Research Goal and Data:

This research aims to investigate what types of factors can be used to estimate the miles per gallon of specific vehicles. I originally planned to use variables such as number of cylinders between 4 and 8 (cylinders), engine displacement in cu. inches (displacement), engine horsepower (horsepower), vehicle weight in pounds (weight), time to accelerate from 0 to 60 miles per hour in seconds (acceleration), car model year in modulo 100 (year), whether the car was made in 1. America, 2. Europe, or 3. Japan (origin), and the name of the car (name). However, I felt that the name of the car was applicable to my goals. The original data set contained 408 observations, however, 16 of those observations had missing values and were removed. A higher mile per gallon is viewed to be more desirable for the environment and economy and this study plans to uncover what types of variable may be related to higher miles per gallon. The data used in this study is Auto in the ISLR library in R.

Response Variable:

The response variable is miles per gallon (mpg). This is a quantitative (numeric) variable.

Predictor Variable:

The predictor variables for this study are number of cylinders, engine displacement, horsepower, weight, acceleration time from 0 to 60 mph, car model year, origin of the car. Though the brand and style of car is a recorded variable, I chose to omit it for this study.

Statistical Methods:

This study used four statistical methods to predict the miles per gallon of these vehicles. I also used three subsets of the boosted regression tree method. One was the default method, then I specified the shrinkage at two different values. All of these methods required a training and test

set to obtain predicted values for miles per gallon. Table 1 shows the mean square error, or MSE, for each of the different methods. The MSE is used to quantify the quality of the predictions. The methods used are all appropriate for a quantitative response variable such as miles per gallon, and used seven predictor variables.

Table 1: *This table shows the mean squared error for the statistical methods used. Again, the lower the MSE the more appropriate the model is for prediction.*

Method	MSE
Linear Regression Model	11.3998
Regression Tree	10.952
Boosted Regression Tree	7.211634
Boosted Regression Tree (Lambda 0.02)	9.698558
Boosted Regression Tree (Lambda 0.01)	9.160304
Random Forest	7.511008

Table 2: This table shows the results from a multiple linear regression fit.

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	-17.218435	4.644294	-3.707	0.00024	***
cylinders	-0.493376	0.323282	-1.526	0.12780	
displacement	0.019896	0.007515	2.647	0.00844	**
horsepower	-0.016951	0.013787	-1.230	0.21963	
weight	-0.006474	0.000652	-9.929	< 2e-16	***
acceleration	0.080576	0.098845	0.815	0.41548	
year	0.750773	0.050973	14.729	< 2e-16	***
origin	1.426141	0.278136	5.127	4.67e-07	***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Figure 1: *Regression tree, which shows the splits and predictions of miles per gallon dependent upon the predictor variables.*

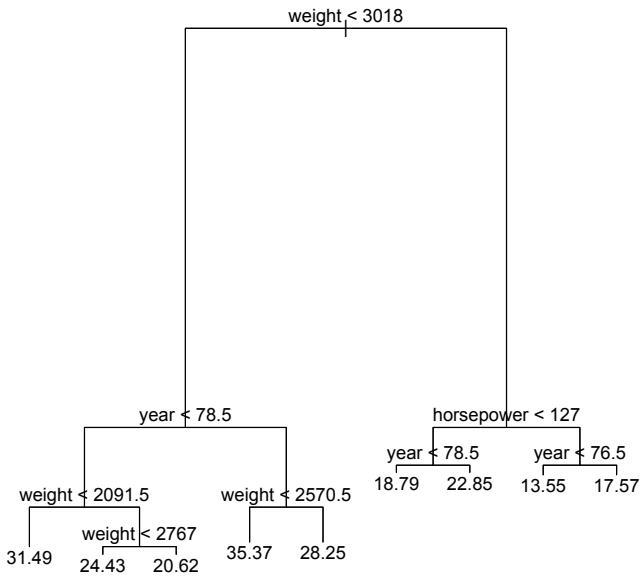
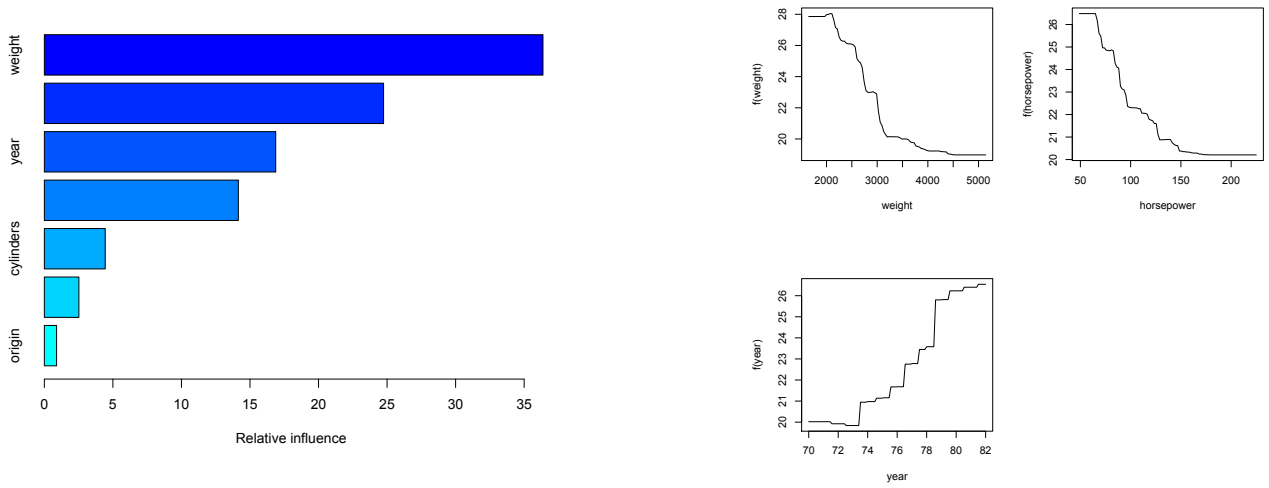


