Exam 2 - Question 3

Adam McQuistan

Tuesday, April 05, 2016

Problem 3 - Do problem 6.18 on page 252.

- Only do parts (c-g).
- For part (g) do either Levene or Pagan test

Part C. Fit the regerssion model for the listed predictors below and state the estimated regression equation.

- Y (Rental Rates)
- X1 (Age)
- X2 (Operating Expense and Taxes)
- X3 (Vacancy Rate)
- X4 (Square Footage)

```
options(scipen=999)
df <- read.csv("data/6.18.csv")
result <- lm(Rental ~ Age + Expense + Vacancy + Footage, data=df)
result_smry <- summary(result)
result_smry
```

```
##
## Call:
## lm(formula = Rental ~ Age + Expense + Vacancy + Footage, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1872 -0.5911 -0.0910  0.5579  2.9441
##
## Coefficients:
##                 Estimate   Std. Error t value             Pr(>|t|)
## (Intercept) 12.200585882  0.577956174  21.110 < 0.0000000000000002 ***
## Age         -0.142033644  0.021342610  -6.655        0.00000000389 ***
## Expense      0.282016530  0.063172350   4.464        0.00002747396 ***
## Vacancy      0.619343503  1.086812829   0.570                 0.57
## Footage      0.000007924  0.000001385   5.722        0.00000019760 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.137 on 76 degrees of freedom
## Multiple R-squared:  0.5847, Adjusted R-squared:  0.5629
## F-statistic: 26.76 on 4 and 76 DF,  p-value: 0.00000000000007272
```

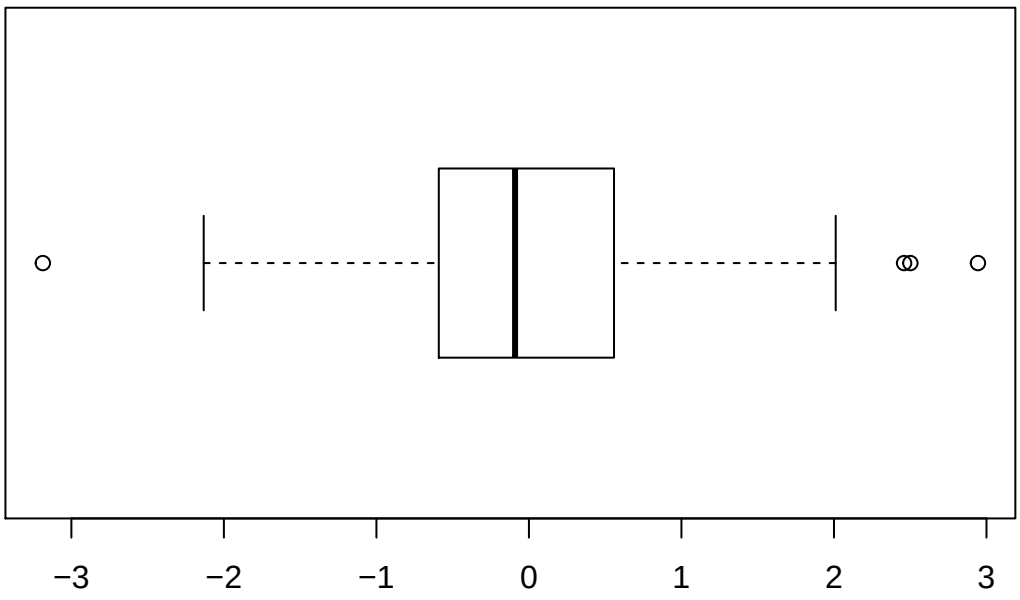
$$Rate = 12.2 - 0.142(Age) + 0.282(Expense) + 0.619(Vacancy) + 0.000008(Footage)$$

Part D. Make and interpret box plot

```
library(knitr)
iqr <- IQR(result$residuals)
smry <- summary(result$residuals)
smry <- t(as.matrix(smry, nrow=1))
smry <- data.frame(smry)
names(smry) = c("Min", "FirstQtr", "Median", "Mean", "ThirdQtr", "Max")
smry <- cbind(smry,
              Low1.5xIQR=(smry$FirstQtr - (1.5 * iqr)),
              Uppr1.5xIQR=(smry$ThirdQtr + (1.5 * iqr)),
              Low3xIQR=(smry$FirstQtr - (3 * iqr)),
              Uppr3xIQR=(smry$ThirdQtr + (3 * iqr)))
smry <- round(smry, 3)
kable(smry)
```

