AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Science and Technology

Project Cover Page

Assignment Title:	Midterm Project of Introduction to Data Science.		
Assignment No:	01	Date of Submission:	11 November 2023
Course Title:	Introduction to Data Science.		
Course Code:	CSC4180	Section:	B
Semester:	Fall 2023-24	Course Teacher:	Tohedul Islam

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaborationhas been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand thatPlagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

* *Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.: 01

No	Name	ID	Program	Signature
1	Satyajit Das	21-44374-1	BSc [CSE]	
2	Tazrif Yamshit Raim	21-45012-1	BSc [CSE]	
3			Choose an item.	
4			Choose an item.	
5			Choose an item.	
6			Choose an item.	
7			Choose an item.	
8			Choose an item.	
9			Choose an item.	
10			Choose an item.	

Faculty use only

FACULTYCOMMENTS	Marks Obtained	
	Total Marks	

Description about the dataset:

The Diabetes Prediction Dataset is a vital resource containing medical and demographic data from patients, along with their diabetes status. It includes key parameters such as age, gender, BMI, hypertension, heart disease, smoking history, HbA1c, and blood glucose levels. This dataset is crucial for developing machine learning models to predict the risk of diabetes based on a patient's health history and demographics, aiding healthcare professionals in early detection and personalized care planning. It also serves as a significant tool for researchers studying the interplay between various health and demographic factors and the onset of diabetes.

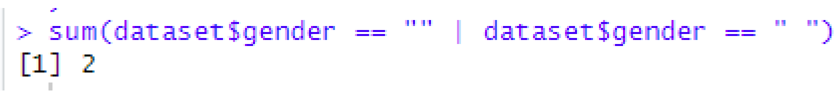
The dataset contains the following attributes:

- **gender**: Gender of the individual.
- **age**: Age of the individual.
- **hypertension**: Indicator of hypertension. Type of this attribute is categorical. But it is represented through numerical value where 0 means patient has no hypertension and 1 means patient has hypertension.
- **heart_disease**: Indicator of heart disease. Type of this attribute is categorical. But it is represented through numerical value where 0 means patient has no heart disease and 1 means patient has heart disease.
- **smoking_history**: Smoking history of the individual.
- **bmi**: Body Mass Index.
- **HbA1c_level**: HbA1c is your average blood glucose (sugar) levels for the last two to three months.
- **blood_glucose_level**: Blood glucose level.
- **diabetes**: Indicator of diabetes. Type of this attribute is categorical. But it is represented through numerical value where 0 means patient has no diabetes and 1 means patient has diabetes.

Data Preparation:

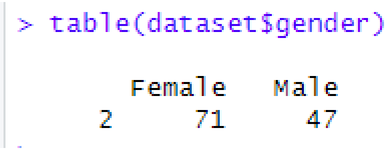
1.For gender attribute:

Male and female categorical values are contained in this attribute. It is evident that some values are missing. Let’s count the number of missing values first.

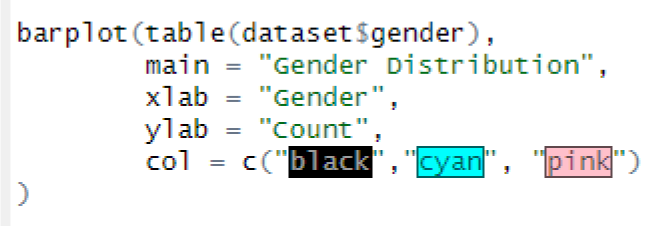


```
> sum(dataset$gender == "" | dataset$gender == " ")
[1] 2
```

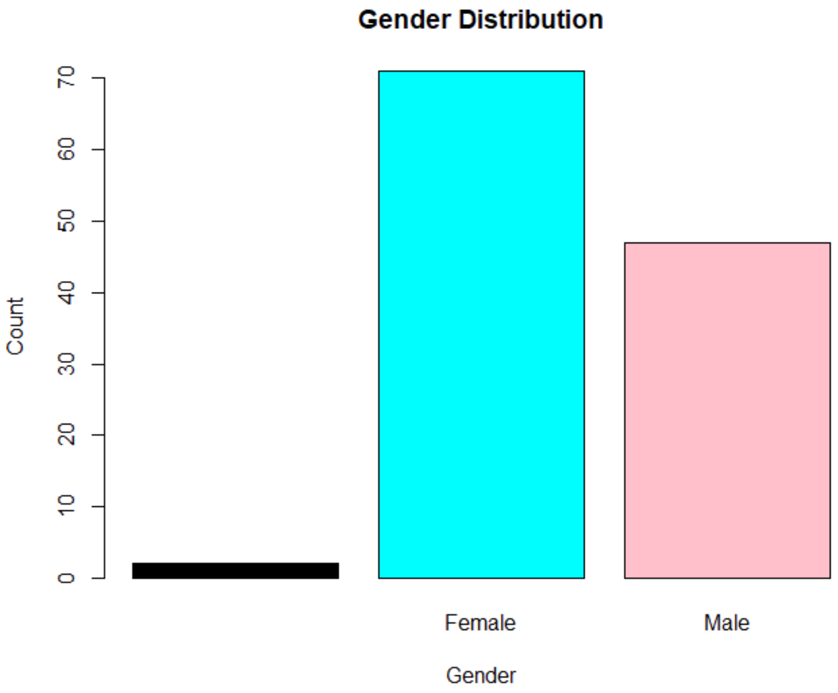
In other words, there are 118 instances in this attribute when the male is 47 and the female is 71. Given that it is a categorical attribute, let's create a bar plot using univariate explanation.



```
> table(dataset$gender)

        Female   Male
     2      71     47
```

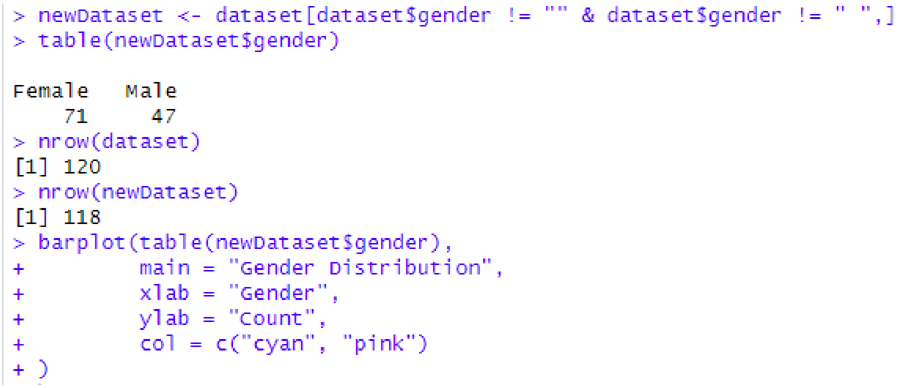


```
barplot(table(dataset$gender),
        main = "Gender Distribution",
        xlab = "Gender",
        ylab = "Count",
        col = c("black","cyan", "pink")
)
```

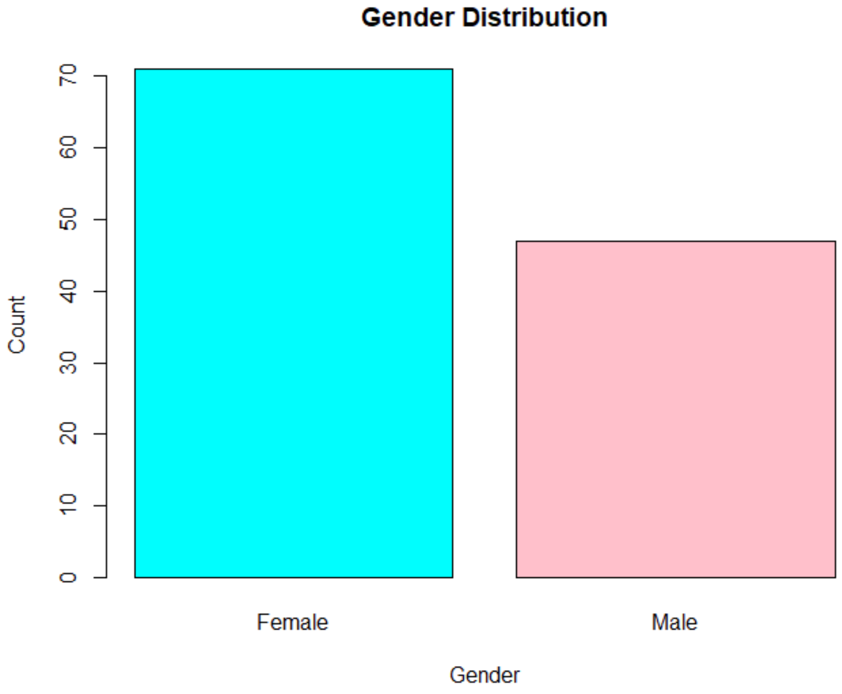


Three methods exist for us to eliminate this missing instance.

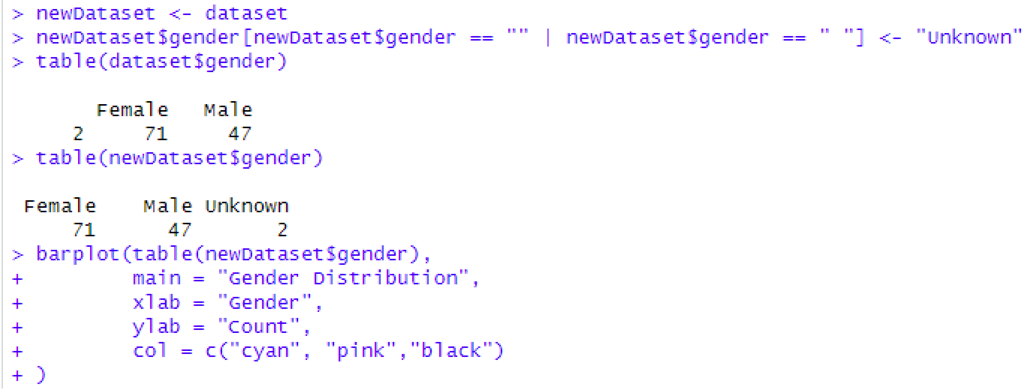
1.The missing values in the dataset can be eliminated.



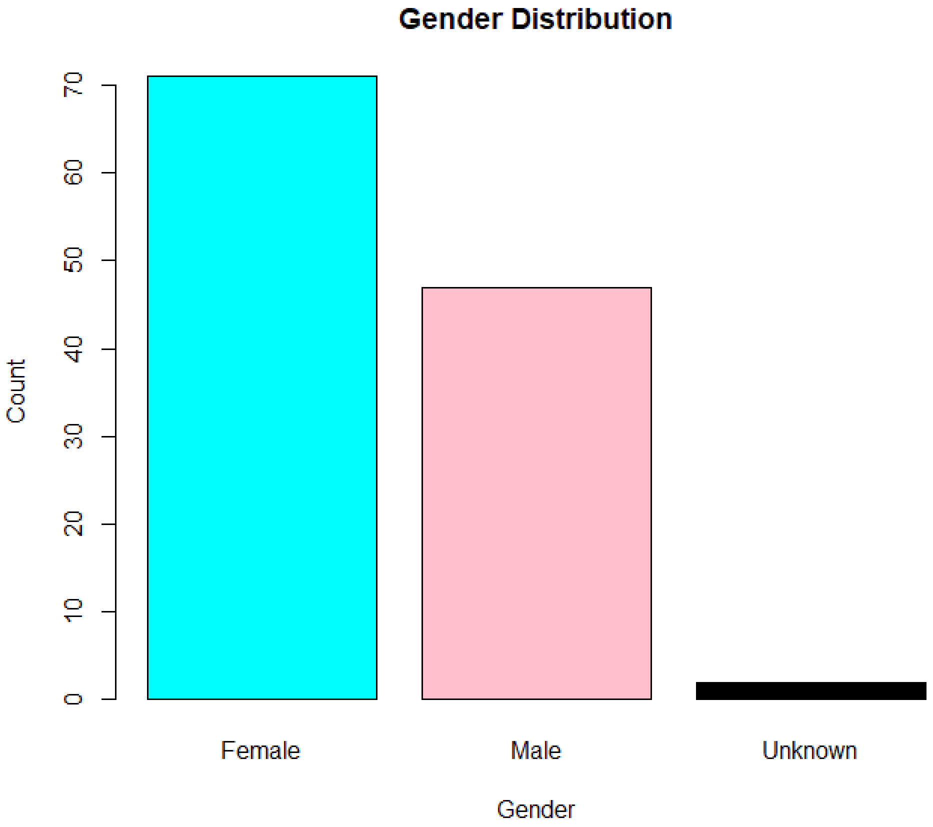
```
> newDataset <- dataset[dataset$gender != "" & dataset$gender != " ",]
> table(newDataset$gender)

Female   Male 
    71     47 
> nrow(dataset)
[1] 120
> nrow(newDataset)
[1] 118
> barplot(table(newDataset$gender),
+         main = "Gender Distribution",
+         xlab = "Gender",
+         ylab = "Count",
+         col = c("cyan", "pink")
+ )
```



