Stat 330 Final Project

Daniel Ironhat, Daniel Brewer, Andrew Fisher, Casey Jones

11/7/2019

```
library(xlsx)
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.2.1 ──
```

```
## ✓ ggplot2 3.2.1     ✓ purrr   0.3.3
## ✓ tibble  2.1.3     ✓ dplyr   0.8.3
## ✓ tidyr   1.0.0     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.4.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(ggfortify)
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
library(dvmisc)
```

```
## Loading required package: rbenchmark
```

```
##
## Attaching package: 'dvmisc'
```

```
## The following object is masked from 'package:tidyr':
##
##     expand_grid
```

```
library(bestglm)
```

```
## Loading required package: leaps
```

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 3.0-1
```

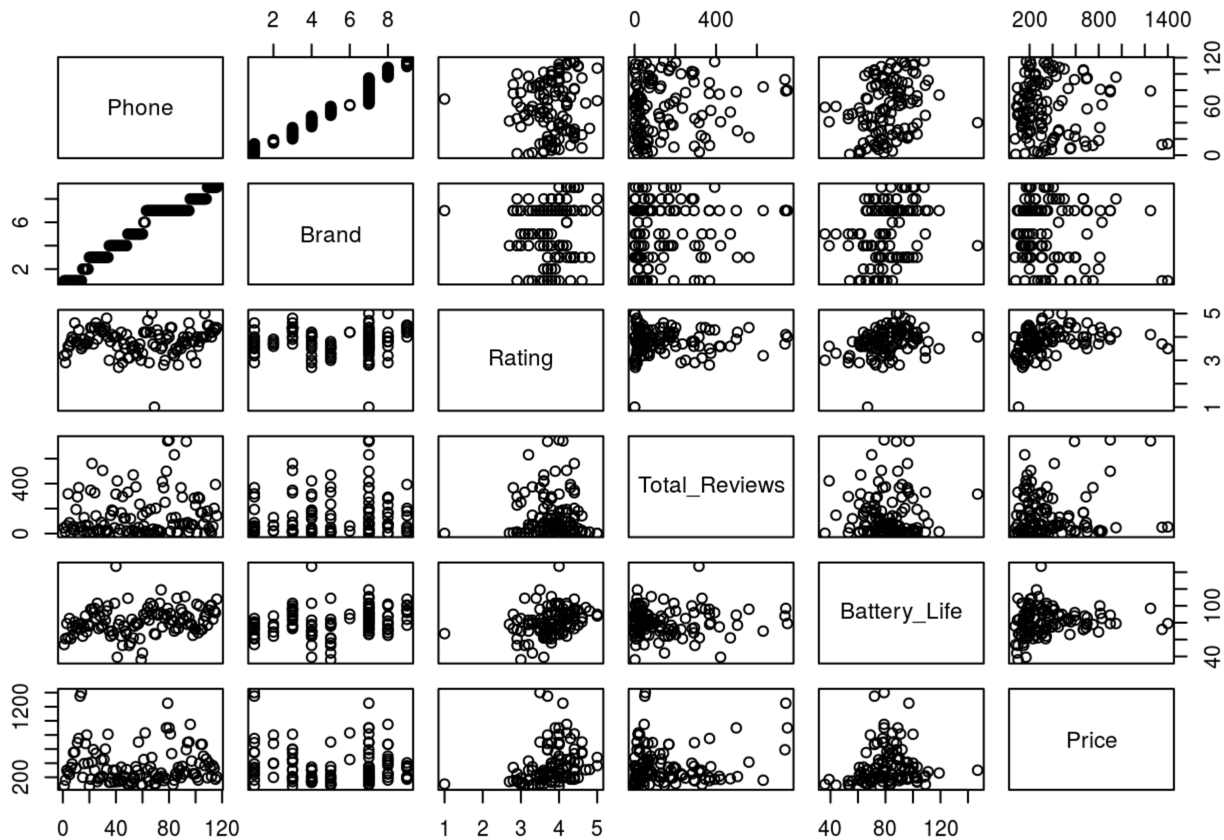
```
library(varhandle)

phone_data <- read.csv("Phone Data - Final_Data.csv", header = TRUE)
phone_data <- phone_data %>%
  select(-2)

phone_data <- read.csv("Phone Data - Final_Data.csv", header = TRUE)
phone_data <- phone_data %>%
  select(-2)

phone_price <- cbind(phone_data[,1], phone_data[,3:5],phone_data$Battery_Life,phone_data$Prices)
names(phone_price) <- c("Phone","Brand","Rating","Total_Reviews","Battery_Life","Price")
```

Part 1 Exploratory Data Analysis Methods

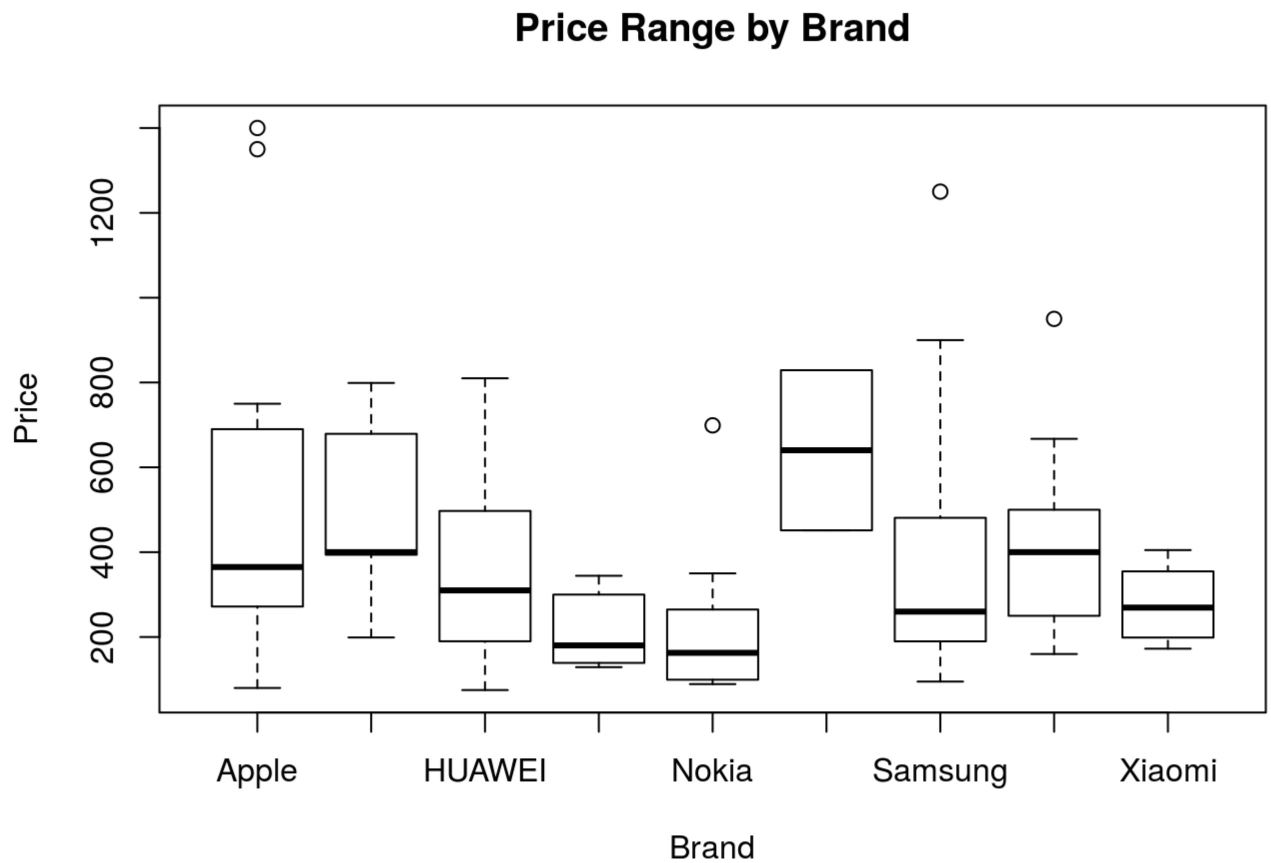
```
#Exploratory measures by AEF
#correlation matrix
cor(phone_price[,3:6])
```

```
##                    Rating Total_Reviews Battery_Life     Price
## Rating        1.000000000   0.001039107  0.337767327 0.2902744
## Total_Reviews 0.001039107   1.000000000  0.008857901 0.1187502
## Battery_Life  0.337767327   0.008857901  1.000000000 0.1074344
## Price         0.290274415   0.118750242  0.107434431 1.0000000
```

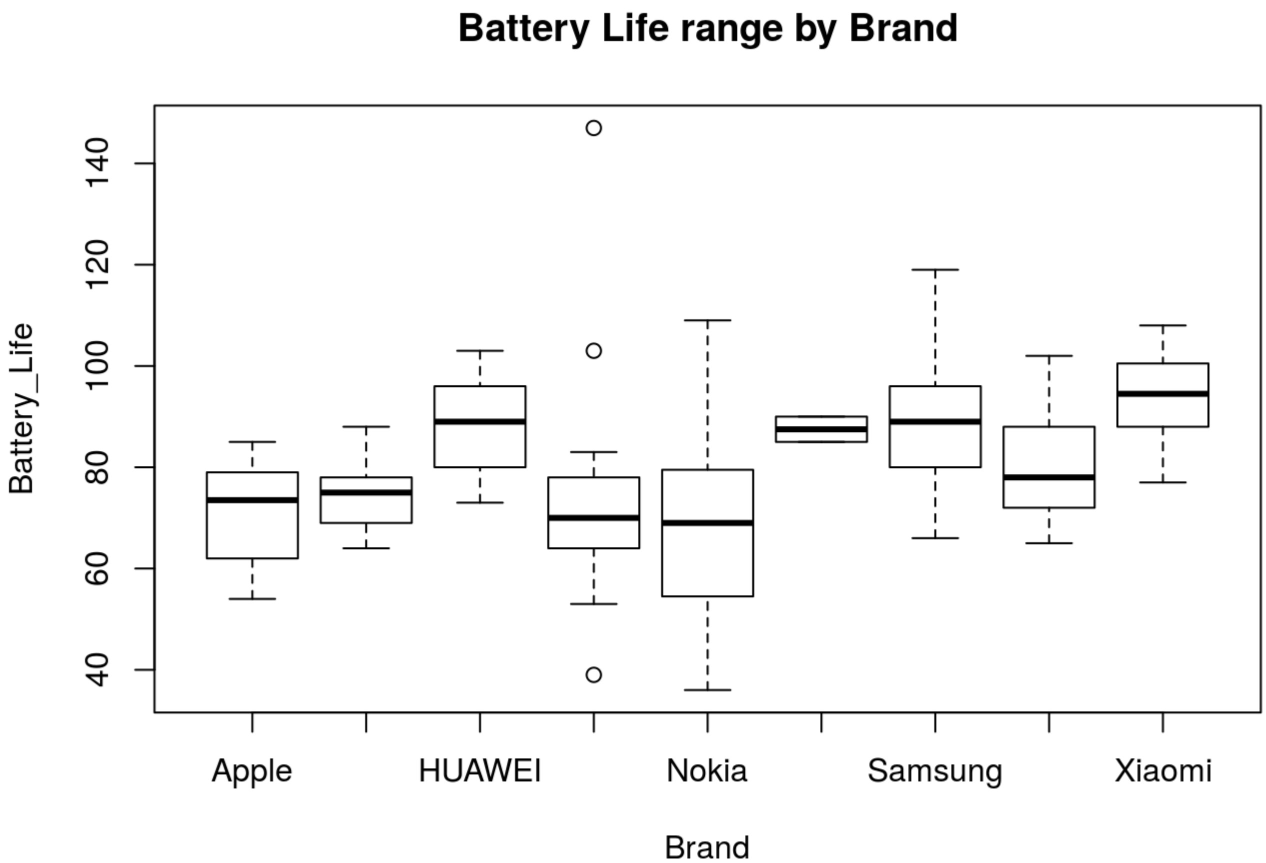
```
#Scatterplot Matrix
pairs(phone_price[,1:6])
```



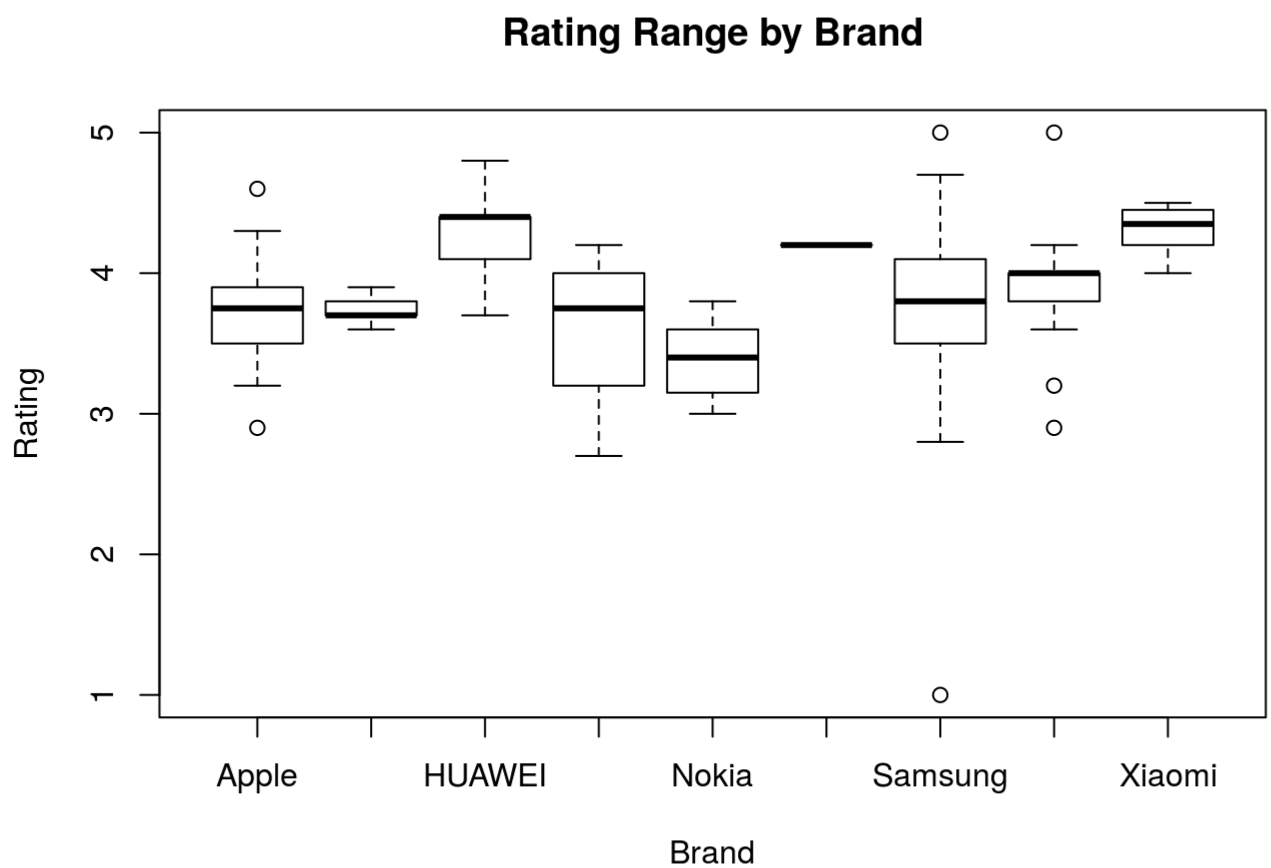

