MODULE ONE PROJECT

REGRESSION DIAGNOSTICS

Submitted By: **HARSHIT GAUR**
MASTER OF PROFESSIONAL STUDIES IN ANALYTICS
ALY 6015 : INTERMEDIATE ANALYTICS
CRN : 21454
FEBRUARY 27, 2022
WINTER 2022

Submitted To: **PROF. ROY WADA**

INTRODUCTION

The **Linear Regression** model is one where the input variables (x) and the single output variable (y) assume a linear relationship. In other words, the y of a linear regression is a function of the x of the input variables.

There are two types of linear regression methods: **Simple Linear Regression** when there is just one input variable (x) and **Multiple Linear Regression** when there are two or more input variables.

A linear regression equation can be prepared using various techniques, the most common of which is ***Ordinary Least Squares***. A model prepared this way is typically referred to as Ordinary Least Squares Regression or as just Least Squares Regression.

A linear equation represents this representation as the predicted output for a given set of input values (x), with the solution of that equation being the input values (x). Thus, both the input values (x) and the output value (y) are numeric. Each input value or column is given a scale factor, called a coefficient, which is represented by the Greek capital letter Beta (β). An additional coefficient is also added, giving the line an additional degree of freedom and referred to as an intercept.

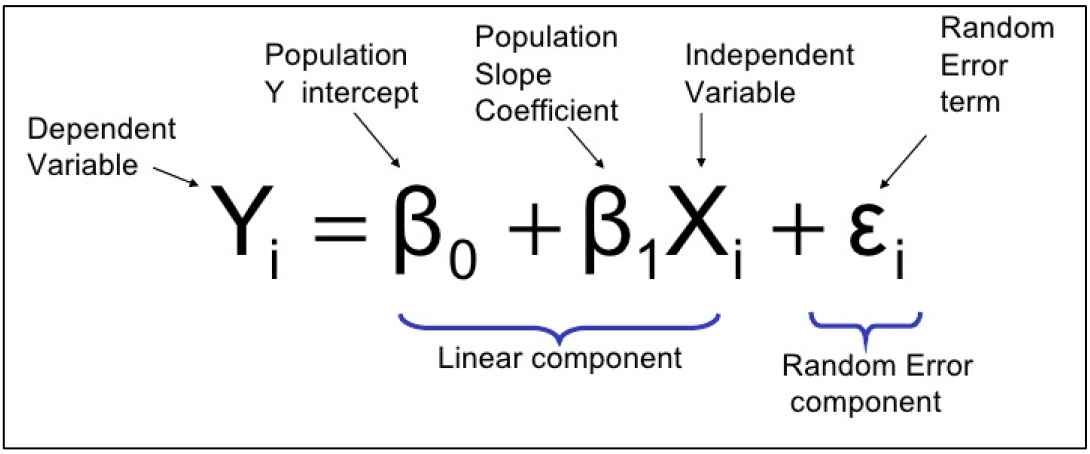

*Figure 1.1: Mathematical Equation of **Simple Linear Regression***

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon$$

where, for $i = n$ observations:

y_i = dependent variable

x_i = expanatory variables

β_0 = y-intercept (constant term)

β_p = slope coefficients for each explanatory variable

ϵ = the model's error term (also known as the residuals)

*Figure 1.2: Mathematical Equation of **Multiple Linear Regression***

Limitation of Ordinary Least Squares -

There are several assumptions required for Ordinary Least Squares. Failure to meet them will mean that the results may not be reliable. OLS can be susceptible to overfitting which occurs when the model learns the data too well. This leads to the problem that the model learns the data so well that it performs poorly when provided with new data. The assumptions are:

- Normality
- Independence of errors
- Linearity
- Homoscedasticity

We will talk about them later in the report when discussing them with the plots and observations.

ANALYSIS

With the medium of this project, we will be performing *Regression Diagnostics* of the ***Ames Housing*** dataset. We will use the implementations of Simple Linear Regression, Check-for-Multicollinearity to analyse the data set.

In trying to unravel its patterns, it is easy to become engrossed in the Ames Housing dataset's enormous amount of features. It is both alluring and bewildering in equal measure.

The dataset contains various numeric and categorical data. The variables are defined using *Character domain* and *Integer domain.* The dataset contains information regarding the attributes of an house (example- year built, total area, overall quality, number of floors, conditions of basement, conditions of floors, porch area, etc.) and has:

- 2,930 data points
- 82 Features

A glimpse or the structure of the dataset using the ***glimpse()*** function of *tidyverse* library.

```
> # Get a Glimpse/View of the data set
> glimpse(AmesHousing)
Rows: 2,930
Columns: 82
$ Order           <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, …
$ PID             <int> 526301100, 526350040, 526351010, 526353030, 527105010, 527105030, 5…
$ MS.SubClass     <int> 20, 20, 20, 20, 60, 60, 120, 120, 120, 60, 60, 20, 60, 20, 120, 60,…
$ MS.Zoning       <chr> "RL", "RH", "RL", "RL", "RL", "RL", "RL", "RL", "RL", "RL", "RL", "…
$ Lot.Frontage    <int> 141, 80, 81, 93, 74, 78, 41, 43, 39, 60, 75, NA, 63, 85, NA, 47, 15…
$ Lot.Area        <int> 31770, 11622, 14267, 11160, 13830, 9978, 4920, 5005, 5389, 7500, 10…
$ Street          <chr> "Pave", "Pave", "Pave", "Pave", "Pave", "Pave", "Pave", "Pave", "Pa…
$ Alley           <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
$ Lot.Shape       <chr> "IR1", "Reg", "IR1", "Reg", "IR1", "IR1", "Reg", "IR1", "IR1", "Reg…
$ Land.Contour    <chr> "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", "HLS", "Lvl", "Lvl…
$ Utilities       <chr> "AllPub", "AllPub", "AllPub", "AllPub", "AllPub", "AllPub", "AllPub…
$ Lot.Config      <chr> "Corner", "Inside", "Corner", "Corner", "Inside", "Inside", "Inside…
$ Land.Slope      <chr> "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", "Gtl…
$ Neighborhood    <chr> "NAmes", "NAmes", "NAmes", "NAmes", "Gilbert", "Gilbert", "StoneBr"…
$ Condition.1     <chr> "Norm", "Feedr", "Norm", "Norm", "Norm", "Norm", "Norm", "Norm", "N…
$ Condition.2     <chr> "Norm", "Norm", "Norm", "Norm", "Norm", "Norm", "Norm", "Norm", "No…
$ Bldg.Type       <chr> "1Fam", "1Fam", "1Fam", "1Fam", "1Fam", "1Fam", "TwnhsE", "TwnhsE",…
$ House.Style     <chr> "1Story", "1Story", "1Story", "1Story", "2Story", "2Story", "1Story…
$ Overall.Qual    <int> 6, 5, 6, 7, 5, 6, 8, 8, 8, 7, 6, 6, 6, 7, 8, 8, 8, 9, 4, 6, 6, 7, 7…
$ Overall.Cond    <int> 5, 6, 6, 5, 5, 6, 5, 5, 5, 5, 5, 7, 5, 5, 5, 5, 7, 2, 5, 6, 6, 6, 5…
$ Year.Built      <int> 1960, 1961, 1958, 1968, 1997, 1998, 2001, 1992, 1995, 1999, 1993, 1…
$ Year.Remod.Add  <int> 1960, 1961, 1958, 1968, 1998, 1998, 2001, 1992, 1996, 1999, 1994, 2…
$ Roof.Style      <chr> "Hip", "Gable", "Hip", "Hip", "Gable", "Gable", "Gable", "Gable", "…
$ Roof.Matl       <chr> "CompShg", "CompShg", "CompShg", "CompShg", "CompShg", "CompShg", "…
$ Exterior.1st    <chr> "BrkFace", "VinylSd", "Wd Sdng", "BrkFace", "VinylSd", "VinylSd", "…
$ Exterior.2nd    <chr> "Plywood", "VinylSd", "Wd Sdng", "BrkFace", "VinylSd", "VinylSd", "…
$ Mas.Vnr.Type    <chr> "Stone", "None", "BrkFace", "None", "None", "BrkFace", "None", "Non…
$ Mas.Vnr.Area    <int> 112, 0, 108, 0, 0, 20, 0, 0, 0, 0, 0, 0, 0, 0, 0, 603, 0, 350, 0, 1…
$ Exter.Qual      <chr> "TA", "TA", "TA", "Gd", "TA", "TA", "Gd", "Gd", "Gd", "TA", "TA", "…
$ Exter.Cond      <chr> "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "…
$ Foundation      <chr> "CBlock", "CBlock", "CBlock", "CBlock", "PConc", "PConc", "PConc", …
```

Figure 1.3: Glimpse/ Structure of the Ames Housing dataset

The structure/glimpse in the above figure shows that most of the variables belong to either character domain or integer domain. From the reference textbook, we can concur that some of the character domain variables are actually factors. So, we will first explore the data more to find patterns and then convert it into factors.

Some of the data points from the data set of Ames Housing are present below:

	Order	PID	MS.SubClass	MS.Zoning	Lot.Frontage	Lot.Area	Street	Alley	Lot.Shape	Land.Contour	Utilities	Lot.Confi
1	1	526301100	20	RL	141	31770	Pave	NA	IR1	Lvl	AllPub	Corner
2	2	526350040	20	RH	80	11622	Pave	NA	Reg	Lvl	AllPub	Inside
3	3	526351010	20	RL	81	14267	Pave	NA	IR1	Lvl	AllPub	Corner
4	4	526353030	20	RL	93	11160	Pave	NA	Reg	Lvl	AllPub	Corner
5	5	527105010	60	RL	74	13830	Pave	NA	IR1	Lvl	AllPub	Inside
6	6	527105030	60	RL	78	9978	Pave	NA	IR1	Lvl	AllPub	Inside
7	7	527127150	120	RL	41	4920	Pave	NA	Reg	Lvl	AllPub	Inside
8	8	527145080	120	RL	43	5005	Pave	NA	IR1	HLS	AllPub	Inside
9	9	527146030	120	RL	39	5389	Pave	NA	IR1	Lvl	AllPub	Inside
10	10	527162130	60	RL	60	7500	Pave	NA	Reg	Lvl	AllPub	Inside
11	11	527163010	60	RL	75	10000	Pave	NA	IR1	Lvl	AllPub	Corner
12	12	527165230	20	RL	NA	7980	Pave	NA	IR1	Lvl	AllPub	Inside
13	13	527166040	60	RL	63	8402	Pave	NA	IR1	Lvl	AllPub	Inside
14	14	527180040	20	RL	85	10176	Pave	NA	Reg	Lvl	AllPub	Inside
15	15	527182190	120	RL	NA	6820	Pave	NA	IR1	Lvl	AllPub	Corner
16	16	527216070	60	RL	47	53504	Pave	NA	IR2	HLS	AllPub	CulDSac
17	17	527225035	50	RL	152	12134	Pave	NA	IR1	Bnk	AllPub	Inside
18	18	527258010	20	RL	88	11394	Pave	NA	Reg	Lvl	AllPub	Corner
19	19	527276150	20	RL	140	19138	Pave	NA	Reg	Lvl	AllPub	Corner
20	20	527302110	20	RL	85	13175	Pave	NA	Reg	Lvl	AllPub	Inside

Figure 1.4: Data Points (Summary) of the Ames Housing dataset

INITIAL DATA ANALYSIS

Upon observing the dataset, we can find that two of the variables are actually identifier variables. Their purpose is for the identification of the data point/ record in the data set.

