Predicting Home Prices in Ames, Iowa

STAT 4620 Final Project

Cody Collins, Charles Costanzo, Kevin Gonzalez, Ashley Zhang

December 11, 2023


[1]

Part I: Exploratory Data Analysis

For exploratory data analysis, the first step was to explore each variable in the Ames, Iowa real estate data set. In order to get a sense of the characteristics of the data set overall, we combined the test and training data sets for exploratory data analysis, matching based on each data set's column names. The variables named `1stFlrSF`, `2ndFlrSF`, and `3SsnPorch` in the training data set were named `X1stFlrSF`, `X2ndFlrSF`, `X3SsnPorch` in the test data set, so we set the names in the training data set to match the test data set to enable merging the datasets. Another issue with the data was that one of the values for `GarageYrBlt` was listed as the year 2207, which would be impossible. Since `YearBuilt` indicated that the house was built in 2006, this value was changed to 2007.

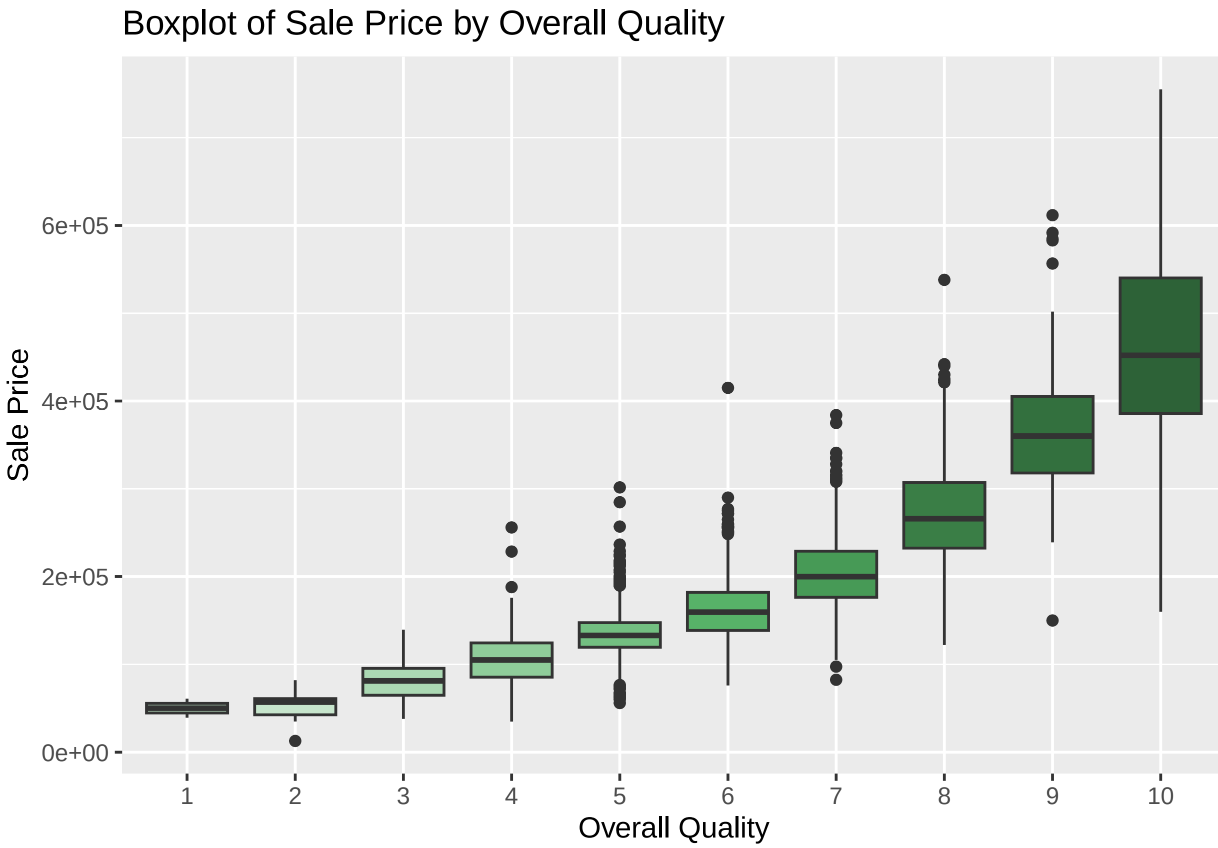
[1]https://www.cityofames.org/about-ames

Data Structure:

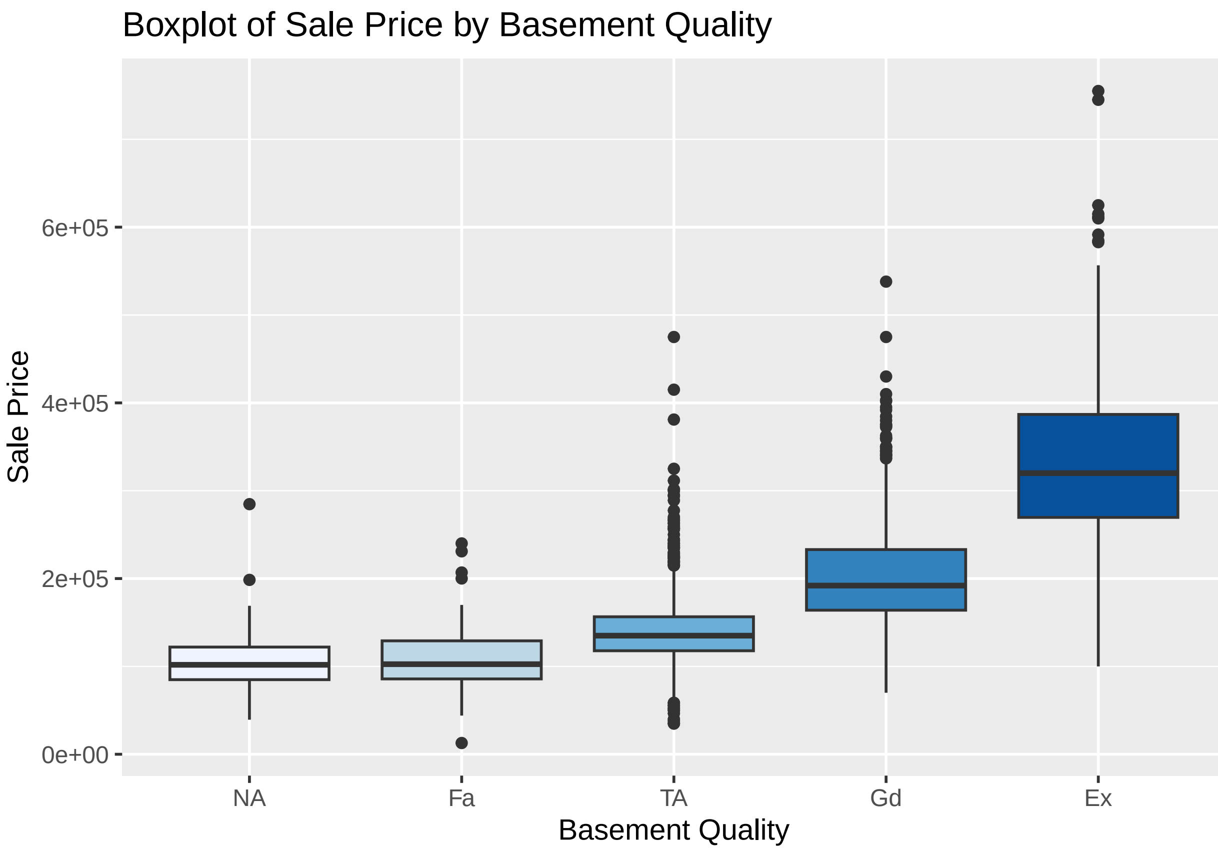
Categorical Variables		Continuous Variables
Ordinal	Nominal	
OverallQual	MSSubClass	LotFrontage
Fence	MSZoning	LotArea
PoolQC	Street	YearBuilt
OverallCond	Alley	YearRemodAdd
ExterQual	LotShape	MasVnrArea
ExterCond	LandContour	BsmtFinSF1
BsmtQual	Utilities	BsmtFinSF2
GarageCond	LotConfig	BsmtUnfSF
BsmtCond	LandSlope	TotalBsmtSF
BsmtExposure	Neighborhood	1stFlrSF
BsmtFinType1	Condition1	2ndFlrSF
BsmtFinType2	Condition2	LowQualFinSF
HeatingQC	BldgType	GrLivArea
CentralAir	HouseStyle	TotRmsAbvGrd
GarageQual	RoofStyle	GarageYrBlt
	RoofMatl	GarageArea
	Exterior1st	WoodDeckSF
	Exterior2nd	OpenPorchSF
	MasVnrType	EnclosedPorch
	Foundation	3SsnPorch
	Heating	ScreenPorch
	Electrical	PoolArea
	BsmtFullBath	MiscVal
	BsmtHalfBath	MoSold
	HalfBath	YrSold
	FullBath	SalePrice
	BedroomAbvGr	
	KitchenAbvGr	
	KitchenQual	
	Functional	
	Fireplaces	
	FireplaceQu	
	GarageType	
	GarageFinish	
	GarageCars	
	PavedDrive	
	MiscFeature	
	SaleType	
	SaleCondition	
	Id	

