1. (15 points) The Cleveland heart-disease study was conducted by the Cleveland

Clinic Foundation. The response variable is “diag1” (diagnosis of heart disease: buff = healthy, sick = heart disease). There is a second “diag2” that contains stage information about the sick, this can be disregarded. There were 303 patients in the study, and 13 predictive variables, including age, gender, and a range of biological measurements.

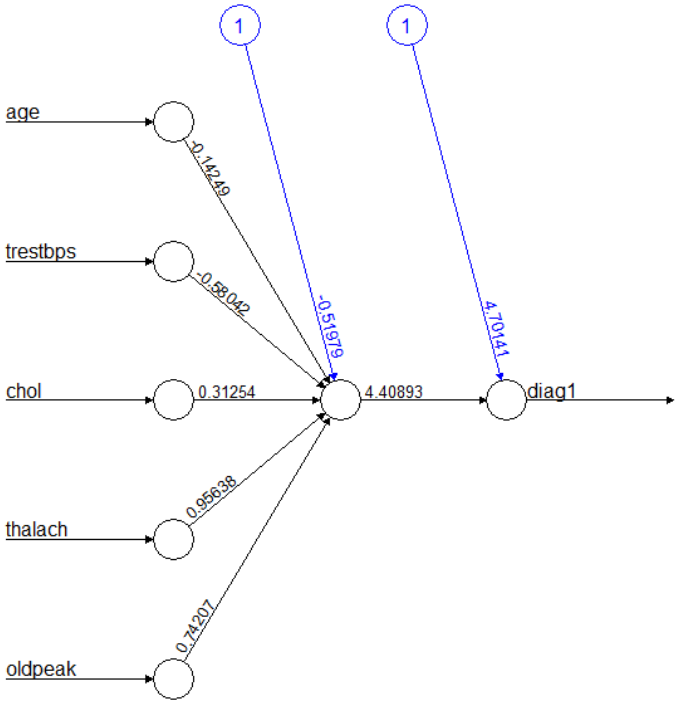
Fit a neural network, CART model and a random forest to the Cleveland heartdisease data. Compare the results, and comment on the performance.

First, I made the cleveland data diag1 column into table function.

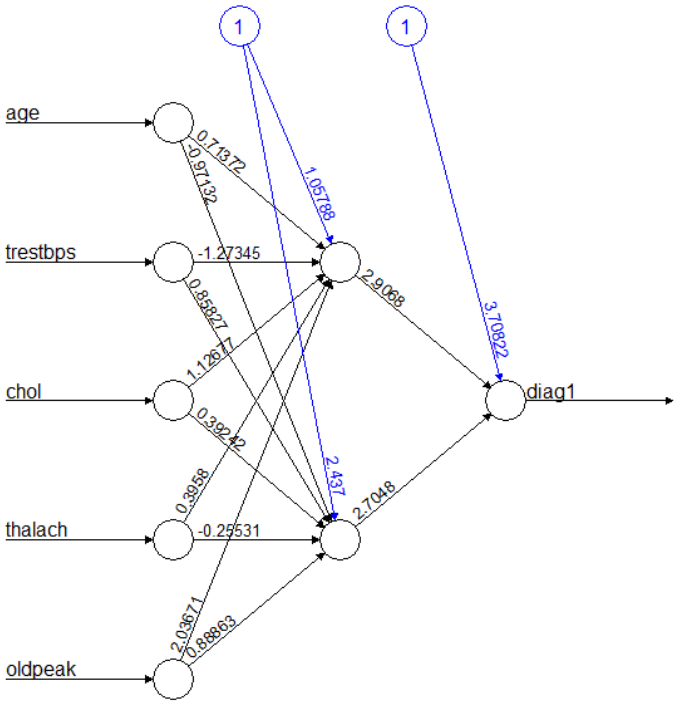
	Buff	Sick
Number	160	136
Probability	0.5405405	0.4594595

We could see that the data are evenly spread out. Also, I set the diag1 into numeric value; 1 is Buff and 2 is sick.

