MSDS660_Week7_Discussion_APeetz

2022-11-25

Adam Peetz

MSDS660 Week 7 Discussion

Regis University

Dr. Siripun Sanguansintukul

December 1st 2022

Logistic Regression for Diabetes

Data: was acquired from here: https://www.kaggle.com/uciml/pima-indians-diabetes-database

Objective: diagnostically predict whether or not a patient has diabetes, based on certain measurements included in the dataset.

```
#set working directory
setwd("C:\\Users\\adamg\\Documents\\MSDS_660\\Week_7")

#load libraries
library(data.table)
#suppressWarnings(expr)
library(car)
library(caTools)
library(readr)
library(caret)
library(Hmisc)
library(tidyverse)

# load data
data <- read_csv("diabetes.csv",show_col_types = FALSE)
# convert data to table
df <-as.data.table(data)
```

```
# convert 0s and 99s to NAs
# https://www.kaggle.com/code/dpintaric/diabetes-imputation-and-classification
df$Glucose       <- ifelse(df$Glucose       ==  0, NA, df$Glucose)
df$BloodPressure <- ifelse(df$BloodPressure ==  0, NA, df$BloodPressure)
df$SkinThickness <- ifelse(df$SkinThickness ==  0, NA, df$SkinThickness)
df$SkinThickness <- ifelse(df$SkinThickness == 99, NA, df$SkinThickness)
df$Insulin       <- ifelse(df$Insulin       ==  0, NA, df$Insulin)
df$BMI           <- ifelse(df$BMI           ==  0, NA, df$BMI)
```

```
#impute missing data
df$imputed_Glucose <- impute(df$Glucose, median)
df$imputed_BloodPressure <- impute(df$BloodPressure, median)
```

```r
df$imputed_SkinThickness <- impute(df$SkinThickness, median)
df$imputed_Insulin <- impute(df$Insulin, median)
df$imputed_BMI <- impute(df$BMI, median)
```

```r
# subset dataframe
df_1 <- df %>% dplyr::select(Pregnancies, imputed_Glucose, imputed_BloodPressure,
```

Train Test Split

```r
set.seed(1)
# model diabetes ('type' column) based on other measurements
samp <- sample.split(df_1$Outcome, SplitRatio = 0.8)
train <- subset(df_1, samp == TRUE)
test <- subset(df_1, samp == FALSE)
```

Model #1, Using All Available Data:

```r
# Create a multi linear binomial logistic regression on verified_income vs a subset of variables
model <- glm(Outcome ~ ., data = train, family = "binomial")

# Look at the model summary
summary(model)
```

```
##
## Call:
## glm(formula = Outcome ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6540  -0.6996  -0.3946   0.6972   2.1828
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -9.4134077  0.9402750 -10.011  < 2e-16 ***
## Pregnancies               0.0925563  0.0365623   2.531  0.01136 *
## imputed_Glucose           0.0383471  0.0043647   8.786  < 2e-16 ***
## imputed_BloodPressure    -0.0084003  0.0095180  -0.883  0.37747
## imputed_SkinThickness    -0.0051422  0.0153497  -0.335  0.73762
## imputed_Insulin           0.0002055  0.0013173   0.156  0.87602
## imputed_BMI               0.0987169  0.0199196   4.956  7.2e-07 ***
## DiabetesPedigreeFunction  1.0066370  0.3397765   2.963  0.00305 **
## Age                       0.0166063  0.0109444   1.517  0.12918
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 793.94  on 613  degrees of freedom
## Residual deviance: 560.50  on 605  degrees of freedom
## AIC: 578.5
##
```

```
## Number of Fisher Scoring iterations: 5
```

```
# Check for collinearity
vif(model)
```

```
##              Pregnancies          imputed_Glucose    imputed_BloodPressure
##                 1.443289                 1.188671                 1.186735
##    imputed_SkinThickness          imputed_Insulin              imputed_BMI
##                 1.357860                 1.141413                 1.438603
## DiabetesPedigreeFunction                      Age
##                 1.013234                 1.559993
```

