Group 24 HW5

Pratik Mante and Jiaqi Wang

4/6/2020

Problem 1:

a. Run a regression tree (RT) with the output variable Price and input variables Age_08_04, KM, Fuel_Type, HP, Automatic, Doors, Quarterly_Tax, Mfg_Guarantee, Guarantee_Period, Airco, Automatic_Airco, CD Player, Powered_Windows, Sport_Model, and Tow_Bar.

```
library(readxl)
ToyotaCorolla <- read_excel("/Users/pratikmante/Downloads/ToyotaCorolla.xlsx", sheet = "data")
library(fastDummies)
ToyotaCorolla_New <- fastDummies::dummy_cols(ToyotaCorolla ,select_columns = "Fuel_Type")
ToyotaCorolla_New <- ToyotaCorolla_New[,-8]
ToyotaCorolla_New <- fastDummies::dummy_cols(ToyotaCorolla_New,select_columns = "Color")
ToyotaCorolla_New <- ToyotaCorolla_New[,-10]
set.seed(100)
index <- sample(1:3, size = nrow(ToyotaCorolla_New),replace = T, prob = c(0.5,0.3,0.2))
ToyotaCorolla_train <- ToyotaCorolla_New[index==1,]
ToyotaCorolla_validation <- ToyotaCorolla_New[index==2,]
ToyotaCorolla_test <- ToyotaCorolla_New[index==3,]
library(rpart)
RT1 <- rpart(Price ~ Age_08_04 + KM + Fuel_Type_Diesel + Fuel_Type_Petrol + Fuel_Type_CNG + HP + Automa
printcp(RT1)
```

```
##
## Regression tree:
## rpart(formula = Price ~ Age_08_04 + KM + Fuel_Type_Diesel + Fuel_Type_Petrol +
##     Fuel_Type_CNG + HP + Automatic + Doors + Quarterly_Tax +
##     Mfr_Guarantee + Guarantee_Period + Airco + Automatic_airco +
##     CD_Player + Powered_Windows + Sport_Model + Tow_Bar, data = ToyotaCorolla_train,
##     method = "anova", control = rpart.control(maxdepth = 3))
##
## Variables actually used in tree construction:
## [1] Age_08_04       Automatic_airco HP              KM
##
## Root node error: 8989924787/694 = 12953782
##
## n= 694
##
##         CP nsplit rel error  xerror     xstd
## 1 0.661003      0   1.00000 1.00520 0.080381
## 2 0.112450      1   0.33900 0.34309 0.018684
## 3 0.021579      2   0.22655 0.23729 0.018237
## 4 0.021531      3   0.20497 0.23152 0.018304
## 5 0.020090      4   0.18344 0.21357 0.017343
```

```
## 6 0.015910      5   0.16335 0.20026 0.016409
## 7 0.010000      6   0.14744 0.18556 0.015014
```

i. Which appear to be the three or four most important car specifications for predicting the car's price?

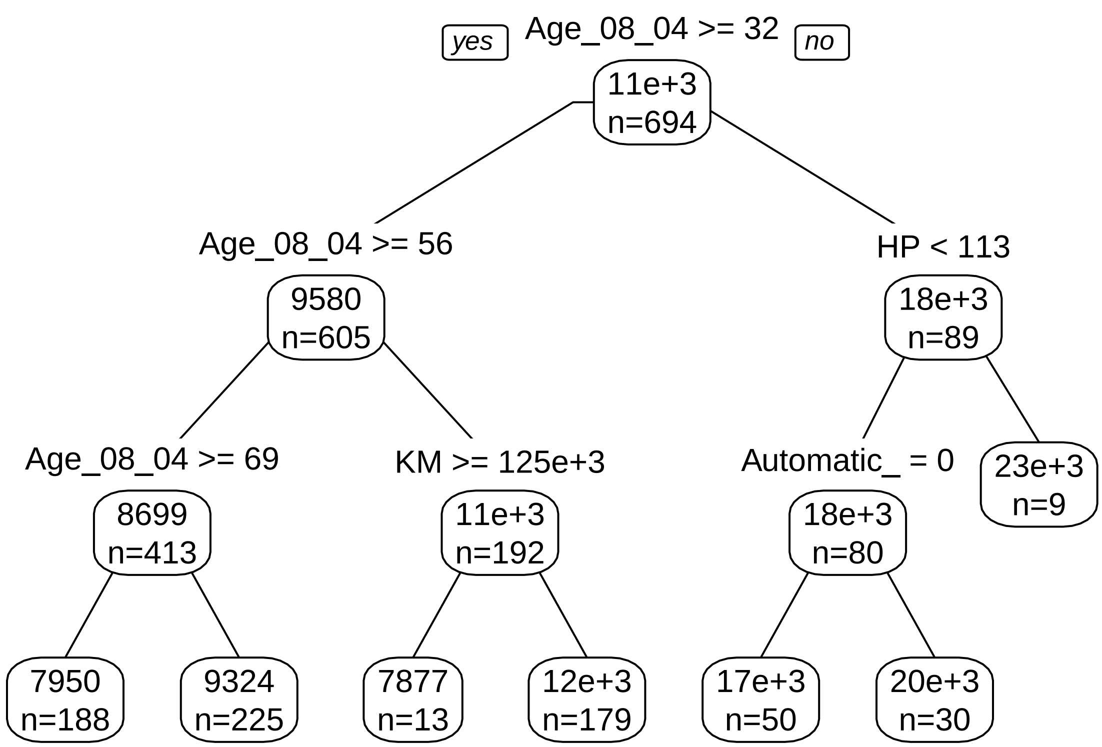
```r
summary(RT1)
```

```
## Call:
## rpart(formula = Price ~ Age_08_04 + KM + Fuel_Type_Diesel + Fuel_Type_Petrol +
##     Fuel_Type_CNG + HP + Automatic + Doors + Quarterly_Tax +
##     Mfr_Guarantee + Guarantee_Period + Airco + Automatic_airco +
##     CD_Player + Powered_Windows + Sport_Model + Tow_Bar, data = ToyotaCorolla_train,
##     method = "anova", control = rpart.control(maxdepth = 3))
##   n= 694
##
##           CP nsplit rel error    xerror       xstd
## 1 0.66100301      0 1.0000000 1.0052047 0.08038131
## 2 0.11244959      1 0.3389970 0.3430928 0.01868397
## 3 0.02157895      2 0.2265474 0.2372923 0.01823712
## 4 0.02153063      3 0.2049684 0.2315193 0.01830396
## 5 0.02009041      4 0.1834378 0.2135708 0.01734312
## 6 0.01591005      5 0.1633474 0.2002593 0.01640858
## 7 0.01000000      6 0.1474373 0.1855577 0.01501434
##
## Variable importance
##        Age_08_04  Automatic_airco               KM               HP
##               51               17               13                6
##    Quarterly_Tax Guarantee_Period        CD_Player
##                6                5                2
##
## Node number 1: 694 observations,    complexity param=0.661003
##   mean=10702.35, MSE=1.295378e+07
##   left son=2 (605 obs) right son=3 (89 obs)
##   Primary splits:
##       Age_08_04       < 31.5    to the right, improve=0.6610030, (0 missing)
##       KM              < 34740.5 to the right, improve=0.3455962, (0 missing)
##       Automatic_airco < 0.5     to the left,  improve=0.3367961, (0 missing)
##       CD_Player       < 0.5     to the left,  improve=0.2307198, (0 missing)
##       Airco           < 0.5     to the left,  improve=0.2009007, (0 missing)
##   Surrogate splits:
##       Automatic_airco  < 0.5     to the left,  agree=0.919, adj=0.371, (0 split)
##       KM               < 23395.5 to the right, agree=0.903, adj=0.247, (0 split)
##       Quarterly_Tax    < 203.5   to the left,  agree=0.889, adj=0.135, (0 split)
##       Guarantee_Period < 12.5    to the left,  agree=0.886, adj=0.112, (0 split)
##       HP               < 113     to the left,  agree=0.883, adj=0.090, (0 split)
##
## Node number 2: 605 observations,    complexity param=0.1124496
##   mean=9580.031, MSE=3836960
##   left son=4 (413 obs) right son=5 (192 obs)
##   Primary splits:
##       Age_08_04 < 55.5    to the right, improve=0.4354831, (0 missing)
##       KM        < 56004.5 to the right, improve=0.2195044, (0 missing)
##       HP        < 88      to the left,  improve=0.1542413, (0 missing)
##       CD_Player < 0.5     to the left,  improve=0.1381244, (0 missing)
##       Airco     < 0.5     to the left,  improve=0.1357493, (0 missing)
```

