Predicting House Prices in Ames, Iowa Using Multiple Linear Regression

Authors:

Joel Laskow

Chris Johnson

Introduction:

Any homebuyer in the 21st century will know the struggle of finding affordable housing. Over the past 20 years, popular discourse has grown increasingly louder regarding the state of the housing market. What was once a market of modestly priced dwellings has propagated into an ocean of listings many consider wildly unaffordable.

The scenario has many consumers wondering if house ownership is worth the effort, or if buying a home today is even a safe investment. With the daunting prices of today's home deterring demand, there is a possibility of a collapse in the housing market; it's imperative, then, that housing agencies work to identify the true value of a house in today's market. If this market is to stabilize, consumers and agencies alike must be able to agree on a reasonable price for a new home.

We will attempt to tackle this issue by identifying key variables related to house prices in Ames, Iowa. From these findings we will make predictions on housing prices.

This report will serve as a guide to walk readers through our findings and conclusions.

Data Description:

All training and test datasets (Train and Test, respectively) were obtained through *Kaggle.* This data contains 79 explanatory variables for 1460 housing observation in the training set and 1459 in the test set. . These variables include numeric and categorical descriptors of a litany of features found in different houses. Full datasets with all 79 variables can be obtained from the links provided in the Appendix, along with key code and graphs from our study.

Data Cleaning:

Numeric

The Housing dataset obtained from Kaggle was assessed for missing variables. Respectively, 0.6% of raw numeric variables in the training set were missing; 0.6% of raw numeric variables were missing in the test set. This accounted for 18% of LotsFrontage values in the training set and 16% in the test dataset, 6% of GarageYrBlt values in the training set and 5% of GarageYrBlt values in the test set. 1% of MasVnrArea values were missing from the test set. Missing values were imputed with the mean of the respective explanatory variable**.** In the train set, 94% of Alley, 47% of FireplaceQu, 100% PoolQC, 81% Fence, and 95% of MiscFeature variables were missing; respectively, 93%, 50%, 100%, 80%, and 95% were missing from the test set. Given the proportion of missing values, these variables were not included in later tests. Remaining variables with missing values were imputed with the respective explanatory variable mode. Full graphs of missing variables before and after cleaning are included in **Figure 6.** Less than 0.1% of the test set was still missing after cleaning; however, due to time constraints and technical difficulties we were unable to identify and fix the remaining missing values. We moved forward under the assumption that these missing datapoints would not influence predictions.

Only one instance of MSSubClass of level “150” appeared in our data (test and train). Given the level’s similarity to MSSubClass level “50”, this datapoint’s MSSubClass factor level was changed to Level “50”. For descriptions of each MSSubClass level, see **Appendix II.**

Lastly, an RShiny application was created to explore distribution of GrLivArea and the distribution of SalePrice, as well as scatterplots exploring the correlation of these variables, in all Neighborhoods within our training set.

Analysis:

Part 1

Our first endeavor sought to quantify the relationship between GrLivArea (Living Area Above Ground), and Sale Price in the 3 neighborhoods: Brookside (BrkSide), Edwards (Edwards), and North Ames (NAmes).

Normality Checks

We checked the distribution of GrLivArea and SalePrice using visual means. Both variables were logged to account for outliers within the dataset. QQ and Histograms plots, as well as scatterplots comparing SalePrice and GrLivArea, before and after log transformation are shown in **Figure 2**. Visual assessment of log-transformed variables provided little evidence against assumptions of normality within the dataset.

Outlier Check

Residual vs Fitted plots, as well as Scale Location plots and Residualvs Leverage plots were used to visually identify potential high-leverage outliers (**Figure 3a).** Potential high-leverage outliers were identified at row 339, 136, 131, 190. 104, and 186, 411 of the dataset. After removal of these points, the dataset appeared free of high-leverage points (**Figure 3b).**

K-Fold Cross Validation was implemented to check the validity of the full model. At k=5, the mean CV Press score for our regression model was 4.847129. Adjusted R Squared with and without outliers were identified as 0.4857 and 0.4899 respectively.

Final Model Assessment

Final assessment of regression model comparing log(GrLivArea) to log(SalePrice) generated with and without high-leverage outliers (**Figure 4)**. With outliers, we identified the coefficients associated with each parameter of our model (**Figure 5).**

The outlier-included data suggest that a doubling of GrLivArea is associated with, at the 95% confidence interval, a multiplicative change in median SalePrice of (3.1076, 3.6247) for Brookside, (1.89092, 2.06258) for Edwards, and (2.11195, 2.31203) for NAmes.

The outlier-omitted data suggest that a doubling of GrLivArea is associated with, at the 95% confidence interval, a multiplicative change in median SalePrice of (3.13508, 3.6724) for Brookside, (1.88357, 2.04337) for Edwards, and (1.88357, 2.27093) for NAmes.

Part 2

Variable Selection:

We constructed correlation matrices to identify potential multicollinearity among the 74 remaining explanatory variables in the cleaned test and train datasets. The logged variables, such as SalePrice, MsVnRArea, TotalBsmtSF, X1stFlrSF, X2ndFlrSF, LotFrontage, LotArea, YearBuilt, YearRemodAdd, GarageArea, WoodDeckSF, and OpenPorchSF, revealed some evidence of multicollinearity: LotArea with LotFrontage, YearBuilt with YearRemodAdd, and TotalBsmntSF with X1stFlrSF. Consequently, we addressed multicollinearity by removing LotFrontage, YearBuilt, TotalBsmntSF, and MasVnrArea. Additionally, WoodDeckSF, OpenPorchSF, MasVnrArea, BsmtFullBath, BsmtHalfBath, and HalfBath were excluded due to a lack of clear correlation with SalePrice." (**Figure 7)**

We moved forward with the following numerical variables: FirePlaces, GarageYrBlt, FullBath, TotRmsAbvGrd, X1stFlrSF, and LotArea, YearRemodAdd. Due to time constraints we focused solely on the categorical variable MSSubClass, which identifies the type of dwelling involved in the sale. This variable was chosen because it handled descriptors and characteristics that were explained by other categorical variables. This left us with 8 variables out of the initial 74.

Full Regression Model:

log(SalePrice) = Fireplaces + GarageYrBlt + FullBath + TotRmsAbvGrd + log(X1stFlrSF) + log(LotArea) + YearRemodAdd + MSSubClass

Forward, Backward, and Stepwise selection was conducted on candidate variables in SAS. Neither forward nor stepwise selection recommended any variables for removal. Backward selection elected for removal of TotRmsAbvGrd, however change in CV Press with removal was negligible compared to those of the stepwise and forward selection models. A summary of our forward, backward, and stepwise selection models are found in **Figure 8 and Figure 9.** Final regression model was as follows:

log(SalePrice) = Fireplaces + GarageYrBlt + FullBath + TotRmsAbvGrd + log(X1stFlrSF) + log(LotArea) + YearRemodAdd + MSSubClass

Assumption Checks

After constructing our final model, we examined residual distributions to assess normality, variance, linearity, and independence assumptions. The Residual Plot displayed random distribution without observable variance changes, while QQPlots and Histograms supported normality assumptions **(Figure 10)**. Overall, assessments from scatterplots, QQ Plots, and Histograms found little evidence against the assumptions of normality, constant variance, or linear trend between log(SalePrice) and our model's variables. Lastly, we assume independence among observations in our dataset.

Detecting High-Leverage Points

After checking assumptions, we checked for high-leverage datapoints within our training datasets. High leverage outliers were identified visually through Cook's D Bar Plot, and a Studentized Residuals vs Leverage Plot (F**igure 11**). Potential high leverage outliers were identified in rows 51, 513, and 859 of the cleaned training dataset.

To test the effects of these outliers on the final model, the training dataset was edited to remove these points. Final Adjusted R-Square changed from 0.7757 (with high-leverage outliers) to 0.7618 (without). Residual Standard Error changed from 0.1926 to 0.193. Given the minimal change in Adjusted R-Square and Residual Standard Error, the outliers were left in the dataset.

Predictions and Model Comparison

House SalePrice values for the test dataset were predicted using the final linear regression model (MLR). Predictions were also made with a simple linear regression model (SLR) using only

YearBuilt, and a second multiple linear regression model (Custom) using GrLivArea and FullBath. Respectively, Adjusted R2 values were 0.7691, 0.3436, and 0.93, while CV PRESS values were 55.51860, 153.34537, and 1149. **(Figure 12).** Final Kaggle scores were, 0.20623 (MLR), 0.32043 (SLR), and 0.32034 (Custom).

Conclusion:

We are confident that our results reliably quantify the relationship between key housing variables and sales price in Ames, Iowa. It's important to iterate that the findings found in this report are limited to the population of Ames, Iowa. Additional information and comprehensive testing is required to extrapolate results to the larger housing population; nevertheless, we consider our findings invaluable for housing agencies and buyers seeking to identify optimal housing prices within Ame Countys. We further believe that similar testing methods might be applied to samples representative of other housing populations. Such testing could yield profound results that allow predictions of the wider housing population.