Feature Selection by Stepwise AIC

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
# Perform stepAIC and remove variables with high p-values
stepAIC(model, direction = 'both')
```

```
## Start:  AIC=578.5
## Outcome ~ Pregnancies + imputed_Glucose + imputed_BloodPressure +
##     imputed_SkinThickness + imputed_Insulin + imputed_BMI + DiabetesPedigreeFunction +
##     Age
##
##                            Df Deviance    AIC
## - imputed_Insulin           1   560.53 576.53
## - imputed_SkinThickness     1   560.61 576.61
## - imputed_BloodPressure     1   561.28 577.28
## <none>                          560.50 578.50
## - Age                       1   562.79 578.79
## - Pregnancies               1   567.05 583.05
## - DiabetesPedigreeFunction  1   569.47 585.47
## - imputed_BMI               1   587.41 603.41
## - imputed_Glucose           1   659.08 675.08
##
## Step:  AIC=576.53
## Outcome ~ Pregnancies + imputed_Glucose + imputed_BloodPressure +
##     imputed_SkinThickness + imputed_BMI + DiabetesPedigreeFunction +
##     Age
##
##                            Df Deviance    AIC
## - imputed_SkinThickness     1   560.64 574.64
## - imputed_BloodPressure     1   561.33 575.33
## <none>                          560.53 576.53
## - Age                       1   562.82 576.82
## + imputed_Insulin           1   560.50 578.50
## - Pregnancies               1   567.07 581.07
## - DiabetesPedigreeFunction  1   569.55 583.55
## - imputed_BMI               1   587.71 601.71
## - imputed_Glucose           1   675.66 689.66
##
## Step:  AIC=574.64
## Outcome ~ Pregnancies + imputed_Glucose + imputed_BloodPressure +
##     imputed_BMI + DiabetesPedigreeFunction + Age
##
##                            Df Deviance    AIC
## - imputed_BloodPressure     1   561.46 573.46
## <none>                          560.64 574.64
## - Age                       1   562.87 574.87
## + imputed_SkinThickness     1   560.53 576.53
## + imputed_Insulin           1   560.61 576.61
## - Pregnancies               1   567.12 579.12
## - DiabetesPedigreeFunction  1   569.59 581.59
## - imputed_BMI               1   594.46 606.46
## - imputed_Glucose           1   675.70 687.70
##
## Step:  AIC=573.46
## Outcome ~ Pregnancies + imputed_Glucose + imputed_BMI + DiabetesPedigreeFunction +
##     Age
##
##                            Df Deviance    AIC
## - Age                       1   563.23 573.23
## <none>                          561.46 573.46
## + imputed_BloodPressure     1   560.64 574.64
## + imputed_SkinThickness     1   561.33 575.33
## + imputed_Insulin           1   561.41 575.41
## - Pregnancies               1   567.75 577.75
## - DiabetesPedigreeFunction  1   570.77 580.77
## - imputed_BMI               1   594.81 604.81
## - imputed_Glucose           1   675.80 685.80
##
## Step:  AIC=573.23
## Outcome ~ Pregnancies + imputed_Glucose + imputed_BMI + DiabetesPedigreeFunction
##
##                            Df Deviance    AIC
## <none>                          563.23 573.23
## + Age                       1   561.46 573.46
## + imputed_BloodPressure     1   562.87 574.87
## + imputed_SkinThickness     1   563.16 575.16
## + imputed_Insulin           1   563.19 575.19
## - DiabetesPedigreeFunction  1   572.65 580.65
## - Pregnancies               1   577.92 585.92
## - imputed_BMI               1   595.39 603.39
## - imputed_Glucose           1   689.18 697.18
```

