Project

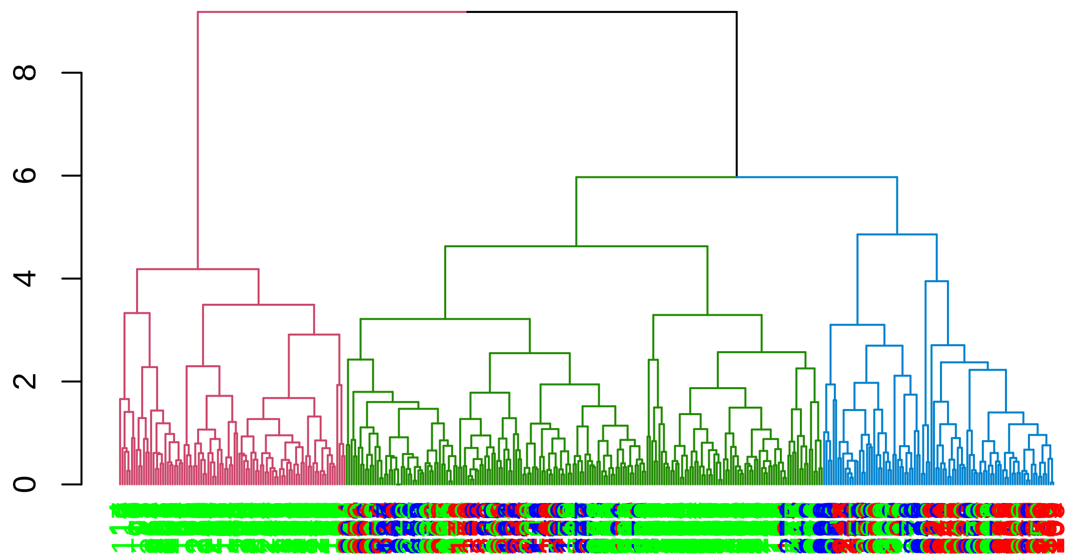
Brandon Hom

11/2/2021

Contents

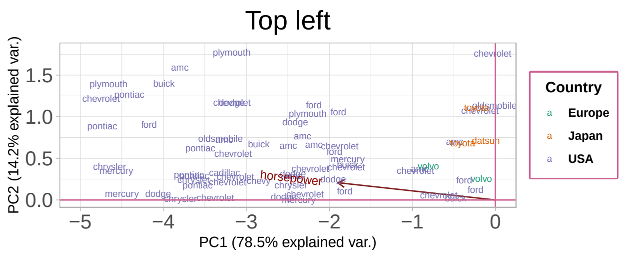
Introduction

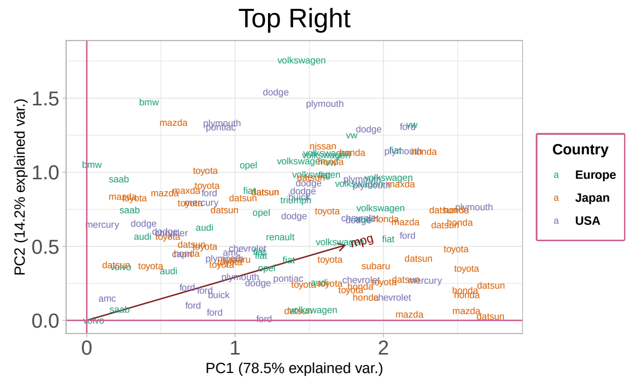
Exploratory data analysis

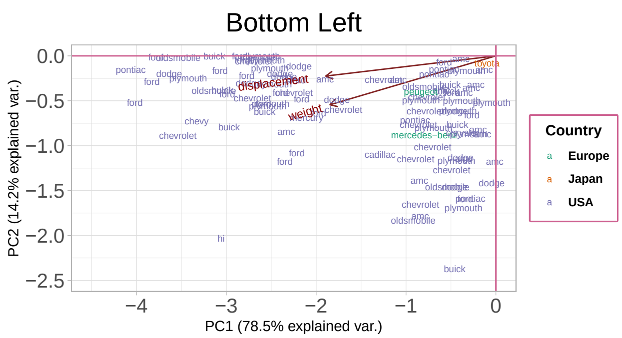


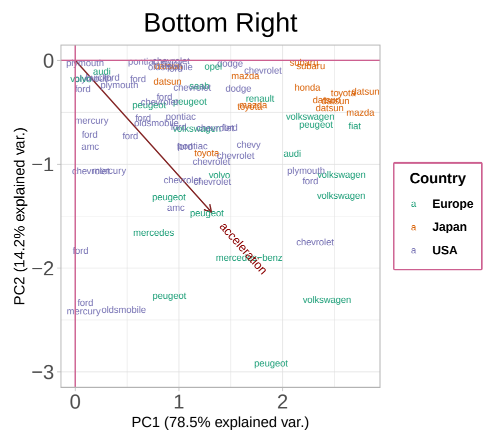
cluster	freq	mean.displacement	mean.horsepower	mean.weight	mean.acceleration
1	95	348.7895	162.4211	4150.474	12.58526
2	200	165.0425	94.4700	2789.890	15.65550
3	97	103.7732	68.3299	2215.876	18.20103

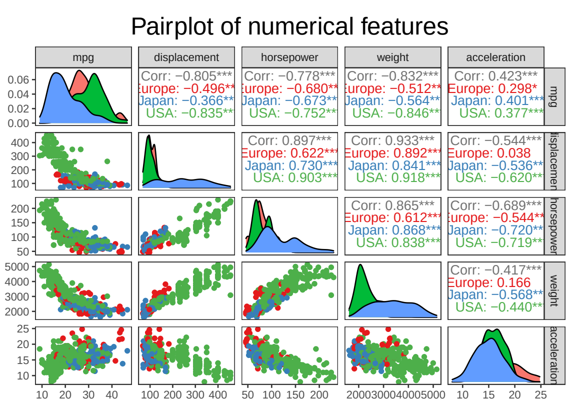
cluster	origin	freq	mean.displacement	mean.horsepower	mean.weight	mean.acceleration
1	USA	95	348.7895	162.42105	4150.474	12.58526
2	Europe	42	111.3571	90.09524	2451.071	15.02619
2	Japan	37	116.5676	95.45946	2453.919	14.73243
2	USA	121	198.5000	95.68595	3010.231	16.15620
3	Europe	26	106.8462	65.15385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

