Regression Tree & Random Forest

Victoria Okereke

```r
#importing libraries
library(faraway)
library(visdat)
library(olsrr)
```

```
##
## Attaching package: 'olsrr'

## The following object is masked from 'package:faraway':
##
##     hsb

## The following object is masked from 'package:datasets':
##
##     rivers
```

```r
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

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

```r
library(rpart)
```

```
##
## Attaching package: 'rpart'

## The following object is masked from 'package:faraway':
##
##     solder
```

```r
library(rpart.plot)
library(randomForest)
```

```
## randomForest 4.6-14

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

```r
library(caret)
```

```
## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':
```

```
##
##     margin
```

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

```
##
## Attaching package: 'lattice'
```

```
## The following object is masked from 'package:faraway':
##
##     melanoma
```

```
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

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

```
library(ipred)
#setting seed
set.seed(123)
```

Data Exploration

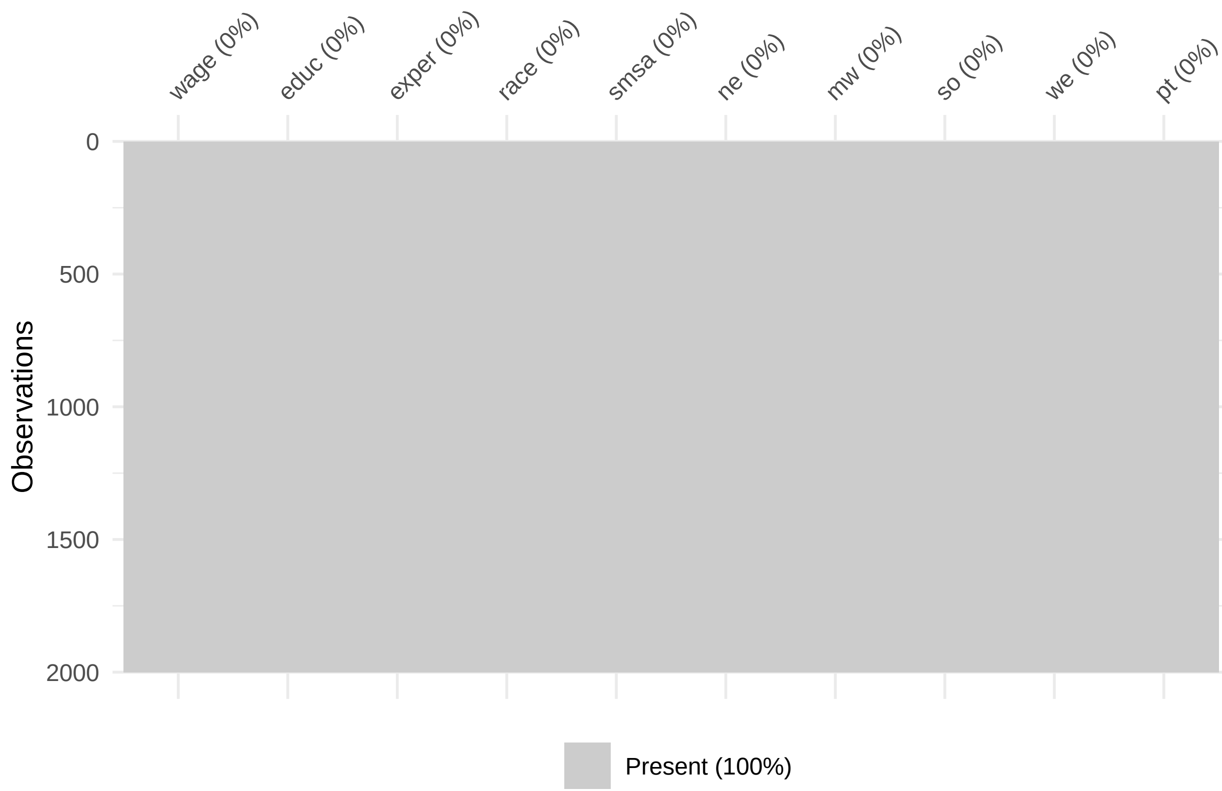
```
#reading in dataset
data("uswages")
#viewing data structure
str(uswages)
```

```
## 'data.frame':    2000 obs. of  10 variables:
##  $ wage : num  772 617 958 617 902 ...
##  $ educ : int  18 15 16 12 14 12 16 16 12 12 ...
##  $ exper: int  18 20 9 24 12 33 42 0 36 37 ...
##  $ race : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ smsa : int  1 1 1 1 1 1 1 1 1 0 ...
##  $ ne   : int  1 0 0 1 0 0 0 0 0 0 ...
##  $ mw   : int  0 0 0 0 1 0 0 1 0 1 ...
##  $ so   : int  0 0 1 0 0 0 1 0 0 0 ...
##  $ we   : int  0 1 0 0 0 1 0 0 1 0 ...
##  $ pt   : int  0 0 0 0 0 0 1 1 1 0 ...
```

```
#viewing first 6 rows of data
head(uswages)
```

```
##         wage educ exper race smsa ne mw so we pt
## 6085  771.60   18    18    0    1  1  0  0  0  0
## 23701 617.28   15    20    0    1  0  0  0  1  0
## 16208 957.83   16     9    0    1  0  0  1  0  0
## 2720  617.28   12    24    0    1  1  0  0  0  0
## 9723  902.18   14    12    0    1  0  1  0  0  0
## 22239 299.15   12    33    0    1  0  0  0  1  0
```

```
#viewing the pattern of missingness
vis_miss(uswages)
```



No missing data so we do not need to worry about missingness.

A careful review of the data shows that columns ne, mw, so, and we seem to have been coded from the same categorical variable so we will drop one of them from the model

```
#dropping the 'we' variable
uswages_reduced = uswages[-c(9)]
```

Now let us fit the regression tree model

```
set.seed(123)
#Splitting data into train and test
split_data = createDataPartition(y = uswages_reduced$wage, p = .9, list = FALSE)
train_data = uswages_reduced[split_data,]
test_data = uswages_reduced[-split_data,]
```

```
set.seed(123)
#fitting the model to the train set and setting cp to 0 to
#allow the tree to grow very deep
uswages_tree = rpart(wage ~ ., data = train_data,cp=0)
uswages_tree
```

