ToyotaCorolla_regression

Priya Roopa

11/9/2019

Input data, choose predictors

```
car.df <- read.csv("ToyotaCorolla.csv") # Read the ToyotaCorolla CSV file
car.df <- car.df[1:1000, ] # use first 1000 rows of data
View(car.df)# view the  car dataset
t(t(names(car.df)))#names(car.df)holds names of columns in car.df. Transpose
column names of dataset twice to get output.
```

```
##       [,1]
##  [1,] "Id"
##  [2,] "Model"
##  [3,] "Price"
##  [4,] "Age_08_04"
##  [5,] "Mfg_Month"
##  [6,] "Mfg_Year"
##  [7,] "KM"
##  [8,] "Fuel_Type"
##  [9,] "HP"
## [10,] "Met_Color"
## [11,] "Color"
## [12,] "Automatic"
## [13,] "CC"
## [14,] "Doors"
## [15,] "Cylinders"
## [16,] "Gears"
## [17,] "Quarterly_Tax"
## [18,] "Weight"
## [19,] "Mfr_Guarantee"
## [20,] "BOVAG_Guarantee"
## [21,] "Guarantee_Period"
## [22,] "ABS"
## [23,] "Airbag_1"
## [24,] "Airbag_2"
## [25,] "Airco"
## [26,] "Automatic_airco"
## [27,] "Boardcomputer"
## [28,] "CD_Player"
## [29,] "Central_Lock"
## [30,] "Powered_Windows"
## [31,] "Power_Steering"
## [32,] "Radio"
## [33,] "Mistlamps"
```

```
## [34,] "Sport_Model"
## [35,] "Backseat_Divider"
## [36,] "Metallic_Rim"
## [37,] "Radio_cassette"
## [38,] "Parking_Assistant"
## [39,] "Tow_Bar"

selected.var <- c(3, 4, 7, 8, 9, 10, 12, 13, 14, 17, 18)# select variables
for regression
```

###Summary:Data were collected on all previous sales of used Toyota Corollas at the dealership and saved as ToyotaCorolla.csv. The data include the sales price and other information on the car, such as its age, mileage, fuel type, and engine size.There are 39 variables in total.The total number of records in the dataset is 1000 cars (we used the first 1000 cars from the dataset ToyotoCorolla.csv).We are selecting the predictors by reducing the number of variables from 39 to 11 variables(“Price”,“Age”,“KM”,“Fuel_Type”,“HP”,“Met_Color”,“Automatic”,“CC”,“Doors”,“Quarterly_tax”,“Weight”)

Partition the data

```
set.seed(1)  # set seed for reproducing the partition
train.index <- sample(c(1:1000), 600)#Here we are taking 60% of the data as
training set.Train.index holds 600 index of rows of the 1000
head(train.index)#first several rows of train.index

## [1] 836 679 129 930 509 471

train.df <- car.df[train.index, selected.var]# Assigning sampled 60% of the
data to training set.
valid.df <- car.df[-train.index, selected.var]#Assigning the remaining 40% of
the data to validation
head(valid.df)#Several first rows of valid.df

##    Price Age_08_04    KM Fuel_Type  HP Met_Color Automatic   CC Doors
## 2  13750        23 72937    Diesel  90         1         0 2000     3
## 7  16900        27 94612    Diesel  90         1         0 2000     3
## 8  18600        30 75889    Diesel  90         1         0 2000     3
## 9  21500        27 19700    Petrol 192         0         0 1800     3
## 10 12950        23 71138    Diesel  69         0         0 1900     3
## 12 19950        22 43610    Petrol 192         0         0 1800     3
##    Quarterly_Tax Weight
## 2            210   1165
## 7            210   1245
## 8            210   1245
## 9            100   1185
## 10           185   1105
## 12           100   1185

head(train.df)#several first rows of train.df
```

```
##     Price Age_08_04     KM Fuel_Type  HP Met_Color Automatic   CC Doors
## 836  9750        67  67762    Petrol 110         1         0 1600     3
## 679  9895        68 102494    Petrol 110         0         0 1600     5
## 129 17950        17  33740    Petrol  97         1         0 1400     5
## 930  9995        57  55844    Petrol  86         1         0 1300     5
## 509 10500        50  54465    Petrol 110         0         0 1600     5
## 471 10900        50  65471    Petrol  97         1         0 1400     5
##     Quarterly_Tax Weight
## 836            85   1065
## 679            85   1090
## 129            85   1135
## 930            69   1045
## 509            85   1075
## 471            85   1060
```

###Summary : Here, we are splitting the data set into training ie 60% of the data(600 rows out of 1000) and validating ie 40% of the remaining data(400 rows out of 1000).We have taken the data set for tarining and validating which are selected variables for regression ie Price,Age,KM,Fuel type,HP,Metallic color,Automatic Transmission,CC,Doors,Quarterly Tax,Weight.

