ECE625 Assignment_3
Yilong Wu 1679741

- **Q1**

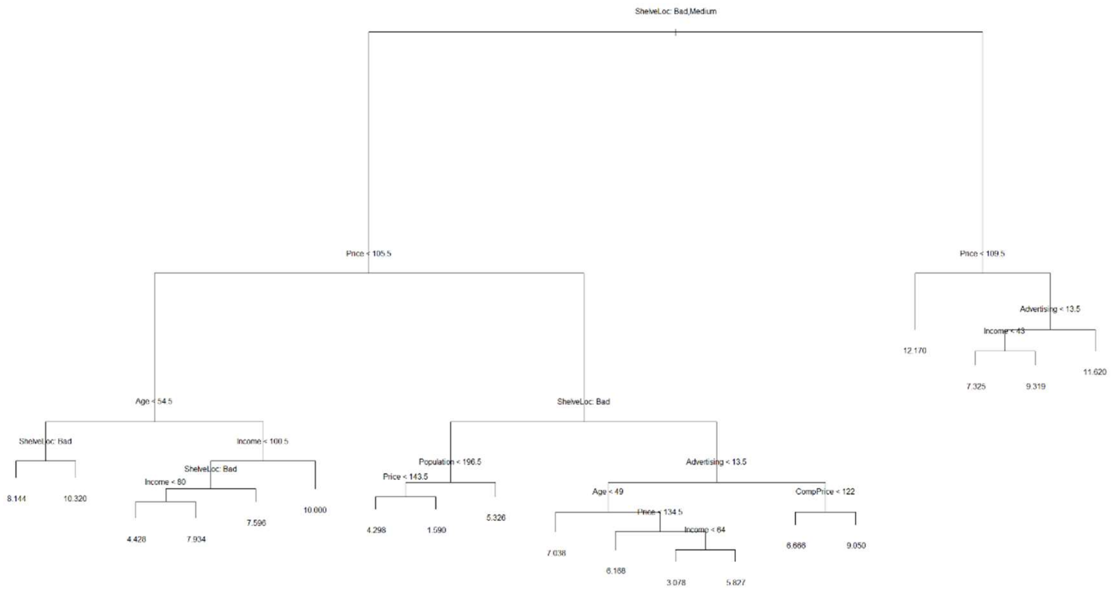
  Majority vote: 6 for Red vs 4 for Green

  Average probability: P (Class is Red|X) = 4.5/10 = 0.45 -> Green
  Classy X as Green as the average of 10 probabilities is 0.45

- **Q2**

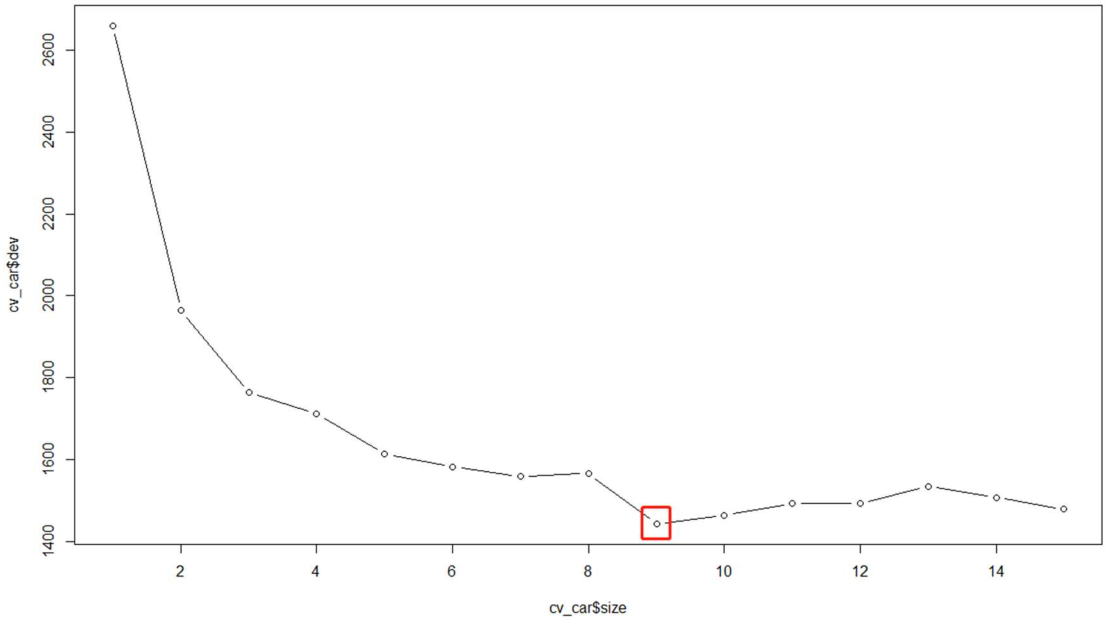
  (b)

```
Regression tree:
tree(formula = Sales ~ ., data = train)
Variables actually used in tree construction:
[1] "ShelveLoc"   "Price"       "Age"         "Income"      "Population"  "Advertising"
[7] "CompPrice"
Number of terminal nodes:  19
Residual mean deviance:  2.626 = 790.4 / 301
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-4.50900 -1.05900 -0.04286  0.00000  1.07400  4.81400
```




```
> mean((yhat - test$Sales)^2)
[1] 3.866483
```

(c)




```
> mean((yhat - test$Sales)^2)
[1] 4.122399
```

It doesn’t improve the test MSE

(d)

```
> mean((yhat_bag - test$Sales)^2)
[1] 1.60762
```

The test MSE is 1.60762

```
> importance(bagging)
                %IncMSE IncNodePurity
CompPrice    29.1452483     264.76370
Income        8.2806986     142.86126
Advertising  24.7170306     210.85887
Population   -0.7464594      83.71752
Price        65.8171044     681.64067
ShelveLoc    74.7849598     872.72059
Age          21.2643470     223.23974
Education    -0.1822062      77.40663
Urban        -1.2880258      11.68213
US            6.0705093      16.33858
```

The Price and Shelveloc are the most important.

