Project

Brandon Hom

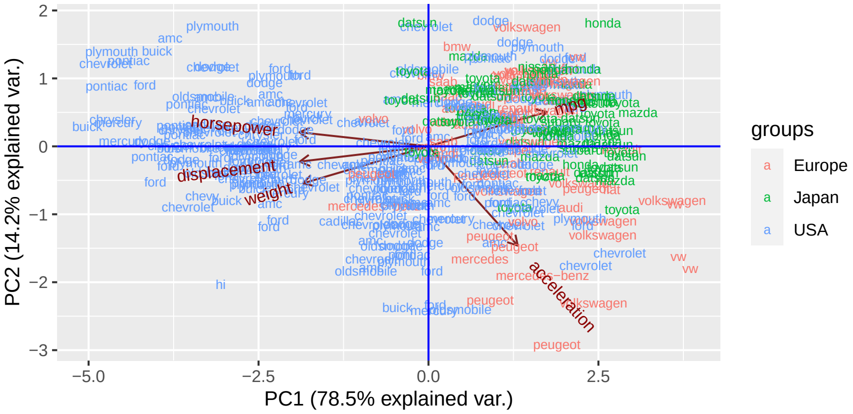
11/2/2021

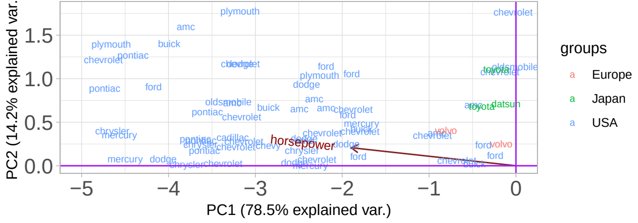
Introduction

Exploratory data analysis

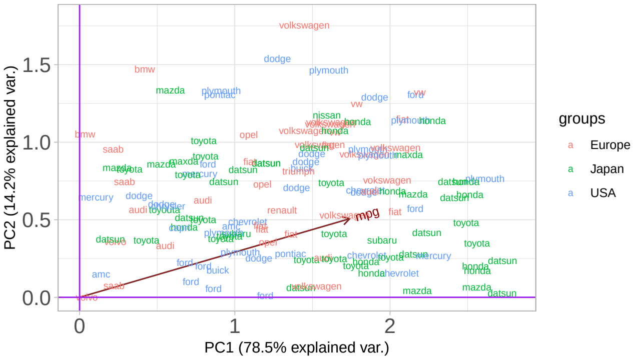
```
##                     mpg displacement horsepower     weight acceleration
## mpg           1.0000000   -0.8051269 -0.7784268 -0.8322442    0.4233285
## displacement -0.8051269    1.0000000  0.8972570  0.9329944   -0.5438005
## horsepower   -0.7784268    0.8972570  1.0000000  0.8645377   -0.6891955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

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5
## Standard deviation     1.9816 0.8438 0.47500 0.28788 0.22966
## Proportion of Variance 0.7853 0.1424 0.04512 0.01658 0.01055
## Cumulative Proportion  0.7853 0.9277 0.97288 0.98945 1.00000
```

