Stat 521 Predictive Modeling Final Project Writeup

```
## Warning: package 'corrplot' was built under R version 3.3.3
```

```
## Warning: package 'forcats' was built under R version 3.3.3
```

```
## Warning: package 'glmnet' was built under R version 3.3.3
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
check = function(pred, true_value){
  return(data.frame(RMSE = RMSE(pred[,1],true_value),
                    BIAS = BIAS(pred[,1],true_value),
                    maxDeviation = maxDeviation(pred[,1],true_value),
                    MeanAbsDeviation = MeanAbsDeviation(pred[,1],true_value),
                    Coverage = coverage(pred[,2], pred[,3], true_value)))
}
RMSE = function(y,pred) {
  rmse = sqrt(mean((y-pred)^2))
  return(rmse)
}

BIAS = function(pred, true_value){
  return(mean(pred-true_value))
}
maxDeviation = function(pred, true_value){
  return(max(abs(pred-true_value)))
}
MeanAbsDeviation = function(pred, true_value){
  return(mean(abs(pred-true_value)))
}
coverage = function(lwr,upr,true_value){
  mean(lwr<true_value & true_value<upr)
}
```

After cleaning and processing data carefully, we analyzed the distribution characteristics of variables and developed multiple models to assess how well they fit the data.

Data Cleaing and Processing

We put the 3 data sets (training, testing and validation) together for data cleaning. For all variables with more than half observations "NA", we delete the variable. For the remaining variables:

- **Categorical variables**

Without NA: no changes

With NA: add a new level "NA"

- **Nominal variables**

Without NA: convert to numeric variables in the ascending order

With NA: add a new level "NA", and convert to numeric variables in the ascending order with "NA" lowest.

"": delete the observation(s) (Garage.Finish), or convert to most popular values (Basement Exposure).

- **Discrete variables**

Without NA: no changes

With NA: delete the variable (year), change to 0 (Bsmt Full Bath, Bsmt Half Bath)

- **Continuous variables**

Without NA: no changes

With NA: change to 0 (lot.Frontage),

Exploratory data analysis

After the data cleaning, there are 74 variables in the data set, and 1499 observations in the training set, 500 observations in the testing set, and 413 observations in the validation set.

- **Dependent variable: price**

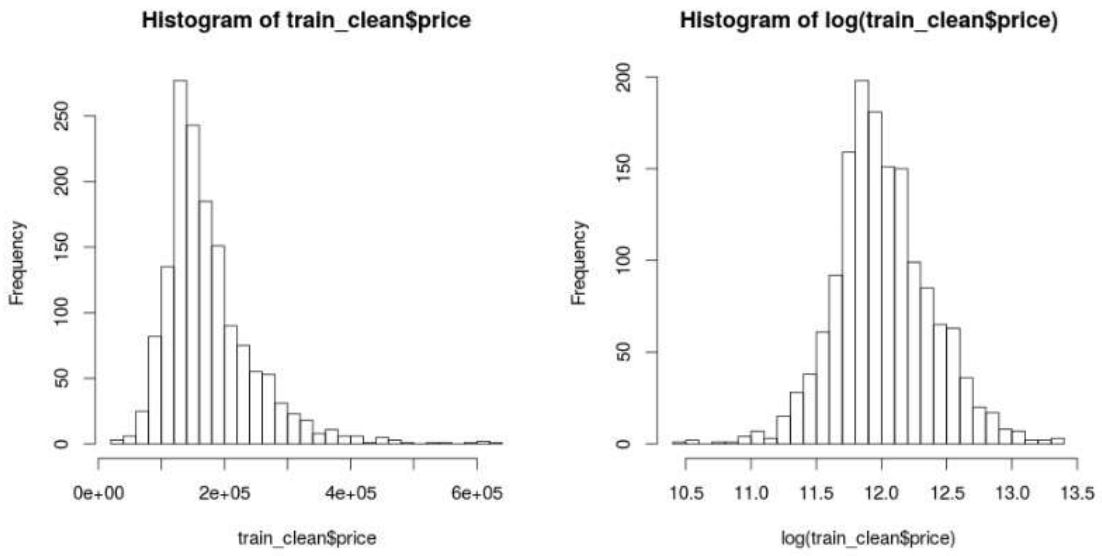


Figure 1: Distribution of the Dependent Variable

From the distribution histogram of the variable `price` in the training data in Figure 1, we can see its original values are highly skewed below its mean. In order to obtain a more normal distribution to fit the requirement of most models, we take log transformation of the variable, and from the histogram plot we can see that its distribution is more symmetric and less heavy-tailed.

- **Correlation between continuous variables**

From the correlation plot in Figure 2, we can see that `price` is highly correlated with `area`, `Overall.Qual`, `TotalSq`, `Year.Built`, `Year.Remod.Add`, `Exter.Qual`, `Bsmt.Qual`, `Total.Bsmt.SF`, `X1stFlr.SF`, `Full.Bath`, `Kitchen.Qual`, `Fireplaces`, `Garage.Finish`, `Garage.Cars` and `Garage.Area`. In this way, in the model developing period, we can pay attention to these variables and consider a set of possible explanatory variables out of them.