```
##
## Call:  glm(formula = Outcome ~ Pregnancies + imputed_Glucose + imputed_BMI +
##     DiabetesPedigreeFunction, family = "binomial", data = train)
##
## Coefficients:
##             (Intercept)               Pregnancies           imputed_Glucose
##                -9.48687                   0.11654                   0.03903
##              imputed_BMI  DiabetesPedigreeFunction
##                  0.09020                   1.02381
## 
## Degrees of Freedom: 613 Total (i.e. Null);  609 Residual
## Null Deviance:       793.9
## Residual Deviance: 563.2     AIC: 573.2
```

Model2: Features Selected by Stepwise AIC

```
# Update model based on the stepAIC
model2 <-  glm(formula = Outcome ~ Pregnancies + imputed_Glucose + imputed_BMI +
    DiabetesPedigreeFunction, family = "binomial", data = train)
```

Confusion Matrix for Predictions on Training

```
#Make a Prediction
trainpreds <- predict(model2, type = 'response', train)


# Round prediction values at 0.5 cutoff factor and change labels
trainp <- factor(trainpreds >= 0.5,labels = c('0', '1'))

# Buld a confustion matrix to see results
trainCM <- confusionMatrix(trainp, as.factor(train$Outcome))
trainCM
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 354  93
##          1  46 121
## 
##                Accuracy : 0.7736
##                  95% CI : (0.7384, 0.8061)
##     No Information Rate : 0.6515
##     P-Value [Acc > NIR] : 3.308e-11
## 
##                   Kappa : 0.4747
## 
##  Mcnemar's Test P-Value : 9.553e-05
## 
##             Sensitivity : 0.8850
##             Specificity : 0.5654
##          Pos Pred Value : 0.7919
##          Neg Pred Value : 0.7246
##              Prevalence : 0.6515
##          Detection Rate : 0.5765
##    Detection Prevalence : 0.7280
##       Balanced Accuracy : 0.7252
## 
##        'Positive' Class : 0
```

```
##
```

Confusion Matrix for Predictions on Test

```r
# predict on the test data
testpreds <- predict(model2, type = 'response', test)

# Round prediction values at 0.5 cutoff factor and change labels
testp <- factor(testpreds >= 0.5, labels = c('0', '1'))

# Build a confusion matrix to see results
testCM <- confusionMatrix(testp, as.factor(test$Outcome))
testCM
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 91 25
##          1  9 29
##
##                Accuracy : 0.7792
##                  95% CI : (0.7054, 0.842)
##     No Information Rate : 0.6494
##     P-Value [Acc > NIR] : 0.0003315
##
##                   Kappa : 0.4797
##
##  Mcnemar's Test P-Value : 0.0100973
##
##             Sensitivity : 0.9100
##             Specificity : 0.5370
##          Pos Pred Value : 0.7845
##          Neg Pred Value : 0.7632
##              Prevalence : 0.6494
##          Detection Rate : 0.5909
##    Detection Prevalence : 0.7532
##       Balanced Accuracy : 0.7235
##
##        'Positive' Class : 0
##
```

ROC Curve and Threshold

```r
#Create a Roc curve and plot results for the prediction-based data
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
```

