Data 621 - Blog 3

Leticia Salazar

November 14, 2022

Contents

Multiple Linear Regression

The second blog demonstrated how to create a simple linear regression model to find relationships between two quantitative variables where there's one dependent variable (y) and one independent variable (x). In this third blog I am introducing multiple linear regression, where there's still one dependent variable but one or more independent variables.

Below I'll demonstrate how to start a multiple linear regression model using the `diamonds` package from the library `ggplot2`. For a more in depth multiple regression model I'd like to share the work my team and I create for a homework.

Load Libraries

```
library(ggplot2)
library(tidyverse)
library(jtools)
library(hrbrthemes)
library(gtsummary)
```

Loading Data

Loading data using the `head()` function

```
## # A tibble: 6 x 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
```

```
## 3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
## 4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
## 5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

Using `summary()` function to get the statistical structure of our data

```
##      carat               cut        color        clarity          depth
##  Min.   :0.2000   Fair      : 1610   D: 6775   SI1    :13065   Min.   :43.00
##  1st Qu.:0.4000   Good      : 4906   E: 9797   VS2    :12258   1st Qu.:61.00
##  Median :0.7000   Very Good:12082   F: 9542   SI2    : 9194   Median :61.80
##  Mean   :0.7979   Premium   :13791   G:11292   VS1    : 8171   Mean   :61.75
##  3rd Qu.:1.0400   Ideal     :21551   H: 8304   VVS2   : 5066   3rd Qu.:62.50
##  Max.   :5.0100                     I: 5422   VVS1   : 3655   Max.   :79.00
##                                     J: 2808   (Other): 2531
##      table           price             x                y
##  Min.   :43.00   Min.   :  326   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710   1st Qu.: 4.720
##  Median :57.00   Median : 2401   Median : 5.700   Median : 5.710
##  Mean   :57.46   Mean   : 3933   Mean   : 5.731   Mean   : 5.735
##  3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540   3rd Qu.: 6.540
##  Max.   :95.00   Max.   :18823   Max.   :10.740   Max.   :58.900
##
##        z
##  Min.   : 0.000
##  1st Qu.: 2.910
##  Median : 3.530
##  Mean   : 3.539
##  3rd Qu.: 4.040
##  Max.   :31.800
##
```

Using `gtsummary()` function as well to view the data's structure differently

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	**N = 53,940**
carat	0.70 (0.40, 1.04)
cut	
Fair	1,610 (3.0%)
Good	4,906 (9.1%)
Very Good	12,082 (22%)
Premium	13,791 (26%)
Ideal	21,551 (40%)
color	
D	6,775 (13%)

Characteristic	N = 53,940
E	9,797 (18%)
F	9,542 (18%)
G	11,292 (21%)
H	8,304 (15%)
I	5,422 (10%)
J	2,808 (5.2%)
clarity	
I1	741 (1.4%)
SI2	9,194 (17%)
SI1	13,065 (24%)
VS2	12,258 (23%)
VS1	8,171 (15%)
VVS2	5,066 (9.4%)
VVS1	3,655 (6.8%)
IF	1,790 (3.3%)
depth	61.80 (61.00, 62.50)
table	57.00 (56.00, 59.00)
price	2,401 (950, 5,324)
x	5.70 (4.71, 6.54)
y	5.71 (4.72, 6.54)
z	3.53 (2.91, 4.04)

The `diamonds` data set does not require any cleaning or transformation in order to use but always keep in mind the data sets we use in the future will.

Model Building

Just like in simple linear regression, we start by creating a model to look at the overall data. This serves as a base model to compare our other models to. The equation for a multiple linear regression model is: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 ... + \beta_p x_p + \epsilon$

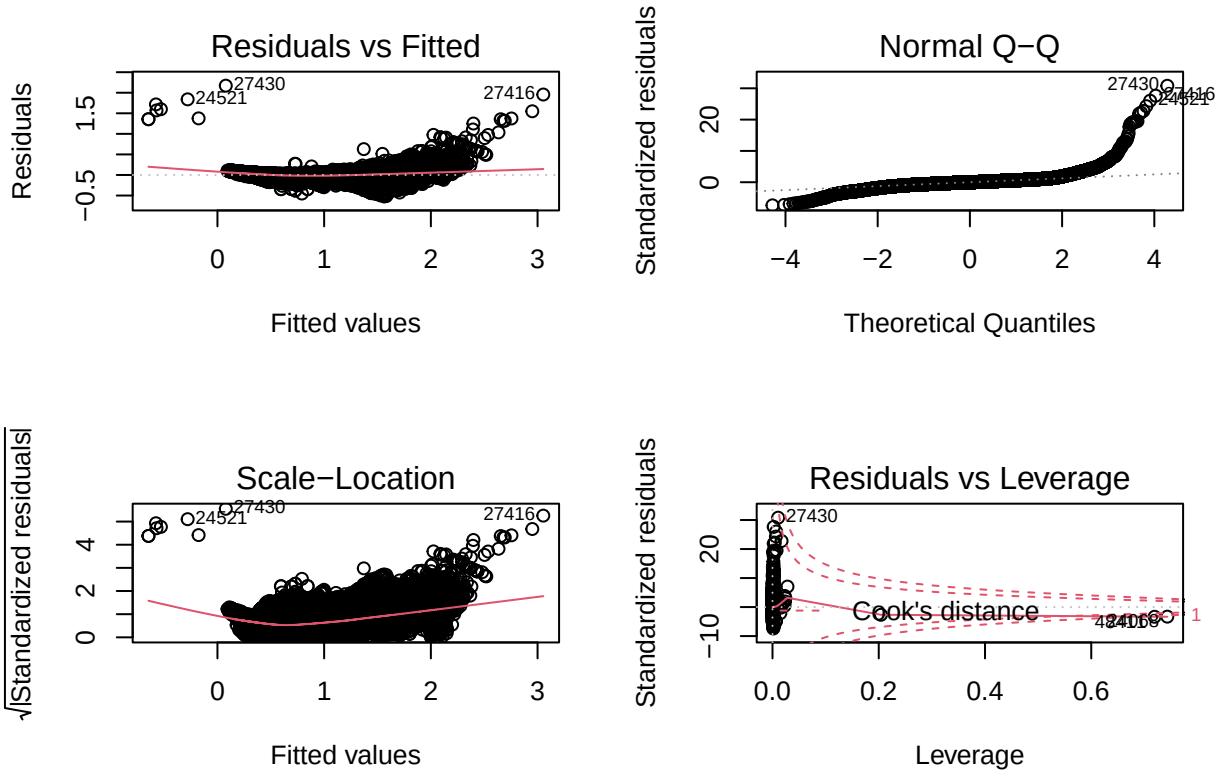
```
# creating first model
model <- lm(diamonds)
summary(model)
```

