Introduction to Data Mining and Machine Learning

Shaina Race, PhD
Institute for Advanced Analytics
North Carolina State University

Preparing for Model Validation

• • •

Splitting into Training/Validation/Test Sets

Deciding on Cross Validation

Data Preprocessing

- When you **first** receive your data, **you'll explore** for distributions/outliers, and missing values.

- Before you look at any relationships between input variables and target variables, you should split into training, validation and test samples.
  (Or decide on Cross-Validation / Testing)

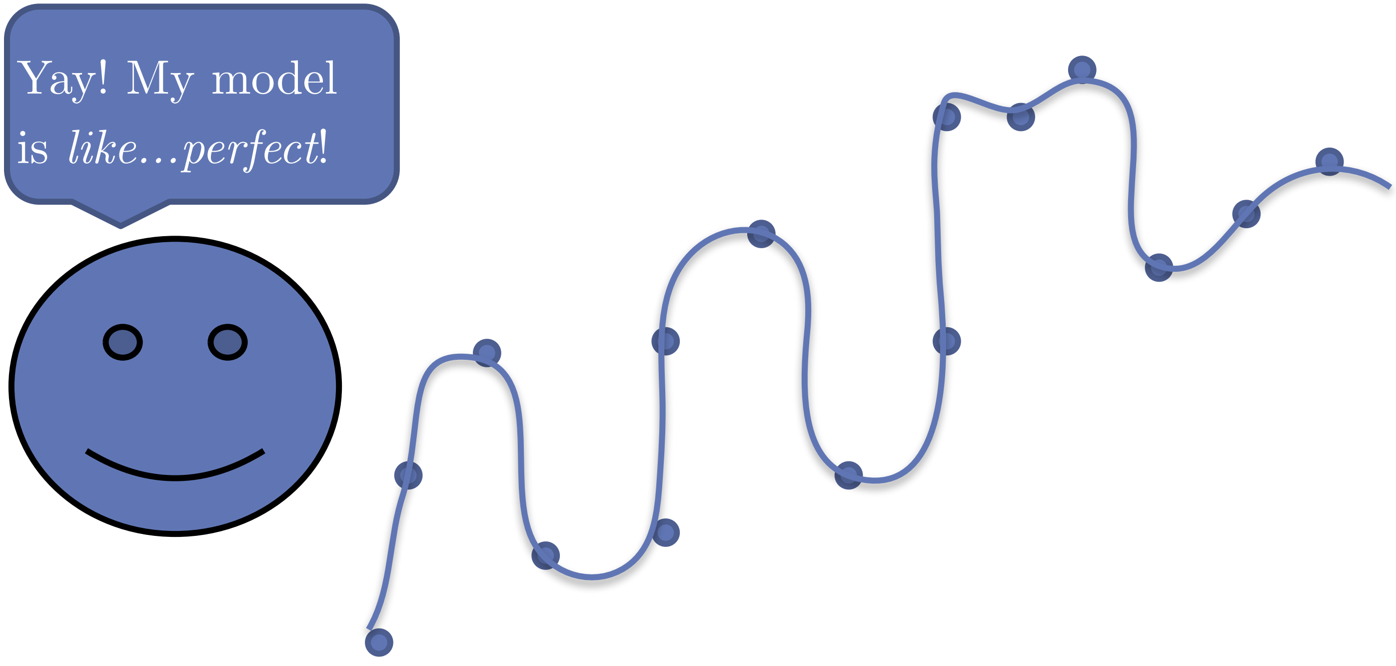
The Problem of Overfitting

- Left unchecked, **models will capture nuances of the data on which they're built** (the training data).

- When these "patterns" do not hold up in validation or test data, the model performance suffers. **We say the model does not generalize well. The model is overfit.**

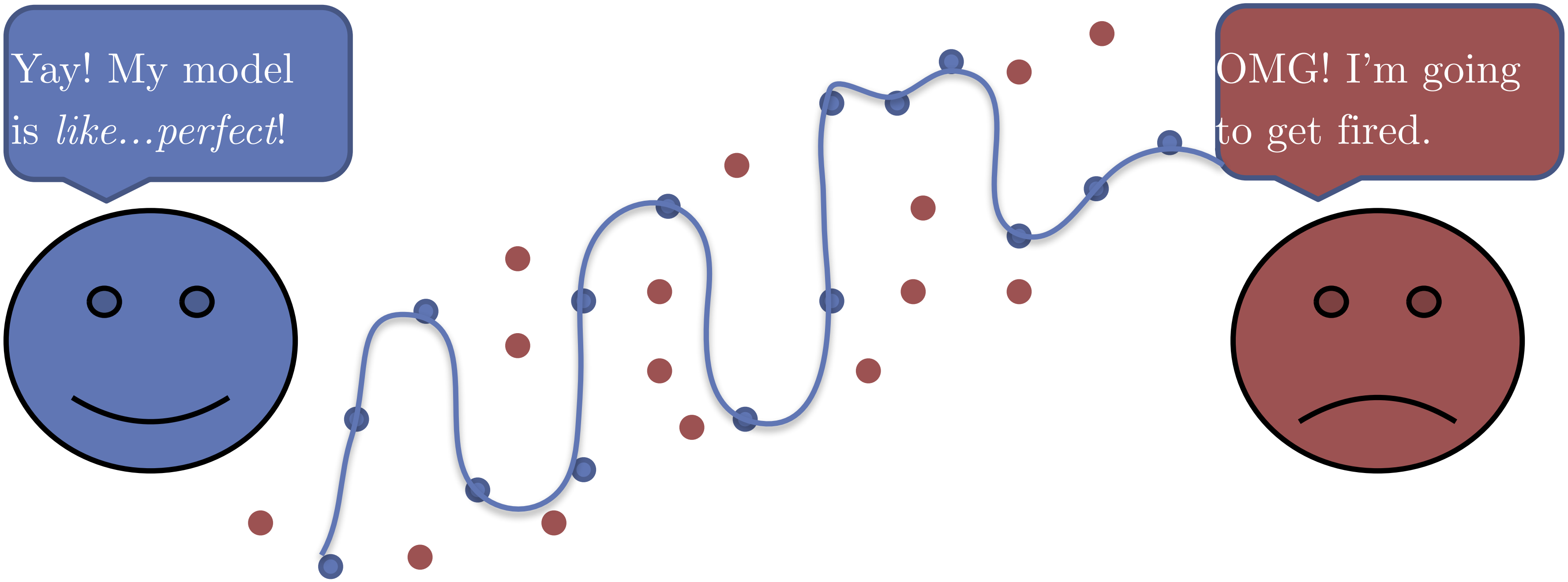
The Problem of Overfitting

- Error on the training data *does not* predict *future* performance.
- Complexity can undermine model's performance on future data.



The Problem of Overfitting

- Error on the training data *does not* predict *future* performance.
- Complexity can undermine model's performance on future data.



The Bias-Variance Tradeoff

- Bias = underfitting
  - The modeled value's distance from "truth".
  - Want a model with low bias.
- Variance = overfitting
  - The model parameters will vary greatly on different training samples.
  - Want a model with low variance.

The concepts are inversely related:

Lower Bias → Higher Variance

Lower Variance → Higher Bias.

(Hence the term "Bias-Variance ***Tradeoff***")

Training/Validation/Test

- Want to make sure your models are generalizable
  - Not just good models of training sample.
  - Can predict equally well on out-of-sample data.

- Split into Training + Validation + Test sets is necessary
  - Somewhere around 2/3 training, 1/3 validation/test is typical.
    - Lots of data? 50-40-10 split
    - Not so much data? 70-20-10 split
    - Not enough data? Use Cross-Validation

Training/Validation

- Use the Training set to build your model.
- Evaluate and tune the model based on how it performs on the validation data
- **Never** report accuracy metrics from training set!

Testing

- Continually adapting a model to perform better on validation data essentially trains the model to the validation data.

- Once you've chosen a final model, re-run it on (training+validation) data to finalize your parameters, and report accuracy on test data.

- Before deploying that final model to the customer, you can update parameters using **entire** dataset.

10-Fold Cross Validation:

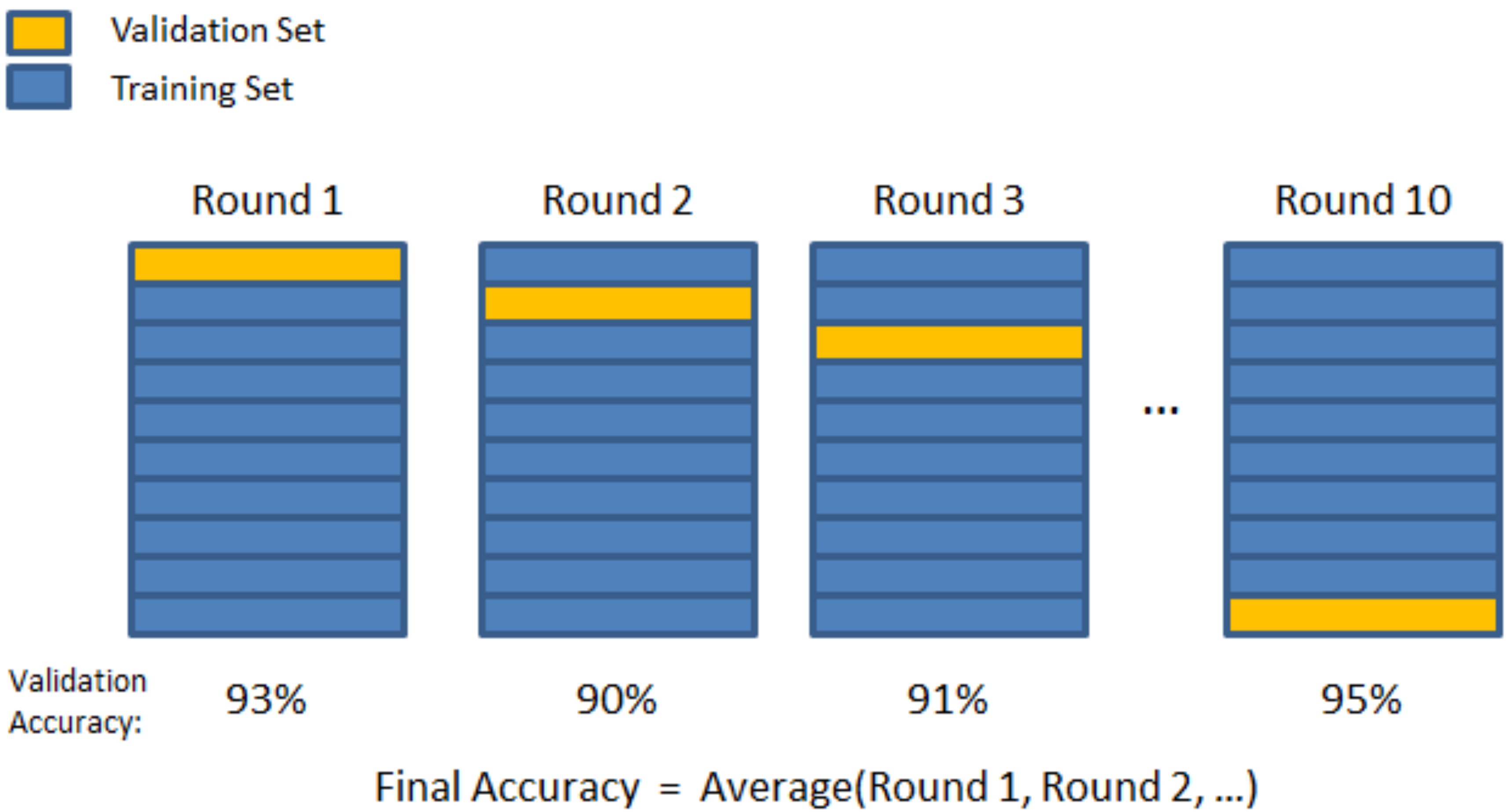


Image by Chris McCormick.

