Homework 10

Prakash Paudyal

Please do the following problems from the text book ISLR. (use set.seed(702) to replicate your results).

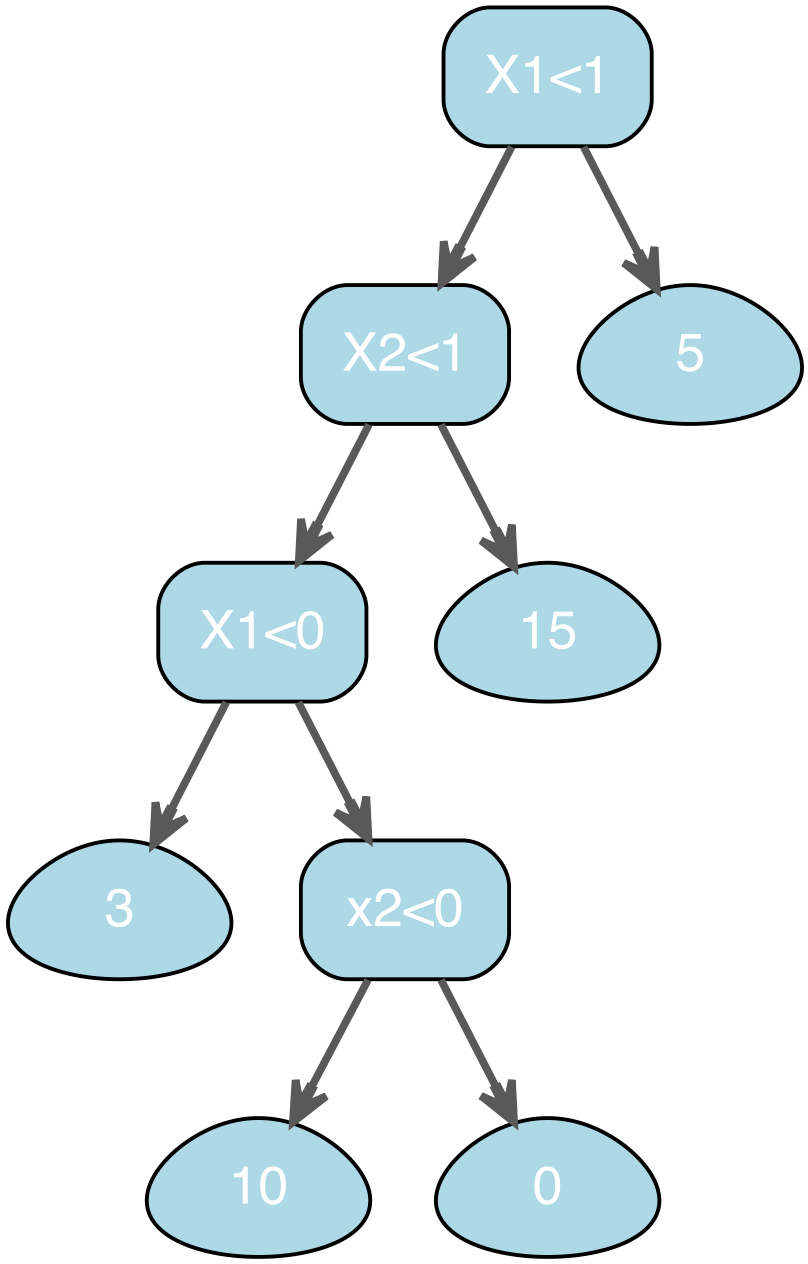
1. Question 8.4.4 pg 332

(4.)This question relates to the plots in Figure 8.12.

(a)

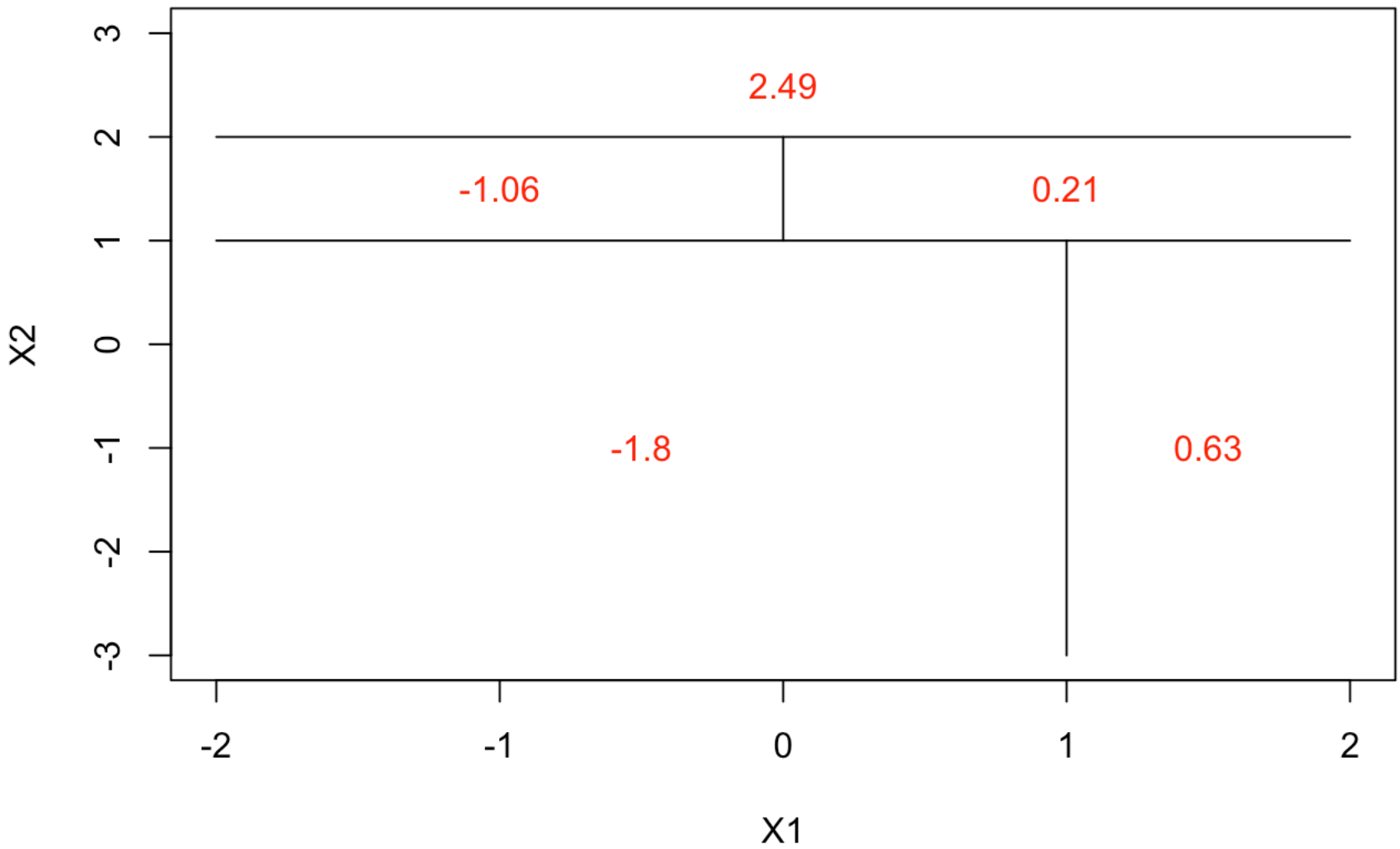
Sketch the tree corresponding to the partition of the predictor space illustrated in the left-hand panel of Figure 8.12. The numbers inside the boxes indicate the mean of Y within each region.

```
##             levelName
## 1 X1<1
## 2   ¦--X2<1
## 3   ¦    ¦--X1<0
## 4   ¦    ¦    ¦--3
## 5   ¦    ¦    °--x2<0
## 6   ¦    ¦         ¦--10
## 7   ¦    ¦         °--0
## 8   ¦    °--15
## 9   °--5
```



(b)

Create a diagram similar to the left-hand panel of Figure 8.12,using the tree illustrated in the right-hand panel of the same figure. You should divide up the predictor space into the correct regions, and indicate the mean for each region.



2. Question 8.4.8 pg 332

8. In the lab, a classification tree was applied to the Carseats data set after converting Sales into a qualitative response variable. Now we will seek to predict Sales using regression trees and related approaches, treating the response as a quantitative variable.

(a)

Split the data set into a training set and a test set.

Splited data into 60% to training set and 40% to test set.

```
## 'data.frame':    400 obs. of  11 variables:
##  $ Sales       : num  9.5 11.22 10.06 7.4 4.15 ...
##  $ CompPrice   : num  138 111 113 117 141 124 115 136 132 132 ...
##  $ Income      : num  73 48 35 100 64 113 105 81 110 113 ...
##  $ Advertising: num  11 16 10 4 3 13 0 15 0 0 ...
##  $ Population : num  276 260 269 466 340 501 45 425 108 131 ...
##  $ Price       : num  120 83 80 97 128 72 108 120 124 124 ...
##  $ ShelveLoc   : Factor w/ 3 levels "Bad","Good","Medium": 1 2 3 3 1 1 3 2 3 3 ...
##  $ Age         : num  42 65 59 55 38 78 71 67 76 76 ...
##  $ Education   : num  17 10 12 14 13 16 15 10 10 17 ...
##  $ Urban       : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 1 1 ...
##  $ US          : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 1 2 1 2 ...
```

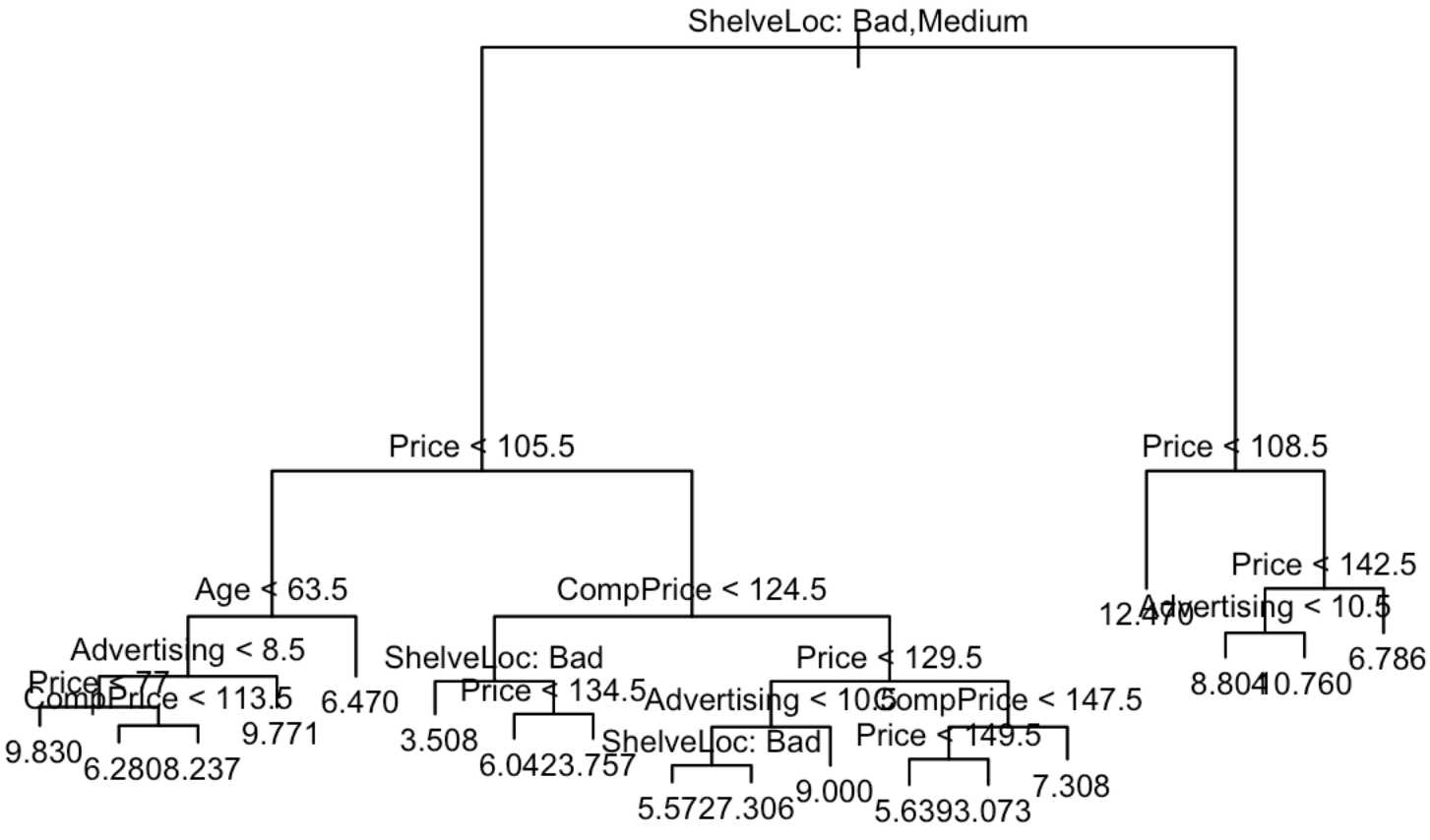
Sales	CompPrice	Income	Advertising	Population	Price	ShelveLoc	Age	Education	Urban	US
9.50	138	73	11	276	120	Bad	42	17	Yes	Yes
11.22	111	48	16	260	83	Good	65	10	Yes	Yes
10.06	113	35	10	269	80	Medium	59	12	Yes	Yes
7.40	117	100	4	466	97	Medium	55	14	Yes	Yes
4.15	141	64	3	340	128	Bad	38	13	Yes	No
10.81	124	113	13	501	72	Bad	78	16	No	Yes

