REGRESSION ANALYSIS FINAL PROJECT

Akshat Vijayvargia
S3826627

AKSHAT VIJAYVARGIA(S3826627)

INTRODUCTION

Does the pricing of secondhand vehicles only depend on wear-tear? I believe there are tons of parameters to figure that out. So, lets come and explore all those features.

The purpose of this investigation is to compare different models and predict the selling price of vehicles. Our focus is to gather more and more useful information and try to get close to the real-life pricing of these used vehicles.

DATASET EXPLORATION

This dataset is of an Indian company CARDEKHO (Dekho means look/ observe), this company is dedicated to all sorts of cars, which are ready to sell irrespective of how many times it is used.

I have picked this dataset from Kaggle Repository: Used Cars data form websites at

https://www.kaggle.com/nehalbirla/vehicle-dataset-from-cardekho.

This dataset has 4340 observations and 8 features of cars.

FEATURES OF THE DATASET

Regressor features are-

Name – Talks about the brand name and model number.

Year – Talks about the year in which the vehicle was manufactured.

Kms_driven – How many distance the car has travelled from the date of manufacture.

Fuel- Whether the vehicle runs on Petrol, Diesel, CNG or LPG

Transmission – Whether the car is Automatic or Manual.

Owner – Firsthand, Second hand, Third hand or Fourth or above

Age – Age is mutated feature, which tells about the age of vehicle from the date of manufacture to till date.

Target Feature is –

Selling_ Price- The selling price of a used vehicles.

METHODOLOGY

I briefly have breakdown the steps involved in this report to have essence-

Data Preprocessing – In this step, I have imported the dataset, dropped some irrelevant columns, and renamed all the selected columns.

AKSHAT VIJAYVARGIA(S3826627)

Methods – This is the most important section, where I used Manual Backward Selection, Manual Forward Selection, Automatic Stepwise Selection, Best Possible Subset Regression to fit the multiple regression model.

Model Prediction – In this section, I split the dataset in train- test in 80-20 percent and try to predict the values of our target feature Selling_Price.

OUTPUTS AND INTERPRETATION

```
A tibble: 6 x 8
##       X1      Y      X2 X3     X4         X5     X6              X7
##    <dbl>  <dbl>   <dbl> <chr>  <chr>      <chr>  <chr>        <dbl>
## 1   2007  60000   70000 Petrol Individual Manual First Owner     14
## 2   2007 135000   50000 Petrol Individual Manual First Owner     14
## 3   2012 600000 100000 Diesel Individual Manual First Owner      9
## 4   2017 250000   46000 Petrol Individual Manual First Owner      4
```

Renamed all the variables in X and Y coefficients. Xs are for regressors, and Y is for target feature.

METHODS

1) Manual Backward Selection

In this method we drop the features with the smallest F-value, so that we can know which features are significant enough to be included in the model.

```
MANUAL_BACK=lm(log(Y)~., data = CAR)
drop1(MANUAL_BACK,test="F")

## Single term deletions
##
## Model:
## log(Y) ~ X1 + X2 + X3 + X4 + X5 + X6 + X7
##        Df Sum of Sq     RSS     AIC   F value    Pr(>F)
## <none>               959.25 -6523.2
## X1      0     0.000  959.25 -6523.2
## X2      1     0.976  960.22 -6520.7    4.4022  0.035950 *
## X3      4   241.620 1200.87 -5556.2  272.4142 < 2.2e-16 ***
## X4      2    33.255  992.50 -6379.2   74.9868 < 2.2e-16 ***
## X5      1   241.245 1200.49 -5551.5 1087.9644 < 2.2e-16 ***
## X6      4     4.746  963.99 -6509.7    5.3512  0.000269 ***
## X7      0     0.000  959.25 -6523.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

drop1(update(MANUAL_BACK, ~ . -X1-X2), test = "F")
```

AKSHAT VIJAYVARGIA(S3826627)

```
## Single term deletions
##
## Model:
## log(Y) ~ X3 + X4 + X5 + X6 + X7
##        Df Sum of Sq     RSS     AIC  F value    Pr(>F)
## <none>              960.22 -6520.7
## X3      4    267.04 1227.26 -5463.8  300.833 < 2.2e-16 ***
## X4      2     34.53  994.75 -6371.4   77.796 < 2.2e-16 ***
## X5      1    244.12 1204.34 -5539.6 1100.048 < 2.2e-16 ***
## X6      4      5.37  965.59 -6504.5    6.050 7.495e-05 ***
## X7      1    748.57 1708.79 -4021.3 3373.254 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
LM1=lm(log(Y)~X3+X4+X5+X6+X7,data=CAR)
LM1$coefficients
```

```
##            (Intercept)               X3Diesel             X3Electric
##            14.20245662             0.57783328             0.20839773
##                  X3LPG               X3Petrol           X4Individual
##            -0.05237128             0.07948789            -0.16014593
##     X4Trustmark Dealer               X5Manual X6Fourth & Above Owner
##             0.31026257            -0.80572586            -0.14432763
##         X6Second Owner        X6Test Drive Car          X6Third Owner
##            -0.04955676             0.17478041            -0.12348231
##                     X7
##            -0.11446467
```

From the above results, it is clear that only feature X3,X4,X5,X6 and X7 features are significant. Now, have the look at the summary below-

```
summary(LM1)
```

```
##
## Call:
## lm(formula = log(Y) ~ X3 + X4 + X5 + X6 + X7, data = CAR)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7509 -0.3007  0.0020  0.2912  2.3760
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            14.202457   0.079151 179.436  < 2e-16 ***
## X3Diesel                0.577833   0.075322   7.671 2.09e-14 ***
## X3Electric              0.208398   0.478027   0.436  0.66289
## X3LPG                  -0.052371   0.123483  -0.424  0.67150
## X3Petrol                0.079488   0.075374   1.055  0.29168
## X4Individual           -0.160146   0.018152  -8.822  < 2e-16 ***
```

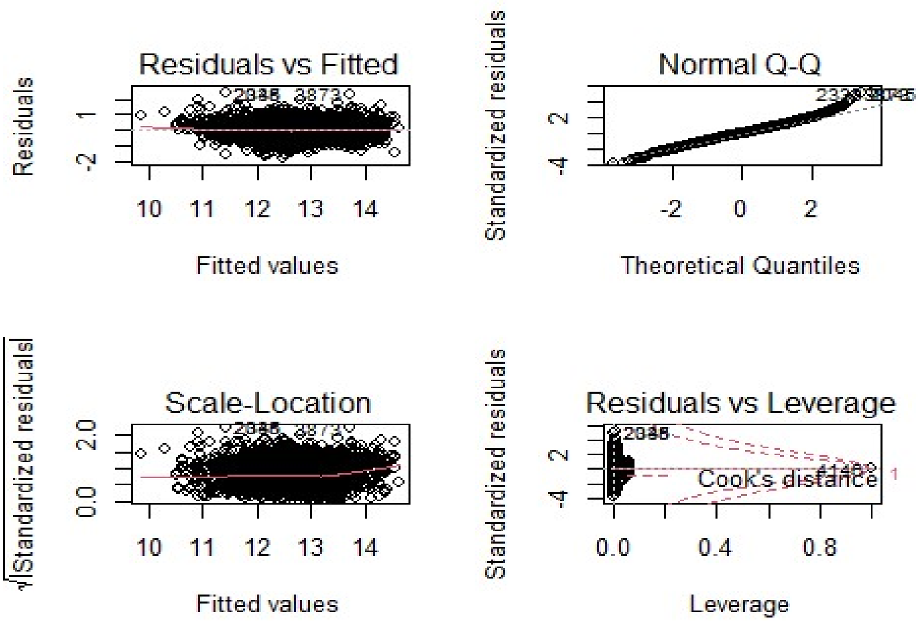
```
## X4Trustmark Dealer      0.310263   0.049153   6.312 3.03e-10 ***
## X5Manual               -0.805726   0.024293 -33.167  < 2e-16 ***
## X6Fourth & Above Owner -0.144328   0.055080  -2.620  0.00882 **
## X6Second Owner         -0.049557   0.018372  -2.697  0.00701 **
## X6Test Drive Car        0.174780   0.115819   1.509  0.13135
## X6Third Owner          -0.123482   0.030532  -4.044 5.34e-05 ***
## X7                     -0.114465   0.001971 -58.080  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4711 on 4327 degrees of freedom
## Multiple R-squared:  0.6858, Adjusted R-squared:  0.6849
## F-statistic: 787.1 on 12 and 4327 DF,  p-value: < 2.2e-16
```

We can say that all the predictors explain approx. 68% in the Selling Price of the vehicles.

