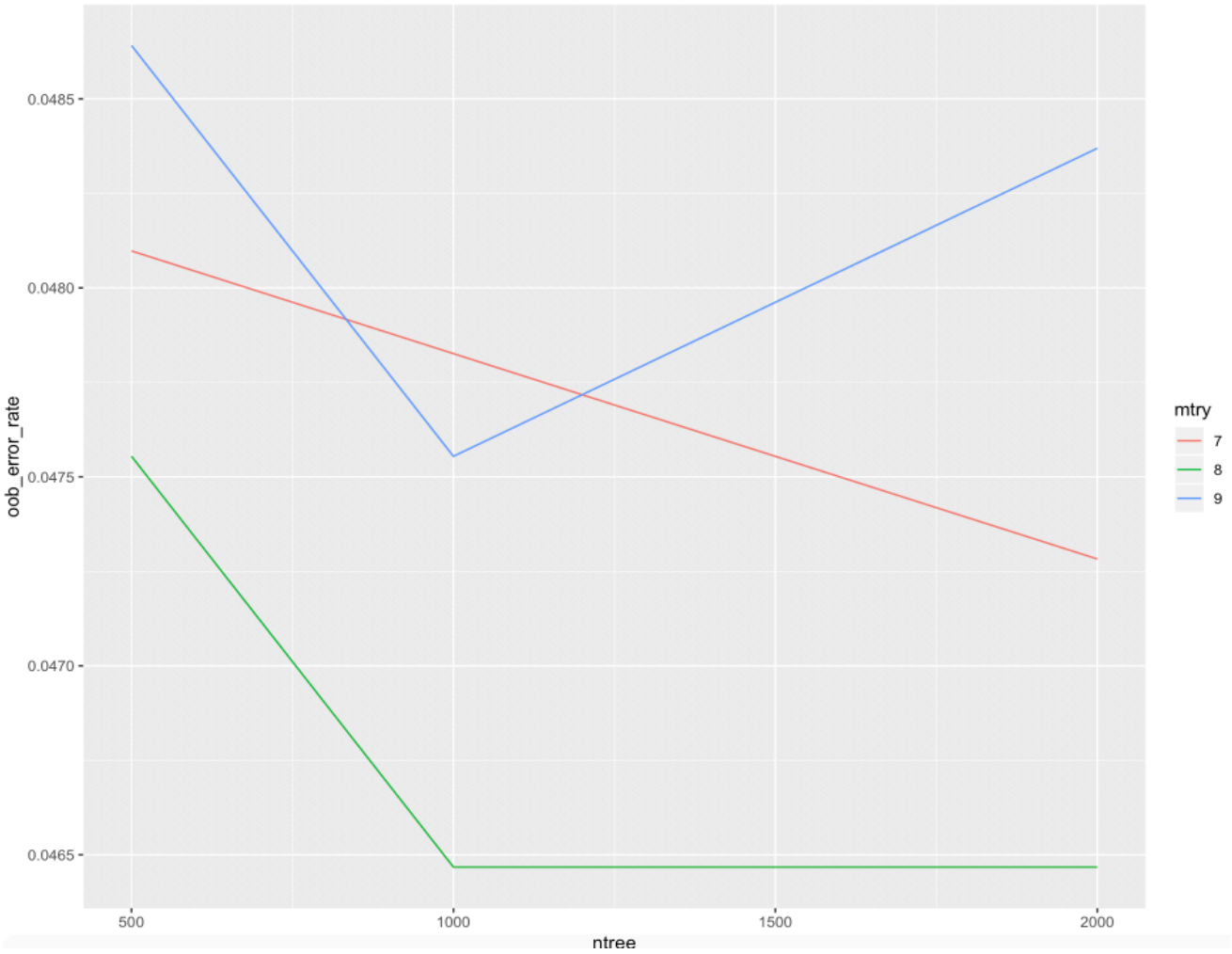
Priya Karadi – 26
UB IT: 50291880

Assignment – 5

Q1. Used the 'spam' dataset to explore the number of randomly selected inputs for each tree.

Sol:

- Below is the graph plotted against OOB_Error vs Randomly chosen NTree.



- Minimum OOB_Error

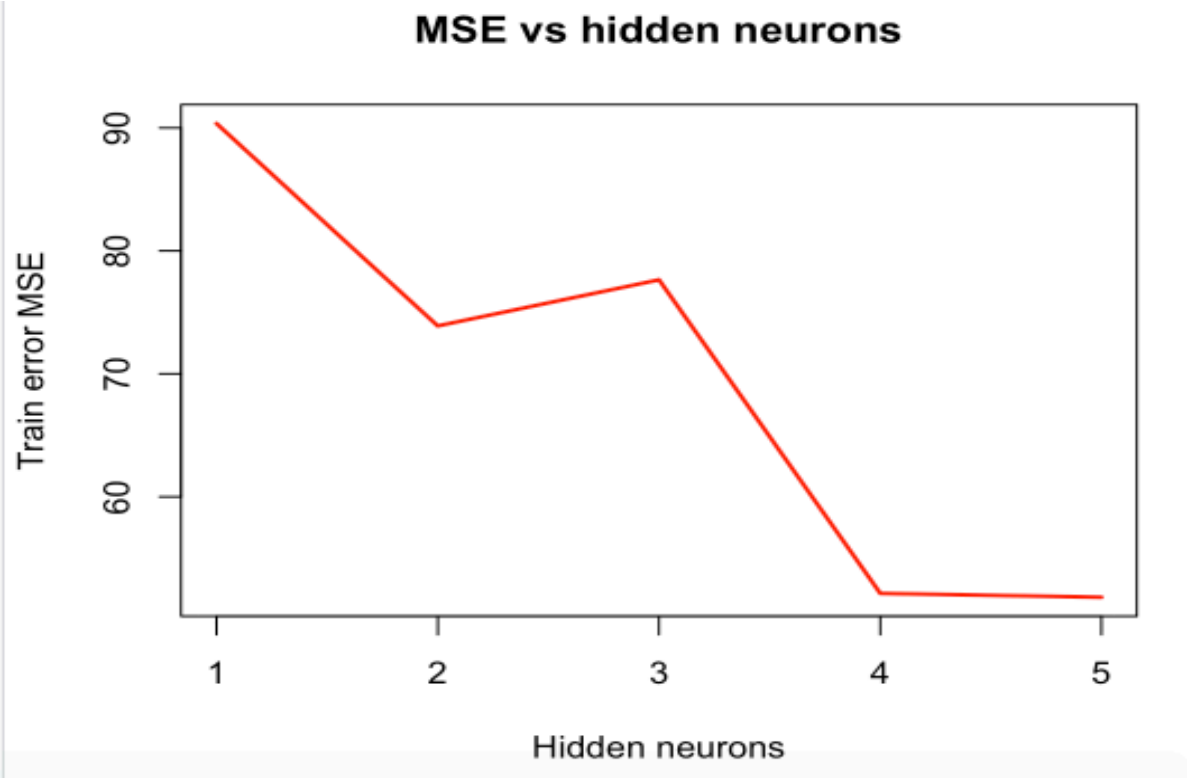
```
> min(oob_err$oob_error_rate)
[1] 0.0464673913
```

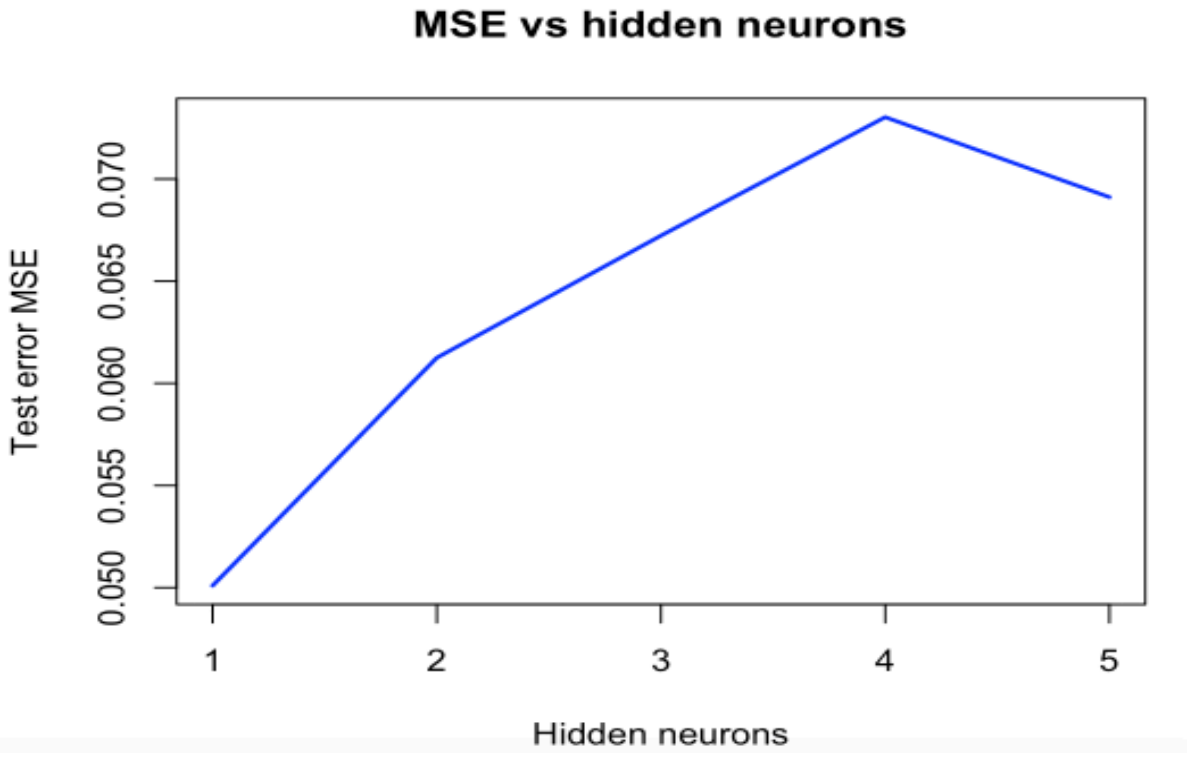
- Confusion Matrix and mean error observed:

```
> table(y_train,yhat)
       yhat
y_train email spam
  email   532   17
  spam     31  341
> mean(y_train != yhat) # error rate
[1] 0.05211726384
```

Q2. Used the 'spam' dataset to fit a neural network.

