MGT 6203 – Project 1

Deepali

5/21/2020

Question 1

Part A

```
# read in the data
airbnb = read_csv('airbnb_data.csv')
```

```
## Parsed with column specification:
## cols(
##   room_id = col_double(),
##   survey_id = col_double(),
##   host_id = col_double(),
##   room_type = col_character(),
##   city = col_character(),
##   reviews = col_double(),
##   overall_satisfaction = col_double(),
##   accommodates = col_double(),
##   bedrooms = col_double(),
##   price = col_double()
## )
```

```
# convert data frame to a tibble
airbnb = as_tibble(airbnb)
# select only the columns of interest (non id columns)
airbnb = airbnb %>% dplyr::select(room_type, city, reviews, overall_satisfaction, accommodates, bedroom

# look at the count of each city
dplyr::count(airbnb, city)
```

```
## # A tibble: 1 x 2
##   city      n
##   <chr>     <int>
## 1 Asheville 854
```

```
# it looks like all of the listings are in Asheville so we can remove this column as well
airbnb = airbnb %>% dplyr::select(-city)

# look at the different values for room type
dplyr::count(airbnb, room_type)
```

```
## # A tibble: 3 x 2
##   room_type           n
##   <chr>           <int>
## 1 Entire home/apt   512
## 2 Private room      334
## 3 Shared room         8
```

```r
# convert room_type to a factor
airbnb$room_type = as.factor(airbnb$room_type)

# view the data
head(airbnb, 5)
```

```
## # A tibble: 5 x 6
##   room_type   reviews overall_satisfaction accommodates bedrooms price
##   <fct>         <dbl>                <dbl>        <dbl>    <dbl> <dbl>
## 1 Shared room       0                  0              4        1    67
## 2 Shared room      32                  5              4        1    76
## 3 Shared room       4                  4.5            2        1    45
## 4 Shared room      24                  4.5            6        1    26
## 5 Shared room     152                  4.5            6        1    26
```

```r
# fit the multiple linear regression model with price as the response
airbnb_reg = lm(price~., data = airbnb)
summary(airbnb_reg)
```

```
##
## Call:
## lm(formula = price ~ ., data = airbnb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -367.8   -49.2     3.2    38.6  4032.7
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -23.36172   21.88618  -1.067  0.28609
## room_typePrivate room   -0.93115   13.21827  -0.070  0.94386
## room_typeShared room   -76.66780   59.90939  -1.280  0.20099
## reviews                  0.01090    0.09982   0.109  0.91310
## overall_satisfaction   -10.48160    3.47320  -3.018  0.00262 **
## accommodates            23.00721    5.23952   4.391 1.27e-05 ***
## bedrooms                85.64533   11.45983   7.474 1.95e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 167.1 on 847 degrees of freedom
## Multiple R-squared:  0.3228, Adjusted R-squared:  0.318
## F-statistic:  67.3 on 6 and 847 DF,  p-value: < 2.2e-16
```

The variables that are statistically significant are overall_satisfaction, accomodates, and bedrooms. These variables have a p-value below the 0.05 threshold, therefore they are significant in determining the price.

Part B

The coefficient for the predictor room_type(Shared room) means that the average price for a shared room is about $76.67 lower than the base case of an entire home/apt (assuming all other values held constant).

The coefficient for the predictor bedrooms means that for each additional bedroom in a listing, the price of the listing increases by about $85.65 on average (assuming all other values are held constant).