```
## n= 1802
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##      1) root 1802 338436200.00  605.6148
##        2) educ< 15.5 1330 166713500.00  524.4594
##          4) exper< 12.5 530  41234040.00  387.4127
##            8) exper< 4.5 189   9007581.00  268.1435
```

```
##              16) pt>=0.5 57    526351.30  141.0053
##                32) exper< 2.5 48    254185.50  126.9475
##                  64) educ< 14.5 40    198735.80  118.1272
##                   128) ne>=0.5 10     10689.76   87.3500 *
##                   129) ne< 0.5 30    175416.20  128.3863
##                     258) educ< 13.5 23     70499.08  122.5457
##                       516) exper< 0.5 12     14643.67  105.4217 *
##                       517) exper>=0.5 11     48497.98  141.2264 *
##                     259) educ>=13.5 7    101554.50  147.5771 *
##                  65) educ>=14.5 8     36778.46  171.0488 *
##                33) exper>=2.5 9    212089.10  215.9800 *
##              17) pt< 0.5 132   7162016.00  323.0442
##                34) exper>=-0.5 124   3041082.00  311.3554
##                  68) exper< 1.5 44   1109861.00  256.8784
##                   136) educ< 12.5 25    410356.00  230.6924
##                     272) so< 0.5 17    167926.40  195.1276 *
##                     273) so>=0.5 8    175234.40  306.2675 *
##                   137) educ>=12.5 19    659806.10  291.3337 *
##                  69) exper>=1.5 80   1728821.00  341.3178
##                   138) ne< 0.5 58   1223543.00  311.3807
##                     276) educ< 12.5 45    694923.40  295.4447
##                       552) smsa< 0.5 13    223343.90  267.4577 *
##                       553) smsa>=0.5 32    457260.40  306.8144
##                        1106) mw< 0.5 24    223573.30  294.2208
##                          2212) so< 0.5 9     55740.14  240.1267 *
##                          2213) so>=0.5 15    125696.20  326.6773 *
##                        1107) mw>=0.5 8    218461.70  344.5950 *
##                     277) educ>=12.5 13    477632.60  366.5438 *
##                   139) ne>=0.5 22    316255.90  420.2427
##                     278) educ< 13.5 13    221529.50  383.4823 *
##                     279) educ>=13.5 9     51784.15  473.3411 *
##                35) exper< -0.5 8   3841394.00  504.2200 *
##             9) exper>=4.5 341  28047770.00  453.5179
##              18) educ< 13.5 268  22770850.00  428.5318
##                36) educ< 9.5 16    165523.70  245.1863 *
##                37) educ>=9.5 252  22033330.00  440.1728
##                  74) educ< 11.5 37   9293959.00  388.2173
##                   148) exper< 9.5 19    397684.00  283.4774 *
##                   149) exper>=9.5 18   8467818.00  498.7761 *
##                  75) educ>=11.5 215  12622300.00  449.1140
##                   150) mw< 0.5 153   9778284.00  437.7348
##                     300) exper>=5.5 139   5818653.00  430.4014
##                       600) race>=0.5 13    395350.00  298.5715 *
##                       601) race< 0.5 126   5174064.00  444.0029
##                        1202) exper< 6.5 18    232103.80  345.7117 *
##                        1203) exper>=6.5 108   4739076.00  460.3847
##                          2406) smsa< 0.5 31   1050868.00  409.1168
##                            4812) so>=0.5 16    354546.70  316.0138 *
##                            4813) so< 0.5 15    409693.30  508.4267 *
##                          2407) smsa>=0.5 77   3573925.00  481.0251
##                            4814) educ< 12.5 69   3209707.00  466.1901
##                              9628) exper< 11.5 59   2782953.00  456.5580
##                               19256) exper>=10.5 14    710274.00  443.4029 *
##                               19257) exper< 10.5 45   2069502.00  460.6507
```

```
##                                38514) ne< 0.5 28   1242912.00  447.7293
##                                  77028) so< 0.5 11    588362.80  411.3636 *
##                                  77029) so>=0.5 17    630589.70  471.2600 *
##                                38515) ne>=0.5 17    814214.90  481.9329 *
##                             9629) exper>=11.5 10    388983.60  523.0200 *
##                           4815) educ>=12.5 8    218060.70  608.9762 *
##                    301) exper< 5.5 14   3877937.00  510.5450 *
##                  151) mw>=0.5 62   2775318.00  477.1948
##                    302) smsa< 0.5 20    284362.60  408.0655
##                      604) exper< 9.5 10    156621.70  384.9340 *
##                      605) exper>=9.5 10    117039.60  431.1970 *
##                    303) smsa>=0.5 42   2349865.00  510.1136
##                      606) exper>=11.5 7    169299.90  446.5957 *
##                      607) exper< 11.5 35   2146676.00  522.8171
##                       1214) exper< 10.5 28   1761329.00  497.8439
##                         2428) exper>=7.5 14    509246.80  454.8336 *
##                         2429) exper< 7.5 14   1200285.00  540.8543 *
##                       1215) exper>=10.5 7    298034.20  622.7100 *
##             19) educ>=13.5 73   4495364.00  545.2477
##               38) exper< 7.5 24    866256.20  425.3921
##                 76) exper< 5.5 8    226325.00  317.9188 *
##                 77) exper>=5.5 16    501325.00  479.1288 *
##               39) exper>=7.5 49   3115473.00  603.9524
##                 78) exper< 11.5 40   2613651.00  579.4195
##                  156) educ< 14.5 27   1605346.00  557.4985
##                    312) exper>=10.5 10    455360.90  497.1140 *
##                    313) exper< 10.5 17   1092074.00  593.0188 *
##                  157) educ>=14.5 13    968383.30  624.9477 *
##                 79) exper>=11.5 9    370748.90  712.9878 *
##          5) exper>=12.5 800 108930300.00  615.2528
##           10) pt>=0.5 47   8027382.00  287.4338
##             20) exper>=19.5 40    807625.80  190.3500
##               40) exper>=48.5 12     12631.27  110.4117 *
##               41) exper< 48.5 28    685449.40  224.6093
##                 82) exper< 35 12     72811.90  169.8108 *
##                 83) exper>=35 16    549577.20  265.7081 *
##             21) exper< 19.5 7   4688398.00  842.1986 *
##           11) pt< 0.5 753  95536830.00  635.7143
##             22) educ< 13.5 632  76587270.00  606.8406
##               44) smsa< 0.5 183  10489500.00  505.9013
##                 88) educ< 11.5 55   2096696.00  404.5471
##                  176) educ< 5.5 7     58436.82  236.0443 *
##                  177) educ>=5.5 48   1810522.00  429.1204
##                    354) exper< 18.5 12    305883.70  326.6392 *
##                    355) exper>=18.5 36   1336600.00  463.2808
##                      710) so>=0.5 20    637577.60  402.1985
##                       1420) educ>=8.5 12    249088.20  325.9492 *
##                       1421) educ< 8.5 8    214070.50  516.5725 *
##                      711) so< 0.5 16    531124.80  539.6338 *
##                 89) educ>=11.5 128   7585037.00  549.4520
##                  178) exper< 29.5 96   5387994.00  520.7906
##                    356) exper>=27.5 8    388220.10  331.9050 *
##                    357) exper< 27.5 88   4688405.00  537.9620
##                      714) exper>=21.5 27   1440654.00  488.8315
```

