edx_flora_ownproject_diabetes_prediction

Flora NIYOKWIZERWA

11/21/2022

Introduction

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether a patient has diabetes, based on certain diagnostic measurements included in the dataset. different approaches were used to carry oout this project like cross varidation model, regression model to predict the outcomes I downloaded the dataset from kaggle.com and placed inside my project i Then read the csv file from my working directory. The goal of this project is to predict the person who is likely to have a diabete basing on different criterias

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
## Warning: package 'tibble' was built under R version 4.1.3
```

```
## Warning: package 'tidyr' was built under R version 4.1.3
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```
## Warning: package 'purrr' was built under R version 4.1.3
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
## Warning: package 'stringr' was built under R version 4.1.3
```

```
## Warning: package 'forcats' was built under R version 4.1.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(readr)
library(dplyr)
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.1.3
```

```
## corrplot 0.92 loaded
```

```
library(tidyr)
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.1.3
```

```
library(naivebayes)
```

```
## Warning: package 'naivebayes' was built under R version 4.1.3
```

```
## naivebayes 0.9.7 loaded
```

```
# here i'm going to set my working directory where my dataset will be loaded
setwd("C:/Users/Flora/OneDrive/Desktop/RStudio Projects/diabetes/data")

# after setting working directory, I'm going to read my data set
diabetes_dataset <- read.csv("diabetes.csv")

#now after reading my dataset I start data exploratory and analysis of my dataset

#I use head to return my observation of data set
head(diabetes_dataset)
```

```
##   Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1           6     148            72            35       0 33.6
## 2           1      85            66            29       0 26.6
## 3           8     183            64             0       0 23.3
## 4           1      89            66            23      94 28.1
## 5           0     137            40            35     168 43.1
## 6           5     116            74             0       0 25.6
##   DiabetesPedigreeFunction Age Outcome
## 1                    0.627  50       1
## 2                    0.351  31       0
## 3                    0.672  32       1
## 4                    0.167  21       0
## 5                    2.288  33       1
## 6                    0.201  30       0
```

Method/Analysis section and results

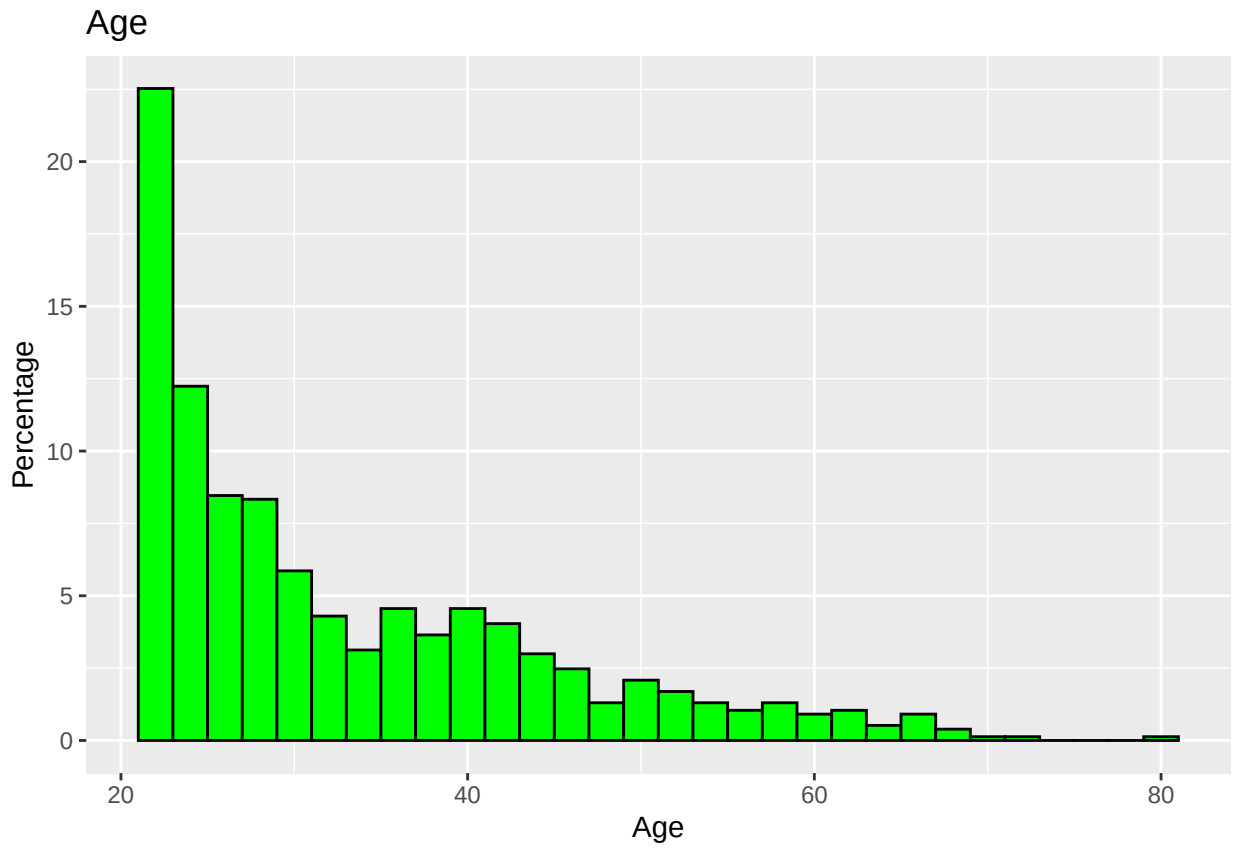
In this section I carried out my analysis on my dataset to achieve my goal and different data visualization was presented within this section

