Write Up

Yasong Zhou, Xichu Liu, Shaoji Li, Zhanhan Yu

4/27/2017

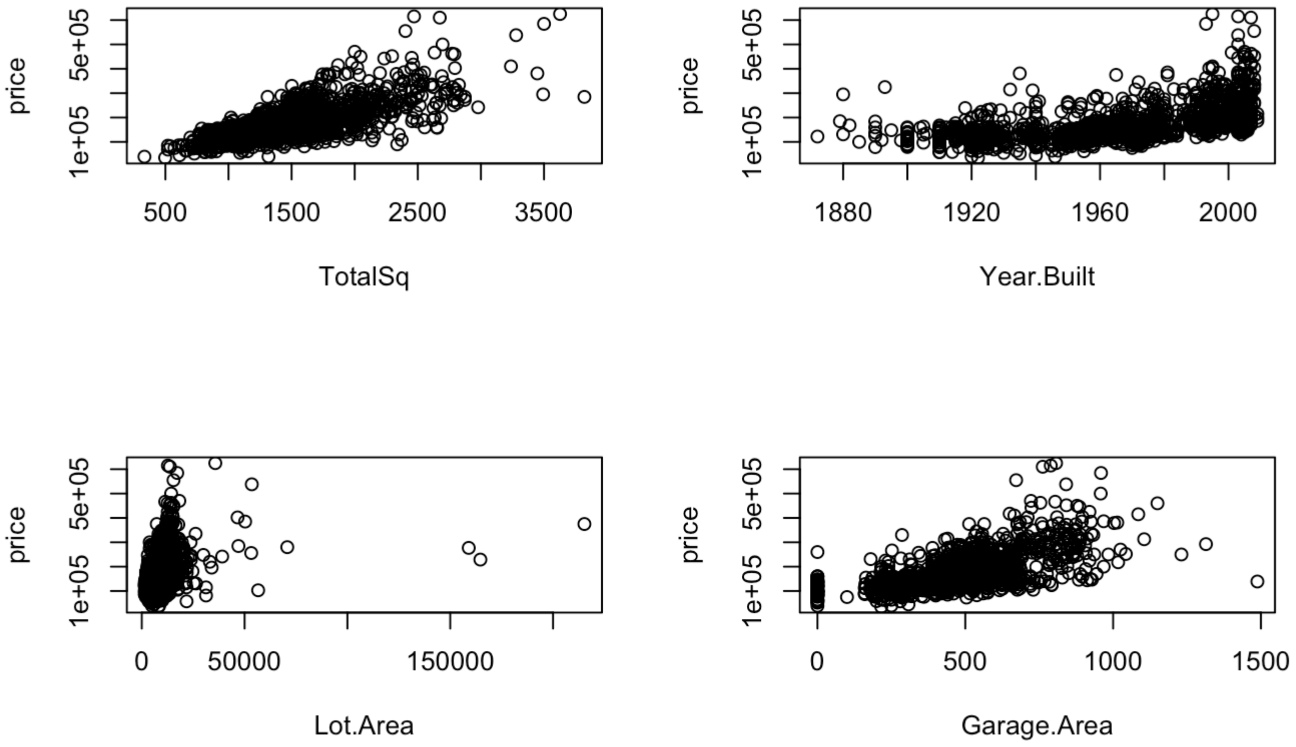
1. Exploratory data analysis (20 points): must include three correctly labeled graphs and an explanation that highlight the most important features that went into your model building.

1.1 Variables Transformation

- At the first glance of quantitive variables, we find some potential nonlinearity in `price` , `TotalSq` , and `Lot.Area` etc.
- Figure 1.1 shows the scatterplots of `price` , `TotalSq` , `Garage.Area` , `Lot.Area` and `Year.Build` . Figure 1.1 implies nonlinearity problem may exsits in variables `TotalSq` , `Lot.Area` , and `price` . Also, there may be some outliers exists in variable `Lot.Area` as shown in the third plot of Figure 1.1. But our intuition tells us that these three variables could be relative to house price. For example, total square of house and the year which the house built could definitely influence the house value. Taking into account the nonlinearity existing in the potential predictors, we consider using nonliear model to capture these features or we could transform these variables to alleviate the nonlinearity problem. As can be seen in the figure 1.2, the nonlinearity problem is alleviated after transformation.

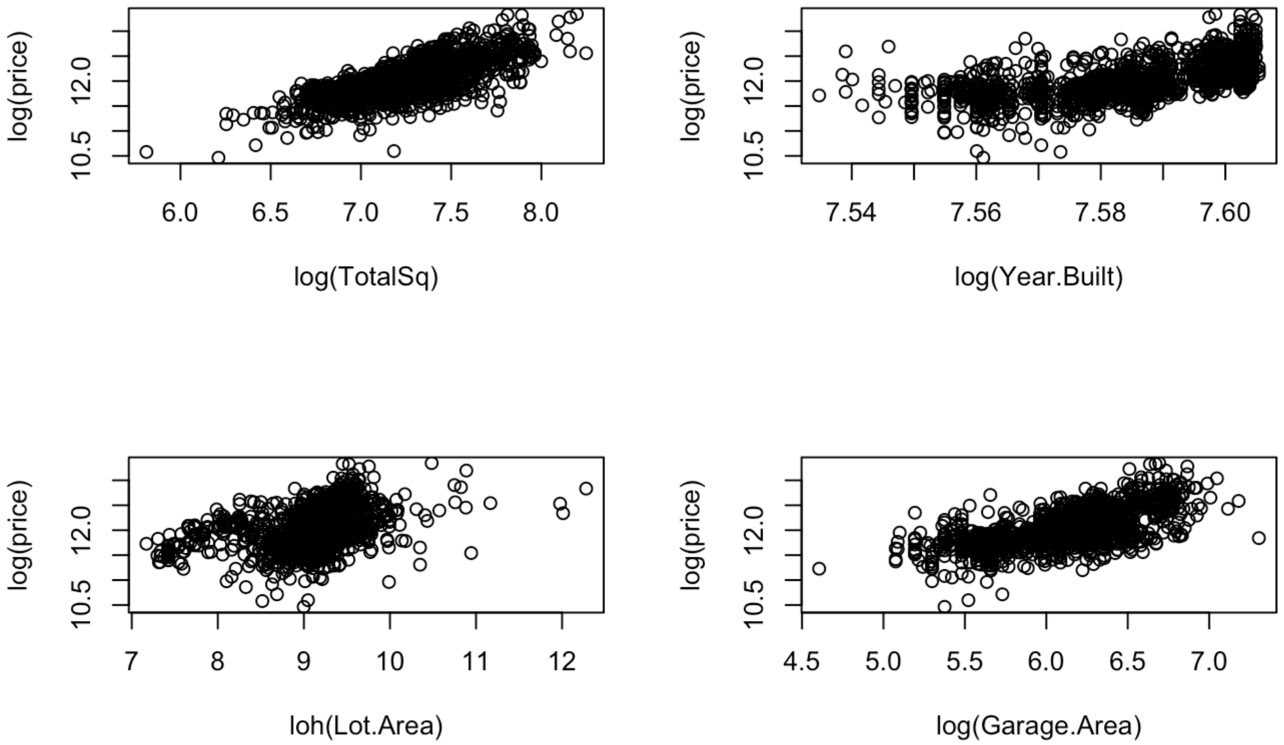
```
load("ames_train.Rdata")
par(mfrow = c(2,2),oma=c(0,0,2,0))
plot(ames_train$TotalSq, ames_train$price,xlab="TotalSq",ylab="price")
plot(ames_train$Year.Built, ames_train$price,xlab="Year.Built",ylab="price")
plot(ames_train$Lot.Area, ames_train$price,xlab="Lot.Area",ylab="price")
plot(ames_train$Garage.Area, ames_train$price,xlab="Garage.Area",ylab="price")
title("Figure 1.1 Scatterplots (before transformation", outer=TRUE)
```

Figure 1.1 Scatterplots (before transformation



```
par(mfrow = c(2,2),oma=c(0,0,2,0))
plot(log(ames_train$TotalSq), log(ames_train$price),xlab="log(TotalSq)",ylab="log(pric
e)")
plot(log(ames_train$Year.Built), log(ames_train$price),xlab="log(Year.Built)",ylab="log
(price)")
plot(log(ames_train$Lot.Area), log(ames_train$price),xlab="loh(Lot.Area)",ylab="log(pric
e)")
plot(log(ames_train$Garage.Area), log(ames_train$price),xlab="log(Garage.Area)",ylab="lo
g(price)")
title("Figure 1.2 Scatterplots (after transformation", outer=TRUE)
```

Figure 1.2 Scatterplots (after transformation



1.2 Inbalanced Factor Level and Missing Value

- Some qualitative variables contain a level named “NA” which do not represent missing data. For example, in variable `Bsmt.Cond` , NA means “No Basement” and in variable `Alley` , NA is “No alley access”. In this case, we replace NA with a new level to avoid unnecessary confusion. For example, we change the new level “No basement” in the variable `Bsmt.Qual` to replace with “NA”.

- Moreover, some quantitive variables, such as `Mas.Vnr.Area` , and `Lot.Frontage` contain missing value. We couldn’t claim that those missing data are at random. Simply excluding these missing data is reckless. Therefore, we use information from related observations or using mean imputation method to handle with missing data.

- We also find that the factor levels are not consistent among the training data, testing data and validation data. This issue prevents us from conducting prediction and model test. To figure out this issue, we do the imputation based on logical rules. For example, in the training and testing data set, level “Ex” is not included in the variable `Bsmt.Qual` , but it exist in validation data set. We merge levels “Ex” (Excellent) and “Gd” (Good) by replacing “Ex” with “Gd”(“Good”).

```
suppressMessages(library(dplyr))
load("ames_train.Rdata")

levels(ames_train$Bsmt.Cond) = c(levels(ames_train$Bsmt.Cond)[-1], "No Basement")
ames_train$Bsmt.Cond[is.na(ames_train$Bsmt.Cond)] = "No Basement"
levels(ames_train$Bsmt.Exposure) = c(levels(ames_train$Bsmt.Exposure)[-1], "No
Basement")
ames_train$Bsmt.Exposure[is.na(ames_train$Bsmt.Exposure)] = "No Basement"
levels(ames_train$Bsmt.Qual) = c(levels(ames_train$Bsmt.Qual)[-1], "No Basement")
ames_train$Bsmt.Qual[is.na(ames_train$Bsmt.Qual)] = "No Basement"
levels(ames_train$BsmtFin.Type.1) = c(levels(ames_train$BsmtFin.Type.1)[-1], "No Basemen
t")
ames_train$BsmtFin.Type.1[is.na(ames_train$BsmtFin.Type.1)] = "No Basement"
levels(ames_train$BsmtFin.Type.2) = c(levels(ames_train$BsmtFin.Type.2)[-1], "No Basemen
t")
ames_train$BsmtFin.Type.2[is.na(ames_train$BsmtFin.Type.2)] = "No Basement"

levels(ames_train$Alley) = c(levels(ames_train$Alley)[-1], "No alley access")
ames_train$Alley[is.na(ames_train$Alley)] = "No alley access"
levels(ames_train$Fireplace.Qu) = c(levels(ames_train$Fireplace.Qu)[-1], "No Fireplace")
ames_train$Fireplace.Qu[is.na(ames_train$Fireplace.Qu)] = "No Fireplace"
levels(ames_train$Garage.Type) = c(levels(ames_train$Garage.Type)[-1], "No Garage")
ames_train$Garage.Type[is.na(ames_train$Garage.Type)] = "No Garage"
levels(ames_train$Garage.Finish) = c(levels(ames_train$Garage.Finish)[-1], "No Garage")
ames_train$Garage.Finish[is.na(ames_train$Garage.Finish)] = "No Garage"

levels(ames_train$Garage.Qual) = c("Po","Fa","TA","Gd","Ex", "No Garage")
ames_train$Garage.Qual[is.na(ames_train$Garage.Qual)] = "No Garage"
levels(ames_train$Garage.Cond) = c(levels(ames_train$Garage.Cond)[-1], "No Garage")
ames_train$Garage.Cond[is.na(ames_train$Garage.Cond)] = "No Garage"
levels(ames_train$Pool.QC) = c(levels(ames_train$Pool.QC)[-1], "No Pool")
ames_train$Pool.QC[is.na(ames_train$Pool.QC)] = "No Pool"

levels(ames_train$Fence) = c(levels(ames_train$Fence)[-1], "No Fence")
ames_train$Fence[is.na(ames_train$Fence)] = "No Fence"
levels(ames_train$Misc.Feature) = c("None", levels(ames_train$Misc.Feature)[-1])
ames_train$Misc.Feature[is.na(ames_train$Misc.Feature)] = "None"

levels(ames_train$Garage.Yr.Blt) = c(levels(ames_train$Garage.Yr.Blt)[-1], "No Garage Ye
ar")
ames_train$Garage.Yr.Blt[is.na(ames_train$Garage.Yr.Blt)] = mean(ames_train$Garage.Yr.Bl
t[!is.na(ames_train$Garage.Yr.Blt)])

ames_train= ames_train %>%
  mutate(MS.SubClass = as.factor(MS.SubClass))
ames_train_new = ames_train%>% na.omit()

load("ames_test.Rdata")
ames_test$Lot.Frontage[is.na(ames_test$Lot.Frontage)]= 21 # mean of variables
ames_test$Mas.Vnr.Area[is.na(ames_test$Mas.Vnr.Area)]= 0
ames_test$Garage.Yr.Blt[is.na(ames_test$Garage.Yr.Blt)]= 1895

levels(ames_test$Bsmt.Cond) = c(levels(ames_test$Bsmt.Cond)[-1], "No Basement")
```

