Statistical Methods for the Prediction of Housing Prices

Sean Daylor & Michael Eiger

University of Rhode Island

Executive Summary

Dr. Dean De Cock's House Pricing data set contains 2,930 observations across 81 variables pertaining to the prices of houses sold in Ames, Iowa between 2006 and 2010. The objective of this study is to utilize a combination of statistical learning and machine learning methods to predict housing prices using various combinations of nine pre-selected explanatory variables from the data set. After the preprocessing stage, these researchers employed several tree-based regression methods to predict the variable of interest. After training and cross-validating both boosted regression tree and random forest models, the optimal model was the random forests model that was able to a house's price with an average discrepancy of $23,136.94 between the predicted price and actual price among records in the test set. Both models found the total measure of a house's above-ground living area to be the most important predictor variable when determining a house's sales price.

Introduction and Problem Formulation

While the COVID-19 pandemic has compromised the profitability of many industries during 2020, the cost of housing has increased during the same period (Friedman, 2020). It is thus critical to understand how certain variables can contribute to the affordability of a house if one is in the real estate market, especially during uncertain economic times. Consequently, both of these students are aiming to earn a comfortable living to become real-estate owners. Unfortunately, these researchers are unfamiliar with factors that influence the prices of houses.

Given the interest of these researchers, the problem pertains to identifying finite variables that influence the prices of houses. For instance, what impact does the year that a house is built have on the house's market value? Does the square footage of various sections of a house significantly impact its price? The objective is to identify whether these variables, among others, influence housing prices in a way that allows us to predict prices for houses of interest. Whether or not certain variables meaningfully contribute to a house's market value is an essential question to not only these researchers but all of those who are in the real estate market despite ongoing, pandemic-induced economic turbulence.

Background and Literature Review

The House Prices data set was created and organized by Dr. Dean De Cock, a professor at Iowa State University, in 2011 (De Cock, 2011, p. 1). De Cock was seeking an alternative to the Boston data set to give to his students for purposes of regression-based data analyses as a final project. Each record is composed of 81 variables, encompassing 80 independent variables and one dependent variable (i.e., the price of a house). De Cock intended for the data set to provide data on enough predictor variables, 80 in total, for students to conduct multiple or simple linear regression on a wide combination of features to predict house prices (pp. 4-6). However, a review of available literature demonstrates how the data set has been used with alternate statistical learning and machine learning methods. For instance, Fan et al. (2018) trained and 10-fold cross-validated a ridge linear regression model that produced a mean squared error (MSE) of 0.01263 when predicting housing prices in the test set (Fan et al., 2018). Another study by Abbasi (2020) involved the training of a lasso regression model that produced a MSE of 0.0124 when predicting house prices in the test set (Abbasi, 2020). Both studies also tried alternate predictive methods such as random forests and support vector regression models that yielded larger MSE values.

Data Characterization

Although the housing price data set was curated by Dr. De Cock, the data was not collected by him (De Cock, 2011, p. 1). The data was instead collected from the years 2006 to 2010 by the Ames City Assessor's Office (ACAO) in Ames, Iowa. The ACAO recorded data on 113 variables for 3,970 individual property sales that occurred during this period at the close of each sale (p. 2). After being given access to the data upon request, De Cock removed variables and observations that required "special [domain] knowledge," resulting in a data set including 2,930 records with each containing 81 variables. Of the 81 variables in the curated data set, the following were chosen for use in the data analysis:

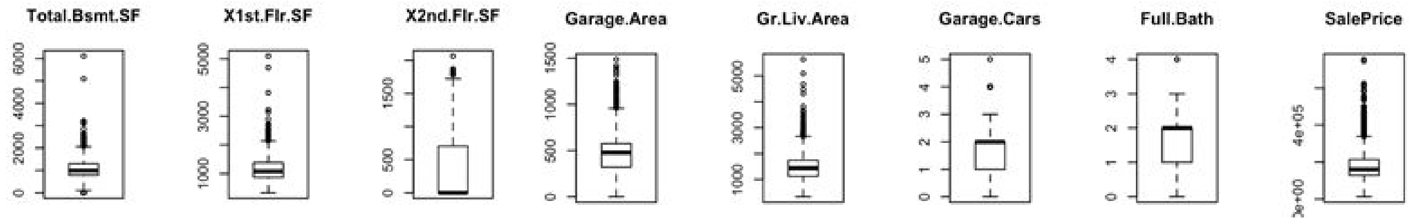
Variable Name	**Description**	**Data Type**	**Unit**
YearBuilt	The year that a house was built	Categorical (Level of Measurement: Ordinal)	N/A
YearRemodAdd	The most recent year a house was remodeled	Categorical (Level of Measurement: Ordinal)	N/A
TotalBsmtSF	The total area of a house's basement	Quantitative (Level of Measurement: Ratio (Continuous))	Square Feet
X1stFlrSF	The total area of a house's first floor	Quantitative (Level of Measurement: Ratio (Continuous))	Square Feet
X2ndFlrSF	The total area of a house's second floor	Quantitative (Level of Measurement: Ratio (Continuous))	Square Feet
GarageArea	The total area of a house's garage	Quantitative (Level of Measurement: Ratio (Continuous))	Square Feet
GrLivArea	The total measure of a house's above-ground living area	Quantitative (Level of Measurement: Ratio (Continuous))	Square Feet
GarageCars	The number of cars that can fit in a house's garage	Quantitative (Level of Measurement: Ratio (Discrete))	# of Cars
FullBath	The number of full-sized bathrooms a house has	Quantitative (Level of Measurement: Ratio (Discrete))	# of Full-Size Bathrooms

SalePrice	The amount that a property sold for	Quantitative (Level of Measurement: Ratio (Continuous))	USD ($)

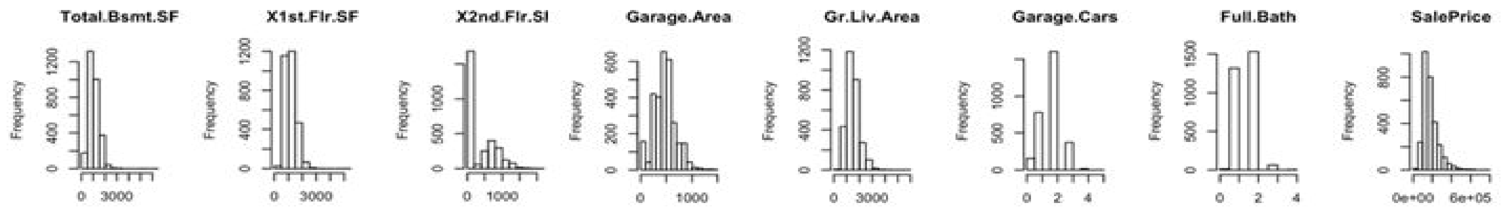
Descriptive Analysis and Visualization

During the pre-processing stage of data preparation, the dataset was checked for a variety of issues. The first check was for both observations that had at least one missing value relative to any given variable. There were just two observations with a single missing value across the 2,930 records, thus the data was quite clean upon receiving it. However, a complete case control was applied to each row to control for the observations with missing values that resulted in the removal of two records. The next check was for duplicate records to ensure that only unique observations were included in the analysis. After applying a duplicate case control to each row, an additional six records were removed from the data set. Eight rows were removed cumulatively after applying the complete and duplicate case controls.

Relative to outliers, the box plots below visualize how a majority of the variables exhibit at least one outlier value that falls outside of the lower fence or upper fence. To control for the outliers, all records that contained at least one value that fell outside of the lower or upper fence relative to any applicable column were removed from the dataset. This resulted in the removal of 352 additional records, leaving 2,570 observations in the dataset.



Lastly, each quantitative feature’s distribution was analyzed visually by generating a histogram and statistically via the Shapiro-Wilks (SW) test. The histograms below show that each visualized feature exhibits a positively-skewed distribution, indicating that the median value of these columns is greater than their mean value.



The SW tests confirmed no features exhibited a normal distribution, the closest was GarageArea, with a p-value of 4.802894e-22. However, if houses without second floors were removed, X2ndFlrSF would also exhibit a shape more consistent with an approximately normal distribution.

Methods

For our analyses, we desired to primarily rely on regression methods. Before considering specific regression methods and their assumptions about the data, if any, we considered said methods to generally be appropriate given that we are attempting to predict a quantitative variable of interest with continuous values (SalePrice). Additionally, most of our predictor variables of interest are quantitative variables with continuous values, which regression methods are equipped to handle. Given this predisposition, we initially considered building a multiple linear regression model. For the theoretical linear regression model, the response variable would be SalePrice, a variable with continuous values, and the predictor variables would be all of the other continuous quantitative features. YearBuilt and YearRemodAdd would also be considered for the model after binning values within each column and coding each as a dummy variable. However, the model's diagnostic (see Appendix) revealed violated assumptions of constant variance, independence of errors, and normality. Outliers, high-leverage points, and collinearity between predictor variables were also found. Prior to building a linear regression model, these issues must be addressed through means such as data transformations, manually removing problematic observations, and dropping one or more features with a high variance inflation factor (VIF) from the regression model.