```
##                        1428) so>=0.5 11    155286.00  371.8136 *
##                        1429) so< 0.5 16   1031188.00  569.2812 *
##                       715) exper< 21.5 61   3153731.00  559.7084
##                        1430) exper< 18.5 48   2559757.00  534.0869
##                          2860) exper>=13.5 37   2129121.00  509.6449
##                            5720) mw>=0.5 10    221491.70  451.2840 *
##                            5721) mw< 0.5 27   1860954.00  531.2600
##                             11442) exper>=15.5 15    547791.80  460.2167 *
##                             11443) exper< 15.5 12   1142821.00  620.0642 *
##                          2861) exper< 13.5 11    334182.00  616.3009 *
##                        1431) exper>=18.5 13    446118.00  654.3108 *
##                  179) exper>=29.5 32   1881598.00  635.4359
##                    358) exper>=37.5 13    819870.90  558.5638 *
##                    359) exper< 37.5 19    932343.80  688.0326 *
##               45) smsa>=0.5 449  63473300.00  647.9807
##                 90) educ< 8.5 68  25849720.00  510.4057
##                  180) exper>=19.5 61   3686435.00  445.7741
##                    360) educ< 4.5 17    224853.00  287.2576 *
##                    361) educ>=4.5 44   2869373.00  507.0191
##                      722) so>=0.5 16    461205.70  421.9381 *
##                      723) so< 0.5 28   2226164.00  555.6368
##                       1446) exper>=35 21   1503039.00  527.5133
##                         2892) exper< 43.5 9    597976.10  438.6156 *
##                         2893) exper>=43.5 12    780593.90  594.1867 *
##                       1447) exper< 35 7    656686.40  640.0071 *
##                  181) exper< 19.5 7  19687960.00 1073.6240 *
##                 91) educ>=8.5 381  36106850.00  672.5347
##                  182) race>=0.5 45   2752729.00  541.8447
##                    364) educ>=11.5 33   1947901.00  512.1982
##                      728) so>=0.5 16    522394.70  456.5800 *
##                      729) so< 0.5 17   1329430.00  564.5447 *
##                    365) educ< 11.5 12    696062.70  623.3725 *
##                  183) race< 0.5 336  32482590.00  690.0379
##                    366) exper< 17.5 80   5883877.00  624.3524
##                      732) educ< 12.5 68   4413448.00  601.4732
##                       1464) exper< 14.5 32   1569267.00  551.8681
##                         2928) so< 0.5 24   1130987.00  529.5021
##                           5856) exper< 13.5 12    516620.30  498.2475 *
##                           5857) exper>=13.5 12    590922.70  560.7567 *
##                         2929) so>=0.5 8    390256.60  618.9662 *
##                       1465) exper>=14.5 36   2695447.00  645.5667
##                         2930) mw>=0.5 13    423989.60  590.6277 *
##                         2931) mw< 0.5 23   2210042.00  676.6191
##                           5862) exper>=15.5 16    857844.80  655.0944 *
##                           5863) exper< 15.5 7   1327840.00  725.8186 *
##                      733) educ>=12.5 12   1233130.00  754.0008 *
##                    367) exper>=17.5 256  26145680.00  710.5646
##                      734) exper>=45.5 8    232567.70  522.7600 *
##                      735) exper< 45.5 248  25621850.00  716.6228
##                       1470) educ< 10.5 30   3130902.00  641.0653
##                         2940) exper< 32 12    454465.80  506.7883 *
##                         2941) exper>=32 18   2315830.00  730.5833 *
##                       1471) educ>=10.5 218  22296110.00  727.0206
##                         2942) exper>=36.5 62   5803775.00  690.7008
```

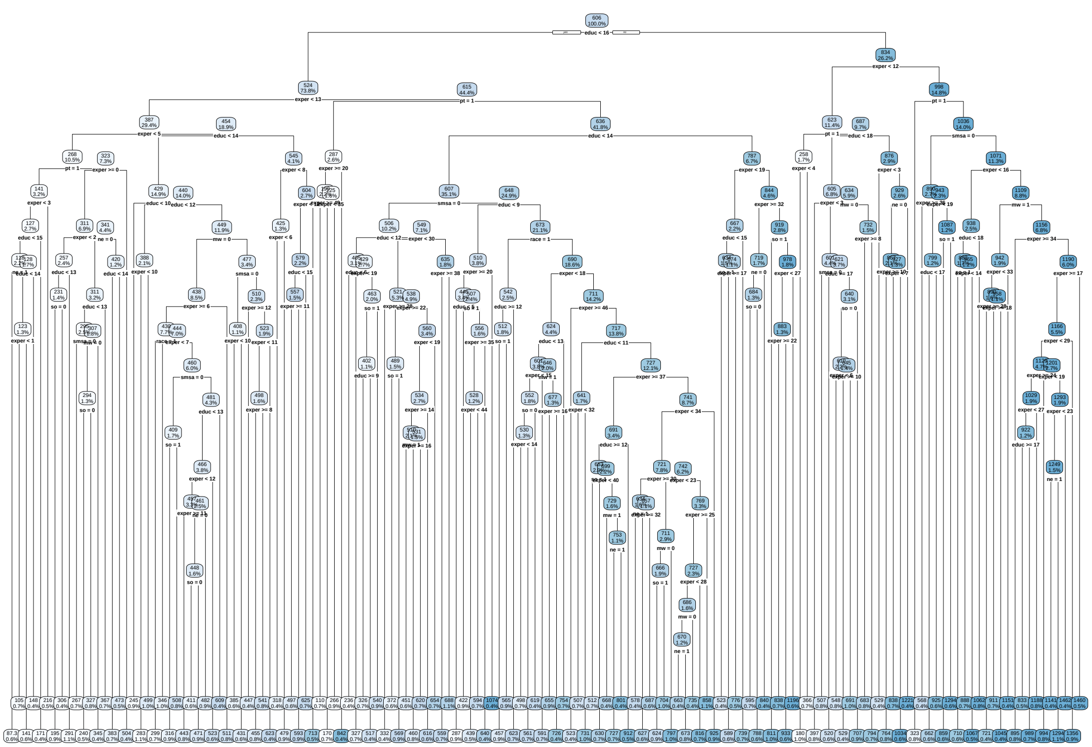
```
##                      5884) educ>=11.5 53   4143602.00  653.1075  
##                       11768) so>=0.5 13    444626.50  512.0238 *
##                       11769) so< 0.5 40   3356119.00  698.9597  
##                         23538) exper< 39.5 12    916239.30  629.6075 *
##                         23539) exper>=39.5 28   2357427.00  728.6821  
##                           47078) mw>=0.5 8   1057326.00  668.1175 *
##                           47079) mw< 0.5 20   1259018.00  752.9080  
##                             94158) ne>=0.5 13    461653.80  727.1162 *
##                             94159) ne< 0.5 7    772656.30  800.8071 *
##                      5885) educ< 11.5 9   1144178.00  912.0833 *
##                    2943) exper< 36.5 156  16378040.00  741.4554  
##                      5886) exper< 33.5 140  12516600.00  720.5238  
##                       11772) exper>=29.5 28   1615053.00  634.2496  
##                         23544) ne>=0.5 8    616152.00  577.6675 *
##                         23545) ne< 0.5 20    963044.30  656.8825  
##                           47090) exper>=31.5 10    445579.60  626.8170 *
##                           47091) exper< 31.5 10    499386.00  686.9480 *
##                       11773) exper< 29.5 112  10641030.00  742.0923  
##                         23546) exper< 22.5 52   3879442.00  711.4915  
##                           47092) mw< 0.5 34   1676813.00  666.4285  
##                             94184) so>=0.5 16   1047747.00  624.0694 *
##                             94185) so< 0.5 18    574838.00  704.0811 *
##                           47093) mw>=0.5 18   2003172.00  796.6106 *
##                         23547) exper>=22.5 60   6670695.00  768.6130  
##                           47094) exper>=24.5 41   2769666.00  727.3641  
##                             94188) exper< 27.5 28   1812094.00  686.0854  
##                              188376) mw< 0.5 21   1512982.00  669.7890  
##                                376752) ne>=0.5 7    640940.80  662.8014 *
##                                376753) ne< 0.5 14    871528.20  673.2829 *
##                              188377) mw>=0.5 7    276804.50  734.9743 *
##                             94189) exper>=27.5 13    807101.20  816.2723 *
##                           47095) exper< 24.5 19   3680734.00  857.6237 *
##                      5887) exper>=33.5 16   3263394.00  924.6069 *
##         23) educ>=13.5 121  15670640.00  786.5255  
##           46) exper< 18.5 39   1895673.00  666.6341  
##             92) educ< 14.5 27   1136233.00  634.4822  
##              184) so>=0.5 7    219001.40  522.9957 *
##              185) so< 0.5 20    799775.50  673.5025  
##                370) exper>=16.5 11    323136.90  589.3973 *
##                371) exper< 16.5 9    303726.20  776.2978 *
##             93) educ>=14.5 12    668728.90  738.9758 *
##           47) exper>=18.5 82  12947760.00  843.5470  
##             94) exper>=31.5 31   4708959.00  719.0210  
##              188) ne< 0.5 24   3165886.00  683.7317  
##                376) so< 0.5 13   1937580.00  595.3708 *
##                377) so>=0.5 11   1006853.00  788.1582 *
##              189) ne>=0.5 7   1410711.00  840.0129 *
##             95) exper< 31.5 51   7465900.00  919.2392  
##              190) so>=0.5 18   1707540.00  811.4583 *
##              191) so< 0.5 33   5435204.00  978.0288  
##                382) exper< 26.5 23   2141420.00  883.3191  
##                  764) exper>=21.5 12    516898.00  838.0192 *
##                  765) exper< 21.5 11   1573033.00  932.7373 *
##                383) exper>=26.5 10   2612967.00 1195.8610 *
```