Min	FirstQtr	Median	Mean	ThirdQtr	Max	Low1.5xIQR	Uppr1.5xIQR	Low3xIQR	Uppr3xIQR
-3.187	-0.591	-0.091	0	0.558	2.944	-2.315	2.281	-4.038	4.005

```
boxplot(result$residuals, horizontal=T)
```



Outlier	Definition
Lower Mild	1st Qtr - 1.5 x IQR
Upper Mild	3rd Qtr + 1.5 x IQR
Lower Extreme	1st Qtr - 3 x IQR
Upper Extreme	3rd Qtr + 3 x IQR

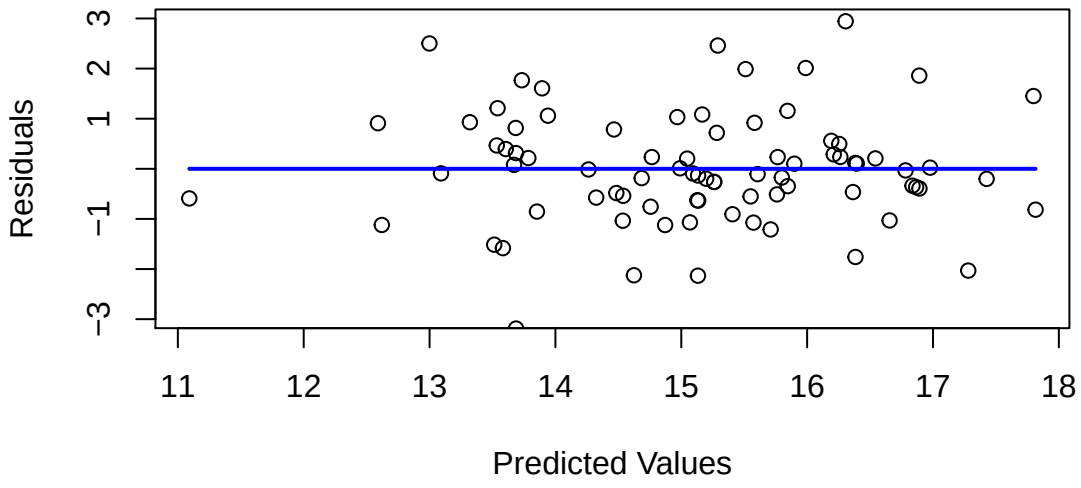
There are mild lower outliers and mild upper outliers. Fifty percent of the data is between the first and third quartiles. Overall, it is mildly skewed to the right.

Part E. Plot Residuals vs Predicted Values, Vs each explanatory variable, vs each two factor ineraction term on separate plots. Prepare a normal probability plot of the residuals. Analyze and state your findings.

```
df_model <- result$model[, 1:5]
df_model$Residuals <- result_smry$residuals
df_model$PredictedVals <- result$fitted.values
df_model$AgeExpense <- df_model$Age * df_model$Expense
df_model$AgeVacancy <- df_model$Age * df_model$Vacancy
df_model$AgeFootage <- df_model$Age * df_model$Footage
df_model$ExpenseVacancy <- df_model$Expense * df_model$Vacancy
df_model$ExpenseFootage <- df_model$Expense * df_model$Footage
df_model$VacancyFootage <- df_model$Vacancy * df_model$Footage
```

