STA 521 Final Project Writeup

BayeStar

April 26, 2017

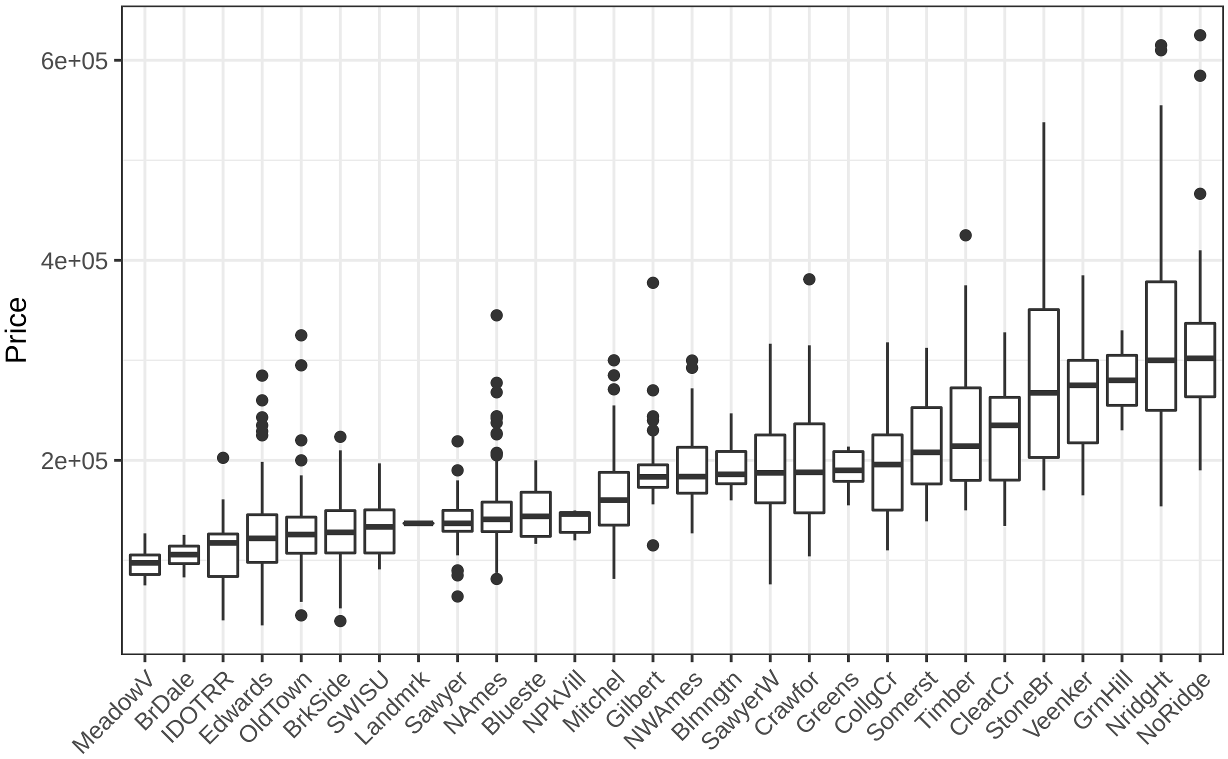
Exploratory data analysis

```
load("ames_train.Rdata")
load("ames_test.Rdata")
```

- Figure 1 shows the boxplot of prices by each neighborhood. It is apparent that prices vary a lot by different neighborhoods. Outliers are also captured and we considered deleting some of them.
- Figure 2 is a scatterplot of prices against kitchen quality. The left panel corresponds to houses without porch, while the right with porch. We can see that the relationship differs between with and without porch, so we decided to add a dummy variable indicating the existence of porch and an interaction between the dummy and porch area.
- Figure 3 shows the prices of houses built in various years. There is a non-linear trend, so we considered adding quadratic term of year to capture the non-linearity.
- Figure 4 depicts the relationship between prices and total square by neighborhood. (Only four of neighborhoods are shown for plot purpose.) Prices and total square are linearly related, but the linear relationship changes across neigborhood. We considered adding interaction term of total square and neighborhood to capture different slopes.
- Figure 5 is a vilion plot showing the distribution of prices by overall quality. Prices are higher, but also vary more for higher quality.

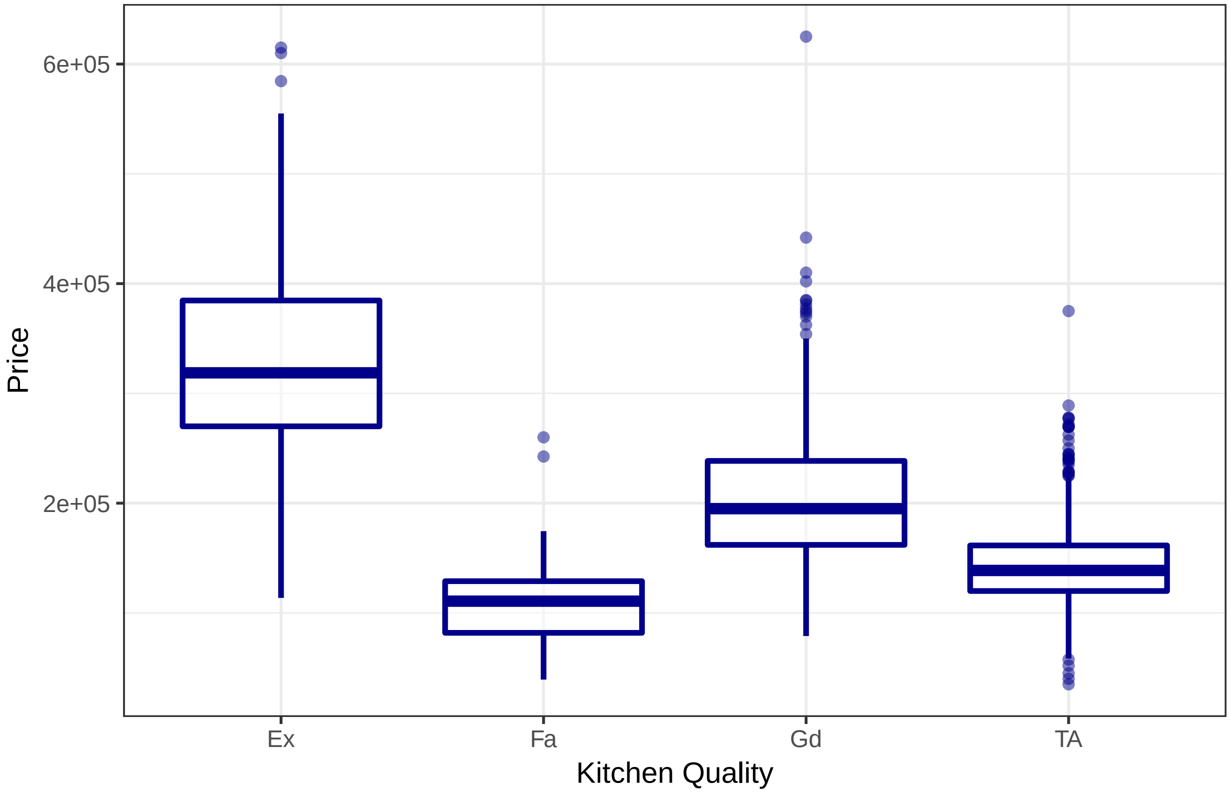
```
# price v.s. neighborhood
ggplot(ames_train, aes(x=reorder(Neighborhood, price, FUN=median), y=price))+
  theme_bw()+
  theme(axis.text.x=element_text(angle=45, hjust=1))+
  geom_boxplot()+
  xlab('')+
  ylab('Price')+
  ggtitle("Figure 1. Prices by Neighborhood")
```

