Data Description

# Analysis Question 1:

## Question of Interest

How does Sale Price vary with Ground Living Area of homes in Iowa? Does neighborhood affect the Sale Price?

## Checking Assumptions

### Original Data

In the original data, constant variance is violated as seen in Code and Plots section “Original Diagnostics” Residual Plot. There are a few points whose residuals are visually far from rest. Furthermore, the quantile plot of residuals show a right skew. From a scatter plot there is a strong positive correlation between GLA and SalePrice.

### Log Transform

A log transform seemed appropriate due to the right skew QQ plot and many outliers from residual plot.

After log transform, qqplot is normally distributed and residual plot has no outliers. But the Cook’s D still shows an observation with potentially high influence.

### Influential Observation Removed

After removing Id 131 which had a Ground Living Area of 4676, Cook’s D of 5, we observe the following diagnostic plots in Code and Plots Section. There is still an observation 339 with Cook’s D of 2.5.

After removing Id 338 which had a Ground Living Area of \_, Cook’s D of 2.5, we observe the following diagnostic plots in Code and Plots Section.

Because we deleted the two observations who had Ground Living Area of 4676 and above , we must limit the conclusions of this report to houses with Ground Living Area below 4676 square feet.

## Parameters

Mean Log SalePrice = B0 (Brk Side) + B1\*GrLivArea + B2\* Names + B3\*Names\*GrLivArea + B4\*Edwards +

B5\*Edwards\*GrLivArea

### Estimates

Mean Log SalePrice = 10.79159+ 0.00073822\*GrLivArea + 0.65175\* Names + -0.00041410\*Names\*GrLivArea + 0.23389\*Edwards + -0.00019957\*Edwards\*GrLivArea

### Interpretation and Confidence Intervals

For houses in BrkSide, every 100 square feet increase in ground living area there will be a 7.7% increase in sale price (95% Confidence Interval: 6.3% to 9%). For houses in Names, 3.3% (95% CI 1.8% 4.8%).

For houses in Edwards, 5.4% (95% CI 3.7% to 7%). Houses with 0 ground living area in BrkSide are expected to have $48,533 Sale Price. Houses in Names are expect to have $1.9 more Sale Price at 0 Ground Living Area. Houses in Edwards are expected to have $1.3 more sale price at 0 ground living area.

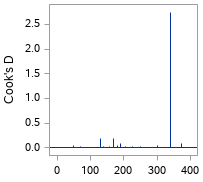
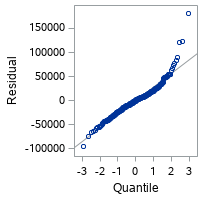
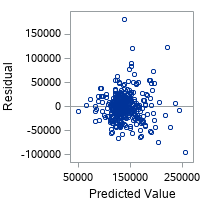
## Conclusion

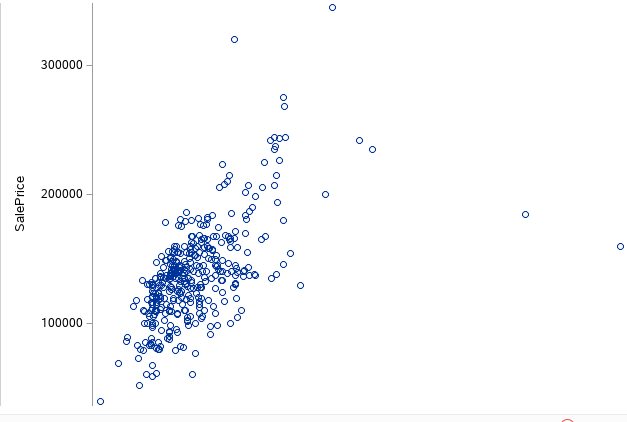
Increasing ground living area will have both a significant and practical effect on Sale Prices. In all 3 neighborhoods, increasing living area by 100 square feet will increase the Sale Price anywhere from 1.8% to 9% depending on the neighborhood. When a typical house sale price is in the $100,000 order of magnitude, then this percentage due to ground living area will be very practical.

Additionally, the increase in Sale Price due to Living Area differs between each neighborhood. Names has the lowest increase in Sale Price for every additional 100 square feet. Brk Side has the highest increase in Sale Price for every additional 100 square feet.