The fitted equation is

Y_hat=14.202457+0.813348*X3+0.15011664*X4-0.80572586*X5-0.14258629*X6-0.11446467*X7

Now let us check the assumptions-




```
# EVALUATE HOMOSCEDASTICITY
# Non-constant error variance test
 # H0: Errors have a constant variance
 # H1: Errors have a non-constant variance
ncvTest(LM1)
```

AKSHAT VIJAYVARGIA(S3826627)

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.2338074, Df = 1, p = 0.62871
```

Since the p-value is greater than 0.05, we reject the Ho hypothesis. Hence, the constant vari ance assumption is not violated.

```
# TEST FOR NORMALLY DISTRIBUTED ERRORS
 # H0: Errors are normally distributed
 # H1: Errors are not normally distributed
shapiro.test(LM1$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  LM1$residuals
## W = 0.99419, p-value = 3.16e-12
```

Since the p-value is less than 0.05, we cannot reject the Ho hypothesis. Hence, the normality assumption is violated.

```
# TEST FOR AUTOCORRELATED ERRORS
 # H0: Errors are uncorrelated
 # H1: Errors are correlated
acf(LM1$residuals)
durbinWatsonTest(LM1)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.07152044      1.856248       0
##  Alternative hypothesis: rho != 0
```

Since the p-value is less than 0.05, we cannot reject the Ho hypothesis. Hence, the uncorrelat ed error assumption is violated.

```
# TEST FOR MULTICOLLINEARITY
vif(LM1)
```

```
##                X3Diesel              X3Electric                   X3LPG
##                 27.7370                  1.0295                  1.5720
##                X3Petrol            X4Individual     X4Trustmark Dealer
##                 27.7640                  1.2164                  1.0844
##                X5Manual X6Fourth & Above Owner         X6Second Owner
##                  1.0684                  1.0867                  1.2535
##        X6Test Drive Car          X6Third Owner                     X7
##                  1.0236                  1.1876                  1.3495
```

The VIF shows the presence of multicollinearity as X3 is more than 5. To corr ect this issue, we would have to remove it.

AKSHAT VIJAYVARGIA(S3826627)

2) Manual Forward Selection

In this method, we add the variable with the largest F-value. Just opposite of Manual Backward Selection.

```
null=lm(Y~1, data = CAR) # Start with null model with no variables
full=lm(Y~ ., data=CAR)
add1(null, scope =full, test = "F") # Manual F-test-based forward selection

## Single term additions
##
## Model:
## Y ~ 1
##        Df  Sum of Sq        RSS    AIC  F value    Pr(>F)
## <none>               1.4523e+15 115170
## X1      1 2.4883e+14 1.2035e+15 114356  896.902 < 2.2e-16 ***
## X2      1 5.3700e+13 1.3986e+15 115008  166.556 < 2.2e-16 ***
## X3      4 1.1716e+14 1.3352e+15 114813   95.094 < 2.2e-16 ***
## X4      2 8.4888e+13 1.3675e+15 114912  134.615 < 2.2e-16 ***
## X5      1 4.0828e+14 1.0441e+15 113739 1696.366 < 2.2e-16 ***
## X6      4 8.2710e+13 1.3696e+15 114923   65.446 < 2.2e-16 ***
## X7      1 2.4883e+14 1.2035e+15 114356  896.902 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

add1(update(null, ~ . +X5), scope = full, test = "F")

## Single term additions
##
## Model:
## Y ~ X5
##        Df  Sum of Sq        RSS    AIC F value    Pr(>F)
## <none>               1.0441e+15 113739
## X1      1 1.6910e+14 8.7496e+14 112974 838.204 < 2.2e-16 ***
## X2      1 2.4350e+13 1.0197e+15 113639 103.564 < 2.2e-16 ***
## X3      4 9.7541e+13 9.4652e+14 113322 111.656 < 2.2e-16 ***
## X4      2 2.7753e+13 1.0163e+15 113626  59.204 < 2.2e-16 ***
## X6      4 5.4647e+13 9.8942e+14 113514  59.843 < 2.2e-16 ***
## X7      1 1.6910e+14 8.7496e+14 112974 838.204 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

add1(update(null, ~ . +X5+X7), scope = full, test = "F")

## Single term additions
##
## Model:
## Y ~ X5 + X7
##        Df  Sum of Sq        RSS    AIC F value    Pr(>F)
## <none>               8.7496e+14 112974
## X1      0 0.0000e+00 8.7496e+14 112974
```

```
## X2      1 1.8446e+11 8.7478e+14 112975  0.9143   0.33902
## X3      4 7.0444e+13 8.0452e+14 112618 94.8498 < 2.2e-16 ***
## X4      2 1.0540e+13 8.6442e+14 112926 26.4285 3.905e-12 ***
## X6      4 2.5288e+12 8.7243e+14 112970  3.1399   0.01373 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
add1(update(null, ~ . +X5+X7+X3), scope = full, test = "F")
```

```
## Single term additions
##
## Model:
## Y ~ X5 + X7 + X3
##        Df  Sum of Sq        RSS    AIC F value    Pr(>F)
## <none>                8.0452e+14 112618
## X1      0 0.0000e+00 8.0452e+14 112618
## X2      1 8.6622e+12 7.9586e+14 112573  47.150 7.508e-12 ***
## X4      2 1.0806e+13 7.9371e+14 112563  29.483 1.915e-13 ***
## X6      4 4.6981e+12 7.9982e+14 112601   6.357 4.259e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
add1(update(null, ~ . +X5+X7+X3+X2), scope = full, test = "F")
```

```
## Single term additions
##
## Model:
## Y ~ X5 + X7 + X3 + X2
##        Df  Sum of Sq        RSS    AIC F value    Pr(>F)
## <none>                7.9586e+14 112573
## X1      0 0.0000e+00 7.9586e+14 112573
## X4      2 8.8283e+12 7.8703e+14 112529 24.2854 3.249e-11 ***
## X6      4 3.1543e+12 7.9270e+14 112564  4.3054  0.001774 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
add1(update(null, ~ . +X5+X7+X3+X2+X4), scope = full, test = "F")
```

```
## Single term additions
##
## Model:
## Y ~ X5 + X7 + X3 + X2 + X4
##        Df  Sum of Sq        RSS    AIC F value Pr(>F)
## <none>                7.8703e+14 112529
## X1      0 0.0000e+00 7.8703e+14 112529
## X6      4 1.6826e+12 7.8535e+14 112527  2.3171 0.0549 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
add1(update(null, ~ . +X5+X7+X3+X2+X4+X6), scope = full, test = "F")
```

AKSHAT VIJAYVARGIA(S3826627)

```
## Single term additions
##
## Model:
## Y ~ X5 + X7 + X3 + X2 + X4 + X6
##        Df Sum of Sq        RSS    AIC F value Pr(>F)
## <none>               7.8535e+14 112527
## X1      0         0 7.8535e+14 112527
```

