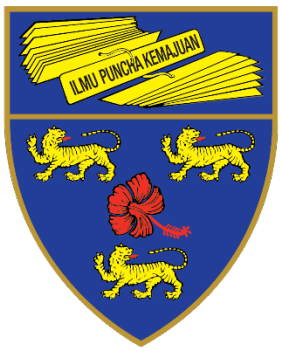


UNIVERSITI MALAYA

Master of Data Science (Semester 1 – 2022/2023)

Faculty of Computer Science & Information Technology

WQD7005 Data Mining

Group Project

Project Title: House Price Prediction

Instructor: Prof Dr Teh Ying Wah

Name	**Matric Number**
Jasmeen Kah Ying Bong	S2142739
Hii Yew Han	S2037987
Dinesh V M Ramachandran	S2119167
Aw Yeong Fung Mun	17197465
Lee Ziteng Nicholas	S2132376

Table of Contents

1 Introduction

The fluctuation of house prices is a contentious issue because it can have a significant impact on the economy as a whole. An increase in house prices means an increase in non-financial assets, which in turn increases personal wealth and stimulates household consumption, boosting the economy. However, a decrease in house prices limits an individual's borrowing capacity, causing investments to decline due to the decrease in the value of collateral. The 2008 housing bubble is a perfect example of the importance of stable, predictable house prices. Unexpected changes in house prices can lead to an increase in real long-term interest rates, bankruptcies in financial institutions, and global economic depression. While it may be difficult to control house prices, it is possible to predict them.

Therefore, in this report, we used a dataset from Kaggle covering house sales from May 2014 through May 2015 at King County, which is in the U.S. state of Washington. According to the data gathered by the 2020 United States census, King County has the highest estimated population of 2,269,675 among all counties in Washington and is also the 13th-most populous county in the United States. With a higher population, King County has more potential house buyers and higher house demands; thus, house price data in King County will be more complete and more precise. In this dataset, it includes categorical data and numerical data, we can perform multi-faceted analysis and prediction on the dataset.

2 Our Dataset

2.1 Table Information

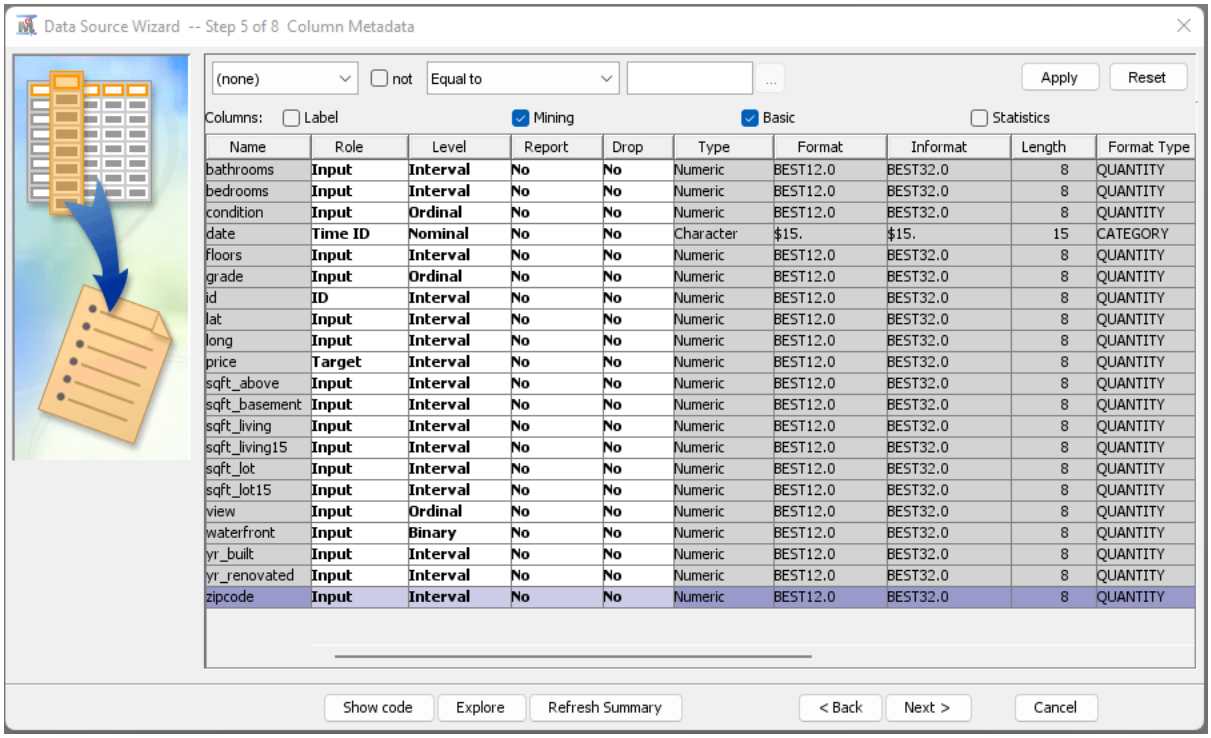
<table>
<tr><td>Data Source</td><td colspan="3">https://www.kaggle.com/datasets/harlfoxem/housesalesprediction</td></tr>
<tr><td>Data Name</td><td colspan="3">kc_house_data.csv</td></tr>
<tr><td>File Size</td><td colspan="3">2.52 MB</td></tr>
<tr><td>Year</td><td colspan="3">2016</td></tr>
<tr><td>Dimension</td><td colspan="3">21613 rows, 21 columns</td></tr>
<tr><td>Columns</td><td colspan="3">
<table>
<tr><td>Attributes</td><td>Types</td><td>Description</td></tr>
<tr><td>id</td><td>Numerical</td><td>A notation for a house</td></tr>
<tr><td>date</td><td>Numerical</td><td>Date house was sold</td></tr>
<tr><td>price</td><td>Numerical</td><td>Price of each home sold</td></tr>
<tr><td>bedrooms</td><td>Numerical</td><td>Number of bedrooms</td></tr>
<tr><td>bathrooms</td><td>Numerical</td><td>Number of bathrooms</td></tr>
<tr><td>sqft_living</td><td>Numerical</td><td>Square footage of the apartments interior living space</td></tr>
<tr><td>sqft_lot</td><td>Numerical</td><td>Square footage of the land space</td></tr>
<tr><td>floors</td><td>Numerical</td><td>Number of floors</td></tr>
<tr><td>waterfront</td><td>Categorical</td><td>Overlooking the waterfront or not:
● 1 = Yes
● 0 = No</td></tr>
<tr><td>view</td><td>Numerical</td><td>An index from 0 to 4 of how good the view of the property was</td></tr>
<tr><td>condition</td><td>Numerical</td><td>An index from 1 to 5 on the condition of the property</td></tr>
<tr><td>grade</td><td>Numerical</td><td>An index from 1 to 13 based on King County grading system</td></tr>
<tr><td>sqft_above</td><td>Numerical</td><td>The square footage of the interior housing space that is above ground level</td></tr>
<tr><td>sqft_basement</td><td>Numerical</td><td>The square footage of the interior housing space that is below ground level</td></tr>
</table>
</td></tr>
</table>

	yr_built	Numerical	Year when the house was initially built
	yr_renovated	Numerical	Year when house was renovated
	zipcode	Numerical	5-digit zip code area the house is in
	lat	Numerical	Latitude
	long	Numerical	Longitude
	sqft_living15	Numerical	The square footage of interior housing living space for the nearest 15 neighbours
	sqft_lot15	Numerical	The square footage of the land lots of the nearest 15 neighbours

2.2 Column Metadata

This is our metadata; we have a total of 21 variables. They are divided into two types of data, namely numeric data, and character data. The following is a detailed display of the metadata:

Column	Model Role	Measurement Level	Description
bathrooms	Input	Interval	Number of bathrooms
bedrooms	Input	Interval	Number of bedrooms
condition	Input	Ordinal	Condition of the property
date	Time ID	Nominal	Date house was sold
floors	Input	Interval	Number of floors
grade	Input	Ordinal	Grade level of the property
id	ID	Interval	A notation for a house
lat	Input	Interval	Latitude
long	Input	Interval	Longitude
price	Target	Interval	Price of each home sold
sqft_above	Input	Interval	The square footage of the interior housing space that is above ground level
sqft_basement	Input	Interval	The square footage of the interior housing space that is below ground level
sqft_living	Input	Interval	Square footage of the apartments interior living space
sqft_living15	Input	Interval	The average square footage of interior housing living space for the nearest 15 neighbours
sqft_lot	Input	Interval	Square footage of the land space
sqft_lot15	Input	Interval	The average square footage of the land lots of the nearest 15 neighbours
view	Input	Ordinal	View index of the property
waterfront	Input	Binary	House which has a view to a waterfront
yr_built	Input	Interval	Year when the house was initially built
yr_renovated	Input	Interval	Year when house was renovated
zipcode	Input	Interval	5-digit zip code area the house is in



Name	Role	Level	Report	Drop	Type	Format	Informat	Length	Format Type
bathrooms	Input	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
bedrooms	Input	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
condition	Input	Ordinal	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
date	Time ID	Nominal	No	No	Character	$15.	$15.	15	CATEGORY
floors	Input	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
grade	Input	Ordinal	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
id	ID	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
lat	Input	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
long	Input	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
price	Target	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
sqft_above	Input	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
sqft_basement	Input	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
sqft_living	Input	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
sqft_living15	Input	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
sqft_lot	Input	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
sqft_lot15	Input	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
view	Input	Ordinal	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
waterfront	Input	Binary	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
yr_built	Input	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
yr_renovated	Input	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY
zipcode	Input	Interval	No	No	Numeric	BEST12.0	BEST32.0	8	QUANTITY

Figure 2.1 Column Metadata

3 Business Understanding

3.1 Analysis Goal

We wish to predict house prices. Therefore, our objectives are:

1. To find features that are relevant in predicting house prices.
2. To build a predictive model for house prices.

3.2 Analysis Data

This dataset contains house sales sold between May 2014 and May 2015 for King County, which includes Seattle. It contains essential attributes of a house including but not limited to bedrooms, bathrooms, and square foot size. This dataset is available publicly on Kaggle. The target variable is the price of the house sold.

4 Methodology

The methodology we adopted for this report is SEMMA, which stands for Sampling, Exploring, Modifying, Modelling, and Assessing. Developed by the SAS Institute as a process for data mining large amounts of data to uncover previously unknown patterns which can be utilized as a business advantage. In this report, the first two items are mainly discussed, namely Sample and Explore.

4.1 SEMMA Process

Figure 4.1 SEMMA Process

4.2 SEMMA Description

Sample

- In this step, a subset of the relevant volume dataset is selected from a sizable dataset provided for the model's building. Identification of variables or factors (both dependent and independent) impacting the process is the aim of the first stage of the process. The gathered data is then divided into categories for preparation and validation.

Explore

- Univariate, bivariate and multivariate analysis are carried out in this step, in order to investigate interrelated relationships between variables and to find data gaps. The univariate analysis examines each factor separately to comprehend its role in the broader scheme, whereas the bivariate and multivariate analysis explores the link between variables. With a focus on data visualisation, all the influencing factors that might have an impact on the study's conclusion are examined.

Modify

- In this step, business logic is used to derive the lessons discovered during the exploration phase from the data gathered during the sample phase. In other words, the data is analysed to determine whether it needs to be refined or transformed before moving on to the modelling stage.

Model

- After the variables have been clarified and the data have been cleaned, the modelling step employs a variety of data mining techniques to create a projected model of how this data achieves the process's final, desired result.

Assess

- In the SEMMA process' last stage, the model's applicability and dependability to the subject under study are assessed. Now that the data has been tested, it can be used to determine how effectively it performs.

5 Results

5.1 SAMPLE – Accessing and Assaying Prepared Data

The dataset we acquired is a comma-separated values (CSV) file, in order to make use of it in SAS Enterprise Miner, we need a compatible SAS file, which is an .sas*7bdat* file. This is done by running the File Import tool under the Sample Tab of the SEMMA tools palette and connecting it to the Save Data tool under the Utility Tab. Subsequently, running Save Data will produce an object file in a SAS compatible format that can be imported in the data source.

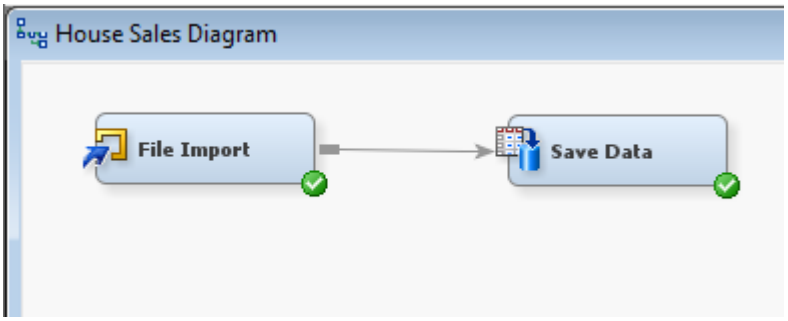


Figure 5.1 Convert Data Format

The dataset we have chosen is the data on house sales in King County, USA. In the following data, the variables related to the of the house are the ID, price of the house, number of bedrooms, number of bathrooms, square foot of living space, square foot of whole lot, number of floors, if there is a waterfront, view, condition, grade, square foot above and square foot basement. The variables related to time are dates of the house being sold, built year, and renovated year. Whereas the variables related to the location are zip code, latitude, and longitude of the house. In addition, additional information is provided such as, the average square footage of living space and whole lot for the nearest 15 neighbours.

In this house sales dataset, we have a total of 21 variables. They are divided into two types of data, namely numeric data, and character data. The following is a detailed display of the data:

Name	Role	Level	Report	Order	Drop	Lower	Upper	Type	Format	Informat	Length
bathrooms	**Input**	**Interval**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
bedrooms	**Input**	**Nominal**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
condition	**Input**	**Nominal**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
date	**Rejected**	**Nominal**	**No**		**No**	.	.	Character	$15.	$15.	15
floors	**Input**	**Nominal**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
grade	**Input**	**Nominal**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
id	**ID**	**Interval**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
lat	**Input**	**Interval**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
long	**Input**	**Interval**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
price	**Input**	**Interval**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
sqft_above	**Input**	**Interval**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
sqft_basement	**Input**	**Interval**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
sqft_living	**Input**	**Interval**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
sqft_living15	**Input**	**Interval**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
sqft_lot	**Input**	**Interval**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
sqft_lot15	**Input**	**Interval**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
view	**Input**	**Nominal**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
waterfront	**Input**	**Binary**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
yr_built	**Input**	**Interval**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
yr_renovated	**Input**	**Interval**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8
zipcode	**Rejected**	**Interval**	**No**		**No**	.	.	Numeric	BEST12.0	BEST32.0	8

Figure 5.2 Details of Data

After the data set is imported into SAS Enterprise Miner, we can see that the system will automatically set the measurement level for the metadata according to the attributes of the variables. By default, numeric data with only 1's and 0's is set to binary level, other numeric data is set to the interval level, and character data is set to the nominal level. In addition, there are some attributes which roles have been set rejected because SAS Enterprise Miner does not automatically recognize the format.

Metadata Completed.

Library: GROUP5A
Data Source: EM_SAVE_TRAIN
Role: Raw

Role	Level	Count
ID	Interval	1
Input	Binary	1
Input	Interval	12
Input	Nominal	5
Rejected	Interval	1
Rejected	Nominal	1

Figure 5.3 Result of Basic Setting

Our assessment shows that the default metadata set by SAS Enterprise Miner is not meeting our requirements. Therefore, we need to make manual changes to the roles and levels of the attributes. For levels, we have changed bedrooms, condition, floors, grade, and view to levels that reflect the data such as intervals and ordinals. For roles, we have changed date and zipcode from rejected to TimeID and Input respectively. Finally, price has been changed from input to target since it is a target variable. The following is a comparison of the changes that we have made:

Name	Role	Level
bathrooms	Input	Interval
bedrooms	Input	Nominal
condition	Input	Nominal
date	Rejected	Nominal
floors	Input	Nominal
grade	Input	Nominal
id	ID	Interval
lat	Input	Interval
long	Input	Interval
price	Input	Interval
sqft_above	Input	Interval
sqft_basement	Input	Interval
sqft_living	Input	Interval
sqft_living15	Input	Interval
sqft_lot	Input	Interval
sqft_lot15	Input	Interval
view	Input	Nominal
waterfront	Input	Binary
yr_built	Input	Interval
yr_renovated	Input	Interval
zipcode	Rejected	Interval

Name	Role	Level
bathrooms	Input	Interval
bedrooms	Input	Interval
condition	Input	Ordinal
date	TimeID	Nominal
floors	Input	Interval
grade	Input	Ordinal
id	ID	Interval
lat	Input	Interval
long	Input	Interval
price	Target	Interval
sqft_above	Input	Interval
sqft_basement	Input	Interval
sqft_living	Input	Interval
sqft_living15	Input	Interval
sqft_lot	Input	Interval
sqft_lot15	Input	Interval
view	Input	Ordinal
waterfront	Input	Binary
yr_built	Input	Interval
yr_renovated	Input	Interval
zipcode	Input	Interval

Figure 5.4 Setting Comparison

In these metadata, we have four measurement levels, they are interval, binary, ordinal and nominal attributes. The following is the updated content based on our understanding of the data sources, and it is also the sample result that we will use in the end:

Metadata Completed.

Library: GROUP5A
Data Source: EM_SAVE_TRAIN
Role: Raw

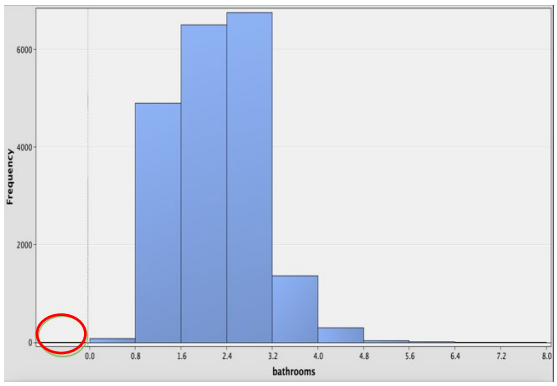
Role	Level	Count
ID	Interval	1
Input	Binary	1
Input	Interval	14
Input	Ordinal	3
Target	Interval	1
Time ID	Nominal	1

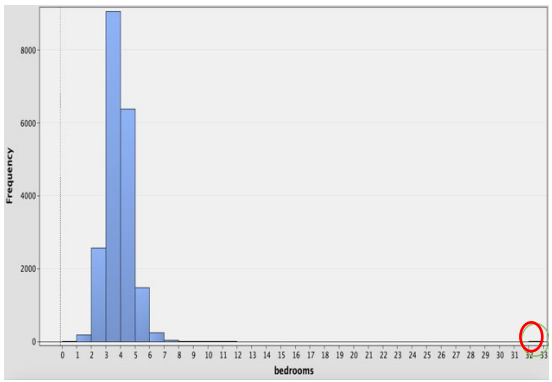
Figure 5.5 Result of Advanced Setting

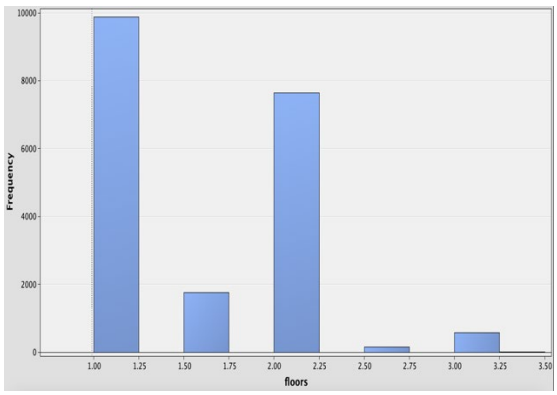
5.2 EXPLORE – Exploring Data Science

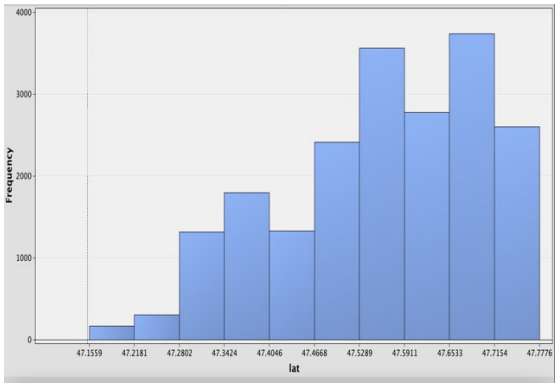
5.2.1 Univariate Analysis

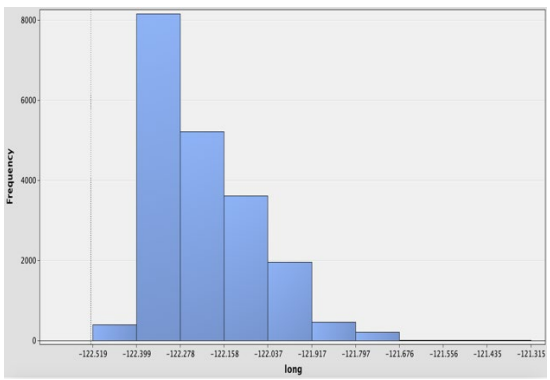
A descriptive analysis that uses just one variable is called a univariate analysis. This analysis's objectives are to provide us with the initial impression of each variable and to look for incomplete, noisy, and inconsistent data.

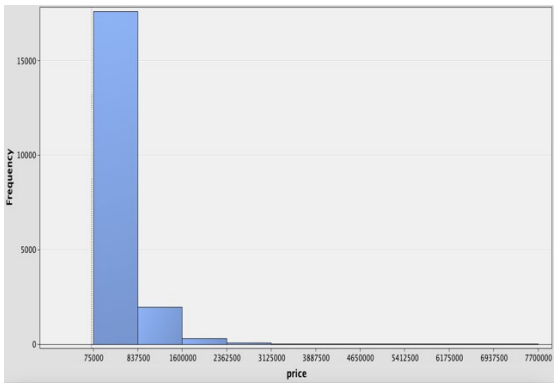


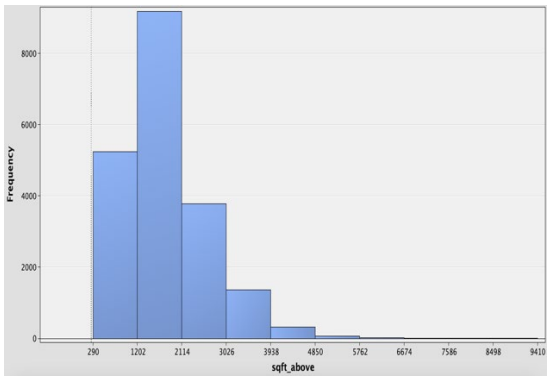


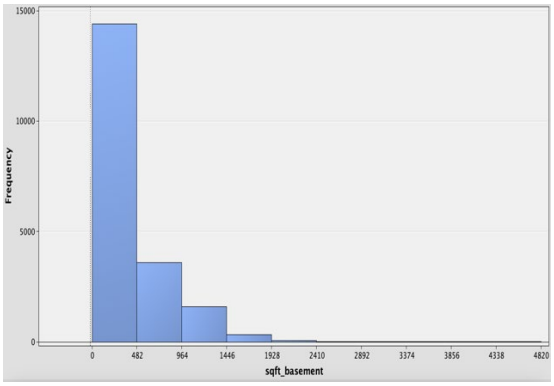


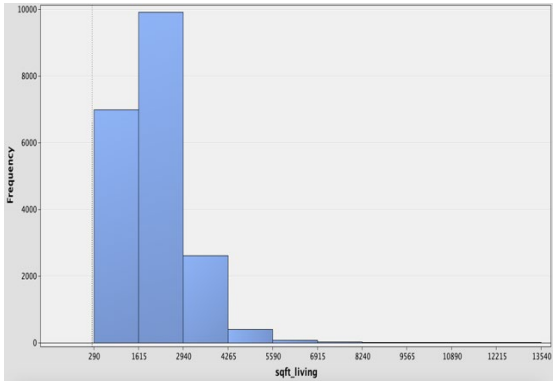


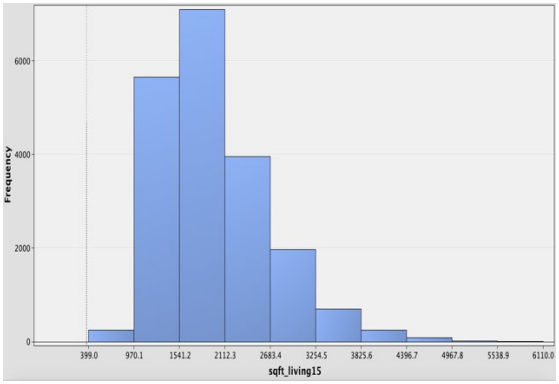


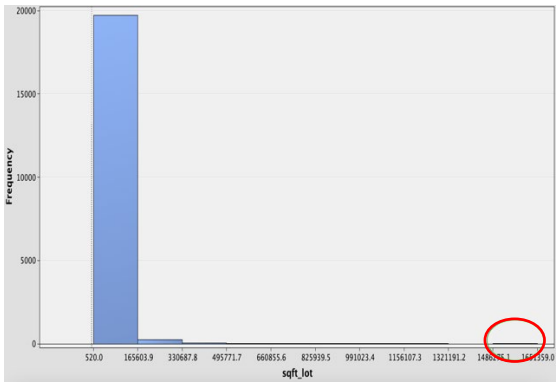


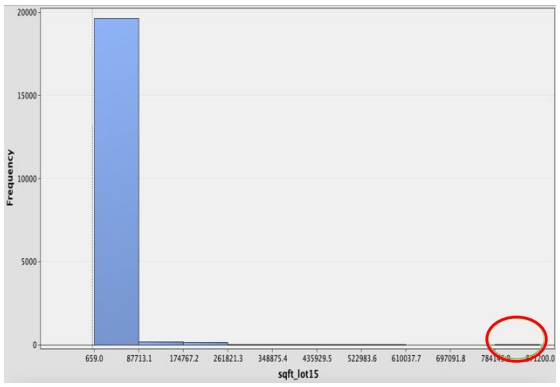

Frequency
sqft_above

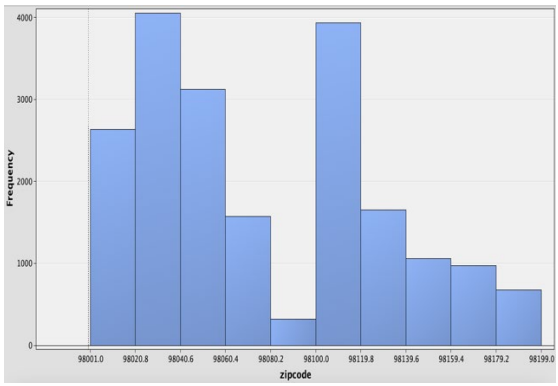

Frequency
sqft_basement

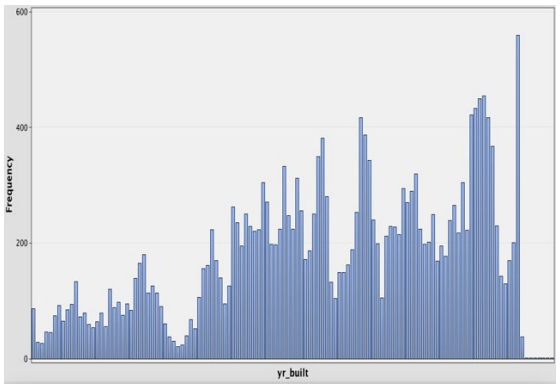

Frequency
sqft_living


Frequency
sqft_living15


Frequency
sqft_lot


Frequency
sqft_lot15


Frequency
zipcode


Frequency
yr_built

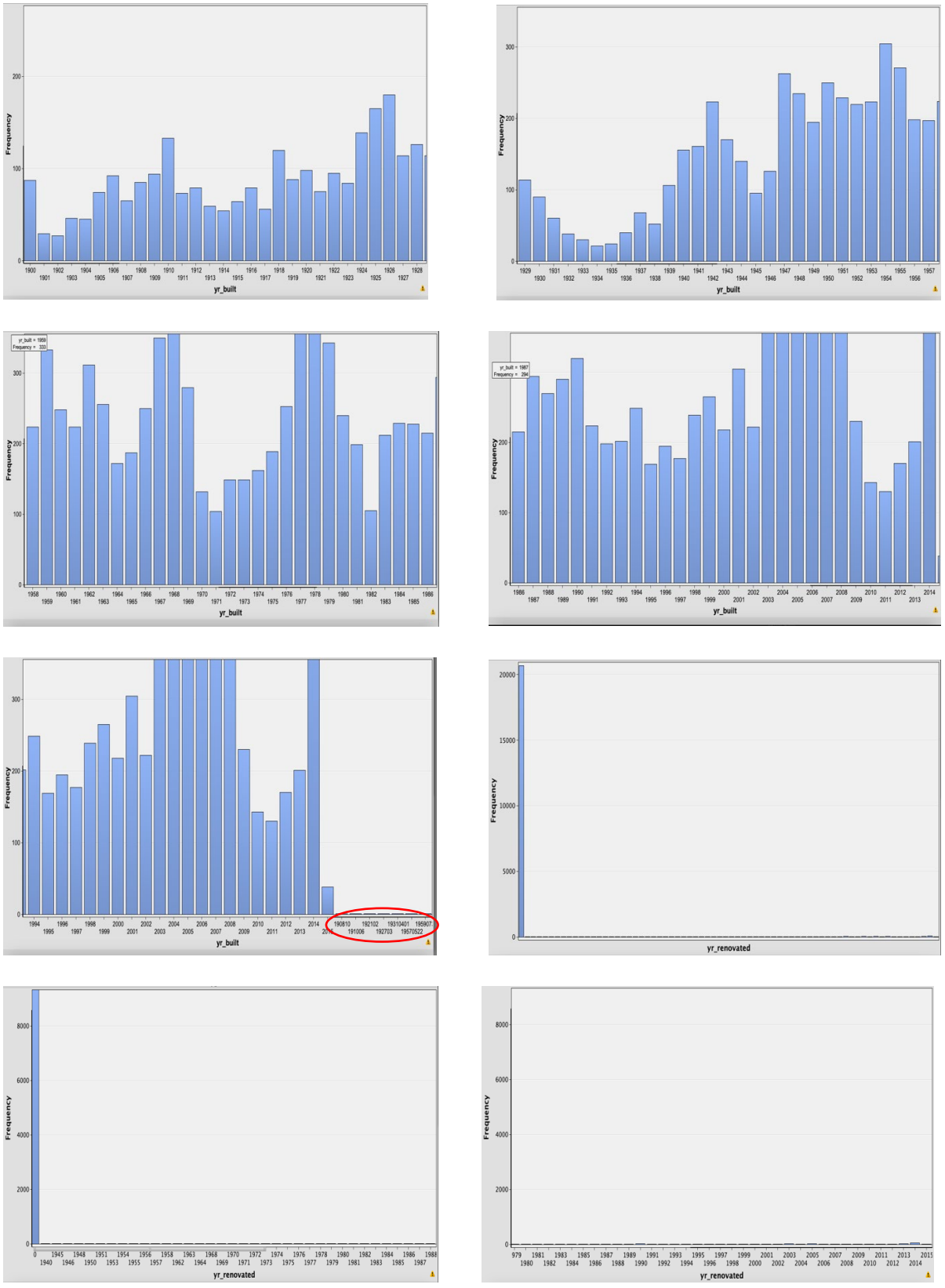


Figure 5.6 Histogram

We use histogram to show the distribution of interval variables and nominal variables. The above shows the histogram of different interval variable. We observed that the bathrooms variable has min value of 0 and max value of 8. The mode is the 4th bin ranges between 2.4 to 3.2 with around 6000 observations in this range. We found an error from the histogram. There is an incomplete data error in the bathrooms variable as highlighted in the red circle showing the missing bin in the bathrooms histogram. This may be due to failure recording the value due to human error. We observed that bedrooms variable has min value of 0 and max value of 33. The mode is the 4th bin ranges between 3 to 4 with around 8000 observations in this range. We found an error from the histogram. There is an outlier in the bedrooms variable as highlighted in the red circle. We observed that floors variable has min value of 1 to max value of 3.5. The mode is the 1st bin ranges between 1 to 1.25 with around 10000 observations. We observed that lat variable has min value of 47.1559 to max value of 47.7776. The mode is the 9th bin ranges between 47.6533 to 47.7154 with around 4000 observations. We observed that long variable has min value of -122.519 and max value of -121.315. The mode is the 2nd bin ranges between -122.399 to -122.278 with around 8000 observations. We observed that price variable has min value of 75000 and max value of 7700000. The mode is the 1st bin ranges between 75000 to 837500 with around 15000 observations. We observed that sqft_above variable has min value of 290 and max value of 9410. The mode is the 2nd bin ranges between 1202 to 2114 with around 8000 observations. We observed that sqft_basement variable has min value of 0 and max value of 4820. The mode is the 1st bin ranges between 0 to 482 with around 15000 observations. We observed that sqft_living variable has min value of 290 and max value of 13540. The mode is the 2nd bin ranges between 1615 to 2940 with around 10000 observations. We observed that sqft_living15 variable has min value of 399 and max value of 6210. The mode is 3rd bin ranges between 1541.2 to 2112.3 with around 6000 observations. We observed that sqft_lot has min value of 520 and max value of 1651359. The mode is 1st bin ranges between 520 to 165603.9 with around 20000 observations. We found an error from the histogram. There is an outlier in the sqft_lot variable as highlighted in the red circle. We observed that sqft_lot15 variable has min value of 651 and max value of 871200. The mode is 1st bin ranges between 659 to 87713.3 with around 20000 observations. We found an error from the histogram. There is an outlier in the sqft_lot15 variable as highlighted in the red circle. We observed that zipcode variable has min value of 98001 to max value of 98199. The mode is 2nd bin ranges between 98020.8 to 98040.6 with around 4000 observations. We observed that yr_built has a min value of 1900 and max value of 2015. The mode is 2014 which has the highest frequency of 559 with around 600 observations. We found an error from the histogram.

There is an inconsistency data error in the yr_built variable as highlighted in the red circle, there are eight values, which are 190810, 191006, 192102, 192703, 19310401, 19570522 and 19590731 which are supposed to be record in years but are recorded together with months and dates due to human error. We observed that yr_renovated has a min value of 0 and max value of 2015. The mode is 0 which has the highest frequency of 20699 with around 20000 observations.

Ordered Inputs	Data Role	Variable	Median	Missing	Non Missing	Minimum	Maximum	Mean	Standard Deviation	Skewness	Kurtosis	Role	Label	Abs C.V.	Coefficient of Variation	Sign
1	TRAIN	sqft_lot	7617	0	21613	520	1651359	15106.97	41420.51	13.06002	285.0778	INPUT	sqft_lot	2.741815	2.741815	+
2	TRAIN	sqft_lot15	7620	0	21613	651	871200	12768.46	27304.18	9.506743	150.7631	INPUT	sqft_lot15	2.138409	2.138409	+
3	TRAIN	sqft_basement	0	0	21613	0	4820	291.509	442.575	1.577965	2.715574	INPUT	sqft_bas...	1.518221	1.518221	+
4	TRAIN	price	450000	0	21613	75000	7700000	540182.2	367362.2	4.021716	34.52244	INPUT	price	0.680071	0.680071	+
5	TRAIN	sqft_above	1560	0	21613	290	9410	1788.391	828.091	1.446664	3.402304	INPUT	sqft_ab...	0.463037	0.463037	+
6	TRAIN	sqft_living	1910	0	21613	290	13540	2079.9	918.4409	1.471555	5.243093	INPUT	sqft_living	0.441579	0.441579	+
7	TRAIN	bathrooms	2.25	5	21608	0	8	2.114807	0.770164	0.511142	1.280413	INPUT	bathroo...	0.364177	0.364177	+
8	TRAIN	floors	1.5	0	21613	1	3.5	1.494309	0.539989	0.616177	-0.48472	INPUT	floors	0.361364	0.361364	+
9	TRAIN	sqft_living15	1840	0	21613	399	6210	1986.552	685.3913	1.108181	1.597096	INPUT	sqft_livi...	0.345015	0.345015	+
10	TRAIN	bedrooms	3	0	21613	0	33	3.370842	0.930062	1.9743	49.06365	INPUT	bedrooms	0.275914	0.275914	+
11	TRAIN	lat	47.5718	0	21613	47.1559	47.7776	47.56005	0.138564	-0.48527	-0.67663	INPUT	lat	0.002913	0.002913	+
12	TRAIN	long	-122.231	0	21613	-122.519	-121.315	-122.214	0.140828	0.885027	0.959911	INPUT	long	0.001152	-0.00115	+
13	TRAIN	zipcode	98065	0	21613	98001	98199	98077.94	53.50503	0.405497	-0.33233	INPUT	zipcode	.0005455	.0005455	+

Figure 5.7 Summary Statistics for Interval Variables

The above summary statistics for interval variables are shown to support the visualization. From the summary statistics, there is 5 missing values in bathrooms as highlighted in the red circle.

Data Role	Variable Name	Level	CODE	Frequency Count	Type	Percent	Level Index	Role	Label	Plot
TRAIN	yr_built	2014	34	559	N	2.586406	115	INPUT	yr_built	1
TRAIN	yr_built	2006	54	454	N	2.100588	107	INPUT	yr_built	1
TRAIN	yr_built	2005	26	450	N	2.08208	106	INPUT	yr_built	1
TRAIN	yr_built	2004	73	433	N	2.003424	105	INPUT	yr_built	1
TRAIN	yr_built	2003	9	422	N	1.952529	104	INPUT	yr_built	1
TRAIN	yr_built	1977	12	417	N	1.929394	78	INPUT	yr_built	1
TRAIN	yr_built	2007	75	417	N	1.929394	108	INPUT	yr_built	1
TRAIN	yr_built	1978	64	387	N	1.790589	79	INPUT	yr_built	1
TRAIN	yr_built	1968	20	381	N	1.762828	69	INPUT	yr_built	1
TRAIN	yr_built	2008	40	367	N	1.698052	109	INPUT	yr_built	1
TRAIN	yr_built	1967	59	350	N	1.619396	68	INPUT	yr_built	1
TRAIN	yr_built	1979	14	343	N	1.587008	80	INPUT	yr_built	1
TRAIN	yr_built	1959	36	333	N	1.540739	60	INPUT	yr_built	1
TRAIN	yr_built	1990	70	320	N	1.48059	91	INPUT	yr_built	1
TRAIN	yr_built	1962	56	312	N	1.443576	63	INPUT	yr_built	1
TRAIN	yr_built	1954	42	305	N	1.411188	55	INPUT	yr_built	1
TRAIN	yr_built	2001	5	305	N	1.411188	102	INPUT	yr_built	1
TRAIN	yr_built	1987	4	294	N	1.360292	88	INPUT	yr_built	1
TRAIN	yr_built	1989	45	290	N	1.341785	90	INPUT	yr_built	1
TRAIN	yr_built	1969	18	280	N	1.295517	70	INPUT	yr_built	1
TRAIN	yr_built	1955	0	271	N	1.253875	56	INPUT	yr_built	1
TRAIN	yr_built	1988	55	270	N	1.249248	89	INPUT	yr_built	1
TRAIN	yr_built	1999	78	265	N	1.226114	100	INPUT	yr_built	1
TRAIN	yr_built	1947	19	263	N	1.216866	48	INPUT	yr_built	1
TRAIN	yr_built	1963	7	256	N	1.184472	64	INPUT	yr_built	1
TRAIN	yr_built	1976	76	253	N	1.170592	77	INPUT	yr_built	1
TRAIN	yr_built	1950	39	250	N	1.156711	51	INPUT	yr_built	1
TRAIN	yr_built	1966	37	250	N	1.156711	67	INPUT	yr_built	1
TRAIN	yr_built	1994	15	249	N	1.152084	95	INPUT	yr_built	1
TRAIN	yr_built	1960	8	248	N	1.147458	61	INPUT	yr_built	1
TRAIN	yr_built	1980	61	240	N	1.110443	81	INPUT	yr_built	1
TRAIN	yr_built	1998	95	239	N	1.105816	99	INPUT	yr_built	1
TRAIN	yr_built	1948	25	235	N	1.087309	49	INPUT	yr_built	1
TRAIN	yr_built	2009	107	230	N	1.064174	110	INPUT	yr_built	1
TRAIN	yr_built	1951	1	229	N	1.059547	52	INPUT	yr_built	1
TRAIN	yr_built	1984	33	229	N	1.059547	85	INPUT	yr_built	1
TRAIN	yr_built	1985	21	228	N	1.054921	86	INPUT	yr_built	1
TRAIN	yr_built	1958	99	224	N	1.036413	59	INPUT	yr_built	1
TRAIN	yr_built	1961	53	224	N	1.036413	62	INPUT	yr_built	1
TRAIN	yr_built	1991	41	224	N	1.036413	92	INPUT	yr_built	1
TRAIN	yr_built	1942	10	223	N	1.031786	43	INPUT	yr_built	1
TRAIN	yr_built	1953	38	223	N	1.031786	54	INPUT	yr_built	1

Data Role	Variable Name	Level	CODE	Frequency Count	Type	Percent	Level Index	Role	Label	Plot
TRAIN	yr_built	2002	49	222	N	1.02716	103	INPUT	yr_built	1
TRAIN	yr_built	1952	52	220	N	1.017906	53	INPUT	yr_built	1
TRAIN	yr_built	2000	32	218	N	1.008652	101	INPUT	yr_built	1
TRAIN	yr_built	1986	47	215	N	0.994772	87	INPUT	yr_built	1
TRAIN	yr_built	1983	63	212	N	0.980891	84	INPUT	yr_built	1
TRAIN	yr_built	1993	80	202	N	0.934623	94	INPUT	yr_built	1
TRAIN	yr_built	2013	97	201	N	0.929996	114	INPUT	yr_built	1
TRAIN	yr_built	1981	28	199	N	0.920742	82	INPUT	yr_built	1
TRAIN	yr_built	1956	48	198	N	0.916115	57	INPUT	yr_built	1
TRAIN	yr_built	1992	50	198	N	0.916115	93	INPUT	yr_built	1
TRAIN	yr_built	1957	84	197	N	0.911488	58	INPUT	yr_built	1
TRAIN	yr_built	1949	77	195	N	0.902235	50	INPUT	yr_built	1
TRAIN	yr_built	1996	31	195	N	0.902235	97	INPUT	yr_built	1
TRAIN	yr_built	1975	60	189	N	0.874474	76	INPUT	yr_built	1
TRAIN	yr_built	1965	3	187	N	0.86522	66	INPUT	yr_built	1
TRAIN	yr_built	1926	72	180	N	0.832832	27	INPUT	yr_built	1
TRAIN	yr_built	1997	82	177	N	0.818952	98	INPUT	yr_built	1
TRAIN	yr_built	1964	51	172	N	0.795817	65	INPUT	yr_built	1
TRAIN	yr_built	1943	83	170	N	0.786564	44	INPUT	yr_built	1
TRAIN	yr_built	2012	100	170	N	0.786564	113	INPUT	yr_built	1
TRAIN	yr_built	1995	6	169	N	0.781937	96	INPUT	yr_built	1
TRAIN	yr_built	1925	44	165	N	0.763429	26	INPUT	yr_built	1
TRAIN	yr_built	1974	88	162	N	0.749549	75	INPUT	yr_built	1
TRAIN	yr_built	1941	22	161	N	0.744922	42	INPUT	yr_built	1
TRAIN	yr_built	1940	85	156	N	0.721788	41	INPUT	yr_built	1
TRAIN	yr_built	1972	46	149	N	0.6894	73	INPUT	yr_built	1
TRAIN	yr_built	1973	43	149	N	0.6894	74	INPUT	yr_built	1
TRAIN	yr_built	2010	67	143	N	0.661639	111	INPUT	yr_built	1
TRAIN	yr_built	1944	105	140	N	0.647758	45	INPUT	yr_built	1
TRAIN	yr_built	1924	69	139	N	0.643131	25	INPUT	yr_built	1
TRAIN	yr_built	1910	62	133	N	0.61537	11	INPUT	yr_built	1
TRAIN	yr_built	1970	109	132	N	0.610744	71	INPUT	yr_built	1
TRAIN	yr_built	2011	102	130	N	0.60149	112	INPUT	yr_built	1
TRAIN	yr_built	1928	87	126	N	0.582982	29	INPUT	yr_built	1
TRAIN	yr_built	1946	58	126	N	0.582982	47	INPUT	yr_built	1
TRAIN	yr_built	1918	86	120	N	0.555221	19	INPUT	yr_built	1
TRAIN	yr_built	1927	11	114	N	0.52746	28	INPUT	yr_built	1
TRAIN	yr_built	1929	27	114	N	0.52746	30	INPUT	yr_built	1
TRAIN	yr_built	1939	57	106	N	0.490446	40	INPUT	yr_built	1
TRAIN	yr_built	1982	92	105	N	0.485819	83	INPUT	yr_built	1
TRAIN	yr_built	1971	66	104	N	0.481192	72	INPUT	yr_built	1
TRAIN	yr_built	1920	81	98	N	0.453431	21	INPUT	yr_built	1

Data Role	Variable Name ▼	Level	CODE	Frequency Count	Type	Percent	Level Index	Role	Label	Plot
TRAIN	yr_built	1922	35	95	N	0.43955	23	INPUT	yr_built	1
TRAIN	yr_built	1945	68	95	N	0.43955	46	INPUT	yr_built	1
TRAIN	yr_built	1909	24	94	N	0.434923	10	INPUT	yr_built	1
TRAIN	yr_built	1906	114	92	N	0.42567	7	INPUT	yr_built	1
TRAIN	yr_built	1930	29	90	N	0.416416	31	INPUT	yr_built	1
TRAIN	yr_built	1919	113	88	N	0.407162	20	INPUT	yr_built	1
TRAIN	yr_built	1900	13	87	N	0.402536	1	INPUT	yr_built	1
TRAIN	yr_built	1908	93	85	N	0.393282	9	INPUT	yr_built	1
TRAIN	yr_built	1923	74	84	N	0.388655	24	INPUT	yr_built	1
TRAIN	yr_built	1912	101	79	N	0.365521	13	INPUT	yr_built	1
TRAIN	yr_built	1916	16	79	N	0.365521	17	INPUT	yr_built	1
TRAIN	yr_built	1921	17	75	N	0.347013	22	INPUT	yr_built	1
TRAIN	yr_built	1905	65	74	N	0.342387	6	INPUT	yr_built	1
TRAIN	yr_built	1911	89	73	N	0.33776	12	INPUT	yr_built	1
TRAIN	yr_built	1937	91	68	N	0.314625	38	INPUT	yr_built	1
TRAIN	yr_built	1907	98	65	N	0.300745	8	INPUT	yr_built	1
TRAIN	yr_built	1915	23	64	N	0.296118	16	INPUT	yr_built	1
TRAIN	yr_built	1931	94	60	N	0.277611	32	INPUT	yr_built	1
TRAIN	yr_built	1913	96	59	N	0.272984	14	INPUT	yr_built	1
TRAIN	yr_built	1917	103	56	N	0.259103	18	INPUT	yr_built	1
TRAIN	yr_built	1914	71	54	N	0.24985	15	INPUT	yr_built	1
TRAIN	yr_built	1938	112	52	N	0.240596	39	INPUT	yr_built	1
TRAIN	yr_built	1903	108	46	N	0.212835	4	INPUT	yr_built	1
TRAIN	yr_built	1904	30	45	N	0.208208	5	INPUT	yr_built	1
TRAIN	yr_built	1936	90	40	N	0.185074	37	INPUT	yr_built	1
TRAIN	yr_built	1932	104	38	N	0.17582	33	INPUT	yr_built	1
TRAIN	yr_built	2015	110	38	N	0.17582	116	INPUT	yr_built	1
TRAIN	yr_built	1933	2	30	N	0.138805	34	INPUT	yr_built	1
TRAIN	yr_built	1901	79	29	N	0.134179	2	INPUT	yr_built	1
TRAIN	yr_built	1902	106	27	N	0.124925	3	INPUT	yr_built	1
TRAIN	yr_built	1935	115	24	N	0.111044	36	INPUT	yr_built	1
TRAIN	yr_built	1934	111	21	N	0.097164	35	INPUT	yr_built	1
TRAIN	yr_built	190810	118	1	N	0.004627	117	INPUT	yr_built	1
TRAIN	yr_built	191006	122	1	N	0.004627	118	INPUT	yr_built	1
TRAIN	yr_built	192102	116	1	N	0.004627	119	INPUT	yr_built	1
TRAIN	yr_built	192703	120	1	N	0.004627	120	INPUT	yr_built	1
TRAIN	yr_built	19310401	119	1	N	0.004627	121	INPUT	yr_built	1
TRAIN	yr_built	19570522	117	1	N	0.004627	122	INPUT	yr_built	1
TRAIN	yr_built	19590731	121	1	N	0.004627	123	INPUT	yr_built	1