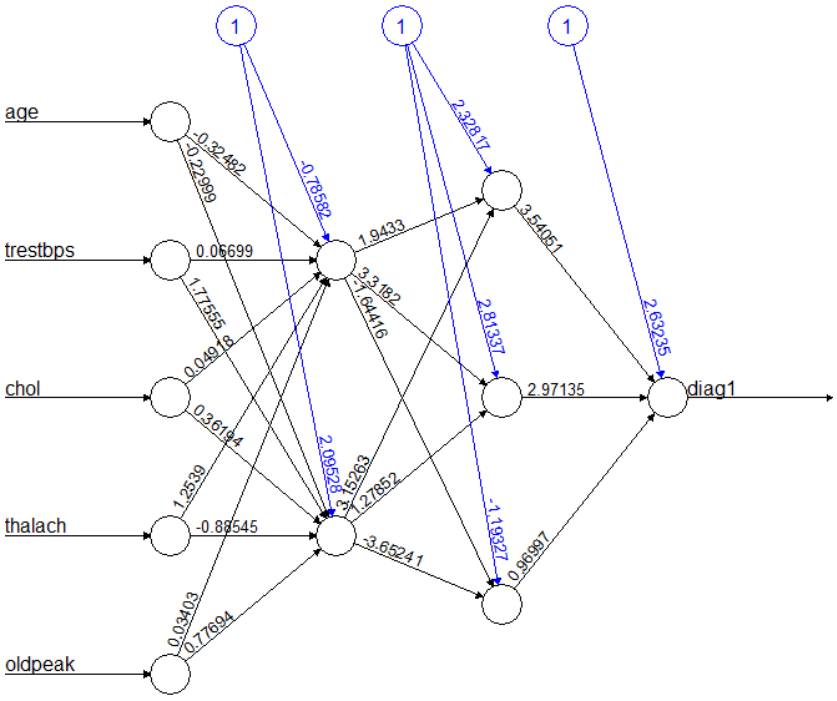
I setup the neurons into png image.



Error: 45.009947 Steps: 46



Error: 45.008067 Steps: 31



Error: 45.009639 Steps: 24

The blue line represents the outliers to each node. As I increased the hidden and stepmax was added, the outliers reduced accordingly.

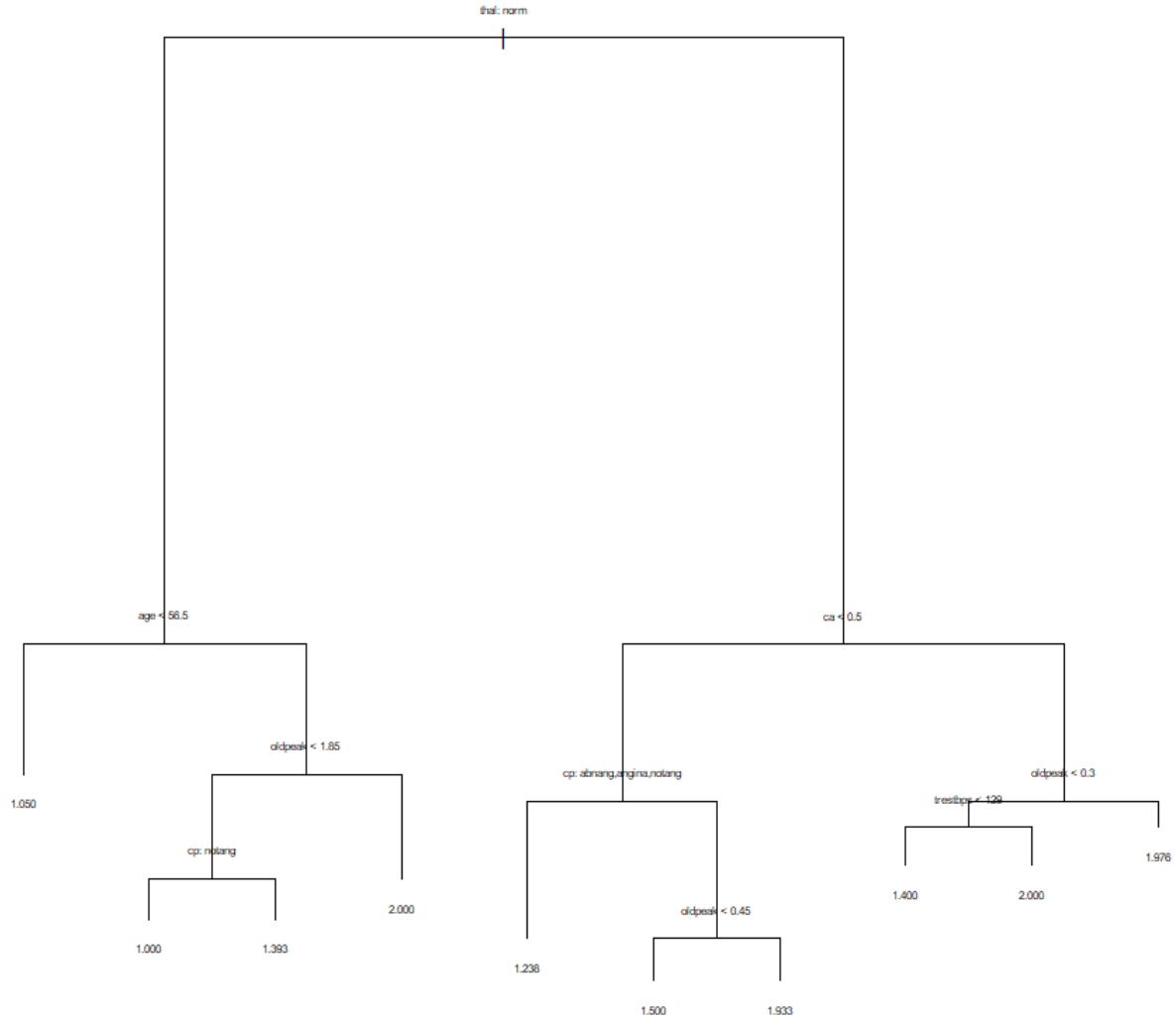
I also made a prediction on second picture in order to see the errors.