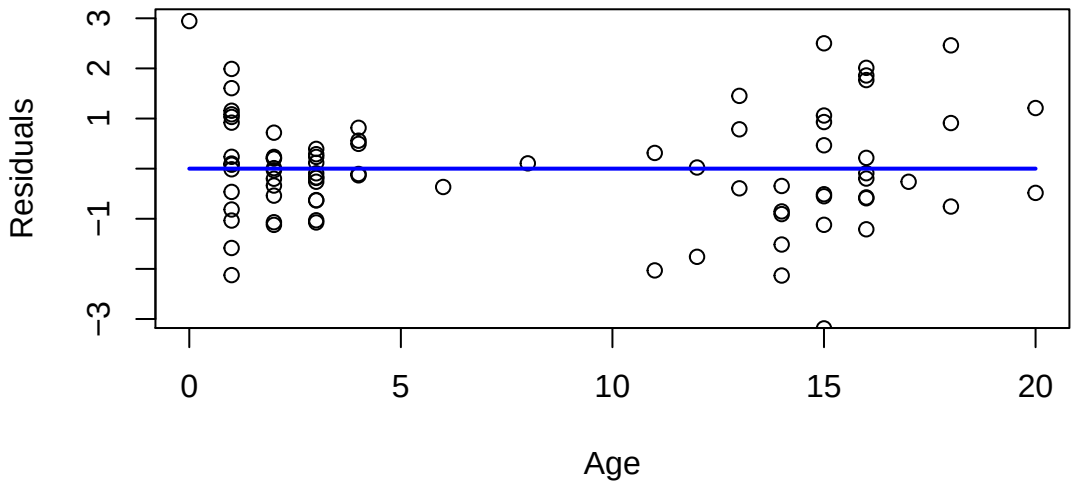
```
with(df_model, {
     plot(x=PredictedVals, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Predicted Values", ylab="Residuals", main="")

     points(c(min(PredictedVals), max(PredictedVals)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



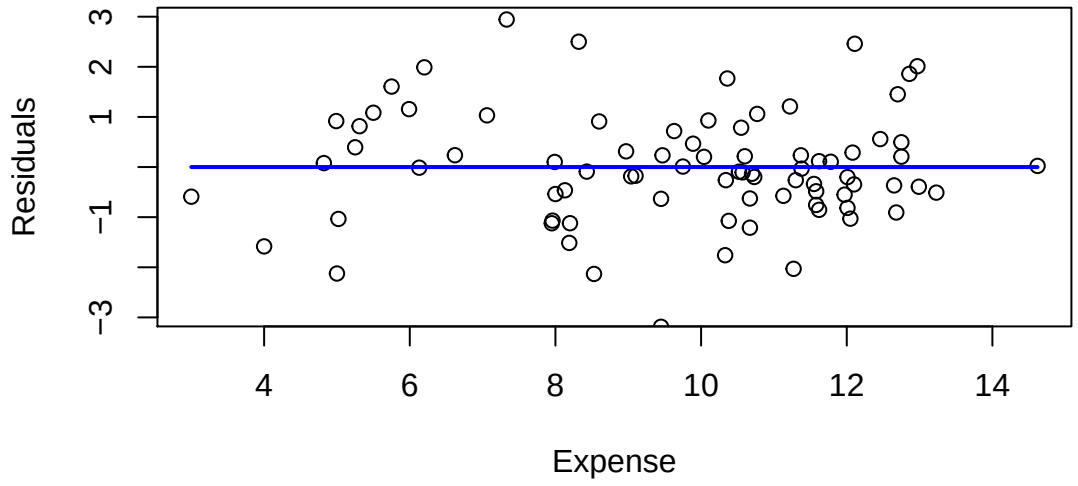

```
with(df_model, {
     plot(x=Age, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Age", ylab="Residuals", main="")

     points(c(min(Age), max(Age)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



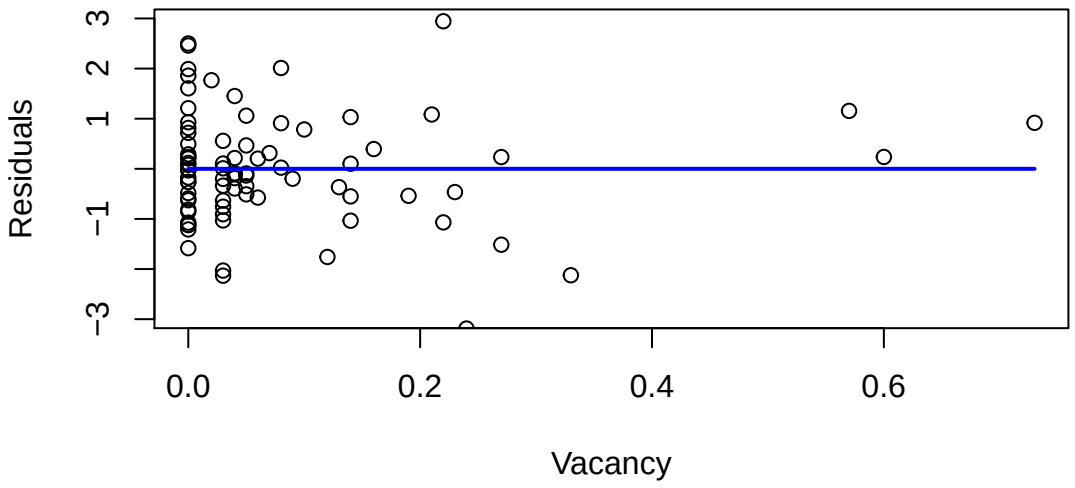
```r
with(df_model, {
     plot(x=Expense, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Expense", ylab="Residuals", main="")

     points(c(min(Expense), max(Expense)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



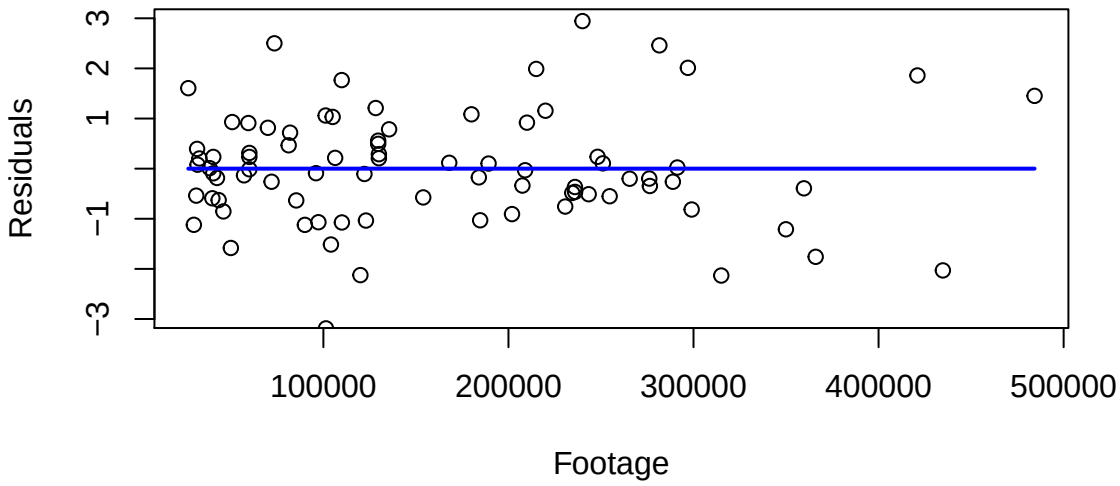