```
#here I'm going to get statistical analysis of my dataset

summary(diabetes_dataset)
```

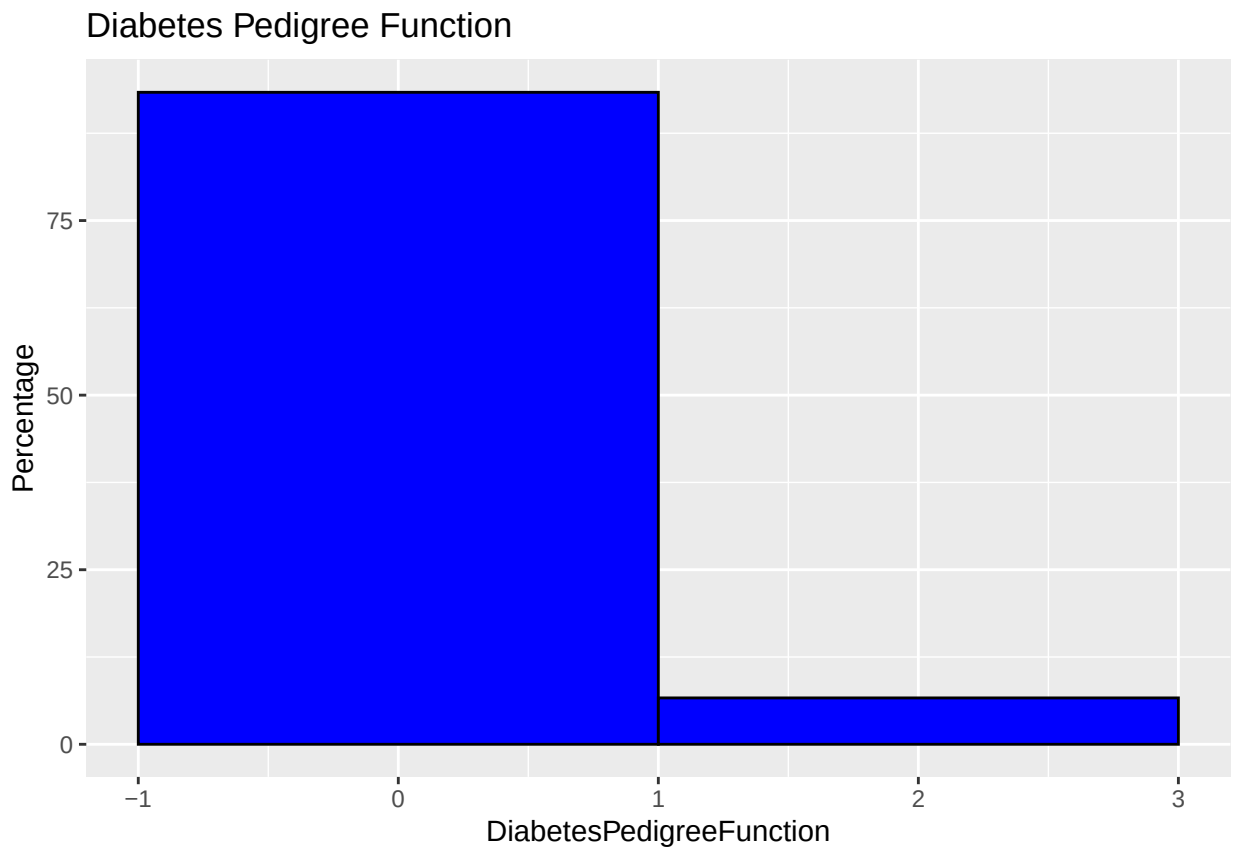
```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##     Insulin           BMI        DiabetesPedigreeFunction      Age
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780           Min.   :21.00
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437           1st Qu.:24.00
##  Median : 30.5   Median :32.00   Median :0.3725           Median :29.00
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719           Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262           3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200           Max.   :81.00
```