There is no mandate requirement of these two variables in the dataset for our analysis and hence, we can remove them from the dataset.

```
# Removing 'Identifier' variables from the data set
AmesHousing <- AmesHousing %>%
  dplyr::select(-Order, -PID)
```

Figure 1.5: Removal of 'Identifier' variables from the dataset.

```
> # Get a Glimpse/View of the data set
> glimpse(AmesHousing)
Rows: 2,930
Columns: 80
$ MS.SubClass      <int> 20, 20, 20, 20, 60, 60, 120, 120, 120, 60, 60, 20, 60, 20, 120, 60, 50, 20, 20, 20, 20, …
$ MS.Zoning        <chr> "RL", "RH", "RL", "RL", "RL", "RL", "RL", "RL", "RL", "RL", "RL", "RL", "RL", "RL", "RL"…
$ Lot.Frontage     <int> 141, 80, 81, 93, 74, 78, 41, 43, 39, 60, 75, NA, 63, 85, NA, 47, 152, 88, 140, 85, 105, …
$ Lot.Area         <int> 31770, 11622, 14267, 11160, 13830, 9978, 4920, 5005, 5389, 7500, 10000, 7980, 8402, 1017…
$ Street           <chr> "Pave", "Pave", "Pave", "Pave", "Pave", "Pave", "Pave", "Pave", "Pave", "Pave", "Pave", …
$ Alley            <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
$ Lot.Shape        <chr> "IR1", "Reg", "IR1", "Reg", "IR1", "IR1", "Reg", "IR1", "IR1", "Reg", "IR1", "IR1", "IR1…
$ Land.Contour     <chr> "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", "HLS", "Lvl", "Lvl", "Lvl", "Lvl", "Lvl…
$ Utilities        <chr> "AllPub", "AllPub", "AllPub", "AllPub", "AllPub", "AllPub", "AllPub", "AllPub", "AllPub"…
```

Figure 1.6: Structure of the dataset after removing the identifier variables.

EXPLORATORY DATA ANALYSIS

The descriptive statistics of the features/variables of the dataset can be summarised to be in order to easily calculate the statistics.

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se	IQR	Q0.25	Q0.75
MS.SubClass	1	2930	57.38737201	42.63802455	50.0	50.0	44.4780	20	190	170	1.35618974	1.37937166	0.787704389	50.00	20.00	70.00
MS.Zoning*	2	2930	5.96723549	0.86565171	6.0	6.0	0.0000	1	7	6	-2.61395502	8.40518201	0.015992243	0.00	6.00	6.00
Lot.Frontage	3	2440	69.22459016	23.36533497	68.0	68.0	17.7912	21	313	292	1.49722474	11.19773896	0.473017380	22.00	58.00	80.00
Lot.Area	4	2930	10147.92184300	7880.01775944	9436.5	9436.5	3024.5040	1300	215245	213945	12.80777396	264.38697056	145.577208077	4115.00	7440.25	11555.25
Street*	5	2930	1.99590444	0.06387630	2.0	2.0	0.0000	1	2	1	-15.52172448	239.00550300	0.001180065	0.00	2.00	2.00
Alley*	6	198	1.39393939	0.48986027	1.0	1.0	0.0000	1	2	1	0.43083693	-1.82351281	0.034812853	1.00	1.00	2.00
Lot.Shape*	7	2930	2.94027304	1.41210537	4.0	4.0	0.0000	1	4	3	-0.60626528	-1.60269619	0.026087550	3.00	1.00	4.00
Land.Contour*	8	2930	3.77781570	0.70319906	4.0	4.0	0.0000	1	4	3	-3.12265071	8.43622338	0.012991056	0.00	4.00	4.00
Utilities*	9	2930	1.00170648	0.05540585	1.0	1.0	0.0000	1	3	2	34.02017668	1187.95713981	0.001023580	0.00	1.00	1.00
Lot.Config*	10	2930	4.05529010	1.60392177	5.0	5.0	0.0000	1	5	4	-1.19376664	-0.44471094	0.029631209	2.00	3.00	5.00
Land.Slope*	11	2930	1.05358362	0.24830416	1.0	1.0	0.0000	1	3	2	4.98305950	26.62467063	0.004587227	0.00	1.00	1.00
Neighborhood*	12	2930	15.29522184	7.02207496	16.0	16.0	8.8956	1	28	27	-0.19601921	-1.18535478	0.129727381	13.00	8.00	21.00
Condition.1*	13	2930	3.04027304	0.87240769	3.0	3.0	0.0000	1	9	8	2.98768538	15.73853187	0.016117055	0.00	3.00	3.00
Condition.2*	14	2930	3.00204778	0.20903762	3.0	3.0	0.0000	1	8	7	12.07671399	308.97293476	0.003861808	0.00	3.00	3.00
Bldg.Type*	15	2930	1.51706485	1.21889735	1.0	1.0	0.0000	1	5	4	2.15183912	3.00389360	0.022518182	0.00	1.00	1.00
House.Style*	16	2930	4.02389078	1.91062934	3.0	3.0	0.0000	1	8	7	0.32117830	-0.95014281	0.035297393	3.00	3.00	6.00
Overall.Qual	17	2930	6.09488055	1.41102608	6.0	6.0	1.4826	1	10	9	0.19043881	0.04819422	0.026067611	2.00	5.00	7.00
Overall.Cond	18	2930	5.56313993	1.11153656	5.0	5.0	0.0000	1	9	8	0.57384146	1.48379654	0.020534775	1.00	5.00	6.00

Figure 1.7: Summary (Descriptive Statistics) of the dataset.

Ames Housing Summary Statistics

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 50	Pctl. 75	Max
MS.SubClass	2930	57.387	42.638	20	20	50	70	190
Lot.Frontage	2440	69.225	23.365	21	58	68	80	313
Lot.Area	2930	10147.922	7880.018	1300	7440.25	9436.5	11555.25	215245
Street	2930							
... Grvl	12	0.4%						
... Pave	2918	99.6%						
Alley	198							
... Grvl	120	60.6%						
... Pave	78	39.4%						
Lot.Shape	2930							
... IR1	979	33.4%						
... IR2	76	2.6%						
... IR3	16	0.5%						
... Reg	1859	63.4%						
Land.Contour	2930							
... Bnk	117	4%						
... HLS	120	4.1%						
... Low	60	2%						
... Lvl	2633	89.9%						
Utilities	2930							
... AllPub	2927	99.9%						
... NoSeWa	1	0%						
... NoSewr	2	0.1%						

Figure 1.8: Summary (Descriptive Statistics) of the dataset in a different way.

We loaded **'PSYCH'** package/library to utilize the **'DESCRIBE()'** function for the descriptive statistics of the data set. The following observations have been figured out -

1. The **mean** of the attribute *Lot.Frontage* is around **69.22** with a **standard deviation** of **23.36** and **quartiles value (Lower Quartile - 58, Higher Quartile - 80).**
   From the observations, we can keep an anticipation that the data points around **maximum value (313)** can be *outliers* to the feature since the Inter-Quartile range is around 22.

2. The **mean** of the attribute *Overall.Qual* is around **6.09** with a **standard deviation** of **1.41** and **quartiles value (Lower Quartile - 5, Higher Quartile - 7).**

3. The **mean** of the attribute *SalePrice* is around **180796.06** with a **standard deviation** of **79886.69** and **quartiles value (Lower Quartile - 129500, Higher Quartile - 213500).**
   From the observations, we can keep an anticipation that the data points around **maximum value (755000)** can be *outliers* to the feature since the Inter-Quartile range is around 84000.

Different visualization graphs have been plot to analyse the patterns in the dependent and independent variables of the dataset.

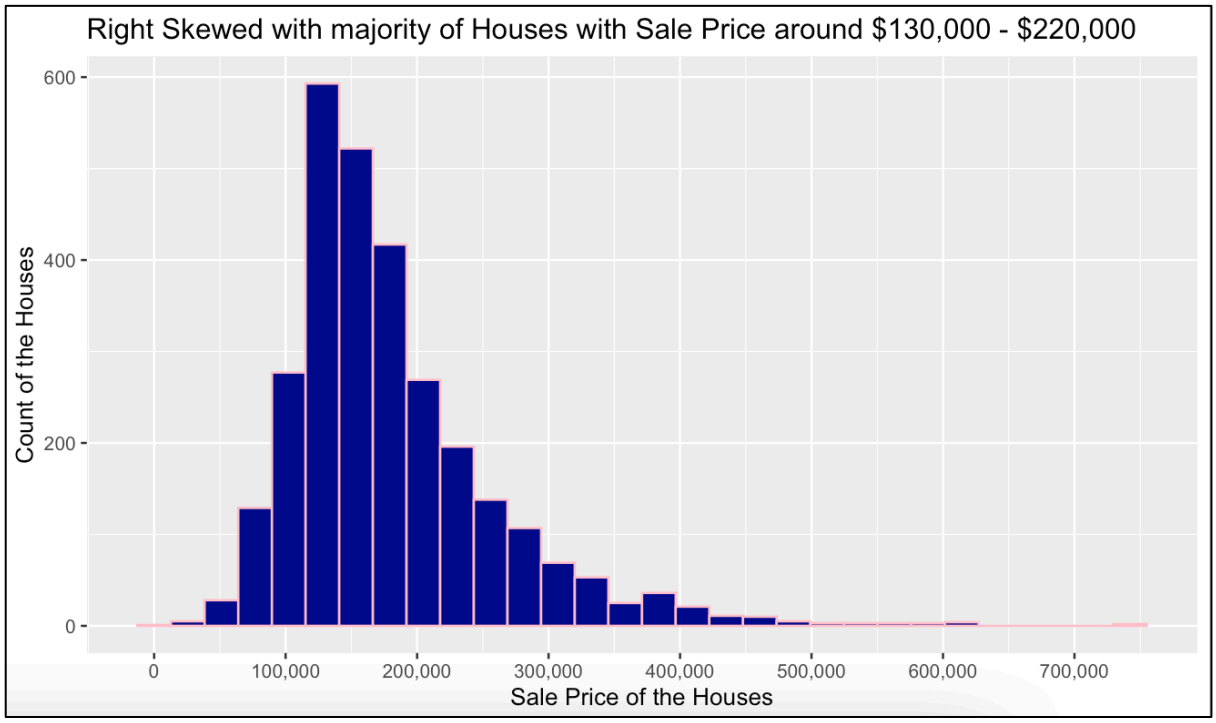

Figure 1.9: Histogram of Sale Price which shows Right-Skewness.

- From the histogram of Sale Price attribute (Figure 1.9) in the dataset, we can figure out that the feature is Right-Skewed in nature.
- The majority of houses have their sale prices around $130,000 - $220,000.

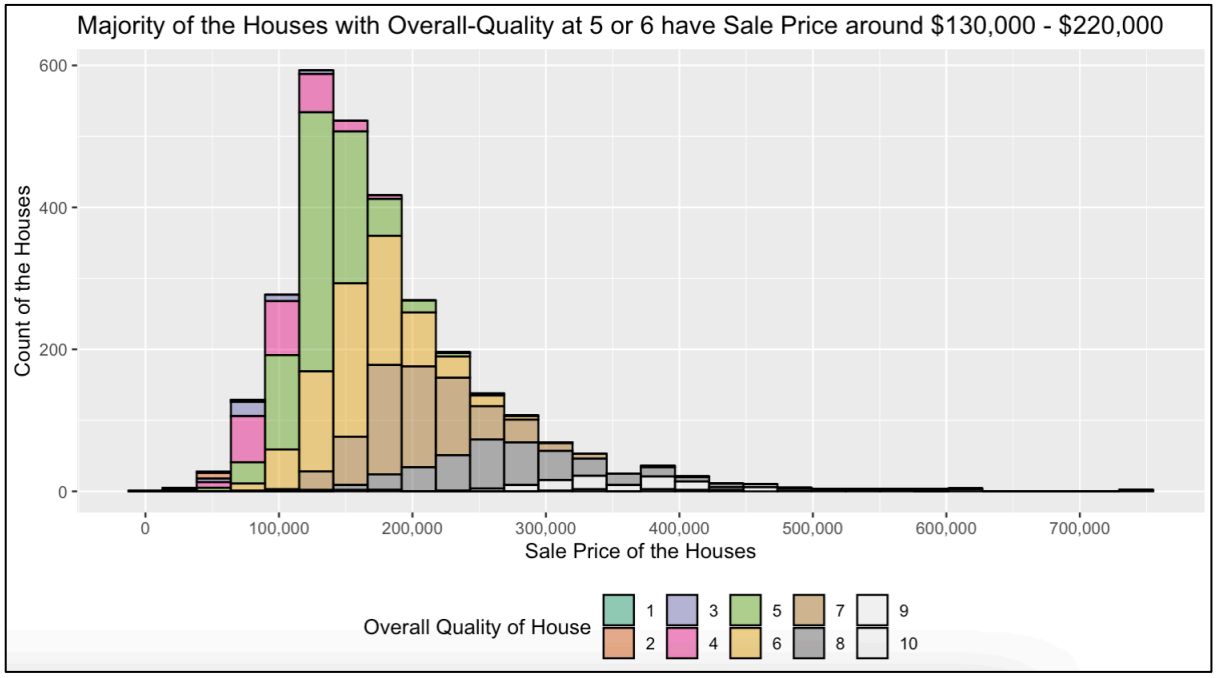

Figure 1.10: Histogram of Sale Price grouped by Overall-Quality.