```
##         3) educ>=15.5 472 138280100.00  834.2942
##           6) exper< 11.5 206  34775030.00  622.5582
##            12) pt>=0.5 31    905411.20  257.9142
##              24) exper< 3.5 18    114124.80  179.6611 *
##              25) exper>=3.5 13    528445.10  366.2646 *
##            13) pt< 0.5 175  29017520.00  687.1523
##              26) educ< 17.5 122  12400530.00  605.0953
##                52) exper< 2.5 15    569694.40  396.5447 *
##                53) exper>=2.5 107  11086980.00  634.3314
##                 106) mw< 0.5 80   6449157.00  601.3760
##                   212) smsa< 0.5 14    861053.80  506.6386 *
##                   213) smsa>=0.5 66   5435797.00  621.4718
##                     426) educ>=16.5 10    842370.70  520.3660 *
##                     427) educ< 16.5 56   4472948.00  639.5264
##                       854) so< 0.5 39   3634163.00  610.3246
##                        1708) exper< 5.5 14    670097.20  547.6421 *
##                        1709) exper>=5.5 25   2878254.00  645.4268
##                          3418) exper>=9.5 7    420346.90  528.9857 *
##                          3419) exper< 9.5 18   2326089.00  690.7094 *
##                       855) so>=0.5 17    729232.40  706.5188 *
##                 107) mw>=0.5 27   4293500.00  731.9770
##                   214) exper>=7.5 15   2193886.00  682.7053 *
##                   215) exper< 7.5 12   2017679.00  793.5667 *
##              27) educ>=17.5 53  13904600.00  876.0381
##                54) exper< 2.5 7    237706.90  528.9214 *
##                55) exper>=2.5 46  12695110.00  928.8602
##                 110) ne< 0.5 38   7859479.00  867.3453
##                   220) exper>=9.5 14   1983350.00  764.2714 *
##                   221) exper< 9.5 24   5640626.00  927.4717
##                     442) exper< 6.5 13   2181224.00  837.5531 *
##                     443) exper>=6.5 11   3230072.00 1033.7390 *
##                 111) ne>=0.5 8   4008812.00 1221.0560 *
##           7) exper>=11.5 266  87117430.00  998.2703
##            14) pt>=0.5 14    715871.50  323.1150 *
##            15) pt< 0.5 252  79665330.00 1035.7790
##              30) smsa< 0.5 49  12337970.00  889.6080
##                60) exper>=35.5 7    312940.80  568.0129 *
##                61) exper< 35.5 42  11180400.00  943.2071
##                 122) exper< 18.5 21   2155056.00  799.4105
##                   244) educ< 16.5 10    310523.20  661.6800 *
##                   245) educ>=16.5 11   1482385.00  924.6200 *
##                 123) exper>=18.5 21   8156889.00 1087.0040
##                   246) so>=0.5 10   1102699.00  859.3100 *
##                   247) so< 0.5 11   6064432.00 1293.9980 *
##              31) smsa>=0.5 203  66027730.00 1071.0620
##                62) exper< 15.5 45  11424590.00  938.3060
##                 124) educ< 17.5 31   8264148.00  882.4377
##                   248) so>=0.5 10    861716.00  709.6080 *
##                   249) so< 0.5 21   6961492.00  964.7376
##                     498) exper< 13.5 12   3463232.00  887.8033 *
##                     499) exper>=13.5 9   3332530.00 1067.3170 *
##                 125) educ>=17.5 14   2849429.00 1062.0140 *
##                63) exper>=15.5 158  53584180.00 1108.8720
##                 126) mw>=0.5 35   4103320.00  942.2451
```