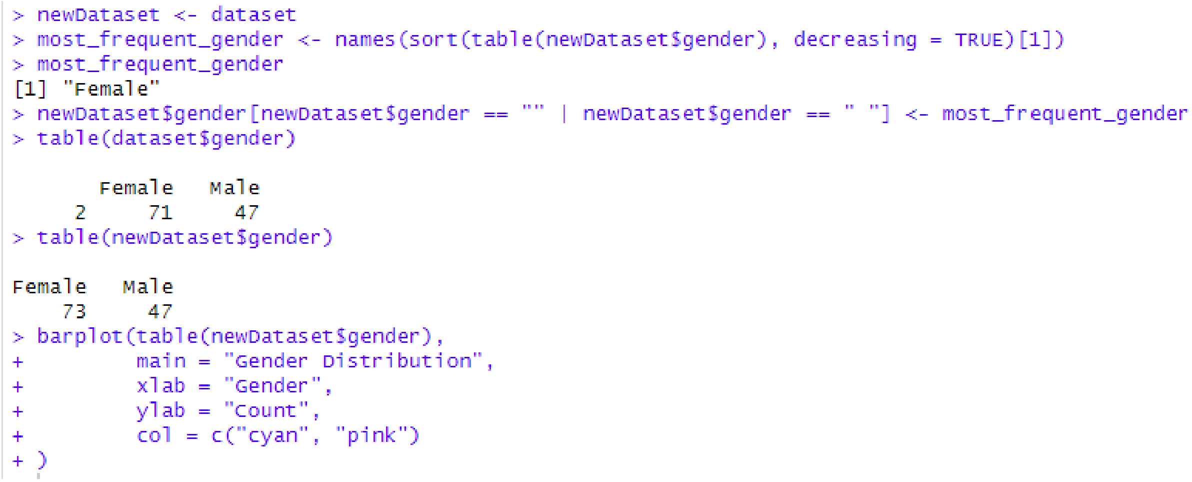
2. Use a place holder to fill in the missing value.



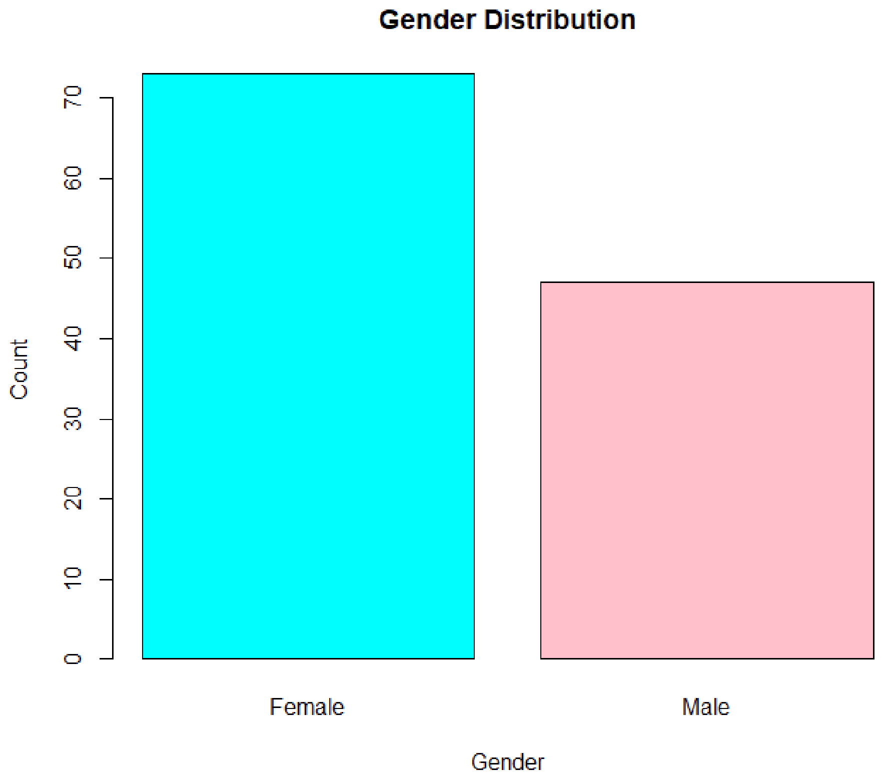
```
> newDataset <- dataset
> newDataset$gender[newDataset$gender == "" | newDataset$gender == " "] <- "Unknown"
> table(dataset$gender)

       Female   Male 
    2      71     47 
> table(newDataset$gender)

 Female    Male Unknown 
     71      47       2 
> barplot(table(newDataset$gender),
+         main = "Gender Distribution",
+         xlab = "Gender",
+         ylab = "Count",
+         col = c("cyan", "pink","black")
+ )
```



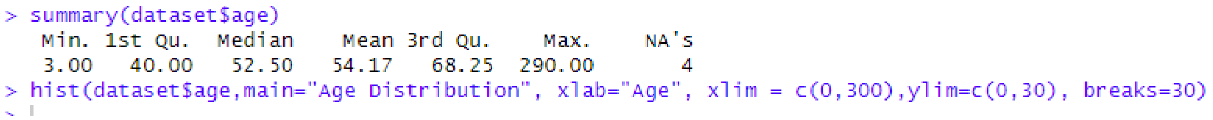
3. Replace with the value that occurs most frequently.



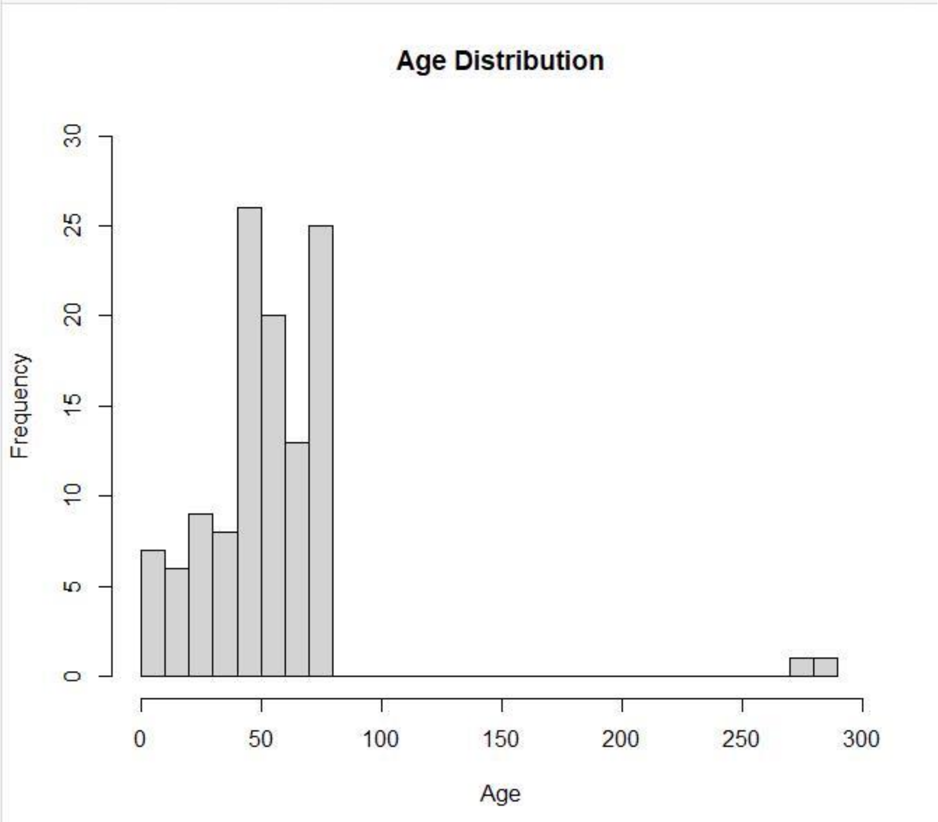
```
> newDataset <- dataset
> most_frequent_gender <- names(sort(table(newDataset$gender), decreasing = TRUE)[1])
> most_frequent_gender
[1] "Female"
> newDataset$gender[newDataset$gender == "" | newDataset$gender == " "] <- most_frequent_gender
> table(dataset$gender)

       Female   Male 
     2     71     47 
> table(newDataset$gender)

Female   Male 
    73     47 
> barplot(table(newDataset$gender),
+         main = "Gender Distribution",
+         xlab = "Gender",
+         ylab = "Count",
+         col = c("cyan", "pink")
+ )
```



2.For age Attribute:

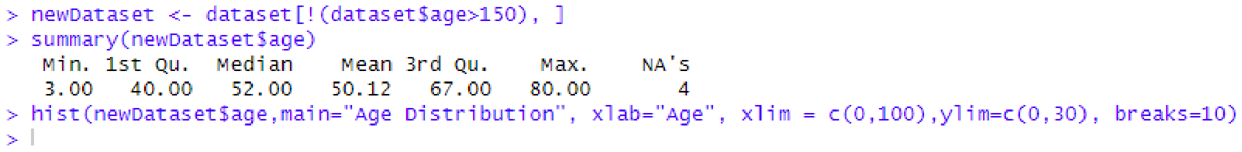


```
> summary(dataset$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
   3.00   40.00   52.50   54.17   68.25  290.00       4 
> hist(dataset$age,main="Age Distribution", xlab="Age", xlim = c(0,300),ylim=c(0,30), breaks=30)
> 
```

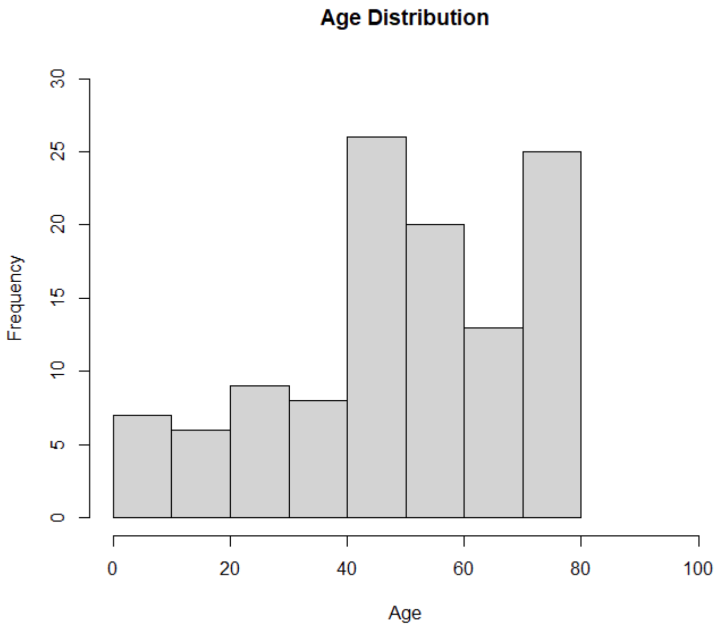


The minimum age value is within the valid range. Some age values are more than 250 which are numerical but impossible. To get rid of them 3 things can be done-

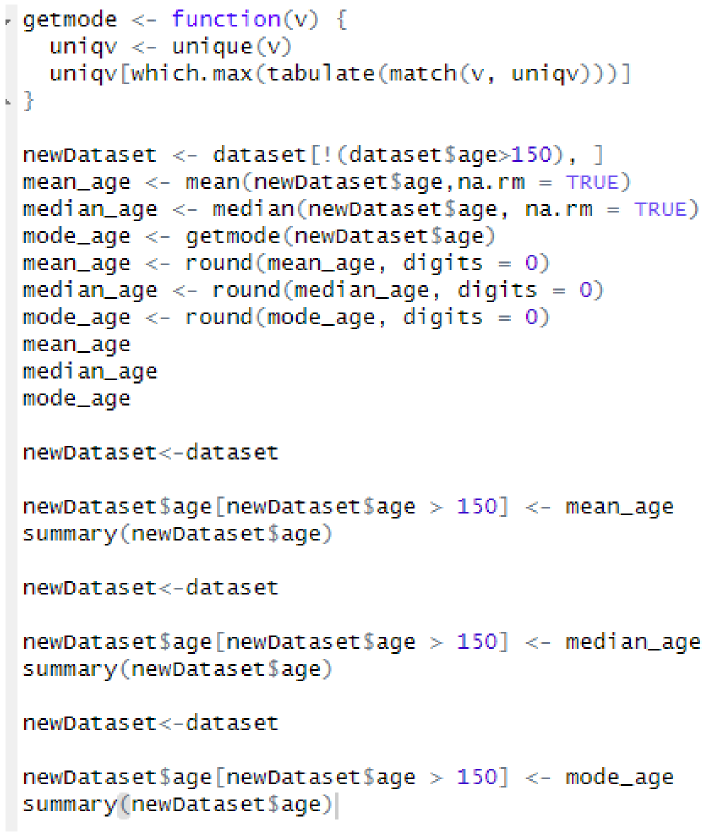
1. Remove the instances with outliers.