```
##
## Call:
## lm(formula = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52251 -0.03060 -0.00102  0.02905  2.17188
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.660e+00  2.388e-02 -69.512  < 2e-16 ***
## cut.L       -2.057e-02  1.416e-03 -14.531  < 2e-16 ***
```

```
## cut.Q        9.257e-03  1.131e-03   8.186 2.76e-16 ***
## cut.C       -6.879e-03  9.715e-04  -7.081 1.45e-12 ***
## cut^4        1.741e-03  7.762e-04   2.244  0.02487 *
## color.L      1.085e-01  1.115e-03  97.305  < 2e-16 ***
## color.Q      4.106e-02  9.904e-04  41.461  < 2e-16 ***
## color.C      5.772e-03  9.243e-04   6.245 4.28e-10 ***
## color^4     -5.063e-03  8.482e-04  -5.969 2.40e-09 ***
## color^5      5.713e-03  8.014e-04   7.130 1.02e-12 ***
## color^6      1.959e-03  7.285e-04   2.689  0.00716 **
## clarity.L   -1.784e-01  2.058e-03 -86.670  < 2e-16 ***
## clarity.Q    1.081e-01  1.785e-03  60.536  < 2e-16 ***
## clarity.C   -5.677e-02  1.518e-03 -37.391  < 2e-16 ***
## clarity^4    1.727e-02  1.211e-03  14.259  < 2e-16 ***
## clarity^5   -1.232e-02  9.885e-04 -12.464  < 2e-16 ***
## clarity^6   -1.793e-03  8.602e-04  -2.084  0.03717 *
## clarity^7    1.719e-04  7.595e-04   0.226  0.82093
## depth        1.212e-02  2.801e-04  43.278  < 2e-16 ***
## table        2.203e-03  1.825e-04  12.073  < 2e-16 ***
## price        4.428e-05  1.913e-07 231.494  < 2e-16 ***
## x            2.425e-01  1.800e-03 134.732  < 2e-16 ***
## y            5.961e-03  1.212e-03   4.917 8.82e-07 ***
## z            4.586e-03  2.100e-03   2.184  0.02899 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07088 on 53916 degrees of freedom
## Multiple R-squared:  0.9777, Adjusted R-squared:  0.9776
## F-statistic: 1.025e+05 on 23 and 53916 DF,  p-value: < 2.2e-16
```

Plotting our models is the easiest part, first you'll want to create a matrix of nrows x ncols so that the plots are fitted into one page. Using the `plot()` function we can insert our model name and we get 4 different plots:

```
# plots
par(mfrow=c(2,2)) # matrix of plots
plot(model) # create plots for model
```



1. **Residuals vs Fitted**: scatter plot of the residuals on the y-axis and fitted values on the x-axis. The plot is used to detect non-linearity, whether homoskedasticity holds and outliers.
2. **Normal Q-Q**: scatter plot of two sets of quantiles against one another. The theoretical quantities helps us assess if a set of data came from the theoretical distribution such as a Normal or exponential.
3. **Scale-Location**: scatter plot showing the spread of the residuals along the ranges of the predictors. Similarly to the residuals vs fitted it simplifies the analysis of the homoskedasticity assumption.
4. **Residuals vs. Leverage**: scatter plot that looks at the spread of the standardized residuals and it's changes in leverage or sensitivity of the fitted. It can also be used to detect heteroskedasticity and non-linearity. It takes into consideration Cook's distance to detect any points that have influence on the model.

With multiple linear regression there are assumptions to keep in mind:

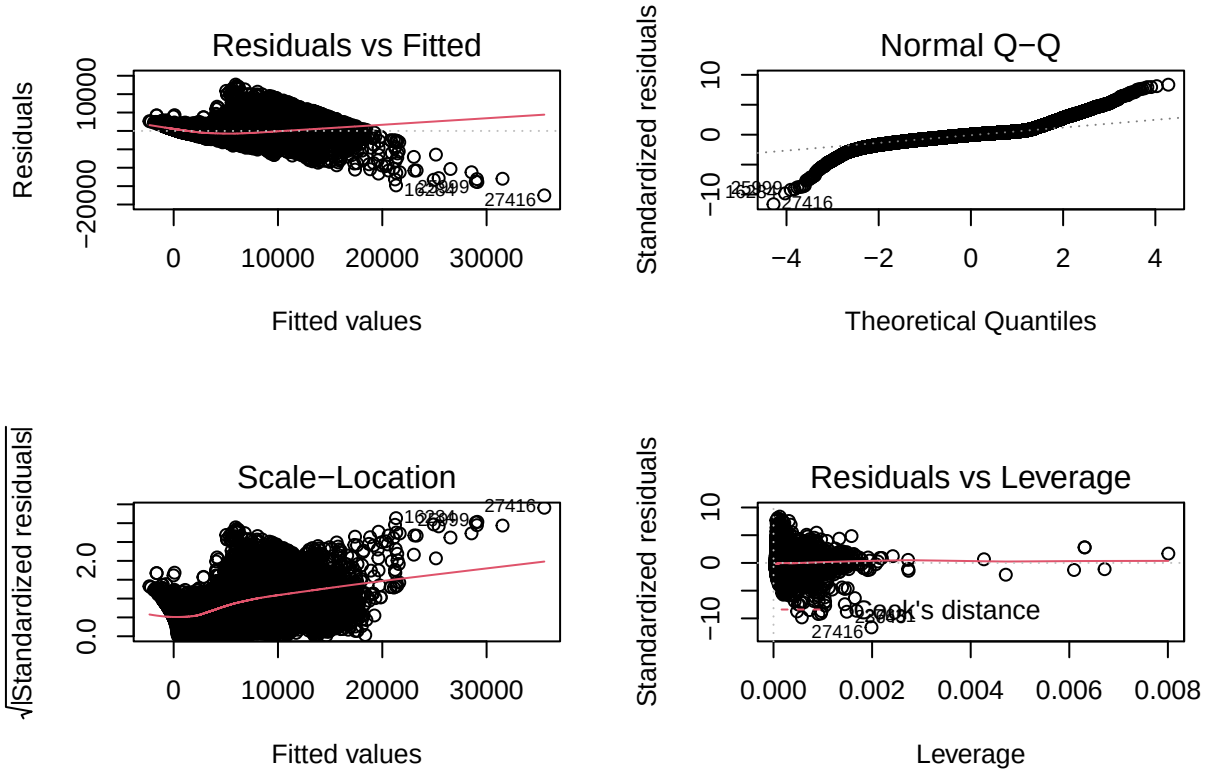
- Linearity of the relationships between the dependent and independent variables
- Independence of the observations
- Normality of the residuals
- Homoscedasticity of the residuals
- No influential points (outliers)
- No multicollinearity arises when there is a strong linear correlation between the independent variables, conditional on the other variables in the model. It is important to check it because it may lead to an imprecision or an instability of the estimated parameters when a variable changes.

The second model we create, similar to the first, will be with less variables to check if this model is fits better.

##

