Understanding and Predicting the Sale Price of Houses in Ames

Submitted By

Prayas Sachdeva

Introduction and Background

Today, many individuals engage in real estate since it may occasionally provide a substantial amount of capital income, a phenomenon that has evolved rapidly in our nation. Investment in real estate reflects, to some extent, the local real estate development condition. For instance, if investment is booming, it indicates that the development is imbalanced, and supply is limited. If an investment is frigid, the real estate market has been steady as of late. Moreover, it reflects a city's progress (Cao & Yang, 2018).

Technology has been propelled forward by data, and it is now possible to forecast any outcome. This strategy relies heavily on machine learning. Machine learning involves providing a trustworthy dataset and basing predictions on it; the machine itself learns how relevant a specific event may be to the entire system based on its pre-loaded data and forecasts the outcome properly. Numerous recent applications of this technique include forecasting stock prices, estimating the likelihood of an earthquake, and forecasting corporation sales, to name a few (Kauko, Hooimeijer et al. 2002).

Hypothesis

Number	Variable	Relationship
H1	Overall Quality and Sale Price (T.L.Ooi et al., 2014)	Positive
H2	Year Built and Sale Price (Green et al., 1999)	Positive
H3	Year Remodelling Added and Sale Price (Wilhelmsson et al., 2007)	Positive
H4	Total Sqft of Basement Area and Sale Price (Palmquist, 1980)	Positive
H5	Garage Cars and Sale Price (Ebru & Eban , 2009)	Positive

Methodology

1. Business Understanding

The objective of the report is to explore the data set consisting of the information for the house sold in Ames, Iowa from 2006 to 2010 and understand it to formulate a predictive model using the R language to predict the house prices. To achieve the objective an analytics base table should be created selecting the input variables that are likely to influence the sale prices of the house followed by identifying and eliminating any data quality issues that could weaken the accuracy of the model. The cleaned data should then be split into train and test data sets to formulate the linear regression model and finally use the assumptions testing methods on our final model for further assessment.

2. Data Understanding

The data is collected from the Assessor's office in Iowa and is available in the .xlsx format with 80 variables and 2897 observations with 1 variable pertaining to the Identification numbers. The variables are a mix of various data types, specifically 23 nominal, 23 ordinal, 14 discrete and 20 continuous. Using the summary() function of the R, it is clear that the data has numerous quality issues like outliers, missing values, out of range observation etc. These are addressed in the following sub section for further clarification.

3. Data Preparation

Firstly, an analytics base table (ABT) is created by selecting 17 variables out of previously mentioned 80 variables. The variable selected are assumed to be of an influence on the predetermined dependent variable, i.e, Sale Price.

```
ABT                    2896 obs. of 17 variables
  $ Lot.Frontage  : num [1:2896] 141 80 81 93 74 78 41 43 39 60 ...
  $ Lot.Area      : num [1:2896] 31770 11622 14267 11160 13830 ...
  $ Street        : chr [1:2896] "Pave" "Pave" "Pave" "Pave" ...
  $ Land.Contour  : chr [1:2896] "Lvl" "Lvl" "Lvl" "Lvl" ...
  $ Central.Air   : chr [1:2896] "Y" "Y" "Y" "Y" ...
  $ Overall.Qual  : num [1:2896] 6 5 6 7 5 6 8 8 8 7 ...
  $ Overall.Cond  : num [1:2896] 5 6 6 5 5 6 5 5 5 5 ...
  $ Year.Built    : num [1:2896] 1960 1961 1958 1968 1997 ...
  $ Year.Remod.Add: num [1:2896] 1960 1961 1958 1968 1998 ...
  $ Total.Bsmt.SF : num [1:2896] 1080 882 1329 2110 928 ...
  $ Full.Bath     : num [1:2896] 1 1 1 2 2 2 2 2 2 2 ...
  $ Bedroom.AbvGr : num [1:2896] 3 2 3 3 3 3 2 2 2 3 ...
  $ Kitchen.AbvGr : num [1:2896] 1 1 1 1 1 1 1 1 1 1 ...
  $ Fireplaces    : num [1:2896] 2 0 0 2 1 1 0 0 1 1 ...
  $ TotRms.AbvGrd : num [1:2896] 7 5 6 8 6 7 6 5 5 7 ...
  $ Garage.Cars   : num [1:2896] 2 1 1 2 2 2 2 2 2 2 ...
  $ Sale.Price    : num [1:2896] 258000 126000 206400 292800 227880 ...
```

Using the summary() function, a quick glimpse allows for the identification of data quality issues that requires immediate attention to eliminate the opportunity of bias in the model. For instance, 485 missing (NA) values in the Lot. Frontage, Total.Bsmt.SF and Garage.Cars variables.

```
> summary(ABT)
  Lot.Frontage       Lot.Area          Street          Land.Contour       Central.Air
 Min.   : 21.0   Min.   :   1300   Length:2896        Length:2896        Length:2896
 1st Qu.: 59.0   1st Qu.:   7448   Class :character   Class :character   Class :character
 Median : 68.0   Median :   9452   Mode  :character   Mode  :character   Mode  :character
 Mean   : 69.4   Mean   :  10837
 3rd Qu.: 80.0   3rd Qu.:  11558
 Max.   :313.0   Max.   :2152450
 NA's   :485
  Overall.Qual     Overall.Cond      Year.Built    Year.Remod.Add Total.Bsmt.SF    Full.Bath
 Min.   : 1.000   Min.   :  1.000   Min.   :1872   Min.   :1950   Min.   :   0   Min.   :0.000
 1st Qu.: 5.000   1st Qu.:  5.000   1st Qu.:1953   1st Qu.:1965   1st Qu.: 793   1st Qu.:1.000
 Median : 6.000   Median :  5.000   Median :1973   Median :1993   Median : 990   Median :2.000
 Mean   : 6.103   Mean   :  6.938   Mean   :1971   Mean   :1984   Mean   :1052   Mean   :1.566
 3rd Qu.: 7.000   3rd Qu.:  6.000   3rd Qu.:2001   3rd Qu.:2004   3rd Qu.:1301   3rd Qu.:2.000
 Max.   :11.000   Max.   :999.000   Max.   :2025   Max.   :2010   Max.   :6110   Max.   :4.000
                                                                  NA's   :1
 Bedroom.AbvGr   Kitchen.AbvGr     Fireplaces      TotRms.AbvGrd     Garage.Cars      Sale.Price
 Min.   :0.000   Min.   :0.000   Min.   :0.0000   Min.   : 2.000   Min.   :0.000   Min.   : 15347
 1st Qu.:2.000   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.: 5.000   1st Qu.:1.000   1st Qu.:155370
 Median :3.000   Median :1.000   Median :1.0000   Median : 6.000   Median :2.000   Median :192000
 Mean   :2.855   Mean   :1.044   Mean   :0.5984   Mean   : 6.444   Mean   :1.768   Mean   :216999
 3rd Qu.:3.000   3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.: 7.000   3rd Qu.:2.000   3rd Qu.:256275
 Max.   :8.000   Max.   :3.000   Max.   :4.0000   Max.   :15.000   Max.   :5.000   Max.   :906000
                                                                   NA's   :1
```

The step-by-step data cleansing and preparation is as follows:

- Outliers

Using the boxplot() and hist() function, Sale.Price is plotted to identify any prevailing outliers.