Figure 2: (Left) *The variable importance plot from the boosted tree.*

Figure 3: (Right two panels) *Partial dependence plots for Weight and Horsepower from the boosted tree method, which were found to be the two most influential variables.*



Interpretation of Results:

We can see from out results that these methods can be used to predict miles per gallon using our predictor variables. From figure 3, we can see the relationship between horsepower, weight, and mpg. As weight and horsepower increase, the miles per gallons decreases rapidly. In looking at our methods further, we can also notice that the age of a vehicle does generally have an influence on the miles per gallon too. It can be observed that newer cars have a higher mpg.

In the review of our mean squared error we can see that for our purposes the default boosted regression tree would be the preferred model due to the lower value of 7.2116. However, there is not much of a difference between the boosted regression tree and the random forest method. For the purposes of this study, the linear regression model yields the highest MSE and would thus be least preferred.

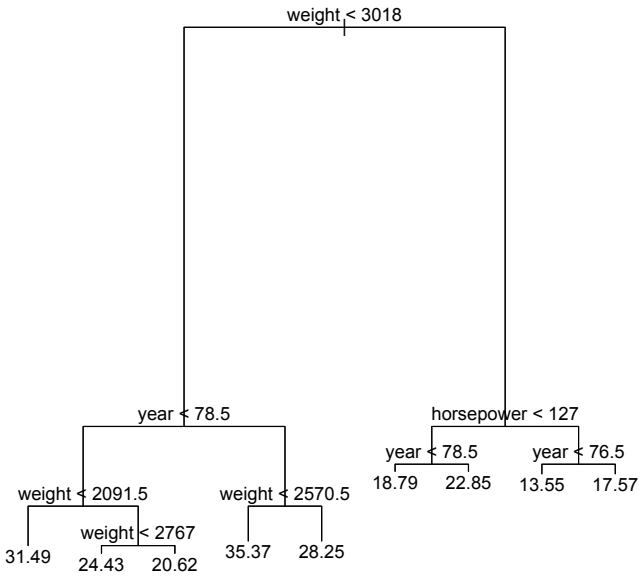
In future studies, multiple data set splits may provide more information about the possible predictive value of our model and our variables.

