Kaggle Code

Benjamin Gerber

3/22/2021

Setup

```
library(readr)
carsTrain <- read_csv("~/Desktop/Stats 101A/carsTrain.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   Manufacturer = col_character(),
##   Model = col_character(),
##   Type = col_character(),
##   AirBags = col_character(),
##   DriveTrain = col_character(),
##   Cylinders = col_character(),
##   Man.trans.avail = col_character(),
##   Origin = col_character(),
##   Make = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
carsTestNoY <- read_csv("~/Desktop/Stats 101A/carsTestNoY.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   Manufacturer = col_character(),
##   Model = col_character(),
##   Type = col_character(),
##   AirBags = col_character(),
##   DriveTrain = col_character(),
##   Cylinders = col_character(),
##   Man.trans.avail = col_character(),
##   Origin = col_character(),
##   Make = col_character()
## )
## See spec(...) for full column specifications.
```

```
library(ggplot2)
library(car)
```

```
## Loading required package: carData
```

```
library(MASS)
library(corrplot)
```

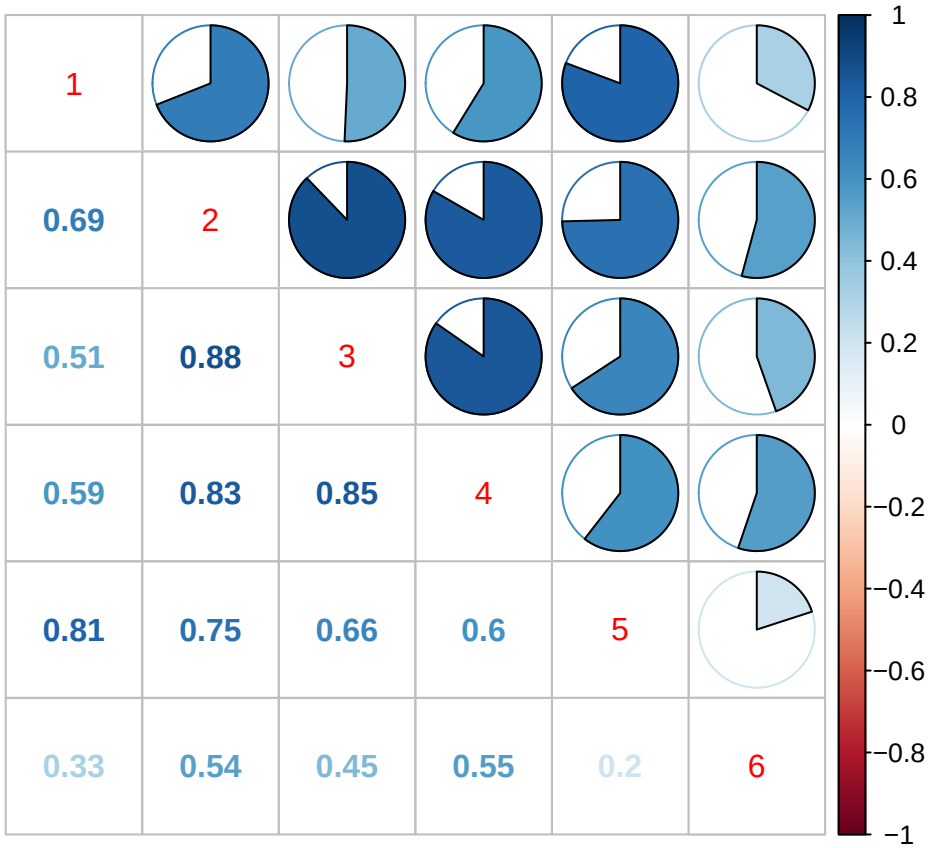
```
## corrplot 0.84 loaded
```

```
library(gridExtra)
```

```
corr<- round(cor(cbind(carsTrain$PriceNew,carsTrain$Weight,carsTrain$Width,carsTrain$Length,carsTrain$H
corr
```