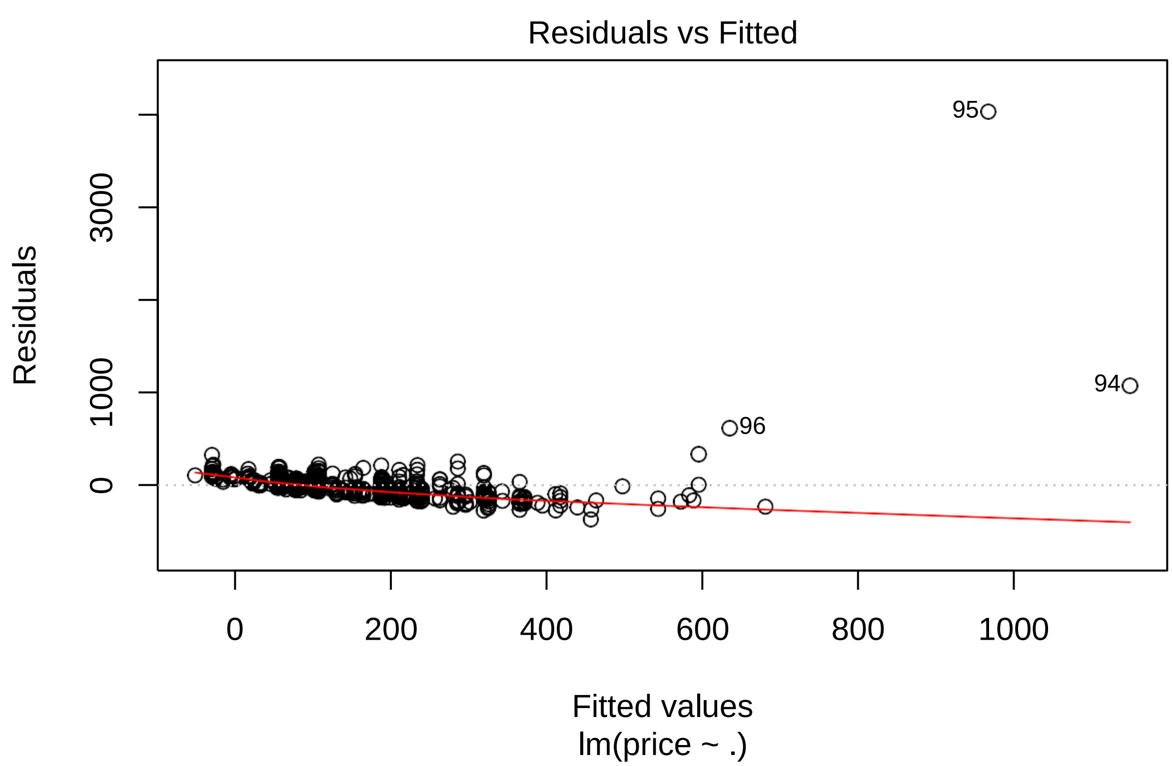
Part C

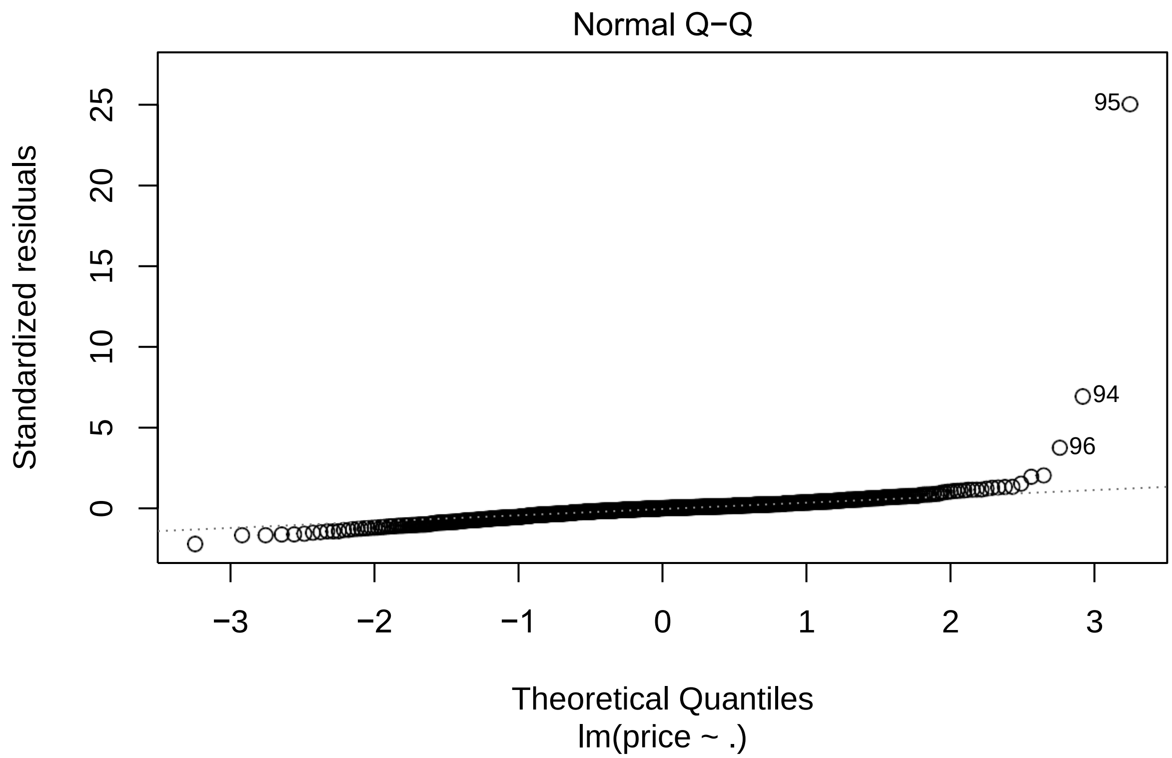
```
# create a data frame of the test listing
test_pt = data.frame(room_type = as.factor('Private room'), reviews = 70, overall_satisfaction = 4, acc

# predict the price of this test listing
paste0("The predicted price of this listing is: $", round(predict(airbnb_reg, test_pt),0))
```

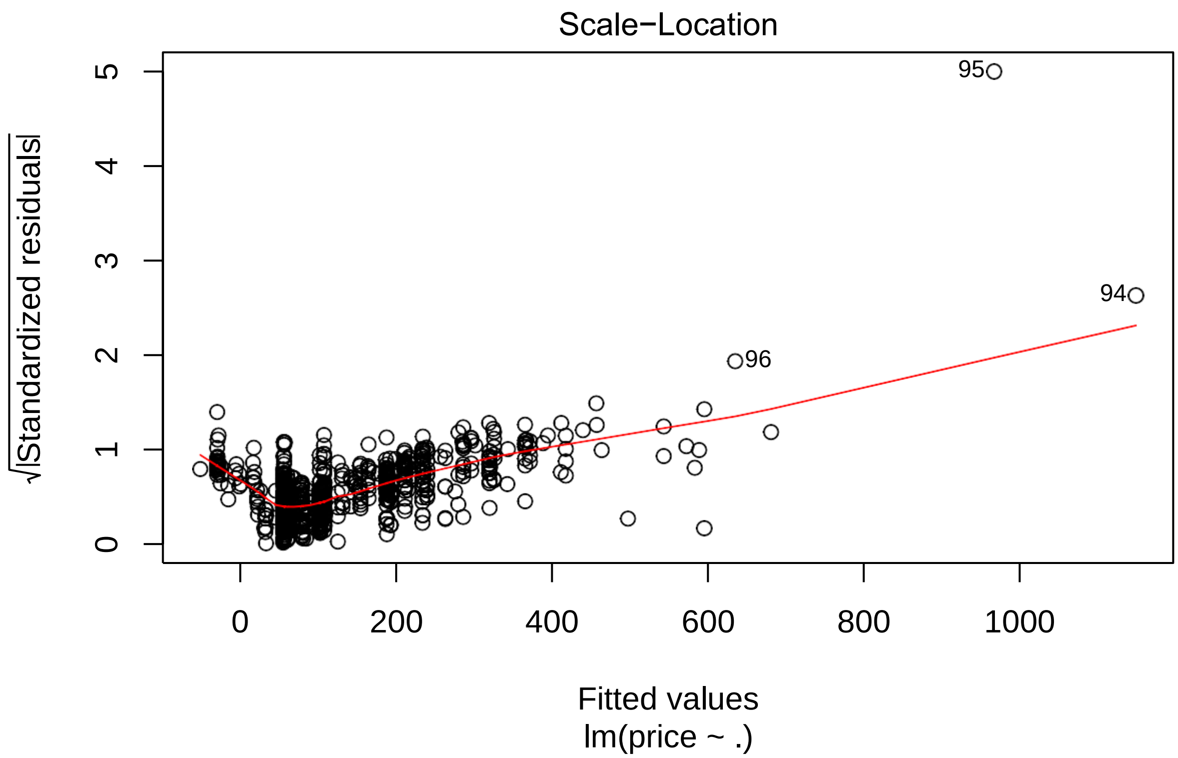
```
## [1] "The predicted price of this listing is: $66"
```