Appendix I

RShiny Link:

- *https://cjohnson4510.shinyapps.io/Housing_App/*

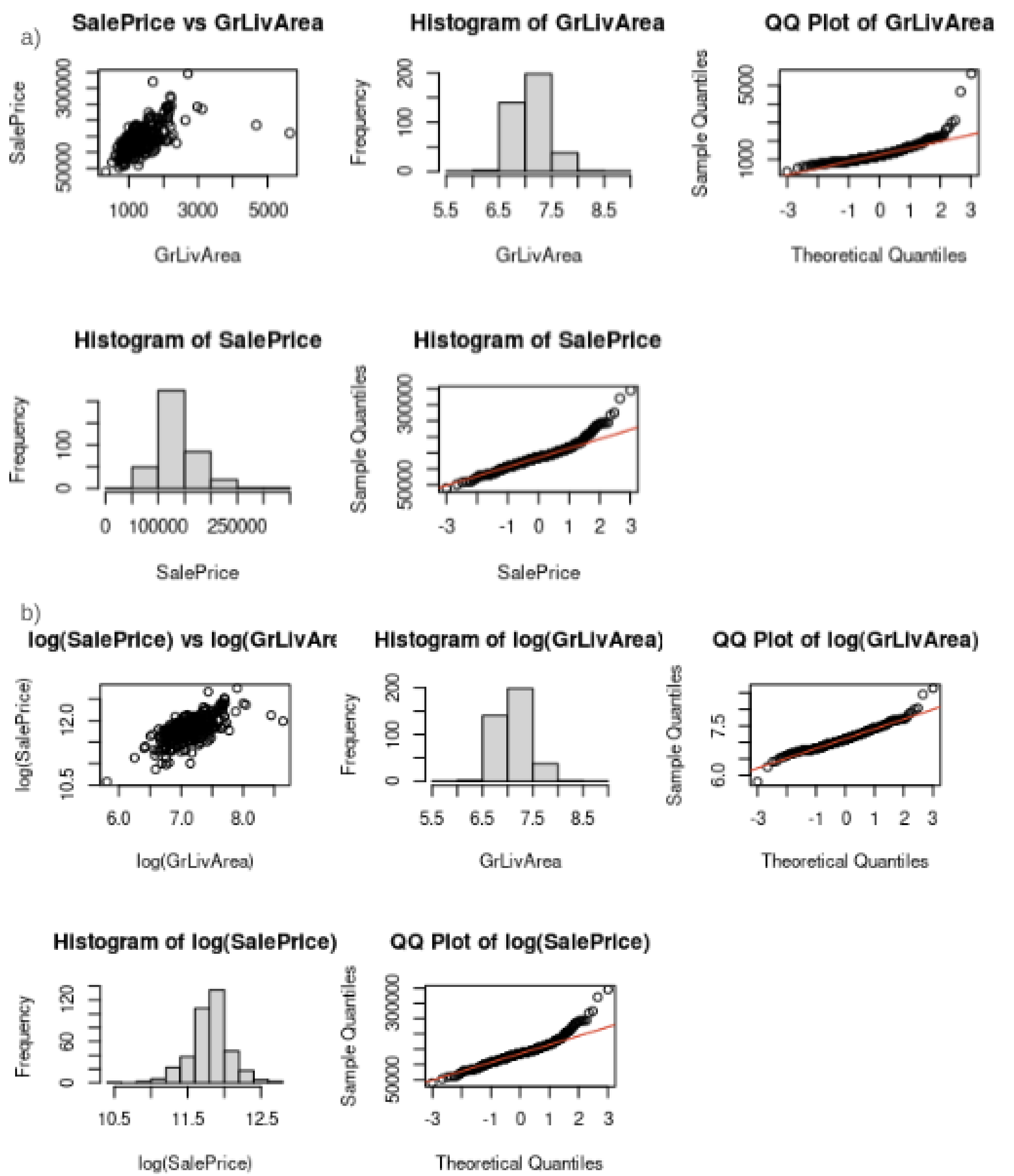


Figure 2. Normality graphs comparing GrLivArea to SalePrice. a) Scatterplot (top left) of SalePrice and GrLivArea, as well as histograms and QQ plots of each variable. b) Scatterplots, QQ Plots, and Histograms of variables after log transformation.

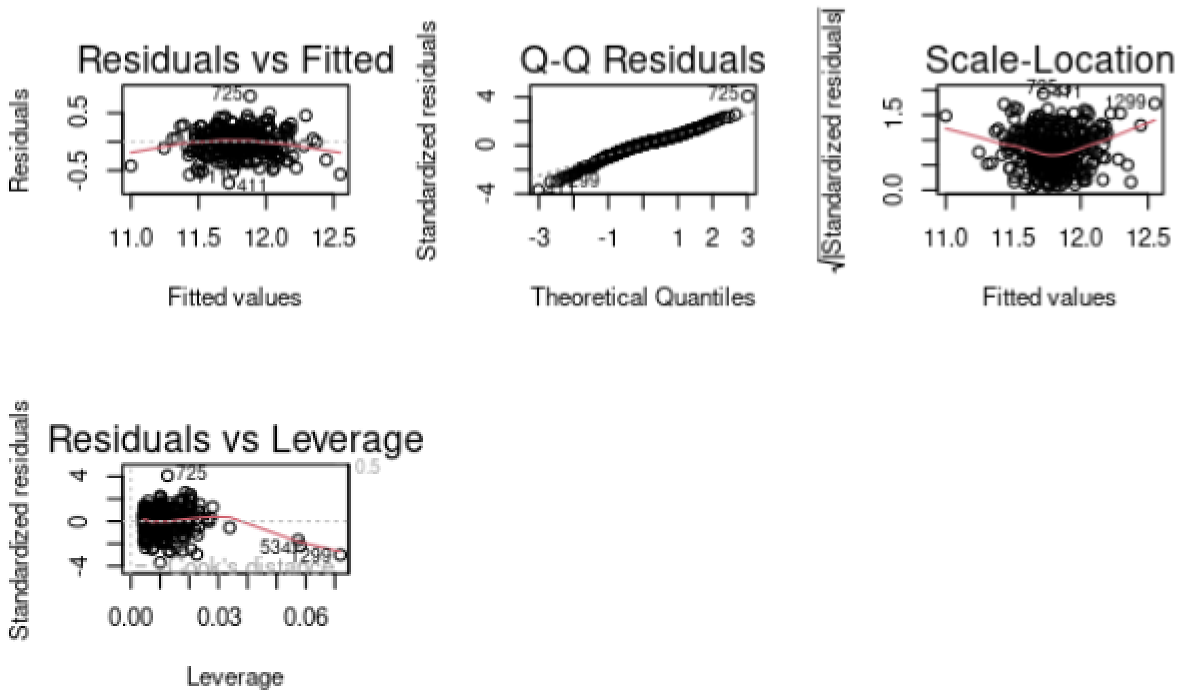


Figure 3. Visual assessment of high-leverage datapoints in Ames dataset. a) Residual and QQ plots with high-leverage datapoints. b) Residual and QQ Plots without high-leverage datapoints.

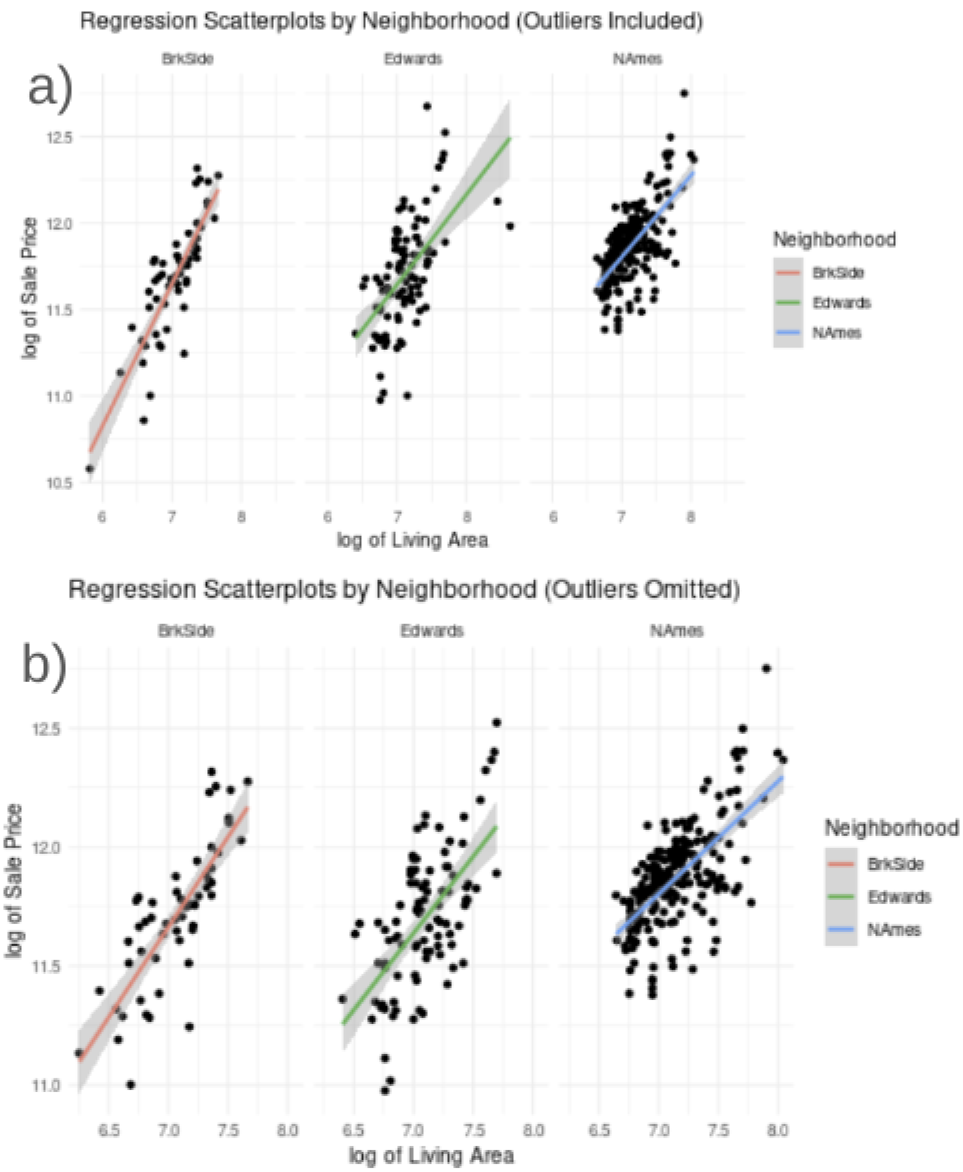
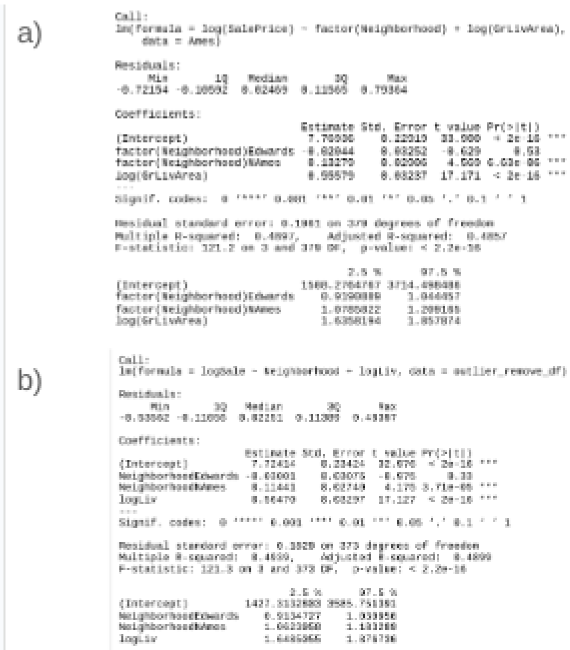