```
##     Outcome
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
```

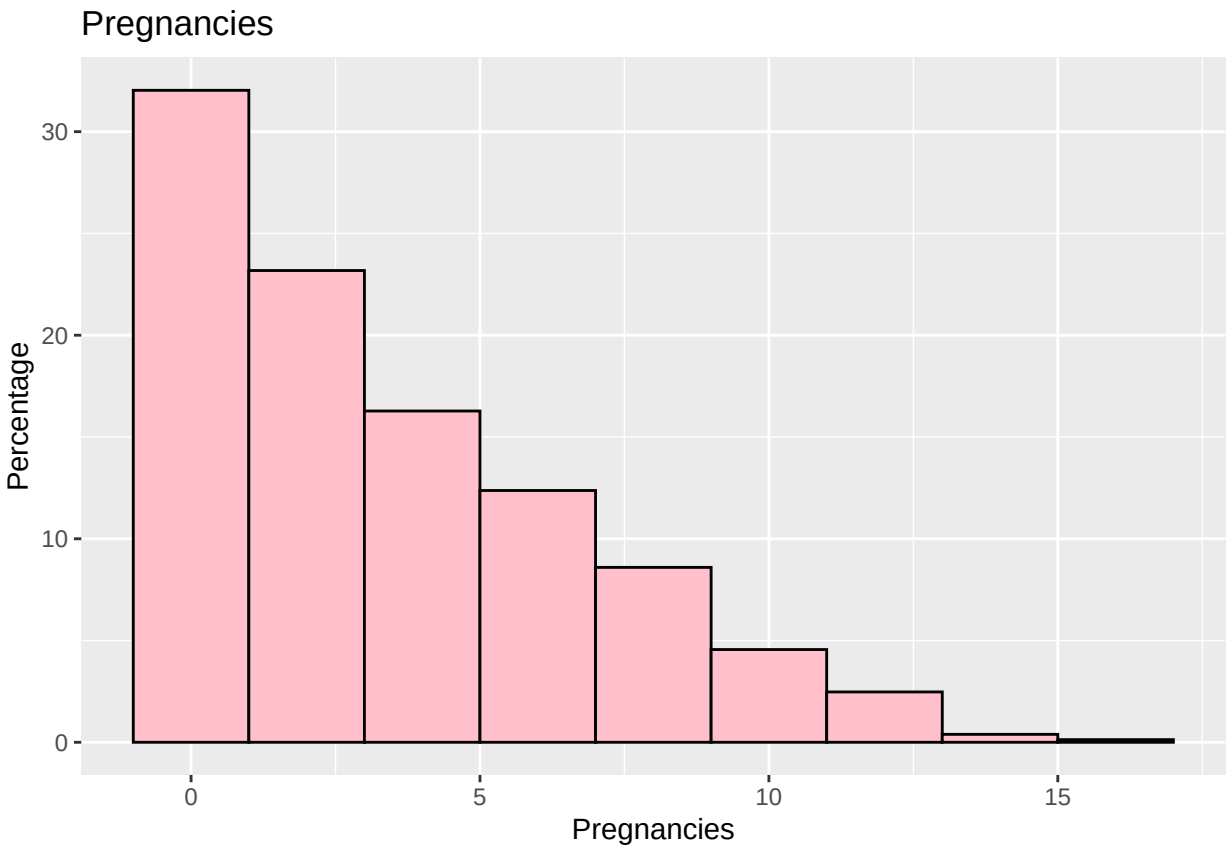
```
#I'm going to represent my data in visual representation to check if they have reasonable distribution
# age histogram
diabetes_dataset %>%
  ggplot(aes(x=Age))+ggtitle("Age")+
  geom_histogram(aes(y= 100*(..count..)/sum(..count..)),binwidth = 2, color="black",fill="green")+ylab(
```



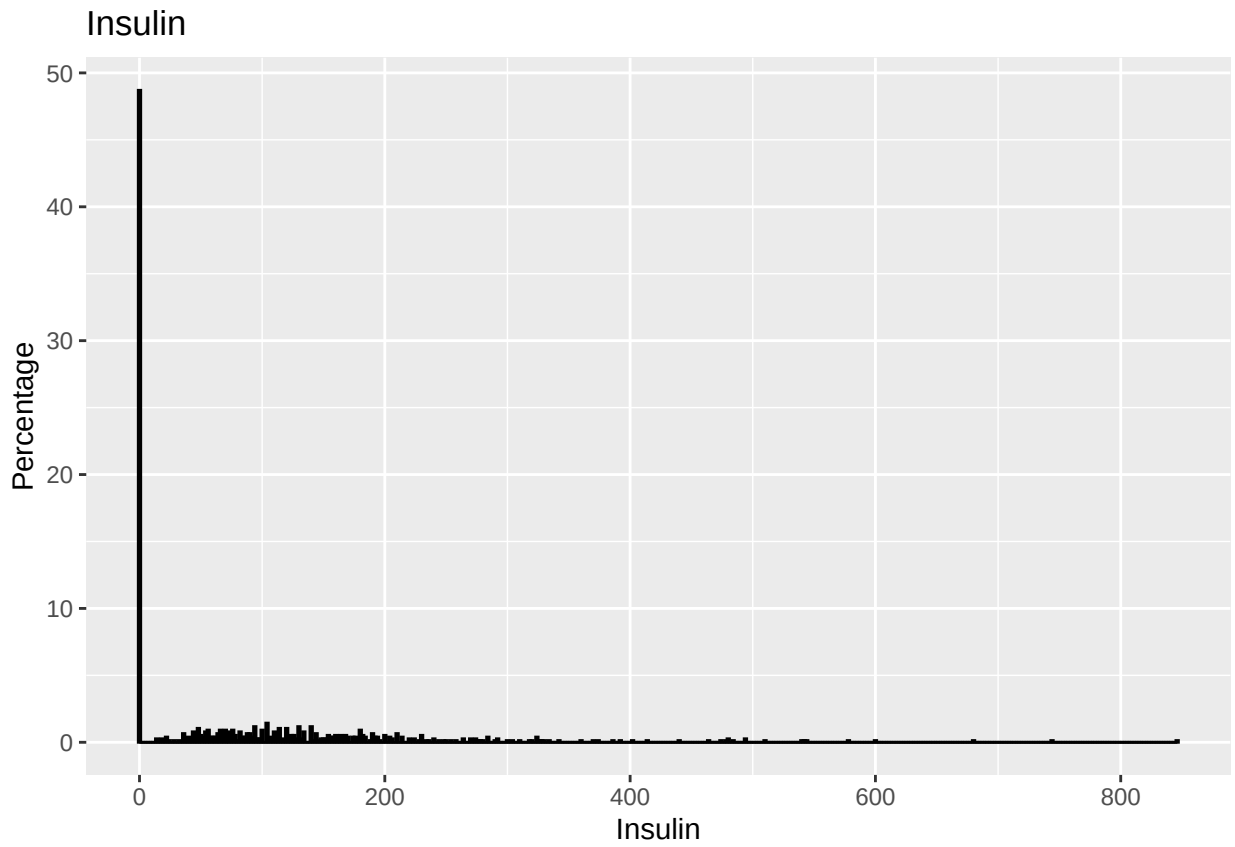