These researchers initially considered applying controls and transformations to the data set to coerce both homoscedasticity and a normal distribution of the residuals, dropping high VIF features from the regression model, and removing both additional outliers and high-leverage points. However, tree-based regression models were considered an appropriate alternative that did not require validation of the linear regression assumptions. We foresaw a single regression tree model to be too weak of a predictor for our purposes despite its ease of interpretability. Instead, aggregating many regression trees through the use of ensemble tree-based regression methods (e.g., random forests, boosted regression trees) was suspected to yield significantly greater predictive power when predicting SalePrice with the chosen features. A possible limitation of the chosen tree-based regression methods is relative to interpretability, as interpretability of obtained results is more cumbersome using ensemble methods versus the ease of interpreting a single tree. Another limitation to consider is that although regression trees can handle qualitative predictors, the qualitative variables of interest both contain integer values in the original data set. Controls must be taken to ensure that such variables are not interpreted as quantitative variables with continuous values when training the regression models, such as by converting said variables to factors during pre-processing.

Classification methods of analysis are not applicable for predicting the response variable of SalePrice in the data set at present given that said variable is not categorical. SalePrice could be converted to a categorical variable through binning, which would allow for the use of all of the explanatory features in the data set to predict a class label. However, classification methods are currently not of primary interest for use in predicting the variable of interest.

The applicability of classical inference parametric multivariate statistical methods (PMSMs) such as Hotelling's and MANOVA is partly dependent on the assumption of multivariate normality being satisfied. Based on the Chi-Square Q-Q Plot and Henze-Zirkler test (see Appendix), the assumption of multivariate normal among values for specified SalePrice ranges (used due to the current absence of class labels) cannot be satisfied. Therefore, such PMSMs are not applicable to the given data set.

Data Analysis and Main Results

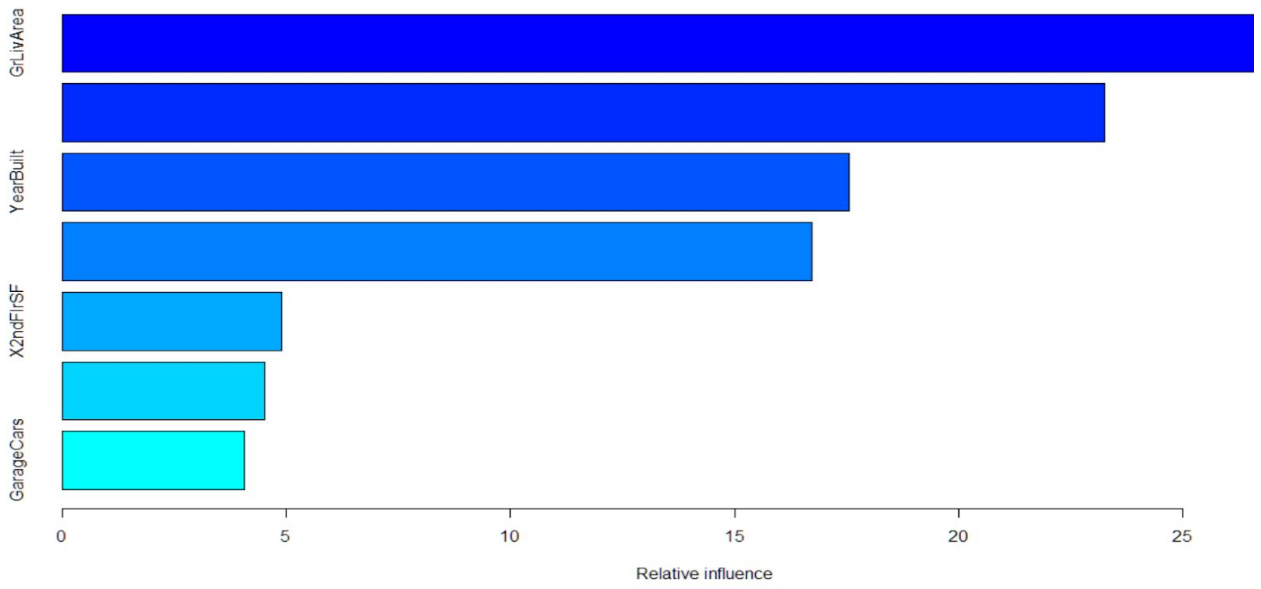
Method #1: Boosted Regression Trees

The first tree-based regression method we chose, boosted regression trees, aimed to predict the response variable SalePrice (i.e., the amount that a property sold for). SalePrice is a quantitative variable with a ratio level of measurement and continuous values that is measured in U.S. dollars. To determine the optimal subset of explanatory variables to include in the model, subset selection was conducted using an adjusted R^2 value. Using the maximum adjusted R^2 value from out of the set of each combination of all possible variables, the following subset of seven explanatory variables were selected:

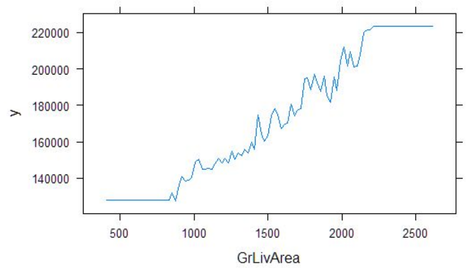
Variable Name	**Description**	**Data Type**	**Unit**
YearBuilt	The year that a house was built	Categorical (Level of Measurement: Ordinal)	N/A
YearRemodAdd	The most recent year a house was remodeled	Categorical (Level of Measurement: Ordinal)	N/A
TotalBsmtSF	The total area of a house's basement	Quantitative (Level of Measurement: Ratio (Continuous))	Square Feet
X2ndFlrSF	The total area of a house's second floor	Quantitative (Level of Measurement: Ratio (Continuous))	Square Feet
GarageArea	The total area of a house's garage	Quantitative (Level of Measurement: Ratio (Continuous))	Square Feet
GrLivArea	The total measure of a house's above-ground living area	Quantitative (Level of Measurement: Ratio (Continuous))	Square Feet
GarageCars	The number of cars that can fit in a house's garage	Quantitative (Level of Measurement: Ratio (Discrete))	# of Cars

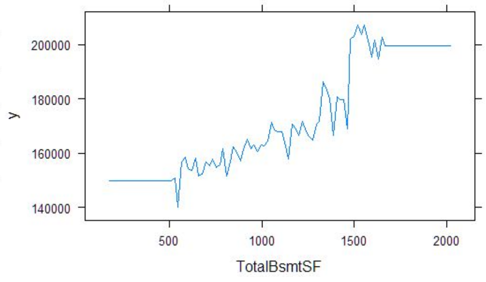
Our general hypothesis is that each variable noted above does meaningfully contribute to the sales price of a house, although we anticipate variance in the degree to which each does so. Boosted regression trees make no assumptions about the data that require validation, thereby allowing users to jump directly into training after data preprocessing and selection of the optimal subset of predictors. We thus began to consider how to tune the hyperparameters to be used when constructing the boosted regression trees immediately post-preprocessing and subset selection. 10-fold cross-validation was initially considered to optimize the following parameters: the number of trees included in the model, the number of splits in each tree, the learning rate of boosting, and the minimum observations per terminal node. However, the number of different hyperparameter combinations, 625, resulted in a 10-fold cross-validation process with a very expensive computational cost that was impractical for timely implementation. Hyperparameter tuning was instead conducted by running a grid search using 80 percent of the training data and using the remaining 20 percent of the training data to calculate out-of-sample estimates of the loss function. The combination of parameters that yielded the lowest root mean square error (RMSE) would be considered the optimal combination. This process resulted in the training of a boosted regression tree model that included 1500 trees in the model with both five splits and a minimum of 50 observations per terminal node in each tree, as well as a shrinkage parameter of 0.2; a combination that returned the minimum RMSE during the grid search.

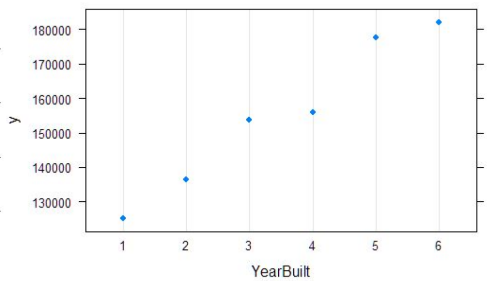
The boosted regression trees model was then trained with the optimal hyperparameters and 10-fold cross-validated to gauge model performance using training data before applying it to the test set, returning a RMSE of $24,461.32. Using the model to make predictions for SalePrice in the test set resulted in a slightly greater RMSE of $25,296.49. The RMSE communicates that the model was able to predict the price of a house with an average discrepancy of $25,296.49 between the predicted price and actual price among records in the test set.



The relative influence plot above was derived from the boosted regression trees model and communicates the relative influence of each predictor that was considered by the model. The plot indicates that the residual sum of squares (RSS) decreased most due to splits over the total area of a house's above-ground living area (GrLivArea) averaged across all 1500 trees, thus rendering said feature to be the model's most important predictor. Calling the summary() function within R (see appendix) indicates on the model reveals that it considered the total area of a house's basement (TotalBsmtSF) and the year during which a house was built (YearBuilt) to be the second and third most important predictors, respectively. The partial dependence plots below illustrate that as each of the model's three most important variables increased in value, the amount of dollars a property sold for generally increased too.