```r
with(df_model, {
     plot(x=Vacancy, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Vacancy", ylab="Residuals", main="")

     points(c(min(Vacancy), max(Vacancy)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



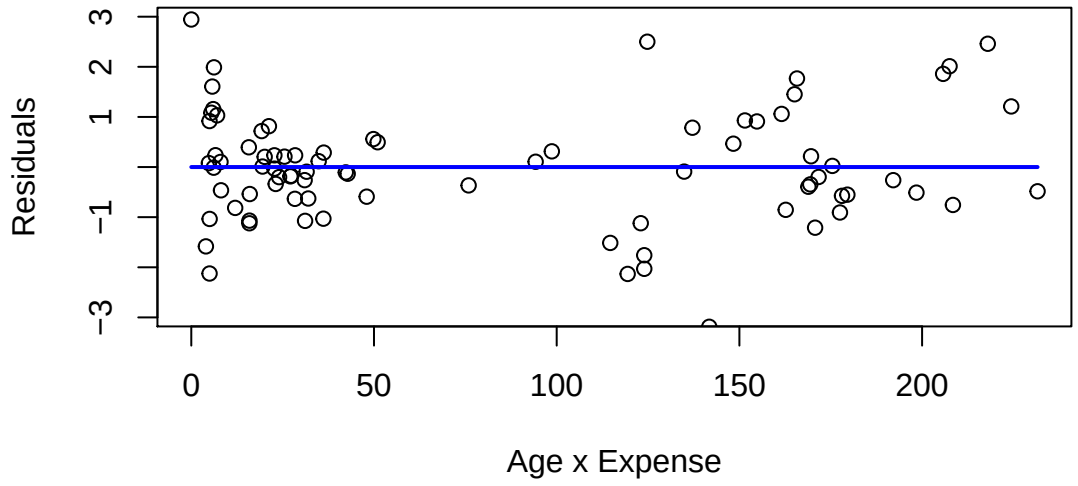

```
with(df_model, {
     plot(x=Footage, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Footage", ylab="Residuals", main="")

     points(c(min(Footage), max(Footage)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



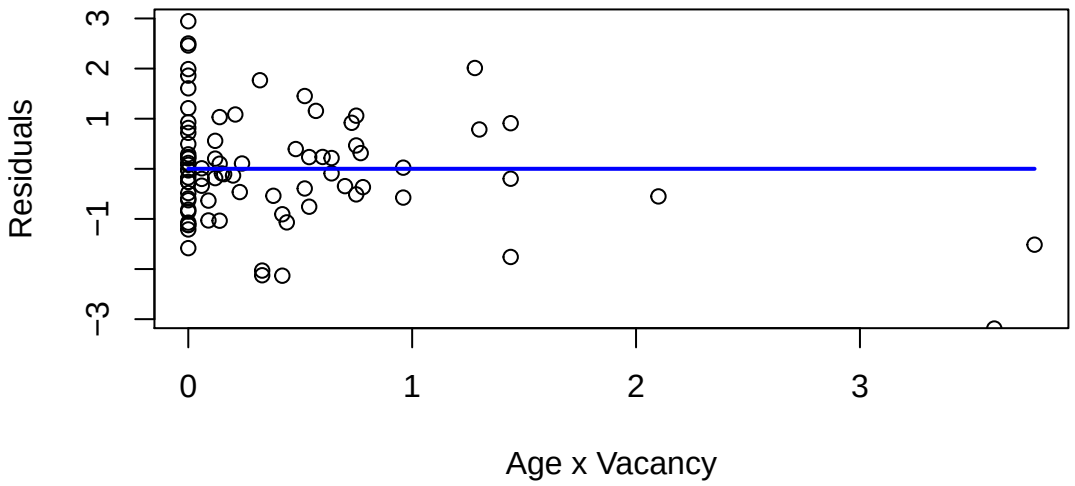
```r
with(df_model, {
     plot(x=AgeExpense, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Age x Expense", ylab="Residuals", main="")

