PREDICTIVE ANALYSIS OF U.S. HOUSE PRICING

A PROJECT REPORT

Submitted by

AGNIVA KONAR

UPASYA BOSE

TIYASHA SAMANTA

Supervised by

MR. BRATIN DAS

In partial fulfillment for the award of the degree of

MASTER OF SCIENCE

IN

APPLIED STATISTICS AND ANALYTICS

IN YEAR 2020-2021

MAULANA ABUL KALAM AZAD
UNIVERSITY OF TECHNOLOGY,
WEST BENGAL

Utech

In Pursuit Of Knowledge And Excellence

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

NH-12 (Old NH-34) Simhat

Haringhata, Nadia 741249, West Bengal

BONAFIDE CERTIFICATE

Certified that this project report **“PREDICTIVE ANALYSIS OF U.S. HOUSE PRICING”** is the bonafide work of AGNIVA KONAR, UPASYA BOSE & TIYASHA SAMANTA who carried out the project work under my supervision.

____

SUPERVISOR

____

HOD, Department of Applied Science

SONASS

____

External Examiner

ACKNOWLEDGEMENT

It is a great pleasure for us to undertake this project. We are utterly grateful to our project guide Mr. Bratin Das. This project would not have been completed without his enormous help and worthy experience. Whenever we were in a fix, he was there to guide us and provide us with a perfect solution. Although, this project report has been prepared with utmost care and deep routed interest, even then, we accept our flaws and imperfection.

Date:

TABLE OF CONTENTS

SUMMARY/ABSTRACT:-

The given data consisted of numerous factors that can affect the price of a house in US and also the sale price of the particular house. It can be seen that the given data at hand is a multivariate data. The main objective is to predict the sale price of a house in US using the given data. For that, it is required to plot a regression equation using the Sale price as the response variable and the other factors as the predictor or the independent variables. At first, the non-numeric data should be converted into categorical data for ease of calculation and then the data needs to be cleansed i.e., the missing values should be replaced and the non-numeric columns containing too many missing data must be dropped. Some tests should be carried out to extract some useful conclusions about the data and also to drop some insignificant predictor variables. At last, when these necessary steps are done, the final regression equation needs to be plotted using sale price as the response variable and the significant predictor variables as the covariates. With help of that regression model, the necessary predictions can be made.

KEYWORDS:-

1. Multiple Linear Regression Equation
2. Multivariate Data
3. Data Conversion
4. Data Cleansing
5. Outliers
6. Influential Observations
7. High Leverage Values
8. Autocorrelation
9. Homoscedasticity
10. Multicollinearity

INTRODUCTION:-

Regression Analysis is a concept which is very beneficial while deciding the relation between a response variable (or and a dependent variable) and other predictor variables (or independent variables). It has mainly three types of variations, and those are, Linear Regression, Multiple Linear Regression and Non Linear Regression.

In linear regression, there are some basic assumptions which are as follows:-

a) The response and the predictor variables show a linear relationship between the slope and the intercept.
b) The value of the error is not correlated with all other observations.
c) The error values follow a normal distribution.
d) The variance of the errors is constant across all observation (homoscedasticity).

In this case, we are dealing with a multivariate data and hence in this case, a multiple linear regression is a valid option. One thing that is most important about a multiple linear regression model is that, the independent variables should show a minimum correlation with each other. High correlation among the independent variables will indicate insignificancy.

Using the multiple linear regression model, we can make prediction about the response variable. Lesser the deviation of the prediction from the original value, better the prediction is and more accurate fitting of the linear model is.

AIM:

To make a prediction of Sale Price of houses in US using a regression model. Since the Sale Price needs to be predicted, hence, this becomes our response variable for the regression model whereas the other variables are our respective predictor variables.

OBJECTIVES:

a) The primary data at hand contains non numeric columns which must be converted to categorical data for ease of calculation.

b) The primary data may contain missing numeric values. Those values must be replaced.

c) Some columns may not contain appropriate data. Such columns may lead to insignificant conclusions if kept. Hence, those columns need to be dropped.

d) Some tests need to be carried out for drawing proper conclusions about the data, such as, Test for Autocorrelation and Test for Homoscedasticity.

e) Test for Multicollinearity needs to be done as well to find out which covariates have high correlation among themselves and then those covariates need to be dropped to ensure a better prediction model.

f) Then using the significant covariates, we need to plot a regression model and make prediction about the Sale Price.

SCOPE AND NEED OF THE PROJECT:

The project basically provides an algorithm to predict the price of a house in US. The prediction may not be an accurate one, but it can give a brief idea about the price range. If anyone is interested in buying or selling a plot, they can use this algorithm and plan their budget accordingly. This algorithm is purely based on statistical techniques and it can be put to a good use.

HYPOTHESIS TESTING:

a) TEST FOR AUTOCORRELATION:

Autocorrelation is another name of serial correlation which is the degree of correlation between the values of variables across different data sets. We usually use this process when we work with time series data where the observations occur at different time points. For example, wind speed measured on different days of the week. If the wind speed values measured that occurred closer in time are more similar than the values that occurred farther apart in time, the data is said to be correlated.

Durbin Watson Statistic

The test statistic is used to detect autocorrelation in the residuals from a regression analysis is known as Durbin Watson statistic. It is named after a British statistician and econometrician professor James Durbin and an Australian statistician Geoffrey Stuart Watson.

The formula for the test is:

$$DW = \frac{\Sigma_{t=2}^{T}((e_t - e_{t-1})^2)}{\Sigma_{t=1}^{T} e_t^2}$$

Where:

- et is the residual figure
- T is the number of observations of the experiment.