The table above lists all variables in the Ames, Iowa real estate data set. The data set contains continuous variables and two distinct types of categorical variables (nominal and ordinal). In total, there were 26 continuous variables, 15 ordinal categorical variables, and 40 nominal categorical variables.

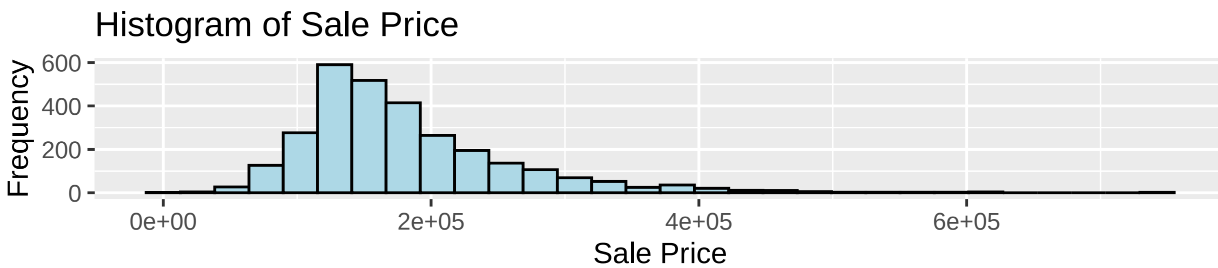
In these boxplots and all subsequent boxplots, the lowest whisker represents the first quartile minus 1.5 multiplied by the interquartile range, the bottom of the box represents the first quartile (25th percentile), the bold horizontal black line on each box represents the median, the top of the box represents the third quartile (75th percentile) and the highest whisker represents the third quartile plus 1.5 multiplied by the interquartile range. Outliers are indicated by black points.

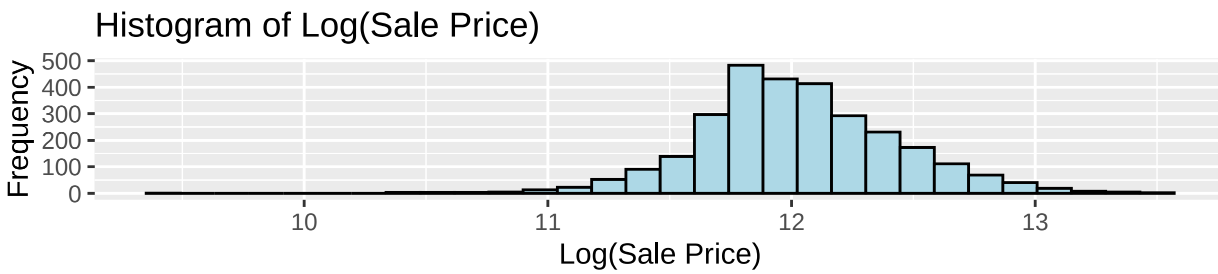


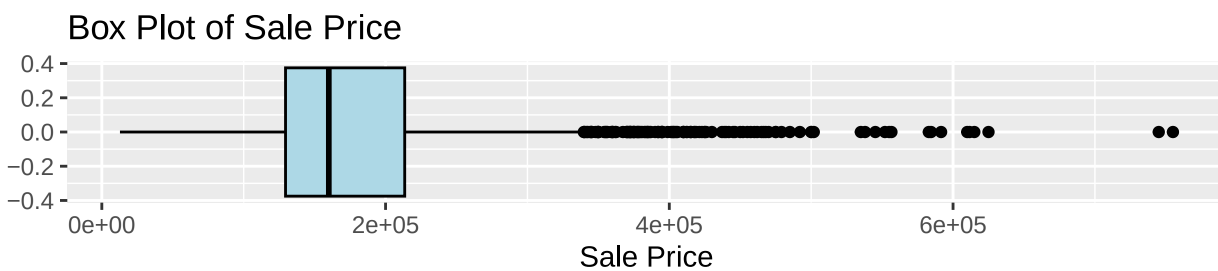
This first set of boxplots displays how the sale price of homes varies as overall home quality increases. Color corresponds to the overall quality, with darker colors correspond to higher overall quality. In terms of `SalePrice`, there appears to be an increasing pattern between `OverallQual` and `SalePrice.`



