Linear Regression Final Project

Christopher Pang[1], Yueling Wu[1], Tianxiang Zhou[1]
University of San Francisco[1]

Report table of contents

Description of the dataset

- Resources: https://www.kaggle.com/anthonypino/melbourne-housing-market
- Dimension: 8887 records and 21 variables after dropping NAs
- Variable Descriptions:
    - Suburb: Suburb
    - Address: Address
    - Rooms: Number of rooms
    - Price: Price in Australian dollars
    - Method: How it is sold
        - S - property sold;
        - SP - property sold prior;
        - PI - property passed in;
        - PN - sold prior not disclosed;
        - SN - sold not disclosed;
        - NB - no bid;
        - VB - vendor bid;
        - W - withdrawn prior to auction;
        - SA - sold after auction;
        - SS - sold after auction price not disclosed.
        - N/A - price or highest bid not available.
    - Type: type of the house
        - br - bedroom(s);
        - h - house,cottage,villa, semi,terrace;
        - u - unit, duplex;
        - t - townhouse;
        - dev site - development site;
        - o res - other residential.
    - Date: Date sold
    - Distance: Distance from CBD in Kilometres. CBD stands for central business district or Melbourne city centre
    - Regionname: General Region
        - Northern Metropolitan
        - Western Metropolitan
        - Southern Metropolitan
        - Eastern Metropolitan
        - South-Eastern Metropolitan
        - Northern Victoria
        - Eastern Victoria
        - Western Victoria
    - Propertycount: Number of properties that exist in the suburb.
    - Bedroom2 : Scraped # of Bedrooms (from different source)
    - Bathroom: Number of Bathrooms
    - Car: Number of carspots

- Landsize: Land Size in Metres
- BuildingArea: Building Size in Metres
- YearBuilt: Year the house was built
- CouncilArea: Governing council for the area
- Lattitude: Self explanatory
- Longtitude: Self explanatory
- SellerG: The seller
- Postcode: Self explanatory

Initial starting variables:
Response variable: Price

Predictor requirement:
At least 3 numeric variables: Distance, Bedroom2, Bathroom, Car,Landsize, BuildingArea, YearBuilt, Lattitude, Longtitude, Propertycount
At least 3 categorical variables: Method, Type, Regionname are the initial categorical variables we started with.

Other categorical variables were not used since there were too many levels and the professor said we should have around 5 levels for a categorical level. We didn't use Address,Postcode(nominal variable),Suburb,SellerG,Date,CouncilArea.

Our initial DataFrame dimension is 8887x21 after dropping NaN.

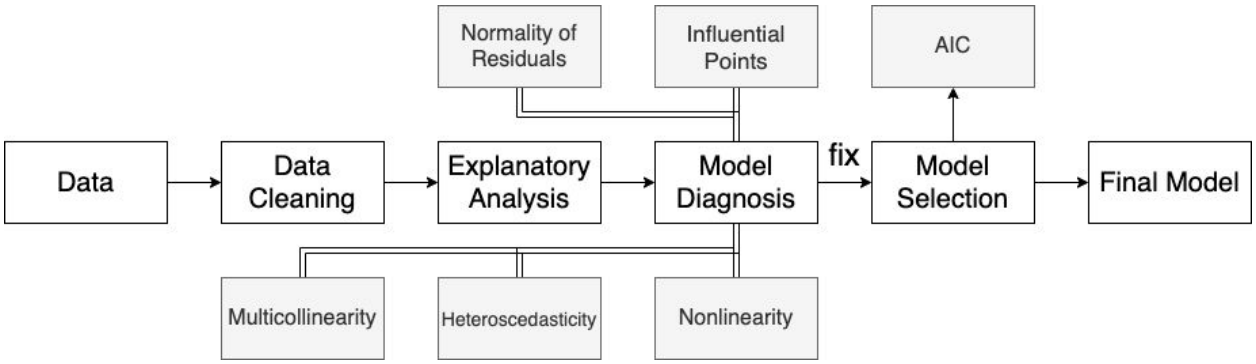
Problem Statement

Melbourne is the capital city of Australia and it's very populous. Hence, due to the demand for housing, it's essential to study the relationship between the housing price and other relevant variables based on the historical data. We propose **multiple linear regression** as our main method to discover these relationships. In this study, we will be exploring the dataset and constructing one or more linear regression models based on our analysis; then we will discuss the methods and interpretations. Lastly, we will draw inferences from our final model.

Summary of methods

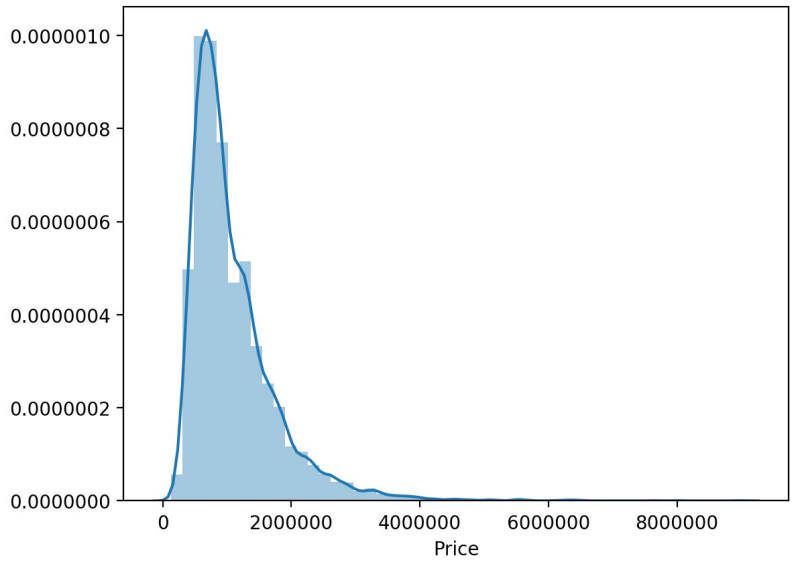
We followed the professor's flowchart on canvas for the linear regression workflow. Details: We started from scratch by cleaning the data then we did some exploratory data analysis to better understand the data. We then fit a linear regression initial model 1 and tried to diagnose the potential problems like multicollinearity,existence of influential points, non-normality of residuals, heteroscedasticity, nonlinearity. Based on the model diagnostics, we

removed influential points and transformed the data. We then fit an initial model 2 and did model diagnostics again to see the improvements. After attempting to fix the problems we perform model selection based on AIC. Lastly, the final model was decided based on the AIC criterion. An illustration of the methods are described below.

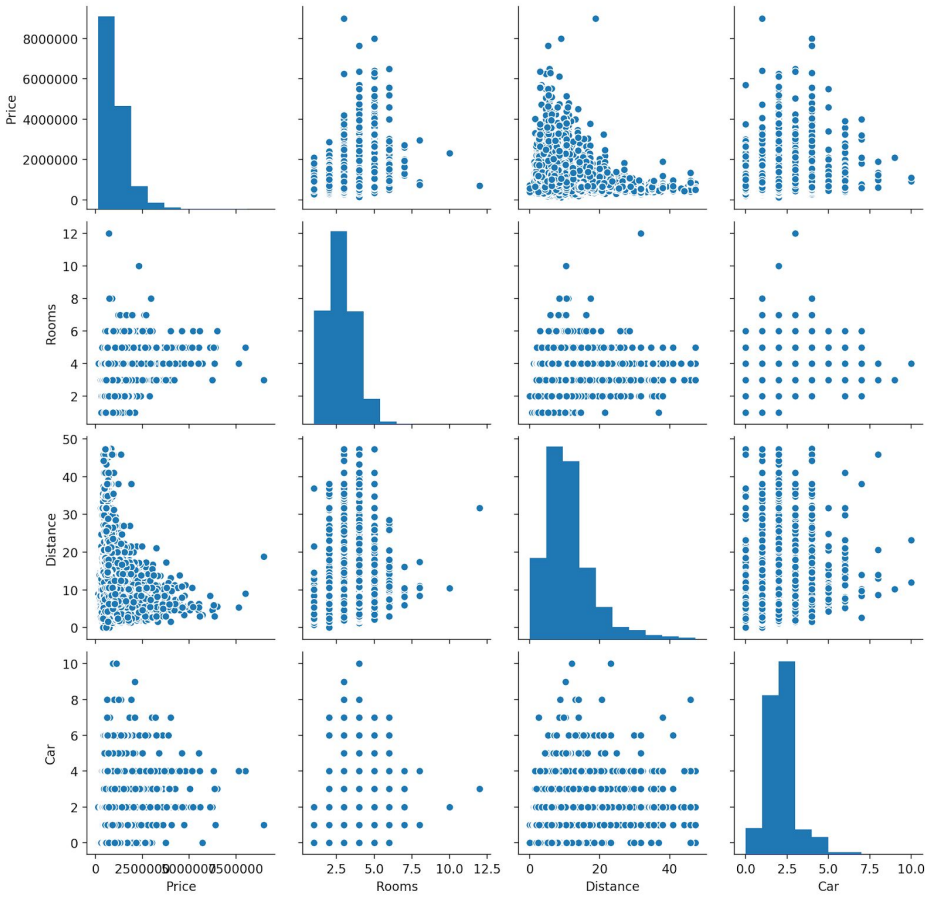


Explanatory Analysis

The purpose of this section is to help us better understand the data from different perspectives. First thing that we did was to drop some rows that contain nulls. Then we study the distribution of the dependent variable. We can do that by plotting the distribution. We can see that the price seems nicely normal-distributed.



Intuitively speaking, we think that the price of a house is mostly related to the number of rooms, the location(how far it is from the main business district) and the number of parking spots it has. So we plot them against each other,



From this plot, we can see that our intuition is not well-reflected. Namely, there's no direct linear relationship between any of these variables just by looking at the plot. However, this needs to be further validated in the later sections.

There were 1060 rows with some strange data points where Landsize, BuildingArea equal to 0. It’s impossible for there to be a building if the land size or building area of 0 so 0 may be a default value when none is given. So these points may influence our model.

Extra interpretation and creativity (1 point of 16 points):

We are also given longitude and latitude, we can get the region of this dataset by making a scatterplot,

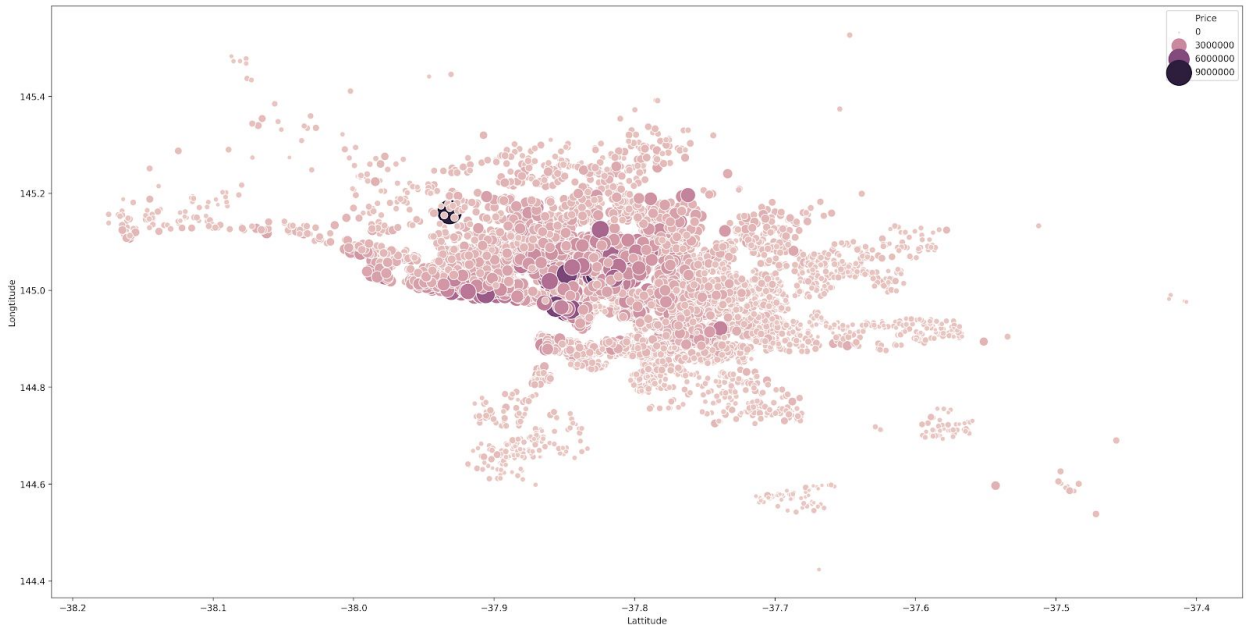


Figure: 2D scatter plot. X=latitude, y=longitude, size\color=price.

We can see that most expensive houses are centered in one area, this is a good sign of possible relationship between house price and longitude&latitude. We can further investigate the black dot, i.e. the most expensive house using google map, it's a quite nice house. Credit @googlemap2020.

Figure: The most expensive house from Google map

To visualize this from a different dimension. We can make a simple 3D scatter plot.

