Scarborough Housing Analysis

Authors

By: Zhiye Luan (1004282867), Yang Liu (1002100353), Ziyue Xian (1002106106)

Introduction

This report exists as a review of housing prices of apartments, houses, and townhouses in the heart of Scarborough in the years of 2017 and 2019. The factors that will be considered are the number of bedrooms in the house, the number of bathrooms and, and the type of housing it is. The goal of this report is to find a correlation between the two features that a type of property has to its listing price.

To begin, we have to import these libraries to use in RStudio.

```
library(tidyverse)
library(readxl)
library(devtools)
```

```
## Error in get(genname, envir = envir) : object 'testthat_print' not found
```

```
library(qqplotr)
library(MASS)
library(broom)
```

Gathering Data

Reading in data.

```
houses <- read_xlsx("houses.xlsx")
houses
```

```
## # A tibble: 54 x 5
##    `mls listing` asking bedrooms bathrooms type
##            <dbl>  <dbl>    <dbl>     <dbl> <chr>
##  1       3573407 275000      1.5         1 apartment
##  2       3580709 434999      3.5         3 townhouse
##  3       3582848 675000      5           2 house
##  4       3585741 385000      3.5         3 townhouse
##  5       3599567 529000      3.5         2 apartment
##  6       3602510 849888      5           2 house
##  7       3607807 139900      3           2 apartment
##  8       3610979 334900      2           2 apartment
##  9       3611464 249000      3           2 townhouse
## 10       3617396 259000      5           2 townhouse
## # ... with 44 more rows
```

As shown in the chart above, the data is seperated into five seperate columns. The first column is the "mls listing". This is a string of numbers (sometimes a character followed by a number) akin to a serial number in stores but reserved exclusively for real estate listings. In theory, this number is arbitrary and only holds

information regarding the chronological order of when the listing was made, so it should hold no significance in predicting or describing the price of real estate. The second column is called “asking”. This column represents the monetary compensation of purchasing the residence associated with property in Canadian dollars. The third column is called “bedrooms”. This column represents how many bedrooms are available in purchasing the property in float values. If you look carefully at the table, it becomes clear that the numbers are not in whole integers. The “half a room” value describes children’s bedrooms or very small rooms as real estate agents see as having half of the worth of a full sized bedroom.

The fourth column “bathrooms is describing the number of bathrooms that comes with the property, and finally the fifth column”type" describes which type of property that is being listed falls under. Values of the fifth column are either an apartment, a house, or a townhouse.

Observations and Analysis

To start analysis off, we will do a multiple regressional model that will model the number of bedrooms and bathrooms a property has to its listing price. This will be the basis for all other models in this report.

```
total_reg_model = lm(asking ~ bedrooms + bathrooms, data = houses)
summary(total_reg_model)
```

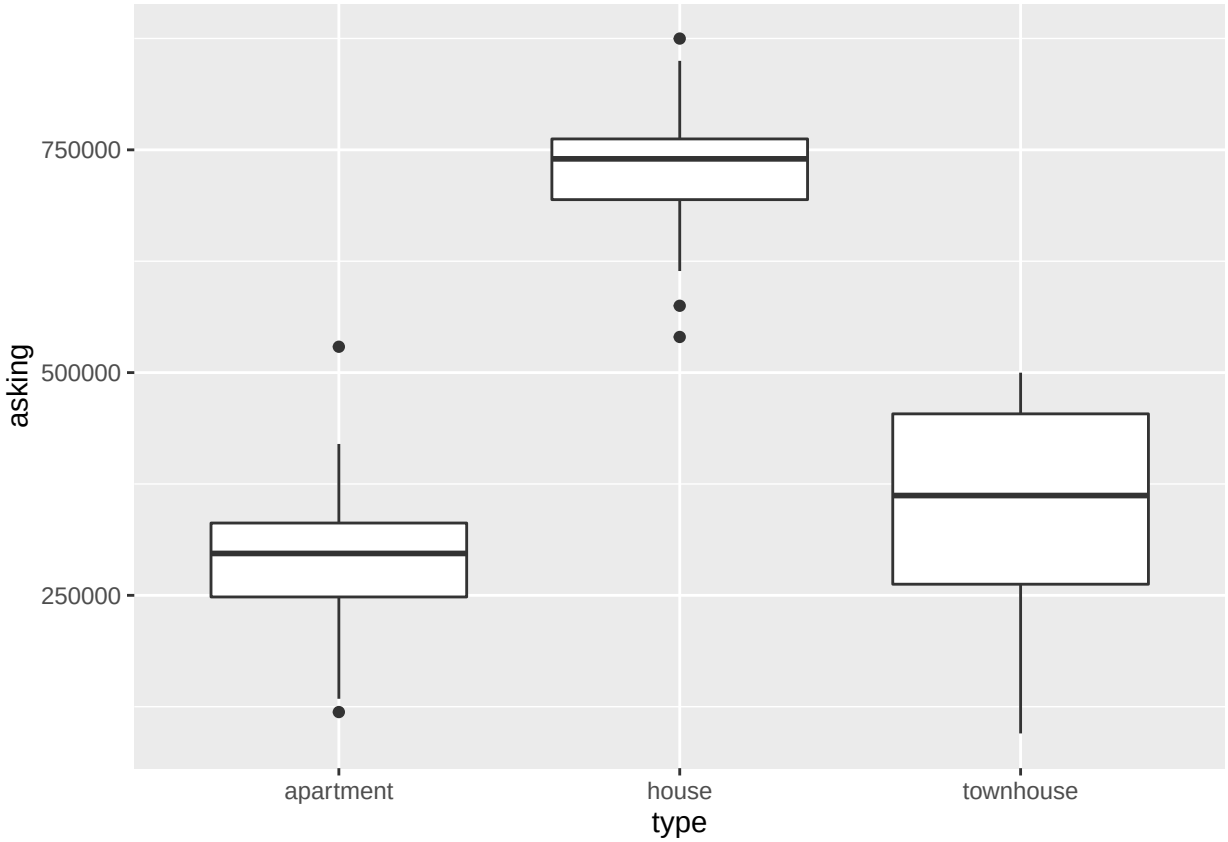
```
##
## Call:
## lm(formula = asking ~ bedrooms + bathrooms, data = houses)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -425042  -64976   16301   76715  314002
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    70741      55112   1.284    0.205
## bedrooms      114814      18760   6.120 1.33e-07 ***
## bathrooms      19615      32398   0.605    0.548
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 149400 on 51 degrees of freedom
## Multiple R-squared:  0.5806, Adjusted R-squared:  0.5642
## F-statistic:  35.3 on 2 and 51 DF,  p-value: 2.381e-10
```

As seen from the figure above, the most noticable feature is the exceeding low p-value of the bedrooms factor. This value is 1.33e-07 which is very impressive considering the minimum acceptable value is at $\alpha = 0.05$. From a glance, the model tells us that the most significant factor affecting the listing price of a property is the number of bedrooms that it has at an average price increase of $$$114,000 CAD per additional bedroom. From a practical standpoint, it is very reasonable to have a house be worth more simply because more people can live in it as the primary objective for a home is to provide shelter. The model also displays bathrooms as having a very low impact on housing prices at around $$$19,615 CAD per bathroom which sounds like a steep cost. However, when compared to the price of the entire property, it is clear that the number of bathrooms is not a major factor in predicting the asking price. This claim is back up by the fact that the p-value of bathrooms as a predictor is around 0.548 which is much larger than $\alpha = 0.05$. The intercept in this case could be the amenities that come with purchasing such a property. However, the p-value is much too high at 0.205 for it to count for anything. One concerning observation is that the R squared statistic is only at round 56% of the data.

For further analysis, the three different types of properties will be split up into their respective categories.

Boxplot of the asking prices of the three different types of propertys:

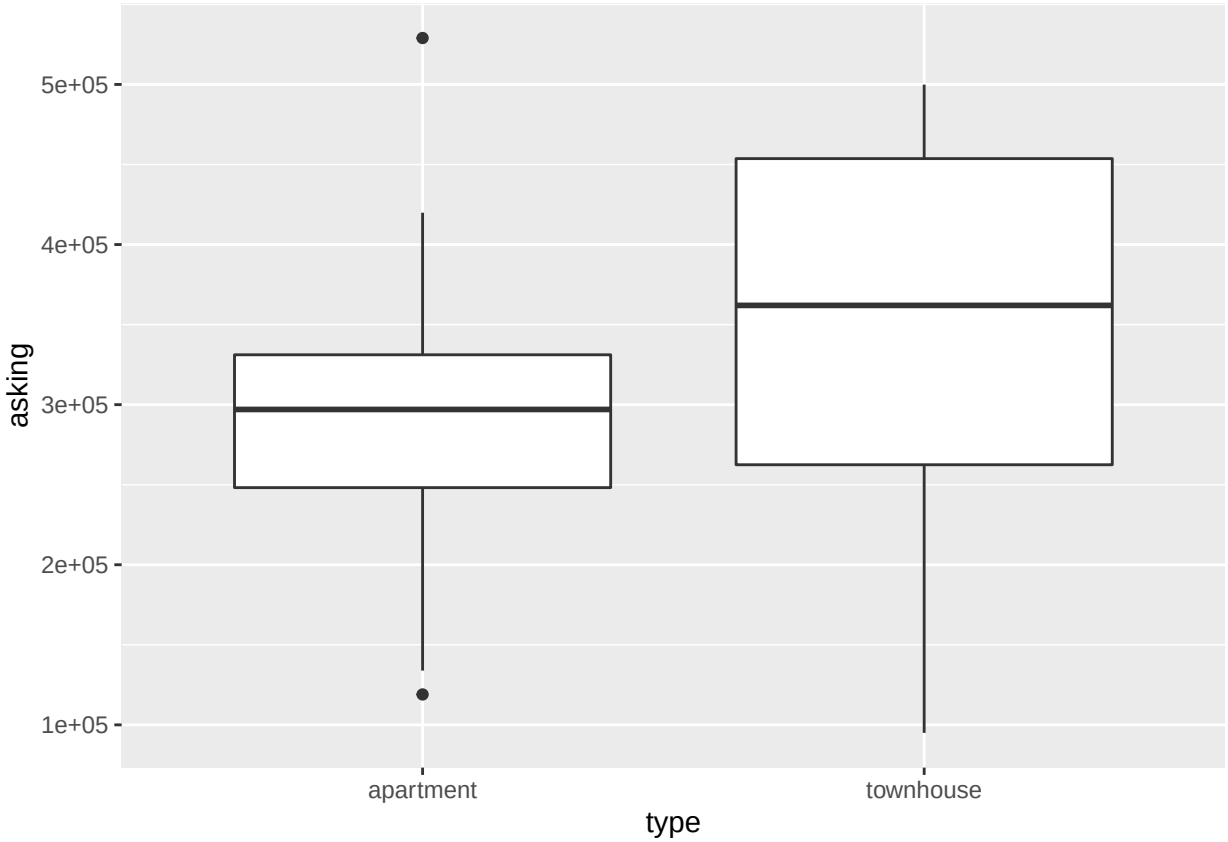
```
ggplot(houses, aes(x = type, y = asking)) + geom_boxplot()
```



Clearly, houses are more expensive than apartments and townhouses with the median house costing around $$$740,00 CAD. It is hard to distinguish the difference between apartments and townhouses from this graph alone. Let us take a closer look at only the apartment and townhouse boxplots.

Boxplot of asking prces for townhouses and apartments only:

```
no_house<-houses[!(houses$type=="house"),]
ggplot(no_house, aes(x = type, y = asking)) + geom_boxplot()
```



The boxplot displays that the average townhouse is generally around $$$6000 CAD more expensive than apartments but the townhouse data seems to have a bigger σ^2 and to be skewed to the left. Now, the data will be split up into their respective type categories.

Shapiro-Wilk test for normality

```
house_only <- houses [houses$type=="house",]
apt_only <- houses [houses$type=="apartment",]
town_only<- houses [houses$type=="townhouse",]

shapiro.test(houses$asking)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  houses$asking
## W = 0.91564, p-value = 0.001026
```

```
shapiro.test(house_only$asking)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_only$asking
## W = 0.9588, p-value = 0.5201
```

```
shapiro.test(apt_only$asking)
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  apt_only$asking
## W = 0.94853, p-value = 0.2517
```

```
shapiro.test(town_only$asking)
```

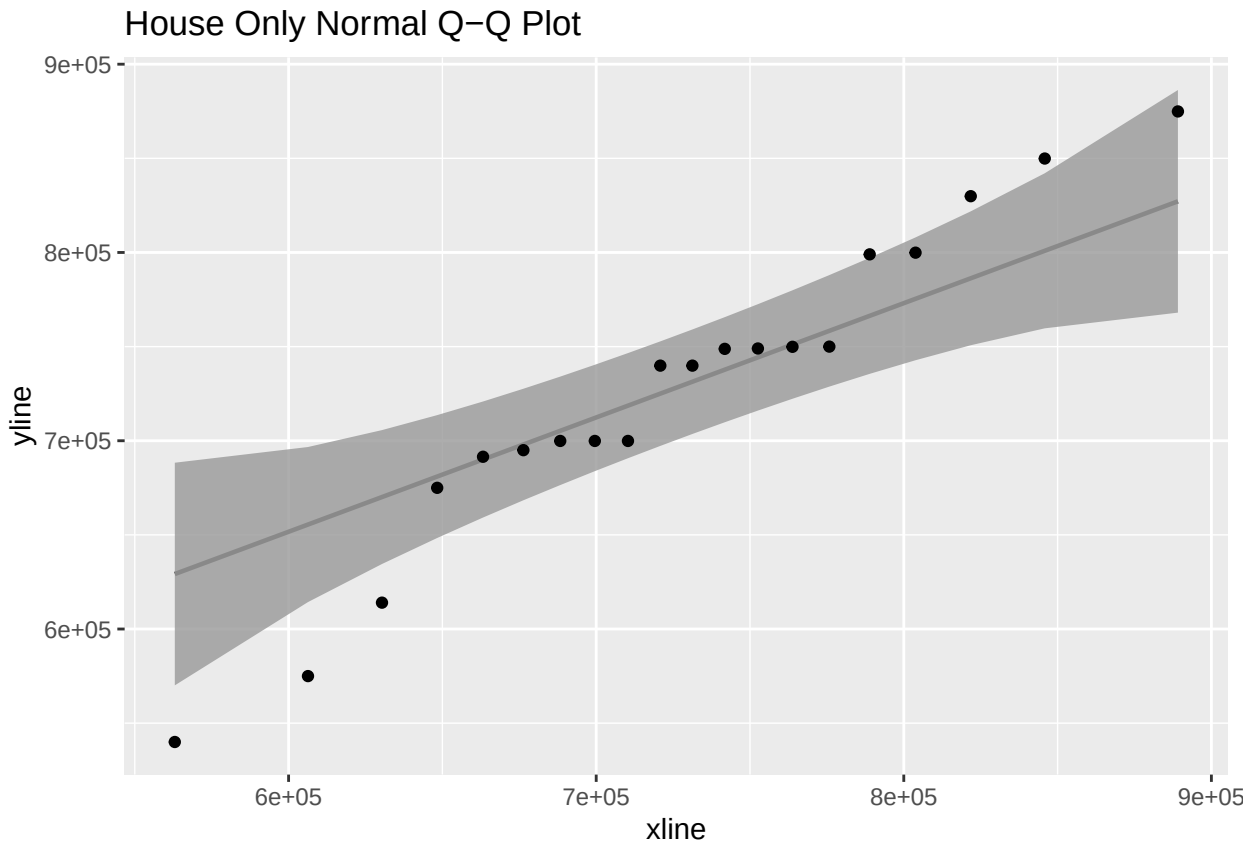
```
##
##  Shapiro-Wilk normality test
##
## data:  town_only$asking
## W = 0.93171, p-value = 0.465
```

The tests that are shown above are called the Shapiro-Wilk test which are to believed more effective at predicting the normality of a data set than observation by eye. These tests have two outputs: W and p-value. The W value stands for how much of the data falls under a normal distrbution where 1 means it has zero deviance from a normal distribution. The null hypothesis of this test is that the data is normal with the critical value being at 0.05. As seen from the test above, while we can confirm with a test that the whole of the data itself should be normal with a W value at around 91% and a p-value of 0.001, the same could not be said for the individial types of housing with the lowest p-value amongst them being 0.2517.

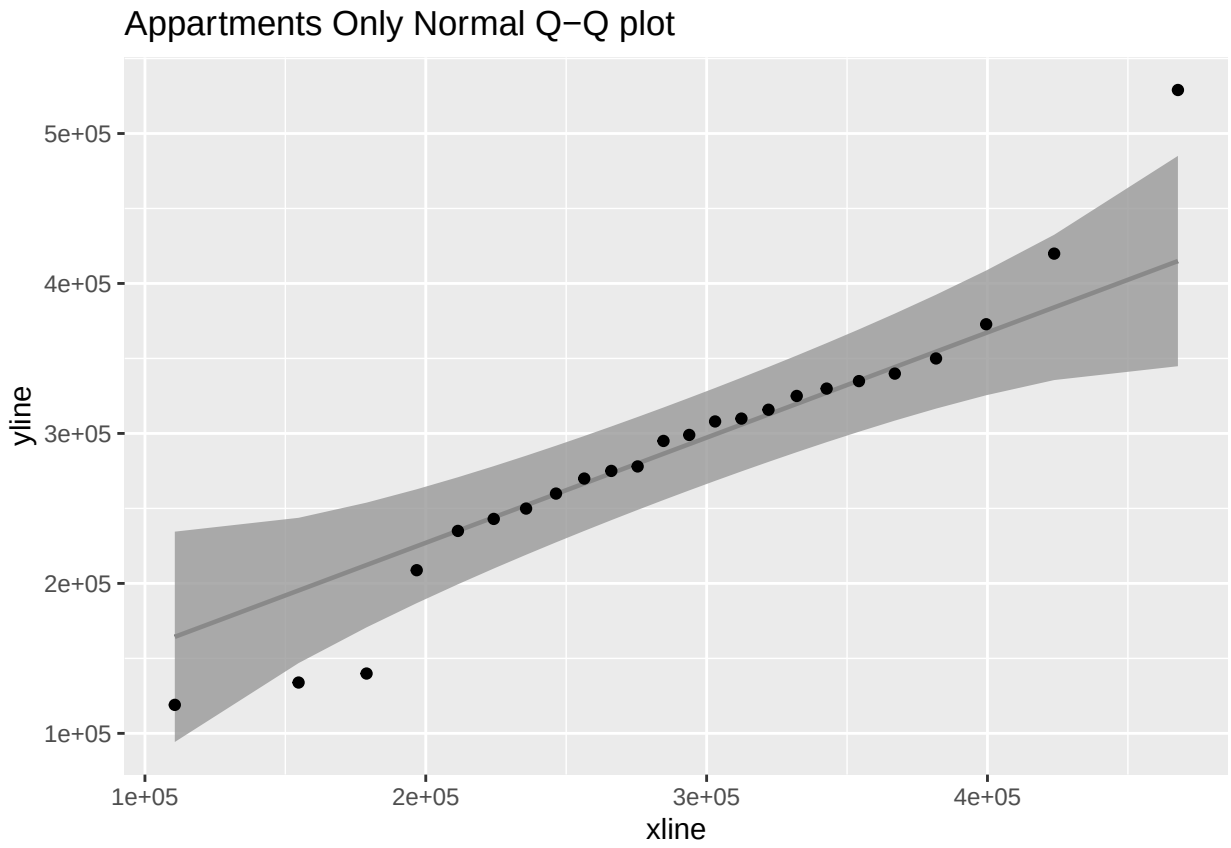
It maybe that relying on a test may not be the wisest course of action. Perhaps a visual inference may help.

