STAT 4601 - Assignment 3

Nathan Situ

2023-11-29

Part 1: Information about the Data

We first start by looking at the data itself.

```
## Dimensions of the Data by Row and Column : 768 9
```

```
## Column names : pregnancies glucose diastolic thickness insulin BMI pedigree age diabetes
```

```
## Summary Statistics of the data :
```

```
##   pregnancies       glucose        diastolic       thickness
##  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           pedigree           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
##     diabetes
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
```

```
## [1] "First few rows of the data :"
```

```
## # A tibble: 6 x 9
##   pregnancies glucose diastolic thickness insulin   BMI pedigree   age diabetes
##         <dbl>   <dbl>     <dbl>     <dbl>   <dbl> <dbl>    <dbl> <dbl>    <dbl>
## 1           6     148        72        35       0  33.6    0.627    50        1
## 2           1      85        66        29       0  26.6    0.351    31        0
## 3           8     183        64         0       0  23.3    0.672    32        1
## 4           1      89        66        23      94  28.1    0.167    21        0
## 5           0     137        40        35     168  43.1    2.29     33        1
## 6           5     116        74         0       0  25.6    0.201    30        0
```

There are 768 observations with 9 columns, with the 'diabetes' columns being the variable interest we are trying to predict in patients, with the rest of the 8 columns representing the input variables we'll use to help in our prediction.

Visualzing Data and Imputation

Before we start our modelling, we should do some visualizations of our data. It was noted that missing values were encoded with zero values, we should verify that this is the case and see if skews the distributions for certain output variables.

```
## [1] "Number of rows with NA values: "
```

```
## [1] 0
```

```
## [1] "Number of zeroes in each of the column variables: "
```

```
## pregnancies     glucose   diastolic   thickness     insulin         BMI
##         111           5          35         227         374          11
##    pedigree         age    diabetes
##           0           0         500
```

We that several of the outputs seem to have unusually high zero values, including insulin, thickness, and pregnancies. We next draw a histogram to see if they affect the distributions.

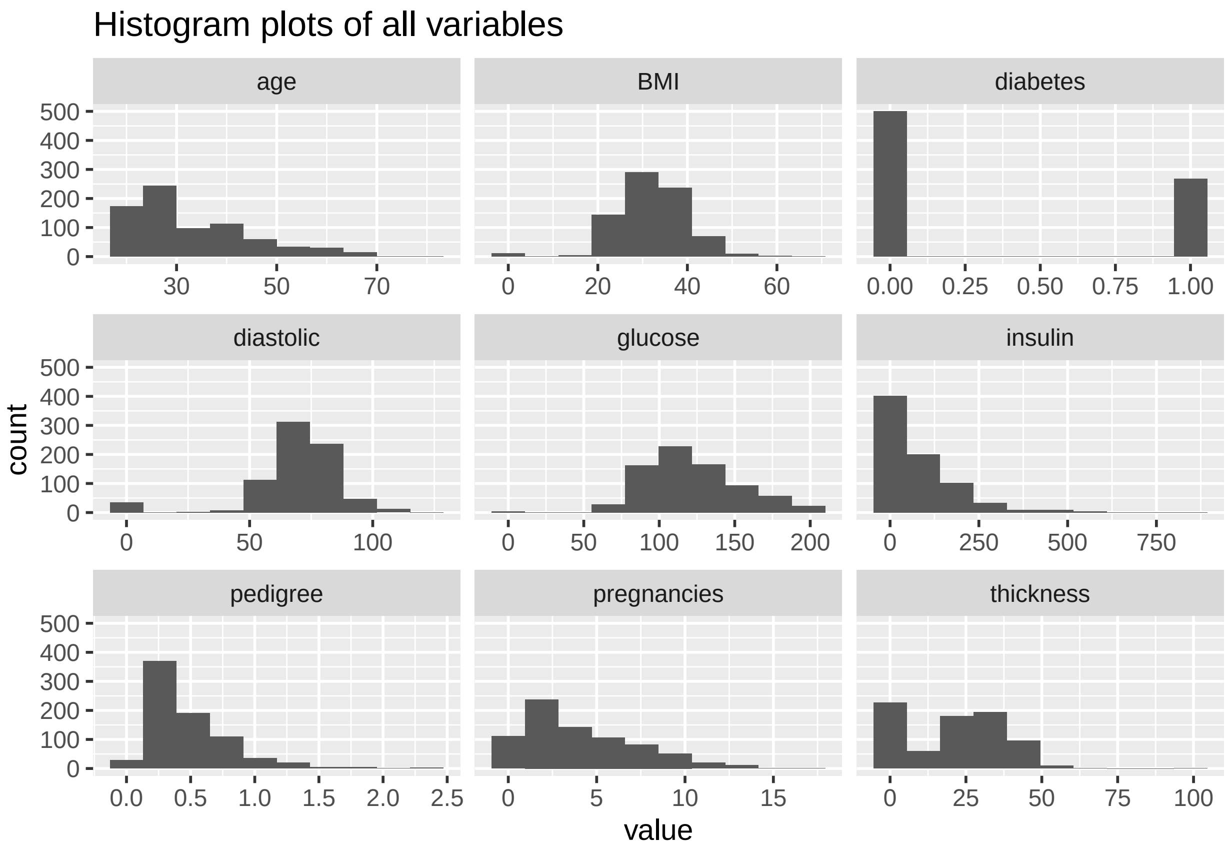


Figure 1: Figure 1

In the histograms for several of the output variables, we see many of them are indeed skewed to the right, possibly due to the zero values involved.

We can fix this issue by means of imputation, where for each column, each zero value will be replaced with a certain value given. In this case, the pregnancies, glucose, and diastolic outputs will have its zero values replaced with the means of the outputs. While the thickness, insulin, and BMI outputs will use the median values.

```
## [1] "Number of zeroes in each of the column variables: "
```

```
## pregnancies     glucose   diastolic   thickness     insulin         BMI
##           0           0           0           0           0           0
##    pedigree         age    diabetes
##           0           0         500
```

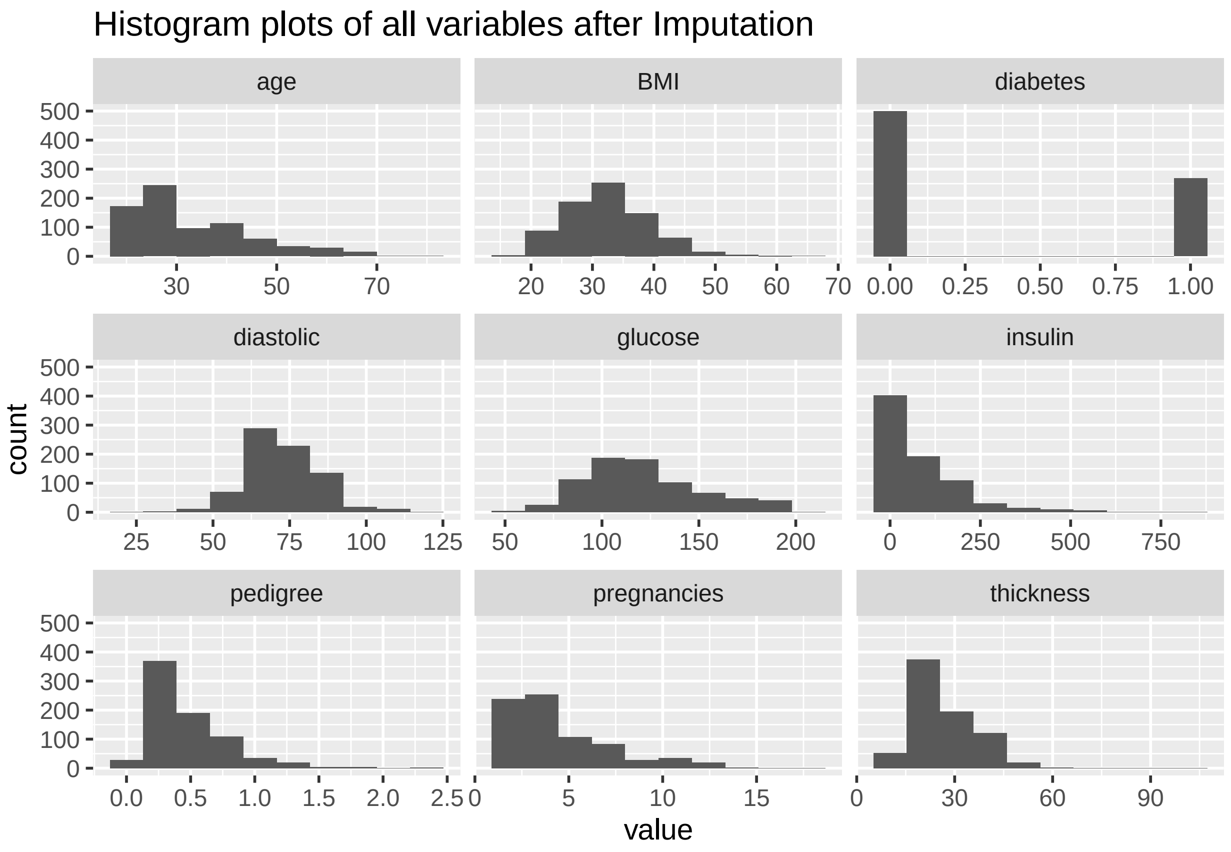


Figure 2: Figure 2

We can see now that all the zero values for each out of the output variables are out of the data, and the distributions for each of the output variables are noticeable less right-skewed.

