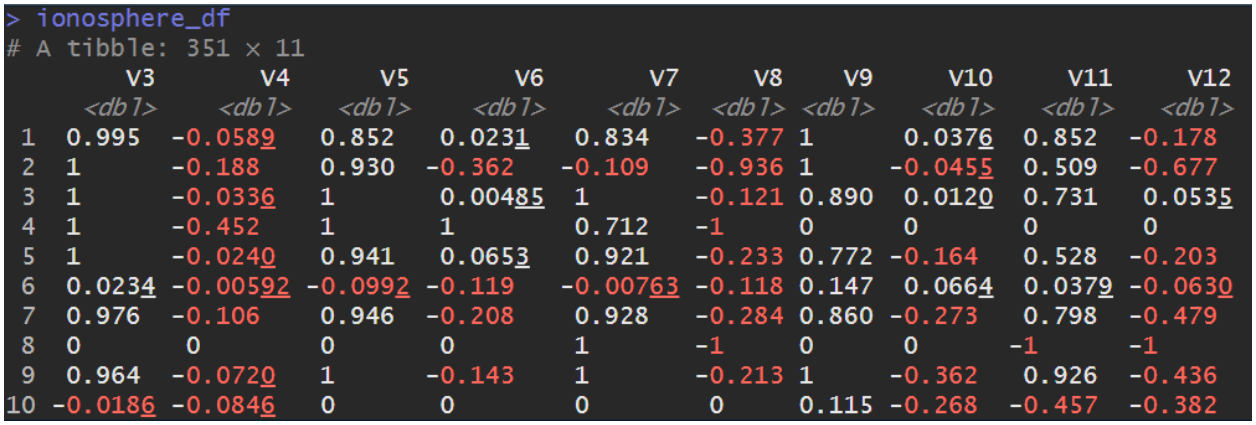
STAT201 Assignment 8

- Joshua Boryer 41497475

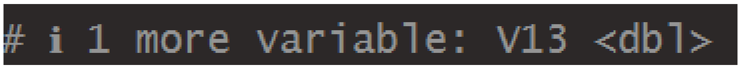
Load the data, print the structure. Check that all variables are numerical. Scale the data and plot the withiness against the number of clusters. How many clusters would you choose for a k-mean clustering analysis?

1a)

Are all variables numerical?



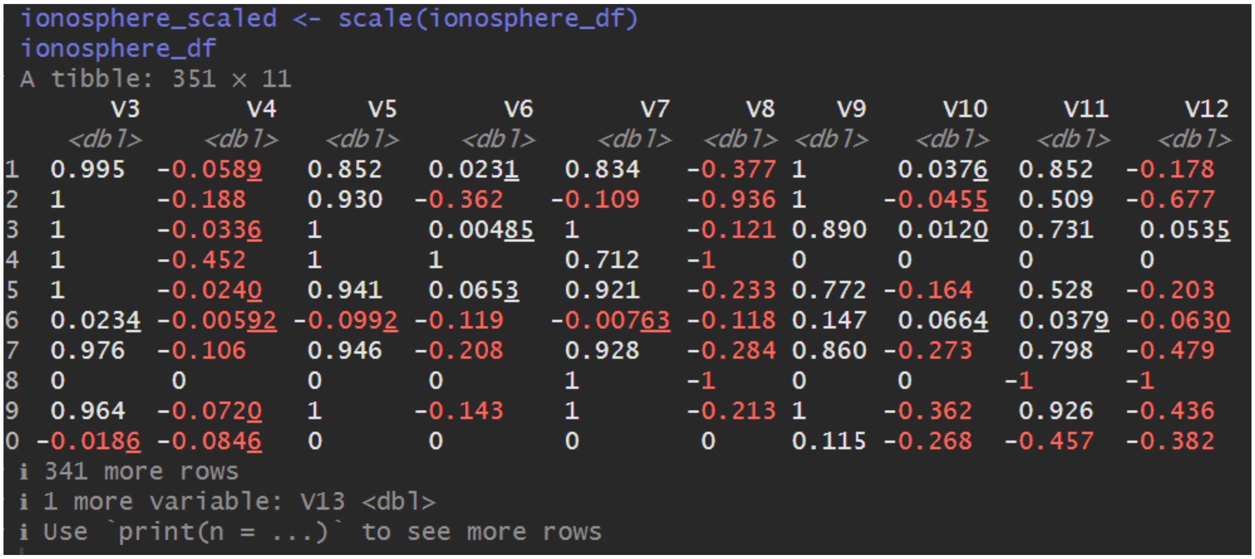
```
> ionosphere_df
# A tibble: 351 × 11
        V3       V4      V5       V6       V7     V8    V9      V10     V11     V12
     <dbl>    <dbl>   <dbl>    <dbl>    <dbl>  <dbl> <dbl>    <dbl>   <dbl>   <dbl>
 1  0.995  -0.0589   0.852   0.0231   0.834   -0.377 1       0.0376   0.852  -0.178
 2  1      -0.188    0.930  -0.362   -0.109   -0.936 1      -0.0455   0.509  -0.677
 3  1      -0.0336   1       0.00485  1       -0.121 0.890   0.0120   0.731   0.0535
 4  1      -0.452    1       1        0.712   -1     0       0        0       0
 5  1      -0.0240   0.941   0.0653   0.921   -0.233 0.772  -0.164    0.528  -0.203
 6  0.0234 -0.00592 -0.0992 -0.119   -0.00763 -0.118 0.147   0.0664   0.0379 -0.0630
 7  0.976  -0.106    0.946  -0.208    0.928   -0.284 0.860  -0.273    0.798  -0.479
 8  0       0        0       0        1       -1     0       0       -1      -1
 9  0.964  -0.0720   1      -0.143    1       -0.213 1      -0.362    0.926  -0.436
10 -0.0186 -0.0846   0       0        0        0     0.115  -0.268   -0.457  -0.382
```



```
# ℹ 1 more variable: V13 <dbl>
```