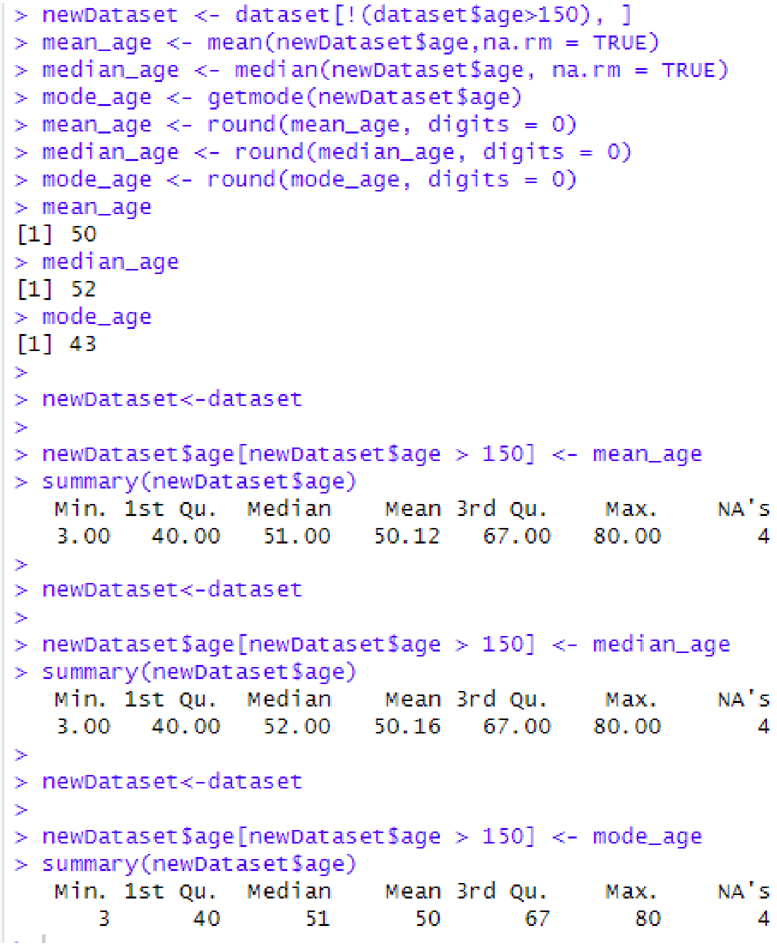
```
> newDataset <- dataset[!(dataset$age>150), ]
> summary(newDataset$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
   3.00   40.00   52.00   50.12   67.00   80.00       4 
> hist(newDataset$age,main="Age Distribution", xlab="Age", xlim = c(0,100),ylim=c(0,30), breaks=10)
> 
```



2. Replace with mean/mode/median of valid values.



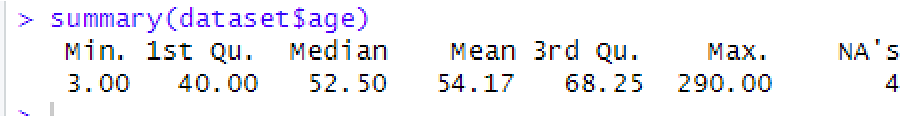
```
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

newDataset <- dataset[!(dataset$age>150), ]
mean_age <- mean(newDataset$age,na.rm = TRUE)
median_age <- median(newDataset$age, na.rm = TRUE)
mode_age <- getmode(newDataset$age)
mean_age <- round(mean_age, digits = 0)
median_age <- round(median_age, digits = 0)
mode_age <- round(mode_age, digits = 0)
mean_age
median_age
mode_age

newDataset<-dataset

newDataset$age[newDataset$age > 150] <- mean_age
summary(newDataset$age)

newDataset<-dataset

newDataset$age[newDataset$age > 150] <- median_age
summary(newDataset$age)

newDataset<-dataset

newDataset$age[newDataset$age > 150] <- mode_age
summary(newDataset$age)
```



```
> newDataset <- dataset[!(dataset$age>150), ]
> mean_age <- mean(newDataset$age,na.rm = TRUE)
> median_age <- median(newDataset$age, na.rm = TRUE)
> mode_age <- getmode(newDataset$age)
> mean_age <- round(mean_age, digits = 0)
> median_age <- round(median_age, digits = 0)
> mode_age <- round(mode_age, digits = 0)
> mean_age
[1] 50
> median_age
[1] 52
> mode_age
[1] 43
>
> newDataset<-dataset
>
> newDataset$age[newDataset$age > 150] <- mean_age
> summary(newDataset$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
   3.00   40.00   51.00   50.12   67.00   80.00       4 
>
> newDataset<-dataset
>
> newDataset$age[newDataset$age > 150] <- median_age
> summary(newDataset$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
   3.00   40.00   52.00   50.16   67.00   80.00       4 
>
> newDataset<-dataset
>
> newDataset$age[newDataset$age > 150] <- mode_age
> summary(newDataset$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
      3      40      51      50      67      80       4 
```

3.Retrieving data from simple univariate exploration in this case is difficult as both values are above 250(3 digits number) and it is difficult to determine what went wrong.

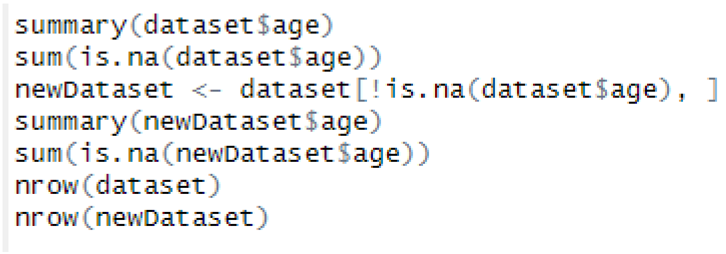
Summary shows that there are 4 missing values in the Age Attribute.



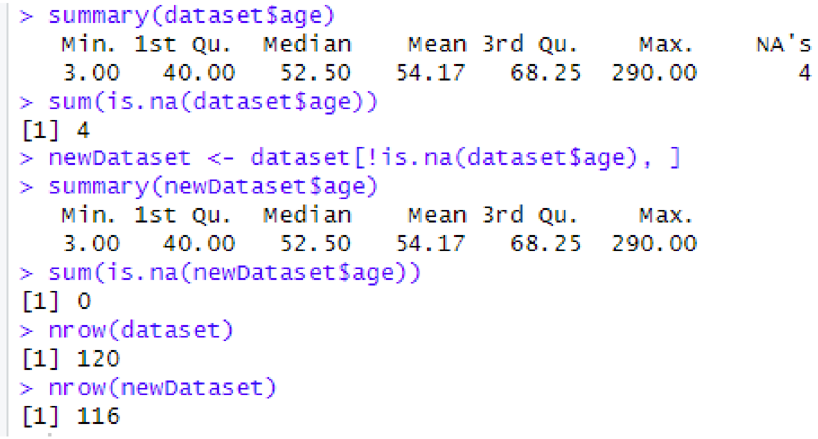
```
> summary(dataset$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
   3.00   40.00   52.50   54.17   68.25  290.00       4 
```

2 things can be done-

1. Remove instances that have missing values in the Age Attribute

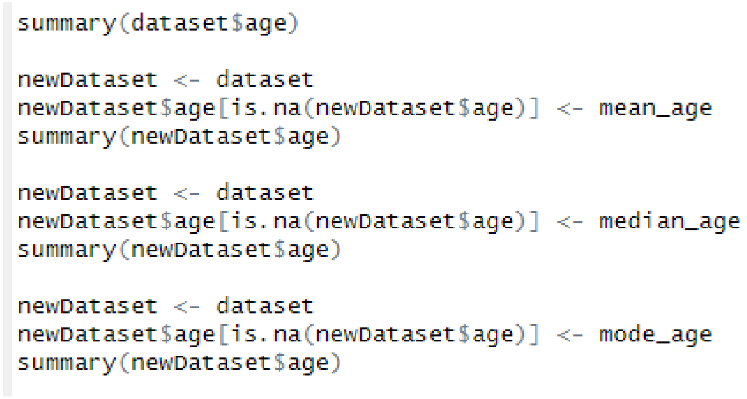


```
summary(dataset$age)
sum(is.na(dataset$age))
newDataset <- dataset[!is.na(dataset$age), ]
summary(newDataset$age)
sum(is.na(newDataset$age))
nrow(dataset)
nrow(newDataset)
```

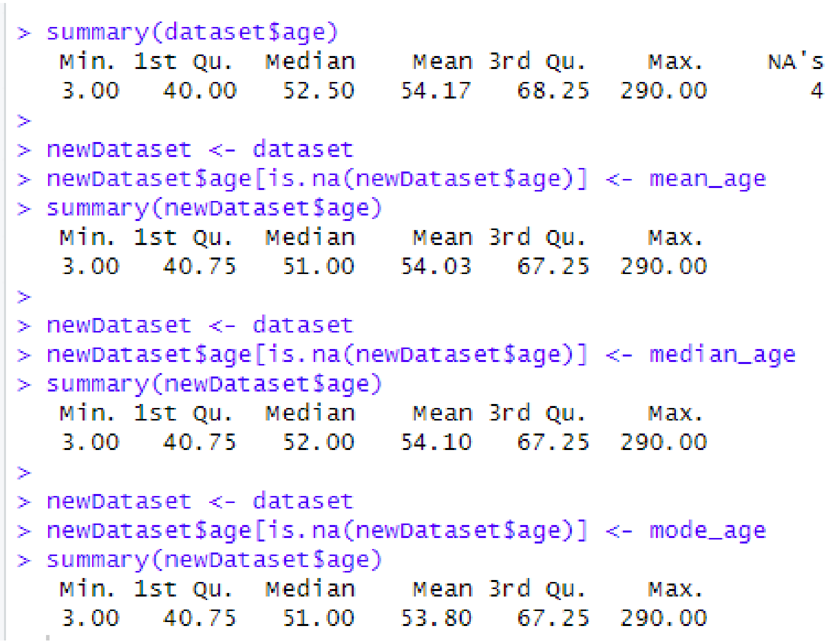


```
> summary(dataset$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
   3.00   40.00   52.50   54.17   68.25  290.00       4 
> sum(is.na(dataset$age))
[1] 4
> newDataset <- dataset[!is.na(dataset$age), ]
> summary(newDataset$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   3.00   40.00   52.50   54.17   68.25  290.00 
> sum(is.na(newDataset$age))
[1] 0
> nrow(dataset)
[1] 120
> nrow(newDataset)
[1] 116
```

2. Replace missing values with mean/median/mode of valid Age values.



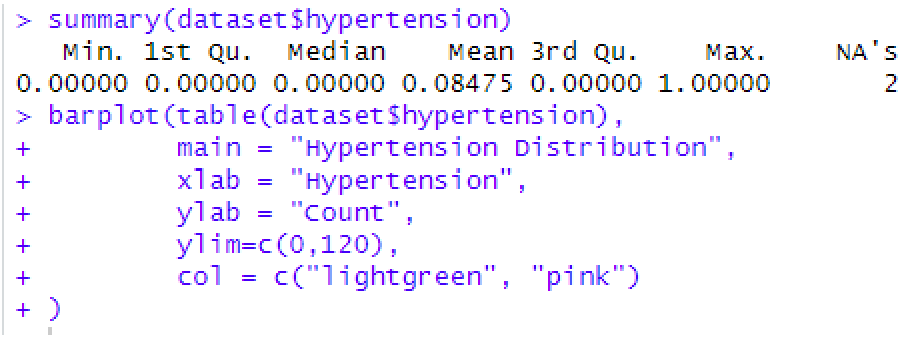
```
summary(dataset$age)

newDataset <- dataset
newDataset$age[is.na(newDataset$age)] <- mean_age
summary(newDataset$age)

newDataset <- dataset
newDataset$age[is.na(newDataset$age)] <- median_age
summary(newDataset$age)

newDataset <- dataset
newDataset$age[is.na(newDataset$age)] <- mode_age
summary(newDataset$age)
```



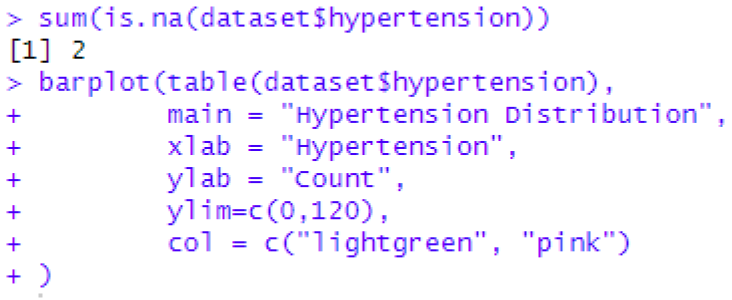
```
> summary(dataset$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
   3.00   40.00   52.50   54.17   68.25  290.00       4 
> 
> newDataset <- dataset
> newDataset$age[is.na(newDataset$age)] <- mean_age
> summary(newDataset$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   3.00   40.75   51.00   54.03   67.25  290.00 
> 
> newDataset <- dataset
> newDataset$age[is.na(newDataset$age)] <- median_age
> summary(newDataset$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   3.00   40.75   52.00   54.10   67.25  290.00 
> 
> newDataset <- dataset
> newDataset$age[is.na(newDataset$age)] <- mode_age
> summary(newDataset$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   3.00   40.75   51.00   53.80   67.25  290.00 
```

How Mean, median and mode were determined is shown in previous figure where impossible values were being replaced.

