

# Tutorial 1: Data Management

In this session, the goal is using linear regression to understand the relationship of attributes affect the price of diamond.

## Data Information

This classic dataset contains the prices and other attributes of almost 54,000 diamonds. This datasets is available from <https://www.kaggle.com/shivam2503/diamonds> which contribute by shivamagrawal (2017).

The dataset explores the price of diamonds affected by various of variables.

The dataset include 10 variables for exploring including price variable.

### The Attributes:

**Carat:** Carat weight of the diamond.

**Cut:** Cut quality of the diamond. (High - Low quality : Ideal, Premium, Very Good, Good, Fair)

**Color:** Color of the diamond. (Best - Worst: D - J)

**Clarity:** The absence of the inclusions and blemishes. (Best - Worst: IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1)

**Depth:** The height of a Diamond, measured from the Culter to the table, divided by its average Girdle Diameter.

**Table:** The width of the diamond's table expressed as a percentage of its average diameter.

**Price:** The price of the Diamond

**X:** Length of the Diamond in mm.

**Y:** Width of the Diamond in mm.

**Z:** Height of the Diamond in mm.

## HOME MODULE

Select New Project create new project: Project Name: Diamond

Select Project Type: Data Analytics

Description: Write the project description

## DATA UPLOAD MODEL

Upload Data: diamonds.csv

# DATA MANAGER MODEL

Raw data and data types need to be checked before processing. The processing can be done in **Data Processing**.

## Using Filter to handle missing value

In this case, we need to go to **Filter**

The screenshot shows the 'Data Processing' tab in the Data Manager Model. On the left, a list of functions is available, with 'Filter' highlighted. The main area displays a data table with the following columns: carat, cut, color, clarity, depth, table, price, x, y, z, and index. The table contains 20 rows of data. The 'x', 'y', and 'z' columns contain values representing the length, width, and height of the diamond in mm, respectively. The 'depth' column contains values representing the depth of the diamond in mm. The 'price' column contains values representing the price of the diamond. The 'index' column contains values representing the index of the diamond.

carat	cut	color	clarity	depth	table	price	x	y	z	index
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.08	2.81	1
0.21	Premium	E	SI1	59.0	61	326	3.89	3.04	2.81	2
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.81	3
0.29	Premium	I	VS2	62.6	58	334	4.2	4.23	2.83	4
0.31	Good	F	SI2	64.3	58	335	4.84	4.85	2.75	5
0.24	Very Good	F	VS2	62.8	57	336	3.94	3.06	2.48	6
0.24	Very Good	I	VS1	62.3	57	336	3.95	3.08	2.47	7
0.26	Very Good	H	SI1	61.9	55	337	4.07	4.11	2.53	8
0.22	Fair	E	VS2	65.1	61	337	3.87	3.78	2.49	9
0.23	Very Good	H	VS1	59.4	61	338	4	4.05	2.39	10
0.3	Good	F	SI1	64	55	338	4.25	4.28	2.73	11
0.23	Ideal	F	VS1	62.8	56	348	3.89	3.9	2.46	12
0.22	Premium	F	SI1	63.6	61	342	3.88	3.84	2.31	13
0.31	Ideal	F	SI2	62.2	54	344	4.35	4.37	2.71	14
0.2	Premium	E	SI2	63.2	63	345	3.79	3.75	2.27	15

x, y, z means the length, width, height of the diamond in mm. therefore, 0 value means missing data in these rows. Also, the depth column without zero value, which is calculated by x, y, z. The depth column also can confirm 0 value is missing value in x, y, z columns.

Choose x, y, z columns and select the rows is not 0 separately and select run.

check the number of rows. There have 53940 rows before data filtering and 53920 rows after data filtering, which means there have 20 rows missing data. In this case 20/53940 influence little for the result of the whole dataset. We can filter missing data and save the data directly.

Data Transform

1

Data Processing

diamonds.csv

Module 1

Test

2

x

is not

0

Module 2

Test

2

y

is not

0

Module 3

Test

2

z

is not

0

Reset

Run

53920 rows

carat	cut
0.23	Ideal
0.21	Premium
0.23	Good
0.29	Premium
0.31	Good
0.24	Very Good
0.24	Very Good
0.26	Very Good
0.22	Fair
0.23	Very Good
0.3	Good
0.23	Ideal
0.22	Premium
0.31	Ideal
0.2	Premium

In this case, the new data named as ‘ New\_Diamonds ’.

Save

First Page

carat	cut	color	clarity	depth	table	price	x	y	z	index
0.23	Ideal	E	S12	61.5	55	326	3.95	3.98	2.43	1
0.21	Premium	E	S11	59.8	61	326	3.89	3.84	2.21	2
0.23	Good	F	VS1	58.9	65	327	4.05	4.07	2.51	3
0.29	Premium	I	VS2	62.4	58	334	4.2	4.23	2.63	4
0.31	Good	J	S11	59.4	61	338	4.34	4.35	2.76	5
0.24	Very Good	D	S11	60.4	56	340	3.94	3.96	2.48	6
0.24	Very Good	D	S11	60.4	56	340	3.95	3.98	2.47	7
0.26	Very Good	D	S11	60.4	56	342	4.07	4.11	2.53	8
0.22	Fair	J	S11	58.7	61	342	3.87	3.78	2.45	9
0.23	Very Good	H	VS1	59.4	61	338	4	4.09	2.29	10
0.3	Good	J	S11	54	55	339	4.25	4.28	2.73	11
0.23	Ideal	J	VS1	62.8	56	340	3.93	3.9	2.46	12
0.22	Premium	F	S11	60.6	61	342	3.88	3.84	2.33	13
0.31	Ideal	J	S12	62.2	56	344	4.35	4.37	2.71	14
0.2	Premium	E	S12	60.2	62	345	3.79	3.75	2.27	15

53920 rows

Page 1 of 3595

Previous

Next

Name the Dataset

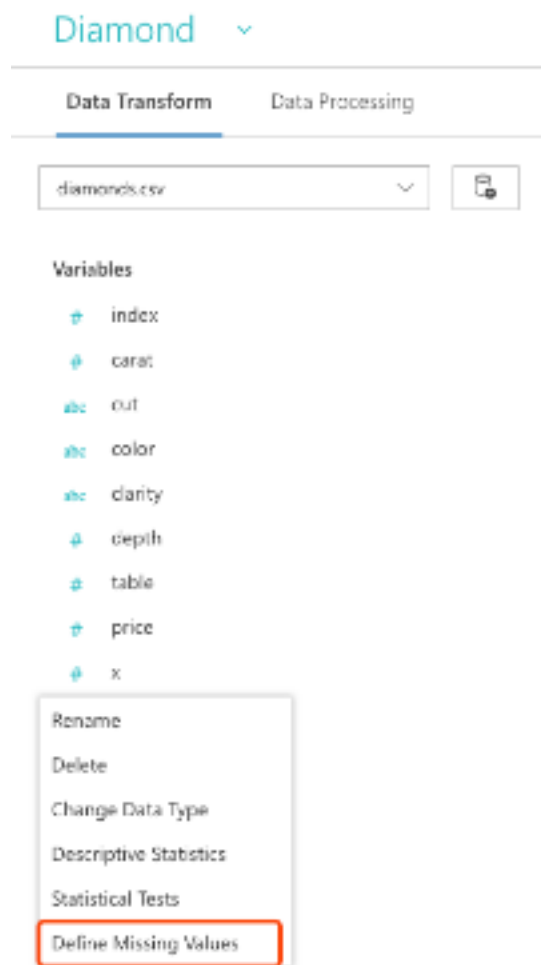
Name: new\_diamonds

Save

Cancel

There has another way to handle few missing value.

Go back to Data Transform and select the variable you want to handle.



Then enter the missing value for further filtering.

Define Missing Values

Variable:

x

Null Values:

0

Enter null values separated by semicolon (;)

Update

Cancel

If you define missing value in variables, the variable name will change to red. It indicates that the variable has missing value.

Data Transform

Data Processing

diamonds.csv

Variables

# index

# carat

abc cut

abc color

abc clarity

# depth

# table

# price

# x

# y

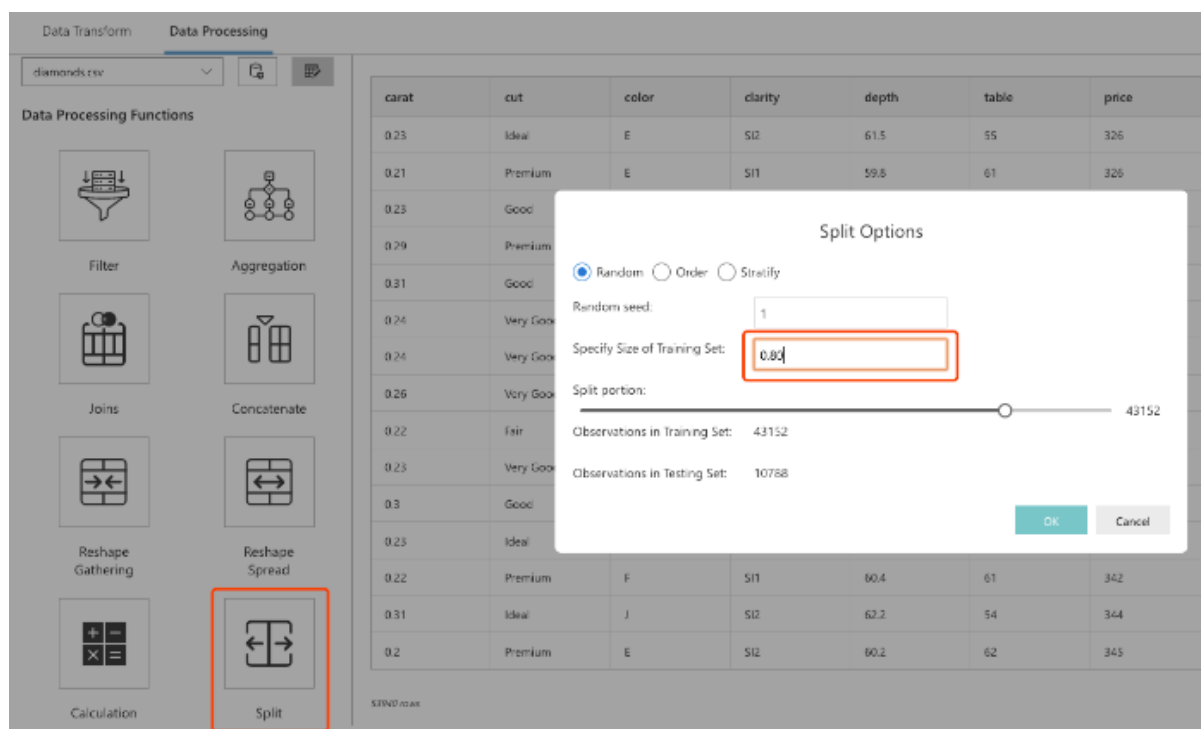
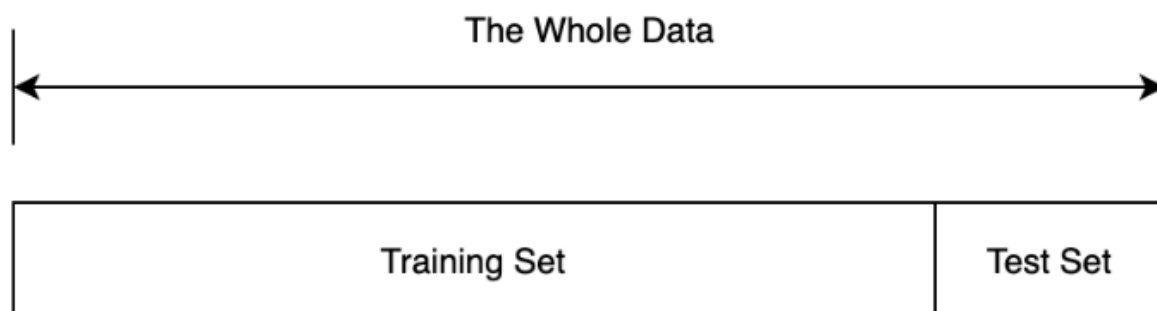
# z

Remember rename the variables if you want to use this method.

## Splitting

Please go to data processing site and select Split.




For Machine Learning, if we want to perform classification or predict statistical models, we can split the data into training and testing sets. Testing set can be used to test if the model will overfit when we use new samples. If a model fit to the training dataset also fits the test dataset well, minimal overfitting has taken place. A better fitting of the training dataset as opposed to the test dataset usually points to overfitting.



Change the specific size of training set from 0.90 to 0.80 which means the whole dataset separate as 80% of training data and 20% of test data. You can change the percentage of the training and testing sets based on your requirement.


Go back to Data Upload module, you can see the split data has been settled.


AutoStat®





Diamond


Uploaded Datasets


 **New Data**

 Search

 diamonds.csv

 new\_diamonds.csv

 new\_diamonds\_train.csv

 new\_diamonds\_test.csv

New Data

From Local

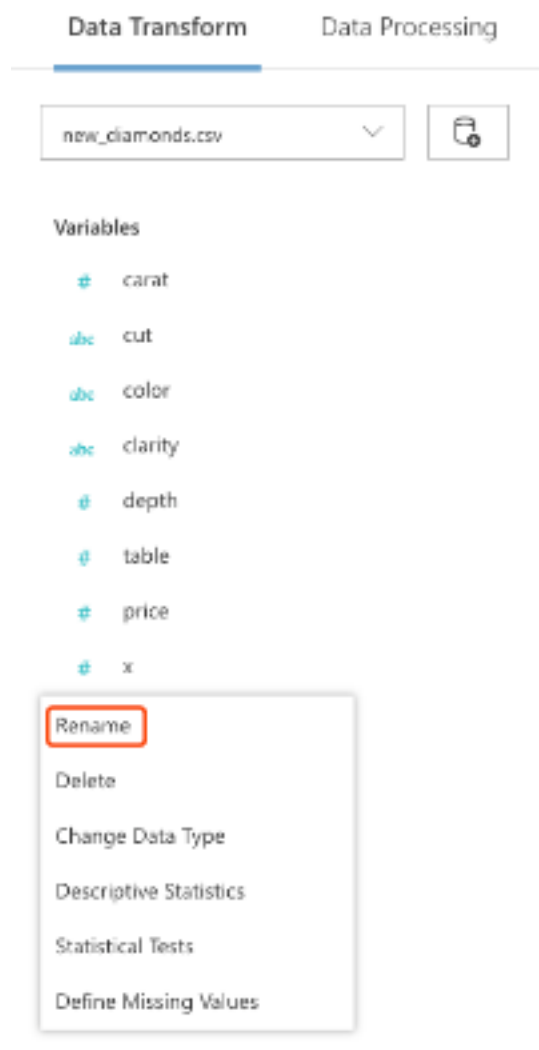
CSV

## Name variables with meaningful words.

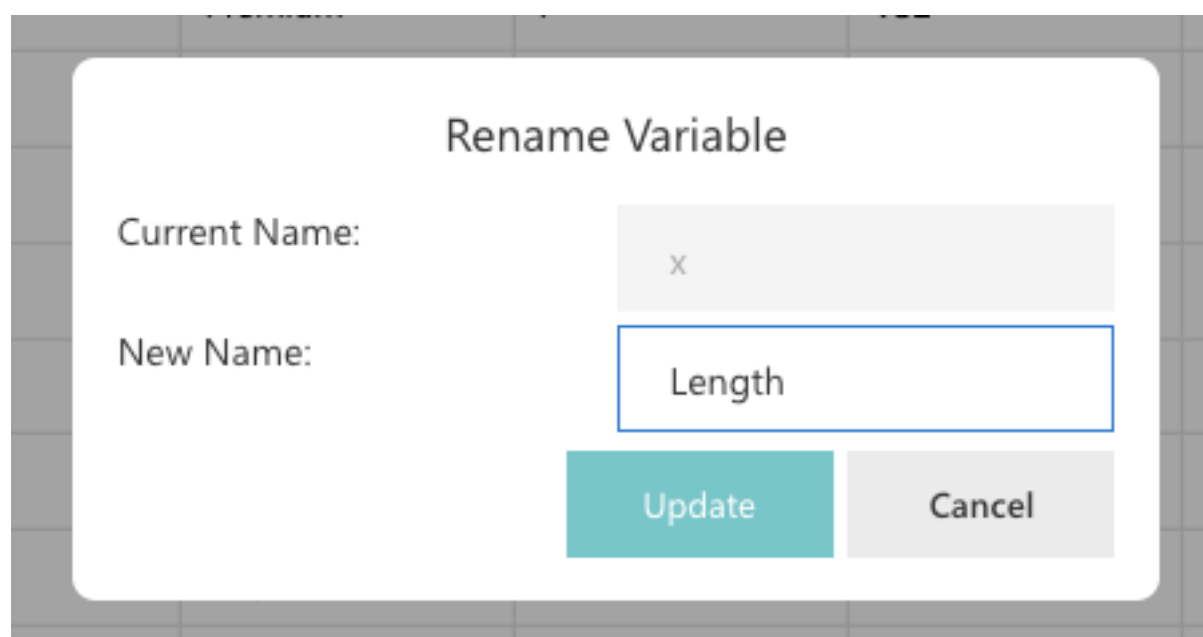
It is easier for us to rename unclear row name for further steps.