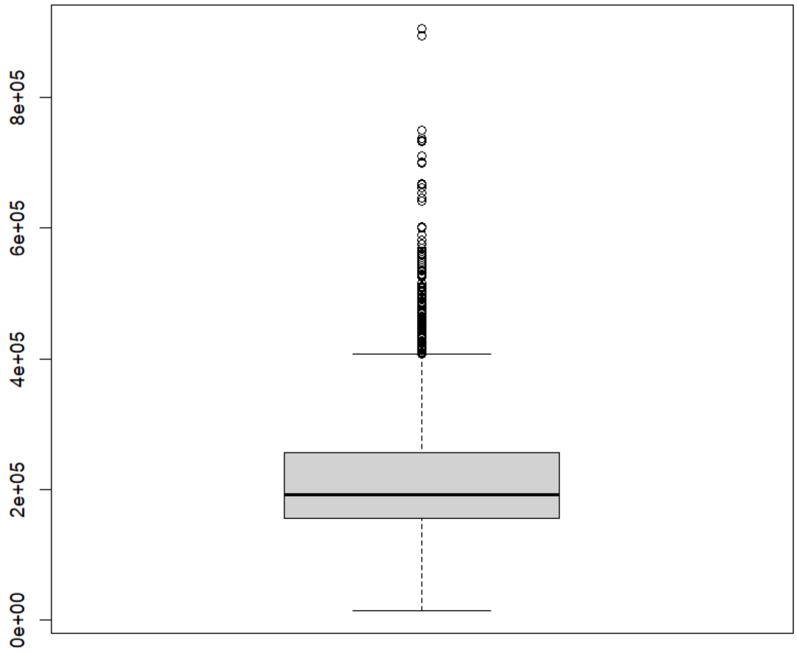

Figure 1. Boxplot of Sale Price

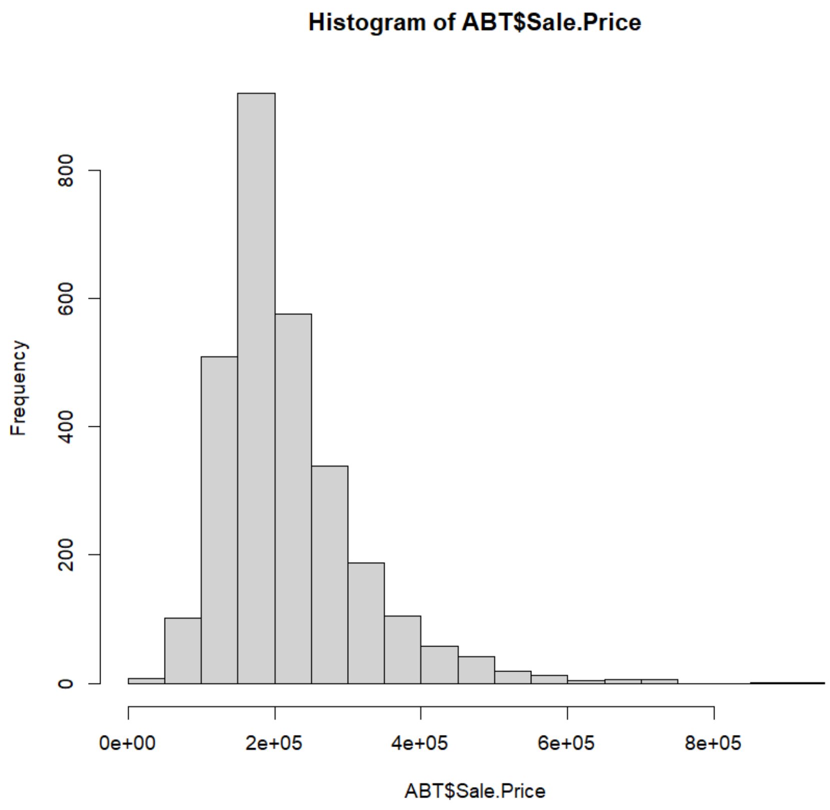

Figure 2 Histogram of Sale Price

For clearing the said variable, the observations falling beyond 700000 are assumed to be outliers, thus using the subset() functionality those observations are removed form the ABT. Similarly, with the Lot.Area variable the maximum value 2152450 is clearly an outlier and thus all the values going for 50000 are replaced NA as such observation are only 14.

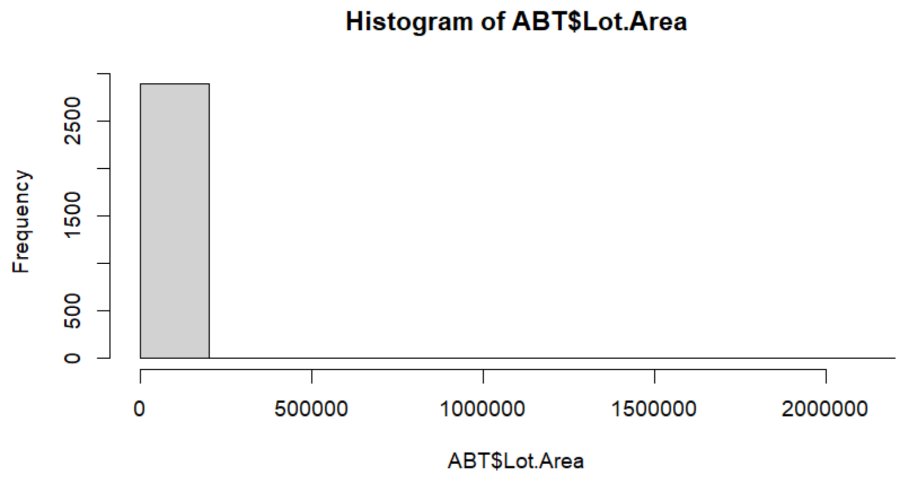

Figure 3 Histogram of Lot Area

Likewise, for TotRms.AbvGrd by using boxplot() the observations above 10 can be identified as outliers. However, using the hist() function for the variable, a close look gives an understanding that the value over 12 should be considered as outliers and thus are replaced with NA values.

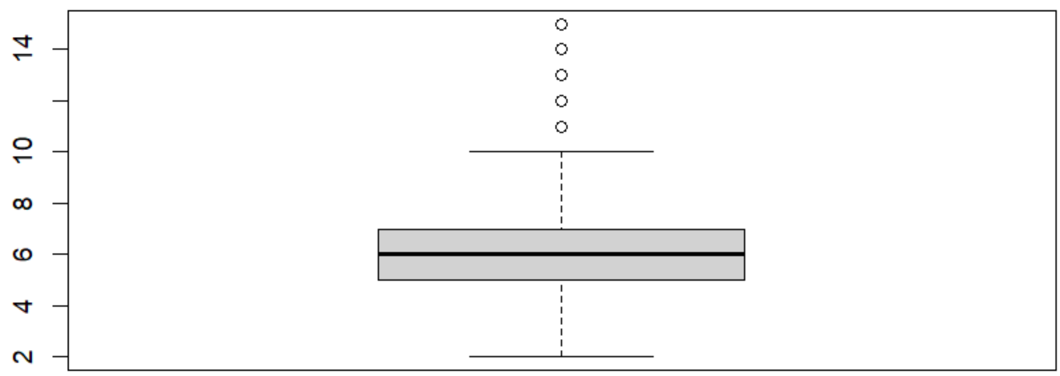

Figure 4 Box plot for Total Rooms Above Grade

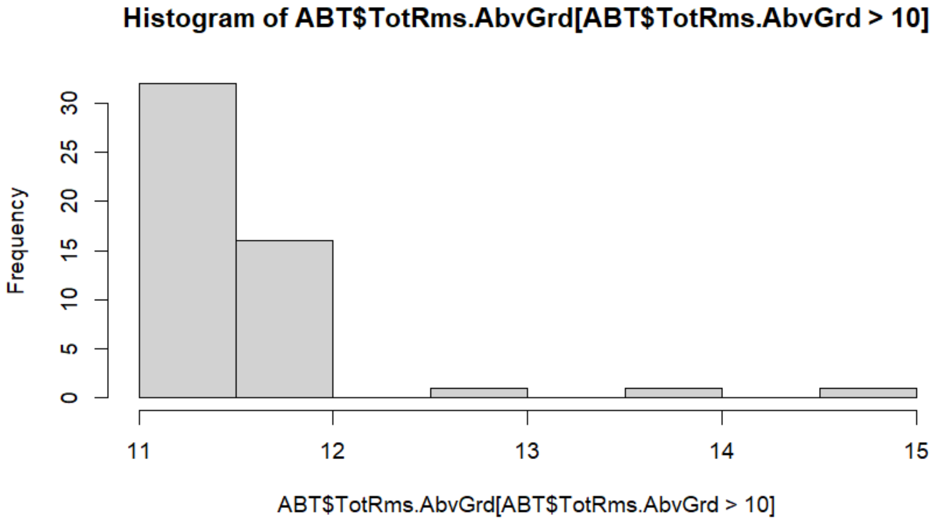

Figure 5 Histogram over Total Rooms greater than 10

Similarly, for the input variables Bedroom.AbvGr using the boxplot() function and hist() function for a closer look at the observations with values above 5 are deemed as outliers. Thus, such observations are replaced by NA.

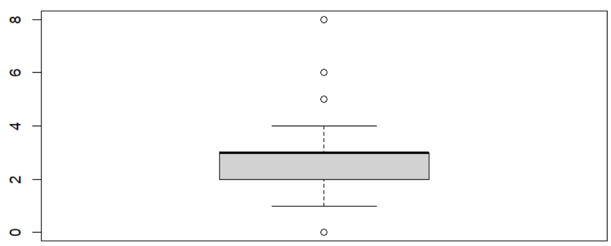

Figure 6 Boxplot for Bedroom Above Grade variable

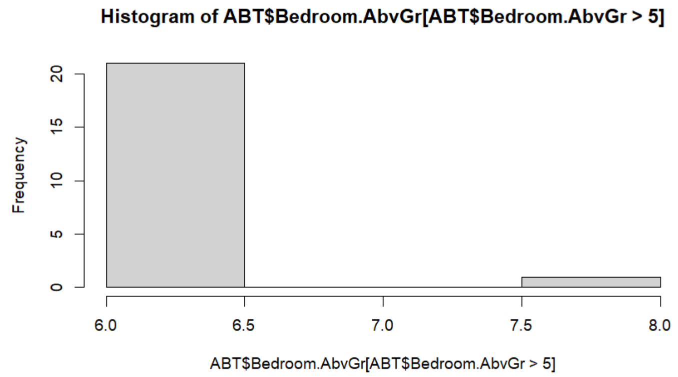

Figure 7 Histogram of the Bedroom Above Grade variable with a condition

For Garage Cars and Fireplace variables, using the same the concept the observations deemed as outliers were replaced with NA.