Run model

```
car.lm <- lm(Price ~ ., data = train.df)# use lm() to run a linear regression of Price on all 11 predictors in the training set. use . after ~ to include all the remaining columns in train.df as predictors.
options(scipen = 999)
summary(car.lm)

##
## Call:
## lm(formula = Price ~ ., data = train.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9781.2  -729.9     0.9   739.3  6912.9
##
## Coefficients:
##                     Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)     -4754.379821  1661.719608  -2.861             0.004372 **
## Age_08_04        -133.271592     4.901960 -27.187 < 0.0000000000000002 ***
## KM                 -0.020992     0.002304  -9.111 < 0.0000000000000002 ***
## Fuel_TypeDiesel   896.206322   603.164063   1.486             0.137857
## Fuel_TypePetrol  2191.368250   575.629429   3.807             0.000155 ***
## HP                 37.257956     5.233283   7.119    0.00000000000317 ***
## Met_Color          51.315188   123.395390   0.416             0.677664
## Automatic          63.567598   262.282017   0.242             0.808583
## CC                  0.010747     0.097711   0.110             0.912456
## Doors             -55.700492    63.966255  -0.871             0.384230
## Quarterly_Tax      13.080021     2.608396   5.015    0.00000070465597 ***
## Weight             16.219638     1.526915  10.622 < 0.0000000000000002 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1392 on 588 degrees of freedom
## Multiple R-squared:  0.8703, Adjusted R-squared:  0.8679
## F-statistic: 358.7 on 11 and 588 DF,  p-value: < 0.00000000000000022
```

Summary:Residuals is the difference between the actual observed response values and the response values that the model predicted. We can see that the distribution of the residuals do not appear to be strongly symmetrical (-9781.2 to 6912.9 with median at 0.9). That means that the model predicts certain points that fall far away from the actual observed points.

###Coefficients ### Intercept: a negative value for intercept means that the expected value on our dependent variable (Price) will be less than 0 when all independent/predictor variables are set to 0.

###Estimate: The effect of predictor on the response variable value. For instance, for Age_08_04, as the age increases, the price drops by -133.271592

###Standard Error: measures the average amount that the coefficient estimates vary from the actual average value of our response variable. For instance, Age_08_04 estimate can vary by 4.901960.

###t-value ###t-value far from zero indicates that we can reject the null hypothesis - that is we could declare a relation between Price and Age_08_04.

###Pr(>t) ###A small p-value indicates that it is unlikely we will observe a relationship between the predictor (Age_08_04) and response variables (Price) due to chance. Typically, a p-value of 5% or less is a good cut-off point.

###t-value far from zero and small Pr(>t), typically less than 5% indicates that we can reject the null hypothesis or declare that there exists a relation between the predictor (e.g., Age_08_04) and the response variables (e.g., Price). So, the predictors that have relationship with our response variable Price are: Age_08_04, KM, Fuel_TypePetrol, HP, Quarterly_Tax, and Weight. Other predictors do not seem to have strong relationship with Price.

###Residual standard error ###measure of the quality of a linear regression fit.The average amount that the response (Price) will deviate from the true regression line.In our case it is 1392.

###Multiple R-squared, Adjusted R-squared ###measure of how well the model is fitting the actual data.Measure of the linear relationship between our predictor variable (e.g., Age_08_04) and our response / target variable (Price) ###In our example, the R-squared we get is 0.8703. Or roughly 87% of the variance found in the response variable (Price) can be explained by the predictor variables (e.g., Age_08_04).

###F-Statistic is a indicator of whether there is a relationship between our predictor and the response variables. The further the F-statistic is from 1 the better it is. When the number of data points is large, an F-statistic that is only a little bit larger than 1 is already sufficient to reject the null hypothesis (H0 : There is no relationship between target variable and predictor variables). In our case it is: 358.7, which indicates that we can reject the null hypothesis.