Hypothesis and Assumptions for the test

The hypotheses followed for the Durbin Watson statistic:

H_0 = First-order autocorrelation does not exist.

vs

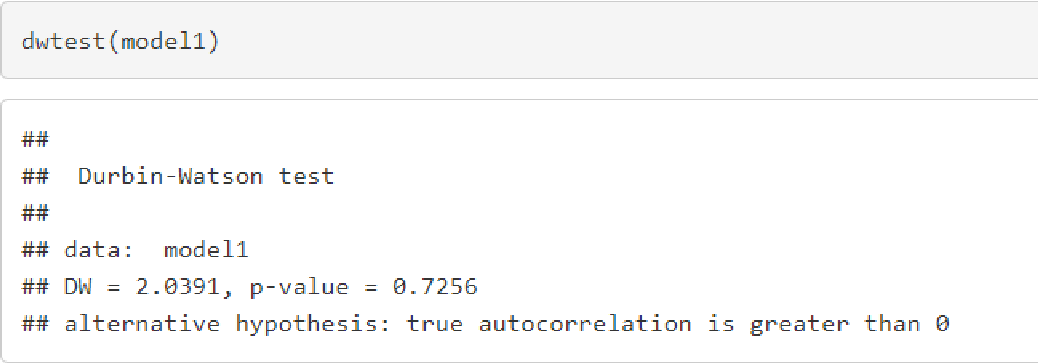
H_1= First-order autocorrelation exists.

The assumptions of the test are:

- Errors are normally distributed with a mean value of 0
- All errors are stationary.

Procedure & Interpretation of the Durban Watson Test Statistic

We use the dwtest() in RStudio to perform this test. The Durbin Watson statistic will always assume a value between 0 and 4. A value of DW = 2 indicates that there is no autocorrelation. When the value is below 2, it indicates a positive autocorrelation, and a value higher than 2 indicates a negative autocorrelation.



```
dwtest(model1)
```

```
##
##  Durbin-Watson test
##
## data:  model1
## DW = 2.0391, p-value = 0.7256
## alternative hypothesis: true autocorrelation is greater than 0
```

Conclusion of the test

Since p–value = 0.7256 > level of significance = 0.05. Hence, we accept the null hypothesis H_0 which states that "First-order autocorrelation does not exist". Data indicates population autocorrelation equals to zero (0). In other words, there is no autocorrelation since DW statistic is approximately equals to two (2).

b) TEST FOR HOMOSCEDASTICITY:

Homoscedasticity is a phenomenon which arises when all the variables have the same finite variance. The assumption of homoscedasticity is that, every variance error in a regression model is constant. The error term does not vary much as the value of the predictor variable changes. A scatterplot of residuals versus predicted values is good way to check for homoscedasticity. Homoscedasticity test is one of the most important test for fitting good linear regression model. A homoscedastic data makes it easier to plot the regression model and work with it. The lack of homoscedasticity may suggest that the regression model may need to include additional predictor variables to explain the performance of the response variable.

Way of testing for homoscedasticity:

As said earlier, homoscedasticity can be tested by plotting a scatterplot of residuals against the predicted values. But, here, we are going to use the Breusch-Pagan Test to check for homoscedasticity. This test was developed by Trevor Breusch and Adrian Pagan in 1979. It is used for checking heteroscedasticity in a linear regression model. Absence of heteroscedasticity indicates presence of homoscedasticity in the model. It mainly tests whether the variance of the errors from a regression is dependent on the values of the predictor variables. If this happens, then it strongly suggests presence of heteroscedasticity, which means, absence of homoscedasticity.

Hypothesis for the Breusch-Pagan test:

The null hypothesis, denoted by H_0 and the alternative hypothesis, denoted by H_1 is given as follows:

H_0: “The given model is homoscedastic”

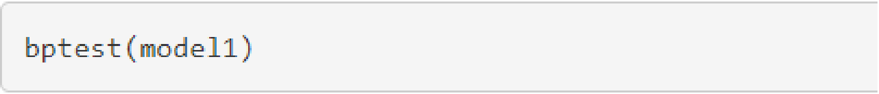
vs

H_1: “The given model is heteroscedastic”

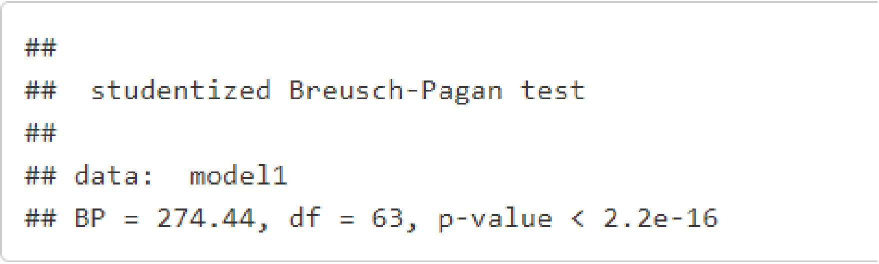
Procedure and Interpretation of the test:

A function, named as bptest(), stored in the lmtest package is called and the variable, in which the linear model is stored is sent as the parameter. The function gives a certain p value corresponding to the test as well as a value of the test statistic.