```
##                    252) exper< 32.5 28   2985021.00  890.1539
##                      504) exper>=27.5 8    552250.60  721.3238 *
##                      505) exper< 27.5 20   2113530.00  957.6860
##                       1010) exper>=17.5 13   1294837.00  910.8677 *
##                       1011) exper< 17.5 7    737277.50 1044.6340 *
##                    253) exper>=32.5 7    738409.20 1150.6100 *
##                  127) mw< 0.5 123  48232590.00 1156.2860
##                    254) exper>=33.5 14   2169765.00  894.8921 *
##                    255) exper< 33.5 109  44983390.00 1189.8590
##                      510) exper>=16.5 100  32300520.00 1165.5040
##                       1020) exper< 28.5 84  27787720.00 1129.1880
##                         2040) exper>=23.5 35   8876777.00 1028.5600
##                           4080) exper< 26.5 21   5657526.00  922.1319
##                             8160) educ>=16.5 9   3589838.00  832.6578 *
##                             8161) educ< 16.5 12   1941600.00  989.2375 *
##                           4081) exper>=26.5 14   2624589.00 1188.2010 *
##                         2041) exper< 23.5 49  18303390.00 1201.0650
##                           4082) exper< 18.5 15   4869715.00  993.6787 *
##                           4083) exper>=18.5 34  12503920.00 1292.5590
##                             8166) exper< 22.5 27  10572740.00 1248.5240
##                              16332) ne>=0.5 8   2526504.00 1140.5850 *
##                              16333) ne< 0.5 19   7913782.00 1293.9720 *
##                             8167) exper>=22.5 7   1676877.00 1462.4100 *
##                       1021) exper>=28.5 16   3820393.00 1356.1640 *
##                      511) exper< 16.5 9  11964470.00 1460.4720 *
```

```r
#plotting the tree
rpart.plot(uswages_tree, digits = 3)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



Now let's measure the performance of the model

First let's measure the in-sample performance

```r
#predicting on the train set
uswages_train_pred = predict(uswages_tree, train_data)
```

```r
#Measuring performance on the train set with the mean absolute error
MAE_train = mean(abs(train_data$wage - uswages_train_pred))
MAE_train
```

```
## [1] 215.4881
```

We have a Mean Absolute Error of 215.4881 in-sample

Now let's measure the out-of-sample MAE

```r
#predicting on the test set
uswages_test_pred = predict(uswages_tree, test_data)

#Measuring performance on the test set with the mean absolute error
uswages_tree_MAE = mean(abs(test_data$wage - uswages_test_pred))
uswages_tree_MAE
```

```
## [1] 302.0125
```

The MAE of the model on the test data is 302.0125. There is a huge difference between the in-sample and out-of-sample performance. We are most likely overfitting to the training set.

Let's see if we can improve the performance of the model by pruning the tree.

