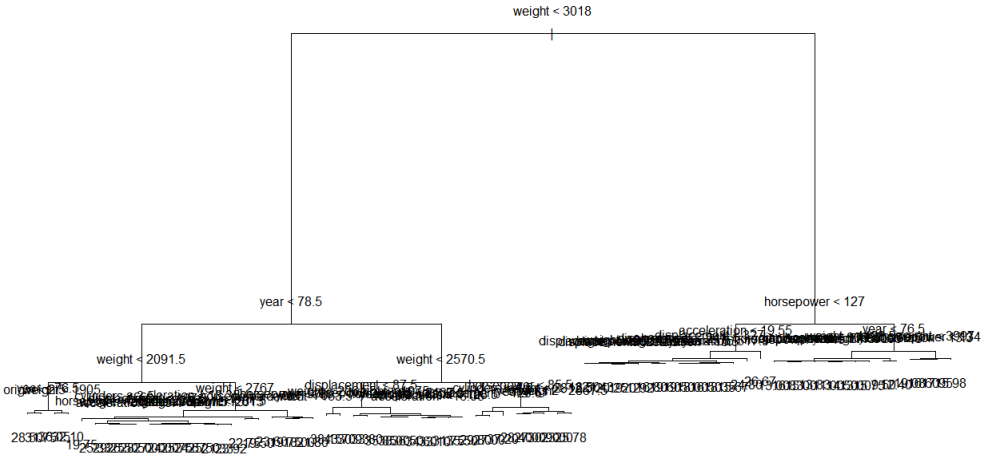
Regression Trees on Auto Dataset:

After loading the required libraries, the Auto dataset and splitting the data into a training set, I initially generated the tree to predict MPG from the entire dataset except the name variable. The tree generated as below and R used everything except name to generate the tree. Below is the summary and the tree: The tree looks messy because it has a lot of overlapping names.

```
Regression tree:
tree(formula = mpg ~ . - name, data = Auto, subset = train, control = tree.control(196,
    mincut = 2, minsize = 4, mindev = 1e-04))
Number of terminal nodes:  69
Residual mean deviance:  1.395 = 177.2 / 127
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.900  -0.500   0.000   0.000   0.500   4.033
```

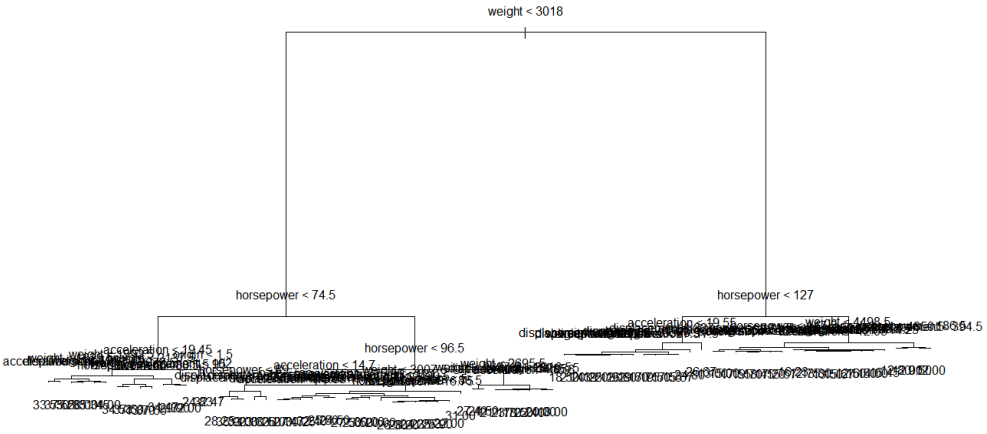
In the context of the regression tree, the deviance is simply the sum of squared errors for the tree.