Data Role	Variable Name ▲	Level	CODE	Frequency Count	Type	Percent	Level Index	Role	Label	Plot
TRAIN	yr_renovated	0	0	20699	N	95.77106	1	INPUT	yr_renovated	1
TRAIN	yr_renovated	2014	14	91	N	0.421043	69	INPUT	yr_renovated	1
TRAIN	yr_renovated	2013	6	37	N	0.171193	68	INPUT	yr_renovated	1
TRAIN	yr_renovated	2003	11	36	N	0.166566	58	INPUT	yr_renovated	1
TRAIN	yr_renovated	2000	25	35	N	0.16194	55	INPUT	yr_renovated	1
TRAIN	yr_renovated	2005	9	35	N	0.16194	60	INPUT	yr_renovated	1
TRAIN	yr_renovated	2007	32	35	N	0.16194	62	INPUT	yr_renovated	1
TRAIN	yr_renovated	2004	29	26	N	0.120298	59	INPUT	yr_renovated	1
TRAIN	yr_renovated	1990	19	25	N	0.115671	45	INPUT	yr_renovated	1
TRAIN	yr_renovated	2006	35	24	N	0.111044	61	INPUT	yr_renovated	1
TRAIN	yr_renovated	1989	28	22	N	0.101791	44	INPUT	yr_renovated	1
TRAIN	yr_renovated	2002	2	22	N	0.101791	57	INPUT	yr_renovated	1
TRAIN	yr_renovated	2009	31	22	N	0.101791	64	INPUT	yr_renovated	1
TRAIN	yr_renovated	1991	1	20	N	0.092537	46	INPUT	yr_renovated	1
TRAIN	yr_renovated	1993	50	19	N	0.08791	48	INPUT	yr_renovated	1
TRAIN	yr_renovated	1994	7	19	N	0.08791	49	INPUT	yr_renovated	1
TRAIN	yr_renovated	1998	26	19	N	0.08791	53	INPUT	yr_renovated	1
TRAIN	yr_renovated	2001	37	19	N	0.08791	56	INPUT	yr_renovated	1
TRAIN	yr_renovated	1983	17	18	N	0.083283	38	INPUT	yr_renovated	1
TRAIN	yr_renovated	1984	12	18	N	0.083283	39	INPUT	yr_renovated	1
TRAIN	yr_renovated	1987	33	18	N	0.083283	42	INPUT	yr_renovated	1
TRAIN	yr_renovated	2008	10	18	N	0.083283	63	INPUT	yr_renovated	1
TRAIN	yr_renovated	2010	3	18	N	0.083283	65	INPUT	yr_renovated	1
TRAIN	yr_renovated	1985	36	17	N	0.078656	40	INPUT	yr_renovated	1
TRAIN	yr_renovated	1986	30	17	N	0.078656	41	INPUT	yr_renovated	1
TRAIN	yr_renovated	1992	5	17	N	0.078656	47	INPUT	yr_renovated	1
TRAIN	yr_renovated	1999	4	17	N	0.078656	54	INPUT	yr_renovated	1
TRAIN	yr_renovated	1995	24	16	N	0.07403	50	INPUT	yr_renovated	1
TRAIN	yr_renovated	2015	45	16	N	0.07403	70	INPUT	yr_renovated	1
TRAIN	yr_renovated	1988	20	15	N	0.069403	43	INPUT	yr_renovated	1
TRAIN	yr_renovated	1996	52	15	N	0.069403	51	INPUT	yr_renovated	1
TRAIN	yr_renovated	1997	41	15	N	0.069403	52	INPUT	yr_renovated	1
TRAIN	yr_renovated	2011	15	13	N	0.060149	66	INPUT	yr_renovated	1
TRAIN	yr_renovated	1980	38	11	N	0.050895	35	INPUT	yr_renovated	1
TRAIN	yr_renovated	1982	56	11	N	0.050895	37	INPUT	yr_renovated	1
TRAIN	yr_renovated	2012	47	11	N	0.050895	67	INPUT	yr_renovated	1
TRAIN	yr_renovated	1979	40	10	N	0.046268	34	INPUT	yr_renovated	1
TRAIN	yr_renovated	1970	27	9	N	0.041642	25	INPUT	yr_renovated	1
TRAIN	yr_renovated	1968	46	8	N	0.037015	23	INPUT	yr_renovated	1
TRAIN	yr_renovated	1977	22	8	N	0.037015	32	INPUT	yr_renovated	1
TRAIN	yr_renovated	1975	61	6	N	0.027761	30	INPUT	yr_renovated	1
TRAIN	yr_renovated	1978	8	6	N	0.027761	33	INPUT	yr_renovated	1
TRAIN	yr_renovated	1958	62	5	N	0.023134	15	INPUT	yr_renovated	1
TRAIN	yr_renovated	1964	63	5	N	0.023134	20	INPUT	yr_renovated	1
TRAIN	yr_renovated	1965	67	5	N	0.023134	21	INPUT	yr_renovated	1
TRAIN	yr_renovated	1973	34	5	N	0.023134	28	INPUT	yr_renovated	1
TRAIN	yr_renovated	1981	23	5	N	0.023134	36	INPUT	yr_renovated	1
TRAIN	yr_renovated	1960	65	4	N	0.018507	17	INPUT	yr_renovated	1
TRAIN	yr_renovated	1963	48	4	N	0.018507	19	INPUT	yr_renovated	1
TRAIN	yr_renovated	1969	43	4	N	0.018507	24	INPUT	yr_renovated	1
TRAIN	yr_renovated	1972	53	4	N	0.018507	27	INPUT	yr_renovated	1
TRAIN	yr_renovated	1945	18	3	N	0.013881	5	INPUT	yr_renovated	1
TRAIN	yr_renovated	1953	54	3	N	0.013881	10	INPUT	yr_renovated	1
TRAIN	yr_renovated	1955	55	3	N	0.013881	12	INPUT	yr_renovated	1
TRAIN	yr_renovated	1956	57	3	N	0.013881	13	INPUT	yr_renovated	1
TRAIN	yr_renovated	1957	21	3	N	0.013881	14	INPUT	yr_renovated	1
TRAIN	yr_renovated	1974	16	3	N	0.013881	29	INPUT	yr_renovated	1
TRAIN	yr_renovated	1976	59	3	N	0.013881	31	INPUT	yr_renovated	1
TRAIN	yr_renovated	1940	58	2	N	0.009254	3	INPUT	yr_renovated	1
TRAIN	yr_renovated	1946	60	2	N	0.009254	6	INPUT	yr_renovated	1
TRAIN	yr_renovated	1950	42	2	N	0.009254	8	INPUT	yr_renovated	1
TRAIN	yr_renovated	1962	51	2	N	0.009254	18	INPUT	yr_renovated	1
TRAIN	yr_renovated	1967	66	2	N	0.009254	22	INPUT	yr_renovated	1
TRAIN	yr_renovated	1971	39	2	N	0.009254	26	INPUT	yr_renovated	1
TRAIN	yr_renovated	1934	68	1	N	0.004627	2	INPUT	yr_renovated	1
TRAIN	yr_renovated	1944	69	1	N	0.004627	4	INPUT	yr_renovated	1
TRAIN	yr_renovated	1948	44	1	N	0.004627	7	INPUT	yr_renovated	1
TRAIN	yr_renovated	1951	49	1	N	0.004627	9	INPUT	yr_renovated	1
TRAIN	yr_renovated	1954	13	1	N	0.004627	11	INPUT	yr_renovated	1
TRAIN	yr_renovated	1959	64	1	N	0.004627	16	INPUT	yr_renovated	1

Figure 5.8 Summary Statistics for Nominal Variables

The above summary statistics for nominal variables are shown to support the visualization.

There is an inconsistency data error in the yr_built variable as highlighted in the red circle, which are 190810, 191006, 192102, 192703, 19310401, 19570522 and 19590731.

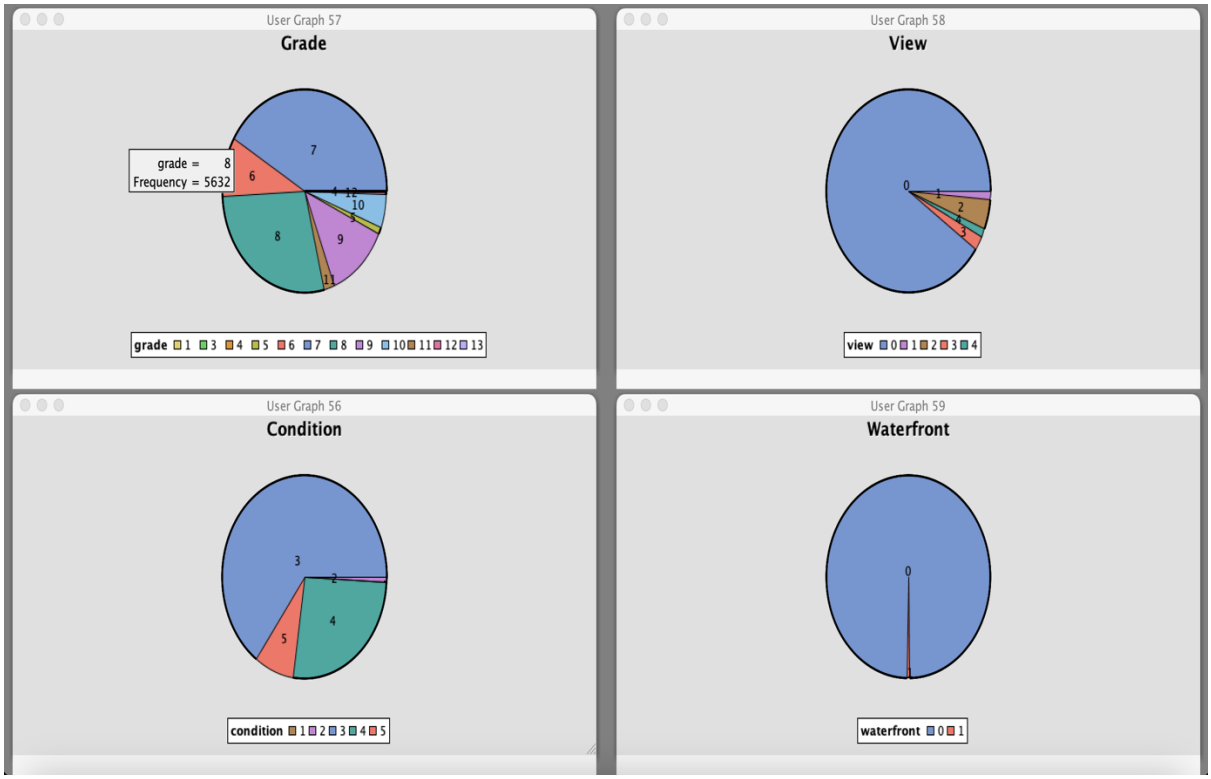


Figure 5.9 Pie Chart

We use Pie Charts to check the distribution of categorical variables. We observed that all our categorical variables are imbalance in term of the distribution.

Data Role	Variable Name	Level	CODE	Frequency Count	Type	Percent	Level Index	Role	Label	Plot
TRAIN	condition	3	0	14031	N	64.91926	3	INPUT	condition	1
TRAIN	condition	4	2	5679	N	26.27585	4	INPUT	condition	1
TRAIN	condition	5	1	1701	N	7.870263	5	INPUT	condition	1
TRAIN	condition	2	4	172	N	0.795817	2	INPUT	condition	1
TRAIN	condition	1	3	30	N	0.138805	1	INPUT	condition	1
TRAIN	grade	7	0	8981	N	41.55369	6	INPUT	grade	1
TRAIN	grade	8	2	6068	N	28.0757	7	INPUT	grade	1
TRAIN	grade	9	4	2615	N	12.0992	8	INPUT	grade	1
TRAIN	grade	6	1	2038	N	9.42951	5	INPUT	grade	1
TRAIN	grade	10	6	1134	N	5.246842	9	INPUT	grade	1
TRAIN	grade	11	3	399	N	1.846111	10	INPUT	grade	1
TRAIN	grade	5	5	242	N	1.119696	4	INPUT	grade	1
TRAIN	grade	12	7	90	N	0.416416	11	INPUT	grade	1
TRAIN	grade	4	8	29	N	0.134179	3	INPUT	grade	1
TRAIN	grade	13	10	13	N	0.060149	12	INPUT	grade	1
TRAIN	grade	3	9	3	N	0.013881	2	INPUT	grade	1
TRAIN	grade	1	11	1	N	0.004627	1	INPUT	grade	1
TRAIN	view	0	0	19489	N	90.17258	1	INPUT	view	1
TRAIN	view	2	3	963	N	4.455652	3	INPUT	view	1
TRAIN	view	3	1	510	N	2.359691	4	INPUT	view	1
TRAIN	view	1	4	332	N	1.536113	2	INPUT	view	1
TRAIN	view	4	2	319	N	1.475964	5	INPUT	view	1
TRAIN	waterfront	0	0	21450	N	99.24582	1	INPUT	waterfront	1
TRAIN	waterfront	1	1	163	N	0.754176	2	INPUT	waterfront	1

Figure 5.10 Summary Statistics for Class Variables

The above summary statistics for class variables are shown to support the visualization.

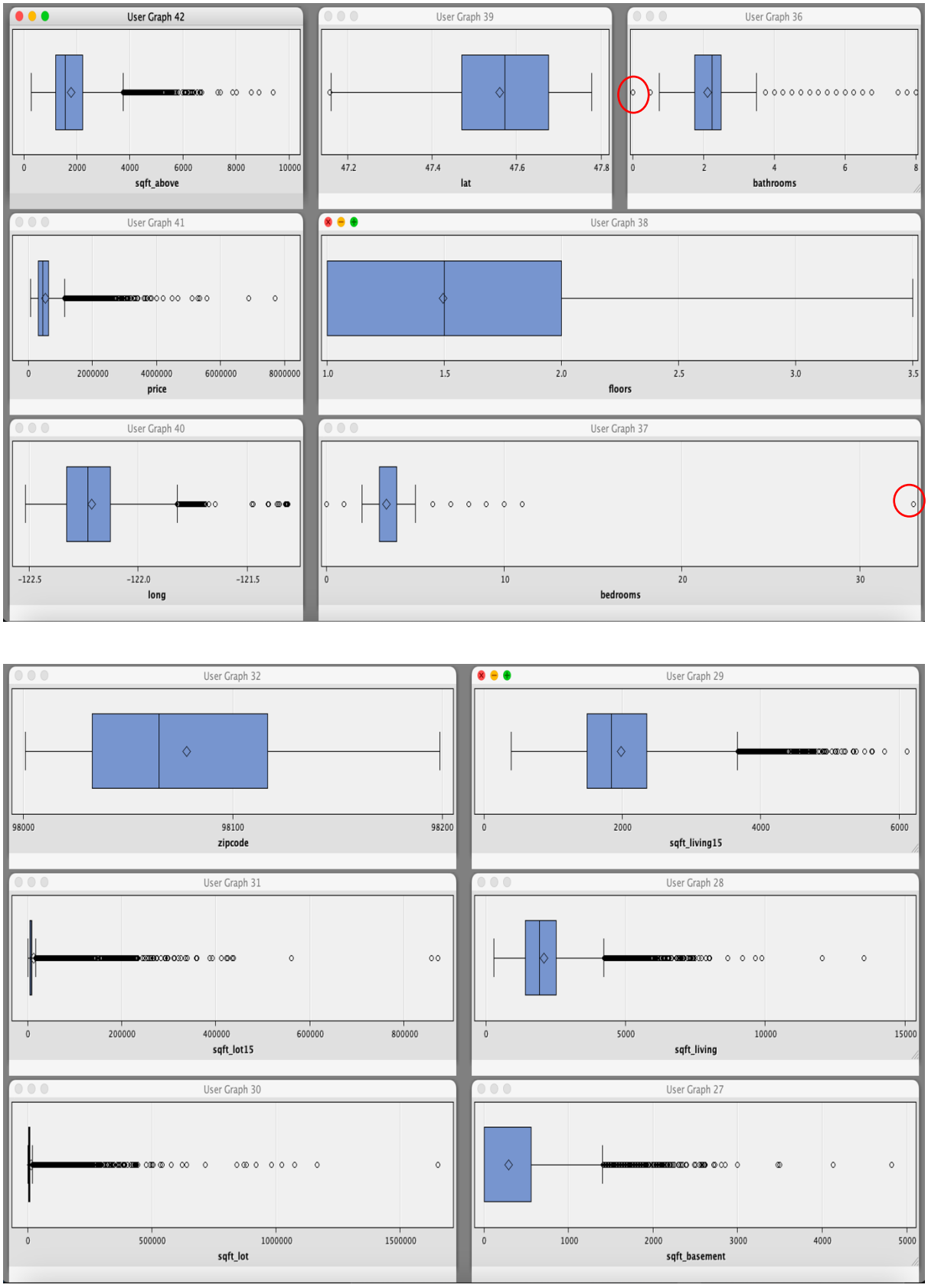


Figure 5.11 Boxplot

We use boxplot to check for outliers in an interval variable. The figure above shows the boxplot of different interval variables. There are outliers in bathrooms variable, bedrooms variable, sqft_above variable, price variable, lat variable, long variable, sqft_living15 variable, sqft_lot15 variable, sqft_living variable, sqft_lot variable and sqft_basement variable. If the outliers are due to error, it should be removed. For sqft_above variables, price variables, lat variables, long variables, sqft_living15 variables, sqft_lot15 variables, sqft_living variables, sqft_lot variables and sqft_basement variables, we observed that outliers have mostly the same distribution as the non-outliers, because there are many points outside the fence, the outliers are unlikely due to some error and should be kept.

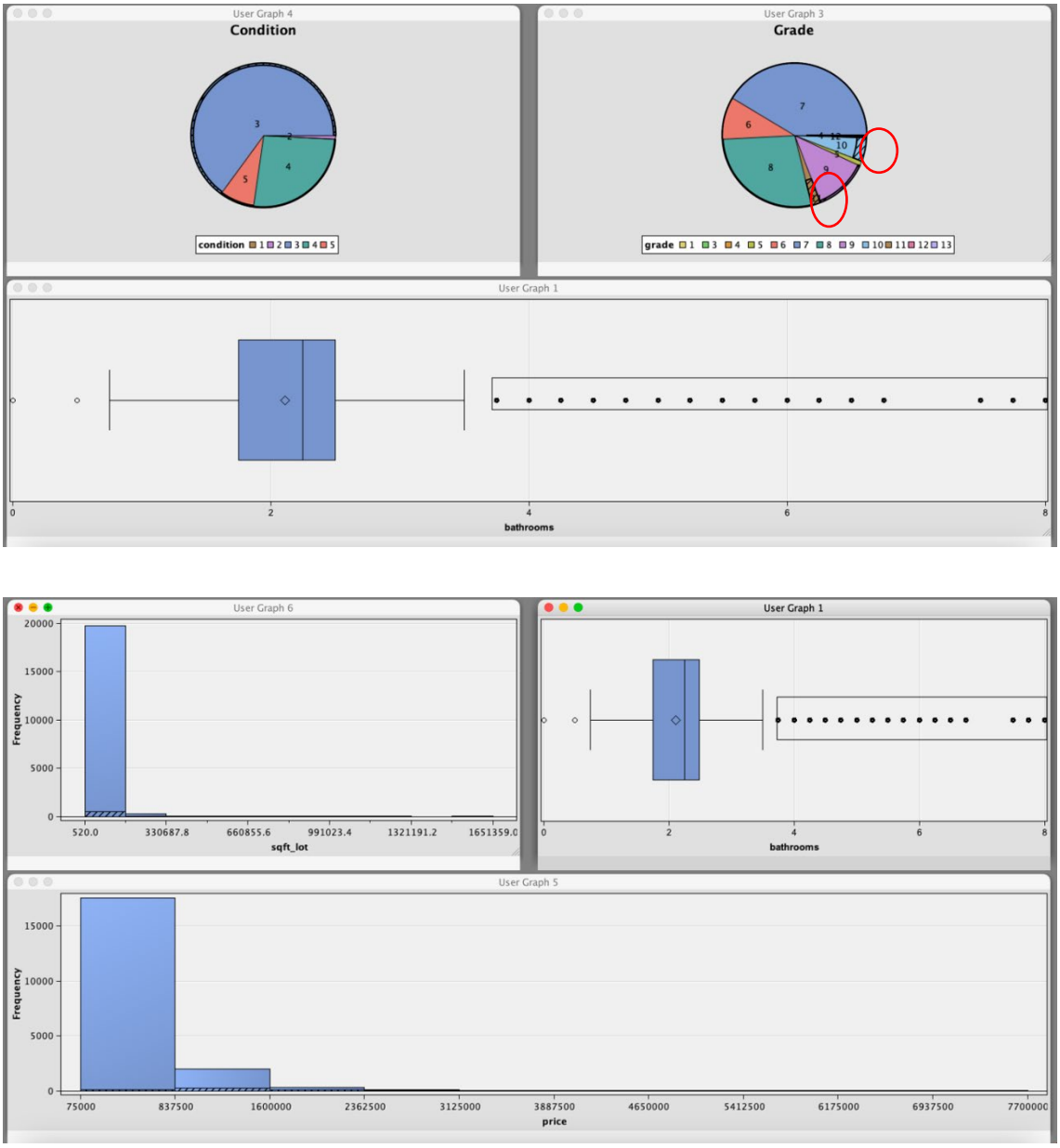
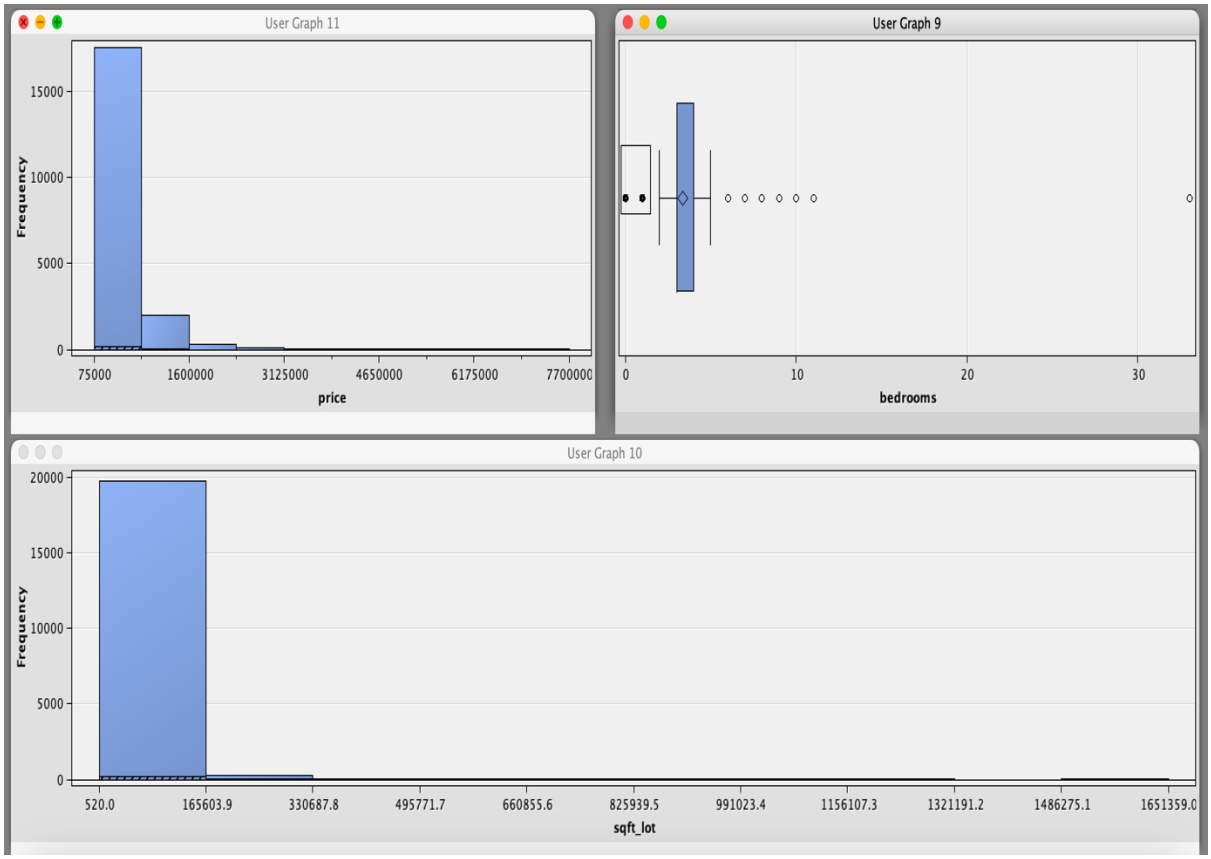
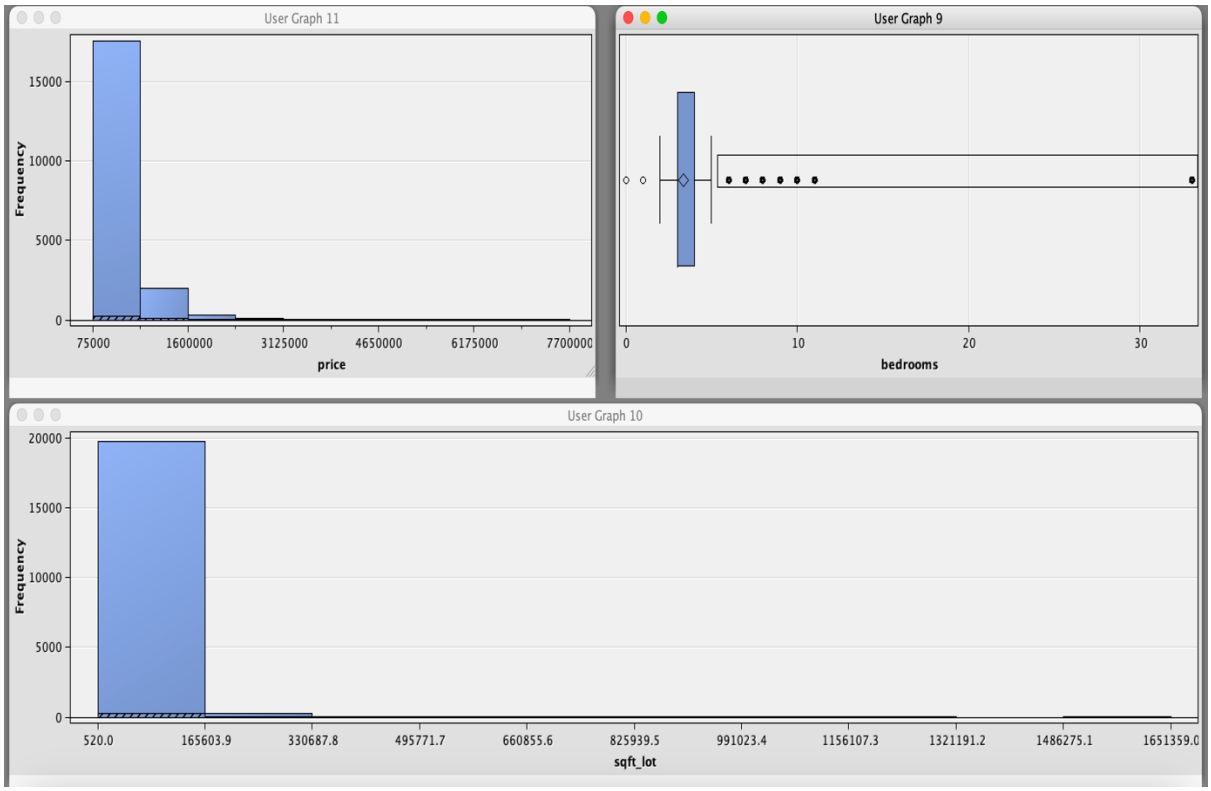


Figure 5.12 Variable Association of Outliers in Bathrooms and Other Variables

We investigate whether the outlier should be removed by examining the variable association between the outlier and other variables. First is the bathrooms variable. We observe that the bathrooms variable has an outlier of 0 bathrooms. There are ten observations with 0 bathrooms. Since it is not conventional to have houses without bathroom, these outliers should be removed. For other outliers, we observed that house with more bathrooms tend be higher grade as highlighted in the red circle. These characteristics are consistent with price. As the outliers have mostly the same distribution as the non-outliers, so the outliers are unlikely to be due to some error and should be kept.


User Graph 11
Frequency
15000
10000
5000
0
75000
1600000
3125000
4650000
6175000
7700000
price
User Graph 9
0
10
20
30
bedrooms
User Graph 10
Frequency
20000
15000
10000
5000
0
520.0
165603.9
330687.8
495771.7
660855.6
825939.5
991023.4
1156107.3
1321191.2
1486275.1
1651359.0
sqft_lot


User Graph 11
Frequency
15000
10000
5000
0
75000
1600000
3125000
4650000
6175000
7700000
price
User Graph 9
0
10
20
30
bedrooms
User Graph 10
Frequency
20000
15000
10000
5000
0
520.0
165603.9
330687.8
495771.7
660855.6
825939.5
991023.4
1156107.3
1321191.2
1486275.1
1651359.0
sqft_lot

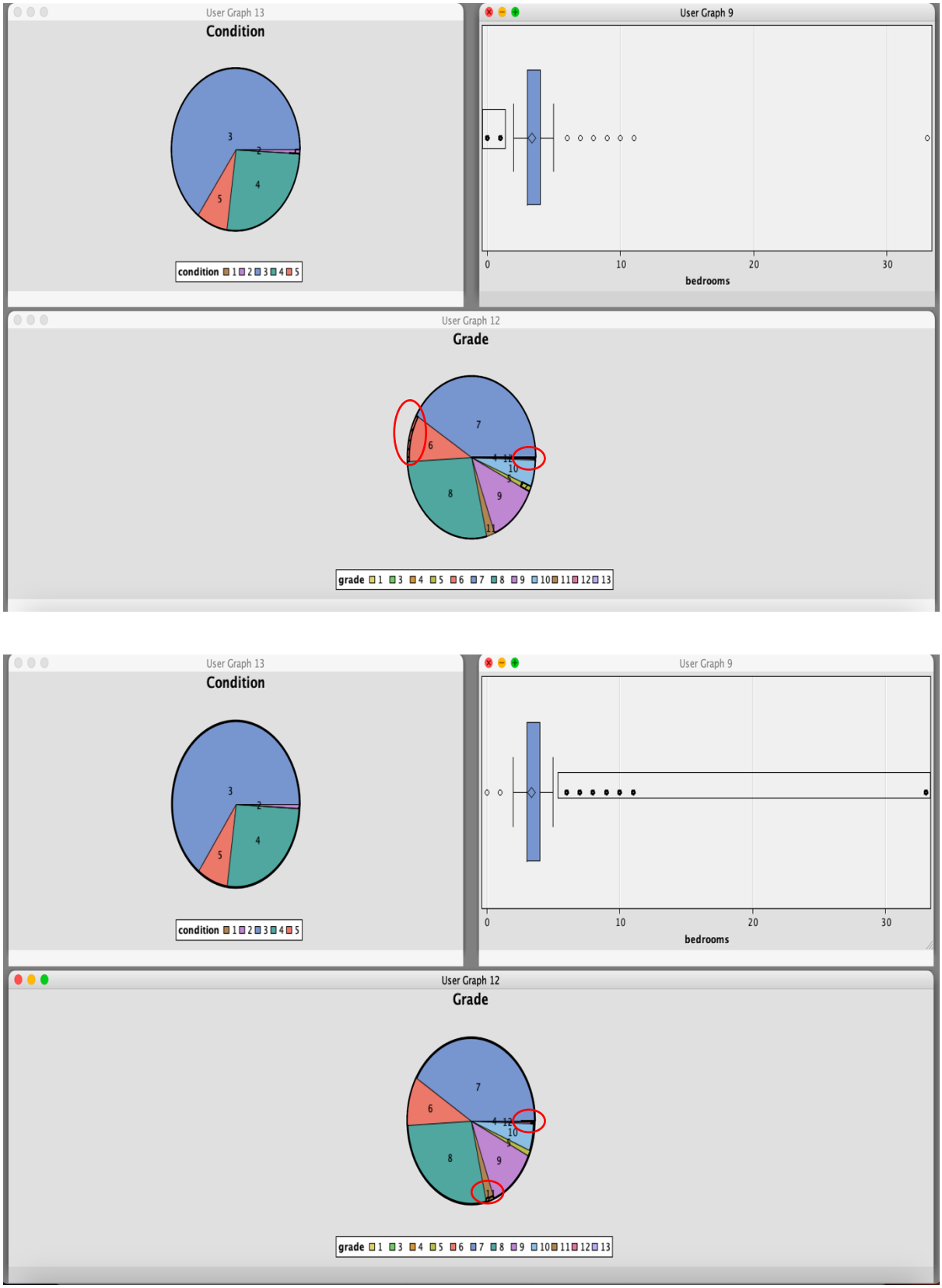


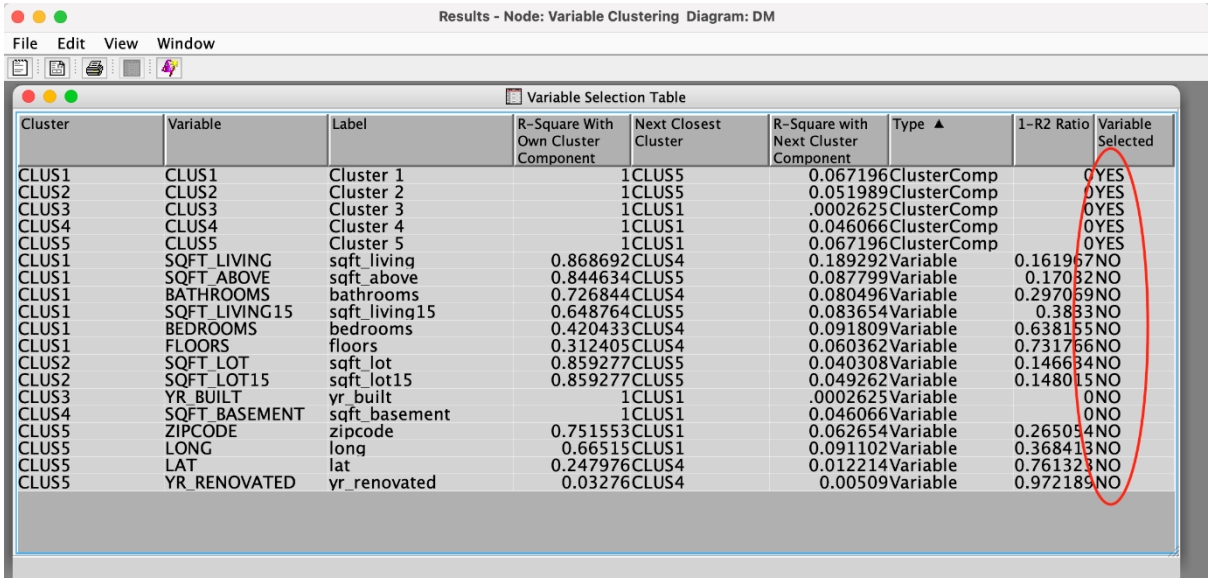
Figure 5.13 Variable Association of Outliers in Bedrooms and Other Variables

Second is the bedrooms variable. We observe that there is a suspicious 33 room entry which is totally unrealizable. Therefore, it is likely that human error causes the outlier, and the point should be removed. We observe that the bedrooms variable has an outlier of 0 bedrooms. There are ten observations with 0 bathrooms. Although we noticed that the house with 0 bathrooms has grade of 5 to 6 as highlighted in the red circle, but since it is not conventional to have house without bathrooms, these outliers should be removed. For other outliers, we observed that house with more bedrooms tend be higher grade as highlighted in the red circle. These characteristics are consistent with price. As the outliers have mostly the same distribution as the non-outliers, so the outliers are unlikely to be due to some error and should be kept.

As a conclusion for univariate analysis, we learned that 1) There is incomplete error in the bathrooms variable. 2) There are inconsistency errors in the yr_built variable. 3) There are noisy errors in the bathrooms variable, bedrooms variable, sqft_above variable, price variable, lot variable, long variable, sqft_living15 variable, sqft_lot15 variable, sqft_living variable, sqft_lot variable and sqft_basement variable.

5.2.2 Bivariate Analysis

Bivariate analysis is an analysis conducted with 2 different variables mainly **to understand if there is a relationship among the chosen variables**. At the same time, this **analysis is also used to identify the redundant variables** and to **examine for relevant variables**.



Cluster	Variable	Label	R-Square With Own Cluster Component	Next Closest Cluster	R-Square with Next Cluster Component	Type ▲	1-R2 Ratio	Variable Selected
CLUS1	CLUS1	Cluster 1	1	CLUS5	0.067196	ClusterComp	0	YES
CLUS2	CLUS2	Cluster 2	1	CLUS5	0.051989	ClusterComp	0	YES
CLUS3	CLUS3	Cluster 3	1	CLUS1	.0002625	ClusterComp	0	YES
CLUS4	CLUS4	Cluster 4	1	CLUS1	0.046066	ClusterComp	0	YES
CLUS5	CLUS5	Cluster 5	1	CLUS1	0.067196	ClusterComp	0	YES
CLUS1	SQFT_LIVING	sqft_living	0.868692	CLUS4	0.189292	Variable	0.161967	NO
CLUS1	SQFT_ABOVE	sqft_above	0.844634	CLUS5	0.087799	Variable	0.17032	NO
CLUS1	BATHROOMS	bathrooms	0.726844	CLUS4	0.080496	Variable	0.297069	NO
CLUS1	SQFT_LIVING15	sqft_living15	0.648764	CLUS5	0.083654	Variable	0.3833	NO
CLUS1	BEDROOMS	bedrooms	0.420433	CLUS4	0.091809	Variable	0.638155	NO
CLUS1	FLOORS	floors	0.312405	CLUS4	0.060362	Variable	0.731766	NO
CLUS2	SQFT_LOT	sqft_lot	0.859277	CLUS5	0.040308	Variable	0.146634	NO
CLUS2	SQFT_LOT15	sqft_lot15	0.859277	CLUS5	0.049262	Variable	0.148015	NO
CLUS3	YR_BUILT	yr_built	1	CLUS1	.0002625	Variable	0	NO
CLUS4	SQFT_BASEMENT	sqft_basement	1	CLUS1	0.046066	Variable	0	NO
CLUS5	ZIPCODE	zipcode	0.751553	CLUS1	0.062654	Variable	0.265054	NO
CLUS5	LONG	long	0.66515	CLUS1	0.091102	Variable	0.368413	NO
CLUS5	LAT	lat	0.247976	CLUS4	0.012214	Variable	0.761323	NO
CLUS5	YR_RENOVATED	yr_renovated	0.03276	CLUS4	0.00509	Variable	0.972189	NO

Figure 5.14 Variable Clustering

First, we will be looking at the redundant variable as shown in Figure 5.14. SAS Enterprise Miner tool, namely variable clustering was used to spot for redundant variables. In references to the figure above, no variable belongs to a cluster which indicates that the variables are not redundant and should not be removed from the dataset for next phase.

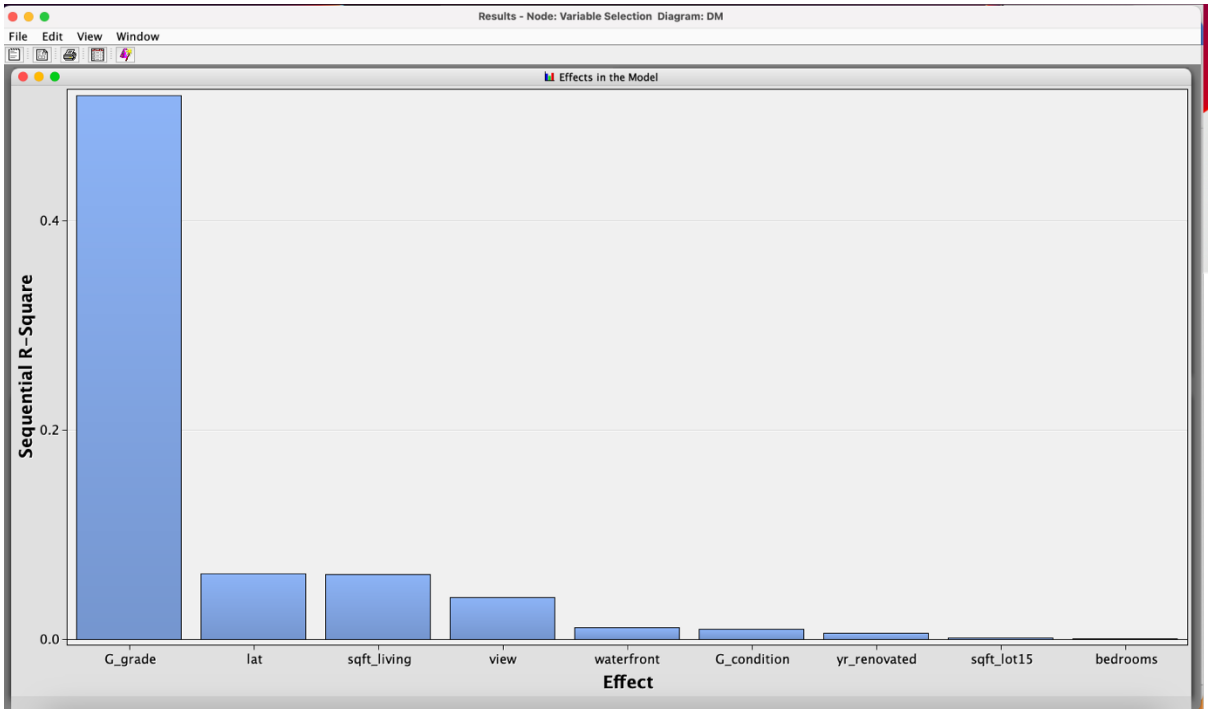
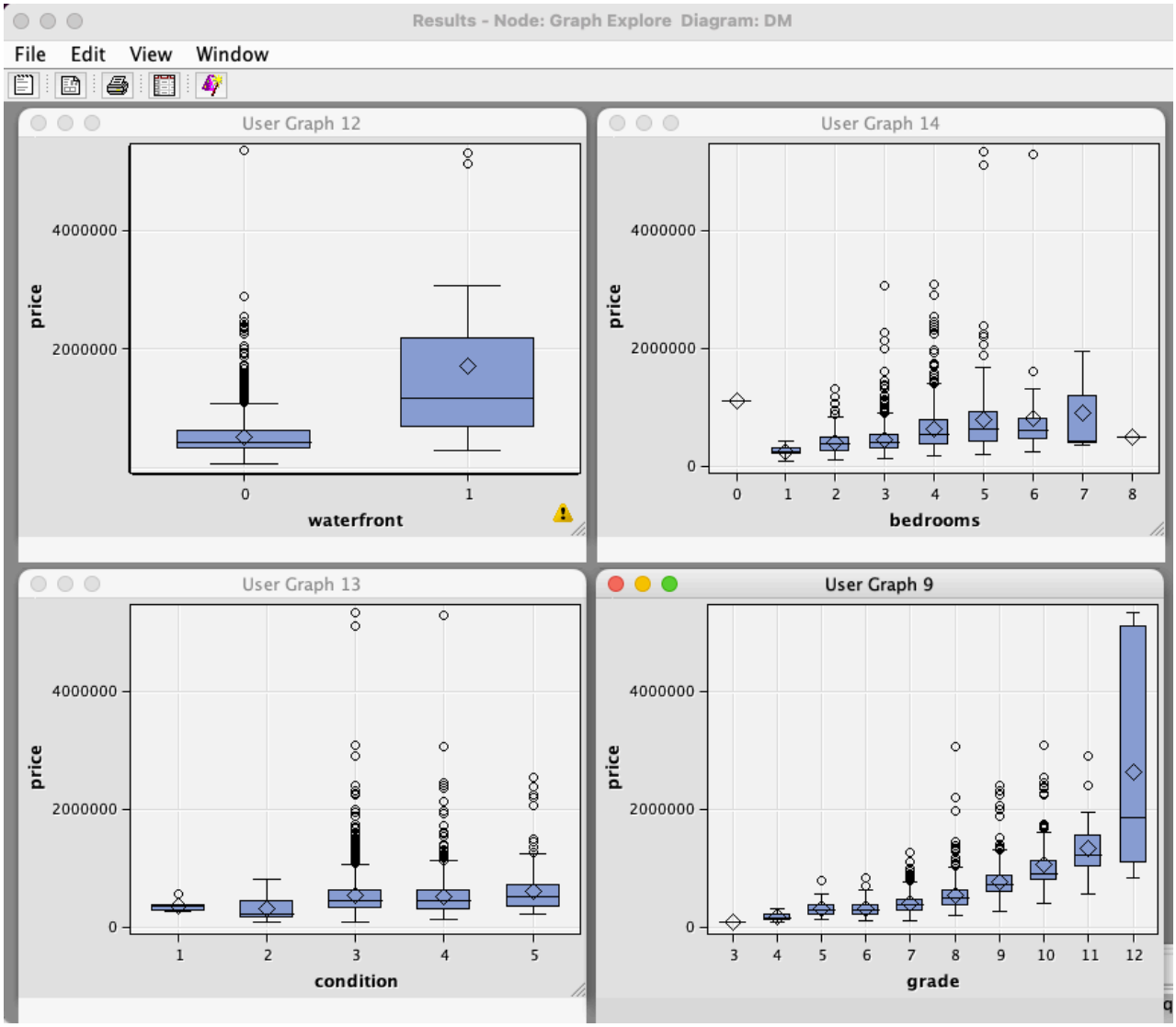


Figure 5.15 Variable Selection

Next, SAS Enterprise Miner was used to locate the relevant variables that contributes to the target variable house sales price (price) using the variable selection tool. Relevant variables identified are the grade, latitude, sqft_living, view, waterfront, condition, yr_renovated, sqft_lot15 and bedrooms. The other variable not listed will be dropped from the dataset.