Q-Q plots for individual property types:

```
housescatter <- ggplot(house_only, aes(sample = asking)) + stat_qq_line() + stat_qq_band() + stat_qq_po
aptscatter <- ggplot(apt_only, aes(sample = asking)) + stat_qq_line() + stat_qq_band() + stat_qq_point()
townscatter <- ggplot(town_only, aes(sample = asking)) + stat_qq_line() + stat_qq_band() + stat_qq_poin

print(housescatter + ggtitle("House Only Normal Q-Q Plot"))
```




```
print(aptscatter + ggtitle("Appartments Only Normal Q-Q plot"))
```

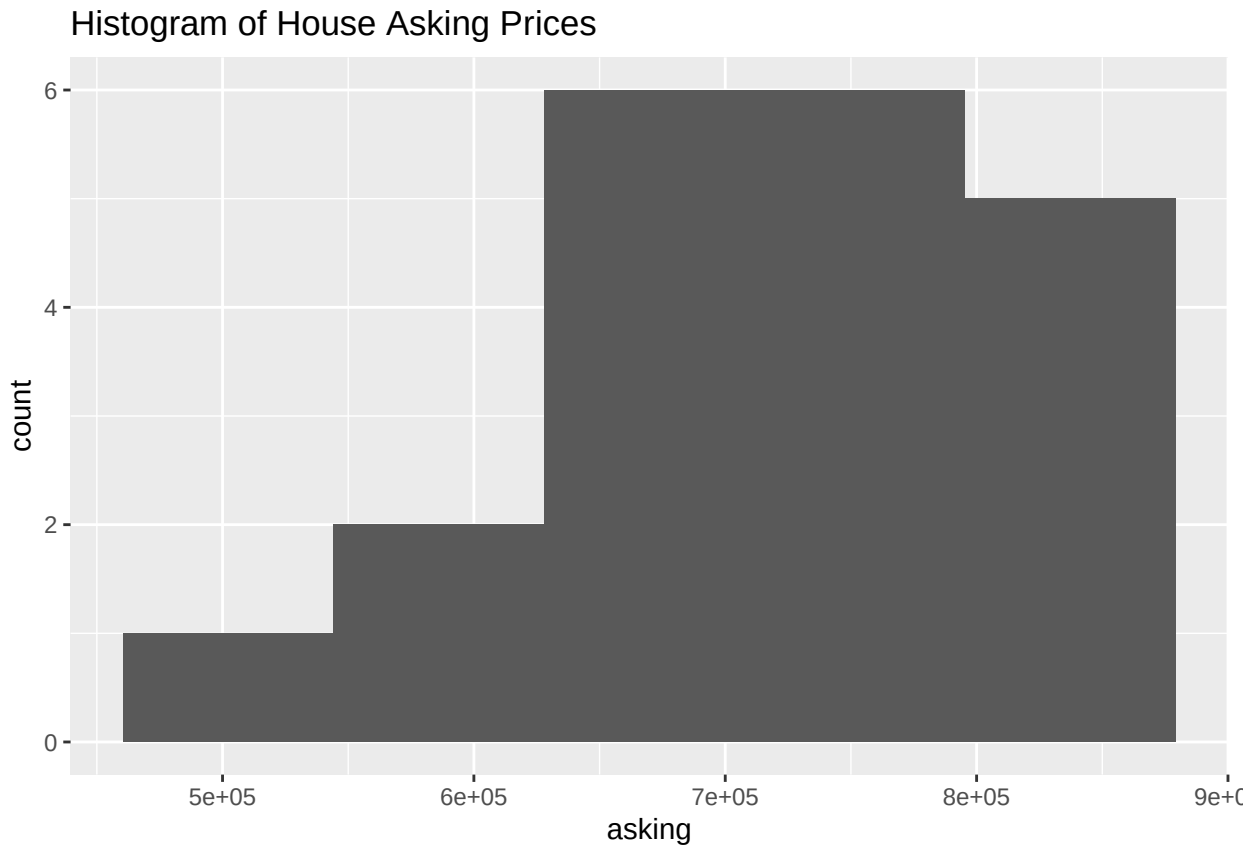



```
print(townscatter + ggtitle("Townhouse Only Normal Q-Q plot"))
```

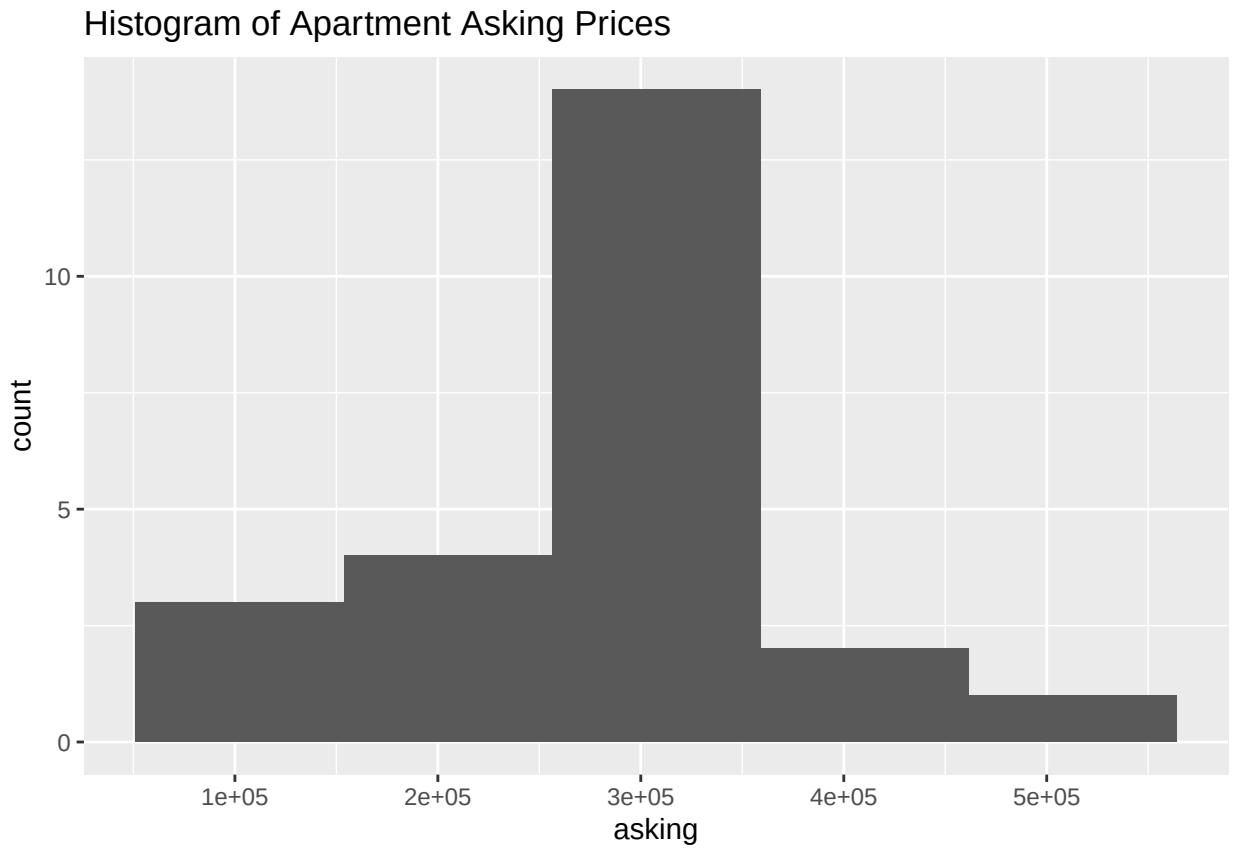
T With these normal Q-Q plot of these different types of housing avaiable, we can make some observations. Asking prices for house has 5 points outside of the 95% confidence band which is 25% of the data. Apartment has 3 points outside the band which is 12.5% of the data. Finally, townhouses seem to have no points outside the confidence band. With these scatterplots, we can see that an issue arrises. There may be fundamentally to little samples of individual types of housing to do any kind of significant statistical anaylsis. This is reflected in the Shapiro-Wilks test where there is a high W value but a low p-value.

Histograms for different types of property:

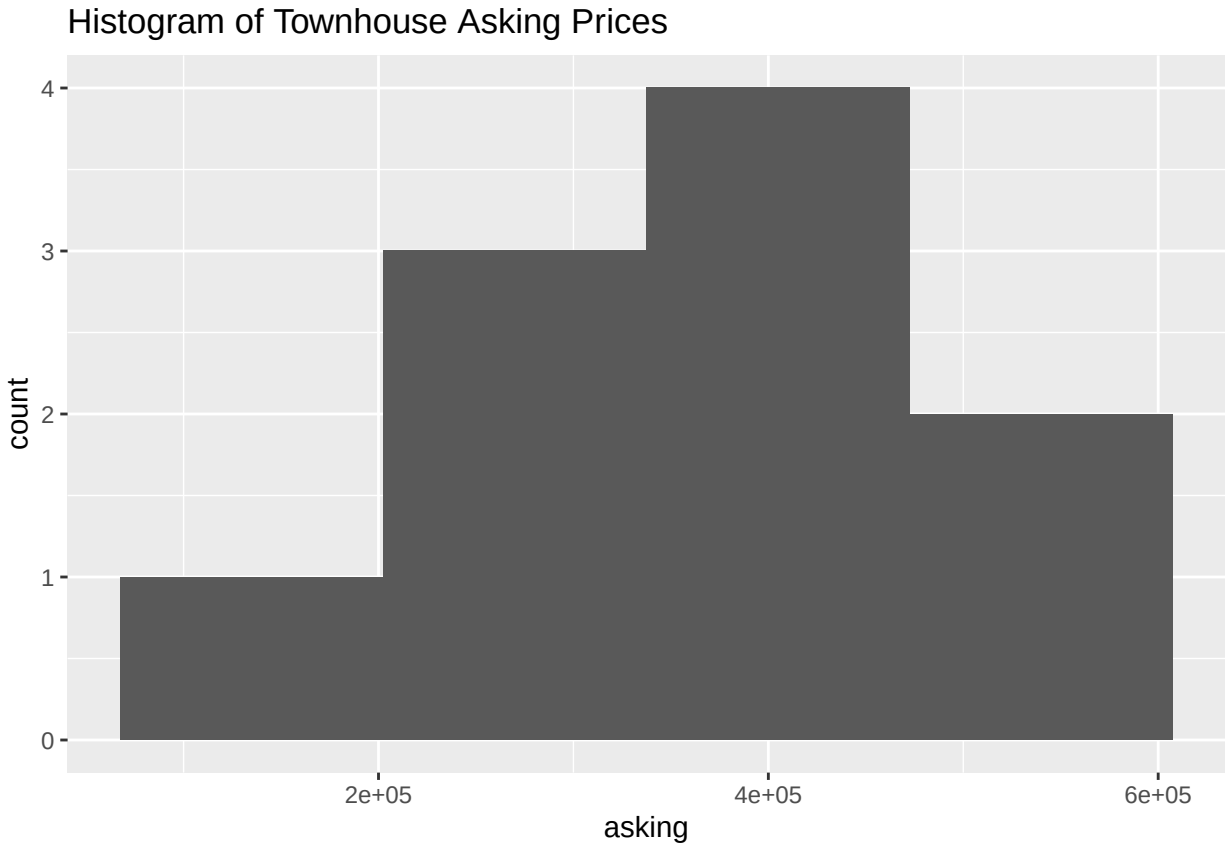
```
house_histo = ggplot(house_only, aes(x = asking)) + geom_histogram(bins = 5)
apt_histo = ggplot(apt_only, aes(x = asking)) + geom_histogram(bins = 5)
town_histo = ggplot(town_only, aes(x = asking)) + geom_histogram(bins = 4)

print(house_histo + ggtitle("Histogram of House Asking Prices"))
```




```
print(apt_histo + ggtitle("Histogram of Apartment Asking Prices"))
```




```
print(town_histo + ggtitle("Histogram of Townhouse Asking Prices"))
```



With the histrogram, we can see more clearly that the data for each of these types of properties are fundamentally skewed. Anyhow, let us try to do the same regressional anaylsis that was done for the the data as a whole but for individual types of properties. While we're at it, the residual plots for the different types of data will be analyzed as well.

Residual plots of the 3 types of housing

```
house_only_reg = lm(asking ~ bedrooms + bathrooms, data = house_only)
apt_only_reg = lm(asking ~ bedrooms + bathrooms, data = apt_only)
town_only_reg = lm(asking ~ bedrooms + bathrooms, data = town_only)
summary(house_only_reg)
```

```
##
## Call:
## lm(formula = asking ~ bedrooms + bathrooms, data = house_only)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -147062  -56762    8624   49283  118612
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   567060      93346   6.075 1.24e-05 ***
## bedrooms       22107      23490   0.941    0.360
## bathrooms      26841      21183   1.267    0.222
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 80100 on 17 degrees of freedom
## Multiple R-squared:  0.2121, Adjusted R-squared:  0.1194
## F-statistic: 2.289 on 2 and 17 DF,  p-value: 0.1318
```

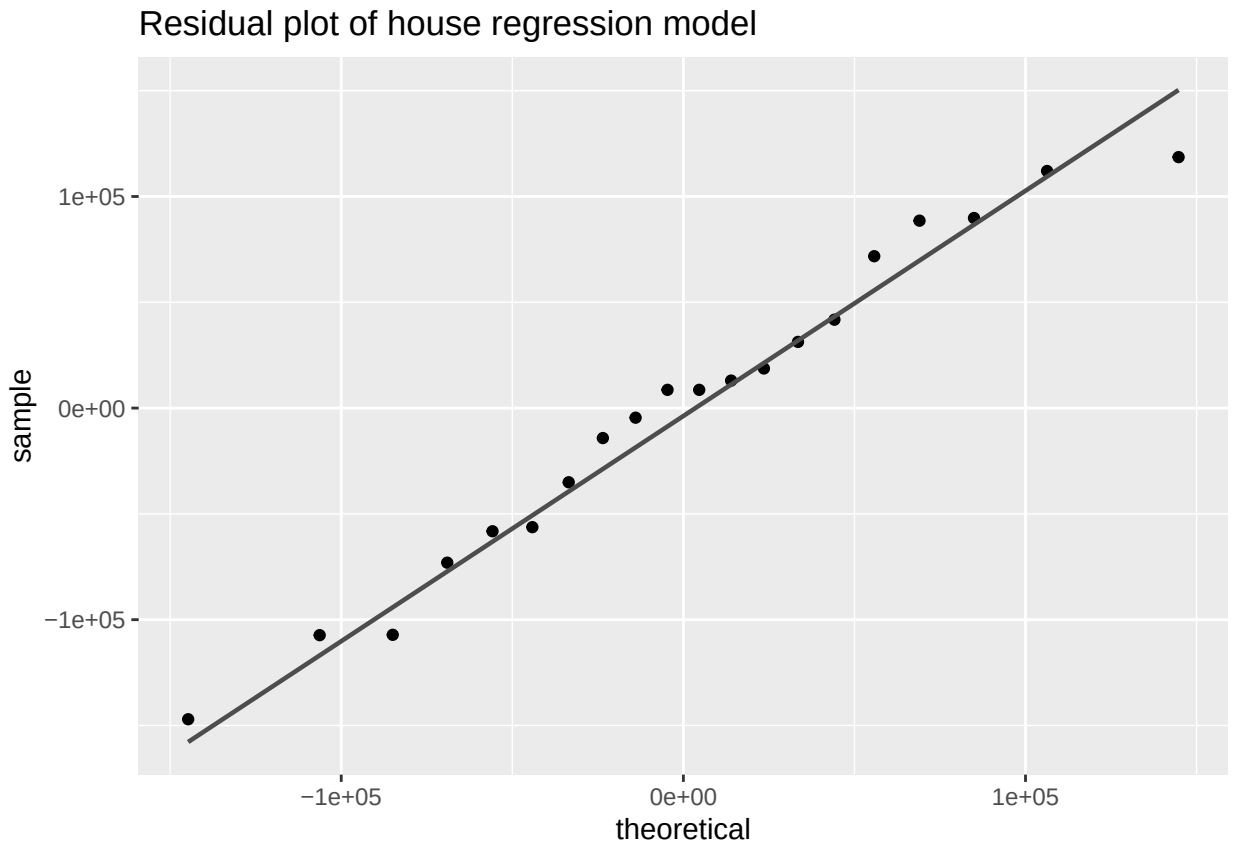
```
summary(apt_only_reg)
```

```
##
## Call:
## lm(formula = asking ~ bedrooms + bathrooms, data = apt_only)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -201822  -15557    7807   35479  209315
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   201357      58278   3.455  0.00237 **
## bedrooms       -2274      29509  -0.077  0.93930
## bathrooms      63144      43987   1.436  0.16587
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 87870 on 21 degrees of freedom
## Multiple R-squared:  0.1212, Adjusted R-squared:  0.03749
## F-statistic: 1.448 on 2 and 21 DF,  p-value: 0.2576
```

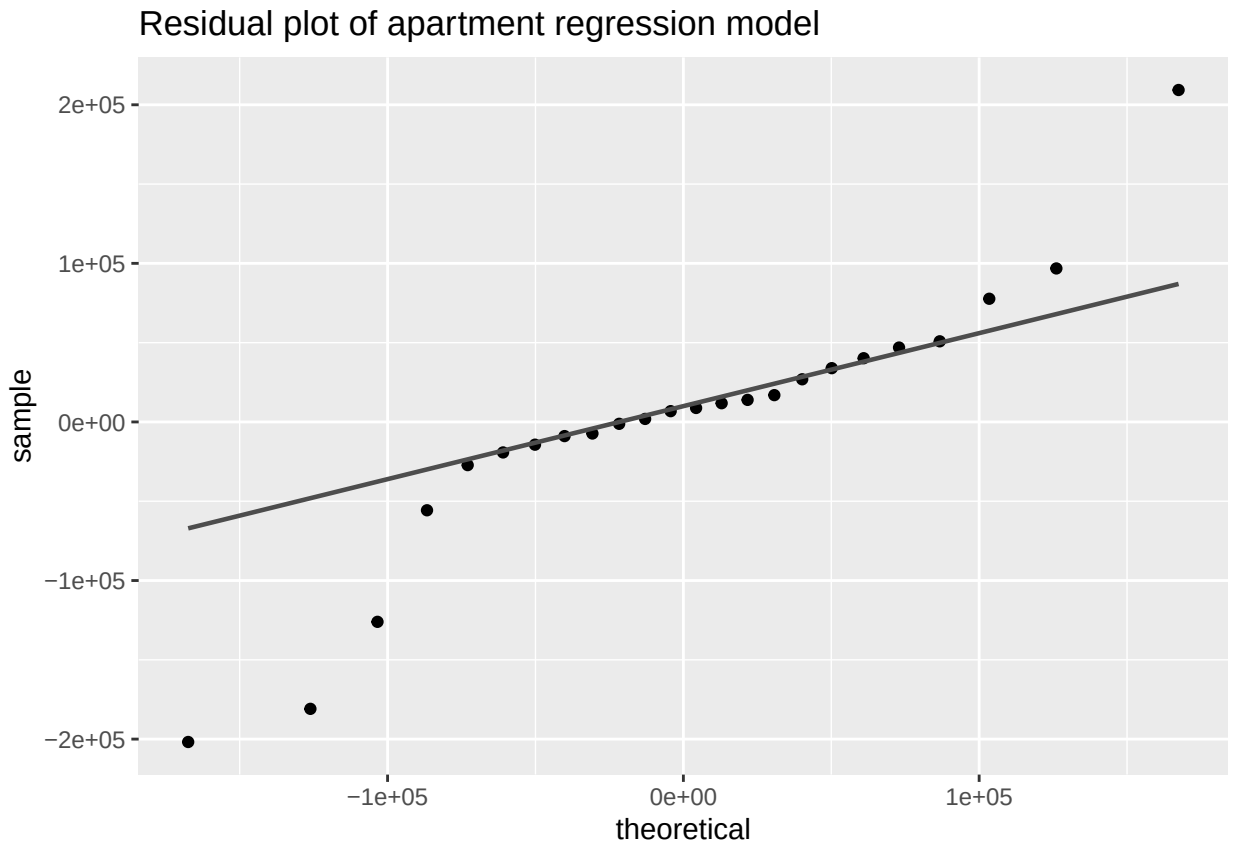
```
summary(town_only_reg)
```

```
##
## Call:
## lm(formula = asking ~ bedrooms + bathrooms, data = town_only)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -124618  -73358  -34571   90623  155704
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)     7476     193838   0.039    0.970
## bedrooms       29711      55169   0.539    0.607
## bathrooms     113794      75282   1.512    0.174
##
## Residual standard error: 117300 on 7 degrees of freedom
## Multiple R-squared:  0.3557, Adjusted R-squared:  0.1716
## F-statistic: 1.932 on 2 and 7 DF,  p-value: 0.2147
```

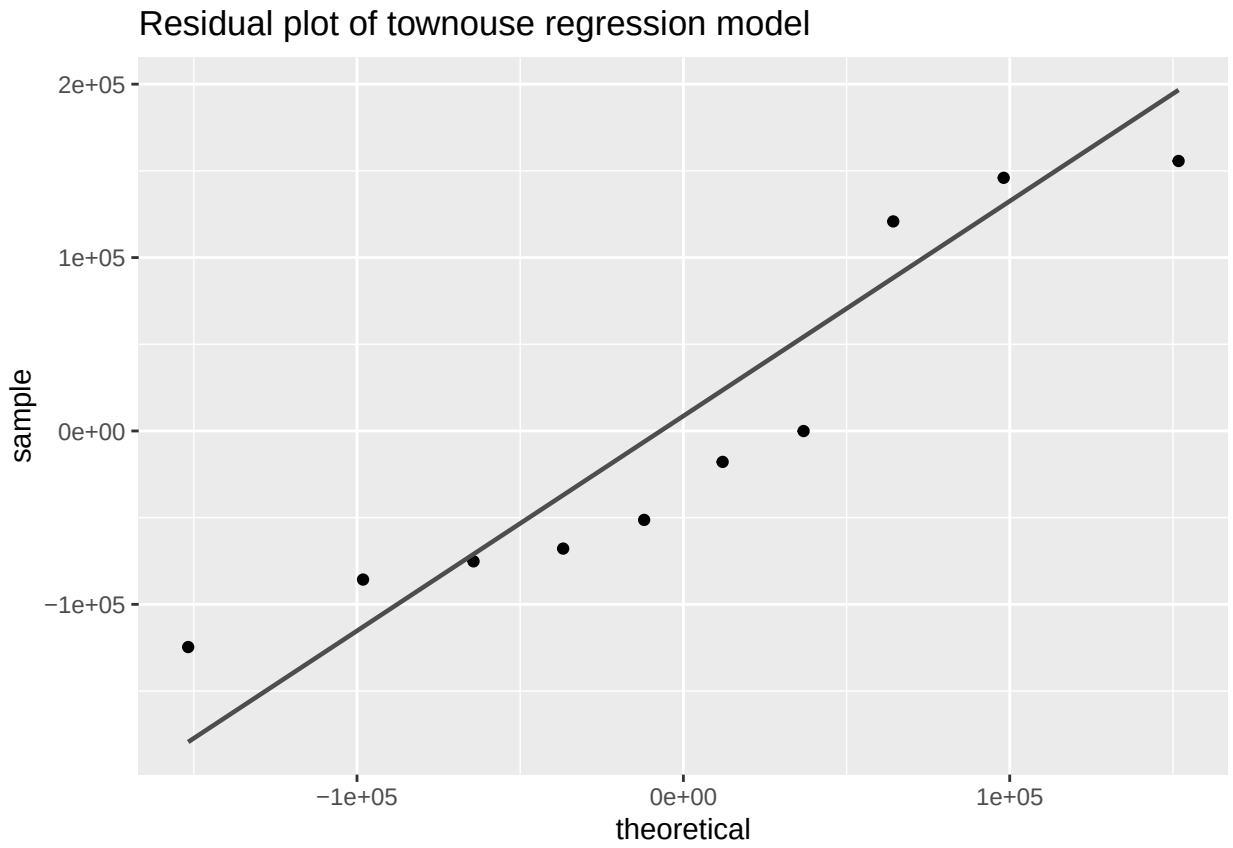
```
house_res = ggplot(house_only_reg, aes(sample = .resid)) + stat_qq_point() + stat_qq_line()
apt_res = ggplot(apt_only_reg, aes(sample = .resid)) + stat_qq_point() + stat_qq_line()
town_res = ggplot(town_only_reg, aes(sample = .resid)) + stat_qq_point() + stat_qq_line()

house_resfit = ggplot(house_only_reg, aes(y = .resid, x = .fitted)) + geom_point()
apt_resfit = ggplot(apt_only_reg, aes(y = .resid, x = .fitted)) + geom_point()
town_resfit = ggplot(town_only_reg, aes(y = .resid, x = .fitted)) + geom_point()

print(house_res + ggtitle("Residual plot of house regression model"))
```