```
## Call:
## lm(formula = price ~ carat + cut + depth + table, data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17513.5   -786.6    -39.9    521.4  12596.8
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4296.951    468.456   9.173  < 2e-16 ***
## carat       7890.772     14.047 561.745  < 2e-16 ***
## cut.L       1028.088     29.667  34.654  < 2e-16 ***
## cut.Q       -483.298     23.749 -20.350  < 2e-16 ***
## cut.C        325.172     20.528  15.840  < 2e-16 ***
## cut^4         59.229     16.466   3.597 0.000322 ***
## depth        -73.691      5.302 -13.898  < 2e-16 ***
## table        -41.816      3.881 -10.775  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1508 on 53932 degrees of freedom
## Multiple R-squared:  0.8571, Adjusted R-squared:  0.8571
## F-statistic: 4.62e+04 on 7 and 53932 DF,  p-value: < 2.2e-16
```

Plots for model2

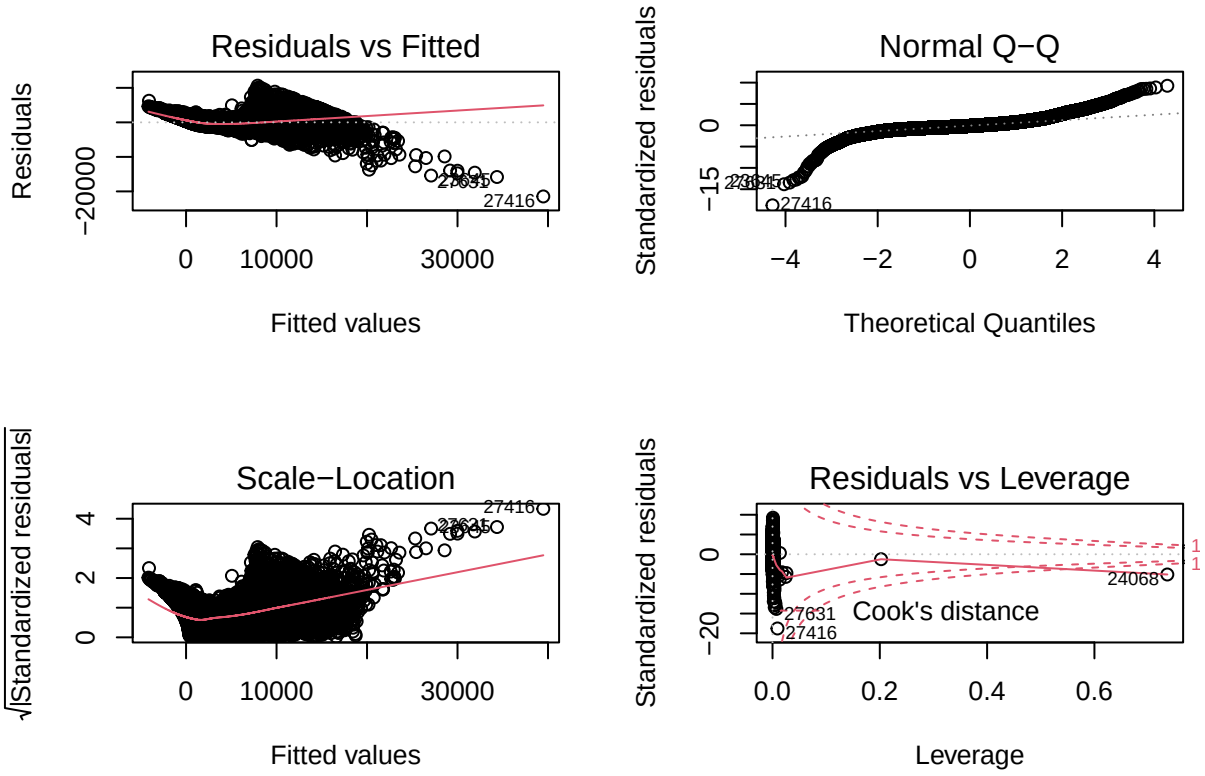


The next two models (`model3` and `model4`) are created using backwards elimination to view the differences in statistical output we receive and compare the adjusted R^2.

model3