In this case, x, y, z as column names are hard for us to remember the meaning. What we can do is to change the column name directly.

Go back to **Data Transform** Select the variable you want to change the name



Insert the new name





Please change y and z as well. The result as below

### Data Transform

new\_diamon... ▾

[

#### Variables

abc

cut

abc

color

abc

clarity

#

depth

#

table

#

price

#

Length

#

Width

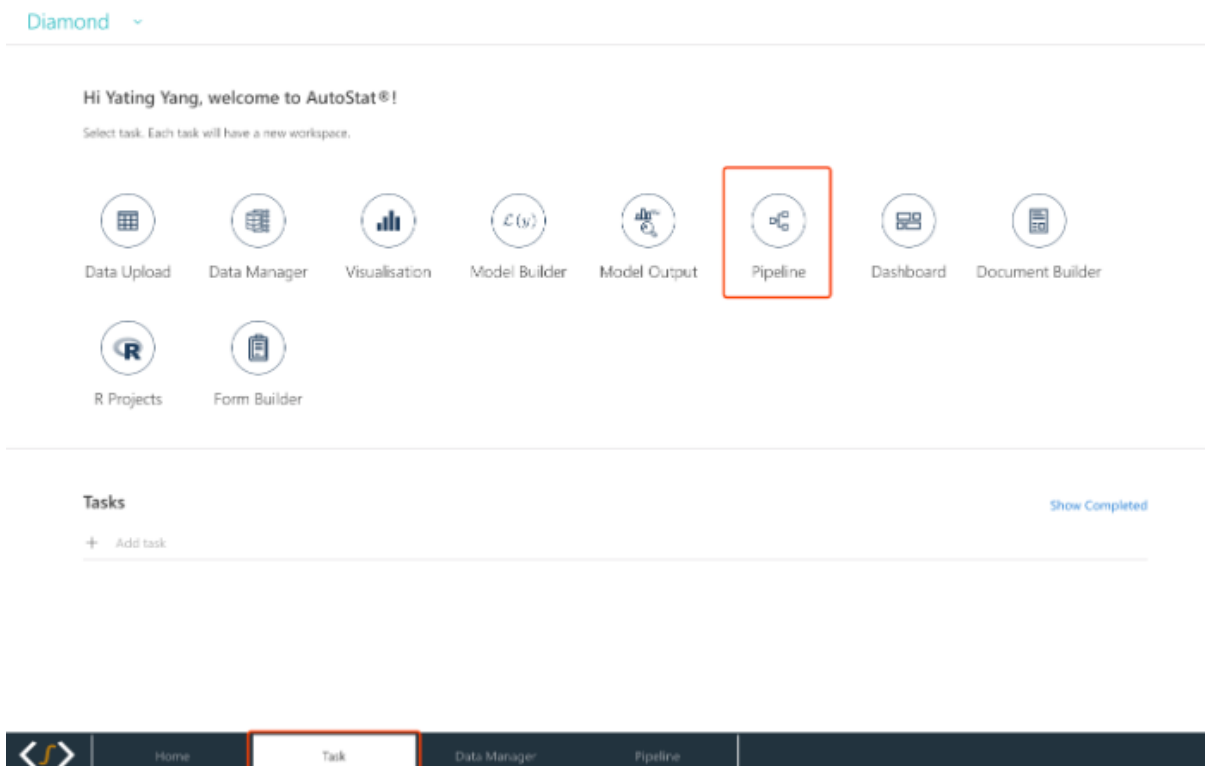
#

Height

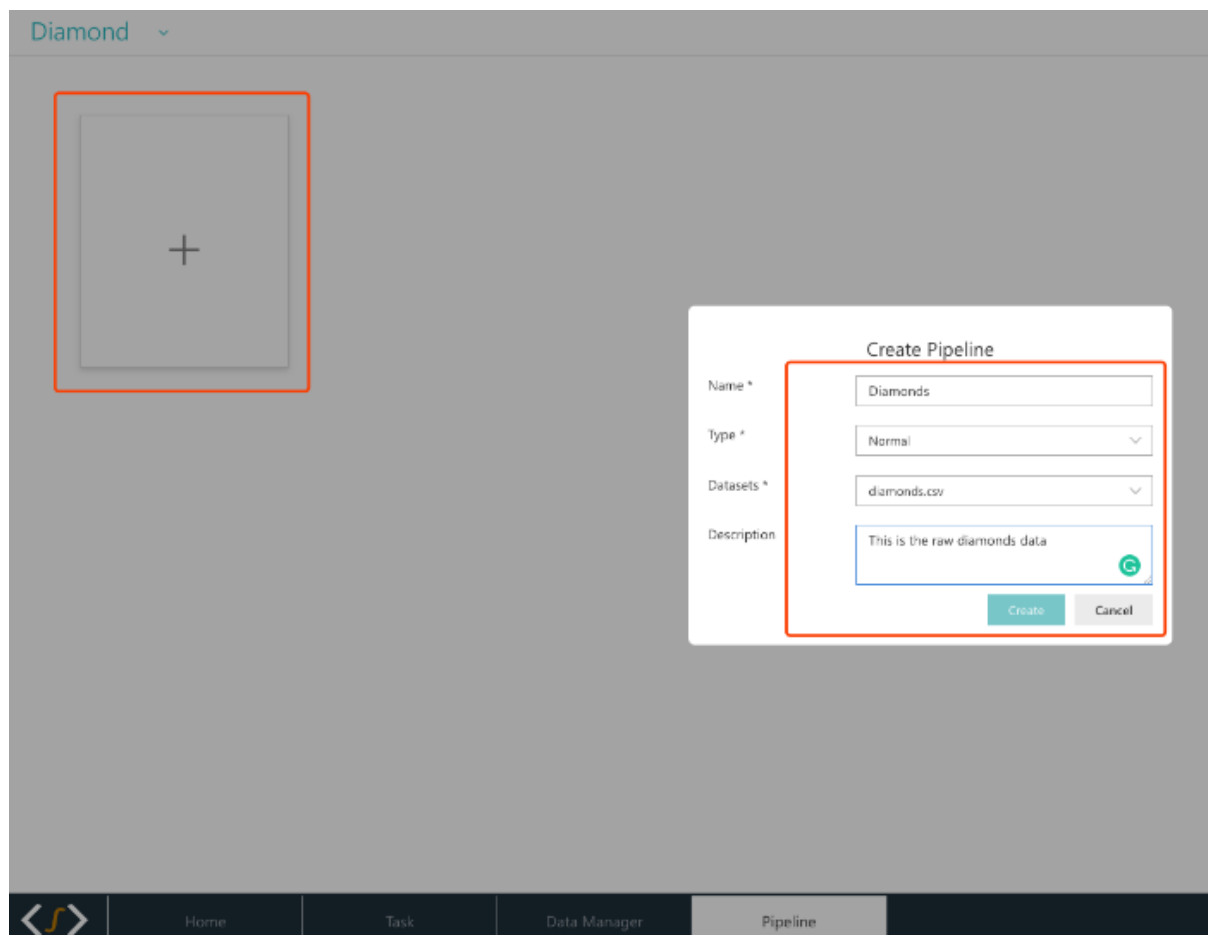
## PIPELINE MODEL

Pipeline is also data processing tool, which have the record for all data processing stepd. It is really useful when you work in a team. Other team member can clear know what you've done, and it is fast to track back wrong data processing and fix the wrong processing steps.

Go back to **Task Module** and select **Pipeline**

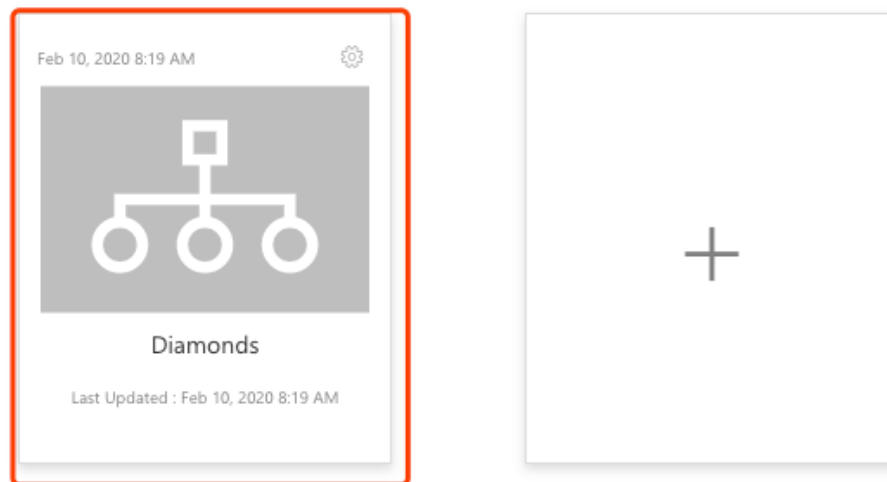


Select a new pipeline by clicking Plus figure and name the new pipeline as Diamonds with introduction.



Then select Diamonds entering into Diamonds pipeline.

Diamond ▼

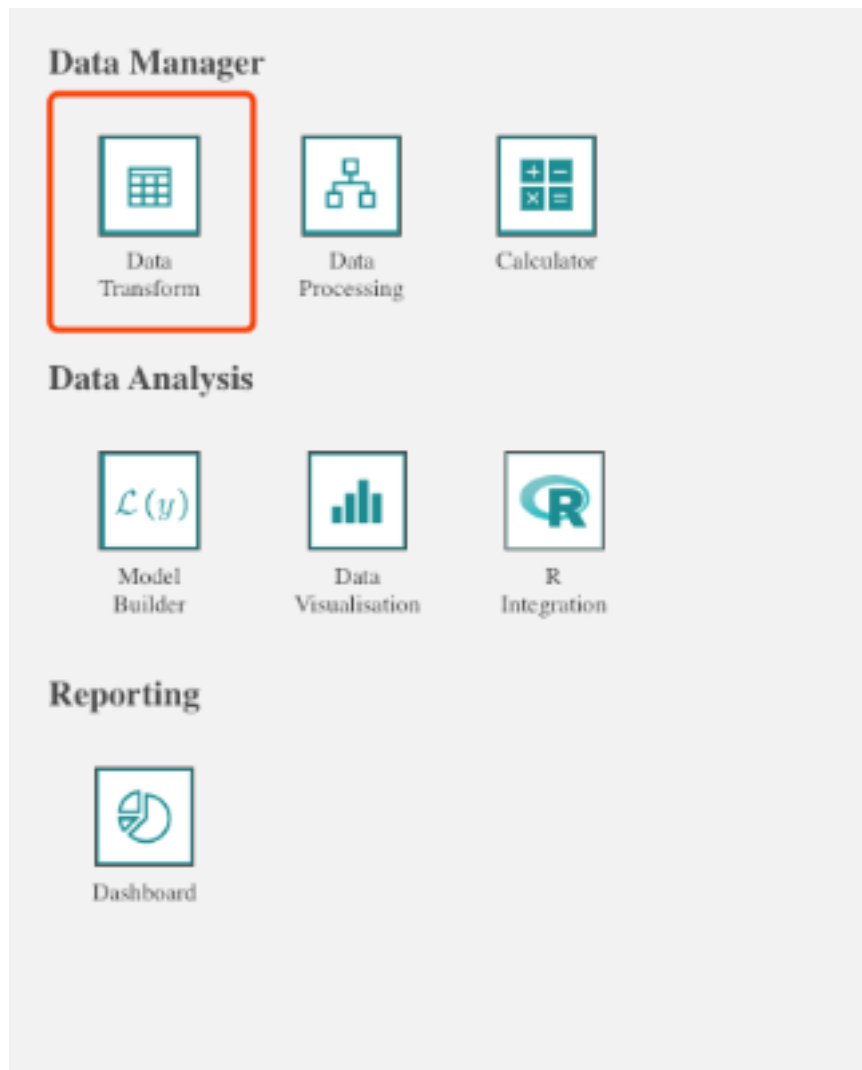


The diamond circle is diamonds raw dataset.

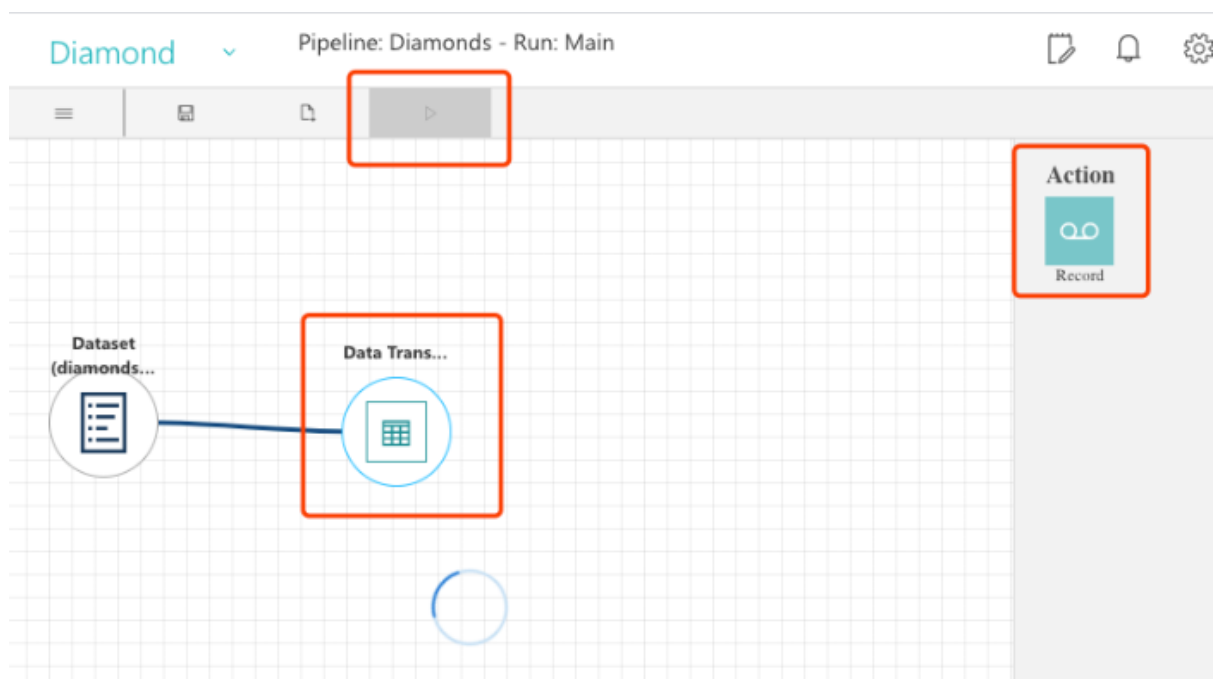
Name	Type
index	number
carat	number
cut	string
color	string
clarity	string
depth	number
table	number
price	number
x	number
y	number
z	number
volume	number

Go to the right panel, which can do data cleaning process.