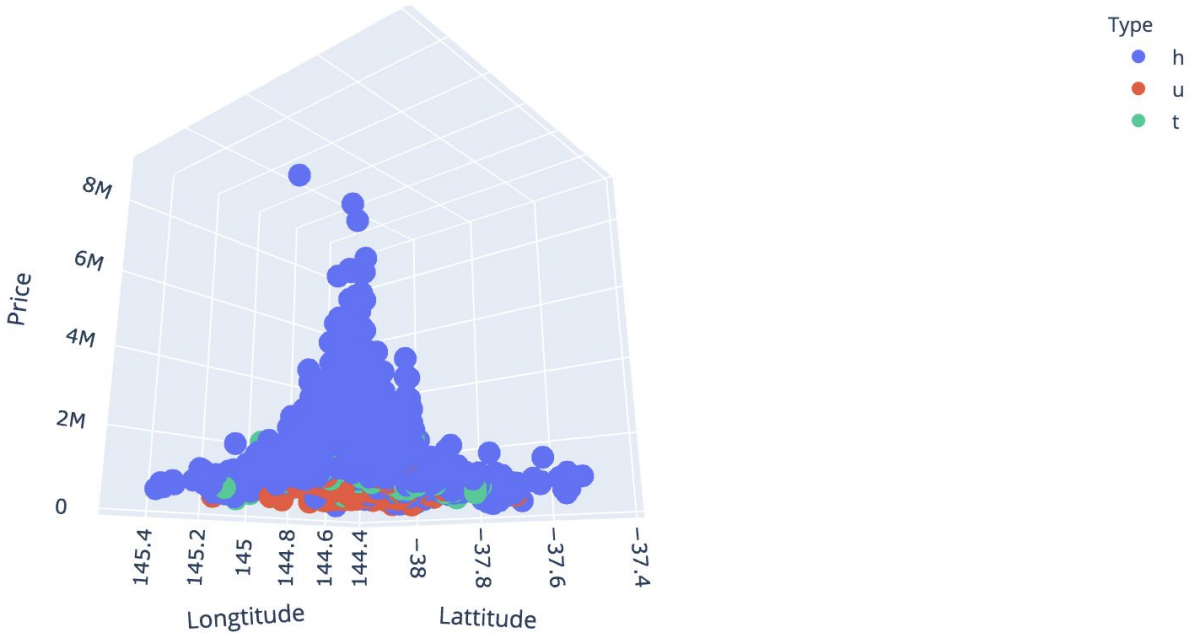


Figure: 3D scatter plot. X=longitude, y=latitude, z=price, color=Type.

As we can see, high priced houses are clustered in the center and most of them are of type h, which stands for house,cottage,villa, semi,terrace.

Next, we can study the price and how it is related to different categorical variables.

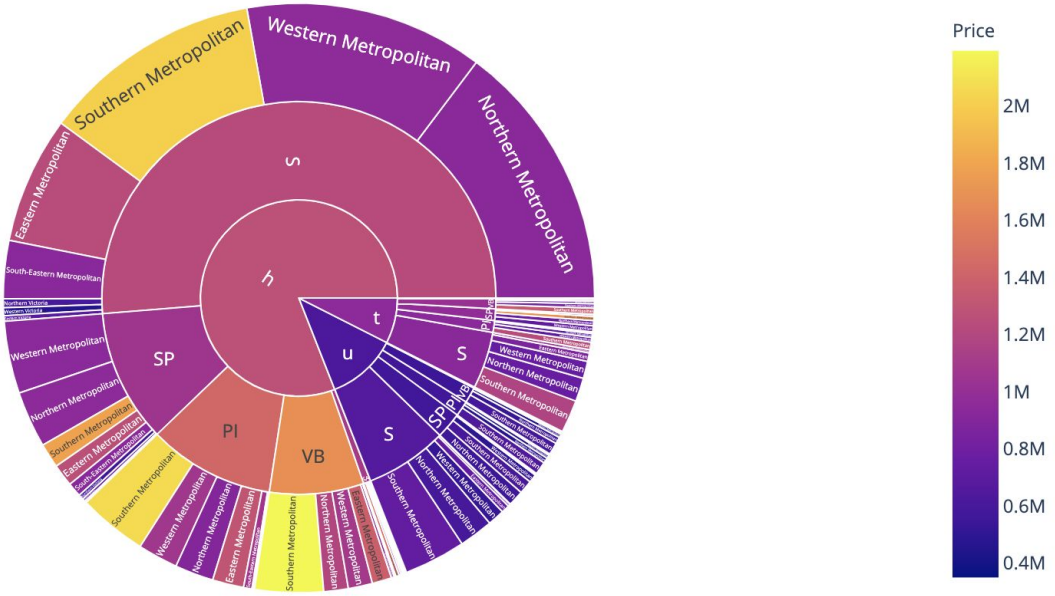


Figure: Sunburst plot of different categorical variables where the color is represented by the color. We can see that under Type='h', Method='VB' and Regionname='Southern Metropolitan', the houses are the most expensive.

Last but not least, we can analyze the relationship between price, rooms and land size. We can see that in this region we don't have a lot of houses with large landsize and the price cannot be determined well by the landsize and rooms. Number of carsplots is also not a good indicator of price as there's no clear pattern.

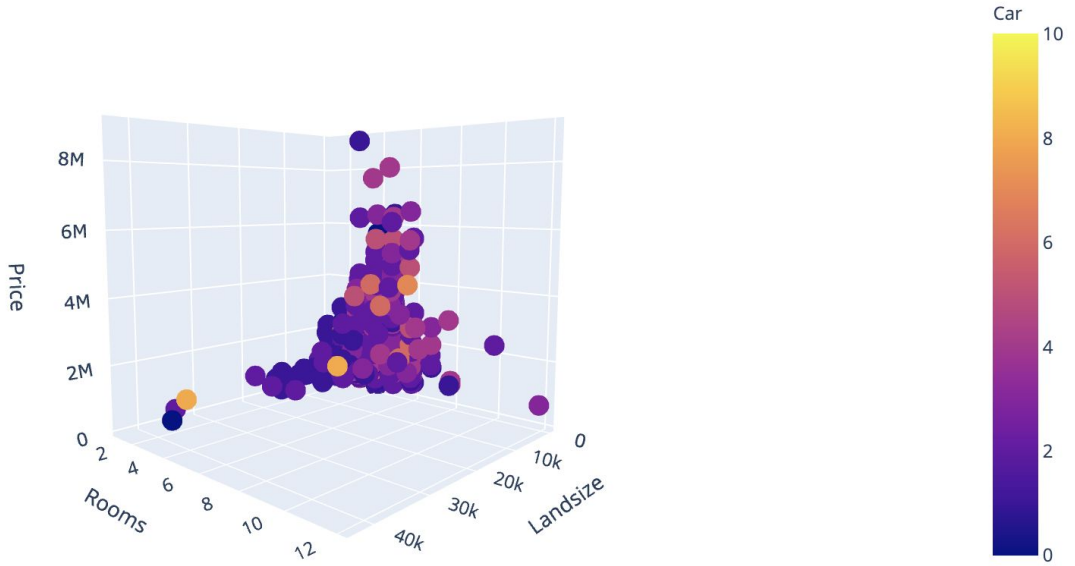


Figure: 3D scatterplot. X=landsize, y=rooms, z=price, color=Car.

Initial Model 1's Evaluation and Initial Model 2's Evaluation

Correlation

Before constructing the model, we started by checking the correlation between different predictors as shown below.

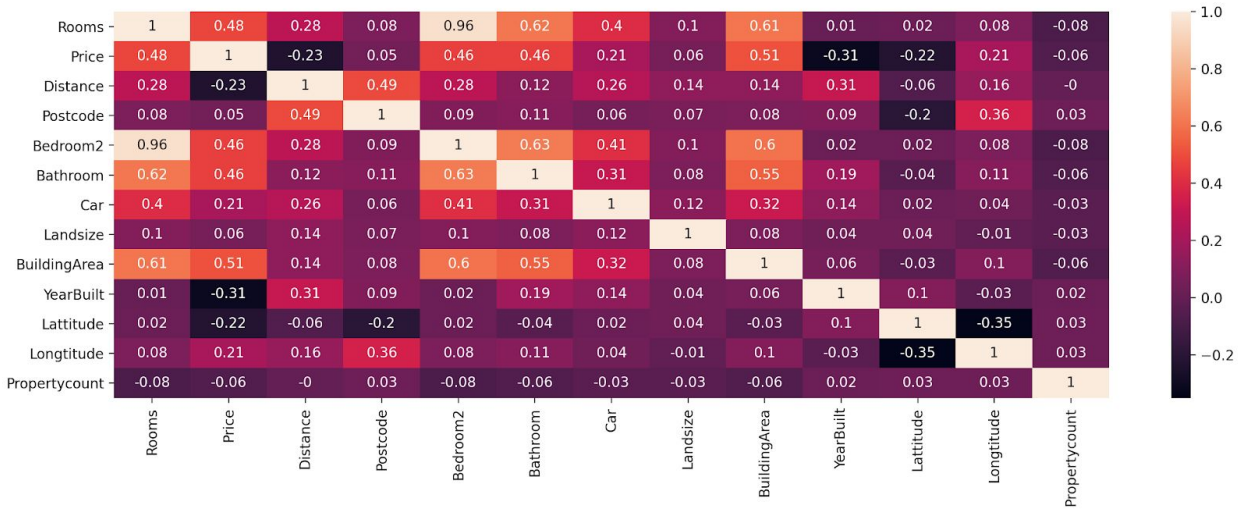


Figure: Heatmap of the correlations

	Rooms	Price	Distance	Postcode	Bedroom2	Bathroom	Car	Landsize	BuildingArea	YearBuilt
Rooms	1.000000	0.480000	0.280000	0.080000	0.960000	0.620000	0.400000	0.100000	0.610000	0.010000
Price	0.480000	1.000000	-0.230000	0.050000	0.460000	0.460000	0.210000	0.060000	0.510000	-0.310000
Distance	0.280000	-0.230000	1.000000	0.490000	0.280000	0.120000	0.260000	0.140000	0.140000	0.310000
Postcode	0.080000	0.050000	0.490000	1.000000	0.090000	0.110000	0.060000	0.070000	0.080000	0.090000
Bedroom2	0.960000	0.460000	0.280000	0.090000	1.000000	0.630000	0.410000	0.100000	0.600000	0.020000
Bathroom	0.620000	0.460000	0.120000	0.110000	0.630000	1.000000	0.310000	0.080000	0.550000	0.190000
Car	0.400000	0.210000	0.260000	0.060000	0.410000	0.310000	1.000000	0.120000	0.320000	0.140000
Landsize	0.100000	0.060000	0.140000	0.070000	0.100000	0.080000	0.120000	1.000000	0.080000	0.040000
BuildingArea	0.610000	0.510000	0.140000	0.080000	0.600000	0.550000	0.320000	0.080000	1.000000	0.060000
YearBuilt	0.010000	-0.310000	0.310000	0.090000	0.020000	0.190000	0.140000	0.040000	0.060000	1.000000
Lattitude	0.020000	-0.220000	-0.060000	-0.200000	0.020000	-0.040000	0.020000	0.040000	-0.030000	0.100000
Longtitude	0.080000	0.210000	0.160000	0.360000	0.080000	0.110000	0.040000	-0.010000	0.100000	-0.030000
Propertycount	-0.080000	-0.060000	-0.000000	0.030000	-0.080000	-0.060000	-0.030000	-0.030000	-0.060000	0.020000

Figure: correlation for a subset of the predictors are shown. correlations in yellow have correlations >0.8. It is easier to see the highly correlated predictors here between Bedroom2 and Rooms.

We can see that there is a high correlation between Bedroom2 and Rooms. To further validate this, we should check the variance inflation factors(VIFs). To do that, we should fit the full model.

Initial model 1_0 evaluation. model 1_0 is initial model 1 and with influential points and before transform

Checking multicollinearity

13	1.54	C(Regionname)[T.Western …
14	15	Rooms
15	2.44	Distance
16	14.75	Bedroom2
17	2.02	Bathroom

Figure: A subset of the predictors is shown. We can see Rooms and Bedroom2 have VIF >10

We can see rooms and bedrooms have VIF >10 so there is high multicollinearity for those 2 variables. In practice, VIF<10 is ok based on lecture 7 part a. Therefore, it is acceptable that for Regionname predictor, Northern Metropolitan, Southern Metropolitan, Western Metropolitan has moderate multicollinearity since $4<VIF_j<10$.

To fix this problem, we need to drop one of the two variables. We decided to drop Rooms and keep Bedroom2 because Bedroom is theoretically more important than Rooms (which may include living room, formal dining room) when predicting price.

1	1.29	C(Type)[T.t]
2	1.86	C(Type)[T.u]
3	2.32	C(Method)[T.S]
4	1.06	C(Method)[T.SA]
5	1.94	C(Method)[T.SP]
6	1.62	C(Method)[T.VB]
7	1.29	C(Regionname)[T.Eastern …
8	4.85	C(Regionname)[T.Northern…
9	1.38	C(Regionname)[T.Northern…
10	2.22	C(Regionname)[T.South-Ea…
11	3.87	C(Regionname)[T.Southern…
12	6.69	C(Regionname)[T.Western …
13	1.54	C(Regionname)[T.Western …
14	2.44	Distance
15	2.62	Bedroom2
16	2.01	Bathroom
17	1.28	Car
18	1.07	Landsize
19	1.76	BuildingArea
20	1.63	YearBuilt
21	2.78	Lattitude
22	3.88	Longtitude
23	1.14	Propertycount

Figure: After dropping Rooms, the problem of multicollinearity is considered as resolved, i.e. all VIFs were smaller than 10.

Fitting an initial model 1_0(with influential points and before transformation).

The full model we used:
full_model='Price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+Longtitude+ C(Regionname)+ Propertycount

We got an R-squared of 0.646 for this model which can be improved by subsequent work. We will use this number again in our comparison of models' goodness of fit.

Next, we will discuss evaluation of model assumptions such as normality, heteroskedasticity and linearity.

Checking normality

When we checked normality, here were the test statistics we got Omnibus: 6538.718. Jarque-Bera (JB): 493185.215, which were very high and the p values for both tests were 0.00. We will compare these test statistics again to show subsequent models have improved normality.

Checking Heteroskedasticity

Next we check Heteroskedasticity (non equal variance) using the Breusch-Pagan (BP) test. The p-value was 8.26e-135 for model1_0 (with influential point and before transform), suggesting that there Is significant heteroskedasticity. We will use this p value for comparison to show subsequent models have improved heteroskedasticity.