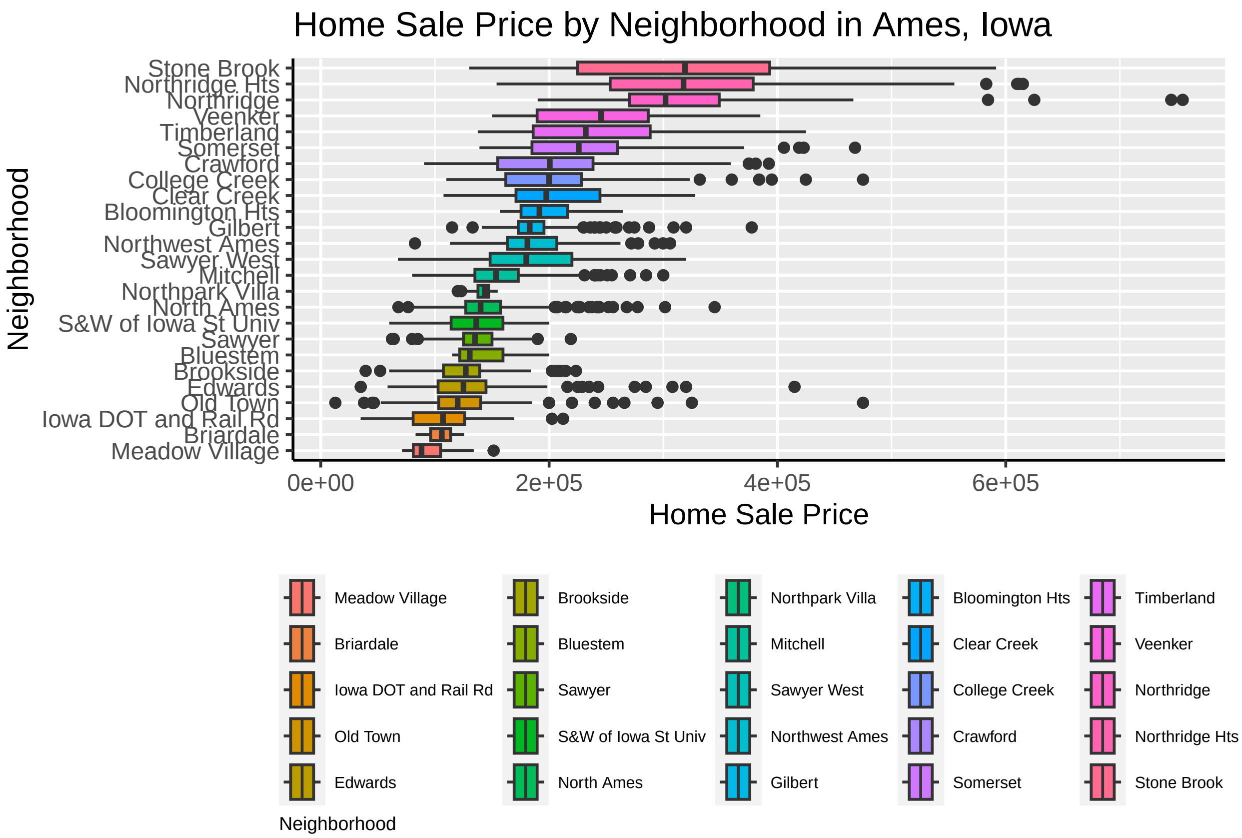
This next set of boxplots displays how the sale price of homes varies as basement quality (height of the basement) increases. Color corresponds to the basement quality, with darker colors correspond to higher basement quality. Here, “Ex” refers to Excellent (100+ inches), “Gd” refers to Good (90-99 inches), “TA” refers to Typical (80-89 inches), “Fa” refers to Fair (70-79 inches), and “NA” refers to No Basement. Based on the right plot, as basement quality (height) increases, the sale price of homes also tends to increase.





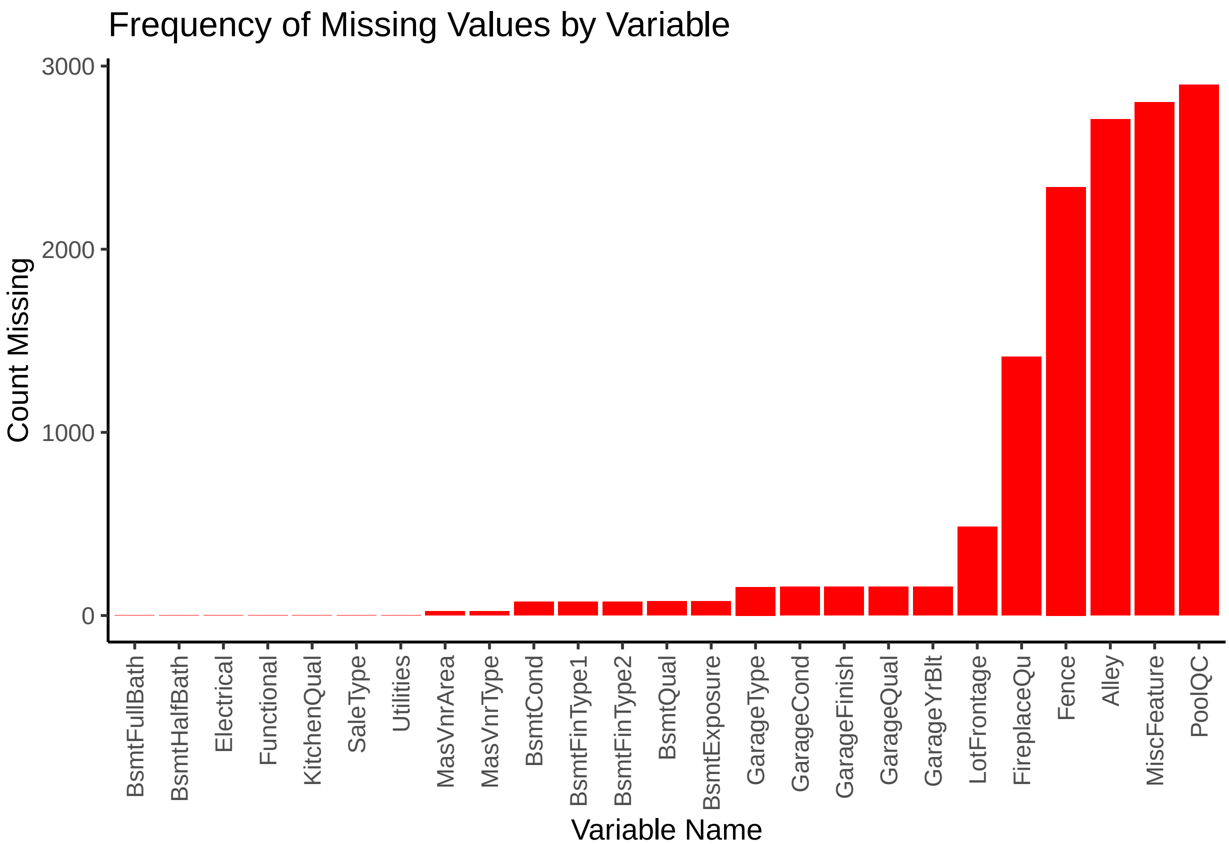


As shown in the boxplot above, sale price had a minimum value of 34900, a maximum value of 755000, a mean of 180921, and a median of 163000. There are also 61 outliers displays on the boxplot, shown as black points. Both the first histogram and boxplot visually show a right-skewed distribution for home sale price. Home sale price seems left-skewed, however, when visualized on the natural logarithmic scale, perhaps due to some outlier(s). It appears that there are two homes with a sale price of over $700,000. If we ignore these outliers, the distribution appears approximately normally distributed.



