Prediction

Summary

The final model is $price \sim saledate^4 + gba + grade + bathrm^2 + fireplaces + hf\_bathrm^2$ where x_i^j is the jth square of x_i.

Preprocessing

Missing Data

price: I take the square-root of the response variable price
yr_rmdl: I replaced the missing data (NA) in yr_rmdl with the mean 2006.
stories: I replaced the missing data (NA) in stories with the mean and rounded it to integer 2

Transformation

price: I squared-root the response variate price.
saledate: I modify the string of date into integer "year"
grade: I modify the level of factor to "Low Quality" < "Fair Quality" < "Average" < "Above Average" < "Good Quality" < "Very Good" < "Superior"
ac: I modify "Y" to numerical value 1 and "N" to 0

Model Building

Stepwise selection with a p-value threshold of 0.05.
Main function used: `step` function in `base` package
Using x_i v.s. residual plot to determine the polynomial terms of the plots

1.Preprocessing

1.1 Loading data

```
dtrain <- read.csv("training.csv")
```

1.2 Missing Data

First finding the data which has NA value in it.

```
colnames(dtrain)[colSums(is.na(dtrain)) > 0]
```

```
## [1] "yr_rmdl" "stories"
```

From the above result, both "yr_rmdl" and "stories" fields have missing data. Since the value of missing data is unknown, using the mean of other values ("yr_rmdl" = 2006, "stories" = 2) minimize the independent effect of NA values.

```
dtrain$yr_rmdl[is.na(dtrain$yr_rmdl)] <- 2006
dtrain$stories[is.na(dtrain$stories)] <- 2
```

I used the mean of years rounded to integers as replace of the NA data in "yr_rmdl"
I used the mean of stories rounded to integers as replace of the NA data in "stories"

1.3 Perdictor Transformation

1.3.1 Transform "Saledate"

```
dtrain$saledate <- as.integer(substr(as.Date(dtrain$saledate),1,4))
```

Replace the "saledate"'s value with its year as numerical value

1.3.2 Transform "Grade"

```
dtrain$grade <- factor(dtrain$grade,
                       levels = c("Low Quality","Fair Quality","Average",
                                  "Above Average","Good Quality","Very Good","Superior"))
```

Since there exist a ranking behind the categorical data. Re-ranking the given data from lowest -> "Low Quality" to highest -> "Superior" in order to provide a direct meaning to lm function for a more accuracy predictions.

1.3.3 Transform "ac" to integer

```
for (row in 1:length(dtrain$ac)) {
   if (dtrain$ac[row] == "Y") {
    dtrain$ac[row] = as.integer(1)
   }
   if (dtrain$ac[row] == "N") {
    dtrain$ac[row] = as.integer(0)
  }
}
dtrain$ac = as.integer(dtrain$ac)
```

In order change categorical data to numerical data, replace "ac" with Y to 1 means exist, and "N" to 0 means not exists.

1.4 Outliers

```
nums <- unlist(lapply(dtrain, is.numeric))
dim(dtrain[ , nums])
```

```
## [1] 1303   16
```

Since categorical is hard to have any outliers, filtering out the numerical predictors. There are 14 (15 - price) numerical predictor and n = 1303

```
hat_mean = (14 + 1) / 1303
fit <- lm(price ~., data=dtrain[ , nums])
leverage = data.frame(seq.int(1303) ,hatvalues(fit))[hatvalues(fit) > 2 * hat_mean, 1]
residual = data.frame(seq.int(1303) ,rstudent(fit))[rstudent(fit) > 2 | rstudent(fit) < -2, 1]
```

Since data will be influential case only if both leverage and residual are large (not normal)