- The histogram of Sale Price attribute grouped with the Overall-Quality feature (Figure 1.10) in the dataset depicts that the majority of the houses belong to the Overall-Qualities of either 5 or 6.
- This means that most of the houses around the price range of $130,000 - $220,000 have an overall quality of either 5 or 6.

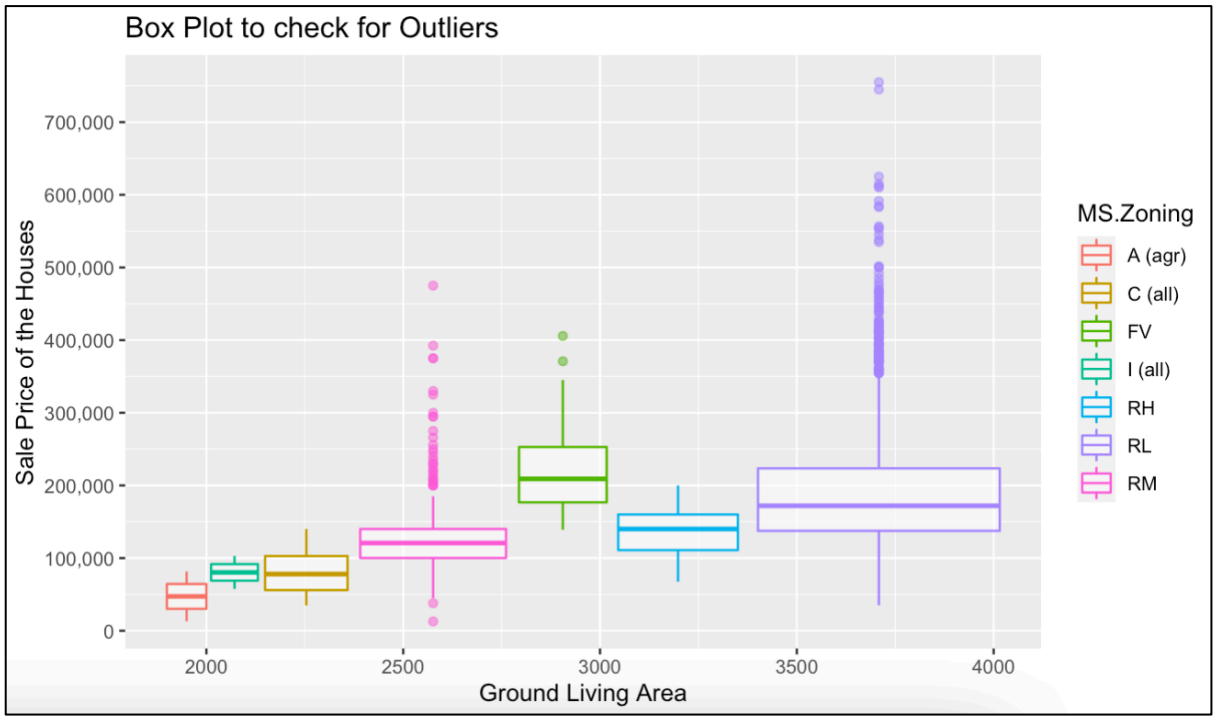

Figure 1.11: Boxplot of Ground Living Area with Sale Price varied by MS Zoning.

- The boxplot of Ground Living Area with Sale Price varied by MS Zoning attribute shows that there are several Outliers belonging to **RM** zone and **RL** zone.
- It also depicts that Sale Prices of houses belonging to RL zone are usually higher than the sale prices of houses belonging to other zones.

DATA IMPUTATION

Data cleaning and imputation is an important and necessary factor in the data analysis process. Upon checking the dataset for missing values/ NA values, we found various features containing them. We can check the records/ data points containing NA/missing values in any of their features using the below code-script.

```
# Checking the records with missing/NA values
AmesHousing %>%
  filter(!complete.cases(AmesHousing)) %>%
  View()

# Summing up the records with missing/NA values according to the Variables respectively
colSums(is.na(AmesHousing))
```

Figure 1.12: Code-Script to check the missing/NA values in the dataset.

```
> # Summing up the records with missing/NA values according to the Variables respectively
> colSums(is.na(AmesHousing))
    MS.SubClass       MS.Zoning    Lot.Frontage        Lot.Area          Street           Alley
              0               0             490               0               0            2732
      Lot.Shape     Land.Contour       Utilities      Lot.Config     Land.Slope    Neighborhood
              0               0               0               0               0               0
    Condition.1     Condition.2        Bldg.Type     House.Style    Overall.Qual    Overall.Cond
              0               0               0               0               0               0
     Year.Built    Year.Remod.Add      Roof.Style       Roof.Matl     Exterior.1st     Exterior.2nd
              0               0               0               0               0               0
   Mas.Vnr.Type     Mas.Vnr.Area      Exter.Qual      Exter.Cond      Foundation       Bsmt.Qual
              0              23               0               0               0              79
      Bsmt.Cond   Bsmt.Exposure   BsmtFin.Type.1   BsmtFin.SF.1   BsmtFin.Type.2   BsmtFin.SF.2
             79              79              79               1              79               1
    Bsmt.Unf.SF    Total.Bsmt.SF         Heating      Heating.QC     Central.Air      Electrical
              1               1               0               0               0               0
    X1st.Flr.SF     X2nd.Flr.SF   Low.Qual.Fin.SF    Gr.Liv.Area   Bsmt.Full.Bath   Bsmt.Half.Bath
              0               0               0               0               2               2
      Full.Bath       Half.Bath   Bedroom.AbvGr   Kitchen.AbvGr    Kitchen.Qual   TotRms.AbvGrd
              0               0               0               0               0               0
     Functional      Fireplaces    Fireplace.Qu     Garage.Type    Garage.Yr.Blt   Garage.Finish
              0               0            1422             157             159             157
    Garage.Cars     Garage.Area     Garage.Qual     Garage.Cond     Paved.Drive     Wood.Deck.SF
              1               1             158             158               0               0
  Open.Porch.SF   Enclosed.Porch     X3Ssn.Porch    Screen.Porch       Pool.Area         Pool.QC
              0               0               0               0               0            2917
          Fence    Misc.Feature        Misc.Val         Mo.Sold         Yr.Sold       Sale.Type
           2358            2824               0               0               0               0
 Sale.Condition       SalePrice
              0               0
```

Figure 1.13: Sum of missing/NA values in the dataset according to features.

- From the observations, we can figure out that the feature *Lot.Frontage* has 490 missing/NA values in it.
- We can also figure out that the feature *Bsmt,Qual* has 79 missing/NA values in it and so does some of the other features related to Basement (*BSMT*).

To impute these missing/NA values, we have performed the following steps:

- Missing/NA values will be imputed by some text ***(example- No Basement, No Fence, No Alley, etc)*** in the Character (we will factorize them later because of their nature) variables.
- Missing/NA values in the numerical variables will be replaced by their **mean** values.

- Some of the missing/NA values in the variables related to Basement belong to those records which do not have basements in them. Therefore, for these variables, the missing/NA values of
    - numerical features are replaced with **0**
    - categorical features are replaced with **"No Basement"**.

```
AmesHousing <- AmesHousing %>%
  mutate(Alley = replace_na(Alley, "No Alley")) %>%
  mutate(Lot.Frontage = replace_na(Lot.Frontage, as.integer(mean(Lot.Frontage, na.rm = TRUE)))) %>%
  mutate(Mas.Vnr.Area = replace_na(Mas.Vnr.Area, 0)) %>%
  mutate(Bsmt.Qual = replace_na(Bsmt.Qual, "No Basement")) %>%
  mutate(Bsmt.Cond = replace_na(Bsmt.Cond, "No Basement")) %>%
  mutate(Bsmt.Exposure = replace_na(Bsmt.Exposure, "No Basement")) %>%
  mutate(BsmtFin.Type.1 = replace_na(BsmtFin.Type.1, "No Basement")) %>%
  mutate(BsmtFin.SF.1 = replace_na(BsmtFin.SF.1, 0)) %>%
  mutate(BsmtFin.Type.2 = replace_na(BsmtFin.Type.2, "No Basement")) %>%
  mutate(BsmtFin.SF.2 = replace_na(BsmtFin.SF.2, 0)) %>%
  mutate(Bsmt.Unf.SF = replace_na(Bsmt.Unf.SF, 0)) %>%
  mutate(Total.Bsmt.SF = replace_na(BsmtFin.SF.2, 0)) %>%
```

Figure 1.14: Imputation of missing/NA values in the data set.

CORRELATION

A correlation matrix and correlation graph have been generated for the numeric & integer features of the dataset. It will help in figuring out the most correlated features and the least correlated features with our dependent feature - Sale Price.

```
#####################################################################
# Correlation
#####################################################################

numIntFeatures_AmesHousing <- AmesHousing[sapply(AmesHousing, is.numeric)]
View(round(cor(numIntFeatures_AmesHousing, use = "pairwise"), 5))
corrplot(cor(numIntFeatures_AmesHousing, use = "pairwise"), tl.cex = 0.7, type = "upper",
         title = "Correlation Plot", mar = c(0,0,1,0),
         col = brewer.pal(n = ncol(numIntFeatures_AmesHousing), name = "RdYlBu"))
```

Figure 1.15: Code-Script to find the Correlation Matrix and Correlation Plot.

	MS.SubClass	Lot.Frontage	Lot.Area	Overall.Qual	Overall.Cond	Year.Built
Garage.Cars	-0.04607	0.29045	0.17946	0.59955	-0.18170	0.53798
Garage.Area	-0.10337	0.33795	0.21275	0.56356	-0.15392	0.48073
Wood.Deck.SF	-0.01731	0.10430	0.15721	0.25566	0.02034	0.22896
Open.Porch.SF	-0.01482	0.15037	0.10376	0.29841	-0.06893	0.19836
Enclosed.Porch	-0.02287	0.01177	0.02187	-0.14033	0.07146	-0.37436
X3Ssn.Porch	-0.03796	0.02544	0.01624	0.01824	0.04385	0.01580
Screen.Porch	-0.05061	0.07009	0.05504	0.04161	0.04406	-0.04144
Pool.Area	-0.00343	0.16073	0.09378	0.03040	-0.01679	0.00221
Misc.Val	-0.02925	0.03582	0.06919	0.00518	0.03406	-0.01101
Mo.Sold	0.00035	0.01023	0.00386	0.03110	-0.00729	0.01458
Yr.Sold	-0.01791	-0.00696	-0.02309	-0.02072	0.03121	-0.01320
SalePrice	-0.08509	0.34067	0.26655	0.79926	-0.10170	0.55843

Figure 1.16: Correlation Matrix of the numeric features of the dataset.

