STOR 455 Homework 3

20 points - Due Thursday 2/23 at 12:00pm

Situation: Suppose that you are interested in purchasing a used vehicle. How much should you expect to pay? Obviously the price will depend on the type of vehicle that you get (the model) and how much it's been used. For this assignment you will investigate how the price might depend on the vehicle's year and mileage.

Data Source: To get a sample of vehicles, begin with the UsedCars CSV file. The data was acquired by scraping TrueCar.com for used vehicle listings on 9/24/2017 and contains more than 1.2 million used vehicles. For this assignment you will choose a vehicle *Model* from a US company for which there are at least 100 of that model listed for sale in North Carolina. Note that whether the companies are US companies or not is not contained within the data. It is up to you to determine which *Make* of vehicles are from US companies. After constructing a subset of the UsedCars data under these conditions, check to make sure that there is a reasonable amount of variability in the years for your vehicle, with a range of at least six years.

Directions: The code below should walk you through the process of selecting data from a particular model vehicle of your choice. Each of the following two R chunks begin with {r, eval=FALSE}. eval=FALSE makes these chunks not run when I knit the file. Before you knit these chunks, you should revert them to {r}.

```
library(readr)
library(car)
```

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

```
library(leaps)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v dplyr   1.0.10
## v tibble  3.1.8      v stringr 1.5.0
## v tidyr   1.3.0      v forcats 0.5.2
## v purrr   1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::recode() masks car::recode()
## x purrr::some()   masks car::some()
```

```
# This line will only run if the UsedCars.csv is stored in the same directory as this notebook!
UsedCars <- read_csv("UsedCars.csv")
```

```
## Rows: 1048575 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): City, State, Vin, Make, Model
## dbl (4): Id, Price, Year, Mileage
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
StateHW2 = "NC"

# Creates a dataframe with the number of each model for sale in North Carolina
Vehicles = as.data.frame(table(UsedCars$Model[UsedCars$State==StateHW2]))
```

```r
# Renames the variables
names(Vehicles)[1] = "Model"
names(Vehicles)[2] = "Count"

# Restricts the data to only models with at least 100 for sale
# Vehicles from non US companies are contained in this data
# Before submitting, comment this out so that it doesn't print while knitting
#Enough_Vehicles = subset(Vehicles, Count>=100)
#Enough_Vehicles
```

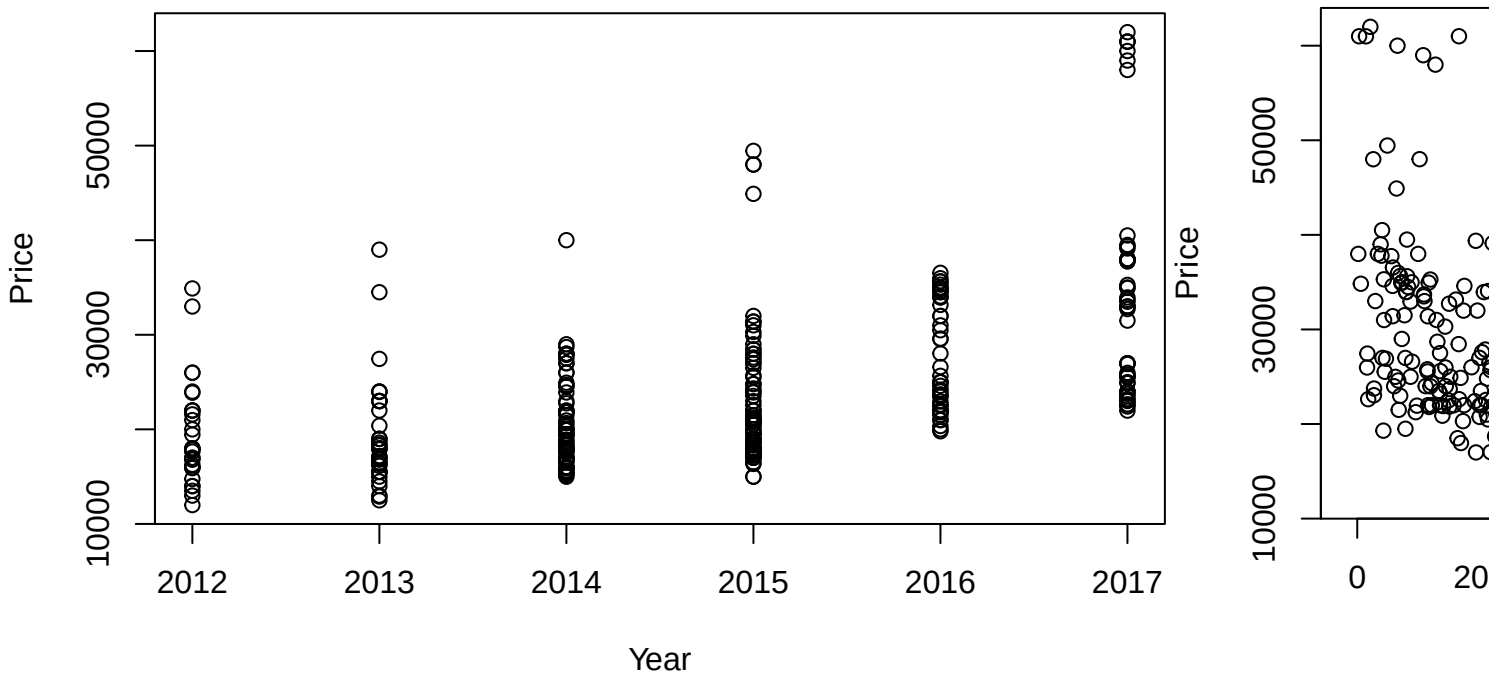
```r
# Delete the ** below and enter the model that you chose from the Enough_Vehicles data.
ModelOfMyChoice = "CamaroCoupe"

# Takes a subset of your model vehicle from North Carolina
MyVehicles = subset(UsedCars, Model==ModelOfMyChoice & State==StateHW2)

# Check to make sure that the vehicles span at least 6 years.
range(MyVehicles$Year)
```

```
## [1] 2012 2017
```