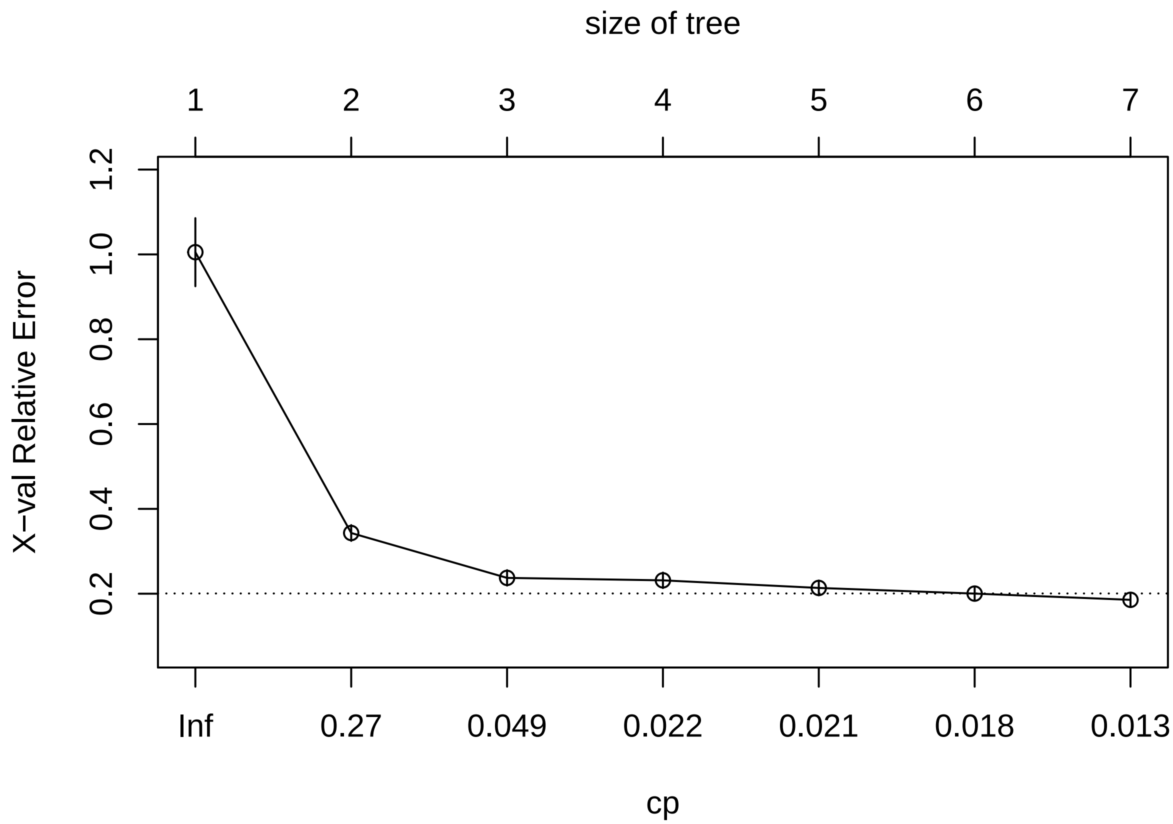
```
##   Surrogate splits:
##       CD_Player     < 0.5     to the left,  agree=0.767, adj=0.266, (0 split)
##       KM            < 33509.5 to the right, agree=0.727, adj=0.141, (0 split)
##       HP            < 70.5    to the right, agree=0.711, adj=0.089, (0 split)
##       Quarterly_Tax < 203.5   to the left,  agree=0.689, adj=0.021, (0 split)
##
## Node number 3: 89 observations,    complexity param=0.02157895
##   mean=18331.62, MSE=8159510
##   left son=6 (80 obs) right son=7 (9 obs)
##   Primary splits:
##       HP              < 113     to the left,  improve=0.2671360, (0 missing)
##       Automatic_airco < 0.5     to the left,  improve=0.2520412, (0 missing)
##       Age_08_04       < 18      to the right, improve=0.1694447, (0 missing)
##       KM              < 25085   to the right, improve=0.1523332, (0 missing)
##       Quarterly_Tax   < 222     to the left,  improve=0.1480581, (0 missing)
##   Surrogate splits:
##       Age_08_04 < 30.5    to the left,  agree=0.921, adj=0.222, (0 split)
##
## Node number 4: 413 observations,    complexity param=0.02153063
##   mean=8698.668, MSE=1783441
##   left son=8 (188 obs) right son=9 (225 obs)
##   Primary splits:
##       Age_08_04     < 68.5    to the right, improve=0.26278710, (0 missing)
##       KM            < 81427.5 to the right, improve=0.11367350, (0 missing)
##       Airco         < 0.5     to the left,  improve=0.10904710, (0 missing)
##       Quarterly_Tax < 78.5    to the left,  improve=0.09027762, (0 missing)
##       HP            < 93.5    to the left,  improve=0.07546663, (0 missing)
##   Surrogate splits:
##       KM            < 81427.5 to the right, agree=0.613, adj=0.149, (0 split)
##       Airco         < 0.5     to the left,  agree=0.574, adj=0.064, (0 split)
##       Tow_Bar       < 0.5     to the right, agree=0.574, adj=0.064, (0 split)
##       Quarterly_Tax < 52      to the left,  agree=0.564, adj=0.043, (0 split)
##       Automatic     < 0.5     to the right, agree=0.554, adj=0.021, (0 split)
##
## Node number 5: 192 observations,    complexity param=0.02009041
##   mean=11475.88, MSE=2988992
##   left son=10 (13 obs) right son=11 (179 obs)
##   Primary splits:
##       KM               < 124745  to the right, improve=0.3147160, (0 missing)
##       Age_08_04        < 43.5    to the right, improve=0.2348494, (0 missing)
##       HP               < 79      to the left,  improve=0.2006139, (0 missing)
##       Fuel_Type_Petrol < 0.5     to the left,  improve=0.1537284, (0 missing)
##       Fuel_Type_Diesel < 0.5     to the right, improve=0.1462282, (0 missing)
##
## Node number 6: 80 observations,    complexity param=0.01591005
##   mean=17836.42, MSE=6353228
##   left son=12 (50 obs) right son=13 (30 obs)
##   Primary splits:
##       Automatic_airco < 0.5     to the left,  improve=0.2814124, (0 missing)
##       Age_08_04       < 21      to the right, improve=0.2560410, (0 missing)
##       KM              < 21572   to the right, improve=0.1878980, (0 missing)
##       HP              < 104     to the left,  improve=0.1716506, (0 missing)
##       Quarterly_Tax   < 222     to the left,  improve=0.1419533, (0 missing)
##   Surrogate splits:
```

```
##       HP             < 104     to the left,  agree=0.738, adj=0.300, (0 split)
##       Automatic      < 0.5     to the left,  agree=0.662, adj=0.100, (0 split)
##       Quarterly_Tax < 222     to the left,  agree=0.650, adj=0.067, (0 split)
##
## Node number 7: 9 observations
##   mean=22733.33, MSE=2660556
##
## Node number 8: 188 observations
##   mean=7949.734, MSE=974180.2
##
## Node number 9: 225 observations
##   mean=9324.444, MSE=1599362
##
## Node number 10: 13 observations
##   mean=7876.923, MSE=4015621
##
## Node number 11: 179 observations
##   mean=11737.26, MSE=1905431
##
## Node number 12: 50 observations
##   mean=16800.7, MSE=4363171
##
## Node number 13: 30 observations
##   mean=19562.63, MSE=4902317
```

```
library(rpart.plot)
prp(RT1, type = 1, extra = 1, split.font = 1, varlen = -10)
```




```
plotcp(RT1)
```



ii. Compare the prediction errors of the training, validation, and test sets by examining their RMS error and by plotting the three boxplots. What is happening with the training set predictions? How does the predictive performance of the test set compare to the other two? Why does this occur?

Answer: It can be observed that RMSE value of training set is less than that of validation and test set. We get this result as model is trained on training set and from these results we can tell that there is issue of overfitting. It can also be observed that RMSE of test data is highest which means that it has least efficiency of prediction. This is due to not training the model on it and the new data.

```
RT1_train_pred <- predict(RT1, ToyotaCorolla_train, type = "vector")
RMSE_train = function(x,y){ sqrt(mean((RT1_train_pred - ToyotaCorolla_train$Price)^2))}
RMSE_train(RT1_train_pred, ToyotaCorolla_train$Price)
```

```
## [1] 1381.981
```