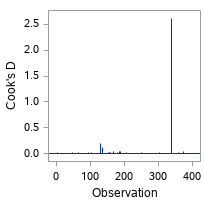
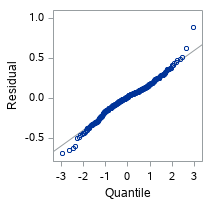
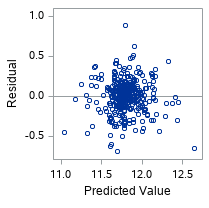
### Code and Plots

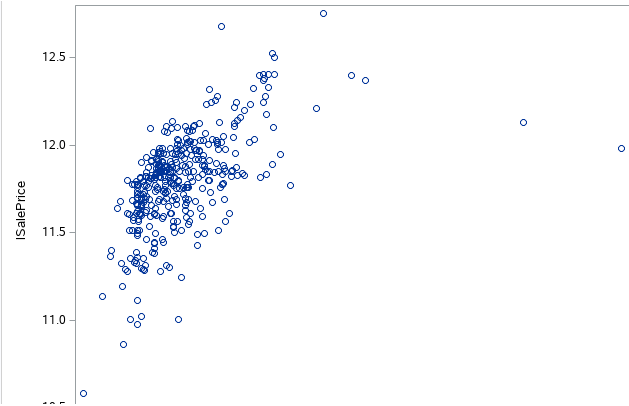
Original Diagnostics





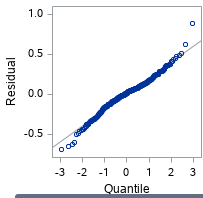
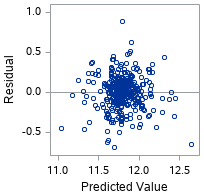
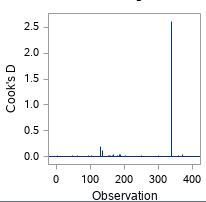
Transformed Diagnostics



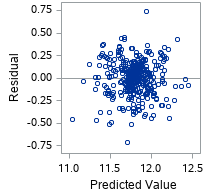
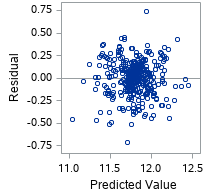
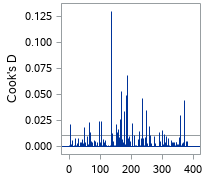


Diagnostics with Removed Influential Points

Observation 131 removed



Observation 131 and 339 removed



# Analysis Question 2

## Restatement of Problem

What is the Sale Price of a house based on any combination and transformation of the 81 attributes of the house?

## Model Selection

### Stepwise

### Checking Assumptions

#### Residual Plots

The residual plot shows a random cloud centered at 0 and constant variance. The residuals based on QQ plot look normally distributed with a few outliers on the left tail.

#### 

There do not seem to be points that have Cook’s D that are substantially higher than others. There are points with high leverage but their Cook’s D are not large.

### Forward

### Checking Assumptions

#### Residual Plots

Residual plots show random cloud centered at 0 with constant variance. Qq plot shows normally distributed residuals.

#### Influential point analysis (Cook’s D and Leverage)

After removing the house with Id of 411 and 1000 we see that there are evenly distributed Cook’s Ds.

### Backward

#### Residual Plots

From the residual plots we observe a random cloud centered at 0 with constant variance. From the qq plot we see that the residuals are normally distributed.

#### Influential point analysis (Cook’s D and Leverage)

There is one observation with a Cook’s D of around 0.125 which is above the 0.0027 threshold for Cook’s D using 4/number of observations. But there are no observations with both high leverage and high studentized residuals so we can leave it alone.

### Custom

## 

## Comparing Competing Models

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Adj R2 | Internal CV Press | Kaggle Score |
| Stepwise | 0.9366 | 18.27958 | 0.1211 |
| Forward | 0.9333 | 17.20477 |  |
| Backward | 0.9162 | 20.60715 |  |
| Custom |  |  |  |

### Conclusion:

The stepwise model seems the best because of its Kaggle Score. It has the most variables which we included because of EDA. The other models have less Adjusted R squared and CV Press.