     points(c(min(AgeExpense), max(AgeExpense)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



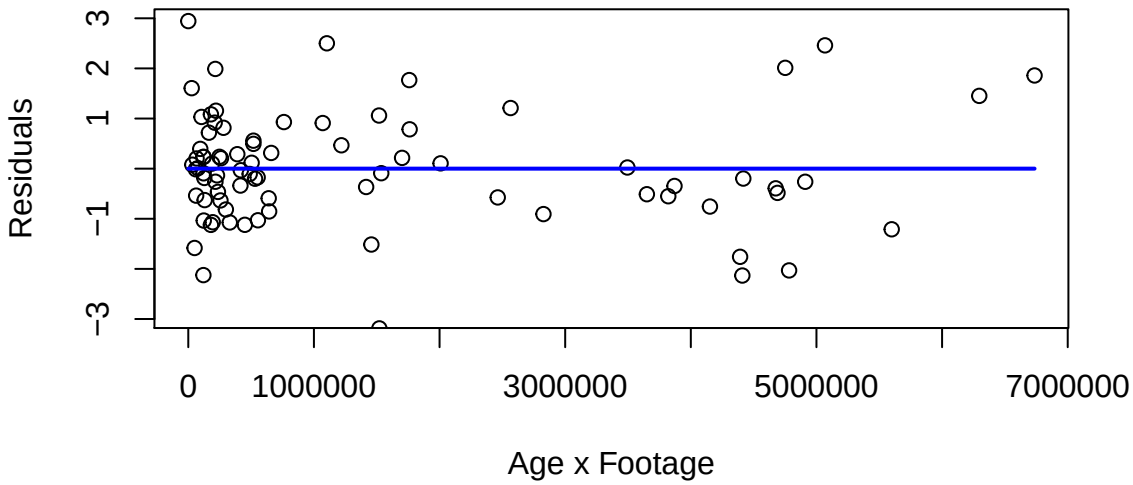

```r
with(df_model, {
     plot(x=AgeVacancy, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Age x Vacancy", ylab="Residuals", main="")

     points(c(min(AgeVacancy), max(AgeVacancy)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



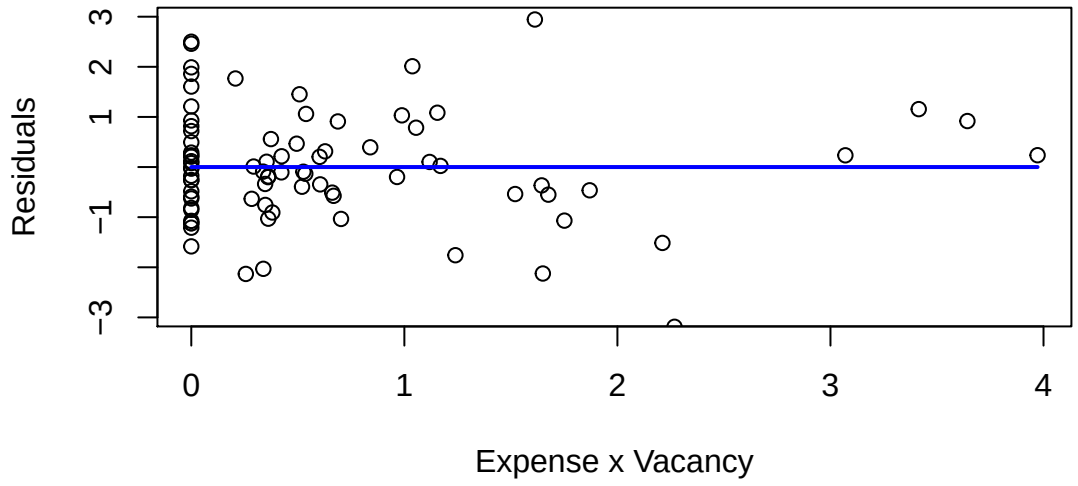