(b)

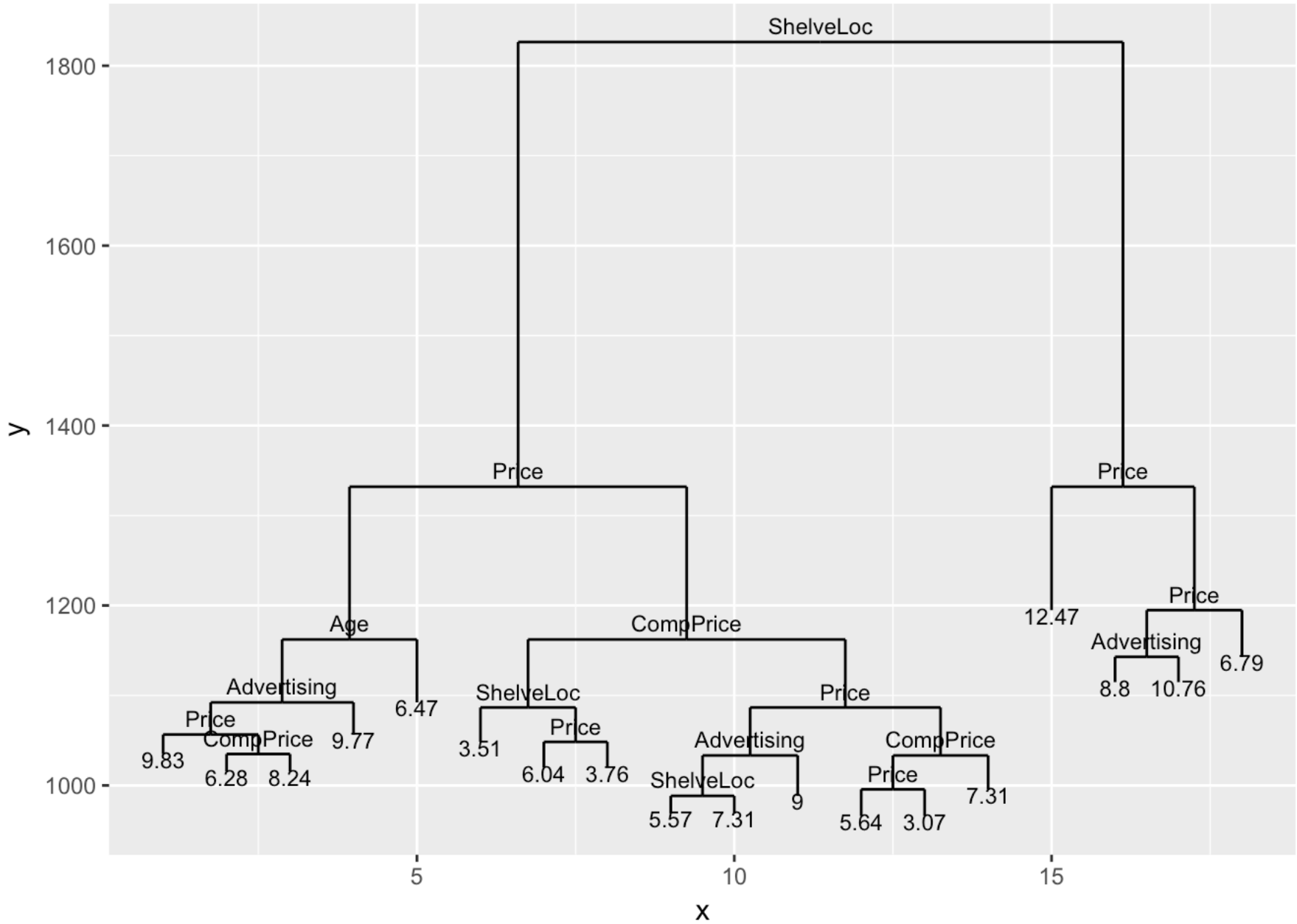
Fit a regression tree to the training set. Plot the tree, and interpret the results. What test MSE do you obtain?

```
## Warning: package 'tree' was built under R version 3.4.4
```

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = Carseats.train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "Advertising" "CompPrice"
## Number of terminal nodes:  18
## Residual mean deviance:  2.121 = 470.8 / 222
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.59700 -1.14400 -0.06052  0.00000  0.99870  3.80400
```



ggplot for carseats tree



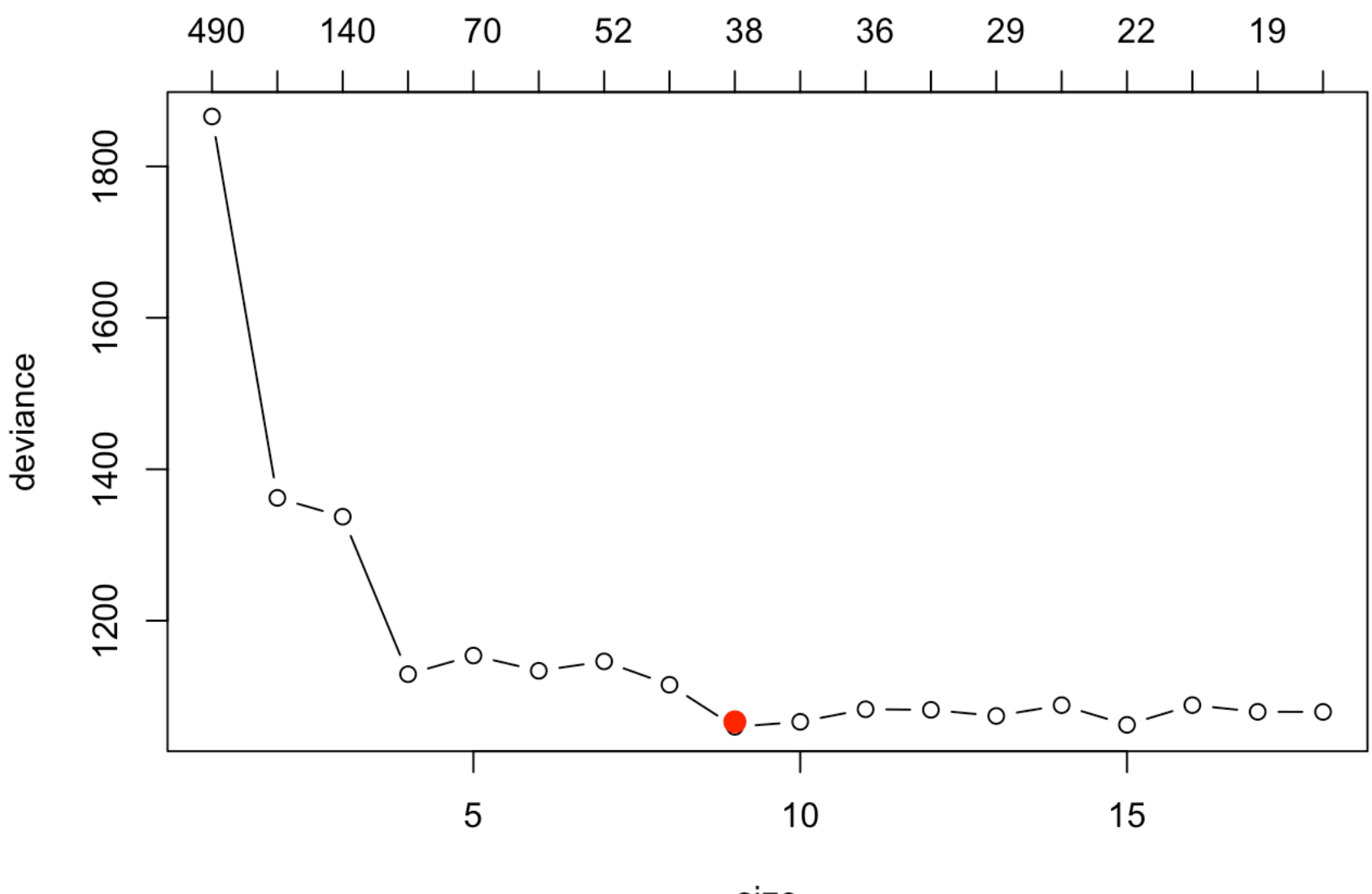