```
#Boxplots examining the brands against each other just for fun
boxplot(Price ~ Brand, data = phone_price,
        main="Price Range by Brand")
```



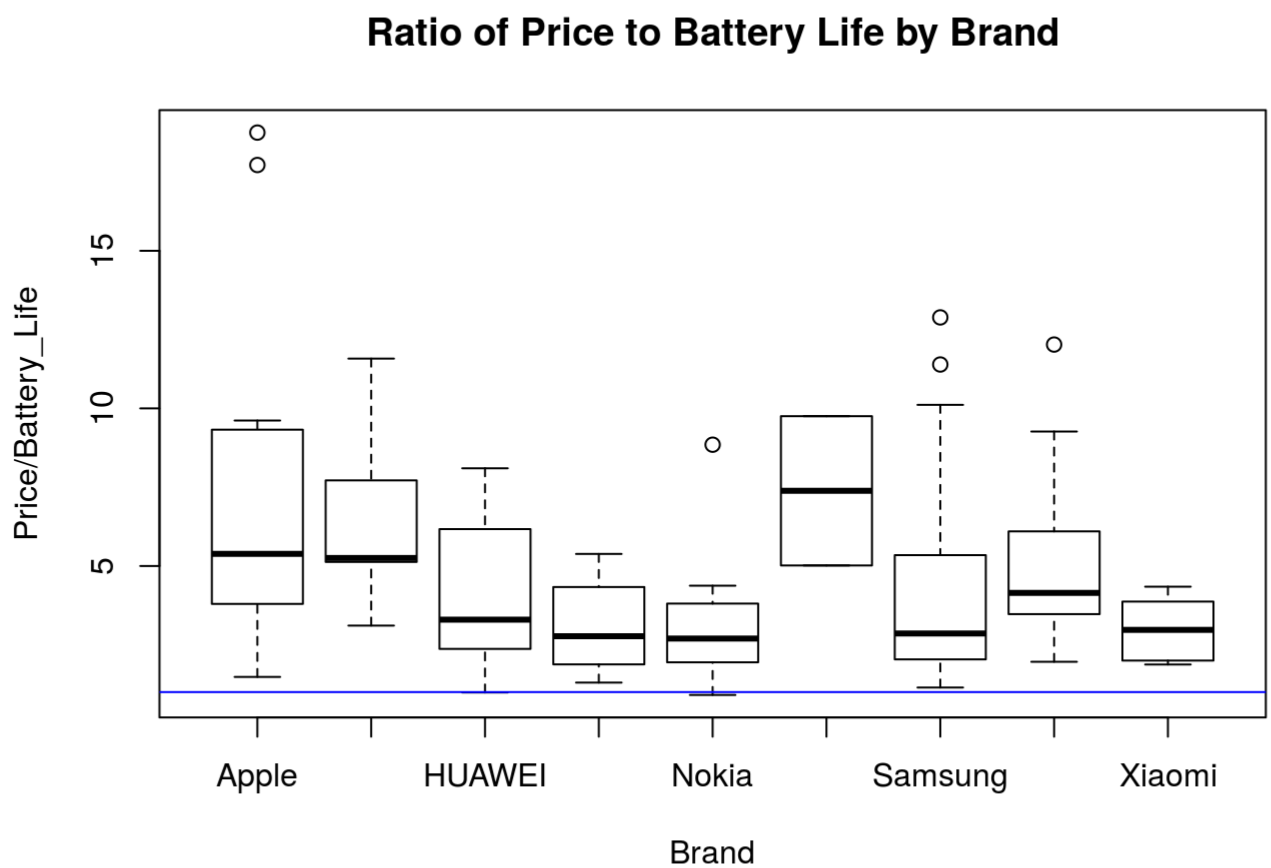

```
boxplot(Battery_Life ~ Brand, data = phone_price,
        main="Battery Life range by Brand")
```



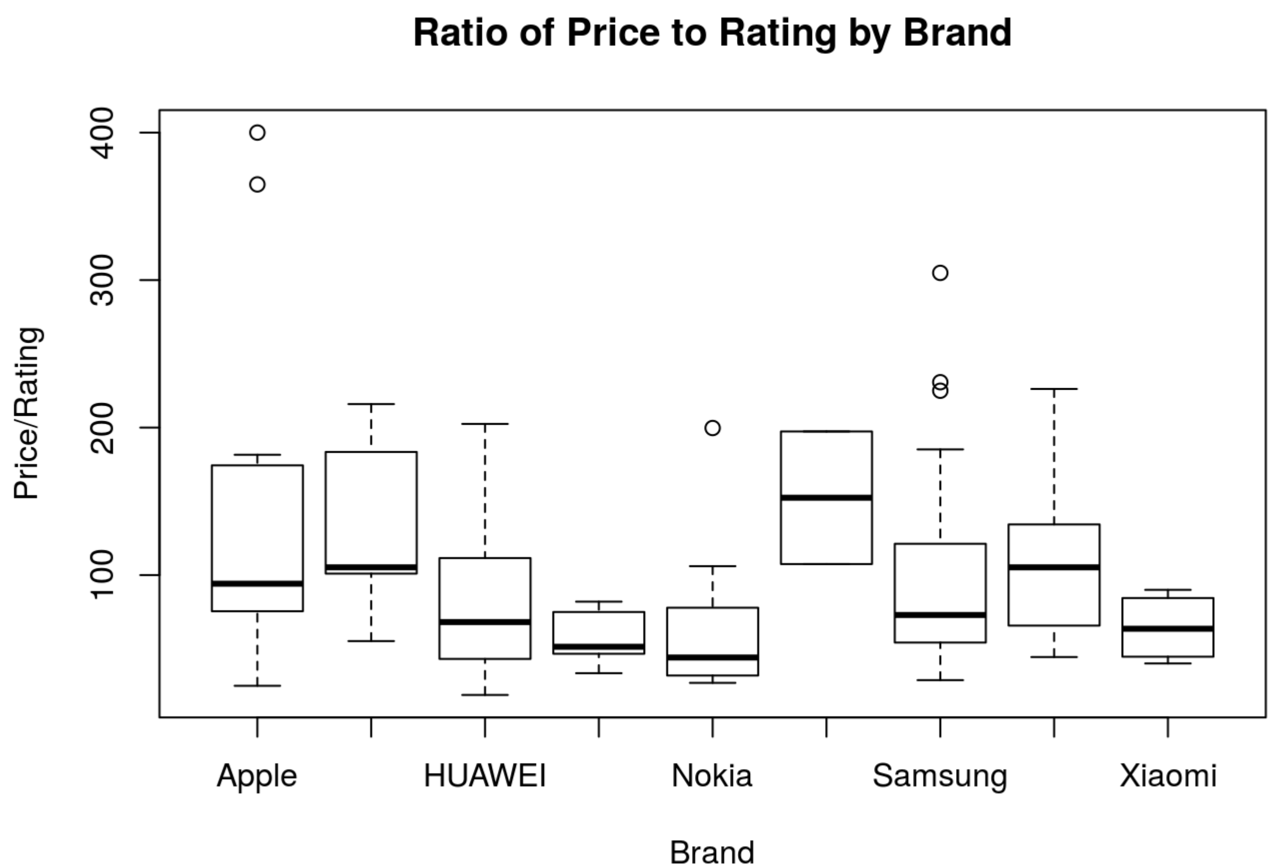

```
boxplot(Rating ~ Brand, data = phone_price,
        main="Rating Range by Brand")
```



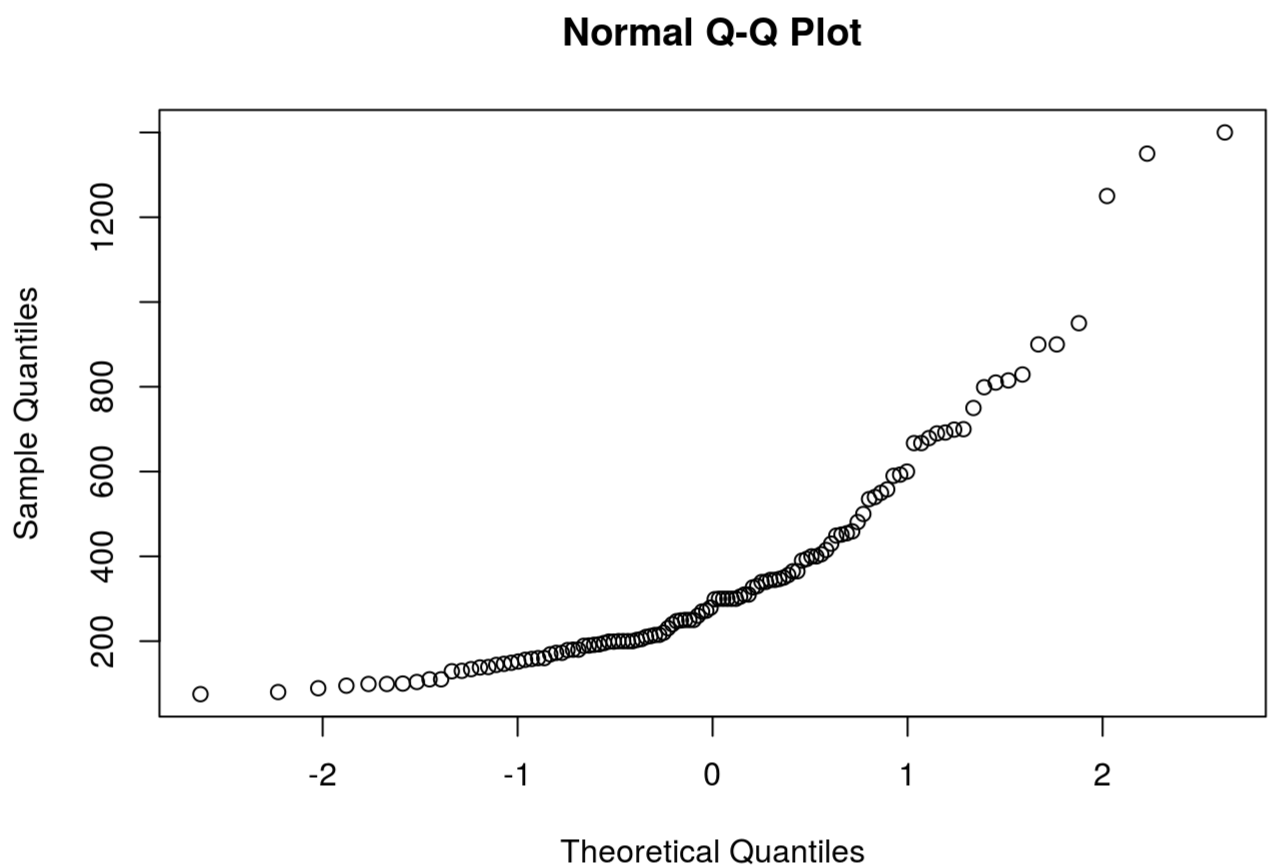

```
boxplot(Price / Battery_Life ~ Brand, data = phone_price,
        main="Ratio of Price to Battery Life by Brand")
abline(h=1, col="blue")
```



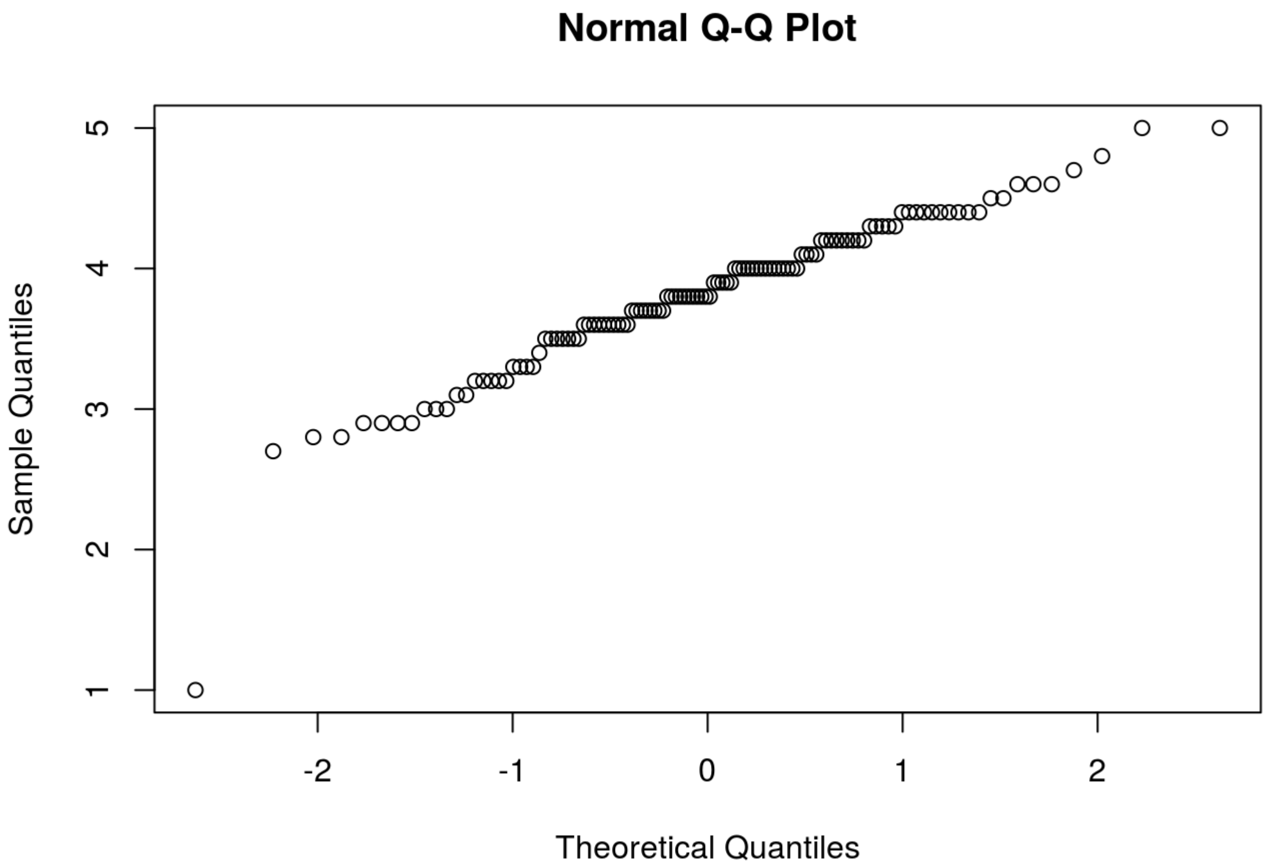

```
boxplot(Price / Rating ~ Brand, data = phone_price,
        main="Ratio of Price to Rating by Brand")
```



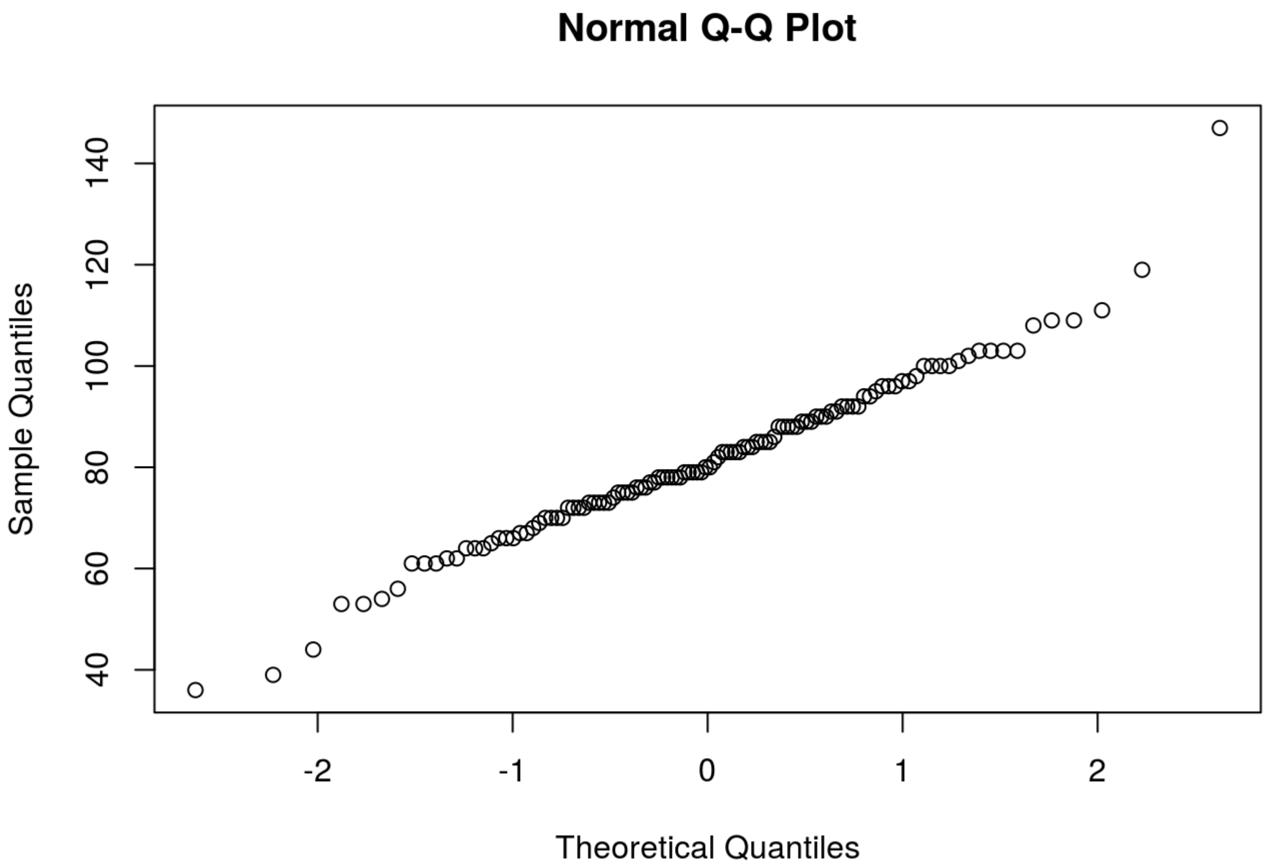