Checking linearity

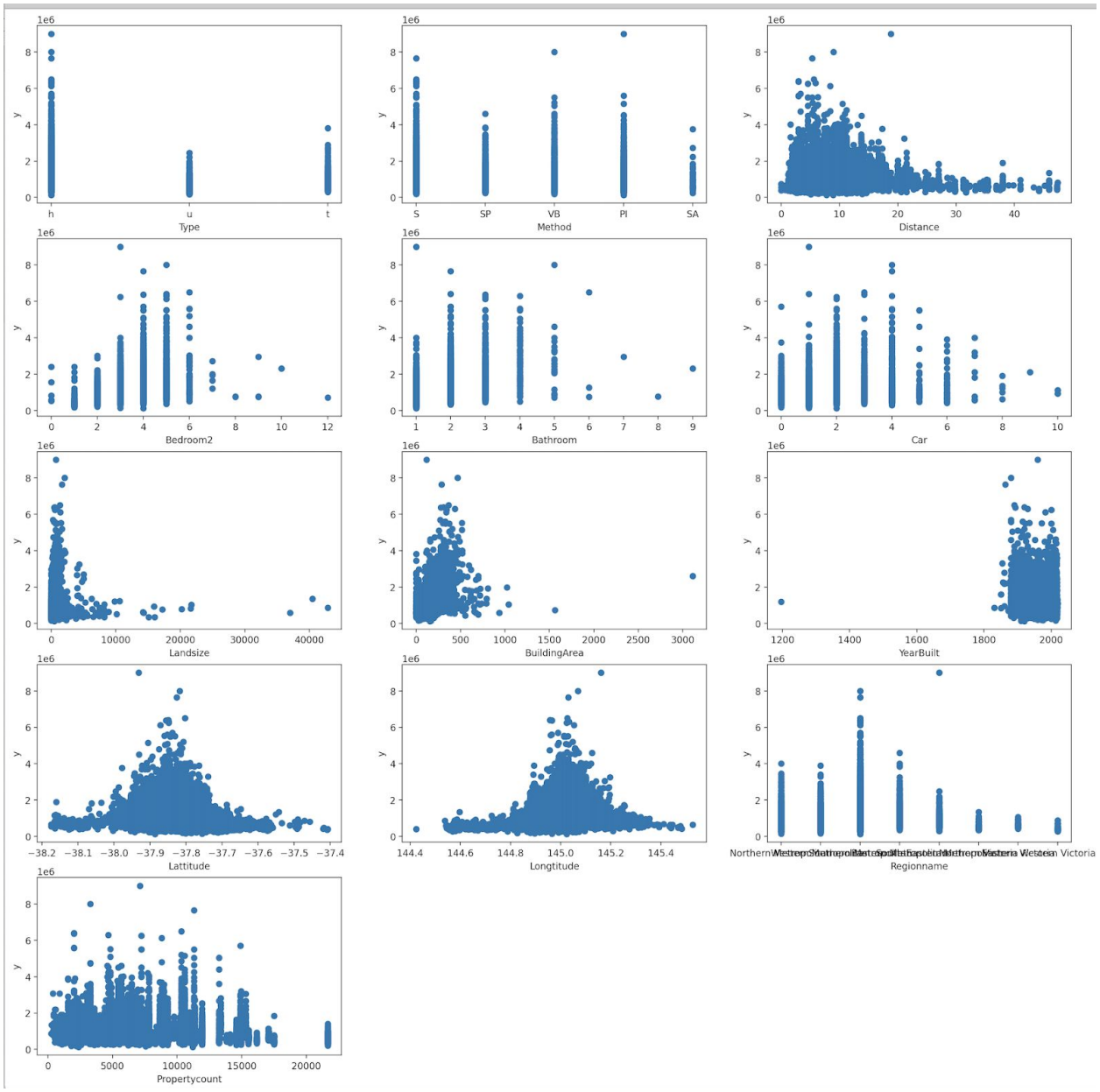


Figure: checking linearity with predictors vs price scatterplot

Next we check linearity assumption based on y vs predictor scatterplot.

- All plots look nonlinear. We can see possible influential points in Landsize,
- BuildingArea, YearBuilt and the influential points are making the range of the x axis larger.

Checking and removing influential points. Initial model 1_1 evaluation. model 1_1 is initial model 1 and **without** influential points and before transform

Using initial model 1_0 (with influential points and before transform) and cook's distance, we identified and removed 390 influential points. So our new DataFrame's dimension is 8497x21.

We then fit initial model1_1 (**without** influential points and before transform)

- R-squared=0.646 with influential points from model1_0
- R-squared =0.756 without influential points from model1_1

Therefore, getting rid of influential points leads to better fit.

Comparison between the influence plots before and after removing influential points

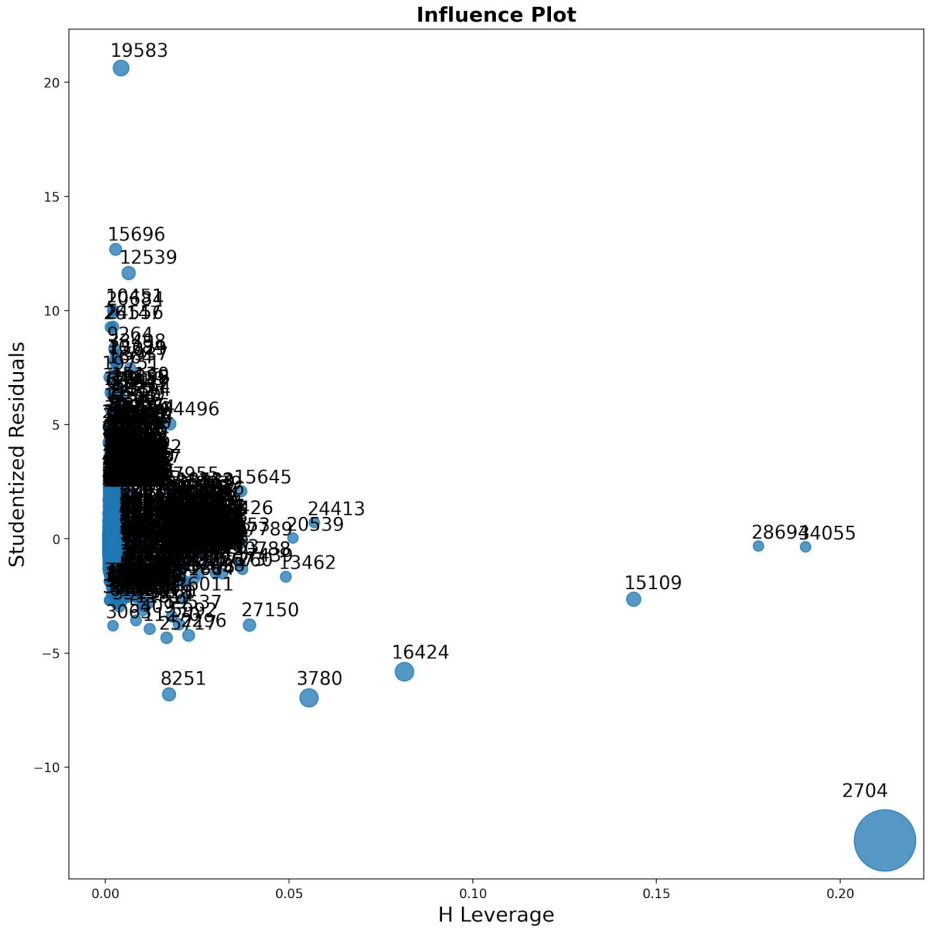


Figure: influence plot of model1_0 (with influential points). Cook's distance is the size of a dot.

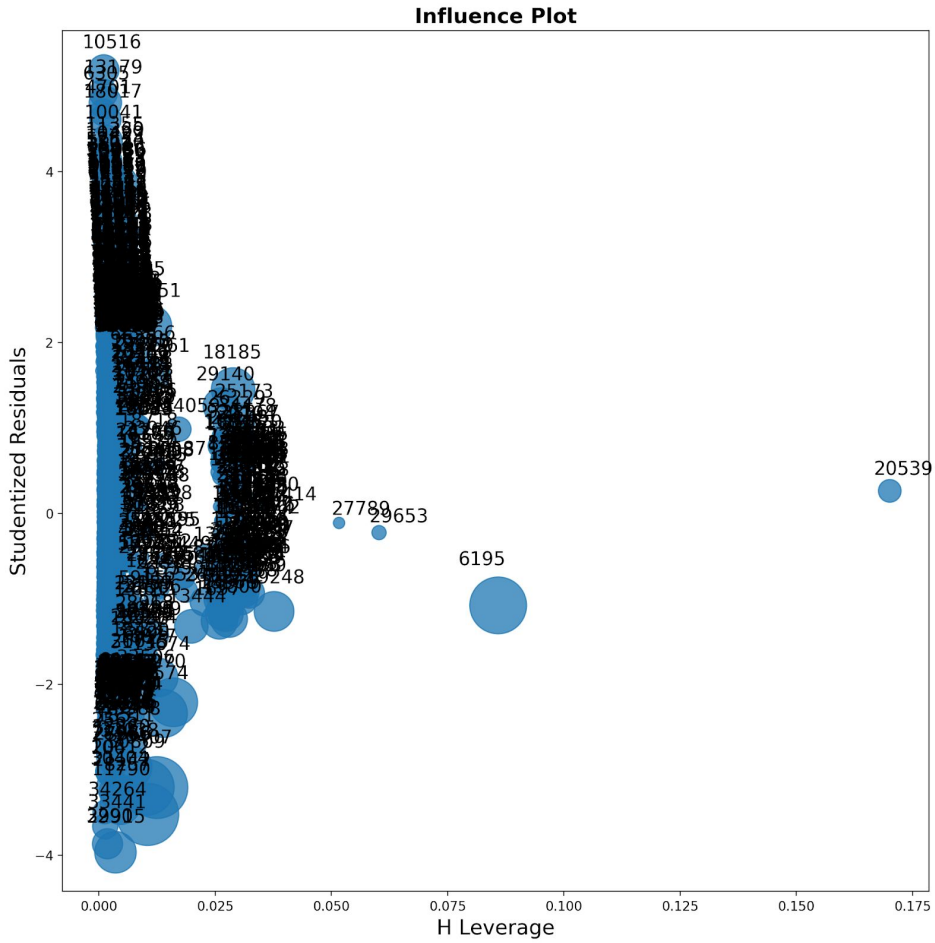


Figure: influence plot of model1_1 (**without** influential points). Cook's distance is the size of a dot.

Based on the comparison of the influence plot and looking at the scale of the x and y axis between the 2 plots. We can see that removing influential points using Cook's distance has culled extreme influential points. For example, observations (python index) 19583 and 2704 have high studentized residuals and leverage, respectively.We no longer see those 2 points as well as other influential points in our new DataFrame.

Checking normality (with multiple ways)

Omnibus test combines skewness and kurtosis.
from the summary table, Omnibus p value <0.05 so we Conclude H1: residuals are significantly non-normal. Violates non-normal assumptions. Kurtosis is relatively high at 4.5.
JB p value <0.05. Conclude: H1: residuals are significantly non-normal.
Note JB test is very sensitive to outliers.

Test statistics for normality have improved Substantially after removing influential points.
Model with influential point:

- Omnibus: 6538.718.
- Jarque-Bera (JB): 493185.215

Model without influential point

- Omnibus 838.630
- Jarque-Bera (JB):1487.364

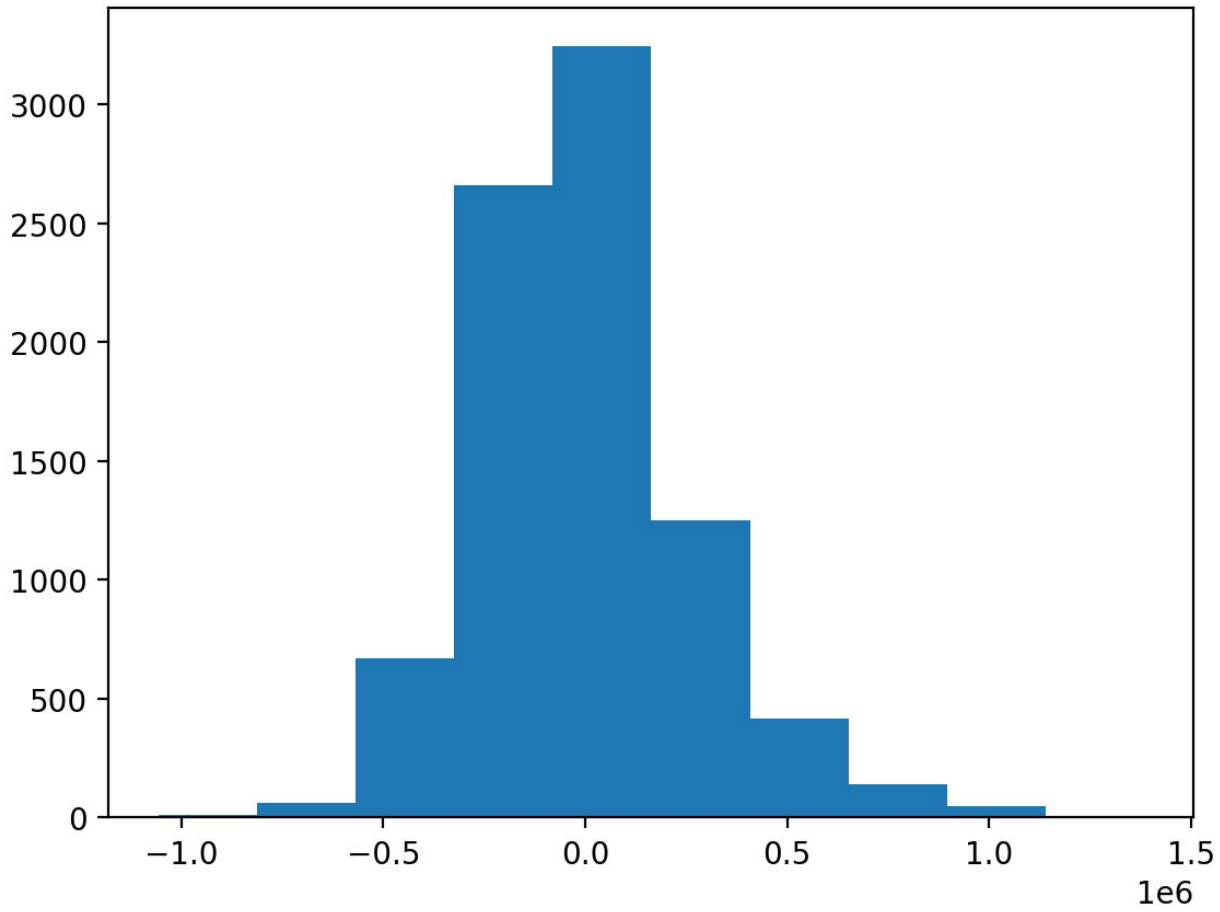


Figure: For model without influential point, here is the histogram of residual to observe the shape/skewness. Unimodal, approximately symmetric.