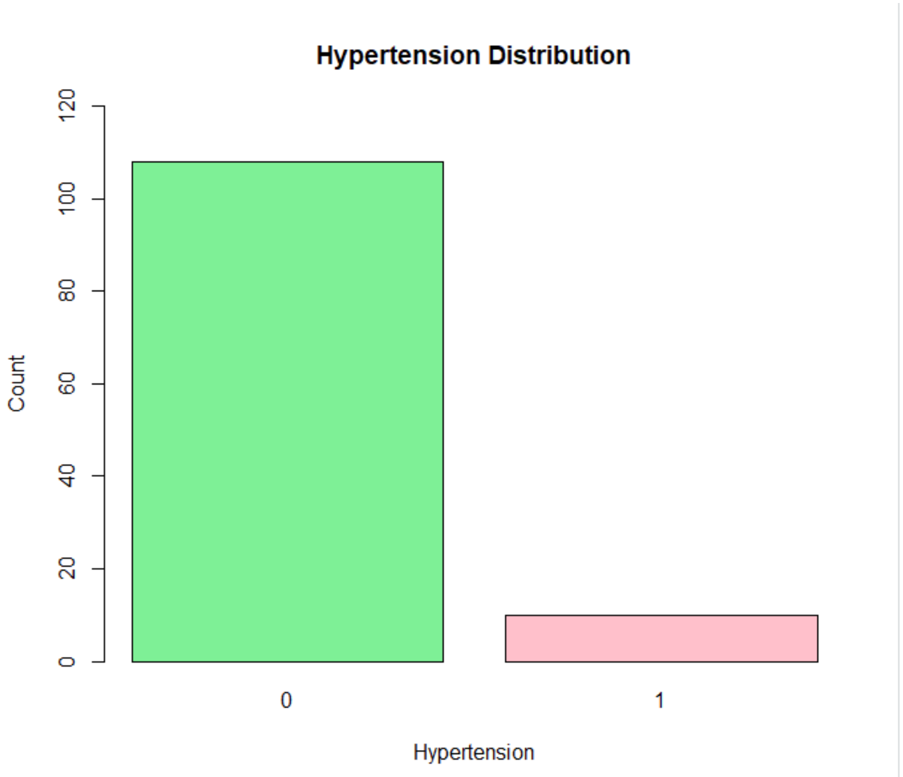
3.For hypertension attribute:



```
> summary(dataset$hypertension)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
0.00000 0.00000 0.00000 0.08475 0.00000 1.00000       2 
> barplot(table(dataset$hypertension),
+         main = "Hypertension Distribution",
+         xlab = "Hypertension",
+         ylab = "Count",
+         ylim=c(0,120),
+         col = c("lightgreen", "pink")
+ )
```

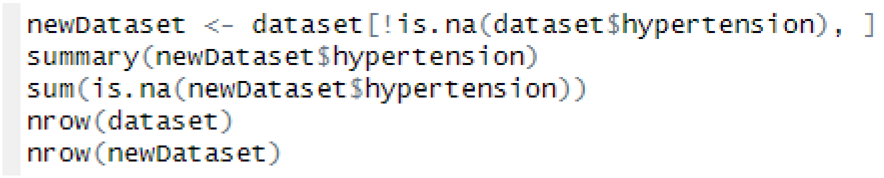


```
> sum(is.na(dataset$hypertension))
[1] 2
> barplot(table(dataset$hypertension),
+         main = "Hypertension Distribution",
+         xlab = "Hypertension",
+         ylab = "Count",
+         ylim=c(0,120),
+         col = c("lightgreen", "pink")
+ )
```

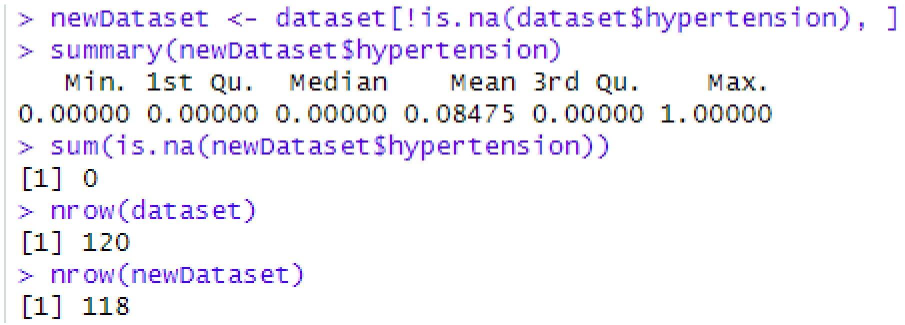


There are 2 missing values. So, three things can be done-

1.Remove instances where hypertension has missing values

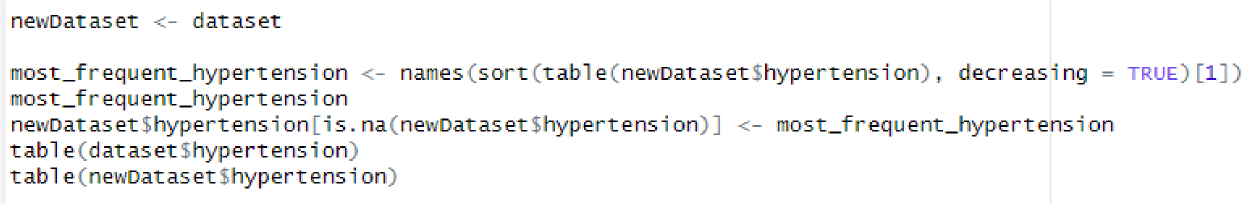


```
newDataset <- dataset[!is.na(dataset$hypertension), ]
summary(newDataset$hypertension)
sum(is.na(newDataset$hypertension))
nrow(dataset)
nrow(newDataset)
```

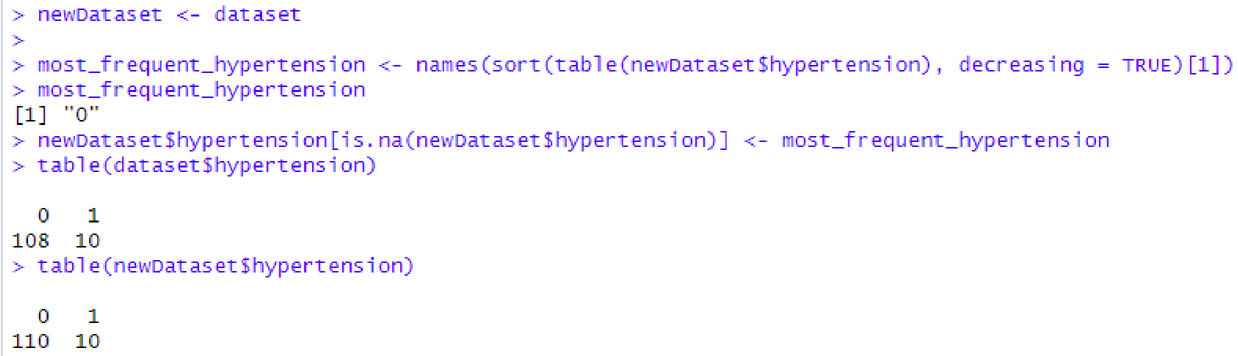


```
> newDataset <- dataset[!is.na(dataset$hypertension), ]
> summary(newDataset$hypertension)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
0.00000 0.00000 0.00000 0.08475 0.00000 1.00000 
> sum(is.na(newDataset$hypertension))
[1] 0
> nrow(dataset)
[1] 120
> nrow(newDataset)
[1] 118
```

2.Replace with highest occurrence.



```
newDataset <- dataset

most_frequent_hypertension <- names(sort(table(newDataset$hypertension), decreasing = TRUE)[1])
most_frequent_hypertension
newDataset$hypertension[is.na(newDataset$hypertension)] <- most_frequent_hypertension
table(dataset$hypertension)
table(newDataset$hypertension)
```



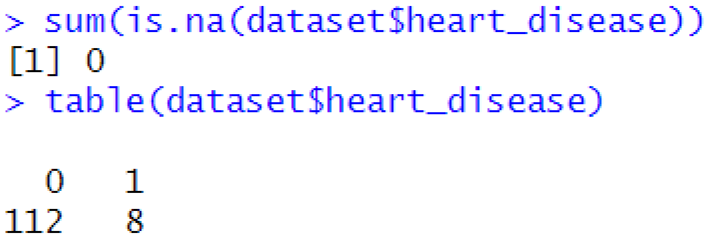
```
> newDataset <- dataset
> 
> most_frequent_hypertension <- names(sort(table(newDataset$hypertension), decreasing = TRUE)[1])
> most_frequent_hypertension
[1] "0"
> newDataset$hypertension[is.na(newDataset$hypertension)] <- most_frequent_hypertension
> table(dataset$hypertension)

  0   1 
108  10 
> table(newDataset$hypertension)

  0   1 
110  10 
```

3.Missing values can be kept as it is for future edition or some implementation might perform better with missing values in place. It is contextual.

4.For heart_disease attribute:

In this attribute there are 120 instances and there is only numerical value which is only 0 and 1. So, let's check for any missing value first-

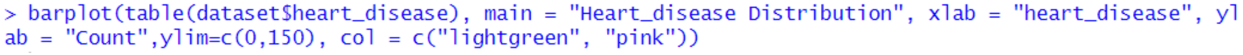
For this, we can apply is.na() function.



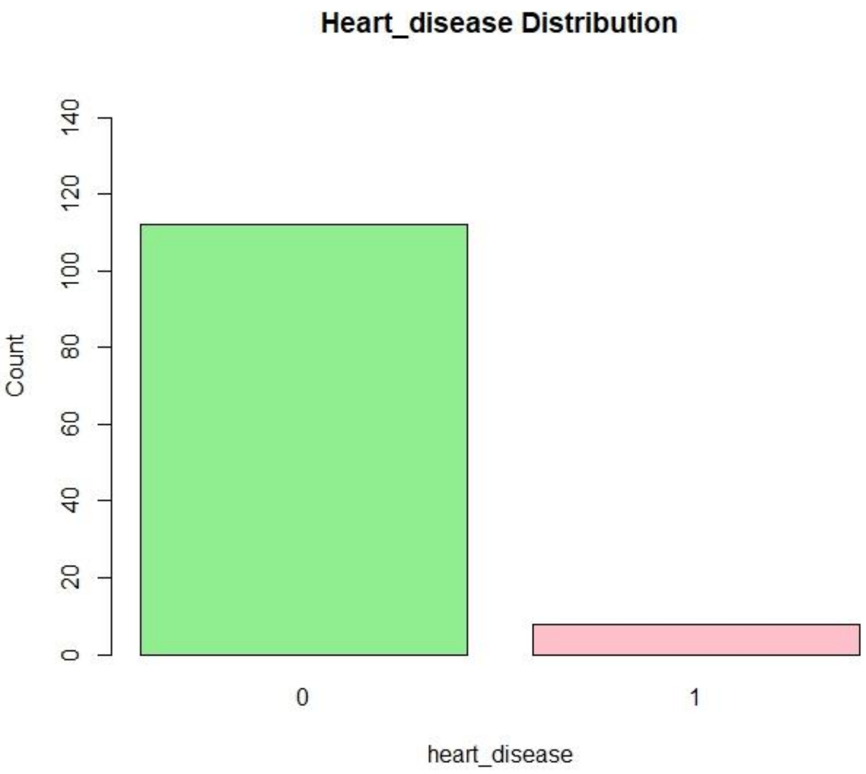
```
> sum(is.na(dataset$heart_disease))
[1] 0
> table(dataset$heart_disease)

  0   1 
112   8 
```

That means there is no missing value in this attribute.