Method #2: Random Forests

The second tree-based regression method we chose, random forests regression, again aimed to predict the response variable SalePrice (i.e., the amount that a property sold for); a quantitative variable with a ratio level of measurement and continuous values that is measured in U.S. dollars. Unlike for the boosted regression trees model, an optimal subset of the explanatory variables was not determined via formal subset selection using adjusted R^2 values. Instead, the random forests algorithm will consider random subsets of the chosen explanatory variables independently at each split. All of the nine explanatory variables originally chosen for inclusion in the research project were considered for this method accordingly and were as follows:

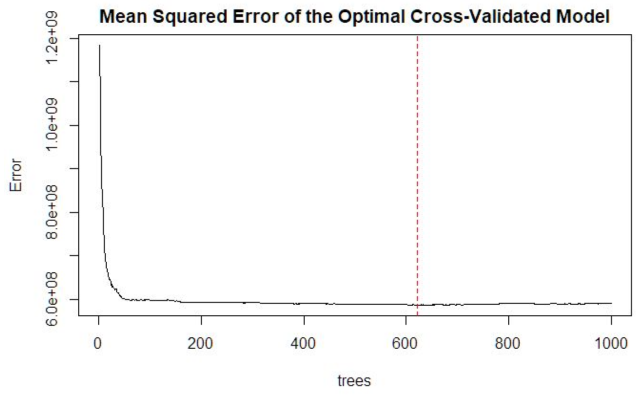
Variable Name	**Description**	**Data Type**	**Unit**
YearBuilt	The year that a house was built	Categorical (Level of Measurement: Ordinal)	N/A
YearRemodAdd	The most recent year a house was remodeled	Categorical (Level of Measurement: Ordinal)	N/A
TotalBsmtSF	The total area of a house's basement	Quantitative (Level of Measurement: Ratio (Continuous))	Square Feet
X1stFlrSF	The total area of a house's first	Quantitative	Square Feet

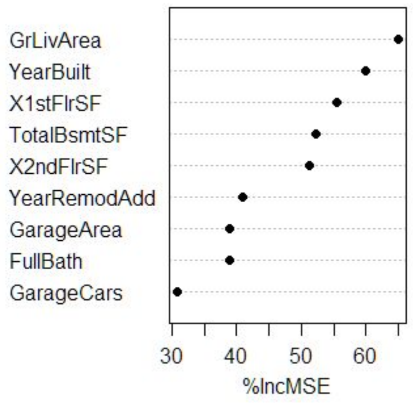
	floor	(Level of Measurement: Ratio (Continuous))	
X2ndFlrSF	The total area of a house's second floor	Quantitative (Level of Measurement: Ratio (Continuous))	Square Feet
GarageArea	The total area of a house's garage	Quantitative (Level of Measurement: Ratio (Continuous))	Square Feet
GrLivArea	The total measure of a house's above-ground living area	Quantitative (Level of Measurement: Ratio (Continuous))	Square Feet
GarageCars	The number of cars that can fit in a house's garage	Quantitative (Level of Measurement: Ratio (Discrete))	# of Cars
FullBath	The number of full-sized bathrooms a house has	Quantitative (Level of Measurement: Ratio (Discrete))	# of Full-Size Bathrooms

Our general hypothesis is again that each variable noted above does meaningfully contribute to the sales price of a house, still anticipating variance in the degree to which each does so. Like boosted regression trees, random forests regression models make no assumptions about the data that require validation, allowing users to jump directly into training after data preprocessing. However, before building a random forests model, optimal hyperparameters had to be established regarding both the number of trees included in the model and the number of predictor variables to be randomly considered at each split in each tree. Using 10-fold cross-validation to tune said parameters, the optimal number of trees to include in the model was determined to be 1000 with two random predictor variables being considered at each split within each tree.

The best performing 10-fold cross-validated random forests regression model using the optimal hyperparameters had an average RMSE of $24,489.63 on the training data with the minimum average RMSE noted at around 620 trees as seen in the error curve below. The model's RMSE on the training data was very similar to the RMSE of the optimal boosted regression trees model on the training set ($24,461.32), albeit with a slightly greater discrepancy between the predicted and actual prices. Using the model to make predictions for SalePrice in the test set resulted in a lesser RMSE of $23,136.94; an improvement from boosted regression trees model's RMSE when predicting house prices in the test set

($25,296.49). The model was thus able to predict the price of a house with an average discrepancy of $23,136.94 between the predicted price and actual price among records in the test set.





The relative influence plot above was derived from the random forests regression model and communicates the relative influence of each predictor that was considered by the model. The plot indicates that the RSS decreased most due to splits over the total area of a house's above-ground living area (GrLivArea) averaged across all 1000 trees, thus making said feature the model's most important predictor. The relative influence plot reveals that the model considered the year a house was built (YearBuilt), the total area of a house's first floor (X1stFlrSF), and the total area of a house's basement (TotalBsmtSF), to be the second, third, and fourth most important predictors, respectively.

Conclusions

Based on the results of the tree-based regression analyses, the optimal model for predicting house prices in the test set was the random forests regression model. Said model was able to predict the price of a house with an average discrepancy of $23,136.94 between the predicted price and actual price among records in the test set, which bested the performance of the boosted regression trees model. Both models considered the total area of a house's above ground living area to be the most important predictor of the amount at which houses in the test set sold for. Both models also considered the year during which a house was built and the total square footage of a house's basement to be among the most important predictors of a house's price. The models cumulatively communicate that each of the originally selected explanatory variables meaningfully contribute to the price of a house, albeit to varying degrees. Although all variables yielded some degree of importance in determining house price, the least important variable in both models was the number of cars that can fit in a house's garage.

Research Directions

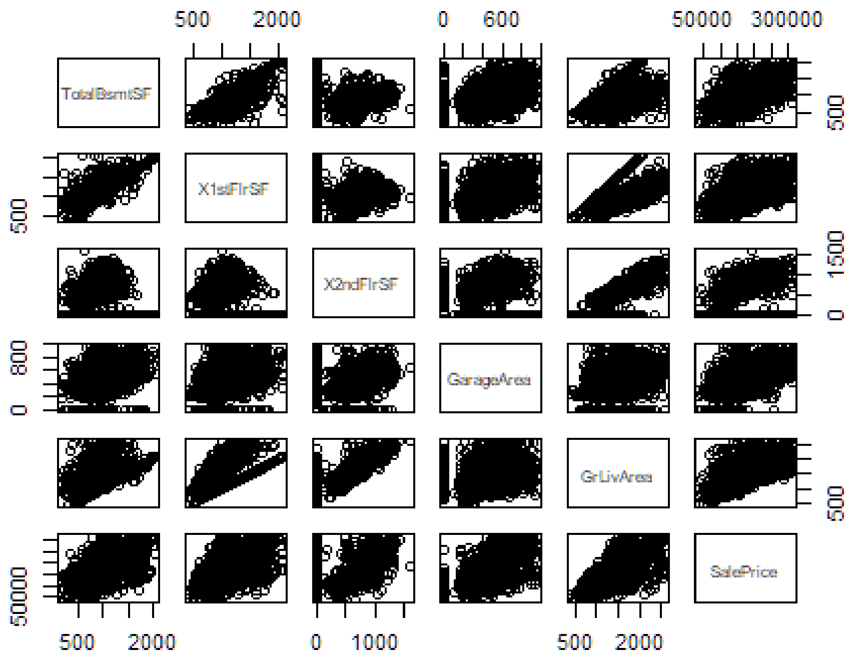
The primary limitation regarding this research is that just nine of the 80 features were selected for analysis without having statistically determined optimal features to include in a given model. Additionally, the dataset pertains only to Ames, Iowa, and having been aggregated between 2006 and 2010, may no longer be applicable to the current year or outside of Ames, Iowa. Future research should consider including additional features when constructing models, statistical methods to determine optimal features for inclusion, or expanding the data set with additional observations from the same period and region. Other considerations may include expanding the dataset to encompass samples from a more diverse geographic region, or or updating the dataset to include home sales from 2010 to the current year.