- All variables are represented as numerical floating point number variables <dbl>

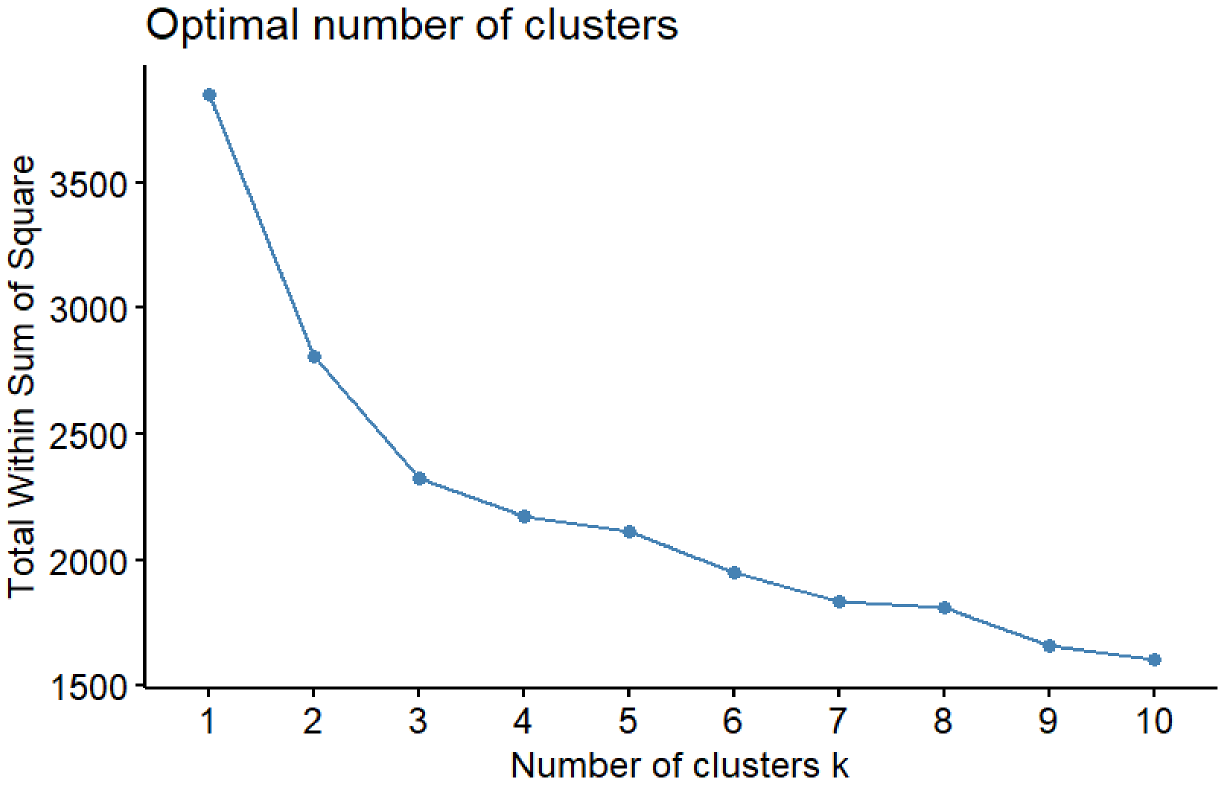
Scale the data:



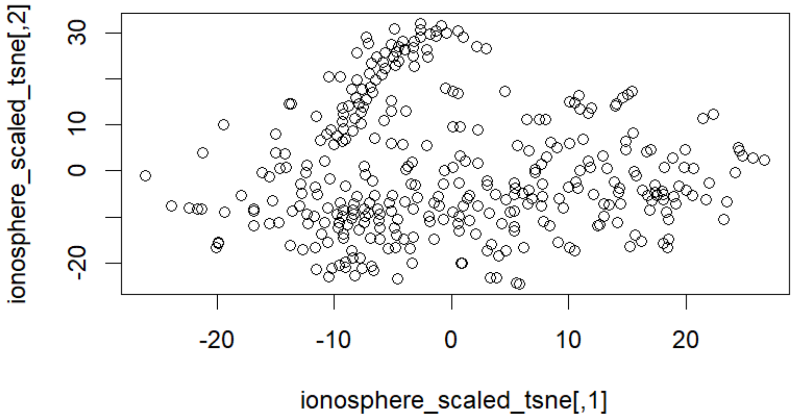
```
ionosphere_scaled <- scale(ionosphere_df)
ionosphere_df
A tibble: 351 × 11
       V3       V4      V5       V6       V7     V8    V9      V10     V11     V12
    <dbl>    <dbl>   <dbl>    <dbl>    <dbl>  <dbl> <dbl>    <dbl>   <dbl>   <dbl>
1  0.995  -0.0589   0.852   0.0231   0.834   -0.377 1       0.0376   0.852  -0.178
2  1      -0.188    0.930  -0.362   -0.109   -0.936 1      -0.0455   0.509  -0.677
3  1      -0.0336   1       0.00485  1       -0.121 0.890   0.0120   0.731   0.0535
4  1      -0.452    1       1        0.712   -1     0       0        0       0
5  1      -0.0240   0.941   0.0653   0.921   -0.233 0.772  -0.164    0.528  -0.203
6  0.0234 -0.00592 -0.0992 -0.119   -0.00763 -0.118 0.147   0.0664   0.0379 -0.0630
7  0.976  -0.106    0.946  -0.208    0.928   -0.284 0.860  -0.273    0.798  -0.479
8  0       0        0       0        1       -1     0       0       -1      -1
9  0.964  -0.0720   1      -0.143    1       -0.213 1      -0.362    0.926  -0.436
0 -0.0186 -0.0846   0       0        0        0     0.115  -0.268   -0.457  -0.382
ℹ 341 more rows
ℹ 1 more variable: V13 <dbl>
ℹ Use `print(n = ...)` to see more rows
```

- The data is scaled using the scale() function generalising the data and making the units of equal scale.

Plot the withiness against the number of clusters:



- Using the elbow test, the results suggest 3 or 4 clusters are ok for this data but for further accuracy as the elbow test can't be completely trusted, using a t-SNE test can help define the amount of clusters "k" appropriate.



The t_SNE test data looks very noisy making it hard to tell which number of "k" to use so the elbow rule will have to be sufficient here choosing k = 4.

How many clusters would you choose for a k-mean clustering analysis?

The amount of clusters chosen for a k-mean clustering analysis is 4.

