MORE ON CORRELATION AND REGRESSION WITH R

ALY6015: Intermediate Analytics

DHRUV VIJAY GUJRATHI

Northeastern University

DHRUV VIJAY GUJRATHI, College of Professional Studies, Northeastern University, Boston, MA 02115.

This report is regarding correlation and regression with R.

Dhruv Gujrathi is now a student at Department of Analytics, Northeastern University

Contact: gujrathi.d@northeastrn.edu

(NUID: 001029464)

INTRODUCTION

Linear Regression is used to predict or determine the value of an outcome variable based on one or many variables. This is done to establish a linear relationship (linear equation) between the outcome variable and the remaining variables. Regression analysis is a widely used tool to determine the connection or relationship between two variables. The variables are related to eachother through an equation where the power of both these variables are 1 and the if one visualizes the equation, it can be depicted using straight line. If there is an error in the equation, it will create a curve.

The equation can be depicted as follows:

$Y = ax+b$

where a and b are constants, x is the predictor variable and Y is the response variable.

Correlation is one of the most common statistics. Using one single value, it describes the "degree of relationship" between two variables. Correlation ranges from -1 to +1. Negative values of correlation indicate that as one variable increases the other variable decreases. Positive values of correlation indicate that as one variable increase the other variable increases as well. There are three options to calculate correlation in R, and we will introduce two of them below.

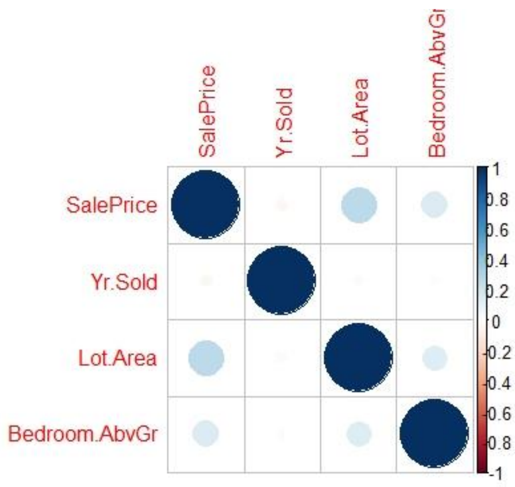
ANALYSIS

1. We load the dataset “AmesHousing.csv” in R using read.csv function.
2. In the step 2, we perform the exploratory data analysis using the library ‘tidyverse’. Basically, exploratory data analysis is the process of analyzing and visualizing the data to get a better understanding of it. This can involve determining the mean, median, max value, min value, etc. This is foremost done to examine the data for its distribution, outliers, and anomalies. We use cleandata() function to clean the variables and the data for any discrepancies so that the describe() function can be used. We select the variables or vectors that need to be analyzed and then form a table with all the corresponding values. Next, we plot the graph using ggplot() which will help us to determine the outliers and anomalies in the dataset if any value is not fitting in or if any value is missing. We can also determine which value is an outlier if that value is out of range as compared to the other values.
3. Imputing values in the missing spaces is done to fill in the vacant spaces in the dataset so that they can interact with the analysis properly without raising any errors or ambiguities. To impute the missing values, we follow certain steps which are:
    - Choosing values that keep the relationship in the dataset intact in place of missing values
    - Creating independently drawn imputed datasets
    - Calculating new standard errors using variation across datasets to consider the uncertainty created by these imputed datasets.

4. Correlation test is used to evaluate the association between two or more varia
bles. The cor( ) function is used to produce correlations to produces covarian
ces. The rcorr( ) function from the <u>Hmisc</u> package creates correlations/ covar
iances and significance levels for pearson and spearman correlations. Howev
er, input must be a matrix wherein a pairwise deletion is used. A correlation
matrix is a table of correlation coefficients for a set of variables used to deter
mine if a relationship exists between the variables. The coefficient indicates b
oth the strength of the relationship as well as the direction (positive vs. negati
ve correlations).