```
##
## Call:
## lm(formula = price ~ carat + color + clarity + x + y, data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21463.8   -598.8   -178.9    390.1  10627.9
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept)    42.25      79.13    0.534   0.5934
## carat       10968.05      47.96  228.710  < 2e-16 ***
## color.L     -1964.82      17.61 -111.548  < 2e-16 ***
## color.Q      -671.29      16.03  -41.885  < 2e-16 ***
## color.C      -174.13      14.96  -11.642  < 2e-16 ***
## color^4        33.79      13.74    2.459   0.0139 *
## color^5       -96.31      12.98   -7.419 1.19e-13 ***
## color^6       -54.48      11.80   -4.616 3.91e-06 ***
## clarity.L    4344.21      30.11  144.271  < 2e-16 ***
## clarity.Q   -2032.91      28.42  -71.531  < 2e-16 ***
## clarity.C    1076.63      24.38   44.162  < 2e-16 ***
## clarity^4    -418.98      19.51  -21.480  < 2e-16 ***
## clarity^5     266.60      15.98   16.680  < 2e-16 ***
## clarity^6     -13.07      13.92   -0.939   0.3478
## clarity^7     104.88      12.29    8.531  < 2e-16 ***
## x            -958.86      27.58  -34.764  < 2e-16 ***
## y              39.06      19.39    2.015   0.0439 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1148 on 53923 degrees of freedom
## Multiple R-squared:  0.9172, Adjusted R-squared:  0.9171
## F-statistic: 3.731e+04 on 16 and 53923 DF,  p-value: < 2.2e-16
```

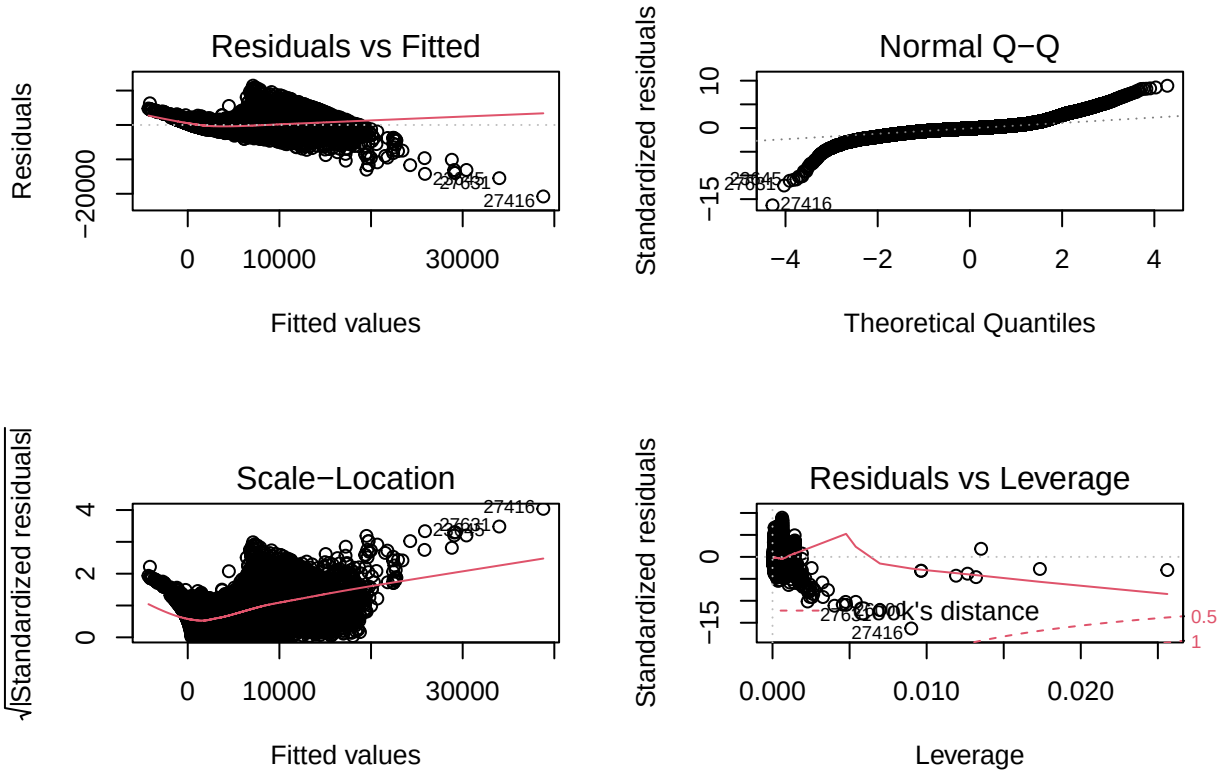
Plots for model3



model4