```
#Checking some assumptions
#normality
qqnorm(phone_price$Price)
```




```
qqnorm(phone_price$Rating)
```




```
qqnorm(phone_price$Battery_Life)
```




```
#Multi-co-Linearity
temp <- lm(Price~ Rating + Total_Reviews + Battery_Life
    + Brand, data = phone_price) #a naive temporary Linear Model
vif(temp)
```

```
##                   GVIF Df GVIF^(1/(2*Df))
## Rating        1.389840  1        1.178915
## Total_Reviews 1.050464  1        1.024921
## Battery_Life  1.421258  1        1.192165
## Brand         1.738160  8        1.035156
```

```
#End AEF
```

Part 2 Multiple Linear Regression

```
#DANIEL IRONHAT
#Create Linear Models
phone_rating_lm <- lm(data = phone_price, formula = Rating ~ Brand + Total_Reviews + Battery_Lif
e + Price)

phone_rating_lm2 <- lm(data = phone_price, formula = Rating ~ Brand + Price + Battery_Life)

#phone_rating_lm_int <- lm(data = phone_price, formula = Rating ~ Brand * Price * Battery_Life)
#phone_rating_lm_int <- lm(data = phone_price, formula = Rating ~ Brand * Battery_Life + Price )
#phone_rating_lm_int <- lm(data = phone_price, formula = Rating ~ Brand * Price + Battery_Life)

summary(phone_rating_lm)
```

```
##
## Call:
## lm(formula = Rating ~ Brand + Total_Reviews + Battery_Life +
##     Price, data = phone_price)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.42552 -0.18455  0.05635  0.22114  1.20466
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.944e+00  2.710e-01  10.863  < 2e-16 ***
## BrandGoogle    2.224e-02  2.502e-01   0.089  0.92935
## BrandHUAWEI    5.536e-01  1.904e-01   2.908  0.00445 **
## BrandMotorola  8.104e-02  1.928e-01   0.420  0.67509
## BrandNokia    -1.181e-01  1.981e-01  -0.596  0.55223
## BrandOnePlus   3.136e-01  3.664e-01   0.856  0.39402
## BrandSamsung  -1.594e-02  1.687e-01  -0.095  0.92489
## BrandSony      1.601e-01  1.882e-01   0.851  0.39682
## BrandXiaomi    5.986e-01  2.314e-01   2.587  0.01106 *
## Total_Reviews -3.409e-05  2.682e-04  -0.127  0.89909
## Battery_Life   6.527e-03  3.202e-03   2.039  0.04403 *
## Price          5.824e-04  1.902e-04   3.062  0.00280 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4789 on 104 degrees of freedom
## Multiple R-squared:   0.34,  Adjusted R-squared:  0.2702
## F-statistic:  4.87 on 11 and 104 DF,  p-value: 4.858e-06
```

```
summary(phone_rating_lm2)
```

```
##
## Call:
## lm(formula = Rating ~ Brand + Price + Battery_Life, data = phone_price)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4195 -0.1868  0.0505  0.2195  1.2117
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.9401994  0.2682787  10.959  < 2e-16 ***
## BrandGoogle    0.0238807  0.2486717   0.096  0.92368
## BrandHUAWEI    0.5513903  0.1887013   2.922  0.00426 **
## BrandMotorola  0.0784410  0.1908052   0.411  0.68183
## BrandNokia    -0.1205536  0.1962303  -0.614  0.54031
## BrandOnePlus   0.3162196  0.3641330   0.868  0.38715
## BrandSamsung  -0.0198249  0.1651382  -0.120  0.90467
## BrandSony      0.1585503  0.1868821   0.848  0.39815
## BrandXiaomi    0.5963871  0.2296534   2.597  0.01076 *
## Price          0.0005780  0.0001863   3.103  0.00246 **
## Battery_Life   0.0065542  0.0031798   2.061  0.04176 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4766 on 105 degrees of freedom
## Multiple R-squared:  0.3399, Adjusted R-squared:  0.277
## F-statistic: 5.406 on 10 and 105 DF,  p-value: 2.002e-06
```

```
#summary(phone_rating_lm_int)
# it appears that interactions are not significantly effecting Rating.



#Check variable selection
phone_price2 <- phone_price[ , c("Brand","Price", "Total_Reviews", "Battery_Life", "Rating")]

best_subsets_method <- bestglm(phone_price2,
                            IC = "BIC",
                            method = "exhaustive",
                            TopModels = 10)
```

```
## Morgan-Tatar search since factors present with more than 2 levels.
```

```
summary(best_subsets_method$BestModel)
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.51669 -0.29068  0.06285  0.36063  1.11728
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.7472791  0.2459901  11.168  < 2e-16 ***
## Price        0.0005462  0.0001822   2.998 0.003345 **
## Battery_Life 0.0106379  0.0029399   3.618 0.000445 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5123 on 113 degrees of freedom
## Multiple R-squared:  0.1793, Adjusted R-squared:  0.1648
## F-statistic: 12.35 on 2 and 113 DF,  p-value: 1.412e-05
```

```
best_subsets_method <- bestglm(phone_price2,
                            IC = "AIC",
                            method = "exhaustive",
                            TopModels = 10)
```

```
## Morgan-Tatar search since factors present with more than 2 levels.
```

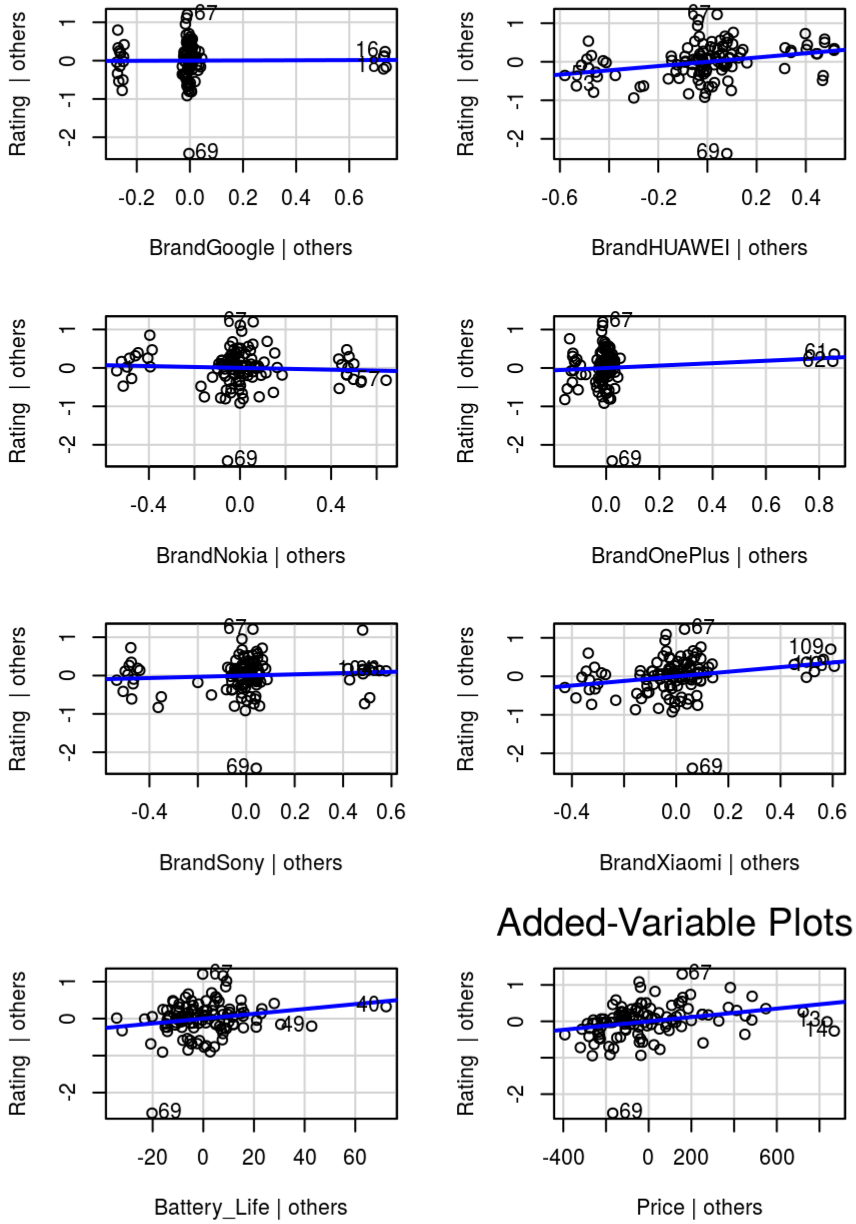
```
summary(best_subsets_method$BestModel)
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4195 -0.1868  0.0505  0.2195  1.2117
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.9401994  0.2682787  10.959  < 2e-16 ***
## BrandGoogle    0.0238807  0.2486717   0.096  0.92368
## BrandHUAWEI    0.5513903  0.1887013   2.922  0.00426 **
## BrandMotorola  0.0784410  0.1908052   0.411  0.68183
## BrandNokia    -0.1205536  0.1962303  -0.614  0.54031
## BrandOnePlus   0.3162196  0.3641330   0.868  0.38715
## BrandSamsung  -0.0198249  0.1651382  -0.120  0.90467
## BrandSony      0.1585503  0.1868821   0.848  0.39815
## BrandXiaomi    0.5963871  0.2296534   2.597  0.01076 *
## Price          0.0005780  0.0001863   3.103  0.00246 **
## Battery_Life   0.0065542  0.0031798   2.061  0.04176 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4766 on 105 degrees of freedom
## Multiple R-squared:  0.3399, Adjusted R-squared:  0.277
## F-statistic: 5.406 on 10 and 105 DF,  p-value: 2.002e-06
```

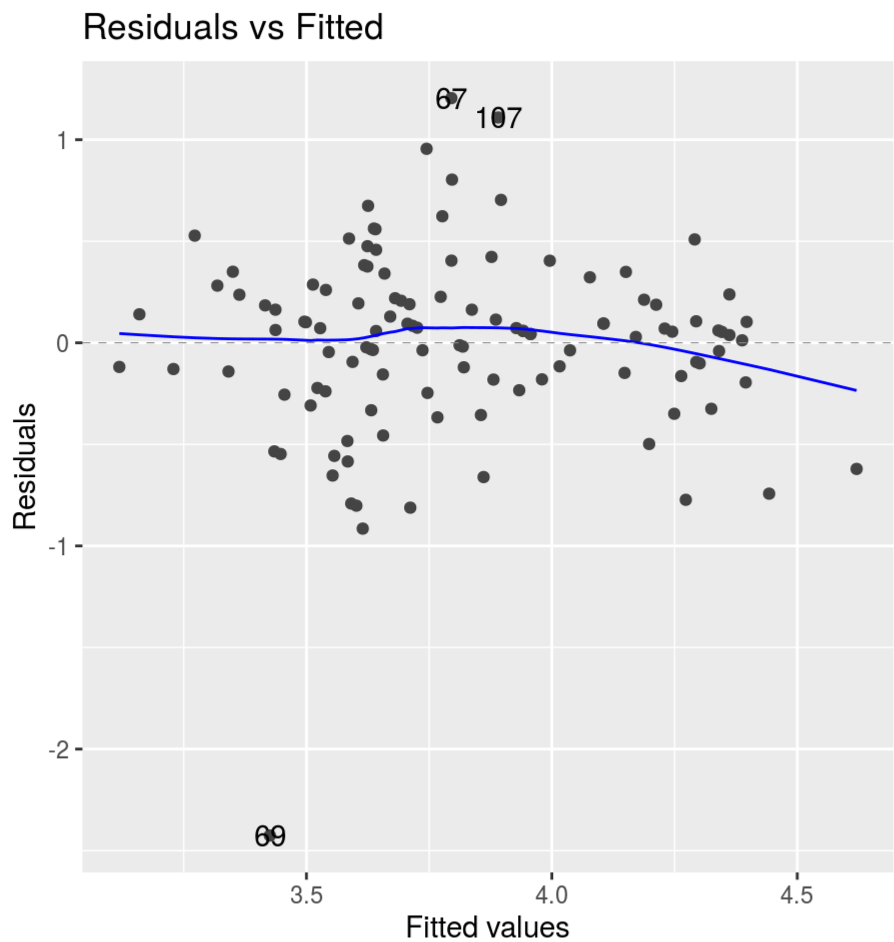
```
#Our variable selection method suggests that we drop Total_Reviews and Brand when using the BIC
 method, but only Total_Reviews with the AIC method. We will proceed to check our other assumpti
ons before making further decisions.

phone_price$Rating_Residuals <- phone_rating_lm$residuals

phone_price$Rating_Fitted <- phone_rating_lm$fitted.values
```

#assumption checking, remedial measures, exploring if interaction terms or higher-order variables are needed, using variable selection methods

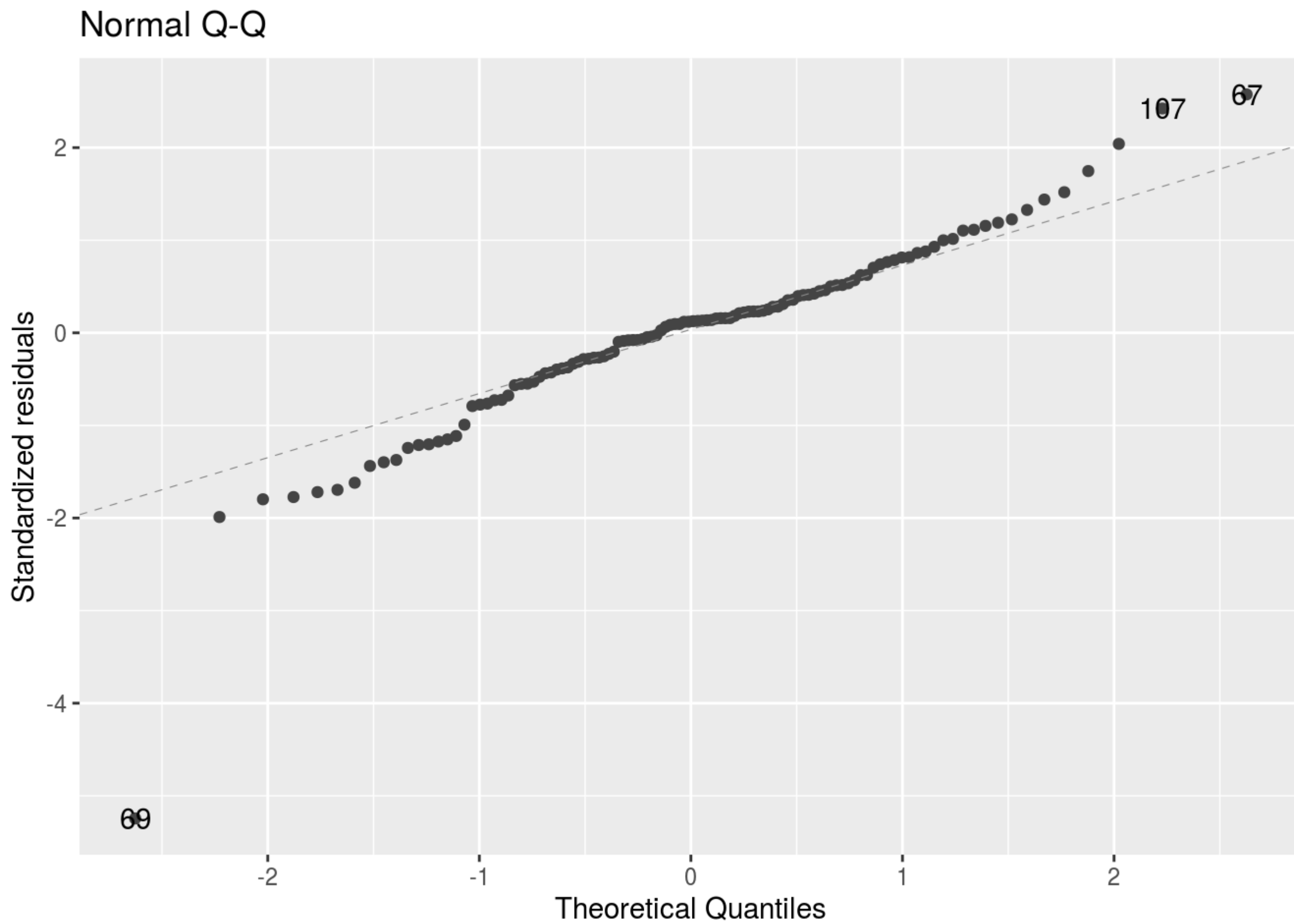
```
# 1. Linearity of X's vs Y.