Now, for univariate data exploration in heart_disease attribute lets draw a bar plot.

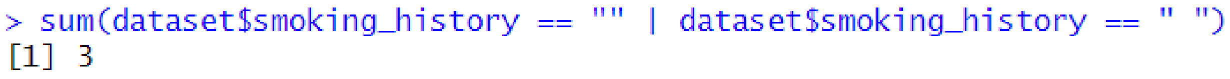


```
> barplot(table(dataset$heart_disease), main = "Heart_disease Distribution", xlab = "heart_disease", yl
ab = "Count",ylim=c(0,150), col = c("lightgreen", "pink"))
```

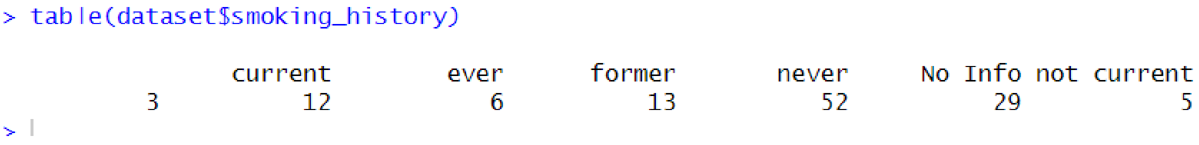


5.For smoking_history attribute:

Categorical values are contained in this attribute. It is evident that some values are missing. Let’s count the number of missing values first.

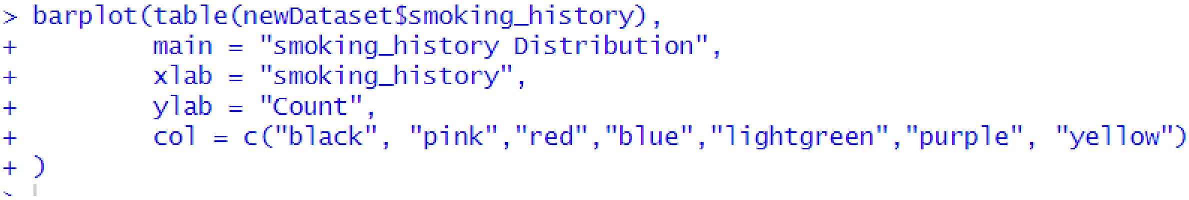


```
> sum(dataset$smoking_history == "" | dataset$smoking_history == " ")
[1] 3
```

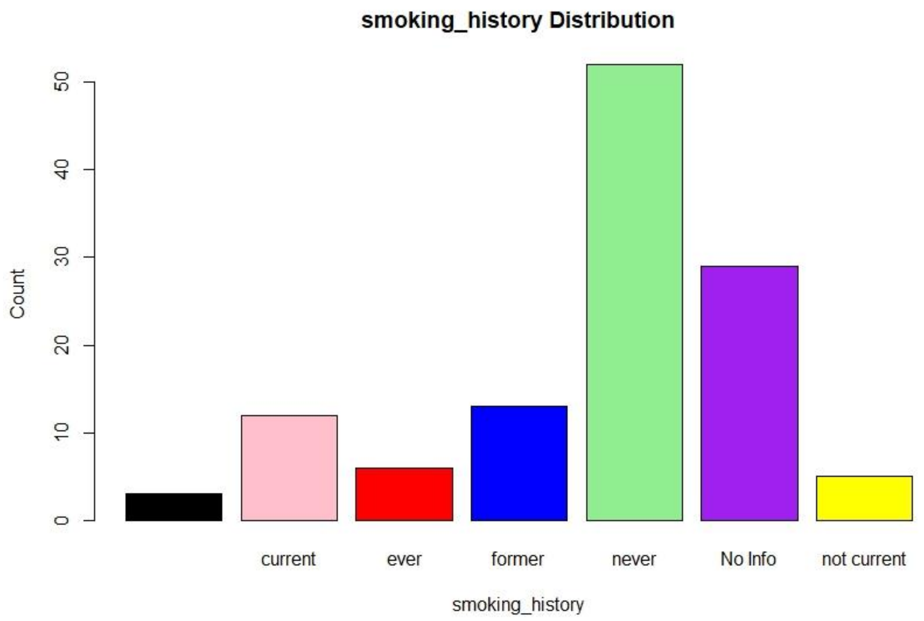


```
> table(dataset$smoking_history)

                 current        ever      former       never     No Info not current 
          3           12           6          13          52          29           5 
> 
```

In other words, there are 117 instances in this attribute when the current smoker is 12, former smoker is 13, never smoked 52, ever smoker 6, not current smoker is 5 and there is no info for 29 patients. Given that it is a categorical attribute, let's create a bar plot using univariate explanation.

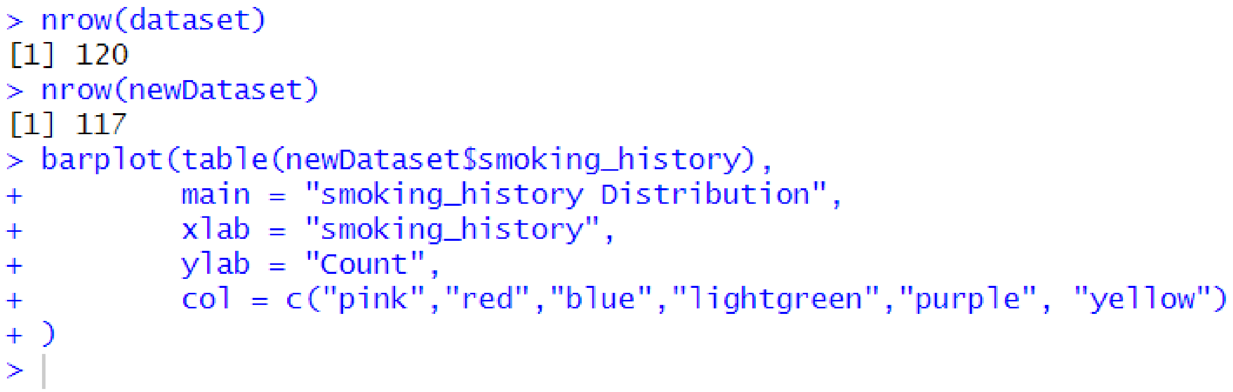


```
> barplot(table(newDataset$smoking_history),
+         main = "smoking_history Distribution",
+         xlab = "smoking_history",
+         ylab = "Count",
+         col = c("black", "pink","red","blue","lightgreen","purple", "yellow")
+ )
>
```

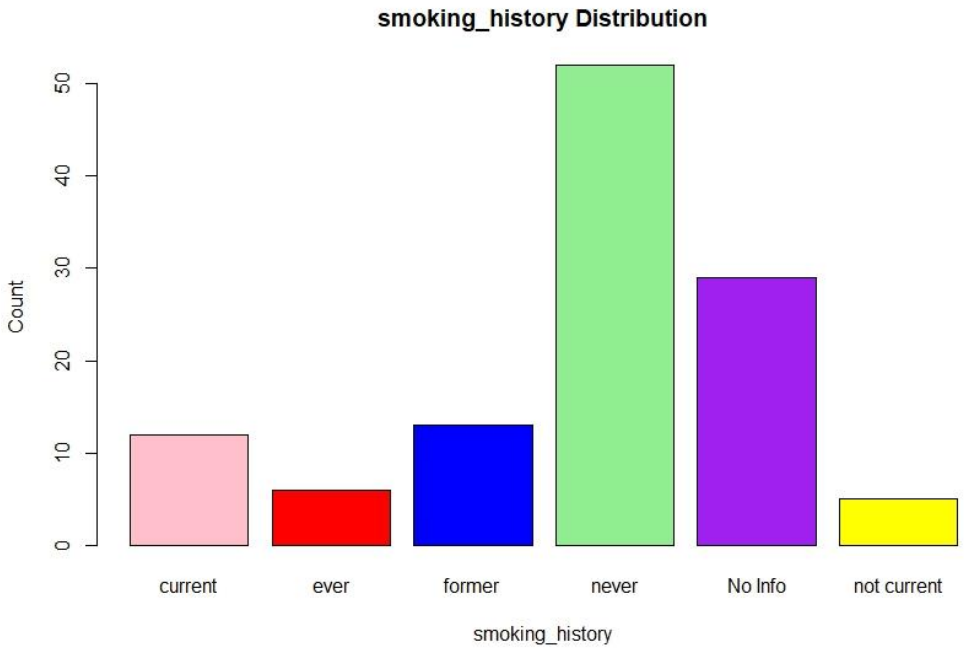


Three methods exist for us to eliminate this missing instance.

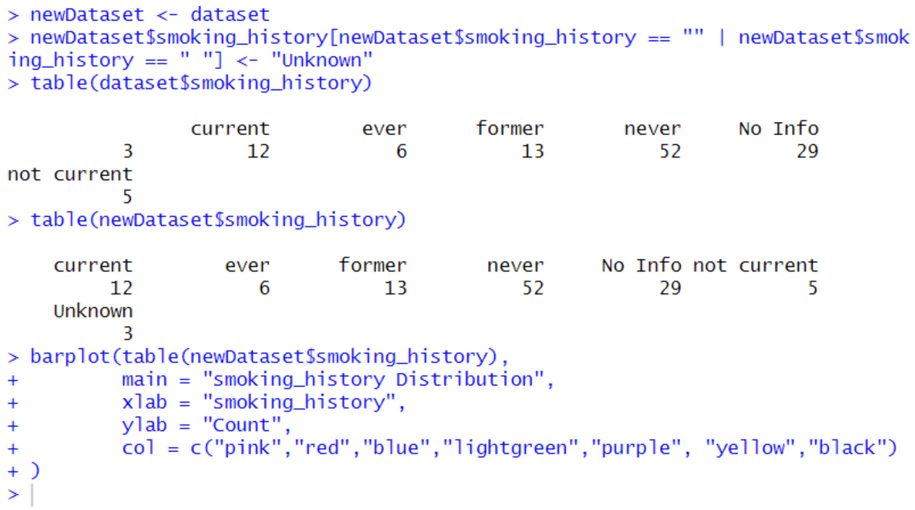
1.The missing values in the dataset can be eliminated.