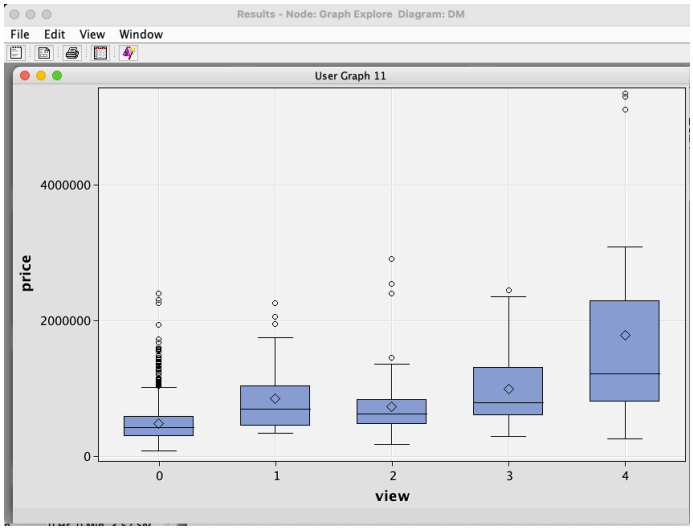


Figure 5.16 Selected Categorical Variable with Target

To further verify the selected findings by visualization, box plot and scatter plot are plotted. Figure 5.16 above shows the bivariate relationship between the target variable, house price (price) and the relevant variables as highlighted earlier.

From the figure above, we noticed that the waterfront (0-without waterfront, 1-with waterfront), bedrooms, condition, grade, view variables increase, the house price variable increases with those variables.

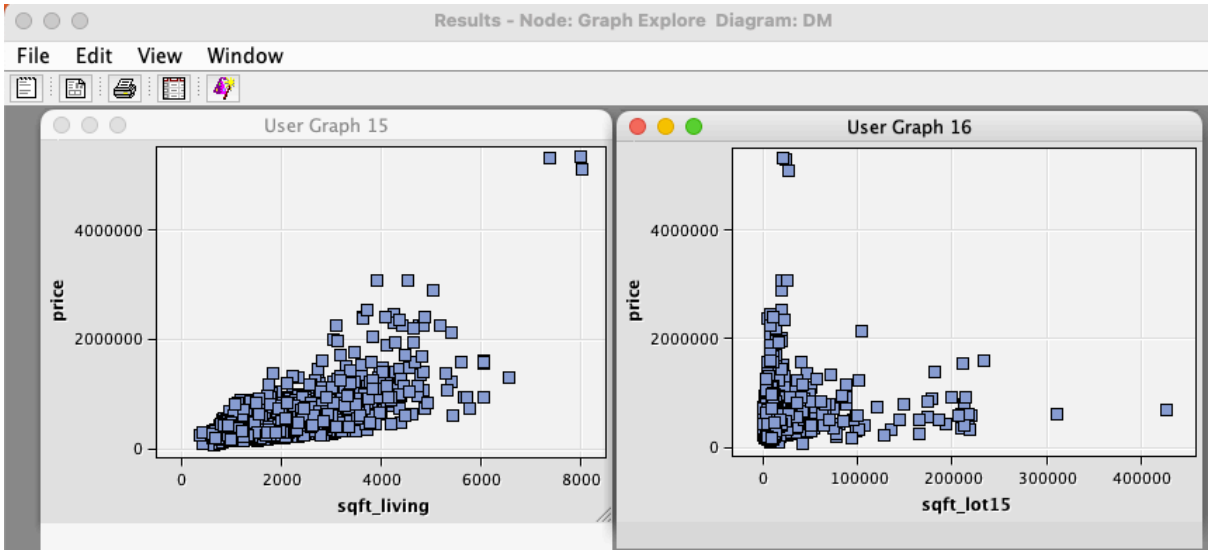


Figure 5.17 Selected Continuous Variable with Target

Referring to Figure 5.17, the scatter plots were plotted for the price variable and both sqft_living and sqft_lot15 respectively. In the figure above, we can see the as the sqft_living increases the price variable increases proportionally. For the sqft_lot15 variable we can see that the higher the increase in the sqft_lot15 the increase in price were in a complex mix or trend given that the outlier positioned at 0 are not taken into consideration.

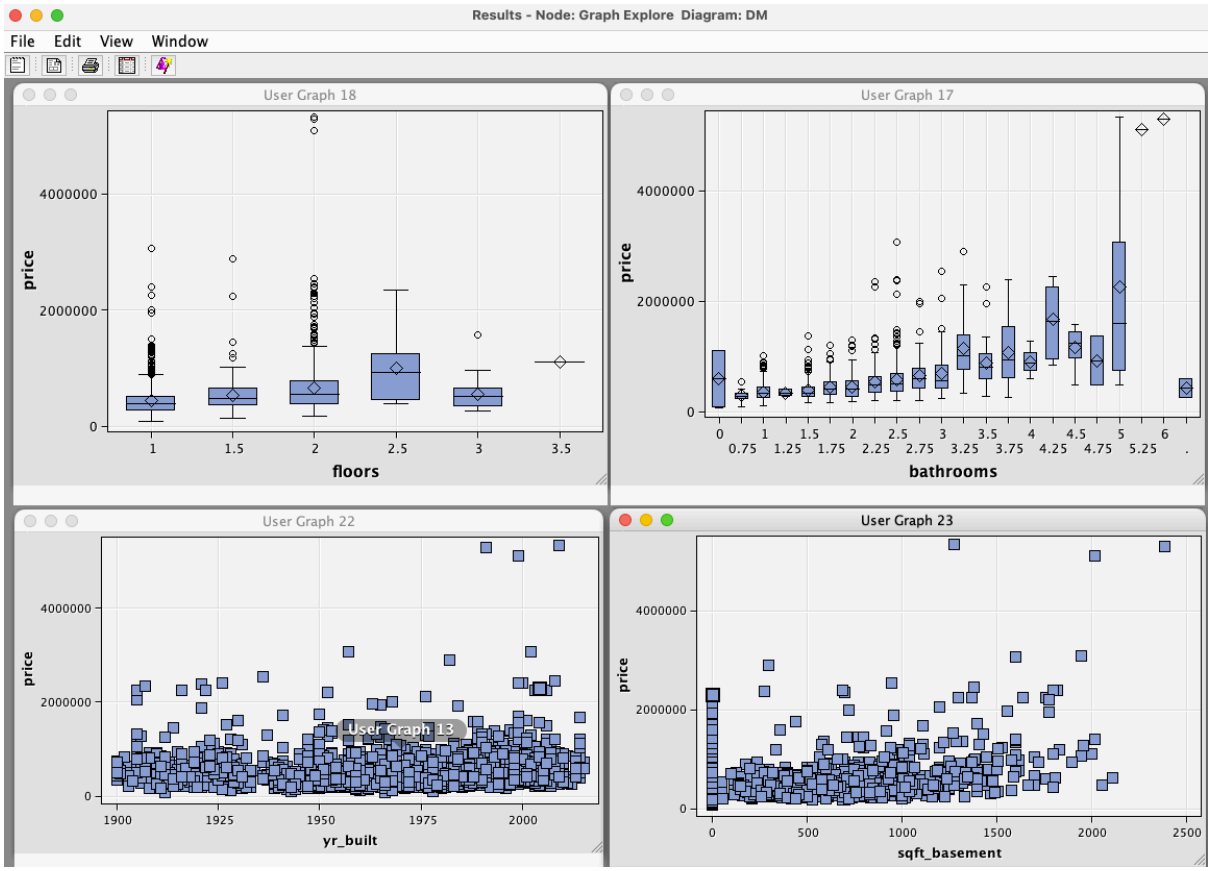


Figure 5.18 Dropped Variables

The figure above shows the several scatter plot and box plot for sqft_basement, yr_built, floor, bathrooms variable with house price (price) variable. From our observations the variable to be dropped are plotted in a constant points instead of following the price trend which proves that these variables do not contribute towards the house the prices and should be dropped as per the results of the previous charts.

In conclusion, the visualization for the relevant variable selection namely grade, latitude, sqft_living, view, waterfront, condition, yr_renovated, sqft_lot15 and bedrooms should be retained and to drop the other variables.

5.2.3 Multivariate Analysis

Multivariate analysis is a type of statistical method that analyses the relationship between several variables. With multivariate analysis, we can observe the patterns and make comparisons between multiple factors at once. It is more useful as we always compare how several factors affects the target variable in real life situations.

Figure 5.19 shows a correlation matrix that shows the correlation between the interval variables. If the color is red, the variables are in positive relationship, blue shows negative relationship and grey shows weaker relationship. The square footage of the apartments is mostly in red, showing positive relationships with several variables based on the correlation matrix.

Besides, we can validate the relationship between the variables using correlation table. Figure 5.20 shows part of the correlation table where the full result is listed in Appendix. We observed that the strongest positive relationship is between the square footage of the apartments interior living space (sqft_living) and the square footage of the interior housing space that is above ground level (sqft_above) with a correlation coefficient of 0.8766. Meanwhile, the strongest negative relationship is between zipcode and longitude with correlation coefficient of -0.5640.

Thus, we can conclude that square footage of the apartments interior living space (sqft_living) and the square footage of the interior housing space that is above ground level (sqft_above) are highly correlated and have a strong positive relationship.

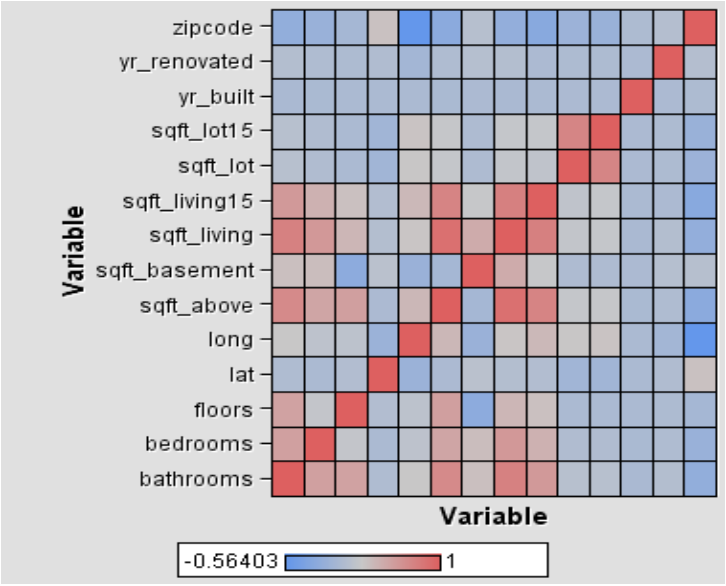


Figure 5.19 Correlation Matrix

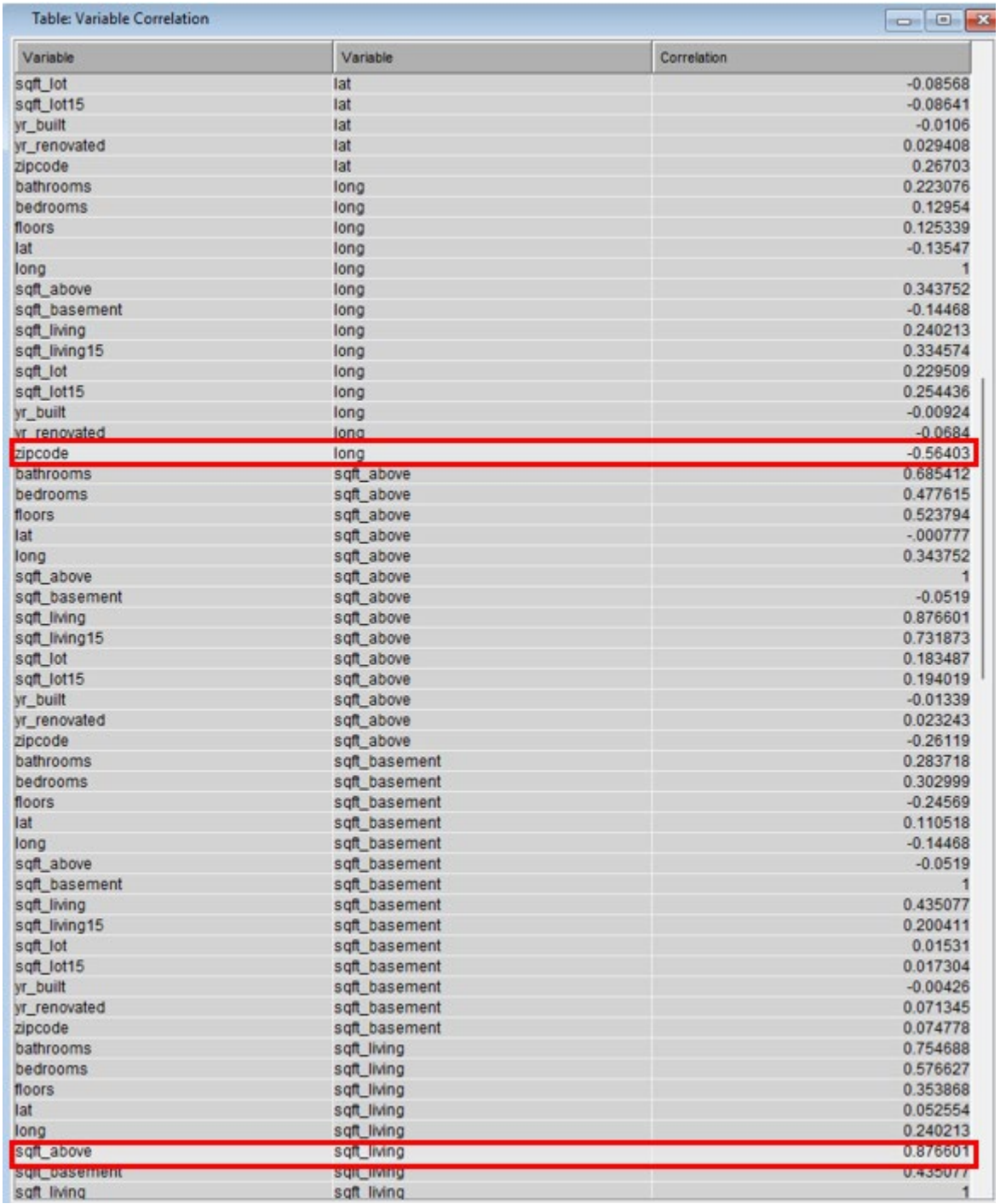


Table: Variable Correlation

Variable	Variable	Correlation
sqft_lot	lat	-0.08568
sqft_lot15	lat	-0.08641
yr_built	lat	-0.0106
yr_renovated	lat	0.029408
zipcode	lat	0.26703
bathrooms	long	0.223076
bedrooms	long	0.12954
floors	long	0.125339
lat	long	-0.13547
long	long	1
sqft_above	long	0.343752
sqft_basement	long	-0.14468
sqft_living	long	0.240213
sqft_living15	long	0.334574
sqft_lot	long	0.229509
sqft_lot15	long	0.254436
yr_built	long	-0.00924
yr_renovated	long	-0.0684
zipcode	long	-0.56403
bathrooms	sqft_above	0.685412
bedrooms	sqft_above	0.477615
floors	sqft_above	0.523794
lat	sqft_above	-.000777
long	sqft_above	0.343752
sqft_above	sqft_above	1
sqft_basement	sqft_above	-0.0519
sqft_living	sqft_above	0.876601
sqft_living15	sqft_above	0.731873
sqft_lot	sqft_above	0.183487
sqft_lot15	sqft_above	0.194019
yr_built	sqft_above	-0.01339
yr_renovated	sqft_above	0.023243
zipcode	sqft_above	-0.26119
bathrooms	sqft_basement	0.283718
bedrooms	sqft_basement	0.302999
floors	sqft_basement	-0.24569
lat	sqft_basement	0.110518
long	sqft_basement	-0.14468
sqft_above	sqft_basement	-0.0519
sqft_basement	sqft_basement	1
sqft_living	sqft_basement	0.435077
sqft_living15	sqft_basement	0.200411
sqft_lot	sqft_basement	0.01531
sqft_lot15	sqft_basement	0.017304
yr_built	sqft_basement	-0.00426
yr_renovated	sqft_basement	0.071345
zipcode	sqft_basement	0.074778
bathrooms	sqft_living	0.754688
bedrooms	sqft_living	0.576627
floors	sqft_living	0.353868
lat	sqft_living	0.052554
long	sqft_living	0.240213
sqft_above	sqft_living	0.876601
sqft_basement	sqft_living	0.435077
sqft_living	sqft_living	1

Figure 5.20 Part of Correlation Table

5.2.4 Interesting Visualizations

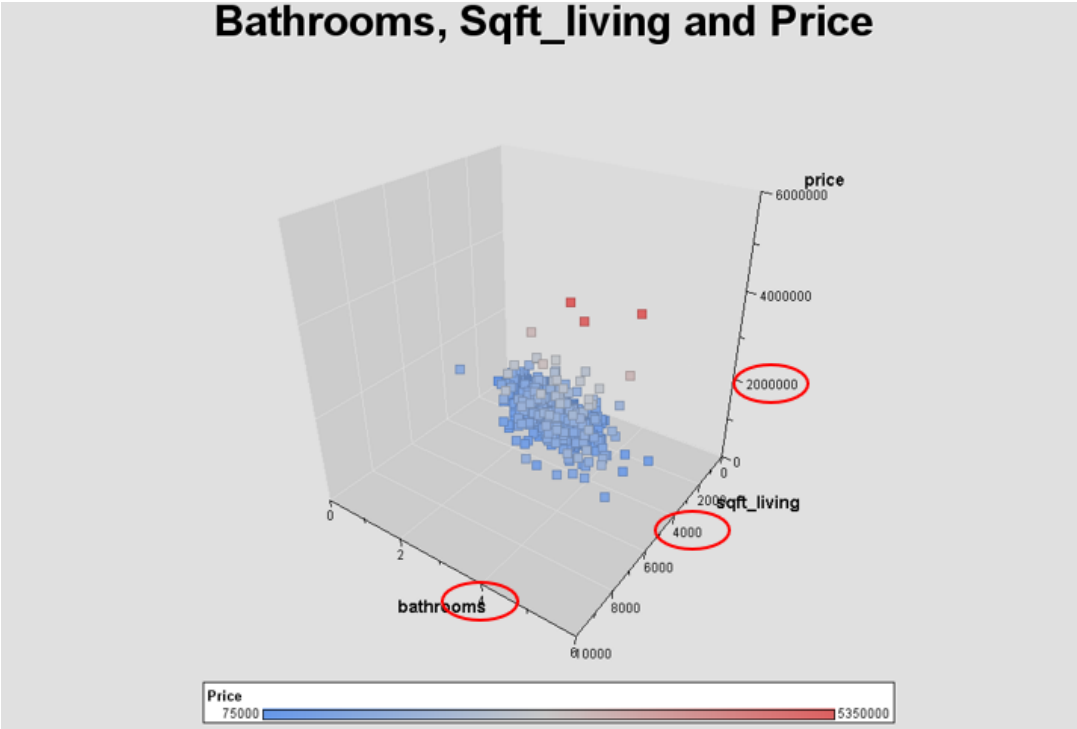


Figure 5.21 Bathrooms, Sqft_living and Price

(1) What does the distribution look like for Bathrooms and Sqft_living by Price?

Figure 5.21 shows a scatterplot of the number of bathrooms (bathrooms), the square footage of the apartment's interior living space (stationing), and the price of each house sold. Based on the scatterplot, we can observe that most of the data points are concentrated in the center, showing most of the price of each house sold is less than $2,000,000, acquiring less than 4 bathrooms and less than 4000 square feet of interior living space. It is also clearly shown that there are some outliers at extreme house prices.

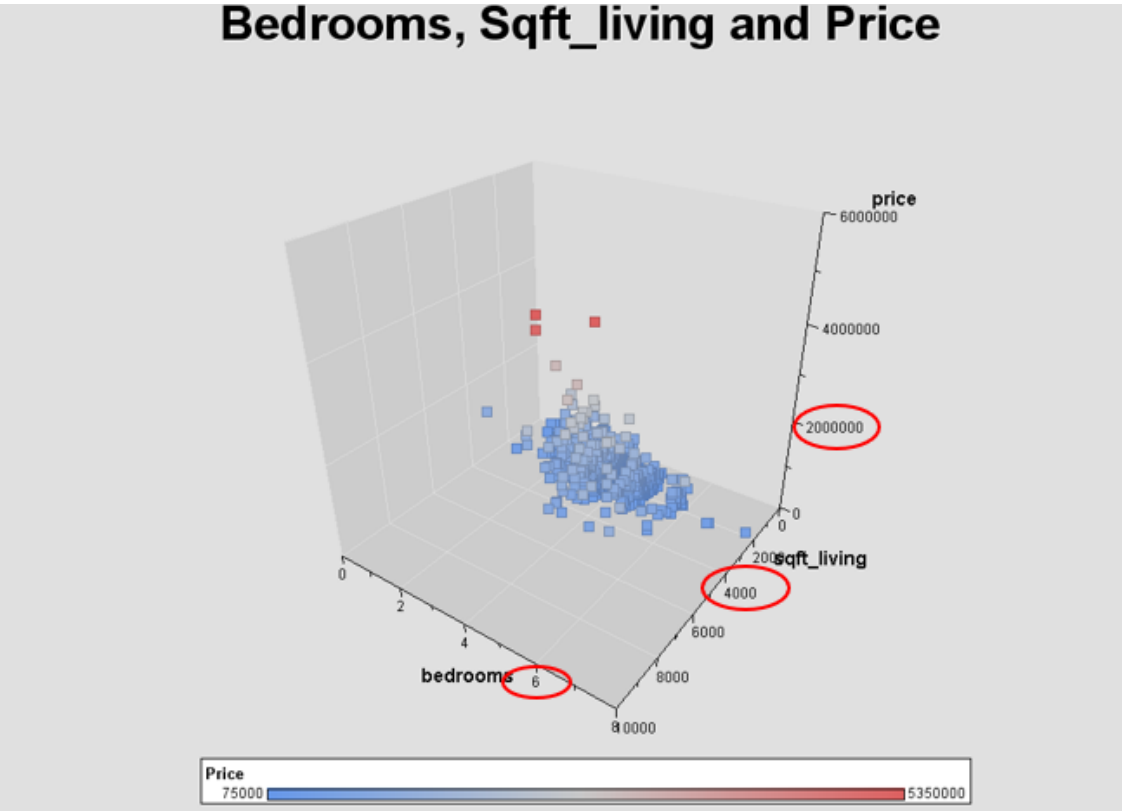


Figure 5.22 Bedrooms, Sqft_living and Price

(2) What does the distribution look like for Bedrooms and Sqft_living by Price?

Figure 5.22 shows a scatterplot of the number of bedrooms (bedrooms), the square footage of the apartment's interior living space (sqft_living), and the price of each house sold. Based on the scatterplot, we can observe that most of the data points are concentrated in the center same as Figure 5.21. This clearly shows that most of the price of each house sold is also less than $2,000,000, acquiring less than 6 bedrooms and less than 4000 square feet of interior living space. It is also clearly shown that there are some outliers at extreme house prices and number of bedrooms.

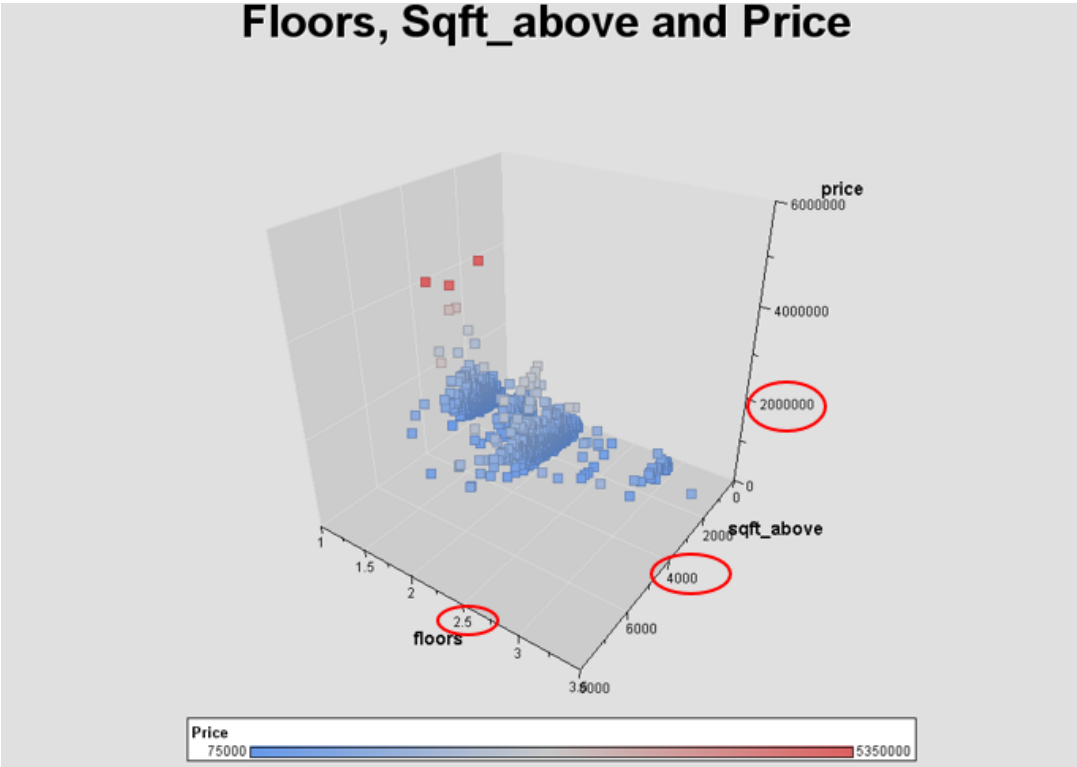


Figure 5.23 Floors, Sqft_above and Price

(3) What does the distribution look like for Floors and Sqft_above by Price?

Figure 5.23 shows a scatterplot of the number of bedrooms (bedrooms), the square footage of the interior housing space that is above ground level (sqft_above), and the price of each house sold. Based on the scatterplot, we can observe that most of the data points are too concentrated in the center with each house mostly sold equal or less than $2,000,000, acquiring less than 2.5 floors and less than 4000 square feet of interior house space. It is also clearly shown that there are some outliers at extreme house prices ($4,000,000 and above) having only less than 1.5 floors.

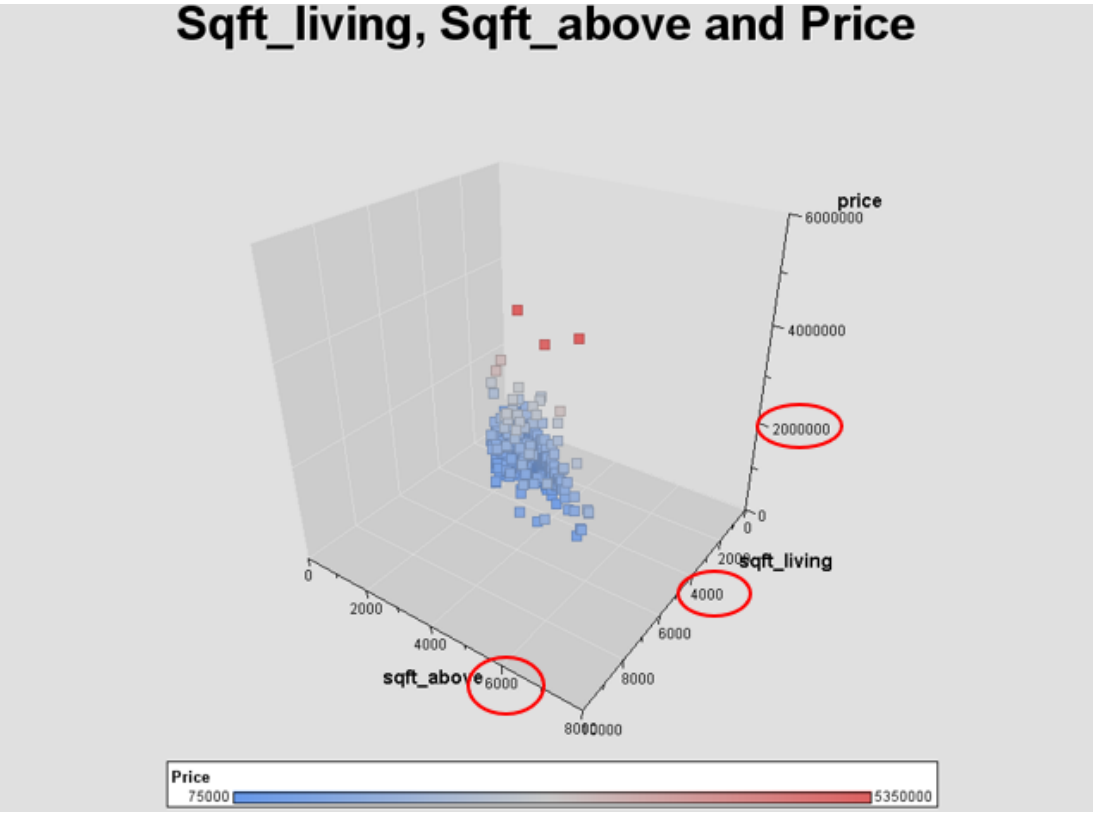


Figure 5.24 Sqft_living, Sqft_above and Price

(4) What does the distribution look like for Sqft_living and Sqft_above by Price?

Figure 5.24 shows a scatterplot of the square footage of the apartment's interior living space (sqft_living), the square footage of the interior housing space that is above ground level (sqft_above), and the price of each house sold. Based on the scatterplot, we can observe that most of the data points are too concentrated in the center with each house mostly sold around $2,000,000, acquiring less than 4000 square feet of interior living space and less than 6000 square feet of interior house space above ground level. Outliers are also identified at extreme house prices ($4,000,000).

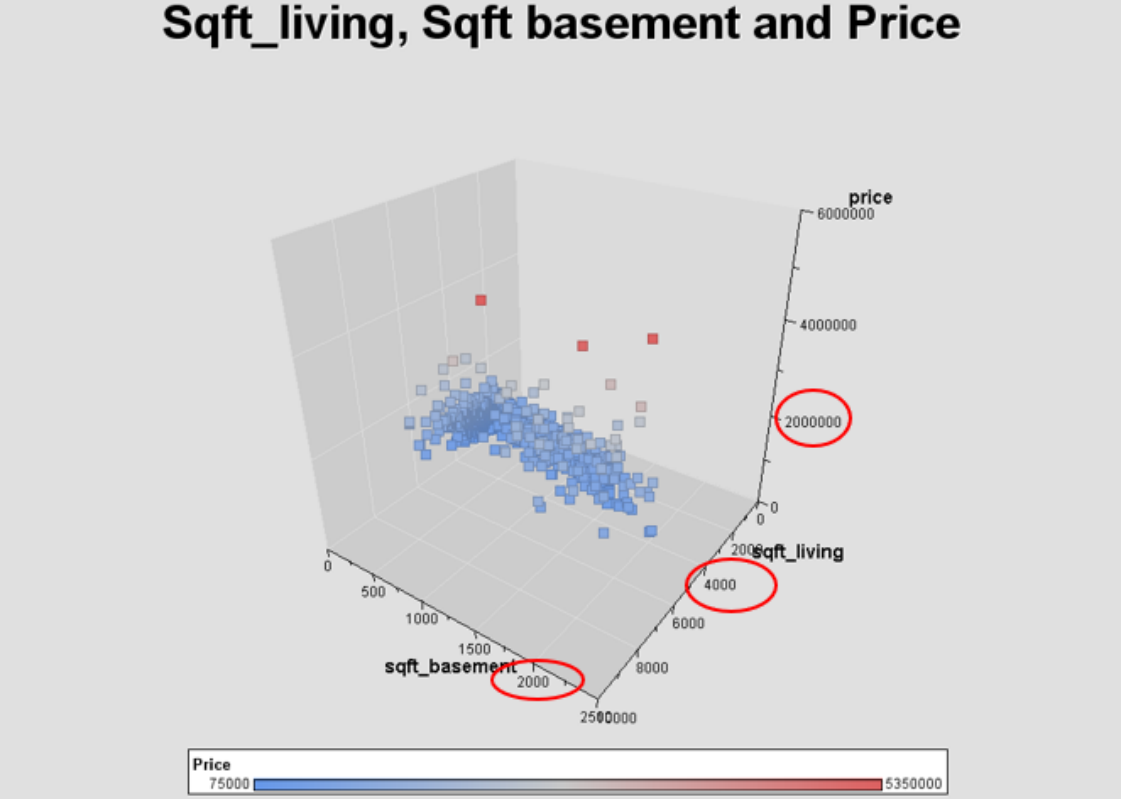


Figure 5.25 Sqft_living, Sqft_basement and Price

(5) What does the distribution look like for Sqft_living and Sqft_basement by Price?

Figure 5.25 shows a scatterplot of the square footage of the apartment's interior living space (sqft_living), the square footage of the interior housing space that is below ground level (sqft_basement), and the price of each house sold. Based on the scatterplot, we can observe that most of the data points are evenly distributed in contrast with previous figures. Each house mostly sold around $2,000,000, acquiring less than 4000 square feet of interior living space and 2000 square feet of interior house basement.

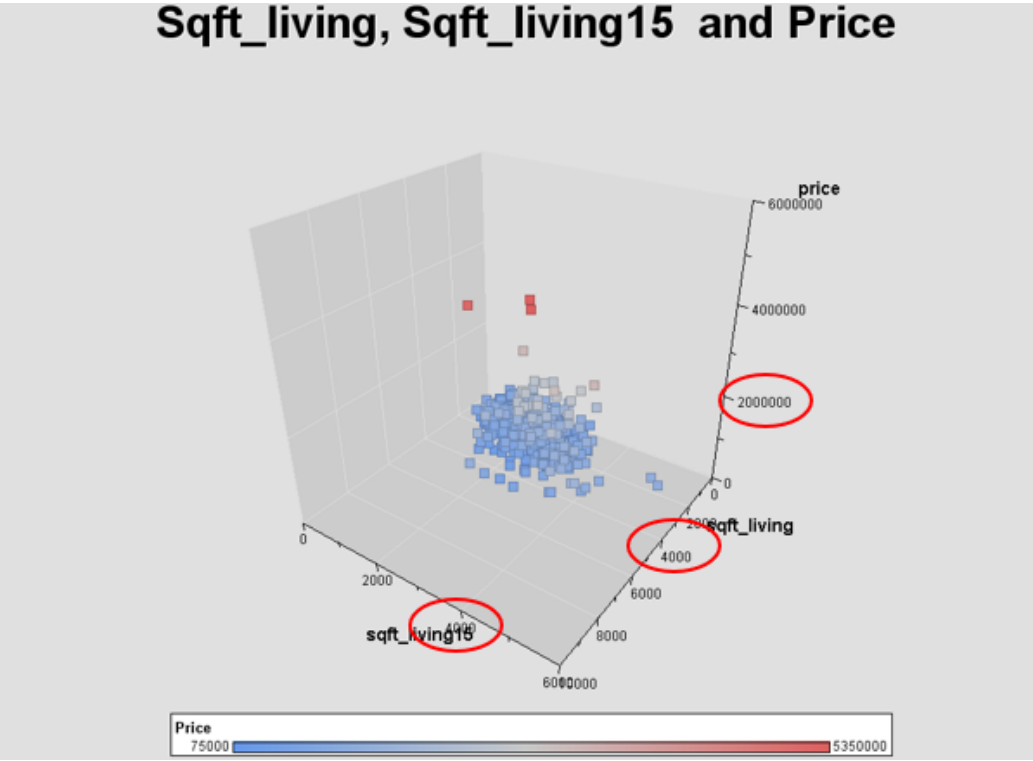


Figure 5.26 Sqft_living, Sqft_living15 and Price

(6) What does the distribution look like for Sqft_living and Sqft_ living15 by Price?

Figure 5.26 shows a scatterplot of the square footage of the apartment's interior living space (sqft_living), the square footage of the interior housing living space for the nearest 15 neighbours and the price of each house sold. Based on the scatterplot, we can observe that most customers prefer purchasing houses for around $2,000,000, acquiring less than 4000 square feet of interior living space and 4000 square feet for the nearest 15 neighbours. Outliers are also detected like the previous figures.

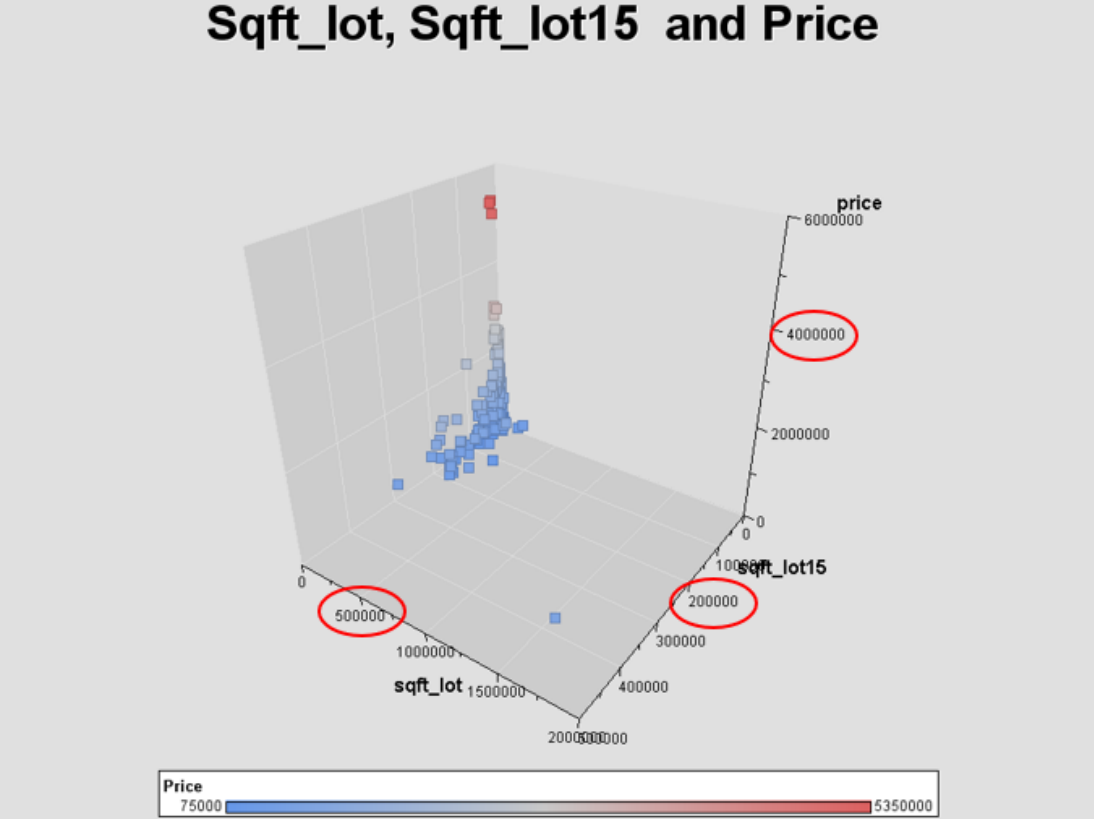


Figure 5.27 Sqft_lot, Sqft_lot15 and Price

(7) What does the distribution look like for Sqft_lot and Sqft_lot15 by Price?

Figure 5.27 shows a scatterplot of the square footage of the land space (sqft_lot), the square footage of the land lots of the nearest 15 neighbours, and the price of each house sold. Based on the scatterplot, we can observe that most customers prefer purchasing houses less than $4,000,000, acquiring less than 500000 square feet of land and 200000 square feet for the nearest 15 neighbors. There are still a few customers who still prefer to buy $6,000,000 during their stay in King County.

5.3 MODIFY – Data Modification

In this phase, we will alter, clean, reduce, and transform the data after we have explored it. This stage is crucial for further data modelling and has a direct impact on the precision of the prediction model.

5.3.1 Modifying and Correcting Source Data

To perform classification, we have added an attribute based on the price named price_range. We have taken the median of the price attributes as our baseline, if the value of the price is over the median the price range is 1, if lower or equal to the median the value of price_range is 0. And we have set the price_range attributes as target.

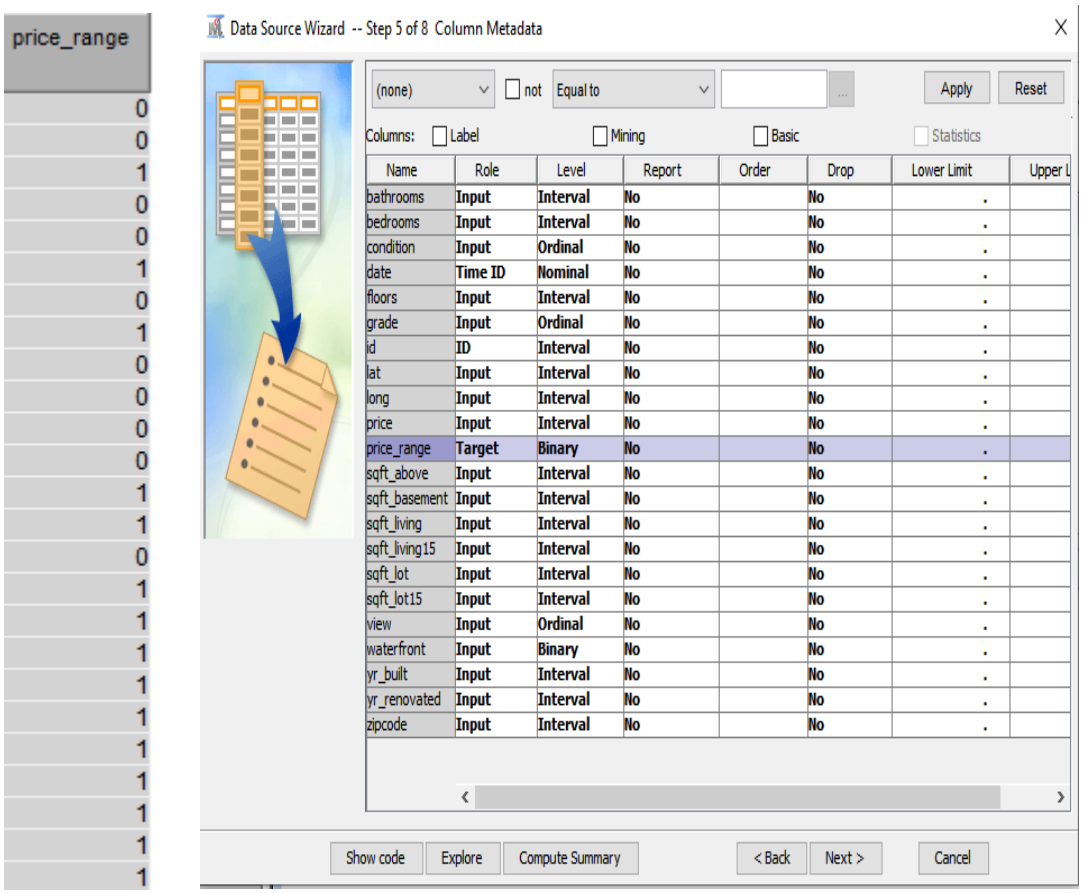


Figure 5.28: Create price_range variable

During data exploration, we found incomplete data, noisy data, and inconsistent data. These variables are as follows:

Variable	Error Type
bathrooms	Incomplete
yr_built	Inconsistent
bathrooms bedrooms sqft_above price lat long sqft_living15 sqft_lot15 sqft_living sqft_lot sqft_basement	Noisy

Next, we will modify these error type. To modify inconsistent data, we will use Talend, which is open-source data integration platform which provides various software and services for data integration, data management, enterprise application integration, data quality, cloud storage and Big Data. In the data exploration, we found the following inconsistent errors in our yr_built attributes. And we replace them with the correct values.

Inconsistent yr_built value	After replacement
192102	1921
19570522	1957
190810	1908
19310401	1931
192703	1927
19590731	1959
191006	1910

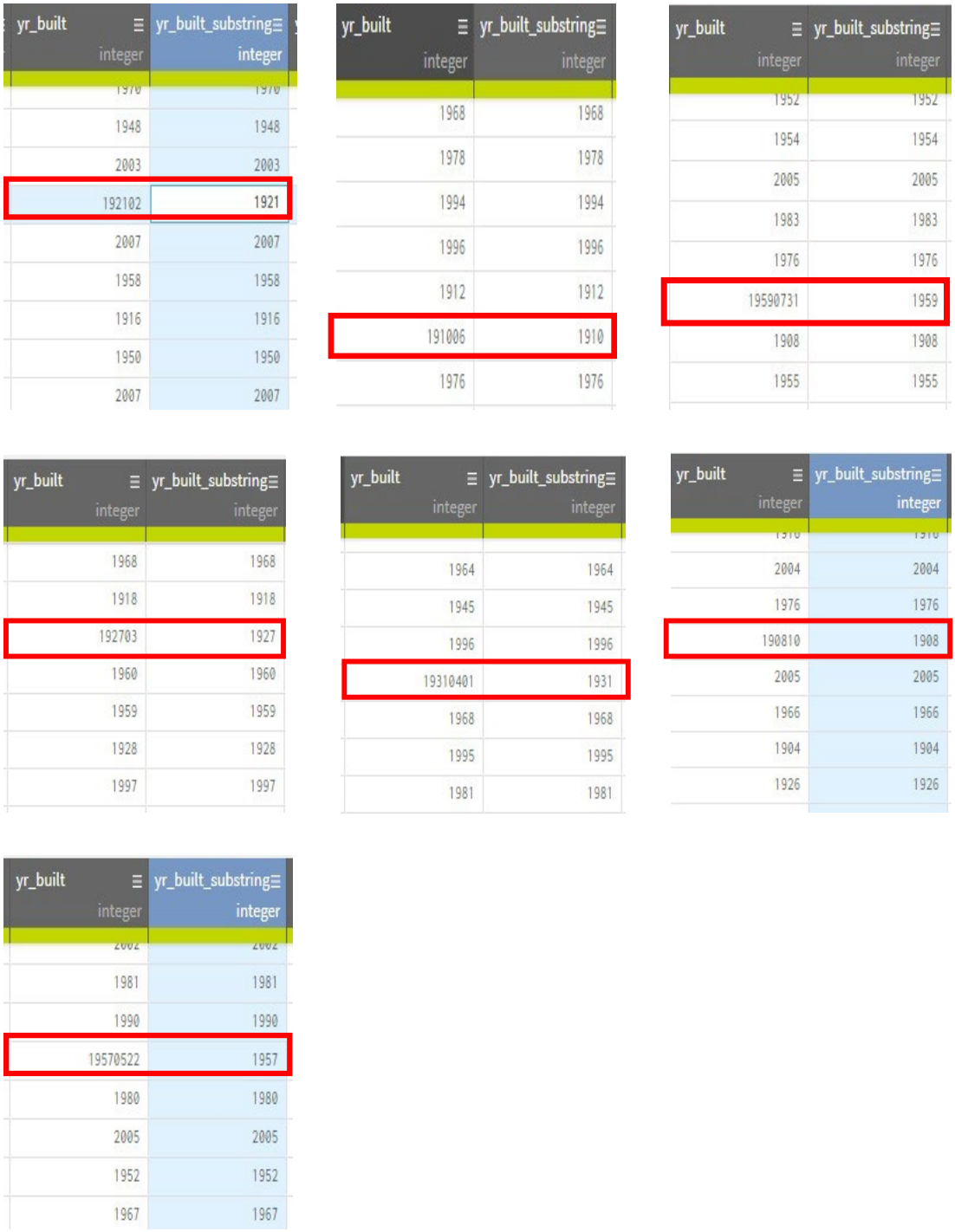


Figure 5.29 Replace inconsistent data with correct values

Next, we modify the noise data and missing data. To modify the noise data and missing data, we will use SAS Enterprise Miner. Limited methods are followed, and we replace these values with missing values.

