STOR 664 Final Project

Introduction

The introduction of electric cars has come recently as a solution to the long-term inability of sustaining traditional gasoline-powered vehicles. While this is a large step in reducing transportation reliance on fossil fuels, how efficient and cost-effective is using one of these vehicles today? While being electric means that there is no money spent on filling up the tank with gas, there is still a cost of charging these vehicles, not to mention the upfront cost that comes with it. This analysis aims to answer the question of whether or not electric cars are more cost-effective to travel in, and if so, how much more cost-effective?

This report covers findings on the efficiency and cost-effectiveness of electric cars compared to other fuel types. Car data includes mileage, emissions, and other areas from cars created between 1984 and 2017. Models are created from this data to deduce the greatest factors towards car usage cost and these findings are then used to conclude that electric cars are overall more cost-effective in the measurement of travel distance. Another question answered in this report is how has car emissions changed over time and between makes? Electric cars have no fuel emission when in use, so this question is aimed towards traditional transportation vehicles. A model was created to deduce the greatest factors affecting carbon dioxide emission and the findings were used to see how certain variables affected carbon dioxide output.

Data Overview

The data set consists of car models made from the years 1984 to 2017. This fuel economy data is the result of vehicle testing done by the Environmental Protection Agency's National Vehicle and Fuel Emissions Laboratory and by vehicle manufacturers. The data was retrieved from Kaggle at the url https://www.kaggle.com/datasets/thedevastator/fuel-economy-data-how-efficient-are-today-s-cars?resource=download.

The original data had 37936 and 84 variables which describe the travel efficiency and emissions of different cars. Notable variables include Miles Per Gallon, CO2 emission (grams per mile), year manufactured, and fuel types.

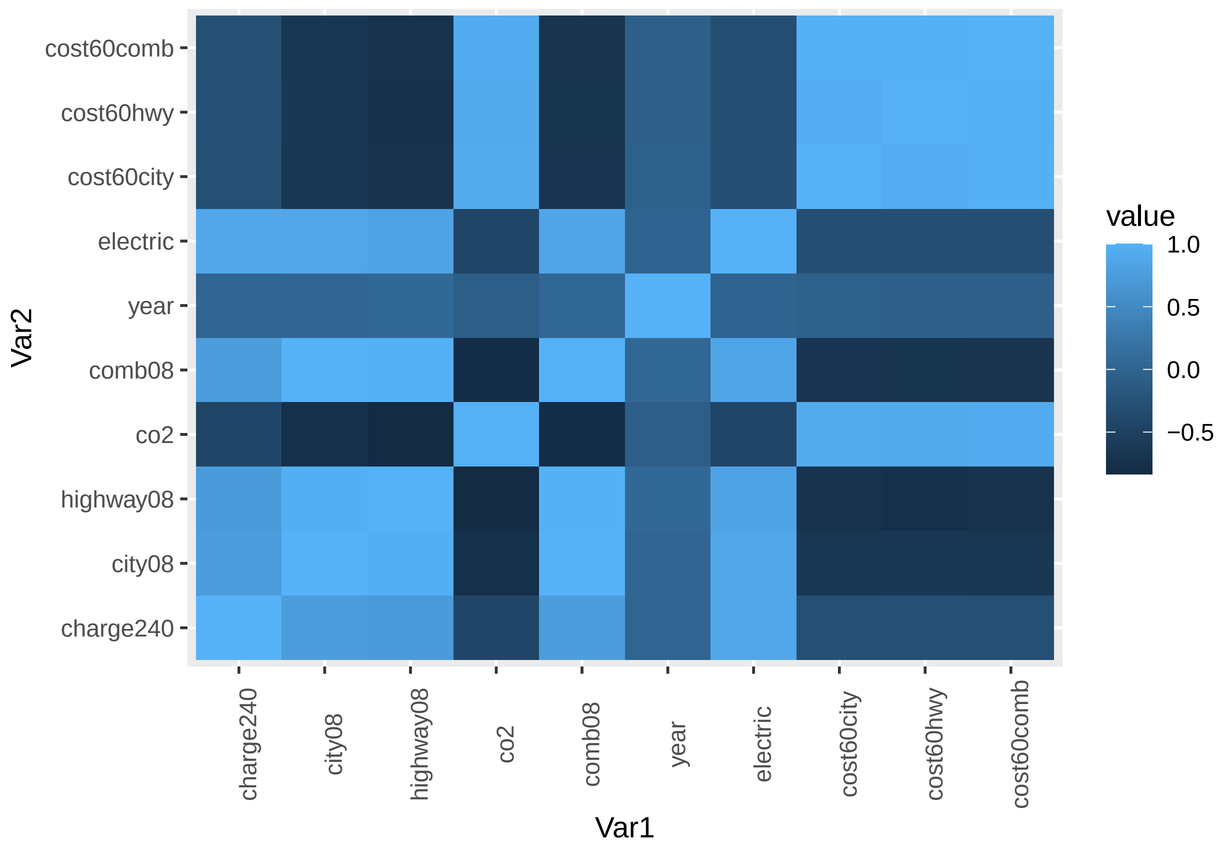
This data set was interesting for a few reasons. The first was that the data spans almost 3 decades. This means that there was a lot of potential for measuring fuel trends over an extended period of time. The second reason was the possibility of using the variables to build a cost of travel prediction model. Many other data sets focused on upfront cost of the cars, but few had the required variables to calculate car efficiency easily. Variables such as fuel type and make also made it easier to distinguish electric cars from traditional cars, making it easier to find electric car effects. The data also allowed for a model predicting CO2 emissions over a long range of time.