- We used cross-validation to determine the number of neurons in the layer.



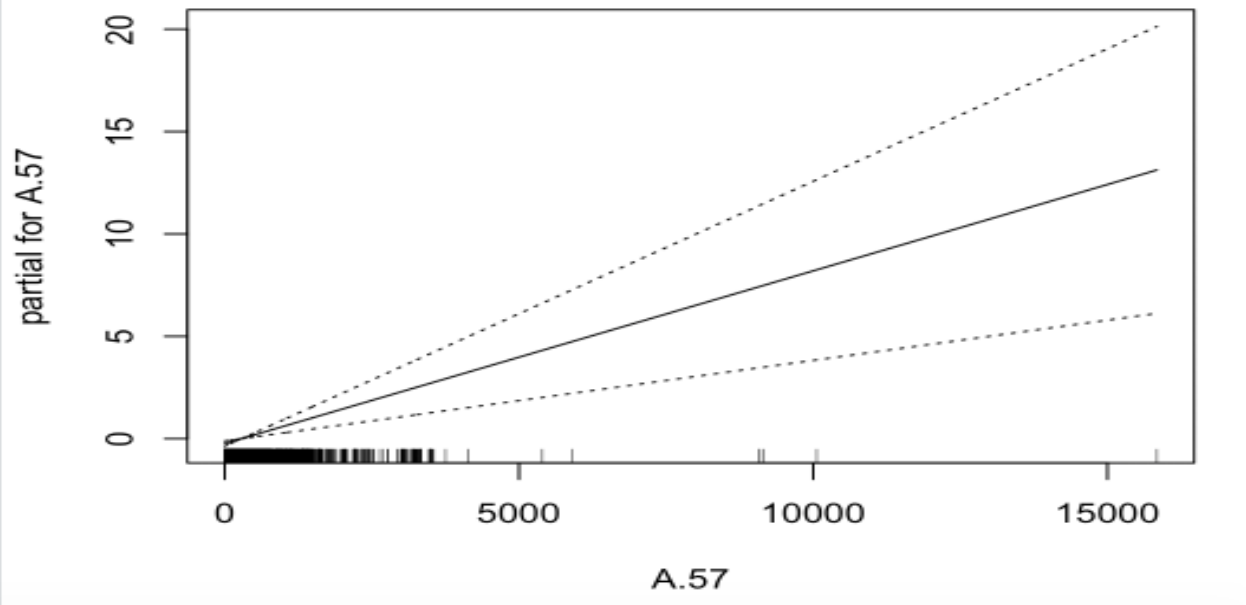


```
> # Print out test and train error vectors
> test.error
[1] 0.05009774255 0.06125053337 0.06723008976 0.07302726873 0.06911173107
> train.error
[1] 90.35803752 73.89572330 77.64728695 52.14418894 51.82841341
>
> # Plot train error
> plot(train.error,main='MSE vs hidden neurons',xlab="Hidden neurons",ylab='Train error MSE',type='l',col='red',lwd=2)
> # Plot test error
> plot(test.error,main='MSE vs hidden neurons',xlab="Hidden neurons",ylab='Test error MSE',type='l',col='blue',lwd=2)
>
> # Number of neurons (index) that minimizes test/train error
> which(min(test.error) == test.error)
[1] 1
> which(min(train.error) == train.error)
[1] 5
```

- Comparing with Additive Model, the test error is as follows:

```
> error_rate
[1] 0.1019522777
```

- For Additive Model, a graph similar to the one below, is generated for each of the 57 predictors.

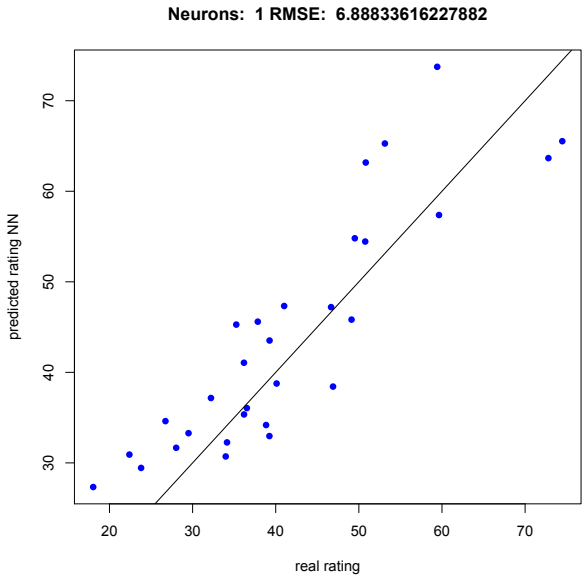


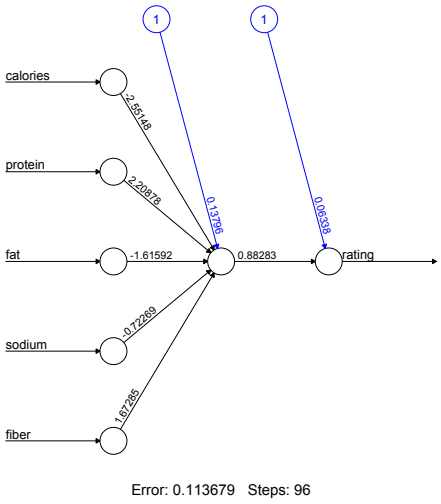
- Additive models are more interpretable, because the relationship between each predictor and the response is now modeled using a curve. However, neural-nets have better performance.

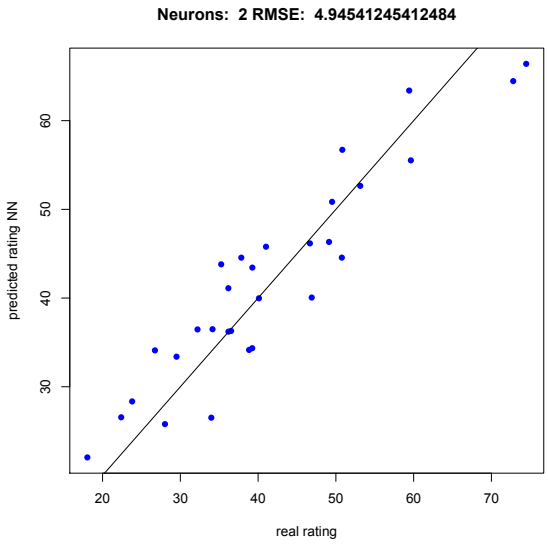
Q3. Used the 'cereals.csv' for this question.

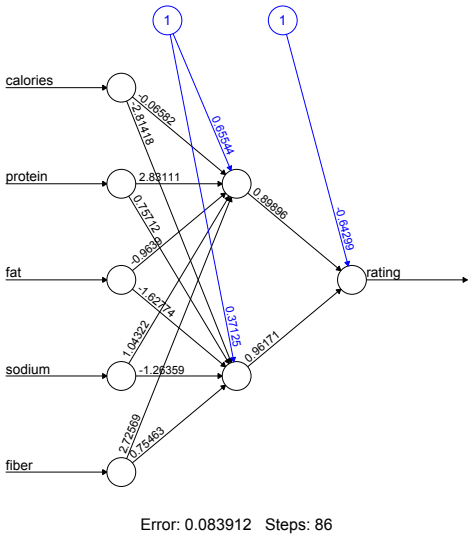
Sol:

- This dataset has 5 predictors and 1 response variable. All are either integers or numbers.
- Divided the dataset for training and testing.
- Fit the Neural Net on training set with neuron count ranging from 1 to 5.
- Run each model on the testing dataset and observed that the model with 5 neurons, gave the best RMSE value.