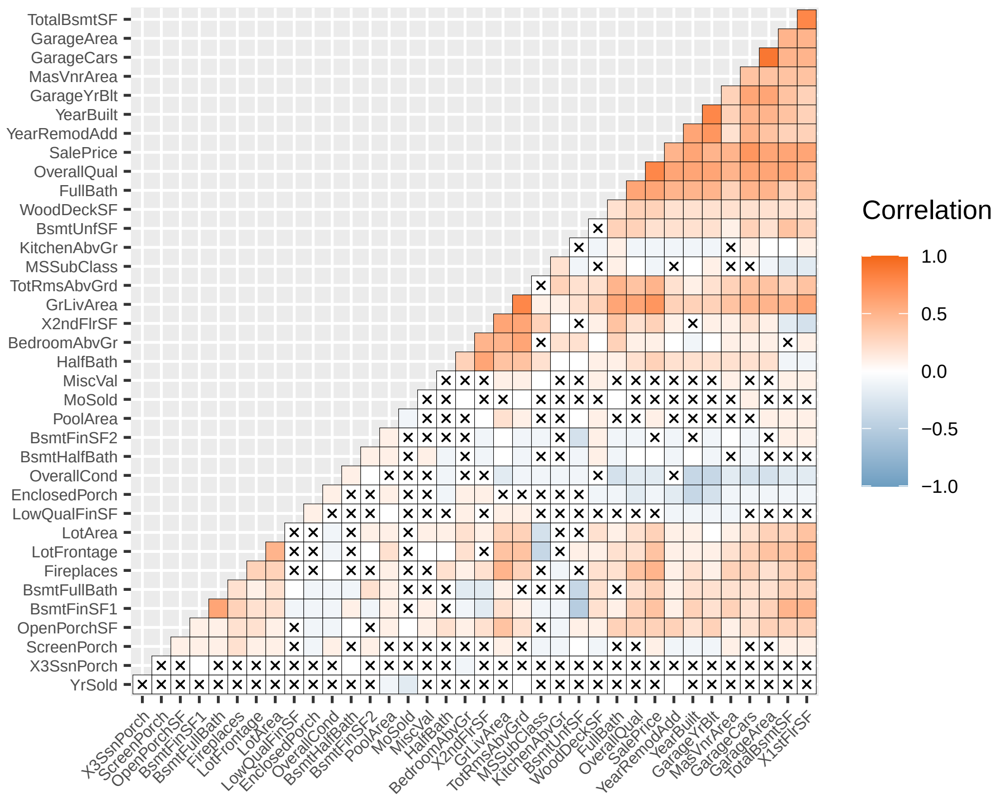
We also visualized the home sale price by neighborhood using boxplots. Here, the x-axis corresponds to the sale price of the home and the y-axis corresponds to the Ames, Iowa neighborhood. The color of each box also corresponds to the home's neighborhood.

We can see that the price of a home varies widely by neighborhood, with `StoneBr` having the highest median sale price overall and `MeadowV` having the lowest median sale price. There are also a few outliers for some of the neighborhoods.



A few variables in the real estate data set contained missing values. In order to visualize the degree of missingness in each variable, the barplot above was created. Each bar corresponds to an individual variable that contains missing values, and the height of the bar corresponds to the number of missing values in that variable. Based on the barplot, there are 5 variables (`FireplaceQu`, `Fence`, `Alley`, `MiscFeature`, and `PoolQC`) with over 1000 (70%) missing values. In total, there are 25 variables with missing values. When possible, missing values were re-coded with values based on the description of the data provided. A detailed description of procedures used to address missing data is provided in the attached "imputation.xlsx" file, and code used to deal with missingness is included in the appendix at the end of this report.

Lower Correlation Matrix for Ames Iowa Dataset



In order to visualize the correlations between variables, the above Pearson correlation matrix was created. Statistically insignificant correlations at the $\alpha = 0.05$ level are marked with a bold "X". It appears that there are many strong positive correlations between variables, which would likely violate assumptions of many statistical models, such as linear regression models. As a result, statistical learning techniques that can handle multicollinearity might be more appropriate for this project.

Variable 1	Variable 2	Pearson Correlation
SalePrice	OverallQual	0.8017662
SalePrice	GrLivArea	0.7085881
SalePrice	GarageCars	0.6515848
SalePrice	GarageArea	0.6433487
SalePrice	TotalBsmtSF	0.6316651
SalePrice	X1stFlrSF	0.6216871

Because the correlation matrix plot is a bit unwieldy and hard to interpret visually, we calculated the variables that had correlation values of at least 0.6 with the response variable, `SalePrice`. As shown in the table above, a total of 6 variables satisfied this criteria. We might (tentatively) expect that these varables would be important for prediction.

Part II: Model Analysis

Analysis: LASSO Model

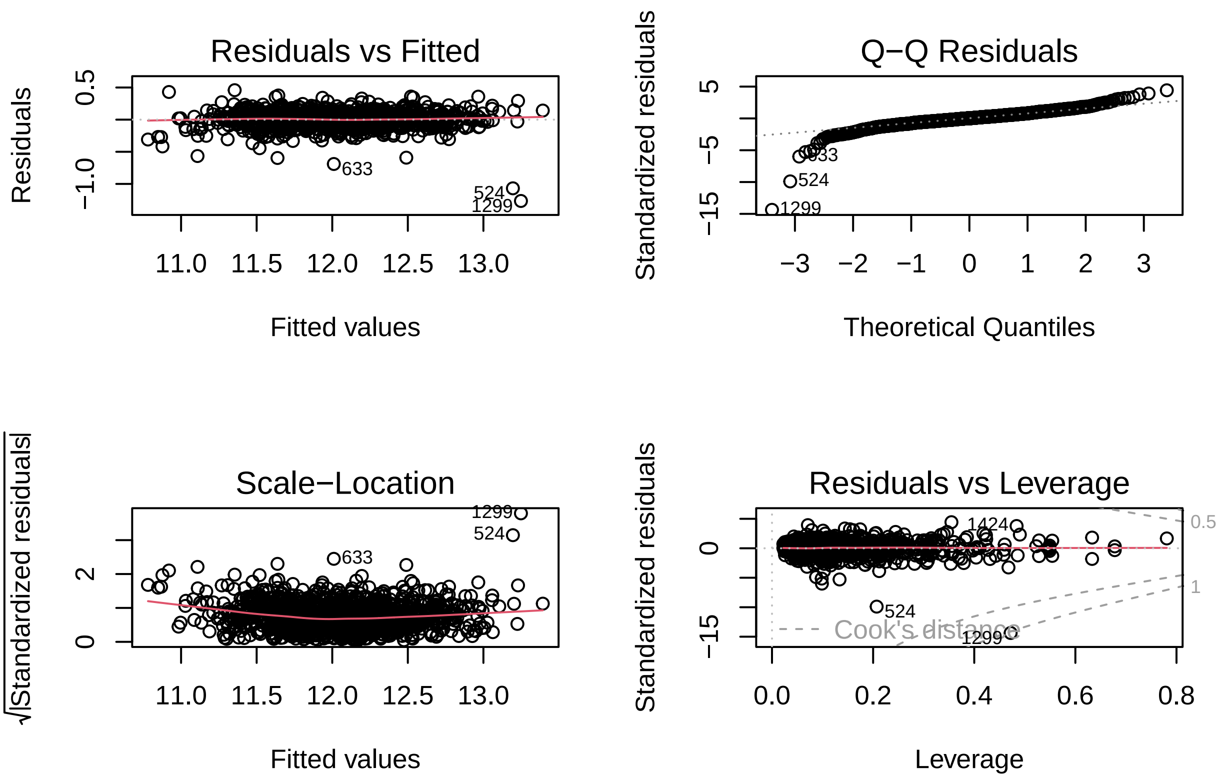
Mathematical Description

The Least Absolute Selection and Shrinkage Operator (LASSO) is a penalized regression model that was developed by Robert Tibshirani in the early 1990's. The LASSO solves the following optimization problem:

$$\min_{\beta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Assumptions

The LASSO solution depends on the scale of the predictors. To avoid this problem, it is best to perform LASSO regression after standardizing the predictors. The `glmnet()` function does this for us when we fit our model.