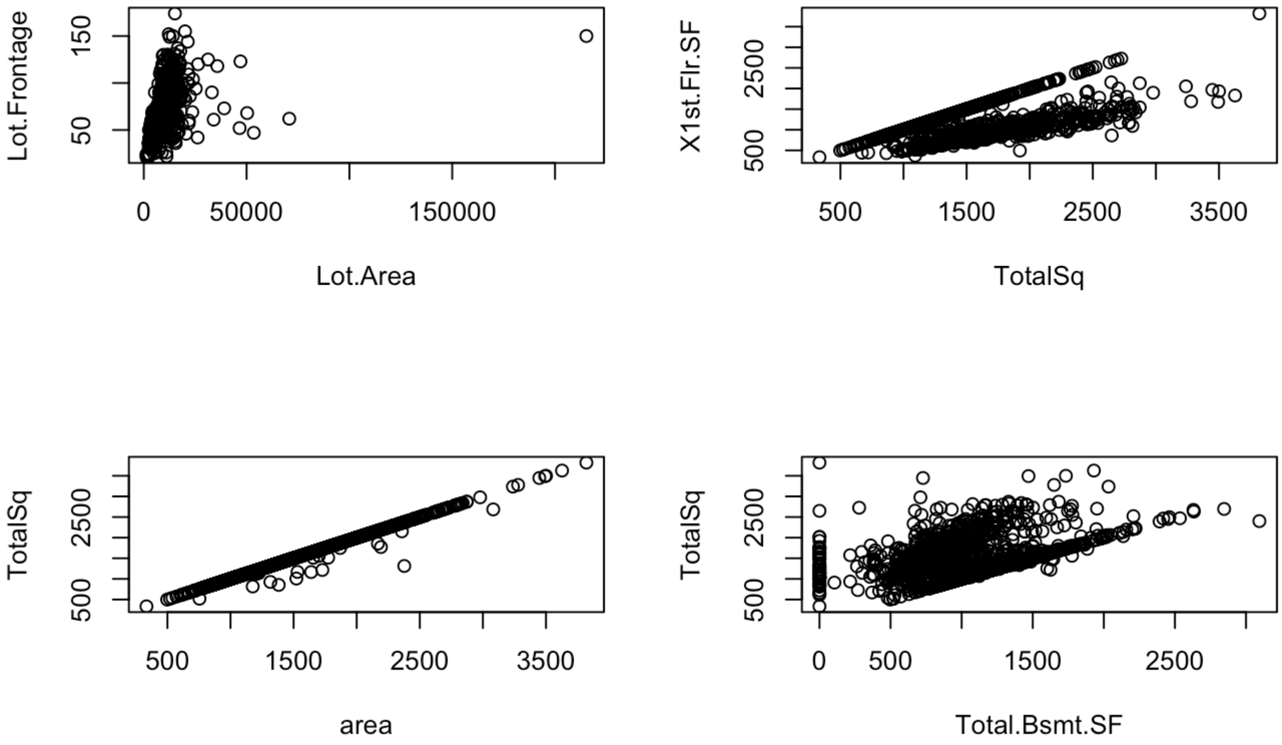
```
ames_test$Bsmt.Cond[is.na(ames_test$Bsmt.Cond)] = "No Basement"
levels(ames_test$Bsmt.Exposure) = c(levels(ames_test$Bsmt.Exposure)[-1], "No Basement")
ames_test$Bsmt.Exposure[is.na(ames_test$Bsmt.Exposure)] = "No Basement"
levels(ames_test$Bsmt.Qual) = c(levels(ames_test$Bsmt.Qual)[-1], "No Basement")
ames_test$Bsmt.Qual[is.na(ames_test$Bsmt.Qual)] = "No Basement"
levels(ames_test$BsmtFin.Type.1) = c(levels(ames_test$BsmtFin.Type.1)[-1], "No
Basement")
ames_test$BsmtFin.Type.1[is.na(ames_test$BsmtFin.Type.1)] = "No Basement"
levels(ames_test$BsmtFin.Type.2) = c(levels(ames_test$BsmtFin.Type.2)[-1], "No
Basement")
ames_test$BsmtFin.Type.2[is.na(ames_test$BsmtFin.Type.2)] = "No Basement"

levels(ames_test$Alley) = c(levels(ames_test$Alley)[-1], "No alley access")
ames_test$Alley[is.na(ames_test$Alley)] = "No alley access"
levels(ames_test$Fireplace.Qu) = c(levels(ames_test$Fireplace.Qu)[-1], "No Fireplace")
ames_test$Fireplace.Qu[is.na(ames_test$Fireplace.Qu)] = "No Fireplace"
levels(ames_test$Garage.Type) = c(levels(ames_test$Garage.Type)[-1], "No Garage")
ames_test$Garage.Type[is.na(ames_test$Garage.Type)] = "No Garage"
levels(ames_test$Garage.Finish) = c(levels(ames_test$Garage.Finish)[-1], "No Garage")
ames_test$Garage.Finish[is.na(ames_test$Garage.Finish)] = "No Garage"

levels(ames_test$Garage.Qual) = c("Po","Fa","TA","Gd","Ex", "No Garage")
ames_test$Garage.Qual[is.na(ames_test$Garage.Qual)] = "No Garage"
levels(ames_test$Garage.Cond) = c(levels(ames_test$Garage.Cond)[-1], "No Garage")
ames_test$Garage.Cond[is.na(ames_test$Garage.Cond)] = "No Garage"
levels(ames_test$Pool.QC) = c(levels(ames_test$Pool.QC)[-1], "No Pool")
ames_test$Pool.QC[is.na(ames_test$Pool.QC)] = "No Pool"

levels(ames_test$Fence) = c(levels(ames_test$Fence)[-1], "No Fence")
ames_test$Fence[is.na(ames_test$Fence)] = "No Fence"
levels(ames_test$Misc.Feature) = c("None", levels(ames_test$Misc.Feature)[-1])
ames_test$Misc.Feature[is.na(ames_test$Misc.Feature)] = "None"

levels(ames_test$Garage.Yr.Blt) = c("No Garage Year", levels(ames_test$Garage.Yr.Blt)
[-1])
ames_test$Garage.Yr.Blt[is.na(ames_test$Garage.Yr.Blt)] =
mean(ames_test$Garage.Yr.Blt[!is.na(ames_test$Garage.Yr.Blt)])

ames_test = ames_test %>%
  mutate(MS.SubClass = as.factor(MS.SubClass))
```

1.3 Multicollinearity

- It is worth to notice that multicollinearity exists in the variables. For example, `Lot.Frontage` and `Lot.Area`, `TotalSq` and `X1st.Flr.SF`, `area` and `TotalSq`, and `Total.Bsmt.SF` and `TotalSq` have strong linear relationship as shown in Figure 2. It is not a wise choice to include variables which may cause multicollinearity problem in the model.

```
par(mfrow = c(2,2),oma=c(0,0,2,0))
attach(ames_train)
plot(Lot.Area,Lot.Frontage)
plot(TotalSq,X1st.Flr.SF)
plot(area,TotalSq)
plot(Total.Bsmt.SF,TotalSq)
title("Figure 2. Multicollinearity between Variables", outer=TRUE)
```

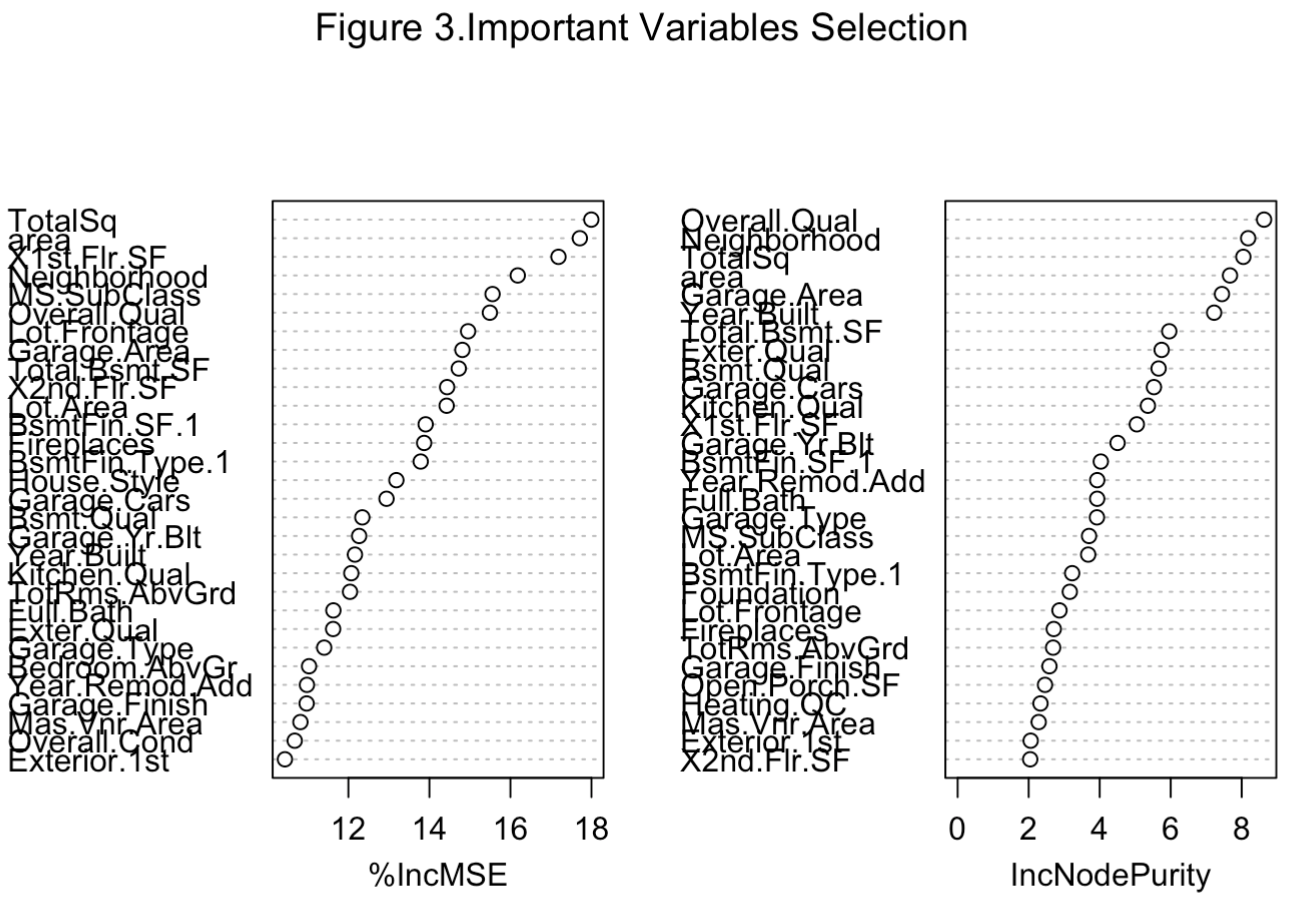
Figure 2. Multicollinearity between Variables



1.4 Variable Selection

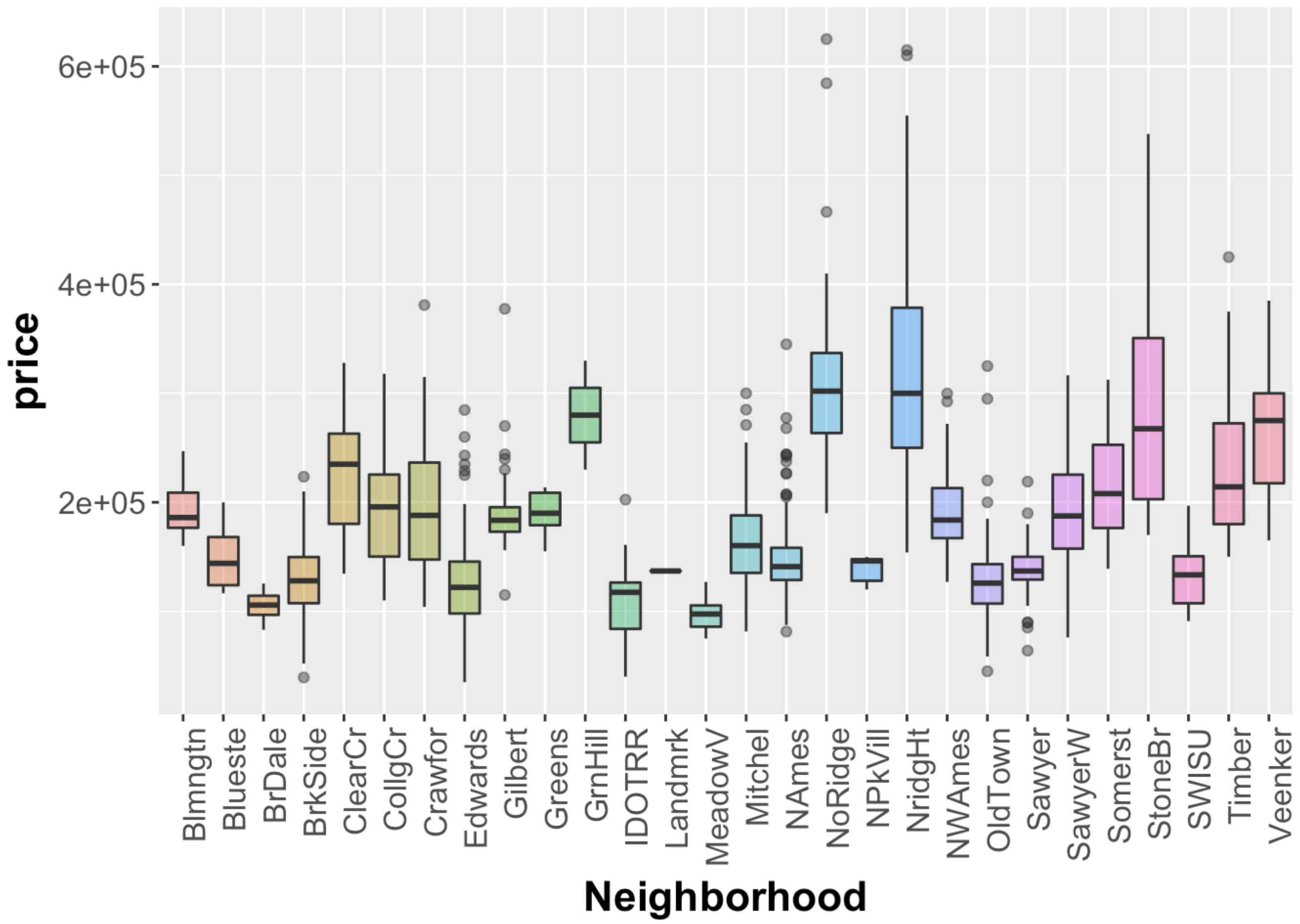
- After data cleaning, now we try to figure out which variables may be useful to predict house price. We use our intuition first, and then refer to Random Forest method to choose important variables.
- Figure 3. presents the result of random forest, 30 of 80 variables are important variabels which may have power to explain `price` . Variables in the Figure 3. confirm our intuition. For example, total square of the dwelling (`TotalSq`), the community the house belonging to (`Neighborhood`), and the overall quality of the material and finish of the house (`Overall.Qual`) could be the most powerful predictors. The type of dwelling (`MS.Subclass`) is also a significant factor that affect the house price, thus affecting the house price.

```
suppressMessages(library(randomForest))
suppressMessages(library(dplyr))
suppressMessages(attach(ames_train))
rf= randomForest(log(price) ~ . -PID,data=ames_train_new,
                            mtry=3, importance =TRUE)
varImpPlot(rf,main="Figure 3.Important Variables Selection",scale = T)
```