Figure 1. Prices by Neighborhood



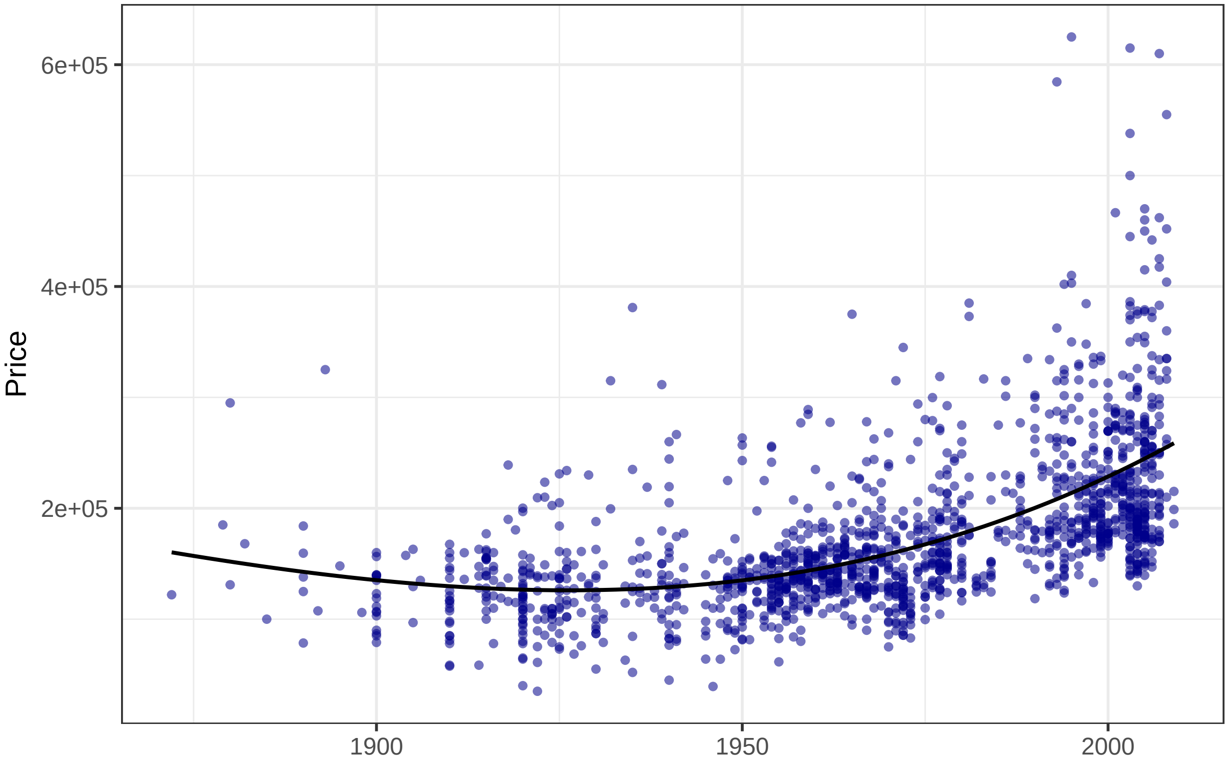
```
# price v.s. Kitchen.Qual
ames_train %>%
  ggplot(aes(x=Kitchen.Qual, y=price))+
  geom_boxplot(col="dark blue", cex=1, alpha=0.5)+
  xlab("Kitchen Quality")+
  ylab("Price")+
  theme_bw()+
  ggtitle("Figure 2. Prices against Kitchen Quality")
```

Figure 2. Prices against Kitchen Quality



```
# year
ggplot(ames_train, aes(x=Year.Built, y=price,alpha = 0.5))+
  geom_point(col="dark blue", cex=1)+
  geom_smooth(method = "lm", formula = y ~ x + I(x^2)+I(x^3),
              col="black", se=F, size=0.7)+
  theme_bw()+
  theme(legend.position="none")+
  ylab('Price')+
  xlab('')+
  ggtitle("Figure 3. Prices of Houses Built in Years")
```

Figure 3. Prices of Houses Built in Years



```
# Neighborhood and TotalSq
ames_train %>%
  filter(Neighborhood=="OldTown" |Neighborhood=="NridgHt"|
           Neighborhood=="NWAmes"|Neighborhood=="StoneBr") %>%
  ggplot(aes(x=TotalSq, y=price))+
  geom_point(col="dark blue", cex=1, alpha=0.5)+
  facet_wrap(~Neighborhood)+
  geom_smooth(method="lm", se=F, col="black", size=0.7)+
  theme_bw()+
  xlab("Total Square")+
  ylab("Price")+
  ggtitle("Figure 4. Prices by Total Square in Different Neighborhoods")
```