```
LM2=lm(log(Y)~X5+X7+X3+X2+X4+X6,data=CAR)
```

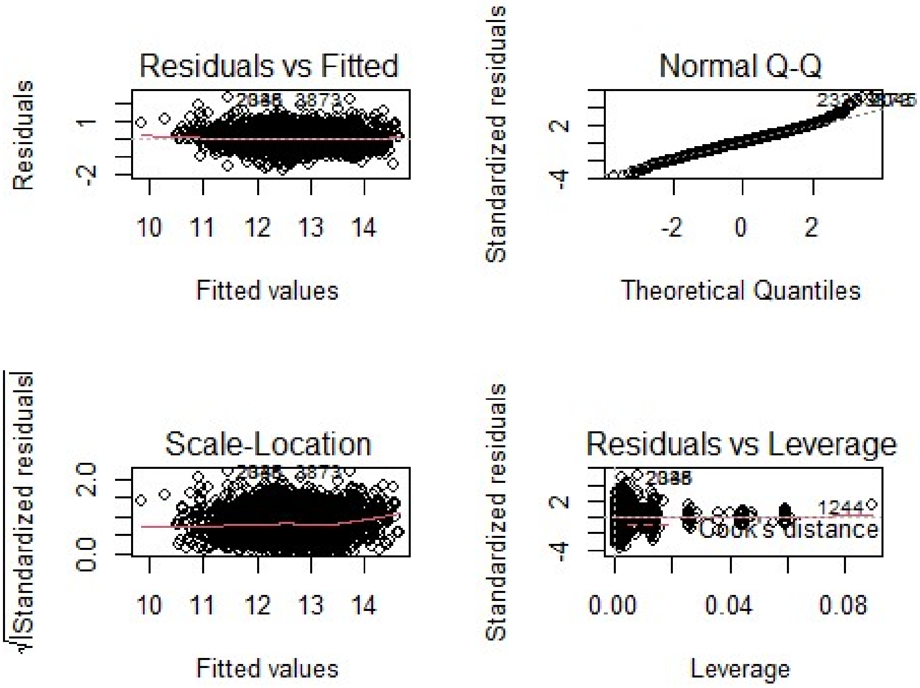
```
summary(LM2)
```

```
##
## Call:
## lm(formula = log(Y) ~ X5 + X7 + X3 + X2 + X4 + X6, data = CAR)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.76930 -0.30211 -0.00082  0.28965  2.34532
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.421e+01  7.918e-02 179.456  < 2e-16 ***
## X5Manual               -8.025e-01  2.433e-02 -32.984  < 2e-16 ***
## X7                     -1.128e-01  2.120e-03 -53.216  < 2e-16 ***
## X3Diesel                5.839e-01  7.535e-02   7.749 1.14e-14 ***
## X3Electric              1.983e-01  4.779e-01   0.415 0.678172
## X3LPG                  -5.011e-02  1.234e-01  -0.406 0.684779
## X3Petrol                7.311e-02  7.541e-02   0.970 0.332323
## X2                     -3.902e-07  1.860e-07  -2.098 0.035950 *
## X4Individual           -1.569e-01  1.821e-02  -8.616  < 2e-16 ***
## X4Trustmark Dealer      3.085e-01  4.914e-02   6.278 3.76e-10 ***
## X6Fourth & Above Owner -1.398e-01  5.510e-02  -2.537 0.011227 *
## X6Second Owner         -4.624e-02  1.843e-02  -2.509 0.012154 *
## X6Test Drive Car        1.659e-01  1.159e-01   1.432 0.152147
## X6Third Owner          -1.164e-01  3.071e-02  -3.792 0.000152 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4709 on 4326 degrees of freedom
## Multiple R-squared:  0.6861, Adjusted R-squared:  0.6852
## F-statistic: 727.5 on 13 and 4326 DF,  p-value: < 2.2e-16
```

We can say that all the predictors explain approx. 68% in the Selling Price of the vehicles just as the Manual Backward Selection.

Now lets check the assumptions of LM2-

AKSHAT VIJAYVARGIA(S3826627)




```
# EVALUATE HOMOSCEDASTICITY
# Non-constant error variance test
 # H0: Errors have a constant variance
 # H1: Errors have a non-constant variance
ncvTest(LM2)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.2157866, Df = 1, p = 0.64227
```

Since the p-value is greater than 0.05, we reject the Ho hypothesis. Hence, the constant vari ance assumption is not violated.

```
# TEST FOR NORMALLY DISTRIBUTED ERRORS
 # H0: Errors are normally distributed
 # H1: Errors are not normally distributed
shapiro.test(LM2$residuals)

##
##  Shapiro-Wilk normality test
##
## data:  LM2$residuals
## W = 0.99425, p-value = 3.881e-12
```

Since the p-value is less than 0.05, we cannot reject the Ho hypothesis. Hence, the normality assumption is violated.

```
# TEST FOR AUTOCORRELATED ERRORS
 # H0: Errors are uncorrelated
 # H1: Errors are correlated
acf(LM2$residuals)
durbinWatsonTest(LM2)

##  lag Autocorrelation D-W Statistic p-value
##    1      0.07120179      1.856883       0
##  Alternative hypothesis: rho != 0
```

Since the p-value is less than 0.05, we cannot reject the Ho hypothesis. Hence, the uncorrelated error assumption is violated.

```
# TEST FOR MULTICOLLINEARITY
vif(LM2)

##               X5Manual                     X7               X3Diesel
##                 1.0726                 1.5628                27.7780
##             X3Electric                  X3LPG               X3Petrol
##                 1.0296                 1.5721                27.8100
##                     X2           X4Individual     X4Trustmark Dealer
##                 1.4728                 1.2252                 1.0847
## X6Fourth & Above Owner        X6Second Owner        X6Test Drive Car
##                 1.0884                 1.2628                 1.0249
##         X6Third Owner
##                 1.2020
```

The VIF shows the presence of multicollinearity as X3 is more than 5. To correct this issue, we would have to remove it. Multicollinearity results are like that of Backward Selection.

3) Automatic Stepwise Selection

3.1) Automatic Forward Selection

```
step(null, scope=list(lower=null, upper=full), direction="forward") #SUGGESTS MODEL2 LM2

## Start:  AIC=115169.6
## Y ~ 1
##
##        Df  Sum of Sq        RSS    AIC
## + X5    1 4.0828e+14 1.0441e+15 113739
## + X7    1 2.4883e+14 1.2035e+15 114356
## + X1    1 2.4883e+14 1.2035e+15 114356
## + X3    4 1.1716e+14 1.3352e+15 114813
## + X4    2 8.4888e+13 1.3675e+15 114912
## + X6    4 8.2710e+13 1.3696e+15 114923
## + X2    1 5.3700e+13 1.3986e+15 115008
```

```
## <none>               1.4523e+15 115170
##
## Step:  AIC=113739.2
## Y ~ X5
##
##        Df  Sum of Sq        RSS    AIC
## + X7    1 1.6910e+14 8.7496e+14 112974
## + X1    1 1.6910e+14 8.7496e+14 112974
## + X3    4 9.7541e+13 9.4652e+14 113322
## + X6    4 5.4647e+13 9.8942e+14 113514
## + X4    2 2.7753e+13 1.0163e+15 113626
## + X2    1 2.4350e+13 1.0197e+15 113639
## <none>               1.0441e+15 113739
##
## Step:  AIC=112974.3
## Y ~ X5 + X7
##
##        Df  Sum of Sq        RSS    AIC
## + X3    4 7.0444e+13 8.0452e+14 112618
## + X4    2 1.0540e+13 8.6442e+14 112926
## + X6    4 2.5288e+12 8.7243e+14 112970
## <none>               8.7496e+14 112974
## + X2    1 1.8446e+11 8.7478e+14 112975
##
## Step:  AIC=112618.1
## Y ~ X5 + X7 + X3
##
##        Df  Sum of Sq        RSS    AIC
## + X4    2 1.0806e+13 7.9371e+14 112563
## + X2    1 8.6622e+12 7.9586e+14 112573
## + X6    4 4.6981e+12 7.9982e+14 112601
## <none>               8.0452e+14 112618
##
## Step:  AIC=112563.4
## Y ~ X5 + X7 + X3 + X4
##
##        Df  Sum of Sq        RSS    AIC
## + X2    1 6.6844e+12 7.8703e+14 112529
## + X6    4 2.4705e+12 7.9124e+14 112558
## <none>               7.9371e+14 112563
##
## Step:  AIC=112528.7
## Y ~ X5 + X7 + X3 + X4 + X2
##
##        Df  Sum of Sq        RSS    AIC
## + X6    4 1.6826e+12 7.8535e+14 112527
## <none>               7.8703e+14 112529
##
## Step:  AIC=112527.4
## Y ~ X5 + X7 + X3 + X4 + X2 + X6
```