```
with(df_model, {
     plot(x=AgeFootage, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Age x Footage", ylab="Residuals", main="")

     points(c(min(AgeFootage), max(AgeFootage)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



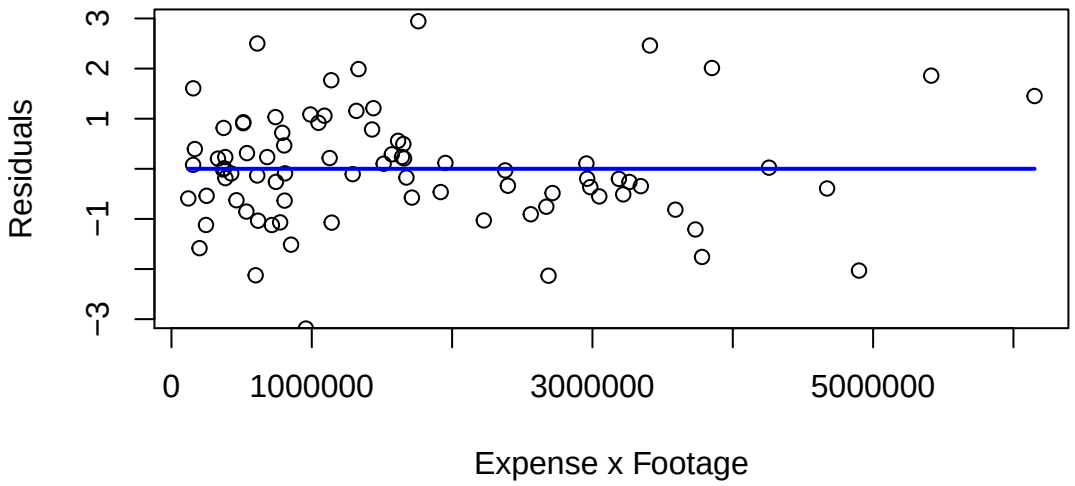
```
with(df_model, {
     plot(x=ExpenseVacancy, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Expense x Vacancy", ylab="Residuals", main="")

     points(c(min(ExpenseVacancy), max(ExpenseVacancy)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



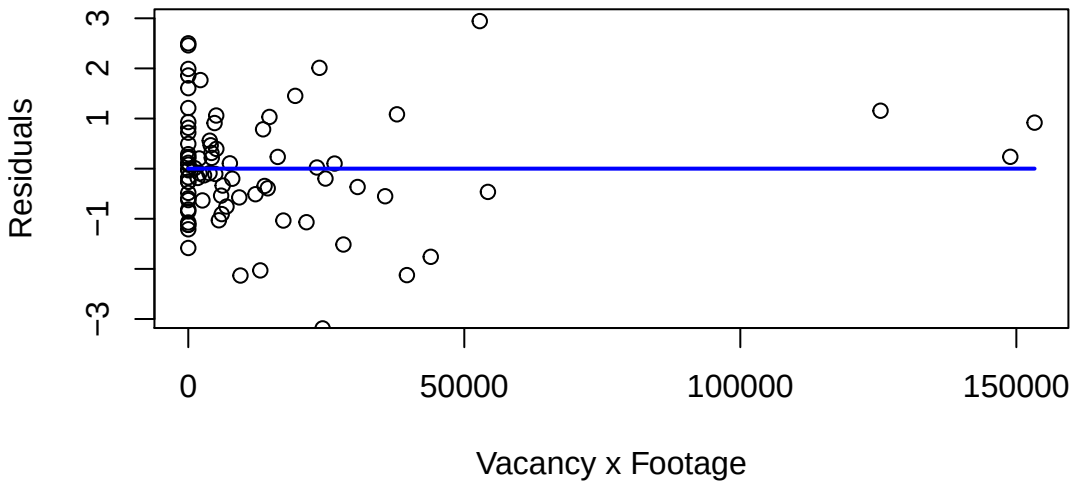

```
with(df_model, {
     plot(x=ExpenseFootage, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Expense x Footage", ylab="Residuals", main="")

     points(c(min(ExpenseFootage), max(ExpenseFootage)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```