Now we can start looking at the data and how it changes involving patients that have diabetes or not.

From the boxplots we can see those who are diabetic will tend to have higher levels in just about every variable, most notably glucose, age, and BMI.

In this pairs plot, there is separation of the data between those diagnosed with diabetes and ones that don't, but the separations aren't always exactly clear between each variable, with perhaps the glucose variable giving the least overlap. Also those who have diabetes tend to have a spread out distribution or density, than those that don't.

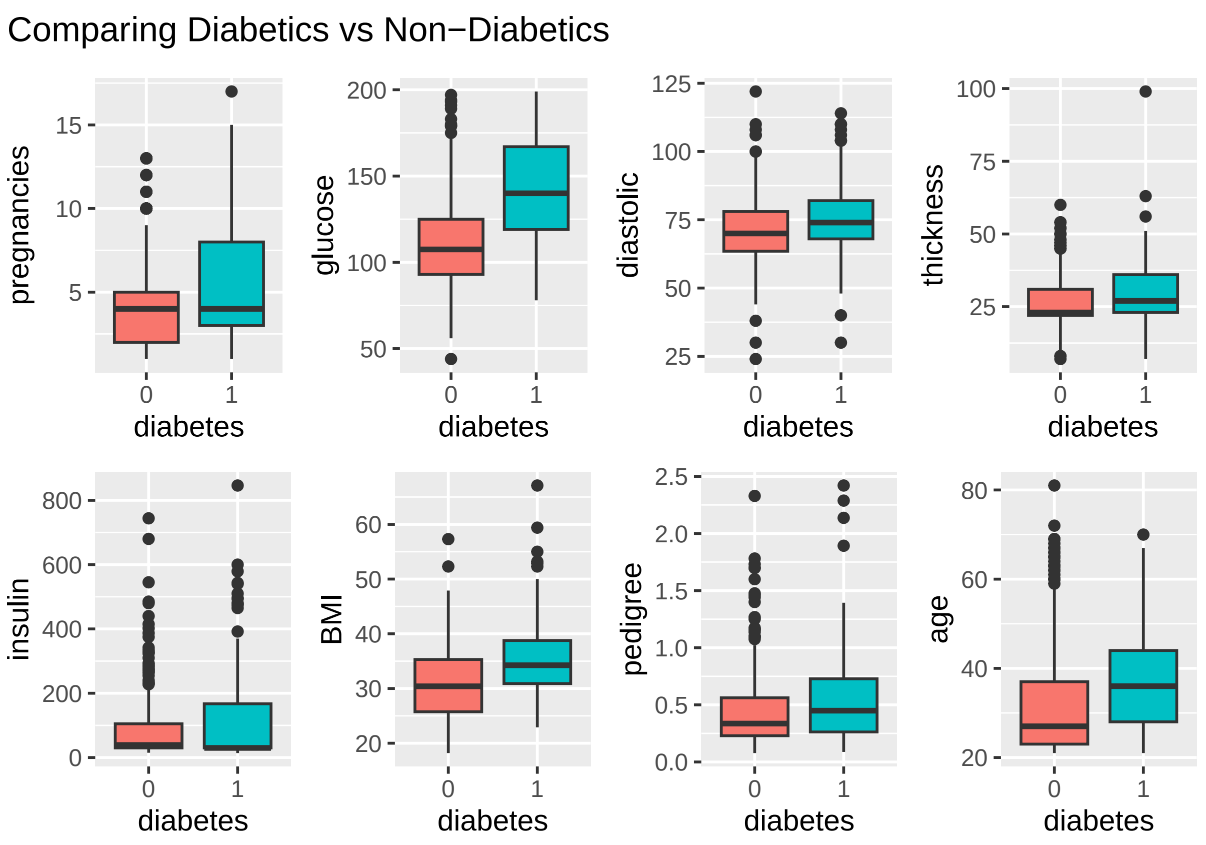


Figure 3: Figure 3

Pairs plots between the Output Variables

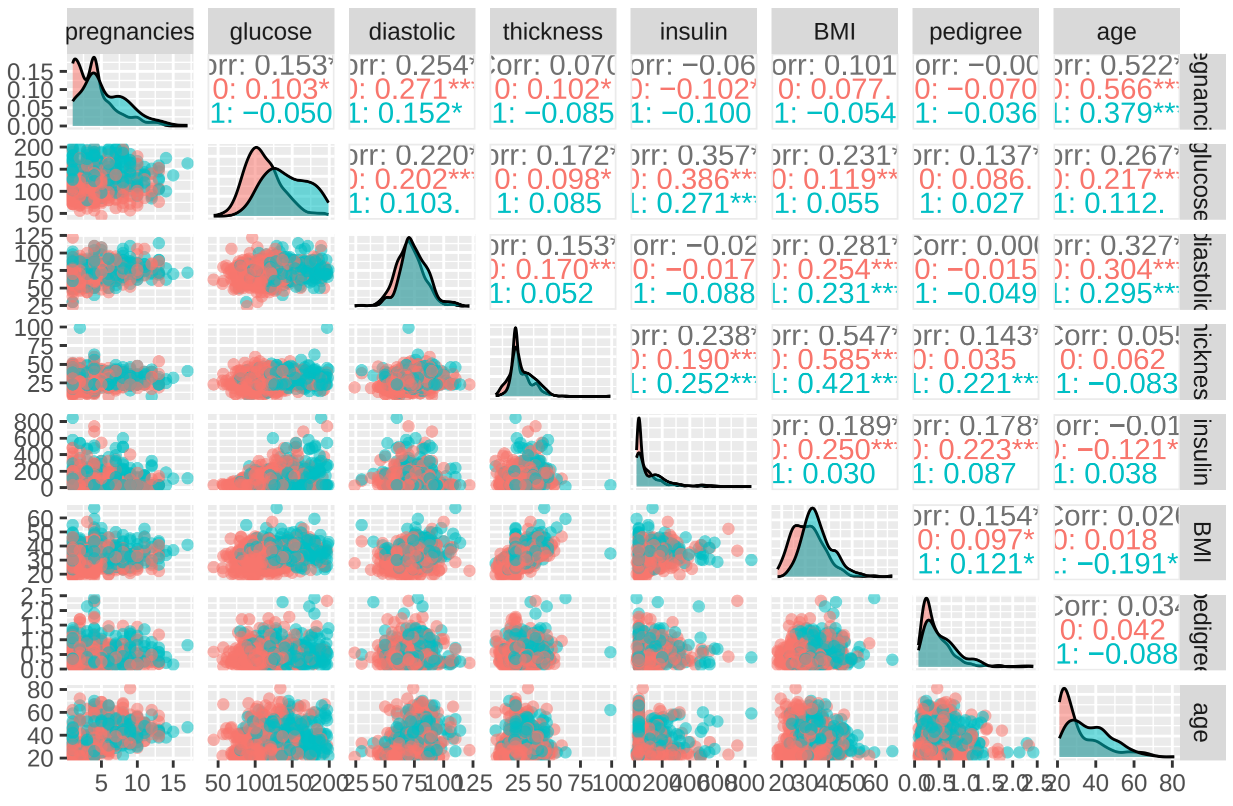
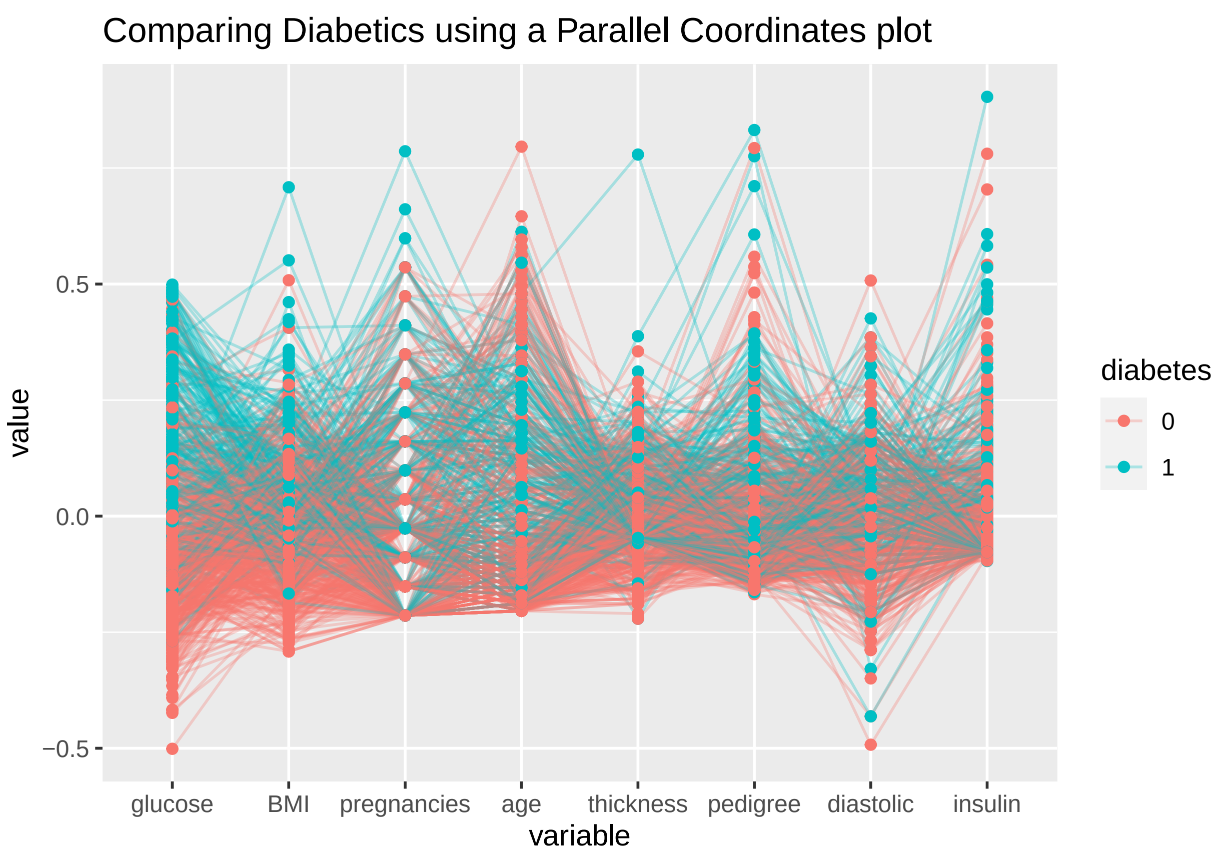


