**Predictive Analytics and Business Intelligence Project**

**Thao Nguyen**

**Maria Cooper**

1. **Business/Research Understanding Phase**

**Project Objective**

The purpose of this data project is to determine the 20 most luxurious homes out of 100 candidate properties for which the sales prices are not known. The broker of the properties gave the team a list of 100 properties and their information as the Score data set, consisting of many assessed values for them. However, this “Score” data set does not contain the final sale prices or a couple of other variables. For some properties, some values of a few variables may be missing. The team will need to develop models and appropriate procedures to predict and rank the “Sale Price” and recommend the 20 most luxurious houses. The reason for investing in the top luxury houses is that their potential investment growth is believed to be the greatest.

**Project Data**

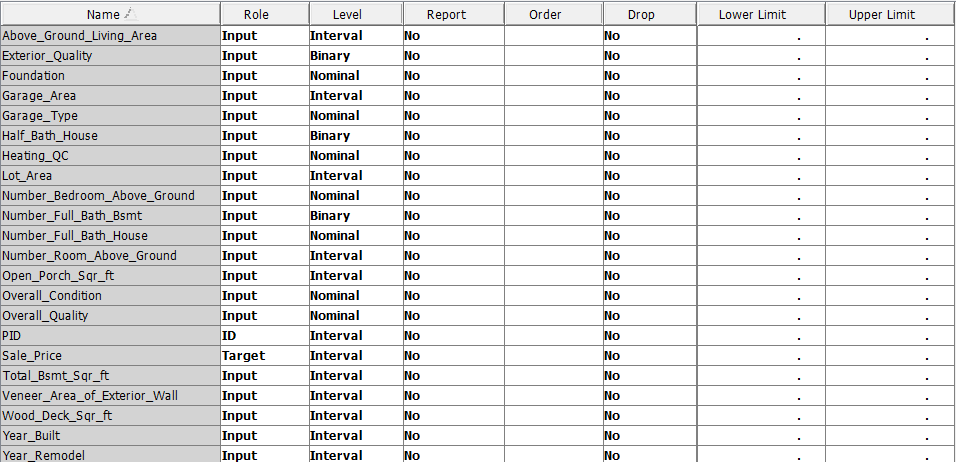
The first of the two data sets provided is called “property”. The data file's values are for individual residential real estate properties sold in that city over 4 years. This file is for data mining purposes. The team is also being given a second file called “Score Data – No Sale Price” which will include many of the variables in the initial data file but there will be some missing variables.

1. **Data Understanding Phase**

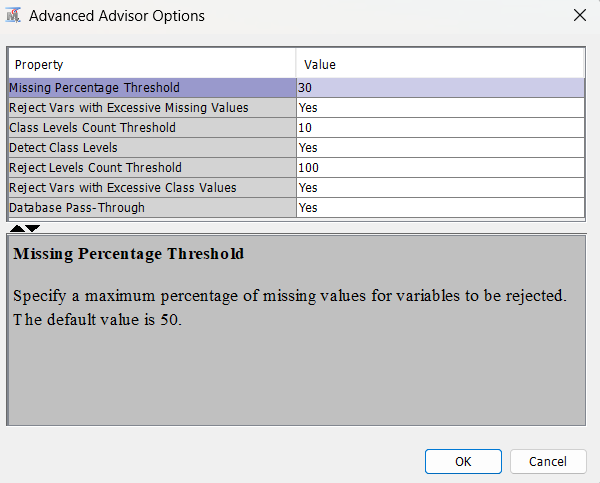
The reasons why we chose the level of each variable was due to:

Above\_Ground\_Living\_Area, Garage\_Area, Lot\_Area, Number\_Room\_Above\_Ground , Open\_Porch\_Sqr\_ft, Total\_Bsmt\_Sqr\_ft, Veneer\_Area\_of\_Exterior\_Wall, Wood\_Deck\_Sqr\_ft, Year\_Built, and Year\_Remodel are all intervals as they are in a range of values and will allow for a wide range of statistical analyses.

Exterior\_Quality, Foundation, Garage\_Type, Half\_Bath\_House, Heating\_QC, Number\_Bedroom\_Above\_Ground, Number\_Full\_Bath\_Bsmt, Number\_Full\_Bath\_House, Overall\_Condition, and Overall\_Quality are class variables (binary and nominal variables) as each of them has from 2 to less than 10 unique values and they are mutually exclusive.

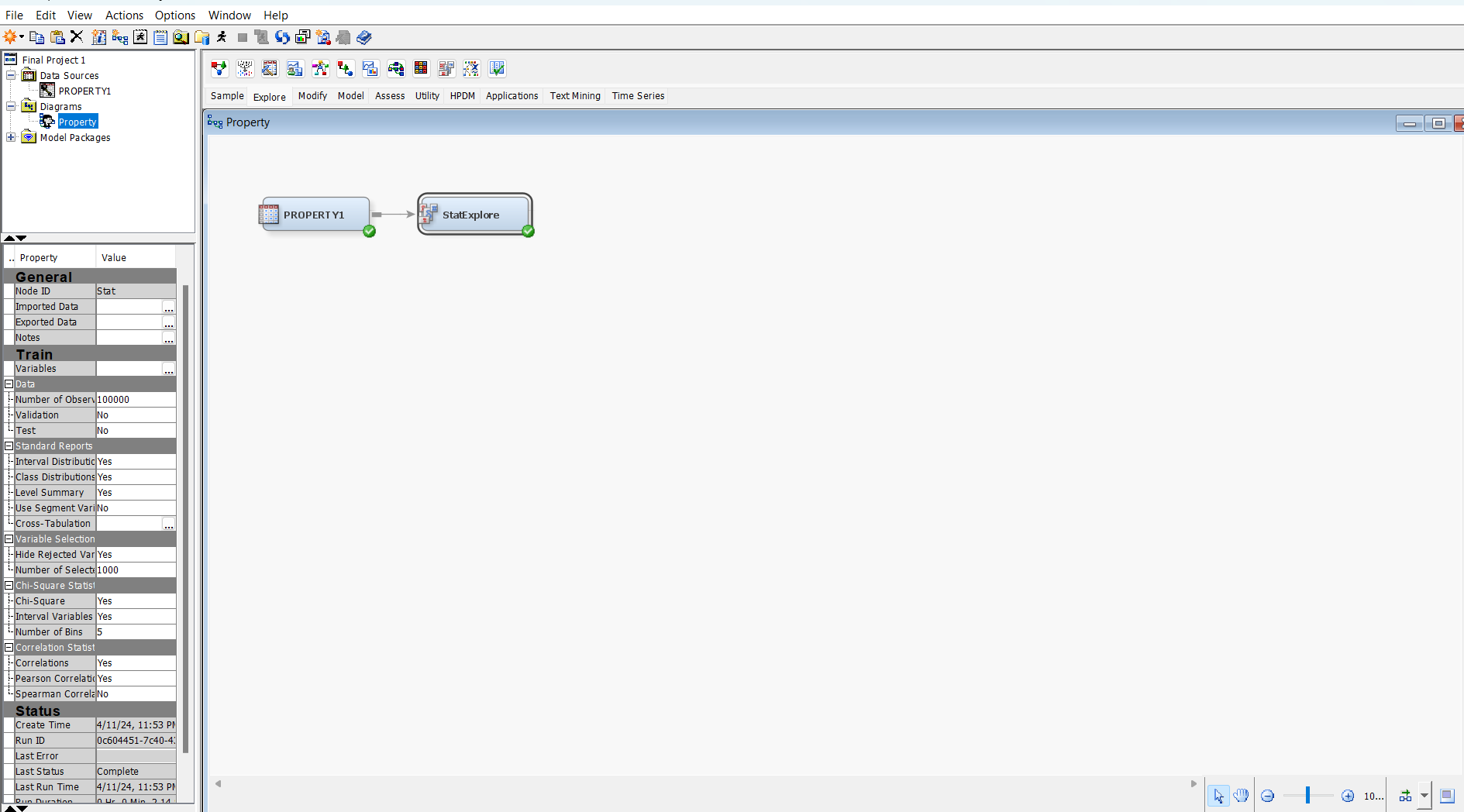
Upon project and library creation, the first dataset was imported into the SAS project file. The screenshot below depicts the initial configuration for each variable. 

We opted to set the Missing Percentage Threshold to 30, meaning that any variable with a missing percentage of 30% or higher would be disregarded. Similarly, the Class Levels Count Threshold was established at 10, designating variables with 10 or more distinct values as interval variables. Additionally, the Reject Levels Count Threshold was set to 100, indicating our preference to limit the maximum number of levels to 100 for a class variable before it is flagged as REJECTED.

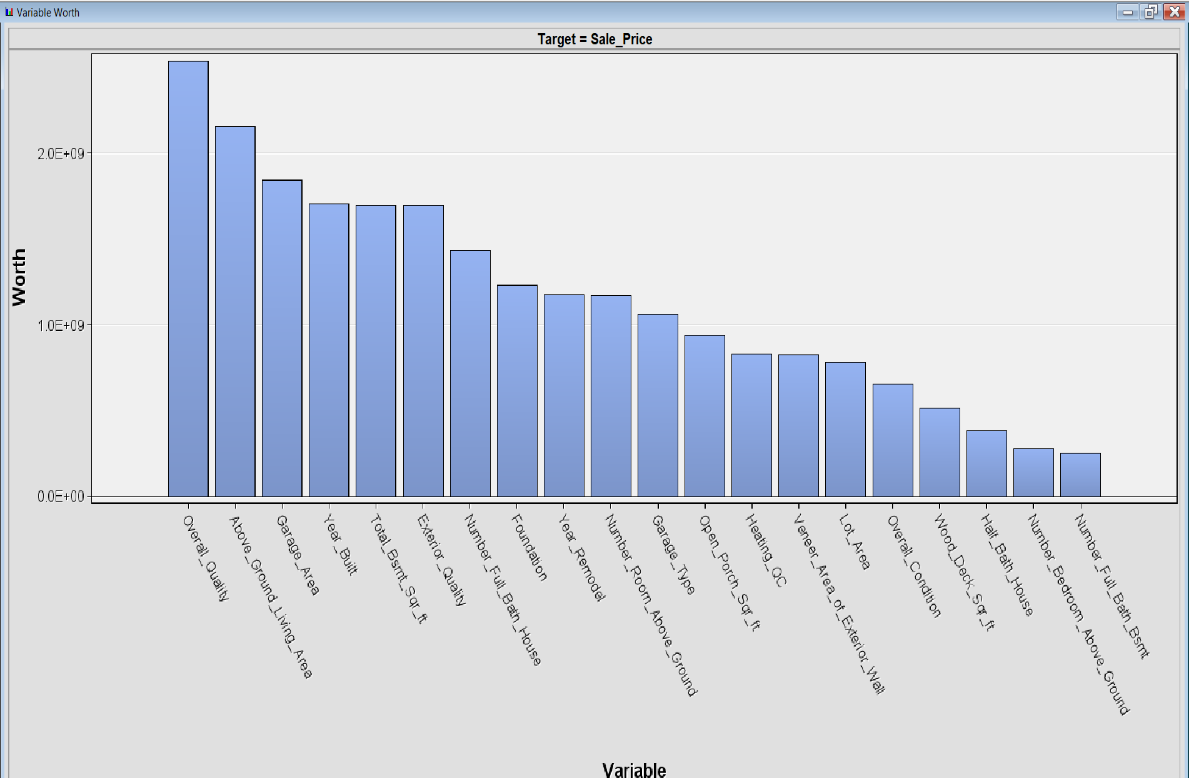


**Stat Explore Analysis**

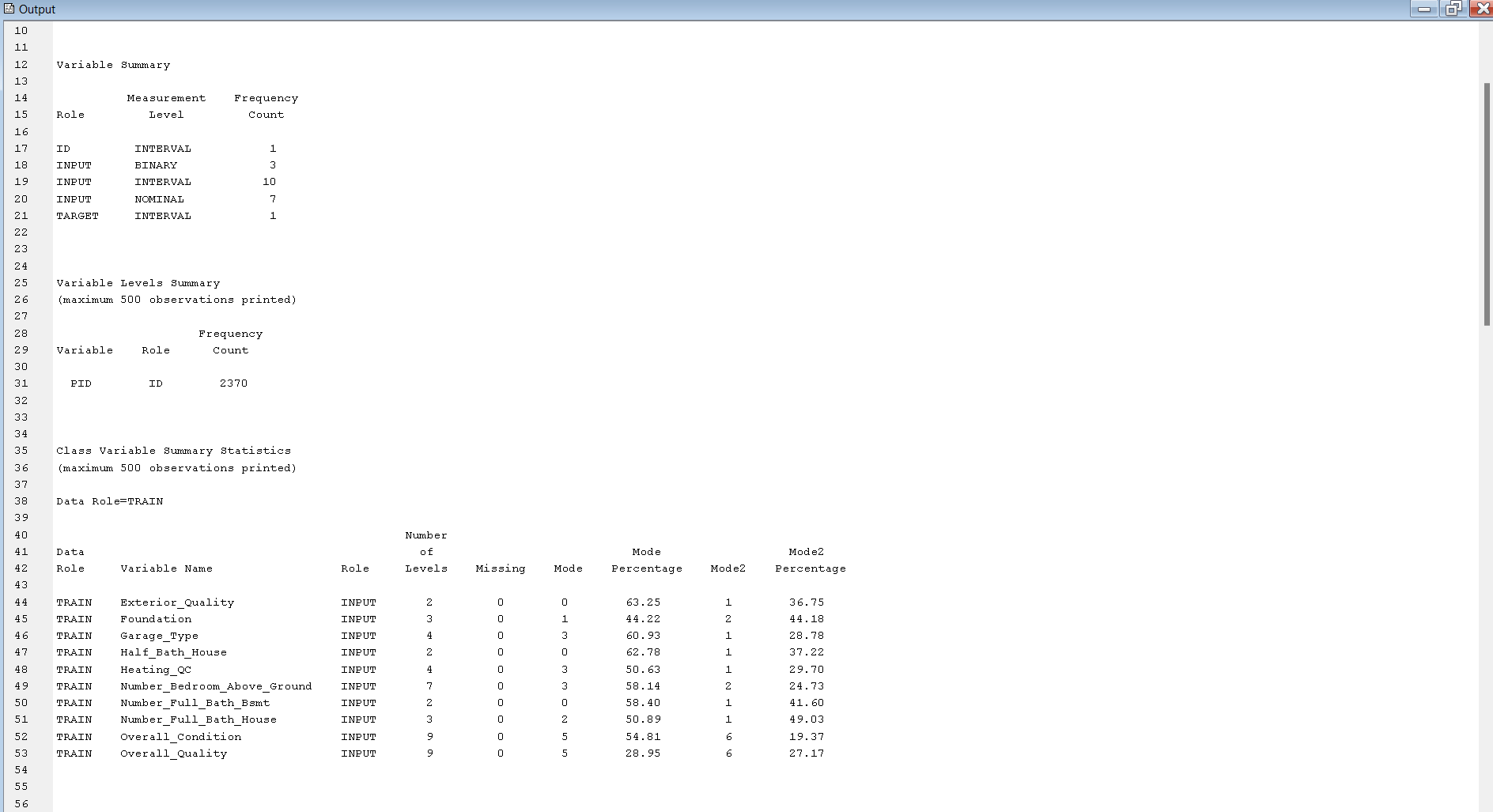
Following the dataset importation, we accessed the StatExplore node via the Explore button situated on the top Toolbar of the Diagram workspace. This facilitated an exploration into the correlation between each input variable and the target variable, enabling the detection of any missing values and the observation of variable distributions. Notably, we included all interval variables in this analysis by selecting "yes" for the interval variable option within the Chi-Square Statistics property. In total, there were 20 input variables under scrutiny.

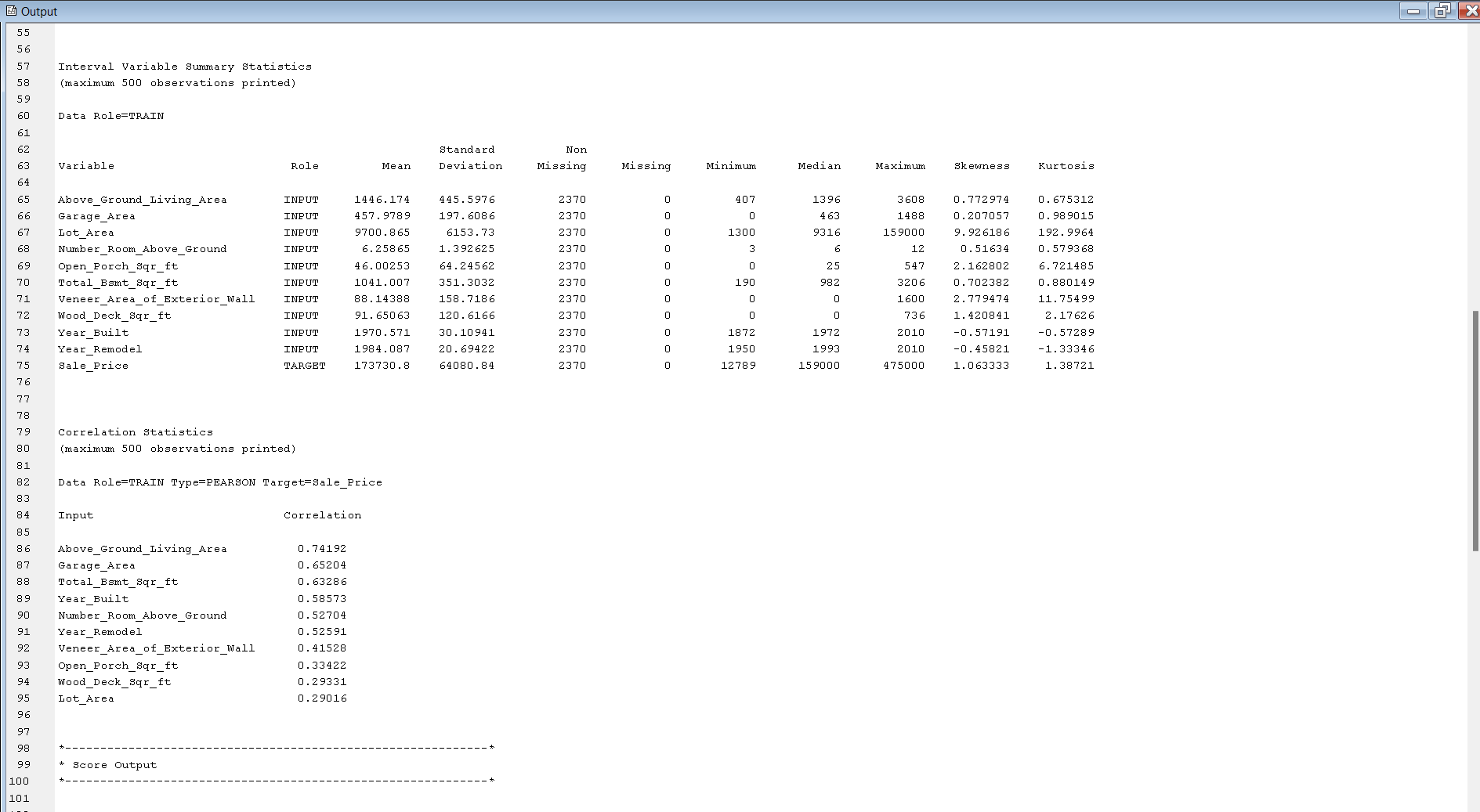


Upon executing this node, the outcome window displays the correlations between the input variables and the target variable. A higher bin indicates a stronger correlation of the input variables with the target variable, namely, the sale price. It is evident from the results that the overall quality emerges as the most influential variable affecting the sale price.



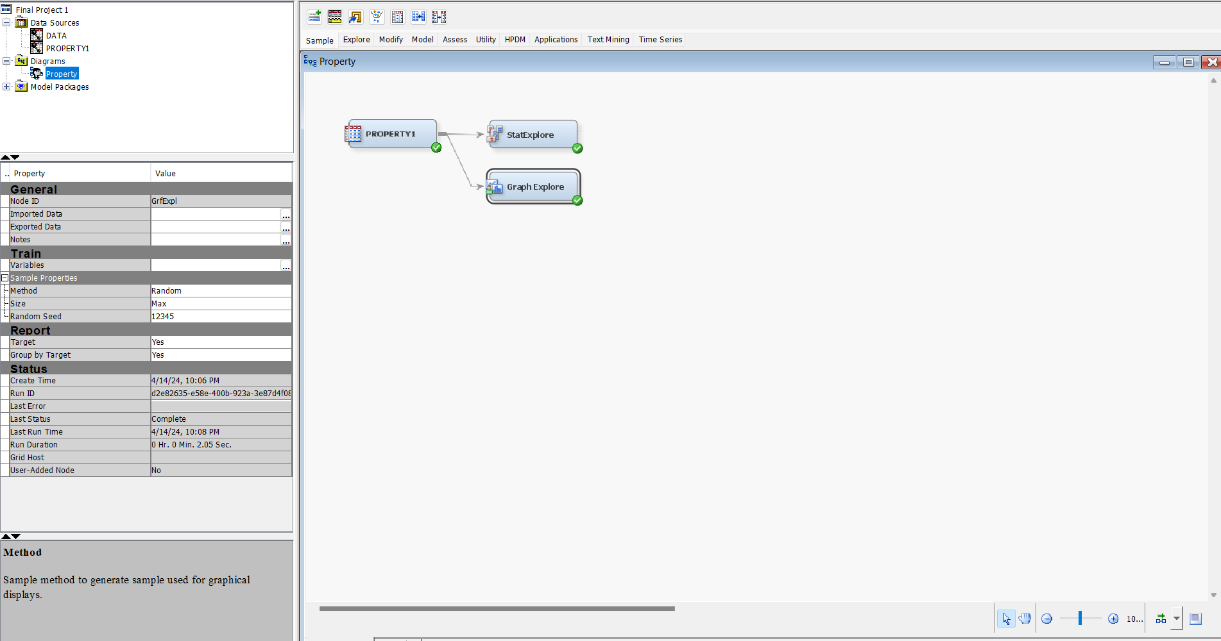
Excluding the ID and target variables, our dataset comprises 10 interval variables, 3 binary variables, and 7 nominal variables, all of which contain no missing values. The following output screenshots present the statistical summary of these variables.