Variable	Error Type	Limits method
bathrooms bedrooms sqft_above price lat long sqft_living15 sqft_lot15 sqft_living sqft_lot sqft_basement	Noisy	Extreme Percentiles
bathrooms	Incomplete	Mean

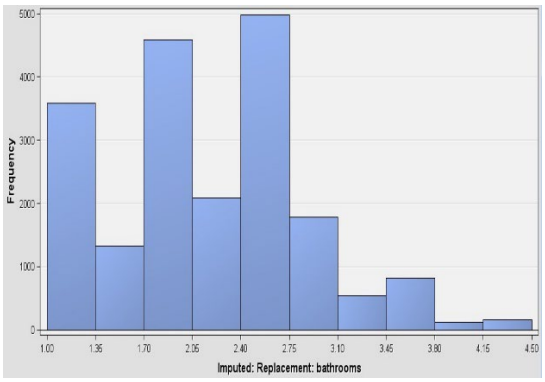
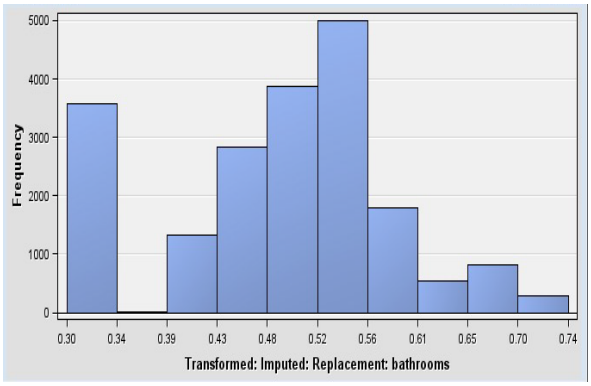
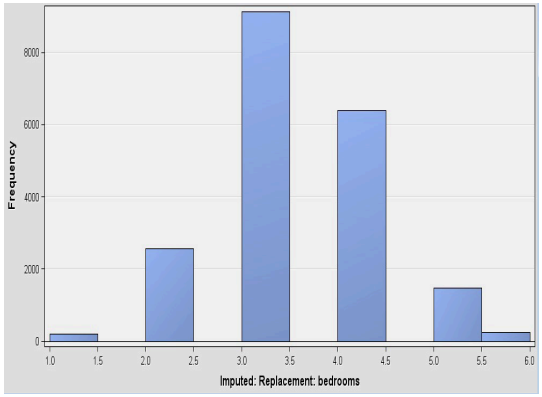
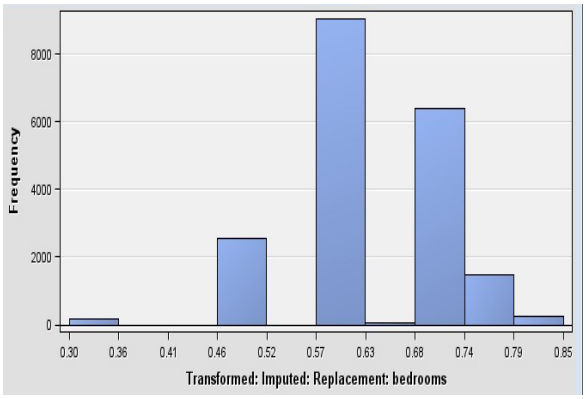
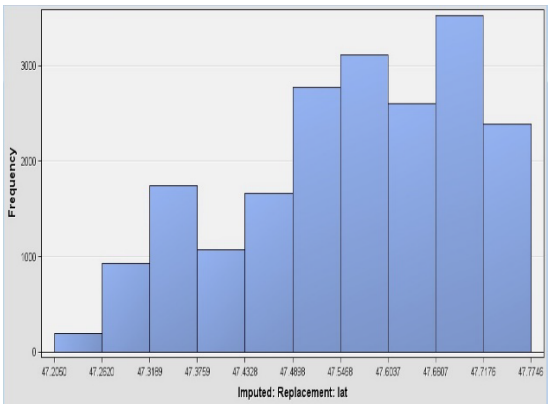
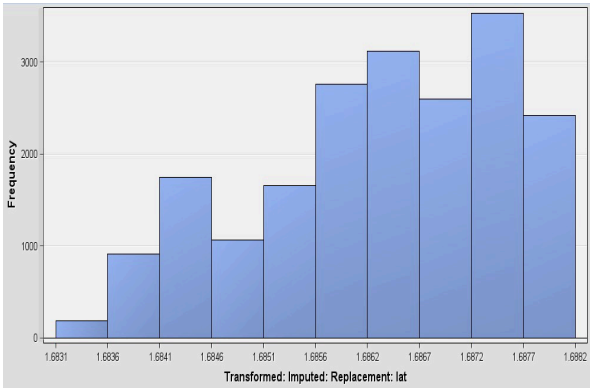
We padded incomplete and missing data with mean. The result is follows:

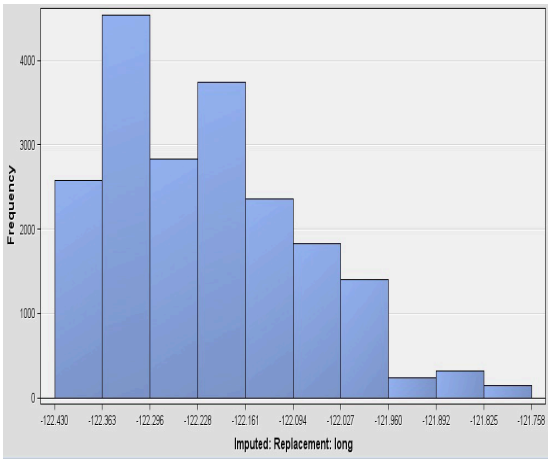
Variable Name	Impute Method	Imputed Variable	Impute Value	Role	Measurement Level	Label	Number of Missing for TRAIN
REP_bathrooms	MEAN	IMP_REP_bathrooms	2.107488	INPUT	INTERVAL	Replacement: bathr...	178
REP_bedrooms	MEAN	IMP_REP_bedrooms	3.359458	INPUT	INTERVAL	Replacement: bedro...	75
REP_lat	MEAN	IMP_REP_lat	47.56079	INPUT	INTERVAL	Replacement: lat	215
REP_long	MEAN	IMP_REP_long	-122.215	INPUT	INTERVAL	Replacement: long	214
REP_price	MEAN	IMP_REP_price	528856.8	INPUT	INTERVAL	Replacement: price	216
REP_sqft_above	MEAN	IMP_REP_sqft_above	1774.22	INPUT	INTERVAL	Replacement: sqft_...	201
REP_sqft_basement	MEAN	IMP_REP_sqft_bas...	282.1509	INPUT	INTERVAL	Replacement: sqft_...	103
REP_sqft_living	MEAN	IMP_REP_sqft_living	2063.344	INPUT	INTERVAL	Replacement: sqft_li...	200
REP_sqft_living15	MEAN	IMP_REP_sqft_livin...	1978.331	INPUT	INTERVAL	Replacement: sqft_li...	213
REP_sqft_lot	MEAN	IMP_REP_sqft_lot	13092.99	INPUT	INTERVAL	Replacement: sqft_lot	214
REP_sqft_lot15	MEAN	IMP_REP_sqft_lot15	11534.16	INPUT	INTERVAL	Replacement: sqft_l...	213

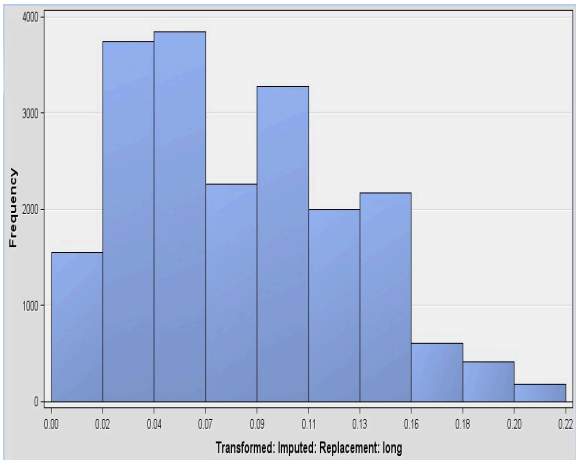
Figure 5.30 Imputation Summary

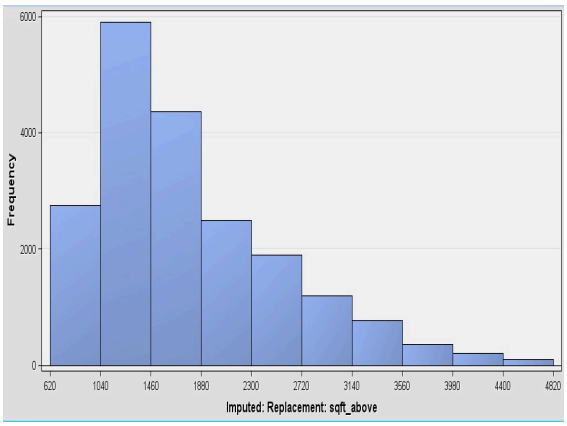
We will modify the variables and apply normalisation after substituting the data. The advantage of normalisation is that it lessens skewness. This step is advantageous for machine learning algorithms that assume the feature variable has a normal distribution, lowering the level of measurement while keeping the ratio constant. This is another advantage of normalisation because it enhances the model's training efficiency.

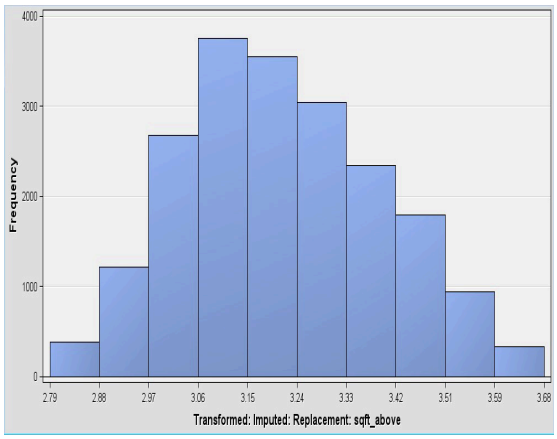
All the variables that we used for the log 10 transformation normalisation approach are listed in the table below.

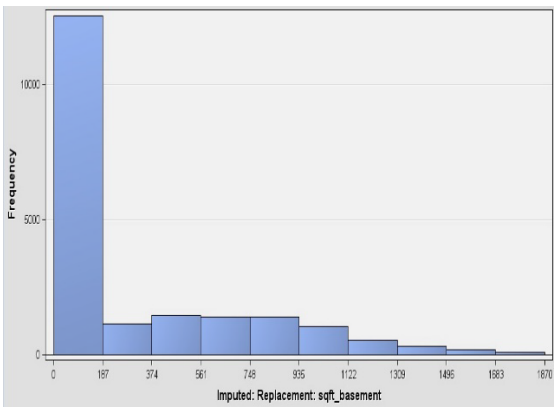
Before Transformation	After Transformation
	
	
	

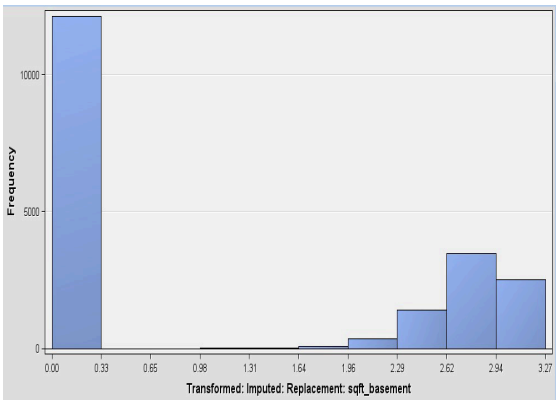

Frequency
4000
3000
2000
1000
0
-122.430
-122.363
-122.296
-122.228
-122.161
-122.094
-122.027
-121.960
-121.892
-121.825
-121.758
Imputed: Replacement: long

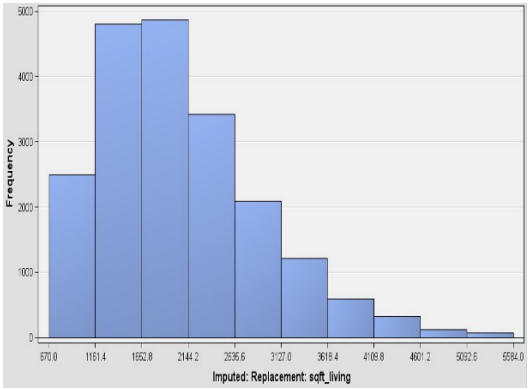

Frequency
4000
3000
2000
1000
0
0.00
0.02
0.04
0.07
0.09
0.11
0.13
0.16
0.18
0.20
0.22
Transformed: Imputed: Replacement: long

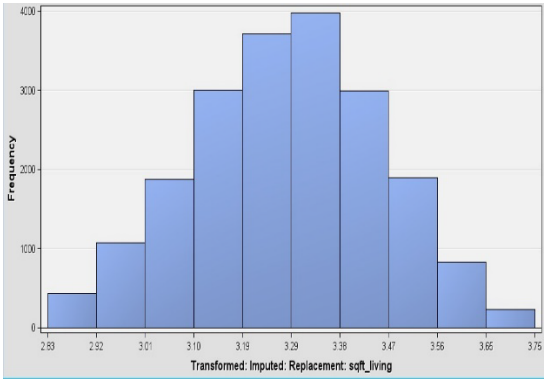

Frequency
6000
4000
2000
0
620
1040
1460
1880
2300
2720
3140
3560
3980
4400
4820
Imputed: Replacement: sqft_above

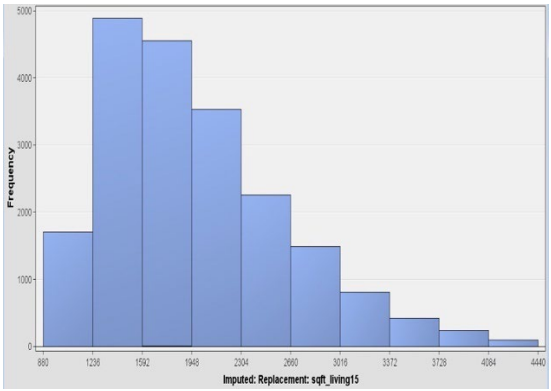

Frequency
4000
3000
2000
1000
0
2.79
2.88
2.97
3.06
3.15
3.24
3.33
3.42
3.51
3.59
3.68
Transformed: Imputed: Replacement: sqft_above

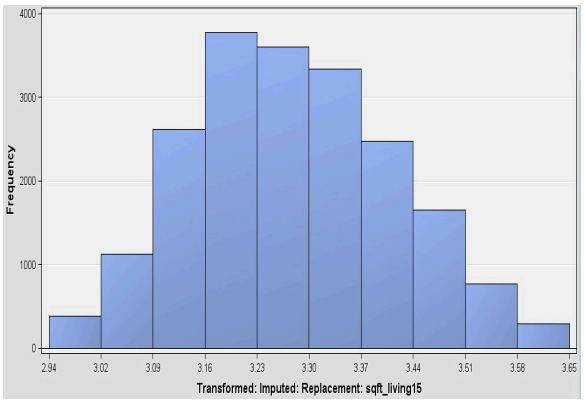

Frequency
10000
5000
0
0
187
374
561
748
935
1122
1309
1496
1683
1870
Imputed: Replacement: sqft_basement

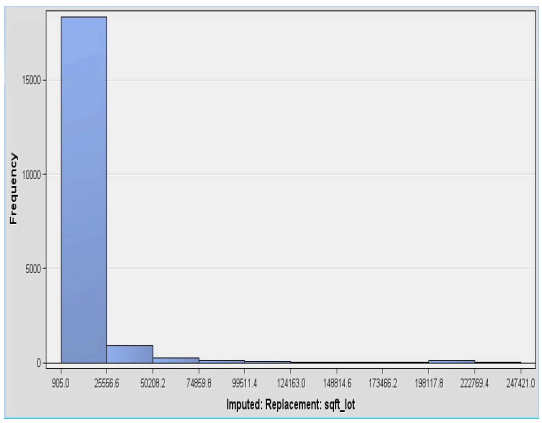

Frequency
10000
5000
0
0.00
0.33
0.65
0.98
1.31
1.64
1.96
2.29
2.62
2.94
3.27
Transformed: Imputed: Replacement: sqft_basement

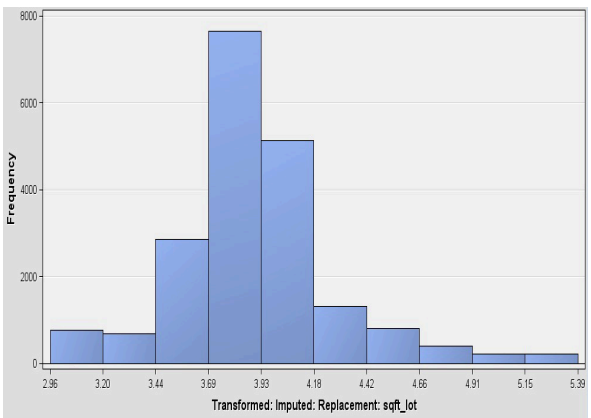

Frequency
5000
4000
3000
2000
1000
0
670.0
1161.4
1652.8
2144.2
2635.6
3127.0
3618.4
4109.8
4601.2
5092.6
5584.0
Imputed: Replacement: sqft_living


Frequency
4000
3000
2000
1000
0
2.83
2.92
3.01
3.10
3.19
3.29
3.38
3.47
3.56
3.65
3.75
Transformed: Imputed: Replacement: sqft_living


Frequency
5000
4000
3000
2000
1000
0
880
1236
1592
1948
2304
2660
3016
3372
3728
4084
4440
Imputed: Replacement: sqft_living15


Frequency
4000
3000
2000
1000
0
2.94
3.02
3.09
3.16
3.23
3.30
3.37
3.44
3.51
3.58
3.65
Transformed: Imputed: Replacement: sqft_living15


Frequency
15000
10000
5000
0
905.0
25556.6
50208.2
74859.8
99511.4
124163.0
148814.6
173466.2
198117.8
222769.4
247421.0
Imputed: Replacement: sqft_lot


Frequency
8000
6000
4000
2000
0
2.96
3.20
3.44
3.69
3.93
4.18
4.42
4.66
4.91
5.15
5.39
Transformed: Imputed: Replacement: sqft_lot

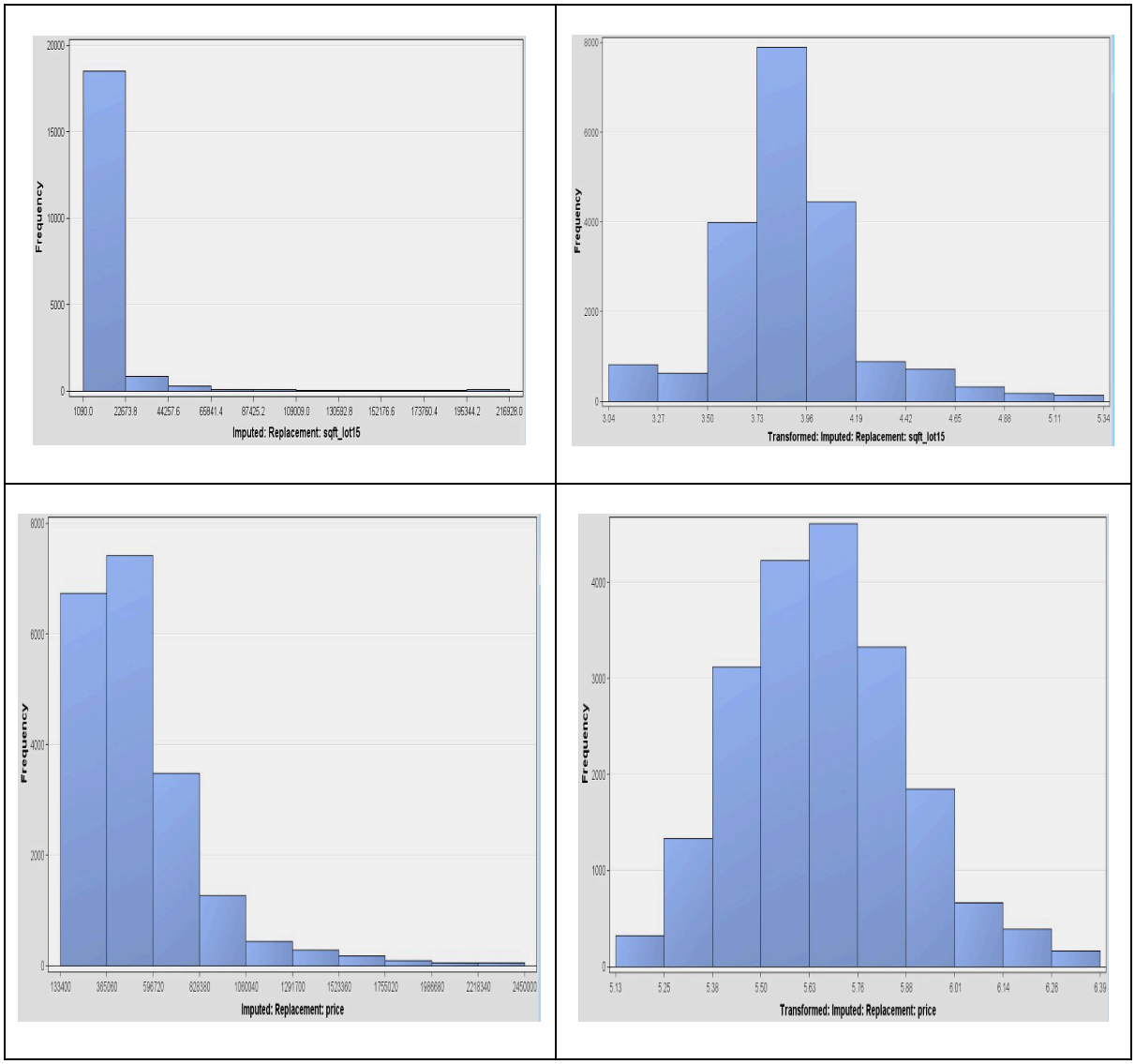


Figure 5.31 Transformation Data

5.3.2 Examining Exported Data

In the previous section, we identify that there is an incomplete data error in the bathrooms variable. The following is the histogram of the bathrooms variable. We can see that the missing bin is 0. Therefore, the issue is fixed.

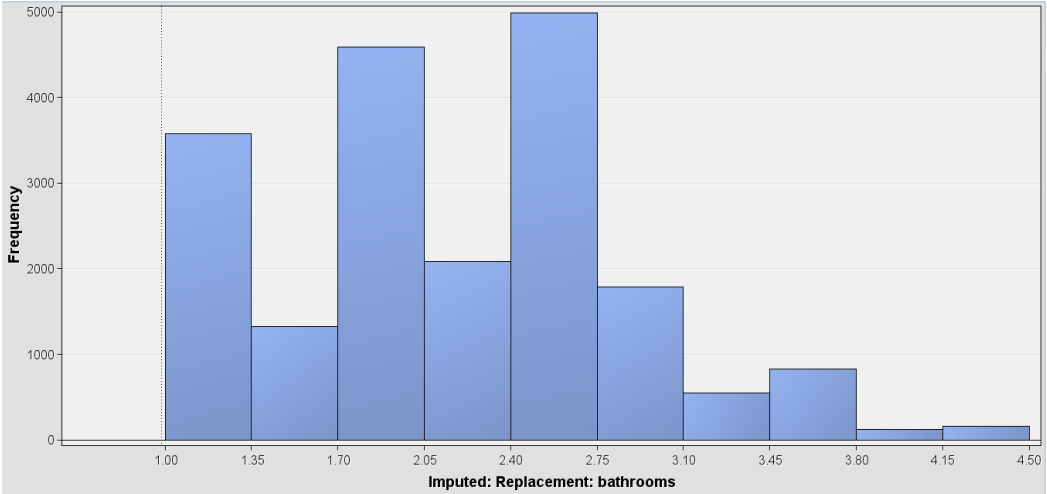


Figure 5.32 Histogram of bathrooms

In the previous section, we identify there is an inconsistent data in the yr_built variable. The following is the histogram of the yr_built variable. We can see that now there is no value that has month and date. Therefore, the issue is fixed.

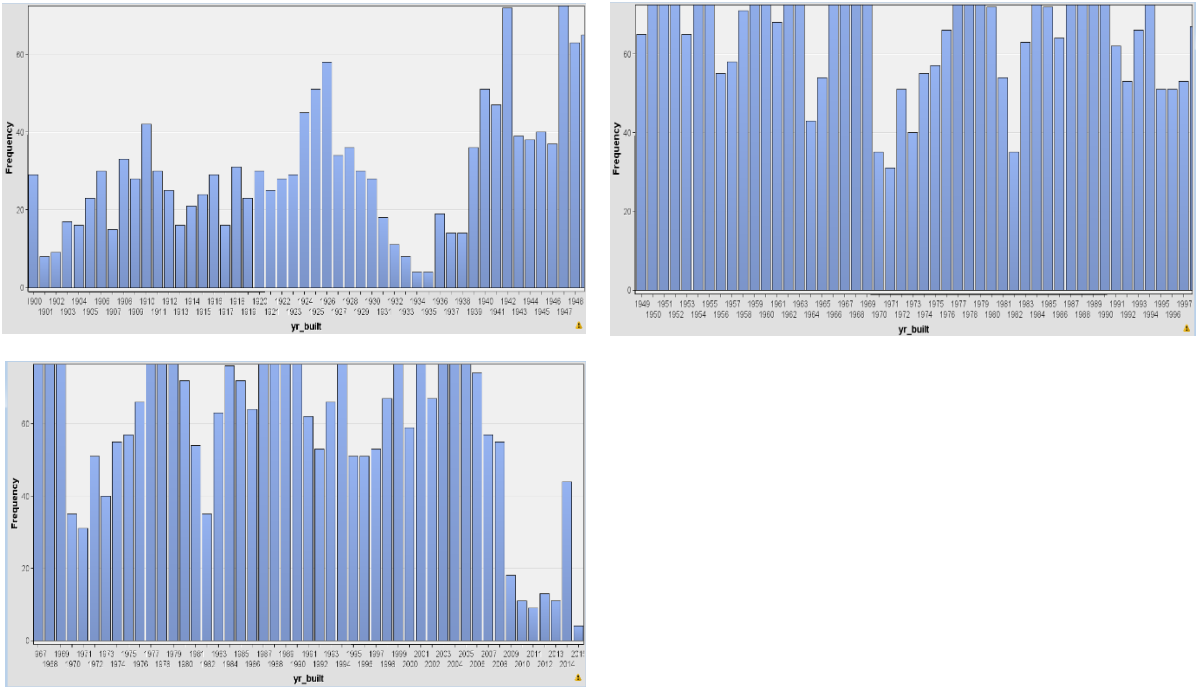


Figure 5.33 Histogram of yr_built

In the previous section, we identify that there are noisy outliers error in bathrooms variable, bedrooms variable, sqft_above variable, price variable, lat variable, long variable, sqft_living15 variable, sqft_lot15 variable, sqft_living variable, sqft_lot variable and sqft_basement variable. The following is the boxplot of the abovementioned variable. We observed that after replacement and imputation there are still some outliers in the box plot. To maintain the originality of the dataset to prevent overfitting, we decide to keep the remaining outliers.

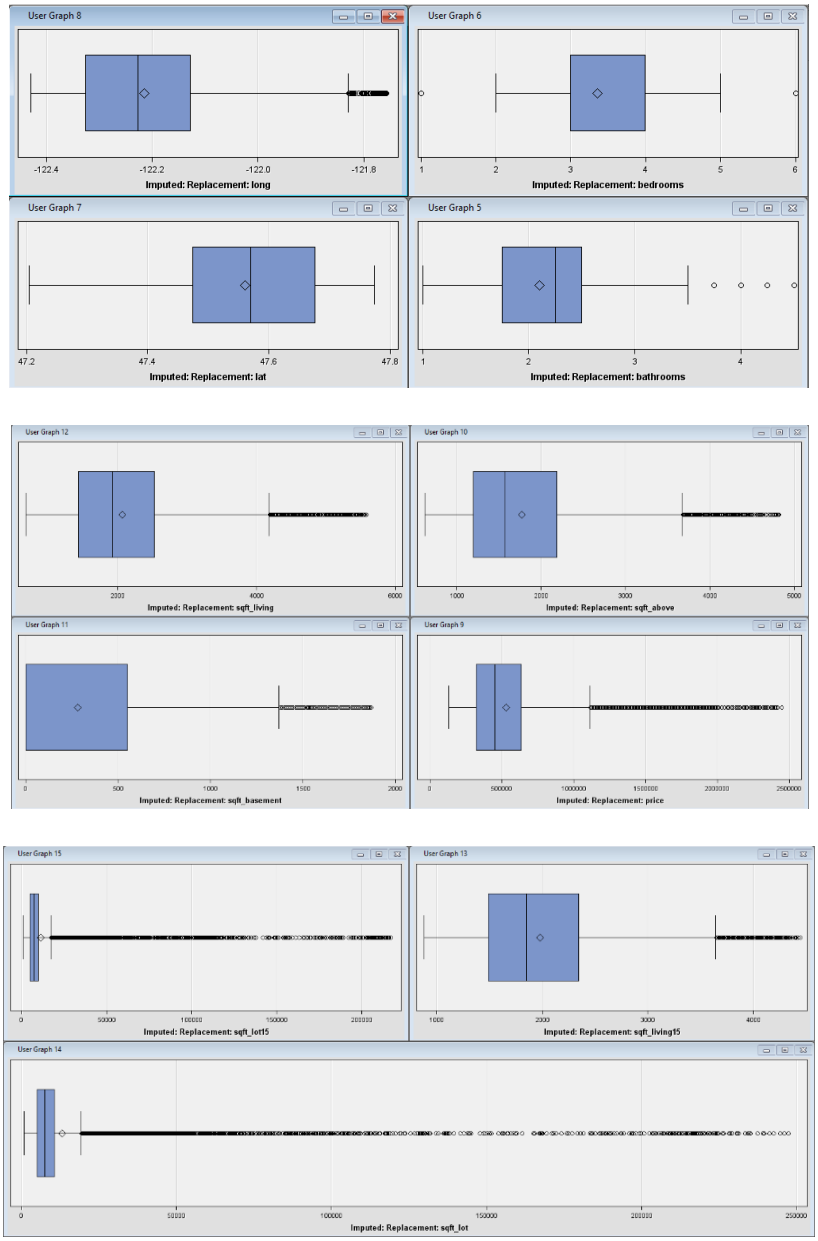


Figure 5.34 Boxplot

5.3.3 Creating Training and Validation Data

By using the Data Partition node, we are able to split the dataset to create Training and Validation sets. The split ratio for this project is 50:50 on Training and Validation as shown on the figure below:

Data Set Allocations	
Training	50.0
Validation	50.0
Test	0.0

Figure 5.35 Data Set Allocations

Output

```
*------------------------------------------------------------*
* Report Output
*------------------------------------------------------------*



Summary Statistics for Class Targets

Data=DATA

                Numeric     Formatted     Frequency
 Variable        Value        Value         Count      Percent    Label

price_range         0            0         10864      50.2660
price_range         1            1         10749      49.7340


Data=TRAIN

                Numeric     Formatted     Frequency
 Variable        Value        Value         Count      Percent    Label

price_range         0            0          5431      50.2591
price_range         1            1          5375      49.7409


Data=VALIDATE

                Numeric     Formatted     Frequency
 Variable        Value        Value         Count      Percent    Label

price_range         0            0          5433      50.2730
price_range         1            1          5374      49.7270

```

Figure 5.36 Report Output for Data Partition node

Based on our analysis previously, we would then focus on specific variables and drop variables that are insignificant to our modelling. We would keep LG10_IMP_REP_bedrooms, LG10_IMP_REP_lat, LG10_IMP_REP_living, LG10_IMP_REP_lot15, condition, grade, price_range, view, waterfront and yr_renovated.

Name	Drop	Role	Level
LG10_IMP_REP_bathrooms	Yes	Input	Interval
LG10_IMP_REP_bedrooms	No	Input	Interval
LG10_IMP_REP_lat	No	Input	Interval
LG10_IMP_REP_long	Yes	Input	Interval
LG10_IMP_REP_price	Yes	Input	Interval
LG10_IMP_REP_sqft_above	Yes	Input	Interval
LG10_IMP_REP_sqft_basement	Yes	Input	Interval
LG10_IMP_REP_sqft_living	No	Input	Interval
LG10_IMP_REP_sqft_living15	Yes	Input	Interval
LG10_IMP_REP_sqft_lot	Yes	Input	Interval
LG10_IMP_REP_sqft_lot15	No	Input	Interval
condition	No	Input	Ordinal
date	Yes	Time ID	Nominal
floors	Yes	Input	Interval
grade	No	Input	Ordinal
id	Yes	ID	Interval
price_range	No	Target	Binary
view	No	Input	Ordinal
waterfront	No	Input	Binary
yr_built	Yes	Input	Interval
yr_renovated	No	Input	Interval
zipcode	Yes	Input	Interval

Figure 5.3.7 Drop Variables

5.4 MODEL – Data Modelling

After the data are split into training and validation set with a ratio of 50:50, the training data is used to build several models. In this study, we build a decision tree, gradient boosting, logistic regression, and neural network models to predict house prices.

5.4.1 Constructing a Decision Tree Predictive Model

Firstly, a decision tree predictive model is built. A decision tree is a supervised model, and it has a tree-based structure that consists of a root node, internal nodes, branches, and leaf nodes. In each node, decision rules are generated to divide the targets to solve the classification task. The decision tree model is preferred as it is simple, quick, and able to manage a high volume of data as we have 21,613 rows in this study.

The decision tree is built as shown below.

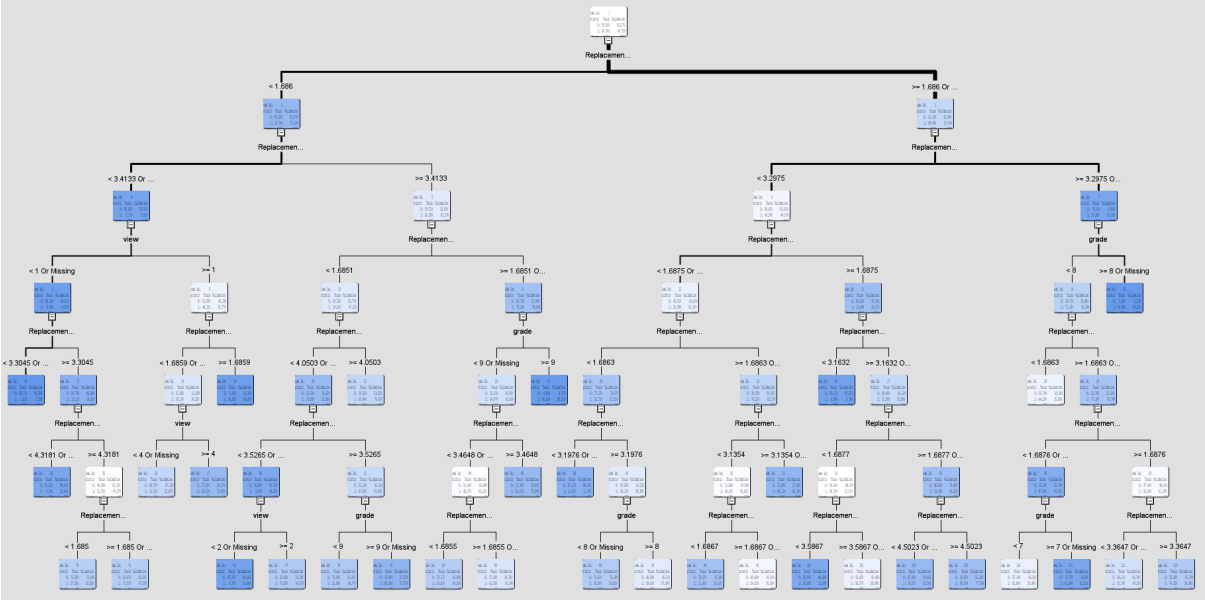


Figure 5.4.1.1: Decision Tree

Based on the decision tree developed, we found several interesting points where (1) latitude has a high level of information gain as it is selected to be the root node and appears in internal nodes several times as it is able to split the data into classes effectively, (2) the shortest split is after 3 layers of rules and (3) at most situations when the grade is high, the price range is high. The list of decision rules is as below:

- If lat < 1.686, sqft_living < 3.4133 Or Missing, view < 1 Or Missing and sqft_living < 3.3045 Or Missing, then price_range =low.

- If lat < 1.686, sqft_living < 3.4133 Or Missing, view < 1 Or Missing, sqft_living >= 3.3045 and sqft_lot15 < 4.3181 Or Missing, then price_range =low.
- If lat < 1.686, sqft_living < 3.4133 Or Missing, view < 1 Or Missing, sqft_living >= 3.3045, sqft_lot15 >= 4.3181 and lat < 1.685, then price_range =low.
- If lat < 1.686, sqft_living < 3.4133 Or Missing, view < 1 Or Missing, sqft_living >= 3.3045, sqft_lot15 >= 4.3181 and lat >= 1.685, then price_range =high.
- If lat < 1.686, sqft_living < 3.4133 Or Missing, view >= 1 and lat >= 1.6859, then price_range =high.
- If lat < 1.686, sqft_living < 3.4133 Or Missing, view >= 1, lat < 1.6859 Or Missing and view < 4 Or Missing, then price_range =low.
- If lat < 1.686, sqft_living < 3.4133 Or Missing, view >= 1, lat < 1.6859 Or Missing and view >=4 Or Missing, then price_range =high.
- If lat < 1.686, sqft_living >= 3.4133, lat < 1.6851, sqft_lot15 < 4.0503 Or Missing, sqft_living < 3.5265 Or Missing and view < 2 Or Missing, then price_range =low.
- If lat < 1.686, sqft_living >= 3.4133, lat < 1.6851, sqft_lot15 < 4.0503 Or Missing, sqft_living < 3.5265 Or Missing and view>= 2 Or Missing, then price_range =high.
- If lat < 1.686, sqft_living >= 3.4133, lat < 1.6851 and sqft_lot15 >= 4.0503, then price_range =high.
- If lat < 1.686, sqft_living >= 3.4133, lat < 1.6851, sqft_lot15 < 4.0503 Or Missing, sqft_living >= 3.5265 and grade < 9, then price_range =low.
- If lat < 1.686, sqft_living >= 3.4133, lat < 1.6851, sqft_lot15 < 4.0503 Or Missing, sqft_living >= 3.5265 and grade >= 9, then price_range =high.
- If lat < 1.686, sqft_living >= 3.4133, lat >= 1.6851 Or Missing and grade >= 9, then price_range =high.
- If lat < 1.686, sqft_living >= 3.4133, lat >= 1.6851 Or Missing, grade < 9 Or Missing and sqft_living >= 3.4648, then price_range =high.
- If lat < 1.686, sqft_living >= 3.4133, lat >= 1.6851 Or Missing, grade < 9 Or Missing, sqft_living < 3.4648 Or Missing and lat < 1.6855, then price_range =low.
- If lat < 1.686, sqft_living >= 3.4133, lat >= 1.6851 Or Missing, grade < 9 Or Missing, sqft_living < 3.4648 Or Missing and lat >= 1.6855, then price_range =high.
- If lat >= 1.686 Or Missing, sqft_living < 3.2975, lat < 1.6875 Or Missing, lat < 1.6863 and sqft_living < 3.1976 Or Missing, then price_range =low.

- If lat >= 1.686 Or Missing, sqft_living < 3.2975, lat < 1.6875 Or Missing, lat < 1.6863, sqft_living >= 3.1976 and grade < 8 Or Missing, then price_range =low.
- If lat >= 1.686 Or Missing, sqft_living < 3.2975, lat < 1.6875 Or Missing, lat < 1.6863, sqft_living >= 3.1976 and grade >=8 Or Missing, then price_range =high.
- If lat >= 1.686 Or Missing, sqft_living < 3.2975, lat < 1.6875 Or Missing, lat >= 1.6863 Or Missing and sqft_living >= 3.1354 Or Missing, then price_range =high.
- If lat >= 1.686 Or Missing, sqft_living < 3.2975, lat < 1.6875 Or Missing, lat >= 1.6863 Or Missing, sqft_living < 3.1354 and lat < 1.6867, then price_range =low.
- If lat >= 1.686 Or Missing, sqft_living < 3.2975, lat < 1.6875 Or Missing, lat >= 1.6863 Or Missing, sqft_living < 3.1354 and lat >= 1.6867, then price_range =high.
- If lat >= 1.686 Or Missing, sqft_living < 3.2975, lat >= 1.6875 and sqft_living < 3.1632, then price_range =low.
- If lat >= 1.686 Or Missing, sqft_living < 3.2975, lat >= 1.6875, sqft_living >= 3.1632 Or Missing, lat < 1.6877 and sqft_lot15 < 3.5867, then price_range =low.
- If lat >= 1.686 Or Missing, sqft_living < 3.2975, lat >= 1.6875, sqft_living >= 3.1632 Or Missing, lat < 1.6877 and sqft_lot15 >=3.5867, then price_range =high.
- If lat >= 1.686 Or Missing, sqft_living < 3.2975, lat >= 1.6875, sqft_living >= 3.1632 Or Missing, lat >= 1.6877 Or Missing and sqft_lot15 < 4.5023 Or Missing, then price_range =low.
- If lat >= 1.686 Or Missing, sqft_living < 3.2975, lat >= 1.6875, sqft_living >= 3.1632 Or Missing, lat >= 1.6877 Or Missing and sqft_lot15 >= 4.5023 Or Missing, then price_range =high.
- If lat >= 1.686 Or Missing, sqft_living >= 3.2975 and grade >= 8 Or Missing, then price_range =high.
- If lat >= 1.686 Or Missing, sqft_living >= 3.2975 Or Missing, grade < 8 and lat < 1.6863, then price_range =low.
- If lat >= 1.686 Or Missing, sqft_living >= 3.2975, grade < 8, lat >= 1.6863, lat < 1.6876 Or Missing and grade < 7, then price_range =low.
- If lat >= 1.686 Or Missing, sqft_living >= 3.2975, grade < 8, lat >= 1.6863, lat < 1.6876 Or Missing and grade >=7, then price_range =high.
- If lat >= 1.686 Or Missing, sqft_living >= 3.2975 Or Missing, grade < 8, lat >= 1.6863 Or Missing, lat >= 1.6876, sqft_living < 3.3647 Or Missing, then price_range =low.

- If lat >= 1.686 Or Missing, sqft_living >= 3.2975 Or Missing, grade < 8, lat >= 1.6863 Or Missing, lat >= 1.6876, sqft_living >=3.3647 Or Missing, then price_range =high.

The decision tree predictive model was built with the train data and the statistical results are obtained as shown below. The result shows that the misclassification rate of the decision tree model is 0.12751.

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_range		_NOBS_	Sum of Frequencies	10806	10807	
price_range		_MISC_	Misclassification Rate	0.117897	0.12751	
price_range		_MAX_	Maximum Absolute Error	0.981655	0.981655	
price_range		_SSE_	Sum of Squared Errors	1884.061	2050.38	
price_range		_ASE_	Average Squared Error	0.087177	0.094863	
price_range		_RASE_	Root Average Squared Error	0.295257	0.307999	
price_range		_DIV_	Divisor for ASE	21612	21614	
price_range		_DFT_	Total Degrees of Freedom	10806	.	

Figure 5.4.1.2: Statistical output

The variable importance for decision tree is acquired as below. It is found that the three variables that have highest importance are latitude (lat), square footage of the apartments interior living space (sqft_living) and grade with importance 1.000, 0.8081 and 0.2468 respectively.

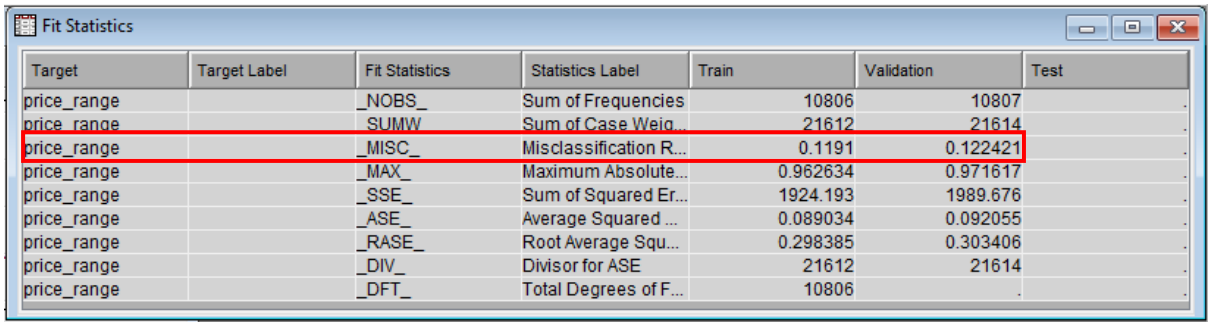
Variable Name	Label	Number of Splitting Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
REP_LG10_IMP_REP_lat	Replacement: Transformed: Imputed: Replac...	11	1.0000	1.0000	1.0000
REP_LG10_IMP_REP_sqft_living	Replacement: Transformed: Imputed: Replac...	9	0.8081	0.8390	1.0383
grade		5	0.2468	0.2648	1.0733
REP_LG10_IMP_REP_sqft_lot15	Replacement: Transformed: Imputed: Replac...	4	0.2153	0.2150	0.9984
view		3	0.1921	0.1637	0.8525
REP_LG10_IMP_REP_bedrooms	Replacement: Transformed: Imputed: Replac...	0	0.0000	0.0000	.
REP_yr_renovated	Replacement: yr_renovated	0	0.0000	0.0000	.
condition		0	0.0000	0.0000	.
waterfront		0	0.0000	0.0000	.

Figure 5.4.1.3: Variable importance

5.4.2 Constructing a Gradient Boosting Predictive Model

Gradient boosting is a machine learning method used for both regression and classification tasks. In this project, it will be applied specifically for classification tasks. Unlike a single decision tree, gradient boosting is less prone to overfitting the data. The accuracy of a model is crucial, and the gradient boosting algorithm can aid in reducing the bias error of the model.

The Gradient Boosting Node of SAS Enterprise Miner can be utilized to connect and analyze previously divided data. The statistics resulting from running this node are illustrated in the figure below. The results indicate that the misclassification rate of the gradient boosting model produced is 0.122421.

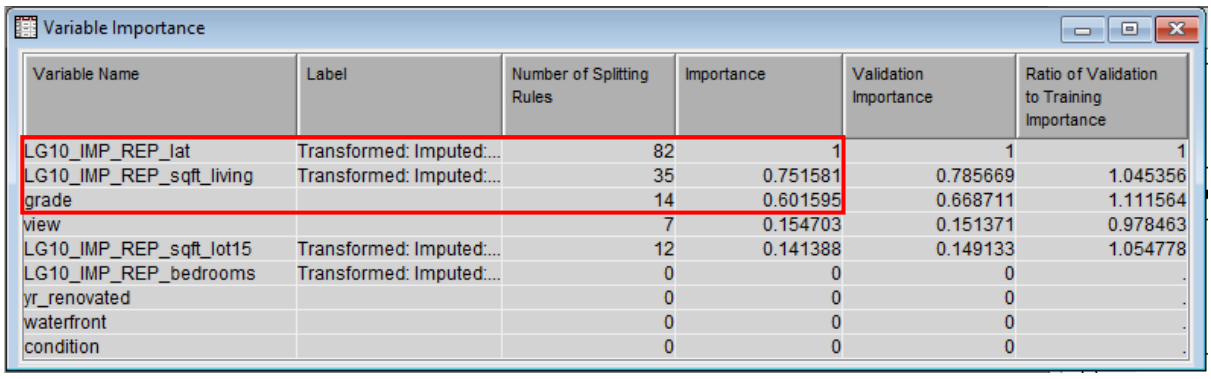


Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_range		_NOBS_	Sum of Frequencies	10806	10807	.
price_range		_SUMW_	Sum of Case Weig...	21612	21614	.
price_range		_MISC_	Misclassification R...	0.1191	0.122421	.
price_range		_MAX_	Maximum Absolute...	0.962634	0.971617	.
price_range		_SSE_	Sum of Squared Er...	1924.193	1989.676	.
price_range		_ASE_	Average Squared ...	0.089034	0.092055	.
price_range		_RASE_	Root Average Squ...	0.298385	0.303406	.
price_range		_DIV_	Divisor for ASE	21612	21614	.
price_range		_DFT_	Total Degrees of F...	10806	.	.