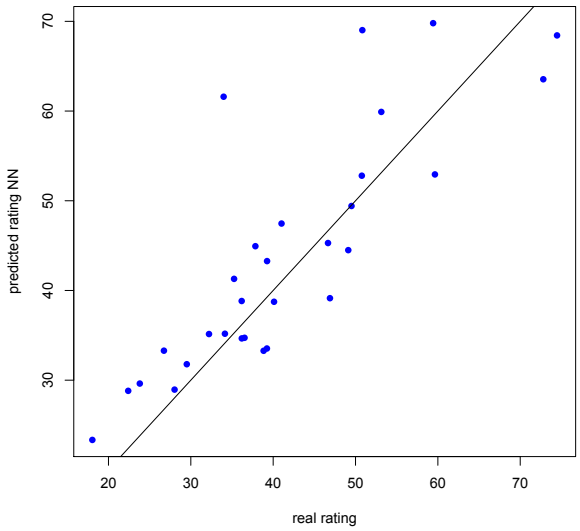


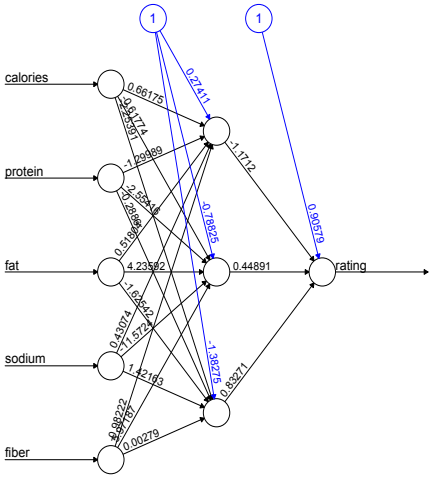






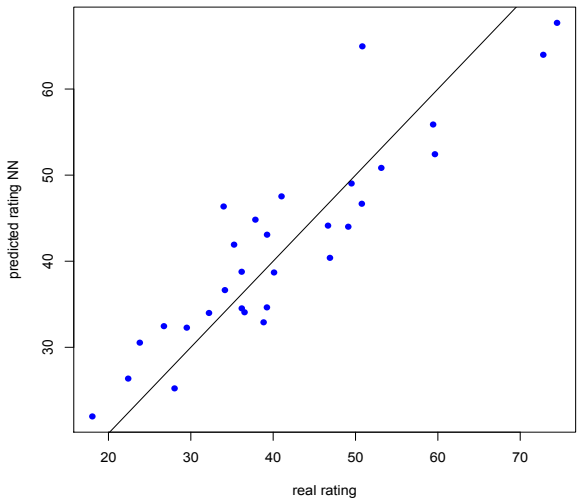
Neurons: 3 RMSE: 7.92839731899379

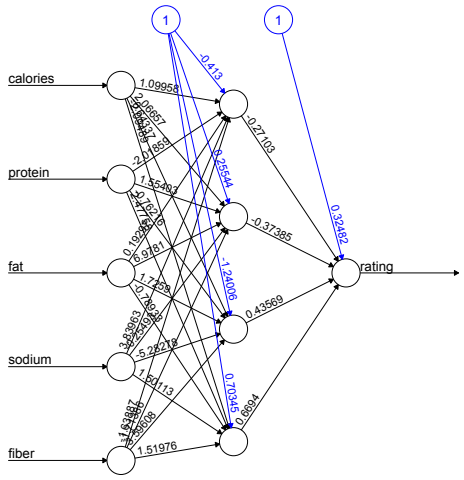




Error: 0.070124 Steps: 157

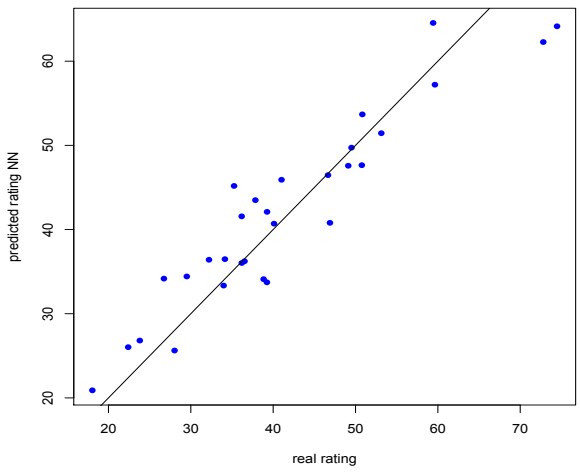
Neurons: 4 RMSE: 5.76451144168192

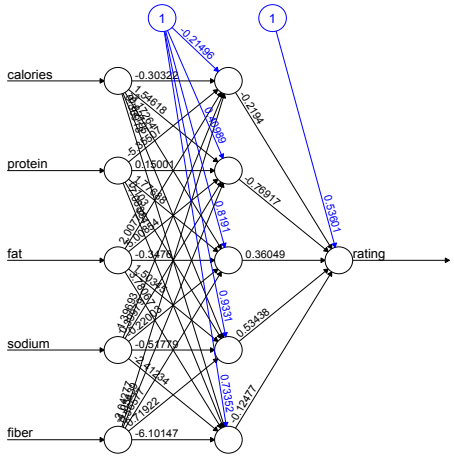




Error: 0.061691 Steps: 143

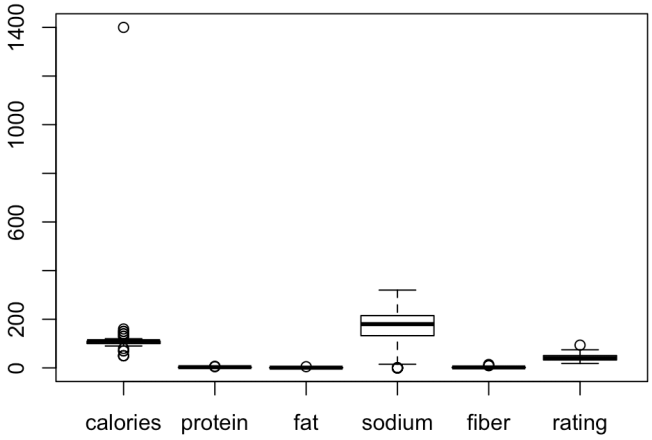
Neurons: 5 RMSE: 4.80946044316649



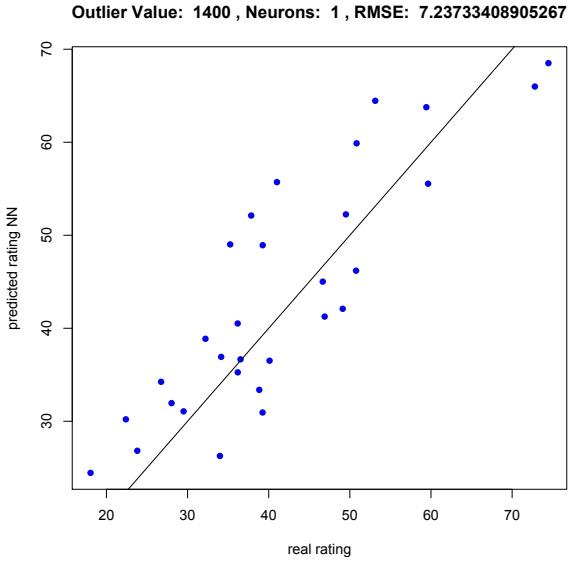


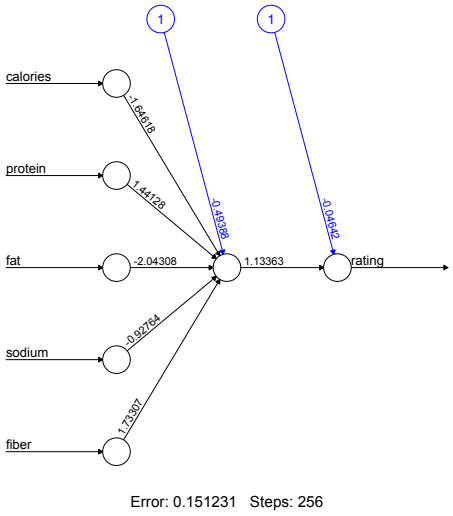
Error: 0.074605 Steps: 145

- Modified the data to introduce univariate outlier. The original value was calories = 140, we modified it to 1400.



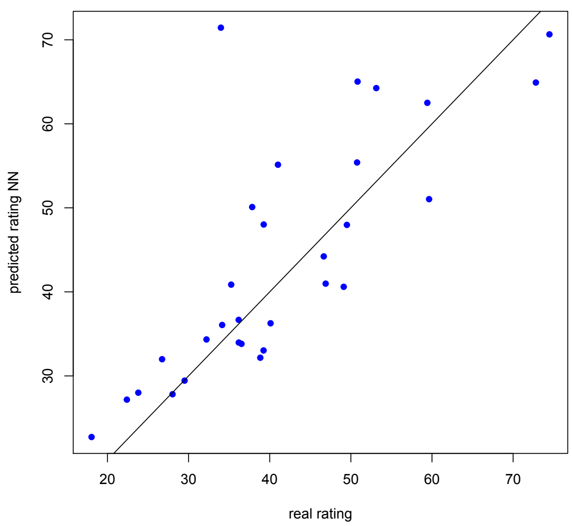
- Used holdout method to compare and select the most suitable number of neurons. In this case, 5 neurons give the best result.
  Below are the models trained/tested on dataset containing outlier:

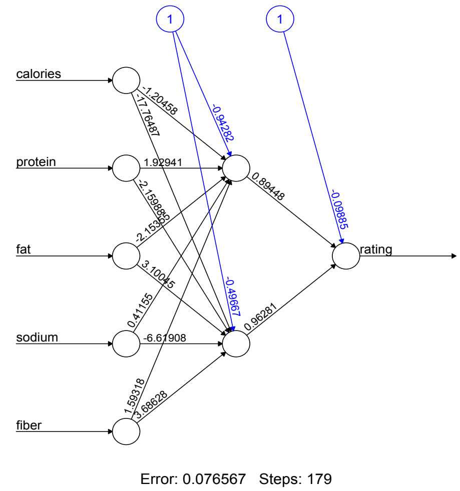




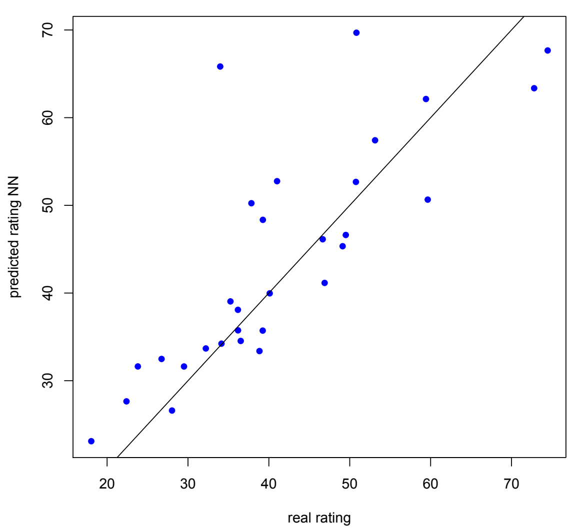
Priya Karadi – 26
UB IT: 50291880

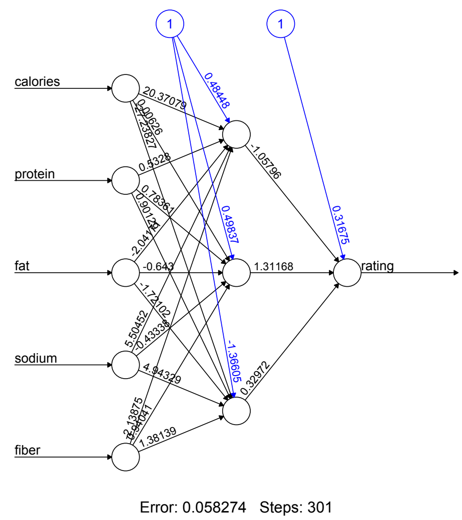
Outlier Value: 1400 , Neurons: 2 , RMSE: 9.47346043068772