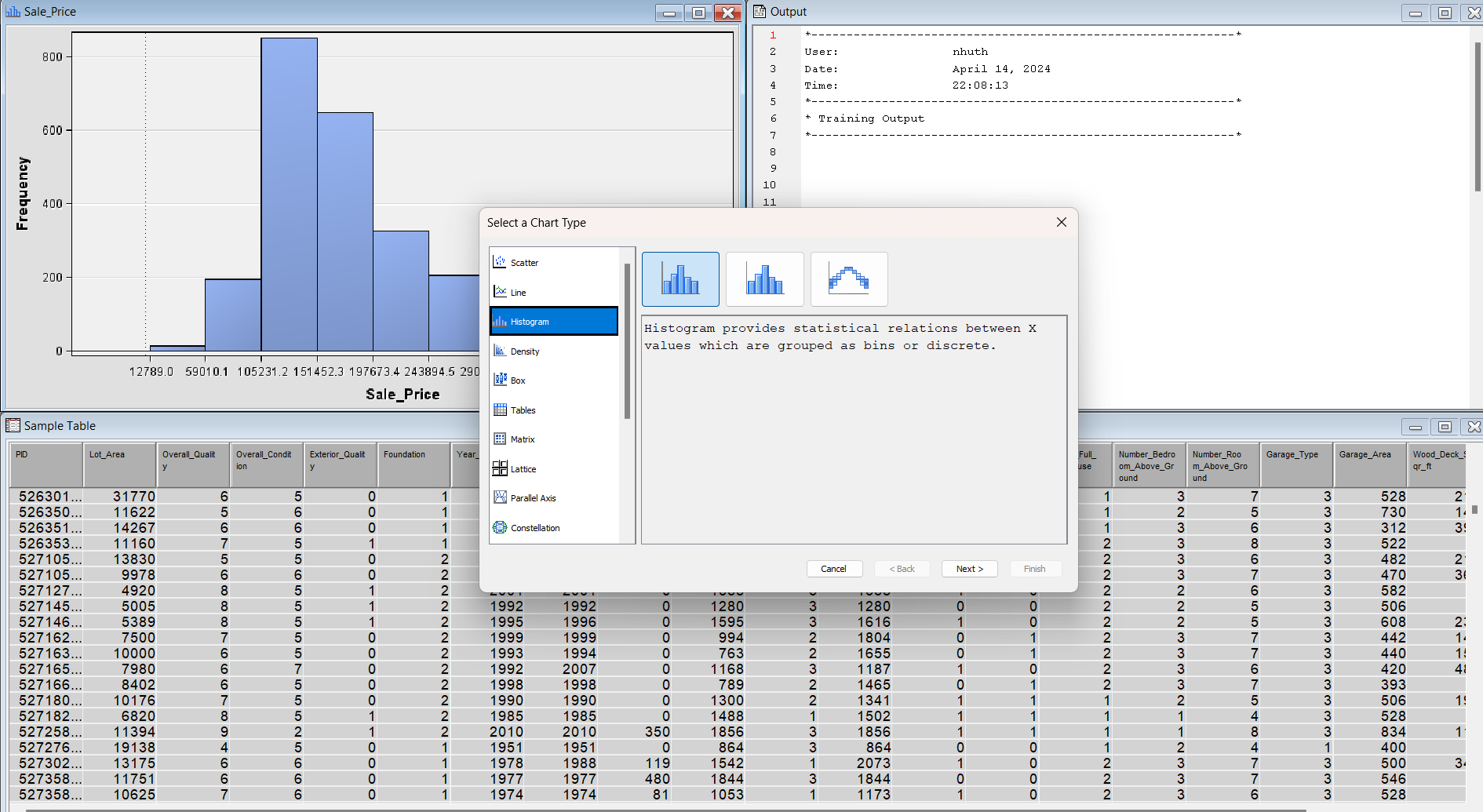


**Graph Explore Analysis**

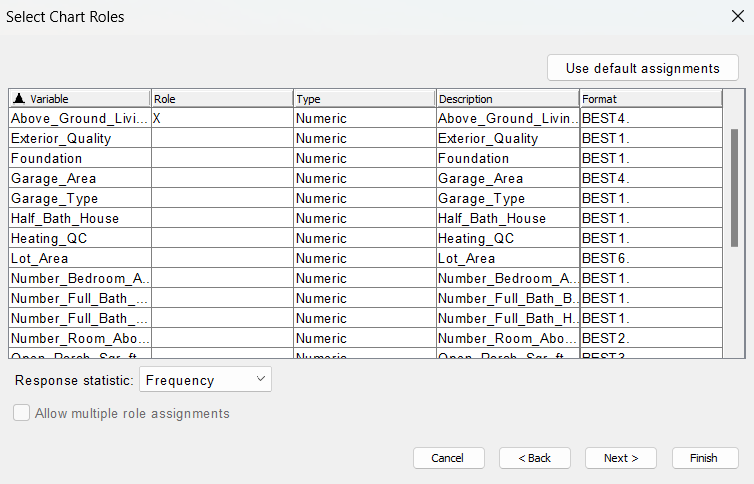
To gain visual insights into each individual variable, we utilized the GraphExplore node, accessed from the "Explore" option. We opted for the default method for the sample properties and set the size to max, allowing for observation of the entire dataset.



After running the node, we selected “Plot” from the View menu. Histogram is selected for all variables to see the distributions.

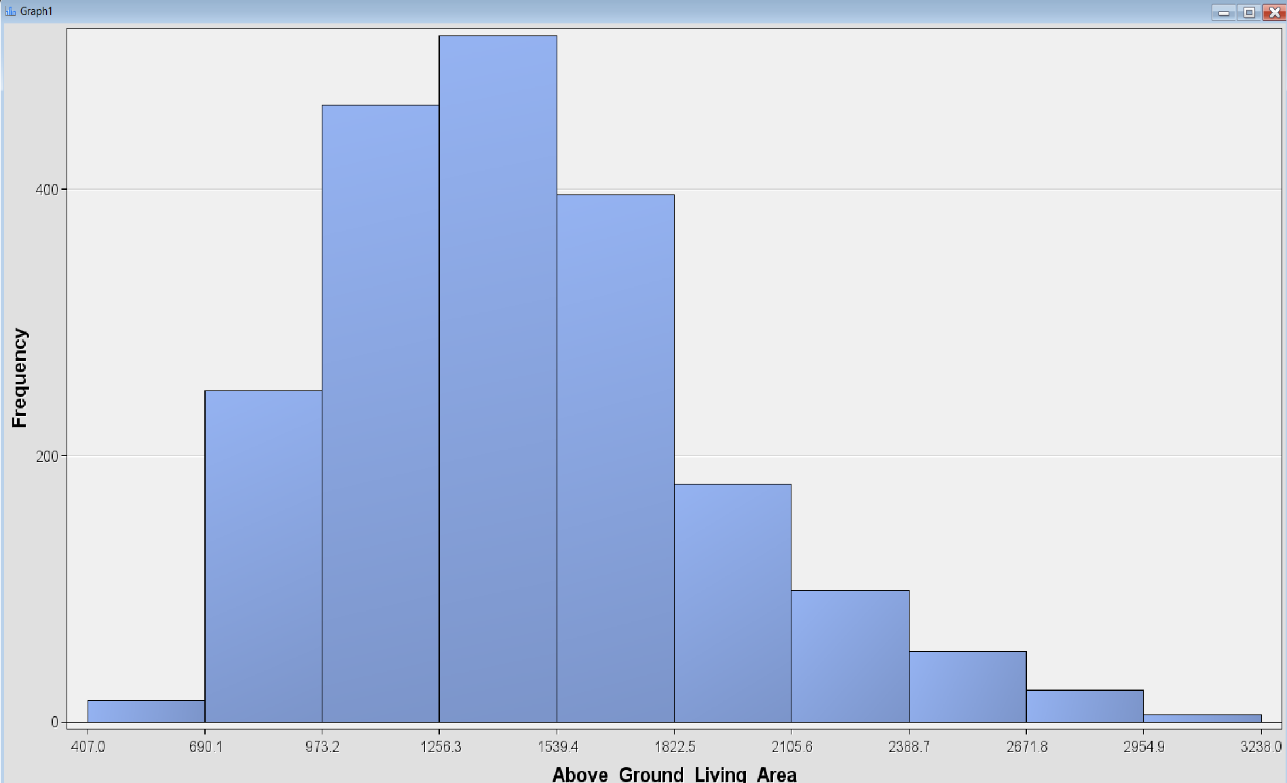


Each variable was chosen as role of x, the y is the frequency.



As a result, we subsequently created histograms for each variable by repeating the above 2 steps.

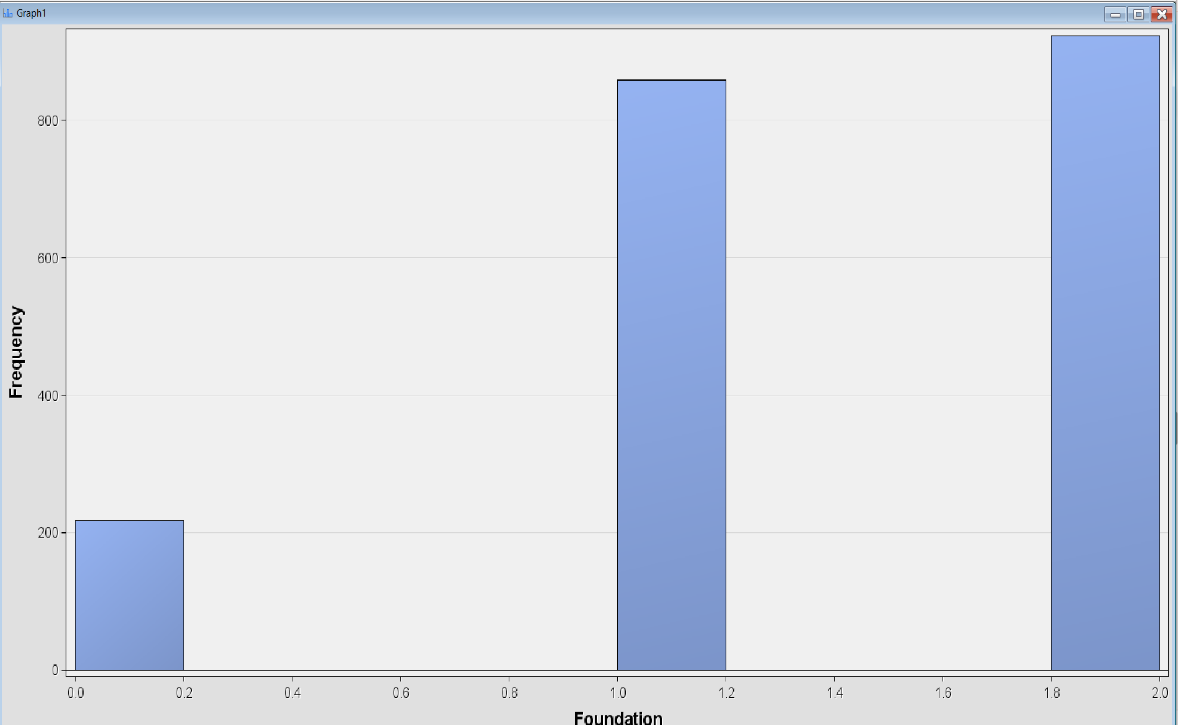
From the screenshot below, we can see the frequency of the above ground living areas distribution. With the most popular being around 1256.3 sq Ft to about 1539.4 sq Ft with the frequency being almost 500.



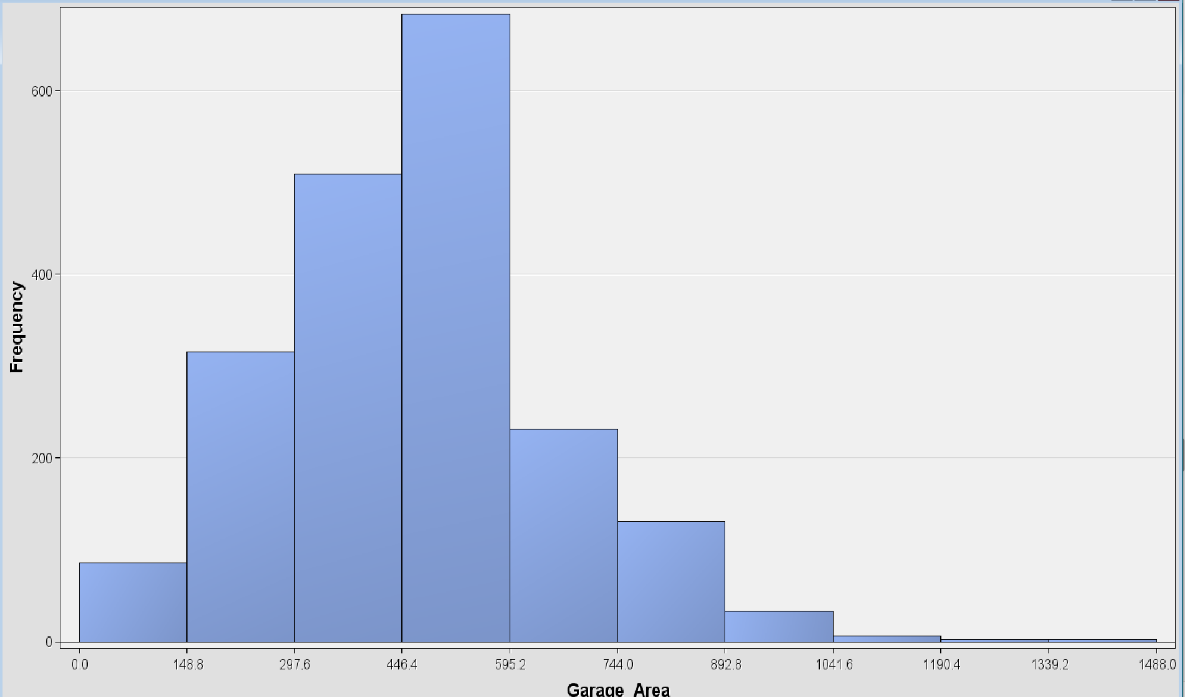
Exterior quality is a binary variable which is why there are two bins. If the house is good quality, it is represented by a 1 and if it is represented by a 0 it means the house is average quality. Looking at the frequency we can see there are more houses with an average exterior quality (>1000) than houses with a good exterior quality (<1000).



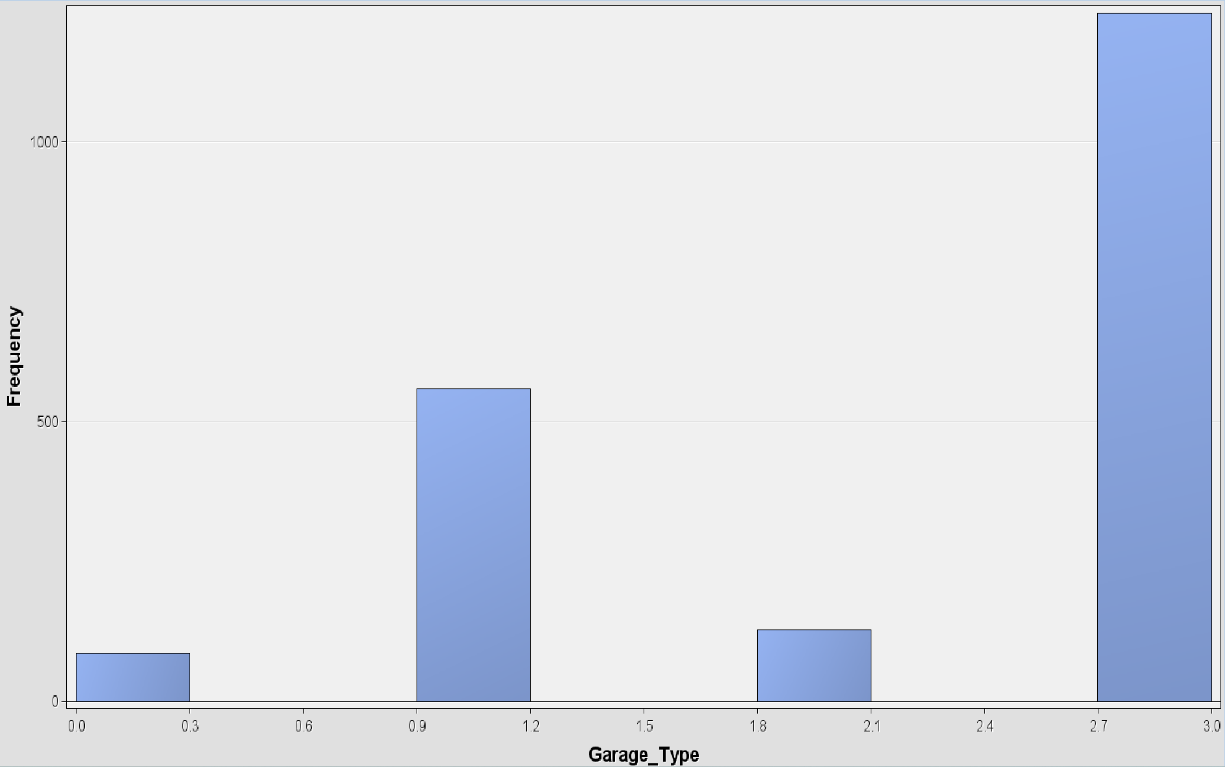
Foundation is a nominal variable. A concrete foundation is indicated by a value of 2; a cinder-block foundation by a value of 1; and brick foundation by a value of 0. According to the data, most of the houses have a concrete foundation with a frequency of almost 1000. Followed by cinder-block foundation which also has a frequency of almost 900. Brick foundation appears to be the lowest frequency.



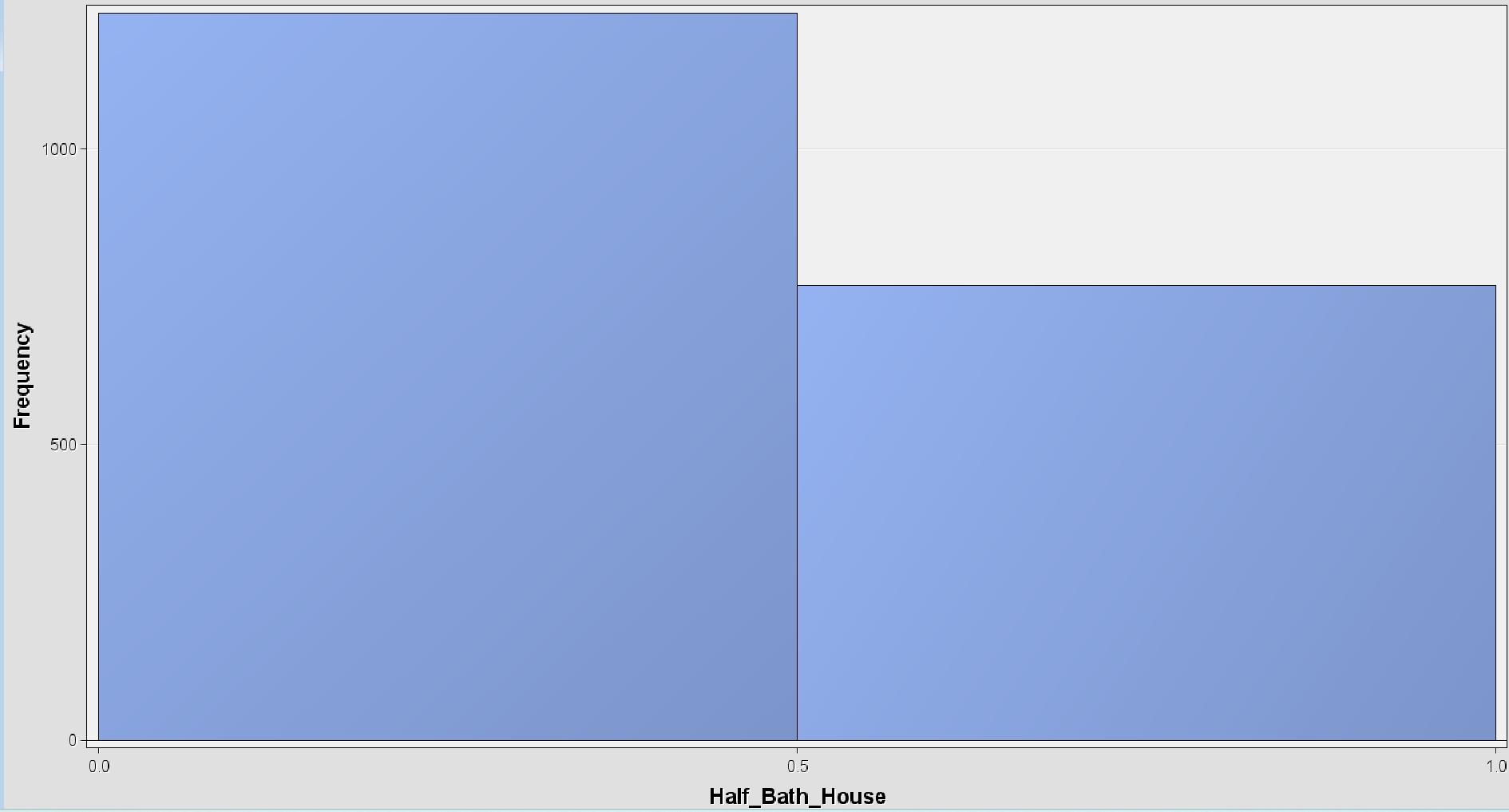
Garage area distribution ranges from 148 Sq Ft to 1488Sq Ft. Most houses have a garage that ranges from 300 Sq Ft to 595 Sq Ft. With a frequency of about 500, the largest garage areas are from 446 Sq Ft to 595 Sq Ft.