```
#DiabetesPedigreeFunction histogram
diabetes_dataset %>%
  ggplot(aes(x=DiabetesPedigreeFunction))+ggtitle("Diabetes Pedigree Function")+
  geom_histogram(aes(y= 100*(..count..)/sum(..count..)),binwidth = 2, color="black",fill="blue")+ylab("
```



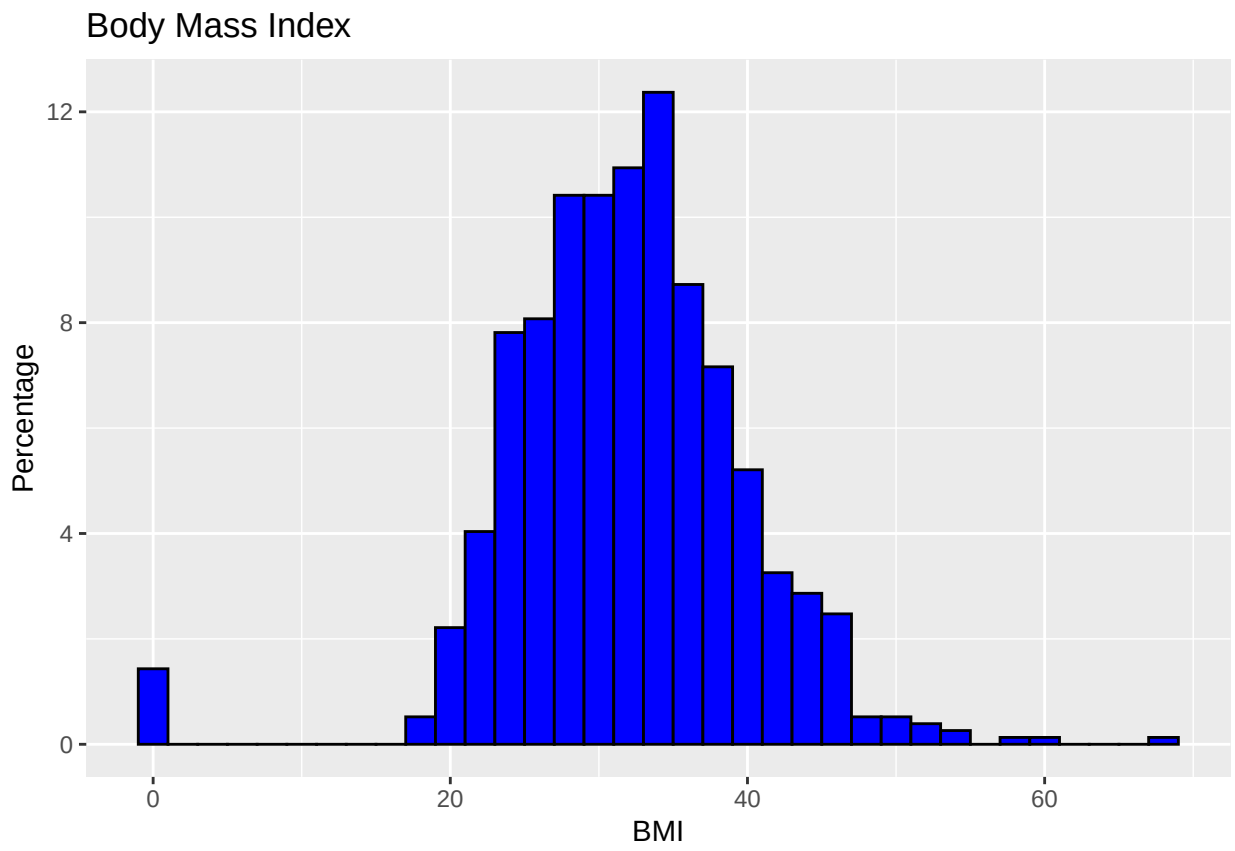

```
#Pregnancies diagram

diabetes_dataset %>%
  ggplot(aes(x=Pregnancies))+ggtitle("Pregnancies")+
  geom_histogram(aes(y= 100*(..count..)/sum(..count..)),binwidth = 2, color="black",fill="pink")+ylab("
```



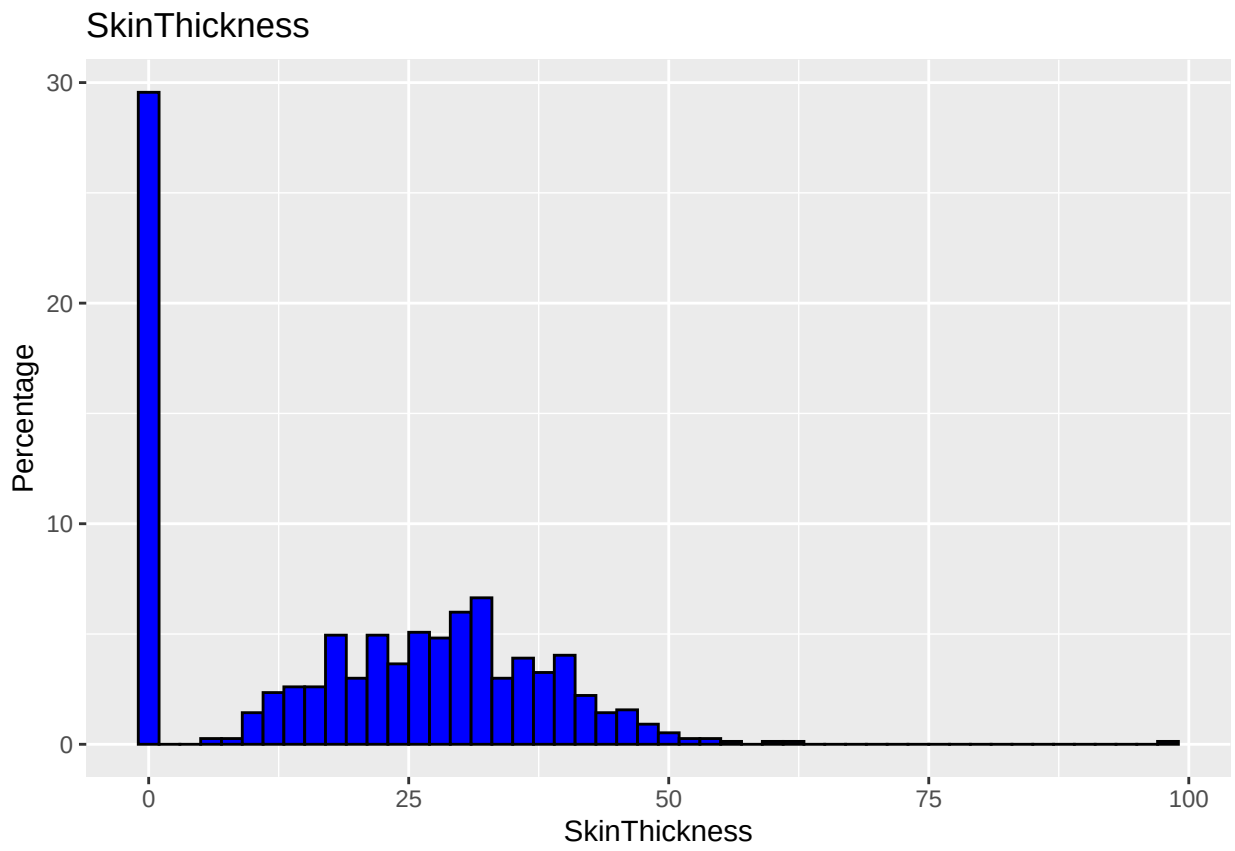