## Code and Plots

### Data Cleaning

/\*everything is the same as stepwise until the line 111, with proc corr, \*/

/\*DISCLAIMER: The idea to plot residuals against potential explanatory variables

was borrowed from Doing Data Science Week 3 Exploratory Data Analysis by Bivin Sadler \*/

data train;

infile "/folders/myfolders/Kaggle/train.csv" dlm="," firstobs=2;

input Id MSSubClass MSZoning $ LotFrontage LotArea Street $ Alley $ LotShape $ LandContour $ Utilities $ LotConfig $ LandSlope $ Neighborhood $ Condition1 $ Condition2 $ BldgType $ HouseStyle $ OverallQual OverallCond YearBuilt YearRemodAdd RoofStyle $ RoofMatl $ Exterior1st $ Exterior2nd $ MasVnrType $ MasVnrArea ExterQual $ ExterCond $ Foundation $ BsmtQual $ BsmtCond $ BsmtExposure $ BsmtFinType1 $ BsmtFinSF1 BsmtFinType2 $ BsmtFinSF2 BsmtUnfSF TotalBsmtSF Heating $ HeatingQC $ CentralAir $ Electrical $ \_1stflrsf \_2ndflrsf LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr KitchenQual $ TotRmsAbvGrd Functional $ Fireplaces FireplaceQu $ GarageType $ GarageYrBlt GarageFinish $ GarageCars GarageArea GarageQual $ GarageCond $ PavedDrive $ WoodDeckSF OpenPorchSF EnclosedPorch \_3SsnPorch ScreenPorch PoolArea PoolQC $ Fence $ MiscFeature $ MiscVal MoSold YrSold SaleType $ SaleCondition $ SalePrice ;

data test;

infile "/folders/myfolders/Kaggle/test.csv" dlm="," firstobs=2;

input Id MSSubClass MSZoning $ LotFrontage LotArea Street $ Alley $ LotShape $ LandContour $ Utilities $ LotConfig $ LandSlope $ Neighborhood $ Condition1 $ Condition2 $ BldgType $ HouseStyle $ OverallQual OverallCond YearBuilt YearRemodAdd RoofStyle $ RoofMatl $ Exterior1st $ Exterior2nd $ MasVnrType $ MasVnrArea ExterQual $ ExterCond $ Foundation $ BsmtQual $ BsmtCond $ BsmtExposure $ BsmtFinType1 $ BsmtFinSF1 BsmtFinType2 $ BsmtFinSF2 BsmtUnfSF TotalBsmtSF Heating $ HeatingQC $ CentralAir $ Electrical $ \_1stflrsf \_2ndflrsf LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr KitchenQual $ TotRmsAbvGrd Functional $ Fireplaces FireplaceQu $ GarageType $ GarageYrBlt GarageFinish $ GarageCars GarageArea GarageQual $ GarageCond $ PavedDrive $ WoodDeckSF OpenPorchSF EnclosedPorch \_3SsnPorch ScreenPorch PoolArea PoolQC $ Fence $ MiscFeature $ MiscVal MoSold YrSold SaleType $ SaleCondition $ ;

/\*Prepare the train and test sets for prediction\*/

data test;

set test;

SalePrice=.;

;

data trainTest;

set train test;

run;

\*Find the columns with NAs or .;

proc format;

value $missfmt 'NA'='Missing' other='Not Missing';

value missfmt . ='Missing' other='Not Missing';

run;

proc freq data=trainTest;

format \_CHAR\_ $missfmt.; /\* apply format for the duration of this PROC \*/

tables \_CHAR\_ / missing missprint nocum nopercent;

format \_NUMERIC\_ missfmt.;

tables \_NUMERIC\_ / missing missprint nocum nopercent;

run;

/\* If BsmtCond is NA that means there is no basement. So put 0 square feet

for TotalBsmtSquareFeet. If MasVnrType is NA that means no Veneer so put 0 for MasVnrArea.

Etc... \*/

data trainTest;

set trainTest;

if MasVnrType ="NA" AND MasVnrArea =. THEN MasVnrArea=0;

if BsmtCond="NA" AND BsmtFinSF1=. THEN BsmtFinSF1=0;

if BsmtCond="NA" AND BsmtFinSF2=. THEN BsmtFinSF2=0;

if BsmtCond="NA" AND BsmtUnfSF=. THEN BsmtUnfSF=0;

if BsmtCond="NA" AND TotalBsmtSF=. THEN TotalBsmtSF=0;

if BsmtCond="NA" AND BsmtFullBath=. THEN BsmtFullBath=0;

if BsmtCond="NA" AND BsmtHalfBath=. THEN BsmtHalfBath=0;

if PoolQC="NA" AND PoolArea=. THEN PoolArea=0;

if GarageCond="NA" AND GarageArea=. THEN GarageArea=0;

if GarageCond="NA" AND GarageCars=. THEN GarageCars=0;

if FireplaceQu="NA" AND Fireplaces=. THEN Fireplaces=0;