```
## [1] "Test MSE of tree model  =  4.59338130486658"
```

Discussion: Here we can see the variables ShelveLoc" , “Price”, “Age”, “Advertising” and “CompPrice” are used to grow the regression tree.The regression tree for predicting sales from carseats data based on the shelving location for car seat, price , average age of the local population, competitor price and advertising budget for company each location. Among them shelveloc variable is most important predictor which devide the predectior space into two branch according to either observation is bad or mdeium versus good.we can also see that good branch of tree has higher sales prediction than medium or bad branch side of tree.The predictor “price” is second most important predictor of carseat sales.Similarly predictor space is further devided acooriding to age and comprice and advertisiment.Tree has 18 terminal nodes or leaves which show the mean of the response for the observations that fall there.

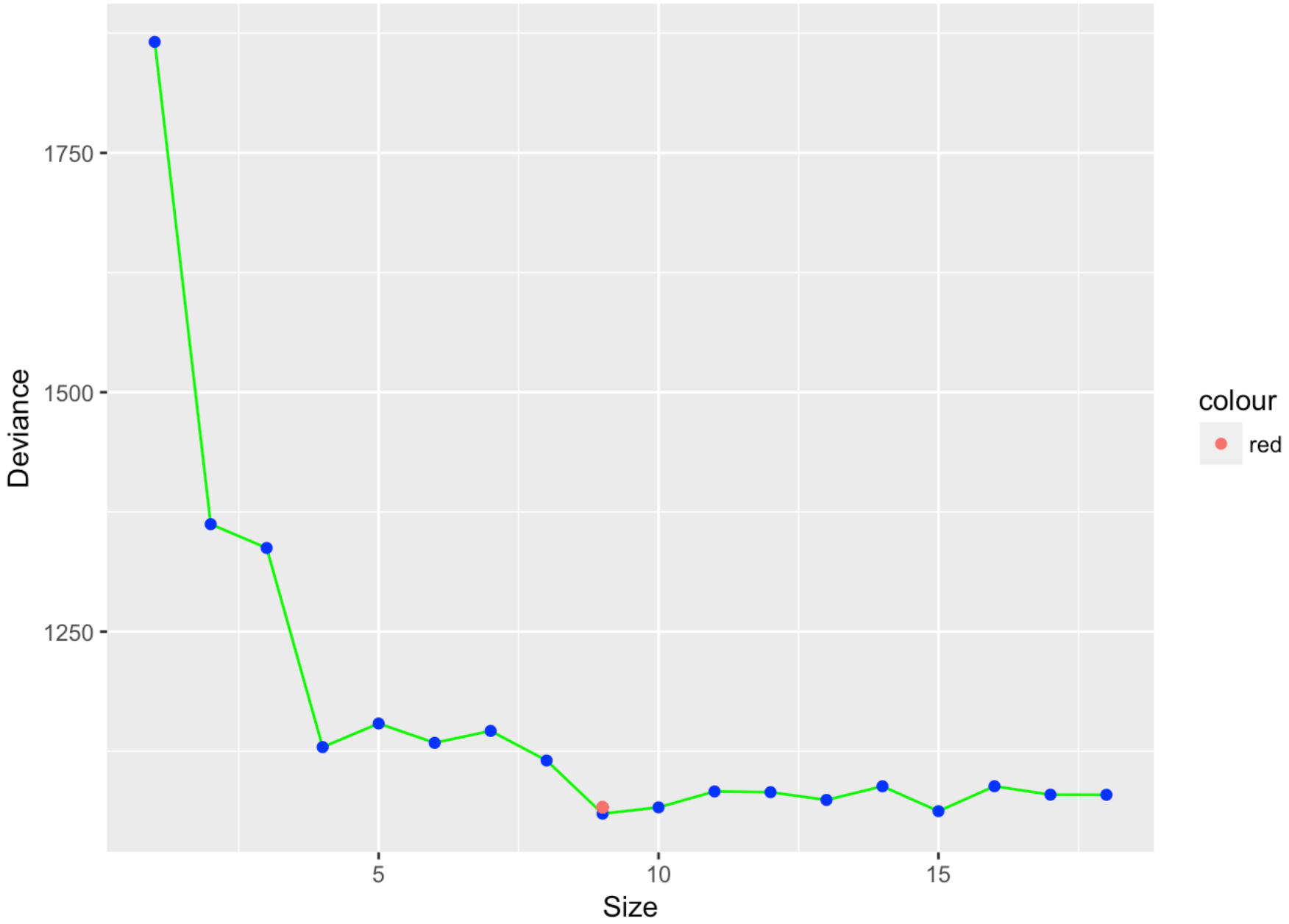
(c)

Use cross-validation in order to determine the optimal level of tree complexity. Does pruning the tree improve the test MSE?

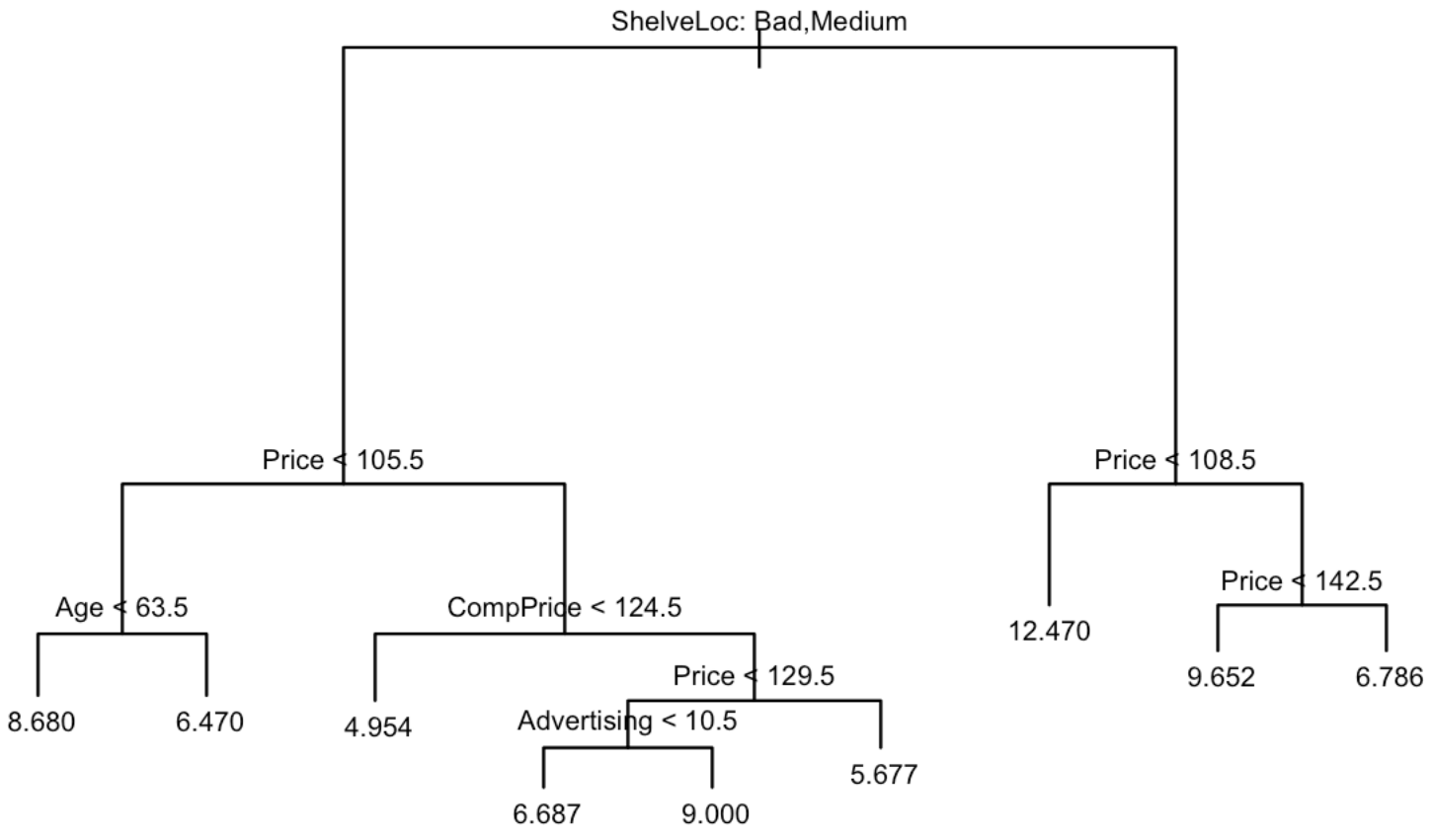
```
## [1] "Size with the lowest deviance:  9"
```


490
140
70
52
38
36
29
22
19
1800
1600
1400
1200
deviance
5
10
15
size

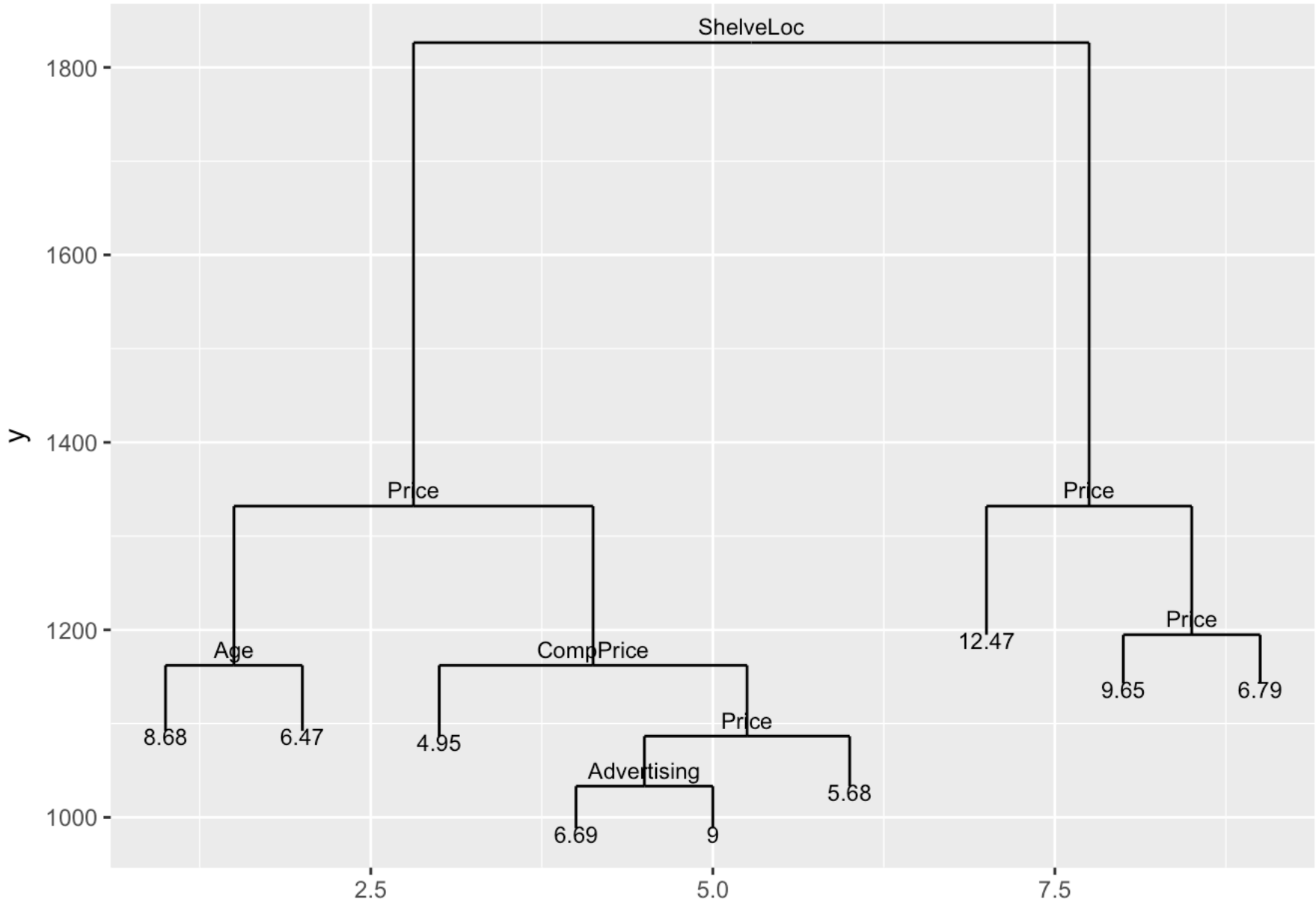
ggplot



** Prune and create new tree**



ggplot purne tree




```
## [1] "Test MSE of Purned tree model  =  4.78720523400631"
```

Crossvalidation method selected the size of 9 and reduced the tree size but it did not improve the test error.

(d)

Use the bagging approach in order to analyze this data. What test MSE do you obtain? Use the importance() function to determine which variables are most important.

ANS Bagging is simply a special case of a random forest with m = p. We used 10 predictor (mtry=10)

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## [1] "Test MSE of Baging tree model  =  3.10595693149298"
```

Baging method decreased the test error to 3.11

	%IncMSE	IncNodePurity
CompPrice	25.379533	180.478377
Income	7.613740	87.848164
Advertising	17.920275	113.525780
Population	4.043980	71.369174
Price	64.384735	503.755337
ShelveLoc	72.012259	574.139072
Age	21.996106	192.018479
Education	1.720849	41.786597
Urban	-2.512028	8.923027
US	2.253205	9.163668

Discussion: Importance table shows the effect on MSE and purity of node by excluding the perticular variables from tree model.Here, Model's MSE increased by 72% and 64.3% if ShelveLoc" and "Price"variables are exculded from model respectievly. The variable with largest mean decrease are ShelveLoc" and "Price", hence these two variables are most important.

(e)

Use random forests to analyze this data. What test MSE do you obtain? Use the importance() function to determine which variables are most important. Describe the effect of m, the number of variables considered at each split, on the error rate obtained

ANS

```
## [1] "MSE test error of random forest= 3.13329069467398"
```

```
##                %IncMSE IncNodePurity
## CompPrice   19.1715675     163.00788
## Income       6.0009910     111.98248
## Advertising 14.6576895     131.08648
## Population   5.0074016      96.03092
## Price       51.9807606     454.68571
## ShelveLoc   55.4511507     514.86478
## Age         20.6169715     216.52138
## Education   -0.3992538      56.45394
## Urban       -1.4439706      11.58196
## US           3.3166603      13.80368
```

Discussion: Changing value of m , test errror increased to 3.13. By default, randomForest() uses p/3 variables when building a random forest of regression trees. Here we used mtry =5, which means 5 predictors should be considered for each split of the tree. By looking at mean decrease rate at importance table ,Still ShelveLoc and Price are most important variables for predicting sales.

3. Question 8.4.9 pg 334

(9.) This problem involves the OJ data set which is part of the ISLR package. ##(a)

Create a training set containing a random sample of 800 observations,and a test set containing the remaining observations.

```
## 'data.frame':    1070 obs. of  18 variables:
##  $ Purchase      : Factor w/ 2 levels "CH","MM": 1 1 1 2 1 1 1 1 1 1 ...
##  $ WeekofPurchase: num  237 239 245 227 228 230 232 234 235 238 ...
##  $ StoreID       : num  1 1 1 1 7 7 7 7 7 7 ...
##  $ PriceCH       : num  1.75 1.75 1.86 1.69 1.69 1.69 1.69 1.75 1.75 1.75 ...
##  $ PriceMM       : num  1.99 1.99 2.09 1.69 1.69 1.99 1.99 1.99 1.99 1.99 ...
##  $ DiscCH        : num  0 0 0.17 0 0 0 0 0 0 0 ...
##  $ DiscMM        : num  0 0.3 0 0 0 0 0.4 0.4 0.4 0.4 ...
##  $ SpecialCH     : num  0 0 0 0 0 0 1 1 0 0 ...
##  $ SpecialMM     : num  0 1 0 0 0 1 1 0 0 0 ...
##  $ LoyalCH       : num  0.5 0.6 0.68 0.4 0.957 ...
##  $ SalePriceMM   : num  1.99 1.69 2.09 1.69 1.69 1.99 1.59 1.59 1.59 1.59 ...
##  $ SalePriceCH   : num  1.75 1.75 1.69 1.69 1.69 1.69 1.69 1.75 1.75 1.75 ...
##  $ PriceDiff     : num  0.24 -0.06 0.4 0 0 0.3 -0.1 -0.16 -0.16 -0.16 ...
##  $ Store7        : Factor w/ 2 levels "No","Yes": 1 1 1 1 2 2 2 2 2 2 ...
##  $ PctDiscMM     : num  0 0.151 0 0 0 ...
##  $ PctDiscCH     : num  0 0 0.0914 0 0 ...
##  $ ListPriceDiff : num  0.24 0.24 0.23 0 0 0.3 0.3 0.24 0.24 0.24 ...
##  $ STORE         : num  1 1 1 1 0 0 0 0 0 0 ...
```

(b)

Fit a tree to the training data, with Purchase as the response and the other variables as predictors. Use the summary() function to produce summary statistics about the tree, and describe the results obtained. What is the training error rate? How many terminal nodes does the tree have?

```
##
## Classification tree:
## tree(formula = Purchase ~ ., data = OJ.train)
## Variables actually used in tree construction:
## [1] "LoyalCH"   "PriceDiff"
## Number of terminal nodes:  8
## Residual mean deviance:  0.771 = 610.6 / 792
## Misclassification error rate: 0.1662 = 133 / 800
```

Discusssion: The tree used two variables LoyalCH“an”PriceDiff to grow the tree and it has 8 terminal nodes. The traning error rate is 16%. For classification tree deviance reported at summary is 0.77.