Figure 4. Scatterplot of log(SalePrice) vs log(GrLivArea). a) Outliers included. b) Outliers omitted.



a)

```
Call:
lm(formula = log(SalePrice) ~ factor(Neighborhood) + log(GrLivArea),
    data = Ames)

Residuals:
     Min       1Q   Median       3Q      Max
-0.72154 -0.10592  0.02469  0.11565  0.79364

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                   7.76936    0.22313  33.900  < 2e-16 ***
factor(Neighborhood)Edwards  -0.02044    0.03252  -0.629     0.53
factor(Neighborhood)NAmes     0.13279    0.02906   4.569 6.63e-06 ***
log(GrLivArea)                0.55579    0.03237  17.171  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1981 on 379 degrees of freedom
Multiple R-squared:  0.4897,    Adjusted R-squared:  0.4857
F-statistic: 121.2 on 3 and 379 DF,  p-value: < 2.2e-16

                                  2.5 %      97.5 %
(Intercept)                 1588.2704787 3714.490488
factor(Neighborhood)Edwards    0.9190889    1.044957
factor(Neighborhood)NAmes      1.0785822    1.208185
log(GrLivArea)                 1.6358194    1.857874
```

b)

```
Call:
lm(formula = logSale ~ Neighborhood + logLiv, data = outlier_remove_df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.53562 -0.11056  0.02251  0.11309  0.48387

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           7.72414    0.23424  32.976  < 2e-16 ***
NeighborhoodEdwards  -0.03001    0.03075  -0.976     0.33
NeighborhoodNAmes     0.11443    0.02740   4.176 3.71e-05 ***
logLiv                0.56479    0.03297  17.127  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1828 on 373 degrees of freedom
Multiple R-squared:  0.4939,    Adjusted R-squared:  0.4899
F-statistic: 121.3 on 3 and 373 DF,  p-value: < 2.2e-16

                           2.5 %      97.5 %
(Intercept)         1427.3132883 3585.751381
NeighborhoodEdwards    0.9134727    1.033956
NeighborhoodNAmes      1.0623868    1.183289
logLiv                 1.6485065    1.876736
```

Figure 5. Coefficient summary and confidence intervals. a) Coefficients with high-leverage outliers. b) Coefficients without high-leverage outliers.

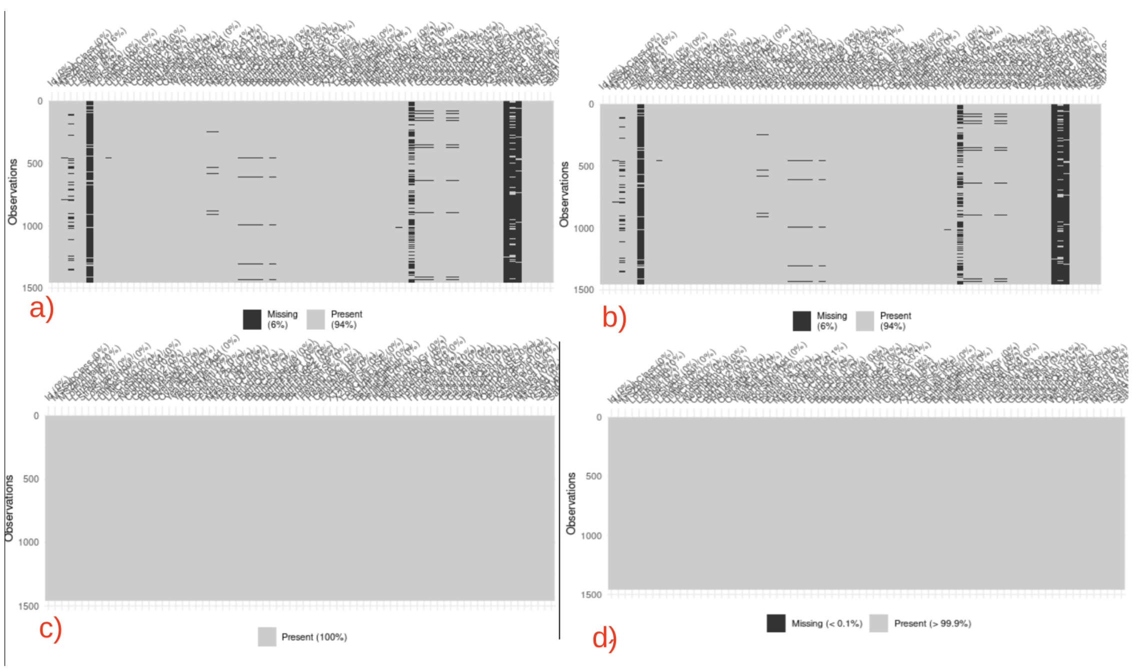


Figure 6: Test and Train datasets before and after cleaning. a) Train dataset before data cleaning. b) Test dataset before data cleaning. c) Train dataset after cleaning. d) Test dataset after cleaning.

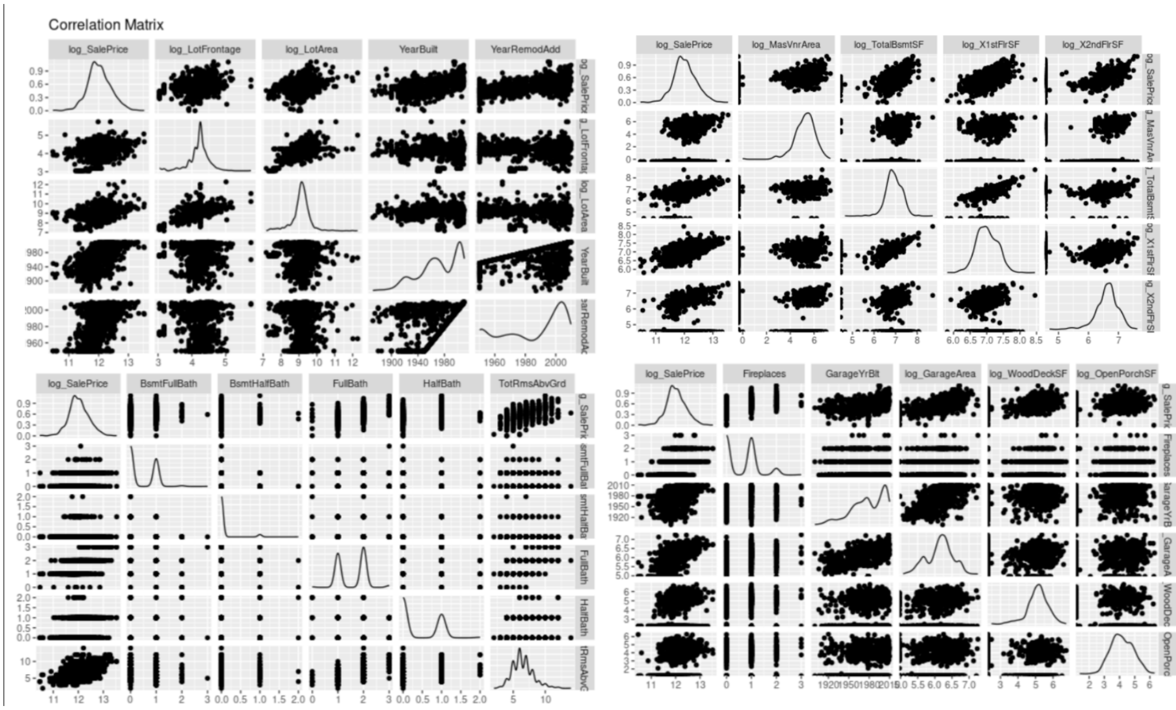


Figure 7: Correlation Matrices, Explanatory Variables vs log(SalePrice)

The GLMSELECT Procedure

Backward Selection Summary						
Step	Effect Removed	Number Effects In	Number Parms In	Adjusted R-Square	SBC	CV PRESS
0		9	22	0.7691*	-4681.6571*	55.0320*

Selection stopped at a local minimum of the cross validation PRESS.

