CSC423 Final Project

Spring Monday

Price prediction of used Hondas

By: Yiming WANG

About the data:

The data set is from (Project MOSAIC):

http://www.mosaic-web.org/go/datasets/

This data set includes used Honda car information:

1. Price of the used Honda
2. Year of the car was made
3. Mileage of the car
4. Location of the car now (try to sell in which city)
5. Color of car
6. Age of the car

Why this project:

This project is to build a model which could predict sale price of used Hondas (dependent), depending on 4 attributes (independents):

- Attribute 3 (Mileage of the car)
- Attribute 4 (Location of the car now)
- Attribute 5 (Color of car)
- Attribute 6 (Age of the car)

Would not use the year of car because it is the same meaning with age of the car.

Analysis:

By checking data set, the data is clean. So, no data cleaning process.

Start with plotting data:

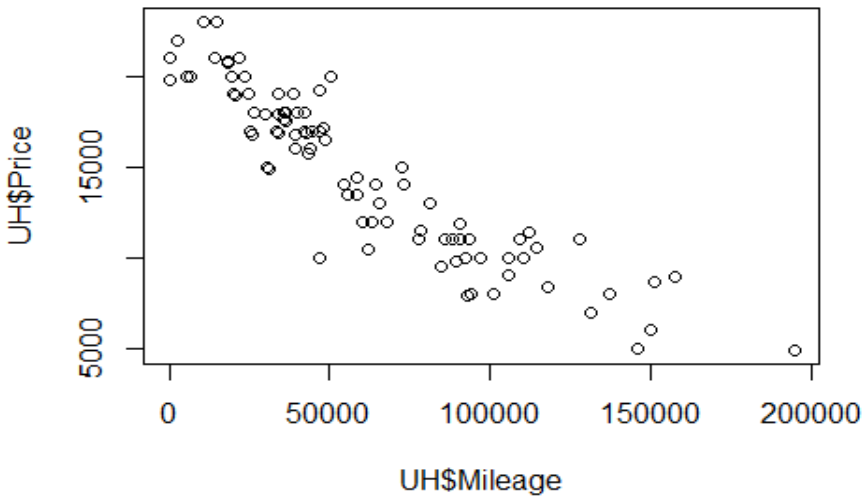


Figure 1

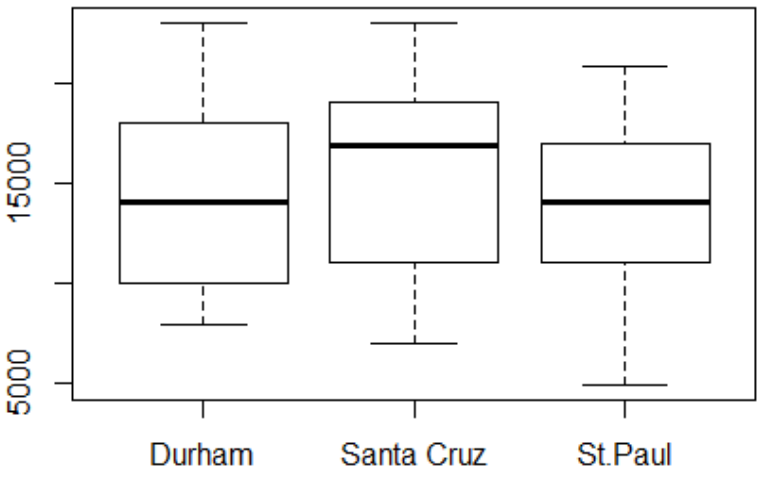


Figure 2

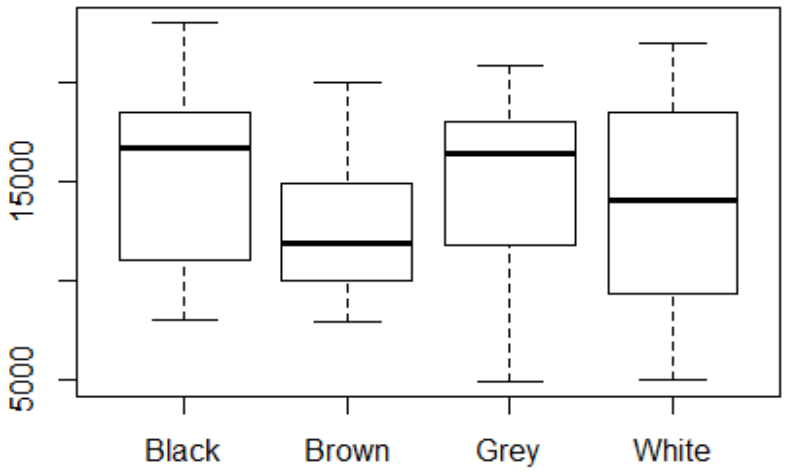


Figure 3

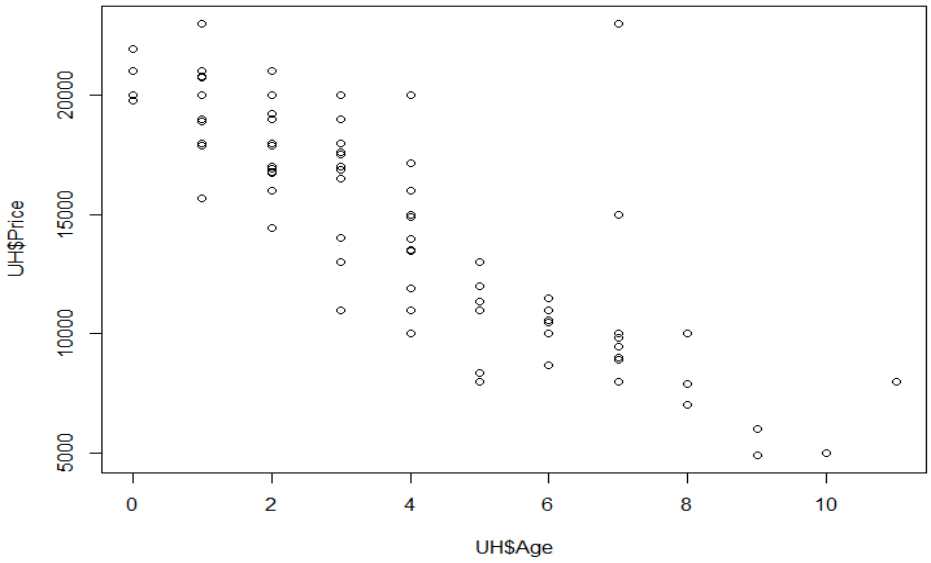


Figure 4

Depending on figure 1 to 4, the model looks more like a first order model with dummy variable and may be interactive terms.

Table 1

Cor	Price	Mileage	Age
Price	1	-0.91268	-0.82306
Mileage	-0.91268	1	0.779695
Age	-0.82306	0.779695	1

From table 1, we can see the correlation of each variables. It looks like age and mileage have high correlation. It is true that "older" car usually has higher mileage. But, it really depend on how the owner use the car. The other concern is, age would be more like the year of design. The newer car the more fashion. So, could not just treat this as multicollinearity because of high cor. But should be careful and check multicollinearity (VIF) of the models.

So, would try to search a best model from:

a. Simple first order model with dummy variables
b. First order model with dummy variables and interactive terms
c. Second order model with dummy variables and interactive terms
d. Third order model with dummy variables and interactive terms

Create dummy attributes:

For colors:

Pick White as base level (when C1=C2=C3=0, the car is white, otherwise, it is not).

Create C1 = 1 (Black) or 0 (not Black)

Create C2 = 1 (Brown) or 0 (not Brown)

Create C3 = 1 (Grey) or 0 (not Grey)

For Location:

Pick Santa Cruz as base level (when L1=L2 =0, the car is in Santa Cruz, otherwise, it is not).

Create L1 = 1 (St.Paul) or 0 (not St.Paul)

Create L2 = 1 (Durham) or 0 (not Durham)

Add interactive terms, Quadratic terms and Cube terms.

Mileage * Age; Mileage^2; Mileage^3; Age^2; Age^3

Searching best model:

Use Best Subset. Considering, high adjR^2, good Cp (Cp=p, and should be low).

Pick 12 models from all models above for further analysis:

model1: Price~Mileage+Age+C2 Cp=4.5, adjR^2=0.87

model2: Price~Mileage+Age+C2+L1 Cp=3.0, adjR^2=0.88

model3: Price~Mileage+Age+C1+C2+C3+L1+L2 Cp=8.0, adjR^2=0.87

in.model1: Price~Mileage+Age+C2+L1+M_A Cp=4.6, adjR^2=0.88

in.model2: Price~Mileage+Age+C2+L1+L2+M_A Cp=6.4, adjR^2=0.88

in.model3: Price~Mileage+Age+C1+C2+C3+L1+L2+M_A Cp=9.0, adjR^2=0.88

sq.model1: Price~Mileage+L1+M_A+M_SQ Cp=4.8, adjR^2=0.90

sq.model2: Price~Mileage+C2+M_A+M_SQ Cp=5.4, adjR^2=0.89

sq.model3: Price~Mileage+C2+L1+M_A+M_SQ Cp=3.4, adjR^2=0.90

sq.model4: Price~Mileage+Age+C1+C2+C3+L1+L2+M_A+M_SQ+A_SQ Cp=11, adjR^2=0.89

cu.model1: Price~Mileage+C2+L1+M_A+M_CU+A_CU Cp=3.8, adjR^2=0.90

cu.model2: Price~Mileage+Age+C1+C2+C3+L1+L2+M_A+M_SQ+A_SQ+M_CU+A_CU

Cp=13, adjR^2=0.90

Check betas of the models. Delete the models with beta > .05. After deleting:

model1: Price~Mileage+Age+C2

in.model1: Price~Mileage+Age+C2+L1+M_A

sq.model2: Price~Mileage+C2+M_A+M_SQ

Check multicollinearity:

```
> vif(model1)
 Mileage      Age       C2
2.645901 2.745880 1.077875

> vif(in.model1)
 Mileage      Age       C2       L1      M_A
6.587398 4.962061 1.095482 1.028312 11.741803

> vif(sq.model2)
 Mileage       C2      M_A     M_SQ
12.437452 1.143465 8.316363 15.266095
>
```

Because of the interactive term M_A, M_A and Mileage has a high VIF>10. It is reasonable.