(c)

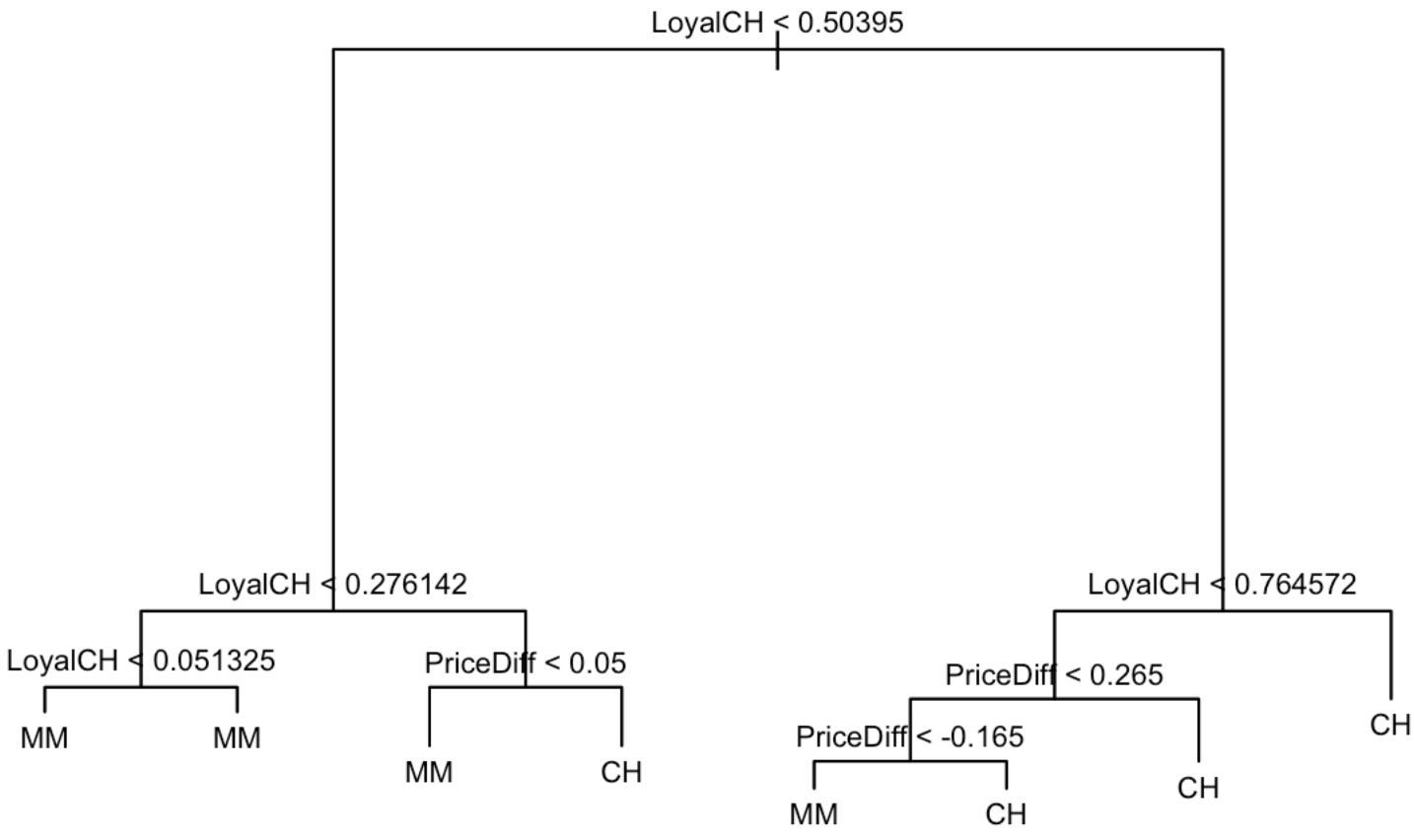
Type in the name of the tree object in order to get a detailed text output. Pick one of the terminal nodes, and interpret the information displayed.

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 800 1073.00 CH ( 0.60500 0.39500 )
##    2) LoyalCH < 0.50395 348  411.90 MM ( 0.27874 0.72126 )
##      4) LoyalCH < 0.276142 163  121.40 MM ( 0.12270 0.87730 )
##        8) LoyalCH < 0.051325 60   10.17 MM ( 0.01667 0.98333 ) *
##        9) LoyalCH > 0.051325 103   98.49 MM ( 0.18447 0.81553 ) *
##      5) LoyalCH > 0.276142 185  251.20 MM ( 0.41622 0.58378 )
##       10) PriceDiff < 0.05 76   72.61 MM ( 0.18421 0.81579 ) *
##       11) PriceDiff > 0.05 109  148.40 CH ( 0.57798 0.42202 ) *
##    3) LoyalCH > 0.50395 452  372.30 CH ( 0.85619 0.14381 )
##      6) LoyalCH < 0.764572 191  223.70 CH ( 0.72775 0.27225 )
##       12) PriceDiff < 0.265 114  154.50 CH ( 0.58772 0.41228 )
##         24) PriceDiff < -0.165 34   42.81 MM ( 0.32353 0.67647 ) *
##         25) PriceDiff > -0.165 80   97.74 CH ( 0.70000 0.30000 ) *
##       13) PriceDiff > 0.265 77   37.01 CH ( 0.93506 0.06494 ) *
##      7) LoyalCH > 0.764572 261  103.30 CH ( 0.95019 0.04981 ) *
```

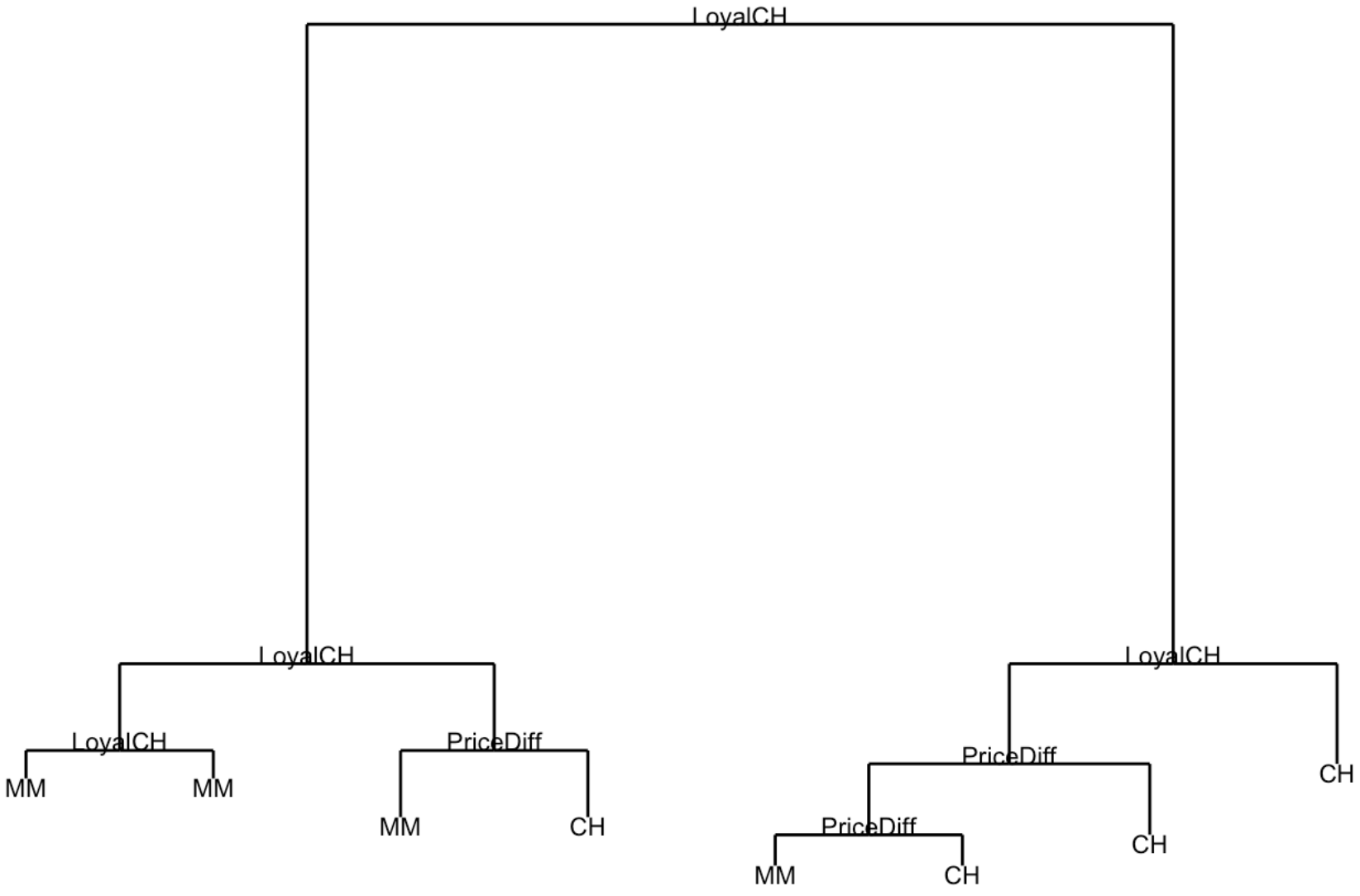
Discussion: In the tree structure , we picked the node 24 with * which is terminal node.The spliting variable at this node is PriceDiff.The split decision criterion on this node is PriceDiff < -0.165.There are 34 observation at this branch of tree with deviance of 42.81.The prediction on this node is purchase=MM. About 32% of observation in this branch take value of CH and remaing 78% of observation take value of MM.

(d)

Create a plot of the tree, and interpret the results.



ggplot of tree of Orange Juce data



Discussion As we can see in tree, the Customer brand loyalty for CH (LOyalCH) is most important variable. Predictor space is devided by whether chustomer brand loyalty is less than 0.5 or not.Second level of tree is also further devided by checking LOyalCH. Here customer LOyalCH<0.27 are predected as MM with comparing with price difference. and LOyalCH>0.76 are predictes as CH with comparing with or with out price.

(e)

Predict the response on the test data, and produce a confusion matrix comparing the test labels to the predicted test labels. What is the test error rate?

```
##     oj.pred
##       CH  MM
##   CH 153  16
##   MM  26  75
```

Test error