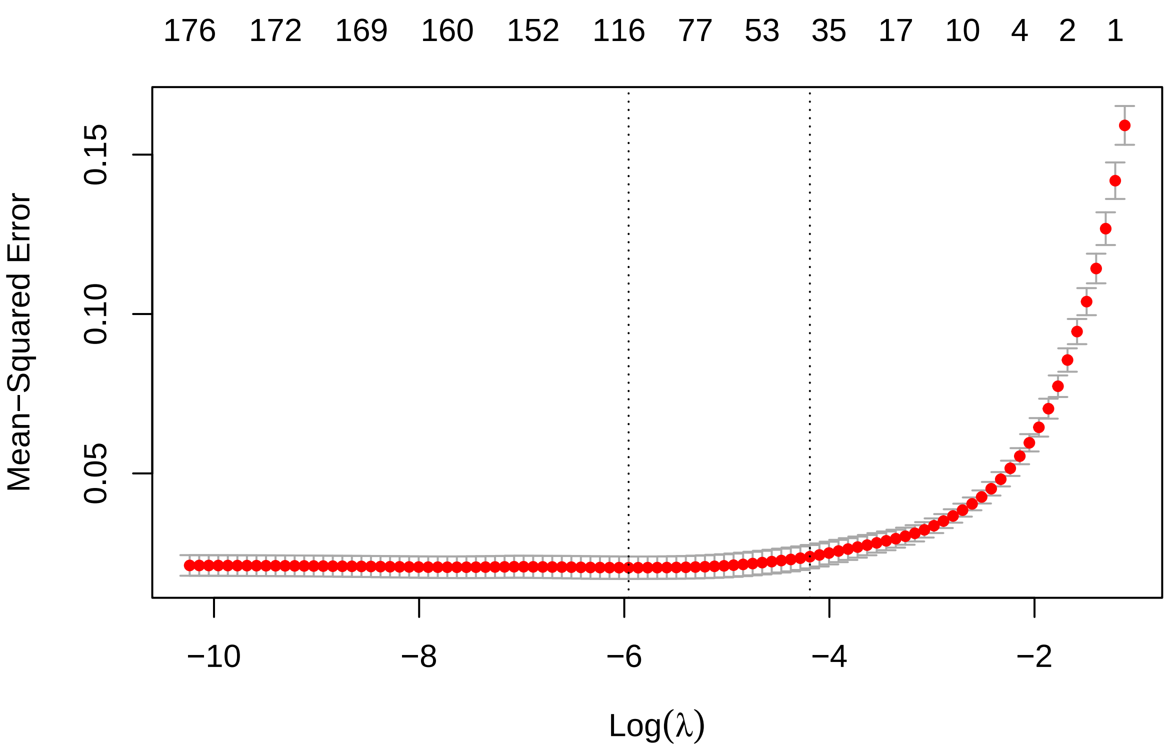


The residuals vs. fitted plot shows a random scatter of points that are centered around the horizontal line at zero without forming a pattern; thus, there is no indication that the assumptions of homoscedasticity and linearity are violated. Furthermore, while the Q-Q Plot of Residuals has deviation at the tails, overall the data appear to be normally distributed. While there are some large values towards the right of the scale-location plot, overall the red line appears to be horizontal across the plot with no clear pattern among the residuals; thus, there is no indication that the homoscedasticity assumption is violated. Finally, the residuals vs. leverage plot shows all points lying within Cook's distance. Thus, there do not appear to be any influential points.

Motivation and Model Building

Unlike Ridge Regression, due to the unique geometry of the LASSO solution for the β coefficients, many of these coefficients can be shrunk to exactly 0 (as opposed to simply very small). Thus, LASSO could be more useful for dimensionality reduction as this data set has 81 variables. For this reason, we chose to fit a LASSO regression model to this data set.

Our final LASSO model included 67 predictor variables. We transformed our response variable, `SalePrice`, using `log()` because doing so made the distribution look closer to a normal, which is better from a model assumption perspective. A total of 13 variables were excluded from the model. The variables `PoolQC`, `MiscFeature`, `Alley`, and `Fence` were not included in the model because >70% of their values were missing. The variables `Condition2`, `RoofMatl`, `Exterior1st`, `Exterior2nd`, `Heating`, `HouseStyle`, `Utilities`, `Electrical`, and `GarageQual` were removed because all unique values of each categorical variables were not present in both the test and training data sets, which prevented validation of the model on the test data set and calculation of mean squared error.



The appropriate value for λ was chosen using cross validation, as shown in the plot above. The cross-validated λ value used in the model was 0.

Results

The test data set was used to calculate the mean squared error for the LASSO model, which was 0.01589. The top 15 largest LASSO coefficients are included in the following table:

Variable	Coefficient Estimate	High Correlation (>= 0.6) with Sale Price
(Intercept)	9.3440200	No
FunctionalSev	-0.2177859	No
FunctionalMaj2	-0.1716217	No
NeighborhoodStoneBr	0.1423768	No
NeighborhoodNridgHt	0.1279725	No
NeighborhoodCrawfor	0.1171519	No
LotShapeIR3	-0.0939067	No
NeighborhoodMeadowV	-0.0876026	No
StreetPave	0.0857463	No
SaleTypeNew	0.0846813	No
BldgTypeTwnhs	-0.0774352	No
NeighborhoodSomerst	0.0751401	No
NeighborhoodNoRidge	0.0724518	No
CentralAirY	0.0697412	No
NeighborhoodIDOTRR	-0.0692966	No

These top 15 variables do not correspond with the variables that are had Pearson correlation values of at least 0.6 with `SalePrice`. In other words, the results from this model do not line up very well with our expectations. This may reflect the fact that LASSO uses a penalized least squares optimization problem, as opposed to correlations with the response variables, to calculate coefficients.

A total of 70 coefficients out of 185 included in the model were shrunk to 0. These terms might not be very important for predicting log sale price of homes. However, due to the large number of remaining coefficient terms in the model, it is difficult to interpret all of these terms. LASSO is more useful for prediction as opposed to inference, so this is not unexpected.

