Sta 141A Project: Auto-MPG Analysis

11/2/2021

Name	Email	Contributions
Aditi Goyal	adigoyal@ucdavis.edu	Report Writing, table formatting.
Brandon Hom	bwhom@ucdavis.edu	Unsupervised and supervised learning analyses, web app, data visualization, tables
Tammie Tam	tastam@ucdavis.edu	Report Writing , plot formatting

Contents

Introduction

Fuel economy is defined as how much a car can travel per volume of fuel, which is usually measured as miles per gallon (mpg). Cars with low mpg have good fuel economy and vice versa. Logically, consumers prefer to own cars with good fuel economy, since they would spend less money on gas. In addition, cars with poor fuel economy consumes more gas, which, in turn, contributes to global warming; gas is ultimately a limited resource [1]. The effects of pollution produced from the consumption of gas can be mitigated by using cars with good fuel economy. The ability to predict a car's fuel economy based on a set of a given car's characteristics, or information on car models with good fuel economy, can allow individual to make a more informed decision when purchasing a car - a decision that can have a positive impact on both spending and global warming.

Dataset Description

The data set on fuel economy was obtained from kaggle: https://www.kaggle.com/uciml/autompg-dataset. Although it is data from the late 1900s, the features within the data set can provide us insight on what variables have impact on fuel economy.The dimensions of the data set are 398 by 9, meaning that we have 9 features within our data set. `Model year` is not useful for our purposes, so it will be dropped during the analysis. `Weight`, `acceleration`, `horsepower`, `displacement`, `cylinders` are all numerical features, while `origin` is categorical. `Weight`, `acceleration`, and `horsepower` are self-explanatory. `Cylinders` are indicative of the power of an engine, where more cylinders means more power but more consumption of gas. `Displacement` refers to how much volume of air and fuel moved through the cylinders of the engine.

Research Questions

- Is it appropriate to fit a linear model to this data? If so, what numerical variables have the most impact on mpg? For example, if increasing weight contributes the most to MPG, individuals should be wary about purchasing heavy cars, since it will lead to more consumption of fuel.
- Is it possible to build a predictive model with reasonable performance to predict a car's fuel economy? If it is possible, individuals will be able to make better car-purchasing decisions by inputting a car's features into the model and getting an estimated mpg.
- The origin column has Europe, Japan and USA encoded. Are cars from these regions similar, or are they completely different? Is there a region that tends to make cars with good fuel economy? What about a region that produces cars with poor fuel economy?
- What brand of cars are similar in terms of fuel economy and other features such as weight? Knowing this information can allow individuals to potentially buy cars with desired feature levels or even avoid buying cars with poor fuel economy.

These questions may be answered by using unsupervised and supervised learning methods.

Unsupervised Learning Analysis

Hierarchical Clustering Analysis

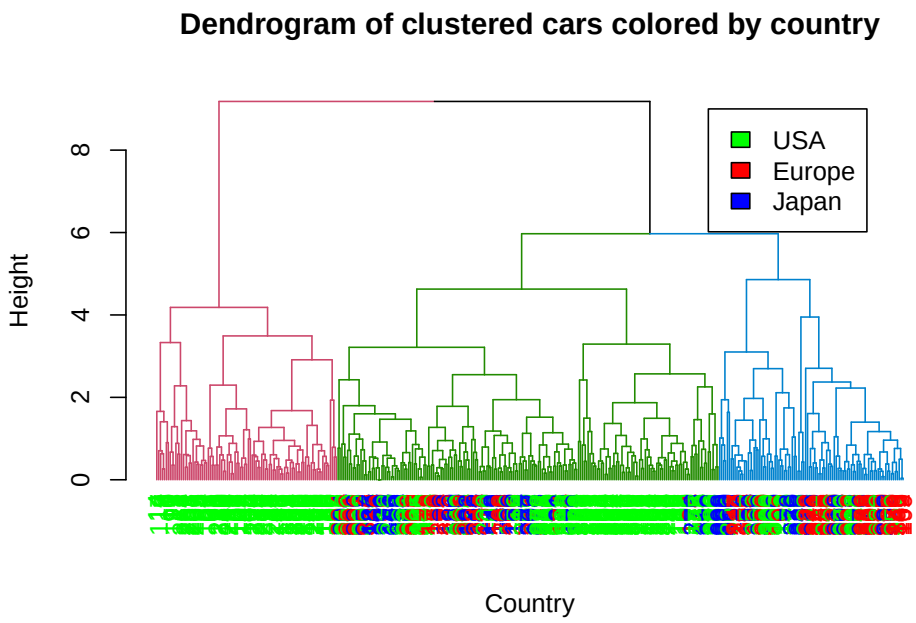


Figure 1: Hierarchical Clustering results

The numerical features were scaled to have standard deviation one and complete linkage was used to cluster the cars. From the dendrogram above, there seems to be three clusters. The cluster colored with red branches consists entirely of cars from the United States, while the other two clusters are mixed.

To gain better insight on the clusters, the mean of the numerical features was calculated for the clusters and within clusters. As seen from Table 1 of the clusters analysis, it appears that going from cluster 1 to cluster 3 generally leads to a decrease in all the features except acceleration with cluster 3 having the highest acceleration.

Table 1: Cluster Analysis

cluster	freq	mean.mpg	mean.displacement	mean.horsepower	mean.weight	mean.acceleration
1	95	14.46421	348.7895	162.4211	4150.474	12.58526
2	200	23.48550	165.0425	94.4700	2789.890	15.65550
3	97	32.16082	103.7732	68.3299	2215.876	18.20103

Table 2 below shows the within cluster analysis that provides insight on the countries within the clusters. As mentioned before, Cluster 1 consists of cars from USA and has the worst fuel economy. Cluster 2 consists of many cars from the US, while Cluster 3 is somewhat balanced with Japan having the highest count. Interestingly, it can be seen that lower mpg, displacement, horsepower and weight along with a higher acceleration seem to lead to a higher mpg value. Further analysis may confirm this trend.