```
1 - (153+ 75) / 270
```

```
## [1] 0.1555556
```

Test error rate is about 15%

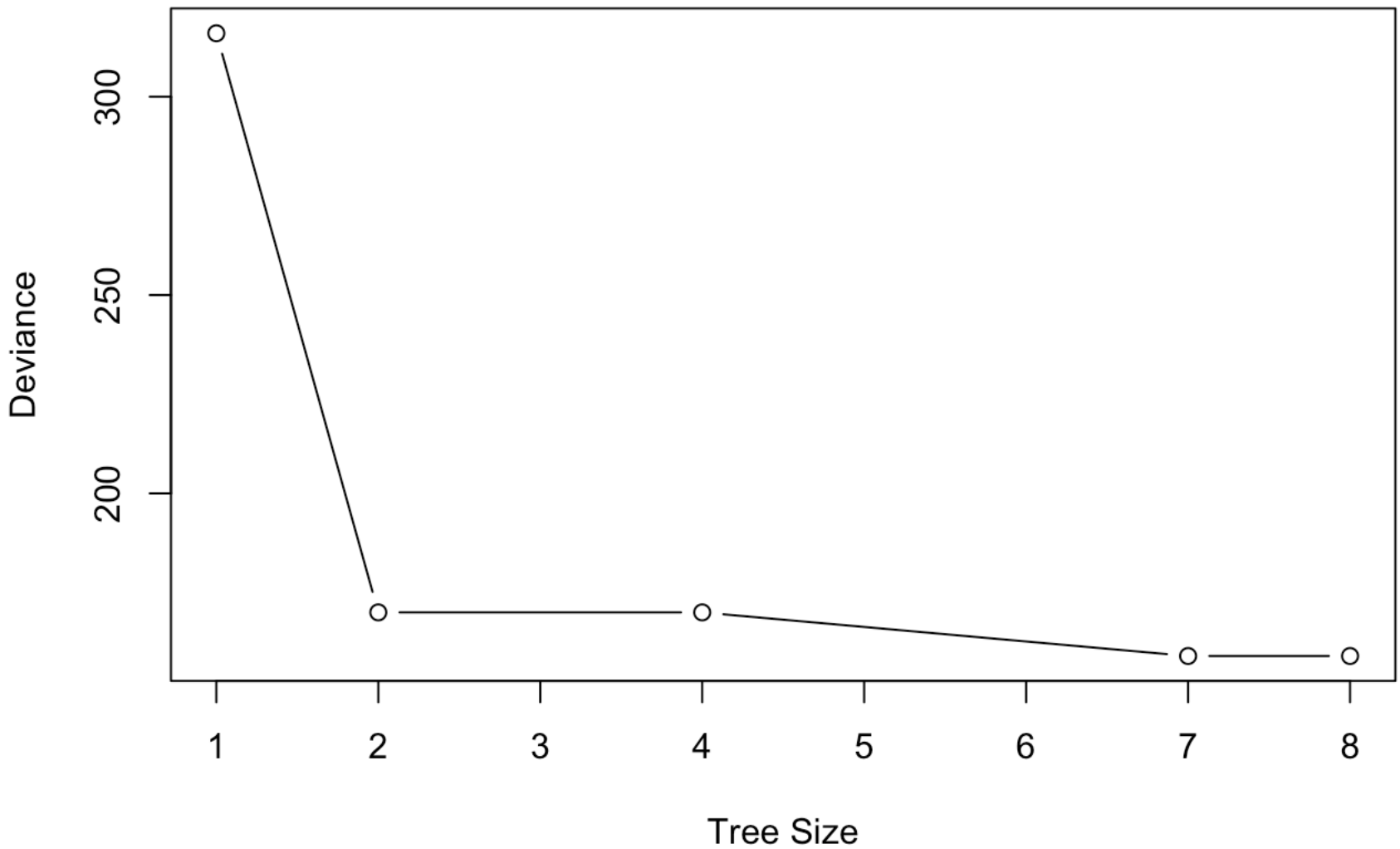
(f)

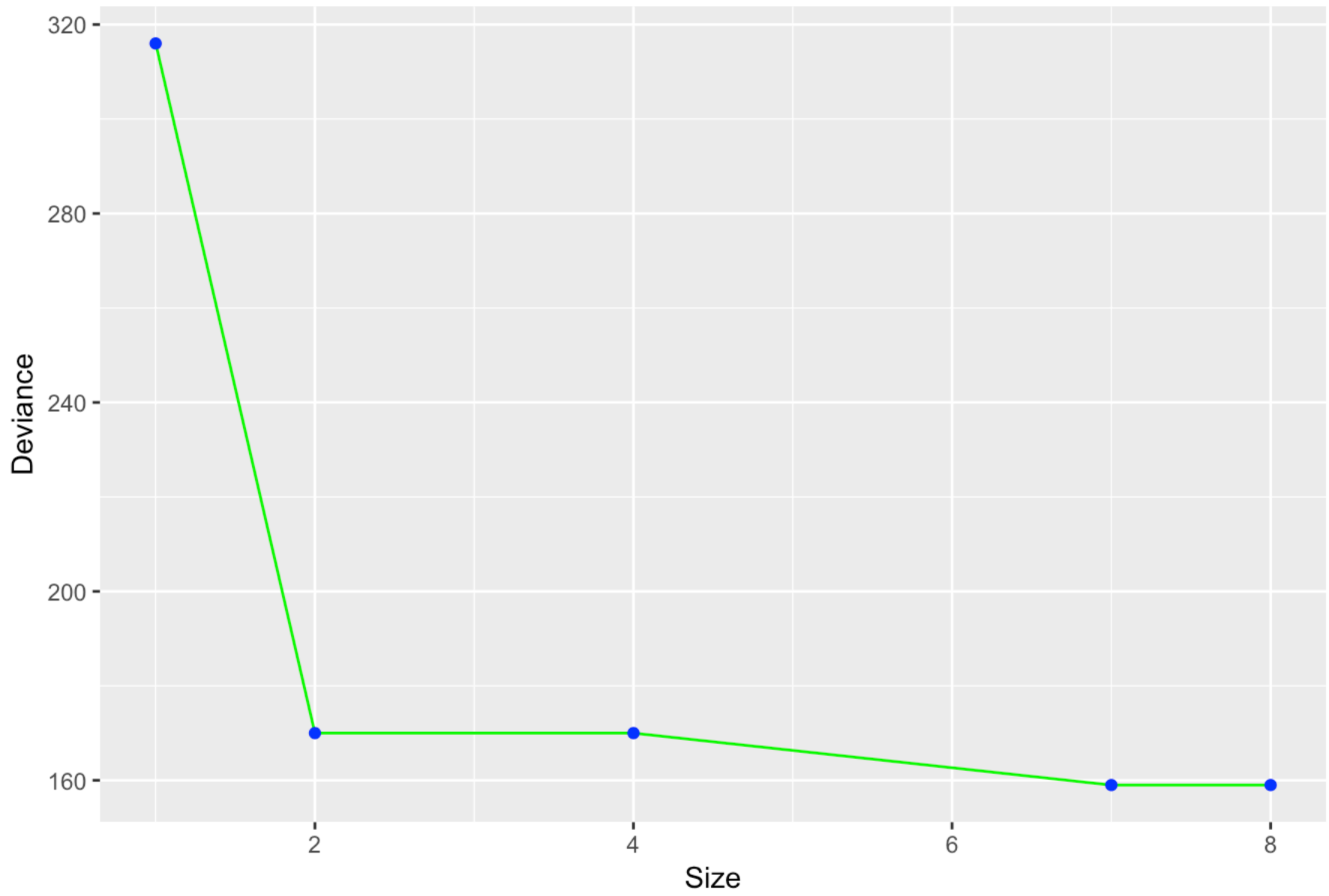
Apply the cv.tree() function to the training set in order to determine the optimal tree size.

```
## $size
## [1] 8 7 4 2 1
##
## $dev
## [1] 159 159 170 170 316
##
## $k
## [1]  -Inf   0.0   4.0   8.5 154.0
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

(g)

Produce a plot with tree size on the x-axis and cross-validated classification error rate on the y-axis.


Deviance
300
250
200
1
2
3
4
5
6
7
8
Tree Size



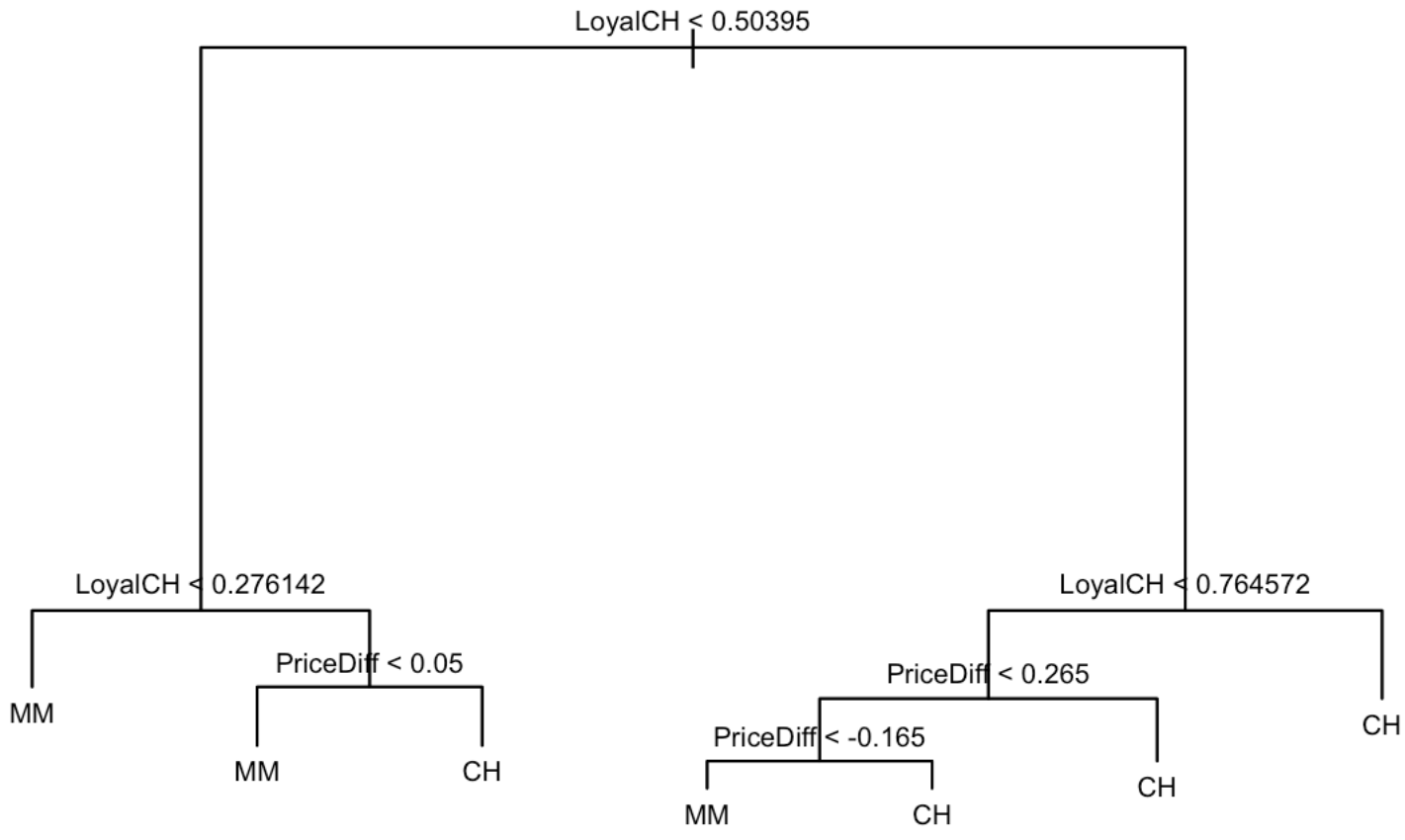
##(h) Which tree size corresponds to the lowest cross-validated classification error rate?

```
## [1] "Size with the lowest deviance:  8"
```

The tree with 8 and 7 terminal nodes results in the lowest cross-validation error rate, with 159 cross-validation errors.

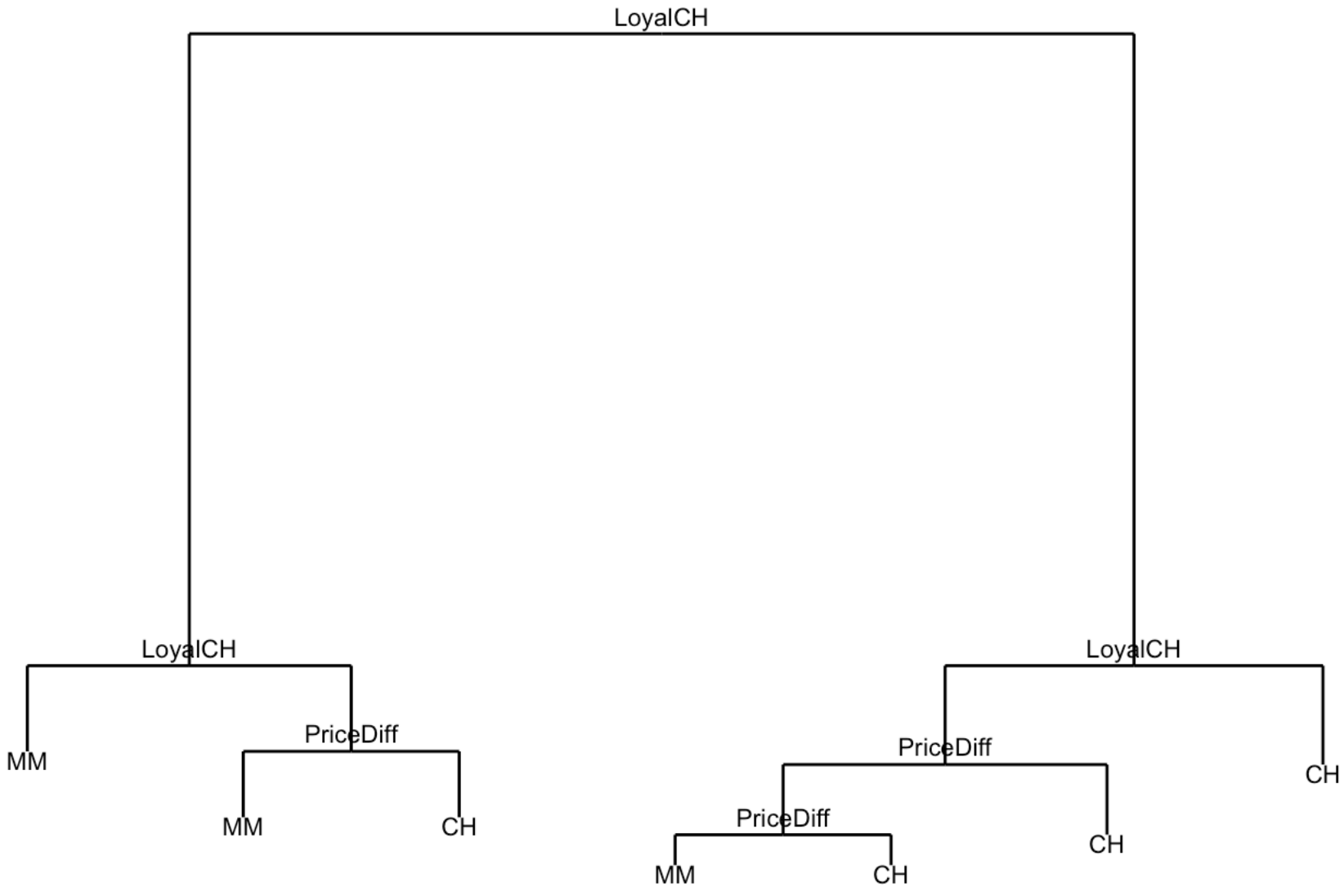
(i)

Produce a pruned tree corresponding to the optimal tree size obtained using cross-validation. If cross-validation does not lead to selection of a pruned tree, then create a pruned tree with five terminal nodes.



ggplot of purned tree

ggplot- purned tree



Discussion: cross validation did not select a purned tree since the minimum best size occure at best =8 which is same as original tree. Then I selected best=5 but purne.misclass() did not produce tree with 5 terminals nodes.According to R Docunetation, if there is no tree in sequence of the requested size , the next largest is returned by purne.misclass().So I got the purne tree with 7 terminal nodes.