K-fold Cross-Validation Summary

- Divide your data into k equally-sized samples (*folds*)
  - k=10 or k=100 are common.
  - Depends on time complexity of model and size of the dataset!
- For each fold, train the model on all other data, using that fold as a validation set
- Record measures of error/goodness-of-fit
- In the end, report summary of error/goodness-of-fit measure (average, std. deviation etc)
- Use that report summary to choose a model

Cross-Validation

- Can use cross-validation in any situation.

- Will be necessary if you do not have **sufficient** observations to split into training/validation/test

- What is **sufficient**? It depends!
  - **Rule of thumb:** AT LEAST 10 observations per input variable in training set
  - Don't Forget: For **categorical variables – each level counts!**

Leave-One-Out Cross-Validation (Jackknife)

- n-fold cross validation where n is number of obs.
- Use only one observation as the validation-set
- Repeat for every observation in the dataset

- **Can be extremely time consuming! Only use when necessary (very small sample sizes)**

Dealing with Transactional Data

• • •

Moving from Long to WIDE

Transactional Data

Transactional data is **long** and has many rows per modeling observation.

CustID	Date	Items	Cost
2	10/10	10	100
2	10/12	5	20
2	12/4	1	2
9	10/03	25	46
9	10/04	5	12
12	10/01	20	300
12	12/27	20	300
12	12/28	21	301

Transactional Data

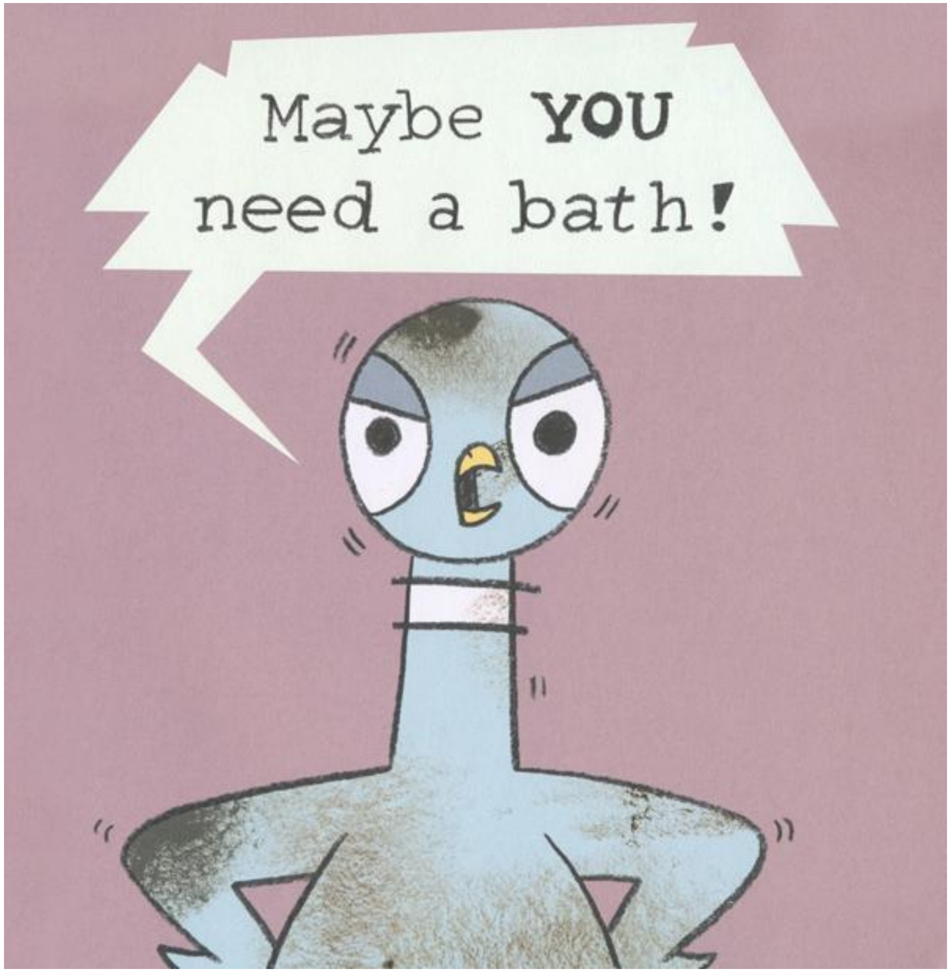
- Typically, the solution for modeling with transactional data is to “roll it up” so it has one row per observation modeled.
- It is transformed from long to **<u>wide</u>**
- In this case, we’d have 3 observations (customers)
- One big *group by* SQL query

Transactional Data

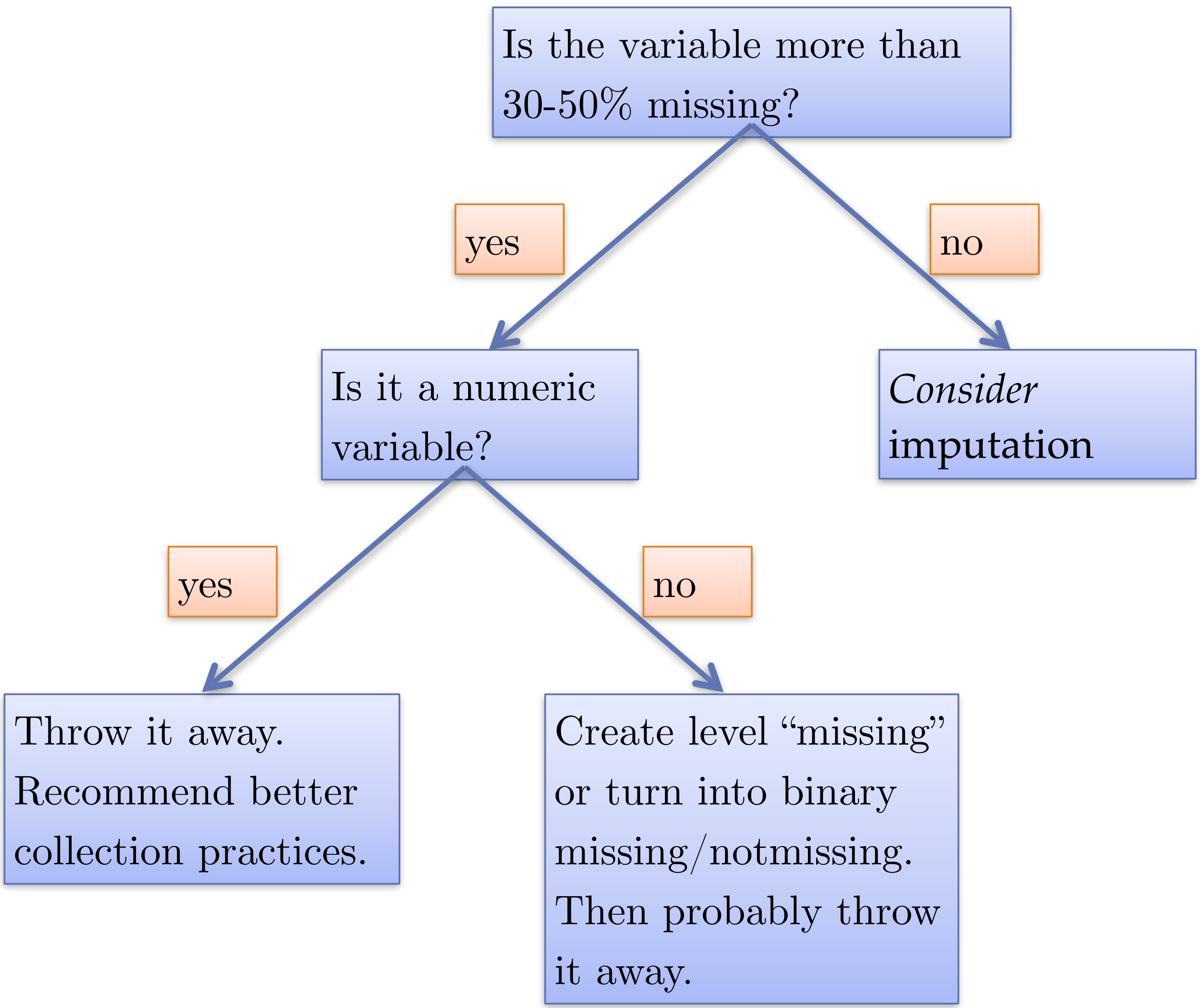
A **subset** of columns we might consider in the process:

1. ID
2. Date of first transaction
3. Date of last transaction
4. Total number of transactions
5. Average time between transactions
6. Maximum number of items purchased
7. Average number of items purchased
8. Minimum number of items purchased
9. Std Deviation of number of items purchased
10. Maximum cost of items purchased
11. Average cost of items purchased
12. Minimum cost of items purchased
13. Stand. Deviation of cost of items purchased
14. Slope of regression line of cost over time

Data Cleaning
Handling Missing Values



Handling Missing Values



Missing Value Imputation

Imputation: Replacing missing values with a substitute value, typically a guess at what you think the value should have been.

★ i.e. falsifying records. making up data.

Imputing Missing Values

- *Always* Always *Always* create a binary flag $= 1$ indicating that the value has been imputed and include the flag in your model.

Obs.	Gender	Q1 Response
1	M	5
2	M	4
3	F	NA
4	M	1
5	F	NA

Obs.	Gender	Q1 Response	Q1 Flag
1	M	5	0
2	M	4	0
3	F	3	1
4	M	1	0
5	F	3	1

- Nonresponse might be an important indicator of target or relate to another variable.

Categorical Variables

- Option one: Create **new level** of variable as "**missing.**" (No flag necessary in this case.)

- Option two: Replace missing values with the **mode**.

- Option three: Try to **predict** the missing value using other attributes.

    - Decision trees, RandomForests, KNN methods popular for missing values (Coming Fall 2 & 3)
    - "Hotdeck imputation" – PROC SURVERYIMPUTE method = hotdeck

Numeric Variables

- <u>Option one</u>: Replace missing values with the **mean**
- <u>Option two</u>: Replace missing values with the **median** (for skewed distributions).
- <u>Option three</u>: **Predict** the missing values using other attributes.
    - Multiple Regression or Regression Trees popular
- <u>Option four</u>: **Discretize** (bin) the numeric variable into categories and create 'missing' category.

Ordinal Variables

- Depends on the variable
- Likely to treat 'level of education' differently than 'Likert scale response'
- Use one of the options prescribed for numeric or categorical variables

More Sophisticated Approaches

- Previously mentioned approaches are simple but naïve.
- More sophisticated methods exist that are more complicated but principled.
- These will be **necessary** for statistical inference!