Table 2: Within Cluster Analysis

cluster	origin	freq	mean.mpg	mean.displacement	mean.horsepower	mean.weight	mean.acceleration
1	USA	95	14.46421	348.7895	162.42105	4150.474	12.58526
2	Europe	42	25.13095	111.3571	90.09524	2451.071	15.02619
2	Japan	37	26.23243	116.5676	95.45946	2453.919	14.73243
2	USA	121	22.07438	198.5000	95.68595	3010.231	16.15620
3	Europe	26	31.59615	106.8462	65.15385	2405.038	19.65000
3	Japan	42	34.16667	90.5000	66.07143	2016.238	17.44048
3	USA	29	29.76207	120.2414	74.44828	2335.414	18.00345

Principal Component analysis

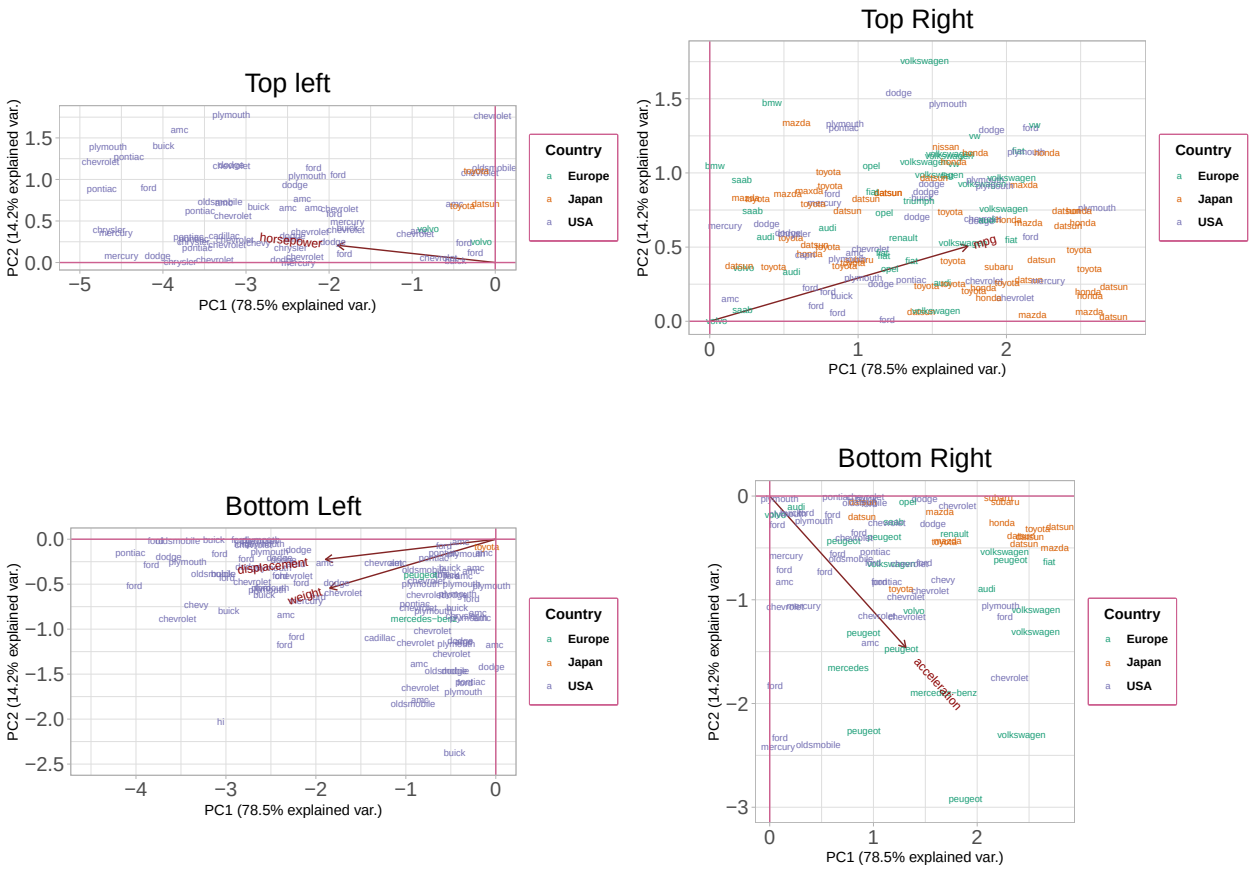


Figure 2: Biplot of PCA results

Table 3: Correlation Matrix

	mpg	displacement	horsepower	weight	acceleration
mpg	1.0000000	-0.8051269	-0.7784268	-0.8322442	0.4233285
displacement	-0.8051269	1.0000000	0.8972570	0.9329944	-0.5438005
horsepower	-0.7784268	0.8972570	1.0000000	0.8645377	-0.6891955
weight	-0.8322442	0.9329944	0.8645377	1.0000000	-0.4168392
acceleration	0.4233285	-0.5438005	-0.6891955	-0.4168392	1.0000000

For principal component analysis (PCA), all numerical features were scaled and then ran into the PCA algorithm to construct the plots below. The points are labeled by car brand. The whole plot was difficult to see, so it was broken down into four quadrants. *See the shiny app for a more interactive experience that includes zooming into the plots, pca variance, pca importance, explanation of PCA trend, etc..*

With principal components 1 and 2, about 92.7% of the variance can be explained. From the correlation matrix, we can see that mpg is strongly correlated with displacement, horsepower and weight. As a result, the loading values for these four variables were relatively similar in PC1 with absolute values ranging from .44 to .49, while acceleration being slightly correlated had a smaller loading value of .335 in terms of absolute value.

When interpreting the trend of the PCA biplots , loading values were considered along with whether a point was negative or positive. It was decided to have acceleration be explained by PC2. For PC1, $\frac{78.5}{5} = 15.7$, so if all the loading values were equal each one would contribute 15.7% of the variability. However due to acceleration's loading value the contributed variance is roughly 11%, whereas in the case of PC2 the contributed variability is about 12%. Putting this all together, the table below summarizes the general trend when looking at points that are positioned at a specific direction, such as bottom right.