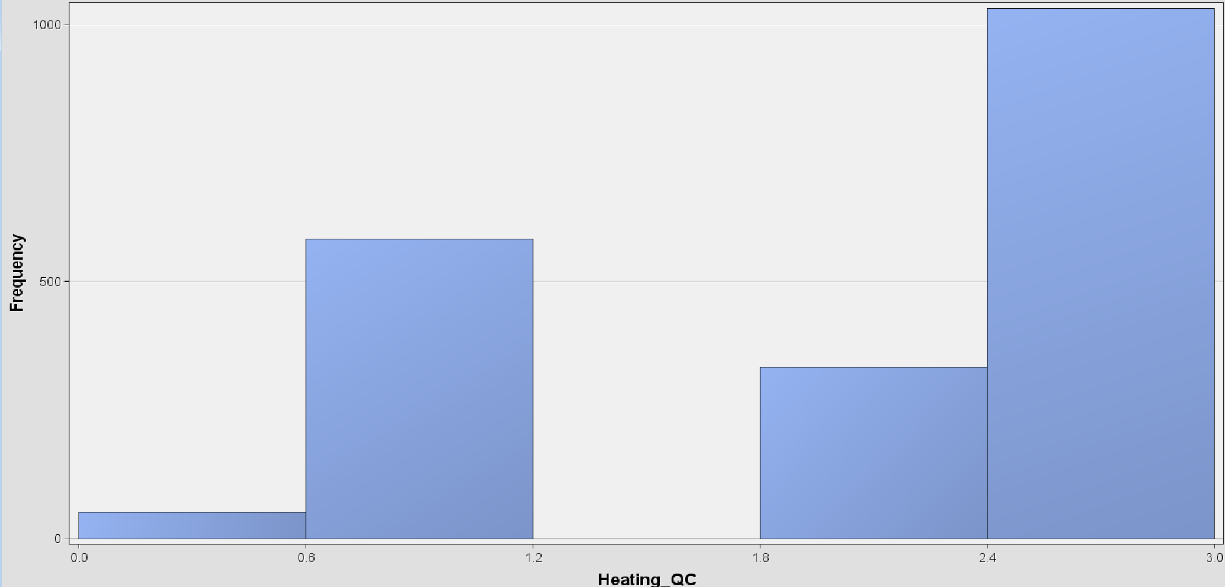
Garage type is a nominal variable. Most houses have attached garages with the frequency of about 1200, followed by a frequency of over 500 for houses that have garages detached from home. There are about 200 houses that have no garage, that represents the lowest frequency.



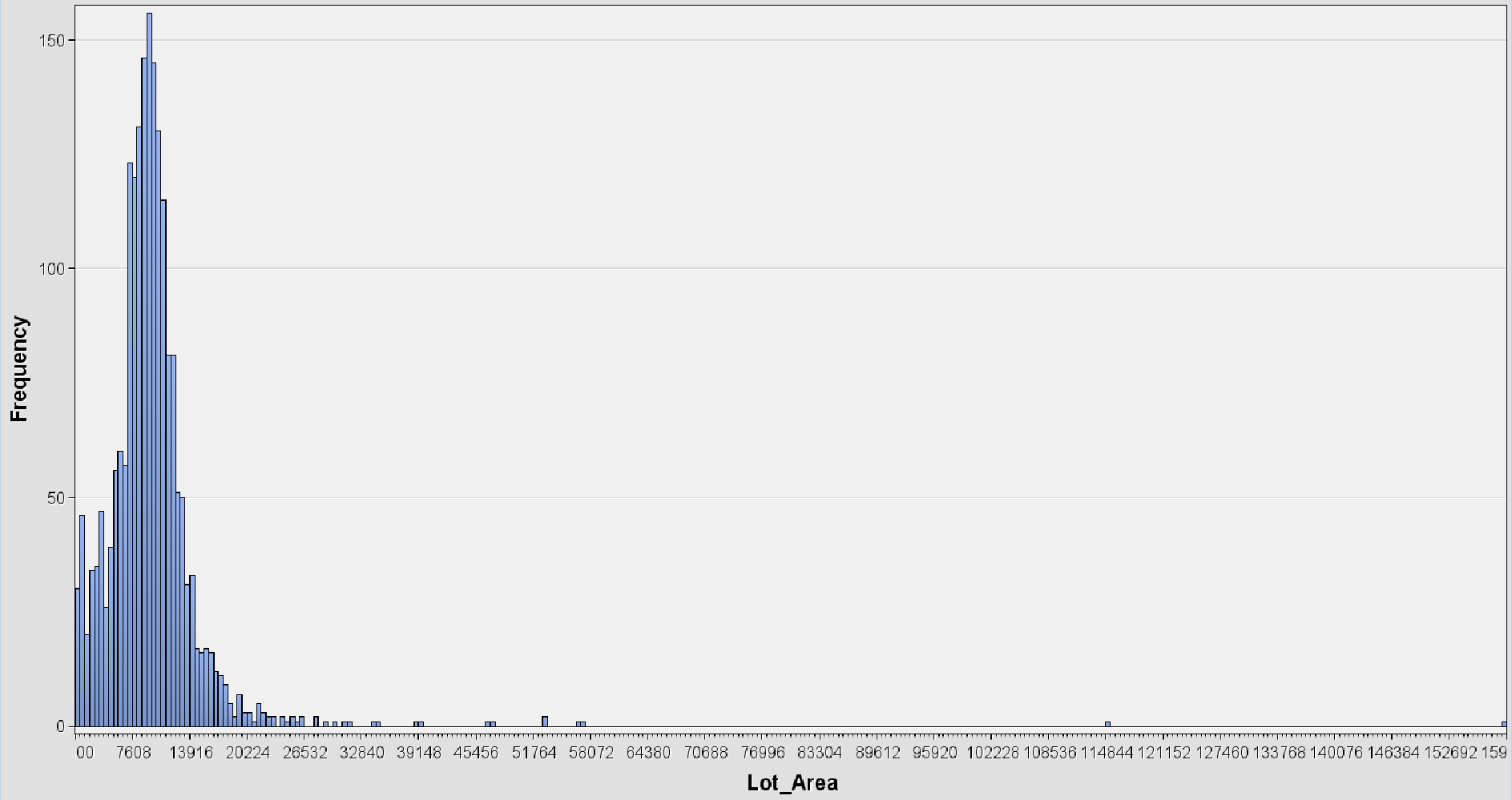
There are more than 1200 houses that don’t have a half-bath, and around 800 houses that have one. This variable is a binary one with 2 values.



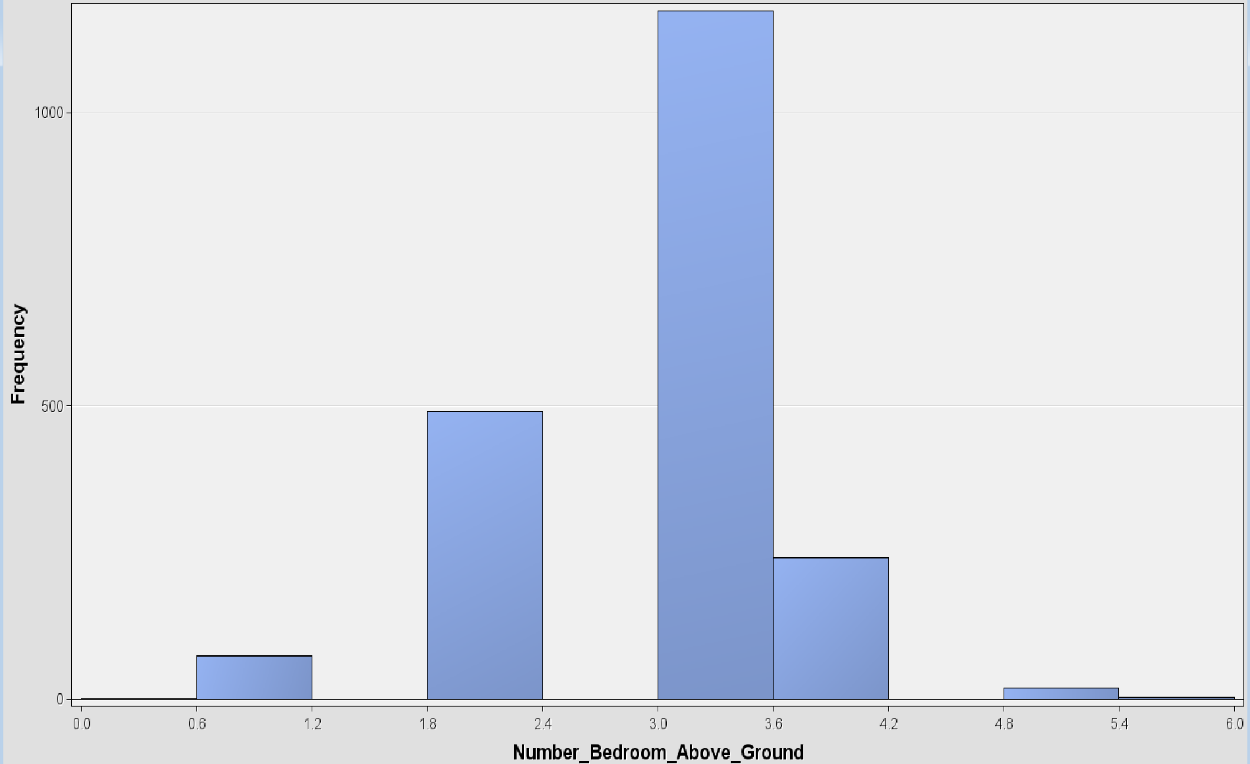
Most houses have average to excellent heating quality conditions, while there are less than 100 houses with fair quality heating condition. The most significant frequency belongs to the excellent heating quality condition, which is over 1000.

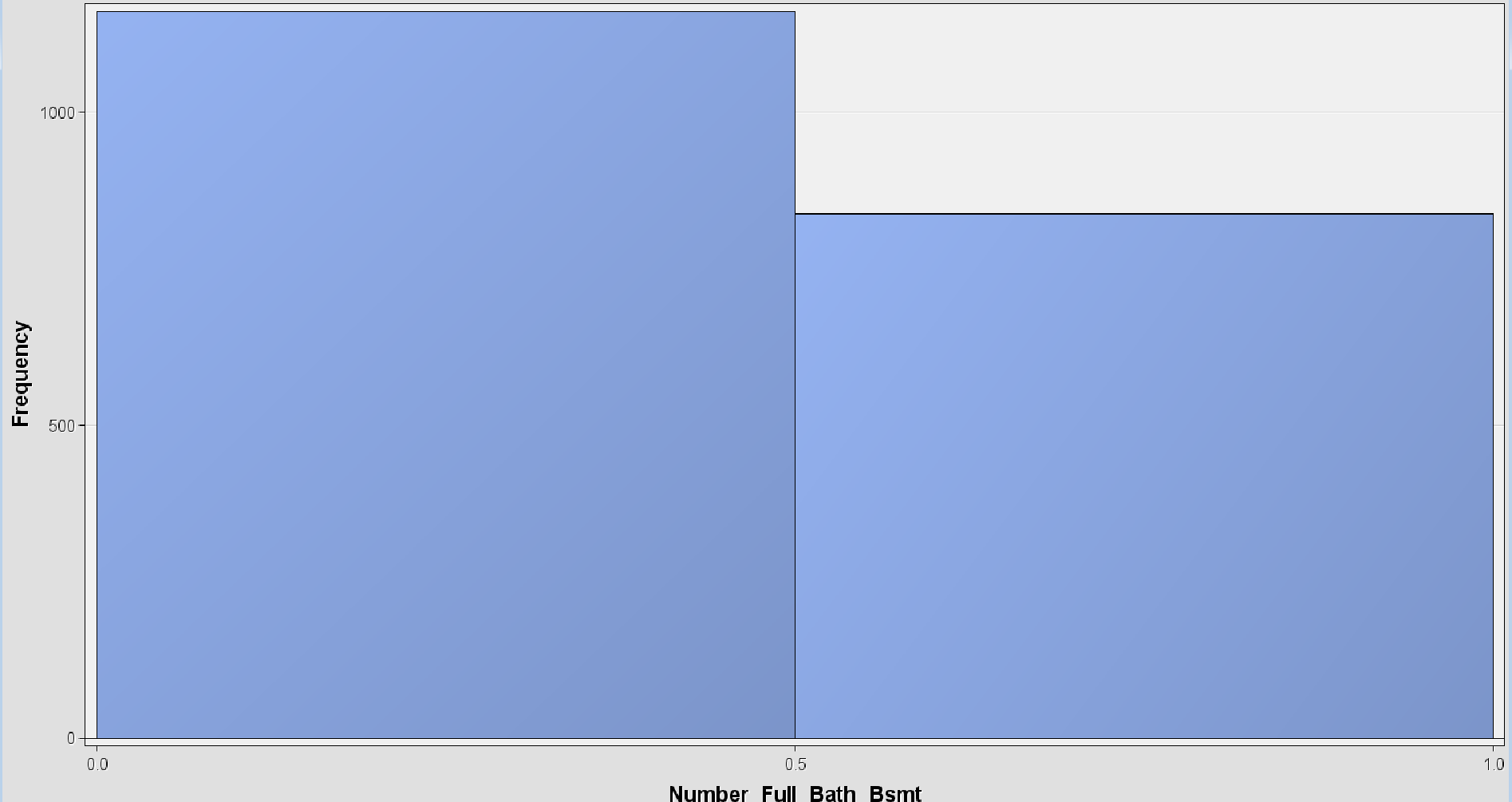


Properties have their lot area from 1300 Sq Ft to 17070 Sq Ft. Most of them rank from 7082 Sq Ft to 10762 Sq Ft with frequency from 130 to 150 properties. Less frequency of properties that have lot area larger than 14000 Sq Ft.

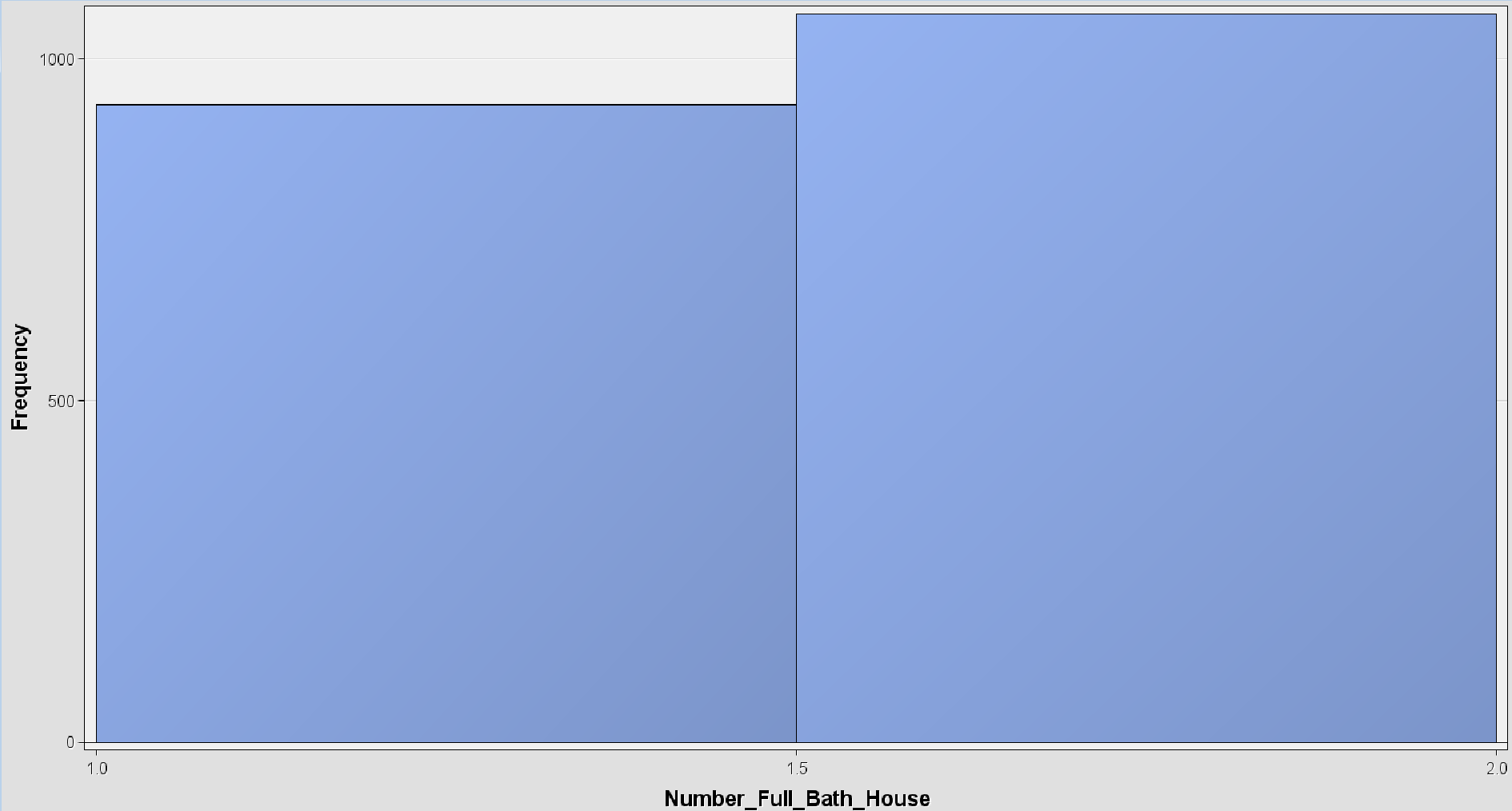


Most properties have 3 to 4 bedrooms above ground, with a frequency of 1100. The least common number of bedrooms is found in properties with 4 bedrooms.