(NOTE: The Appendix begins on the following page)

```
# Import the necessary packages
library(dplyr)
library(plyr)
library(lmtest)
library(car)
library(olsrr)
library(MVN)

# Read in the training data set
df = read.csv('ames.csv', header=T)
df = df[,c("Year.Built", "Year.Remod.Add", "Total.Bsmt.SF",
           "X1st.Flr.SF", "X2nd.Flr.SF", "Garage.Area",
           "Gr.Liv.Area", "Garage.Cars", "Full.Bath", "SalePrice")]
names(df) = c("YearBuilt", "YearRemodAdd", "TotalBsmtSF",
            "X1stFlrSF", "X2ndFlrSF", "GarageArea",
            "GrLivArea", "GarageCars", "FullBath", "SalePrice")

# Drop any rows with at least one missing value in any given column
df = na.omit(df) # Dropped 2 rows

# Drop any duplicate rows
df = distinct(df) # Dropped 6 rows

# Dropping outliers (352 rows dropped)
df = subset(df, TotalBsmtSF <= (quantile(df$TotalBsmtSF, p=.75) +
(1.5*IQR(df$TotalBsmtSF))))
df = subset(df, X1stFlrSF <= (quantile(df$X1stFlrSF, p=.75) +
(1.5*IQR(df$X1stFlrSF))))
df = subset(df, X2ndFlrSF <= (quantile(df$X2ndFlrSF, p=.75) +
(1.5*IQR(df$X2ndFlrSF))))
df = subset(df, GarageArea <= (quantile(df$GarageArea, p=.75) +
(1.5*IQR(df$GarageArea))))
df = subset(df, GrLivArea <= (quantile(df$GrLivArea, p=.75) +
(1.5*IQR(df$GrLivArea))))
df = subset(df, GarageCars <= (quantile(df$GarageCars, p=.75) +
(1.5*IQR(df$GarageCars))))
df = subset(df, FullBath <= (quantile(df$FullBath, p=.75) +
(1.5*IQR(df$FullBath))))
df = subset(df, SalePrice <= (quantile(df$SalePrice, p=.75) +
(1.5*IQR(df$SalePrice))))

df = subset(df, TotalBsmtSF >= (quantile(df$TotalBsmtSF, p=.25) -
(1.5*IQR(df$TotalBsmtSF))))
df = subset(df, X1stFlrSF >= (quantile(df$X1stFlrSF, p=.25) -
(1.5*IQR(df$X1stFlrSF))))
df = subset(df, X2ndFlrSF >= (quantile(df$X2ndFlrSF, p=.25) -
(1.5*IQR(df$X2ndFlrSF))))
df = subset(df, GarageArea >= (quantile(df$GarageArea, p=.25) -
(1.5*IQR(df$GarageArea))))
df = subset(df, GrLivArea >= (quantile(df$GrLivArea, p=.25) -
(1.5*IQR(df$GrLivArea))))
df = subset(df, GarageCars >= (quantile(df$GarageCars, p=.25) -
(1.5*IQR(df$GarageCars))))
df = subset(df, FullBath >= (quantile(df$FullBath, p=.25) -
(1.5*IQR(df$FullBath))))
df = subset(df, SalePrice >= (quantile(df$SalePrice, p=.25) -
(1.5*IQR(df$SalePrice))))

# Define binning cuts
breaks = c(-Inf, 1900, 1925, 1950, 1975, 2000, Inf)

# Define binning factor labels
labels <- c("1","2", "3", "4", "5", "6")

# Bin YearBuilt and YearRemodAdd and return factor
df$YearBuilt <- cut(df$YearBuilt, breaks = breaks, labels = labels)
df$YearRemodAdd <- cut(df$YearRemodAdd, breaks = breaks, labels = labels)
```

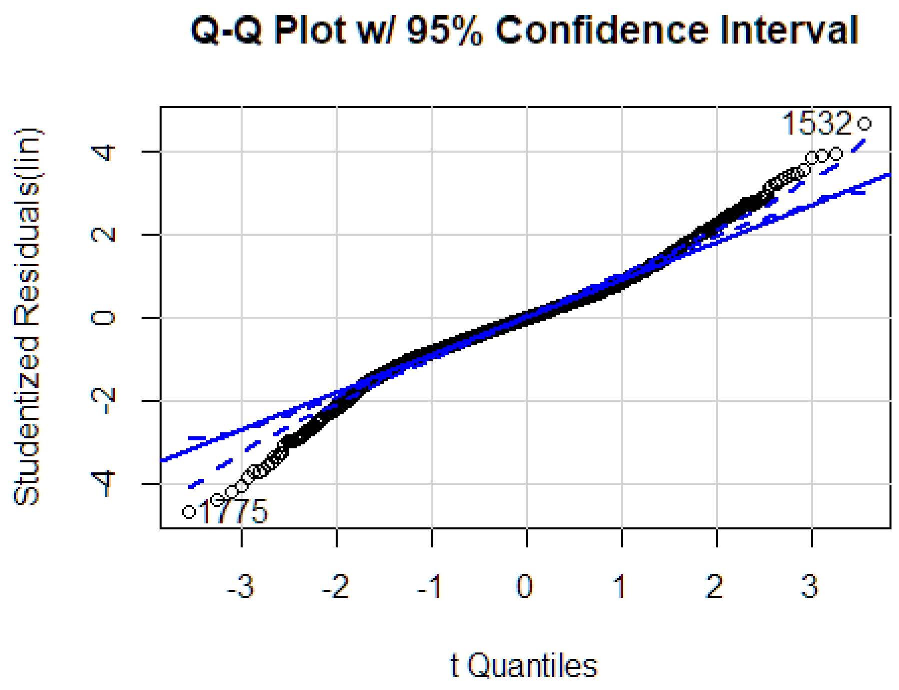
LINEAR REGRESSION ASSUMPTION TESTING

```
# 1) RANDOM SAMPLE: Simple random sampling was used to construct the training
data from the set of all observations in the data set. The assumption of
Random Sampling is thus satisfied.

# 2) LINEARITY:
pairs(df[,c(3:7,10)])
```



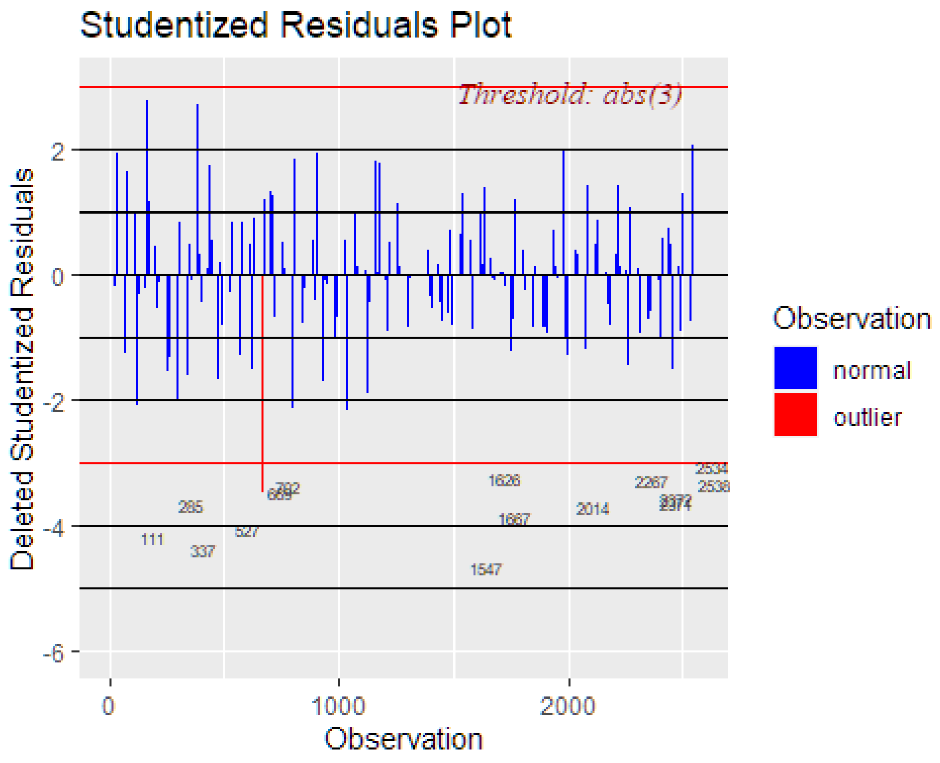
```
# Based on visual inspection of the generated scatterplot matrix, linear
relationships with strengths ranging from weak to moderate appear to be
exhibited between the response variable (SalePrice) and each of the
respective quantitative predictor variables considered for inclusion in the
linear regression model pre-optimal subset selection. The assumption of
linearity is thereby satisfied. Of note, YearBuilt and YearRemodAdd are
expectedly not visualized here given that they have been converted to
factors, but would also be included in the model in the form of dummy
variables.

# 3) ZERO MEAN:
lin = lm(SalePrice ~ YearBuilt + YearRemodAdd + TotalBsmtSF + X1stFlrSF +
X2ndFlrSF
         + GarageArea + GrLivArea, data = df)
mean(residuals(lin))

## [1] 2.706504e-13

# The mean of all of the model's error terms is very close to 0 (-
0.0000000000002706504). While not exactly 0, this result indicates that the
error distribution is approximately centered at 0 and thus the assumption of
zero mean is (approximately) satisfied.

# 4) CONSTANT VARIANCE:
bptest(lin)

##
##  studentized Breusch-Pagan test
##
## data:  lin
## BP = 301.45, df = 13, p-value < 2.2e-16

# The Breusch-Pagan test (conducted below) produces a BP value (301.45) with
a p-value of <0.001. The BP test result forces us to reject the null
hypothesis that the variance among error terms from the model are all
equal/constant at a 99.9% confidence level. The results therefore communicate
that heteroscedasticity (i.e., non-constant variance among error terms) in
the regression model is present with a 99.9% confidence level. This finding
indicates that non-constant variance among error terms is present in the
model, and thus the model does NOT satisfy the assumption of constant
variance.

# 5) INDEPENDENCE OF ERRORS:
dwtest(lin)

##
##  Durbin-Watson test
##
## data:  lin
## DW = 1.5819, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

```r
# The Durbin-Watson test returns a value of 1.5819 with a p-value <0.001.
Given this result, we can NOT accept the null hypothesis that the true
autocorrelation of error term's from the model is equal to 0 at a 95%
confidence level. The true autocorrelation of the error terms from the model
is thus greater than 0 and so the assumption that error terms from the model
are independent of each other is NOT satisfied.