Table 4: PCA interpretation

Position	MPG	Displacement	Horsepower	Weight	Acceleration
Top Right	Larger	Lower	Lower	Lower	Lower
Top Left	Lower	Larger	Larger	Larger	Larger
Bottom Right	Larger	Lower	Lower	Lower	Lower
Bottom Left	Lower	Larger	Larger	Larger	Larger

Supervised Learning Analysis

Appropriateness of a linear model and analysis

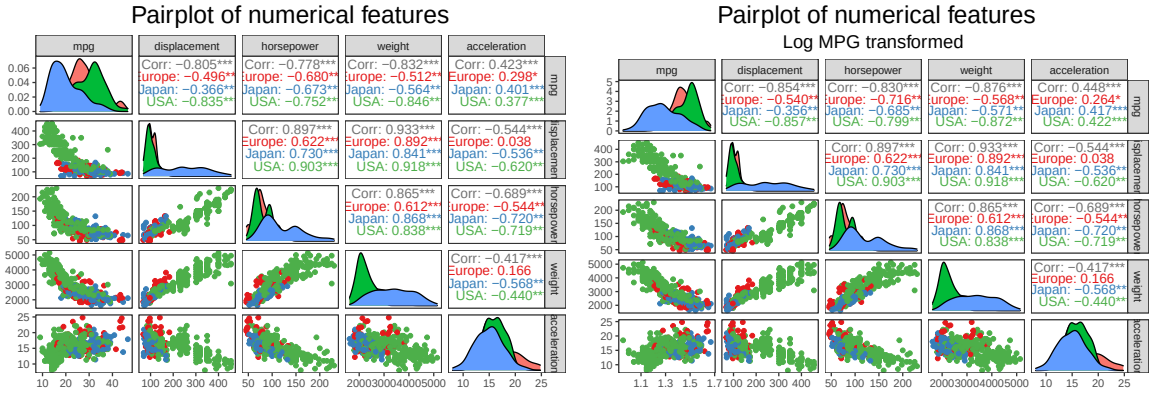


Figure 3: Pairplot of Numerical Features

The pairplots above display the the distribution and relationship of the numerical features. Looking at the graphs that pair mpg with the other features, the relationship is roughly linear for the first half but curves during the later half. Upon applying a log_{10} transformation on mpg, we can see from the transformed pairplot that the relationship begins to look more linear. Furthermore, the absolute value of the correlations are higher.

Next, all the numerical features along with the categorical features of cylinders and origin were fitted to a linear regression model that aims to predict mpg without any transformation (mpg is still in miles per gallon). In other words, the model is

$$\beta_0+\beta_1\text{cylinders6}+\beta_2\text{cylinders8}+\beta_3\text{displacement}+\beta_4\text{horsepower}+\beta_5\text{weight}+\beta_6\text{acceleration}+\beta_7\text{originJapan}+\beta_8\text{originUSA}$$

With this model, the residual standard error is 3.987 and the R^2 is .7444. However as we saw from the pairplots, it seems that the log_{10} transformation may lead to a better model.

Fitting all the features with the transformed mpg (log base 10 of miles per gallon), the residual standard error is .6509 $log_{10}(mpg)$ or 1.16 in mpg units, and the R^2 value is .8097. The log transformation definitely improved our linear model. However, from the pairplot of the log_{10} transformation, there still seems to be a slight curve. In an attempt to further improve the residual standard error and R^2 value, a combination of quadratic terms were added, such as $weight^2$+$horsepower^2$, $acceleration^2$+$weight^2$+$horsepower^2$, etc.

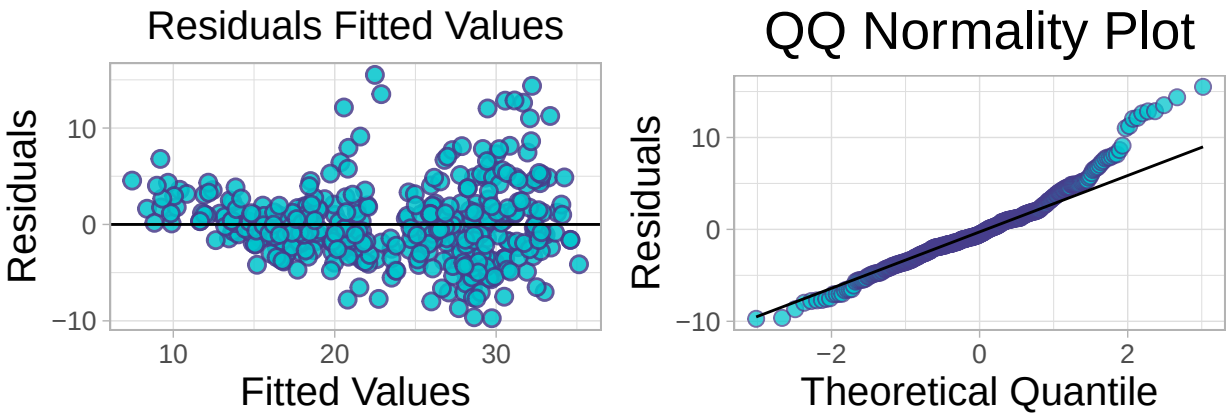
After testing many combinations, the model that was found to lead to the most improvement was

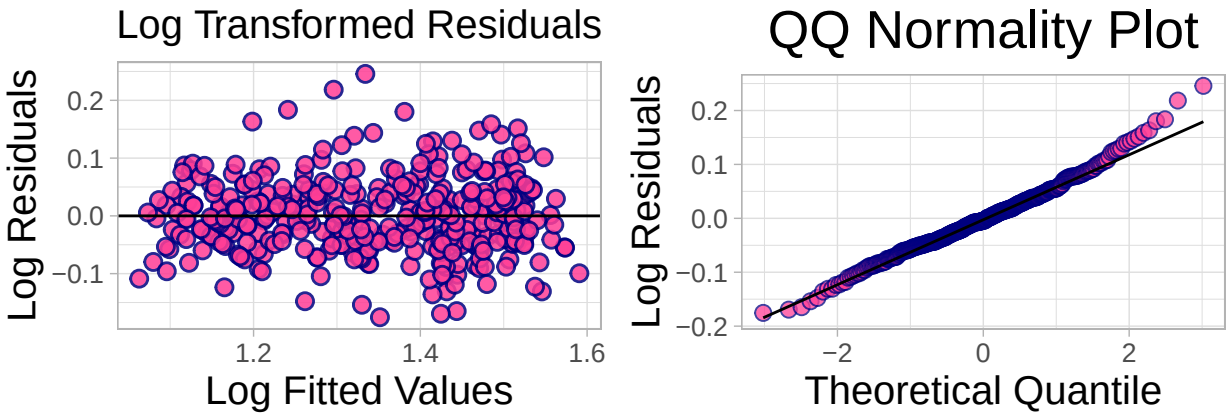
β_0+$\beta_1$cylinders6+$\beta_2$cylinders8+$\beta_3$displacement+$\beta_4$horsepower+$\beta_5$weight+$\beta_6$acceleration+$\beta_7$originJapan+$\beta_8$originUSA+β_9 $horsepower^2$

this model had a residual standard error of .06367 $log_{10}(mpg)$ or 1.158 mpg and a R^2 of .8179, meaning that 81.79% of the variation can be explained. Table 5 below shows the coefficients along with their significance.