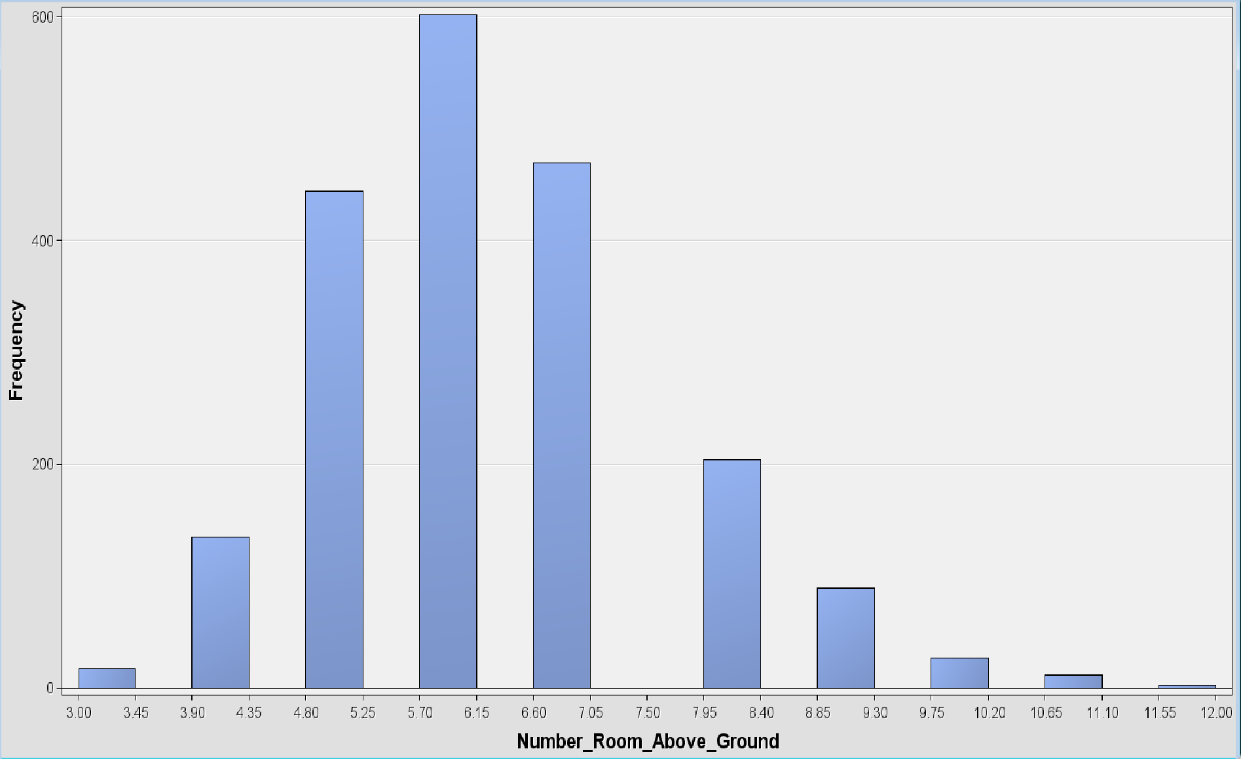
The number of full bath basement variable is binary and takes on two values: 1 or 0. A value of 1 indicates the presence of a full bathroom in the basement, while 0 indicates its absence. The frequency of the 0 value outweighs that of the 1 value.



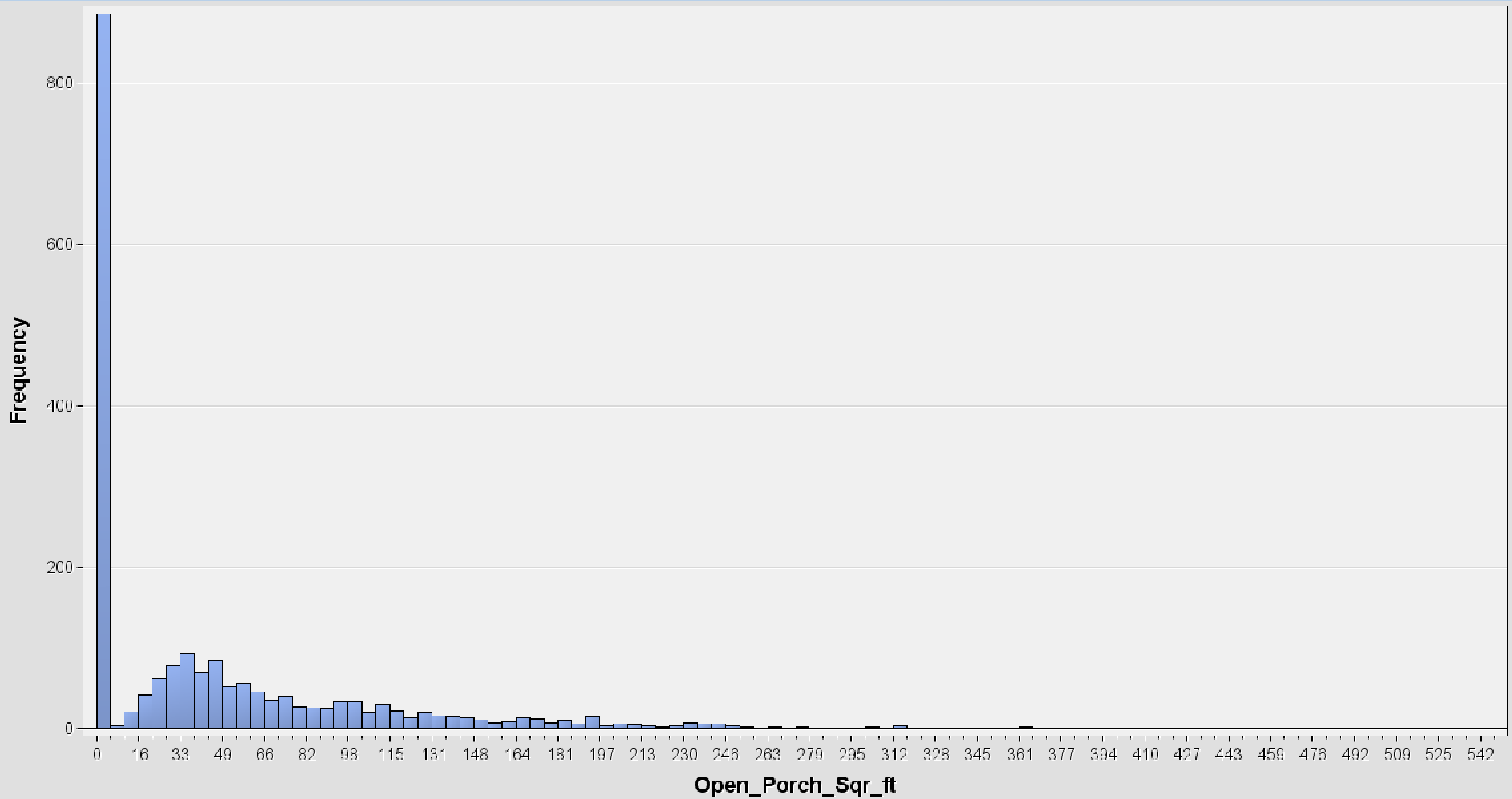
Number of full bath house indicates the number of full bathrooms there are in the house, not including bathroom in the basement. The number of properties with 2 full bathrooms is greater than that of ones with 1 full bathroom.



The number of rooms above ground is an interval variable. Most properties have 5, 6, or 7 rooms above ground, with 7 rooms being the most common, occurring 600 times.



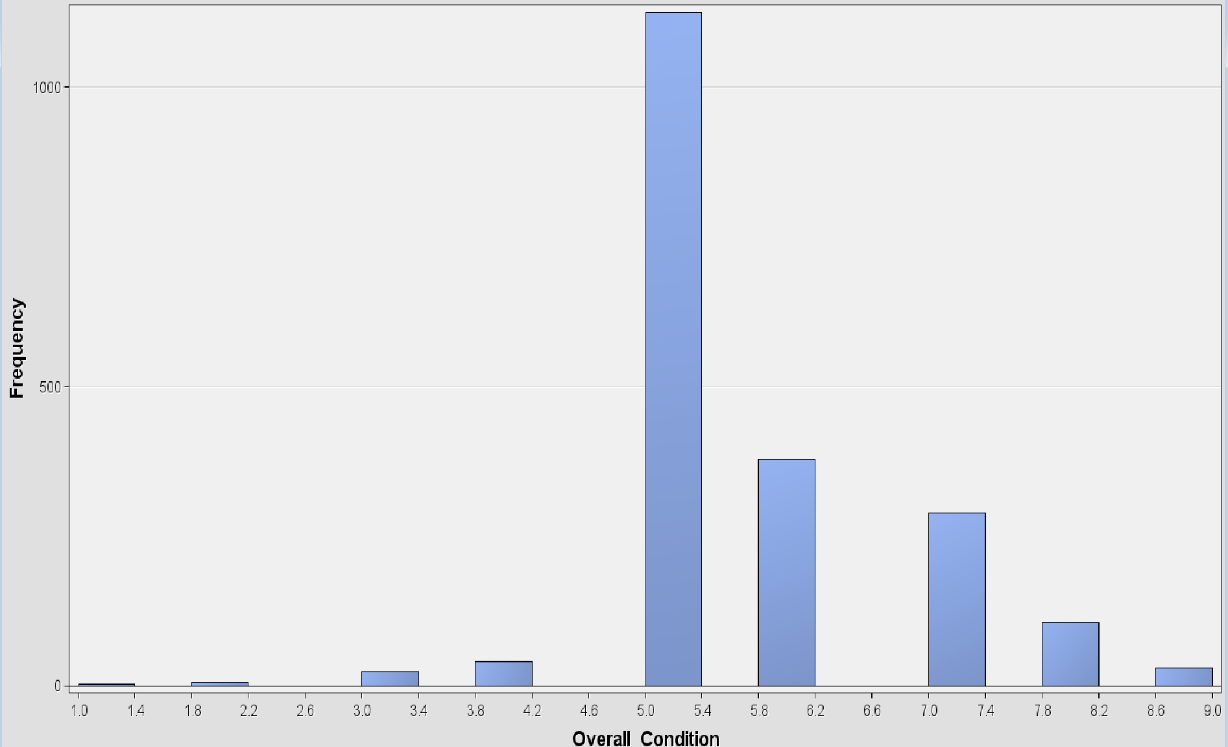
The distribution of the open Porch Sq Ft variable reveals that properties with no or less than 6 sq Ft occur almost 900 times, being the largest number of occurrences.



The overall condition is a nominal variable with values ranging from 0 to 10, where:

* 10: Very Excellent
* 9: Excellent
* 8: Very Good
* 7: Good
* 6: Above Average
* 5: Average
* 4: Below Average
* 3: Fair
* 2: Poor
* 1: Very Poor.

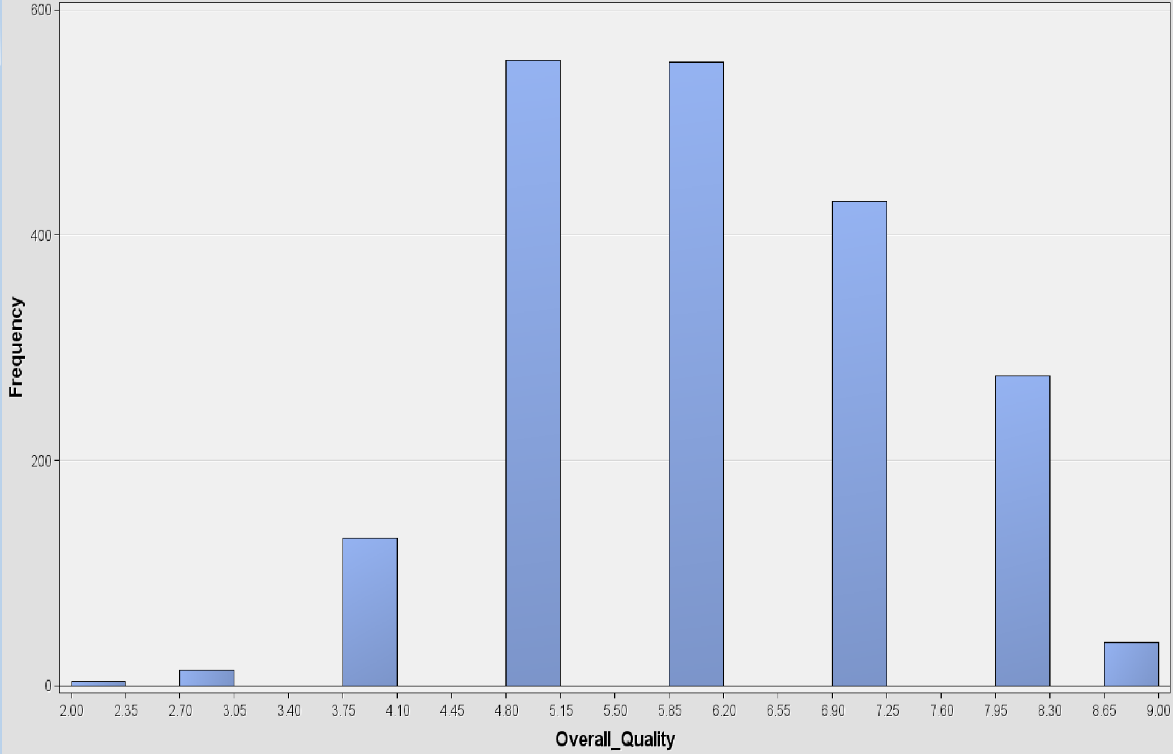
In the distribution of overall condition variable, there are no occurrences of the value 10. Meanwhile the value of 5 is the most common condition found in more than 1000 properties.



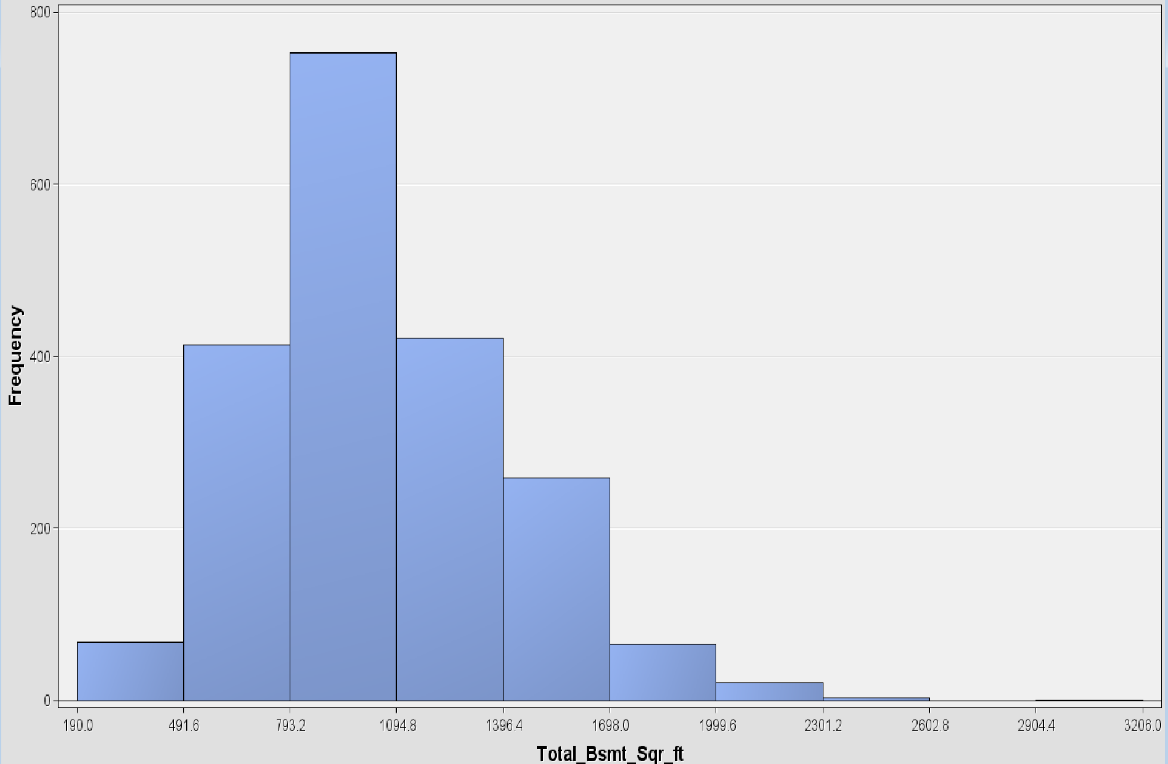
The overall quality is also a nominal variable with values ranging from 0 to 10, where:

* 10: Very Excellent
* 9: Excellent
* 8: Very Good
* 7: Good
* 6: Above Average
* 5: Average
* 4: Below Average
* 3: Fair
* 2: Poor
* 1: Very Poor.

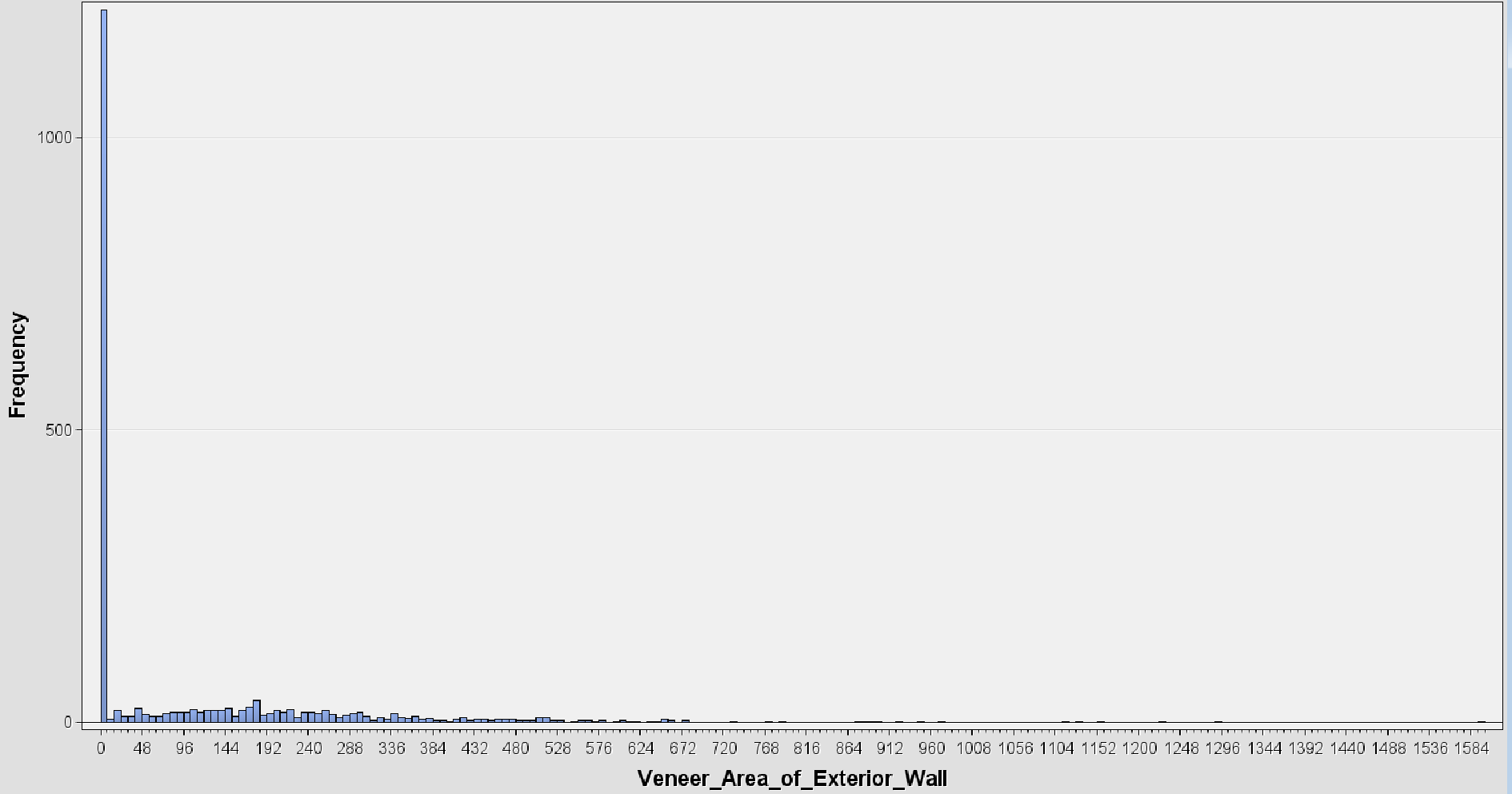
In overall quality distribution, there are no occurrences of the value 10, while values 5 and 6 are the most common qualities for properties, each occurring around 550 times.