```
#insulin diagram
diabetes_dataset %>%
  ggplot(aes(x=Insulin))+ggtitle("Insulin")+
  geom_histogram(aes(y= 100*(..count..)/sum(..count..)),binwidth = 2, color="black",fill="black")+ylab(
```



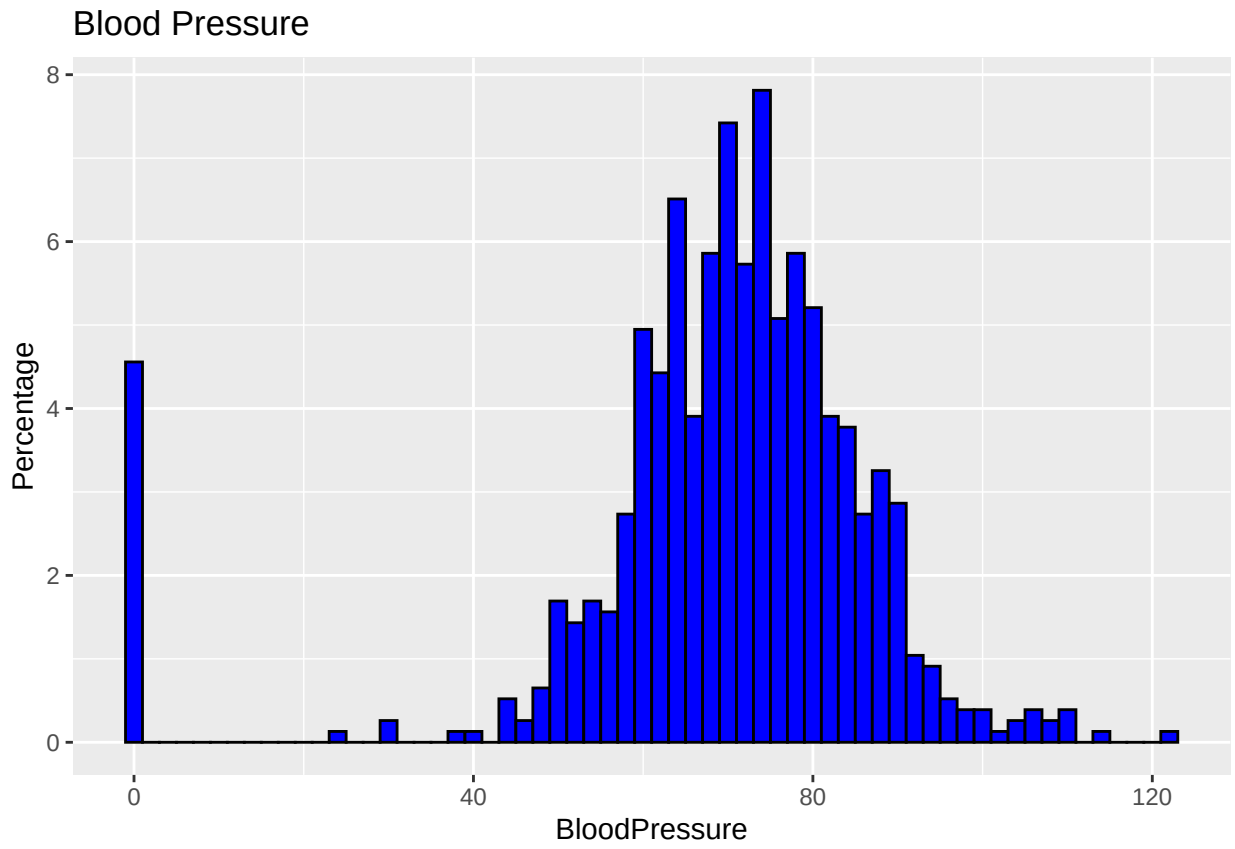

```
#Body Mass Index diagram
diabetes_dataset %>%
  ggplot(aes(x=BMI))+ggtitle("Body Mass Index")+
  geom_histogram(aes(y= 100*(..count..)/sum(..count..)),binwidth = 2, color="black",fill="blue")+ylab("
```



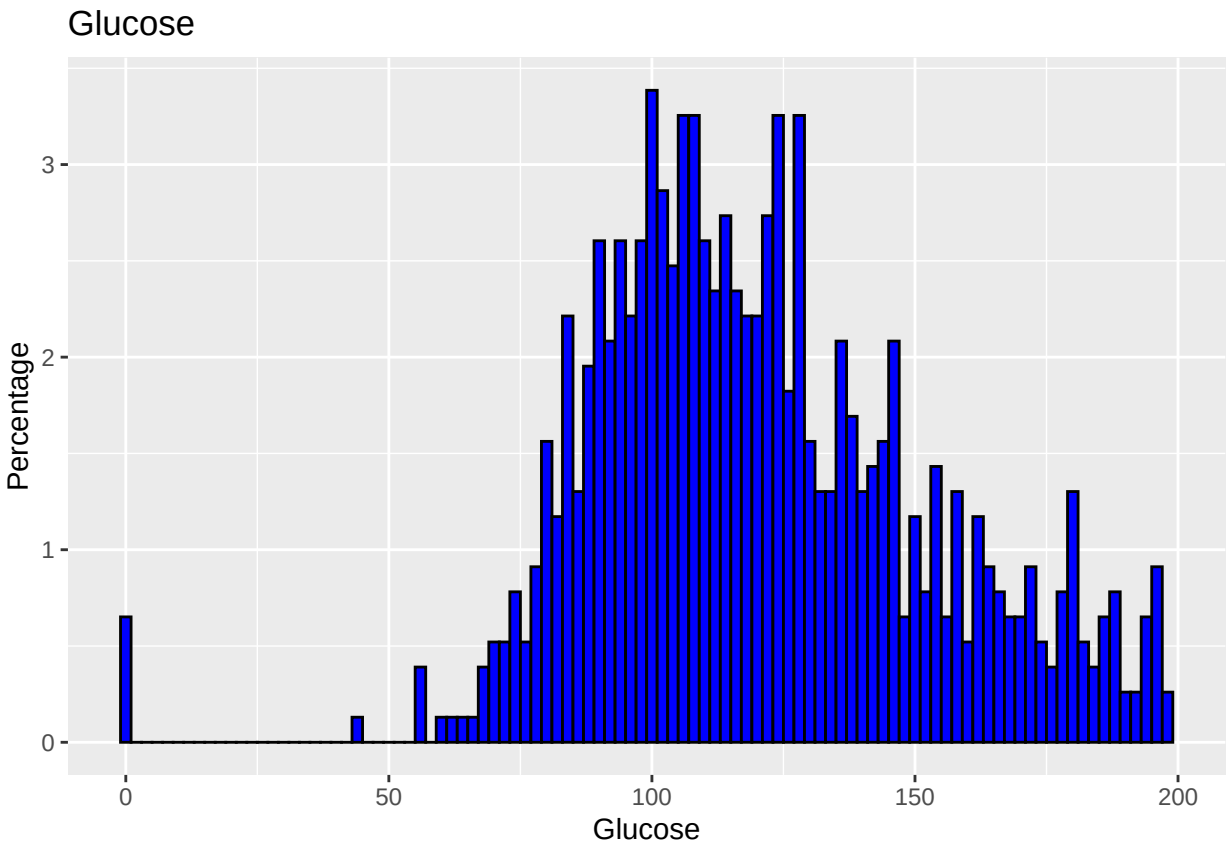

