Linear Regression Example

Jeff Bostwick

11/10/2020

Table of Contents

## Load Packages and Explore Data

Loading Packages

library(sjPlot)  
library(dplyr)  
library(sjlabelled)  
library(sjmisc)  
library(ggplot2)  
theme\_set(theme\_sjplot())

Loading Dataset

Previewing the Dataset

head(data)

## Make Model Year Engine.Fuel.Type Engine.HP Engine.Cylinders  
## 1 BMW 1 Series M 2011 premium unleaded (required) 335 6  
## 2 BMW 1 Series 2011 premium unleaded (required) 300 6  
## 3 BMW 1 Series 2011 premium unleaded (required) 300 6  
## 4 BMW 1 Series 2011 premium unleaded (required) 230 6  
## 5 BMW 1 Series 2011 premium unleaded (required) 230 6  
## 6 BMW 1 Series 2012 premium unleaded (required) 230 6  
## Transmission.Type Driven\_Wheels Number.of.Doors  
## 1 MANUAL rear wheel drive 2  
## 2 MANUAL rear wheel drive 2  
## 3 MANUAL rear wheel drive 2  
## 4 MANUAL rear wheel drive 2  
## 5 MANUAL rear wheel drive 2  
## 6 MANUAL rear wheel drive 2  
## Market.Category Vehicle.Size Vehicle.Style highway.MPG  
## 1 Factory Tuner,Luxury,High-Performance Compact Coupe 26  
## 2 Luxury,Performance Compact Convertible 28  
## 3 Luxury,High-Performance Compact Coupe 28  
## 4 Luxury,Performance Compact Coupe 28  
## 5 Luxury Compact Convertible 28  
## 6 Luxury,Performance Compact Coupe 28  
## city.mpg Popularity MSRP  
## 1 19 3916 46135  
## 2 19 3916 40650  
## 3 20 3916 36350  
## 4 18 3916 29450  
## 5 18 3916 34500  
## 6 18 3916 31200

Viewing the Structure of the Dataset

str(data)

## 'data.frame': 11914 obs. of 16 variables:  
## $ Make : chr "BMW" "BMW" "BMW" "BMW" ...  
## $ Model : chr "1 Series M" "1 Series" "1 Series" "1 Series" ...  
## $ Year : int 2011 2011 2011 2011 2011 2012 2012 2012 2012 2013 ...  
## $ Engine.Fuel.Type : chr "premium unleaded (required)" "premium unleaded (required)" "premium unleaded (required)" "premium unleaded (required)" ...  
## $ Engine.HP : int 335 300 300 230 230 230 300 300 230 230 ...  
## $ Engine.Cylinders : int 6 6 6 6 6 6 6 6 6 6 ...  
## $ Transmission.Type: chr "MANUAL" "MANUAL" "MANUAL" "MANUAL" ...  
## $ Driven\_Wheels : chr "rear wheel drive" "rear wheel drive" "rear wheel drive" "rear wheel drive" ...  
## $ Number.of.Doors : int 2 2 2 2 2 2 2 2 2 2 ...  
## $ Market.Category : chr "Factory Tuner,Luxury,High-Performance" "Luxury,Performance" "Luxury,High-Performance" "Luxury,Performance" ...  
## $ Vehicle.Size : chr "Compact" "Compact" "Compact" "Compact" ...  
## $ Vehicle.Style : chr "Coupe" "Convertible" "Coupe" "Coupe" ...  
## $ highway.MPG : int 26 28 28 28 28 28 26 28 28 27 ...  
## $ city.mpg : int 19 19 20 18 18 18 17 20 18 18 ...  
## $ Popularity : int 3916 3916 3916 3916 3916 3916 3916 3916 3916 3916 ...  
## $ MSRP : int 46135 40650 36350 29450 34500 31200 44100 39300 36900 37200 ...