Table 5: Summary of final linear model

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.9550754	0.0587021	33.3050114	0.0000000
cylinders6	-0.0466512	0.0142944	-3.2635908	0.0011992
cylinders8	-0.0339915	0.0250286	-1.3581036	0.1752324
displacement	-0.0000960	0.0001535	-0.6254368	0.5320579
horsepower	-0.0045125	0.0007376	-6.1176939	0.0000000
weight	-0.0000430	0.0000145	-2.9777216	0.0030892
acceleration	-0.0065358	0.0021114	-3.0954813	0.0021098
originJapan	0.0301615	0.0107609	2.8028835	0.0053230
originUSA	0.0050842	0.0110965	0.4581832	0.6470814
I(horsepower^2)	0.0000097	0.0000023	4.1386630	0.0000430





To asses the appropriateness of a linear model, residual plots and Q-Q plots were constructed to check for the assumptions of equal variance and normality. The plots with colored with turquoise refer to the model without the log transformation. The equal variance assumption seem to be violated, since heteroskedasticity seems to be present (the variance seems to be larger as we move further out along the x-axis). The normality assumption may also be violated, as points pass theoretical quantile 2 start to deviate from the line.

The plots colored with pink refer to the log transformed model, which includes the $horsepower^2$ term. The assumptions in this case do not seem to be heavily violated. From the residual vs fitted values plot, we can see that the variance is now roughly equal. The Q-Q plot does not seem to have points that extremely deviate from the theoretical line.

Now that we know a linear model is appropriate with the predictor variables cylinders, displacement, horsepower, weight, acceleration, origin and $horsepower^2$, we can move onto making a general predictive model.

Predictive model

Train-Test splitting

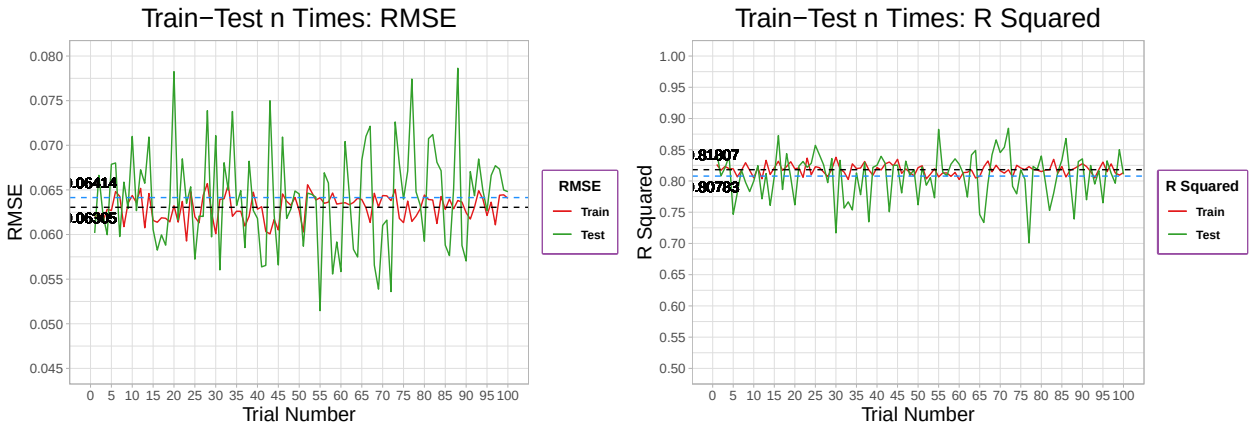


Figure 5: Train-Test split results n times

One thing to consider about a train-test split approach to building a predictive model is the randomness of the performance. To get a better idea of the randomness of different train-test split sizes, a custom function was built to test the performance of different split sizes as well as the performance for when certain predictor variables are removed. The plot above shows the output of the function,which is just a dataframe of the performance metrics, for an 80-20 split (20% for testing) for the full model. With an 80-20 split, it can be seen that sometimes the R^2 and RMSE values for the test set fluctuate a lot, where the testing performance is sometimes below the training set. The blue dashed line represents the average performance metrics for the test set across the repetitions, while the black dashed line is for the training set. With an 80-20 split, the average R^2 value fluctuates between .8 and .82, and the average RMSE fluctuates between .062 and .064 in $log_{10}(mpg)$ units. This is both for training and testing performance metrics. Due to the random nature of train-test splitting, and the difficulties of recording the model coefficients from the custom function, a k-fold cross validation was applied to assess prediction error and construct a final model. *See the shiny app's third tab to play around with the function and see how the performance metrics change*

K-fold Cross Validation

Using the `caret` package's train function, a 10x10-fold cross validation was applied to the transformed mpg dataset and the performance is given by table 6. Due to the randomness of the cross validation, the RMSE and R^2 values will not be exactly the same each time it is ran, but the value of RMSE seems to be roughly .064 $log_{10}(mpg)$, or 1.158 mpg, and the R^2 value seems to be roughly .81.

Table 6: 10x10-fold cross validation results

intercept	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
TRUE	0.0647293	0.813591	0.0501291	0.0073564	0.0398516	0.0056104

The table below is the coefficients that the cross validation produced, suggesting that with these coefficients we can obtain an average prediction error of 1.158 mpg and an average R^2 value of roughly 81%. *To play around with the predictive model, check out tab 2 of the shiny app*

Table 7: Final mode coefficients

	x
(Intercept)	1.9550754
cylinders6	-0.0466512
cylinders8	-0.0339915
displacement	-0.0000960
horsepower	-0.0045125
weight	-0.0000430
acceleration	-0.0065358
originJapan	0.0301615
originUSA	0.0050842
'I(horsepower^2)'	0.0000097

Interpretation of results

1. Is it appropriate to fit a linear model to this data? If so, what numerical variables have the most impact on mpg?

From the residual plots and QQ-plots on page 6, a linear model with the target variable transformed is appropriate, since the assumptions of equal variance and normality are not extremely violated. Furthermore, we can see from the transformed pair plot on page 5 that the relationship of the numerical features becomes more linear with the target variable, $log_{10}(mpg)$.