```
#skin thickness
diabetes_dataset %>%
  ggplot(aes(x=SkinThickness))+ggtitle("SkinThickness")+
  geom_histogram(aes(y= 100*(..count..)/sum(..count..)),binwidth = 2, color="black",fill="blue")+ylab("
```



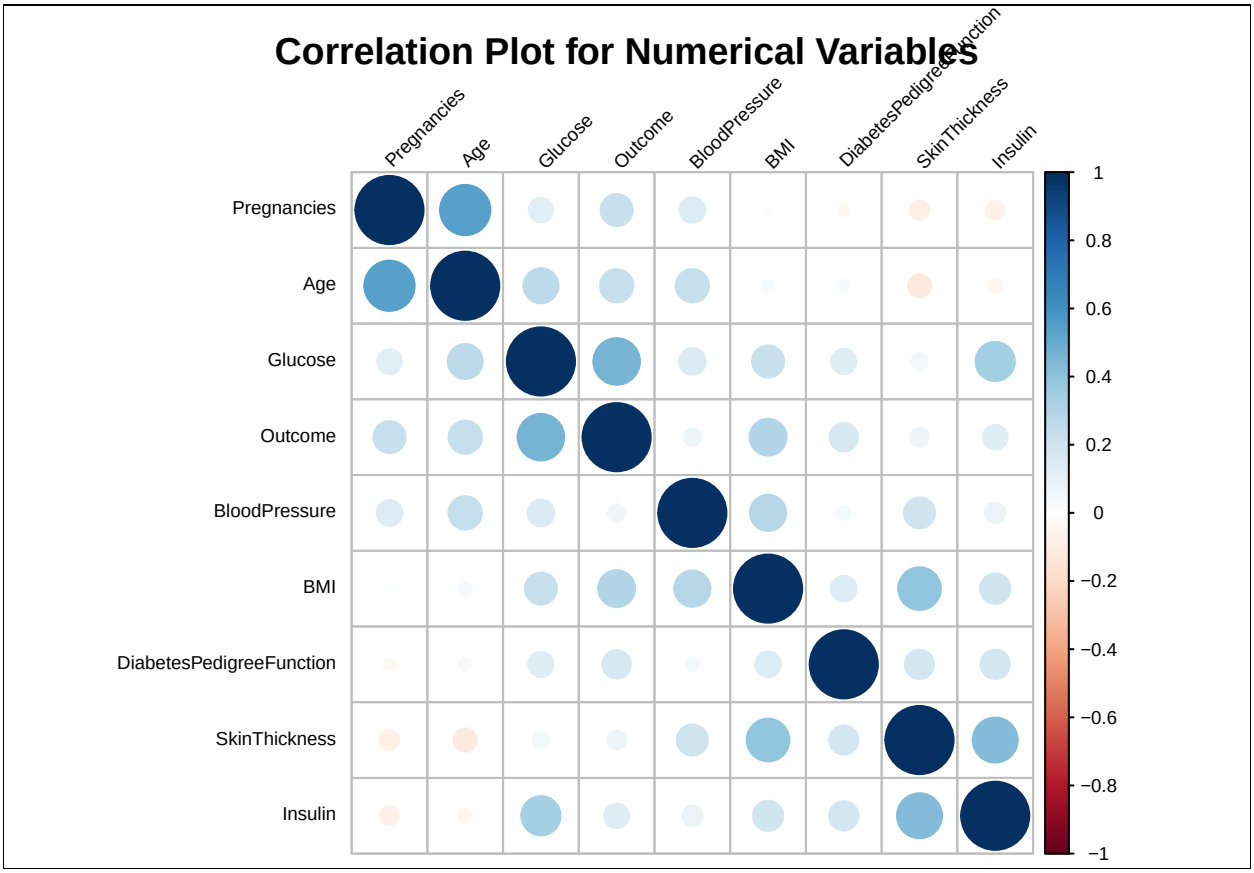