If the p value of the test is less than 0.05, which is the level of significance, then our null hypothesis H_0 is rejected, else, our H_0 is accepted.



```
bptest(model1)
```



```
##
##  studentized Breusch-Pagan test
##
## data:  model1
## BP = 274.44, df = 63, p-value < 2.2e-16
```

Conclusion of the test:

Here we can see, that the p value of our test is less than 0.05, which is the level of significance. Hence we can conclude that, our null hypothesis H_0

is rejected. That means, our model isn’t homoscedastic, which implies that our linear model is heteroscedastic. Hence, we have to implement the technique of Generalized Least Squares to solve this issue of heteroscedasticity.

c) TEST FOR MULTICOLLINEARITY:

Multicollinearity is a situation which occurs when the predictor variables in a multiple linear regression model are highly correlated to each other. In other words, it basically refers to an incident where an independent variable of a regression model can be linearly predicted using other covariates. Presence of multicollinearity in a data isn’t a good sign as it will affect the statistical inference made from the data. Those inference won’t be reliable enough. It also lowers the accuracy of the estimate of coefficients of our model, and hence it becomes difficult to find out the significant factors from their corresponding p values. However, these problems arise when the multicollinearity is very high. If the data shows moderate multicollinearity, then there’s no need to remove/ reduce it.

Way of testing multicollinearity:-

One very popular way of testing for multicollinearity is by finding the Variance Inflation Factor (VIF). Using this, we can find out which predictor variables are affected by multicollinearity and the strength of correlation and hence we can act accordingly. VIF has a lower limit of 1 but has no upper limit. For calculating VIF, we need to plot a regression model using a predictor variable as the response and the rest of the predictor variables as the covariates. Then we need to find the coefficient of determination of the following regression model. Then VIF is calculated as:-

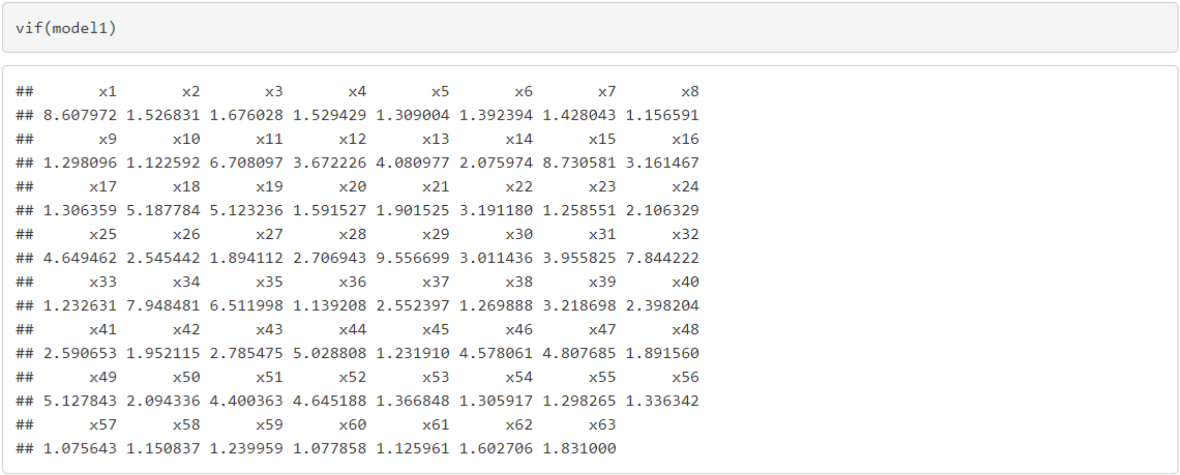
$$\mathrm{VIF}_i = \frac{1}{1 - R_i^2} = \frac{1}{\mathrm{Tolerance}}$$

If the VIF has a value of 1, then it means that there is no multicollinearity, or in simpler terms, there is no correlation present between that independent variable and any others. Hence, the corresponding predictor variable doesn’t need to be dropped.

If the VIF value lies between 1 and 10, then it suggests the presence of moderate multicollinearity. Hence, even in this case, the corresponding predictor variable doesn't need to be dropped.

But, if the VIF value comes out as greater than 10, then that indicates high multicollinearity and that will lead to erroneous results and hence that predictor variable needs to be dropped.

We carry out the test of multicollinearity using the vif() in RStudio. The result is given as:-



```
vif(model1)
```

```
##       x1       x2       x3       x4       x5       x6       x7       x8
## 8.607972 1.526831 1.676028 1.529429 1.309004 1.392394 1.428043 1.156591
##       x9      x10      x11      x12      x13      x14      x15      x16
## 1.298096 1.122592 6.708097 3.672226 4.080977 2.075974 8.730581 3.161467
##      x17      x18      x19      x20      x21      x22      x23      x24
## 1.306359 5.187784 5.123236 1.591527 1.901525 3.191180 1.258551 2.106329
##      x25      x26      x27      x28      x29      x30      x31      x32
## 4.649462 2.545442 1.894112 2.706943 9.556699 3.011436 3.955825 7.844222
##      x33      x34      x35      x36      x37      x38      x39      x40
## 1.232631 7.948481 6.511998 1.139208 2.552397 1.269888 3.218698 2.398204
##      x41      x42      x43      x44      x45      x46      x47      x48
## 2.590653 1.952115 2.785475 5.028808 1.231910 4.578061 4.807685 1.891560
##      x49      x50      x51      x52      x53      x54      x55      x56
## 5.127843 2.094336 4.400363 4.645188 1.366848 1.305917 1.298265 1.336342
##      x57      x58      x59      x60      x61      x62      x63
## 1.075643 1.150837 1.239959 1.077858 1.125961 1.602706 1.831000
```

Conclusion of the test:

We can observe, that the vif values for each covariate is lesser than 10. Hence, we can conclude that there is only presence of moderate muticollinearity in our dataset and thus, we don't have to drop any covariates.

Once these tests are done, we have all the required information about our model and we can finally proceed to building our predictive model. After successfully creating the predictive linear model, we use it to make our predictions using the testing dataset.