Figure 4: Figure 4



We can see more clearly diabetics have higher levels glucose, BMI, but tend to be in the mid range of pedigree and age.

It is also worth checking if there is collinearity involved between each of the output variables, as this can overinflate or deflate the predictive power of our models.

The correlation for all the variables remain relatively low, suggesting they can be all included when we start fitting our classification models.

Part 2: Classification of Diabetes

We will mainly use logistic regression, classification trees, and random forests to analyze the data, and compare their performances with each other.

To accurately assess the performance of our models, we should first split our data into two parts. The training data which we will make our model on, and the test data, for which we will validate the performance of our model on.

We reserve 80% for training, with the remaining 20% for testing.

```
## Dimensions of the Data by Row and Column : 615 9
```

```
## Dimensions of the Data by Row and Column : 153 9
```

Let us first start with a logistic regression model, we fit it on the output variables we have.

```
## [1] "Summary of the model: "
```

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

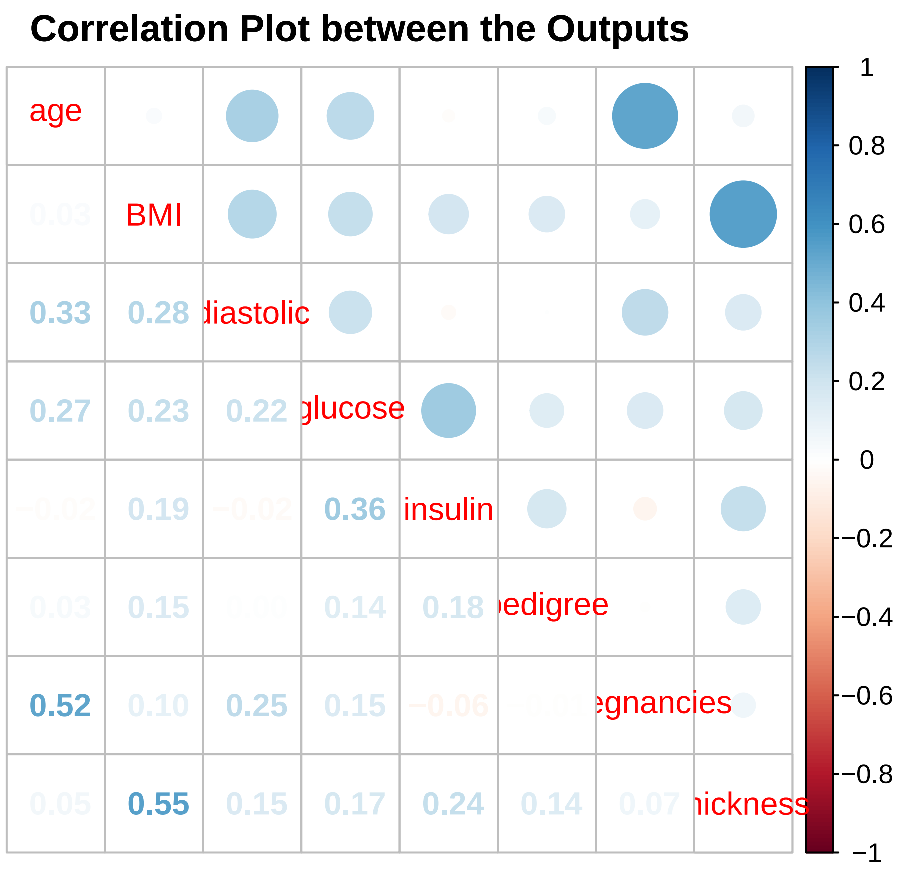


Figure 5: Figure 6

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5580  -0.7524  -0.4195   0.7223   2.3915
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.5596973  0.8981194  -9.531  < 2e-16 ***
## pregnancies  0.1184734  0.0402364   2.944  0.00324 **
## glucose      0.0383685  0.0042922   8.939  < 2e-16 ***
## diastolic   -0.0143739  0.0095896  -1.499  0.13390
## thickness   -0.0030798  0.0134546  -0.229  0.81895
## insulin     -0.0010905  0.0009855  -1.107  0.26850
## BMI          0.0886798  0.0198463   4.468 7.88e-06 ***
## pedigree     0.8741426  0.3397404   2.573  0.01008 *
## age          0.0137769  0.0101058   1.363  0.17280
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 796.05  on 614  degrees of freedom
## Residual deviance: 585.75  on 606  degrees of freedom
## AIC: 603.75
##
## Number of Fisher Scoring iterations: 5

## [1] "Pseudo R squared value (McFadden): "

## [1] 0.2641843

## [1] "Confusion Matrix: "

##
## predicted.classes  0  1
##                 0 90 21
##                 1 10 32

## [1] "Resulting Accuracy: "

## [1] 0.7973856
```

We get the resulting logit model with esimataed coefficients as above. The pseudo R squared is as 0.2641843, which according to McFadden, suggests the model fits the data well. From the confusion matrix, we get the predictive accuracy being 79.7%, which is fairly good.

Next, we fit a classification tree, once again using all the variables.