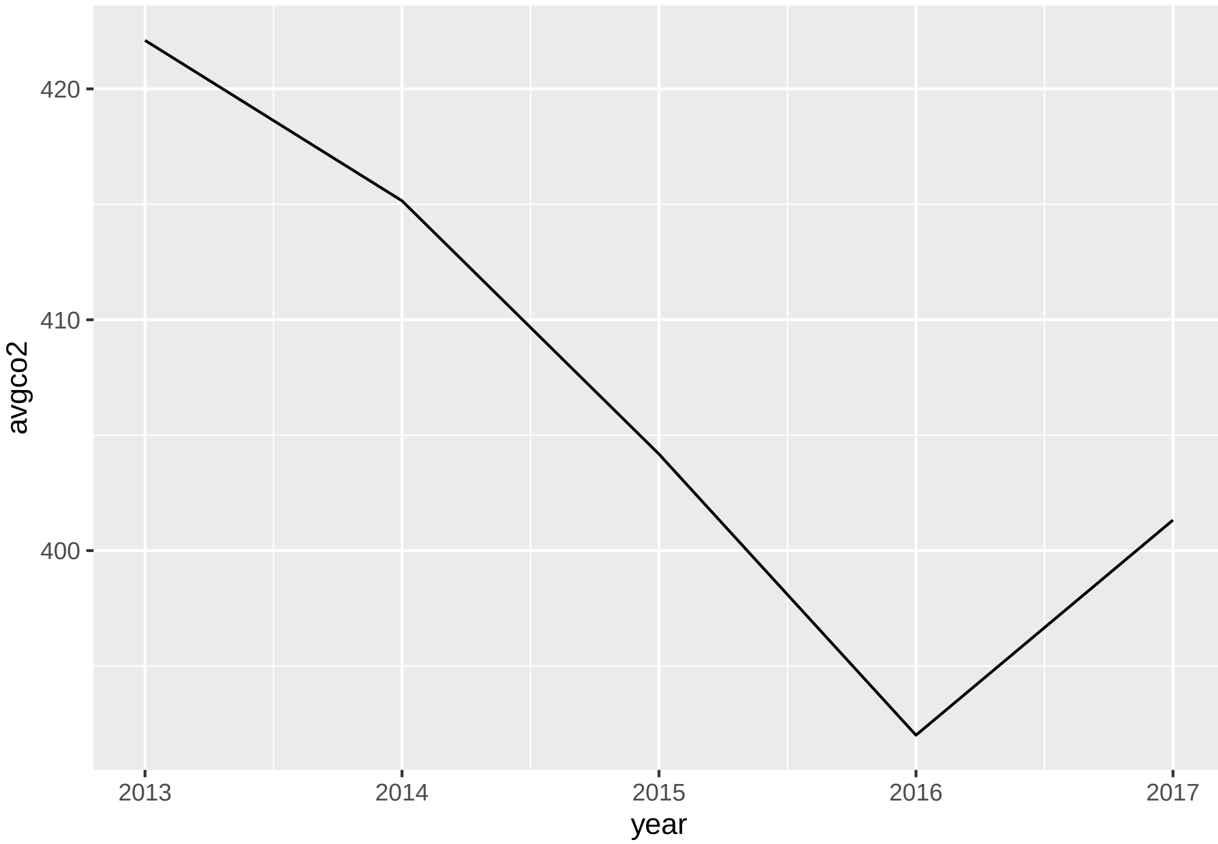
The data contained a very high number of improbable values for CO2 emissions (the value being -1). Further investigation found that the cars without proper values ranged mostly between 1986 and 2012. As a big question of this project was to see the effects of various variables on CO2 emission, it was decided to remove those rows without CO2 emissions rather than imputing values, as these would greatly affect our CO2 model. After removing these rows, the data was left with 5783 full observations spanning the years 2013 to 2017.

To prepare the data for cost modeling and exploratory analysis, a column was created based on fuel type to show if a car was fully electric or not, in this study hybrid cars were treated as traditional vehicles as they still relied on gas. 3 other columns were also created by using each car's variables for miles per gallon in the city, highway, and combination of the two (calculated as the mean of city and highway mpg). These mpg values were transformed based on the car's fuel type to determine the cost of gas for traveling 60 miles with the car. Price data was taken from gasprices.aaa.com on the date 11/25/2022. Price for electric cars was calculated by treating 33.7kWh as the electrical equivalent of one gallon of gasoline, with its own price taken on the same day. These would be the main variable to measure a car's cost-effectiveness in travel and transportation.