RESEARCH METHODOLOGY:

We have obtained our data from kaggle.com. The data is in form of an Excel spreadsheet containing 1460 rows of data containing values corresponding to the predictor variables and response variable Sale Price. The data is a multivariate data. Before plotting our multiple linear regression model, the given data at hand must be converted, as it contained some non-numeric columns of data. The data also contained some missing numeric values whereas some columns of data contained huge amount of inappropriate data. Thus, cleansing of data was also necessary. The process of data conversion and data cleansing are discussed in details below.

DATA CONVERSION:-

As said earlier, the non-numeric columns must be converted to categorical data by using the as.factor() function in RStudio and by using the IF and VLOOKUP commands in MS Excel.

Here we have done this is MS Excel. Following the data description, we have assigned a numeric value for each type of non-numeric data. This process is carried out in MS Excel using the above mentioned commands.

IF command:-

The general syntax for the IF command in Excel is given by:-

```
=IF(logical_test, value_if_true, value_if_false)
```

In case of a nested IF structure, we introduce another IF statement in place of the 3rd parameter “value_if_false”.

This command goes through an array of data and assigns a different value to that particular non-numeric data.

For ex:-

```
=IF(G2="Reg",0,IF(G2="IR1",1,IF(G2="IR2",2,3)))
```

Here, if G2=Reg, then it assigns the value 0, else if, G2=IR1, then it assigns the value 1,else if, G2=IR2, then it assigns the value 2, or else, if G2 contains some different value, then it assigns the value 3, hence converting it into a numeric data.

VLOOKUP command:-

The general syntax for the VLOOKUP command in Excel is given by:-

```
=VLOOKUP(look_up_val, table_array, col_index_num, logic_parameter)
```

First, we have to create a VLOOKUP table containing the given data that needs to be changed and the data it needs to be converted into. Then this command will go through the VLOOKUP table and assign the numeric values corresponding to each non-numeric data.

For ex:-

VLOOKUP Table

Reg	0
IR1	1
IR2	2
IR3	3

```
=VLOOKUP(G2,$E$6:$F$9,2,FALSE)
```

This command looks up in the VLOOKUP table for the value corresponding to the data occurred at G2 and assigns the respective value, hence converting it into a numeric data. Hence, in this way, we convert our entire non-numeric data into categorical data.

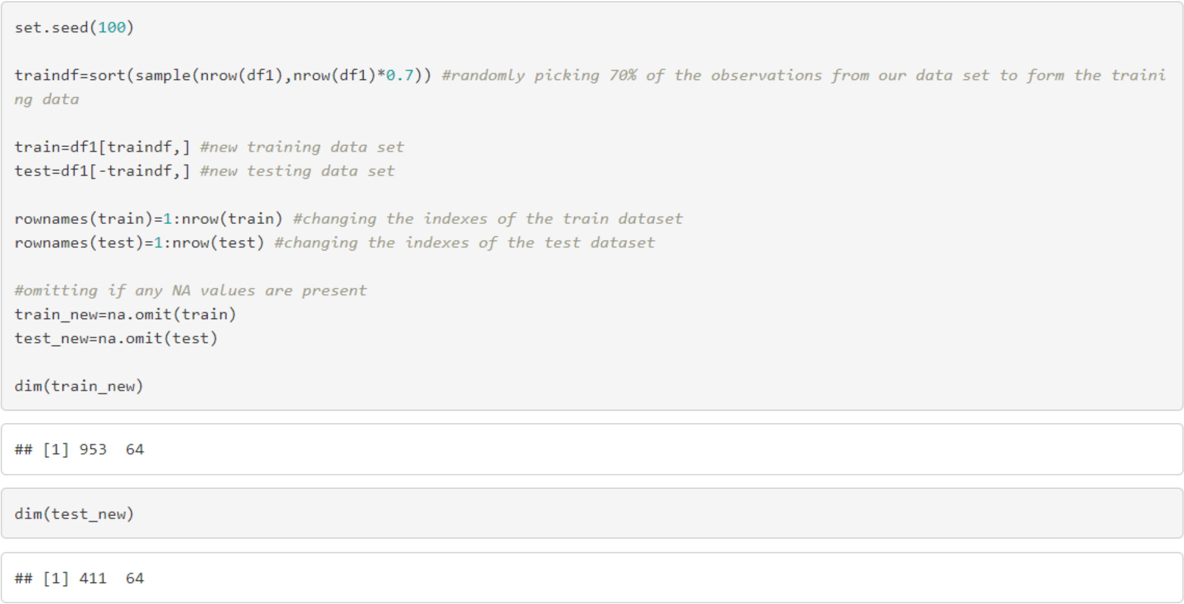
SPLITTING OF DATASET:

Before moving on to the data cleansing, we first have to split the dataset into training and testing dataset. This splitting must be done randomly.

We split the dataset in the ratio of 70:30. The 70% of the dataset is named as the training dataset whereas the remaining 30% is named as the testing dataset.

All kinds of data cleansing must be done only on the training dataset, because this is the data that we are going to feed the model for the sake of it’s learning. Once the model is ready, then we will perform the prediction on the basis of testing dataset, to look how well it is performing when it faces an unknown dataset.