```
##
##        Df Sum of Sq        RSS    AIC
## <none>              7.8535e+14 112527

##
## Call:
## lm(formula = Y ~ X5 + X7 + X3 + X4 + X2 + X6, data = CAR)
##
## Coefficients:
##           (Intercept)              X5Manual                    X7
##             1.545e+06            -8.703e+05            -3.526e+04
##              X3Diesel            X3Electric                 X3LPG
##             2.863e+05            -6.059e+05             4.700e+04
##              X3Petrol          X4Individual     X4Trustmark Dealer
##            -4.245e+03            -6.638e+04             1.675e+05
##                    X2  X6Fourth & Above Owner        X6Second Owner
##            -9.591e-01            -1.454e+03            -4.093e+04
##       X6Test Drive Car          X6Third Owner
##             1.687e+05            -3.993e+04
```

Automatic Forward suggests X2, X3,X4,X5,X6,X7 – like Model 2.

3.2) Automatic Backward Selection

```
# AUTOMATIC BACKWARD SELECTION
step(full, data=CAR, direction="backward")

## Start:  AIC=112527.4
## Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7
##
##
## Step:  AIC=112527.4
## Y ~ X1 + X2 + X3 + X4 + X5 + X6
##
##        Df  Sum of Sq        RSS    AIC
## <none>               7.8535e+14 112527
## - X6    4 1.6826e+12 7.8703e+14 112529
## - X2    1 5.8965e+12 7.9124e+14 112558
## - X4    2 7.3566e+12 7.9270e+14 112564
## - X1    1 6.1324e+13 8.4667e+14 112852
## - X3    4 7.7256e+13 8.6260e+14 112927
## - X5    1 2.8373e+14 1.0691e+15 113864

##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6, data = CAR)
##
## Coefficients:
##           (Intercept)                    X1                    X2
##            -6.971e+07             3.526e+04            -9.591e-01
##              X3Diesel            X3Electric                 X3LPG
```

```
##                  2.863e+05                 -6.059e+05                  4.700e+04
##                   X3Petrol               X4Individual       X4Trustmark Dealer
##                 -4.245e+03                 -6.638e+04                  1.675e+05
##                   X5Manual    X6Fourth & Above Owner           X6Second Owner
##                 -8.703e+05                 -1.454e+03                 -4.093e+04
##          X6Test Drive Car               X6Third Owner
##                  1.687e+05                 -3.993e+04
```

Automatic Backward uses all the variable except X7.

3.3) Automatic Stepwise Selection (Both)

```
# AUTOMATIC STEPWISE
step(null, scope = list(upper=full), data=CAR, direction="both") #SUGGESTS MO
DEL2 LM2

## Start:  AIC=115169.6
## Y ~ 1
##
##          Df  Sum of Sq        RSS    AIC
## + X5      1 4.0828e+14 1.0441e+15 113739
## + X7      1 2.4883e+14 1.2035e+15 114356
## + X1      1 2.4883e+14 1.2035e+15 114356
## + X3      4 1.1716e+14 1.3352e+15 114813
## + X4      2 8.4888e+13 1.3675e+15 114912
## + X6      4 8.2710e+13 1.3696e+15 114923
## + X2      1 5.3700e+13 1.3986e+15 115008
## <none>                 1.4523e+15 115170
##
## Step:  AIC=113739.2
## Y ~ X5
##
##          Df  Sum of Sq        RSS    AIC
## + X7      1 1.6910e+14 8.7496e+14 112974
## + X1      1 1.6910e+14 8.7496e+14 112974
## + X3      4 9.7541e+13 9.4652e+14 113322
## + X6      4 5.4647e+13 9.8942e+14 113514
## + X4      2 2.7753e+13 1.0163e+15 113626
## + X2      1 2.4350e+13 1.0197e+15 113639
## <none>                 1.0441e+15 113739
## - X5      1 4.0828e+14 1.4523e+15 115170
##
## Step:  AIC=112974.3
## Y ~ X5 + X7
##
##          Df  Sum of Sq        RSS    AIC
## + X3      4 7.0444e+13 8.0452e+14 112618
## + X4      2 1.0540e+13 8.6442e+14 112926
## + X6      4 2.5288e+12 8.7243e+14 112970
## <none>                 8.7496e+14 112974
```

```
## + X2    1 1.8446e+11 8.7478e+14 112975
## - X7    1 1.6910e+14 1.0441e+15 113739
## - X5    1 3.2855e+14 1.2035e+15 114356
##
## Step:  AIC=112618.1
## Y ~ X5 + X7 + X3
##
##         Df  Sum of Sq        RSS    AIC
## + X4    2 1.0806e+13 7.9371e+14 112563
## + X2    1 8.6622e+12 7.9586e+14 112573
## + X6    4 4.6981e+12 7.9982e+14 112601
## <none>              8.0452e+14 112618
## - X3    4 7.0444e+13 8.7496e+14 112974
## - X7    1 1.4201e+14 9.4652e+14 113322
## - X5    1 3.1853e+14 1.1230e+15 114064
##
## Step:  AIC=112563.4
## Y ~ X5 + X7 + X3 + X4
##
##         Df  Sum of Sq        RSS    AIC
## + X2    1 6.6844e+12 7.8703e+14 112529
## + X6    4 2.4705e+12 7.9124e+14 112558
## <none>              7.9371e+14 112563
## - X4    2 1.0806e+13 8.0452e+14 112618
## - X3    4 7.0710e+13 8.6442e+14 112926
## - X7    1 1.2663e+14 9.2034e+14 113204
## - X5    1 2.8887e+14 1.0826e+15 113908
##
## Step:  AIC=112528.7
## Y ~ X5 + X7 + X3 + X4 + X2
##
##         Df  Sum of Sq        RSS    AIC
## + X6    4 1.6826e+12 7.8535e+14 112527
## <none>              7.8703e+14 112529
## - X2    1 6.6844e+12 7.9371e+14 112563
## - X4    2 8.8283e+12 7.9586e+14 112573
## - X3    4 7.6762e+13 8.6379e+14 112925
## - X7    1 7.8675e+13 8.6570e+14 112940
## - X5    1 2.8286e+14 1.0699e+15 113859
##
## Step:  AIC=112527.4
## Y ~ X5 + X7 + X3 + X4 + X2 + X6
##
##         Df  Sum of Sq        RSS    AIC
## <none>              7.8535e+14 112527
## - X6    4 1.6826e+12 7.8703e+14 112529
## - X2    1 5.8965e+12 7.9124e+14 112558
## - X4    2 7.3566e+12 7.9270e+14 112564
## - X7    1 6.1324e+13 8.4667e+14 112852
```

```
## - X3    4 7.7256e+13 8.6260e+14 112927
## - X5    1 2.8373e+14 1.0691e+15 113864

##
## Call:
## lm(formula = Y ~ X5 + X7 + X3 + X4 + X2 + X6, data = CAR)
##
## Coefficients:
##           (Intercept)               X5Manual                     X7
##             1.545e+06             -8.703e+05             -3.526e+04
##              X3Diesel             X3Electric                  X3LPG
##             2.863e+05             -6.059e+05              4.700e+04
##              X3Petrol           X4Individual     X4Trustmark Dealer
##            -4.245e+03             -6.638e+04              1.675e+05
##                    X2  X6Fourth & Above Owner         X6Second Owner
##            -9.591e-01             -1.454e+03             -4.093e+04
##       X6Test Drive Car          X6Third Owner
##             1.687e+05             -3.993e+04
```

The above output uses X2, X3,X4,X5,X6 and X7 for linear model. Moreover, this has the same result to that of Automatic Forward Selection and Model 2.

4) Best Possible Subsets