MODEL #4: Use Year and Mileage as predictors for Price

1. Construct a model using two predictors (*Year* and *Mileage*) with *Price* as the response variable and provide the summary output. Comment on the diagnostic plots.

```
CamaroModel = lm(Price~Year+Mileage, data = MyVehicles)
plot(Price~Year+Mileage, data = MyVehicles)
```




```
summary(CamaroModel)
```

```
##
## Call:
## lm(formula = Price ~ Year + Mileage, data = MyVehicles)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9552  -5008  -1974   3709  30686
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.414e+06  6.793e+05  -5.026 8.38e-07 ***
## Year         1.709e+03  3.369e+02   5.071 6.73e-07 ***
## Mileage     -1.444e-01  2.271e-02  -6.359 7.01e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7006 on 319 degrees of freedom
## Multiple R-squared:  0.3644, Adjusted R-squared:  0.3605
## F-statistic: 91.46 on 2 and 319 DF,  p-value: < 2.2e-16
```

The Price~Mileage plot has a negative correlation vs the Price~Year plot which has a positive correlation. The data of the Price~Year graph has a linear shape with some price outliers at the top of each year. The data of the Price~Mileage plot looks like a decaying exponential curve, so a linear curve won't fit well leading to the graph having a low correlation.

2. Assess the importance of each of the predictors in the regression model - be sure to indicate the specific value(s) from the summary output you are using to make the assessments. Include hypotheses and conclusions in

context.

```
cor.test(MyVehicles$Price, MyVehicles$Year)
```

```
##
##  Pearson's product-moment correlation
##
## data:  MyVehicles$Price and MyVehicles$Year
## t = 11.262, df = 320, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4496785 0.6067559
## sample estimates:
##       cor
## 0.5327908
```

```
cor.test(MyVehicles$Price, MyVehicles$Mileage)
```

```
##
##  Pearson's product-moment correlation
##
## data:  MyVehicles$Price and MyVehicles$Mileage
## t = -12.08, df = 320, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6303916 -0.4796980
## sample estimates:
##        cor
## -0.5596537
```

```
cor.test(MyVehicles$Year, MyVehicles$Mileage)
```

```
##
##  Pearson's product-moment correlation
##
## data:  MyVehicles$Year and MyVehicles$Mileage
## t = -14.975, df = 320, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.7019417 -0.5727757
## sample estimates:
##        cor
## -0.6418897
```

Both year and mileage are accurate in predicting the price of a car. A t-test was used to see if there was a relationship between year, price, and mileage. The Ho states that p = 0 and there is no correlation between year, price, and mileage. The Ha states that p is != 0 and there is a correlation between year, price, and mileage. The p-value for the correlation between all of these variables is less than 2.2e-16 but still greater than 0. Knowing this we reject the null and can assume that year, price, and mileage is correlated.

