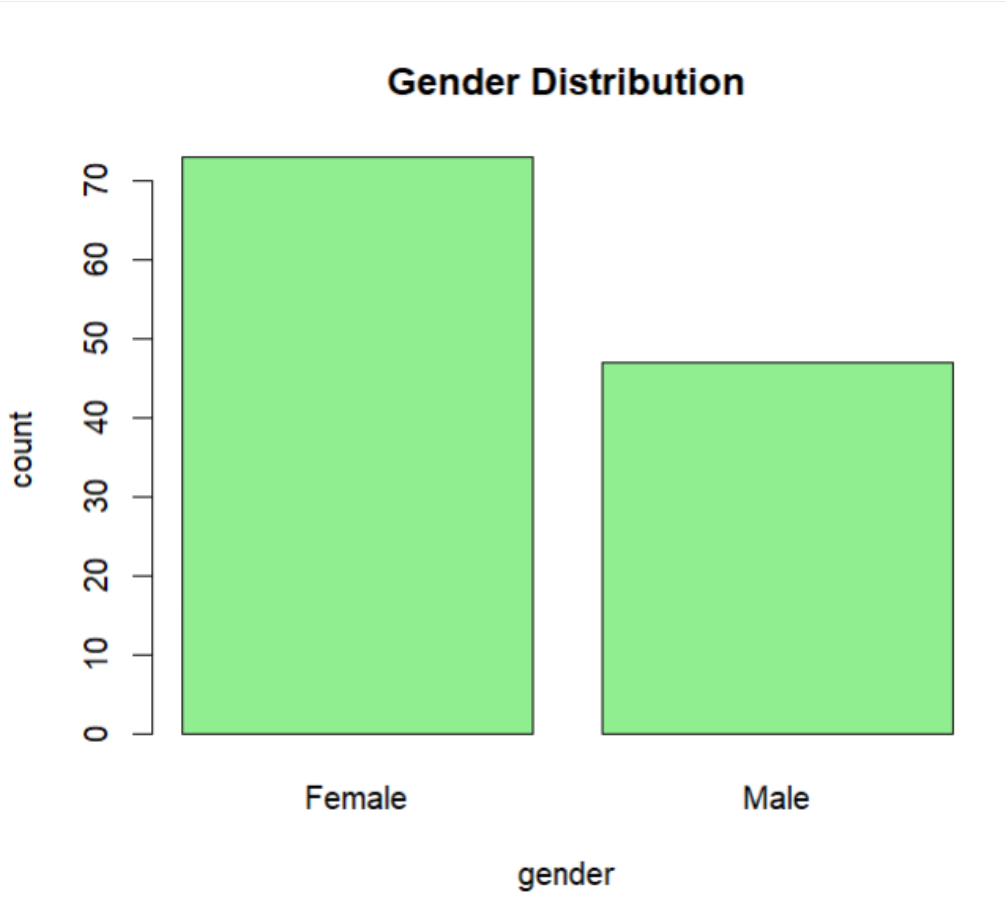
Description:

Show the heart disease overview in bar plot representing.

Code:

```
> barplot(table(dataset$gender),main = "Gender Distribution",xlab = 'gender',ylab = 'count',col = 'lightgreen')
> 
```

Output:



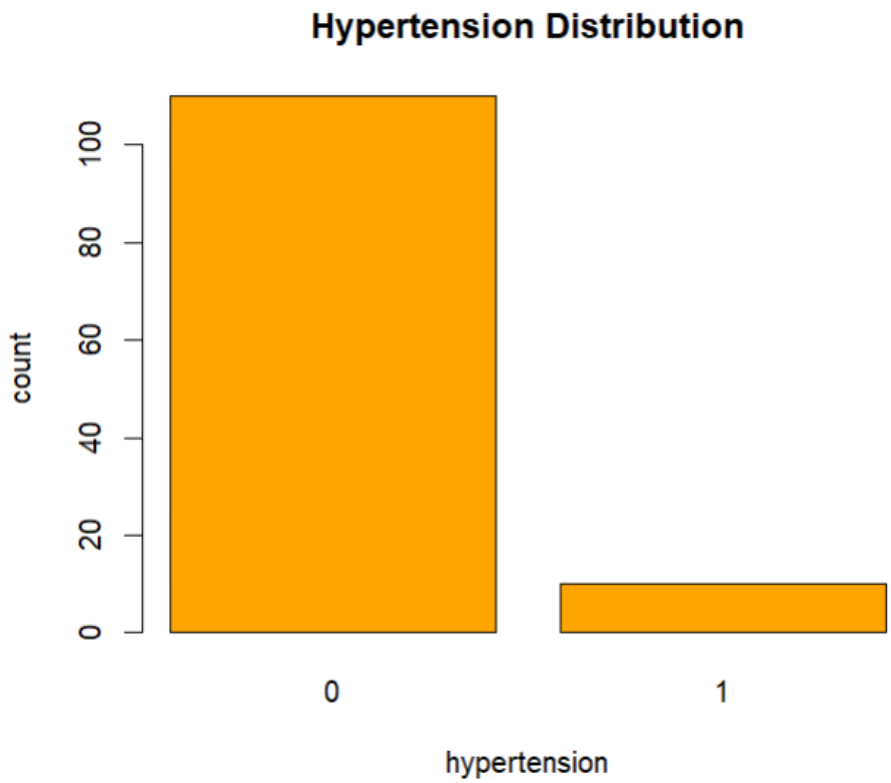
Description:

Show the gender overview in bar plot representing.

Code:

```
> barplot(table(dataset$hypertension),main = "Hypertension Distribution",xlab = 'hypertension',ylab = 'count',col = 'or
ange')
>
```

Output:



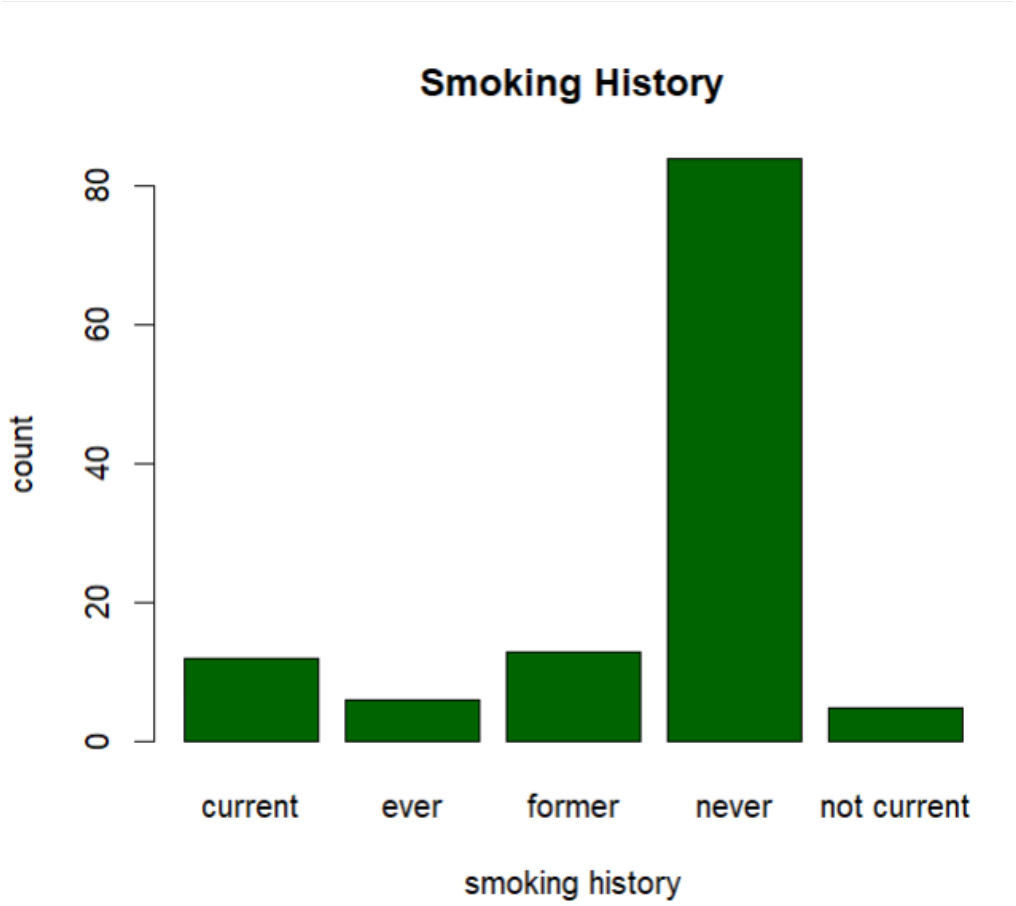
Description:

Show the hypertension overview in bar plot representing.