/\*LotFrontage is simply missing values so impute the MEDIAN\*/

proc stdize data=trainTest out=trainTest missing=MEDIAN reponly;

var LotFrontage GarageYrBlt; /\* you can list multiple variables to impute \*/

/\*Find value of highest frequency for categorical variables with missing values\*/

/\*only for variables whose "NA" does not mean "None"\*/

PROC FREQ data=trainTest;

TABLES MSZoning Exterior1st Exterior2nd

Electrical KitchenQual Functional FireplaceQu

MiscFeature SaleType;

data trainTest;

set trainTest;

IF MSZoning = "NA" THEN MSZoning = "RL";

IF Utilities = "NA" THEN Utilities = "AllPub";

IF Exterior1st = "NA" THEN Exterior1st = "VinylSd";

IF Exterior2nd = "NA" THEN Exterior2nd = "VinylSd";

IF MasVnrType = "NA" THEN MasVnrType = "None";

IF Electrical = "NA" THEN Electrical = "SBrkr";

IF KitchenQual = "NA" THEN KitchenQual = "TA";

IF Functional = "NA" THEN Functional = "Typ";

IF SaleType = "NA" THEN SaleType = "WD";

/\*The only values with missing variables should be Alley MasVnrType BsmtQual BsmtExposure BsmtFintype

GarageTypeQualFinishCond PoolQC FireplaceQu Fence\*/

proc freq data=trainTest;

format \_CHAR\_ $missfmt.; /\* apply format for the duration of this PROC \*/

tables \_CHAR\_ / missing missprint nocum nopercent;

format \_NUMERIC\_ missfmt.;

tables \_NUMERIC\_ / missing missprint nocum nopercent;

run;

data trainTest;

set trainTest;

lSalePrice=log(SalePrice);

\*From question 1 there were influential points for GrLivArea;

data trainTest;

set trainTest;

if \_n\_ = 1299 then delete;

data trainTest;

set trainTest;

if \_n\_ = 524 then delete;

### Question 1

proc glm data=traintest;

where Neighborhood in ("NAmes","Edwards","BrkSide");

class Neighborhood;

model lsaleprice=grlivarea neighborhood;

### Data Transformation

\*Log transform \_1stflrsf;

data trainTest;

set trainTest;

if \_1stflrsf = 0 then l\_1stflrsf=log(\_1stflrsf+0.1);

else l\_1stflrsf=log(\_1stflrsf);

data traintest;

set traintest;

lgrlivarea=log(grlivarea);

llotarea=log(lotarea);

llotfrontage=log(lotfrontage);

if TotalBsmtsf=0 then lTotalBsmtSF=log(TotalBsmtSF+100);

else lTotalBsmtSF=log(totalbsmtsf);

if bsmtfinsf1=0 then lbsmtfinsf1=log(bsmtfinsf1+100);

else lbsmtfinsf1=log(bsmtfinsf1);

if openporchsf=0 then lopenporchsf=log(openporchsf+10);

else lopenporchsf=log(openporchsf);

totalArea=screenporch + grlivarea;

ltotalArea=log(totalArea);

totalrooms=TotRmsAbvGrd + bsmtfullbath + bsmthalfbath;

totalporch=openporchsf + enclosedporch +screenporch+\_3SsnPorch;

baths=bsmtfullbath+(bsmthalfbath/2)+fullbath+(halfbath/2);

/\*Recode ordinal variables as numbers\*/

proc sql;

