A Complete R Approach for Analyzing Health Care Data (Machine Learning Algorithms, GLM).

This article focuses on a thorough examination of diabetes data using the base model, which includes the following analysis:

1. Data investigation (Data distribution inferences, Univariate Data analysis, Two-sample t-test).
2. Analysis of data correlation.

3. Feature Selection (using Logistic regression).

4. Detection of Outliers (using principal component graph).

5. Simple Parameter Tweaking (CV, complexity parameter).

6. Data modeling.

Simple GLM (With all Features and eliminating a few features based on AIC).

Logistic Regression.

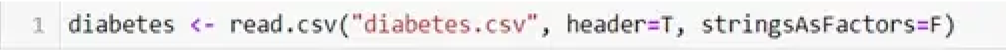
Decision Tree.

Naïve Bayes.

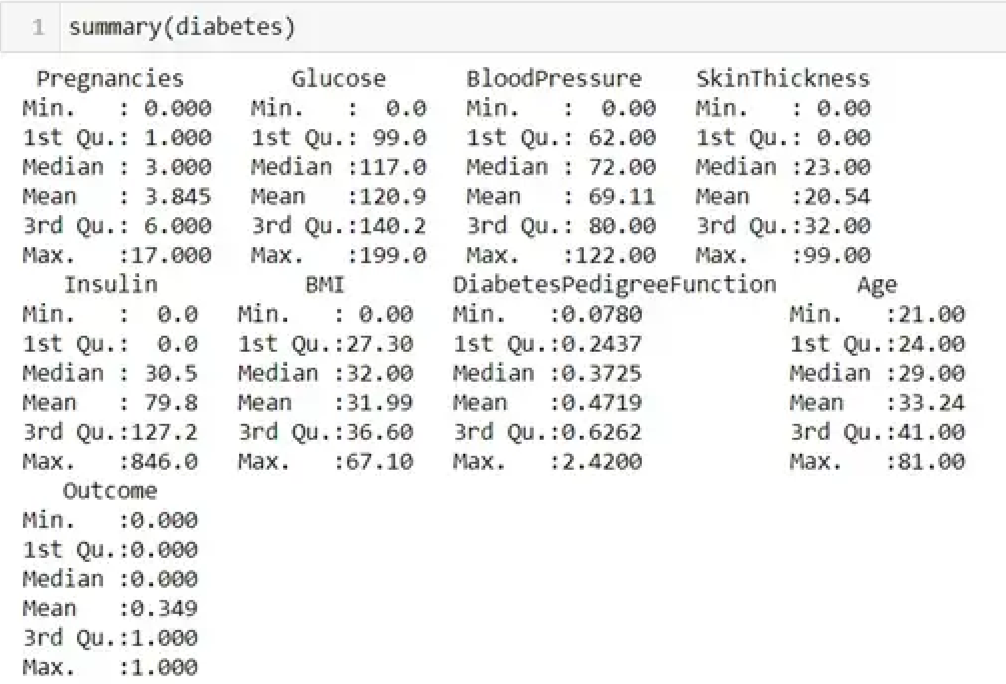
Ref: https://rb.gy/xej8wd

Basic EDA

We can download details from. https://www.kaggle.com/uciml/pima-indians-diabetes-database.



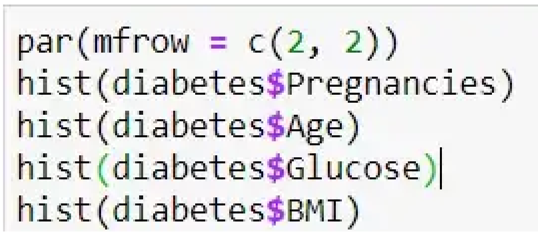
```
diabetes <- read.csv("diabetes.csv", header=T, stringsAsFactors=F)
```



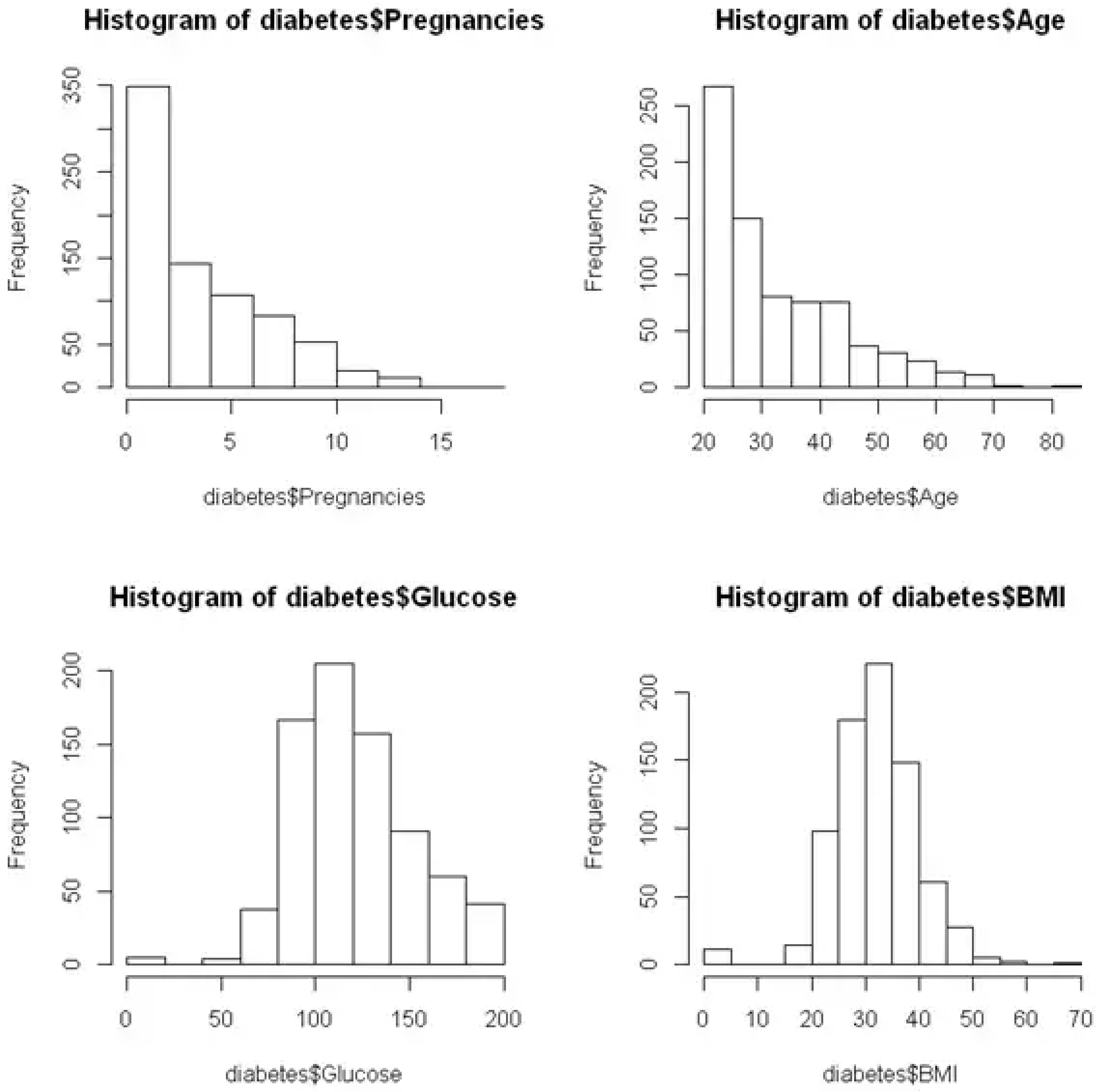
```
summary(diabetes)
```

```
  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
```

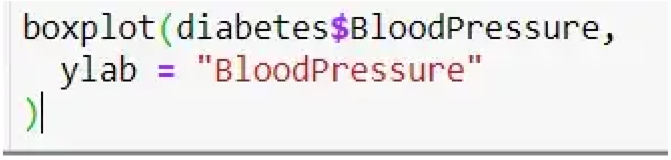
Univariate analysis



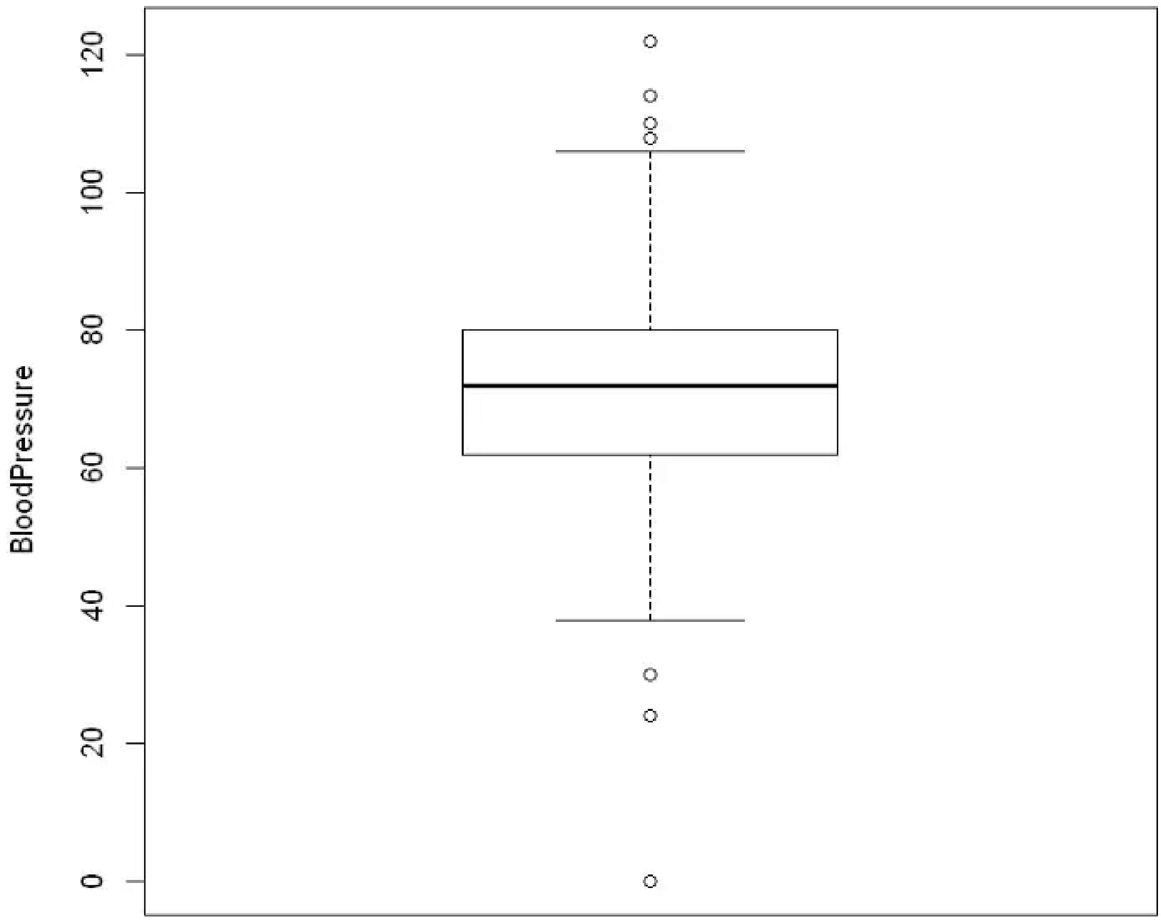
```
par(mfrow = c(2, 2))
hist(diabetes$Pregnancies)
hist(diabetes$Age)
hist(diabetes$Glucose)
hist(diabetes$BMI)
```



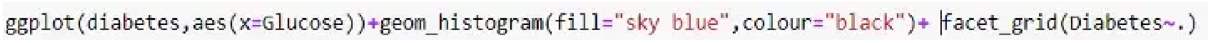
Age and the number of pregnancies are not distributed normally in these distribution graphs, as would be predicted given that the population at large shouldn't be distributed normally either. BMI and glucose levels both exhibit a normal distribution.



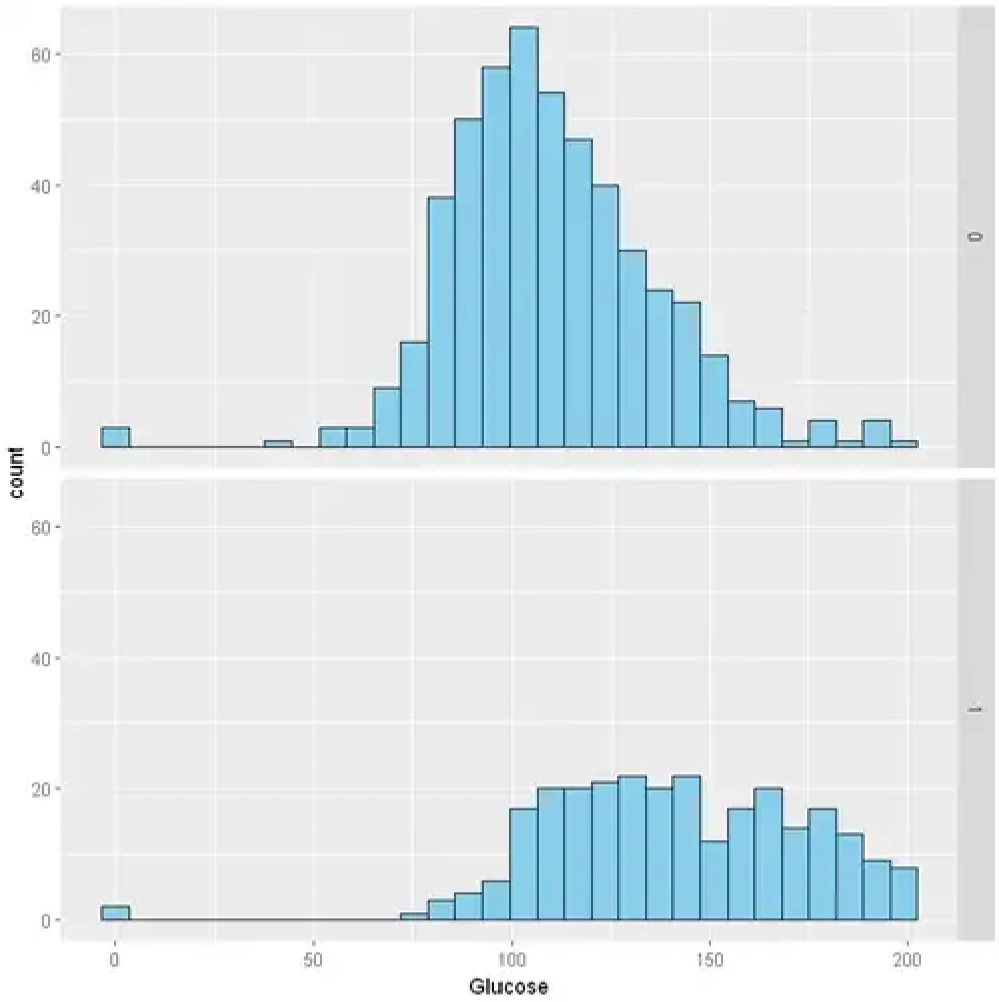
```
boxplot(diabetes$BloodPressure,
  ylab = "BloodPressure"
)
```