Table 5 on page 6 shows the significance of the coefficients/predictor variables. Horesepower, weight, acceleration, and $horsepower^2$ appear to be significant if $\alpha = .05$. Displacement, on the other hand, does not appear to be significant due to its p-value being .53. To further test the significance of displacement, stepwise regression was ran and removed displacement from the model. Additionally, the repeated train-test splitting interfaced on the shiny app also shows that removing displacement from the model hardly changes the performance.

2. Is it possible to build a predictive model with reasonable performance to predict a car's fuel economy?

From the repeated train-test splitting, it can be seen that an 80-20 split leads to a reasonable performance with the R^2 being .81 and the RMSE value being roughly .063 on average. Unfortunately, model coefficients were hard to record, and it is important to keep in mind the randomness of a single train-test split; performing just one train-test split and using those coefficients may end up being one of the poor- performance splits. With a 10x10-fold cross validation with the caret package, we were able to see that the out of prediction error is roughly 1.15 mpg and an R^2 value of roughly 81% is achieved using the coefficients in Table 7. Overall, the model performs well.

3. Are cars from these regions similar, or are they completely different? Is there a region that tends to make cars with good fuel economy? What about a region that produces cars with poor fuel economy?

From the hierarchical clustering analysis on page 3, cars across the regions are indeed similar as seen by the clustering. With the exception of cluster 1, cluster 2 and 3 are relatively mixed. Cluster 1 consists entirely of cars from the United states, and, as seen from table 1, has the worst fuel economy. Looking into the clusters using Table 2, it can also be seen that cars from the United States still have the worst fuel economy compared to that of Europe and Japan. However, it is important to note that cluster 2 is imbalanced with the United states having nearly four times the counts of Europe and Japan. In regards to the region that produces cars with good fuel economy, it appears to be Japan. Despite the imbalance in cluster 2, Japan has the highest average mpg. Cluster 3 is relatively balanced and Japan also has the highest average mpg value there too.

4. What brand of cars are similar in terms of fuel economy and other features such as weight?

From the biplot on page 4, and using the trend described with Table 4 on page 5, we can see that, similar to the hierarchical clustering, there seems to be a grouping of cars from the United States. The cars from the United states that lean more towards the left have poorer fuel economy and are all larger with regards to the numerical features in Table 4. The opposite is true for cars that lean more towards the right as described by the trend from Table 4. What probably interests us the most is the cars that are similar with regards to good fuel economy. Note that when interpreting the biplot, we define more positive values to be good fuel economy. since scores were standardized and the loading value for mpg was positive, so a positive standardized score for mpg means it was above the average.

As seen from the right side of the biplot, Japanese cars tend to have good fuel economy with Toyota, Datsun, Honda, etc.. being the best in terms of fuel economy. For European cars, Volkswagen models seem to be in a similar position. Interestingly, a few American car models also seem to be similar, such as Plymouth and Chevrolet, though they are not as frequent compared to Japan.

Conclusion

Using supervised learning and unsupervised learning techniques, we were able to draw interesting insights from our data. PCA and hierarchical clustering allowed us to discover that cars from the United states generally have poor fuel economy compared to Japan and Europe. Among the cars from the United States, Ford, Amc, Dodge, Chevrolet, and more American car brands had many models with poor fuel economy. Japan, on the other hand, was found to make cars with good fuel economy with the Toyota, Datsun, and Honda brands being the most notable. Europe's volkswagen models were also similar.

Using all the features of the dataset to predict mpg, a log transformation lead to better performance along with the addition of the $horsepower^2$ term. Despite the randomness of the train-test splitting, cross validation enabled us to construct a model that had an out of prediction error of 1.15 mpg and an R^2 of 81%. However, is it important to take into account that the dataset was only 398 rows long, which is quite small. With more data the model may be improved such that the RMSE value decreases and the R^2 value increases.

Overall, a small dataset was a limitation of this analysis. Despite this dataset being old, it shows how the United States has a history of contributing to the emissions of greenhouse gases, and poorly manufactured automobiles is reflective of that. Today, the United States continues to be a major contributor [5]. Perhaps the United States should look at the American models similar to Japan and Europe and use those as inspiration to build better cars, or import more cars from Japan and Europe and produced less American cars.

References

1. Transportation Technologies and Innovation. Union of Concerned Scientists. (n.d.). Retrieved November 12, 2021, from https://www.ucsusa.org/transportation/technologies.
2. McGregor, H. V., Gergis, J., Abram, N. J., & Phipps, S. J. (2016). The Industrial Revolution kick-started global warming much earlier than we realised.
3. Learning, U. C. I. M. (2017, July 2). Auto-mpg dataset. Kaggle. Retrieved November 12, 2021, from https://www.kaggle.com/uciml/autompg-dataset.
4. Rdocumentation: https://www.rdocumentation.org/
5. Althor, G., Watson, J. E., & Fuller, R. A. (2016). Global mismatch between greenhouse gas emissions and the burden of climate change. Scientific reports, 6(1), 1-6.