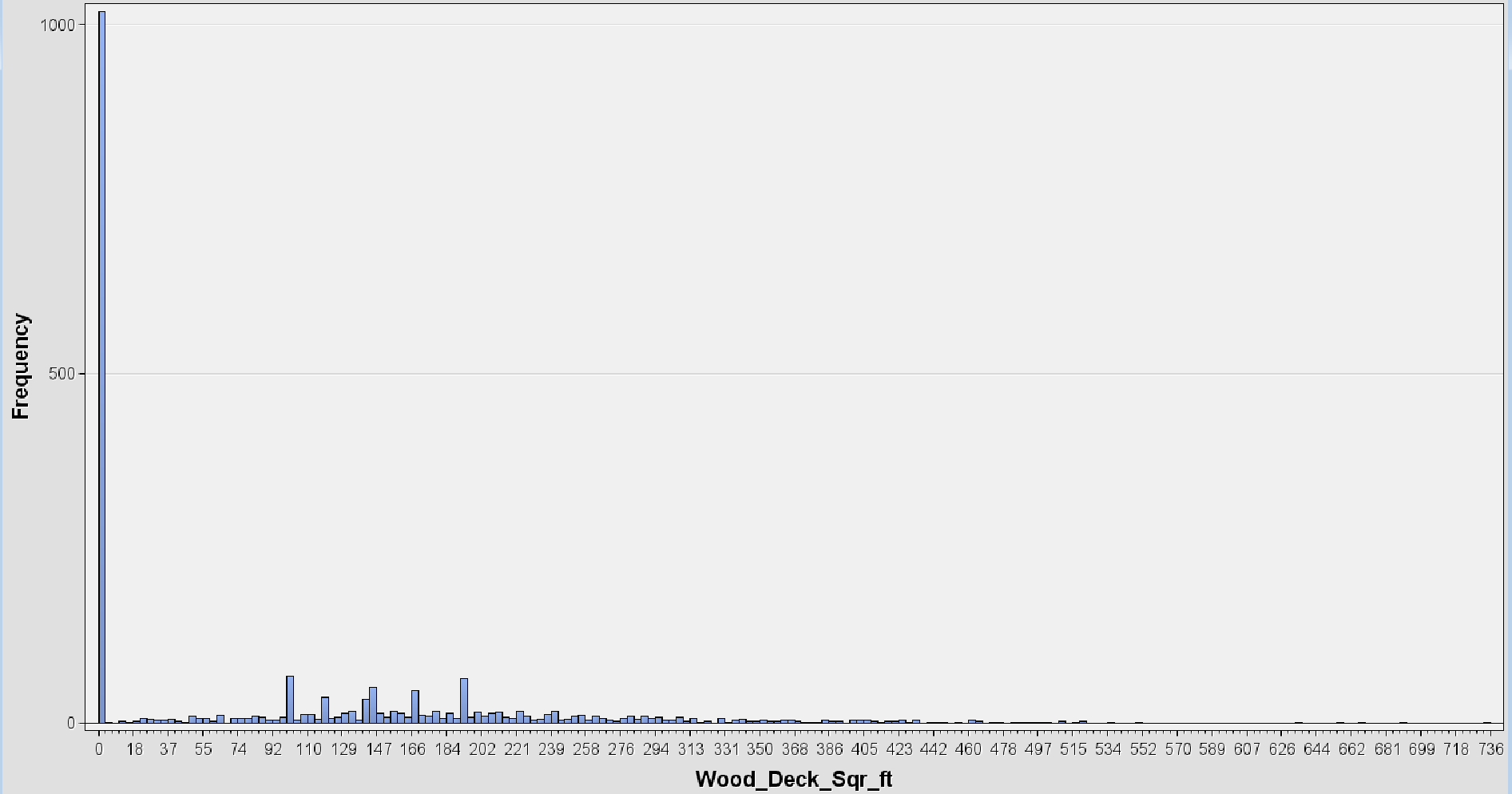
Total basement Sq Ft is an interval variable with values ranging from 190 sq Ft to 3,206 sq Ft. The most common area for basement is from 793 sq Ft to 1095 sq Ft for almost 750 properties.



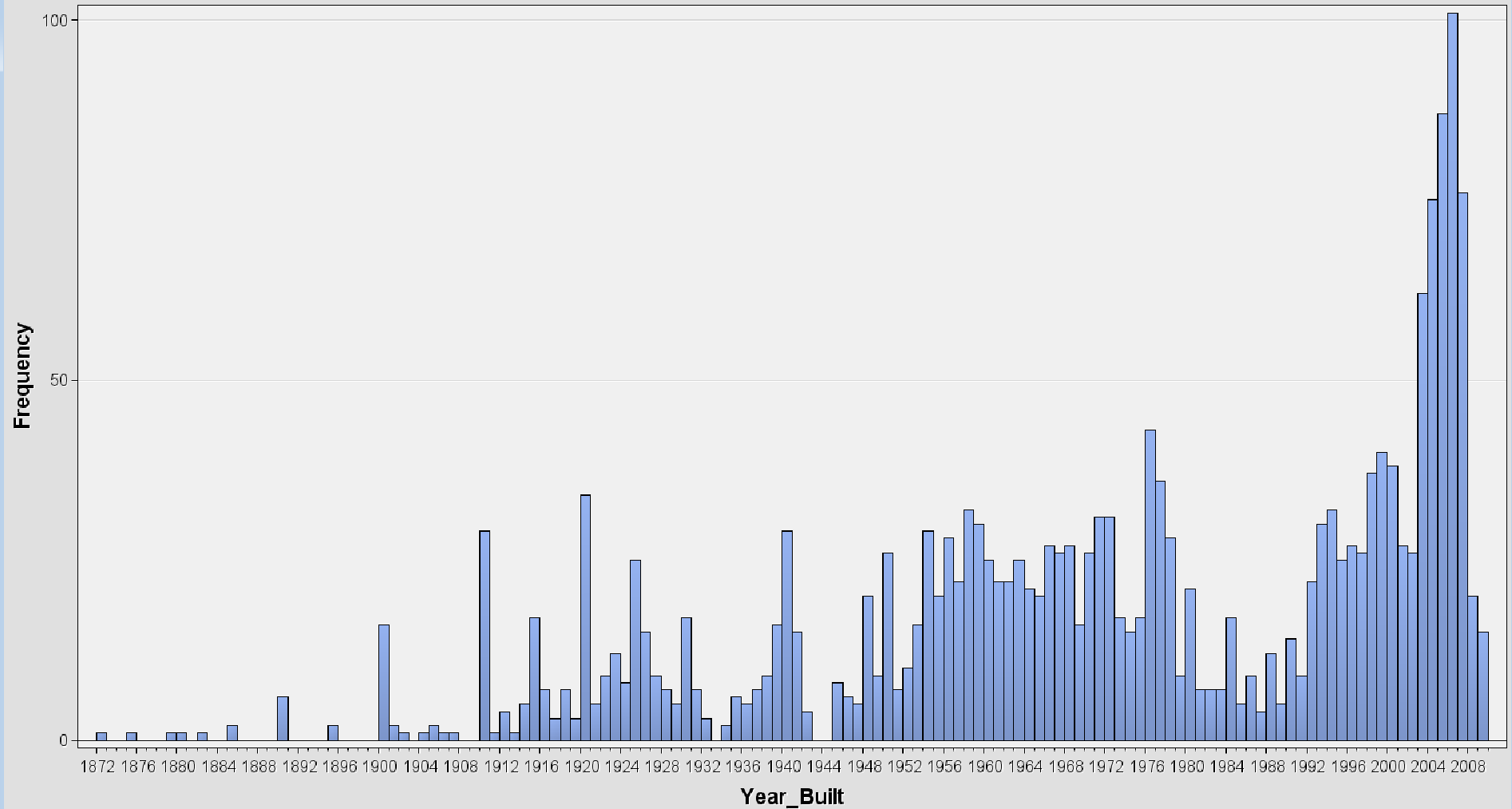
The veneer area of exterior wall is an interval variable that has values ranging from 0 to 1600. 1200 properties have almost no veneer on the exterior wall, which is the most common occurrence.



Wood deck Sq Ft is an interval variable that has the most value of 0. This means that most properties do not have wood decks, and this occurs more than 1000 times.



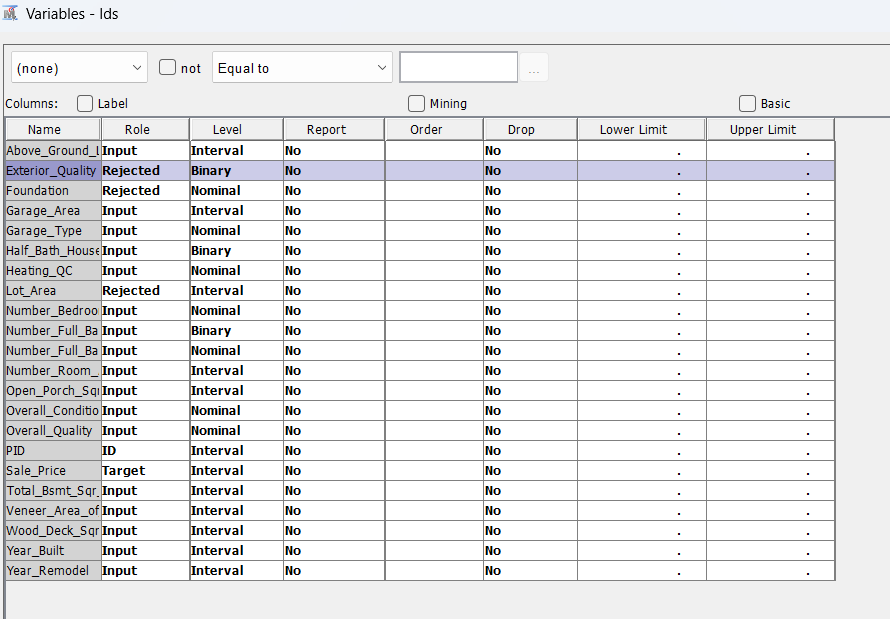
Year built is an interval variable with values ranging from 1872 to 2008. We observe an increasing trend in properties built since 1948. However, the peak occurs between 2004 and 2008, with an average frequency exceeding 70. The highest frequency, with more than 100 properties, is seen in 2006-2007.



Similarly, Year remodel is also an interval variable with values ranging from 1950 to 2010. A peak is indicated by more than 200 properties being remodeled in 1950.



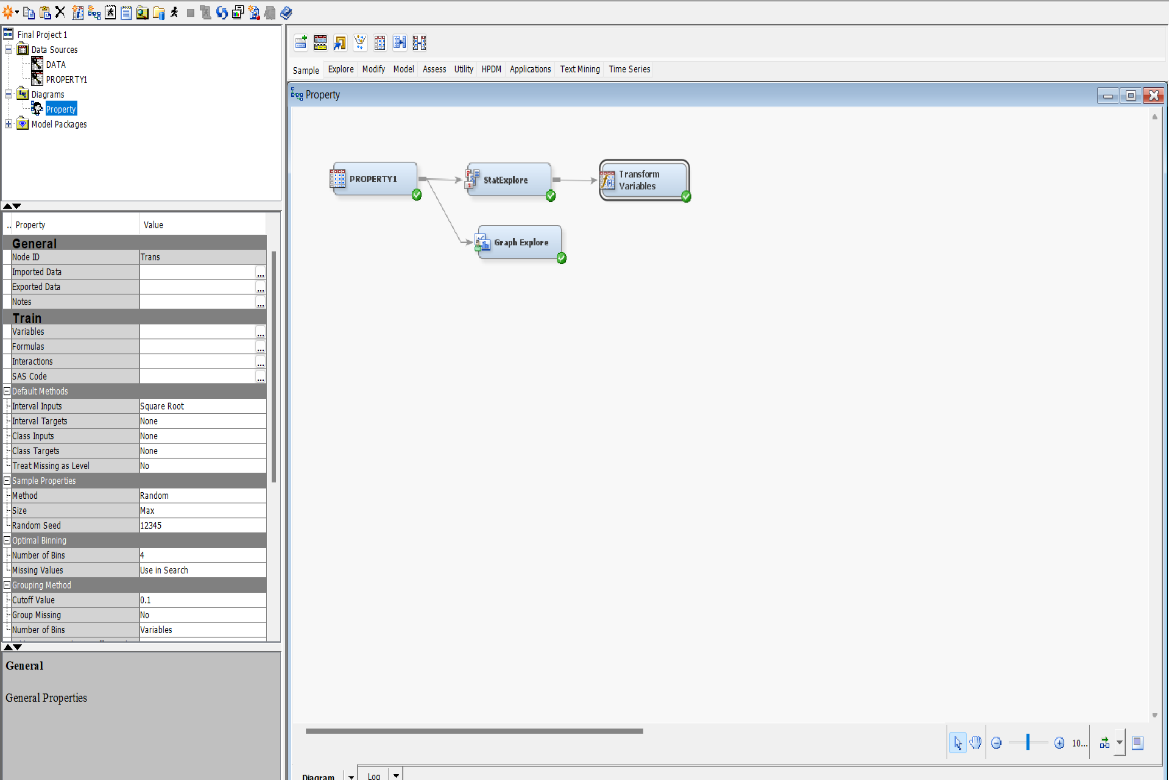
As there isn't a distinct distribution for the Lot area, Exterior quality, and Foundation variables, we've made the decision to exclude them to ensure the input pools remain more manageable. We simply right clicked on the Property1 Data node, selected "Edit variables," and then rejected these variables, as shown in the screenshot below.



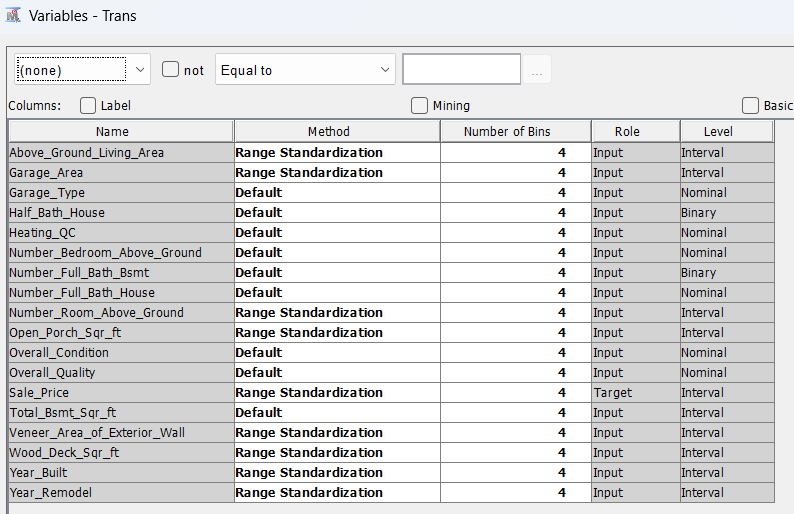
In this phase, we have observed that most of the remaining interval variables are not normally distributed and have varying scales. This poses an obstacle for our algorithm. The next phase will handle this issue.

1. **Data Preparation Phase**

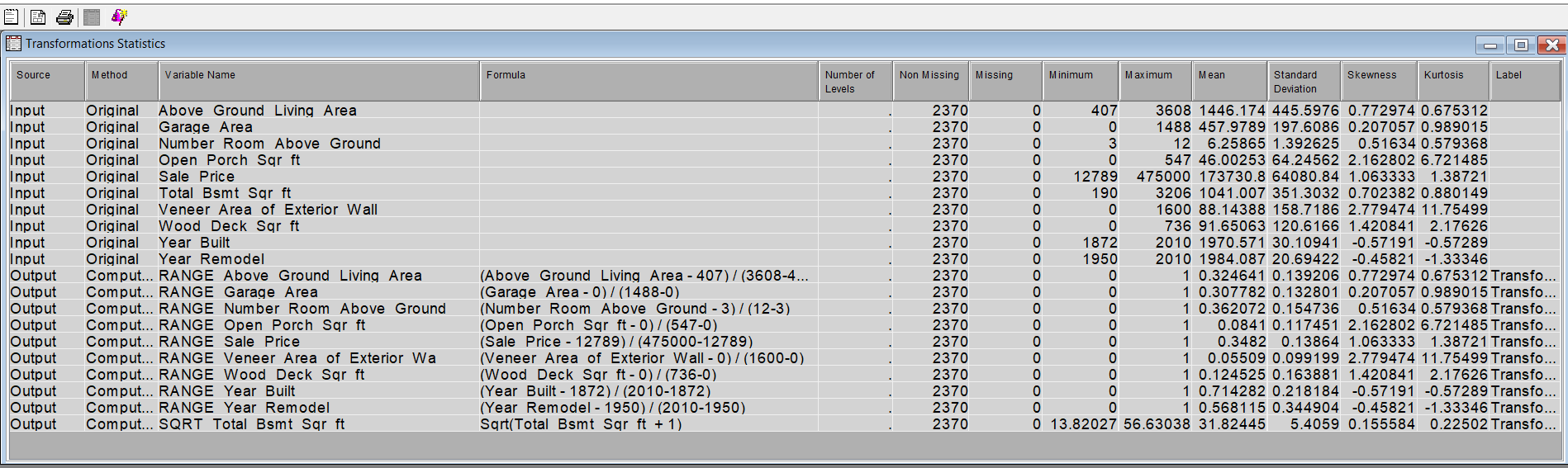
This phase aims to standardize all interval variables, ensuring they are on the same scale and reducing skewness where applicable. Although decision tree algorithms, which we'll utilize later, don't require variable transformation, methods like K-nearest Neighbor and Linear Regression do benefit from it. Performing this step before partitioning the dataset simplifies the project presentation. We accessed the "Transform Variables" node from the "Modify" menu. In the Property Panel, we selected the Square Root method for Interval Inputs to mitigate skewness. We set the Sample Properties size to max (to encompass the entire dataset) and used the random method.



Next, in the "Variables" tab of the Train property, we opted for the Range Standardization method for the interval variables, which encompasses the target variable. This adjustment ensures that the intervals are scaled from 0 to 1.



The below result indicates that interval variables are normalized using Range Standardization, and Square Root method.



1. **Modeling and Evaluation Phase**

In this phase, we decide to perform Decision Tree, K-Nearest Neighbors, and Multiple Regression Algorithm, then select the best model using the validation set.

We chose these three data mining models due to how accurately they measure our data, and it closely relates to what we are looking for.

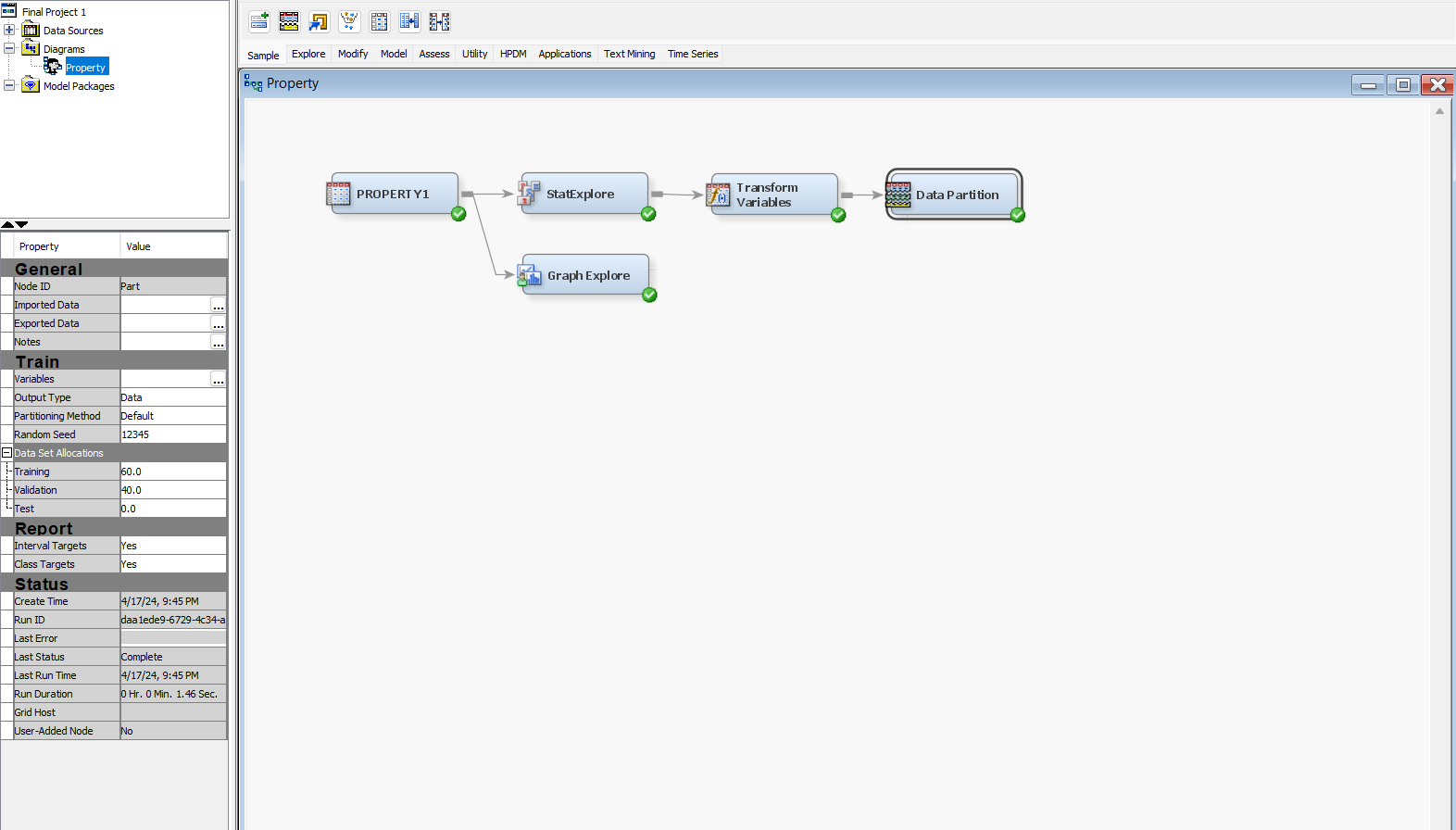
A Decision Tree is a versatile algorithm for classification and regression tasks. It constructs a tree-like structure by repeatedly splitting the data based on features. Predictions are made by traversing the tree from root to leaf. Decision Trees are easy to interpret but prone to overfitting. Techniques like pruning and ensemble methods enhance their performance and robustness.