Impact of Glucose on Diabetes



```
ggplot(diabetes,aes(x=Glucose))+geom_histogram(fill="sky blue",colour="black")+ facet_grid(Diabetes~.)
```



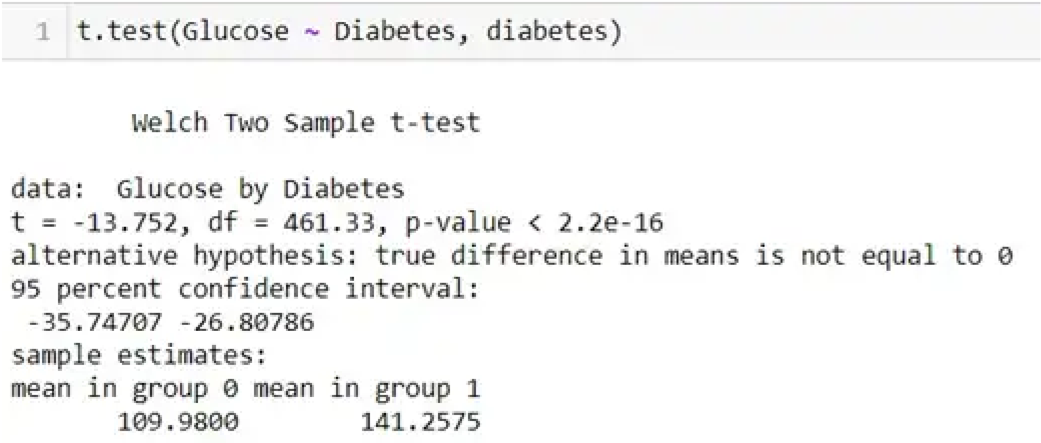
Develops a hypothesis to evaluate the average glucose level difference between the positive and negative groups.

Conditions.

People are independent of one another.

Although the sample size is greater than 30, the distributions in this instance are skewed.

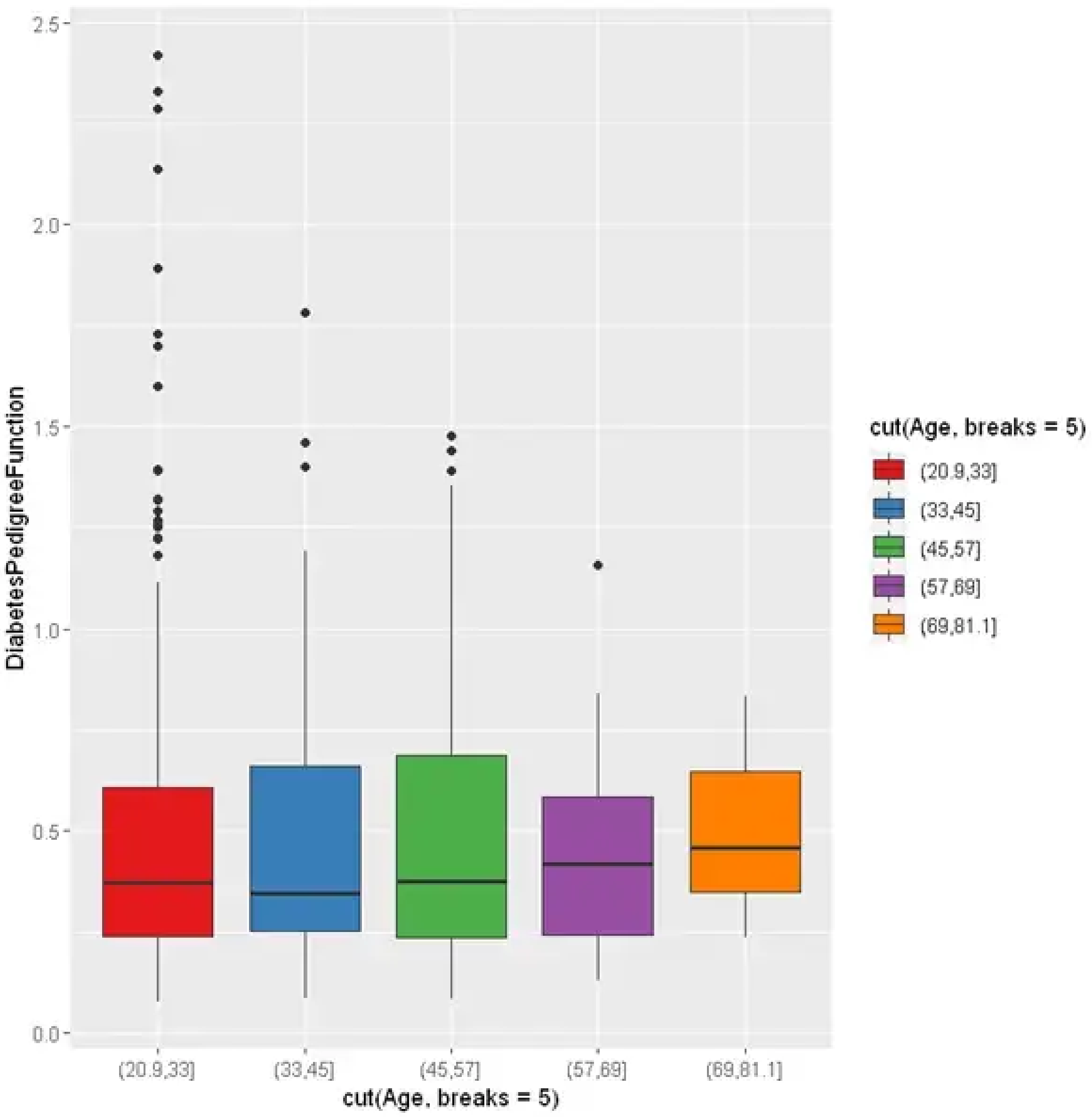
The sample size is less than 10% of the population, and both groups are independent of one another.



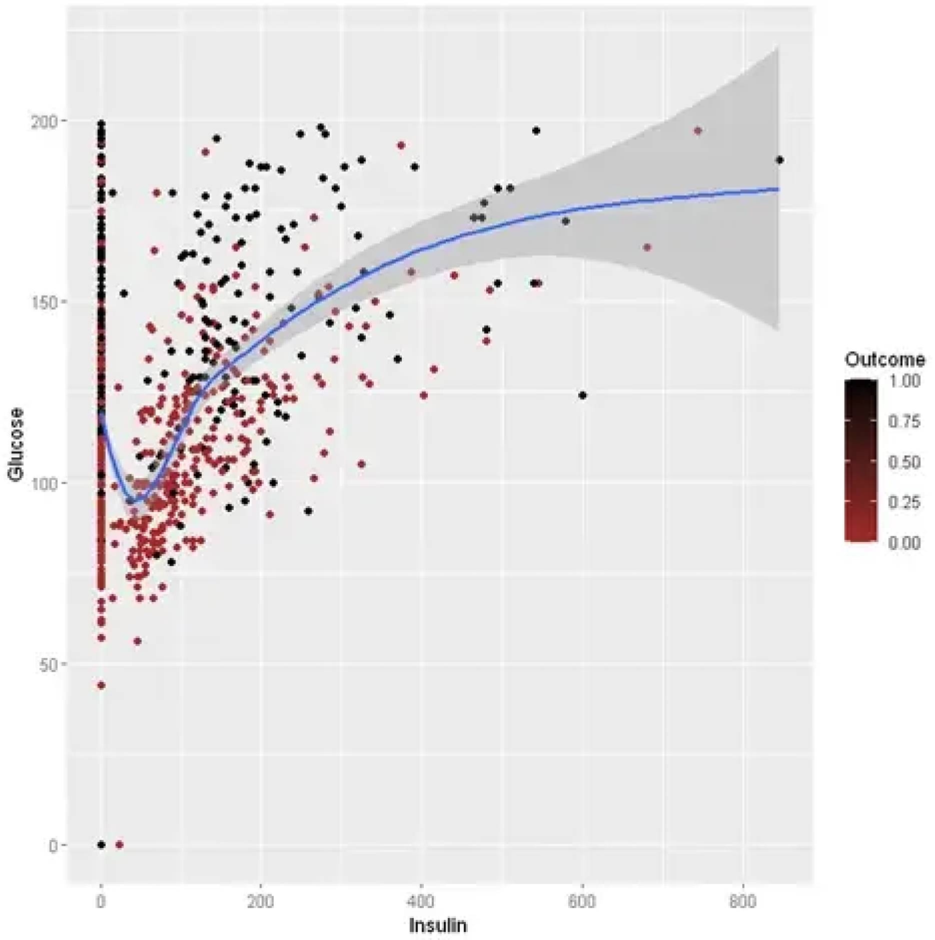
```r
t.test(Glucose ~ Diabetes, diabetes)
```

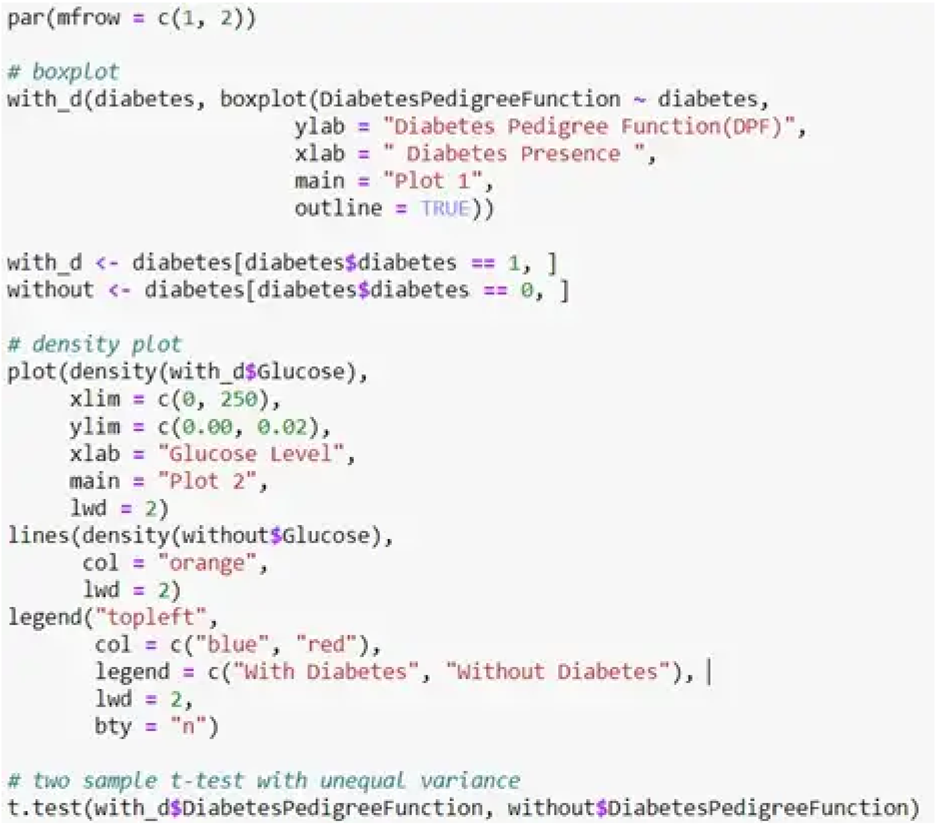
```
        Welch Two Sample t-test

data:  Glucose by Diabetes
t = -13.752, df = 461.33, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -35.74707 -26.80786
sample estimates:
mean in group 0 mean in group 1
       109.9800        141.2575
```

Since the p-value for the alternate hypothesis is less than the threshold value of 0.05, the null hypothesis is rejected. We can state with 95% certainty that those without diabetes have average blood glucose levels that are similar to those of people with diabetes.



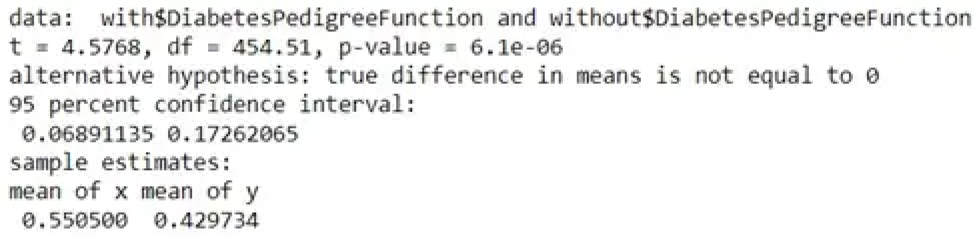
Insulin Vs Glucose based on Outcome as diabetes.