3. Assess the overall effectiveness of this model (with a formal test). Again, be sure to include hypotheses and the specific value(s) you are using from the summary output to reach a conclusion.

```
summary(CamaroModel)
```

```
##
## Call:
## lm(formula = Price ~ Year + Mileage, data = MyVehicles)
##
## Residuals:
```

```
##    Min     1Q Median     3Q    Max
##  -9552  -5008  -1974   3709  30686
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.414e+06  6.793e+05  -5.026 8.38e-07 ***
## Year         1.709e+03  3.369e+02   5.071 6.73e-07 ***
## Mileage     -1.444e-01  2.271e-02  -6.359 7.01e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7006 on 319 degrees of freedom
## Multiple R-squared:  0.3644, Adjusted R-squared:  0.3605
## F-statistic: 91.46 on 2 and 319 DF,  p-value: < 2.2e-16
```

Using a t test, the R-squared value of 0.3644, and a p-value around 0, I was able to see how effective my model was. Ho, p-value = 0, Ha, p-value !=0. The R value of 0.6036555 is decently high showing that there is good correlation between Year+Mileage and Price. The p-value being 2.2e-16 is extremely small and close to 0 which allows us to reject the null hypothesis and conclude a linear relationship with our model. We can also conclude that our model appears effective from the R and R-squared values and the p-value.

4. Compute and interpret the variance inflation factor (VIF) for your predictors.

```
vif(CamaroModel)
```

```
##     Year  Mileage
## 1.700745 1.700745
```

The VIF values are between 1 and 5 meaning that there is some correlation between Price~Year and Price~Mileage. The variance for both of these variables is about 70% bigger than what you would expect if there was no multi-collinearity.

5. Suppose that you are interested in purchasing a car of this model that is from the year 2017 with 50K miles. Determine each of the following: a 95% confidence interval for the mean price at this year and odometer reading, and a 95% prediction interval for the price of an individual car at this year and odometer reading. Write sentences that carefully interpret each of the intervals (in terms of car prices).

```
car = data.frame(Year = 2017, Mileage = 50000)
predict.lm(CamaroModel, car, interval = "confidence", level = 0.95)
```

```
##        fit      lwr      upr
## 1 25181.73 22863.81 27499.65
```

```
predict.lm(CamaroModel, car, interval = "prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 25181.73 11204.54 39158.92
```

We are 95% confident that the mean price of a 2017 Camaro Coupe with 50k miles is between 22,863.81 and 27,499.65
We are 95% confident that the next vehicle price will fall within 11,204.54 and 39,158.92