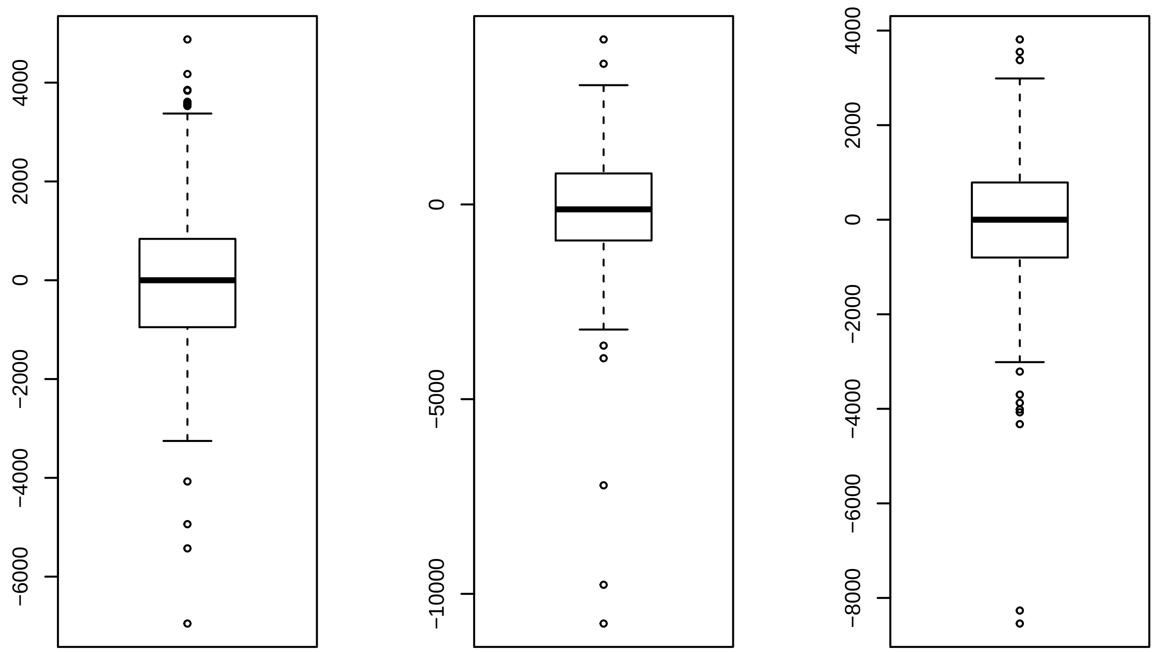
```
RT1_valid_pred <- predict(RT1, ToyotaCorolla_validation, type = "vector")
RMSE_valid = function(x,y){ sqrt(mean((RT1_valid_pred - ToyotaCorolla_validation$Price)^2))}
RMSE_valid(RT1_valid_pred, ToyotaCorolla_validation$Price)
```

```
## [1] 1493.076
```

```
RT1_test_pred <- predict(RT1, ToyotaCorolla_test, type = "vector")
RMSE_test = function(x,y){ sqrt(mean((RT1_test_pred - ToyotaCorolla_test$Price)^2))}
RMSE_test(RT1_test_pred, ToyotaCorolla_test$Price)
```

```
## [1] 1479.683
```

```
par(mfrow=c(1,3))
boxplot(RT1_train_pred-ToyotaCorolla_train$Price)
boxplot(RT1_valid_pred-ToyotaCorolla_validation$Price)
boxplot(RT1_test_pred-ToyotaCorolla_test$Price)
```

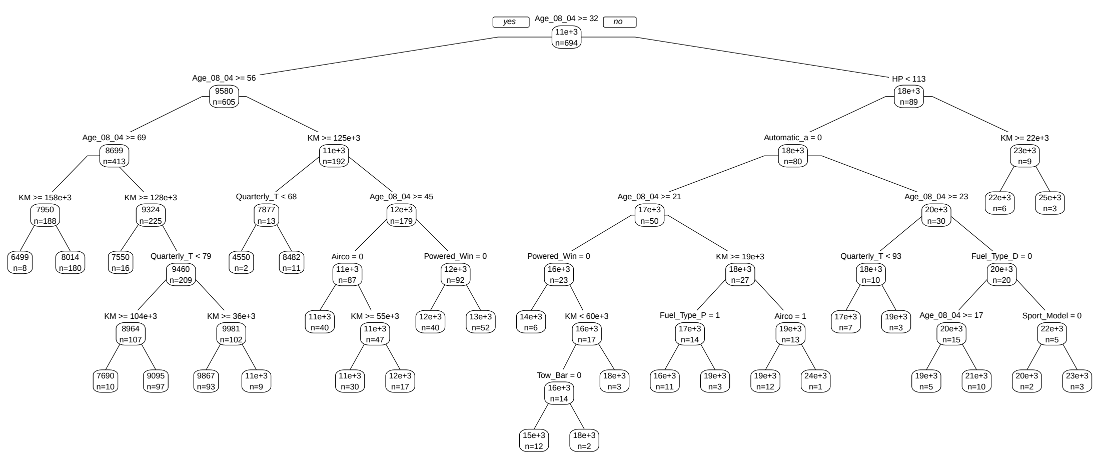



```r
par(mfrow=c(1,1))
```

iii. If we used the full tree instead of the best pruned tree to score the validation set, how would this affect the predictive performance for the validation set? (Hint: Does the full tree use the validation data?)

Answer: It can be observed that pruned tree has better predictive performance than that of full tree. Pruned tree has reduced validation error which is 1275.245.

```r
RT1.cv <- rpart(Price ~ Age_08_04 + KM + Fuel_Type_Diesel + Fuel_Type_Petrol + Fuel_Type_CNG + HP + Aut
RT1_Pruned <- prune(RT1.cv,cp=RT1.cv$cptable[which.min(RT1.cv$cptable[,"xerror"])])
prp(RT1_Pruned, type = 1, extra = 1, split.font = 1, varlen = -10)
```




```r
printcp(RT1_Pruned)
```

```
##
## Regression tree:
## rpart(formula = Price ~ Age_08_04 + KM + Fuel_Type_Diesel + Fuel_Type_Petrol +
##     Fuel_Type_CNG + HP + Automatic + Doors + Quarterly_Tax +
##     Mfr_Guarantee + Guarantee_Period + Airco + Automatic_airco +
##     CD_Player + Powered_Windows + Sport_Model + Tow_Bar, data = ToyotaCorolla_train,
```

```
##     method = "anova", cp = 1e-05, minsplit = 2, xval = 5)
##
## Variables actually used in tree construction:
##  [1] Age_08_04        Airco            Automatic_airco  Fuel_Type_Diesel
##  [5] Fuel_Type_Petrol HP               KM               Powered_Windows
##  [9] Quarterly_Tax    Sport_Model      Tow_Bar
##
## Root node error: 8989924787/694 = 12953782
##
## n= 694
##
##           CP nsplit rel error  xerror     xstd
## 1  0.6610030      0  1.000000 1.00151 0.080249
## 2  0.1124496      1  0.338997 0.34261 0.019058
## 3  0.0215790      2  0.226547 0.23673 0.018633
## 4  0.0215306      3  0.204968 0.23027 0.018524
## 5  0.0200904      4  0.183438 0.21801 0.017902
## 6  0.0159101      5  0.163347 0.18501 0.016060
## 7  0.0106018      6  0.147437 0.17666 0.014759
## 8  0.0080155      7  0.136836 0.15712 0.011649
## 9  0.0060325      8  0.128820 0.14769 0.010579
## 10 0.0060121      9  0.122788 0.14517 0.010509
## 11 0.0056351     10  0.116775 0.14517 0.010509
## 12 0.0029507     11  0.111140 0.14807 0.011354
## 13 0.0029101     12  0.108190 0.14770 0.011494
## 14 0.0027911     13  0.105280 0.14849 0.011508
## 15 0.0023600     14  0.102488 0.14817 0.011562
## 16 0.0023155     15  0.100128 0.14850 0.011434
## 17 0.0019909     16  0.097813 0.14702 0.011316
## 18 0.0019568     17  0.095822 0.14773 0.011341
## 19 0.0019464     18  0.093865 0.14764 0.011298
## 20 0.0018666     19  0.091919 0.14658 0.011289
## 21 0.0018132     20  0.090052 0.14530 0.011245
## 22 0.0017803     21  0.088239 0.14541 0.011235
## 23 0.0017755     22  0.086459 0.14672 0.011293
## 24 0.0015145     23  0.084683 0.14587 0.011375
## 25 0.0014741     24  0.083169 0.14505 0.011276
## 26 0.0014462     25  0.081695 0.14369 0.011254
## 27 0.0014073     26  0.080248 0.14384 0.011282
## 28 0.0014056     27  0.078841 0.14351 0.011281
## 29 0.0013783     28  0.077436 0.14420 0.011306
## 30 0.0013682     29  0.076057 0.14348 0.011123
```

```
pruned_valid <- predict(RT1_Pruned, ToyotaCorolla_validation, type = "vector")
RMSE_pruned = function(x,y){ sqrt(mean((pruned_valid - ToyotaCorolla_validation$Price)^2))}
RMSE_pruned(pruned_valid, ToyotaCorolla_validation$Price)
```

```
## [1] 1267.87
```

b. Let us see the effect of turning the price variable into a categorical variable. First, create a new variable that categorizes price into 20 bins of equal counts. Now repartition the data keeping Binned Price instead of Price. Run a classification tree (CT) with the same set of input variables as in the RT, and with Binned Price as the output variable.

```r
ToyotaCorolla_New$Binnedprice <- as.factor(as.numeric(cut(ToyotaCorolla_New$Price,20)))
ToyotaCorolla_New <- ToyotaCorolla_New[,-3]
set.seed(100)
index <- sample(1:3, size = nrow(ToyotaCorolla_New), replace = T, prob = c(0.5,0.3,0.2))
ToyotaCorolla_train2 <- ToyotaCorolla_New[index==1,]
ToyotaCorolla_validation2 <- ToyotaCorolla_New[index==2,]
ToyotaCorolla_test2 <- ToyotaCorolla_New[index==3,]
```

