STAT5002 Assignment

Preparation

Introduction

The Ames Iowa Housing data set contains detailed information from the Ames Assessor's Office used in computing assessed values for individual residential properties sold in Ames, IA from 2006 to 2010. The given question in this assignment utilizes the above describable data set to calculate some statistic problem. At the very beginning of the section, data cleaning is illustrated, followed by the explanation on each question and problem.

Data Cleaning

Based on the content in problem one and two, we need four values and two categories in the data set. The values are "SalePrice", "Lot.Area", "Overall.Qual", and "MS.SubClass". The categories are basement and pool. Since we only need to analyze the probability that a select a random household has a corresponding terms, I use "Bsmt.Qual" to represents the basement category and using "Pool.QC" to represents the pool category. Both of them are variables depicting the quality of amenities. If the quality variable has the NA value, it means this house does not contain such amenity. Here, I change the data quality variable from range characters to integer. For instance, if the quality is Ex(meaning excellent), it will be reassigned a value with 5.

```
# read the file and retrieve target data
total_data = read.table("AmesHousing-1.txt",sep = "\t", header = T, stringsAsFactors = FALSE)
colSums(sapply(total_data, is.na))
target_data = subset(total_data, select = c("Bsmt.Qual", "Pool.QC", "SalePrice", "Lot.Area",
"Overall.Qual", "MS.SubClass"))
colSums(sapply(target_data, is.na))

# basement
target_data$Bsmt.Qual[target_data$Bsmt.Qual == "Ex"] = 5
target_data$Bsmt.Qual[target_data$Bsmt.Qual == "Gd"] = 4
target_data$Bsmt.Qual[target_data$Bsmt.Qual == "TA"] = 3
target_data$Bsmt.Qual[target_data$Bsmt.Qual == "Fa"] = 2
target_data$Bsmt.Qual[target_data$Bsmt.Qual == "Po"] = 1
target_data$Bsmt.Qual[is.na(target_data$Bsmt.Qual)] = 0
# pool has the same operation method as basement
target_data$Pool.QC[target_data$Pool.QC == "Ex"] = 5
```

Problem 1

Q1 - the selected household has a basement

```
basement = target_data[,c("Bsmt.Qual")]
total_num = length(basement)
basement_num = sum(basement > 0)
basement_prop = basement_num/total_num
basement_prop
```

The probability that a random selected householder having a basement equals to the number of house with Basement divide by total sample population, which is **0.9726962.**

Q2 - the selected household has a pool

```
pool = target_data[,c("Pool.QC")]
pool_num = sum(pool > 0)
pool_prop = pool_num/total_num
pool_prop
```

The probability that a random selected householder having a pool is **0.00443686.**

Q3 - the selected household has a pool and a basement

```
bp_prop = basement_prop*pool_prop
bp_prop
```

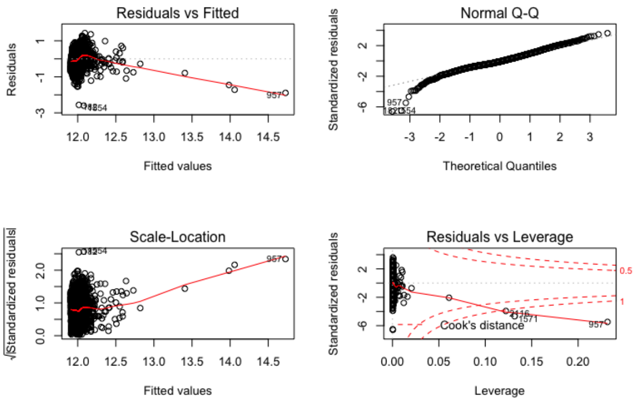
Here we use the join of two datas calculated from previous questions to reach the result. The result is **0.004315717**.