(j)

Compare the training error rates between the pruned and unpruned trees. Which is higher?

```
##
## Classification tree:
## snip.tree(tree = oj.tree, nodes = 4L)
## Variables actually used in tree construction:
## [1] "LoyalCH"   "PriceDiff"
## Number of terminal nodes:  7
## Residual mean deviance:  0.786 = 623.3 / 793
## Misclassification error rate: 0.1662 = 133 / 800
```

```
##
## Classification tree:
## tree(formula = Purchase ~ ., data = OJ.train)
## Variables actually used in tree construction:
## [1] "LoyalCH"   "PriceDiff"
## Number of terminal nodes:  8
## Residual mean deviance:  0.771 = 610.6 / 792
## Misclassification error rate: 0.1662 = 133 / 800
```

Discussion: Unpurned and purned tree produce same classification error rate for training data with 8 and 7 terminal nodes respectievly.

(k)

Compare the test error rates between the pruned and unpruned trees. Which is higher?

```
## [1] 0.1555556
```

```
## [1] 0.1555556
```

Discussion: Both test error rate are same.

4. Question 8.4.10 pg 334

(10.) We now use boosting to predict Salary in the Hitters data set. ##(a) Remove the observations for whom the salary information is unknown, and then log-transform the salaries.

```
## 'data.frame':    322 obs. of  20 variables:
##  $ AtBat    : int  293 315 479 496 321 594 185 298 323 401 ...
##  $ Hits     : int  66 81 130 141 87 169 37 73 81 92 ...
##  $ HmRun    : int  1 7 18 20 10 4 1 0 6 17 ...
##  $ Runs     : int  30 24 66 65 39 74 23 24 26 49 ...
##  $ RBI      : int  29 38 72 78 42 51 8 24 32 66 ...
##  $ Walks    : int  14 39 76 37 30 35 21 7 8 65 ...
##  $ Years    : int  1 14 3 11 2 11 2 3 2 13 ...
##  $ CAtBat   : int  293 3449 1624 5628 396 4408 214 509 341 5206 ...
##  $ CHits    : int  66 835 457 1575 101 1133 42 108 86 1332 ...
##  $ CHmRun   : int  1 69 63 225 12 19 1 0 6 253 ...
##  $ CRuns    : int  30 321 224 828 48 501 30 41 32 784 ...
##  $ CRBI     : int  29 414 266 838 46 336 9 37 34 890 ...
##  $ CWalks   : int  14 375 263 354 33 194 24 12 8 866 ...
##  $ League   : Factor w/ 2 levels "A","N": 1 2 1 2 2 1 2 1 2 1 ...
##  $ Division : Factor w/ 2 levels "E","W": 1 2 2 1 1 2 1 2 2 1 ...
##  $ PutOuts  : int  446 632 880 200 805 282 76 121 143 0 ...
##  $ Assists  : int  33 43 82 11 40 421 127 283 290 0 ...
##  $ Errors   : int  20 10 14 3 4 25 7 9 19 0 ...
##  $ Salary   : num  NA 475 480 500 91.5 750 70 100 75 1100 ...
##  $ NewLeague: Factor w/ 2 levels "A","N": 1 2 1 2 2 1 1 1 2 1 ...
```

```
## [1] 59
```

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

```
## 'data.frame':    263 obs. of  20 variables:
##  $ AtBat    : int  315 479 496 321 594 185 298 323 401 574 ...
##  $ Hits     : int  81 130 141 87 169 37 73 81 92 159 ...
##  $ HmRun    : int  7 18 20 10 4 1 0 6 17 21 ...
##  $ Runs     : int  24 66 65 39 74 23 24 26 49 107 ...
##  $ RBI      : int  38 72 78 42 51 8 24 32 66 75 ...
##  $ Walks    : int  39 76 37 30 35 21 7 8 65 59 ...
##  $ Years    : int  14 3 11 2 11 2 3 2 13 10 ...
##  $ CAtBat   : int  3449 1624 5628 396 4408 214 509 341 5206 4631 ...
##  $ CHits    : int  835 457 1575 101 1133 42 108 86 1332 1300 ...
##  $ CHmRun   : int  69 63 225 12 19 1 0 6 253 90 ...
##  $ CRuns    : int  321 224 828 48 501 30 41 32 784 702 ...
##  $ CRBI     : int  414 266 838 46 336 9 37 34 890 504 ...
##  $ CWalks   : int  375 263 354 33 194 24 12 8 866 488 ...
##  $ League   : Factor w/ 2 levels "A","N": 2 1 2 2 1 2 1 2 1 1 ...
##  $ Division : Factor w/ 2 levels "E","W": 2 2 1 1 2 1 2 2 1 1 ...
##  $ PutOuts  : int  632 880 200 805 282 76 121 143 0 238 ...
##  $ Assists  : int  43 82 11 40 421 127 283 290 0 445 ...
##  $ Errors   : int  10 14 3 4 25 7 9 19 0 22 ...
##  $ Salary   : num  6.16 6.17 6.21 4.52 6.62 ...
##  $ NewLeague: Factor w/ 2 levels "A","N": 2 1 2 2 1 1 1 2 1 1 ...
```

Discussion: There were 59 observation with missing values of salary , which were removed. Log transformation of salary variable was done.

(b)

Create a training set consisting of the first 200 observations, and a test set consisting of the remaining observations.

(c)

Perform boosting on the training set with 1,000 trees for a range of values of the shrinkage parameter $lamda$. Produce a plot with different shrinkage values on the x-axis and the corresponding training set MSE on the y-axis.

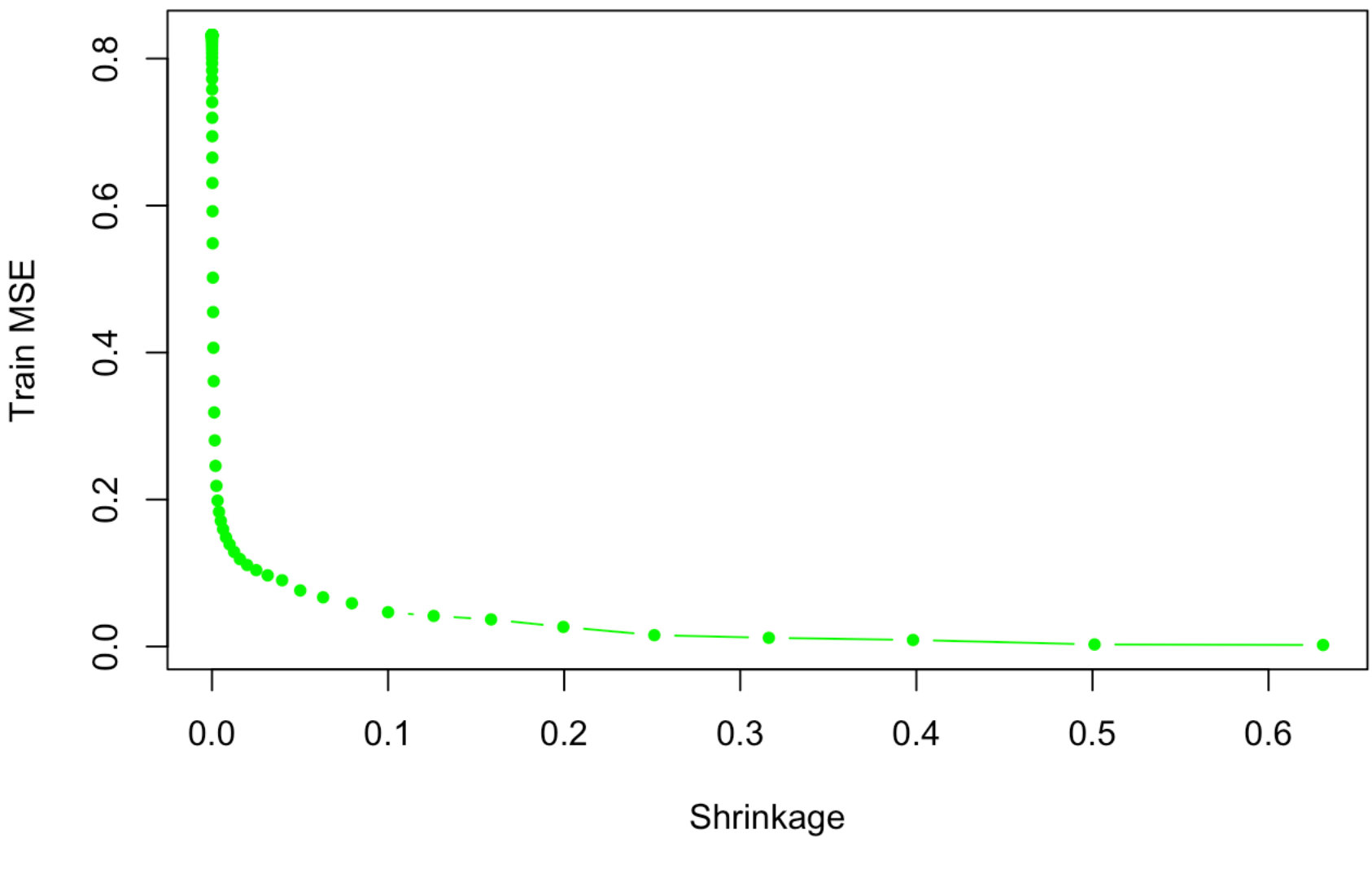
```
## Loading required package: survival
```

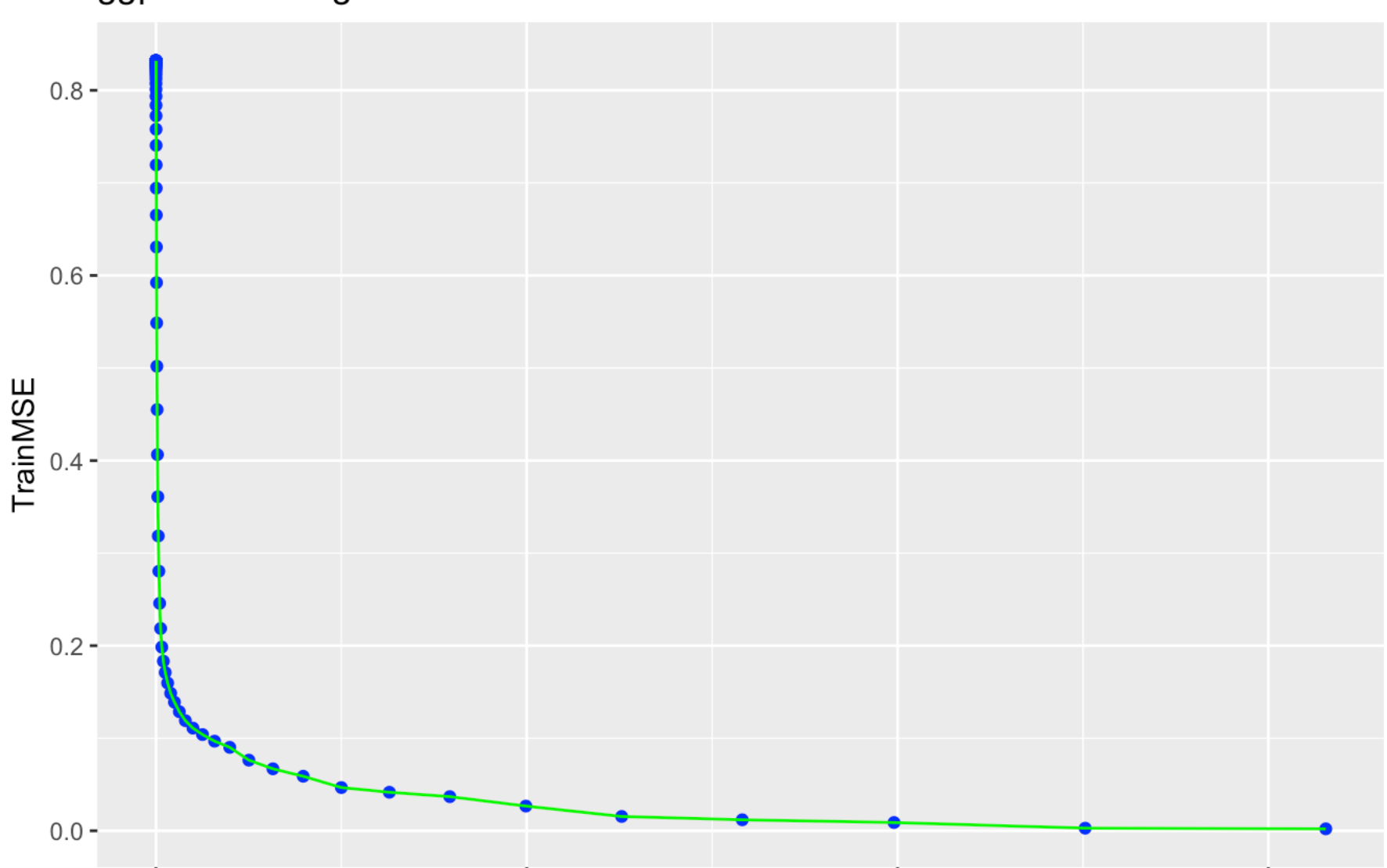
```
## Loading required package: lattice
```

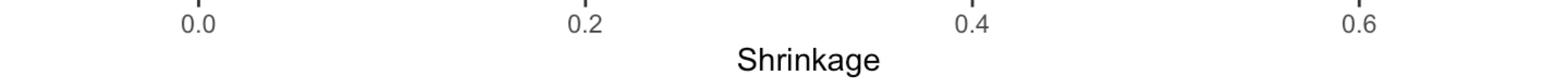
```
## Loading required package: splines
```

```
## Loading required package: parallel
```

```
## Loaded gbm 2.1.3
```