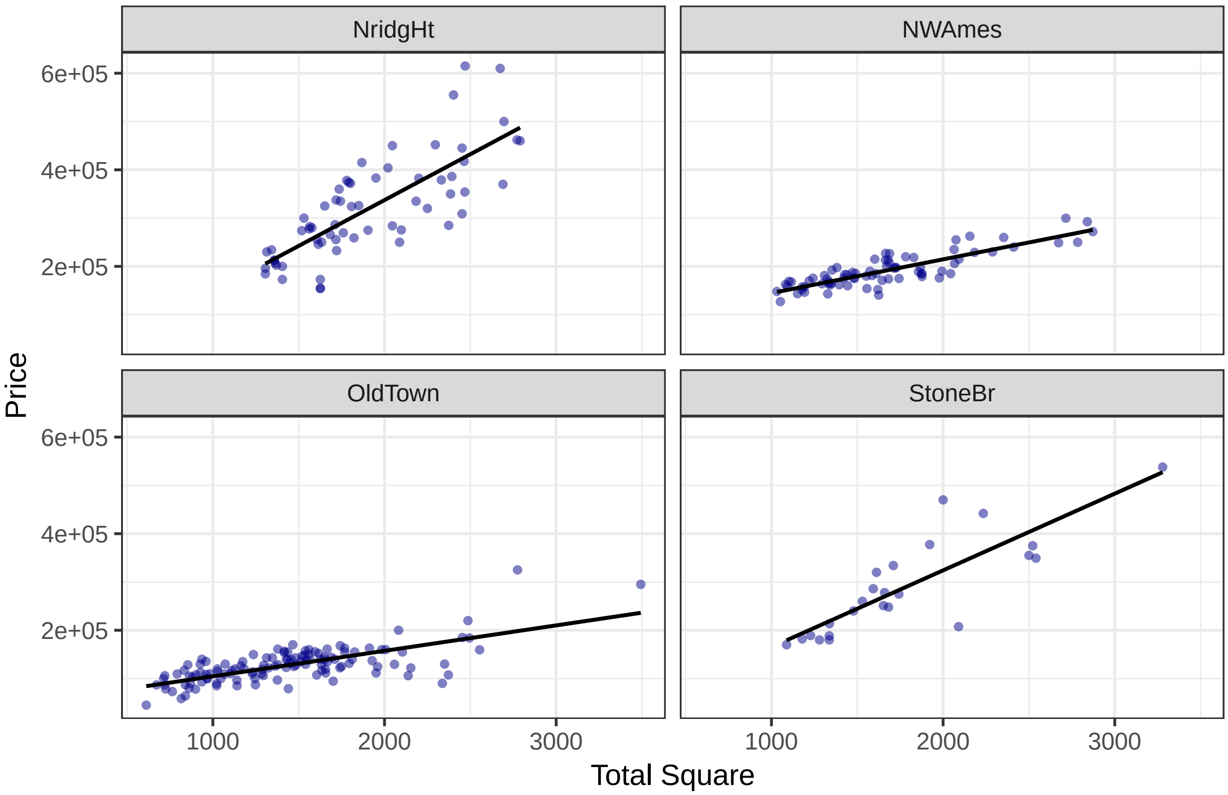
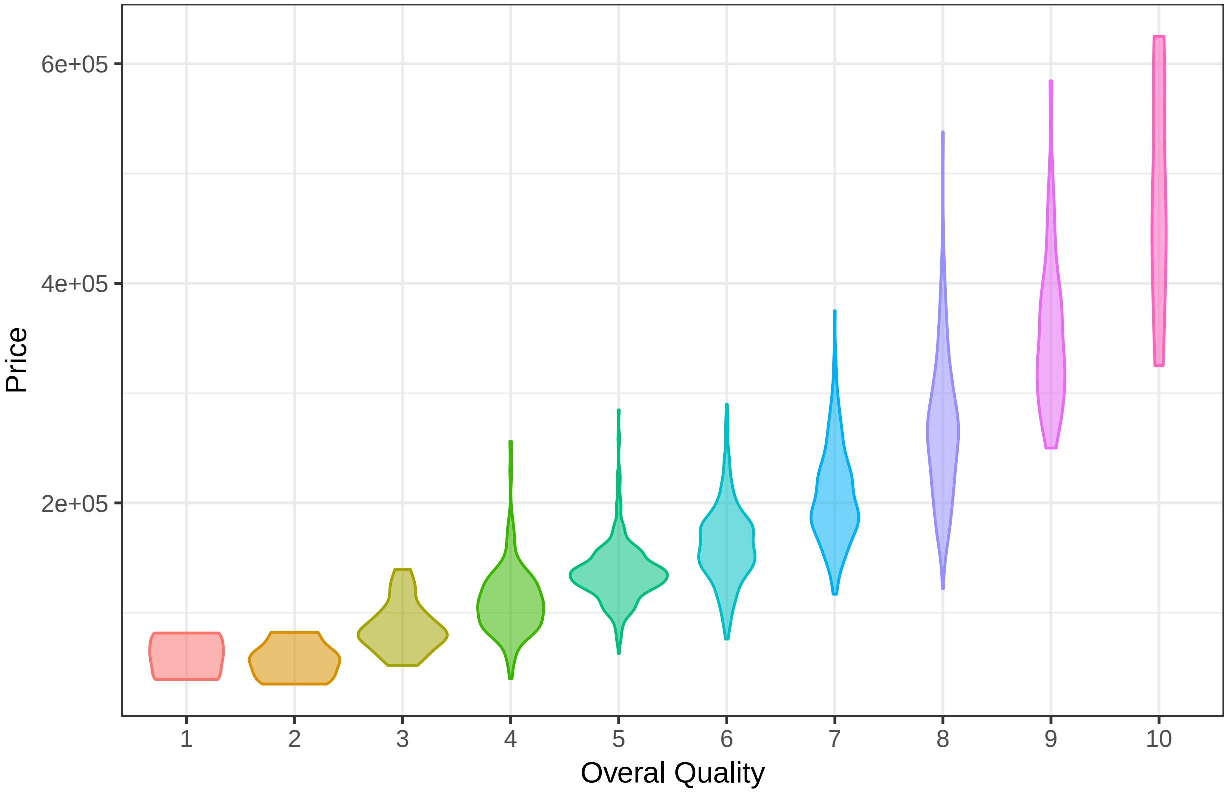


Figure 4. Prices by Total Square in Different Neighborhoods

```
# Overall.Qual
ggplot(ames_train, aes(x=factor(Overall.Qual), y=price))+
  geom_violin(aes(color=factor(Overall.Qual),
                  fill = factor(Overall.Qual), alpha=0.5))+
  theme_bw() +
  theme(legend.position="none")+
  xlab("Overal Quality")+
  ylab("Price")+
  ggtitle("Figure 5. Prices by Overall Quality")
```