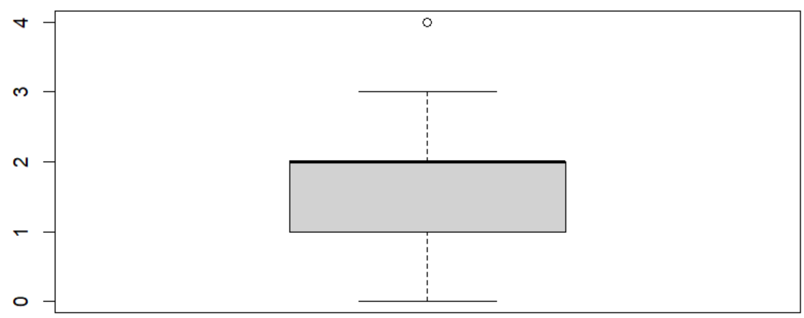

Figure 8 Boxplot of Garage Cars

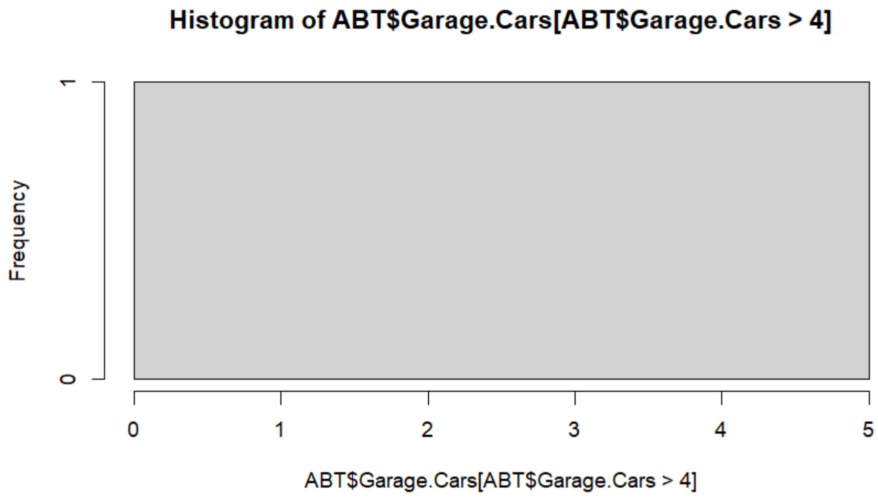

Figure 9 Histogram of Garage Cars

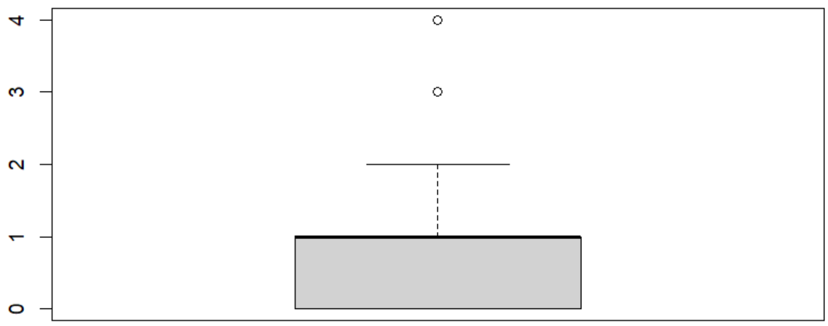

Figure 10 Box Plot of Fireplaces

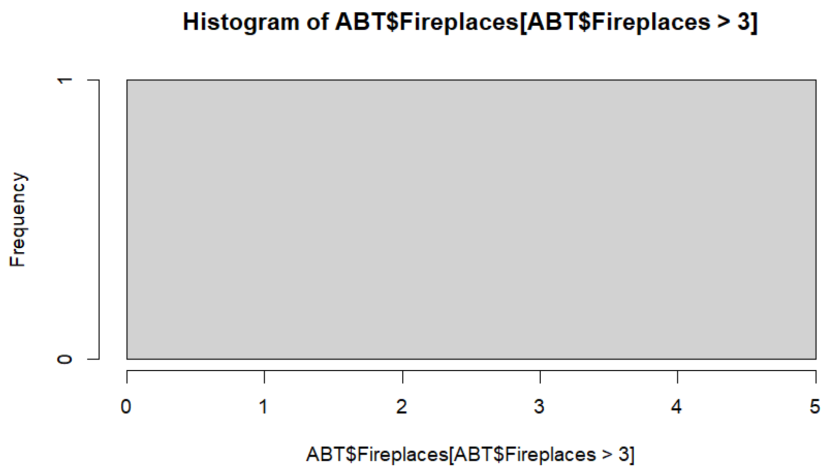

Figure 11 Histogram of Fireplaces

- Range Constraints

For Year.Built variable, the values exceeding 2010 are out of set range as the data given is for the houses sold till 2010 thus, year built cannot exceed 2010. Since the frequency of such observation is less, those values are replaced by NA.

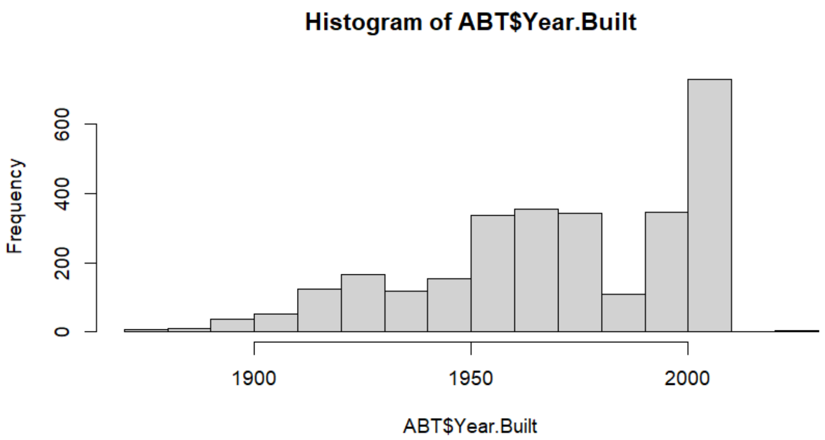

Figure 12 Histogram of Year Built

Also, using the boxplot() function, it was identified there are 7 observations below 1880 and are also replaced with NA as they are outside the lower whisker or 1st Quartile.

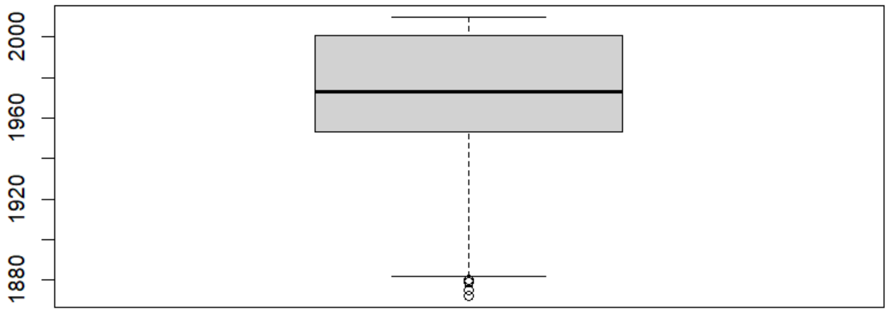

Figure 13 Box plot of Year Built

Likewise, for the Overall.Qual variable the given range is 1-10 thus anything above 10 is clearly a quality issue and such observations are replaced by NA.

- Replacing data type

For the test of association, some categorical variables such as Central.Air, Land.Contour and Street with 2, 4 and 2 levels respectively are replaced with numerical values and are converted to numerical data type from character data type using the str_replace() function.

```
#Data Types and Replacing Values to numeric
ABT$Central.Air<-  str_replace(ABT$Central.Air, "Y", "1")
ABT$Central.Air<-  str_replace(ABT$Central.Air, "N", "0")

ABT$Land.Contour<-  str_replace(ABT$Land.Contour, "Lvl", "1") #Level
ABT$Land.Contour<-  str_replace(ABT$Land.Contour, "Bnk", "2") #Banked
ABT$Land.Contour<-  str_replace(ABT$Land.Contour, "HLS", "3") #Hillside
ABT$Land.Contour<-  str_replace(ABT$Land.Contour, "Low", "4") #Depression

ABT$Street<-  str_replace(ABT$Street, "Grvl", "0") #Gravel
ABT$Street<-  str_replace(ABT$Street, "Pave", "1") #Paved

ABT <- ABT %>%
  mutate_if(is.character, as.numeric)
```

- High count of NA values

The variable Lot.Frontage has 485 NA values which are too high for data set comprised of only 2897 observations. Thus, they are replaced with the mean value of the said variable. Other NA values that were formed as part of data cleansing and preparation are omitted using the omit() function.

```
> colSums(is.na(ABT))
  Lot.Frontage        Lot.Area          Street    Land.Contour     Central.Air    Overall.Qual 
             0              13               0               0               0               0 
  Overall.Cond      Year.Built  Year.Remod.Add   Total.Bsmt.SF       Full.Bath    Bedroom.AbvGr 
             4               7               0               1               0               0 
 Kitchen.AbvGr      Fireplaces    TotRms.AbvGrd     Garage.Cars      Sale.Price 
             0               0              49               1               0 
```

```
> colSums(is.na(ABT))
  Lot.Frontage        Lot.Area          Street    Land.Contour     Central.Air    Overall.Qual 
             0               0               0               0               0               0 
  Overall.Cond      Year.Built  Year.Remod.Add   Total.Bsmt.SF       Full.Bath    Bedroom.AbvGr 
             0               0               0               0               0               0 
 Kitchen.AbvGr      Fireplaces    TotRms.AbvGrd     Garage.Cars      Sale.Price 
             0               0               0               0               0 
```

4. Modelling

After the data is prepared for the analysis, the first step is to derive the correlation between the variables available in the ABT data set created in the previous steps. Using the cor() function, the correlation is calculated for all the variables and is placed in the vector ‘Correlation’. Now using the corPlot() function the same is graphically plotted for further analysis.