Make predictions on a hold-out set

```
library(forecast)
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

```
# use predict() to make predictions on a new set.
car.lm.pred <- predict(car.lm, valid.df)# predicting the price of validation
data using the linear regression model.car.lm.pred is an array of 400 price
predictions.
options(scipen=999, digits = 0)
some.residuals <- valid.df$Price[1:20] - car.lm.pred[1:20]#computing the
diffrence between the validation dataset price and predicted prices for the
first 20 prices .
data.frame("Predicted" = car.lm.pred[1:20], "Actual" = valid.df$Price[1:20],
           "Residual" = some.residuals)# creating a table with 3 columns
showing Predicted,Actual and Residual prices and the table contains 20 rows.
```

```
##    Predicted Actual Residual
## 2      16447  13750    -2697
## 7      16757  16900      143
## 8      16750  18600     1850
## 9      20959  21500      541
## 10     14350  12950    -1400
## 12     21124  19950    -1174
## 13     20964  19600    -1364
## 14     20408  21500     1092
## 18     16817  17950     1133
## 21     15053  15950      897
## 23     15800  15950      150
## 24     16307  16950      643
## 26     16786  15950     -836
## 30     16484  17950     1466
```

```
## 32     16233  15750     -483
## 34     15752  14950     -802
## 36     15485  15750      265
## 38     16629  14950    -1679
## 46     18069  19000      931
## 47     17441  17950      509
```

```
options(scipen=999, digits = 3)
# use accuracy() to compute common accuracy measures.
accuracy(car.lm.pred, valid.df$Price)
```

```
##            ME RMSE  MAE   MPE MAPE
## Test set 19.6 1325 1049 -0.75 9.35
```

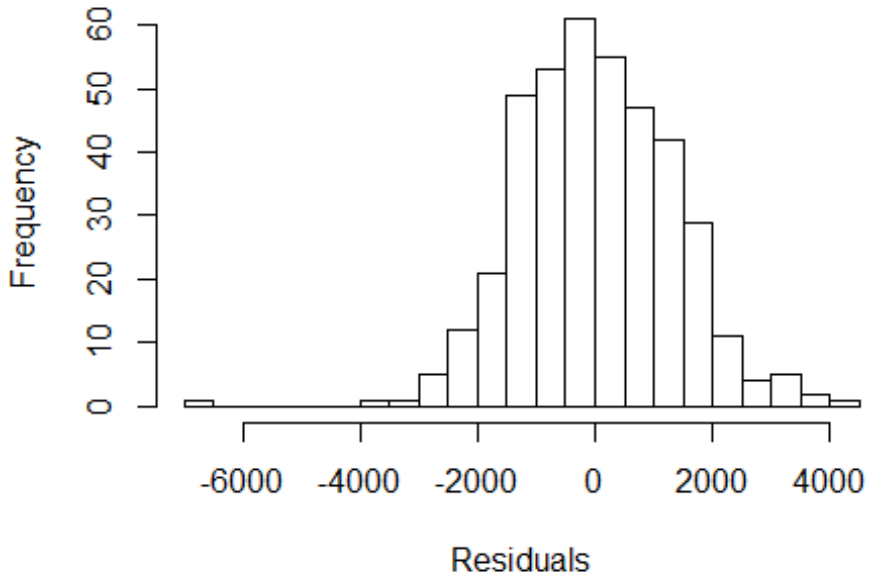
###Summary: Here, we have created 3 columns showing the predicted,Actual and residual prices and the table contains 20 rows.The function accuracy gives us multiple measures of accuracy of the model fit. ##Mean error(ME) 19.6 is an informal term that usually refers to the average of all errors in a set. ### The Root Mean Square Error(RMSE)value is 1325.The RMSE is the square root of the variance of the residuals. It indicates the absolute fit of the model to the data–how close the observed data points are to the model's predicted values. The RMSE will always be larger or equal to the MAE.The value of MAE is 1049.We can interpret that the average diffrence between the predicted and the actual price is 276.The greater difference between them, the greater the variance in the individual errors. ###Mean Absolute Error(MAE) value is 1049.It is the absolute average of the difference between predicted and actual values of price in the test. ###Mean Percentage Error(MPE)is -0.75.It is average value of percentage errors by which predicted values differ from actual values. A negetive value here indicates that the predicted values are less than the actual values. ###Mean Absolute Percentage Error(MAPE) value is 9.35.It is a measure to validate predicted values. MAPE states that our model's predictions are, on average, 9.35% off from actual value.

###Histogram of residuals