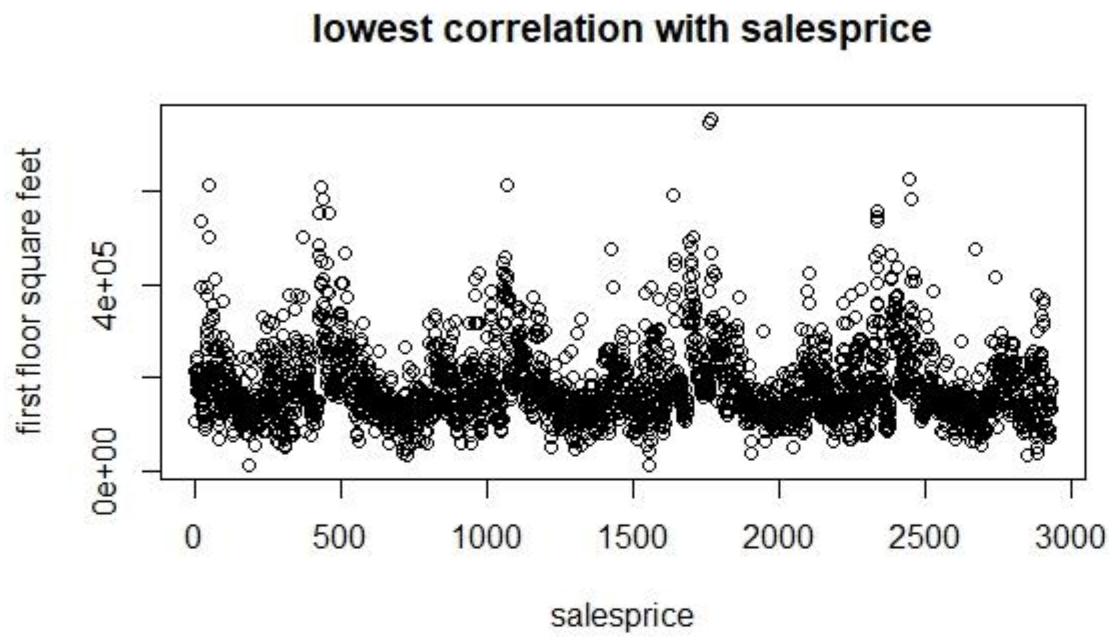
```
             SalePrice  Yr.Sold  Lot.Area   Bedroom.AbvGr
SalePrice        1.000   -0.031     0.267           0.144
Yr.Sold         -0.031    1.000    -0.023          -0.018
Lot.Area         0.267   -0.023     1.000           0.137
Bedroom.AbvGr    0.144   -0.018     0.137           1.000
```

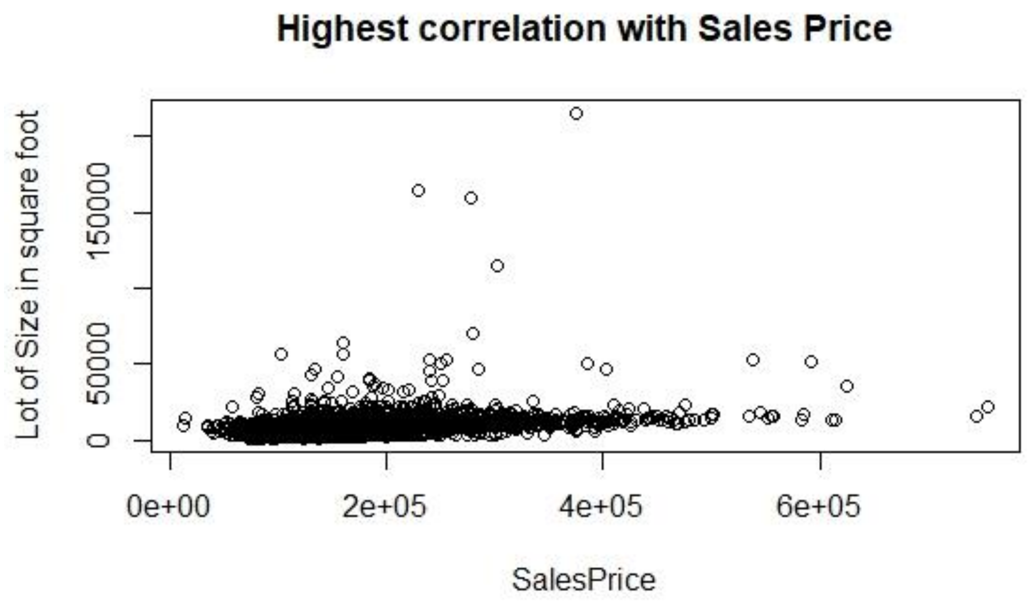
5. A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. It is used to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analyses. We firstly install the package Hmisc() to create correlogram. It takes the correlation matrix as the first argument and the second argument is used to display the upper triangular of the correlation matrix**.** It is important to note that if the data does not follow the normal distribution, at least closely enough, it is recommended to use the non-parametric correlation. The function corrplot() creates a graphical display of a correlation matrix, highlighting the most correlated variables in a data table. In a plot, correlation coefficients are coloured according to the value. Correlation matrix can be also reordered according to the degree of association between variables.