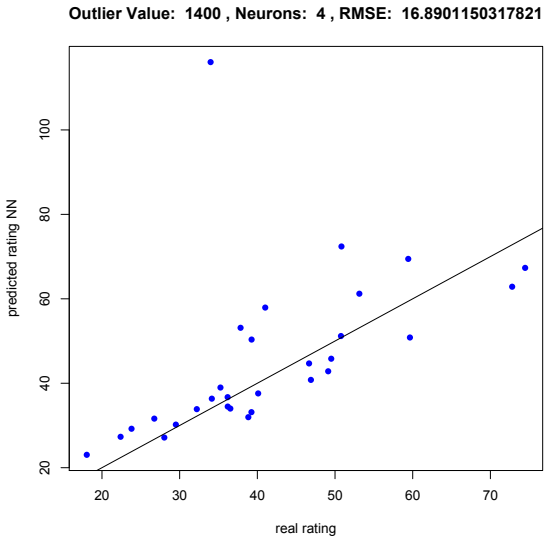


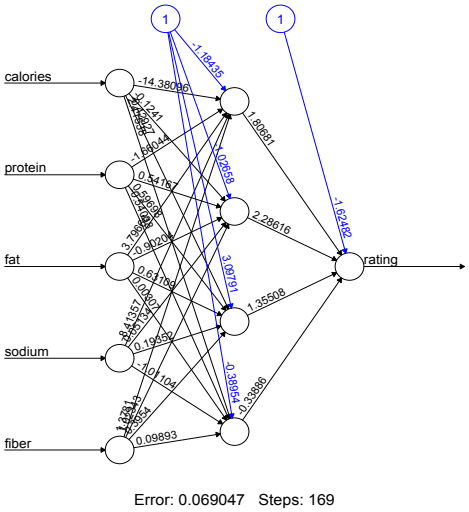


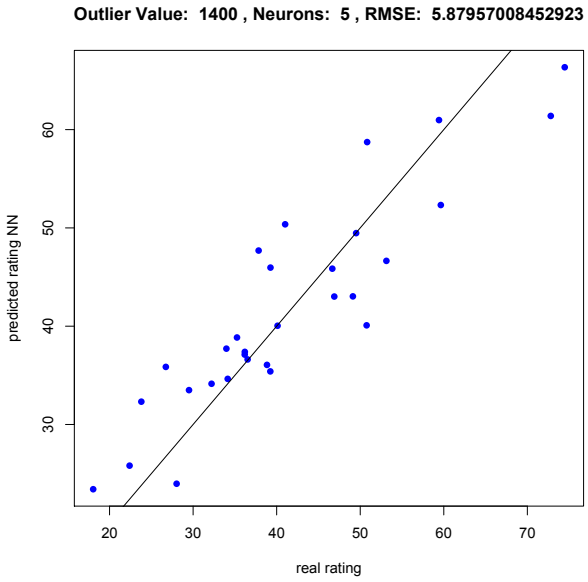
Outlier Value: 1400 , Neurons: 3 , RMSE: 8.69170626260077

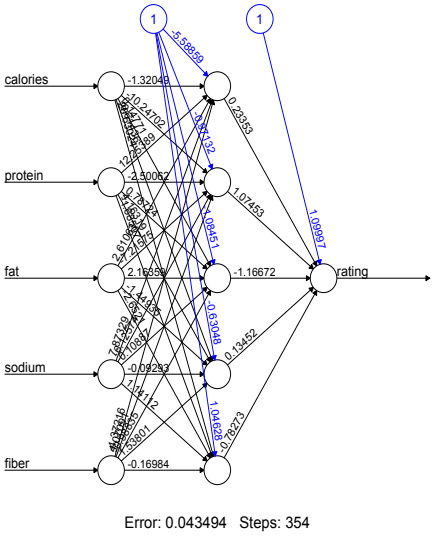












- We concluded that 5-neuron model was the best fit for this outlier dataset.

- Now, we shrink the value of the outlier by a step size of 150.
- Following results were obtained on RMSE for both models.

5-NN Outlier on Outlier Model

```
[1] 1400
[1] 5.879570085
[1] 1250
[1] 5.979344406
[1] 1100
[1] 6.3786187
[1] 950
[1] 7.21740829
[1] 800
[1] 8.643146177
[1] 650
[1] 10.81225521
[1] 500
[1] 13.87972888
[1] 350
[1] 17.61557336
[1] 200
[1] 20.8486392
```

5-NN Outlier on Original model

```
[1] 1400
[1] 18.57585989
[1] 1250
[1] 18.39772529
[1] 1100
[1] 18.16820468
[1] 950
[1] 17.86134832
[1] 800
[1] 17.43024083
[1] 650
[1] 16.78057031
[1] 500
[1] 15.69118663
[1] 350
[1] 13.50030542
[1] 200
[1] 7.314554533
```

- The above results show that as the outlier value shrinks, when fit to the outlier model (containing outlier), the model performs badly. On the other hand, when the outlier value shrinks and is fit using the Original model, the model gives better RMSE.
- On making a change of roughly 1000, to the outlier, following coefficients were returned.

```
> out[45,]
    calories      protein          fat       sodium        fiber    nn-output
0.8181818182 0.4000000000 0.2000000000 0.5937500000 0.2857142857 0.2314064001
> out2[45,]
    calories      protein          fat       sodium        fiber    nn-output
1.0000000000 0.4000000000 0.2000000000 0.5937500000 0.2857142857 0.1393564789
```

- We can conclude that neural nets are resilient to outliers.

Q4. In this question, we have used 'OJ' dataset from 'ISLR' package to predict 'Purchase' using various SVM flavors.

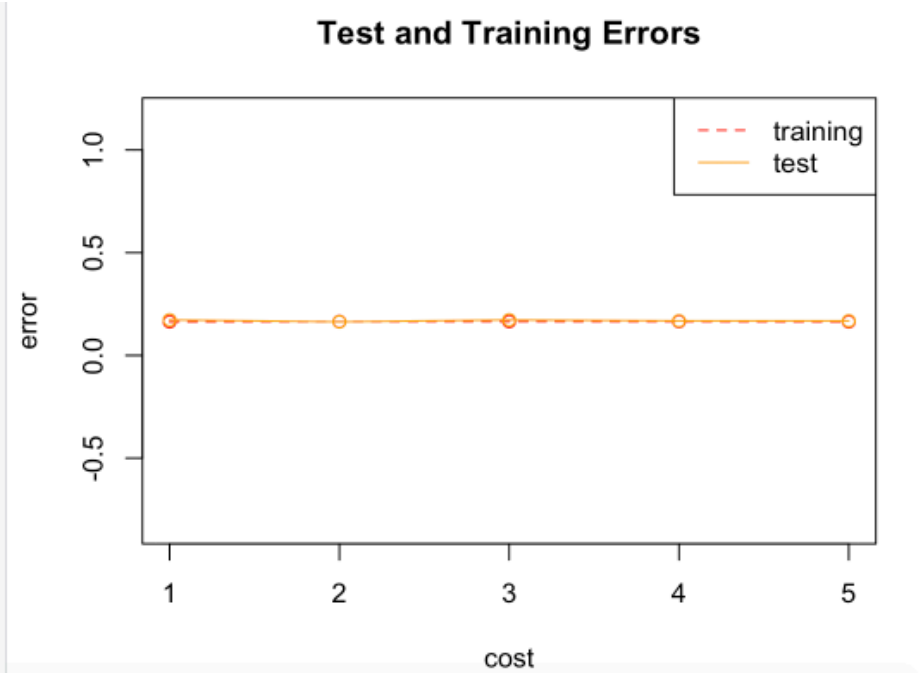
Sol:

We used SVC, SVM – radial and polynomial to check training and testing errors.

Observation: It was observed that we are getting minimum test error with polynomial kernel followed by radial and linear SVM.