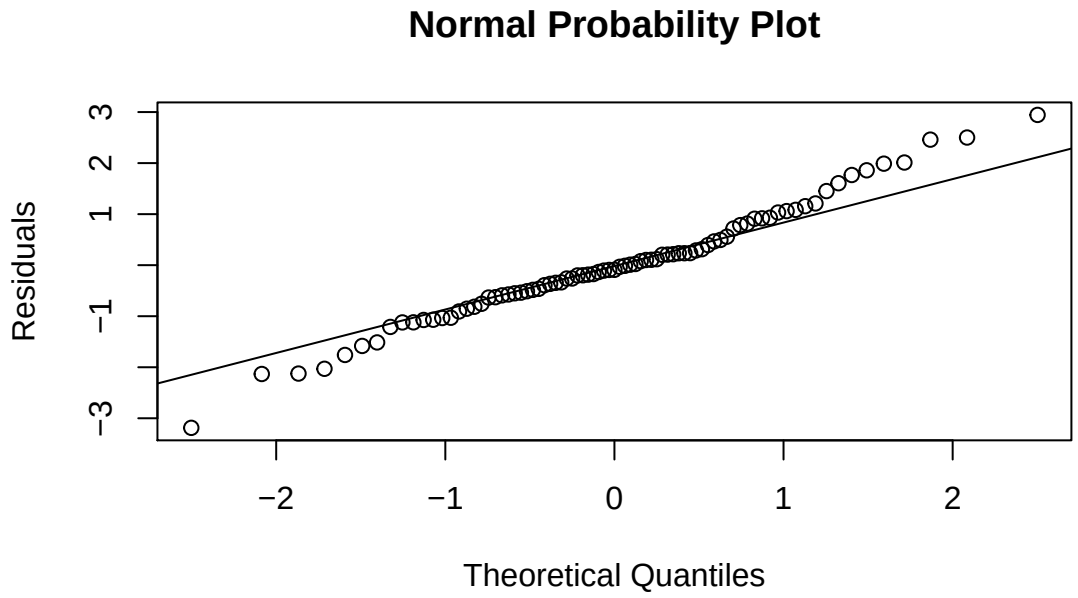



```
with(df_model, {
     plot(x=VacancyFootage, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Vacancy x Footage", ylab="Residuals", main="")

     points(c(min(VacancyFootage), max(VacancyFootage)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



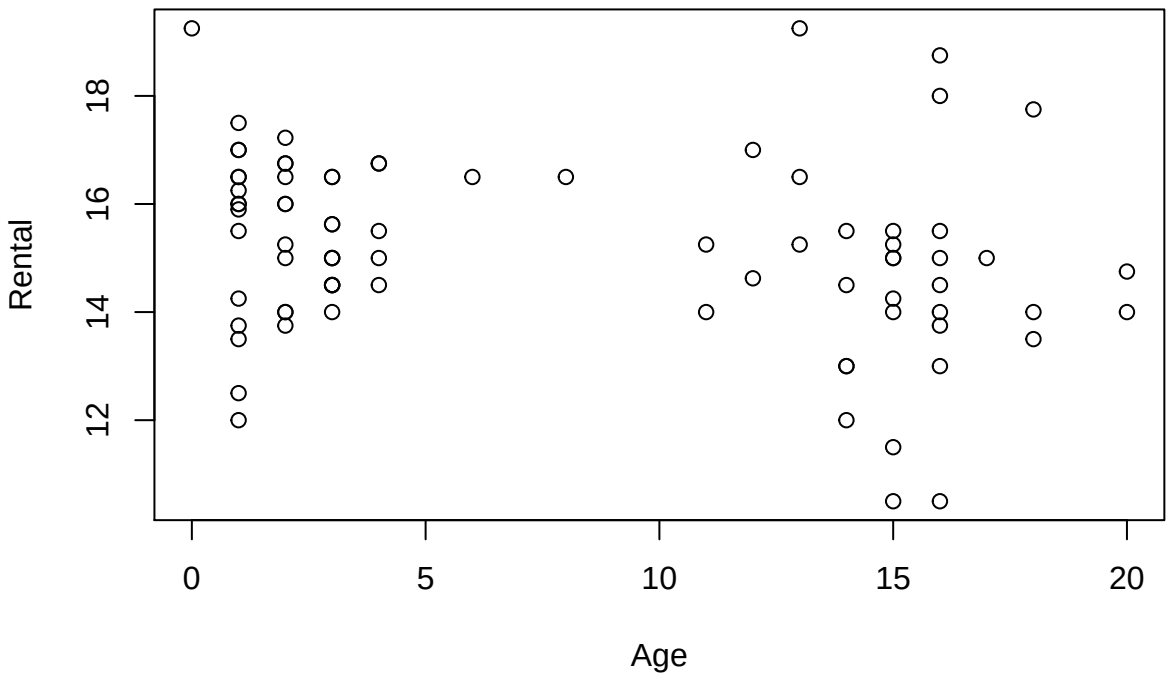
```
qqnorm(result$residuals, ylab="Residuals", main="Normal Probability Plot")
qqline(result$residuals)
```



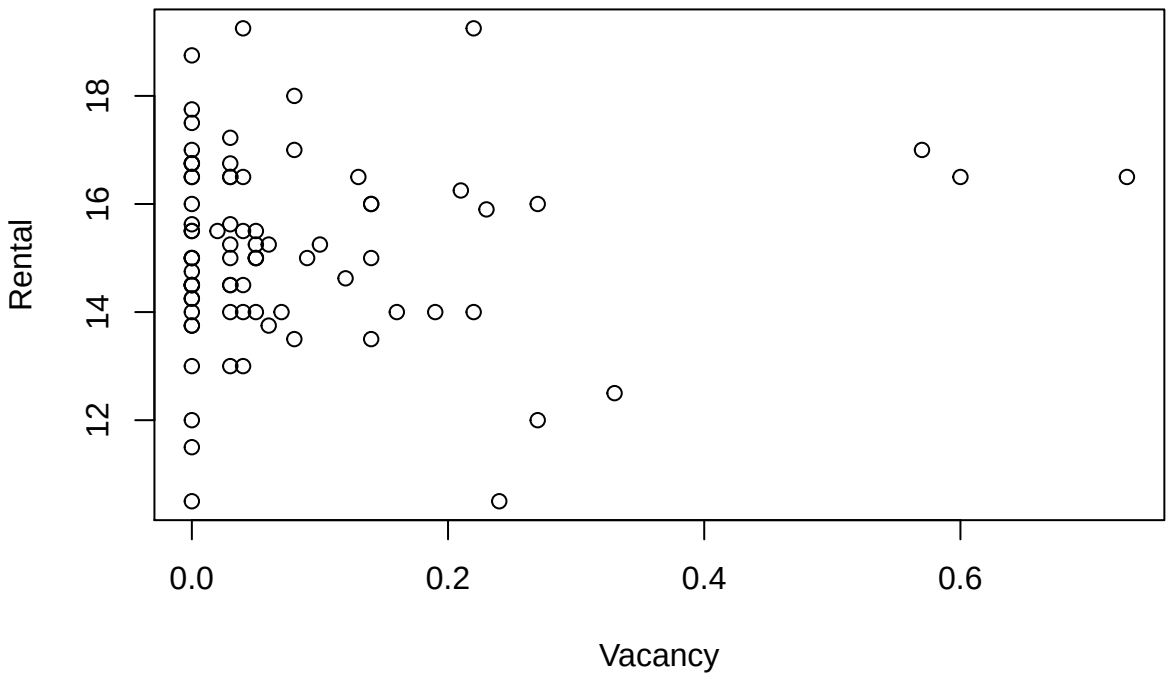
As a whole the model appears to have constant variance of the error terms but, individually some of the predictors do not have constant variance across all their values (the domain of predictor values). For example, Age and Vacancy appear to have a non-constant variance in their error terms.

Also, it does not appear that the relationship between Age or Vacancy against Rental is linear as shown below.