More Sophisticated Approaches

Numeric Variables

- Maximum Likelihood Imputation
  - EM Algorithm in R
  - PROC MI Default
- Multiple Imputation.
  - PROC MI and PROC MIANALYZE
  - MICE package in R

Categorical Variables

- Fully Efficient Fractional Imputation (FEFI)
  - PROC SURVEYIMPUTE default
  - FHDI package in R

Pay Attention!

- Blind imputation can potentially generate impossible or highly unlikely data
- For Example:
  - A 16 year old who makes $80,000 a year
  - A male patient who is menopausal

So what *should* I *DO*?

It Depends!!

- Only the person closest to the data and to the problem can make these judgment calls!

- Can try several methods to see what works best.

- The binary flag indicating imputed value will show you if there is something special about missing values.

More Information for Self-Study

• • •

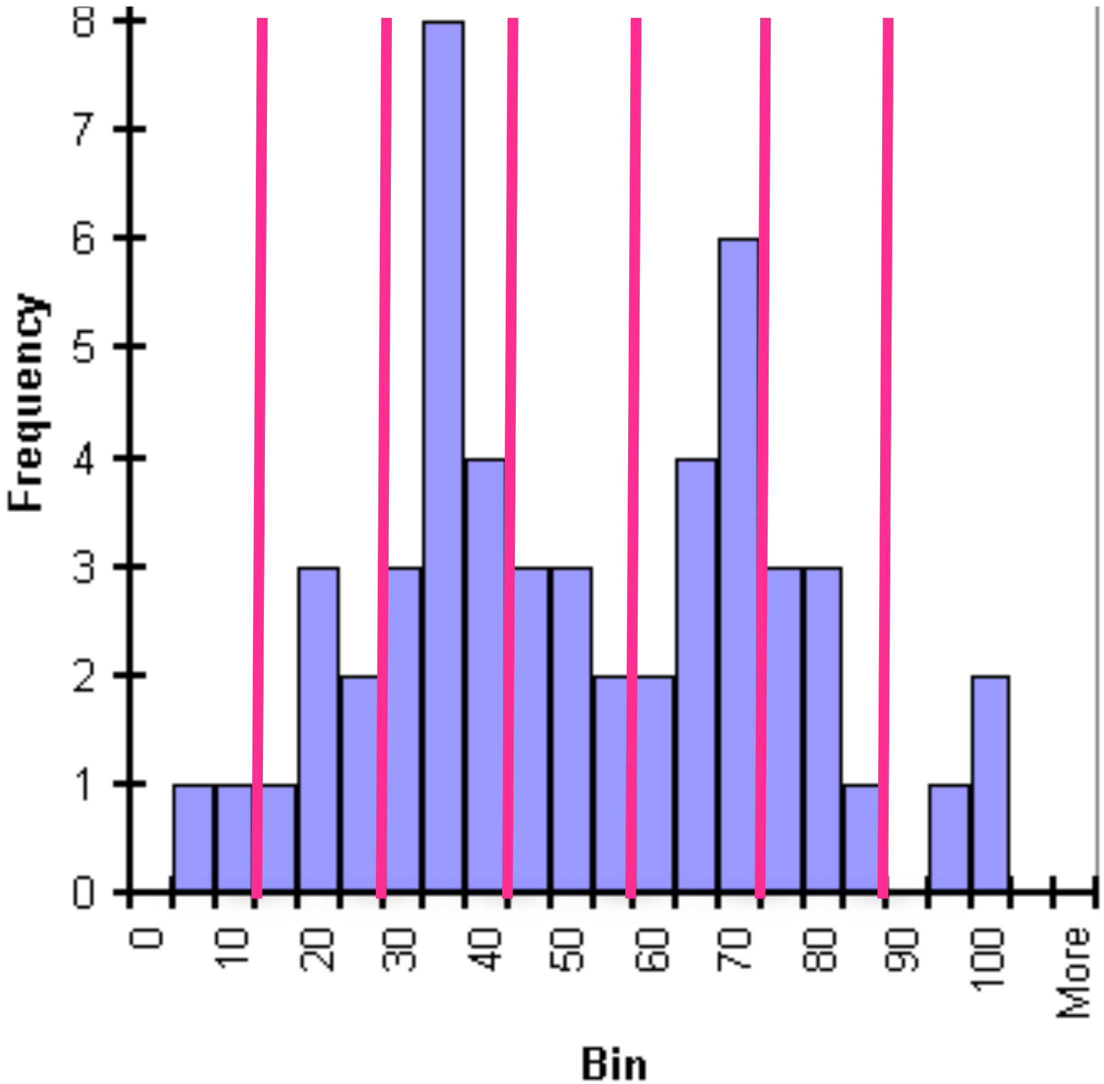
Variable Transformations

Variable Transformations

- Discretizing (Binning) Numeric Variables
  - Equal Width
  - Equal Depth
  - Supervised Binning
- Standardization and Normalization
  - Statistical Standardization
  - Range, MinMax Standardization
  - Considerations
- Log Transformation and Percent Change

Binning Numeric Variables

Unsupervised Approach 1: *Equal Width*



Each bin has the same width in variable values

Each bin has a different number of observations

Binning Numeric Variables

Unsupervised Approach 2: *Equal Depth*

Name	Team	nAtBat
Bochy, Bruce	San Diego	127
Simmons, Ted	Atlanta	127
Daulton, Darren	Philadelphia	138
Spilman, Harry	San Francisco	143
Howell, Jack	California	151
Speier, Chris	Chicago	155
Porter, Darrell	Texas	155
Dwyer, Jim	Baltimore	160
Meacham, Bobby	New York	161
Willard, Jerry	Oakland	161
Reed, Jeff	Minneapolis	165
Rivera, Luis	Montreal	166
Puhl, Terry	Houston	172
O'Malley, Tom	Baltimore	181
Daniels, Kal	Cincinnati	181
Robidoux, Billy Jo	Milwaukee	181
Beane, Billy	Minneapolis	1

Take percentiles of the population.

Each bin has the same number of observations.

Binning Numeric Variables

Supervised Approach

- Use target variable info to 'optimally' bin numeric variables for prediction.
- *Typically* used in classification problems.
- Want bins that result in the most *pure* set of target classes.

Binning Numeric Variables

Supervised Approach

Incom	Vehicle Color
15K	mixed
19K	brown
20K	mixed
50K	blue
55K	green
60K	blue
65K	blue
85K	green
150K	mixed
175K	red
995K	mixed

Binning Numeric Variables

Supervised Approach

- Decision tree methods can be helpful to create these bins.
- Also, weight of evidence
- More on these techniques later.

Standardization and Normalization

- Standardization in statistics (Z-score standardization) transform units to “number of standard deviations away from the mean”:

$$\frac{x - \bar{x}}{\sigma_x}$$

- Avoid having variable with large values (e.g. income) dominate a calculation.
- Many other ways to standardize/normalize
  - Range Standardization: Divide by the range of the variable
  - MinMax Standardization: Subtract min. and divide by (max-min.)
    - Puts variable on a scale from 0 to 1
  - Divide by 2-norm, Divide by 1-norm, Divide by sum

Transformation Considerations

- Transformations change the nature of the data.
    - Ex: x={1,2,3} transform to $1/x = \{1, \frac{1}{2}, \frac{1}{3}\}$
        - The sorting order of the observations reverses
        - Observations close to 0 will get **very** large

- Always consider the following questions:
    - Does the order of the data need to be maintained? (other code/documentation)
    - Does the transformation apply to all values, especially negative values and 0? (Think log(x) and 1/x)
    - What is the effect on values between 0 and 1?

Interpreting Logarithmic Transformations in Linear Models

Logarithm on Independent Variable

$$y = a\log(x)$$

1% increase/decrease in x implies y increases/decreases by 0.01a units

This interpretation only valid for changes of up to +/- 20%

Example: Logarithm on Independent Variable

$$\textbf{oil\_consumption} = \mathbf{2} \cdot \mathbf{log(GDP)}$$

- 1% increase in GDP implies $0.01 \cdot 2 = 0.02$ **unit increase** in oil consumption.
- 5% decrease in GDP implies $0.05 \cdot 2 = 0.1$ **unit decrease** in oil consumption.

This interpretation only valid for changes of up to +/- 20%

Logarithm on Dependent Variable

$$\log(y) = a\,x$$

1 unit increase/decrease in x implies y increases/decreases by a%

This interpretation only valid for changes of up to +/- 20%

Example: Logarithm on Dependent Variable

$$\log(\text{oil\_consumption}) = \mathbf{2} \cdot \text{GDP}$$

- 1 unit increase in GDP implies 2% increase in oil consumption.
- 5 unit decrease in GDP implies 10% decrease in oil consumption.

This interpretation only valid for changes of up to +/- 20%

Logarithm on Both Variables

$$\log(y) = a \log(x)$$

1% increase/decrease in x implies y increases/decreases by a%

This interpretation only valid for changes of up to +/- 20%

Example: Logarithm on both variables

More concrete example:

$$\log(\text{oil\_consumption}) = 2 \cdot \log(\text{GDP})$$

- 1% increase in GDP implies 2% increase in oil consumption.
- 5% decrease in GDP implies 10% decrease in oil consumption.

This interpretation only valid for changes of up to +/- 20%

Details for Logarithmic Interpretation

• • •

Why is it only valid for changes up to ~20%?

Log Transformation and Percent Change

$$y = \alpha \log(x)$$

% change in x

Additive change in y for coefficient of 1.

r	log(1+r)
-50%	-0.693
-40%	-0.511
-30%	-0.357
-20%	-0.223
[illegible]	-0.105
[illegible]	-0.051
[illegible]	-0.020
[illegible]	0.000
[illegible]	0.020
[illegible]	0.049
[illegible]	0.095
[illegible]	0.182
[illegible]	0.262
[illegible]	0.336
[illegible]	0.405
100%	0.693

For an *r* percent increase in x, what happens to y?

$$y = \alpha \log(x)$$
$$y' = \alpha \log(x(1+r))$$
$$y' = \alpha[\log(x) + \log(1+r)]$$

so

$$y' - y = \alpha \boxed{\log(1+r)} \approx \alpha \boxed{r}$$

Log Transformation and Percent Change

$$\log(y) = \alpha \log(x)$$

% change in x

Additive change in y for coefficient of 1.