```
library(forecast)
car.lm.pred <- predict(car.lm, valid.df)# predicting the price of validation
data using the linear regression model.car.lm.pred is an array of 400 price
predictions.
all.residuals <- valid.df$Price - car.lm.pred #computing the diffrence
between the validation data price and price predictions.
length(all.residuals[which(all.residuals > -2000 & all.residuals <
2000)])/400#length of all residual values that are greater than -2000 and all
residual values less than 2000.
```

```
## [1] 0.892
```

```
hist(all.residuals, breaks = 25, xlab = "Residuals", main = "")
```



###0.892……. ###Summary:The Histogram of the Residual can be used to check whether the variance is normally distributed.Here, the histogram shows that the residurals are normally distributed, however it is negetively left skewed and also has outliers.

Run an exhaustive search for the best model

```
# use regsubsets() in package leaps to run an exhaustive search.
# unlike with lm, categorical predictors must be turned into dummies
manually.

# create dummies for fuel type
train.df <- car.df[train.index, selected.var]# Assigning sampled 60% of the
data to training set.
valid.df <- car.df[-train.index, selected.var]##Assigning the remaining 40%
of the data to validation
train.index <- sample(c(1:1000), 600)#Here we are taking 60% of the data as
training set.Train.index holds 600 index of rows of the 1000
train.df <- car.df[train.index, selected.var]# Assigning sampled 60% of the
data to training set.
dim(train.df)##dimension of training data set.contains 600 rows and 11
columns.

## [1] 600  11

Fuel_Type1 <- as.data.frame(model.matrix(~ 0 + Fuel_Type, data=train.df))#
Fuel_Type1 is a table with 600 rows and three newly created columns. Here,
Fuel_Type column is split into three columns: Fuel_TypeCNG, Fuel_TypeDiesel,
```

```
and Fuel_TypePetrol. The values for these columns are 0 or 1, indicating
whether the original column had CNG, Disel, or Petrol.
# replace Fuel_Type column with 2 dummies
train.df <- cbind(train.df[,-4], Fuel_Type1[,])# Fuel_Type column is removed
and three new Fuel_Type1 columns are inserted into train.df. train.df now
contains 13 columns.
head(train.df)
```

```
##     Price Age_08_04    KM  HP Met_Color Automatic   CC Doors Quarterly_Tax
## 979  8745        65 45681 110         0         0 1600     3            69
## 209 11450        41 84312 110         0         0 1600     5            85
## 148 24500        13 19988 110         1         0 1600     5            85
## 704 10500        65 93428 110         1         0 1600     5            85
## 501  9700        51 57645 110         0         0 1600     5            85
## 343 14950        42 29640 110         0         0 1600     3            85
##     Weight Fuel_TypeCNG Fuel_TypeDiesel Fuel_TypePetrol
## 979   1050            0               0               1
## 209   1080            0               0               1
## 148   1130            0               0               1
## 704   1075            0               0               1
## 501   1080            0               0               1
## 343   1055            0               0               1
```

```
Fuel_Type2 <- as.data.frame(model.matrix(~ 0 + Fuel_Type, data=valid.df))#
Similar to Fuel_Type1, but for the valid.df.
# replace Fuel_Type column with 2 dummies
valid.df <- cbind(valid.df[,-4], Fuel_Type2[,])# Replaced Fuel_Type column
with three new columns from Fuel_Type2. Now valid.df has 13 columns.
head(valid.df)
```

```
##    Price Age_08_04    KM  HP Met_Color Automatic   CC Doors Quarterly_Tax
## 2  13750        23 72937  90         1         0 2000     3           210
## 7  16900        27 94612  90         1         0 2000     3           210
## 8  18600        30 75889  90         1         0 2000     3           210
## 9  21500        27 19700 192         0         0 1800     3           100
## 10 12950        23 71138  69         0         0 1900     3           185
## 12 19950        22 43610 192         0         0 1800     3           100
##    Weight Fuel_TypeCNG Fuel_TypeDiesel Fuel_TypePetrol
## 2    1165            0               1               0
## 7    1245            0               1               0
## 8    1245            0               1               0
## 9    1185            0               0               1
## 10   1105            0               1               0
## 12   1185            0               0               1
```

```
dim(valid.df)
```

```
## [1] 400  13
```

```
#install.packages("leaps")
library(leaps)
```

```r
search <- regsubsets(Price ~ ., data = train.df, nbest = 1, nvmax =
dim(train.df)[2],
                     method = "exhaustive") # search is a list of 28 models
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : nvmax reduced to 11
```

```r
sum <- summary(search)# regsubsets helps with variable selection. We have 12
variables and summary provides their importance across 11 models.