(e)

```
> mean((yhat.rf - test$Sales)^2)
[1] 1.963152
```

```
> importance(rf)
              %IncMSE IncNodePurity
CompPrice   18.7029963     227.65489
Income       4.0711087     187.21247
Advertising 17.1762212     230.24372
Population   0.9946491     160.10621
Price       46.8862673     589.82060
ShelveLoc   50.1139797     661.02731
Age         16.4790350     284.40563
Education    2.7871598     110.78218
Urban       -1.2545889      20.83014
US           5.3623412      34.78439
```

The ShelveLoc is the most important.

- **Q3**

(b)

```
Classification tree:
tree(formula = Purchase ~ ., data = train)
Variables actually used in tree construction:
[1] "LoyalCH"     "SalePriceMM" "PriceDiff"   "DiscCH"
Number of terminal nodes:  7
Residual mean deviance:  0.7643 = 606.1 / 793
Misclassification error rate: 0.185 = 148 / 800
```

Training error rate: 18.5%

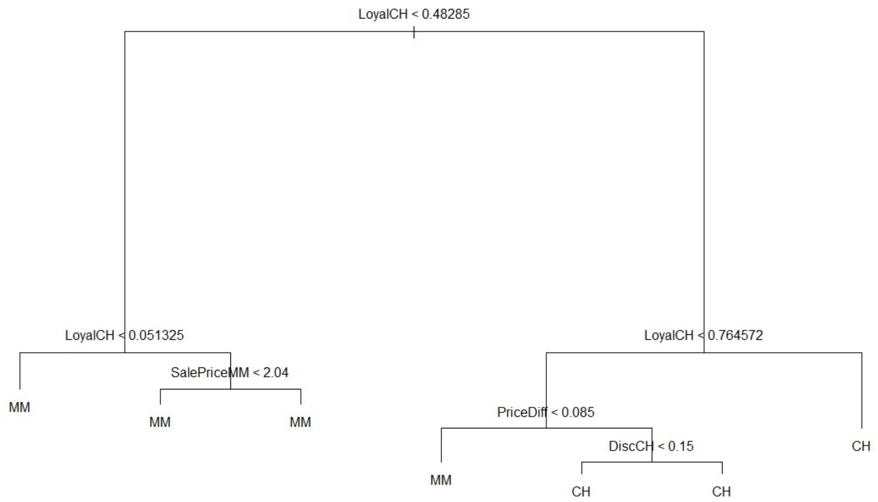
7 terminal nodes

(c)

```
> tree
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

 1) root 800 1059.00 CH ( 0.62500 0.37500 )
   2) LoyalCH < 0.48285 290  320.50 MM ( 0.24138 0.75862 )
     4) LoyalCH < 0.051325 53    0.00 MM ( 0.00000 1.00000 ) *
     5) LoyalCH > 0.051325 237  287.70 MM ( 0.29536 0.70464 )
      10) SalePriceMM < 2.04 132  131.00 MM ( 0.19697 0.80303 ) *
      11) SalePriceMM > 2.04 105  142.80 MM ( 0.41905 0.58095 ) *
   3) LoyalCH > 0.48285 510  443.10 CH ( 0.84314 0.15686 )
     6) LoyalCH < 0.764572 246  295.60 CH ( 0.71138 0.28862 )
      12) PriceDiff < 0.085 86  119.20 MM ( 0.48837 0.51163 ) *
      13) PriceDiff > 0.085 160  145.20 CH ( 0.83125 0.16875 )
        26) DiscCH < 0.15 134  134.70 CH ( 0.79851 0.20149 ) *
        27) DiscCH > 0.15 26    0.00 CH ( 1.00000 0.00000 ) *
     7) LoyalCH > 0.764572 264   78.51 CH ( 0.96591 0.03409 ) *
```

13) PriceDiff: Number of observations: 160, deviance: 145.20, overall prediction: CH, fraction of observation in this branch that takes on values of “CH” and “MM” = (0.83125, 0.16875)

(d)



Interpretation: the most important indicator of "purchase" appears to be "loyalch". its value of less than 0.48285 is pretty much going to be classified as "MM". value >= 0.764572 as "CH". and values below 0.764572 will be classified based on "Pricediff" predictor.

(e)

```
> table(pred, purchase.test)
    purchase.test
pred  CH  MM
  CH 116  13
  MM  37 104
```