# Partial Regression Plot
avPlots(phone_rating_lm)
```


Added-Variable Plots
Rating | others
BrandGoogle | others
BrandHUAWEI | others
BrandMotorola | others
BrandNokia | others
BrandOnePlus | others
BrandSamsung | others
BrandSony | others
BrandXiaomi | others
Total_Reviews | others
Battery_Life | others
Price | others

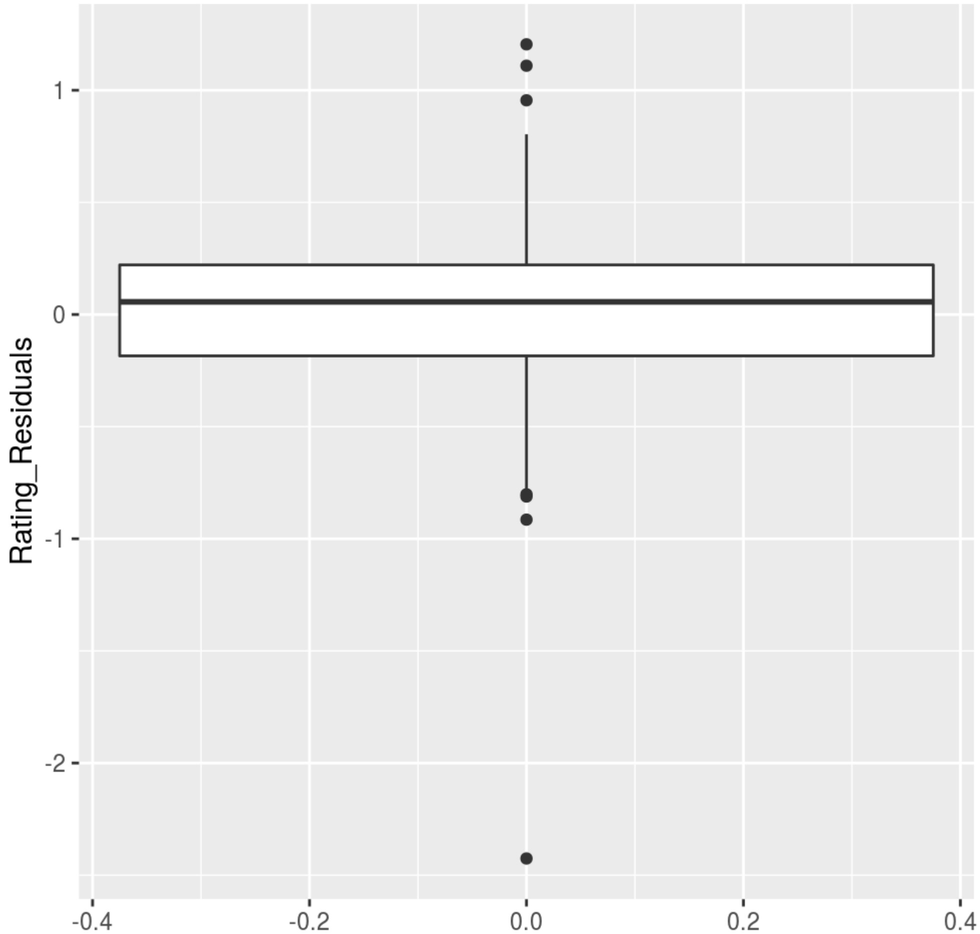
```
# RvF Plot
(phone_rating_RvF_plot <- autoplot(phone_rating_lm, which = 1, ncol = 1, nrow = 1) +
  theme(aspect.ratio = 1))
```



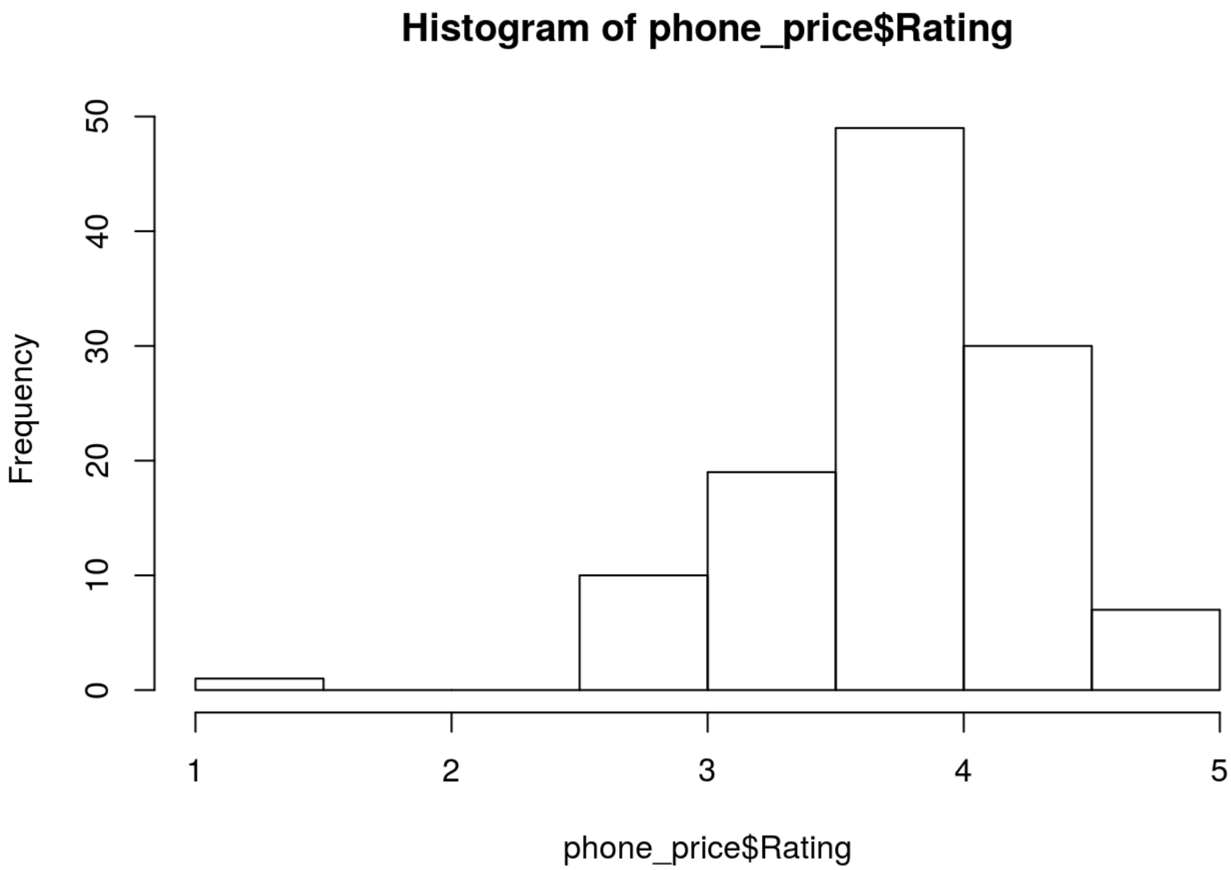

```
#The partial regression plots look linear, the scatterplots for rating look roughly linear, and
 the Residuals vs Fitted Values Plot looks linear. This assumption is passed.

# 2. Independence - Our dataset is all of the the reviews for these phones on Amazon. There may
 be a problem with independence due to the nature of reviews. Since the reviews were not written
all at the same time it is likely that many reviews are dependant on the reviews of others. This
assumption is unclear.

# 3. Normality
# QQ Normal plot
(phone_rating_normprob_plot <- autoplot(phone_rating_lm , which = 2, ncol = 1, nrow = 1) )
```




```
# Residual Boxplot
(phone_rating_boxplot <- ggplot(data = phone_price, mapping = aes( y = Rating_Residuals))+
  geom_boxplot()+
  theme(aspect.ratio = 1))
```



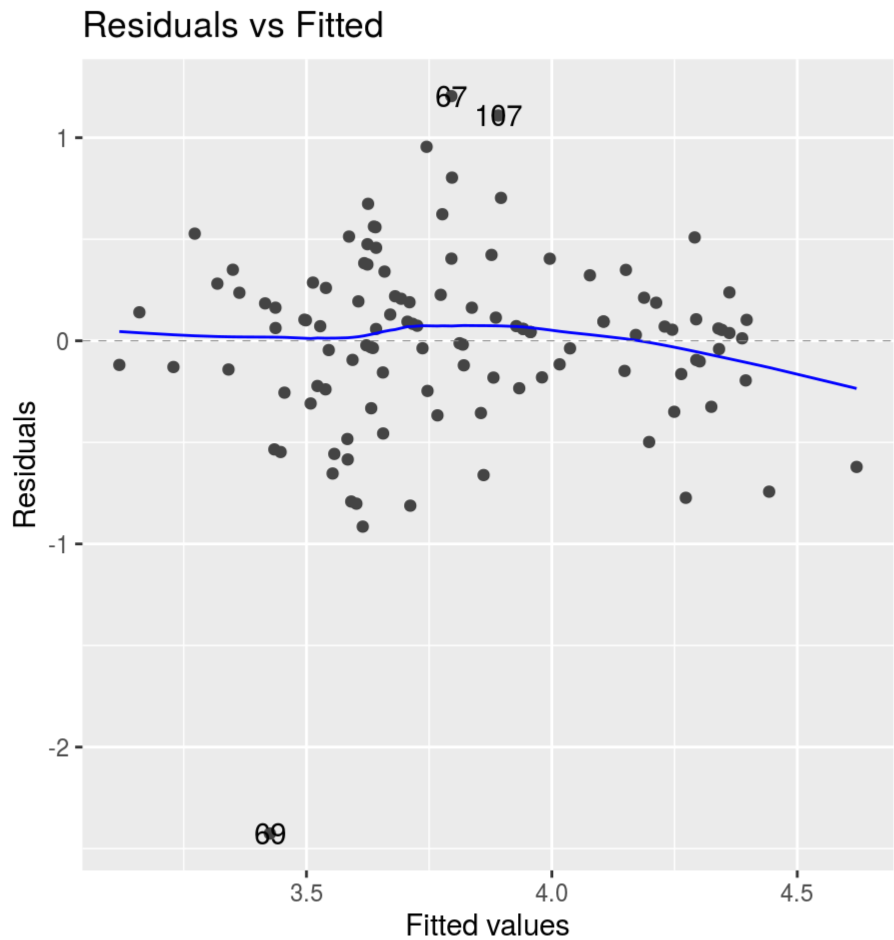

```
hist(phone_price$Rating)
```



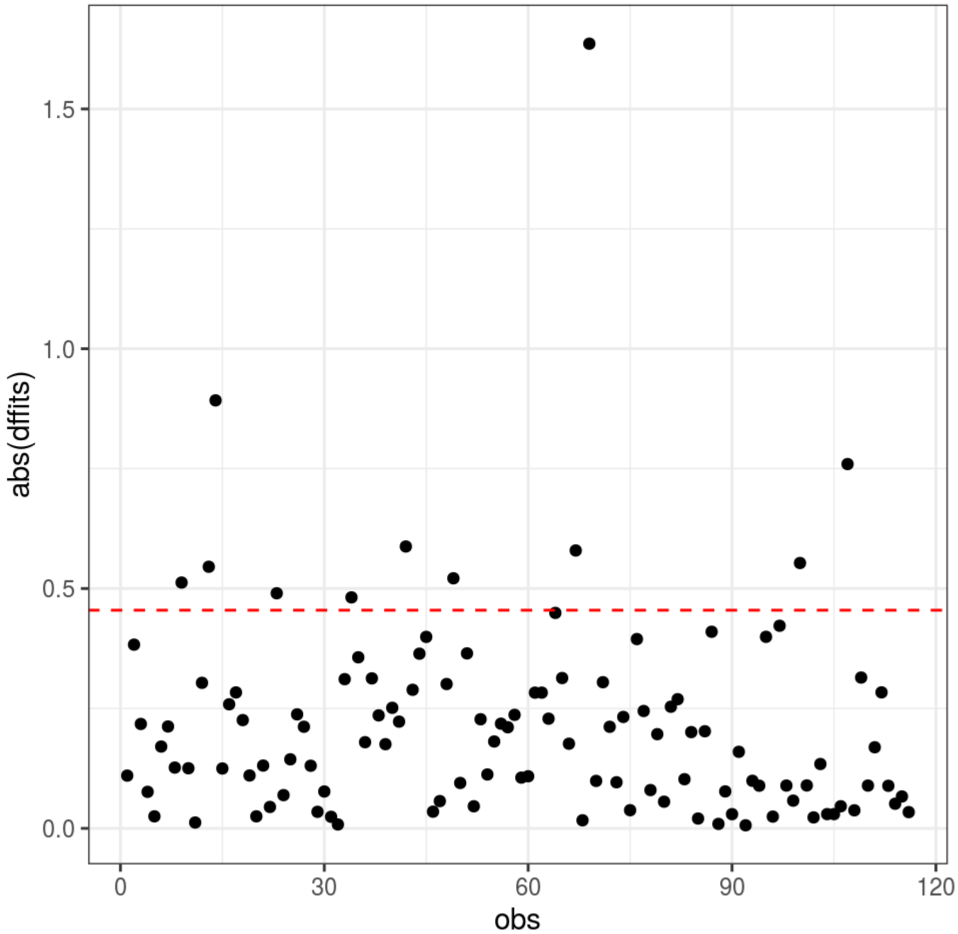

```
# The Normal Probability Plot is roughly following the dotted line and indicates normality. The
 boxplot of the residuals looks normally distributed.This may not be passed.

# 4. Homoscedasticity
#Brown-Forsythe Test
grp <- as.factor(c(rep("lower", floor(dim(phone_price)[1] / 2)),
                   rep("upper", ceiling(dim(phone_price)[1] / 2))))
leveneTest(phone_price$Rating_Residuals ~ grp, center = median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1    0.35 0.5553
##       114
```

```
#RvF Plot
phone_rating_RvF_plot
```




```
# The Brown-Forsythe Test has a null hypothesis that the variance is constant and it shows that
 we do not have sufficient evidence to reject our null hypothesis.The RvF Plot appears to have r
oughly equal spread above and below the horizontal line. This assumption is passed.