Appendix: R code used

```
#global options
# keeps this here to remove comments from knitted output.
knitr::opts_chunk$set(comments=NA)
knitr::opts_chunk$set(echo=F)
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
library(tidyverse)
library(knitr)
library(leaps)
library(GGally)
library(ggbiplot)
library(caret)
library(RColorBrewer)
library(dendextend)
library(cowplot)
library(kableExtra)
library(ggpubr)
# data cleaning up
data <- read.csv('auto-mpg.csv')
#convert horsepower chr->dbl
data$horsepower <- as.numeric(data$horsepower)
#remove rows with missing values
data <- na.omit(data)
#translate origin numbers to country strings
data$origin <- ifelse(data$origin==1,"USA",ifelse(data$origin==2,"Europe","Japan"))
data$origin <- as.factor(data$origin)
#cylinders count for 3 and 5 low combine with 4 and 6 respectively
data$cylinders <- replace(data$cylinders,data$cylinders %in% c(3,5),c(4,6))
data$cylinders <- as.factor(data$cylinders)
#remove model.year, not interested in this feature
data <- data[-c(7)]
data$car.name <- word(data$car.name,1)
#car.name fix typos
data$car.name[160] <- "chevrolet"
data$car.name[330] <- "volkswagen"
data$car.name[82] <- "toyota"
h.clustering.complete <- hclust(dist(scale(data[-c(2,7,8)])),method="complete") %>% as.dendrogram() %>%
color.order <- as.numeric(data$origin)
colors.dendro <- color.order[order.dendrogram(h.clustering.complete)]
colors.dendro <- ifelse(colors.dendro==3,"green",ifelse(colors.dendro==2,"red","blue"))
labels_colors(h.clustering.complete) <- colors.dendro
h.clustering.complete <- h.clustering.complete %>% set("labels_col",colors.dendro)
plot(h.clustering.complete,
     main="Dendrogram of clustered cars colored by country",
     ylab="Height",
     xlab="Country")
legend(x=290,y=9,legend=c("USA","Europe","Japan"),fill=c("green","red","blue"))
hclust.data <- data.frame(data,cluster=cutree(h.clustering.complete,h=5))
hclust.data.clusters <- data.frame(data,cluster=cutree(h.clustering.complete,h=5)) %>% count(c('cluster
hclust.in.clusters <- data.frame(data,cluster=cutree(h.clustering.complete,h=5)) %>% count(c('cluster',

#cluster analysis
c.1 <- hclust.data %>% filter(cluster==1 ) %>% summarise(mean.mpg=mean(mpg),
```

```
                                                                    mean.displacement=mean(displacement)
                                                                    mean.horsepower=mean(horsepower),
                                                                    mean.weight=mean(weight),
                                                                    mean.acceleration=mean(acceleration)
                                                                     )

c.2 <- hclust.data %>% filter(cluster==2 ) %>% summarise(mean.mpg=mean(mpg),
                                                                    mean.displacement=mean(displacement)
                                                                    mean.horsepower=mean(horsepower),
                                                                    mean.weight=mean(weight),
                                                                    mean.acceleration=mean(acceleration)
                                                                     )

c.3 <- hclust.data %>% filter(cluster==3 ) %>% summarise(mean.mpg=mean(mpg),
                                                                    mean.displacement=mean(displacement)
                                                                    mean.horsepower=mean(horsepower),
                                                                    mean.weight=mean(weight),
                                                                    mean.acceleration=mean(acceleration)
                                                                     )

clusters.data <- rbind(c.1,c.2,c.3)
hclust.data.clusters <- cbind(hclust.data.clusters,clusters.data)

# Within cluster analysis
US.1 <-  hclust.data %>% filter(cluster==1 & origin=="USA") %>% summarise(mean.mpg=mean(mpg),
                                                                    mean.displacement=mean(displacement)
                                                                    mean.horsepower=mean(horsepower),
                                                                    mean.weight=mean(weight),
                                                                    mean.acceleration=mean(acceleration)
                                                                     )
EU.2 <-  hclust.data %>% filter(cluster==2 & origin=="Europe") %>% summarise(mean.mpg=mean(mpg),
                                                                    mean.displacement=mean(displacement)
                                                                    mean.horsepower=mean(horsepower),
                                                                    mean.weight=mean(weight),
                                                                    mean.acceleration=mean(acceleration)
                                                                     )
JN.2 <-  hclust.data %>% filter(cluster==2 & origin=="Japan") %>% summarise(mean.mpg=mean(mpg),
                                                                    mean.displacement=mean(displacement)
                                                                    mean.horsepower=mean(horsepower),
                                                                    mean.weight=mean(weight),
                                                                    mean.acceleration=mean(acceleration)
                                                                     )
US.2 <-  hclust.data %>% filter(cluster==2 & origin=="USA") %>% summarise(mean.mpg=mean(mpg),
                                                                    mean.displacement=mean(displacement)
                                                                    mean.horsepower=mean(horsepower),
                                                                    mean.weight=mean(weight),
                                                                    mean.acceleration=mean(acceleration)
                                                                     )
EU.3 <-  hclust.data %>% filter(cluster==3 & origin=="Europe") %>% summarise(mean.mpg=mean(mpg),
                                                                    mean.displacement=mean(displacement)
                                                                    mean.horsepower=mean(horsepower),
                                                                    mean.weight=mean(weight),
                                                                    mean.acceleration=mean(acceleration)