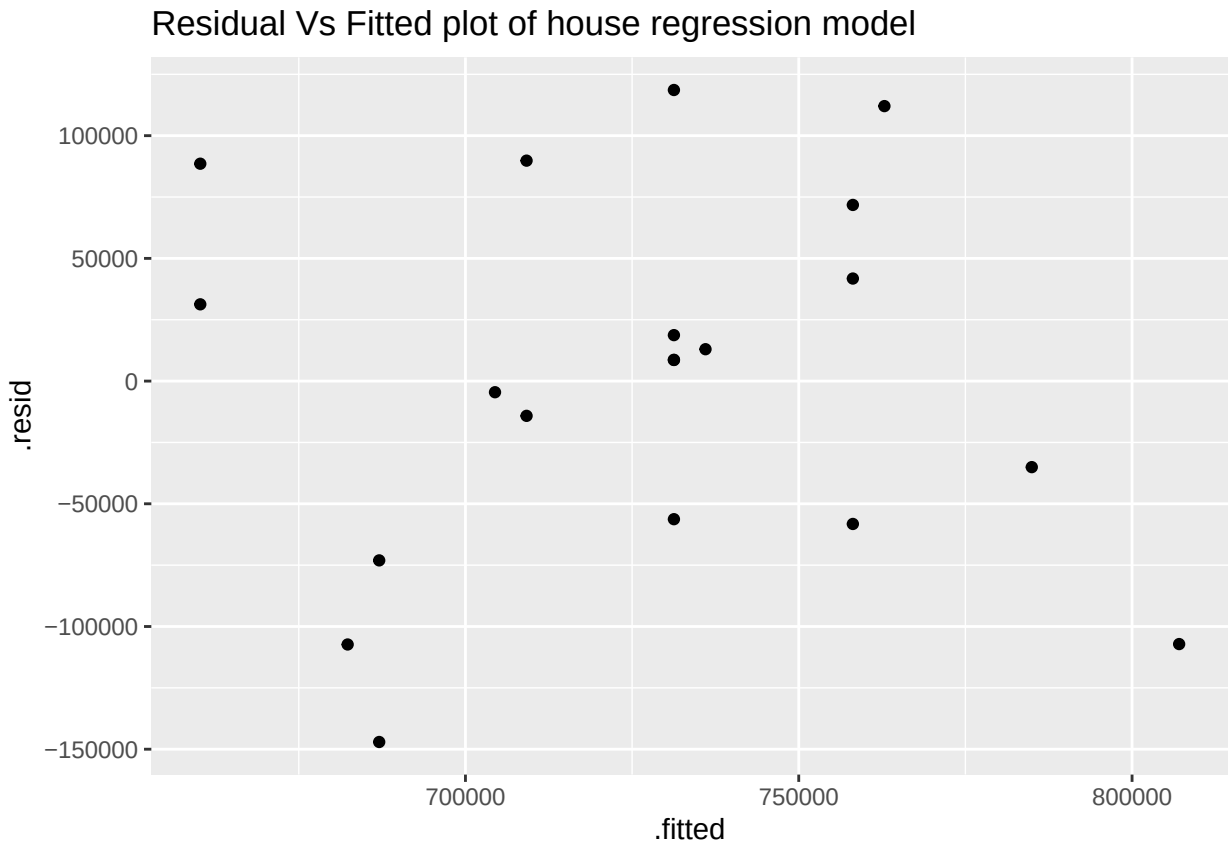

```
print(apt_res + ggtitle("Residual plot of apartment regression model"))
```



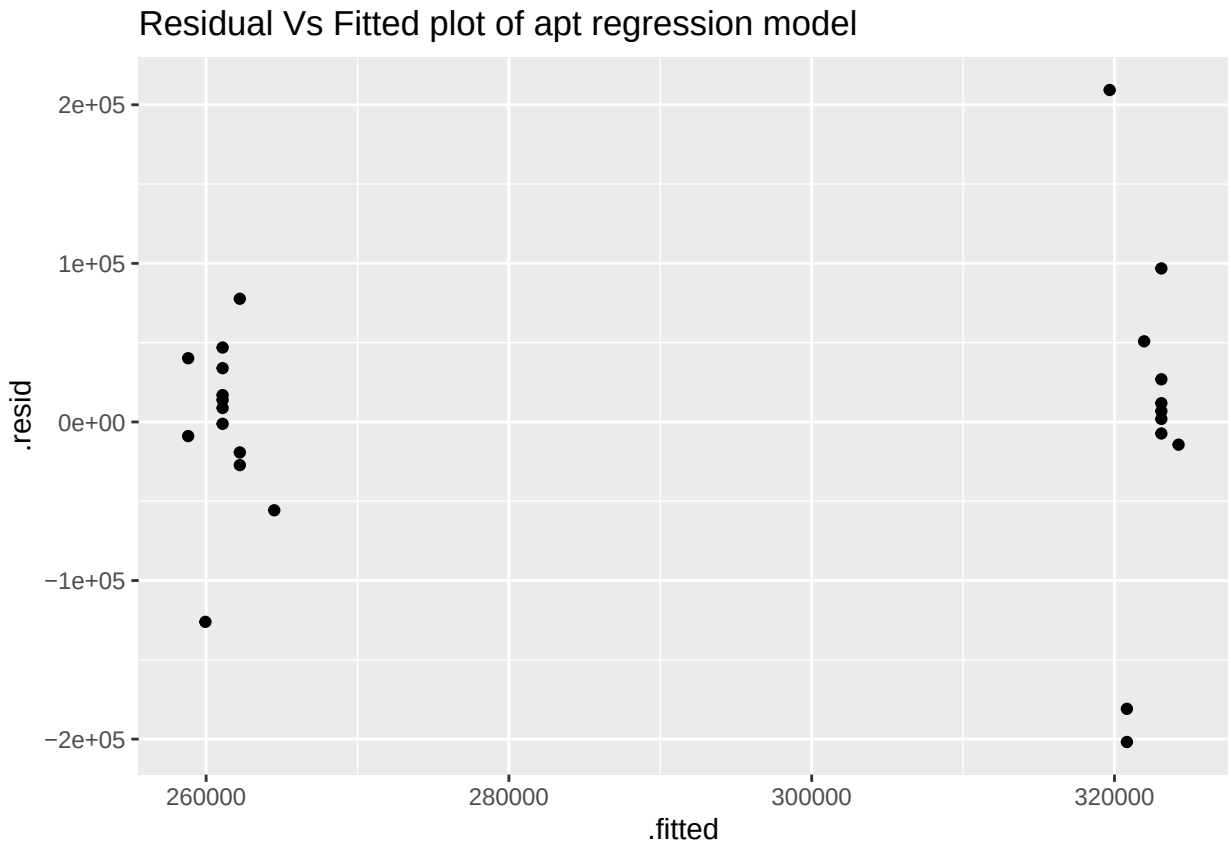

```
print(town_res + ggtitle("Residual plot of townouse regression model"))
```



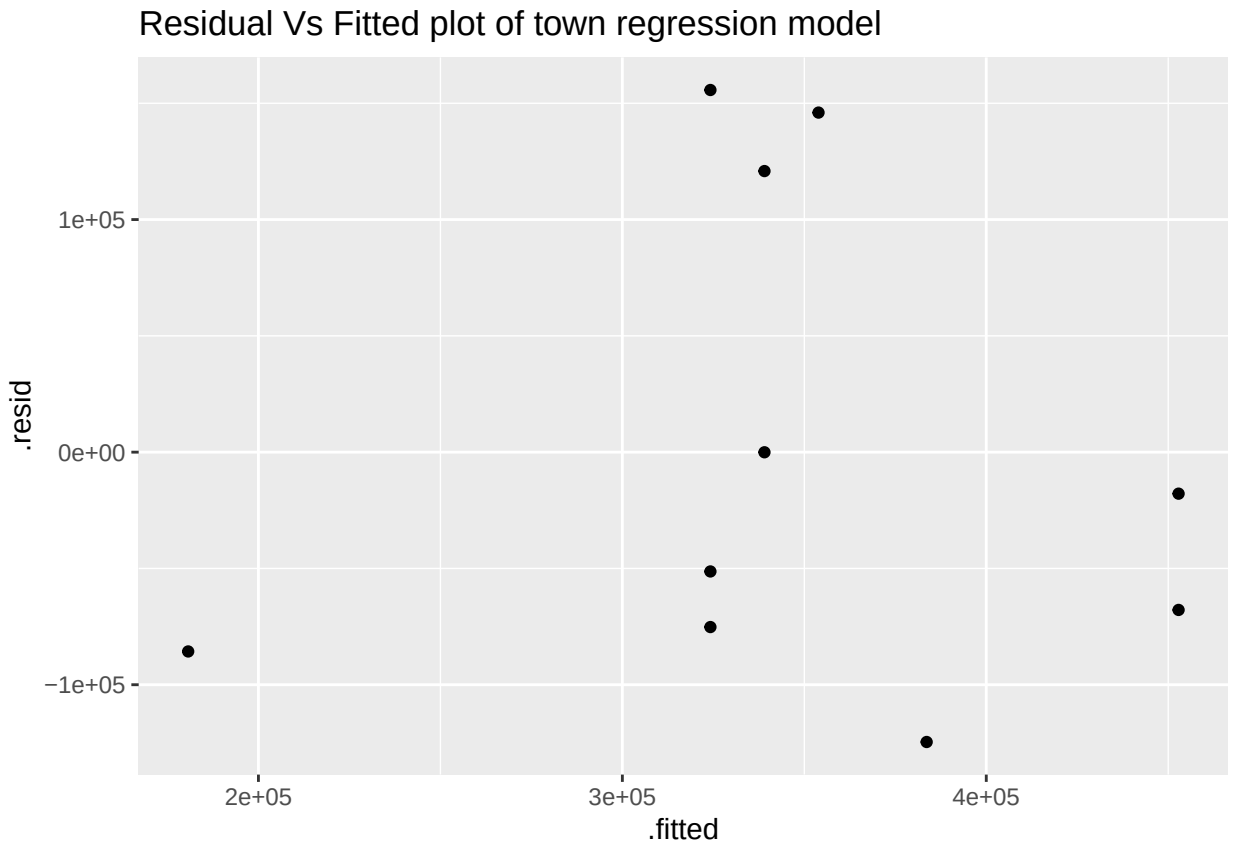

```
print(house_resfit + ggtitle("Residual Vs Fitted plot of house regression model"))
```



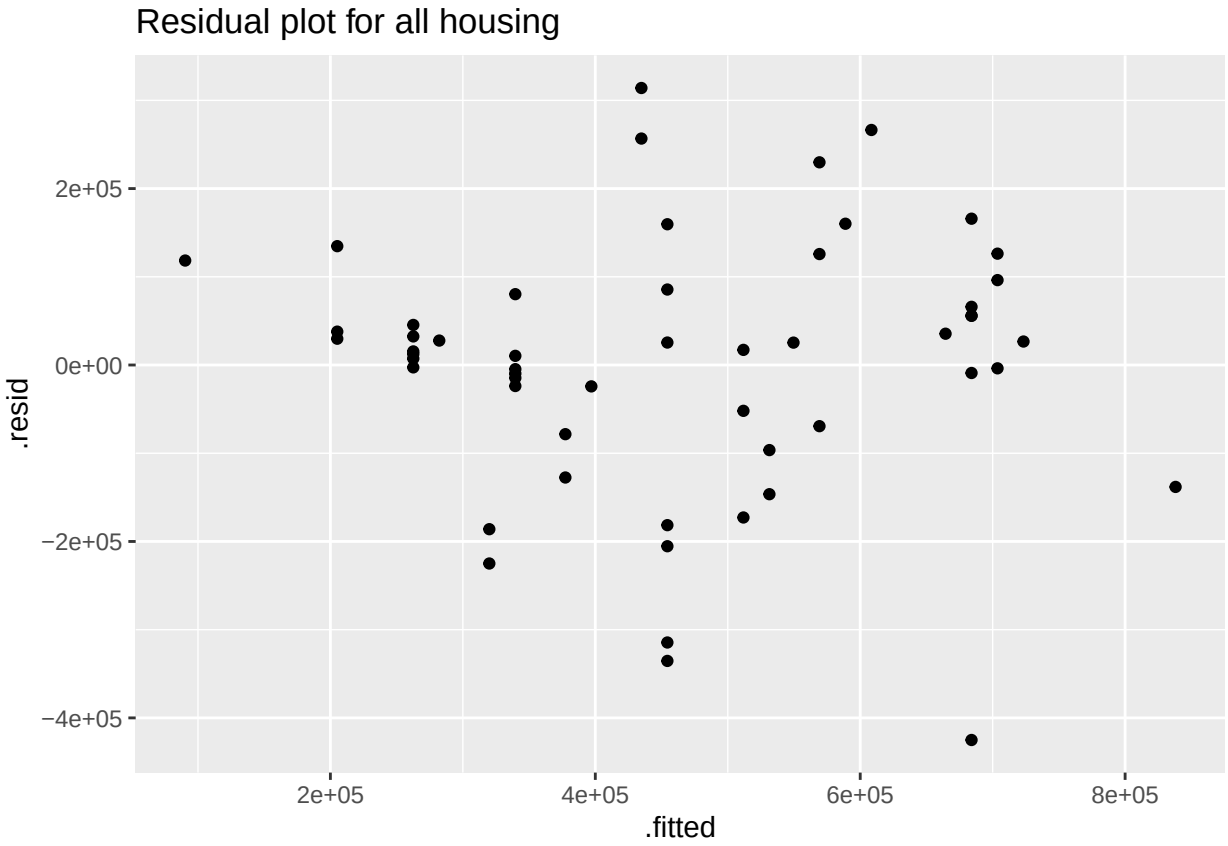

```
print(apt_resfit + ggtitle("Residual Vs Fitted plot of apt regression model"))
```




```
print(town_resfit + ggtitle("Residual Vs Fitted plot of town regression model"))
```




```
total_res = ggplot(total_reg_model, aes (y = .resid, x = .fitted)) + geom_point()
print(total_res + ggtitle("Residual plot for all housing"))
```



When looking at the regressional models for each individual type of property, while the data as a whole seems to be relatively normal from its residual plot, the flaws of splitting the data become noticeable. The first consequence of the flaw is that there is simply not enough data. This is reflected by the scarcity of points on the residual plots. Another major consequence is that the data may stray away from normality. This is visible when observing the residual plots for apartments and townhouses. Some points trail off on the ends of the fitted line creating tails or even may not follow the line very strictly at all. The combined effects of these two phenomenon are reflected in the regressional models where neither bedrooms nor bathrooms seem to have any significant effects with their low estimations and high p-values and the fact that the highest R squared value for any of these individual models tops off at around 0.26 with the lowest one as low as 0.037. Also supporting this argument, the residual vs fitted plots of some of these models are absolutely terrible at following the theoretical line with the apartments being the worst offender by being grouped to two ends of the graph. To get a closer look at how bathrooms and bedrooms are affecting the price, we need to look at the residual plots of the bedroom factor and the bathroom factor for each type of housing.

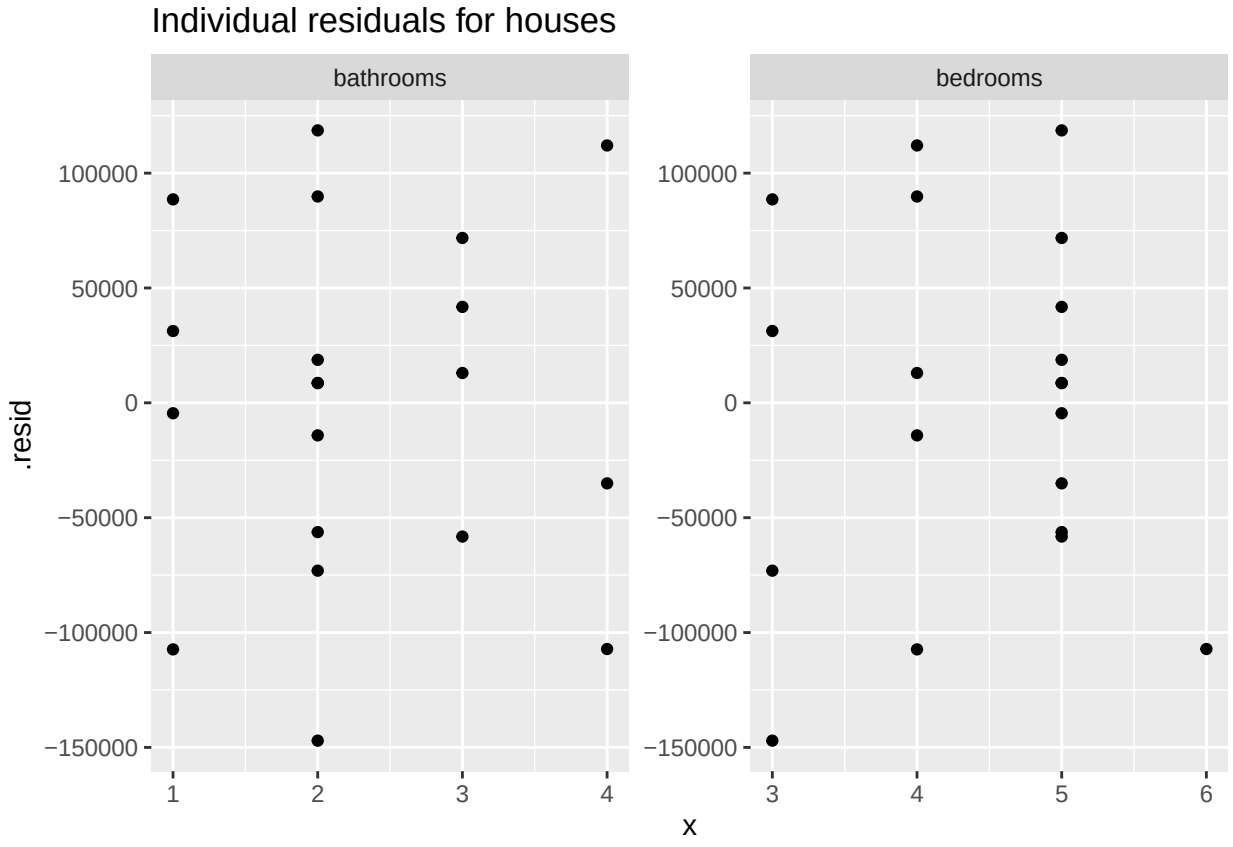
Residuals for individual betas

```
house_only_reg %>% augment(house_only) -> h2
h2 %>%
  gather(xname, x, c(bedrooms:bathrooms)) %>%
  ggplot(aes(x = x, y = .resid)) + geom_point() + facet_wrap(~xname, scales = "free") ->
  house_only_betas

apt_only_reg %>% augment(apt_only) -> a2
a2 %>%
  gather(xname, x, c(bedrooms:bathrooms)) %>%
  ggplot(aes(x = x, y = .resid)) + geom_point() + facet_wrap(~xname, scales = "free") ->
  apt_only_betas
```

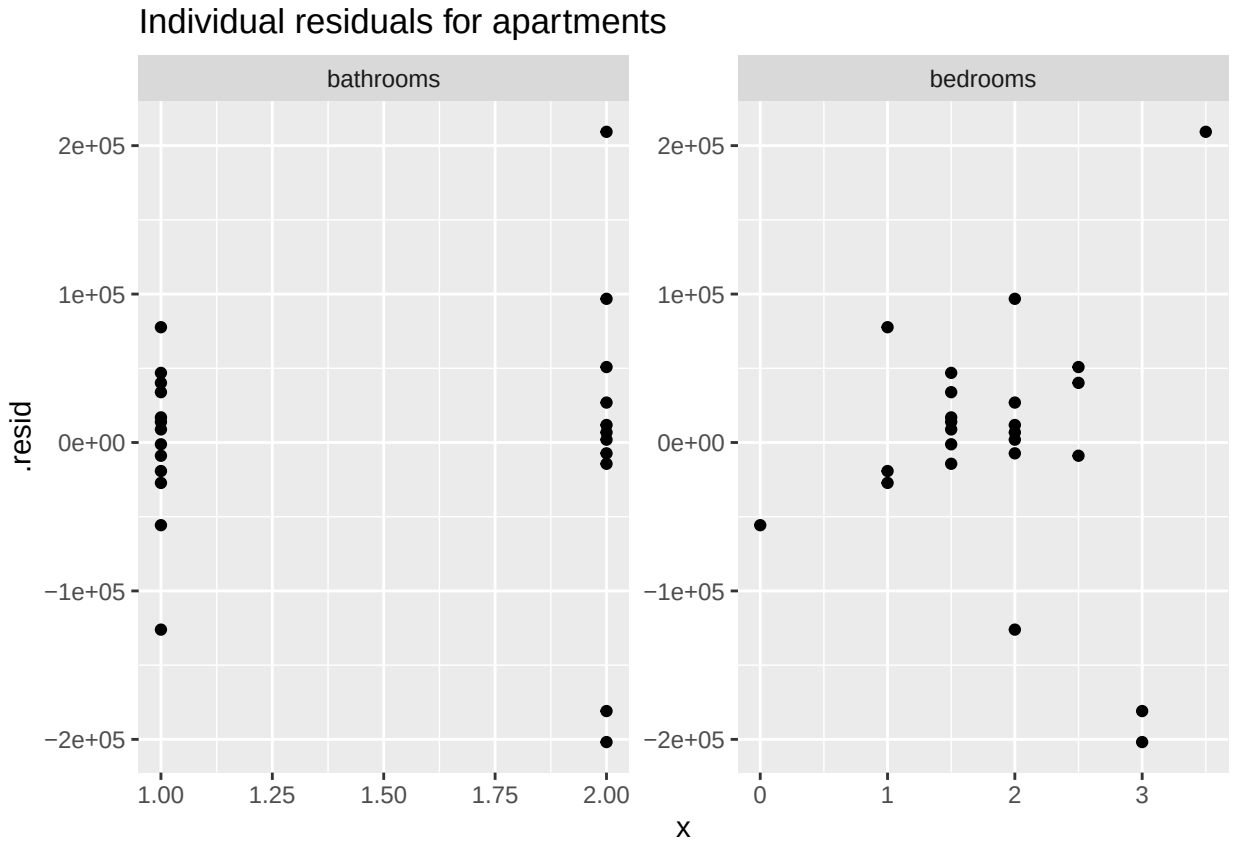
```
town_only_reg %>% augment(house_only) -> t2
```

```
## Warning in data.frame(..., check.names = FALSE): row names were found from a
## short variable and have been discarded
```