# 5. Describes all observations (no influential points)
# DFFITS
phone_rating_dffits <- data.frame("dffits" = dffits(phone_rating_lm))
phone_rating_dffits$obs <- 1:length(phone_price$Rating)

ggplot(data = phone_rating_dffits) +
  geom_point(mapping = aes(x = obs, y = abs(dffits))) +
  geom_hline(mapping = aes(yintercept = 2 * sqrt(6 / length(obs))),
             color = "red", linetype = "dashed") +  # for n > 30
  theme_bw() +
  theme(aspect.ratio = 1)
```




```
phone_rating_dffits[abs(phone_rating_dffits$dffits) > 2 * sqrt(6/length(phone_price$Rating)), ]
```

```
##         dffits obs
## 9    0.5124763   9
## 13  -0.5454231  13
## 14  -0.8923283  14
## 23  -0.4901419  23
## 34  -0.4815902  34
## 42  -0.5876746  42
## 49  -0.5213649  49
## 67   0.5793633  67
## 69  -1.6360318  69
## 100 -0.5532132 100
## 107  0.7595144 107
```

```
#Cook's Distance
phone_price$cooksd <- cooks.distance(phone_rating_lm)
phone_price[phone_price$cooksd >= 4 / length(phone_price$cooksd), ]
```

```
##                   Phone   Brand Rating Total_Reviews Battery_Life   Price
## 14  Apple iPhone XS Max   Apple    3.5            53           79 1399.99
## 69    Samsung Galaxy J2 Samsung    1.0             1           67  103.74
## 107     Sony Xperia XZ2    Sony    5.0             1           88  364.85
##     Rating_Residuals Rating_Fitted     cooksd
## 14        -0.7729067      4.272907 0.06491583
## 69        -2.4255171      3.425517 0.16564667
## 107        1.1093241      3.890676 0.04579955
```

```
# Both our DFFITS and Cook's Distance diagnostics indicate that we have several potential influe
ntial points. It appears that some of these points only had one person rating the phone which ma
y mean that we simply do not have enough information about those particular phones to use in our
regression model. This assumption is likely not passed.

# 6 & 7. Additional predictor variables are unnecessary and No multicolinearity
# Check significance levels of factors
summary(phone_rating_lm)
```

```
##
## Call:
## lm(formula = Rating ~ Brand + Total_Reviews + Battery_Life +
##     Price, data = phone_price)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.42552 -0.18455  0.05635  0.22114  1.20466
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.944e+00  2.710e-01  10.863  < 2e-16 ***
## BrandGoogle    2.224e-02  2.502e-01   0.089  0.92935
## BrandHUAWEI    5.536e-01  1.904e-01   2.908  0.00445 **
## BrandMotorola  8.104e-02  1.928e-01   0.420  0.67509
## BrandNokia    -1.181e-01  1.981e-01  -0.596  0.55223
## BrandOnePlus   3.136e-01  3.664e-01   0.856  0.39402
## BrandSamsung  -1.594e-02  1.687e-01  -0.095  0.92489
## BrandSony      1.601e-01  1.882e-01   0.851  0.39682
## BrandXiaomi    5.986e-01  2.314e-01   2.587  0.01106 *
## Total_Reviews -3.409e-05  2.682e-04  -0.127  0.89909
## Battery_Life   6.527e-03  3.202e-03   2.039  0.04403 *
## Price          5.824e-04  1.902e-04   3.062  0.00280 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4789 on 104 degrees of freedom
## Multiple R-squared:   0.34,  Adjusted R-squared:  0.2702
## F-statistic:  4.87 on 11 and 104 DF,  p-value: 4.858e-06
```

```
# Check VIF
vif(phone_rating_lm)
```

```
##                   GVIF Df GVIF^(1/(2*Df))
## Brand         1.735534  8        1.035058
## Total_Reviews 1.083350  1        1.040841
## Battery_Life  1.373231  1        1.171850
## Price         1.261479  1        1.123156
```

```
mean(vif(phone_rating_lm))
```

```
## [1] 1.735375
```

```
# Check correlation matrix
cor(phone_price[,3:6])
```

```
##                    Rating Total_Reviews Battery_Life     Price
## Rating        1.000000000   0.001039107  0.337767327 0.2902744
## Total_Reviews 0.001039107   1.000000000  0.008857901 0.1187502
## Battery_Life  0.337767327   0.008857901  1.000000000 0.1074344
## Price         0.290274415   0.118750242  0.107434431 1.0000000
```

```
# Since our VIF's are all close to 1 and our correlation matrix does not show any predictors tha
t are extremely correlated to eachother the no multicolinearity assumption is passed. It appears
that many of our predictor variables are significantly affecting the response, however, we may b
e better off dropping Total_Reviews since it does not seems to have a significant effect on Rati
ng.
```

Part 2.2 LM

```
#Daniel Ironhat

# We will be deleting points with only 1 Total_Reviews from our dataset and predictor variable T
otal_Reviews to better meet the regression assumptions.


# New model
phone_price2 <- phone_price[-107,]
phone_price2 <- phone_price2[-69,]
phone_price2 <- phone_price2[-67,]
phone_price2 <- phone_price2[-35,]
phone_price2 <- phone_price2[-20,]

phone_price2 <- phone_price2[ , c("Brand","Price", "Total_Reviews", "Battery_Life", "Rating")]
#Check variable selection
best_subsets_method <- bestglm(phone_price2,
                            IC = "BIC",
                            method = "exhaustive",
                            TopModels = 10)
```

```
## Morgan-Tatar search since factors present with more than 2 levels.
```

```
summary(best_subsets_method$BestModel)
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.94118 -0.18190  0.03012  0.21074  0.97626
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.4500623  0.1298261  26.574  < 2e-16 ***
## BrandGoogle    0.0432150  0.1997475   0.216  0.82915
## BrandHUAWEI    0.6583827  0.1465131   4.494 1.87e-05 ***
## BrandMotorola  0.1217196  0.1552362   0.784  0.43482
## BrandNokia    -0.1665661  0.1579649  -1.054  0.29419
## BrandOnePlus   0.4303014  0.2902053   1.483  0.14125
## BrandSamsung   0.1214120  0.1256192   0.967  0.33610
## BrandSony      0.1093390  0.1514894   0.722  0.47211
## BrandXiaomi    0.7236969  0.1740377   4.158 6.74e-05 ***
## Price          0.0004992  0.0001507   3.313  0.00128 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3833 on 101 degrees of freedom
## Multiple R-squared:  0.3994, Adjusted R-squared:  0.3459
## F-statistic: 7.463 on 9 and 101 DF,  p-value: 2.696e-08
```

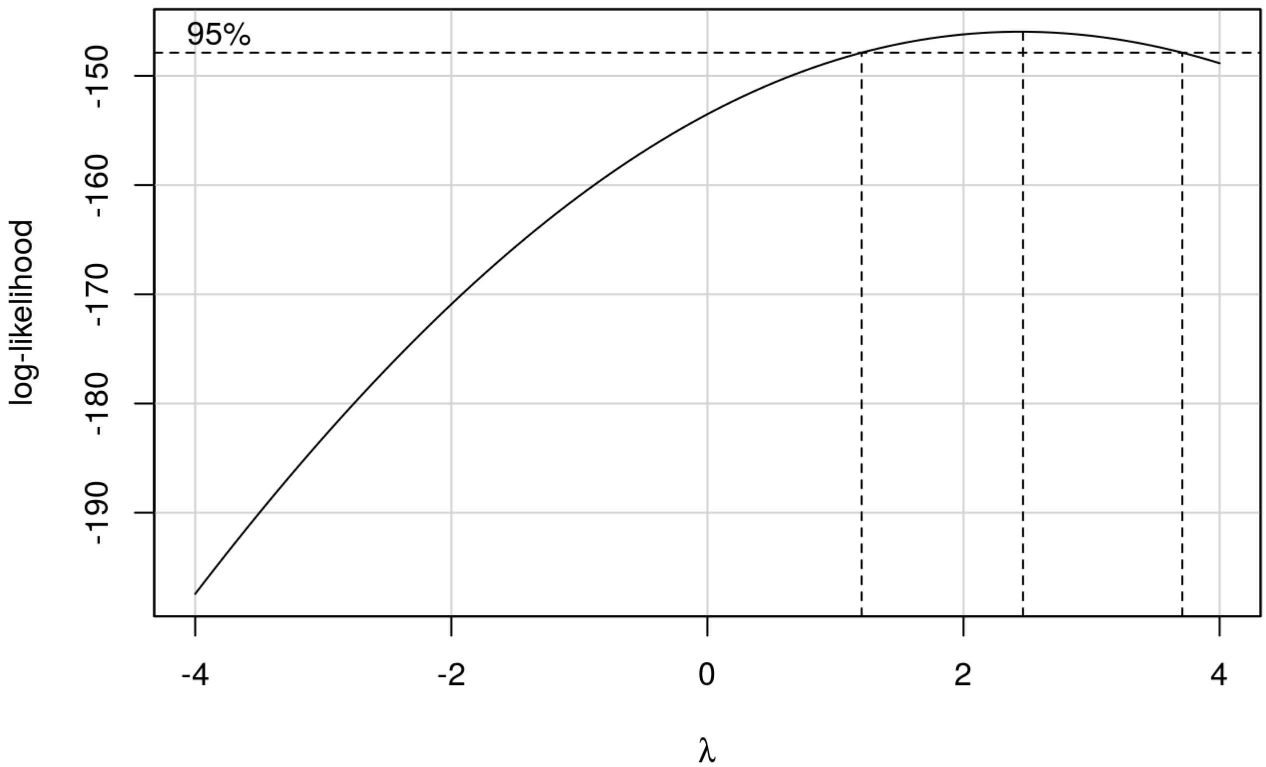
```
best_subsets_method <- bestglm(phone_price2,
                            IC = "AIC",
                            method = "exhaustive",
                            TopModels = 10)
```

```
## Morgan-Tatar search since factors present with more than 2 levels.
```

```
summary(best_subsets_method$BestModel)
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95237 -0.17824  0.04058  0.19959  0.94130
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.1990803  0.2183387  14.652  < 2e-16 ***
## BrandGoogle    0.0300446  0.1989486   0.151 0.880266
## BrandHUAWEI    0.5900343  0.1534498   3.845 0.000212 ***
## BrandMotorola  0.1001450  0.1551880   0.645 0.520201
## BrandNokia    -0.1614652  0.1572041  -1.027 0.306849
## BrandOnePlus   0.3736521  0.2914538   1.282 0.202797
## BrandSamsung   0.0537255  0.1336944   0.402 0.688650
## BrandSony      0.0784304  0.1522718   0.515 0.607642
## BrandXiaomi    0.6356146  0.1838454   3.457 0.000803 ***
## Price          0.0004772  0.0001507   3.166 0.002052 **
## Battery_Life   0.0036772  0.0025791   1.426 0.157049
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3813 on 100 degrees of freedom
## Multiple R-squared:  0.4114, Adjusted R-squared:  0.3525
## F-statistic: 6.988 on 10 and 100 DF,  p-value: 3.157e-08
```

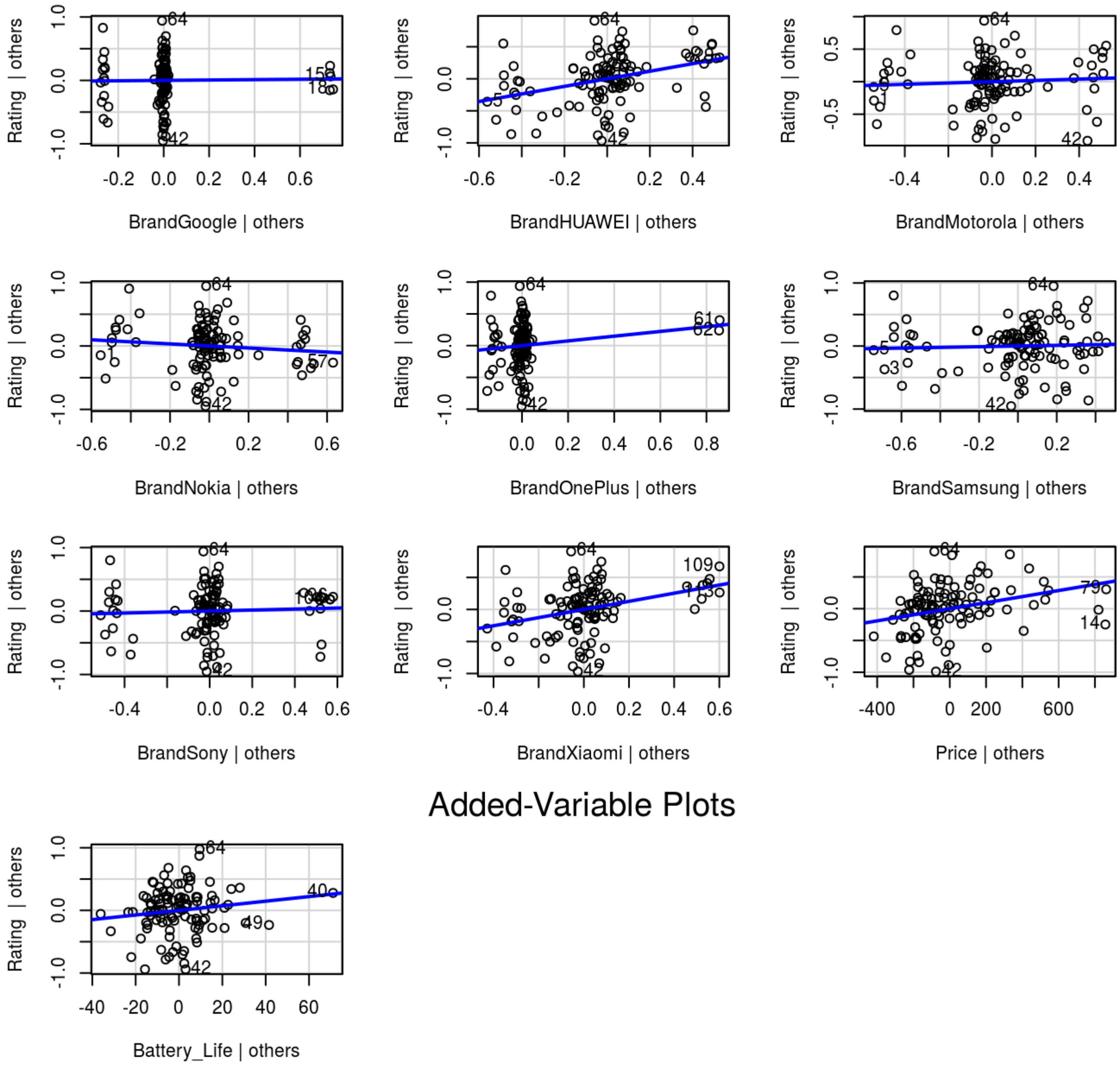
```
# We will go with the AIC method to have the better predictive power rather than interprettabili
ty since our model struggles with R^2 and adjusted R^2.