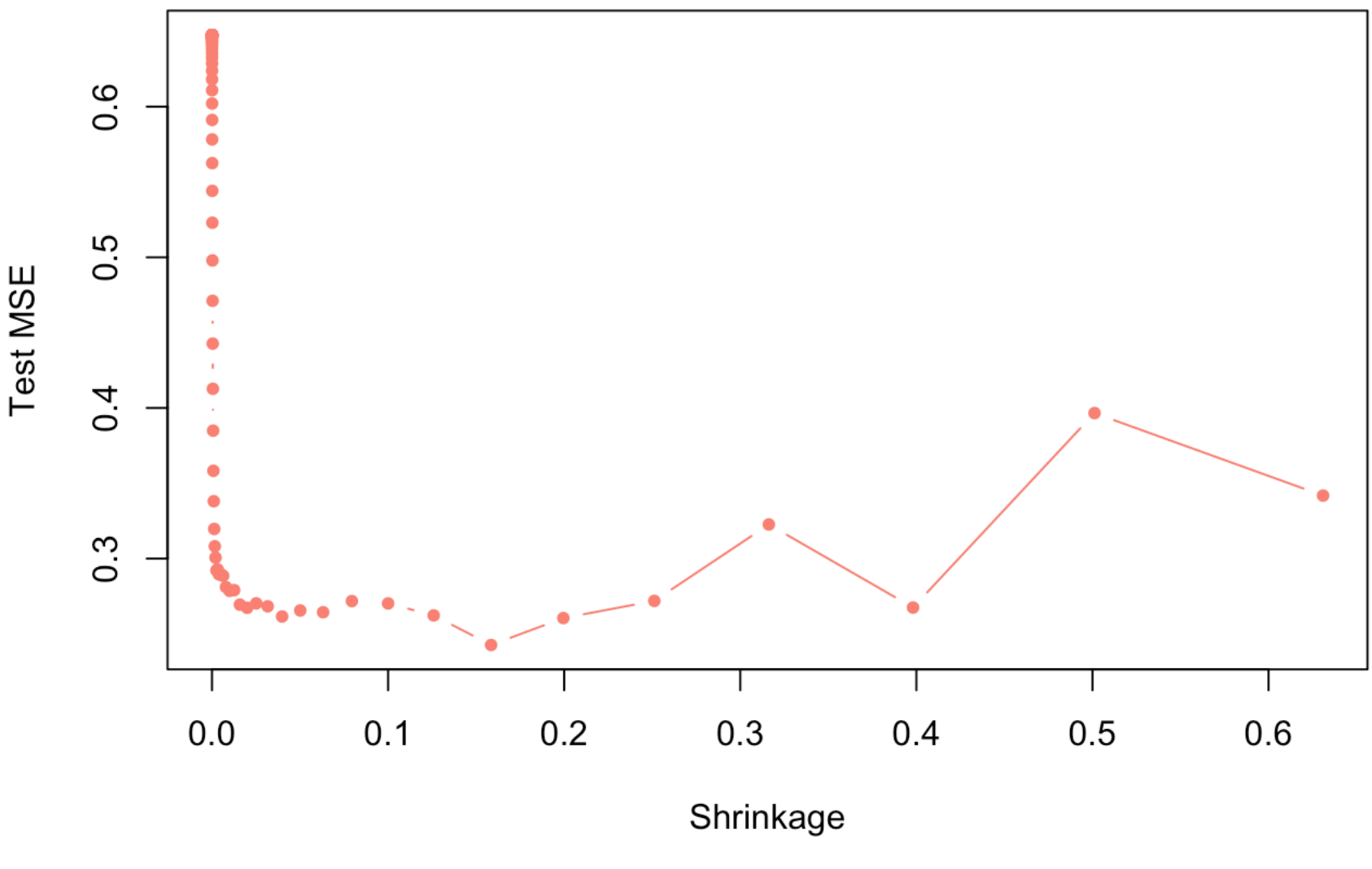

Train MSE vs Shinkage for bosting
Train MSE
0.0
0.2
0.4
0.6
0.8
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Shrinkage

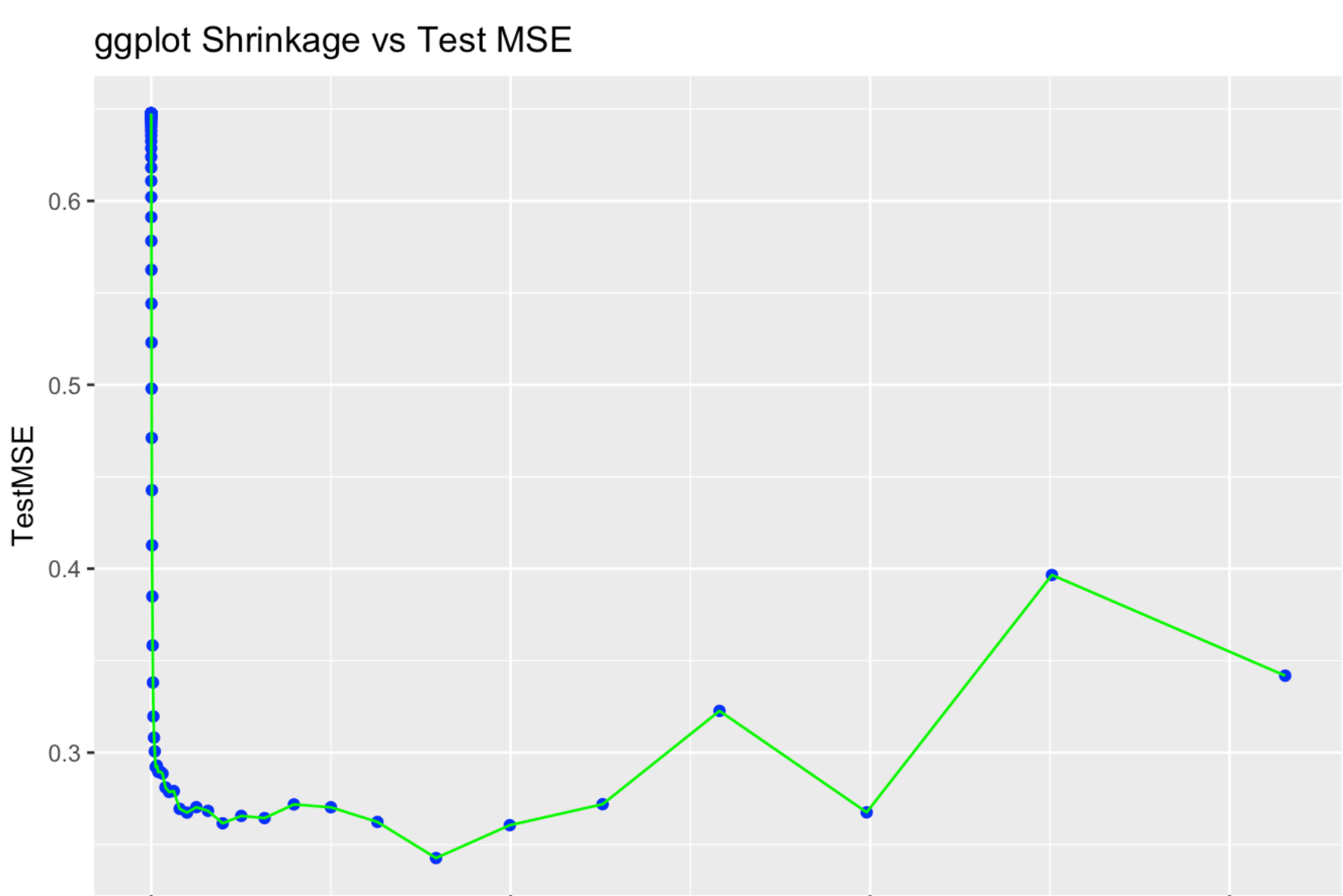

ggplot Shrinkage vs Train MSE
TrainMSE
0.0
0.2
0.4
0.6
0.8

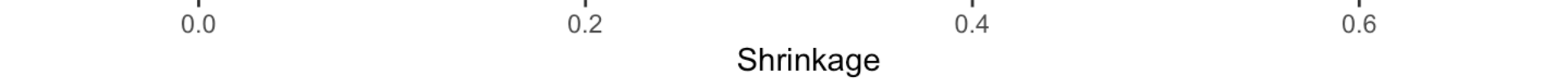


(d)

Produce a plot with different shrinkage values on the x-axis and the corresponding test set MSE on the y-axis.


Test MSE vs Shinkage for bosting
Test MSE
0.3
0.4
0.5
0.6
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Shrinkage


ggplot Shrinkage vs Test MSE
TestMSE
0.3
0.4
0.5
0.6




```
## [1] 0.2426616
```

```
## [1] 0.1584893
```

Discussion: Minimum test error is 0.2426616 at lamda=0.1584893

(e)

Compare the test MSE of boosting to the test MSE that results from applying two of the regression approaches seen in Chapters 3 and 6. linear model

```
## [1] 0.4917959
```

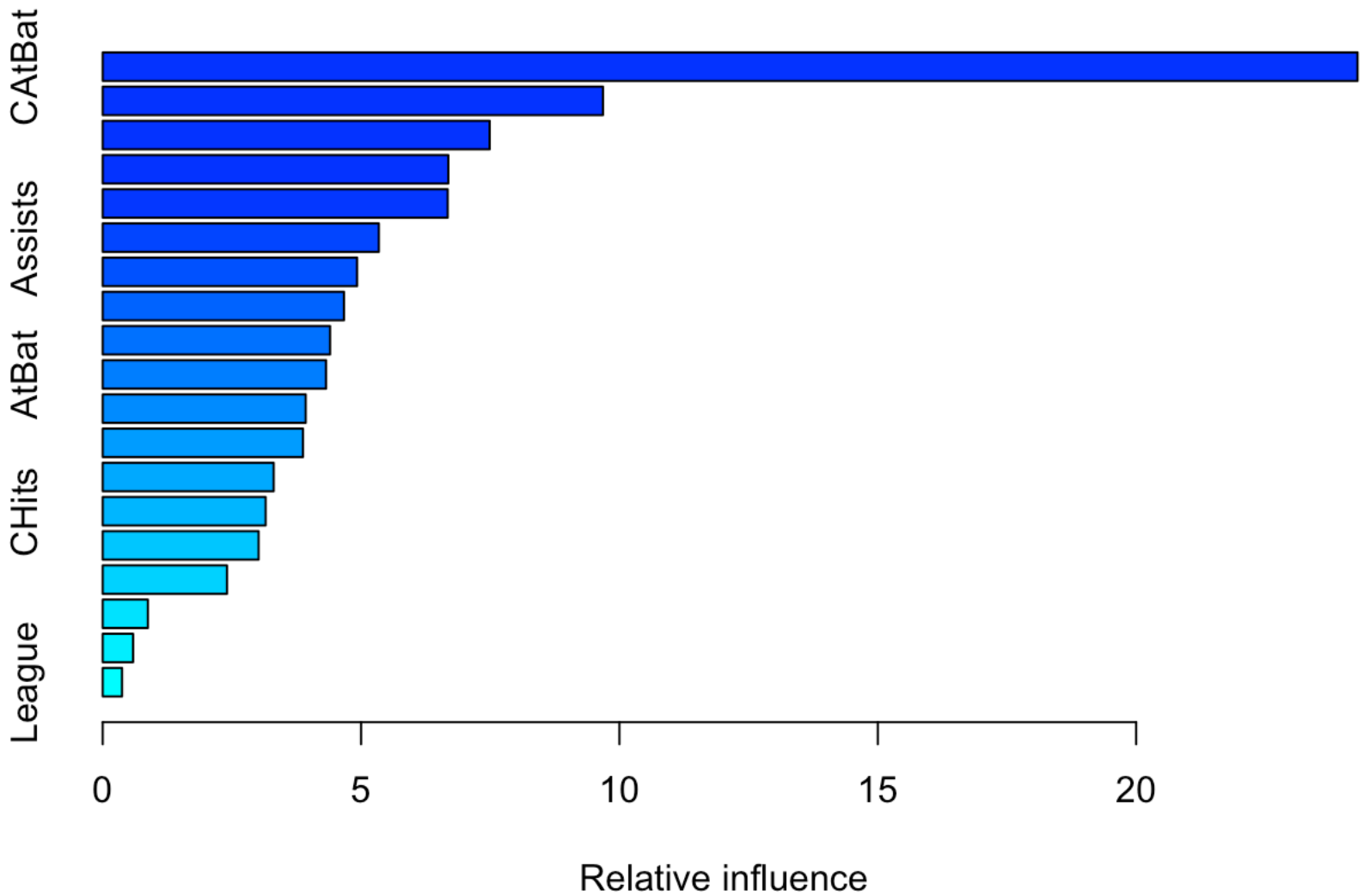
lasso

```
## [1] 0.4700537
```

Disscussion: MSE for boost=0.2426616 which is less than linear linear regression and regularization method lasso.

(f)

Which variables appear to be the most important predictors in the boosted model?



	var <fctr>	rel.inf <dbl>
CAtBat	CAtBat	24.2844773
PutOuts	PutOuts	9.6811188
Walks	Walks	7.4878834
CRuns	CRuns	6.6878121
RBI	RBI	6.6766513
Assists	Assists	5.3445841
CWalks	CWalks	4.9231378
Years	Years	4.6689906
CHmRun	CHmRun	4.3997763
AtBat	AtBat	4.3225089

Disscussion:

CAtBat, PutOut and Walks are the most important variables to predict the salary of hitter data.