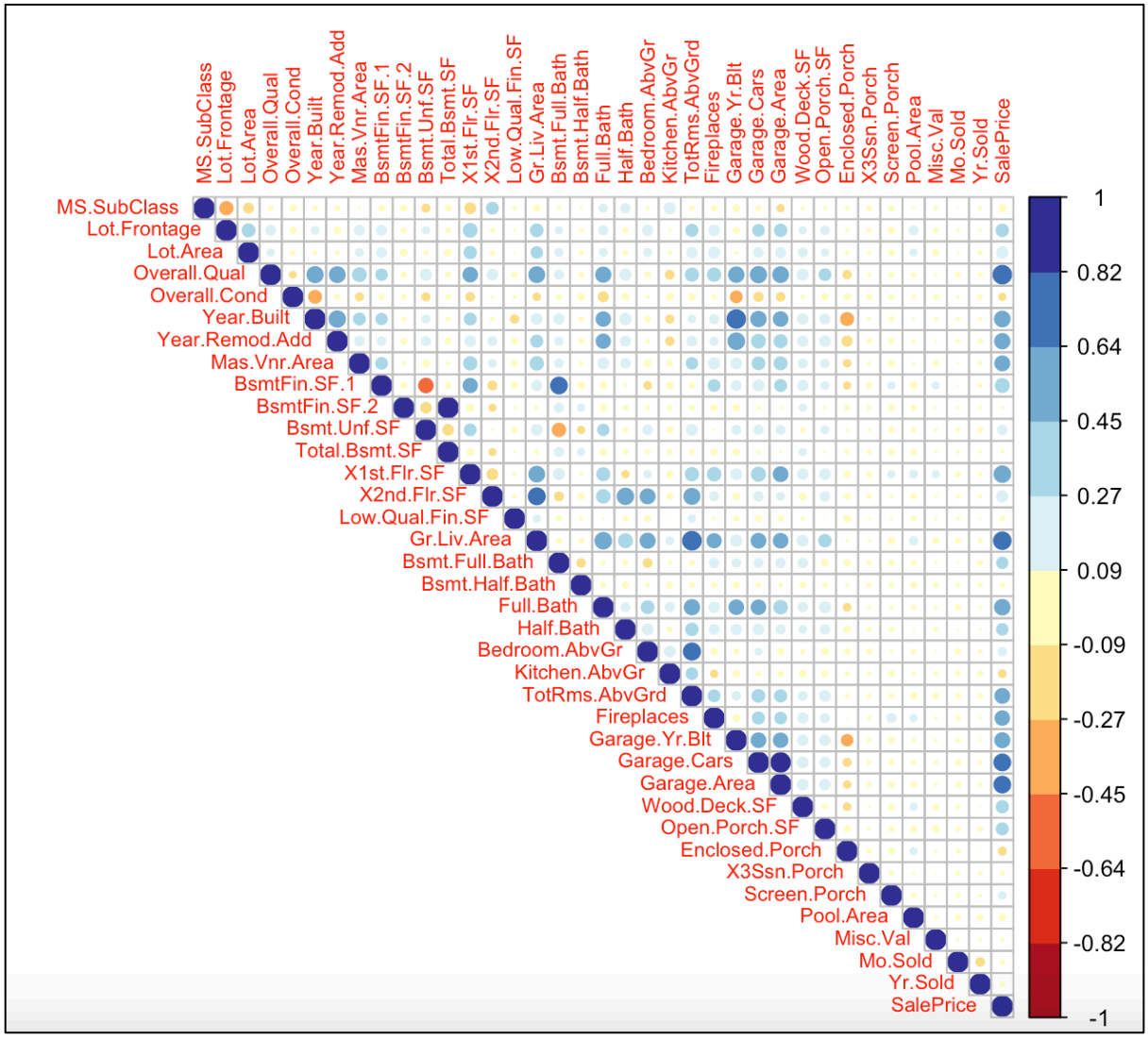

Figure 1.16: Correlation Plot of the features of the dataset.

- Interpretation of Correlation matrix and Correlation Plot :
  - Values closer to |1| (modulus 1) means that the corresponding features are highly correlated to each other.
  - Values closer to |0| (modulus 0) means that the corresponding features are very less correlated to each other.
  - Color coding represents the nature of correlation between the features.
  - Color in the shade of blue represents positive correlation.
  - Color in the shade of red/yellow represents negative correlation.

- From the correlation matrix and correlation plot, we can figure out that the feature *Overall.Qual* is highly correlated to the feature *SalePrice. They are positively correlated.*
- *Garage.Area* is also highly correlated to the *SalePrice* feature. *They are positively correlated.*
- *BsmtFin.SF.2* is highly correlated to the *Bsmt.Unit.SF* feature. *But, in this case, they are negatively correlated.*

Some of the correlation statistics according to the questions asked in the assignment are :

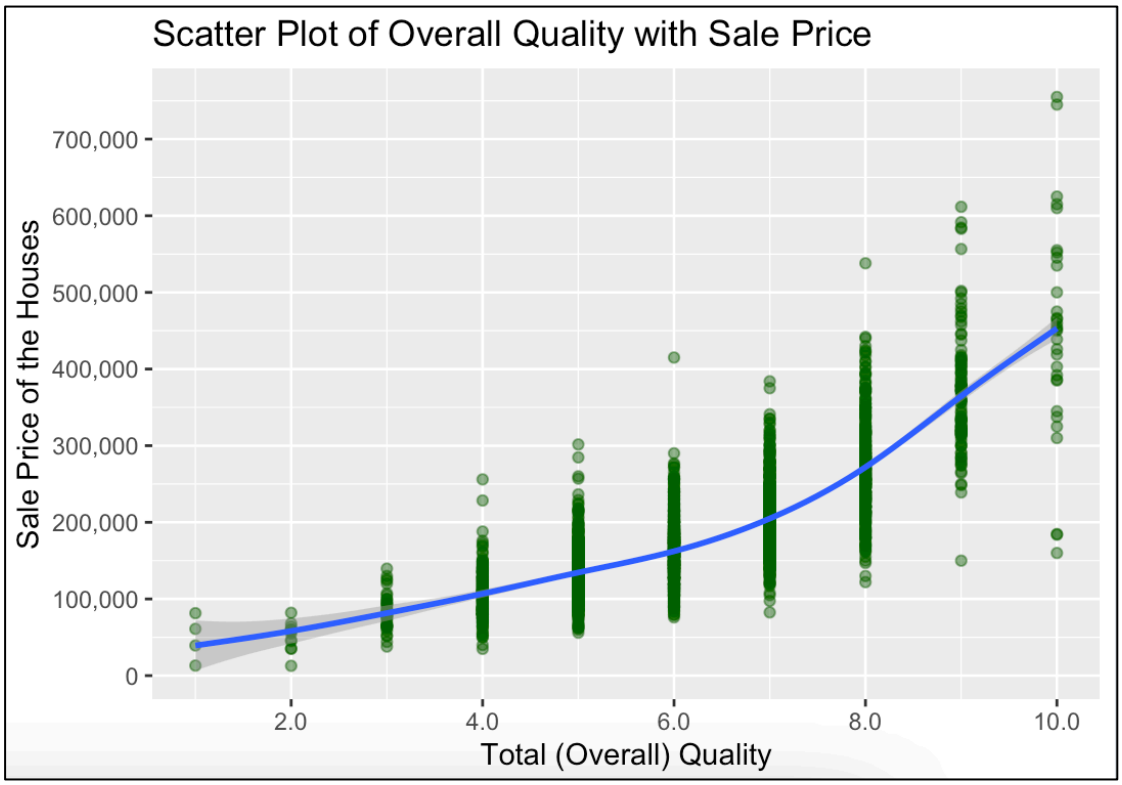

Figure 1.17: Scatterplot of highest correlated feature with dependent variable (Sale Price).

- **Highest Correlated** feature with our dependent feature (Sale Price) is : **Overall.Qual**
  Correlation value: **0.79926**
- There are many outliers at the overall quality of 9 and 10.

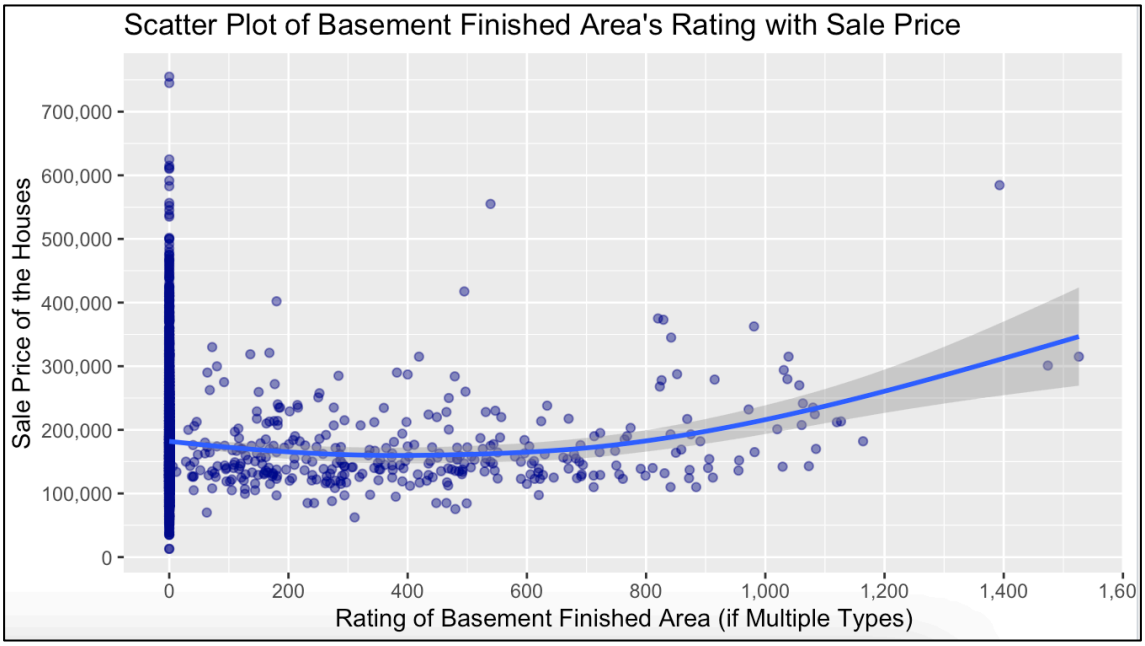

Figure 1.18: Scatterplot of least correlated feature with dependent variable (Sale Price).

- **Least Correlated** feature with our dependent feature (Sale Price) is : **BsmtFin.SF.2**
  Correlation value: **0.00602**
- The values are positively correlated but to a very small extent and the confidence interval increases with the increase of Rating of Basement Finished Area.