Select data Transform for transforming into **Data Manager**.



Select the Record button in the Action panel for recording the process.



You are in **Data Manager Module** now. Just change the x, y, z name as we done before.

Data Transform

diamonds.csv

Variables

index

carat

cut

color

clarity

depth

table

price

x

y

z

volumn

index	carat	cut	color	clarity	depth	table
1	0.23	Ideal	E	S12	61.5	55
2	0.21	Premium	E	S11	59.8	61
3	0.23	Good	E	VS1	56.9	65
4	0.29	Premium	I	VS2	62.4	58
5	0.31	Good	J	S12	63.3	58
6	0.24	Very Good				
7	0.24	Very Good				
8	0.26	Very Good				
9	0.22	Fair				
10	0.23	Very Good	H	VS1	59.4	61
11	0.3	Good	J	S11	64	55
12	0.23	Ideal	J	VS1	62.8	56
13	0.22	Premium	F	S11	60.4	61

Rename Variable

Current Name:

x

New Name:

Length

Update

Cancel

Select save and exit, and you will go back to **Pipeline Module**.

Diamond

Pipeline: Diamonds

Exit

Save and Exit

Data Transform

diamonds.csv

Variables

# index

# carat

abc cut

abc color

abc clarity

# depth

# table

# price

# Length

# Width

# Height

index	carat	cut
1	0.23	Idea
2	0.21	Pren
3	0.23	GoD
4	0.29	Pren
5	0.31	GoD
6	0.24	Very
7	0.24	Very
8	0.26	Very
9	0.22	Fair
10	0.23	Very
11	0.3	GoD
12	0.23	Idea
13	0.22	Pren

Select **Data Transform** button you can see what you change in data transform process.

Dataset (diamonds...)

Data Trans...

Dataset (diamonds...)

Action

Record

Recorded Actions

Name	Parameters	Actions
Rename Variable	{ "Old Name": "x", "New Name": "Length" }	
Rename Variable	{ "Old Name": "y", "New Name": "Width" }	
Rename Variable	{ "Old Name": "z", "New Name": "Height" }	

You can try to filter x, y, z is not 0 as we done before as well.

After filterig, select save button.

Data Processing

elements.csv

All, index, carat, cut, color, clarity, depth, table, price, length, width, height

Filtering

Module 1

Length is not

Module 2

Width is not

Module 3

Height is not

Reset Run Load more

carat	cut	color	clarity	depth	table	price	index	Length	Width	Height
0.23	Ideal	E	S12	61.5	55	326	1	3.95	3.90	2.43
0.21	Premium	E	S11	59.8	61	325	2	3.09	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	3	4.05	4.07	2.31
0.29	Premium	I	VS2	67.4	58	334	4	4.2	4.23	2.63
0.31	Good	I	S12	63.1	58	335	5	4.34	4.35	2.75
0.24	Very Good	I	VS2	62.8	57	336	6	3.24	3.28	2.48
0.24	Very Good	I	VS1	62.3	57	336	7	3.05	3.28	2.47
0.26	Very Good	H	S11	61.9	55	337	8	4.07	4.11	2.53
0.22	Fair	S	VS2	65.1	61	337	9	3.07	3.79	2.49
0.22	Very Good	H	VS1	59.4	61	338	10	4	4.05	2.39
0.3	Good	I	S11	64	55	339	11	4.25	4.28	2.73
0.28	Ideal	I	VS1	62.8	56	340	12	3.93	3.9	2.46
0.22	Premium	F	S11	60.4	61	342	13	3.88	3.84	2.61
0.31	Ideal	I	S12	62.2	54	344	14	4.35	4.37	2.71
0.2	Premium	E	S12	60.2	62	345	15	3.79	3.75	2.27

Page 1 of 155

Save the new dataset name  
new\_diamonds.csv

Name the Dataset

Name: new\_diamonds.csv

Save Cancel

Select save and exit

The record looks like figure below:

## Action



Record

## Recorded Actions

Name	Parameters	Actions
Filter	{ "File Name": "new_diamonds.csv", "Fields": ["index", "carat", "cut", "color", "clarity", "depth", "table", "price", "Length", "Wi"]	

Next step is splitting like what we've done before (80% training data and 20% test data, or you can change as you want)

Split Options

☒ Random ☐ Order ☐ Stratify

Random seed: 1

Specify Size of Training Set: 0.80

Split portion: 0.80

Observations in Training Set: 43152

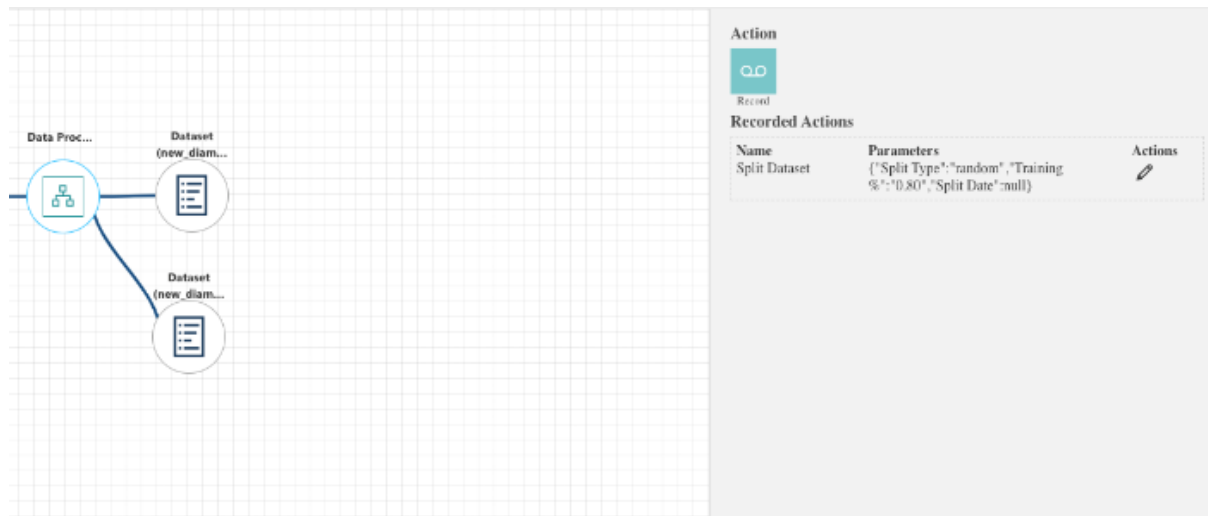
Observations in Testing Set: 10788

OK Cancel

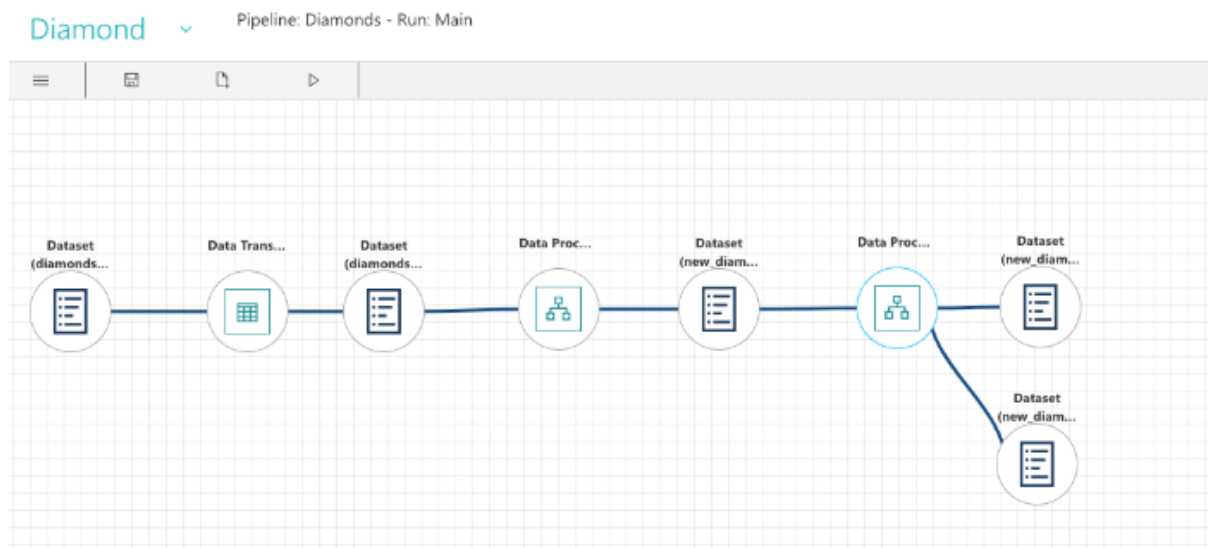
carat	cut	color	clarity	depth	table	price	index	Length
0.23	Ideal	F	SI2	61.5	55	326	1	3.95
0.21	Premium	F	SI1	59.8	61	326	2	3.89
0.23	Good						3	4.05
0.29	Premium						4	4.2
0.31	Good						5	4.34
0.24	Very Good						6	3.94
0.24	Very Good						7	3.95
0.26	Very Good						8	4.07
0.22	Fair						9	3.87
0.23	Very Good						10	4
0.3	Good						11	4.25
0.23	Ideal	J	VS1	62.8	56	340	12	3.93
0.22	Premium	F	SI1	60.4	61	342	13	3.88
0.31	Ideal	J	SI2	62.2	54	344	14	4.35
0.2	Premium	E	SI2	60.2	62	345	15	3.79



Record is shown below:



In this case, the whole data processing in **pipeline** like this:



The data preparation is done.

**Pipeline Module** provides a great record to team cooperation.

## Tutorial 2: Visualization of Data

In this session, the goal is using linear regression to understand the relationship of attributes affect the price of diamond.

### Data Information

This classic dataset contains the prices and other attributes of almost 54,000 diamonds. This datasets is available from <https://www.kaggle.com/shivam2503/diamonds> which contribute by shivamagrwal (2017).

The dataset explores the price of diamonds affected by various of variables.

The dataset include 10 variables for exploring including price variable.

## **The Attributes:**

**Carat:** Carat weight of the diamond.

**Cut:** Cut quality of the diamond. (High - Low quality : Ideal, Premium, Very Good, Good, Fair)

**Color:** Color of the diamond. (Best - Worst: D - J)

**Clarity:** The absence of the inclusions and blemishes. (Best - Worst: IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1)

**Depth:** The height of a Diamond, measured from the Cultet to the table, divided by its average Girdle Diameter.

**Table:** The width of the diamond's table expressed as a percentage of its average diameter.

**Price:** The price of the Diamond

**Length:** Length of the Diamond in mm.

**Width:** Width of the Diamond in mm.

**Height:** Height of the Diamond in mm.

## **VISUALIZATION MODEL**

AutoStat provides an easy drag and drop visualization module.

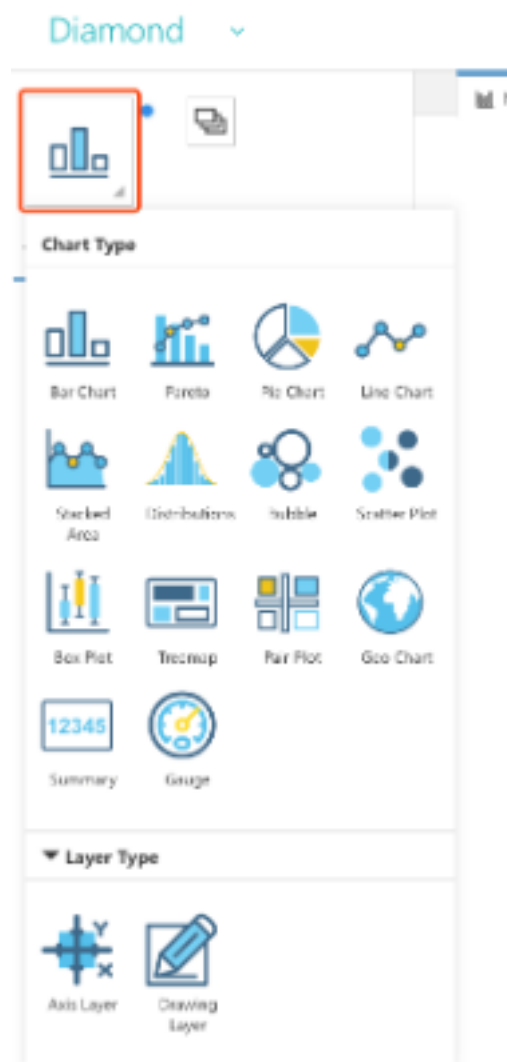
Before starting modelling data, we should visualize the data variables for deeper understanding the relationships between variables.

In this case, we are exploring the related variables of diamonds.

Here are some sample visualization charts to explore:

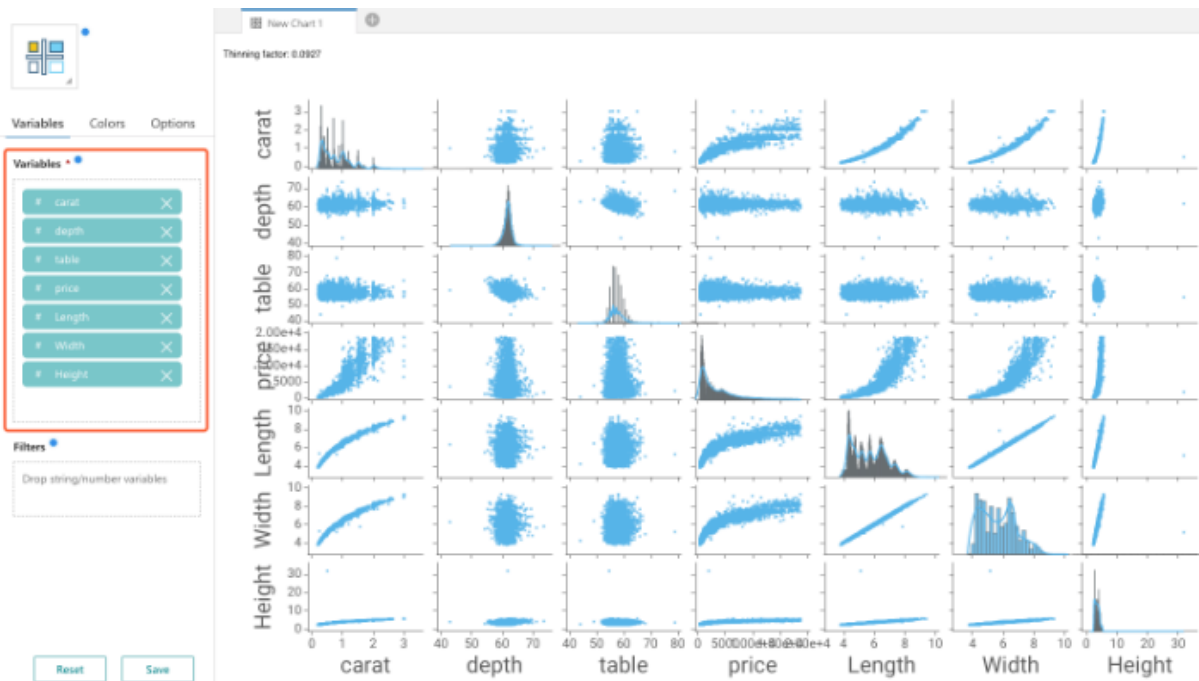
1. Pair Plot
2. Scatter Plot
3. Bar Chart
4. Boxplot

There has a variety of Chart Type, click the chart type for start. For each chart start, just click reset, then the chart will empty.