create table traintestr as

select \*, case ExterQual

when "Ex" then 5

when "Gd" then 4

when "TA" then 3

when "Fa" then 2

else 1

end as ExterQualR,

case ExterCond

when "Ex" then 5

when "Gd" then 4

when "TA" then 3

when "Fa" then 2

else 1

end as ExterCondR,

case BsmtQual

when "Ex" then 5

when "Gd" then 4

when "TA" then 3

when "Fa" then 2

when "Po" then 1

else 0

end as BsmtQualR,

case BsmtCond

when "Ex" then 5

when "Gd" then 4

when "TA" then 3

when "Fa" then 2

when "Po" then 1

else 0

end as BsmtCondR,

case BsmtExposure

when "Gd" then 4

when "Av" then 3

when "Mn" then 2

when "No" then 1

else 0

end as BsmtExposureR,

case BsmtFinType1

when "GLQ" then 6

when "ALQ" then 5

when "BLQ" then 4

when "Rec" then 3

when "LwQ" then 2

when "Unf" then 1

else 0

end as BsmtFinType1R,

case BsmtFinType2

when "GLQ" then 6

when "ALQ" then 5

when "BLQ" then 4

when "Rec" then 3

when "LwQ" then 2

when "Unf" then 1

else 0

end as BsmtFinType2R,

case HeatingQC

when "Ex" then 5

when "Gd" then 4

when "TA" then 3

when "Fa" then 2

else 1

end as HeatingQCR,

case KitchenQual

when "Ex" then 5

when "Gd" then 4

when "TA" then 3

when "Fa" then 2

else 1

end as KitchenQualR,

case FireplaceQu

when "Ex" then 5

when "Gd" then 4

when "TA" then 3

when "Fa" then 2

when "Po" then 1

else 0

end as FireplaceQuR,

case Functional

when "Typ" then 8

when "Min1" then 7

when "Min2" then 6

when "Mod" then 5

when "Maj1" then 4

when "Maj2" then 3

when "Sev" then 2

else 1

end as FunctionalR,

case GarageFinish

when "Fin" then 3

when "RFn" then 2

when "Unf" then 1

else 0

end as GarageFinishR,

case GarageQual

when "Ex" then 5

when "Gd" then 4

when "TA" then 3

when "Fa" then 2

when "Po" then 1

else 0

end as GarageQualR,

case GarageCond

when "Ex" then 5

when "Gd" then 4

when "TA" then 3

when "Fa" then 2

when "Po" then 1

else 0

end as GarageCondR,

case PoolQC

when "Ex" then 4

when "Gd" then 3

when "TA" then 2

when "Fa" then 1

else 0

end as PoolQCR,

case Fence

when "GdPrv" then 4

when "MnPrv" then 3

when "GdWo" then 2

when "MnWw" then 1

else 0

end as FenceR

from traintest

;

quit;

data traintestsd;

set traintest;

/\*Reduce collinearity between year variables\*/

proc stdize data=traintest out=traintestsd method=std;

var yearbuilt yearremodadd ;

\*9232;

data traintest;

set traintest;

if bsmtunfsf=0 then lbsmtunfsf= log(bsmtunfsf+10);

else lbsmtunfsf=log(bsmtunfsf);

if bsmtfinsf2=0 then lbsmtfinsf2= log(bsmtfinsf2+5);

else lbsmtfinsf2=log(bsmtfinsf2);

if \_3SsnPorch=0 then l\_3SsnPorch= log(\_3SsnPorch+15);

else l\_3SsnPorch=log(\_3SsnPorch);

if miscval=0 then lmiscval= log(miscval+50);

else lmiscval=log(miscval);

if miscval=0 then lmssubclass= log(mssubclass+50);

else lmssubclass=log(mssubclass);

if wooddecksf=0 then lwooddecksf= log(wooddecksf+3);

else lwooddecksf=log(wooddecksf);

l\_1stflrsf=log(\_1stflrsf);

if \_2ndflrsf=0 then l\_2ndflrsf=log(\_2ndflrsf+100);

else l\_2ndflrsf=log(\_2ndflrsf);

if enclosedporch=0 then lenclosedporch=log(enclosedporch+10);

else lenclosedporch=log(enclosedporch);

if poolarea=0 then lpoolarea=log(poolarea+100);

else lpoolarea=log(poolarea);

if lowqualfinsf=0 then llowqualfinsf=log(lowqualfinsf+40);

else llowqualfinsf=log(lowqualfinsf);

proc stdize data=traintestr out=traintests method=std;

var \_1stflrsf \_2ndflrsf;

### Stepwise Model

#### Kaggle Submission

proc glm data=traintests plots=all;

where id ne 496;

class centralair neighborhood mszoning salecondition kitchenqual bsmtexposure bsmtqual

garagequal;

model lSalePrice=llotarea overallqual overallcond yearbuilt BsmtFinSF1

lgrlivarea totalbsmtsf bedroomabvgr Fireplaces GarageCars totalporch

\_2ndflrsf \_1stflrsf

centralair neighborhood mszoning salecondition kitchenqual bsmtqual

bsmtexposure garagequal

/solution;

output out=results p=Predict;

run;

\*transform log numbers into normal numbers;

data results;

set results;

Predict=exp(Predict);

SalePrice=Predict;

\*if sale price is below 0 change it;

data results2;

set results;

if Predict>0 then SalePrice = Predict;

if Predict < 0 then SalePrice=10000;

keep id SalePrice;

where id>1460;