Besides the relationship between the explanatory variables and the dependent variable, there are some evident possible correlations among explanatory variables, i.e., `area` and `TotalSq`, `Lot.Frontage` and `Bsmt.Full.Bath`, `Lot.Frontage` and `Bsmt.Half.Bath`, `Bsmt.Full.Bath` and `Bsmt.Half.Bath`, `Garage.Cond` and `Garage.Qual`, `Garage.Cars` and `Garage.Area`, `BsmtFin.SF.2` and `BsmtFin.Type.2`. With strong correlation between these variables, when some variable pair occur together in a model, we should check that there is no potential problems of multicollinearity.

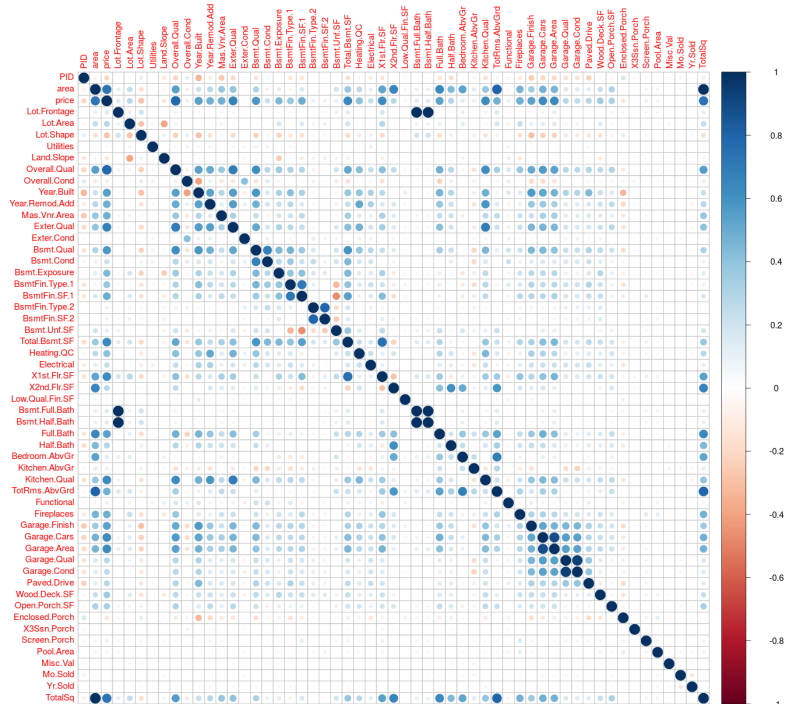


Figure 2: Correlation between Continuous Variables

- **Measurement of variable importance from different models**

We employed random forest and boosting to obtain a more reliable measurement of variable importance. In both rough models, we used the processed training data and all variables. From the following Figure 3 and Table 1, we can see that from both measures, `Overall.Qual`, `Neighborhood`, `TotalSq`, `area`, `Total.Bsmt.SF`, `Garage.Area`, `Exter.Qual`, `Garage.Cars`, `Bsmt.Qual`, `BsmtFin.SF.1`, `Kitchen.Qual`, `X1st.Flr.SF`, `MS.SubClass` and `Lot.Area` are all variables which are considered as top 10 important in each model.

Variable	Overall.Qual	Neighborhood	TotalSq	area	Total.Bsmt.SF
Relative Influence	27.5692	17.3513	11.8593	6.2747	5.1457
Variable	Garage.Area	BsmtFin.SF.1	Kitchen.Qual	X1st.Flr.SF	MS.SubClass
Relative Influence	3.6117	2.7965	2.3262	2.3201	2.0250

Table 1: Variable Importance from Boosting (top 10)

Development and assessment of an initial model from Part I

- Initial model

The simple model with the best performance in RMSE is the OLS model with BIC and common sense variable selection.

The summary of the final simple model is listed as following. The adjusted R-square of the model is 0.9443.

```
##
## Call:
## lm(formula = log(price) ~ area + log(Lot.Area + 1) + Neighborhood +
##     Bldg.Type + Overall.Qual + Overall.Cond + Year.Built + Bsmt.Exposure +
```