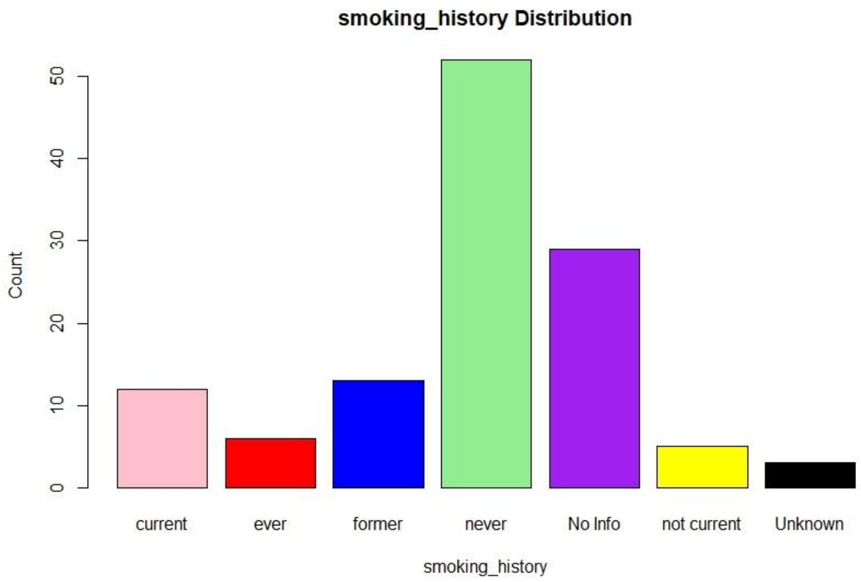
```
> nrow(dataset)
[1] 120
> nrow(newDataset)
[1] 117
> barplot(table(newDataset$smoking_history),
+         main = "smoking_history Distribution",
+         xlab = "smoking_history",
+         ylab = "Count",
+         col = c("pink","red","blue","lightgreen","purple", "yellow")
+ )
>
```



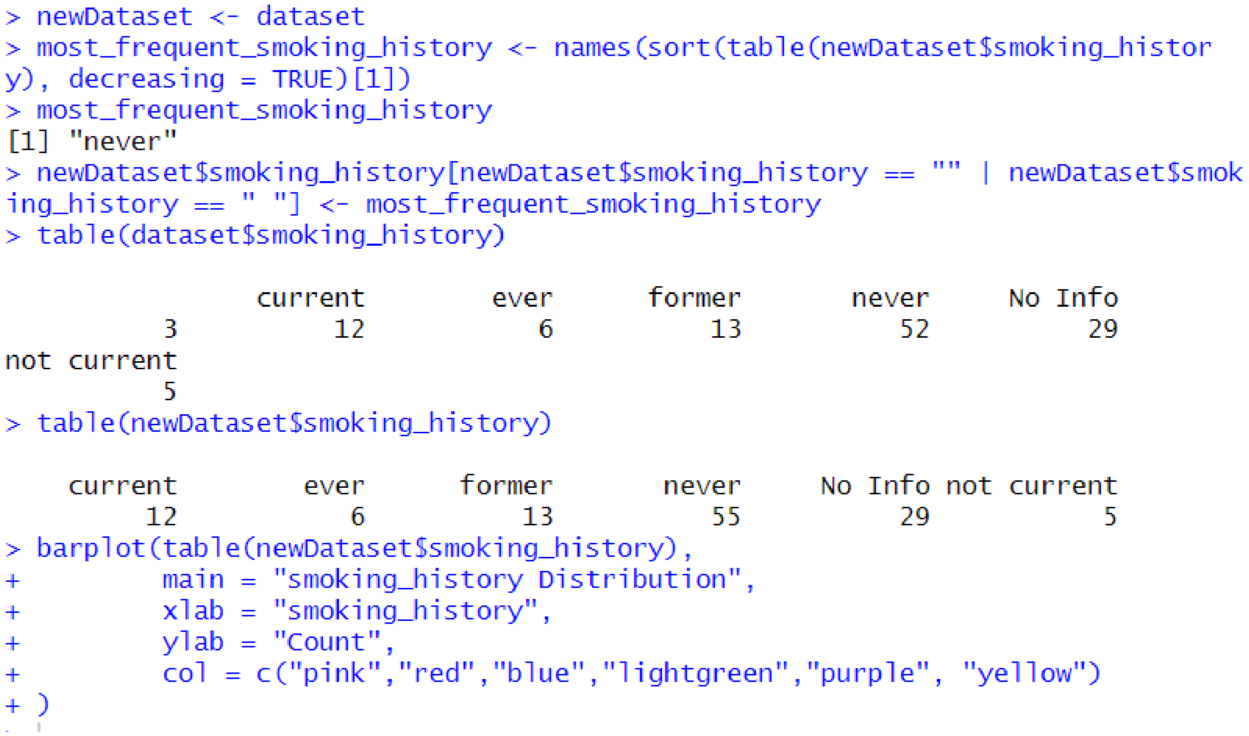
2. Use a place holder to fill in the missing value.

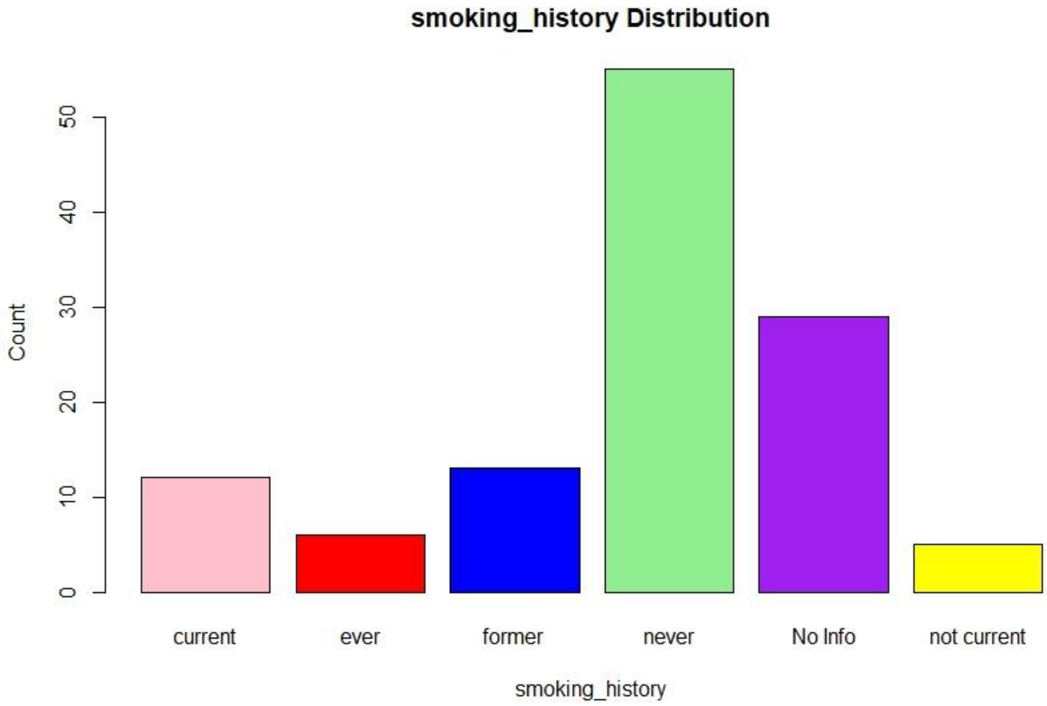


```
> newDataset <- dataset
> newDataset$smoking_history[newDataset$smoking_history == "" | newDataset$smok
ing_history == " "] <- "Unknown"
> table(dataset$smoking_history)

                current        ever      former       never     No Info 
          3          12           6          13          52          29 
not current 
          5 
> table(newDataset$smoking_history)

    current        ever      former       never     No Info not current 
         12           6          13          52          29           5 
    Unknown 
          3 
> barplot(table(newDataset$smoking_history),
+         main = "smoking_history Distribution",
+         xlab = "smoking_history",
+         ylab = "Count",
+         col = c("pink","red","blue","lightgreen","purple", "yellow","black")
+ )
> 
```

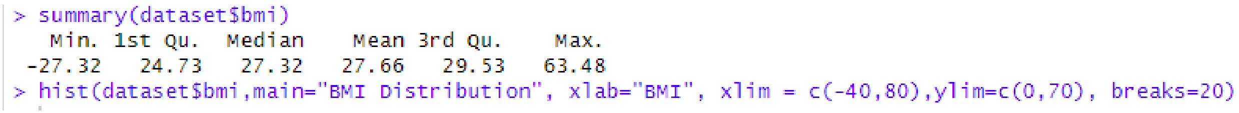


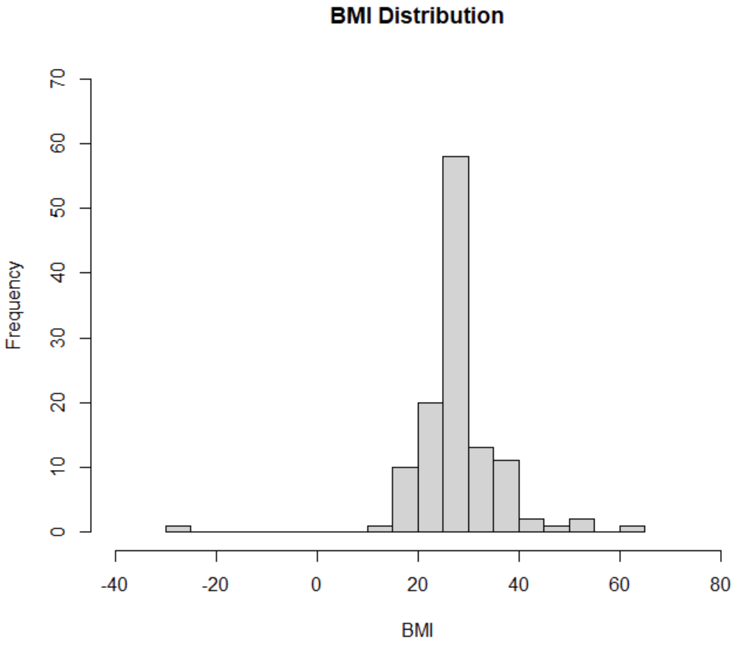
3. Replace with the value that occurs most frequently.


```
> newDataset <- dataset
> most_frequent_smoking_history <- names(sort(table(newDataset$smoking_histor
y), decreasing = TRUE)[1])
> most_frequent_smoking_history
[1] "never"
> newDataset$smoking_history[newDataset$smoking_history == "" | newDataset$smok
ing_history == " "] <- most_frequent_smoking_history
> table(dataset$smoking_history)

                current        ever      former       never     No Info 
          3          12           6          13          52          29 
not current 
          5 
> table(newDataset$smoking_history)

    current        ever      former       never     No Info not current 
         12           6          13          55          29           5 
> barplot(table(newDataset$smoking_history),
+         main = "smoking_history Distribution",
+         xlab = "smoking_history",
+         ylab = "Count",
+         col = c("pink","red","blue","lightgreen","purple", "yellow")
+ )
```



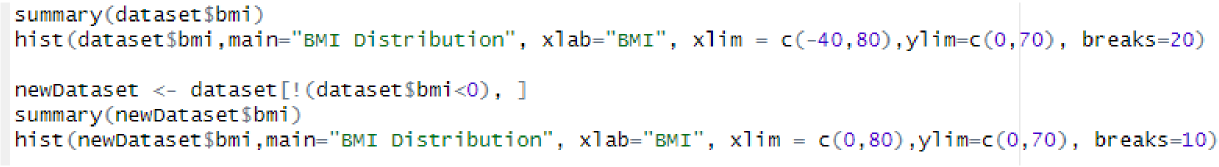
6.For bmi attribute:


```
> summary(dataset$bmi)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 -27.32   24.73   27.32   27.66   29.53   63.48 
> hist(dataset$bmi,main="BMI Distribution", xlab="BMI", xlim = c(-40,80),ylim=c(0,70), breaks=20)
```

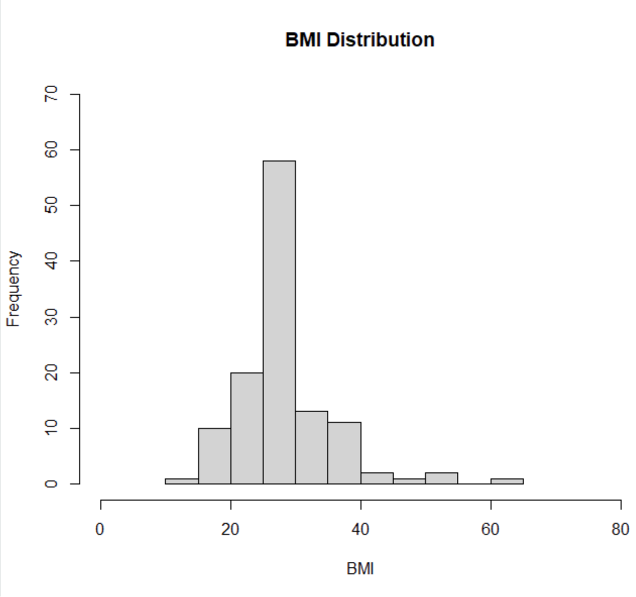


There are no missing values. It has invalid value. BMI cannot be negative. 3 things can be done-

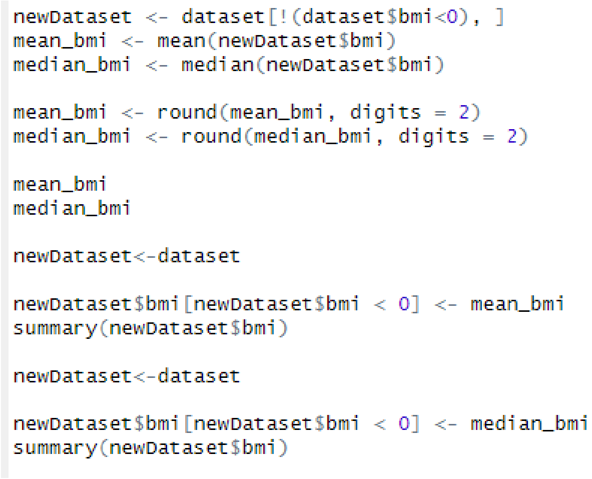
1.Remove the instances that have negative BMI value.