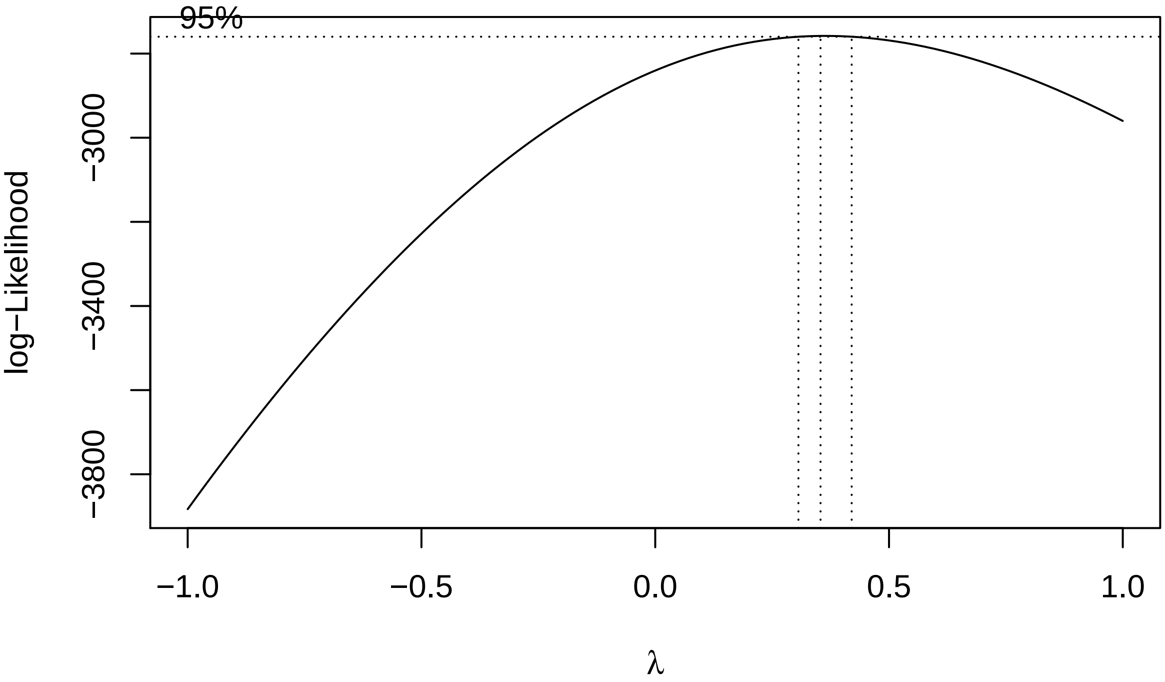
```
intersect(leverage,residual)
```

```
##  [1]    2  141  147  223  296  313  471  512  557  780  966 1236
```

Remove these influence cases from training data

1.5 Response Transformation

```
fit <- lm(price ~. , data=dtrain)
library(MASS)
boxcox(fit, lambda = seq(-1,1,1/20))
```



From above graph, the 95% confidence interval are closed to lambda 0.5 Therefore, I decided to using square-root of response value

2.Model Selection

2.1 Checking correlation between price and predictors

```
cor(dtrain[ , nums])[,"price"]
```

```
##           X      bathrm   hf_bathrm          ac       rooms       bedrm 
## -0.39949090  0.53201366  0.18517412  0.25323748  0.36836444  0.46695990 
##         ayb     yr_rmdl         eyb     stories    saledate       price 
## -0.06812407  0.21799058  0.23270733  0.24321614  0.67242944  1.00000000 
##         gba    kitchens  fireplaces    landarea 
##  0.49482898  0.14529565  0.20414081  0.14668340
```

From the correlation between price and all numerical predictors, bathrm (0.53201366), bedrm (0.46695990), saledate(0.67242944), and gba (0.49482898) have higher correlation with the response variable “price”.

```
cor(dtrain[ , nums])[,"bedrm"]
```

```
##          X     bathrm  hf_bathrm         ac      rooms      bedrm        ayb 
## -0.0661952  0.6177090  0.1511272  0.2333872  0.6446478  1.0000000  0.1024059 
##    yr_rmdl        eyb    stories   saledate      price        gba   kitchens 
##  0.1654267  0.3209646  0.2357411  0.2353410  0.4669599  0.5824250  0.1628645
## fireplaces   landarea
##  0.1026266  0.1924927
```

However, the “bedrm” predictors has strong correlation with many other predictors (bathrm,rooms,gba). These predictors also have strong correlations with response variable. In order to avoid misleading result, “bedrm” will not be consider as an useful predictor.