```
## [1] "Confusion Matrix: "

##
## predd  0  1
##     0 85 22
##     1 15 31

## [1] "Resulting Accuracy: "

## [1] 0.7581699
```

The accuracy we get is 75.8%, just a bit lower than logistic regression!

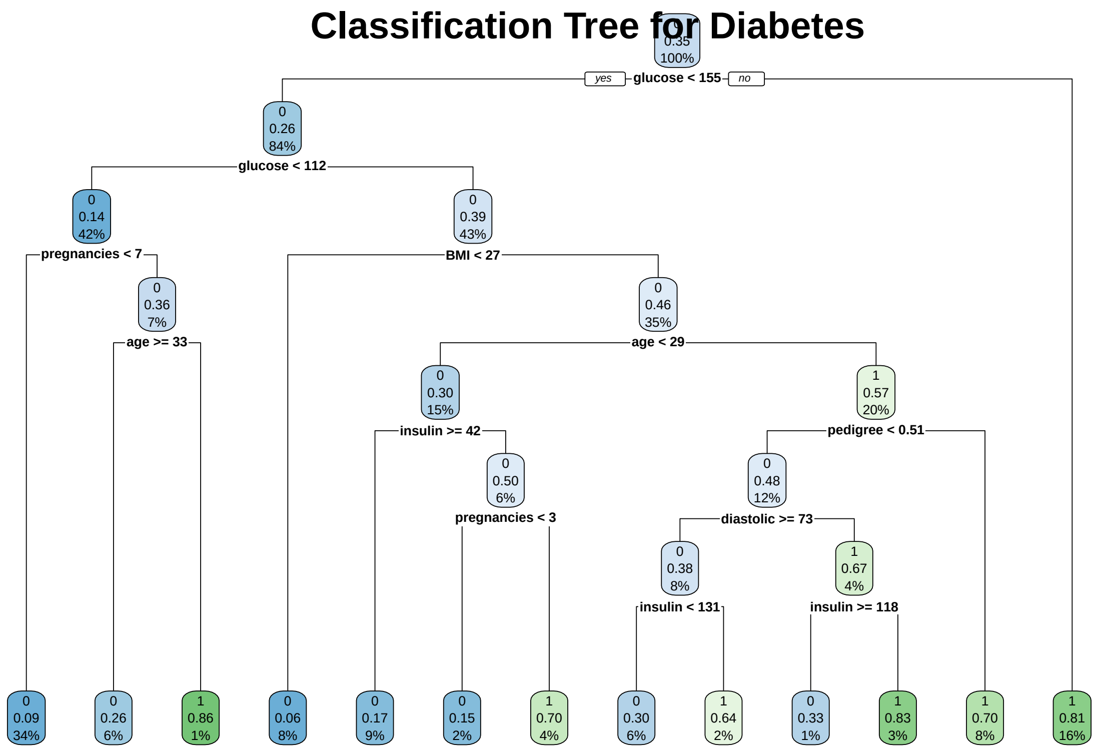


Figure 6: Figure 7

Finally, we fit a random forest:

```
## [1] "Random Forest Result: "
```

```
##
## Call:
##  randomForest(formula = diabetes ~ ., data = train.data, importance = T,      proximity = T)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 25.69%
## Confusion matrix:
##     0   1 class.error
## 0 339  61   0.1525000
## 1  97 118   0.4511628
```

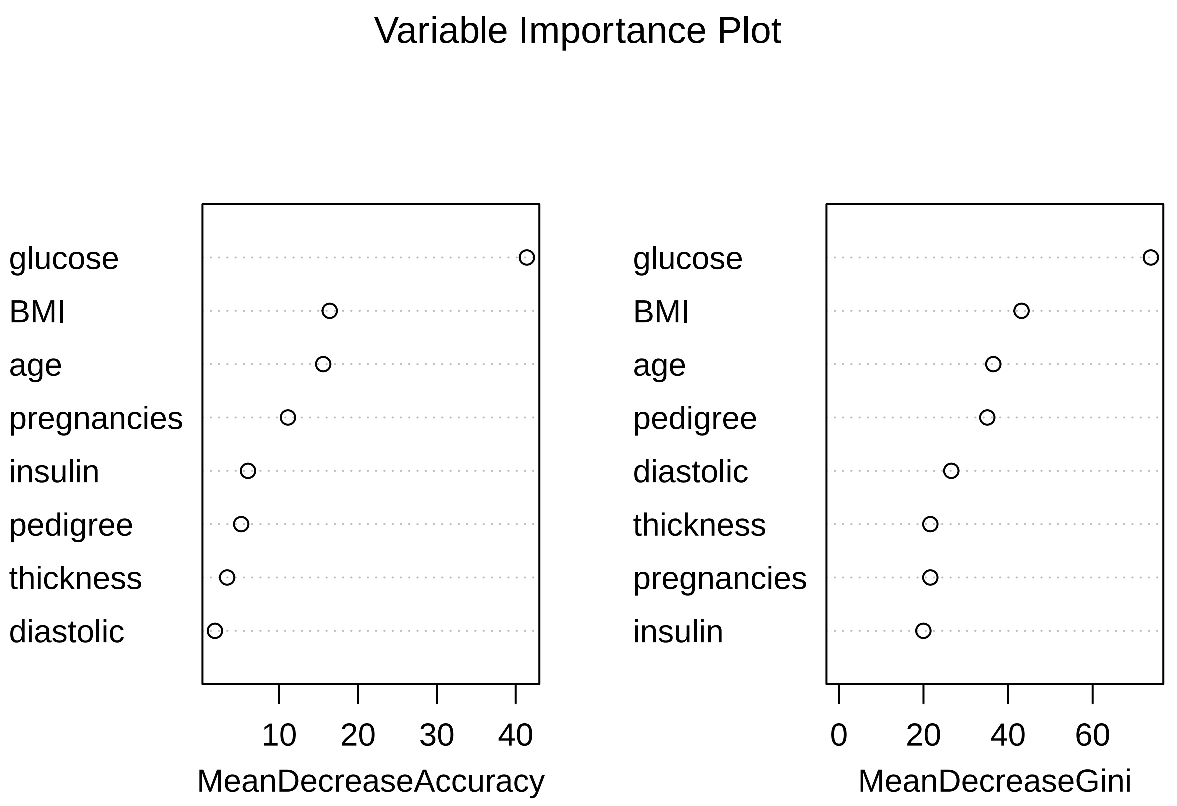
We see that this random forest has an accuracy of 339 + 113 / 615 = 0.73496, or 73.4% accuracy on the training data. Let's see how it performs on the test data.

```
## [1] "Random Forest Results on the Test Data: "
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 90 19
##          1 10 34
##
##                Accuracy : 0.8105
##                  95% CI : (0.7393, 0.8692)
##     No Information Rate : 0.6536
##     P-Value [Acc > NIR] : 1.46e-05
##
##                   Kappa : 0.564
##
##  Mcnemar's Test P-Value : 0.1374
##
##             Sensitivity : 0.9000
##             Specificity : 0.6415
##          Pos Pred Value : 0.8257
##          Neg Pred Value : 0.7727
##              Prevalence : 0.6536
##          Detection Rate : 0.5882
##    Detection Prevalence : 0.7124
##       Balanced Accuracy : 0.7708
##
##        'Positive' Class : 0
##
```

We see that the random forest performs the "best" out of all the models in terms of accuracy at 81.7%, which makes for the classification tree, since random forests are multiple trees combined.

We can also check which varies contributed most to the splitting of the trees



Variables with high mean decrease accuracy and gini are considered most significant, suggesting that the glucose contributes to the predicting process greatly, with age and BMI in second and third place.

Part 3: Possible Imbalance?

We have seen that the accuracy rates hover at around 75-80% accuracy rates, which could possibly be improved. This is an important thing to consider when dealing with medical data, so having a correct diagnosis is very important to have.

We recall that this data was supposedly imbalanced, that is, the number of observations between each class are not close to each other, one class has much more observations than the other. This can affect a model's predictive power since it will be trained on the majority class, and therefore is more used to that class in general when giving a diagnosis. This can very obviously lead to misclassification when it encounters an observation of the other class.

Let's start by comparing the number of observations between each class again.

```
## [1] "Patients with diabetes (1) or not (0): "
```

```
##
##   0   1
## 500 268
```

There are about twice as many 0s as there are 1s in the data. That is, twice as many patients diagnosed with diabetes. We see this is the case with the test and train data.