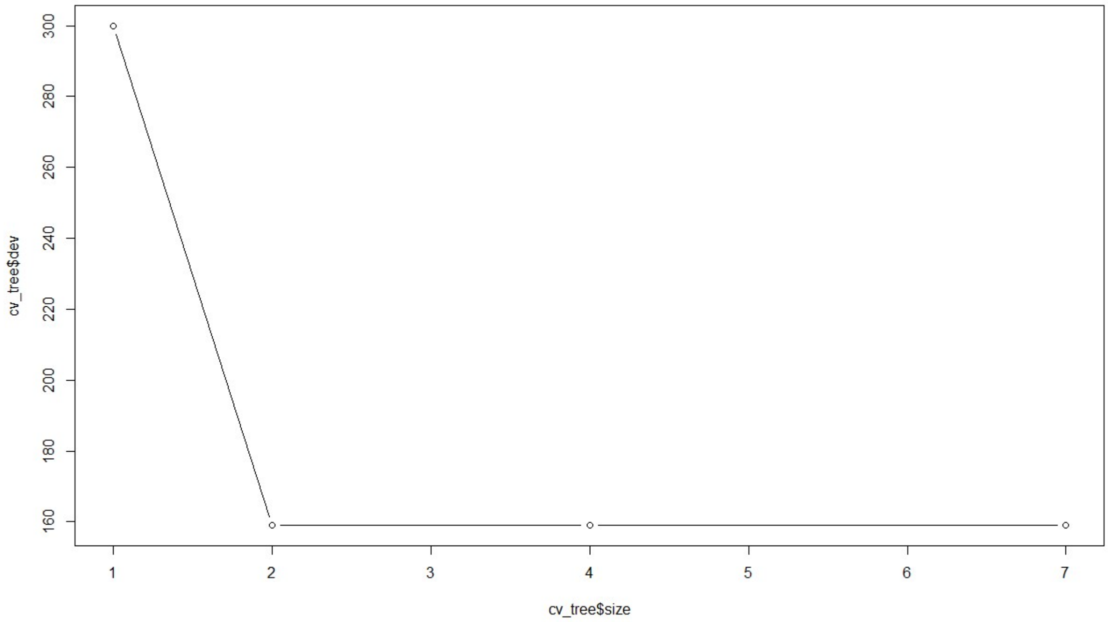
```
> test_error
[1] 18.52
```

The test error is 18.52%

(f)

```
> cv_tree
$size
[1] 7 4 2 1

$dev
[1] 159 159 159 300

$k
[1] -Inf    0    1  150

$method
[1] "misclass"

attr(,"class")
[1] "prune"         "tree.sequence"
```

(g)



(h) tree size of 2 has the lowest cross-validation

(i)

```
> summary(pruned)

Classification tree:
snip.tree(tree = tree, nodes = 2:3)
Variables actually used in tree construction:
[1] "LoyalCH"
Number of terminal nodes:  2
Residual mean deviance:  0.957 = 763.7 / 798
Misclassification error rate: 0.1875 = 150 / 800
```

(j)

Training error of pruned tree is 18.75%, for unpruned tree is 18.5%, pruned tree is higher

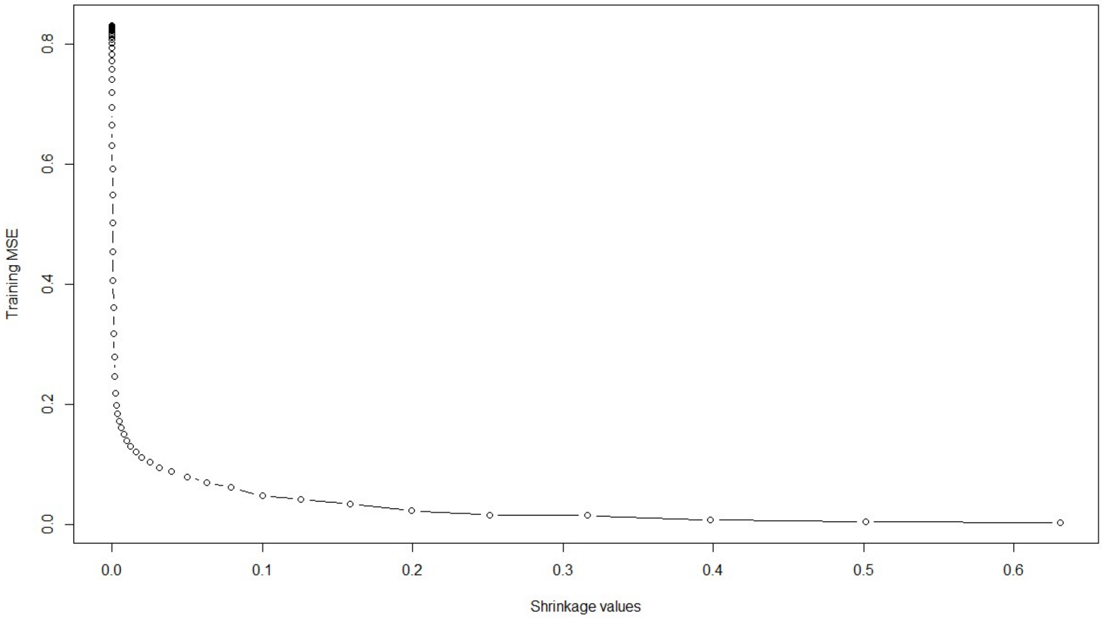
(k)

	Model_type	Test_Error_rate
1	unPruned	18.52
2	Pruned w/ Term.nodes=2	20.00

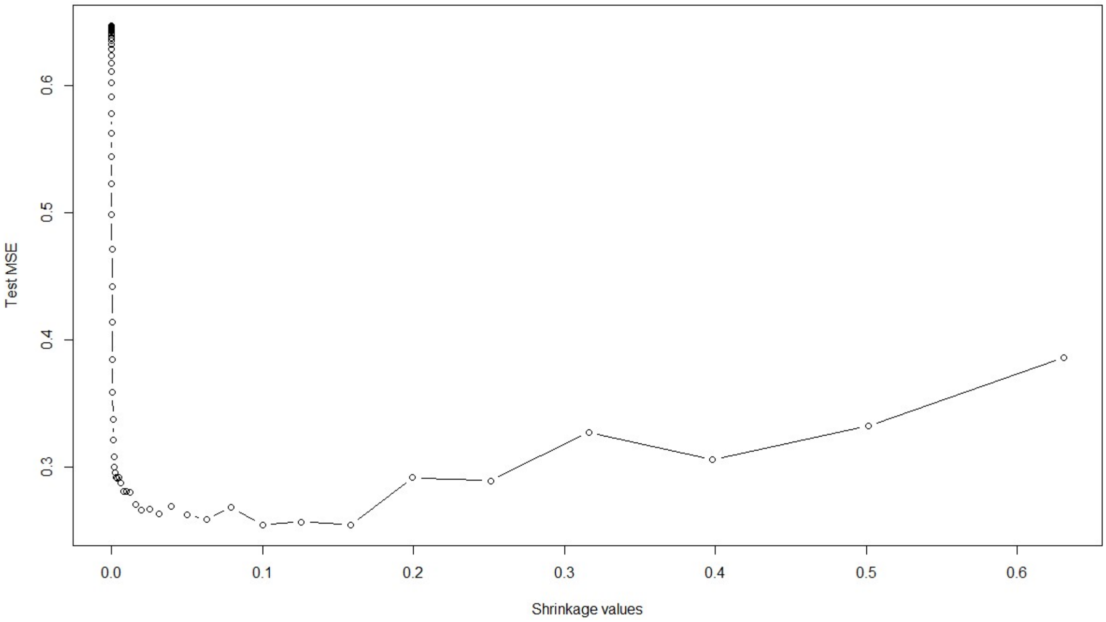
Pruned is higher

- **Q4**

(c)