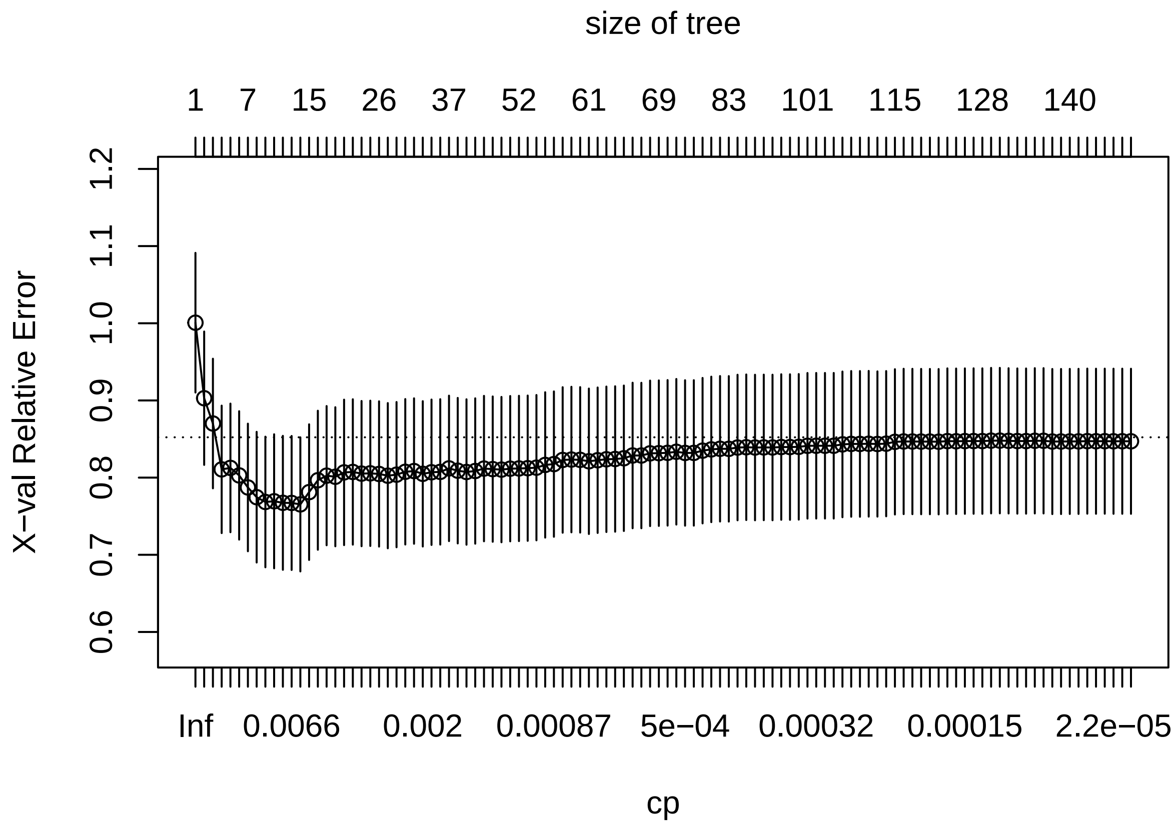
```r
#to decide where to prune the tree
printcp(uswages_tree)
```

```
##
## Regression tree:
## rpart(formula = wage ~ ., data = train_data, cp = 0)
##
## Variables actually used in tree construction:
## [1] educ  exper mw    ne    pt    race  smsa  so
##
## Root node error: 338436161/1802 = 187811
##
## n= 1802
##
##              CP nsplit rel error  xerror     xstd
## 1   9.8815e-02      0   1.00000 1.00073 0.090499
## 2   4.8899e-02      1   0.90119 0.90290 0.086430
## 3   4.8422e-02      2   0.85229 0.87011 0.083983
## 4   1.9904e-02      3   0.80386 0.81070 0.082589
## 5   1.5856e-02      4   0.78396 0.81259 0.083358
## 6   1.4337e-02      5   0.76810 0.80289 0.083218
## 7   1.2347e-02      6   0.75377 0.78726 0.082680
## 8   9.6884e-03      7   0.74142 0.77466 0.084670
## 9   8.0145e-03      8   0.73173 0.76846 0.084641
## 10  7.7547e-03      9   0.72372 0.76935 0.086851
## 11  7.4796e-03     10   0.71596 0.76734 0.086831
## 12  5.8978e-03     11   0.70848 0.76717 0.086810
## 13  3.8980e-03     13   0.69669 0.76535 0.086897
```

```
## 14  3.8401e-03     14   0.69279 0.78121 0.087902
## 15  3.3496e-03     15   0.68895 0.79661 0.090094
## 16  3.1895e-03     17   0.68225 0.80254 0.090202
## 17  2.8714e-03     18   0.67906 0.80112 0.090134
## 18  2.6621e-03     19   0.67619 0.80677 0.094262
## 19  2.5752e-03     22   0.66820 0.80735 0.094216
## 20  2.4442e-03     23   0.66563 0.80521 0.094116
## 21  2.4431e-03     24   0.66318 0.80558 0.094168
## 22  2.3868e-03     25   0.66074 0.80476 0.094151
## 23  2.3093e-03     26   0.65835 0.80251 0.094118
## 24  2.2838e-03     27   0.65605 0.80386 0.094111
## 25  2.1979e-03     28   0.65376 0.80755 0.094252
## 26  2.1778e-03     29   0.65156 0.80852 0.094254
## 27  1.7571e-03     33   0.64285 0.80491 0.094179
## 28  1.7498e-03     34   0.64110 0.80697 0.094207
## 29  1.6901e-03     35   0.63935 0.80743 0.094211
## 30  1.5177e-03     36   0.63766 0.81198 0.094260
## 31  1.4833e-03     37   0.63614 0.80901 0.094280
## 32  1.3386e-03     39   0.63317 0.80729 0.094251
## 33  1.1225e-03     40   0.63183 0.80856 0.094197
## 34  1.1109e-03     41   0.63071 0.81172 0.094227
## 35  1.0701e-03     43   0.62849 0.81102 0.094169
## 36  1.0219e-03     44   0.62742 0.81039 0.094179
## 37  1.0174e-03     50   0.62129 0.81159 0.094154
## 38  1.0131e-03     51   0.62027 0.81189 0.094156
## 39  9.4328e-04     52   0.61926 0.81223 0.094156
## 40  9.3207e-04     53   0.61831 0.81290 0.094169
## 41  9.2002e-04     54   0.61738 0.81645 0.094210
## 42  8.2597e-04     55   0.61646 0.81752 0.094204
## 43  8.0595e-04     56   0.61564 0.82289 0.094307
## 44  7.7664e-04     58   0.61402 0.82340 0.094285
## 45  7.6976e-04     59   0.61325 0.82301 0.094265
## 46  7.5140e-04     60   0.61248 0.82118 0.094238
## 47  7.0117e-04     61   0.61173 0.82254 0.094250
## 48  6.9586e-04     62   0.61102 0.82377 0.094247
## 49  6.7762e-04     63   0.61033 0.82414 0.094252
## 50  6.7291e-04     64   0.60965 0.82508 0.094252
## 51  5.9805e-04     65   0.60898 0.82870 0.094298
## 52  5.5852e-04     66   0.60838 0.82864 0.094255
## 53  5.3778e-04     67   0.60782 0.83146 0.094303
## 54  5.2272e-04     68   0.60728 0.83170 0.094305
## 55  5.1442e-04     70   0.60624 0.83210 0.094307
## 56  5.0295e-04     72   0.60521 0.83351 0.094337
## 57  5.0266e-04     75   0.60370 0.83198 0.094260
## 58  4.8969e-04     78   0.60219 0.83203 0.094250
## 59  4.5003e-04     79   0.60170 0.83493 0.094262
## 60  4.4461e-04     80   0.60125 0.83657 0.094265
## 61  4.3947e-04     81   0.60081 0.83738 0.094270
## 62  4.3688e-04     82   0.60037 0.83730 0.094271
## 63  4.0955e-04     83   0.59993 0.83899 0.094317
## 64  3.9943e-04     84   0.59952 0.83936 0.094318
## 65  3.9136e-04     92   0.59571 0.83902 0.094315
## 66  3.8729e-04     93   0.59532 0.83915 0.094314
## 67  3.8230e-04     94   0.59493 0.83905 0.094315
```

```
## 68  3.7533e-04     95   0.59455 0.83964 0.094313
## 69  3.7256e-04     98   0.59342 0.83960 0.094314
## 70  3.5598e-04     99   0.59305 0.83989 0.094314
## 71  3.2370e-04    100   0.59269 0.84122 0.094339
## 72  3.2368e-04    101   0.59237 0.84130 0.094339
## 73  3.2152e-04    102   0.59205 0.84131 0.094339
## 74  3.2138e-04    104   0.59140 0.84131 0.094339
## 75  3.0874e-04    105   0.59108 0.84307 0.094417
## 76  2.8388e-04    108   0.59016 0.84369 0.094424
## 77  2.8204e-04    109   0.58987 0.84366 0.094415
## 78  2.4363e-04    111   0.58931 0.84388 0.094417
## 79  2.4210e-04    112   0.58906 0.84358 0.094110
## 80  2.4056e-04    113   0.58882 0.84398 0.094118
## 81  1.8633e-04    114   0.58858 0.84634 0.094168
## 82  1.8147e-04    115   0.58839 0.84688 0.094174
## 83  1.7906e-04    116   0.58821 0.84681 0.094174
## 84  1.7751e-04    118   0.58786 0.84671 0.094174
## 85  1.6023e-04    119   0.58768 0.84670 0.094174
## 86  1.5792e-04    120   0.58752 0.84654 0.094174
## 87  1.5305e-04    122   0.58720 0.84731 0.094265
## 88  1.5214e-04    123   0.58705 0.84710 0.094265
## 89  1.5065e-04    124   0.58690 0.84730 0.094265
## 90  1.4454e-04    125   0.58675 0.84739 0.094266
## 91  1.4190e-04    127   0.58646 0.84737 0.094266
## 92  1.2688e-04    128   0.58631 0.84791 0.094280
## 93  1.2139e-04    129   0.58619 0.84795 0.094280
## 94  1.1160e-04    130   0.58607 0.84765 0.094123
## 95  1.0595e-04    131   0.58595 0.84740 0.094123
## 96  7.3007e-05    132   0.58585 0.84735 0.094107
## 97  7.1970e-05    133   0.58578 0.84772 0.094134
## 98  7.0601e-05    134   0.58570 0.84772 0.094134
## 99  6.9273e-05    137   0.58549 0.84675 0.094026
## 100 6.5915e-05    138   0.58542 0.84679 0.094026
## 101 5.5169e-05    139   0.58536 0.84690 0.094026
## 102 5.3418e-05    140   0.58530 0.84698 0.094026
## 103 3.8916e-05    141   0.58525 0.84729 0.094032
## 104 3.7318e-05    144   0.58513 0.84715 0.094032
## 105 3.1620e-05    145   0.58509 0.84721 0.094034
## 106 1.5838e-05    146   0.58506 0.84712 0.094034
## 107 1.5149e-06    148   0.58503 0.84715 0.094034
## 108 0.0000e+00    149   0.58503 0.84708 0.094034
```