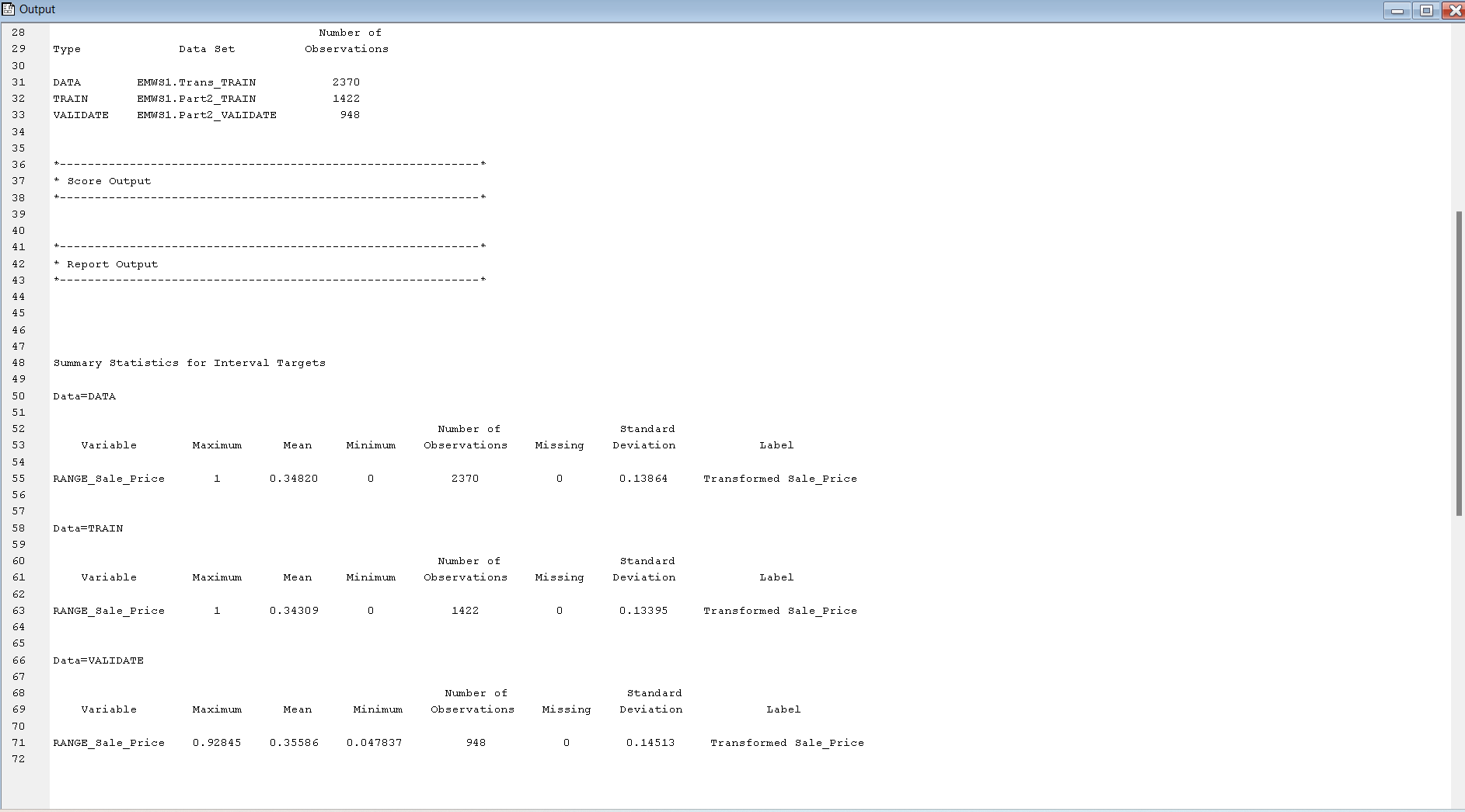
KNN is a simple algorithm for classification and regression. It stores all data points in training and predicts based on the closest neighbors. It's easy to understand but sensitive to parameter choice and data characteristics.

Multiple regression extends simple linear regression to analyze relationships between multiple independent variables and a dependent variable. It estimates coefficients to minimize the difference between observed and predicted values. Interpretations focus on the impact of each independent variable on the dependent variable. Assumptions include linearity, independence of errors, constant variance, and normality. Evaluation metrics include R^2 and significance tests. It's widely used in various fields but requires careful interpretation and consideration of assumptions.

-Data partition:

Given the relatively modest size of this dataset, we chose to partition it into a 60/40 ratio, allocating 60% of the data to the training set and 40% to the validation set. To achieve this, we imported the Data Partition node from the Sample button on the top Toolbar. Within the Train property, we specified the allocations in the Data Set Allocation section, as depicted in the screenshot below.

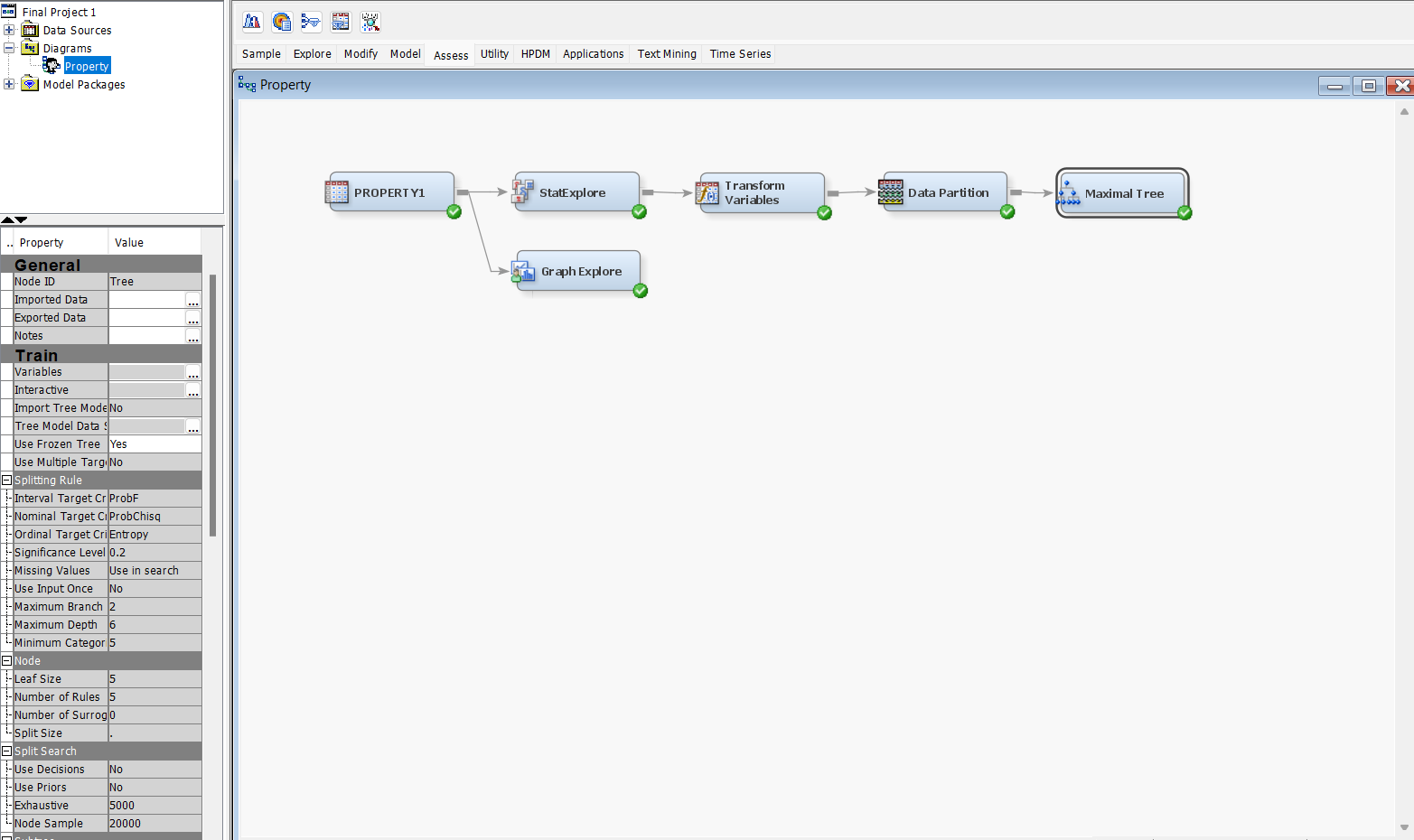




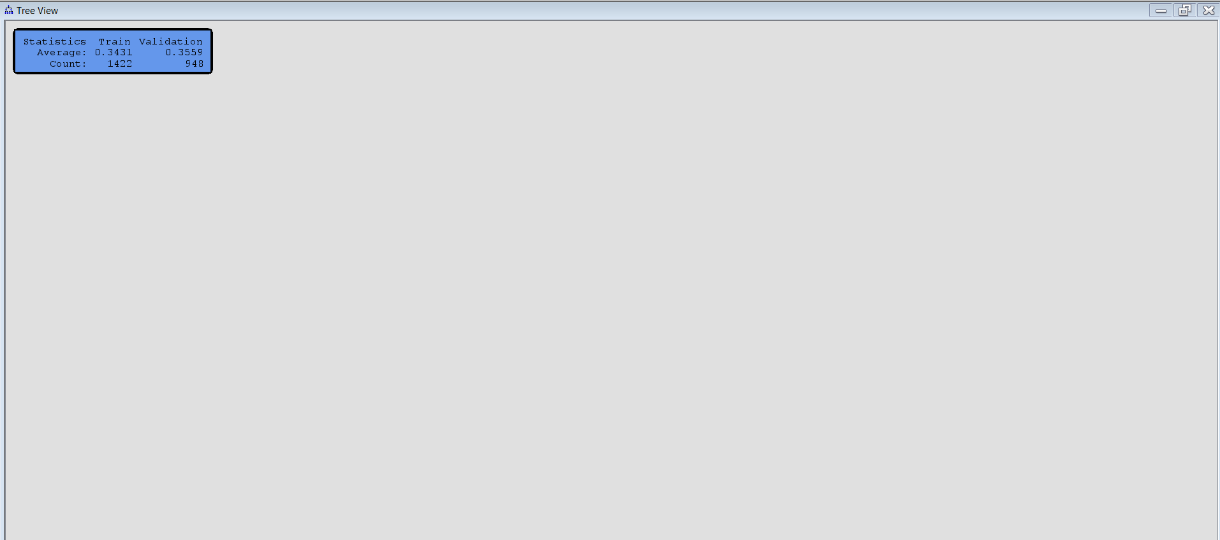
**Decision Tree:**

MAXIMAL TREE

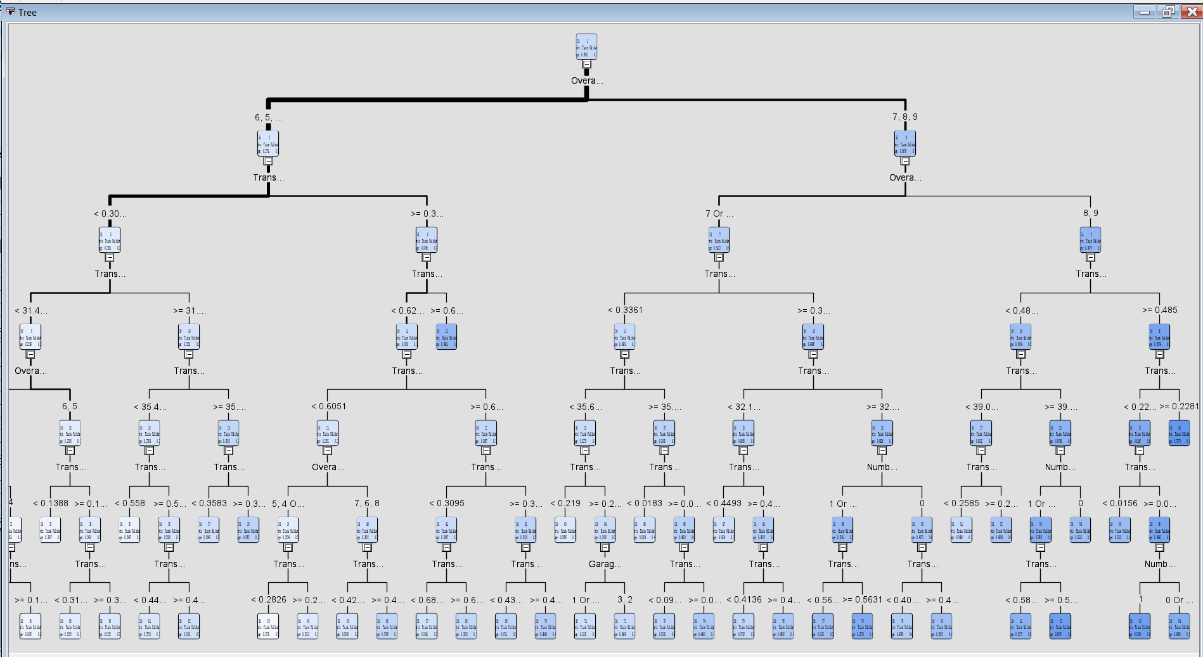
We proceeded with the maximal decision tree by importing the Decision Tree node from “Model”.



From Train-Interactive, we opened this window and ran the Train Node option from the root node.



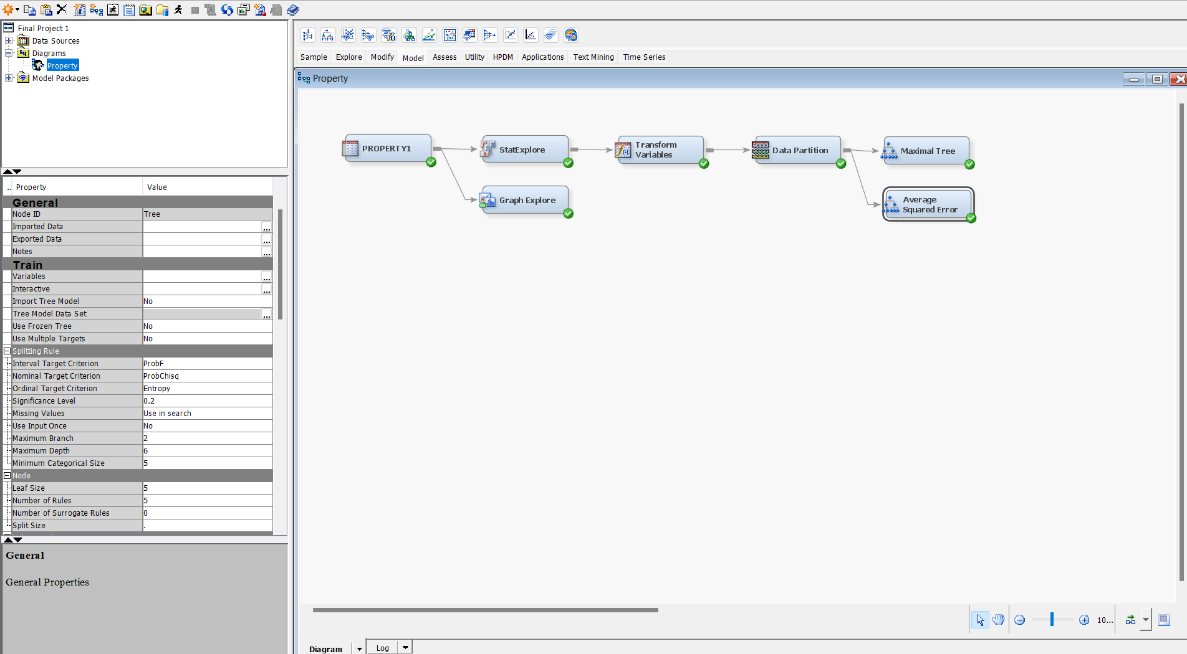
We later obtained the maximal tree created by SAS.



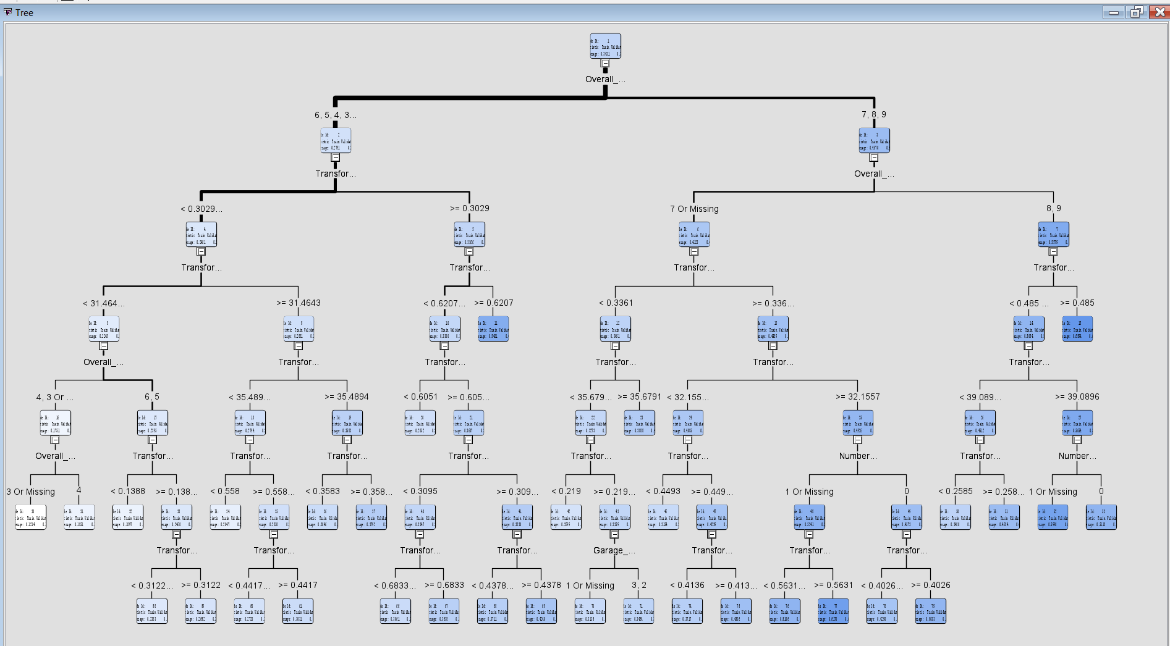
Then we froze this tree by choosing yes for “Use frozen Tree” property.

BEST CLASSIFICATION TREE:

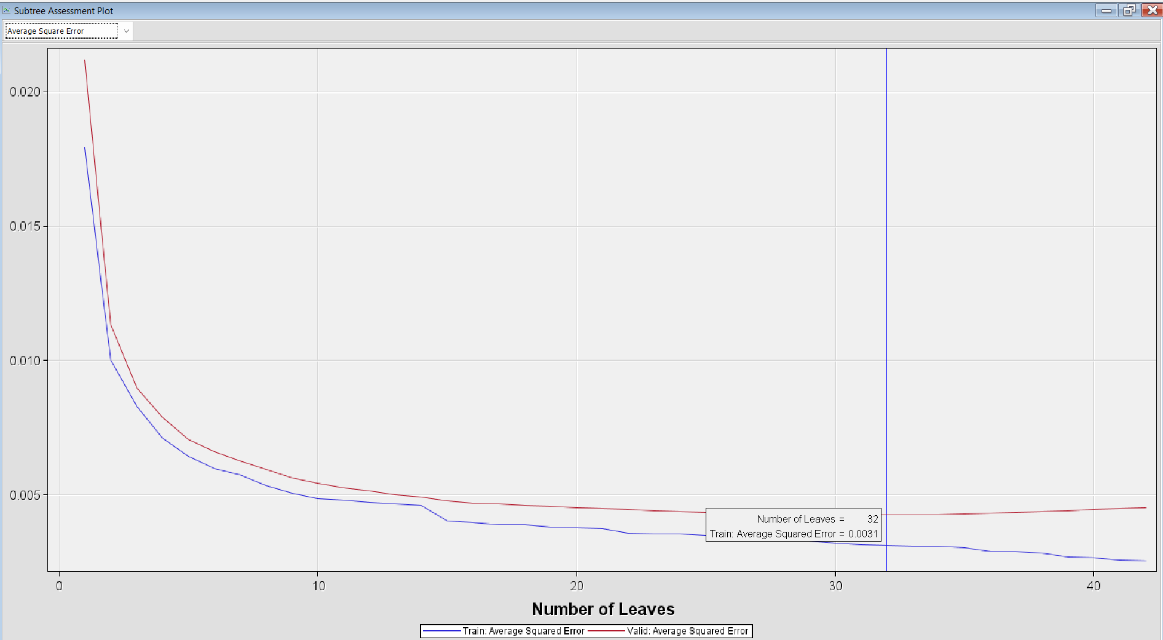
Subsequently, we sought to assess whether pruning branches based on Average Square Error could enhance the model's performance. To do so, we generated a second decision tree, utilizing Average Square Error as the performance metric. We followed the same procedures outlined previously but selected the Misclassification option for the Assessment Measure. The rationale behind choosing Average Square Error as our target variable is because it is interval, rendering this method appropriate for our analysis.