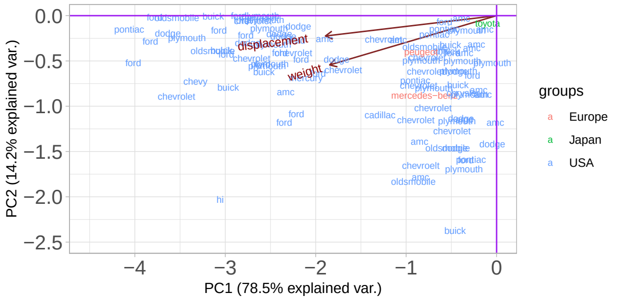




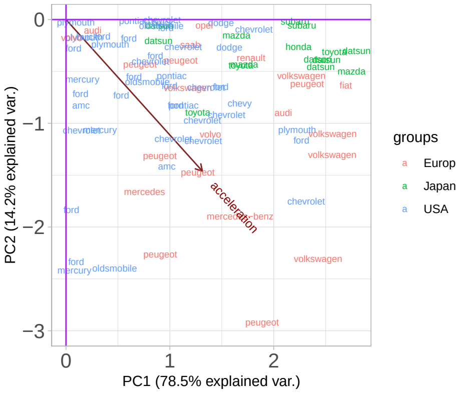
Top left



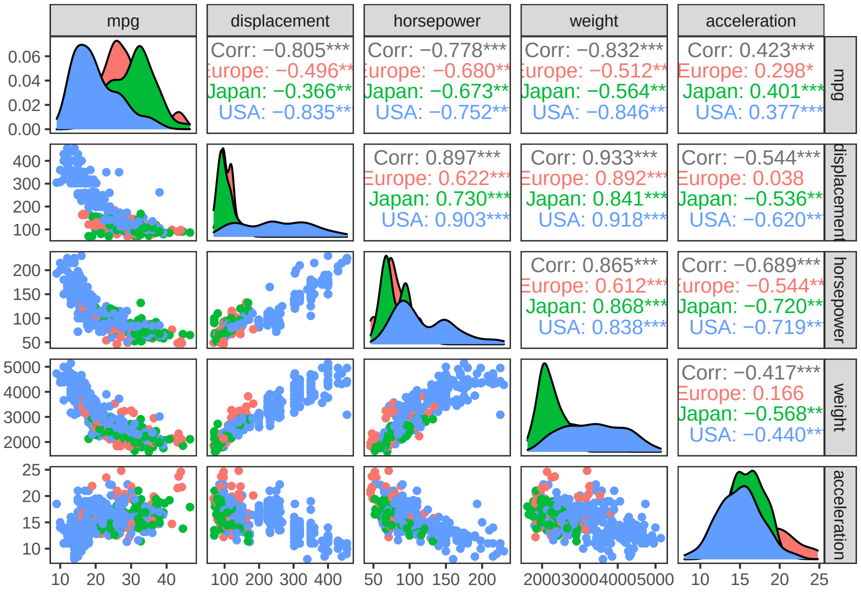
Top Right



Bottom Left

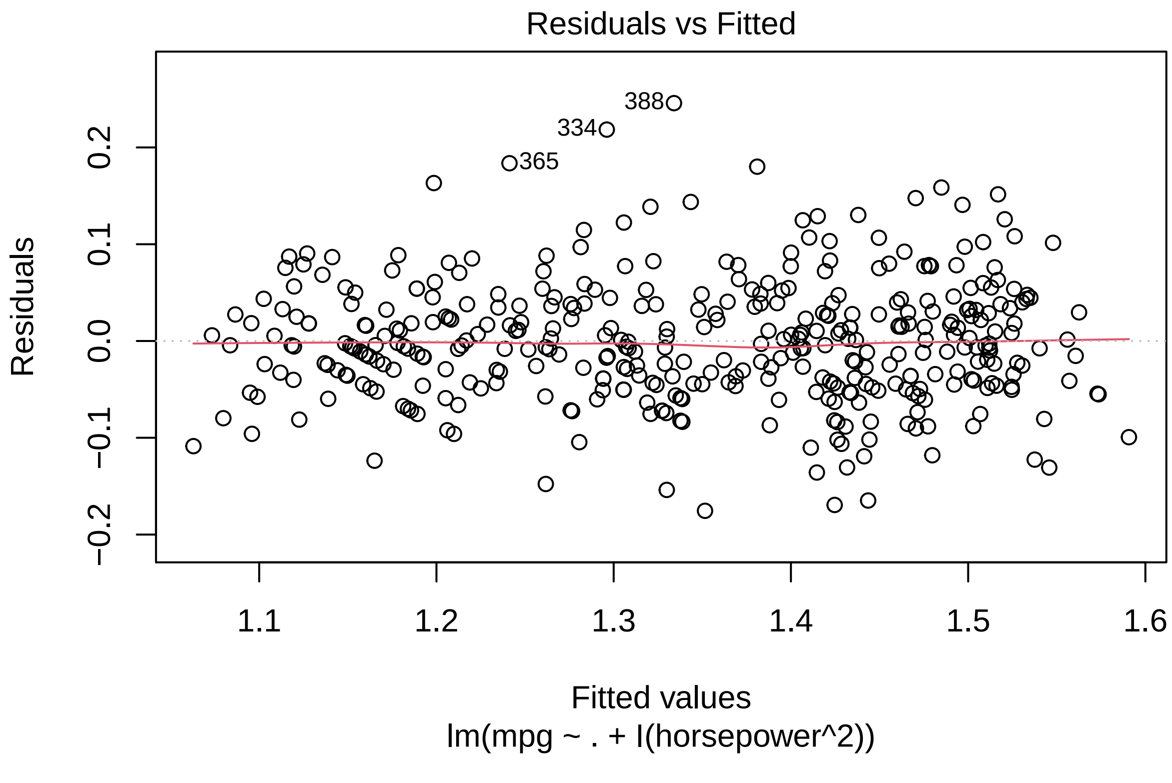


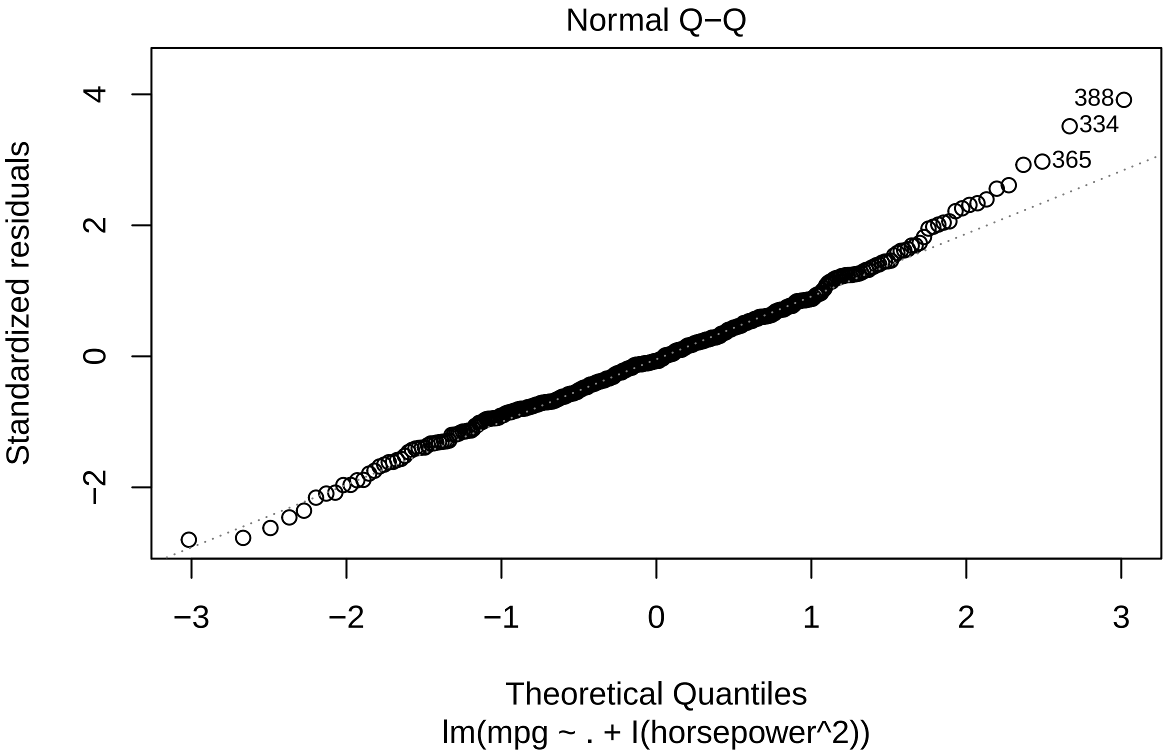
Bottom Right

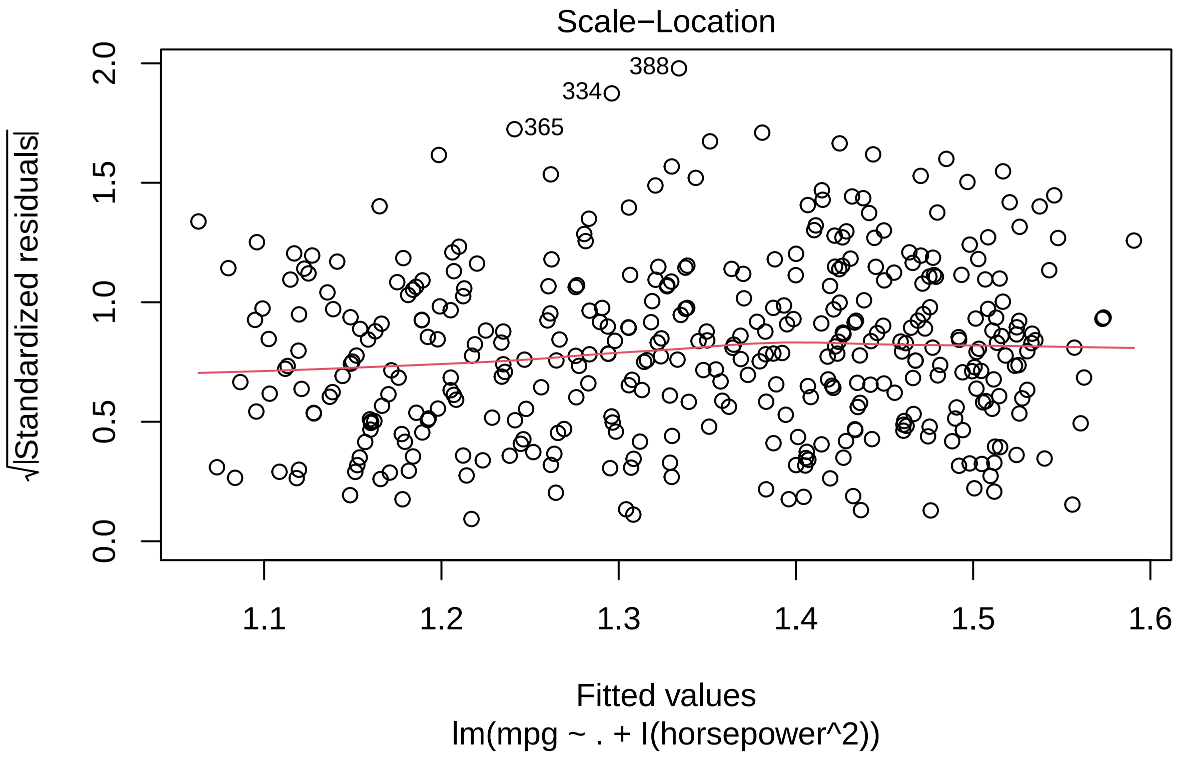