```
##     cov, smooth, var
```

```
# Create a Roc curve and results for the Train data
train_roc_curve <- roc(train$Outcome, trainpreds)
```

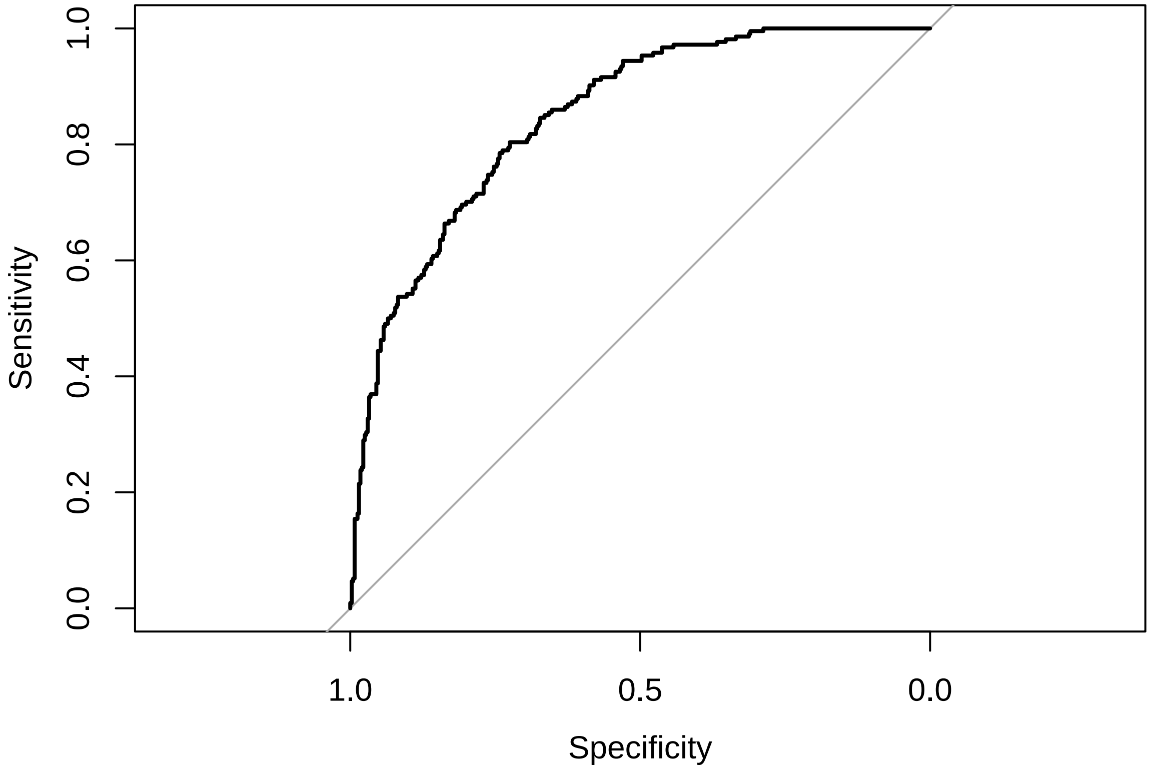
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
train_roc_curve
```

```
##
## Call:
## roc.default(response = train$Outcome, predictor = trainpreds)
##
## Data: trainpreds in 400 controls (train$Outcome 0) < 214 cases (train$Outcome 1).
## Area under the curve: 0.8478
```

```
plot(train_roc_curve)
```



```
train_rocc <- coords(roc=train_roc_curve, x = 'best', best.method = 'closest.topleft')
train_rocc
```

```
##   threshold specificity sensitivity
## 1 0.3010611      0.7425   0.7850467
```