# show models
sum$which # provides which variables (or predictors) are included
(TRUE/FALSE) in the models (there are 11 models)
```

```
##    (Intercept) Age_08_04    KM    HP Met_Color Automatic    CC Doors
## 1         TRUE      TRUE FALSE FALSE     FALSE     FALSE FALSE FALSE
## 2         TRUE      TRUE FALSE FALSE     FALSE     FALSE FALSE FALSE
## 3         TRUE      TRUE FALSE FALSE     FALSE     FALSE FALSE FALSE
## 4         TRUE      TRUE  TRUE FALSE     FALSE     FALSE FALSE FALSE
## 5         TRUE      TRUE  TRUE FALSE     FALSE     FALSE FALSE  TRUE
## 6         TRUE      TRUE  TRUE FALSE     FALSE     FALSE FALSE  TRUE
## 7         TRUE      TRUE  TRUE FALSE     FALSE      TRUE FALSE  TRUE
## 8         TRUE      TRUE  TRUE  TRUE     FALSE      TRUE FALSE  TRUE
## 9         TRUE      TRUE  TRUE  TRUE     FALSE      TRUE  TRUE  TRUE
## 10        TRUE      TRUE  TRUE  TRUE     FALSE      TRUE  TRUE  TRUE
## 11        TRUE      TRUE  TRUE  TRUE      TRUE      TRUE  TRUE  TRUE
##    Quarterly_Tax Weight Fuel_TypeCNG Fuel_TypeDiesel Fuel_TypePetrol
## 1          FALSE  FALSE        FALSE            FALSE           FALSE
## 2          FALSE   TRUE        FALSE            FALSE           FALSE
## 3          FALSE   TRUE        FALSE            FALSE            TRUE
## 4          FALSE   TRUE        FALSE            FALSE            TRUE
## 5          FALSE   TRUE        FALSE            FALSE            TRUE
## 6           TRUE   TRUE        FALSE            FALSE            TRUE
## 7           TRUE   TRUE        FALSE            FALSE            TRUE
## 8           TRUE   TRUE        FALSE            FALSE            TRUE
## 9           TRUE   TRUE        FALSE            FALSE            TRUE
## 10          TRUE   TRUE         TRUE            FALSE            TRUE
## 11          TRUE   TRUE         TRUE             TRUE           FALSE
```

```r
# show metrics
sum$rsq # the r-squared metric for all the 11 models
```

```
##  [1] 0.749 0.812 0.875 0.885 0.890 0.891 0.893 0.894 0.895 0.896 0.896
```

```r
sum$adjr2 # adjusted r-squared metric for all the 11 models
```

```
##  [1] 0.748 0.812 0.874 0.884 0.889 0.890 0.892 0.892 0.893 0.894 0.894
```

```r
sum$cp # mallow's cp metric for all the 11 models
```

```
##  [1] 820.0 463.1 114.1  59.4  34.2  25.6  19.0  15.9  10.9  10.5  11.0
```

Summary: The exhaustive search using regsubsets showed inclusion of predictors across 11 models. We are looking for predictors that are not included for most of the models to deem them unimportant. For instance, Fuel_TypeCNG is only included in one model. CC and Met_Color are other such predictors. Whereas, some predictors like Age_08_04 is included in all the models. Finally, rsq, adjr2, and cp metrics are shown.

use step() to run stepwise regression, backward selection.

```
head(valid.df)#first 6 rows of validation data set.
```

```
##    Price Age_08_04    KM  HP Met_Color Automatic   CC Doors Quarterly_Tax
## 2  13750        23 72937  90         1         0 2000     3           210
## 7  16900        27 94612  90         1         0 2000     3           210
## 8  18600        30 75889  90         1         0 2000     3           210
## 9  21500        27 19700 192         0         0 1800     3           100
## 10 12950        23 71138  69         0         0 1900     3           185
## 12 19950        22 43610 192         0         0 1800     3           100
##    Weight Fuel_TypeCNG Fuel_TypeDiesel Fuel_TypePetrol
## 2    1165            0               1               0
## 7    1245            0               1               0
## 8    1245            0               1               0
## 9    1185            0               0               1
## 10   1105            0               1               0
## 12   1185            0               0               1
```