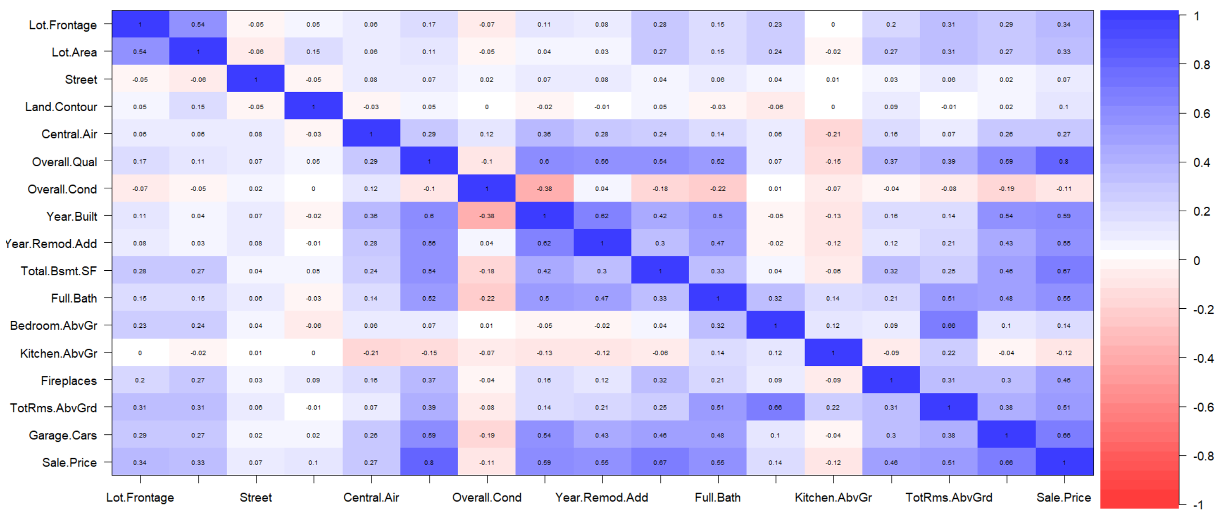

Figure 14 Correlation Plot of ABT

From the plot above, 6 variables are observed to have the high correlation with the pre-determined dependent variable (Sale.Price). Those 7 input variables are Overall.Qual (0.8), Garage.Cars(0.66), Total.Bsmt.SF (0.67), Year.Built (0.59), Full.Bath (0.55), TotRms.AbvGrd (0.51) and Year.Remod.Add (0.55). Furthermore, there are 3 other variables that will be used to create variation of model in further steps that have correlation above 0.30, these are; Fireplaces (0.46), Lot.Area (0.33) and Lot.Frontage (0.34). Remaining variables from the ABT data set that still reflect not so significant but positive correlation with Sale Price are Land.Contour, Street and Bedroom.AbvGr.

Using the information above and the ggplot() function, a few of these relations are plotted for further analysis.