1.5 The Influence of Neighborhood on Price

```
suppressMessages(library(ggplot2))
ggplot(data = ames_train, aes(x = Neighborhood, y = price)) +
  geom_boxplot(data = ames_train, aes(fill = Neighborhood), alpha = 0.5) +
  guides(fill = FALSE)+
  theme_grey() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  theme(axis.title=element_text(size=16,face="bold"),
        axis.text = element_text(size=12),
        plot.title = element_text(size=20,face="bold", hjust = 0.5),
        legend.text = element_text(size = 12),
        legend.title = element_text(size = 14, face = "bold"))
```



2. Development and assessment of an initial model from Part I (10 points)

2.1 Initial model: must include a summary table and an explanation/discussion for variable selection. Interpretation of coefficients desirable for full points.

Summary Table

```
model1 = lm(log(price) ~ Neighborhood  + Exter.Qual+
    Heating + Central.Air + Bedroom.AbvGr + Functional + Garage.Cars +
    Paved.Drive + Wood.Deck.SF + House.Style + MS.SubClass +
    log(Year.Built) + sqrt(Lot.Area) + log(Year.Remod.Add) + Overall.Qual +
    Bsmt.Exposure + BsmtFin.Type.1 + Overall.Cond +
    log(TotalSq), data=ames_train)
s1 = summary(model1)
s1
```

```
## 
## Call:
## lm(formula = log(price) ~ Neighborhood + Exter.Qual + Heating + 
##     Central.Air + Bedroom.AbvGr + Functional + Garage.Cars + 
##     Paved.Drive + Wood.Deck.SF + House.Style + MS.SubClass + 
##     log(Year.Built) + sqrt(Lot.Area) + log(Year.Remod.Add) + 
##     Overall.Qual + Bsmt.Exposure + BsmtFin.Type.1 + Overall.Cond + 
##     log(TotalSq), data = ames_train)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.75361 -0.05047  0.00293  0.05306  0.40063 
## 
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)    
## (Intercept)              -4.703e+01  4.471e+00 -10.518  < 2e-16 ***
## NeighborhoodBlueste       3.295e-02  4.509e-02   0.731 0.464957    
## NeighborhoodBrDale        1.123e-02  4.049e-02   0.277 0.781606    
## NeighborhoodBrkSide       7.314e-02  3.495e-02   2.093 0.036548 *  
## NeighborhoodClearCr       9.671e-02  3.686e-02   2.624 0.008794 ** 
## NeighborhoodCollgCr       1.902e-02  3.041e-02   0.625 0.531880    
## NeighborhoodCrawfor       1.691e-01  3.431e-02   4.928 9.26e-07 ***
## NeighborhoodEdwards      -1.276e-02  3.257e-02  -0.392 0.695184    
## NeighborhoodGilbert      -1.105e-02  3.171e-02  -0.348 0.727610    
## NeighborhoodGreens        6.609e-02  4.597e-02   1.438 0.150726    
## NeighborhoodGrnHill       4.834e-01  7.296e-02   6.625 4.91e-11 ***
## NeighborhoodIDOTRR       -1.382e-02  3.639e-02  -0.380 0.704225    
## NeighborhoodLandmrk      -3.893e-03  1.015e-01  -0.038 0.969419    
## NeighborhoodMeadowV      -8.304e-02  4.727e-02  -1.757 0.079180 .  
## NeighborhoodMitchel       1.460e-02  3.215e-02   0.454 0.649760    
## NeighborhoodNAmes         1.718e-02  3.149e-02   0.546 0.585410    
## NeighborhoodNoRidge       1.272e-01  3.330e-02   3.821 0.000139 ***
## NeighborhoodNPkVill       5.479e-02  4.156e-02   1.318 0.187605    
## NeighborhoodNridgHt       1.124e-01  3.111e-02   3.611 0.000315 ***
## NeighborhoodNWAmes        2.528e-03  3.220e-02   0.079 0.937430    
## NeighborhoodOldTown       1.116e-02  3.405e-02   0.328 0.743123    
## NeighborhoodSawyer        3.229e-02  3.269e-02   0.988 0.323423    
## NeighborhoodSawyerW      -3.207e-03  3.164e-02  -0.101 0.919275    
## NeighborhoodSomerst       1.167e-01  3.100e-02   3.763 0.000175 ***
## NeighborhoodStoneBr       1.588e-01  3.307e-02   4.800 1.75e-06 ***
## NeighborhoodSWISU         6.275e-02  3.861e-02   1.625 0.104385    
## NeighborhoodTimber        1.040e-02  3.420e-02   0.304 0.761007    
## NeighborhoodVeenker       7.238e-02  3.992e-02   1.813 0.069998 .  
## Exter.QualFa             -1.300e-01  3.419e-02  -3.801 0.000150 ***
## Exter.QualGd             -1.058e-01  1.969e-02  -5.376 8.90e-08 ***
## Exter.QualTA             -1.114e-01  2.162e-02  -5.151 2.96e-07 ***
## HeatingGasA               1.215e-01  9.819e-02   1.238 0.216026    
## HeatingGasW               2.175e-01  1.014e-01   2.145 0.032109 *  
## HeatingGrav               5.197e-02  1.206e-01   0.431 0.666673    
## HeatingOthW               2.154e-02  1.212e-01   0.178 0.859023    
## HeatingWall               1.991e-01  1.077e-01   1.849 0.064642 .  
## Central.AirY              7.341e-02  1.348e-02   5.445 6.12e-08 ***
## Bedroom.AbvGr            -1.186e-02  4.764e-03  -2.490 0.012895 *  
## FunctionalMaj2            -2.100e-01  5.526e-02  -3.800 0.000151 ***
## FunctionalMin1            -1.061e-02  3.686e-02  -0.288 0.773484    
## FunctionalMin2             2.354e-02  3.670e-02   0.641 0.521361    
## FunctionalMod             -4.091e-02  3.976e-02  -1.029 0.303708    
## FunctionalTyp              6.388e-02  3.377e-02   1.891 0.058775 .  
## Garage.Cars                4.053e-02  4.918e-03   8.240 3.87e-16 ***
## Paved.DriveP               1.658e-02  1.954e-02   0.849 0.396224    
## Paved.DriveY               3.613e-02  1.229e-02   2.940 0.003334 ** 
## Wood.Deck.SF               4.912e-05  2.087e-05   2.354 0.018712 *  
## House.Style1.5Unf          2.935e-01  1.233e-01   2.381 0.017391 *  
## House.Style1Story          5.360e-02  3.264e-02   1.642 0.100790    
## House.Style2.5Fin          7.851e-02  9.000e-02   0.872 0.383194    
## House.Style2.5Unf          4.435e-02  6.669e-02   0.665 0.506155    
## House.Style2Story          2.419e-02  3.030e-02   0.798 0.424813    
## House.StyleSFoyer          1.376e-01  4.189e-02   3.285 0.001046 ** 
## House.StyleSLvl            7.196e-02  5.622e-02   1.280 0.200811    
## MS.SubClass30             -6.919e-03  1.712e-02  -0.404 0.686092    
## MS.SubClass40              6.514e-02  5.990e-02   1.087 0.277011    
## MS.SubClass45             -2.347e-01  1.256e-01  -1.868 0.061985 .  
## MS.SubClass50             -1.309e-02  3.451e-02  -0.379 0.704469    
## MS.SubClass60             -4.169e-02  2.754e-02  -1.514 0.130271    
## MS.SubClass70             -4.780e-02  3.053e-02  -1.566 0.117678    
## MS.SubClass75             -5.055e-02  6.691e-02  -0.756 0.450061    
## MS.SubClass80             -6.823e-02  5.324e-02  -1.282 0.200215    
## MS.SubClass85             -9.667e-02  3.686e-02  -2.623 0.008816 ** 
## MS.SubClass90             -1.462e-01  1.970e-02  -7.422 1.98e-13 ***
## MS.SubClass120            -5.251e-02  1.571e-02  -3.343 0.000851 ***
## MS.SubClass150            -2.143e-01  1.053e-01  -2.035 0.042055 *  
## MS.SubClass160            -1.681e-01  3.419e-02  -4.916 9.85e-07 ***
## MS.SubClass180            -1.608e-01  7.062e-02  -2.276 0.022974 *  
## MS.SubClass190            -5.865e-02  2.852e-02  -2.056 0.039942 *  
## log(Year.Built)            5.477e+00  5.307e-01  10.319  < 2e-16 ***
## sqrt(Lot.Area)             1.296e-03  1.336e-04   9.697  < 2e-16 ***
## log(Year.Remod.Add)        1.645e+00  3.832e-01   4.294 1.88e-05 ***
## Overall.Qual               5.628e-02  3.757e-03  14.980  < 2e-16 ***
## Bsmt.ExposureGd            9.303e-02  9.649e-02   0.964 0.335097    
## Bsmt.ExposureMn            1.414e-01  9.682e-02   1.460 0.144454    
## Bsmt.ExposureNo            4.762e-02  9.668e-02   0.493 0.622361    
## Bsmt.ExposureNo Basement   6.196e-02  9.636e-02   0.643 0.520348    
## BsmtFin.Type.1GLQ         -1.829e-02  1.032e-02  -1.773 0.076384 .  
## BsmtFin.Type.1LwQ          1.336e-02  9.247e-03   1.445 0.148790    
## BsmtFin.Type.1Rec         -3.865e-02  1.377e-02  -2.807 0.005062 ** 
## BsmtFin.Type.1Unf         -3.299e-02  1.059e-02  -3.115 0.001877 ** 
## BsmtFin.Type.1No Basement -7.420e-02  8.975e-03  -8.267 3.12e-16 ***
## Overall.Cond               3.691e-02  3.093e-03  11.932  < 2e-16 ***
## log(TotalSq)               5.551e-01  1.623e-02  34.191  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.0956 on 1416 degrees of freedom
## Multiple R-squared:  0.9375, Adjusted R-squared:  0.9339 
## F-statistic:   256 on 83 and 1416 DF,  p-value: < 2.2e-16
```

- In the summary table, 30 variables are included in model 1 and all of them are significant. The adjusted R square of model 1 is 0.9338541.
- In order to alleviate nonlinearity and heterogeneity problem, we take logarithm on price, Garage.Yr.Blt, and Year.Remod.Add, Year.Built, and area.

Coefficient Interpretation

- There are 19 predictors in this model, while most of them are statistically significant. This means the true `price` of a house is influenced by many factors. Variance in one of these predictors may cause variance in house price.
- We here take variable `log(TotalSq)` as an example for interpretation. The value of coefficient `log(TotalSq)` is 0.5550638, which means one unit increasing in `TotalSq` from A to (A + 1) will result in increasing of `price` to (A+1)/A times larger.