One interesting observation can be made, and that is, how the model is handling when it faces an outlier, or an influential observation, or a high leverage value of the testing dataset.



```
set.seed(100)

traindf=sort(sample(nrow(df1),nrow(df1)*0.7)) #randomly picking 70% of the observations from our data set to form the traini
ng data

train=df1[traindf,] #new training data set
test=df1[-traindf,] #new testing data set

rownames(train)=1:nrow(train) #changing the indexes of the train dataset
rownames(test)=1:nrow(test) #changing the indexes of the test dataset

#omitting if any NA values are present
train_new=na.omit(train)
test_new=na.omit(test)

dim(train_new)
```

```
## [1] 953  64
```

```
dim(test_new)
```

```
## [1] 411  64
```

DATA CLEANSING:

Data Cleansing is performed on the training dataset and it is mainly done in four ways:

Step 1: Removing the NA values from the dataset:

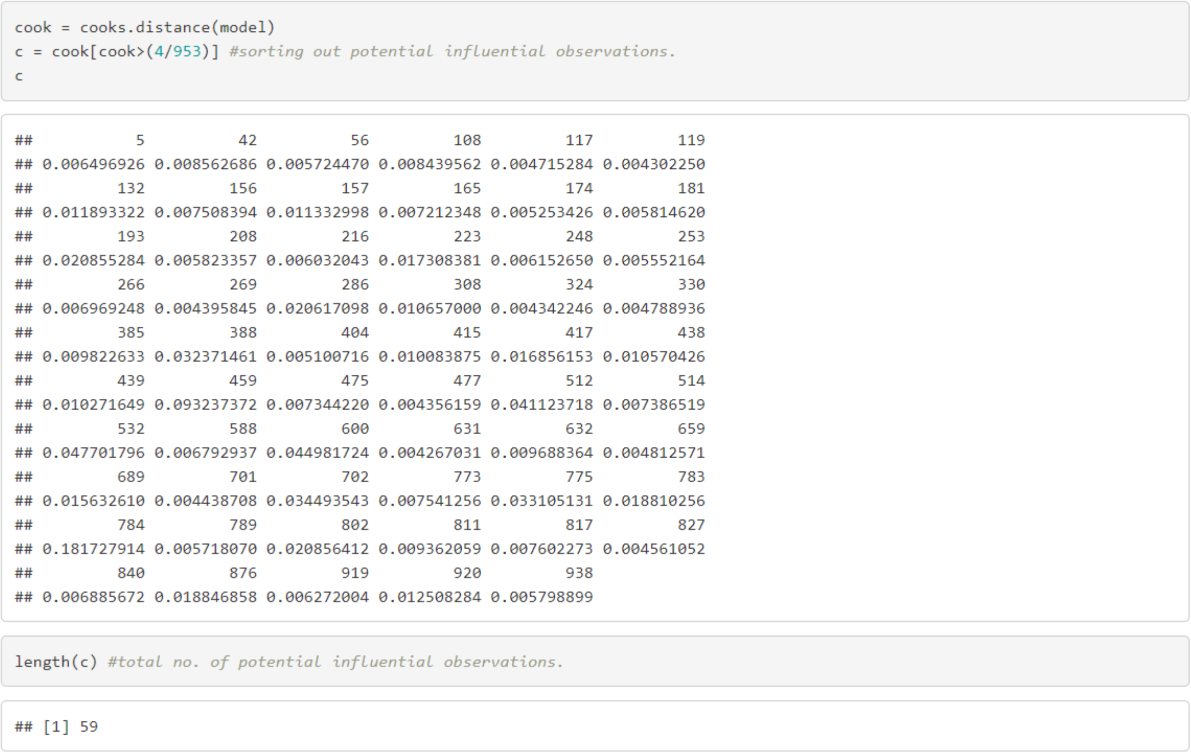
This can be done using the na.omit() in RStudio, passing the dataset as a parameter to this function or this can be done manually in MS Excel as well. In this case, we have done it in RStudio because the latter one is very time consuming.

Step 2: Detection of Influential Observations using Cook’s Distance:

These are data anomalies and they change the regression outcome mainly by changing the slope of a regression model i.e., the regression line will be shifted because of the presence of influential observations in the dataset.

We use the cooks.distance() to find these observations.

If Cook’s distance for i^{th} observation is greater than 4/n, where n is the no. of observation in the dataset, then the i^{th} observation can be reared as potential influential observation.



```
cook = cooks.distance(model)
c = cook[cook>(4/953)] #sorting out potential influential observations.
c
```

```
##           5          42          56         108         117         119
## 0.006496926 0.008562686 0.005724470 0.008439562 0.004715284 0.004302250
##         132         156         157         165         174         181
## 0.011893322 0.007508394 0.011332998 0.007212348 0.005253426 0.005814620
##         193         208         216         223         248         253
## 0.020855284 0.005823357 0.006032043 0.017308381 0.006152650 0.005552164
##         266         269         286         308         324         330
## 0.006969248 0.004395845 0.020617098 0.010657000 0.004342246 0.004788936
##         385         388         404         415         417         438
## 0.009822633 0.032371461 0.005100716 0.010083875 0.016856153 0.010570426
##         439         459         475         477         512         514
## 0.010271649 0.093237372 0.007344220 0.004356159 0.041123718 0.007386519
##         532         588         600         631         632         659
## 0.047701796 0.006792937 0.044981724 0.004267031 0.009688364 0.004812571
##         689         701         702         773         775         783
## 0.015632610 0.004438708 0.034493543 0.007541256 0.033105131 0.018810256
##         784         789         802         811         817         827
## 0.181727914 0.005718070 0.020856412 0.009362059 0.007602273 0.004561052
##         840         876         919         920         938
## 0.006885672 0.018846858 0.006272004 0.012508284 0.005798899
```