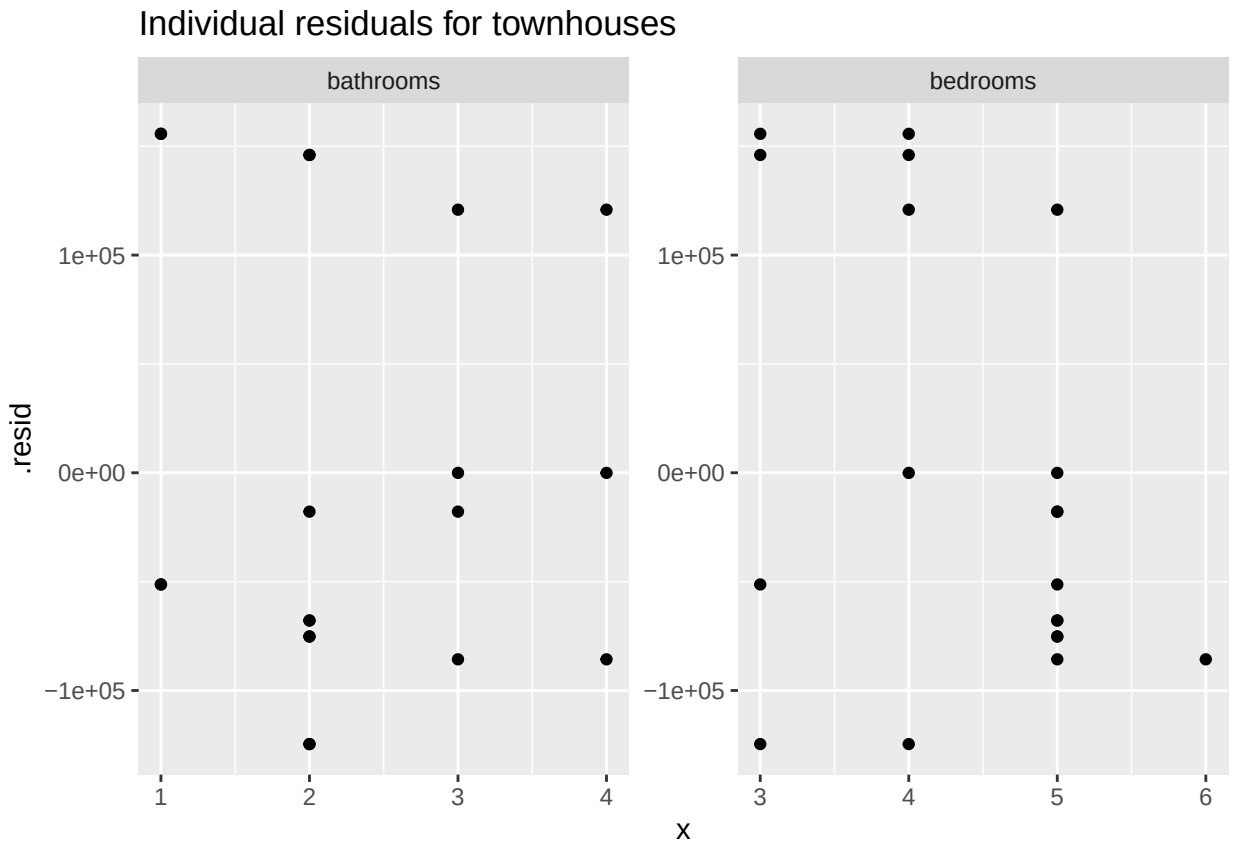
```
t2 %>%
  gather(xname, x, c(bedrooms:bathrooms)) %>%
  ggplot(aes(x = x, y = .resid)) + geom_point() + facet_wrap(~xname, scales = "free") ->
  town_only_betas

total_reg_model %>% augment(houses) -> tot2
tot2 %>%
  gather(xname, x, c(bedrooms:bathrooms)) %>%
  ggplot(aes(x = x, y = .resid)) + geom_point() + facet_wrap(~xname, scales = "free") ->
  total_betas

print(house_only_betas + ggtitle("Individual residuals for houses"))
```



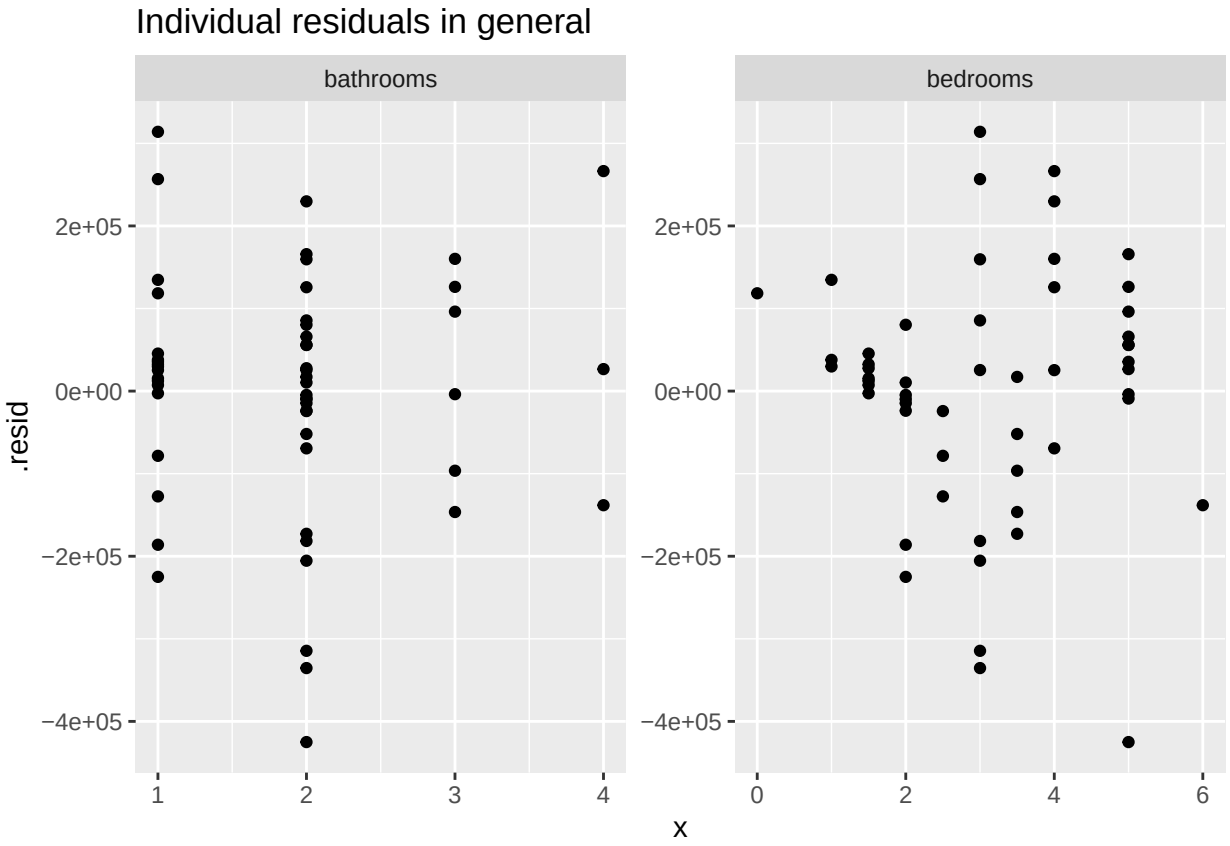

```
print(apt_only_betas + ggtitle("Individual residuals for apartments"))
```




```
print(town_only_betas + ggtitle("Individual residuals for townhouses"))
```




```
print(total_betas + ggtitle("Individual residuals in general"))
```



When making observations on these individual residuals for bedrooms and bathrooms, the foremost comment is that these charts look rather stilted. This is due to the fact that our predictor variable is not continuous and is discrete as the number of bedrooms and bathrooms are inherently categorical (e.g. you cannot have 0.151235613 of a bathroom). While we could do a spearman regressional model instead, we can just take what we already know about residuals and apply them here. The main goal of residual plots is to make sure the plot looks random. When looking at the residual plots, the main residuals that do not look random are that of apartments and of townhouses. For apartments, it seems the residual points for bathrooms are clustered on the left and right side and the bedroom residual plot has most of its points clustered about the middle. For the bathrooms of townhouses, there seems to be a consistent line of plots far away from the other ones.

From all of the residual plots, graphs, and regressional models, the one issue with overwhelming evidence is the overwheling lack of evidence. To remedy this issue, our team has went out and gathered 150 new samples of data from the same site using the same techniques in the same area as the old data sample. But first, a brief conclusion of the data that was provided.

Conclusion for 2017 Housing Data

While theres hardly enough samples for any significant statistical inference, there is still inference to be made. In general, properties with more bedrooms can expect a higher property value at around $$$115,000 CAD per bedroom (from the regressional model of all types of housing). The number of bathrooms may make a difference of around fifteen thousand dollars but it is very hard to say with the data that is given (see regressional models having relatively low estimates and high p-value). Houses are worth the most with the average house being $$$740,000 CAD, apartments and townhouses being worth similar amounts but townhouses have a slight edge over the apartments with apartments being asked at around $$$350,000 CAD and townhouses being asked around $$$400,000 CAD. The reasoning on why houses being worth so much more with the average amount of bedrooms and bathrooms ranging from 3 to 5 (see residual plot of bedrooms in general) may suggest a lurking variable out there that was not recorded when the data was being gathered. An example of such a variable would be something like the prestige of owning a certain type of house. Perhaps

owning a house may be much more luxurious to sellers that they would charge the buyer an exorbitant amount of money simply for the prestige of having a house as opposed to an apartment. Another factor could be the total square inch of a property. A house or a townhouse may be significanly larger in terms of surface area that is available for use in a home.

Analysis of 2019 Data

As mentioned before, the data gathered for the new data is in the same way the old data was gathered. Everything remained consistent with the old method of data gathering. The same area was picked, the same method of going on https://www.realtor.ca was used, and the same variables were recorded. The only difference this time is that the listings are recorded in the year of 2019.

```
new_data <- read_xlsx("Data.xlsx")
new_data
```

```
## # A tibble: 149 x 5
##    `mls listing` asking bedrooms bathrooms type
##    <chr>          <dbl>    <dbl>     <dbl> <chr>
##  1 E4621325      450000      2           2 apartment
##  2 E4637392      459000      2.5         2 apartment
##  3 E4625335      489000      2           1 apartment
##  4 E4626425      489900      2.5         2 apartment
##  5 E4627778      528000      3           2 townhouse
##  6 E4631823      539000      2           2 apartment
##  7 E4626619      549900      2.5         2 apartment
##  8 E4630908      569900      3.5         3 townhouse
##  9 E4630485      400000      1.5         1 apartment
## 10 E4638069      415000      1           1 apartment
## # ... with 139 more rows
```

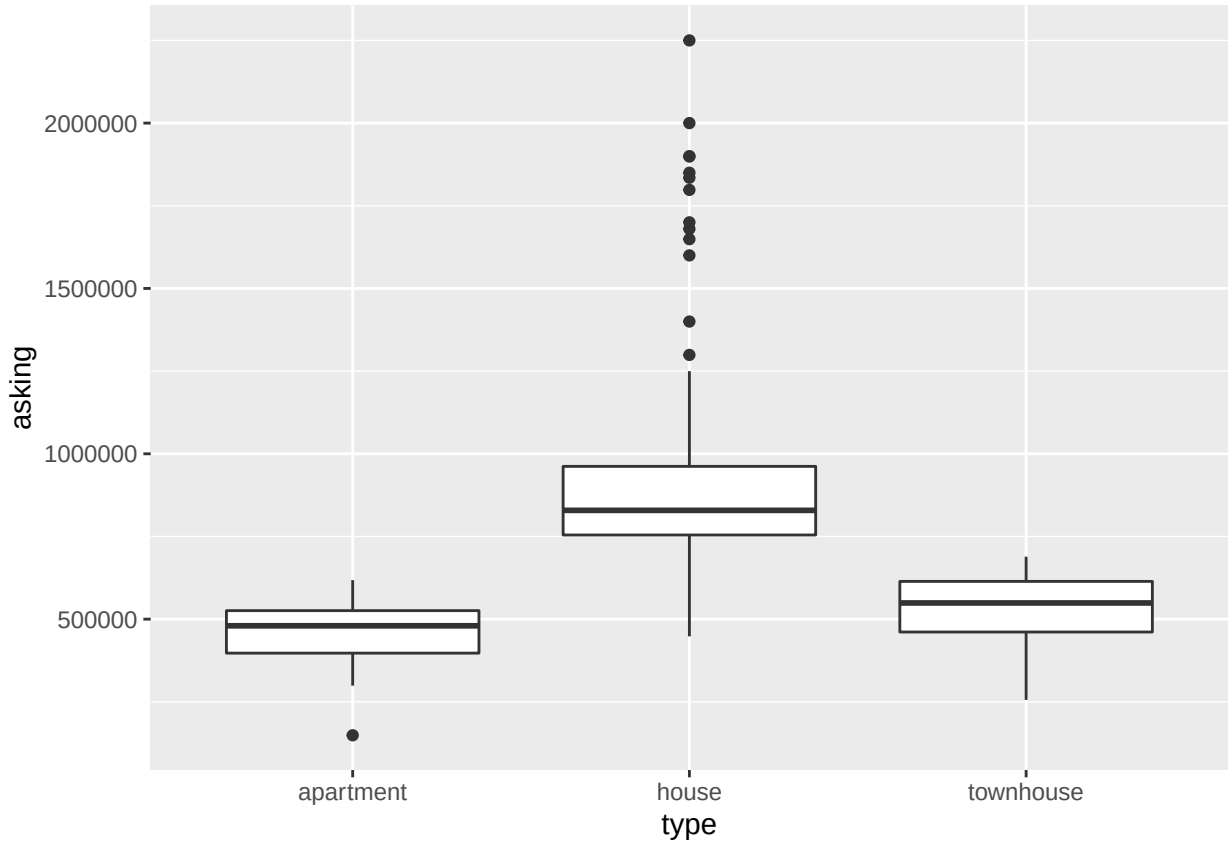
Here is a the same regressional model for all types of housing with the number of bedrooms and bathrooms being the predictor variables and the asking price being the variable to be predicted along side a boxplot of the new data.

```
new_total_reg_model = lm(asking ~ bedrooms + bathrooms, data = new_data)
summary(new_total_reg_model)
```

```
##
## Call:
## lm(formula = asking ~ bedrooms + bathrooms, data = new_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -542105 -126383  -23928  100013 1047184
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    10224      64771   0.158    0.875
## bedrooms      106934      24118   4.434 1.81e-05 ***
## bathrooms     164362      27444   5.989 1.57e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 260700 on 146 degrees of freedom
## Multiple R-squared:  0.5301, Adjusted R-squared:  0.5236
```

```
## F-statistic: 82.34 on 2 and 146 DF,  p-value: < 2.2e-16
```

```
ggplot(new_data, aes(x = type, y = asking)) + geom_boxplot()
```



Similarly to the old data, the new data has bedrooms being an important factor in determining the price of a house with a high estimate and a very low p-value of 1.81e-05. However, this time it seems that bathrooms may play a bigger role in deciding the asking price of a home as the average price increase per bathroom seems to be around $$$160,000 CAD now with a p-value well below $\alpha = 0.05$ at 1.57e-08.

Now to look at the each individual data seperately.

Checking Normality

```
newhouse_only <- new_data [new_data$type=="house",]
newapt_only <- new_data [new_data$type=="apartment",]
newtown_only<- new_data [new_data$type=="townhouse",]

newhouse_only
```

```
## # A tibble: 95 x 5
##    `mls listing`  asking bedrooms bathrooms type
##    <chr>           <dbl>    <dbl>     <dbl> <chr>
##  1 E4622082       975000      4.5         5 house
##  2 E4633606       448000      3.5         2 house
##  3 E4634102       469900      3           2 house
##  4 E4638096       469900      4           3 house
##  5 E4633288       484900      4           3 house
##  6 E4632754       499900      3           3 house
##  7 E4638480       549000      3           3 house
```

```
##  8 E4569477       899000      4.5         3 house
##  9 E4636146      1899000      4           3 house
## 10 E4631283       799900      3.5         2 house
## # ... with 85 more rows
```

```
newapt_only
```

```
## # A tibble: 44 x 5
##    `mls listing` asking bedrooms bathrooms type
##    <chr>          <dbl>    <dbl>     <dbl> <chr>
##  1 E4621325      450000      2           2 apartment
##  2 E4637392      459000      2.5         2 apartment
##  3 E4625335      489000      2           1 apartment
##  4 E4626425      489900      2.5         2 apartment
##  5 E4631823      539000      2           2 apartment
##  6 E4626619      549900      2.5         2 apartment
##  7 E4630485      400000      1.5         1 apartment
##  8 E4638069      415000      1           1 apartment
##  9 E4633971      449900      1.5         1 apartment
## 10 E4637773      498000      2           2 apartment
## # ... with 34 more rows
```

```
newtown_only
```

```
## # A tibble: 10 x 5
##    `mls listing` asking bedrooms bathrooms type
##    <chr>          <dbl>    <dbl>     <dbl> <chr>
##  1 E4627778      528000      3           2 townhouse
##  2 E4630908      569900      3.5         3 townhouse
##  3 E4534277      255900      3           2 townhouse
##  4 E4624613      299000      4           2 townhouse
##  5 E4616853      688900      3.5         4 townhouse
##  6 E4633122      439900      3           2 townhouse
##  7 E4634285      525000      3           2 townhouse
##  8 E4627733      569900      3           3 townhouse
##  9 E4579258      629000      5           3 townhouse
## 10 E4520750      685000      3.5         3 townhouse
```

```
shapiro.test(newhouse_only$asking)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  newhouse_only$asking
## W = 0.78096, p-value = 1.397e-10
```

```
shapiro.test(newapt_only$asking)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  newapt_only$asking
## W = 0.94009, p-value = 0.02381
```

```
shapiro.test(newtown_only$asking)
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  newtown_only$asking
## W = 0.9056, p-value = 0.2521
```

```
shapiro.test(new_data$asking)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  new_data$asking
## W = 0.84747, p-value = 3.876e-11
```

As seen from the Shapiro-Wilks test, while the W value has gone down in some of the individual types of homes and in general compared to the data in 2017, the p-value has dropped significantly which means that this data is more likely to be normal that that of 2017. Now, a regressional model of each individual types of homes.

```
new_house_reg = lm(asking ~ bedrooms + bathrooms, data = newhouse_only)
new_apt_reg = lm(asking ~ bedrooms + bathrooms, data = newapt_only)
new_town_reg = lm(asking ~ bedrooms + bathrooms, data = newtown_only)
summary(new_house_reg)
```

```
##
## Call:
## lm(formula = asking ~ bedrooms + bathrooms, data = newhouse_only)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -525767 -148515  -22098   76139  940575
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   172700     124695   1.385   0.1694
## bedrooms       80837      35637   2.268   0.0256 *
## bathrooms     162526      32497   5.001 2.72e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 292700 on 92 degrees of freedom
## Multiple R-squared:  0.3771, Adjusted R-squared:  0.3636
## F-statistic: 27.85 on 2 and 92 DF,  p-value: 3.495e-10
```

```
summary(new_apt_reg)
```

```
##
## Call:
## lm(formula = asking ~ bedrooms + bathrooms, data = newapt_only)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -206490  -33568    8218   48649  158519
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   261776      37538   6.974  1.8e-08 ***
## bedrooms       28182      23261   1.212   0.2326
## bathrooms      79523      32639   2.436   0.0193 *
```

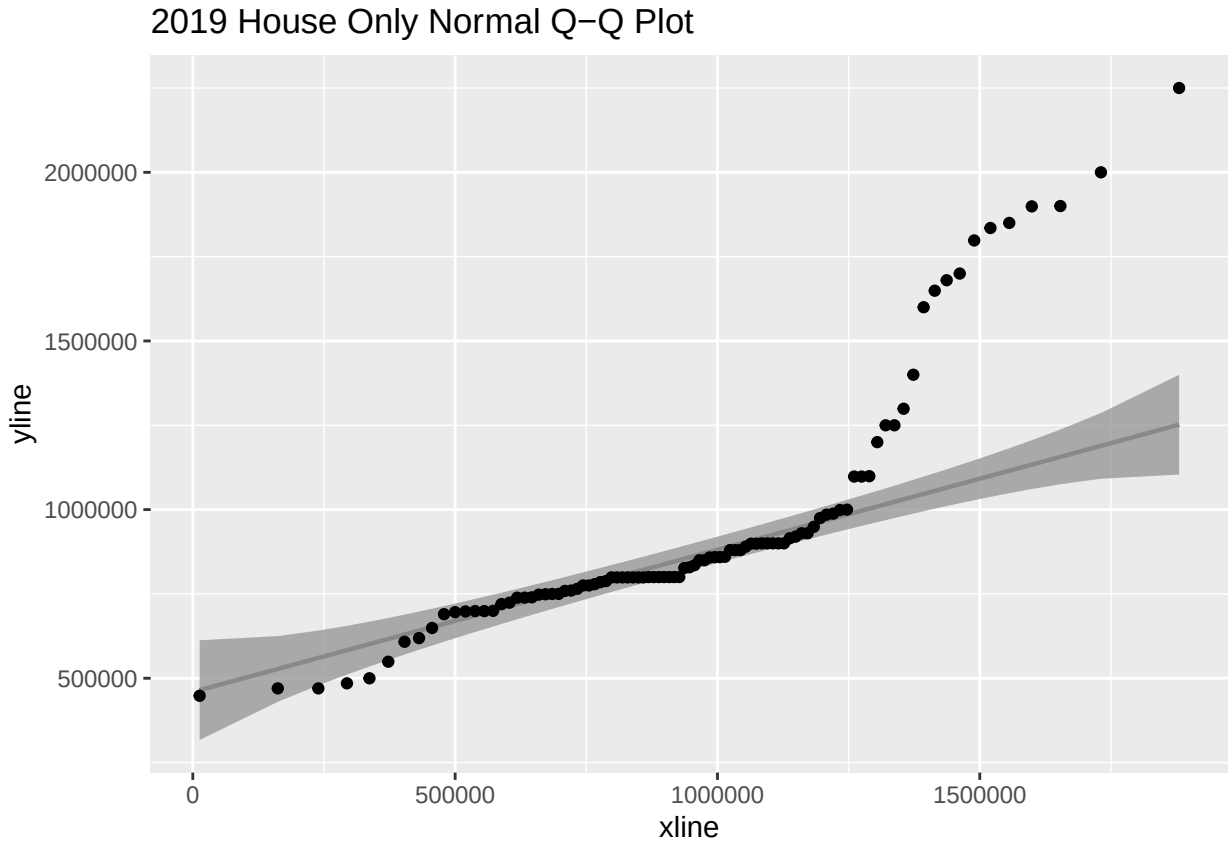
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 69760 on 41 degrees of freedom
## Multiple R-squared:  0.4157, Adjusted R-squared:  0.3872
## F-statistic: 14.59 on 2 and 41 DF,  p-value: 1.643e-05
```