Use Training and testing partition of data to test models, get results for the 3 of chosen models:

model1: Price~Mileage+Age+C2

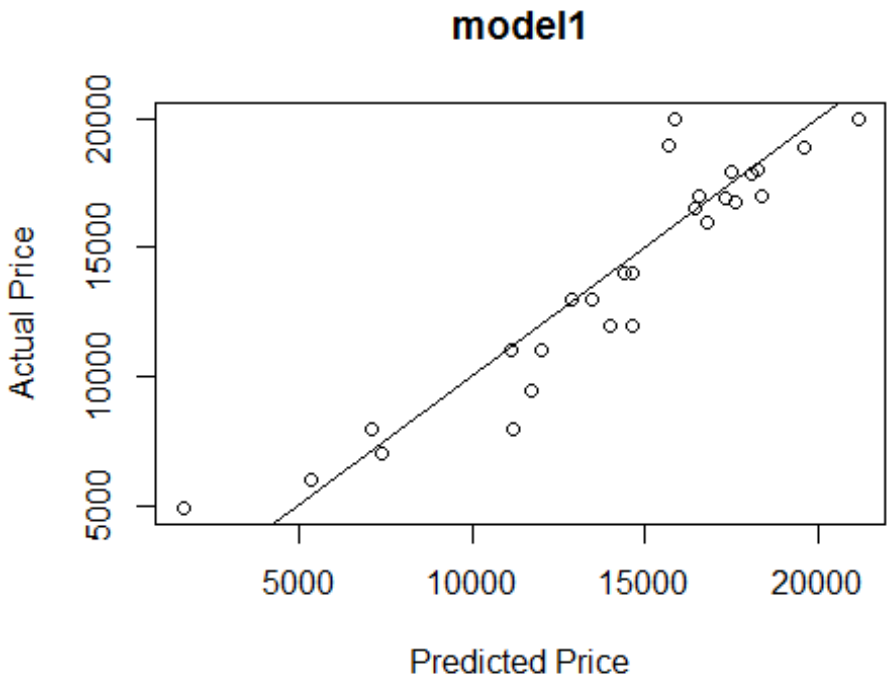


Figure 5

```
##   description cor.Price_hat.Price train.rmse test.rmse percent.error
## 1      model1           0.9455067   1677.173  1732.109      3.275541
```

in.model1: Price~Mileage+Age+C2+L1+M_A

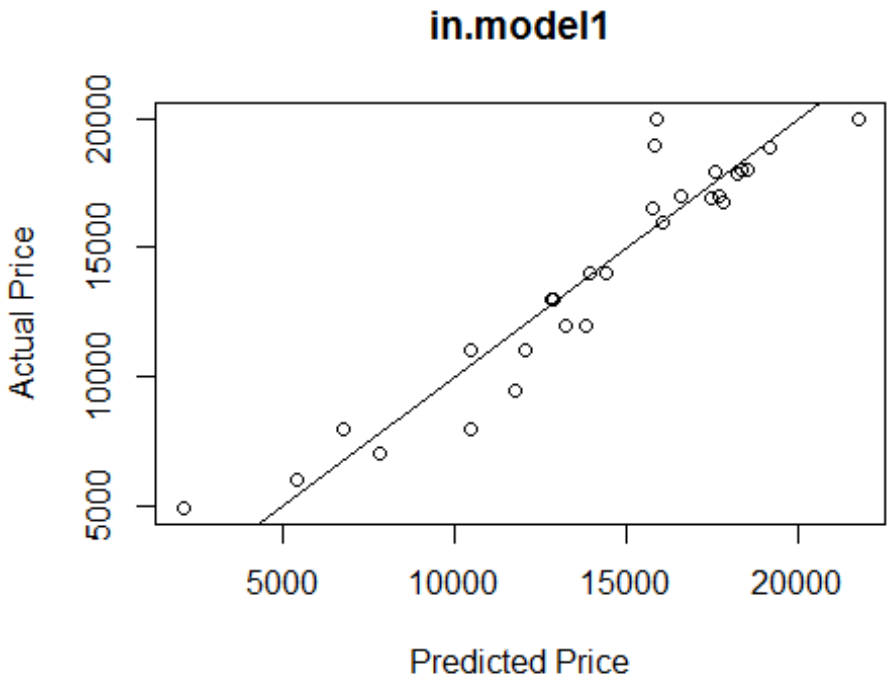


Figure 6

```
##   description cor.Price_hat.Price train.rmse test.rmse percent.error
## 1   in.model1           0.9556649   1675.764  1629.282     -2.773752
```

sq.model2: Price~Mileage+C2+M_A+M_SQ

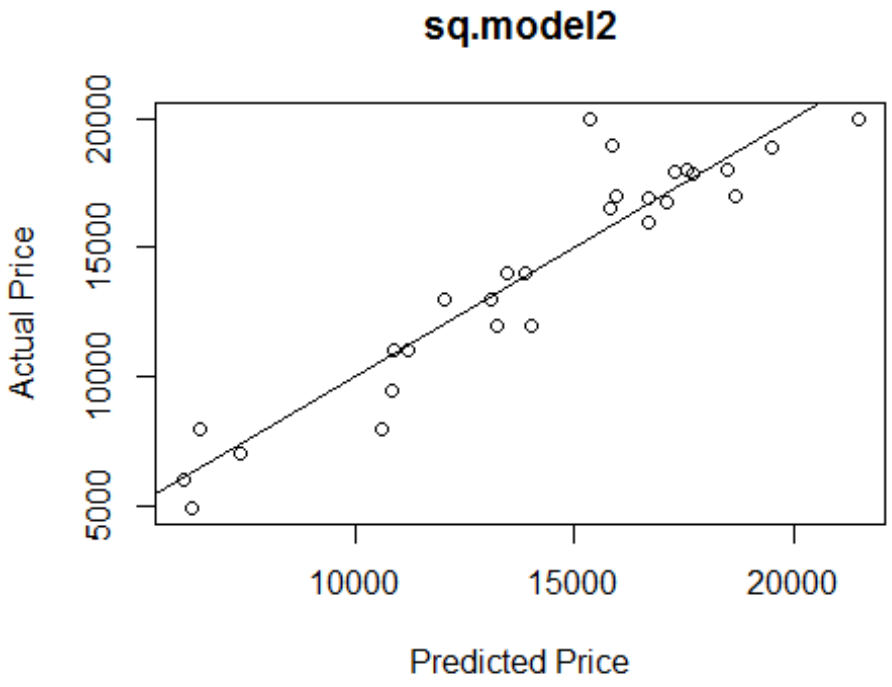


Figure 7

```
##   description cor.Price_hat.Price train.rmse test.rmse percent.error
## 1   sq.model2           0.9603641   1565.116  1519.912     -2.888264
```

Pick in.model1: Price~Mileage+Age+C2+L1+M_A as the best model because of low percent.error. And the model is not too complicated.

Heteroscedasticity:

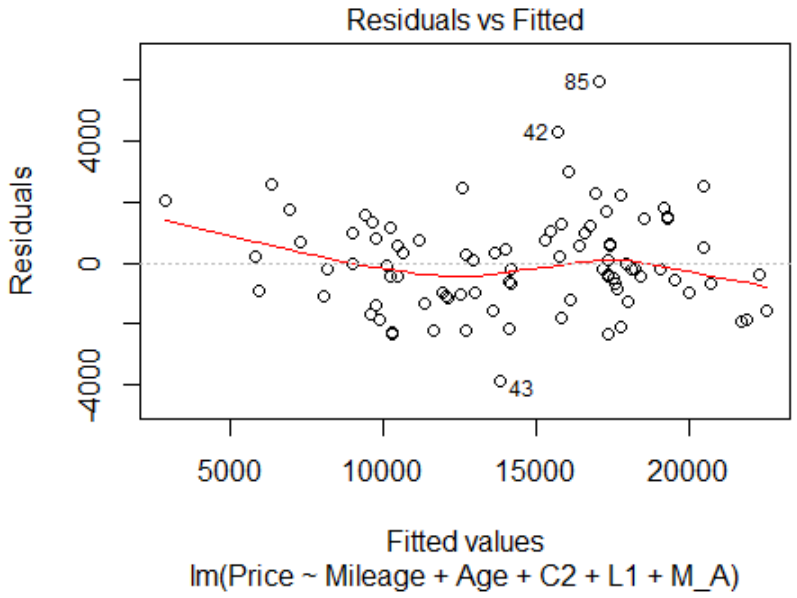


Figure 8

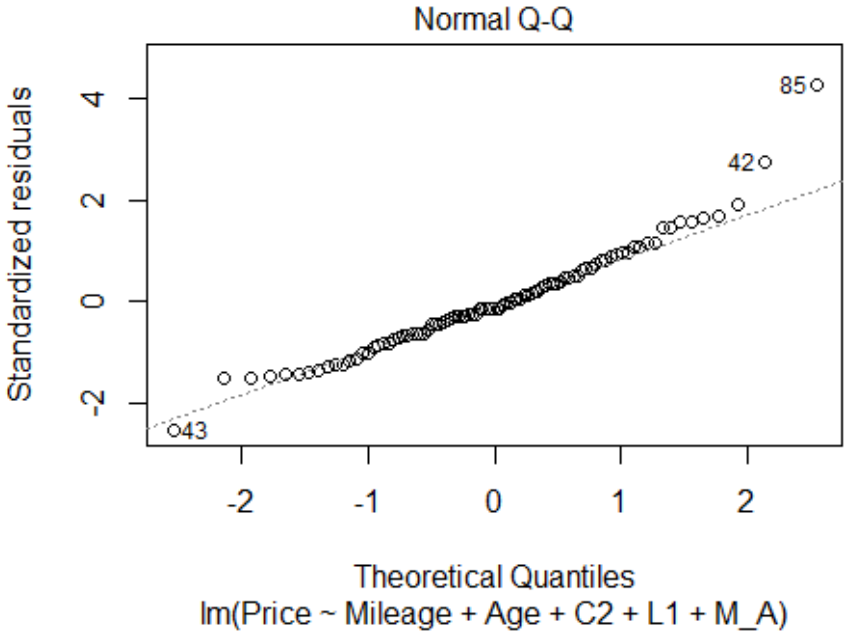


Figure 9

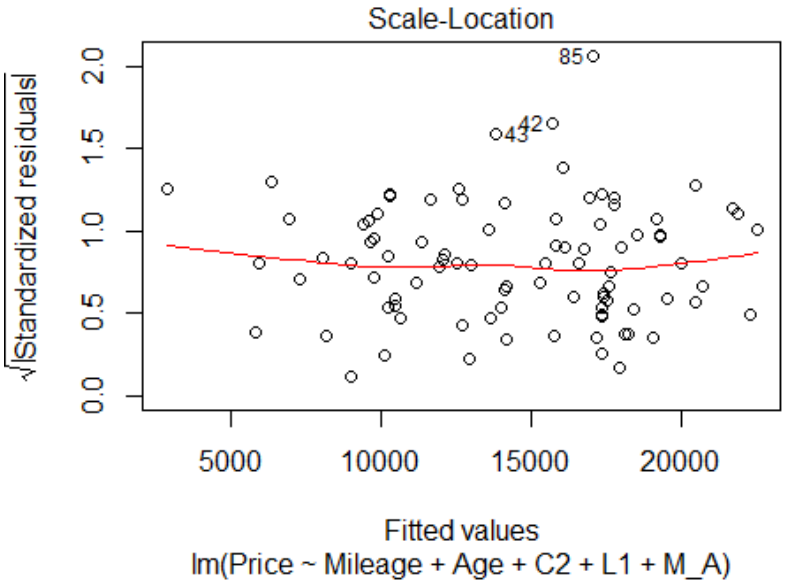


Figure 10

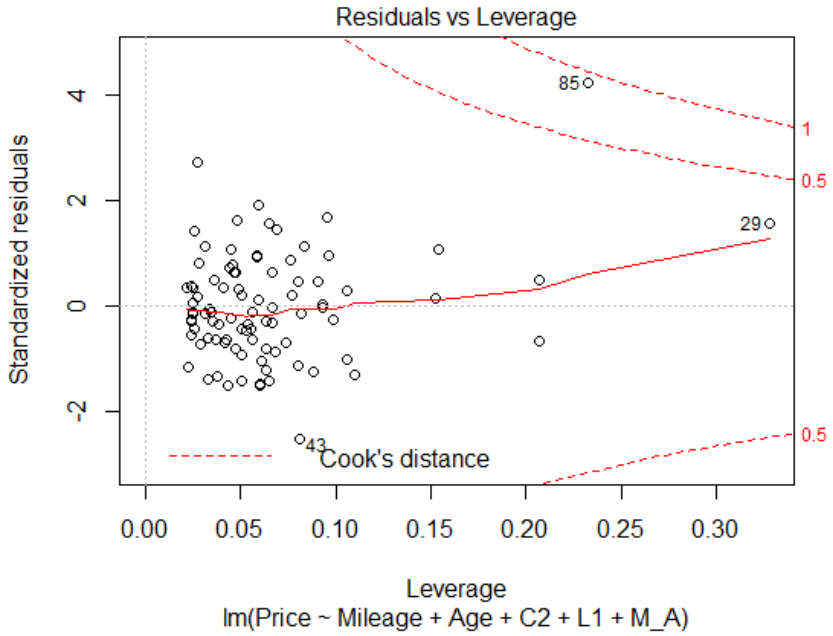


Figure 11

From Figure 8 to 11, none of the point's Cook's distance > 1.