(d)

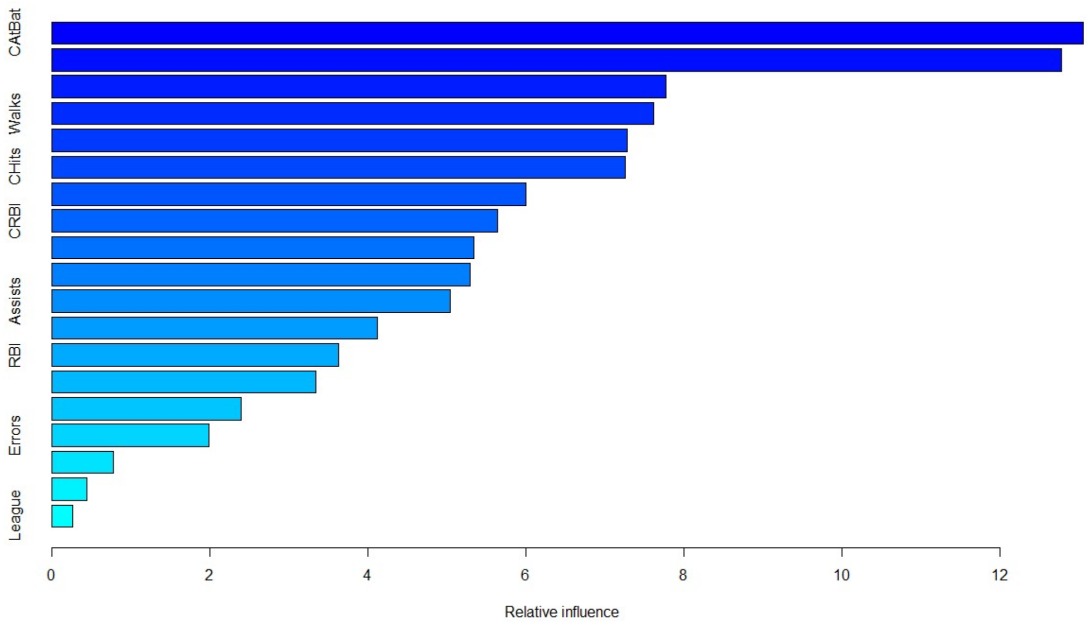


(e)

Ridge Regression test MSE: 0.457

Lasso Regression test MSE: 0.470

(f)




```
                  var    rel.inf
CAtBat         CAtBat 13.0565086
CRuns           CRuns 12.7788901
PutOuts       PutOuts  7.7721538
Walks           Walks  7.6124721
CWalks         CWalks  7.2840637
CHits           CHits  7.2525690
Hits             Hits  5.9992368
CRBI             CRBI  5.6443918
CHmRun         CHmRun  5.3353727
Years           Years  5.2916655
Assists       Assists  5.0413226
AtBat           AtBat  4.1140300
RBI               RBI  3.6218812
HmRun           HmRun  3.3355243
Runs             Runs  2.3947275
Errors         Errors  1.9857975
Division     Division  0.7797455
NewLeague NewLeague  0.4403400
League         League  0.2593074
```

CAtBat is the most important variable

(g)

```
> bagg_test_mse
[1] 0.2331601
```

The test MSE is 0.2331601

- **Q5**

(a)

(a) Forward: $R(\theta) = \sum_{i=1}^{N} R_i = \sum_{i=1}^{N} \sum_{k=1}^{K} (y_{ik} - f_k(x_i))^2$

Backward: $\frac{\partial R_i}{\partial \beta_{km}} = -2(y_{ik} - f_k(x_i)) g_k'(\beta_k^T z_i) z_{mi}$

$$\frac{\partial R_i}{\partial \alpha_{ml}} = -\sum_{k=1}^{K} 2(y_{ik} - f_k(x_i)) g_k'(\beta_k^T z_i) \beta_{km} \sigma'(\alpha_m^T x_i) x_{il}$$

$$\beta_{km}^{(r+1)} = \beta_{km}^{r} - \gamma_r \sum_{i=1}^{N} \frac{\partial R_i}{\partial \beta_{km}^{r}}$$

$$\alpha_{ml}^{(r+1)} = \alpha_{ml}^{r} - \gamma_r \sum_{i=1}^{N} \frac{\partial R_i}{\partial \alpha_{ml}^{r}}$$

(b)

(b) Forward. $Z_m = \sigma(\alpha_{0m} + \alpha_m^T X)$, $m = 1, \cdots, M$

$T_k = \beta_{0k} + \beta_k^T Z$, $k = 1, \cdots, K$

$f_k(X) = g_k(T)$, $k = 1, \cdots, K$

where $Z = (Z_1, Z_2, \cdots Z_M)$ and $T = (T_1, T_2, \cdots T_K)$

$$R_\theta = \sum_{i=1}^{N} R_i = -\sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log f_k(x_i)$$

Backwork: $Z_m = \sigma(\alpha_{0m} + \alpha_m^T x_i)$

$$g_k(T) = \frac{e^{T_k}}{\sum_{l=1}^{K} e^{T_l}}$$

$$R(\theta) \equiv \sum_{i=1}^{N} R_i = \sum_{i=1}^{N} \sum_{k=1}^{K} (y_{ik} - f_k(x_i))^2$$

with derivatives

$$\frac{\partial R_i}{\partial \beta_{km}} = -2(y_{ik} - f_k(x_i)) g_k'(\beta_k^T z_i) z_{mi}$$

$$\frac{\partial R_i}{\partial \alpha_{ml}} = -\sum_{k=1}^{K} 2(y_{ik} - f_k(x_i)) g_k'(\beta_k^T z_i) \beta_{km} \sigma'(\alpha_m^T x_i) x_{il}$$