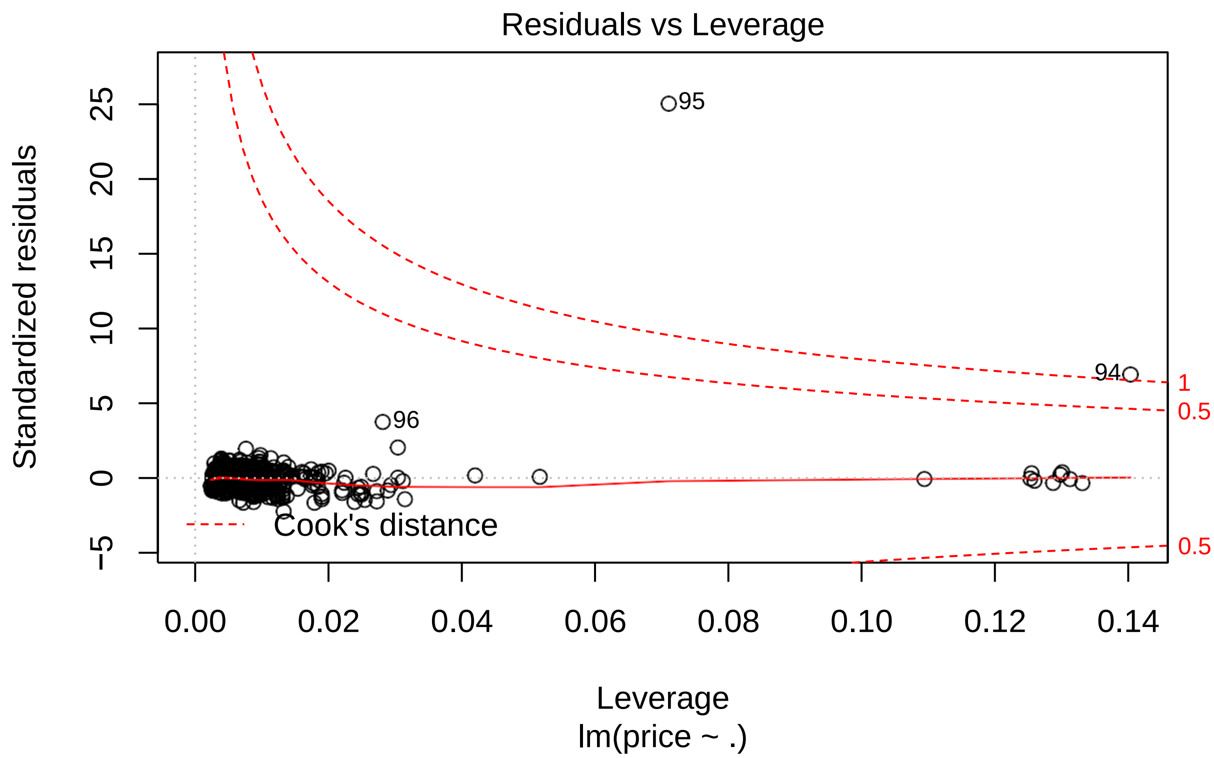
Part D

```
# plot the regression to get the Cook's distance plot
plot(airbnb_reg)
```




Normal Q−Q
Standardized residuals
0
5
10
15
20
25
95
94
96
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(price ~ .)


Scale−Location
√|Standardized residuals|
0
1
2
3
4
5
0
200
400
600
800
1000
95
94
96
Fitted values
lm(price ~ .)




```r
# get the index of the points from the dataset where the Cook's Distance is greater than 1
which(cooks.distance(airbnb_reg) > 1)
```