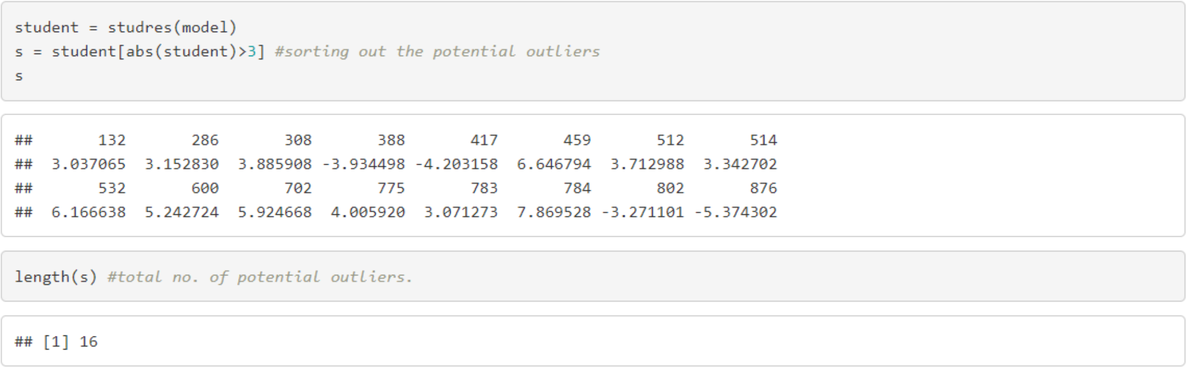
```
length(c) #total no. of potential influential observations.
```

```
## [1] 59
```

Step 3: Detection of Outliers using Studentized Residuals:

Once again, these are also data impurities and presence of this affects the parameter estimate of our model as the parameters are highly non robust. These can be recognized with the help of the concept of Studentized residuals.

We use the studres() in RStudio to detect the outliers. If the absolute value of studentized residuals in greater than 3, then that observation can be regarded as a potential outlier.



```
student = studres(model)
s = student[abs(student)>3] #sorting out the potential outliers
s
```

```
##       132       286       308       388       417       459       512       514 
##  3.037065  3.152830  3.885908 -3.934498 -4.203158  6.646794  3.712988  3.342702 
##       532       600       702       775       783       784       802       876 
##  6.166638  5.242724  5.924668  4.005920  3.071273  7.869528 -3.271101 -5.374302
```

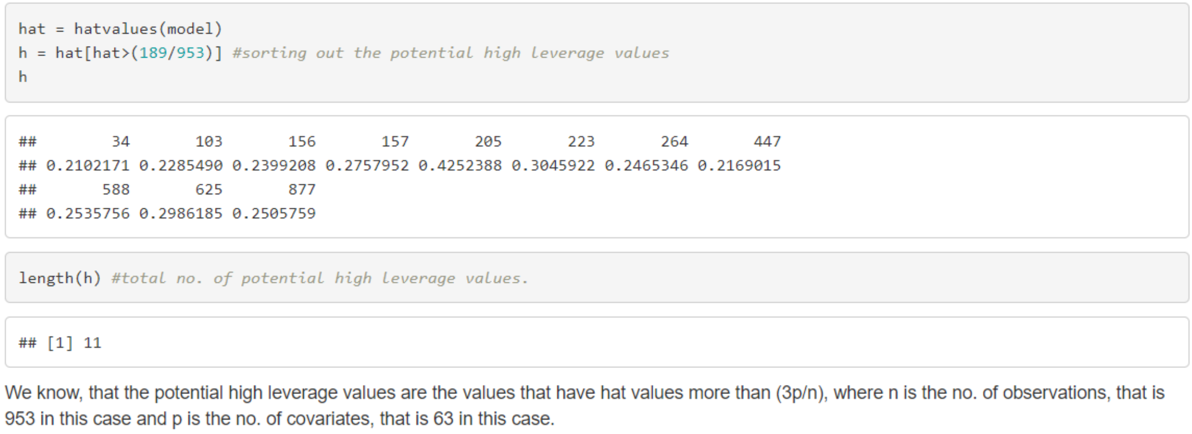
```
length(s) #total no. of potential outliers.
```

```
## [1] 16
```

Step 4: Detection of High Leverage Values using Hat matrix and hat values:

A data point for which the predictor value is unusual is called a high leverage value or a leverage point. These values can cause an impact to the parameter estimates. In fact, statistical significance of model parameter gets reduced in presence of a high leverage value. The best course to deal with such values is to delete/remove them from the analysis.

We use the concept of Hat matrix and hat values to recognize such values.



```
hat = hatvalues(model)
h = hat[hat>(189/953)] #sorting out the potential high leverage values
h
```

```
##        34       103       156       157       205       223       264       447 
## 0.2102171 0.2285490 0.2399208 0.2757952 0.4252388 0.3045922 0.2465346 0.2169015 
##       588       625       877 
## 0.2535756 0.2986185 0.2505759
```

```
length(h) #total no. of potential high leverage values.
```

```
## [1] 11
```

We know, that the potential high leverage values are the values that have hat values more than (3p/n), where n is the no. of observations, that is 953 in this case and p is the no. of covariates, that is 63 in this case.

The ith observation of the covariates is considered as a high leverage if its corresponding hat value is greater than 3p/n, where p is the no. of covariates and n is the no. of observations n the dataset.

We have to keep one thing in mind and that is, data is a very precious tool when it comes to analysis and losing huge amount of data for the sake of data cleansing can prove to be disadvantageous. Hence, we should remove these data anomalies tactically.