```
with(df, plot(x=Age, y=Rental, type="p"))
```




```
with(df, plot(x=Vacancy, y=Rental, type="p"))
```



Part F. Can you conduct a formal lack of fit test?

Again, I will have to assess whether there are enough naturally occuring repeating vectors of the combination of explanatory variables.

```
df_model$CommonLevels <- with(df_model,
                              paste(Age,
                                    Expense,
                                    Vacancy,
                                    Footage, sep=""))
```

Total Data Instances vs Unique Instances

```
n <- dim(df_model)[1]
m <- length(unique(df_model$CommonLevels))
txt = paste("Total Instances ", n, "  Unique Instances ", m, "\n", sep="")
cat(txt, sep="")
```

```
## Total Instances 81  Unique Instances 81
```

No because there are not atleast two repeating rows of predictor variabel data.

Part G. Conduct a levene test for constant variance of the different groups.

```
# order the df_model datastructure lowest to highest
# for the fitted (predicted) values
df_model <- df_model[order(df_model$PredictedVals),]
case40Prediction <- df_model$PredictedVals[40]
df_model$LeveneGrps <- ifelse(df_model$PredictedVals <= case40Prediction, "Low", "High")
library(lawstat)
with(df_model, levene.test(Residuals, as.factor(LeveneGrps), location="mean"))
```

```
##
##  classical Levene's test based on the absolute deviations from the
##  mean ( none not applied because the location is not set to median
##  )
##
## data:  Residuals
## Test Statistic = 0.2438, p-value = 0.6228
```

The Leven test assumes the two population's variances are equal

H_o: no difference in population variances

H_o: there are differences in the populations variances

Since p-value > 0.05 we do not reject the null hypothsis and conclude equal variance of the error terms between the groups.