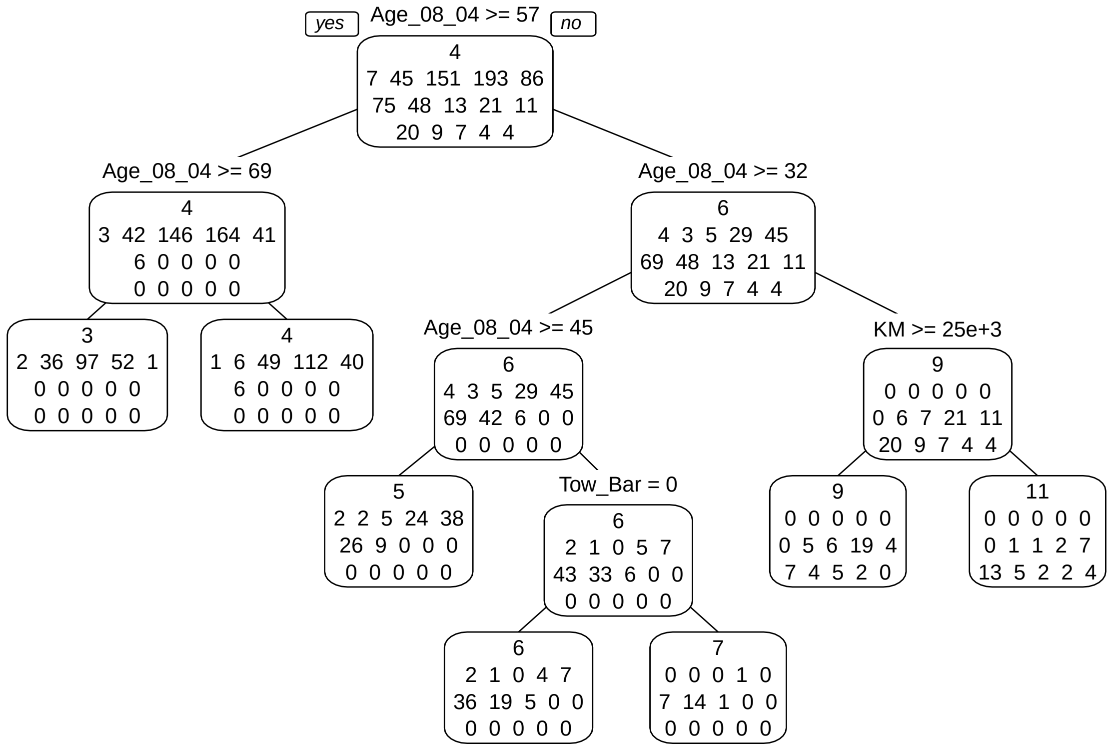
i. Compare the tree generated by the CT with the one generated by the RT. Are they different? (Look at structure, the top predictors, size of tree, etc.) Why?

Answer: It is observed that classification tree has more branches than that of regression tree. Also, regression tree had 4 more important variables wheread classification tree has 2 important variables: Age_08_04 and KM.

```r
CT <- rpart(Binnedprice ~ Age_08_04 + KM + Fuel_Type_Diesel + Fuel_Type_CNG + Fuel_Type_Petrol + HP + Au
Mfr_Guarantee + Guarantee_Period + Airco + Automatic_airco + CD_Player + Powered_Windows + Sport_Model +
printcp(CT)
```

```
##
## Classification tree:
## rpart(formula = Binnedprice ~ Age_08_04 + KM + Fuel_Type_Diesel +
##     Fuel_Type_CNG + Fuel_Type_Petrol + HP + Automatic + Doors +
##     Quarterly_Tax + Mfr_Guarantee + Guarantee_Period + Airco +
##     Automatic_airco + CD_Player + Powered_Windows + Sport_Model +
##     Tow_Bar, data = ToyotaCorolla_train2, method = "class")
##
## Variables actually used in tree construction:
## [1] Age_08_04 KM        Tow_Bar
##
## Root node error: 501/694 = 0.7219
##
## n= 694
##
##         CP nsplit rel error  xerror     xstd
## 1 0.084830      0   1.00000 1.00000 0.023560
## 2 0.041916      2   0.83034 0.83034 0.025766
## 3 0.023952      3   0.78842 0.80639 0.025934
## 4 0.021956      4   0.76447 0.81038 0.025908
## 5 0.013972      5   0.74251 0.78443 0.026059
## 6 0.010000      6   0.72854 0.80240 0.025959
```

```r
prp(CT, type = 1, extra = 1, split.font = 1, varlen = -10)
```




```r
summary(CT)
```

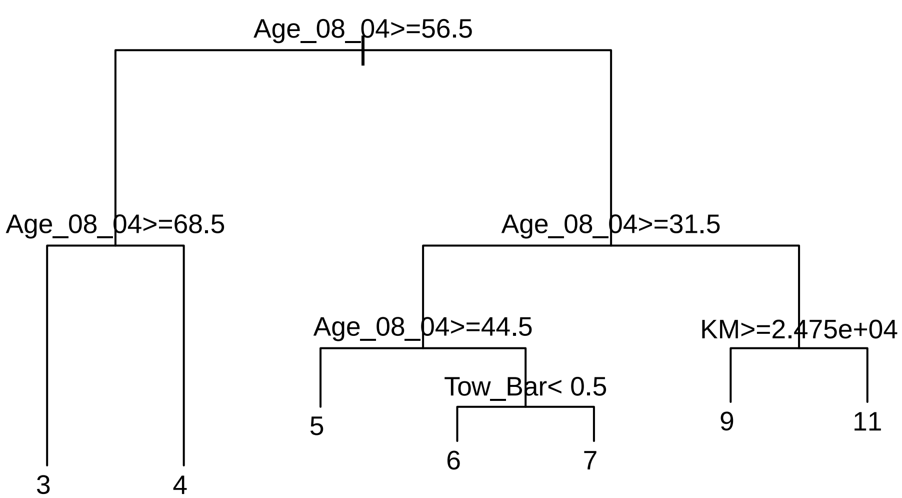
```
## Call:
## rpart(formula = Binnedprice ~ Age_08_04 + KM + Fuel_Type_Diesel +
##     Fuel_Type_CNG + Fuel_Type_Petrol + HP + Automatic + Doors +
##     Quarterly_Tax + Mfr_Guarantee + Guarantee_Period + Airco +
##     Automatic_airco + CD_Player + Powered_Windows + Sport_Model +
##     Tow_Bar, data = ToyotaCorolla_train2, method = "class")
##   n= 694
##
##           CP nsplit rel error    xerror       xstd
## 1 0.08483034      0 1.0000000 1.0000000 0.02356026
## 2 0.04191617      2 0.8303393 0.8303393 0.02576627
## 3 0.02395210      3 0.7884232 0.8063872 0.02593416
## 4 0.02195609      4 0.7644711 0.8103792 0.02590847
## 5 0.01397206      5 0.7425150 0.7844311 0.02605926
## 6 0.01000000      6 0.7285429 0.8023952 0.02595894
##
## Variable importance
##        Age_08_04               KM        CD_Player  Automatic_airco
##               48               16                9                7
##            Airco      Sport_Model    Quarterly_Tax          Tow_Bar
##                5                5                4                2
## Guarantee_Period    Mfr_Guarantee            Doors
##                1                1                1
##
## Node number 1: 694 observations,    complexity param=0.08483034
##   predicted class=4   expected loss=0.721902  P(node) =1
##     class counts:     7    45   151   193    86    75    48    13    21    11    20     9     7
##    probabilities: 0.010 0.065 0.218 0.278 0.124 0.108 0.069 0.019 0.030 0.016 0.029 0.013 0.010 0.00
```

```
##   left son=2 (402 obs) right son=3 (292 obs)
##   Primary splits:
##       Age_08_04 < 56.5    to the right, improve=53.76043, (0 missing)
##       KM        < 56066   to the right, improve=22.31163, (0 missing)
##       CD_Player < 0.5     to the left,  improve=16.69778, (0 missing)
##       Airco     < 0.5     to the left,  improve=15.38055, (0 missing)
##       HP        < 88      to the left,  improve=11.83687, (0 missing)
##   Surrogate splits:
##       KM              < 53791   to the right, agree=0.744, adj=0.390, (0 split)
##       CD_Player       < 0.5     to the left,  agree=0.744, adj=0.390, (0 split)
##       Airco           < 0.5     to the left,  agree=0.661, adj=0.195, (0 split)
##       Automatic_airco < 0.5     to the left,  agree=0.630, adj=0.120, (0 split)
##       Quarterly_Tax   < 203.5   to the left,  agree=0.611, adj=0.075, (0 split)
##
## Node number 2: 402 observations,    complexity param=0.08483034
##   predicted class=4   expected loss=0.5920398  P(node) =0.5792507
##     class counts:     3    42   146   164    41     6     0     0     0     0     0     0     0
##    probabilities: 0.007 0.104 0.363 0.408 0.102 0.015 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.00
##   left son=4 (188 obs) right son=5 (214 obs)
##   Primary splits:
##       Age_08_04      < 68.5    to the right, improve=20.393330, (0 missing)
##       Airco          < 0.5     to the left,  improve= 8.736897, (0 missing)
##       KM             < 81427.5 to the right, improve= 6.987451, (0 missing)
##       Quarterly_Tax  < 78.5    to the left,  improve= 6.023962, (0 missing)
##       Powered_Windows < 0.5    to the left,  improve= 3.764136, (0 missing)
##   Surrogate splits:
##       KM            < 81427.5 to the right, agree=0.607, adj=0.160, (0 split)
##       Airco         < 0.5     to the left,  agree=0.577, adj=0.096, (0 split)
##       Tow_Bar       < 0.5     to the right, agree=0.567, adj=0.074, (0 split)
##       Quarterly_Tax < 52      to the left,  agree=0.555, adj=0.048, (0 split)
##       Mfr_Guarantee < 0.5     to the left,  agree=0.555, adj=0.048, (0 split)
##
## Node number 3: 292 observations,    complexity param=0.04191617
##   predicted class=6   expected loss=0.7636986  P(node) =0.4207493
##     class counts:     4     3     5    29    45    69    48    13    21    11    20     9     7
##    probabilities: 0.014 0.010 0.017 0.099 0.154 0.236 0.164 0.045 0.072 0.038 0.068 0.031 0.024 0.01
##   left son=6 (203 obs) right son=7 (89 obs)
##   Primary splits:
##       Age_08_04       < 31.5    to the right, improve=21.659410, (0 missing)
##       Sport_Model     < 0.5     to the left,  improve=11.748960, (0 missing)
##       Automatic_airco < 0.5     to the left,  improve= 9.572499, (0 missing)
##       Powered_Windows < 0.5     to the left,  improve= 7.373486, (0 missing)
##       Airco           < 0.5     to the left,  improve= 6.514570, (0 missing)
##   Surrogate splits:
##       Sport_Model      < 0.5     to the left,  agree=0.853, adj=0.517, (0 split)
##       Automatic_airco  < 0.5     to the left,  agree=0.815, adj=0.393, (0 split)
##       KM               < 23395.5 to the right, agree=0.777, adj=0.270, (0 split)
##       Quarterly_Tax    < 203.5   to the left,  agree=0.740, adj=0.146, (0 split)
##       Guarantee_Period < 12.5    to the left,  agree=0.729, adj=0.112, (0 split)
##
## Node number 4: 188 observations
##   predicted class=3   expected loss=0.4840426  P(node) =0.2708934
##     class counts:     2    36    97    52     1     0     0     0     0     0     0     0     0
##    probabilities: 0.011 0.191 0.516 0.277 0.005 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.00
```