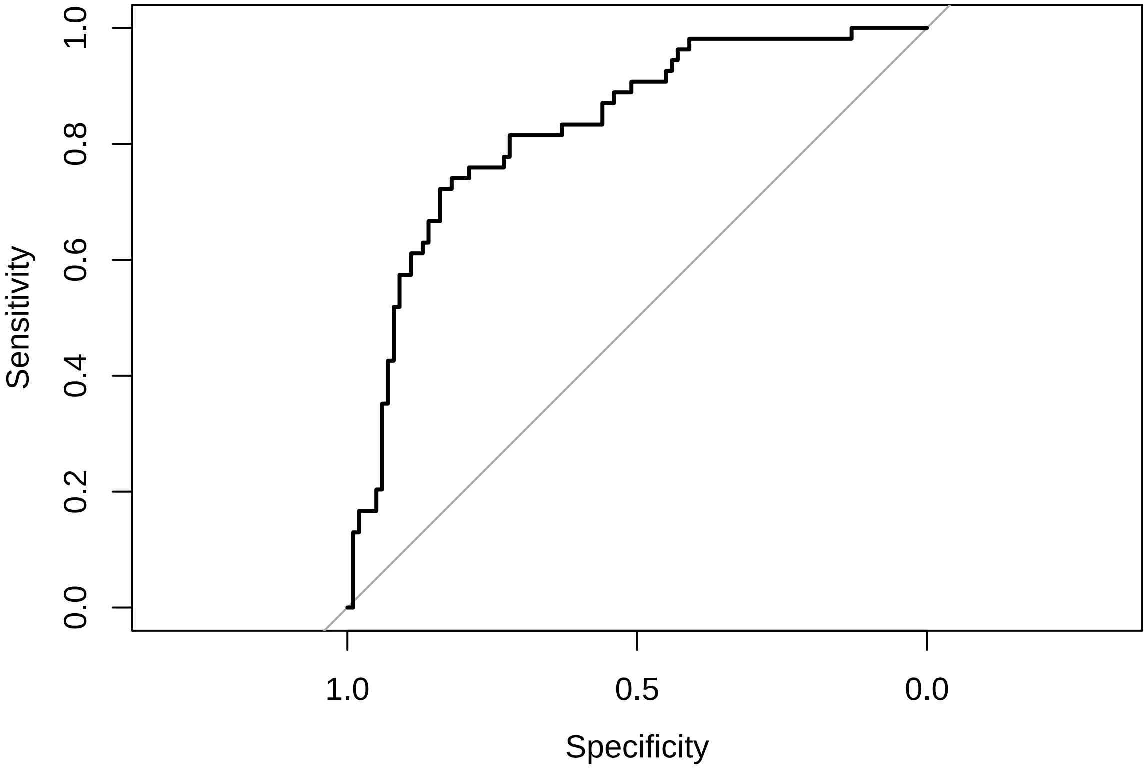
```
# Create a Roc curve and results for the Test data
test_roc_curve <- roc(test$Outcome, testpreds)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
test_roc_curve
```

```
##
## Call:
## roc.default(response = test$Outcome, predictor = testpreds)
##
## Data: testpreds in 100 controls (test$Outcome 0) < 54 cases (test$Outcome 1).
## Area under the curve: 0.8276
```

```
plot(test_roc_curve)
```



```
#set the threshold
thresh <- coords(roc=test_roc_curve, x = 'best', best.method = 'closest.topleft', transpose=TRUE)

#look at what the best threshold is
thresh
```

```
##   threshold specificity sensitivity
##   0.3270255   0.8200000   0.7407407
```

Modifying Predictions with a Fine-tuned Threshold

```
#round prediction
rounded_preds <- as.factor(as.integer(testpreds > thresh[1]))
targets <- as.factor(as.integer(test$Outcome))
```

```r
library(caret)
# prepare data for confusion matrix
postResample(pred = rounded_preds, obs = targets)
```