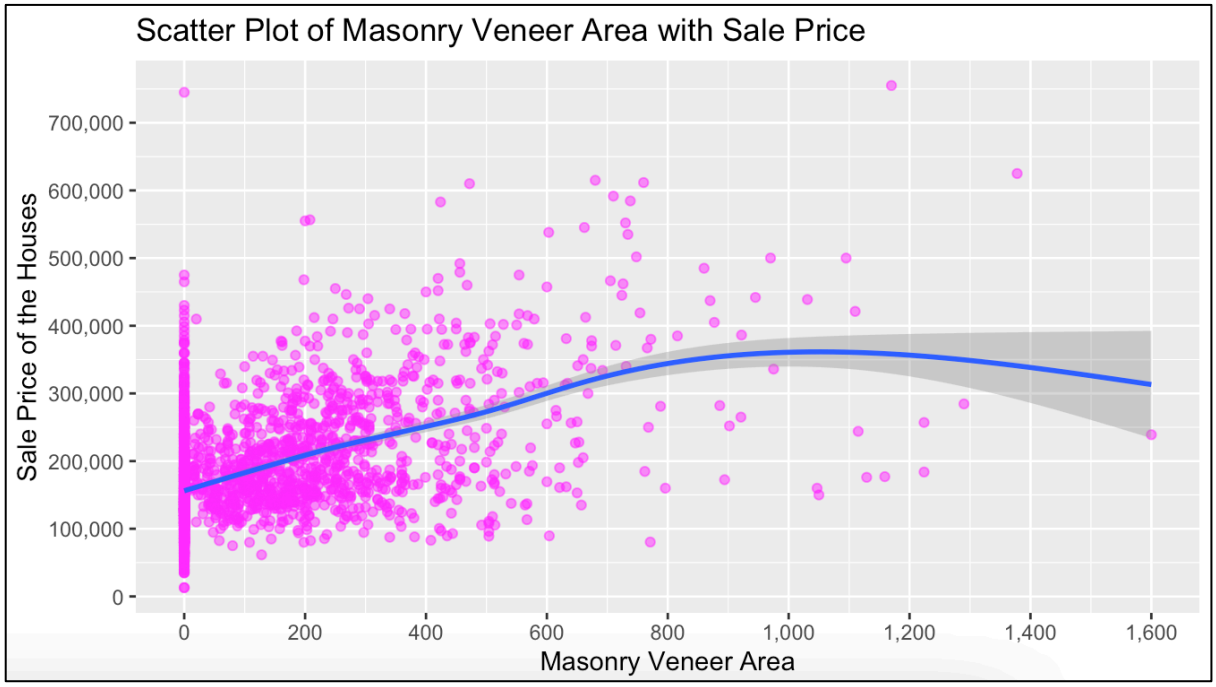

Figure 1.19: Scatterplot of half-way correlated feature with dependent variable (Sale Price).

- **Correlated** feature with **close to 0.5 correlation** with our dependent feature (Sale Price) is : **Mas.Vnr.Area**
  Correlation value: **0.50220**
- Masonry Veneer Area feature increases with 0.5 slope with the increase of Sale Price up to the mark of 800 square feet. But, after that mark the variables show an indifferent behaviour.

REGRESSION MODEL

For this step, we are fitting the regression model using **lm()** function. This part is being performed in various steps to check the features and verify their contribution to the fit of regression model.

For the first fit, I've taken all the 37 numeric features to fit the regression model and checked their contributions.

```
Call:
lm(formula = SalePrice ~ ., data = regressionFittingFeatures)

Residuals:
    Min      1Q  Median      3Q     Max
-544937  -16079   -2172   12892  282790

Coefficients: (2 not defined because of singularities)
                    Estimate   Std. Error t value             Pr(>|t|)
(Intercept)     766295.90056 937647.10206   0.817             0.413851
MS.SubClass       -157.11582     18.01462  -8.722 < 0.0000000000000002
Lot.Frontage        10.89567     35.81466   0.304             0.760979
```

Figure 1.20: Regression Model Fit using all 37 numeric features.

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 32480 on 2895 degrees of freedom
Multiple R-squared:  0.8366,    Adjusted R-squared:  0.8347
F-statistic: 436.1 on 34 and 2895 DF,  p-value: < 0.00000000000000022
```

Figure 1.21: Statistics of the Regression Model Fit using all 37 numeric features.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	766295.90056	937647.10206	0.817	0.413851
MS.SubClass	-157.11582	18.01462	-8.722	< 0.0000000000000002
Lot.Frontage	10.89567	35.81466	0.304	0.760979
Lot.Area	0.45081	0.08673	5.198	0.0000002154755212
Overall.Qual	17240.05772	758.59956	22.726	< 0.0000000000000002
Overall.Cond	4113.04701	669.23124	6.146	0.0000000009040333
Year.Built	261.51150	43.48247	6.014	0.0000000020347793
Year.Remod.Add	186.50208	44.88580	4.155	0.0000334657840888
Mas.Vnr.Area	34.00563	3.99610	8.510	< 0.0000000000000002
BsmtFin.SF.1	26.37397	2.94547	8.954	< 0.0000000000000002
BsmtFin.SF.2	15.08075	4.44744	3.391	0.000706
Bsmt.Unf.SF	11.53785	2.62770	4.391	0.0000116955533027
Total.Bsmt.SF	NA	NA	NA	NA
X1st.Flr.SF	54.84263	3.66756	14.953	< 0.0000000000000002
X2nd.Flr.SF	54.17632	3.32999	16.269	< 0.0000000000000002
Low.Qual.Fin.SF	23.37195	13.47802	1.734	0.083011
Gr.Liv.Area	NA	NA	NA	NA
Bsmt.Full.Bath	6585.88161	1675.87172	3.930	0.0000869968081582
Bsmt.Half.Bath	-1962.96710	2639.46246	-0.744	0.457119
Full.Bath	1784.73822	1820.42909	0.980	0.326973
Half.Bath	-1553.46162	1750.66480	-0.887	0.374961
Bedroom.AbvGr	-8524.52668	1109.61671	-7.682	0.0000000000000212
Kitchen.AbvGr	-12076.84448	3512.91662	-3.438	0.000595
TotRms.AbvGrd	3085.82663	802.70956	3.844	0.000124
Fireplaces	3693.37548	1157.95590	3.190	0.001440
Garage.Yr.Blt	155.38144	43.90927	3.539	0.000408
Garage.Cars	5266.61139	1907.10969	2.762	0.005789
Garage.Area	11.88712	6.64307	1.789	0.073655
Wood.Deck.SF	16.81295	5.26663	3.192	0.001426
Open.Porch.SF	-11.99120	9.84236	-1.218	0.223200
Enclosed.Porch	20.84642	10.41209	2.002	0.045363
X3Ssn.Porch	0.88010	24.08709	0.037	0.970856

Figure 1.22: Statistics of the Regression Model Fit using all 37 numeric features.

- The statistics figure tells us that there are 2 features with **NA** values (example - *Total.Bsmt.SF & Gr.Liv.Area*)
- We can check if these features are show Perfect Multicollinearity using **vif()** and **alias()** function

```
> # Check 'Perfect Multi-Collinearity' because of "NA" values in summary
 of the fit model.
> vif(fit)
Error in vif.default(fit) : there are aliased coefficients in the model
```

Figure 1.23: Checking 'Perfect Multi-Collinearity' of two features with NA values in fit.

- The statistics figure tells us that there are 2 features with **NA** values (example - *Total.Bsmt.SF & Gr.Liv.Area*)
- We can check if these features are show Perfect Multicollinearity using **vif()** and **alias()** function

```
Complete :
              (Intercept) MS.SubClass Lot.Frontage Lot.Area Overall.Qual
Total.Bsmt.SF 0           0          0            0        0
Gr.Liv.Area   0           0          0            0        0
              Overall.Cond Year.Built Year.Remod.Add Mas.Vnr.Area BsmtFin.SF.1
Total.Bsmt.SF 0            0          0              0            0
Gr.Liv.Area   0            0          0              0            0
              BsmtFin.SF.2 Bsmt.Unf.SF X1st.Flr.SF X2nd.Flr.SF Low.Qual.Fin.SF
Total.Bsmt.SF 1            0           0           0           0
Gr.Liv.Area   0            0           1           1           1
              Bsmt.Full.Bath Bsmt.Half.Bath Full.Bath Half.Bath Bedroom.AbvGr
Total.Bsmt.SF 0               0              0         0         0
Gr.Liv.Area   0               0              0         0         0
              Kitchen.AbvGr TotRms.AbvGrd Fireplaces Garage.Yr.Blt Garage.Cars
Total.Bsmt.SF 0             0             0          0             0
Gr.Liv.Area   0             0             0          0             0
              Garage.Area Wood.Deck.SF Open.Porch.SF Enclosed.Porch
Total.Bsmt.SF 0           0            0            0
Gr.Liv.Area   0           0            0            0
              X3Ssn.Porch Screen.Porch Pool.Area Misc.Val Mo.Sold Yr.Sold
Total.Bsmt.SF 0           0           0         0        0       0
Gr.Liv.Area   0           0           0         0        0       0
```

Figure 1.24: Total.Bsmt.SF & Gr.Liv.Area can be derived from other features.

- These features can be derived from other features with the value 1 and therefore, they possess perfect multicollinearity in them.
- We can remove these variables as they don't contribute to the regression model fit.
- We were able to verify and conclude that these features are showing Perfect Multicollinearity in the dataset with the help of **vif()** and **alias()** functions.

```
# First Batch of Variables Removed
regressionFittingFeatures <- regressionFittingFeatures %>%
  dplyr::select(-Total.Bsmt.SF, -Gr.Liv.Area)
fit <- lm(formula = SalePrice ~ ., data = regressionFittingFeatures)
summary(fit)

# Second Batch of Variables Removed
regressionFittingFeatures <- regressionFittingFeatures %>%
  dplyr::select(-Lot.Frontage, -X3Ssn.Porch, -Mo.Sold)
fit <- lm(formula = SalePrice ~ ., data = regressionFittingFeatures)
summary(fit)

# Third Batch of Variables Removed
regressionFittingFeatures <- regressionFittingFeatures %>%
  dplyr::select(-Bsmt.Half.Bath, -Half.Bath)
fit <- lm(formula = SalePrice ~ ., data = regressionFittingFeatures)
summary(fit)
```

Figure 1.25: Code-Script to remove insignificant features from the regression model fitting.

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 32460 on 2900 degrees of freedom
Multiple R-squared:  0.8366,    Adjusted R-squared:  0.8349
F-statistic: 511.9 on 29 and 2900 DF,  p-value: < 0.00000000000000022
```

Figure 1.26: Statistics of the regression model fitting after elimination.

- Features with p-value greater than 0.5 (ex- *Lot.Frontage*, *X3Ssn.Porch*) do not contribute to the regression model and can be eliminated from the model.
- The Adjusted R-squared value has also increased when insignificant features have been eliminated from the regression model fitting.
- The new Adjusted R-squared value is **0.8349**