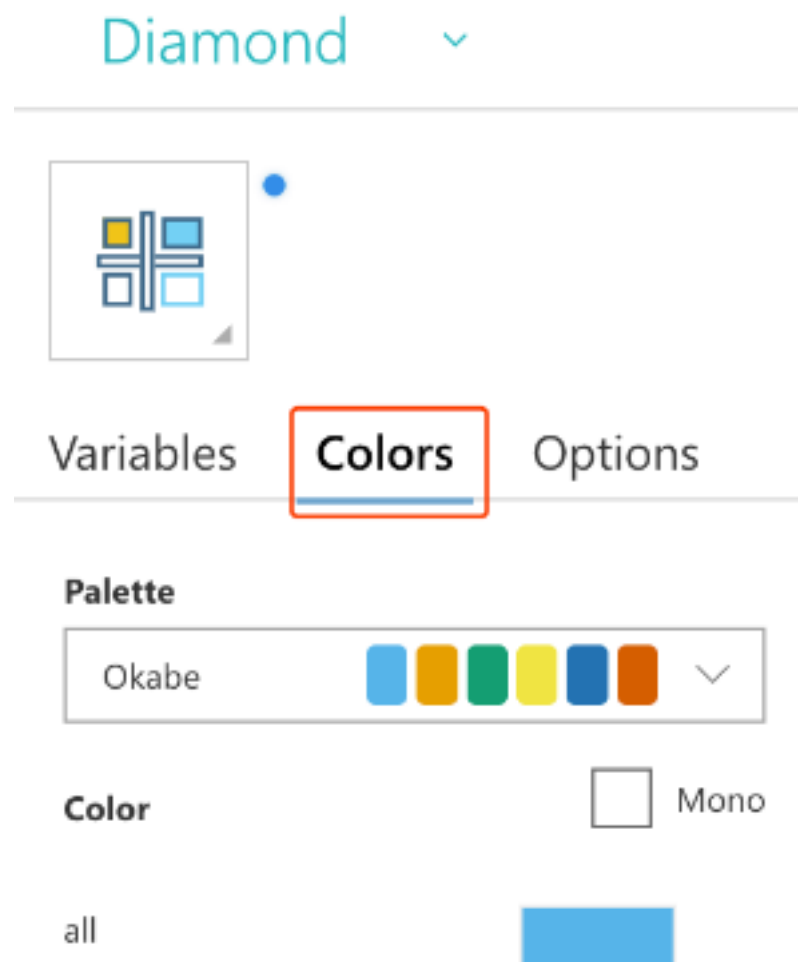


## Pairplot:

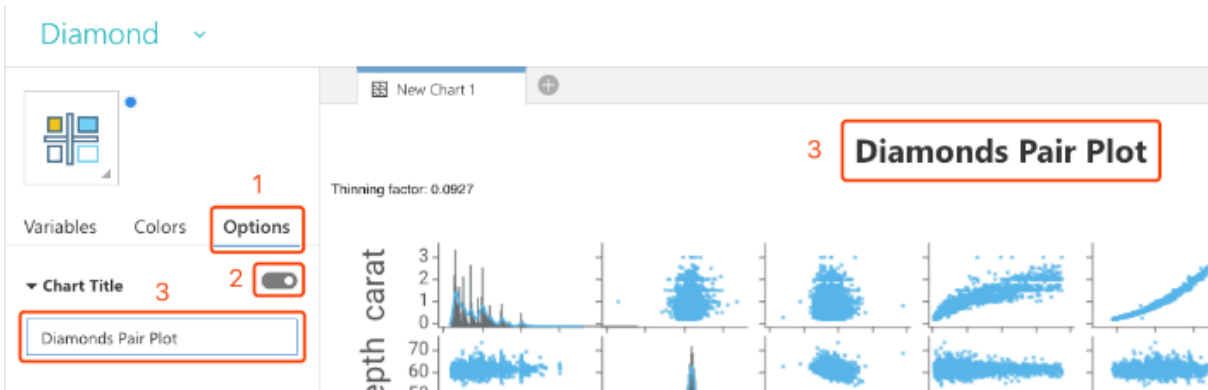
All numeric variables have been choose to pairplot.



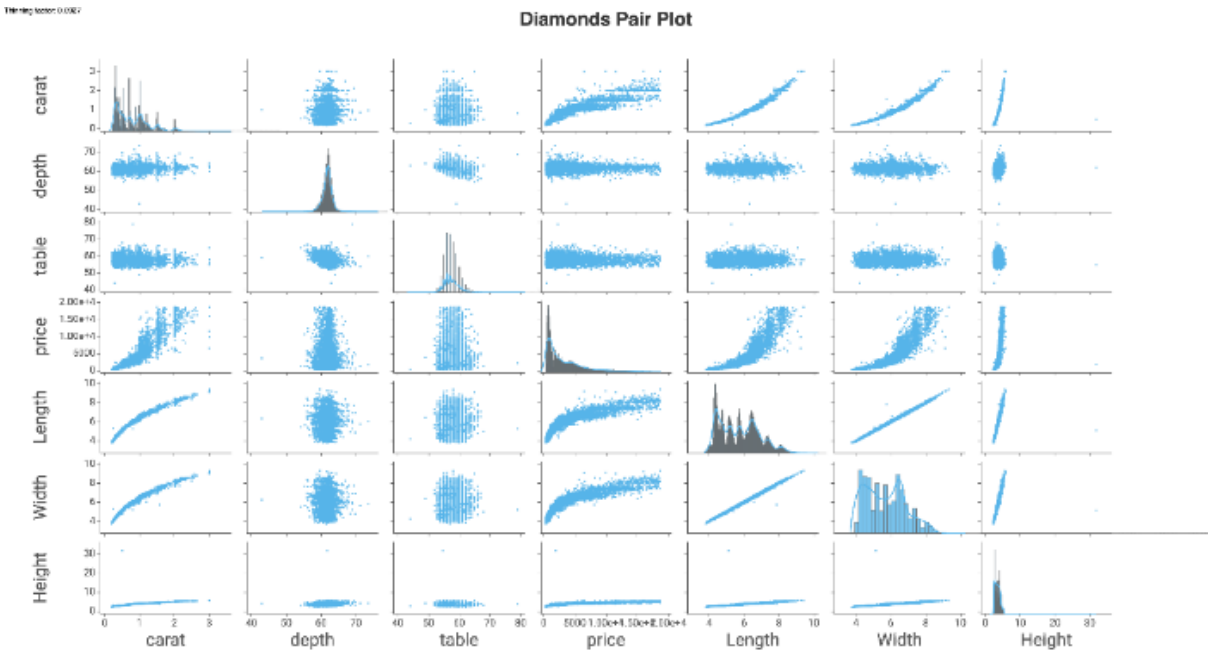
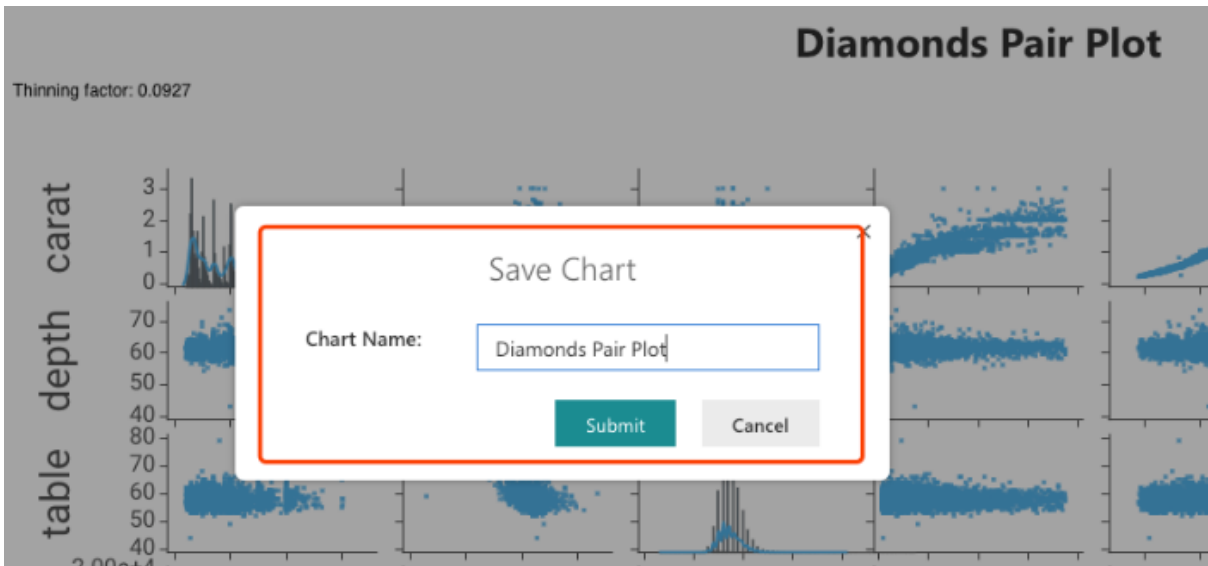
You can choose the color you want in color function. In this case, we use 'Okabe'.



Graph title can be added in Option function.

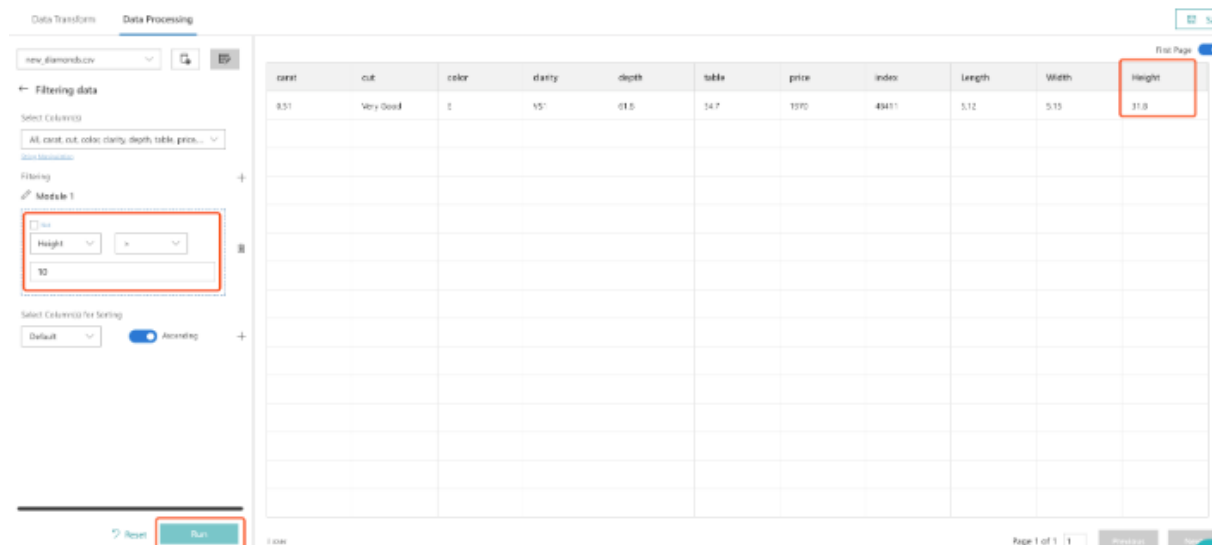


Click save and name this chart as 'Diamond Pairplot'.



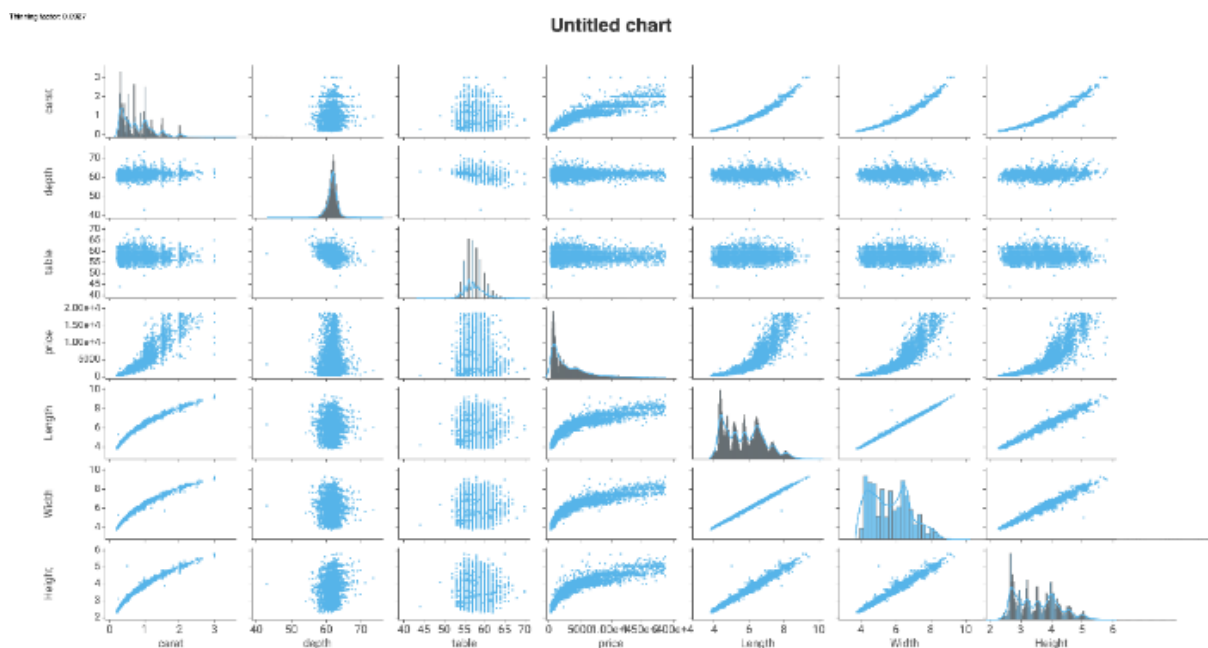
The Height row in the chart has many blank space with few points. These points can be seemed as missing value. The missing value should over 10 based on the Pair Plot.

Come back to Data Manager -> Data Processing -> Filter -> Height > 10 -> run



The table shows that there has one height value over than 10, and the number seems unreasonable. The value is mistake. The missing value only have one, we need to remove this missing value come back to Visualisation Module (If you forget how to filter, please go back to Tutorial 1).

The new Pair Plot shown below, which is much clear than before for comparing the variables relationship.



The charts allows us to see diamonds data distribution of single variables and relationships between two variables, which can be separate with linear relationships or non-linear relationships. It is a great chart type for us to identify the trends for further analysis.

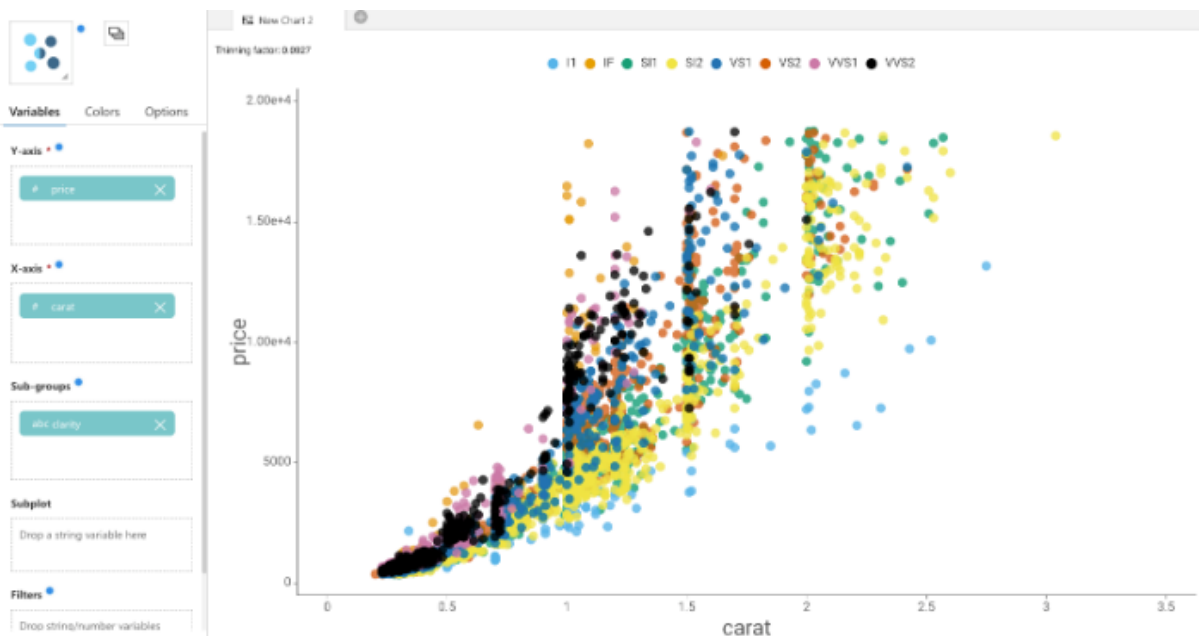
It seems there has non-linear relationships between two variables below (Shape like curve):

- carat and Height
- carat and Width
- carat and Length
- carat and price
- price and Length
- price and Width
- price and Height
- price and carat

Due to the shape, we need to do the spline (The detailed discussion will discuss later) for the relationship mentioned above in the future analysis in Model Builder Module.