```
#Blood Pressure
diabetes_dataset %>%
  ggplot(aes(x=BloodPressure))+ggtitle("Blood Pressure")+
  geom_histogram(aes(y= 100*(..count..)/sum(..count..)),binwidth = 2, color="black",fill="blue")+ylab("
```



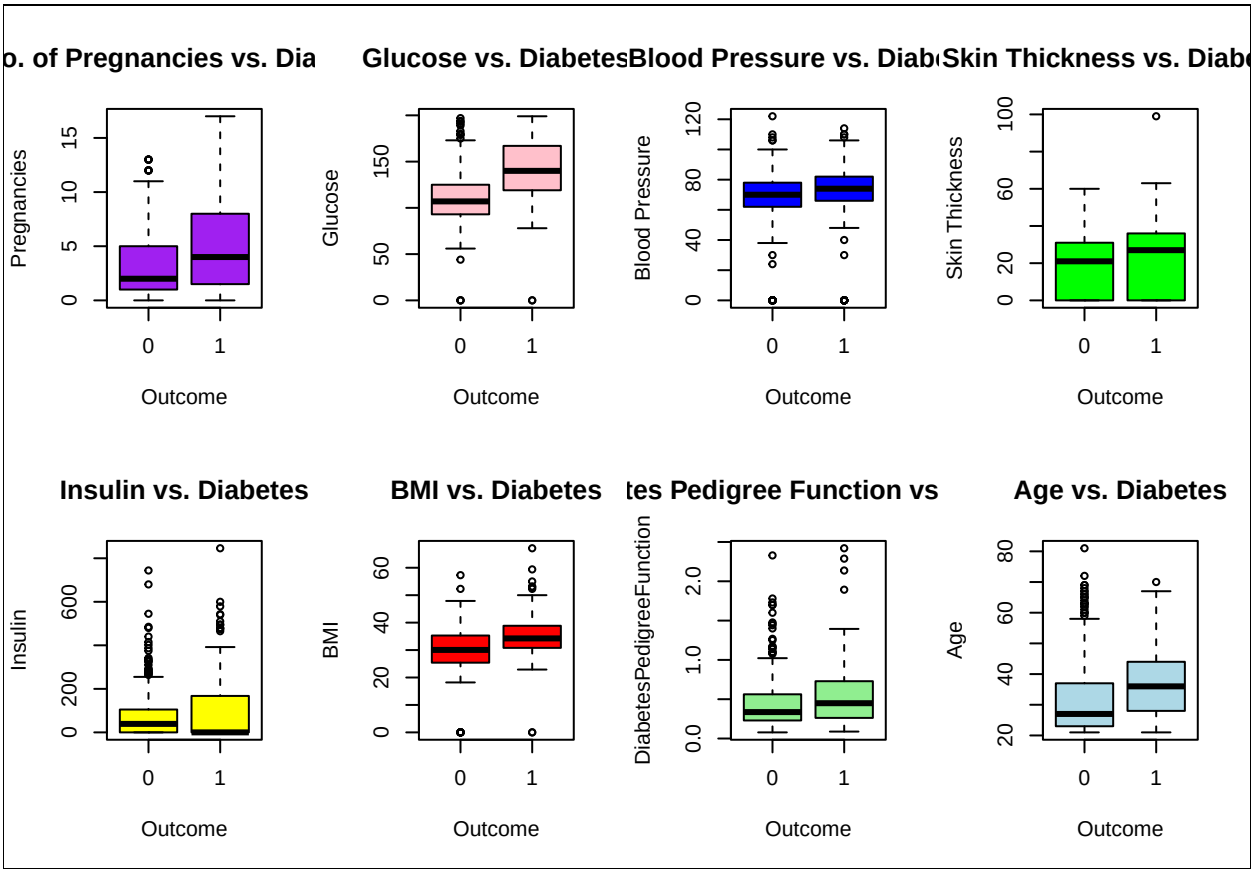

```
#Glucose diagram
diabetes_dataset %>%
  ggplot(aes(x=Glucose))+ggtitle("Glucose")+
  geom_histogram(aes(y= 100*(..count..)/sum(..count..)),binwidth = 2, color="black",fill="blue")+ylab("
```




```
# as all numeric variables have reasonable distribution, I WILL USE THEM for regression
#Correlation between numerical variables
numeric.var <-sapply(diabetes_dataset,is.numeric)
corr.matrix <- cor(diabetes_dataset[,numeric.var])
corrplot(corr.matrix, main="\n\nCorrelation Plot for Numerical Variables", order = "hclust", tl.col = "
box(which = "outer", lty = "solid")
```




```
#now we check correlation between numerical variables and outcome

attach(diabetes_dataset)
par(mfrow=c(2,4))
boxplot(Pregnancies~Outcome, main="No. of Pregnancies vs. Diabetes",
        xlab="Outcome", ylab="Pregnancies",col="purple")
boxplot(Glucose~Outcome, main="Glucose vs. Diabetes",
        xlab="Outcome", ylab="Glucose",col="pink")
boxplot(BloodPressure~Outcome, main="Blood Pressure vs. Diabetes",
        xlab="Outcome", ylab="Blood Pressure",col="blue")
boxplot(SkinThickness~Outcome, main="Skin Thickness vs. Diabetes",
        xlab="Outcome", ylab="Skin Thickness",col="green")
boxplot(Insulin~Outcome, main="Insulin vs. Diabetes",
        xlab="Outcome", ylab="Insulin",col="yellow")
boxplot(BMI~Outcome, main="BMI vs. Diabetes",
        xlab="Outcome", ylab="BMI",col="red")
boxplot(DiabetesPedigreeFunction~Outcome, main="Diabetes Pedigree Function vs. Diabetes", xlab="Outcome
boxplot(Age~Outcome, main="Age vs. Diabetes",
        xlab="Outcome", ylab="Age",col="lightblue")
box(which = "outer", lty = "solid")
```




```
# Blood pressure and skin thickness show little variation with diabetes so they will be escarded
#now we will train our set with regression model
diabetes_dataset$BloodPressure <- NULL
diabetes_dataset$SkinThickness <- NULL
train <- diabetes_dataset[1:540,]
test <- diabetes_dataset[541:768,]
model <-glm(Outcome ~.,family=binomial(link='logit'),data=train)
summary(model)
```

```
##
## Call:
## glm(formula = Outcome ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4366  -0.7741  -0.4312   0.8021   2.7310
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -8.3461752  0.8157916 -10.231  < 2e-16 ***
## Pregnancies               0.1246856  0.0373214   3.341 0.000835 ***
## Glucose                   0.0315778  0.0042497   7.431 1.08e-13 ***
## Insulin                  -0.0013400  0.0009441  -1.419 0.155781
## BMI                       0.0881521  0.0164090   5.372 7.78e-08 ***
## DiabetesPedigreeFunction  0.9642132  0.3430094   2.811 0.004938 **
```

```
## Age                       0.0018904  0.0107225   0.176 0.860053
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 700.47  on 539  degrees of freedom
## Residual deviance: 526.56  on 533  degrees of freedom
## AIC: 540.56
##
## Number of Fisher Scoring iterations: 5
```