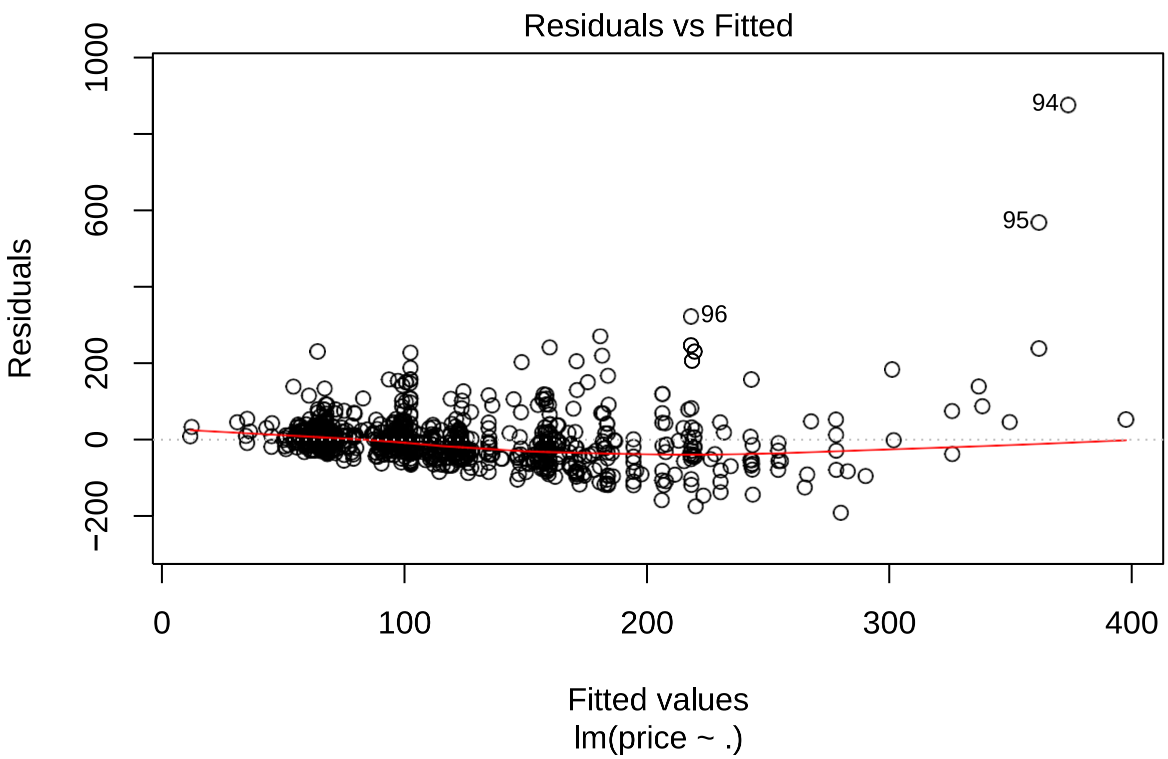
```
## 94 95
## 94 95
```

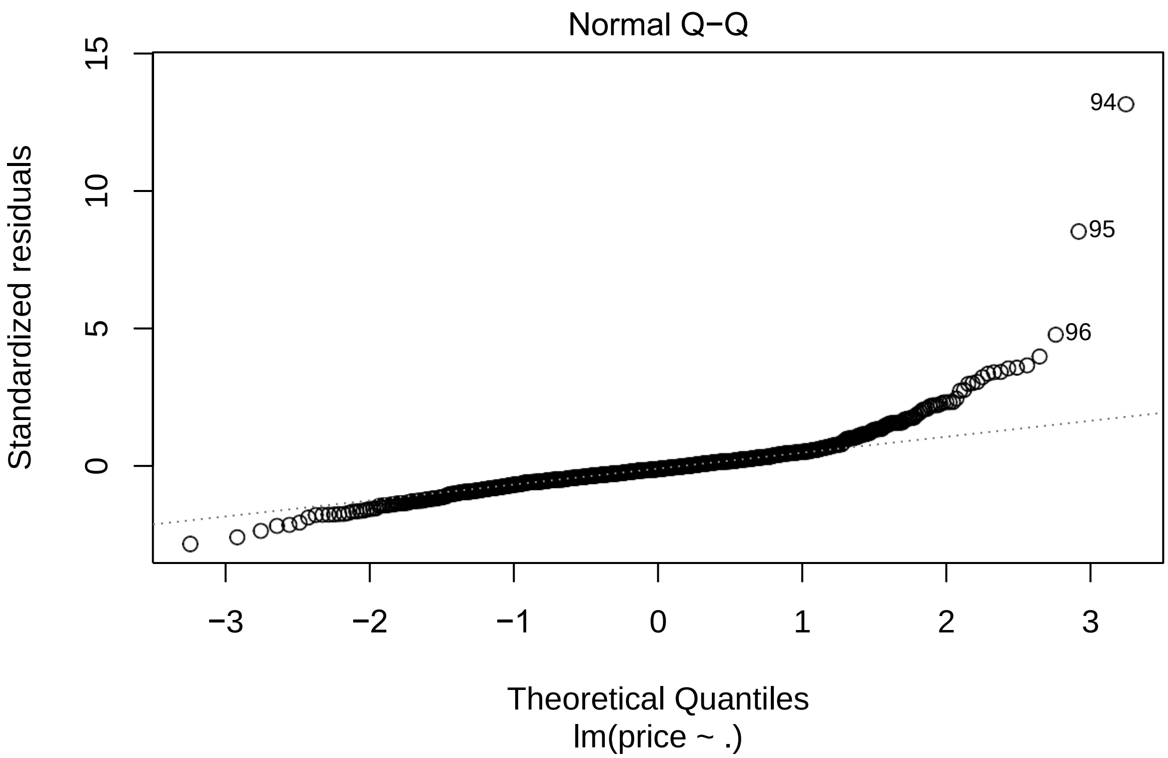
```r
# it appears that points 94 and 95 have a Cook's Distance greater than 1, so lets remove these points
airbnb = airbnb[-c(94,95),]


# rerun the regression after removing these outlier points
airbnb_reg_no_outliers = lm(price~., data = airbnb)
summary(airbnb_reg_no_outliers)
```

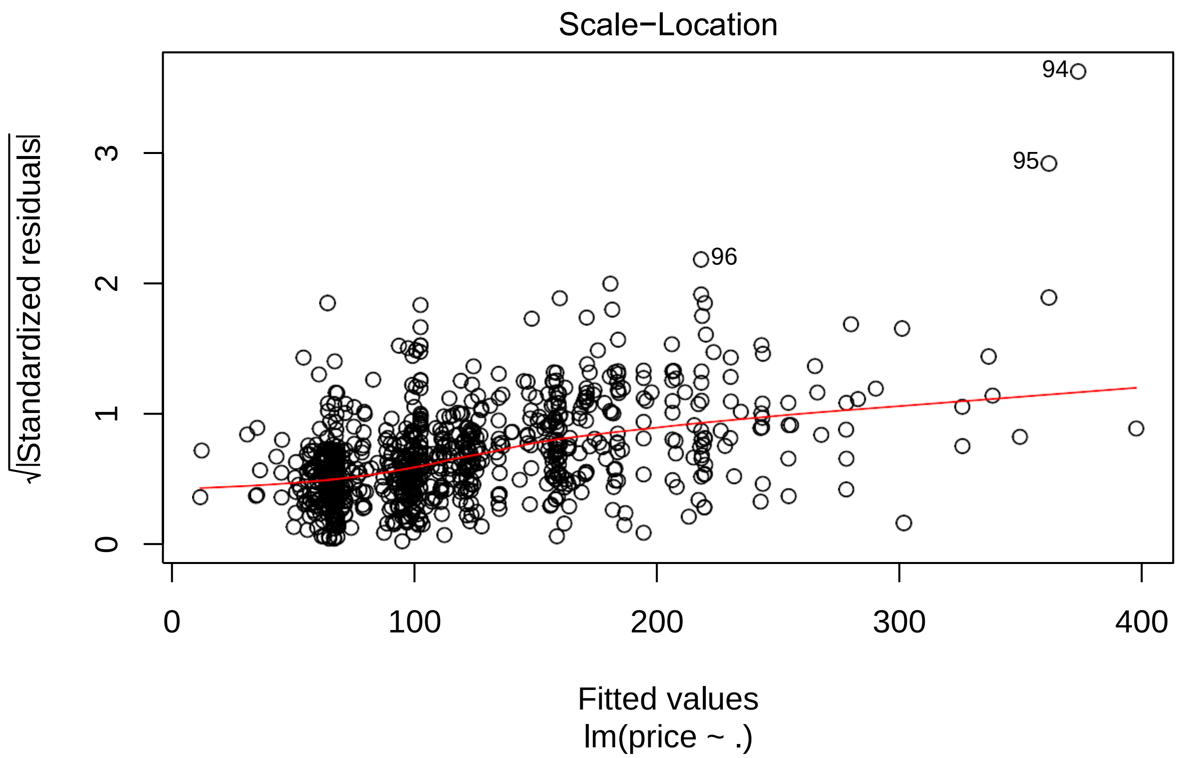
```
##
## Call:
## lm(formula = price ~ ., data = airbnb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -190.95  -32.43   -7.09   20.35  876.26
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              75.01310    9.09152   8.251 6.01e-16 ***
```