```

```
                                                                    )
JN.3 <-  hclust.data %>% filter(cluster==3 & origin=="Japan") %>% summarise(mean.mpg=mean(mpg),
                                                                    mean.displacement=mean(displacement)
                                                                    mean.horsepower=mean(horsepower),
                                                                    mean.weight=mean(weight),
                                                                    mean.acceleration=mean(acceleration)
                                                                    )
US.3 <-  hclust.data %>% filter(cluster==3 & origin=="USA") %>% summarise(mean.mpg=mean(mpg),
                                                                    mean.displacement=mean(displacement)
                                                                    mean.horsepower=mean(horsepower),
                                                                    mean.weight=mean(weight),
                                                                    mean.acceleration=mean(acceleration)
                                                                    )
in.clusters.data <- rbind(US.1,EU.2,JN.2,US.2,EU.3,JN.3,US.3)
hclust.in.clusters <- cbind(hclust.in.clusters,in.clusters.data)
kable(hclust.data.clusters,format="latex",booktabs=T,longtable=T,caption = "Table 1: Cluster Analysis")
#hclust.data.clusters
#hclust.in.clusters
kable(hclust.in.clusters,format="latex",booktabs=T,longtable=T,caption="Table 2: Within Cluster Analysis
pcs.out <- prcomp(data[-c(2,7,8)],scale.=T)
pcs.dat <- data.frame(rownames(pcs.out$rotation),pcs.out$rotation)
colnames(pcs.dat)[1] <- "Features"
pcs.importance <- data.frame(summary(pcs.out)[6])
pcs.importance <- cbind(c("Standard deviation","Proportion of Variance","Cumulative Proportion"),pcs.imp
colnames(pcs.importance) <- c("Metrics","PC1","PC2","PC3","PC4","PC5")
cols <- brewer.pal(3, "Dark2")
PCA.interp <- data.frame(Position=c("Top Right","Top Left","Bottom Right","Bottom Left"),
           MPG=c("Larger","Lower","Larger","Lower"),
           Displacement=c("Lower","Larger","Lower","Larger"),
           Horsepower=c("Lower","Larger","Lower","Larger"),
           Weight=c("Lower","Larger","Lower","Larger"),
           Acceleration=c("Lower","Larger","Lower","Larger"))
corr.matrix <- cor(data[-c(2,7,8)])

 ggbiplot(pcs.out,labels = data$car.name,groups=data$origin,obs.scale = 1,labels.size = 2.3)+
  geom_hline(yintercept = 0,col="hotpink3")+
  geom_vline(xintercept = 0,col="hotpink3")+
  ylim(0,1.8)+
  xlim(-5,0)+
  theme_light()+
  theme(plot.title=element_text(hjust=.5,size=20),
        axis.text = element_text(size=15)
        )+
  labs(title="Top left",
       )+
  scale_color_manual(values=cols)+ theme(legend.box.background = element_rect(linetype="solid", colour =
        legend.title = element_text(face="bold", hjust = .5),
        legend.text = element_text(face="bold"))+
  guides(colour=guide_legend("Country"))



 ggbiplot(pcs.out,labels = data$car.name,groups=data$origin,obs.scale = 1,labels.size = 2.3)+
```

```
  geom_hline(yintercept = 0,col="hotpink3")+
  geom_vline(xintercept = 0,col="hotpink3")+
  ylim(0,1.8)+
  xlim(0,2.8)+
  theme_light()+
  theme(plot.title=element_text(hjust=.5,size=20),
        axis.text = element_text(size=15)
        )+
  labs(title="Top Right",
       )+
  scale_color_manual(values=cols)+ theme(legend.box.background = element_rect(linetype="solid", colour =
        legend.title = element_text(face="bold", hjust = .5),
        legend.text = element_text(face="bold"))+
  guides(colour=guide_legend("Country"))


 ggbiplot(pcs.out,labels = data$car.name,groups=data$origin,obs.scale = 1,labels.size = 2.3)+
  geom_hline(yintercept = 0,col="hotpink3")+
  geom_vline(xintercept = 0,col="hotpink3")+
  ylim(-2.5,0)+
  xlim(-4.5,0)+
  theme_light()+
  theme(plot.title=element_text(hjust=.5,size=20),
        axis.text = element_text(size=15)
        )+
  labs(title="Bottom Left",
       )+
    scale_color_manual(values=cols)+
  theme(legend.box.background = element_rect(linetype="solid", colour ="hotpink3", size=1.25),
        legend.title = element_text(face="bold", hjust = .5),
        legend.text = element_text(face="bold"))+
  guides(colour=guide_legend("Country"))


 ggbiplot(pcs.out,labels = data$car.name,groups=data$origin,obs.scale = 1,labels.size = 2.3)+
  geom_hline(yintercept = 0,col="hotpink3")+
  geom_vline(xintercept = 0,col="hotpink3")+
  ylim(-3,0)+
  xlim(0,2.8)+
  theme_light()+
  theme(plot.title=element_text(hjust=.5,size=20),
        axis.text = element_text(size=15)
        )+
  labs(title="Bottom Right",
       )+
    scale_color_manual(values=cols)+
   theme(legend.box.background = element_rect(linetype="solid", colour ="hotpink3", size=1.25),
        legend.title = element_text(face="bold", hjust = .5),
        legend.text = element_text(face="bold"))+
  guides(colour=guide_legend("Country"))

kable(corr.matrix,format="latex",booktabs=T,longtable=T,caption="Table 3: Correlation Matrix") %>% kabl
kable(PCA.interp,format="latex",booktabs=T,longtable=T,caption="Table 4: PCA interpretation") %>% kable
```

```r
data.transformed <- data
data.transformed$mpg <- log(data.transformed$mpg,base=10)
data.transformed <- data.transformed[-c(8)]
ggpairs(data[-c(2,7,8)],aes(color=data$origin))+
  theme_bw()+
  theme(panel.grid=element_blank(),
        plot.title=element_text(hjust=.5,size=20)) +
  labs(title="Pairplot of numerical features")+
  scale_color_manual(values=brewer.pal(3,"Set1"))

ggpairs(data.transformed[-c(2,7,8)],aes(color=data$origin))+
  theme_bw()+
  theme(panel.grid=element_blank(),
        plot.title=element_text(hjust=.5,size=20),
        plot.subtitle =element_text(hjust=.5,size=15)) +
  labs(title=" Pairplot of numerical features",
       subtitle = "Log MPG transformed")+
  scale_color_manual(values=brewer.pal(3,"Set1"))
lr.data <- lm(mpg~.,data=data[-c(8)])
#residuals vs fitted plot
p1 <- ggplot(lr.data)+
  theme_light()+
  labs(title = "Residuals Fitted Values",x="Fitted Values",y="Residuals")+
  geom_point(aes(x=lr.data$fitted.values,y=lr.data$residuals),col="darkslateblue",pch=21,fill="turquois
  geom_hline(yintercept = 0)+
  theme(axis.title = element_text(size=15),
        axis.text = element_text(size=10),
        plot.title = element_text(hjust = .5, size = 15))

p2 <- ggplot(lr.data,aes(sample=lr.data$residuals))+
  labs(title = "QQ Normality Plot",x="Theoretical Quantile",y="Residuals")+
  theme_light()+
  stat_qq(col="darkslateblue",pch=21,fill="turquoise3",alpha=.75,size=2.5,stroke=0.5)+
  geom_qq_line()+