```
summary(new_town_reg)
```

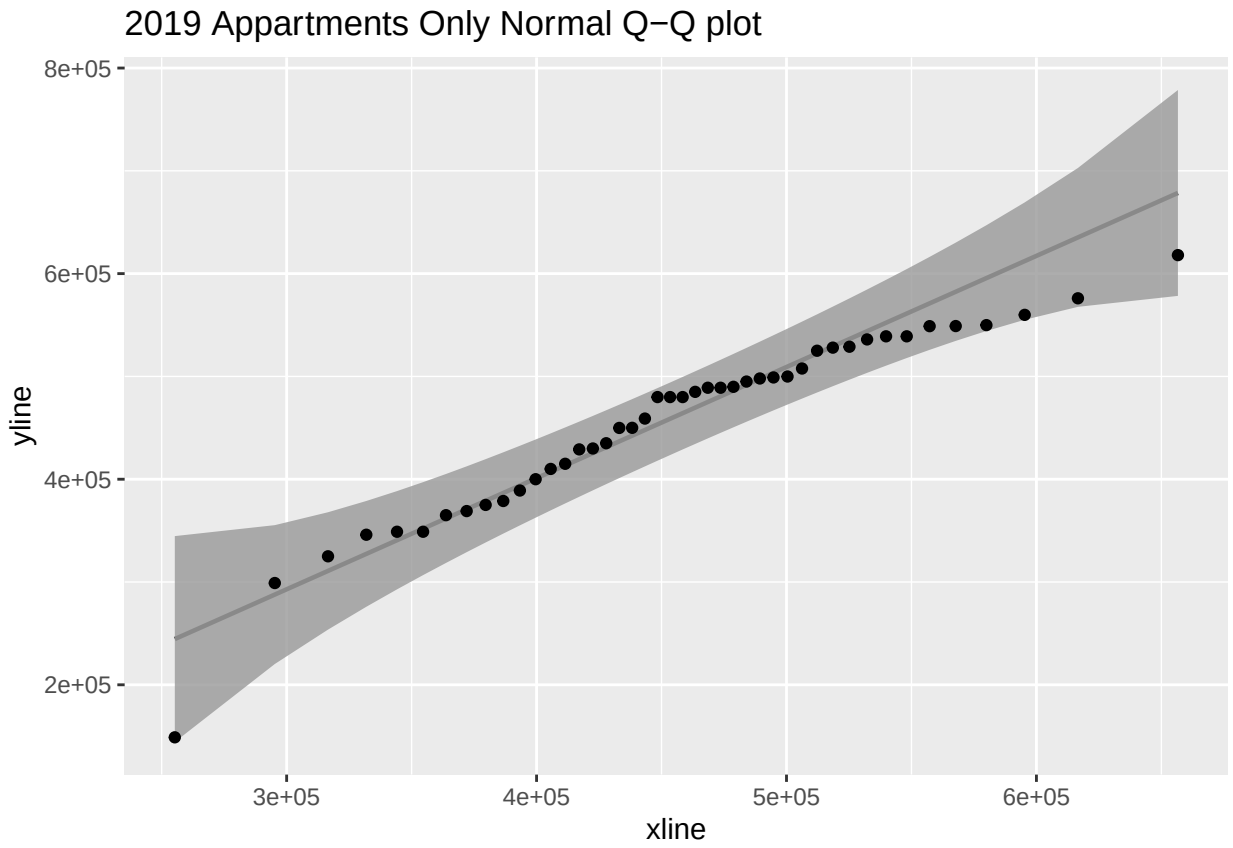
```
##
## Call:
## lm(formula = asking ~ bedrooms + bathrooms, data = newtown_only)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -166867  -49705    1195   88463  105233
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   110246     208950   0.528   0.6141
## bedrooms       -5614      58727  -0.096   0.9265
## bathrooms     164681      54035   3.048   0.0186 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 107300 on 7 degrees of freedom
## Multiple R-squared:  0.592,  Adjusted R-squared:  0.4755
## F-statistic: 5.079 on 2 and 7 DF,  p-value: 0.04338
```

As seen from the models above, there are some mixed blessings. While the R squared values have gone up significantly, the predictor variables have actually gone down in p-value and significance with the worse offender being the townhouse model wich acually says bedrooms actually *decrease* the asking price. Other models have shown that bathrooms have become the new factor that is increasing the price of homes. To look closer at this, we will need the normal Q-Q plots and the residual plots.

```
newhousescatter <- ggplot(newhouse_only, aes(sample = asking)) + stat_qq_line() + stat_qq_band() + stat
newaptscatter <- ggplot(newapt_only, aes(sample = asking)) + stat_qq_line() + stat_qq_band() + stat_qq_l
newtownscatter <- ggplot(newtown_only, aes(sample = asking)) + stat_qq_line() + stat_qq_band() + stat_q

print(newhousescatter + ggtitle("2019 House Only Normal Q-Q Plot"))
```




```
print(newaptscatter + ggtitle("2019 Appartments Only Normal Q-Q plot"))
```

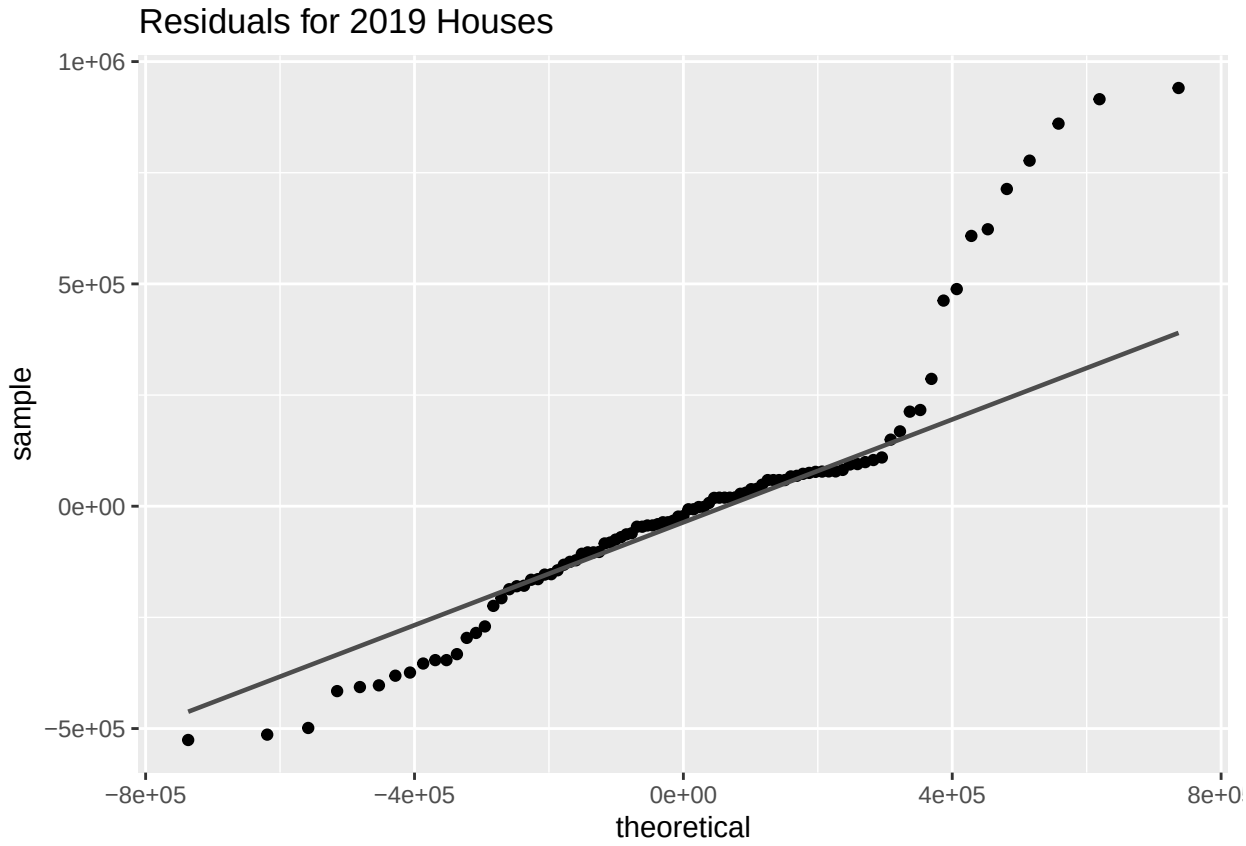



```
print(newtownscatter + ggtitle("2019 Townhouse Only Normal Q-Q plot"))
```

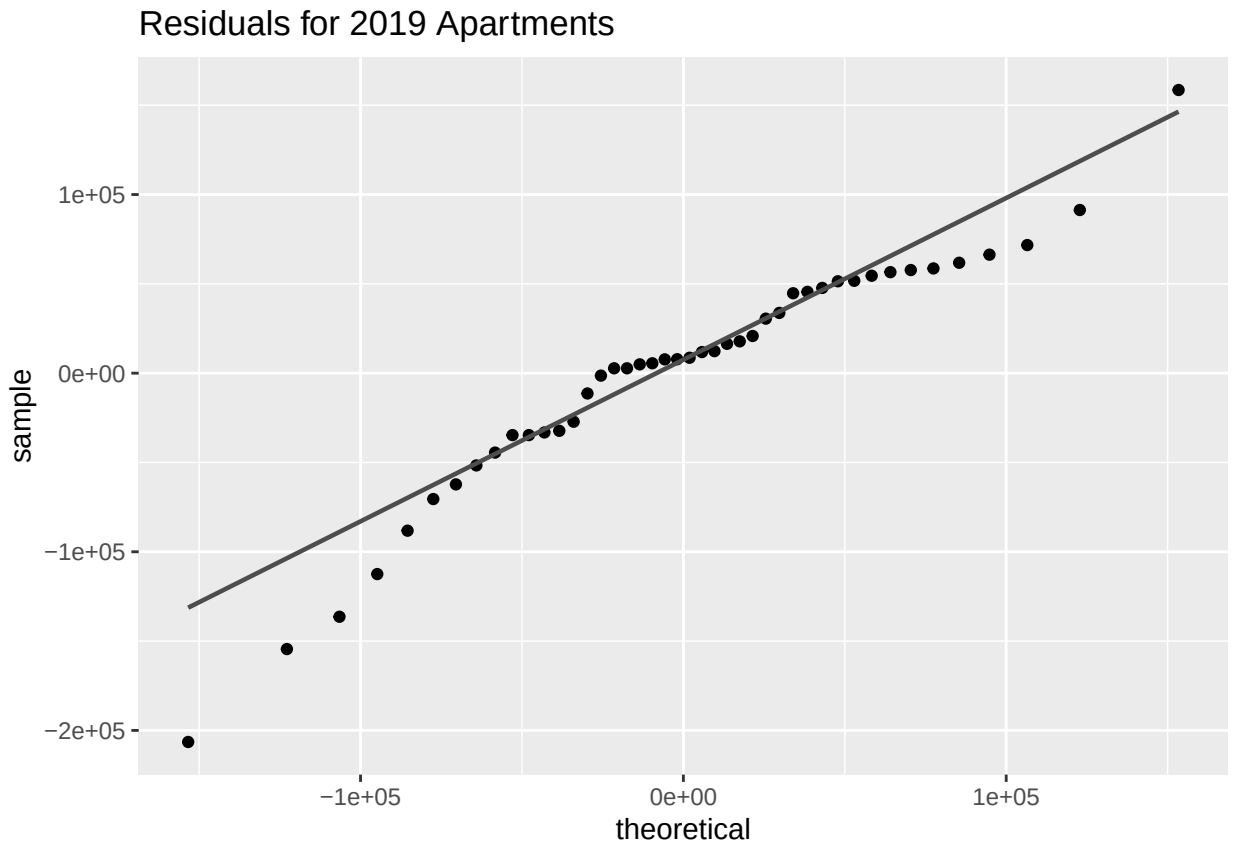
When looking at the Q-Q plot for houses only, we can see why we had such a low W value. We can also see that why we had such weird outcomes for anything regarding townhouses, there still is not enough data.

Residual plots for individual types of houses

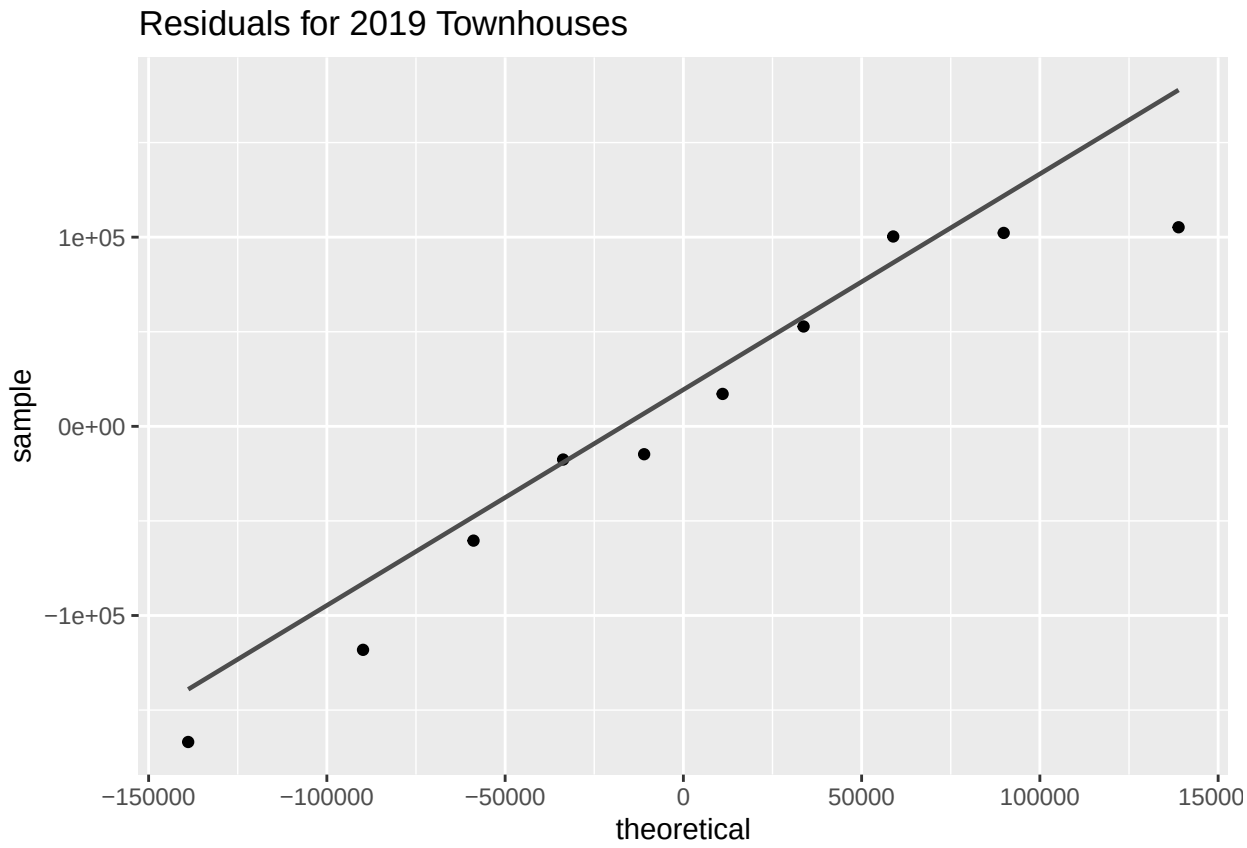
```
new_house_resfit = ggplot(new_house_reg, aes(y=.resid, x = .fitted)) + geom_point()
new_apt_resfit = ggplot(new_apt_reg, aes(y=.resid, x = .fitted)) + geom_point()
new_town_resfit = ggplot(new_town_reg, aes(y=.resid, x = .fitted)) + geom_point()
new_total_resfit = ggplot(new_total_reg_model, aes(y = .resid, x = .fitted)) + geom_point()

new_house_res = ggplot(new_house_reg, aes(sample = .resid)) + stat_qq_point() + stat_qq_line()
new_apt_res = ggplot(new_apt_reg, aes(sample = .resid)) + stat_qq_point() + stat_qq_line()
new_town_res = ggplot(new_town_reg, aes(sample = .resid)) + stat_qq_point() + stat_qq_line()
new_total_res = ggplot(new_total_reg_model, aes(sample = .resid)) + stat_qq_point() + stat_qq_line()

print(new_house_res + ggtitle("Residuals for 2019 Houses"))
```



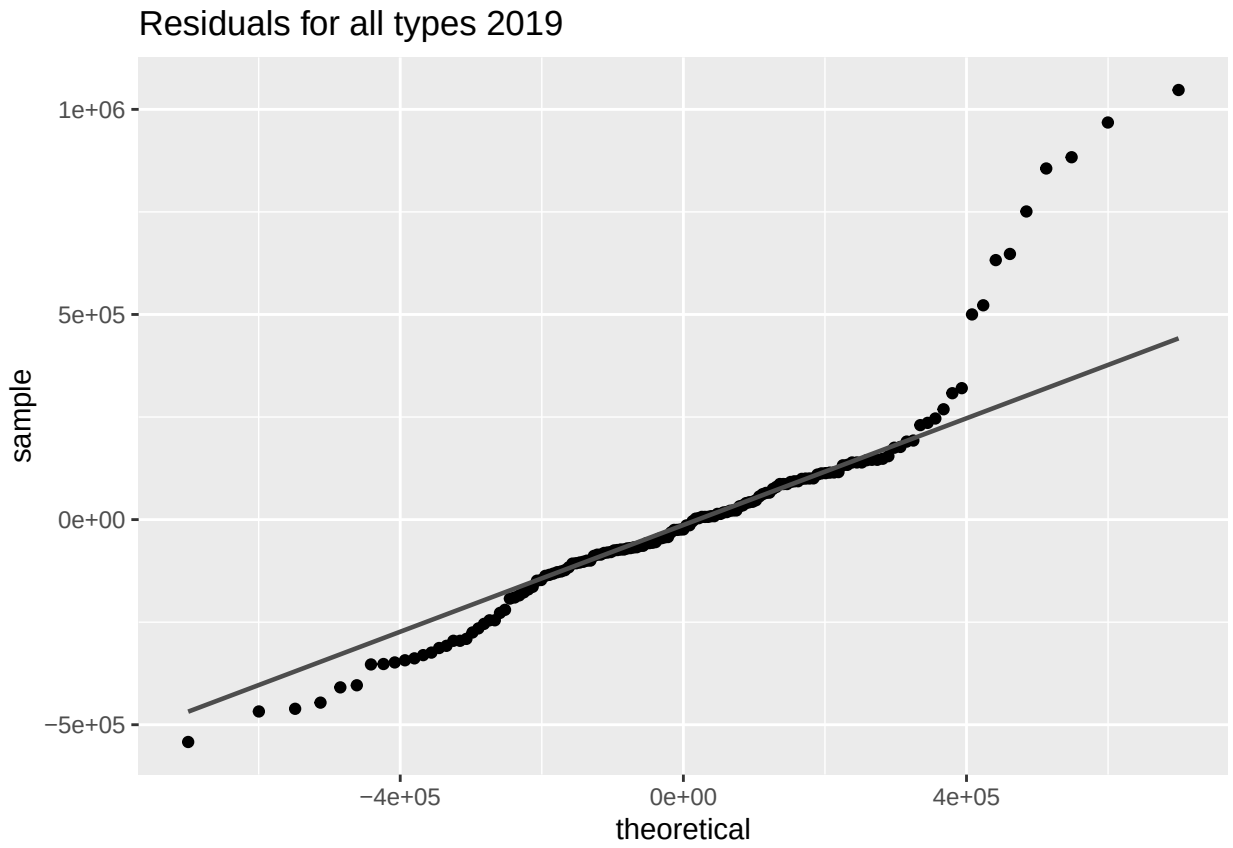

```
print(new_apt_res + ggtitle("Residuals for 2019 Apartments"))
```



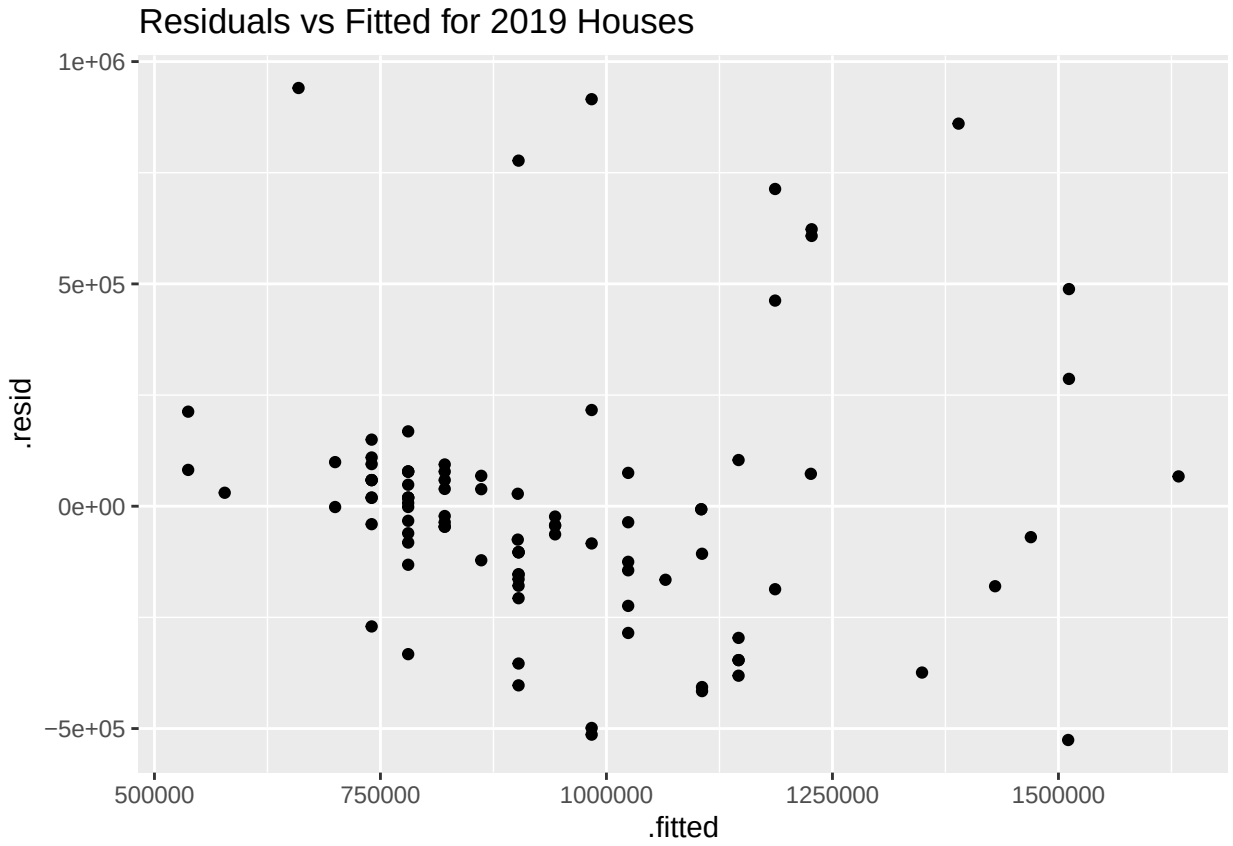