Figure 5.4.2.1: Fit Statistics

The table below shows the importance of variables in the Gradient Boosting model. The three key factors that have the greatest impact on the results of this model are LG10_IMP_REP_lat, LG10_IMP_REP_sqft_living, and grade.



Variable Importance

Variable Name	Label	Number of Splitting Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
LG10_IMP_REP_lat	Transformed: Imputed:...	82	1	1	1
LG10_IMP_REP_sqft_living	Transformed: Imputed:...	35	0.751581	0.785669	1.045356
grade		14	0.601595	0.668711	1.111564
view		7	0.154703	0.151371	0.978463
LG10_IMP_REP_sqft_lot15	Transformed: Imputed:...	12	0.141388	0.149133	1.054778
LG10_IMP_REP_bedrooms	Transformed: Imputed:...	0	0	0	.
yr_renovated		0	0	0	.
waterfront		0	0	0	.
condition		0	0	0	.

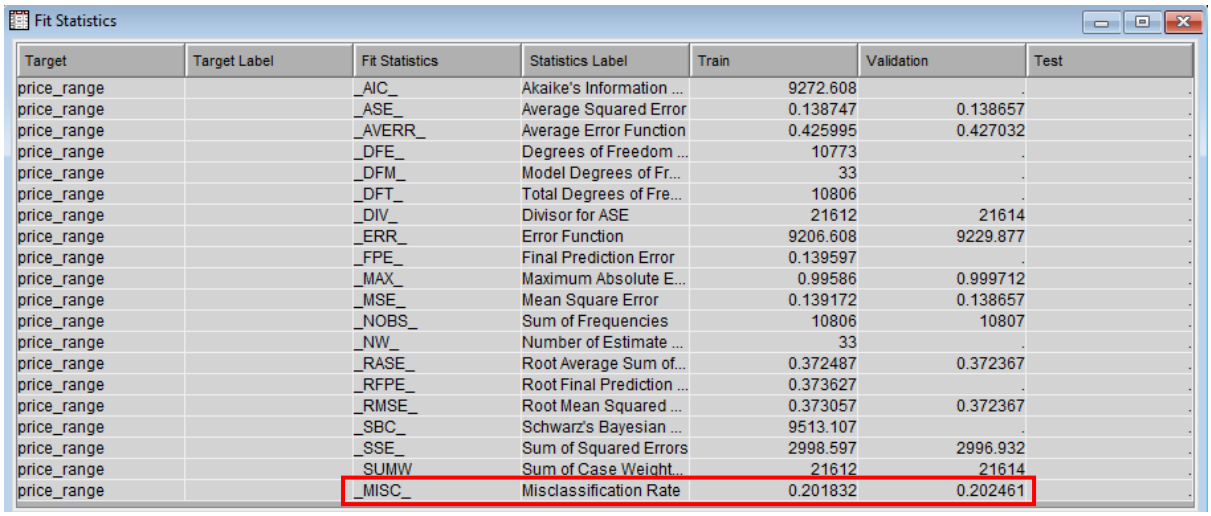
Figure 5.4.2.2: Variable Importance

5.4.3 Constructing a Logistics Regression Predictive Model

Logistic regression is a statistical model used for classification and predictive analytics. It is used to compute the probability of an event taking place by having the log-odds for the event be a linear combination of one or more independent variables. Logistic regression uses a logistic function to run a regression analysis where the dependent variable is binary ranging from 0 and 1. At the core of logistic regression analysis is the task of estimating the log odds of an event. Logistic regression is used to explain the relationship between of all data type (nominal, interval, ordinal, ratio) of independent variables with binary dependant variables.

The SAS Enterprise miner model number regression was run through the stepwise method.

The regression node was used to link with the data partition node. Logistic regression was selected as the regression type. The figure shows a misclassification rate of. 0.202461 This gives us an accuracy of 79.75%. The result of the statistics test is shown as figure below.



Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_range		_AIC_	Akaike's Information ...	9272.608	.	.
price_range		_ASE_	Average Squared Error	0.138747	0.138657	.
price_range		_AVERR_	Average Error Function	0.425995	0.427032	.
price_range		_DFE_	Degrees of Freedom ...	10773	.	.
price_range		_DFM_	Model Degrees of Fr...	33	.	.
price_range		_DFT_	Total Degrees of Fre...	10806	.	.
price_range		_DIV_	Divisor for ASE	21612	21614	.
price_range		_ERR_	Error Function	9206.608	9229.877	.
price_range		_FPE_	Final Prediction Error	0.139597	.	.
price_range		_MAX_	Maximum Absolute E...	0.99586	0.999712	.
price_range		_MSE_	Mean Square Error	0.139172	0.138657	.
price_range		_NOBS_	Sum of Frequencies	10806	10807	.
price_range		_NW_	Number of Estimate ...	33	.	.
price_range		_RASE_	Root Average Sum of...	0.372487	0.372367	.
price_range		_RFPE_	Root Final Prediction ...	0.373627	.	.
price_range		_RMSE_	Root Mean Squared ...	0.373057	0.372367	.
price_range		_SBC_	Schwarz's Bayesian ...	9513.107	.	.
price_range		_SSE_	Sum of Squared Errors	2998.597	2996.932	.
price_range		_SUMW_	Sum of Case Weight...	21612	21614	.
price_range		_MISC_	Misclassification Rate	0.201832	0.202461	.

Figure 5.4.3.1 Statistical results for Logistic Regression

The Odds Ratio Estimates shows on the strongness an event is associated with exposure where in this scenario would be on the price range.

Odds Ratio Estimates

Effect		price_range	Point Estimate
LG10_IMP_REP_bathrooms		1	4.283
LG10_IMP_REP_bedrooms		1	0.103
LG10_IMP_REP_lat		1	.
LG10_IMP_REP_long		1	29.239
LG10_IMP_REP_sqft_above		1	9.406
LG10_IMP_REP_sqft_basement		1	1.464
LG10_IMP_REP_sqft_living		1	3.435
LG10_IMP_REP_sqft_living15		1	57.098
LG10_IMP_REP_sqft_lot		1	0.873
LG10_IMP_REP_sqft_lot15		1	0.452
condition	1 vs 5	1	0.412
condition	2 vs 5	1	0.357
condition	3 vs 5	1	0.583
condition	4 vs 5	1	0.629
floors		1	1.962

Figure 5.4.3.2 Odd Ratio Estimates for Logistic Regression

Based on the output of Odds Ratio Estimates from we found that:

1) Sqft_living15 has 57 times the odds of having higher price range than lower price range level.

2) Long(Longitude) has 29 times the odds of having higher price range than lower price range level.

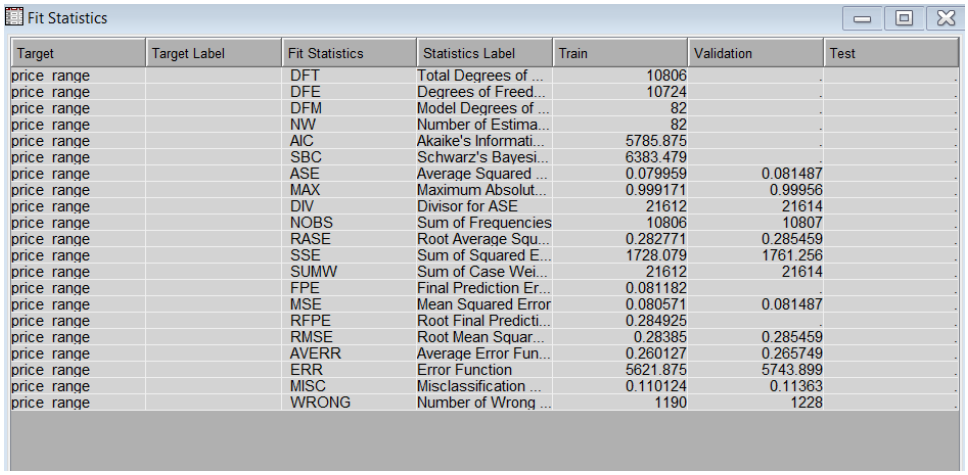
3) Sqft_above has 9 times the odd of having higher price range than lower price range level.

5.4.4 Constructing a Neural Network Predictive Model

A neural network is a type of machine learning process (deep learning) and a collection of algorithms that employ linked neurons or nodes in a layered framework to mimic the human brain. Therefore, it develops an adaptive system that continually learns from previous errors and improvements. With this adaptation to the changes of input, the network can achieve success without having to change the output criterion.

The existing neural network node of SAS Enterprise Miner can be utilized to connect and analyze previously divided data. Different hidden layers with optimized hidden units of neural network models are drawn and shown below. The statistics resulting from running neural network node are also illustrated below in figures.

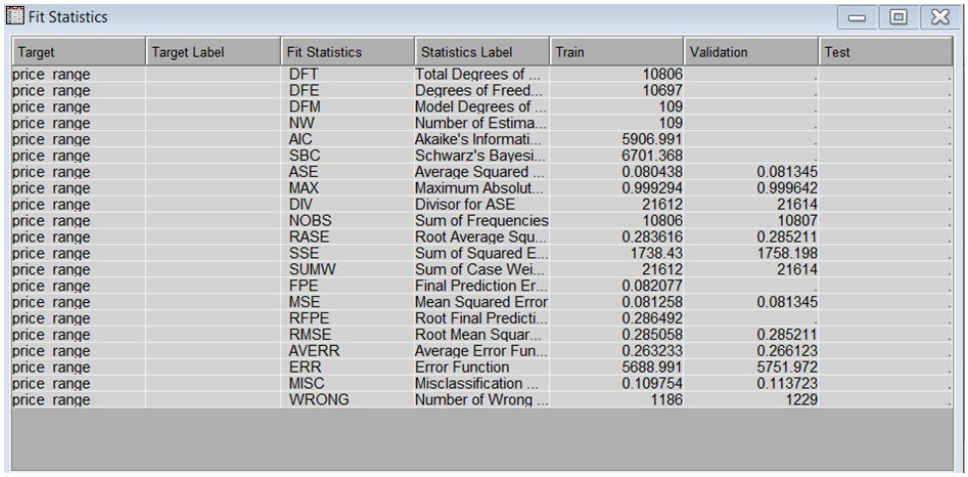
From the statistics below, the three models with the lowest misclassification rate are the number of hidden units of 5, 3 and 4. The misclassification rate of a neural network model with several hidden units of 5 is 0.111872. The misclassification rate of a neural network model with several hidden units of 3 is 0.11363. The misclassification rate of a neural network model with several hidden units of 4 is 0.113723.



Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_range		DFT	Total Degrees of ...	10806		
price_range		DFE	Degrees of Freed...	10724		
price_range		DFM	Model Degrees of ...	82		
price_range		NW	Number of Estima...	82		
price_range		AIC	Akaike's Informati...	5785.875		
price_range		SBC	Schwarz's Bayesi...	6383.479		
price_range		ASE	Average Squared ...	0.079959	0.081487	
price_range		MAX	Maximum Absolut...	0.999171	0.99956	
price_range		DIV	Divisor for ASE	21612	21614	
price_range		NOBS	Sum of Frequencies	10806	10807	
price_range		RASE	Root Average Squ...	0.282771	0.285459	
price_range		SSE	Sum of Squared E...	1728.079	1761.256	
price_range		SUMW	Sum of Case Wei...	21612	21614	
price_range		FPE	Final Prediction Er...	0.081182		
price_range		MSE	Mean Squared Error	0.080571	0.081487	
price_range		RFPE	Root Final Predicti...	0.284925		
price_range		RMSE	Root Mean Squar...	0.28385	0.285459	
price_range		AVERR	Average Error Fun...	0.260127	0.265749	
price_range		ERR	Error Function	5621.875	5743.899	
price_range		MISC	Misclassification ...	0.110124	0.11363	
price_range		WRONG	Number of Wrong ...	1190	1228	

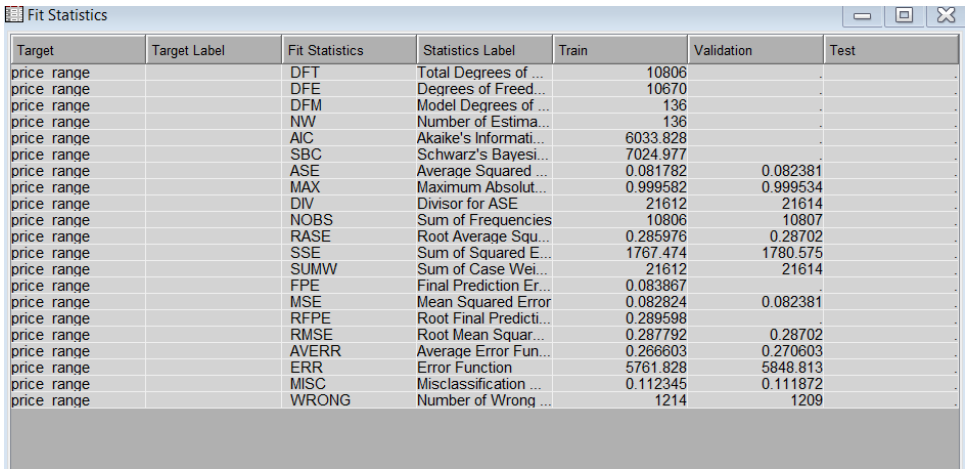
Figure 5.4.4.1: Statistical results for 3 layers neural network (Misclassification rate= 0.11363)



Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_range		DFT	Total Degrees of ...	10806		
price_range		DFE	Degrees of Freed...	10697		
price_range		DFM	Model Degrees of ...	109		
price_range		NW	Number of Estima...	109		
price_range		AIC	Akaike's Informati...	5906.991		
price_range		SBC	Schwarz's Bayesi...	6701.368		
price_range		ASE	Average Squared ...	0.080438	0.081345	
price_range		MAX	Maximum Absolut...	0.999294	0.999642	
price_range		DIV	Divisor for ASE	21612	21614	
price_range		NOBS	Sum of Frequencies	10806	10807	
price_range		RASE	Root Average Squ...	0.283616	0.285211	
price_range		SSE	Sum of Squared E...	1738.43	1758.198	
price_range		SUMW	Sum of Case Wei...	21612	21614	
price_range		FPE	Final Prediction Er...	0.082077		
price_range		MSE	Mean Squared Error	0.081258	0.081345	
price_range		RFPE	Root Final Predicti...	0.286492		
price_range		RMSE	Root Mean Squar...	0.285058	0.285211	
price_range		AVERR	Average Error Fun...	0.263233	0.266123	
price_range		ERR	Error Function	5688.991	5751.972	
price_range		MISC	Misclassification ...	0.109754	0.113723	
price_range		WRONG	Number of Wrong ...	1186	1229	

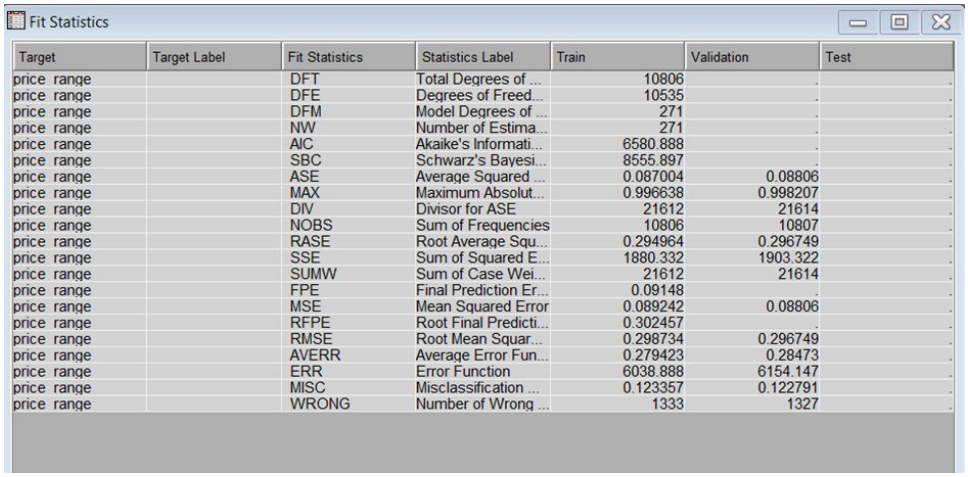
Figure 5.4.4.2: Statistical results for 4 layers neural network (Misclassification rate= 0.113723)



Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_range		DFT	Total Degrees of ...	10806		
price_range		DFE	Degrees of Freed...	10670		
price_range		DFM	Model Degrees of ...	136		
price_range		NW	Number of Estima...	136		
price_range		AIC	Akaike's Informati...	6033.828		
price_range		SBC	Schwarz's Bayesi...	7024.977		
price_range		ASE	Average Squared ...	0.081782	0.082381	
price_range		MAX	Maximum Absolut...	0.999582	0.999534	
price_range		DIV	Divisor for ASE	21612	21614	
price_range		NOBS	Sum of Frequencies	10806	10807	
price_range		RASE	Root Average Squ...	0.285976	0.28702	
price_range		SSE	Sum of Squared E...	1767.474	1780.575	
price_range		SUMW	Sum of Case Wei...	21612	21614	
price_range		FPE	Final Prediction Er...	0.083867		
price_range		MSE	Mean Squared Error	0.082824	0.082381	
price_range		RFPE	Root Final Predicti...	0.289598		
price_range		RMSE	Root Mean Squar...	0.287792	0.28702	
price_range		AVERR	Average Error Fun...	0.266603	0.270603	
price_range		ERR	Error Function	5761.828	5848.813	
price_range		MISC	Misclassification ...	0.112345	0.111872	
price_range		WRONG	Number of Wrong ...	1214	1209	

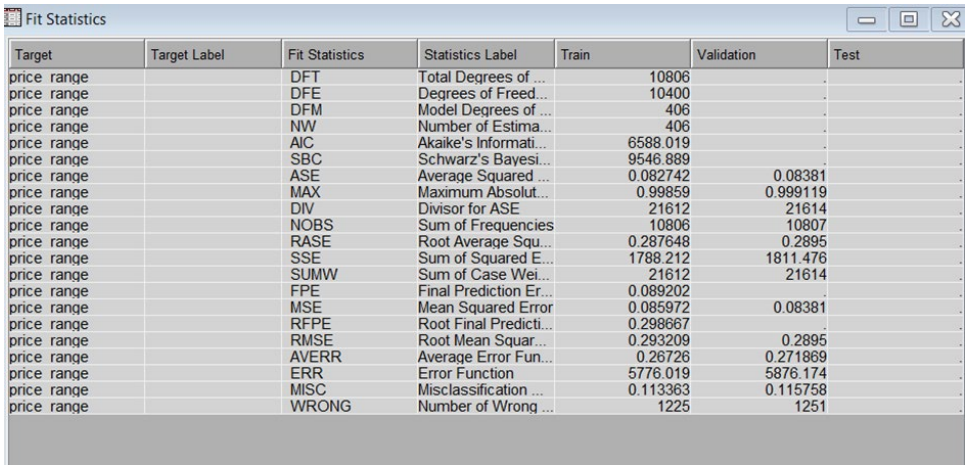
Figure 5.4.4.3: Statistical results for 5 layers neural network (Misclassification rate= 0.111872)



Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_range		DFT	Total Degrees of ...	10806		
price_range		DFE	Degrees of Freed...	10535		
price_range		DFM	Model Degrees of ...	271		
price_range		NW	Number of Estima...	271		
price_range		AIC	Akaike's Informati...	6580.888		
price_range		SBC	Schwarz's Bayesi...	8555.897		
price_range		ASE	Average Squared ...	0.087004	0.08806	
price_range		MAX	Maximum Absolut...	0.996638	0.998207	
price_range		DIV	Divisor for ASE	21612	21614	
price_range		NOBS	Sum of Frequencies	10806	10807	
price_range		RASE	Root Average Squ...	0.294964	0.296749	
price_range		SSE	Sum of Squared E...	1880.332	1903.322	
price_range		SUMW	Sum of Case Wei...	21612	21614	
price_range		FPE	Final Prediction Er...	0.09148		
price_range		MSE	Mean Squared Error	0.089242	0.08806	
price_range		RFPE	Root Final Predicti...	0.302457		
price_range		RMSE	Root Mean Squar...	0.298734	0.296749	
price_range		AVERR	Average Error Fun...	0.279423	0.28473	
price_range		ERR	Error Function	6038.888	6154.147	
price_range		MISC	Misclassification ...	0.123357	0.122791	
price_range		WRONG	Number of Wrong ...	1333	1327	

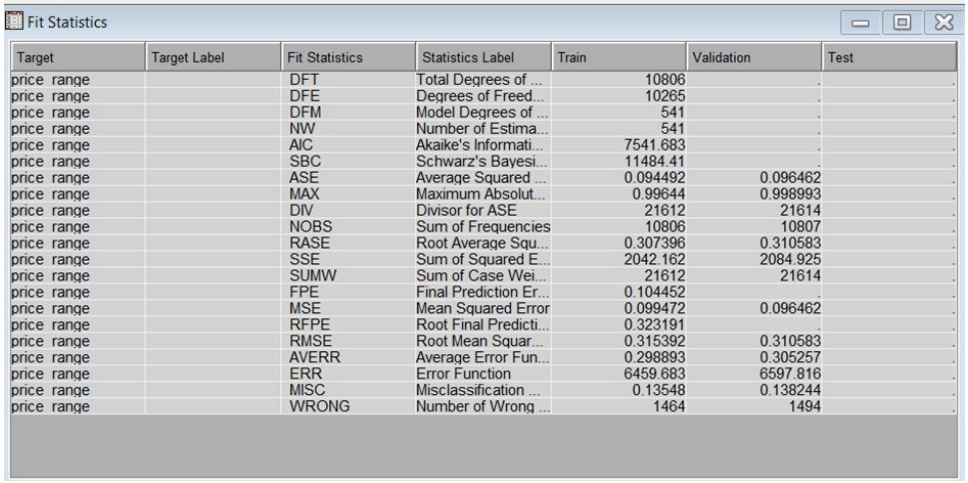
Figure 5.4.4.4: Statistical results for 10 layers neural network (Misclassification rate= 0.122791)



Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_range		DFT	Total Degrees of ...	10806		
price_range		DFE	Degrees of Freed...	10400		
price_range		DFM	Model Degrees of ...	406		
price_range		NW	Number of Estima...	406		
price_range		AIC	Akaike's Informati...	6588.019		
price_range		SBC	Schwarz's Bayesi...	9546.889		
price_range		ASE	Average Squared ...	0.082742	0.08381	
price_range		MAX	Maximum Absolut...	0.99859	0.999119	
price_range		DIV	Divisor for ASE	21612	21614	
price_range		NOBS	Sum of Frequencies	10806	10807	
price_range		RASE	Root Average Squ...	0.287648	0.2895	
price_range		SSE	Sum of Squared E...	1788.212	1811.476	
price_range		SUMW	Sum of Case Wei...	21612	21614	
price_range		FPE	Final Prediction Er...	0.089202		
price_range		MSE	Mean Squared Error	0.085972	0.08381	
price_range		RFPE	Root Final Predicti...	0.298667		
price_range		RMSE	Root Mean Squar...	0.293209	0.2895	
price_range		AVERR	Average Error Fun...	0.26726	0.271869	
price_range		ERR	Error Function	5776.019	5876.174	
price_range		MISC	Misclassification ...	0.113363	0.115758	
price_range		WRONG	Number of Wrong ...	1225	1251	

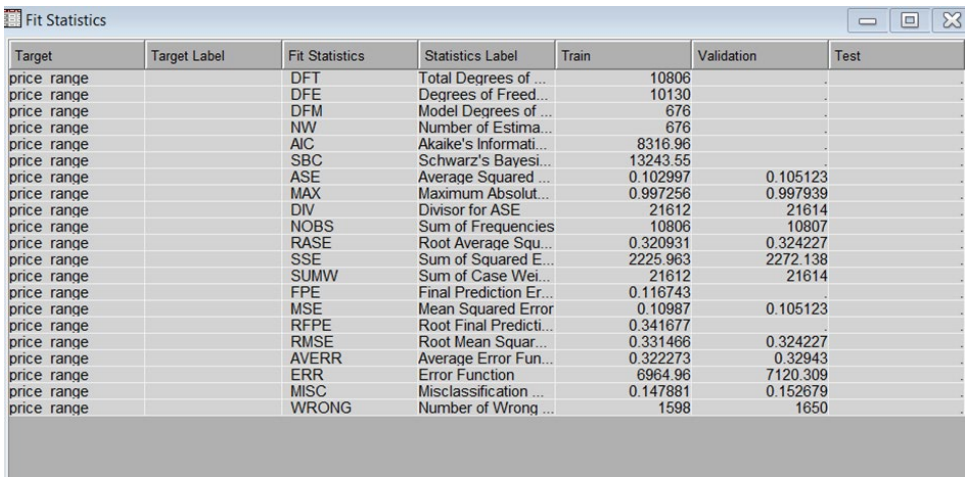
Figure 5.4.4.5: Statistical results for 15 layers neural network (Misclassification rate= 0.115758)



Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_range		DFT	Total Degrees of ...	10806		
price_range		DFE	Degrees of Freed...	10265		
price_range		DFM	Model Degrees of ...	541		
price_range		NW	Number of Estima...	541		
price_range		AIC	Akaike's Informati...	7541.683		
price_range		SBC	Schwarz's Bayesi...	11484.41		
price_range		ASE	Average Squared ...	0.094492	0.096462	
price_range		MAX	Maximum Absolut...	0.99644	0.998993	
price_range		DIV	Divisor for ASE	21612	21614	
price_range		NOBS	Sum of Frequencies	10806	10807	
price_range		RASE	Root Average Squ...	0.307396	0.310583	
price_range		SSE	Sum of Squared E...	2042.162	2084.925	
price_range		SUMW	Sum of Case Wei...	21612	21614	
price_range		FPE	Final Prediction Er...	0.104452		
price_range		MSE	Mean Squared Error	0.099472	0.096462	
price_range		RFPE	Root Final Predicti...	0.323191		
price_range		RMSE	Root Mean Squar...	0.315392	0.310583	
price_range		AVERR	Average Error Fun...	0.298893	0.305257	
price_range		ERR	Error Function	6459.683	6597.816	
price_range		MISC	Misclassification ...	0.13548	0.138244	
price_range		WRONG	Number of Wrong ...	1464	1494	

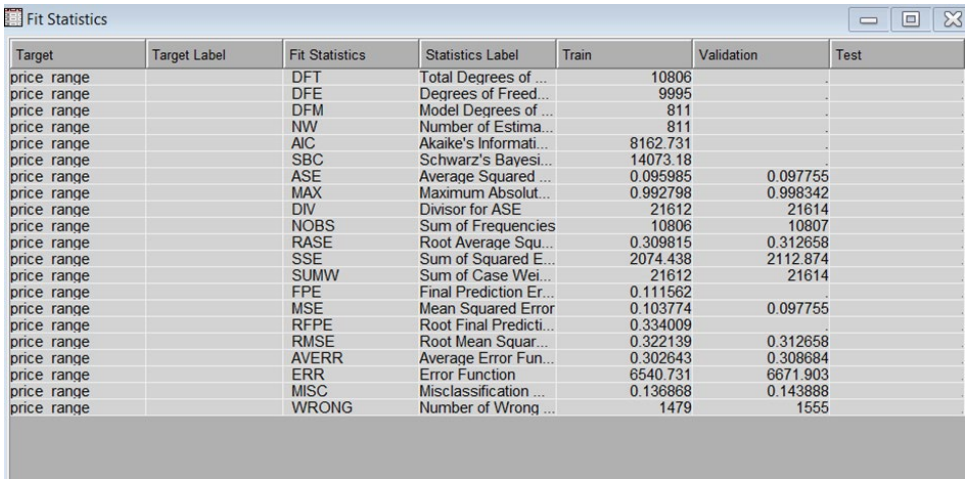
Figure 5.4.4.6: Statistical results for 20 layers neural network (Misclassification rate= 0.138244)



Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_range		DFT	Total Degrees of ...	10806		
price_range		DFE	Degrees of Freed...	10130		
price_range		DFM	Model Degrees of ...	676		
price_range		NW	Number of Estima...	676		
price_range		AIC	Akaike's Informati...	8316.96		
price_range		SBC	Schwarz's Bayesi...	13243.55		
price_range		ASE	Average Squared ...	0.102997	0.105123	
price_range		MAX	Maximum Absolut...	0.997256	0.997939	
price_range		DIV	Divisor for ASE	21612	21614	
price_range		NOBS	Sum of Frequencies	10806	10807	
price_range		RASE	Root Average Squ...	0.320931	0.324227	
price_range		SSE	Sum of Squared E...	2225.963	2272.138	
price_range		SUMW	Sum of Case Wei...	21612	21614	
price_range		FPE	Final Prediction Er...	0.116743		
price_range		MSE	Mean Squared Error	0.10987	0.105123	
price_range		RFPE	Root Final Predicti...	0.341677		
price_range		RMSE	Root Mean Squar...	0.331466	0.324227	
price_range		AVERR	Average Error Fun...	0.322273	0.32943	
price_range		ERR	Error Function	6964.96	7120.309	
price_range		MISC	Misclassification ...	0.147881	0.152679	
price_range		WRONG	Number of Wrong ...	1598	1650	

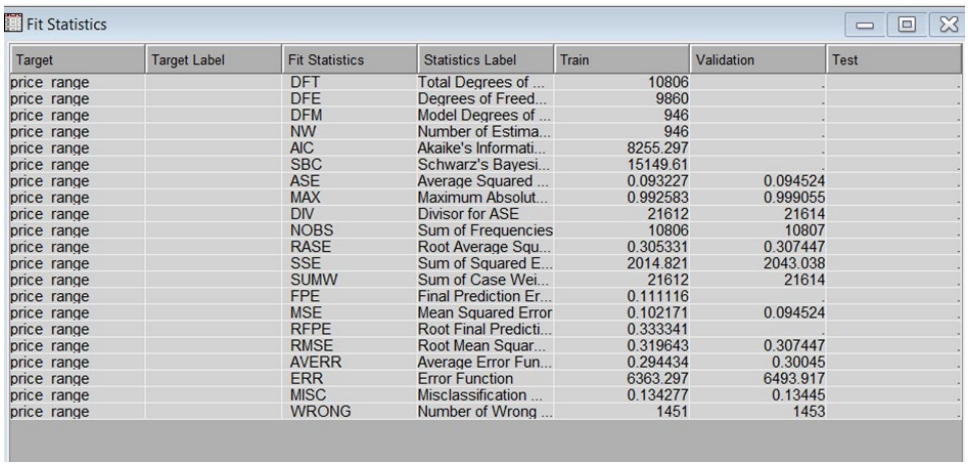
Figure 5.4.4.7: Statistical results for 25 layers neural network (Misclassification rate= 0.152679)



Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_range		DFT	Total Degrees of ...	10806		
price_range		DFE	Degrees of Freed...	9995		
price_range		DFM	Model Degrees of ...	811		
price_range		NW	Number of Estima...	811		
price_range		AIC	Akaike's Informati...	8162.731		
price_range		SBC	Schwarz's Bayesi...	14073.18		
price_range		ASE	Average Squared ...	0.095985	0.097755	
price_range		MAX	Maximum Absolut...	0.992798	0.998342	
price_range		DIV	Divisor for ASE	21612	21614	
price_range		NOBS	Sum of Frequencies	10806	10807	
price_range		RASE	Root Average Squ...	0.309815	0.312658	
price_range		SSE	Sum of Squared E...	2074.438	2112.874	
price_range		SUMW	Sum of Case Wei...	21612	21614	
price_range		FPE	Final Prediction Er...	0.111562		
price_range		MSE	Mean Squared Error	0.103774	0.097755	
price_range		RFPE	Root Final Predicti...	0.334009		
price_range		RMSE	Root Mean Squar...	0.322139	0.312658	
price_range		AVERR	Average Error Fun...	0.302643	0.308684	
price_range		ERR	Error Function	6540.731	6671.903	
price_range		MISC	Misclassification ...	0.136868	0.143888	
price_range		WRONG	Number of Wrong ...	1479	1555	

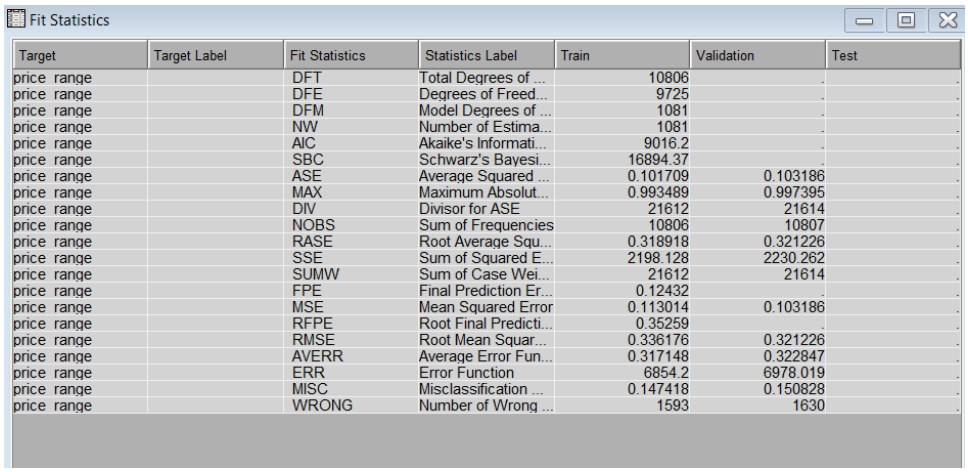
Figure 5.4.4.8: Statistical results for 30 layers neural network (Misclassification rate= 0.143888)



Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_range		DFT	Total Degrees of ...	10806		
price_range		DFE	Degrees of Freed...	9860		
price_range		DFM	Model Degrees of ...	946		
price_range		NW	Number of Estima...	946		
price_range		AIC	Akaike's Informati...	8255.297		
price_range		SBC	Schwarz's Bayesi...	15149.61		
price_range		ASE	Average Squared ...	0.093227	0.094524	
price_range		MAX	Maximum Absolut...	0.992583	0.999055	
price_range		DIV	Divisor for ASE	21612	21614	
price_range		NOBS	Sum of Frequencies	10806	10807	
price_range		RASE	Root Average Squ...	0.305331	0.307447	
price_range		SSE	Sum of Squared E...	2014.821	2043.038	
price_range		SUMW	Sum of Case Wei...	21612	21614	
price_range		FPE	Final Prediction Er...	0.111116		
price_range		MSE	Mean Squared Error	0.102171	0.094524	
price_range		RFPE	Root Final Predicti...	0.333341		
price_range		RMSE	Root Mean Squar...	0.319643	0.307447	
price_range		AVERR	Average Error Fun...	0.294434	0.30045	
price_range		ERR	Error Function	6363.297	6493.917	
price_range		MISC	Misclassification ...	0.134277	0.13445	
price_range		WRONG	Number of Wrong ...	1451	1453	

Figure 5.4.4.9: Statistical results for 35 layers neural network (Misclassification rate= 0.13445)



Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_range		DFT	Total Degrees of ...	10806		
price_range		DFE	Degrees of Freed...	9725		
price_range		DFM	Model Degrees of ...	1081		
price_range		NW	Number of Estima...	1081		
price_range		AIC	Akaike's Informati...	9016.2		
price_range		SBC	Schwarz's Bayesi...	16894.37		
price_range		ASE	Average Squared ...	0.101709	0.103186	
price_range		MAX	Maximum Absolut...	0.993489	0.997395	
price_range		DIV	Divisor for ASE	21612	21614	
price_range		NOBS	Sum of Frequencies	10806	10807	
price_range		RASE	Root Average Squ...	0.318918	0.321226	
price_range		SSE	Sum of Squared E...	2198.128	2230.262	
price_range		SUMW	Sum of Case Wei...	21612	21614	
price_range		FPE	Final Prediction Er...	0.12432		
price_range		MSE	Mean Squared Error	0.113014	0.103186	
price_range		RFPE	Root Final Predicti...	0.35259		
price_range		RMSE	Root Mean Squar...	0.336176	0.321226	
price_range		AVERR	Average Error Fun...	0.317148	0.322847	
price_range		ERR	Error Function	6854.2	6978.019	
price_range		MISC	Misclassification ...	0.147418	0.150828	
price_range		WRONG	Number of Wrong ...	1593	1630	

Figure 5.4.4.10: Statistical results for 40 layers neural network (Misclassification rate= 0.150828)

5.5 ASSESS – Data Assessment

At first, we compare the performance between Decision Tree, Logistic Regression, and Gradient Boosting. We observed that the Gradient Boosting model has the lowest misclassification rate of 0.122421, resulting with the highest accuracy of 88%, a very close margin with Decision Tree with 0.12751 misclassification rate, resulting in 87% of accuracy.

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Train: Sum of Frequencies	Train: Misclassification Rate	Selection Criterion: Valid: Misclassification Rate
Y	Boost	Boost	Gradient Bo...	price_range	10806	0.1191	0.122421
	Tree	Tree	Decision Tr...	price_range	10806	0.117897	0.12751
	Reg	Reg	Regression	price_range	10806	0.201832	0.202461

Figure 5.5.1: Performance between Decision Tree, Logistic Regression, Gradient Boosting

Second, we compare the performance between the neural networks. The 5 layers of neural network has the lowest misclassification rate of 0.111872 and therefore achieve an accuracy of 88.81%

Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate
Neural3	5 layer N...	price ran...		0.111872
Neural	3 layer N...	price ran...		0.11363
Neural2	4 layer N...	price ran...		0.113723
Neural5	15 layer ...	price ran...		0.115758
Neural9	10 layer ...	price ran...		0.122791
Neural7	35 layer ...	price ran...		0.13445
Neural10	20 layer ...	price ran...		0.138244
Neural8	30 layer ...	price ran...		0.143888
Neural6	40 layer ...	price ran...		0.150828
Neural4	25 layer ...	price ran...		0.152679

Figure 5.5.2 Performance between the neural networks

Finally, we compare the two best models from the above which is the Gradient Boosting Model and the 5-layer Neural Network. We observe that 5-layer Neural Network has the best classification accuracy with 89%.

Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate
Neural11	5 layer N...	price ran...		0.111872
Boost	Gradient ...	price ran...		0.122421

Figure 5.5.3: Performance between 5-layer Neural Network and Gradient Boosting

6 Conclusion

After we revised the data, there are 21 variables in the house sales dataset including 16 interval variables, 1 binary variable, 3 ordinal variables, and 1 nominal variable they became as follows:

Role	Variable	Count
ID	Interval	1
Input	Binary	1
Input	Interval	14
Input	Ordinal	3
Target	Interval	1
Time ID	Nominal	1

During data exploration, we found incomplete data, noisy data, inconsistent data, and intentional data. These variables are as follows:

Variable	Error Type
bathrooms	Incomplete
yr_built	Inconsistent
bathrooms bedrooms sqft_above price lat long sqft_living15 sqft_lot15 sqft_living sqft_lot sqft_basement	Noisy

SAS Enterprise Miner was used to locate the relevant variables as listed in the table below using the variable selection tool.

Number	Variable
1	grade
2	lat
3	sqft_living
4	view
5	waterfront
6	condition
7	yr_renovated
8	sqft_lot15
9	bedrooms

Through the Modelling and Assessment Phase of SEMMA, we are able to build several predictive models such as Decision Tree, Gradient Boosting, Logistic Regression, and Neural Networks and subsequently, our results find that 5-layer Neural Network produces the best accuracy.

Overall, through SEMMA, we find some interesting patterns that stands out.

Interesting Patterns	**Found In**
Higher grade level shows a high positive relationship with price. Longer boxplot body length and higher price were observed as the grade increased	Exploration
Sqft_living shows a positive relationship with price	Exploration
Sqft_living and sqft_above show a strong correlation coefficient, 0.8766.	Exploration
Latitude is selected as the root node as it has a high level of information gain	Modelling Phase – Decision Tree
When the grade is high, the price range is more likely to be high.	Modelling Phase – Decision Tree

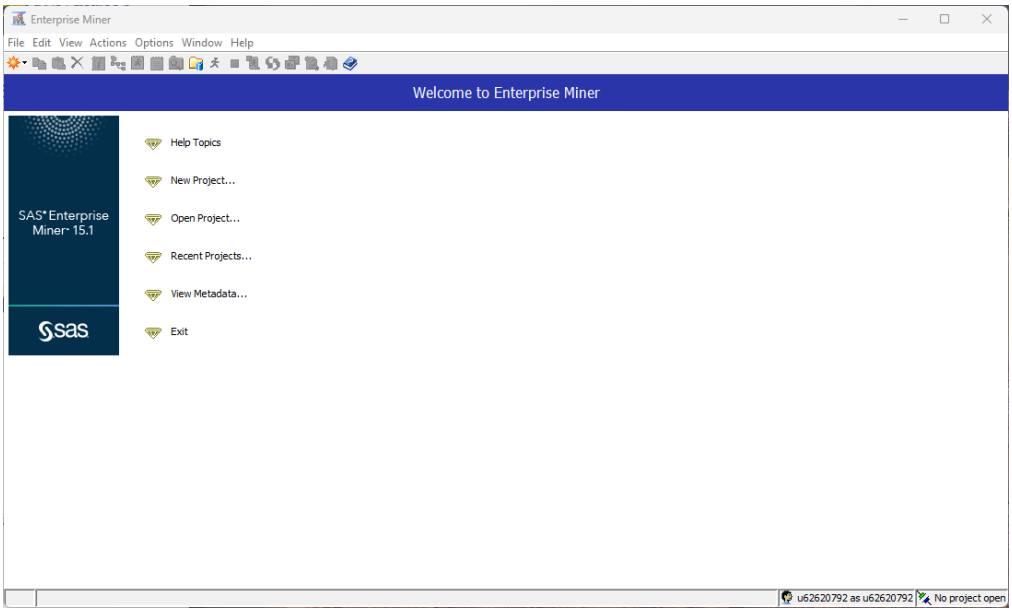
7 Appendix

In 7.1 and 7.2, there are steps consisting of the first 2 characters of the SAS SEMMA for creating a SAS Enterprise Miner project. For 7.3, it shows the full correlation table.

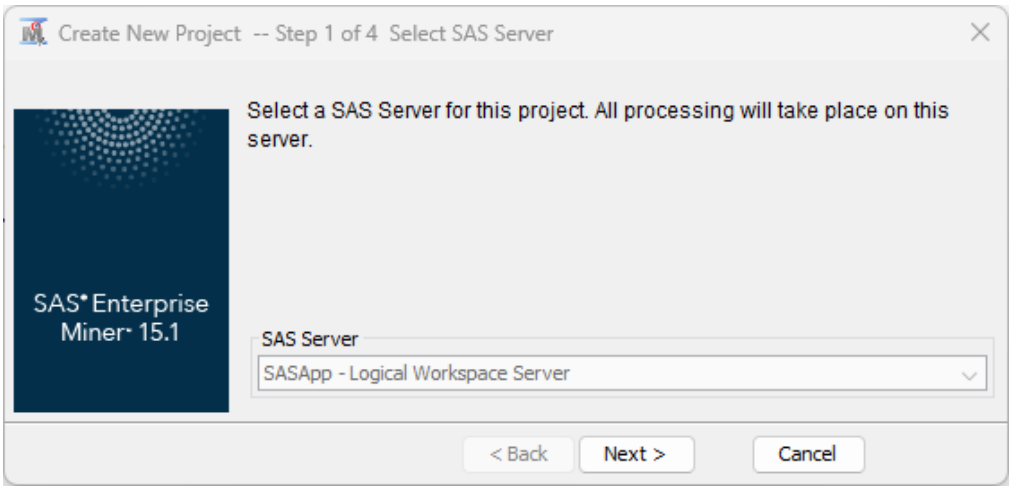
7.1 SAMPLE

7.1.1 Create a SAS Enterprise Miner Project

- Select **File** → **New** → **Project** from the main menu or select **New Project** directly

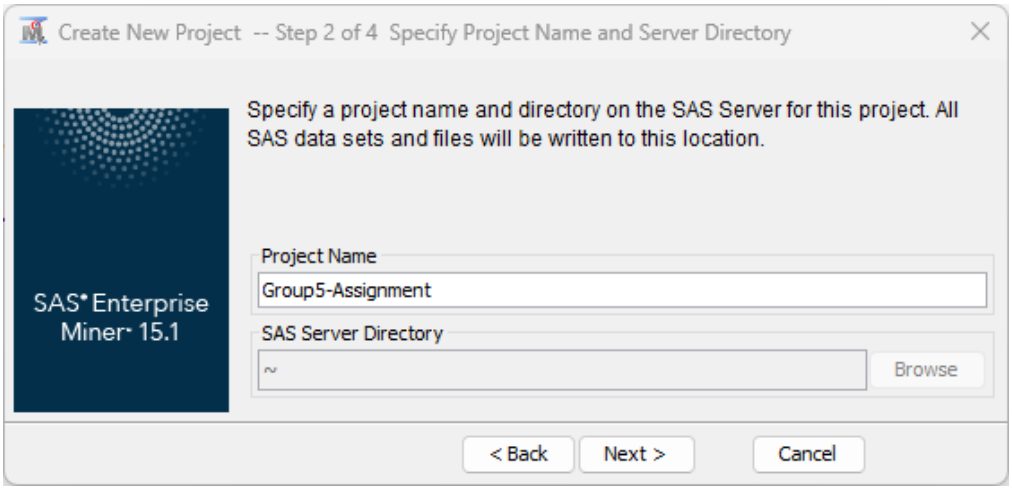


- Step 1 of 4 - Select SAS Server. We will use the default server. Select **Next >**

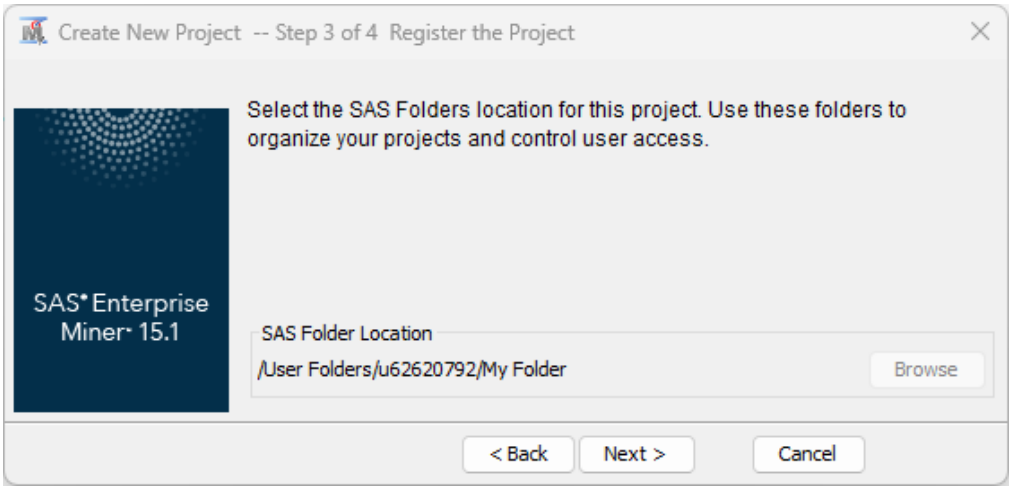


- tep 2 of 4 - Specify Project Name and Server Directory.