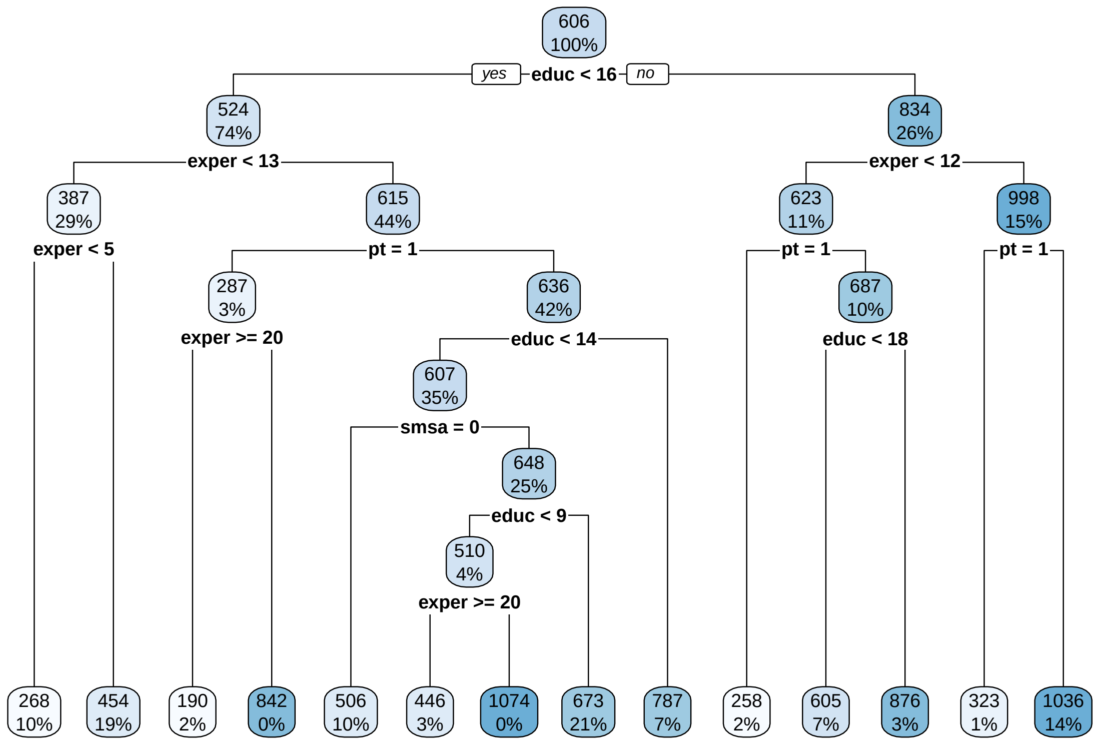
```
plotcp(uswages_tree)
```



We have the lowest xerror of (0.77175) at cp = 0.0038980

Let's prune the tree with this cp value

```
pruned_tree = prune(uswages_tree,cp=0.0038980)
rpart.plot(pruned_tree)
```




```r
#predicting on the test set
pruned_test_pred = predict(pruned_tree, test_data)

#Measuring performance on the test set with the mean absolute error
pruned_tree_MAE = mean(abs(test_data$wage - pruned_test_pred))
pruned_tree_MAE
```

```
## [1] 284.2735
```

The Mean Absolute Error out of sample is 284.2735, which is lower and better than the un-pruned tree.