- Overall Quality and Sale Price

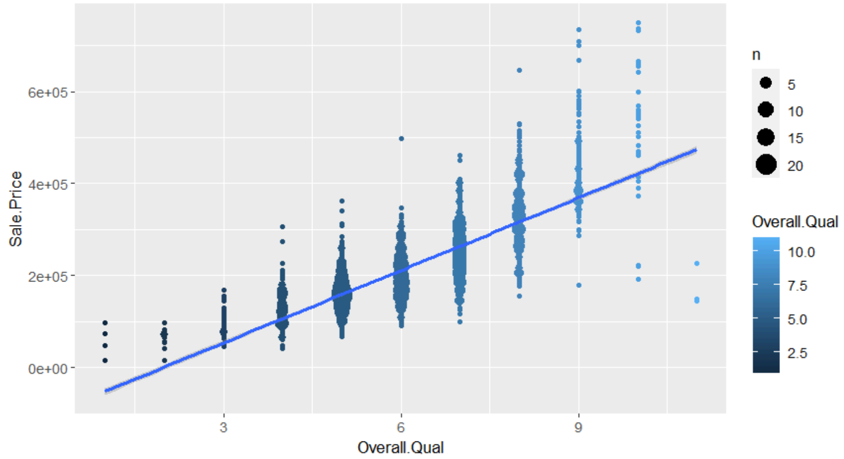

Figure 15 Count plot Overall Quality and Sale Price

- Year Built and Sale Price

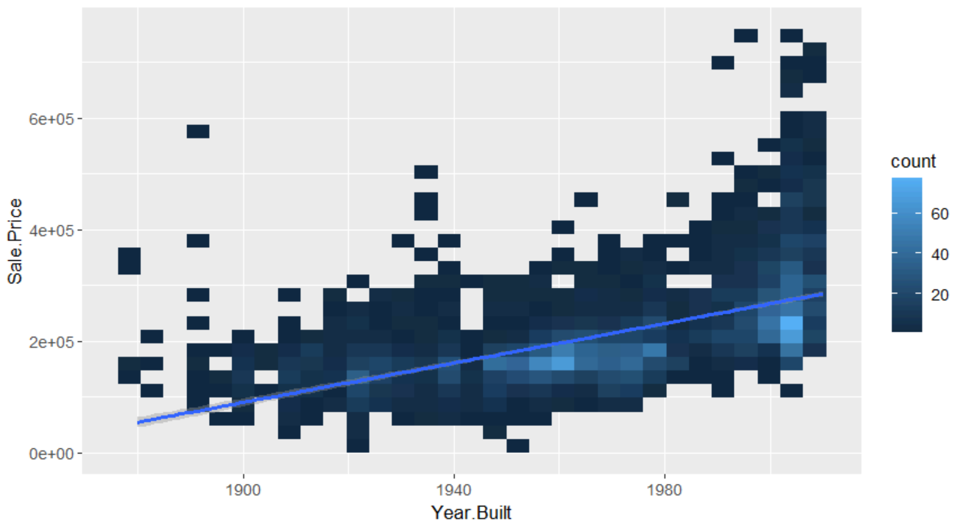

Figure 16 Heat plot Year Built and Sale Price

- Year Remodelling Added and Sale Price

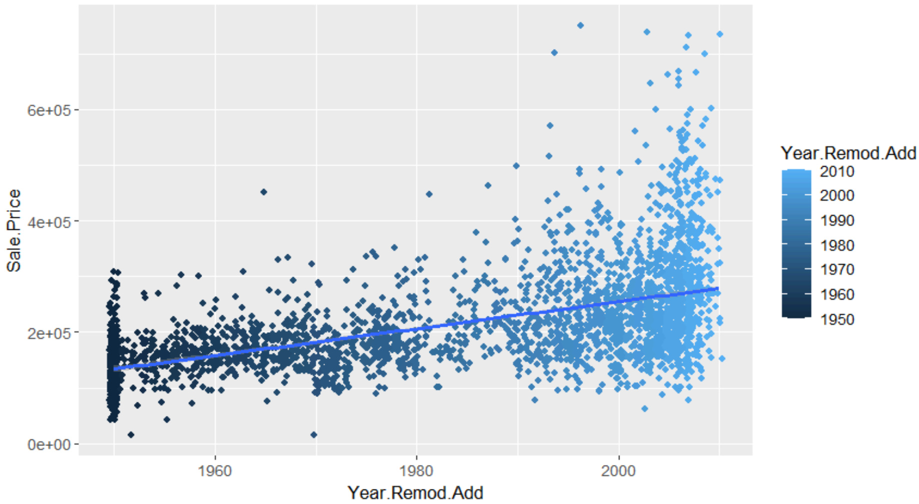

Figure 17 Jitter Plot for Year Built and Sale Price

- Total Basement Sqft. and Sale Price

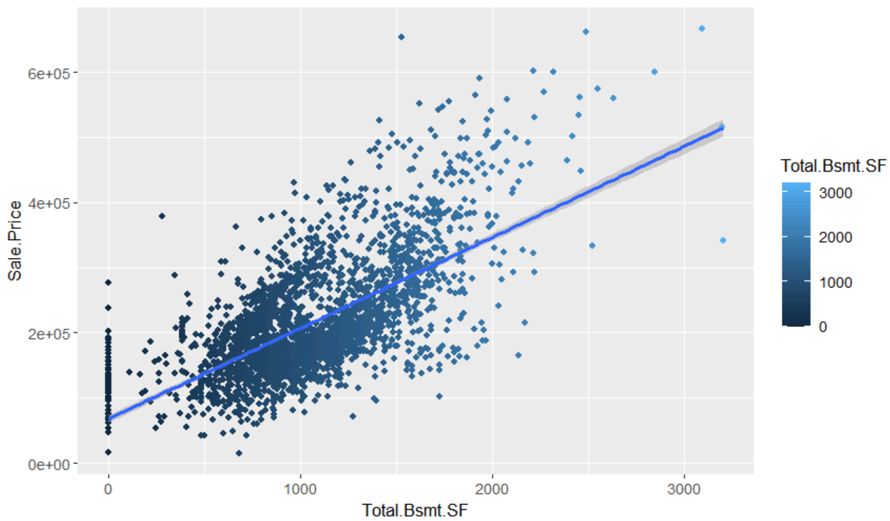

Figure 18 Scatter Plot for Total Bsmt SF and Sale Price

- Garage Cars and Sale Price

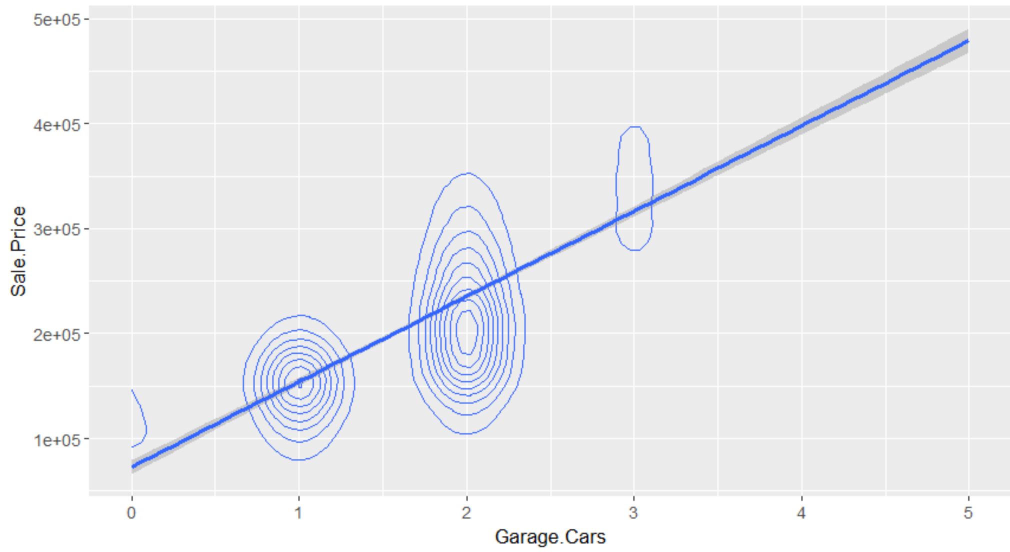

Figure 19 Density Plot for Garage Cars and Sale Price

- Correlation of Analytics base table rounded off to 2 digits after decimal

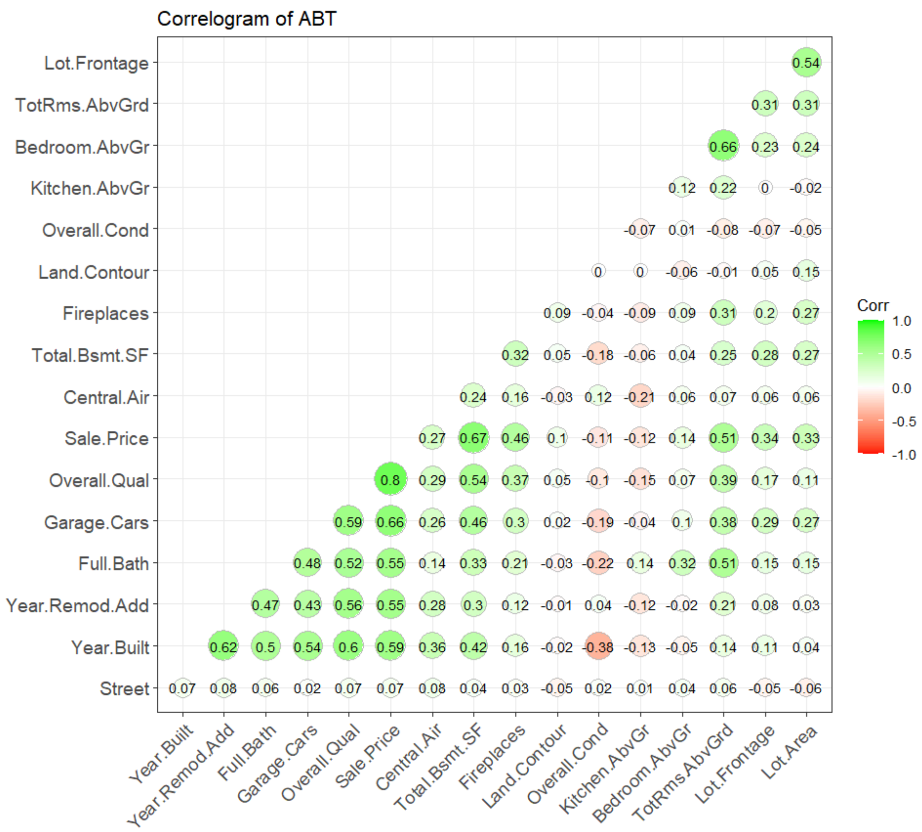

Figure 20 Correlogram of ABT

- Regression Model

To create a linear regression model to predict the sale price, the data is first split using createDataPartition() function in to train (80%) and test (20%) datasets. Multiple regression model is created using lm() function on the train data set. Different variations of the input variables were used and finally a model with Multiple R-Squared value 0.8196 (0.82) was selected to create predictions. The selected variables for the formula are as follows:

```
> Formula <- Sale.Price ~ Overall.Qual + Garage.Cars + Total.Bsmt.SF + TotRms.AbvGrd + Year.Remod.Add +
Fireplaces + Lot.Area + Land.Contour + Bedroom.AbvGr + Year.Built
```

5. Evaluation

After the finalising the multiple linear regression model created using the train data set, it is then used to create the prediction model using the predict() function on the test dataset. To evaluate the accuracy of the prediction model postResample() function is used to compare the predicted values with the actual values in the 'test' dataset and is placed in the vector 'accuracy'. The value of the 'accuracy' vector is rounded off to 2 decimal places.

```
> round(accuracy, digits = 2) #0.80
    RMSE Rsquared      MAE
37136.18     0.80 27385.20
```

6. Assumption Checks

There are certain assumptions that can underpin the regression model create. Thus, assumption checking is an important step in regression model building. It ensures that the model is valid and that the results obtained from the model are reliable. Assumption checking helps to identify potential problems with the data, model specification, and model assumptions. It also helps to determine potential solutions that can improve the model (James et al., 2022).

- Multicolinearity – The input variables selected should not be highly correlated with each other. To check this assumption, a Variance Inflation Factor (VIF) test using the vif() function from the "car" package is done on the multiple regression model created on the train data set.

```
> vif(multiple_model10) #all values are < 3
  Overall.Qual    Garage.Cars  Total.Bsmt.SF  TotRms.AbvGrd Year.Remod.Add     Fireplaces
      2.645989       1.944942       1.608930       2.569066       1.827949       1.311626
      Lot.Area   Land.Contour  Bedroom.AbvGr     Year.Built
      1.298611       1.040770       2.009339       2.158371
> mean(vif(multiple_model10)) #1.73
[1] 1.841559
```

While the VIF for input variables is lower than 3, however the average VIF is a little over 1. Thus, it shows a scope of bias in our model even though it's very low (Glen, 2020).

- Autocorrelation – A Durbin-Watson (DW) test using the dwtest() function from the “lmtest” package suggest is done to determine the extent to which the values of the variables are correlated with each other.

```
> dwtest(multiple_model10) #1.57

        Durbin-Watson test

data:  multiple_model10
DW = 1.5792, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0
```

  With a value of 1.60, it can be assumed that the independent errors are positively correlated (Kenton, 2022).

- Influential Outliers – This assumption is checked using the cooks distance method. Using the cooks. distance() function a ‘cook’ vector is created to identify any influential observations that can affect the accuracy of the model.

```
> sum(cook > 1) #0
[1] 0
```

  With no observations above 1, it can be inferred that there are no influential outliers (Glen, 2022).

- Homoscedasticity - The assumption of homoscedasticity is that the variance of the errors is constant across all values of the independent variables. If this assumption is violated, the estimated model coefficients may be biased and the standard errors may be incorrect (Yang et al., 2019).

