# Introduction

There are many factors that determine the value of a home in the real estate market. We have been provided a dataset containing explanatory variables and the final sale prices of homes. We will attempt to predict the final sales price of a home for Ames, Iowa using regression.

# Data Description

The data was provided from the following educational kaggle competition:

<https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data>

The data consists of 79 explanatory variables and the final Sales Price of a home in Ames, Iowa. The [Ames Housing dataset](http://www.amstat.org/publications/jse/v19n3/decock.pdf) was compiled by Dean De Cock for use in data science education.

For analysis question 1, we are only interested in the following variables:

GrLIvArea

SalesPrice

Neighborhoods (NWAmes, Edwards and BrkSide specifically)

For analysis question 2, the best model we calculated, from a forward model, had the following variables:

OverallQual

GrLivArea

Neighborhood

BsmtQual

**Maybe explain the highlighted variables.**

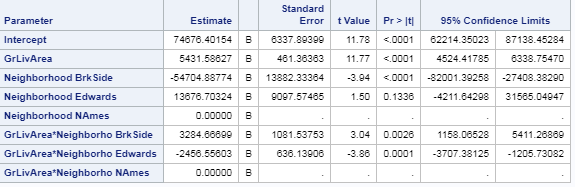
# Analysis Question 1:

## Restatement of Problem

Century 21 Ames only sells houses in the NAmes, Edwards and BrkSide neighborhoods and would like to simply get an estimate of how the SalePrice of the house is related to the square footage of the living area of the house (GrLIvArea) and if the SalesPrice (and its relationship to square footage) depends on which neighborhood the house is located in.

## Build and Fit the Model

**The following model was built using glm regression (code referenced in Appendix).**



|  |  |
| --- | --- |
|  | Note that the Edwards neighborhood regression line seems to be heavily influenced by two points with a GrLivArea greater than 30. |

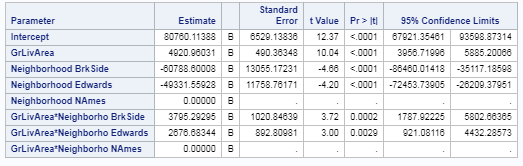
## Checking Assumptions

|  |  |
| --- | --- |
|  | **Normality:** The data appears to be normal given the bell shaped distribution in the histogram highlighted in the red box of illustration. We also see evidence of normality in the qqplot in the red box as well.  **Linear Trend:** The pairwise scatter plot in orange box indicates a strong linear trend.  **Equal SD:** There is little evidence from the scatter plots of heteroscedasticity and the Residual plot in the blue box shows a nice random cloud.  **Independence:**  We will assume the observations are independent.  **Influential point analysis:**  We see that we have two points circled in green that have high leverage over the rest of the data. I believe these are the points above 30 in GrLivArea.  Also, we see that the Cook’s D has a very high point, circled in purple, at 2.5 that is concerning. |

## Comparing Competing Models

An R-squared of .4785 was obtained as well as an adjusted R-squared of .4714.

## Parameters



Estimates

Interpretation

Confidence Intervals

## Conclusion

A short summary of the analysis.

# Analysis Question 2

## Restatement of Problem

We would like to build the most predictive model for sale prices of homes in all of Ames Iowa.

## Model Selection

We built four models with different methods of selecting important variables. The following models were completed:

* + - * Stepwise
      * Forward
      * Forward without Outliers
      * Backward
      * Custom

## Checking Assumptions

|  |  |
| --- | --- |
|  | **Normality:** The data appears to be normal given the bell shaped distribution in the histogram.  **Linear Trend:** The pairwise scatter a strong linear trend.  **Equal SD:** There is little evidence from the scatter plots of heteroscedasticity and the Residual plot shows a nice random cloud.  **Independence:**  We will assume the observations are independent.  **Influential point analysis:**  We see that we have two points that have high leverage over the rest of the data. When influential points were removed, we noticed it negatively reduced our kaggle score so we are leaving them in.  There 1 pretty high cooks d, but we left it in for a better kaggle score as explained above. |