# 6) NORMALITY OF ERRORS:
qqPlot(lin, main = 'Q-Q Plot w/ 95% Confidence Interval')
```

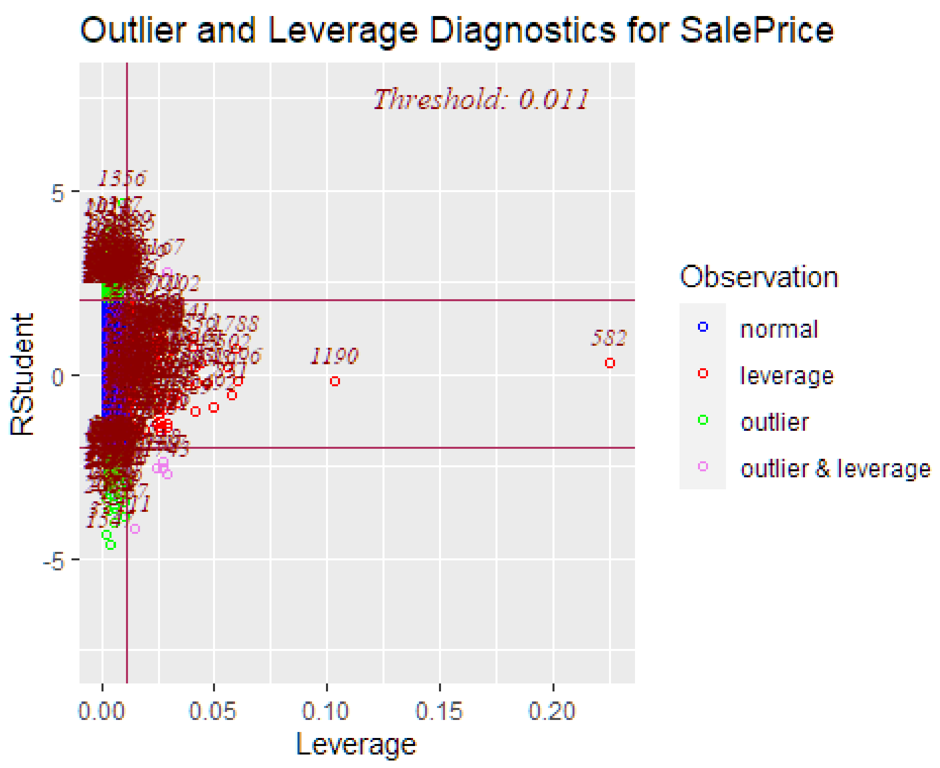


```
## 1532 1775
## 1356 1547
```

```r
# The Q-Q Plot suggests approximate normality among the standardized
residuals for all theoretical quantiles between -3 and 2 with a 95%
confidence level, although a deviation of values from the reference line
between quantiles (-3 and -2) and (2 and 3) suggests some degree of a non-
normal distribution for residuals within that range at the same confidence
level. The assumption that all random errors from the model follow a normal
distribution thus is NOT satisfied.

# OUTLIERS:
ols_plot_resid_stud(lin)
```



```
# Many studentized residuals fall outside the threshold of -3 to 3,
suggesting that there are outlying observations in the data set that need to
be dropped.

# HIGH LEVERAGE POINTS:
ols_plot_resid_lev(lin)
```

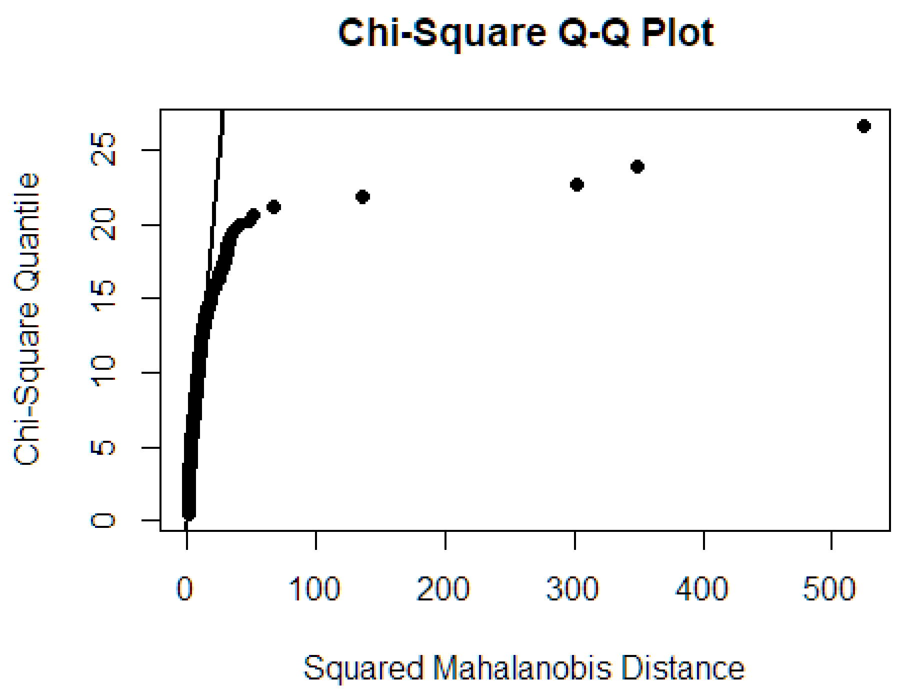


```
# Many observations were found to have high leverage (see plot points in red
and pink) and thus are determined to have a high impact on how the model's
estimated regression line was fit. Analyzing the context of each observation
found to have high leverage may be of value if aiming to optimize the
fit/predictive power of the multiple linear regression model. Of note, the
generated plot uses residuals (not studentized residuals) to determine
outliers, which accounts for the discrepancy in outlying observations found
between this plot and the studentized residuals plot from a previous line.

# COLLINEARITY:
vif(lin) # vif() was called to calculate the presence of collinearity among
predictor variables in the regression model. Several predictor variables
returned a variance inflation factor (VIF) of greater than 5 (YearBuilt,
YearRemodAdd, X1stFlrSF, X2nsFlySF, GrLivArea), indicating the presence of
collinearity among the implicated predictors. Dropping at least one of the
collinear predictors or combining them into a single  predictor may be
effective at mitigating collinearity in order to optimize the fit/predictive
power of the multiple linear regression model.

##                   GVIF Df GVIF^(1/(2*Df))
## YearBuilt    20.321519  5        1.351436
## YearRemodAdd 15.431040  3        1.577851
## TotalBsmtSF   4.412158  1        2.100514
## X1stFlrSF    55.886253  1        7.475711
## X2ndFlrSF    80.059925  1        8.947621
## GarageArea    1.519142  1        1.232535
## GrLivArea    79.986862  1        8.943537
```

OVERALL LIST OF ISSUES RELATED TO THE MODEL:
- Assumption of Constant Variance NOT satisfied (heteroscedasticity is present in the regression model with a 99.9% confidence level)
- Assumption of independence of errors NOT satisfied (autocorrelation is present in the regression model with a 99.9% confidence level)
- Assumption of Normality NOT satisfied (based on the results of the Q-Q Plot, which suggested some degree of a non-normal distribution for residuals within the 2nd to 3rd quantile range at the 95% confidence level)
- Many outliers and high-leverage points found (based on the respective residual and studentized residual plots)
- Collinearity noted relative to several of the predictor variables included in the regression model based on their variance inflation factor

MULTIVARIATE NORMAL ASSUMPTION TESTING

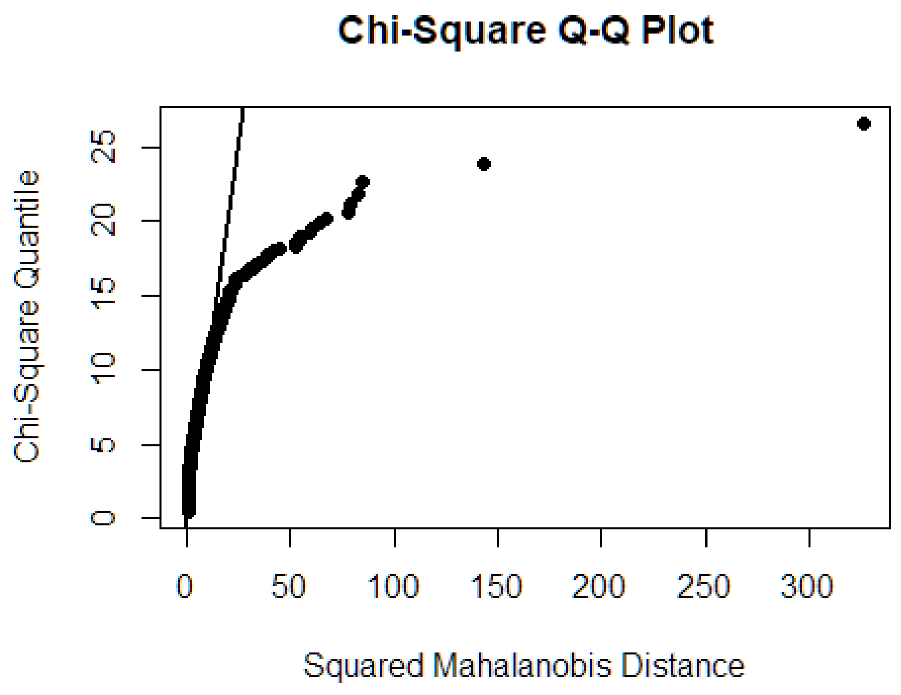
```
mvn(subset(df[,3:9], df$SalePrice >= median(df$SalePrice)), mvnTest = "hz",
multivariatePlot = "qq")$multivariateNormality
```