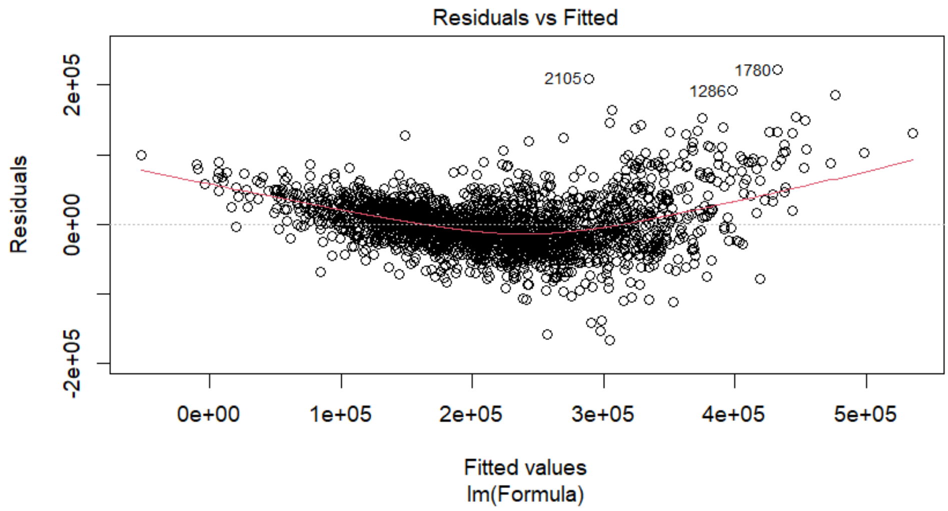

Figure 21 Residual Vs Fitted

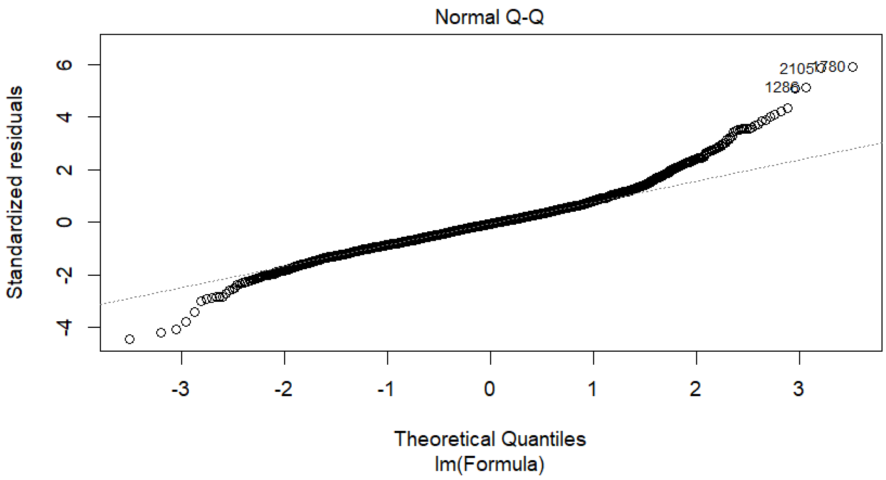

Figure 22 Normal Q-Q plot

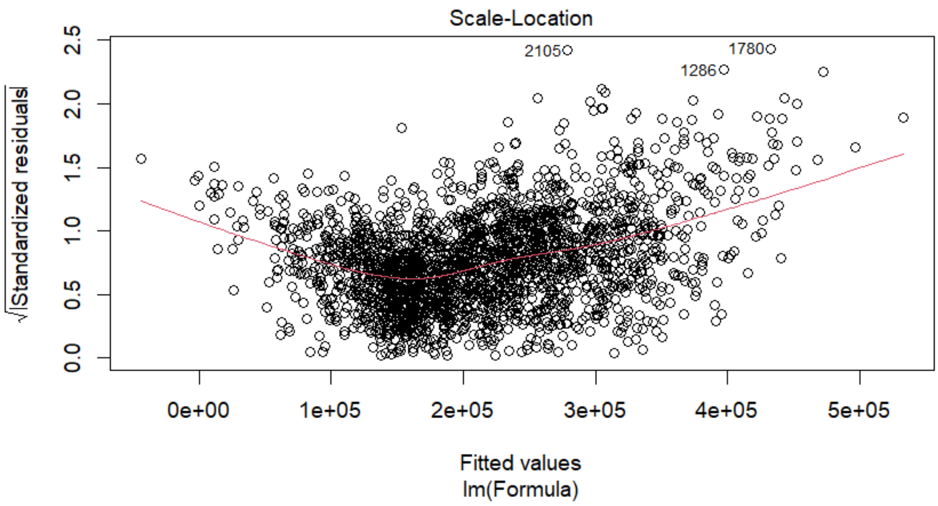

Figure 23 Square Root Standardised Residual Plot

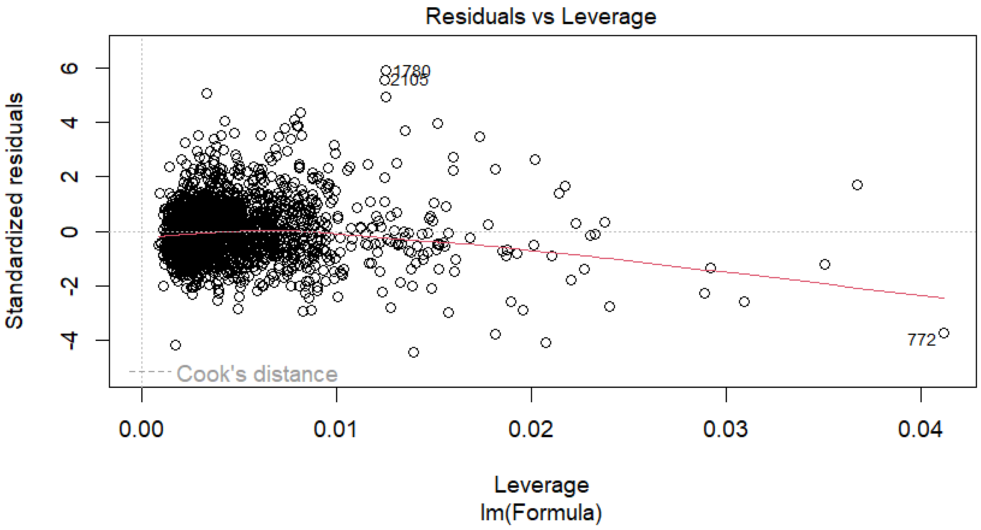

Figure 24 Standardised Residual Plot

- Normality: The assumption of normality is that the errors in the model are normally distributed. If this assumption is violated, the estimated model coefficients may be biased and the standard errors may be incorrect (Spanos, 2011).

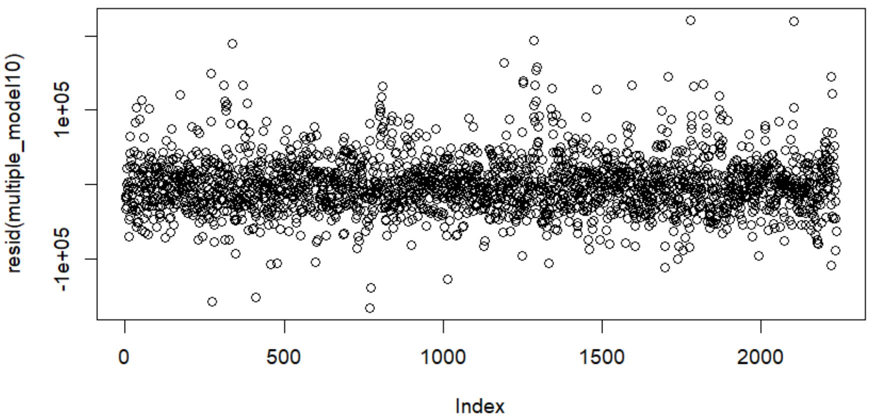

Figure 25 Residual Plot

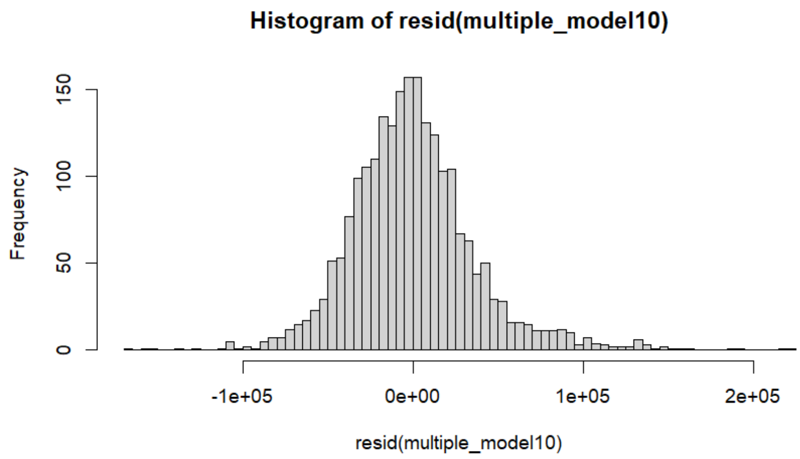

Figure 26 Histogram of Residuals

Conclusion

Inferring from the above presented information, it can be concluded that:

- H1: There is a positive relationship between the overall quality of the house and the sale price. A 1 unit change in overall quality results in $ 24350 increase in sale price.
- H2: There is a positive relationship between the year a house is built in and its sale price. A unit change in the Year Built results in $ 228.8 increase in price.
- H3: A positive impact can be observed between the year remodelling added and it sales price as a unit change in Year Remod Add result in $ 474.8 in sale price.
- H4: There is a positive association between the total sqft. Of the basement area in the sale price as a 1 unit change in the total basement sqft. Area results in $ 50.61 increase in its price.
- H5: The number of cars in a garage has positive impact on the sale price of the house as a unit change in the Garage Car results in $ 13970 jump in price.

```
> summary(multiple_model10) #0.8224(0.82)

Call:
lm(formula = Formula, data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-167006  -22438   -2395   18666  221531 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -1.497e+06  9.883e+04 -15.143  < 2e-16 ***
Overall.Qual    2.308e+04  9.211e+02  25.060  < 2e-16 ***
Garage.Cars     1.397e+04  1.498e+03   9.325  < 2e-16 ***
Total.Bsmt.SF   5.061e+01  2.412e+00  20.981  < 2e-16 ***
TotRms.AbvGrd   1.383e+04  8.990e+02  15.385  < 2e-16 ***
Year.Remod.Add  4.748e+02  5.136e+01   9.245  < 2e-16 ***
Fireplaces      1.256e+04  1.464e+03   8.578  < 2e-16 ***
Lot.Area        2.236e+00  2.054e-01  10.882  < 2e-16 ***
Land.Contour    7.118e+03  1.424e+03   5.000 6.18e-07 ***
Bedroom.AbvGr  -8.966e+03  1.457e+03  -6.154 8.95e-10 ***
Year.Built      2.288e+02  3.901e+01   5.866 5.14e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 37690 on 2227 degrees of freedom
Multiple R-squared:  0.8224,	Adjusted R-squared:  0.8216 
F-statistic:  1031 on 10 and 2227 DF,  p-value: < 2.2e-16
```

Other factors like ‘Total Rooms Above Grade’, ‘Number of Fireplace’ and ‘Land Contour’ has high positive impact on the sale price of the house. While ‘Bedrooms Above Grade’ has a negative association with the sale price of the houses. From the assumption checks conducted, it can be inferred that they underpin the multiple regression model created.

Reflective Summary

While taking on the objectives of the assignment using statistics in R initially seemed daunting, however it gradually changed with every task carried out. Data preparation for the creation of the model and selecting input variables for the final model was quite challenging. There were 9 model created before one final was selected with R squared value. The assumption checks output highlights towards the not so robust nature of the final model.

References

Kauko, T., et al. (2002). "Capturing housing market segmentation: An alternative approach based on neural network modelling." Housing Studies 17(6): 875-894.

Cao, B. and Yang, B. (2018) Research on ensemble learning-based housing price prediction model, Big Geospatial Data and Data Science. Clausius Scientific Press. Available at: https://www.clausiuspress.com/article/141.html (Accessed: December 19, 2022).

James, G. et al. (2022) An introduction to statistical learning: With applications in R. Boston, Massachusetts: Springer.

Li, X. (2022) Prediction and Analysis of Housing Price Based on the Generalized Linear Regression Model, Shibboleth authentication request. Available at: https://www-hindawi-com.queens.ezp1.qub.ac.uk/journals/cin/2022/3590224/ (Accessed: December 19, 2022).

Mann, T. (2018) Linear regression assumptions and diagnostics in R: Essentials, STHDA. Available at: http://www.sthda.com/english/articles/39-regression-model-diagnostics/161-linear-regression-assumptions-and-diagnostics-in-r-essentials/ (Accessed: December 22, 2022).

Ebru, Ç. and Eban , A. (2009) Determinants of house prices in Istanbul: A quantile ... - springer, Determinants of house prices in Istanbul: a quantile regression approach. Available at: https://link.springer.com/content/pdf/10.1007/s11135-009-9296-x.pdf (Accessed: December 22, 2022).

Palmquist, R.B. (1980) Alternative techniques for developing real estate price indexes - JSTOR, Alternative Techniques for Developing Real Estate Price Indexes. Available at: https://www.jstor.org/stable/1927112 (Accessed: December 22, 2022).

Spanos, A. (2011) On normality and the linear regression model, Taylor & Francis. Available at: https://www.tandfonline.com/doi/abs/10.1080/07474939508800314?casa_token=4qN6nks1dKoAAAAA%3A89dixLWuA1o2DnU3A6-agnpzbpdIojDbzTYRRboQbXckvLUa3URS8b5DuDVmJcc-6Zo7eZmLf_iH (Accessed: December 22, 2022).