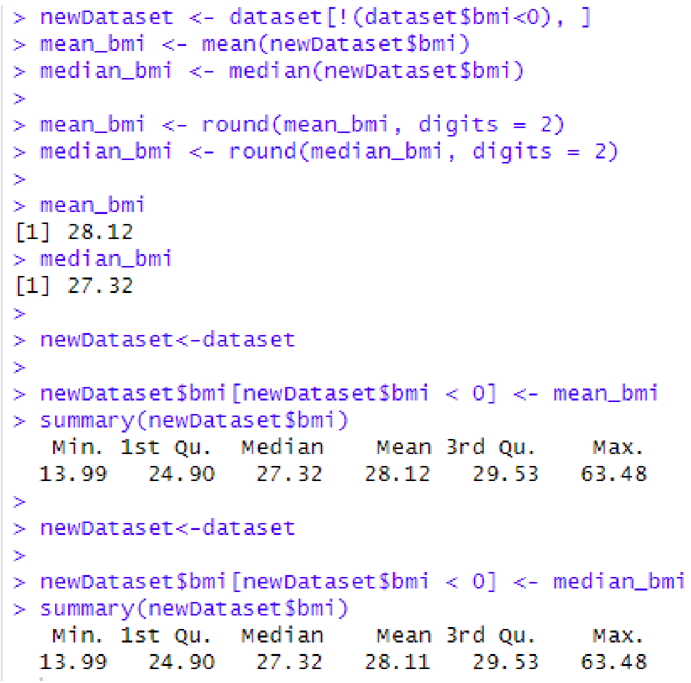
```
summary(dataset$bmi)
hist(dataset$bmi,main="BMI Distribution", xlab="BMI", xlim = c(-40,80),ylim=c(0,70), breaks=20)

newDataset <- dataset[!(dataset$bmi<0), ]
summary(newDataset$bmi)
hist(newDataset$bmi,main="BMI Distribution", xlab="BMI", xlim = c(0,80),ylim=c(0,70), breaks=10)
```



2.Replace the negative BMI values with mean/median of valid BMI values.

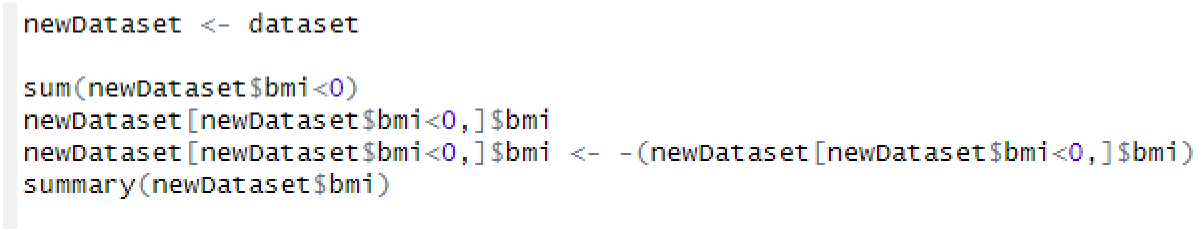


```
newDataset <- dataset[!(dataset$bmi<0), ]
mean_bmi <- mean(newDataset$bmi)
median_bmi <- median(newDataset$bmi)

mean_bmi <- round(mean_bmi, digits = 2)
median_bmi <- round(median_bmi, digits = 2)

mean_bmi
median_bmi

newDataset<-dataset

newDataset$bmi[newDataset$bmi < 0] <- mean_bmi
summary(newDataset$bmi)

newDataset<-dataset

newDataset$bmi[newDataset$bmi < 0] <- median_bmi
summary(newDataset$bmi)
```

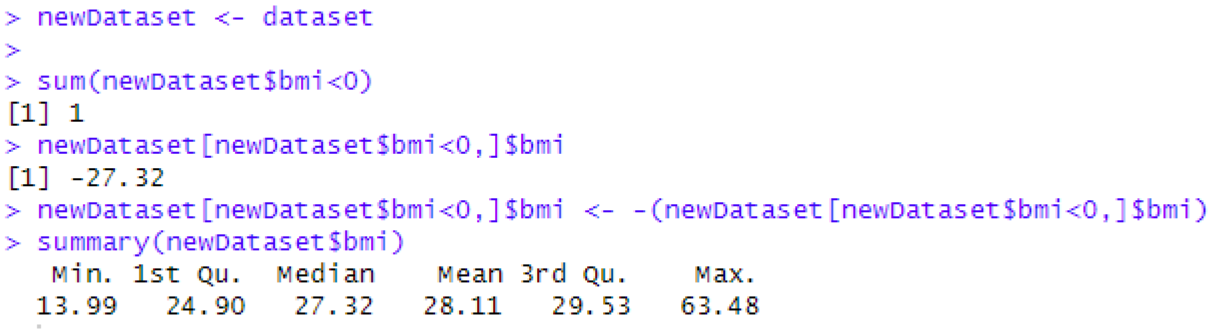


```
> newDataset <- dataset[!(dataset$bmi<0), ]
> mean_bmi <- mean(newDataset$bmi)
> median_bmi <- median(newDataset$bmi)
>
> mean_bmi <- round(mean_bmi, digits = 2)
> median_bmi <- round(median_bmi, digits = 2)
>
> mean_bmi
[1] 28.12
> median_bmi
[1] 27.32
>
> newDataset<-dataset
>
> newDataset$bmi[newDataset$bmi < 0] <- mean_bmi
> summary(newDataset$bmi)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  13.99   24.90   27.32   28.12   29.53   63.48 
>
> newDataset<-dataset
>
> newDataset$bmi[newDataset$bmi < 0] <- median_bmi
> summary(newDataset$bmi)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  13.99   24.90   27.32   28.11   29.53   63.48 
```

3.Retrieving the data



```
newDataset <- dataset

sum(newDataset$bmi<0)
newDataset[newDataset$bmi<0,]$bmi
newDataset[newDataset$bmi<0,]$bmi <- -(newDataset[newDataset$bmi<0,]$bmi)
summary(newDataset$bmi)
```



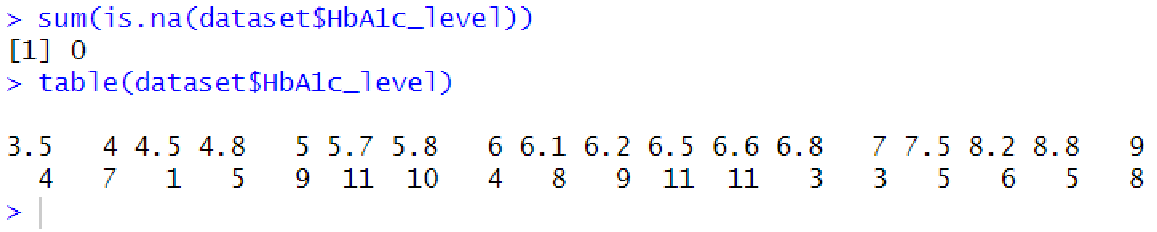
```
> newDataset <- dataset
>
> sum(newDataset$bmi<0)
[1] 1
> newDataset[newDataset$bmi<0,]$bmi
[1] -27.32
> newDataset[newDataset$bmi<0,]$bmi <- -(newDataset[newDataset$bmi<0,]$bmi)
> summary(newDataset$bmi)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  13.99   24.90   27.32   28.11   29.53   63.48 
```

There was only 1 data in the negative range. Upon closer inspection, it is possible that the data got corrupted by human or machine mistake and a negative sign came to be. Removing the negative sign can possibly retrieve the actual data.

7.For HbA1c_level attribute:

In this attribute there are 120 instances and there is only numerical value. So, let’s check for any missing value first-

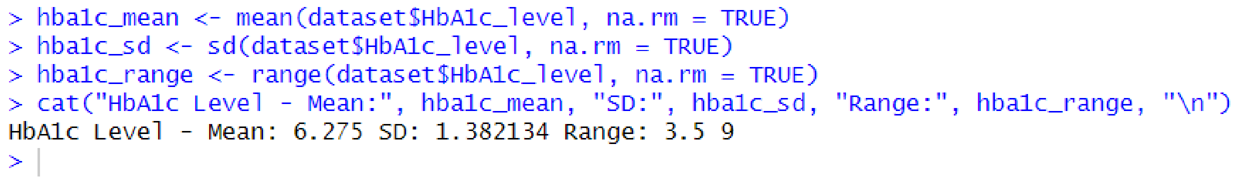
For this, we can apply is.na() function.



```
> sum(is.na(dataset$HbA1c_level))
[1] 0
> table(dataset$HbA1c_level)

3.5   4 4.5 4.8   5 5.7 5.8   6 6.1 6.2 6.5 6.6 6.8   7 7.5 8.2 8.8   9 
  4   7   1   5   9  11  10   4   8   9  11  11   3   3   5   6   5   8 
> 
```

That means there is no missing value in this attribute.

Now let’s check the mean, standard deviation and range of HbA1c attribute in this dataset.



```
> hba1c_mean <- mean(dataset$HbA1c_level, na.rm = TRUE)
> hba1c_sd <- sd(dataset$HbA1c_level, na.rm = TRUE)
> hba1c_range <- range(dataset$HbA1c_level, na.rm = TRUE)
> cat("HbA1c Level - Mean:", hba1c_mean, "SD:", hba1c_sd, "Range:", hba1c_range, "\n")
HbA1c Level - Mean: 6.275 SD: 1.382134 Range: 3.5 9
> 
```

HbA1c Level

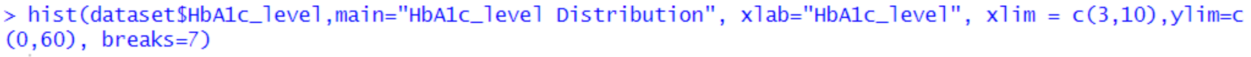
Count: 120 readings

Mean: Average HbA1c level is 6.275%.

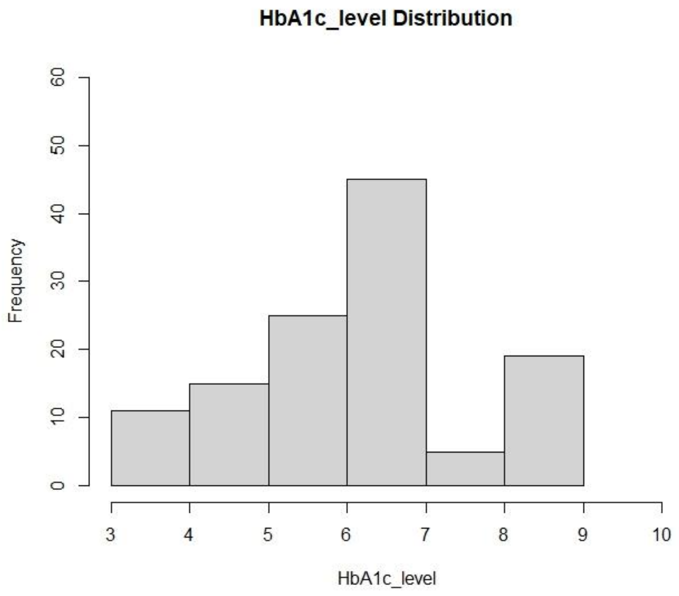
Standard Deviation: 1.382, indicating some variation in HbA1c levels among individuals.

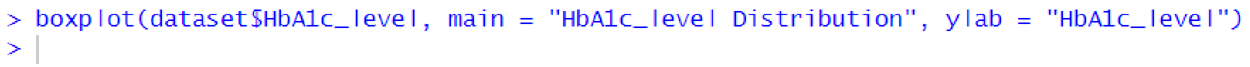
Range: From 3.5% to 9.0%, covering a broad range of HbA1c values.

Since the HbA1c attribute is a numerical attribute, now let's create a box plot and a histogram for univariate data exploration.