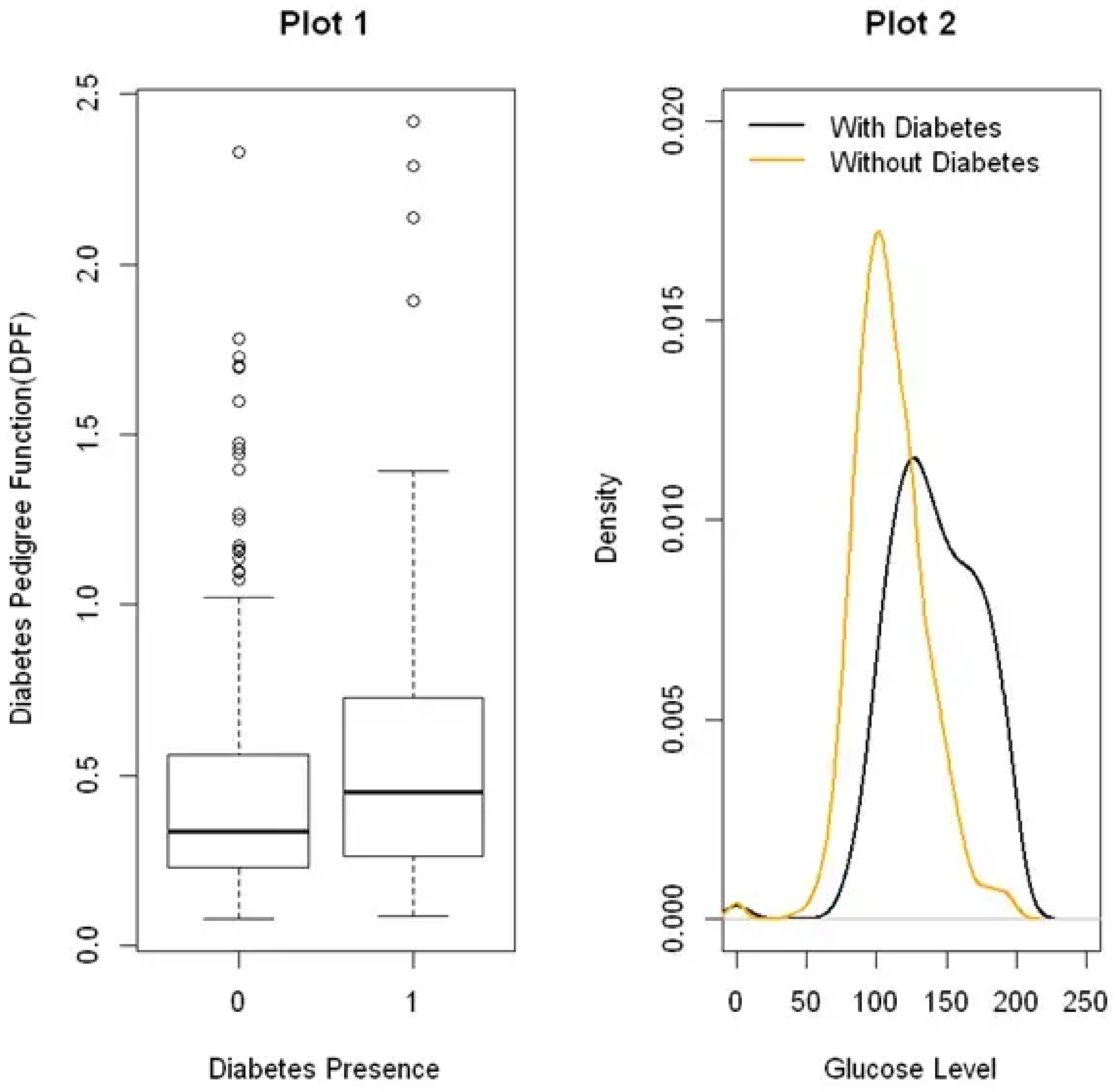


```
par(mfrow = c(1, 2))

# boxplot
with_d(diabetes, boxplot(DiabetesPedigreeFunction ~ diabetes,
                         ylab = "Diabetes Pedigree Function(DPF)",
                         xlab = " Diabetes Presence ",
                         main = "Plot 1",
                         outline = TRUE))

with_d <- diabetes[diabetes$diabetes == 1, ]
without <- diabetes[diabetes$diabetes == 0, ]

# density plot
plot(density(with_d$Glucose),
     xlim = c(0, 250),
     ylim = c(0.00, 0.02),
     xlab = "Glucose Level",
     main = "Plot 2",
     lwd = 2)
lines(density(without$Glucose),
      col = "orange",
      lwd = 2)
legend("topleft",
       col = c("blue", "red"),
       legend = c("With Diabetes", "Without Diabetes"),
       lwd = 2,
       bty = "n")

# two sample t-test with unequal variance
t.test(with_d$DiabetesPedigreeFunction, without$DiabetesPedigreeFunction)
```

Welch Two Sample t-test



```
data:  with$DiabetesPedigreeFunction and without$DiabetesPedigreeFunction
t = 4.5768, df = 454.51, p-value = 6.1e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.06891135 0.17262065
sample estimates:
mean of x mean of y
 0.550500  0.429734
```

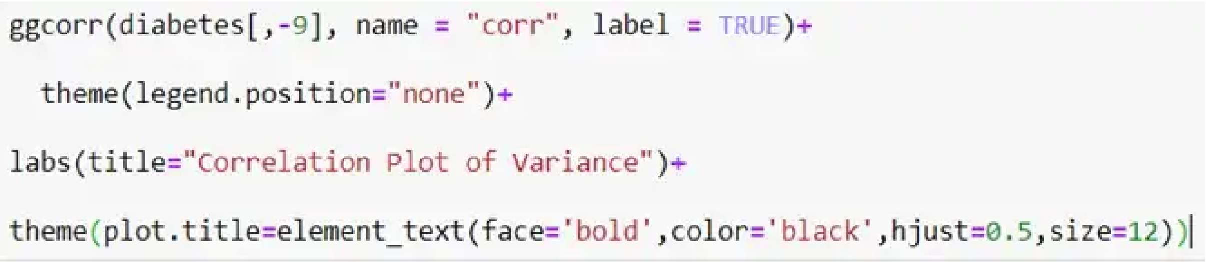


For those without diabetes, the distribution is moved to the left from Plot 2.

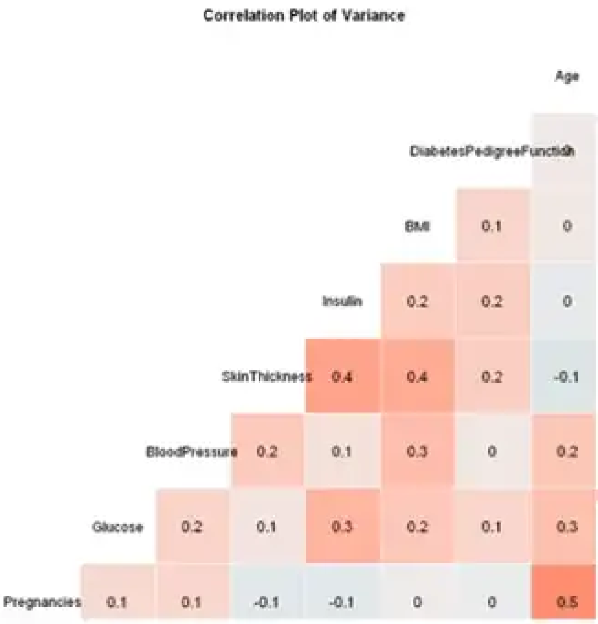
This shows that individuals without diabetes typically have lower blood glucose levels.

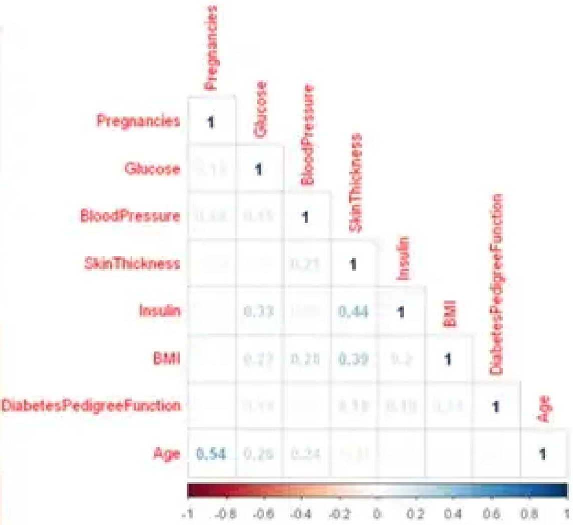
Relationships between the various variables.

All-column scatter matrix.



```
ggcorr(diabetes[,-9], name = "corr", label = TRUE)+

  theme(legend.position="none")+

labs(title="Correlation Plot of Variance")+

theme(plot.title=element_text(face='bold',color='black',hjust=0.5,size=12))
```

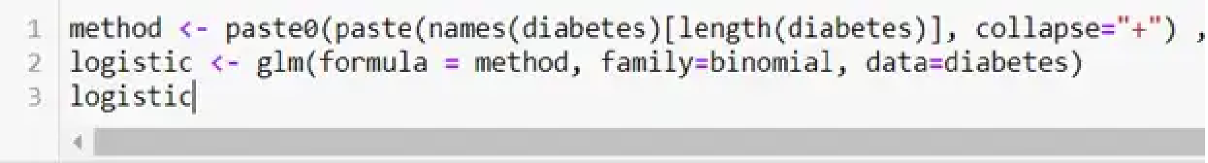




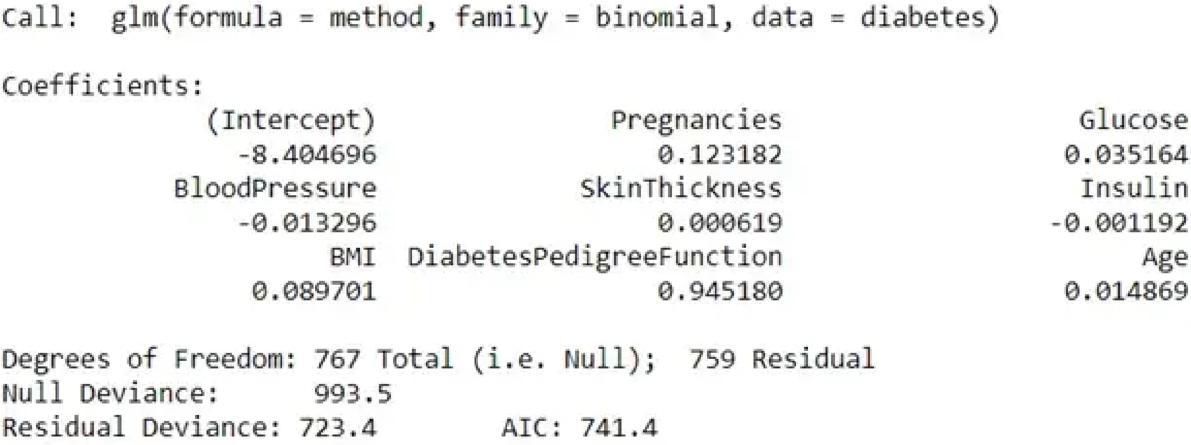
Higher link exists between pregnancy, age, insulin, and skin thickness.

Applying a logistic regression model to evaluate the significance of predictors.

Fitting a GLM (General Linear Model) with the 'probit' link function.
The distribution of the target variable 'diabetes' is estimated to be binomial.
This implementation makes no data-specific assumptions.



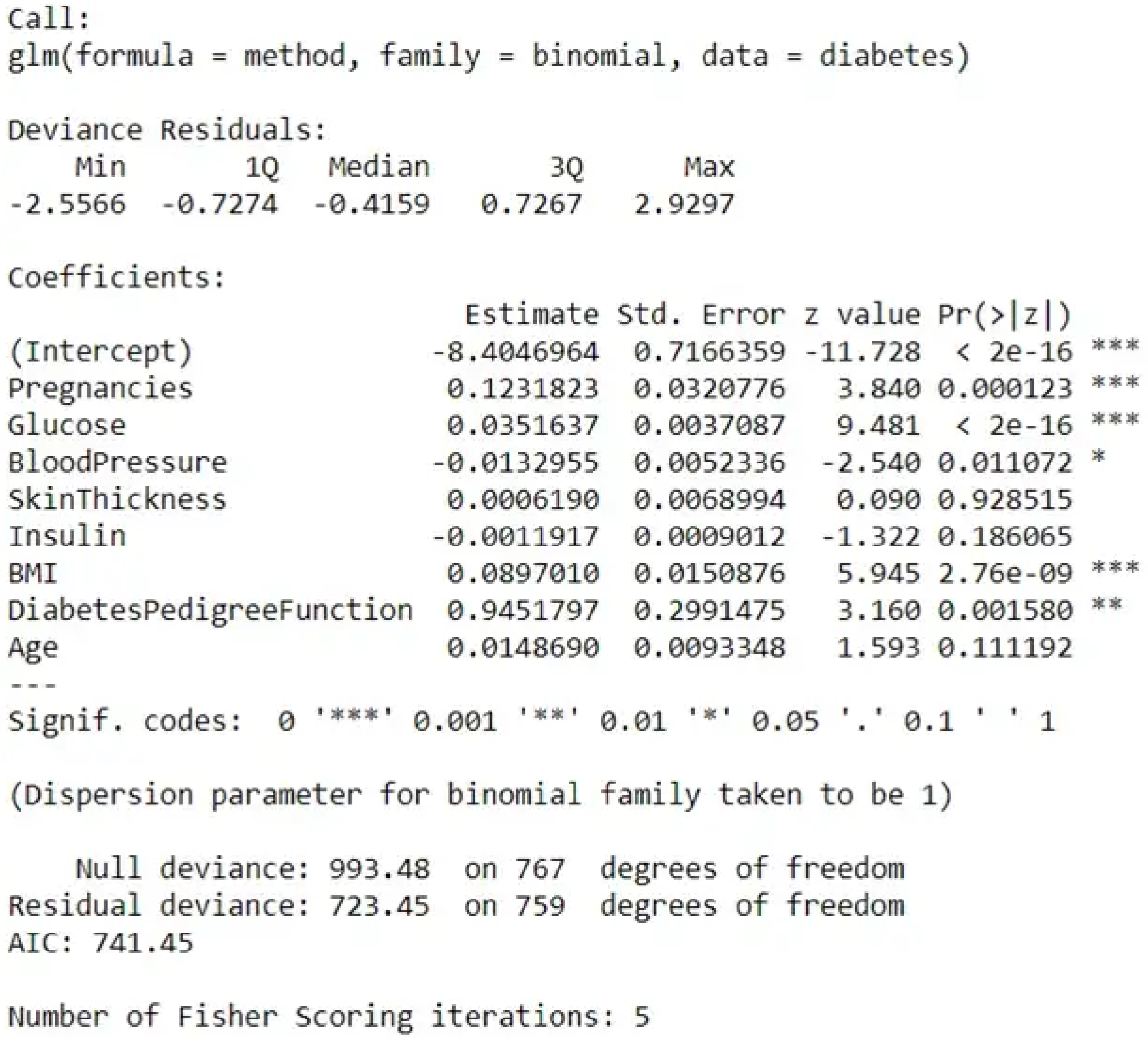
```
method <- paste0(paste(names(diabetes)[length(diabetes)], collapse="+") ,
logistic <- glm(formula = method, family=binomial, data=diabetes)
logistic
```



```
Call:  glm(formula = method, family = binomial, data = diabetes)

Coefficients:
             (Intercept)               Pregnancies                   Glucose  
               -8.404696                  0.123182                  0.035164  
           BloodPressure             SkinThickness                   Insulin  
               -0.013296                  0.000619                 -0.001192  
                     BMI  DiabetesPedigreeFunction                       Age  
                0.089701                  0.945180                  0.014869  

Degrees of Freedom: 767 Total (i.e. Null);  759 Residual
Null Deviance:      993.5 
Residual Deviance: 723.4        AIC: 741.4
```