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 39200 on 2924 degrees of freedom
Multiple R-squared:  0.7596,    Adjusted R-squared:  0.7592
F-statistic:  1848 on 5 and 2924 DF,  p-value: < 0.00000000000000022
```

Figure 1.27: Statistics of the regression model fitting with 5 highly correlated features..

- Simple Linear regression model fit using top 5 highly correlated variables (*Overall.Qual, Gr.Liv.Area, Garage.Cars, Garage.Area, Total.Bsmt.SF*) gives Adjusted R-squared value of 0.759 which is less than what we've got from the above selection of features.
- For now, we will proceed with the previous selection of features with high Adjusted R-Squared value.

REGRESSION EQUATION

Y = 830620.2133
*+ (-159.8296) * MS.SubClass + (0.4533) * Lot.Area + (17263.4249) * Overall.Qual*
*+ (4086.3071) * Overall.Cond + (250.6693) * Year.Built +* ...

REGRESSION PLOT

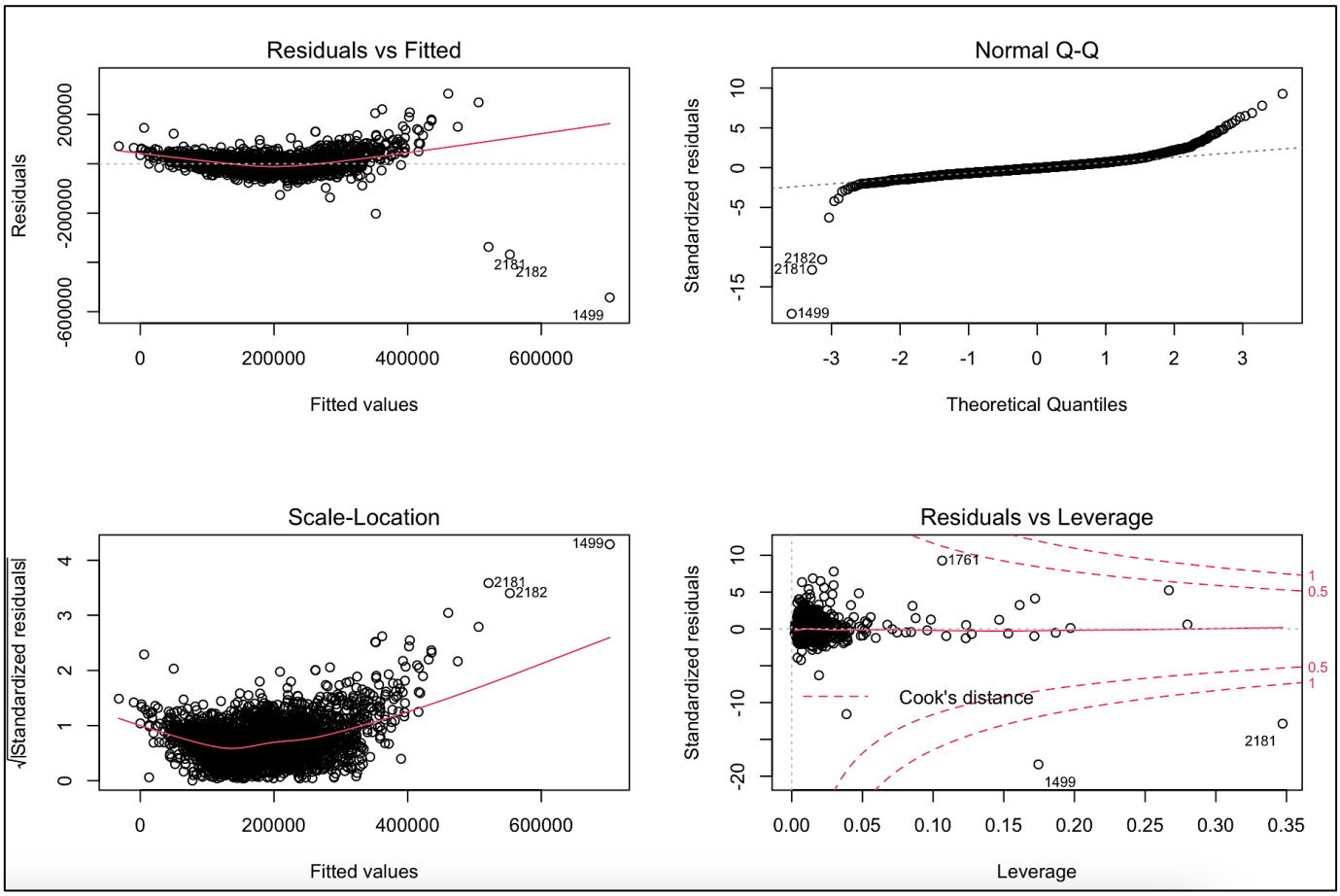

Figure 1.28: Regression Model Plot.

- *Residuals vs Fitted* -
  - This graph is used to check the linearity of the dataset.
  - We can see there is a fan pattern in the data points.
  - This could be because either the dependent variable might be non-linear or some of the independent variables.
  - This shows that there exists a non-linearity
  - We can also look at the outliers marked with their index value in the graph.

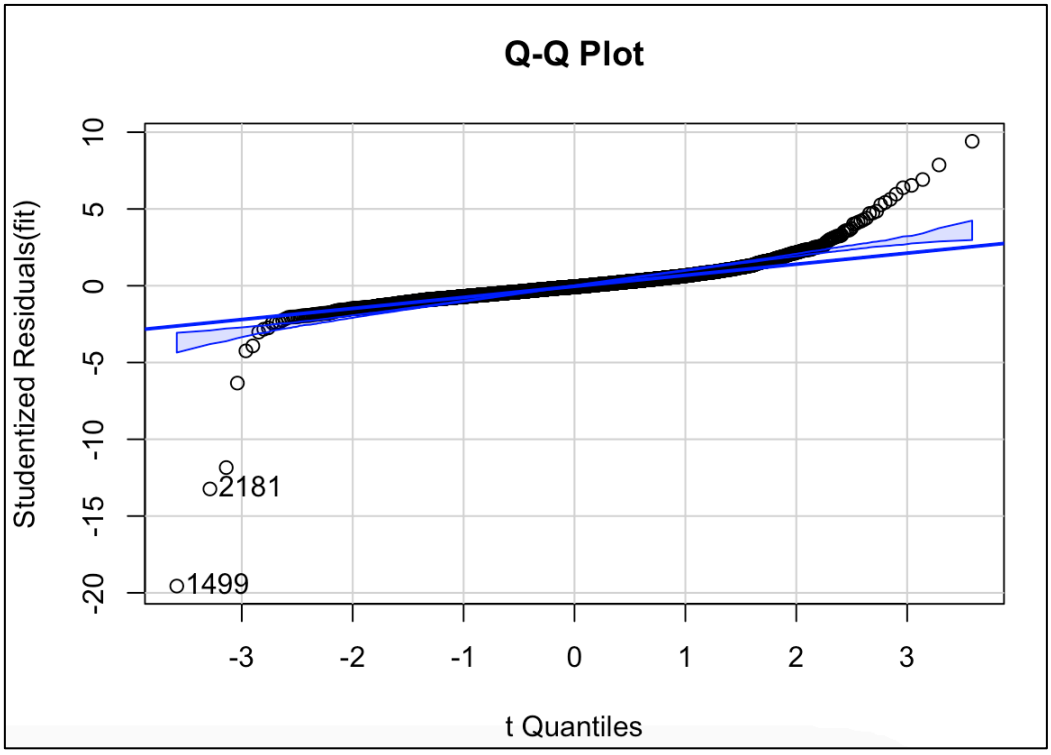

Figure 1.29: Q-Q Plot.

- *Q-Q Plot* -
  - The data points lie with the regression line and there exists many potential outliers which might affect the normality of the data set.
  - We can also look at the definite outliers marked with their index value in the graph.

- *Scale-Location* -
  - This graph is used to find the homoscedasticity of the dataset.
  - There exists a distinctive pattern in these observations and hence the dataset is not homoscedastic.
  - We can also look at the definite outliers marked with their index value in the graph.

- *Residuals vs Leverage* -
  - This graph is used to identify the unusual observations in the dataset.
  - We can find some of the observations with either very high or very low residuals which are marked using their index values.

OUTLIER TEST & REMOVAL

We will test the dataset for Outliers using **outlierTest**() function. Further, we will remove outlier from the dataset which were found from the multicollinearity test and outlier test. We will also remove outliers found from the Cook's distance formula and plot.

```
> # Check for Unusual Observations
> outlierTest(model = fit)
       rstudent
1499 -19.560428
2181 -13.234356
2182 -11.843046
1761   9.406230
1768   7.866392
434    6.917335
45     6.535806
2333   6.384866
1183  -6.339631
1638   5.965642
```

Figure 1.30: Outlier Test to find out the outliers on every iteration.

```
# Remove Outlier Records from the dataset (1st Round)
regressionFittingFeatures <- regressionFittingFeatures %>%
  dplyr::filter(!row_number() %in% c(2181,2182,1499,1761,1768,
                                     434,45,2333,1183,1638))

# Remove Outlier Records from the dataset (2nd Round)
regressionFittingFeatures <- regressionFittingFeatures %>%
  dplyr::filter(!row_number() %in% c(1062,432,2441,2322,1776,
                                     2436,2325,1636,423,372))

# Remove Outlier Records from the dataset (3rd, 4th, 5th, and 6th Round)
regressionFittingFeatures <- regressionFittingFeatures %>%
  dplyr::filter(!row_number() %in% c(1552,2324,452,16,2647,1682,1690,1633, 2358,
                                     420,1896, 1681, 1676, 135, 2220, 429,2218,428,
                                     1890,1672,1675,364,2223,2695,1677,2839,1598))
```

Figure 1.31: Outliers removal from the dataset and used for regression model fitting.

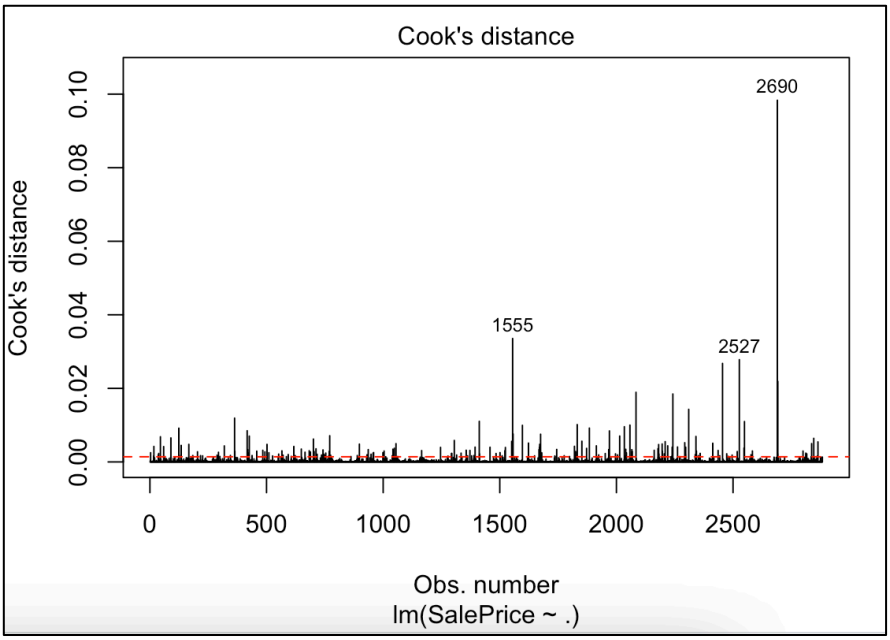

Figure 1.32: Outliers identification using Cook's Distance method.

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 24150 on 2853 degrees of freedom
Multiple R-squared:  0.8897,    Adjusted R-squared:  0.8885
F-statistic: 793.2 on 29 and 2853 DF,  p-value: < 0.00000000000000022
```

Figure 1.32: Regression Model Fit after removal of Outliers.

- The Adjusted R-Squared value gets increased to 0.8885 after removal of outliers from the dataset.
- Outliers are eliminated incrementally after each iteration of outlier test.

STEPWISE SELECTION *(FEATURE SELECTION)*

We will use the Stepwise method to find out the best possible subset of features with the best possible optimization of the regression model fit. We'll use 3 types of Stepwise selection method as follows:

- Forward Stepwise Selection
- Backward Stepwise Selection
- Stepwise Stepwise Selection (both)

From the observations we can figure out that the Stepwise Stepwise Selection gives the best possible subset of features

```
Call:
lm(formula = SalePrice ~ MS.SubClass + Lot.Area + Overall.Qual +
    Overall.Cond + Year.Built + Year.Remod.Add + Mas.Vnr.Area +
    BsmtFin.SF.1 + BsmtFin.SF.2 + Bsmt.Unf.SF + X1st.Flr.SF +
    X2nd.Flr.SF + Low.Qual.Fin.SF + Bsmt.Full.Bath + Bedroom.AbvGr +
    Kitchen.AbvGr + TotRms.AbvGrd + Fireplaces + Garage.Yr.Blt +
    Garage.Area + Wood.Deck.SF + Enclosed.Porch + Screen.Porch +
    Yr.Sold, data = regressionFittingFeatures)

Coefficients:
    (Intercept)      MS.SubClass         Lot.Area     Overall.Qual     Overall.Cond
    382289.3613        -128.4687           0.5277       14562.2633        4282.6512
     Year.Built   Year.Remod.Add     Mas.Vnr.Area     BsmtFin.SF.1     BsmtFin.SF.2
       289.8544         198.9693          29.2809          37.2193          19.4876
    Bsmt.Unf.SF      X1st.Flr.SF      X2nd.Flr.SF  Low.Qual.Fin.SF   Bsmt.Full.Bath
        18.1402          62.1626          61.0018          15.6090        3283.2174
  Bedroom.AbvGr    Kitchen.AbvGr    TotRms.AbvGrd       Fireplaces    Garage.Yr.Blt
    -10177.7922      -13525.2001        2477.9359        2814.2911         128.4900
    Garage.Area     Wood.Deck.SF   Enclosed.Porch     Screen.Porch          Yr.Sold
        27.0118           7.4578          13.7564          16.2161         -819.9027
```

Figure 1.33: Best Subset of features from Stepwise Stepwise Selection.