2.2 Stepwise Model Slection

Using “stepwise” method to auto-select the fitted model for response variable. (results are to long to be included)

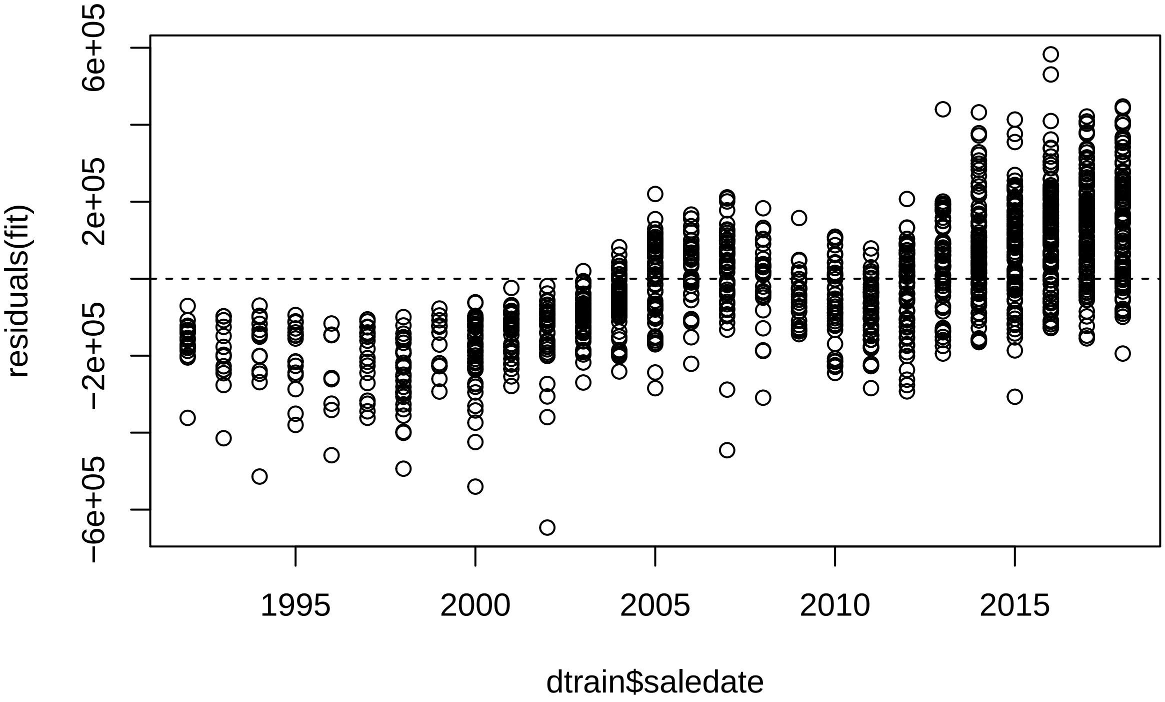
```
nullmodel<-lm(price ~1,data=dtrain)
fullmodel<-lm(price ~., data=dtrain)
summary(step(nullmodel,scope=list(upper=fullmodel),direction="both"))
```

From the stepwise algorithm (process too large to be included), the AIC of the model shows significantly decrease until “price~ saledate + gba + grade+ ayb + bathrm+ fireplaces”. This result include most of the predictors which have strong correlation with price.
Also from the p-value, hf_bathrm has the p-value lower than 0.05 which meets our sharehold. However, ayb’s correlation is the smallest among all predictors. I decided to not include it in the model.
Therefore, in order to choose the best model between simplify and accuracy, price~ saledate + gba + grade + bathrm + fireplaces + hf_bathrm" will be the best fit model.