```
head(train.df)# first 6 rows of training data set.
```

```
##     Price Age_08_04    KM  HP Met_Color Automatic   CC Doors Quarterly_Tax
## 979  8745        65 45681 110         0         0 1600     3            69
## 209 11450        41 84312 110         0         0 1600     5            85
## 148 24500        13 19988 110         1         0 1600     5            85
## 704 10500        65 93428 110         1         0 1600     5            85
## 501  9700        51 57645 110         0         0 1600     5            85
## 343 14950        42 29640 110         0         0 1600     3            85
##     Weight Fuel_TypeCNG Fuel_TypeDiesel Fuel_TypePetrol
## 979   1050            0               0               1
## 209   1080            0               0               1
## 148   1130            0               0               1
## 704   1075            0               0               1
## 501   1080            0               0               1
## 343   1055            0               0               1
```

```
car.lm <- lm(Price ~ ., data = train.df)# use lm() to run a linear regression
of Price on all 11 predictors in the training set. use . after ~ to include
all the remaining columns in train.df as predictors.
```

```
car.lm.step <- step(car.lm, direction = "backward")#computing backward
selection on the linear regression model car.lm using step command.

## Start:  AIC=8541
## Price ~ Age_08_04 + KM + HP + Met_Color + Automatic + CC + Doors +
##     Quarterly_Tax + Weight + Fuel_TypeCNG + Fuel_TypeDiesel +
##     Fuel_TypePetrol
##
##
## Step:  AIC=8541
## Price ~ Age_08_04 + KM + HP + Met_Color + Automatic + CC + Doors +
##     Quarterly_Tax + Weight + Fuel_TypeCNG + Fuel_TypeDiesel
##
##                   Df Sum of Sq        RSS  AIC
## - Met_Color        1   2266092  878937940 8540
## <none>                         876671847 8541
## - Automatic        1   8366413  885038261 8545
## - CC               1  14321587  890993434 8549
## - Fuel_TypeDiesel  1  14811707  891483554 8549
## - Quarterly_Tax    1  15756166  892428014 8550
## - HP               1  18363944  895035791 8551
## - Doors            1  25213122  901884970 8556
## - Fuel_TypeCNG     1  54031878  930703726 8575
## - KM               1  68220130  944891977 8584
## - Weight           1 389146231 1265818078 8759
## - Age_08_04        1 758018466 1634690313 8913
##
## Step:  AIC=8540
## Price ~ Age_08_04 + KM + HP + Automatic + CC + Doors + Quarterly_Tax +
##     Weight + Fuel_TypeCNG + Fuel_TypeDiesel
##
##                   Df Sum of Sq        RSS  AIC
## <none>                         878937940 8540
## - Automatic        1   8574722  887512662 8544
## - CC               1  13626991  892564931 8548
## - Fuel_TypeDiesel  1  14991775  893929715 8549
## - Quarterly_Tax    1  15326578  894264518 8549
## - HP               1  18229552  897167491 8551
## - Doors            1  24564771  903502711 8555
## - Fuel_TypeCNG     1  53685703  932623643 8574
## - KM               1  68369633  947307572 8583
## - Weight           1 388073155 1267011095 8758
## - Age_08_04        1 775312160 1654250100 8918

summary(car.lm.step) # Which variables did it drop?##
Met_color,Fuel_typePetrol got dropped.

##
## Call:
## lm(formula = Price ~ Age_08_04 + KM + HP + Automatic + CC + Doors +
```

```
##     Quarterly_Tax + Weight + Fuel_TypeCNG + Fuel_TypeDiesel,
##     data = train.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -3944   -833   -120    740   6034
##
## Coefficients:
##                     Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)     -20308.35538   2279.27485   -8.91 < 0.0000000000000002 ***
## Age_08_04         -108.81655      4.77395  -22.79 < 0.0000000000000002 ***
## KM                  -0.01530      0.00226   -6.77       0.000000000031 ***
## HP                  28.24247      8.08046    3.50              0.00051 ***
## Automatic         -614.56814    256.37842   -2.40              0.01684 *
## CC                  -2.33286      0.77199   -3.02              0.00262 **
## Doors             -241.72075     59.57706   -4.06       0.000056338057 ***
## Quarterly_Tax        8.29217      2.58742    3.20              0.00142 **
## Weight              36.83627      2.28423   16.13 < 0.0000000000000002 ***
## Fuel_TypeCNG     -3332.89616    555.66612   -6.00       0.000000003487 ***
## Fuel_TypeDiesel  -2146.48096    677.20783   -3.17              0.00161 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1220 on 589 degrees of freedom
## Multiple R-squared:  0.896,  Adjusted R-squared:  0.894
## F-statistic:  505 on 10 and 589 DF,  p-value: <0.0000000000000002
```