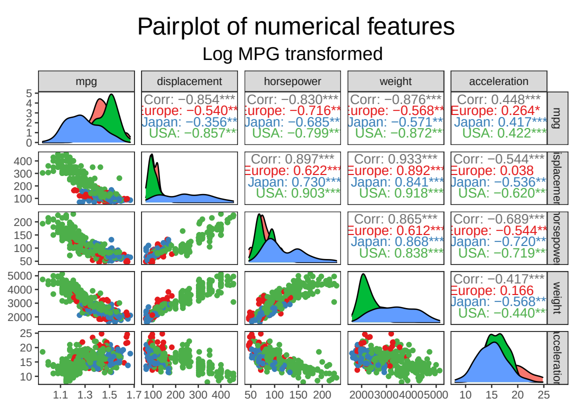








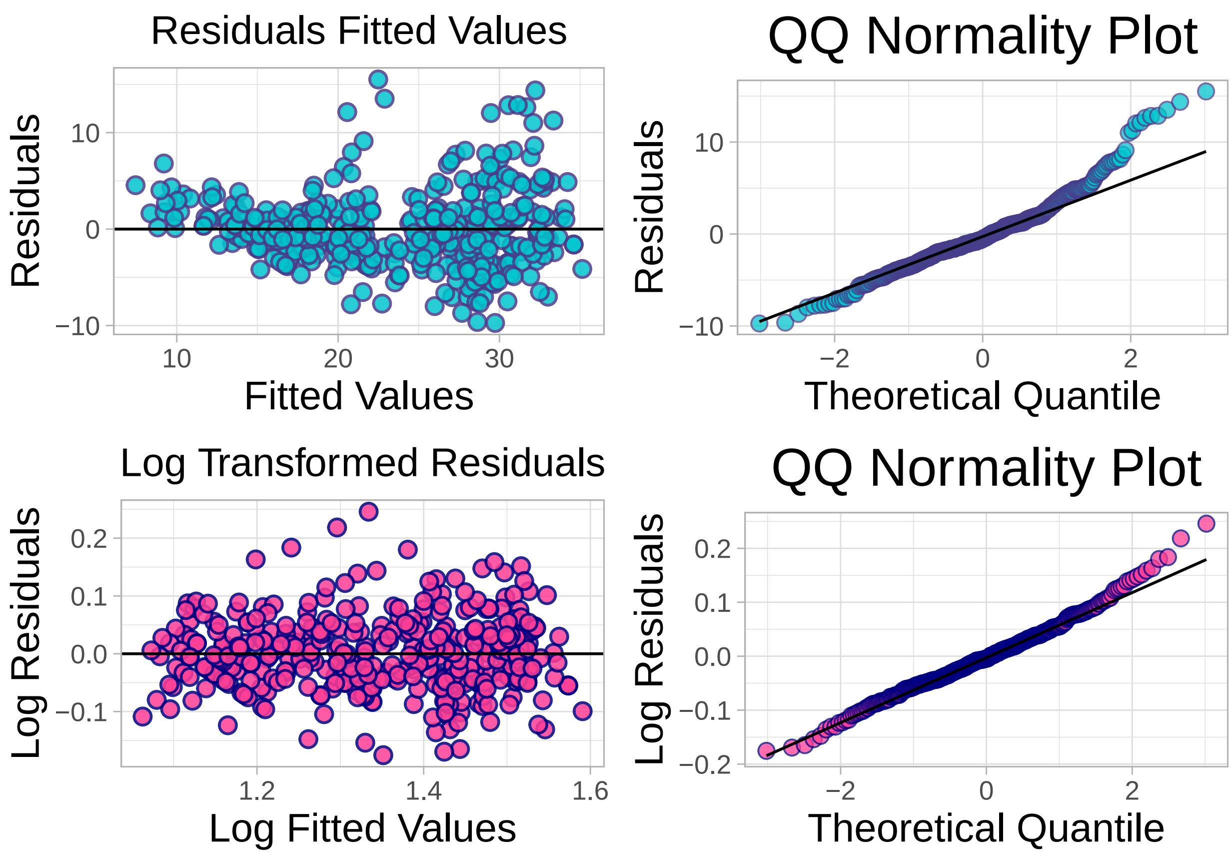





```
##
## Call:
## lm(formula = mpg ~ ., data = data[-c(8)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7287 -2.3413 -0.5307  1.7955 15.5121
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  43.9236266  2.4378221  18.018  < 2e-16 ***
## cylinders6   -4.0784024  0.8358754  -4.879 1.57e-06 ***
```

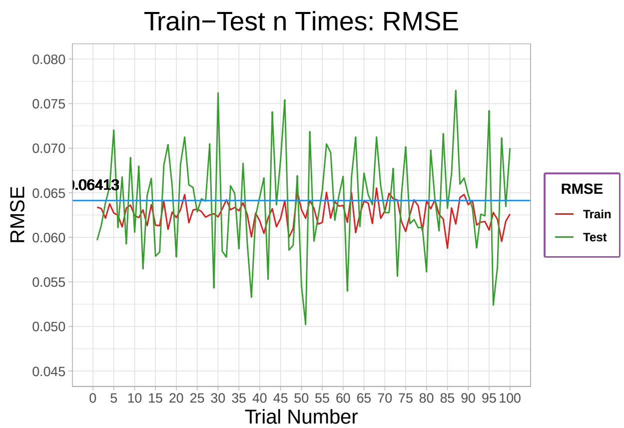
```
## cylinders8   -2.1875310  1.5136786  -1.445  0.14923
## displacement  0.0125671  0.0087680   1.433  0.15259
## horsepower   -0.0822381  0.0165086  -4.982 9.57e-07 ***
## weight       -0.0041588  0.0007832  -5.310 1.86e-07 ***
## acceleration -0.0353203  0.1183637  -0.298  0.76556
## originJapan   1.9720779  0.6722152   2.934  0.00355 **
## originUSA    -0.5203577  0.6810074  -0.764  0.44528
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.987 on 383 degrees of freedom
## Multiple R-squared:  0.7444, Adjusted R-squared:  0.739
## F-statistic: 139.4 on 8 and 383 DF,  p-value: < 2.2e-16
```