Figure 5. Prices by Overall Quality



Simple Model

Data Cleaning

```
# clean data
clean_data = function(xdata){
xdata %>%
    mutate(# replace NAs with new levels
           Alley = as.factor(ifelse(is.na(as.character(Alley)),
                                    "No alley access", as.character(Alley))),
           Bsmt.Qual = as.factor(ifelse(as.character(Bsmt.Qual)=="Po",
                                        "Fa", as.character(Bsmt.Qual))),
           Bsmt.Qual = as.factor(ifelse(is.na(as.character(Bsmt.Qual)),
                                        "No Basement", as.character(Bsmt.Qual))),
           Bsmt.Cond = as.factor(ifelse(is.na(as.character(Bsmt.Cond)),
                                        "No Basement", as.character(Bsmt.Cond))),
           BsmtFin.Type.1 = as.factor(ifelse(is.na(as.character(BsmtFin.Type.1)),
                                        "No Basement", as.character(BsmtFin.Type.1))),
           BsmtFin.Type.2 = as.factor(ifelse(is.na(as.character(BsmtFin.Type.2)),
                                        "No Basement", as.character(BsmtFin.Type.2))),
           Bsmt.Exposure = as.factor(ifelse(is.na(as.character(Bsmt.Exposure))|
                                           as.character(Bsmt.Exposure) == "",
                                        "No Basement", as.character(Bsmt.Exposure))),
           Bsmt.Unf.Rate.SF = ifelse(Total.Bsmt.SF!=0, Bsmt.Unf.SF/Total.Bsmt.SF, 0),
           Bsmt.Full.Bath = ifelse(is.na(Bsmt.Full.Bath),0,Bsmt.Full.Bath),
           Bsmt.Half.Bath = ifelse(is.na(Bsmt.Half.Bath),0,Bsmt.Half.Bath),
           Fireplace.Qu = as.factor(ifelse(is.na(as.character(Fireplace.Qu)),
                                           "No Fireplace", as.character(Fireplace.Qu))),
           Garage.Type = as.factor(ifelse(is.na(as.character(Garage.Type)),
                                           "No Garage", as.character(Garage.Type))),
           Garage.Finish = as.factor(ifelse(is.na(as.character(Garage.Finish))|
                                              as.character(Garage.Finish) == "",
                                           "No Garage", as.character(Garage.Finish))),
           Garage.Qual = as.factor(ifelse(as.character(Garage.Qual)=="Ex",
                                          "Gd", as.character(Garage.Qual))),
           Garage.Qual = as.factor(ifelse(is.na(as.character(Garage.Qual)),
                                           "No Garage", as.character(Garage.Qual))),
           Garage.Cond = as.factor(ifelse(as.character(Garage.Cond)=="Ex",
                                          "Gd", as.character(Garage.Cond))),
           Garage.Cond = as.factor(ifelse(is.na(as.character(Garage.Cond))|
                                            as.character(Garage.Cond)=="Po",
                                           "No Garage", as.character(Garage.Cond))),
           # deal with new level issue in test data
           Fence = as.factor(ifelse(is.na(as.character(Fence)),
                                           "No Fence", as.character(Fence))),
           Misc.Feature = as.factor(ifelse(is.na(as.character(Misc.Feature)),
                                           "None", as.character(Misc.Feature))),
           Mas.Vnr.Type = as.factor(ifelse(as.character(Mas.Vnr.Type) == "",
                                           "None", as.character(Mas.Vnr.Type))),
           Mas.Vnr.Area = ifelse(is.na(Mas.Vnr.Area),0,Mas.Vnr.Area),
           Kitchen.Qual = as.factor(ifelse(as.character(Kitchen.Qual)=="Po",
                                           "Fa", as.character(Kitchen.Qual))),
           Heating.QC = as.factor(ifelse(as.character(Heating.QC)=="Po",
                                         "Fa", as.character(Heating.QC))),
           Electrical = as.factor(ifelse(as.character(Electrical) == "",
                                         "SBrkr", as.character(Electrical))),
           Condition.2 = as.factor(ifelse(as.character(Condition.2) %in%
                                            c("Artery","RRAn","RRAe"),
                                          "Feedr", as.character(Condition.2))),
           Neighborhood = as.factor(ifelse(as.character(Neighborhood)=="Blueste",
                                          "NPkVill", as.character(Neighborhood))),
           # create new variables
           Enclosed.Porch.is = as.factor(ifelse(Enclosed.Porch==0,"N","Y")),
           Pool.Area = as.factor(ifelse(Pool.Area==0,"N", "Y")),
           Garage.Yr.Blt = ifelse(is.na(Garage.Yr.Blt), Year.Built-2, Garage.Yr.Blt)
          )%>%
    dplyr::select(-c(Lot.Frontage,Pool.QC,Pool.Area))
}

# remove outliers
ames_train = clean_data(ames_train)
ames_train = ames_train[-c(462,168,183),]
ames_train = ames_train[ames_train$price<500000,]
ames_train = ames_train[ames_train$price>50000,]
ames_train = ames_train[ames_train$X1st.Flr.SF<3500,]
ames_train = ames_train[ames_train$Kitchen.AbvGr%in%c(1,2),]
remove_idx1 = c(1:nrow(ames_train))[ames_train$Neighborhood %in%c("Gilbert")&ames_train$price>350000]
remove_idx2 = c(1:nrow(ames_train))[ames_train$Neighborhood %in%c("NAmes")&ames_train$price>300000]
remove_idx3 = c(1:nrow(ames_train))[ames_train$Neighborhood %in%c("Landmrk","GrnHill")]
```

```
ames_train = ames_train[-c(remove_idx1, remove_idx2, remove_idx3),]
```

Initial Model

- Transforamtion of the response variable `price`: We first fitted a simplest model called `simple_model` to identify possible transformations of response variable `price`. The diagnostic plots showed below indicate a non-constant variance of the residuals, which means a transformation is needed for `price`. With the help of `boxcox()` function and the plot produced below, we found that `price` needs a log transformation.
- Model fitting and model selection: We first putted in a few variables that are intuitively important as out base model and kept all the variables put in significant. The variables we chose include `TotalSq`, `Year.Built`, `Garage.Area`, `Overall.Qual`, `Kitchen.Qual`, `Garage.Cond`, etc. Based on this base model, forward selection was used to select more variables that can be included in our model and improve the prediction accuracy. In this process we try to avoid the variables might be correlated with other variables, such as `TotalSq` an `area`. With the help of the diagonostic plots, we also removed some outliers, and finally end up with a model with 20 variables.

- Results and explanation of coeficients: Based on the summary table of the selected model, all the selected continuous variables are extremely significant. These continuous variables include `area`, `Year.Built`, `Year.Remod.Add`, `Garage.Area`, `Overall.Qual`, `Lot.Area`, `BsmtFin.SF.1`, `Overall.Cond`, `Total.Bsmt.SF`, `Bsmt.Full.Bath` and `Screen.Porch`. Additionally, some categorical variables having a lot of significant levels are selected, such as `Neighborhood`, `Kitchen.Qual`, `Exter.Qual`.

a. Area is the most significant continuous variable in the model with a coefficient of 1.373e-03, which indicates when the other conditions stay the same, one unit increase in area leads to exp(2.702e-04)=1.0002 times of original price. The reason for having such a small increasing ratio is that the 1.0002 times a price with a large magnitude can still lead to a big increment.