From analyzing the coefficients with the highest values in our model, we noticed that the type of neighborhood had a higher impact on `log(SalePrice)`, with 5 separate categories for the variable Neighborhood within the top 15. These neighborhoods all had positive coefficients, which would indicate having homes in these neighborhoods led to the model predicting a higher log sale price. This intuitively makes sense, as the relevant neighborhoods (Northridge, Stone Brook, Northridge Heights, Crawford, and Somerset) are all within the top seven neighborhoods in terms of median log sales price. We also noted that `StreetPave` was the sixth highest coefficient and also positive, meaning that the model predicted paved streets as having higher log sale prices. With `SaleTypeCon` and `SaleTypeNew` having positive coefficients in the top 15, we also can attribute these sale types as an important part of determining higher log sale price predictions.

`FunctionalSev` had the highest negative coefficients, which indicated that having a severely damaged home led to a much lower prediction of home sale price. This intuitively makes sense since damage will likely reduce the value of a home. Furthermore, having an irregular lot shape (`LotShapeIR3`) and having "typical/average" kitchen quality (`KitchenQualTA`) also led to lower predicted home sale prices; these results are similarly intuitive. However, having good basement quality (`BsmtQualGd`) and kitchen quality (`KitchenQualGd`) led to lower predicted home sale prices. This finding might be less expected as "good" quality would intuitively lead to higher sale prices.

Appendix

Non-Zero LASSO Model Coefficients:

Variable	Coefficient Estimate	High Correlation (>= 0.6) with Sale Price
(Intercept)	9.3440200	No
FunctionalSev	-0.2177859	No
FunctionalMaj2	-0.1716217	No
NeighborhoodStoneBr	0.1423768	No
NeighborhoodNridgHt	0.1279725	No
NeighborhoodCrawfor	0.1171519	No
LotShapeIR3	-0.0939067	No
NeighborhoodMeadowV	-0.0876026	No
StreetPave	0.0857463	No
SaleTypeNew	0.0846813	No
BldgTypeTwnhs	-0.0774352	No
NeighborhoodSomerst	0.0751401	No
NeighborhoodNoRidge	0.0724518	No
CentralAirY	0.0697412	No
NeighborhoodIDOTRR	-0.0692966	No
OverallQual	0.0633041	Yes
NeighborhoodEdwards	-0.0614314	No
NeighborhoodClearCr	0.0588283	No
GarageCars	0.0579939	Yes
NeighborhoodVeenker	0.0573205	No
BsmtExposureGd	0.0509437	No
SaleTypeConLD	0.0495616	No
SaleConditionNormal	0.0493525	No
Condition1Norm	0.0464704	No
BsmtFullBath	0.0454265	No
SaleTypeCon	0.0429970	No
FunctionalTyp	0.0410672	No
FoundationWood	-0.0380826	No
BsmtFinType1Unf	-0.0377959	No
BsmtQualNo Basement	-0.0370798	No
MSZoningRL	0.0360344	No
Condition1RRAe	-0.0354219	No
OverallCond	0.0344084	No
NeighborhoodOldTown	-0.0343724	No
SaleTypeCWD	0.0338255	No
GarageCondFa	-0.0334072	No
LotConfigCulDSac	0.0327937	No
ExterCondFa	-0.0322217	No
FullBath	0.0303084	No
SaleConditionAdjLand	0.0294309	No
FoundationPConc	0.0261238	No
BldgType2fmCon	0.0259360	No
FireplaceQuNo Fireplace	-0.0258414	No
NeighborhoodBrDale	-0.0258134	No
KitchenAbvGr	-0.0254866	No
HeatingQCTA	-0.0237632	No

BldgTypeTwnhsE	-0.0229719	No
BsmtFinType2No Basement	-0.0229096	No
MSZoningFV	0.0219266	No
BsmtFinType2BLQ	-0.0213147	No
LandContourLvl	0.0202712	No
HalfBath	0.0198097	No
BsmtExposureNo Basement	-0.0187469	No
HeatingQCFa	-0.0187084	No
LotShapeIR2	0.0183109	No
NeighborhoodTimber	0.0173154	No
ExterQualFa	-0.0157880	No
FunctionalMod	-0.0143276	No
Condition1RRNn	0.0136781	No
LandContourHLS	0.0136440	No
PavedDriveY	0.0128526	No
LotConfigFR3	-0.0125982	No
FoundationStone	0.0125153	No
Condition1Feedr	-0.0121756	No
LotConfigFR2	-0.0115782	No
HeatingQCGd	-0.0115507	No
LandSlopeMod	0.0107995	No
TotRmsAbvGrd	0.0104412	No
MSZoningRH	0.0098773	No
BsmtExposureNo	-0.0097887	No
Fireplaces	0.0096616	No
Condition1RRAn	0.0093338	No
ExterQualTA	-0.0093190	No
GarageTypeAttchd	0.0086619	No
SaleTypeOth	0.0083480	No
GarageTypeCarPort	-0.0083366	No
BsmtFinType1GLQ	0.0073822	No
KitchenQualTA	-0.0072430	No
ExterCondPo	-0.0070826	No
ExterCondTA	0.0066798	No
RoofStyleGable	-0.0065817	No
BsmtFinType2GLQ	0.0065495	No
BsmtQualTA	-0.0059213	No
GarageFinishNo Garage	-0.0056452	No
FireplaceQuGd	0.0050647	No
GarageFinishUnf	-0.0046339	No
BsmtCondTA	0.0043345	No
GarageCondNo Garage	-0.0043324	No
NeighborhoodMitchel	-0.0040257	No
KitchenQualGd	-0.0018404	No
LotConfigInside	-0.0018330	No
BedroomAbvGr	0.0014660	No
YrSold	-0.0013026	No
NeighborhoodBrkSide	0.0012505	No
YearBuilt	0.0012169	No
KitchenQualFa	-0.0011137	No
BsmtCondNo Basement	-0.0009566	No

FoundationSlab	-0.0007534	No
YearRemodAdd	0.0007141	No
MSSubClass	-0.0003650	No
ScreenPorch	0.0002615	No
BsmtFinType1No Basement	-0.0001891	No
GrLivArea	0.0001809	Yes
X3SsnPorch	0.0000948	No
WoodDeckSF	0.0000894	No
PoolArea	-0.0000726	No
EnclosedPorch	0.0000546	No
X1stFlrSF	0.0000451	Yes
LotFrontage	-0.0000378	No
TotalBsmtSF	0.0000189	Yes
GarageArea	0.0000153	Yes
GarageTypeNo Garage	-0.0000075	No
OpenPorchSF	0.0000050	No
Id	-0.0000040	No
LotArea	0.0000015	No

Zero LASSO Model Coefficients:

Variable	Coefficient Estimate	High (>= 0.6) Correlation with Sale Price
MSZoningRM	0	No
LotShapeReg	0	No
LandContourLow	0	No
LandSlopeSev	0	No
NeighborhoodBlueste	0	No
NeighborhoodCollgCr	0	No
NeighborhoodGilbert	0	No
NeighborhoodNAmes	0	No
NeighborhoodNPkVill	0	No
NeighborhoodNWAmes	0	No
NeighborhoodSawyer	0	No
NeighborhoodSawyerW	0	No
NeighborhoodSWISU	0	No
Condition1PosA	0	No
Condition1PosN	0	No
Condition1RRNe	0	No
BldgTypeDuplex	0	No
RoofStyleGambrel	0	No
RoofStyleHip	0	No
RoofStyleMansard	0	No
RoofStyleShed	0	No
MasVnrTypeBrkFace	0	No
MasVnrTypeNone	0	No
MasVnrTypeStone	0	No
MasVnrTypeUnknown	0	No
MasVnrArea	0	No
ExterQualGd	0	No
ExterCondGd	0	No
FoundationCBlock	0	No
BsmtQualFa	0	No
BsmtQualGd	0	No
BsmtCondGd	0	No
BsmtCondPo	0	No
BsmtExposureMn	0	No
BsmtExposureUnknown	0	No
BsmtFinType1BLQ	0	No
BsmtFinType1LwQ	0	No
BsmtFinType1Rec	0	No
BsmtFinSF1	0	No
BsmtFinType2LwQ	0	No
BsmtFinType2Rec	0	No
BsmtFinType2Unf	0	No
BsmtFinSF2	0	No
BsmtUnfSF	0	No
HeatingQCPo	0	No
X2ndFlrSF	0	No
LowQualFinSF	0	No
BsmtHalfBath	0	No
FunctionalMin1	0	No

FunctionalMin2	0	No
FireplaceQuFa	0	No
FireplaceQuPo	0	No
FireplaceQuTA	0	No
GarageTypeBasment	0	No
GarageTypeBuiltIn	0	No
GarageTypeDetchd	0	No
GarageYrBlt	0	No
GarageFinishRFn	0	No
GarageCondGd	0	No
GarageCondPo	0	No
GarageCondTA	0	No
PavedDriveP	0	No
MiscVal	0	No
MoSold	0	No
SaleTypeConLI	0	No
SaleTypeConLw	0	No
SaleTypeWD	0	No
SaleConditionAlloca	0	No
SaleConditionFamily	0	No
SaleConditionPartial	0	No