Stop Details				
Candidate For	Effect	Candidate CV PRESS		Compare CV PRESS
Removal	TotRmsAbvGrd	55.1482	>	55.0320

Root MSE	0.19193
Dependent Mean	12.02405
R-Square	0.7725
Adj R-Sq	0.7691
AIC	-3335.95334
AICC	-3335.18454
SBC	-4681.65712
CV PRESS	55.03205

The GLMSELECT Procedure

Forward Selection Summary						
Step	Effect Entered	Number Effects In	Number Parms In	Adjusted R-Square	SBC	CV PRESS
0	Intercept	1	1	0.0000	-2673.2872	233.1096
1	log_X1stFlrSf	2	2	0.3704	-3342.4651	146.7944
2	MSSubClass	3	16	0.6717	-4205.0769	77.4820
3	YearRemodAdd	4	17	0.7324	-4497.4628	62.9954
4	Fireplaces	5	18	0.7452	-4562.8544	59.9625
5	GarageYrBlt	6	19	0.7554	-4616.0642	57.7340
6	log_LotArea	7	20	0.7631	-4656.4895	56.0357
7	FullBath	8	21	0.7680	-4681.1151	54.9612
8	TotRmsAbvGrd	9	22	0.7691*	-4681.6571*	54.8469*
* Optimal Value of Criterion						

Root MSE	0.19193
Dependent Mean	12.02405
R-Square	0.7725
Adj R-Sq	0.7691
AIC	-3335.95334
AICC	-3335.18454
SBC	-4681.65712
CV PRESS	54.84692

Selection stopped because all effects are in the final model.

The GLMSELECT Procedure

Stepwise Selection Summary							
Step	Effect Entered	Effect Removed	Number Effects In	Number Parms In	Adjusted R-Square	SBC	CV PRESS
0	Intercept		1	1	0.0000	-2673.2872	233.1248
1	log_X1stFlrSf		2	2	0.3704	-3342.4651	147.0949
2	MSSubClass		3	16	0.6717	-4205.0769	77.9763
3	YearRemodAdd		4	17	0.7324	-4497.4628	63.5033
4	Fireplaces		5	18	0.7452	-4562.8544	60.3709
5	GarageYrBlt		6	19	0.7554	-4616.0642	58.0424
6	log_LotArea		7	20	0.7631	-4656.4895	56.3586
7	FullBath		8	21	0.7680	-4681.1151	55.2389
8	TotRmsAbvGrd		9	22	0.7691*	-4681.6571*	55.0554*
* Optimal Value of Criterion							

Root MSE	0.19193
Dependent Mean	12.02405
R-Square	0.7725
Adj R-Sq	0.7691
AIC	-3335.95334
AICC	-3335.18454
SBC	-4681.65712
CV PRESS	55.05540

Selection stopped because all effects are in the final model.

Figure 8. Summary Results for Backward (top), Forward (middle), and Stepwise (bottom).

	Adjusted R2	CV Press	Variables for Removal
Backward	0.7691	55.03205	TotRmsAbvGrd
Forward	0.7691	54.84692	None
Stepwise	0.7691	55.05540	None

Figure 9. Primary values used for evaluating selection methods.

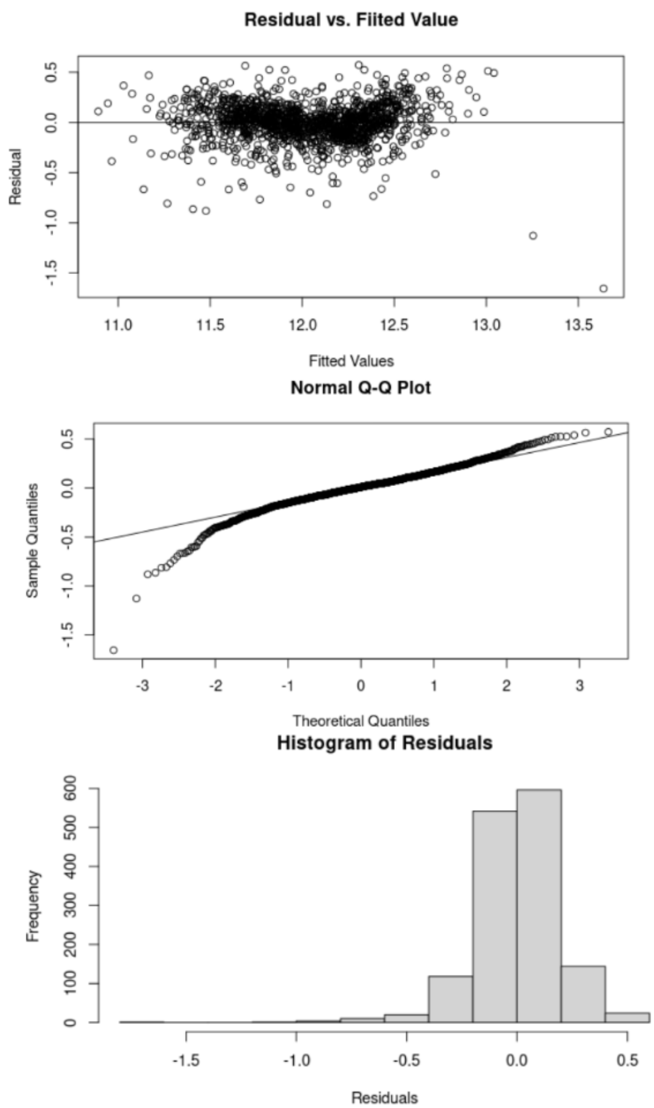


Figure 10. Residual plots for final model. Top) Residual plot; Middle) QQ plot of residuals; Bottom) Histogram of residuals.

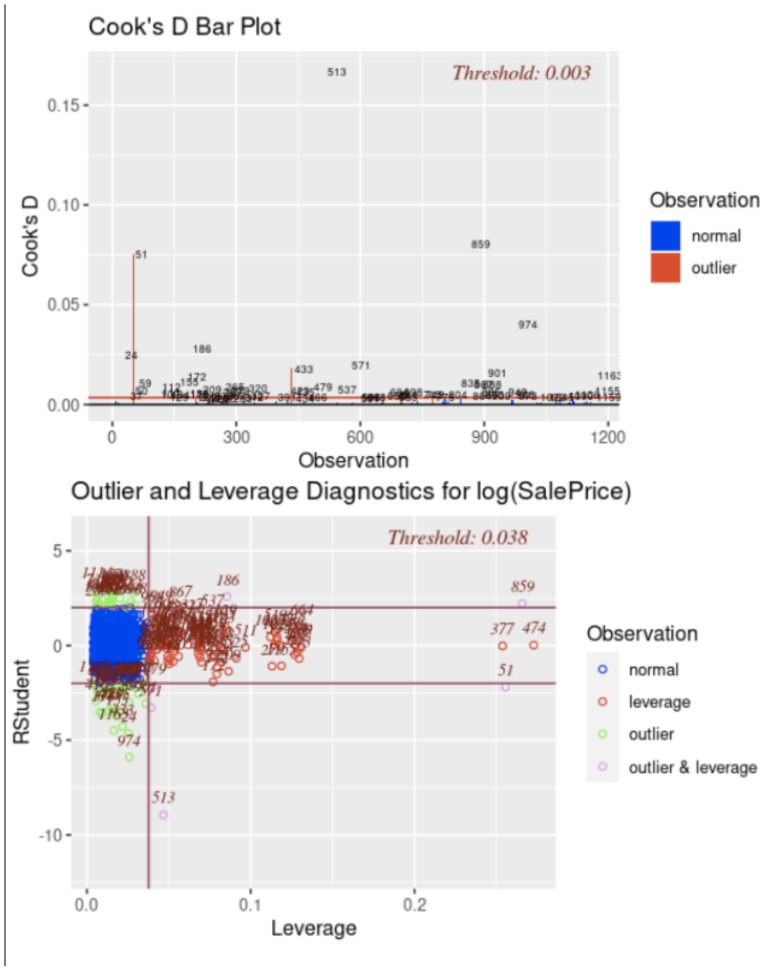


Figure 11. Graphical visualization of potential high-leverage outliers. Top) Cook's D per Observation. Bottom) RStudent values vs Leverage.

Predictive Models	**Adjusted R2**	**CV PRESS**	**Kaggle Score**
Multiple Linear Regression Model (MLR)	0.7691	55.51860	0.20623

Simple Linear Regression (YearBuilt)	0.3436	153.34537	0.32043
Custom (GrLivArea + FullBath)	0.93	1149	0.32034

Figure 12. Adjusted R2, CV PRESS, and Kaggle Score. Top) Final multiple linear regression model. Middle) Simple linear regression with YearBuilt . Bottom) Custom model, GrLivArea and FullBath.

Appendix II

Variables Descriptors
MSSubClass: Identifies the type of dwelling involved in the sale.