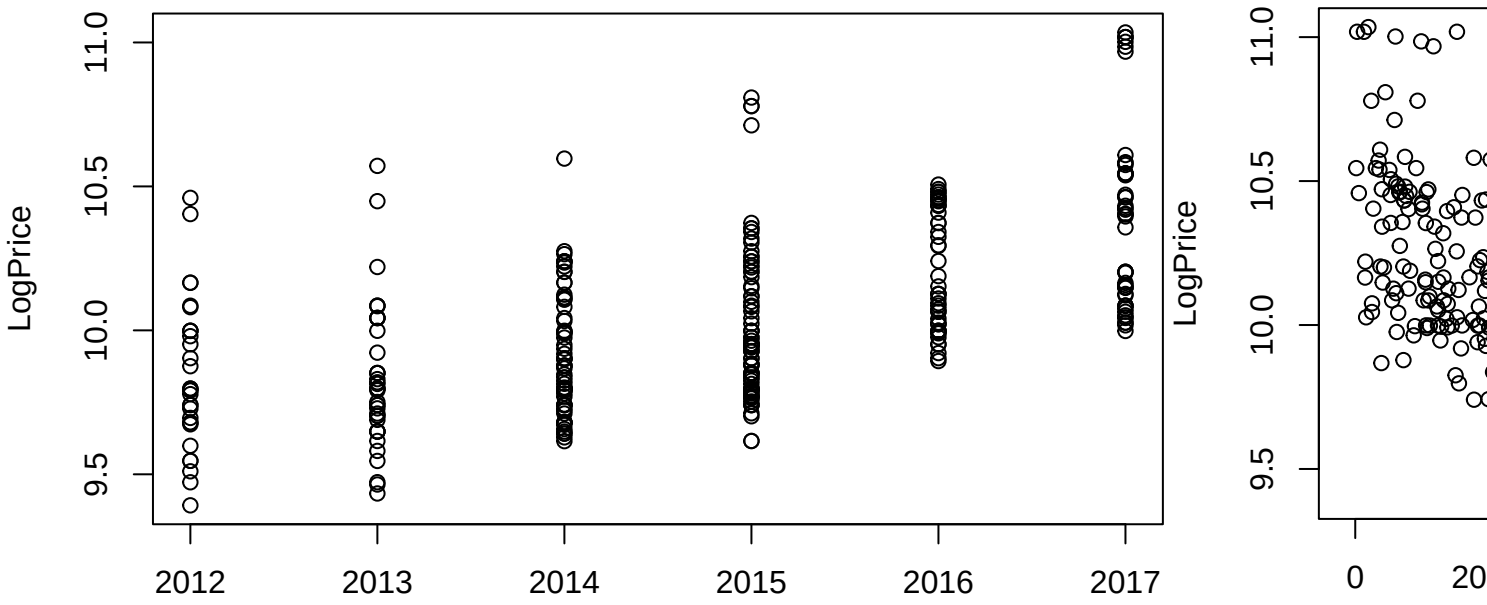
MODEL #5: Use Year, Mileage, Mileage^2 and Mileage^3 as predictors for log(Price)

6. Add a column of *logPrice* as the (natural) logarithm of the prices. Construct a model using two predictors (*Year* and *Mileage*) with *logPrice* as the response variable and provide the summary output. Comment on the diagnostic plots.

```
LogPrice = log(MyVehicles$Price)
MyVehicles = cbind(MyVehicles, LogPrice)

LogPriceMod = lm(LogPrice~Year+Mileage, data = MyVehicles)
summary(LogPriceMod)
```

```
##
## Call:
## lm(formula = LogPrice ~ Year + Mileage, data = MyVehicles)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40328 -0.17622 -0.06736  0.17421  0.74521
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.114e+02  2.256e+01  -4.938 1.27e-06 ***
## Year         6.038e-02  1.119e-02   5.395 1.34e-07 ***
## Mileage     -6.145e-06  7.542e-07  -8.148 8.47e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2327 on 319 degrees of freedom
## Multiple R-squared:  0.4475, Adjusted R-squared:  0.4441
## F-statistic: 129.2 on 2 and 319 DF,  p-value: < 2.2e-16
```

```
plot(LogPrice~Year+Mileage, data = MyVehicles)
```

