1 Background

As a statistical consultant working for a real estate investment firm, your task is to develop a model to predict the selling price of a given home in Ames, Iowa. Your employer hopes to use this information to help assess whether the asking price of a house is higher or lower than the true value of the house. If the home is undervalued, it may be a good investment for the firm.

2 Training Data and relevant packages

In order to better assess the quality of the model you will produce, the data have been randomly divided into three separate pieces: a training data set, a testing data set, and a validation data set. For now we will load the training data set, the others will be loaded and used later.

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

Use the code block below to load any necessary packages

```
library(statsr)
library(dplyr)
library(ggplot2)
library(GGally)
df<-ames_train
```

2.1 Part 1 - Exploratory Data Analysis (EDA)

When you first get your data, it’s very tempting to immediately begin fitting models and assessing how they perform. However, before you begin modeling, it’s absolutely essential to explore the structure of the data and the relationships between the variables in the data set.

Do a detailed EDA of the ames_train data set, to learn about the structure of the data and the relationships between the variables in the data set (refer to Introduction to Probability and Data, Week 2, for a reminder about EDA if needed). Your EDA should involve creating and reviewing many plots/graphs and considering the patterns and relationships you see.

After you have explored completely, submit the three graphs/plots that you found most informative during your EDA process, and briefly explain what you learned from each (why you found each informative).

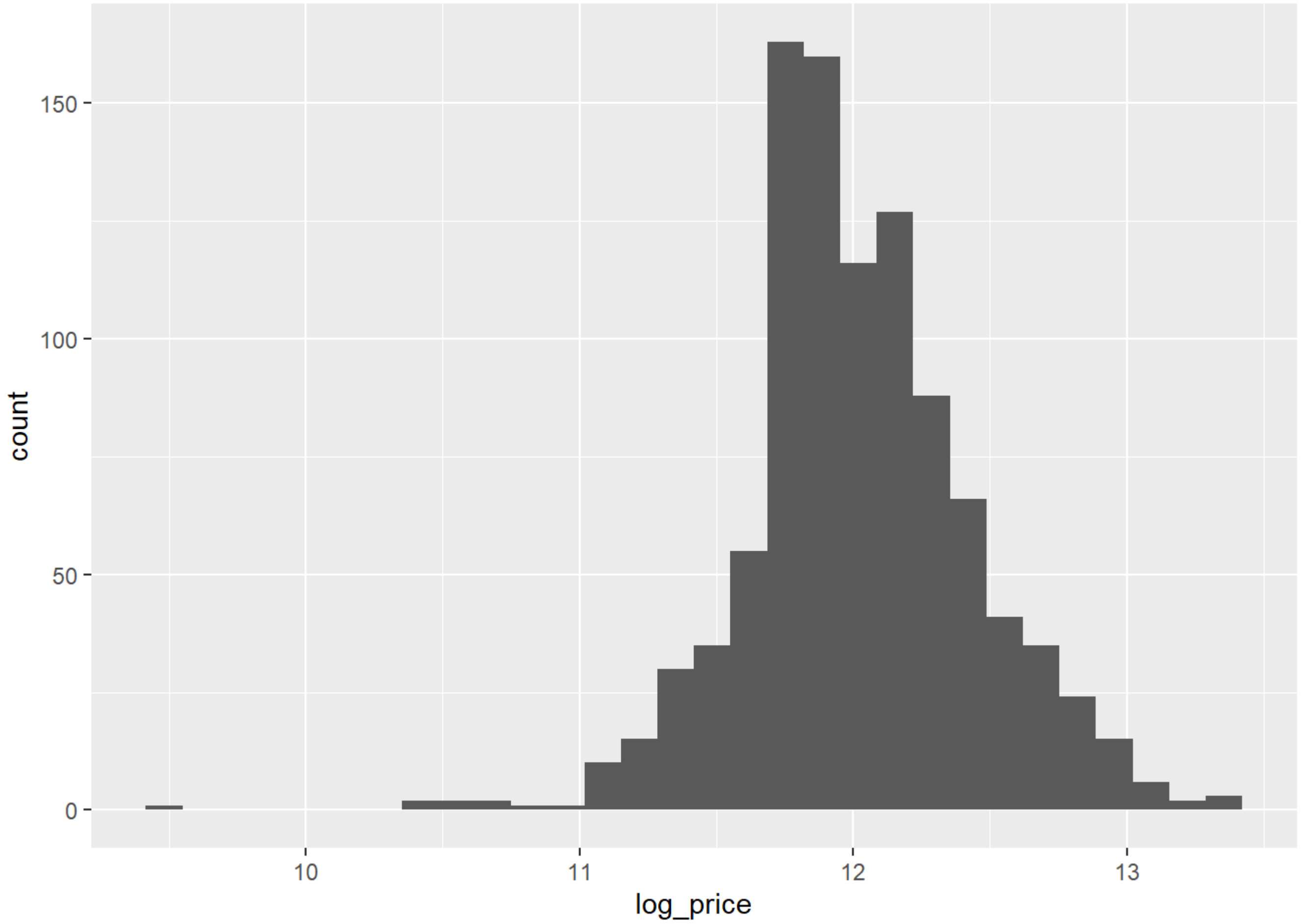
First of all, we will create log_transformed variables which will aid in further analysis as indicated by previous quizzes:

```
df<-df%>%mutate(log_price=log(price))
df<-df%>%mutate(log_area=log(area))
df<-df%>%mutate(log_lot_area=log(Lot.Area))
```

Let us visualize the distribution of log_price:

```
df%>%ggplot(aes(x=log_price))+geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We can see that the distribution is fairly normal.

Let us investigate if the presence of Central Air Conditioning is associated with higher price of a house:

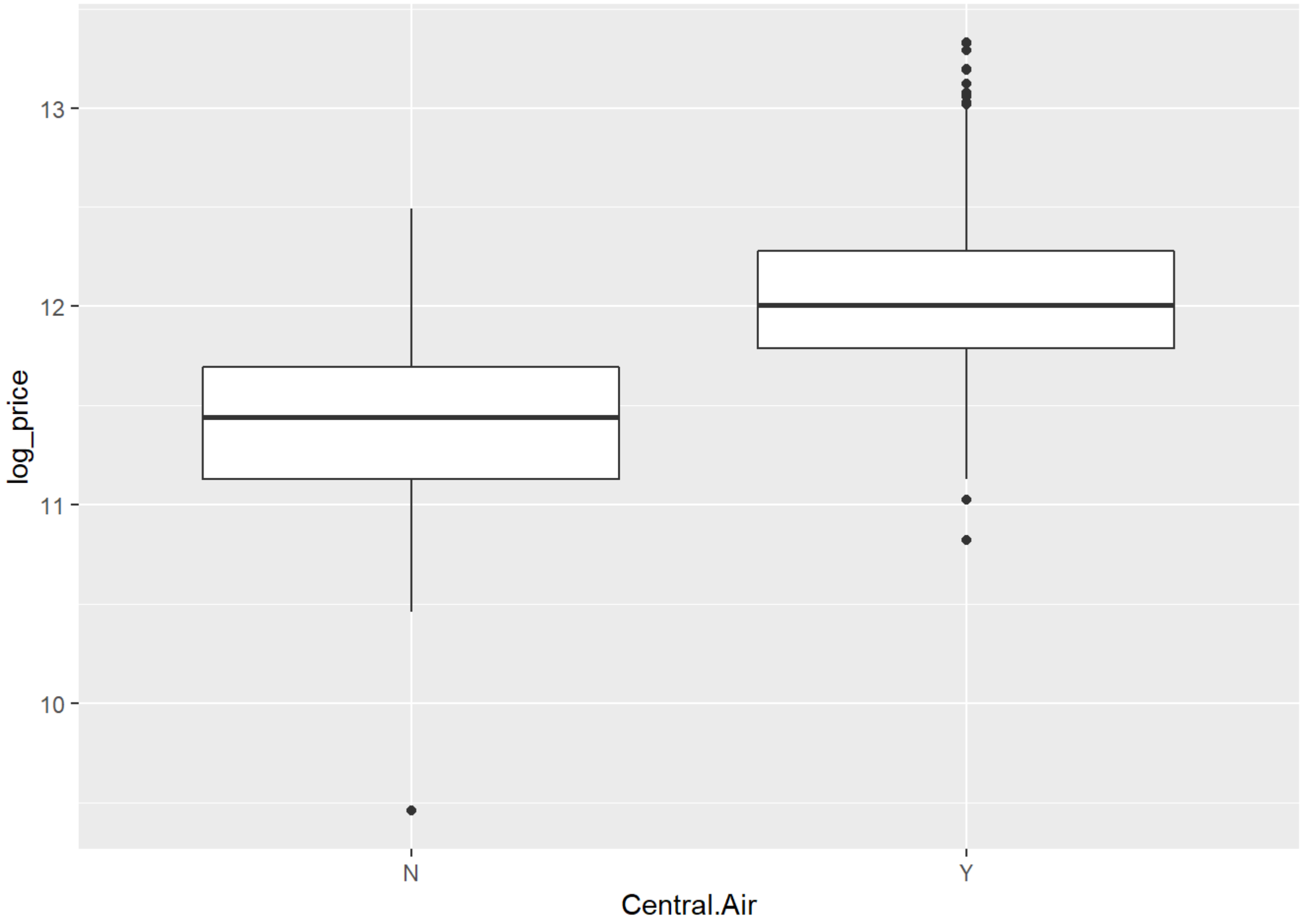
Let us see some summary statistics and plots pertaining to this question:

```
df%>%group_by(Central.Air)%>%summarise(mean_log_price=mean(log_price),median_log_price=median(lo
g_price),sd_log_price=sd(log_price),number=n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 5
##   Central.Air mean_log_price median_log_price sd_log_price number
##   <fct>                <dbl>            <dbl>        <dbl>  <int>
## 1 N                     11.4             11.4        0.495     55
## 2 Y                     12.1             12.0        0.384    945
```

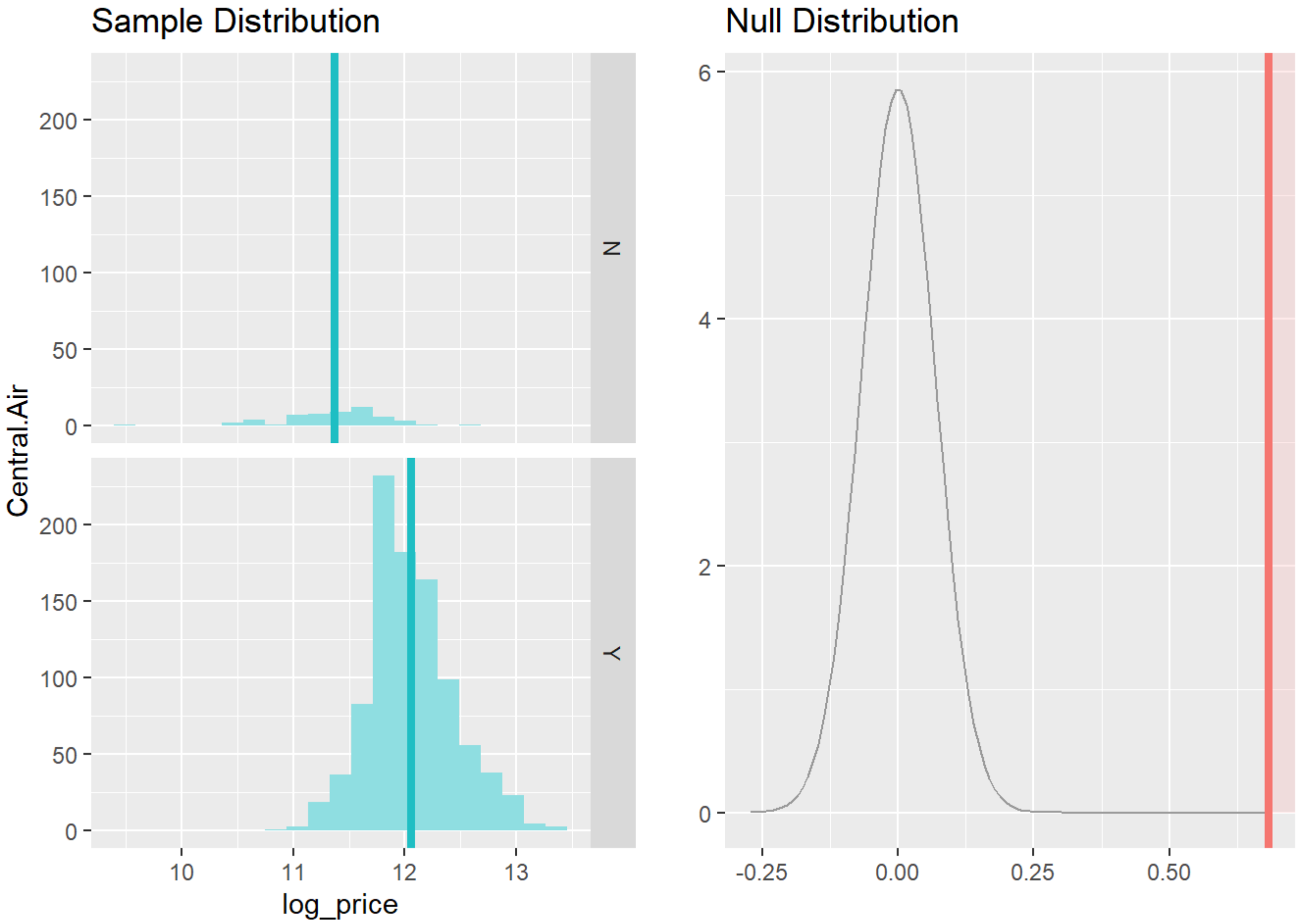
```
df%>%ggplot(aes(x=Central.Air,y=log_price))+geom_boxplot()
```



We can see that both the summary statistics and the box-plot indicate that price of homes with Central Air Conditioning is higher. Mean log price for homes with CAC is 12.1 while that for homes without CAC is 11.4. Also, the variance for homes without Central Air conditioning is higher. Let us see if this difference is statistically significant. The conditions for the t-test for difference between 2 means is met.

```
inference(data=df,x=Central.Air,y=log_price,typ='ht',method='theoretical',statistic='mean',null=
0,alternative='greater',order = c('Y','N'))
```

```
## Response variable: numerical
## Explanatory variable: categorical (2 levels)
## n_Y = 945, y_bar_Y = 12.0561, s_Y = 0.3839
## n_N = 55, y_bar_N = 11.3719, s_N = 0.4954
## H0: mu_Y =  mu_N
## HA: mu_Y > mu_N
## t = 10.0689, df = 54
## p_value = < 0.0001
```



From this test, we get a t-statistic greater than 10 which gives us a p-value < 0.0001. According to this, assuming the two means are equal, the probability of obtaining two samples of given size from the two groups whose means differ by at least 12.0561-11.3719 is less than 0.0001. Thus, we can say that mean log price for homes with Central Air conditioning is higher than that for homes without Central Air Conditioning.

The houses in the data-set have different types of foundations as shown below:

```
summary(df$Foundation)
```

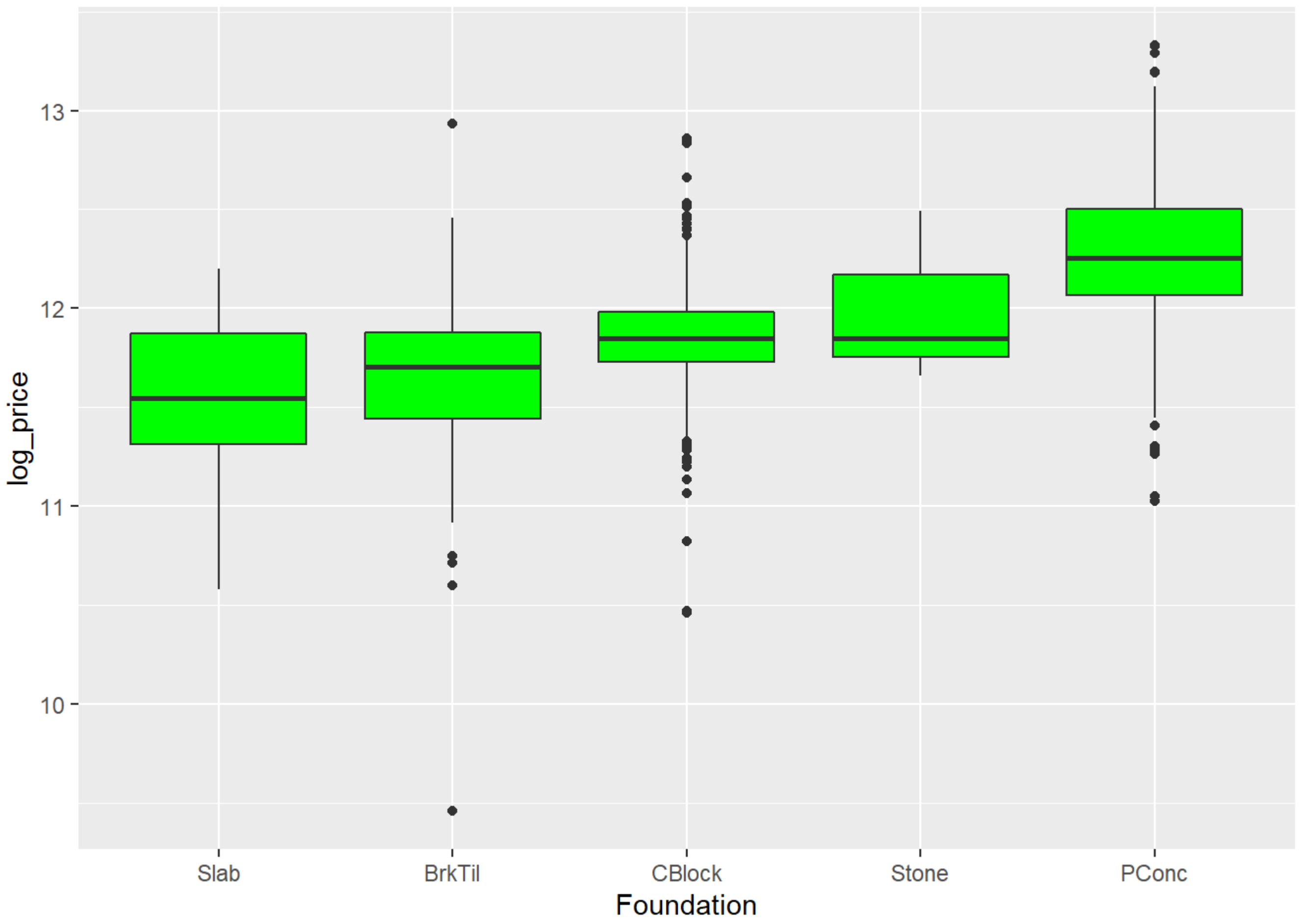
```
## BrkTil CBlock  PConc   Slab  Stone   Wood 
##    102    430    453     12      3      0
```

Let us see if the price of a house varies with the type of foundation:

```
df%>%group_by(Foundation)%>%summarise(mean_log_price=mean(log_price),median_log_price=median(log
_price),sd_log_price=sd(log_price),number=n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
## # A tibble: 5 x 5
##   Foundation mean_log_price median_log_price sd_log_price number
##   <fct>               <dbl>            <dbl>        <dbl>  <int>
## 1 BrkTil               11.6             11.7        0.421    102
## 2 CBlock               11.9             11.8        0.284    430
## 3 PConc                12.3             12.3        0.373    453
## 4 Slab                 11.5             11.5        0.446     12
## 5 Stone                12.0             11.8        0.435      3
```

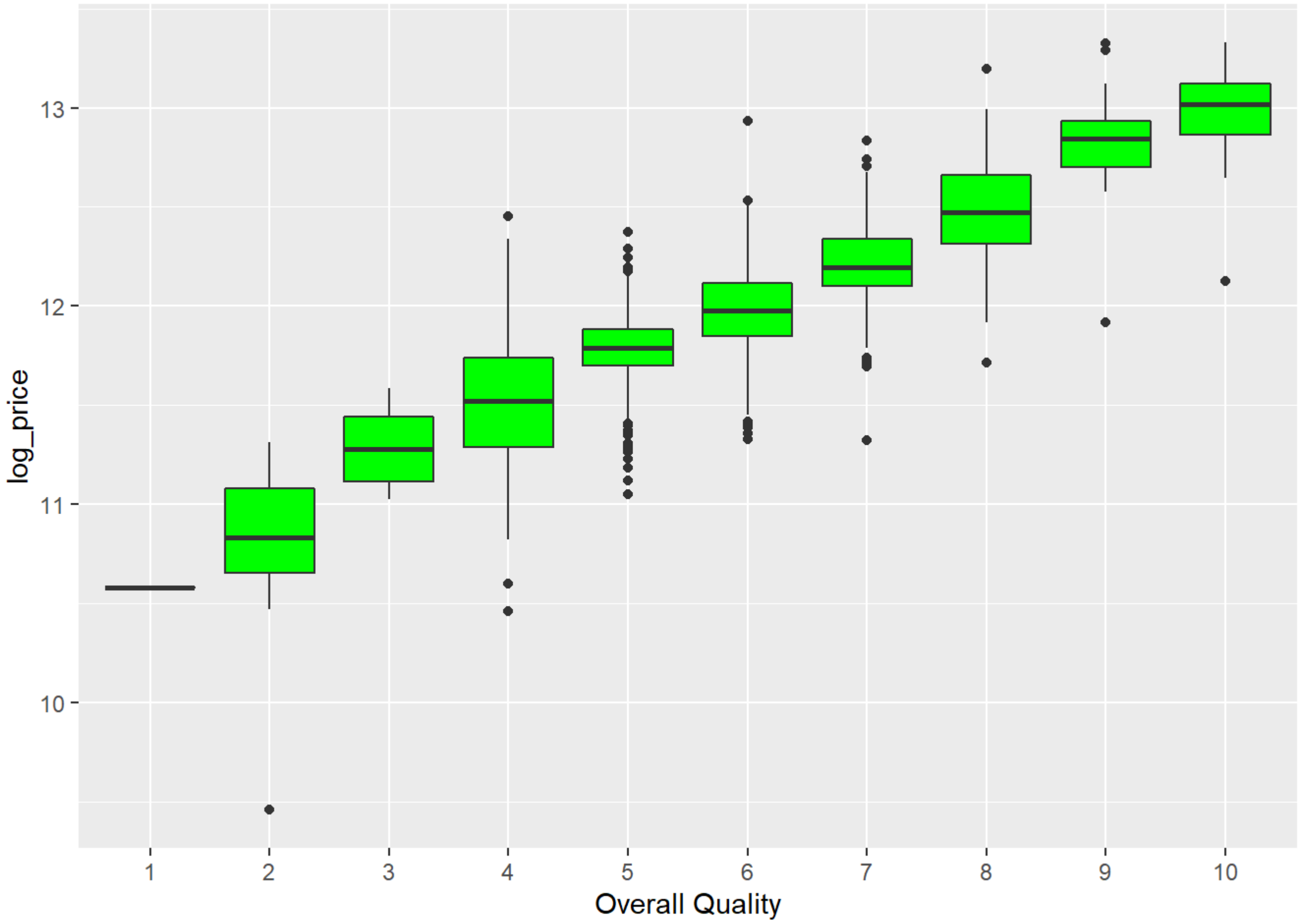
```
df%>%ggplot(aes(x=reorder(Foundation,log_price),y=log_price))+geom_boxplot(fill='green')+xlab('F
oundation')+ylab('log_price')
```



We can see that Concrete foundation homes are the most expensive with mean log price of 12.3.

Now, we expect higher quality homes to have higher selling prices. Let us confirm this notion with the help of some summary statistics and plots:

```
df%>%ggplot(aes(x=factor(Overall.Qual),y=log_price))+geom_boxplot(fill='green')+xlab('Overall Qu
ality')
```



```
df%>%group_by(Overall.Qual)%>%summarise(mean_log_price=mean(log_price),median_log_price=median(l
og_price),sd_log_price=sd(log_price),number=n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 10 x 5
##    Overall.Qual mean_log_price median_log_price sd_log_price number
##           <int>          <dbl>            <dbl>        <dbl>  <int>
##  1            1           10.6             10.6       NA          1
##  2            2           10.7             10.8        0.578      8
##  3            3           11.3             11.3        0.197      9
##  4            4           11.5             11.5        0.339     68
##  5            5           11.8             11.8        0.188    305
##  6            6           12.0             12.0        0.239    238
##  7            7           12.2             12.2        0.210    200
##  8            8           12.5             12.5        0.249    122
##  9            9           12.8             12.8        0.228     40
## 10           10           12.9             13.0        0.356      9
```

From the summary statistics as well as from the box-plots, we can observe that higher quality is associated with higher price. Thus, maintaining good quality can earn rich dividends.

Now, other two variables which I found interesting were Land.Contour and Lot.Shape. Let us check them out:

```
summary(df$Land.Contour)
```

```
## Bnk HLS Low Lvl
##  33  38  20 909
```

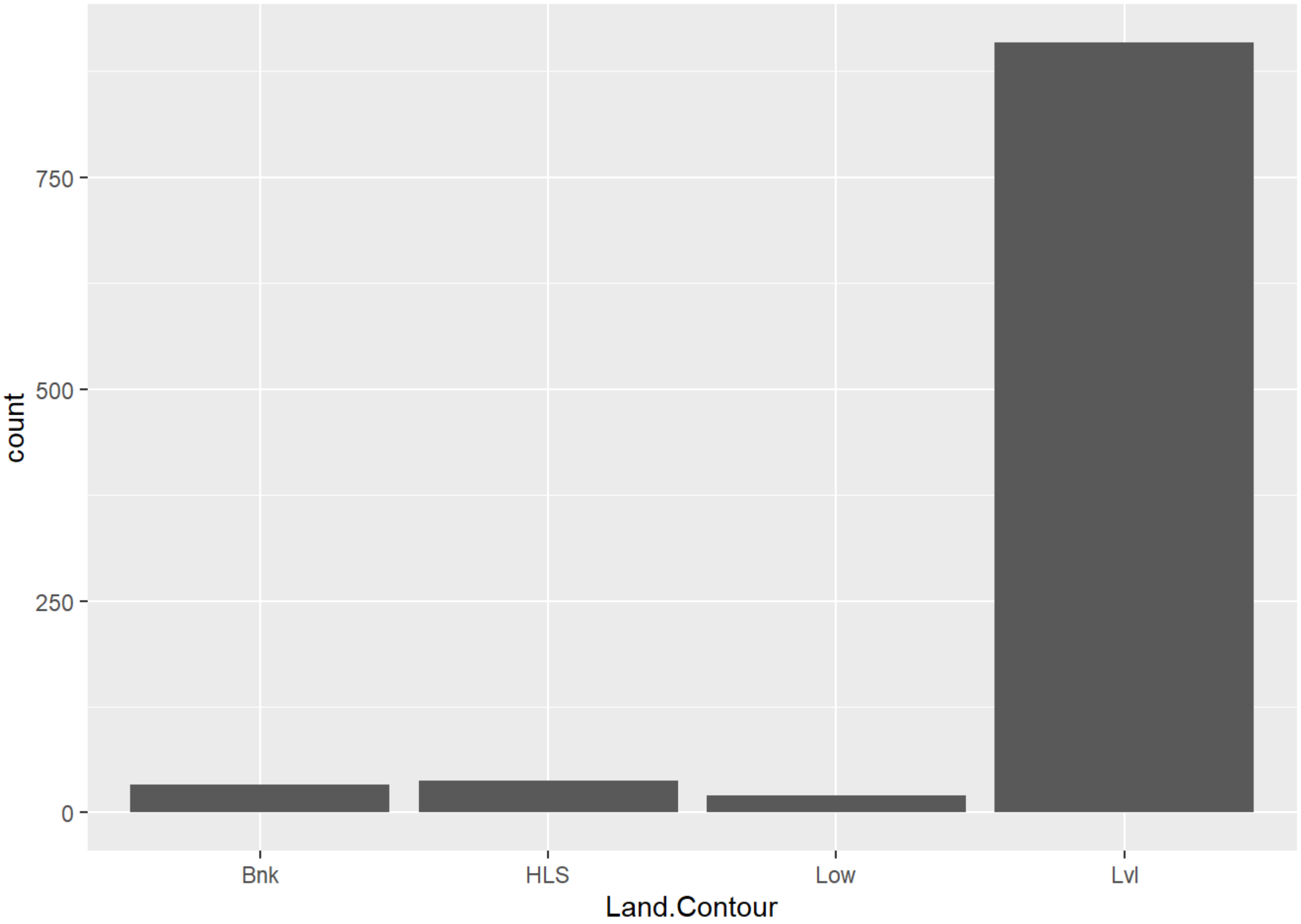
```
summary(df$Lot.Shape)
```

```
## IR1 IR2 IR3 Reg
## 338  30   3 629
```

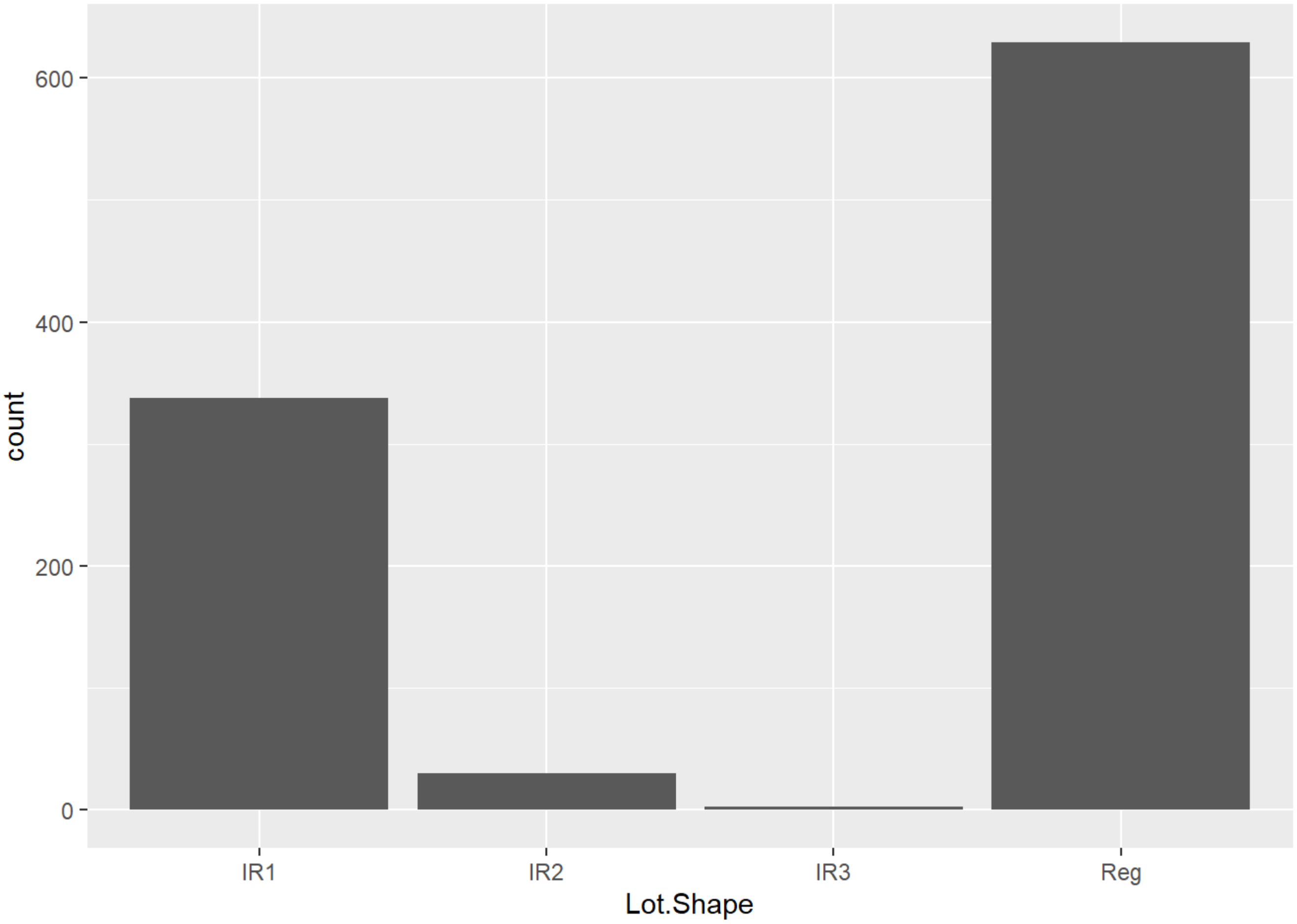
We can see that Land.Contour can be divided into two broad categories: Level/Not-Level. Similarly, Lot.Shape can be divided into two broad categories: Regular/Irregular.

Let us investigate these variables in more detail:

```
df%>%ggplot(aes(x=Land.Contour))+geom_bar()
```



```
df%>%ggplot(aes(x=Lot.Shape))+geom_bar()
```



We can see that majority of Land Contours are level and majority of Lot Shapes are regular. We want to investigate if the Land Contour is associated with regular or irregular lot shape. Let us divide these variables into the two broad categories as mentioned earlier:

```
x_dat<-df
x_dat<-x_dat%>%mutate(Land.Contour.New=factor(ifelse(Land.Contour=='Lvl','Level','Not Level')))
x_dat<-x_dat%>%mutate(Lot.Shape.New=factor(ifelse(Lot.Shape=='Reg','Regular','Irrelgular')))
```

Let us visualize the contingency table for the two variables:

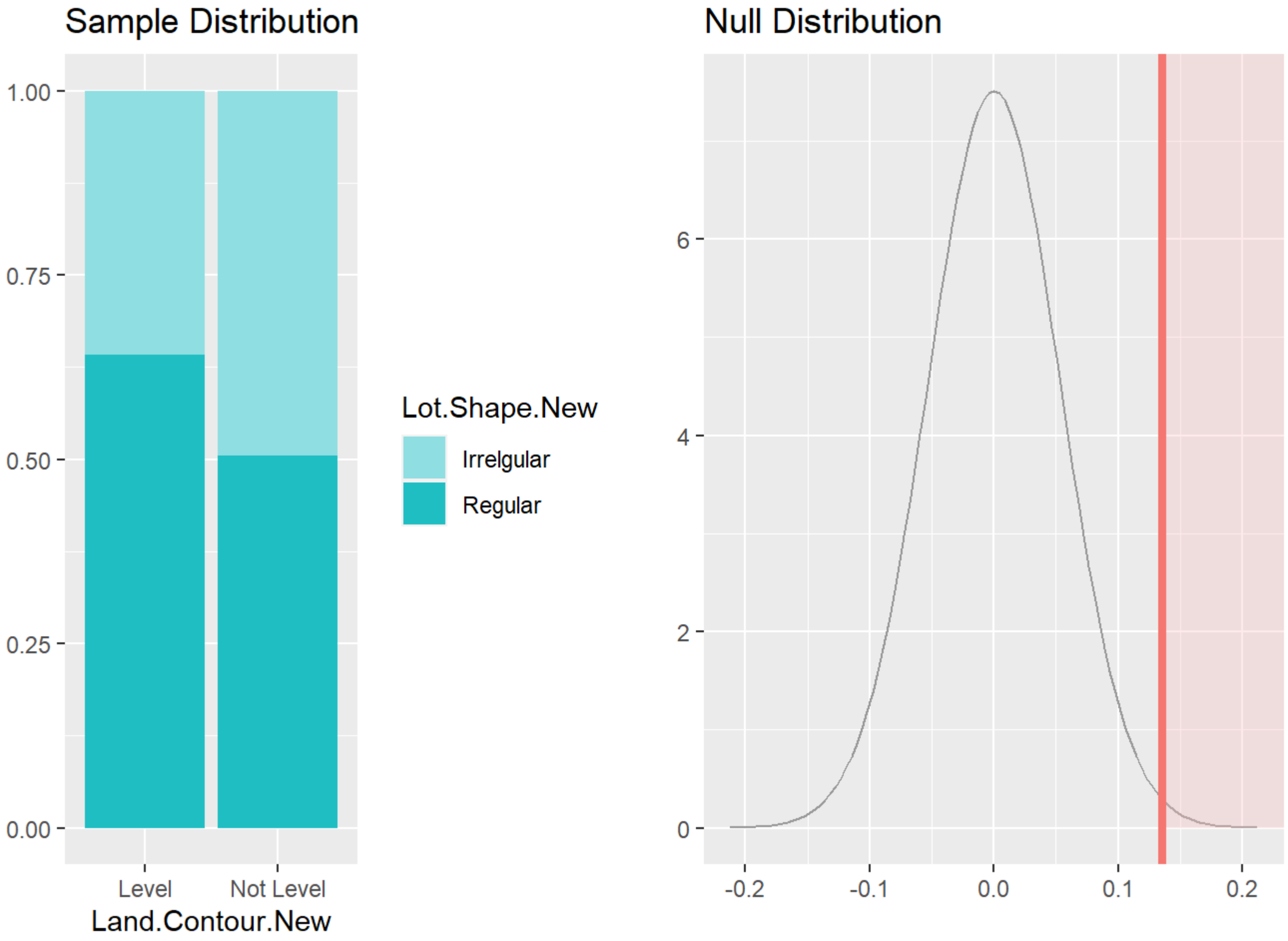
```
table(x_dat$Lot.Shape.New,x_dat$Land.Contour.New)
```

```
##
##              Level Not Level
##   Irrelgular   326        45
##   Regular      583        46
```

The proportion of Regular Lots is 0.6413 and 0.5055 for Level and Not-Level Contours respectively. Let us see if this difference is statistically significant: Let us conduct a test for the difference between the two proportions:

```
inference(data=x_dat,y=Lot.Shape.New,x=Land.Contour.New,type='ht',statistic='proportion',method=
'theoretical',null=0,alternative='greater',success='Regular',order=c('Level','Not Level'))
```

```
## Response variable: categorical (2 levels, success: Regular)
## Explanatory variable: categorical (2 levels)
## n_Level = 909, p_hat_Level = 0.6414
## n_Not Level = 91, p_hat_Not Level = 0.5055
## H0: p_Level =  p_Not Level
## HA: p_Level > p_Not Level
## z = 2.5581
## p_value = 0.0053
```



We get a Z-score of 2.5581

From this test, we can infer that Land Contours which are level are more likely to have Regular Shaped Plots.

2.2 Part 2 - Development and assessment of an initial model, following a semi-guided process of analysis

2.2.1 Section 2.1 An Initial Model

In building a model, it is often useful to start by creating a simple, intuitive initial model based on the results of the exploratory data analysis. (Note: The goal at this stage is **not** to identify the “best” possible model but rather to choose a reasonable and understandable starting point. Later you will expand and revise this model to create your final model.

Based on your EDA, select *at most* 10 predictor variables from “ames_train” and create a linear model for `price` (or a transformed version of price) using those variables. Provide the *R code* and the *summary output table* for your model, a *brief justification* for the variables you have chosen, and a *brief discussion* of the model results in context (focused on the variables that appear to be important predictors and how they relate to sales price).

From the EDA, We can observe that the basement variables have a lot of null values. These mean that th house has no basement. Thus, let us re-code them. Also, some of them have a single null value. We will eliminate that observation:

```
df<-df%>%mutate(Bsmt.Qual=factor(ifelse(is.na(df$Bsmt.Qual),'No_Bsmt',Bsmt.Qual)))
df<-df%>%mutate(Bsmt.Cond=factor(ifelse(is.na(df$Bsmt.Cond),'No_Bsmt',Bsmt.Cond)))
df<-df%>%mutate(Bsmt.Exposure=factor(ifelse(is.na(df$Bsmt.Exposure),'No_Bsmt',Bsmt.Exposure)))
df<-df%>%mutate(BsmtFin.Type.1=factor(ifelse(is.na(df$BsmtFin.Type.1),'No_Bsmt',BsmtFin.Type.
1)))
df<-df%>%mutate(BsmtFin.Type.2=factor(ifelse(is.na(df$BsmtFin.Type.2),'No_Bsmt',BsmtFin.Type.
2)))
df<-df%>%mutate(Fireplace.Qu=factor(ifelse(is.na(df$Fireplace.Qu),'No_Fireplace',Fireplace.Qu)))
df<-df%>%mutate(Garage.Qual=factor(ifelse(is.na(df$Garage.Qual),'No_Garage',Garage.Qual)))
df<-df%>%mutate(Garage.Cond=factor(ifelse(is.na(df$Garage.Cond),'No_Garage',Garage.Cond)))
df<-df%>%filter(!(is.na(BsmtFin.SF.1)),!(is.na(BsmtFin.SF.2)),!(is.na(Bsmt.Unf.SF)),!(is.na(Tota
l.Bsmt.SF)),!(is.na(Bsmt.Full.Bath)),!(is.na(Bsmt.Half.Bath)))
df<-df%>%filter(!(is.na(Garage.Cars)))
```

Further, some Ordinal Categorical variables are not encoded ordianlly. Let us redo that:

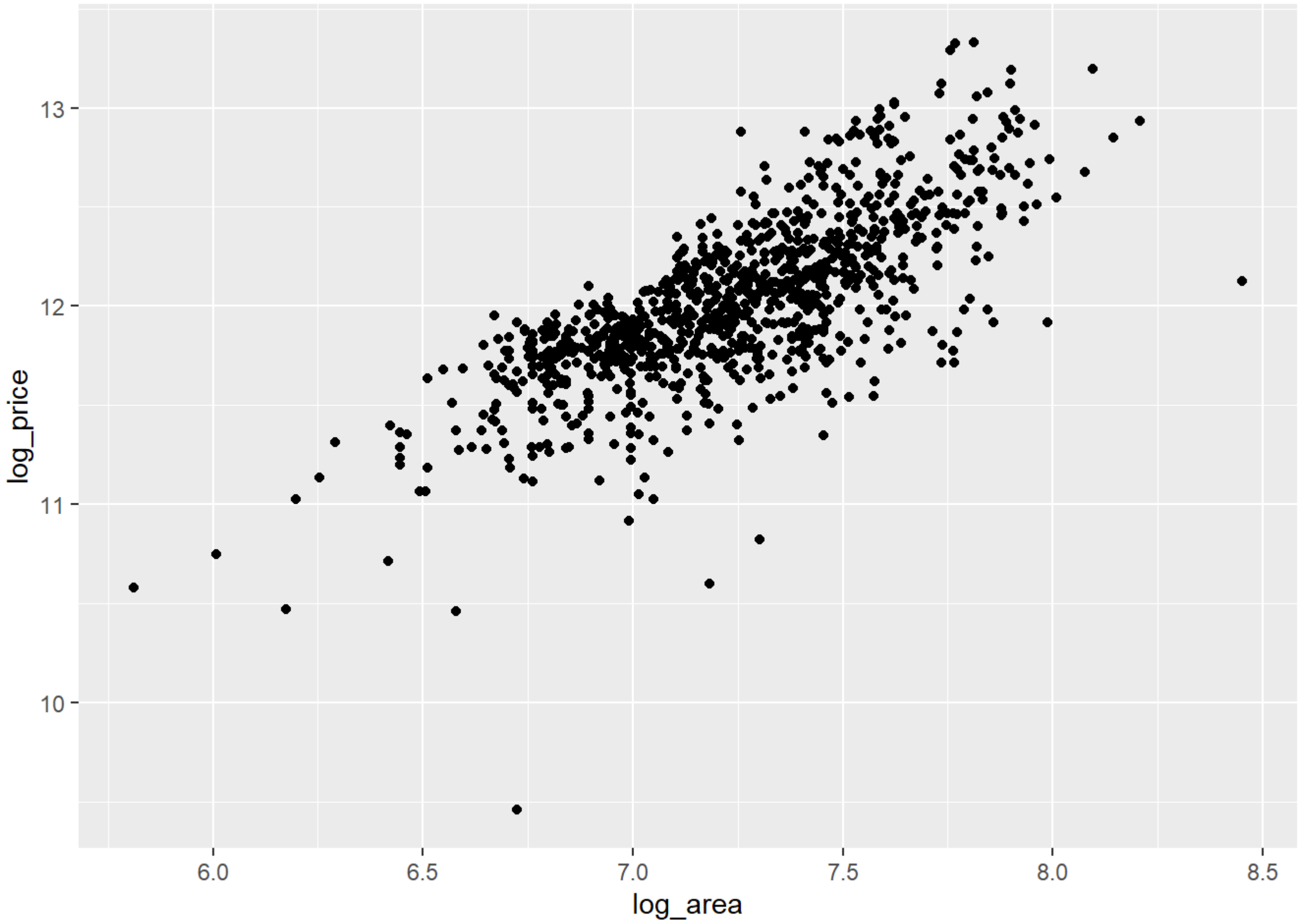
```
df$Exter.Qual<-factor(df$Exter.Qual,ordered=T,levels=c('Po','Fa','TA','Gd','Ex'))
df$Exter.Cond<-factor(df$Exter.Cond,ordered=T,levels=c('Po','Fa','TA','Gd','Ex'))
df$Heating.QC<-factor(df$Heating.QC,ordered=T,levels=c('Po','Fa','TA','Gd','Ex'))
df$Kitchen.Qual<-factor(df$Kitchen.Qual,ordered=T,levels=c('Po','Fa','TA','Gd','Ex'))
df$Functional<-factor(df$Functional,ordered=T,levels=c('Sal','Sev','Maj2','Maj1','Mod','Min2','M
in1','Typ'))
```

Based on the EDA which we have conducted, here is a initial model which will be updated later: I have included the following variables: Lot.Shape, Neighborhood, Overall.Qual, log_area, Foundation, X1st.Flr.SF, Total.Bsmt, Central.Air, Bedroom.AbvGr, Sale.Condition:

```
model_ini<-lm(data=df,log_price~Lot.Shape+Overall.Qual+Neighborhood+Bldg.Type+log_area+Bsmt.Cond
+Central.Air+Fireplace.Qu+Garage.Qual+Sale.Condition)
```

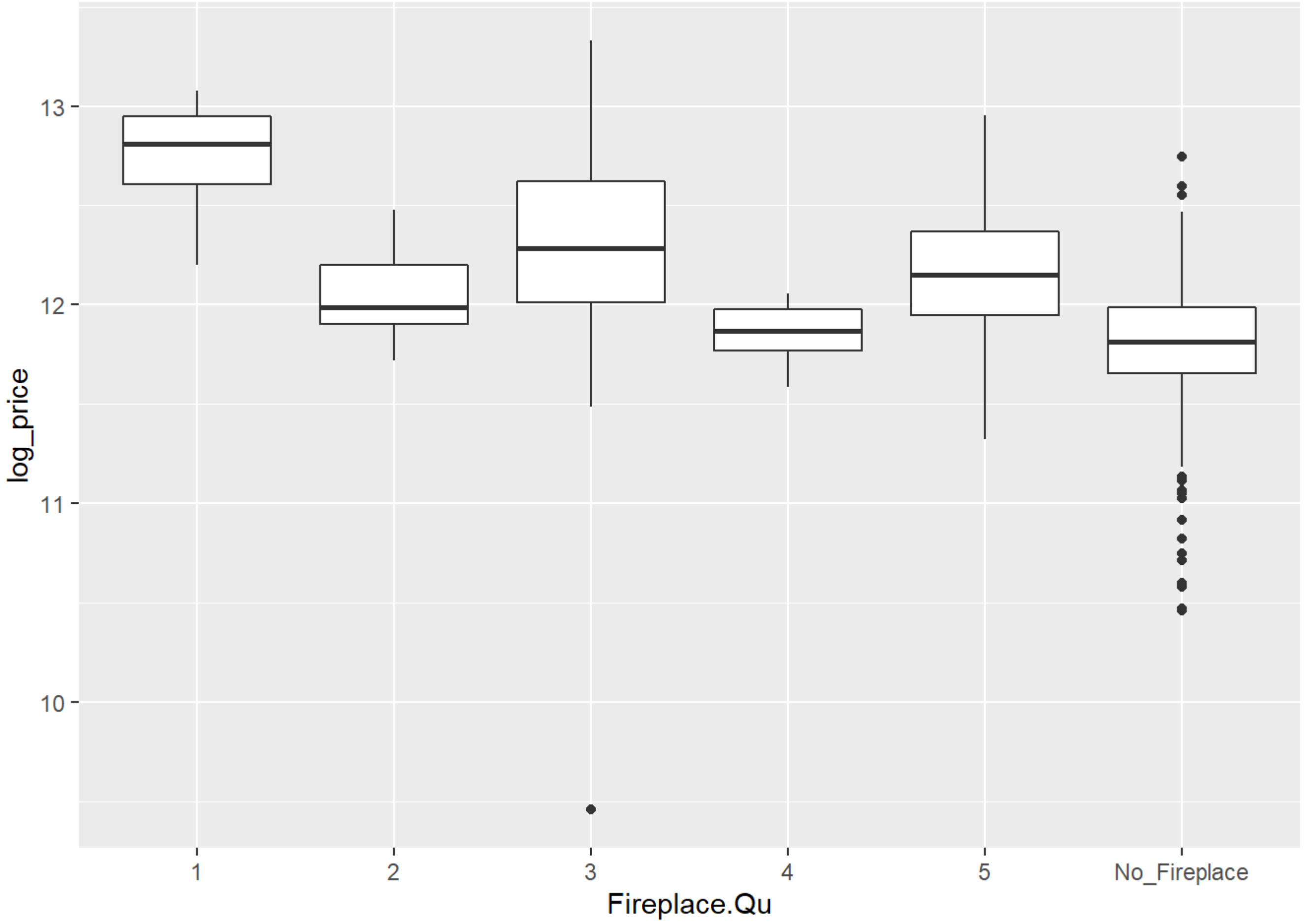
Reasons for including these variables: Based on the EDA shown, Lot.Shape, Central.Air, Overall.Qual, were found to be useful predictors. Further, log_area has a strong correlation with log_price as shown:

```
df%>%ggplot(aes(x=log_area,y=log_price))+geom_point()
```



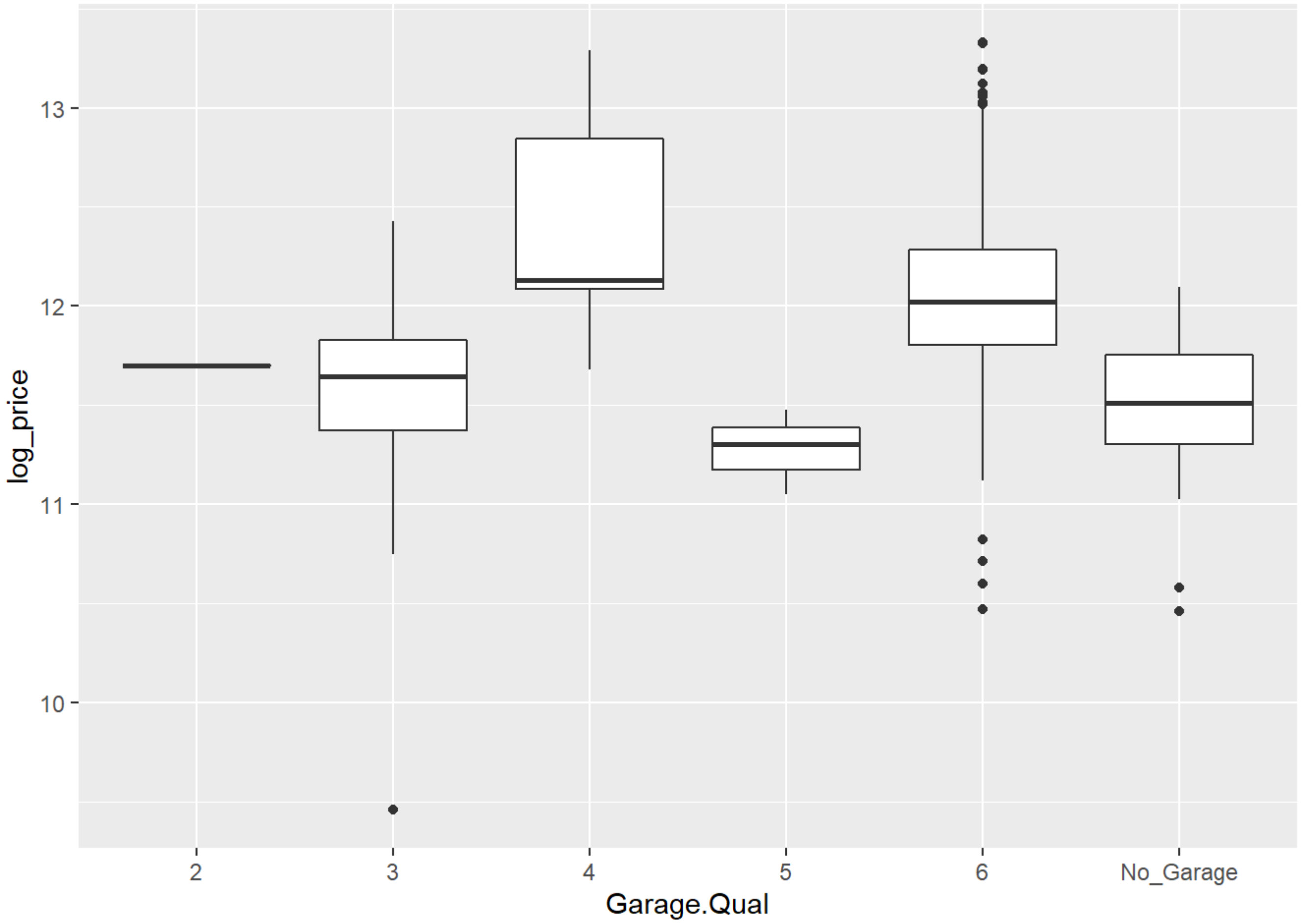
Price was seen to vary with Neighborhood in the earlier quizzes. Price was seen to vary with Fireplace.Qu as shown:

```
df%>%ggplot(aes(x=Fireplace.Qu,y=log_price))+geom_boxplot()
```



Similarly, Garage.Qual and log_price:

```
df%>%ggplot(aes(x=Garage.Qual,y=log_price))+geom_boxplot()
```



These predictors were found to be influential using initial estimates. Also, since we are required to select only 10 variables, each variable from broad categories such as quality, basement, garage, area etc was selected. A more exhaustive model will be developed later:

Let us check out the summary of our model:

```
summary(model_ini)
```

```
##
## Call:
## lm(formula = log_price ~ Lot.Shape + Overall.Qual + Neighborhood +
##     Bldg.Type + log_area + Bsmt.Cond + Central.Air + Fireplace.Qu +
##     Garage.Qual + Sale.Condition, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.37734 -0.07772  0.00071  0.08293  0.59623
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               8.668660   0.254623  34.045  < 2e-16 ***
## Lot.ShapeIR2              0.078071   0.031132   2.508 0.012317 *
## Lot.ShapeIR3              0.229113   0.092783   2.469 0.013713 *
## Lot.ShapeReg             -0.007336   0.011819  -0.621 0.534963
## Overall.Qual              0.095127   0.006597  14.421  < 2e-16 ***
## NeighborhoodBlueste      -0.126871   0.104688  -1.212 0.225856
## NeighborhoodBrDale       -0.242945   0.076268  -3.185 0.001493 **
## NeighborhoodBrkSide      -0.222392   0.059131  -3.761 0.000180 ***
## NeighborhoodClearCr       0.023176   0.069542   0.333 0.739001
## NeighborhoodCollgCr      -0.028832   0.055165  -0.523 0.601342
## NeighborhoodCrawfor      -0.009470   0.059631  -0.159 0.873858
## NeighborhoodEdwards      -0.151708   0.056464  -2.687 0.007341 **
## NeighborhoodGilbert      -0.095104   0.056963  -1.670 0.095335 .
## NeighborhoodGreens        0.081834   0.093383   0.876 0.381076
## NeighborhoodGrnHill       0.496095   0.122623   4.046 5.64e-05 ***
## NeighborhoodIDOTRR       -0.373709   0.061311  -6.095 1.59e-09 ***
## NeighborhoodMeadowV      -0.252293   0.065500  -3.852 0.000125 ***
## NeighborhoodMitchel      -0.018262   0.057622  -0.317 0.751375
## NeighborhoodNAmes        -0.110566   0.055034  -2.009 0.044819 *
## NeighborhoodNoRidge       0.080896   0.060460   1.338 0.181219
## NeighborhoodNPkVill      -0.103043   0.096038  -1.073 0.283571
## NeighborhoodNridgHt       0.176790   0.054441   3.247 0.001206 **
## NeighborhoodNWAmes       -0.086491   0.058224  -1.485 0.137749
## NeighborhoodOldTown      -0.277647   0.057345  -4.842 1.50e-06 ***
## NeighborhoodSawyer       -0.103677   0.057307  -1.809 0.070746 .
## NeighborhoodSawyerW      -0.085037   0.056865  -1.495 0.135139
## NeighborhoodSomerst       0.043642   0.052705   0.828 0.407851
## NeighborhoodStoneBr       0.175128   0.060851   2.878 0.004093 **
## NeighborhoodSWISU        -0.264208   0.070563  -3.744 0.000192 ***
## NeighborhoodTimber        0.071571   0.063845   1.121 0.262572
## NeighborhoodVeenker       0.019551   0.070750   0.276 0.782349
## Bldg.Type2fmCon           0.044119   0.038256   1.153 0.249094
## Bldg.TypeDuplex          -0.052096   0.031210  -1.669 0.095409 .
## Bldg.TypeTwnhs           -0.186912   0.036379  -5.138 3.38e-07 ***
## Bldg.TypeTwnhsE          -0.097455   0.023215  -4.198 2.95e-05 ***
## log_area                  0.409667   0.023384  17.519  < 2e-16 ***
## Bsmt.Cond3               -0.167484   0.119461  -1.402 0.161245
## Bsmt.Cond4                0.008319   0.116363   0.071 0.943022
## Bsmt.Cond5                0.118346   0.197793   0.598 0.549763
## Bsmt.Cond6               -0.003430   0.114245  -0.030 0.976052
## Bsmt.CondNo_Bsmt         -0.173088   0.119693  -1.446 0.148483    
## Central.AirY              0.174233   0.027709   6.288 4.91e-10 ***
## Fireplace.Qu2            -0.037355   0.054035  -0.691 0.489534    
## Fireplace.Qu3            -0.046949   0.042079  -1.116 0.264815    
## Fireplace.Qu4            -0.065867   0.057785  -1.140 0.254627    
## Fireplace.Qu5            -0.071971   0.043997  -1.636 0.102213    
## Fireplace.QuNo_Fireplace -0.095788   0.044222  -2.166 0.030558 *  
## Garage.Qual3             -0.280349   0.161837  -1.732 0.083549 .  
## Garage.Qual4             -0.160834   0.170706  -0.942 0.346349    
## Garage.Qual5             -0.455157   0.185093  -2.459 0.014109 *  
## Garage.Qual6             -0.260361   0.159744  -1.630 0.103465    
## Garage.QualNo_Garage     -0.293847   0.161443  -1.820 0.069058 .  
## Sale.ConditionAdjLand     0.187525   0.119602   1.568 0.117238    
## Sale.ConditionAlloca      0.207006   0.095723   2.163 0.030826 *  
## Sale.ConditionFamily     -0.079238   0.044025  -1.800 0.072210 .  
## Sale.ConditionNormal      0.085255   0.021509   3.964 7.94e-05 ***
## Sale.ConditionPartial     0.149934   0.029676   5.052 5.24e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.156 on 941 degrees of freedom
## Multiple R-squared:  0.8701, Adjusted R-squared:  0.8623 
## F-statistic: 112.5 on 56 and 941 DF,  p-value: < 2.2e-16
```

We can see that we are getting an overall adjusted R-squared of 0.8623 which is good for a start! For the F-statistic of 112.5, we are getting a p-value of <2.2e-16 which means that the model is statistically significant as a whole. From the results of the model, we can see that Bsmt.Cond is not a statistically significant predictor having all p-values greater than the significance level.

Intepretation of coefficients:

1. Overall.Qual: All else held constant, a unit increase in Overall.Qual will lead to an increase of 0.09513 in log_price
2. Central.AirY: All else held constant, on average, houses which do have Central Air Conditioning have a log_price 0.1742 more than those who do not.

Similar interpretations follow for the rest of the coefficients. The interpretation of intercept is sometimes meaningless in context.

2.2.2 Section 2.2 Model Selection

Now either using `BAS` another stepwise selection procedure choose the “best” model you can, using your initial model as your starting point. Try at least two different model selection methods and compare their results. Do they both arrive at the same model or do they disagree? What do you think this means?

Let us do model selection using different techniques, starting with backwards elimination by AIC:

```
model_ini_AIC<-step(model_ini,k=2)
```

```
## Start:  AIC=-3653.41
## log_price ~ Lot.Shape + Overall.Qual + Neighborhood + Bldg.Type +
##     log_area + Bsmt.Cond + Central.Air + Fireplace.Qu + Garage.Qual +
##     Sale.Condition
##
##                  Df Sum of Sq    RSS     AIC
## <none>                        22.892 -3653.4
## - Garage.Qual     5    0.2597 23.152 -3652.2
## - Fireplace.Qu    5    0.3076 23.200 -3650.1
## - Lot.Shape       3    0.3340 23.226 -3645.0
## - Bldg.Type       4    0.9142 23.806 -3622.3
## - Bsmt.Cond       5    0.9776 23.870 -3621.7
## - Sale.Condition  5    1.1144 24.006 -3616.0
## - Central.Air     1    0.9619 23.854 -3614.3
## - Overall.Qual    1    5.0592 27.951 -3456.1
## - Neighborhood   26    8.3603 31.252 -3394.7
## - log_area        1    7.4667 30.359 -3373.7
```

```
summary(model_ini_AIC)
```

```
##
## Call:
## lm(formula = log_price ~ Lot.Shape + Overall.Qual + Neighborhood +
##     Bldg.Type + log_area + Bsmt.Cond + Central.Air + Fireplace.Qu +
##     Garage.Qual + Sale.Condition, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.37734 -0.07772  0.00071  0.08293  0.59623
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                8.668660   0.254623  34.045  < 2e-16 ***
## Lot.ShapeIR2               0.078071   0.031132   2.508 0.012317 *
## Lot.ShapeIR3               0.229113   0.092783   2.469 0.013713 *
## Lot.ShapeReg              -0.007336   0.011819  -0.621 0.534963
## Overall.Qual               0.095127   0.006597  14.421  < 2e-16 ***
## NeighborhoodBlueste       -0.126871   0.104688  -1.212 0.225856
## NeighborhoodBrDale        -0.242945   0.076268  -3.185 0.001493 **
## NeighborhoodBrkSide       -0.222392   0.059131  -3.761 0.000180 ***
## NeighborhoodClearCr        0.023176   0.069542   0.333 0.739001
## NeighborhoodCollgCr       -0.028832   0.055165  -0.523 0.601342
## NeighborhoodCrawfor       -0.009470   0.059631  -0.159 0.873858
## NeighborhoodEdwards       -0.151708   0.056464  -2.687 0.007341 **
## NeighborhoodGilbert       -0.095104   0.056963  -1.670 0.095335 .
## NeighborhoodGreens         0.081834   0.093383   0.876 0.381076
## NeighborhoodGrnHill        0.496095   0.122623   4.046 5.64e-05 ***
## NeighborhoodIDOTRR        -0.373709   0.061311  -6.095 1.59e-09 ***
## NeighborhoodMeadowV       -0.252293   0.065500  -3.852 0.000125 ***
## NeighborhoodMitchel       -0.018262   0.057622  -0.317 0.751375
## NeighborhoodNAmes         -0.110566   0.055034  -2.009 0.044819 *
## NeighborhoodNoRidge        0.080896   0.060460   1.338 0.181219
## NeighborhoodNPkVill       -0.103043   0.096038  -1.073 0.283571
## NeighborhoodNridgHt        0.176790   0.054441   3.247 0.001206 **
## NeighborhoodNWAmes        -0.086491   0.058224  -1.485 0.137749
## NeighborhoodOldTown       -0.277647   0.057345  -4.842 1.50e-06 ***
## NeighborhoodSawyer        -0.103677   0.057307  -1.809 0.070746 .
## NeighborhoodSawyerW       -0.085037   0.056865  -1.495 0.135139
## NeighborhoodSomerst        0.043642   0.052705   0.828 0.407851
## NeighborhoodStoneBr        0.175128   0.060851   2.878 0.004093 **
## NeighborhoodSWISU         -0.264208   0.070563  -3.744 0.000192 ***
## NeighborhoodTimber         0.071571   0.063845   1.121 0.262572
## NeighborhoodVeenker        0.019551   0.070750   0.276 0.782349
## Bldg.Type2fmCon            0.044119   0.038256   1.153 0.249094
## Bldg.TypeDuplex           -0.052096   0.031210  -1.669 0.095409 .
## Bldg.TypeTwnhs            -0.186912   0.036379  -5.138 3.38e-07 ***
## Bldg.TypeTwnhsE           -0.097455   0.023215  -4.198 2.95e-05 ***
## log_area                   0.409667   0.023384  17.519  < 2e-16 ***
## Bsmt.Cond3                -0.167484   0.119461  -1.402 0.161245
## Bsmt.Cond4                 0.008319   0.116363   0.071 0.943022
## Bsmt.Cond5                 0.118346   0.197793   0.598 0.549763
## Bsmt.Cond6                -0.003430   0.114245  -0.030 0.976052
## Bsmt.CondNo_Bsmt         -0.173088   0.119693  -1.446 0.148483    
## Central.AirY              0.174233   0.027709   6.288 4.91e-10 ***
## Fireplace.Qu2            -0.037355   0.054035  -0.691 0.489534    
## Fireplace.Qu3            -0.046949   0.042079  -1.116 0.264815    
## Fireplace.Qu4            -0.065867   0.057785  -1.140 0.254627    
## Fireplace.Qu5            -0.071971   0.043997  -1.636 0.102213    
## Fireplace.QuNo_Fireplace -0.095788   0.044222  -2.166 0.030558 *  
## Garage.Qual3             -0.280349   0.161837  -1.732 0.083549 .  
## Garage.Qual4             -0.160834   0.170706  -0.942 0.346349    
## Garage.Qual5             -0.455157   0.185093  -2.459 0.014109 *  
## Garage.Qual6             -0.260361   0.159744  -1.630 0.103465    
## Garage.QualNo_Garage     -0.293847   0.161443  -1.820 0.069058 .  
## Sale.ConditionAdjLand     0.187525   0.119602   1.568 0.117238    
## Sale.ConditionAlloca      0.207006   0.095723   2.163 0.030826 *  
## Sale.ConditionFamily     -0.079238   0.044025  -1.800 0.072210 .  
## Sale.ConditionNormal      0.085255   0.021509   3.964 7.94e-05 ***
## Sale.ConditionPartial     0.149934   0.029676   5.052 5.24e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.156 on 941 degrees of freedom
## Multiple R-squared:  0.8701, Adjusted R-squared:  0.8623 
## F-statistic: 112.5 on 56 and 941 DF,  p-value: < 2.2e-16
```

We can see that no variable has been eliminated using the AIC criterion.

Now, let us do the same thing using the BIC criterion or AIC With k=log(n):