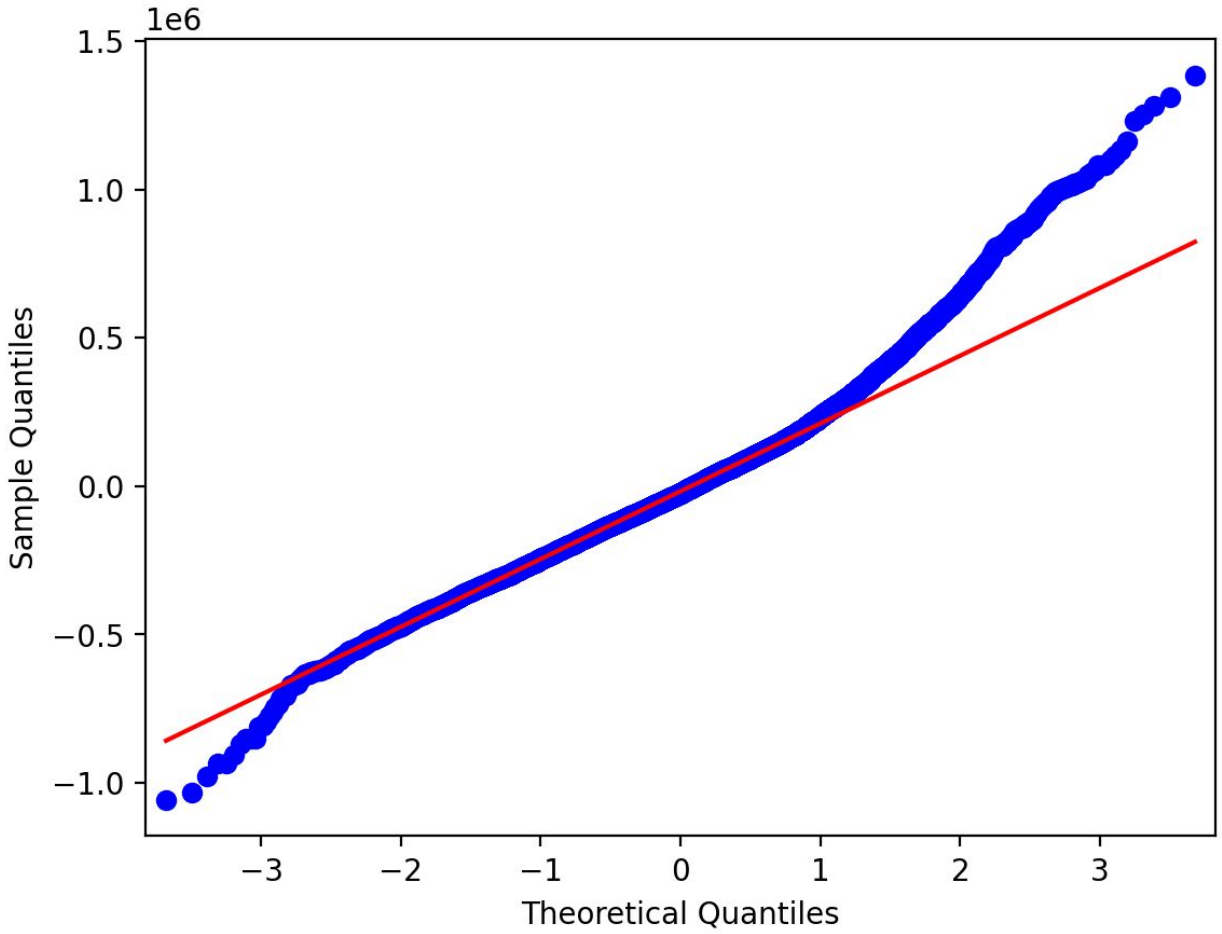


Figure: QQ plot to check for normality. Curve away in opposite directions at both ends from straight line suggests heavy tail/kurtosis.

omnibus test, JB test, QQplot all agree and suggest normal assumption is violated. However, our sample size is 8497 even with the initial model with influential points removed so CLT guarantees that residuals are approximately normally distributed.

Checking Heteroskedasticity

Breusch-Pagan is for "systematic" Heteroscedasticity. A random pattern could pass the BP test. P value=6.3e-212, so p value<<0.05 so there is significant Heteroscedasticity, even after removing influential points.

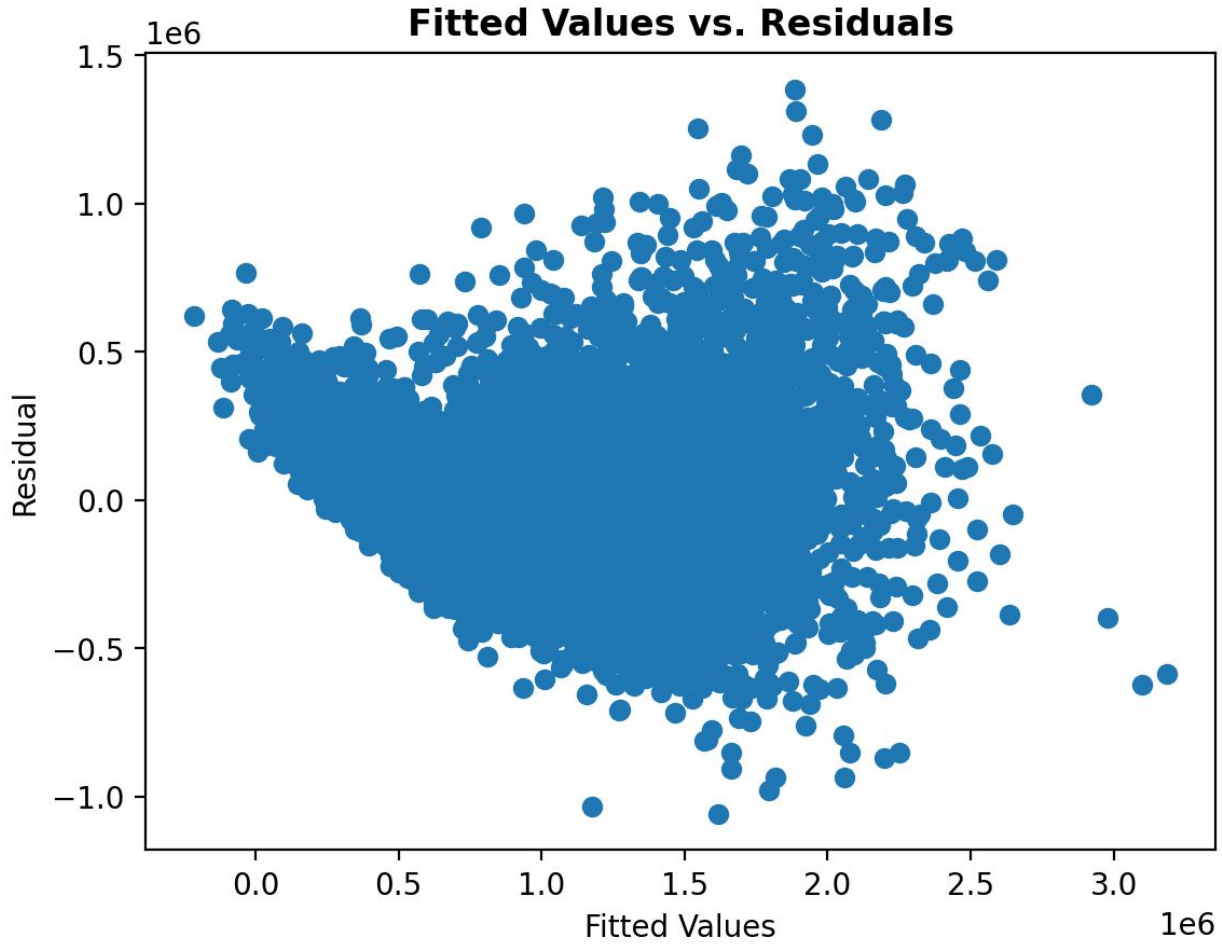


Figure: Fitted Values vs. Residuals Plot

The residual plot agrees with BP that there is heteroscedasticity. There is a funnel shape(cluster in beginning and spread out at end) where the band is smaller for smaller fitted values and band is larger for larger fitted values.

Transformation

Transform response variable

Because we saw significant Heteroskedasticity and nonlinearity in the initial model with influential points, we will do Transformation. First, let's transform our response variable price to become log_price. And we get a new model

```
full_model = 'log_price~C(Type)+C(Method)+ Distance + Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+Longtitude+ C(Regionname)+ Propertycount'
```

Check Heteroskedasticity before and after transform.

Original BP p value=6.3e-212 before log transform on y. After log transform on y, p value= 1.99e-35 so the degree of 'systematic' heteroscedasticity seems to be reduced although p value is <0.05 so it suggests there is still heteroscedasticity.

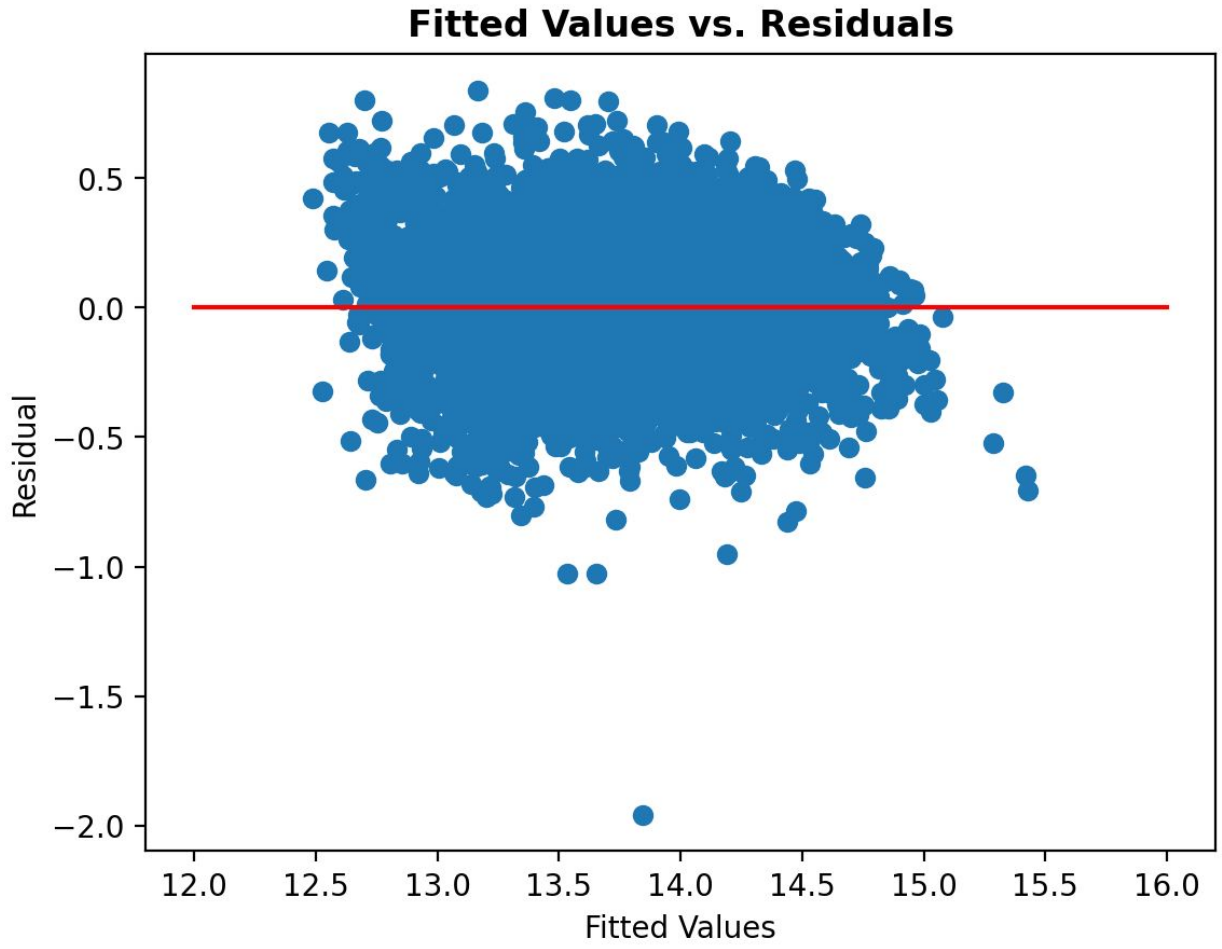


Figure: Fitted Values vs. Residuals Plot

Comment: the band width looks much more constant across the fitted values after log transform on y vs the original looked like a funnel shape. From the residual plot, we can also comment on the linearity. The values look approximately evenly distributed about the 0 line.

Transform selected predictors

We will transform selected predictors to improve the linearity assumption since before transformation, all scatter plots of predictors and price were nonlinear.