2.2 Model selection: must include a discussion

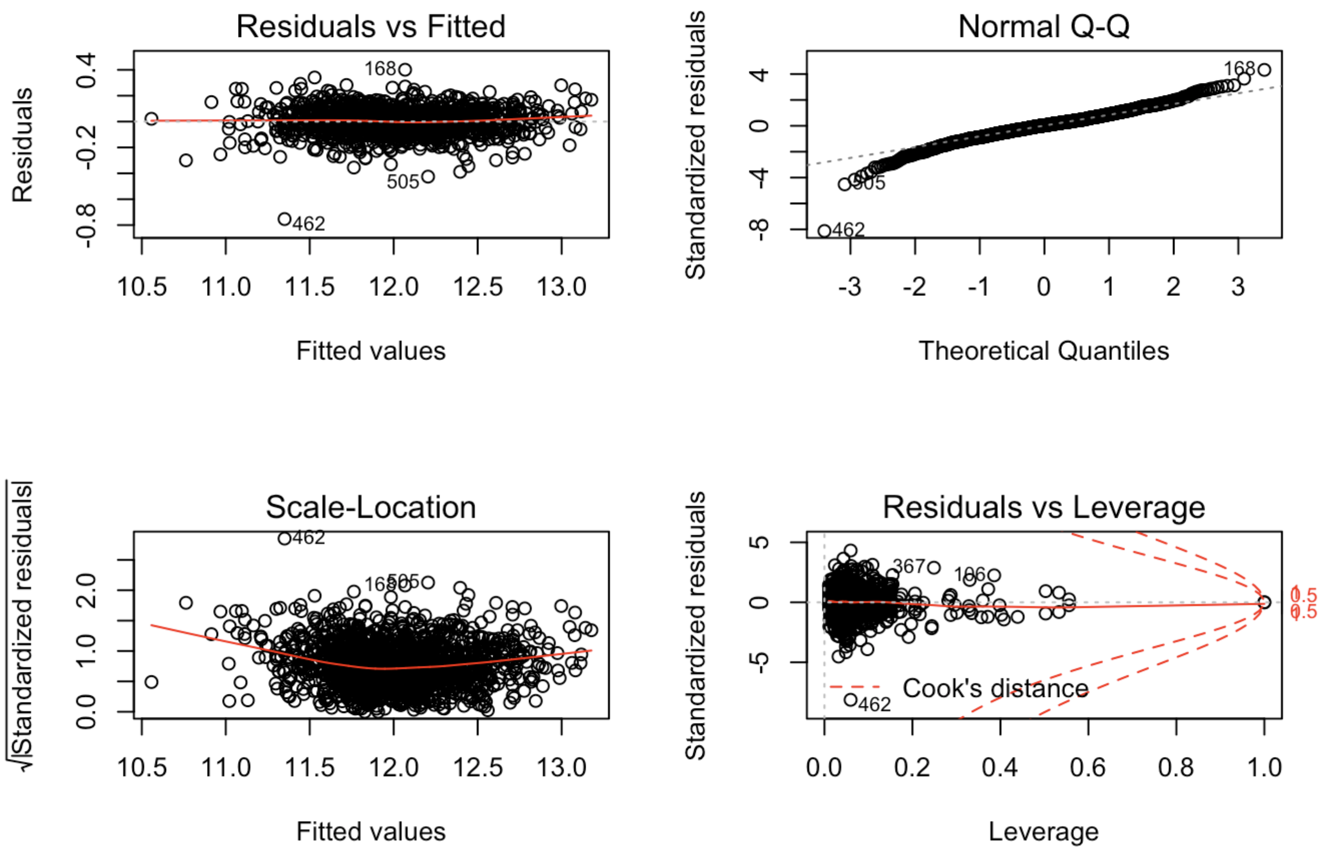
Model/Variable Selection

```
rf.var = data.frame(rf$importance)
rf.var$variable = rownames(rf.var)
rf.var = rf.var %>%
  arrange(desc(X.IncMSE))
```

- From Part I, we got 30 most important variables in term of MSE by RandomForest. The 30 variables are TotalSq, area, Overall.Qual, Year.Built, Neighborhood, Total.Bsmt.SF, Garage.Area, X1st.Flr.SF, Garage.Yr.Blt, Bsmt.Qual, Exter.Qual, Garage.Cars, Full.Bath, Kitchen.Qual, MS.SubClass, Garage.Type, Year.Remod.Add, Lot.Area, BsmtFin.SF.1, X2nd.Flr.SF, Fireplaces, Foundation, BsmtFin.Type.1, TotRms.AbvGrd, Lot.Frontage, Garage.Finish, Heating.QC, House.Style, Open.Porch.SF, Overall.Cond
- Start with the 30 variables, we used this as our full model, and then used AIC stepwise to conduct variable selection.
- We also used our intuition and common knowledge to select variables which might be correlated to price.
- Then we used correlation plot to identify those variables which have strong correlation with each other and just kept one of them.

2.3 Residual: must include a residual plot and a discussion

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



- According to diagnostic plots, model 1 may not have seriously nonliearity and heterogeneity.
- We do not identify any possible influential outliers whose cook’s distance larger than 1 in this model.
- There is a slightly left heavy tail in the Normality Plot, which indicates there may be some outliers in the data.

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

```
##                        outsample      insample
## RMSE                  16300.3000  1.793070e+04
## Bias                    503.0012 -8.421462e+02
## Coverage                  0.9520  9.566667e-01
## MeanAbsoluteDeviation 11943.8040  1.212294e+04
## MaximumDeviation      90692.2742  1.431245e+05
```

- The out of sample RMSE is 1.63003×10^4, bias is 503.0011885 and the coverage is 0.952. Notice that the in sample RMSE is 1.79307×10^4, which is smaller than out of sample RMSE. The large RMSE implies that the out of sample prediction of model 1 may not be good. But the bias of prediction is relatively small, which is -842.1462382. Small bias and large out of sample predicted variance follow the bias-variance trade off property.

2.5 Model testing: must include an explanation

Model Check for the training data

$$\begin{gathered}
\log price = -47.026 + 0.017 * NeighborhoodBrkSide - 0.111 * Exter.QualTA \\
+ 0.122 * HeatingGasA + 0.073 * Central.AirY \\
- 0.012 * Bedroom.AbvGr + 0.064 * FunctionalTyp + 0.041 * Garage.Cars \\
+ 0.036 * Paved.DriveN + 0.000 * Wood.Deck.SF + 0.054 * House.Style1.5Fin \\
- 0.048 * MS.SubClass70 + 5.477 * \log Year.Built + 0.001 * \sqrt{Lot.Area} \\
+ 1.645 * \log(Year.Remod.Add) + 0.056 * Overall.Qual \\
+ 0.048 * Bsmt.ExposureNoBasement - 0.074 * BsmtFin.Type.1NoBasement \\
+ 0.037 * Overall.Cond + 0.555 * \log TotalSq
\end{gathered}$$

$$\begin{gathered}
\hat{\log} price = -47.026 + 0.073 * 1 - 0.111 * 1 + 0.122 * 1 + 0.073 * 1 \\
- 0.012 * 3 + 0.064 * 1 + 0.041 * 1 + 0.000 * 1 \\
+ 0.000 * 244 + 0.000 * 1 - 0.048 * 20 + 5.477 * 7.565 + 0.001 * 70.427 \\
+ 1.645 * 7.592 + 0.056 * 5 + 0.062 * 1 - 0.074 * 1 + 0.037 * 7 + 0.555 * 7.411 \\
= 11.819
\end{gathered}$$

$$\hat{price} = \exp(11.819) = 135796.461$$

$$price = 137000$$

Model Check for the testing data

$$\begin{gathered}
\log price = -47.026 + 0.017 * NeighborhoodNAmes - 0.111 * Exter.QualTA \\
+ 0.122 * HeatingGasA + 0.073 * Central.AirY - 0.012 * Bedroom.AbvGr + 0.064 * FunctionalTyp + \\
0.041 * Garage.Cars + 0.036 * Paved.DriveY + 0.000 * Wood.Deck.SF + 0.054 * House.Style1STORY \\
+ 0.000 * MS.SubClass20 + 5.477 * \log Year.Built + 0.001 * \sqrt{Lot.Area} \\
+ 1.645 * \log(Year.Remod.Add) + 0.056 * Overall.Qual \\
+ 0.048 * Bsmt.ExposureNo - 0.074 * BsmtFin.Type.1NoBasement \\
+ 0.037 * Overall.Cond + 0.555 * \log TotalSq
\end{gathered}$$

$$\begin{gathered}
\log \hat{price} = -47.026 + 0.017 * 1 - 0.111 * 1 \\
+ 0.122 * 1 + 0.073 * 1 - 0.012 * 3 + 0.064 * 1 + 0.041 * 2 \\
+ 0.036 * 1 + 0.000 * 0 + 0.054 * 1 + 0.000 * 20 \\
+ 5.477 * 7.585 + 0.001 * 108.291 + 1.645 * 7.585 \\
+ 0.056 * 7 + 0.048 * 1 - 0.074 * 1 + 0.037 * 6 \\
+ 0.555 * 7.523 = 12.20
\end{gathered}$$

$$\hat{price} = \exp(12.20) = 198910.63$$

$$price = 192100$$

- According to the manually model check of the first observation of training and testing data we conduct in this part,the prediction in model 1 performs well.

ANOVA Test

```
model0 = lm(log(price) ~ log(area) + MS.SubClass +
    Lot.Area + log(Year.Built) +
    log(Year.Remod.Add) + BsmtFin.SF.1 + BsmtFin.SF.2 +
    X1st.Flr.SF + X2nd.Flr.SF + Misc.Val + Yr.Sold, data=ames_train)
anova(model1, model0)
```

```
## Analysis of Variance Table
##
## Model 1: log(price) ~ Neighborhood + Exter.Qual + Heating + Central.Air +
##     Bedroom.AbvGr + Functional + Garage.Cars + Paved.Drive +
##     Wood.Deck.SF + House.Style + MS.SubClass + log(Year.Built) +
##     sqrt(Lot.Area) + log(Year.Remod.Add) + Overall.Qual + Bsmt.Exposure +
##     BsmtFin.Type.1 + Overall.Cond + log(TotalSq)
## Model 2: log(price) ~ log(area) + MS.SubClass + Lot.Area + log(Year.Built) +
##     log(Year.Remod.Add) + BsmtFin.SF.1 + BsmtFin.SF.2 + X1st.Flr.SF +
##     X2nd.Flr.SF + Misc.Val + Yr.Sold
##   Res.Df    RSS  Df Sum of Sq      F    Pr(>F)
## 1   1416 12.941
## 2   1474 29.794 -58   -16.852 31.791 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Here, we use ANOVA to test model 1. First, we run a model 0 including 11 randomly chose variables. From ANOVA test, the model 1 is significantly better than model 0.

3. Development of the final model (20 points)

3.1 Final model: must include a summary table

- We used Negative Binomial Generalized Linear Model which uses a log link function. This choice is consistent with the log transformation on the response variable, ‘price’.
- Also, we compared it with other regression models (omitted in the notebook) such as Random Forests, Decision Tree, Boosting and GAM. It performed better than those models as it produced a smaller RMSE.
- Here’s a summary of our complex model.

```
suppressMessages(library(MASS))
model2 = glm.nb(price ~ Neighborhood + Utilities + Exter.Qual + Foundation +
    Heating + Central.Air + Bedroom.AbvGr + Functional + Garage.Cars +
    Paved.Drive + House.Style + MS.Zoning + factor(MS.SubClass) +
    log(Year.Built) + log(Lot.Area) + log(Year.Remod.Add) + Overall.Qual +
    Bsmt.Exposure + Bsmt.Qual + Overall.Cond + BsmtFin.Type.1 +
    sqrt(TotalSq) + TotalSq:Neighborhood + TotalSq:Overall.Cond + TotalSq:Exterior.1st,
data = ames_train)
summary(model2)
```