b. Kichen.Qual if the most significant categorical variable. The base case is level "Ex". Level "Gd" has a coefficient of -5.298e-02, which means properties with a good quality kitchen have prices exp(-5.298e-02)=0.948 times of the prices of properties with an excellent quality kitchen. Level "TA" and "Fa" have lower ratios 0.9296 and 0.9218 respectively, which indicates properties with a better kitchen will have a higher price.

```
model1 = lm(log(price) ~ area + Year.Built + Year.Remod.Add +
              Garage.Area + Overall.Qual +  log(Lot.Area) +
              BsmtFin.SF.1 + Overall.Cond + Total.Bsmt.SF +
              Central.Air + Bsmt.Full.Bath + Screen.Porch +
              Kitchen.Qual + Exter.Qual + Bldg.Type + Bsmt.Qual +
              Garage.Cond + Neighborhood + Heating.QC, data=ames_train)
summary(model1)
```

```
##
## Call:
## lm(formula = log(price) ~ area + Year.Built + Year.Remod.Add +
##     Garage.Area + Overall.Qual + log(Lot.Area) + BsmtFin.SF.1 +
##     Overall.Cond + Total.Bsmt.SF + Central.Air + Bsmt.Full.Bath +
##     Screen.Porch + Kitchen.Qual + Exter.Qual + Bldg.Type + Bsmt.Qual +
##     Garage.Cond + Neighborhood + Heating.QC, data = ames_train)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44427 -0.04852 -0.00067  0.05263  0.28964
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3.141e+00  5.299e-01   5.928 3.85e-09 ***
## area                  2.720e-04  7.344e-06  37.039  < 2e-16 ***
## Year.Built            3.011e-03  2.115e-04  14.237  < 2e-16 ***
## Year.Remod.Add        5.135e-04  1.807e-04   2.841 0.004559 **
## Garage.Area           1.214e-04  1.856e-05   6.545 8.31e-11 ***
## Overall.Qual          5.747e-02  3.265e-03  17.599  < 2e-16 ***
## log(Lot.Area)         9.650e-02  8.107e-03  11.904  < 2e-16 ***
## BsmtFin.SF.1          7.858e-05  8.096e-06   9.706  < 2e-16 ***
## Overall.Cond          4.146e-02  2.750e-03  15.077  < 2e-16 ***
## Total.Bsmt.SF         9.290e-05  9.618e-06   9.659  < 2e-16 ***
## Central.AirY          4.778e-02  1.224e-02   3.905 9.88e-05 ***
## Bsmt.Full.Bath        2.509e-02  5.795e-03   4.330 1.60e-05 ***
## Screen.Porch          1.572e-04  4.448e-05   3.534 0.000423 ***
## Kitchen.QualFa       -1.063e-01  2.212e-02  -4.806 1.70e-06 ***
## Kitchen.QualGd       -5.633e-02  1.523e-02  -3.698 0.000225 ***
## Kitchen.QualTA       -7.826e-02  1.635e-02  -4.787 1.87e-06 ***
## Exter.QualFa         -6.988e-02  3.329e-02  -2.099 0.035992 *
## Exter.QualGd         -2.503e-02  2.118e-02  -1.182 0.237522
## Exter.QualTA         -1.939e-02  2.274e-02  -0.853 0.393921
## Bldg.Type2fmCon      -2.481e-02  1.808e-02  -1.372 0.170270
## Bldg.TypeDuplex      -7.550e-02  1.467e-02  -5.147 3.02e-07 ***
## Bldg.TypeTwnhs       -3.495e-02  1.958e-02  -1.785 0.074439 .
## Bldg.TypeTwnhsE      -5.121e-03  1.385e-02  -0.370 0.711730
## Bsmt.QualFa          -2.319e-02  2.106e-02  -1.101 0.270974
## Bsmt.QualGd          -3.412e-02  1.249e-02  -2.733 0.006362 **
## Bsmt.QualNo Basement -4.411e-02  2.276e-02  -1.939 0.052751 .
## Bsmt.QualTA          -4.083e-02  1.478e-02  -2.762 0.005818 **
## Garage.CondGd        -2.095e-02  3.965e-02  -0.528 0.597439
## Garage.CondNo Garage  9.789e-03  1.826e-02   0.536 0.592070
## Garage.CondTA         2.972e-02  1.508e-02   1.971 0.048863 *
## NeighborhoodBrDale   -1.202e-01  3.491e-02  -3.442 0.000594 ***
## NeighborhoodBrkSide  -4.401e-02  3.185e-02  -1.382 0.167211
## NeighborhoodClearCr   1.413e-02  3.371e-02   0.419 0.675189
## NeighborhoodCollgCr  -8.343e-02  2.781e-02  -3.000 0.002750 **
## NeighborhoodCrawfor   5.877e-02  3.150e-02   1.865 0.062339 .
## NeighborhoodEdwards  -1.286e-01  2.994e-02  -4.294 1.88e-05 ***
## NeighborhoodGilbert  -7.167e-02  2.935e-02  -2.442 0.014740 *
## NeighborhoodGreens    6.049e-02  4.285e-02   1.412 0.158270
## NeighborhoodIDOTRR   -1.359e-01  3.314e-02  -4.101 4.35e-05 ***
## NeighborhoodMeadowV  -1.650e-01  3.830e-02  -4.310 1.75e-05 ***
## NeighborhoodMitchel  -9.186e-02  2.976e-02  -3.087 0.002060 **
## NeighborhoodNAmes    -9.936e-02  2.891e-02  -3.436 0.000606 ***
## NeighborhoodNoRidge  -2.459e-02  3.071e-02  -0.801 0.423397
## NeighborhoodNPkVill  -2.131e-02  3.312e-02  -0.643 0.520014
## NeighborhoodNridgHt  -2.700e-02  2.944e-02  -0.917 0.359192
## NeighborhoodNWAmes   -1.006e-01  2.951e-02  -3.408 0.000672 ***
## NeighborhoodOldTown  -1.043e-01  3.111e-02  -3.351 0.000826 ***
## NeighborhoodSawyer   -9.429e-02  3.004e-02  -3.139 0.001732 **
## NeighborhoodSawyerW  -1.055e-01  2.910e-02  -3.625 0.000299 ***
## NeighborhoodSomerst   3.694e-03  2.738e-02   0.135 0.892710
## NeighborhoodStoneBr   1.540e-02  3.127e-02   0.493 0.622418
## NeighborhoodSWISU    -8.148e-02  3.472e-02  -2.347 0.019085 *
## NeighborhoodTimber   -7.494e-02  3.159e-02  -2.372 0.017802 *
## NeighborhoodVeenker  -3.430e-02  3.671e-02  -0.934 0.350269
## Heating.QCFa         -1.887e-02  1.475e-02  -1.279 0.201185
## Heating.QCGd         -9.540e-03  7.094e-03  -1.345 0.178899
## Heating.QCTA         -1.366e-02  6.809e-03  -2.007 0.044985 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08784 on 1423 degrees of freedom
## Multiple R-squared:  0.9408, Adjusted R-squared:  0.9385
## F-statistic: 403.8 on 56 and 1423 DF,  p-value: < 2.2e-16
```

Model Selection

We compared our base model and models with selected added terms from stepwise selection, then used anova to show the significance of adding those variables. All of the added variables are significant, which leads us to our final initial model.

```
base = lm(log(price) ~ X1st.Flr.SF + Year.Built + Year.Remod.Add +
            Garage.Area + Overall.Qual + Kitchen.Qual + Neighborhood,
          data=ames_train)

base1 = lm(log(price) ~ area + Year.Built + Year.Remod.Add +
             Garage.Area + Overall.Qual + Kitchen.Qual +
             Neighborhood + log(Lot.Area) , data=ames_train)

base2 = lm(log(price) ~ area + Year.Built + Year.Remod.Add +
             Garage.Area + Overall.Qual + Kitchen.Qual +
             Neighborhood + log(Lot.Area) + BsmtFin.SF.1,
           data=ames_train)

base3 = lm(log(price) ~ area + Year.Built + Year.Remod.Add +
             Garage.Area + Overall.Qual + Kitchen.Qual + Neighborhood +
             log(Lot.Area) + BsmtFin.SF.1 + Overall.Cond , data=ames_train)

base4 = lm(log(price) ~ area + Year.Built + Year.Remod.Add +
             Garage.Area + Overall.Qual + Kitchen.Qual + Neighborhood +
             log(Lot.Area) + BsmtFin.SF.1 +  Overall.Cond + Total.Bsmt.SF,
           data=ames_train)

base5 = lm(log(price) ~ area + Year.Built + Year.Remod.Add + Garage.Area
           + Overall.Qual + Kitchen.Qual + Neighborhood + log(Lot.Area) +
             BsmtFin.SF.1 +  Overall.Cond + Total.Bsmt.SF + Central.Air, data=ames_train)

model = lm(log(price) ~ area + Year.Built + Year.Remod.Add + Garage.Area +
              Overall.Qual + Kitchen.Qual + Neighborhood +log(Lot.Area) +
              BsmtFin.SF.1 + Overall.Cond + Total.Bsmt.SF + Central.Air +
              Bsmt.Full.Bath + Screen.Porch  + Exter.Qual + Bldg.Type +
              Bsmt.Qual + Garage.Cond  + Heating.QC, data=ames_train)
```

```
kable(anova(base,base1,base2, base3, base4, base5, model), caption = "ANOVA")
```