For an r percent increase in x, what happens to y?

$$\log(y) = \alpha \log(x)$$

$$\log(y') = \alpha \log(x(1+r))$$

$$\log(y') = \alpha[\log(x) + \log(1+r)]$$

so

$$\log(y') - \log(y) = \alpha \log(1+r) \approx \alpha r$$

$$\log\left(\frac{y'}{y}\right) = \alpha \log(1+r) \approx \alpha r$$

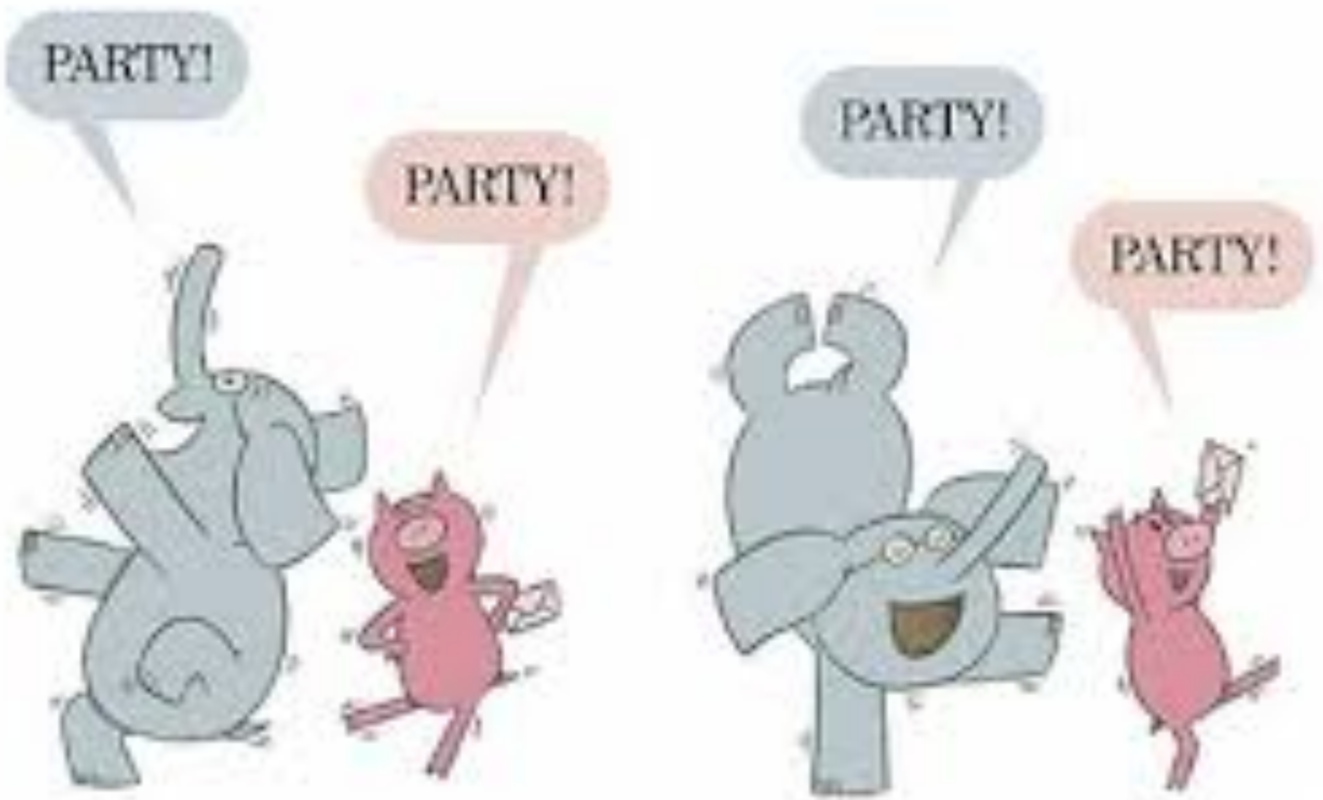
$$\log(1+r_y) \approx \alpha r$$

$$r_y \approx \alpha r$$

where r_y is percent increase in y.

r	log(1+r)
-50%	-0.693
0%	-0.511
0%	-0.357
0%	-0.223
0%	-0.105
%	-0.051
%	-0.020
%	0.000
%	0.020
%	0.049
%	0.095
%	0.182
%	0.262
%	0.336
%	0.405
0%	0.693

So what *should* I ***DO***?



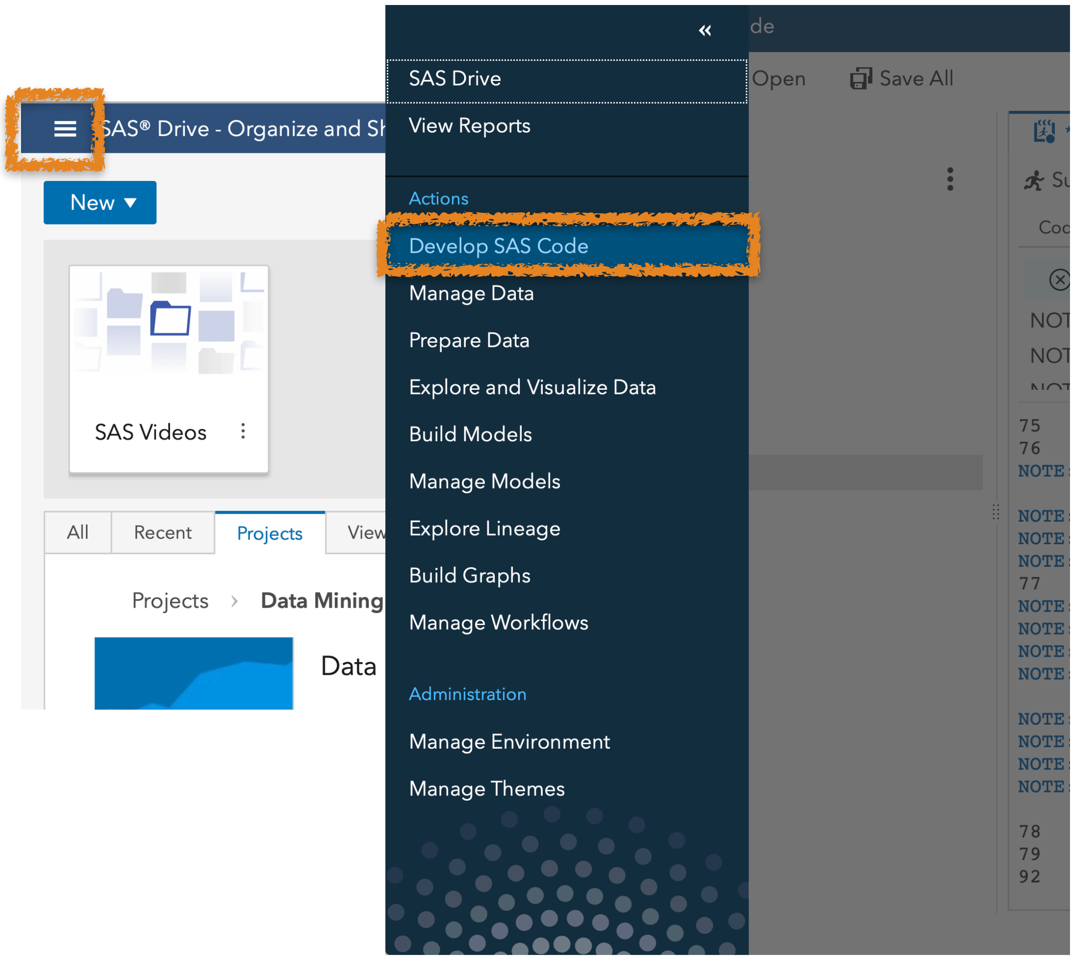
It Depends!!

- Only the person closest to the data and to the problem can make these judgment calls!

- Can try several methods to see what works best.

- Transformations are typically either required to meet assumptions of a model, or something done in hindsight to improve performance of a given model.