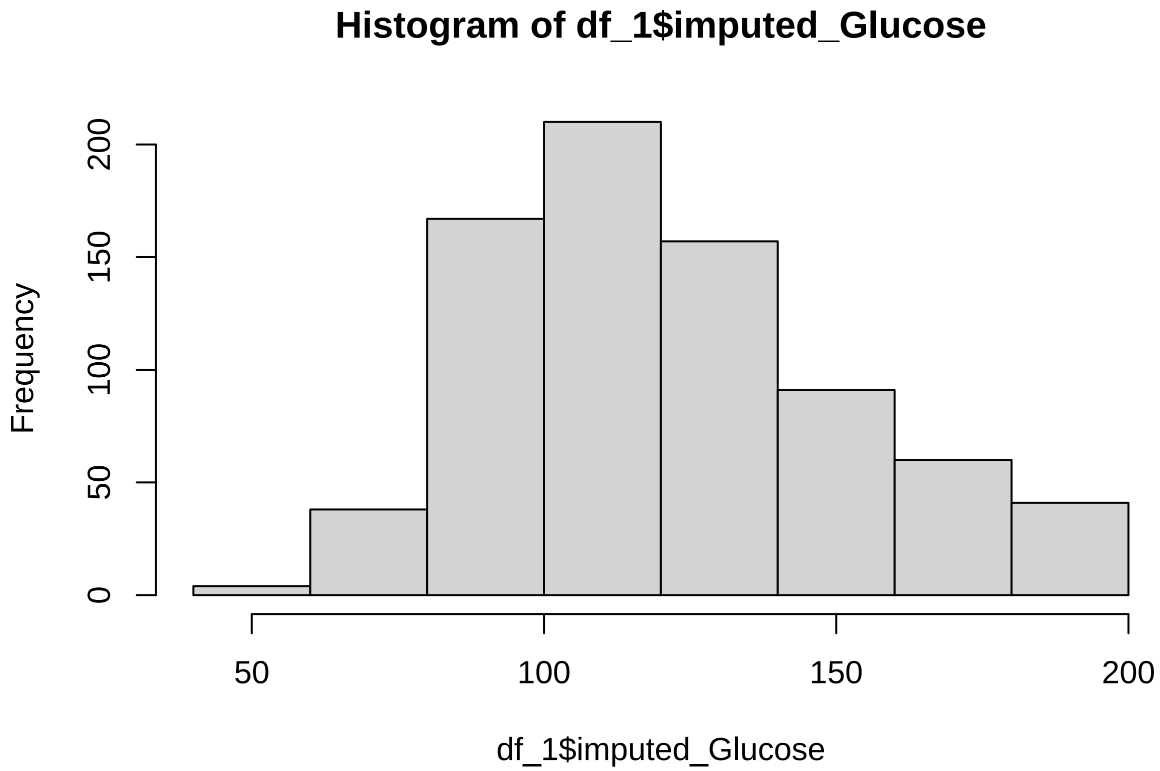
```
##  Accuracy     Kappa 
## 0.7922078 0.5513474
```

```r
# Accuracy needs to be higher than No Information Rate (guesses)
confusionMatrix(rounded_preds, targets)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  0  1
##          0 82 14
##          1 18 40
## 
##                Accuracy : 0.7922
##                  95% CI : (0.7195, 0.8533)
##     No Information Rate : 0.6494
##     P-Value [Acc > NIR] : 8.061e-05
## 
##                   Kappa : 0.5513
## 
##  Mcnemar's Test P-Value : 0.5959
## 
##             Sensitivity : 0.8200
##             Specificity : 0.7407
##          Pos Pred Value : 0.8542
##          Neg Pred Value : 0.6897
##              Prevalence : 0.6494
##          Detection Rate : 0.5325
##    Detection Prevalence : 0.6234
##       Balanced Accuracy : 0.7804
## 
##        'Positive' Class : 0
## 
```