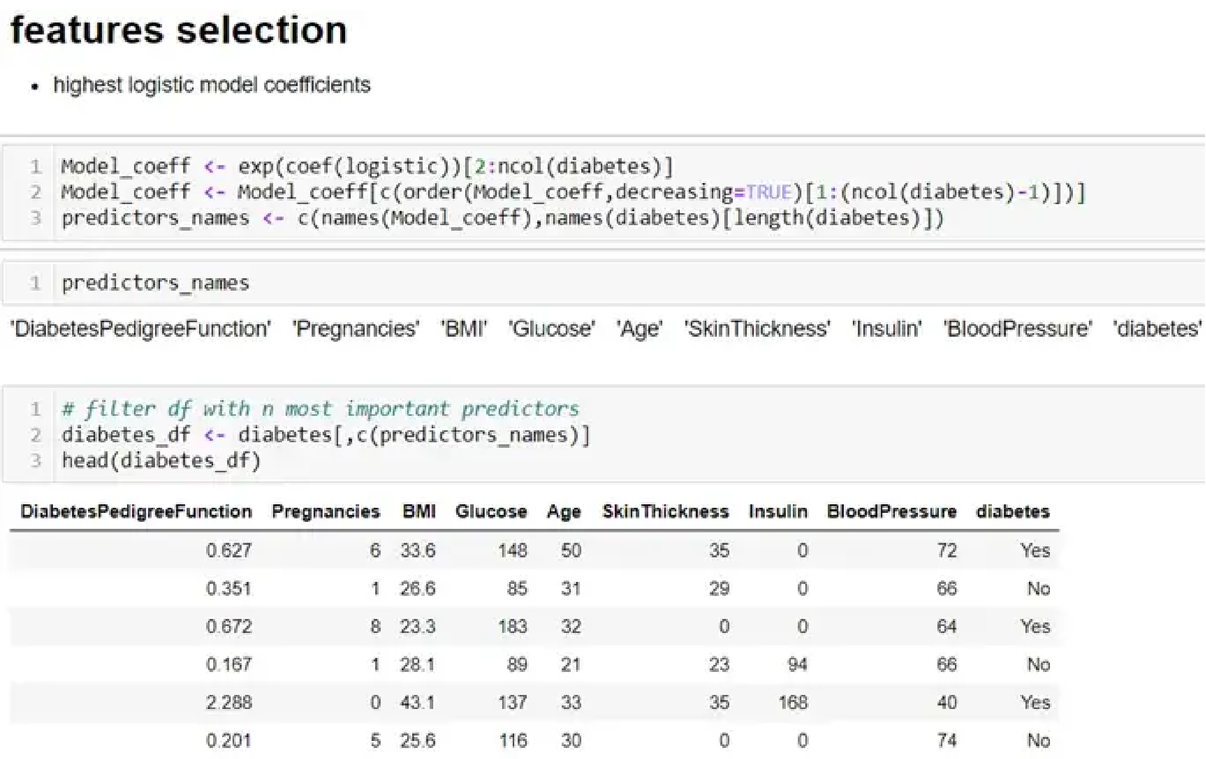
In the GLM model, the most significant predictors are filtered out.

- The extraction of the N most significant GLM coefficients.



```
Call:
glm(formula = method, family = binomial, data = diabetes)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5566  -0.7274  -0.4159   0.7267   2.9297

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)              -8.4046964  0.7166359 -11.728  < 2e-16 ***
Pregnancies               0.1231823  0.0320776   3.840 0.000123 ***
Glucose                   0.0351637  0.0037087   9.481  < 2e-16 ***
BloodPressure            -0.0132955  0.0052336  -2.540 0.011072 *
SkinThickness             0.0006190  0.0068994   0.090 0.928515
Insulin                  -0.0011917  0.0009012  -1.322 0.186065
BMI                       0.0897010  0.0150876   5.945 2.76e-09 ***
DiabetesPedigreeFunction  0.9451797  0.2991475   3.160 0.001580 **
Age                       0.0148690  0.0093348   1.593 0.111192
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 993.48  on 767  degrees of freedom
Residual deviance: 723.45  on 759  degrees of freedom
AIC: 741.45

Number of Fisher Scoring iterations: 5
```

- Using Logistic Regression:



features selection

- highest logistic model coefficients

```
Model_coeff <- exp(coef(logistic))[2:ncol(diabetes)]
Model_coeff <- Model_coeff[c(order(Model_coeff,decreasing=TRUE)[1:(ncol(diabetes)-1)])]
predictors_names <- c(names(Model_coeff),names(diabetes)[length(diabetes)])
```

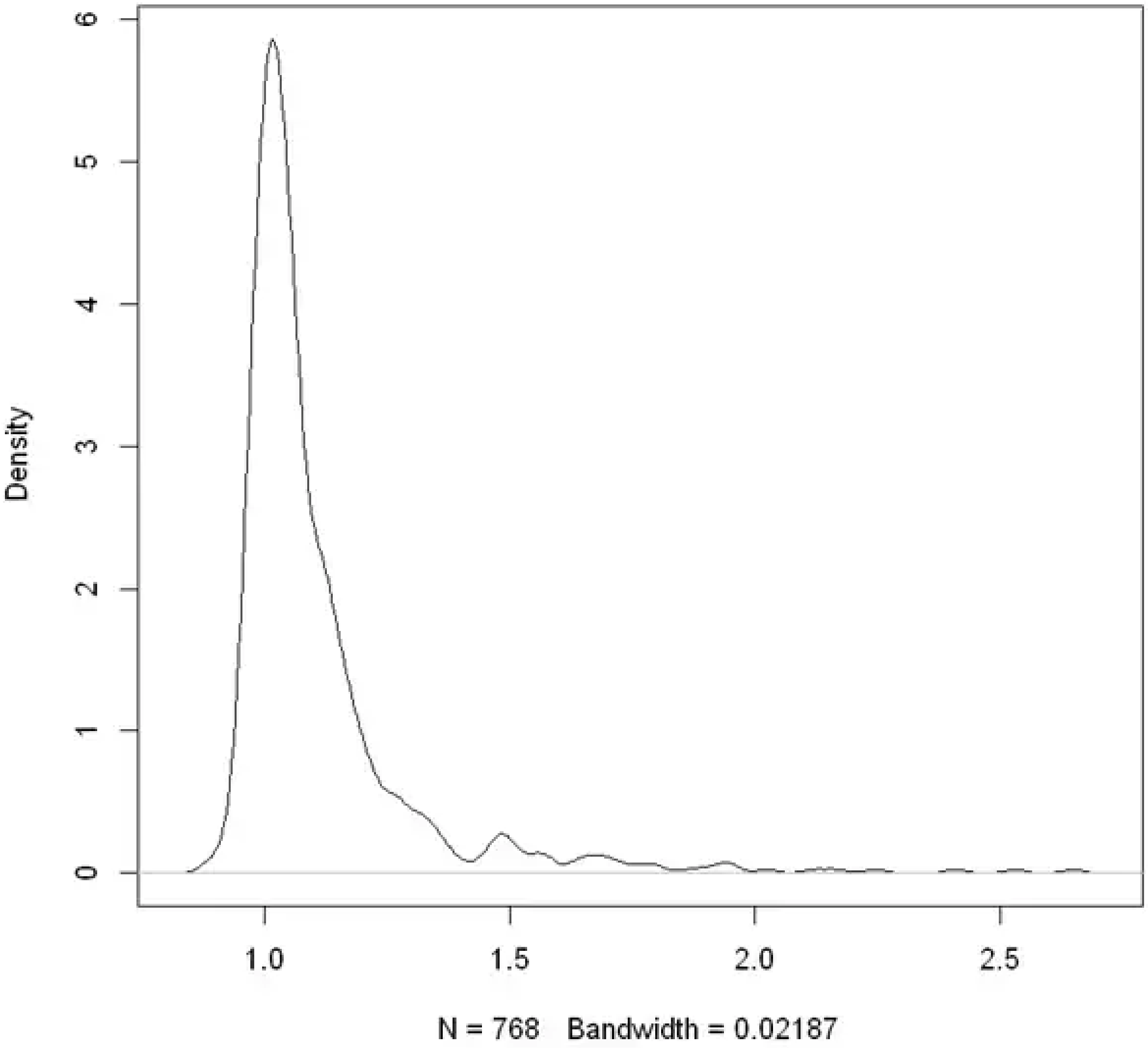
```
predictors_names
```

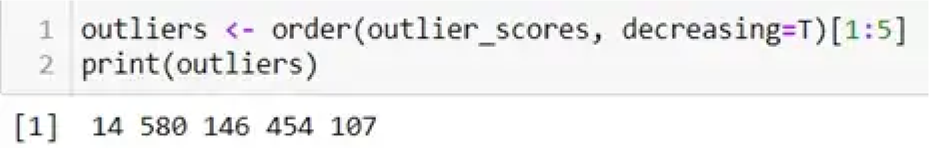
```
'DiabetesPedigreeFunction'   'Pregnancies'   'BMI'   'Glucose'   'Age'   'SkinThickness'   'Insulin'   'BloodPressure'   'diabetes'
```

```
# filter df with n most important predictors
diabetes_df <- diabetes[,c(predictors_names)]
head(diabetes_df)
```

DiabetesPedigreeFunction	Pregnancies	BMI	Glucose	Age	SkinThickness	Insulin	BloodPressure	diabetes
0.627	6	33.6	148	50	35	0	72	Yes
0.351	1	26.6	85	31	29	0	66	No
0.672	8	23.3	183	32	0	0	64	Yes
0.167	1	28.1	89	21	23	94	66	No
2.288	0	43.1	137	33	35	168	40	Yes
0.201	5	25.6	116	30	0	0	74	No

Detection of outliers

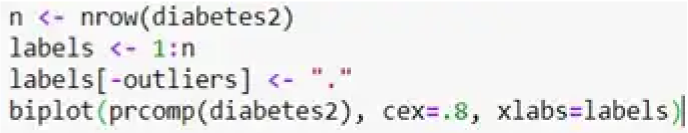




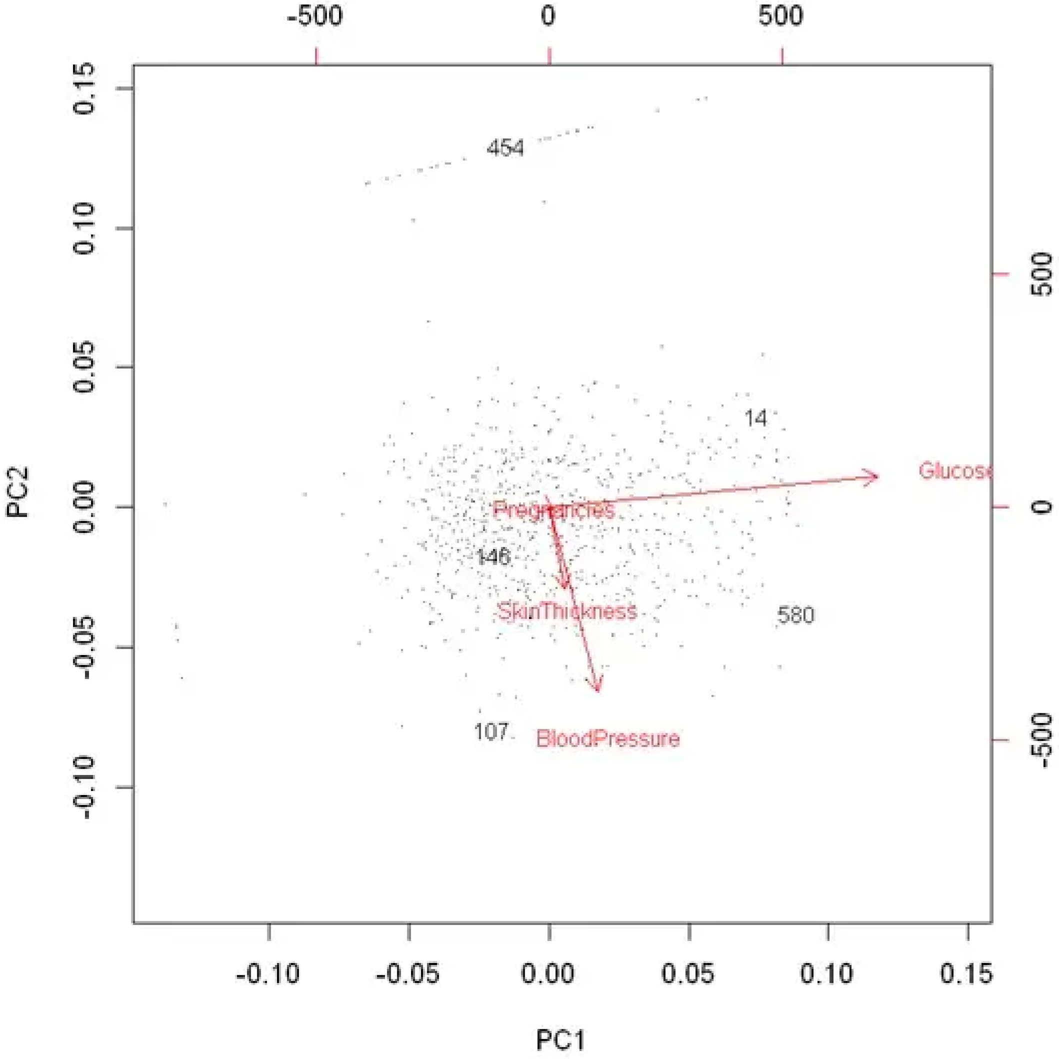
```
outliers <- order(outlier_scores, decreasing=T)[1:5]
print(outliers)
```

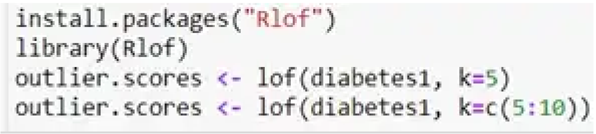
```
[1]  14 580 146 454 107
```

The five outliers found in the output correspond to the diabetes1 data's row numbers, which were taken from the diabetes data set.

