

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

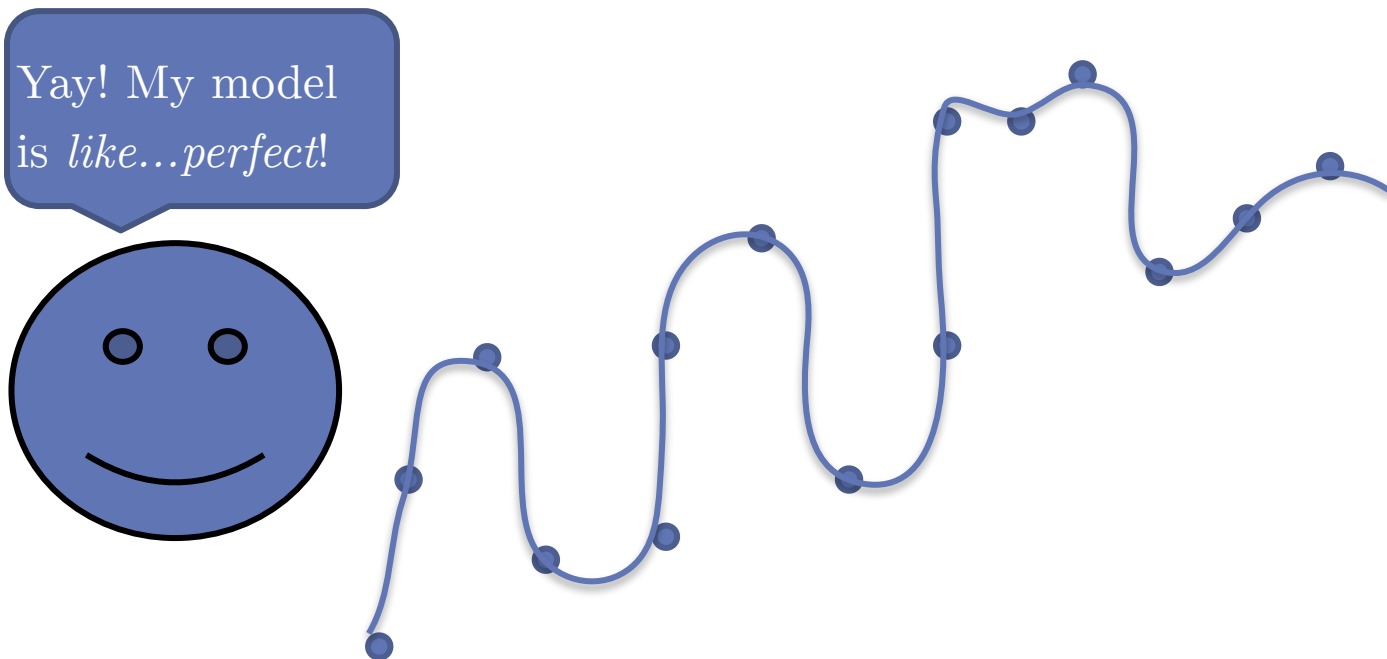
- When you **first** receive your data, **you'll explore** for distributions/outliers, and missing values.
- Before you look at any relationships between input variables and target variables, you should split into training, validation and test samples.
(Or decide on Cross-Validation / Testing)

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.