## Comparing Competing Models

|  |  |  |  |
| --- | --- | --- | --- |
| **Predictive Models** | **Adjusted R2** | **CV PRESS** | **Kaggle Score** |
| Forward Model | 0.8068 | 1848 | .18533 |
| Forward Model with No Outliers | 0.9071 | 9188 | .28203 |
| Backward Model | .9212 | 4079 | 1.35659 |
| Stepwise Model | .8068 | 1842 | .18533 |
| Custom Model | .8101 | 2072 | .24523 |

## Conclusion: A short summary of the analysis.

After completing the Forward, Backward Model, and Stepwise model with outliers, we created a custom model by getting results (p-values) from the other models and chose variables that were extremely significant. Once we had all four models completed, we submitted to kaggle to determine which model had the best kaggle test score to focus on that model. The forward model seemed to have the best score. We then evaluated the assumptions and outliers and removed outliers from the training set, but found that the kaggle score decreased so we are going with the original Forward Model including all the data.

# Appendix

## Question 1 Code:

/\*

\* Import the Training Data Set

\*/

PROC IMPORT OUT= WORK.train

DATAFILE= "/home/marinfamily1010/sasuser.v94/Data/train.csv"

DBMS=CSV REPLACE;

GETNAMES=YES;

DATAROW=2;

RUN;

proc sql;

create table work.train3 as

/\*Dividing the GrLivArea by 100\*/

select Neighborhood, GrLivArea/100 as GrLivArea, SalePrice

from train

where Neighborhood in ('NAmes','Edwards','BrkSide')

order by Neighborhood;

run;

/\*

\* At least one neighborhood has a different slope. Using Different slopes model.

\*/

proc sgscatter data = work.train3;

by Neighborhood;

plot SalePrice \* GrLIvArea;

run;

proc glm data = work.train3 plots = ALL;

class Neighborhood (ref = "NAmes");

model SalePrice = GrLIvArea | Neighborhood / solution clparm;

output out = cookd cookd = cookd;

run;

/\*Limiting the high cookd value and the high leverage points out of data (influential points)\*/

proc sql;

create table work.train4 as

select Neighborhood, GrLivArea as GrLivArea, SalePrice

from COOKD

where Neighborhood in ('NAmes','Edwards','BrkSide')

and cookd < 2.74

and GrLivArea < 30

order by Neighborhood;

run;

/\*Re-running scatter and model without influential points\*/

proc sgscatter data = work.train4;

by Neighborhood;

plot SalePrice \* GrLIvArea;

run;

proc glm data = work.train4 plots = ALL;

class Neighborhood (ref = "NAmes");

model SalePrice = GrLIvArea | Neighborhood / solution clparm;

output out = cookd2 cookd = cookd2;

run;

## Question 2 Code:

PROC IMPORT OUT= WORK.train

DATAFILE= "/home/marinfamily1010/sasuser.v94/Data/train.csv"

DBMS=CSV REPLACE;

GETNAMES=YES;

DATAROW=2;

RUN;

PROC IMPORT OUT= WORK.test

DATAFILE= "/home/marinfamily1010/sasuser.v94/Data/test.csv"

DBMS=CSV REPLACE;

GETNAMES=YES;

DATAROW=2;

RUN;

data work.test;

set test;

SalePrice = .;

;

data work.train2;

set train test;

run;

data work.train1;

set work.train2;

rename '1stFlrSF'n = FirstFlrSF '2ndFlrSF'n = SecondFlrSF '3SsnPorch'n = ThreesnPorch;

run;

options mlogic symbolgen; /\*\*\*\*\*\*\*\*\*\*Options are used to help see the & stuff\*\*\*\*\*\*\*\*\*\*\*\*\*/;

/\*\*\*\*\*\*\*Dealt with missing years for GarageYrBlt and also Converted LotFrontage to Numeric\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/;

data kaggle;

set work.train1;/\*Update to your imported data set\*/

if LotFrontage ne "NA" then LotFrontage\_clean = input(lotfrontage,8.);

if GarageYrBlt = . then GarageYrBlt = YearBuilt;

drop LotFrontage;

run;