K = 4 (changed: to k = 3 see below)

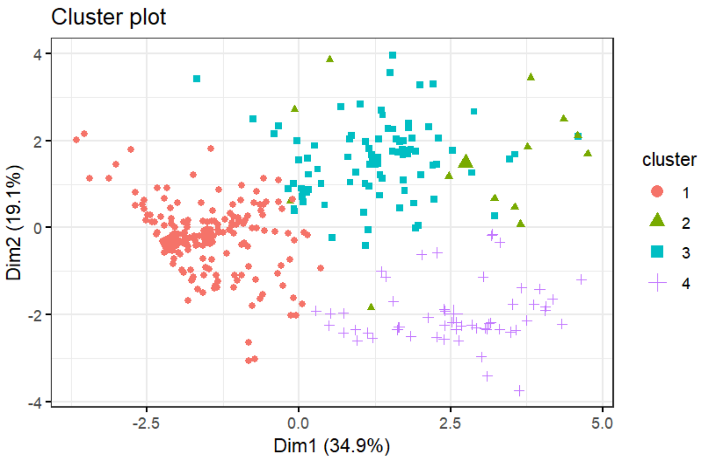
1b)

Perform a k-mean clustering analysis with the number of clusters chosen

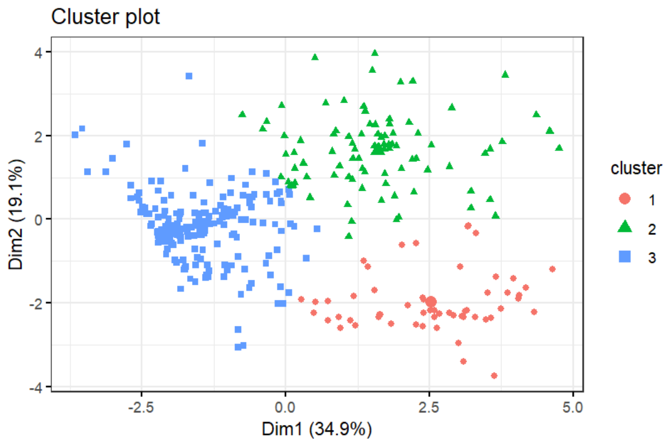
The k-mean algorithm is performed in R using k=4 and the code given in the assignment below.

```
set.seed(0)
ionosphere_k<-kmeans(ionosphere_scaled,4)
```

Visualise the clusters in the first two principal component plane using fviz_cluster function



Interpreting this cluster, I observe that cluster 2 has a very small amount of points and is converging with cluster 3 and cluster 4. To change this, using a smaller k value, k = 3 instead of k = 4 should output a better accurate result of clusters.



Using K = 3 has outputted a much more interpretable, precise, and accurate cluster plot.

Is the first principal component enough to separate the clusters?

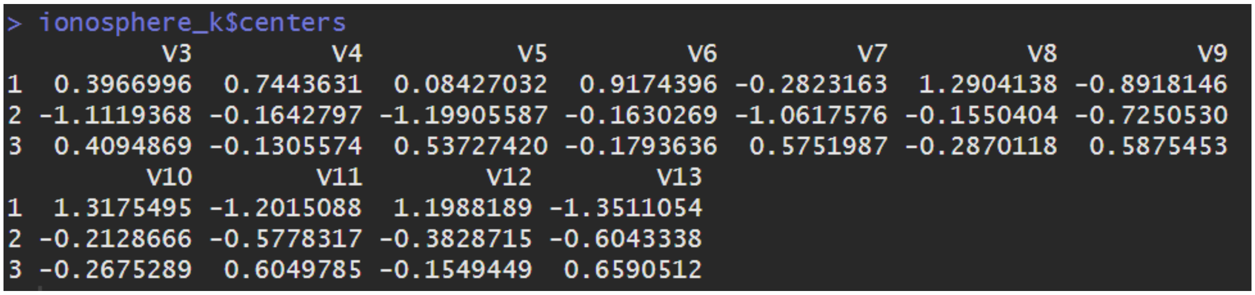
No, the first principal component isn't enough to separate the clusters. Cluster 3 is separated from cluster 2 and 3 along the First Principal component (x axis) although this isn't true for clusters 2 and 3 as they're separated mostly by the Second Principal Component (y axis).

Would you obtain identical results if you do not run set.seed(0) before performing clustering analysis?

The K-means algorithm runs on randomness, by using set.seed(0) it makes the centroids of each cluster start at 0. If this wasn't done then the k-means algorithm would pick a random starting centroid point changing the results. So no, Identical results would not be obtained if set.seed(0) wasn't run before performing cluster analysis.

1c)

Based on the output which of the cluster average has the highest value of pulse number V13?