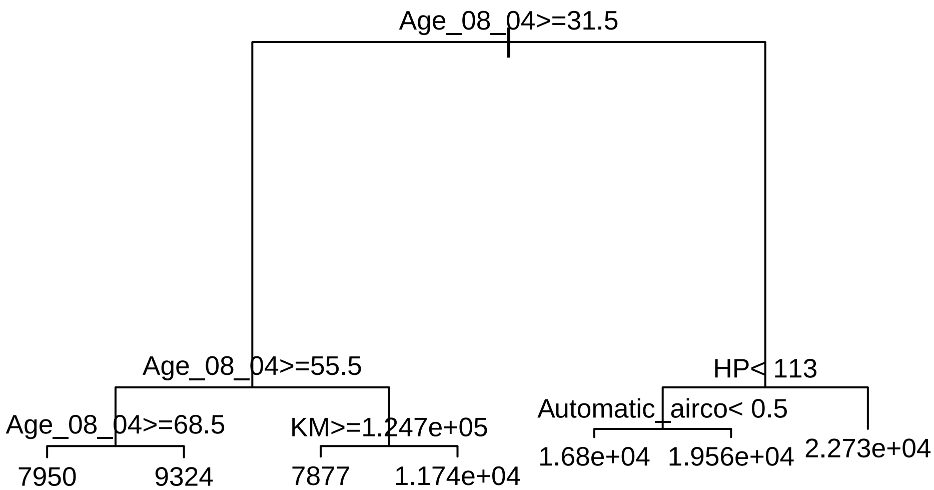
```
##
## Node number 5: 214 observations
##   predicted class=4   expected loss=0.4766355  P(node) =0.3083573
##     class counts:     1     6    49   112    40     6     0     0     0     0     0     0     0     0
##    probabilities: 0.005 0.028 0.229 0.523 0.187 0.028 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.00
##
## Node number 6: 203 observations,    complexity param=0.0239521
##   predicted class=6   expected loss=0.6600985  P(node) =0.2925072
##     class counts:     4     3     5    29    45    69    42     6     0     0     0     0     0     0
##    probabilities: 0.020 0.015 0.025 0.143 0.222 0.340 0.207 0.030 0.000 0.000 0.000 0.000 0.000 0.00
##   left son=12 (106 obs) right son=13 (97 obs)
##   Primary splits:
##       Age_08_04     < 44.5    to the right, improve=11.298930, (0 missing)
##       Doors         < 3.5     to the left,  improve= 6.109370, (0 missing)
##       Quarterly_Tax < 78.5    to the left,  improve= 5.786206, (0 missing)
##       Airco         < 0.5     to the left,  improve= 5.302420, (0 missing)
##       Mfr_Guarantee < 0.5     to the left,  improve= 5.047244, (0 missing)
##   Surrogate splits:
##       KM              < 47016   to the right, agree=0.660, adj=0.289, (0 split)
##       Mfr_Guarantee   < 0.5     to the left,  agree=0.601, adj=0.165, (0 split)
##       Doors           < 4.5     to the left,  agree=0.586, adj=0.134, (0 split)
##       Quarterly_Tax   < 78.5    to the left,  agree=0.567, adj=0.093, (0 split)
##       Powered_Windows < 0.5     to the left,  agree=0.557, adj=0.072, (0 split)
##
## Node number 7: 89 observations,    complexity param=0.02195609
##   predicted class=9   expected loss=0.7640449  P(node) =0.1282421
##     class counts:     0     0     0     0     0     0     6     7    21    11    20     9     7     4
##    probabilities: 0.000 0.000 0.000 0.000 0.000 0.000 0.067 0.079 0.236 0.124 0.225 0.101 0.079 0.04
##   left son=14 (52 obs) right son=15 (37 obs)
##   Primary splits:
##       KM              < 24750   to the right, improve=4.024878, (0 missing)
##       Automatic_airco < 0.5     to the left,  improve=3.301475, (0 missing)
##       Quarterly_Tax   < 92.5    to the left,  improve=3.180077, (0 missing)
##       HP              < 113     to the left,  improve=2.846286, (0 missing)
##       Age_08_04       < 12.5    to the right, improve=2.523573, (0 missing)
##   Surrogate splits:
##       Age_08_04        < 15.5    to the right, agree=0.787, adj=0.486, (0 split)
##       HP               < 97.5    to the left,  agree=0.685, adj=0.243, (0 split)
##       Guarantee_Period < 7.5     to the left,  agree=0.652, adj=0.162, (0 split)
##       Automatic        < 0.5     to the left,  agree=0.596, adj=0.027, (0 split)
##       Quarterly_Tax    < 41.5    to the right, agree=0.596, adj=0.027, (0 split)
##
## Node number 12: 106 observations
##   predicted class=5   expected loss=0.6415094  P(node) =0.1527378
##     class counts:     2     2     5    24    38    26     9     0     0     0     0     0     0     0
##    probabilities: 0.019 0.019 0.047 0.226 0.358 0.245 0.085 0.000 0.000 0.000 0.000 0.000 0.000 0.00
##
## Node number 13: 97 observations,    complexity param=0.01397206
##   predicted class=6   expected loss=0.556701  P(node) =0.1397695
##     class counts:     2     1     0     5     7    43    33     6     0     0     0     0     0     0
##    probabilities: 0.021 0.010 0.000 0.052 0.072 0.443 0.340 0.062 0.000 0.000 0.000 0.000 0.000 0.00
##   left son=26 (74 obs) right son=27 (23 obs)
##   Primary splits:
##       Tow_Bar         < 0.5     to the left,  improve=2.940579, (0 missing)
```

```
##       Powered_Windows < 0.5     to the left,  improve=2.691563, (0 missing)
##       Airco           < 0.5     to the left,  improve=2.276473, (0 missing)
##       Age_08_04       < 33.5    to the right, improve=2.213130, (0 missing)
##       KM              < 48671.5 to the right, improve=2.095971, (0 missing)
##   Surrogate splits:
##       Automatic < 0.5     to the left,  agree=0.773, adj=0.043, (0 split)
##
## Node number 14: 52 observations
##   predicted class=9   expected loss=0.6346154  P(node) =0.07492795
##     class counts:     0     0     0     0     0     0     5     6    19     4     7     4     5     1
##    probabilities: 0.000 0.000 0.000 0.000 0.000 0.000 0.096 0.115 0.365 0.077 0.135 0.077 0.096 0.03
##
## Node number 15: 37 observations
##   predicted class=11  expected loss=0.6486486  P(node) =0.05331412
##     class counts:     0     0     0     0     0     0     1     1     2     7    13     5     2     1
##    probabilities: 0.000 0.000 0.000 0.000 0.000 0.000 0.027 0.027 0.054 0.189 0.351 0.135 0.054 0.05
##
## Node number 26: 74 observations
##   predicted class=6   expected loss=0.5135135  P(node) =0.1066282
##     class counts:     2     1     0     4     7    36    19     5     0     0     0     0     0     0
##    probabilities: 0.027 0.014 0.000 0.054 0.095 0.486 0.257 0.068 0.000 0.000 0.000 0.000 0.000 0.00
##
## Node number 27: 23 observations
##   predicted class=7   expected loss=0.3913043  P(node) =0.03314121
##     class counts:     0     0     0     1     0     7    14     1     0     0     0     0     0     0
##    probabilities: 0.000 0.000 0.000 0.043 0.000 0.304 0.609 0.043 0.000 0.000 0.000 0.000 0.000 0.00
```

```r
plot(CT, margin = 0.07)
text(CT, cex = 0.8)
```




```r
plot(RT1, margin = 0.07)
text(RT1, cex = 0.8)
```