Summary of the Dataset

summary(data)

## Make Model Year Engine.Fuel.Type   
## Length:11914 Length:11914 Min. :1990 Length:11914   
## Class :character Class :character 1st Qu.:2007 Class :character   
## Mode :character Mode :character Median :2015 Mode :character   
## Mean :2010   
## 3rd Qu.:2016   
## Max. :2017   
##   
## Engine.HP Engine.Cylinders Transmission.Type Driven\_Wheels   
## Min. : 55.0 Min. : 0.000 Length:11914 Length:11914   
## 1st Qu.: 170.0 1st Qu.: 4.000 Class :character Class :character   
## Median : 227.0 Median : 6.000 Mode :character Mode :character   
## Mean : 249.4 Mean : 5.629   
## 3rd Qu.: 300.0 3rd Qu.: 6.000   
## Max. :1001.0 Max. :16.000   
## NA's :69 NA's :30   
## Number.of.Doors Market.Category Vehicle.Size Vehicle.Style   
## Min. :2.000 Length:11914 Length:11914 Length:11914   
## 1st Qu.:2.000 Class :character Class :character Class :character   
## Median :4.000 Mode :character Mode :character Mode :character   
## Mean :3.436   
## 3rd Qu.:4.000   
## Max. :4.000   
## NA's :6   
## highway.MPG city.mpg Popularity MSRP   
## Min. : 12.00 Min. : 7.00 Min. : 2 Min. : 2000   
## 1st Qu.: 22.00 1st Qu.: 16.00 1st Qu.: 549 1st Qu.: 21000   
## Median : 26.00 Median : 18.00 Median :1385 Median : 29995   
## Mean : 26.64 Mean : 19.73 Mean :1555 Mean : 40595   
## 3rd Qu.: 30.00 3rd Qu.: 22.00 3rd Qu.:2009 3rd Qu.: 42231   
## Max. :354.00 Max. :137.00 Max. :5657 Max. :2065902   
##

## Task 2 - Clean your dataset

Capture all columns that are character fields

cols <- names(data)[vapply(data, is.character, logical(1))]  
data[,cols] <- lapply(data[,cols],trimws)

Convert missing values to NAs

data[data=="N/A"] = NA

Use sapply(), which is like a for loop that goes through each column of the dataset and applys the function to it

sapply(data, function(x) mean(is.na(x)))

## Make Model Year Engine.Fuel.Type   
## 0.0000000000 0.0000000000 0.0000000000 0.0000000000   
## Engine.HP Engine.Cylinders Transmission.Type Driven\_Wheels   
## 0.0057915058 0.0025180460 0.0000000000 0.0000000000   
## Number.of.Doors Market.Category Vehicle.Size Vehicle.Style   
## 0.0005036092 0.3140842706 0.0000000000 0.0000000000   
## highway.MPG city.mpg Popularity MSRP   
## 0.0000000000 0.0000000000 0.0000000000 0.0000000000

It looks like Market.Category column has a high number of missing values (roughly 31.4%) It might be smart to remove this column from the dataset by running the code below

data$Market.Category <- NULL

Return only observations that have no missing values and preview the dataset

data <- data[complete.cases(data),]  
head(data)

## Make Model Year Engine.Fuel.Type Engine.HP Engine.Cylinders  
## 1 BMW 1 Series M 2011 premium unleaded (required) 335 6  
## 2 BMW 1 Series 2011 premium unleaded (required) 300 6  
## 3 BMW 1 Series 2011 premium unleaded (required) 300 6  
## 4 BMW 1 Series 2011 premium unleaded (required) 230 6  
## 5 BMW 1 Series 2011 premium unleaded (required) 230 6  
## 6 BMW 1 Series 2012 premium unleaded (required) 230 6  
## Transmission.Type Driven\_Wheels Number.of.Doors Vehicle.Size Vehicle.Style  
## 1 MANUAL rear wheel drive 2 Compact Coupe  
## 2 MANUAL rear wheel drive 2 Compact Convertible  
## 3 MANUAL rear wheel drive 2 Compact Coupe  
## 4 MANUAL rear wheel drive 2 Compact Coupe  
## 5 MANUAL rear wheel drive 2 Compact Convertible  
## 6 MANUAL rear wheel drive 2 Compact Coupe  
## highway.MPG city.mpg Popularity MSRP  
## 1 26 19 3916 46135  
## 2 28 19 3916 40650  
## 3 28 20 3916 36350  
## 4 28 18 3916 29450  
## 5 28 18 3916 34500  
## 6 28 18 3916 31200