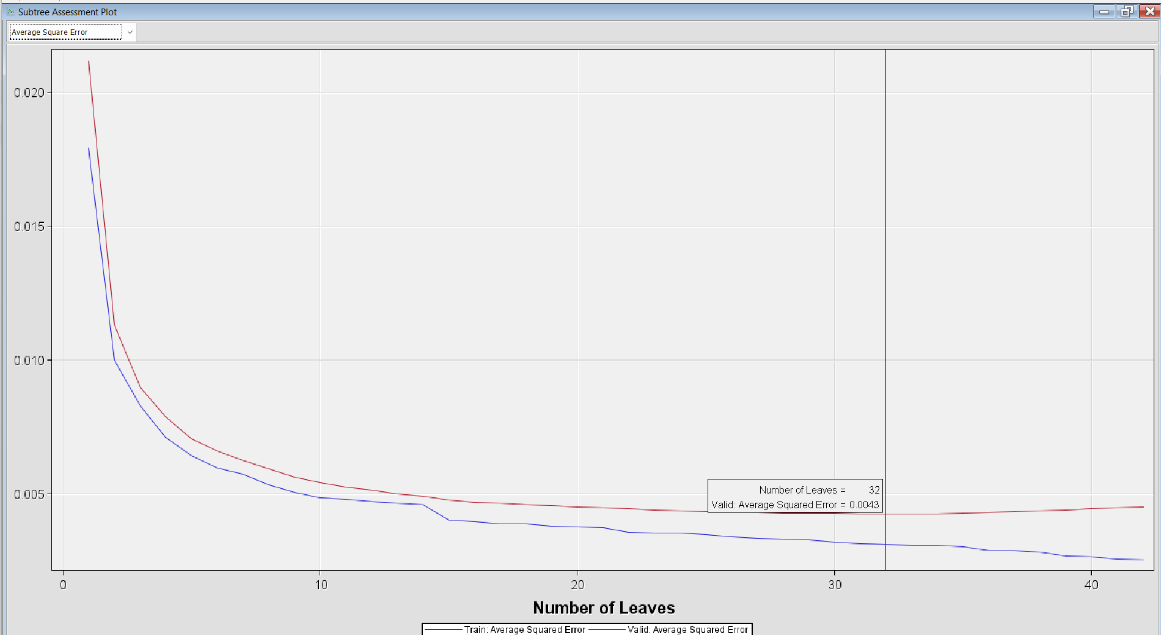
By running the node, we obtained the tree below.

Likewise, we froze this tree by choosing yes for “Use frozen Tree” property.

From View – Subtree Assessment Plot, we can see that the optimal model is reached at 32 leaf nodes. Average Squared Error at the training set is 0.0031.

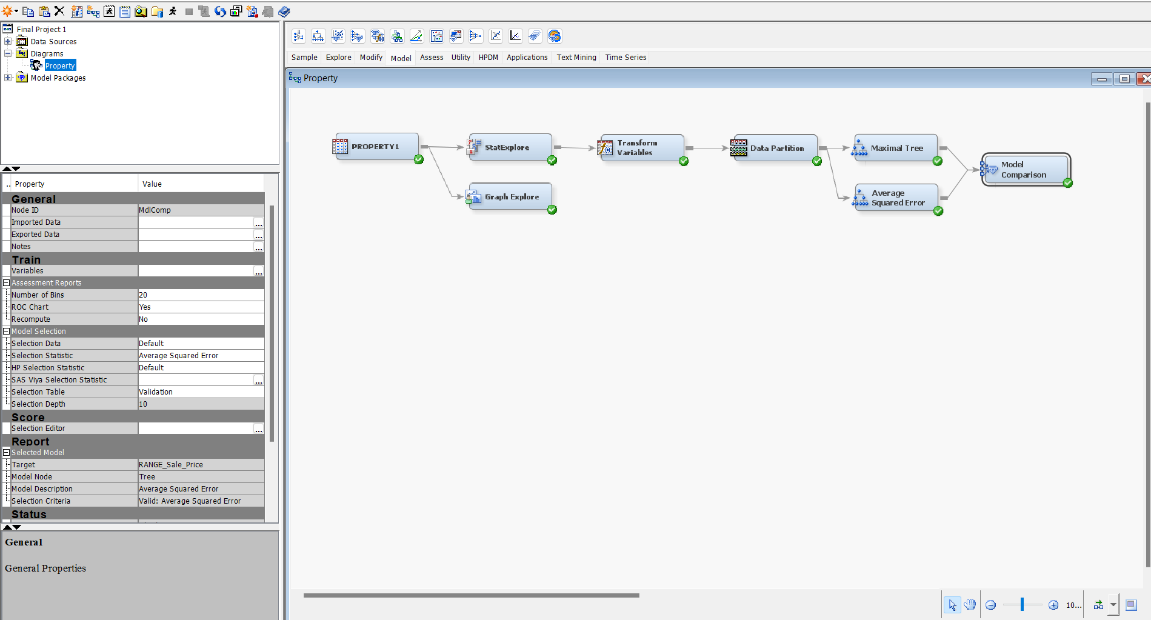


And that of the validation set is at 0.0043.

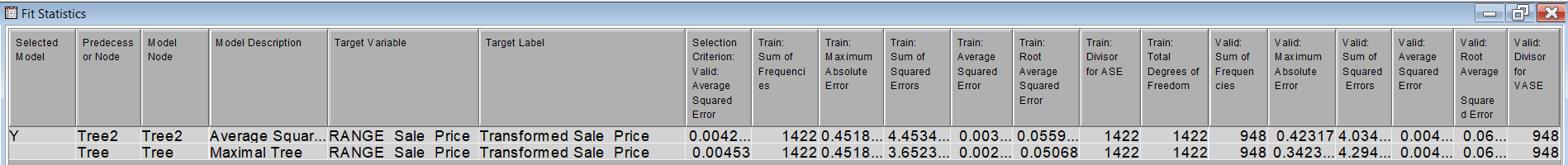


**Model Comparision.**

We imported the Model Comparision node from Assess. On the property panel, the Selection Statistics is Average Squared Error as our target variable is interval. This Comparision is performed on the Validation set, so “Selection Table” property is “Validation”.

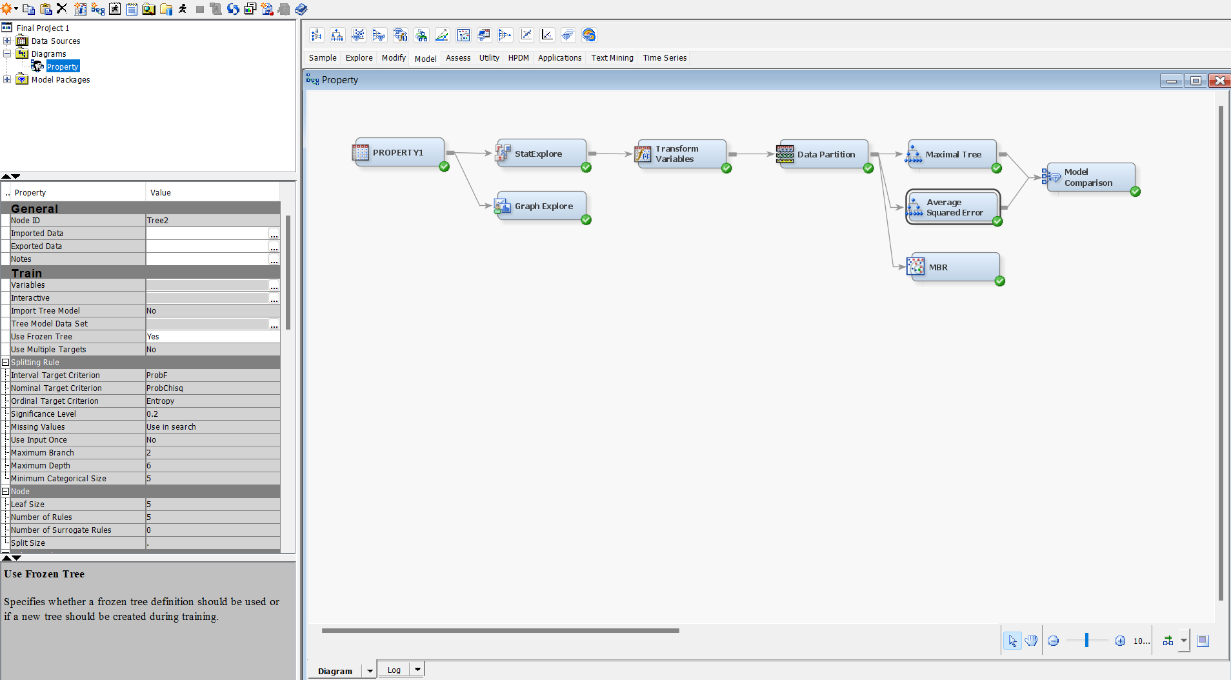


The Fit Statistics Result reveals that Average Squared Error Tree was selected as it has a lower average squared error compared to the other one.

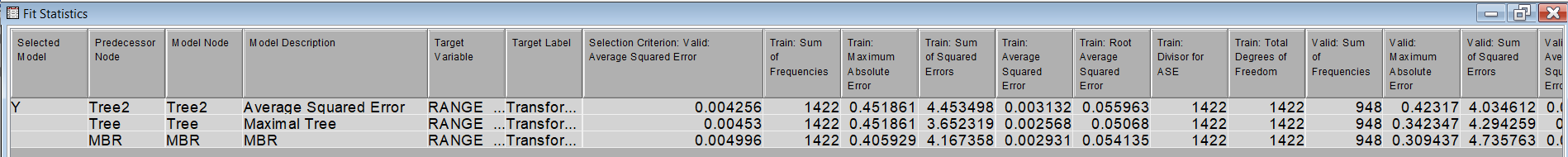


**K-Nearest Neighbors:**

We imported MBR node from Model, choosing 3 for number of neighbors of the Train property. However, this number became 4, as SAS identifies the data point itself as a neighbor.



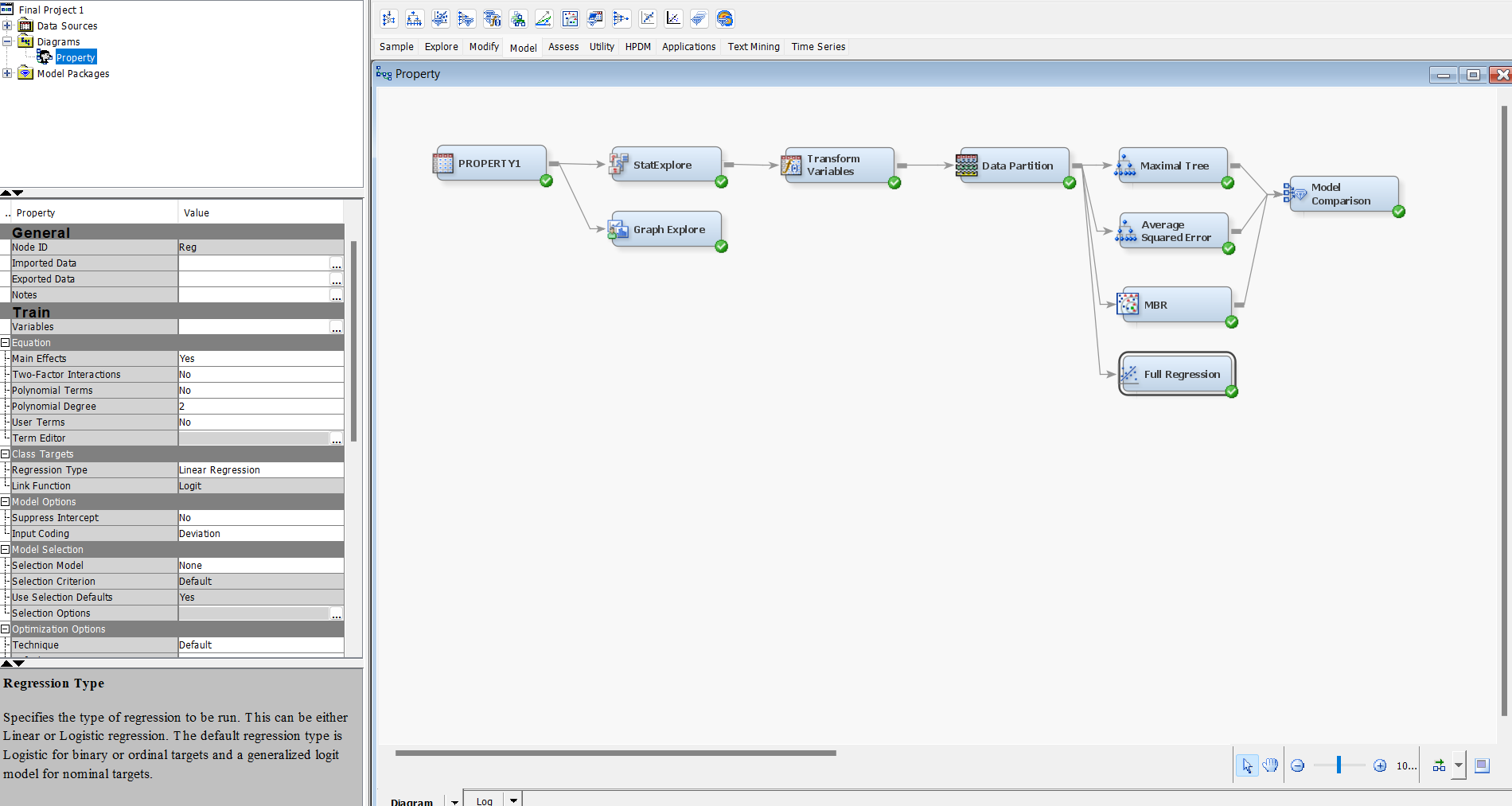
Then we connected this node with the Model Comparision node and run the Model Comparision node again, the result is below.

 Average Squared Error Model is still the selected one due to its lowest Average Squared Error.

**Multiple Regression**

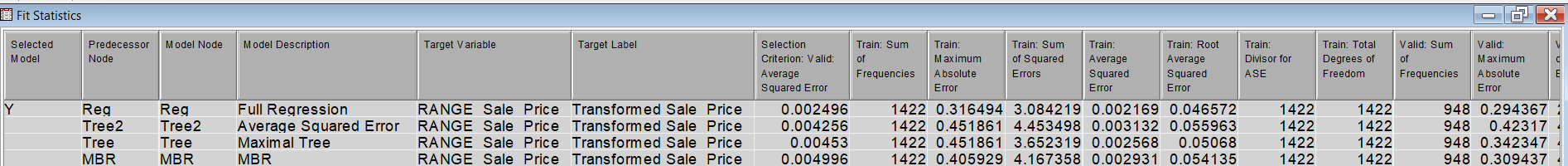
Since we had more than one input variable, we ran Multiple Regression (MR) instead of Simple Linear Regression. In addition, MR has a higher R-Square which demonstrates the model’s reliability.

FULL REGRESSION

We imported the Full Regression node from Model. On the Regression Type property, we selected Linear Regression as the target variable is interval. 

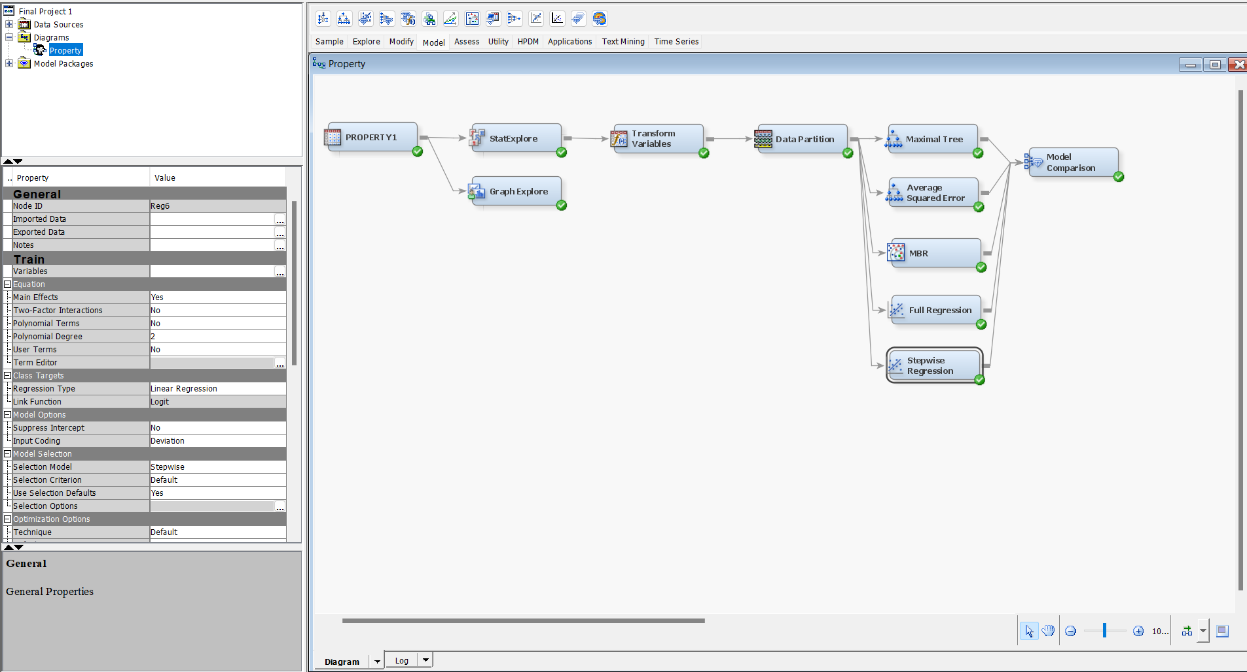
After running this node, we then connected it to the Model Comparision node.

In the below result window, Full Regression model was selected due to its lowest Average Squared Error.

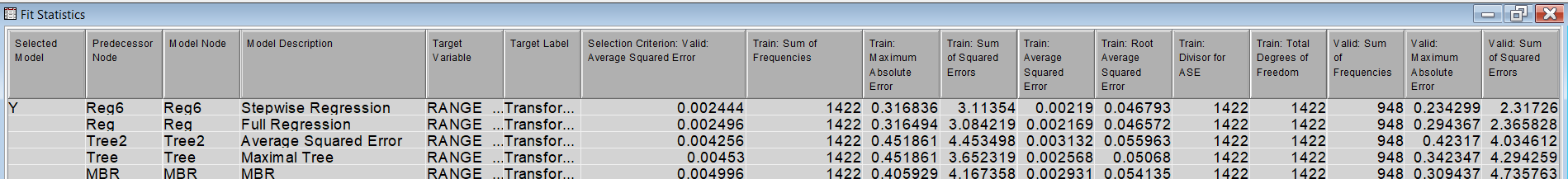


STEPWISE REGRESSION

Likewise, we created a Stepwise Regression node by repeating the above step and chose “Selection Model” as Stepwise. Then connected it to the Model Comparision node.



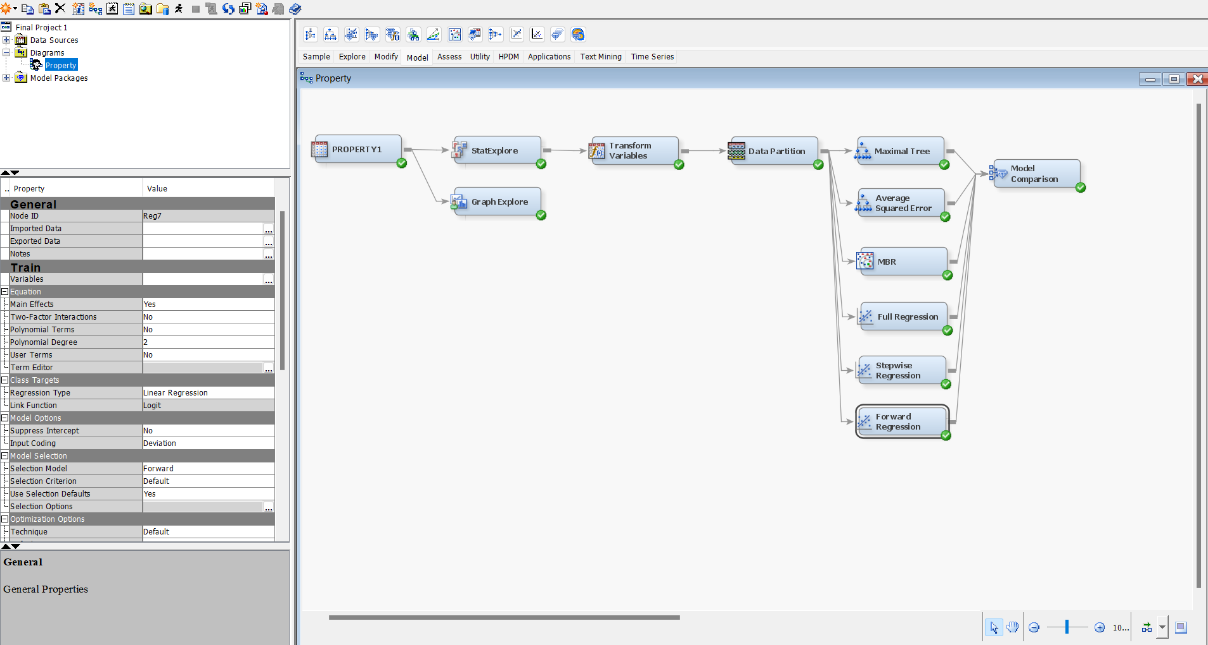
We obtained the result in which this model was selected.