Code:

```
> barplot(table(dataset$smoking_history),main = "Smoking History",xlab = 'smoking history',ylab = 'count',col = 'darkgr
een')
>
```

Output:



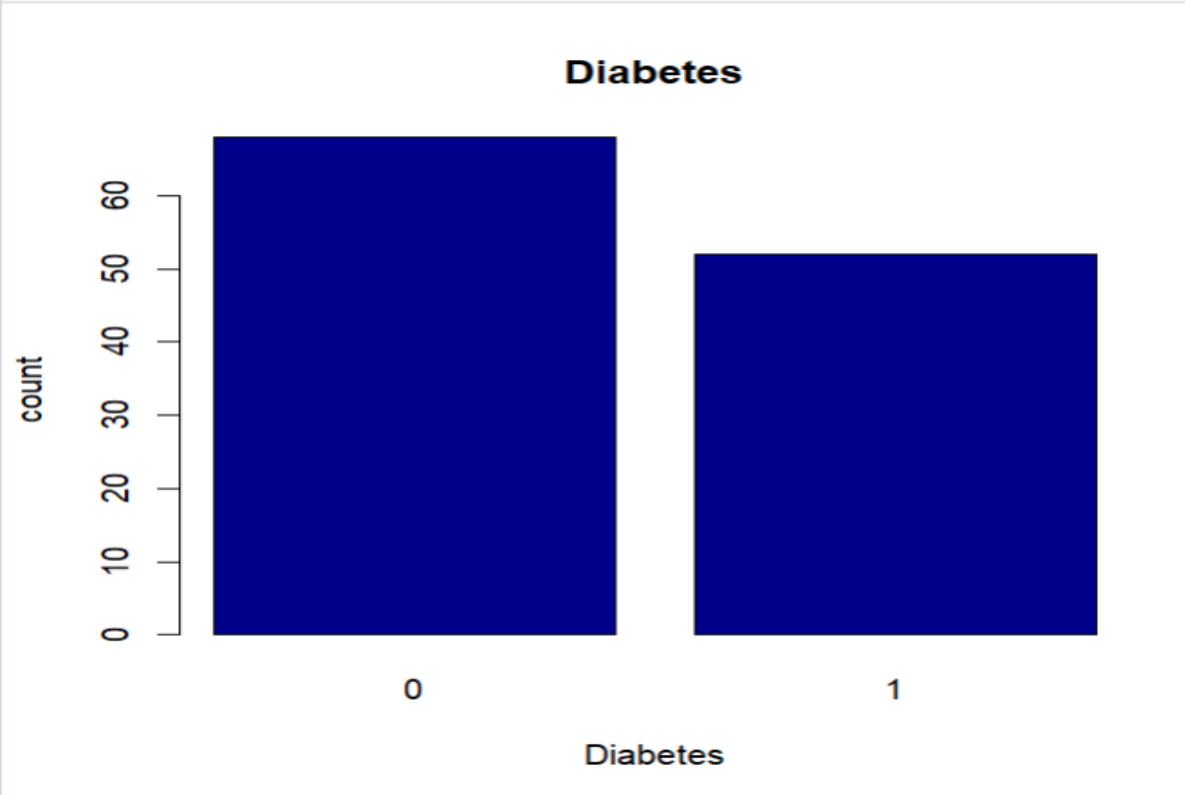
Description:

Show the smoking history overview in bar plot representing.

Code:

```
> barplot(table(dataset$diabetes),main = 'Diabetes',xlab = 'Diabetes',ylab = 'count',col = 'darkblue')
>
```

Output:



Description:

Show the diabetes overview in bar plot representing.