Train error is 0.4568528. Test error is 0.4646465.

For tuning, I did a loop on neural data. The picture below is the result.

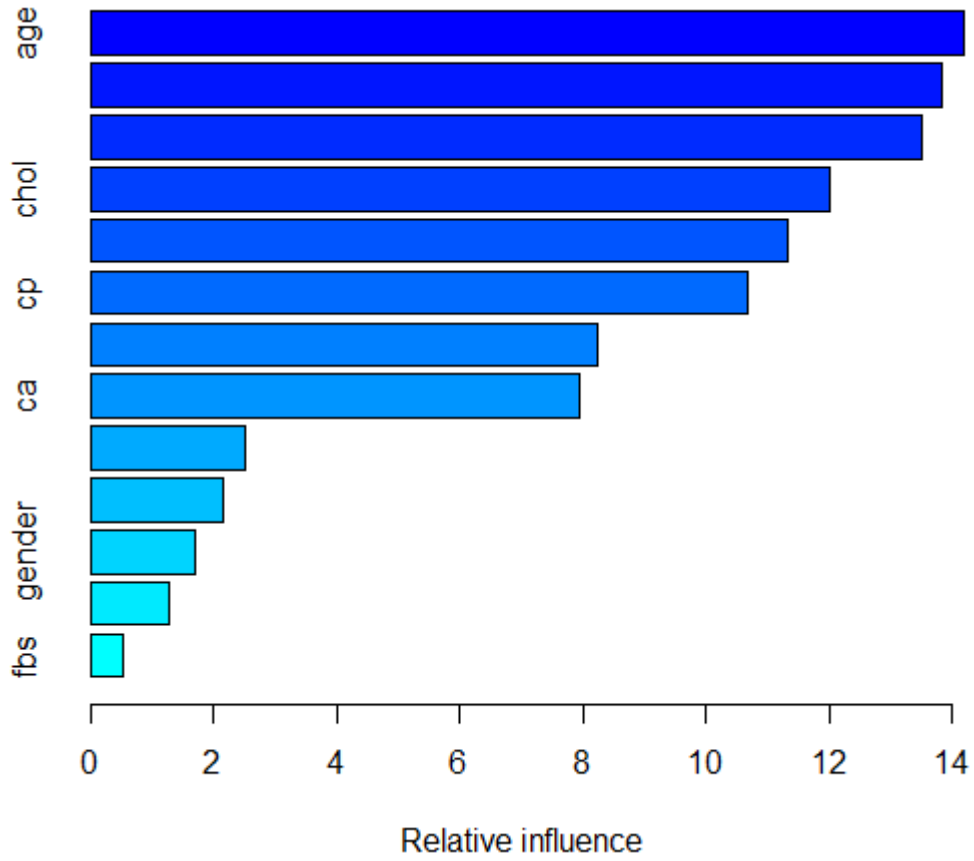
```
> train_err_store
[1] 0.4568528 0.4568528 0.4568528 0.4568528
> test_err_store
[1] 0.4646465 0.4646465 0.4646465 0.4646465
```

Comparing neural network to CART model, I used tree model to set the pruned tree and the error rate. The p