ii. Predict the price, using the RT and the CT, of a used Toyota Corolla with the specifications listed in Table below.

```
new_test <- data.frame(Age_08_04=77,KM=117000,Fuel_Type_CNG=0,Fuel_Type_Diesel=0,Fuel_Type_Petrol=1,HP=
predict(RT1,new_test)
```

```
##        1
## 7949.734
```

```
predict(CT,new_test, type = "class")
```

```
## 1
## 3
## Levels: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 19 20
```

```
(max(ToyotaCorolla[which(ToyotaCorolla_New$Binnedprice == 3),3]) + min(ToyotaCorolla[which(ToyotaCorolla
```

```
## [1] 7850
```

iii. Compare the predictions in terms of the predictors that were used, the magnitude of the difference between the two predictions, and the advantages and disadvantages of the two methods.

Answer: Predicted Values for regression tree: 7949.734 Predicted Values for classification tree: 7850 Magnitude Difference between two predictors: 7949.734 - 7850 = 99.734 Advantage of Regression tree: performs variable screening implicitly and it is easy to interpret Advantage of Classification tree: consumes less time and less complex compared to regression tree Disadvantage of Regression tree: high complexity and time consuming Disadvantage of Classification tree: it is not easy to interpret

---

Problem 2:

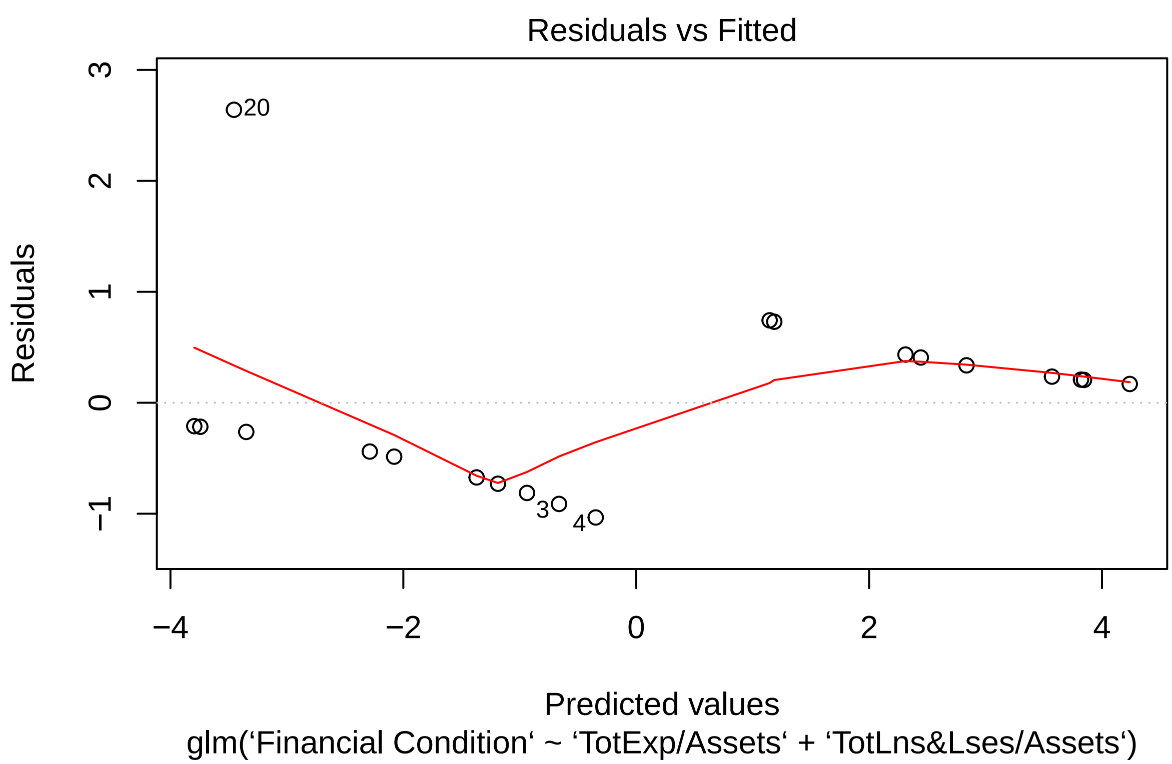
a. Write the estimated equation that associates the financial condition of a bank with its two predictors in three formats:

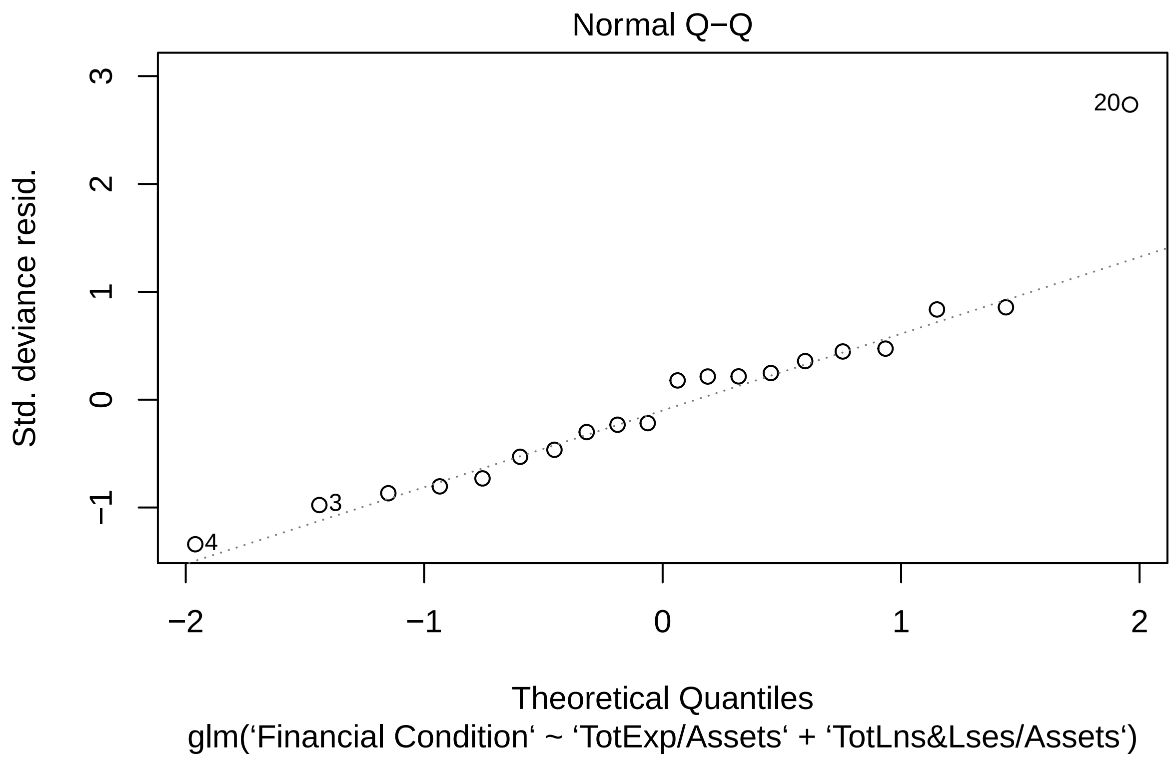
```
library(readxl)
Bank <- read_excel("/Users/pratikmante/Downloads/Banks.xlsx")
Bank$`Financial Condition` <- factor(Bank$`Financial Condition`, levels = c(1,0))
logit.bank <- glm(`Financial Condition` ~ `TotExp/Assets` + `TotLns&Lses/Assets`, data = Bank, family =
summary(logit.bank)
```