  theme(axis.title = element_text(size=15),
        axis.text = element_text(size=10),
        plot.title = element_text(hjust = .5, size = 20))

log.lr.data <- lm(mpg~.+I(horsepower^2),data=data.transformed)
log.coef <- summary(log.lr.data)$coef
#residuals vs fitted plot
p1a <- ggplot(log.lr.data)+
  labs(title = "Log Transformed Residuals", x = "Log Fitted Values", y = "Log Residuals")+
  theme_light()+
  geom_point(aes(x=log.lr.data$fitted.values,y=log.lr.data$residuals),col="navyblue",pch=21,fill="viole
  geom_hline(yintercept = 0)+
  theme(axis.title = element_text(size=15),
        axis.text = element_text(size=10),
        plot.title = element_text(hjust = .5, size = 15))


p2a <- ggplot(log.lr.data,aes(sample=log.lr.data$residuals))+
  labs(title = "QQ Normality Plot",x="Theoretical Quantile",y="Log Residuals")+
```

```
  theme_light()+
  stat_qq(col="navyblue",pch=21,fill="violetred1",alpha=.75,size=2.5,stroke=0.5)+
  geom_qq_line()+
  theme(axis.title = element_text(size=15),
        axis.text = element_text(size=10),
        plot.title = element_text(hjust = .5, size = 20))
kable(log.coef,format="latex",caption="Table 5: Summary of final linear model") %>% kable_styling(font_
plot_grid(p1, p2, p1a, p2a)
library(MASS)
step.model <- stepAIC(log.lr.data, direction = "both",
trace = FALSE)
summary(step.model)

train.test <- function(data,split.size){
  #randomize the data
  randomized.rows <- sample(nrow(data))
  randomized.data <- data[randomized.rows,]
  #split based on desired size
  split <- round(nrow(randomized.data)*split.size)
  train <- randomized.data[1:split,]
  test <- randomized.data[(split+1):nrow(randomized.data),]
  return(list(train,test))
}

#computes the Rsquared and MSE
model.metrics <- function(predicted,actual,data){
  SSE <- sum((predicted-actual)^2)
  SSTO <- sum((actual-mean(actual))^2)
  R.squared <- 1-(SSE/SSTO)
  R.MSE <- sqrt(SSE/nrow(data))
  results <- c(R.MSE,R.squared)
  names(results) <- c("RMSE","R.squared")
  return(results)
}

#From the full model:mpg~.+I(horsepower^2), specify what features to remove
build.model.features <- function(data,feats="None"){
  if(sum(!feats%in%"None")!=0) {
  #input validation
  if(sum(!feats %in% colnames(data))!=0){
    return("Error: No Such feature(s)")
  }
  features <- as.formula(paste("mpg~.+I(horsepower^2)-",paste(feats,collapse= "-")))
  return(features)

  }
  else return(as.formula(paste("mpg~.+I(horsepower^2)")))
}

# Combines usage of build.model.features and model.metrics to simulate a train-test split evaluation
build.and.evaluate <- function(data,split.size,feats="None"){
  #train-test split
  train <- train.test(data,split.size)[[1]]
```

```
  test <- train.test(data,split.size)[[2]]
  #build model
  model <- lm(build.model.features(data,feats),train)
  print(build.model.features(data,feats))
  #predict on test set
  p.train <- predict(model,train)
  p.test <- predict(model,test)
  #evaluate model
  metric.results <- c(model.metrics(p.train,train$mpg,train),
                      model.metrics(p.test,test$mpg,test))
  names(metric.results) <- c("Train.RMSE","Train.R.Squared","Test.RMSE","Test.R.Squared")
  return(metric.results)
}

# Runs build and evaluate n times and returns a dataframe of the results
n.build.and.evaluate <- function(n,data,split.size,feats="None"){
  df <- data.frame(matrix(ncol=4,nrow = 0))
  for(i in 1:n){
    metric.results <- build.and.evaluate(data,split.size,feats)
    df <- rbind(df,metric.results)
  }
  df <- cbind(1:n,df)
  colnames(df) <- c("Trial.number","Train.RMSE","Train.R.Squared","Test.RMSE","Test.R.Squared")
  return(df)
}

b <- n.build.and.evaluate(100,data.transformed,.8)
avg.b.RMSE <- round(mean(b$Test.RMSE),5)
avg.b.Rsq <- round(mean(b$Test.R.Squared),5)
avg.tr.RMSE <- round(mean(b$Train.RMSE),5)
avg.tr.Rsq <- round(mean(b$Train.R.Squared),5)
ggplot(data=b,aes(x=Trial.number))+
  labs(title="Train-Test n Times: RMSE", x="Trial Number", y="RMSE")+
  theme_light()+
  geom_line(aes(y=Train.RMSE,col="Train.RMSE"))+
  geom_line(aes(y=Test.RMSE,col="Test.RMSE"))+
  coord_cartesian(xlim=c(0,100),ylim=c(0.045,.08))+
  scale_x_continuous(breaks=seq(0,100,5))+
  scale_y_continuous(breaks=seq(0.045,0.08,0.005))+
  scale_color_manual(values = c(Train.RMSE="#E31A1C",Test.RMSE="#33A02C"), labels = c("Train", "Test"))+
  theme(legend.box.background = element_rect(linetype="solid", colour ="#984EA3", size=1.25),
        legend.title = element_text(face="bold", hjust = .5),
        legend.text = element_text(face="bold"),
        panel.grid.minor.x = element_blank(),
        axis.title = element_text(size=15),
        axis.text = element_text(size=10),
        plot.title = element_text(hjust = .5, size = 20))+
  guides(colour=guide_legend("RMSE"))+
  geom_hline(yintercept = avg.b.RMSE,col='dodgerblue',linetype='dashed')+
  geom_text(aes(0,avg.b.RMSE,label = avg.b.RMSE, vjust = -.9))+
  geom_hline(yintercept = avg.tr.RMSE,col='black',linetype='dashed')+
  geom_text(aes(0,avg.tr.RMSE,label = avg.tr.RMSE, vjust = 1.5))
```

```
ggplot(data=b,aes(x=Trial.number))+
  labs(title="Train-Test n Times: R Squared", x="Trial Number", y="R Squared")+
  theme_light()+
  geom_line(aes(y=Train.R.Squared,col="Train.R.Squared"))+
  geom_line(aes(y=Test.R.Squared,col="Test.R.Squared"))+
  scale_color_manual(values = c(Train.R.Squared="#E31A1C",Test.R.Squared="#33A02C"), labels = c("Train"
  coord_cartesian(xlim=c(0,100),ylim=c(.5,1))+
  scale_x_continuous(breaks=seq(0,100,5))+
  scale_y_continuous(breaks=seq(.5,1,0.05))+
  theme(legend.position="right",
    legend.box.background = element_rect(linetype="solid", colour ="#984EA3", size=1.25),
        legend.title = element_text(face="bold", hjust = .5),
        legend.text = element_text(face="bold"),
        panel.grid.minor.x = element_blank(),
        axis.title = element_text(size=15),
        axis.text = element_text(size=10),
        plot.title = element_text(hjust = .5, size = 20))+
  guides(colour=guide_legend("R Squared"))+
  geom_hline(yintercept = avg.b.Rsq,col='dodgerblue',linetype='dashed')+
  geom_text(aes(0,avg.b.Rsq,label = avg.b.Rsq, vjust = 1.5))+
  geom_hline(yintercept = avg.tr.Rsq,col='black',linetype='dashed')+
  geom_text(aes(0,avg.tr.Rsq,label = avg.tr.Rsq, vjust = -.9))

k.fold.model <- train(
  build.model.features(data.transformed),
  data.transformed,
  method = "lm",
  trControl = trainControl(
    method = "repeatedcv",
    number = 10,
    repeats = 10,
    verboseIter = TRUE
  )
)
kable(k.fold.model$results,format="latex",caption="Table 6: 10x10-fold cross validation results ") %>% k
kable(k.fold.model$finalModel$coefficients,format="latex",caption="Table 7: Final mode coefficients ") %
```