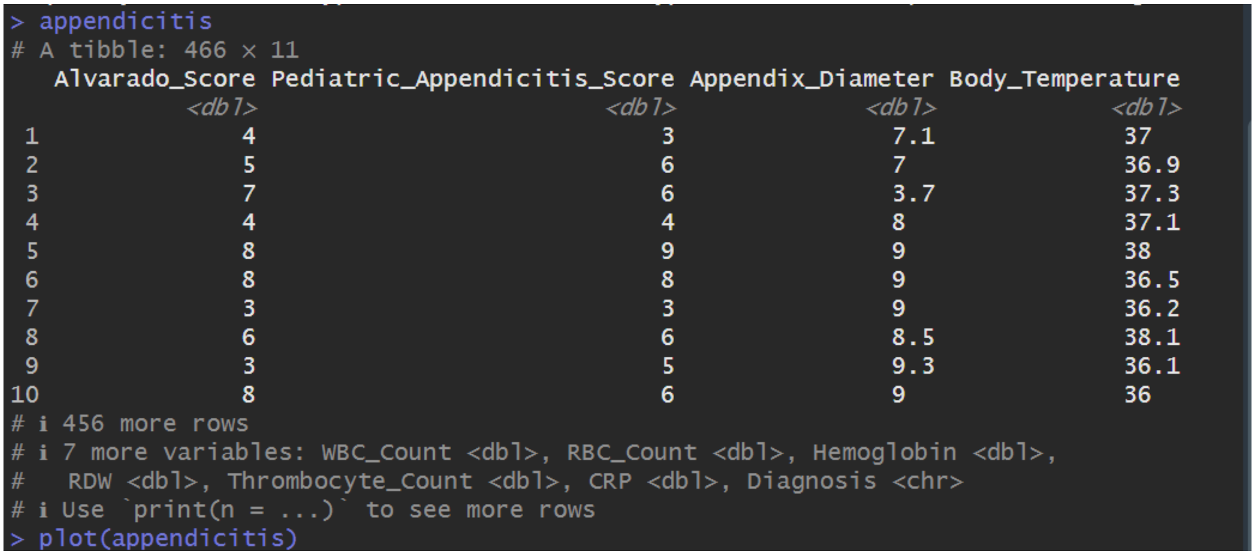


```
> ionosphere_k$centers
          V3         V4          V5         V6         V7         V8         V9
1  0.3966996  0.7443631  0.08427032  0.9174396 -0.2823163  1.2904138 -0.8918146
2 -1.1119368 -0.1642797 -1.19905587 -0.1630269 -1.0617576 -0.1550404 -0.7250530
3  0.4094869 -0.1305574  0.53727420 -0.1793636  0.5751987 -0.2870118  0.5875453
         V10        V11        V12        V13
1  1.3175495 -1.2015088  1.1988189 -1.3511054
2 -0.2128666 -0.5778317 -0.3828715 -0.6043338
3 -0.2675289  0.6049785 -0.1549449  0.6590512
```

Cluster1: -1.4, Cluster2: -0.6, Cluster3: 0.7: Therefore Cluster 3 has the highest pulse number for V13 with a value of 0.7.

2a)

Loading the data and checking the structure



```
> appendicitis
# A tibble: 466 × 11
   Alvarado_Score Pediatric_Appendicitis_Score Appendix_Diameter Body_Temperature
            <dbl>                        <dbl>             <dbl>            <dbl>
 1              4                            3               7.1             37
 2              5                            6               7               36.9
 3              7                            6               3.7             37.3
 4              4                            4               8               37.1
 5              8                            9               9               38
 6              8                            8               9               36.5
 7              3                            3               9               36.2
 8              6                            6               8.5             38.1
 9              3                            5               9.3             36.1
10              8                            6               9               36
# ℹ 456 more rows
# ℹ 7 more variables: WBC_Count <dbl>, RBC_Count <dbl>, Hemoglobin <dbl>,
#   RDW <dbl>, Thrombocyte_Count <dbl>, CRP <dbl>, Diagnosis <chr>
# ℹ Use `print(n = ...)` to see more rows
> plot(appendicitis)
```

All variables are numerical allowing floating point numbers apart from diagnosis which is a categorical variable containing a string value “appendicitis” or “no appendicitis” which is a binary identification categorical variable.

Convert The Outcome Variable Into Factor

Using the code:

appendicitis$Diagnosis<-as.factor(appendicitis$Diagnosis)

The categorical variable Diagnosis which is also the outcome variable is converted to a factor.

Which Of The Two Outcomes Is The “Positive” outcome?