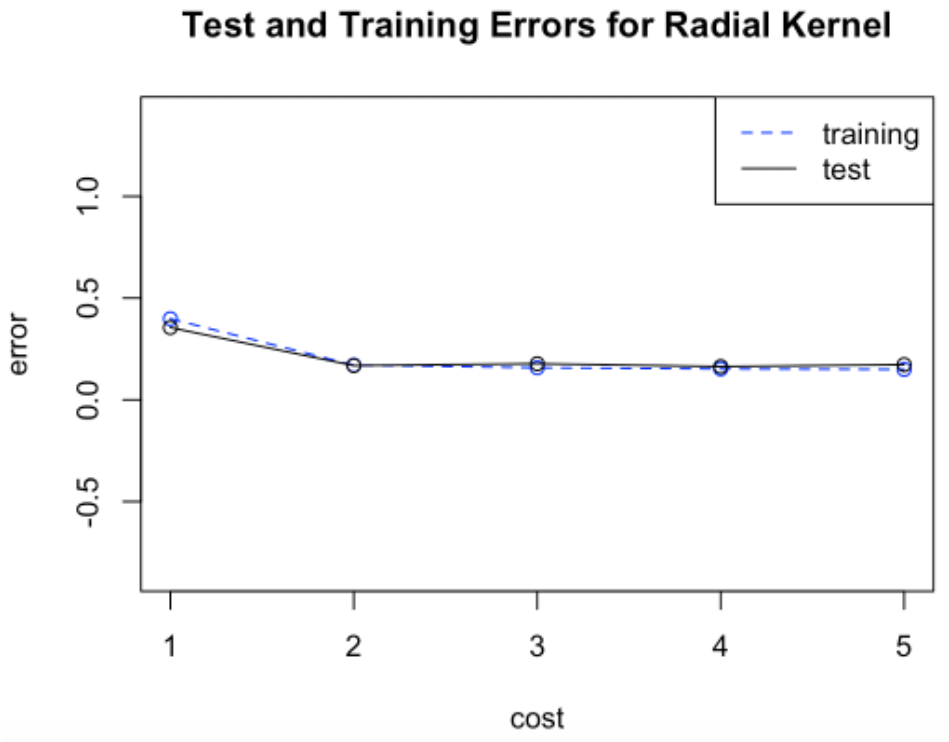
a) Linear Kernel:

```
> linear_test_error
[1] 0.1728971963 0.1635514019 0.1728971963 0.1682242991 0.1682242991
> linear_train_error
[1] 0.1635514019 0.1647196262 0.1647196262 0.1647196262 0.1635514019
```



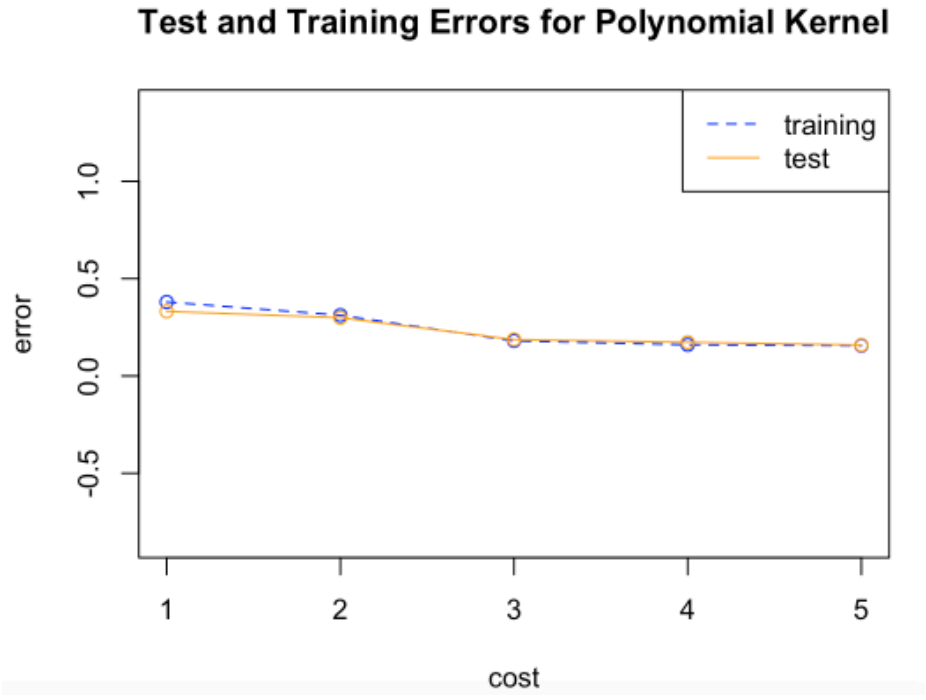
b) Radial Kernel:

```
> radial_test_error
[1] 0.3551401869 0.1682242991 0.1775700935 0.1635514019 0.1728971963
> radial_train_error
[1] 0.3983644860 0.1705607477 0.1577102804 0.1518691589 0.1495327103
```



Polynomial Kernel:

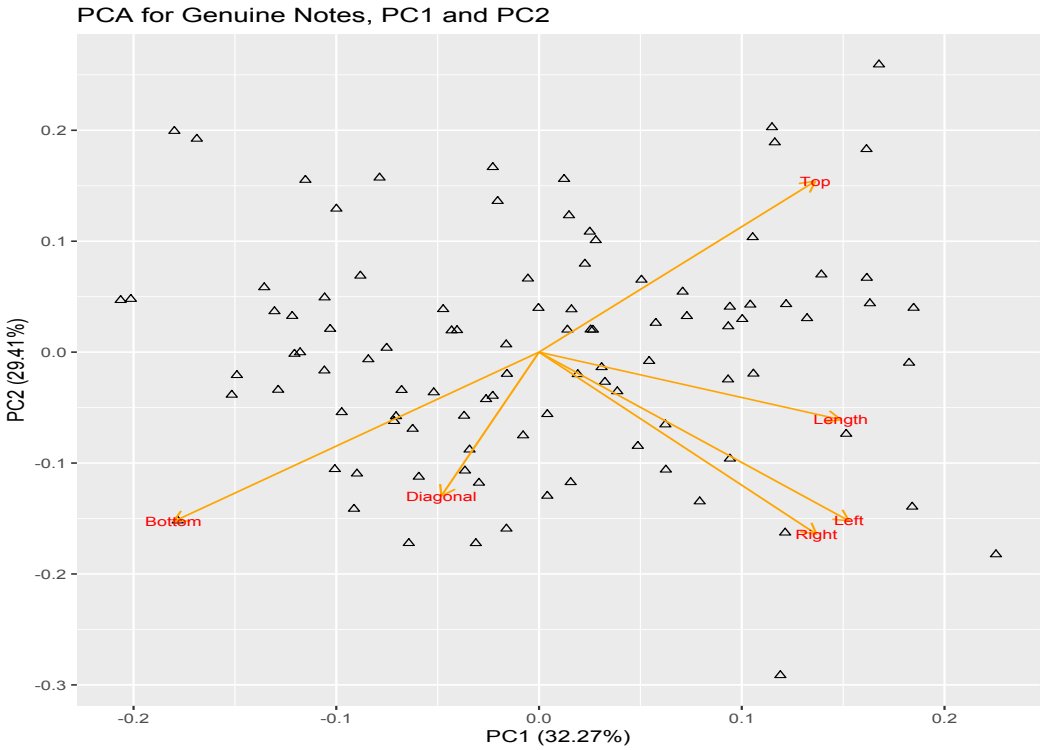
```
> poly_test_error
[1] 0.3317757009 0.2990654206 0.1869158879 0.1728971963 0.1588785047
> poly_train_error
[1] 0.3796728972 0.3119158879 0.1799065421 0.1600467290 0.1553738318
```