#### Variable Selection

proc glmselect data=trainTest seed=1;

class MSZoning Street Alley LotShape LandContour Utilities LotConfig LandSlope

Neighborhood Condition1 Condition2 BldgType HouseStyle RoofStyle RoofMatl

Exterior1st Exterior2nd MasVnrType ExterQual ExterCond Foundation BsmtQual

BsmtCond BsmtExposure BsmtFinType1 BsmtFinType2 Heating HeatingQC CentralAir

Electrical KitchenQual Functional FireplaceQu GarageType GarageFinish

GarageQual GarageCond PavedDrive Fence MiscFeature SaleType SaleCondition PoolQC;

model lSalePrice=MSSubClass--SaleCondition / selection=stepwise;

/\* Variable Selection returns these variables:

mszoning

lotarea

bldgtype !

overallqual

overallcond

yearbuilt

yearremodadd !

bsmtfinsf1

totalbsmtsf

centralair

grlivarea

bsmtfullbath

bedroomabvgr

kitchenqual

functional

fireplaces

garagecars

wooddecksf !!!

salecondition

\*/

PROC CORR DATA=traintest out=corrs;

VAR

lsaleprice

lotarea

overallqual

overallcond

yearbuilt

yearremodadd

bsmtfinsf1

totalbsmtsf

grlivarea

bsmtfullbath

bedroomabvgr

fireplaces

garagecars

wooddecksf

;

/\*Year remodadd removed due to high correlation with yearbuilt\*/

/\*Chose the 5 least correlated variables and determine who is leaving\*/

proc sgscatter data=traintest;

matrix lsaleprice wooddecksf overallcond LlotArea bsmtfullbath BedroomAbvGr;

/\*wooddecksf too many 0s and cannot log transform, bsmtfullbath no visual correlation

Both are removed\*/

#### Exploratory Data Analysis

/\*

Read in all variables as numerical or factor and their number. This is so SAS can loop through each variable and make a plot. The next 2 pages of code are identical to data cleaning section.

\*/

### Numbering Columns, Cleaning Again for Automated EDA

data trainNum;

infile "/folders/myfolders/Kaggle/train.csv" dlm="," firstobs=2;

input num1 num2 fac1 $ num3 num4 fac2 $ fac3 $ fac4 $ fac5 $ fac6 $ fac7 $ fac8 $ fac9 $ fac10 $ fac11 $ fac12 $ fac13 $ num5 num6 num7 num8 fac14 $ fac15 $ fac16 $ fac17 $ fac18 $ num9 fac19 $ fac20 $ fac21 $ fac22 $ fac23 $ fac24 $ fac25 $ num10 fac26 $ num11 num12 num13 fac27 $ fac28 $ fac29 $ fac30 $ num14 num15 num16 num17 num18 num19 num20 num21 num22 num23 fac31 $ num24 fac32 $ num25 fac33 $ fac34 $ num26 fac35 $ num27 num28 fac36 $ fac37 $ fac38 $ num29 num30 num31 num32 num33 num34 fac39 $ fac40 $ fac41 $ num35 num36 num37 fac42 $ fac43 $ SalePrice ;

data testNum;

infile "/folders/myfolders/Kaggle/test.csv" dlm="," firstobs=2;

input num1 num2 fac1 $ num3 num4 fac2 $ fac3 $ fac4 $ fac5 $ fac6 $ fac7 $ fac8 $ fac9 $ fac10 $ fac11 $ fac12 $ fac13 $ num5 num6 num7 num8 fac14 $ fac15 $ fac16 $ fac17 $ fac18 $ num9 fac19 $ fac20 $ fac21 $ fac22 $ fac23 $ fac24 $ fac25 $ num10 fac26 $ num11 num12 num13 fac27 $ fac28 $ fac29 $ fac30 $ num14 num15 num16 num17 num18 num19 num20 num21 num22 num23 fac31 $ num24 fac32 $ num25 fac33 $ fac34 $ num26 fac35 $ num27 num28 fac36 $ fac37 $ fac38 $ num29 num30 num31 num32 num33 num34 fac39 $ fac40 $ fac41 $ num35 num36 num37 fac42 $ fac43 $ ;

\*Vertically merge numbered columns;

data trainTestNum;

set trainNum testNum;

run;