SAS Viya Introduction

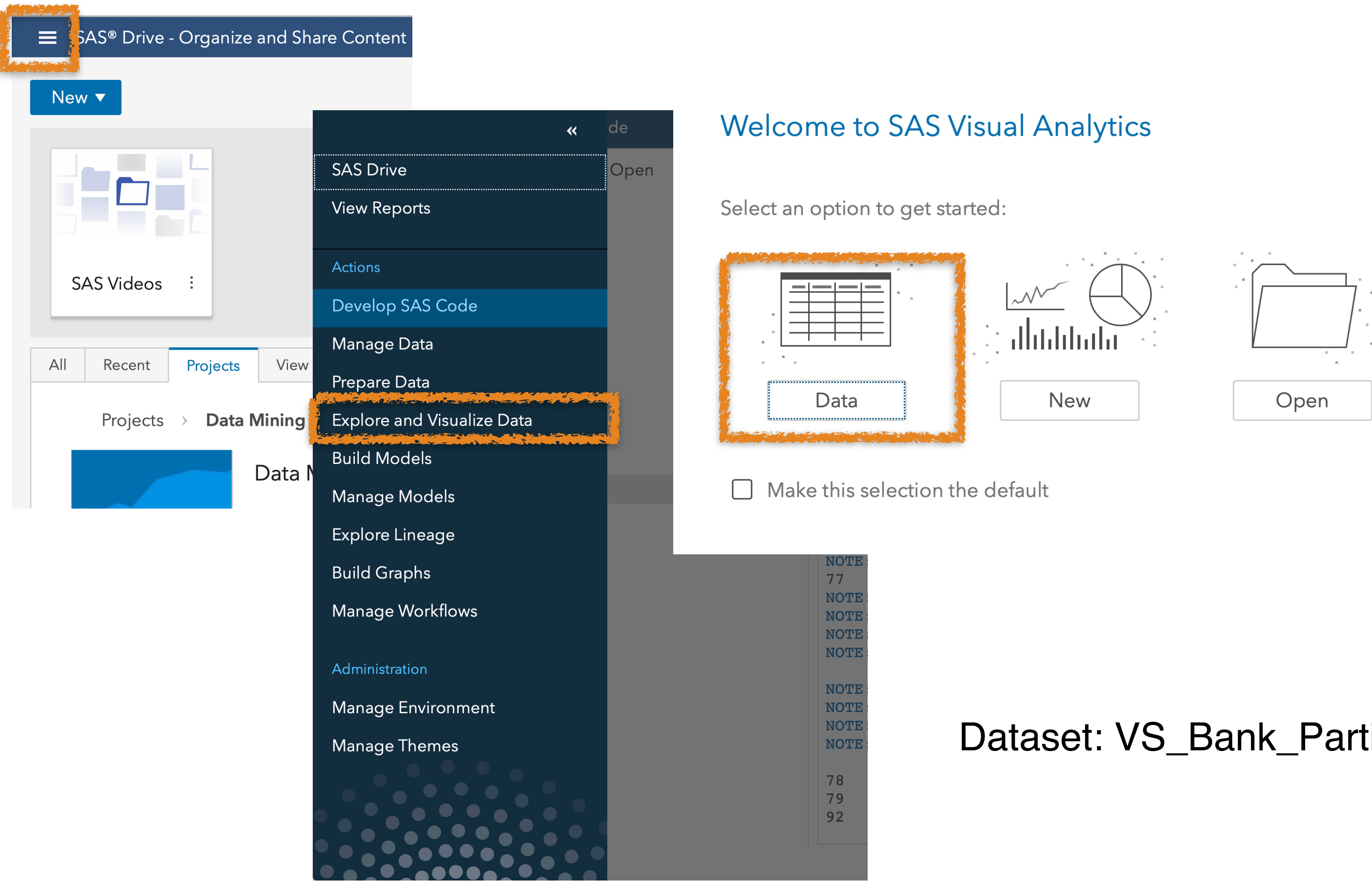


Submit Code:

```
cas;
caslib _all_ assign;
```

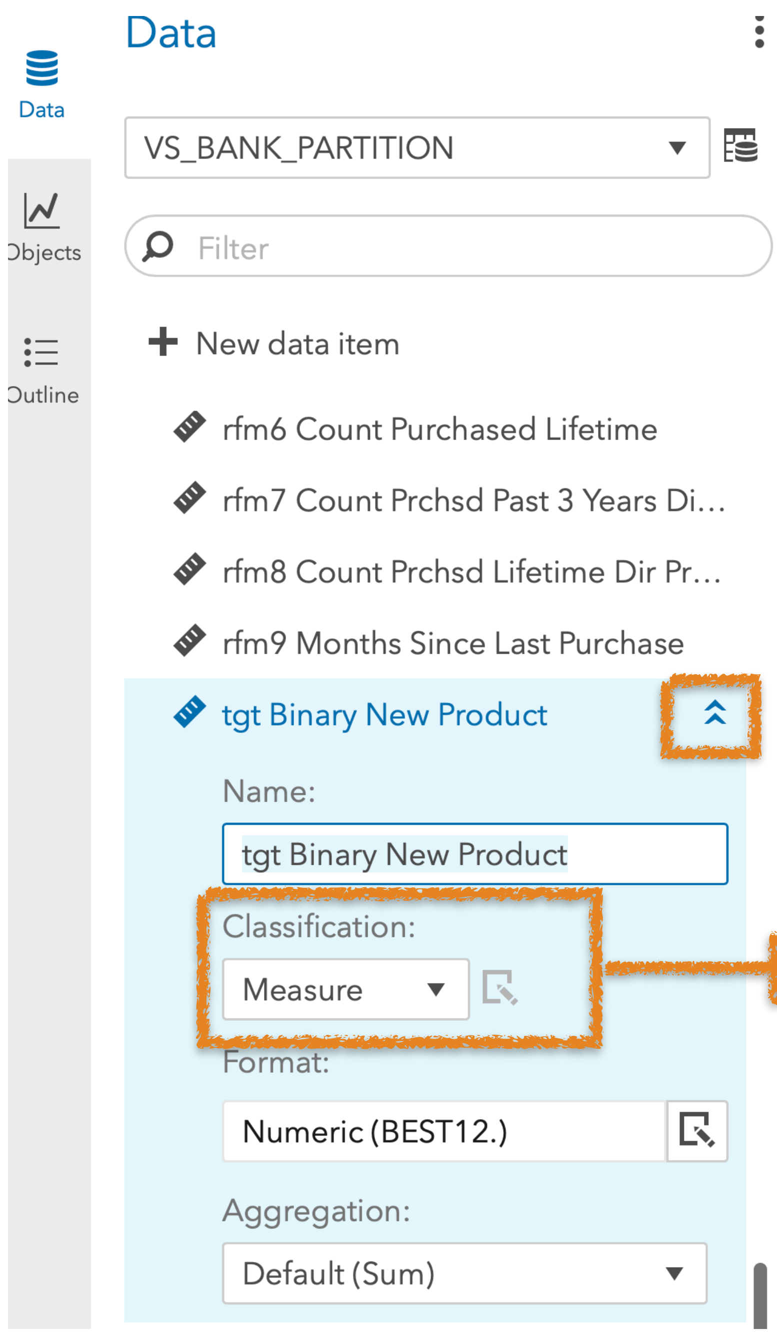
You will repeat this step EVERY time you use Viya to load the Public library!

SAS Viya Introduction



Dataset: VS_Bank_Partition

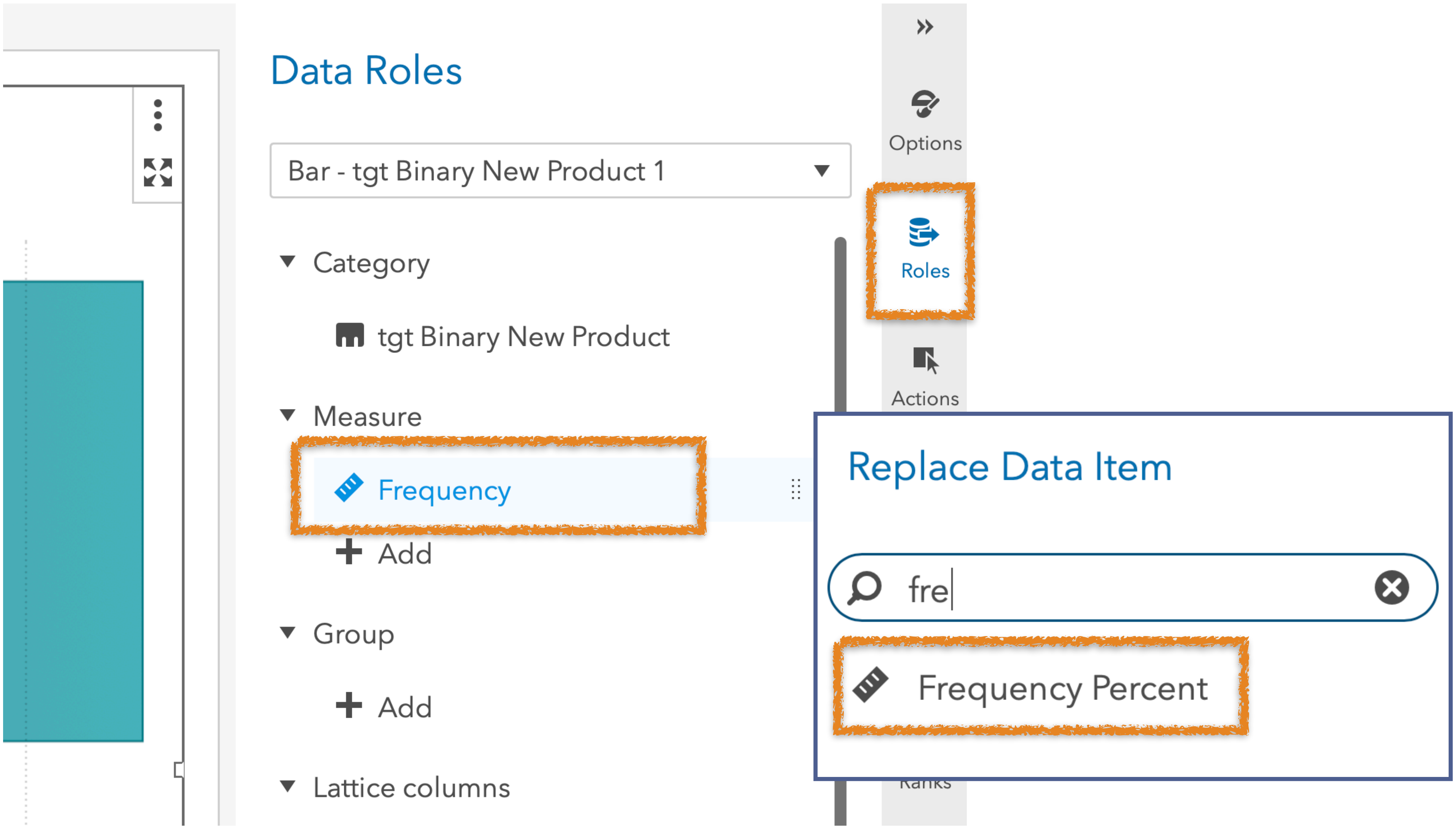
Explore Target Variable



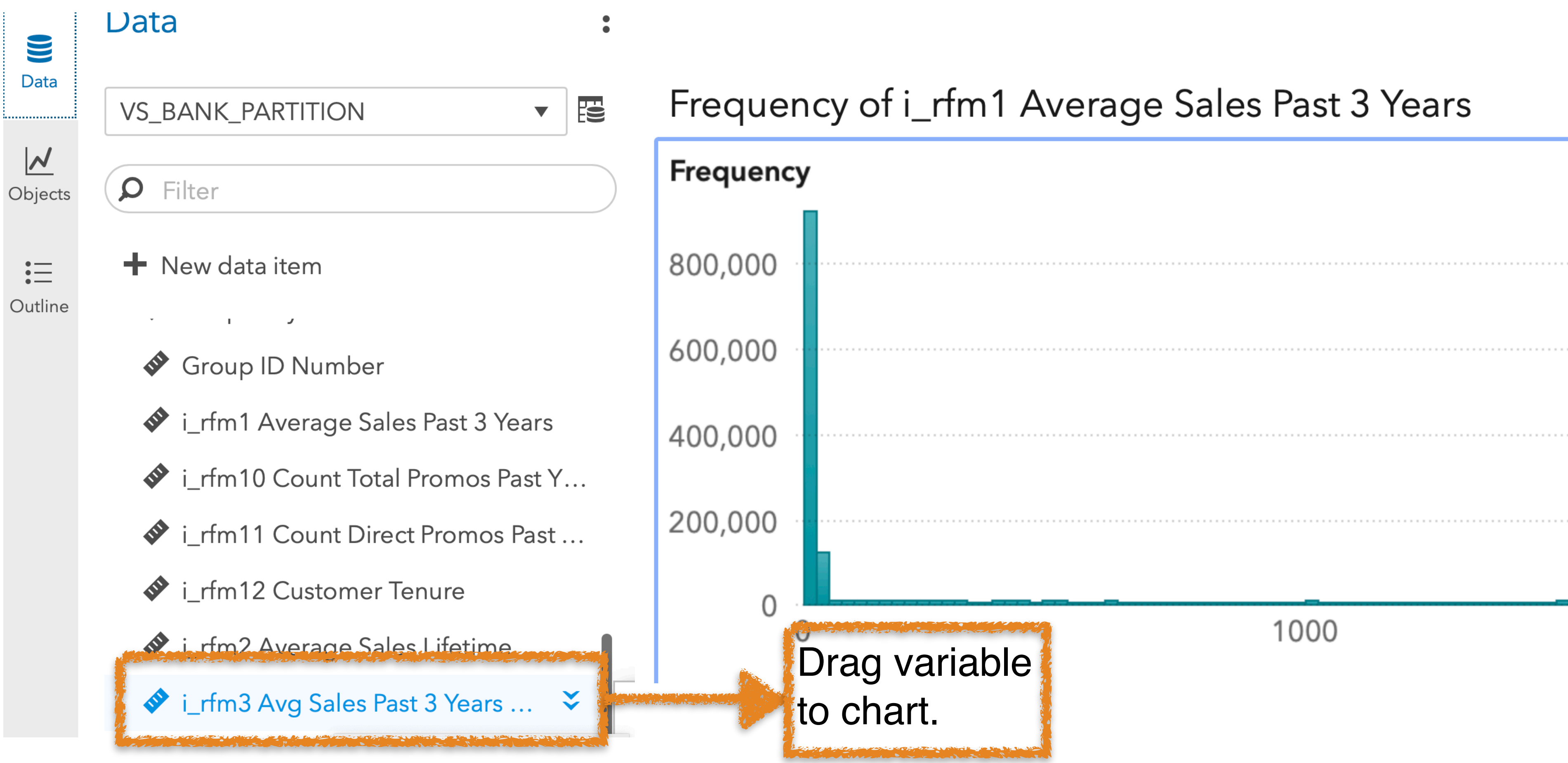
Change to Category

Scroll up and drag variable to chart.