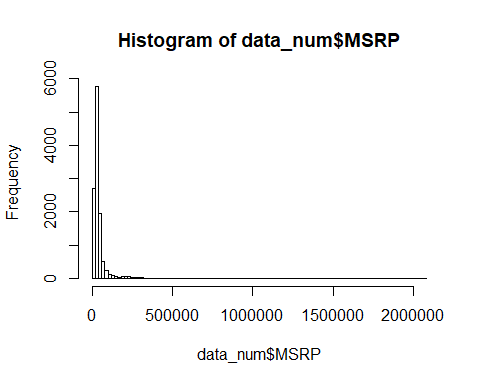
## Task 3 - Split into training and test set

Now we will only be selecting numeric columns from the dataset for our linear regression model

data\_num <- data %>% select\_if(is.numeric)

The target variable for our machine learning model is the price column (MSRP) Now we will create a histogram to see the MSRP distribution

hist(data\_num$MSRP, breaks=100)



This shows us that there are some outliers in our column as the majority of cars have the price in this region. these outliers can cause issues with the model, so we can filter the dataset to include cars with a price range of 15,000 and 50,000.

data\_num <- data\_num %>% filter(MSRP > 15000) %>% filter(MSRP < 50000)

Now we will split our dataset into a training and test set. To get consistent results, and to make sure the partitions are reproducable, the seed will need to be set to any integer. We will now select 80% of the dataset to training and remaining 20% will be the test dataset. To do this, we will get the number of rows that will account to 80%, and then use the floor() function to round up to the next integer.

set.seed(123)  
size <- floor(0.8 \* nrow(data\_num))

Now we will use the sample() function to randomly select 80% of rows from your dataset and store the row numbers.

train\_ind <- sample(seq\_len(nrow(data\_num)), size = size)

To get the training dataset set you can filter the dataset to include the row numbers, to get the test dataset you can filter the dataset to ignore the row numbers.

train <- data\_num[train\_ind, ]  
test <- data\_num[-train\_ind, ]

## Task 4 - Fit linear regression model and interpret model summary statistics

A linear regression model is a model that assumes a linear relationship between the predictors and the response variable.

This means that the response variable can be calculated from a linear combination of predictors.

In this dataset the the response variable is the MSRP column, while the remaininag columns are the predictors.

The goals is to build a model to predict the MSRP column, by using the characteristcs such as Engine, HP, number of doors, etc.

To build the linear regression model we will use the lm() function and focus on the model equation and the dataset to be used.

The model equations can be returned in column names while the dataset we will be using will be the training dataset.

model <- lm(MSRP ~ .,data = train)  
summary(model)

##   
## Call:  
## lm(formula = MSRP ~ ., data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -18494.4 -3819.5 -717.9 3407.0 20580.3   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.784e+05 4.012e+04 -6.940 4.32e-12 \*\*\*  
## Year 1.430e+02 2.006e+01 7.128 1.13e-12 \*\*\*  
## Engine.HP 1.121e+02 1.779e+00 63.006 < 2e-16 \*\*\*  
## Engine.Cylinders -1.144e+03 9.981e+01 -11.461 < 2e-16 \*\*\*  
## Number.of.Doors 3.567e+02 9.030e+01 3.950 7.91e-05 \*\*\*  
## highway.MPG -9.825e+01 2.898e+01 -3.390 0.000703 \*\*\*  
## city.mpg 1.638e+02 2.544e+01 6.440 1.29e-10 \*\*\*  
## Popularity -2.279e-01 4.962e-02 -4.593 4.45e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5538 on 6384 degrees of freedom  
## Multiple R-squared: 0.5929, Adjusted R-squared: 0.5925   
## F-statistic: 1328 on 7 and 6384 DF, p-value: < 2.2e-16

In the Coefficients we can see the statistical significance. Predictive variables that are significantly associated with the outcome variables are marked with stars.