```
##              Test       HZ p value MVN
## 1 Henze-Zirkler 35.91345       0  NO
```

```
# Outputs the result of the Henze-Zirkler test of Multivariate normality.
Considering the p-value, the results indicate that values contained within
the rows where SalePrice is greater than or equal to the median of values in
the column DO NOT follow a multivariate normal distribution with a
significance level of 0.05 Also output a Chi-Square Q-Q Plot, which exhibits
```

```
that the Mahalanobis Distance^2 metric for values begins to skew away from
the reference line beginning roughly at the 15th Chi-Square quantile, further
validating the conclusion of the Henze-Zirkler test regarding the absence of
multivariate normal.

mvn(subset(df[,3:9], df$SalePrice < median(df$SalePrice)), mvnTest = "hz",
multivariatePlot= "qq")$multivariateNormality
```



```
##             Test       HZ p value MVN
## 1 Henze-Zirkler 33.74759       0  NO

# Outputs the result of the Henze-Zirkler test of Multivariate normality.
Considering the p-value, the results indicate that values contained within
the rows where SalePrice is less than the median of values in the column DO
NOT follow a multivariate normal distribution with a significance level of
0.05. Also outputs a Chi-Square Q-Q Plot, which exhibits that the Mahalanobis
Distance^2 metric for values begins to skew away from the reference line
beginning roughly at the 13th Chi-Square quantile, further validating the
conclusion of the Henze-Zirkler test regarding the absence of multivariate
normal.
```

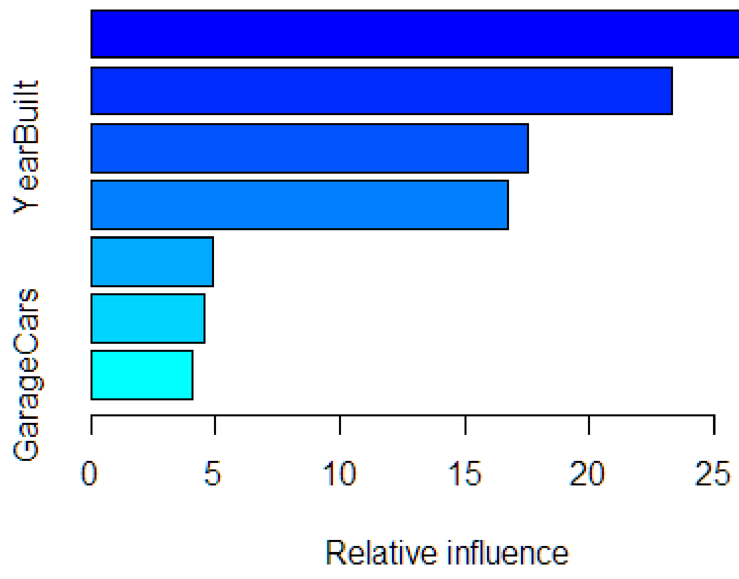
SETUP & PRE-PROCESSING

```r
# Import the necessary packages
library(dplyr) # Imports distinct()
library(leaps) # Imports regsubets()
library(randomForest) # Imports randomForest()
library(mltools) # Imports rmse() and rmsle()
library(e1071) # Imports tune()
library(gbm) # Imports gbm()

# Read in the training data set
df = read.csv('ames.csv', header=T)
df = df[,c("Year.Built", "Year.Remod.Add", "Total.Bsmt.SF",
           "X1st.Flr.SF", "X2nd.Flr.SF", "Garage.Area",
           "Gr.Liv.Area", "Garage.Cars", "Full.Bath", "SalePrice")]
names(df) = c("YearBuilt", "YearRemodAdd", "TotalBsmtSF",
            "X1stFlrSF", "X2ndFlrSF", "GarageArea",
            "GrLivArea", "GarageCars", "FullBath", "SalePrice")

# Drop any rows with at least one missing value in any given column
df = na.omit(df) # Dropped 2 rows

# Drop any duplicate rows
df = distinct(df) # Dropped 6 rows

# Dropping outliers (352 rows dropped)
df = subset(df, TotalBsmtSF <= (quantile(df$TotalBsmtSF, p=.75) +
(1.5*IQR(df$TotalBsmtSF))))
df = subset(df, X1stFlrSF <= (quantile(df$X1stFlrSF, p=.75) +
(1.5*IQR(df$X1stFlrSF))))
df = subset(df, X2ndFlrSF <= (quantile(df$X2ndFlrSF, p=.75) +
(1.5*IQR(df$X2ndFlrSF))))
df = subset(df, GarageArea <= (quantile(df$GarageArea, p=.75) +
(1.5*IQR(df$GarageArea))))
df = subset(df, GrLivArea <= (quantile(df$GrLivArea, p=.75) +
(1.5*IQR(df$GrLivArea))))
df = subset(df, GarageCars <= (quantile(df$GarageCars, p=.75) +
(1.5*IQR(df$GarageCars))))
df = subset(df, FullBath <= (quantile(df$FullBath, p=.75) +
(1.5*IQR(df$FullBath))))
df = subset(df, SalePrice <= (quantile(df$SalePrice, p=.75) +
(1.5*IQR(df$SalePrice))))

df = subset(df, TotalBsmtSF >= (quantile(df$TotalBsmtSF, p=.25) -
(1.5*IQR(df$TotalBsmtSF))))
df = subset(df, X1stFlrSF >= (quantile(df$X1stFlrSF, p=.25) -
(1.5*IQR(df$X1stFlrSF))))
df = subset(df, X2ndFlrSF >= (quantile(df$X2ndFlrSF, p=.25) -
(1.5*IQR(df$X2ndFlrSF))))
df = subset(df, GarageArea >= (quantile(df$GarageArea, p=.25) -
(1.5*IQR(df$GarageArea))))
df = subset(df, GrLivArea >= (quantile(df$GrLivArea, p=.25) -
(1.5*IQR(df$GrLivArea))))
df = subset(df, GarageCars >= (quantile(df$GarageCars, p=.25) -
(1.5*IQR(df$GarageCars))))
df = subset(df, FullBath >= (quantile(df$FullBath, p=.25) -
(1.5*IQR(df$FullBath))))
df = subset(df, SalePrice >= (quantile(df$SalePrice, p=.25) -
(1.5*IQR(df$SalePrice))))

# Define binning cuts
breaks = c(-Inf, 1900, 1925, 1950, 1975, 2000, Inf)

# Define binning factor labels
labels <- c("1","2", "3", "4", "5", "6")

# Bin YearBuilt and YearRemodAdd and return factor
df$YearBuilt <- cut(df$YearBuilt, breaks = breaks, labels = labels)
df$YearRemodAdd <- cut(df$YearRemodAdd, breaks=breaks, labels=labels)

# Split into training and test sets
set.seed(1)
train = sample(1:nrow(df), 2056) # 80% Training - 20% Test split
SalePrice.train = df[train,]
SalePrice.test = df[-train,]
```

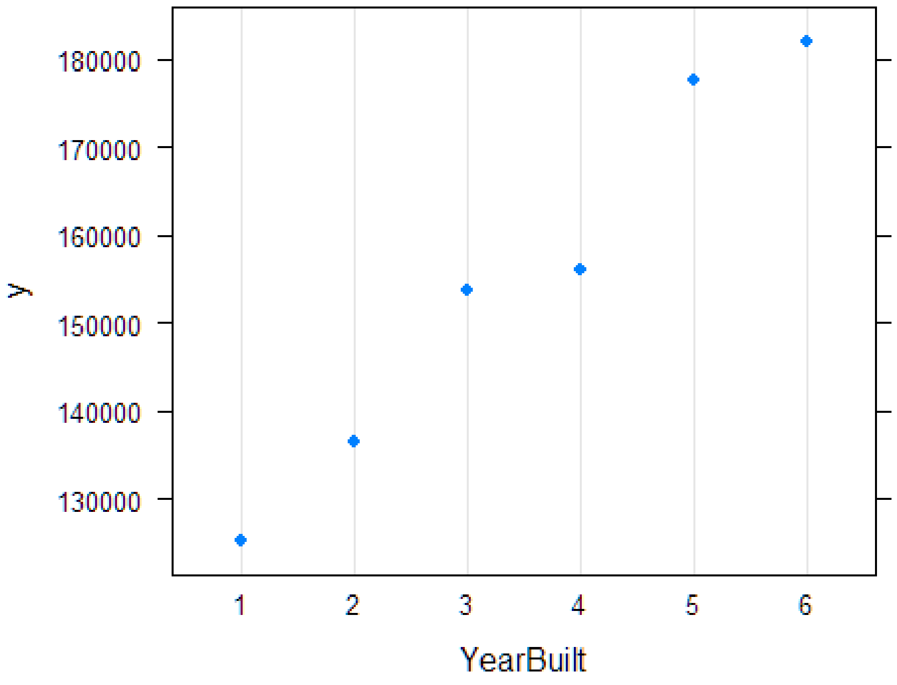
BOOSTED REGRESSION TREES