20	1-STORY 1946 & NEWER ALL STYLES
30	1-STORY 1945 & OLDER
40	1-STORY W/FINISHED ATTIC ALL AGES
45	1-1/2 STORY - UNFINISHED ALL AGES
50	1-1/2 STORY FINISHED ALL AGES
60	2-STORY 1946 & NEWER
70	2-STORY 1945 & OLDER
75	2-1/2 STORY ALL AGES
80	SPLIT OR MULTI-LEVEL
85	SPLIT FOYER
90	DUPLEX - ALL STYLES AND AGES
120	1-STORY PUD (Planned Unit Development) - 1946 & NEWER
150	1-1/2 STORY PUD - ALL AGES
160	2-STORY PUD - 1946 & NEWER
180	PUD - MULTILEVEL - INCL SPLIT LEV/FOYER
190	2 FAMILY CONVERSION - ALL STYLES AND AGES

MSZoning: Identifies the general zoning classification of the sale.

A	Agriculture
C	Commercial
FV	Floating Village Residential
I	Industrial
RH	Residential High Density
RL	Residential Low Density
RP	Residential Low Density Park
RM	Residential Medium Density

LotFrontage: Linear feet of street connected to property

LotArea: Lot size in square feet

Street: Type of road access to property

Grvl	Gravel
Pave	Paved

Alley: Type of alley access to property

Grvl	Gravel
Pave	Paved
NA	No alley access

LotShape: General shape of property

Reg	Regular
IR1	Slightly irregular
IR2	Moderately Irregular
IR3	Irregular

LandContour: Flatness of the property

Lvl	Near Flat/Level
Bnk	Banked - Quick and significant rise from street grade to building
HLS	Hillside - Significant slope from side to side
Low	Depression

Utilities: Type of utilities available

AllPub	All public Utilities (E,G,W,& S)
NoSewr	Electricity, Gas, and Water (Septic Tank)
NoSeWa	Electricity and Gas Only
ELO	Electricity only

LotConfig: Lot configuration

Inside	Inside lot
Corner	Corner lot
CulDSac	Cul-de-sac
FR2	Frontage on 2 sides of property
FR3	Frontage on 3 sides of property

LandSlope: Slope of property

Gtl	Gentle slope
Mod	Moderate Slope
Sev	Severe Slope

Neighborhood: Physical locations within Ames city limits

Blmngtn	Bloomington Heights
Blueste	Bluestem
BrDale	Briardale
BrkSide	Brookside
ClearCr	Clear Creek
CollgCr	College Creek
Crawfor	Crawford
Edwards	Edwards
Gilbert	Gilbert
IDOTRR	Iowa DOT and Rail Road
MeadowV	Meadow Village
Mitchel	Mitchell
Names	North Ames
NoRidge	Northridge
NPkVill	Northpark Villa
NridgHt	Northridge Heights
NWAmes	Northwest Ames
OldTown	Old Town
SWISU	South & West of Iowa State University
Sawyer	Sawyer
SawyerW	Sawyer West
Somerst	Somerset
StoneBr	Stone Brook
Timber	Timberland
Veenker	Veenker

Condition1: Proximity to various conditions

Artery	Adjacent to arterial street
Feedr	Adjacent to feeder street
Norm	Normal
RRNn	Within 200' of North-South Railroad
RRAn	Adjacent to North-South Railroad
PosN	Near positive off-site feature--park, greenbelt, etc.
PosA	Adjacent to postive off-site feature
RRNe	Within 200' of East-West Railroad
RRAe	Adjacent to East-West Railroad

Condition2: Proximity to various conditions (if more than one is present)

Artery	Adjacent to arterial street
Feedr	Adjacent to feeder street
Norm	Normal
RRNn	Within 200' of North-South Railroad
RRAn	Adjacent to North-South Railroad
PosN	Near positive off-site feature--park, greenbelt, etc.
PosA	Adjacent to postive off-site feature
RRNe	Within 200' of East-West Railroad
RRAe	Adjacent to East-West Railroad

BldgType: Type of dwelling

1Fam	Single-family Detached
2FmCon	Two-family Conversion; originally built as one-family dwelling
Duplx	Duplex
TwnhsE	Townhouse End Unit
TwnhsI	Townhouse Inside Unit

HouseStyle: Style of dwelling

1Story	One story
1.5Fin	One and one-half story: 2nd level finished
1.5Unf	One and one-half story: 2nd level unfinished
2Story	Two story
2.5Fin	Two and one-half story: 2nd level finished
2.5Unf	Two and one-half story: 2nd level unfinished
SFoyer	Split Foyer
SLvl	Split Level

OverallQual: Rates the overall material and finish of the house

10	Very Excellent
9	Excellent

8 Very Good
7 Good
6 Above Average
5 Average
4 Below Average
3 Fair
2 Poor
1 Very Poor

OverallCond: Rates the overall condition of the house

10 Very Excellent
9 Excellent
8 Very Good
7 Good
6 Above Average
5 Average
4 Below Average
3 Fair
2 Poor
1 Very Poor

YearBuilt: Original construction date

YearRemodAdd: Remodel date (same as construction date if no remodeling or additions)

RoofStyle: Type of roof

Flat Flat
Gable Gable
Gambrel Gabrel (Barn)
Hip Hip
Mansard Mansard
Shed Shed

RoofMatl: Roof material

ClyTile Clay or Tile
CompShg Standard (Composite) Shingle
Membran Membrane
Metal Metal
Roll Roll
Tar&Grv Gravel & Tar
WdShake Wood Shakes
WdShngl Wood Shingles

Exterior1st: Exterior covering on house

AsbShng Asbestos Shingles
AsphShn Asphalt Shingles
BrkComm Brick Common
BrkFace Brick Face
CBlock Cinder Block
CemntBd Cement Board
HdBoard Hard Board
ImStucc Imitation Stucco
MetalSd Metal Siding
Other Other
Plywood Plywood
PreCast PreCast
Stone Stone
Stucco Stucco
VinylSd Vinyl Siding
Wd Sdng Wood Siding
WdShing Wood Shingles

Exterior2nd: Exterior covering on house (if more than one material)

AsbShng Asbestos Shingles
AsphShn Asphalt Shingles
BrkComm Brick Common
BrkFace Brick Face
CBlock Cinder Block
CemntBd Cement Board
HdBoard Hard Board
ImStucc Imitation Stucco
MetalSd Metal Siding
Other Other
Plywood Plywood
PreCast PreCast

Stone	Stone
Stucco	Stucco
VinylSd	Vinyl Siding
Wd Sdng	Wood Siding
WdShing	Wood Shingles

MasVnrType: Masonry veneer type

BrkCmn	Brick Common
BrkFace	Brick Face
CBlock	Cinder Block
None	None
Stone	Stone

MasVnrArea: Masonry veneer area in square feet

ExterQual: Evaluates the quality of the material on the exterior

Ex	Excellent
Gd	Good
TA	Average/Typical
Fa	Fair
Po	Poor

ExterCond: Evaluates the present condition of the material on the exterior

Ex	Excellent
Gd	Good
TA	Average/Typical
Fa	Fair
Po	Poor

Foundation: Type of foundation

BrkTil	Brick & Tile
CBlock	Cinder Block
PConc	Poured Contrete
Slab	Slab
Stone	Stone
Wood	Wood

BsmtQual: Evaluates the height of the basement

Ex	Excellent (100+ inches)
Gd	Good (90-99 inches)
TA	Typical (80-89 inches)
Fa	Fair (70-79 inches)
Po	Poor (<70 inches
NA	No Basement

BsmtCond: Evaluates the general condition of the basement

Ex	Excellent
Gd	Good
TA	Typical - slight dampness allowed
Fa	Fair - dampness or some cracking or settling
Po	Poor - Severe cracking, settling, or wetness
NA	No Basement

BsmtExposure: Refers to walkout or garden level walls

Gd	Good Exposure
Av	Average Exposure (split levels or foyers typically score average or above)
Mn	Mimimum Exposure
No	No Exposure
NA	No Basement

BsmtFinType1: Rating of basement finished area

GLQ	Good Living Quarters
ALQ	Average Living Quarters
BLQ	Below Average Living Quarters
Rec	Average Rec Room
LwQ	Low Quality
Unf	Unfinshed
NA	No Basement

BsmtFinSF1: Type 1 finished square feet

BsmtFinType2: Rating of basement finished area (if multiple types)

GLQ	Good Living Quarters
ALQ	Average Living Quarters
BLQ	Below Average Living Quarters
Rec	Average Rec Room
LwQ	Low Quality
Unf	Unfinshed
NA	No Basement

BsmtFinSF2: Type 2 finished square feet

BsmtUnfSF: Unfinished square feet of basement area

TotalBsmtSF: Total square feet of basement area

Heating: Type of heating

Floor	Floor Furnace
GasA	Gas forced warm air furnace
GasW	Gas hot water or steam heat
Grav	Gravity furnace
OthW	Hot water or steam heat other than gas
Wall	Wall furnace

HeatingQC: Heating quality and condition

Ex	Excellent
Gd	Good
TA	Average/Typical
Fa	Fair
Po	Poor

CentralAir: Central air conditioning

N	No
Y	Yes

Electrical: Electrical system

SBrkr	Standard Circuit Breakers & Romex
FuseA	Fuse Box over 60 AMP and all Romex wiring (Average)
FuseF	60 AMP Fuse Box and mostly Romex wiring (Fair)
FuseP	60 AMP Fuse Box and mostly knob & tube wiring (poor)
Mix	Mixed

1stFlrSF: First Floor square feet

2ndFlrSF: Second floor square feet

LowQualFinSF: Low quality finished square feet (all floors)

GrLivArea: Above grade (ground) living area square feet

BsmtFullBath: Basement full bathrooms

BsmtHalfBath: Basement half bathrooms

FullBath: Full bathrooms above grade

HalfBath: Half baths above grade

Bedroom: Bedrooms above grade (does NOT include basement bedrooms)