```
print(new_town_res + ggtitle("Residuals for 2019 Townhouses"))
```



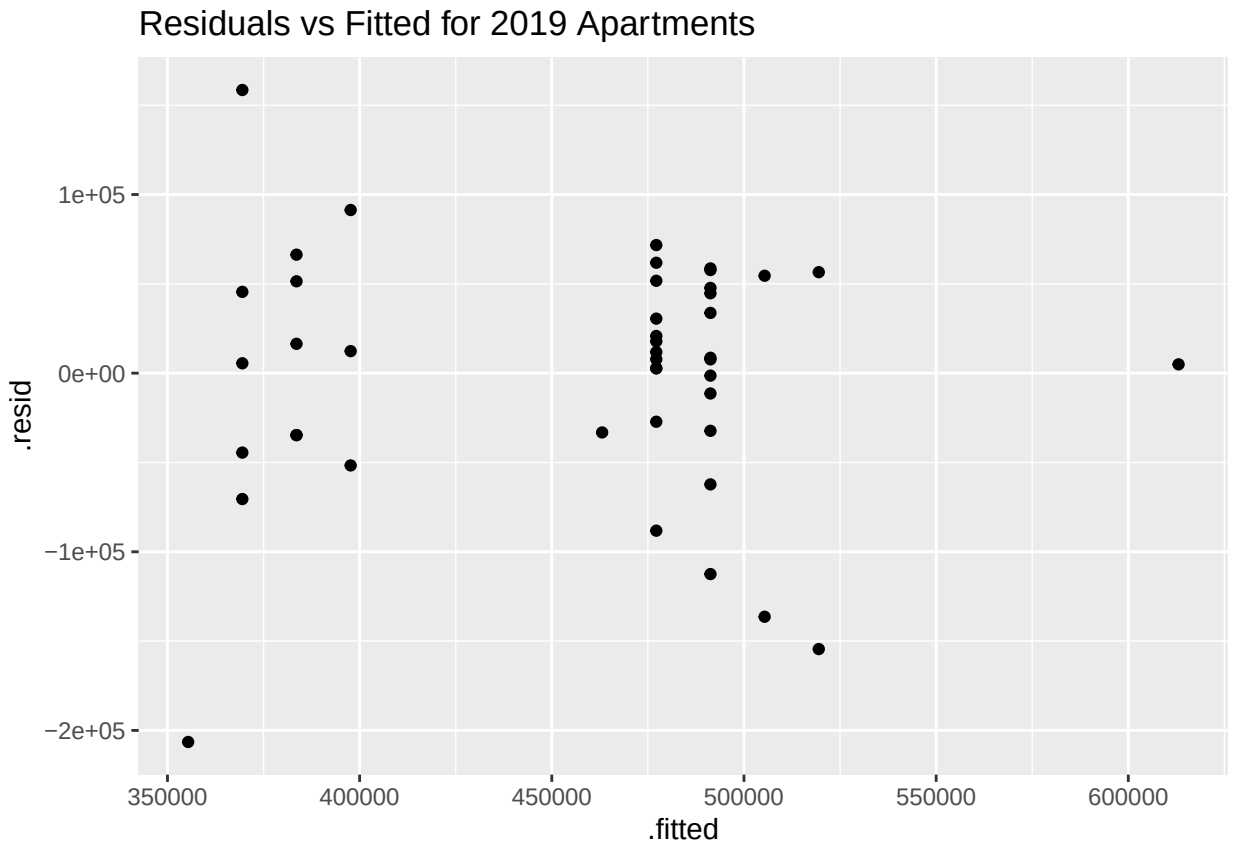

```
print(new_total_res + ggtitle("Residuals for all types 2019"))
```



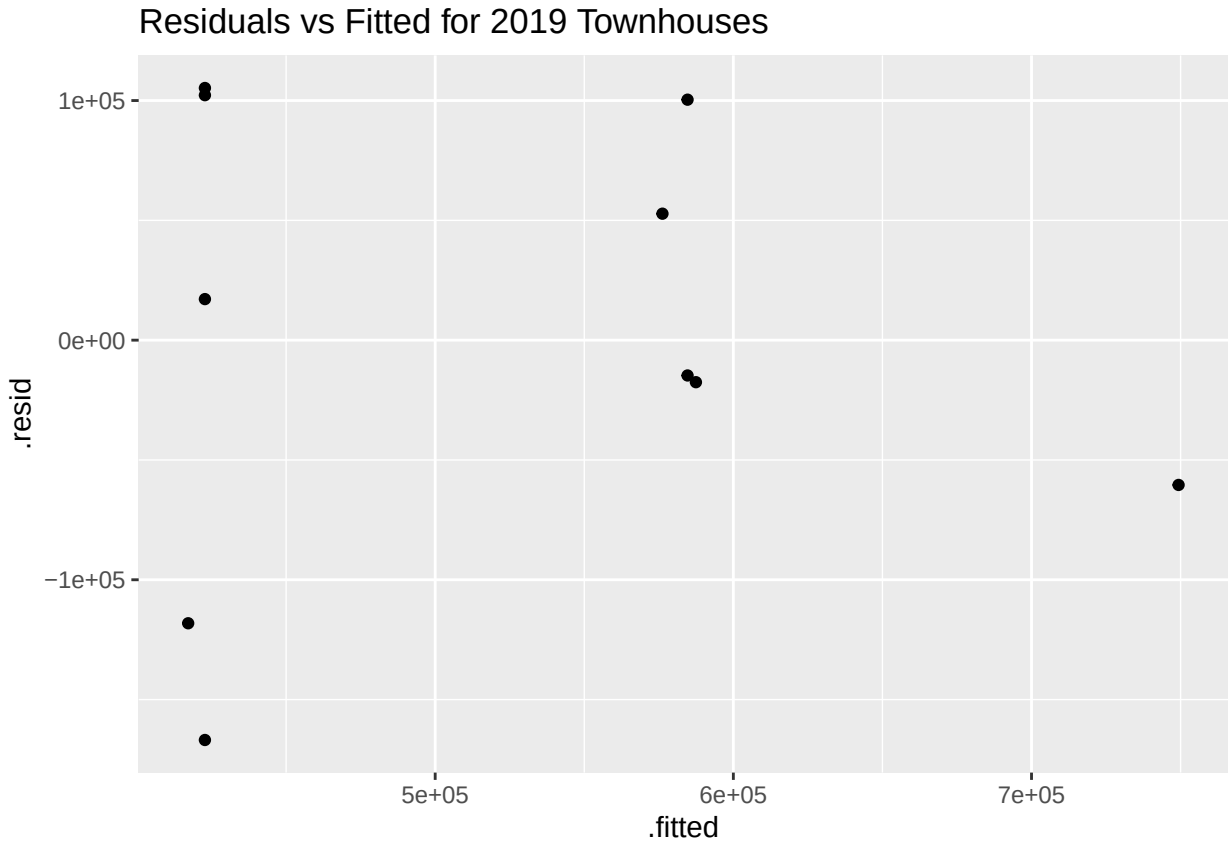

```
print(new_house_resfit + ggtitle("Residuals vs Fitted for 2019 Houses"))
```



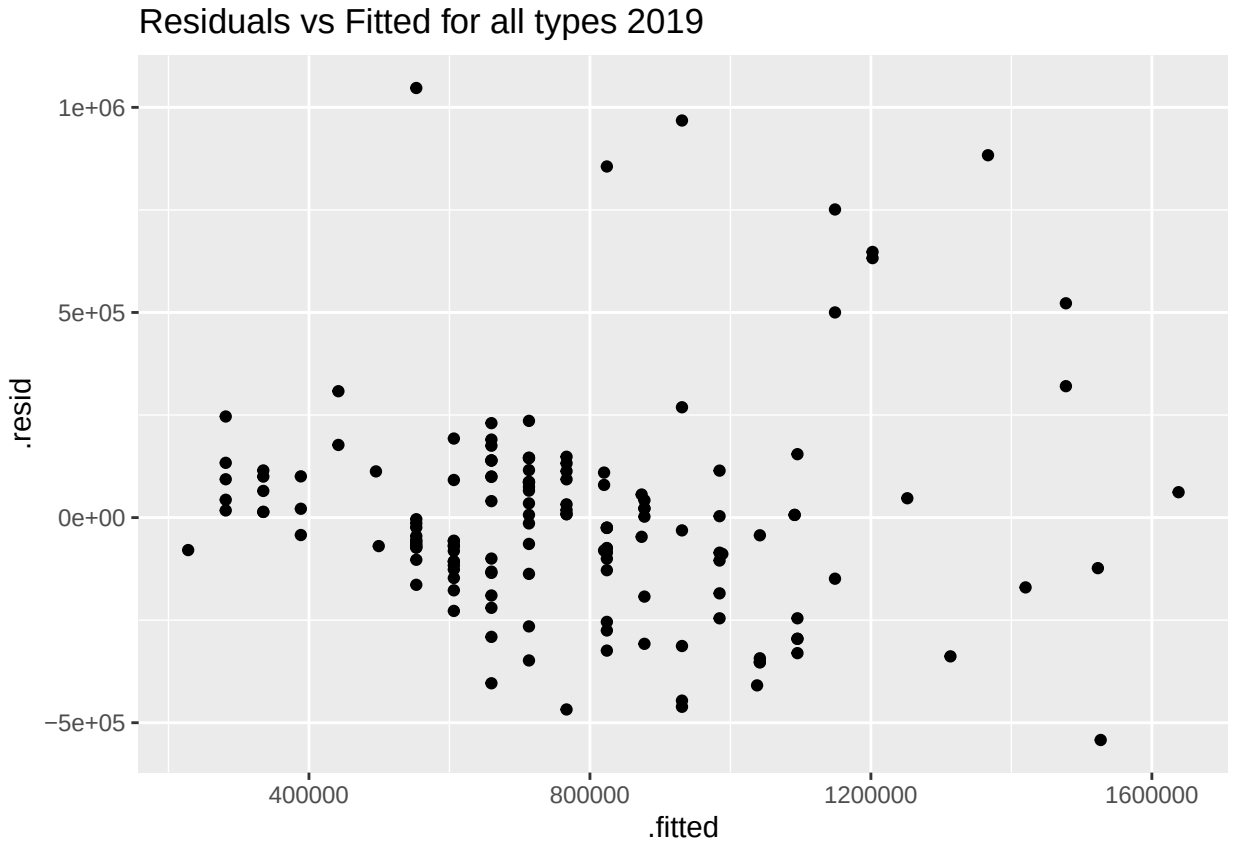

```
print(new_apt_resfit + ggtitle("Residuals vs Fitted for 2019 Apartments"))
```




```
print(new_town_resfit + ggtitle("Residuals vs Fitted for 2019 Townhouses"))
```




```
print(new_total_resfit + ggtitle("Residuals vs Fitted for all types 2019"))
```



That was a lot of data being displayed, let us start from the first graph. The first four graphs are the Q-Q normal plots for residuals for each regressional model. All of these graphs are quite egregious in their own rights, let us explain why that's so. The first graph is the residual plot of the house category. While the middle of the graph follows the estimated line quite nicely, the ends tend to trail off with the right side trailing off extremely hard; creating a very hard right skew. It trailed off of the estimated line so hard that it singlehandedly brought all of the data in general to a right skew as well as seen in the fourth chart. This is also reflected in the previous boxplots where there is an absolute abundance of outliers on the right side.

The second residual plot of apartments is the only one that has a normal shape despite it having a left-skew tail.

The third residual plot is that of townhouses which mimics the issue with 2017 data of having too little samples to make any ind of significant statistical inference.

The residual V.S. fitted value plots for the houses only data seem to have a fanning effect to the right if the three outliers in the left upper sides are ignored which suggest a transformation may be in place. In this report, we will choose to ignore the three outliers as we while it is important for any statiscal analysis to take into account outliers, the goal here is to see if there is a *general* relationship in property prices according to its number bedrooms and bathrooms.

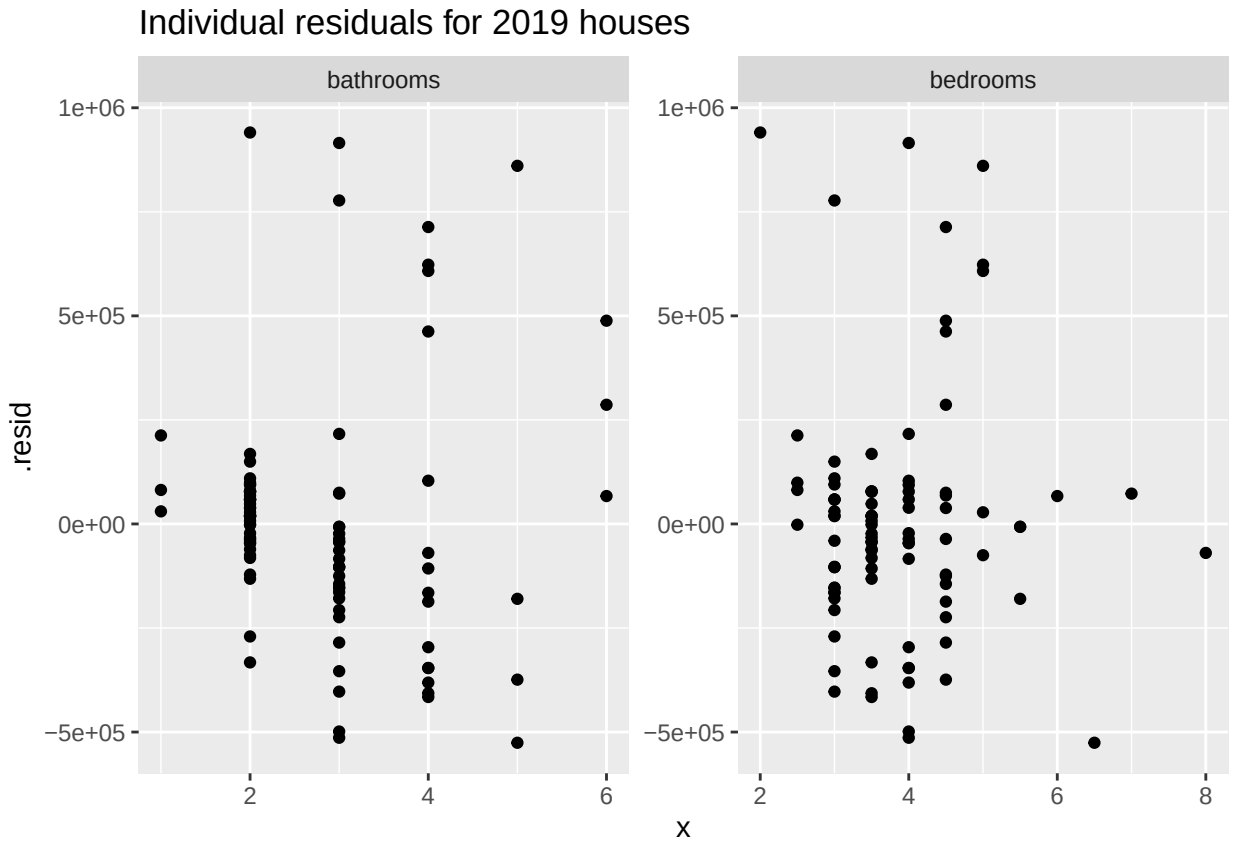
The residual V.S. fitted value plots for apartments appear to have its data clustered at two critical points.

Finally, we have the residual V.S. fitted value plot for townhouses which looks quite sad with the absolute lack of any data to make any kind of inference.

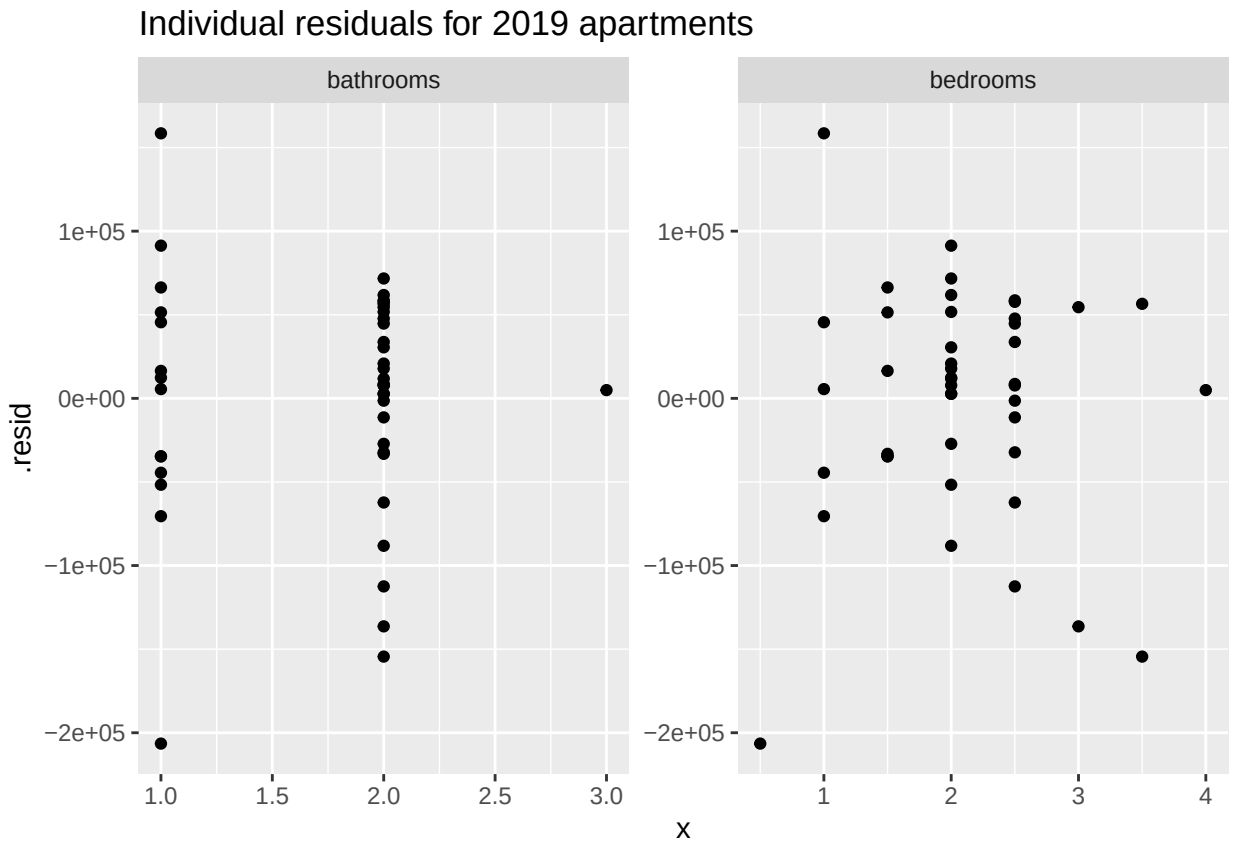
Now, we will be looking at the residual plots for each individual predictor variables.

Residuals of individual betas

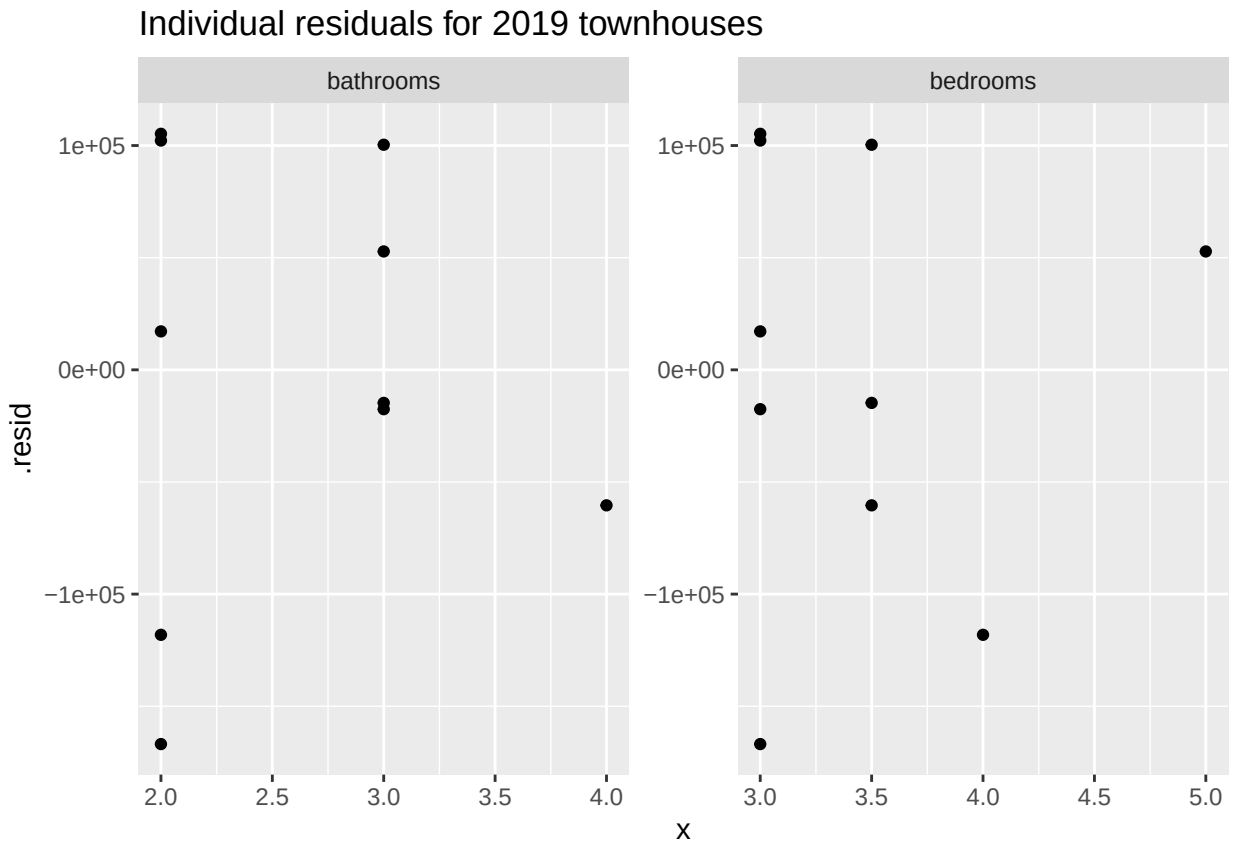
```
new_house_reg %>% augment(newhouse_only) -> nh2
nh2 %>%
  gather(xname, x, c(bedrooms:bathrooms)) %>%
  ggplot(aes(x = x, y = .resid)) + geom_point() + facet_wrap(~xname, scales = "free") ->
  new_house_betas