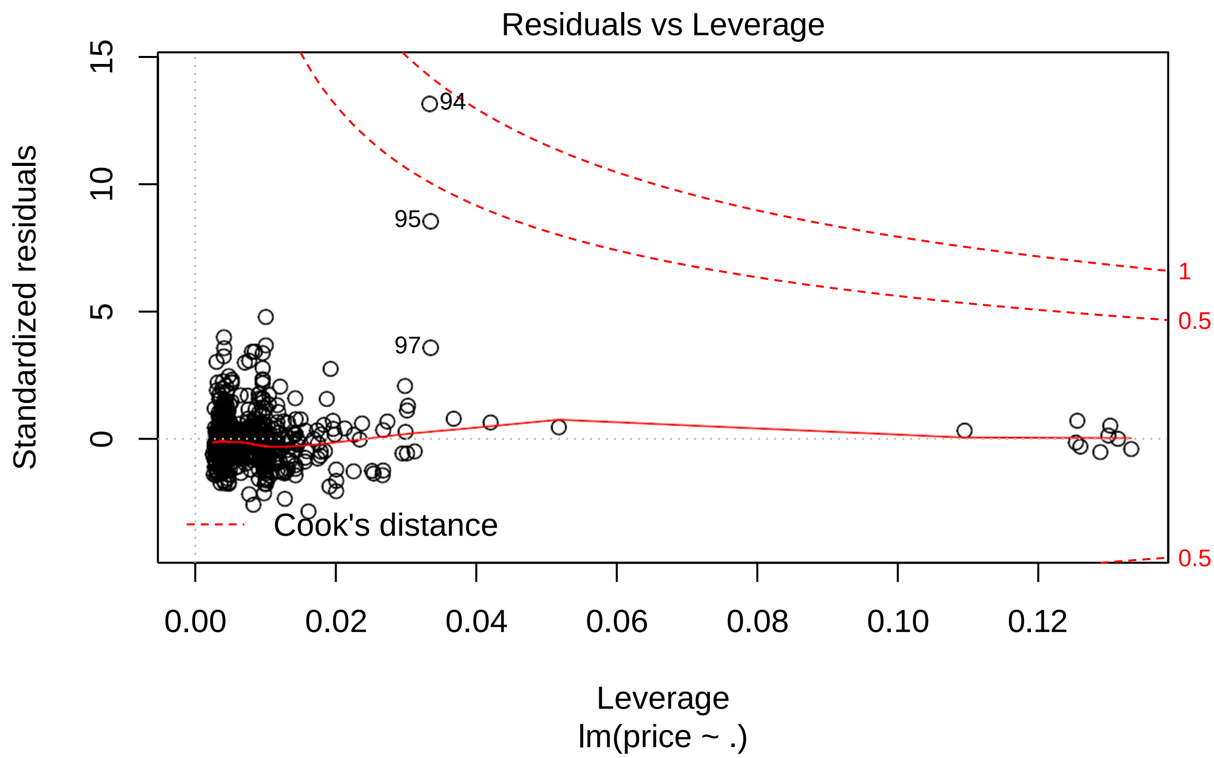
```
## room_typePrivate room  -32.28201    5.38034  -6.000 2.92e-09 ***
## room_typeShared room   -91.69951   24.28958  -3.775 0.000171 ***
## reviews                 -0.05915    0.04047  -1.462 0.144202
## overall_satisfaction    -6.78957    1.41118  -4.811 1.78e-06 ***
## accommodates            11.90698    2.14267   5.557 3.68e-08 ***
## bedrooms                35.93177    4.87968   7.364 4.25e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 67.73 on 845 degrees of freedom
## Multiple R-squared:  0.4249,  Adjusted R-squared:  0.4208
## F-statistic: 104 on 6 and 845 DF, p-value: < 2.2e-16
```

```r
plot(airbnb_reg_no_outliers)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(price ~ .)
94
95
96
0
5
10
15
−3
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
0
1
2
3
0
100
200
300
400
Fitted values
lm(price ~ .)
94
95
96



Question 2

Part A

```r
# read in data
marketing = read_csv("direct_marketing_2.csv")
```

```
## Parsed with column specification:
## cols(
##   Age = col_character(),
##   Gender = col_character(),
##   OwnHome = col_character(),
##   Married = col_character(),
##   Location = col_character(),
##   Salary = col_double(),
##   Children = col_double(),
##   History = col_character(),
##   Catalogs = col_double(),
##   AmountSpent = col_double()
## )
```

```r
# create indicator variables for history column
marketing$History_Low = ifelse(marketing$History == "Low", 1, 0)
```

```r
marketing$History_Medium = ifelse(marketing$History == "Medium", 1, 0)
marketing$History_High = ifelse(marketing$History == "High", 1, 0)

# create salary variables for each level of history
marketing = marketing %>% mutate(LowSalary = History_Low * Salary) %>%
                          mutate(MediumSalary = History_Medium * Salary) %>%
                          mutate(HighSalary = History_High * Salary)

# fit a regression using AmountSpent as the response and the indicator and salary variables as predicto
marketing_reg = lm(AmountSpent~History_Low + History_Medium + History_High + LowSalary + MediumSalary +
# view the model summary
summary(marketing_reg)
```

```
##
## Call:
## lm(formula = AmountSpent ~ History_Low + History_Medium + History_High +
##     LowSalary + MediumSalary + HighSalary, data = marketing)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -214.33  -35.19   -7.49   25.17  374.41
##
## Coefficients:
##                 Estimate Std. Error t value  Pr(>|t|)
##  (Intercept)     1.240e+02  3.912e+00  31.694  < 2e-16 ***
## History_Low     -9.658e+01  8.548e+00 -11.299  < 2e-16 ***
## History_Medium  -4.273e+01  1.423e+01  -3.004  0.00274 **
## History_High    -4.935e+01  1.732e+01  -2.850  0.00447 **
## LowSalary        2.573e-04  1.901e-04   1.354  0.17620
## MediumSalary     2.488e-04  2.321e-04   1.072  0.28397
## HighSalary       1.723e-03  1.954e-04   8.820  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 68.1 on 993 degrees of freedom
## Multiple R-squared: 0.501, Adjusted R-squared: 0.498
## F-statistic: 166.1 on 6 and 993 DF, p-value: < 2.2e-16
```

Part B

```r
# create test point for a low historic type and 10000 salary
test_low = data.frame(History_Low = 1, History_Medium=0, History_High = 0, LowSalary = 10000, MediumSal
# predict the amount spent for this individual
paste0("The predicted amount spent by a customer of the low historic type is: $", round(predict(marketi
```

```
## [1] "The predicted amount spent by a customer of the low historic type is: $29.98"
```

```r
# create test point for a medium historic type and 10000 salary
test_med = data.frame(History_Low = 0, History_Medium=1, History_High = 0, LowSalary = 0, MediumSalary
# predict the amount spent for this individual
paste0("The predicted amount spent by a customer of the medium historic type is: $", round(predict(mark
```