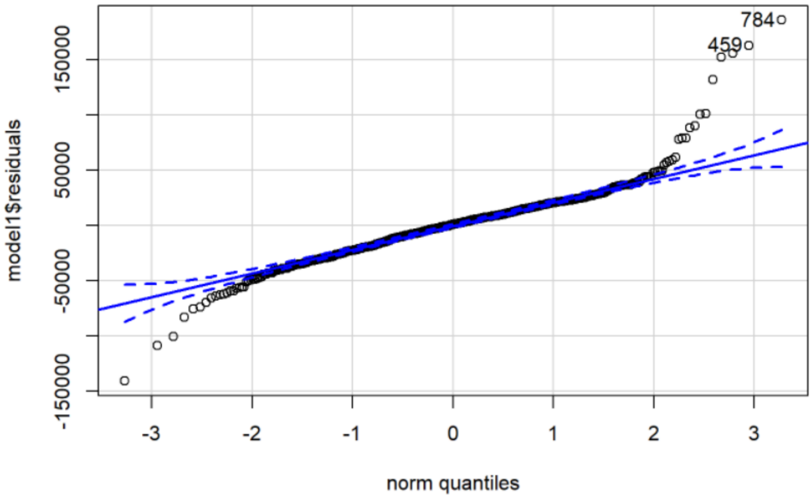
PROCEDURE:

Once we have completed all the steps mentioned in the research methodology section, we have our refined and cleansed dataset which can be used for primary model building.

The first thing that is very important in linear regression analysis is to check whether our model satisfies the condition of normality or not. For this, we use the QQplot to draw conclusions.

The qqPlot() function stored in the car package in RStudio helps us to plot this.

```
qqPlot(model1$residuals)
```



```
## 784 459
## 758 445
```

We can see that the quantiles of our cleansed data fits moderately with those of a normal distribution. Thus we can conclude that our normality assumption is satisfied.

After doing this, we need to check for presence of autocorrelation and multicollinearity and we also need to test for homoscedasticity as discussed above. Once, our model satisfies all these criterions, we have our final predictive model at hand and we can use that to make predictions.

We can also find the significant covariates by using the summary() and the covariates having p value lesser than 0.05 are considered to be significant.

In our case, the significant covariates are:

LotFrontage, LotArea, Alley, Housestyle, OverallQual, OverallCond, YearBuilt, RoofStyle, Exterior1st, MasVnrType, MasVnrArea, ExterQual, BsmtCond, BsmtExposure, BsmtFinSF1, BsmtFinSF2, BsmtUnfSF, 1stFlrSF, 2ndFlrSF, BedroomAbvGr, KitchenAbvGr, KitchenQual, Functional, FireplaceQU, GarageType, SaleCondition.

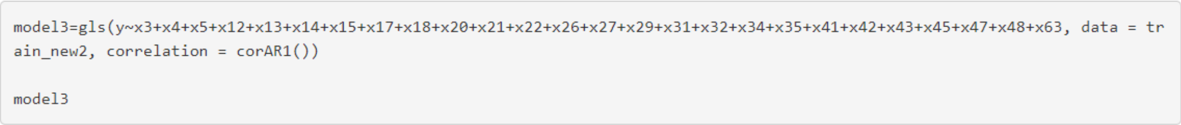
We will also compare our predictions with the response variables of our testing dataset to make sure how efficient our model and predictions are.

OUTCOMES:

To remove the issue of heteroscedasticity, we use the concept of Generalised Least Squares which helps us fulfilling the criteria of homoscedasticity, i.e., the variance of all the error components of a model are the same.

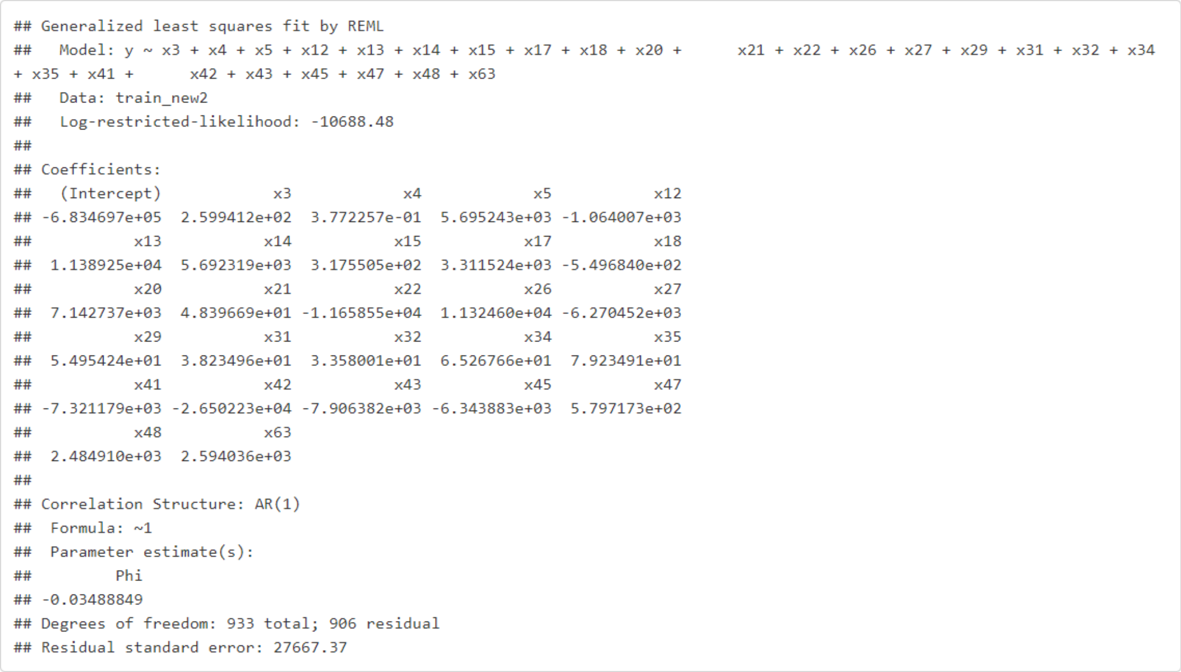
We use the gls() to implement this technique in our model.