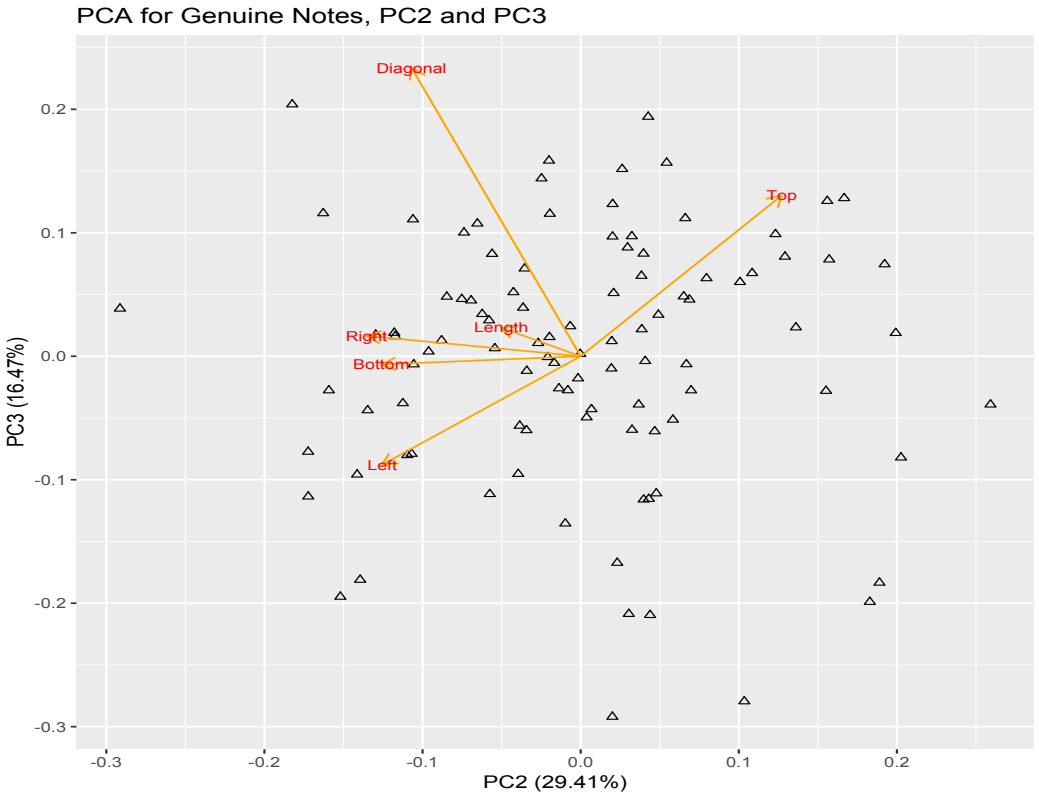


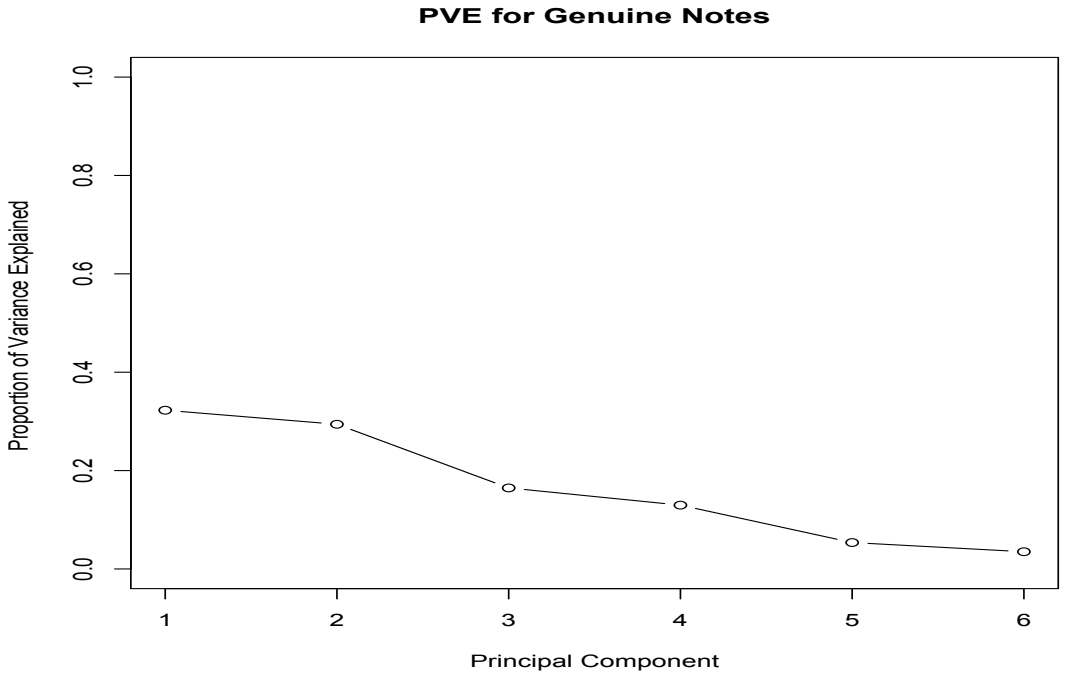
Q5. Using the 'banknote' dataset from 'mclust' package, we identified the principal components for 100 genuine notes, 100 counterfeit notes and 200 total notes.

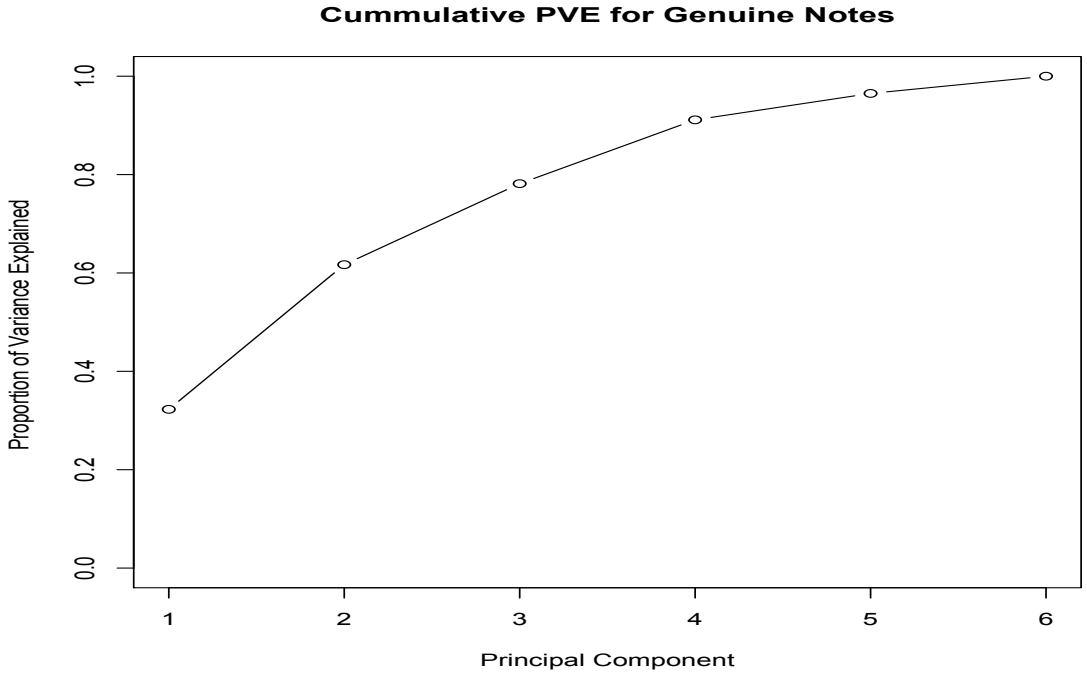
Sol:

- **For Genuine Notes:**

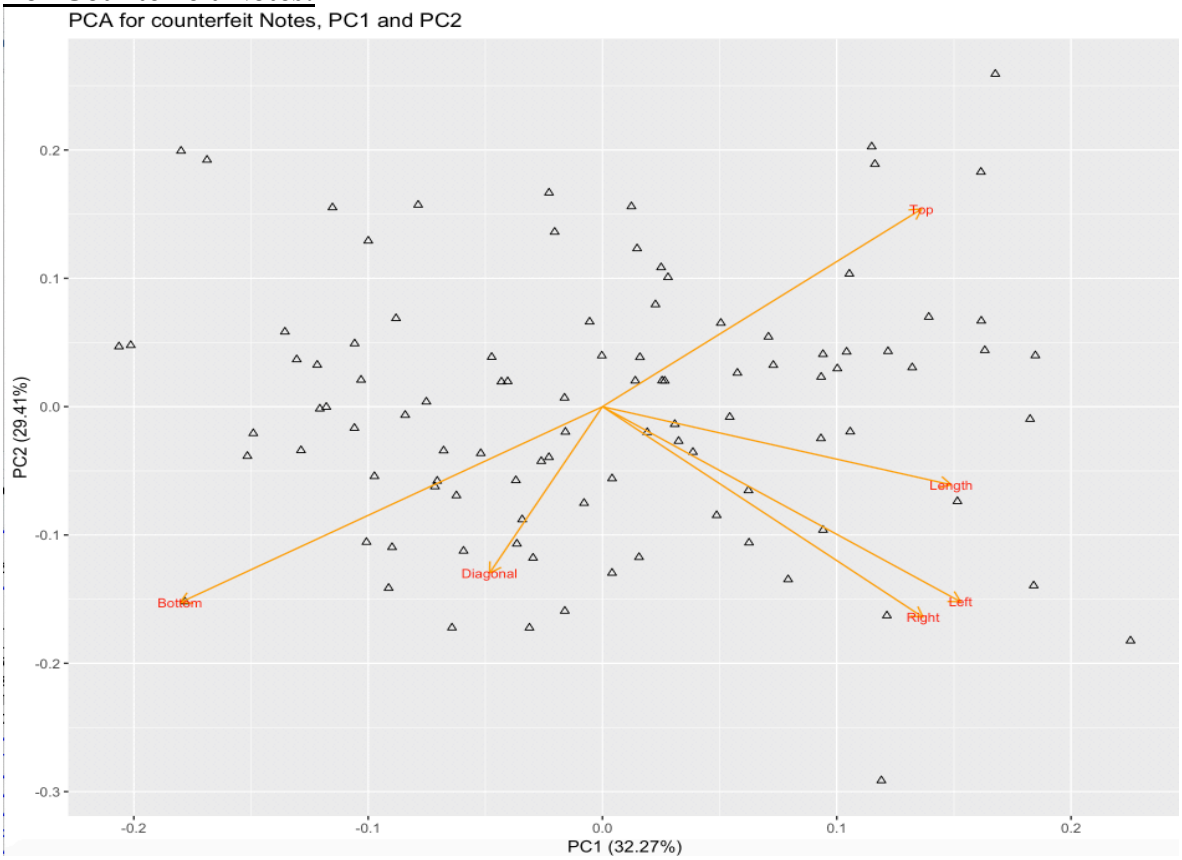


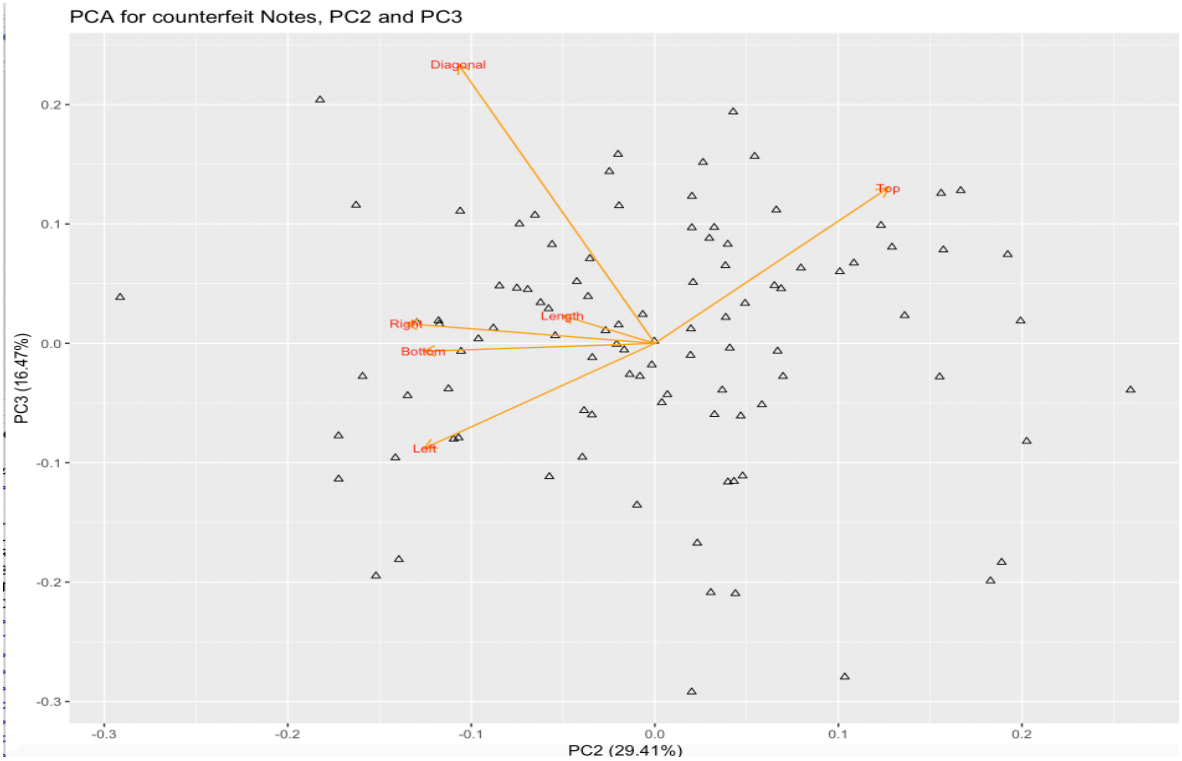


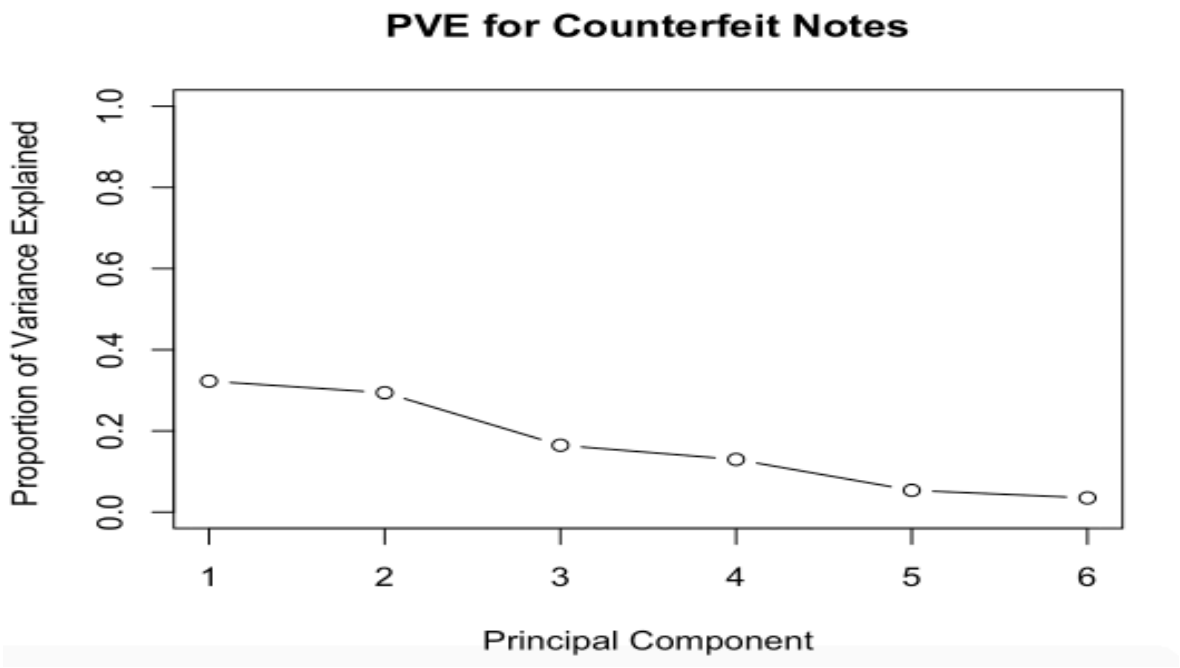

PVE for Genuine Notes
Proportion of Variance Explained
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5
6
Principal Component

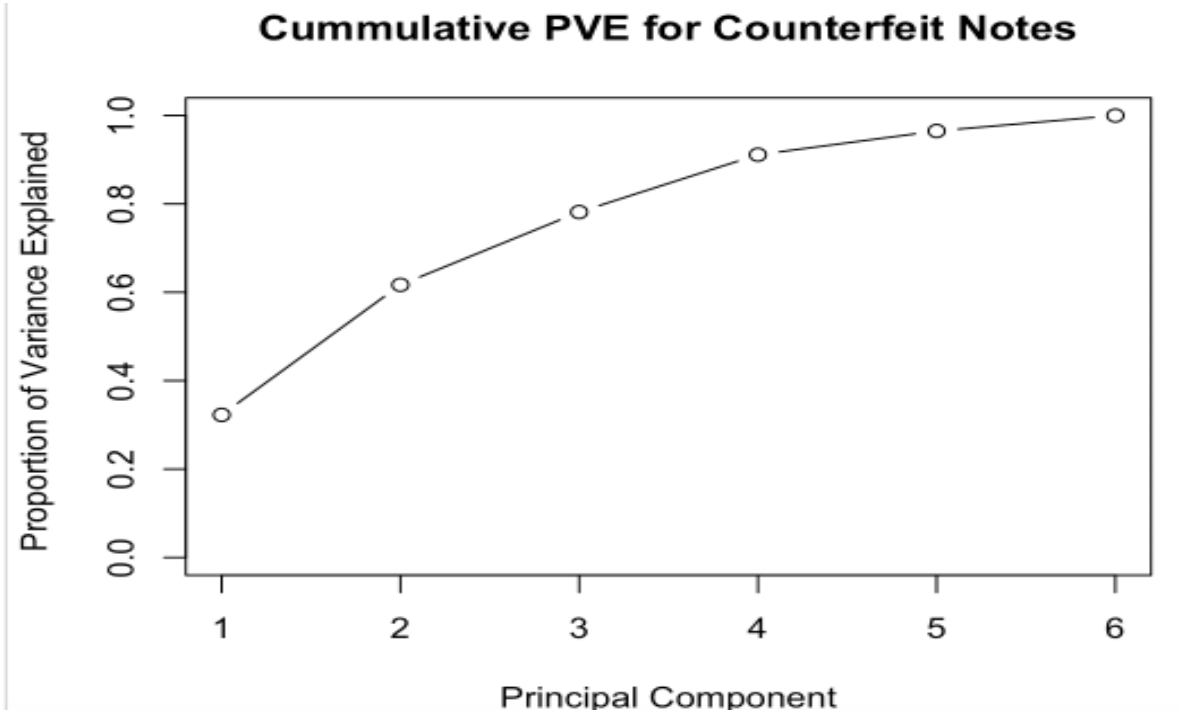

Cummulative PVE for Genuine Notes
Proportion of Variance Explained
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5
6
Principal Component