Kitchen: Kitchens above grade

KitchenQual: Kitchen quality

Ex	Excellent
Gd	Good
TA	Typical/Average
Fa	Fair
Po	Poor

TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)

Functional: Home functionality (Assume typical unless deductions are warranted)

Typ	Typical Functionality
Min1	Minor Deductions 1
Min2	Minor Deductions 2

Mod Moderate Deductions
Maj1 Major Deductions 1
Maj2 Major Deductions 2
Sev Severely Damaged
Sal Salvage only

Fireplaces: Number of fireplaces

FireplaceQu: Fireplace quality

Ex Excellent - Exceptional Masonry Fireplace
Gd Good - Masonry Fireplace in main level
TA Average - Prefabricated Fireplace in main living area or Masonry Fireplace in basement
Fa Fair - Prefabricated Fireplace in basement
Po Poor - Ben Franklin Stove
NA No Fireplace

GarageType: Garage location

2Types More than one type of garage
Attchd Attached to home
Basment Basement Garage
BuiltIn Built-In (Garage part of house - typically has room above garage)
CarPort Car Port
Detchd Detached from home
NA No Garage

GarageYrBlt: Year garage was built

GarageFinish: Interior finish of the garage

Fin Finished
RFn Rough Finished
Unf Unfinished
NA No Garage

GarageCars: Size of garage in car capacity

GarageArea: Size of garage in square feet

GarageQual: Garage quality

Ex Excellent
Gd Good
TA Typical/Average
Fa Fair
Po Poor
NA No Garage

GarageCond: Garage condition

Ex Excellent
Gd Good
TA Typical/Average
Fa Fair
Po Poor
NA No Garage

PavedDrive: Paved driveway

Y Paved
P Partial Pavement
N Dirt/Gravel

WoodDeckSF: Wood deck area in square feet

OpenPorchSF: Open porch area in square feet

EnclosedPorch: Enclosed porch area in square feet

3SsnPorch: Three season porch area in square feet

ScreenPorch: Screen porch area in square feet

PoolArea: Pool area in square feet

PoolQC: Pool quality

Ex Excellent
Gd Good

TA Average/Typical
Fa Fair
NA No Pool

Fence: Fence quality

GdPrv Good Privacy
MnPrv Minimum Privacy
GdWo Good Wood
MnWw Minimum Wood/Wire
NA No Fence

MiscFeature: Miscellaneous feature not covered in other categories

Elev Elevator
Gar2 2nd Garage (if not described in garage section)
Othr Other
Shed Shed (over 100 SF)
TenC Tennis Court
NA None

MiscVal: $Value of miscellaneous feature

MoSold: Month Sold (MM)

YrSold: Year Sold (YYYY)

SaleType: Type of sale

WD Warranty Deed - Conventional
CWD Warranty Deed - Cash
VWD Warranty Deed - VA Loan
New Home just constructed and sold
COD Court Officer Deed/Estate
Con Contract 15% Down payment regular terms
ConLw Contract Low Down payment and low interest
ConLI Contract Low Interest
ConLD Contract Low Down
Oth Other

SaleCondition: Condition of sale

Normal Normal Sale
Abnorml Abnormal Sale - trade, foreclosure, short sale
AdjLand Adjoining Land Purchase
Alloca Allocation - two linked properties with separate deeds, typically condo with a garage unit
Family Sale between family members
Partial Home was not completed when last assessed (associated with New Homes)

Githib Links:

- *Joel Laskow:* https://github.com/jlaskow
- Chris Johnson: https://github.com/cjohnson4510

---

title: "DS 6371 Regression Project_Second Draft" output: html_document date: "2023-11-27" authors:

- "Joel Laskow"
- "Christopher Johnson"

The purpose of this document serves to walk readers through our analysis of housing data found on Kaggle

```
library(leaps)
library(ggfortify)
library(ggcorrplot)
library(base)
library(visdat)
library(tidyverse)
library(ggplot2)
library(olsrr)
library(caret)
library(dplyr)
library(corrplot)
library(ggpubr)
library(GGally)
```

Analysis 1 - Chris Johnson

Read in data and view column names

```
url = "https://raw.githubusercontent.com/cjohnson4510/Housing-Project/main/train.csv"
Htrain = read.csv(url)
colnames(Htrain)
```

Change neighborhood to factor

```
class(Htrain$Neighborhood)
Htrain$Neighborhood=as.factor(Htrain$Neighborhood)
levels(Htrain$Neighborhood)
```

Create 'Ames' Dataframe with neighborhoods of interest

```
am=grep("NAmes|Edwards|BrkSide", Htrain$Neighborhood, ignore.case = TRUE)
Ames=Htrain[am,]
Ames$Neighborhood
```

Use Naniar library to find missing values in variables of interest

```
library(naniar)
mv=miss_var_summary(Ames)
print(mv, n=100)
Ames$GrLivArea
```

No Missing values in variables of interest Create model and Plot to See if Data is normally Distributed

```
plot(Ames$GrLivArea,Ames$SalePrice)
hist(Ames$GrLivArea)
hist(Ames$SalePrice)
model=lm(SalePrice~Neighborhood+GrLivArea, Ames)
summary(model)
plot(model)
```

GrlivArea and Sale Price non-normally distributed, log transformation performed and plotted

```
logLiv=log(Ames$GrLivArea)
logSale=log(Ames$SalePrice)
hist(logLiv)
hist(logSale)
plot(logLiv, logSale)
```

Create new data frame with log transformations. Model coeffecients, confidence intervals and adj r-sqaured. Assumptions: QQ plot looks normal, residuals and leverages address below

```
logAmes=cbind(Ames,logSale, logLiv)
logModel=lm(logSale~Neighborhood+logLiv, logAmes)
summary(logModel)
confint(logModel)
plot(logModel)
```

Calculate CV Press of the model

```
set.seed(123)
k <- 5
fold_size <- nrow(logAmes) / k
cv_press <- 0
for (i in 1:k) {
  test_indices <- ((i-1) * fold_size + 1):(i * fold_size)
  test_data <- logAmes[test_indices, ]
  train_data <- logAmes[-test_indices, ]
  model <- lm(logSale ~ Neighborhood + logLiv, data = train_data)
  predictions <- predict(model, test_data)
  cv_press[i] <- cv_press + sum((test_data$logSale - predictions) ^ 2)
}
mean(cv_press)
```

To address leverages and residuals we omitted the 3 largest leverage and the 3 largest residual data points Created new model and plot for comparison Leverages and residuals looks normally distributed after datapoints omitted

```
leverages=hatvalues(logModel)
order(leverages, decreasing=TRUE)

resid=abs(resid(logModel))
order(resid, decreasing=TRUE)

outlier_remove_df=logAmes[-c(339, 136, 131, 190, 104, 186  ),]
ordfModel=lm(logSale~Neighborhood+logLiv, outlier_remove_df)
summary(ordfModel)
confint(ordfModel)
plot(ordfModel)
```

Orginal data, Plot log Model, all neighborhoods

```
library(ggplot2)
ggplot(logAmes, aes(x = logLiv, y = logSale)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE) +
  labs(title = "Scatter Plot with Regression Lines all neighborhoods", x = "log of Living Are
  theme_minimal()
```

Omitted data, Plot log model, all neighborhoods

```
ggplot(outlier_remove_df, aes(x = logLiv, y = logSale)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE) +
  labs(title = "Scatter Plot with Regression Lines all neighborhoods", x = "log of Living Are
  theme_minimal()
```

Original data, Plot Log Data by Neighboorhood

```
ggplot(logAmes, aes(x = logLiv, y = logSale, color=Neighborhood)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Scatter Plot with Regression Lines all neighborhoods", x = "log of Living Are
  theme_minimal()
```

Omitted data, Plot Log Data by Neighboorhood

```
ggplot(outlier_remove_df, aes(x = logLiv, y = logSale, color=Neighborhood)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Scatter Plot with Regression Lines all neighborhoods", x = "log of Living Are
  theme_minimal()
```

Original data, Separate plots for each Neighboorhood

```
ggplot(logAmes, aes(x = logLiv, y = logSale)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE, aes(color = Neighborhood)) +
  labs(title = "Regression Scatterplots by Neighborhood (Outliers Included)", x = "log of Liv
  theme_minimal() +
  facet_wrap(~Neighborhood, scales = "fixed")
```

Omitted data, Separate plots for each Neighboorhood

```
ggplot(outlier_remove_df, aes(x = logLiv, y = logSale)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE, aes(color = Neighborhood)) +
  labs(title = "Regression Scatterplots by Neighborhood (Outliers Omitted)", x = "log of Livi
  theme_minimal() +
  facet_wrap(~Neighborhood, scales = "fixed")
```

Transform logmodel back to original scale for final interpretation with confidence intervals

```
summary(logModel)
exp(confint(logModel))
```

Transform omitted model back to original scale for final interpretation with confidence intervals

The data suggest that a doubling of GrLivArea with equates to a multiplicative change of 2^0.55579 in the median of the SalePrice. A 9% confidence interval for the Brookside neighborhood multiplicative increase is (2^1.6358194, 2^1.857874) ; for Edwards, we expect (2^0.9190889, 2^1.044457); and for NAmes we expect (2^1.0785822, 2^1.209165))

```
summary(ordfModel)
exp(confint(ordfModel))
```