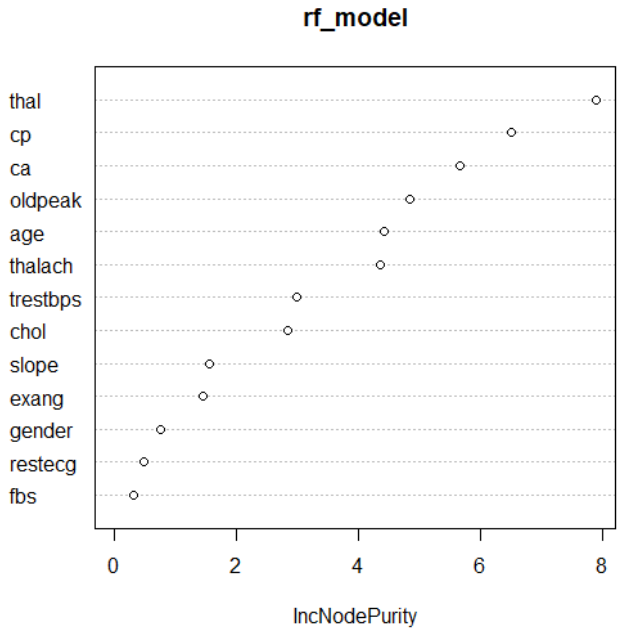
Pruned train error rate is 0.1370558. Pruned test error rate is 0.2727273.

I also did boosting on diag1, and the related figure is shown below:



As you can see, age is the most important variable in this case, however, variables like fbs and gender are factors that less influence diag1. The mse for boosting was 0.1740702.

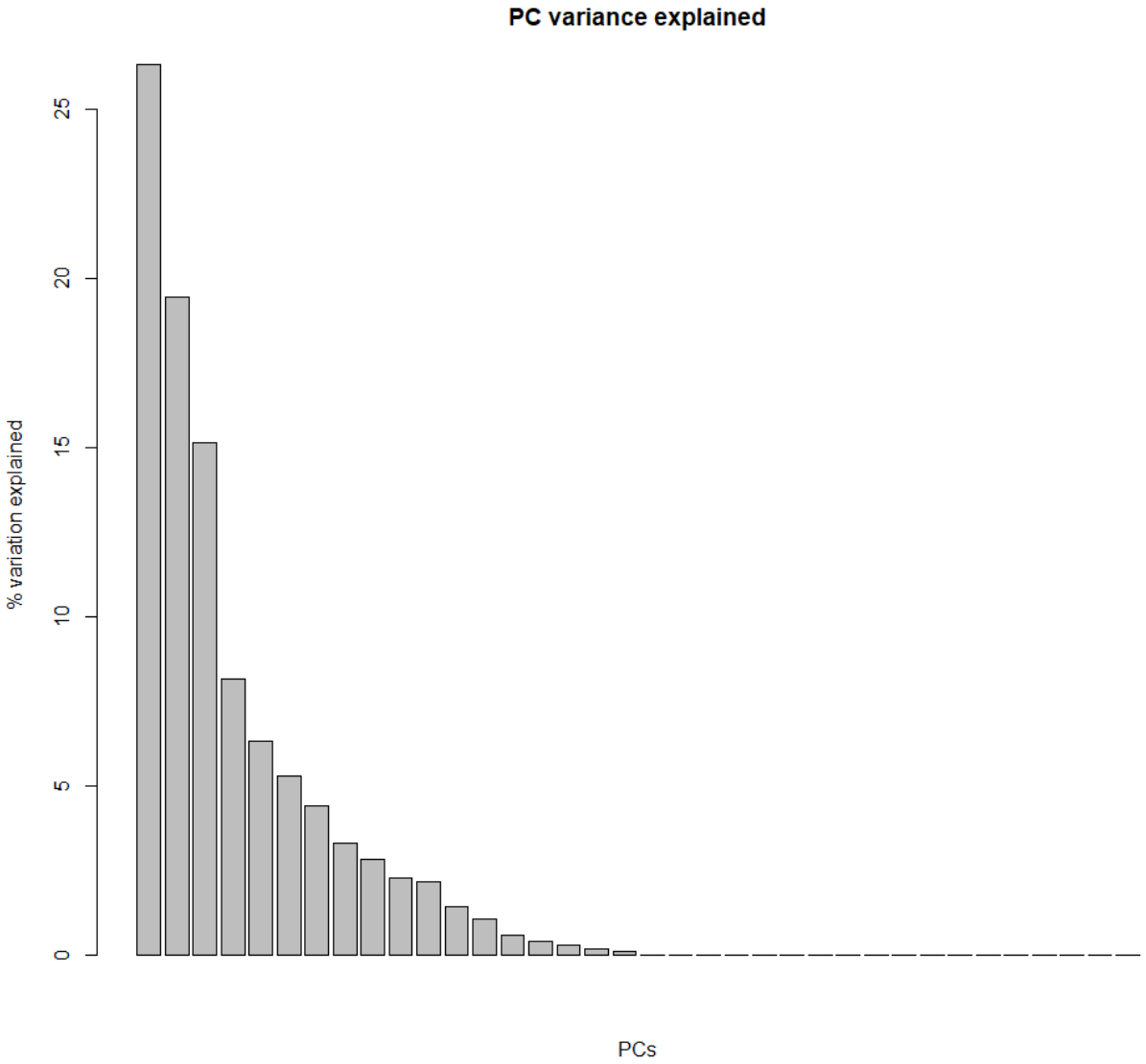
Lastly, I looked at random forest model. For random forest, I looked at the varImpPlot to help us with visualization.