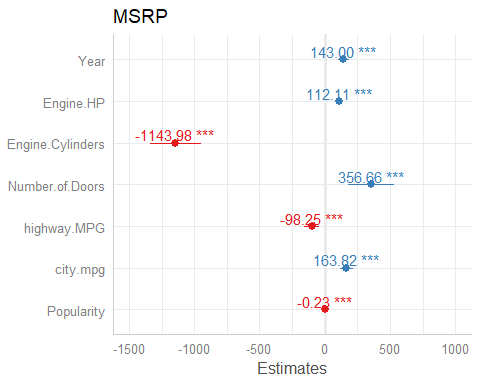
The higher the amount of stars, the more significant the predictors are. For a given predictor value, the coefficient (which is called the estimate) can be interepted as the average effect on the response varable of one unit increase in predictor given that all other predictors are fixed.

For example, if the engine HP increases by 1 unit and all other predictors are kept constant, then the price of the car will increase by 111 units.

The Residual Standard Error, Mult R-Squared, and F-Statistics are metrics that are used to check how well the model fits your data.

Residual Standard Error - corresponds to the prediction error in the training set and represents roughly the average difference between the observed values and the predicted values of teh model.  
In this model the Residual Standard Error is 5495. That mean on average you can expect a diviation of 5495 in the price prediction. The R-Squared rangeds from 0 to 1 and represents the proportion of teh variation and the response variable that can be explained by the model predictor variables.  
The higher the R-Squared value, the better the model is. However, a problem with the R squared is that it will always increase as more predictors are added to teh model. even if the predictors are only weakly associated with the outcome of the respnose variable. A solution is to adjust the R squared value by taking into the account the number of predictive variables. the adjestment in the adjusteed R-squared value in the summary output is the correction for the number of predictive variables included in the model. So you mainly consider the adjusted R squared value your value it .59 (which is good). F-statistic gives the overall significance of the model. it assess whether at least one predict value has a non-zero coefficient. the p value of less than 2.22-16 shows that the model is highly siginificant

plot\_model(model, show.values = TRUE, value.offset = 0.2)



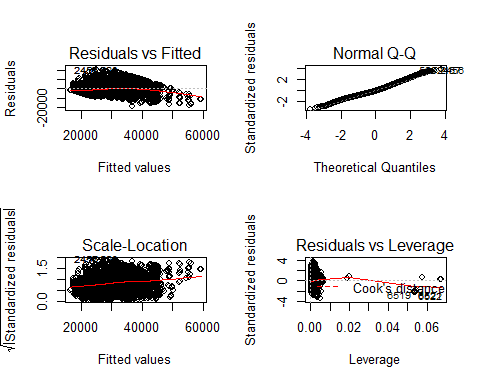
this plot shows the coefficient and significance value now you can build a linear regression model by sepcifying the predictors that you want. for examples you might want to include only 3 predictors instead all of them in you dataset

model2 <- lm(MSRP ~ Engine.HP + highway.MPG + Engine.Cylinders, data = train)

## Task 5 - Plot and analyse model residuals

residuals can show how poorly a matter represents data. They are left over values of teh response variable after fitting a model to data, and they can reveil unexplained patterns in the data by the fitted model. Using this information, not only could we check if linear regression assumptions are met, but you could improve your model as well.

par(mfrow=c(2,2))  
plot(model)

 this wil set the plotting pane to include 4 plots( 2 rows and 2 columns)