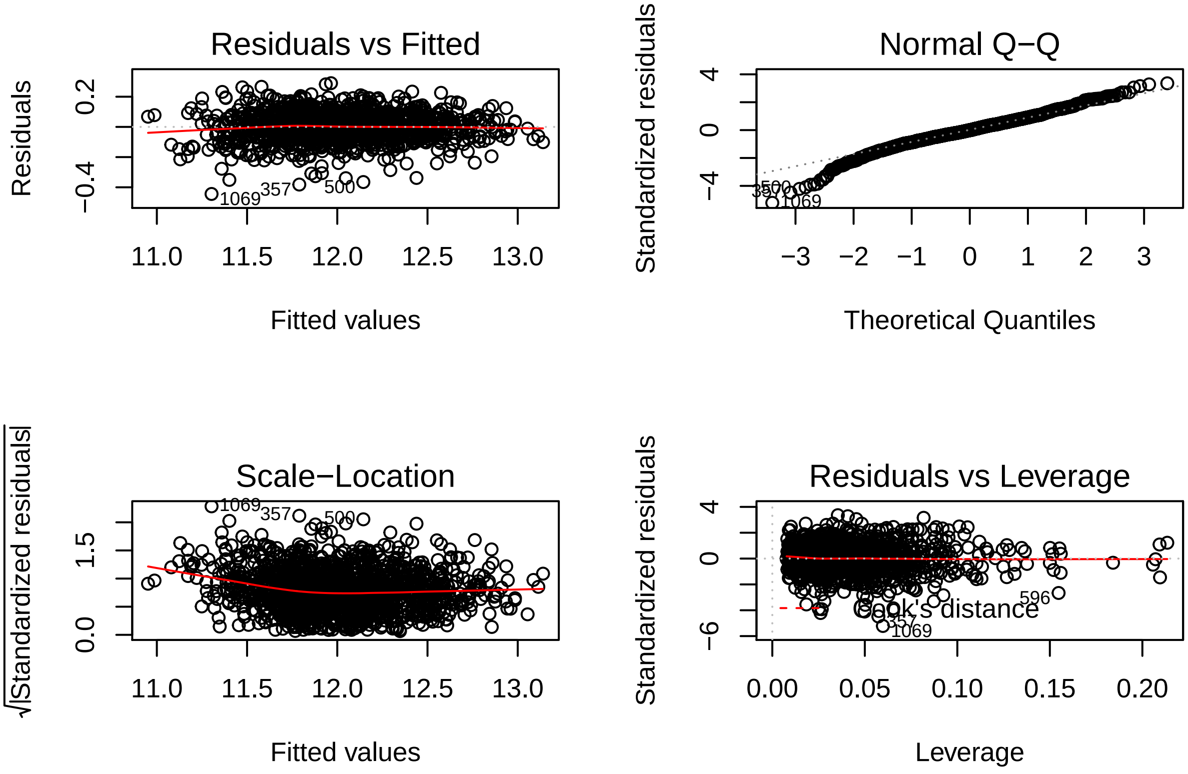
Table 1: ANOVA

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1447	28.87855	NA	NA	NA	NA
1446	18.85622	1	10.0223367	1299.063242	0
1445	15.23400	1	3.6222226	469.500916	0
1444	13.21175	1	2.0222403	262.116319	0
1443	12.07458	1	1.1371764	147.397166	0
1442	11.74968	1	0.3248958	42.111959	0
1423	10.97851	19	0.7711676	5.260857	0

Residual Diagnostics

The residual plots shows no non-constant variance, and the qq-plot shows a good normality of the redisuals except for a few potential outliers.

```
par(mfrow = c(2,2)); plot(model1)
```



Model Evaluation

Since log transformation was used on `price`, it needs to be transformed back to the original scale.

```
ames_test = clean_data(ames_test)
Yhat1 = predict(model, newdata=ames_test, interval = "pred")
Yhat1 = exp(Yhat1)
# test criteria
```

```r
rmse = function(y, yhat){
  sqrt(mean((y-yhat)^2))
}

bias = function(yhat, y){
  mean(yhat-y)
}
maxDeviation = function(yhat, y){
  max(abs(yhat-y))
}
meanDeviation = function(yhat, y){
  mean(abs(yhat-y))
}
coverage = function(y, lwr, upr){
  mean(y>=lwr && y<=upr)
}
# evaluation
rmse1 = rmse(Yhat1[,1], ames_test$price)
bias1 = bias(Yhat1[,1], ames_test$price)
maxDeviation1 = maxDeviation(Yhat1[,1], ames_test$price)
meanDeviation1 = meanDeviation(Yhat1[,1], ames_test$price)
coverage1 = coverage(ames_test$price, Yhat1[,2], Yhat1[,3])
```

Model Checking

Based on our model above, we calculated prediction for the first observation in both the training and test data. For the training data, the prediction for the first observation (with PID 526354020) is 11.78. After converting it to original unit, we got 130614, which is very close to the true value 137000. For the test data, the prediction for the first observation is 12.21, which is 200787. This is also a reasonable prediction since the true value is 192100.

Complex Model

Model Fitting

We decided to keep using linear regression to fit the data. Variable Selection, data transformation, and interaction between variables are three steps we considered.

1. Variable Selection

We selected a subgroup of variables out of all the predictors by the following criterion. * Some variables are highly imbalanced. For example, `Utilities` has three categories, but one of the category has 99% of the observations in it. There is small variation and little information within these variables. Thus, these variables are out of our consideration. The fucntion `nearZeroVar()` in package `caret` was used to identify these variables. Both continuous variables and categorical variables can be identified. * Some variables have a large proportion of missing values. For instance, `Alley` has 1395 missing values, consisting of 93% of the training data. We remove these variables since they are noninformative and may cause damage to our models. * There is also multicolinearity in the training data. For example, `area` and `TotalSq` have a correlation around 0.99. Since To avoid multicolinearity, we will not consider `area` in our models.

After removing imbalanced, missing, and correlated variables, we ended up with 41 predictors. We built our models based on these predictors.