Given these derivatives, a grandiene descent update at the (r+1)st iteration has the form,

$$\beta_{km}^{(r+1)} = \beta_{km}^{(r)} - \gamma_r \sum_{i=1}^{N} \frac{\partial R_i}{\partial \beta_{km}^{(r)}},$$

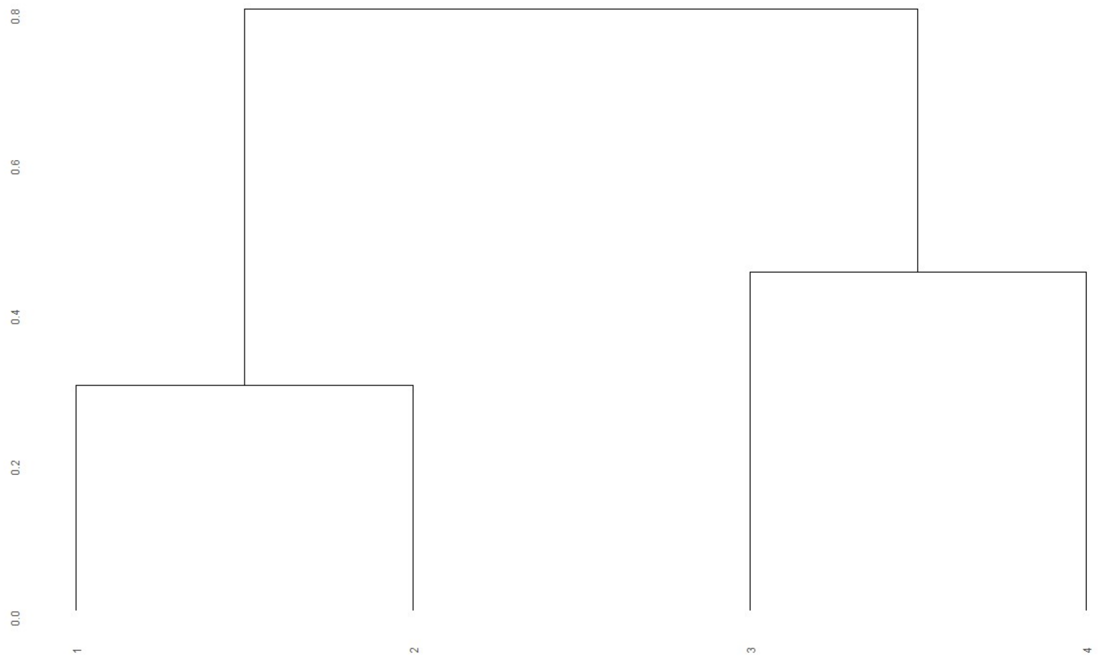
$$\alpha_{ml}^{(r+1)} = \alpha_{ml}^{(r)} - \gamma_r \sum_{i=1}^{N} \frac{\partial R_i}{\partial \alpha_{ml}^{(r)}},$$

- **Q6**

(a)

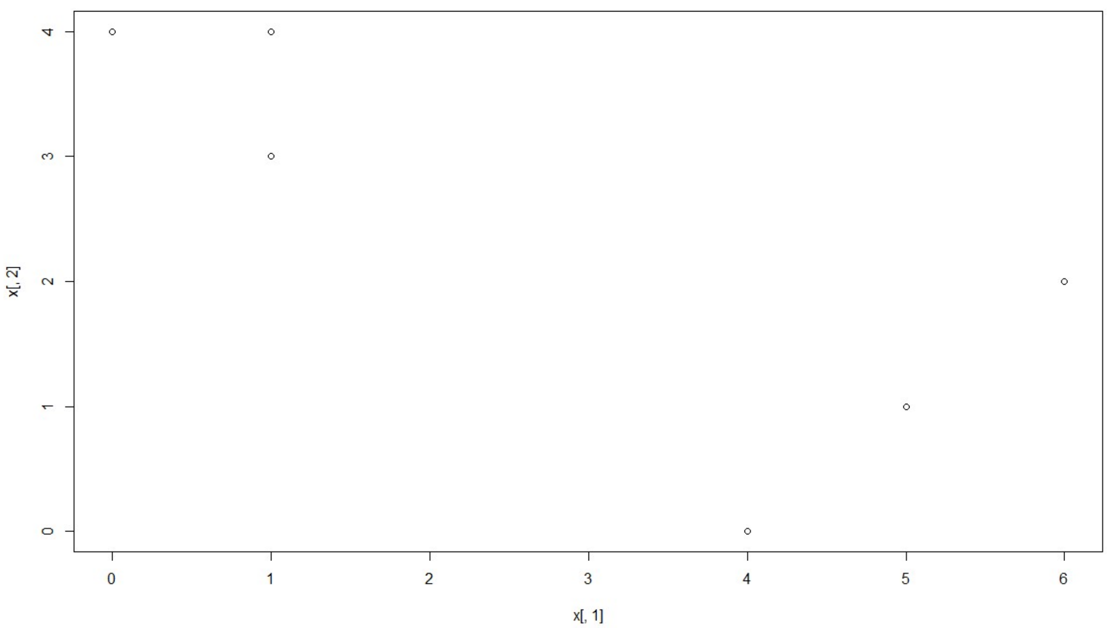
The number of hidden units is 5 perform best