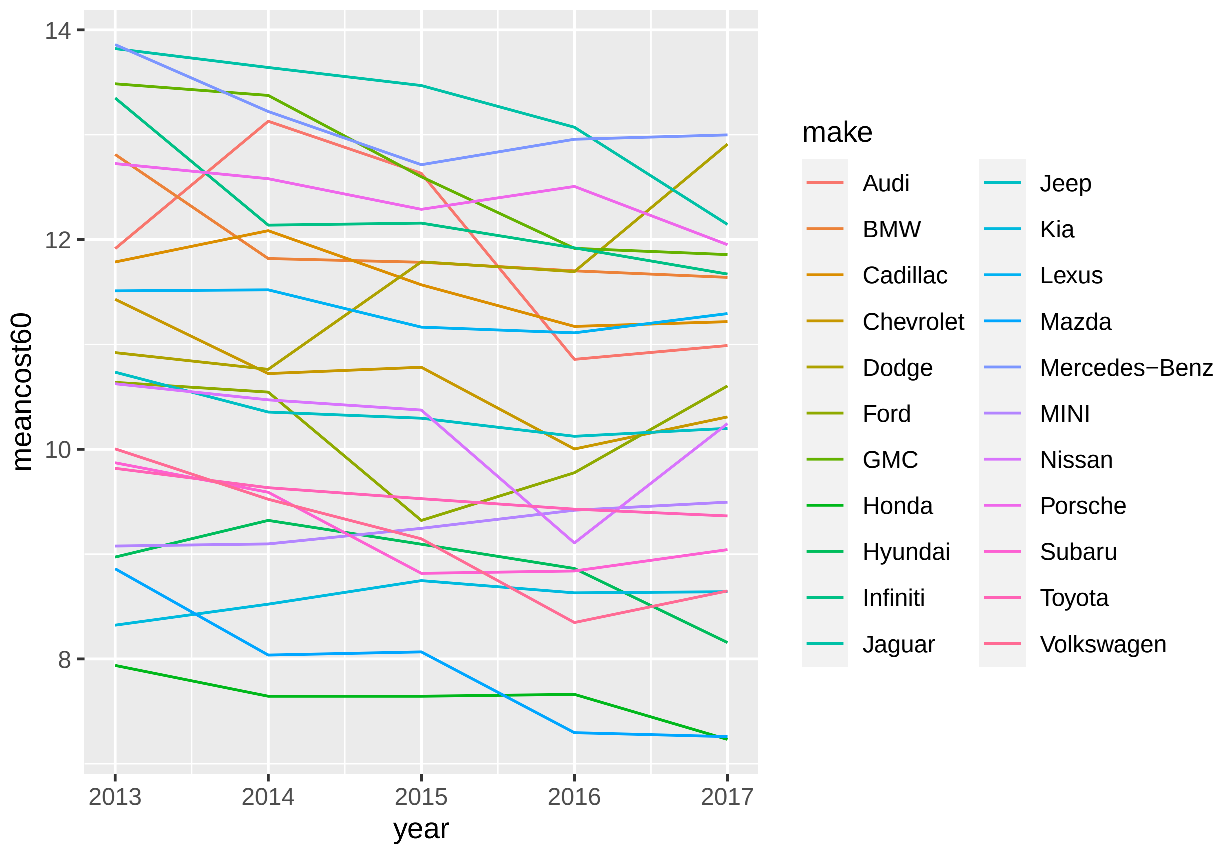
Exploratory Data Analysis

Plotting a correlation plot between the numerical variables shows that CO2 has some strong negative correlations with mileage as well as moderate correlations with year. It is also shown that the cost of traveling 60 miles correlates negatively with mileage and being electric.