```
Model=lm(log(Y)~.,data=CAR)

suppressWarnings({ MODELCOMPARE<-ols_step_best_subset(Model)
MODELCOMPARE })

##        Best Subsets Regression
## ----------------------------------
## Model Index    Predictors
## ----------------------------------
##      1         X7
##      2         X5 X7
##      3         X3 X5 X7
##      4         X3 X4 X5 X7
##      5         X3 X4 X5 X6 X7
##      6         X2 X3 X4 X5 X6 X7
##      7         X1 X2 X3 X4 X5 X6 X7
## ----------------------------------
##
##                                                      Subsets Regression Sum
mary
## ----------------------------------------------------------------------------
-------------------------------------------------------
##                         Adj.        Pred
## Model     R-Square             C(p)          AIC        SBIC       SBC
MSEP        FPE        HSP       APC
## ----------------------------------------------------------------------------
```

```
-----------------------------------------------------------
##   1        0.4840      0.4839      0.4835    2775.6320    7928.5973      N
A    7947.7242    1577.6580    0.3637    1e-04    0.5164
##   2        0.5844      0.5842      0.5835    1394.6880    6992.0931      N
A    7017.5956    1271.1562    0.2931    1e-04    0.4162
##   3        0.6705      0.6700      0.6695     209.8209    5992.5250      N
A    6043.5301    1008.0310    0.2326    1e-04    0.3301
##   4        0.6841      0.6835      0.6829      24.6210    5813.8526      N
A    5877.6089     966.7061    0.2232    1e-04    0.3167
##   5        0.6858      0.6849      0.6841       2.4022    5797.6477      N
A    5886.9066     961.5515    0.2222    1e-04    0.3151
##   6        0.6861      0.6852        -Inf       0.0000    5795.2335      N
A    5890.8680     960.7958    0.2221    1e-04    0.3149
##   7        0.6861      0.6852        -Inf       0.0000    5797.2335      N
A    5899.2436     960.7958    0.2221    1e-04    0.3149
## ------------------------------------------------------------------------------
-----------------------------------------------------------
## AIC: Akaike Information Criteria
##  SBIC: Sawa's Bayesian Information Criteria
##  SBC: Schwarz Bayesian Criteria
##  MSEP: Estimated error of prediction, assuming multivariate normality
##  FPE: Final Prediction Error
##  HSP: Hocking's Sp
##  APC: Amemiya Prediction Criteria
```

The best model is Model6 as it has the highest Adj R square and lowest AIC value. Again Model 6 uses the same variable as Model 2, so Model 2 is still the best.

5) Full Model Fitting

This is the last model, where we fit the model with all the 7 predictors.

```
##
## Call:
## lm(formula = log(Y) ~ ., data = CAR)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.76930 -0.30211 -0.00082  0.28965  2.34532
##
## Coefficients: (1 not defined because of singularities)
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -2.138e+02  4.279e+00 -49.968  < 2e-16 ***
## X1                      1.128e-01  2.120e-03  53.216  < 2e-16 ***
## X2                     -3.902e-07  1.860e-07  -2.098 0.035950 *
## X3Diesel                5.839e-01  7.535e-02   7.749 1.14e-14 ***
## X3Electric              1.983e-01  4.779e-01   0.415 0.678172
## X3LPG                  -5.011e-02  1.234e-01  -0.406 0.684779
## X3Petrol                7.311e-02  7.541e-02   0.970 0.332323
## X4Individual           -1.569e-01  1.821e-02  -8.616  < 2e-16 ***
```

```
## X4Trustmark Dealer      3.085e-01  4.914e-02   6.278 3.76e-10 ***
## X5Manual               -8.025e-01  2.433e-02 -32.984  < 2e-16 ***
## X6Fourth & Above Owner -1.398e-01  5.510e-02  -2.537 0.011227 *
## X6Second Owner         -4.624e-02  1.843e-02  -2.509 0.012154 *
## X6Test Drive Car        1.659e-01  1.159e-01   1.432 0.152147
## X6Third Owner          -1.164e-01  3.071e-02  -3.792 0.000152 ***
## X7                             NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4709 on 4326 degrees of freedom
## Multiple R-squared:  0.6861, Adjusted R-squared:  0.6852
## F-statistic: 727.5 on 13 and 4326 DF,  p-value: < 2.2e-16
```

We can say that all the predictors explain approx. 68% in the Selling Price. The best fitted equation for Model 3 is –

DATA TRANSFORMATION

In this section, I will transform all data with capping function and will also remove features, with more than 5 VIF.

```
capped<- function(x){
 quantiles <- quantile( x, c(.05, 0.25, 0.75, .95 ) )
 x[ x < quantiles[2] - 1.5*IQR(x) ] <- quantiles[1]
 x[ x > quantiles[3] + 1.5*IQR(x) ] <- quantiles[4]
 x
}
Y_TRANSFORMED <- capped(CAR$Y)
```

Data Transformation on Model1

```
##
## Call:
## lm(formula = log(Y_TRANSFORMED) ~ X4 + X5 + X6 + X7, data = CAR)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.02211 -0.33459  0.01063  0.32634  2.40762
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            14.369290   0.027988 513.418  < 2e-16 ***
## X4Individual           -0.172569   0.019322  -8.931  < 2e-16 ***
## X4Trustmark Dealer      0.245100   0.052353   4.682 2.93e-06 ***
## X5Manual               -0.605555   0.025839 -23.435  < 2e-16 ***
## X6Fourth & Above Owner -0.122800   0.058615  -2.095   0.0362 *
```

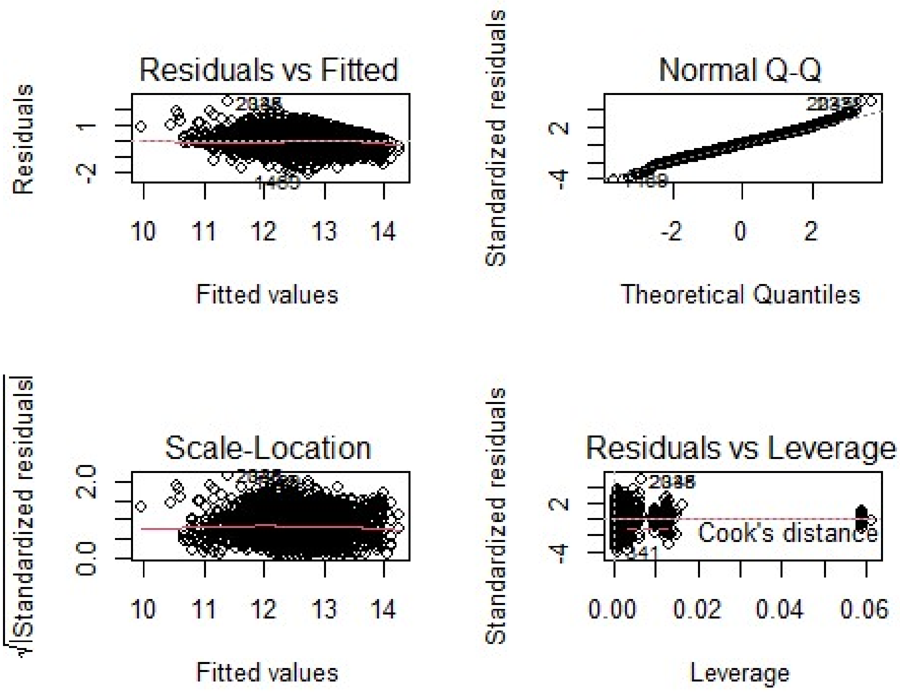
AKSHAT VIJAYVARGIA(S3826627)

```
## X6Second Owner          -0.008432   0.019523  -0.432   0.6658
## X6Test Drive Car         0.108560   0.123373   0.880   0.3789
## X6Third Owner           -0.056899   0.032479  -1.752   0.0799 .
## X7                      -0.120887   0.002078 -58.186  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.502 on 4331 degrees of freedom
## Multiple R-squared:  0.5907, Adjusted R-squared:  0.5899
## F-statistic: 781.2 on 8 and 4331 DF,  p-value: < 2.2e-16
```

We can see that all predictors explain approx. 59% of the Selling Price, which was earlier 68%. Hence, it seems that transformation did not improve the Model1.

Lets look at the assumptions for better understanding-

```
par(mfrow=c(2,2))
plot(NOM_MOD1)
```




```
# EVALUATE HOMOSCEDASTICITY
# Non-constant error variance test
 # H0: Errors have a constant variance
 # H1: Errors have a non-constant variance
ncvTest(NOM_MOD1)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 31.17189, Df = 1, p = 2.3616e-08
```

AKSHAT VIJAYVARGIA(S3826627)

Since the p-value is less than 0.05, we cannot reject the Ho hypothesis. Hence, the covariance error assumption is violated.