R code and R output:

```
> attach(Auto);
> Auto=na.omit(Auto);
> Auto=Auto[,1:8];
> dim(Auto)
[1] 392   8
> names(Auto)
[1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"
[6] "acceleration" "year"         "origin"
> summary(Auto)
      mpg          cylinders      displacement     horsepower
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0
 1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0
 Median :22.75   Median :4.000   Median :151.0   Median : 93.5
 Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0
    weight      acceleration        year           origin
 Min.   :1613   Min.   : 8.00   Min.   :70.00   Min.   :1.000
 1st Qu.:2225   1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000
 Median :2804   Median :15.50   Median :76.00   Median :1.000
 Mean   :2978   Mean   :15.54   Mean   :75.98   Mean   :1.577
 3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000
 Max.   :5140   Max.   :24.80   Max.   :82.00   Max.   :3.000
> ?Auto
starting httpd help server ... done
>
> #Multiple Linear Regression
>
> lm.fit=lm(mpg~.,data=Auto)
> summary(lm.fit)

Call:
```

```
lm(formula = mpg ~ ., data = Auto)

Residuals:
    Min      1Q  Median      3Q     Max
-9.5903 -2.1565 -0.1169  1.8690 13.0604

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
cylinders     -0.493376   0.323282  -1.526  0.12780
displacement   0.019896   0.007515   2.647  0.00844 **
horsepower    -0.016951   0.013787  -1.230  0.21963
weight        -0.006474   0.000652  -9.929  < 2e-16 ***
acceleration   0.080576   0.098845   0.815  0.41548
year           0.750773   0.050973  14.729  < 2e-16 ***
origin         1.426141   0.278136   5.127 4.67e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.328 on 384 degrees of freedom
Multiple R-squared:  0.8215,       Adjusted R-squared:  0.8182
F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16

>
> #Training and Test Set
>
> set.seed(1)
> train.Auto=sample(1:nrow(Auto),nrow(Auto)/2)
> test.Auto=Auto[-train.Auto,"mpg"]
>
>
> #Linear Regression Model
>
> lm.fit<-lm(mpg~.,data=Auto[train.Auto,])
> lm.predict<-predict(lm.fit,newdata=Auto[-train.Auto,])
> mean((lm.predict-test.Auto)^2)
```

```
[1] 11.3998
>
> #Regression Tree
>
> tree.Auto=tree(mpg~., data=Auto[train.Auto,])
> summary(tree.Auto)

Regression tree:
tree(formula = mpg ~ ., data = Auto[train.Auto, ])
Variables actually used in tree construction:
[1] "weight"     "year"       "horsepower"
Number of terminal nodes:  9
Residual mean deviance:  7.615 = 1424 / 187
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-6.43200 -1.58100  0.09921  0.00000  1.56800 11.23000
> plot(tree.Auto)
```




```
>
text(tree.Auto,pretty=0)
> tree.Auto
```

```
node), split, n, deviance, yval
      * denotes terminal node

 1) root 196 11690.00 23.00
   2) weight < 3018 106  3717.00 28.37
     4) year < 78.5 59  1203.00 25.21
       8) weight < 2091.5 13    89.35 31.49 *
       9) weight > 2091.5 46   456.10 23.44
        18) weight < 2767 34   264.50 24.43 *
        19) weight > 2767 12    62.44 20.62 *
     5) year > 78.5 47  1184.00 32.34
      10) weight < 2570.5 27   317.40 35.37 *
      11) weight > 2570.5 20   285.60 28.25 *
   3) weight > 3018 90  1313.00 16.68
     6) horsepower < 127 36   360.20 19.92
      12) year < 78.5 26    97.59 18.79 *
      13) year > 78.5 10   143.50 22.85 *
     7) horsepower > 127 54   323.10 14.52
      14) year < 76.5 41   135.40 13.55 *
      15) year > 76.5 13    28.17 17.57 *
> tree.pred=predict(tree.Auto,newdata=Auto[-train.Auto,])
> mean((tree.pred-test.Auto)^2)
[1] 10.952
>
> #Boosted Regression Tree
>
>
boost.Auto=gbm(mpg~.,data=Auto[train.Auto,],distribution="gaussian",n.trees=5000,interaction.depth=4)
> summary(boost.Auto)
                    var    rel.inf
weight           weight 37.5616878
horsepower     horsepower 23.3311264
year             year 16.9340335
displacement displacement 14.6040960
cylinders       cylinders  4.2328115
acceleration acceleration  2.4407893
```

```
origin          origin  0.8954554
> par(mfrow=c(2,2))
> plot(boost.Auto,i="weight")
> plot(boost.Auto,i="horsepower")
> plot(boost.Auto,i="year")
> yhat.boost=predict(boost.Auto,newdata=Auto[-train.Auto,],n.trees=5000)
> mean((yhat.boost-test.Auto)^2)
[1] 7.211634
>
>
> #Specify Shrinkage lambda=0.02
>
>
boost.Auto2=gbm(mpg~.,data=Auto[train.Auto,],distribution="gaussian",n.trees=5000,interaction.depth=3,
shrinkage=0.02,verbose=F)
> yhat.boost2=predict(boost.Auto2,newdata=Auto[-train.Auto,],n.trees=5000)
> mean((yhat.boost2-test.Auto)^2)
[1] 9.698558
>
> #Specify Shrinkage lambda=0.01
>
>
boost.Auto2=gbm(mpg~.,data=Auto[train.Auto,],distribution="gaussian",n.trees=5000,interaction.depth=3,
shrinkage=0.01,verbose=F)
> yhat.boost2=predict(boost.Auto2,newdata=Auto[-train.Auto,],n.trees=5000)
> mean((yhat.boost2-test.Auto)^2)
[1] 9.160304
>
>
> #Random Forest
>
> rf.Auto=randomForest(mpg~.,data=Auto,subset=train.Auto, mtry=6, importance=TRUE)
> yhat.rf=predict(rf.Auto,newdata=Auto[-train.Auto,])
> mean((yhat.rf-test.Auto)^2)
[1] 7.511008
```