2.3 Polynoimal and Higher terms in the model using residual plot

Since residual versus all predictor (x_i) will shows the corrsponding patterns when there exist any higher order term, plotting all predictors and getting following result.

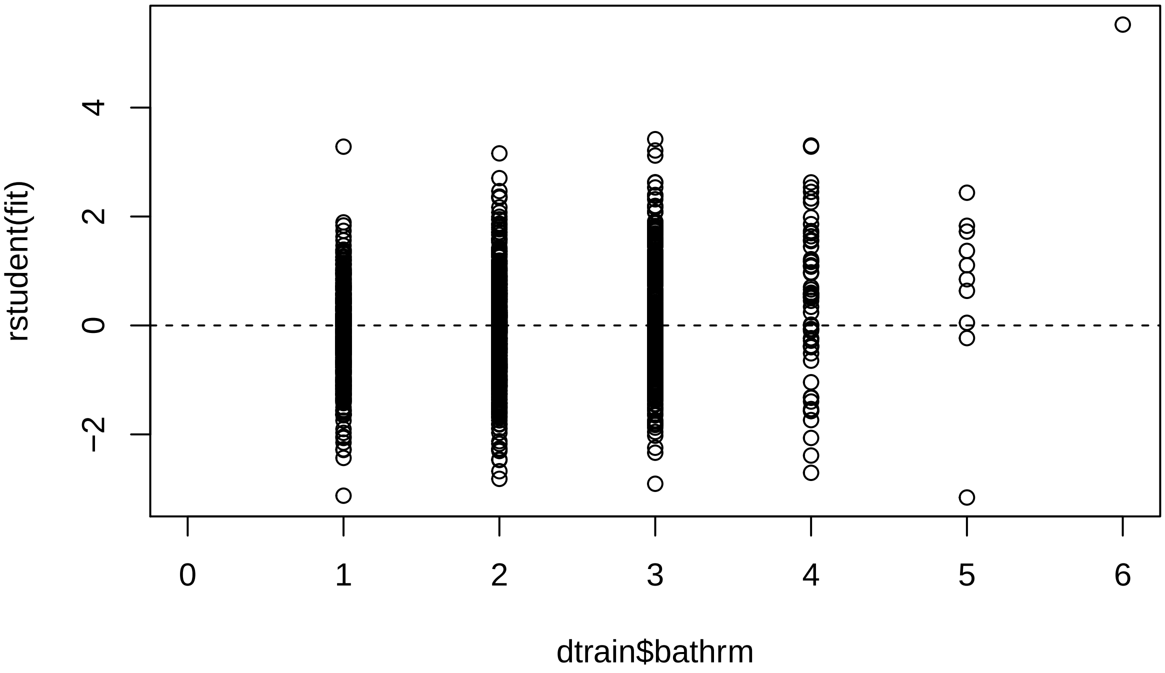
```
fit <- lm(price ~ gba + grade + bathrm + fireplaces + hf_bathrm, data=dtrain)
plot(dtrain$saledate, residuals(fit))
abline(h=c(0), lty=2)
```



From the “saledate” v.s. residual graph, we can see its close to a 3-polynomial pattern in “saledate” field.