```
# From above analysis model we see that the most relevant feauters are Pregnancies, Glucose and BMI bec
#Age and Insulin are rejected because their p-value are not statistically significant
anova(model,test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Outcome
##
## Terms added sequentially (first to last)
##
##
##                           Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                        539     700.47
## Pregnancies                1   26.314       538     674.16 2.901e-07 ***
## Glucose                    1  102.960       537     571.20 < 2.2e-16 ***
## Insulin                    1    0.062       536     571.14  0.803341
## BMI                        1   36.135       535     535.00 1.841e-09 ***
## DiabetesPedigreeFunction   1    8.414       534     526.59  0.003723 **
## Age                        1    0.031       533     526.56  0.860201
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Cross validation model

fitted.results <- predict(model,newdata=test,type='response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)
misClasificError <- mean(fitted.results != test$Outcome)
print(paste('Accuracy',1-misClasificError))
```

```
## [1] "Accuracy 0.789473684210526"
```

```
(conf_matrix_logi <-table(fitted.results,test$Outcome))
```

```
##
## fitted.results   0   1
##              0 136  34
##              1  14  44
```

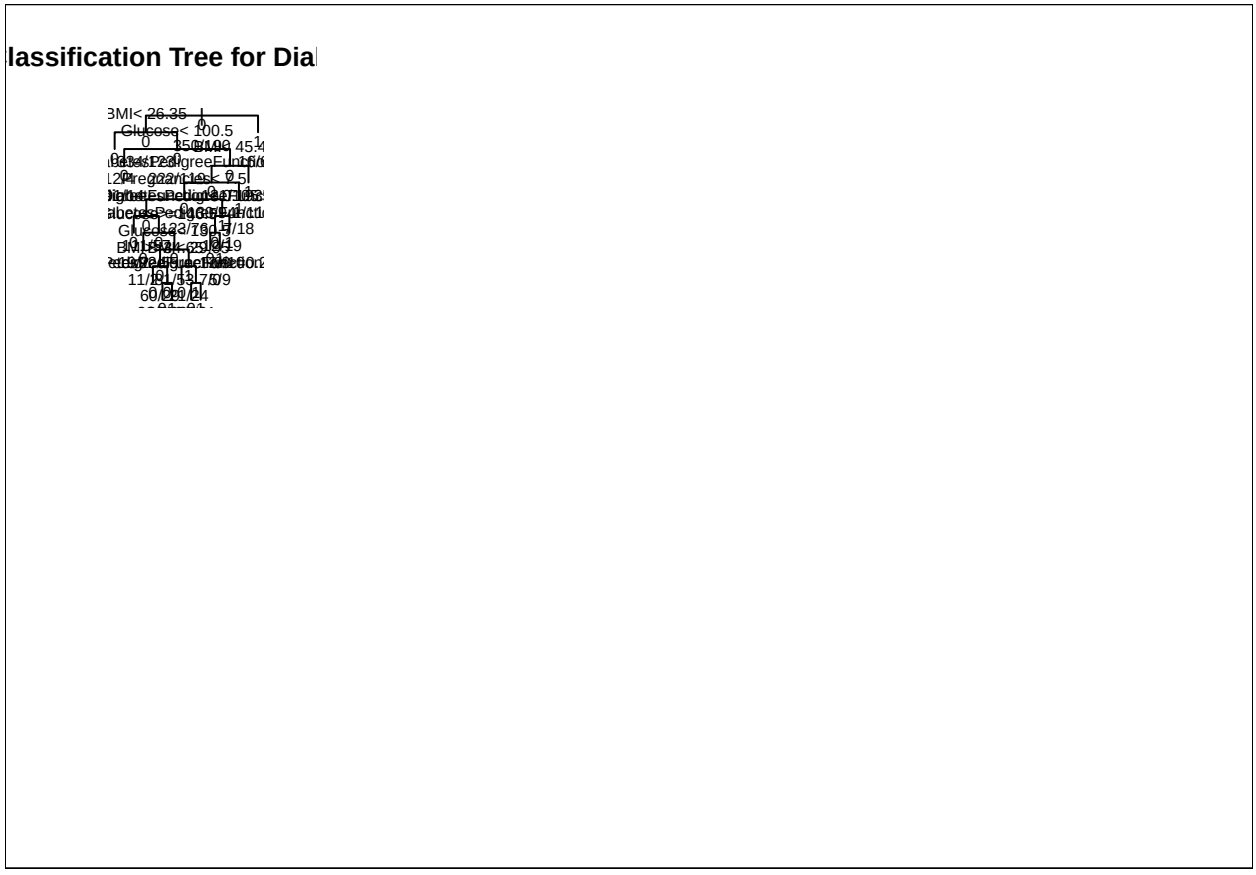
```
#decision tree
library(rpart)
model2 <- rpart(Outcome ~ Pregnancies + Glucose + BMI + DiabetesPedigreeFunction, data=train,method="cla
plot(model2, uniform=TRUE,
     main="Classification Tree for Diabetes")
```

```
text(model2, use.n=TRUE, all=TRUE, cex=.8)
box(which = "outer", lty = "solid")

#if the BMI is less than 45.4 and diabetes pedigree functions is less than 0.8745, the person is likely

#confusion table and accuracy

tree_prediction <- predict(model2,test, type = 'class')
(conf_matrix_tree <-table(tree_prediction,test$Outcome))
```

```
##
## tree_prediction   0   1
##               0 121  29
##               1  29  49
```

```
mean(tree_prediction == test$Outcome)
```

```
## [1] 0.745614
```



##Results

from the above analysis, I found that if the person's BMI is less than 45.4 and his diabetes pedigree function is less than 0.8745 then the person is likely to have diabete

##Conclusion The objective of this project was to train my data set so that I may be able to predict if the person may have diabetes basing on different models and I was able to achieve the goal. Further analysis may be done on this data set but as it was my first project i limited on two and I will continue working on other different models to reach more accurate predictions. Logistic Regression performed better with 79%

accuracy compared to decision tree with 74%.