```
##
##   0   1
## 400 215
```

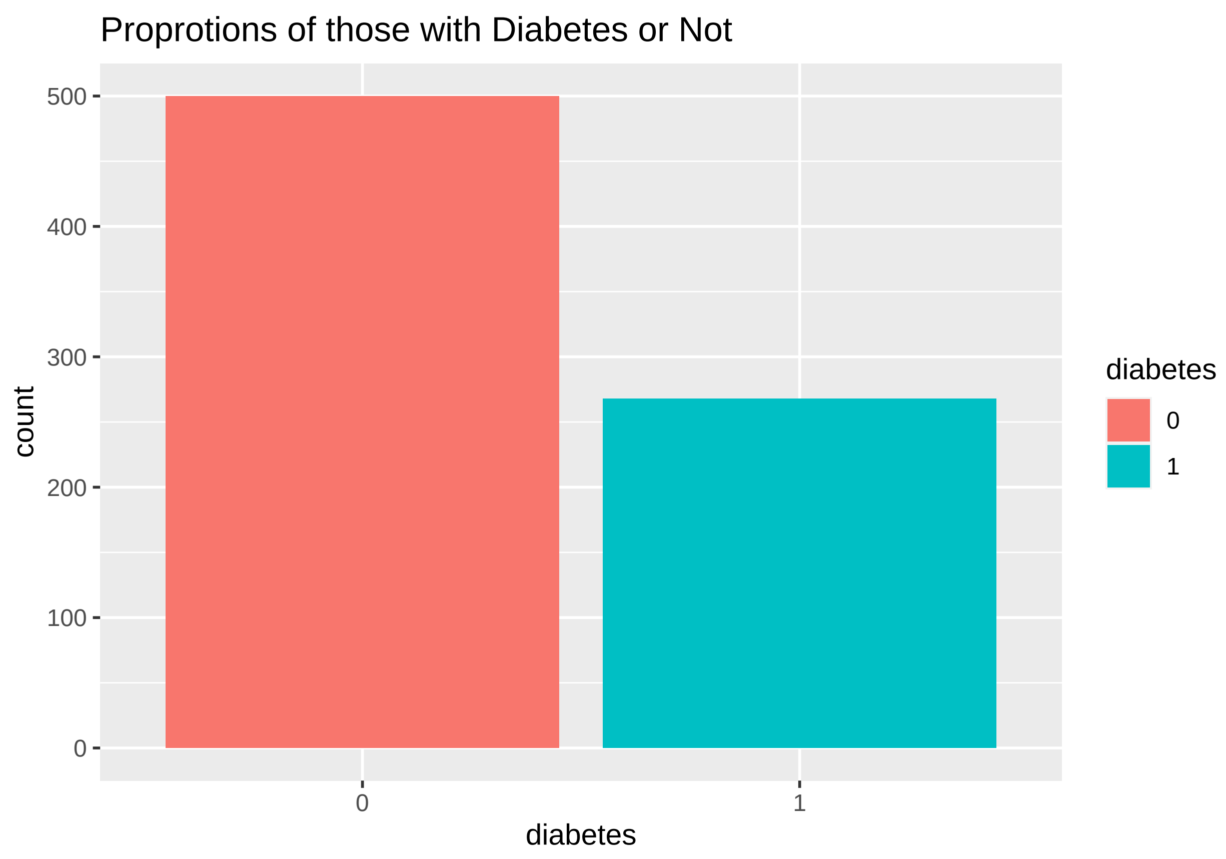


Figure 7: Figure 9

```
##
##   0   1
## 100  53
```

In all the data sets we've used, it is clear that number of 0s very much outweigh the 1s each time, which can be a problem with the training data, since we've obtained our models using the training, it will try to predict patients with no diabetes (0) more often than those that do (1). If we look back at at our confusion matrices, we can notice that the error rates for predicting those with diabetes was much higher at 30% or so, while error rates for predicting those without was much lower at 10-15%.

We might suspect that there is some sort of imbalance happening. In that case, we can try to overcome it by using oversampling and undersampling methods to balance out the classes out, and then we can try to fit a model on that data and test it again. We run the corresponding methods to get our re-sampled training data, and give the corresponding table counts for each.

```
## [1] "Oversampling count: "

##
##   0   1
## 400 300

## [1] "Undersampling count: "

##
##   0   1
## 215 215

## [1] "Using both sampling methods: "

##
##   0   1
## 247 253

## [1] "Using the ROSE method: "

##
##   0   1
## 317 333
```

Let's test out these re-balanced datasets on putting them through our previous models.

```
## [1] "Confusion Matrix (Over): "

##
## predicted.classes  0  1
##                 0 87 18
##                 1 13 35

## [1] "Resulting Accuracy: "

## [1] 0.7973856

## [1] "Confusion Matrix (Under): "

##
## predicted.classes  0  1
##                 0 82 13
##                 1 18 40

## [1] "Resulting Accuracy: "

## [1] 0.7973856

## [1] "Confusion Matrix (Both): "
```

```
##
## predicted.classes  0  1
##                 0 82 13
##                 1 18 40

## [1] "Resulting Accuracy: "

## [1] 0.7973856

## [1] "Confusion Matrix (ROSE): "

##
## predicted.classes  0  1
##                 0 82 12
##                 1 18 41

## [1] "Resulting Accuracy: "

## [1] 0.8039216

## [1] "Confusion Matrix (Over): "

##
## predd  0  1
##     0 86 17
##     1 14 36

## [1] "Resulting Accuracy: "

## [1] 0.7973856

## [1] "Confusion Matrix (Under): "

##
## predd  0  1
##     0 81 13
##     1 19 40

## [1] "Resulting Accuracy: "

## [1] 0.7908497

## [1] "Confusion Matrix (Both): "

##
## predd  0  1
##     0 80 10
##     1 20 43

## [1] "Resulting Accuracy: "

## [1] 0.8039216

## [1] "Confusion Matrix (Rose): "

##
## predd  0  1
##     0 74  8
##     1 26 45

## [1] "Resulting Accuracy: "

## [1] 0.7777778

## [1] "Results for Over: "
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 91 22
##          1  9 31
##
##                Accuracy : 0.7974
##                  95% CI : (0.7249, 0.858)
##     No Information Rate : 0.6536
##     P-Value [Acc > NIR] : 7.169e-05
##
##                   Kappa : 0.5252
##
##  Mcnemar's Test P-Value : 0.03114
##
##             Sensitivity : 0.9100
##             Specificity : 0.5849
##          Pos Pred Value : 0.8053
##          Neg Pred Value : 0.7750
##              Prevalence : 0.6536
##          Detection Rate : 0.5948
##    Detection Prevalence : 0.7386
##       Balanced Accuracy : 0.7475
##
##        'Positive' Class : 0
##
```

```
## [1] "Results for Under: "
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 79  9
##          1 21 44
##
##                Accuracy : 0.8039
##                  95% CI : (0.7321, 0.8636)
##     No Information Rate : 0.6536
##     P-Value [Acc > NIR] : 3.3e-05
##
##                   Kappa : 0.5889
##
##  Mcnemar's Test P-Value : 0.04461
##
##             Sensitivity : 0.7900
##             Specificity : 0.8302
##          Pos Pred Value : 0.8977
##          Neg Pred Value : 0.6769
##              Prevalence : 0.6536
##          Detection Rate : 0.5163
##    Detection Prevalence : 0.5752
##       Balanced Accuracy : 0.8101
##
```

```
##        'Positive' Class : 0
##
## [1] "Results for Both: "
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 78 12
##          1 22 41
##
##                Accuracy : 0.7778
##                  95% CI : (0.7036, 0.8409)
##     No Information Rate : 0.6536
##     P-Value [Acc > NIR] : 0.000586
##
##                   Kappa : 0.5301
##
##  Mcnemar's Test P-Value : 0.122713
##
##             Sensitivity : 0.7800
##             Specificity : 0.7736
##          Pos Pred Value : 0.8667
##          Neg Pred Value : 0.6508
##              Prevalence : 0.6536
##          Detection Rate : 0.5098
##    Detection Prevalence : 0.5882
##       Balanced Accuracy : 0.7768
##
##        'Positive' Class : 0
##
## [1] "Results for Rose: "
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 86 15
##          1 14 38
##
##                Accuracy : 0.8105
##                  95% CI : (0.7393, 0.8692)
##     No Information Rate : 0.6536
##     P-Value [Acc > NIR] : 1.46e-05
##
##                   Kappa : 0.5796
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8600
##             Specificity : 0.7170
##          Pos Pred Value : 0.8515
##          Neg Pred Value : 0.7308
##              Prevalence : 0.6536
##          Detection Rate : 0.5621
##    Detection Prevalence : 0.6601
##       Balanced Accuracy : 0.7885
##
##        'Positive' Class : 0
##
```

Looking at our results overall, we see predictions and accuracy have improved by a not huge margin. In fact our accuracy for the random forests seemed to have went down! It is worth noting that the prediction for just the patients with diabetes has notably improved, especially when using the ROSE method. But it may have also increased the error rate for the classifying the non-diabetic ones! Overall, resampling to allow for balanced classes does help out our models.

Our final method to deal with imbalancing and improving accuracy is using ensembling methods.