```
model_ini_BIC<-step(model_ini,k=log(998))
```

```
## Start:  AIC=-3373.78
## log_price ~ Lot.Shape + Overall.Qual + Neighborhood + Bldg.Type + 
##     log_area + Bsmt.Cond + Central.Air + Fireplace.Qu + Garage.Qual + 
##     Sale.Condition
## 
##                  Df Sum of Sq    RSS     AIC
## - Garage.Qual     5    0.2597 23.152 -3397.1
## - Fireplace.Qu    5    0.3076 23.200 -3395.0
## - Lot.Shape       3    0.3340 23.226 -3380.0
## <none>                        22.892 -3373.8
## - Bsmt.Cond       5    0.9776 23.870 -3366.6
## - Bldg.Type       4    0.9142 23.806 -3362.3
## - Sale.Condition  5    1.1144 24.006 -3360.9
## - Central.Air     1    0.9619 23.854 -3339.6
## - Neighborhood   26    8.3603 31.252 -3242.7
## - Overall.Qual    1    5.0592 27.951 -3181.4
## - log_area        1    7.4667 30.359 -3099.0
## 
## Step:  AIC=-3397.05
## log_price ~ Lot.Shape + Overall.Qual + Neighborhood + Bldg.Type + 
##     log_area + Bsmt.Cond + Central.Air + Fireplace.Qu + Sale.Condition
## 
##                  Df Sum of Sq    RSS     AIC
## - Fireplace.Qu    5    0.3705 23.522 -3415.7
## - Lot.Shape       3    0.3223 23.474 -3404.0
## <none>                        23.152 -3397.1
## - Bsmt.Cond       5    0.9169 24.069 -3392.8
## - Bldg.Type       4    0.9116 24.063 -3386.1
## - Sale.Condition  5    1.1132 24.265 -3384.7
## - Central.Air     1    1.5198 24.672 -3340.5
## - Neighborhood   26    8.5071 31.659 -3264.3
## - Overall.Qual    1    5.1595 28.311 -3203.2
## - log_area        1    7.5201 30.672 -3123.2
## 
## Step:  AIC=-3415.73
## log_price ~ Lot.Shape + Overall.Qual + Neighborhood + Bldg.Type + 
##     log_area + Bsmt.Cond + Central.Air + Sale.Condition
## 
##                  Df Sum of Sq    RSS     AIC
## - Lot.Shape       3    0.3798 23.902 -3420.5
## <none>                        23.522 -3415.7
## - Bsmt.Cond       5    0.8430 24.365 -3415.1
## - Sale.Condition  5    1.1164 24.639 -3404.0
## - Bldg.Type       4    1.1342 24.657 -3396.4
## - Central.Air     1    1.5463 25.069 -3359.1
## - Neighborhood   26    9.0644 32.587 -3270.0
## - Overall.Qual    1    5.7702 29.293 -3203.7
## - log_area        1    9.4390 32.961 -3085.9
## 
## Step:  AIC=-3420.47
## log_price ~ Overall.Qual + Neighborhood + Bldg.Type + log_area + 
##     Bsmt.Cond + Central.Air + Sale.Condition
```

```
##
##                  Df Sum of Sq    RSS     AIC
## <none>                        23.902 -3420.5
## - Bsmt.Cond       5    0.8826 24.785 -3418.8
## - Sale.Condition  5    1.1809 25.083 -3406.9
## - Bldg.Type       4    1.0954 24.998 -3403.4
## - Central.Air     1    1.5608 25.463 -3364.2
## - Neighborhood   26    9.4991 33.401 -3266.1
## - Overall.Qual    1    5.7541 29.656 -3212.1
## - log_area        1    9.9540 33.856 -3079.9
```

```
summary(model_ini_BIC)
```

```
##
## Call:
## lm(formula = log_price ~ Overall.Qual + Neighborhood + Bldg.Type +
##     log_area + Bsmt.Cond + Central.Air + Sale.Condition, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.31113 -0.08192  0.00172  0.08165  0.63493
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            8.094834   0.189583  42.698  < 2e-16 ***
## Overall.Qual           0.099290   0.006552  15.155  < 2e-16 ***
## NeighborhoodBlueste   -0.111498   0.105927  -1.053 0.292792
## NeighborhoodBrDale    -0.239033   0.076642  -3.119 0.001870 **
## NeighborhoodBrkSide   -0.220491   0.059065  -3.733 0.000200 ***
## NeighborhoodClearCr    0.045593   0.069082   0.660 0.509425
## NeighborhoodCollgCr   -0.031681   0.054643  -0.580 0.562208
## NeighborhoodCrawfor   -0.007046   0.059685  -0.118 0.906046
## NeighborhoodEdwards   -0.160046   0.056349  -2.840 0.004604 **
## NeighborhoodGilbert   -0.083293   0.056650  -1.470 0.141811
## NeighborhoodGreens     0.088050   0.093582   0.941 0.347004
## NeighborhoodGrnHill    0.474060   0.123439   3.840 0.000131 ***
## NeighborhoodIDOTRR    -0.376494   0.061039  -6.168 1.02e-09 ***
## NeighborhoodMeadowV   -0.262574   0.065551  -4.006 6.67e-05 ***
## NeighborhoodMitchel   -0.021533   0.057607  -0.374 0.708643
## NeighborhoodNAmes     -0.107577   0.054882  -1.960 0.050268 .
## NeighborhoodNoRidge    0.077643   0.060501   1.283 0.199690
## NeighborhoodNPkVill   -0.080707   0.096231  -0.839 0.401861
## NeighborhoodNridgHt    0.189584   0.054195   3.498 0.000490 ***
## NeighborhoodNWAmes    -0.092132   0.058160  -1.584 0.113499
## NeighborhoodOldTown   -0.283200   0.057039  -4.965 8.14e-07 ***
## NeighborhoodSawyer    -0.100393   0.057202  -1.755 0.079572 .
## NeighborhoodSawyerW   -0.098855   0.056706  -1.743 0.081606 .
## NeighborhoodSomerst    0.035859   0.052594   0.682 0.495524
## NeighborhoodStoneBr    0.185399   0.060507   3.064 0.002245 **
## NeighborhoodSWISU     -0.264673   0.070764  -3.740 0.000195 ***
## NeighborhoodTimber     0.092657   0.063485   1.459 0.144757
## NeighborhoodVeenker    0.041914   0.070926   0.591 0.554696
## Bldg.Type2fmCon        0.051049   0.038238   1.335 0.182177
## Bldg.TypeDuplex       -0.072406   0.030919  -2.342 0.019397 *
## Bldg.TypeTwnhs        -0.201900   0.036543  -5.525 4.25e-08 ***
## Bldg.TypeTwnhsE       -0.093910   0.023301  -4.030 6.01e-05 ***
## log_area               0.439665   0.022058  19.932  < 2e-16 ***
## Bsmt.Cond3            -0.198979   0.118098  -1.685 0.092344 .
## Bsmt.Cond4            -0.021839   0.115319  -0.189 0.849840
## Bsmt.Cond5             0.027765   0.196783   0.141 0.887823
## Bsmt.Cond6            -0.038917   0.112883  -0.345 0.730356
## Bsmt.CondNo_Bsmt      -0.190265   0.118771  -1.602 0.109499
## Central.AirY           0.202921   0.025710   7.893 8.09e-15 ***
## Sale.ConditionAdjLand  0.169328   0.120017   1.411 0.158611
## Sale.ConditionAlloca   0.189525   0.096968   1.955 0.050932 .
## Sale.ConditionFamily  -0.085267   0.044456  -1.918 0.055411 .  
## Sale.ConditionNormal   0.084641   0.021666   3.907 0.000100 ***
## Sale.ConditionPartial  0.155892   0.029897   5.214 2.26e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1583 on 954 degrees of freedom
## Multiple R-squared:  0.8643, Adjusted R-squared:  0.8582 
## F-statistic: 141.3 on 43 and 954 DF,  p-value: < 2.2e-16
```

Using the BIC step-wise selection, we can see that Lot.Shape, Fireplace.Qu, Garage.Qual have been eliminated.

Now let us try model selection using backwards elimination and the p-value method:

```
summary(model_ini)
```

```
## 
## Call:
## lm(formula = log_price ~ Lot.Shape + Overall.Qual + Neighborhood + 
##     Bldg.Type + log_area + Bsmt.Cond + Central.Air + Fireplace.Qu + 
##     Garage.Qual + Sale.Condition, data = df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.37734 -0.07772  0.00071  0.08293  0.59623 
## 
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            8.668660   0.254623  34.045  < 2e-16 ***
## Lot.ShapeIR2           0.078071   0.031132   2.508 0.012317 *  
## Lot.ShapeIR3           0.229113   0.092783   2.469 0.013713 *  
## Lot.ShapeReg          -0.007336   0.011819  -0.621 0.534963    
## Overall.Qual           0.095127   0.006597  14.421  < 2e-16 ***
## NeighborhoodBlueste   -0.126871   0.104688  -1.212 0.225856    
## NeighborhoodBrDale    -0.242945   0.076268  -3.185 0.001493 ** 
## NeighborhoodBrkSide   -0.222392   0.059131  -3.761 0.000180 ***
## NeighborhoodClearCr    0.023176   0.069542   0.333 0.739001    
## NeighborhoodCollgCr   -0.028832   0.055165  -0.523 0.601342    
## NeighborhoodCrawfor   -0.009470   0.059631  -0.159 0.873858    
## NeighborhoodEdwards   -0.151708   0.056464  -2.687 0.007341 ** 
## NeighborhoodGilbert   -0.095104   0.056963  -1.670 0.095335 .  
## NeighborhoodGreens     0.081834   0.093383   0.876 0.381076    
## NeighborhoodGrnHill    0.496095   0.122623   4.046 5.64e-05 ***
## NeighborhoodIDOTRR    -0.373709   0.061311  -6.095 1.59e-09 ***
## NeighborhoodMeadowV   -0.252293   0.065500  -3.852 0.000125 ***
## NeighborhoodMitchel   -0.018262   0.057622  -0.317 0.751375    
## NeighborhoodNAmes     -0.110566   0.055034  -2.009 0.044819 *  
## NeighborhoodNoRidge    0.080896   0.060460   1.338 0.181219    
## NeighborhoodNPkVill   -0.103043   0.096038  -1.073 0.283571    
## NeighborhoodNridgHt    0.176790   0.054441   3.247 0.001206 ** 
## NeighborhoodNWAmes    -0.086491   0.058224  -1.485 0.137749    
## NeighborhoodOldTown   -0.277647   0.057345  -4.842 1.50e-06 ***
## NeighborhoodSawyer    -0.103677   0.057307  -1.809 0.070746 .  
## NeighborhoodSawyerW   -0.085037   0.056865  -1.495 0.135139    
## NeighborhoodSomerst    0.043642   0.052705   0.828 0.407851    
## NeighborhoodStoneBr    0.175128   0.060851   2.878 0.004093 ** 
## NeighborhoodSWISU     -0.264208   0.070563  -3.744 0.000192 ***
## NeighborhoodTimber     0.071571   0.063845   1.121 0.262572    
## NeighborhoodVeenker    0.019551   0.070750   0.276 0.782349    
## Bldg.Type2fmCon        0.044119   0.038256   1.153 0.249094    
## Bldg.TypeDuplex       -0.052096   0.031210  -1.669 0.095409 .  
## Bldg.TypeTwnhs        -0.186912   0.036379  -5.138 3.38e-07 ***
## Bldg.TypeTwnhsE       -0.097455   0.023215  -4.198 2.95e-05 ***
## log_area               0.409667   0.023384  17.519  < 2e-16 ***
## Bsmt.Cond3            -0.167484   0.119461  -1.402 0.161245    
## Bsmt.Cond4             0.008319   0.116363   0.071 0.943022    
## Bsmt.Cond5             0.118346   0.197793   0.598 0.549763    
## Bsmt.Cond6            -0.003430   0.114245  -0.030 0.976052    
## Bsmt.CondNo_Bsmt         -0.173088   0.119693  -1.446 0.148483    
## Central.AirY              0.174233   0.027709   6.288 4.91e-10 ***
## Fireplace.Qu2            -0.037355   0.054035  -0.691 0.489534    
## Fireplace.Qu3            -0.046949   0.042079  -1.116 0.264815    
## Fireplace.Qu4            -0.065867   0.057785  -1.140 0.254627    
## Fireplace.Qu5            -0.071971   0.043997  -1.636 0.102213    
## Fireplace.QuNo_Fireplace -0.095788   0.044222  -2.166 0.030558 *  
## Garage.Qual3             -0.280349   0.161837  -1.732 0.083549 .  
## Garage.Qual4             -0.160834   0.170706  -0.942 0.346349    
## Garage.Qual5             -0.455157   0.185093  -2.459 0.014109 *  
## Garage.Qual6             -0.260361   0.159744  -1.630 0.103465    
## Garage.QualNo_Garage     -0.293847   0.161443  -1.820 0.069058 .  
## Sale.ConditionAdjLand     0.187525   0.119602   1.568 0.117238    
## Sale.ConditionAlloca      0.207006   0.095723   2.163 0.030826 *  
## Sale.ConditionFamily     -0.079238   0.044025  -1.800 0.072210 .  
## Sale.ConditionNormal      0.085255   0.021509   3.964 7.94e-05 ***
## Sale.ConditionPartial     0.149934   0.029676   5.052 5.24e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.156 on 941 degrees of freedom
## Multiple R-squared:  0.8701, Adjusted R-squared:  0.8623 
## F-statistic: 112.5 on 56 and 941 DF,  p-value: < 2.2e-16
```

Iteration 1 (Eliminate Bsmt.Cond):

```
model_ini_iter<-model_ini<-lm(data=df,log_price~Lot.Shape+Overall.Qual+Neighborhood+Bldg.Type+log_area+Central.Air+Fireplace.Qu+Garage.Qual+Sale.Condition)
summary(model_ini_iter)
```

```
## 
## Call:
## lm(formula = log_price ~ Lot.Shape + Overall.Qual + Neighborhood + 
##     Bldg.Type + log_area + Central.Air + Fireplace.Qu + Garage.Qual + 
##     Sale.Condition, data = df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.46217 -0.07987  0.00312  0.08201  0.57134 
## 
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            8.619454   0.233216  36.959  < 2e-16 ***
## Lot.ShapeIR2           0.082773   0.031661   2.614 0.009082 ** 
## Lot.ShapeIR3           0.244389   0.094192   2.595 0.009617 ** 
## Lot.ShapeReg          -0.007219   0.012007  -0.601 0.547831    
## Overall.Qual           0.101574   0.006622  15.339  < 2e-16 ***
## NeighborhoodBlueste   -0.122931   0.106593  -1.153 0.249090    
## NeighborhoodBrDale    -0.258957   0.077567  -3.339 0.000875 ***
## NeighborhoodBrkSide   -0.218238   0.060204  -3.625 0.000304 ***
## NeighborhoodClearCr    0.008377   0.070709   0.118 0.905714    
## NeighborhoodCollgCr   -0.027741   0.056164  -0.494 0.621475    
## NeighborhoodCrawfor   -0.006914   0.060720  -0.114 0.909371    
## NeighborhoodEdwards   -0.162116   0.057455  -2.822 0.004878 ** 
## NeighborhoodGilbert   -0.090961   0.057988  -1.569 0.117073    
## NeighborhoodGreens     0.072839   0.095069   0.766 0.443764    
## NeighborhoodGrnHill    0.408348   0.123204   3.314 0.000953 ***
## NeighborhoodIDOTRR    -0.363730   0.062227  -5.845 6.96e-09 ***
## NeighborhoodMeadowV   -0.247007   0.066664  -3.705 0.000223 ***
## NeighborhoodMitchel   -0.011928   0.058653  -0.203 0.838889    
## NeighborhoodNAmes     -0.108215   0.056029  -1.931 0.053733 .  
## NeighborhoodNoRidge    0.080310   0.061527   1.305 0.192113    
## NeighborhoodNPkVill   -0.100944   0.097713  -1.033 0.301838    
## NeighborhoodNridgHt    0.175686   0.055420   3.170 0.001573 ** 
## NeighborhoodNWAmes    -0.080848   0.059285  -1.364 0.172983    
## NeighborhoodOldTown   -0.289216   0.058320  -4.959 8.39e-07 ***
## NeighborhoodSawyer    -0.101583   0.058352  -1.741 0.082032 .  
## NeighborhoodSawyerW   -0.093043   0.057867  -1.608 0.108197    
## NeighborhoodSomerst    0.041165   0.053665   0.767 0.443225    
## NeighborhoodStoneBr    0.172094   0.061945   2.778 0.005575 ** 
## NeighborhoodSWISU     -0.264828   0.071822  -3.687 0.000240 ***
## NeighborhoodTimber     0.070106   0.065007   1.078 0.281111    
## NeighborhoodVeenker    0.018224   0.072020   0.253 0.800293    
## Bldg.Type2fmCon        0.058529   0.038833   1.507 0.132094    
## Bldg.TypeDuplex       -0.069330   0.031156  -2.225 0.026300 *  
## Bldg.TypeTwnhs        -0.186420   0.037045  -5.032 5.80e-07 ***
## Bldg.TypeTwnhsE       -0.096328   0.023623  -4.078 4.93e-05 ***
## log_area               0.402531   0.023630  17.035  < 2e-16 ***
## Central.AirY           0.216253   0.027341   7.910 7.18e-15 ***
## Fireplace.Qu2         -0.013507   0.054581  -0.247 0.804596    
## Fireplace.Qu3         -0.034833   0.042710  -0.816 0.414952    
## Fireplace.Qu4         -0.053761   0.058367  -0.921 0.357240    
## Fireplace.Qu5            -0.056328   0.044650  -1.262 0.207427    
## Fireplace.QuNo_Fireplace -0.076528   0.044828  -1.707 0.088123 .  
## Garage.Qual3             -0.279375   0.164748  -1.696 0.090258 .  
## Garage.Qual4             -0.163574   0.173788  -0.941 0.346831    
## Garage.Qual5             -0.407267   0.188306  -2.163 0.030806 *  
## Garage.Qual6             -0.262536   0.162631  -1.614 0.106795    
## Garage.QualNo_Garage     -0.281964   0.164196  -1.717 0.086262 .  
## Sale.ConditionAdjLand     0.221121   0.121613   1.818 0.069346 .  
## Sale.ConditionAlloca      0.171246   0.096785   1.769 0.077159 .  
## Sale.ConditionFamily     -0.075445   0.044768  -1.685 0.092275 .  
## Sale.ConditionNormal      0.082071   0.021789   3.767 0.000176 ***
## Sale.ConditionPartial     0.148116   0.030118   4.918 1.03e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1588 on 946 degrees of freedom
## Multiple R-squared:  0.8645, Adjusted R-squared:  0.8572 
## F-statistic: 118.4 on 51 and 946 DF,  p-value: < 2.2e-16
```

Iteration 2 (Eliminate Fireplace.Qu):

```
model_ini_iter<-lm(data=df,log_price~Lot.Shape+Overall.Qual+Neighborhood+Bldg.Type+log_area+Central.Air+Garage.Qual+Sale.Condition)
summary(model_ini_iter)
```