Check the value of each possible influencial point in Figure 8 to 11:

```
standardized.residuals[85-1]

##         84
## -0.2436802

standardized.residuals[43-1]

##      42
## 2.74549

standardized.residuals[42-1]

##       41
## 1.448006

standardized.residuals[29-1]

##         28
## -0.6502143
```

None of the point locates out of 3s.

So, could be treated as no outlier, no infuencial points.

Plot the final model (Price~Mileage+Age+C2+L1+M_A):

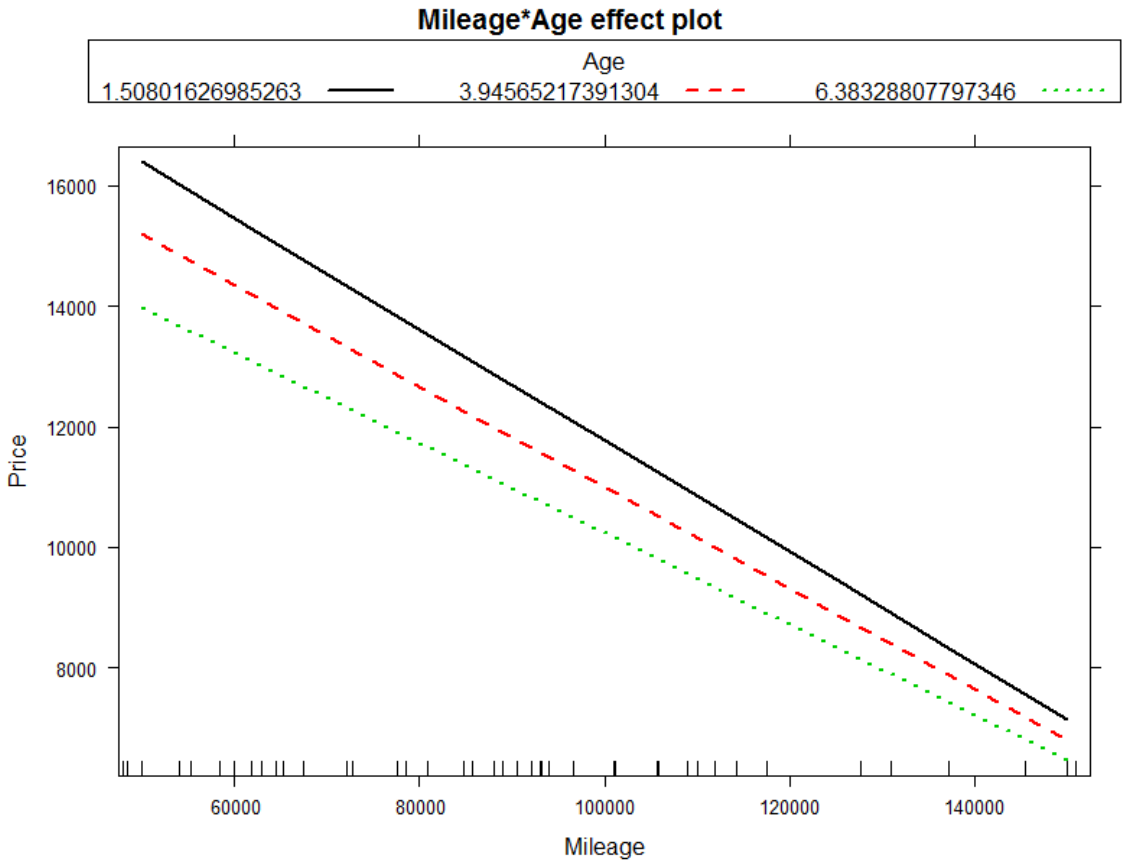


Figure 12

```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.254e+04  5.185e+02  43.468  < 2e-16 ***
## Mileage     -9.775e-02  1.030e-02  -9.489 4.96e-15 ***
## Age         -6.685e+02  1.526e+02  -4.382 3.31e-05 ***
## C2          -1.104e+03  4.141e+02  -2.666  0.00916 **
## L1          -8.172e+02  3.624e+02  -2.255  0.02667 *
## Mileage:Age  3.513e-03  1.607e-03   2.186  0.03154 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1593 on 86 degrees of freedom
## Multiple R-squared:  0.8875, Adjusted R-squared:  0.881
## F-statistic: 135.7 on 5 and 86 DF,  p-value: < 2.2e-16
```

All betas<.05, the model p-value<.05. So,the final model for predicting used Honda is:

Price=22540-0.09775*Mileage-668.5*Age-1104*C2-817.2*L1+0.003513*Mileage*Age

Beta explanation:

Beta0=22540, means a new Honda car not Brown, not located in St.Paul, could sale 22540 dollars.

Beta1=-0.09775, beta5=0.003513. (-0.09775+0.003513*Age) means when the color, location and age of a used Honda hold fixed, per mile increasing in mileage would cause -(-0.09775+0.003513*Age) dollar decrease of sale price.

Beta2=-668.5, beta5=0.003513, (-668.5+0.003513*Mileage) means when location, mileage and color hold fixed, per 1 year increase of age of a used Honda would cause decrease of -(-668.5+0.003513*Mileage) dollar in sale price.

Beta3=-1104 means when color, mileage and location hold fixed, if the used Honda is in Durham, the sale price would be 1104 dollar lower. Other location would not cause this.

Beta4=-817.2 means when location, age and mileage hold fixed. If the color of used Honda is Brown, the sale price would decrease 817.2 dollar. Other colors would not cause this.

Do 3 prediction, use the final model:

i. Mileage=20000, Age=1, Color= Brown, location= Durham, do prediction.

Mileage=20000, Age=1, C2=1,L1=0

Price=22540-0.09775*20000-668.5*1-1104*1-817.2*0+0.003513*20000*1= **$18878.83**

ii. Mileage=50000, Age=7, Color= Black, location= Santa Cruz, do prediction.
Mileage=50000, Age=7, C2=0, L1=0

Price=22540-0.09775*50000-668.5*7-1104*0-817.2*0+0.003513*50000*7= **$14198.31**

iii. Mileage=80000, Age=2, Color= White, location= St. Paul, do prediction.

Mileage=80000, Age=2, C2=0,L1=1

Price=22540-0.09775*80000-668.5*2-1104*0-817.2*1+0.003513*80000*2= **$13123.73**

What does this model tell us:

1. Can use model

Price=22540-0.09775*Mileage-668.5*Age-1104*C2-817.2*L1+0.003513*Mileage*Age

To predict sale price of a used Honda.

C2: 1 or 0; =1 means brown color, 0 means not brown (white, black, grey)

L1: 1 or 0; =1 means in St. Paul, 0 means not St. Paul (Durham, Santa Cruz)

2. The sale price of a used Honda depends on its mileage, age, color and also the location to sell it.

3. Once you drive your Honda, it loses value. (- 0.09775+ 0.003513*Age) *Mileage

4. Even you just park your Honda and never drive it, it loses value.
(- 668.5 + 0.003513*Mileage)*Age

5. If not really a huge fan of brown, choose a different color for your Honda. The car would worth a better price when you sell it. (- 1104*C2)

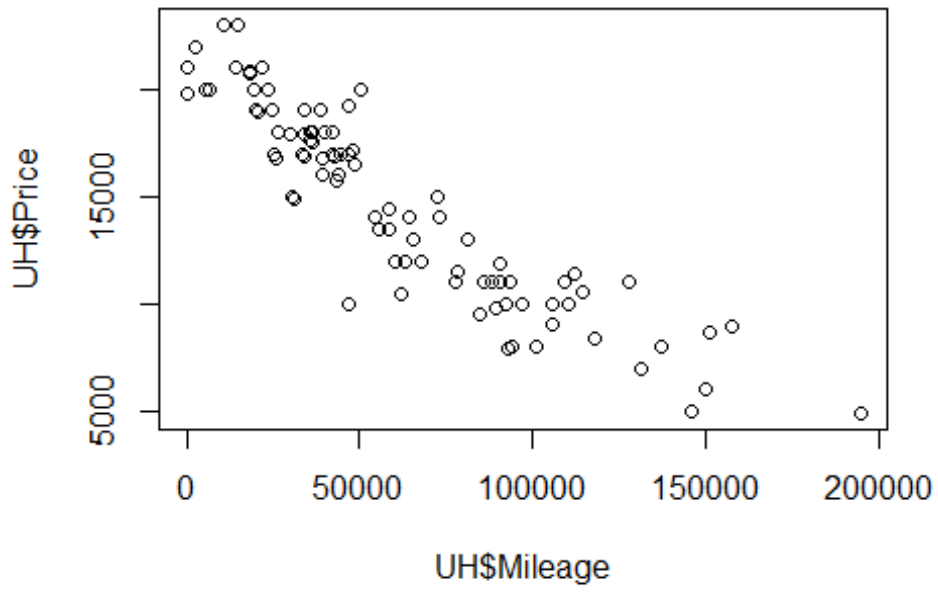
6. If it is possible, do not sell in St. Paul. Try Durham, Santa Cruz. (Long trip…)
(- 817.2*L1)

7. Noticed that, when mileage is more than 190,293 mile, or age is more than 27.83, (- 0.09775+ 0.003513*Age) and (- 668.5 + 0.003513*Mileage) can be more than 0. My opinion is that, none of the cases in the data set is from very heavily used or very old Honda, so, I assume that, if that kind of Honda, this model may not perform well. Would need further research and more data to build new model for that kind of situation.