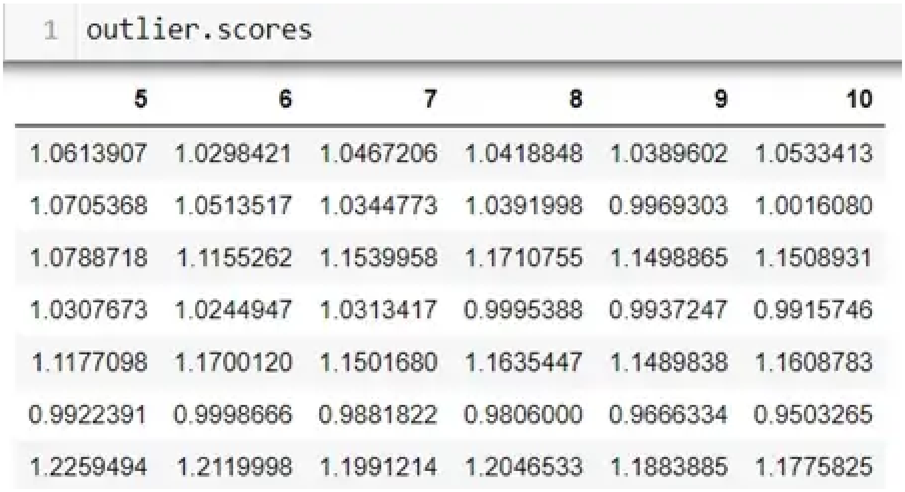


```
n <- nrow(diabetes2)
labels <- 1:n
labels[-outliers] <- "."
biplot(prcomp(diabetes2), cex=.8, xlabs=labels)
```





```
install.packages("Rlof")
library(Rlof)
outlier.scores <- lof(diabetes1, k=5)
outlier.scores <- lof(diabetes1, k=c(5:10))
```

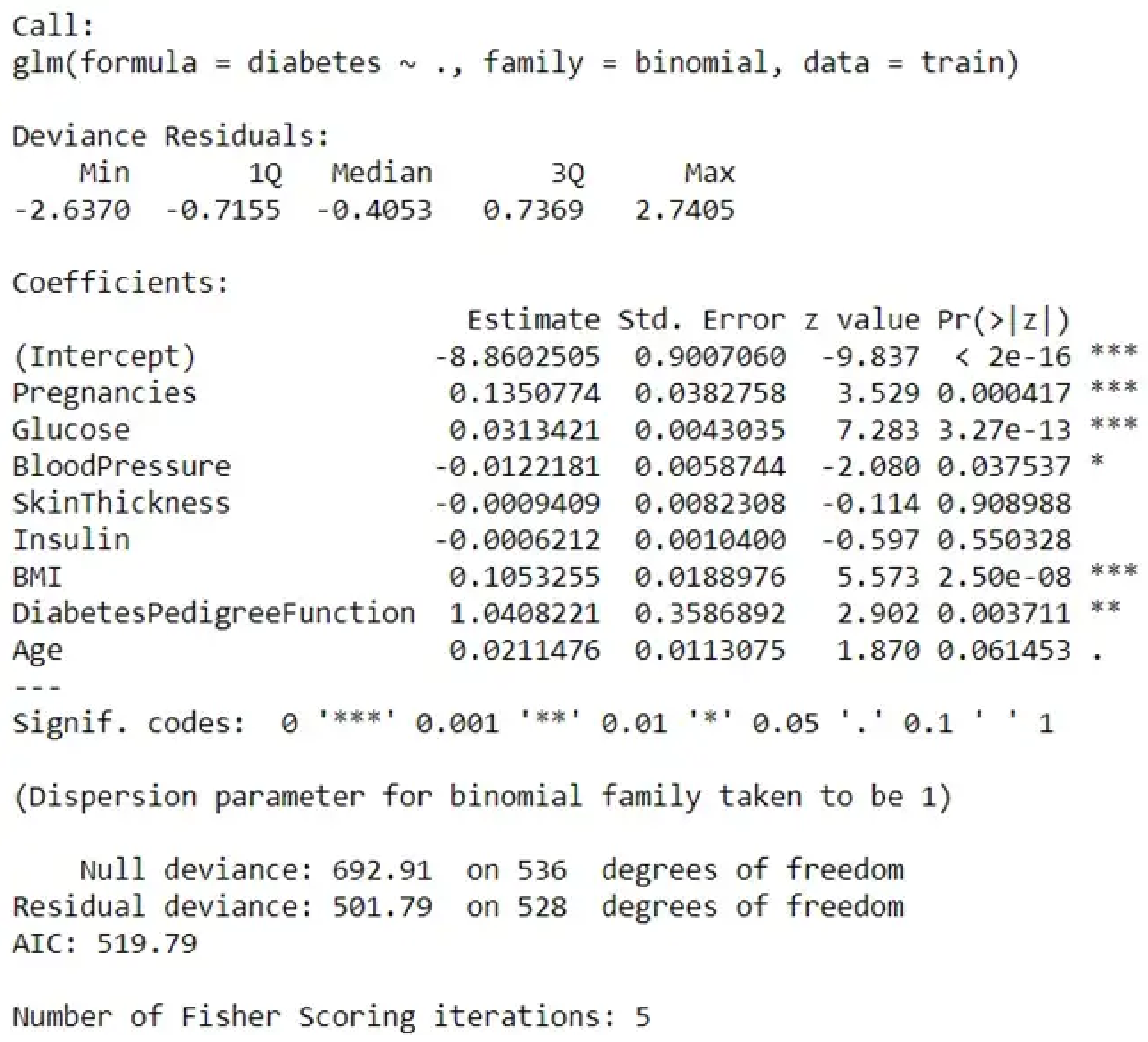


```
outlier.scores
```

5	6	7	8	9	10
1.0613907	1.0298421	1.0467206	1.0418848	1.0389602	1.0533413
1.0705368	1.0513517	1.0344773	1.0391998	0.9969303	1.0016080
1.0788718	1.1155262	1.1539958	1.1710755	1.1498865	1.1508931
1.0307673	1.0244947	1.0313417	0.9995388	0.9937247	0.9915746
1.1177098	1.1700120	1.1501680	1.1635447	1.1489838	1.1608783
0.9922391	0.9998666	0.9881822	0.9806000	0.9666334	0.9503265
1.2259494	1.2119998	1.1991214	1.2046533	1.1883885	1.1775825

Data Modelling

1. **Basic GLM with all Variables**



```
Call:
glm(formula = diabetes ~ ., family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6370  -0.7155  -0.4053   0.7369   2.7405

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -8.8602505  0.9007060  -9.837  < 2e-16 ***
Pregnancies              0.1350774  0.0382758   3.529 0.000417 ***
Glucose                  0.0313421  0.0043035   7.283 3.27e-13 ***
BloodPressure           -0.0122181  0.0058744  -2.080 0.037537 *
SkinThickness           -0.0009409  0.0082308  -0.114 0.908988
Insulin                 -0.0006212  0.0010400  -0.597 0.550328
BMI                      0.1053255  0.0188976   5.573 2.50e-08 ***
DiabetesPedigreeFunction 1.0408221  0.3586892   2.902 0.003711 **
Age                      0.0211476  0.0113075   1.870 0.061453 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 692.91  on 536  degrees of freedom
Residual deviance: 501.79  on 528  degrees of freedom
AIC: 519.79

Number of Fisher Scoring iterations: 5
```

The outcome demonstrates that Triceps_Skin, Serum_Insulin, and Age do not have statistical significance. We can try eliminating it because the p_values are greater than 0.01.

Rational Model.

explanatory variables xk as an input, and p with k parameters as the prediction.

The logit transformation limits the range [0, 1] for the value of p.

$$\operatorname{logit}(p(\boldsymbol{x};\boldsymbol{\beta})) = \ln\left(\frac{p(\boldsymbol{x};\boldsymbol{\beta})}{1-p(\boldsymbol{x};\boldsymbol{\beta})}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m = \boldsymbol{\beta}^\top \boldsymbol{x}$$

$$p(\boldsymbol{x};\boldsymbol{\beta}) = \frac{\exp(\boldsymbol{\beta}^\top \boldsymbol{x})}{1+\exp(\boldsymbol{\beta}^\top \boldsymbol{x})}$$

βk denotes the feature's log-odds. When predictor xk rises, xk indicates how much the logarithm of the probability of a favorable result (i.e., the logit transform) increases.

The model's likelihood is as follows:

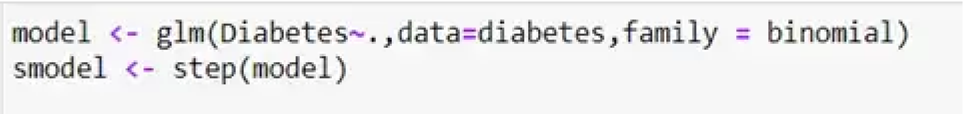
$$\ell(\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(n)};\boldsymbol{\beta}) = \prod_{i=1}^{n} p(\boldsymbol{x}^{(i)})^{y^{(i)}}(1-p(\boldsymbol{x}^{(i)}))^{1-y^{(i)}}$$

Yi = the result of subject i.

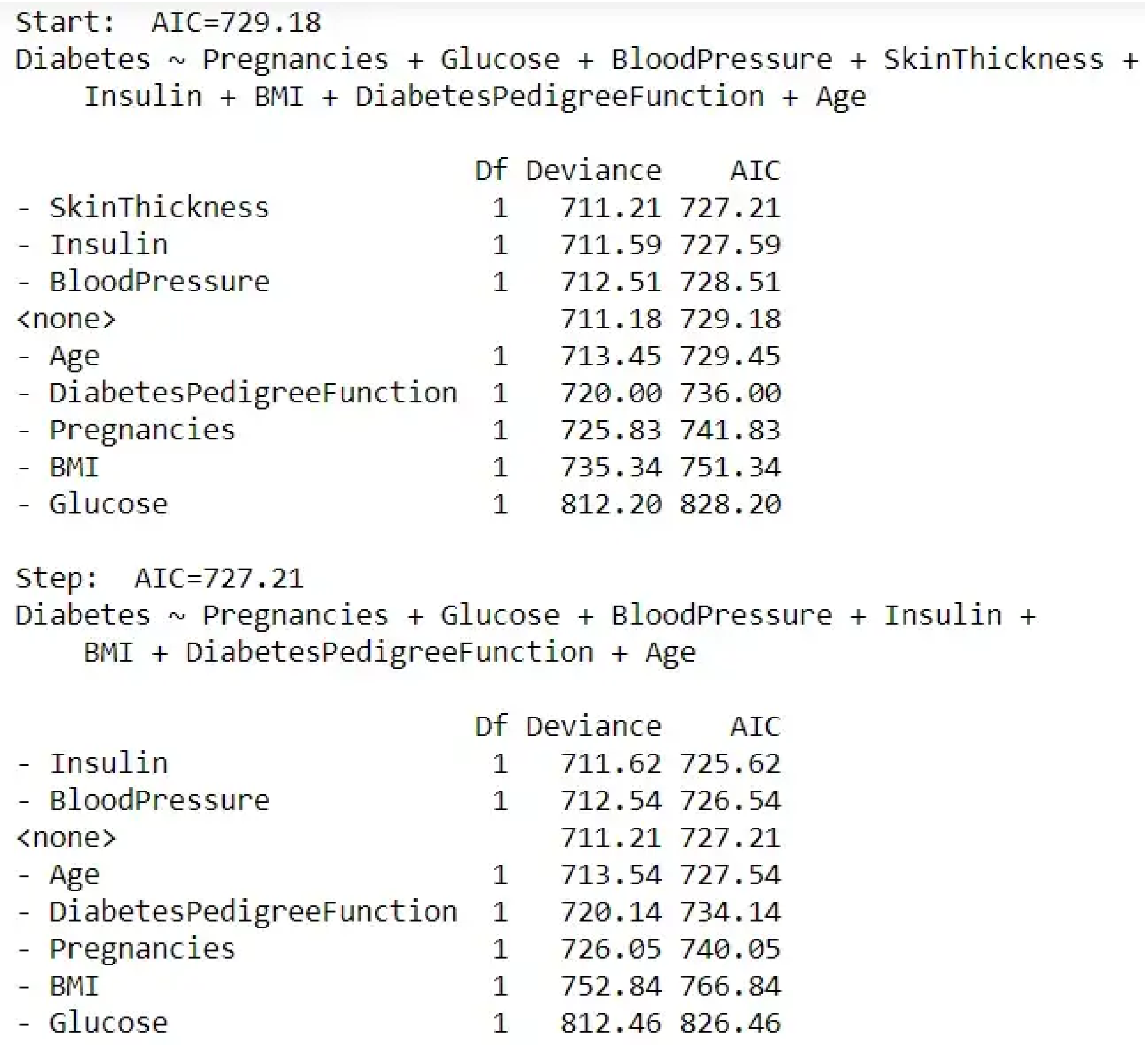
The likelihood is increased by increasing the log-likelihood (model).

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(n)};\boldsymbol{\beta}) &= \log(\ell(\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(n)};\boldsymbol{\beta})) \\ &= \sum_{i=1}^{n}\left[y^{(i)}\log(p(\boldsymbol{x}^{(i)})) + (1-y^{(i)})\log(1-p(\boldsymbol{x}^{(i)}))\right]\end{aligned}$$

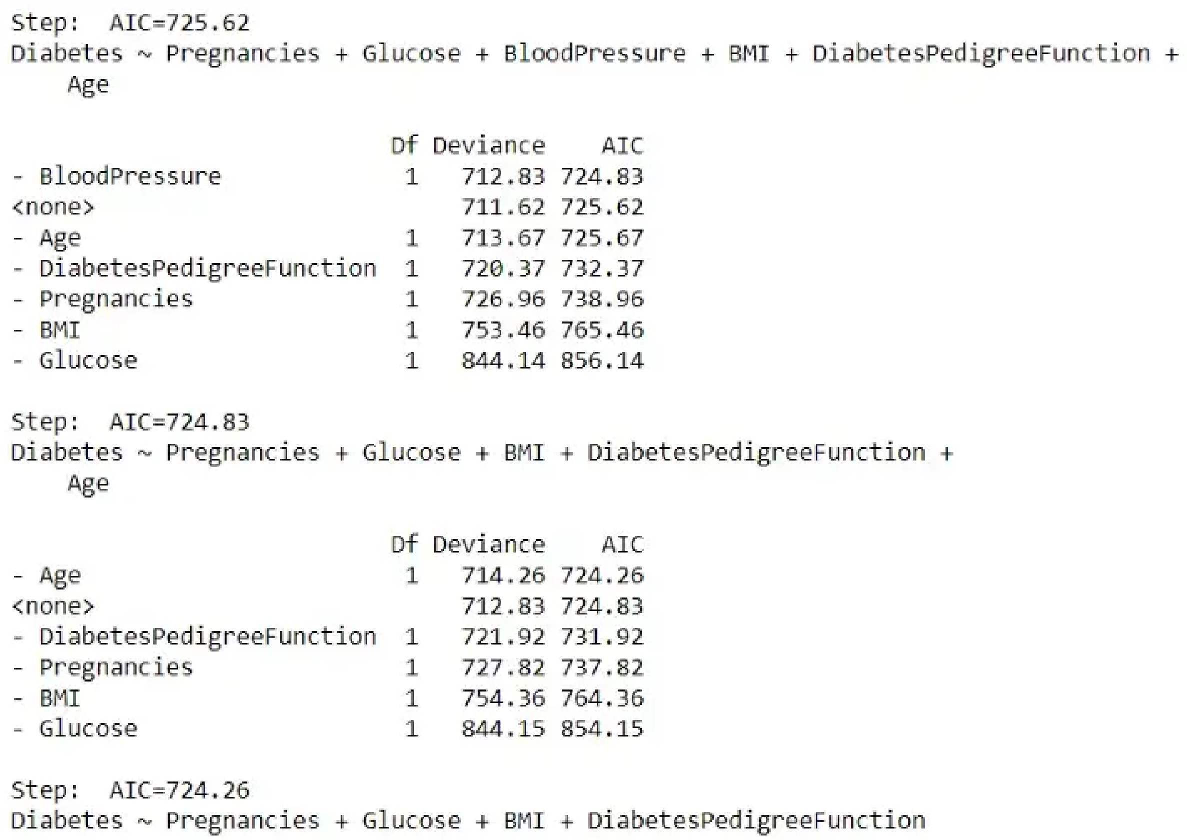
For logistic regression, the aforementioned equation is non-linear, and it is often minimized numerically using iteratively re-weighted least-squares.