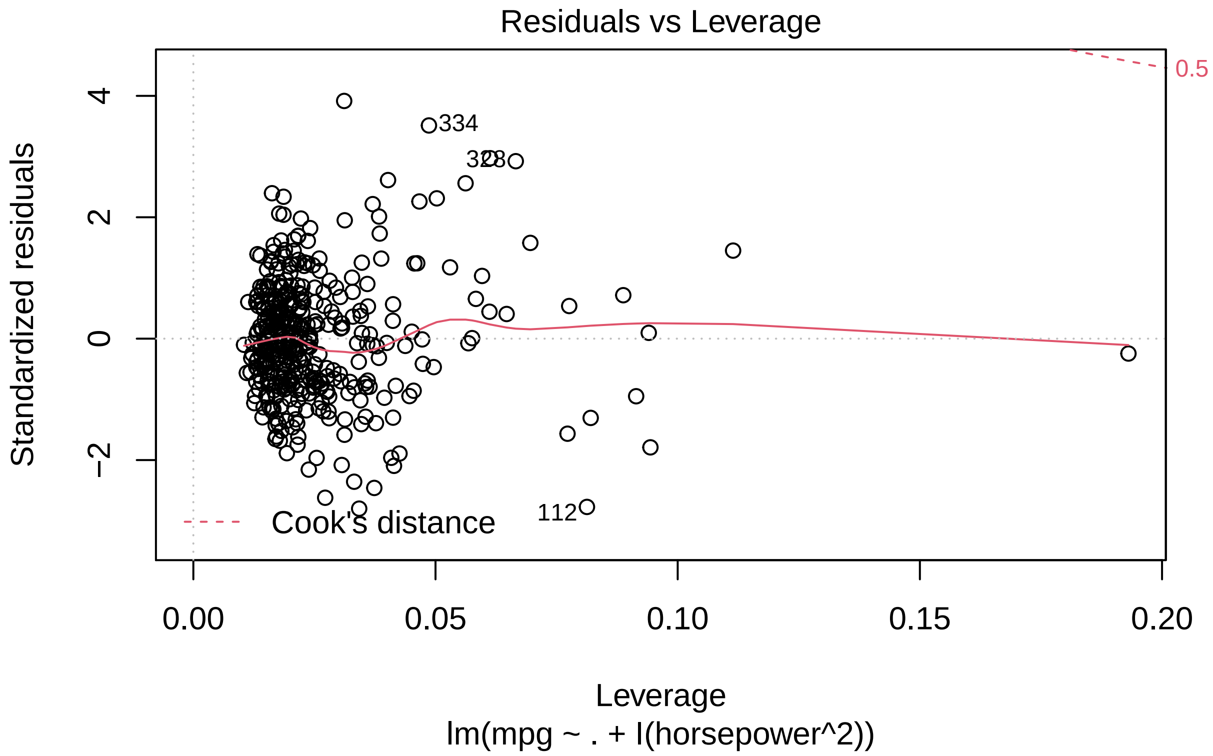
```
##
## Call:
## lm(formula = mpg ~ . + I(horsepower^2), data = data.transformed)
##
## Residuals:
```

```
##       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
```




Normal Q−Q
Standardized residuals
4
2
0
−2
388
334
365
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(mpg ~ . + I(horsepower^2))


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
388
334
365
1.1
1.2
1.3
1.4
1.5
1.6
Fitted values
lm(mpg ~ . + I(horsepower^2))



```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

##
## 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