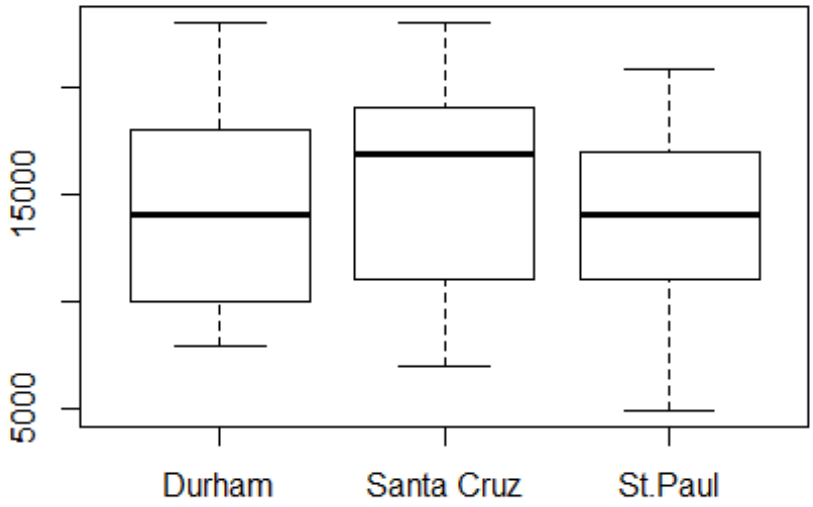
Appendix:

R Code Out put and codes:

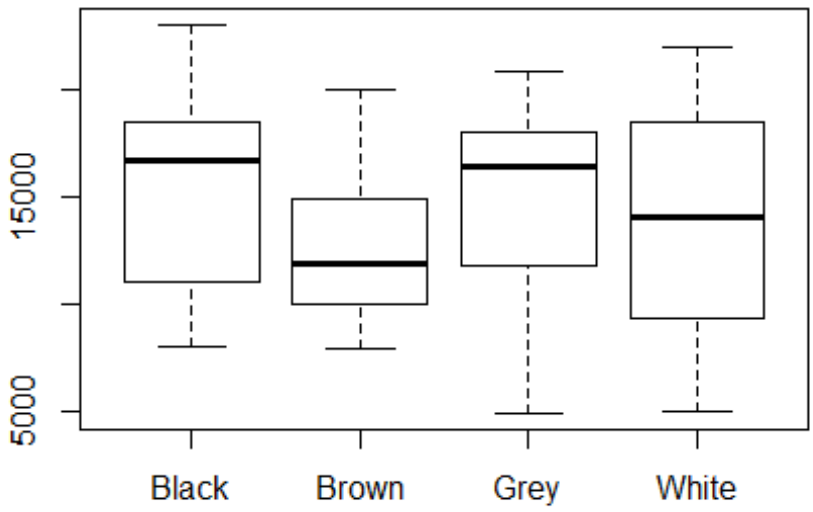
```
UH=read.csv("e:/used.csv")
save(UH,file ="e:/used_hondas.RData")
load("e:/used_hondas.RData")
head(UH)

##   Price Year Mileage Location Color Age
## 1 20746 2006   18394  St.Paul  Grey   1
## 2 19787 2007       8  St.Paul Black   0
## 3 17987 2005   39998  St.Paul  Grey   2
## 4 17588 2004   35882  St.Paul Black   3
## 5 16987 2004   25306  St.Paul  Grey   3
## 6 16987 2005   33399  St.Paul Black   2

#try to predict used hondas' price
#Use Age of the car make sence rather than use the year of the car mad
e.
#Plot data
plot(UH$Mileage,UH$Price)
```