## Scatter Plot:

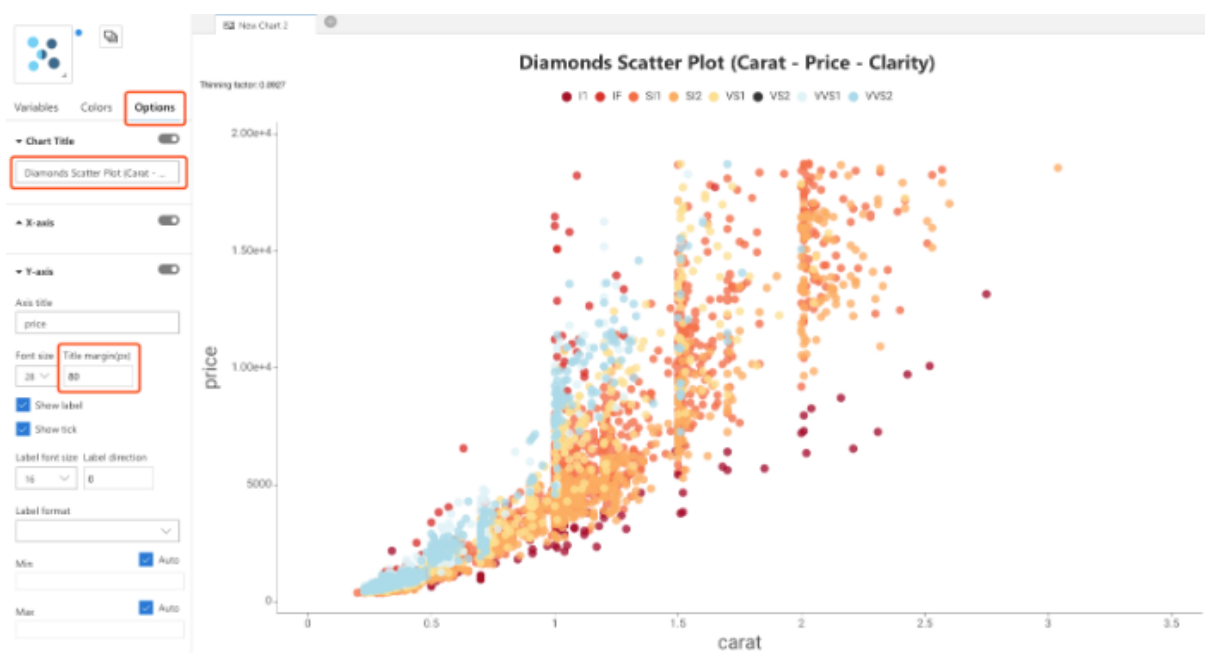
- Y-axis = price
- x-axis = carat
- sub-groups = clarity



The colour looks not good, we just need to choose colours function and select what palette we like. In this case, diverging 1 has been chosen. You can move the mouse to specific colour, and the detail colour information will be showed.



It looks Y-axis title has been covered, and we need to fix it. Options function can adjust the chart. Go to Option functions. Name the chart title and select Y-axis title margin from 40 to 80px, and the chart seems good now.

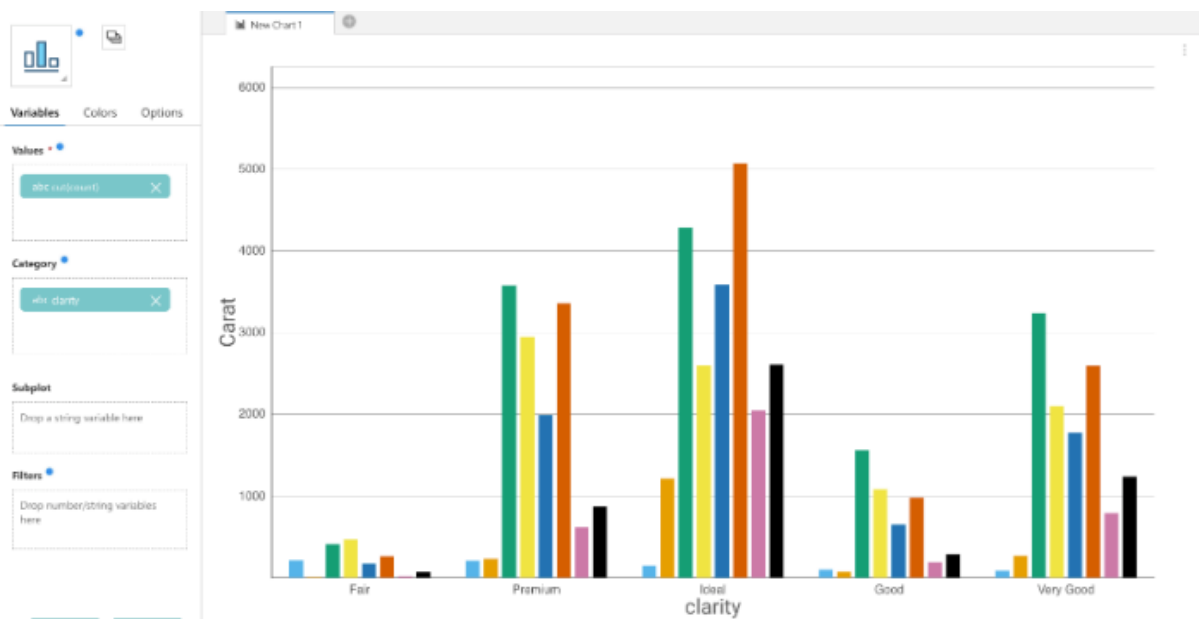


The chart shows carat have strong positive relationship with price. The larger carat the higher price. Clarity has strong positive relationship with price as well. The higher level of clarity the higher the price. The higher level clarity are possible to smaller and less than the lower level clarity.

## Bar Chart:

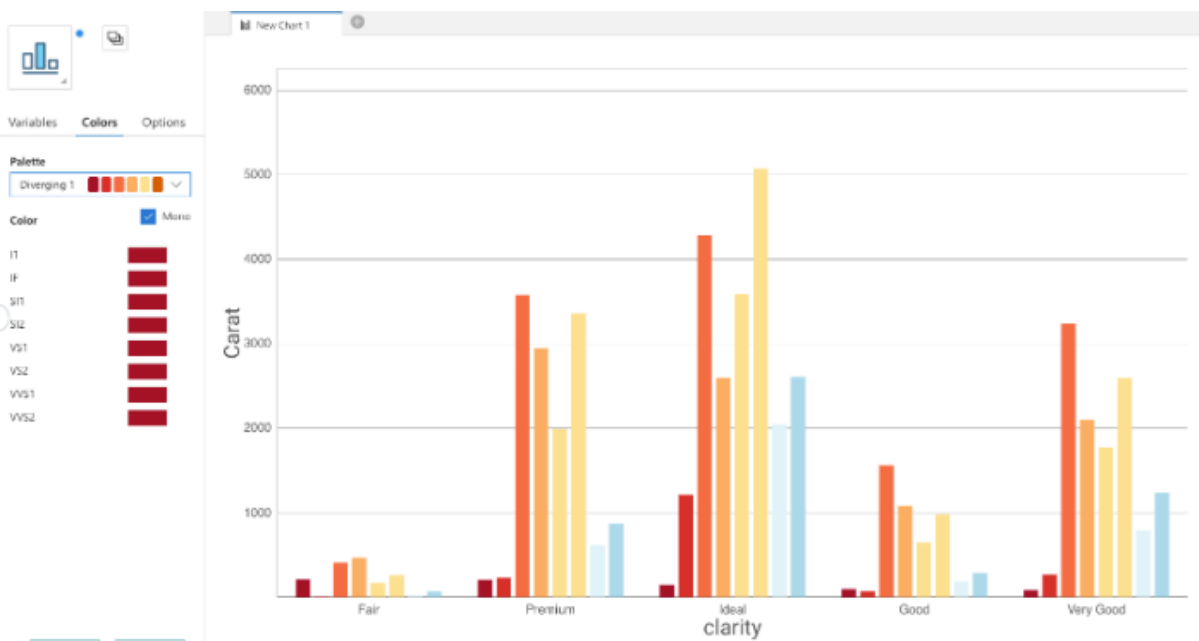
- Values = cut (count)
- Category = clarity



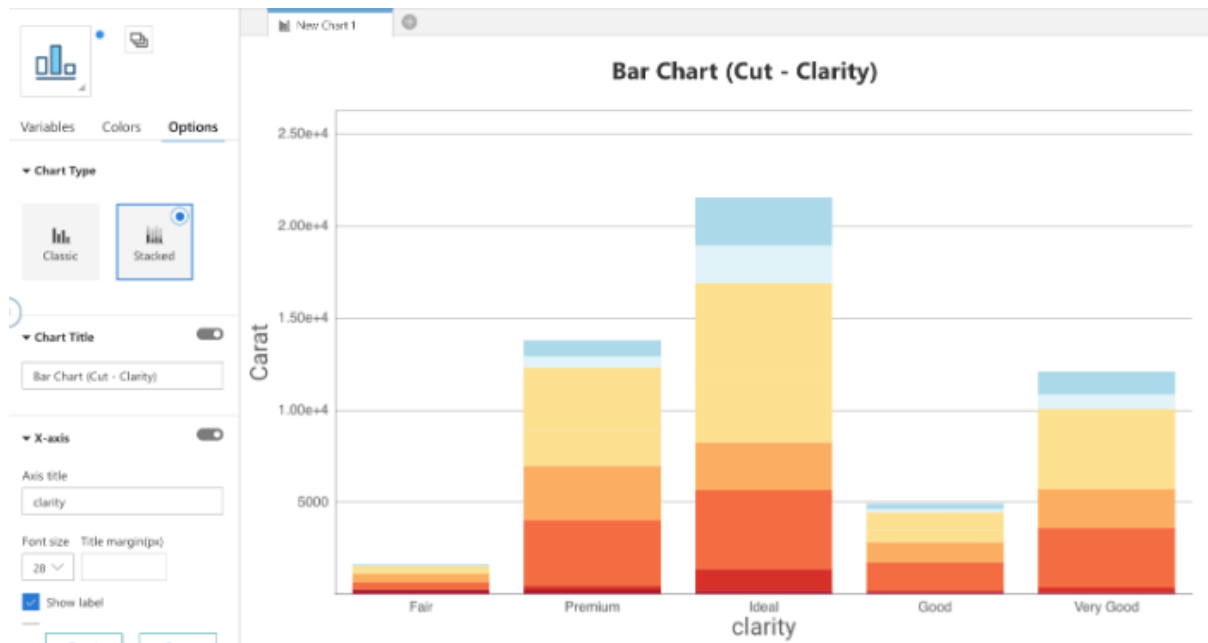


It seems the chart is not clear with the relationship, then we need to fix it.

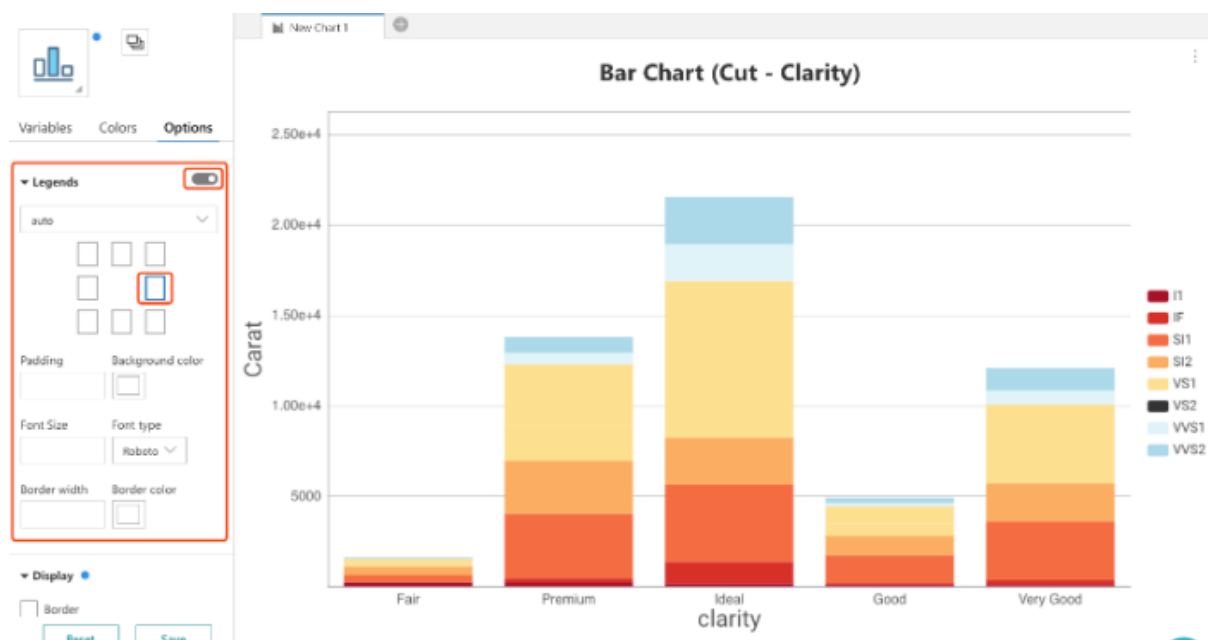
Go to Colors function and select the colour you want. In this case, we select 'Diverging 1' which is my favored color.



Go to Options Select the chart type from classic to stacked, and name the chart title as before. The chart seems better. But we still miss the legend. We need to fix it.



Scroll down the Option function, there has legends. Open legends and select which position you want. In this case, the legend locates in the right. It seems good, and we need to save it as before.



The chart shows that the number of is highly relative to the cut level. The number of high quality cut level is much more than the number of low quality cut level. There also has a strong relationship between clarity and cut. The higher level of clarity, the higher quality of cut level.

## Boxplot:

- #The bottom line indicates the min value.
- #The upper line indicates the max value.
- #The middle line of the box is the 50% percentile.
- #The side lines of the box are the 25 and 75 percentiles respectively.
- Values = price

- Category = clarity

In this case, the color is 'Gradient 1'. Name the title in the options function, and adjust the Y-axis title margin.



The chart shows that VS1 and VS2 affect the price with high price margin, and IF and VVS1 affect the price with low price margin. Nearly 75% of I1 and SI1 price within 5000 and nearly 75% of IF and VVS1 price within 2500. The min value of different level of clarity seems similar.

## Tutorial 3: Regression Analysis

In this session, a series of linear regressions need to be conducted.

- Standard Frequentist
- Variable selection with Frequentist (Lasso) and Bayesian (G-prior spike-slab) approaches

## Data Information

This classic dataset contains the prices and other attributes of almost 54,000 diamonds. This datasets is available from <https://www.kaggle.com/shivam2503/diamonds> which contribute by shivamagrawal (2017).