Yang, K., Tu, J. and Chen, T. (2019) Homoscedasticity: An overlooked critical assumption for linear regression, General psychiatry. U.S. National Library of Medicine. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6802968/ (Accessed: December 22, 2022).

Field, A. (2020) “8.2.4,” in Discovering statistics using IBM SPSS statistics. London, United Kingdom: Sage, pp. 300–303.

Field, A., Miles, J. and Field Zoë (2022) "7.9," in Discovering statistics using R. Thousand Oaks, California: SAGE/Texts, pp. 287–298.

Glen, S. (2020) Residual values (residuals) in regression analysis, Statistics How To. Available at: https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/residual/ (Accessed: December 21, 2022).

Glen, S. (2020) Variance inflation factor, Statistics How To. Available at: https://www.statisticshowto.com/variance-inflation-factor/ (Accessed: December 21, 2022).

Glen, S. (2022) Cook's distance / cook's D: Definition, interpretation, Statistics How To. Available at: https://www.statisticshowto.com/cooks-distance/ (Accessed: December 21, 2022).

Green, R. et al. (1999) Age, housing demand, and Real House prices, Regional Science and Urban Economics. North-Holland. Available at: https://www.sciencedirect.com/science/article/pii/016604629602128X (Accessed: December 22, 2022).

Guo, J.-qiang et al. (2020) Can machine learning algorithms associated with text mining from internet data improve housing price prediction performance?, International Journal of Strategic Property Management. Available at: https://journals.vilniustech.lt/index.php/IJSPM/article/view/12742 (Accessed: December 19, 2022).

Kenton, W. (2022) Durbin Watson test: What it is in statistics, with examples, Investopedia. Investopedia. Available at: https://www.investopedia.com/terms/d/durbin-watson-statistic.asp#:~:text=The%20Durbin%20Watson%20statistic%20is,above%202.0%20indicates%20negative%20autocorrelation. (Accessed: December 21, 2022).

Lee, J.R. (2021) Prediction of housing price index using Artificial Neural Network, Journal of the Korea Academia-Industrial cooperation Society. The Korea Academia-Industrial cooperation Society. Available at: https://www.koreascience.or.kr/article/JAKO202113855736415.page (Accessed: December 19, 2022).

Liu, R. and Liu, L. (2019) Predicting housing price in China based on long short-term memory incorporating modified genetic algorithm - soft computing, SpringerLink. Springer Berlin Heidelberg. Available at: https://link.springer.com/article/10.1007/s00500-018-03739-w (Accessed: December 19, 2022).

Montero, J.-M., Mínguez, R. and Fernández-Avilés, G. (2017) Housing price prediction: Parametric versus semi-parametric spatial hedonic models - journal of geographical systems, SpringerLink.

Springer Berlin Heidelberg. Available at: https://link.springer.com/article/10.1007/s10109-017-0257-y (Accessed: December 19, 2022).

Wilhelmsson, M. et al. (2007) House price depreciation rates and level of maintenance, Journal of Housing Economics. Academic Press. Available at: https://www.sciencedirect.com/science/article/pii/S1051137707000435?casa_token=-Hnj9TB-hb8AAAAA%3AN-of96QlUqvTgAniRRLd1CW-PnmgXdt5Qh9AENFEOJtWTLNSP25gKqA3RjFpLlydpn93HTxydg (Accessed: December 22, 2022).

Yakub, A.R.A. et al. (2020) The effect of adopting micro and macro-economic variables on real ..., THE EFFECT OF ADOPTING MICRO AND MACRO-ECONOMIC VARIABLES ON REAL ESTATE PRICE PREDICTION MODELS USING ANN: A SYSTEMATIC LITERATURE REVIEW. Available at: https://www.researchgate.net/profile/Abdurraheem-Yakub-2/publication/342438565_Journal_of_Critical_Reviews_THE_EFFECT_OF_ADOPTING_MICRO_AND_MACRO-ECONOMIC_VARIABLES_ON_REAL_ESTATE_PRICE_PREDICTION_MODELS_USING_ANN_A_SYSTEMATIC_LITERATURE_REVIEW/links/5ef47b7d299bf15a2ea09ed6/Journal-of-Critical-Reviews-THE-EFFECT-OF-ADOPTING-MICRO-AND-MACRO-ECONOMIC-VARIABLES-ON-REAL-ESTATE-PRICE-PREDICTION-MODELS-USING-ANN-A-SYSTEMATIC-LITERATURE-REVIEW.pdf (Accessed: December 19, 2022).

Yu, L. et al. (2018) Prediction on housing price based on Deep Learning, International Journal of Computer and Information Engineering. Available at: http://publications.waset.org/10008599/prediction-on-housing-price-based-on-deep-learning (Accessed: December 19, 2022).

T.L.Ooi, J. et al. (2014) The impact of construction quality on house prices, Journal of Housing Economics. Academic Press. Available at: https://www.sciencedirect.com/science/article/pii/S1051137714000497?casa_token=GTDk8cSZVwgAAAAA%3Ay4WVT6X0ajC4Gcehd6696U92KtFfdnpcKB7DQEullkUo6GOZgoZXbvfqJl5dWdQGVK-R3gjyUw (Accessed: December 22, 2022).