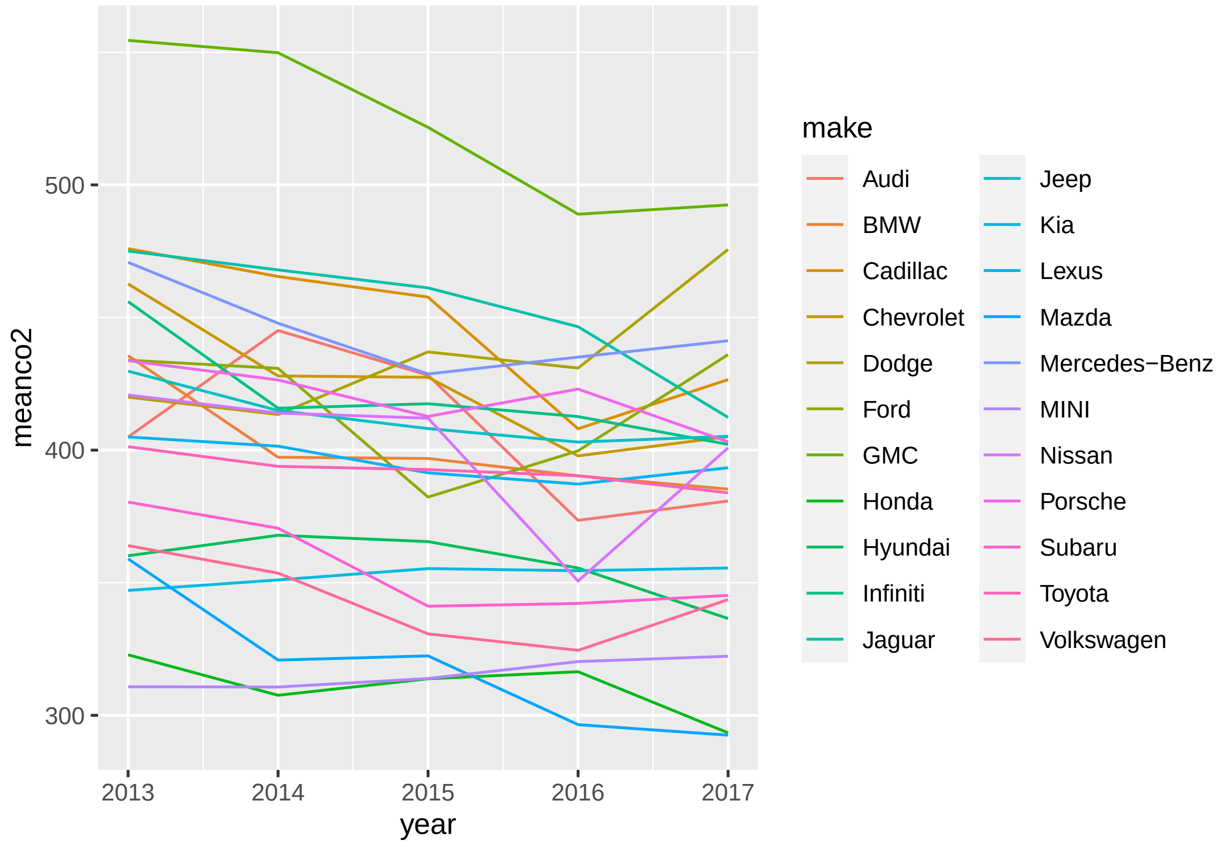




| Plotting avgco2 emission against year shows an average decrease in CO_2 emissions from cars as years increase.



| For categorical variables the main one focused on was the make of the car. From this graph we see that almost every make of car has had a decrease in cost per 60 miles over time, we also note that no makes have drastic increases or decreases, but instead generally stay the same relative to one another.



| It is shown that there seems to be a small decreasing trend in average CO2 emissions from each individual make, while GMC has significantly higher average emission than other makes.

Do electric cars have more city/highway mileage?

```
t.test(filter(df2,electric==TRUE)$city08,filter(df2,electric==FALSE)$city08,alternative="greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  filter(df2, electric == TRUE)$city08 and filter(df2, electric == FALSE)$city08
## t = 40.419, df = 83.202, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  83.41783      Inf
## sample estimates:
## mean of x mean of y
## 107.13095  20.13283
```

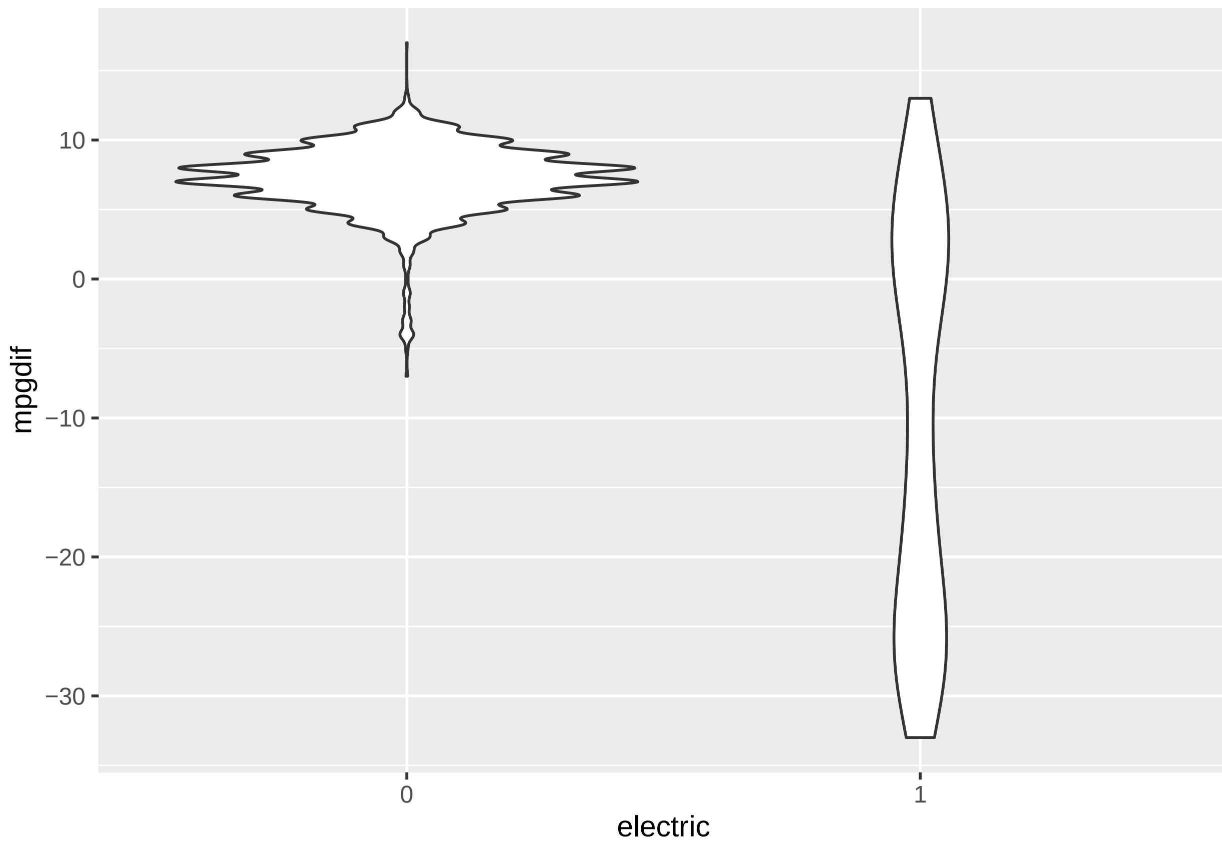
Do electric cars have more city mileage than highway mileage?

```
t.test(filter(df2,electric==TRUE)$city08,filter(df2,electric==TRUE)$highway08,alternative="greater")
```

```
##
##  Welch Two Sample t-test
##
```

```
## data:  filter(df2, electric == TRUE)$city08 and filter(df2, electric == TRUE)$highway08
## t = 4.2258, df = 126.53, p-value = 2.26e-05
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  6.259803      Inf
## sample estimates:
## mean of x mean of y
## 107.13095  96.83333
```