```
## [1] "The predicted amount spent by a customer of the medium historic type is: $83.75"
```

```r
# create test point for a high historic type and 10000 salary
test_high = data.frame(History_Low = 0, History_Medium=0, History_High = 1, LowSalary = 0, MediumSalary
# predict the amount spent for this individual
paste0("The predicted amount spent by a customer of the high historic type is: $", round(predict(market
```

```
## [1] "The predicted amount spent by a customer of the high historic type is: $91.87"
```

Amount Spent by Historic Type:

- Low: $29.98
- Medium: $83.75
- High: $91.87

Part C

```r
# read in the data
airbnb2 = read_csv('airbnb_data.csv')
```

```
## Parsed with column specification:
## cols(
##   room_id = col_double(),
##   survey_id = col_double(),
##   host_id = col_double(),
##   room_type = col_character(),
##   city = col_character(),
##   reviews = col_double(),
##   overall_satisfaction = col_double(),
##   accommodates = col_double(),
##   bedrooms = col_double(),
##   price = col_double()
## )
```

```r
# select out the 2 fields that will be used in the regression
airbnb2 = airbnb2 %>% dplyr::select(price, overall_satisfaction)
# add in a ln(overall_satisfaction) column to take care of the 0 values accordingly
# ln(0) is undefined, so for any nonzero value do ln(x), but if the value is 0, do ln(x+1) for the tran
airbnb2$ln_overall_satisfaction = ifelse(airbnb2$overall_satisfaction == 0, log(airbnb2$overall_satisfa

# fit a linear-linear model to the data
lin_lin = lm(price~overall_satisfaction, data = airbnb2)
summary(lin_lin)
```

```
##
## Call:
## lm(formula = price ~ overall_satisfaction, data = airbnb2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -167.0   -51.3   -24.2    16.8  4805.0
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           194.967     17.698  11.016  < 2e-16 ***
## overall_satisfaction  -16.353      3.903  -4.189 3.09e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 200.4 on 852 degrees of freedom
## Multiple R-squared:  0.02018,	Adjusted R-squared:  0.01903
## F-statistic: 17.55 on 1 and 852 DF,  p-value: 3.088e-05
```

```r
# fit a linear-log model to the data
lin_log = lm(price~ln_overall_satisfaction, data = airbnb2)
summary(lin_log)
```

```
##
## Call:
## lm(formula = price ~ ln_overall_satisfaction, data = airbnb2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -168.0   -51.4   -24.5    16.5  4804.0
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               196.01      17.74   11.05  < 2e-16 ***
## ln_overall_satisfaction   -51.25      12.09   -4.24 2.48e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 200.4 on 852 degrees of freedom
## Multiple R-squared:  0.02067,	Adjusted R-squared:  0.01952
## F-statistic: 17.98 on 1 and 852 DF,  p-value: 2.479e-05
```

```r
# fit a log-linear model to the data
log_lin = lm(log(price)~overall_satisfaction, data = airbnb2)
summary(log_lin)
```

```
##
## Call:
## lm(formula = log(price) ~ overall_satisfaction, data = airbnb2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -1.6234  -0.3525  -0.0432   0.3302   3.7220
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            4.79515    0.05083  94.339  < 2e-16 ***
## overall_satisfaction  -0.04401    0.01121  -3.926 9.33e-05 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5757 on 852 degrees of freedom
## Multiple R-squared:  0.01777,	Adjusted R-squared:  0.01662
## F-statistic: 15.41 on 1 and 852 DF,  p-value: 9.331e-05
```

```
# fit a log-log model to the data
log_log = lm(log(price)~ln_overall_satisfaction, data = airbnb2)
summary(log_log)
```