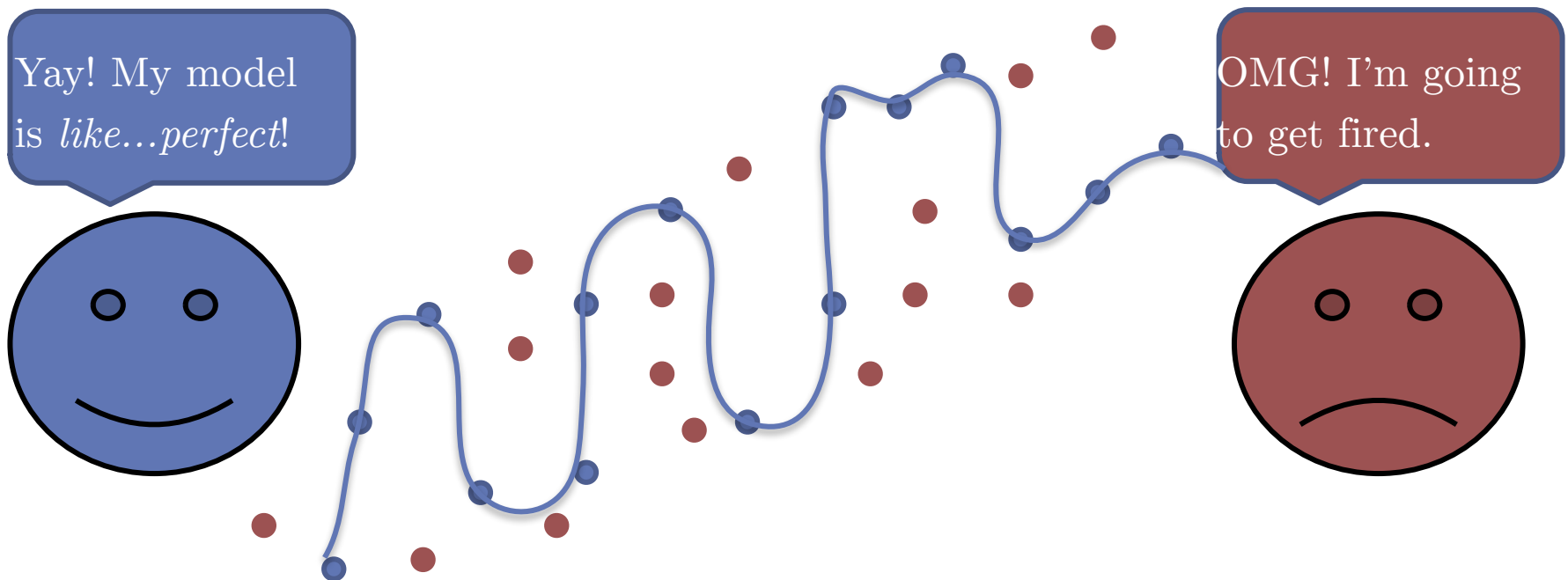
The Problem of Overfitting

- 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

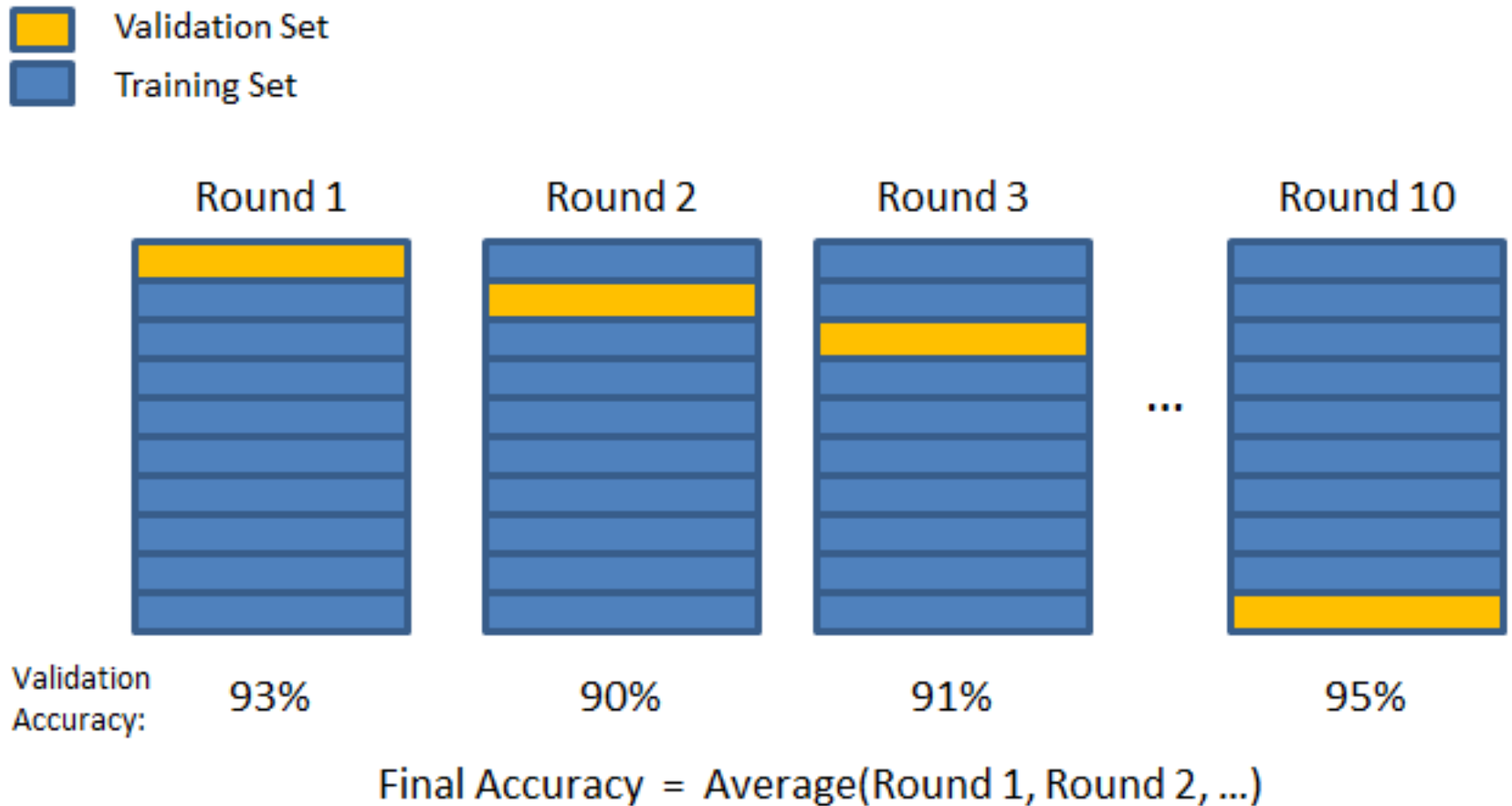
Training/Validation

- 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:



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 wide
- In this case, we’d have 3 observations (customers)
- One big *group by* SQL query

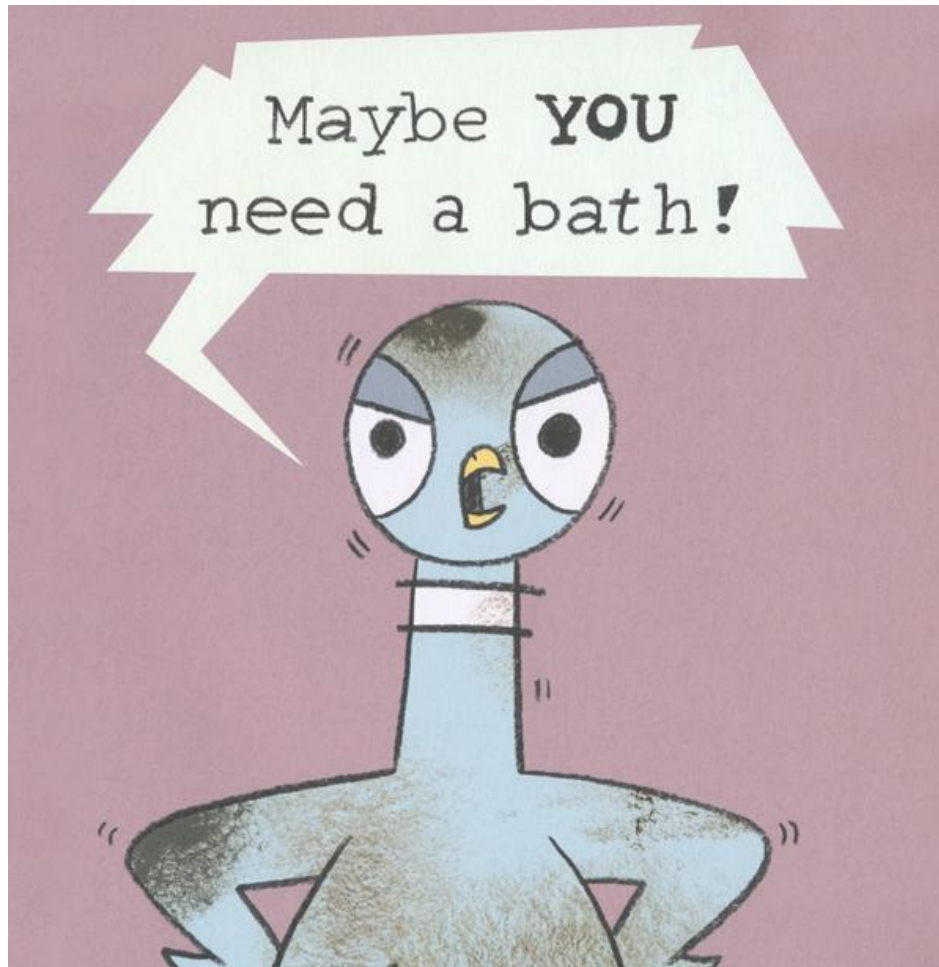
Transactional Data

A subset of columns we might consider in the process:

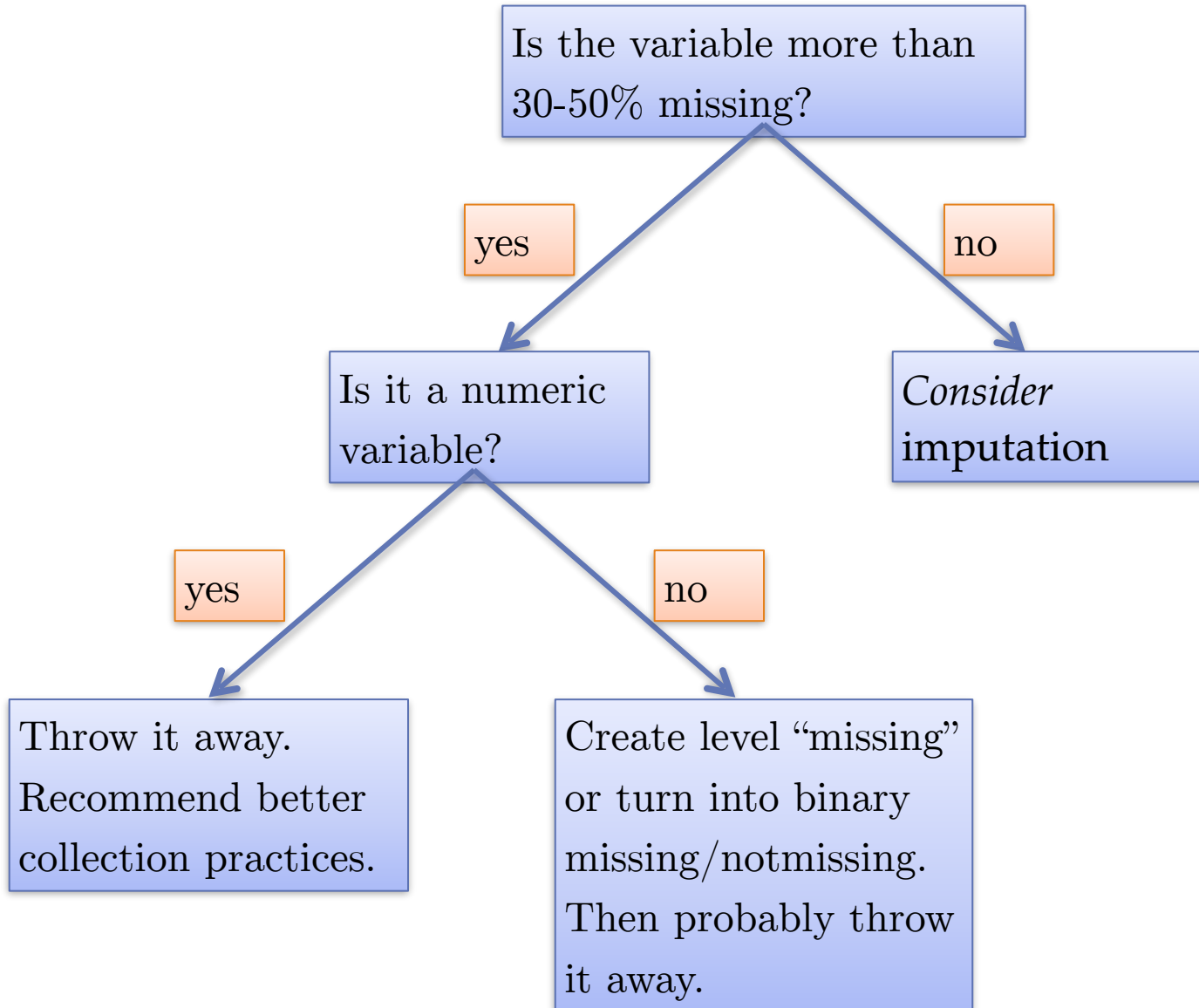
1. ID
2. Date of first transaction
3. Date of last transaction
4. Total number of transactions
5. Average time between transactions
6. Maximum number of items purchased
7. Average number of items purchased
8. Minimum number of items purchased
9. Std Deviation of number of items purchased
10. Maximum 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 SURVEYIMPUTE method = hotdeck