```
## mpg ~ . + I(horsepower^2)
## <environment: 0x7f93872f92f8>
```

```
##      Train.RMSE Train.R.Squared       Test.RMSE  Test.R.Squared
##      0.06301239      0.81931697      0.05744134      0.83059552
```

Cross-Fold validation model

```
## Linear Regression
##
## 392 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 352, 354, 351, 353, 353, 353, ...
## Resampling results:
##
##   RMSE        Rsquared   MAE
##   0.06461669  0.8109213  0.05022901
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
##       1
## 26.8867
```

Conclusion

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)
library(tidyverse)
library(knitr)
library(leaps)
library(GGally)
library(ggbiplot)
library(caret)
# data cleaning up
data <- read.csv('auto-mpg.csv')
#convert horsepower chr->dbl
data$horsepower <- as.numeric(data$horsepower)
#remove rows with missing values
```

```r
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)
cor(data[-c(2,7,8)])
#h.clustering.complete <- hclust(dist(data),method="complete")
#dendro.data <- dendro_data(as.dendrogram(h.clustering.complete),type="rectangle")

#ggplot(dendro.data$segments) +
  #geom_segment(aes(x = x, y = y, xend = xend, yend = yend))+
  #geom_text(data = dendro.data$labels, aes(x, y, label = label),
           # hjust=1,angle = 90, size = 3.5)+ylim(-1,5000)+
  #theme_light()+
  #theme(panel.grid=element_blank(),
   #     plot.title=element_text(hjust=.5,size=20),
      #  axis.text = element_text(size=15)
       # )+
  #labs(title="Complete linkage Dendrogram of Cars",
   #    x="Car Name",
    #   y="height")
pcs.out <- prcomp(data[-c(2,7,8)],scale.=T)
summary(pcs.out)
ggbiplot(pcs.out,labels = data$car.name,groups=data$origin,obs.scale = 1,labels.size = 2.3)+
  geom_hline(yintercept = 0,col="blue")+
  geom_vline(xintercept = 0,col="blue")

ggbiplot(pcs.out,labels = data$car.name,groups=data$origin,obs.scale = 1,labels.size = 2.3)+
  geom_hline(yintercept = 0,col="purple")+
  geom_vline(xintercept = 0,col="purple")+
  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",
       )


ggbiplot(pcs.out,labels = data$car.name,groups=data$origin,obs.scale = 1,labels.size = 2.3)+
  geom_hline(yintercept = 0,col="purple")+
  geom_vline(xintercept = 0,col="purple")+
  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",
       )

ggbiplot(pcs.out,labels = data$car.name,groups=data$origin,obs.scale = 1,labels.size = 2.3)+
  geom_hline(yintercept = 0,col="purple")+
  geom_vline(xintercept = 0,col="purple")+
  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",
       )

ggbiplot(pcs.out,labels = data$car.name,groups=data$origin,obs.scale = 1,labels.size = 2.3)+
  geom_hline(yintercept = 0,col="purple")+
  geom_vline(xintercept = 0,col="purple")+
  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",
       )
ggpairs(data[-c(2,7,8)],aes(color=data$origin))+theme_bw()+theme(panel.grid=element_blank())
data.transformed <- data
data.transformed$mpg <- log(data.transformed$mpg,base=10)
data.transformed <- data.transformed[-c(8)]
fit <- lm(mpg~.+I(horsepower^2),data=data.transformed)
summary(fit)
plot(fit)
library(MASS)
step.model <- stepAIC(fit, 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)+I(weight^2), specify what features to remove
build.model <- 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)")))

}

build.and.evaluate <- function(data,split.size,feats="None"){
  train <- train.test(data,split.size)[[1]]
  test <- train.test(data,split.size)[[2]]
  model <- lm(build.model(data,feats),train)
  print(build.model(data,feats))
  p.train <- predict(model,train)
  p.test <- predict(model,test)
  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)
}

build.and.evaluate(data.transformed,.8)
model <- train(
  build.model(data.transformed),
  data.transformed,
  method = "lm",
  trControl = trainControl(
    method = "repeatedcv",
    number = 10,
    repeats = 10,
    verboseIter = TRUE
  )
)
model.no <- train(
  build.model(data.transformed,c("displacement")),
  data.transformed,
  method = "lm",
  trControl = trainControl(
    method = "cv",
    number = 10,
    verboseIter = TRUE
  )
)
model
new.dat <- data.frame(cylinders=as.factor(6),displacement=80,horsepower=69,weight=2020,acceleration=19,
10^predict(model,new.dat)
```