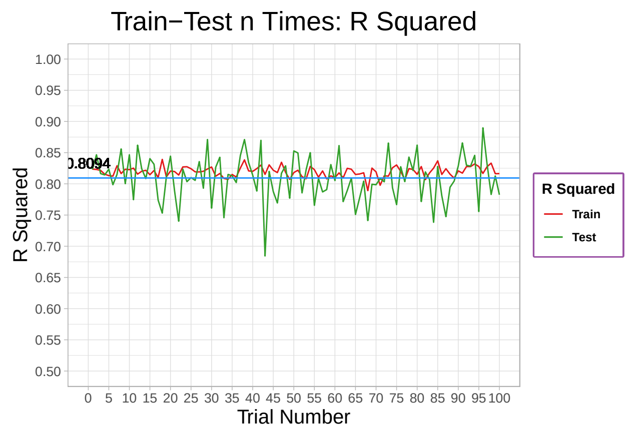
```
##
## Call:
## lm(formula = mpg ~ . + I(horsepower^2), data = data.transformed)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.175448 -0.043319 -0.004396  0.037993  0.245763
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.955e+00  5.870e-02  33.305  < 2e-16 ***
## cylinders6      -4.665e-02  1.429e-02  -3.264  0.00120 **
## cylinders8      -3.399e-02  2.503e-02  -1.358  0.17523
## displacement    -9.599e-05  1.535e-04  -0.625  0.53206
## horsepower      -4.513e-03  7.376e-04  -6.118 2.35e-09 ***
## weight          -4.304e-05  1.445e-05  -2.978  0.00309 **
## acceleration    -6.536e-03  2.111e-03  -3.095  0.00211 **
## originJapan      3.016e-02  1.076e-02   2.803  0.00532 **
## originUSA        5.084e-03  1.110e-02   0.458  0.64708
## I(horsepower^2)  9.702e-06  2.344e-06   4.139 4.30e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06376 on 382 degrees of freedom
## Multiple R-squared:  0.8179, Adjusted R-squared:  0.8136
## F-statistic: 190.6 on 9 and 382 DF,  p-value: < 2.2e-16
```




```
##
## Call:
## lm(formula = mpg ~ cylinders + horsepower + weight + acceleration +
##     origin + I(horsepower^2), data = data.transformed)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.178942 -0.043369 -0.004149  0.037136  0.240756
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.945e+00  5.626e-02  34.565  < 2e-16 ***
## cylinders6      -5.173e-02  1.175e-02  -4.403 1.39e-05 ***
## cylinders8      -4.401e-02  1.921e-02  -2.291 0.022492 *
## horsepower      -4.381e-03  7.066e-04  -6.201 1.46e-09 ***
## weight          -4.780e-05  1.228e-05  -3.892 0.000117 ***
## acceleration    -6.213e-03  2.046e-03  -3.037 0.002552 **
## originJapan      2.999e-02  1.075e-02   2.790 0.005529 **
## originUSA        2.418e-03  1.024e-02   0.236 0.813393
## I(horsepower^2)  9.106e-06  2.140e-06   4.255 2.63e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06371 on 383 degrees of freedom
## Multiple R-squared:  0.8177, Adjusted R-squared:  0.8139
## F-statistic: 214.7 on 8 and 383 DF,  p-value: < 2.2e-16
```

Predictive model





Cross-Fold validation

```
## Linear Regression
##
## 392 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 352, 353, 353, 353, 352, 352, ...
## Resampling results:
##
##   RMSE        Rsquared   MAE
##   0.06457377  0.8122216  0.05006974
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
##   cylinders displacement horsepower weight acceleration origin
## 1         8           80         69   2020           19    USA
```

```
##        1
## 27.68199
```

```
##   6 8
## 4 0 0
## 6 1 0
## 8 0 1
```

```
##        Japan USA
## Europe     0   0
## Japan      1   0
## USA        0   1
```