```
## [1] "Ensemble results for Original Training: "

##
## preds  0  1
##     0 89 18
##     1 11 35

## [1] 0.8104575

## [1] "Ensemble results for Over: "

##
## preds  0  1
##     0 87 21
##     1 13 32

## [1] 0.7777778

## [1] "Ensemble results for Under: "

##
## preds  0  1
##     0 77 10
##     1 23 43

## [1] 0.7843137

## [1] "Ensemble results for Both: "

##
## preds  0  1
##     0 77 12
##     1 23 41

## [1] 0.7712418

## [1] "Ensemble results for ROSE: "

##
## preds  0  1
##     0 87 15
##     1 13 38

## [1] 0.8169935
```

Once again, the accuracy rates have only gone up by a little, which given how well the models did already may not be so surprising. What does seem to change is that error rates for each of the classes, with the error rates for the class 1 prediction is definitely better, but it seems to come at a cost for the class 0 prediction, being most noticeable when using undersampling and both methods. Using the ROSE method seems to have obtained the best accuracy possible.

Conclusion: Given that the accuracy rates have not improved in the two methods we've done, this might suggest there is no evidence of imbalance going on with the two classes. In context, this might make more sense. In the general population, there is certainly far less people with diabetes than those that don't.

Part 4: Clustering

The reason our predictions may not have been as good as we want it to be could be linked spefically to the data. While those who are diagnosed with diabetes often exhitbit similar traits in the variables, high glucose or BMI, it may be the case that there are patients who are experiencing symptoms of diabetes, but are not actually diagnosed with such. This could explain why our models have not properly classified all our patients properly. We might consider there is a 3rd group of patients inside our data, known as pre-diabetics. Clustering can help us determine if this is the case.

We first look at only our output variables and scale them.

```
##      pregnancies    glucose   diastolic  thickness      insulin        BMI
## [1,]   0.5290497  0.8646901 -0.02063182  0.8305724 -0.609378480  0.1671312
## [2,]  -1.1485316 -1.2052271 -0.51579562  0.1804488 -0.609378480 -0.8509963
## [3,]   1.2000823  2.0146442 -0.68085022 -0.4696748 -0.609378480 -1.3309707
## [4,]  -1.1485316 -1.0738038 -0.51579562 -0.4696748 -0.003868209 -0.6328261
## [5,]  -0.1419828  0.5032760 -2.66150540  0.8305724  0.696253042  1.5488758
## [6,]   0.1935335 -0.1866964  0.14442277 -0.4696748 -0.609378480 -0.9964431
##        pedigree         age
## [1,]  0.4681869  1.42506672
## [2,] -0.3648230 -0.19054773
## [3,]  0.6040037 -0.10551539
## [4,] -0.9201630 -1.04087112
## [5,]  5.4813370 -0.02048305
## [6,] -0.8175458 -0.27558007
```

We perform a k-means clustering on our data with 2 clusters for now.

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
## [1] "Cluster 1 Data: "
```

```
##    pregnancies    glucose  diastolic  thickness      insulin        BMI
## 2   -1.1485316 -1.2052271 -0.5157956  0.1804488 -0.609378480 -0.8509963
## 4   -1.1485316 -1.0738038 -0.5157956 -0.4696748 -0.003868209 -0.6328261
## 6    0.1935335 -0.1866964  0.1444228 -0.4696748 -0.609378480 -0.9964431
## 7   -0.4774991 -1.4352179 -1.8362324  0.5055106 -0.060634797 -0.2110304
## 8    1.8711148 -0.2195522 -0.2682137 -0.4696748 -0.609378480  0.4143908
## 11  -0.1419828 -0.3838314  1.6299142 -0.4696748 -0.609378480  0.7489184
##      pedigree        age fit.cluster
## 2  -0.3648230 -0.1905477           1
## 4  -0.9201630 -1.0408711           1
## 6  -0.8175458 -0.2755801           1
## 7  -0.6756927 -0.6157094           1
## 8  -1.0197620 -0.3606124           1
## 11 -0.8477273 -0.2755801           1
```

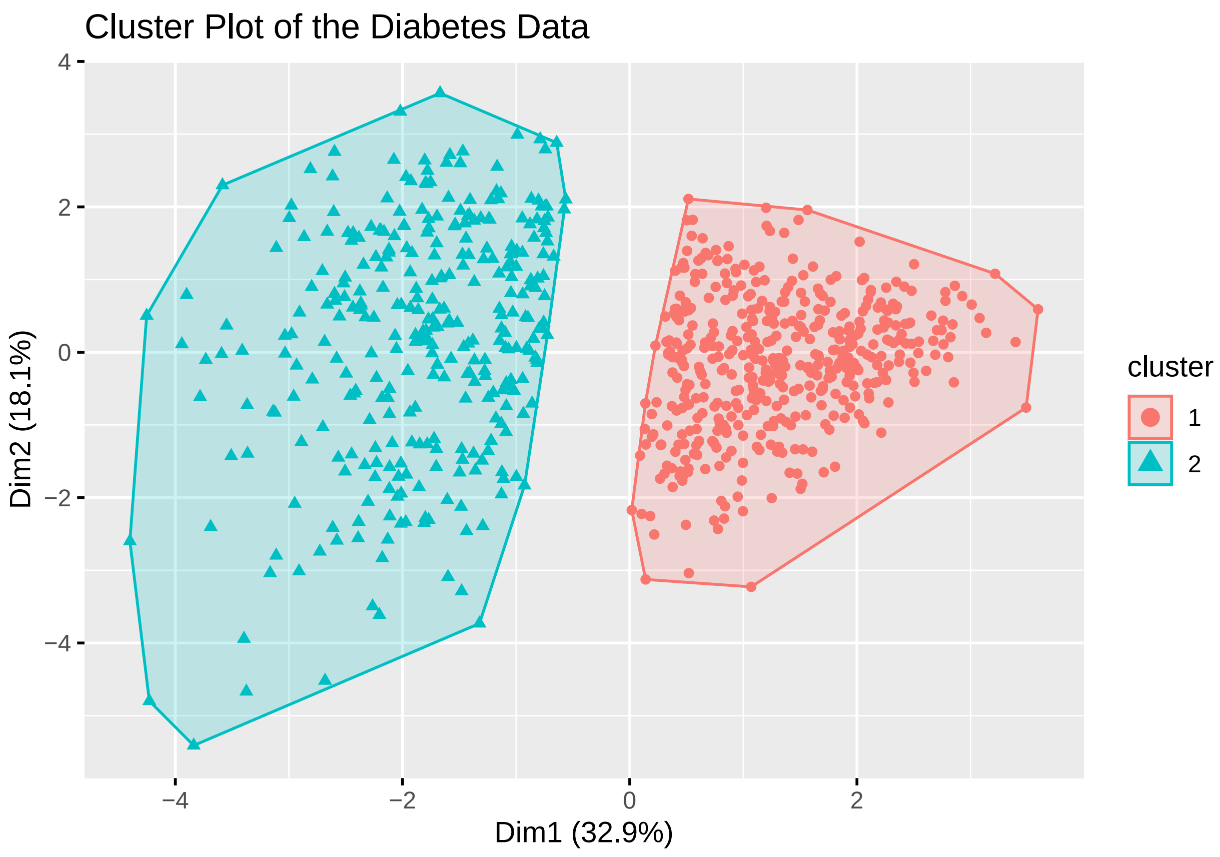
```
## [1] "Cluster 2 Data: "
```

```
##    pregnancies   glucose   diastolic  thickness    insulin         BMI
## 1    0.5290497 0.8646901 -0.02063182  0.8305724 -0.6093785  0.16713124
## 3    1.2000823 2.0146442 -0.68085022 -0.4696748 -0.6093785 -1.33097072
## 5   -0.1419828 0.5032760 -2.66150540  0.8305724  0.6962530  1.54887577
## 9   -0.8130153 2.4746258 -0.18568642  1.9141117  4.2441648 -0.28375381
## 10   1.2000823 0.1090061  1.96002336 -0.4696748 -0.6093785 -0.06558363
## 12   1.8711148 1.5218067  0.14442277 -0.4696748 -0.6093785  0.80709713
##      pedigree         age fit.cluster
## 1   0.4681869  1.42506672           2
## 3   0.6040037 -0.10551539           2
## 5   5.4813370 -0.02048305           2
## 9  -0.9473263  1.68016374           2
## 10 -0.7239831  1.76519608           2
## 12  0.1965532  0.06454929           2
```