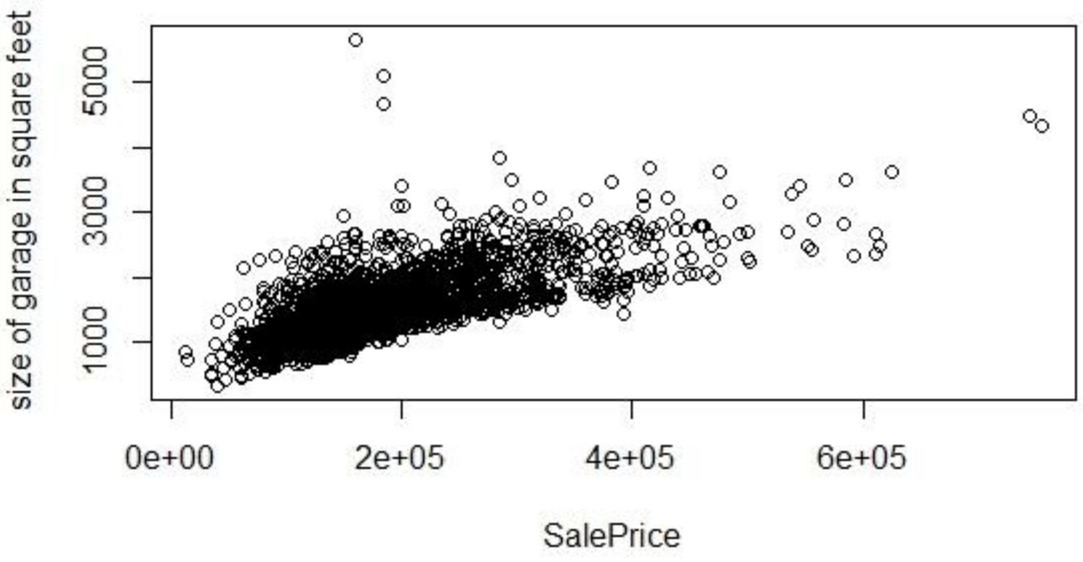
6. Scatter plots are like line graphs in that they use horizontal and vertical axes to plot data points. However, they have a very specific purpose. Scatter plots show how much one variable is affected by another. The relationship between two variables is called their correlation**.** In the first graph, since the plot is between the continuous variable with highest correlation versus SalePrice, it has a very strong positive correlation. The set of data points that are facing upward are relatively thin, thus there won't be a lot of difference in the results when one of the variables is entered. In the second graph, since the plot is between the continuous variable with lowest correlation versus SalePrice, it has a very strong negative correlation. The data points of the second graph are much more spread out, although they follow a downward pattern. In the third plot between continuous variable and SalePrice with correlation closest to 0.5,