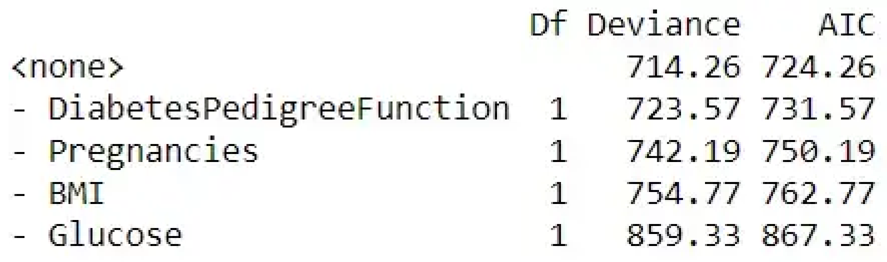
```
model <- glm(Diabetes~.,data=diabetes,family = binomial)
smodel <- step(model)
```



```
Start:  AIC=729.18
Diabetes ~ Pregnancies + Glucose + BloodPressure + SkinThickness + 
    Insulin + BMI + DiabetesPedigreeFunction + Age

                           Df Deviance    AIC
- SkinThickness             1   711.21 727.21
- Insulin                   1   711.59 727.59
- BloodPressure             1   712.51 728.51
<none>                          711.18 729.18
- Age                       1   713.45 729.45
- DiabetesPedigreeFunction  1   720.00 736.00
- Pregnancies               1   725.83 741.83
- BMI                       1   735.34 751.34
- Glucose                   1   812.20 828.20

Step:  AIC=727.21
Diabetes ~ Pregnancies + Glucose + BloodPressure + Insulin + 
    BMI + DiabetesPedigreeFunction + Age

                           Df Deviance    AIC
- Insulin                   1   711.62 725.62
- BloodPressure             1   712.54 726.54
<none>                          711.21 727.21
- Age                       1   713.54 727.54
- DiabetesPedigreeFunction  1   720.14 734.14
- Pregnancies               1   726.05 740.05
- BMI                       1   752.84 766.84
- Glucose                   1   812.46 826.46
```



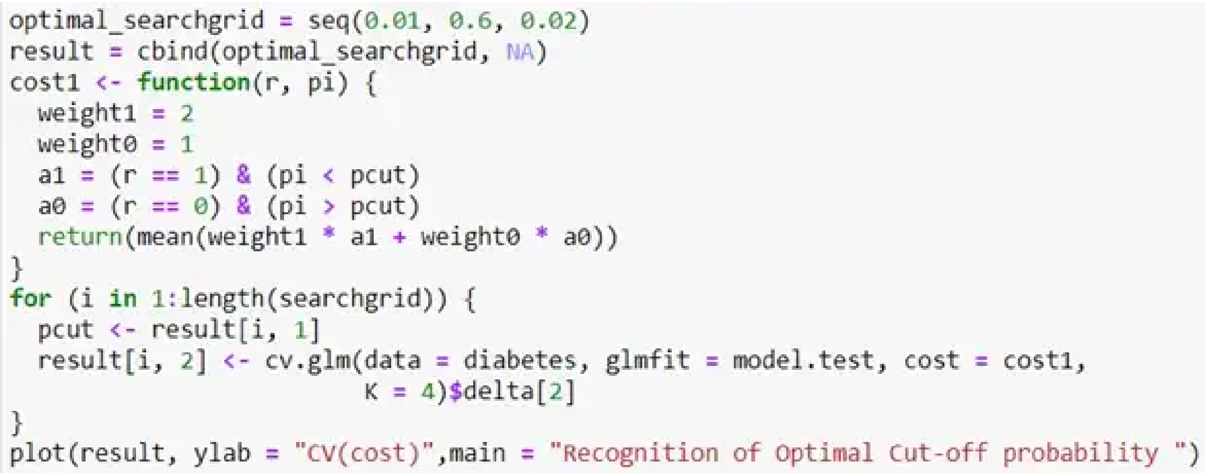
```
Step:  AIC=725.62
Diabetes ~ Pregnancies + Glucose + BloodPressure + BMI + DiabetesPedigreeFunction + 
    Age

                           Df Deviance    AIC
- BloodPressure             1   712.83 724.83
<none>                          711.62 725.62
- Age                       1   713.67 725.67
- DiabetesPedigreeFunction  1   720.37 732.37
- Pregnancies               1   726.96 738.96
- BMI                       1   753.46 765.46
- Glucose                   1   844.14 856.14

Step:  AIC=724.83
Diabetes ~ Pregnancies + Glucose + BMI + DiabetesPedigreeFunction + 
    Age

                           Df Deviance    AIC
- Age                       1   714.26 724.26
<none>                          712.83 724.83
- DiabetesPedigreeFunction  1   721.92 731.92
- Pregnancies               1   727.82 737.82
- BMI                       1   754.36 764.36
- Glucose                   1   844.15 854.15

Step:  AIC=724.26
Diabetes ~ Pregnancies + Glucose + BMI + DiabetesPedigreeFunction
```



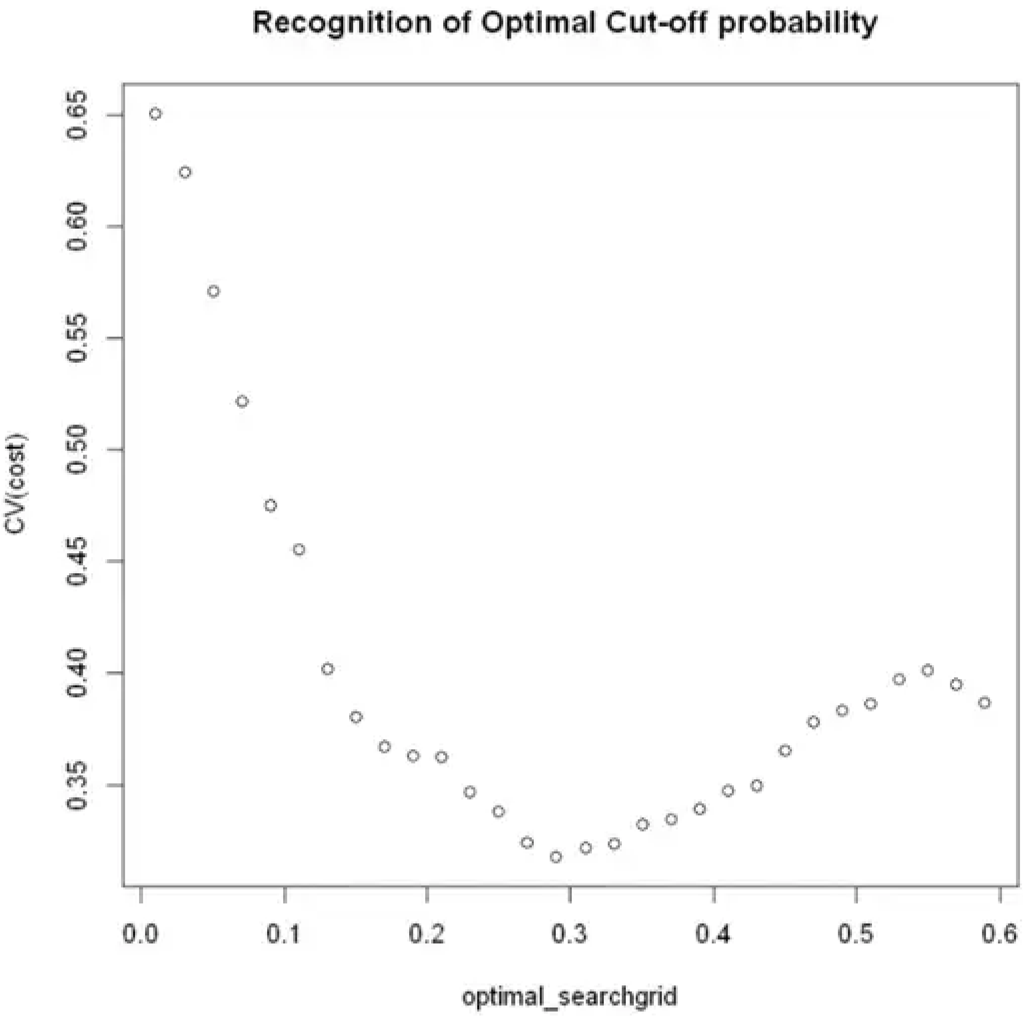
```
                           Df Deviance    AIC
<none>                          714.26 724.26
- DiabetesPedigreeFunction  1   723.57 731.57
- Pregnancies               1   742.19 750.19
- BMI                       1   754.77 762.77
- Glucose                   1   859.33 867.33
```

The logistic regression model with the lowest AIC value, 584.68, is used to determine the selection for the final model.

Setting the initial parameters.



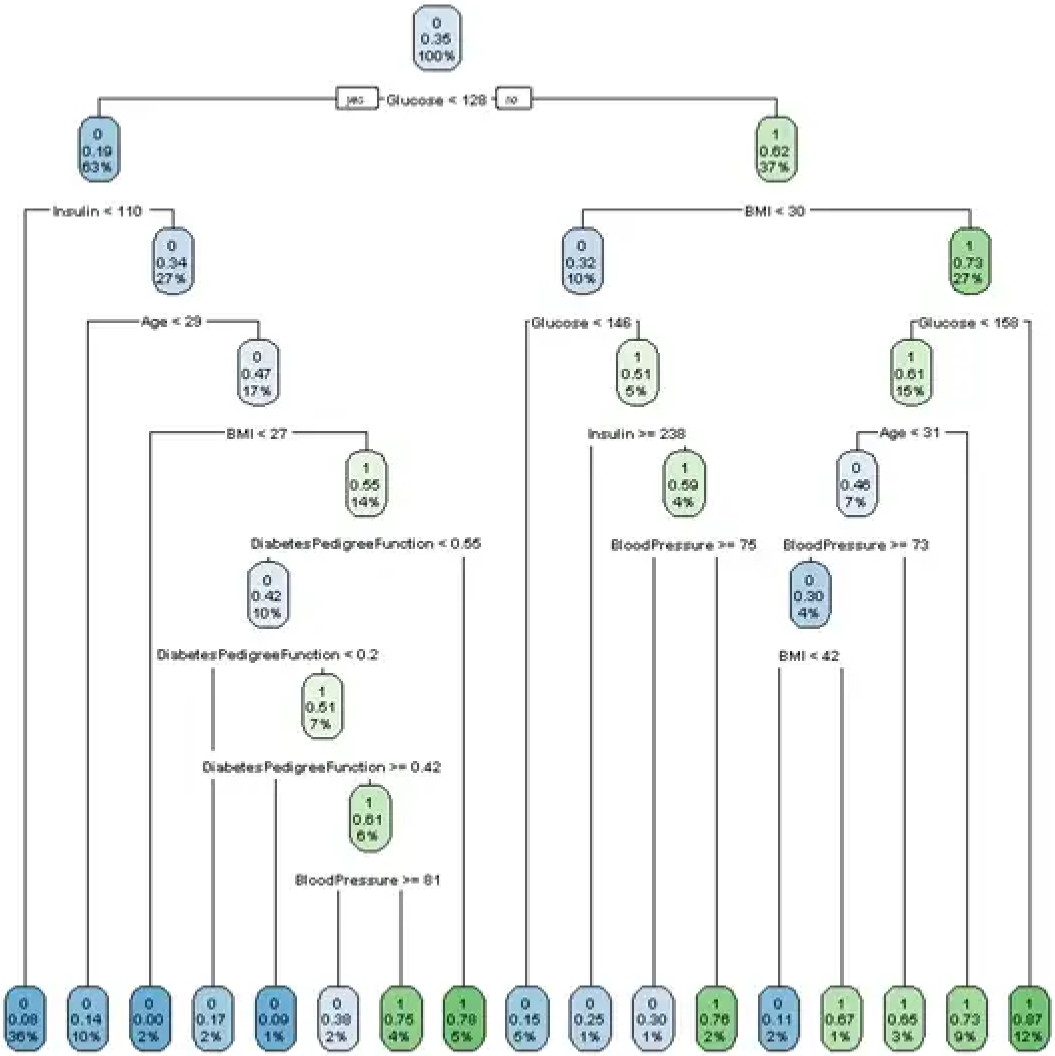
```
optimal_searchgrid = seq(0.01, 0.6, 0.02)
result = cbind(optimal_searchgrid, NA)
cost1 <- function(r, pi) {
  weight1 = 2
  weight0 = 1
  a1 = (r == 1) & (pi < pcut)
  a0 = (r == 0) & (pi > pcut)
  return(mean(weight1 * a1 + weight0 * a0))
}
for (i in 1:length(searchgrid)) {
  pcut <- result[i, 1]
  result[i, 2] <- cv.glm(data = diabetes, glmfit = model.test, cost = cost1,
                         K = 4)$delta[2]
}
plot(result, ylab = "CV(cost)",main = "Recognition of Optimal Cut-off probability ")
```