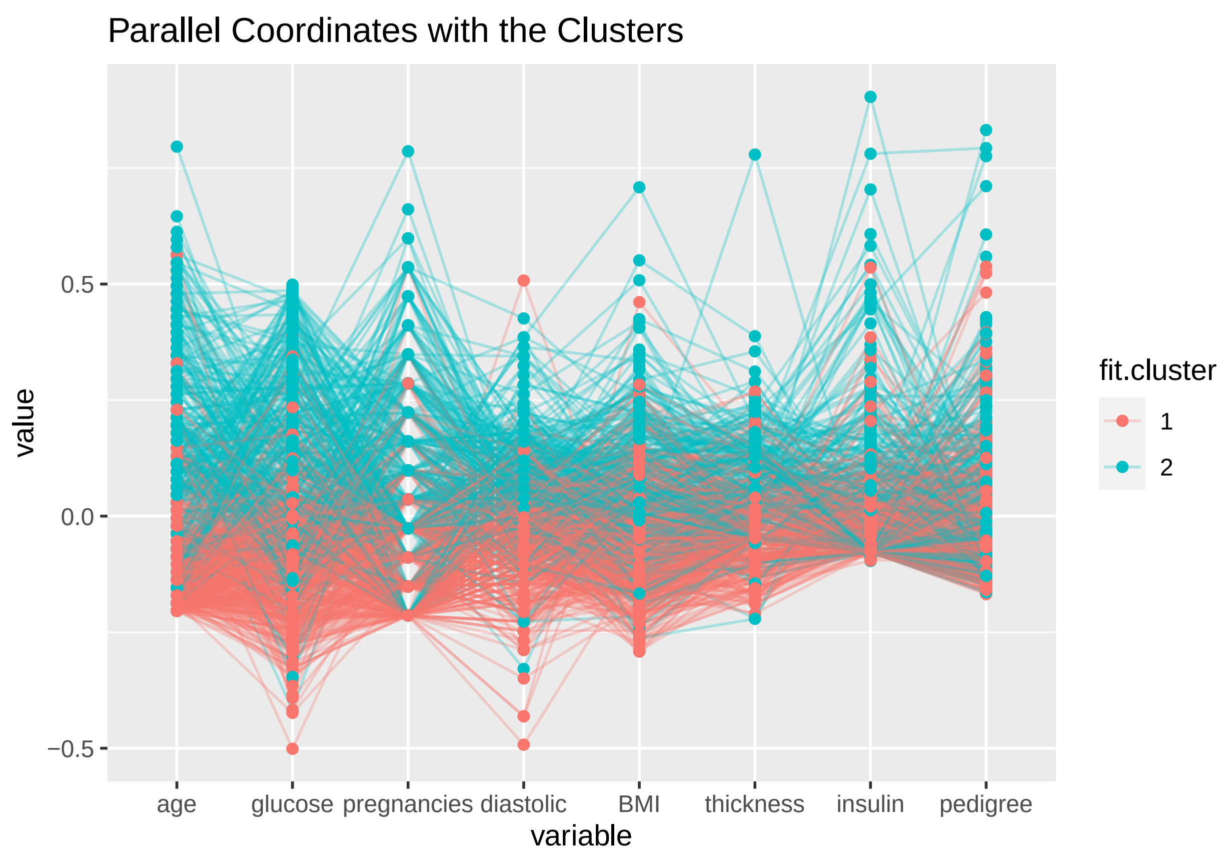
Looking at the means for each cluster:

```
##   Group.1 pregnancies    glucose  diastolic  thickness    insulin        BMI
## 1       1  -0.4248396 -0.4640556 -0.3962606 -0.2913524 -0.1913083 -0.3162021
## 2       2   0.5884424  0.6427602  0.5488578  0.4035502  0.2649798  0.4379694
##     pedigree        age
## 1 -0.1377917 -0.5163779
## 2  0.1908543  0.7152315
```

We see the values seem to be overall higher in magnitude in the first cluster than in the second, suggesting that this cluster is representing patients that have diabetes, or at least those with symptoms of diabetes. Notably insulin, age, and BMI are higher, which are often signs of diabetes.



In visualizing the clusters overlaying the data, we see that they cleanly separate the data into two groups, with group 1 containing the supposed "diabetic" patients, while group 2 do not. But it could be possible that there are non-diabetics in group 1 with diabetic like sypmptoms.



Comparing to Figure 5, we see near similarities in the plots, cluster 1 being almost identical to the diabetes class. A quick comparison of means below suggest they are not exact.

```
## [1] "Mean Comparison in Insulin (Cluster 1 vs Class 1): "

## [1] 74.18834

## [1] 115.7836

## [1] "Mean Comparison in BMI (Cluster 1 vs Class 1): "

## [1] 30.27691

## [1] 35.38134

## [1] "Mean Comparison in Glucose (Cluster 1 vs Class 1): "

## [1] 107.5583

## [1] 142.1604
```

Let's see what happens if we try to fit 3 clusters instead.

```
##   Group.1 pregnancies    glucose  diastolic   thickness    insulin         BMI
## 1       1  -0.5126220 -0.4631198 -0.4214462 -0.44489966 -0.2640875 -0.46686451
## 2       2  -0.2798944  0.7092801  0.1540425  1.07985292  1.0879309  0.93880232
## 3       3   0.9574569  0.2169415  0.5284134 -0.05998947 -0.3366486  0.06796298
##      pedigree        age
## 1 -0.17203547 -0.6161779
## 2  0.45231843 -0.2208046
## 3 -0.04660756  1.0729907

## [1] "Cluster 1 Data: "
##    pregnancies    glucose  diastolic  thickness      insulin        BMI
## 2   -1.1485316 -1.2052271 -0.5157956  0.1804488 -0.609378480 -0.8509963
## 4   -1.1485316 -1.0738038 -0.5157956 -0.4696748 -0.003868209 -0.6328261
## 6    0.1935335 -0.1866964  0.1444228 -0.4696748 -0.609378480 -0.9964431
## 7   -0.4774991 -1.4352179 -1.8362324  0.5055106 -0.060634797 -0.2110304
## 16   0.8645660 -0.7123897 -0.2682137 -0.4696748 -0.609378480 -0.3564772
## 18   0.8645660 -0.4823989  0.1444228 -0.4696748 -0.609378480 -0.4146559
##       pedigree        age fit.cluster
## 2  -0.36482303 -0.1905477           1
## 4  -0.92016296 -1.0408711           1
## 6  -0.81754580 -0.2755801           1
## 7  -0.67569267 -0.6157094           1
## 16  0.03659116 -0.1055154           1
## 18 -0.65758376 -0.1905477           1
```

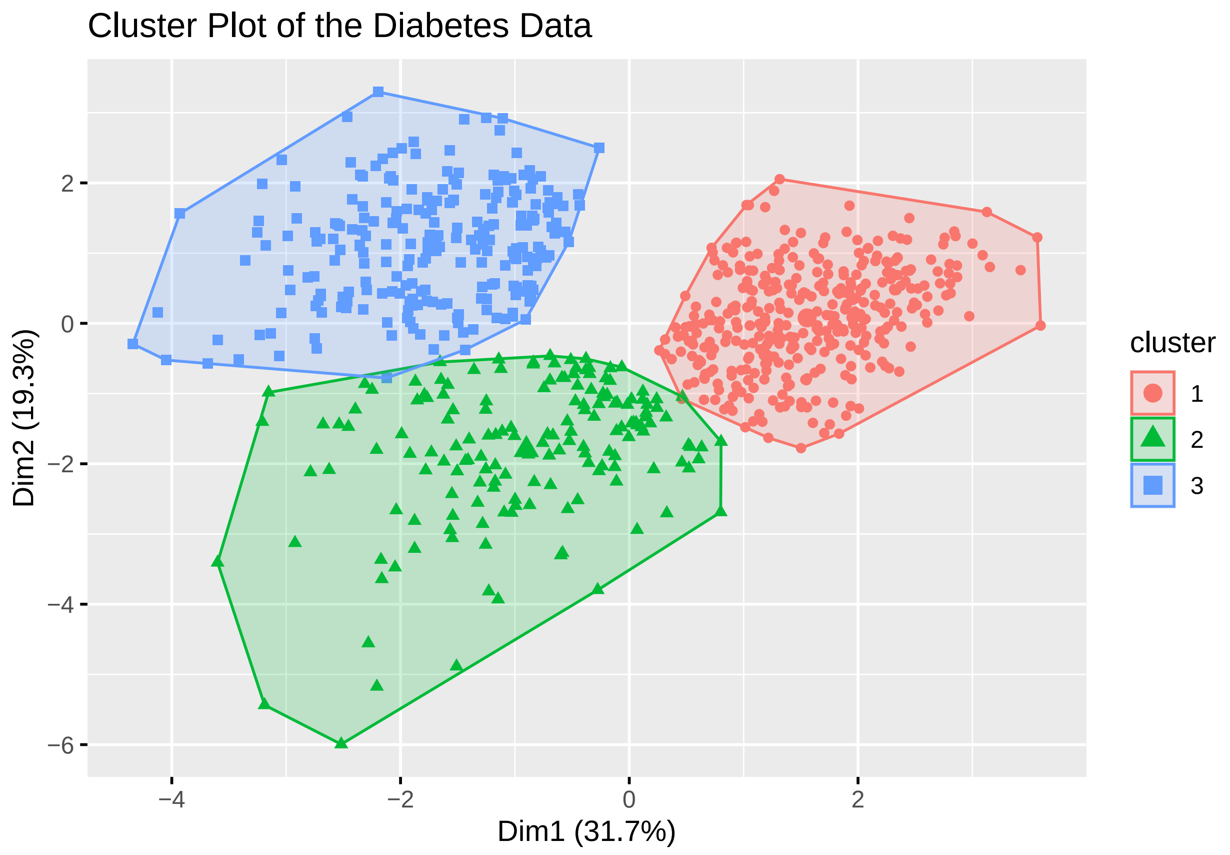
```
## [1] "Cluster 2 Data: "
```

```
##    pregnancies    glucose  diastolic  thickness  insulin        BMI   pedigree
## 5   -0.1419828  0.5032760 -2.6615054  0.8305724 0.696253  1.5488758  5.4813370
## 9   -0.8130153  2.4746258 -0.1856864  1.9141117 4.244165 -0.2837538 -0.9473263
## 14  -1.1485316  2.2117792 -1.0109594 -0.4696748 7.110877 -0.3419325 -0.2229699
## 17  -0.1419828 -0.1209847  0.9696958  2.1308196 1.282841  1.9415821  0.2388073
## 21  -0.4774991  0.1418619  1.2998050  1.4806960 1.330147  0.9961780  0.7005846
## 32  -0.4774991  1.1932484  0.3094774  0.9389263 1.424758 -0.1237623  1.1442529
##            age fit.cluster
## 5  -0.02048305           2
## 9   1.68016374           2
## 14  2.19035777           2
## 17 -0.19054773           2
## 21 -0.53067709           2
## 32 -0.44564475           2
```