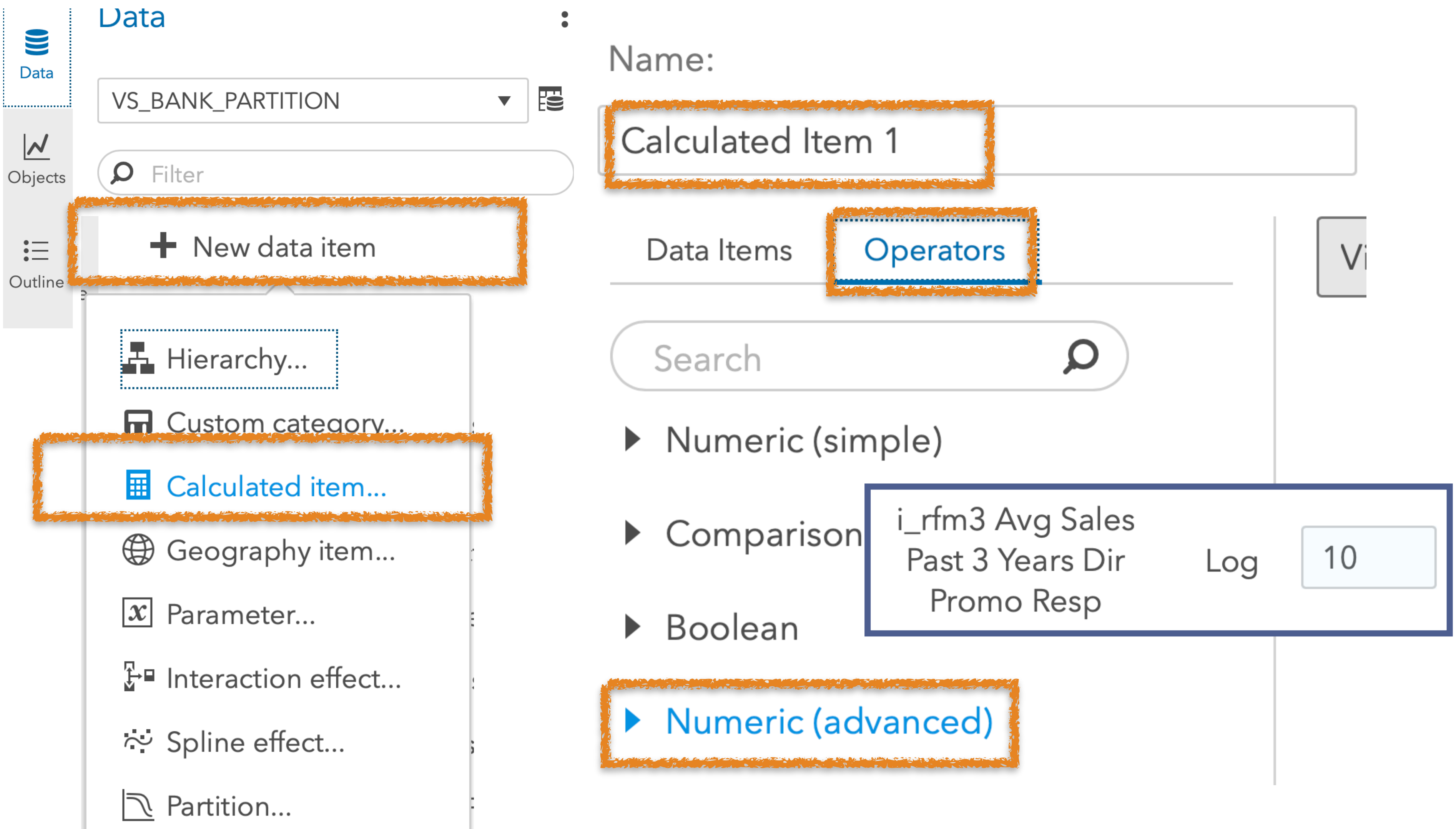
Change Chart to %



Explore/Transform

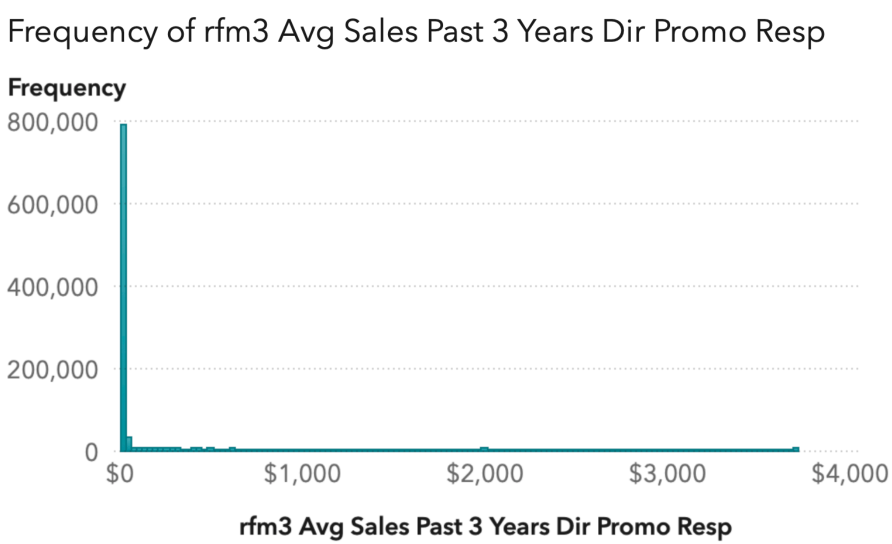


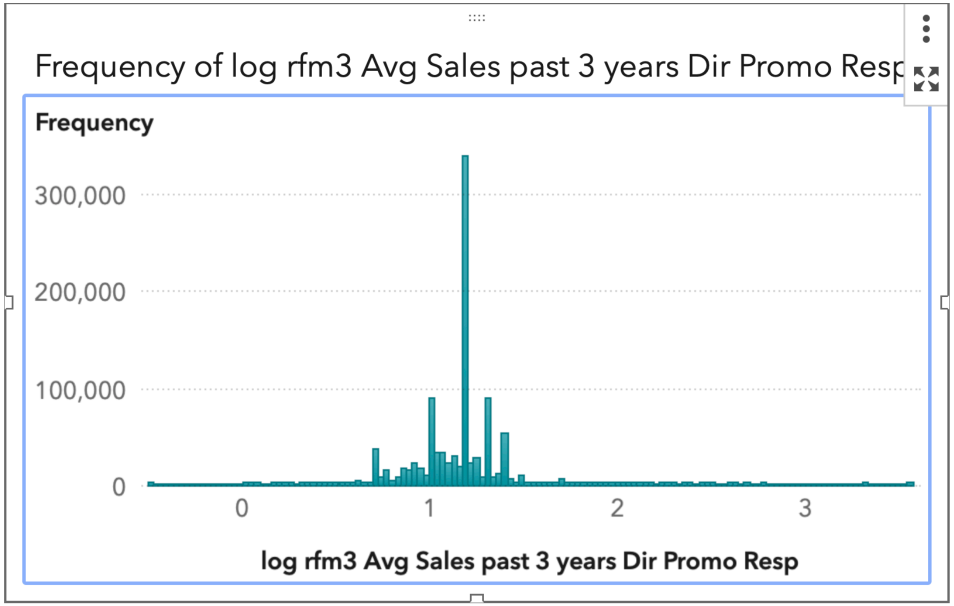
Log Transform



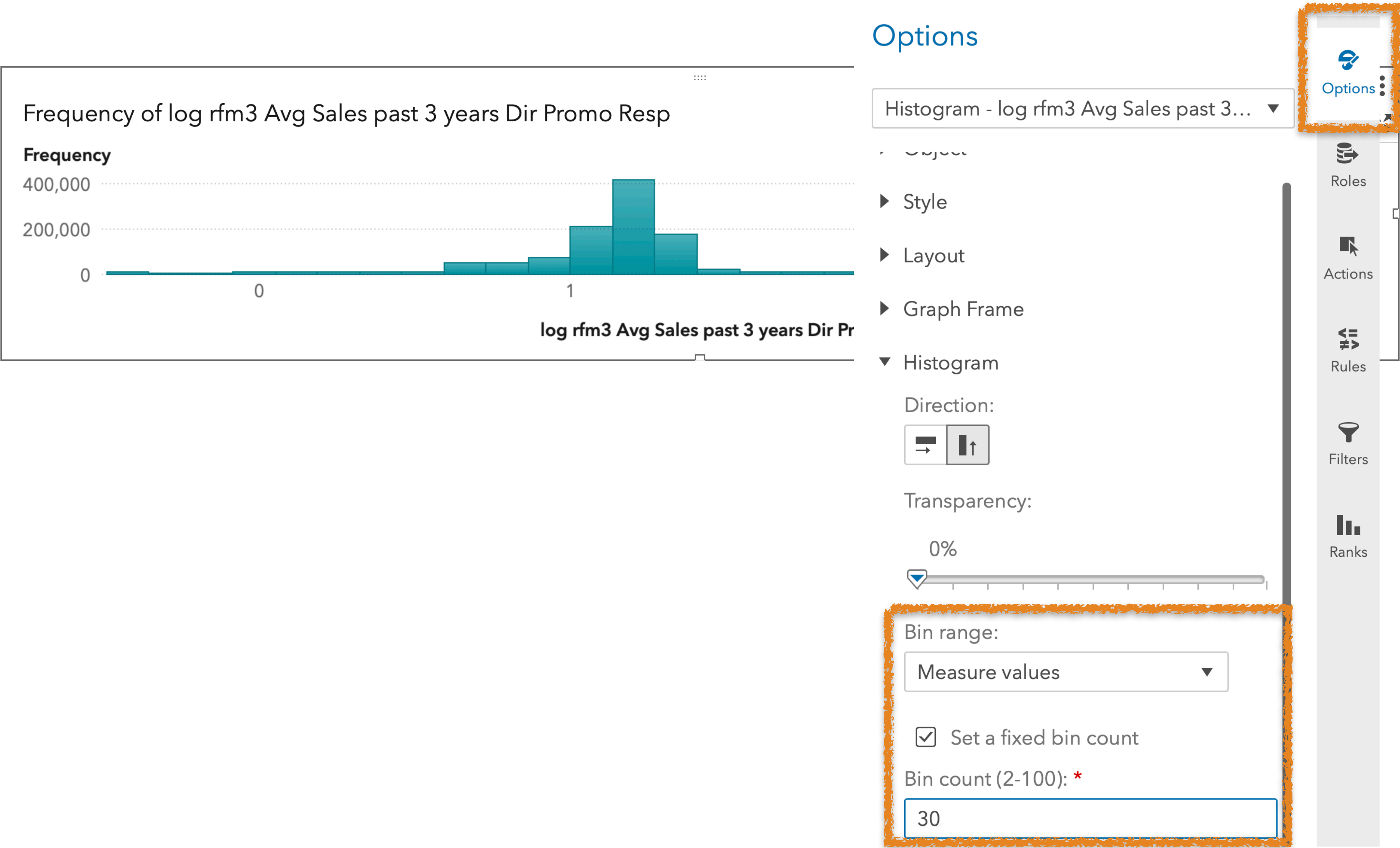
Side-by-side Comparison

Drag created variable to the right of the previous histogram. Make sure the spot is highlighted for placement.