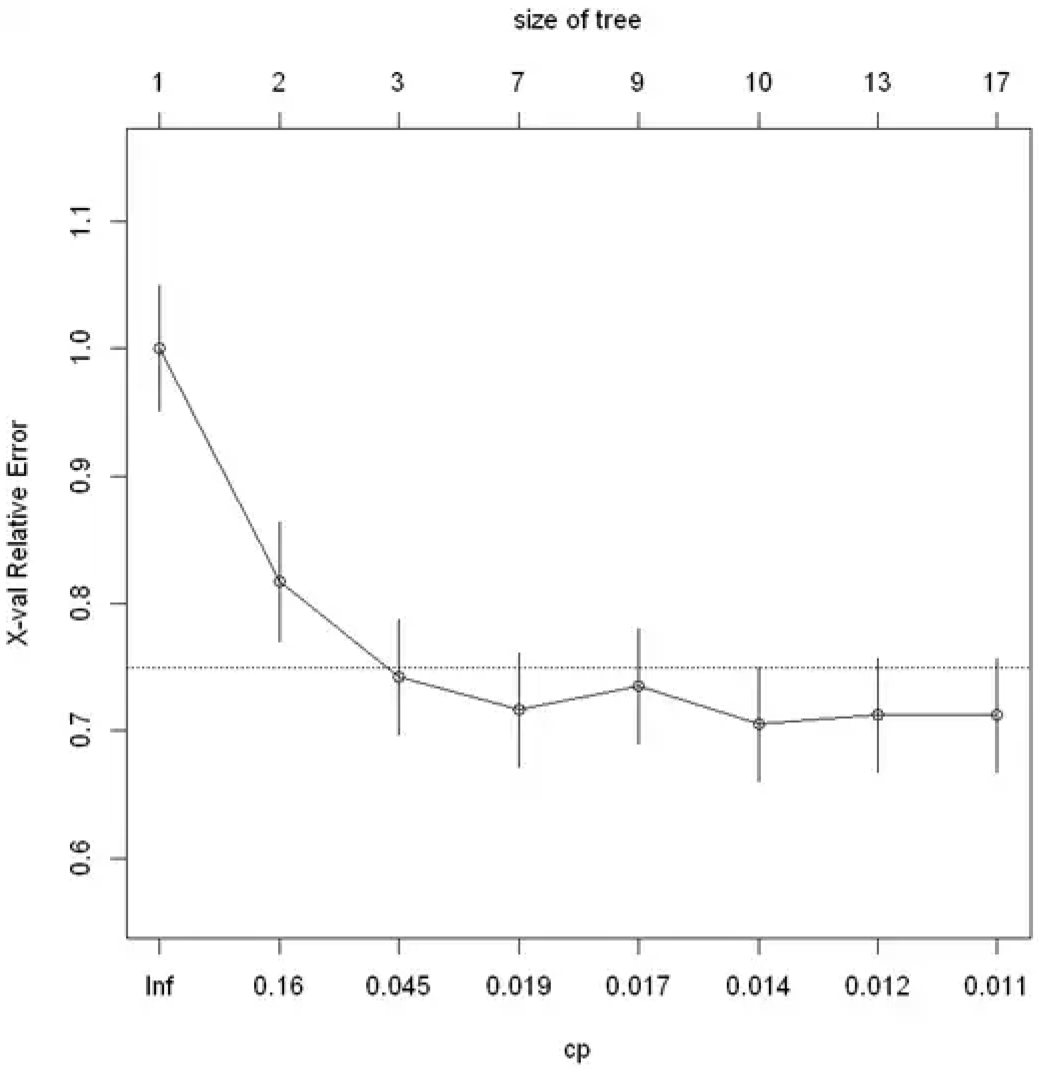
With a CV cost of 0.3370, the cross-validated cost pcut 0.28 is selected from this graph as the ideal cut-off probability.

tree <- rpart(Diabetes~., data=diabetes, method=”class”).

rpart.plot(tree)

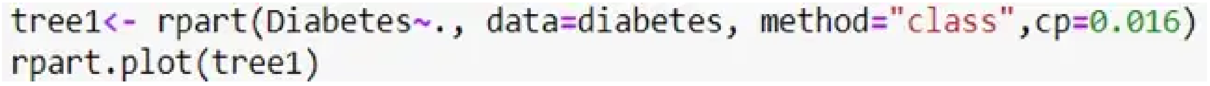


plotcp(tree)

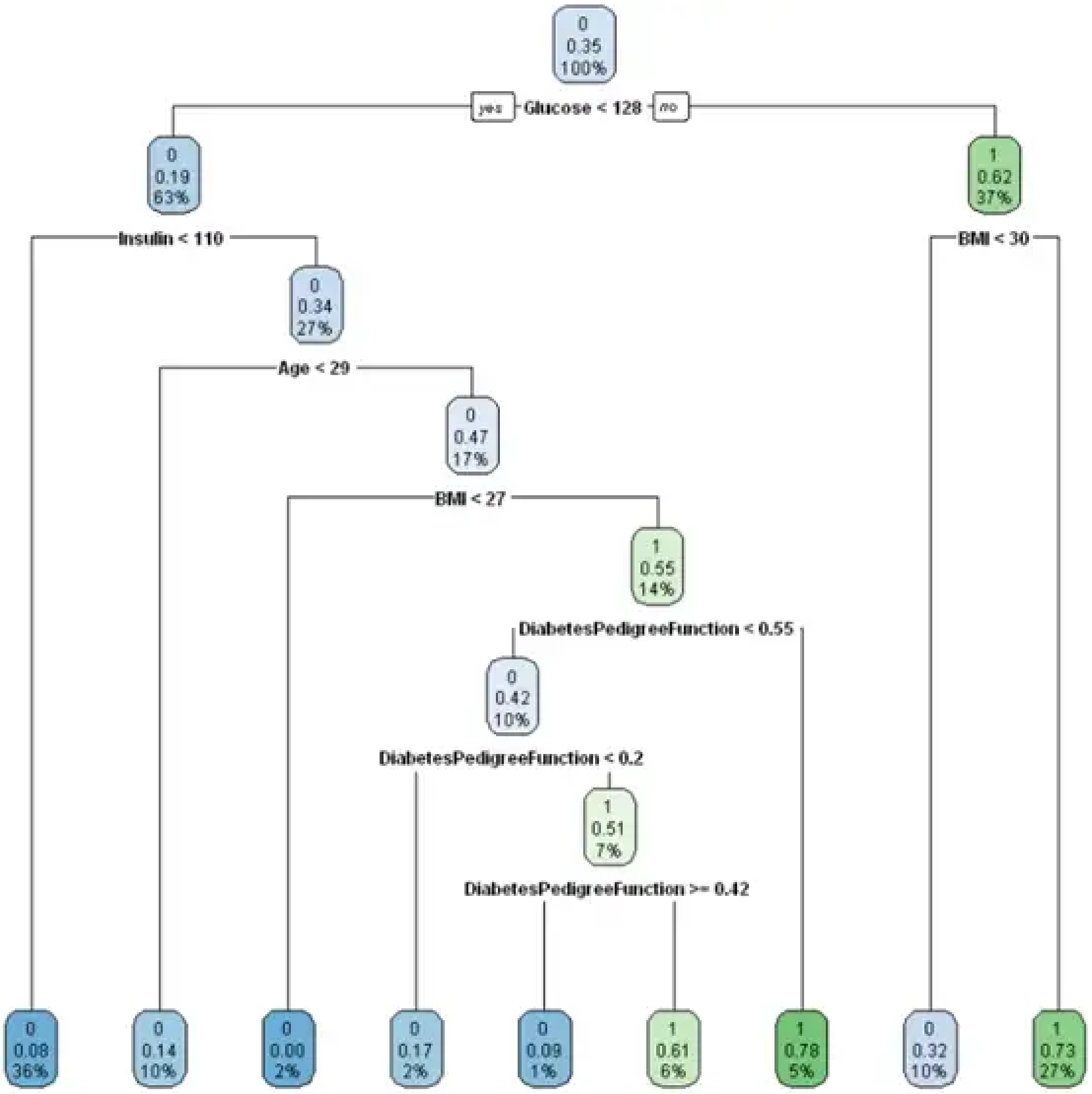


Complexity criterion.

The relative error VS complexity parameter was used as a tuning reference for the aforementioned tree. The decision tree was pruned using the Cp value of 0.016 from the previous figure. final decision-making tree.

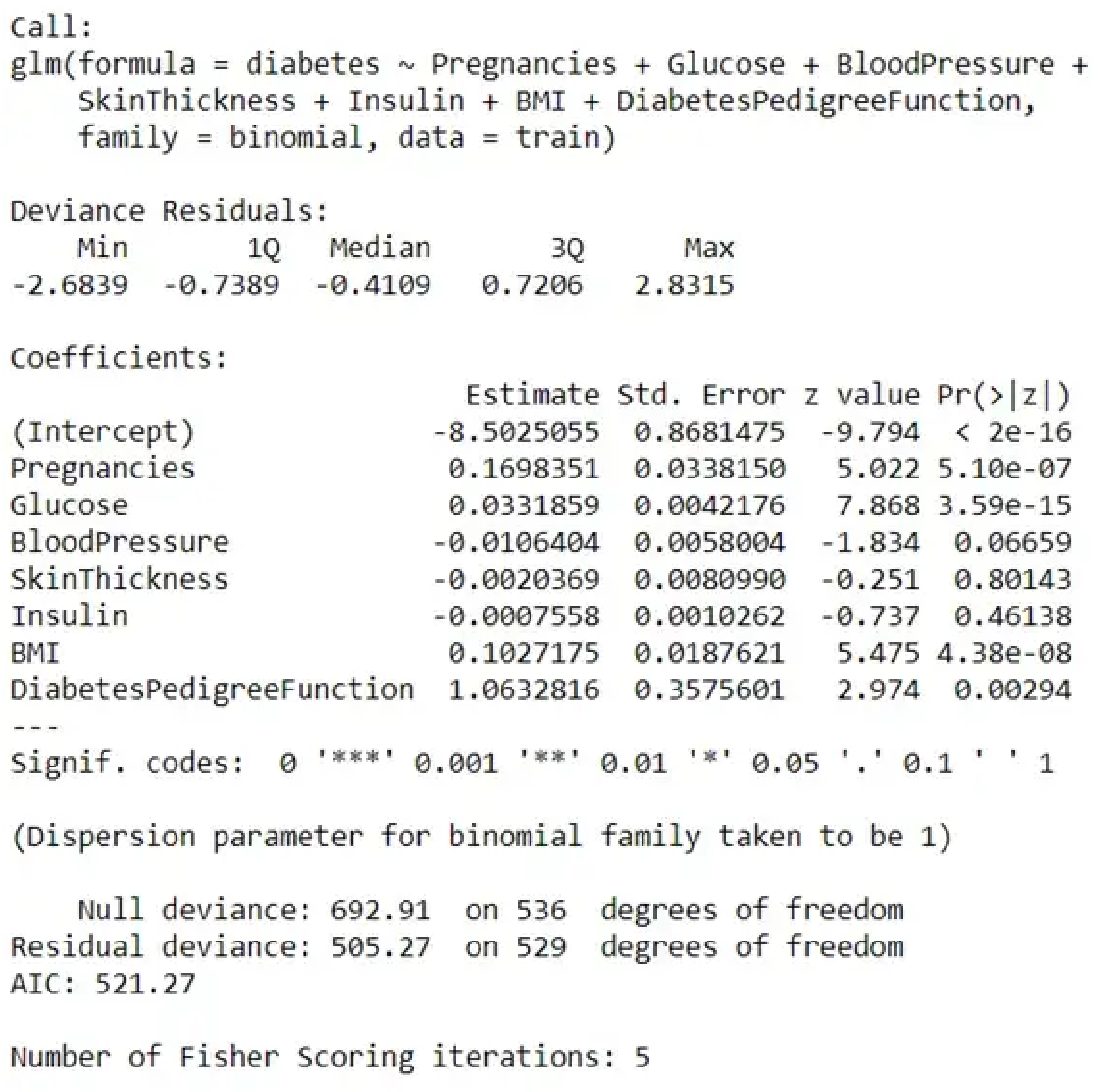


```
tree1<- rpart(Diabetes~., data=diabetes, method="class",cp=0.016)
rpart.plot(tree1)
```



The size of the tree will increase if CP is lower. No tree will be provided if cp = 1, which aids with tree pruning. An over-pruned tree can result from complexity values that are higher.