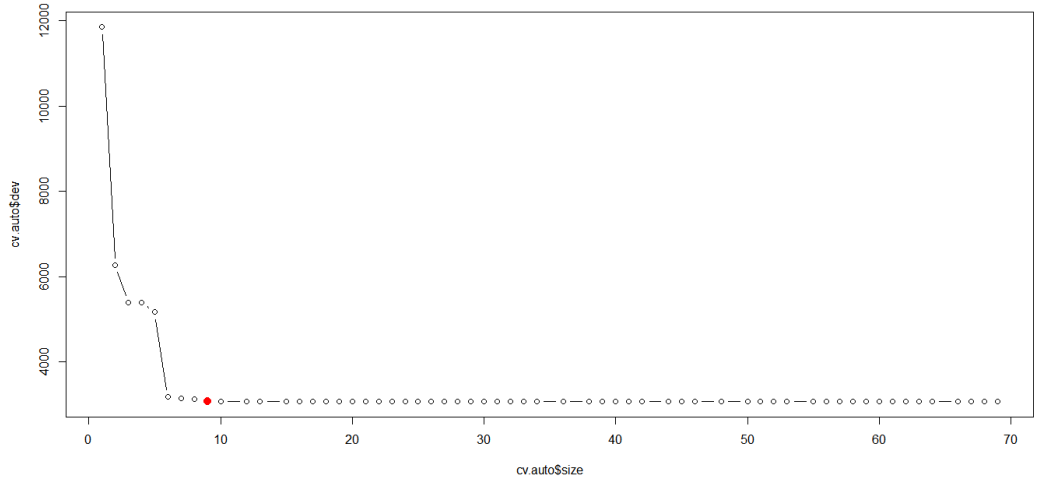
I then removed the variable year from the tree since I thought the split on the variable year was a little odd. This time the tree generated used acceleration, weight and horsepower and created a tree.

```
Regression tree:
tree(formula = mpg ~ . - name - year, data = Auto, subset = train,
    control = tree.control(196, mincut = 2, minsize = 4, mindev = 1e-04))
Number of terminal nodes:  78
Residual mean deviance:  4.816 = 568.3 / 118
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -7.0    -0.8     0.0     0.0     0.8     7.0
```

There was a reduction in the number of terminal nodes although there was an increase in the residual mean deviance. Below is the tree plot and the explanation:



We can see that the first split was on the weight. Here we observe that cars having lower weight (lower than 3018) had better mileage than cars which were heavier than 3018. There are 79 terminal nodes for the above tree. So we need to prune the tree to make it more interpretable.

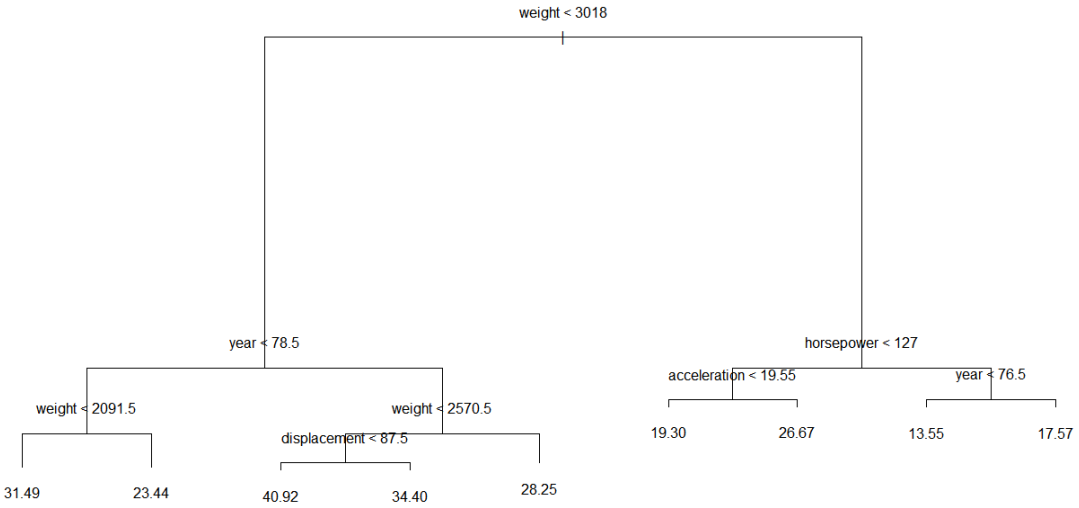


This is cross validation plot for the first tree (which contains year also as a predictor). We see that 9 is a good number to prune the tree. So I have pruned the tree at 9.