Graph of mpg difference between highway and city (highway-city) for electric and non-electric vehicles



Taking a look at overall mileage, we see that electric cars do have a significantly higher city and highway mileage. We also notice that the difference between an electric car's highway and city mileage is significantly larger than a gas car's difference. It is also found that the majority of electric cars actually have a higher city mileage than highway mileage, as opposed to traditional cars almost always having a higher highway mileage. This is a strong indicator that electric cars could be much more effective in cities, as well as being more cost-effective in general.

Methods and Modeling

Three linear regression methods were chosen to train one model for CO2 and two for cost per 60 miles (one for city, one for highway).

OLS

Ordinary Least Squares regression is the simplest of linear regression models, which aims to find the onedimensional line in data that will result in the minimum of the sum of all squared distances of observation points from that line. It is used to model linear regression and predict a response variable by predictor variables, assuming they have a linear relationship.

Ridge Regression

Ridge Regression is a method of linear regression that adds a penalty term that is equal to the square of the coefficient of each predictor. There is also a coefficient added to the penalty term that penalizes large predictor coefficients. If the penalty term is zero, then the method as OLS. As we increase the value of the penalty term, it causes the value of the coefficient to trend towards zero. This leads to lower variance and low training bias.

LASSO Regression

LASSO Regression, short for Least Absolute Shrinkage and Selection Operator Regression, is a linear regression model that, in a similar fashion to Ridge Regression, adds a penalty term and a regularization . It adds a penalty term to the cost function. This term is the sum of the absolute value of the coefficients. As the value of coefficients increases from 0 this term increases, causing the model to decrease the value of coefficients in order to reduce loss. As opposed to Ridge Regression, which lowers the value of coefficients but won't reduce dimensionality, LASSO Regression tends to set coefficients equal to zero.

Predictor variables for each of the 3 models were chosen by backwards selection, while also removing variables with high multi-collinearity. These models were tested for proper linear fit and after analysis the two mileage models had their response squared to better fit a linear regression. 3 methods were then used to train each model resulting in 9 total models. Each model was tested using 5-folds Cross Validation to determine the 3 best models, one predicting CO2 emissions, one for city mileage, and another for highway mileage. The models with the lose RMSE were selected as final models, which were Ridge for CO2, OLS for city mileage, and Ridge for highway mileage.

Results and Discussion

CO2 Model

The CO2 model was predicted by the cost to travel 60 miles, the years since 2012 that the car was manufactured, and the make of the car. The model was highly effective in that it represented 93.86% of the variability in CO2 emissions. The model deduced that each of these variables were significant. The output of the ridge regression coefficients shows that CO2 in grams per mile emission is greatly increased for each dollar that it costs to travel 60 miles, with a coefficient of 35.132. It also shows the makes of cars that have the greatest effect in increasing CO2 emissions, with the top ones being Mobility Ventures LLC, GMC, VPG, Lincoln, and Ford. The model also shows a decrease in CO2 emissions with years since 2012, with a coefficient of -1.105.

While the model was very effective in predicting CO2 emissions and showed the significant effects of mileage and make on CO2 emissions. It would have been more insightful to have emission data from a longer span of years to see emission trends in a longer, more stable period of time.

City/Highway cost mileage model

The city cost for 60 miles was predicted by city mileage, years since 2012, and the make of the car. The model represented 64.35% of the variability in the training data and had a mean average Percentile error of 68.45%. While this is a rather large error, it is good to know that this could be compounded by the squared response variable. The model output shows that city mileage and years since 2012 are significant and negatively correlating with cost of going 60 miles in a city. It is important to take note that many electric car manufacturers such as Tesla have a positive coefficient, which suggest that those makes increase cost for city mileage, but this could be due to the correlations between electric cars and mileage, which means that since electric cars have such high mileage it may make the make coefficients “wrong” in some cases.

The highway cost for 60 miles was predicted by highway mileage, years since 2012, and the make of the car. The model accounted for 68.99% of the variability in the training data, similar to the city model. Again the unpredictability of the model could be compounded by the squared response variable. The model shows that highway mileage and years since 2012 are significant and negatively correlating with the cost of going 60 miles on a highway. Again the same error with mainly electric manufacturers for the city model can be said for the highway model, as Tesla has a very large coefficient, but when looking at mileage Tesla has very high mileage.

While all the variables in the city and highway models are significant, the one that links cost to electric vehicles is the mileage. The t-test for difference of means shows that electric cars do have a significantly greater mileage than non-electric cars in the city and highway. This paired with the coefficients of the models is evidence to support that electric cars do have a lower operating cost than traditional gas vehicles. Another discovery is that electric cars have significantly higher average city mileage that highway mileage. The city and highway models also show that city mileage variable in the city model has a more effective coefficient, -3.76, than the highway mileage variable in the highway model, -2.87. This provides evidence that for one, electric cars are more efficient in city areas than suburban or rural. It also implies that switching to an electric car will save you more if you are in the city.