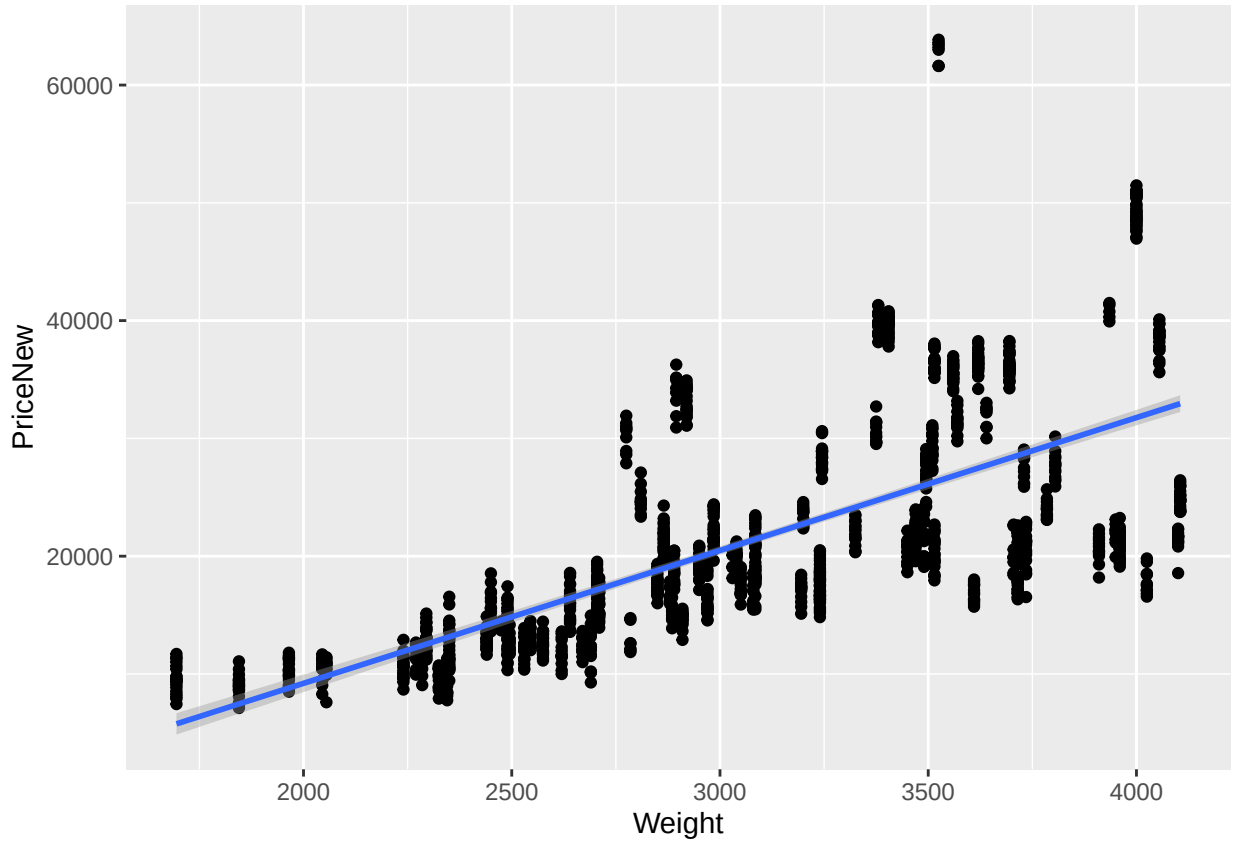
```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
## [1,] 1.000 0.690 0.507 0.588 0.807 0.327
## [2,] 0.690 1.000 0.878 0.833 0.746 0.542
## [3,] 0.507 0.878 1.000 0.847 0.658 0.446
## [4,] 0.588 0.833 0.847 1.000 0.605 0.552
## [5,] 0.807 0.746 0.658 0.605 1.000 0.200
## [6,] 0.327 0.542 0.446 0.552 0.200 1.000
```

```
corrplot.mixed(corr, lower="number",upper="pie")
```