Check linearity before and after log transform

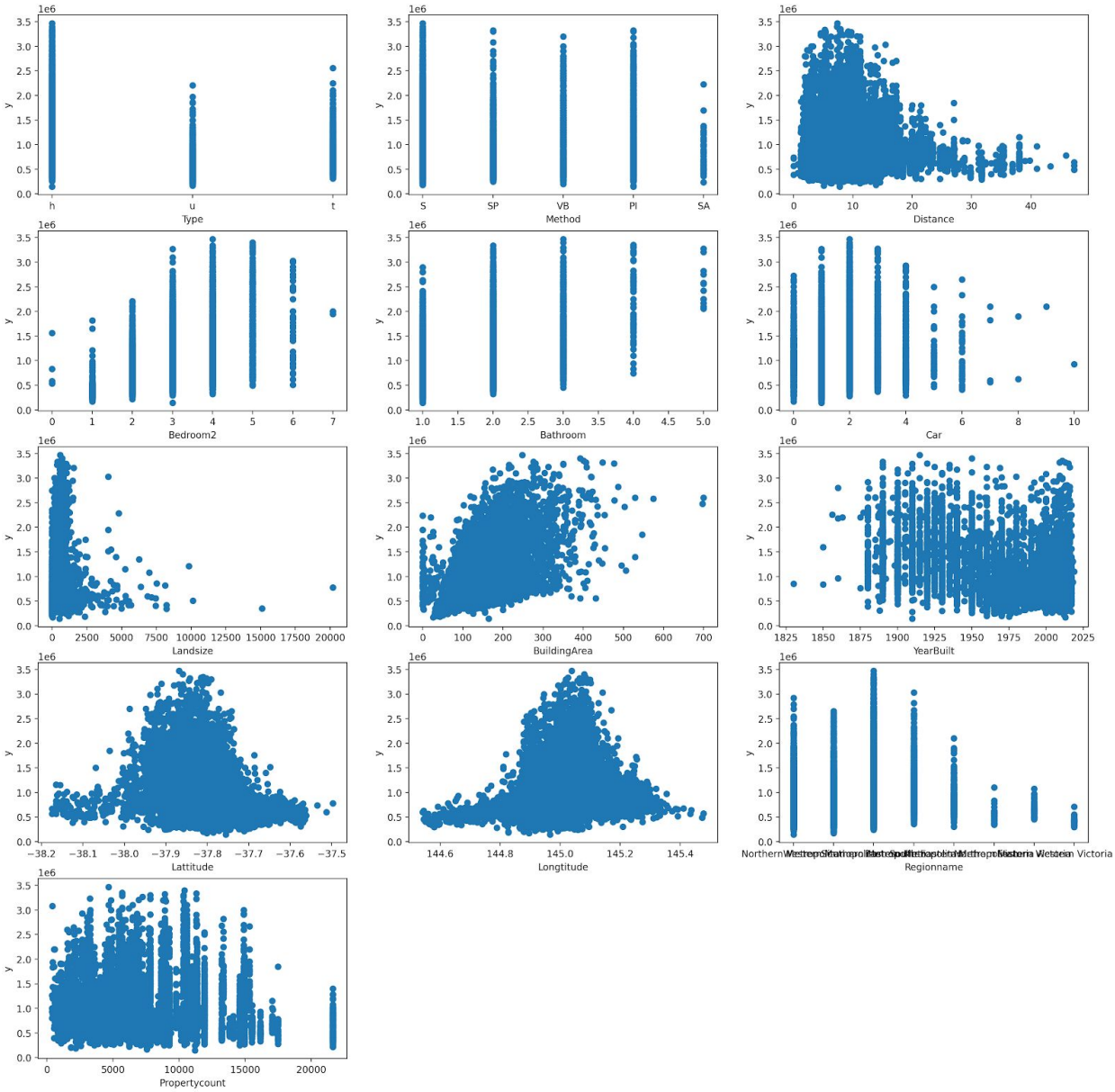


Figure: scatter plot before log log transform. All look nonlinear.

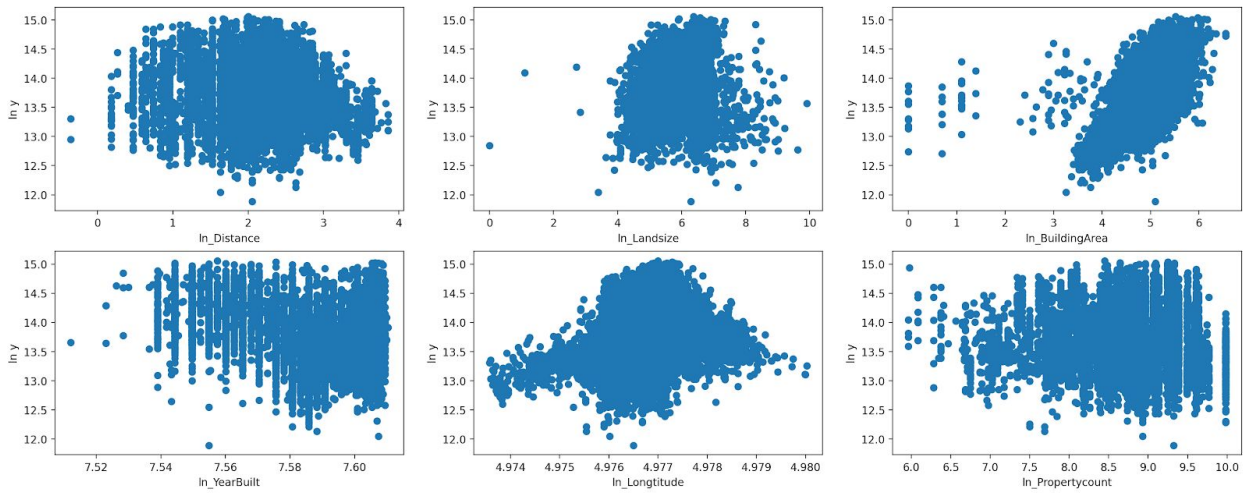


Figure: scatter plot after log log transform. Linearity looks improved although there still seems to be a nonlinear phenomenon

We only log transform the quantitative predictors . 'Lattitude' has negative number so we can't log transform it. log transform YearBuilt didn't seem to help so keep original YearBuilt without log transform for new model. After the log transformation, linearity assumption is better graphically. we haven't learned a numerical way to evaluate linearity.

Fit Initial Model 2_0 with Influential Points and After Transformation

Note:
log_Landsize, log_BuildingArea, long-Distance causing LinAlgError since there are 0s in Landsize, BuildingArea and Distance. Therefore, keep Landsize and BuildingArea without transformation in initial model 2.

Model2_0 with influential points and after log log transform of selected variables
Full_model='"log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname)+ log_Propertycount"'

Before transform:

- R-squared = 0.646 with influential points. For model1_0.
- R-squared = 0.756 without influential points. For model1_1.

Getting rid of influential points leads to better fit.

After transform:

- R-squared = 0.797 for model2_0.

Model after transform has a slightly better fit compared to models before transformation.

Check multicollinearity

1	1.34	C(Type)[T.t]
2	1.97	C(Type)[T.u]
3	2.32	C(Method)[T.S]
4	1.04	C(Method)[T.SA]
5	1.95	C(Method)[T.SP]
6	1.61	C(Method)[T.VB]
7	1.23	C(Regionname)[T.Eastern …
8	4.94	C(Regionname)[T.Northern…
9	1.23	C(Regionname)[T.Northern…
10	2.22	C(Regionname)[T.South-Ea…
11	3.87	C(Regionname)[T.Southern…
12	6.97	C(Regionname)[T.Western …
13	1.48	C(Regionname)[T.Western …
14	2.39	Distance
15	2.84	Bedroom2
16	2.11	Bathroom
17	1.29	Car
18	1.09	Landsize
19	2.57	BuildingArea
20	1.76	YearBuilt
21	2.75	Lattitude
22	3.98	log_Longtitude
23	1.1	log_Propertycount

24 rows × 2 columns

Figure: All VIF <10 for model2_0 with influential points and after log log transform. So no high multicollinearity.

Check and remove influential points. Initial model 2_1 evaluation. model 2_1 means initial model 2 and without influential points and after transform

451 influential points found using Cook's distance with threshold of 4/n, where n is the number of samples. And we end up with a new DataFrame of 8046x21 dimension.

model2_1 without influential points
full_model='"log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname) + log_Propertycount"'

Before transform:

- R-squared=0.646 with influential points from model1_0
- R-squared =0.756 without influential points from model1_1

After transform:

- R-squared =0.797 for model2_0. With influential points
- R-squared =0.838 for model2_1. Without influential points

The r-squared suggests without influential points and after transfer has the highest r-squared compared to previous models.

Checking normality (with multiple ways)

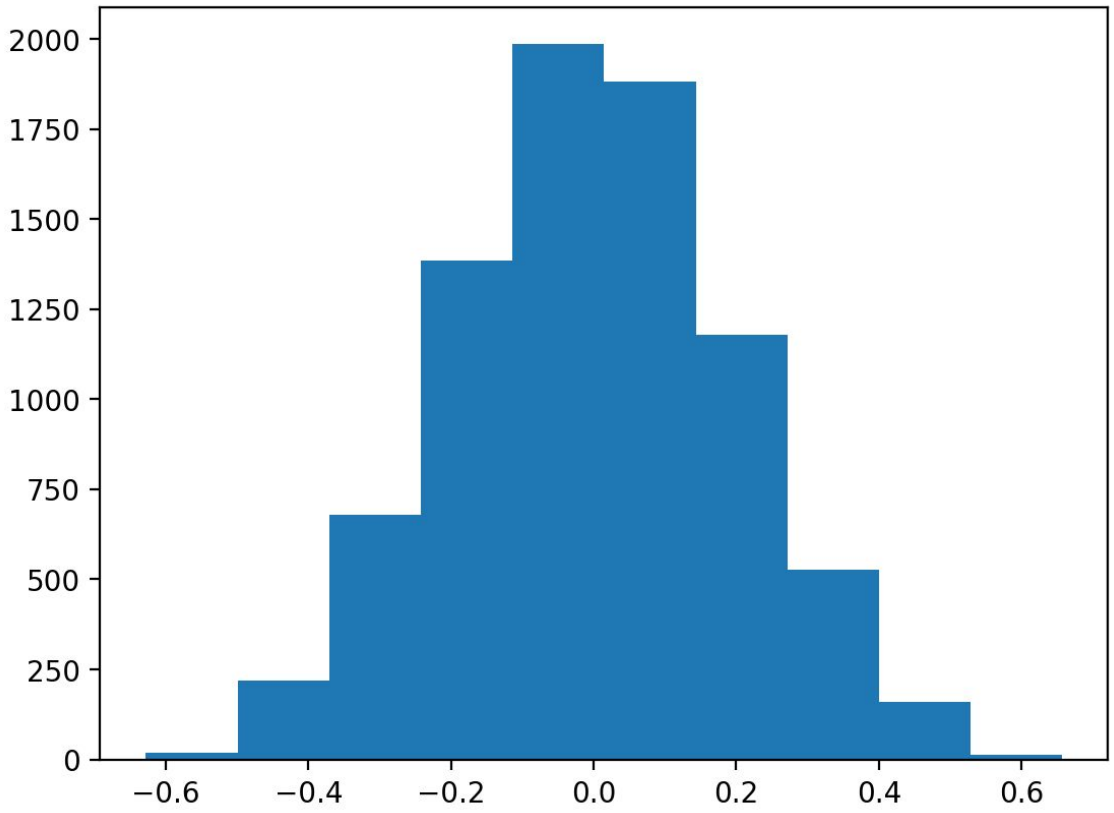


Figure: histogram of residuals to observe shape/skewness

Comments: unimodal, approx symmetric,

Comparison:

- model1_0(with influential points), Omnibus test stat is 6538.718 and JB test stat is 493185.215.
- model1_1(without influential points), omnibus test statistic is 838 and JB test statistic is 1487.
- model2_1(without influential points, transformed), omnibus test statistic is 31 and JB test statistic is 24. So normality assumption is improved significantly without influential points and improved further with transformation.

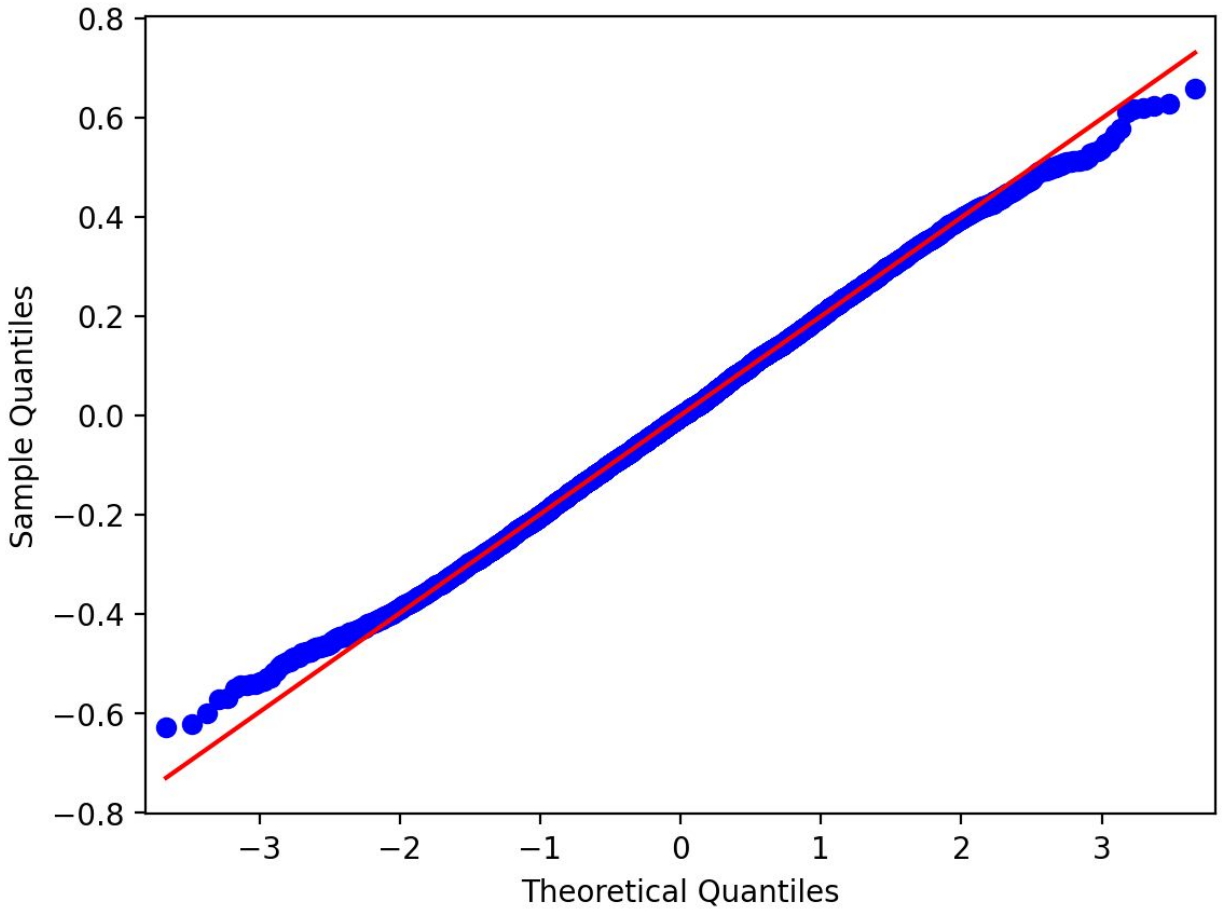


Figure: QQplot of model2_1 without influential points and after transform

QQ plot to check for normality. Although the scale of the axis is different for this plot, there appears to be less curvature at both ends in model2_1 compared to model1_1. So the normality assumption is improved. However, there is still some slight curvature

Omnibus test, JB test, QQplot all agree and suggest normal assumption is violated. However, our sample size is ~8000 even with the initial model with influential points removed. So the central limit theorem(CLT) guarantees that residuals are approximately normally distributed.