BEST SUBSET SELECTION *(FEATURE SELECTION)*

We will use the Best Subset method to find out the best possible subset of features with the best possible optimization of the regression model fit.

From the observations we can figure out that the Stepwise Stepwise Selection gives the best possible subset of features

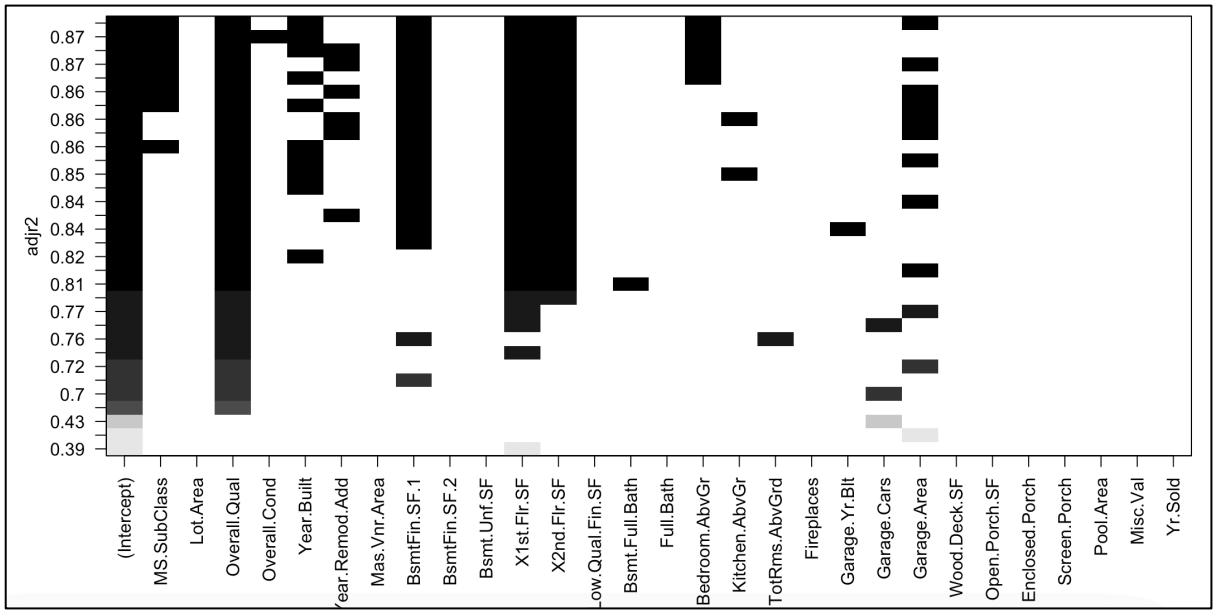

Figure 1.34: Best Subset method to find out best features from Stepwise Stepwise Selection.

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 31500 on 2873 degrees of freedom
Multiple R-squared:  0.811,     Adjusted R-squared:  0.8104
F-statistic:  1370 on 9 and 2873 DF,  p-value: < 0.00000000000000022
```

Figure 1.35: Best Subset method statistics

- The Adjusted R-Squared value is a healthy value of **0.8104**
- The total number of features used to fit the regression model is also significantly less than the other methods used to fit the model.
- If we compare the subset selection methods,
    - Even though Adjusted R-Squared value of the model from Stepwise Selection methods are higher, but they also use very large number of features.
    - The Best Subset method can give a similar and a healthy value of Adjusted R-Squared value of **0.8104** with a very less number of features.

REGRESSION EQUATION

$$Y = -1155168.715 + (-51.016) * MS.SubClass + (424.303) * Year.Remod.Add + (25250.478) * Overall.Qual + (1179.602) * Overall.Cond + (112.705) * Year.Built + (26.712) * BsmtFin.SF.1 + (41.528) * X1st.Flr.SF + (8863.104) * Bedroom.AbvGr + (59.113) * Garage.Area + 31500$$

CONCLUSION

We have performed the *Regression Diagnostics* to analyse the Ames Housing dataset which contains data points related to houses with their Sale Price as a dependent variable. We found the below observations and conclusions from the analysis performed in this project.

- The dataset has 2,930 data points with 82 Features belonging to the attributes of houses.
- The **mean** of the attribute *SalePrice* is around **180796.06** with a **standard deviation** of **79886.69** and **quartiles value (Lower Quartile - 129500, Higher Quartile - 213500).**
  From the observations and further analysis, we can anticipate that the data points around **maximum value (755000)** are be *outliers*.

- The Sale Price feature is Right-Skewed in nature and the majority of houses have their sale prices around $130,000 - $220,000 belonging to the Quality of 5 or 6.

- The feature *Overall.Qual* is highly positively correlated to the feature *SalePrice.*
- The feature *BsmtFin.SF.2* is the least positively correlated to the feature *SalePrice.*

- The features *Total.Bsmt.SF & Gr.Liv.Area* are showing Perfect Multicollinearity which was proved using **vif()** and **alias()** function.
- Features with p-value greater than 0.5 (ex- *Lot.Frontage*, *X3Ssn.Porch*) did not contribute to the regression model and were eliminated from the model

- We found various outliers in the dataset which the assumptions of OLS test ascertained. The outliers found in the plots of Linearity, Normality, Homoscedasticity, and Unusual Observations were investigated and dealt accordingly. if they were outliers, we removed them from the dataset.
- We performed Outlier Tests (iteratively, eliminating the outlier records from the dataset as well).
- We performed Cook's Distance method to find out other outliers using it and eliminated them.

- Afterwards, we utilized the Stepwise Selection method to find out the best possible subset of features using all the 3 ways of Stepwise Selection method.
- ***The Stepwise Stepwise Selection method gave out the best subset of features with 0.8885 Adjusted R-squared value.***

- ***But, the Best Subset method gave out its best subset of features with 0.8104 Adjusted R-Squared value.***
- ***This method was able to give such a healthy value with very less number of features in comparison to the Stepwise Stepwise selection method.***

BIBLIOGRAPHY

1. *Home - RDocumentation*. (2021). Functions in R - Documentation. https://www.rdocumentation.org/

2. ALY 6015 - Prof Roy Wada - *Lesson 1-1 — Linear Regression* (2022, February), https://northeastern.instructure.com/courses/98028/pages/lesson-1-1-linear-regression?module_item_id=6646913

3. ALY 6015 - Prof Roy Wada - *Lesson 1-3 — Multicollinearity Problem and Evaluation* (2022, February), https://northeastern.instructure.com/courses/98028/pages/lesson-1-3-multicollinearity-problem-and-evaluation?module_item_id=6646918

4. ALY 6015 - Prof Roy Wada - *Lesson 1-5 — Model Comparison Metrics* (2022, February), https://northeastern.instructure.com/courses/98028/pages/lesson-1-5-model-comparison-metrics?module_item_id=6646924

5. ALY 6015 - Prof Roy Wada - *Lesson 1-6 — Feature/Variable Selection* (2022, February), https://northeastern.instructure.com/courses/98028/pages/lesson-1-6-feature-slash-variable-selection?module_item_id=6646926

6. Brownlee, J. (2020, August 15). *Linear Regression for Machine Learning*. Machine Learning Mastery. https://machinelearningmastery.com/linear-regression-for-machine-learning

7. *Increase number of axis ticks*. (2012, July 4). Stack Overflow. Retrieved 2022, from https://stackoverflow.com/questions/11335836/increase-number-of-axis-ticks

APPENDIX

```
#-------- ALY6015_M1_RegressionDiagnostics_HarshitGaur --------#

print("Author : Harshit Gaur")
print("ALY 6015 Week 1 Assignment - Regression Diagnostics")

# Declaring the names of packages to be imported
packageList <- c("tidyverse", "vtable", "RColorBrewer", "corrplot", "car", "MASS", "leaps", "psych")

for (package in packageList) {
  if (!package %in% rownames(installed.packages()))
  { install.packages(package) }

  # Import the package
  library(package, character.only = TRUE)
}

# Import/Load the 'Ames Housing' data set
AmesHousing <- read.csv("~/Documents/Northeastern University/MPS Analytics/ALY 6015/Class
1/Assignment/AmesHousing.csv")

# Get a Glimpse/View of the data set
glimpse(AmesHousing)

######################################################################
# Exploratory Data Analysis
######################################################################

# Summary in tabular format of the data set
st(AmesHousing, title = "Ames Housing Summary Statistics", add.median = TRUE)
options(scipen = 999)
View(describe(AmesHousing, IQR = TRUE, quant = c(0.25, 0.75), trim = 2))

# Removing 'Identifier' variables from the data set
AmesHousing <- AmesHousing %>%
  dplyr::select(-Order, -PID)


# Histogram for the SalePrice variable of the data set
AmesHousing %>%
  ggplot(aes(SalePrice)) +
  geom_histogram(color = "PINK", fill = "DARKBLUE") +
  scale_x_continuous(labels = scales::comma, breaks = scales::pretty_breaks(n=7)) +
  labs(title = "Right Skewed with majority of Houses with Sale Price around $130,000 - $220,000",
       x = "Sale Price of the Houses",
       y = "Count of the Houses")

# Histogram for the SalePrice variable of the data set color-separated by
AmesHousing %>%
  ggplot(aes(SalePrice, fill = as.factor(Overall.Qual))) +
  geom_histogram(color = "BLACK", alpha = 0.5) +
  scale_x_continuous(labels = scales::comma, breaks = scales::pretty_breaks(n=7)) +
  labs(title = "Majority of the Houses with Overall-Quality at 5 or 6 have Sale Price around $130,000 -
$220,000",
       x = "Sale Price of the Houses",
       y = "Count of the Houses",
       fill = "Overall Quality of House") +
  theme(legend.position = "bottom") +
  scale_fill_brewer(palette = "Dark2")
```

```
# Box Plot for the Gr.Liv.Area vs SalePrice variables of the data set
AmesHousing %>%
 ggplot(aes(x = Gr.Liv.Area, y = SalePrice, group = MS.Zoning, color = MS.Zoning)) +
 geom_boxplot(alpha = 0.5) +
 scale_y_continuous(labels = scales::comma, breaks = scales::pretty_breaks(n=7)) +
 labs(title = "Box Plot to check for Outliers",
     x = "Ground Living Area",
     y = "Sale Price of the Houses")

#####################################################################
# Data Imputation
#####################################################################

# Checking the records with missing/NA values
AmesHousing %>%
 filter(!complete.cases(AmesHousing)) %>%
 View()

# Summing up the records with missing/NA values according to the Variables respectively
colSums(is.na(AmesHousing))

# Checking records for the below variables which are found to belong to "No -(attribute)" records.
View(AmesHousing[which(is.na(AmesHousing$Mas.Vnr.Area)),])
View(AmesHousing[which(is.na(AmesHousing$BsmtFin.SF.1)),])
View(AmesHousing[which(is.na(AmesHousing$BsmtFin.SF.2)),])
View(AmesHousing[which(is.na(AmesHousing$Bsmt.Unf.SF)),])
View(AmesHousing[which(is.na(AmesHousing$Total.Bsmt.SF)),])
View(AmesHousing[which(is.na(AmesHousing$Bsmt.Half.Bath)),])
View(AmesHousing[which(is.na(AmesHousing$Bsmt.Full.Bath)),])
View(AmesHousing[which(is.na(AmesHousing$Garage.Area)),])