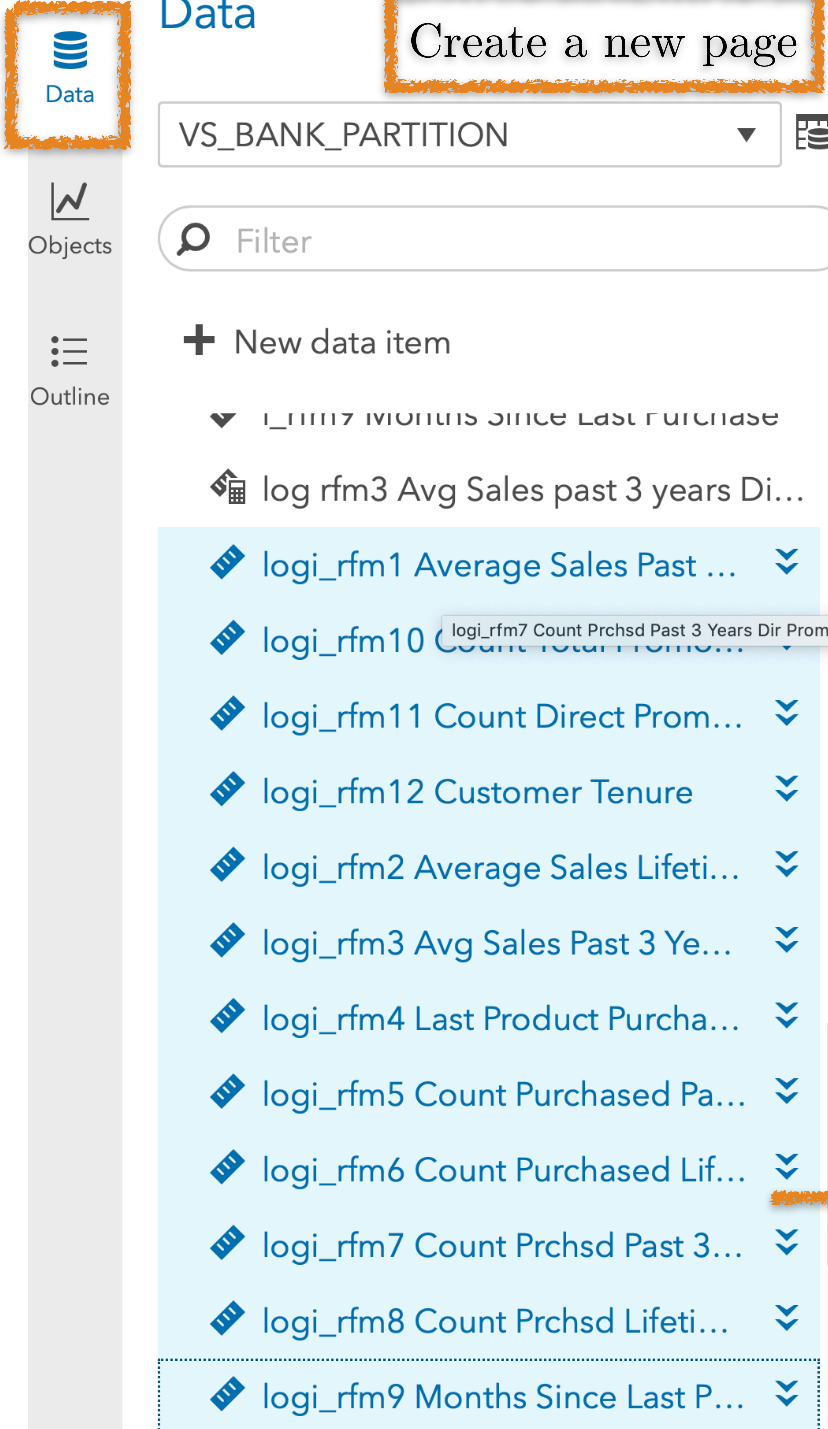




Log Transform

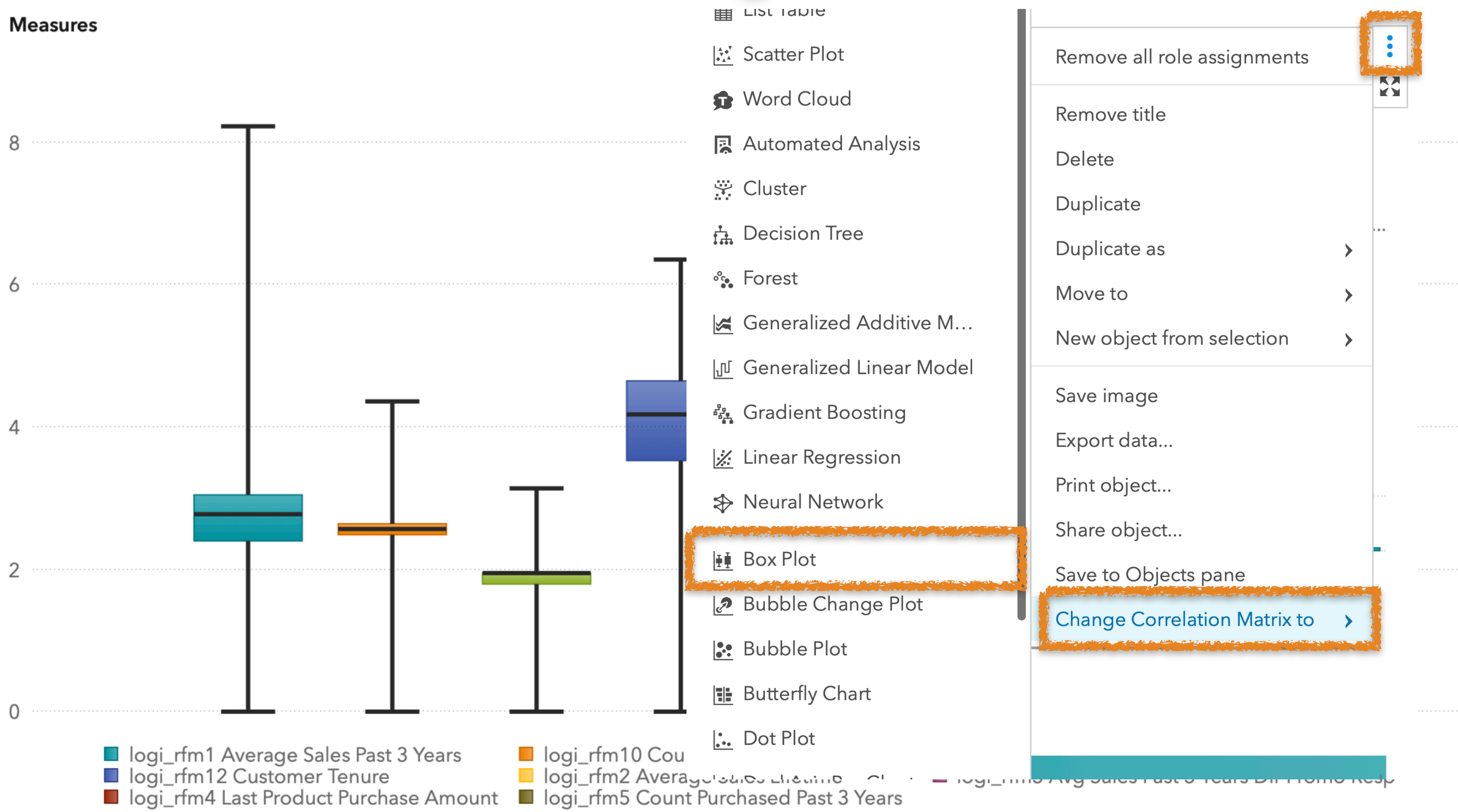


Visualize Multiple Relationships at once

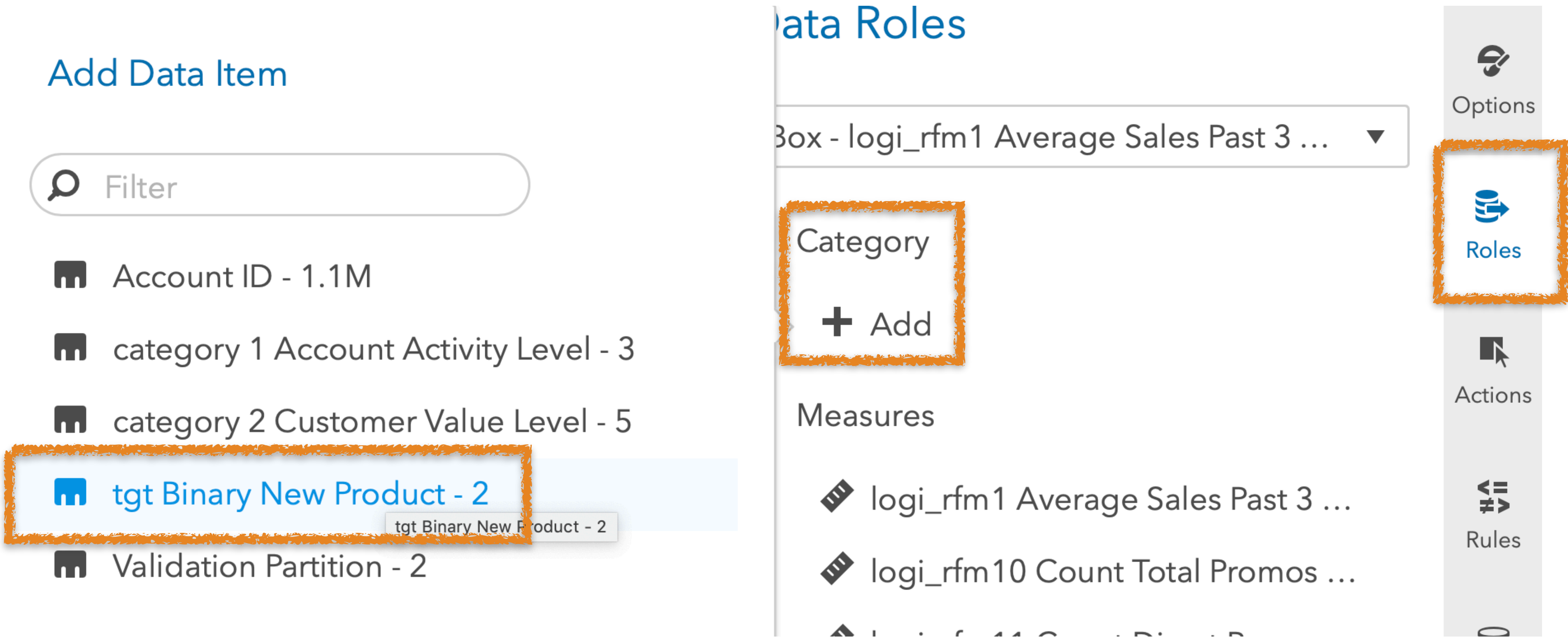


Drag all variables to chart.