Problem 2

Q1 - determine the best module among 4 given bivariate modules

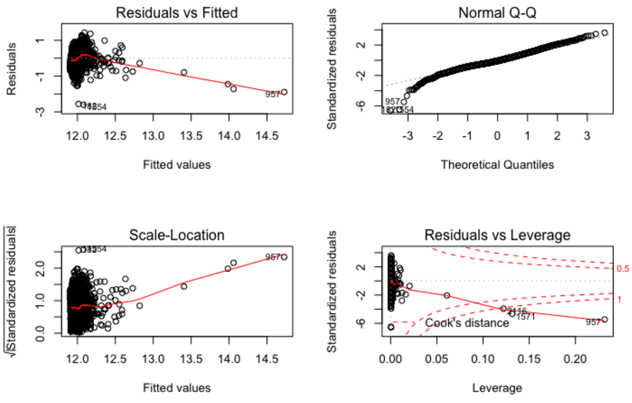
```
price = target_data[,c("SalePrice", "Lot.Area", "Overall.Qual", "MS.SubClass")]
# define four model
model1 = lm(SalePrice~Lot.Area,data = price)
summary(model1)
par(mfrow=c(2,2))
plot(model1)
# redo the above work to get information about model2 model3 and model4
```

Model 1: y = b0 + b1x + e



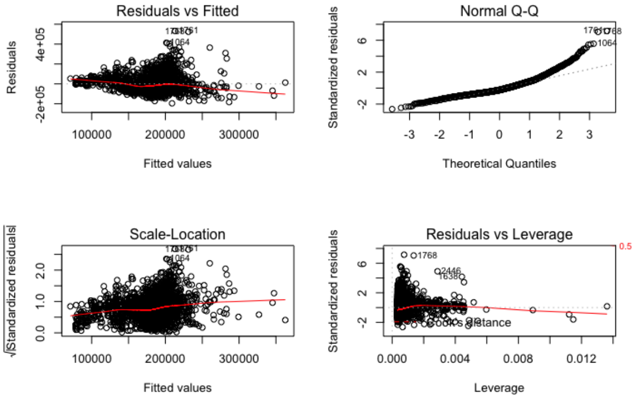
Multiple R-squared: 0.06504
Adjusted R-squared: 0.06472

Model 2: log(y) = b0 + b1x + e



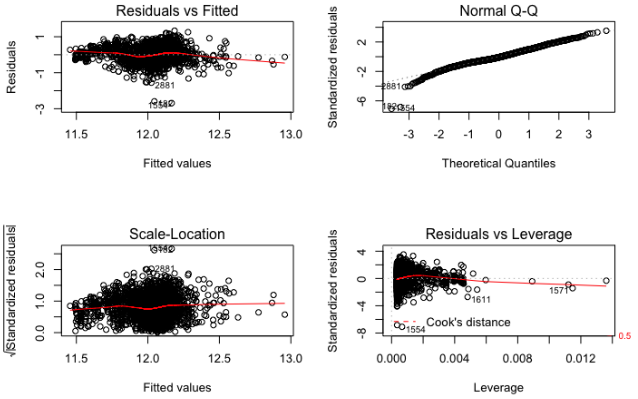
Multiple R-squared: 0.06504
Adjusted R-squared: 0.06472

Model 3: y = b0 + b1log(x) + e



Multiple R-squared: 0.1333
Adjusted R-squared: 0.133

Model 4: log(y) = b0 + b1log(x) + e



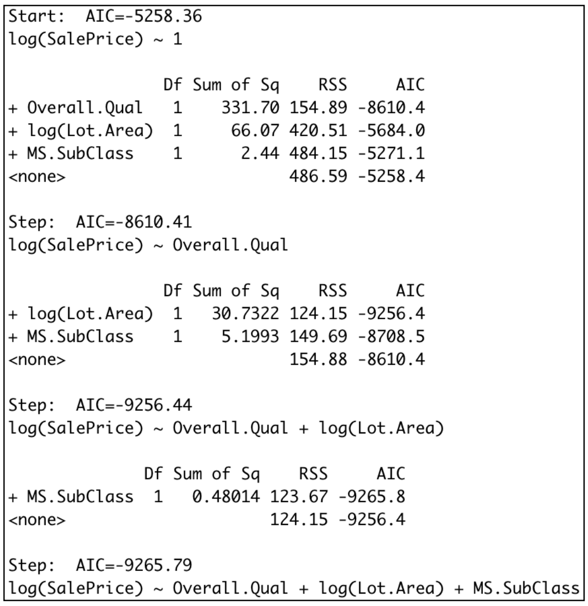
Multiple R-squared: 0.1358
Adjusted R-squared: 0.1355

According to 4 figures showed above, it can be said that **model 4 log(y) = b0 + b1log(x) + e** fits the assumption well.