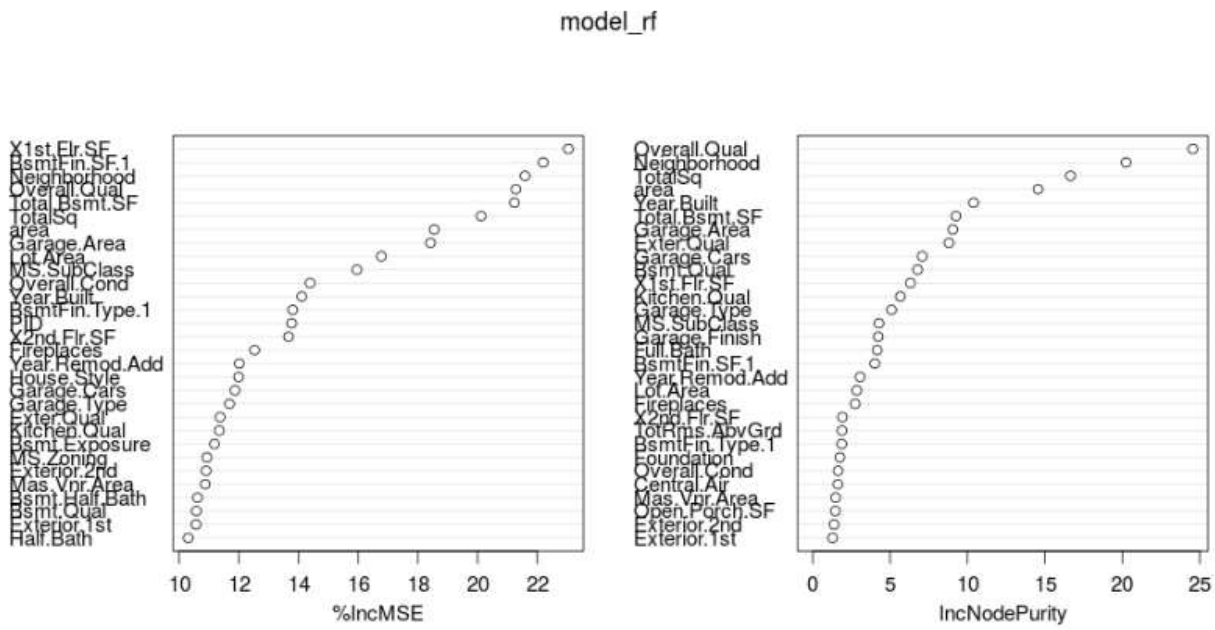


Figure 3: Variable Importance from Random Forest (top 10)

```
##     BsmtFin.SF.1 + BsmtFin.SF.2 + Bsmt.Unf.SF + Central.Air +
##     Kitchen.Qual + Functional + Fireplaces + Garage.Cars + Paved.Drive +
##     Open.Porch.SF, data = train_clean[-c(168, 183, 462), ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37255 -0.04959  0.00243  0.05226  0.29792
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3.925e+00  3.962e-01   9.908  < 2e-16 ***
## area                  2.673e-04  7.467e-06  35.800  < 2e-16 ***
## log(Lot.Area + 1)     9.137e-02  7.913e-03  11.548  < 2e-16 ***
## NeighborhoodBlueste -1.995e-02  3.946e-02  -0.506  0.61327
## NeighborhoodBrDale  -9.674e-02  3.356e-02  -2.882  0.00401 **
## NeighborhoodBrkSide -2.217e-02  3.124e-02  -0.710  0.47797
## NeighborhoodClearCr  1.592e-02  3.320e-02   0.479  0.63171
## NeighborhoodCollgCr -5.563e-02  2.766e-02  -2.011  0.04447 *
## NeighborhoodCrawfor  7.378e-02  3.096e-02   2.383  0.01731 *
## NeighborhoodEdwards -9.703e-02  2.957e-02  -3.281  0.00106 **
## NeighborhoodGilbert -5.713e-02  2.856e-02  -2.000  0.04567 *
## NeighborhoodGreens   1.031e-02  4.201e-02   0.245  0.80616
## NeighborhoodGrnHill  4.264e-01  6.696e-02   6.368 2.57e-10 ***
## NeighborhoodIDOTRR  -1.019e-01  3.279e-02  -3.108  0.00192 **
## NeighborhoodLandmrk -5.254e-02  9.173e-02  -0.573  0.56692
## NeighborhoodMeadowV -1.507e-01  3.753e-02  -4.016 6.22e-05 ***
## NeighborhoodMitchel -6.501e-02  2.927e-02  -2.221  0.02654 *
## NeighborhoodNAmes   -7.803e-02  2.842e-02  -2.745  0.00612 **
## NeighborhoodNoRidge -7.332e-03  3.034e-02  -0.242  0.80906
```

```
## NeighborhoodNPkVill -2.767e-02  3.678e-02  -0.752  0.45208
## NeighborhoodNridgHt  3.120e-02  2.806e-02   1.112  0.26635
## NeighborhoodNWAmes  -9.035e-02  2.912e-02  -3.103  0.00195 **
## NeighborhoodOldTown -8.149e-02  3.072e-02  -2.653  0.00807 **
## NeighborhoodSawyer  -6.876e-02  2.959e-02  -2.324  0.02027 *
## NeighborhoodSawyerW -7.828e-02  2.891e-02  -2.708  0.00684 **
## NeighborhoodSomerst  3.777e-02  2.718e-02   1.390  0.16488
## NeighborhoodStoneBr  5.812e-02  3.038e-02   1.913  0.05595 .
## NeighborhoodSWISU   -5.460e-02  3.425e-02  -1.594  0.11114
## NeighborhoodTimber  -4.815e-02  3.115e-02  -1.546  0.12233
## NeighborhoodVeenker -4.299e-02  3.635e-02  -1.183  0.23717
## Bldg.Type2fmCon     -1.937e-02  1.767e-02  -1.096  0.27306
## Bldg.TypeDuplex     -7.787e-02  1.385e-02  -5.621 2.28e-08 ***
## Bldg.TypeTwnhs      -3.790e-02  1.902e-02  -1.993  0.04647 *
## Bldg.TypeTwnhsE     -9.711e-03  1.336e-02  -0.727  0.46747
## Overall.Qual         5.589e-02  3.128e-03  17.865  < 2e-16 ***
## Overall.Cond         4.096e-02  2.485e-03  16.482  < 2e-16 ***
## Year.Built           2.877e-03  1.942e-04  14.812  < 2e-16 ***
## Bsmt.Exposure        1.121e-02  2.724e-03   4.116 4.06e-05 ***
## BsmtFin.SF.1         1.823e-04  8.682e-06  20.999  < 2e-16 ***
## BsmtFin.SF.2         1.367e-04  1.427e-05   9.581  < 2e-16 ***
## Bsmt.Unf.SF          8.529e-05  8.239e-06  10.353  < 2e-16 ***
## Central.AirY         5.653e-02  1.113e-02   5.079 4.29e-07 ***
## Kitchen.Qual         3.870e-02  5.101e-03   7.586 5.86e-14 ***
## Functional           2.603e-02  3.754e-03   6.933 6.20e-12 ***
## Fireplaces           2.328e-02  4.413e-03   5.277 1.52e-07 ***
## Garage.Cars          3.726e-02  4.450e-03   8.374  < 2e-16 ***
## Paved.Drive          1.594e-02  5.255e-03   3.033  0.00246 **
## Open.Porch.SF        1.236e-04  3.943e-05   3.134  0.00176 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08685 on 1448 degrees of freedom
## Multiple R-squared:  0.9461, Adjusted R-squared:  0.9443
## F-statistic: 540.5 on 47 and 1448 DF,  p-value: < 2.2e-16
```