```
##
## Call:
## glm.nb(formula = price ~ Neighborhood + Utilities + Exter.Qual +
##     Foundation + Heating + Central.Air + Bedroom.AbvGr + Functional +
##     Garage.Cars + Paved.Drive + House.Style + MS.Zoning + factor(MS.SubClass) +
##     log(Year.Built) + log(Lot.Area) + log(Year.Remod.Add) + Overall.Qual +
##     Bsmt.Exposure + Bsmt.Qual + Overall.Cond + BsmtFin.Type.1 +
##     sqrt(TotalSq) + TotalSq:Neighborhood + TotalSq:Overall.Cond +
##     TotalSq:Exterior.1st, data = ames_train, init.theta = 142.8002142,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.2721  -0.5829   0.0000   0.5780   3.6962
##
## Coefficients: (2 not defined because of singularities)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -4.043e+01  4.521e+00  -8.943  < 2e-16 ***
## NeighborhoodBlueste          2.487e-01  3.055e-01   0.814 0.415557
## NeighborhoodBrDale           3.500e-03  2.685e-01   0.013 0.989600
## NeighborhoodBrkSide          1.691e-01  2.288e-01   0.739 0.459962
## NeighborhoodClearCr          3.806e-01  2.380e-01   1.599 0.109813
## NeighborhoodCollgCr          1.158e-01  2.253e-01   0.514 0.607360
## NeighborhoodCrawfor          3.017e-01  2.301e-01   1.311 0.189758
## NeighborhoodEdwards          1.353e-01  2.259e-01   0.599 0.549318
## NeighborhoodGilbert          1.305e-01  2.308e-01   0.565 0.571816
## NeighborhoodGreens           1.984e-01  3.162e-01   0.627 0.530355
## NeighborhoodGrnHill         -5.722e-01  8.424e-01  -0.679 0.496950
## NeighborhoodIDOTRR           3.530e-01  2.307e-01   1.530 0.126007
## NeighborhoodLandmrk         -1.596e-01  2.611e-01  -0.611 0.541037
## NeighborhoodMeadowV         -8.479e-02  2.904e-01  -0.292 0.770278
## NeighborhoodMitchel          1.757e-01  2.277e-01   0.771 0.440432
## NeighborhoodNAmes            2.100e-01  2.251e-01   0.933 0.350845
## NeighborhoodNoRidge          4.654e-02  2.401e-01   0.194 0.846328
## NeighborhoodNPkVill          3.298e-02  2.596e-01   0.127 0.898920
## NeighborhoodNridgHt          5.278e-02  2.295e-01   0.230 0.818105
## NeighborhoodNWAmes           1.722e-01  2.277e-01   0.756 0.449513
## NeighborhoodOldTown          2.434e-01  2.265e-01   1.074 0.282626
## NeighborhoodSawyer           2.492e-01  2.259e-01   1.103 0.269965
## NeighborhoodSawyerW          4.236e-02  2.271e-01   0.187 0.852038
## NeighborhoodSomerst          1.781e-01  2.344e-01   0.760 0.447351
## NeighborhoodStoneBr          8.390e-02  2.324e-01   0.361 0.718050
## NeighborhoodSWISU            1.932e-01  2.336e-01   0.827 0.408222
## NeighborhoodTimber           1.053e-01  2.350e-01   0.448 0.654166
## NeighborhoodVeenker          7.774e-02  2.410e-01   0.323 0.746969
## UtilitiesNoSewr             -2.425e-01  7.569e-02  -3.204 0.001355 **
## Exter.QualFa                -8.583e-02  3.143e-02  -2.731 0.006309 **
## Exter.QualGd                -6.868e-02  1.857e-02  -3.698 0.000218 ***
## Exter.QualTA                -7.460e-02  2.021e-02  -3.692 0.000223 ***
## FoundationCBlock             3.635e-03  1.117e-02   0.325 0.744865
## FoundationPConc              1.478e-02  1.195e-02   1.237 0.216203
## FoundationSlab              -9.078e-02  2.115e-02  -4.292 1.77e-05 ***
## FoundationStone             -4.795e-02  3.797e-02  -1.263 0.206639
## FoundationWood              -8.857e-02  5.242e-02  -1.690 0.091089 .  
## HeatingGasA                  1.809e-01  8.827e-02   2.050 0.040396 *  
## HeatingGasW                  2.536e-01  9.122e-02   2.781 0.005426 ** 
## HeatingGrav                  7.672e-02  1.080e-01   0.711 0.477308    
## HeatingOthW                  1.066e-01  1.093e-01   0.976 0.329074    
## HeatingWall                  2.847e-01  9.854e-02   2.889 0.003865 ** 
## Central.AirY                 6.331e-02  1.250e-02   5.065 4.09e-07 ***
## Bedroom.AbvGr               -9.500e-03  4.392e-03  -2.163 0.030550 *  
## FunctionalMaj2              -1.869e-01  4.973e-02  -3.760 0.000170 ***
## FunctionalMin1               2.192e-02  3.368e-02   0.651 0.515081    
## FunctionalMin2               5.403e-02  3.395e-02   1.591 0.111522    
## FunctionalMod               -1.547e-02  3.668e-02  -0.422 0.673131    
## FunctionalTyp                7.821e-02  3.094e-02   2.528 0.011466 *  
## Garage.Cars                  3.690e-02  4.515e-03   8.172 3.03e-16 ***
## Paved.DriveP                 1.659e-02  1.761e-02   0.942 0.346326    
## Paved.DriveY                 3.966e-02  1.127e-02   3.519 0.000432 ***
## House.Style1.5Unf            3.231e-01  1.095e-01   2.951 0.003164 ** 
## House.Style1Story            6.417e-02  2.907e-02   2.207 0.027282 *  
## House.Style2.5Fin            4.935e-02  8.978e-02   0.550 0.582565    
## House.Style2.5Unf            5.720e-02  6.559e-02   0.872 0.383123    
## House.Style2Story            2.058e-02  2.773e-02   0.742 0.457928    
## House.StyleSFoyer            1.411e-01  3.800e-02   3.714 0.000204 ***
## House.StyleSLvl              8.453e-02  5.033e-02   1.680 0.093046 .  
## MS.ZoningC (all)             4.375e-03  1.009e-01   0.043 0.965428    
## MS.ZoningFV                  2.635e-01  9.642e-02   2.733 0.006277 ** 
## MS.ZoningI (all)             1.590e-01  1.252e-01   1.270 0.204187    
## MS.ZoningRH                  1.986e-01  9.623e-02   2.064 0.039039 *  
## MS.ZoningRL                  2.353e-01  9.206e-02   2.556 0.010592 *  
## MS.ZoningRM                  1.702e-01  9.363e-02   1.818 0.069066 .  
## factor(MS.SubClass)30       -4.381e-02  1.658e-02  -2.642 0.008246 ** 
## factor(MS.SubClass)40        4.776e-02  5.325e-02   0.897 0.369728    
## factor(MS.SubClass)45       -2.599e-01  1.116e-01  -2.328 0.019905 *  
## factor(MS.SubClass)50       -1.023e-02  3.079e-02  -0.332 0.739781    
## factor(MS.SubClass)60       -3.637e-02  2.583e-02  -1.408 0.159045    
## factor(MS.SubClass)70       -3.679e-02  2.807e-02  -1.310 0.190057    
## factor(MS.SubClass)75       -3.405e-02  6.333e-02  -0.538 0.590805    
## factor(MS.SubClass)80       -5.950e-02  4.763e-02  -1.249 0.211623    
## factor(MS.SubClass)85       -7.609e-02  3.312e-02  -2.297 0.021623 *  
## factor(MS.SubClass)90       -1.287e-01  1.854e-02  -6.943 3.84e-12 ***
## factor(MS.SubClass)120       2.812e-02  1.710e-02   1.644 0.100118    
## factor(MS.SubClass)150      -1.074e-01  9.427e-02  -1.140 0.254423    
## factor(MS.SubClass)160      -7.853e-02  3.396e-02  -2.313 0.020742 *  
## factor(MS.SubClass)180      -1.678e-02  7.751e-02  -0.216 0.828613    
## factor(MS.SubClass)190      -5.597e-02  2.546e-02  -2.198 0.027921 *  
## log(Year.Built)              4.967e+00  5.351e-01   9.282  < 2e-16 ***
## log(Lot.Area)                1.024e-01  8.581e-03  11.938  < 2e-16 ***
## log(Year.Remod.Add)          1.471e+00  3.508e-01   4.194 2.75e-05 ***
## Overall.Qual                 5.289e-02  3.573e-03  14.801  < 2e-16 ***
## Bsmt.ExposureGd              1.077e-01  8.466e-02   1.272 0.203203    
## Bsmt.ExposureMn              1.465e-01  8.498e-02   1.724 0.084749 .  
## Bsmt.ExposureNo              6.172e-02  8.480e-02   0.728 0.466713    
## Bsmt.ExposureNo Basement     8.268e-02  8.452e-02   0.978 0.327964    
## Bsmt.QualGd                 -1.874e-02  2.102e-02  -0.892 0.372638    
## Bsmt.QualPo                 -5.359e-02  1.222e-02  -4.384 1.17e-05 ***
## Bsmt.QualTA                  5.818e-03  9.430e-02   0.062 0.950807    
## Bsmt.QualNo Basement        -5.099e-02  1.481e-02  -3.443 0.000575 ***
## Overall.Cond                 3.430e-02  6.949e-03   4.936 7.95e-07 ***
## BsmtFin.Type.1GLQ           -1.728e-02  9.130e-03  -1.893 0.058371 .  
## BsmtFin.Type.1LwQ            1.150e-02  8.361e-03   1.375 0.169065    
## BsmtFin.Type.1Rec           -3.811e-02  1.233e-02  -3.090 0.002003 ** 
## BsmtFin.Type.1Unf           -3.517e-02  9.505e-03  -3.701 0.000215 ***
## BsmtFin.Type.1No Basement   -6.997e-02  8.164e-03  -8.570  < 2e-16 ***
## sqrt(TotalSq)                4.397e-02  5.049e-03   8.709  < 2e-16 ***
## NeighborhoodBlmngtn:TotalSq -1.260e-04  1.724e-04  -0.731 0.464980    
## NeighborhoodBlueste:TotalSq -2.173e-04  1.949e-04  -1.115 0.264842    
## NeighborhoodBrDale:TotalSq  -1.381e-05  1.554e-04  -0.089 0.929202    
## NeighborhoodBrkSide:TotalSq -1.645e-04  8.726e-05  -1.885 0.059464 .  
## NeighborhoodClearCr:TotalSq -3.119e-04  8.455e-05  -3.689 0.000225 ***
## NeighborhoodCollgCr:TotalSq -1.960e-04  7.755e-05  -2.527 0.011494 *  
## NeighborhoodCrawfor:TotalSq -2.367e-04  7.722e-05  -3.066 0.002172 ** 
## NeighborhoodEdwards:TotalSq -2.386e-04  7.724e-05  -3.090 0.002005 ** 
## NeighborhoodGilbert:TotalSq -2.221e-04  7.811e-05  -2.844 0.004459 ** 
## NeighborhoodGreens:TotalSq  -2.206e-04  2.116e-04  -1.043 0.297142    
## NeighborhoodGrnHill:TotalSq  6.025e-04  5.831e-04   1.033 0.301486    
## NeighborhoodIDOTRR:TotalSq  -3.396e-04  8.643e-05  -3.930 8.51e-05 ***
## NeighborhoodLandmrk:TotalSq         NA         NA      NA       NA    
## NeighborhoodMeadowV:TotalSq -5.481e-05  1.782e-04  -0.308 0.758363    
## NeighborhoodMitchel:TotalSq -2.449e-04  8.375e-05  -2.925 0.003448 ** 
## NeighborhoodNAmes:TotalSq   -2.791e-04  7.691e-05  -3.629 0.000285 ***
## NeighborhoodNoRidge:TotalSq -1.375e-04  6.703e-05  -2.051 0.040274 *  
## NeighborhoodNPkVill:TotalSq -1.028e-04  1.359e-04  -0.757 0.449258    
## NeighborhoodNridgHt:TotalSq -1.343e-04  7.077e-05  -1.897 0.057806 .  
## NeighborhoodNWAmes:TotalSq  -2.489e-04  7.513e-05  -3.313 0.000924 ***
## NeighborhoodOldTown:TotalSq -2.612e-04  7.785e-05  -3.355 0.000794 ***
## NeighborhoodSawyer:TotalSq  -2.996e-04  7.950e-05  -3.769 0.000164 ***
## NeighborhoodSawyerW:TotalSq -1.711e-04  7.373e-05  -2.320 0.020320 *  
## NeighborhoodSomerst:TotalSq -1.991e-04  8.206e-05  -2.426 0.015246 *  
## NeighborhoodStoneBr:TotalSq -1.327e-04  7.425e-05  -1.788 0.073824 .  
## NeighborhoodSWISU:TotalSq   -2.213e-04  8.660e-05  -2.555 0.010607 *  
## NeighborhoodTimber:TotalSq  -1.944e-04  8.082e-05  -2.406 0.016139 *  
## NeighborhoodVeenker:TotalSq -1.590e-04  8.229e-05  -1.932 0.053400 .  
## Overall.Cond:TotalSq         2.265e-06  4.626e-06   0.490 0.624422    
## TotalSq:Exterior.1stAsphShn -7.843e-05  5.406e-05  -1.451 0.146846    
## TotalSq:Exterior.1stBrkComm  4.291e-05  3.783e-05   1.134 0.256618    
## TotalSq:Exterior.1stBrkFace  5.791e-05  1.588e-05   3.646 0.000267 ***
## TotalSq:Exterior.1stCBlock          NA         NA      NA       NA    
## TotalSq:Exterior.1stCemntBd  2.985e-05  1.575e-05   1.895 0.058086 .  
## TotalSq:Exterior.1stHdBoard  1.349e-05  1.415e-05   0.953 0.340433    
## TotalSq:Exterior.1stImStucc  4.554e-05  5.675e-05   0.803 0.422252    
## TotalSq:Exterior.1stMetalSd  3.128e-05  1.370e-05   2.284 0.022388 *  
## TotalSq:Exterior.1stPlywood  2.724e-05  1.479e-05   1.841 0.065570 .  
## TotalSq:Exterior.1stPreCast  2.943e-04  5.889e-05   4.997 5.82e-07 ***
## TotalSq:Exterior.1stStucco   4.409e-05  1.822e-05   2.420 0.015532 *  
## TotalSq:Exterior.1stVinylSd  2.528e-05  1.402e-05   1.803 0.071367 .  
## TotalSq:Exterior.1stWd Sdng  2.096e-05  1.373e-05   1.527 0.126724    
## TotalSq:Exterior.1stWdShing  2.003e-05  1.741e-05   1.151 0.249858    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for Negative Binomial(142.8002) family taken to be 1)
##
##     Null deviance: 30468.7  on 1499  degrees of freedom
## Residual deviance:  1501.9  on 1360  degrees of freedom
## AIC: 33132
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  142.80
##           Std. Err.:  5.21
##
##  2 x log-likelihood:  -32850.22
```

3.2 Variables: must include an explanation

- The procedure of variable selection is very similar to what we did in Part I. We used the variables chosen from Part I with some additional interaction terms.
- For interactions, we found “Neighborhood”, “Overall.Cond” and “Exterior.1st” had strong interaction effect on “TotalSq”. The model comparison test (Chi-squared Test testing on deviance) confirms that the model with interaction is significant.

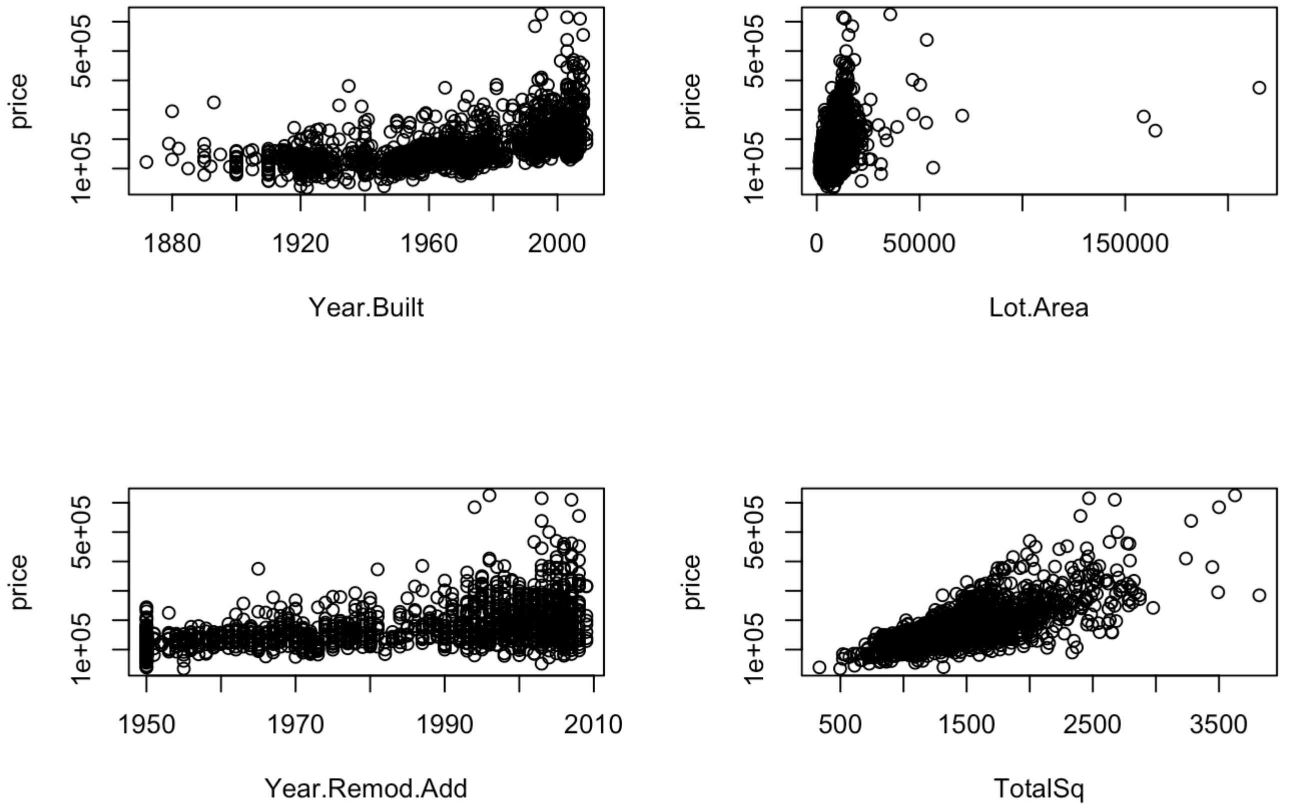
```
model.noInt = glm.nb(price ~ Neighborhood + Utilities + Exter.Qual + Foundation + Heatin
g + Central.Air + Bedroom.AbvGr + Functional + Garage.Cars + Paved.Drive + House.Style +
 MS.Zoning + factor(MS.SubClass) + log(Year.Built) + log(Lot.Area) + log(Year.Remod.Add)
 + Overall.Qual + Bsmt.Exposure + Bsmt.Qual + Overall.Cond + BsmtFin.Type.1 + sqrt(Total
Sq), data = ames_train)
anova(model.noInt, model2, test = "Chisq")
```