phone_rating_lm2 <- lm(data = phone_price2, formula = Rating ~ Brand + Price + Battery_Life)

boxCox(phone_rating_lm2,lambda = seq(-4, 4, 1/10))
```



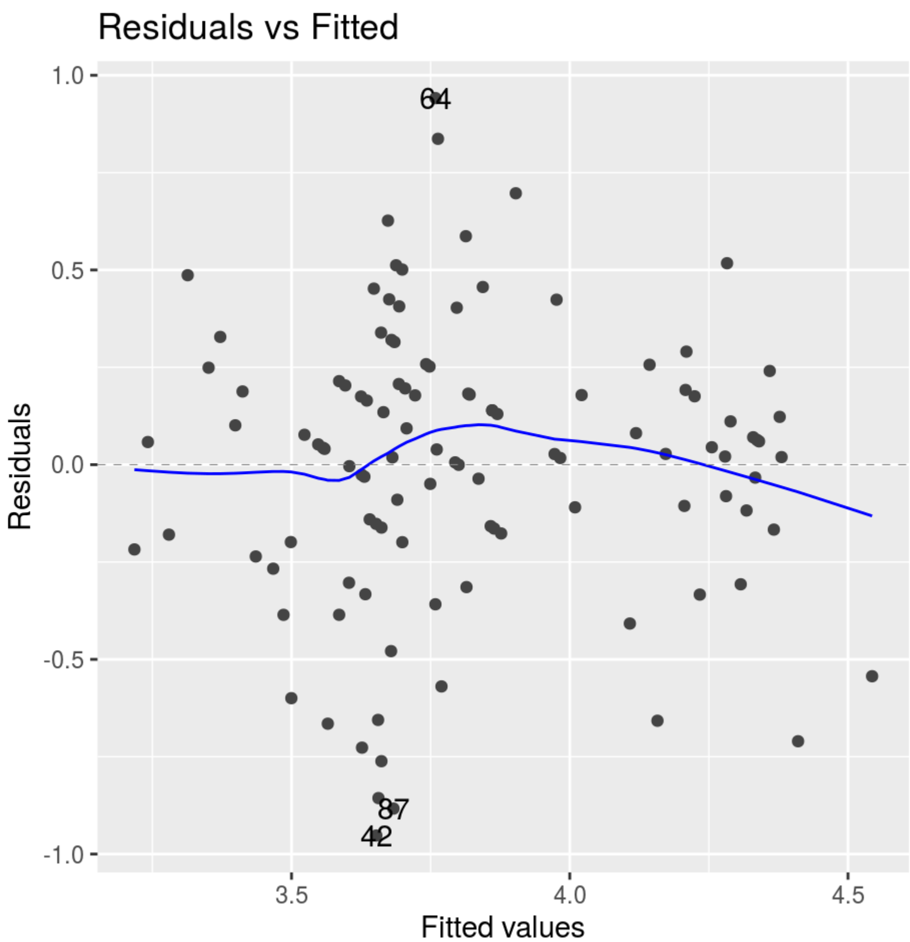

```
#Try y transform of y^2
phone_rating_ytrans_lm <- lm(data = phone_price2, formula = Rating^2 ~ Brand + Price + Battery_L
ife)

phone_price2$Residuals <- phone_rating_lm2$residuals
phone_price2$Fitted <- phone_rating_lm2$fitted.values

phone_price3 <- phone_price2
phone_price3$Rating <- phone_price3$Rating^2
phone_price3$Residuals <- phone_rating_ytrans_lm$residuals
phone_price3$Fitted <- phone_rating_ytrans_lm$fitted.values
```

Part 2.2 Assumptions

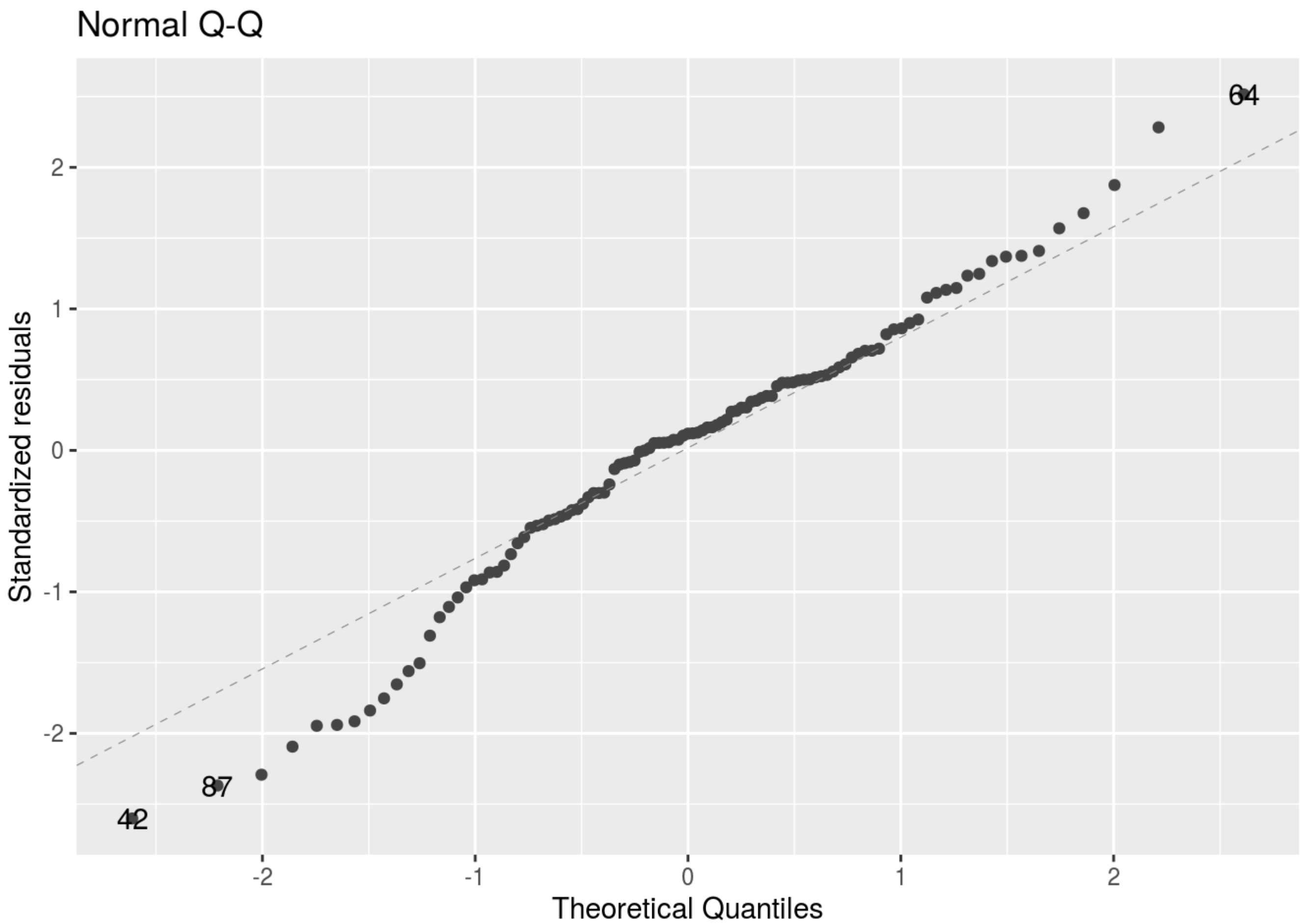
```
# 1. Linearity of X's vs Y.

# Partial Regression Plot
avPlots(phone_rating_lm2)
```


Added-Variable Plots
Rating | others
BrandGoogle | others
BrandHUAWEI | others
BrandMotorola | others
BrandNokia | others
BrandOnePlus | others
BrandSamsung | others
BrandSony | others
BrandXiaomi | others
Price | others
Battery_Life | others

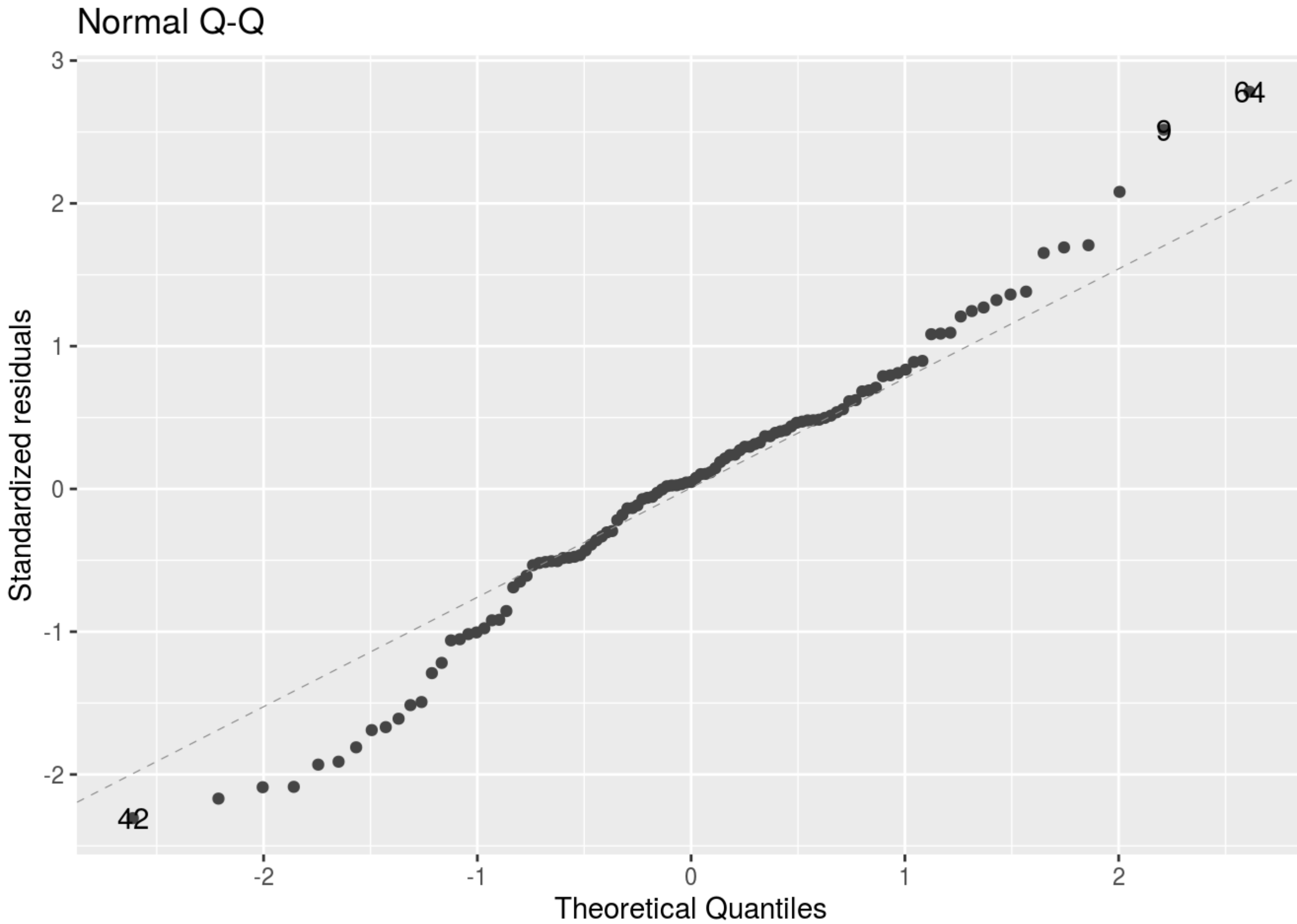
```
# RvF Plot
(phone_rating_RvF_plot <- autoplot(phone_rating_lm2, which = 1, ncol = 1, nrow = 1) +
  theme(aspect.ratio = 1))
```



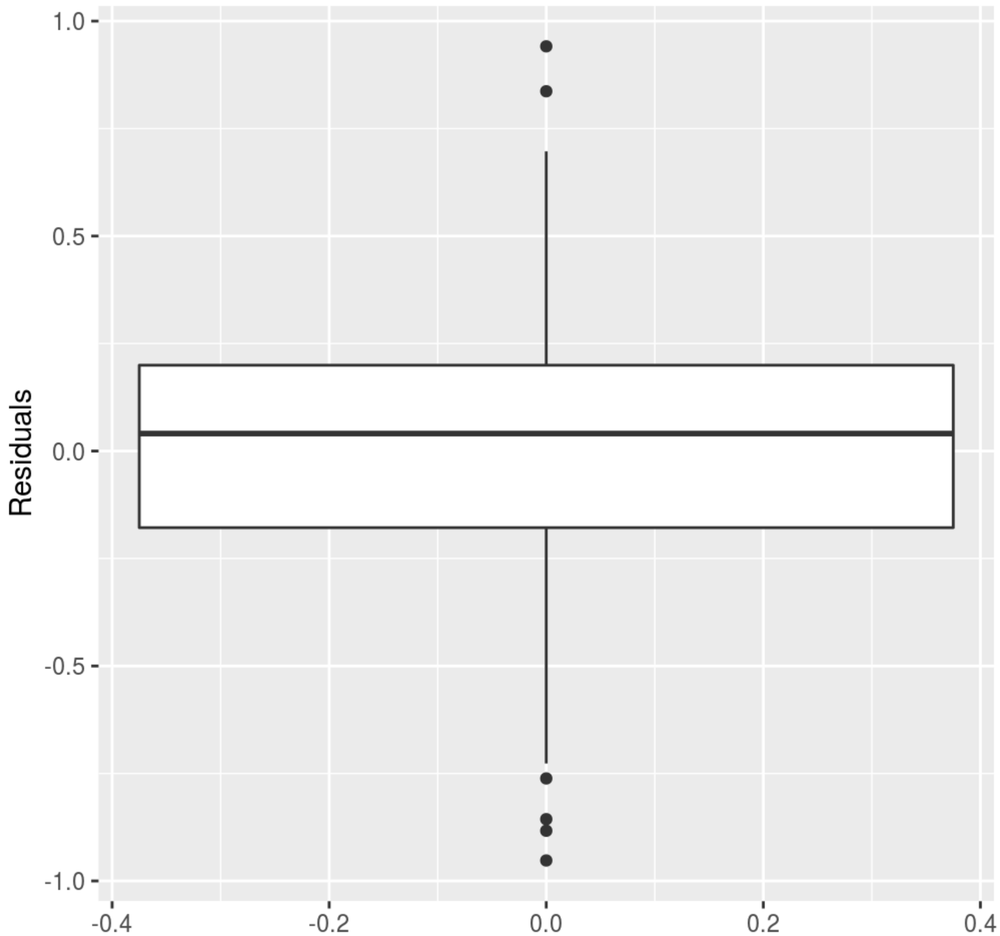

```
#The partial regression plots look linear, the scatterplots for rating look roughly linear, and
 the Residuals vs Fitted Values Plot looks linear. This assumption is passed.