```
# TEST FOR NORMALLY DISTRIBUTED ERRORS
 # H0: Errors are normally distributed
 # H1: Errors are not normally distributed
shapiro.test(NOM_MOD1$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  NOM_MOD1$residuals
## W = 0.99681, p-value = 5.699e-08
```

Since the p-value is less than 0.05, we cannot reject the Ho hypothesis. Hence, the normality assumption is violated.

```
# TEST FOR AUTOCORRELATED ERRORS
 # H0: Errors are uncorrelated
 # H1: Errors are correlated
acf(NOM_MOD1$residuals)
durbinWatsonTest(NOM_MOD1)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.0709005       1.85706       0
##  Alternative hypothesis: rho != 0
```

Since the p-value is less than 0.05, we cannot reject the Ho hypothesis. Hence, the uncorrelated error assumption is violated.

```
# TEST FOR MULTICOLLINEARITY
vif(NOM_MOD1)
```

```
##             X4Individual      X4Trustmark Dealer               X5Manual
##                   1.2137                  1.0833                 1.0644
## X6Fourth & Above Owner         X6Second Owner       X6Test Drive Car
##                   1.0837                  1.2464                 1.0228
##            X6Third Owner                      X7
##                   1.1834                  1.3206
```

Multicollinearity is absent as we have already removed the X3 feature because of unacceptable VIF.

Hence, we can now say that performance of Model 1 is poor after the data transformation.

AKSHAT VIJAYVARGIA(S3826627)

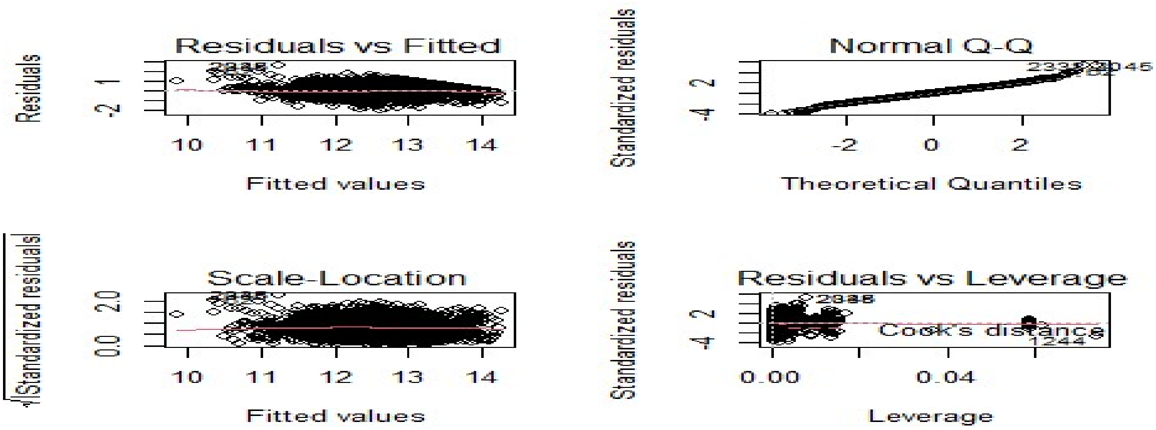
Data Transformation on Model 2

```
##
## Call:
## lm(formula = log(Y_TRANSFORMED) ~ X5 + X7 + X2 + X4 + X6, data = CAR)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.89897 -0.33014  0.01756  0.31757  2.57427
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.433e+01  2.792e-02 513.181  < 2e-16 ***
## X5Manual              -6.189e-01  2.556e-02 -24.217  < 2e-16 ***
## X7                    -1.277e-01  2.155e-03 -59.271  < 2e-16 ***
## X2                     1.890e-06  1.814e-07  10.420  < 2e-16 ***
## X4Individual          -1.860e-01  1.913e-02  -9.721  < 2e-16 ***
## X4Trustmark Dealer     2.590e-01  5.173e-02   5.006 5.77e-07 ***
## X6Fourth & Above Owner -1.500e-01  5.796e-02  -2.589  0.00966 **
## X6Second Owner        -2.932e-02  1.939e-02  -1.512  0.13054
## X6Test Drive Car       1.641e-01  1.220e-01   1.345  0.17869
## X6Third Owner         -9.846e-02  3.233e-02  -3.045  0.00234 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4959 on 4330 degrees of freedom
## Multiple R-squared:  0.6007, Adjusted R-squared:  0.5998
## F-statistic: 723.7 on 9 and 4330 DF,  p-value: < 2.2e-16
```

From the above output, we can see all predictors explain approx. 60% of the Selling Price, which was earlier 68%. Hence, it seems that transformation did not improve the Model2.

Checking the assumptions-

```
par(mfrow=c(2,2))
plot(NOM_MOD2)
```



AKSHAT VIJAYVARGIA(S3826627)

```
# EVALUATE HOMOSCEDASTICITY
# Non-constant error variance test
 # H0: Errors have a constant variance
 # H1: Errors have a non-constant variance
ncvTest(NOM_MOD2)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 36.96576, Df = 1, p = 1.2022e-09
```

Since the p-value is less than 0.05, we cannot reject the Ho hypothesis. Hence, the covariance error assumption is violated.

```
# TEST FOR NORMALLY DISTRIBUTED ERRORS
 # H0: Errors are normally distributed
 # H1: Errors are not normally distributed
shapiro.test(NOM_MOD2$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  NOM_MOD2$residuals
## W = 0.99654, p-value = 1.738e-08
```

Since the p-value is less than 0.05, we cannot reject the Ho hypothesis. Hence, the normality assumption is violated.

```
# TEST FOR AUTOCORRELATED ERRORS
 # H0: Errors are uncorrelated
 # H1: Errors are correlated
acf(NOM_MOD2$residuals)
durbinWatsonTest(NOM_MOD2)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.06601562      1.866885       0
##  Alternative hypothesis: rho != 0
```

Since the p-value is less than 0.05, we cannot reject the Ho hypothesis. Hence, the uncorrelated error assumption is violated.