Conclusion

In regards to CO2 emission over time and between car manufacturers. It is concluded that car emissions have a decreasing trend in years. While there are certain years that average emission from new cars might surpass previous years, the overall trend is a decreasing one. For future studies on CO2 emissions from cars, it would be beneficial to have from from a larger span of years to compare and see if the trends are the same or different in a longer time span.

Using Ridge and OLS Regression methods, moderately strong traveling cost prediction models were created. 80-20 split testing for the models resulted in decent RMSEs. From these models and difference of means testing it is concluded that electric cars are more cost-effective than non0electric vehicles due to their superior mileage. A future attempt could be made with other regression methods, both linear and nonlinear. Additional features from other data sets could also supplement this data to make a stronger model that could tell more about electric cars. Data on car pricing could also be useful in determining how much time it would take to level the upfront cost of an electric vehicle after purchasing it.

Appendix

Output of cross validation and models

```
#specify the cross-validation method
ctrl <- trainControl(method = "cv", number = 5)

# OLS
model <- train(co2~cost60comb+year2012+make,data=df3,method="lm",trControl=ctrl)
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
print(model)
```

```
## Linear Regression
##
## 5699 samples
##    3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 4560, 4559, 4560, 4560, 4557
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   25.98252  0.9368555  16.25025
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
# Ridge
model <- train(co2~cost60comb+year2012+make,data=df3,method="ridge",trControl=ctrl)
```

```
## Warning: model fit failed for Fold1: lambda=0e+00 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance
```

```
## Warning: model fit failed for Fold1: lambda=1e-01 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance
```

```
## Warning: model fit failed for Fold1: lambda=1e-04 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance
```

```
## Warning: model fit failed for Fold4: lambda=0e+00 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance
```

```
## Warning: model fit failed for Fold4: lambda=1e-01 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance
```

```
## Warning: model fit failed for Fold4: lambda=1e-04 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```r
print(model)
```

```
## Ridge Regression
##
## 5699 samples
##    3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 4560, 4559, 4559, 4560, 4558
## Resampling results across tuning parameters:
##
##   lambda  RMSE      Rsquared   MAE
##   0e+00   25.59248  0.9398188  16.02666
##   1e-04   25.59235  0.9398191  16.02750
##   1e-01   27.10877  0.9322883  18.10950
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was lambda = 1e-04.
```

```r
# LASSO
model <- train(co2~cost60comb+year2012+make,data=df3,method="lasso",trControl=ctrl)
```

```
## Warning: model fit failed for Fold2: fraction=0.9 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold3: fraction=0.9 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```r
print(model)
```

```
## The lasso
##
## 5699 samples
##    3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 4559, 4559, 4558, 4560, 4560
## Resampling results across tuning parameters:
##
##   fraction  RMSE      Rsquared   MAE
##   0.1       73.88739  0.8492017  57.52127
##   0.5       30.38956  0.9135196  21.73233
##   0.9       26.26008  0.9345808  16.67303
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was fraction = 0.9.
```

```r
# Make model and look at coefficients
rgmod=lm.ridge(co2~cost60comb+year2012+make,data=df3,lambda=1e-4)
coef(rgmod)
```

```
##                                          cost60comb                  year2012
##                -8.6854556                35.1321398                -1.1045796
##            makeAlfa Romeo           makeAston Martin                  makeAudi
##                 2.5967346               -12.6210420                -0.1036286
##               makeBentley                   makeBMW               makeBugatti
##                -2.7919677                -6.1462771               -57.5786827
##                 makeBuick              makeCadillac             makeChevrolet
##                66.1442487                53.0554864                63.5598646
##              makeChrysler                 makeDodge               makeFerrari
##                61.3325852                39.3037874                -6.7456964
##                  makeFiat                  makeFord               makeGenesis
##                 8.1704301                72.4489689                27.7403658
##                   makeGMC                 makeHonda               makeHyundai
##                89.1327179                56.7553586                57.2185464
##              makeInfiniti               makeJaguar                  makeJeep
##                 2.6221573                -0.1529038                60.7312675
##                   makeKia          makeLamborghini            makeLand Rover
##                65.8722184                -4.3686170                -2.8443779
##                 makeLexus               makeLincoln                 makeLotus
##                10.0716930                78.0765466                -4.7166573
##              makeMaserati                 makeMazda  makeMcLaren Automotive
##                -5.2969641                52.6091954                -5.5976894
##         makeMercedes-Benz                  makeMINI            makeMitsubishi
##                -4.2838470                 1.8981265                47.9623922
## makeMobility Ventures LLC                makeNissan                makePagani
##               123.7613990                57.8655365               -21.2110349
##               makePorsche                   makeRam           makeRolls-Royce
##                -4.3297476                46.0192981                -8.9868471
##     makeRoush Performance                 makeScion                 makesmart
##               -12.8095151                42.2956894                 0.4561808
##                   makeSRT                makeSubaru                makeSuzuki
##                 7.0267925                43.6443444                62.1886219
##                makeToyota           makeVolkswagen                makeVolvo
##                70.0158138                34.7200364                49.1608679
##                   makeVPG
##                87.9788491
```