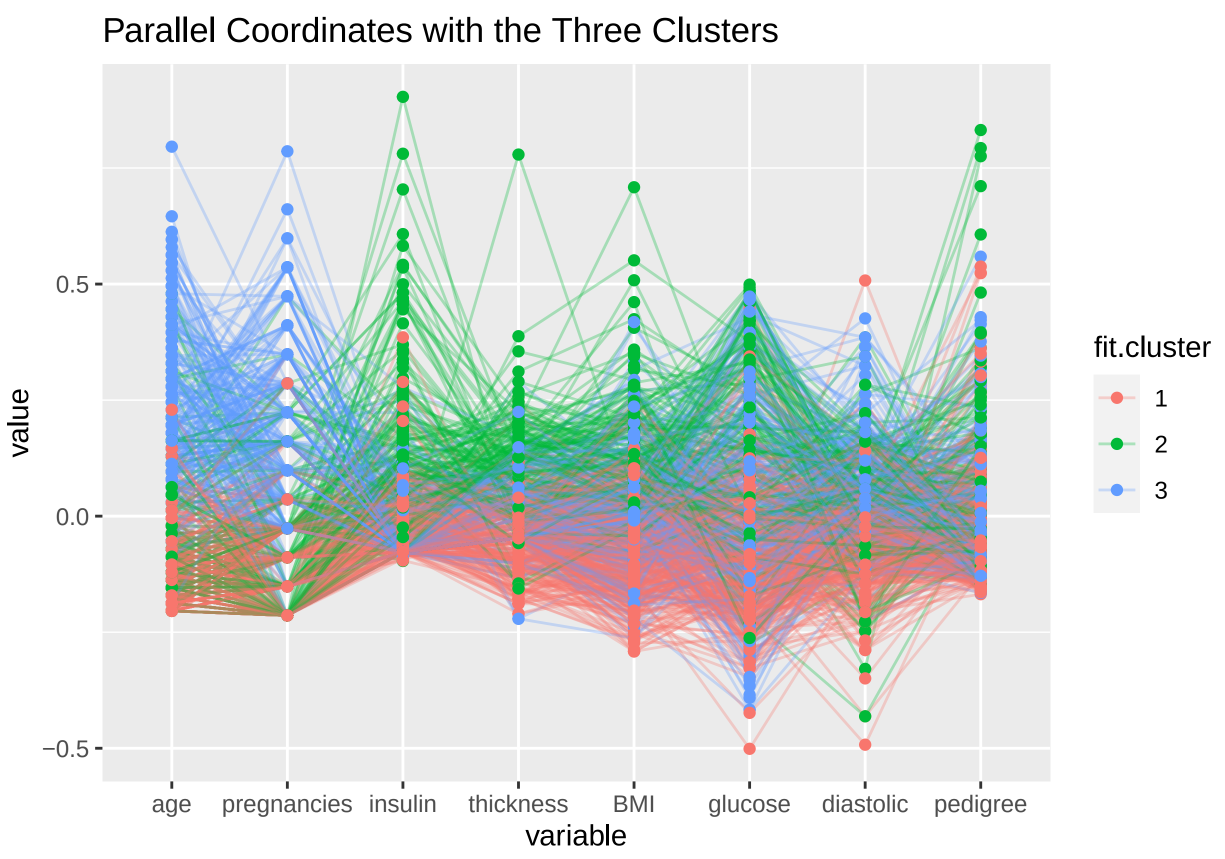
```
## [1] "Cluster 3 Data: "
```

```
##    pregnancies    glucose   diastolic  thickness    insulin         BMI
## 1    0.5290497  0.8646901 -0.02063182  0.8305724 -0.6093785  0.16713124
## 3    1.2000823  2.0146442 -0.68085022 -0.4696748 -0.6093785 -1.33097072
## 8    1.8711148 -0.2195522 -0.26821372 -0.4696748 -0.6093785  0.41439079
## 10   1.2000823  0.1090061  1.96002336 -0.4696748 -0.6093785 -0.06558363
## 11  -0.1419828 -0.3838314  1.62991416 -0.4696748 -0.6093785  0.74891841
## 12   1.8711148  1.5218067  0.14442277 -0.4696748 -0.6093785  0.80709713
##      pedigree         age fit.cluster
## 1   0.4681869  1.42506672           3
## 3   0.6040037 -0.10551539           3
## 8  -1.0197620 -0.36061241           3
## 10 -0.7239831  1.76519608           3
## 11 -0.8477273 -0.27558007           3
## 12  0.1965532  0.06454929           3
```

From the means once again, we can infer that group 2 could be the non-diabetic group of patients, based on the sign and magnitude of the values with respect to the other clusters. Based on the high insulin and glucose levels in group 1, this suggests this cluster is the group with diabetes and are diaganosed with such. For the third group, glucose isn't as high as it is in group 1, the insulin value suggests these patients have yet to take any, and finally BMI is also more elevated than in group 2. Cluster 3 could represent pre-diabetic patients, patients who are starting to experience diabetic symptoms such as elevated glucose, or patients who have been diagnosed, but have yet to receive the insulin treatment need to deal with the diabetes. Also those that are experiencing this symptoms tend be those experiencing increased changes in their weight, and thus their BMI.



The three clusters have cleanly separated the data into the three groups, and each group does have a sizable number of data points in them, so there is good chance there will be many in the 3rd group that have been diagnosed diabetes and just as many that have not.



In the plot above, cluster 3 seems to capture the lines that don't rise in values by a large amount, similar to the non-diabetic group. Cluster 1 which was assumed to be like the diabetic group, still captures most of the higher values in the variables.

In the summary mean above, we do a noticable different between insulin levels, suggesting our statement might be true.

```
## [1] "Mean Comparison in Insulin (Cluster 1 vs Class 1): "

## [1] 66.49587

## [1] 115.7836

## [1] "Mean Comparison in BMI (Cluster 1 vs Class 1): "

## [1] 29.24105

## [1] 35.38134

## [1] "Mean Comparison in Glucose (Cluster 1 vs Class 1): "

## [1] 107.5868

## [1] 142.1604

## [1] "Comparison of Clusters vs Diabetes count: "

##
##       0   1
##   1 304  59
##   2  71  92
##   3 125 117
```

In the first cluster ie first row, we see the majority of the patients do have diabetes, while in the second cluster has the majority consist of non-diabetic patients as was claimed previously, finally the third cluster has even mix of both either patients, suggesting there are indeed a lot of patients experiencing symptoms but are not necessarily getting diagnosed with diabetes.

Conclusion: Based on the clustering above, there is evidence of a pre-diabetic group of patients in the data, and the reason for our models not being to able to better predict patients that may or may not have diabetes could be the sizable existence of such group, classifying those who may have some of the diabetic symptoms as such even if they may not actually yet caught it.