```
# TEST FOR MULTICOLLINEARITY
vif(NOM_MOD2)
```

```
##              X5Manual                    X7                    X2
##                1.0671                1.4566                1.2632
##          X4Individual     X4Trustmark Dealer X6Fourth & Above Owner
##                1.2192                1.0840                1.0859
##         X6Second Owner        X6Test Drive Car        X6Third Owner
##                1.2599                1.0247                1.2017
```

AKSHAT VIJAYVARGIA(S3826627)

```
Multicollinearity is absent as we have already removed the X3 feature
because of unacceptable VIF.
```

Hence, we can now say that performance of Model 2 is also poor after the data transformation.

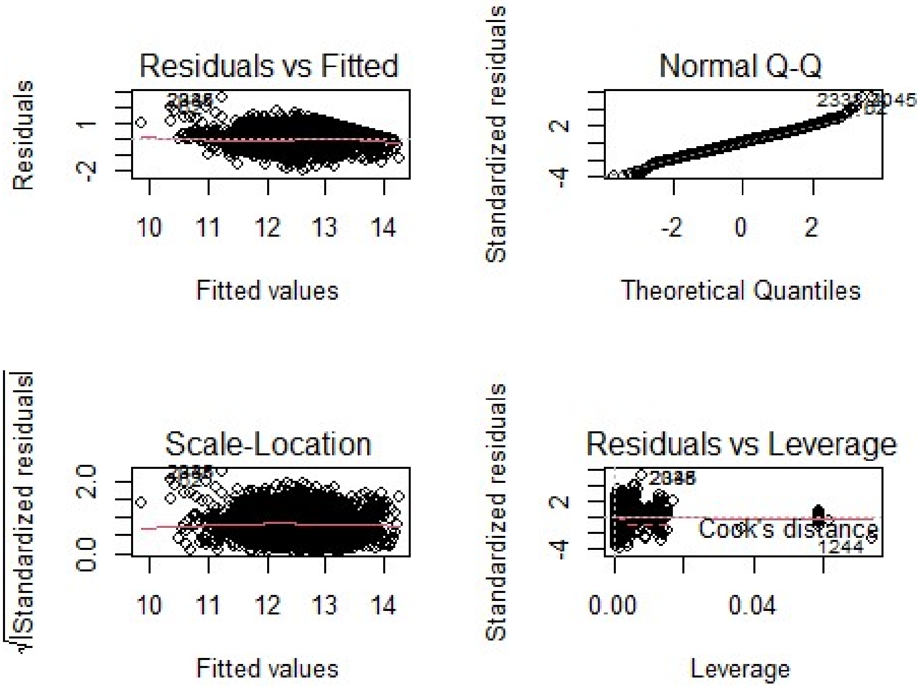
Data Transformation on Model 3

```
##
## Call:
## lm(formula = log(Y_TRANSFORMED) ~ X1 + X2 + X4 + X5 + X6 + X7,
##     data = CAR)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.89897 -0.33014  0.01756  0.31757  2.57427
##
## Coefficients: (1 not defined because of singularities)
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -2.439e+02  4.347e+00 -56.100  < 2e-16 ***
## X1                      1.277e-01  2.155e-03  59.271  < 2e-16 ***
## X2                      1.890e-06  1.814e-07  10.420  < 2e-16 ***
## X4Individual           -1.860e-01  1.913e-02  -9.721  < 2e-16 ***
## X4Trustmark Dealer      2.590e-01  5.173e-02   5.006 5.77e-07 ***
## X5Manual               -6.189e-01  2.556e-02 -24.217  < 2e-16 ***
## X6Fourth & Above Owner -1.500e-01  5.796e-02  -2.589  0.00966 **
## X6Second Owner         -2.932e-02  1.939e-02  -1.512  0.13054
## X6Test Drive Car        1.641e-01  1.220e-01   1.345  0.17869
## X6Third Owner          -9.846e-02  3.233e-02  -3.045  0.00234 **
## X7                             NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4959 on 4330 degrees of freedom
## Multiple R-squared:  0.6007, Adjusted R-squared:  0.5998
## F-statistic: 723.7 on 9 and 4330 DF,  p-value: < 2.2e-16
```

From the above output, we can see all predictors explain approx. 60% of the Selling Price, which was earlier 68%. Hence, it seems that transformation did not improve the Model3 as well.

Let us check the assumptions-

```
par(mfrow=c(2,2))
plot(NOM_MOD3)
```




```
# EVALUATE HOMOSCEDASTICITY
# Non-constant error variance test
 # H0: Errors have a constant variance
 # H1: Errors have a non-constant variance
ncvTest(NOM_MOD3)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 36.96576, Df = 1, p = 1.2022e-09
```

Since the p-value is less than 0.05, we cannot reject the Ho hypothesis. Hence, the the covariance error assumption is violated.

```
# TEST FOR NORMALLY DISTRIBUTED ERRORS
 # H0: Errors are normally distributed
 # H1: Errors are not normally distributed
shapiro.test(NOM_MOD3$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  NOM_MOD3$residuals
## W = 0.99654, p-value = 1.738e-08
```

Since the p-value is less than 0.05, we cannot reject the Ho hypothesis. Hence, the nor mality assumption is violated.

```
# TEST FOR AUTOCORRELATED ERRORS
 # H0: Errors are uncorrelated
 # H1: Errors are correlated
acf(NOM_MOD3$residuals)
durbinWatsonTest(NOM_MOD3)

##  lag Autocorrelation D-W Statistic p-value
##    1      0.06601562      1.866885       0
##  Alternative hypothesis: rho != 0
```

Since the p-value is less than 0.05, we cannot reject the Ho hypothesis. Hence, the uncorrelated error assumption is violated.

```
# TEST FOR MULTICOLLINEARITY
vif(NOM_MOD3)

##                     X1                      X2            X4Individual
##                 1.4566                  1.2632                  1.2192
##      X4Trustmark Dealer                X5Manual X6Fourth & Above Owner
##                 1.0840                  1.0671                  1.0859
##          X6Second Owner       X6Test Drive Car          X6Third Owner
##                 1.2599                  1.0247                  1.2017
```

Multicollinearity is absent as we have already removed the X3 feature because of unacceptable VIF.

Hence, we can now say that performance of Model 3 as poor as Model 1 and Model 2 after the data transformation. In all the three models, we noticed that all the assumptions are violated.

To conclude, our Model 1, Model 2 and Model 3 have two violated assumptions, but Model 2 is the best because it has the highest F value p-value.

MODEL PREDICTIONS

```
DATA<-data.frame(X1,X2,X3,X4,X5,X6,X7)
suppressWarnings({predict.lm(TRAINING_MODEL,DATA)})

##         1         2         3         4
## 1353634.0  660806.6  698192.7 1617890.0
```

Y	X1	X2	X3	X4	X5	X6	X7
₹1,353,634.00	2017	10000	CNG	Dealer	Automatic	Second Owner	4
₹660,806.60	2002	150000	Petrol	Individual	Automatic	First Owner	19
₹698,192.70	1998	45000	LPG	Dealer	Automatic	Third Owner	23
₹1,617,890.00	2010	80000	Diesel	Individual	Automatic	First Owner	2

AKSHAT VIJAYVARGIA(S3826627)

I created 4 random observations of independent variables, and column Red shows the predictions for these observations.

For prediction, I used createDataPartition() function to split data into train-test set and then used predict.lm() function to predict the values of target feature (Selling Price in INR).

FINDINGS OF THE REPORT

While data preprocessing section, we noticed that Name feature has n number of unique variables, which will only make the results useless. So, we decided to remove that variable.

In the Method section, we firstly, created Manual Backward Selection to have significant features for our best model. We started with the full model, and then eventually removing least significant features with lowest F-values. After all the removing steps, we ended up with X3,X4,X5,X6, and X7. And then used lm() function to have their model. Overall the **Model 1** is not fit because only the covariance error was met.

Secondly, we then created Manual Forward Selection to have our significant features for the model. We started off with the null set, and one by one, we added the best features with the highest F-value. In the end, we observe that all the variables were significant except X1, which was the Year column. After having the summary of X2,X3,X4,X5,X6 and X7, we noticed that all predictors explained only 68% of the model. While checking the assumptions, we came to know that our **Model 2** is not fit as it also violated 2 assumptions.

Thirdly, we used Automatic Stepwise Selection, in which we created –

- Automatic Forward Selection – It suggests using X2,X3,X4,X5,X6 and X7 features, same as Model 2.
- Automatic Backward Selection – It suggests using X1,X2,X3,X4,X5 and X6 features.
- Automatic Selection Both – It suggests using X2,X3,X4,X5,X6 and X7 features, same as Model 2.

Afterwards, we use All Possible Subset Regression through olsrr() function, it suggests using Model 6 as it has the lowest AIC and highest Adj R square. The features are pretty like that of Model 2.

Finally, we made a full fitting regression lm(y ~ x1+x2+x3+x4+x5+x6+x7). After having the summary of X1,X2,X3,X4,X5,X6 and X7, we noticed that all predictors explained only 68%approx of the model. While checking the assumptions, we came to know that our **Model 3** is also not fit as it also violated 2 assumptions.

In the end, we tried to transform the data, rather than improving the models, it worsens the performance of the models. Hence, transformation was of no use to us.

AKSHAT VIJAYVARGIA(S3826627)

CONCLUSION

After all the descriptive statistics and residual checks, we concluded that Model 2 is best model. Though, it violated normality and uncorrelated error assumption, but as per the p-value and F-value it seems promising. We transformed the data with capping function, but it deteriorated the performance of Model 2. So, we had to pick the best model from the raw untransformed data.

REFERENCES

1) Dataset website. Kaggle Repository. Available at - Vehicle dataset | Kaggle [Accessed 2021- 4-14]
2) RMIT Study Materials. Available at - Course modules: Regression Analysis (2110) (instructure.com) [Accessed 2021-5-21]
3) Predicting Target Feature in Multiple Regression. Available at - Multiple Regression Prediction in R | educational research techniques [Accessed 2021-5-25]

AKSHAT VIJAYVARGIA(S3826627)

APPENDIX