# 3. Normality
# QQ Normal plot
(phone_rating_normprob_plot <- autoplot(phone_rating_lm2 , which = 2, ncol = 1, nrow = 1) )
```



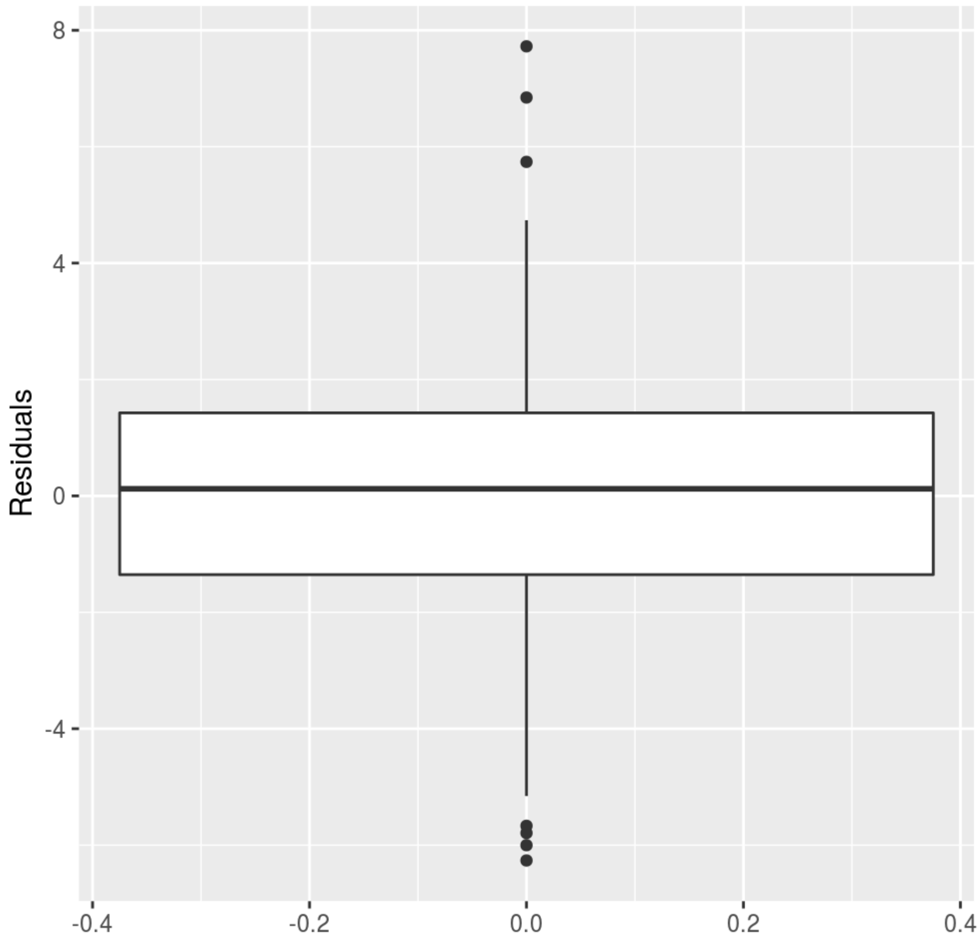

```
(phone_rating_normprob_plot <- autoplot(phone_rating_ytrans_lm , which = 2, ncol = 1, nrow = 1)
 )
```




```
# Residual Boxplot
(phone_rating_boxplot <- ggplot(data = phone_price2, mapping = aes( y = Residuals))+
  geom_boxplot()+
  theme(aspect.ratio = 1))
```




```
(phone_rating_boxplot <- ggplot(data = phone_price3, mapping = aes( y = Residuals))+
  geom_boxplot()+
  theme(aspect.ratio = 1))
```



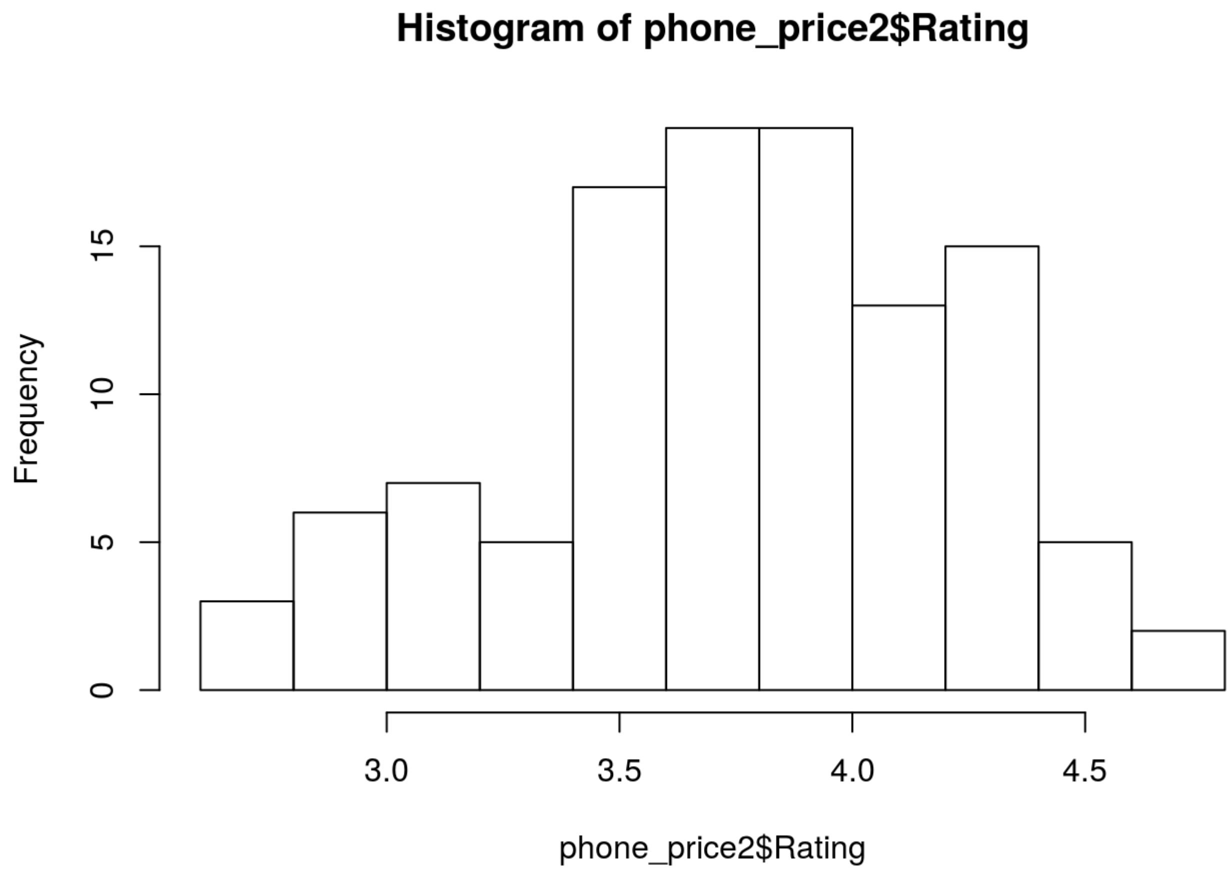

```
#Shapiro-Wilk Test
shapiro.test(phone_rating_lm2$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  phone_rating_lm2$residuals
## W = 0.97831, p-value = 0.06743
```

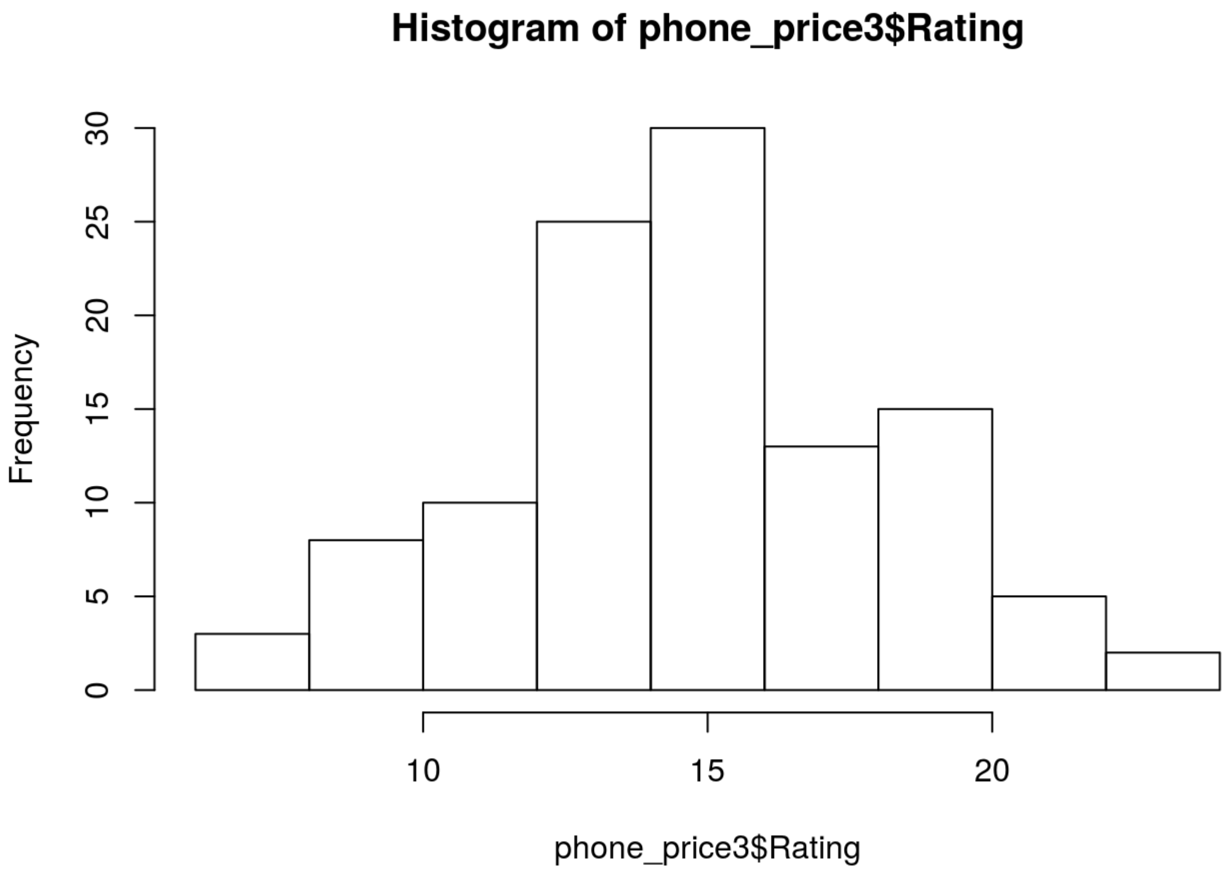
```
shapiro.test(phone_rating_ytrans_lm$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  phone_rating_ytrans_lm$residuals
## W = 0.98332, p-value = 0.1818
```

```
#Histogram of Response
hist(phone_price2$Rating)
```



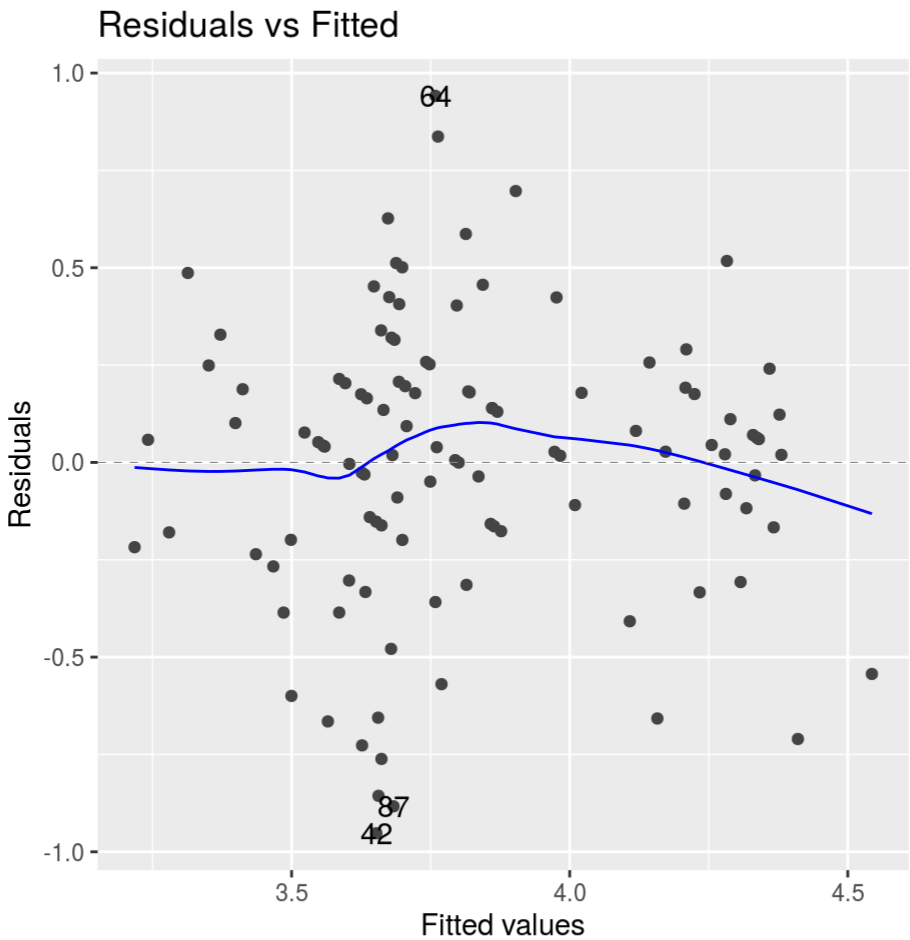

```
hist(phone_price3$Rating)
```



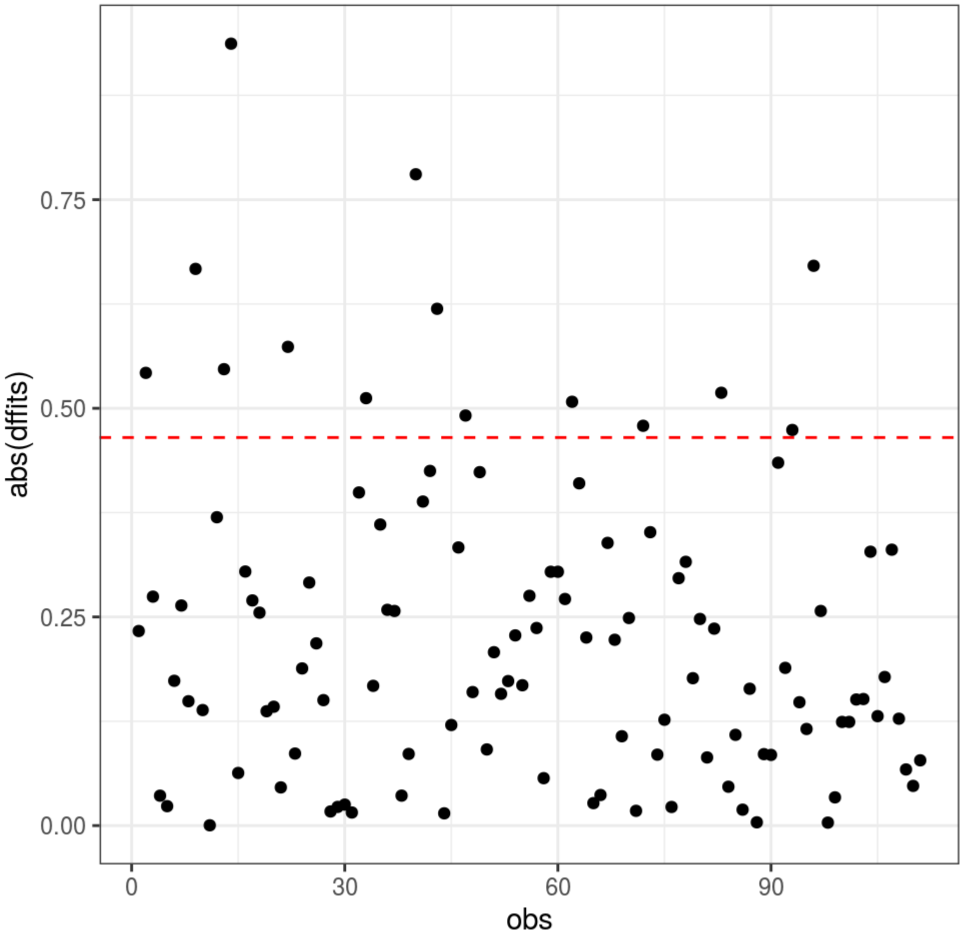

```
# The Normal Probability Plot is roughly following the dotted line and indicates normality and t
he Normal Probability Plot of the y transformed data is not much better. The boxplot of the resi
duals looks normally distributed, but it is not quite centered on zero. After the Y transform th
e boxplot of the residuals is still looking normal and is centered on zero, but there is not a b
ig difference. The Shapiro_wilk test is above 0.05 for both datasets, but the Y-transformed data
shows a noticable improvement for this test. Our histograms of Y and Y-transformed both look rou
ghly normal, but the Y-transformed is noticably better. Transforming the data may help to better
achieve this assumption, but we were borderline passing without the transformations so we will s
tick to the untransformed data so as to not complicate the interpretation of our model.