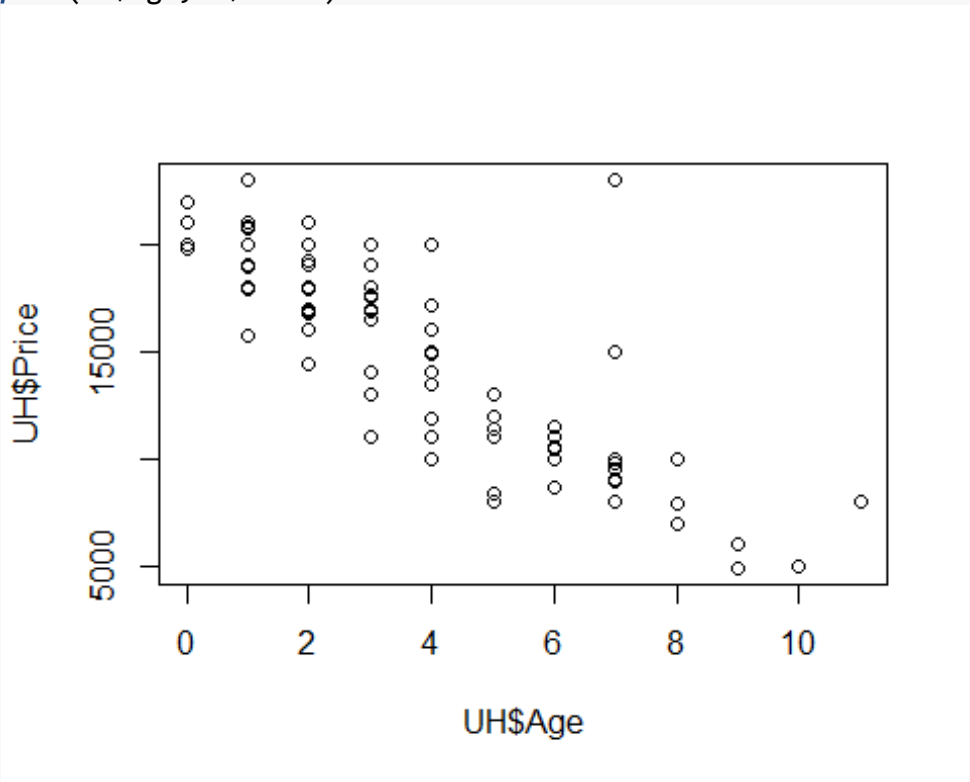

```
plot(UH$Location,UH$Price)
```




```
plot(UH$Color,UH$Price)
```



```
plot(UH$Age,UH$Price)
```




```
#correlation
cor(UH$Price,UH$Mileage)
```

```
## [1] -0.9126773
```

```
cor(UH$Price,UH$Age)
```

```
## [1] -0.8230599
```

```
cor(UH$Mileage,UH$Age)
```

```
## [1] 0.7796946
```

```
#depending on the graphs, it looks more like a first order model.
#Will test with first order moder, then add interactive terms.
#Will also, test second order model and third order(just give it a tr
y).
```

```
paste0(UH$Color)
```

```
##  [1] "Grey"  "Black" "Grey"  "Black" "Grey"  "Black" "Grey"  "Grey"
##  [9] "Black" "Brown" "Brown" "Grey"  "Grey"  "White" "Black" "Brown"
## [17] "White" "Black" "Brown" "Black" "Black" "Brown" "Brown" "Grey"
## [25] "Brown" "White" "Grey"  "White" "Grey"  "Grey"  "Black" "Grey"
## [33] "Brown" "Grey"  "Grey"  "White" "Brown" "Black" "Black" "Brown"
```

```
## [41] "Brown" "White" "Brown" "Black" "Black" "Black" "Grey"  "Brown"
## [49] "Grey"  "Black" "Black" "Black" "Black" "Black" "Black" "Black"
## [57] "Brown" "White" "White" "Brown" "Brown" "Black" "Brown" "White"
## [65] "Black" "Black" "Brown" "Black" "White" "Black" "Black" "White"
## [73] "Black" "Grey"  "Grey"  "Black" "Black" "White" "Black" "Grey"
## [81] "Brown" "Brown" "White" "White" "Black" "Black" "White" "Black"
## [89] "Black" "Brown" "Grey"  "Black"
```

```
#Take Color as dummy variable. "White" would be base level
UH$C1 = ifelse(UH$Color ==  "Black" , 1, 0)
UH$C2 = ifelse(UH$Color ==  "Brown" , 1, 0)
UH$C3 = ifelse(UH$Color ==  "Grey" , 1, 0)
```

```
paste0(UH$Location)
```

```
##  [1] "St.Paul"    "St.Paul"    "St.Paul"    "St.Paul"    "St.Paul"

##  [6] "St.Paul"    "St.Paul"    "St.Paul"    "St.Paul"    "St.Paul"

## [11] "St.Paul"    "St.Paul"    "St.Paul"    "St.Paul"    "St.Paul"

## [16] "St.Paul"    "St.Paul"    "St.Paul"    "St.Paul"    "St.Paul"

## [21] "St.Paul"    "St.Paul"    "St.Paul"    "St.Paul"    "St.Paul"

## [26] "St.Paul"    "St.Paul"    "St.Paul"    "St.Paul"    "Durham"

## [31] "Durham"     "Durham"     "Durham"     "Durham"     "Durham"

## [36] "Durham"     "Durham"     "Durham"     "Durham"     "Durham"

## [41] "Durham"     "Durham"     "Durham"     "Durham"     "Durham"

## [46] "Durham"     "Durham"     "Durham"     "Durham"     "Durham"

## [51] "Durham"     "Durham"     "Durham"     "Durham"     "Durham"

## [56] "Durham"     "Durham"     "Durham"     "Durham"     "Durham"

## [61] "Durham"     "Durham"     "Durham"     "Santa Cruz" "Santa Cruz
"
## [66] "Santa Cruz" "Santa Cruz" "Santa Cruz" "Santa Cruz" "Santa Cruz
"
## [71] "Santa Cruz" "Santa Cruz" "Santa Cruz" "Santa Cruz" "Santa Cruz
"
## [76] "Santa Cruz" "Santa Cruz" "Santa Cruz" "Santa Cruz" "Santa Cruz
"
## [81] "Santa Cruz" "Santa Cruz" "Santa Cruz" "Santa Cruz" "Santa Cruz
"
## [86] "Santa Cruz" "Santa Cruz" "Santa Cruz" "Santa Cruz" "Santa Cruz
```

```
"
## [91] "Santa Cruz" "Santa Cruz"

#Take Location as dummy variable. "Santa Cruz" would be base level
UH$L1 = ifelse(UH$Location ==  "St.Paul" , 1, 0)
UH$L2 = ifelse(UH$Location ==  "Durham" , 1, 0)


#Add possible interactive terms
UH$M_A=UH$Mileage*UH$Age

#Add possible Quadratic terms
UH$M_SQ=UH$Mileage^2
UH$A_SQ=UH$Age^2

#Add Cube terms
UH$M_CU=UH$Mileage^3
UH$A_CU=UH$Age^3


#First order model without interaction
library(leaps)

## Warning: package 'leaps' was built under R version 3.2.5

yvar = c("Price")
xvars = c("Mileage","Age","C1","C2","C3","L1","L2")
model=leaps( x=UH[,xvars], y=UH[,yvar], names=xvars, nbest=2, method="a
djr2")
model$which

##   Mileage   Age    C1    C2    C3    L1    L2
## 1    TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 1   FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 2    TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
## 2    TRUE FALSE FALSE  TRUE FALSE FALSE FALSE
## 3    TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE
## 3    TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE
## 4    TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE
## 4    TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE
## 5    TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
## 5    TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE
## 6    TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
## 6    TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
## 7    TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE

model$adjr2

##  [1] 0.8311241 0.6738435 0.8616191 0.8528467 0.8722074 0.8661978 0.8
758035
##  [8] 0.8721833 0.8750635 0.8747022 0.8739421 0.8739404 0.8728985
```

```
model=leaps( x=UH[,xvars], y=UH[,yvar], names=xvars, nbest=2, method="C
p")
model$which
```

```
##   Mileage   Age    C1    C2    C3    L1    L2
## 1    TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 1   FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 2    TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
## 2    TRUE FALSE FALSE  TRUE FALSE FALSE FALSE
## 3    TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE
## 3    TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE
## 4    TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE
## 4    TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE
## 5    TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
## 5    TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE
## 6    TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
## 6    TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
## 7    TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
model$Cp
```

```
##  [1]  31.580296 142.949984  10.898200  17.040820   4.478471   8.6393
23
##  [7]   3.011596   5.489553   4.535152   4.779554   6.302107   6.3032
25
## [13]   8.000000
```

```
#First order model with interaction
yvar = c("Price")
xvars = c("Mileage","Age","C1","C2","C3","L1","L2","M_A")
model=leaps( x=UH[,xvars], y=UH[,yvar], names=xvars, nbest=2, method="a
djr2")
model$which
```

```
##   Mileage   Age    C1    C2    C3    L1    L2   M_A
## 1    TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1   FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## 2    TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 2    TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
## 3    TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
## 3    TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
## 4    TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
## 4    TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE
## 5    TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE
## 5    TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE
## 6    TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE
## 6    TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
## 7    TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
## 7    TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## 8    TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
model$adjr2

##  [1] 0.8311241 0.7379639 0.8616191 0.8528467 0.8722074 0.8665477 0.8
758035
##  [8] 0.8753839 0.8809727 0.8768376 0.8810313 0.8798692 0.8800075 0.8
799120
## [15] 0.8789667

model=leaps( x=UH[,xvars], y=UH[,yvar], names=xvars, nbest=2, method="C
p")
model$which

##   Mileage   Age    C1    C2    C3    L1    L2   M_A
## 1    TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1   FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## 2    TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 2    TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
## 3    TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
## 3    TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
## 4    TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
## 4    TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE
## 5    TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE
## 5    TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE
## 6    TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE
## 6    TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
## 7    TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
## 7    TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## 8    TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE

model$Cp

##  [1]  37.575590 106.849214  15.756301  22.206889   8.914440  13.0294
57
##  [7]   7.273749   7.575318   4.574638   7.512768   5.550041   6.3661
50
## [13]   7.277661   7.343882   9.000000

#Second order model with interaction
yvar = c("Price")
xvars = c("Mileage","Age","C1","C2","C3","L1","L2","M_A","M_SQ","A_SQ")
model=leaps( x=UH[,xvars], y=UH[,yvar], names=xvars, nbest=2, method="a
djr2")
model$which

##    Mileage   Age    C1    C2    C3    L1    L2   M_A  M_SQ  A_SQ
## 1     TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1    FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## 2     TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
## 2     TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 3     TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
## 3     TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
```

```
## 4     TRUE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE
## 4     TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE
## 5     TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE
## 5     TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE
## 6     TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE
## 6     TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
## 7     TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
## 7     TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
## 8     TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
## 8     TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
## 9     TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
## 9     TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 10    TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE

model$adjr2

##  [1] 0.8311241 0.7379639 0.8670944 0.8616191 0.8915234 0.8868805 0.8
951834
##  [8] 0.8944591 0.8981579 0.8956571 0.8973291 0.8973144 0.8982066 0.8
964719
## [15] 0.8973214 0.8970304 0.8961196 0.8960818 0.8949378

model=leaps( x=UH[,xvars], y=UH[,yvar], names=xvars, nbest=2, method="C
p")
model$which

##    Mileage   Age    C1    C2    C3    L1    L2   M_A  M_SQ  A_SQ
## 1     TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1    FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## 2     TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
## 2     TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 3     TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
## 3     TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
## 4     TRUE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE
## 4     TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE
## 5     TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE
## 5     TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE
## 6     TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE
## 6     TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
## 7     TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
## 7     TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
## 8     TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
## 8     TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
## 9     TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
## 9     TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 10    TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE

model$Cp

##  [1]  56.665035 136.469328  26.586595  31.224843   6.859860  10.7487
30
```

```
##  [7]   4.796624   5.396353   3.364079   5.411136   5.065310   5.0771
59
## [13]   5.386517   6.773389   7.116892   7.346795   9.077568   9.1070
89
## [19]  11.000000

#Third order model with interaction
yvar = c("Price")
xvars = c("Mileage","Age","C1","C2","C3","L1","L2","M_A","M_SQ","A_SQ",
"M_CU","A_CU")
model=leaps( x=UH[,xvars], y=UH[,yvar], names=xvars, nbest=2, method="a
djr2")
model$which

##    Mileage   Age    C1    C2    C3    L1    L2   M_A  M_SQ  A_SQ  M_
CU
## 1     TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FAL
SE
## 1    FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FAL
SE
## 2     TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FAL
SE
## 2     TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TR
UE
## 3     TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FAL
SE
## 3     TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TR
UE
## 4     TRUE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FAL
SE
## 4     TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FAL
SE
## 5     TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TR
UE
## 5     TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE FAL
SE
## 6     TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TR
UE
## 6     TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE FAL
SE
## 7     TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TR
UE
## 7     TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TR
UE
## 8     TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TR
UE
## 8     TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TR
UE
## 9     TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TR
UE
```

```
## 9     TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE
## 10    TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE
## 10    TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE
## 11    TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE
## 11    TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE
## 12    TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE
##     A_CU
## 1  FALSE
## 1  FALSE
## 2  FALSE
## 2  FALSE
## 3  FALSE
## 3  FALSE
## 4  FALSE
## 4  FALSE
## 5  FALSE
## 5  FALSE
## 6   TRUE
## 6   TRUE
## 7   TRUE
## 7   TRUE
## 8   TRUE
## 8   TRUE
## 9   TRUE
## 9   TRUE
## 10  TRUE
## 10  TRUE
## 11  TRUE
## 11  TRUE
## 12  TRUE

model$adjr2

##  [1] 0.8311241 0.7379639 0.8670944 0.8666814 0.8915234 0.8894225 0.8
951834
##  [8] 0.8944591 0.8985078 0.8981579 0.9009696 0.9003231 0.9019836 0.9
016356
## [15] 0.9012810 0.9011858 0.9004581 0.9003756 0.8994090 0.8993249 0.8
983683
## [22] 0.8981569 0.8970950

model=leaps( x=UH[,xvars], y=UH[,yvar], names=xvars, nbest=2, method="C
p")
model$which
```

```
##    Mileage   Age    C1    C2    C3    L1    L2   M_A  M_SQ  A_SQ  M_
CU
## 1     TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FAL
SE
## 1    FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FAL
SE
## 2     TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FAL
SE
## 2     TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TR
UE
## 3     TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FAL
SE
## 3     TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TR
UE
## 4     TRUE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FAL
SE
## 4     TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FAL
SE
## 5     TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TR
UE
## 5     TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE FAL
SE
## 6     TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TR
UE
## 6     TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE FAL
SE
## 7     TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TR
UE
## 7     TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TR
UE
## 8     TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TR
UE
## 8     TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TR
UE
## 9     TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TR
UE
## 9     TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE
## 10    TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE
## 10    TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE
## 11    TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE
## 11    TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE
## 12    TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE
##     A_CU
## 1  FALSE
```

```
## 1  FALSE
## 2  FALSE
## 2  FALSE
## 3  FALSE
## 3  FALSE
## 4  FALSE
## 4  FALSE
## 5  FALSE
## 5  FALSE
## 6   TRUE
## 6   TRUE
## 7   TRUE
## 7   TRUE
## 8   TRUE
## 8   TRUE
## 9   TRUE
## 9   TRUE
## 10  TRUE
## 10  TRUE
## 11  TRUE
## 11  TRUE
## 12  TRUE

model$Cp

##  [1]  59.697716 141.174984  28.946801  29.304001   8.764598  10.5611
69
##  [7]   6.616182   7.228484   4.819289   5.111680   3.799594   4.3335
73
## [13]   4.009502   4.293579   5.623717   5.700476   7.320077   7.3858
62
## [19]   9.178601   9.244745  11.010118  11.174493  13.000000

#Consider both adjR^2 (should be high) and Cp (Cp=p and should be low),

#pick 12 models from all models above for further analysis.
model1=lm(Price~Mileage+Age+C2,data=UH)
model2=lm(Price~Mileage+Age+C2+L1,data=UH)
model3=lm(Price~Mileage+Age+C1+C2+C3+L1+L2,data=UH)

in.model1=lm(Price~Mileage+Age+C2+L1+M_A,data=UH)
in.model2=lm(Price~Mileage+Age+C2+L1+L2+M_A,data=UH)
in.model3=lm(Price~Mileage+Age+C1+C2+C3+L1+L2+M_A,data=UH)

sq.model1=lm(Price~Mileage+L1+M_A+M_SQ,data=UH)
sq.model2=lm(Price~Mileage+C2+M_A+M_SQ,data=UH)
sq.model3=lm(Price~Mileage+C2+L1+M_A+M_SQ,data=UH)
sq.model4=lm(Price~Mileage+Age+C1+C2+C3+L1+L2+M_A+M_SQ+A_SQ,data=UH)

cu.model1=lm(Price~Mileage+C2+L1+M_A+M_CU+A_CU,data=UH)
```

```
cu.model2=lm(Price~Mileage+Age+C1+C2+C3+L1+L2+M_A+M_SQ+A_SQ+M_CU+A_CU,data=UH)


#Check if betas are significant.
summary(model1)

##
## Call:
## lm(formula = Price ~ Mileage + Age + C2, data = UH)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3822.6  -952.1  -159.8   724.3  5506.0
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.149e+04  3.410e+02  63.019  < 2e-16 ***
## Mileage     -8.041e-02  6.765e-03 -11.886  < 2e-16 ***
## Age         -4.475e+02  1.176e+02  -3.806 0.000261 ***
## C2          -1.232e+03  4.256e+02  -2.894 0.004797 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1650 on 88 degrees of freedom
## Multiple R-squared:  0.8764, Adjusted R-squared:  0.8722
## F-statistic:   208 on 3 and 88 DF,  p-value: < 2.2e-16

summary(model2)

##
## Call:
## lm(formula = Price ~ Mileage + Age + C2 + L1, data = UH)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4052.7 -1071.2  -199.7   880.3  5290.3
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.169e+04  3.528e+02  61.478  < 2e-16 ***
## Mileage     -8.034e-02  6.669e-03 -12.046  < 2e-16 ***
## Age         -4.457e+02  1.159e+02  -3.844  0.00023 ***
## C2          -1.218e+03  4.196e+02  -2.902  0.00470 **
## L1          -6.879e+02  3.652e+02  -1.884  0.06296 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1627 on 87 degrees of freedom
```

```
## Multiple R-squared:  0.8813, Adjusted R-squared:  0.8758
## F-statistic: 161.4 on 4 and 87 DF,  p-value: < 2.2e-16
```

```
summary(model3)
```

```
##
## Call:
## lm(formula = Price ~ Mileage + Age + C1 + C2 + C3 + L1 + L2,
##     data = UH)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3972.0  -989.6  -152.0   746.6  5033.5
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.146e+04  5.525e+02  38.848  < 2e-16 ***
## Mileage     -8.020e-02  6.860e-03 -11.692  < 2e-16 ***
## Age         -4.409e+02  1.190e+02  -3.704 0.000378 ***
## C1           4.500e+02  5.097e+02   0.883 0.379810
## C2          -8.708e+02  5.777e+02  -1.507 0.135481
## C3           3.174e+02  5.764e+02   0.551 0.583329
## L1          -8.053e+02  4.492e+02  -1.793 0.076607 .
## L2          -2.346e+02  4.269e+02  -0.550 0.584023
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1646 on 84 degrees of freedom
## Multiple R-squared:  0.8827, Adjusted R-squared:  0.8729
## F-statistic: 90.28 on 7 and 84 DF,  p-value: < 2.2e-16
```

```
summary(in.model1)
```

```
##
## Call:
## lm(formula = Price ~ Mileage + Age + C2 + L1 + M_A, data = UH)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3845.2 -1001.9  -190.4   848.4  5927.8
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.254e+04  5.185e+02  43.468  < 2e-16 ***
## Mileage     -9.775e-02  1.030e-02  -9.489 4.96e-15 ***
## Age         -6.685e+02  1.526e+02  -4.382 3.31e-05 ***
## C2          -1.104e+03  4.141e+02  -2.666  0.00916 **
## L1          -8.172e+02  3.624e+02  -2.255  0.02667 *
## M_A          3.513e-03  1.607e-03   2.186  0.03154 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1593 on 86 degrees of freedom
## Multiple R-squared:  0.8875, Adjusted R-squared:  0.881
## F-statistic: 135.7 on 5 and 86 DF,  p-value: < 2.2e-16
```

```
summary(in.model2)
```

```
##
## Call:
## lm(formula = Price ~ Mileage + Age + C2 + L1 + L2 + M_A, data = UH)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3772.8  -973.6  -137.2   777.9  5853.3
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.262e+04  5.524e+02  40.948  < 2e-16 ***
## Mileage     -9.741e-02  1.038e-02  -9.387 8.87e-15 ***
## Age         -6.703e+02  1.533e+02  -4.372 3.47e-05 ***
## C2          -1.074e+03  4.211e+02  -2.551   0.0125 *
## L1          -9.193e+02  4.268e+02  -2.154   0.0341 *
## L2          -1.886e+02  4.115e+02  -0.458   0.6479
## M_A          3.500e-03  1.615e-03   2.168   0.0330 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1600 on 85 degrees of freedom
## Multiple R-squared:  0.8878, Adjusted R-squared:  0.8799
## F-statistic: 112.1 on 6 and 85 DF,  p-value: < 2.2e-16
```

```
summary(in.model3)
```

```
##
## Call:
## lm(formula = Price ~ Mileage + Age + C1 + C2 + C3 + L1 + L2 +
##     M_A, data = UH)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3761.9  -909.0  -182.3   830.2  5632.8
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.228e+04  6.459e+02  34.486  < 2e-16 ***
## Mileage     -9.897e-02  1.060e-02  -9.334 1.40e-14 ***
## Age         -6.724e+02  1.542e+02  -4.361 3.69e-05 ***
## C1           5.796e+02  5.006e+02   1.158   0.2502
## C2          -6.886e+02  5.694e+02  -1.209   0.2299
## C3           3.298e+02  5.625e+02   0.586   0.5592
## L1          -9.164e+02  4.410e+02  -2.078   0.0408 *
```

```
## L2          -2.195e+02  4.166e+02  -0.527   0.5996
## M_A          3.735e-03  1.636e-03   2.283   0.0250 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1606 on 83 degrees of freedom
## Multiple R-squared:  0.8896, Adjusted R-squared:  0.879
## F-statistic: 83.61 on 8 and 83 DF,  p-value: < 2.2e-16
```

```
summary(sq.model1)
```

```
##
## Call:
## lm(formula = Price ~ Mileage + L1 + M_A + M_SQ, data = UH)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5234.1  -999.6   -37.1   894.5  4632.2
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.256e+04  4.501e+02  50.111  < 2e-16 ***
## Mileage     -1.519e-01  1.311e-02 -11.587  < 2e-16 ***
## L1          -6.303e+02  3.395e+02  -1.857   0.0667 .
## M_A         -5.468e-03  1.246e-03  -4.387 3.21e-05 ***
## M_SQ         5.956e-07  8.404e-08   7.087 3.40e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1500 on 87 degrees of freedom
## Multiple R-squared:  0.8991, Adjusted R-squared:  0.8945
## F-statistic: 193.8 on 4 and 87 DF,  p-value: < 2.2e-16
```

```
summary(sq.model2)
```

```
##
## Call:
## lm(formula = Price ~ Mileage + C2 + M_A + M_SQ, data = UH)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4496.5 -1017.4   110.0   953.2  4590.7
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.233e+04  4.317e+02  51.742  < 2e-16 ***
## Mileage     -1.450e-01  1.328e-02 -10.915  < 2e-16 ***
## C2          -8.011e+02  3.970e+02  -2.018  0.04666 *
## M_A         -5.143e-03  1.269e-03  -4.053  0.00011 ***
## M_SQ         5.364e-07  8.914e-08   6.017  4.1e-08 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1494 on 87 degrees of freedom
## Multiple R-squared:  0.8998, Adjusted R-squared:  0.8952
## F-statistic: 195.3 on 4 and 87 DF,  p-value: < 2.2e-16
```

```
summary(sq.model3)
```

```
##
## Call:
## lm(formula = Price ~ Mileage + C2 + L1 + M_A + M_SQ, data = UH)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4719.6  -912.2    91.1   807.5  4399.6
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.256e+04  4.422e+02  51.025  < 2e-16 ***
## Mileage     -1.467e-01  1.312e-02 -11.178  < 2e-16 ***
## C2          -7.981e+02  3.913e+02  -2.040 0.044464 *
## L1          -6.275e+02  3.335e+02  -1.882 0.063249 .
## M_A         -4.862e-03  1.260e-03  -3.860 0.000219 ***
## M_SQ         5.342e-07  8.788e-08   6.079 3.22e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1473 on 86 degrees of freedom
## Multiple R-squared:  0.9038, Adjusted R-squared:  0.8982
## F-statistic: 161.5 on 5 and 86 DF,  p-value: < 2.2e-16
```

```
summary(sq.model4)
```

```
##
## Call:
## lm(formula = Price ~ Mileage + Age + C1 + C2 + C3 + L1 + L2 +
##     M_A + M_SQ + A_SQ, data = UH)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4444.8  -846.1    18.1   916.4  4810.1
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.273e+04  6.161e+02  36.893  < 2e-16 ***
## Mileage     -1.290e-01  1.854e-02  -6.960 8.07e-10 ***
## Age         -3.956e+02  2.998e+02  -1.319 0.190811
## C1           1.572e+02  4.803e+02   0.327 0.744327
## C2          -6.825e+02  5.346e+02  -1.277 0.205399
## C3           1.467e+02  5.268e+02   0.279 0.781331
## L1          -7.754e+02  4.128e+02  -1.878 0.063941 .
```

```
## L2          -2.138e+02  3.882e+02  -0.551 0.583374
## M_A         -7.353e-03  4.029e-03  -1.825 0.071663 .
## M_SQ         5.259e-07  1.376e-07   3.823 0.000258 ***
## A_SQ         5.334e+01  4.327e+01   1.233 0.221226
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1496 on 81 degrees of freedom
## Multiple R-squared:  0.9065, Adjusted R-squared:  0.8949
## F-statistic: 78.52 on 10 and 81 DF,  p-value: < 2.2e-16
```

```
summary(cu.model1)
```

```
##
## Call:
## lm(formula = Price ~ Mileage + C2 + L1 + M_A + M_CU + A_CU, data = U
H)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4971.5  -996.8    14.1   872.8  4199.3
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.175e+04  4.142e+02  52.522  < 2e-16 ***
## Mileage     -9.165e-02  1.392e-02  -6.585 3.58e-09 ***
## C2          -9.234e+02  3.839e+02  -2.405  0.01833 *
## L1          -7.739e+02  3.302e+02  -2.344  0.02142 *
## M_A         -9.664e-03  3.092e-03  -3.125  0.00243 **
## M_CU         2.308e-12  3.744e-13   6.164 2.29e-08 ***
## A_CU         4.123e+00  2.327e+00   1.771  0.08008 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1453 on 85 degrees of freedom
## Multiple R-squared:  0.9075, Adjusted R-squared:  0.901
## F-statistic:   139 on 6 and 85 DF,  p-value: < 2.2e-16
```

```
summary(cu.model2)
```