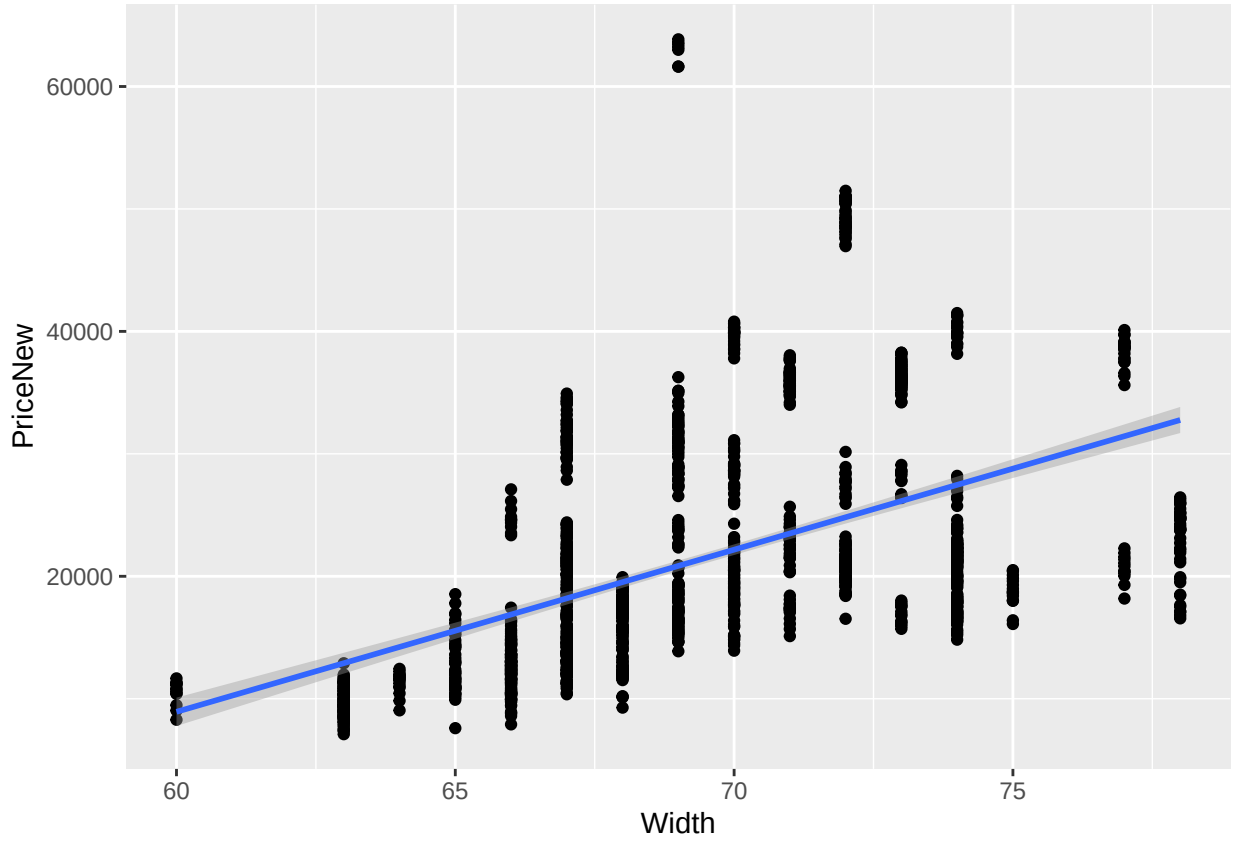
```
ggplot(carsTrain,aes(Weight,PriceNew)) +geom_point() +geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



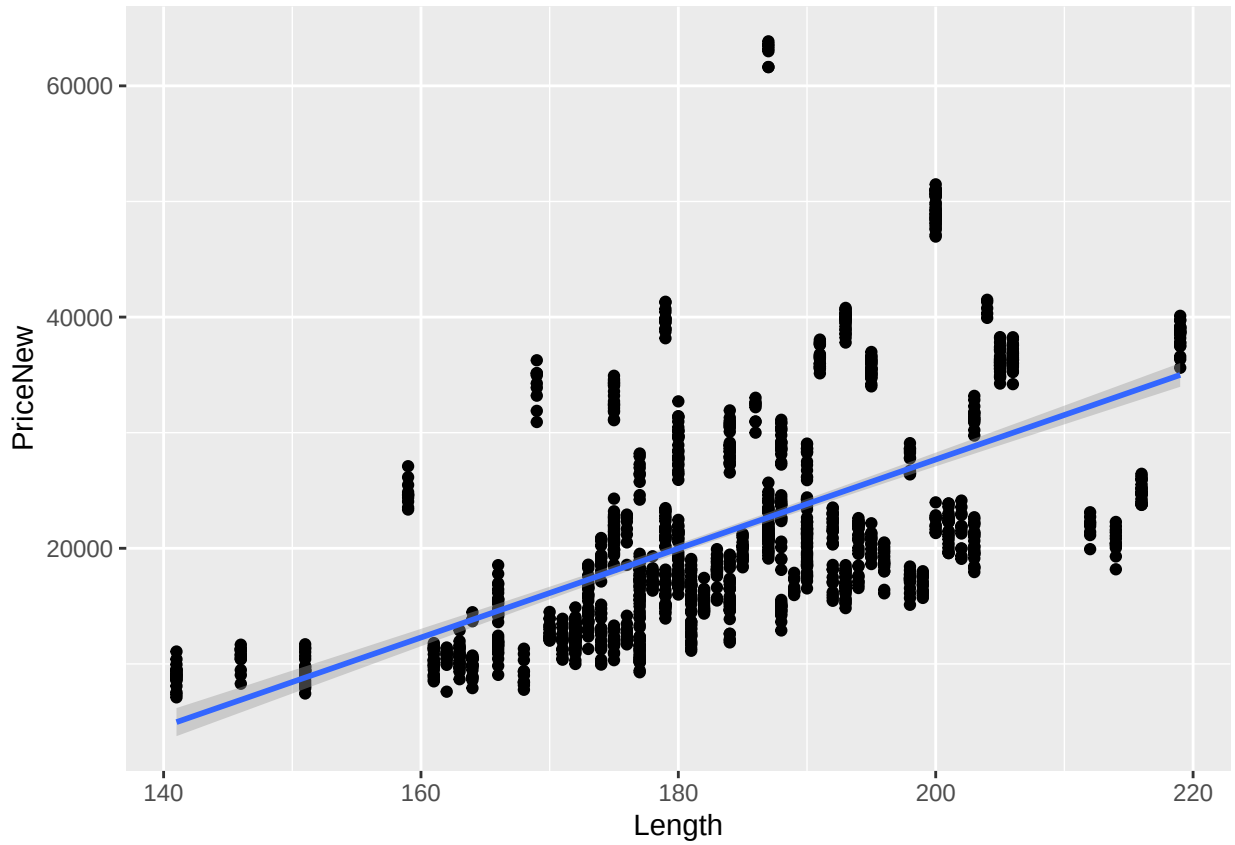

```
ggplot(carsTrain,aes(Width,PriceNew)) +geom_point() +geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