```
# Selecting the optimal subset of predictor variables using Adjusted R^2
rf.subsets = regsubsets(SalePrice~., data=df, nvmax=15) # nvmax=15 to ensure
that the dummies for each categorical variable is also considered in subset
selection

## Reordering variables and trying again:

summary(rf.subsets)$adjr2

##  [1] 0.4282210 0.5915109 0.6623252 0.7107679 0.7374390 0.7458434 0.7505601
##  [8] 0.7547116 0.7599637 0.7624524 0.7646448 0.7663448 0.7671023 0.7671388
## [15] 0.7670484

which.max(summary(rf.subsets)$adjr2) # Optimal subset has Adjusted R^2 of
0.7670484

## [1] 14

coef(rf.subsets,14)

##   (Intercept)    YearBuilt2    YearBuilt3    YearBuilt4    YearBuilt5
## -30831.461840  19673.009200  34567.141313  36509.136211  49100.467811
##    YearBuilt6 YearRemodAdd4 YearRemodAdd5   TotalBsmtSF     X2ndFlrSF
##   57230.988783   3036.697778  15819.022015     38.358060      3.941056
##     GarageArea     GrLivArea    GarageCars YearRemodAdd2 YearRemodAdd6
##      28.802781     53.677158   7672.315043      0.000000  23985.927784

# Tuning to determine optimal hyperparameters. Below grid search has 625
combinations, far too many to cross validate each model. Instead, using 80%
for training and remaining 20% used to calculate out-of-sample estimates of
the loss function

set.seed(1)

params = expand.grid(n.trees = c(500, 1000, 1500, 2000, 2500), shrinkage =
c(0.001, 0.01, 0.05, 0.1, 0.2),
                     interaction.depth = c(1, 2, 3, 4, 5), n.minobsinnode =
c(10, 20, 30, 40, 50),
                     optimal_trees = 0, min_RMSE = 0)

for(i in 1:nrow(params)) {
  boost.SalePrice.tune = gbm(
    formula = SalePrice~YearBuilt + YearRemodAdd + TotalBsmtSF + X2ndFlrSF +
GarageArea + GrLivArea + GarageCars,
    distribution = "gaussian",
    data = SalePrice.train,
    n.trees = params$n.trees[i],
    interaction.depth = params$interaction.depth[i],
    shrinkage = params$shrinkage[i],
    n.minobsinnode = params$n.minobsinnode[i],
    train.fraction = 0.8,
  )
  params$optimal_trees[i] = which.min(boost.SalePrice.tune$valid.error)
  params$min_RMSE[i] = sqrt(min(boost.SalePrice.tune$valid.error))
}

# The optimal parameters for the boosted regression tree model
best.models = arrange(params, min_RMSE)
best.models[1,]

##     n.trees shrinkage interaction.depth n.minobsinnode optimal_trees
min_RMSE
## 1      1500     0.200                 5             50           182
22866.62

# Best RMSE = 22866.62
# n.trees = 1500
# shrinkage = 0.2
# interaction.depth = 5
# n.minobsinnode = 50


# Training the model with optimal hyperparameters
set.seed(1)
```

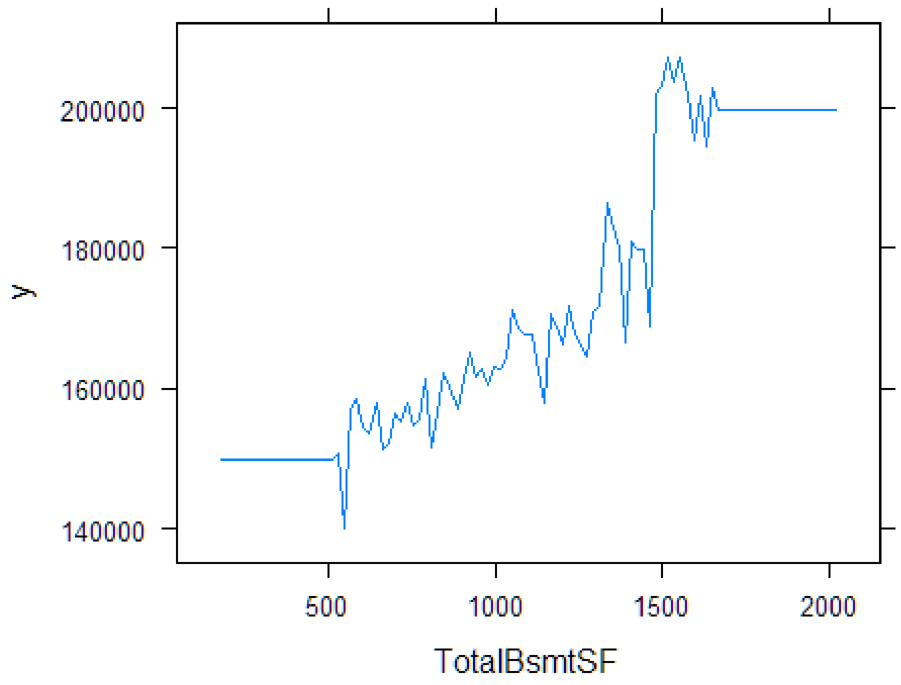
```r
boost.SalePrice = gbm(
  formula = SalePrice~YearBuilt + YearRemodAdd + TotalBsmtSF + X2ndFlrSF +
GarageArea + GrLivArea + GarageCars,
  distribution = "gaussian",
  data = SalePrice.train,
  n.trees = best.models$n.trees[1],
  interaction.depth = best.models$interaction.depth[1],
  shrinkage = best.models$shrinkage[1],
  n.minobsinnode = best.models$n.minobsinnode[1],
  cv.folds = 10,
)

# Calculating Root Mean Square Error (RMSE) to gauge model performance on the
training/validation sets
sqrt(min(boost.SalePrice$cv.error)) # RMSE = 24461.32

## [1] 24461.32

# Make predictions with the cross-validated boosted regression trees model on
the test data
boost.pred = predict(boost.SalePrice,newdata=SalePrice.test, n.trees =
best.models$n.trees[1])

# Calculating Root Mean Square Error (RMSE) to gauge model performance on the
test set
rmse(preds = boost.pred, actuals = SalePrice.test[,10]) # RMSE = 25296.49

## [1] 25296.49

# Will produce a relative influence plot and the relative influence of each
predictor included in the model
summary(boost.SalePrice)
```



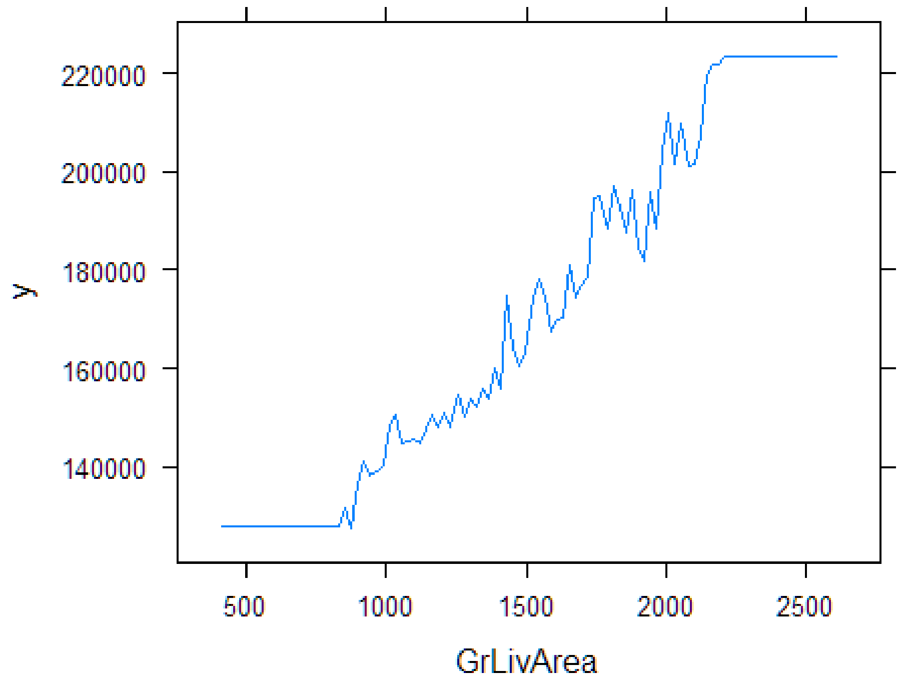
```
##                       var   rel.inf
## GrLivArea       GrLivArea 28.983926
## TotalBsmtSF   TotalBsmtSF 23.262591
## YearBuilt       YearBuilt 17.567915
## GarageArea     GarageArea 16.716818
## X2ndFlrSF       X2ndFlrSF  4.892131
## YearRemodAdd YearRemodAdd  4.519460
## GarageCars     GarageCars  4.057159