From the estimated coefficients and their significance levels, we can see that among continous variables, `area`, `log(Lot.Area + 1)`, `BsmtFin.SF.1`, `BsmtFin.SF.2`, `Bsmt.Unf.SF` and `Open.Porch.SF` are significantly positive. This means that when one of these variables increases, the sale price of the house increases as well.

Among ordinal variables, `Overall.Qual`, `Overall.Cond`, `Bsmt.Exposure`, `Kitchen.Qual`, `Functional`, `Fireplaces`, `Garage.Cars` and `Paved.Drive` are significantly positive. This means that for these variables, when their levels get higher, the sale price of the house increases as well.

Among categorical variables, for `Neighborhood`, some neighborhood areas have significantly lower house prices, some areas have significantly higher house prices, which some areas are not so affective to house prices. For `Bldg.Type`, sale prices of houses of types "duplex" and "townhouse end unit" are significantly lower than those of houses of the type "single-family detached". For `Central.Air`, sales prices of houses with central air are significantly higher than those of houses without central air.

Considering the mechanism of the housing market, we consider the predictive results on these variables reasonable and close to reality.

- Model selection

To obtain the model, we have tried among OLS, OLS and BIC, random forest, boosting, ridge and lasso

models. By comparing the performance in terms of RMSE on both the training and testing data, we find that the OLS and BIC model is the best one, with smallest discrepency between two data sets, and smallest RMSE on the testing data too.

We first put all explanatory variables into the linear model. During this step, we find out from the diagnostic plots that there are several outliers with extreme values (index 168, 183, 462 in the training set) and we delete them to fit the same model again.

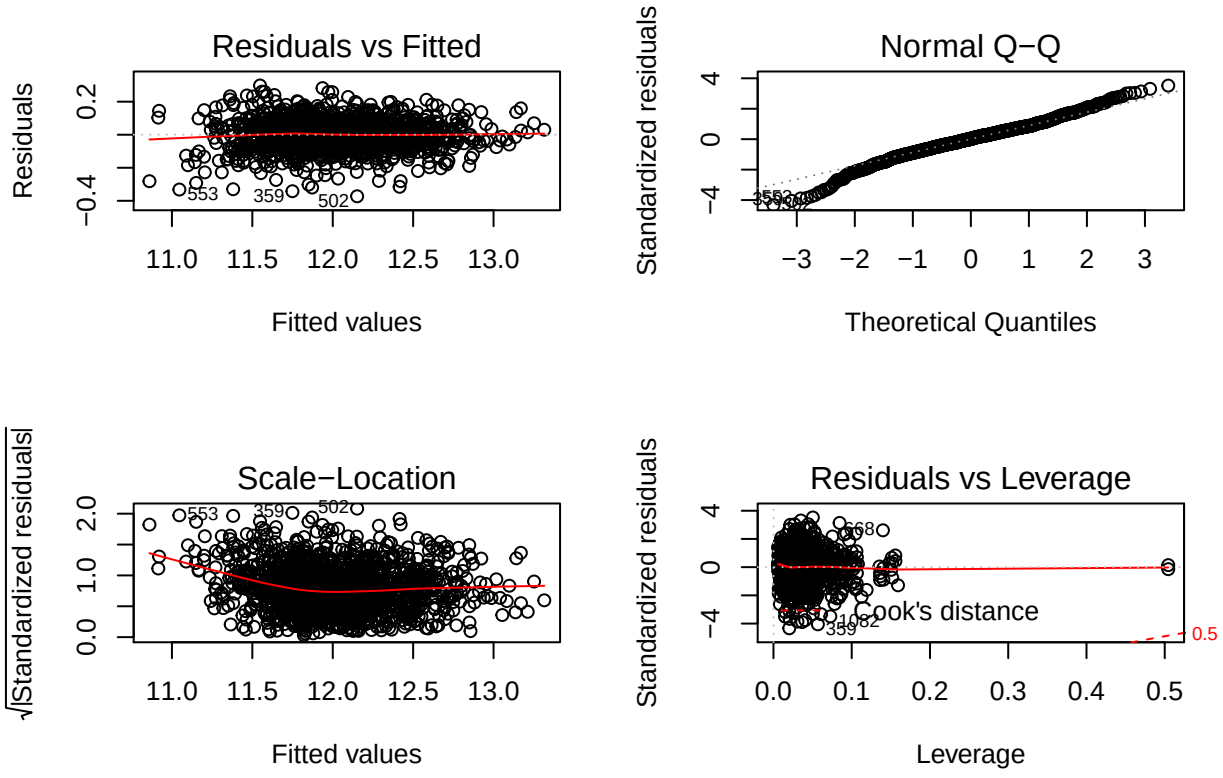
Then we do BIC variable selection on the output model. There are 22 variables (except for the intercept) in the selected model with minimal BIC value.