new_apt_reg %>% augment(newapt_only) -> na2
na2 %>%
  gather(xname, x, c(bedrooms:bathrooms)) %>%
  ggplot(aes(x = x, y = .resid)) + geom_point() + facet_wrap(~xname, scales = "free") ->
  new_apt_betas

new_town_reg %>% augment(newtown_only) -> nt2
nt2 %>%
  gather(xname, x, c(bedrooms:bathrooms)) %>%
  ggplot(aes(x = x, y = .resid)) + geom_point() + facet_wrap(~xname, scales = "free") ->
  new_town_betas

new_total_reg_model %>% augment(new_data) -> nh2
nh2 %>%
  gather(xname, x, c(bedrooms:bathrooms)) %>%
  ggplot(aes(x = x, y = .resid)) + geom_point() + facet_wrap(~xname, scales = "free") ->
  new_total_betas

print(new_house_betas + ggtitle("Individual residuals for 2019 houses"))
```



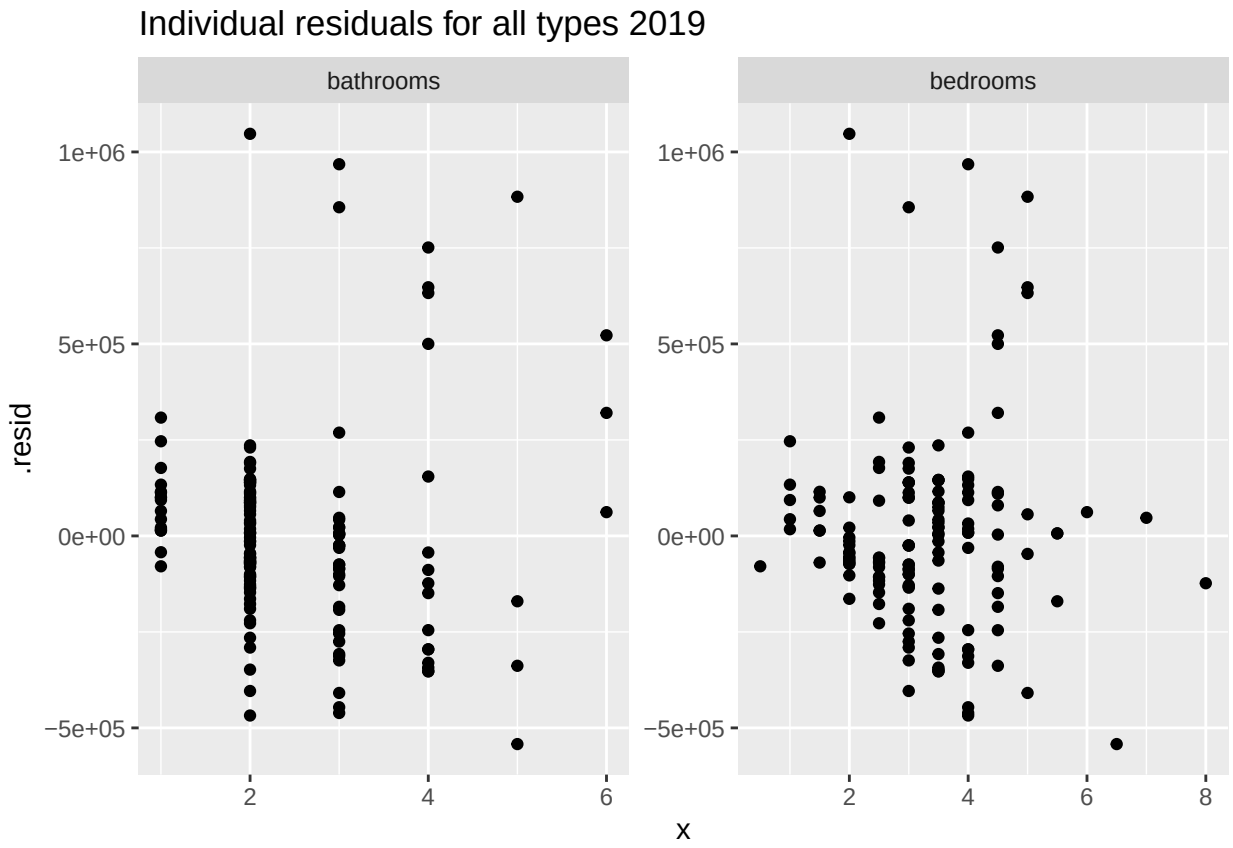
```
print(new_apt_betas + ggtitle("Individual residuals for 2019 apartments"))
```




```
print(new_town_betas + ggtitle("Individual residuals for 2019 townhouses"))
```




```
print(new_total_betas + ggtitle("Individual residuals for all types 2019"))
```

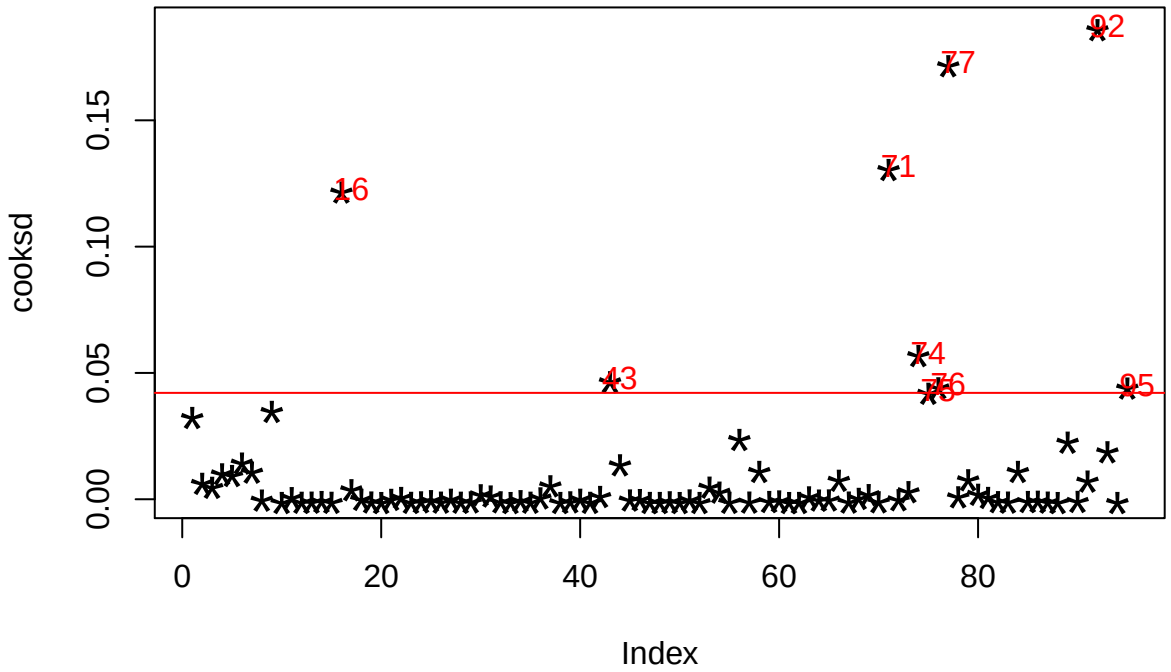


Due to the nature of the categories, it is hard to extrapolate anything meaningful from the individual predictor residuals. Instead, let us try to fix the mess of a data that is the house only category as the long right tail from the data may suggest a right skew.

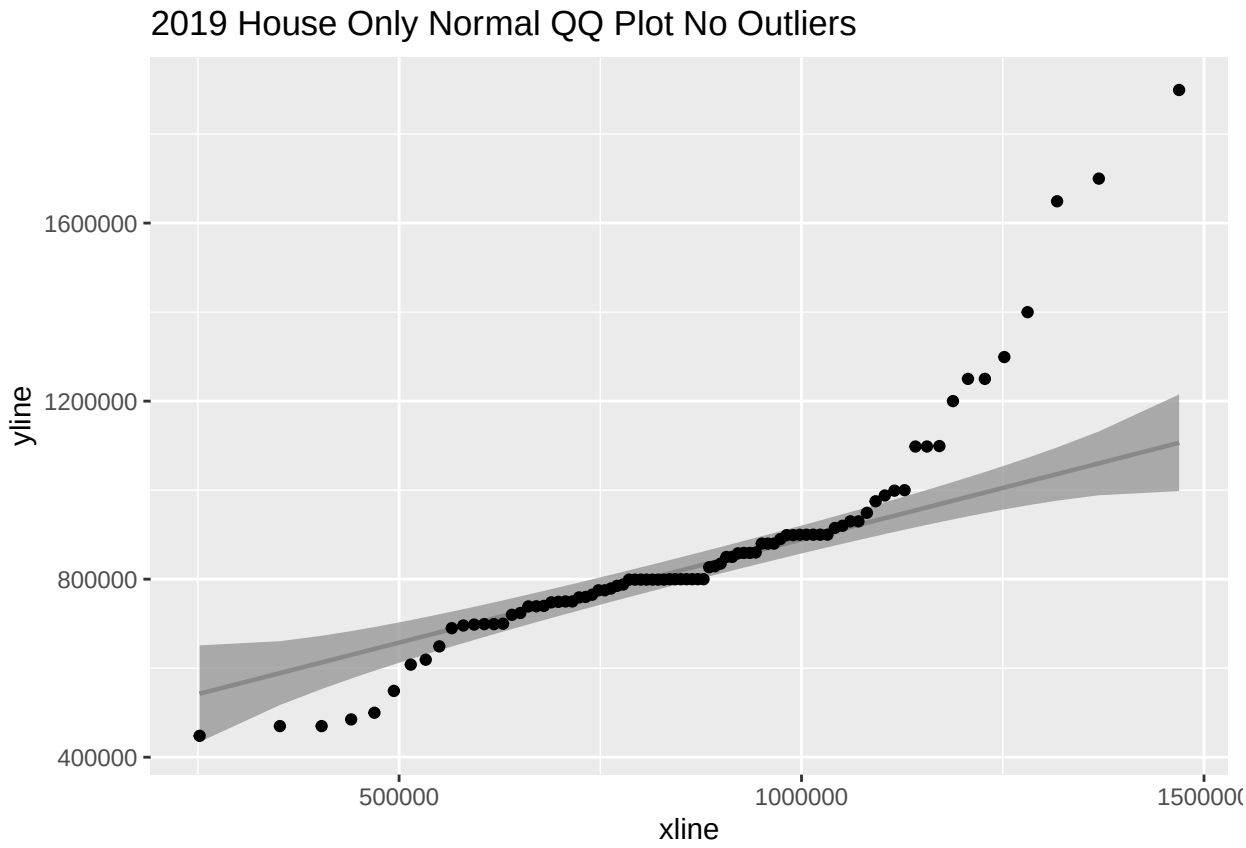
House Cleaning

First to remove outliers from the data. Since we want to know the trend of asking prices *in general*, it is best to remove the outliers by removing any point with a large and significant Cook's distance. Cook's distance is a statistical measure of how far a point is away from the rest of the data and is used to determine if a certain point is an outlier.

```
cooksd <- cooks.distance(new_house_reg)
sample_size <- nrow(newhouse_only)
plot(cooksd, pch="*", cex=2, main="Influential House Obs by Cooks D")
abline(h = 4/sample_size, col="red")
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4/sample_size, names(cooksd),""), col="red")
```




```
influential <- as.numeric(names(cooksd)[(cooksd > (4/sample_size))])
general_house <- newhouse_only[-influential, ]
newhouses_no_outliers <- ggplot(general_house, aes(sample = asking)) + stat_qq_line() + stat_qq_band() +
print(newhouses_no_outliers + ggtitle("2019 House Only Normal QQ Plot No Outliers"))
```



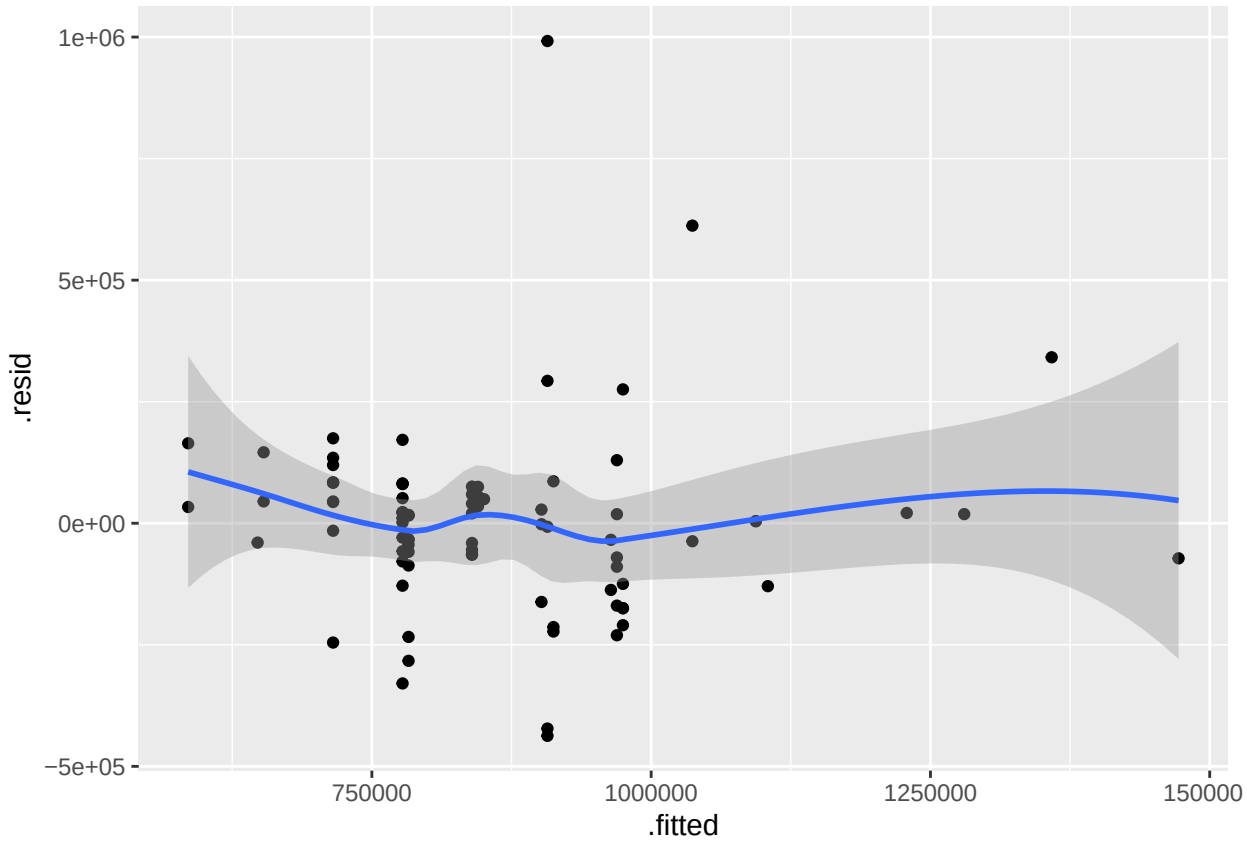
Here is the data without outliers.

```
new_house_reg_no_outliers = lm(asking~bedrooms + bathrooms, data = general_house)
summary(new_house_reg_no_outliers)
```

```
##
## Call:
## lm(formula = asking ~ bedrooms + bathrooms, data = general_house)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -437277  -71748   13358   58278  991823
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   207142      90188   2.297  0.02415 *
## bedrooms      124347      24604   5.054 2.54e-06 ***
## bathrooms      67549      24801   2.724  0.00787 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 189900 on 83 degrees of freedom
## Multiple R-squared:  0.4015, Adjusted R-squared:  0.3871
## F-statistic: 27.84 on 2 and 83 DF,  p-value: 5.589e-10
```

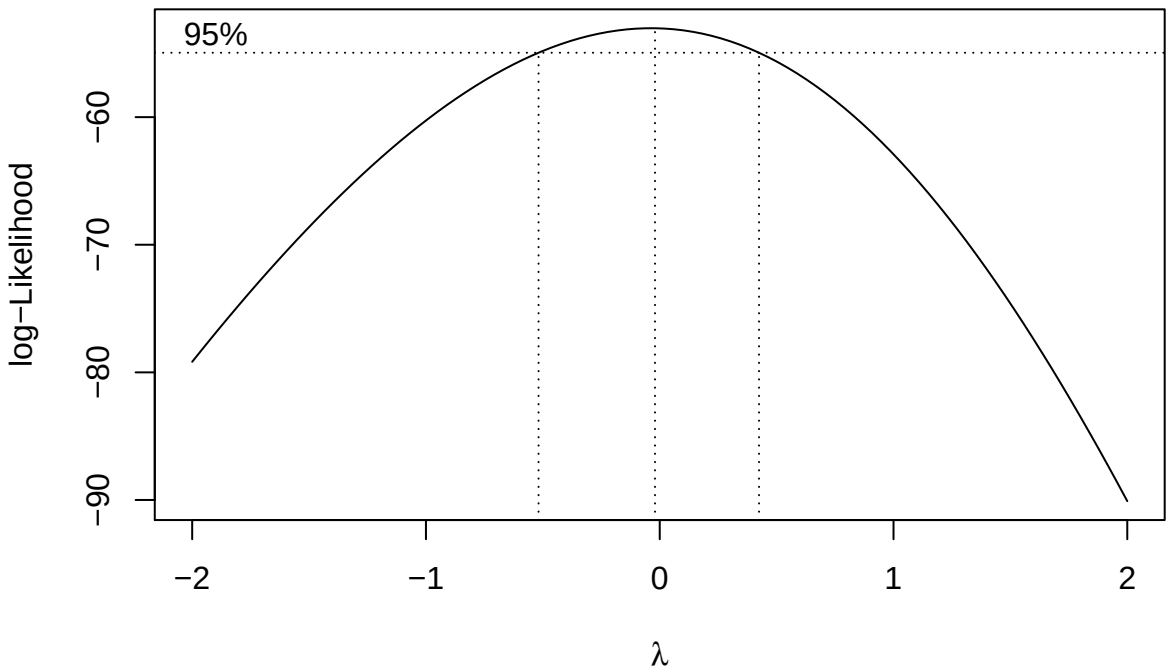
```
ggplot(new_house_reg_no_outliers, aes(y = .resid, x = .fitted)) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Now the data is primed for a Box-Cox graph to determine the kind of transformation that should be applied to the data set. ## Box Cox

```
boxcox(asking ~ bedrooms + bathrooms, data = general_house)
```

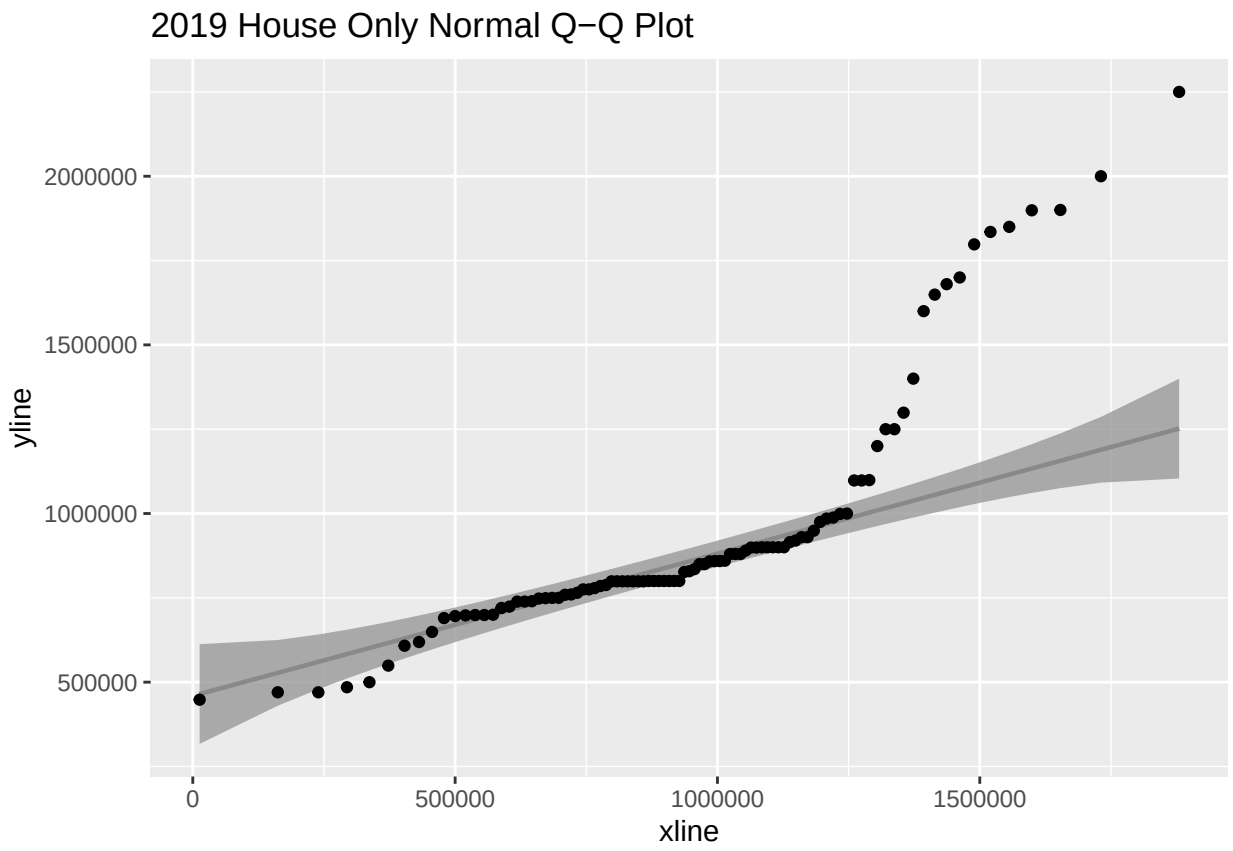


With the Box-Cox transformation, we can see that the lambda value λ is around zero which calls for a log transformation.