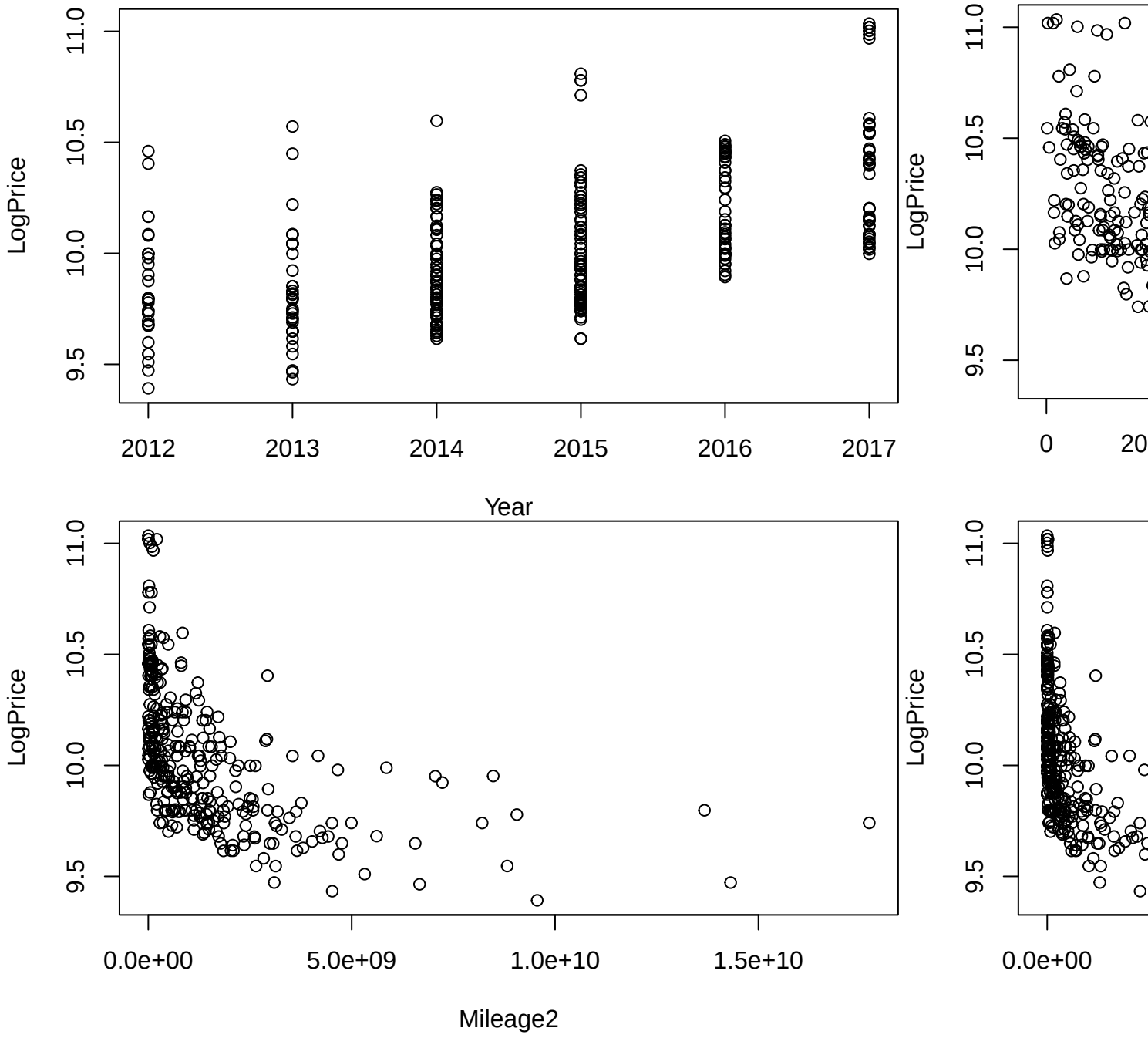


Like above the LogPrice~Year plot still has a negative correlation and the LogPrice~Mileage plot still has a positive correlation. The LogPrice~Year graph is shifted upwards compared to the Price~Year graph and the data points look a little more clustered together for each year. The LogPrice~Year graph still has a linear shape with some price outliers at the top of each year. With the transformed LogPrice~Mileage graph the data looks more linear rather than a decaying exponential curve like Price~Mileage above. This higher linear correlation is also proven with the R-squared being higher than the R-squared of the CamaroModel.

7. Add two columns of *Mileage2* and *Mileage3* as Mileage^2 and Mileage^3 respectively. Construct a model using four predictors (*Year*,*Mileage*, *Mileage2* and *Mileage3*) with *logPrice* as the response variable and provide the summary output. Call this model *Full*. Comment on the diagnostic plots.

```
Mileage2 = (MyVehicles$Mileage)^2
Mileage3 = (MyVehicles$Mileage)^3
MyVehicles = cbind(MyVehicles, Mileage2, Mileage3)
```

```
Full = lm(LogPrice~Year+Mileage+Mileage2+Mileage3, data = MyVehicles)
plot(LogPrice~Year+Mileage+Mileage2+Mileage3, data = MyVehicles)
```




```
summary(Full)
```

```
##
## Call:
## lm(formula = LogPrice ~ Year + Mileage + Mileage2 + Mileage3,
##     data = MyVehicles)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44983 -0.16660 -0.04111  0.13965  0.75988
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.283e+01  2.191e+01  -3.781 0.000187 ***
## Year         4.628e-02  1.086e-02   4.260  2.7e-05 ***
## Mileage     -2.039e-05  3.267e-06  -6.240  1.4e-09 ***
## Mileage2     2.321e-10  7.100e-11   3.268 0.001201 **
## Mileage3    -8.697e-16  4.198e-16  -2.072 0.039078 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2207 on 317 degrees of freedom
## Multiple R-squared:  0.5061, Adjusted R-squared:  0.4998
## F-statistic:  81.2 on 4 and 317 DF,  p-value: < 2.2e-16
```

Both the Price~Mileage2 and the Price~Mileage3 graphs both still have a negative correlation like the Price~Mileage graph. The big difference between these new transformed Mileage columns and the original Mileage Column is that they look more clustered due to the axes being stretched out to account for the outliers at the edge of the x-axes.