```
##
## Call:
## lm(formula = price ~ carat + clarity + x, data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20779.7   -570.8   -111.1    416.0  11419.4
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -48.08      88.06  -0.546    0.585
## carat       10165.82      52.82 192.464   <2e-16 ***
## clarity.L    4157.89      33.35 124.663   <2e-16 ***
## clarity.Q   -2005.70      31.62 -63.433   <2e-16 ***
## clarity.C    1060.55      27.16  39.049   <2e-16 ***
## clarity^4    -487.67      21.72 -22.455   <2e-16 ***
## clarity^5     305.51      17.80  17.162   <2e-16 ***
## clarity^6      12.29      15.52   0.792    0.428
## clarity^7     186.09      13.68  13.605   <2e-16 ***
## x            -755.46      22.46 -33.632   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1280 on 53930 degrees of freedom
## Multiple R-squared:  0.897,  Adjusted R-squared:  0.897
## F-statistic: 5.219e+04 on 9 and 53930 DF,  p-value: < 2.2e-16
```

Plots of model4



After plotting our models and analyzing the summary outputs we conclude that the model that better fits our data is `model3`. Comparing statistical outputs, `model3` had the highest R^2 at .9171, compared to `model2` at .8571 and `model4` at .897. While this simple analysis works with our data, a deeper analysis will always help you select the best fit model for your data, perform predictions and more.

If you'd like to dig deeper into multiple linear regression I found a great article from R-Bloggers that takes you step by step on the process to create your own models.

References:

- Assumptions of multiple linear regression | by MD Sohel Mahmood ... (n.d.). Retrieved November 14, 2022, from https://towardsdatascience.com/assumptions-of-multiple-linear-regression-d16f2eb8a2e7
- Alex. (2022, February 2). Linear regression plots: Fitted vs residuals. Boostedml. Retrieved November 14, 2022, from https://boostedml.com/2019/03/linear-regression-plots-fitted-vs-residuals.html
- Alex. (2020, May 20). The QQ plot in linear regression. Boostedml. Retrieved November 14, 2022, from https://boostedml.com/2019/03/linear-regression-plots-how-to-read-a-qq-plot.html
- Alex. (2020, May 20). The scale location plot: Interpretation in R. Boostedml. Retrieved November 14, 2022, from https://boostedml.com/2019/03/linear-regression-plots-scale-location-plot.html
- Alex. (2020, September 10). Linear regression plots: Residuals vs leverage. Boostedml. Retrieved November 14, 2022, from https://boostedml.com/2019/03/linear-regression-plots-residuals-vs-leverage.html
- Multiple linear regression made simple. Stats and R. (n.d.). Retrieved November 14, 2022, from https://statsandr.com/blog/multiple-linear-regression-made-simple/#multiple-linear-regression