- **For Counterfeit Notes:**

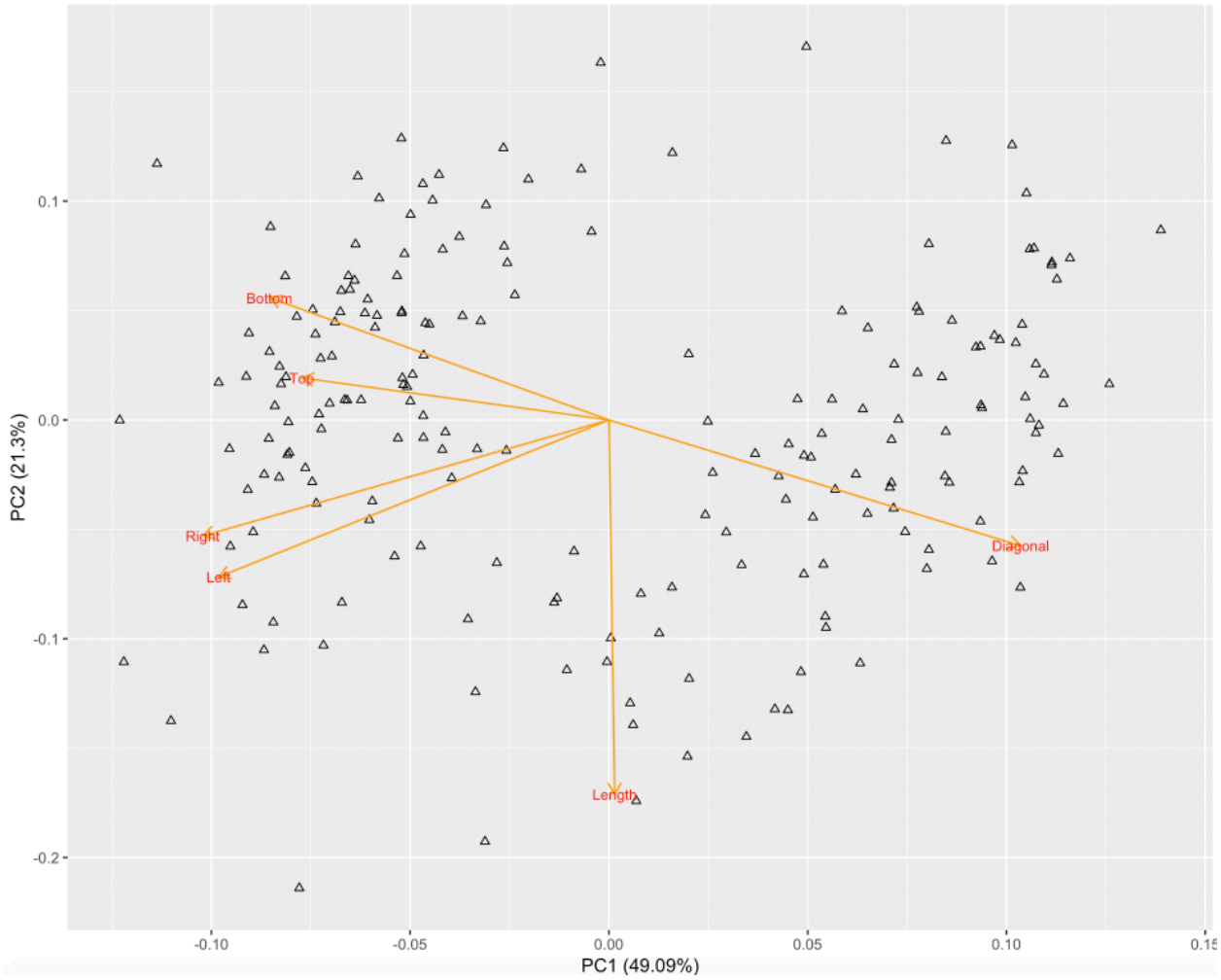


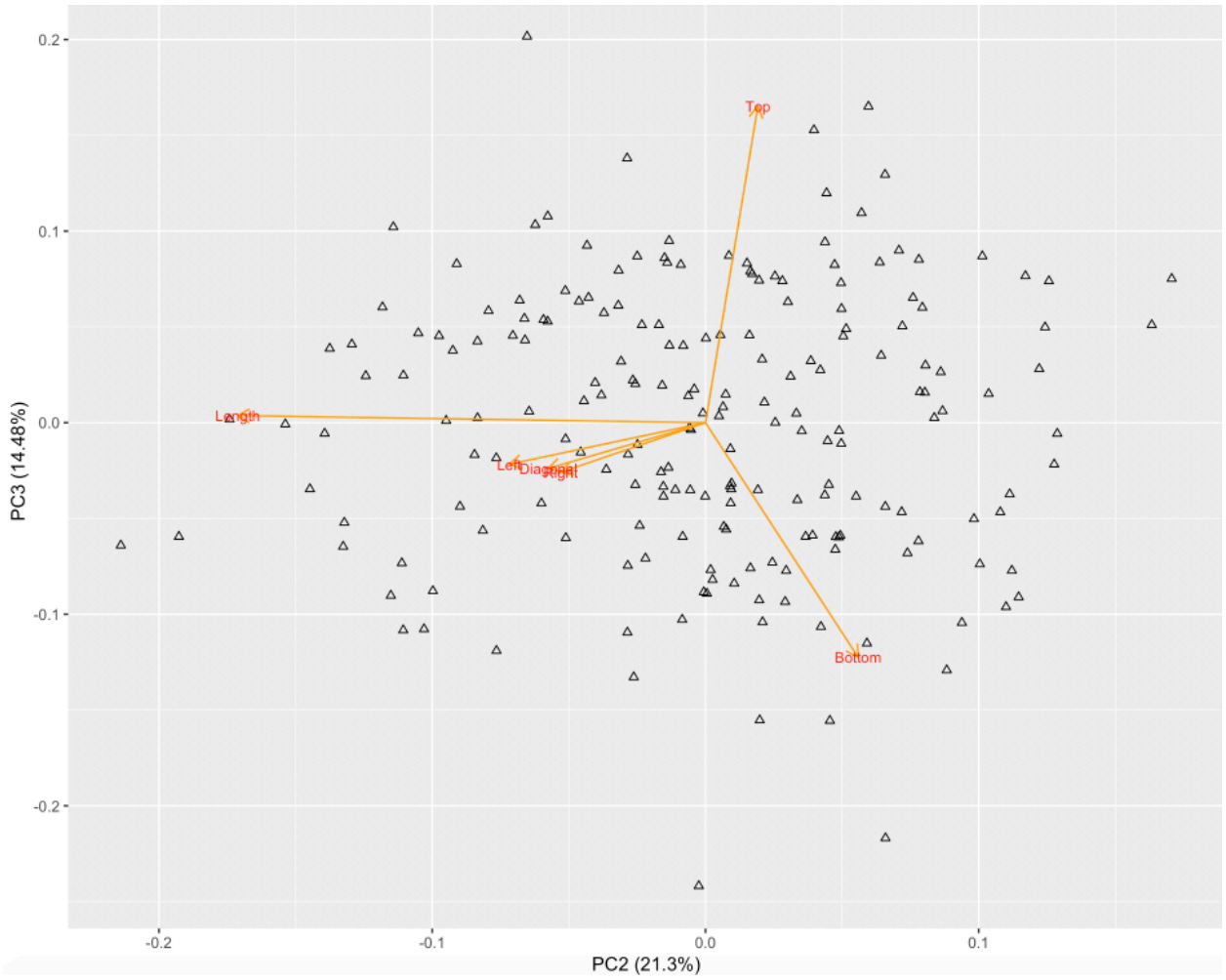


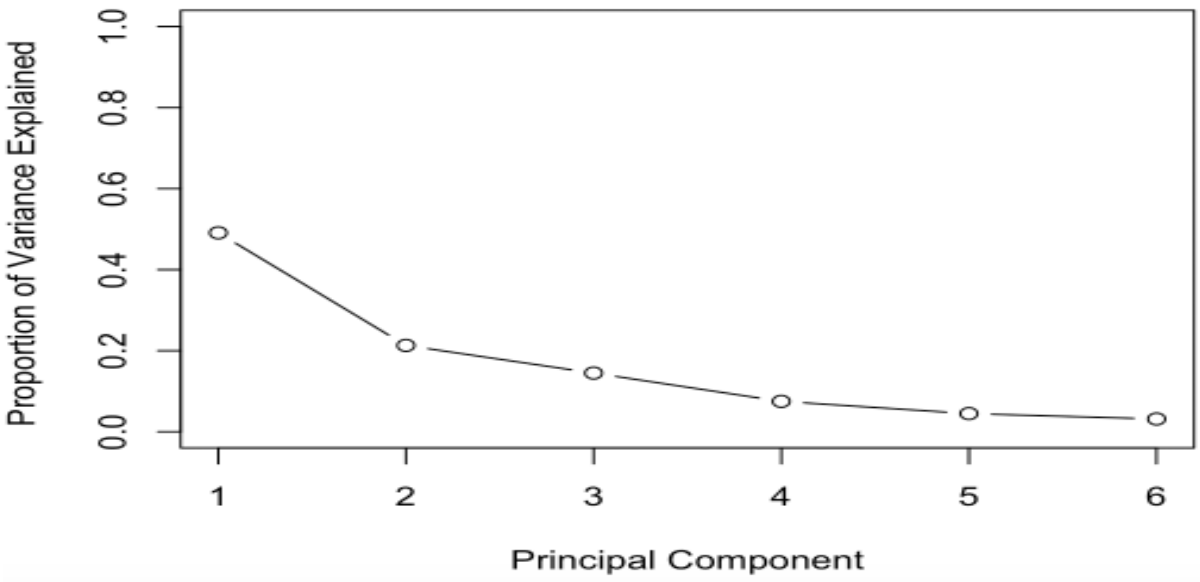

PVE for Counterfeit Notes
Proportion of Variance Explained
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5
6
Principal Component

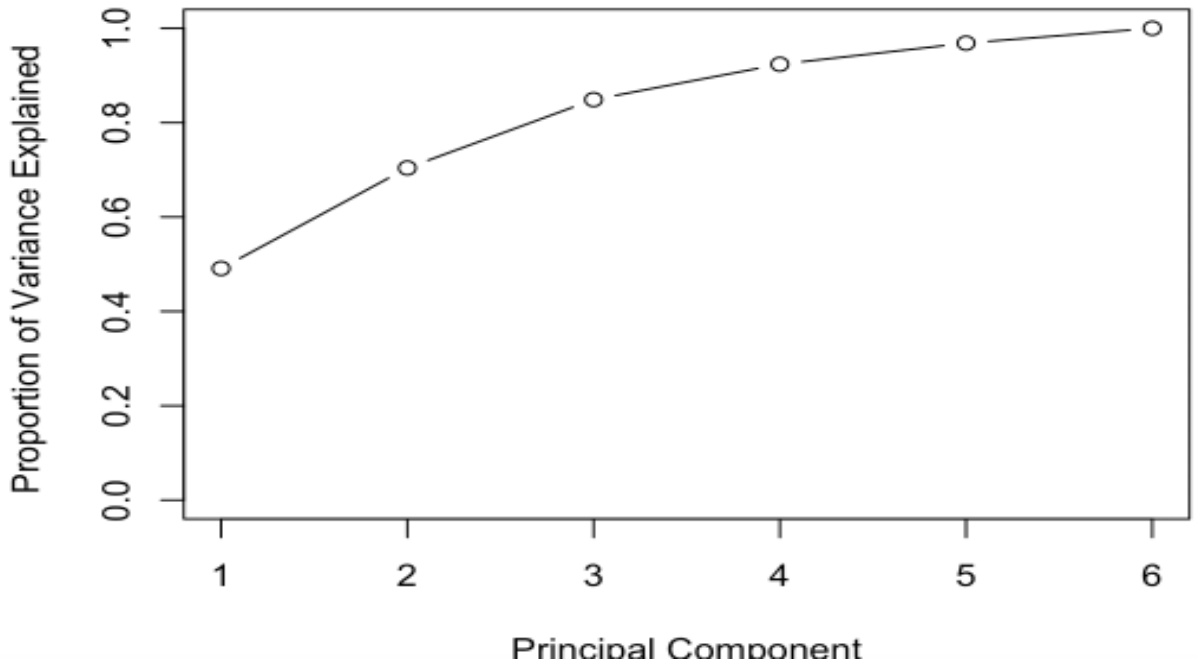

Cummulative PVE for Counterfeit Notes
Proportion of Variance Explained
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5
6
Principal Component

- **For Combined (Genuine and Counterfeit) Notes:**




Top
Length
Left
Diagonal
Right
Bottom
PC3 (14.48%)
PC2 (21.3%)
0.2
0.1
0.0
-0.1
-0.2
-0.2
-0.1
0.0
0.1


Proportion of Variance Explained
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5
6
Principal Component



- **Observation**: It was observed that PC1, PC2 and PC3 were needed to describe 78% of the data when analyzing genuine and counterfeit notes separately. However, PC1 and PC2 were sufficient to describe 75% of the data of the combined dataset.