/\*Data cleaning same as unnumbered SASS

See columnNames.xlsx to match up factor and number variables \*/

data trainTestNum;

set trainTestNum;

if fac18 ="NA" AND num9 =. THEN num9=0;

if fac23="NA" AND num10=. THEN num10=0;

if fac23="NA" AND num11=. THEN num11=0;

if fac23="NA" AND num12=. THEN num12=0;

if fac23="NA" AND num13=. THEN num13=0;

if fac23="NA" AND num18=. THEN num18=0;

if fac23="NA" AND num19=. THEN num19=0;

if fac39="NA" AND num34=. THEN num34=0;

if fac37="NA" AND num28=. THEN num28=0;

if fac37="NA" AND num27=. THEN num27=0;

if fac33="NA" AND num25=. THEN num25=0;

/\*LotFrontage is simply missing values so impute the MEDIAN\*/

proc stdize data=trainTestNum out=trainTestNum missing=MEDIAN reponly;

var num3 num26; /\* you can list multiple variables to impute \*/

data trainTestNum;

set trainTestNum;

IF fac1 = "NA" THEN fac1 = "RL";

IF fac6 = "NA" THEN fac6 = "AllPub";

IF fac16 = "NA" THEN fac16 = "VinylSd";

IF fac17 = "NA" THEN fac17 = "VinylSd";

IF fac18 = "NA" THEN fac18 = "None";

IF fac30 = "NA" THEN fac30 = "SBrkr";

IF fac31 = "NA" THEN fac31 = "TA";

IF fac32 = "NA" THEN fac32 = "Typ";

IF fac42 = "NA" THEN fac42 = "WD";

/\*double check missing\*/

data trainTestNum;

set trainTestNum;

lSalePrice=log(SalePrice);

data trainTestNum;

set trainTestNum;

if \_n\_ = 1299 then delete;

data trainTestNum;

set trainTestNum;

if \_n\_ = 524 then delete;

/\*log transform all numerical explanatory variables\*/

%macro logAll();

%do n=1 %to 37;

data trainTestNum;

set trainTestNum;

if num&n = 0 then lnum&n=log(num&n+0.1);

else lnum&n=log(num&n);

%end;

%mend;

%logAll();

data traintestnum;

set traintestnum;

num38=num18+(num19/2)+num20+(num21/2);

num39=num30+num31+num33+num32;

data traintestnumr;

set traintestnum;

proc stdize data=traintestnumr out=traintestnums method=std;

var num15 num14 ;

#### EDA Automated

%macro alldotplot();

%do i=1 %to 43;

proc sgplot data=traintestnums;

scatter x=fac&i y=lsaleprice/ jitter transparency=0.5 jitterwidth=0.5;

%end;

%mend;

%alldotplot();

\*Fac12 which is BldgType does not have much difference in lSaleprice between levels so remove;

\*Dotplots of Garagequal and Bsmtqual look good so add them to model;

/\*Look at the other continuous variables to add to plot\*/

proc sgscatter data=traintest;

matrix lsaleprice mssubclass lotfrontage

masvnrarea fullbath;

proc sgscatter data=traintest;

matrix lsaleprice garageyrblt

wooddecksf openporchsf;

proc sgscatter data=traintest;

matrix lsaleprice openporchsf

enclosedporch

\_3ssnporch

screenporch

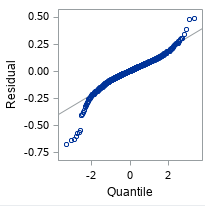
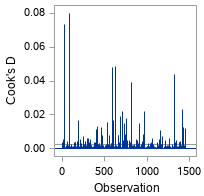
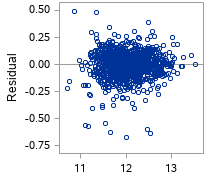
;

/\*THe porch variables as a group looked correlated so a totalporch variable (sum of porch areas) was created in the “DATA TRANSFORMATION” section \*/

\*Also added neighborhood because felt it was important;

#### /\*Final Stepwise model\*/

proc glm data=traintests plots=all;  
where id ne 496;  
class centralair neighborhood mszoning salecondition kitchenqual bsmtexposure bsmtqual   
garagequal;  
model lSalePrice=llotarea overallqual overallcond yearbuilt BsmtFinSF1  
 lgrlivarea totalbsmtsf bedroomabvgr Fireplaces GarageCars totalporch  
 \_2ndflrsf \_1stflrsf   
 centralair neighborhood mszoning salecondition kitchenqual bsmtqual  
 bsmtexposure garagequal  
 /solution;



Residuals appear random cloud equivariance around 0, no Cook’s D too different from rest, and

The quantiles show normally distributed residuals.