/\*\*\*\*\*\*\*\*This code replaces the missing data for LotFrontage\_clean and MasVnrArea with the Median

Value\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

proc stdize data=kaggle out=kaggle\_clean method=median missing=median reponly;

var LotFrontage\_clean MasVnrArea;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*A way to check for missing data\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/;

proc contents data=kaggle\_clean out=contents;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*Pulls all the variables names that are categorical\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/;

proc sql;

select name into: variables separated by " " from contents where format ="$";

quit;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Pulls all the variable names that are numerical\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/;

proc sql;

select name into: numerical separated by " " from contents where (format ne"$" and name ne "Id");

quit;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Pulls all variable names except ID and SalePrice\*\*\*\*\*\*\*\*\*\*\*\*\*/;

proc sql;

select name into: modelvariables separated by " " from contents where (name ne "Id" and name ne "SalePrice");

quit;

%put &variables;

/\*\*\*\*\*\*\*\*\*\*\*\*\*Raw Data Set\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Looks at the Frequency count for categorical Variables\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/;

proc freq data=kaggle;

table &variables;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Looks at some of the summary numbers for the numerical variables\*\*\*\*\*\*\*\*\*\*/;

proc means data=kaggle n nmiss mean median std;

var &numerical;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Cleaned Data Set\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/;

/\*\*\*\*\*\*\*Follows the same as above for check\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/;

proc freq data=kaggle\_clean;

table &variables;

run;

proc means data=kaggle\_clean n nmiss mean median std;

var &numerical;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*The forward, backward, and stepwise selection\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/;

proc glmselect data=kaggle\_clean plots(stepaxis = number) = (criterionpanel ASEPlot);

class &variables;

model SalePrice = &modelvariables / selection=forward(stop=CV) cvmethod=random(5) select = sl slentry = .1

stats=adjrsq;

output out = results p = Predict;

run;

proc glm data = kaggle\_clean plots=all;

class &variables;

model SalePrice = OverallQual GrLivArea Neighborhood BsmtQual / solution clparm;

run;

data forward\_model;

set results;

if Predict > 0 then SalePrice = Predict;

if Predict < 0 then SalePrice = 10000;

keep id SalePrice;

where id > 1460;

proc glmselect data=kaggle\_clean ;

class &variables;

model SalePrice = &modelvariables / selection=backward(stop=CV) cvmethod=random(5) select = sl slstay = .1 stb showpvalues

stats=adjrsq;

output out = results1 p = Predict;

run;

data backward\_model;

set results1;

if Predict > 0 then SalePrice = Predict;

if Predict < 0 then SalePrice = 10000;

keep id SalePrice;

where id > 1460;

;

proc glmselect data=kaggle\_clean;

class &variables;

model SalePrice = &modelvariables / selection=stepwise(stop=CV) cvmethod=random(5) select = sl slentry = .1 stb showpvalues

stats=adjrsq;

output out = results2 p = Predict;

run;

data stepwise\_model;

set results2;

if Predict > 0 then SalePrice = Predict;

if Predict < 0 then SalePrice = 10000;

keep id SalePrice;

where id > 1460;

;

proc glmselect data=kaggle\_clean ;

class &variables;

model SalePrice = BedroomAbvGr BsmtFinSF1 BsmtFinSF2 BsmtFullBath BsmtUnfSF Fireplaces FirstFlrSF FullBath GarageArea GarageCars GrLivArea KitchenAbvGr LotArea LotFrontage\_clean LowQualFinSF MSSubClass MasVnrArea MoSold OverallCond OverallQual PoolArea ScreenPorch ThreesnPorch TotRmsAbvGrd WoodDeckSF YearBuilt YearRemodAdd YrSold / selection=backward(stop=CV) cvmethod=random(5) select = sl slstay = .01 stb showpvalues

stats=adjrsq;

output out = results3 p = Predict;

run;

data custom\_model;

set results3;

if Predict > 0 then SalePrice = Predict;

if Predict < 0 then SalePrice = 10000;

keep id SalePrice;

where id > 1460;

;