AmesHousing <- AmesHousing %>%
 mutate(Alley = replace_na(Alley, "No Alley")) %>%
 mutate(Lot.Frontage = replace_na(Lot.Frontage, as.integer(mean(Lot.Frontage, na.rm = TRUE)))) %>%
 mutate(Mas.Vnr.Area = replace_na(Mas.Vnr.Area, 0)) %>%
 mutate(Bsmt.Qual = replace_na(Bsmt.Qual, "No Basement")) %>%
 mutate(Bsmt.Cond = replace_na(Bsmt.Cond, "No Basement")) %>%
 mutate(Bsmt.Exposure = replace_na(Bsmt.Exposure, "No Basement")) %>%
 mutate(BsmtFin.Type.1 = replace_na(BsmtFin.Type.1, "No Basement")) %>%
 mutate(BsmtFin.SF.1 = replace_na(BsmtFin.SF.1, 0)) %>%
 mutate(BsmtFin.Type.2 = replace_na(BsmtFin.Type.2, "No Basement")) %>%
 mutate(BsmtFin.SF.2 = replace_na(BsmtFin.SF.2, 0)) %>%
 mutate(Bsmt.Unf.SF = replace_na(Bsmt.Unf.SF, 0)) %>%
 mutate(Total.Bsmt.SF = replace_na(BsmtFin.SF.2, 0)) %>%
 mutate(Bsmt.Half.Bath = replace_na(Bsmt.Half.Bath, 0)) %>%
 mutate(Bsmt.Full.Bath = replace_na(Bsmt.Full.Bath, 0)) %>%
 mutate(Fireplace.Qu = replace_na(Fireplace.Qu, "No Fireplace")) %>%
 mutate(Garage.Type = replace_na(Garage.Type, "No Garage")) %>%
 mutate(Garage.Yr.Blt = replace_na(Garage.Yr.Blt, as.integer(mean(Garage.Yr.Blt, na.rm = TRUE)))) %>%
 mutate(Garage.Finish = replace_na(Garage.Finish, "No Garage")) %>%
 mutate(Garage.Cars = replace_na(Garage.Cars, 0)) %>%
 mutate(Garage.Area = replace_na(Garage.Area, 0)) %>%
 mutate(Garage.Qual = replace_na(Garage.Qual, "No Garage")) %>%
 mutate(Garage.Cond = replace_na(Garage.Cond, "No Garage")) %>%
 mutate(Pool.QC = replace_na(Pool.QC, "No Pool")) %>%
 mutate(Fence = replace_na(Fence, "No Fence")) %>%
 mutate(Misc.Feature = replace_na(Misc.Feature, "None"))

# Checking the records with missing/NA values
AmesHousing %>%
 filter(!complete.cases(AmesHousing)) %>%
 View()
```

```
# Checking variables with 'Quality Assessment Abbreviated Text' in them
unique(AmesHousing$Bsmt.Qual)
unique(AmesHousing$Kitchen.Qual)
unique(AmesHousing$Overall.Qual)

# Factorize these 'Quality Assessment Texts' in the data set
AmesHousing[sapply(AmesHousing, is.character)] <- lapply(AmesHousing[sapply(AmesHousing, is.character)],
as.factor)


#####################################################################
# Correlation
#####################################################################

numIntFeatures_AmesHousing <- AmesHousing[sapply(AmesHousing, is.numeric)]
View(round(cor(numIntFeatures_AmesHousing, use = "pairwise"), 5))
corrplot(cor(numIntFeatures_AmesHousing, use = "pairwise"), tl.cex = 0.7, type = "upper",
       title = "Correlation Plot", mar = c(0,0,1,0),
       col = brewer.pal(n = ncol(numIntFeatures_AmesHousing), name = "RdYlBu"))


#####################################################################
# Scatterplots - Question 6
#####################################################################

# Scatterplot for the Total (Overall) Quality vs SalePrice variables of the data set - Highest Correlation
# Correlation Value: 0.79926
AmesHousing %>%
 ggplot(aes(x = Overall.Qual, y = SalePrice)) +
 geom_point(color = "DARKGREEN", alpha = 0.5) +
 scale_x_continuous(labels = scales::comma, breaks = scales::pretty_breaks(n=7)) +
 scale_y_continuous(labels = scales::comma, breaks = scales::pretty_breaks(n=7)) +
 labs(title = "Scatter Plot of Overall Quality with Sale Price ",
     x = "Total (Overall) Quality",
     y = "Sale Price of the Houses") +
 geom_smooth()

# Scatterplot for the Rating of Basement Finished Area (2) vs SalePrice variables of the data set - Lowest
Correlation
# Correlation Value: 0.00602
AmesHousing %>%
 ggplot(aes(x = BsmtFin.SF.2, y = SalePrice)) +
 geom_point(color = "DARKBLUE", alpha = 0.5) +
 scale_x_continuous(labels = scales::comma, breaks = scales::pretty_breaks(n=8)) +
 scale_y_continuous(labels = scales::comma, breaks = scales::pretty_breaks(n=7)) +
 labs(title = "Scatter Plot of Basement Finished Area's Rating with Sale Price",
     x = "Rating of Basement Finished Area (if Multiple Types)",
     y = "Sale Price of the Houses") +
 geom_smooth()

# Scatterplot for the Masonry Veneer Area vs SalePrice variables of the data set - Closest 0.5 Correlation
# Correlation Value: 0.50220
AmesHousing %>%
 ggplot(aes(x = Mas.Vnr.Area, y = SalePrice)) +
 geom_point(color = "MAGENTA", alpha = 0.5) +
 scale_x_continuous(labels = scales::comma, breaks = scales::pretty_breaks(n=10)) +
 scale_y_continuous(labels = scales::comma, breaks = scales::pretty_breaks(n=7)) +
 labs(title = "Scatter Plot of Masonry Veneer Area with Sale Price",
     x = "Masonry Veneer Area",
     y = "Sale Price of the Houses") +
 geom_smooth()
```

```
#####################################################################
# Regression Fit - Model the data
#####################################################################

regressionFittingFeatures <- numIntFeatures_AmesHousing
fit <- lm(formula = SalePrice ~ ., data = regressionFittingFeatures)
summary(fit)

# Check 'Perfect Multi-Collinearity' because of "NA" values in summary of the fit model.
vif(fit)
alias(fit)

# First Batch of Variables Removed
regressionFittingFeatures <- regressionFittingFeatures %>%
  dplyr::select(-Total.Bsmt.SF, -Gr.Liv.Area)
fit <- lm(formula = SalePrice ~ ., data = regressionFittingFeatures)
summary(fit)

# Second Batch of Variables Removed
regressionFittingFeatures <- regressionFittingFeatures %>%
  dplyr::select(-Lot.Frontage, -X3Ssn.Porch, -Mo.Sold)
fit <- lm(formula = SalePrice ~ ., data = regressionFittingFeatures)
summary(fit)

# Third Batch of Variables Removed
regressionFittingFeatures <- regressionFittingFeatures %>%
  dplyr::select(-Bsmt.Half.Bath, -Half.Bath)
fit <- lm(formula = SalePrice ~ ., data = regressionFittingFeatures)
summary(fit)

# Regression Fit with Top 5 Most Correlated Variables with Sale Price.
fit <- lm(formula = SalePrice ~
        Overall.Qual + Gr.Liv.Area + Garage.Cars + Garage.Area + Total.Bsmt.SF, data = AmesHousing)
summary(fit)

# Akaike Information Criteria
AIC(fit)

# Bayesian Information Criteria
BIC(fit)

#####################################################################
# Review Diagnostic Plots
#####################################################################
# Residual vs Fitted - Linearity
# Normal Q-Q Plot - Normality
# Scale ~ Location - Homoscedasticity (Constant Variance)
# Residuals vs Leverage - Unusual Observations
par(mfrow = c(2,2))
plot(fit)
dev.off()

## Individual Plots ##
# Q-Q Plot
qqPlot(fit, labels = rownames(regressionFittingFeatures$SalePrice), simulate = TRUE, main = "Q-Q Plot")

# Components + Residual - Linearity
crPlots(model = fit)

# Spread-Level Plot for fit - Homoscedasticity
spreadLevelPlot(fit)
```

```
#####################################################################
# Multi-Collinearity
#####################################################################
# Check for Multi-Collinearity
vif(fit)

#####################################################################
# Outlier Investigation and Elimination
#####################################################################

# Cook's Distance for Outliers (Influential Observations)
cutoff <- 4 / (nrow(regressionFittingFeatures) - length(fit$coefficients) - 2)
plot(fit, which = 4, cook.levels = cutoff)
abline(h = cutoff, lty = 2, col = "RED")

# Regression Model Fitting (For Outlier Removal)
fit <- lm(formula = SalePrice ~ ., data = regressionFittingFeatures)
summary(fit)

# Check for Unusual Observations
outlierTest(model = fit)

# Remove Outlier Records from the dataset (1st Round)
regressionFittingFeatures <- regressionFittingFeatures %>%
  dplyr::filter(!row_number() %in% c(2181,2182,1499,1761,1768,
                    434,45,2333,1183,1638))

# Remove Outlier Records from the dataset (2nd Round)
regressionFittingFeatures <- regressionFittingFeatures %>%
  dplyr::filter(!row_number() %in% c(1062,432,2441,2322,1776,
                    2436,2325,1636,423,372))

# Remove Outlier Records from the dataset (3rd, 4th, 5th, and 6th Round)
regressionFittingFeatures <- regressionFittingFeatures %>%
  dplyr::filter(!row_number() %in% c(1552,2324,452,16,2647,1682,1690,1633,
                    2358,420,1896,1681,1676,135,2220,429,2218,428,
                    1890,1672,1675,364,2223,2695,1677,2839,1598))


#####################################################################
# Feature Selection
#####################################################################

# Backward Stepwise Selection
stepAIC(fit, direction = "backward")

# Forward Stepwise Selection
stepAIC(fit, direction = "forward")

# Stepwise Stepwise Selection
stepAIC(fit, direction = "both")

# Regression Model Fit using Features from Stepwise Stepwise Selection
fit <- lm(formula = SalePrice ~ MS.SubClass + Lot.Area + Overall.Qual +
   Overall.Cond + Year.Built + Year.Remod.Add + Mas.Vnr.Area +
   BsmtFin.SF.1 + BsmtFin.SF.2 + Bsmt.Unf.SF + X1st.Flr.SF +
   X2nd.Flr.SF + Low.Qual.Fin.SF + Bsmt.Full.Bath + Bedroom.AbvGr +
   Kitchen.AbvGr + TotRms.AbvGrd + Fireplaces + Garage.Yr.Blt +
   Garage.Area + Wood.Deck.SF + Enclosed.Porch + Screen.Porch +
   Yr.Sold,
   data = regressionFittingFeatures)
summary(fit)
```

```
# Best Subset Method
leaps <- regsubsets(SalePrice ~ ., data = regressionFittingFeatures, nbest = 4)
summary(leaps)
plot(leaps, scale = "adjr2")

# Regression Model Fit using Features from Best Subset Selection
fit <- lm(formula = SalePrice ~ MS.SubClass + Year.Remod.Add +
        Overall.Qual + Overall.Cond + Year.Built +
        BsmtFin.SF.1 + X1st.Flr.SF + Bedroom.AbvGr + Garage.Area,
        data = regressionFittingFeatures)
summary(fit)
vif(fit)

par(mfrow = c(2,2))
plot(fit)
dev.off()

#-------- END --------#
```