The dataset explores the price of diamonds affected by various of variables. The dataset include 10 variables for exploring including price variable.

## MODEL BUILDER MODULE

For standard linear regression, All variables will be included in the model.

The model can be written as:

$$\text{price} \approx \beta_0 + \beta_1 * \text{carat} + \beta_2 * \text{cut} + \beta_3 * \text{color} + \beta_4 * \text{clarity} + \beta_5 * \text{depth} + \beta_6 * \text{table} + \beta_7 * \text{Length} + \beta_8 * \text{Width} + \beta_9 * \text{Height}$$

$$\text{price} \approx \beta_0 + \beta_1 * \text{carat} + \beta_2 * \text{cut} + \beta_3 * \text{color} + \beta_4 * \text{clarity} + \beta_5 * \text{depth} + \beta_6 * \text{table} + \beta_7 * \text{Length} + \beta_8 * \text{Width} + \beta_9 * \text{Height}$$

and is usually written in short form:

$$Y = X\beta + \epsilon \quad Y = X\beta + \epsilon, \text{ where } \epsilon \sim N(0, \sigma^2) \quad \epsilon \sim N(0, \sigma^2)$$

Linear regression is aiming to estimate values of  $\hat{\beta}$  such that the residual sum of squares (RSS) is minimized.

The process of setting of the model involves choice via radio button and dropdown menus, along with drop and drag features.

This button will take you straight to the Model we intend to build:

### Standard linear regression

Select Datasets new\_diamond.csv and put it into training data. Then choose Linear Regression.

The screenshot shows the AutoStat interface for building a model. The dataset 'Diamond' is selected. In the 'Uploaded Datasets' section, 'new\_diamonds.csv' is highlighted as the 'Training Data'. The 'Test / Validation Data (Optional)' section is empty, with a 'Data Filter' link below it. In the 'Select Model' section, the 'Regression' dropdown is open, and 'Linear' is selected. The 'Variables' list on the right shows the following variables: # carat, cut, color, clarity, # depth, # table, # price, # Length, # Width, # Height, and # index. The # symbol indicates that the variable is specified as continuous. The text 'No test options' is displayed at the bottom right.

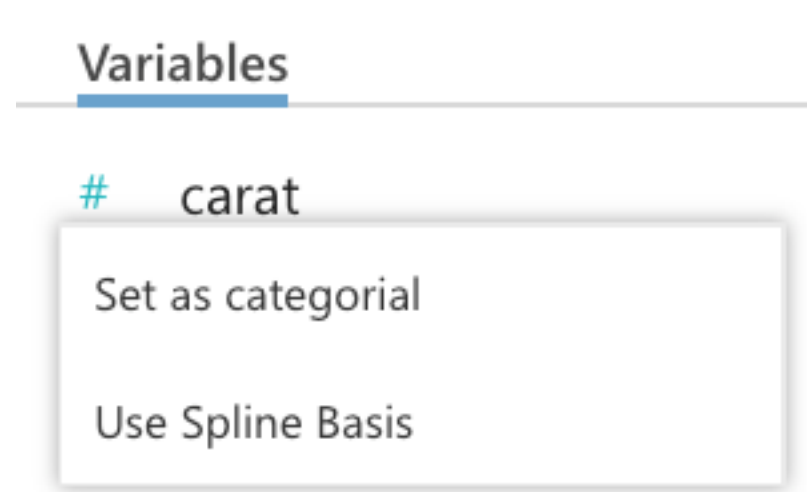
Once data is selected, the available variables are shown.

The following information is conveyed in the variables list:

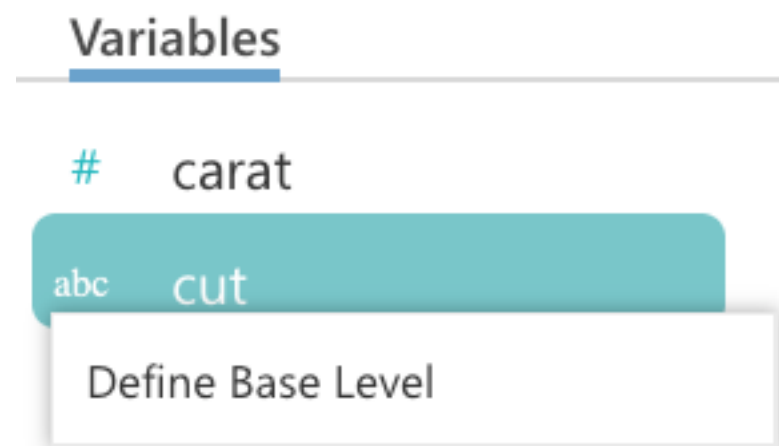
1. The # symbol indicates that the variable is specified as continuous

2. The abc symbol indicates a variable is categorical
3. The spl indicate that the variable will be modelled as a spline (more on this in a later tutorial)
4. Variables which are in black text indicate there are no missing values. If the variable name is in red text, missing values are present.

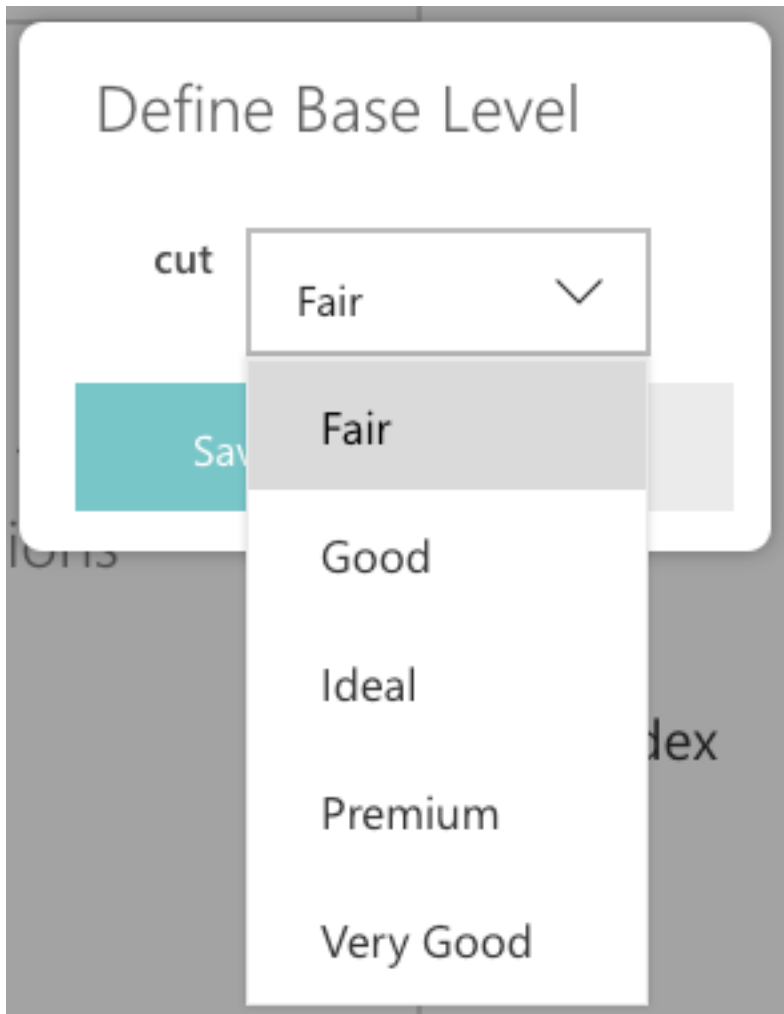
Sometimes, you may wish to **temporarily change** the properties of the variable, such as continuous to categorical. You can set this temporary change in model builder by right clicking on the variable. To make a **permanent change**, use the Data Manager. For a continuous variable you have the options below.



For categorical variables, you can change the base level of the variable. This will then be the level at which all departure comparisons can be made.



For example, here we may want to change from Fair (the default base level due to alphabetical ordering) to Ideal. This is an important consideration, as many fields of research may have a default base level, such as Gender, or all treatments may be compared to a control, or standard current best treatment.



Now you need to define your model. To do this, drag your Forecast (outcome) variable into the appropriate space (see below). In our case, this variable is price. Then drag the explanatory variables into the lower space. In our case this is all the other variables in the dataset.

Define Variables

Forecast Variable(s)

price

Explanatory Variable(s)

carat

cut

color

clarity

depth

table

Length

Width

Height

If you are using all (or most) of the other variables, use this shortcut. If you want to remove some of the variables, simply click to make them disappear. Note, you should have you Forecast variable defined before you do this, or it will appear in your explanatory variables.

Now you can choose your favourite paradigm (Frequentist or Bayesian), and options therein.

## Parameter Settings ●

### Linear

Frequentist

▼

No Regularisation

▼

▼ Standard Errors

☒ Standard

☐ White's Hetero Consistent Standard Errors

Firstly, we choose Frequentist with No Regularisation and run the model by clicking Analyse button.

Training Data ●

new\_diamonds.csv

Test / Validation Data (Optional) ●

Drag a Dataset

Data Filter

No test options

Analyse ●

Variables

# carat

# cut

# color

# clarity

# depth

# table

# price

# Length

# Width

# Height

# index

Define Variables ●

Forecast Variable(s)

price

Explanatory Variable(s) ●

carat

cut

color

clarity

depth

table

Length

Width

Height

Parameter Settings ●

Linear

Frequentist

▼

No Regularisation

▼

▼ Standard Errors

☒ Standard

☐ White's Hetero Consistent Standard Errors

Once the Analysis is implemented, the **Output Module** will be shown with three tabs:

- Standard output: Tables of the usual reported metrics



- Diagnostics: Graphs of model residuals
- Variable impact charts: A quick view of the response for each coefficient

All the elements in the three tabs can be sent to the **Document Builder** which is ready for importing into reports or papers. Clicking hamburger dots located in the top right corner of each elements. The three documents have been prepared for further steps.



Linear Regression Model Output

Forecast variable: price

Number of observations: 53920

Number of regressors: 24

R-sq: 0.9201

adj-R-sq: 0.9201

F-statistic: 26987.7819 (0)

Log-likelihood: -455421.8545

AIC: 910893.7089

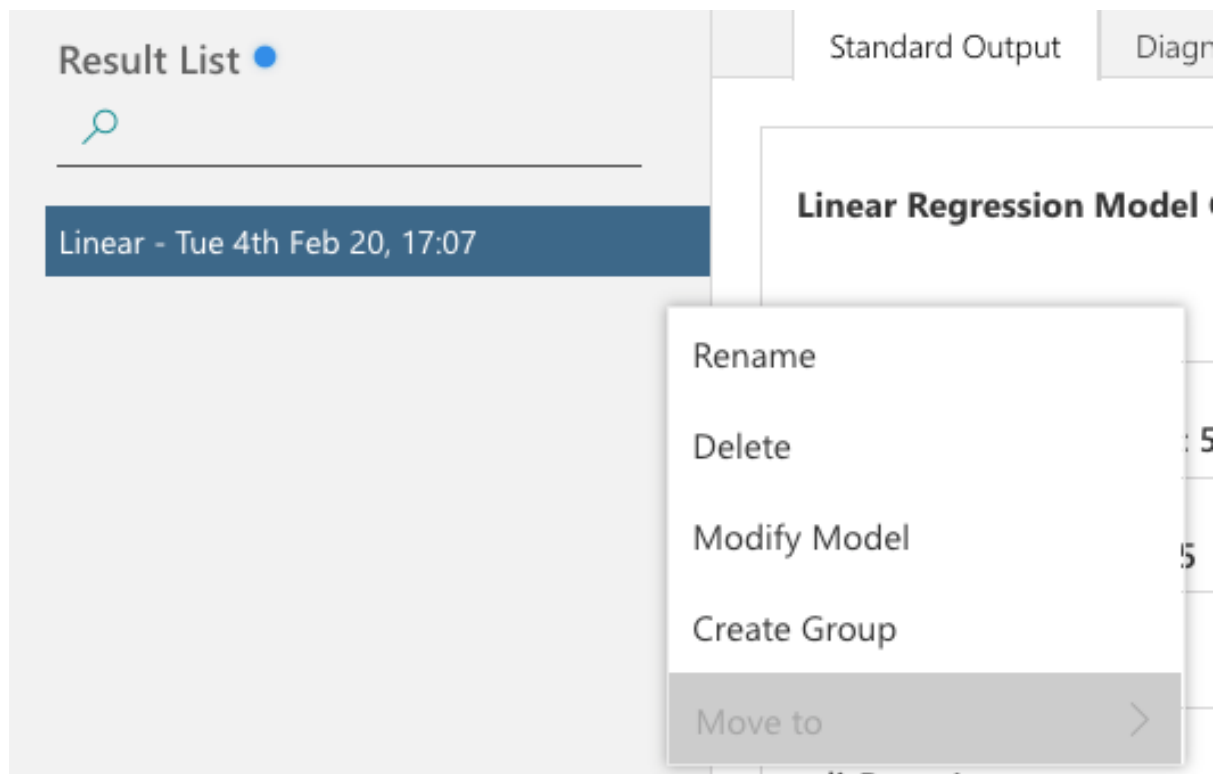
BIC: 911116.0904

Residual Standard Error: 1127.2486

Linear Regression Model Output

Variable	Coefficient	Standard Error	t-value	p-value	CI 2.5%	CI 97.5%
CONSTANT	2711.9832	413.7977	6.5539	5.6963e-11	1900.9363	3523.0301

If you want to keep your model you may like rename your model in more appropriate words, or you can delete as you want. You can even modify it by clicking **Modify Model**.



Go to **Standard Output** and have a look at the **Linear Regression Model Output**

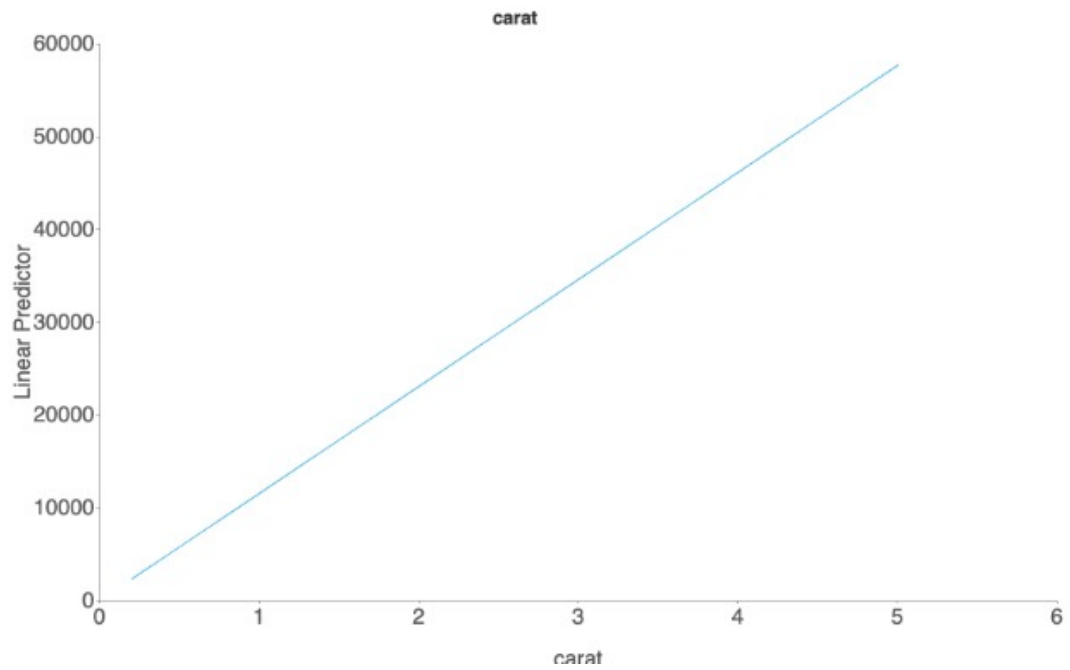
The summary table of coefficients indicates that there is a relationship between some of the variables. People often make this assessment based on the *p-value* of the coefficient, which is considered to be an indicator of whether the value is significantly different to zero, and is based on the point estimate of the coefficient, the standard error and Normal theory. If the *p-value* is significant, the provided 95% Confidence (*p-value* < 0.05) interval will also not include zero.