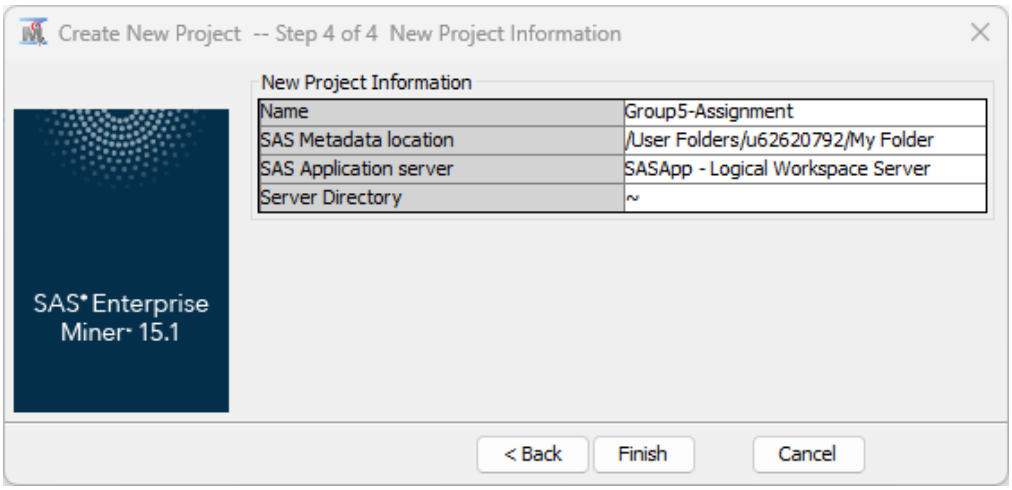
  Type Project Name: **Group5-Assignment** and Select **Next >**



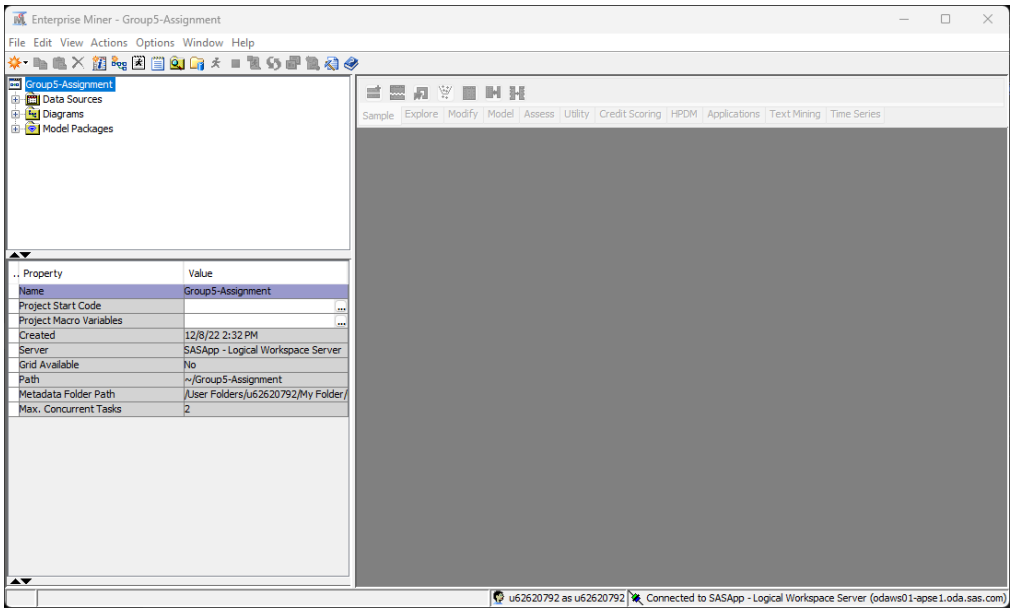
- Step 3 of 4 - Register the Project. Select **Next >**



- Step 4 of 4 - New Project Information. Select **Finish**



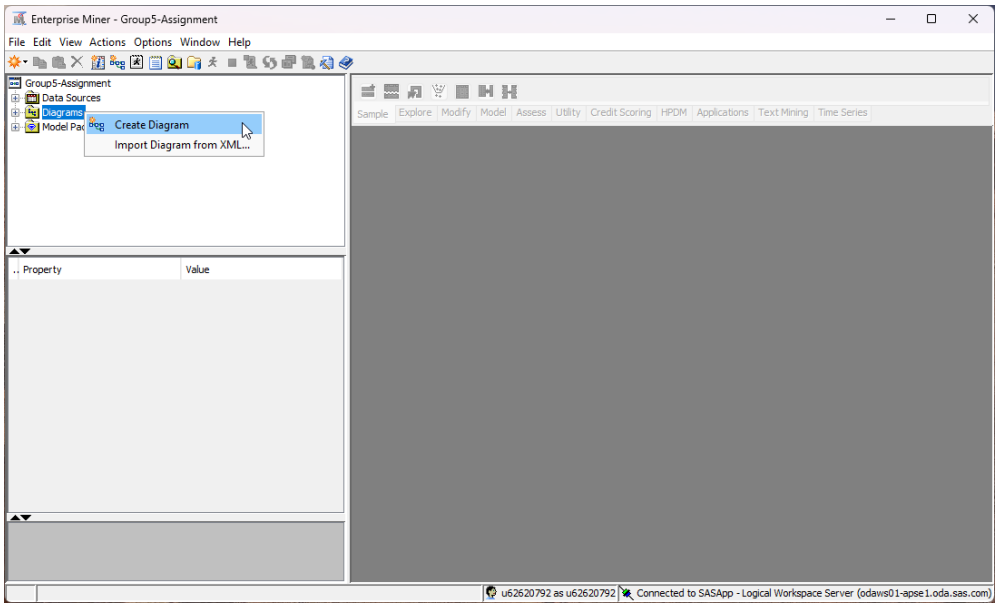
- The SAS Enterprise Miner Project is created successfully.



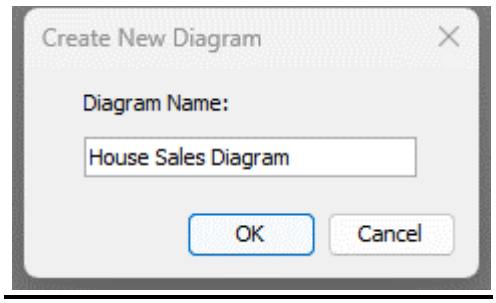
7.1.2 Create a SAS Enterprise Miner Diagram

- Select **File** → **New** → **Diagram**… from the Main Menu or

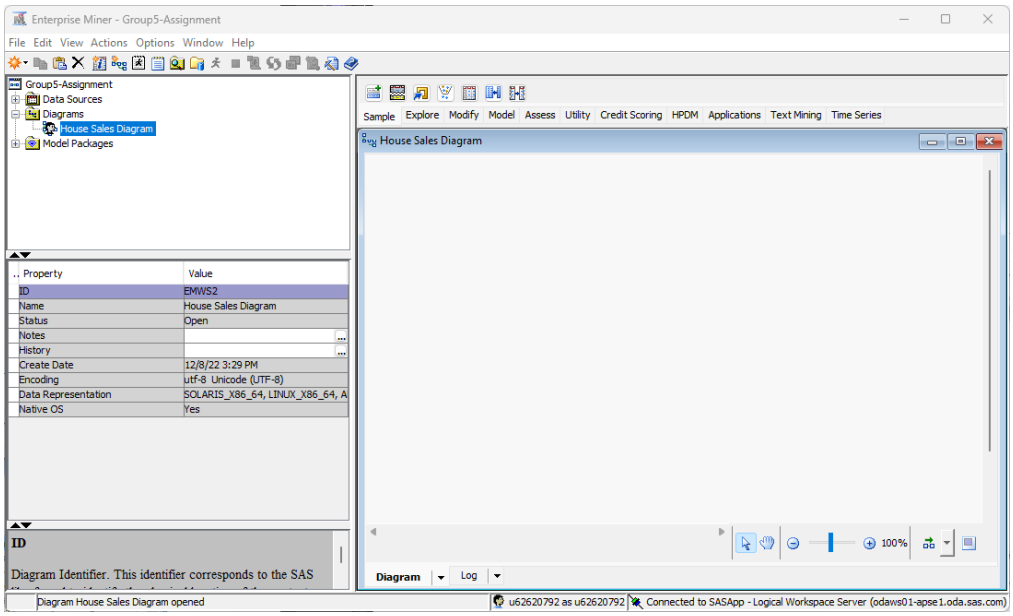
  Select Right-click **Diagrams** directly, Select **Create Diagram**



- Type Diagram Name: **House Sales Diagram** and Select **OK**

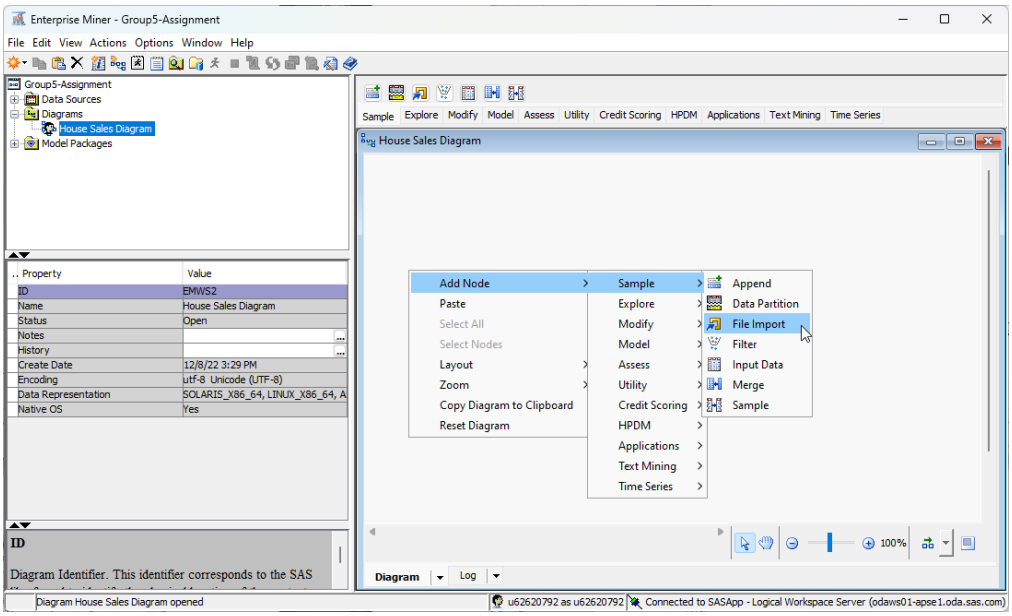


- **House Sales Diagram** is created successfully



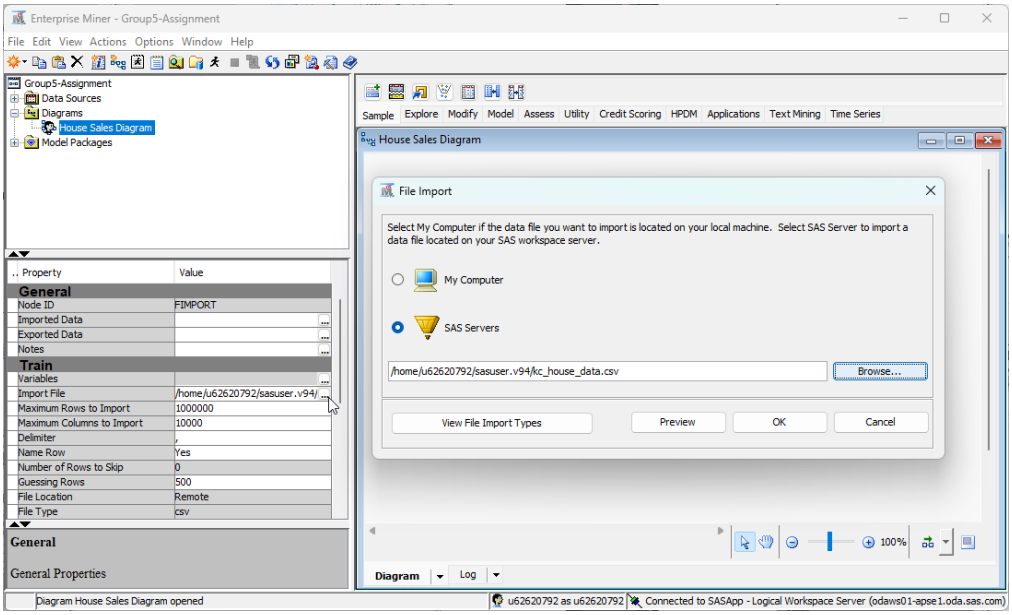
- Create a Sample Node to import the dataset into the SAS

  Right click on **House Sales Diagram** Window and select **Add Note** → **Sample** → **File Import**

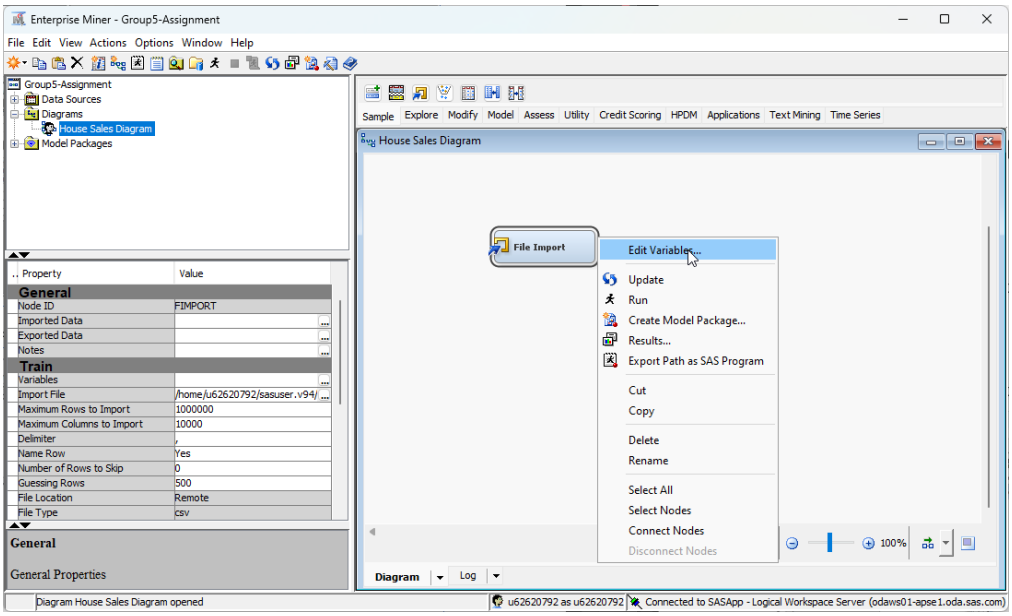


- Select **Import File** under Train on the left sidebar.

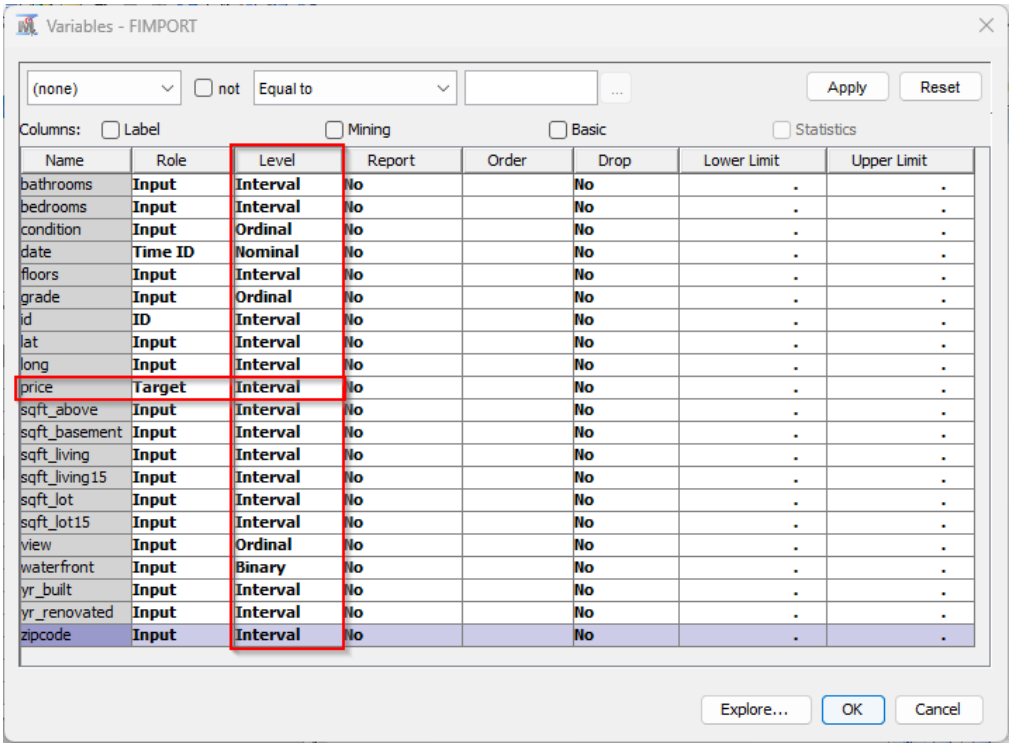
  Browse the dataset directory, and select **OK**



- Right-click the **File Import** node and select **Edit Variables…**

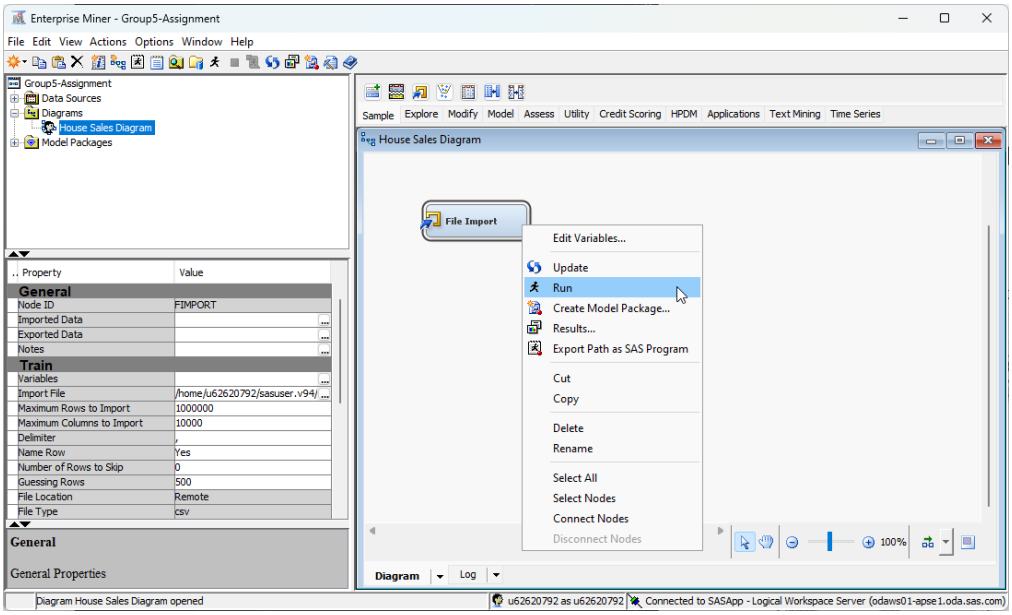


- Set *price* as the role of Target and update levels of the attributes because SAS Enterprise Miner will default to set all the numeric variables as interval and character variables as Nominal variables. After updating all attributes, select **OK**

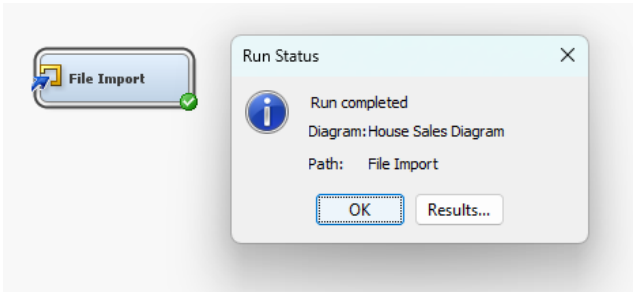


Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
bathrooms	Input	Interval	No		No	.	.
bedrooms	Input	Interval	No		No	.	.
condition	Input	Ordinal	No		No	.	.
date	Time ID	Nominal	No		No	.	.
floors	Input	Interval	No		No	.	.
grade	Input	Ordinal	No		No	.	.
id	ID	Interval	No		No	.	.
lat	Input	Interval	No		No	.	.
long	Input	Interval	No		No	.	.
price	Target	Interval	No		No	.	.
sqft_above	Input	Interval	No		No	.	.
sqft_basement	Input	Interval	No		No	.	.
sqft_living	Input	Interval	No		No	.	.
sqft_living15	Input	Interval	No		No	.	.
sqft_lot	Input	Interval	No		No	.	.
sqft_lot15	Input	Interval	No		No	.	.
view	Input	Ordinal	No		No	.	.
waterfront	Input	Binary	No		No	.	.
yr_built	Input	Interval	No		No	.	.
yr_renovated	Input	Interval	No		No	.	.
zipcode	Input	Interval	No		No	.	.

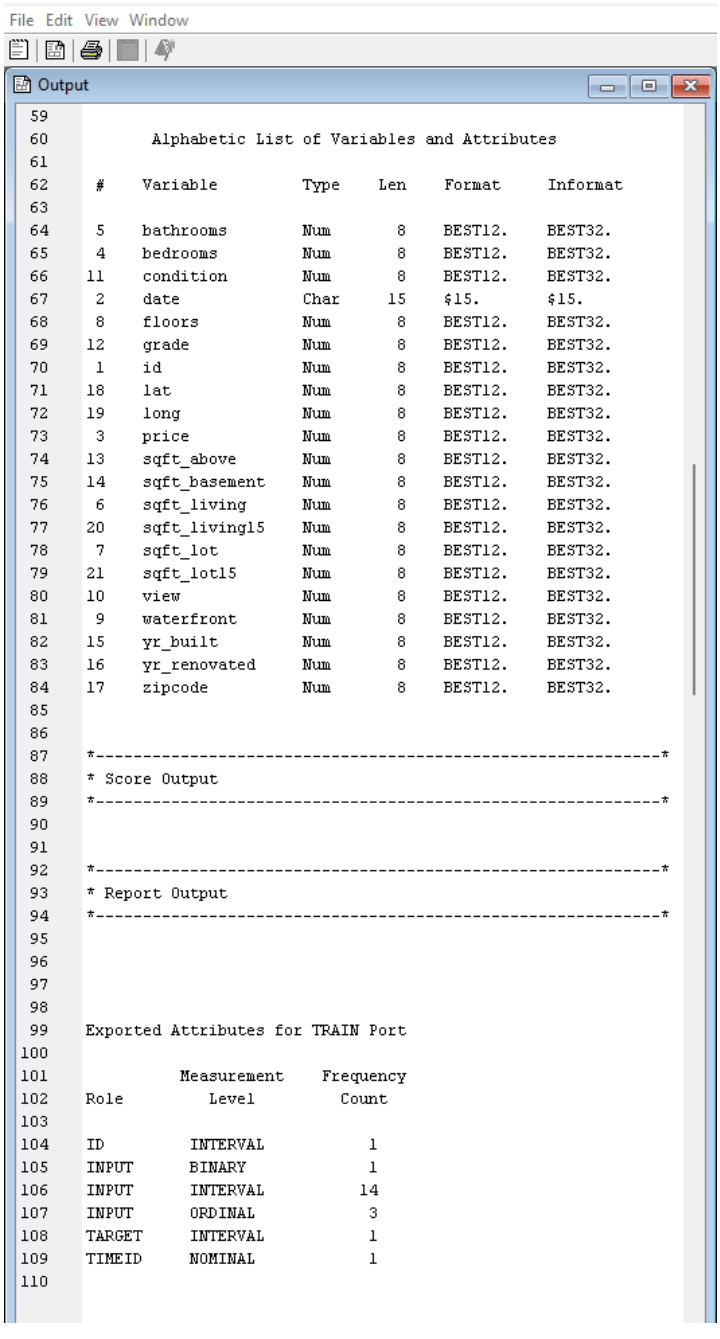
- Right-click the **File Import** node and select **Run**



- Once the run has completed, a green arrow is displayed on the **File Import** node, indicating that the file was imported successfully. Select **Results…**

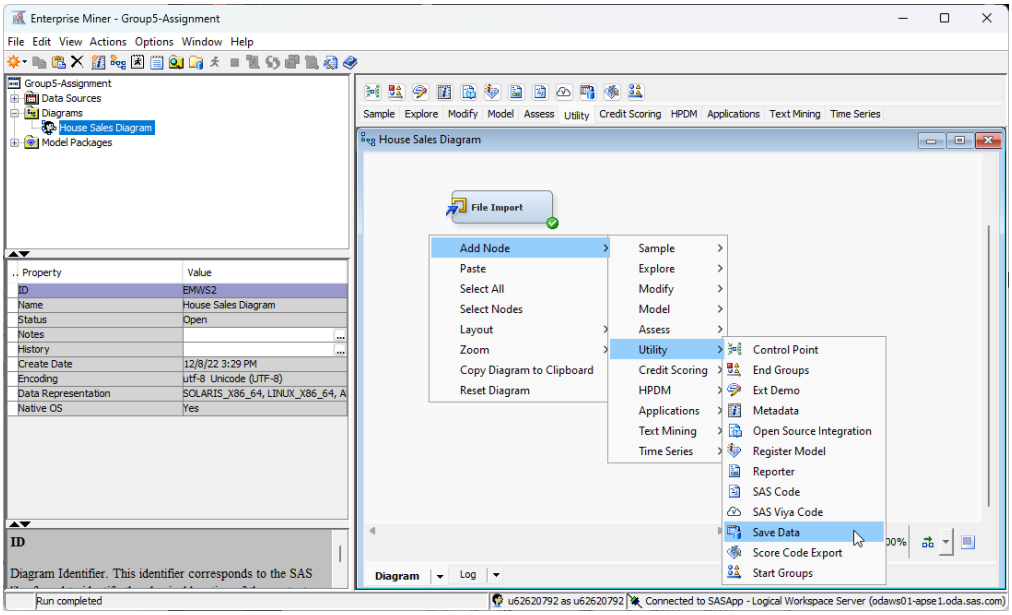


- **Results Window** shown. Once verified, close the window.

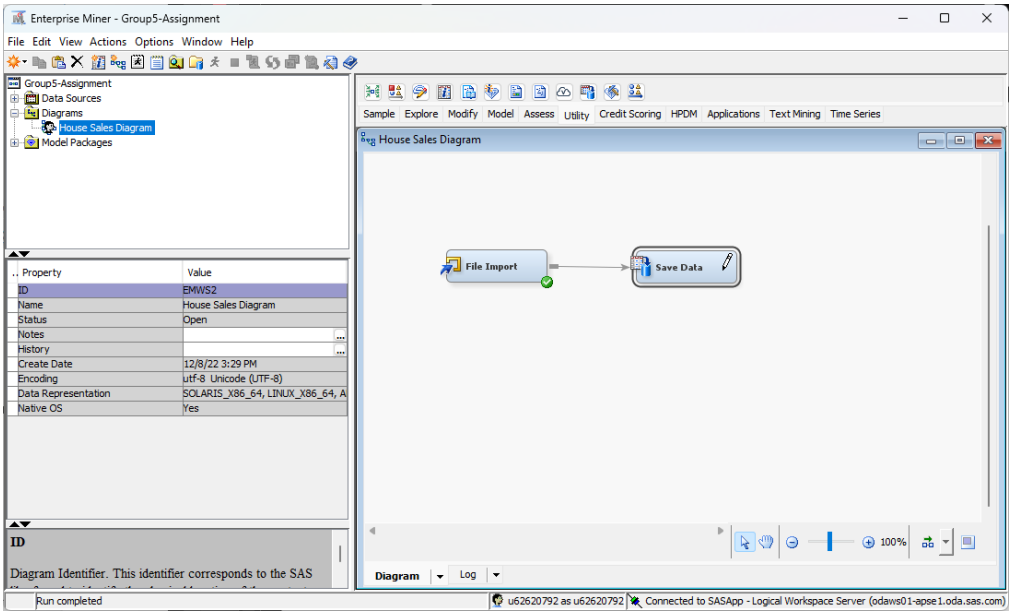



```
59
60                Alphabetic List of Variables and Attributes
61
62     #    Variable         Type    Len    Format     Informat
63
64     5    bathrooms        Num       8    BEST12.    BEST32.
65     4    bedrooms         Num       8    BEST12.    BEST32.
66    11    condition        Num       8    BEST12.    BEST32.
67     2    date             Char     15    $15.       $15.
68     8    floors           Num       8    BEST12.    BEST32.
69    12    grade            Num       8    BEST12.    BEST32.
70     1    id               Num       8    BEST12.    BEST32.
71    18    lat              Num       8    BEST12.    BEST32.
72    19    long             Num       8    BEST12.    BEST32.
73     3    price            Num       8    BEST12.    BEST32.
74    13    sqft_above       Num       8    BEST12.    BEST32.
75    14    sqft_basement    Num       8    BEST12.    BEST32.
76     6    sqft_living      Num       8    BEST12.    BEST32.
77    20    sqft_living15    Num       8    BEST12.    BEST32.
78     7    sqft_lot         Num       8    BEST12.    BEST32.
79    21    sqft_lot15       Num       8    BEST12.    BEST32.
80    10    view             Num       8    BEST12.    BEST32.
81     9    waterfront       Num       8    BEST12.    BEST32.
82    15    yr_built         Num       8    BEST12.    BEST32.
83    16    yr_renovated     Num       8    BEST12.    BEST32.
84    17    zipcode          Num       8    BEST12.    BEST32.
85
86
87    *------------------------------------------------------------*
88    * Score Output
89    *------------------------------------------------------------*
90
91
92    *------------------------------------------------------------*
93    * Report Output
94    *------------------------------------------------------------*
95
96
97
98
99    Exported Attributes for TRAIN Port
100
101              Measurement    Frequency
102   Role          Level         Count
103
104   ID          INTERVAL            1
105   INPUT       BINARY              1
106   INPUT       INTERVAL           14
107   INPUT       ORDINAL             3
108   TARGET      INTERVAL            1
109   TIMEID      NOMINAL             1
110
```

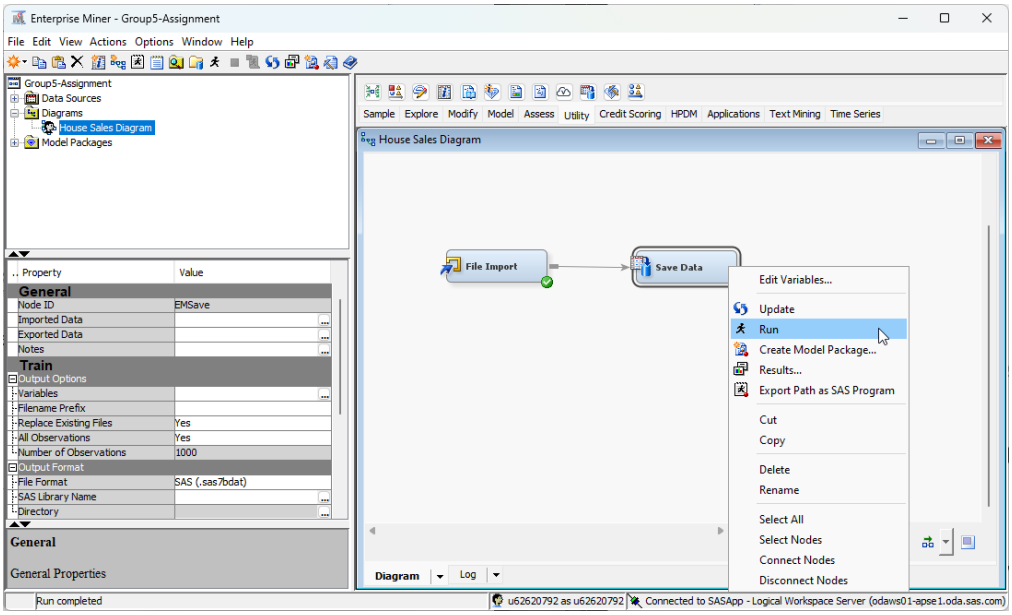
- **Right click** on **House Sales Diagram** Window and select **Add Note** → **Utility** → **Save Data**



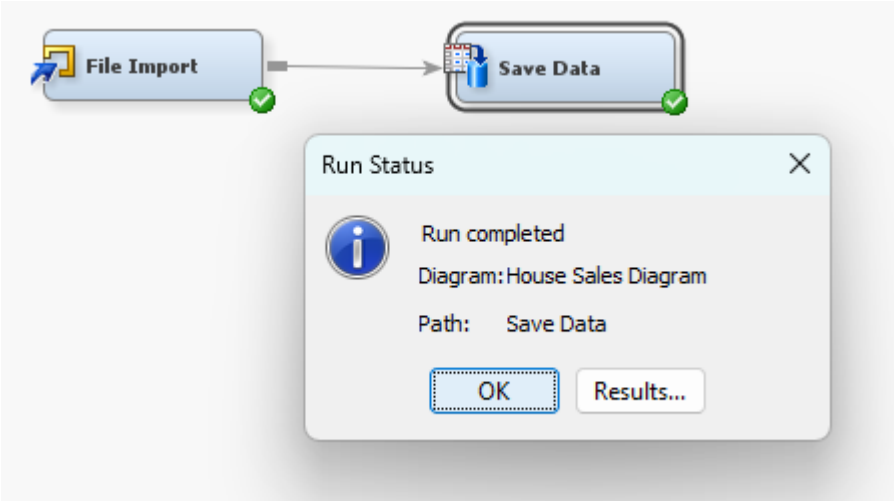
- Connect the Two Nodes



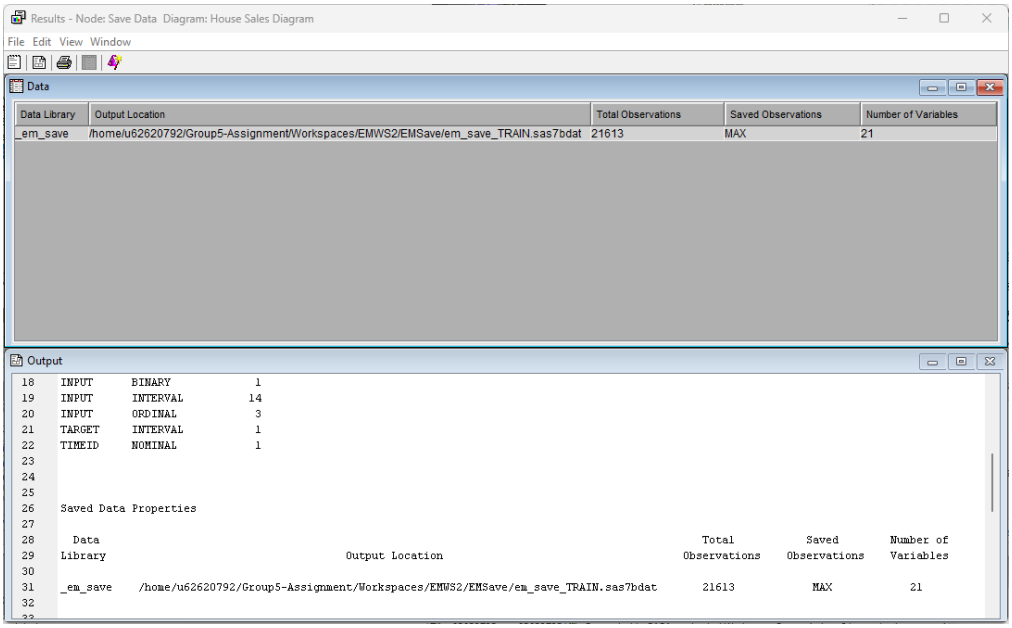
- Right-click the **Save Data** node and select **Run**



- Once the run has completed, a green arrow is displayed on the **Save Data** node, indicating that the file was imported successfully. Select **Results…**



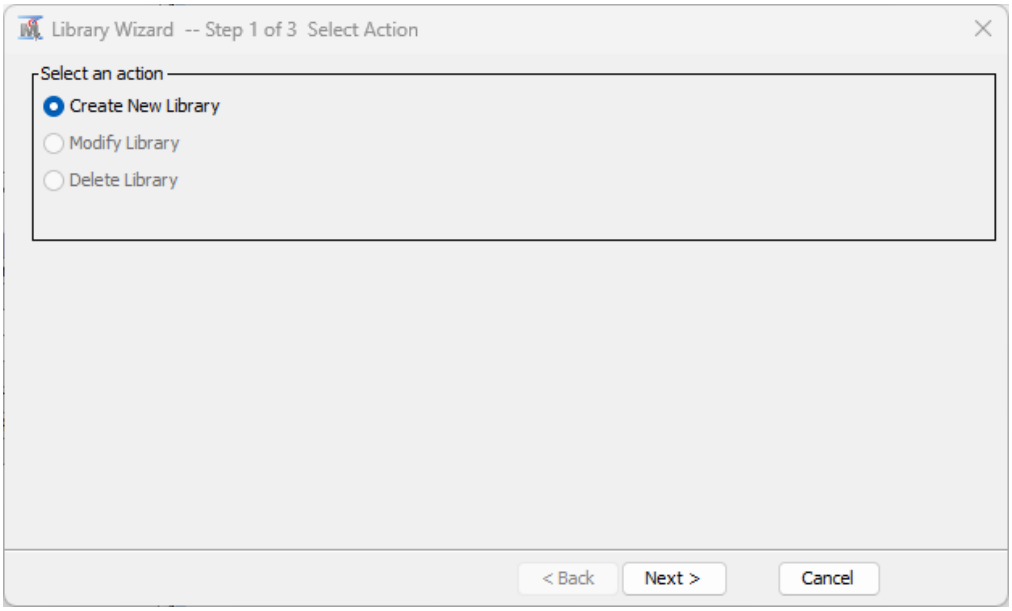
- **Results Window** shown. The **Output Location** of the .sas*7bdat* file is identified. Once verified, close the window.



7.1.3 Create a SAS Enterprise Miner Library

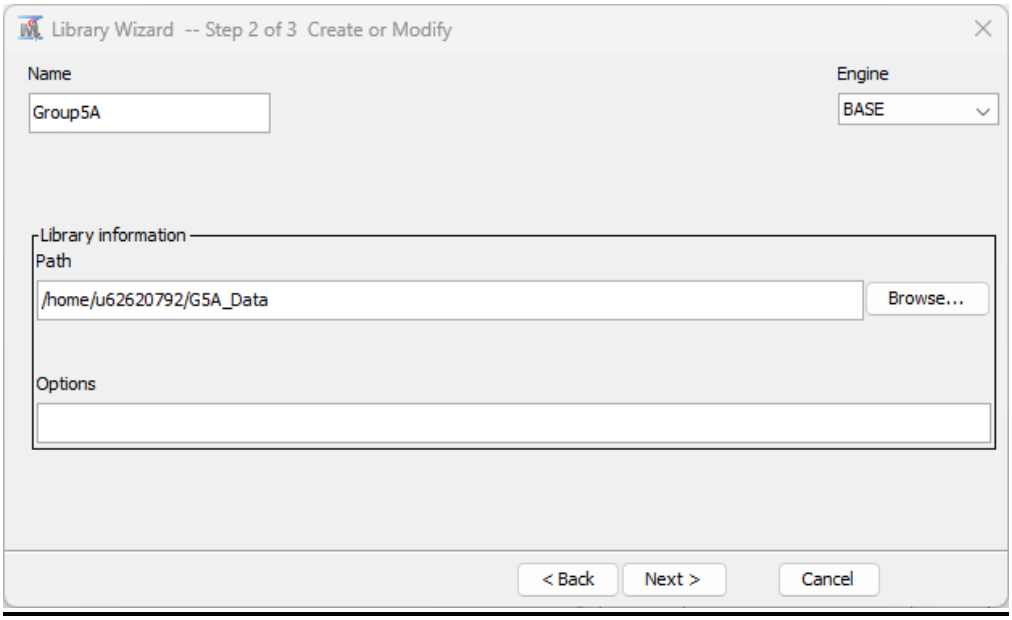
- Select **File** → **New** → **Library** from the Main Menu.

  Select the **Create New Library** and **Next >**

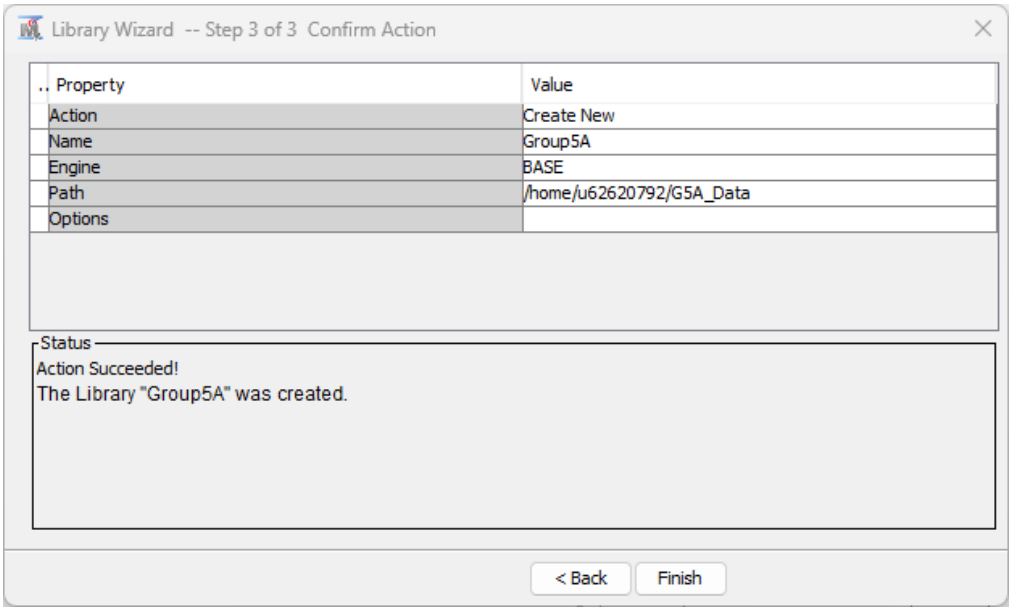


- Type the Library Name: **Group5A**. Select **Browse…** and input Library Path.

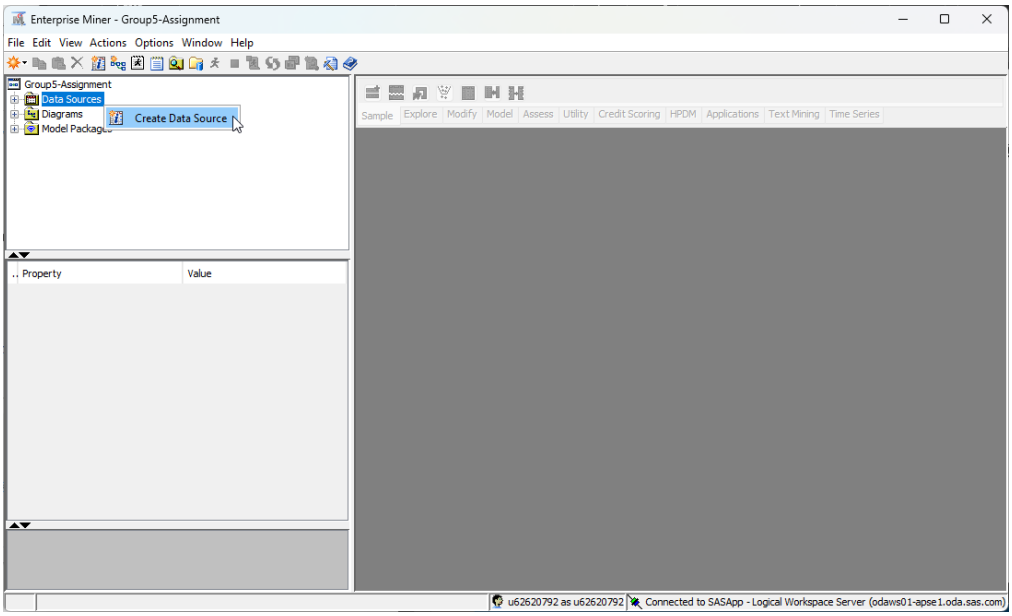
  Then select **Next >**



- The Library *Group5A* was created. Select **Finish**.

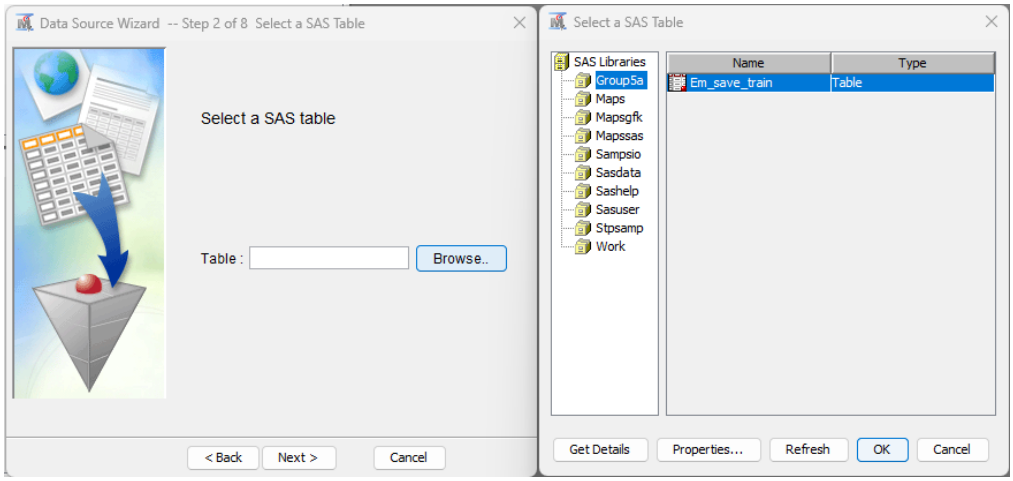


- Right-click the **Data Sources** and select **Create Data Source**.