Below is the pruned tree results:

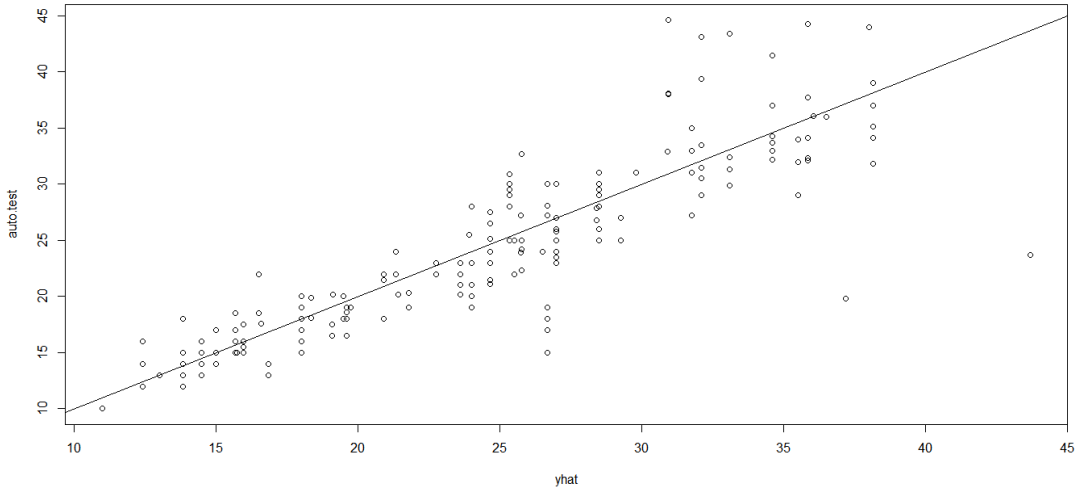
```
Regression tree:
snip.tree(tree = tree.auto, nodes = c(15L, 20L, 21L, 8L, 14L,
12L, 11L, 9L))
Variables actually used in tree construction:
[1] "weight"       "year"         "displacement" "horsepower"   "acceleration"
Number of terminal nodes:  9
Residual mean deviance:  7.37 = 1378 / 187
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-6.25500 -1.72800 -0.05152  0.00000  1.55000  9.74500
```

The pruned tree uses weight, horsepower, displacement, year and acceleration and makes the tree more interpretable.



To test the prediction accuracy, we will be using the unpruned tree.

Below is the plot of prediction:




```
> yhat=predict (tree.auto ,newdata =Auto [-train ,])
> auto.test=Auto [-train ,"mpg"]
> plot(yhat ,auto.test)
> abline (0,1)
> mean((yhat -auto.test)^2)
[1] 14.09672
```

The mean squared error is 14.09672 and hence the square root of this is 3.75 which indicates that the test predictions are within 3.75 of the true mileage of the car.

R-code:

```
library(tree)
library(ISLR)
library(MASS)
attach(Auto)
Auto=Auto
set.seed(1)

train = sample (1: nrow(Auto), nrow(Auto)/2)

#All except name
tree.auto =tree(mpg~.-name,Auto,control=tree.control(196, mincut = 2, minsize = 4, mindev = 0.0001),subset =train)
summary (tree.auto)
plot(tree.auto)
text(tree.auto, pretty = 0)
```

```
#All except name and year

tree.auto1 =tree(mpg~.-name-year,Auto,control=tree.control(196, mincut = 2, minsize = 4, mindev = 0.0001),subset
=train)

summary (tree.auto1)

plot(tree.auto1)

text(tree.auto1, pretty = 0)


#Cross validation test

cv.auto =cv.tree(tree.auto)

plot(cv.auto$size ,cv.auto$dev ,type='b')

tree.min = which.min(cv.auto$dev)

points(cv.auto$size[tree.min], cv.auto$dev[tree.min], col = "red", cex = 2, pch = 20)



#Prune tree

prune.auto=prune.tree(tree.auto, best=9)

summary(prune.auto)

plot(prune.auto)

text(prune.auto, pretty = 0)


#Mean squared error

yhat=predict (tree.auto ,newdata =Auto [-train ,])

auto.test=Auto [-train ,"mpg"]

plot(yhat ,auto.test)

abline (0,1)

mean((yhat -auto.test)^2)
```