Checking Heteroscedasticity

Breusch-Pagan is for "systematic" Heteroscedasticity. A random pattern could pass the BP test so also need to do a residual plot.

Comparison:

BP p-value 8.26e-135 for model1_0 with influential points and before transform after removing influential point and log transform on y with model2_1, p value= 2.0e-52 so the degree of 'systematic' heteroscedasticity seems to be reduced although there is still significant Heteroscedasticity.

Checking Heteroscedasticity and Nonlinearity

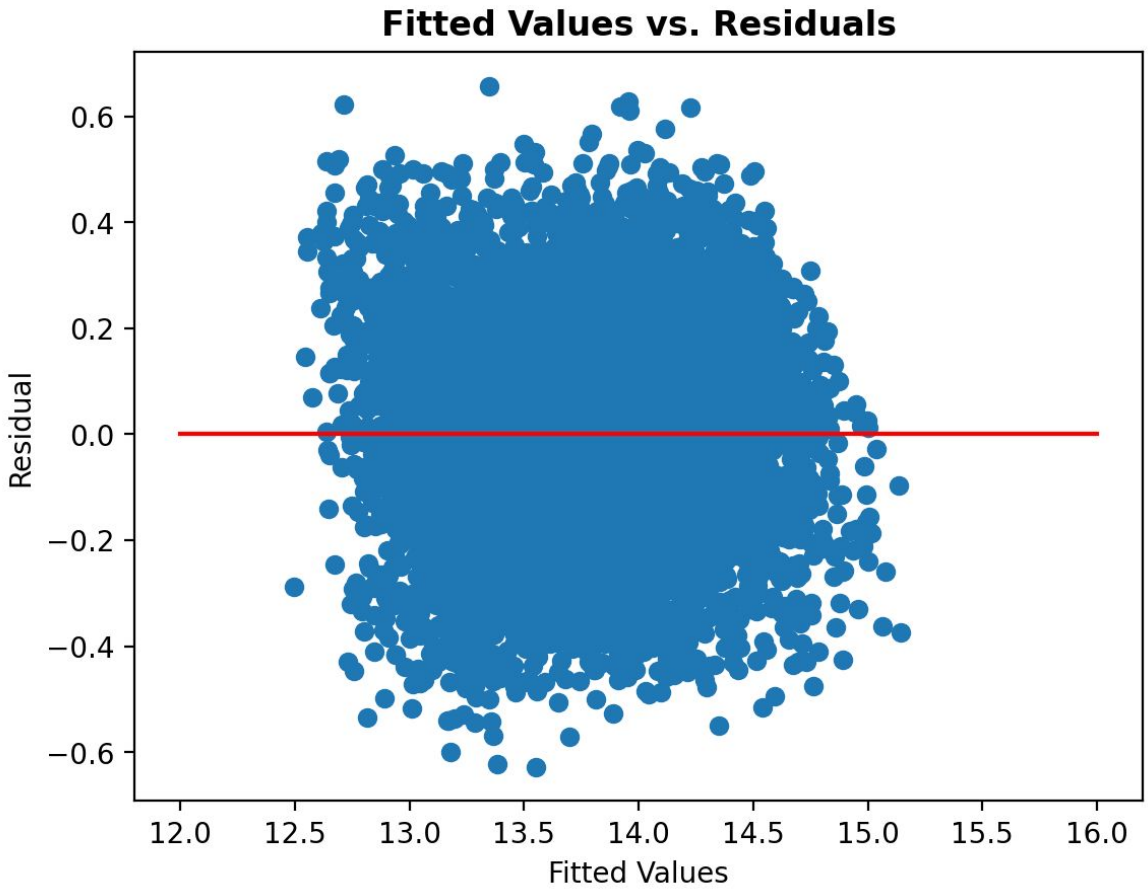


Figure: Fitted Values vs. Residuals Plot

Comment:

- The band width looks much more constant across the fitted values after log transform on y.
- From the residual plot, we can also comment on the linearity. The values look approximately evenly distributed about the 0 line.

Checking linearity

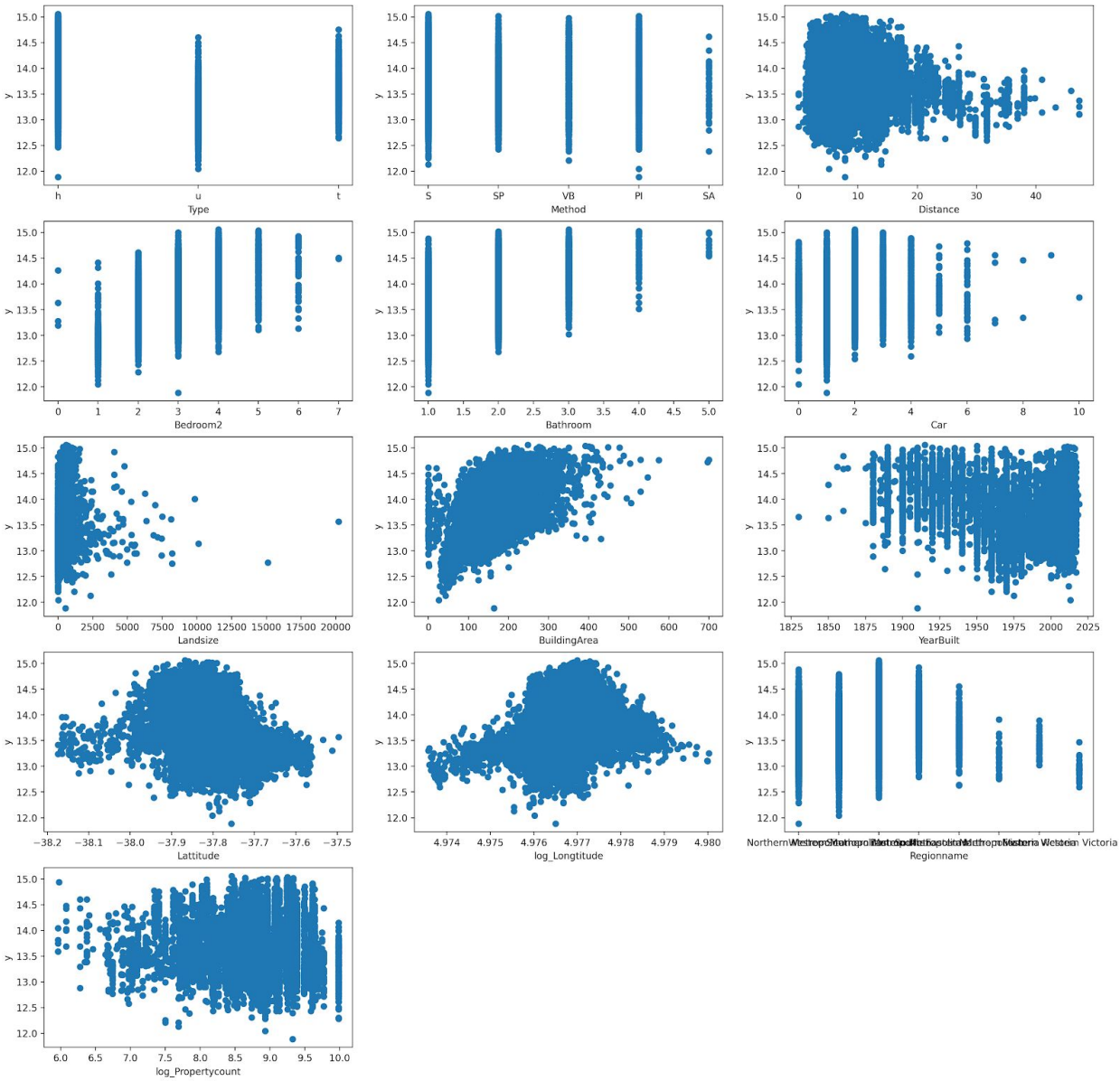


Figure: checking linearity with scatter plot after removing influential points and after transforming price and selected predictors.

Comparison:

when influential points were included and before log transform, all plots were nonlinear. After removing influential points and log transform price, longitude, propertycount, linearity is improved. linearity is improved for log_Longtitude and log_Propertycount. For other variables, we were unable to do a log transformation because some of the variables contain zero. Which resulted in an LinAlgError when fitting the model.

Model Selection

Since there are too many possible subsets, we use Stepwise. Choose AIC as criterion. Start with full_model and stepwise from both directions.

Model without influential points

```
model12_1 = 'log_price~C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname) + log_Propertycount
model12_2 = 'log_price~C(Type)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname) + log_Propertycount'
model12_3 = 'log_price~C(Type)+C(Method)+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname) + log_Propertycount'
model12_4 = 'log_price~C(Type)+C(Method)+ Distance+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname) + log_Propertycount'
model12_5 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname) + log_Propertycount'
model12_6 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname) + log_Propertyc
model12_7 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname) + log_Propertycount'
model12_8 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname) + log_Propertycount'
model12_9 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ Lattitude+log_Longtitude+ C(Regionname) + log_Propertycount'
model12_10 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+log_Longtitude+ C(Regionname) + log_Propertycount'
model12_11 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+ C(Regionname) + log_Propertycount'
model12_12 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ log_Propertycount'
model12_13 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname)'

for i in range(1,14):
    print(f'aic for model12_{i} is: ',smf.ols(eval(f'model12_{i}'),data=df_without_infl_pt).fit().aic)
```

```
aic for model12_1 is:  -630.3707020384572
aic for model12_2 is:  -3064.6554255333867
aic for model12_3 is:  424.72180462126926
aic for model12_4 is:  -2858.7174791379584
aic for model12_5 is:  -3133.0117240509826
aic for model12_6 is:  -3189.941106803606
aic for model12_7 is:  -3235.4737793249515
aic for model12_8 is:  -1678.7430275622246
aic for model12_9 is:  -2438.8693673757807
aic for model12_10 is:  -2815.1609445368813
aic for model12_11 is:  -2641.511847628957
aic for model12_12 is:  -2414.3303121807803
aic for model12_13 is:  -3287.5484938885165
```

model12_13 has the smallest aic -3287.5

choose model12_13: log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname)

```
#drop log_Propertycount
full_model='log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname) + l
model11_1 = 'log_price~C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname)'
model11_2 = 'log_price~C(Type)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname)'
model11_3 = 'log_price~C(Type)+C(Method)+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname)'
model11_4 = 'log_price~C(Type)+C(Method)+ Distance+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname)'
model11_5 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname)'
model11_6 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom +Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname)'
model11_7 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car + BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname)'
model11_8 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname)'
model11_9 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ Lattitude+log_Longtitude+ C(Regionname)'
model11_10 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt +log_Longtitude+ C(Regionname)'
model11_11 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+ C(Regionname)'
model11_12 = 'log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude'

print(f'aic for full_model is: ',smf.ols(full_model,data=df_without_infl_pt).fit().aic)
for i in range(1,13):
    print(f'aic for model11_{i} is: ',smf.ols(eval(f'model11_{i}'),data=df_without_infl_pt).fit().aic)
```

```
aic for full_model is:  -3285.905460088459
aic for model11_1 is:  -626.0636974892659
aic for model11_2 is:  -3066.130447028212
aic for model11_3 is:  428.78816466192075
aic for model11_4 is:  -2860.1889178800557
aic for model11_5 is:  -3134.6923493242684
aic for model11_6 is:  -3191.643899985964
aic for model11_7 is:  -3236.9394232536542
aic for model11_8 is:  -1680.6774427932542
aic for model11_9 is:  -2439.9748302141343
aic for model11_10 is:  -2814.733430916167
aic for model11_11 is:  -2642.4199034599733
aic for model11_12 is:  -2415.159431031083
```

Aic all become larger. Stop.

Finally pick model12_13.

log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname)

```
smf.ols(model12_13,data=df_without_infl_pt).fit().summary()
```

OLS Regression Results

Dep. Variable:	log_price	**R-squared:**	0.838
Model:	OLS	**Adj. R-squared:**	0.838
Method:	Least Squares	**F-statistic:**	1890.
Date:	Mon, 30 Nov 2020	**Prob (F-statistic):**	0.00
Time:	08:21:35	**Log-Likelihood:**	1666.8
No. Observations:	8046	**AIC:**	-3288.
Df Residuals:	8023	**BIC:**	-3127.
Df Model:	22		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-757.1181	28.664	-26.414	0.000	-813.307	-700.929
C(Type)[T.t]	-0.1236	0.009	-13.337	0.000	-0.142	-0.105
C(Type)[T.u]	-0.4535	0.008	-55.566	0.000	-0.469	-0.437
C(Method)[T.S]	0.0804	0.007	11.486	0.000	0.067	0.094
C(Method)[T.SA]	0.0347	0.046	0.758	0.448	-0.055	0.124
C(Method)[T.SP]	0.0416	0.009	4.774	0.000	0.025	0.059

C(Method)[T.VB]	-0.0082	0.010	-0.847	0.397	-0.027	0.011
C(Regionname)[T.Eastern Victoria]	-0.0857	0.039	-2.184	0.029	-0.163	-0.009
C(Regionname)[T.Northern Metropolitan]	-0.0570	0.011	-5.312	0.000	-0.078	-0.036
C(Regionname)[T.Northern Victoria]	0.3565	0.056	6.315	0.000	0.246	0.467
C(Regionname)[T.South-Eastern Metropolitan]	-0.0539	0.017	-3.200	0.001	-0.087	-0.021
C(Regionname)[T.Southern Metropolitan]	0.1220	0.010	12.806	0.000	0.103	0.141
C(Regionname)[T.Western Metropolitan]	-0.0319	0.014	-2.277	0.023	-0.059	-0.004
C(Regionname)[T.Western Victoria]	0.3701	0.051	7.245	0.000	0.270	0.470
Distance	-0.0376	0.001	-68.653	0.000	-0.039	-0.037
Bedroom2	0.0863	0.004	20.970	0.000	0.078	0.094
Bathroom	0.0600	0.005	12.486	0.000	0.051	0.069
Car	0.0268	0.003	9.911	0.000	0.022	0.032
Landsize	4.115e-05	5.67e-06	7.255	0.000	3e-05	5.23e-05
BuildingArea	0.0024	5.65e-05	42.142	0.000	0.002	0.002
YearBuilt	-0.0025	8.22e-05	-29.891	0.000	-0.003	-0.002
Lattitude	-0.9354	0.042	-22.084	0.000	-1.018	-0.852
log_Longtitude	148.6908	5.736	25.922	0.000	137.447	159.935