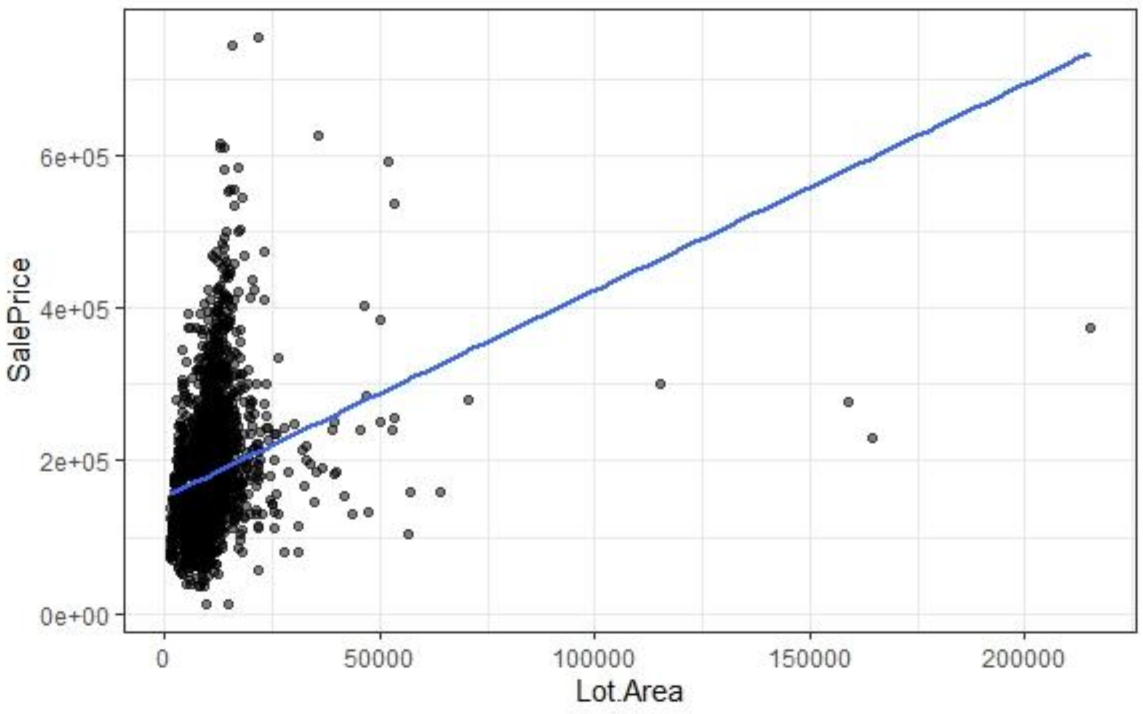
the plot appears to have a positive correlation, although the data points are not very close together.





Correlation with salesprice



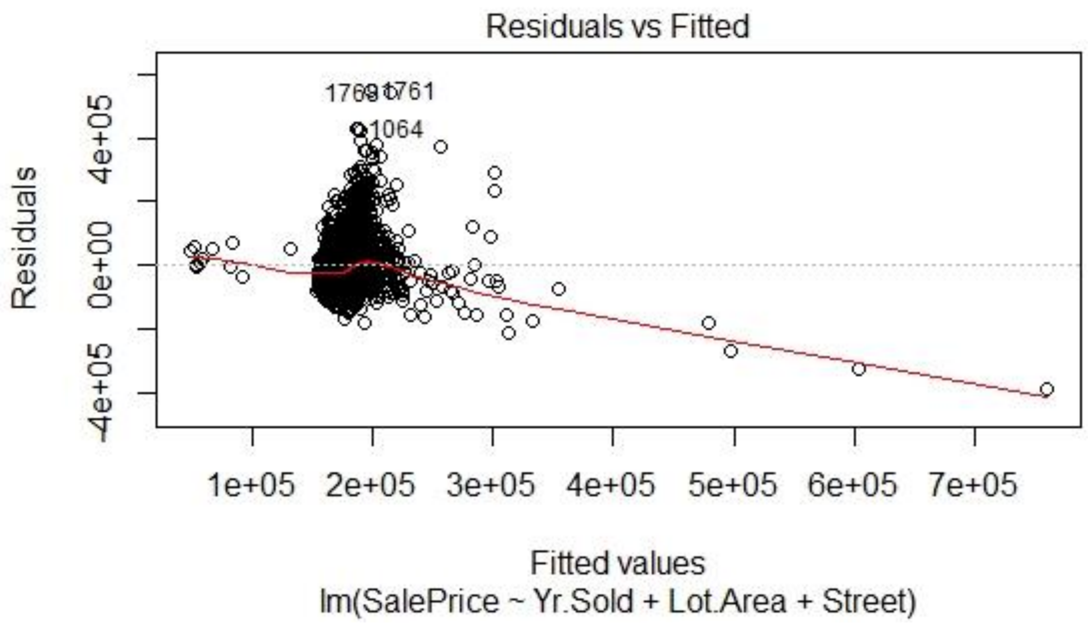


7. To fit the regression model in R, we use the tidyverse package. R has a built-in function for fitting linear regression models. The function lm() can be used to fit bivariate and multiple regression models, as well as analysis of variance, analysis of covariance, and other linear models. The first argument to lm() is an R “formula”, the second argument is a data frame. Calling summary on a fit model provides more detailed output. The fitted values component gives the predicted values for each observed value in the regression fit. The predicted values are plotted along the regression line. The ggplot library includes a geometric mapping, geom_smooth() that will fit a linear model for us and generate the corresponding regression plot.