Our final predictive model looks like this:



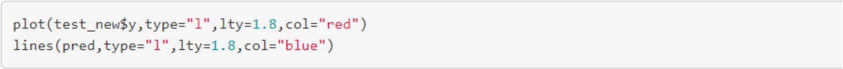
```
model3=gls(y~x3+x4+x5+x12+x13+x14+x15+x17+x18+x20+x21+x22+x26+x27+x29+x31+x32+x34+x35+x41+x42+x43+x45+x47+x48+x63, data = tr
ain_new2, correlation = corAR1())

model3
```

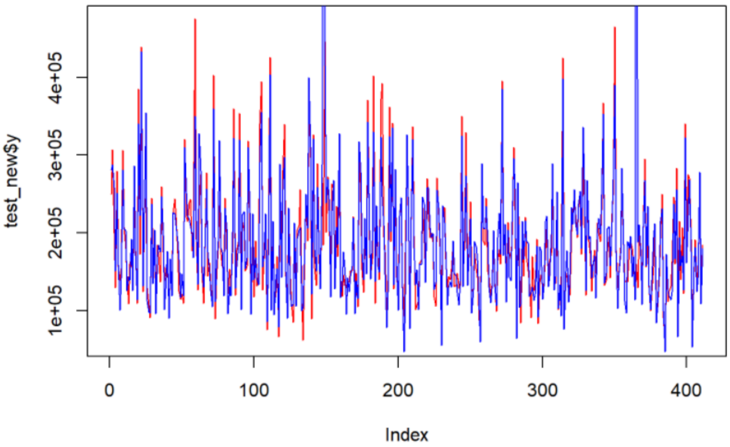


```
## Generalized least squares fit by REML
##   Model: y ~ x3 + x4 + x5 + x12 + x13 + x14 + x15 + x17 + x18 + x20 +      x21 + x22 + x26 + x27 + x29 + x31 + x32 + x34
+ x35 + x41 +      x42 + x43 + x45 + x47 + x48 + x63
##   Data: train_new2
##   Log-restricted-likelihood: -10688.48
##
## Coefficients:
##   (Intercept)            x3            x4            x5           x12
## -6.834697e+05  2.599412e+02  3.772257e-01  5.695243e+03 -1.064007e+03
##           x13           x14           x15           x17           x18
##  1.138925e+04  5.692319e+03  3.175505e+02  3.311524e+03 -5.496840e+02
##           x20           x21           x22           x26           x27
##  7.142737e+03  4.839669e+01 -1.165855e+04  1.132460e+04 -6.270452e+03
##           x29           x31           x32           x34           x35
##  5.495424e+01  3.823496e+01  3.358001e+01  6.526766e+01  7.923491e+01
##           x41           x42           x43           x45           x47
## -7.321179e+03 -2.650223e+04 -7.906382e+03 -6.343883e+03  5.797173e+02
##           x48           x63
##  2.484910e+03  2.594036e+03
##
## Correlation Structure: AR(1)
##  Formula: ~1
##  Parameter estimate(s):
##         Phi
## -0.03488849
## Degrees of freedom: 933 total; 906 residual
## Residual standard error: 27667.37
```

Using this model, we perform our predictions and compare it with the response variables of the testing dataset.



```
plot(test_new$y,type="l",lty=1.8,col="red")
lines(pred,type="l",lty=1.8,col="blue")
```



We can see, that the red lines (indicates predictions) almost overlaps with the blue lines (indicates the response variable values) and hence, we can conclude that our model did a good job as the fitting was moderate.

RESOURCES

- **Kaggle** (source of data)
- **MS Excel** (data conversion & cleansing)
- **RStudio** (linear models and testing)
- **Google** and **YouTube** (coding guidelines & subject knowledge)

BIBLIOGRAPHY

- Durbin, J. “Testing for Serial Correlation in Least Squares Regression When Some of the Regressors are Lagged Dependent Variables,” Econometrica, Vol. 38, May (1970), pp. 410-21.
- Automatic Autocorrelation and Spectral Analysis. Piet M. T. Broersen (2010)
- Robert L. Kaufman · (2013). Heteroskedasticity in Regression Detection and Correction.

- Thomas B. Fomby, R. Carter Hill, Stanley R. Johnson · (2012). Advanced Econometric Methods.
- Gilles Zumbach · (2012). Discrete Time Series, Processes, and Applications in Finance.
- Joseph P. Weir, William J. Vincent · (2020). Statistics in Kinesiology
- Richard Haase, Richard F. Haase · (2011). Multivariate General Linear Models.
- W. Kraemer, H. Sonnberger · (2012). The Linear Regression Model Under Test.
- John B. Guerard, Jr., John Guerard · (2013). Introduction to Financial Forecasting in Investment Analysis

IMPORTANT LINKS:

1) Google:

- https://www.statisticshowto.com/durbin-watson-test-coefficient/
- https://en.wikipedia.org/wiki/Durbin%E2%80%93Watson
- https://www.statisticshowto.com/breusch-pagan-godfrey-test/
- http://www.sthda.com/english/articles/39-regression-model-diagnostics/160-multicollinearity-essentials-and-vif-in-r/
- https://rpubs.com/cyobero/187387

2) YouTube:

- https://youtu.be/h_DGl79GqoM
- https://www.youtube.com/watch?v=Czb-2XylhNw
- https://www.youtube.com/watch?v=3es54FafNC0
- https://www.youtube.com/watch?v=oqY9vlqZbEY&t=146
- https://www.youtube.com/watch?v=pCGfZps716E
- https://www.youtube.com/watch?v=ME4M4dVRXTE
- https://www.youtube.com/watch?v=9Vz-CY3Gtmw