```
## 
## Call:
## lm(formula = log_price ~ Lot.Shape + Overall.Qual + Neighborhood + 
##     Bldg.Type + log_area + Central.Air + Garage.Qual + Sale.Condition, 
##     data = df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.41490 -0.07917  0.00349  0.08418  0.57879 
## 
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           8.484303   0.223984  37.879  < 2e-16 ***
## Lot.ShapeIR2          0.087504   0.031531   2.775 0.005625 ** 
## Lot.ShapeIR3          0.257569   0.094065   2.738 0.006293 ** 
## Lot.ShapeReg         -0.008629   0.012014  -0.718 0.472779    
## Overall.Qual          0.104329   0.006544  15.942  < 2e-16 ***
## NeighborhoodBlueste  -0.118527   0.106594  -1.112 0.266443    
## NeighborhoodBrDale   -0.265188   0.077135  -3.438 0.000611 ***
## NeighborhoodBrkSide  -0.232053   0.059682  -3.888 0.000108 ***
## NeighborhoodClearCr  -0.004778   0.070598  -0.068 0.946059    
## NeighborhoodCollgCr  -0.049198   0.055520  -0.886 0.375771    
## NeighborhoodCrawfor  -0.010221   0.060699  -0.168 0.866311    
## NeighborhoodEdwards  -0.173816   0.056867  -3.057 0.002302 ** 
## NeighborhoodGilbert  -0.102815   0.057934  -1.775 0.076267 .  
## NeighborhoodGreens    0.066454   0.094924   0.700 0.484051    
## NeighborhoodGrnHill   0.388833   0.123259   3.155 0.001658 ** 
## NeighborhoodIDOTRR   -0.380758   0.061568  -6.184 9.25e-10 ***
## NeighborhoodMeadowV  -0.252540   0.066588  -3.793 0.000158 ***
## NeighborhoodMitchel  -0.025542   0.058255  -0.438 0.661159    
## NeighborhoodNAmes    -0.120903   0.055609  -2.174 0.029939 *  
## NeighborhoodNoRidge   0.064088   0.061211   1.047 0.295367    
## NeighborhoodNPkVill  -0.087152   0.096839  -0.900 0.368360    
## NeighborhoodNridgHt   0.179014   0.054767   3.269 0.001119 ** 
## NeighborhoodNWAmes   -0.098741   0.058981  -1.674 0.094435 .  
## NeighborhoodOldTown  -0.305629   0.057616  -5.305 1.41e-07 ***
## NeighborhoodSawyer   -0.112297   0.058002  -1.936 0.053153 .  
## NeighborhoodSawyerW  -0.114996   0.057295  -2.007 0.045022 *  
## NeighborhoodSomerst   0.025904   0.053112   0.488 0.625855    
## NeighborhoodStoneBr   0.166967   0.061616   2.710 0.006854 ** 
## NeighborhoodSWISU    -0.281164   0.071376  -3.939 8.77e-05 ***
## NeighborhoodTimber    0.060830   0.064838   0.938 0.348393    
## NeighborhoodVeenker   0.014012   0.072073   0.194 0.845894    
## Bldg.Type2fmCon       0.058751   0.038894   1.511 0.131231    
## Bldg.TypeDuplex      -0.080928   0.030876  -2.621 0.008906 ** 
## Bldg.TypeTwnhs       -0.200038   0.036809  -5.434 6.98e-08 ***
## Bldg.TypeTwnhsE      -0.101233   0.023574  -4.294 1.93e-05 ***
## log_area              0.418636   0.022257  18.809  < 2e-16 ***
## Central.AirY          0.214045   0.027386   7.816 1.44e-14 ***
## Garage.Qual3         -0.322296   0.164288  -1.962 0.050080 .  
## Garage.Qual4         -0.196589   0.173535  -1.133 0.257564    
## Garage.Qual5         -0.460199   0.187661  -2.452 0.014374 *  
## Garage.Qual6          -0.303060   0.162215  -1.868 0.062033 .
## Garage.QualNo_Garage  -0.326644   0.163661  -1.996 0.046234 *
## Sale.ConditionAdjLand  0.217684   0.121889   1.786 0.074432 .
## Sale.ConditionAlloca   0.173135   0.097001   1.785 0.074601 .
## Sale.ConditionFamily  -0.074578   0.044851  -1.663 0.096682 .
## Sale.ConditionNormal   0.082057   0.021811   3.762 0.000179 ***
## Sale.ConditionPartial  0.149010   0.030083   4.953 8.63e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1592 on 951 degrees of freedom
## Multiple R-squared:  0.8631, Adjusted R-squared:  0.8565
## F-statistic: 130.4 on 46 and 951 DF,  p-value: < 2.2e-16
```

Thus, we arrive at three different models using the AIC, BIC, p-value methods. Out of the three models, BIC leads to the most parsimonious model. Out of the three approaches, highest adjusted R-squared is obtained by the AIC approach. Let that be our preferred model.

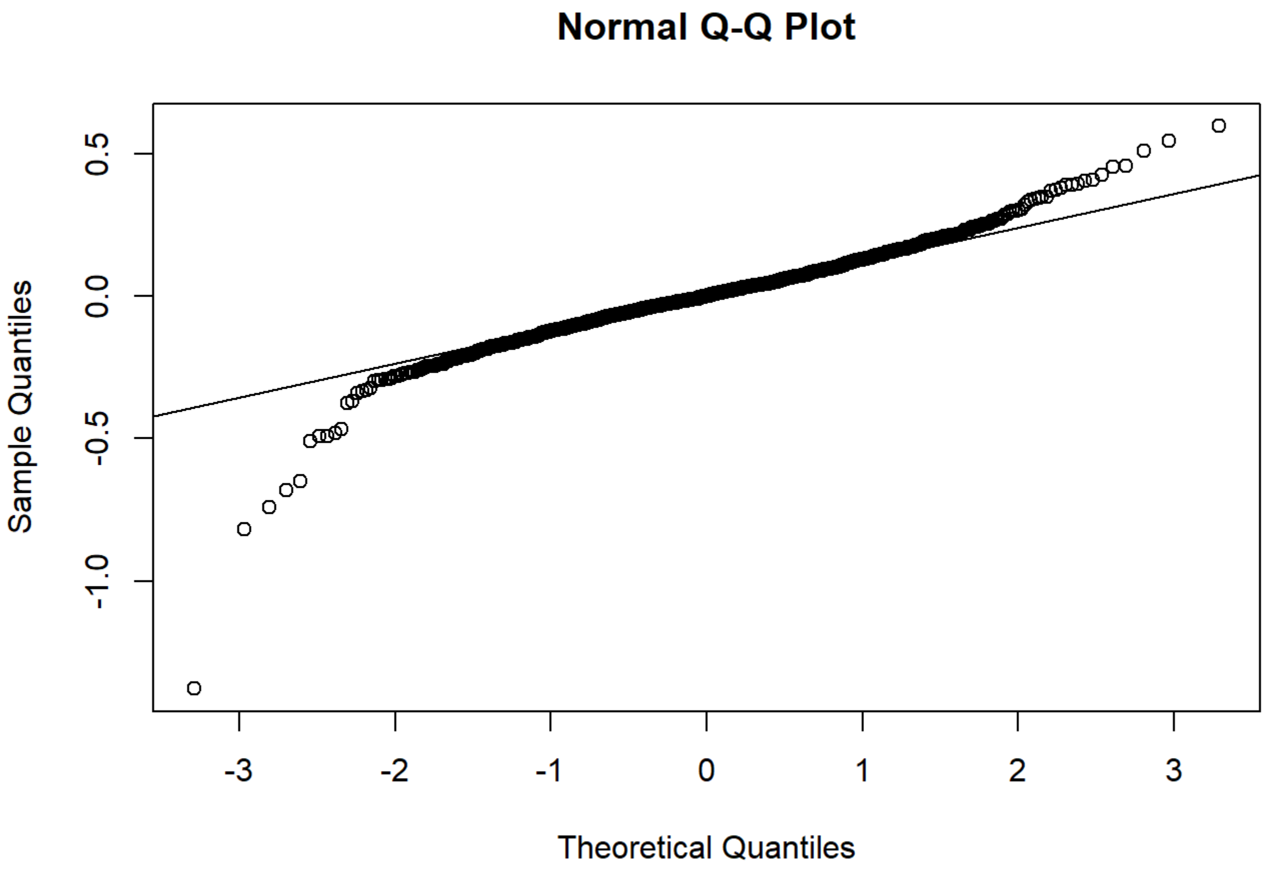
```
final_model_ini<-model_ini_AIC
```

2.2.3 Section 2.3 Initial Model Residuals

One way to assess the performance of a model is to examine the model's residuals. In the space below, create a residual plot for your preferred model from above and use it to assess whether your model appears to fit the data well. Comment on any interesting structure in the residual plot (trend, outliers, etc.) and briefly discuss potential implications it may have for your model and inference / prediction you might produce.

Let us check out the distribution of residuals of our model:

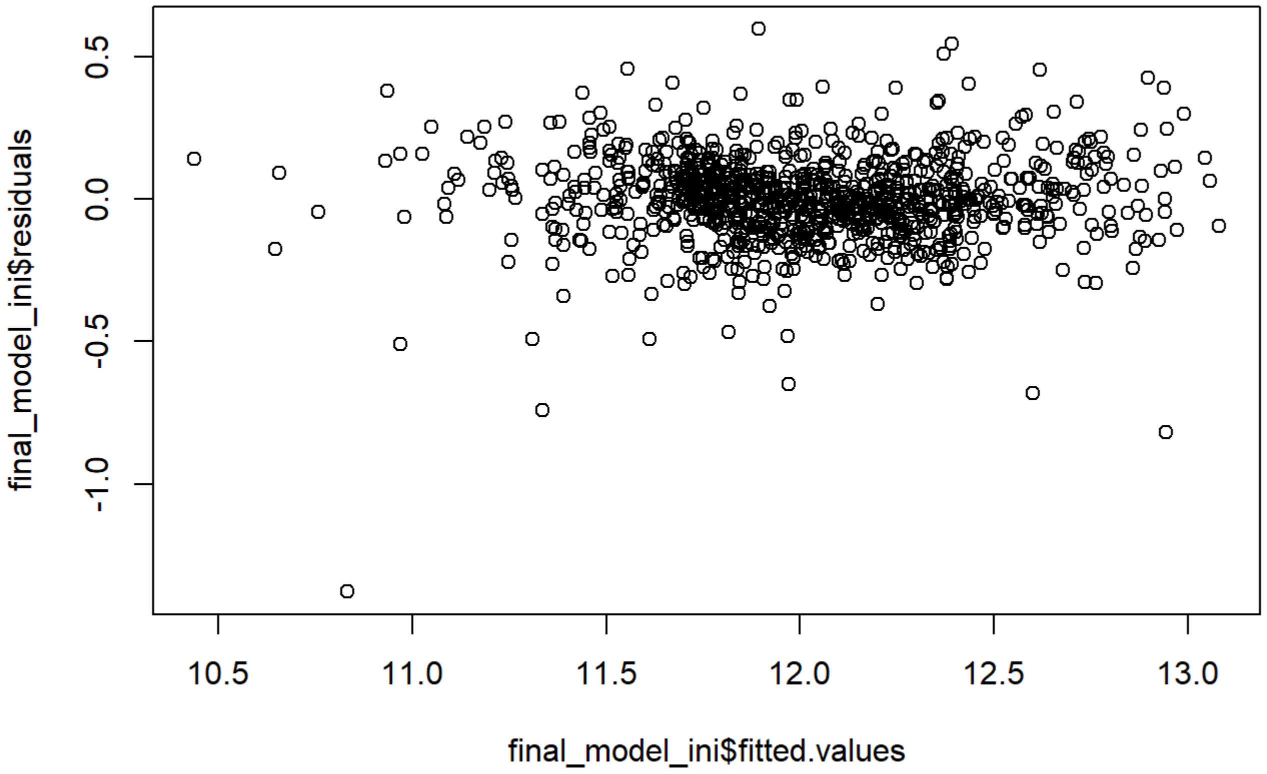
```
qqnorm(final_model_ini$residuals)
qqline(final_model_ini$residuals)
```



From this q-q plot, we can observe that the residuals deviate a little from normalcy at the ends.

Let us check out the residual plot (Residual vs fitted values):

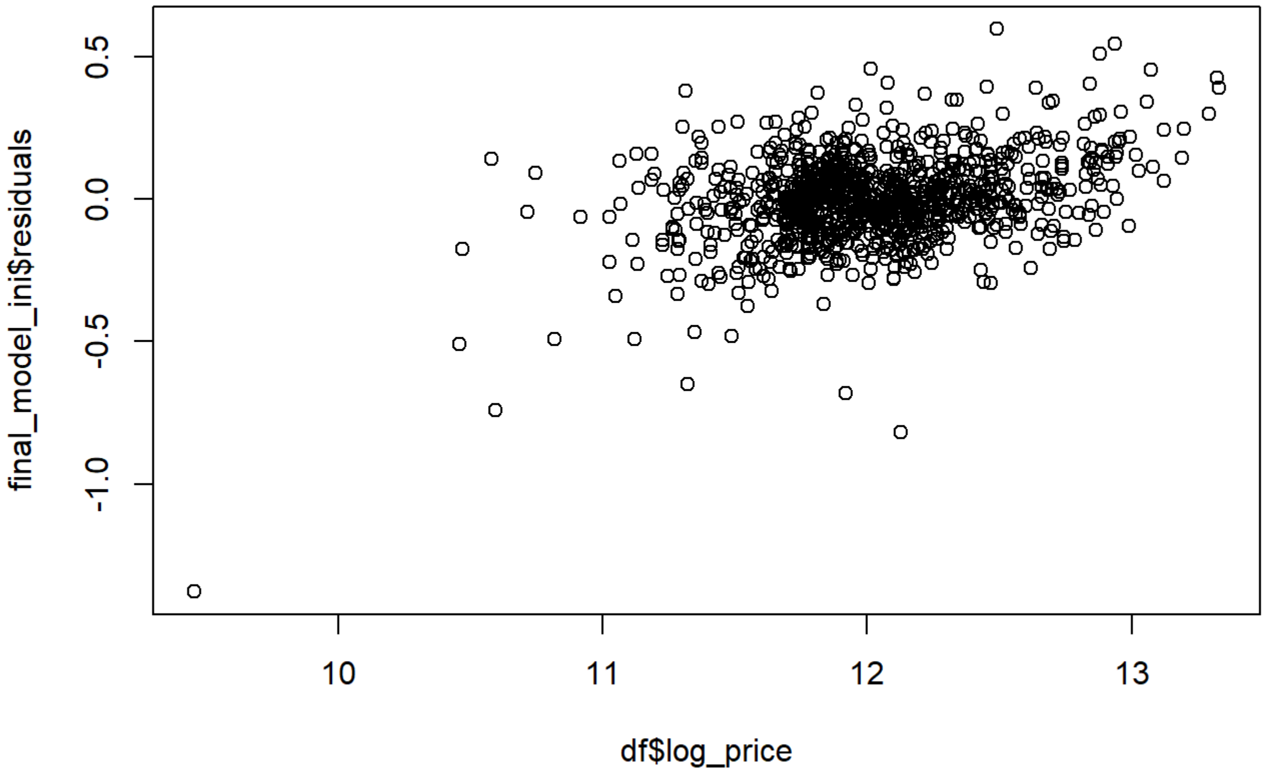
```
plot(final_model_ini$residuals~final_model_ini$fitted.values)
```



One can observe that there is mostly a random scatter around 0. However, to the extreme left, there are some outliers.

Now let us see how residuals compare with actual values:

```
plot(df$log_price,final_model_ini$residuals)
```



One can observe that the model has some negative residuals to the far left. This implies that the model tends to over predict some low-priced houses.

2.2.4 Section 2.4 Initial Model RMSE

You can calculate it directly based on the model output. Be specific about the units of your RMSE (depending on whether you transformed your response variable). The value you report will be more meaningful if it is in the original units (dollars).

To calculate rmse, first we need to convert residuals from log- units to normal $ units:

```
resid_ini<-exp(df$log_price)-exp(final_model_ini$fitted.values)
sqrt(mean(resid_ini^2))
```

```
## [1] 30770.06
```

We are getting a root mean square error value of approx 30,770$. * * *

2.2.5 Section 2.5 Overfitting

The process of building a model generally involves starting with an initial model (as you have done above), identifying its shortcomings, and adapting the model accordingly. This process may be repeated several times until the model fits the data reasonably well. However, the model may do well on training data but perform poorly out-of-sample (meaning, on a dataset other than the original training data) because the model is overly-tuned to specifically fit the training data. This is called “overfitting.” To determine whether overfitting is occurring on a model, compare the performance of a model on both in-sample and out-of-sample data sets. To look at performance of your initial model on out-of-sample data, you will use the data set `ames_test`.

We will first load ames_test and perform all those transformations and operations that we performed on ames_train:

```
load("ames_test.Rdata")
df2<-ames_test



df2<-df2%>%mutate(log_price=log(price))
df2<-df2%>%mutate(log_area=log(area))
df2<-df2%>%mutate(log_lot_area=log(Lot.Area))
df2<-df2%>%mutate(Bsmt.Qual=factor(ifelse(is.na(df2$Bsmt.Qual),'No_Bsmt',Bsmt.Qual)))
df2<-df2%>%mutate(Bsmt.Cond=factor(ifelse(is.na(df2$Bsmt.Cond),'No_Bsmt',Bsmt.Cond)))
df2<-df2%>%mutate(Bsmt.Exposure=factor(ifelse(is.na(df2$Bsmt.Exposure),'No_Bsmt',Bsmt.Exposur
e)))
df2<-df2%>%mutate(BsmtFin.Type.1=factor(ifelse(is.na(df2$BsmtFin.Type.1),'No_Bsmt',BsmtFin.Type.
1)))
df2<-df2%>%mutate(BsmtFin.Type.2=factor(ifelse(is.na(df2$BsmtFin.Type.2),'No_Bsmt',BsmtFin.Type.
2)))
df2<-df2%>%mutate(Fireplace.Qu=factor(ifelse(is.na(df2$Fireplace.Qu),'No_Fireplace',Fireplace.Q
u)))
df2<-df2%>%mutate(Garage.Qual=factor(ifelse(is.na(df2$Garage.Qual),'No_Garage',Garage.Qual)))
df2<-df2%>%mutate(Garage.Cond=factor(ifelse(is.na(df2$Garage.Cond),'No_Garage',Garage.Cond)))
df2<-df2%>%filter(!(Neighborhood=='Landmrk'))
df2<-df2%>%filter(!(is.na(BsmtFin.SF.1)),!(is.na(BsmtFin.SF.2)),!(is.na(Bsmt.Unf.SF)),!(is.na(To
tal.Bsmt.SF)),!(is.na(Bsmt.Full.Bath)),!(is.na(Bsmt.Half.Bath)))
df2<-df2%>%filter(!(is.na(Garage.Cars)))



df2$Exter.Qual<-factor(df2$Exter.Qual,ordered=T,levels=c('Po','Fa','TA','Gd','Ex'))
df2$Exter.Cond<-factor(df2$Exter.Cond,ordered=T,levels=c('Po','Fa','TA','Gd','Ex'))
df2$Heating.QC<-factor(df2$Heating.QC,ordered=T,levels=c('Po','Fa','TA','Gd','Ex'))
df2$Kitchen.Qual<-factor(df2$Kitchen.Qual,ordered=T,levels=c('Po','Fa','TA','Gd','Ex'))
df2$Functional<-factor(df2$Functional,ordered=T,levels=c('Sal','Sev','Maj2','Maj1','Mod','Min2',
'Min1','Typ'))
```

Use your model from above to generate predictions for the housing prices in the test data set. Are the predictions significantly more accurate (compared to the actual sales prices) for the training data than the test data? Why or why not? Briefly explain how you determined that (what steps or processes did you use)?

Let us test the prediction on test data:

```
predict_ini_test<-predict(final_model_ini,df2)
resid_ini_test<-exp(df2$log_price)-exp(predict_ini_test)
sqrt(mean(resid_ini_test^2))
```

```
## [1] 27601.82
```

On the test data, we are getting a rmse value of 27601$ which is less than the rmse value obtained using the same model on training data. Although there is not a significant difference between the two, we can say that our model is not suffering from the problem of over-fitting.

Note to the learner: If in real-life practice this out-of-sample analysis shows evidence that the training data fits your model a lot better than the test data, it is probably a good idea to go back and revise the model (usually by simplifying the model) to reduce this overfitting. For simplicity, we do not ask you to do this on the assignment, however.

2.3 Part 3 Development of a Final Model

Now that you have developed an initial model to use as a baseline, create a final model with *at most* 20 variables to predict housing prices in Ames, IA, selecting from the full array of variables in the dataset and using any of the tools that we introduced in this specialization.

Carefully document the process that you used to come up with your final model, so that you can answer the questions below.

2.3.1 Section 3.1 Final Model

Provide the summary table for your model.

We need to create a single variable for porch attribute because others are correlated and will supply redundant information:

```
df<-df%>%mutate(average_porch=(Open.Porch.SF+Enclosed.Porch+X3Ssn.Porch+Screen.Porch)/4)
```

We will be selecting the following 20 variables for our final model: MS.Zoning, log_area, Lot.Config, Neighborhood, Condition.1, Bldg.Type, Overall.Qual, Year.Built, Foundation, Bsmt.Qual, Total.Bsmt.SF, Heating.QC, Central.Air, X1st.Flr.SF, TotRms.AbvGrd, Kitchen.Qual, Functional, Garage.Qual, average_porch, Sale.Condition.

Let us create the model!

```
final_model_2<-lm(data=df,log_price~MS.Zoning+log_area+Lot.Config+Neighborhood+Condition.1+Bldg.
Type+Overall.Qual+Year.Built+Foundation+Bsmt.Qual+Total.Bsmt.SF+Heating.QC+Central.Air+X1st.Flr.
SF+TotRms.AbvGrd+Kitchen.Qual+Functional+Garage.Qual+average_porch+Sale.Condition)

summary(final_model_2)
```

```
##
## Call:
## lm(formula = log_price ~ MS.Zoning + log_area + Lot.Config +
##     Neighborhood + Condition.1 + Bldg.Type + Overall.Qual + Year.Built +
##     Foundation + Bsmt.Qual + Total.Bsmt.SF + Heating.QC + Central.Air +
##     X1st.Flr.SF + TotRms.AbvGrd + Kitchen.Qual + Functional +
##     Garage.Qual + average_porch + Sale.Condition, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.50061 -0.06471  0.00238  0.06633  0.53521
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.256e+00  9.373e-01   5.608 2.72e-08 ***
## MS.ZoningFV           2.468e-01  7.612e-02   3.243 0.001227 **
## MS.ZoningI (all)      1.171e-01  1.723e-01   0.679 0.497087
## MS.ZoningRH           2.749e-01  8.420e-02   3.265 0.001136 **
## MS.ZoningRL           3.056e-01  6.467e-02   4.726 2.66e-06 ***
## MS.ZoningRM           2.661e-01  5.909e-02   4.504 7.55e-06 ***
## log_area              3.891e-01  3.189e-02  12.201  < 2e-16 ***
## Lot.ConfigCulDSac     3.205e-02  2.088e-02   1.535 0.125099
## Lot.ConfigFR2        -4.087e-02  2.752e-02  -1.485 0.137909
## Lot.ConfigFR3        -9.466e-02  7.240e-02  -1.307 0.191382
## Lot.ConfigInside     -2.182e-02  1.264e-02  -1.727 0.084500 .
## NeighborhoodBlueste   2.637e-02  9.839e-02   0.268 0.788741
## NeighborhoodBrDale   -5.604e-02  7.479e-02  -0.749 0.453852
## NeighborhoodBrkSide   2.863e-02  6.159e-02   0.465 0.642182
## NeighborhoodClearCr   1.197e-01  6.358e-02   1.883 0.060085 .
## NeighborhoodCollgCr  -8.630e-03  4.872e-02  -0.177 0.859447
## NeighborhoodCrawfor   1.559e-01  5.606e-02   2.780 0.005540 **
## NeighborhoodEdwards  -4.761e-02  5.192e-02  -0.917 0.359400
## NeighborhoodGilbert  -1.576e-02  5.118e-02  -0.308 0.758169
## NeighborhoodGreens    1.588e-01  8.574e-02   1.852 0.064378 .
## NeighborhoodGrnHill   5.580e-01  1.133e-01   4.927 9.93e-07 ***
## NeighborhoodIDOTRR   -4.760e-02  6.901e-02  -0.690 0.490484
## NeighborhoodMeadowV  -1.150e-01  6.557e-02  -1.753 0.079862 .
## NeighborhoodMitchel   3.430e-02  5.201e-02   0.659 0.509777
## NeighborhoodNAmes     1.140e-02  5.122e-02   0.223 0.823947
## NeighborhoodNoRidge   9.494e-02  5.395e-02   1.760 0.078802 .
## NeighborhoodNPkVill   1.204e-02  8.758e-02   0.138 0.890649
## NeighborhoodNridgHt   1.207e-01  4.874e-02   2.475 0.013493 *
## NeighborhoodNWAmes   -1.910e-03  5.331e-02  -0.036 0.971421
## NeighborhoodOldTown  -3.950e-02  6.303e-02  -0.627 0.531042
## NeighborhoodSawyer    3.348e-02  5.308e-02   0.631 0.528299
## NeighborhoodSawyerW  -3.561e-02  5.095e-02  -0.699 0.484781
## NeighborhoodSomerst   9.744e-02  5.740e-02   1.698 0.089927 .
## NeighborhoodStoneBr   1.502e-01  5.444e-02   2.759 0.005911 **
## NeighborhoodSWISU    -4.730e-02  6.733e-02  -0.702 0.482587
## NeighborhoodTimber    3.750e-02  5.673e-02   0.661 0.508769
## NeighborhoodVeenker   9.400e-02  6.446e-02   1.458 0.145100
## Condition.1Feedr      1.757e-02  3.923e-02   0.448 0.654355
## Condition.1Norm        8.063e-02  3.255e-02   2.477 0.013436 *  
## Condition.1PosA        1.713e-01  6.104e-02   2.806 0.005122 ** 
## Condition.1PosN       -4.914e-04  5.455e-02  -0.009 0.992815    
## Condition.1RRAe        1.047e-02  5.834e-02   0.179 0.857596    
## Condition.1RRAn        4.643e-02  5.181e-02   0.896 0.370415    
## Condition.1RRNe       -4.184e-03  1.052e-01  -0.040 0.968281    
## Condition.1RRNn       -4.306e-02  8.803e-02  -0.489 0.624824    
## Bldg.Type2fmCon        7.648e-02  3.540e-02   2.161 0.030989 *  
## Bldg.TypeDuplex       -8.109e-02  3.007e-02  -2.696 0.007138 ** 
## Bldg.TypeTwnhs        -1.371e-01  3.468e-02  -3.954 8.28e-05 ***
## Bldg.TypeTwnhsE       -8.326e-02  2.339e-02  -3.559 0.000392 ***
## Overall.Qual           6.103e-02  6.506e-03   9.380  < 2e-16 ***
## Year.Built             1.473e-03  4.473e-04   3.293 0.001031 ** 
## FoundationCBlock       2.541e-02  2.142e-02   1.186 0.235783    
## FoundationPConc        2.568e-02  2.389e-02   1.075 0.282771    
## FoundationSlab         1.246e-01  7.308e-02   1.706 0.088427 .  
## FoundationStone        1.727e-01  8.411e-02   2.053 0.040379 *  
## Bsmt.Qual3            -1.754e-01  4.164e-02  -4.212 2.78e-05 ***
## Bsmt.Qual4            -8.213e-02  2.204e-02  -3.727 0.000206 ***
## Bsmt.Qual5             4.159e-01  1.842e-01   2.258 0.024181 *  
## Bsmt.Qual6            -1.209e-01  2.805e-02  -4.309 1.82e-05 ***
## Bsmt.QualNo_Bsmt      -2.190e-01  6.910e-02  -3.168 0.001584 ** 
## Total.Bsmt.SF          8.644e-05  2.788e-05   3.101 0.001991 ** 
## Heating.QC.L          -1.497e-02  9.851e-02  -0.152 0.879272    
## Heating.QC.Q           8.172e-02  8.245e-02   0.991 0.321910    
## Heating.QC.C          -6.293e-02  5.303e-02  -1.187 0.235688    
## Heating.QC^4           1.931e-02  2.564e-02   0.753 0.451547    
## Central.AirY           1.715e-01  2.728e-02   6.284 5.10e-10 ***
## X1st.Flr.SF            5.795e-05  3.005e-05   1.929 0.054088 .  
## TotRms.AbvGrd          3.716e-05  5.691e-03   0.007 0.994792    
## Kitchen.Qual.L         1.213e-01  1.002e-01   1.210 0.226627    
## Kitchen.Qual.Q         2.080e-02  8.402e-02   0.248 0.804530    
## Kitchen.Qual.C         1.254e-02  5.370e-02   0.234 0.815339    
## Kitchen.Qual^4        -1.422e-02  2.538e-02  -0.560 0.575451    
## Functional.L           4.446e-01  9.904e-02   4.489 8.09e-06 ***
## Functional.Q          -2.935e-01  8.947e-02  -3.280 0.001076 ** 
## Functional.C           1.831e-01  8.721e-02   2.100 0.036019 *  
## Functional^4          -6.488e-02  7.981e-02  -0.813 0.416438    
## Functional^5           4.778e-02  7.394e-02   0.646 0.518348    
## Functional^6           2.785e-03  5.700e-02   0.049 0.961040    
## Garage.Qual3          -1.911e-01  1.453e-01  -1.315 0.188933    
## Garage.Qual4          -9.307e-02  1.537e-01  -0.605 0.545028    
## Garage.Qual5          -3.800e-01  1.778e-01  -2.137 0.032840 *  
## Garage.Qual6          -1.733e-01  1.435e-01  -1.208 0.227352    
## Garage.QualNo_Garage  -2.066e-01  1.445e-01  -1.430 0.153183    
## average_porch          3.088e-04  1.959e-04   1.576 0.115309    
## Sale.ConditionAdjLand  1.583e-01  1.114e-01   1.421 0.155786    
## Sale.ConditionAlloca   2.443e-01  8.729e-02   2.799 0.005230 ** 
## Sale.ConditionFamily  -8.302e-02  3.984e-02  -2.084 0.037444 *  
## Sale.ConditionNormal   7.547e-02  1.988e-02   3.796 0.000157 ***
## Sale.ConditionPartial  9.331e-02  2.727e-02   3.421 0.000651 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1387 on 909 degrees of freedom
## Multiple R-squared:  0.9007, Adjusted R-squared:  0.8911
## F-statistic:  93.7 on 88 and 909 DF,  p-value: < 2.2e-16
```

From the summary output of this model, we can observe that our model has an overall F-statistic of 93.7 which leads to a Adjusted R-squared value of 0.8911. This means that the model is doing well as a whole. However, some predictor variables seem insignificant as indicated by their high p-values. Let us see if we can optimize further:

```
final_model_2_aic<-step(final_model_2,k=2)
```

```
## Start:  AIC=-3857.88
## log_price ~ MS.Zoning + log_area + Lot.Config + Neighborhood + 
##     Condition.1 + Bldg.Type + Overall.Qual + Year.Built + Foundation + 
##     Bsmt.Qual + Total.Bsmt.SF + Heating.QC + Central.Air + X1st.Flr.SF + 
##     TotRms.AbvGrd + Kitchen.Qual + Functional + Garage.Qual + 
##     average_porch + Sale.Condition
## 
##                  Df Sum of Sq    RSS     AIC
## - TotRms.AbvGrd   1   0.00000 17.493 -3859.9
## - Garage.Qual     5   0.17214 17.665 -3858.1
## - Foundation      4   0.13967 17.632 -3857.9
## <none>                        17.493 -3857.9
## - average_porch   1   0.04781 17.541 -3857.2
## - X1st.Flr.SF     1   0.07158 17.564 -3855.8
## - Lot.Config      4   0.25078 17.744 -3851.7
## - Heating.QC      4   0.27733 17.770 -3850.2
## - Total.Bsmt.SF   1   0.18500 17.678 -3849.4
## - Year.Built      1   0.20863 17.701 -3848.0
## - Condition.1     8   0.48569 17.979 -3846.5
## - Kitchen.Qual    4   0.43116 17.924 -3841.6
## - Functional      6   0.50622 17.999 -3841.4
## - MS.Zoning       5   0.47615 17.969 -3841.1
## - Bsmt.Qual       5   0.63610 18.129 -3832.2
## - Bldg.Type       4   0.67273 18.166 -3828.2
## - Sale.Condition  5   0.76215 18.255 -3825.3
## - Central.Air     1   0.76002 18.253 -3817.4
## - Neighborhood   26   2.54616 20.039 -3774.3
## - Overall.Qual    1   1.69299 19.186 -3767.7
## - log_area        1   2.86458 20.357 -3708.5
## 
## Step:  AIC=-3859.88
## log_price ~ MS.Zoning + log_area + Lot.Config + Neighborhood + 
##     Condition.1 + Bldg.Type + Overall.Qual + Year.Built + Foundation + 
##     Bsmt.Qual + Total.Bsmt.SF + Heating.QC + Central.Air + X1st.Flr.SF + 
##     Kitchen.Qual + Functional + Garage.Qual + average_porch + 
##     Sale.Condition
## 
##                  Df Sum of Sq    RSS     AIC
## - Garage.Qual     5    0.1723 17.665 -3860.1
## - Foundation      4    0.1397 17.632 -3859.9
## <none>                        17.493 -3859.9
## - average_porch   1    0.0480 17.541 -3859.1
## - X1st.Flr.SF     1    0.0716 17.564 -3857.8
## - Lot.Config      4    0.2508 17.744 -3853.7
## - Heating.QC      4    0.2773 17.770 -3852.2
## - Total.Bsmt.SF   1    0.1859 17.679 -3851.3
## - Year.Built      1    0.2087 17.701 -3850.0
## - Condition.1     8    0.4859 17.979 -3848.5
## - Kitchen.Qual    4    0.4327 17.925 -3843.5
## - Functional      6    0.5066 17.999 -3843.4
## - MS.Zoning       5    0.4766 17.969 -3843.1
## - Bsmt.Qual       5    0.6403 18.133 -3834.0
## - Bldg.Type       4    0.6794 18.172 -3829.9
## - Sale.Condition  5    0.7622 18.255 -3827.3
## - Central.Air     1    0.7616 18.254 -3819.3
## - Neighborhood   26    2.5493 20.042 -3776.1
## - Overall.Qual    1    1.6939 19.187 -3769.6
## - log_area        1    5.6474 23.140 -3582.7
## 
## Step:  AIC=-3860.1
## log_price ~ MS.Zoning + log_area + Lot.Config + Neighborhood + 
##     Condition.1 + Bldg.Type + Overall.Qual + Year.Built + Foundation + 
##     Bsmt.Qual + Total.Bsmt.SF + Heating.QC + Central.Air + X1st.Flr.SF + 
##     Kitchen.Qual + Functional + average_porch + Sale.Condition
## 
##                   Df Sum of Sq    RSS     AIC
## <none>                         17.665 -3860.1
## - Foundation      4    0.1524 17.817 -3859.5
## - average_porch   1    0.0513 17.716 -3859.2
## - X1st.Flr.SF     1    0.0724 17.737 -3858.0
## - Lot.Config      4    0.2801 17.945 -3852.4
## - Total.Bsmt.SF   1    0.1953 17.860 -3851.1
## - Heating.QC      4    0.3037 17.969 -3851.1
## - Condition.1     8    0.4799 18.145 -3849.3
## - Year.Built      1    0.2558 17.921 -3847.8
## - Kitchen.Qual    4    0.4201 18.085 -3844.6
## - Functional      6    0.5031 18.168 -3844.1
## - MS.Zoning       5    0.4795 18.145 -3843.4
## - Bsmt.Qual       5    0.5682 18.233 -3838.5
## - Bldg.Type       4    0.7117 18.377 -3828.7
## - Sale.Condition  5    0.7595 18.424 -3828.1
## - Central.Air     1    1.1051 18.770 -3801.5
## - Neighborhood   26    2.5601 20.225 -3777.0
## - Overall.Qual    1    1.7212 19.386 -3769.3
## - log_area        1    5.8098 23.475 -3578.3
```

```
summary(final_model_2_aic)
```

```
##
## Call:
## lm(formula = log_price ~ MS.Zoning + log_area + Lot.Config +
##     Neighborhood + Condition.1 + Bldg.Type + Overall.Qual + Year.Built +
##     Foundation + Bsmt.Qual + Total.Bsmt.SF + Heating.QC + Central.Air +
##     X1st.Flr.SF + Kitchen.Qual + Functional + average_porch +
##     Sale.Condition, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.49090 -0.06434  0.00248  0.06523  0.54234
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            4.761e+00  9.079e-01   5.245 1.95e-07 ***
## MS.ZoningFV            2.462e-01  7.534e-02   3.268 0.001125 **
## MS.ZoningI (all)       1.103e-01  1.723e-01   0.640 0.522295
## MS.ZoningRH            2.730e-01  8.369e-02   3.262 0.001148 **
## MS.ZoningRL            3.025e-01  6.377e-02   4.743 2.44e-06 ***
## MS.ZoningRM            2.632e-01  5.825e-02   4.519 7.03e-06 ***
## log_area               3.924e-01  2.262e-02  17.347  < 2e-16 ***
## Lot.ConfigCulDSac      3.179e-02  2.086e-02   1.524 0.127918
## Lot.ConfigFR2         -4.507e-02  2.744e-02  -1.642 0.100861
## Lot.ConfigFR3         -9.603e-02  7.250e-02  -1.325 0.185636
## Lot.ConfigInside      -2.419e-02  1.258e-02  -1.923 0.054809 .
## NeighborhoodBlueste    2.735e-02  9.853e-02   0.278 0.781414
## NeighborhoodBrDale    -5.311e-02  7.489e-02  -0.709 0.478400
## NeighborhoodBrkSide    3.878e-02  6.134e-02   0.632 0.527393
## NeighborhoodClearCr    1.196e-01  6.354e-02   1.882 0.060178 .
## NeighborhoodCollgCr   -5.793e-03  4.859e-02  -0.119 0.905128
## NeighborhoodCrawfor    1.536e-01  5.588e-02   2.749 0.006096 **
## NeighborhoodEdwards   -5.158e-02  5.158e-02  -1.000 0.317632
## NeighborhoodGilbert   -1.325e-02  5.110e-02  -0.259 0.795446
## NeighborhoodGreens     1.566e-01  8.557e-02   1.830 0.067597 .
## NeighborhoodGrnHill    5.508e-01  1.132e-01   4.867 1.33e-06 ***
## NeighborhoodIDOTRR    -3.783e-02  6.831e-02  -0.554 0.579877
## NeighborhoodMeadowV   -1.232e-01  6.496e-02  -1.897 0.058118 .
## NeighborhoodMitchel    3.471e-02  5.190e-02   0.669 0.503744
## NeighborhoodNAmes      1.490e-02  5.105e-02   0.292 0.770516
## NeighborhoodNoRidge    9.859e-02  5.385e-02   1.831 0.067439 .
## NeighborhoodNPkVill    1.387e-02  8.764e-02   0.158 0.874321
## NeighborhoodNridgHt    1.197e-01  4.880e-02   2.452 0.014376 *
## NeighborhoodNWAmes     3.809e-04  5.333e-02   0.007 0.994304
## NeighborhoodOldTown   -3.225e-02  6.282e-02  -0.513 0.607872
## NeighborhoodSawyer     3.567e-02  5.298e-02   0.673 0.500975
## NeighborhoodSawyerW   -3.385e-02  5.086e-02  -0.665 0.505908
## NeighborhoodSomerst    9.604e-02  5.740e-02   1.673 0.094635 .
## NeighborhoodStoneBr    1.523e-01  5.450e-02   2.794 0.005312 **
## NeighborhoodSWISU     -3.902e-02  6.703e-02  -0.582 0.560657
## NeighborhoodTimber     4.140e-02  5.661e-02   0.731 0.464729
## NeighborhoodVeenker    9.325e-02  6.419e-02   1.453 0.146611
## Condition.1Feedr       2.336e-02  3.886e-02   0.601 0.547894
## Condition.1Norm        8.421e-02  3.219e-02   2.616 0.009038 **
## Condition.1PosA        1.739e-01  6.090e-02   2.856 0.004389 **
## Condition.1PosN        9.204e-03  5.429e-02   0.170 0.865428
## Condition.1RRAe        1.412e-02  5.803e-02   0.243 0.807776
## Condition.1RRAn        4.982e-02  5.149e-02   0.968 0.333513
## Condition.1RRNe        2.399e-04  1.050e-01   0.002 0.998177
## Condition.1RRNn       -4.069e-02  8.798e-02  -0.462 0.643853
## Bldg.Type2fmCon        8.358e-02  3.494e-02   2.392 0.016952 *
## Bldg.TypeDuplex       -8.550e-02  2.897e-02  -2.951 0.003247 **
## Bldg.TypeTwnhs        -1.367e-01  3.421e-02  -3.997 6.92e-05 ***
## Bldg.TypeTwnhsE       -8.087e-02  2.271e-02  -3.560 0.000389 ***
## Overall.Qual           6.132e-02  6.494e-03   9.442  < 2e-16 ***
## Year.Built             1.613e-03  4.430e-04   3.640 0.000288 ***
## FoundationCBlock       2.791e-02  2.108e-02   1.324 0.185836
## FoundationPConc        2.297e-02  2.362e-02   0.972 0.331148
## FoundationSlab         1.217e-01  7.309e-02   1.665 0.096161 .
## FoundationStone        1.840e-01  8.389e-02   2.193 0.028550 *
## Bsmt.Qual3            -1.742e-01  4.155e-02  -4.193 3.02e-05 ***
## Bsmt.Qual4            -8.260e-02  2.206e-02  -3.744 0.000193 ***
## Bsmt.Qual5             2.317e-01  1.544e-01   1.501 0.133647
## Bsmt.Qual6            -1.206e-01  2.797e-02  -4.312 1.80e-05 ***
## Bsmt.QualNo_Bsmt      -2.063e-01  6.892e-02  -2.994 0.002832 **
## Total.Bsmt.SF          8.843e-05  2.781e-05   3.180 0.001521 **
## Heating.QC.L          -2.463e-02  9.798e-02  -0.251 0.801568
## Heating.QC.Q           9.340e-02  8.212e-02   1.137 0.255686
## Heating.QC.C          -7.329e-02  5.269e-02  -1.391 0.164603
## Heating.QC^4           2.351e-02  2.552e-02   0.921 0.357082
## Central.AirY           1.926e-01  2.546e-02   7.566 9.37e-14 ***
## X1st.Flr.SF            5.826e-05  3.008e-05   1.936 0.053117 .
## Kitchen.Qual.L         1.104e-01  9.963e-02   1.108 0.267949
## Kitchen.Qual.Q         2.832e-02  8.349e-02   0.339 0.734502
## Kitchen.Qual.C         7.318e-03  5.341e-02   0.137 0.891055
## Kitchen.Qual^4        -1.209e-02  2.519e-02  -0.480 0.631393
## Functional.L           4.435e-01  9.888e-02   4.485 8.22e-06 ***
## Functional.Q          -2.928e-01  8.941e-02  -3.275 0.001095 **
## Functional.C           1.780e-01  8.713e-02   2.043 0.041361 *
## Functional^4          -5.620e-02  7.973e-02  -0.705 0.481071
## Functional^5           3.930e-02  7.393e-02   0.532 0.595161
## Functional^6           9.923e-03  5.693e-02   0.174 0.861677
## average_porch          3.186e-04  1.954e-04   1.631 0.103292
## Sale.ConditionAdjLand  1.446e-01  1.106e-01   1.308 0.191335
## Sale.ConditionAlloca   2.364e-01  8.731e-02   2.707 0.006906 **
## Sale.ConditionFamily  -8.426e-02  3.977e-02  -2.119 0.034381 *
## Sale.ConditionNormal   7.550e-02  1.981e-02   3.811 0.000148 ***
## Sale.ConditionPartial  9.330e-02  2.725e-02   3.424 0.000644 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1389 on 915 degrees of freedom
## Multiple R-squared:  0.8997, Adjusted R-squared:  0.8907
## F-statistic: 100.1 on 82 and 915 DF,  p-value: < 2.2e-16
```

Using AIC, we are getting adjusted R-squared of 0.8907 by eliminating 2 variables.

Using BIC, we are getting adjusted R-squared of 0.8648 by eliminating 11 variables!

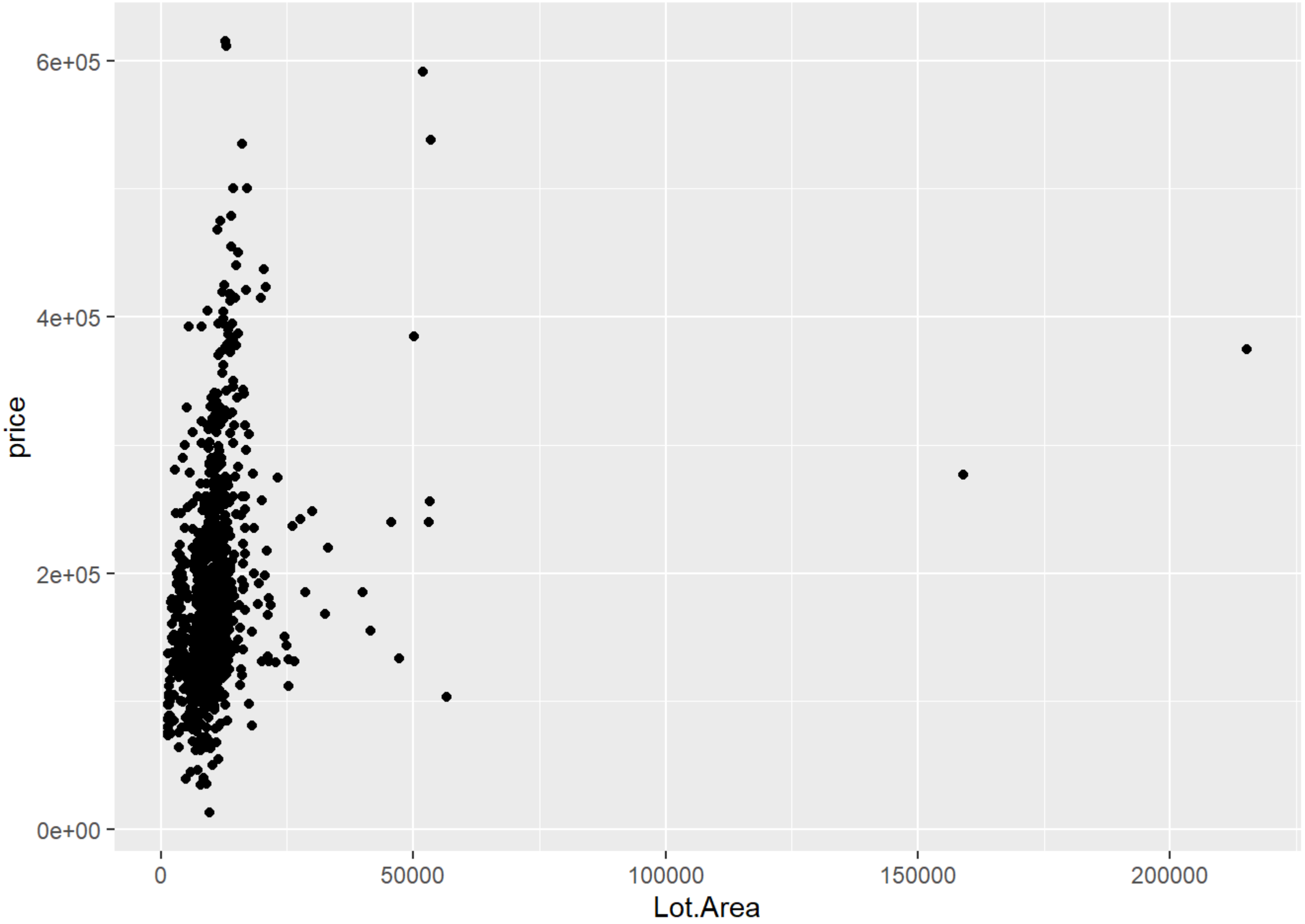
Using backwards elimination by p-value method, we are getting adjusted R-squared of 0.8857 by eliminating 5-variables.

Moreover, since a lot of important variables are getting eliminated by these approaches and not leading to increase in adjusted R-squared, we will stick with the original model.

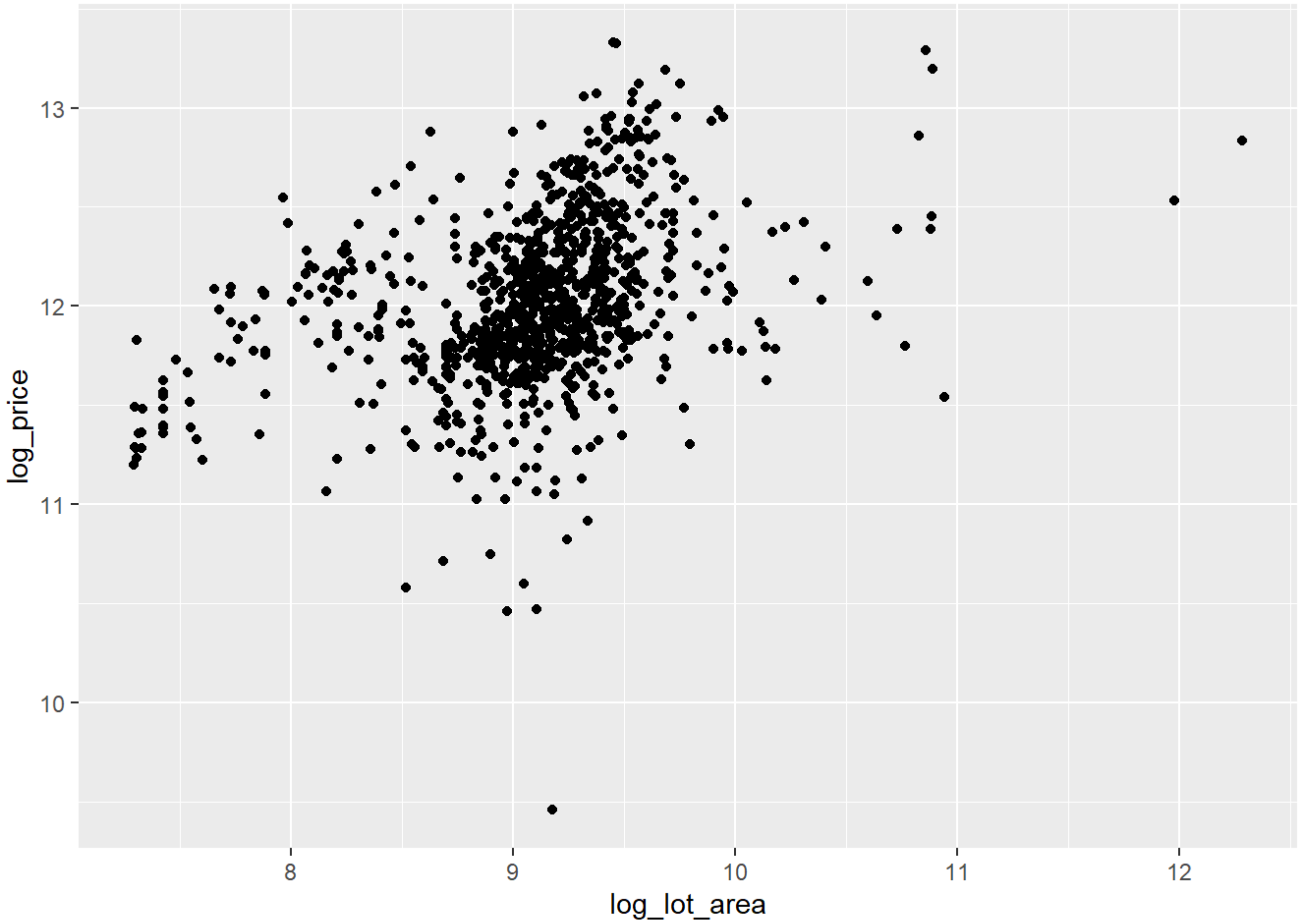
2.3.2 Section 3.2 Transformation

log transformation of area, price, lot.area was done. This is because the relationship between logarithms becomes linear while that between the original variables wasn’t linear. This is as shown in the plots below:

```
df%>%ggplot(aes(x=Lot.Area,y=price))+geom_point()
```



```
df%>%ggplot(aes(x=log_lot_area,y=log_price))+geom_point()
```



We can see that the strength of the linear relationship is considerably improved by the log transformation.

A similar argument holds true for transforming area too.

Some variables like Bsmt.Qual, Heating.QC to name a few were initially nominal categorical variables. However, the levels of the factors were ordinal in nature, hence a transformation from nominal to ordinal factors was done. This is shown below:

```
str(df$Heating.QC)
```

```
##  Ord.factor w/ 5 levels "Po"<"Fa"<"TA"<..: 3 3 5 4 5 5 3 5 3 5 ...
```

```
levels(df$Heating.QC)
```

```
## [1] "Po" "Fa" "TA" "Gd" "Ex"
```

Further, a new variable average_porch was created to account for the porch related variables. This was done due to high collinearity between the porch variables.

NA values of some variables like Bsmt.Qual were recoded as a separate category. * * *

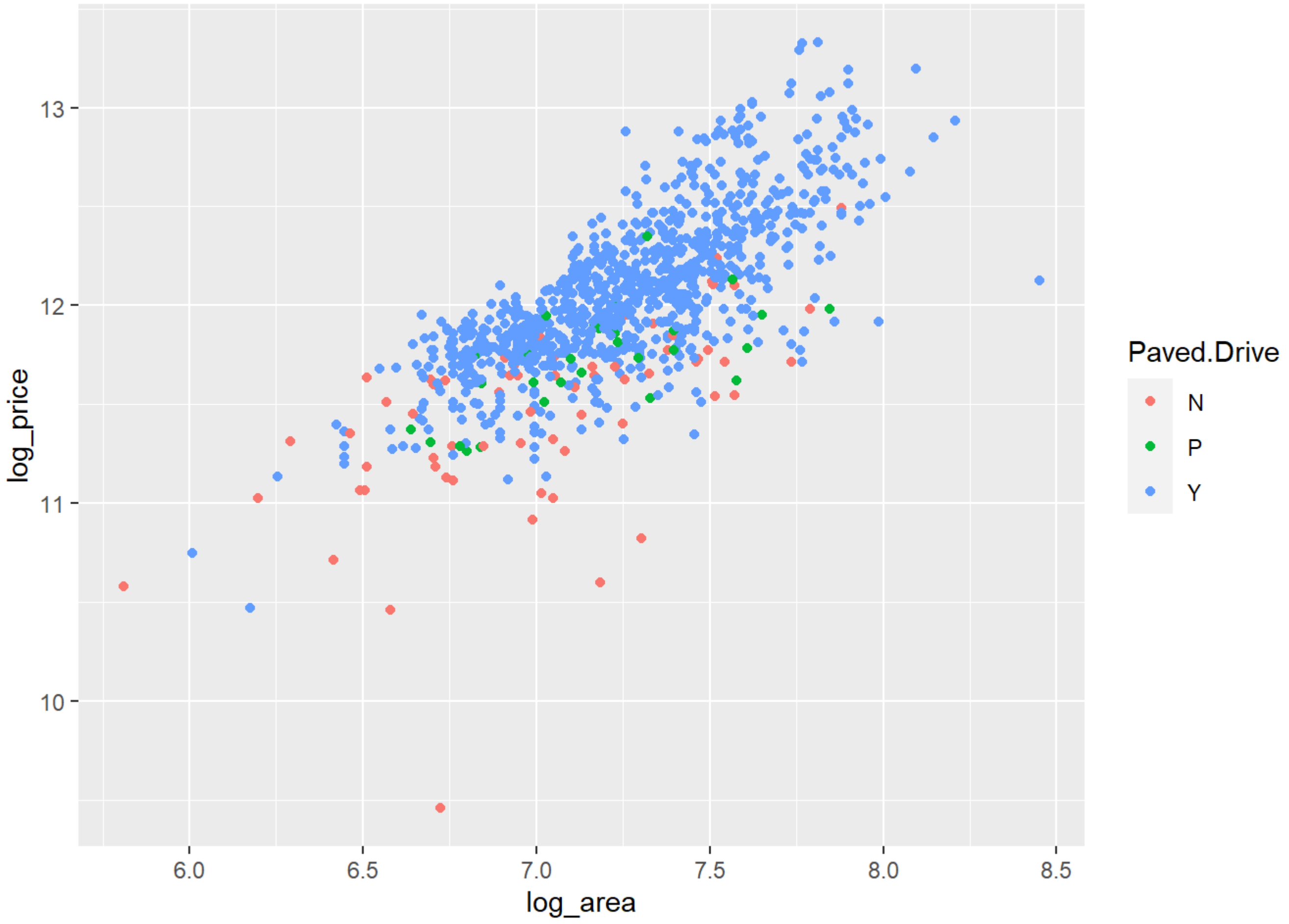
2.3.3 Section 3.3 Variable Interaction

Did you decide to include any variable interactions? Why or why not? Explain in a few sentences.

Some Interaction terms are included in our analysis since if they are significant and we decide to drop them, the model becomes less robust. One could draw improper conclusions without including these variables, so they are included. Consider the example below:

Consider the following plot: The slopes of log_price vs log_area differ slightly across the categories of the third variable. A few interaction variables (not this one) are thus included.

```
df%>%ggplot(aes(x=log_area,y=log_price,color=Paved.Drive))+geom_point()
```



2.3.4 Section 3.4 Variable Selection

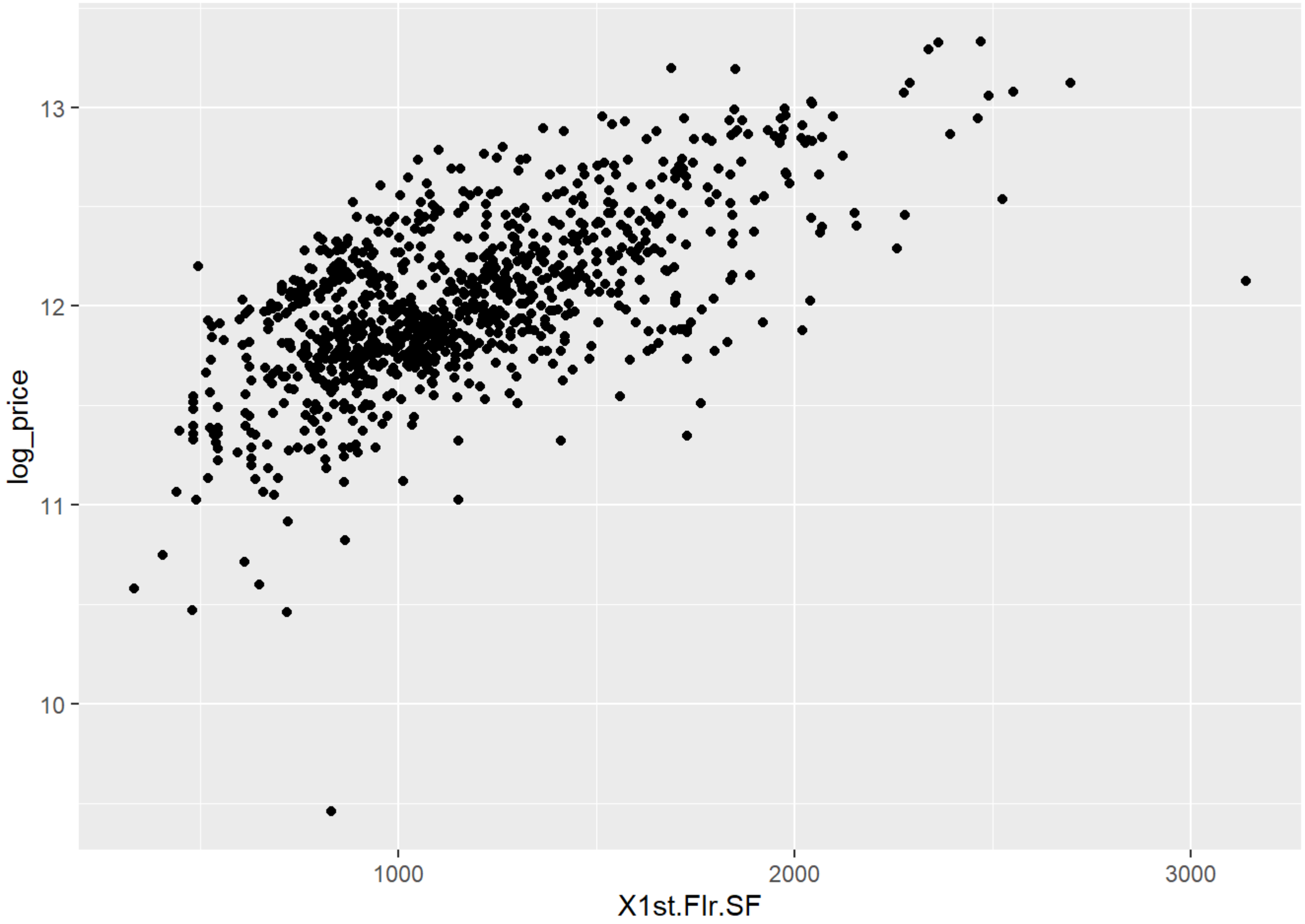
What method did you use to select the variables you included? Why did you select the method you used? Explain in a few sentences.

The following variables were found to have strong relationship with response variable from the EDA section: Central.Air, Foundation, Overall.Qual, log_area, Neighborhood, Garage.Qual, Bldg.Type.

Further, from the previous analysis, it was found that Lot.Shape, Fireplace.Qu were insignificant predictors of the response variables. We were asked to shortlist 20 variables from the set of 81 predictor variables. The house has different attributes like Garage, Basement, Bedrooms, Area etc, so at least one variable from each attribute must be included in those 20 variables to make the model robust.

Examples of variable selection reasons: X1st.Flr.SF was found to have a strong linear relationship with log_price indicated by the scatter plot and correlation coefficient:

```
df%>%ggplot(aes(x=X1st.Flr.SF,y=log_price))+geom_point()
```

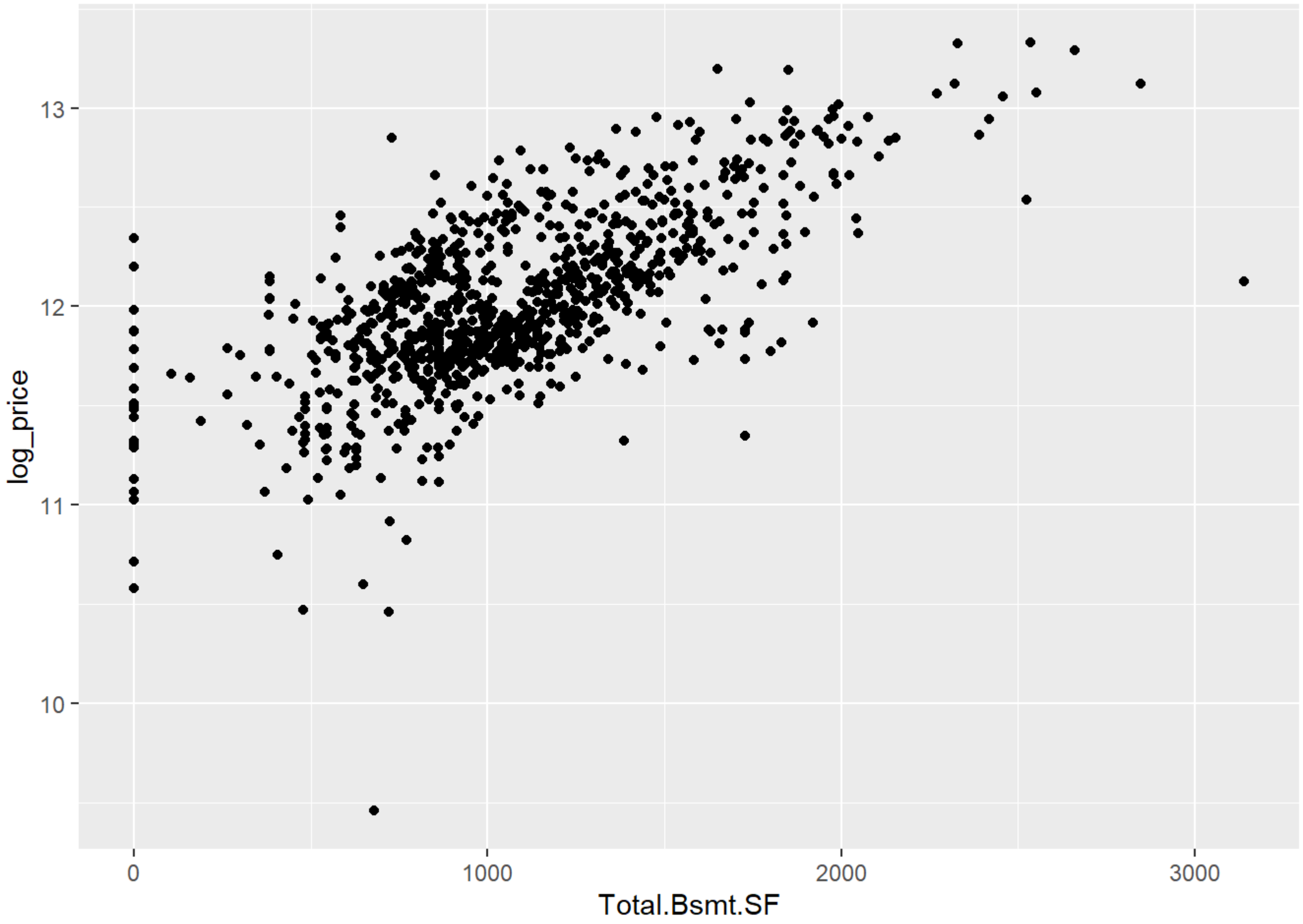


```
cor(df$X1st.Flr.SF,df$log_price)
```

```
## [1] 0.6523931
```

A similar case was observed for Total.Bsmt.SF and log_price:

```
df%>%ggplot(aes(x=Total.Bsmt.SF,y=log_price))+geom_point()
```



```
cor(df$Total.Bsmt.SF,df$log_price)
```

```
## [1] 0.6678017
```

This was how other numerical variables were selected.

Anova test was used to determine the influence of categorical variables on the response variable.

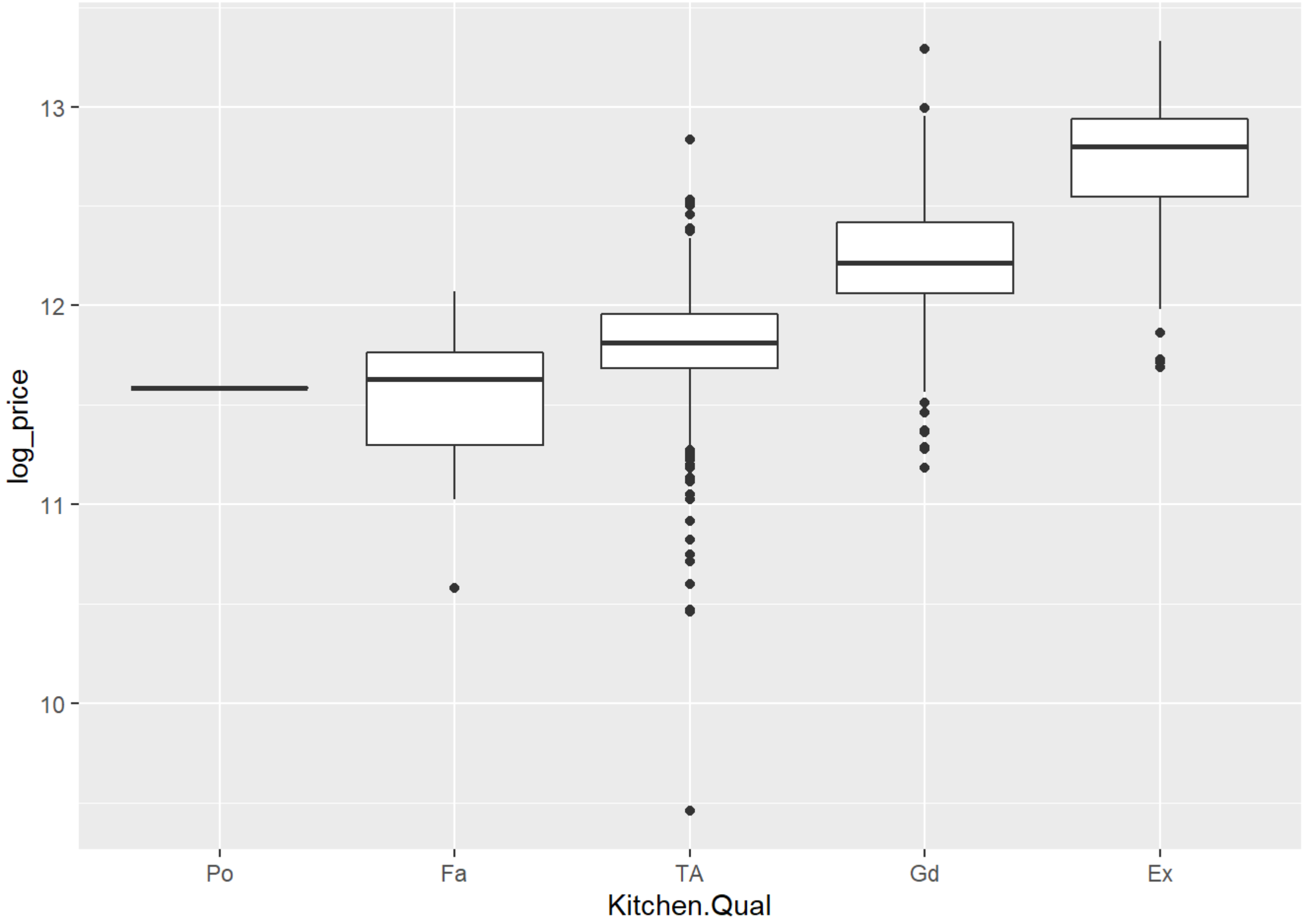
For example, consider the relationship between Kitchen.Qual and log_price:

```
df%>%group_by(Kitchen.Qual)%>%summarise(mean=mean(log_price),median=median(log_price),sd=sd(log_
price),number=n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 5 x 5
##   Kitchen.Qual  mean median     sd number
##   <ord>        <dbl>  <dbl>  <dbl>  <int>
## 1 Po            11.6   11.6 NA          1
## 2 Fa            11.5   11.6  0.362     20
## 3 TA            11.8   11.8  0.304    508
## 4 Gd            12.2   12.2  0.300    402
## 5 Ex            12.7   12.8  0.367     67
```

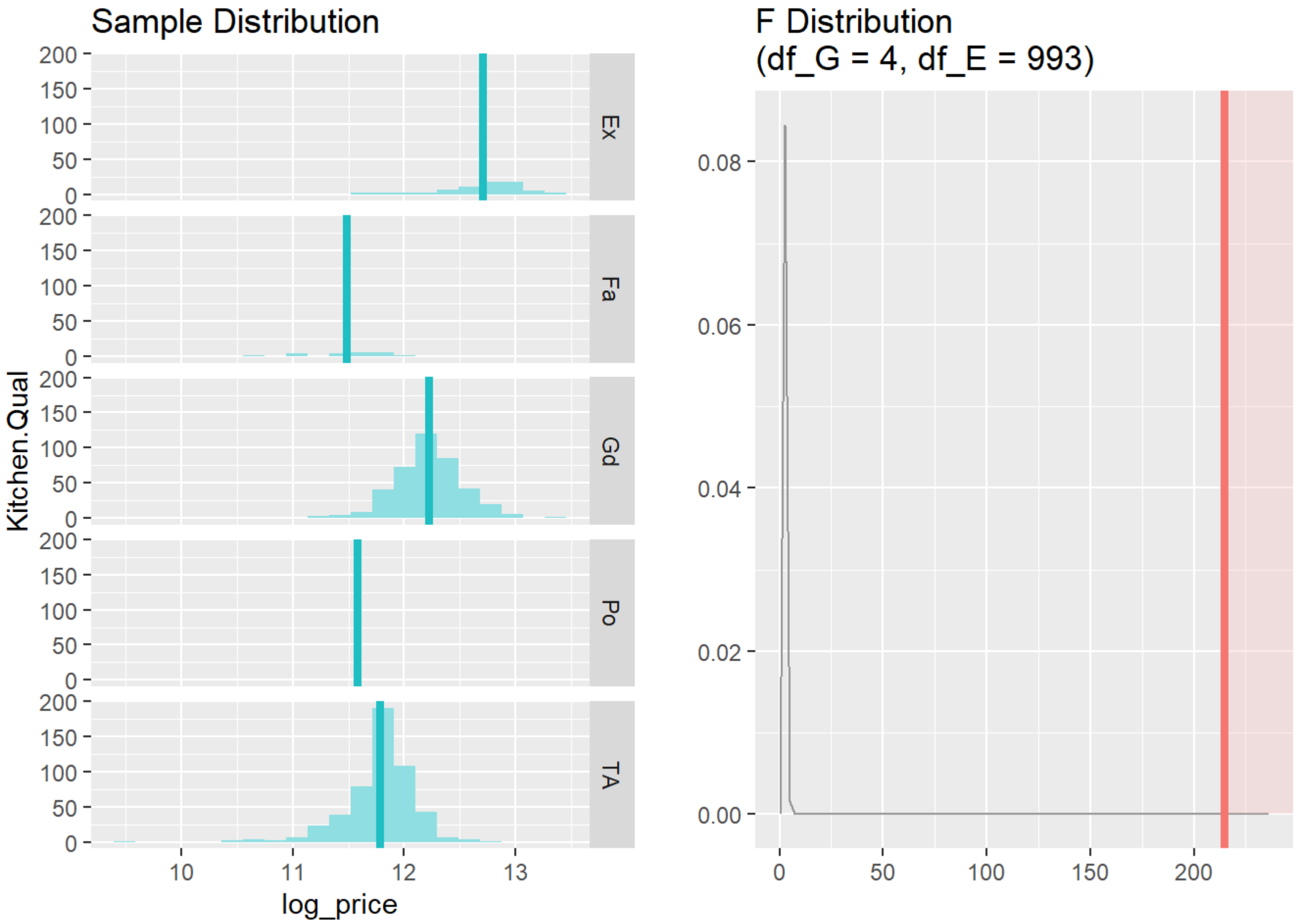
```
df%>%ggplot(aes(x=Kitchen.Qual,y=log_price))+geom_boxplot()
```



Both the summary statistics and box-plots indicate an association between the two. An Anova test confirms this as shown:

```
inference(data=df,x=Kitchen.Qual,y=log_price,type='ht',method='theoretical',statistic='mean',alt
ernative='greater')
```

```
## Response variable: numerical
## Explanatory variable: categorical (5 levels)
## n_Po = 1, y_bar_Po = 11.5852, s_Po = NA
## n_Fa = 20, y_bar_Fa = 11.4893, s_Fa = 0.3624
## n_TA = 508, y_bar_TA = 11.7874, s_TA = 0.3042
## n_Gd = 402, y_bar_Gd = 12.2255, s_Gd = 0.3005
## n_Ex = 67, y_bar_Ex = 12.7047, s_Ex = 0.3665
##
## ANOVA:
##                df   Sum_Sq Mean_Sq        F  p_value
## Kitchen.Qual    4   81.683 20.4207 214.6127 < 0.0001
## Residuals     993  94.4855  0.0952
## Total         997 176.1685
##
## Pairwise tests - t tests with pooled SD:
## # A tibble: 0 x 3
## # ... with 3 variables: group1 <chr>, group2 <chr>, p.value <dbl>
```



Reasons for excluding some variables: Those who failed these tests, were not linearly associated with the response variable and thus would not satisfy the assumptions of regression. Further, since variable selection requires actual experience, important variables were not dropped.

2.3.5 Section 3.5 Model Testing

How did testing the model on out-of-sample data affect whether or how you changed your model? Explain in a few sentences.

Our model is seen to perform better on out-of-sample data, thus, no change has been made after testing on out-of-sample data.

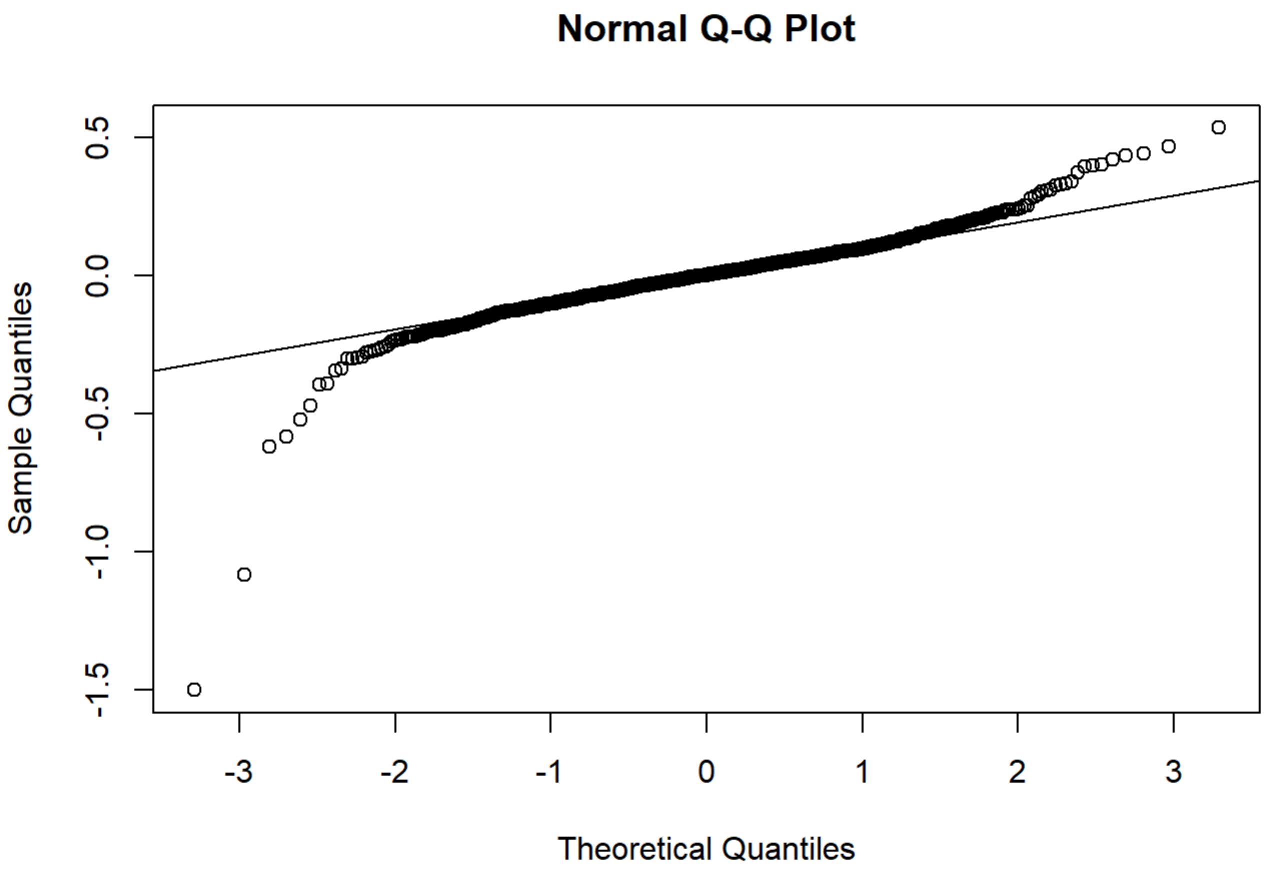
2.4 Part 4 Final Model Assessment

2.4.1 Section 4.1 Final Model Residual

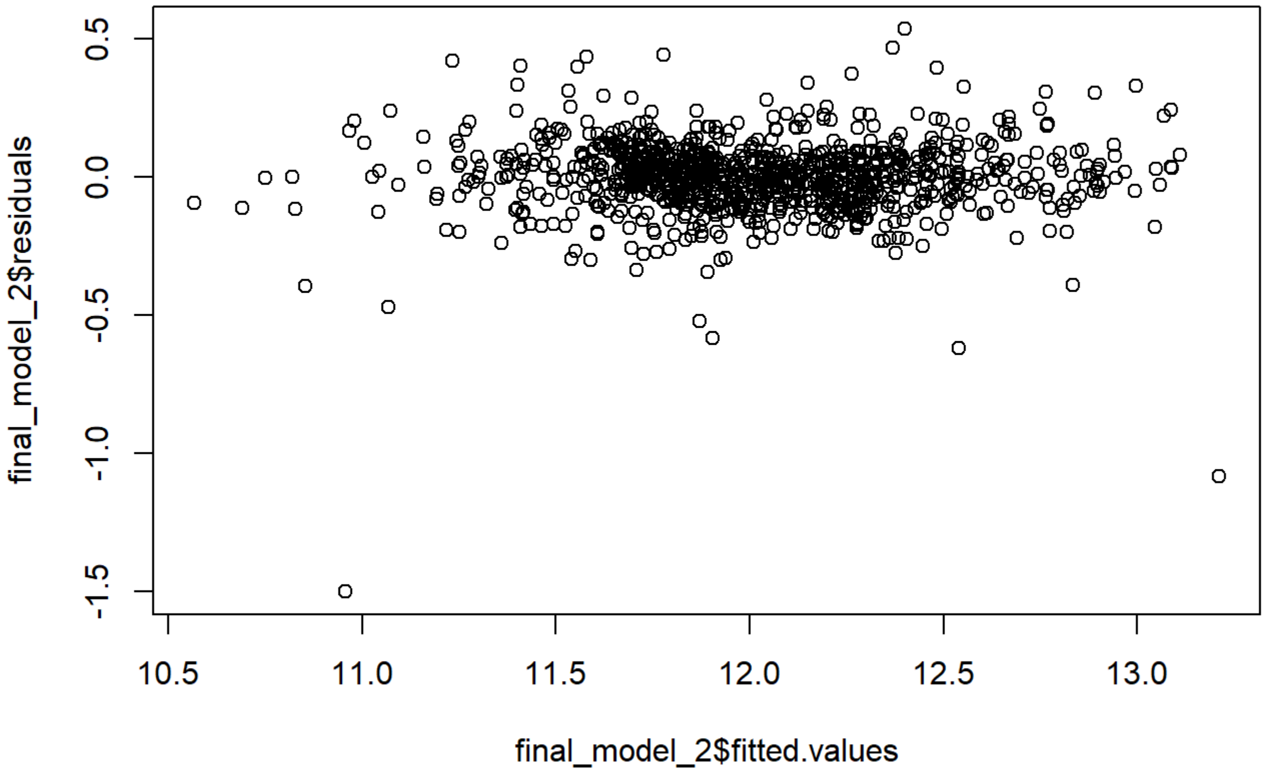
For your final model, create and briefly interpret an informative plot of the residuals.

The following qq-plot shows the distribution of residuals of our final model. It is safe to say that they are normally distributed with a few outliers towards the endpoints.

```
qqnorm(final_model_2$residuals)
qqline(final_model_2$residuals)
```



```
plot(final_model_2$residuals~final_model_2$fitted.values)
```



This plot shows how residuals compare with fitted values. They are clearly normally distributed (random scatter around zero). Also, there is constant variability (No fan-shape). There is just one extreme outlier for a fitted value greater than 13.

2.4.2 Section 4.2 Final Model RMSE

For your final model, calculate and briefly comment on the RMSE.

In order to calculate rmse, let us first create the same variables we created in the training data-set:

```
df2<-df2%>%mutate(average_porch=(Open.Porch.SF+Enclosed.Porch+X3Ssn.Porch+Screen.Porch)/4)
df2<-df2%>%filter(!(Foundation=='Wood'))
```

Let us calculate RMSE for training data:

```
resid_final_train<-exp(df$log_price)-exp(final_model_2$fitted.values)
sqrt(mean(resid_final_train^2))
```

```
## [1] 27093.81
```

We see that we are getting a rmse value of 27093$.

Now, on the test data:

```
predict_final_test<-predict(final_model_2,df2)
resid_final_test<-exp(df2$log_price)-exp(predict_final_test)
sqrt(mean(resid_final_test^2))
```

```
## [1] 22129.62
```

This value (2219.62$) is lower than that obtained on the training data. So once again, our model is doing better on test data.

Moreover, this final model is a substantial improvement over the initial model with a decrease in rmse of about 5000$

2.4.3 Section 4.3 Final Model Evaluation

What are some strengths and weaknesses of your model?

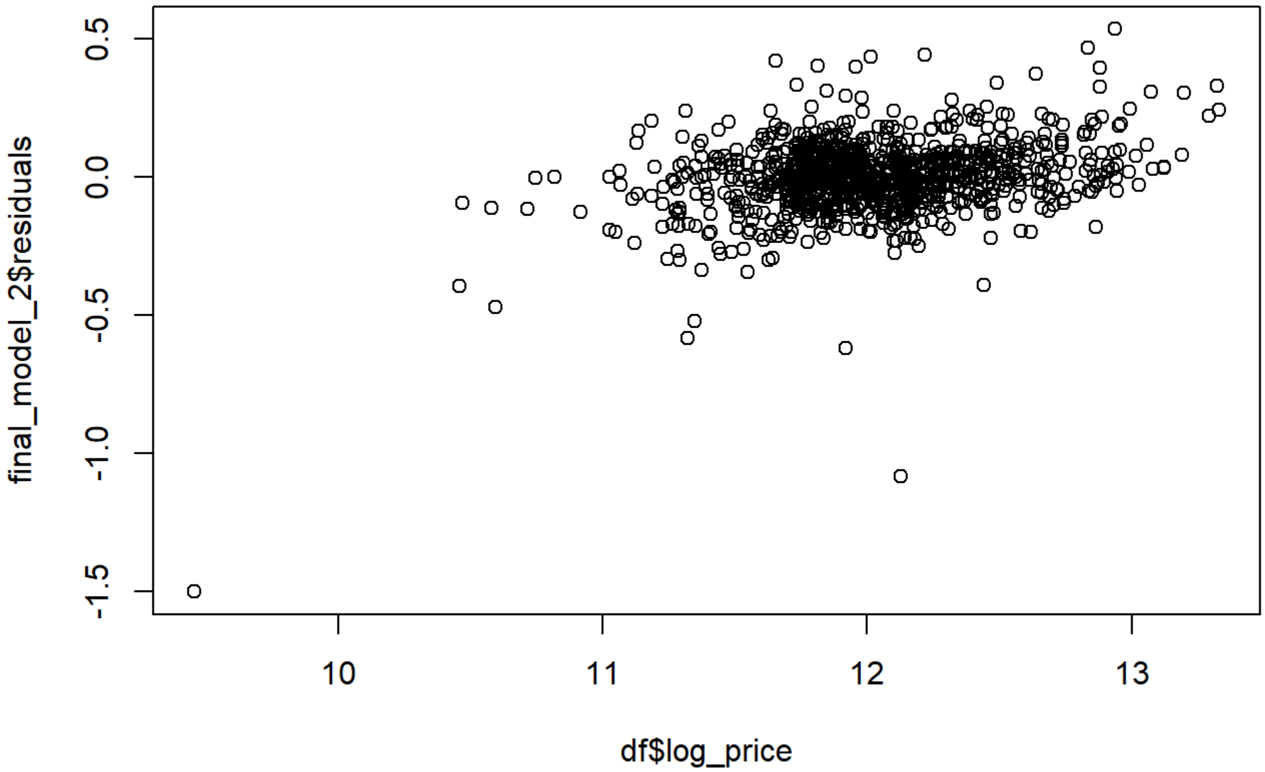
Strengths: Our model has a high adjusted R-squared of about 0.8911 which means that it explains 89% of variability in the data and thus has good predictive performance.

RMSE for test data is less than that of training data which means that we do not have the problem of over fitting.

Since variables from mostly all attributes are included, our model is robust.

Weaknesses: Some categorical variables were extremely imbalanced and could lead to biased estimates.

```
plot(final_model_2$residuals~df$log_price)
```



This plot shows that there are some homes with exceptionally low prices having large negative residuals. This means that the model has over-valued them.

```
combined<-data.frame(cbind(df2,resid_final_test^2))
head(combined%>%group_by(Neighborhood)%>%summarise(mean=mean(resid_final_test.2),median=median(r
esid_final_test.2),number=n())%>%arrange(desc(mean)))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 6 x 4
##   Neighborhood        mean     median number
##   <fct>              <dbl>      <dbl>  <int>
## 1 NoRidge      3036923563. 482318044.     28
## 2 Veenker      1498367885. 552787034.      3
## 3 StoneBr      1468368047. 310139283.     13
## 4 NridgHt      1062851448. 458956544.     30
## 5 ClearCr       981547992. 629051152.     14
## 6 Timber        681188674. 438004622.     16
```

Grouping square residuals by neighborhood, we can see that NoRidge neighborhood is a weakness of our model.

2.4.4 Section 4.4 Final Model Validation

Testing your final model on a separate, validation data set is a great way to determine how your model will perform in real-life practice.

You will use the “ames_validation” dataset to do some additional assessment of your final model. Discuss your findings, be sure to mention: * What is the RMSE of your final model when applied to the validation data? * How does this value compare to that of the training data and/or testing data? * What percentage of the 95% predictive confidence (or credible) intervals contain the true price of the house in the validation data set? * From this result, does your final model properly reflect uncertainty?

```
load("ames_validation.Rdata")
df3<-ames_validation
```

Before prediction on validation data, we need to perform the same operations we performed on previous data frames:

```
df3<-df3%>%mutate(log_price=log(price))
df3<-df3%>%mutate(log_area=log(area))
df3<-df3%>%mutate(log_lot_area=log(Lot.Area))

df3<-df3%>%mutate(Bsmt.Qual=factor(ifelse(is.na(df3$Bsmt.Qual),'No_Bsmt',Bsmt.Qual)))
df3<-df3%>%mutate(Bsmt.Cond=factor(ifelse(is.na(df3$Bsmt.Cond),'No_Bsmt',Bsmt.Cond)))
df3<-df3%>%mutate(Bsmt.Exposure=factor(ifelse(is.na(df3$Bsmt.Exposure),'No_Bsmt',Bsmt.Exposure)))
df3<-df3%>%mutate(BsmtFin.Type.1=factor(ifelse(is.na(df3$BsmtFin.Type.1),'No_Bsmt',BsmtFin.Type.1)))
df3<-df3%>%mutate(BsmtFin.Type.2=factor(ifelse(is.na(df3$BsmtFin.Type.2),'No_Bsmt',BsmtFin.Type.2)))
df3<-df3%>%mutate(Fireplace.Qu=factor(ifelse(is.na(df3$Fireplace.Qu),'No_Fireplace',Fireplace.Qu)))
df3<-df3%>%mutate(Garage.Qual=factor(ifelse(is.na(df3$Garage.Qual),'No_Garage',Garage.Qual)))
df3<-df3%>%mutate(Garage.Cond=factor(ifelse(is.na(df3$Garage.Cond),'No_Garage',Garage.Cond)))
df3<-df3%>%filter(!(is.na(BsmtFin.SF.1)),!(is.na(BsmtFin.SF.2)),!(is.na(Bsmt.Unf.SF)),!(is.na(Total.Bsmt.SF)),!(is.na(Bsmt.Full.Bath)),!(is.na(Bsmt.Half.Bath)))
df3<-df3%>%filter(!(is.na(Garage.Cars)))

df3<-df3%>%filter(!(Foundation=='Wood'))
df3<-df3%>%filter(!(MS.Zoning=='A (agr)'))

df3$Exter.Qual<-factor(df3$Exter.Qual,ordered=T,levels=c('Po','Fa','TA','Gd','Ex'))
df3$Exter.Cond<-factor(df3$Exter.Cond,ordered=T,levels=c('Po','Fa','TA','Gd','Ex'))
df3$Heating.QC<-factor(df3$Heating.QC,ordered=T,levels=c('Po','Fa','TA','Gd','Ex'))
df3$Kitchen.Qual<-factor(df3$Kitchen.Qual,ordered=T,levels=c('Po','Fa','TA','Gd','Ex'))
df3$Functional<-factor(df3$Functional,ordered=T,levels=c('Sal','Sev','Maj2','Maj1','Mod','Min2','Min1','Typ'))

df3<-df3%>%mutate(average_porch=(Open.Porch.SF+Enclosed.Porch+X3Ssn.Porch+Screen.Porch)/4)
```

Now let us test our model on validation data!

```
predict_final_val<-predict(final_model_2,df3)
resid_final_val<-exp(df3$log_price)-exp(predict_final_val)
sqrt(mean(resid_final_val^2))
```

```
## [1] 21899.06
```

We are getting a rmse value of 21899.06$ This is lower than that on both training and testing data! This means that our model is doing very well.

Now, to calculate coverage probability:

```
predict_final_cov_val<-predict(final_model_2,df3,interval='prediction')
mean(df3$log_price>predict_final_cov_val[,'lwr']&df3$log_price<predict_final_cov_val[,'upr'])
```

```
## [1] 0.9802111
```

We are getting a coverage probability of 0.98 which means that the true values fall within the intervals of prediction 98% of the time.

Thus, uncertainty is well reflected in this model.

2.5 Part 5 Conclusion

Provide a brief summary of your results, and a brief discussion of what you have learned about the data and your model.

1. From the EDA we explored the relationship between different variables in the data-set. For example, t-test for difference between two means was used to find out if mean log_price is different for homes with and without Central Air conditioning. Similarly, other tests were used to identify statistical significance.
2. log_transforming certain variables like area, price lead to better fits and are thus recommended.
3. On the initial model, maximum adjusted R-squared was obtained by the AIC criteria.
4. RMSE values of the final model on training and test data are 27093$ and 22129.62$ which are a significant improvement over the initial model which had the same values as 30770.06$ and 27601$.
5. Certain variables like average_porch were created in order to represent other porch variables.
6. Our final model is doing better on test data which means it does not suffer from over fitting.
7. Highest squared residuals were obtained for NoRidge neighborhood.
8. A lot of variables in the data set had null values which meant absence of that feature. Encoding them as a separate category helped in our analysis.
9. Certain categorical variables like Heating.QC were ordinal in nature but encoded as nominal by default. They were encoded as ordinal variables later as needed.