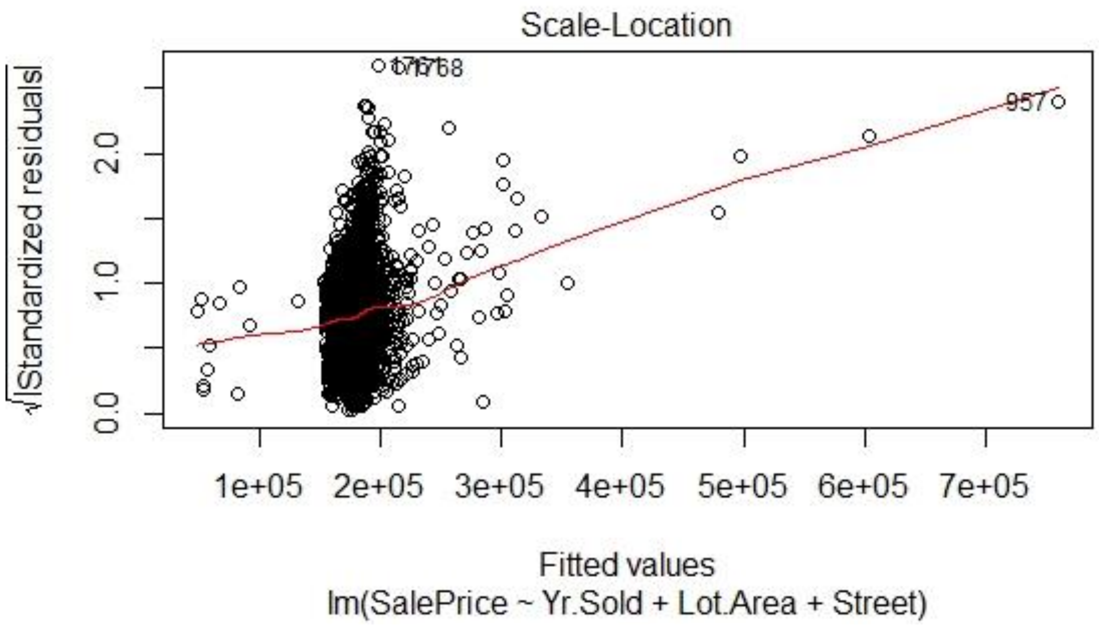
8. We know the basic skeleton of an equation can be written as : $b_0 + b_1 * x + e$
   where:
   b_0 : It is the intercept of the regression line
   b_1 : It is the slope of the regression line.
   e: It is the error term.
   After plotting the scatter plot of the regression model, we can determine that some points are above the line and some are below it. Overall, their mean comes to zero as the error (e). This is called as the residual sum of squares. To determine the regression in the equation form, we load the tidyverse package which helps in data manipulation and visualization. After that, we know the correlation between the components as we have already used the cor () function. The correlation function enables us to determine level of coordination between the two variables (b_0, b_1). Using the lm () function, we can determine the beta co-efficient of the linear model. From the output, we can conclude that the intercept is the b_0 variable and the second variable is b_1 which is known as the regression beta coefficient. In our model, YearsSold will be the b_0 variable, LotArea will be the b_1 variable and StreetPave will be the error €.