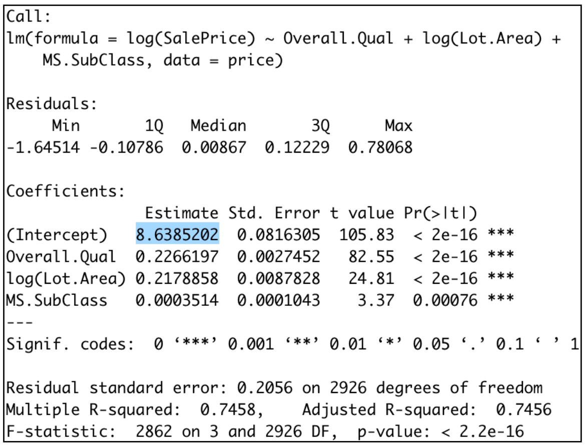
In the top left figure of model 4, there is nearly an even scatter around the red line, meaning that the residuals are randomly distributed. In the top right Q-Q plot figure, the result is an approximately straight line meaning that the set of observation is approximately normally distributed. In the bottom left figure, the red line is nearly flatten and straight, this means that the variance between variables is equal in the residuals at each value of x. In the bottom right figure, no point is outside the red dot line meaning that all of the points will not influence the slope or intercept of the regression line. In addition, model has the largest R-squared value.

Q2 - determine the best regression module

```
# Q2-(2) determine the best linear regression module
M0 = lm(log(SalePrice)~1, data = price)
M1 = lm(log(SalePrice)~ log(Lot.Area) + Overall.Qual + MS.SubClass, data = price)
M1.forw = step(M0, scope = list(lower = M0, upper = M1), direction = "forward", k =2)
summary(M1.forw)
```



```
Start:  AIC=-5258.36
log(SalePrice) ~ 1

                Df Sum of Sq    RSS     AIC
+ Overall.Qual   1    331.70 154.89 -8610.4
+ log(Lot.Area)  1     66.07 420.51 -5684.0
+ MS.SubClass    1      2.44 484.15 -5271.1
<none>                       486.59 -5258.4

Step:  AIC=-8610.41
log(SalePrice) ~ Overall.Qual

                Df Sum of Sq    RSS     AIC
+ log(Lot.Area)  1   30.7322 124.15 -9256.4
+ MS.SubClass    1    5.1993 149.69 -8708.5
<none>                       154.88 -8610.4

Step:  AIC=-9256.44
log(SalePrice) ~ Overall.Qual + log(Lot.Area)

              Df Sum of Sq    RSS     AIC
+ MS.SubClass  1   0.48014 123.67 -9265.8
<none>                     124.15 -9256.4

Step:  AIC=-9265.79
log(SalePrice) ~ Overall.Qual + log(Lot.Area) + MS.SubClass
```



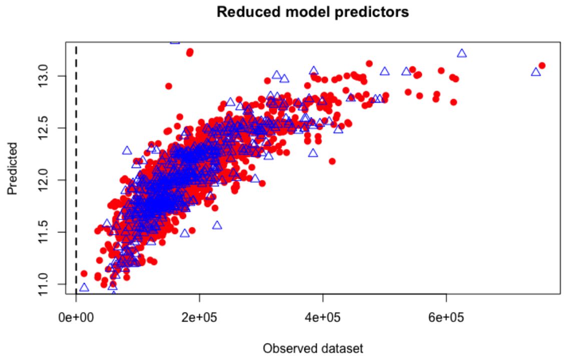
```
Call:
lm(formula = log(SalePrice) ~ Overall.Qual + log(Lot.Area) +
    MS.SubClass, data = price)

Residuals:
     Min       1Q   Median       3Q      Max
-1.64514 -0.10786  0.00867  0.12229  0.78068

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.6385202  0.0816305  105.83  < 2e-16 ***
Overall.Qual  0.2266197  0.0027452   82.55  < 2e-16 ***
log(Lot.Area) 0.2178858  0.0087828   24.81  < 2e-16 ***
MS.SubClass   0.0003514  0.0001043    3.37  0.00076 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2056 on 2926 degrees of freedom
Multiple R-squared:  0.7458,    Adjusted R-squared:  0.7456
F-statistic:  2862 on 3 and 2926 DF,  p-value: < 2.2e-16
```