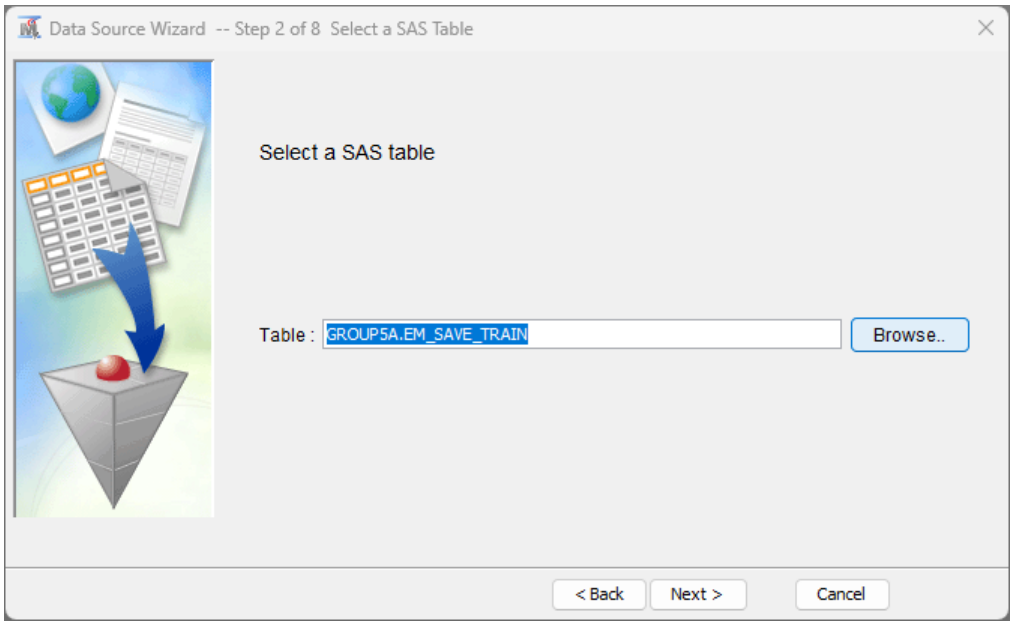


- Step 1 of 8 – Metadata Source. Select **SAS Table** as source and then, select **Next >**

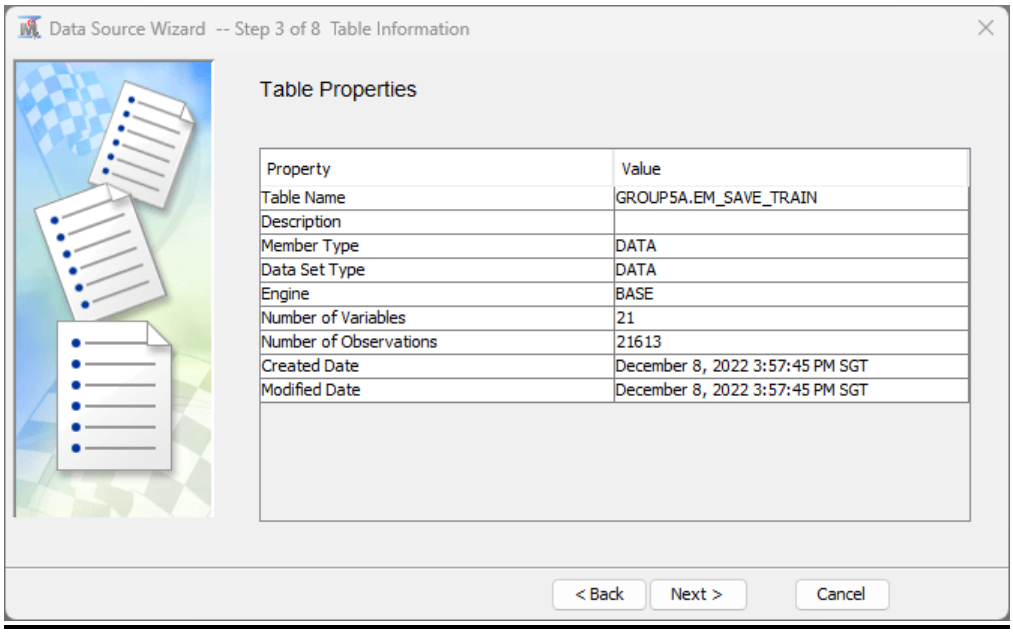
- Step 2 of 8 – Select a SAS Table. **Browse** the table and select target dataset. Then, Select **OK**



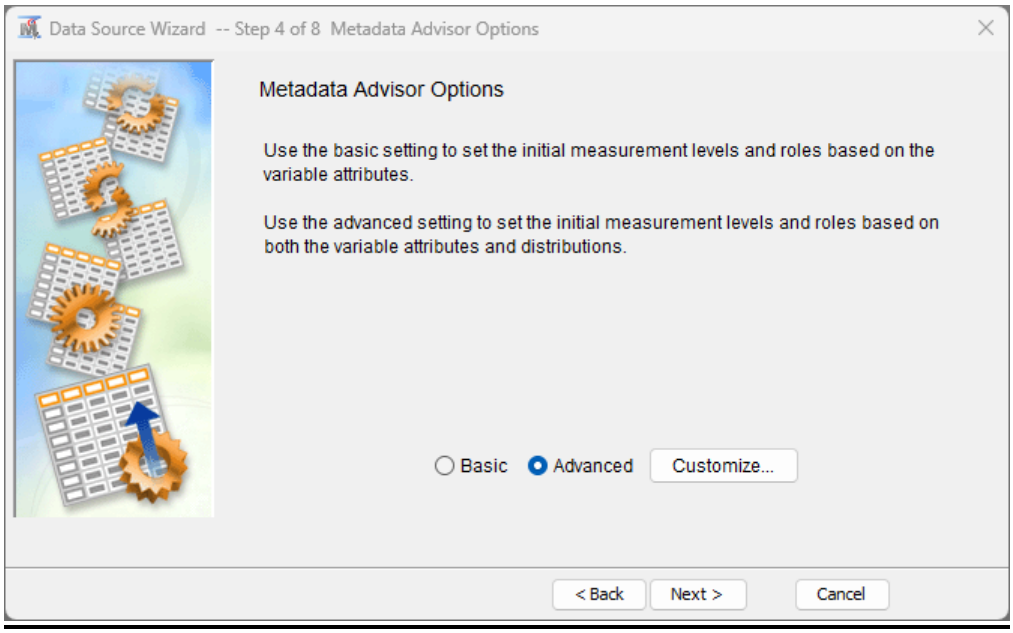
- Select **Next >**



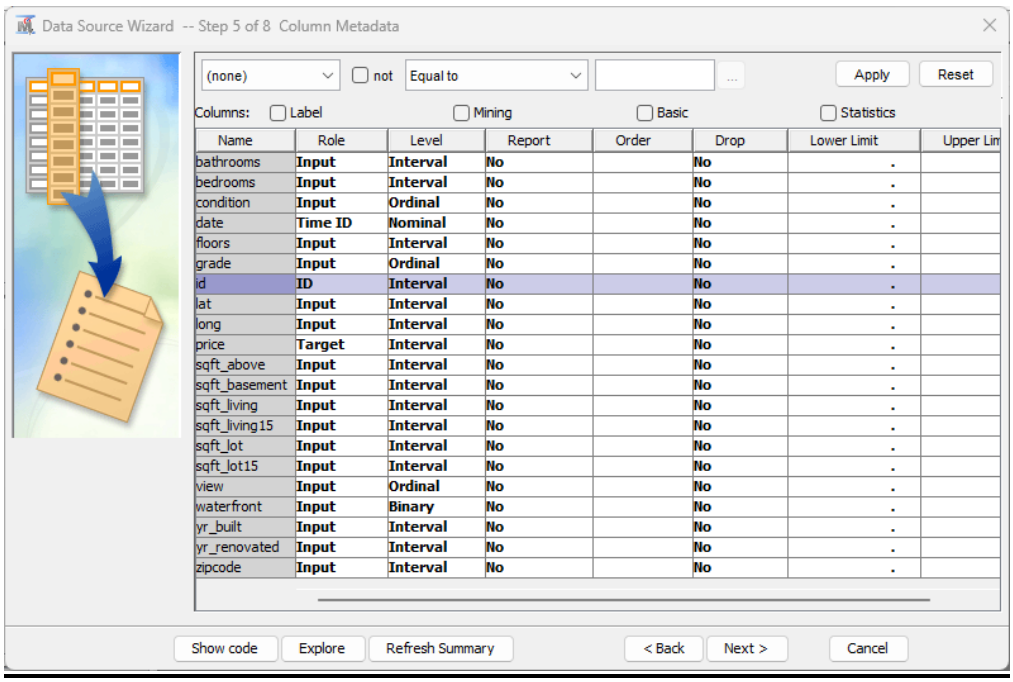
- Step 3 of 8 – Table Information. Select **Next >**



- Step 4 of 8 – Metadata Advisor. Select **Advanced** and **Next >**

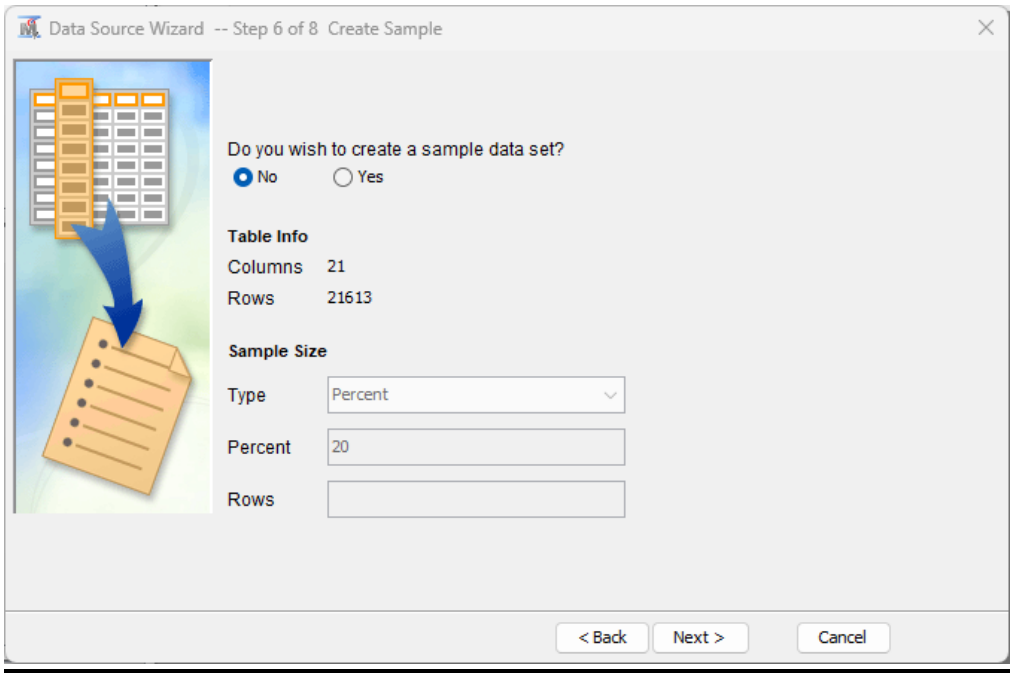


- Step 5 of 8 – Column Metadata. Update attributes on Roles and Levels. Select **Next >**

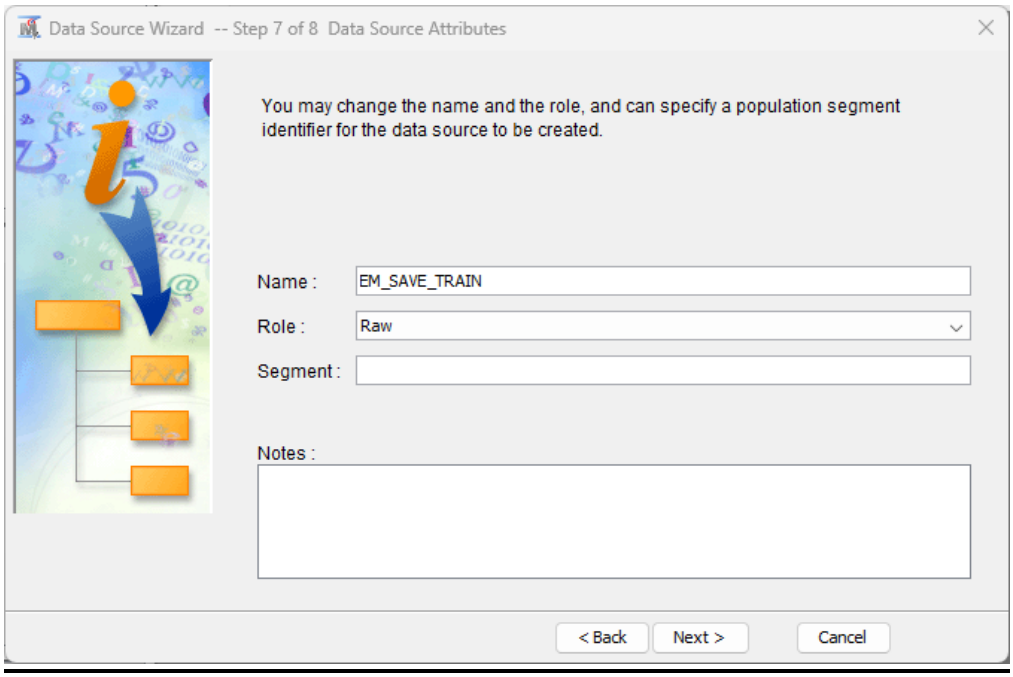


Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Lim
bathrooms	Input	Interval	No		No	.	
bedrooms	Input	Interval	No		No	.	
condition	Input	Ordinal	No		No	.	
date	Time ID	Nominal	No		No	.	
floors	Input	Interval	No		No	.	
grade	Input	Ordinal	No		No	.	
id	ID	Interval	No		No	.	
lat	Input	Interval	No		No	.	
long	Input	Interval	No		No	.	
price	Target	Interval	No		No	.	
sqft_above	Input	Interval	No		No	.	
sqft_basement	Input	Interval	No		No	.	
sqft_living	Input	Interval	No		No	.	
sqft_living15	Input	Interval	No		No	.	
sqft_lot	Input	Interval	No		No	.	
sqft_lot15	Input	Interval	No		No	.	
view	Input	Ordinal	No		No	.	
waterfront	Input	Binary	No		No	.	
yr_built	Input	Interval	No		No	.	
yr_renovated	Input	Interval	No		No	.	
zipcode	Input	Interval	No		No	.	

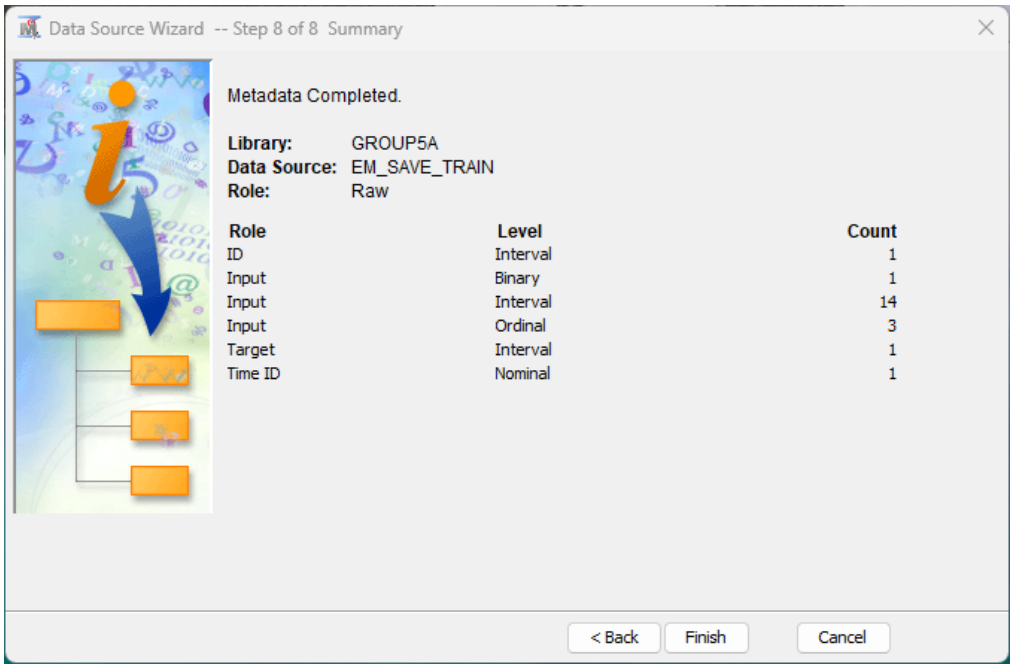
- Step 6 of 8 – Create Sample. Select **Next >**



- Step 7 of 8 – Data Source Attributes. Select **Next >**



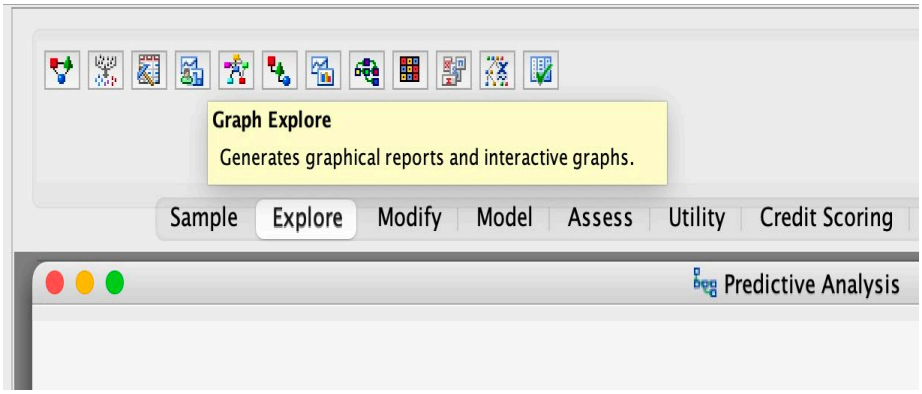
- Step 8 of 8 – Summary. Select **Finish**



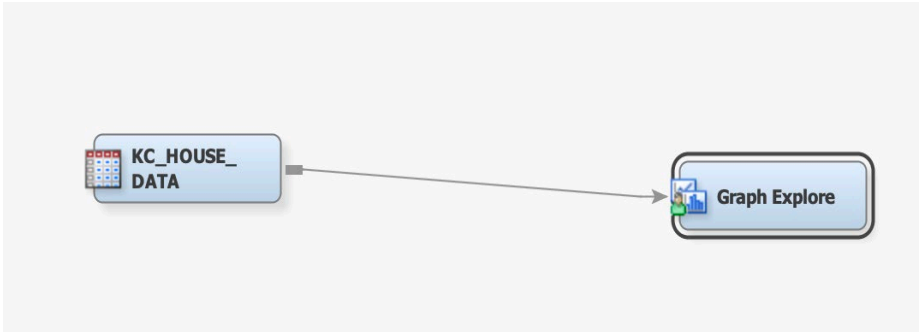
7.2 EXPLORE

7.2.1 Create Histogram, Pie Chart and Boxplot

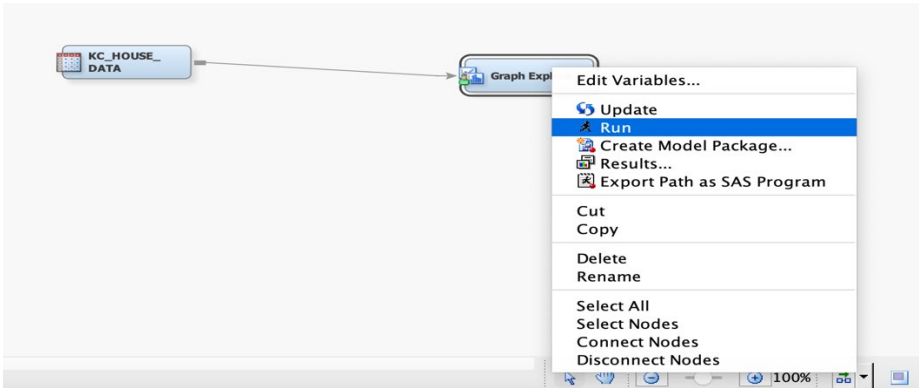
- Click Explore. Drag Graph Explore to the diagram



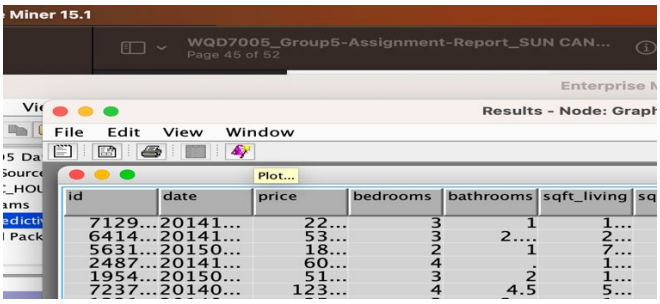
- Connect the Graph Explore Node to data source



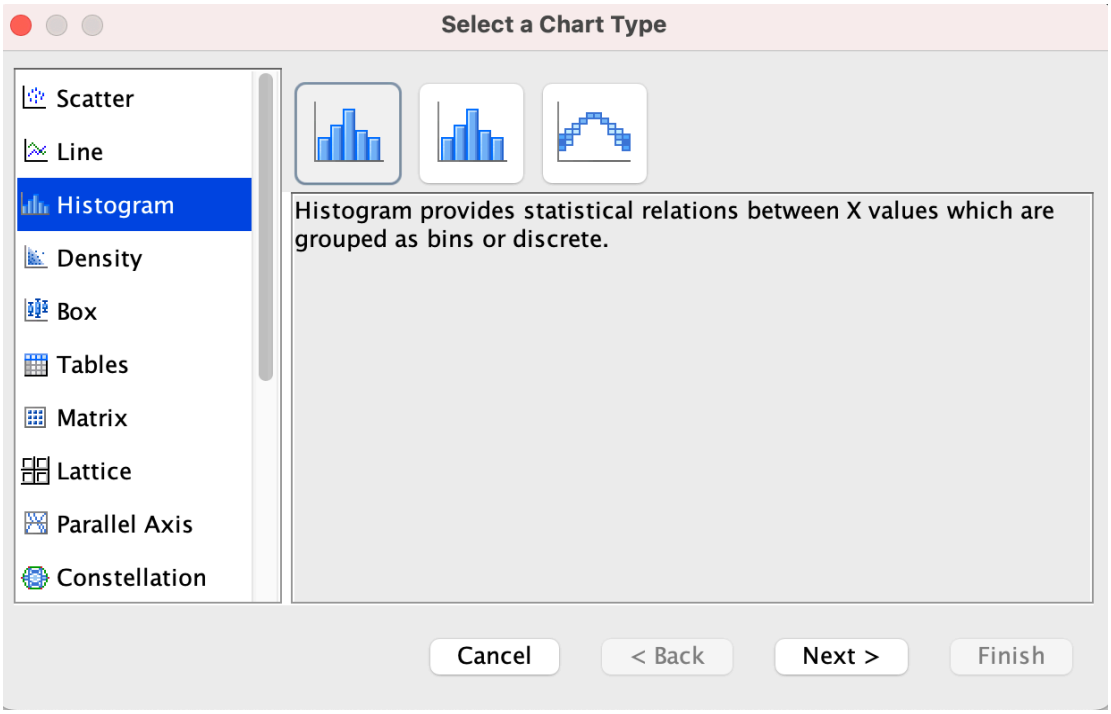
- Right click on Graph Explore Node, select Run and then Results



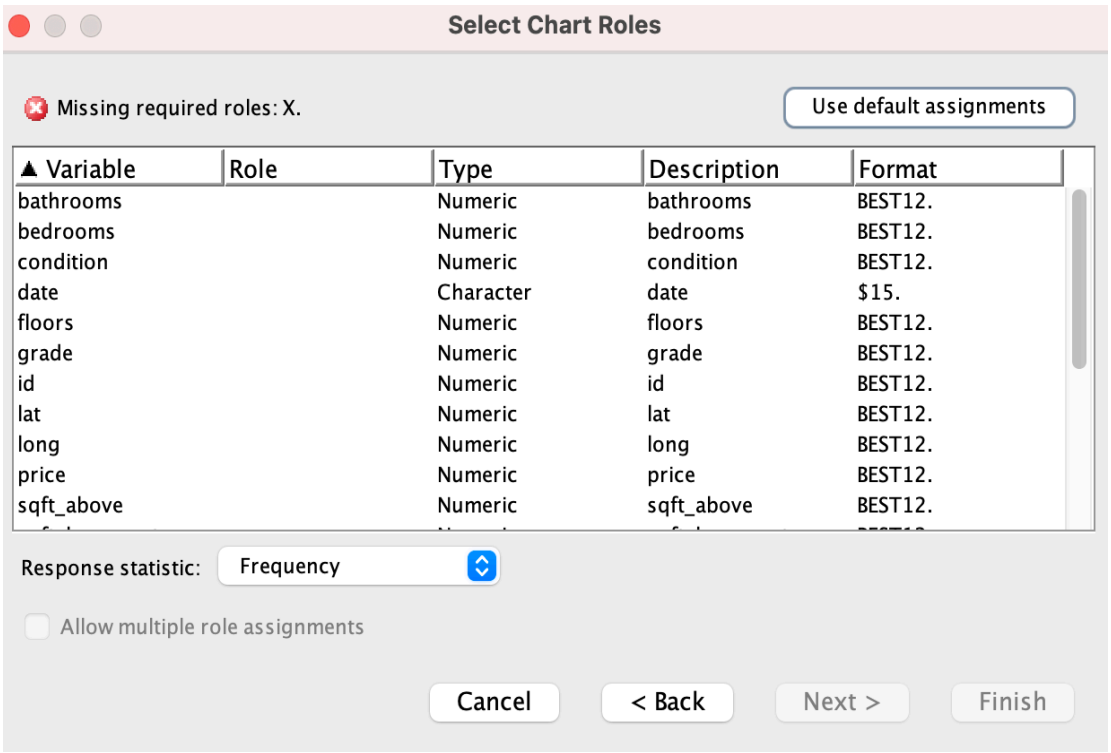
- Select Plot



- Select Histogram, Boxplot or Pie Chart

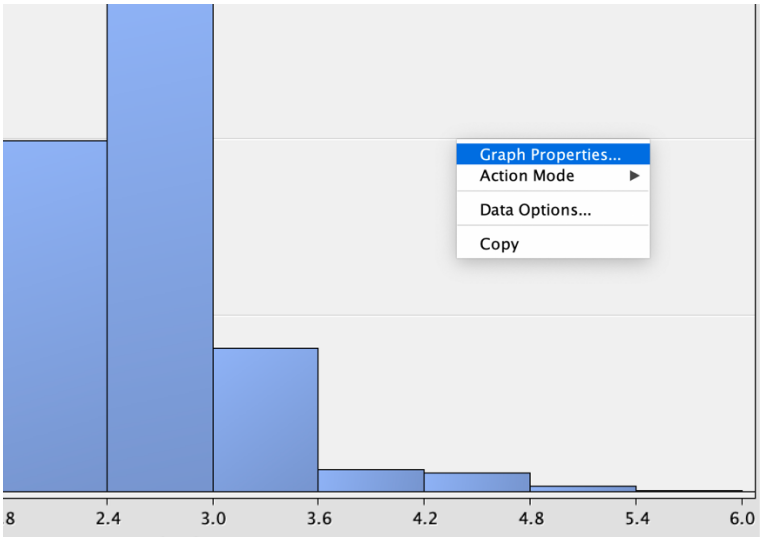


- Specify the variable and click Finish

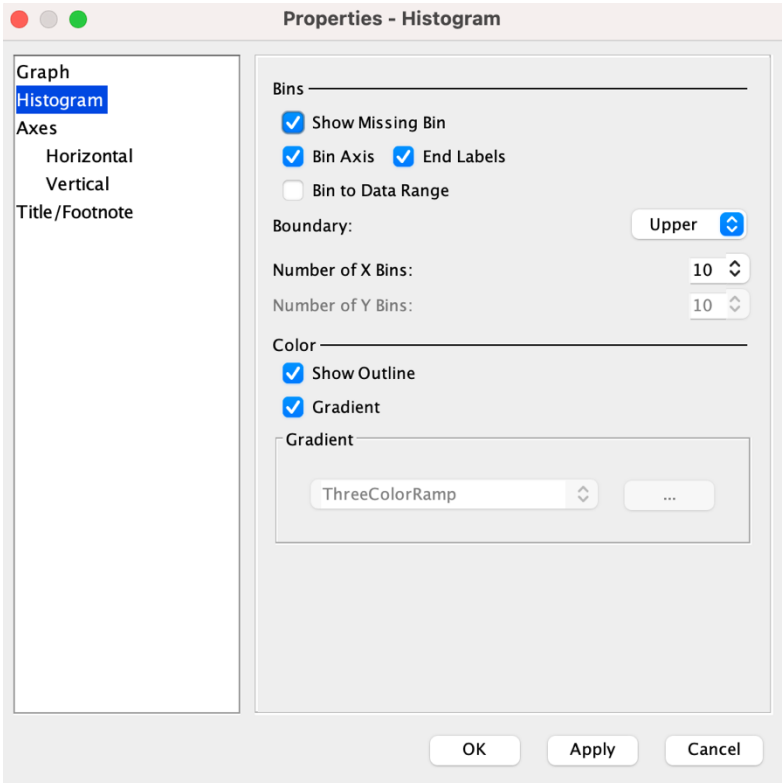


7.2.2 Add Missing Bin and Changing Graph Properties - Number of Bin in Histogram

- Right click on histogram and click graph properties

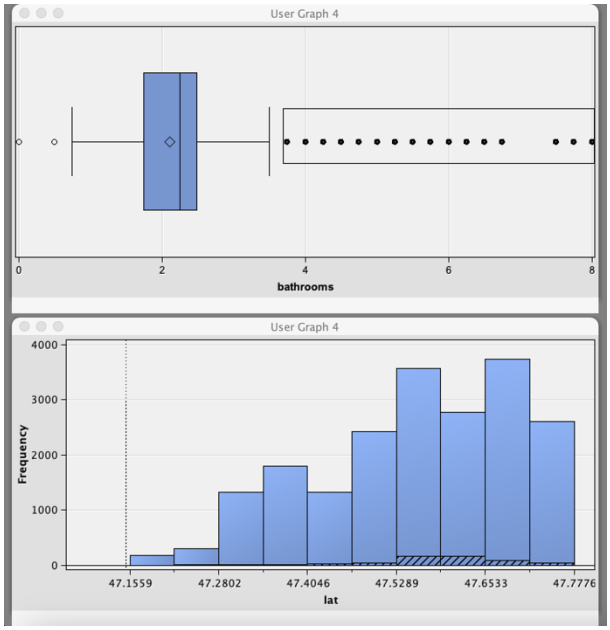


- Click “show missing bin” to display missing bin. Change the number of bin in histogram by changing the value of “Number of X Bins”.



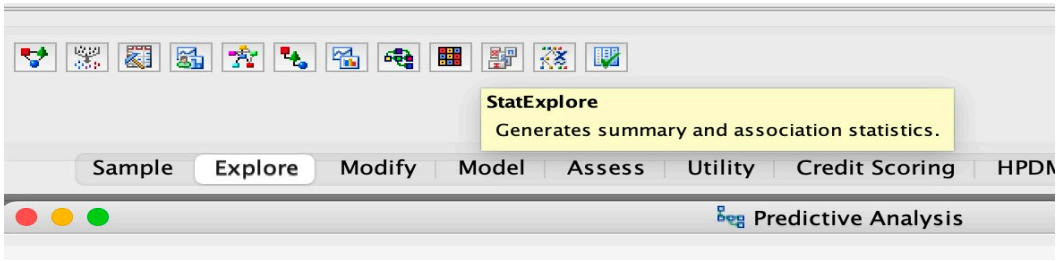
7.2.3 Display Variable Association

- Drag across the points on the diagram

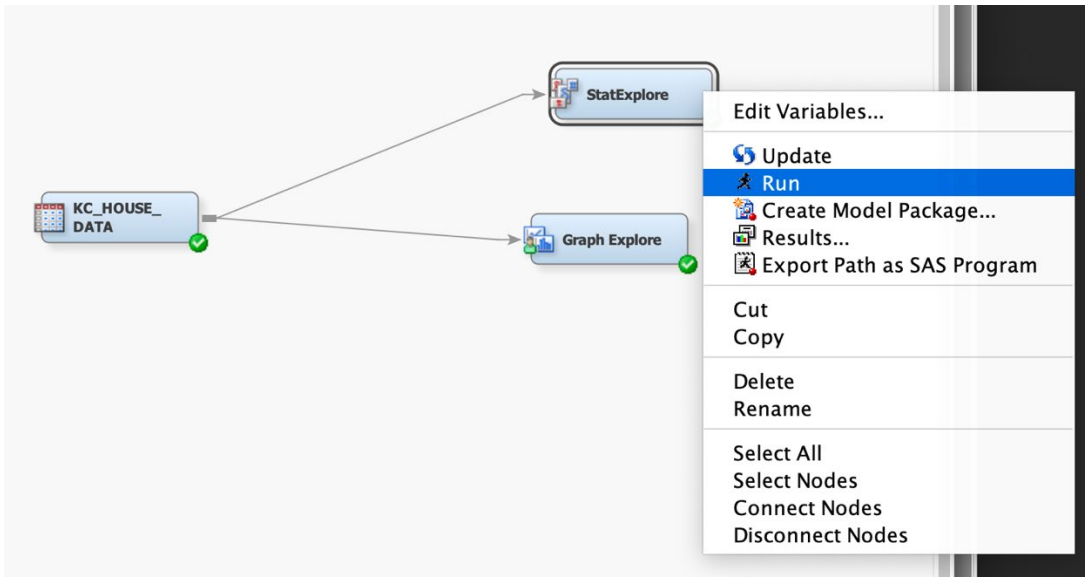


7.2.4 Create Summary Statistics

- Drag StatExplorer to the diagram

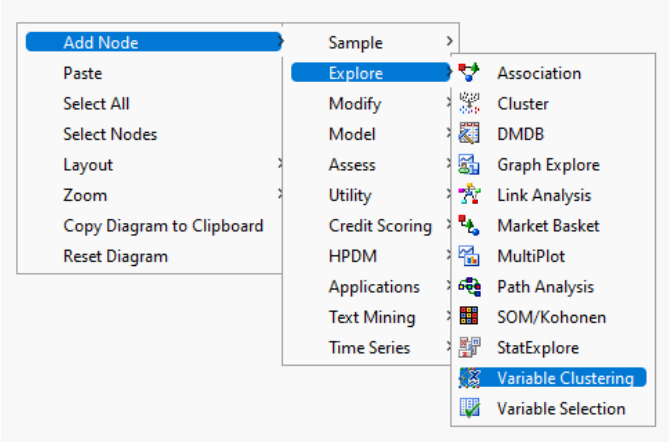


Connect the StatExplorer to the data source. Right click Run and Result

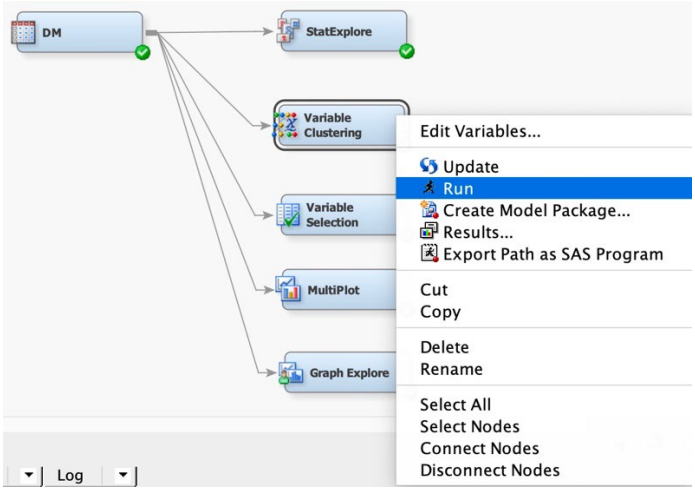


7.2.5 Perform Variable Clustering

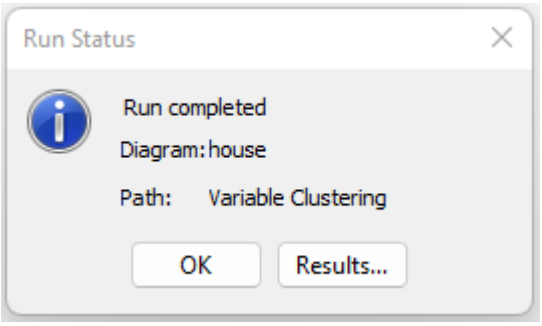
- Select **Add Node** → **Explore** → **Variable Clustering**.



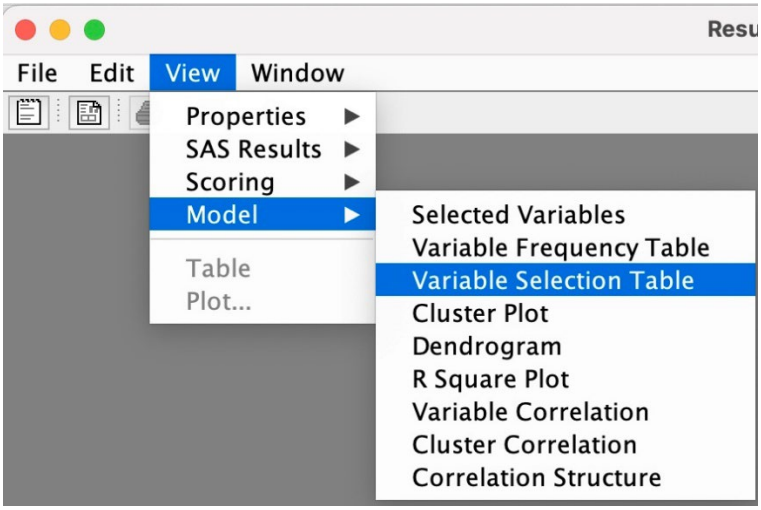
- Connect the data source and variable clustering and click **Run**.

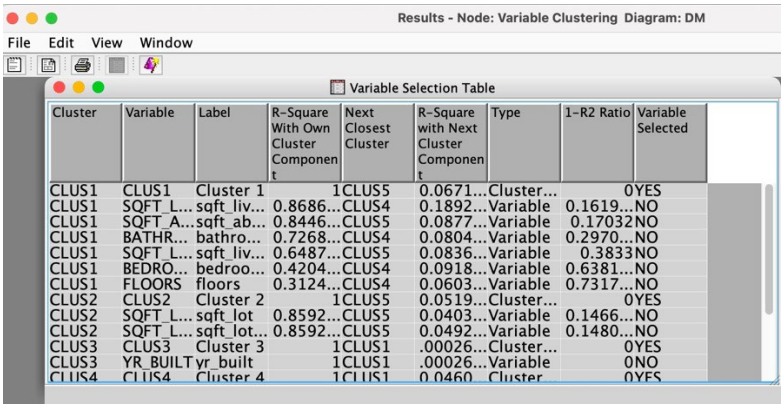


- Once the run completed, click **Results…**



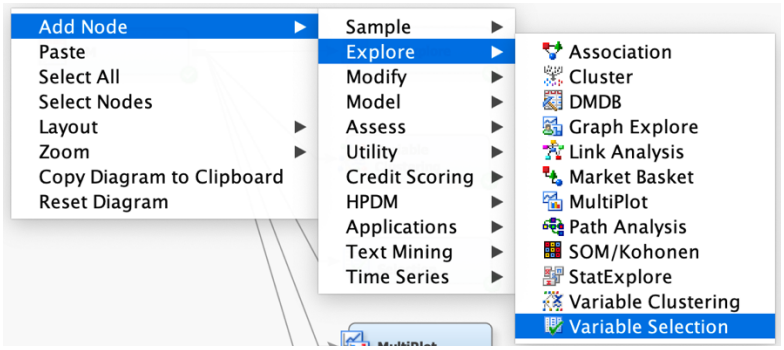
- Select **View**, **Model** and click on **Variable Selection Table**.



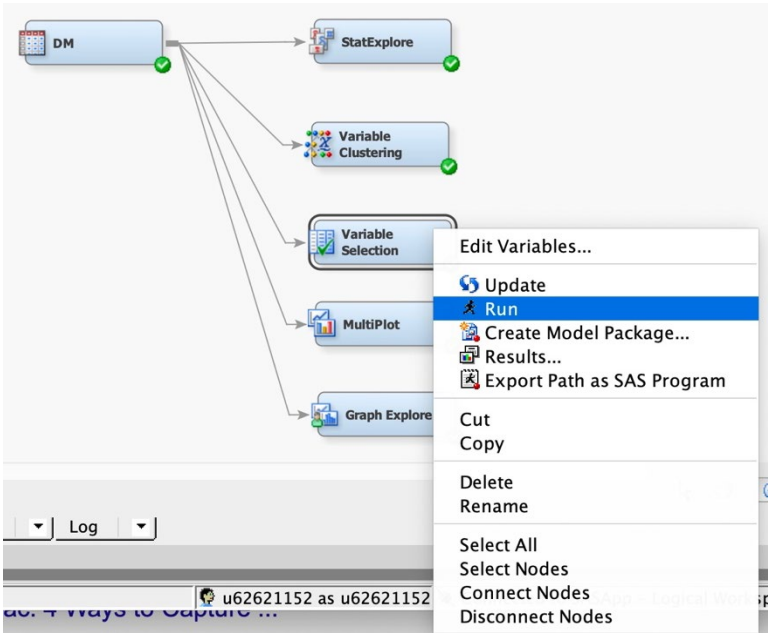


7.2.6 Perform Variable Selection

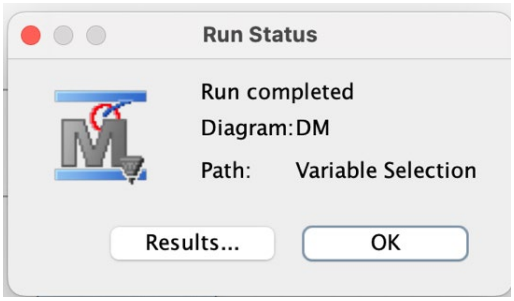
- Select **Add Node** → **Explore** → **Variable Selection.**



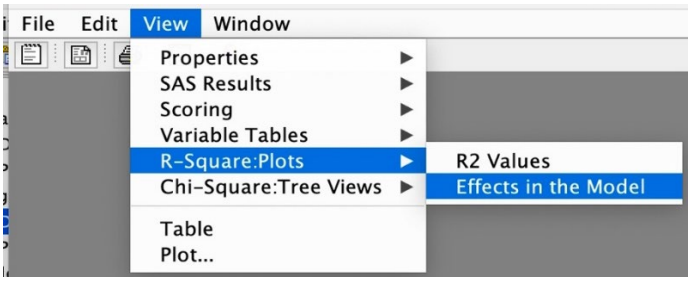
- Connect the data source and variable clustering and click **Run**.

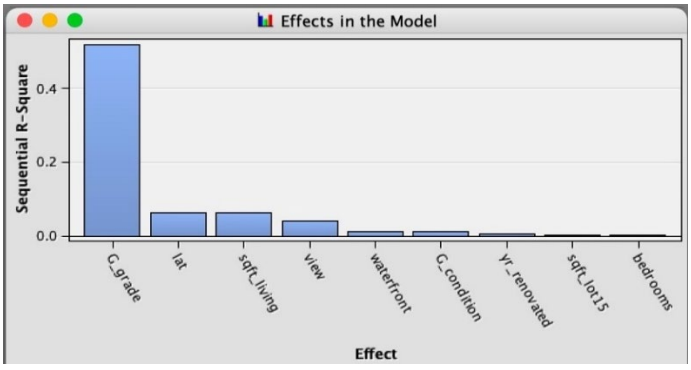


- Once the run completed, click **Results…**



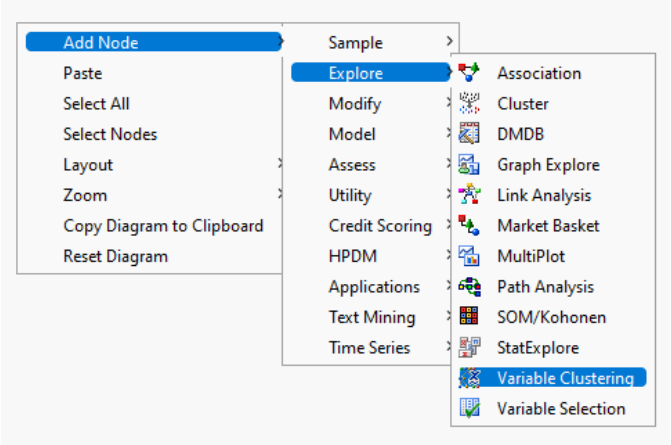
- Select **View**, **R-Square: Plots** and click on **Effects in the Model**.



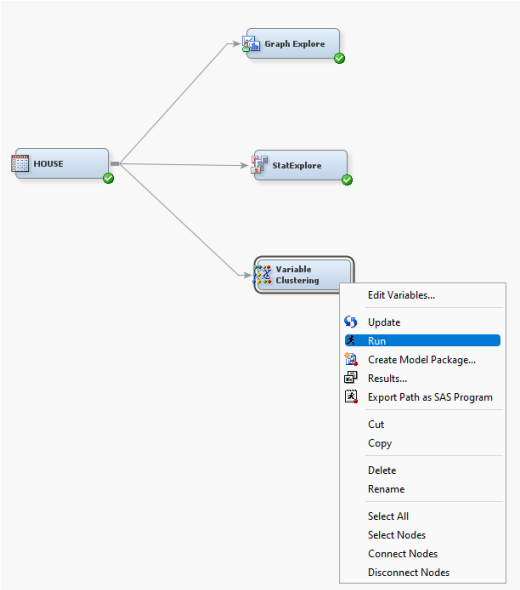


7.2.7 Perform Variable Correlation

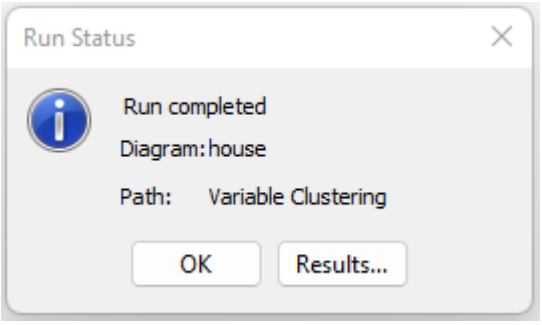
- Select **Add Node** → **Explore** → **Variable Clustering**.



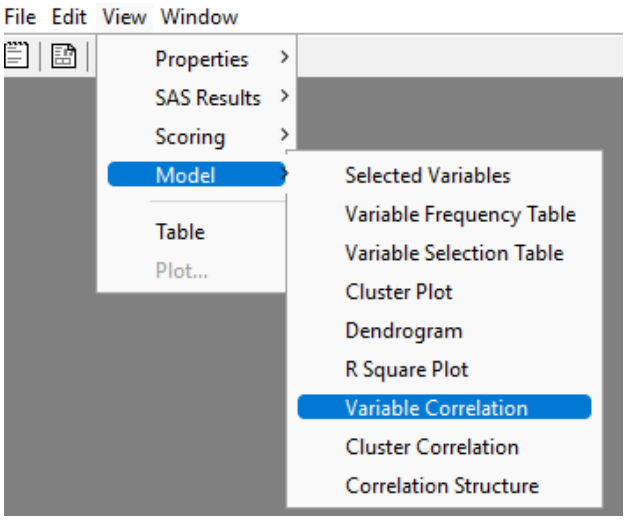
- Connect the data source and variable clustering and click **Run**.