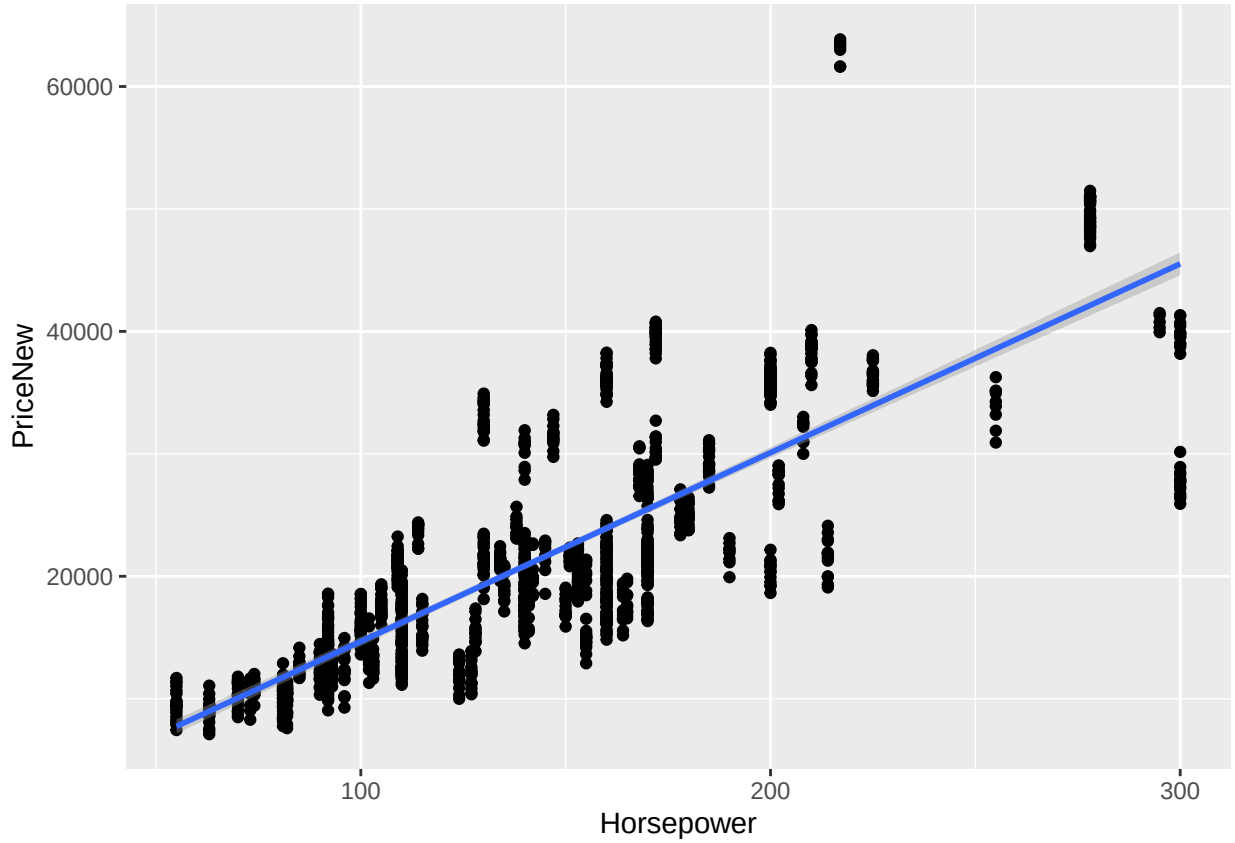

```
ggplot(carsTrain,aes(Length,PriceNew)) +geom_point() +geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



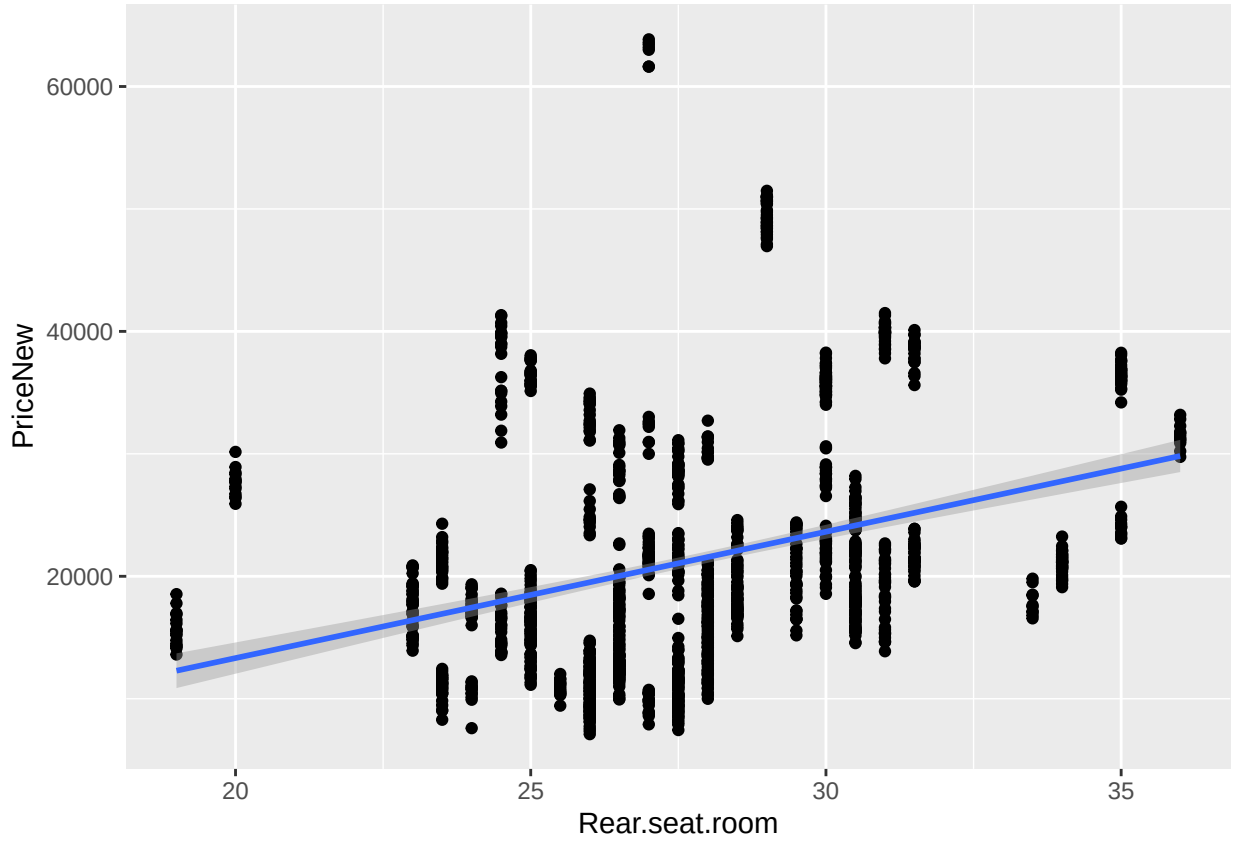