# 4. Homoscedasticity
#Brown-Forsythe Test
grp <- as.factor(c(rep("lower", floor(dim(phone_price2)[1] / 2)),
                   rep("upper", ceiling(dim(phone_price2)[1] / 2))))
leveneTest(phone_price2$Residuals ~ grp, center = median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1  0.0015 0.9695
##       109
```

```
#RvF Plot
phone_rating_RvF_plot
```




```
# 5. Describes all observations (no influential points)
# DFFITS
phone_rating_dffits <- data.frame("dffits" = dffits(phone_rating_lm2))
phone_rating_dffits$obs <- 1:length(phone_price2$Rating)

ggplot(data = phone_rating_dffits) +
  geom_point(mapping = aes(x = obs, y = abs(dffits))) +
  geom_hline(mapping = aes(yintercept = 2 * sqrt(6 / length(obs))),
             color = "red", linetype = "dashed") +  # for n > 30
  theme_bw() +
  theme(aspect.ratio = 1)
```




```
phone_rating_dffits[abs(phone_rating_dffits$dffits) > 2 * sqrt(6/length(phone_price2$Rating)), ]
```

```
##         dffits obs
## 2   -0.5424611   2
## 9    0.6670518   9
## 13  -0.5468790  13
## 14  -0.9369054  14
## 23  -0.5736323  22
## 34  -0.5121184  33
## 42  -0.7804006  40
## 45  -0.6192470  43
## 49  -0.4913344  47
## 64   0.5078250  62
## 76  -0.4790598  72
## 87  -0.5186177  83
## 97  -0.4740186  93
## 100 -0.6707660  96
```

```
#Cook's Distance
phone_price2$cooksd <- cooks.distance(phone_rating_lm2)
phone_price2[phone_price2$cooksd >= 4 / length(phone_price2$cooksd), ]
```

```
##        Brand   Price Total_Reviews Battery_Life Rating  Residuals   Fitted
## 9      Apple  557.99            25           81    4.6  0.8368046 3.763195
## 14     Apple 1399.99            53           79    3.5 -0.6576320 4.157632
## 42  Motorola  139.00             4           78    2.7 -0.9523726 3.652373
## 100     Sony  249.99           228           72    2.9 -0.7615579 3.661558
##         cooksd
## 9   0.03873073
## 14  0.07764946
## 42  0.05214100
## 100 0.03950435
```

```
# 6 & 7. Additional predictor variables are unnecessary and No multicolinearity
# Check significance levels of factors
summary(phone_rating_lm2)
```

```
##
## Call:
## lm(formula = Rating ~ Brand + Price + Battery_Life, data = phone_price2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95237 -0.17824  0.04058  0.19959  0.94130
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.1990803  0.2183387  14.652  < 2e-16 ***
## BrandGoogle    0.0300446  0.1989486   0.151 0.880266
## BrandHUAWEI    0.5900343  0.1534498   3.845 0.000212 ***
## BrandMotorola  0.1001450  0.1551880   0.645 0.520201
## BrandNokia    -0.1614652  0.1572041  -1.027 0.306849
## BrandOnePlus   0.3736521  0.2914538   1.282 0.202797
## BrandSamsung   0.0537255  0.1336944   0.402 0.688650
## BrandSony      0.0784304  0.1522718   0.515 0.607642
## BrandXiaomi    0.6356146  0.1838454   3.457 0.000803 ***
## Price          0.0004772  0.0001507   3.166 0.002052 **
## Battery_Life   0.0036772  0.0025791   1.426 0.157049
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3813 on 100 degrees of freedom
## Multiple R-squared:  0.4114, Adjusted R-squared:  0.3525
## F-statistic: 6.988 on 10 and 100 DF,  p-value: 3.157e-08
```

```
# Check VIF
vif(phone_rating_lm2)
```

```
##                  GVIF Df GVIF^(1/(2*Df))
## Brand        1.648642  8        1.031740
## Price        1.214909  1        1.102229
## Battery_Life 1.382929  1        1.175980
```

```
mean(vif(phone_rating_lm2))
```

```
## [1] 1.950714
```

```
# Check correlation matrix
cor(phone_price2[,c(2,4,5)])
```

```
##                   Price Battery_Life    Rating
## Price        1.00000000   0.09111049 0.2830117
## Battery_Life 0.09111049   1.00000000 0.3343279
## Rating       0.28301173   0.33432787 1.0000000
```

```
#END DANIEL IRONHAT
```

```
##BEGIN PART 3
#statistical inference (confidence intervals and hypothesis tests for the
#slope(s), confidence and prediction intervals for the predicted values of the
#response, etc.)

#Confidence Interval for Slope
yhatconf <- confint(phone_rating_lm2)
yhatconf
```

```
##                       2.5 %       97.5 %
## (Intercept)    2.7659025694 3.6322579530
## BrandGoogle   -0.3646636690 0.4247529482
## BrandHUAWEI    0.2855942869 0.8944742532
## BrandMotorola -0.2077434932 0.4080335820
## BrandNokia    -0.4733536026 0.1504231204
## BrandOnePlus  -0.2045838777 0.9518881273
## BrandSamsung  -0.2115204745 0.3189714113
## BrandSony     -0.2236725312 0.3805332402
## BrandXiaomi    0.2708704399 1.0003586622
## Price          0.0001781155 0.0007762573
## Battery_Life  -0.0014396464 0.0087939643
```

```
#Our company is releasing a new zphone, the zphone 11. The phone reportably cost 699.00 and a ba
ttery life of 71.5 hrs, the same specs as the new iphone 11; however, knowing that brand is a bi
g influence on rating, the company knows their products are more in line with those of Huawei. W
hat is the predicted value for their phone's rating?
#Confidence Interval for predicted values
newdata <- data.frame(Price = 699, Brand = "HUAWEI", Battery_Life = 71.5 )
value <- predict(phone_rating_lm2, newdata, se.fit = TRUE)
# compute the margin of error
me <- qnorm(0.975) * value$se.fit

# compute the 95% confidence interval and point estimate
ci <- c(value$fit-me, value$fit, value$fit+me)
names(ci) <- c("Lower Bound", "Point Estimate", "Upper Bound")
ci
```

```
##    Lower Bound Point Estimate    Upper Bound
##       4.144265       4.385585       4.626904
```

```
#We are 95% confident that the true rating for a phone with a battery life of 71.5, Price of 699
and being a similar brand to Huawei will be between 4.14 and 4.63

#add hypothesis tests
summary(phone_rating_lm2)
```

```
##
## Call:
## lm(formula = Rating ~ Brand + Price + Battery_Life, data = phone_price2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95237 -0.17824  0.04058  0.19959  0.94130
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.1990803  0.2183387  14.652  < 2e-16 ***
## BrandGoogle     0.0300446  0.1989486   0.151 0.880266
## BrandHUAWEI     0.5900343  0.1534498   3.845 0.000212 ***
## BrandMotorola   0.1001450  0.1551880   0.645 0.520201
## BrandNokia     -0.1614652  0.1572041  -1.027 0.306849
## BrandOnePlus    0.3736521  0.2914538   1.282 0.202797
## BrandSamsung    0.0537255  0.1336944   0.402 0.688650
## BrandSony       0.0784304  0.1522718   0.515 0.607642
## BrandXiaomi     0.6356146  0.1838454   3.457 0.000803 ***
## Price           0.0004772  0.0001507   3.166 0.002052 **
## Battery_Life    0.0036772  0.0025791   1.426 0.157049
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3813 on 100 degrees of freedom
## Multiple R-squared:  0.4114, Adjusted R-squared:  0.3525
## F-statistic: 6.988 on 10 and 100 DF,  p-value: 3.157e-08
```

#Based off the F-statistic of 6.202 on 10 and 104 DF, which has a p-value of 2.3e-07, we can conclude that our model is significantly better at predicting rating than the null hypothesis that brand, price and battery life have no affect on the rating of a phone.

#Compare to Apple

```
newdata <- data.frame(Price = 699, Brand = "Apple", Battery_Life = 71.5 )
value <- predict(phone_rating_lm2, newdata, se.fit = TRUE)
# compute the margin of error
me <- qnorm(0.975) * value$se.fit

# compute the 95% confidence interval and point estimate
ci <- c(value$fit-me, value$fit, value$fit+me)
names(ci) <- c("Lower Bound", "Point Estimate", "Upper Bound")
ci
```

```
##    Lower Bound Point Estimate    Upper Bound
##       3.589614       3.795550       4.001487
```

#If the brand is Huawei, the rating for the zphone, with the same specs as the iphone 11, is significantly higher than that of apple. This may be a discrepency in consumer expectations, or apple ratings may be biased due to haters.

```
#BEGIN PART 4
phoneRatingMSE = get_mse(phone_rating_lm2)
phoneRatingRMSE = sqrt(phoneRatingMSE)
phoneRatingRMSE
```

```
## [1] 0.3813225
```

```
# Our RMSE is great, very close to 0. This may be due to our high degrees of freedom. We can con
clude that a much larger proportion of the variation in the residuals is captured by our model t
hat is not captured by our model.

phone_rating_with_dummies <- phone_data
phone_rating_dummies <- to.dummy(phone_rating_with_dummies$Brand, "brand")
phone_rating_with_dummies$Brand <- NULL
phone_rating_with_dummies$Phone <- NULL
phone_rating_with_dummies$Asin <- NULL
phone_rating_with_dummies <- cbind(phone_rating_with_dummies, phone_rating_dummies)
apple_lm <- lm(phone_rating_with_dummies[phone_rating_with_dummies$brand.Apple == 1, ])
google_lm <- lm(phone_rating_with_dummies[phone_rating_with_dummies$brand.Google == 1, ])
huawei_lm <- lm(phone_rating_with_dummies[phone_rating_with_dummies$brand.HUAWEI == 1, ])
moto_lm <- lm(phone_rating_with_dummies[phone_rating_with_dummies$brand.Motorola == 1, ])
nokia_lm <- lm(phone_rating_with_dummies[phone_rating_with_dummies$brand.Nokia == 1, ])
oneplus_lm <- lm(phone_rating_with_dummies[phone_rating_with_dummies$brand.OnePlus == 1, ])
samsung_lm <- lm(phone_rating_with_dummies[phone_rating_with_dummies$brand.Samsung == 1, ])
sony_lm <- lm(phone_rating_with_dummies[phone_rating_with_dummies$brand.Sony == 1, ])
xiaomi_lm <- lm(phone_rating_with_dummies[phone_rating_with_dummies$brand.Xiaomi == 1, ])

adjrSRating <- summary(phone_rating_lm2)$adj.r.squared
adjrSPhoneRatingApple = summary(apple_lm)$adj.r.squared
adjrSPhoneRatingApple
```

```
## [1] 0.04902572
```

```
adjrSPhoneRatingGoogle = summary(google_lm)$adj.r.squared
adjrSPhoneRatingGoogle
```

```
## [1] 0.3836428
```

```
adjrSPhoneRatingHuawei = summary(huawei_lm)$adj.r.squared
adjrSPhoneRatingHuawei
```

```
## [1] -0.1959062
```

```
adjrSPhoneRatingMoto = summary(moto_lm)$adj.r.squared
adjrSPhoneRatingMoto
```

```
## [1] 0.2887962
```

```
adjrSPhoneRatingNokia = summary(nokia_lm)$adj.r.squared
adjrSPhoneRatingNokia
```

```
## [1] -0.255113
```

```
adjrSPhoneRatingOnePlus = summary(oneplus_lm)$adj.r.squared
adjrSPhoneRatingOnePlus
```

```
## [1] NaN
```

```
adjrSPhoneRatingSamsung = summary(samsung_lm)$adj.r.squared
adjrSPhoneRatingSamsung
```

```
## [1] 0.2534785
```

```
adjrSPhoneRatingSony = summary(sony_lm)$adj.r.squared
adjrSPhoneRatingSony
```

```
## [1] 0.1008816
```

```
adjrSPhoneRatingXiaomi = summary(xiaomi_lm)$adj.r.squared
adjrSPhoneRatingXiaomi
```

```
## [1] -0.2659426
```

```
# Our adjusted R-Squared is 0.2701731 This is not really useful for our model as we have categor
ical data but I did it anyway. I then found the R-Squared for each phone brand individually and
 they varied but none were as close to 0 as Apple, which was 0.049. I'm not entirely sure if thi
s is actually a real thing to do but just finding the R-Squared didn't seem to make sense with c
ategorical data.

phone_rating_with_y <- phone_data
phone_rating_with_y$y <- phone_data$Rating
phone_rating_with_y$Rating <- NULL
phone_rating_with_y_dummies <- to.dummy(phone_rating_with_y$Brand, "brand")
phone_rating_with_y$Brand <- NULL
phone_rating_with_y$Phone <- NULL
phone_rating_with_y$Asin <- NULL
phone_rating_with_y_and_dummies <- cbind(phone_rating_with_y, phone_rating_with_y_dummies)

phone_rating_best_subset <- bestglm(Xy = phone_rating_with_y_and_dummies,
                            IC = "AIC",
                            method = "exhaustive",
                            TopModels = 10)

phone_rating_best_subset$BestModels
```

```
##    Total_Reviews Prices Battery_Life     y brand.Apple brand.Google
## 1           TRUE   TRUE         TRUE  TRUE        TRUE         TRUE
## 2           TRUE   TRUE        FALSE  TRUE        TRUE         TRUE
## 3          FALSE   TRUE         TRUE  TRUE        TRUE         TRUE
## 4           TRUE   TRUE         TRUE FALSE        TRUE         TRUE
## 5           TRUE   TRUE        FALSE FALSE        TRUE         TRUE
## 6          FALSE   TRUE        FALSE  TRUE        TRUE         TRUE
## 7           TRUE  FALSE         TRUE  TRUE        TRUE         TRUE
## 8          FALSE   TRUE         TRUE FALSE        TRUE         TRUE
## 9           TRUE  FALSE        FALSE  TRUE        TRUE         TRUE
## 10         FALSE  FALSE         TRUE  TRUE        TRUE         TRUE
##    brand.HUAWEI brand.Motorola brand.Nokia brand.OnePlus brand.Samsung
## 1          TRUE           TRUE        TRUE          TRUE          TRUE
## 2          TRUE           TRUE        TRUE          TRUE          TRUE
## 3          TRUE           TRUE        TRUE          TRUE          TRUE
## 4          TRUE           TRUE        TRUE          TRUE          TRUE
## 5          TRUE           TRUE        TRUE          TRUE          TRUE
## 6          TRUE           TRUE        TRUE          TRUE          TRUE
## 7          TRUE           TRUE        TRUE          TRUE          TRUE
## 8          TRUE           TRUE        TRUE          TRUE          TRUE
## 9          TRUE           TRUE        TRUE          TRUE          TRUE
## 10         TRUE           TRUE        TRUE          TRUE          TRUE
##    brand.Sony Criterion
## 1        TRUE -8327.552
## 2        TRUE -8310.966
## 3        TRUE -8300.783
## 4        TRUE -8294.324
## 5        TRUE -8273.392
## 6        TRUE -8268.664
## 7        TRUE -8252.755
## 8        TRUE -8249.287
## 9        TRUE -8247.624
## 10       TRUE -8246.967
```

```
# I didn't do AIC (or BIC or PMSE) since we found no multicollinearity and don't have a real nee
d to use and variable selection methods. But actually I did do it to double check and we did fin
d that our best model included all predictor variables

summary(phone_rating_lm2)
```

```
##
## Call:
## lm(formula = Rating ~ Brand + Price + Battery_Life, data = phone_price2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95237 -0.17824  0.04058  0.19959  0.94130
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.1990803  0.2183387  14.652  < 2e-16 ***
## BrandGoogle    0.0300446  0.1989486   0.151 0.880266
## BrandHUAWEI    0.5900343  0.1534498   3.845 0.000212 ***
## BrandMotorola  0.1001450  0.1551880   0.645 0.520201
## BrandNokia    -0.1614652  0.1572041  -1.027 0.306849
## BrandOnePlus   0.3736521  0.2914538   1.282 0.202797
## BrandSamsung   0.0537255  0.1336944   0.402 0.688650
## BrandSony      0.0784304  0.1522718   0.515 0.607642
## BrandXiaomi    0.6356146  0.1838454   3.457 0.000803 ***
## Price          0.0004772  0.0001507   3.166 0.002052 **
## Battery_Life   0.0036772  0.0025791   1.426 0.157049
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3813 on 100 degrees of freedom
## Multiple R-squared:  0.4114, Adjusted R-squared:  0.3525
## F-statistic: 6.988 on 10 and 100 DF,  p-value: 3.157e-08
```

```
# Our p-value that tests all our predictor variables at once is 2.23e-07, which shows that all o
ur model is absolutely statistically significant.

#END PART 4
```