```
##
## Call:
## lm(formula = log(price) ~ ln_overall_satisfaction, data = airbnb2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6109 -0.3558 -0.0362  0.3300  3.7161
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              4.80114    0.05095  94.241  < 2e-16 ***
## ln_overall_satisfaction -0.14031    0.03471  -4.043 5.76e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5754 on 852 degrees of freedom
## Multiple R-squared:  0.01882,	Adjusted R-squared:  0.01767
## F-statistic: 16.34 on 1 and 852 DF,  p-value: 5.763e-05
```

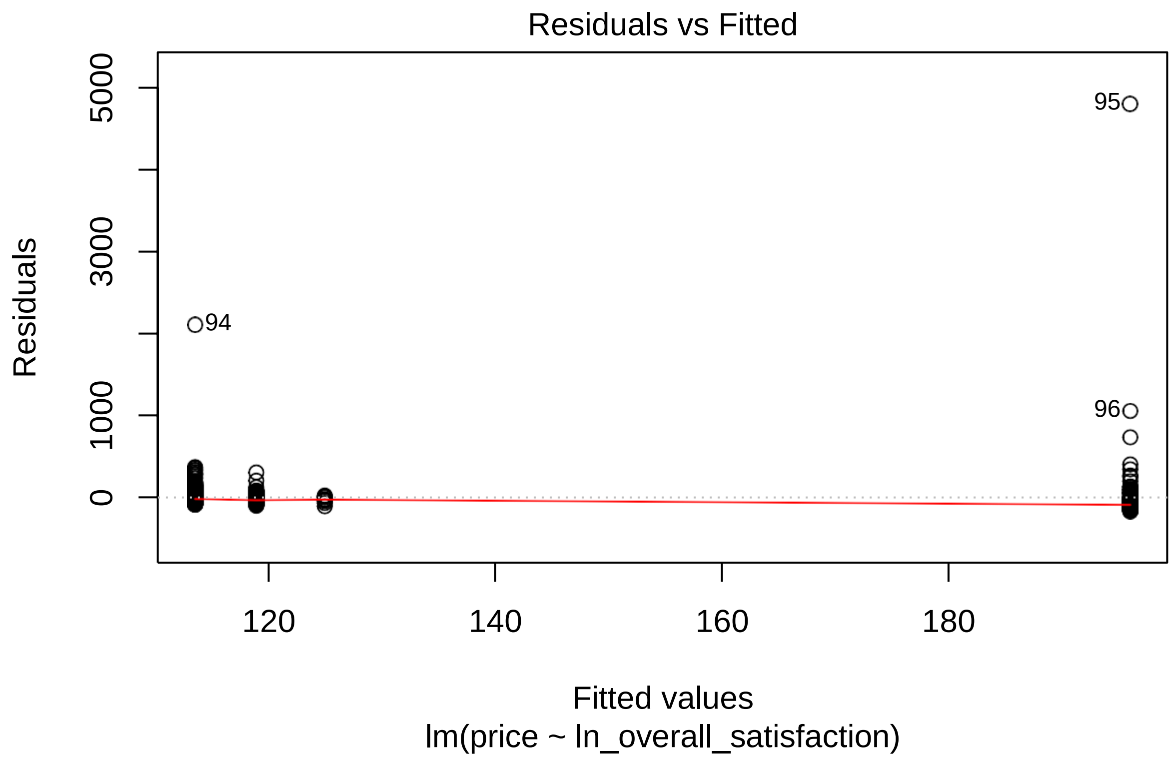
Part D

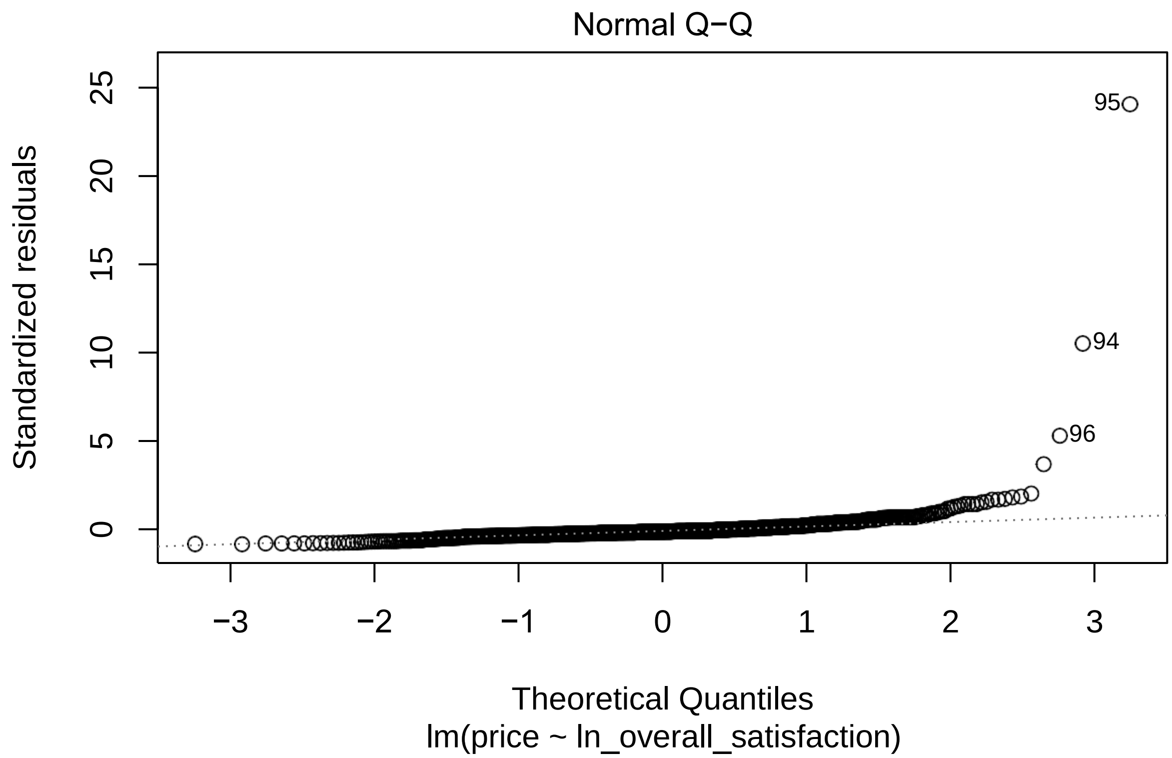
R^2 value of each of the transformed models:

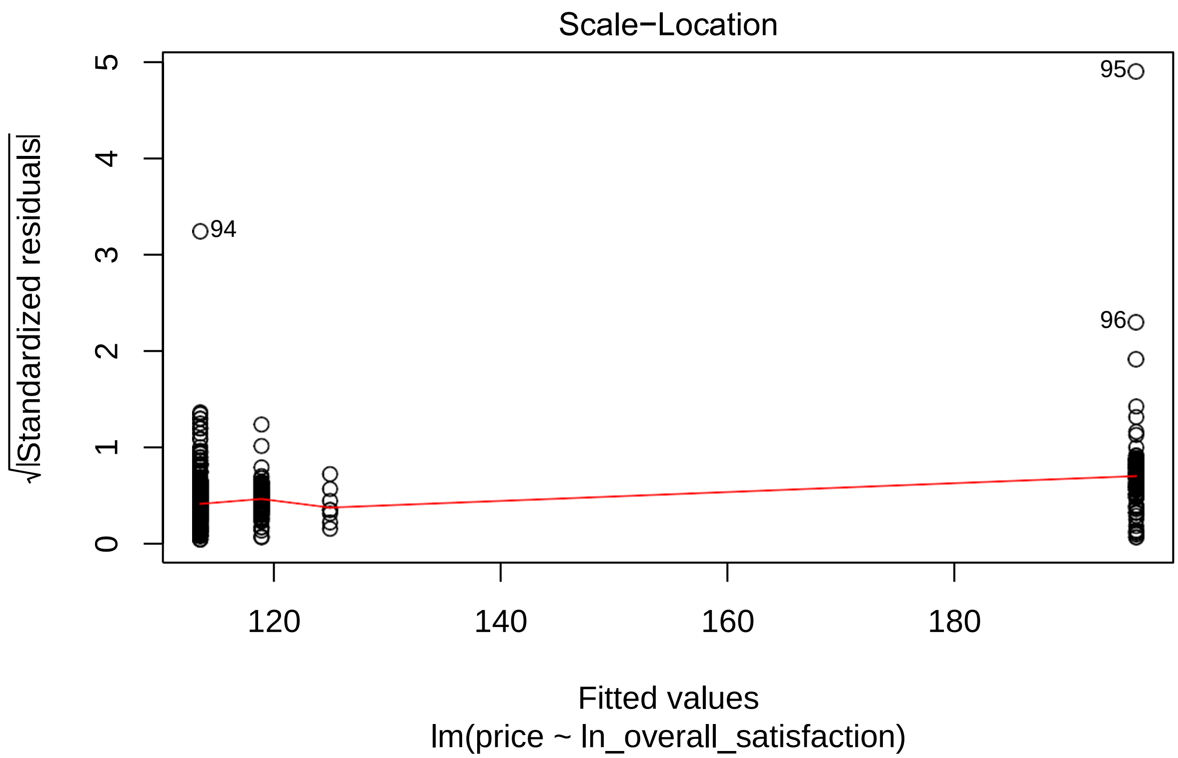
- linear-linear: 0.02018
- **linear-log: 0.02067**
- log-linear: 0.01777
- log-log: 0.01882

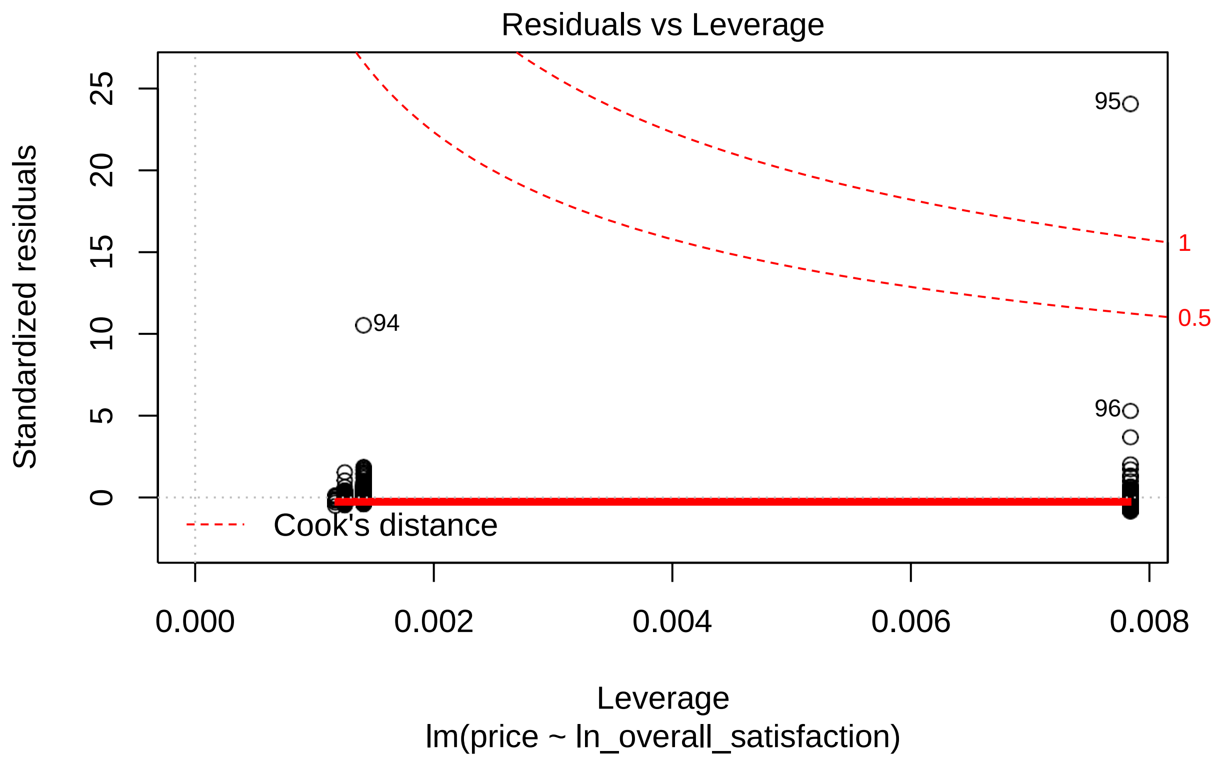
The linear-log model had the best R^2 value. None of these transformations provided very good results with low R^2 values below 3% across the board. I would want to look into using other predictors in this model that would probably be more useful in prediction, as the overall_satisfaction variable explains very little of the variance in the data. Out of these options, I would choose to keep the dependent variable (price) linear and perform a log transform on the the predictor variable (overall_satisfaction) to get the best results in this case.