```
ggplot(carsTrain,aes(Horsepower,PriceNew)) +geom_point() +geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
ggplot(carsTrain,aes(Rear.seat.room,PriceNew)) +geom_point() +geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
m1 <- lm(PriceNew ~ Width + Weight + Horsepower + Cylinders, data=carsTrain)
summary(m1)
```

```
##
## Call:
## lm(formula = PriceNew ~ Width + Weight + Horsepower + Cylinders,
##     data = carsTrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14989.5  -2897.7   -198.6   2309.0  29198.0
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     74398.3422  4378.6862  16.991  < 2e-16 ***
## Width           -1422.3987    78.5406 -18.110  < 2e-16 ***
## Weight              9.5966     0.5924  16.199  < 2e-16 ***
## Horsepower         94.3963     4.8779  19.352  < 2e-16 ***
## Cylinders4        551.4953   790.9213   0.697 0.485735
## Cylinders5       2356.4670  1218.8177   1.933 0.053376 .
## Cylinders6       4087.6841  1066.6876   3.832 0.000132 ***
## Cylinders8      10760.7364  1305.2050   8.244  3.6e-16 ***
## Cylindersrotary  5859.5118  2010.9662   2.914 0.003624 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4982 on 1491 degrees of freedom
## Multiple R-squared:  0.7422, Adjusted R-squared:  0.7408
## F-statistic: 536.5 on 8 and 1491 DF,  p-value: < 2.2e-16
```

```
m2 <- lm(PriceNew ~ Width + Weight + Horsepower + AirBags, data=carsTrain)
summary(m2)
```

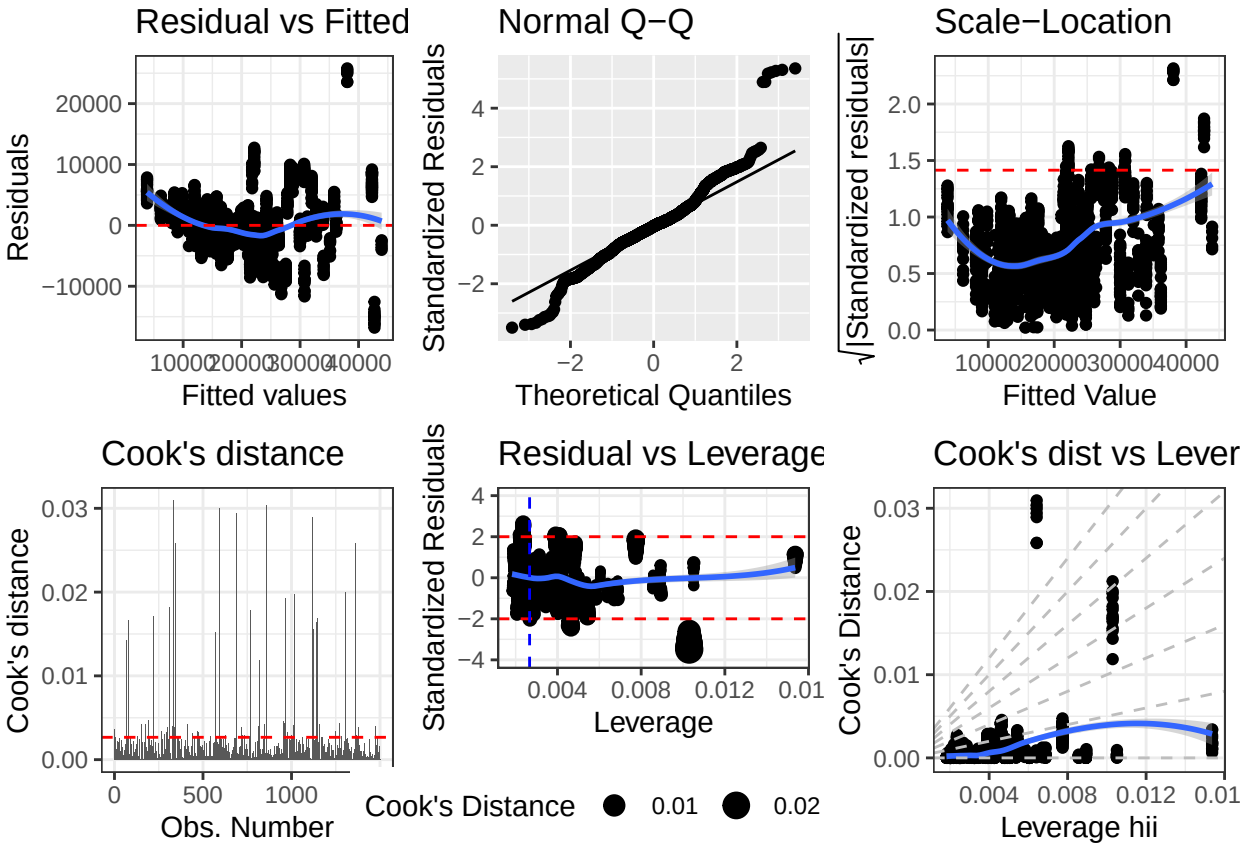
```
##
## Call:
## lm(formula = PriceNew ~ Width + Weight + Horsepower + AirBags,
##     data = carsTrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16805.2  -2598.0   -149.4   2313.0  25785.9
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        70949.8690  3880.6370  18.283   <2e-16 ***
## Width              -1326.0845    70.6649 -18.766   <2e-16 ***
## Weight                 9.9813     0.4956  20.138   <2e-16 ***
## Horsepower           107.9741     3.8301  28.191   <2e-16 ***
## AirBagsDriver only -3119.9392   344.6144  -9.053   <2e-16 ***
## AirBagsNone        -6434.7686   390.7257 -16.469   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4827 on 1494 degrees of freedom
## Multiple R-squared:  0.7575, Adjusted R-squared:  0.7566
## F-statistic: 933.1 on 5 and 1494 DF,  p-value: < 2.2e-16
```