From the termplot of the minimal-BIC ols model, we detect that some variables (`Lot.Area`, `Enclosed.Porch`, and `Screen.Porch`) need log transformations to better fit the data trend. And after taking the meanings and correlations of variables into consideration, we decide to remove 4 variable (`Year.Remod.Add`, `Garage.Area`, `log(Enclosed.Porch+1)` and `log(Screen.Porch + 1)`) from the model.

- Residual

```
## Warning: not plotting observations with leverage one:
##   988
```

```
## Warning: not plotting observations with leverage one:
##   988
```



From the residual plots of the selected model, we can get some measurement of the model.

In the Residual-Fitted plot, for all fitted values, the regression residuals are pretty close to 0, and there is little variation in residuals. This means that the model complies with the assumption that the mean of residuals are 0 and same for all observations very well.

In the Normal Q-Q plot, there are signs of left skewness in the response variable.

In the Scale-Location plot, there are still observations with squared standardized residuals close to 2.0, indicating that they may still be outliers and needs further improvement.

In the leverage plot, from the output warning message, there are observations with leverage value close to 1, indicating that they may be influential points and needs further improvement. We have tried to remove the outlier point and it turns out that new outliers showing up continuously. We regard this as a sign for us to transform and improve the model structure in the sophisticated models.

- RMSE

	RMSE_train	RMSE_test
values	15493.6357	15735.0644

Table 2: RMSE of the OLS-BIC transformed model

From Table 2, we can see that the RMSE for the chosen simple model is 15493.63 on the training data, and 15735.06 on the testing data. From the comparison between two RMSE values, we can tell that there is no overfitting or lack-of-fitting in the proposed model.

- Model testing

Regression formula:

$$\begin{aligned}\log(price) = (intercept) + area + \log(Lot.Area + 1) + Neighborhood + Bldg.Type \\ + Overall.Qual + Overall.Cond + Year.Built + Bsmt.Exposure + BsmtFin.SF.1 \\ + BsmtFin.SF.2 + Bsmt.UnF.SF + Central.Air + Kitchen.Qual + Functional \\ + Fireplaces + Garage.Cars + Paved.Drive + Open.Porch.SF\end{aligned} \tag{1}$$

On the first observation of the training data:

$X = (1, \log(4960 + 1), BrkSide, 1Fam, 5, 7, 1930, 2, 0, 0, 297, Y, 3, 8, 1, 1, 1, 60)$

$$\begin{aligned}\hat{\beta} = (3.925, 2.673 * 10^{-2}, 9.137 * 10^{-2}, -2.217 * 10^{-2}, 0, 5.589 * 10^{-2}, 4.096 * 10^{-2}, \\ 2.877 * 10^{-3}, 1.121 * 10^{-2}, 1.823 * 10^{-4}, 1.367 * 10^{-4}, 8.529 * 10^{-5}, 5.653 * 10^{-2}, 3.870 * 10^{-2}, \\ 2.603 * 10^{-2}, 2.328 * 10^{-2}, 3.726 * 10^{-2}, 1.594 * 10^{-2}, 1.236 * 10^{-4})\end{aligned} \tag{2}$$

After calculation, we can get $X * \hat{\beta} = 11.7535$, $\exp(11.7535) = 127,196.2$, and the true value of `price` in the first observation is $137,000$. The residual value is $137000 - 127196.2 = 9803.8$. We regard this error as a reasonable one.

On the first observation of the testing data:

$X = (1, \log(11727 + 1), NAmes, 1Fam, 7, 6, 1969, 3, 0, 0, 1851, Y, 3, 8, 1, 2, 3, 146)$

$$\begin{aligned}\hat{\beta} = (3.925, 2.673 * 10^{-2}, 9.137 * 10^{-2}, -7.803 * 10^{-2}, 0, 5.589 * 10^{-2}, 4.096 * 10^{-2}, \\ 2.877 * 10^{-3}, 1.121 * 10^{-2}, 1.823 * 10^{-4}, 1.367 * 10^{-4}, 8.529 * 10^{-5}, 5.653 * 10^{-2}, 3.870 * 10^{-2}, \\ 2.603 * 10^{-2}, 2.328 * 10^{-2}, 3.726 * 10^{-2}, 1.594 * 10^{-2}, 1.236 * 10^{-4})\end{aligned} \tag{3}$$

After calculation, we can get $X * \hat{\beta} = 12.2354$, $\exp(12.2354) = 205952.3$, and the true value of `price` in the first observation is $192,100$. The residual value is $192100 - 205952.3 = -13852.3$. We regard this error as a reasonable one.