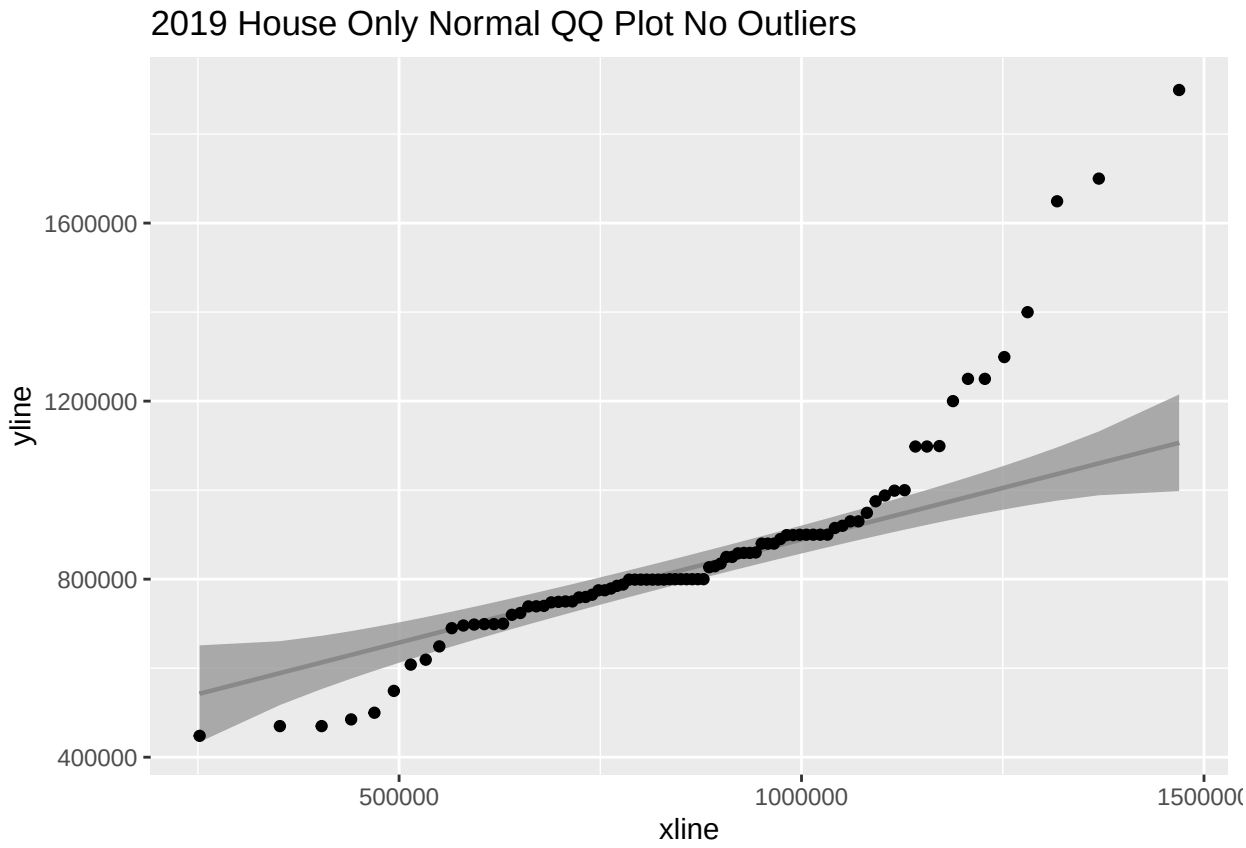
```
general_house_transform = general_house %>%
mutate(asking = log(asking))
```

Comparison

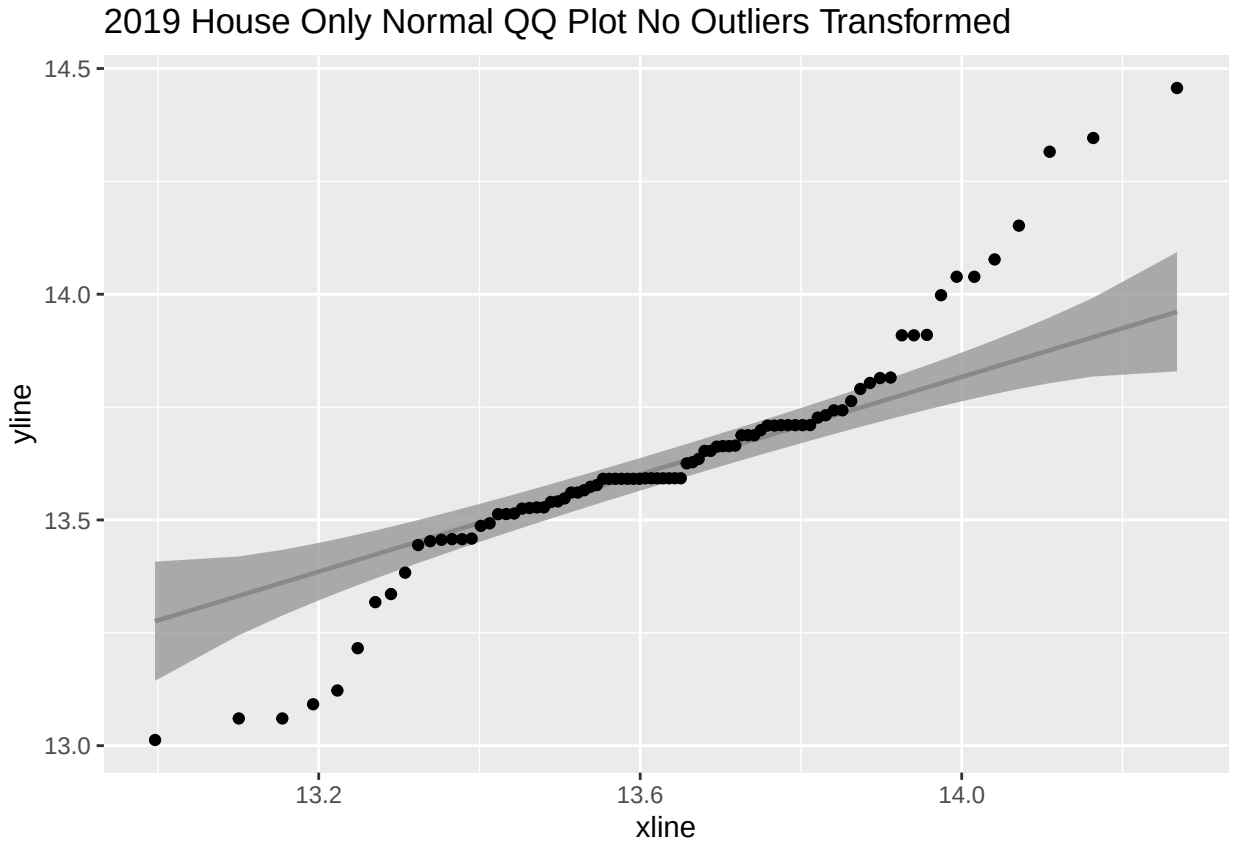
```
newhouses_no_outliers_transform <- ggplot(general_house_transform, aes(sample = asking)) + stat_qq_line

print(newhousescatter + ggtitle("2019 House Only Normal Q-Q Plot"))
```




```
print(newhouses_no_outliers + ggtitle("2019 House Only Normal QQ Plot No Outliers"))
```




```
print(newhouses_no_outliers_transform + ggtitle("2019 House Only Normal QQ Plot No Outliers Transformed
```



Here are the results of the transformation compared to the data set before the transformation and the data set before both transformation and removal of outliers in the form of normal Q-Q plots. The transformed data looks to be a bit more balanced now in terms of the tails having similar lengths now. However, this is all speculation without the model, so let us take a look at that.

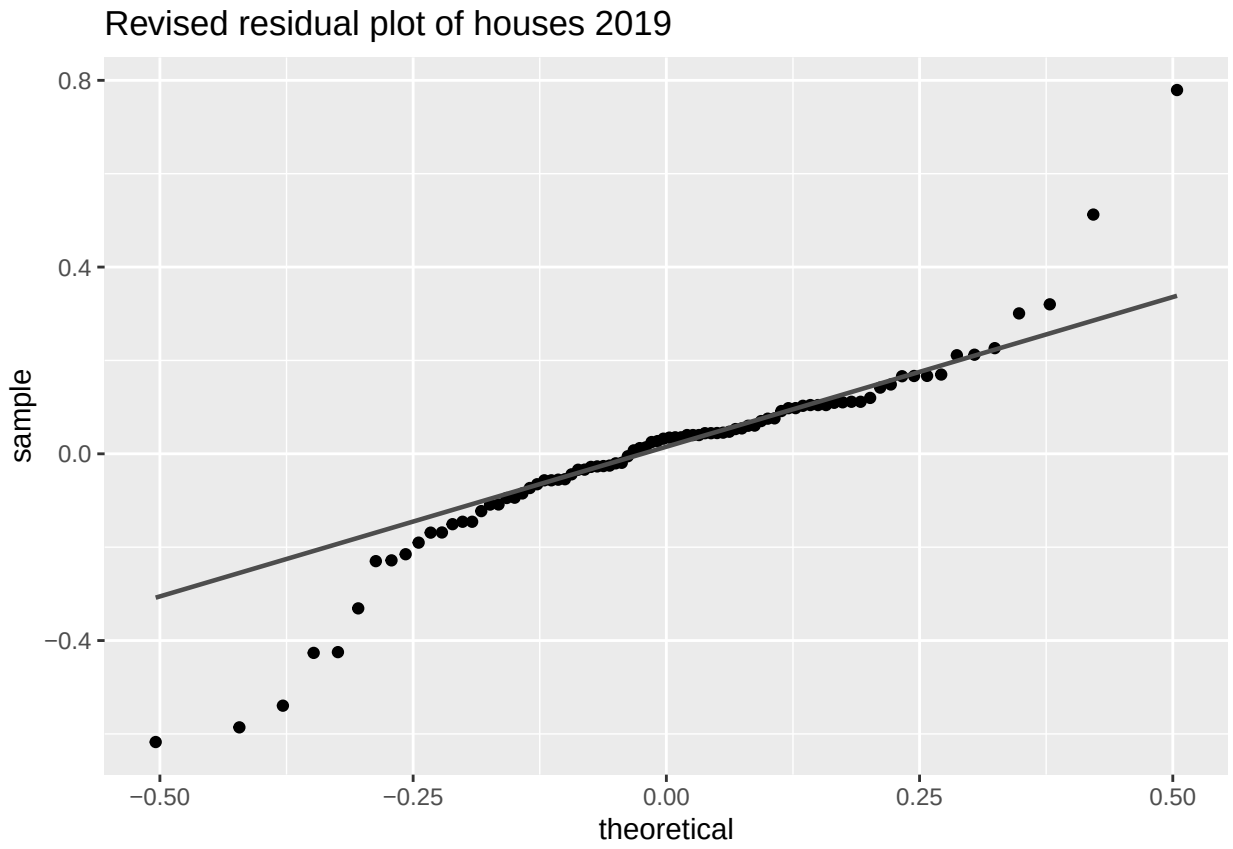
Transformed Results

```
new_house_reg_no_outliers_transform = lm(asking~bedrooms + bathrooms, data = general_house_transform)
summary(new_house_reg_no_outliers_transform)
```

```
##
## Call:
## lm(formula = asking ~ bedrooms + bathrooms, data = general_house_transform)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61736 -0.07125  0.03351  0.10161  0.77920
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.97415    0.09660 134.311  < 2e-16 ***
## bedrooms     0.13069    0.02635   4.959  3.7e-06 ***
## bathrooms    0.06024    0.02656   2.268   0.0259 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2034 on 83 degrees of freedom
## Multiple R-squared:  0.3711, Adjusted R-squared:  0.3559
```

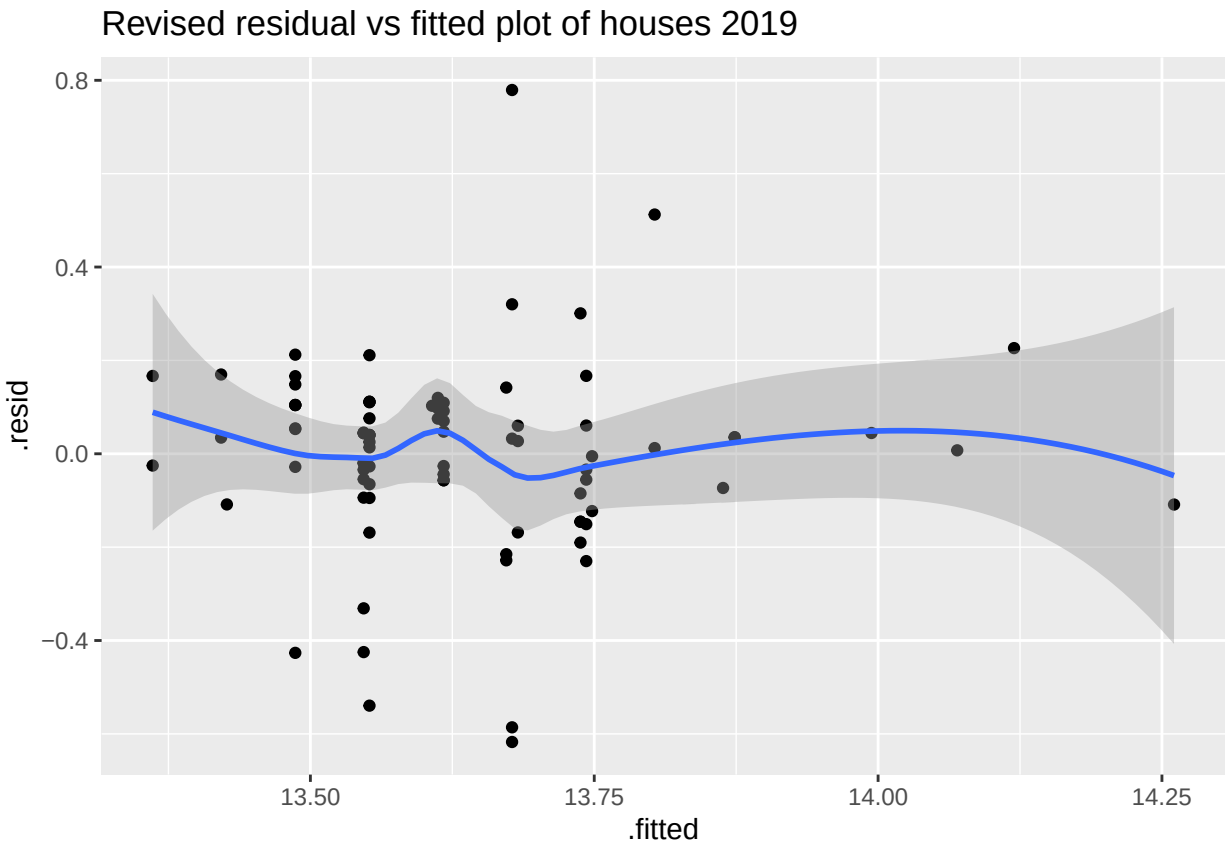
```
## F-statistic: 24.49 on 2 and 83 DF,  p-value: 4.385e-09
```

```
revised_new_house_resfit = ggplot(new_house_reg_no_outliers_transform, aes(y = .resid, x = .fitted)) + g
revised_new_house_res = ggplot(new_house_reg_no_outliers_transform, aes(sample = .resid)) + stat_qq_poi

print(revised_new_house_res + ggtitle("Revised residual plot of houses 2019"))
```




```
print(revised_new_house_resfit + ggtitle("Revised residual vs fitted plot of houses 2019"))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

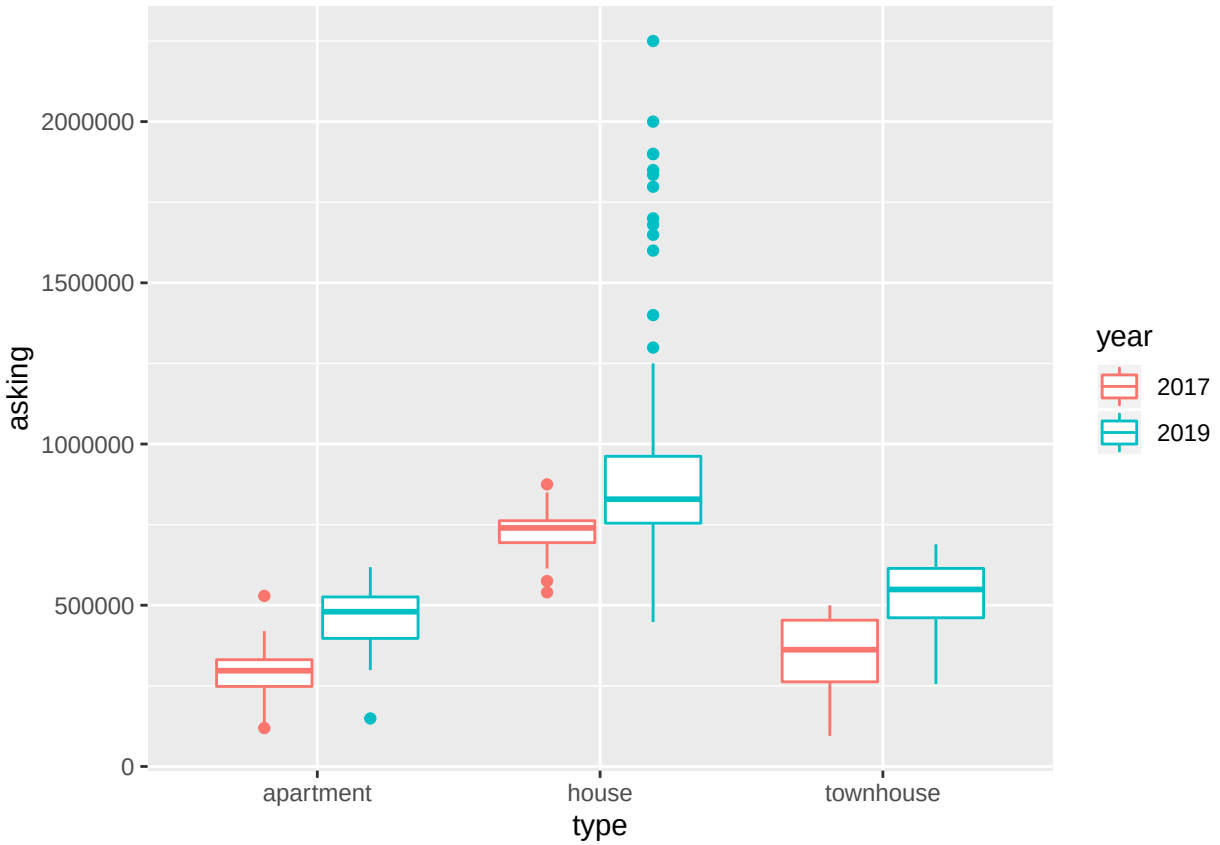


While the significance of the transformation doesn't seem to be very promising as we would like for it to be since the R squared values did not change much, it did improve the data set in a few ways. The first change is that the p-value significance shifted from bathrooms back to bedrooms as being the major asking price influencer. The new estimated value is now around 0.13. Since the data had a log transformation applied to it, this 0.13 implies that for each additional bedroom, instead of increasing price at a flat rate, it adds an about and additional 13% of the asking price. The remnant of the low R squared value may indicate that there may be a transformation in the predictor variables.

Now let us compare the new 2019 data to the old 2017 data.

Comparison OF 2017 and 2019 Data

```
houses$year <- "2017"
new_data$year <- "2019"
combined <- rbind(houses, new_data)
ggplot(combined, aes(x = type, y = asking, colour = year)) + geom_boxplot()
```



From the side-by-side boxplot above comparing `asking` prices of `apartment`, `house` and `townhouse` for `year` 2017 and 2019, it is obvious that in general 2019 has a higher price than 2017. For both years, `house` has the highest price among all three types of dwelling, while `apartment` and `townhouse`are around the same price level. Furthermore, in 2019, `asking` price of `house` shows a significant right skewness with lots of upper outliers, indicating a very hot house market with a dramatic run-up in prices of relatively expensive houses.

Conclusion of 2019 Data

From observations, we see that the main factors that changed from 2017 to 2019 is that bathrooms are now the main predictors for asking prices. This reinforces the fact that there are definitely lurking variables hiding amongst the data that was not gathered.

With same linear models done for predicting `asking` prices of `apartment`, `house` and `townhouse` for both `year` 2017 and 2019, we expected in 2019, the number of `bedrooms` and `bathrooms` should stay insignificant as in 2017. To the contrary, the results surprised us by showing significance level for numbers of `bathrooms` in 2019, while none of `bathrooms` and `bedrooms` showed significant in 2017. Due to the small sample size of 2017 data, adjusted R square values should be the ones to look at instead of original R square values. In 2017, the highest adjusted R square value is only 0.17, while in 2019 it goes up to 0.48, nearly three times higher. It is possible that as time goes, number of `bathrooms` turns out to be a more important factor in property pricing. This was not the case with only the house data as after a relevant transformation, `bedrooms` took back the title of main estimator as a percentage estimator rather than a flat value estimator. However, even year 2019 does not give good R square values as the highest is still below 50%. This indicates that certain lurking variables exist in predicting property prices, other than numbers of `bedrooms` and `bathrooms` taken into consider in our analysis.

Apart from number of bedrooms and bathrooms, the lurking variables that affect housing price could be as following:

Economic factors (e.g. economic growth rate) The rapid economic growth, the corresponding increase in the income of residents, the increase in purchasing power, the price of housing would rise. Otherwise, the price of housing would fall.

Administrative factors (e.g. property tax) Property tax, such as real estate tax, could affect the increase and decrease of residential transaction volume, lower tax rates encourage transactions, and residential prices rise.

Social factors (e.g. community environment) A good community and neighbourhood may promote higher housing prices.

Natural factors (e.g. location) Areas with good location, convenient transportation and developed commerce may have higher housing prices; The further away from this area, the lower the housing price.

Miscellaneous factors (e.g. age, appearance, design, equipment condition, and construction quality) These factors may be considered by buyers and investors, and eventually affect the housing prices.

Among all the residual-residual Q-Q plots for year 2017 and 2019, normality are poor with both upper and lower tails existing. Therefore the assumption of residuals that follow normal distribution is not met for carrying out a linear regression model. What's more, in most of the results, variables are not showing significance with all adjusted R square values lower than 50%, indicating existence of lurking variables. Last but not least, the sample size is not large enough to derive a representative conclusion. Overall, our model strength is weak, and certain improvements like increasing sample size and including more potential factors should be performed in future analysis of house price predicting.

If this experiment was to be done again, the main areas of improvement would be to have more sample sizes and to consult real estate agents and a multitude of city planning professionals to ask for their opinions on what other variables should be taken into consider when looking at homes. We hope that this report proves to be useful to every reader.