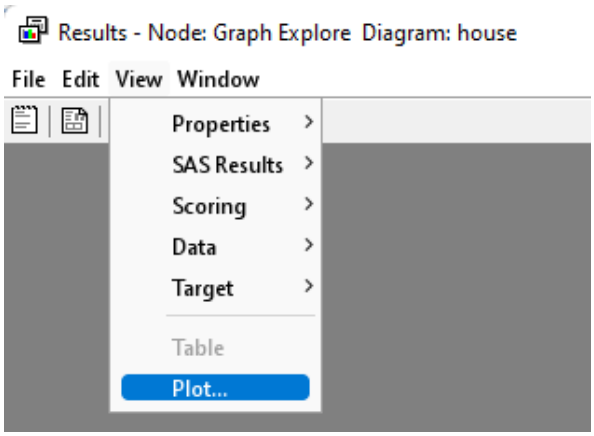
- Once the run completed, click **Results…**



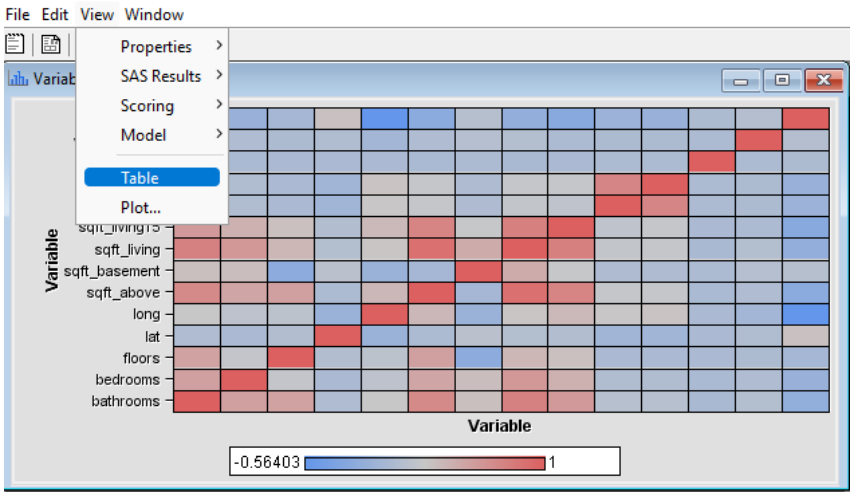
- Select **View**, **Model** and click on **Variable Correlation**.



- Select **View** and click on **Plot…**

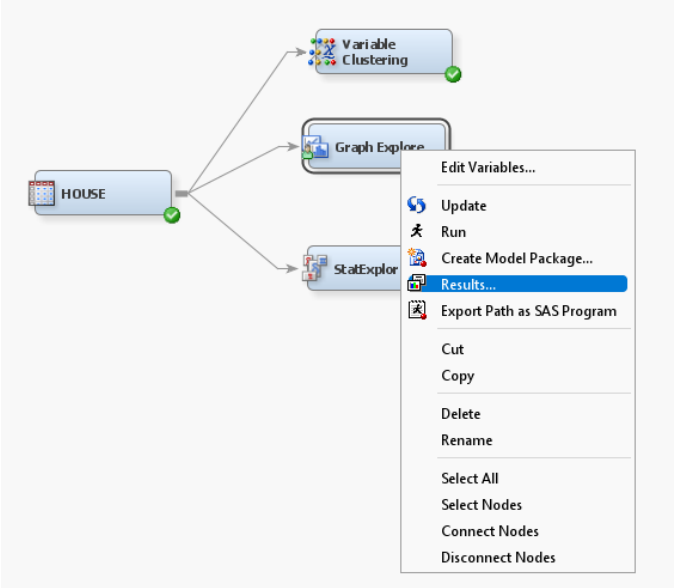


- Then, click on the correlation matrix plot before selecting **View** and click on **Table** to obtain correlation table.

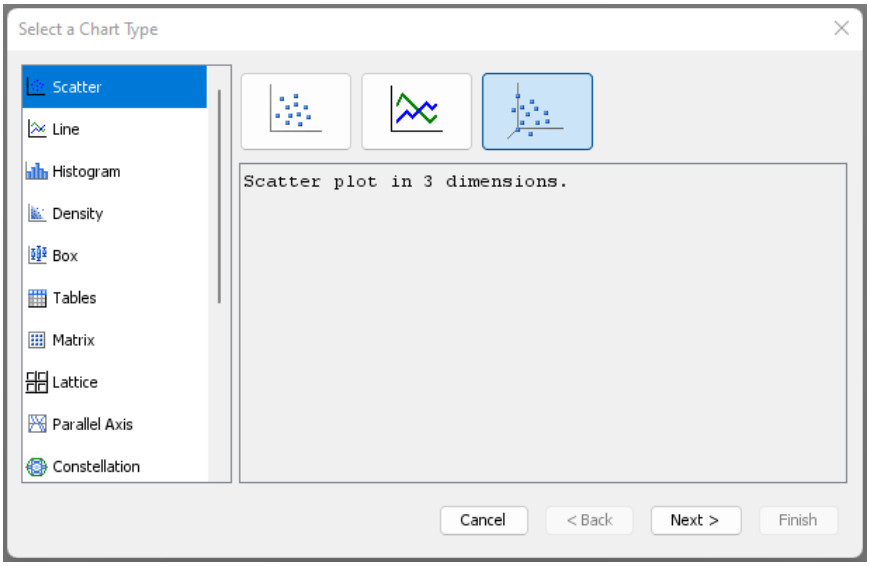


7.2.8 Interesting Visualizations

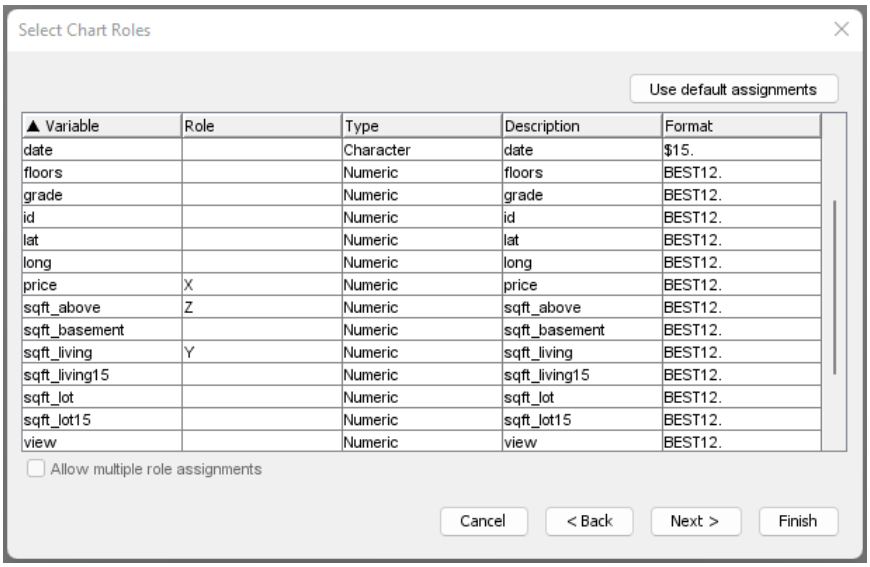
- Right-click the **Graph Explore** node and select **Results.**



- Select **3D Scatter** and click **Next**.



- Select at least 3 variables as X, Y and Z. Then, click **Finish**.



▲ Variable	Role	Type	Description	Format
date		Character	date	$15.
floors		Numeric	floors	BEST12.
grade		Numeric	grade	BEST12.
id		Numeric	id	BEST12.
lat		Numeric	lat	BEST12.
long		Numeric	long	BEST12.
price	X	Numeric	price	BEST12.
sqft_above	Z	Numeric	sqft_above	BEST12.
sqft_basement		Numeric	sqft_basement	BEST12.
sqft_living	Y	Numeric	sqft_living	BEST12.
sqft_living15		Numeric	sqft_living15	BEST12.
sqft_lot		Numeric	sqft_lot	BEST12.
sqft_lot15		Numeric	sqft_lot15	BEST12.
view		Numeric	view	BEST12.

7.2.9 Correlation Table

Variable	Variable	Correlation
bathrooms	bathrooms	1.0000
bedrooms	bathrooms	0.5159
floors	bathrooms	0.5007
lat	bathrooms	0.0246
long	bathrooms	0.2231
sqft_above	bathrooms	0.6854
sqft_basement	bathrooms	0.2837
sqft_living	bathrooms	0.7547
sqft_living15	bathrooms	0.5687
sqft_lot	bathrooms	0.0877
sqft_lot15	bathrooms	0.0872
yr_built	bathrooms	-0.0145
yr_renovated	bathrooms	0.0507
zipcode	bathrooms	-0.2039
bathrooms	bedrooms	0.5159
bedrooms	bedrooms	1.0000
floors	bedrooms	0.1754
lat	bedrooms	-0.0090
long	bedrooms	0.1295
sqft_above	bedrooms	0.4776
sqft_basement	bedrooms	0.3030
sqft_living	bedrooms	0.5766
sqft_living15	bedrooms	0.3916
sqft_lot	bedrooms	0.0317
sqft_lot15	bedrooms	0.0292
yr_built	bedrooms	-0.0131
yr_renovated	bedrooms	0.0188
zipcode	bedrooms	-0.1528
bathrooms	floors	0.5007
bedrooms	floors	0.1754
floors	floors	1.0000
lat	floors	0.0497
long	floors	0.1253
sqft_above	floors	0.5238
sqft_basement	floors	-0.2457
sqft_living	floors	0.3539
sqft_living15	floors	0.2798
sqft_lot	floors	-0.0052
sqft_lot15	floors	-0.0113
yr_built	floors	-0.0108

yr_renovated	floors	0.0063
zipcode	floors	-0.0591
bathrooms	lat	0.0246
bedrooms	lat	-0.0090
floors	lat	0.0497
lat	lat	1.0000
long	lat	-0.1355
sqft_above	lat	-0.0008
sqft_basement	lat	0.1105
sqft_living	lat	0.0526
sqft_living15	lat	0.0488
sqft_lot	lat	-0.0857
sqft_lot15	lat	-0.0864
yr_built	lat	-0.0106
yr_renovated	lat	0.0294
zipcode	lat	0.2670
bathrooms	long	0.2231
bedrooms	long	0.1295
floors	long	0.1253
lat	long	-0.1355
long	long	1.0000
sqft_above	long	0.3438
sqft_basement	long	-0.1447
sqft_living	long	0.2402
sqft_living15	long	0.3346
sqft_lot	long	0.2295
sqft_lot15	long	0.2544
yr_built	long	-0.0092
yr_renovated	long	-0.0684
zipcode	long	-0.5640
bathrooms	sqft_above	0.6854
bedrooms	sqft_above	0.4776
floors	sqft_above	0.5238
lat	sqft_above	-0.0008
long	sqft_above	0.3438
sqft_above	sqft_above	1.0000
sqft_basement	sqft_above	-0.0519
sqft_living	sqft_above	0.8766
sqft_living15	sqft_above	0.7319
sqft_lot	sqft_above	0.1835
sqft_lot15	sqft_above	0.1940
yr_built	sqft_above	-0.0134

yr_renovated	sqft_above	0.0232
zipcode	sqft_above	-0.2612
bathrooms	sqft_basement	0.2837
bedrooms	sqft_basement	0.3030
floors	sqft_basement	-0.2457
lat	sqft_basement	0.1105
long	sqft_basement	-0.1447
sqft_above	sqft_basement	-0.0519
sqft_basement	sqft_basement	1.0000
sqft_living	sqft_basement	0.4351
sqft_living15	sqft_basement	0.2004
sqft_lot	sqft_basement	0.0153
sqft_lot15	sqft_basement	0.0173
yr_built	sqft_basement	-0.0043
yr_renovated	sqft_basement	0.0713
zipcode	sqft_basement	0.0748
bathrooms	sqft_living	0.7547
bedrooms	sqft_living	0.5766
floors	sqft_living	0.3539
lat	sqft_living	0.0526
long	sqft_living	0.2402
sqft_above	sqft_living	0.8766
sqft_basement	sqft_living	0.4351
sqft_living	sqft_living	1.0000
sqft_living15	sqft_living	0.7564
sqft_lot	sqft_living	0.1728
sqft_lot15	sqft_living	0.1833
yr_built	sqft_living	-0.0141
yr_renovated	sqft_living	0.0553
zipcode	sqft_living	-0.1995
bathrooms	sqft_living15	0.5687
bedrooms	sqft_living15	0.3916
floors	sqft_living15	0.2798
lat	sqft_living15	0.0488
long	sqft_living15	0.3346
sqft_above	sqft_living15	0.7319
sqft_basement	sqft_living15	0.2004
sqft_living	sqft_living15	0.7564
sqft_living15	sqft_living15	1.0000
sqft_lot	sqft_living15	0.1446
sqft_lot15	sqft_living15	0.1832
yr_built	sqft_living15	-0.0119

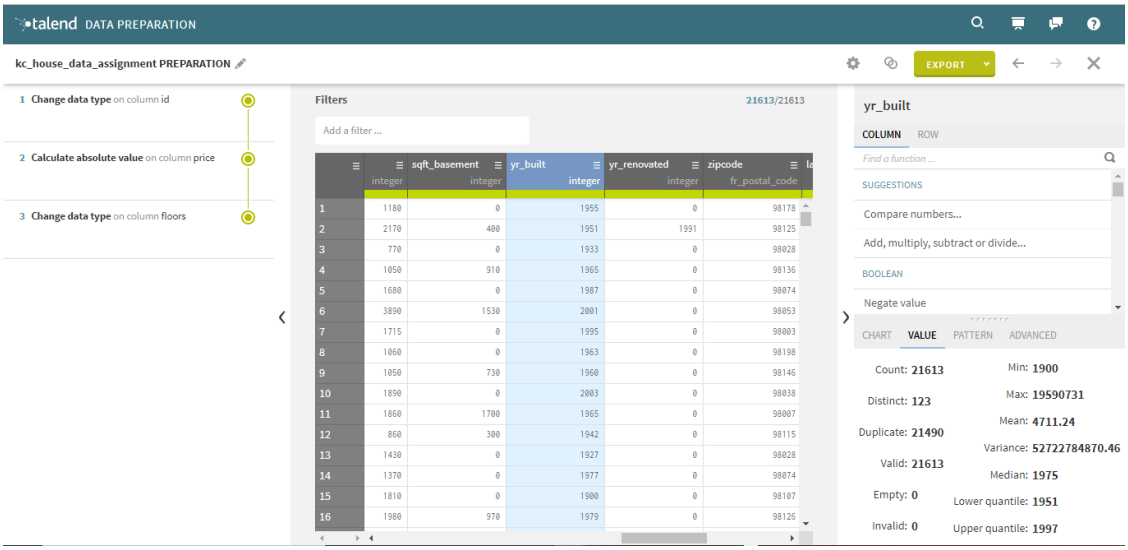
yr_renovated	sqft_living15	-0.0027
zipcode	sqft_living15	-0.2790
bathrooms	sqft_lot	0.0877
bedrooms	sqft_lot	0.0317
floors	sqft_lot	-0.0052
lat	sqft_lot	-0.0857
long	sqft_lot	0.2295
sqft_above	sqft_lot	0.1835
sqft_basement	sqft_lot	0.0153
sqft_living	sqft_lot	0.1728
sqft_living15	sqft_lot	0.1446
sqft_lot	sqft_lot	1.0000
sqft_lot15	sqft_lot	0.7186
yr_built	sqft_lot	-0.0012
yr_renovated	sqft_lot	0.0076
zipcode	sqft_lot	-0.1296
bathrooms	sqft_lot15	0.0872
bedrooms	sqft_lot15	0.0292
floors	sqft_lot15	-0.0113
lat	sqft_lot15	-0.0864
long	sqft_lot15	0.2544
sqft_above	sqft_lot15	0.1940
sqft_basement	sqft_lot15	0.0173
sqft_living	sqft_lot15	0.1833
sqft_living15	sqft_lot15	0.1832
sqft_lot	sqft_lot15	0.7186
sqft_lot15	sqft_lot15	1.0000
yr_built	sqft_lot15	-0.0017
yr_renovated	sqft_lot15	0.0078
zipcode	sqft_lot15	-0.1472
bathrooms	yr_built	-0.0145
bedrooms	yr_built	-0.0131
floors	yr_built	-0.0108
lat	yr_built	-0.0106
long	yr_built	-0.0092
sqft_above	yr_built	-0.0134
sqft_basement	yr_built	-0.0043
sqft_living	yr_built	-0.0141
sqft_living15	yr_built	-0.0119
sqft_lot	yr_built	-0.0012
sqft_lot15	yr_built	-0.0017
yr_built	yr_built	1.0000

yr_renovated	yr_built	-0.0023
zipcode	yr_built	0.0092
bathrooms	yr_renovated	0.0507
bedrooms	yr_renovated	0.0188
floors	yr_renovated	0.0063
lat	yr_renovated	0.0294
long	yr_renovated	-0.0684
sqft_above	yr_renovated	0.0232
sqft_basement	yr_renovated	0.0713
sqft_living	yr_renovated	0.0553
sqft_living15	yr_renovated	-0.0027
sqft_lot	yr_renovated	0.0076
sqft_lot15	yr_renovated	0.0078
yr_built	yr_renovated	-0.0023
yr_renovated	yr_renovated	1.0000
zipcode	yr_renovated	0.0644
bathrooms	zipcode	-0.2039
bedrooms	zipcode	-0.1528
floors	zipcode	-0.0591
lat	zipcode	0.2670
long	zipcode	-0.5640
sqft_above	zipcode	-0.2612
sqft_basement	zipcode	0.0748
sqft_living	zipcode	-0.1995
sqft_living15	zipcode	-0.2790
sqft_lot	zipcode	-0.1296
sqft_lot15	zipcode	-0.1472
yr_built	zipcode	0.0092
yr_renovated	zipcode	0.0644
zipcode	zipcode	1.0000

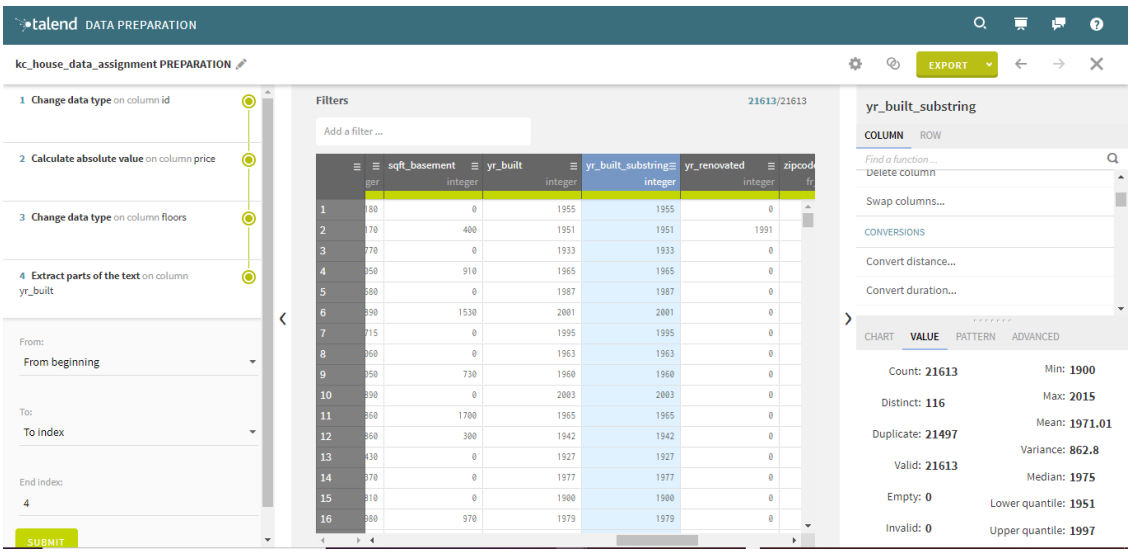
7.3 MODIFY

7.3.1 Modify Inconsistent Data

- In Talend, go to yr_built variable



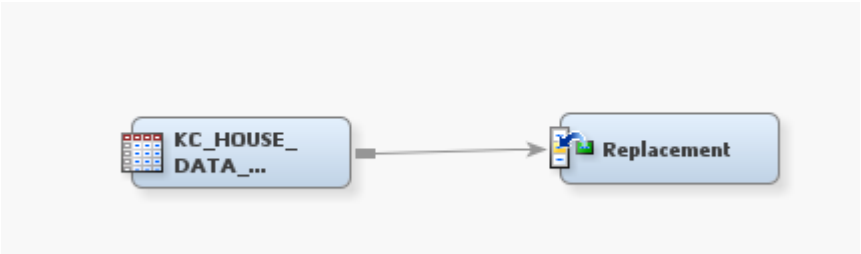
- Apply Extract parts of the text function on yr_built and set the End Index to 4, yr_built substring will be generated, with all the data are consistent



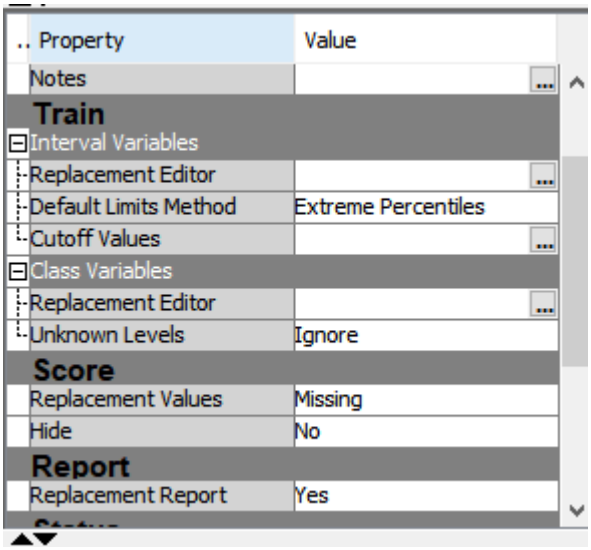
- Remove yr_built column and rename yr_built_substring column to yr_built column

7.3.2 Modify Noisy Data and Incomplete Data

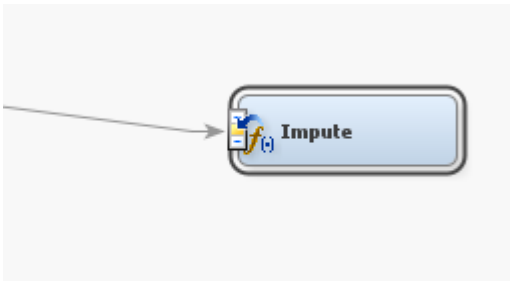
- Add Replacement Node



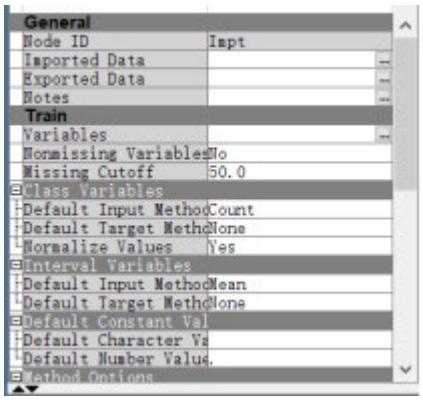
- Select Interval Variable >> Replacement Editor >> Variables that have noisy data >> Yes
- Select Default Limits Method >> Extreme Percentiles
- Select Score >> Replacement Values >> Missing



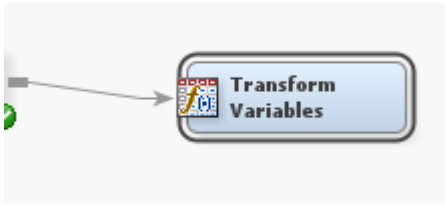
- Add Impute Node



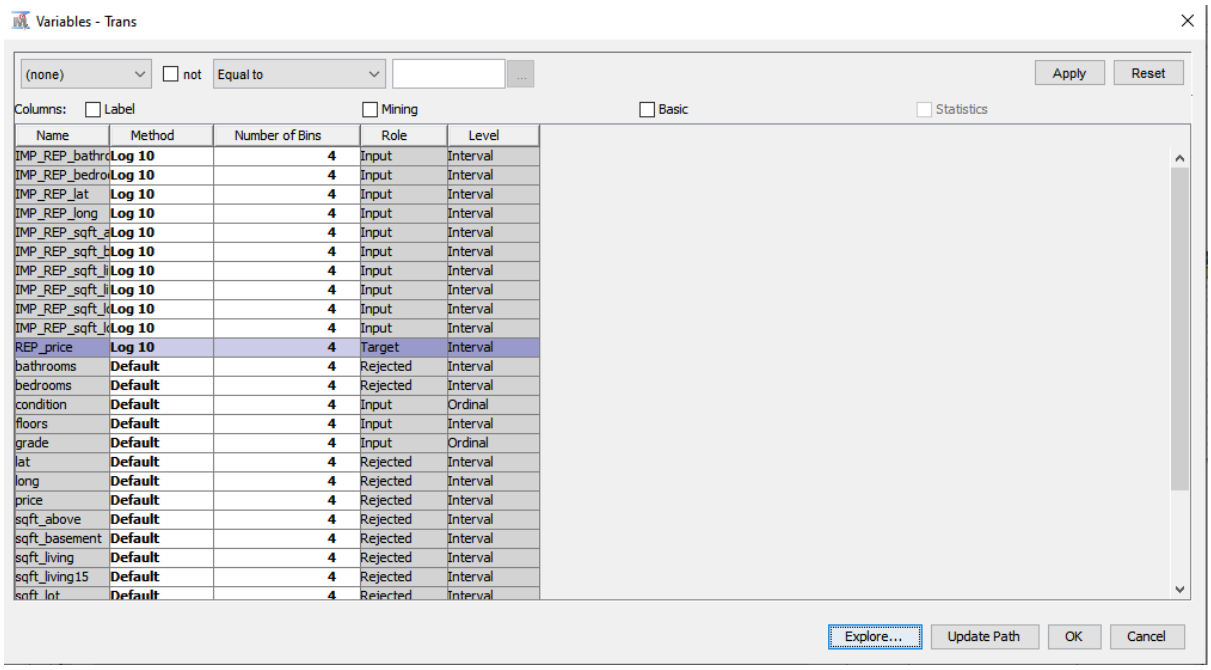
- Select Interval Variables >> Default Input Method >> Mean



- Add Transform Node



- Select Transform variables >> Train >> Variables >> Method >> Log 10



7.3.3 Data Partition

- Add Data Partition Node to Split Training Data and Validation Data

- Under Data Set Allocations, set 50% for Training and 50% for Validation

Data Set Allocations	
Training	50.0
Validation	50.0
Test	0.0

7.4 MODEL

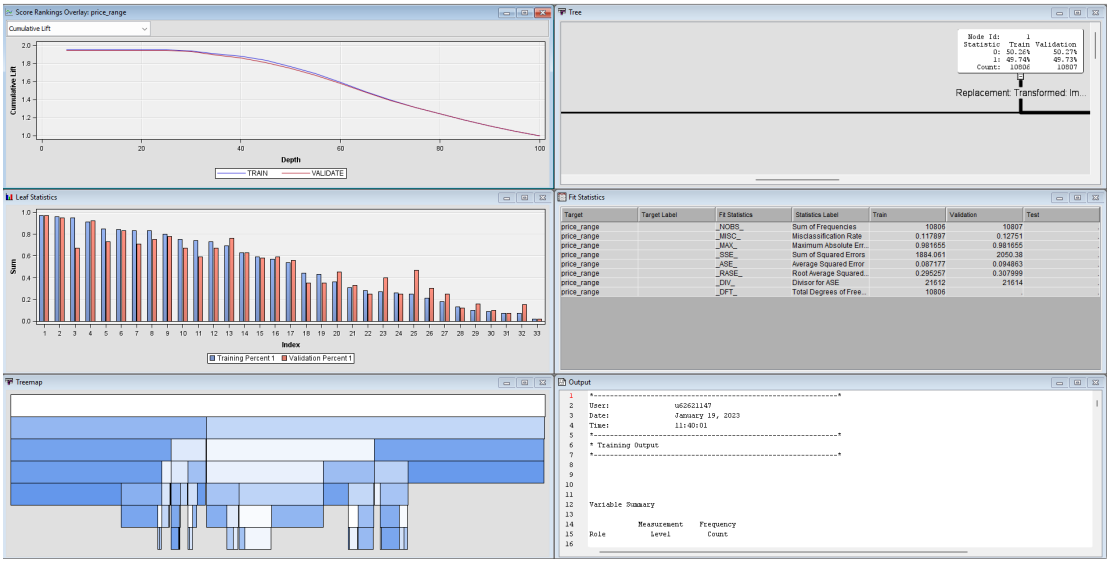
7.4.1 Decision Tree Model

- Add and connect the decision tree node to the diagram.

- The parameter is set as below to build a decision tree model.

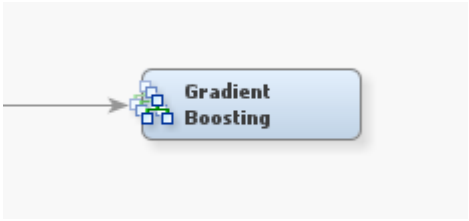
General	
Node ID	Tree
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Interactive	
Import Tree Model	No
Tree Model Data Set	
Use Frozen Tree	No
Use Multiple Targets	No
Splitting Rule	
Interval Target Criterion	Variance
Nominal Target Criterion	Entropy
Ordinal Target Criterion	Entropy
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Maximum Branch	2
Maximum Depth	6
Minimum Categorical Size	30

- Run the decision tree node with the above parameters and the outputs are as below:



7.4.2 Gradient Boosting Model

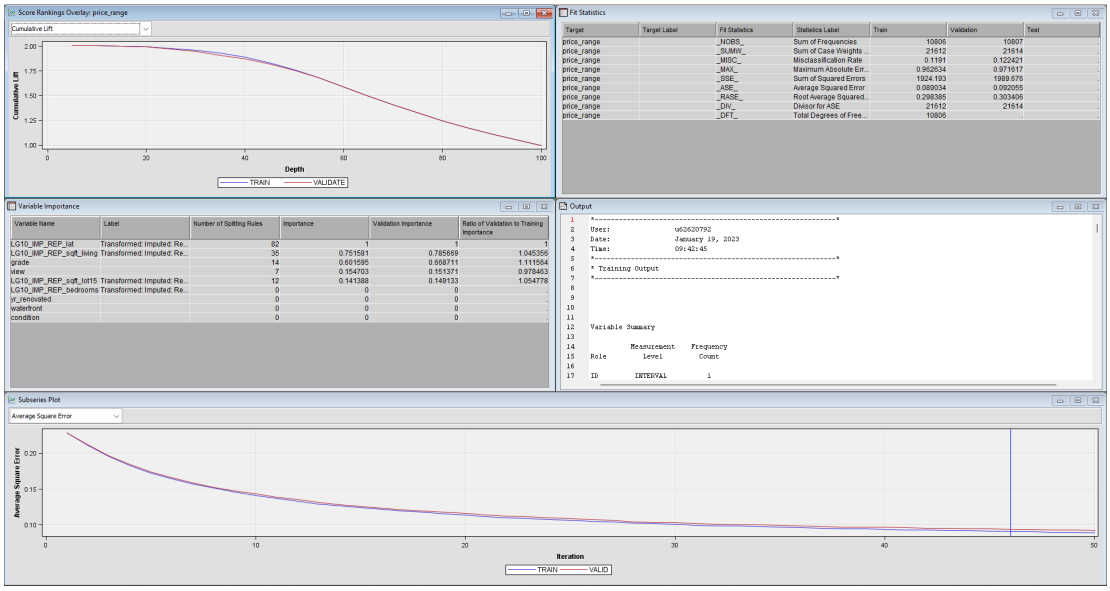
- Add Gradient Boosting Node into the diagram to create a Gradient Boosting Model.



- Set preferred parameters in the Gradient Boosting node:

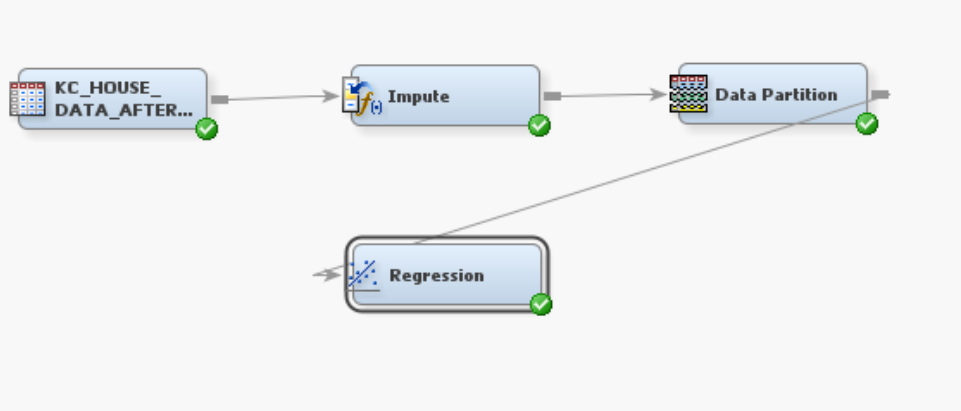
General	
Node ID	Boost
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Series Options	
N Iterations	50
Seed	12345
Shrinkage	0.1
Train Proportion	60
Splitting Rule	
Huber M-Regression	No
Maximum Branch	2
Maximum Depth	2
Minimum Categorical Size	5
Reuse Variable	1
Categorical Bins	30
Interval Bins	100
Missing Values	Use in search
Performance	Disk
Node	
Leaf Fraction	0.001
Number of Surrogate Rules	0
Split Size	.

- After setting the parameters, run Gradient Boosting Node and view the result:



7.4.3 Logistics Regression Model

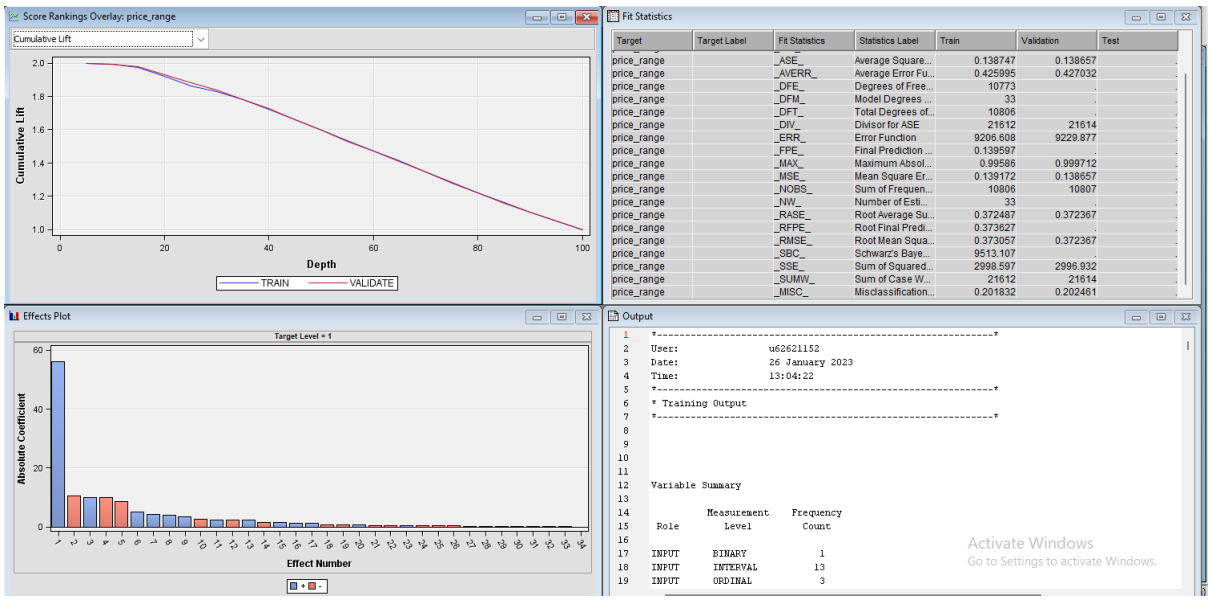
- Drag and drop the logistic regression model into the diagram and connect the previous nodes.



- Select the logistic regression as the regression type in the node and set the specified parameters:

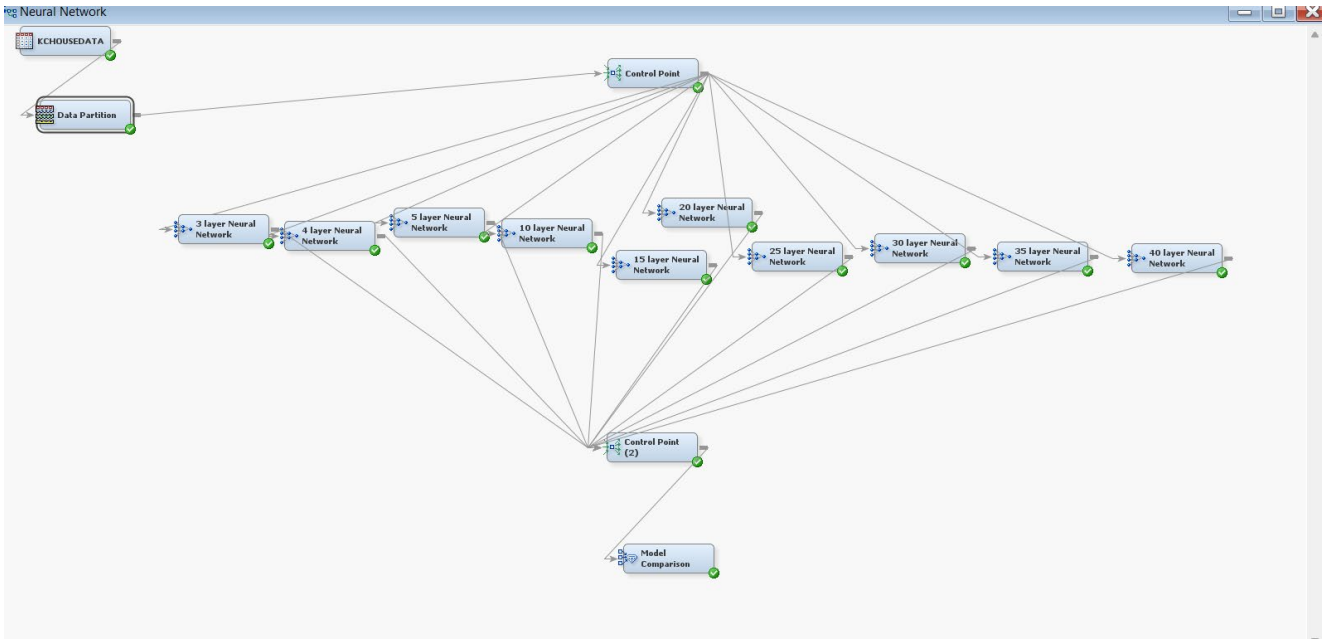
Property	Value
Variables	...
Equation	
Main Effects	Yes
Two-Factor Interactions	No
Polynomial Terms	No
Polynomial Degree	2
User Terms	No
Term Editor	...
Class Targets	
Regression Type	Logistic Regression
Link Function	Logit
Model Options	
Suppress Intercept	No
Input Coding	Deviation
Model Selection	
Selection Model	Stepwise
Selection Criterion	Default
Use Selection Defaults	Yes
Selection Options	...
Optimization Options	

- After setting the parameters, run the logistic regression and view the results as below:



7.4.4 Neural Network Model

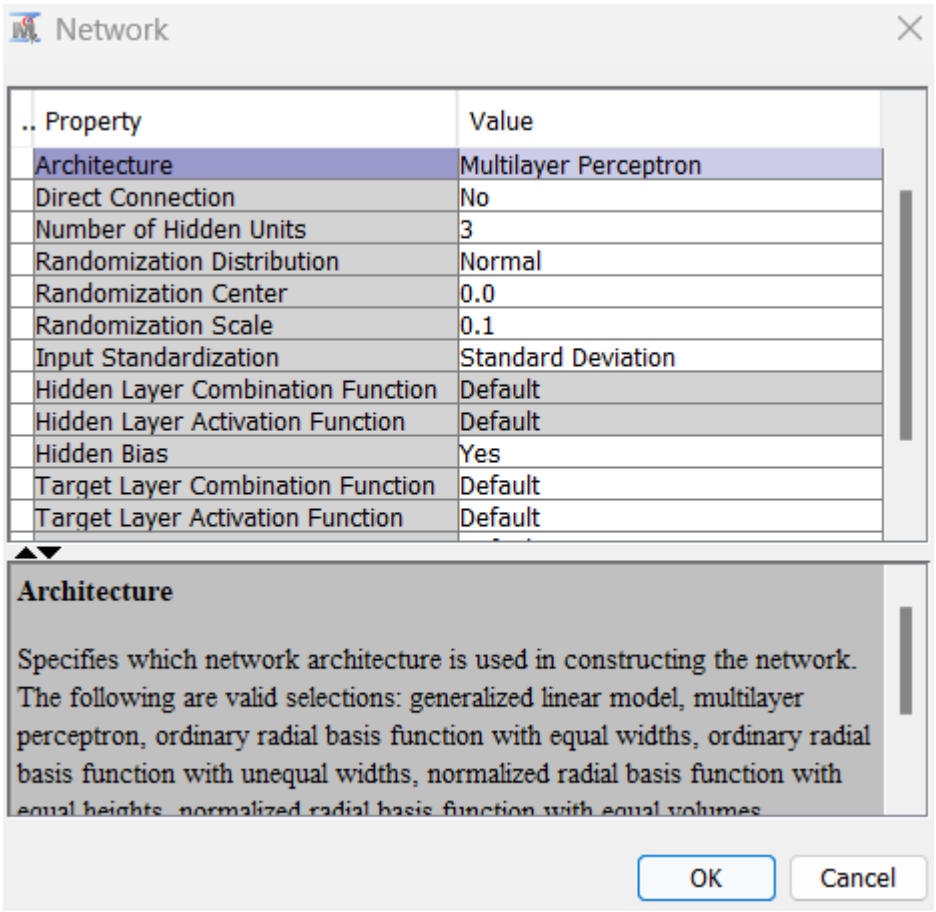
- Create neural network model by dragging and dropping the different neural network nodes with various hidden layers into the diagram and connect the previous nodes.



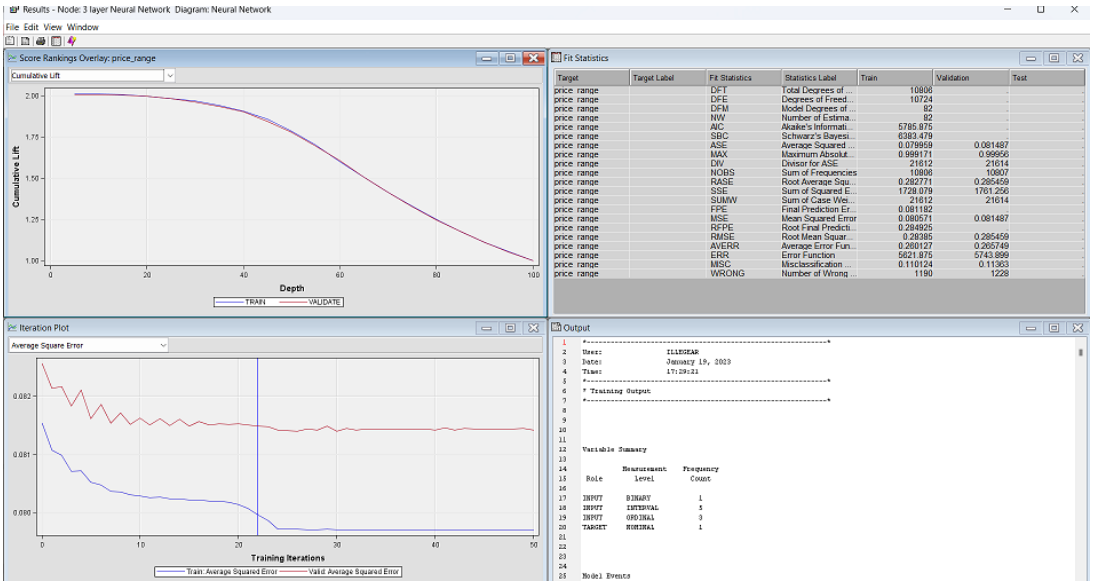
- Set preferred parameters in each neural network node

.. Property	Value
General	
Node ID	Neural
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Continue Training	No
Network	
Optimization	
Initialization Seed	12345
Model Selection Criterion	Profit/Loss
Suppress Output	No
Score	
Hidden Units	No
Residuals	Yes
Standardization	No
Status	
Create Time	1/19/23 10:51 AM
Run ID	42640b96-3ed1-4ddd-b4
Last Error	
Last Status	Complete
Last Run Time	1/19/23 5:28 PM
Run Duration	0 Hr. 0 Min. 28.12 Sec.
Grid Host	
User-Added Node	No

-



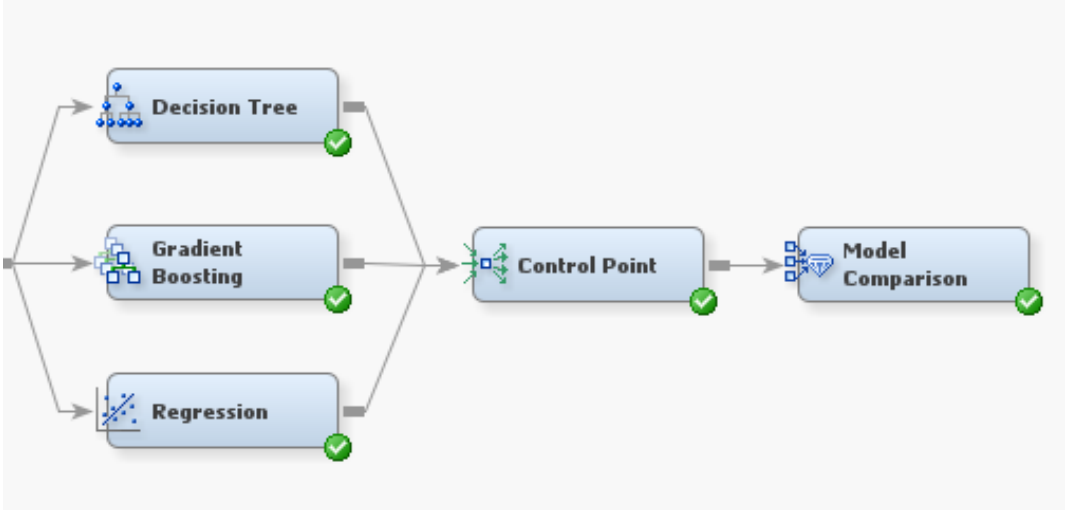
- After setting the parameters, run the neural network node and view the results as follows:



7.5 ASSESS

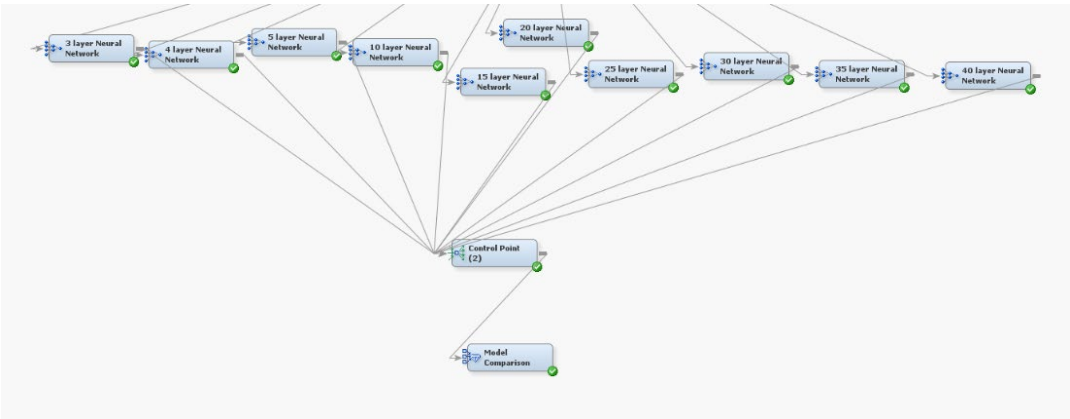
7.5.1 Compare between Decision Tree, Logistic Regression, and Gradient Boosting

Link Decision Tree, Logistic Regression and Gradient Nodes to a Control Point Node from Utility Tab and subsequently link it a Model Comparison Node from the Assess Tab.



7.5.2 Compare Neural Network

Link all the neural network nodes to the control point node and link the control point node to model comparison node.



7.5.3 Compare the Best Model

With two Model Comparison Nodes available, link both Model Comparison Nodes to a new Control Point Node and subsequently link it to a new Model Comparison Node.