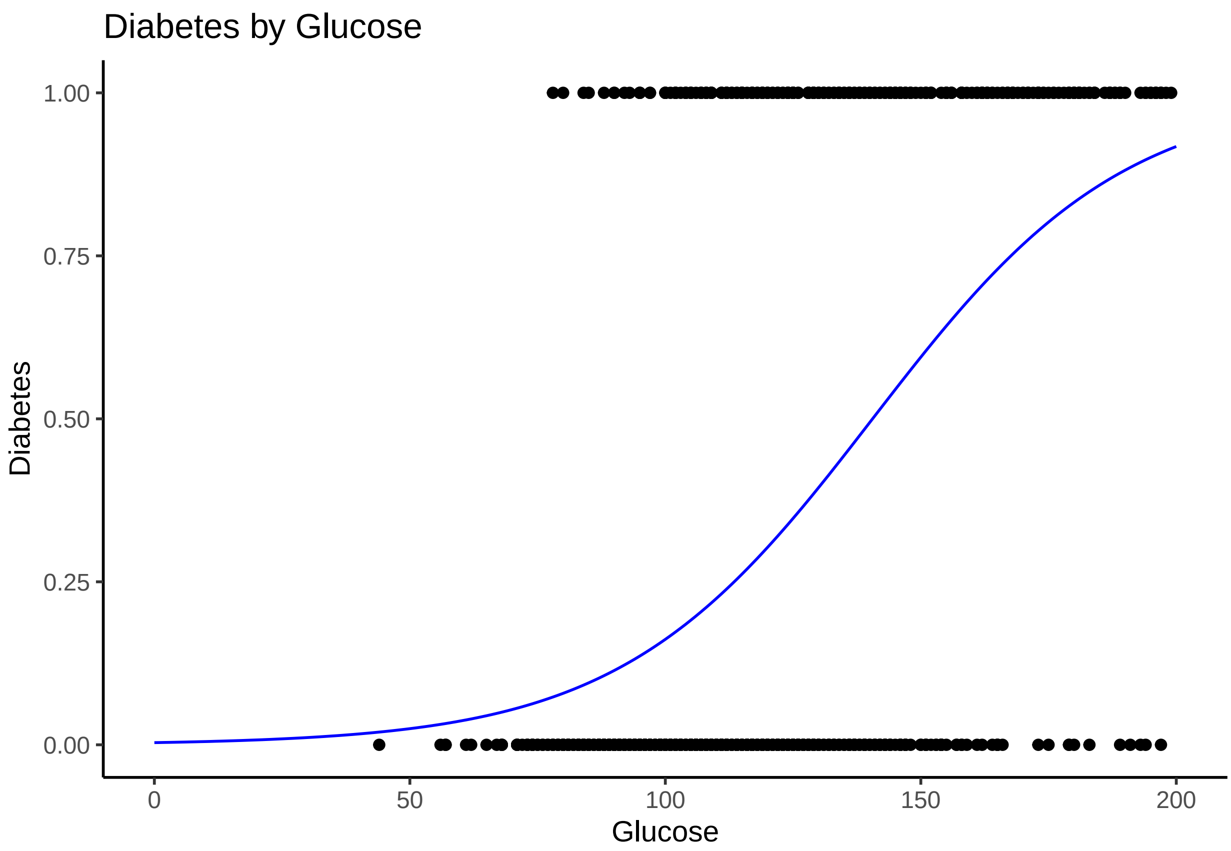
Glucose’s effect on Diabetes

```r
#Finally, predict the probability of a range of glucose values on the potential of having diabetes
model_ir <- glm(Outcome ~ imputed_Glucose, data = df_1, family = "binomial")
hist(df_1$imputed_Glucose)
```