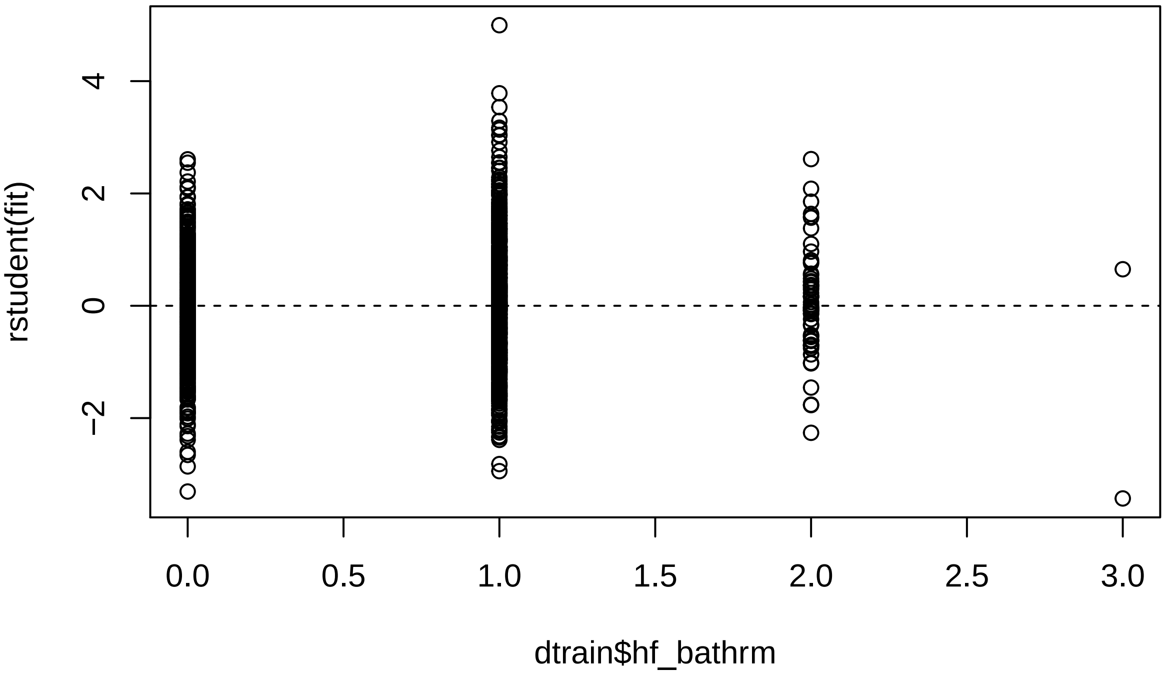
However, the function reaches the second local minimum at saledate = 2013 and increase again. This change does not follow the pattern of order-3 polynomial. I decided to increase the order of saledate. Therefore, apply saledateˆ4 to the model

```
fit <- lm(price ~ saledate + gba + grade + fireplaces + hf_bathrm, data=dtrain)
plot(dtrain$bathrm, rstudent(fit))
abline(h=c(0), lty=2)
```



From the "bathrm" v.s. residual graph, we can see its close to a 2-polynomial (increase from 0, residual reaches highest when bathrm reaches 3 and decrease after it) pattern in "bathrm" field, apply bathrmˆ4 to the model

```
fit <- lm(price ~ saledate + gba + grade + bathrm + fireplaces, data=dtrain)
plot(dtrain$hf_bathrm,rstudent(fit))
abline(h=c(0), lty=2)
```



From the "hf_bathrm" v.s. residual graph, we can see its close to a 2-polynomial (increase from 0 , residual reaches the lowest when hf_bathrm reaches around 1 and increase after it) pattern in "ayb" field, apply bathrm^2 to the model

2.4 ANOVA of final model

```
fit <- lm(sqrt(price) ~ poly(saledate,4) + gba + grade  +
            poly(bathrm,2) + fireplaces + poly(hf_bathrm, 2), data=dtrain)
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: sqrt(price)
##                      Df   Sum Sq Mean Sq  F value    Pr(>F)
## poly(saledate, 4)     4 20059681 5014920 738.6538 < 2.2e-16 ***
## gba                   1  5536294 5536294 815.4476 < 2.2e-16 ***
## grade                 6  1498001  249667  36.7737 < 2.2e-16 ***
## poly(bathrm, 2)       2   572583  286292  42.1682 < 2.2e-16 ***
## fireplaces            1   454991  454991  67.0162 6.422e-16 ***
## poly(hf_bathrm, 2)    2    72163   36082   5.3145  0.005028 **
## Residuals          1286  8731001    6789
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Applying F-test to the model that selected from above, all the p-value is much lower than 0.05. Therefore, there is a strong evidence against the null hypothesis (There is no relationship between predictor and response variable). Therefore, I thinks it is a good model for this dataset.