Conclusion

Appendix: R code used

```
#global options
# keeps this here to remove comments from knitted output.
```

```r
knitr::opts_chunk$set(comments=NA)
knitr::opts_chunk$set(echo=F)
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
library(tidyverse)
library(knitr)
library(leaps)
library(GGally)
library(ggbiplot)
library(caret)
library(RColorBrewer)
library(dendextend)
library(cowplot)
library(kableExtra)
# data cleaning up
data <- read.csv('auto-mpg.csv')
#convert horsepower chr->dbl
data$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)
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.displacement=mean(displacement),
                                                           mean.horsepower=mean(horsepower),
                                                           mean.weight=mean(weight),
                                                           mean.acceleration=mean(acceleration)
                                                            )

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

```r
                                                             mean.acceleration=mean(acceleration)
                                                              )

c.3 <- hclust.data %>% filter(cluster==3 ) %>% summarise(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.displacement=mean(displac
                                                             mean.horsepower=mean(horsepower),
                                                             mean.weight=mean(weight),
                                                             mean.acceleration=mean(acceleration)
                                                              )
EU.2 <-  hclust.data %>% filter(cluster==2 & origin=="Europe") %>% summarise(mean.displacement=mean(disp
                                                             mean.horsepower=mean(horsepower),
                                                             mean.weight=mean(weight),
                                                             mean.acceleration=mean(acceleration)
                                                              )
JN.2 <-  hclust.data %>% filter(cluster==2 & origin=="Japan") %>% summarise(mean.displacement=mean(displ
                                                             mean.horsepower=mean(horsepower),
                                                             mean.weight=mean(weight),
                                                             mean.acceleration=mean(acceleration)
                                                              )
US.2 <-  hclust.data %>% filter(cluster==2 & origin=="USA") %>% summarise(mean.displacement=mean(displac
                                                             mean.horsepower=mean(horsepower),
                                                             mean.weight=mean(weight),
                                                             mean.acceleration=mean(acceleration)
                                                              )
EU.3 <-  hclust.data %>% filter(cluster==3 & origin=="Europe") %>% summarise(mean.displacement=mean(disp
                                                             mean.horsepower=mean(horsepower),
                                                             mean.weight=mean(weight),
                                                             mean.acceleration=mean(acceleration)
                                                              )
JN.3 <-  hclust.data %>% filter(cluster==3 & origin=="Japan") %>% summarise(mean.displacement=mean(displ
                                                             mean.horsepower=mean(horsepower),
                                                             mean.weight=mean(weight),
                                                             mean.acceleration=mean(acceleration)
                                                              )
US.3 <-  hclust.data %>% filter(cluster==3 & origin=="USA") %>% summarise(mean.displacement=mean(displac
                                                             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) %>% kable_styling(font_size = 7)
kable(hclust.in.clusters,format="latex",booktabs=T,longtable=T) %>% kable_styling(font_size = 6)
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")

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"))

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$origin))+
  theme_bw()+
  theme(panel.grid=element_blank(),
        plot.title=element_text(hjust=.5,size=20),
        plot.subtitle =element_text(hjust=.5,size=15)) +
  labs(title=" Pairplot of numerical features",
       subtitle = "Log MPG transformed")+
  scale_color_manual(values=brewer.pal(3,"Set1"))
lr.data <- lm(mpg~.,data=data[-c(8)])
summary(lr.data)
#residuals vs fitted plot
p1 <- ggplot(lr.data)+
  theme_light()+
  labs(title = "Residuals Fitted Values",x="Fitted Values",y="Residuals")+
  geom_point(aes(x=lr.data$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)
summary(log.lr.data)
#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="violet
  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))

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
  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)
```

```
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')+
  geom_text(aes(0,avg.b.RMSE,label = avg.b.RMSE, vjust = -1))

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')+
  geom_text(aes(0,avg.b.Rsq,label = avg.b.Rsq, vjust = -.9))

model <- train(
  build.model.features(data.transformed),
  data.transformed,
  method = "lm",
  trControl = trainControl(
    method = "repeatedcv",
    number = 10,
    repeats = 10,
    verboseIter = TRUE
  )
)
model
```

```
new.dat <- data.frame(cylinders=as.factor(8),displacement=80,horsepower=69,weight=2020,acceleration=19,
new.dat
10^predict(model,new.dat)
contrasts(data.transformed$cylinders)
contrasts(data.transformed$origin)
```