2. Data Transformation

a. log transformation: based on the histogram plots of continuous variables, we found out that some of them are extremely skewed. We took log transformation so that they become more bell shaped. One problem of log transformation is that original zero values change to negative infinity. In order to avoid infinity values, we add 1 unit to those variables with zeros values.
b. polynomial relationships: base on the plots of price vs. some of the features, we found `Year.Built` and `Year.Remod.Add` have non-linear relationships with price. We add polynomial terms on these two variables.
c. factorize continuous variables: some continuous variables are discrete with less than 20 levels and their relationships with price are not directly linear, such as `MS.SubClass Overall.Cond`, we decided to factorize those variables instead of keeping them as numeric values.

3. Interactions

Interactions between variables are also identified by plotting. One significant interaction is `TotalSq` between `House.Style`, which means that for different house style, the slope of the their linear relationship with price is different. We found that the interaction variable added to the `TotalSq` is a variable that can indicate the overall feature of a property. We tried to find some other variables that can also show the overall feature of a property and then added them as interaction terms. Such variables include `Heating.QC`,`Kitchen.Qual`, etc. We also identified some other combination of variables that may have interactions. Such combinations include `Lot.Area` vs. `Lot.shape` and `Garage.Type` vs. `Garage.Area`.

```
model1 = lm(log(price) ~  log(TotalSq)*(Neighborhood + House.Style +
                                          Exterior.2nd + Heating.QC + Kitchen.Qual) +
              Condition.1 + log(Lot.Area)*Lot.Shape +
              factor(Overall.Qual) + factor(Overall.Cond) + factor(MS.SubClass)+
              poly(Year.Built,3) + poly(Year.Remod.Add,2) +
              Bsmt.Exposure + Bsmt.Qual + Bsmt.Full.Bath + BsmtFin.Type.1 +
              BsmtFin.SF.1 +Bsmt.Unf.SF  +
              Garage.Type*(Garage.Area) + Garage.Finish +
              Exterior.1st + Foundation + Functional + Street +
              Bedroom.AbvGr + Full.Bath + Half.Bath + Kitchen.AbvGr +
              Fireplace.Qu + Fireplaces + Lot.Config + Fence +
              Wood.Deck.SF + log(Open.Porch.SF+1) + log(Screen.Porch+1) +
              Roof.Style + Mas.Vnr.Type + Mas.Vnr.Area, data=ames_train)
```

```
Yhat.test = predict(model1, newdata=ames_test, interval = "pred")
Yhat.test = exp(Yhat.test)
Yhat.train = predict(model1, newdata=ames_train, interval = "pred")
Yhat.train = exp(Yhat.train)
```

Model Summary

Summary table of the complex model:

```
summaries = summary(model1)
df = summaries$coefficients[c("(Intercept)","log(Lot.Area)",
                              "poly(Year.Built, 3)1","poly(Year.Built, 3)2",
                              "poly(Year.Built, 3)3","poly(Year.Remod.Add, 2)1",
                              "poly(Year.Remod.Add, 2)2",
                          "Bsmt.ExposureGd","Bsmt.ExposureMn",
                          "Bsmt.Full.Bath","Bsmt.Unf.SF",
                          "StreetPave","Full.Bath","Half.Bath",
                          "Wood.Deck.SF","log(Open.Porch.SF + 1)",
                          "log(Screen.Porch + 1)"),]
kable(df, caption = "Model Summary")
```

Table 2: Model Summary

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	6.9751254	1.6248670	4.292736	0.0000190
log(Lot.Area)	0.1080637	0.0127719	8.461056	0.0000000
poly(Year.Built, 3)1	3.0800839	0.3619127	8.510571	0.0000000
poly(Year.Built, 3)2	0.4556248	0.2180169	2.089860	0.0368335
poly(Year.Built, 3)3	0.4530837	0.1526354	2.968406	0.0030509
poly(Year.Remod.Add, 2)1	0.5470537	0.1490039	3.671405	0.0002515
poly(Year.Remod.Add, 2)2	-0.1785693	0.1296318	-1.377511	0.1686025
Bsmt.ExposureGd	0.0260984	0.0105107	2.483042	0.0131579
Bsmt.ExposureMn	-0.0309499	0.0103038	-3.003722	0.0027204
Bsmt.Full.Bath	0.0245874	0.0057283	4.292262	0.0000191
Bsmt.Unf.SF	0.0000256	0.0000110	2.324007	0.0202859
StreetPave	0.1262579	0.0399058	3.163899	0.0015945
Full.Bath	0.0227922	0.0074201	3.071690	0.0021748
Half.Bath	0.0201057	0.0071367	2.817217	0.0049210
Wood.Deck.SF	0.0000413	0.0000186	2.222491	0.0264299
log(Open.Porch.SF + 1)	0.0037990	0.0013047	2.911699	0.0036589
log(Screen.Porch + 1)	0.0062289	0.0016143	3.858446	0.0001200

We picked 41 features and fit a simple linear regression model. Among them we picked some whose p-value are most significant as the table shows above. We can conclude that area of the lot, the year to build, the year to renew, basement and bathroom conditions and etc. all play key roles in determining house prices.

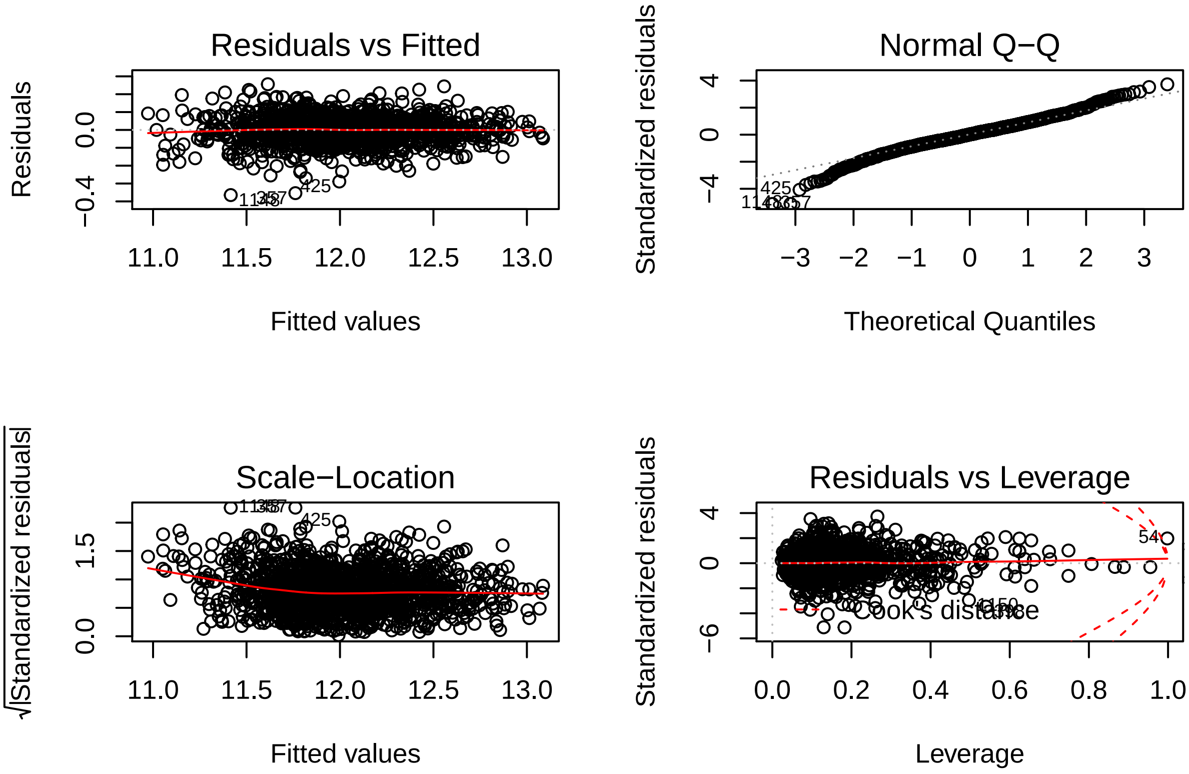
Furthermore, we investigated that the lot area and house prices are postively correlated: 10 percent increase in lot area will lead to $1.1^{6.975125} - 1$ (around 90 percent) increase in price. Moreover, we found out that bathroom conditions and area of wood deck, open porch and screen porch all have positive correlation with price. a 1-unit increase in the area of wood deck leads to $e^{0.1262579}$ (around 1.134575) increase in price. Last but not least, 10 percent increase in areas of open porch and screen porch will lead to $1.1^{0.0038} - 1$ (around 0.03 percent) and $1.1^{0.0062} - 1$ (around 0.05 percent) increase in house prices respectively.