Development of the final model (20 points)

- Final model: We decided to start with a full model with all available predictors entered. We have tried GAM, boosting, random forest, and BMA, most of them have overfitting issues which lead to poor predictions on test set. To prevent from overfitting using so many predictors, we decided to fit a LASSO linear regression to control the size of the coefficients and do some free variable selections. Since there are some multicolinearity issue, we deleted column `Exterior.1st`,`Exterior.2nd`,`Roof.Matl`,`X1st.Flr.SF`,`X2nd.Flr.SF`,`Total.Bsmt.SF`,`Low.Qual.Fin.SF`. To better fit the data, we included interaction terms between `area` and all the factor predictors, since according to common sense, the extra square feet in different region or neighbourhood has different value. Here is the coefficients of all the non-zero predictors.

name	coefficient
(Intercept)	9.8691092
area	0.0000471
Lot.Area	0.0000021
Lot.Shape	-0.0021008
Utilities	0.0453845
Land.Slope	-0.0017000
Overall.Qual	0.0550863
Overall.Cond	0.0343227
Year.Built	0.0019085
Year.Remod.Add	0.0006171
Mas.Vnr.Area	0.0000210
Exter.Qual	0.0106555
Exter.Cond	0.0012467
Bsmt.Qual	0.0006746
Bsmt.Cond	0.0027856
Bsmt.Exposure	0.0117856
BsmtFin.Type.1	0.0030013
BsmtFin.SF.1	0.0001494
BsmtFin.SF.2	0.0000989
Bsmt.Unf.SF	0.0000602
Heating.QC	0.0073169
Electrical	0.0031368
Bsmt.Full.Bath	0.0001602
Bsmt.Half.Bath	0.0000000
Half.Bath	0.0019251
Kitchen.AbvGr	-0.0271750
Kitchen.Qual	0.0214467
Functional	0.0219264
Fireplaces	0.0287695
Garage.Finish	0.0030061
Garage.Cars	0.0272893
Garage.Area	0.0000759
Garage.Qual	0.0000806
Garage.Cond	0.0071330
Paved.Drive	0.0141207
Wood.Deck.SF	0.0000373
Open.Porch.SF	0.0000849
Enclosed.Porch	0.0000699
Screen.Porch	0.0001333
Mo.Sold	-0.0000880
Yr.Sold	-0.0023416

name	coefficient
TotalSq	0.0001333
MS.SubClass30	-0.0308151
MS.SubClass90	-0.0237519
MS.SubClass150	-0.1418124
MS.SubClass160	-0.0634438
MS.SubClass190	-0.0045966
MS.ZoningRH	-0.0133694
MS.ZoningRM	-0.0564737
Lot.ConfigCulDSac	0.0002097
Lot.ConfigFR2	-0.0136911
NeighborhoodBlueste	0.0018940
NeighborhoodBrDale	-0.0130019
NeighborhoodClearCr	0.0731334
NeighborhoodCrawfor	0.0788170
NeighborhoodEdwards	-0.0367262
NeighborhoodGrnHill	0.4903966
NeighborhoodMeadowV	-0.1049082
NeighborhoodNoRidge	0.0180907
NeighborhoodSawyer	0.0033799
NeighborhoodSawyerW	-0.0113412
NeighborhoodSomerst	0.0272190
NeighborhoodSWISU	-0.0098301
NeighborhoodVeenker	0.0196108
Condition.1Feedr	-0.0184476
Condition.1Norm	0.0171411
Condition.1PosA	0.0009279
Condition.1PosN	0.0391830
Bldg.TypeDuplex	-0.0020432
Bldg.TypeTwnhs	-0.0336596
Bldg.TypeTwnhsE	-0.0038881
House.StyleSLvl	0.0064646
Roof.StyleGambrel	-0.0260529
FoundationStone	-0.0107181
FoundationWood	-0.0437429
HeatingGasW	0.0378745
Central.AirY	0.0581594
Garage.TypeAttchd	0.0040149
Sale.TypeConLI	0.0002901
Sale.TypeOth	-0.0384818
NeighborhoodBrkSide:area	0.0000388
NeighborhoodCrawfor:area	0.0000175
NeighborhoodNAmes:area	-0.0000096
NeighborhoodNridgHt:area	0.0000267
NeighborhoodSomerst:area	0.0000178
NeighborhoodStoneBr:area	0.0000309
area:MS.SubClass90	0.0000000
area:MS.SubClass150	-0.0000090
area:MS.ZoningC (all)	-0.0002073
area:StreetPave	0.0000731
area:Land.ContourHLS	0.0000094
area:Condition.1Norm	0.0000133
area:Condition.1RRAe	-0.0000028

name	coefficient
area:House.Style1Story	0.0000126
area:House.Style2.5Fin	-0.0000013
area:House.StyleSFoyer	0.0000063
area:Roof.StyleMansard	-0.0000146
area:Mas.Vnr.TypeBrkCmn	-0.0000188
area:Mas.Vnr.TypeNone	0.0000062
area:Mas.Vnr.TypeStone	0.0000090
area:FoundationPConc	0.0000172
area:HeatingOthW	-0.0000296
area:HeatingWall	0.0000522
area:Garage.TypeBasment	-0.0000039
area:Garage.TypeCarPort	-0.0000337
area:Garage.TypeUnknown	0.0000329
area:Sale.TypeConLI	0.0000002
area:Sale.TypeCWD	0.0000041
area:Sale.TypeOth	-0.0000034

- Variables: The remaining continuous variable includes `area`,`Lot.Area`,`Lot.Frontage`,`Lot.Area`,`Overall.Qual`,`Overall.Year.Remod.Add`,`Mas.Vnr.Area`,`Exter.Qual`,`Exter.Cond`,`Bsmt.Cond`,`Bsmt.Exposure`, `BsmtFin.Type.1`,`BsmtFin.SF.1`,`Bsmt.Full.Bath`,`Bsmt.Half.Bath`,`Full.Bath`,`alf.Bath`,`Kitchen.AbvGr`,`Kitchen.Qual`, `Functional`,`Fireplaces`,`Garage.Cond`,etc. Basically, ordinal data and area data remained in the equation. Categorical data like `neighborhood`,`Bldg.Type`,etc remained in the equation as usual.
- Variable selection/shrinkage: The penalizing constant we picked was 0.001787526 which was determined by cross validation. Under this regularization, the number of predictors reduced from 272 to 109 by using LASSO. As predicted, location variables like `neighborhood` and the its interaction with area remained in the model.