Numeric Variables

- Option one: Replace missing values with the **mean**
- Option two: Replace missing values with the **median** (for skewed distributions).
- Option three: **Predict** the missing values using other attributes.
 - Multiple Regression or Regression Trees popular
- Option four: **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

• • •

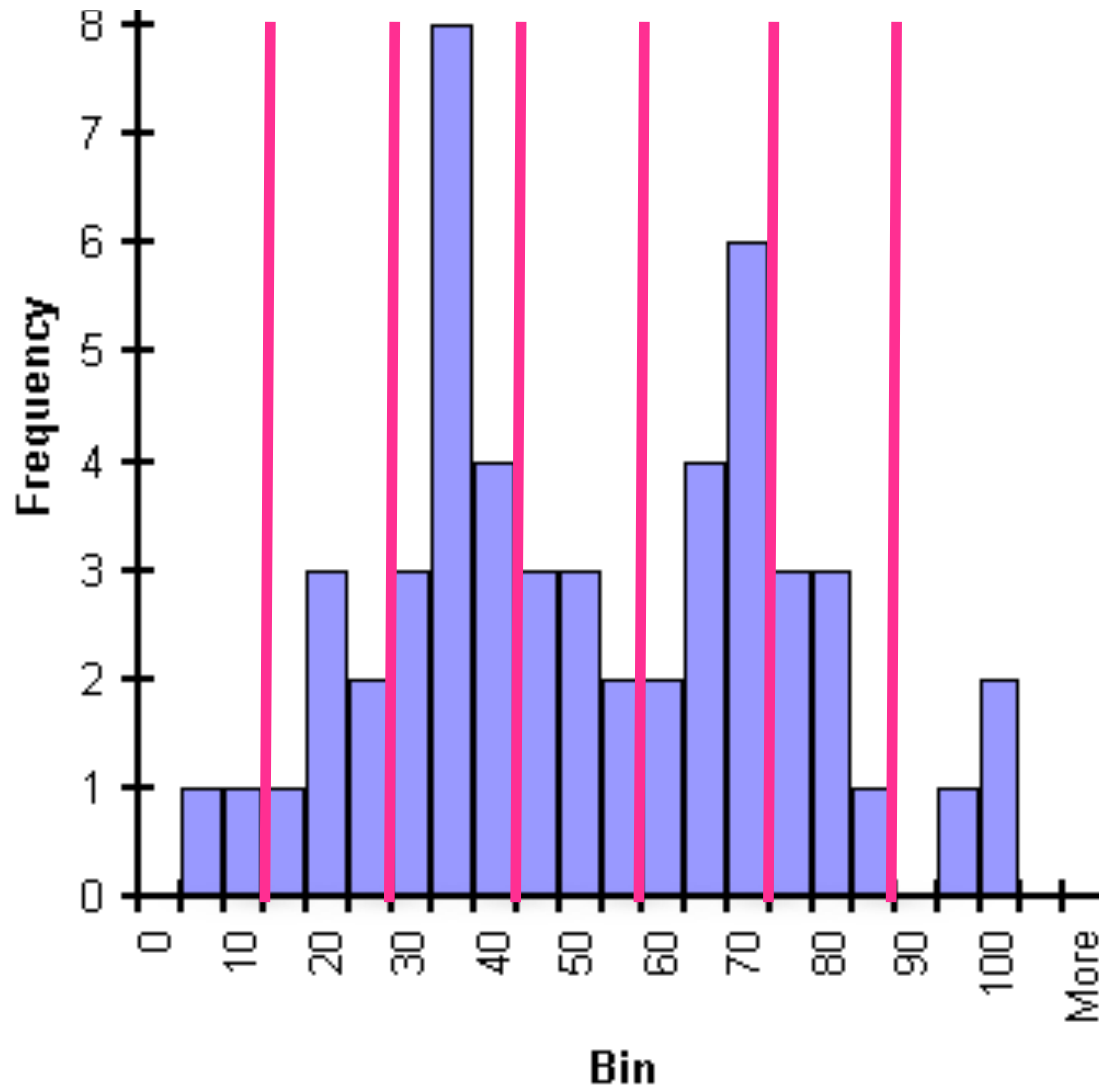
Variable Transformations

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	

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

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

- 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 $\pm 20\%$

Example: Logarithm on Dependent Variable

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

- 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 $\pm 20\%$

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

r	log(1+r)
-50%	-0.693
-40%	-0.511
-30%	-0.357
-20%	-0.223
-10%	-0.105
0%	-0.051
10%	-0.020
20%	0.000
30%	0.020
40%	0.049
50%	0.095
60%	0.182
70%	0.262
80%	0.336
90%	0.405
100%	0.693

Additive change in y
for coefficient of 1.