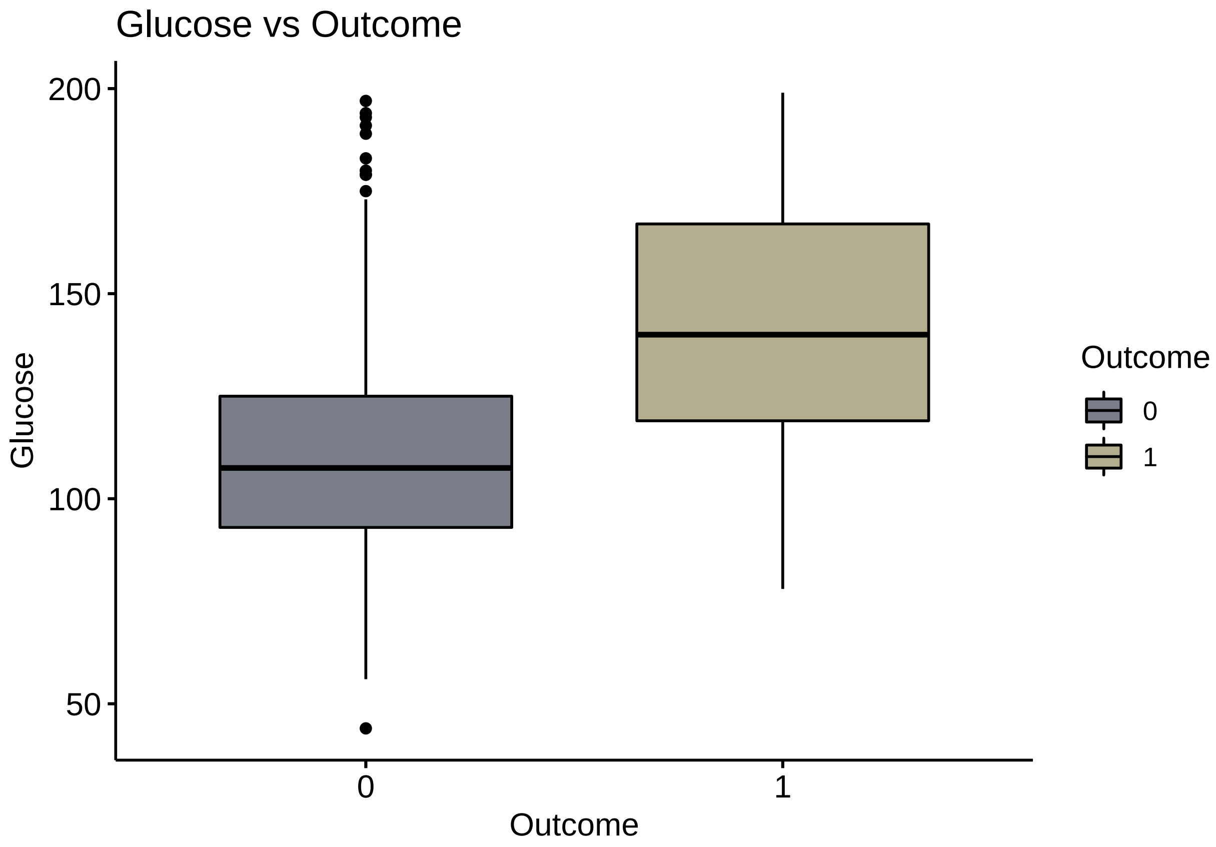
```
x <- data.table(imputed_Glucose=c(0:200))
predictions <- predict(model_ir,newdata=x, type='response') # Create predictions using the fitted model
x$probability<-predictions # Add predictions to datatable for plotting

#Visualize the predictions
ggplot(df_1) +
  aes(x=imputed_Glucose, y=Outcome) +
  geom_point() +
  theme_bw() +
  theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"))+
  geom_line(data=x, aes(x=imputed_Glucose, y=probability), color='blue') +
  labs(title='Diabetes by Glucose', x='Glucose', y='Diabetes') # Scatter verification status by interes
```



Glucose vs Glucose

```
library(ggpubr)
#install.packages("viridis")
library("viridis")
# check normality of distributions
ggboxplot(df_1, x = "Outcome", y='imputed_Glucose', fill='Outcome',
          main = "Glucose vs Outcome",
          xlab = "Outcome",
          ylab = "Glucose") +
          theme(legend.position="right")+
          scale_fill_viridis(discrete = TRUE, alpha=0.8, begin =0.34, end=0.65, option="E")
```



Conclude with a summary of what you did and your results.

Missing Values

The dataset is full of missing values, represented by 0's. These missing values are converted to NA and then filled with the median value for each column in the dataset.

VIF

A VIF check shows all features with VIF values between 1 and 2. There are no multicollinearity issues in the dataset.

StepwiseAIC

StepwiseAIC is performed to select an ideal feature set for the model. Stepwise AIC suggests the ideal combination of features is Pregnancies + imputed_Glucose + imputed_BMI + DiabetesPedigreeFunction.

Performance of Model w/o Threshold Tuning

The stepwise model has an accuracy rate of 77% compared to a 64% null information rate.

Performance of model w/ Threshold Tuning

The performance of the model changes once the threshold is adjusted. An accuracy of 79% is achieved by the model after setting the threshold to 0.3.

Glucose Effect on Outcome

Higher glucose levels correspond to an increased risk of diabetes. Almost all rows containing glucose over 150 have a positive outcome. Plotting glucose for each outcome shows two distinct distributions for each group. Additional research should be done to test if the difference in means between the two groups is significant.