# Will produce a partial dependence plot to better understand how each
predictor included in the mode influences SalePrice
plot(boost.SalePrice,i="YearBuilt")
```



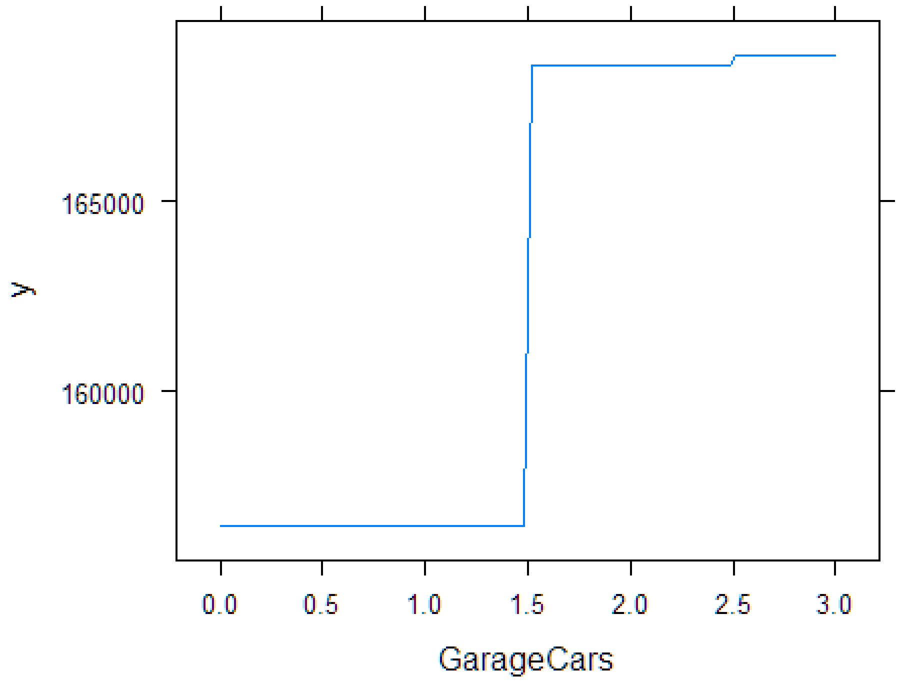
```
plot(boost.SalePrice,i="TotalBsmtSF")
```



```r
plot(boost.SalePrice,i="GrLivArea")
```



```r
plot(boost.SalePrice,i="GarageCars")
```

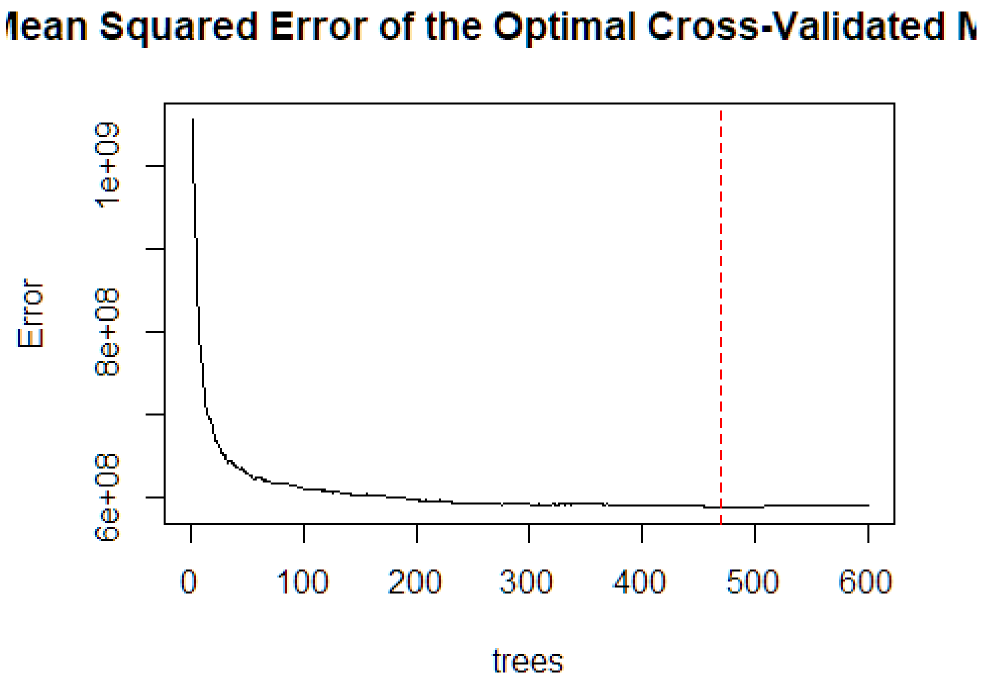


RANDOM FORESTS

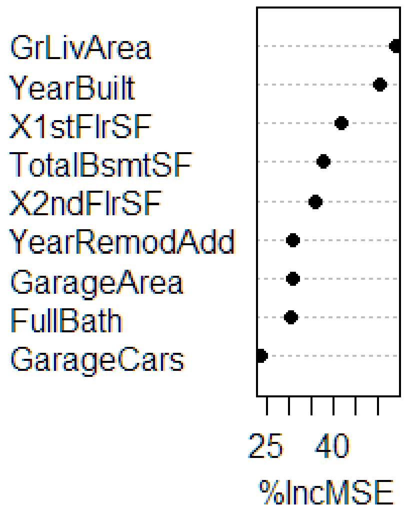
```
# Training + 10-Fold Cross-Validation to establish optimal hyperparameters
set.seed(1)
tune.out = tune(randomForest,SalePrice~.,
data=SalePrice.train,importance=TRUE,ranges=list(ntree=seq(200,1000,200),mtry
=seq(1,5,1)))
summary(tune.out) # Best Hyperparameters: ntree = 1000, mtry = 2

##
## Parameter tuning of 'randomForest':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  ntree mtry
##    1000    2
##
## - best performance: 599741830
##
## - Detailed performance results:
##    ntree<dbl>   mtry<dbl> error<dbl>  dispersion<dbl>

200   1     633798042   111015142

400   1     635142816   107157484

600   1     633198546   108252014
```

```
800   1     632292207   107760659

1000  1     634267142   107777179

200   2     601763210   116422909

400   2     600020585   113528518

600   2     600364121   109833406

800   2     601106168   114742782

1000  2     599741830   112911481

200   3     609901777   119637604

400   3     606197482   117800781

600   3     610420838   120074071

800   3     607355970   117986476

1000  3     608988977   120495414

200   4     617355675   125620827

400   4     614363205   124759980

600   4     614874797   123319809

800   4     613287878   123684744

1000  4     614089909   123021969

200   5     621210008   129144354

400   5     618975232   128168941

600   5     618412975   125993266

800   5     616368784   129350213

1000  5     617466108   127528236
bestmod.rf = tune.out$best.model
tune.out$best.performance

## [1] 599741830

# Calculating Root Mean Square Error (RMSE) to gauge model performance on the
training data
sqrt(599741830) # RMSE = 24489.63

## [1] 24489.63
```

```
# Make predictions with the cross-validated random forests model on the test
data
rf.pred = predict(bestmod.rf,SalePrice.test)

# Calculating Root Mean Square Error (RMSE) to gauge model performance on the
test data
rmse(preds = rf.pred, actuals = SalePrice.test[,10]) # RMSE = 23136.94

## [1] 23136.94

# Plot the MSE relative to the number of trees in the cross-validated model.
abline() will be used to highlight the number of trees that yielded the
minimum MSE
plot(bestmod.rf, main = "Mean Squared Error of the Optimal Cross-Validated
Model")
abline(v = which.min(bestmod.rf$mse), col="red", lty="dashed")
```



```
# Will produce a relative influence plot and a vector quantifying the
relative influence of each predictor included in the model
varImpPlot(bestmod.rf, main = "Variable Importance Plot", pch = 19)
```



```
sort(importance(bestmod.rf)[,1])

##GarageCars      FullBath   GarageArea  YearRemodAdd    X2ndFlrSF

##    30.87615      38.78650      38.85327      40.81604      51.14517

## TotalBsmtSF     X1stFlrSF     YearBuilt     GrLivArea

##    52.25244      55.39857      60.03034      64.91533
```

Works Cited

Abbasi, S. (2020). Advanced Regression Techniques Based Housing Price Prediction Model. 10.13140/RG.2.2.18572.87684.

De Cock, D. (2011). Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project. *Journal of Statistics Education, 19*(3). doi:10.1080/10691898.2011.11889627

Fan, C., Cui, Z., & Zhong, X. (2018). House Prices Prediction with Machine Learning Algorithms. *Proceedings of the 2018 10th International Conference on Machine Learning and Computing - ICMLC 2018*. doi:10.1145/3195106.3195133

Friedman, N. (2020, May 05). Why Home Prices Are Rising During the Pandemic. Retrieved from https://www.wsj.com/articles/why-home-prices-are-rising-during-the-pandemic-11588671002