The *p-value* is the centre of the current debate on reproducibility in science. In general, we would say it is reasonable to look at the *p-value*, but not use it as an absolute bright-line decision making value, but as part of the body of evidence. Later in the tutorial we will look at more robust methods for variable assessment.

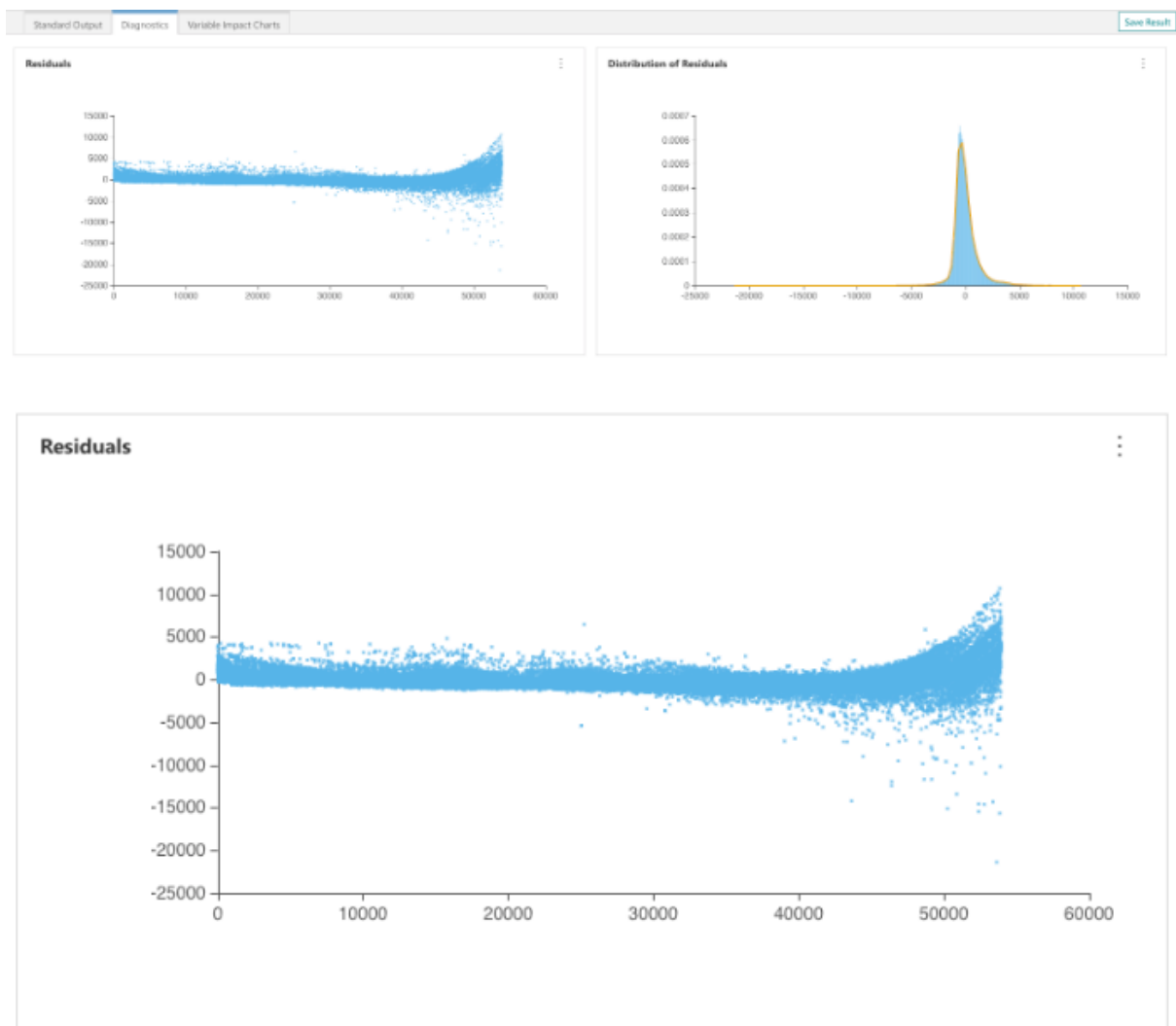
Linear Regression Model Output

Variable	Coefficient	Standard Error	t-value	p-value	CI 2.5%	CI 97.5%
CONSTANT	2711.9832	413.7977	6.5539	5.6363e-11	1903.8363	3523.0301
carat	11525.6708	51.6301	223.2353	0	11424.4753	11626.8663
cut_Good	574.3906	33.5295	17.1324	1.2694e-45	308.6784	840.1027
cut_Ideal	824.8785	33.3888	24.7423	2.1173e-134	759.3342	890.2227
cut_Premium	753.0046	32.1644	23.4111	1.3208e-129	689.9615	816.0464
cut_Very Good	717.3414	32.1782	22.2938	1.3619e-109	654.2719	780.4109
color_E	-268.9032	17.8490	-11.7039	1.3292e-31	-243.8873	-173.9190
color_F	-267.4026	18.0519	-14.8130	1.5117e-49	-302.7844	-232.0207
color_G	-477.3752	17.6757	-26.9960	1.9028e-159	-511.8197	-442.5306
color_H	-979.7585	18.7816	-52.1381	0	-1016.5902	-942.9267
color_I	-1479.2534	21.1136	-69.5354	0	-1511.6362	-1428.8707
color_J	-2376.0662	26.0711	-91.1879	0	-2427.1657	-2324.9666
clarity_IF	5340.2771	59.9678	104.7774	0	5240.3797	5440.1744
clarity_S1	3677.7624	43.6168	84.3198	0	3592.2731	3763.2517
clarity_S2	2716.6898	43.8015	62.0228	0	2630.8385	2802.5410
clarity_VS1	4587.0437	44.5193	103.0349	0	4499.7854	4674.3020
clarity_VS2	4276.3465	43.8315	97.5533	0	4190.4364	4362.2566

Looking at the table, we note that the coefficient for Ideal is 824.8785, which indicates that the average price of this diamond is higher than the base level, which is Fair. We note that the coefficient for carat is big (positive) in magnitude, but magnitude of coefficients is relative to the measurement scale, which is around 0.2 - 5.01, so how do we interpret this coefficient? Increasing by 1 carat unit would increase price by 11525.6708. That seems large. This is difficult to think through at this stage of the analysis, so going to the variable impact charts will give you a quick view of the change in price attributed to carat over the range of the data. We see that the change in price being attributed to a change from 1 units of carat to 5 units of carat (min to max) is around 50000.



Go to **Diagnostics** and have a look at the **Residuals**



The residuals pretty symmetrically distributed, tending to cluster towards the middle of the plot. They're clustered around the lower single digits of the y-axis (e.g., 1000 or 5000, not 20000). When price large enough, the residuals is much larger. It shows the price have larger fluctuations when it is large enough.

## Variable selection

It has long been recognised that including too many variables in the model can have a detrimental effect on the effectiveness of the procedure. There is a tendency for models with too many variables, which may cause overfitting. Overfitting may cause new data not fitting well.

In the past, techniques such as stepwise where terms were either added or subtracted (or stepwise in both directions) one at a time and the resulting model tested for improvement (if adding terms) or worse fit (if deleting terms). However, these techniques are now widely discouraged due to their frequent misuse, and have been replaced by more efficient algorithms. In addition, the increase in large data means that the subtract version is unavailable, as datasets may have more regressors than observations.

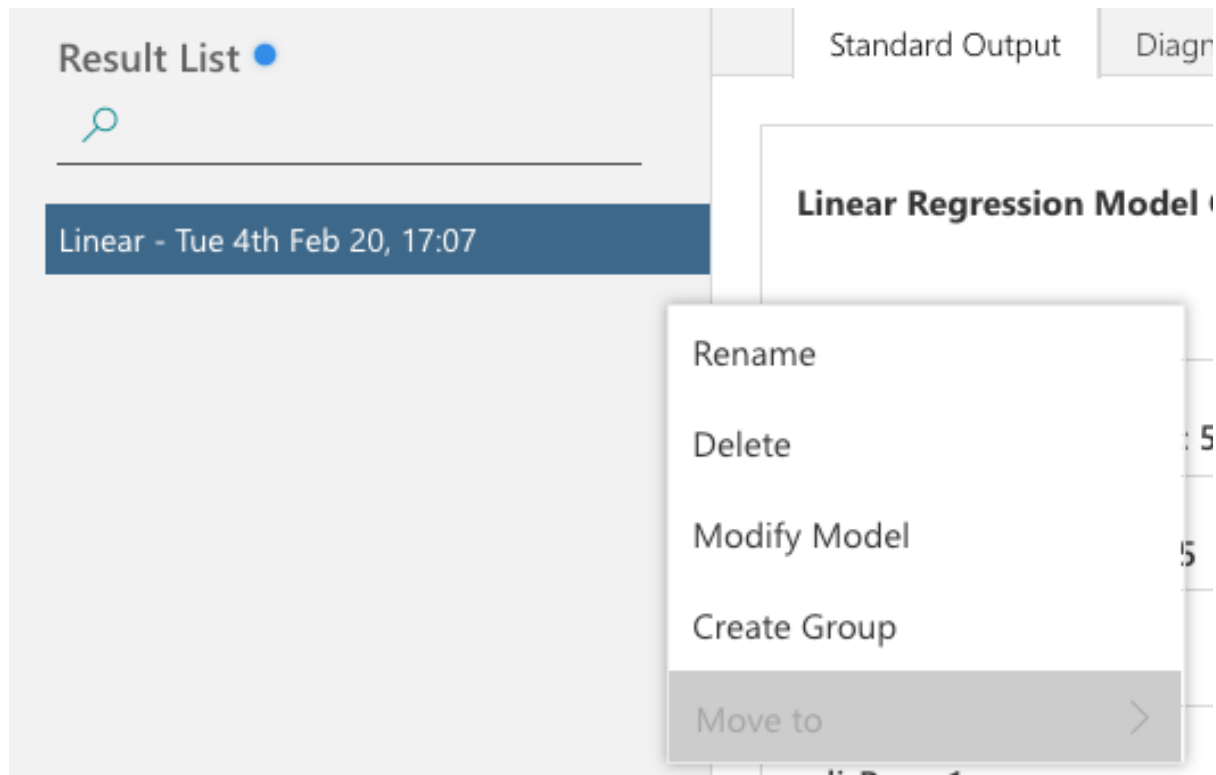
## Lasso regression

The use of standard regression which minimizes the square of the errors, *i.e.*  $SSR = \sum_{i=1}^n (y_i - \beta x_i)^2$  results in a fit to the modelled data set that may not be readily transportable to a different dataset.

One option for reducing the influence or number of variables in the model is known as *shrinkage*, and these methods place a penalty in the minimization process. The **Lasso** estimate is defined as

$$\hat{\beta}_{\text{lasso}} = \arg\min_{\beta} \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j|$$
, subject to a constraint  $\sum_{j=1}^p |\beta_j| \leq t$

This model fits all the required explanatory variables and *shrinks* them towards zero (0). This is completed using **Model Builder**. You can also change to the Lasso model from your results section by *modifying* your current model by right clicking on the 3 vertical dots. You will see these options.



Using the default parameters, the Lasso model reduces the original 24 variables to 23. The y variable will not be used. Lasso automatically remove the unimportant variables for you.

## Linear Regression Model Output

Forecast variable: price

---

Number of observations: 53920

---

Number of regressors: 24

---

Number of non-zero regressors: 23

---

lambda: 0.0009

---

alpha: 1

---

The non-zero coefficients are displayed in the table, you can display all coefficients by choosing this option on the right.

Linear Regression Model Output		No-zero Coefficients	⋮
Variable	Coefficient	Show All Coefficients	
CONSTANT	2923.957	No-zero Coefficients	
carat	11071.5703		
cut_Good	421.7940		
cut_Ideal	677.5437		
cut_Premium	601.6322		
cut_Very Good	575.0714		
color_E	-141.6994		
color_F	-204.7740		
color_G	-407.9716		
color_H	-905.6171		
color_I	-1381.1692		
color_J	-2275.2923		
clarity_IF	4678.6343		
clarity_SI1	3028.1242		
clarity_SI2	2074.5945		
clarity_VS1	3931.0381		
clarity_VS2	3628.4700		
clarity_VVS1	4352.2592		

Lets save this table to documents, and then create a document with both tables of estimated coefficients so we can better compare them.

## Horseshoe

## Linear Regression Model Output

Linear Regression Model Output

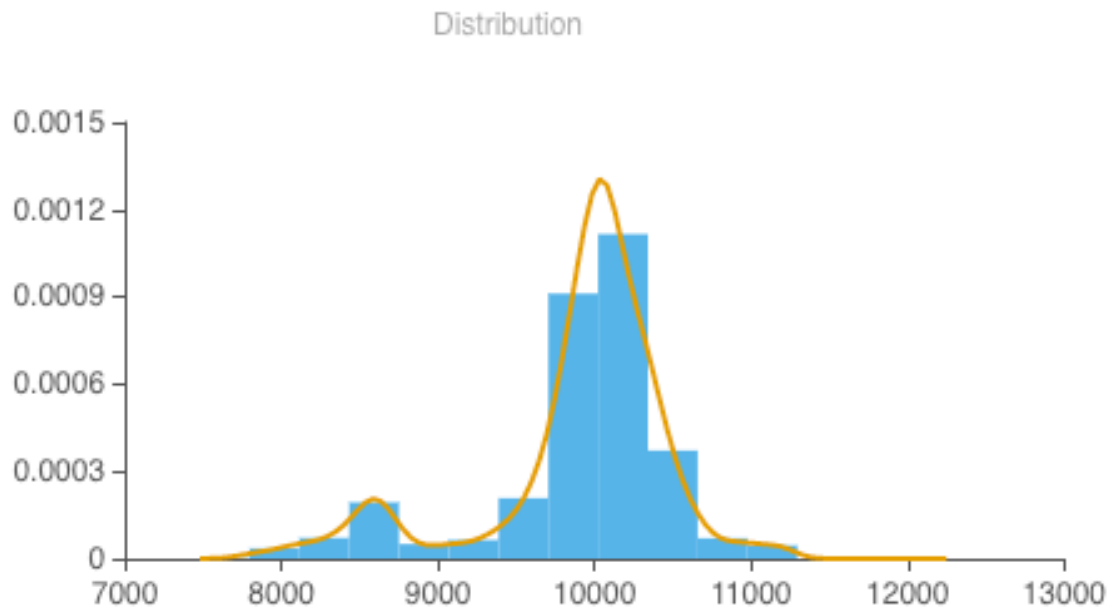
Variable	Mean	SD	HPD 2.5%	HPD 97.5%	IF
CONSTANT	61.9433	904.6528	-0.7855	0.9849	47.9004
carat	9903.8304	594.4320	8263.2984	10720.9271	65.4588
cut_Good	-8.6639	52.6541	-38.0143	2.9607	35.4453
cut_Ideal	122.7771	150.4456	-0.9056	351.1081	137.1999
cut_Premium	7.5329	54.0415	-2.2141	5.9527	24.3748
cut_Very Good	7.7114	54.0763	-3.2380	8.0110	28.3933
color_E	4.9405	33.3492	-1.0826	1.4560	42.1144
color_F	0.7129	11.5385	-0.8536	1.0523	16.2404
color_G	-11.0815	48.0136	-135.4546	10.0131	33.6248
color_H	-472.9222	297.2405	-728.6237	0.3456	169.2577
color_I	-931.6581	352.8020	-1174.1819	0.4534	129.9565
color_J	-1846.7986	372.4482	-2163.8250	-1335.0819	88.4470
clarity_IF	1002.6743	1139.8401	-1.3597	2834.6535	222.1912
clarity_SI1	-102.4632	782.1557	-815.1946	1165.7711	226.4646
clarity_SI2	-920.7054	837.9485	-1708.3280	1.9944	352.1296
clarity_VS1	661.8623	881.8783	-1.0991	2051.3974	280.3483
clarity_VS2	496.0984	811.4376	-132.8700	1816.2531	255.8077
clarity_VVS1	1039.7213	947.1592	-0.7181	2508.2245	244.2470
clarity_VVS2	1019.6502	945.6086	-1.1037	2466.6757	274.0431

IF is referred as Inefficiency Factor, which is a ratio of some measure of performance to an unexpected value. The output shows the cut\_Premium perform the best while the clarity\_SI2 perform the worst. The standard of IF is that the higher the number, the worse the variable perform.