### Forward

/\* trainTest is dataset\*/

/\*Finding best subset of varialbes\*/

/\*\*/

proc glmselect data=trainTest plots=ALL;

class MSZoning Street Alley LotShape LandContour Utilities LotConfig LandSlope Neighborhood Condition1 Condition2 BldgType HouseStyle RoofStyle RoofMatl Exterior1st Exterior2nd MasVnrType ExterQual ExterCond Foundation BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinType2 Heating HeatingQC CentralAir Electrical KitchenQual Functional FireplaceQu GarageType GarageFinish

GarageQual GarageCond PavedDrive PoolQC Fence MiscFeature SaleType SaleCondition;

model lSalePrice=MSSubClass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF \_1stFlrSF \_2ndflrsf LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces GarageYrBlt GarageCars GarageArea WoodDeckSF OpenPorchSF EnclosedPorch \_3SsnPorch ScreenPorch PoolArea MiscVal MoSold YrSold MSZoning Street Alley LotShape LandContour Utilities LotConfig LandSlope Neighborhood Condition1 Condition2 BldgType HouseStyle RoofStyle RoofMatl Exterior1st Exterior2nd MasVnrType ExterQual ExterCond Foundation BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinType2 Heating HeatingQC CentralAir Electrical

KitchenQual Functional FireplaceQu GarageType GarageFinish GarageQual GarageCond PavedDrive PoolQC Fence MiscFeature SaleType SaleCondition/selection=forward;

/\*dummy coding variables to use proc reg\*/

/\*Removing Houses with id 411 and 1000 because they are influential\*/

proc glmmod data=trainTest outdesign=GLMDesign outparm=GLMParm NOPRINT;

where Id not in (411,1000);

class Mszoning LotConfig

Neighborhood Condition1 Exterior1st BsmtExposure HeatingQC CentralAir KitchenQual Functional GarageQual PoolQC SaleCondition ;

model lSalePrice=MSSubclass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd BsmtFinSF1 TotalBsmtSF GrLivArea BsmtFullBath FullBath HalfBath Fireplaces GarageCars WoodDeckSF OpenPorchSF EnclosedPorch ScreenPorch Mszoning LotConfig Neighborhood Condition1 Exterior1st

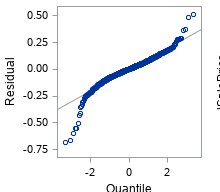
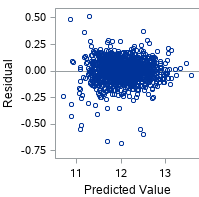
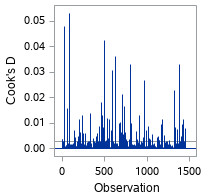
BsmtExposure HeatingQC CentralAir KitchenQual Functional GarageQual PoolQC SaleCondition;

/\*creating regression model from dummy coded varialbes\*/

proc reg data=GLMDesign;

DummyVars: model lSalePrice = COL2-COL118

/influence;



### Backward

\*Select best subset of variables from all;

/\*cleanTrainTranInf is the dataset after transformation and influence observation removal\*/

proc glmselect data=trainTest seed=1;

class MSZoning Street Alley LotShape LandContour Utilities LotConfig LandSlope

Neighborhood Condition1 Condition2 BldgType HouseStyle RoofStyle RoofMatl

Exterior1st Exterior2nd MasVnrType ExterQual ExterCond Foundation BsmtQual

BsmtCond BsmtExposure BsmtFinType1 BsmtFinType2 Heating HeatingQC CentralAir

Electrical KitchenQual Functional FireplaceQu GarageType GarageFinish

GarageQual GarageCond PavedDrive Fence MiscFeature SaleType SaleCondition PoolQC;

model lSalePrice=MSSubClass--SaleCondition / selection=backward;

/\*Dummy code the best categorical variables\*/

proc glmmod data=trainTest outdesign=GLMDesign outparm=GLMParm NOPRINT;

where Id ne 411;

class Mszoning Foundation CentralAir KitchenQual Functional SaleCondition ;

model lSalePrice= OverallQual OverallCond YearBuilt YearRemodAdd BsmtFinSF1 BsmtFinSF2

BsmtUnfSF \_1stFlrSF \_2ndflrsf LowQualFinSF Fireplaces GarageCars WoodDeckSF ScreenPorch Mszoning Foundation CentralAir KitchenQual Functional SaleCondition;

/\*Build model using the dummy coded categorical variables and best numerical variables\*/

proc reg data=GLMDesign;

DummyVars: model lSalePrice = COL2-COL45

/influence;