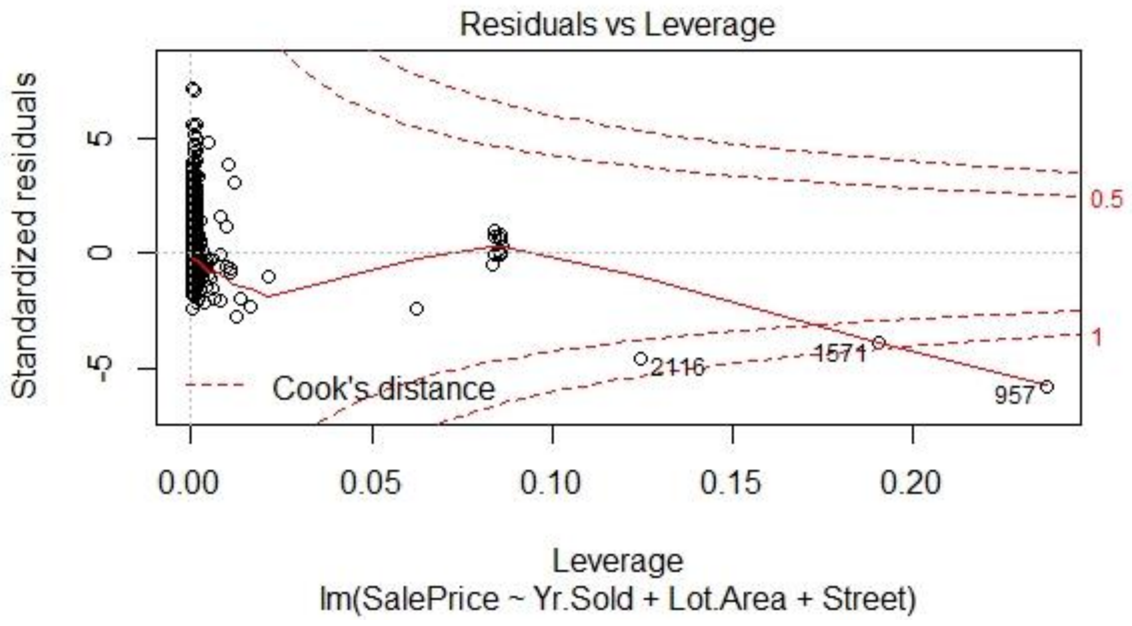
9.

- In the plot below we can see that the shape of the plot is not a curved or bell shape. Thus, the data is a non-linear data

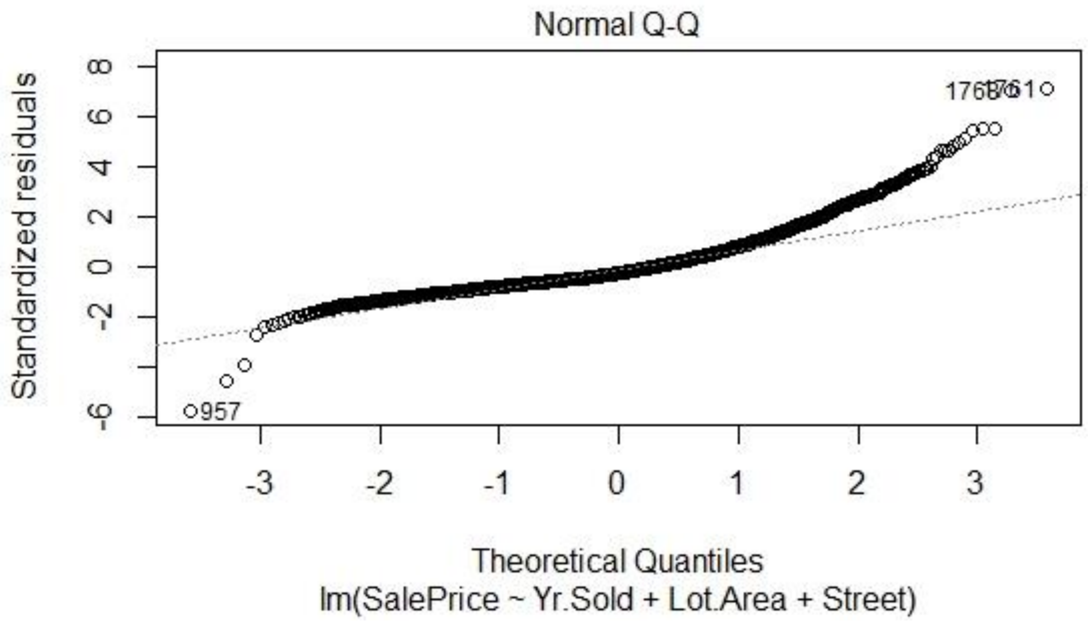


- This plot is also useful to determine heteroskedasticity. From the plot below we can determine that heteroskedasticity is not present