Outcome 1: No appendicitis
Outcome 2: Appendicitis

The positive outcome is outcome 2: “Appendicitis”

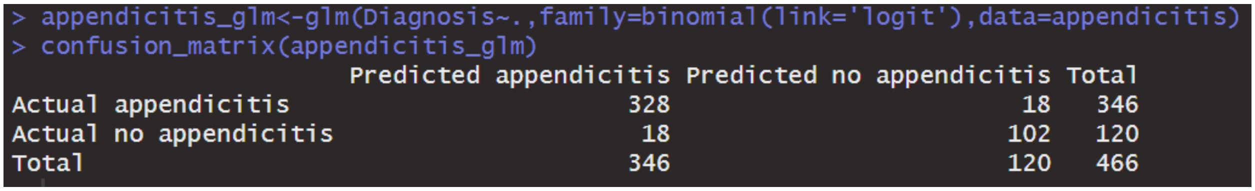
2b)

Fit a logistic regression model to the appendicitis dataset

Below is the code used for fitting the regression model to the appendicitis dataset using the outcome “Diagnosis”

appendicitis_glm<-glm(Diagnosis~.,family=binomial(link='logit'),data=appendicitis)

Print the confusion matrix and calculate misclassification error



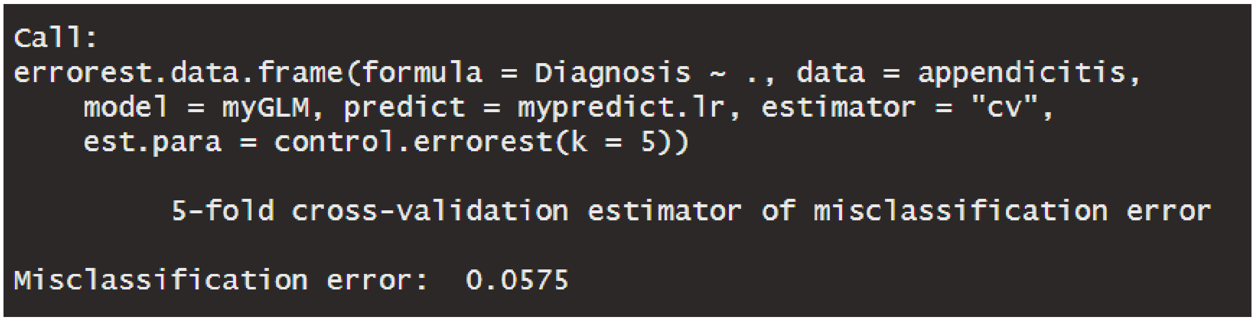
```
> appendicitis_glm<-glm(Diagnosis~.,family=binomial(link='logit'),data=appendicitis)
> confusion_matrix(appendicitis_glm)
                        Predicted appendicitis Predicted no appendicitis Total
Actual appendicitis                        328                        18   346
Actual no appendicitis                      18                       102   120
Total                                      346                       120   466
```

Misclassification Error = Number of incorrect predictions / Total number of predictions
Misclassification Error = (18 + 18) / (346 + 120)
= (36) / 466
= 0.077

The misclassification error is 0.077 or 7.7%

2c)

Why is the error different from the one calculated in subquestion b?



```
Call:
errorest.data.frame(formula = Diagnosis ~ ., data = appendicitis,
    model = myGLM, predict = mypredict.lr, estimator = "cv",
    est.para = control.errorest(k = 5))

         5-fold cross-validation estimator of misclassification error

Misclassification error:  0.0575
```

Misclassification error for 2b = 0.077
Misclassification error for 2c = 0.058

The errors are different because of random variation in folds.

Which of The Errors Reflects The Prediction Error?

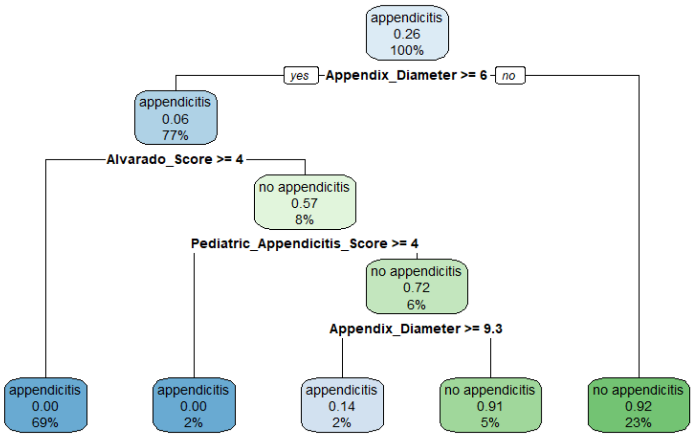
The cross validation error (0.058) reflects the prediction error it estimates how often the model is likely to misclassify a new observation.