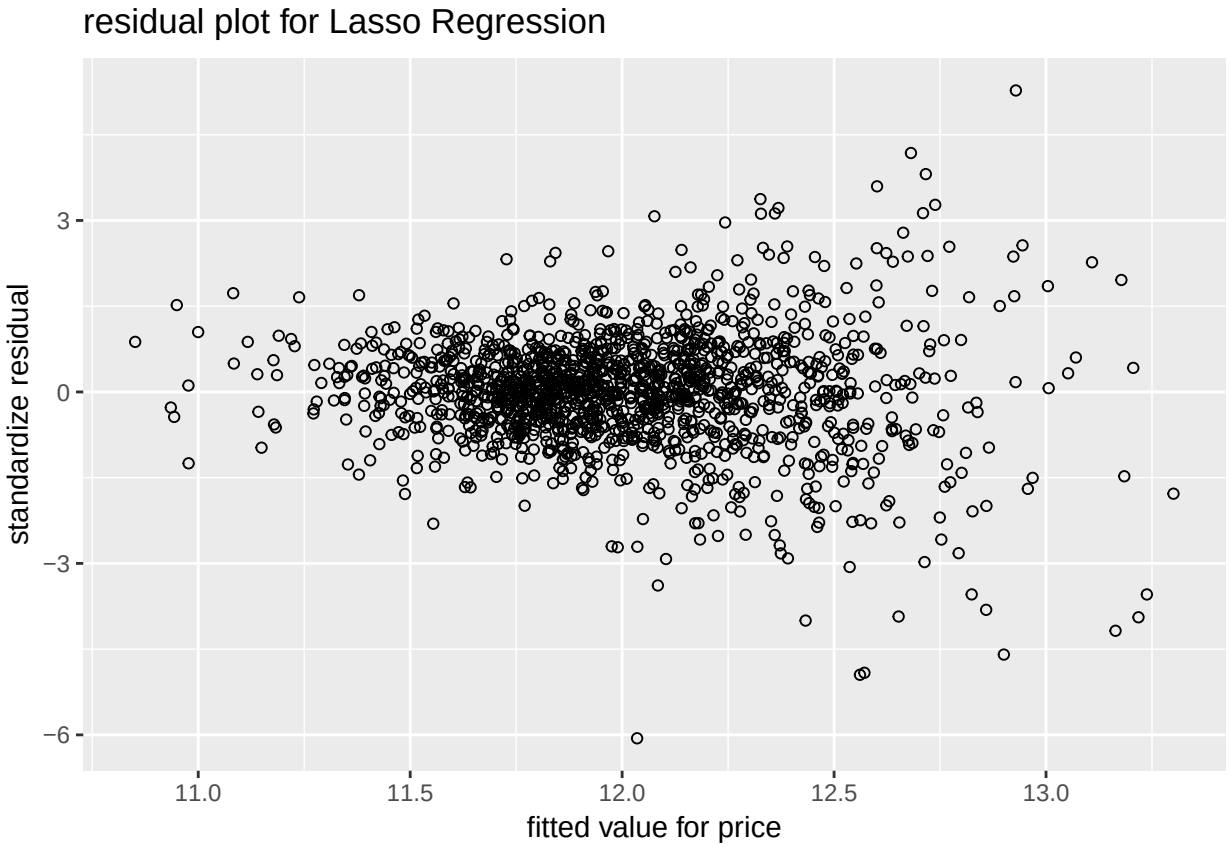
Assessment of the final model (25 points)

- Residual: must include a residual plot and a discussion

```
X_transformed = model.matrix(fmla,train_clean)
X_transformed2= model.matrix(fmla,test_clean)


yhat = predict(model3,
                      newx = X_transformed[,-1],
                      s = model3.lambda.best)
residual = scale(exp(yhat)- train_clean$price)

ggplot(data.frame(cbind(residual,train_clean$price)), aes(x= yhat, y=residual)) +
    geom_point(shape=1)  +
    ggtitle("residual plot for Lasso Regression") +
    labs(x="fitted value for price",y="standardize residual")
```



Since we can not use package to plot residuals of Lasso Regression, we plot manually. First, we calculate the residual of training data as y values for the plot. Residuals equal fitted value of training data minius true value, and then we standardized the residual. Our plot has residaul as y values and fitted price as x values. The standardized residual plot shows that most residual are within -3 and +3, and Most of points are between evenly distributed.

MSE: must include an RMSE and an explanation (other criteria desirable)

```
# check RMSE for training data
rmse_lasso_train = RMSE(exp(yhat),train_clean$price)
rmse_lasso_train
```

```
## [1] 14965.16
```

```
model3.lambda.best = model3$lambda.min
model3.lambda.best
```

```
## [1] 0.001787526
```

```
# check RMSE for test data
model3.pred = predict(model3,
                      newx = X_transformed2[,-1],
                      s = model3.lambda.best)
rmse_lasso = RMSE(exp(model3.pred),test_clean$price)
rmse_lasso
```

```
## [1] 14740.65
```

rmse value for training data using simple model is , and for test data using simple model is .

rmse value for training data using Lasso Regression with selected interaction is 14890.75 and for test data with selected interaction is 14727.49.