Appendix

```
#Set WD

setwd("C:/Users/Prayas Sachdeva/Downloads")


#Install and Load the Packages

install.packages('readr')

install.packages('caret')

install.packages('ggplot2')

install.packages('dplyr')

install.packages('psych')

install.packages('Hmisc')

install.packages('stringr')

library(readxl)

library(psych)

library(caret)

library(ggplot2)

library(dplyr)

library(Hmisc)

library(stringr)

#Read the data

ames <- read_excel("ames (2).xlsx")

summary(ames)

#creating ABT
```

```
ABT <- ames %>%
  select(Lot.Frontage, Lot.Area, Street, Land.Contour,Central.Air, Overall.Qual, Overall.Cond, Year.Built, Year.Remod.Add, Total.Bsmt.SF, Full.Bath, Bedroom.AbvGr, Kitchen.AbvGr, Fireplaces, TotRms.AbvGrd, Garage.Cars, Sale.Price)



#data quality issues
glimpse(ABT)
summary(ABT)
describe(ABT)

#DQ sale.price
boxplot(ABT$Sale.Price)
hist(ABT$Sale.Price[ABT$Sale.Price>700000])
sum(ABT$Sale.Price>700000)
ABT$Sale.Price[ABT$Sale.Price>700000] <- NA
ABT <- subset(ABT, !is.na(ABT$Sale.Price))
ABT <- ABT[!is.na(ABT$Sale.Price),]
summary(ABT$Sale.Price)
describe(ABT$Sale.Price)
qqnorm(ABT$Sale.Price)
qqline(ABT$Sale.Price)
Sale.price.ND <- rnorm(ABT$Sale.Price, mean = mean(ABT$Sale.Price), sd(ABT$Sale.Price))
hist(Sale.price.ND)
qqnorm(Sale.price.ND)
qqline(Sale.price.ND)
```

```
summary(Sale.price.ND)

#DQ lot area

boxplot(ABT$Lot.Area)

hist(ABT$Lot.Area)

ABT$Lot.Area[ABT$Lot.Area>50000] <- NA

sum(ABT$Lot.Area>50000)

hist(ABT$Lot.Area)

summary(ABT$Lot.Area)

describe(ABT$Lot.Area)

#DQ Year built

boxplot(ABT$Year.Built)

hist(ABT$Year.Built)

sum(ABT$Year.Built<1880 | ABT$Year.Built>2010)

ABT$Year.Built[ABT$Year.Built<1880 | ABT$Year.Built>2010] <- NA


#range constraints with overall condition

sum(ABT$Overall.Cond<1 | ABT$Overall.Cond>10)

ABT$Overall.Cond[ABT$Overall.Cond>10] <- NA


#DQ Total rooms

hist(ABT$TotRms.AbvGrd)

boxplot(ABT$TotRms.AbvGrd)

sum(ABT$TotRms.AbvGrd>10)

hist(ABT$TotRms.AbvGrd[ABT$TotRms.AbvGrd>10])

ABT$TotRms.AbvGrd[ABT$TotRms.AbvGrd>10] <- NA
```

```
#DQ Bedrooms ABVGR

boxplot(ABT$Bedroom.AbvGr)

hist(ABT$Bedroom.AbvGr[ABT$Bedroom.AbvGr>5])

sum(ABT$Bedroom.AbvGr>5)

ABT$Bedroom.AbvGr[ABT$Bedroom.AbvGr>5] <- NA


#DQ Lot Frontage

ABT$Lot.Frontage[is.na(ABT$Lot.Frontage)] <- mean(ABT$Lot.Frontage, na.rm = TRUE)


#DQ Garage Cars

boxplot(ABT$Garage.Cars)

hist(ABT$Garage.Cars[ABT$Garage.Cars>4])

sum(ABT$Garage.Cars>4)

ABT$Garage.Cars[ABT$Garage.Cars>4] <- NA


#DQ Fireplaces

boxplot(ABT$Fireplaces)

hist(ABT$Fireplaces[ABT$Fireplaces>3])

sum(ABT$Fireplaces>3)

ABT$Fireplaces[ABT$Fireplaces>3] <- NA


#Data Types and Replacing Values to numeric

ABT$Central.Air<-  str_replace(ABT$Central.Air, "Y", "1")

ABT$Central.Air<-  str_replace(ABT$Central.Air, "N", "0")


ABT$Land.Contour<-  str_replace(ABT$Land.Contour, "Lvl", "1") #Level
```

```
ABT$Land.Contour<-  str_replace(ABT$Land.Contour, "Bnk", "2") #Banked

ABT$Land.Contour<-  str_replace(ABT$Land.Contour, "HLS", "3") #Hillside

ABT$Land.Contour<-  str_replace(ABT$Land.Contour, "Low", "4") #Depression


ABT$Street<-  str_replace(ABT$Street, "Grvl", "0") #Gravel

ABT$Street<-  str_replace(ABT$Street, "Pave", "1") #Paved


ABT <- ABT %>%

  mutate_if(is.character, as.numeric)


#Dealing with NA

colSums(is.na(ABT))

ABT <- na.omit(ABT)

summary(ABT)


#correlation

Correlation <- round(cor(ABT), 2)

corPlot(Correlation)


#Visualization


# Overall Quality with Sales Price

ggplot(ABT, aes(x=Overall.Qual, y=Sale.Price, color = Overall.Qual)) +

  geom_count() +

  geom_smooth(method = "lm")
```

```
# Year Built with Sales Price

ggplot(ABT, aes(x=Year.Built, y=Sale.Price, color = Year.Built)) +

  geom_bin_2d() +

  geom_smooth(method = "lm")


# Year Remod add with Sales Price

ggplot(ABT, aes(x=Year.Remod.Add, y=Sale.Price, color  = Year.Remod.Add)) +

  geom_jitter() +

  geom_smooth(method = "lm")


# Total Basement SF with Sales Price

ggplot(ABT, aes(x=Total.Bsmt.SF, y=Sale.Price, color  = Total.Bsmt.SF)) +

  geom_point() +

  geom_smooth(method = "lm")


# Garage Cars with Sales Price

ggplot(ABT, aes(x=Garage.Cars, y=Sale.Price, fill = Sale.Price))+

  geom_density2d() +

  geom_smooth(method = "lm")


install.packages("ggcorrplot")

library(ggcorrplot)

ggcorrplot(Correlation, hc.order = TRUE,

           type = "lower",

           lab = TRUE,

           lab_size = 3,
```

```
        method="circle",
        colors = c("red", "white", "green"),
        title="Correlogram of ABT",
        ggtheme=theme_bw)
#splitting the data into train(80%) and test(20%) set
set.seed(40386503)
index <- createDataPartition(ABT$Sale.Price, list = FALSE, p=0.8, times = 1)
train <- ABT[index,]
test <- ABT[-index,]

#Simple linear regression model
simplemodel1 <- lm(Sale.Price ~ Overall.Qual, data = train)
simplemodel2 <- lm(Sale.Price ~ Garage.Cars, data = train)
simplemodel3 <- lm(Sale.Price ~ Full.Bath, data = train)
simplemodel4 <- lm(Sale.Price ~ Year.Remod.Add, data = train)
simplemodel5 <- lm(Sale.Price ~ Year.Built, data = train)
simplemodel6 <- lm(Sale.Price ~ TotRms.AbvGrd, data = train)
simplemodel7 <- lm(Sale.Price ~ Lot.Area, data = train)
simplemodel8 <- lm(Sale.Price ~ Lot.Frontage, data = train)
simplemodel9 <- lm(Sale.Price ~ Total.Bsmt.SF, data = train)
simplemodel10 <- lm(Sale.Price ~ Fireplaces, data = train)

#review the model
summary(simplemodel1) #0.64
summary(simplemodel2) #0.45
summary(simplemodel3) #0.29
```

```
summary(simplemodel4) #0.31

summary(simplemodel5) #0.33

summary(simplemodel6) #0.27

summary(simplemodel7) #0.11

summary(simplemodel8) #0.11

summary(simplemodel9) #0.45

summary(simplemodel10) #0.21


#prediction using model

simple_price_prediction1 <- predict(simplemodel1, newdata = test)

simple_price_prediction2 <- predict(simplemodel2, newdata = test)

simple_price_prediction3 <- predict(simplemodel3, newdata = test)

simple_price_prediction4 <- predict(simplemodel4, newdata = test)

simple_price_prediction5 <- predict(simplemodel5, newdata = test)

simple_price_prediction6 <- predict(simplemodel6, newdata = test)

simple_price_prediction7 <- predict(simplemodel7, newdata = test)

simple_price_prediction8 <- predict(simplemodel8, newdata = test)

simple_price_prediction9 <- predict(simplemodel9, newdata = test)

simple_price_prediction10 <- predict(simplemodel10, newdata = test)


#evaluate the accuracy

simplemodel_accuracy1 <- postResample(pred = simple_price_prediction1, obs = test$Sale.Price)

round(simplemodel_accuracy1, digits = 2) #0.62


simplemodel_accuracy2 <- postResample(pred = simple_price_prediction2, obs = test$Sale.Price)

round(simplemodel_accuracy2, digits = 2) #0.39
```

```
simplemodel_accuracy3 <- postResample(pred = simple_price_prediction3, obs = test$Sale.Price)

round(simplemodel_accuracy3, digits = 2) #0.29


simplemodel_accuracy4 <- postResample(pred = simple_price_prediction4, obs = test$Sale.Price)

round(simplemodel_accuracy4, digits = 2) #0.29


simplemodel_accuracy5 <- postResample(pred = simple_price_prediction5, obs = test$Sale.Price)

round(simplemodel_accuracy5, digits = 2) #0.34


simplemodel_accuracy6 <- postResample(pred = simple_price_prediction6, obs = test$Sale.Price)

round(simplemodel_accuracy6, digits = 2) #0.20


simplemodel_accuracy7 <- postResample(pred = simple_price_prediction7, obs = test$Sale.Price)

round(simplemodel_accuracy7, digits = 2) #0.12


simplemodel_accuracy8 <- postResample(pred = simple_price_prediction8, obs = test$Sale.Price)

round(simplemodel_accuracy8, digits = 2) #0.15


simplemodel_accuracy9 <- postResample(pred = simple_price_prediction9, obs = test$Sale.Price)

round(simplemodel_accuracy9, digits = 2) #0.40


simplemodel_accuracy10 <- postResample(pred = simple_price_prediction10, obs = test$Sale.Price)

round(simplemodel_accuracy10, digits = 2) #0.21


#multiple regression model
```

multiple_model1 <- lm(Sale.Price ~ Overall.Qual + Garage.Cars + Total.Bsmt.SF + Year.Built + Full.Bath, data = train)

summary(multiple_model1) #0.76

multiple_model2 <- lm(Sale.Price ~ Overall.Qual + Garage.Cars + Total.Bsmt.SF + Full.Bath + Year.Remod.Add + Year.Built, data = train)

summary(multiple_model2) #0.77

multiple_model3 <- lm(Sale.Price ~ Overall.Qual + Garage.Cars + Total.Bsmt.SF + Year.Built + Year.Remod.Add + Full.Bath + Fireplaces, data = train)

summary(multiple_model3) #0.79

multiple_model4 <- lm(Sale.Price ~ Overall.Qual + Garage.Cars + Total.Bsmt.SF + Full.Bath + Year.Remod.Add + TotRms.AbvGrd + Fireplaces + Year.Built, data = train)

summary(multiple_model4) #0.81

multiple_model5 <- lm(Sale.Price ~ Overall.Qual + Garage.Cars + Total.Bsmt.SF + Full.Bath + Year.Remod.Add + TotRms.AbvGrd + Fireplaces + Year.Built + TotRms.AbvGrd, data = train)

summary(multiple_model5) #0.81

multiple_model6 <- lm(Sale.Price ~ Overall.Qual + Garage.Cars + Total.Bsmt.SF + Full.Bath + Year.Remod.Add + TotRms.AbvGrd + Fireplaces + Year.Built + TotRms.AbvGrd + Lot.Area, data = train)

summary(multiple_model6) #0.82

multiple_model7 <- lm(Sale.Price ~ Overall.Qual + Garage.Cars + Total.Bsmt.SF + Full.Bath + Year.Remod.Add + TotRms.AbvGrd + Fireplaces + Year.Built + TotRms.AbvGrd + Lot.Area + Lot.Frontage, data = train)

summary(multiple_model7) #0.82

```
multiple_model8 <- lm(Sale.Price ~ Overall.Qual + Garage.Cars + Total.Bsmt.SF + Full.Bath + Year.Remod.Add + TotRms.AbvGrd + Fireplaces + Lot.Frontage + Lot.Area + Year.Built, data = train)

summary(multiple_model8) #0.82


multiple_model9 <- lm(Sale.Price ~ Overall.Qual + Garage.Cars + Total.Bsmt.SF + Year.Remod.Add + TotRms.AbvGrd + Fireplaces + Lot.Area + Year.Built + Land.Contour, data = train)

summary(multiple_model9) #0.82


Formula <- Sale.Price ~ Overall.Qual + Garage.Cars + Total.Bsmt.SF + TotRms.AbvGrd + Year.Remod.Add + Fireplaces + Lot.Area + Land.Contour + Bedroom.AbvGr + Year.Built

multiple_model10 <- lm(formula = Formula, data = train)

summary(multiple_model10) #0.8224(0.82)


#prediction using multiple model

multiple_price_prediction_model <- predict(multiple_model10, newdata = test)


#evaluate accuracy of multiple model

accuracy <- postResample(multiple_price_prediction_model, test$Sale.Price)

round(accuracy, digits = 2) #0.80


#testing assumptions

install.packages('car')

library(car)


#Multicolinearity
```

```
vif(multiple_model10) #all values are < 3

mean(vif(multiple_model10)) #1.84


#Homoscadasticity - check if the residual plot looks like random spread

plot(multiple_model10)


#Normality

plot(resid(multiple_model10))

hist(resid(multiple_model10), breaks = 100)

#Durbin-Watson Test

install.packages('lmtest')

library(lmtest)

dwtest(multiple_model10) #1.57


  #Influential cases

cook <- cooks.distance(multiple_model10)

sum(cook > 1) #0
```