Omnibus:	31.092	Durbin-Watson:	1.724
Prob(Omnibus):	0.000	Jarque-Bera (JB):	23.991
Skew:	0.031	Prob(JB):	6.17e-06

```
smf.ols(full_model,data=df_without_infl_pt).fit().summary()
```

OLS Regression Results

Dep. Variable:	log_price	**R-squared:**	0.838
Model:	OLS	**Adj. R-squared:**	0.838
Method:	Least Squares	**F-statistic:**	1808.
Date:	Mon, 30 Nov 2020	**Prob (F-statistic):**	0.00
Time:	08:21:35	**Log-Likelihood:**	1667.0
No. Observations:	8046	**AIC:**	-3286.
Df Residuals:	8022	**BIC:**	-3118.
Df Model:	23		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-756.8750	28.668	-26.401	0.000	-813.072	-700.678
C(Type)[T.t]	-0.1238	0.009	-13.348	0.000	-0.142	-0.106
C(Type)[T.u]	-0.4533	0.008	-55.506	0.000	-0.469	-0.437
C(Method)[T.S]	0.0803	0.007	11.471	0.000	0.067	0.094
C(Method)[T.SA]	0.0340	0.046	0.743	0.458	-0.056	0.124
C(Method)[T.SP]	0.0415	0.009	4.753	0.000	0.024	0.059
C(Method)[T.VB]	-0.0083	0.010	-0.855	0.393	-0.027	0.011

C(Regionname)[T.Eastern Victoria]	-0.0859	0.039	-2.189	0.029	-0.163	-0.009
C(Regionname)[T.Northern Metropolitan]	-0.0560	0.011	-5.158	0.000	-0.077	-0.035
C(Regionname)[T.Northern Victoria]	0.3565	0.056	6.315	0.000	0.246	0.467
C(Regionname)[T.South-Eastern Metropolitan]	-0.0542	0.017	-3.215	0.001	-0.087	-0.021
C(Regionname)[T.Southern Metropolitan]	0.1224	0.010	12.817	0.000	0.104	0.141
C(Regionname)[T.Western Metropolitan]	-0.0321	0.014	-2.288	0.022	-0.060	-0.005
C(Regionname)[T.Western Victoria]	0.3696	0.051	7.232	0.000	0.269	0.470
Distance	-0.0376	0.001	-68.583	0.000	-0.039	-0.036
Bedroom2	0.0862	0.004	20.965	0.000	0.078	0.094
Bathroom	0.0600	0.005	12.487	0.000	0.051	0.069
Car	0.0269	0.003	9.913	0.000	0.022	0.032
Landsize	4.109e-05	5.67e-06	7.242	0.000	3e-05	5.22e-05
BuildingArea	0.0024	5.65e-05	42.143	0.000	0.002	0.002
YearBuilt	-0.0025	8.22e-05	-29.879	0.000	-0.003	-0.002
Lattitude	-0.9379	0.043	-22.033	0.000	-1.021	-0.854
log_Longtitude	148.6262	5.737	25.905	0.000	137.379	159.873
log_Propertycount	-0.0021	0.004	-0.597	0.551	-0.009	0.005

Omnibus:	31.565	Durbin-Watson:	1.725
Prob(Omnibus):	0.000	Jarque-Bera (JB):	24.307
Skew:	0.032	Prob(JB):	5.27e-06

In the full model, t-test for log_Propertycount shows p_value = 0.551 > 0.05, so it is insignificant.
After dropping 'log_Propertycount', R-squared stays unchanged, but AIC and BIC all become smaller.

Model with influential points

```
for i in range(1,14):
    print(f'aic for model12_{i} is: ',smf.ols(eval(f'model12_{i}'),data=origin_df).fit().aic)
```

```
aic for model12_1 is:  3627.7893581086973
aic for model12_2 is:  1520.0139951300262
aic for model12_3 is:  4096.831706809302
aic for model12_4 is:  2009.8991693316966
aic for model12_5 is:  1820.7231339377904
aic for model12_6 is:  1529.973790942433
aic for model12_7 is:  1496.630416702781
aic for model12_8 is:  2066.1623471254497
aic for model12_9 is:  1961.5215050661718
aic for model12_10 is:  1615.7003615100148
aic for model12_11 is:  1659.4856456298512
aic for model12_12 is:  2498.001048814356
aic for model12_13 is:  1415.362074806606
```

Model12_13 has the smallest aic 1415.4
Choose model12_13: log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname)

```
# drop log_Propertycount
print(f'aic for full_model is: ',smf.ols(full_model,data=origin_df).fit().aic)
for i in range(1,13):
    print(f'aic for model11_{i} is: ',smf.ols(eval(f'model11_{i}'),data=origin_df).fit().aic)
```

```
aic for full_model is:  1416.9373788934463
aic for model11_1 is:  3628.76153375626
aic for model11_2 is:  1518.2375752138141
aic for model11_3 is:  4105.334140359264
aic for model11_4 is:  2007.989342299068
aic for model11_5 is:  1819.3805173657638
aic for model11_6 is:  1528.487297104908
aic for model11_7 is:  1494.6769184659643
aic for model11_8 is:  2064.845146425756
aic for model11_9 is:  1960.0178879537416
aic for model11_10 is:  1616.8013302283107
aic for model11_11 is:  1657.4972184471662
aic for model11_12 is:  2496.0152246120015
```

Aic all become larger. Stop.

Finally pick model12_13.

log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname)

```
smf.ols(model12_13,data=origin_df).fit().summary()
```

OLS Regression Results

Dep. Variable:	log_price	R-squared:	0.761
Model:	OLS	Adj. R-squared:	0.761
Method:	Least Squares	F-statistic:	1284.
Date:	Mon, 30 Nov 2020	Prob (F-statistic):	0.00
Time:	08:21:42	Log-Likelihood:	-684.68
No. Observations:	8887	AIC:	1415.
Df Residuals:	8864	BIC:	1578.
Df Model:	22		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-529.8510	33.148	-15.984	0.000	-594.829	-464.873
C(Type)[T.t]	-0.1484	0.012	-12.877	0.000	-0.171	-0.126
C(Type)[T.u]	-0.4989	0.010	-49.906	0.000	-0.519	-0.479
C(Method)[T.S]	0.0799	0.009	9.115	0.000	0.063	0.097
C(Method)[T.SA]	0.0408	0.034	1.198	0.231	-0.026	0.108
C(Method)[T.SP]	0.0478	0.011	4.361	0.000	0.026	0.069
C(Method)[T.VB]	0.0174	0.012	1.444	0.149	-0.006	0.041

C(Regionname)[T.Eastern Victoria]	-0.0503	0.042	-1.203	0.229	-0.132	0.032
C(Regionname)[T.Northern Metropolitan]	-0.1174	0.013	-8.895	0.000	-0.143	-0.092
C(Regionname)[T.Northern Victoria]	0.2924	0.039	7.472	0.000	0.216	0.369
C(Regionname)[T.South-Eastern Metropolitan]	0.0047	0.021	0.226	0.821	-0.036	0.045
C(Regionname)[T.Southern Metropolitan]	0.1446	0.012	12.199	0.000	0.121	0.168
C(Regionname)[T.Western Metropolitan]	-0.1062	0.017	-6.240	0.000	-0.140	-0.073
C(Regionname)[T.Western Victoria]	0.1616	0.050	3.259	0.001	0.064	0.259
Distance	-0.0354	0.001	-56.000	0.000	-0.037	-0.034
Bedroom2	0.1151	0.005	24.767	0.000	0.106	0.124
Bathroom	0.1111	0.005	20.356	0.000	0.100	0.122
Car	0.0346	0.003	10.751	0.000	0.028	0.041
Landsize	2.438e-05	2.7e-06	9.026	0.000	1.91e-05	2.97e-05
BuildingArea	0.0011	4.19e-05	25.966	0.000	0.001	0.001
YearBuilt	-0.0023	9.57e-05	-23.714	0.000	-0.002	-0.002
Lattitude	-0.7293	0.051	-14.327	0.000	-0.829	-0.630
log_Longtitude	104.5220	6.652	15.712	0.000	91.482	117.562

Omnibus:	1120.669	Durbin-Watson:	1.677
Prob(Omnibus):	0.000	Jarque-Bera (JB):	11723.859
Skew:	-0.187	Prob(JB):	0.00
Kurtosis:	8.614	Cond. No.	2.51e+07

Comparison: model selection results for data with and without influential points are the same. The Stepwise result also agrees with the t-test.

But the final R_square is different. R_square = 0.838 for data without influential points. R_square = 0.761 for data with influential points, suggesting a better fit when influential points are removed. The AIC = -3288 for models without influential points is much lower than the AIC = 1415 for models with influential points.

For the same model selection result, data fits the model well after dropping influential points.

```
smf.ols(full_model,data=origin_df).fit().summary()
```

OLS Regression Results

Dep. Variable:	log_price	**R-squared:**	0.761
Model:	OLS	**Adj. R-squared:**	0.761
Method:	Least Squares	**F-statistic:**	1228.
Date:	Mon, 30 Nov 2020	**Prob (F-statistic):**	0.00
Time:	08:21:43	**Log-Likelihood:**	-684.47
No. Observations:	8887	**AIC:**	1417.
Df Residuals:	8863	**BIC:**	1587.
Df Model:	23		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-530.5198	33.165	-15.996	0.000	-595.531	-465.509
C(Type)[T.t]	-0.1482	0.012	-12.864	0.000	-0.171	-0.126
C(Type)[T.u]	-0.4992	0.010	-49.877	0.000	-0.519	-0.480
C(Method)[T.S]	0.0800	0.009	9.129	0.000	0.063	0.097
C(Method)[T.SA]	0.0414	0.034	1.215	0.224	-0.025	0.108
C(Method)[T.SP]	0.0480	0.011	4.380	0.000	0.027	0.069
C(Method)[T.VB]	0.0175	0.012	1.455	0.146	-0.006	0.041

C(Regionname)[T.Eastern Victoria]	-0.0503	0.042	-1.203	0.229	-0.132	0.032
C(Regionname)[T.Northern Metropolitan]	-0.1187	0.013	-8.894	0.000	-0.145	-0.093
C(Regionname)[T.Northern Victoria]	0.2932	0.039	7.488	0.000	0.216	0.370
C(Regionname)[T.South-Eastern Metropolitan]	0.0051	0.021	0.246	0.806	-0.035	0.046
C(Regionname)[T.Southern Metropolitan]	0.1440	0.012	12.114	0.000	0.121	0.167
C(Regionname)[T.Western Metropolitan]	-0.1059	0.017	-6.224	0.000	-0.139	-0.073
C(Regionname)[T.Western Victoria]	0.1629	0.050	3.283	0.001	0.066	0.260
Distance	-0.0354	0.001	-55.876	0.000	-0.037	-0.034
Bedroom2	0.1151	0.005	24.772	0.000	0.106	0.124
Bathroom	0.1111	0.005	20.349	0.000	0.100	0.122
Car	0.0346	0.003	10.746	0.000	0.028	0.041
Landsize	2.446e-05	2.7e-06	9.047	0.000	1.92e-05	2.98e-05
BuildingArea	0.0011	4.19e-05	25.959	0.000	0.001	0.001
YearBuilt	-0.0023	9.57e-05	-23.711	0.000	-0.002	-0.002
Lattitude	-0.7267	0.051	-14.230	0.000	-0.827	-0.627
log_Longtitude	104.6713	6.656	15.725	0.000	91.623	117.719
log_Propertycount	0.0029	0.004	0.651	0.515	-0.006	0.012

Omnibus:	1120.433	Durbin-Watson:	1.677
Prob(Omnibus):	0.000	Jarque-Bera (JB):	11712.746
Skew:	-0.187	Prob(JB):	0.00

Both the full model and final model has an Rsquared=0.761. Dropping log_Propertycount based on model selection does not affect our R-squared.