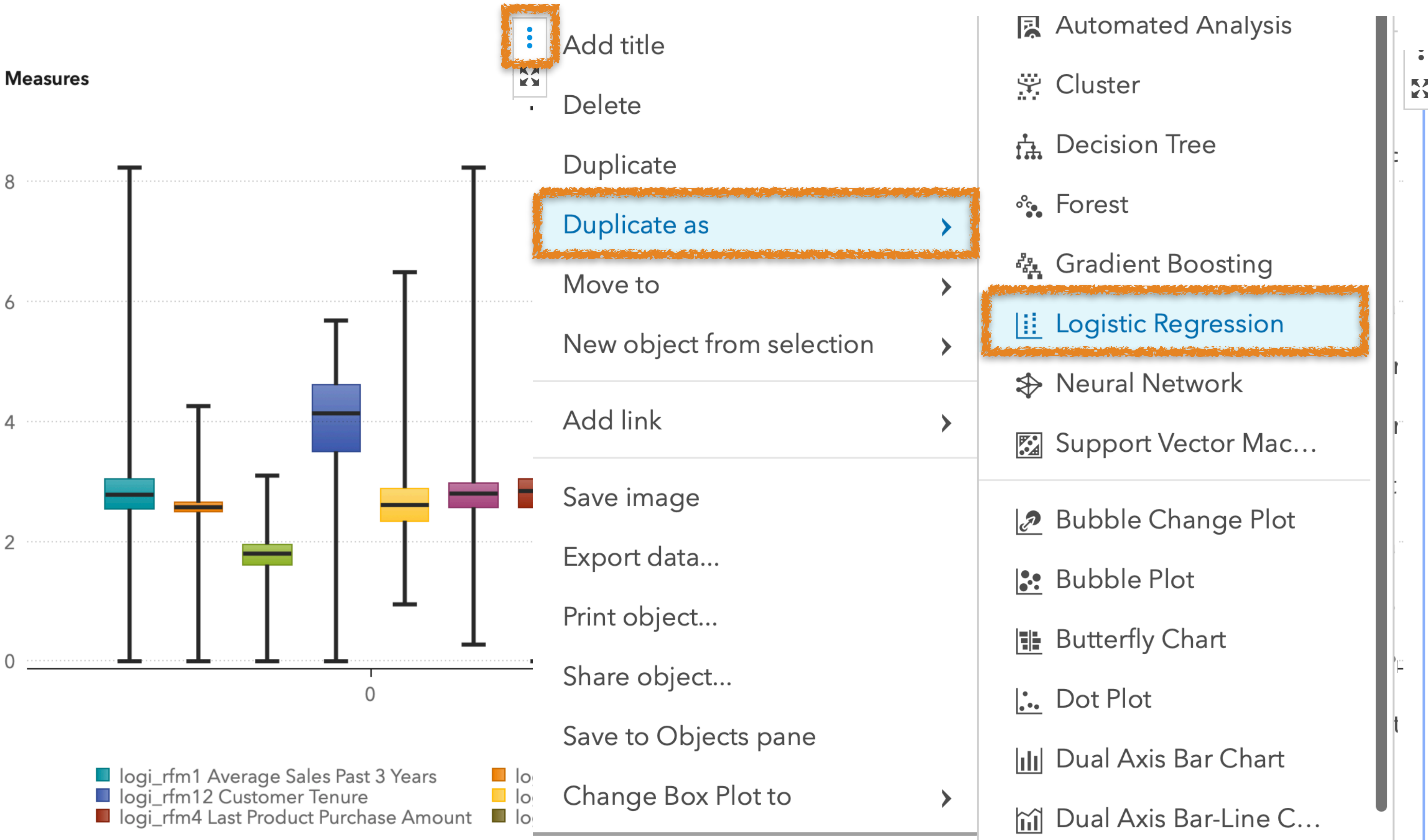
Visualize Multiple Relationships at once



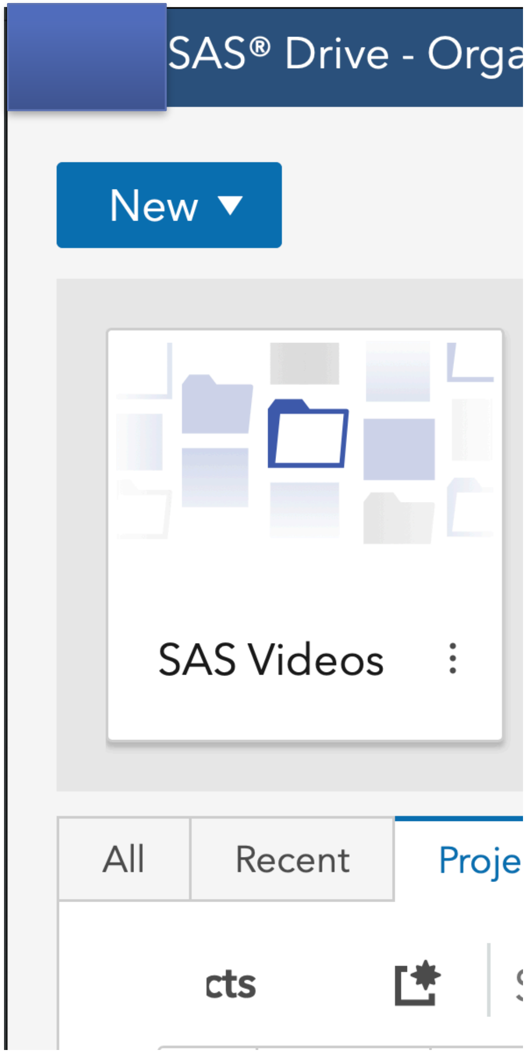
Group by Target Values

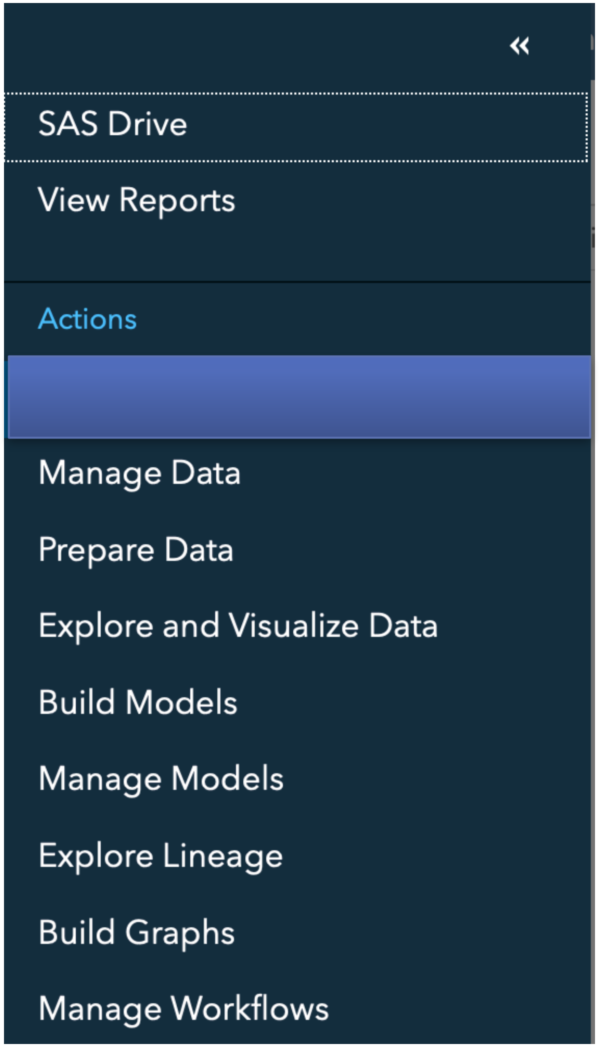


Explore with Logistic Reg.

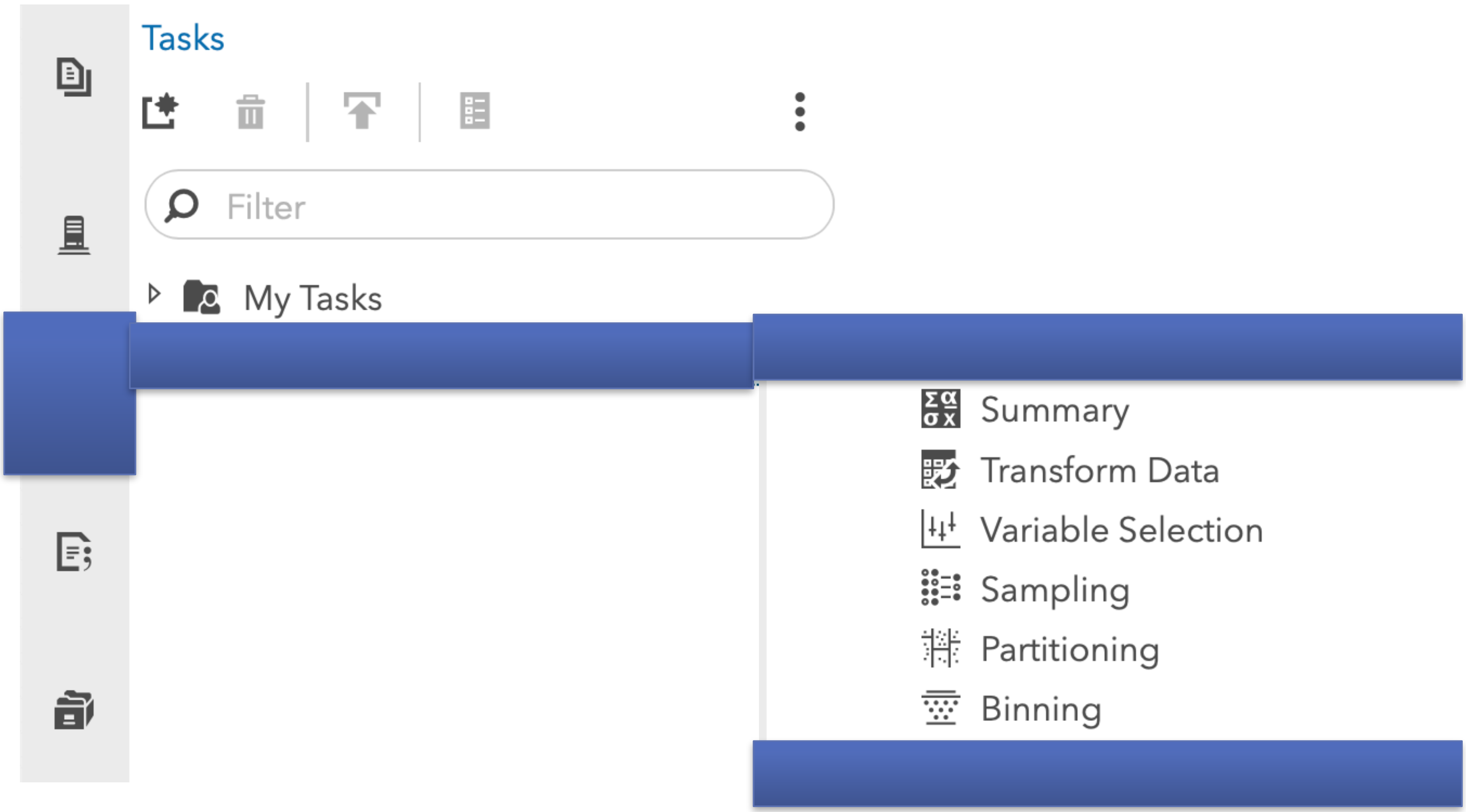


Imputation Task





Imputation Task



Imputation Task