```r
# OLS
model <- train(cost60city^2~city08+year2012+make,data=df2,method="lm",trControl=ctrl)
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```r
print(model)
```

```
## Linear Regression
##
## 5783 samples
##    3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 4625, 4627, 4626, 4626, 4628
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   64.65877  0.6350036  46.06318
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
# Ridge
model <- train(cost60city^2~city08+year2012+make,data=df2,method="ridge",trControl=ctrl)
```

```
## Warning: model fit failed for Fold4: lambda=0e+00 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold4: lambda=1e-01 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold4: lambda=1e-04 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold5: lambda=0e+00 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold5: lambda=1e-01 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold5: lambda=1e-04 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```
print(model)
```

```
## Ridge Regression
##
## 5783 samples
##    3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 4625, 4627, 4628, 4626, 4626
## Resampling results across tuning parameters:
##
```

```
##   lambda  RMSE      Rsquared   MAE
##   0e+00   64.10803  0.6406474  46.34155
##   1e-04   64.10797  0.6406474  46.34105
##   1e-01   64.15177  0.6397367  46.37508
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was lambda = 1e-04.
```

```
# LASSO
model <- train(cost60city^2~city08+year2012+make,data=df2,method="lasso",trControl=ctrl)
```

```
## Warning: model fit failed for Fold1: fraction=0.9 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance
```

```
## Warning: model fit failed for Fold3: fraction=0.9 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```
print(model)
```

```
## The lasso
##
## 5783 samples
##    3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 4626, 4627, 4626, 4627, 4626
## Resampling results across tuning parameters:
##
##   fraction  RMSE      Rsquared   MAE
##   0.1       93.37644  0.3050405  69.10428
##   0.5       71.60901  0.5877221  54.22880
##   0.9       64.27080  0.6502864  46.39870
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was fraction = 0.9.
```

```
citycostmod <- lm(cost60city^2~city08+year2012+make,data=df2)
summary(citycostmod)
```

```
##
## Call:
## lm(formula = cost60city^2 ~ city08 + year2012 + make, data = df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -171.15  -37.74  -12.30   22.15  325.77
##
```

```
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)    
## (Intercept)              254.40198    8.35001  30.467  < 2e-16 ***
## city08                    -3.76337    0.08653 -43.493  < 2e-16 ***
## year2012                  -4.59548    0.61759  -7.441 1.15e-13 ***
## makeAlfa Romeo           -30.11161   46.08971  -0.653 0.513572    
## makeAston Martin         204.48126   13.35614  15.310  < 2e-16 ***
## makeAudi                  36.18394    9.11893   3.968 7.34e-05 ***
## makeBentley              248.28673   12.92323  19.212  < 2e-16 ***
## makeBMW                   34.90072    8.44171   4.134 3.61e-05 ***
## makeBugatti              845.86370   37.93105  22.300  < 2e-16 ***
## makeBuick                -42.36477   10.57843  -4.005 6.28e-05 ***
## makeBYD                   19.19293   38.10181   0.504 0.614472    
## makeCadillac              11.59911    9.53865   1.216 0.224030    
## makeChevrolet              2.63178    8.61379   0.306 0.759973    
## makeChrysler             -14.72242   12.62271  -1.166 0.243524    
## makeCODA Automotive       52.70958   64.88375   0.812 0.416613    
## makeDodge                 27.30445    9.53674   2.863 0.004211 ** 
## makeFerrari              228.33604   11.32855  20.156  < 2e-16 ***
## makeFiat                 -12.61235   12.36136  -1.020 0.307627    
## makeFord                 -18.06805    8.59786  -2.101 0.035644 *  
## makeGenesis               67.15016   25.57491   2.626 0.008672 ** 
## makeGMC                   32.47491    9.22653   3.520 0.000435 ***
## makeHonda                -56.89961    9.67494  -5.881 4.30e-09 ***
## makeHyundai              -44.96737    9.15618  -4.911 9.31e-07 ***
## makeInfiniti              32.20723    9.71723   3.314 0.000924 ***
## makeJaguar                67.93388   10.01757   6.781 1.31e-11 ***
## makeJeep                 -25.72272    9.43620  -2.726 0.006431 ** 
## makeKia                  -51.79077    9.22906  -5.612 2.10e-08 ***
## makeLamborghini          294.60658   14.72027  20.014  < 2e-16 ***
## makeLand Rover            92.09958   12.37604   7.442 1.14e-13 ***
## makeLexus                 15.10645    9.56413   1.579 0.114279    
## makeLincoln              -21.18306   10.38180  -2.040 0.041356 *  
## makeLotus                 35.77783   21.81285   1.640 0.101015    
## makeMaserati             147.22840   13.73971  10.716  < 2e-16 ***
## makeMazda                -63.00151    9.96860  -6.320 2.81e-10 ***
## makeMcLaren Automotive    92.21862   20.17883   4.570 4.98e-06 ***
## makeMercedes-Benz         68.46381    8.64905   7.916 2.93e-15 ***
## makeMINI                 -40.22918    9.19132  -4.377 1.23e-05 ***
## makeMitsubishi           -36.21388   10.65166  -3.400 0.000679 ***
## makeMobility Ventures LLC   2.02966   46.09699   0.044 0.964882    
## makeNissan               -11.49326    9.02358  -1.274 0.202824    
## makePagani               361.95456   46.09640   7.852 4.85e-15 ***
## makePorsche               35.87841    8.90654   4.028 5.69e-05 ***
## makeRam                   45.00314   12.08304   3.724 0.000198 ***
## makeRolls-Royce          304.26508   13.36405  22.767  < 2e-16 ***
## makeRoush Performance    246.79719   17.50370  14.100  < 2e-16 ***
## makeScion                -48.86895   14.03406  -3.482 0.000501 ***
## makesmart                 68.44821   17.66862   3.874 0.000108 ***
## makeSRT                  269.19291   46.10225   5.839 5.54e-09 ***
## makeSubaru               -42.79444   10.13276  -4.223 2.44e-05 ***
## makeSuzuki               -68.96272   17.95508  -3.841 0.000124 ***
## makeTesla                118.32402   15.31282   7.727 1.29e-14 ***
## makeToyota               -25.74227    8.89877  -2.893 0.003833 ** 
```