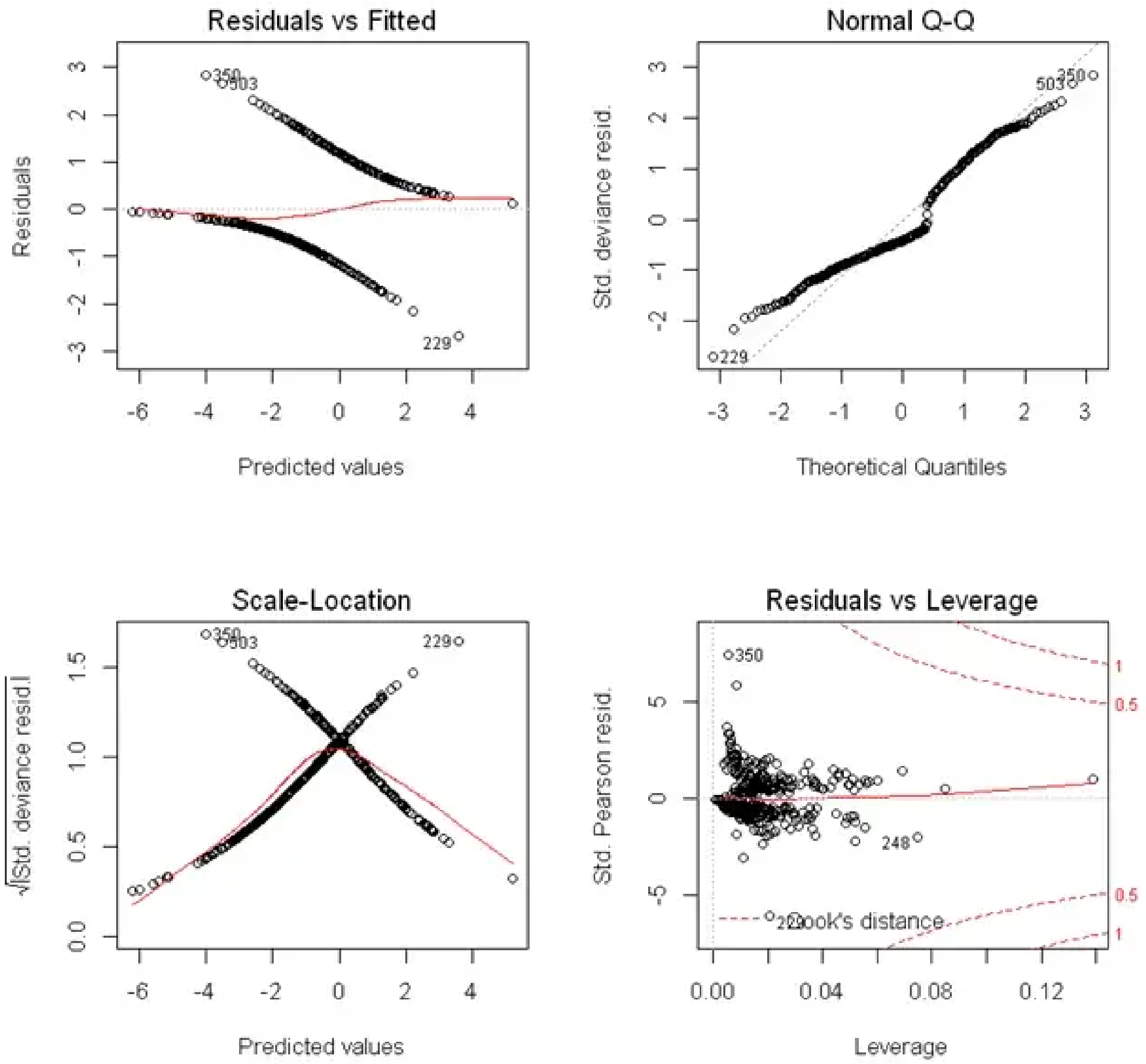
Next Model Eliminating three features:



```
Call:
glm(formula = diabetes ~ Pregnancies + Glucose + BloodPressure + 
    SkinThickness + Insulin + BMI + DiabetesPedigreeFunction, 
    family = binomial, data = train)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.6839  -0.7389  -0.4109   0.7206   2.8315  

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)    
(Intercept)              -8.5025055  0.8681475  -9.794  < 2e-16 ***
Pregnancies               0.1698351  0.0338150   5.022 5.10e-07 ***
Glucose                   0.0331859  0.0042176   7.868 3.59e-15 ***
BloodPressure            -0.0106404  0.0058004  -1.834  0.06659 .  
SkinThickness            -0.0020369  0.0080990  -0.251  0.80143    
Insulin                  -0.0007558  0.0010262  -0.737  0.46138    
BMI                       0.1027175  0.0187621   5.475 4.38e-08 ***
DiabetesPedigreeFunction  1.0632816  0.3575601   2.974  0.00294 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 692.91  on 536  degrees of freedom
Residual deviance: 505.27  on 529  degrees of freedom
AIC: 521.27

Number of Fisher Scoring iterations: 5
```

par(mfrow = c(2,2))

plot(glm_m2)



1. Residuals vs. fitted values; the fit line is indicated by the dashed line at y=0. The fit line's points show zero residual. Positive residuals are present at the locations above, and negative residuals are present at the places below. The smoothed high-order polynomial curve shown by the red line gives an explanation of the principles underlying the pattern of residual movements. Since the residuals, in this case, follow a logarithmic pattern, our model is sound.

2. Normal Q-Q Plot: This plot is typically used to determine whether or not our residuals match the normal distribution. Points are considered to closely follow the dotted line if the residuals are regularly distributed.

With the exception of the observations at 229, 350, and 503, residual points in our example closely follow the dotted line. Therefore, the model's residuals passed the normality test.

3. Scale — Place Plot: Shows how points are distributed over the projected value range.

Assumption:

- Variance over the predictor range should be about comparable (Homoscedasticity)

As a result, this horizontal red line is ideal and shows that residual variance is constant throughout the Predictor range. The red spread line rises as residuals distance themselves from one another. The data in this instance is homoscedastic, or uniform in variance.

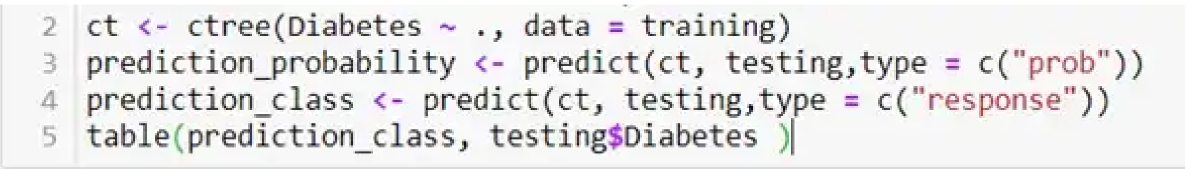
4. Leverage Plot vs. Residuals:

How much the projected scores would change if the observation were deleted can be used to define an observation's influence. Cook's Radius

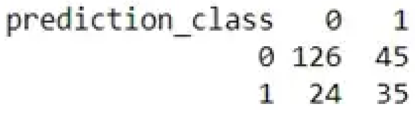
Leverage: How much the observation's value on the predictor variable deviates from the mean of the predictor variable determines the observation's leverage. The potential for an observation to have an impact increases with its level of leverage.

The locations of interest for us are those outside the dotted line on the top right or bottom right corner of our plot, where the dotted red lines indicate the cook's distance. If any point comes inside that range, we say the observation has high leverage or that there is a larger chance that excluding that point will increase its ability to influence our model.

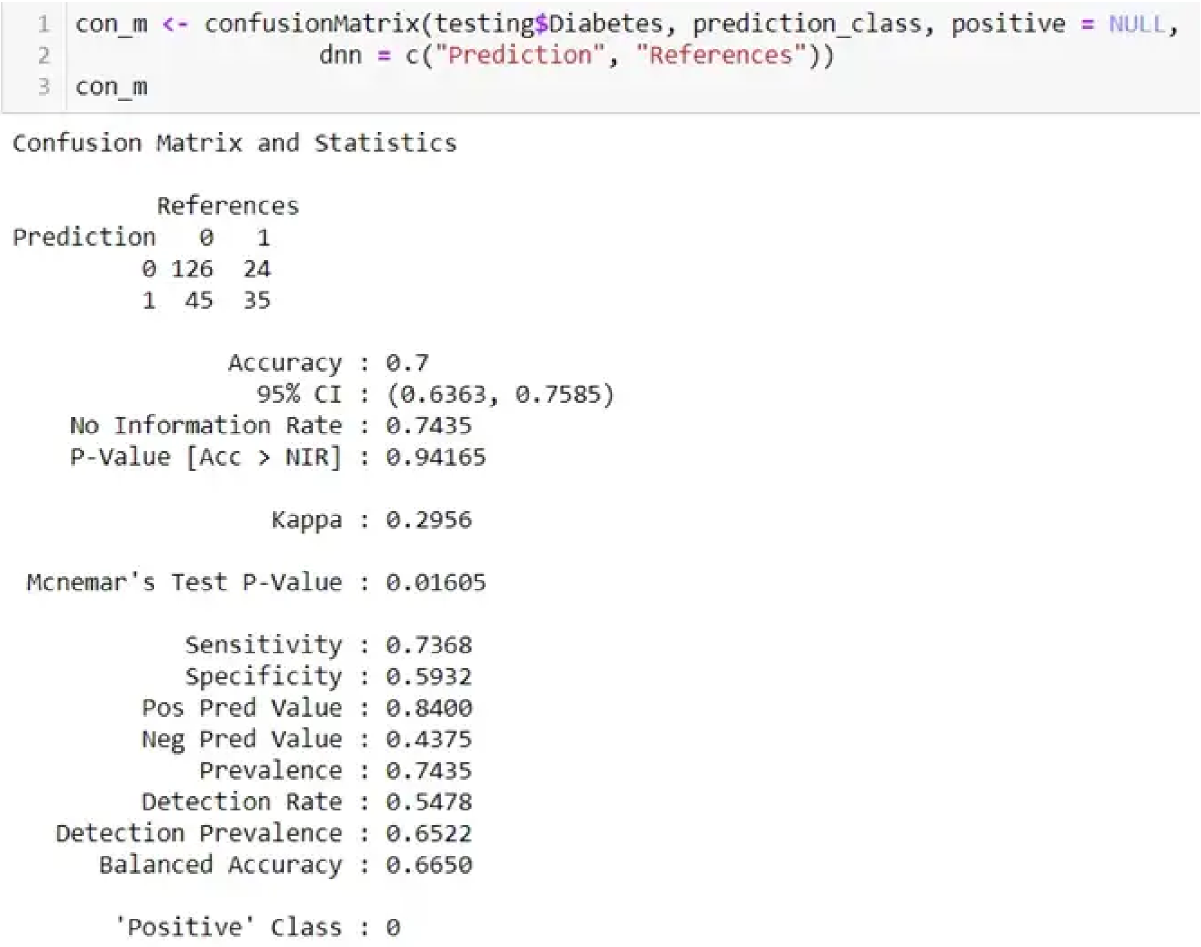
Third Model: Use a Decision Tree to Predict Diabetes Risk in New Patients



```r
ct <- ctree(Diabetes ~ ., data = training)
prediction_probability <- predict(ct, testing,type = c("prob"))
prediction_class <- predict(ct, testing,type = c("response"))
table(prediction_class, testing$Diabetes )
```



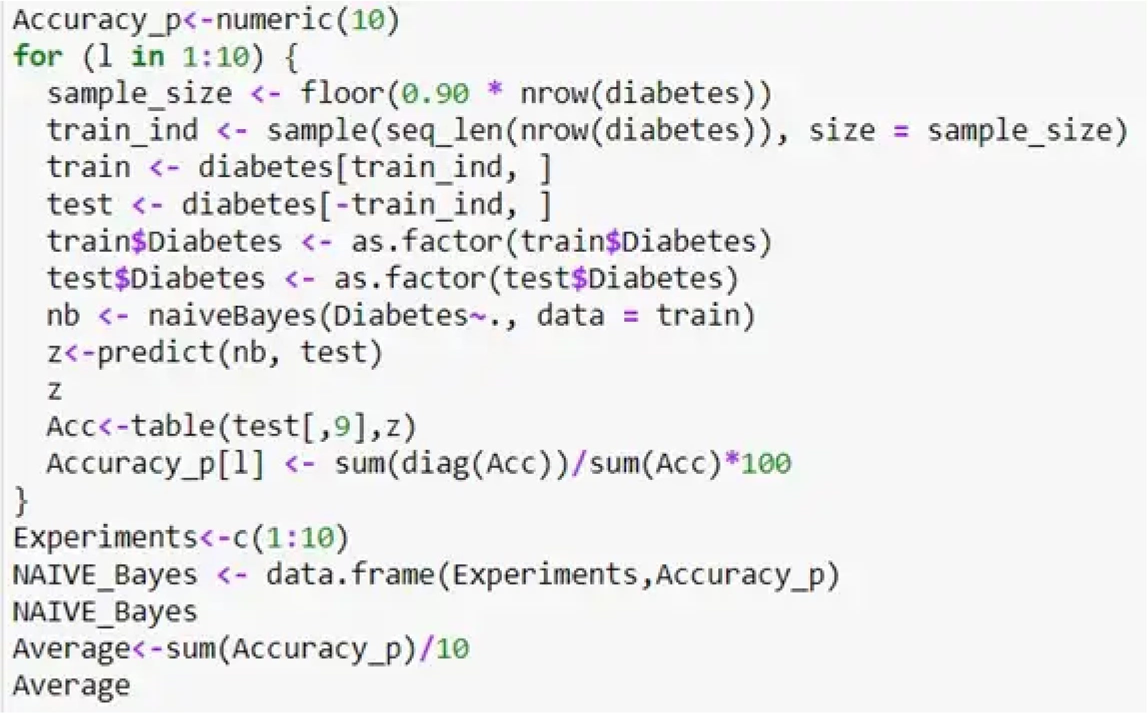
```
prediction_class   0   1
               0 126  45
               1  24  35
```



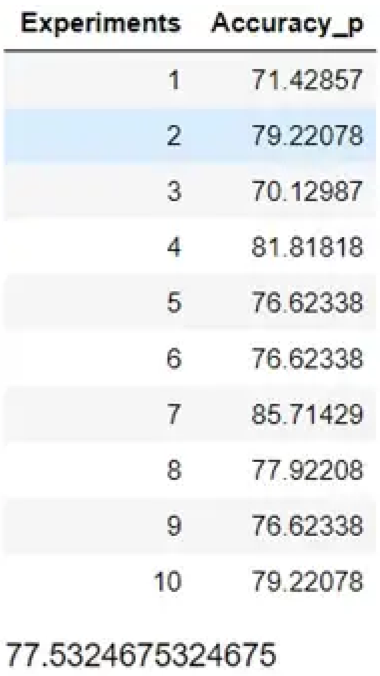
```
con_m <- confusionMatrix(testing$Diabetes, prediction_class, positive = NULL,
                         dnn = c("Prediction", "References"))
con_m
```

```
Confusion Matrix and Statistics

          References
Prediction   0   1
         0 126  24
         1  45  35

               Accuracy : 0.7
                 95% CI : (0.6363, 0.7585)
    No Information Rate : 0.7435
    P-Value [Acc > NIR] : 0.94165

                  Kappa : 0.2956

 Mcnemar's Test P-Value : 0.01605

            Sensitivity : 0.7368
            Specificity : 0.5932
         Pos Pred Value : 0.8400
         Neg Pred Value : 0.4375
             Prevalence : 0.7435
         Detection Rate : 0.5478
   Detection Prevalence : 0.6522
      Balanced Accuracy : 0.6650

       'Positive' Class : 0
```

4th Model Naïve Bayes:



```
Accuracy_p<-numeric(10)
for (l in 1:10) {
  sample_size <- floor(0.90 * nrow(diabetes))
  train_ind <- sample(seq_len(nrow(diabetes)), size = sample_size)
  train <- diabetes[train_ind, ]
  test <- diabetes[-train_ind, ]
  train$Diabetes <- as.factor(train$Diabetes)
  test$Diabetes <- as.factor(test$Diabetes)
  nb <- naiveBayes(Diabetes~., data = train)
  z<-predict(nb, test)
  z
  Acc<-table(test[,9],z)
  Accuracy_p[l] <- sum(diag(Acc))/sum(Acc)*100
}
Experiments<-c(1:10)
NAIVE_Bayes <- data.frame(Experiments,Accuracy_p)
NAIVE_Bayes
Average<-sum(Accuracy_p)/10
Average
```



Experiments	Accuracy_p
1	71.42857
2	79.22078
3	70.12987
4	81.81818
5	76.62338
6	76.62338
7	85.71429
8	77.92208
9	76.62338
10	79.22078

```
77.5324675324675
```

Despite being a simple model, it performed well, with an average accuracy rate of 77%.