- **Q7**

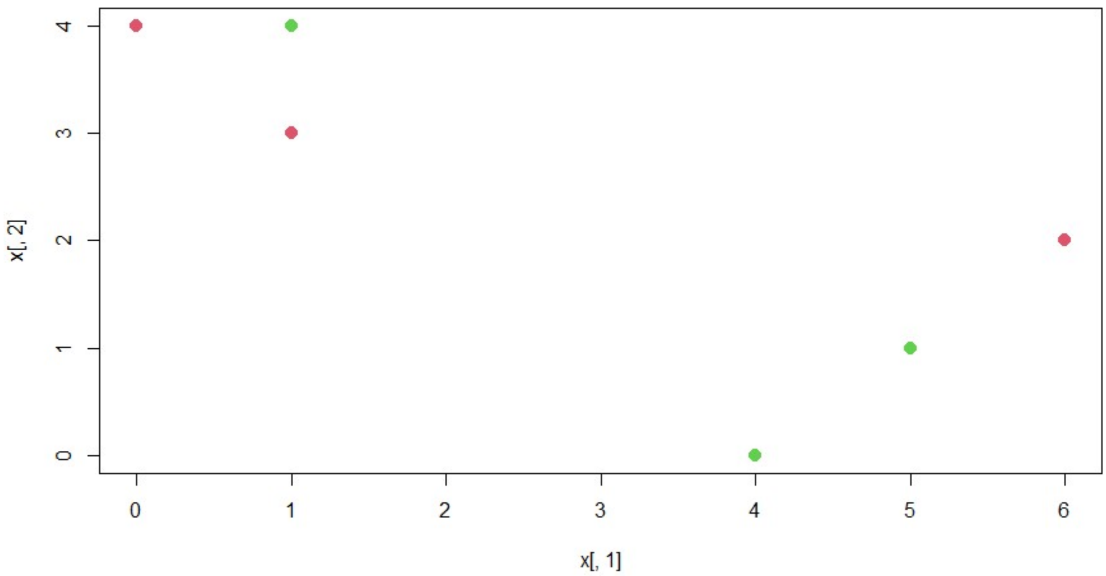


- **Q8**

(a)



(b)

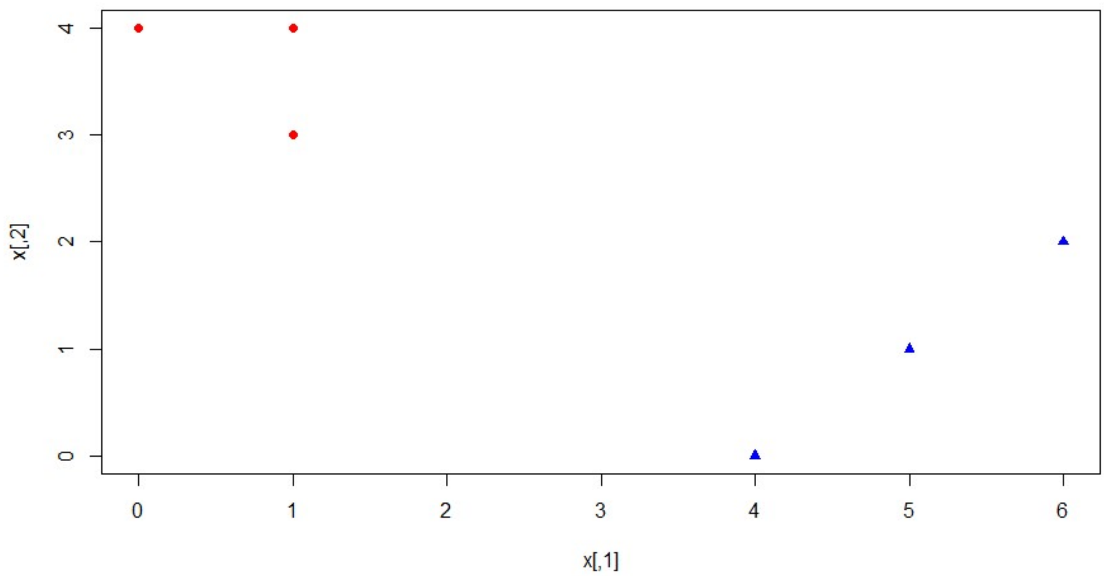


(c)

```
> centroids
  Cluster       V1       V2
1       1 2.333333 3.000000
2       2 3.333333 1.666667
```

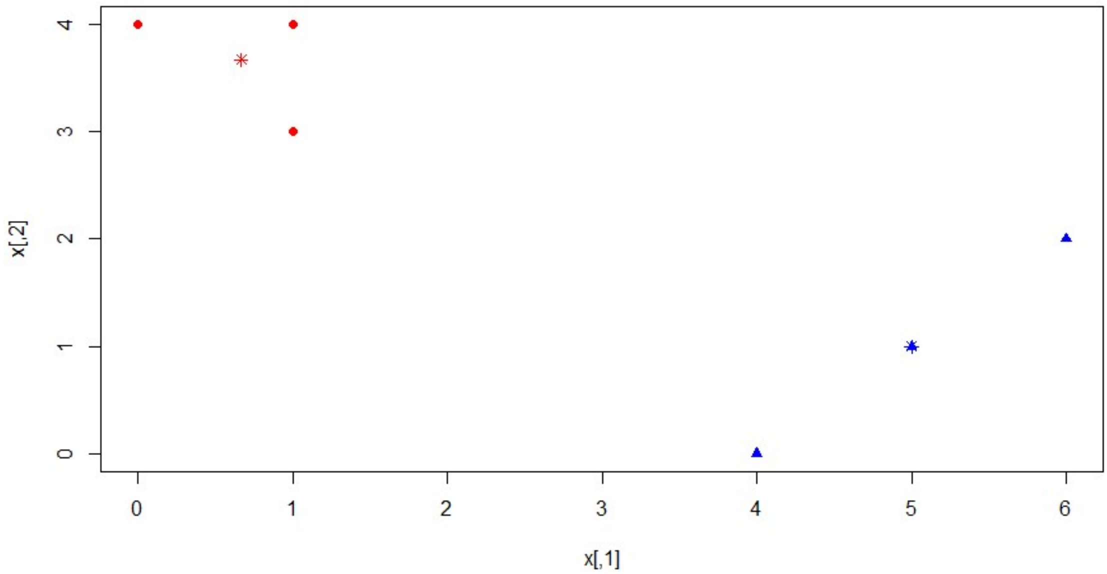
(d)

```
> clusters
[1] 1 1 1 2 2 2
Levels: 1 2
```