```
diagPlot<-function(model){
  p1<-ggplot(model, aes(model$fitted, model$residuals),label=rownames(bonds))+geom_point()
  p1<-p1+stat_smooth(method="loess")+geom_hline(yintercept=0, col="red", linetype="dashed")
  p1<-p1+xlab("Fitted values")+ylab("Residuals")
  p1<-p1+ggtitle("Residual vs Fitted Plot")+theme_bw()

  p2<-ggplot(model,aes(sample=rstandard(model))) + stat_qq() + stat_qq_line()
  p2<-p2+xlab("Theoretical Quantiles")+ylab("Standardized Residuals")
  p2<-p2+ggtitle("Normal Q-Q")

  p3<-ggplot(model, aes(model$fitted, sqrt(abs(rstandard(model)))))+geom_point(na.rm=TRUE)
  p3<-p3+stat_smooth(method="loess", na.rm = TRUE)+xlab("Fitted Value")
  p3<-p3+ylab(expression(sqrt("|Standardized residuals|")))
  p3<-p3+ggtitle("Scale-Location")+theme_bw()+geom_hline(yintercept=sqrt(2), col="red", linetype="dashe

  p4<-ggplot(model, aes(seq_along(cooks.distance(model)), cooks.distance(model)))+geom_bar(stat="identi
  p4<-p4+xlab("Obs. Number")+ylab("Cook's distance")
  p4<-p4+ggtitle("Cook's distance")+theme_bw()+geom_hline(yintercept=4/(length(model$residuals-2)), col=

  p5<-ggplot(model, aes(hatvalues(model), rstandard(model)))+geom_point(aes(size=cooks.distance(model))
  p5<-p5+stat_smooth(method="loess", na.rm=TRUE)
```

```
  p5<-p5+xlab("Leverage")+ylab("Standardized Residuals")
  p5<-p5+ggtitle("Residual vs Leverage Plot")
  p5<-p5+scale_size_continuous("Cook's Distance", range=c(1,5))
  p5<-p5+theme_bw()+theme(legend.position="bottom")+geom_hline(yintercept=c(-2,2),col="red",linetype="da

  p6<-ggplot(model, aes(hatvalues(model), cooks.distance(model)))+geom_point( na.rm=TRUE)+stat_smooth(me
  p6<-p6+xlab("Leverage hii")+ylab("Cook's Distance")
  p6<-p6+ggtitle("Cook's dist vs Leverage")
  p6<-p6+geom_abline(slope=seq(0,3,0.5), color="gray", linetype="dashed")
  p6<-p6+theme_bw()
  return(grid.arrange(p1,p2,p3,p4,p5,p6,ncol=3))
}
```