```
# look at the regression plots of the selected model
plot(lin_log)
```


Residuals vs Fitted
Residuals
5000
3000
1000
0
94
95
96
120
140
160
180
Fitted values
lm(price ~ ln_overall_satisfaction)


Normal Q−Q
Standardized residuals
0
5
10
15
20
25
95
94
96
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(price ~ ln_overall_satisfaction)


Scale−Location
√|Standardized residuals|
0
1
2
3
4
5
94
95
96
120
140
160
180
Fitted values
lm(price ~ ln_overall_satisfaction)



Based on the Q-Q plot, it appears that this transformation still doesn't meet the normality assumption required for linear regression, as the points deviate from the straight line near the upper tail. More transformations or addition of higher order terms would be required to alleviate this concern.

Question 3

Part A

```
# read in the data
titanic = read_csv("titanic_data.csv")
```

```
## Parsed with column specification:
## cols(
##   Name = col_character(),
##   PClass = col_character(),
##   Age = col_double(),
##   Sex = col_character(),
##   Survived = col_double()
## )
```

```
# there are no missing values in the data
sum(is.na(titanic))
```

```
## [1] 0
```

```r
# convert the survived variable to a factor
titanic$Survived = as.factor(titanic$Survived)

# fit a logistic regression model to predict Survived from the Sex variable
titanic_logistic = glm(Survived~Sex, data = titanic, family = 'binomial')
# display the model summary (Sex = Female is the base case)
summary(titanic_logistic)
```

```
##
## Call:
## glm(formula = Survived ~ Sex, family = "binomial", data = titanic)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
##  -1.6735  -0.6776  -0.6776   0.7524   1.7800
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.1172     0.1367   8.171 3.05e-16 ***
## Sexmale      -2.4718     0.1783 -13.861  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1025.57  on 755  degrees of freedom ##
Residual deviance:  796.64  on 754  degrees of freedom ##
AIC: 800.64
##
## Number of Fisher Scoring iterations: 4
```

Part B

The intercept coefficient in this logistic regression model of 1.1172 represents the log odds of survival for females, since Sex = Female is the base case for the model.

The coefficient on the Sexmale variable for this logistic regression model means that being male decreases the log odds of survival by 2.4718 compared to the log odds of survival of a female.

Part C

```r
# create a female test point
female = data.frame(Sex = "female")
# predict the probability of survival for the female
predict(titanic_logistic, female, type = "response")
```

```
##         1
## 0.7534722
```

The probability of survival for a female is about 0.753

Part D

```
# create a male test point
male = data.frame(Sex = "male")
# predict the probability of survival for the male
predict(titanic_logistic, male, type = "response")
```

```
##         1
## 0.2051282
```

The probability of survival for a male is about 0.205, which is much lower than that of the females on the Titanic.