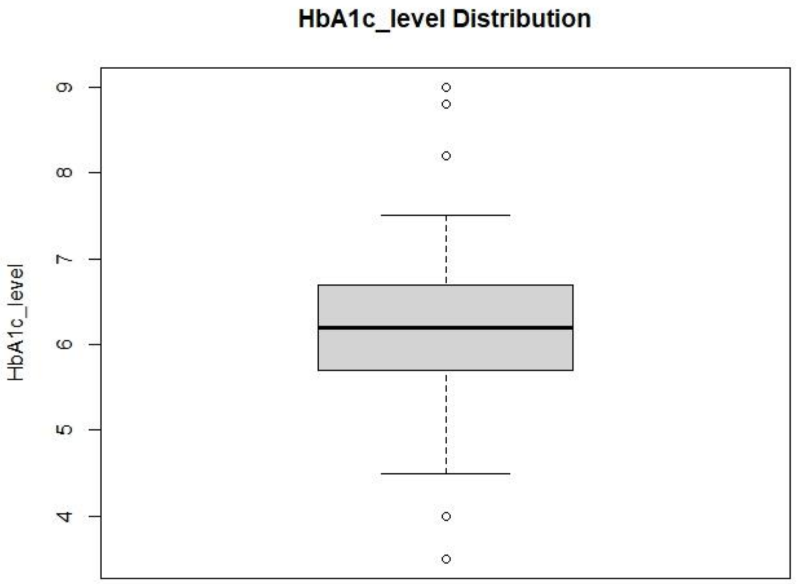


```
> hist(dataset$HbA1c_level,main="HbA1c_level Distribution", xlab="HbA1c_level", xlim = c(3,10),ylim=c
(0,60), breaks=7)
```





```
> boxplot(dataset$HbA1c_level, main = "HbA1c_level Distribution", ylab = "HbA1c_level")
> 
```

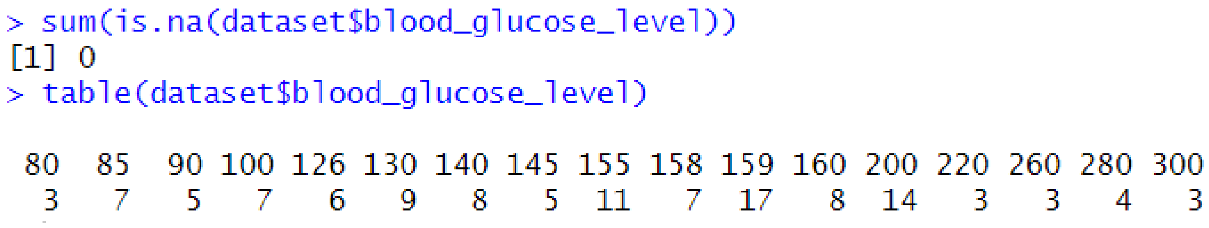


Even though some of the values in this example differ significantly from the majority of the values, they are not outliers or invalid.

8.For blood_glucose_level attribute:

In this attribute there are 120 instances and there is only numerical value. So, let's check for any missing value first-

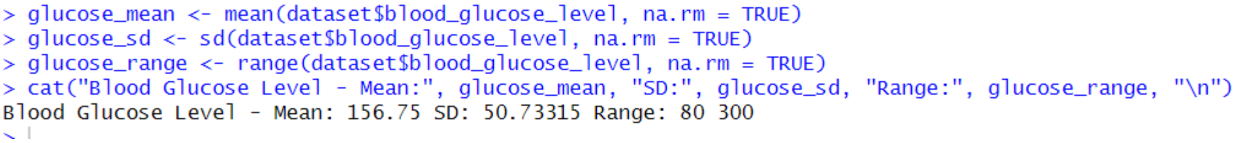
For this, we can apply is.na() function.



```
> sum(is.na(dataset$blood_glucose_level))
[1] 0
> table(dataset$blood_glucose_level)

 80  85  90 100 126 130 140 145 155 158 159 160 200 220 260 280 300 
  3   7   5   7   6   9   8   5  11   7  17   8  14   3   3   4   3 
```

That means there is no missing value in this attribute.

Now let's check the mean, standard deviation and range of blood glucose for this attribute in this dataset.



```
> glucose_mean <- mean(dataset$blood_glucose_level, na.rm = TRUE)
> glucose_sd <- sd(dataset$blood_glucose_level, na.rm = TRUE)
> glucose_range <- range(dataset$blood_glucose_level, na.rm = TRUE)
> cat("Blood Glucose Level - Mean:", glucose_mean, "SD:", glucose_sd, "Range:", glucose_range, "\n")
Blood Glucose Level - Mean: 156.75 SD: 50.73315 Range: 80 300
> 
```

Blood Glucose Level

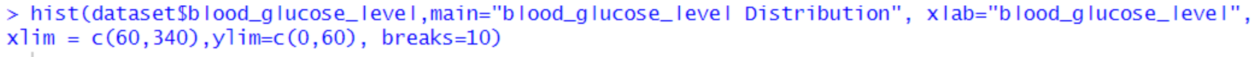
Count: 120 readings

Mean: Average blood glucose level is 156.75 mg/dL.

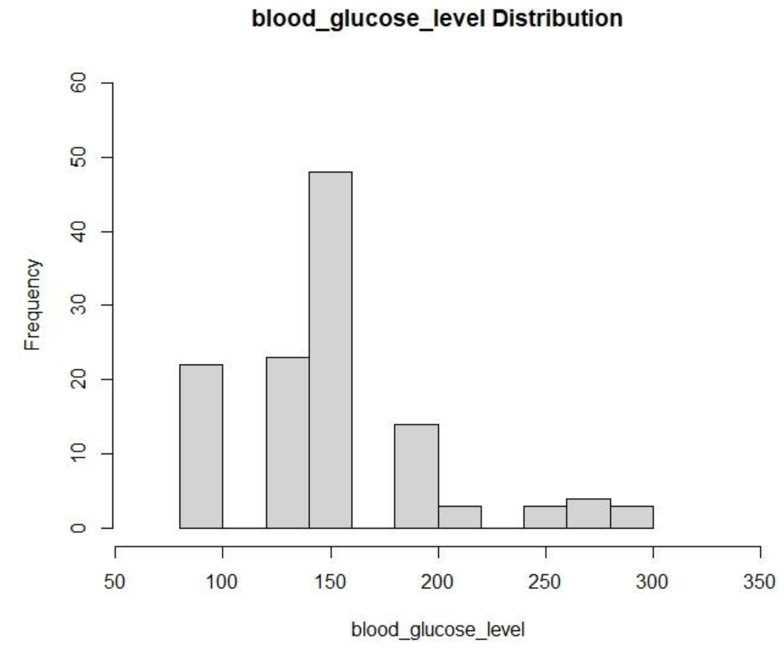
Standard Deviation: 50.733, showing significant variation among individuals.

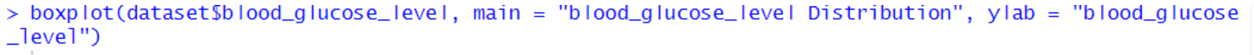
Range: 80 mg/dL to 300 mg/dL, indicating a wide range of glucose levels.

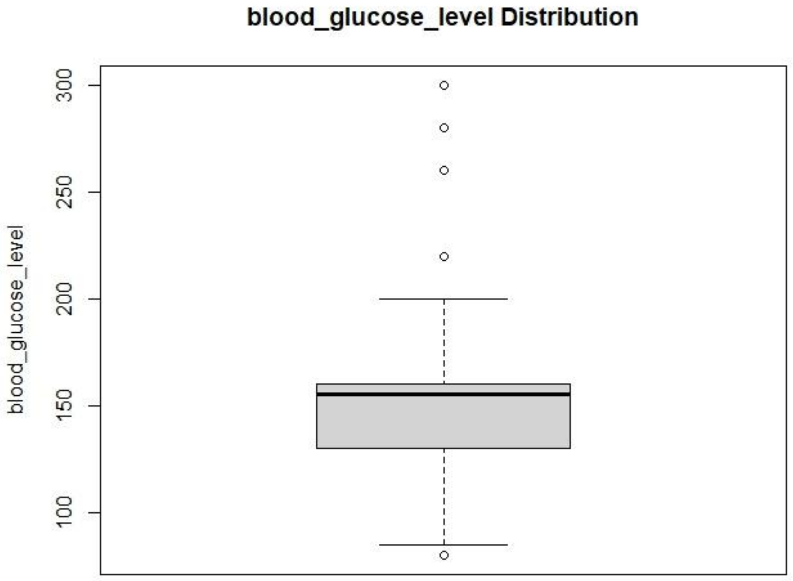
Since the blood glucose attribute is a numerical attribute, now let's create a box plot and a histogram for univariate data exploration.



```
> hist(dataset$blood_glucose_level,main="blood_glucose_level Distribution", xlab="blood_glucose_level",
xlim = c(60,340),ylim=c(0,60), breaks=10)
```




```
> boxplot(dataset$blood_glucose_level, main = "blood_glucose_level Distribution", ylab = "blood_glucose
_level")
```

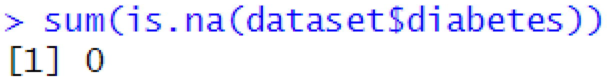


Even though some of the values in this example differ significantly from the majority of the values, they are not outliers or invalid.

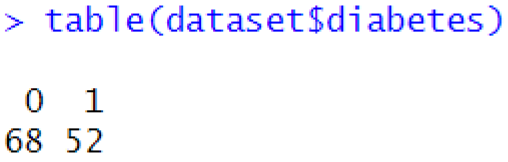
9.For Diabetes attribute:

In this attribute there are 120 instances and there is only numerical value which is only 0 and 1. So, let’s check for any missing value first-

For this, we can apply is.na() function.



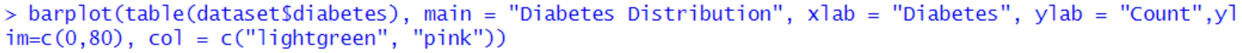
```
> sum(is.na(dataset$diabetes))
[1] 0
```



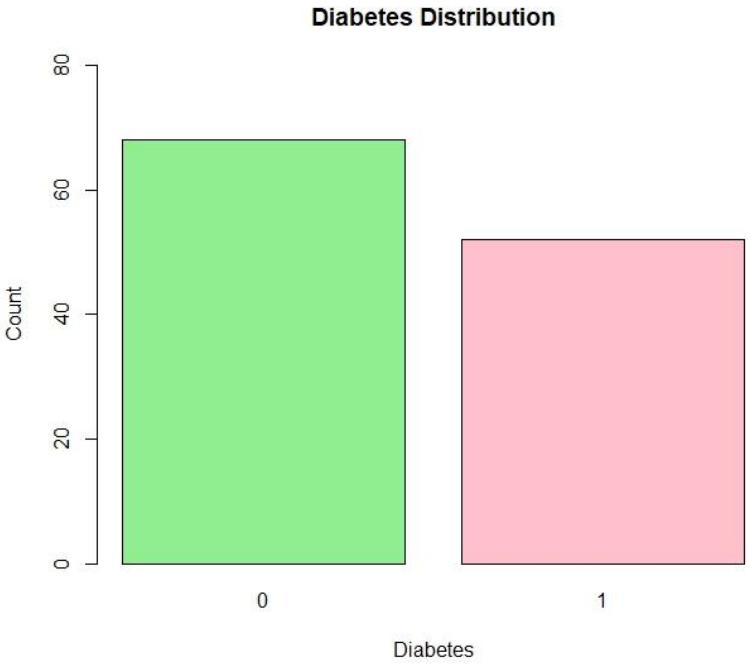
```
> table(dataset$diabetes)

 0  1 
68 52 
```

That means there is no missing value.

Now, for univariate data exploration in diabetes attribute let’s draw a bar plot.

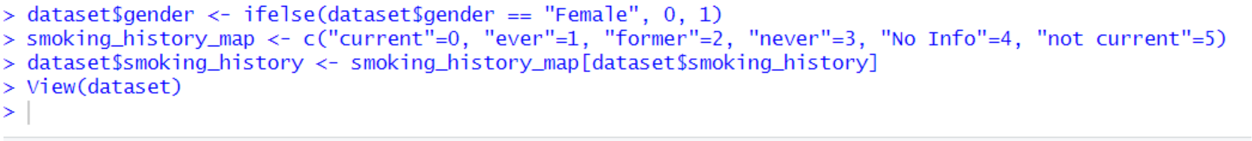


```
> barplot(table(dataset$diabetes), main = "Diabetes Distribution", xlab = "Diabetes", ylab = "Count",yl
im=c(0,80), col = c("lightgreen", "pink"))
```



Data types and conversion:

Some implementations might require numerical representation of categorical values**.**



```
> dataset$gender <- ifelse(dataset$gender == "Female", 0, 1)
> smoking_history_map <- c("current"=0, "ever"=1, "former"=2, "never"=3, "No Info"=4, "not current"=5)
> dataset$smoking_history <- smoking_history_map[dataset$smoking_history]
> View(dataset)
> 
```

Gender and smoking history are categorical attributes in the dataset, but we can substitute numerical values for them, such as female for 0 and male for 1.