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

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

Log Transformation and Percent Change

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

% change in x

r	log(1+r)
-50%	-0.693
-40%	-0.511
-30%	-0.357
-20%	-0.223
-10%	-0.105
0%	-0.051
10%	-0.020
20%	0.000
30%	0.020
40%	0.049
50%	0.095
60%	0.182
70%	0.262
80%	0.336
90%	0.405
100%	0.693

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

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

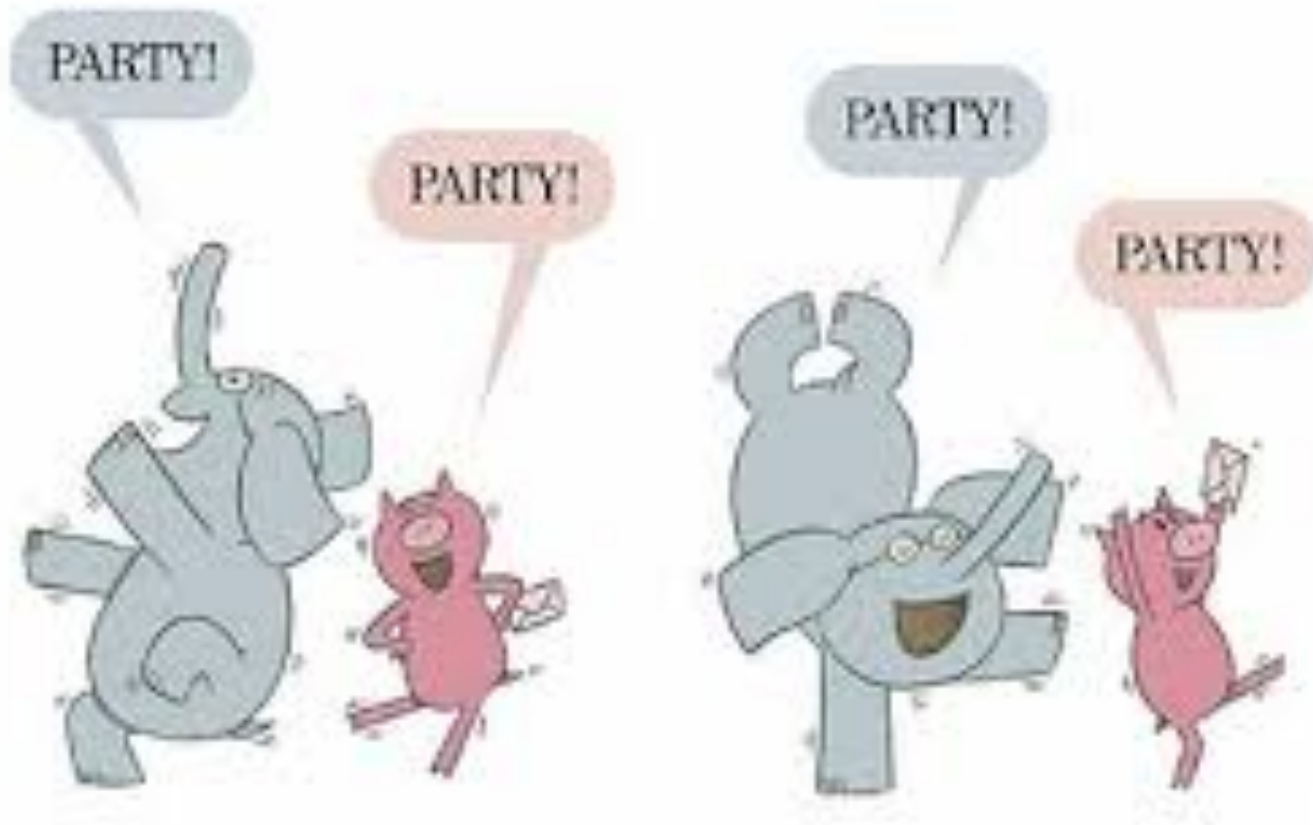
$$\log\left(\frac{y'}{y}\right) = \alpha \log(1 + r) \approx \alpha r$$

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

$$r_y \approx \alpha r$$

where r_y is percent increase in y .