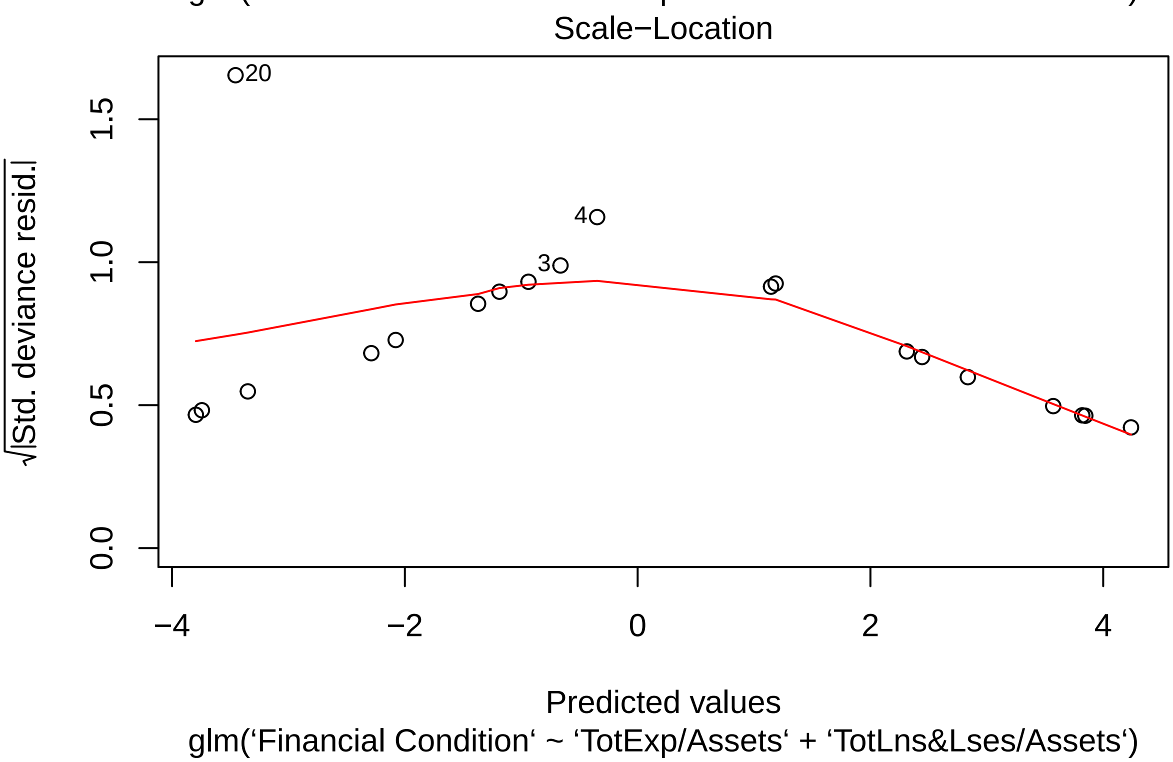
```
##
## Call:
## glm(formula = `Financial Condition` ~ `TotExp/Assets` + `TotLns&Lses/Assets`,
```

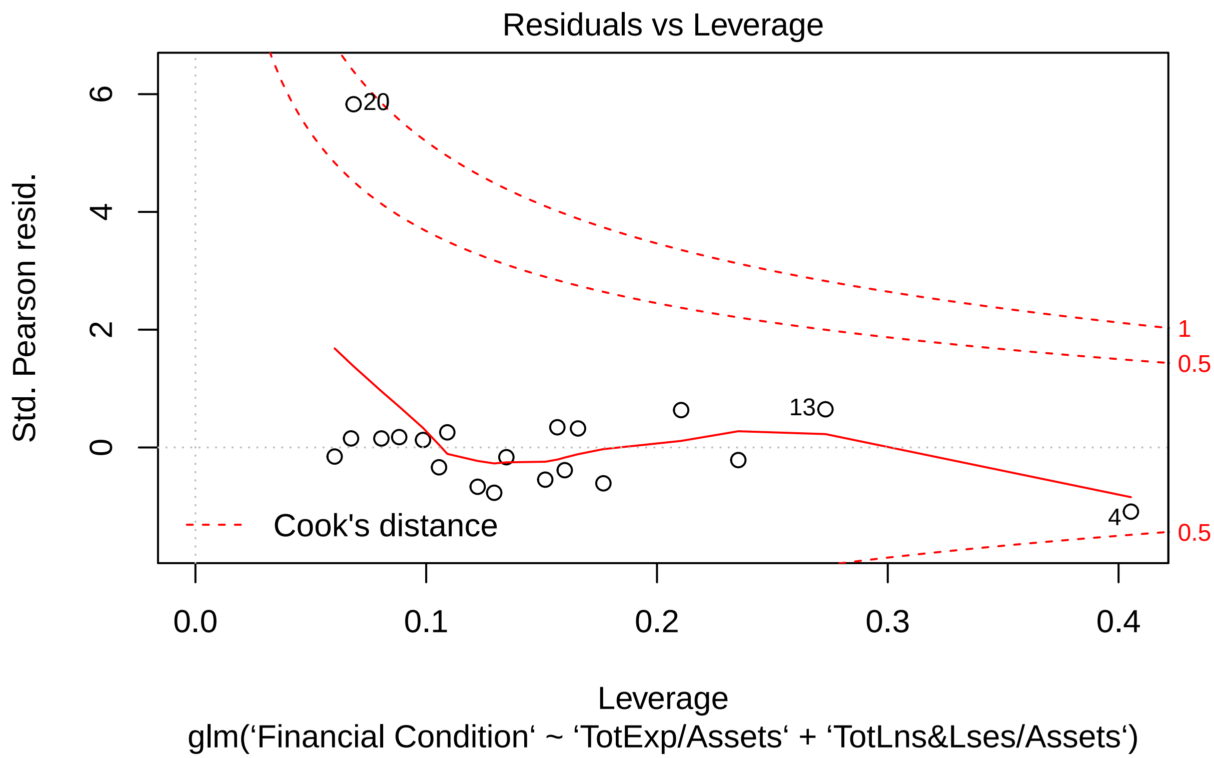
```
##     family = binomial(link = "logit"), data = Bank)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.03373  -0.53234  -0.02079   0.35514   2.64035
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)              14.188      6.122   2.317   0.0205 *
## `TotExp/Assets`         -79.964     39.263  -2.037   0.0417 *
## `TotLns&Lses/Assets`     -9.173      6.864  -1.336   0.1814
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27.726  on 19  degrees of freedom
## Residual deviance: 12.831  on 17  degrees of freedom
## AIC: 18.831
##
## Number of Fisher Scoring iterations: 6
```

```r
plot(logit.bank)
```




Normal Q−Q
Std. deviance resid.
Theoretical Quantiles
glm(‘Financial Condition‘ ~ ‘TotExp/Assets‘ + ‘TotLns&Lses/Assets‘)
20
3
4
−2
−1
0
1
2
3


Scale−Location
√|Std. deviance resid.|
Predicted values
glm(‘Financial Condition‘ ~ ‘TotExp/Assets‘ + ‘TotLns&Lses/Assets‘)
20
4
3
0.0
0.5
1.0
1.5
−4
−2
0
2
4



x1 = TotExp/Assets x2 = TotLns&Lses/Assets

i. The logit as a function of the predictors Answer: log(p/(i-p)) = 14.188 + (-79.964)x1 + (-9.173)x2
ii. The odds as a function of the predictors Answer: odds = e ^ (14.188 + (-79.964)x1 + (-9.173)x2)
iii. The probability as a function of the predictors Answer: probability = 1/(1 + e ^ (14.188 + (-79.964)x1 + (-9.173)x2))

b. Consider a new bank whose total loans and leases/assets ratio = 0.6 and totalexpenses/assets ratio = 0.11. From your logistic regression model, estimate the following four quantities for this bank: the logit, the odds, the probability of being financially weak, and the classification of the bank.

```
test_bank <- data.frame("TotExp/Assets" = 0.11, "TotLns&Lses/Assets" = 0.6, check.names = FALSE)
logit <- predict.glm(logit.bank, test_bank)
logit
```

```
##          1
## -0.1124105
```

```
odds <- exp(logit)
odds
```

```
##         1
## 0.8936774
```

```
probability <- predict.glm(logit.bank, test_bank, type = "response")
probability
```

```
##         1
## 0.4719269
```

```
glm.pred <- ifelse(probability>0.5, "1 - weak", "0 - strong")
glm.pred
```

```
##            1
## "0 - strong"
```

c. The cutoff value of 0.5 is used in conjunction with the probability of being financially weak. Compute the threshold that should be used if we want to make a classification based on the odds of being financially weak, and the threshold for the corresponding logit.