8. Select a subset of predictors in *Full* using each of the four methods: all subsets, backward elimination, forward selection, and stepwise regression. Use Mallows' Cp (AIC) as the criterion.

```
# All Subsets
all = regsubsets(LogPrice~Year+Mileage+Mileage2+Mileage3, data = MyVehicles)
summary(all)
```

```
## Subset selection object
## Call: regsubsets.formula(LogPrice ~ Year + Mileage + Mileage2 + Mileage3,
##     data = MyVehicles)
## 4 Variables  (and intercept)
##          Forced in Forced out
## Year         FALSE      FALSE
## Mileage      FALSE      FALSE
## Mileage2     FALSE      FALSE
## Mileage3     FALSE      FALSE
## 1 subsets of each size up to 4
## Selection Algorithm: exhaustive
##          Year Mileage Mileage2 Mileage3
## 1  ( 1 ) " "  "*"     " "      " "
## 2  ( 1 ) " "  "*"     "*"      " "
## 3  ( 1 ) "*"  "*"     "*"      " "
## 4  ( 1 ) "*"  "*"     "*"      "*"
```

```
# Backward elimination
MSE = (summary(Full)$sigma)^2
step(Full, scale=MSE)
```

```
## Start:  AIC=5
## LogPrice ~ Year + Mileage + Mileage2 + Mileage3
##
##            Df Sum of Sq    RSS     Cp
## <none>                  15.443  5.000
## - Mileage3  1   0.20914 15.652  7.293
## - Mileage2  1   0.52038 15.964 13.682
## - Year      1   0.88401 16.327 21.146
## - Mileage   1   1.89675 17.340 41.934

##
## Call:
## lm(formula = LogPrice ~ Year + Mileage + Mileage2 + Mileage3,
##     data = MyVehicles)
```

```
##
## Coefficients:
## (Intercept)         Year      Mileage     Mileage2     Mileage3
##  -8.283e+01    4.628e-02   -2.039e-05    2.321e-10   -8.697e-16
```

```
# Forward selection
none = lm(LogPrice~1, data = MyVehicles)
step(none, scope = list(upper=Full), scale = MSE, direction = "forward")
```

```
## Start:  AIC=321.79
## LogPrice ~ 1
##
##             Df Sum of Sq    RSS      Cp
## + Mileage    1   12.4164 18.849  68.918
## + Year       1   10.3972 20.869 110.366
## + Mileage2   1    7.1801 24.086 176.402
## + Mileage3   1    3.7444 27.521 246.924
## <none>                   31.266 321.786
##
## Step:  AIC=68.92
## LogPrice ~ Mileage
##
##             Df Sum of Sq    RSS     Cp
## + Mileage2   1    2.2614 16.588 24.499
## + Mileage3   1    1.8526 16.997 32.891
## + Year       1    1.5761 17.273 38.567
## <none>                   18.849 68.918
##
## Step:  AIC=24.5
## LogPrice ~ Mileage + Mileage2
##
##             Df Sum of Sq    RSS     Cp
## + Year       1   0.93565 15.652  7.293
## + Mileage3   1   0.26078 16.327 21.146
## <none>                   16.588 24.499
##
## Step:  AIC=7.29
## LogPrice ~ Mileage + Mileage2 + Year
##
##             Df Sum of Sq    RSS    Cp
## + Mileage3   1   0.20914 15.443 5.000
## <none>                   15.652 7.293
##
## Step:  AIC=5
## LogPrice ~ Mileage + Mileage2 + Year + Mileage3
```

```
##
## Call:
## lm(formula = LogPrice ~ Mileage + Mileage2 + Year + Mileage3,
##     data = MyVehicles)
##
## Coefficients:
## (Intercept)      Mileage     Mileage2         Year     Mileage3
##  -8.283e+01   -2.039e-05    2.321e-10    4.628e-02   -8.697e-16
```

```
# Stepwise regression
step(none, scope=list(upper=Full), scale = MSE)
```

```
## Start:  AIC=321.79
```

```
## LogPrice ~ 1
##
##            Df Sum of Sq    RSS      Cp
## + Mileage   1   12.4164 18.849  68.918
## + Year      1   10.3972 20.869 110.366
## + Mileage2  1    7.1801 24.086 176.402
## + Mileage3  1    3.7444 27.521 246.924
## <none>                  31.266 321.786
##
## Step:  AIC=68.92
## LogPrice ~ Mileage
##
##            Df Sum of Sq    RSS      Cp
## + Mileage2  1    2.2614 16.588  24.499
## + Mileage3  1    1.8526 16.997  32.891
## + Year      1    1.5761 17.273  38.567
## <none>                  18.849  68.918
## - Mileage   1   12.4164 31.266 321.786
##
## Step:  AIC=24.5
## LogPrice ~ Mileage + Mileage2
##
##            Df Sum of Sq    RSS      Cp
## + Year      1    0.9356 15.652   7.293
## + Mileage3  1    0.2608 16.327  21.146
## <none>                  16.588  24.499
## - Mileage2  1    2.2614 18.849  68.918
## - Mileage   1    7.4977 24.086 176.402
##
## Step:  AIC=7.29
## LogPrice ~ Mileage + Mileage2 + Year
##
##            Df Sum of Sq    RSS     Cp
## + Mileage3  1    0.2091 15.443  5.000
## <none>                  15.652  7.293
## - Year      1    0.9356 16.588 24.499
## - Mileage2  1    1.6210 17.273 38.567
## - Mileage   1    3.9246 19.577 85.852
##
## Step:  AIC=5
## LogPrice ~ Mileage + Mileage2 + Year + Mileage3
##
##            Df Sum of Sq    RSS     Cp
## <none>                  15.443  5.000
## - Mileage3  1   0.20914 15.652  7.293
## - Mileage2  1   0.52038 15.964 13.682
## - Year      1   0.88401 16.327 21.146
## - Mileage   1   1.89675 17.340 41.934
##
## Call:
## lm(formula = LogPrice ~ Mileage + Mileage2 + Year + Mileage3,
##     data = MyVehicles)
##
## Coefficients:
## (Intercept)      Mileage     Mileage2         Year     Mileage3
##  -8.283e+01   -2.039e-05    2.321e-10    4.628e-02   -8.697e-16
```