Compared to simple model, rmses for both trainig data and test data using Lasso Regression are smaller than rmses using simple model, which means both bias and variance decreased with complex model.

```
nsim = 50
y_test_pred = matrix(0,500,50)
for(i in 1:nsim){

  model4 = cv.glmnet(X_transformed[,-1],
                     as.matrix(log(train_clean[, "price"])),
                     alpha=1,
                     lambda= 10^seq(4,-3,length= 1000))
  model4.lambda.best = model4$lambda.min


  model4.pred = predict(model4,
                        X_transformed2[,-1],
                        s = model4.lambda.best)
  y_test_pred[,i] = model4.pred
}

y_mean = apply(exp(y_test_pred),1,mean)
y_quantile = apply(exp(y_test_pred),1,quantile,c(0.025,0.975))

coverage(y_quantile[,1],y_quantile[,2],test_clean$price)
```

```
## [1] 0.038
```

```
RMSE(y_mean,test_clean$price)
```

```
## [1] 14696.26
```

The result shows that the coverage for Lasso Regression is 4.4%. Since Lasso Regression is OLS with L2 penalizaiton, which leads to bias with lower variance. So, we get really low coverage due to low variance.

- Model evaluation: must include an evaluation discussion

Method: To predict price of housing, we constructed multiple linear regression with stepwise BIC, Ridge Reression, Lasso Regression, Random Forest, boosting and BMA. Though linear regression with stepwise BIC can do variable selection, its rmse is relative higher. Ridge Regression cannot do variable selection. Random Forest, boosting and BMA are lack of interpretibility. Compared to rmses of other models,we found Lasso has the smallest rmse. Also Lasso has variable selection to help us with selecting predictors. So we choose Lasso Regression as final model.

- Model testing : must include a discussion

Since Lasso Regression involves too many predictors, it is hard to predict manually, so we use our model to predict the 1st observation of training and test data.

```
# predict for 1st observation for training data
pred_1ob_train = exp(predict(model4,
                     X_transformed[,-1],
                     s = model4.lambda.best)[1])[1]
diff_1ob_train = pred_1ob_train - train_clean$price[1]
diff_1ob_train
```

```
## [1] 1399.927
```

The result above shows that using our Lasso Regression model to predict price of 1st observation in training data is 137571.3 and the true value of it is 137000. The difference is only 571.2981, which proves the validation of our model.

```
# predict for 1st observation for test data
pred_1ob_test = exp(predict(model4,
                      X_transformed2[,-1],
                      s = model4.lambda.best)[1])[1]
diff_1ob_test = pred_1ob_test - test_clean$price[1]
diff_1ob_test
```

```
## [1] 9106.858
```

The result above shows that using our Lasso Regression model to predict price of 1st observation in test data is 200813.3 and the true value of it is 192100. The difference is only 8713.273, also proves the validation of our model.

- Model result: must include a selection of the top 10 undervalued and overvalued houses

```
set.seed(2)
diff = data.frame(exp(model3.pred) - test_clean$price)
diff = cbind(diff,test_clean$PID)
overvalued = diff %>%
  arrange(desc(X1)) %>%
  head(.,10)
overvalued
```

```
##          X1 test_clean$PID
## 1  60760.30      906402200
## 2  47936.11      533251110
## 3  45273.74      528102010
## 4  41720.55      527355150
## 5  36567.82      527182020
## 6  33215.76      531380080
## 7  32518.70      904101070
## 8  32246.67      528116010
## 9  31623.73      528327070
## 10 30558.85      906426195
```

```
undervalued = diff %>%
  arrange(X1) %>%
  head(.,10)
undervalued
```

```
##           X1 test_clean$PID
## 1  -66706.08      905376090
## 2  -66371.68      921128020
## 3  -47679.99      528178070
## 4  -47388.95      528344070
## 5  -45950.09      905427010
## 6  -44784.35      905201080
## 7  -44102.93      909275110
## 8  -42982.34      535454070
## 9  -42622.31      903400220
## 10 -36231.13      903429110
```

Conclusion (10 points): must include a summary of results and a discussion of things learned

summary of results: we found linear regression only with three predictors TotalSq, Total.Bsmt.SF, Neighborhood could explained around 82% of the training data. The model we got by using stepwise BIC, which filtered out 19 variables from 82 variables, only improve it to 94%. It suggested TotalSq, toEven this model explained more training data and got a better RMSE on testing data. We still doubted this model may be to complicated.

We have used multiple linear regression with stepwise BIC, Ridge regression, Lasso regression, Poisson regression, Random forest, boosting, BMA to fit the training data and compared rmses both from training data and testing data. We finally found Lasso regression cames with the best testing and training rmses.

Things learned: 1. The first and most important part of data analysis would be data cleaning. We spent much time on cleaning data, including NA imputation, Ordinal variable transfermation to reasonable numerical variable, and factors defination. Only with proper data cleaning, we can start models building.

2. Before fitting any models, we have to fully understand the data. We looked through the description of all variables, and wrote R script to summary features of different variables. More deeply we can understand data, more possiblly we can make proper data transformation and create innovative interaction.

3. We tried through all statistical models we have learned from STA521, and we learned that complicated and advanced models don't guarantee better predictions. The multiple linear regression with simple stepwise BIC generated better preditions than most advanced and complicated model, which were beyond our expectation.