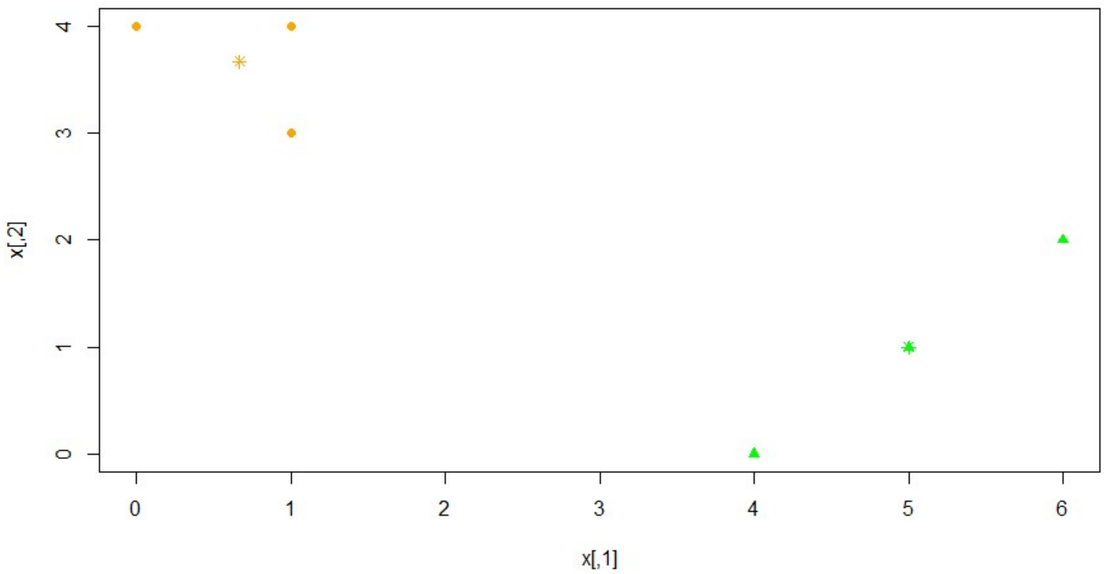
(e)

```
> centroids
  Cluster        V1       V2
1       1 0.6666667 3.666667
2       2 5.0000000 1.000000
```



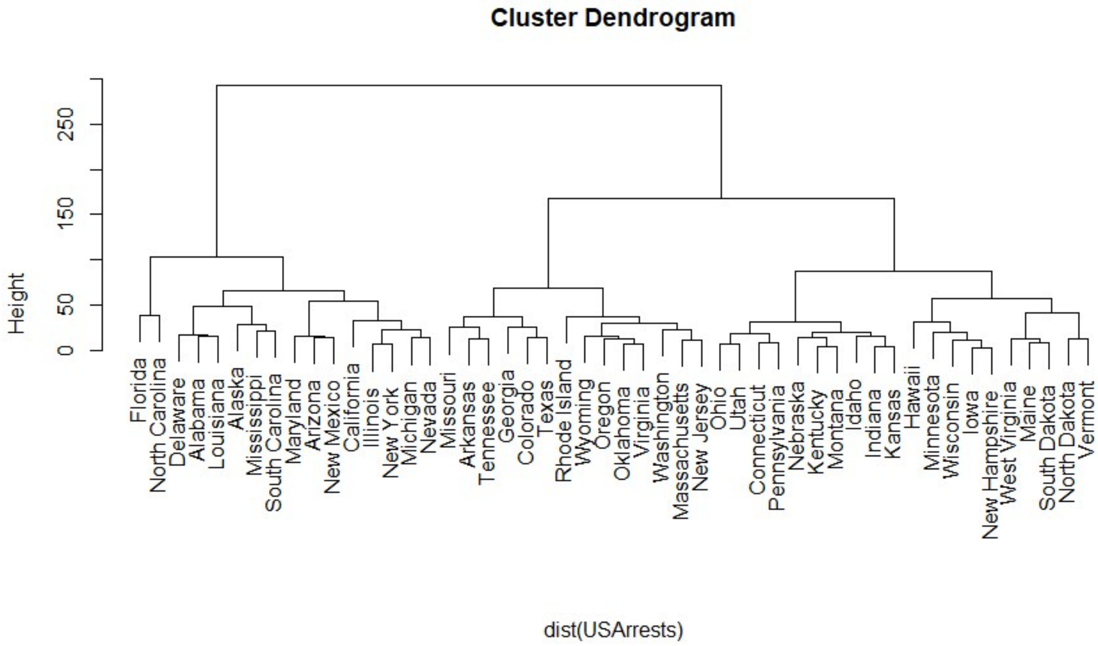

```
> clusters
[1] 1 1 1 2 2 2
Levels: 1 2
```

(f)



- **Q9**

(a)