When comparing the AIC of these models the best is Price ~ Mileage + Mileage2 + Year + Mileage3 which has the lowest AIC value of 5.

9. Assess and compare the overall effectiveness of the four models (some or all of them may be identical).

The two most effective models are the LogPrice ~ Mileage + Mileage2 + Year + Mileage3 and the LogPrice ~ Mileage + Mileage2 + Year models. They are very close with the first one having an AIC value of 5 and the second having an AIC value of 7.29. The next three models drop off heavily and are much less of a fit. LogPrice ~ Mileage + Mileage2 with an AIC of 24.5, LogPrice~Mileage with an AIC of 68.92, and LogPrice~1 with an AIC of 321.79. Overall, LogPrice ~ Mileage + Mileage2 + Year + Mileage3 is quite a good fit due to it's very low AIC value especially when compared to the other models.

10. Suppose that you are interested in purchasing a car of this model that is from the year 2017 with 50K miles. Determine each of the following: a 95% confidence interval for the mean price at this year and odometer reading, and a 95% prediction interval for the price of an individual car at this year and odometer reading. Write sentences that carefully interpret each of the intervals (in terms of car prices).

```
car = data.frame(Year = 2017, Mileage = 50000, Mileage2 = 50000^2, Mileage3 = 50000^3)
BestModel = lm(Price ~ Mileage + Mileage2 + Year + Mileage3, data = MyVehicles)

predict.lm(BestModel, car, interval = "confidence", level = 0.95)
```

```
##        fit      lwr      upr
## 1 22325.34 19925.96 24724.73
```

```
predict.lm(BestModel, car, interval = "prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 22325.34 9159.046 35491.64
```

We are 95% confident that the mean price of a 2017 Camaro Coupe with 50k miles is between 19,925.96 and 24,724.73
We are 95% confident that the next vehicle price will fall within 9,159.046 and 35,491.64