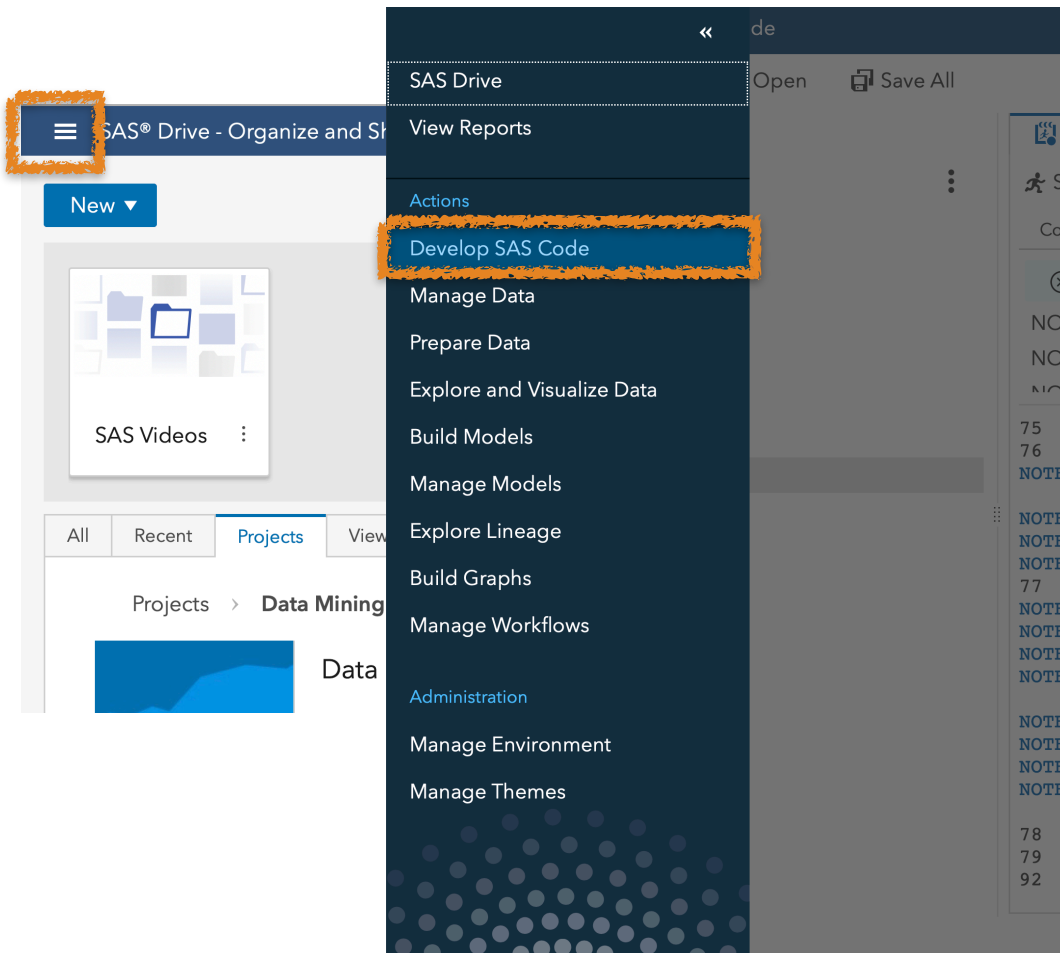
So what *should* I *DO*?