Model Evaluation

1. Residual Diagnostics

For our complex model, residuals cloud around zero with no particular patterns. There are also no points with high leverage or large Cook's distance. The only issue is that we may still have heavy tails, indicating potential outliers.

```
par(mfrow = c(2,2)); plot(model1)
```



We also checked the model using the following criteria. The results of both the simple model and the complex model are shown in the table below. The performace of complex model greatly improves model accuracy.

```
rmse2 = rmse(Yhat.test[,'fit'], ames_test$price)
bias2 = bias(Yhat.test[,'fit'], ames_test$price)
maxDeviation2 = maxDeviation(Yhat.test[, 'fit'], ames_test$price)
meanDeviation2 = meanDeviation(Yhat.test[,'fit'], ames_test$price)
coverage2 = coverage(ames_test$price, Yhat.test[,2], Yhat.test[,3])

res = data.frame(rmse = c(rmse1, rmse2),
                 bias = c(bias1, bias2),
                 maxDeviation = c(maxDeviation1, maxDeviation2),
                 meanDeviation = c(meanDeviation1, meanDeviation2),
                 coverage = c(coverage1, coverage2)
                 )
rownames(res) = c("initial model", "complex model")
kable(res, caption = "Model Accuracy")
```

Table 3: Model Accuracy

	rmse	bias	maxDeviation	meanDeviation	coverage
initial model	15568.79	1338.081	71584.28	10970.80	1
complex model	13035.19	1080.726	53558.77	9655.01	1

2. Model Checking

We can do model checking by using the selected features and interactions, model coefficients and intercept. The first observation in the training data, the prediction is 176547.7, which is higher than the true value 137000. For the testing data, the prediction is 190388.6, which is also close to the true value 192100.

3. Model Results

Top 10 undervalued and overvalued houses are shown in the tables below. The most undervalued house is the one with parcel ID 528102010. We may invest in this house and sell it after the price rises. The most overvalued house is the one with parcel ID 905376090. We may sell this house now since its value may drop in the future.

```r
residual = Yhat.test[,1] - ames_test$price
ntest = dim(ames_test)[1]
nleft = ntest - 10
least_over = sort(residual, partial=nleft)[nleft]
id1 = which(residual > least_over)
df1 = data.frame(
  PID = ames_test$PID[id1],
  Predicted.Value = Yhat.test[id1],
  Real.Value = ames_test$price[id1],
  Difference = residual[id1]
)
kable(df1, caption = "Undervalued Houses")
```

Table 4: Undervalued Houses

	PID	Predicted.Value	Real.Value	Difference
84	528116010	322657.3	296000	26657.33
87	533251110	295323.1	255000	40323.07
88	528365090	317793.9	290000	27793.94
101	527355150	309397.7	278000	31397.66
190	528102010	361616.5	315000	46616.46
191	528174080	213002.7	185850	27152.66
355	528120120	312949.2	275000	37949.24
364	528120160	301876.2	274900	26976.16
480	527182020	159520.3	130000	29520.26
497	531380080	233520.4	205000	28520.37

```r
residual = ames_test$price - Yhat.test[,1]
ntest = dim(ames_test)[1]
nleft = ntest - 10
least_over = sort(residual, partial=nleft)[nleft]
id2 = which(residual > least_over)
df2 = data.frame(
  PID = ames_test$PID[id2],
  Predicted.Value = Yhat.test[id2],
  Real.Value = ames_test$price[id2],
  Difference = -residual[id2]
)
kable(df2, caption = "Overvalued Houses")
```

Table 5: Overvalued Houses

	PID	Predicted.Value	Real.Value	Difference
122	535382020	122785.6	160000	-37214.39
132	909275110	187662.3	238000	-50337.65
169	535454070	129410.4	166000	-36589.63
292	528358030	308357.1	350000	-41642.95

	PID	Predicted.Value	Real.Value	Difference
296	916226030	195840.7	241500	-45659.30
344	533206020	232205.0	280750	-48544.97
349	528178070	382157.7	421250	-39092.26
367	903429110	148217.9	179900	-31682.09
417	905427010	190985.2	235000	-44014.79
499	905376090	162441.2	216000	-53558.77

Conclusion

Our project aims to predict houses prices using the Ames data. Based on the explanations in the codebook, we imputed some variables with missing data by giving missing values reasonable meaning and eliminated outliers. After data cleaning, we first built a linear model with variables selected from forward selection. However, after further inverstigation into these variables, we found that some of them can be excluded in the first place. Highly correlated, serveraly imbalanced, and largely missing variables not only provide little information, but also cause damage to our model. Thus, after summarising and plotting variables, we handpicked 41 of them, excluding those problematic ones.

With these selected variables and their interactions, we built our complex model. We found that house prices are closely related to lot area, the year when the house was built and remodeled, the number of bathrooms, porch area, etc. For example, the larger the lot area, the higher the price. The more the full and half bathrooms, the higher the price. These significant variables help us to make better predictions of house prices. Our complex model does improve prediction accurarcy according to all the five evaluation criteria. With these predictions, we finally gave our suggestions on overvalued and undervalued hourses.

We actually learnt more than we expected when working on this project. We found that variable selection is extremely important when given a number of related and messy variables. Dealing with missing values and removing outliers are also crucial for a better prediction. We were also amazed at the power of ordinary linear models. In fact, in addition to ordinary linear models, we tried Bayesian linear models, panelized linear models, tree ensembles and GAM. The ordinary linear model outperform all these models because it is computationally cheap and also able to capture non-linearity and interactions.