The outlier-omitted data suggest that a doubling of GrLivArea with equates to a multiplicative change of 2^0.56470 in the median of the SalePrice. A 95% confidence interval for the Brookside neighborhood multiplicative increase is (2^1.6485055, 2^1.876736) ; for Edwards, we expect (2^0.9134727, 2^1.030956); and for NAmes we expect (2^1.0623958, 2^1.183289))

Analysis 2 - Joel Laskow

```
# Train Dataset

train <- data.frame(read.csv("/cloud/project/DS 6371 Housing Project/train.csv", header=TRUE)


# Test Dataset

test <- read.csv("/cloud/project/DS 6371 Housing Project/test.csv", header=TRUE)

test<-data.frame(test)
```

Replace any instances of "NA" as a class level with "None" to avoid confusion

```
# train

train<- train %>%
  mutate_all(~ ifelse(. == "NA", "None", .))
```

```
# test

test<- test %>%
  mutate_all(~ ifelse(. == "NA", "None", .))
```

Assessing Numeric NA values within the datasets

```
# training set

numerictrain<-train[sapply(train,is.numeric)]
vis_miss(numerictrain)
vis_miss(train)
```

```
# test set

numerictest<-test[sapply(test,is.numeric)]
vis_miss(test)
vis_miss(numerictest)
```

We see from missing value tests that while we're only missing 0.6% of our training set and 0.6% of our test set, we're missing 18% of our LotsFrontage data in training and 16% in testing. We're further missing 6% of our GarageYrBlt data in the training set and 5% in the testing set. This quantity of missing errors could wildly skew our results.

Assessing Categorical NA values within the datasets

```
# train

nonnumerictrain<-train[sapply(train,is.character)]

vis_miss(nonnumerictrain)
```

```
# test

nonnumerictest<-test[sapply(test,is.character)]

vis_miss(nonnumerictest)
```

10.6% of our training dataset has missing values. Due to the severity of missing values in Alley (93%), FireplaceQu (50%), PoolQC (100%), Fence (80%), and MiscFeat we cannot impute missing values with the most common categorical level. For this reason we will remove these columns from the dataset.

```
# Removing Alley, Fireplace, PookQC, Fence, and MiscFeature from our dataset


# train

trainprime<-subset(train, select= -c(Alley, FireplaceQu, PoolQC, Fence, MiscFeature))


# test

testprime<-subset(test, select= -c(Alley, FireplaceQu, PoolQC, Fence, MiscFeature))
```

Addressing remaining NA values:

```
##### Numeric Datasets

# train

## Find the mean of columns to impute:

### LotsFrontage

x<-mean(train$LotFrontage, na.rm=TRUE)

### GarageYrBlt

y<-mean(train$GarageYrBlt, na.rm=TRUE)
```

```
### MasVnrArea

z<-mean(train$MasVnrArea, na.rm=TRUE)


## Impute mean for each column

### LotFrontage

trainprime$LotFrontage<-replace(train$LotFrontage, is.na(train$LotFrontage), x)

### GarageYrBlt

trainprime$GarageYrBlt<-replace(train$GarageYrBlt, is.na(train$GarageYrBlt), y)

### MasVnrArea

trainprime$MasVnrArea<-replace(train$MasVnrArea, is.na(train$MasVnrArea), z)

# test

## Find the mean of columns to impute:

### LotsFrontage

x<-mean(testprime$LotFrontage, na.rm=TRUE)

### GarageYrBlt

y<-mean(testprime$GarageYrBlt, na.rm=TRUE)

### MasVnrArea

z<-mean(testprime$MasVnrArea, na.rm=TRUE)

# BsmtFinSF1
d<-mean(testprime$BsmtFinSF1, na.rm=TRUE)

# BsmFinSF2
e<-mean(as.numeric(testprime$BsmFinSF2), na.rm=TRUE)

# BsmtUnfSF
f<-mean(testprime$BsmtUnfSF, na.rm=TRUE)
```

```
# TotalBsmtSF
g<-mean(testprime$TotalBsmtSF, na.rm=TRUE)

# BsmtFullBath
h<-mean(testprime$BsmtFullBath, na.rm=TRUE)

# BsmtHalfBath
i<-mean(testprime$BsmtHalfBath, na.rm=TRUE)

# GarageCars
j<-mean(testprime$GarageCars, na.rm=TRUE)

# GarageArea
k<-mean(testprime$GarageArea, na.rm=TRUE)

## Impute mean for each column

### LotFrontage

testprime$LotFrontage<-replace(test$LotFrontage, is.na(test$LotFrontage), x)

### GarageYrBlt

testprime$GarageYrBlt<-replace(test$GarageYrBlt, is.na(test$GarageYrBlt), y)

### MasVnrArea

testprime$MasVnrArea<-replace(test$MasVnrArea, is.na(test$MasVnrArea), z)

# BsmtFinSF1
testprime$MasVnrArea<-replace(test$BsmtFinSF1, is.na(test$MasVnrArea), d)

# BsmFinSF2
testprime$MasVnrArea<-replace(as.numeric(test$BsmFinSF2), is.na(test$MasVnrArea), e)

# BsmtUnfSF
testprime$MasVnrArea<-replace(test$BsmtUnfSF, is.na(test$MasVnrArea), f)


# TotalBsmtSF
testprime$MasVnrArea<-replace(test$TotalBsmtSF, is.na(test$MasVnrArea), g)
```

```
# BsmtFullBath
testprime$MasVnrArea<-replace(test$BsmtFullBath, is.na(test$MasVnrArea), h)


# BsmtHalfBath
testprime$MasVnrArea<-replace(test$BsmtHalfBath, is.na(test$MasVnrArea), i)


# GarageCars

testprime$MasVnrArea<-replace(test$GarageCars, is.na(test$MasVnrArea), j)

# GarageArea

testprime$MasVnrArea<-replace(test$GarageArea, is.na(test$MasVnrArea), k)

##### Categorical (Non-numeric)

# trainnew

get_mode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

for (col in names(trainprime)) {
  if (is.character(trainprime[[col]]) && anyNA(trainprime[[col]])) {
    trainprime[[col]][is.na(trainprime[[col]])] <- get_mode(trainprime[[col]])
  }
}

# testnew
get_mode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

for (col in names(testprime)) {
  if (is.character(testprime[[col]]) && anyNA(testprime[[col]])) {
    testprime[[col]][is.na(testprime[[col]])] <- get_mode(testprime[[col]])
  }
```

```
}

trainduplicate<-trainprime
testduplicate<-testprime



vis_miss(trainprime)


vis_miss(testprime)



# trainnew

get_mode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

for (col in names(trainprime)) {
  if (is.character(trainprime[[col]]) && anyNA(trainprime[[col]])) {
    trainprime[[col]][is.na(trainprime[[col]])] <- get_mode(trainprime[[col]])
  }
}

# Mark MSSubClass as factor

trainprime$MSSubClass<-factor(trainprime$MSSubClass)
testprime$MSSubClass<-factor(testprime$MSSubClass)


vis_miss(trainprime)
```

Our datasets are now clear of NA values

Unequal level between train and test data

If we compare the levels of testprime and trainprime within the MSSubClass feature, we see testprime has a level that does not appear in the training set.

```
levels(testprime$MSSubClass)
```

```
levels(trainprime$MSSubClass)
```

Level 150 appears only in the test set

```
subset(testprime, MSSubClass == "150")
```

Only one row in the test set contains MSSubClass of 150. Descriptor: 150 1-1/2 STORY PUD - ALL AGES

Because of the descriptor similarity to level 50 (50 1-1/2 STORY FINISHED ALL AGES), the level of this row was changed

```
# Before change
table(testprime$MSSubClass)

row_to_change <- which(testprime$MSSubClass == "150")

# Change the level to 50
testprime$MSSubClass[row_to_change] <- "50"

# Verify the change
table(testprime$MSSubClass)
```

If we are to build an efficient model, we must be selective with our variables. We have 3 methods to select our variables, all of which will be discussed later: Forward Selection, Backward Selection, and Stepwise Selection.

```
trainnew <- trainprime %>%
```

```
  mutate(log_SalePrice = log(SalePrice)) %>%
  mutate(log_LotFrontage = log(LotFrontage)) %>%
  mutate(log_LotArea = log(LotArea))

selected_vars <- c("log_SalePrice", "log_LotFrontage", "log_LotArea", "YearBuilt", "YearRemoc

# Calculating correlations

subset_data <- trainnew[, selected_vars]

# Create correlation plot using ggpairs
ggpairs(subset_data, upper=list (continuous="points"), title = "Correlation Matrix")
```

High correlation between LotArea and LotFrontage, YearBuilt and YearRemodAdd.

Variables to remove: LotFrontage, YearBuilt

Next Matrix Batch

```
trainnew <- trainprime %>%
  mutate(log_SalePrice = log(SalePrice)) %>%
  mutate(log_MasVnrArea = log(MasVnrArea)) %>%
  mutate(log_TotalBsmtSF = log(TotalBsmtSF))%>%
  mutate(log_X1stFlrSF = log(X1stFlrSF))%>%
  mutate(log_X2ndFlrSF = log(X2ndFlrSF))

selected_vars <- c("log_SalePrice", "log_MasVnrArea", "log_TotalBsmtSF", "log_X1stFlrSF", "lc

# Calculating correlations

subset_data <- trainnew[, selected_vars]

# Create correlation plot using ggpairs
ggpairs(subset_data, upper=list (continuous="points"), title = "Correlation Matrix")
```

High correlation between total TotalBsmntSF and 1stFlrSF. Minimal correlation between MasVnrArea and Sale Price.