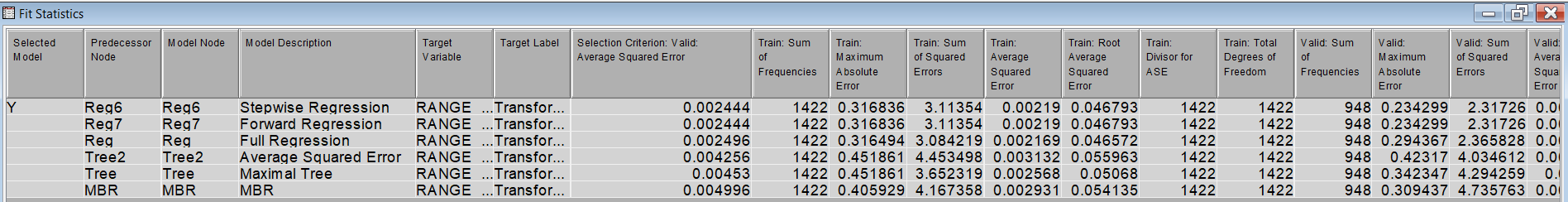
Based on the preceding outcome, the selected model exhibits a slight variance in Average Squared Error compared to the previously chosen model. To explore further, we opted to execute one additional model to ascertain whether it yields a lower Average Squared Error. If not, we will conclude our analysis with the Stepwise Regression model. The final model scheduled for execution is Forward Regression.

FORWARD REGRESSION

We also repeated the steps as we created the Stepwise Regression node, but the “Selection Model” is Forward as the below screenshot.



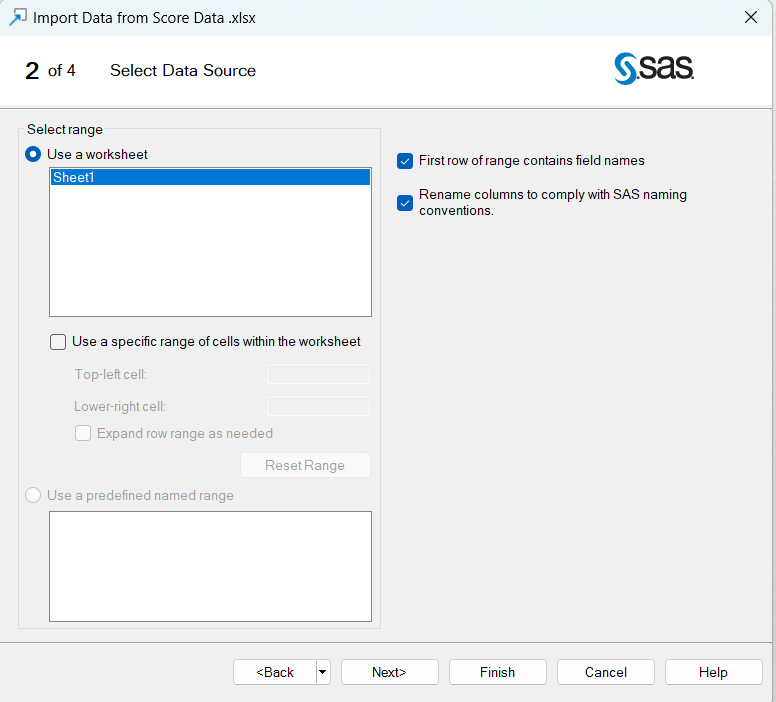
The result presents that Stepwise is still the selected model, and the Selection Criteria did not improve.

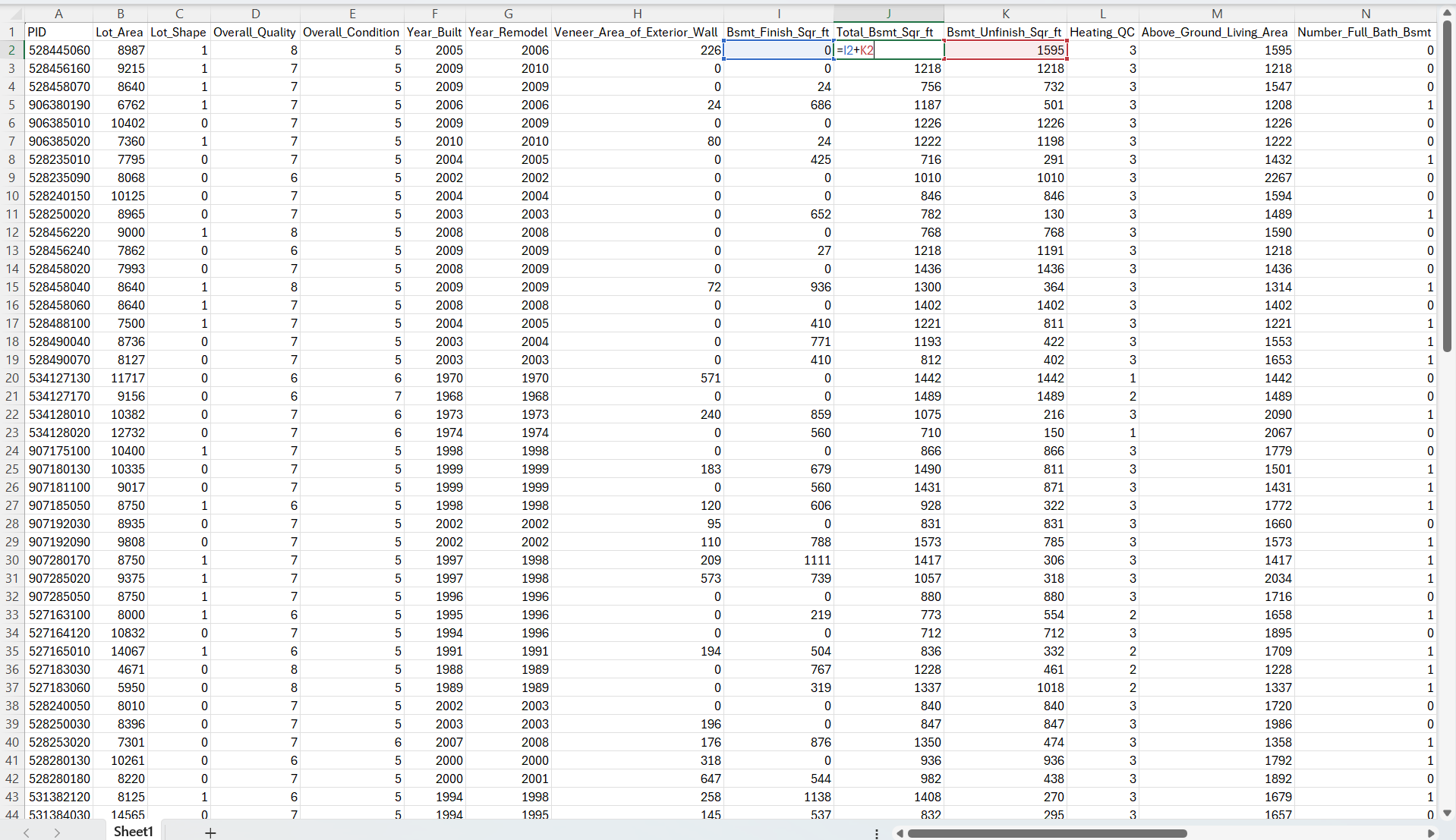


We concluded our Model Selection process by selecting Stepwise Regression for the Deployment Phase.

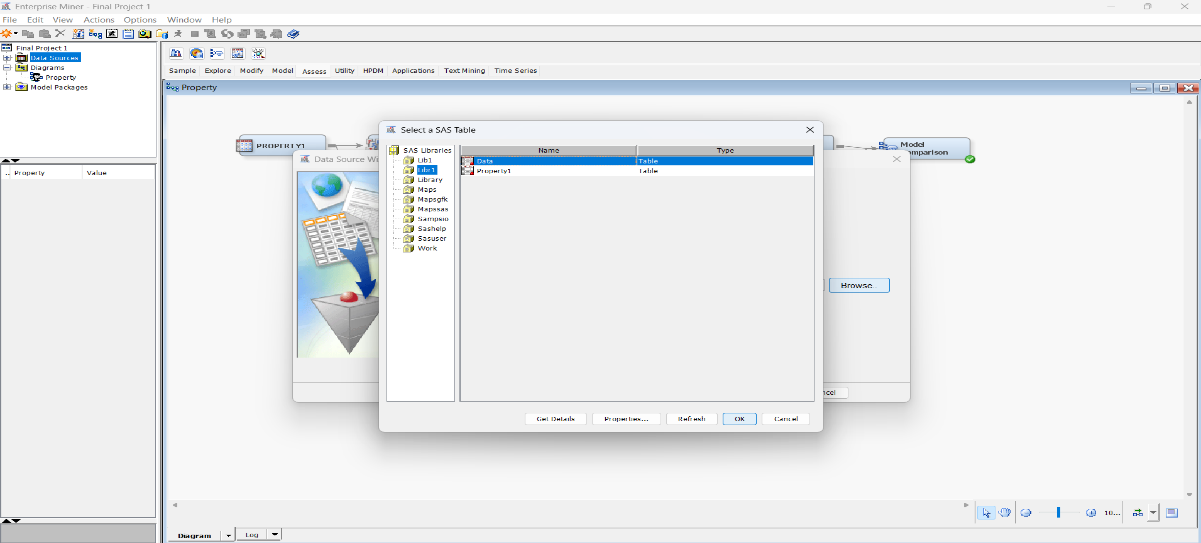
1. **Deployment Phase**

In assessing the significance of the Total Basement Sq Ft variable, which is absent in the score dataset, we undertook a transformation process. This involved creating the new variable Total Basement Sq Ft by amalgamating two existing variables, Basement Finished Sq Ft and Basement Unfinished Sq Ft, using Excel. Subsequently, we reintegrated the modified Excel file into SAS Enterprise Guide.

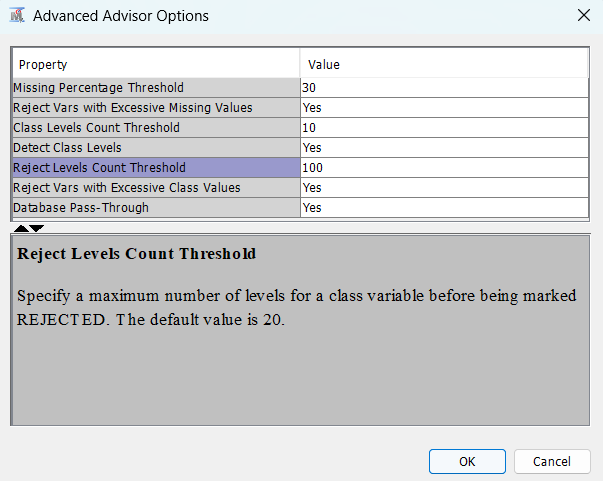




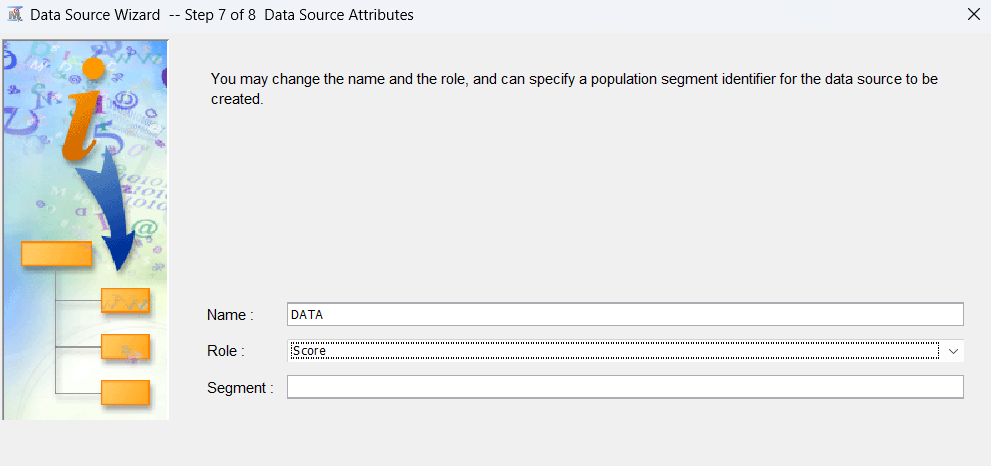
Then we imported the score dataset named Data into the project Data Source using the same format, same variable roles and levels as the 2 screenshots below.



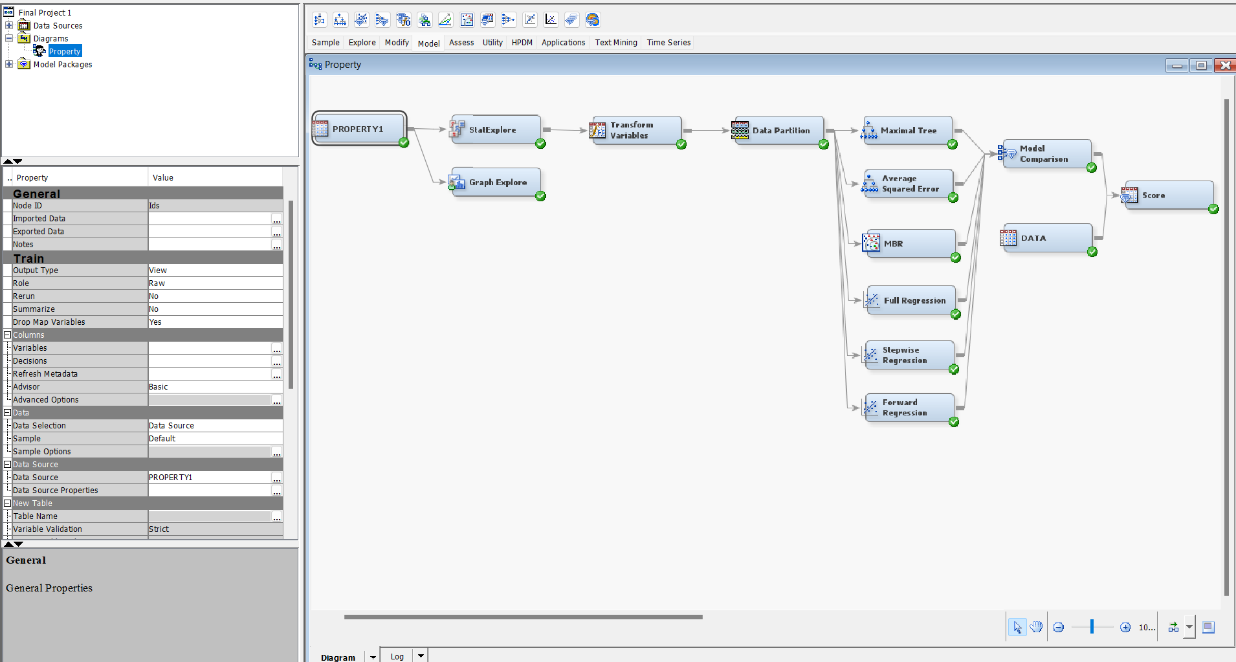
We kept the same setting as the 1st dataset.



The role for this dataset is Score as it will be run on top of our selected model.

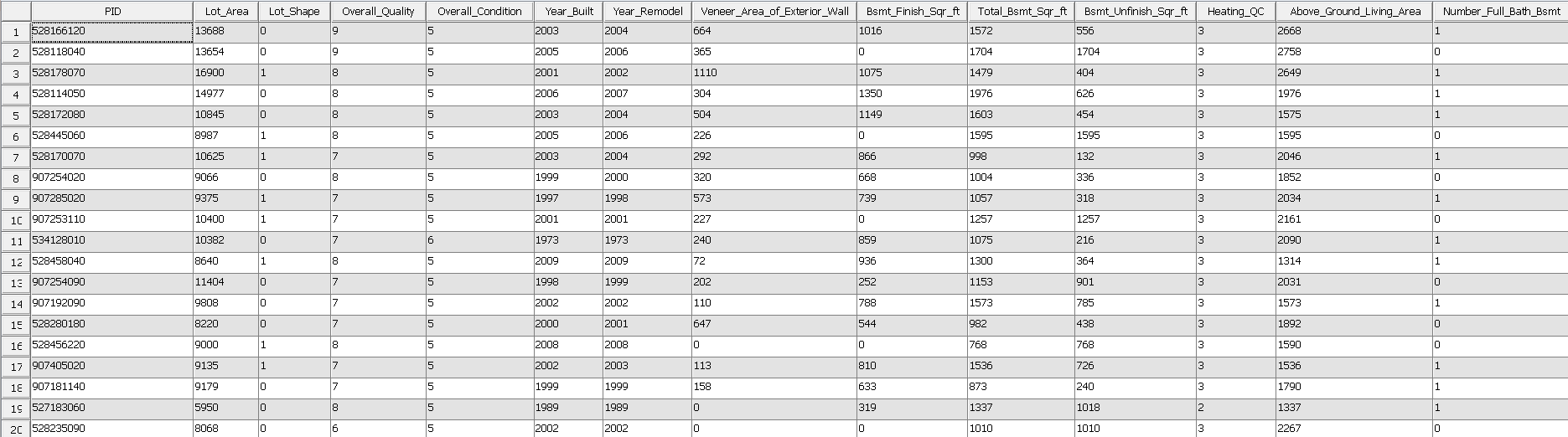


Then we dragged the score dataset onto the Diagram Workspace and imported the Score Node from Assess. Next, we connected the Data node, and Model Comparision Node with Score node to obtain the result.

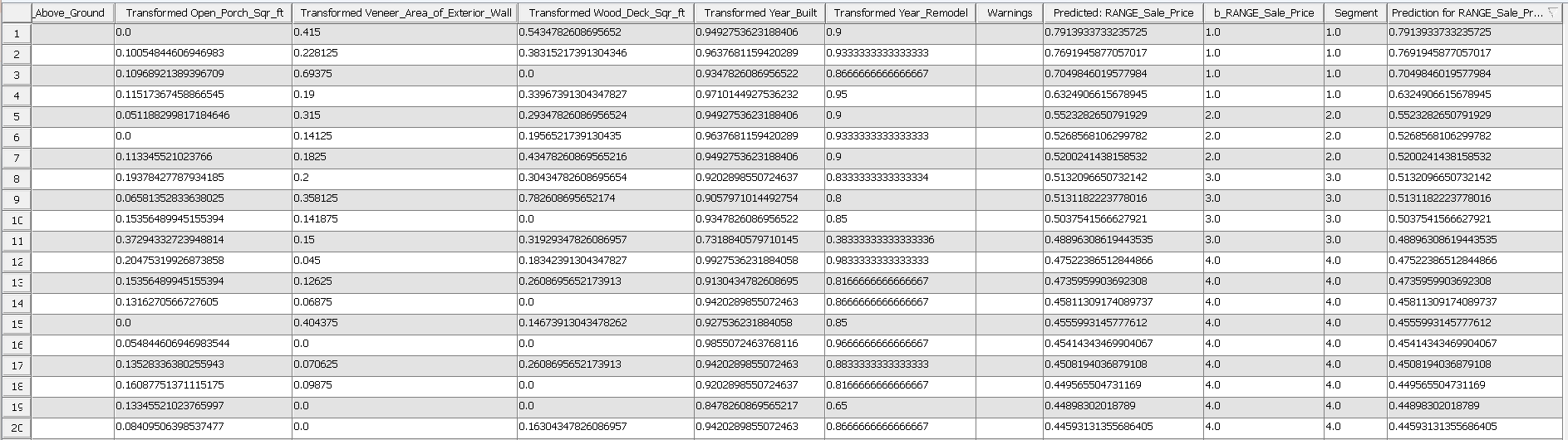


Then we clicked on the Exported data of the Score node to see the predicted prices ranking for the 100 properties on the Score Dataset. The result is below.

This is the list of the most expensive properties ID



And their predicted prices.



**Conclusion**

This report outlines the application of predictive analysis algorithms within SAS Enterprise Miner software to forecast property prices among a list of 100 properties and identify the top 20 highest-valued ones. The analysis involved the utilization of two distinct datasets: the first dataset, comprising 2370 records and 19 variables, was employed for model creation, while the second dataset, featuring 18 variables, was utilized for scoring the target variable.

Throughout the project lifecycle, a series of essential steps were undertaken, encompassing data explorations, data transformations, application of machine learning algorithms, and subsequent evaluation processes, aimed at identifying the optimal model for scoring the dataset.

Several key insights emerged from this endeavor. Notably, the initial phase of data understanding emerged as pivotal, ensuring the avoidance of unnecessary iterations and adjustments later in the process. Moreover, it was observed that decision tree algorithms exhibit robustness in handling varying interval variables without requiring them to be standardized to the same range. In contrast, methods such as KNN and Regression Analysis impose stricter requirements concerning the distribution and range of variables. Additionally, to facilitate seamless deployment, it was imperative to ensure that variables across both datasets maintained identical roles and levels.

These findings underscore the significance of meticulous data preparation and algorithm selection in optimizing predictive analysis outcomes and highlight actionable considerations for future endeavors in similar domains.