Go to **MCMC Charts**

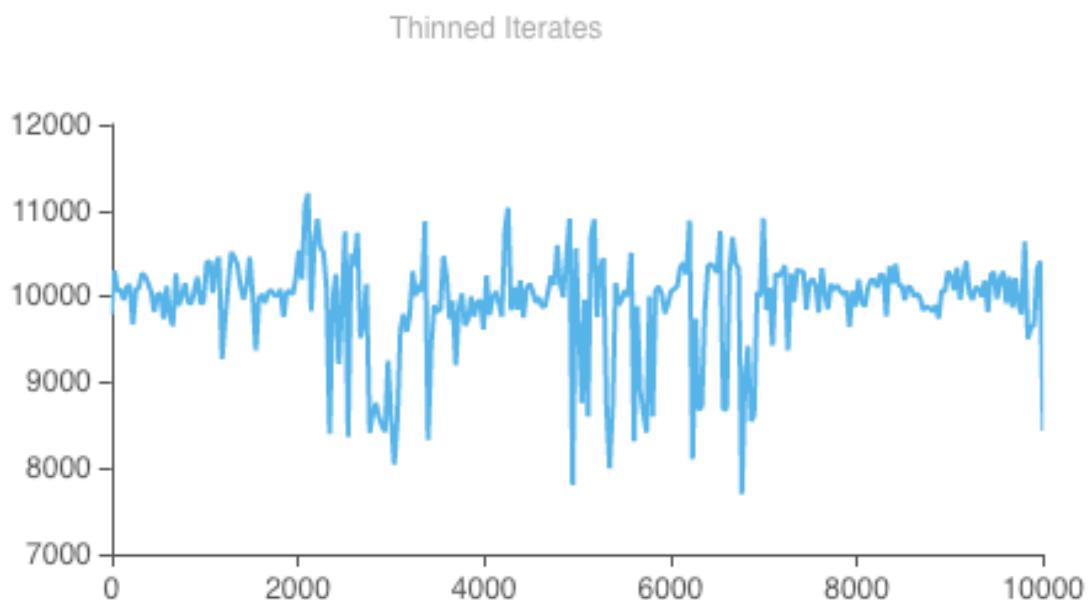
**Distribution**





The posterior distribution of the coefficients should be symmetrically distributed around the posterior mean. However, from the density histograms of our coefficients (see carat), we see a multimodal distribution. This indicates that the simulated draws from the posterior have not found a stable distribution, and this is further evidenced in the trace plot below.

#### Thinned Iterates (Trace Plot)

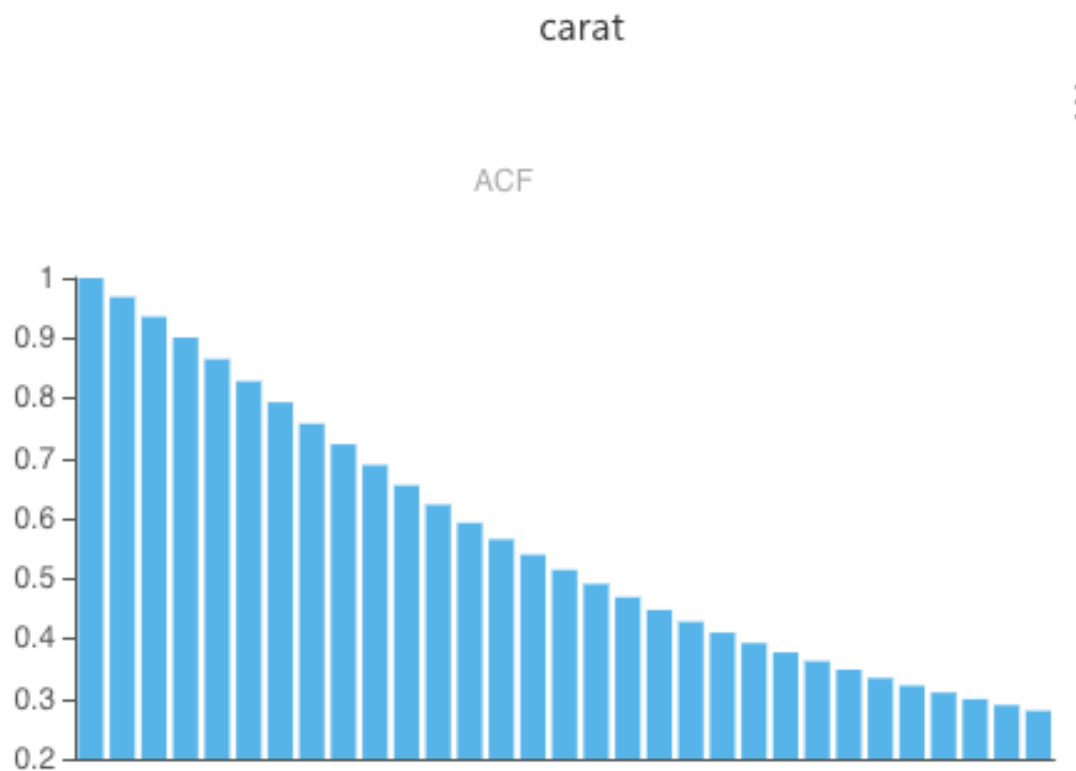


The trace plots the generated values of the parameter against the iteration number. We can see that each state is correlated to the previous — the Markov Chain — but the values oscillate significantly — the Monte Carlo sampling. Given a sufficient number of iterations, MCMC will converge to the true value. However, assessing convergence can be difficult.

When evaluating these trace plots, we are expecting a stationary distribution that looks like white noise. This trace plot looks like it might have a little autocorrelation.

### Autocorrelation function (ACF)

ACF measures how correlated the values in the chain are with their close neighbours. The lag is the distance between the two chains to be compared. An independent chain will have approximately zero autocorrelation at each lag. At lag  $k$ , this is the correlation between series values that are  $k$  intervals apart.



The x-axis indicates the lag at which the autocorrelation is computed. The y-axis indicates the value of correlation (between -1 and 1). For example, a spike at lag 1 in an ACF plot indicates a strong correlation between each series value and the preceding value, a spike at lag 2 indicates a strong correlation between each value and the value occurring two points previously, and so on.

- A positive correlation indicates that large current values correspond with large values at the specified lag; a negative correlation indicates that large current values correspond with small values at the specified lag.
- The absolute value of a correlation is a measure of the strength of the association, with larger absolute values indicating stronger relationships.

### Bayesian G-prior spike slab models

## Parameter Settings ●

### Linear

Bayesian



### Regressors



Horseshoe



G-prior

G-prior Spike Slab

Jeffreys

Horseshoe

Normal Inverted Gamma

Normal Inverted Gamma Spike Slab

Parameter Settings

Linear

Bayesian

Regressors

G-prior Spike Slab

g

53920

Random Seed

12345

Iterations

1000

Burnin

500

In regression, where there are  $k$  potential explanatory variables, the full model likelihood can be specified as  $y|\beta, X, \sigma^2 \sim N(\beta_k X, \sigma^2 I)$  where  $K = \{0, 1, 2, \dots, k\}$  possible regressors ( $K=0$  indicates intercept term).

For explanatory power, we estimated the **most probable models** using a G-prior spike slab algorithm. The unknown parameters  $(\beta, \sigma)$  require a prior distribution to be specified in order to estimate their respective posterior distributions. The G-prior (Zellner, 1986) allows the user to provide an estimate of the parameters likely location, but does not require the specification of a correlation structure between regressors. This makes it a useful prior for model comparison. The prior distributions for the parameters  $\beta_K$  and  $\sigma$  for the full model

are:  $\beta_K | \sigma, X \sim N_{k+1}(\tilde{\beta}, g\sigma^2(X^T X)^{-1})$   $\sigma^2 | X \sim N_{k+1}(\tilde{\sigma}^2, g\sigma^2(X^T X)^{-1})$

where  $g$  is specified to reflect the certainty of the prior.

AutoStats default is to set  $g$  equal to our sample size, which essentially indicates that prior distribution has an influence similar to one observation. Hence, the posterior distribution becomes data driven. The hyper-prior  $\tilde{\beta}$  is set at zero. It is often the case that variables in the full model exhibit collinearity, making assessment of each variables contribution to the outcome problematic. For example, even the sign of the coefficient may give an erroneous indication of that variables role in the outcome, as each coefficient cannot be thought of in isolation. Variable selection for dimension reduction, and then averaging over probable models, is a common technique for approaching this issue. The need to assess  $2^k$  competing models (intercept assumed included), requires a strategy to traverse the space. AutoStat uses stochastic

search variable selection, where each of the  $2^k$  models ( $M_{\gamma}$ ) is associated with a binary vector  $\gamma$  where  $\gamma_j = 1$  when  $\beta_j$  is in the model and 0 otherwise.

As the model probability, ( $M_{\gamma}$ ), is now another parameter to be estimated, we assign the prior as a uniform distribution for all models. The stochastic search algorithm of Marin and Robert, 2007 (p82) is then used to determine the posterior probability of each model.

This button will take you to the **Model Builder** where you can set your priors, iteration numbers, favourite random seed and the run the model using the analyse button. The results should give you 6 Tabs of output.

As we mentioned before, the Pair Plot shows the relationship between carat and Length and the relationship between carat and Height are non linear and like curve. For deeper exploring the relationship of them. We need to make carat and Length, Height spline first.

The screenshot shows a software interface with a 'Variables' panel on the left and a 'Define Variables' panel on the right. The 'Variables' panel lists: # carat, abc cut, abc color, abc clarity. The 'Define Variables' panel has a 'Forecast Variable(s)' section with 'price' selected. A modal dialog titled 'Use Spline Basis' is open in the foreground. It contains the following fields and options:

- Name:** A text box containing 'carat'.
- Type:** A dropdown menu with 'Regression Spline' selected.
- Options:** A dropdown menu with 'Natural Cubic Spline' selected.
- Knots Number:** A text box containing '5'.
- Spline Knots:** A list of values in boxes: [ 0.65, 1.1, 1.56, 2.01, 2.46 ].
- A link: [Add evaluate gradient](#)
- Buttons: 'Done' and 'Cancel'.

For exploring the relationship between carat and table for price, we need to change explanatory variables and click analyse:

Define Variables

Forecast Variable(s)

price

Explanatory Variable(s)

cut

color

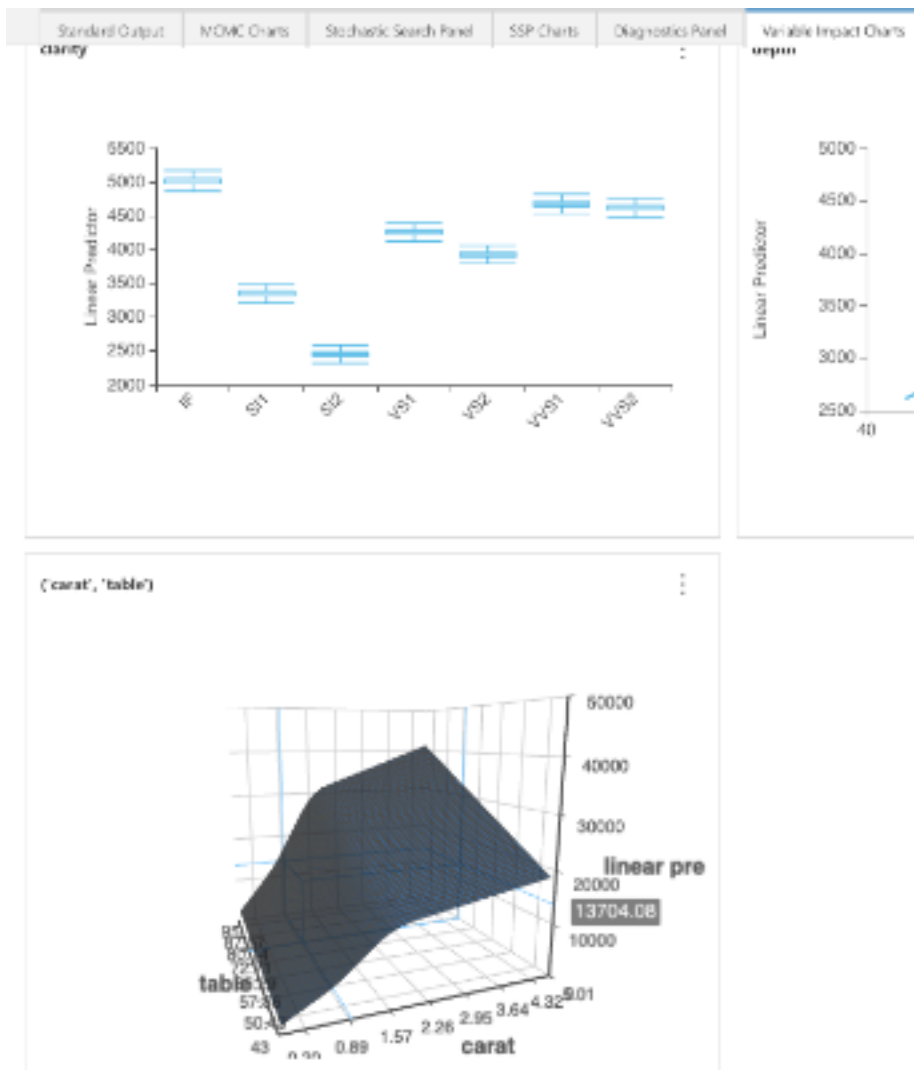
clarity

depth

carat

table

Go to **Variable Impact Charts**. If you scroll down, you can see the 3D chart:



The 3D chart shows that table and carat are positive relationship contributing to price. It seems funny.

Let's explore the relationship of height and carat contributing to price:

### Define Variables •

Forecast Variable(s)

price

Explanatory Variable(s) •

cut

color

clarity

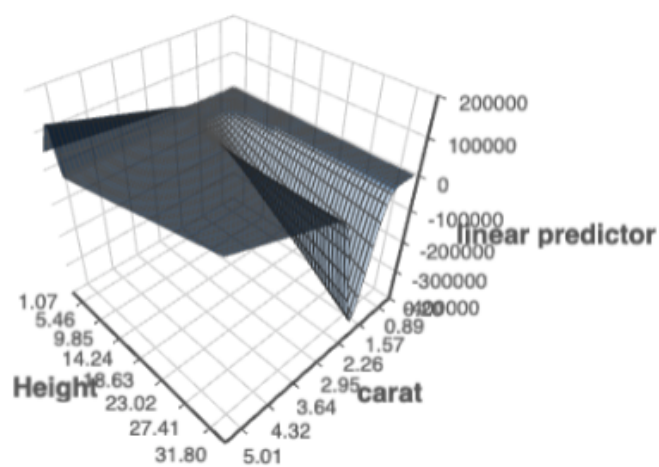
depth

table

carat

Height

('carat', 'Height')





The 3D chart increase the price most related to carat increase. Height seems without relationship contributing to price related to carat. You can try different variable relationships by using this module.