```
diagPlot(m2)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```




```
vif(m2)
```

```
##                GVIF Df GVIF^(1/(2*Df))
## Width      4.517998  1        2.125558
```

```
## Weight     5.658977  1        2.378860
## Horsepower 2.475802  1        1.573468
## AirBags    1.392211  2        1.086241
```

```
m3 <- lm(PriceNew ~ Width + Horsepower + AirBags, data=carsTrain)
anova(m3,m2)
```

```
## Analysis of Variance Table
##
## Model 1: PriceNew ~ Width + Horsepower + AirBags
## Model 2: PriceNew ~ Width + Weight + Horsepower + AirBags
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1   1495 4.4267e+10
## 2   1494 3.4816e+10  1 9450929439 405.55 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(powerTransform(cbind(PriceNew,Width,Weight,Horsepower)~1,data=carsTrain))
```

```
## bcPower Transformations to Multinormality
##            Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## PriceNew     -0.5533       -0.50      -0.6371      -0.4696
## Width        -2.2653       -2.00      -2.8394      -1.6912
## Weight        0.2490        0.33       0.1036       0.3944
## Horsepower   -0.3174       -0.33      -0.4033      -0.2315
##
## Likelihood ratio test that transformation parameters are equal to 0
##  (all log transformations)
##                                    LRT df       pval
## LR test, lambda = (0 0 0 0) 314.4593  4 < 2.22e-16
##
## Likelihood ratio test that no transformations are needed
##                                   LRT df       pval
## LR test, lambda = (1 1 1 1) 1672.59  4 < 2.22e-16
```

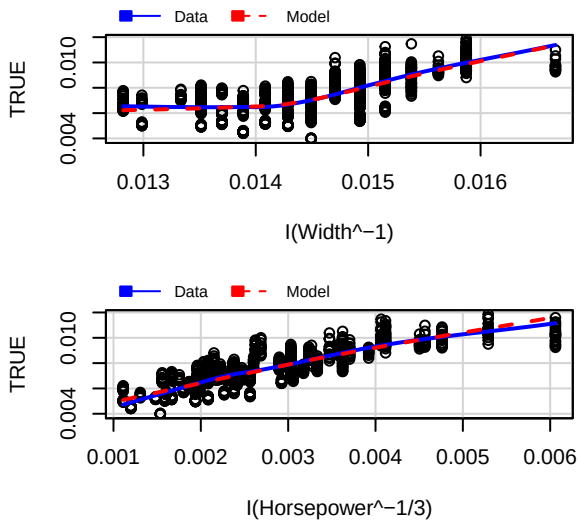
```
final <- lm(((PriceNew^-0.5)) ~ I(Width^-1) + +I(Weight^1/3) + I(Horsepower^-1/3) +AirBags, data=carsTra
summary(final)
```

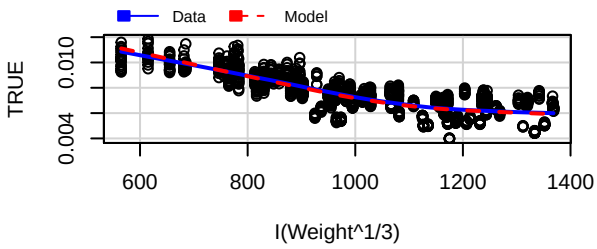
```
##
## Call:
## lm(formula = ((PriceNew^-0.5)) ~ I(Width^-1) + +I(Weight^1/3) +
##     I(Horsepower^-1/3) + AirBags, data = carsTrain)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.320e-03 -4.321e-04 -1.527e-05  4.534e-04  2.296e-03
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.051e-02  8.724e-04  23.516  < 2e-16 ***
## I(Width^-1)         -7.225e-01  4.801e-02 -15.048  < 2e-16 ***
## I(Weight^1/3)       -4.990e-06  2.153e-07 -23.174  < 2e-16 ***
```

```
## I(Horsepower^-1/3)  7.194e-01  3.311e-02  21.727  < 2e-16 ***
## AirBagsDriver only  2.712e-04  4.739e-05   5.723 1.26e-08 ***
## AirBagsNone         9.790e-04  5.489e-05  17.836  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0006608 on 1494 degrees of freedom
## Multiple R-squared:  0.819,  Adjusted R-squared:  0.8184
## F-statistic:  1352 on 5 and 1494 DF,  p-value: < 2.2e-16
```