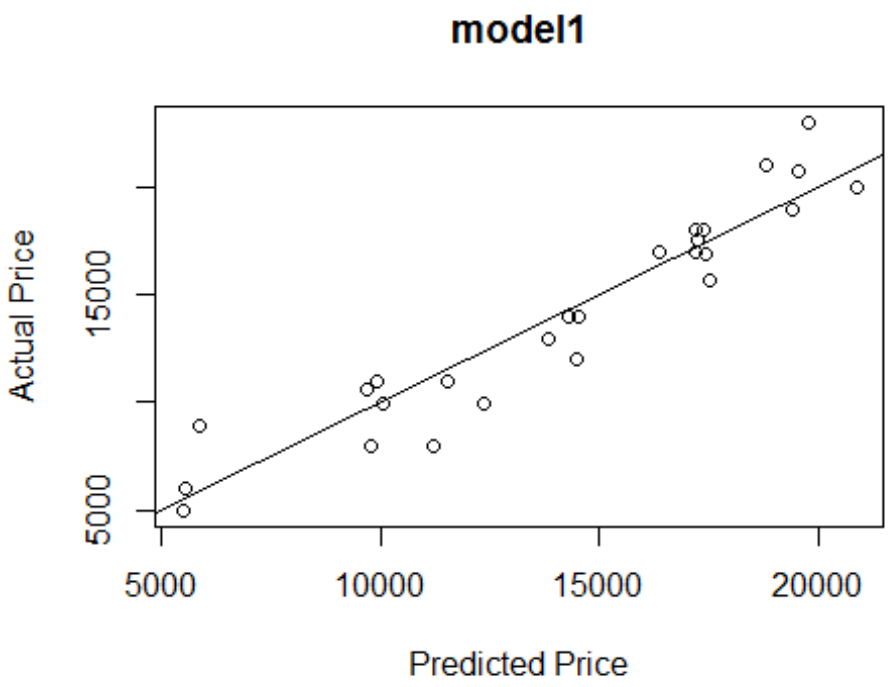
```
##
## Call:
## lm(formula = Price ~ Mileage + Age + C1 + C2 + C3 + L1 + L2 +
##     M_A + M_SQ + A_SQ + M_CU + A_CU, data = UH)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4343.1  -965.1  -130.5   789.6  4547.7
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  2.201e+04  7.156e+02  30.763  < 2e-16 ***
## Mileage     -1.149e-01  4.108e-02  -2.798  0.00646 **
## Age          2.828e+02  6.769e+02   0.418  0.67728
## C1           2.136e+02  4.795e+02   0.446  0.65713
## C2          -6.874e+02  5.414e+02  -1.270  0.20794
## C3           5.420e+01  5.389e+02   0.101  0.92013
## L1          -9.615e+02  4.199e+02  -2.290  0.02469 *
## L2          -2.116e+02  3.865e+02  -0.548  0.58553
## M_A         -6.252e-03  4.250e-03  -1.471  0.14525
## M_SQ         2.455e-07  4.519e-07   0.543  0.58851
## A_SQ        -1.218e+02  1.530e+02  -0.796  0.42852
## M_CU         1.061e-12  1.619e-12   0.656  0.51400
## A_CU         1.105e+01  8.439e+00   1.310  0.19414
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1481 on 79 degrees of freedom
## Multiple R-squared:  0.9107, Adjusted R-squared:  0.8971
## F-statistic: 67.11 on 12 and 79 DF,  p-value: < 2.2e-16

#The betas in following 3 models, all < .05, keep them
summary(model1)

##
## Call:
## lm(formula = Price ~ Mileage + Age + C2, data = UH)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3822.6  -952.1  -159.8   724.3  5506.0
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.149e+04  3.410e+02  63.019  < 2e-16 ***
## Mileage     -8.041e-02  6.765e-03 -11.886  < 2e-16 ***
## Age         -4.475e+02  1.176e+02  -3.806 0.000261 ***
## C2          -1.232e+03  4.256e+02  -2.894 0.004797 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1650 on 88 degrees of freedom
## Multiple R-squared:  0.8764, Adjusted R-squared:  0.8722
## F-statistic:   208 on 3 and 88 DF,  p-value: < 2.2e-16

summary(in.model1)

##
## Call:
## lm(formula = Price ~ Mileage + Age + C2 + L1 + M_A, data = UH)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max 
## -3845.2 -1001.9  -190.4   848.4  5927.8 
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  2.254e+04  5.185e+02  43.468  < 2e-16 ***
## Mileage     -9.775e-02  1.030e-02  -9.489 4.96e-15 ***
## Age         -6.685e+02  1.526e+02  -4.382 3.31e-05 ***
## C2          -1.104e+03  4.141e+02  -2.666  0.00916 ** 
## L1          -8.172e+02  3.624e+02  -2.255  0.02667 *  
## M_A          3.513e-03  1.607e-03   2.186  0.03154 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1593 on 86 degrees of freedom
## Multiple R-squared:  0.8875, Adjusted R-squared:  0.881 
## F-statistic: 135.7 on 5 and 86 DF,  p-value: < 2.2e-16

summary(sq.model2)

##
## Call:
## lm(formula = Price ~ Mileage + C2 + M_A + M_SQ, data = UH)
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4496.5 -1017.4   110.0   953.2  4590.7 
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  2.233e+04  4.317e+02  51.742  < 2e-16 ***
## Mileage     -1.450e-01  1.328e-02 -10.915  < 2e-16 ***
## C2          -8.011e+02  3.970e+02  -2.018  0.04666 *  
## M_A         -5.143e-03  1.269e-03  -4.053  0.00011 ***
## M_SQ         5.364e-07  8.914e-08   6.017  4.1e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1494 on 87 degrees of freedom
## Multiple R-squared:  0.8998, Adjusted R-squared:  0.8952 
## F-statistic: 195.3 on 4 and 87 DF,  p-value: < 2.2e-16

#Check multicollinearity
library(car)

## Warning: package 'car' was built under R version 3.2.5


vif(model1)
```