(g)

Now apply bagging to the training set. What is the test set MSE for this approach?

```
## [1] 0.231884
```

Disscussion: The test set MSE for bagging is 0.231884 which is slightly less than bosting method.

5.

In the past couple of homework assignments you have used different classification methods to analyze the dataset you chose. For this homework, use tree-based classification methods (tree,bagging, randomforest, boosting) to model your data. Find the test error using any/all of methods (VSA, K-fold CV). Compare the results you obtained with the result from previous homework. Did the results improve? (Use the table you previously made to compare results)

Ans: I used tree() to model tree and bagging() for bagging and randomforest() for random forest and boositing () for boosting the data. Among them random forest resulted slighltly less test error.Tree model produced same test error for VSA and CV. Baging has less error in VSA than cv. Cross validation approach slightly improved the error rate in boosting. By comparing the result from pervious homework logistic regression with 5fold cross validation and SVM test error were less than current test error resulted by tree ,paging ,random forest and boosting. Among all the models SVM has very small test error .Hence Tree based model Did not improve the test error result.

Error Table

Method	VSA	LOOCV	FOLD.5CV
Logistic Reg	0.1793391	0.1873464	0.1219478
knn	0.2428449	0.3593366	0.2909104
LDA	0.1770053	0.2168305	0.2194654
QDA	0.2154526	0.213145	0.2096682
MclustDA	0.2117676	NA	0.2307535
MclustDA (EDDA)	0.2067314	0.2067314	0.2027846
SVM	0.0002457	0.001228501	0.0002048
Tree	0.1788887	NA	0.1788887
Baging	0.1788887	NA	0.1854472
RandomForest	0.1745127	NA	0.1705495

Boosting	0.1788887	NA	0.1717061

Q5 details

Tree model for adult data

I have used cleand data .I did not include process of cleaning and variables selection which was done at previous homework.

	X<int>	...<int>	workclass<fctr>	education<fctr>	educationnum<int>	maritalstatus<fctr>	occupation<fctr>	relationship<fctr>
1	1	39	gov	graduate	13	notmarried	clerical	outofamily
2	2	50	selfemp	graduate	13	married	highskillabr	husband
3	3	38	private	secndrysch	9	divorce	lowskillabr	outofamily
4	4	53	private	secndrysch	7	married	lowskillabr	husband
5	5	28	private	graduate	13	married	lowskillabr	wife
6	6	37	private	master	14	married	highskillabr	wife

6 rows | 1-10 of 14 columns

split 75%

Vaildation set approach

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
## Loading required package: strucchange
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

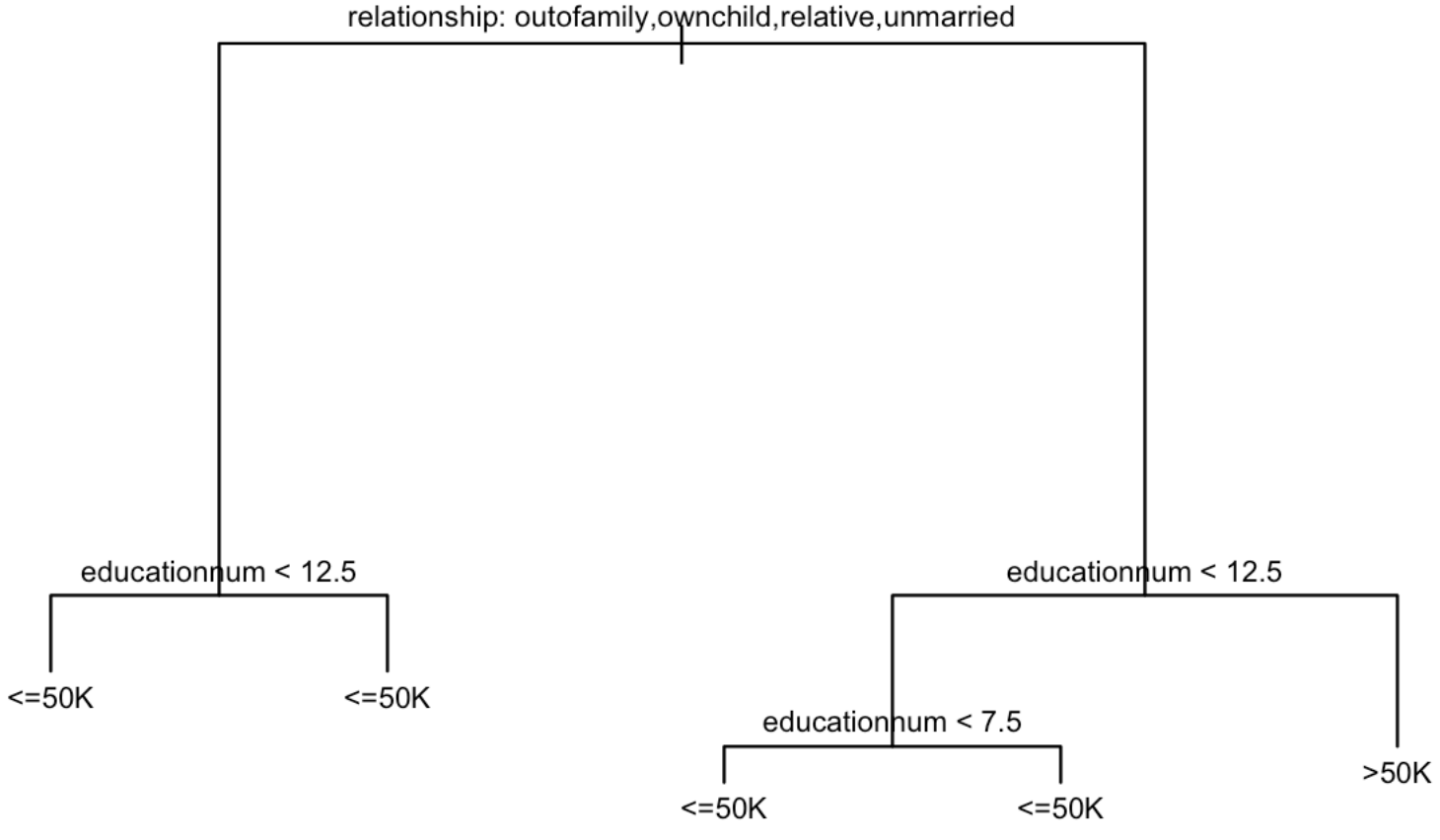
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
## Loading required package: libcoin
```

```
##
## Attaching package: 'partykit'
```

```
## The following objects are masked from 'package:party':
##
##     cforest, ctree, ctree_control, edge_simple, mob, mob_control,
##     node_barplot, node_bivplot, node_boxplot, node_inner,
##     node_surv, node_terminal, varimp
```



error

```
##         ad.pred
##          <=50K >50K
##    <=50K  5397  276
##    >50K   1073  795
```

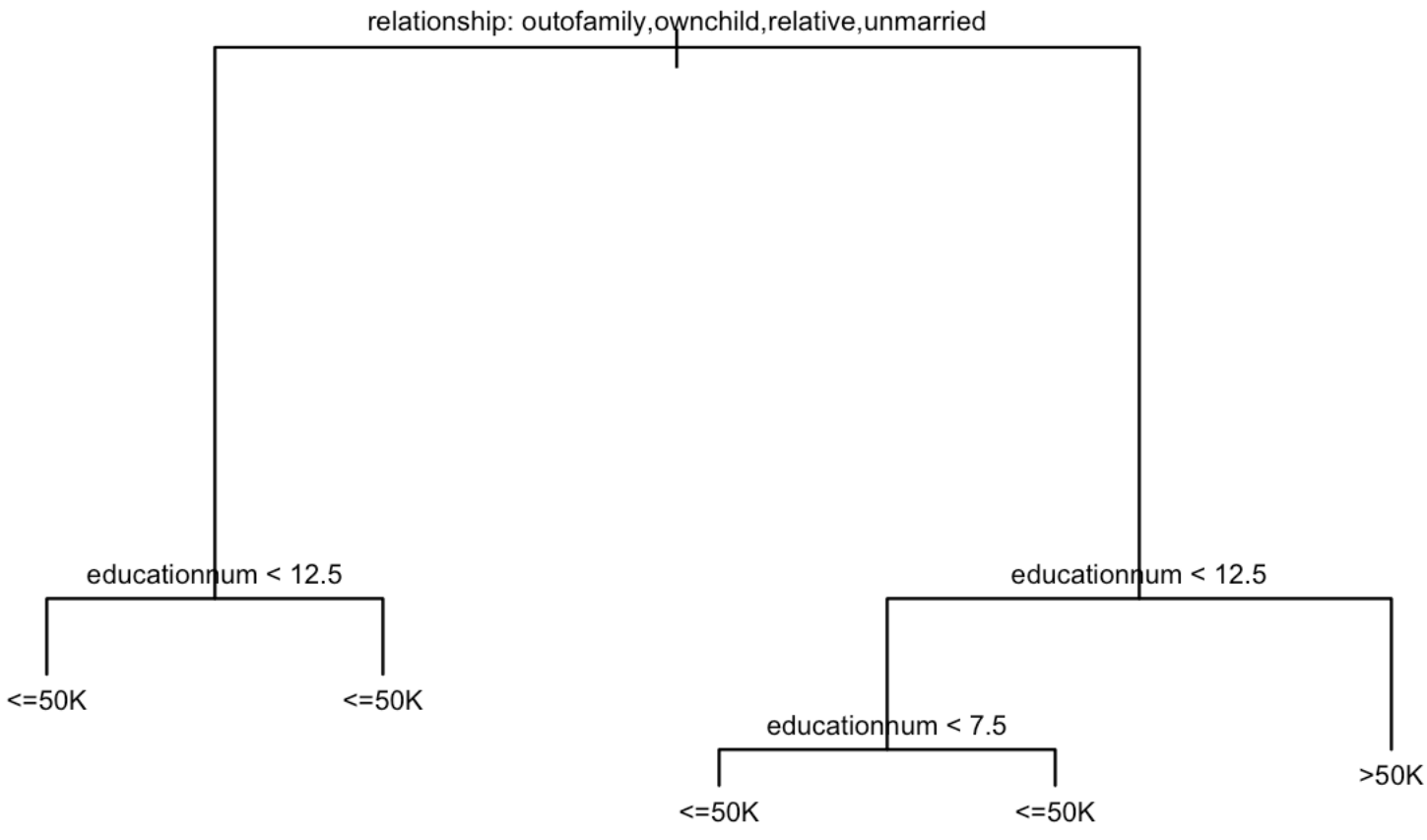
Misclassification Error

```
1 - (5397+ 795) / 7541
```

```
## [1] 0.1788887
```

Test error rate is about 17.8%

cross validation set approach



Disscussion: which is exactly similar to original tree with out cross validation , cross validation indicates the size=5 where error is minimum.

Bagging method

Vaildation set approach

I used Bagging() for bagging tree model for ADULT data

```
## Loading required package: caret
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:survival':
##
##     cluster
```

```
## Loading required package: doParallel
```

```
## Loading required package: iterators
```

```
## [1] "Test Error of  Baging tree model  =  0.178888741546214"
```

cross validation for baging, I used bagging.cv

```
##                 Observed Class
## Predicted Class <=50K  >50K
##           <=50K 16088  3279
##           >50K    893  2361
```

```
## [1] "Test Error of  Baging tree model using cross validation  =  0.184430396534194
"
```

Disscussion: Cross validation approach did not imporve error rate for bagging model.

Random forest

VSA

```
## [1] 0.1755735
```

```
## [1] "Test Error of  Random forest tree model  =  0.175573531361888"
```

Improtance of variables in model

```
##                    <=50K        >50K MeanDecreaseAccuracy MeanDecreaseGini
## X             -0.5047058   5.8227391             3.263587       1448.09755
## age            3.0145208 116.8150534            89.418927       1258.02493
## workclass     25.5596764   0.4478244            21.641520        228.50887
## education     25.4850468  18.7531912            28.717358        317.92108
## educationnum  48.4622398  90.4852951            81.666875        902.26990
## maritalstatus  9.7921642  21.3877870            38.518188        699.55779
## occupation     6.4950263  55.2663471            43.834478        276.02936
## relationship   9.4268849 111.4604667            42.187781       1014.79284
## race           6.5831866  21.3334246            19.782214        146.12812
## sex           31.2937189   7.3454469            48.514890        110.94083
## hoursperweek   1.4185713  79.6194288            55.377402        725.58874
## nativecountry  6.6744890   1.8212620             6.691117         90.89926
```

crossvalidation approach for random forest

I did 5-fold cross validation

```
##        11
## 0.0395208
```

Boosting method

VSA

```
## [1] 0.1758387
```

CV Approach

```
## i:  1 Tue Apr 10 22:52:40 2018
## i:  2 Tue Apr 10 22:54:18 2018
## i:  3 Tue Apr 10 22:54:39 2018
## i:  4 Tue Apr 10 22:54:59 2018
## i:  5 Tue Apr 10 22:55:20 2018
```

```
## [1] 0.1717061
```