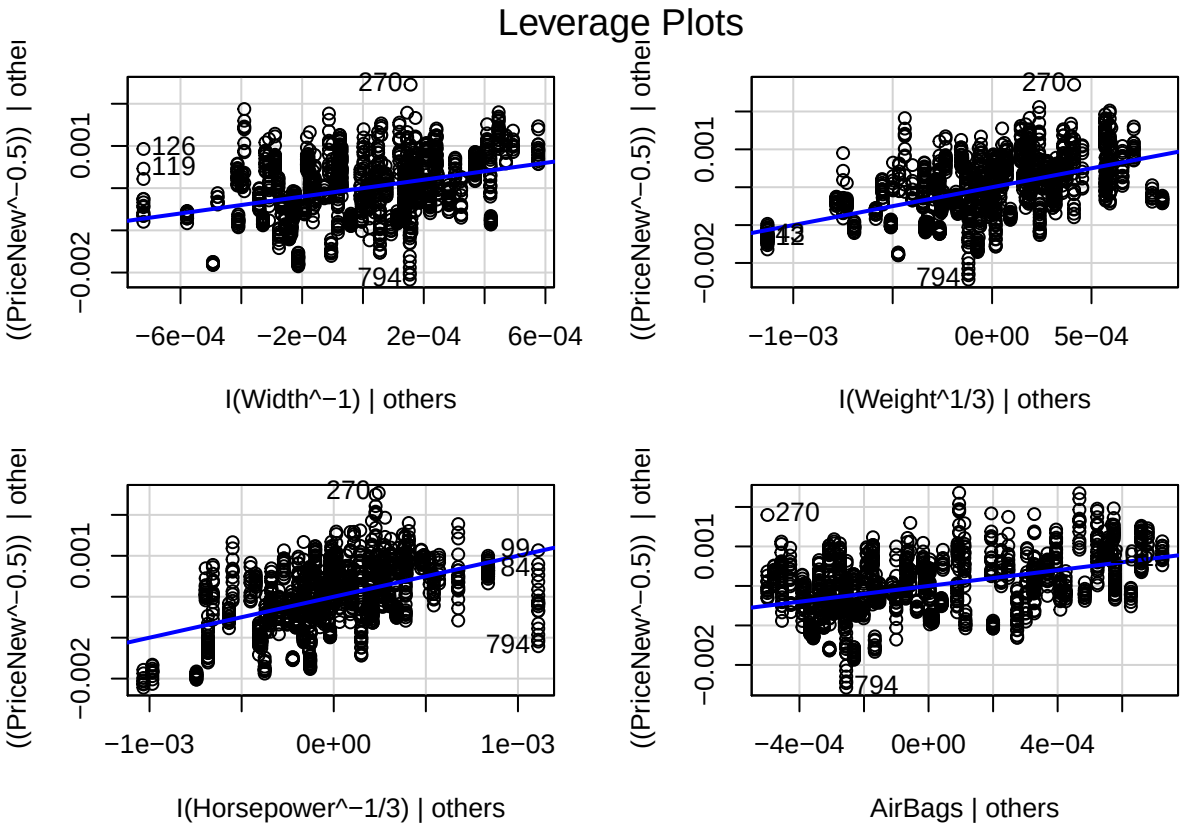
```
mmps(final)
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): NAs introduced by coercion

## Warning in min(x): no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Error in plot.window(...): need finite 'xlim' values
```

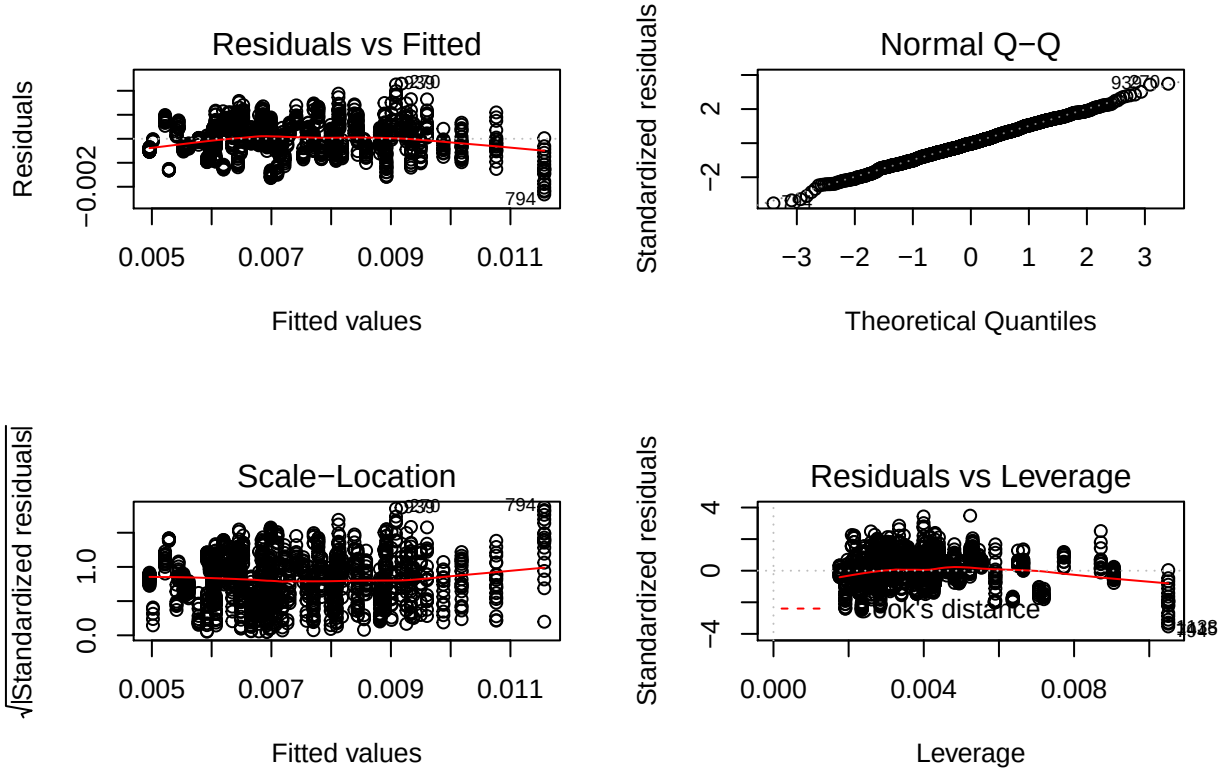





```
leveragePlots(final)
```




```
par(mfrow = c(2,2))
plot(final)
```




```
extractAIC(final,k=log(1500))
```

```
## [1]      6.00 -21928.49
```

```
PriceNew <- I(predict(final,carsTestNoY)^-2)
Ob <- 1:500
KaggleProj <- data.frame(Ob,PriceNew)
write.csv(KaggleProj,file="FINAL2.csv", row.names= FALSE)
```

‘‘‘