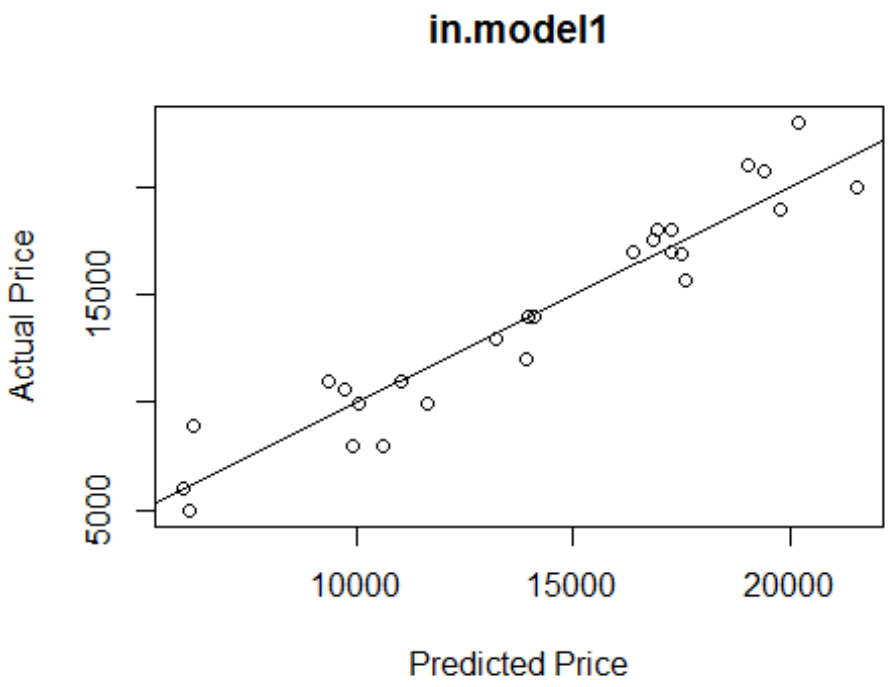
```
##  Mileage      Age       C2
## 2.645901 2.745880 1.077875

vif(in.model1)

##   Mileage       Age        C2        L1       M_A
##  6.587398  4.962061  1.095482  1.028312 11.741803

vif(sq.model2)

##   Mileage        C2       M_A      M_SQ
## 12.437452  1.143465  8.316363 15.266095

#because of the interactive term M_A, M_A and Mileage has a high VIF. I
t is reasonable.

#Use Training and testing partition of data to test models.
train.percent = .70
test.percent = .30
sample = sample(1:nrow(UH), train.percent * nrow(UH)); head(sample)

## [1] 84 40 68 44 49 30

train = UH[sample,]; head(train)

##    Price Year Mileage   Location Color Age C1 C2 C3 L1 L2    M_A
## 84 21910 2007    2637 Santa Cruz White   0  0  0  0  0  0      0
## 40 19995 2006   23533     Durham Brown   1  0  1  0  0  1  23533
## 68 19220 2005   46782 Santa Cruz Black   2  1  0  0  0  0  93564
## 44 10988 2001   85740     Durham Black   6  1  0  0  0  1 514440
## 49  9988 1999   96645     Durham  Grey   8  0  0  1  0  1 773160
## 30 14995 2003   30222     Durham  Grey   4  0  0  1  0  1 120888
##          M_SQ A_SQ         M_CU A_CU
## 84    6953769    0 1.833709e+10    0
## 40  553802089    1 1.303262e+13    1
## 68 2188555524    4 1.023850e+14    8
## 44 7351347600   36 6.303045e+14  216
## 49 9340256025   64 9.026890e+14  512
## 30  913369284   16 2.760385e+13   64

test = UH[-sample,]; head(test)

##    Price Year Mileage Location Color Age C1 C2 C3 L1 L2    M_A
 M_SQ
## 1  20746 2006   18394  St.Paul  Grey   1  0  0  1  1  0  18394  3383
39236
## 3  17987 2005   39998  St.Paul  Grey   2  0  0  1  1  0  79996 15998
40004
## 4  17588 2004   35882  St.Paul Black   3  1  0  0  1  0 107646 12875
17924
## 8  13987 2003   64495  St.Paul  Grey   4  0  0  1  1  0 257980 41596
05025
```

```
## 10 10987 2001   77665  St.Paul Brown   6  0  1  0  1  0 465990 60318
52225
## 15  9995 2003   92097  St.Paul Black   4  1  0  0  1  0 368388 84818
57409
##    A_SQ         M_CU A_CU
## 1     1 6.223412e+12    1
## 3     4 6.399040e+13    8
## 4     9 4.619872e+13   27
## 8    16 2.682737e+14   64
## 10   36 4.684638e+14  216
## 15   16 7.811536e+14   64

evaluate_model <- function(description, formula, plot=TRUE) {
  train.fit = lm(formula, data=train)
  train.summary = summary(train.fit)
  Price_Hat = predict(train.fit, test)  # fit test data using train mod
el
  cor.Price_hat.Price = cor(Price_Hat, test$Price)
  if (plot==TRUE) {
    plot(Price_Hat, test$Price, main=description, xlab="Predicted Price
", ylab="Actual Price")
    abline(0,1)  # 45 degree angle, cosmetic
  }
  train.rmse = train.summary$sigma
  predictors = dim(train.summary$coefficients)[1]  # includes beta0
  test.df = nrow(test) - predictors  # degrees of freedom
  test.rmse = sqrt(sum((test$Price - Price_Hat) ^ 2) / (test.df))
  percent.error = (test.rmse - train.rmse) / train.rmse * 100
  dat = data.frame(description, cor.Price_hat.Price, train.rmse, test.r
mse, percent.error)
  return(dat)
}

evaluate_model("model1", model1)
```



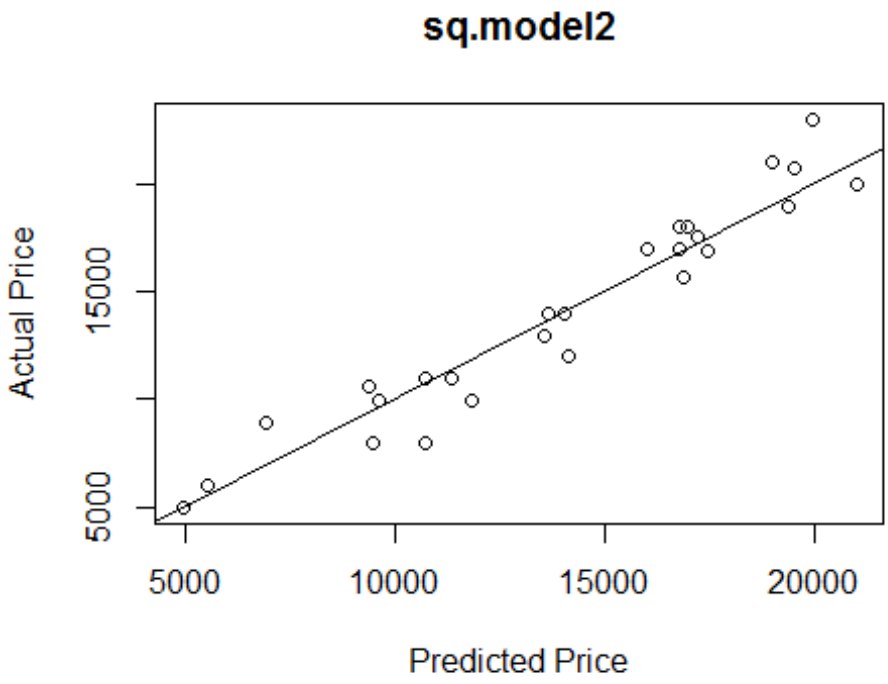

```
##   description cor.Price_hat.Price train.rmse test.rmse percent.error
## 1      model1           0.9455067   1677.173  1732.109      3.275541
```

```
evaluate_model("in.model1", in.model1)
```

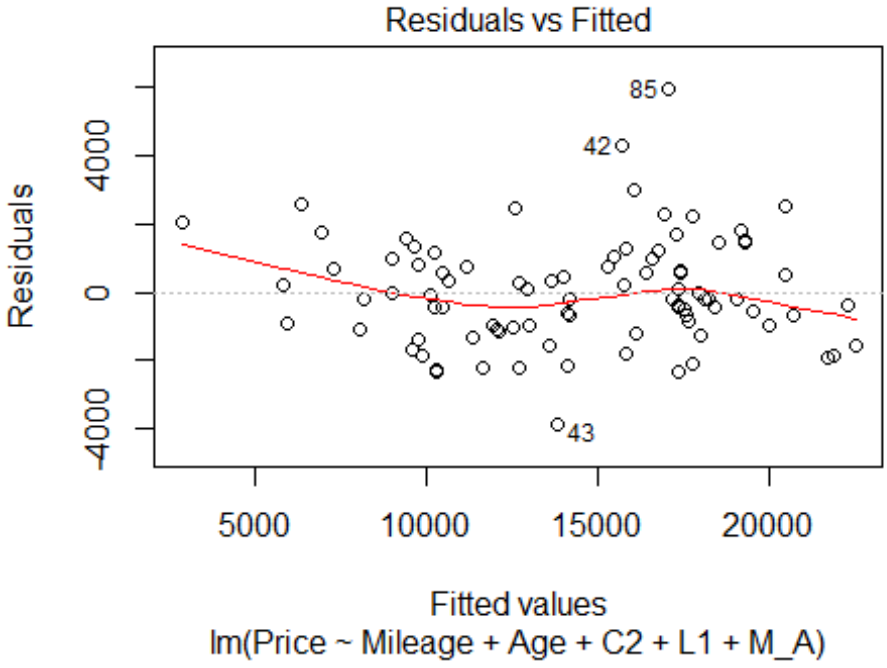


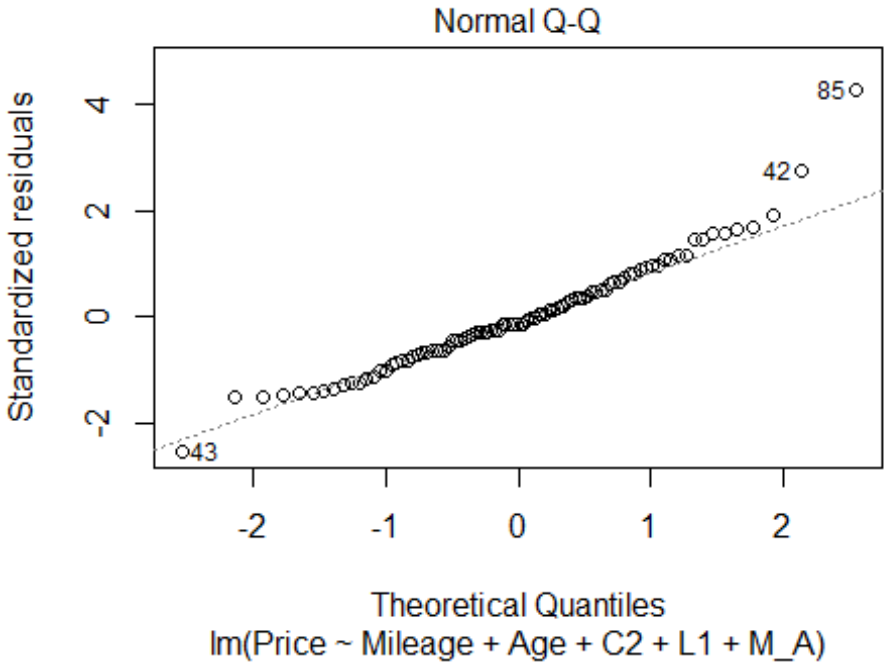

```
##   description cor.Price_hat.Price train.rmse test.rmse percent.error
## 1   in.model1           0.9556649   1675.764  1629.282     -2.773752
```