This is the varImpPlot of randomForest this shows that thal was the most influencing factor. Also same as boosting, gender and fbs has the lowest importance. The error rate for random forest model is 0.1196024.

By comparing from neural network, CART, boosting to random forest, the one that has the lowest error is random forest. However, somehow in my work, even though random forest has the lowest error, the explanation is pretty low, which is 42.23%.

2.



This bar graph shows that the first three PCs are the ones that explains the variation.

```
> pc_var/(sum(pc_var))
 [1] 2.630834e-01 1.943963e-01 1.516544e-01 8.161173e-02 6.339310e-02 5.320142e-02 4.414623e-02
 [8] 3.331509e-02 2.845822e-02 2.278345e-02 2.192263e-02 1.443029e-02 1.088820e-02 6.176820e-03
[15] 4.217738e-03 3.189853e-03 1.866525e-03 1.264666e-03 2.355800e-29 5.009562e-30 4.465650e-31
[22] 3.450735e-31 1.160768e-31 7.648004e-32 7.421911e-32 7.353552e-32 5.485035e-32 4.111905e-32
[29] 1.667839e-32 1.359463e-32 1.101371e-32 8.821714e-33 6.740766e-33 3.501138e-33 1.616485e-33
[36] 3.513547e-34
```

These are the percentages that pcs are explained. By far I am not getting 80% 90% explanation.

Moving on, I thought number handwriting should be between 0~9. I have classified into 3 different colors. Ones that have 0,1,2 are marked as red, 3,4,5 are marked as blue and 6,7,8,9 are marked as green. Other than that, are marked as black. 3d scatterplot is used for PC1, PC2, and PC3.

As a result, there were lots of blacks in this scatterplot. Next, I had green, blue and red.

I tried to do the knn, but it did not allow me to do it for whatever reason I don't understand. I will try my best to finish this anyways.

3.

a) (2 pts) Explain how cross validation is implemented.

Cross validation divides the overall data set in to multiple k subsets. We test the overall data by having one divided data set into the test set one at a time. We change the test set while doing the cross validation, eventually testing all of the data by parts. It helps the data to be more accurate, but it is time consuming.

b) (4 pts) What are the advantages and disadvantages of k-fold cross validation relative to:

i) The holdout method (division of the data into test and training).

We also call holdout method as cross validation. Advantage compared to k-fold cross validation is since it is doing iteration only one time, we do not need lots of computation. However, the test set could be overfitting if we do tuning on the test set multiple times.

ii) LOOCV

In leave one out cross validation, we do testing on every sample, which means there would not be any randomness and more stable compared to validation approach. Moreover, it has few bias. Since LOOCV is testing every data, it is very time consuming than cross validation.

c) (9 points) Write an *.R function to implement k-fold cross validation. Apply it to a data set of your choice. Use it to compare the results of 10-fold, 5-fold and LOOCV. Submit your fully commented function, as well as the application of the dataset that you selected.

For shortening the time, I only used few data from mtcars dataset. I only picked up mpg ~ cyl column. The figures below is the knn result of 5 and 10.

```
> k5$delta
[1] 11.95784 11.67405

> k10$delta
[1] 11.53479 11.43822
```

The accuracy for knn = 5, each had 11.95784, 11.67405 errors respectively. and for knn= 10, 11.53479, 11.43822 errors respectively. Since the data is not that big, I do not see much of a difference between two, but clearly, knn 10 has low errors than 5. Moreover, for LOOCV, I had 11.20843, 11.18290 errors each.

```
> k5$delta
[1] 11.95784 11.67405
```

Comparing all of these three results, LOOCV had the lowest errors. The fully commented function is shown in my r.data.