2d)

Fit a decision tree model

appendicitis_tm <- rpart(Diagnosis ~ ., data=appendicitis)

Plot the obtained decision tree



What is the predicted diagnosis for a patient with Al varado_Score = 3, Pediatric_Appendicitis_Score = 3, and Appendix_Diameter = 6?

The predicted diagnosis for a patient with Al Varado score = 3, Pediatric Appendicitis score = 3, and appendix diameter = 6

Following the tree gives:

The predicted diagnosis is No appendicitis.

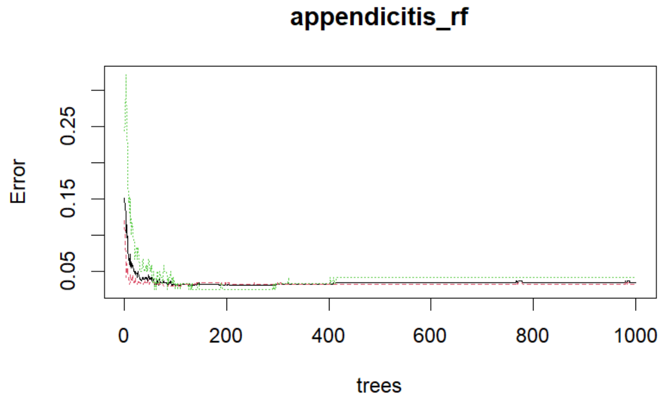
2 e/f)

Fit a random forest model using 1000 trees

Fitted using the code below.

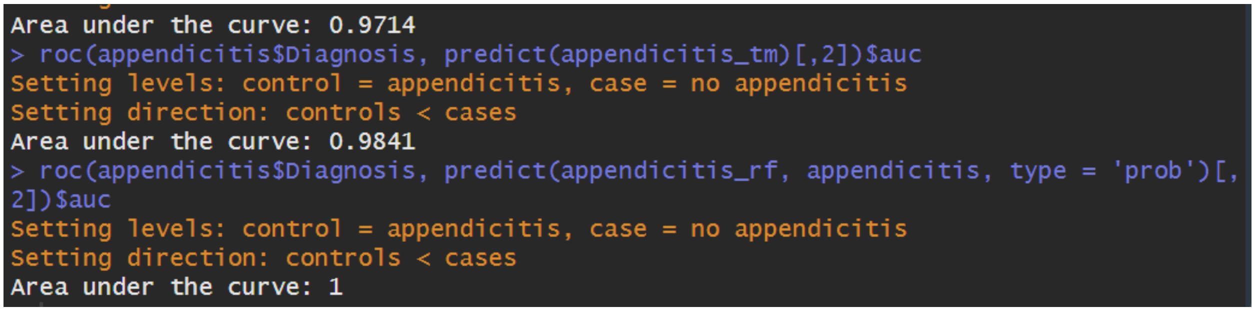
set.seed(0)
appendicitis_rf <- randomForest(Diagnosis ~ ., data=appendicitis, ntree=1000)

Plot the graph of errors for this model. Would 200 trees be enough for this analysis?



Looking at this plot, I observe that 200 trees appears to be enough for this analysis because the error rates stabilise before 200 trees and beyond the 200 value the model doesn't improve making it a sufficient amount.

2 f/g)



```
Area under the curve: 0.9714
> roc(appendicitis$Diagnosis, predict(appendicitis_tm)[,2])$auc
Setting levels: control = appendicitis, case = no appendicitis
Setting direction: controls < cases
Area under the curve: 0.9841
> roc(appendicitis$Diagnosis, predict(appendicitis_rf, appendicitis, type = 'prob')[,
2])$auc
Setting levels: control = appendicitis, case = no appendicitis
Setting direction: controls < cases
Area under the curve: 1
```

Which of the models performs the best on the training data?

The model with the highest area under the curve represents the best performance on the training data.

roc(appendicitis$Diagnosis, predict(appendicitis_rf, appendicitis, type = 'prob')[,2])$auc

The model above has an area under the curve of 1 meaning it has perfect classification being the best performance on the training data.