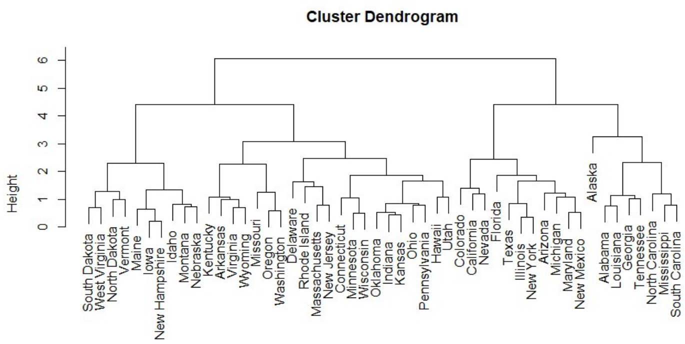


(b)

```
> cutree(complete, 3)
       Alabama         Alaska        Arizona       Arkansas     California       Colorado
             1              1              1              2              1              2
   Connecticut       Delaware        Florida        Georgia         Hawaii          Idaho
             3              1              1              2              3              3
      Illinois        Indiana           Iowa         Kansas       Kentucky      Louisiana
             1              3              3              3              3              1
         Maine       Maryland  Massachusetts       Michigan      Minnesota    Mississippi
             3              1              2              1              3              1
      Missouri        Montana       Nebraska         Nevada  New Hampshire     New Jersey
             2              3              3              1              3              2
    New Mexico       New York North Carolina   North Dakota           Ohio       Oklahoma
             1              1              1              3              3              2
        Oregon   Pennsylvania   Rhode Island South Carolina   South Dakota      Tennessee
             2              3              2              1              3              2
         Texas           Utah        Vermont       Virginia     Washington  West Virginia
             2              3              3              2              2              3
     Wisconsin        Wyoming
             3              2
```

```
> table(cutree(complete, 3))

 1  2  3
16 14 20
```

(c)



(d)

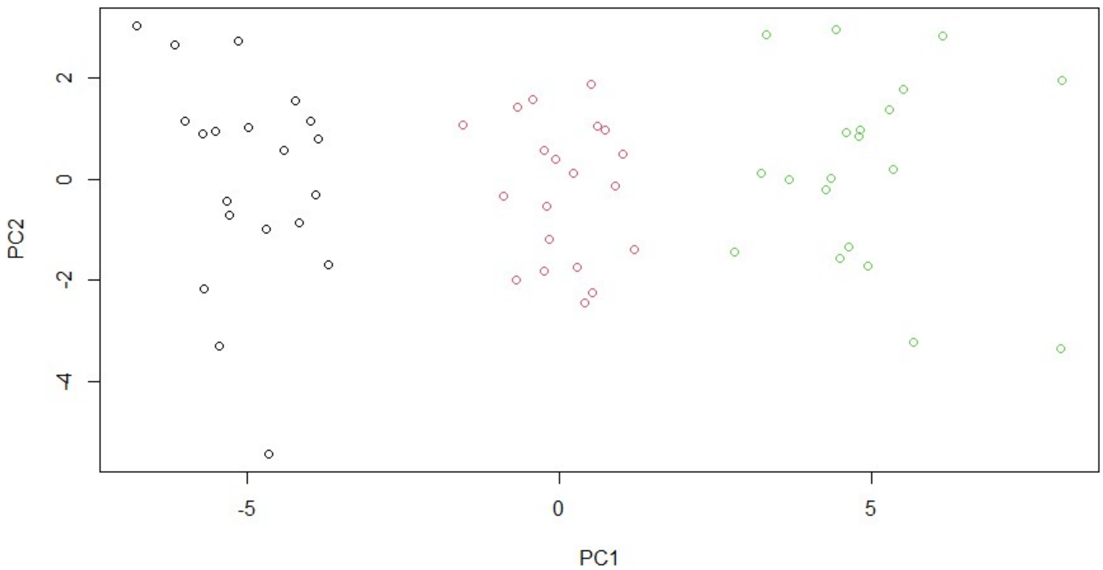
```
       Alabama         Alaska        Arizona       Arkansas     California       Colorado 
             1              1              2              3              2              2 
   Connecticut       Delaware        Florida        Georgia         Hawaii          Idaho 
             3              3              2              1              3              3 
      Illinois        Indiana           Iowa         Kansas       Kentucky      Louisiana 
             2              3              3              3              3              1 
         Maine       Maryland  Massachusetts       Michigan      Minnesota    Mississippi 
             3              2              3              2              3              1 
      Missouri        Montana       Nebraska         Nevada  New Hampshire     New Jersey 
             3              3              3              2              3              3 
    New Mexico       New York North Carolina   North Dakota           Ohio       Oklahoma 
             2              2              1              3              3              3 
        Oregon   Pennsylvania   Rhode Island South Carolina   South Dakota      Tennessee 
             3              3              3              1              3              1 
         Texas           Utah        Vermont       Virginia     Washington  West Virginia 
             2              3              3              3              3              3 
     Wisconsin        Wyoming 
             3              3 
```

```
> table(cutree(hc.s.complete, 3))

 1  2  3 
 8 11 31 
> table(cutree(hc.s.complete, 3), cutree(complete, 3))

     1  2  3
  1  6  2  0
  2  9  2  0
  3  1 10 20
```

The scaling variable will affect the maximum height of the tree graph obtained by hierarchical clustering. At a cursory level, it doesn't affect the richness of the resulting tree. However, it does affect the clustering obtained by cutting the tree graph into three clusters. In my opinion, for this data set, the data should be standardized because there are different units of measured data

- **Q10**

(b)



(c)

```
> table(res$cluster, true_class)
   true_class
     1  2  3
  1 20  0  0
  2  0 20  2
  3  0  0 18
```

We can see it clustered very good

(d)

```
> table(res$cluster, true_class)
   true_class
     1  2  3
  1  0 19 20
  2 20  1  0
```

Classify correctly

(e)

```
> table(res$cluster, true_class)
   true_class
     1  2  3
  1  0  0 14
  2  0  2  6
  3 20  0  0
  4  0 18  0
```

One class is spited to 2 classes

(f)

```
> table(res$cluster, true_class)
   true_class
     1  2  3
  1  0  0 20
  2 20  0  0
  3  0 20  0
```

PCA carries enough information, classified perfectly

(g)

```
> table(res$cluster, true_class)
   true_class
     1  2  3
  1  2 20  1
  2 18  0  0
  3  0  0 19
```

Scaling just make a little effect on the result