```
car.lm.step.pred <- predict(car.lm.step, valid.df)# predicting the price of
validation data using the linear stepwise regression model.car.lm.step .
accuracy(car.lm.step.pred, valid.df$Price)##accuracy of the predicting price
of validation data using actual validation price
```

```
##             ME RMSE  MAE   MPE MAPE
## Test set -4.92 1350 1017 -1.19 9.09
```

Summary: Backward selection (or backward elimination), which starts with all predictors in the model (full model), iteratively removes the least contributive predictors, and stops when you have a model where all predictors are statistically significant. Accordingly, least contributive predictors Fuel_typePetrol,Met_color were removed .The mean error(ME) is -25.5. The Root mean square error(RMSE) is 1362 and the Mean absolute error(MAE)is 1019.The diffrence between RMSE and MAE is 343.The greater difference between them, the greater the variance in the individual errors.Mean percentage error is -1.42. And the Mean absolute percentage error is 9.13.

Forward selection

```
car.lm <- lm(Price ~ ., data = train.df)#use lm() to run a linear regression
of Price on all 11 predictors in the training set. use . after ~ to include
all the remaining columns in train.df as predictors.
car.lm.step <- step(car.lm, direction = "forward")#computing Forward
selection on the linear regression model car.lm using step command

## Start:  AIC=8541
## Price ~ Age_08_04 + KM + HP + Met_Color + Automatic + CC + Doors +
##     Quarterly_Tax + Weight + Fuel_TypeCNG + Fuel_TypeDiesel +
##     Fuel_TypePetrol

summary(car.lm.step) #

##
## Call:
## lm(formula = Price ~ Age_08_04 + KM + HP + Met_Color + Automatic +
##     CC + Doors + Quarterly_Tax + Weight + Fuel_TypeCNG + Fuel_TypeDiesel +
##     Fuel_TypePetrol, data = train.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -3844   -834    -80    740   6014
##
## Coefficients: (1 not defined because of singularities)
##                     Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)     -20399.20522   2279.46102   -8.95 < 0.0000000000000002 ***
## Age_08_04         -108.19499      4.79841  -22.55 < 0.0000000000000002 ***
## KM                  -0.01529      0.00226   -6.76       0.000000000032 ***
## HP                  28.34798      8.07735    3.51              0.00048 ***
## Met_Color          132.93914    107.83107    1.23              0.21813
## Automatic         -607.22139    256.33462   -2.37              0.01817 *
## CC                  -2.39701      0.77340   -3.10              0.00203 **
## Doors             -245.15818     59.61602   -4.11       0.000044757491 ***
## Quarterly_Tax        8.41368      2.58816    3.25              0.00122 **
## Weight              36.89526      2.28373   16.16 < 0.0000000000000002 ***
## Fuel_TypeCNG     -3344.06956    555.49498   -6.02       0.000000003072 ***
```

```
## Fuel_TypeDiesel  -2133.79761    676.98732   -3.15              0.00170 **
## Fuel_TypePetrol           NA           NA      NA                   NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1220 on 588 degrees of freedom
## Multiple R-squared:  0.896,  Adjusted R-squared:  0.894
## F-statistic:  460 on 11 and 588 DF,  p-value: <0.0000000000000002
```

###Summary:Forward selection,which starts with no predictors in the model, iteratively adds the most contributive predictors, and stops when the improvement is no longer statistically significant.so, here the most contributive predictors are Age_08_04 , KM , HP , Met_Color , Automatic ,CC, Doors, Quarterly_Tax, Weight, Fuel_TypeCNG, Fuel_TypeDiesel, Fuel_TypePetrol.

#Stepwise

```
# use step() to run stepwise regression.
car.lm <- lm(Price ~ ., data = train.df)#use lm() to run a linear regression
of Price on all 11 predictors in the training set. use . after ~ to include
all the remaining columns in train.df as predictors.
car.lm.step <- step(car.lm, direction = "both")#computing Forward selection
and backward selection on the linear regression model car.lm using step
command.
```