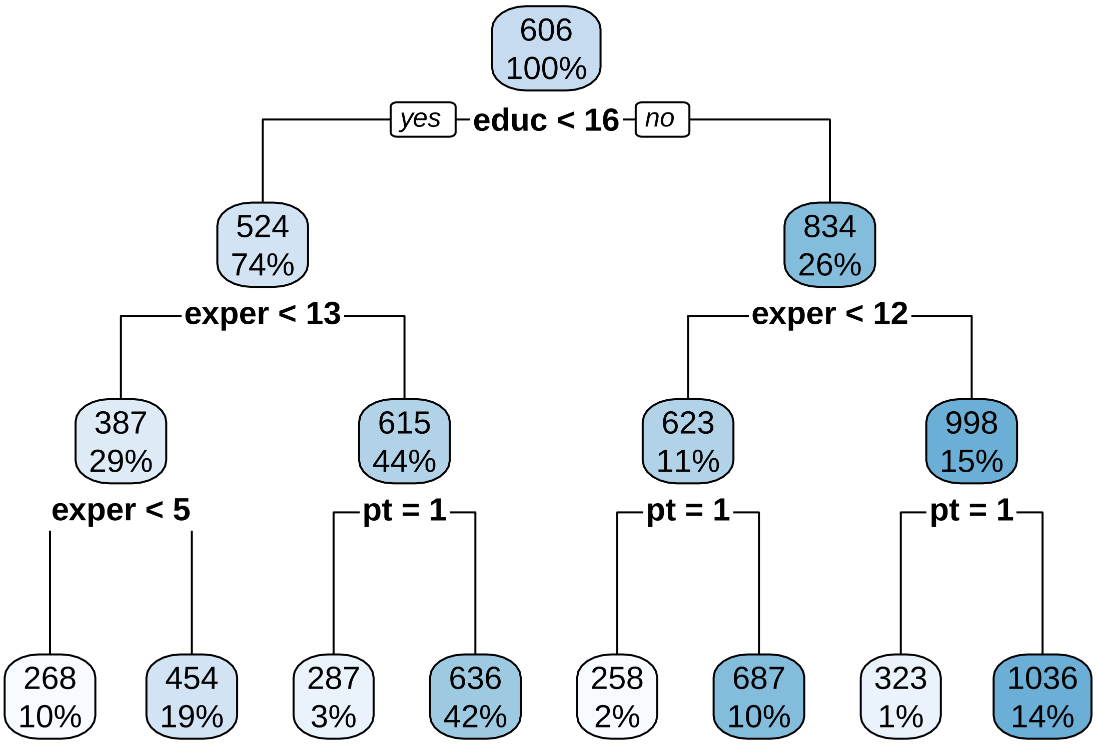
Note that rpart function in R automatically prunes the tree. Let's see how well the model will perform if we use rpart function without making any modifications to the cp value

```r
rpart_tree = rpart(wage ~ ., data = train_data)
rpart_tree
```

```
## n= 1802
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 1802 338436200.0  605.6148
##    2) educ< 15.5 1330 166713500.0  524.4594
##      4) exper< 12.5 530  41234040.0  387.4127
##        8) exper< 4.5 189   9007581.0  268.1435 *
##        9) exper>=4.5 341  28047770.0  453.5179 *
##      5) exper>=12.5 800 108930300.0  615.2528
##       10) pt>=0.5 47   8027382.0  287.4338 *
##       11) pt< 0.5 753  95536830.0  635.7143 *
##    3) educ>=15.5 472 138280100.0  834.2942
##      6) exper< 11.5 206  34775030.0  622.5582
```

```
##       12) pt>=0.5 31    905411.2  257.9142 *
##       13) pt< 0.5 175  29017520.0  687.1523 *
##      7) exper>=11.5 266  87117430.0  998.2703
##       14) pt>=0.5 14    715871.5  323.1150 *
##       15) pt< 0.5 252  79665330.0 1035.7790 *
```

```
rpart.plot(rpart_tree)
```



From the tree plot above, we see that years of education plays a very important role in predicting wages. Individuals with less years of education have smaller wage than individuals with higher years of education. Individuals with lower years of experience also have smaller wage. We also see that individuals working part-time make lower wages compared to the full-time workers.

```
#predicting on the test set
rpart_tree_pred = predict(rpart_tree, test_data)

#Measuring performance on the test set with the mean absolute error
rpart_MAE = mean(abs(test_data$wage - rpart_tree_pred))
rpart_MAE
```

```
## [1] 277.2074
```

We have an even lower MAE of 277.2074

Let's use Random Forest to predict wages

```
set.seed(123)
#fitting the model on the train dataset
uswages_rf = randomForest(wage ~ ., data = train_data, mtry = 3,
                          importance = TRUE)
uswages_rf
```

```
##
```

Table 1: Comparing the 2 models

Random Forest	274.034
Regression Tree	277.207

```
## Call:
##  randomForest(formula = wage ~ ., data = train_data, mtry = 3,      importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 140633.2
##                     % Var explained: 25.12
```

```
#predict wage on the test dataset
pred_rf = predict(uswages_rf,test_data)
#Measuring performance on the test set with mean absolute error
MAE_rf = mean(abs(test_data$wage - pred_rf))
MAE_rf
```

```
## [1] 274.0344
```

The % Var explained (Rsquared) is 25.12, which is the variance explained in the out-of-bag sample. MAE for the test set is 274.0344. Comparing the Random Forest results with the Regression Tree result, we see that the Random Forest outperformed the Regression Tree.

```
rf_result = caret::MAE(pred_rf, test_data$wage)
rt_result = caret::MAE(rpart_tree_pred, test_data$wage)

library(kableExtra)

table_data <- rbind(rf_result,rt_result)
rownames(table_data) <- c("Random Forest", "Regression Tree")
kable(table_data, digits = 3, align = "c", booktabs = TRUE,
      caption = "Comparing the 2 models")
```