```
library(car)

library(MASS)

library(leaps)

library(DAAG)

library(qpcR)

library(olsrr)

library(TSA)

library(readr)

CAR <- read_csv("C:/Users/61422/Desktop/CAR DETAILS FROM CAR DEKHO.csv")

CAR<-CAR[-1]

colnames(CAR)[2]="Y"

colnames(CAR)[1]="X1"

colnames(CAR)[3]="X2"

colnames(CAR)[4]="X3"

colnames(CAR)[5]="X4"

colnames(CAR)[6]="X5"

colnames(CAR)[7]="X6"

colnames(CAR)[8]="X7"

head(CAR)

MANUAL_BACK=lm(log(Y)~., data = CAR)

drop1(MANUAL_BACK,test="F")

drop1(update(MANUAL_BACK, ~ . -X1-X2), test = "F")

LM1=lm(log(Y)~X3+X4+X5+X6+X7,data=CAR)

LM1$coefficients

summary(LM1)

suppressWarnings({par(mfrow=c(2,2))

plot(LM1)})

ncvTest(LM1)

shapiro.test(LM1$residuals)
```

```
acf(LM1$residuals)

durbinWatsonTest(LM1)

vif(LM1)

null=lm(Y~1, data = CAR)

full=lm(Y~ ., data=CAR)

add1(null, scope =full, test = "F")

add1(update(null, ~ . +X5), scope = full, test = "F")

add1(update(null, ~ . +X5+X7), scope = full, test = "F")

add1(update(null, ~ . +X5+X7+X3), scope = full, test = "F")

add1(update(null, ~ . +X5+X7+X3+X2), scope = full, test = "F")

add1(update(null, ~ . +X5+X7+X3+X2+X4), scope = full, test = "F")

add1(update(null, ~ . +X5+X7+X3+X2+X4+X6), scope = full, test = "F")

LM2=lm(log(Y)~X5+X7+X3+X2+X4+X6,data=CAR)

LM2$coefficients

summary(LM2)

suppressWarnings({par(mfrow=c(2,2))

plot(LM2)})

ncvTest(LM2)

shapiro.test(LM2$residuals)

acf(LM2$residuals)

durbinWatsonTest(LM2)

vif(LM2)

step(null, scope=list(lower=null, upper=full), direction="forward") #SUGGESTS MODEL2
LM2

step(full, data=CAR, direction="backward")

step(null, scope = list(upper=full), data=CAR, direction="both") #SUGGESTS MODEL2 LM2

Model=lm(log(Y)~.,data=CAR)

suppressWarnings({ MODELCOMPARE<-ols_step_best_subset(Model)

MODELCOMPARE })

LM3 <- lm(log(Y)~ ., data=CAR)
```

AKSHAT VIJAYVARGIA(S3826627)

```
LM3$coefficients
summary(LM3)
suppressWarnings({par(mfrow=c(2,2))
plot(LM3)})
ncvTest(LM3)
shapiro.test(LM3$residuals)
acf(LM3$residuals)
durbinWatsonTest(LM3)
vif(LM3)
capped<- function(x){
 quantiles <- quantile( x, c(.05, 0.25, 0.75, .95 ) )
 x[ x < quantiles[2] - 1.5*IQR(x) ] <- quantiles[1]
 x[ x > quantiles[3] + 1.5*IQR(x) ] <- quantiles[4]
 x
}
Y_TRANSFORMED <- capped(CAR$Y)
NOM_MOD1 = lm(log(Y_TRANSFORMED) ~ X4+X5+X6+X7, data =CAR)
summary(NOM_MOD1)
par(mfrow=c(2,2))
plot(NOM_MOD1)
ncvTest(NOM_MOD1)
shapiro.test(NOM_MOD1$residuals)
acf(NOM_MOD1$residuals)
durbinWatsonTest(NOM_MOD1)
vif(NOM_MOD1)
capped<- function(x){
 quantiles <- quantile( x, c(.05, 0.25, 0.75, .95 ) )
 x[ x < quantiles[2] - 1.5*IQR(x) ] <- quantiles[1]
 x[ x > quantiles[3] + 1.5*IQR(x) ] <- quantiles[4]
 x
```

```
}
Y_TRANSFORMED <- capped(CAR$Y)
NOM_MOD2 = lm(log(Y_TRANSFORMED) ~ X5+X7+X2+X4+X6, data =CAR)
summary(NOM_MOD2)
par(mfrow=c(2,2))
plot(NOM_MOD2)
ncvTest(NOM_MOD2)
shapiro.test(NOM_MOD2$residuals)
acf(NOM_MOD2$residuals)
durbinWatsonTest(NOM_MOD2)
vif(NOM_MOD2)
capped<- function(x){
 quantiles <- quantile( x, c(.05, 0.25, 0.75, .95 ) )
 x[ x < quantiles[2] - 1.5*IQR(x) ] <- quantiles[1]
 x[ x > quantiles[3] + 1.5*IQR(x) ] <- quantiles[4]
 x
}
Y_TRANSFORMED <- capped(CAR$Y)
NOM_MOD3 = lm(log(Y_TRANSFORMED) ~ X1+X2+X4+X5+X6+X7, data =CAR)
summary(NOM_MOD3)
par(mfrow=c(2,2))
plot(NOM_MOD3)
ncvTest(NOM_MOD3)
shapiro.test(NOM_MOD3$residuals)
acf(NOM_MOD3$residuals)
durbinWatsonTest(NOM_MOD3)
vif(NOM_MOD3)
library(ISLR)
library(ggplot2)
library(caret)
```

AKSHAT VIJAYVARGIA(S3826627)

```
SAMPLE<-createDataPartition(y=CAR$Y,
 p=0.8, list=FALSE)
TRAINING_SET <- CAR[SAMPLE, ]
TESTING_SET <- CAR[-SAMPLE, ]
dim(TRAINING_SET)
dim(TESTING_SET)
TRAINING_MODEL<-lm(Y~ .,data=CAR)
summary(TRAINING_MODEL)
suppressWarnings({ CHECK_TRAINING_MODEL<-train(Y~ .,method="lm",data=CAR) })
DOUBLECHECK_TRAINING_MODEL<-CHECK_TRAINING_MODEL$finalModel
X1<-c(2017,2002,1998,2019)
X2<-c(10000,150000,45000,80000)
X3<-c('CNG','Petrol','LPG','Diesel')
X4<-c('Dealer','Individual','Dealer','Individual')
X5<-c('Automatic','Automatic','Automatic','Automatic')
X6<-c('Second Owner','First Owner','Third Owner','First Owner')
X7<-c(4,19,23,2)
DATA<-data.frame(X1,X2,X3,X4,X5,X6,X7)
suppressWarnings({predict.lm(TRAINING_MODEL,DATA)})
```