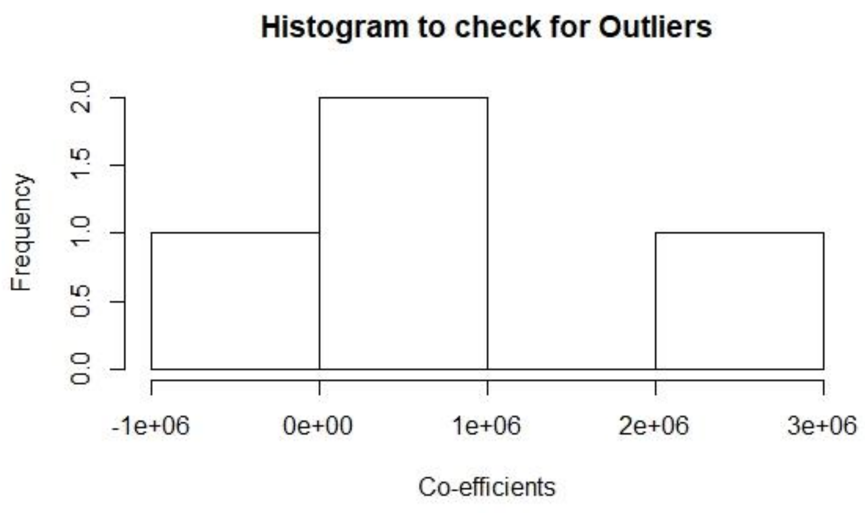




- This plot is used to determine the normal distribution of errors. It uses standardized values of residuals. Since the plot is a straight line, it shows normal distribution.



10. Multicollinearity is a phenomenon in which one predictor variable in a multiple regression model can be linearly predicted from the others, i.e. two or more predictors are strongly correlated. Multicollinearity may lead to severely biased regression coefficients and standard errors. The check collinearity () checks your model predictors for collinearity. The function works for simple as well as complex packages. In our model, since the first predictor value can easily predict the next two values as seen. The variance inflation factor is a measure to analyze the magnitude of multicollinearity of model terms. A VIF less than 5 indicates a low correlation of that predictor with other predictors. A value between 5 and 10 indicates a moderate correlation, while VIF values larger than 10 are a sign for high, not tolerable correlation of model predictors.
11. An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. To find the outliers in the model, we look at the histogram of our model. The isolated bars that are abnormally placed are the outliers in the model. The simplest way to identify the outliers is to find the values that fall outside the distribution. If we plot a scatter plot, the point which is placed very far from the regression line around which all the other points are placed are the outliers. Other ways to determine the outliers are:
    - Sorting the datasheet to find outliers.
    - Graphing the data to identify outliers.
    - Using Z-scores to detect outliers.



12. When I wanted to use the summary function in R. I found out that there were 2930 observations and 82 variables. Lot Frontage has 490 missing values which has the highest NA missing values. With correlation plot I was able to find out which was the highest and lowest correlation. The model which I used was correct and therefore I did not improve my model.

13. Firstly, we determine the summary of the regression model. The summary gives us the residuals, co-efficient and their p-values. To determine the best fit regression value, we need to test all possible combinations of the p -values. The package used here is tidyverse as we have visualized the data to determine the best fit. The summary () function reports the best set of variables for each model size. In case of our model, we can see that the best three-variable model (SalePrice ~ Yr.Sold+ Lot.Area + Street) contain all the three variables. Thus, the best fit model will be the one with the lowest p -value which in the case of our model is $2.2*e^{-16}$.
14. The model in step 12 and 13 are both dependent on the same co -efficient, errors and p-values. The only difference is the values these variables have in each case. As I mentioned in the point (13) that to determine the best fit regression value or model, the one with the lowest p -value will be the best fit model. Considering the above statement, we can say that the model in point (12) is not the best fit as compared to the model in point (13) as the p -value in (12) is higher as compared to the p -value in (13).

CONCLUSION

In this assignment, we were asked to interpret, clean, manipulate and evaluate the dataset that was given. This was done using the correlation matrix and the regression model. We were able to identify the best bit model based on the parameters we evaluated. The lowest p-value gave us the best fit regression model. We also determined the linear relation between the variables which helped us to find the rate of change in the sale price with respect to the other variables as they were linearly dependable or co-related to each other.