R vs Fitted show if residuals have nnn-linear patterns. fitted values are on the x axis and the residuals (which are how far the fitted values are from the observed values)are on the y axis. there could be a non-linear relationship between predictive variables and the response variables and teh pattern could show up in this plot. if the model does not capture the non-linear relationship. if you find equally residuals around the horizontal line, without any distinct patterns, that is a good indication that you dont have non-linear relationships. in this plot we do not see any any distinctive patterns. Normal QQ plot which shows if residuals are naormally distributed. Do The residuals follow a straight line? or do they deviate severly? It is good when residuals are lined well on the straight line. but in reality you will see some deviations. In this plot we do not se e much deviation until towards the end where some data points are deiviated.

The scale location plot show if residualdes are diparred equally along the ranges of predictors. this is how you can check the assumption of equal variance. It is good if you see a horizonalt line withh equally diparred points. in this model the residuals appear to be randomlly spread.

residuals vs leverage plot helps you to find influential cases in your dataset.  
These cases could be extremem cases against the regression line, and could alter results if you exclude them from you model. In this graph pattens are not relevant. You should watch out for outlier values in the upper right and lower right corners. those parts are the places where the cases can be influential against the regression line. look for cases outside of the dash line (the cook distance). cases that are outside the cook distance (meaning they are high cook distance codes) cases are influental to teh regression results. in our model we can see that observation 6519 and 6522 are far beyond the cook distance lines. there are influential cases and will alter your model if you remove them. )

the 4 plots show potential problematic cases, with the row number of the data in your dataset. if some cases are noted in each of the plots, you might wan tto take a closer look at them. Is there anything sepcial with those points? or was there an error in teh data entry? you can go back to the building step and you can try include/excluding predictors and see if the diagnostic plots improve

## Task 6 - Predict future values and calculate model error metrics

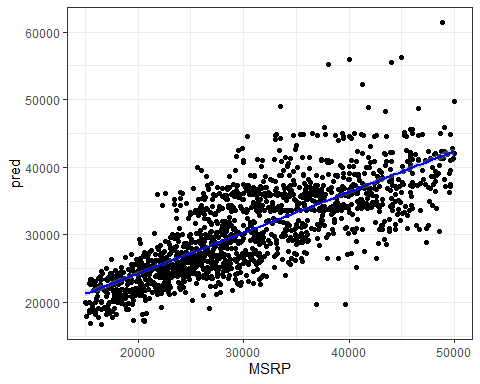
you will predict the MSRP value of the test dataset and compare it to the with the observed MSRP values. you can youse the predict functions with parameters model and test data.

test$pred <- predict(model, newdata = test)

now you can plot the predicted and observed values

par(mfrow=c(1,1))  
ggplot(test, aes(x=MSRP, y=pred))+  
 geom\_point()+  
 geom\_smooth(method="lm", color="blue")+  
 theme\_bw()

## `geom\_smooth()` using formula 'y ~ x'



on teh x axis you can see the observered MSRP values and on the Y you can see the predictive values. Now we will calculate the error metrics of the linear model.. we will first find the error. which is observed value subtraced from the respective predicted value.

error <- test$pred - test$MSRP

we will calc 2 error metrics RMSE is a good measure to see how accurate the model predicts a response. this is a good test for fit, if the main purpose of the model is prediction.

rmse <- sqrt(mean(error^2))  
rmse

## [1] 5371.526

the rmse value is 5546, which is fine since the range of the rsme value is between 15k and 50k the second metric is MAE (mean absolute error). measures the avg magnitue of the errors in your predictions predictions without considering their direction

mae <- mean(abs(error))  
mae

## [1] 4268.753

the mae is 4401. this means that on avg you would expect an error magnitude of 4401 in your predictions. This error can be either positive or negative.

in rmse, since the errors are squared before they are averaged, the rmse give a relativly high weight to large errors. this means the rse should be more usedful when large errors a particularly undesirable. but from an interpretataion stnadpoint mean absolte error is better.