```
## Start:  AIC=8541
## Price ~ Age_08_04 + KM + HP + Met_Color + Automatic + CC + Doors +
##     Quarterly_Tax + Weight + Fuel_TypeCNG + Fuel_TypeDiesel +
##     Fuel_TypePetrol
##
##
## Step:  AIC=8541
## Price ~ Age_08_04 + KM + HP + Met_Color + Automatic + CC + Doors +
##     Quarterly_Tax + Weight + Fuel_TypeCNG + Fuel_TypeDiesel
##
##                   Df Sum of Sq        RSS  AIC
## - Met_Color        1   2266092  878937940 8540
## <none>                          876671847 8541
## - Automatic        1   8366413  885038261 8545
## - CC               1  14321587  890993434 8549
## - Fuel_TypeDiesel  1  14811707  891483554 8549
## - Quarterly_Tax    1  15756166  892428014 8550
## - HP               1  18363944  895035791 8551
## - Doors            1  25213122  901884970 8556
## - Fuel_TypeCNG     1  54031878  930703726 8575
## - KM               1  68220130  944891977 8584
## - Weight           1 389146231 1265818078 8759
## - Age_08_04        1 758018466 1634690313 8913
##
## Step:  AIC=8540
```

```
## Price ~ Age_08_04 + KM + HP + Automatic + CC + Doors + Quarterly_Tax +
##     Weight + Fuel_TypeCNG + Fuel_TypeDiesel
##
##                   Df Sum of Sq        RSS  AIC
## <none>                          878937940 8540
## + Met_Color        1   2266092  876671847 8541
## - Automatic        1   8574722  887512662 8544
## - CC               1  13626991  892564931 8548
## - Fuel_TypeDiesel  1  14991775  893929715 8549
## - Quarterly_Tax    1  15326578  894264518 8549
## - HP               1  18229552  897167491 8551
## - Doors            1  24564771  903502711 8555
## - Fuel_TypeCNG     1  53685703  932623643 8574
## - KM               1  68369633  947307572 8583
## - Weight           1 388073155 1267011095 8758
## - Age_08_04        1 775312160 1654250100 8918
```

```
summary(car.lm.step) #which variables were added/dropped?
```

```
##
## Call:
## lm(formula = Price ~ Age_08_04 + KM + HP + Automatic + CC + Doors +
##     Quarterly_Tax + Weight + Fuel_TypeCNG + Fuel_TypeDiesel,
##     data = train.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -3944   -833   -120    740   6034
##
## Coefficients:
##                     Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)     -20308.35538   2279.27485   -8.91 < 0.0000000000000002 ***
## Age_08_04         -108.81655      4.77395  -22.79 < 0.0000000000000002 ***
## KM                  -0.01530      0.00226   -6.77       0.000000000031 ***
## HP                  28.24247      8.08046    3.50              0.00051 ***
## Automatic         -614.56814    256.37842   -2.40              0.01684 *
## CC                  -2.33286      0.77199   -3.02              0.00262 **
## Doors             -241.72075     59.57706   -4.06       0.000056338057 ***
## Quarterly_Tax        8.29217      2.58742    3.20              0.00142 **
## Weight              36.83627      2.28423   16.13 < 0.0000000000000002 ***
## Fuel_TypeCNG     -3332.89616    555.66612   -6.00       0.000000003487 ***
## Fuel_TypeDiesel  -2146.48096    677.20783   -3.17              0.00161 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1220 on 589 degrees of freedom
## Multiple R-squared:  0.896,  Adjusted R-squared:  0.894
## F-statistic:  505 on 10 and 589 DF,  p-value: <0.0000000000000002
```

```
car.lm.step.pred <- predict(car.lm.step, valid.df)# predicting the price of
validation data using the linear stepwise regression model.car.lm.step .
accuracy(car.lm.step.pred, valid.df$Price)##accuracy of the predicting price
of validation data using actual validation price

##              ME RMSE  MAE   MPE MAPE
## Test set -4.92 1350 1017 -1.19 9.09
```

###Summary :which is a combination of forward and backward selections. You start with no predictors, then sequentially add the most contributive predictors (like forward selection). After adding each new variable, remove any variables that no longer provide an improvement in the model fit (like backward selection).Age_08_04,KM, HP,Doors,Quarterly_Tax,Weight,Fuel_TypeCNG, Fuel_TypeDiesel,CC,Automatic are the contributive predictors.Fuel_typePetrol,Met_color are the least contributive predictors which were dropped.The mean error(ME) is -25.5. The Root mean square error(RMSE) is 1362 and the Mean absolute error(MAE)is 1019.The diffrence between RMSE and MAE is 343.The greater difference between them, the greater the variance in the individual errors.Mean percentage error is -1.42. And the Mean absolute percentage error is 9.13.