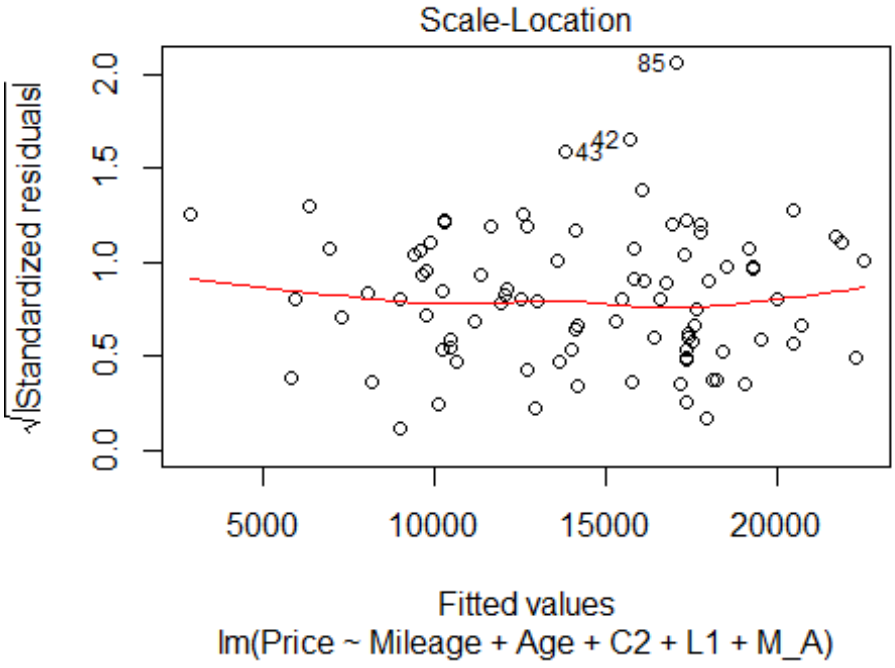
```
evaluate_model("sq.model2", sq.model2)
```

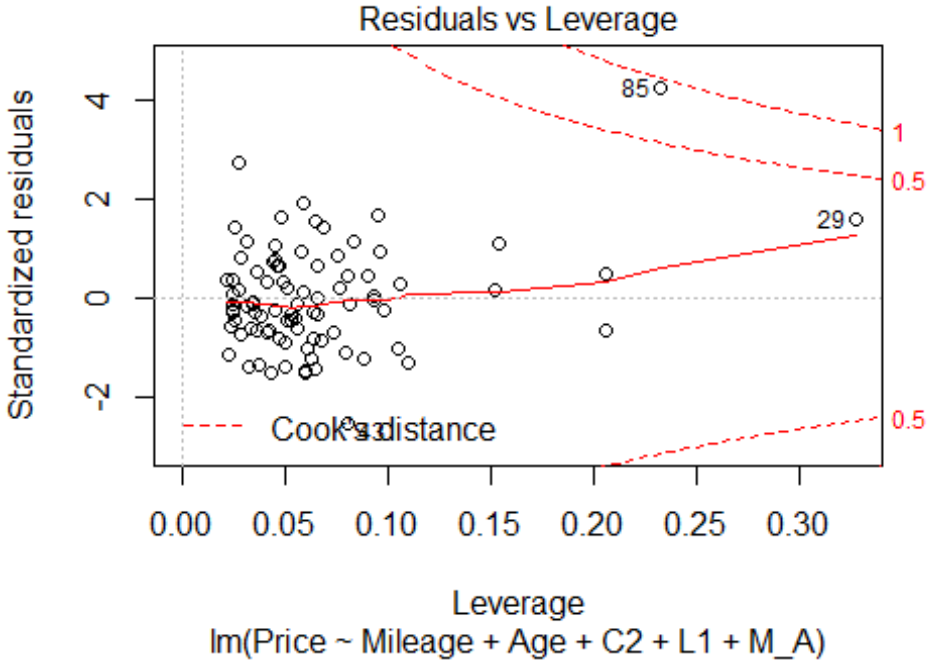



```
##   description cor.Price_hat.Price train.rmse test.rmse percent.error
## 1   sq.model2           0.9603641   1565.116  1519.912     -2.888264

#Pick up in.model1: Price~Mileage+Age+C2+L1+M_A as the best model becau
se of low percent.error.


#heteroscedasticity
library(MASS)
plot(in.model1)
```


Residuals vs Fitted
Residuals
4000
0
-4000
85
42
43
5000
10000
15000
20000
Fitted values
lm(Price ~ Mileage + Age + C2 + L1 + M_A)

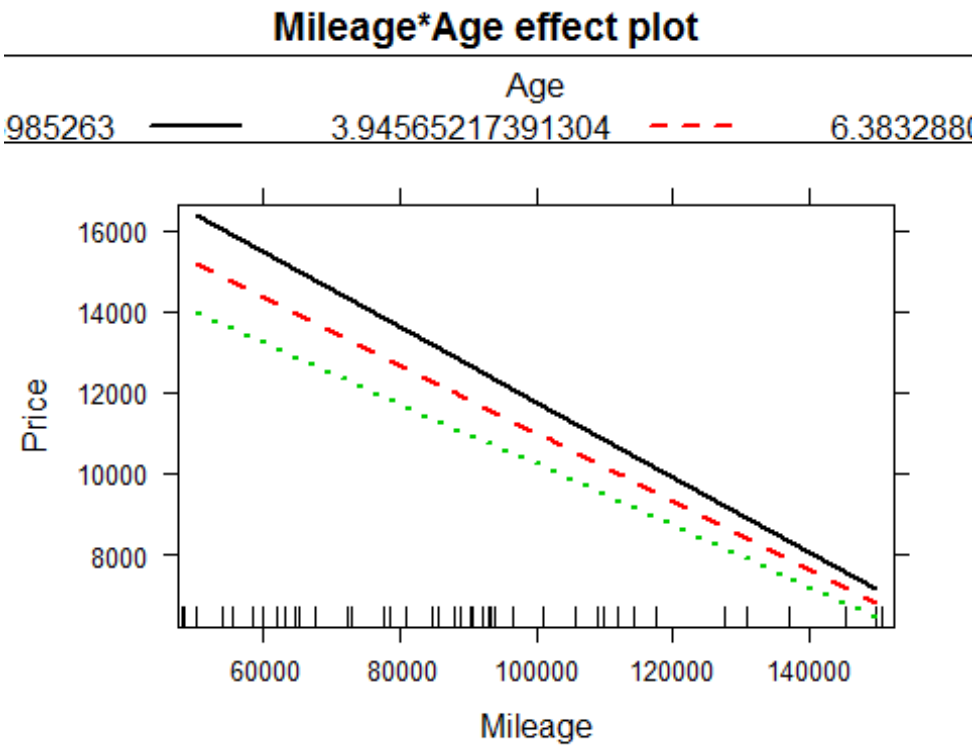

Normal Q-Q
Standardized residuals
4
2
0
-2
85
42
43
-2
-1
0
1
2
Theoretical Quantiles
lm(Price ~ Mileage + Age + C2 + L1 + M_A)






```
standardized.residuals = rstandard(in.model1)

standardized.residuals[85-1]
```

```
##         84
## -0.2436802

standardized.residuals[43-1]

##      42
## 2.74549

standardized.residuals[42-1]

##       41
## 1.448006

standardized.residuals[29-1]

##         28
## -0.6502143

#None of the point's Cook's distance > 1
#None of the point locates out of 3s.
#So, could be treated as no outlier, no infuencial points.


#Plot model
require(effects)

## Loading required package: effects

## Warning: package 'effects' was built under R version 3.2.5

##
## Attaching package: 'effects'

## The following object is masked from 'package:car':
##
##     Prestige

mean = mean(UH$Age)
sd = sd(UH$Age)
fit=lm(Price~Mileage+Age+C2+L1+Mileage:Age,data=UH)
plot(effect("Mileage:Age", fit,, list(Age=c(mean-sd, mean, mean+sd))),
multiline=TRUE)
```




```
summary(fit)

##
## Call:
## lm(formula = Price ~ Mileage + Age + C2 + L1 + Mileage:Age, data = U
H)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3845.2 -1001.9  -190.4   848.4  5927.8
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.254e+04  5.185e+02  43.468  < 2e-16 ***
## Mileage     -9.775e-02  1.030e-02  -9.489 4.96e-15 ***
## Age         -6.685e+02  1.526e+02  -4.382 3.31e-05 ***
## C2          -1.104e+03  4.141e+02  -2.666  0.00916 **
## L1          -8.172e+02  3.624e+02  -2.255  0.02667 *
## Mileage:Age  3.513e-03  1.607e-03   2.186  0.03154 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1593 on 86 degrees of freedom
## Multiple R-squared:  0.8875, Adjusted R-squared:  0.881
## F-statistic: 135.7 on 5 and 86 DF,  p-value: < 2.2e-16
```

```
#The final model for predicting used Honda is:
#Price=22540-0.09775*Mileage-668.5*Age-1104*C2-817.2*L1+0.003513*Mileag
e*Age


#Predict Price:
#i. Mileage=20000, Age=1, Color= Brown, location= Durham,
#Mileage=20000, Age=1, C2=1,L1=0
predict1=predict(fit, data.frame(Mileage=20000, Age=1, C2=1,L1=0))
print(predict1)

##        1
## 18878.83

#Mileage=50000, Age=7, Color= Black, location= Santa Cruz
#Mileage=50000, Age=7, C2=0, L1=0
predict2=predict(fit, data.frame(Mileage=50000, Age=7, C2=0,L1=0))
print(predict2)

##        1
## 14198.31

#Mileage=80000, Age=2, Color= White, location= St. Paul
#Mileage=80000, Age=2, C2=0,L1=1
predict3=predict(fit, data.frame(Mileage=80000, Age=2, C2=0,L1=1))
print(predict3)

##        1
## 13123.73
```