```
## Likelihood ratio tests of Negative Binomial Models
##
## Response: price
##




                                           Model
## 1                                                                      Neighborhood +
Utilities + Exter.Qual + Foundation + Heating + Central.Air + Bedroom.AbvGr + Functional
+ Garage.Cars + Paved.Drive + House.Style + MS.Zoning + factor(MS.SubClass) + log(Year.B
uilt) + log(Lot.Area) + log(Year.Remod.Add) + Overall.Qual + Bsmt.Exposure + Bsmt.Qual +
Overall.Cond + BsmtFin.Type.1 + sqrt(TotalSq)
## 2 Neighborhood + Utilities + Exter.Qual + Foundation + Heating + Central.Air + Bedroo
m.AbvGr + Functional + Garage.Cars + Paved.Drive + House.Style + MS.Zoning + factor(MS.S
ubClass) + log(Year.Built) + log(Lot.Area) + log(Year.Remod.Add) + Overall.Qual + Bsmt.E
xposure + Bsmt.Qual + Overall.Cond + BsmtFin.Type.1 + sqrt(TotalSq) + TotalSq:Neighborho
od + TotalSq:Overall.Cond + TotalSq:Exterior.1st
##      theta Resid. df    2 x log-lik.   Test    df LR stat.      Pr(Chi)
## 1 130.7626      1401       -32982.49
## 2 142.8002      1360       -32850.22 1 vs 2    41 132.2729 1.552614e-11
```

3.3 Variable selection/shrinkage: must use appropriate method and include an explanation

- We transformed some numeric variables according to the scatterplots. According to the scatter plots, a log transformation should be taken on “Year.Built”, “Lot.Area” and “Year.Remod.Add” since they had exponential relationship with the response. And, a square root transformation was needed for “TotalSq” since it had a quadratic relationship with the response variable.

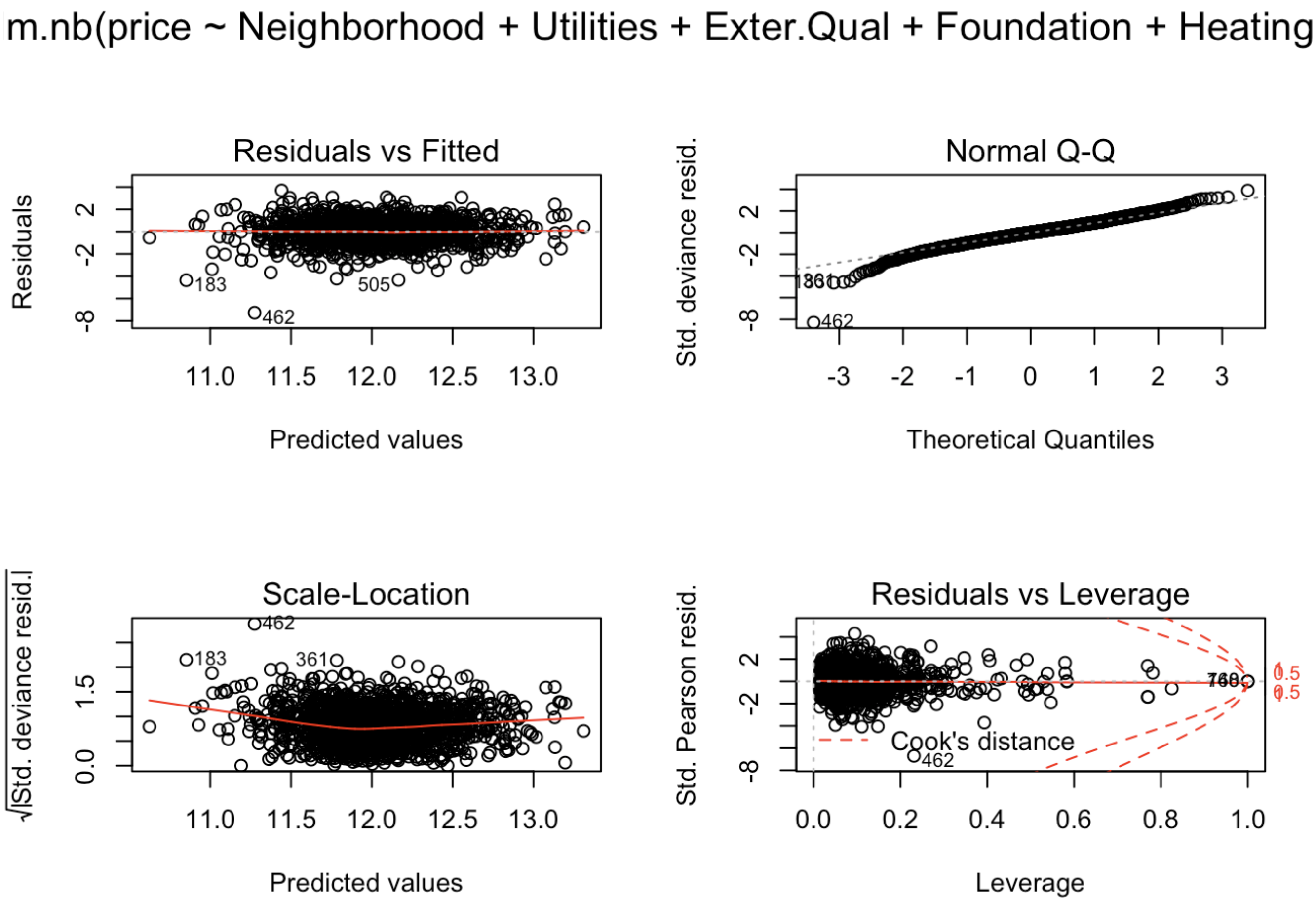
```
par(mfrow = c(2,2))
plot(ames_train$Year.Built, ames_train$price, xlab = "Year.Built", ylab = "price")
plot(ames_train$Lot.Area, ames_train$price, xlab = "Lot.Area", ylab = "price")
plot(ames_train$Year.Remod.Add, ames_train$price, xlab = "Year.Remod.Add", ylab = "pric
e")
plot(ames_train$TotalSq, ames_train$price, xlab = "TotalSq", ylab = "price")
```



4. Assessment of the final model (25 points)

4.1 Residual: must include a residual plot and a discussion

```
par(mfrow = c(2,2),oma=c(0,0,2,0))
plot(model2)
```



- According to the residual plot, our residuals generated by the complex model is normally distributed with a mean of 0. Referring to the leverage plot, we found that all points are within 0.5 Cook's distance, which means there were no influential points.

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

- The in-sample RMSE is 15672.84. Here's a table for bias, Maximum Deviation, Mean Absolute Deviation, RMSE and Coverage for in-sample testing.

```
glm.nb.pred_insample = predict(model2, newdata = ames_train, type = "response", se.fit =
 T)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = residual.scale, type
## = ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
glm.nb.pred_outsample = predict(model2, newdata = ames_test, type = "response", se.fit =
 T)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = residual.scale, type
## = ifelse(type == : prediction from a rank-deficient fit may be misleading
critical = qnorm(0.975)
fit = glm.nb.pred_insample$fit
upr = fit + critical * glm.nb.pred_insample$se.fit
lwr = fit - critical * glm.nb.pred_insample$se.fit
Yhat_insample2 = cbind(fit, lwr, upr)

critical = qnorm(0.975)
fit = glm.nb.pred_outsample$fit
upr = fit + critical * glm.nb.pred_outsample$se.fit
lwr = fit - critical * glm.nb.pred_outsample$se.fit
Yhat_outsample2 = cbind(fit, lwr, upr)

predictions = as.data.frame(Yhat_outsample2)
predictions$PID = ames_test$PID
save(predictions, file="predict.Rdata")

data.frame(c(bias(Yhat_insample2, ames_train$price), maxdev(Yhat_insample2, ames_train$p
rice),
           meanabsdev(Yhat_insample2, ames_train$price), rtsqrerr(Yhat_insample2, ames_t
rain$price),
           coverage(Yhat_insample2, ames_train$price))) %>%
  `rownames<-` (c("Bias", "MaximumDeviation", "MeanAbsoluteDeviation", "RMSE", "Coverag
e")) %>%
  `colnames<-` ("In Sample")
```