Answer: For cutoff value of 0.5, threshold for odds of being financially weak :

d. Interpret the estimated coefficient for the total loans & leases to total assets ratio (TotLns&Lses/Assets) in terms of the odds of being financially weak.

Answer: From the very low value of estimated coefficent it can be interpreted that the ratio (TotLns&Lses/Assets) does not have signoficant effect as compared to other variables.

```
coef_odds <- exp(coefficients(logit.bank)[3])
coef_odds
```

```
## `TotLns&Lses/Assets`
##         0.0001037824
```

e. When a bank that is in poor financial condition is misclassified as financially strong, the misclassification cost is much higher than when a financially strong bank is misclassified as weak. To minimize the expected cost of misclassification, should the cutoff value for classification (which is currently at 0.5) be increased or decreased?

Answer: It can be observed that the rate of predicting "strong when actually it is weak", is high when the value of cutoff is higher than 0.5. But the rate of predicting "strong when actually it is weak" decreases with the decrease in cutoff value. The best cutoff value is 0.02.

```
p <- ifelse(logit.bank$fitted.values>= 0.5, 1,0)
table(p, Bank$`Financial Condition`)
```

```
##
## p    1  0
##   0 10  1
##   1  0  9
```

```
p <- ifelse(logit.bank$fitted.values>= 0.8, 1,0)
table(p, Bank$`Financial Condition`)
```

```
##
## p    1  0
##   0 10  3
##   1  0  7
```

```
p <- ifelse(logit.bank$fitted.values>= 0.9, 1,0)
table(p, Bank$`Financial Condition`)
```

```
##
## p    1  0
##   0 10  3
##   1  0  7
```

```
p <- ifelse(logit.bank$fitted.values>= 0.3, 1,0)
table(p, Bank$`Financial Condition`)
```

```
##
## p   1 0
##   0 8 1
##   1 2 9
```

```
p <- ifelse(logit.bank$fitted.values>= 0.1, 1,0)
table(p, Bank$`Financial Condition`)
```

```
##
## p   1 0
##   0 4 1
##   1 6 9
```

```
p <- ifelse(logit.bank$fitted.values>= 0.05, 1,0)
table(p, Bank$`Financial Condition`)
```

```
##
## p   1 0
##   0 3 1
##   1 7 9
```

```
p <- ifelse(logit.bank$fitted.values>= 0.03, 1,0)
table(p, Bank$`Financial Condition`)
```

```
##
## p    1  0
##   0  2  0
##   1  8 10
```

```
p <- ifelse(logit.bank$fitted.values>= 0.02, 1,0)
table(p, Bank$`Financial Condition`)
```

```
##
## p    1  0
##   1 10 10
```

```
p <- ifelse(logit.bank$fitted.values>= 0.01, 1,0)
table(p, Bank$`Financial Condition`)
```

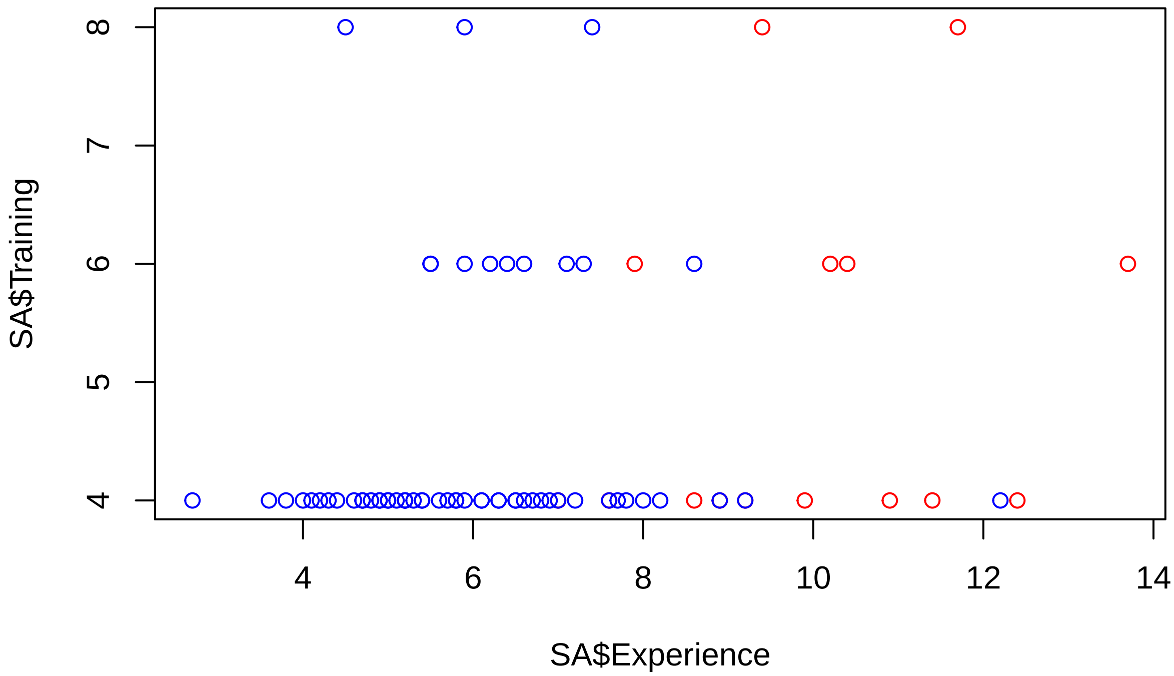
```
##
## p    1  0
##   1 10 10
```

Problem 3:

a. Create a scatterplot of Experience versus Training using color or symbol to differentiate programmers who complete the task from those who did not complete it. Which predictor(s) appear(s) potentially useful for classifying task completion?

Answer: It can be observed from the graph that "experience" has more impact than "training" on task completion. It can be seen that with increase in "experience" task completion also increases.

```
library(readxl)
SA <- read_excel("/Users/pratikmante/Downloads/System Administrators.xlsx")
plot(SA$Experience, SA$Training, col = ifelse(SA$`Completed task` == "Yes", "Red", "Blue"))
```



b. Run a logistic regression model with both predictors using the entire dataset as training data. Among those who complete the task, what is the percentage of programmers who are incorrectly classified as failing to complete the task?

Answer: Percentage of misclassification: (5/15)*100 = 33.33%

```
SA$`Completed task` <- ifelse(SA$`Completed task` == "Yes", "0", "1")
SA$`Completed task` <- as.numeric(SA$`Completed task`)
logit_SA <- glm(`Completed task` ~ Experience + Training, data = SA, family = "binomial")
summary(logit_SA)
```

```
##
## Call:
## glm(formula = `Completed task` ~ Experience + Training, family = "binomial",
##     data = SA)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.21813   0.08196   0.17479   0.34959   2.65306
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  10.9813     2.8919   3.797 0.000146 ***
## Experience   -1.1269     0.2909  -3.874 0.000107 ***
## Training     -0.1805     0.3386  -0.533 0.593970
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 75.060  on 74  degrees of freedom
## Residual deviance: 35.713  on 72  degrees of freedom
## AIC: 41.713
##
```

```
## Number of Fisher Scoring iterations: 6
```

```r
SA_pred <- predict(logit_SA, data = SA, type = "response")
prob <- ifelse(SA_pred > 0.5, "1", "0" )
table(prob, SA$`Completed task`)
```

```
##
## prob  0  1
##    0 10  2
##    1  5 58
```

c. To decrease the percentage in part (b), should the cutoff probability be increased or decreased?

Answer: Cutoff = 0.8; misclassification = (1/15)*100 = 6.67% Cutoff = 0.2; misclassification = (6/15)*100 = 40% From the above value we can say that the cutoff should be increased to decrease the percentage

```r
prob <- ifelse(SA_pred > 0.9, "1", "0" )
table(prob, SA$`Completed task`)
```

```
##
## prob  0  1
##    0 14 13
##    1  1 47
```

```r
prob <- ifelse(SA_pred > 0.4, "1", "0" )
table(prob, SA$`Completed task`)
```

```
##
## prob  0  1
##    0  9  1
##    1  6 59
```

d. How much experience must be accumulated by a programmer with 4 years of training before his or her estimated probability of completing the task exceeds 50%?

Answer: probability = 0.5 x2 = 4 a = 10.9813 b1 = -1.1269 b2 = -0.1805

```
probability = 1 / (1 + e ^ (a + b1*x1 + b2*x2))
0.5 = 1 / (1 + e ^ (10.9813 + (-1.1269)*x1 +(-0.1805)*4))
x1 = approx(9 years)
```