```
## makeVolkswagen            -37.10939    9.40470  -3.946 8.05e-05 ***
## makeVolvo                 -36.36718   10.27180  -3.540 0.000403 ***
## makeVPG                    71.82443   64.70378   1.110 0.267024
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 64.2 on 5728 degrees of freedom
## Multiple R-squared:  0.6435, Adjusted R-squared:  0.6401
## F-statistic: 191.5 on 54 and 5728 DF,  p-value: < 2.2e-16
```

```
MAPE(citycostmod$fitted.values,df2$cost60city^2)
```

```
## [1] 0.6845049
```

```
# OLS
model <- train(cost60hwy^2~highway08+year2012+make,data=df2,method="lm",trControl=ctrl)
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
print(model)
```

```
## Linear Regression
##
## 5783 samples
##    3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 4626, 4626, 4627, 4627, 4626
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   27.56081  0.6628417  17.91399
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
# Ridge
model <- train(cost60hwy^2~highway08+year2012+make,data=df2,method="ridge",trControl=ctrl)
```

```
## Warning: model fit failed for Fold1: lambda=0e+00 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance
```

```
## Warning: model fit failed for Fold1: lambda=1e-01 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold1: lambda=1e-04 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold2: lambda=0e+00 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold2: lambda=1e-01 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold2: lambda=1e-04 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold4: lambda=0e+00 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold4: lambda=1e-01 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold4: lambda=1e-04 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold5: lambda=0e+00 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold5: lambda=1e-01 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold5: lambda=1e-04 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```
print(model)
```

```
## Ridge Regression
##
## 5783 samples
##    3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 4627, 4627, 4627, 4626, 4625
## Resampling results across tuning parameters:
##
##   lambda  RMSE      Rsquared   MAE
##   0e+00   29.16748  0.6337370  17.36483
```

```
##   1e-04   29.16665  0.6337506  17.36505
##   1e-01   28.66513  0.6424214  17.64206
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was lambda = 0.1.
```

```r
# LASSO
model <- train(cost60hwy^2~highway08+year2012+make,data=df2,method="lasso",trControl=ctrl)
```

```
## Warning: model fit failed for Fold4: fraction=0.9 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning: model fit failed for Fold5: fraction=0.9 Error in elasticnet::enet(as.matrix(x), y, lambda =
##   Some of the columns of x have zero variance

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```r
print(model)
```

```
## The lasso
##
## 5783 samples
##    3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 4627, 4626, 4626, 4627, 4626
## Resampling results across tuning parameters:
##
##   fraction  RMSE      Rsquared   MAE
##   0.1       38.55093  0.4008474  28.54611
##   0.5       29.72453  0.6070162  21.26680
##   0.9       27.15625  0.6657159  17.94487
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was fraction = 0.9.
```

```r
# Make model and look at coefficients
rgmod=lm.ridge(cost60hwy^2~highway08+year2012+make,data=df2,lambda=1e-4)
coef(rgmod)
```

```
##                                     highway08                year2012
##              170.887089             -2.866006               -1.493649
##          makeAlfa Romeo     makeAston Martin                makeAudi
##               -9.476411             67.021300                5.047528
##             makeBentley               makeBMW             makeBugatti
##               52.234417             11.094012              176.876681
##               makeBuick               makeBYD            makeCadillac
##              -24.143243             41.529480               -3.518188
##           makeChevrolet          makeChrysler      makeCODA Automotive
```

```
##            -4.504282              -19.997442               41.826083
##            makeDodge             makeFerrari                makeFiat
##            -5.078310              106.882007                6.264858
##             makeFord             makeGenesis                 makeGMC
##           -11.385322               12.722032                5.073654
##            makeHonda             makeHyundai            makeInfiniti
##           -22.710289              -21.202997               12.633525
##           makeJaguar                makeJeep                 makeKia
##            16.473661              -12.530695              -22.827361
##      makeLamborghini         makeLand Rover               makeLexus
##            82.267621               45.166626                6.153587
##          makeLincoln               makeLotus            makeMaserati
##           -16.662049                6.461067               38.673170
##            makeMazda makeMcLaren Automotive       makeMercedes-Benz
##           -23.085816               39.249480               30.334339
##             makeMINI          makeMitsubishi makeMobility Ventures LLC
##            -8.948822              -12.031648              -12.142186
##           makeNissan              makePagani             makePorsche
##            -4.840261              117.270756               12.180582
##              makeRam          makeRolls-Royce    makeRoush Performance
##            11.765772               67.674907              108.347015
##            makeScion               makesmart                 makeSRT
##           -17.514160               42.286679               73.901506
##           makeSubaru              makeSuzuki               makeTesla
##           -16.200294              -29.735585              121.662946
##           makeToyota          makeVolkswagen              makeVolvo
##            -8.930907              -13.645862              -18.361831
##              makeVPG
##            24.440047
```