Model Prediction

True price vs fitted price

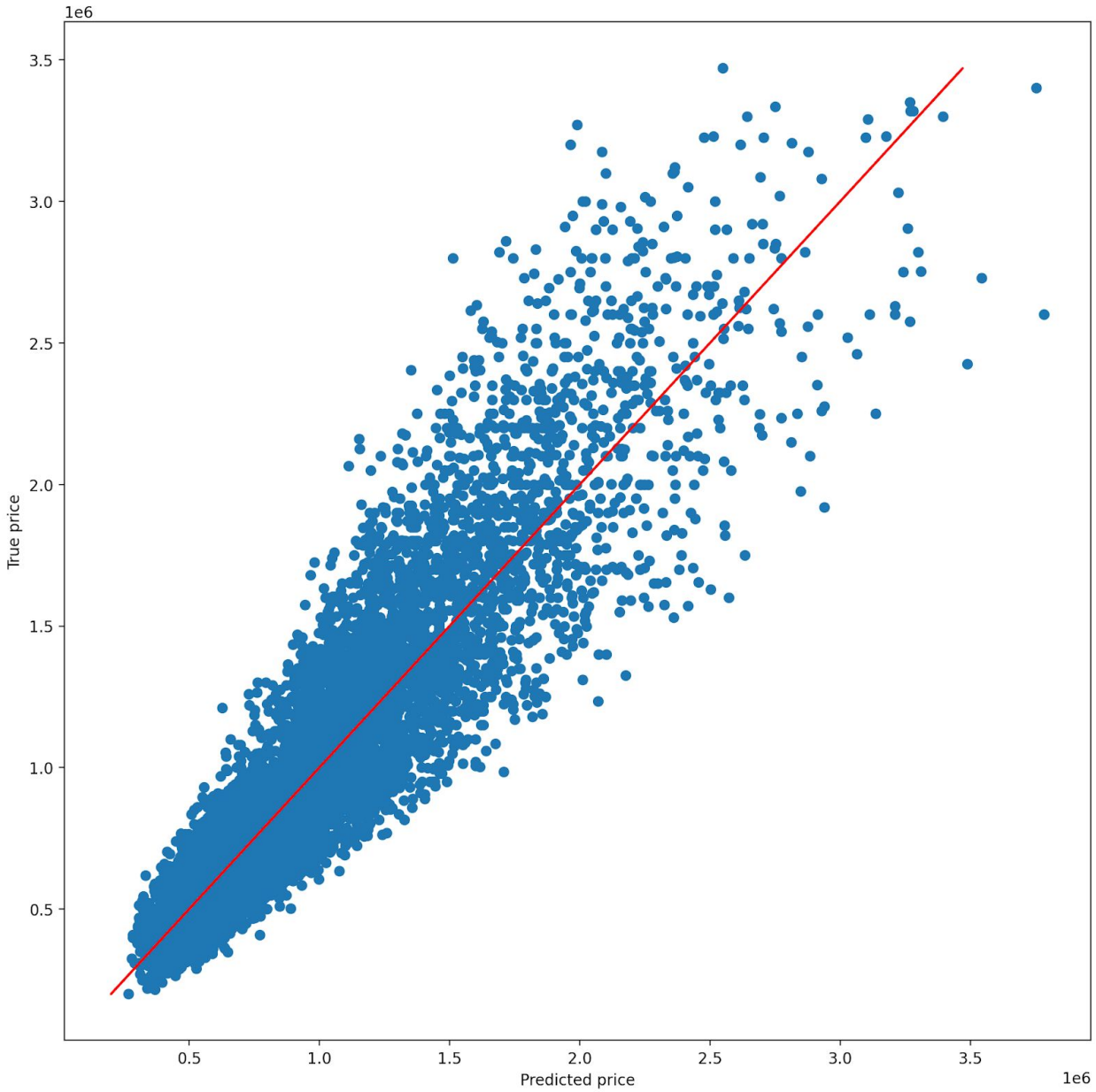


Figure: True vs Predicted price where the red line indicates a perfect fit. The higher the price, the worse the prediction for our final model. For lower prices, our final model has better predictions as it is closer to the red line.

The model explained the variance of y well as our R squared >0.8.

Final Model and Interpretation

Final fitted model:

log_price = -757.1181 - 0.1236*Type_t - 0.4535*Type_u + 0.0804*Method_S + 0.0347*Method_SA + 0.0416*Method_SP - 0.0082*Method_VB -0.0376*Distance+ 0.0863*Bedroom2+ 0.06*Bathroom+ 0.0268*Car + 4.115e-05*Landsize + 0.0024*BuildingArea -0.0025*YearBuilt -0.9354*Lattitude + 148.6908*log_Longtitude -0.0857*Region_Eastern_Victoria - 0.0570*Region_Northern_Metropolitan + 0.3565*Region_Northern_Victoria - 0.0539*Region_South-Eastern_Metropolitan + 0.1220*Region_Southern_Metropolitan - 0.0319*Region_Western_Metropolitan + 0.3701*Region_Western_Victoria

```
fit_model.summary()
```

OLS Regression Results

Dep. Variable:	log_price	R-squared:	0.838
Model:	OLS	Adj. R-squared:	0.838
Method:	Least Squares	F-statistic:	1890.
Date:	Wed, 02 Dec 2020	Prob (F-statistic):	0.00
Time:	04:23:22	Log-Likelihood:	1666.8
No. Observations:	8046	AIC:	-3288.
Df Residuals:	8023	BIC:	-3127.
Df Model:	22		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-757.1181	28.664	-26.414	0.000	-813.307	-700.929
C(Type)[T.t]	-0.1236	0.009	-13.337	0.000	-0.142	-0.105
C(Type)[T.u]	-0.4535	0.008	-55.566	0.000	-0.469	-0.437
C(Method)[T.S]	0.0804	0.007	11.486	0.000	0.067	0.094
C(Method)[T.SA]	0.0347	0.046	0.758	0.448	-0.055	0.124
C(Method)[T.SP]	0.0416	0.009	4.774	0.000	0.025	0.059
C(Method)[T.VB]	-0.0082	0.010	-0.847	0.397	-0.027	0.011
C(Regionname)[T.Eastern Victoria]	-0.0857	0.039	-2.184	0.029	-0.163	-0.009
C(Regionname)[T.Northern Metropolitan]	-0.0570	0.011	-5.312	0.000	-0.078	-0.036

C(Regionname)[T.Northern Victoria]	0.3565	0.056	6.315	0.000	0.246	0.467
C(Regionname)[T.South-Eastern Metropolitan]	-0.0539	0.017	-3.200	0.001	-0.087	-0.021
C(Regionname)[T.Southern Metropolitan]	0.1220	0.010	12.806	0.000	0.103	0.141
C(Regionname)[T.Western Metropolitan]	-0.0319	0.014	-2.277	0.023	-0.059	-0.004
C(Regionname)[T.Western Victoria]	0.3701	0.051	7.245	0.000	0.270	0.470
Distance	-0.0376	0.001	-68.653	0.000	-0.039	-0.037
Bedroom2	0.0863	0.004	20.970	0.000	0.078	0.094
Bathroom	0.0600	0.005	12.486	0.000	0.051	0.069
Car	0.0268	0.003	9.911	0.000	0.022	0.032
Landsize	4.115e-05	5.67e-06	7.255	0.000	3e-05	5.23e-05
BuildingArea	0.0024	5.65e-05	42.142	0.000	0.002	0.002
YearBuilt	-0.0025	8.22e-05	-29.891	0.000	-0.003	-0.002
Lattitude	-0.9354	0.042	-22.084	0.000	-1.018	-0.852
log_Longtitude	148.6908	5.736	25.922	0.000	137.447	159.935

Omnibus:	31.092	Durbin-Watson:	1.724
Prob(Omnibus):	0.000	Jarque-Bera (JB):	23.991
Skew:	0.031	Prob(JB):	6.17e-06
Kurtosis:	2.740	Cond. No.	2.70e+07

Interpretation:

Categorical Variables

Type: 'h' ,'u' ,'t' (choose 'h' as the reference level)
Among all types (house, unit, townhouse), houses are the most expensive and units are the least expensive.
Given all other predictors are the same, the average price will increase 13.16% if we choose houses instead of townhouses, and the price will increase 57.38% if we buy houses instead of uints.

Method: 'S', 'SP', 'VB', 'PI', 'SA' (choose 'PI' as the reference level)
Among all selling methods, houses from property sold are the most expensive. Houses sold by vendor bid tend to have the lowest price.
The average house price sold by property sold will increase 9.26% compared with vender bid.

Regionname:'Northern Metropolitan', 'Western Metropolitan', 'Southern Metropolitan', 'Eastern Metropolitan', 'South-Eastern Metropolitan', 'Northern Victoria', 'Eastern Victoria', 'Western Victoria' (choose 'Eastern Metropolitan' as the reference level)
Houses in Western Victoria Region are the most expensive.
Houses in Eastern Victoria Region are the least expensive.

The house price in Western Victoria Region is on average 57.74% higher than houses in Eastern Victoria Region.

Other Interpretations

The intercept is -757.12, it means given all predictors are 0, the price should be $0 (np.exp(-757.1181)).

Price is directly proportional to the number of Bedrooms and Bathrooms. Houses with more bedrooms and bathrooms are more expensive. If the house has one more bedroom, the price will on average increase 9%. If the house has one more bathroom, the price will increase 6.2% on average.

Houses that are close to CBD tend to have higher price.If the house is 10 km closer to CBD, price will on average increase 45.64%.

Houses with more carspots are more expensive. If the house has one more carspot, the price will increase 2.72% on average.

Houses with larger building areas and landsize are more expensive. If the buildingsize is 10 square meter larger, price will on average increase 2.43%.

New houses tend to be less expensive. If the house's age is ten year older (Yearbuilt), price will on average increase 2.53%.

House price is in direct proportion to longitude and is inversely proportional to latitude. For every 1% increase in the Longtitude, price increases by about 339.08%. If the latitude increases 0.1, the average price will decrease 9.81%.

Influential point analysis

Some influential points are houses with extremely large Landsize, BuildingArea, and too many bedrooms, bathrooms. After inspecting the data, they may just be observations with unusual values. (Most recording mistakes have been dropped at the beginning.)

```
[130]
infl_data[['Bedroom2','Bathroom','Landsize','BuildingArea']].describe()
```

Preview | Raw output | Visualize

	Bedroom2 float64	Bathroom float64	Landsize float64	BuildingArea float64
count	841	841	841	841
mean	3.4340071343638527	2.027348394768133	1121.3674197384066	213.0800713436385
std	1.3132055454920015	1.0421880525173564	3135.444398036004	190.35753769206352
min	0	1	0	0
25%	3	1	365	106
50%	3	2	616	167
75%	4	2	808	280
max	12	9	42800	3112

Among influential points there's a high proportion of Method 'SA'. So some of the influential points may belong to a different group.

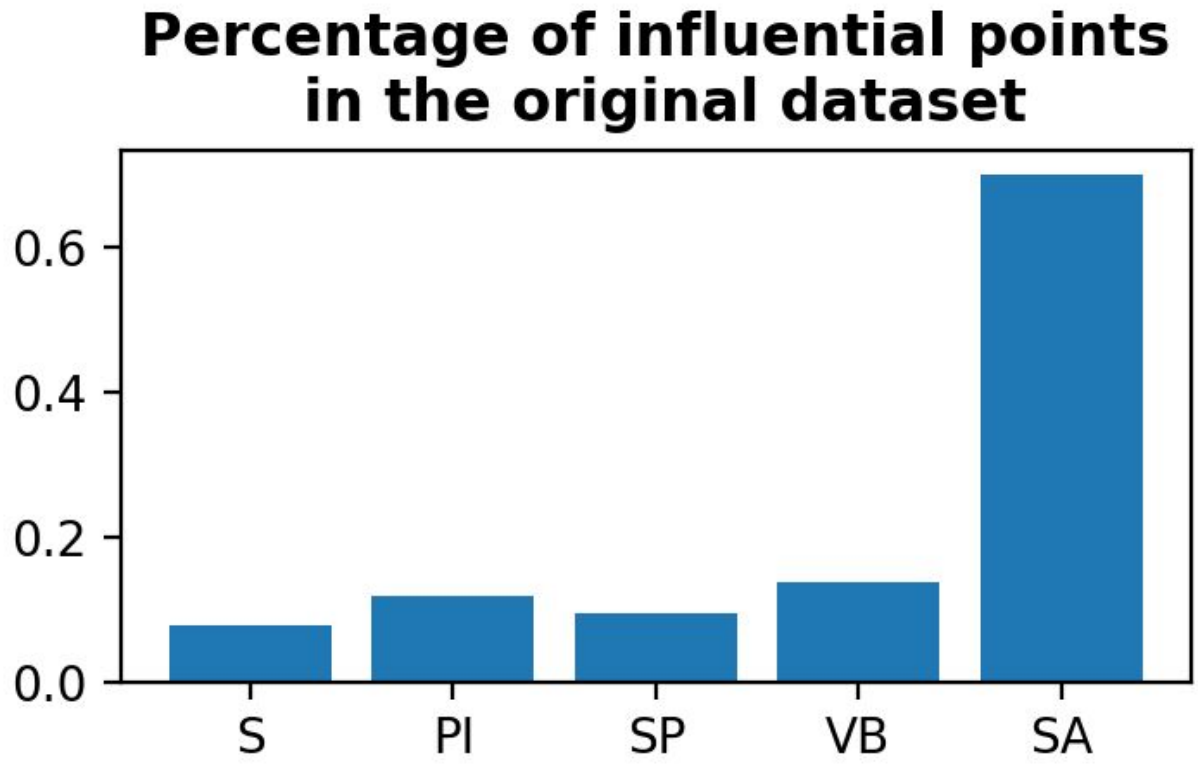


Figure: Percentage of influential points in the original dataset. We can see that a method of sold after auction (SA) has the highest distribution of influential points and is one source that we found of influential points.

Summary

Model Diagnostics

- We found multicollinearity between Rooms and Bedrooms2 so we dropped Rooms and multicollinearity was fixed.
- We performed model evaluation on initial model 1 with influential points (model1_0). Normality, homoscedasticity, linearity assumptions were all violated. Using the initial model 1 with influential points (model1_0), we used cook's distance and a threshold of 4/n to identify hundreds of influential points.
- We fit a new model, called model1_1 without influential points. Normality improved without influential points but heteroscedasticity did not improve.
- Because of the heteroscedasticity problem, we then log transformed the response variable price and it improved the heteroscedasticity from being a funnel shape to a more consistent band but the BP test still indicated significant heteroscedasticity. And then we fit model2_0 (with influential points and with transformation). We also fit model2_1 (without influential points and with transformation)
- Because of the nonlinearity problem, we log transformed Longtitude and PropertyCounts.After removing influential points and log transform price, longitude, propertycount, linearity is improved. linearity is improved for log_Longtitude and log_Propertycount. For other variables, we were unable to do a log transformation
- The Rsquared improved when removing influential points and also after transformation.

Model Selection

- For the model without influential points, we found out that the best model is log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname) with an AIC of -3287.5 and an R^2 of 83.8%
- For the model without influential points, we found out that the best model is log_price~C(Type)+C(Method)+ Distance+ Bedroom2+ Bathroom+ Car+Landsize+ BuildingArea+ YearBuilt+ Lattitude+log_Longtitude+ C(Regionname) with an AIC of 1415 and an R^2 of 0.761.
- During model selection using both direction stepwise (starting from full model), the same model was selected with and without influential points.
- Best selected model without influential points have lower AIC and better R^2 than the best selected model with influential points

Influential point analysis

- May come from extremely large Landsize, BuildingArea, and too many bedrooms, bathrooms
- May come from houses sold after auction(SA) because nearly 70% of the SA samples are determined as influential points

Table of contribution

	Christopher	Yueling	Tianxiang
Proportion of work	33%	33%	33%
List of work	Fit initial models and model diagnostics. Final discussion. Wrote report	Initial check of data. Model selection. Reviewed initial model/diagnostics. Analyzed source of influential points. Final discussion. Wrote report	Found data set. Added EDA visualizations. Final discussion. Wrote report