Based on the result from question1, we set or starting model as log(y) = b0 + b1log(x) + e. We test the module using F-test and AIC, in the order of forwarding. The execution results suggests that,

log(SalePrice)= b0 + b1log(Lot.Area) + b2 Overall.Qual + b3MS.SubClass + e

Module Prediction ability Checking

```
indexes <- sample(1:nrow(target_data), size = 0.20*nrow(target_data))
valid <- target_data[indexes,]
calib <- target_data[-indexes,]
new_remod = M1
# Reduced model
plot(calib$SalePrice, predict(new_remod, newdata = calib), pch = 16, col = "red", cex = 1.2, xlab
= "Observed dataset", ylab = "Predicted", main = "Reduced model predictors")
abline(0, 1, lty = 2, lwd = 2)
points(valid$SalePrice, predict(new_remod, newdata = valid), col = "blue", pch = 2, cex = 1.2)
```



Then, we check the model by separating the sample size into 20% and 80%, finding out that almost all blue hollow triangle (validation) lies within red filled circle (calibration).

In conclusion, the module fits well and the best (parsimonious) regression model that fits the data is:

log(SalePrice)= 8.639 + 0.218log(Lot.Area) + 0.227Overall.Qual + 0.00035MS.SubClass + e

Q3

The fitted module for the given module

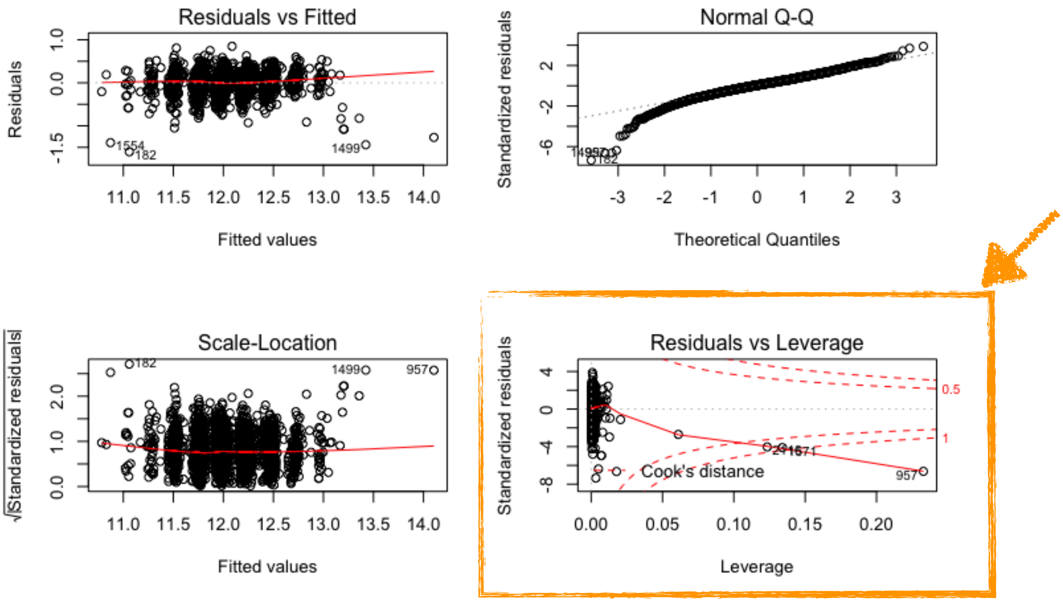
```
new_module = lm(log(SalePrice)~ Lot.Area + Overall.Qual, data = price)
summary(new_module)
```

```
Call:
lm(formula = log(SalePrice) ~ Lot.Area + Overall.Qual, data = price)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.60382 -0.11676  0.01336  0.13096  0.84913 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.050e+01  1.825e-02  575.55   <2e-16 ***
Lot.Area     9.127e-06  5.150e-07   17.72   <2e-16 ***
Overall.Qual 2.335e-01  2.876e-03   81.20   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2186 on 2927 degrees of freedom
Multiple R-squared:  0.7125,	Adjusted R-squared:  0.7123 
F-statistic:  3627 on 2 and 2927 DF,  p-value: < 2.2e-16
```

The module is, **log(SalePrice) = 10.5 + 9.127e-06* Lot.Area + 0.2335*Overall.Qual**

Any outliers?

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



Cook’s distance is a measure of outliers under a particular model and can be used to detect unusual y-value. Here, the bottom right figure it can be seen that there are three points outside of the red dotted lines. **They are the outliers in this module.**

Expected sales price

```
areas = 10000
quality = 9
predict.at = data.frame(Lot.Area = areas, Overall.Qual = quality)
x = predict(new_module, newdata = predict.at)
saleprice = exp(x)
saleprice
```

```
[1] 327120.1
```

The sale price is 327120.1