```
##                           In Sample
## Bias                      -3.699815
## MaximumDeviation      110234.776947
## MeanAbsoluteDeviation  11047.970253
## RMSE                   15672.839838
## Coverage                   0.428000
```

4.3 Model evaluation: must include an evaluation discussion

F-Test

- We used an F-test to test the variances of the two models. Our result showed that our complex model is significant. Thus, our complex model is better than the simple model in terms of prediction.

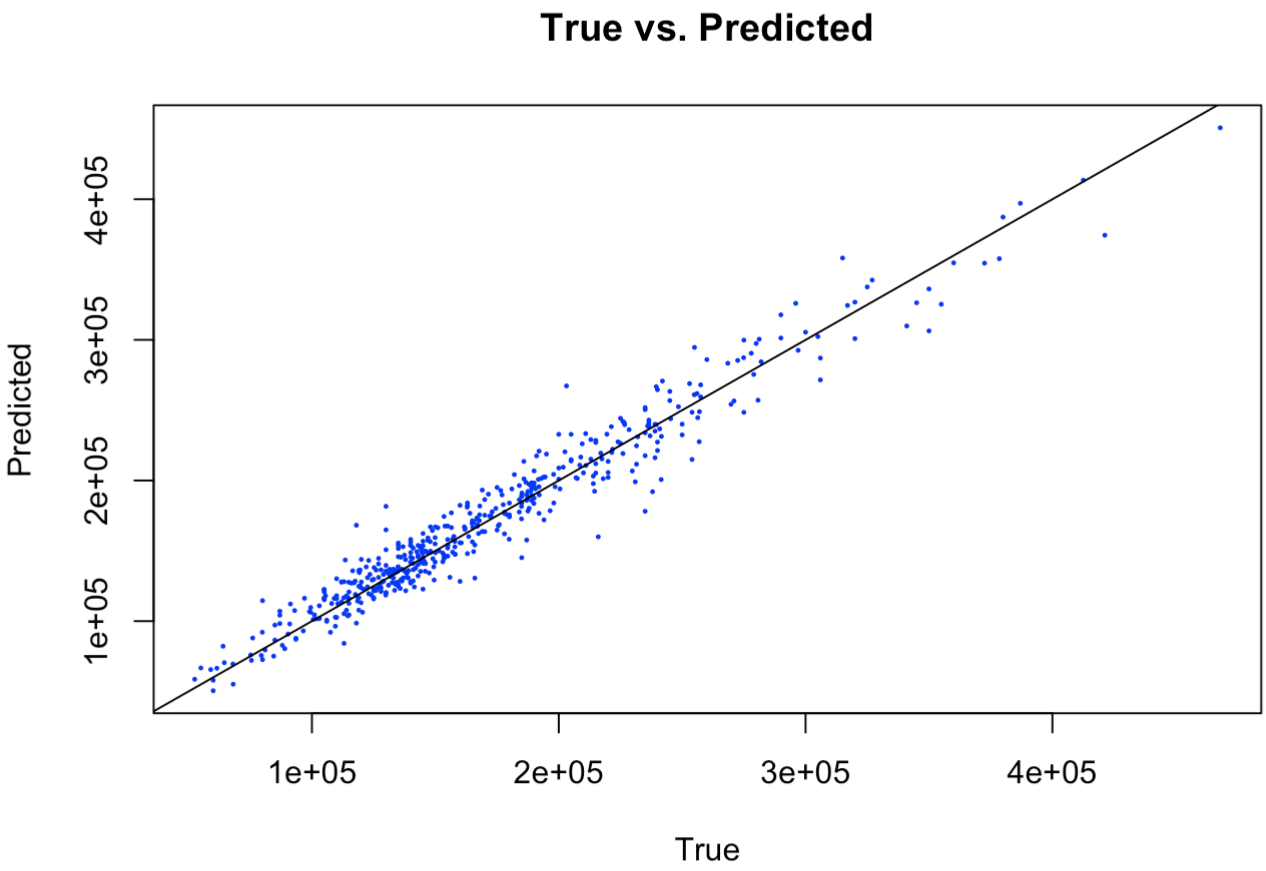
```
var.test(model1, model2)
```

```
## 
##  F test to compare two variances
## 
## data:  model1 and model2
## F = 1.211, num df = 1416, denom df = 1360, p-value = 0.0003719
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.089908 1.345351
## sample estimates:
## ratio of variances 
##           1.210981
```

True vs. Predicted

- We can see that the true values and the predicted values cluster around y = x, which implies good predictions.

```
plot(ames_test$price, Yhat_outsample2[,1], main = "True vs. Predicted", xlab = "True", y
lab = "Predicted", pch = 19, col = "blue", cex = 0.2)
abline(a = 0, b = 1)
```



Dignostic Plot

- Referring to the diagnostic plots in the ‘Residual’ part, we found that the residuals were normally distributed with mean 0.

4.4 Model testing : must include a discussion

- The out-of-sample RMSE is 14460.25. Here’s a table for bias, Maximum Deviation, Mean Absolute Deviation, RMSE and Coverage. As we can see, even though the Bias for the in-sample test is small, the Maximum Deviation, Mean Absolute Deviation, RMSE and Coverage is poorer, which doesn’t imply possible overfitting. The reason that in-sample RMSE is larger may be due to the larger sample size of the training data where the prices are more scattered.

```
data.frame(c(bias(Yhat_insample2, ames_train$price), maxdev(Yhat_insample2, ames_train$p
rice),
           meanabsdev(Yhat_insample2, ames_train$price), rtsqrerr(Yhat_insample2, ames_t
rain$price),
           coverage(Yhat_insample2, ames_train$price)),
           c(bias(Yhat_outsample2, ames_test$price), maxdev(Yhat_outsample2, ames_test$p
rice),
           meanabsdev(Yhat_outsample2, ames_test$price), rtsqrerr(Yhat_outsample2, ames_
test$price),
           coverage(Yhat_outsample2, ames_test$price))) %>%
  `rownames<-` (c("Bias", "MaximumDeviation", "MeanAbsoluteDeviation", "RMSE", "Coverag
e")) %>%
  `colnames<-` (c("In Sample", "Out of Sample"))
```

```
##                            In Sample Out of Sample
## Bias                       -3.699815      1597.750
## MaximumDeviation       110234.776947     64109.651
## MeanAbsoluteDeviation   11047.970253     10756.349
## RMSE                    15672.839838     14460.252
## Coverage                    0.428000         0.434
```

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

- Top 10 Undervalued Houses

```
suppressMessages(library(dplyr))
price_diff = predict(model2, newdata=ames_train, type = "response", se.fit = T)$fit - am
es_train$price
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = residual.scale, type
## = ifelse(type == : prediction from a rank-deficient fit may be misleading
ames_train.new = data.frame(ames_train, price_diff = price_diff)
undervalue = ames_train.new %>%
  arrange(desc(price_diff)) %>%
  mutate(return.rate = price_diff/price) %>%
  select_("PID", "area", "price", "price_diff", "return.rate", "Neighborhood")
```

Top 10 undervalued houses in unit of absolute difference of dollars

```
undervalue.abs = undervalue %>% slice(1:10) %>% select_(-5); undervalue.abs
```

```
## # A tibble: 10 × 5
##          PID  area  price price_diff Neighborhood
##        <int> <int>  <int>      <dbl>       <fctr>
## 1  528166060  2392 386250   91452.71      NridgHt
## 2  909475300  1680 222000   71241.13      Crawfor
## 3  535125010  2207 180000   64707.74        NAmes
## 4  528118090  2790 460000   64331.41      NridgHt
## 5  528365070  2646 260000   62269.76      NoRidge
## 6  533352170  1479 130500   61483.60       Sawyer
## 7  528142020  2088 250000   61119.67      NridgHt
## 8  527225035  1752 164000   52309.85      Gilbert
## 9  533125120  2826 334000   51783.95      NoRidge
## 10 909475040  2365 315000   51704.67      Crawfor
```

Top 10 undervalued houses in unit of return rate, which is the gain per dollar investment

```
undervalue.return = undervalue %>% arrange(desc(return.rate)) %>% slice(1:10) %>% select
_(-4) ; undervalue.return
```

```
## # A tibble: 10 × 5
##          PID  area  price return.rate Neighborhood
##        <int> <int>  <int>       <dbl>       <fctr>
## 1  911102170  1317  40000   0.9698016       IDOTRR
## 2  909101330   498  35000   0.4751488      Edwards
## 3  533352170  1479 130500   0.4711387       Sawyer
## 4  902302150  2337  90000   0.4546893      OldTown
## 5  904351240  1112  63000   0.3823354      Edwards
## 6  535125010  2207 180000   0.3594875        NAmes
## 7  908275110  1560 103500   0.3487548      Edwards
## 8  902205010   612  45000   0.3454180      OldTown
## 9  909281030  1091 104000   0.3289674      Crawfor
## 10 909475300  1680 222000   0.3209060      Crawfor
```

```
overvalued = ames_train.new %>%
  mutate(return.rate = price_diff/price) %>%
  arrange(price_diff) %>%
  select_("PID", "area", "price", "price_diff", "return.rate", "Neighborhood")
```

- Top 10 Overvalued Houses in unit of absolute difference of dollars

```
overvalued %>% slice(1:10) %>% select_(-5)
```

```
## # A tibble: 10 × 5
##          PID  area  price price_diff Neighborhood
##        <int> <int>  <int>      <dbl>       <fctr>
## 1  528164060  2470 615000 -110234.78      NridgHt
## 2  533130130  2349 362500  -79161.06      NoRidge
## 3  528110020  2674 610000  -71555.71      NridgHt
## 4  528360050  3500 584500  -65849.27      NoRidge
## 5  528106020  2402 555000  -56321.00      NridgHt
## 6  528112040  1736 360000  -56013.25      NridgHt
## 7  921128050  1978 425000  -52626.19       Timber
## 8  527256030  2234 441929  -50995.91      StoneBr
## 9  914476450  1689 228500  -50240.66      Mitchel
## 10 527256120  2000 470000  -49238.84      StoneBr
```

Top 10 undervalued houses in unit of return rate, which is the loss per dollar investment

```
overvalued %>% arrange(return.rate) %>% slice(1:10) %>% select_(-4)
```

```
## # A tibble: 10 × 5
##          PID  area  price return.rate Neighborhood
##        <int> <int>  <int>       <dbl>       <fctr>
## 1  904301410  1032 125000  -0.2549799      Edwards
## 2  914476450  1689 228500  -0.2198716      Mitchel
## 3  533130130  2349 362500  -0.2183753      NoRidge
## 4  535402270  1040 163000  -0.2179062        NAmes
## 5  908275200   768 125000  -0.2165132      Edwards
## 6  535381040  2377 142900  -0.2138182      OldTown
## 7  906200230  1822 228500  -0.2054638      SawyerW
## 8  909103020  1273 121000  -0.1928214      Edwards
## 9  528228360  1511 246990  -0.1926542      Blmngtn
## 10 902300110  1627 139500  -0.1885563      OldTown
```

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

5.1 Summary of Results

- Based on the diagonsis and analysis aboved, we see that the complex model, which is the negavtive binormial generalized linear model, outperforms the simple model, the linear model in various ways: maximum deviation, mean absolute deviation, and RMSE. Despite the fact that the complex model's coverage is relatively small and its bias is slightly bigger, which is due to the nature of non-linear model, the overall performance of the complex model is still superior, compared to the simple model. Besides, through model evaluation and testing, the predicted values generated by the complex model aligns well with the true values, in both in-sample testing and out-of-sample testing. In conclusion, we can summarize that our complex model, the negavtive binormial generalized linear model, is a relatively appropriate model, through which we can predict real estate values in future datasets, identify the undervalued ones, and therefore determine whether to purchase them or not.

5.2 Things Learned

- We can't only rely on computer to help us make decision on variables selections, because sometimes it may ignore those variables which are quite important with real-world meaning, and which should be included into our model based on our intuition.
- As the data analysis indicates that the feature "Neighborhood" is quite significant and therefore plays a huge role in the prediction, this result aligns with our common sense about the real estate: location is the key to price.
- Both simple and complex models show that the area of the house is important in predicting price. The result is consistent with our common sense that larger houses have higher prices.
- To evaluate a model, we should not just focus on reducing RMSE. Coverage rate, prediction bias and other criteria matters. Considering the variance-bias trade off, small RMSE could follow with high prediction.
- Suprisingly, the year when the house was built is quite influential to price, since it has a large coefficient and a small p-value. Larger year number results in higher price, which means newer houses are more popular and thus more expensive.

5.3 Business Insights

What Does the Profitable Houses Look Like?

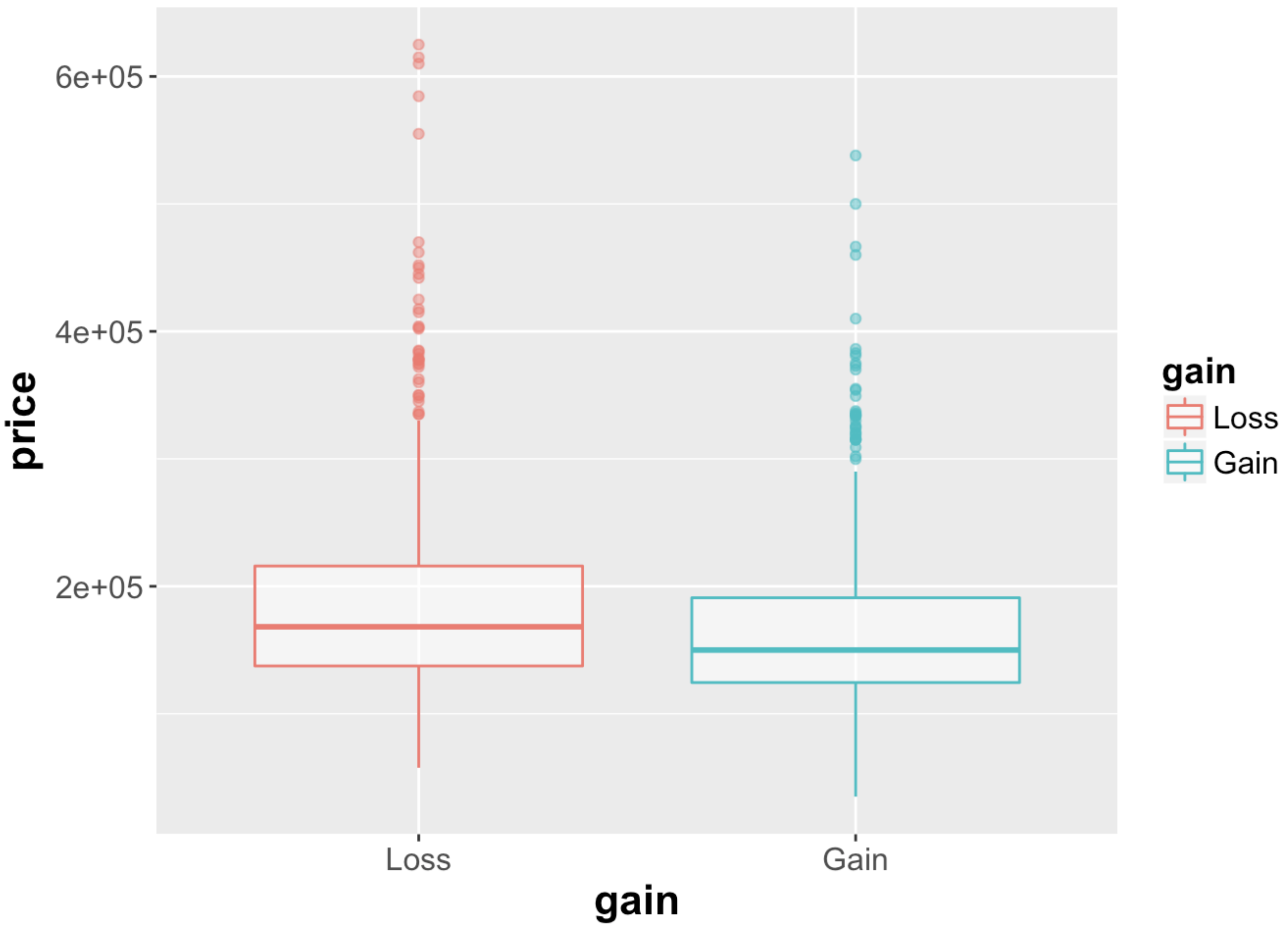
```
avg_gain_price = overvalued %>% filter(return.rate>0) %>% summarise(mean(price))
avg_loss_price = overvalued %>% filter(return.rate<0) %>% summarise(mean(price))

t.test((overvalued %>% filter(return.rate>0)%>%select_(3)), overvalued %>%
filter(return.rate<0)%>%select_(3))
```