Code to Produce Test and Train Data Sets Used in Model

```r
library(tidyverse)
library(corrr)
library(igraph)
library(ggraph)
library(GGally)
library(ggcorrplot)
library(ggstatsplot)

train <- read_csv("data/train.csv")
test <- read_csv("data/test_new.csv")

# explore which names in "train" differ from "test"
names(train) == names(test)

# view "train" names that differ from "test' and vice versa
names(train)[(names(train) == names(test)) == FALSE]
names(test)[(names(train) == names(test)) == FALSE]
# naming variables with names that start with integers might cause problems,
# so let's set the names in "train" to the "test" names (that don't start with
# an integer)

# set names in "train" that differ to the names corresponding to "test"
names(train)[(names(train) == names(test)) == FALSE] <-
  names(test)[(names(train) == names(test)) == FALSE]

# combine the "test" and "train" datasets for exploratory data analysis
ames <- rbind(train, test)

################################################################################
# `Condition2`, `RoofMatl`, `Exterior1st`, `Exterior2nd`, `Heating`, `HouseStyle`
################################################################################
ames %>%
  group_by(Condition2) %>%
  dplyr::summarize(count = n())

ames %>%
  group_by(RoofMatl) %>%
  dplyr::summarize(count = n())

ames %>%
  group_by(Exterior1st) %>%
  dplyr::summarize(count = n())

ames %>%
  group_by(Exterior2nd) %>%
  dplyr::summarize(count = n())

ames %>%
  group_by(Heating) %>%
  dplyr::summarize(count = n())
```

```r
ames %>%
  group_by(HouseStyle) %>%
  dplyr::summarize(count = n())

ames <- ames %>%
  dplyr::select(-c(Condition2, RoofMatl, Exterior1st, Exterior2nd, Heating, HouseStyle))
################################################################################
# `PoolQC`, `MiscFeature`, `Alley`, `Fence`
################################################################################
# remove variables because >70% missing
ames <- ames %>%
  dplyr::select(-c(PoolQC, MiscFeature, Alley, Fence))

################################################################################
# `FireplaceQu`
################################################################################
# check to see if there are any missing values for `FireplaceQu` when a house
# has fireplace(s)
ames[is.na(ames$FireplaceQu) == TRUE & ames$Fireplaces > 0,] # there are none

# now coerce missing values in `FireplaceQu` to "No Fireplace"
ames <- ames %>%
  mutate(FireplaceQu = case_when(
    is.na(FireplaceQu) == TRUE ~ "No Fireplace",
    TRUE ~ FireplaceQu
  ))
################################################################################
# `LotFrontage`
################################################################################
# change missing `LotFrontage` values to 0
ames <- ames %>%
  mutate(LotFrontage = case_when(
    is.na(LotFrontage) == TRUE ~ 0,
    TRUE ~ LotFrontage
  ))

################################################################################
# `GarageYrBlt`
################################################################################
# change value in `GarageYrBlt` from 2207 (impossible) to 2007
# (the house was built in 2006 and sold in 2007 so this is a logical imputation)
ames$GarageYrBlt[ames$GarageYrBlt == 2207 & is.na(ames$GarageYrBlt) == FALSE] <- 2007

# change missing values to -1
ames <- ames %>%
  mutate(GarageYrBlt = case_when(is.na(GarageYrBlt) == TRUE ~ -1,
                                 TRUE ~ GarageYrBlt))

################################################################################
# `GarageFinish`
################################################################################
ames %>%
  group_by(GarageFinish) %>%
```

```
  dplyr::summarize(count = n())

# For missing values, set garage area to "No Garage"

###IGNORE For missing values where garage area is 0, set `GarageFinish` to "No Garage"
###IGNORE For missing values where garage area is >0 (i.e. there is a garage but we
###IGNORE don't know the finish), set to "TA" (Typical/Average)

ames <- ames %>%
  mutate(GarageFinish = case_when(
    (is.na(GarageFinish) == TRUE) & ((GarageArea > 0) == TRUE) ~ "No Garage", # TA
    (is.na(GarageFinish) == TRUE) & ((GarageArea > 0) == FALSE) ~ "No Garage",
    TRUE ~ GarageFinish))

################################################################################
# `GarageQual`
################################################################################
ames %>%
  group_by(GarageQual) %>%
  dplyr::summarize(count = n())

# For missing values where garage area is 0, set `GarageQual` to "No Garage"
# For missing values where garage area is >0 (i.e. there is a garage but we
# don't know the quality), set to "TA" (Typical/Average)
#ames <- ames %>%
#  mutate(GarageQual = case_when(
#    (is.na(GarageQual) == TRUE) & ((GarageArea > 0) == TRUE) ~ "TA",
#    (is.na(GarageQual) == TRUE) & ((GarageArea > 0) == FALSE) ~ "No Garage",
#    TRUE ~ GarageQual))

# remove variable
ames <- ames %>%
  dplyr::select(-c(GarageQual))
################################################################################
# `GarageCond`
################################################################################
ames %>%
  group_by(GarageCond) %>%
  dplyr::summarize(count = n())

# For missing values where garage area is 0, set `GarageCond` to "No Garage"
# For missing values where garage area is >0 (i.e. there is a garage but we
# don't know the condition), set to "TA" (Typical/Average)
ames <- ames %>%
  mutate(GarageCond = case_when(
    (is.na(GarageCond) == TRUE) & ((GarageArea > 0) == TRUE) ~ "TA",
    (is.na(GarageCond) == TRUE) & ((GarageArea > 0) == FALSE) ~ "No Garage",
    TRUE ~ GarageCond))

################################################################################
# `GarageType`
################################################################################
# For missing values where garage area is 0, set `GarageType` to "No Garage"
```

```r
# For missing values where garage area is >0 (i.e. there is a garage but we
# don't know the type), set to "Unknown"
ames <- ames %>%
  mutate(GarageType = case_when(
    (is.na(GarageType) == TRUE) & ((GarageArea > 0) == TRUE) ~ "Unknown",
    (is.na(GarageType) == TRUE) & ((GarageArea > 0) == FALSE) ~ "No Garage",
    TRUE ~ GarageType))
################################################################################
# `BsmtExposure`
################################################################################
# For missing values where total basement square feet is 0,
# set `BsmtExposure` to "No Basement"
# For missing values where total basement square feet is >0 (i.e. there is a
# basement but we don't know the exposure), set to "Unknown"
ames <- ames %>%
  mutate(BsmtExposure = case_when(
    (is.na(BsmtExposure) == TRUE) & ((TotalBsmtSF > 0) == TRUE) ~ "Unknown",
    (is.na(BsmtExposure) == TRUE) & ((TotalBsmtSF > 0) == FALSE) ~ "No Basement",
    TRUE ~ BsmtExposure))

################################################################################
# `BsmtQual`
################################################################################
# For missing values, set `BsmtQual` to "No Basement"

#IGNORE For missing values where total basement square feet is 0,
#IGNORE set `BsmtQual` to "No Basement"
#IGNORE For missing values where total basement square feet is >0 (i.e. there is a
#IGNORE basement but we don't know the quality), set to "No Basement"
ames <- ames %>%
  mutate(BsmtQual = case_when(
    (is.na(BsmtQual) == TRUE) & ((TotalBsmtSF > 0) == TRUE) ~ "No Basement",
    (is.na(BsmtQual) == TRUE) & ((TotalBsmtSF > 0) == FALSE) ~ "No Basement",
    TRUE ~ BsmtQual))

################################################################################
# `BsmtFinType2`
################################################################################
# For missing values where total basement square feet is 0,
# set `BsmtFinType2` to "No Basement"
# For missing values where type 2 finished basement square feet is >0 (i.e. there is a
# basement but we don't know the rating of type 2), set to "Unknown"
ames <- ames %>%
  mutate(BsmtFinType2 = case_when(
    (is.na(BsmtFinType2) == TRUE) & ((BsmtFinSF2 > 0) == TRUE) ~ "No Basement",
    (is.na(BsmtFinType2) == TRUE) & ((BsmtFinSF2 > 0) == FALSE) ~ "No Basement",
    TRUE ~ BsmtFinType2))

################################################################################
# `BsmtCond`
################################################################################
# For missing values where total basement square feet is 0,
# set `BsmtCond` to "No Basement"
```

```
# For missing values where total basement square feet is >0 (i.e. there is a
# basement but we don't know the condition), set to "Unknown"
ames <- ames %>%
  mutate(BsmtCond = case_when(
    (is.na(BsmtCond) == TRUE) & ((TotalBsmtSF > 0) == TRUE) ~ "Unknown",
    (is.na(BsmtCond) == TRUE) & ((TotalBsmtSF > 0) == FALSE) ~ "No Basement",
    TRUE ~ BsmtCond))

################################################################################
# `BsmtFinType1`
################################################################################
# For missing values where total basement square feet is 0,
# set `BsmtFinType1` to "No Basement"
# For missing values where type 1 finished basement square feet is >0 (i.e. there is a
# basement but we don't know the rating of type 1), set to "Unknown"
ames <- ames %>%
  mutate(BsmtFinType1 = case_when(
    (is.na(BsmtFinType1) == TRUE) & ((BsmtFinSF1 > 0) == TRUE) ~ "Unknown",
    (is.na(BsmtFinType1) == TRUE) & ((BsmtFinSF1 > 0) == FALSE) ~ "No Basement",
    TRUE ~ BsmtFinType1))

################################################################################
# `MasVnrType`
################################################################################
# set missing values to "Unknown"
ames <- ames %>%
  mutate(MasVnrType = case_when(
    (is.na(MasVnrType) == TRUE) ~ "Unknown",
    TRUE ~ MasVnrType
  ))

################################################################################
# `MasVnrArea`
################################################################################
# set missing values to 0
ames <- ames %>%
  mutate(MasVnrArea = case_when(
    (is.na(MasVnrArea) == TRUE) ~ 0,
    TRUE ~ MasVnrArea
  ))

################################################################################
# `Utilities`
################################################################################
# set missing values to "Unknown"
ames %>%
  group_by(Utilities) %>%
  dplyr::summarize(count = n())

ames <- ames %>%
  dplyr::select(-c(Utilities))
################################################################################
# `Electrical`
```

```r
################################################################################
# set missing values to "Unknown"
ames %>%
  group_by(Electrical) %>%
  dplyr::summarize(count = n())

# remove because not enough instances of unique values
ames <- ames %>%
  dplyr::select(-c(Electrical))
################################################################################
# `BsmtFullBath`
################################################################################
# set missing values to 0
ames <- ames %>%
  mutate(BsmtFullBath = case_when(
    (is.na(BsmtFullBath) == TRUE) ~ 0,
    TRUE ~ BsmtFullBath
  ))

################################################################################
# `BsmtHalfBath`
################################################################################
# set missing values to 0
ames <- ames %>%
  mutate(BsmtHalfBath = case_when(
    (is.na(BsmtHalfBath) == TRUE) ~ 0,
    TRUE ~ BsmtHalfBath
  ))

################################################################################
# `KitchenQual`
################################################################################
# set missing values to "TA"
ames <- ames %>%
  mutate(KitchenQual = case_when(
    (is.na(KitchenQual) == TRUE) ~ "TA",
    TRUE ~ KitchenQual
  ))

################################################################################
# `Functional`
################################################################################
ames %>%
  group_by(Functional) %>%
  dplyr::summarize(count = n())

# set missing values to "Typ"
ames <- ames %>%
  mutate(Functional = case_when(
    (is.na(Functional) == TRUE) ~ "Typ",
    TRUE ~ Functional
  ))
```

```r
################################################################################
# `SaleType`
################################################################################
# set missing values to "Oth"
ames <- ames %>%
  mutate(SaleType = case_when(
    (is.na(SaleType) == TRUE) ~ "Oth",
    TRUE ~ SaleType
  ))

################################################################################
################################################################################
################################################################################

# output final file with no missing values
write_csv(ames,
          file = paste0(getwd(), "/missing data/output/ames2.csv"))

# subset final file with no missing values into test and training datasets
ames_test <- ames %>%
  filter((ames$Id %in% test$Id) == TRUE)

ames_train <- ames %>%
  filter((ames$Id %in% train$Id) == TRUE)

# output final test and training datasets
write_csv(ames_test,
          file = paste0(getwd(), "/missing data/output/ames_test2.csv"))

write_csv(ames_train,
          file = paste0(getwd(), "/missing data/output/ames_train2.csv"))
```