<u>REFERENCES</u>:

1. R-statistics.co. 2020. *Linear Regression with R*. Available at: <http://r-statistics.co/Linear-Regression.html>
2. R-statistics.co. 2020. *Linear Regression with R*. Available at: <http://r-statistics.co/Linear-Regression.html>

3. Guest, Krishnan, A., Rodrigues, L., Solomon, Katarzyna, The Stig, . . . Yokesh, R. (2020, November 03). Linear Models in R: Plotting Regression Lines. Retrieved November 04, 2020, from https://www.theanalysisfactor.com/linear-models-r-plotting-regression-lines/

APPENDIX:

i) **R-Code**:

```
#ALY6015_Dhruv_Gujrathi_Assignment1_R

install.packages("tidyverse") #Data Manipulation & Visualization

install.packages("janitor") #Examining & Cleaning Data

install.packages("broom") #Converts messy data to interpretable data

install.packages("Hmisc") #Imputing missing values, high level graphics, etc.

install.packages("dplyr") #Manipulating the Data

install.packages("ggplot2") #Creating Graphics


library(tidyverse) # data manipulation and cleaning

library(janitor) # cleaning up column names

library(broom) # tidying analysis results

library(Hmisc) # exploratory data analysis

library(dplyr) #Data manipulation

library(ggplot2) #Creating Graphics


#Loading AmesHousing.csv Dataset

data1<-read.csv("AmesHousing.csv")

#Cleaning the dataset for easing the use with R

clean_names(data1)

#Displaying the data characteristics

glimpse(data1)

#Descriptive Data Analysis

describe(data1 %>% select(SalePrice, Street, Yr.Sold))
```

```
#Selecting top values

head(map(data1, ~sum(is.na(.))))

#Exploratory data analysis

glimpse(data1)

frequency(data1)

#Creating an easy to read data frame

describe(data1)

#Prepare dataset for modelling by imputing missing values with the variable mean value

data1$SalePrice <- as.numeric(data1$SalePrice) #Converting to numeric class

data1_SalePrice_mean <- data1 %>%

  mutate(Street = replace(SalePrice,

                         is.na(SalePrice),

                         mean(SalePrice,  na.rm = T))) #Replacing NA/Nan Values

data1_SalePrice_mean


data1$Street <- as.numeric(data1$Street) #Converting to numeric class

data1_Street_mean <- data1 %>%

  mutate(Street = replace(Street,

                         is.na(Street),

                         mean(Street, na.rm = T))) #Replacing NA/Nan Values

data1_Street_mean


data1$Yr.Sold <- as.numeric(data1$Yr.Sold) #Converting to numeric class

data1_Yrsold_mean <- data1 %>%

  mutate(Yr.Sold = replace(Yr.Sold,
                    is.na(Yr.Sold),
                    mean(Yr.Sold, na.rm = T))) #Replacing NA/Nan Values
data1_Yrsold_mean
#Using the cor function for creating the correlation matrix
x<-data1 %>%
  select(SalePrice, Yr.Sold, Lot.Area, Bedroom.AbvGr) %>%
  cor() %>%
  round(3)
x
#Producing a cor relation plot of the correlation matrix
library(corrplot)
corrplot(x, method = "circle")
#Comparing the correlation plots and determing the patterns
plot(data1$SalePrice,data1$`Lot.Area`,main="Highest correlation with Sales Price",xlab="SalesPrice", ylab="Lot of Size in square foot")
plot(data1$SalePrice, data1$`1st Flr SF`, main = "lowest correlation with salesprice",
     xlab="salesprice", ylab="first floor square feet")
plot(data1$SalePrice,data1$`Gr.Liv.Area`,main="Correlation with salesprice", xlab = "SalePrice", ylab ="size of garage in square feet" )
#plotting ggplot
ggplot(data1,aes(x=`Lot.Area`,y=SalePrice, main="Correlation with salesprice with regression line"))+
  geom_point(alpha=.5)+
  geom_smooth(method = "lm", se=F) +
  theme_bw()
# Fitting Regression Model
```

```
mod <- lm(SalePrice  ~ Yr.Sold+ Lot.Area + Street , data = data1)

print(mod)

summary(mod)

#Rounding the data

tidy(mod) %>%

  mutate(estimate = round(estimate, 3))

# Writing this in equation form

#hat(Price) = 2756000-1356.777 \times Yr.Sold + 2.828 \times Lot.Area + 120662.749
Street$$

#Plotting Regression model

plot(mod)

#Checking for multicolinearity

library(car)

vif(mod)

#Checking for outliers

hist(mod$coefficients,

     xlab = "Co-efficients",

     main = "Histogram to check for Outliers",)

#Subsets regression method to identify best model

summary(mod)

res.sum <- summary(mod)

data.frame(

  Adj.R2 = which.max(res.sum$adjr2),

  CP = which.min(res.sum$cp),

  BIC = which.min(res.sum$bic))
```