```
##
##  Welch Two Sample t-test
##
## data:  (overvalued %>% filter(return.rate > 0) %>% select_(3)) and overvalued %>% fil
ter(return.rate < 0) %>% select_(3)
## t = -5.9122, df = 1460.1, p-value = 4.196e-09
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -28992.05 -14546.57
## sample estimates:
## mean of x mean of y
##  165634.0  187403.3
```

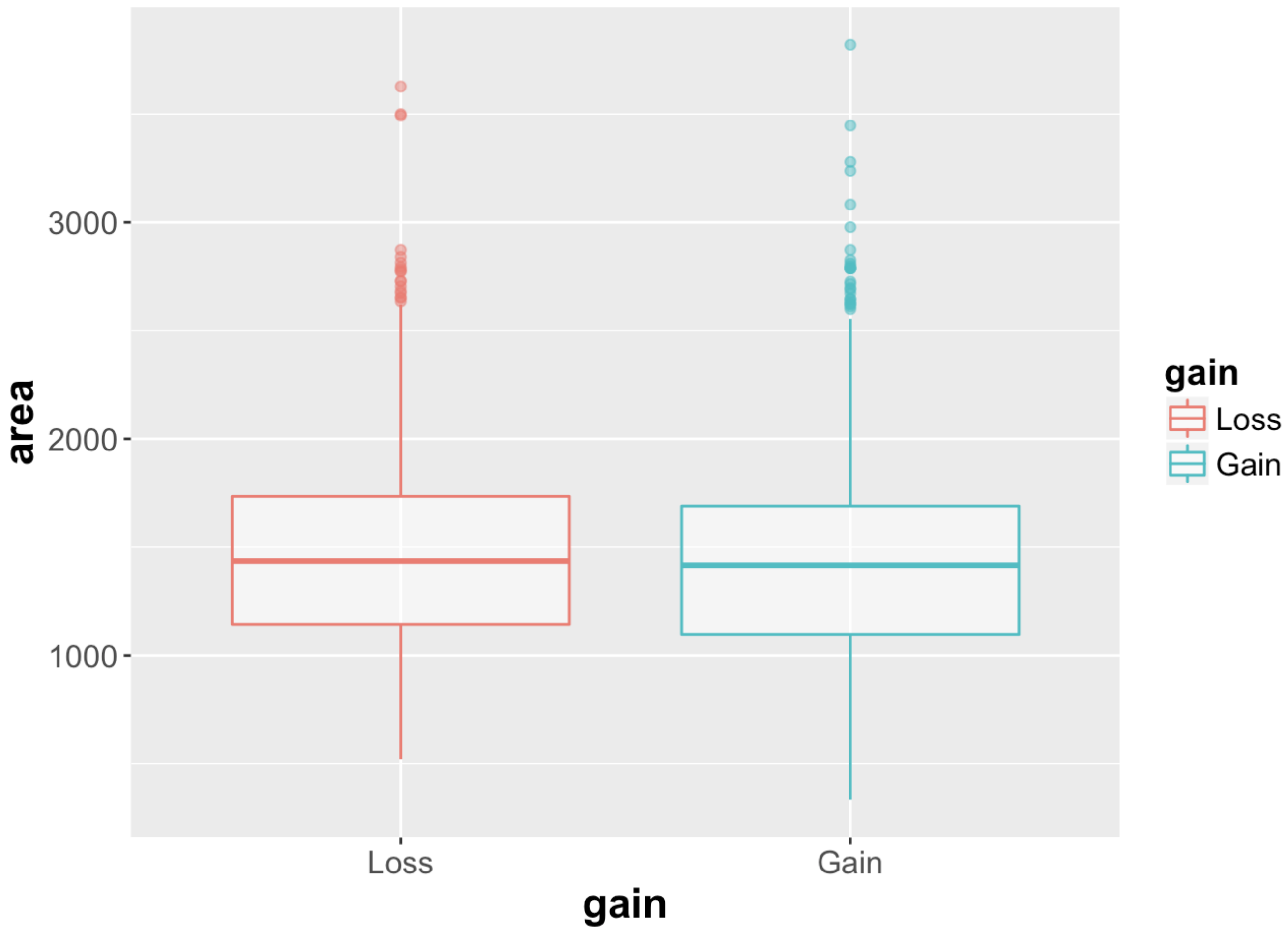
```
avg_gain_area = overvalued %>% filter(return.rate>0) %>% summarise(mean(area))
avg_loss_area = overvalued %>% filter(return.rate<0) %>% summarise(mean(area))

t.test((overvalued %>% filter(return.rate>0)%>%select_(2)), overvalued %>%
filter(return.rate<0)%>%select_(2))
```

```
##
##  Welch Two Sample t-test
##
## data:  (overvalued %>% filter(return.rate > 0) %>% select_(2)) and overvalued %>% fil
ter(return.rate < 0) %>% select_(2)
## t = -0.78903, df = 1495.3, p-value = 0.4302
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -67.74989  28.88039
## sample estimates:
## mean of x mean of y
##  1464.767  1484.202
```

```
#plot gain v.s. price/area
overvalued = overvalued %>% mutate(gain = factor(return.rate >= 0))
levels(overvalued$gain) = c("Loss", "Gain")
ggplot(data = overvalued, aes(x = gain, y = price, color = gain)) +
  geom_boxplot(alpha = 0.5) +
  theme_grey() +
  theme(axis.title=element_text(size=16,face="bold"),
        axis.text = element_text(size=12),
        plot.title = element_text(size=20,face="bold", hjust = 0.5),
        legend.text = element_text(size = 12),
        legend.title = element_text(size = 14, face = "bold"))
```



```
ggplot(data = overvalued, aes(x = gain, y = area, color = gain)) +
  geom_boxplot(alpha = 0.5)+
  theme_grey() +
  theme(axis.title=element_text(size=16,face="bold"),
        axis.text = element_text(size=12),
        plot.title = element_text(size=20,face="bold", hjust = 0.5),
        legend.text = element_text(size = 12),
        legend.title = element_text(size = 14, face = "bold"))
```



- By comparing the average price of houses which may help buyer gain money and houses which may result in loss, we have two useful findings which might instruct investment on house.
- The average price for houses which generate gain is 1.6563401×10^5, while that for houses which result in loss is 1.8740331×10^5. By conducting t-test on these two groups, there was a statistically significant difference between these two.
- The average area for houses which may result in gain is 1464.7673797, while that for houses which may result in loss is 1484.2021277. By conducting t-test on these two groups, there was no statistically significant difference between these two.
- To conclude, we would better buy cheaper house if we want to invest some money on houses, given no other detailed information on houses. On the one hand, cheaper house could have more probability increasing its value in the future. On the other hand, considering the risk of losing their money in investment on house, put less money on one investment could reduce the possible loss in the future, which is a desire advantage of cheaper house, especially when investors are risk aversion.

Part IV

Create predictions for the validation data from your final model and write out to a file `prediction-validation.Rdata` This should have the same format as the models in Part I and II. 10 points

```
load("ames_validation.Rdata")

levels(ames_validation$Bsmt.Cond) = c(levels(ames_validation$Bsmt.Cond)[-1], "No Basement")
ames_validation$Bsmt.Cond[is.na(ames_validation$Bsmt.Cond)] = "No Basement"
levels(ames_validation$Bsmt.Exposure) = c(levels(ames_validation$Bsmt.Exposure)[-1], "No Basement")
ames_validation$Bsmt.Exposure[is.na(ames_validation$Bsmt.Exposure)] = "No Basement"
levels(ames_validation$Bsmt.Qual) = c(levels(ames_validation$Bsmt.Qual)[-1], "No Basement")
ames_validation$Bsmt.Qual[is.na(ames_validation$Bsmt.Qual)] = "No Basement"
ames_validation$Bsmt.Qual[ames_validation$Bsmt.Qual=="Ex"] = "Gd"
levels(ames_validation$BsmtFin.Type.1) = c(levels(ames_validation$BsmtFin.Type.1)[-1], "No Basement")
ames_validation$BsmtFin.Type.1[is.na(ames_validation$BsmtFin.Type.1)] = "No Basement"
ames_validation$BsmtFin.Type.1[ames_validation$BsmtFin.Type.1=="ALQ"] = "Rec"
levels(ames_validation$BsmtFin.Type.2) = c(levels(ames_validation$BsmtFin.Type.2)[-1], "No Basement")
ames_validation$BsmtFin.Type.2[is.na(ames_validation$BsmtFin.Type.2)] = "No Basement"

levels(ames_validation$Alley) = c(levels(ames_validation$Alley)[-1], "No alley access")
ames_validation$Alley[is.na(ames_validation$Alley)] = "No alley access"
levels(ames_validation$Fireplace.Qu) = c(levels(ames_validation$Fireplace.Qu)[-1], "No Fireplace")
ames_validation$Fireplace.Qu[is.na(ames_validation$Fireplace.Qu)] = "No Fireplace"
levels(ames_validation$Garage.Type) = c(levels(ames_validation$Garage.Type)[-1], "No Garage")
ames_validation$Garage.Type[is.na(ames_validation$Garage.Type)] = "No Garage"
levels(ames_validation$Garage.Finish) = c(levels(ames_validation$Garage.Finish)[-1], "No Garage")
ames_validation$Garage.Finish[is.na(ames_validation$Garage.Finish)] = "No Garage"

levels(ames_validation$Garage.Qual) = c("Po","Fa","TA","Gd","Ex", "No Garage")
ames_validation$Garage.Qual[is.na(ames_validation$Garage.Qual)] = "No Garage"
levels(ames_validation$Garage.Cond) = c(levels(ames_validation$Garage.Cond)[-1], "No Garage")
ames_validation$Garage.Cond[is.na(ames_validation$Garage.Cond)] = "No Garage"
levels(ames_validation$Pool.QC) = c(levels(ames_validation$Pool.QC)[-1], "No Pool")
ames_validation$Pool.QC[is.na(ames_validation$Pool.QC)] = "No Pool"

levels(ames_validation$Fence) = c(levels(ames_validation$Fence)[-1], "No Fence")
ames_validation$Fence[is.na(ames_validation$Fence)] = "No Fence"
levels(ames_validation$Misc.Feature) = c("None", levels(ames_validation$Misc.Feature)[-1])
ames_validation$Misc.Feature[is.na(ames_validation$Misc.Feature)] = "None"

levels(ames_validation$Garage.Yr.Blt) = c(levels(ames_validation$Garage.Yr.Blt)[-1], "No Garage Year")
ames_validation$Garage.Yr.Blt[is.na(ames_validation$Garage.Yr.Blt)] = mean(ames_validation$Garage.Yr.Blt[!is.na(ames_validation$Garage.Yr.Blt)])

ames_validation = ames_validation %>%
  mutate(MS.SubClass = as.factor(MS.SubClass))
```

```
ames_validation$price = predict(model2, newdata=ames_validation, type = "response", se.f
it = T)$fit
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = residual.scale, type
## = ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
predictions = ames_validation
predictions$PID = ames_validation$PID
predictions %>% select_("PID", "price")
```

```
## # A tibble: 413 × 2
##          PID     price
##        <int>     <dbl>
## 1  527451060 100694.53
## 2  531452020 120014.61
## 3  907126010 155018.96
## 4  535379060 108182.95
## 5  535303110 142014.83
## 6  907227290 132009.78
## 7  903458170  91675.99
## 8  528477050 374613.93
## 9  907250020 231664.07
## 10 527166030 180899.69
## # ... with 403 more rows
```

```
save(predictions, file="prediction-validation.Rdata")
```

Class Presentations

Each Group should prepare 5 slides in their Github repo: (save as slides.pdf)

- Most interesting graphic (a picture is worth a thousand words prize!)
- Best Model (motivation, how you found it, why you think it is best)
- Best Insights into predicting Sales Price.
- 2 Best Houses to purchase (and why)
- Best Team Name/Graphic

We will select winners based on the above criteria and overall performance.

Finally your repo should have: `writeup.Rmd` , `writeup.pdf` , `slides.Rmd` (and whatever output you use for the presentation) and `predict.Rdata` and `predict-validation.Rdata` .