Variables to remove: TotalBsmntSF, MasVnrArea

3rd Matrix Batch

```
# Deal with outliers by logging large-value columns

# NoteL LowQualFinSF ignored

# GrLivArea ignored, similar to 1stFlrSF but only accounts for living area

# BedroomAbvGr and KitchenAbvGr ignored, similar to TotRmsAbvGr


# Log transform 'SalePrice' column
numerictrainnew <- trainprime %>%
  mutate(log_SalePrice = log(SalePrice))

# Select necessary variables
selected_vars <- c("log_SalePrice", "BsmtFullBath", "BsmtHalfBath", "FullBath", "HalfBath", '

# Create a subset of data with selected variables
subset_data <- numerictrainnew[, selected_vars]

# Create correlation plot using ggpairs
ggpairs(subset_data, upper=list (continuous="points"), title = "Correlation Matrix")
```

No correlation seen between log(SalePrice) and BsmtFullBath, BsmtHalfBath, and HalfBath. Possible positive correlation observed between FullBath and TotRmsAbvGrd

Variables to remove: BsmtFullBath, BsmtHalfBath, and HalfBath

4th Matrix Batch

```
# Log transform 'SalePrice' column
numerictrainnew <- trainprime %>%
  mutate(log_SalePrice = log(SalePrice))%>% mutate(log_GarageArea = log(GarageArea)) %>% muta
```

```
# Select necessary variables

## GarageCars removed, similar to GarageArea

selected_vars <- c("log_SalePrice", "Fireplaces", "GarageYrBlt", "log_GarageArea", "log_WoodD

# Create a subset of data with selected variables
subset_data <- numerictrainnew[, selected_vars]

# Create correlation plot using ggpairs
ggpairs(subset_data, upper=list (continuous="points"), title = "Correlation Matrix")
```

High correlation between GargeYrBlt and and log_GarageArea; both show high correlation with log_SalePrice. log_GarageArea removed for simplicity.

We also see evidence of correlation between Fireplaces and log_SalePrice

Slight correlation observed between log_WoodDeckSF and log_SalePrice, and log_OpenPorchSF and log_SalePrice. Due to the weak correlation, both WoodDeckSF and OpenPorchSF will be removed.

The following variables were also ignored due to the high prevlance of 0 (i.e., "Not Applicable" or "None") within the column. There does not appear to be a sufficient quantity of values to make an informed interpretation from these variables.

- EnclosedPorch
- X3SsnPorch
- PoolArea
- MiscVal

At this point we have the following numerical variables left:

- FirePlaces
- GarageYrBlt
- FullBath

- TotRmsAbvGrd
- log(1stFlrSF)
- log(LotArea)
- YearRemodAdd

  We will move forward with the following categorical variable from the training set:

- MSSubClass

Subdividing our training set (80:20) to produce a validation and training set

```
# We will split out training dataset to train our model and test its quality
indices <- sample(1:nrow(trainprime), 0.8 * nrow(trainprime))

# Creating the training and validation sets
training_set <- trainprime[indices, ]  # 80% of data for training
validation_set <- trainprime[-indices, ]  # Remaining 20% for validation
```

All variable selection was performed in SAS using the trainprime dataset. Final CV Press and Adjusted R2 Values are shown below for each selection method.

Forward:

- No Variables Removed
- CV Press: 54.84692
- Adj. R2: 0.7691

Backward:

- TotRmsAbvGrd suggested for removal
- Candidate CV Press: 55.1482
- Compare CV Press: 55.0320
- Adj. R2: 0.7691

Stepwise:

- No variables removed

- CV Press: 55.05540
- Adj. R2: 0.7691

We will move forward with the full model (no variables removed). Removal of TotRmsAbvGrd provides miminimal change in CV Press.

Full Model:

log(SalePrice) = Fireplaces + GarageYrBlt + FullBath + TotRmsAbvGrd + log(X1stFlrSF) + log(LotArea) + YearRemodAdd + MSSubClass

Assumption checks for final model

```
fitfull<-lm(log(SalePrice) ~ Fireplaces + GarageYrBlt + FullBath + TotRmsAbvGrd + log(X1stFlr
```

```
# Residuals plot

res<-resid(fitfull)

plot(fitted(fitfull), res, xlab="Fitted Values", ylab="Residual", main = "Residual vs. Fiitec
abline(0,0)



# QQ plot
qqnorm(resid(fitfull))
qqline(resid(fitfull))


# Histogram of residuals
hist(resid(fitfull), xlab="Residuals", ylab="Frequency", main="Histogram of Residuals")
```

Visualizing Leverage

```
ols_plot_cooksd_bar(fitfull)
```

```
ols_plot_resid_stand(fitfull)
```

```
ols_plot_resid_lev(fitfull)
```

Testing Adjusted R Square Change with Leverage Points Removed

```
# Make new dataet with high-leverage outliers removed

trainnew_no_leverage <- trainprime[-51, -513, -859]


# Perform 80:20 split with edited dataset
train_indices <- sample(nrow(trainnew_no_leverage), 0.8 * nrow(trainnew_no_leverage))  # 80%
train_data_no_outliers <- trainnew_no_leverage[train_indices, ]
test_data_no_outliers <- trainnew_no_leverage[-train_indices, ]  # Remaining 20% for testing


# Refit the model without high leverage points

fit_no_leverage <- lm(fitfull, data = train_data_no_outliers)

fit_no_leverage

summary(fit_no_leverage)


fit_with_leverage <- lm(fitfull, data = training_set)

summary(fit_with_leverage)
```

No significant change in RSE or Adjusted R Square with removal of high-leverage outliers.

Building Predictions

Final MLR model

```
trainprime3<-trainprime
testprime3<-testprime
# Backward

fitfull <- lm(log(SalePrice) ~ Fireplaces + GarageYrBlt + FullBath + TotRmsAbvGrd + log(X1stF

prediction<-predict(fitfull, newdata=testprime3)

testprime3$logSalePrice<-prediction

testprime3$SalePrice<-exp(testprime3$logSalePrice)

houseprices3<-testprime3[c("Id", "SalePrice")]


write.csv(houseprices3,"testhouseprices_final.csv", row.names=TRUE)
```

Final Kaggle Score: 0.20089

Simple Linear Regression Model (YearBuilt)

```
# SLR


slr <- lm(log(SalePrice) ~ YearBuilt, data = trainprime)

predicted <- predict(slr, newdata = testprime)

testprime3$logSalePrice<-predicted

testprime3$SalePrice<-exp(testprime3$logSalePrice)

houseprices3<-testprime3[c("Id", "SalePrice")]


write.csv(houseprices3,"testhouseprices_SLR.csv", row.names=TRUE)
```

Explanatory variables: GrLivArea + FullBath (Custom)

```
testprime$FullBath<-as.numeric(testprime$FullBath)

custom <- lm(log(SalePrice) ~ log(GrLivArea) + FullBath, data = trainprime)

predicted <- predict(custom, newdata = testprime)

testprime3$logSalePrice<-predicted

testprime3$SalePrice<-exp(testprime3$logSalePrice)

houseprices3<-testprime3[c("Id", "SalePrice")]


write.csv(houseprices3,"testhouseprices_Custom.csv", row.names=TRUE)
```

SAS Code

/* Generated Code (IMPORT) / / Source File: trainprime.csv / / Source Path: /home/u63533968 / / Code generated on: 11/30/23, 1:47 AM */

%web_drop_table(trainprime);

FILENAME REFFILE '/home/u63533968/trainprime.csv';

PROC IMPORT DATAFILE=REFFILE DBMS=CSV OUT=trainprime; GETNAMES=YES; RUN;

PROC CONTENTS DATA=trainprime; RUN;

%web_open_table(trainprime);

data trainprime_log; set trainprime;

```
/* Log transformation for SalePrice, X1stFlrSF, and LotArea */
log_SalePrice = log(SalePrice);
log_X1stFlrSf = log(X1stFlrSf);
log_LotArea = log(LotArea);
log_GrLivArea = log(GrLivArea);
```

run;

proc glmselect data=trainprime_log; class MSSubClass; model log_SalePrice = FirePlaces GarageYrBlt FullBath TotRmsAbvGrd log_X1stFlrSf log_LotArea YearRemodAdd MSSubClass /

```
selection=Backward(stop=CV) cvmethod=random(10) stats=adjrsq; run;

proc glmselect data=trainprime_log; class MSSubClass; model log_SalePrice = FirePlaces GarageYrBlt FullBath TotRmsAbvGrd log_X1stFlrSf log_LotArea YearRemodAdd MSSubClass / selection=Forward(stop=CV) cvmethod=random(10) stats=adjrsq; run;

proc glmselect data=trainprime_log; class MSSubClass; model log_SalePrice = FirePlaces GarageYrBlt FullBath TotRmsAbvGrd log_X1stFlrSf log_LotArea YearRemodAdd MSSubClass / selection=Stepwise(stop=CV) cvmethod=random(10) stats=adjrsq; run;

/* SLR: YearBuilt only */

proc glmselect data=trainprime_log; class MSSubClass; model log_SalePrice = YearBuilt / selection=Backward(stop=CV) cvmethod=random(10) stats=adjrsq; run;

proc glmselect data=trainprime_log; class MSSubClass; model log_SalePrice = FirePlaces GarageYrBlt FullBath TotRmsAbvGrd log_X1stFlrSf log_LotArea YearRemodAdd MSSubClass / selection=Stepwise(stop=CV) cvmethod=random(10) stats=adjrsq; run;
```