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

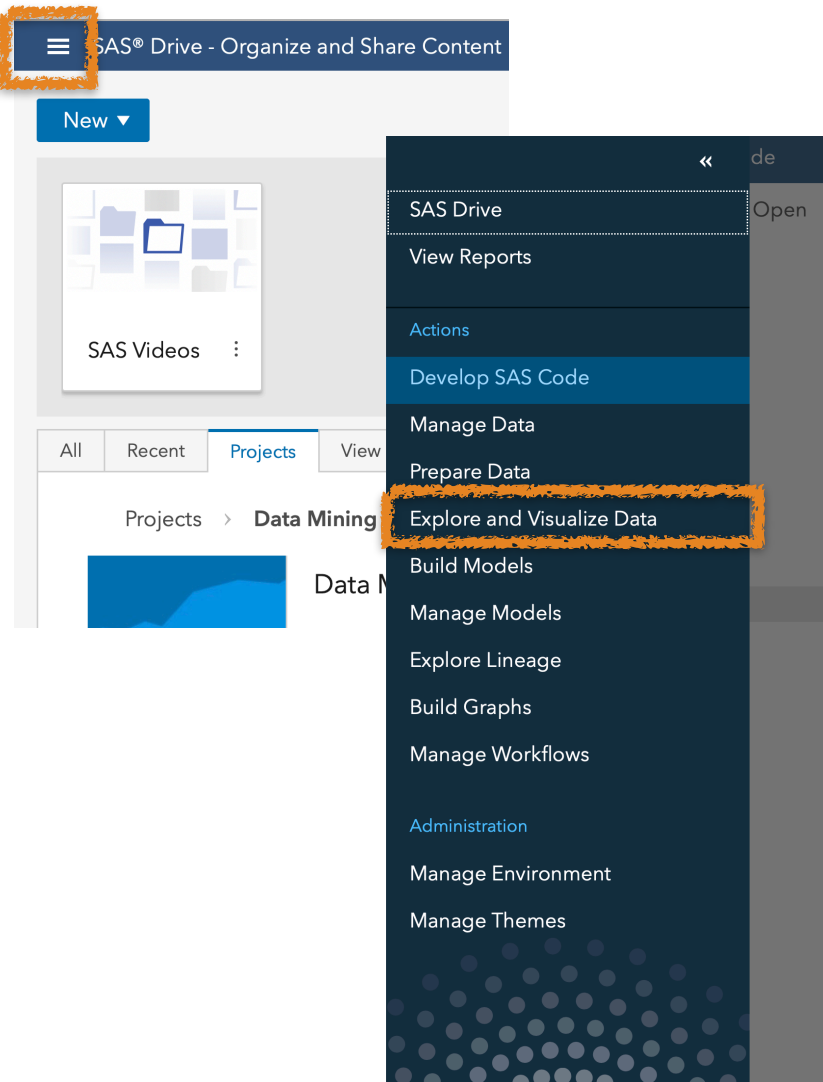


Submit Code:

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

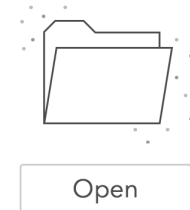
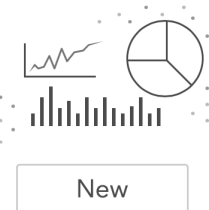
**You will repeat this step
EVERY time you use Viya
to load the Public library!**

SAS Viya Introduction



Welcome to SAS Visual Analytics

Select an option to get started:



☐ Make this selection the default

Dataset: VS_Bank_Partition

Data

VS_BANK_PARTITION

Filter

+ New data item

- rfm6 Count Purchased Lifetime
- rfm7 Count Prchsd Past 3 Years Di...
- rfm8 Count Prchsd Lifetime Dir Pr...
- rfm9 Months Since Last Purchase
- tgt Binary New Product**

Name: tgt Binary New Product

Classification: Measure

Format: Numeric (BEST12.)

Aggregation: Default (Sum)

Explore Target Variable

Change to Category

Scroll up and drag variable to chart.

Change Chart to %

Data Roles

Bar - tgt Binary New Product 1

▼ Category

tgt Binary New Product

▼ Measure

Frequency

+ Add

▼ Group

+ Add

▼ Lattice columns



Options



Roles



Actions

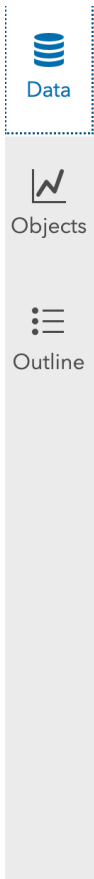
Replace Data Item

fre



Frequency Percent

Explore/Transform



Data

:

VS_BANK_PARTITION



Filter

+ New data item

Group ID Number

i_rfm1 Average Sales Past 3 Years

i_rfm10 Count Total Promos Past Y...

i_rfm11 Count Direct Promos Past ...

i_rfm12 Customer Tenure

i_rfm2 Average Sales Lifetime

i_rfm3 Avg Sales Past 3 Years ...



Frequency of i_rfm1 Average Sales Past 3 Years

Frequency

800,000

600,000

400,000

200,000

0

1000

Drag variable
to chart.

Log Transform

Data

VS_BANK_PARTITION

Filter

+ New data item

- Hierarchy...
- Custom category...
- Calculated item...
- Geography item...
- Parameter...
- Interaction effect...
- Spline effect...
- Partition...

Name:

Calculated Item 1

Data Items Operators

Search

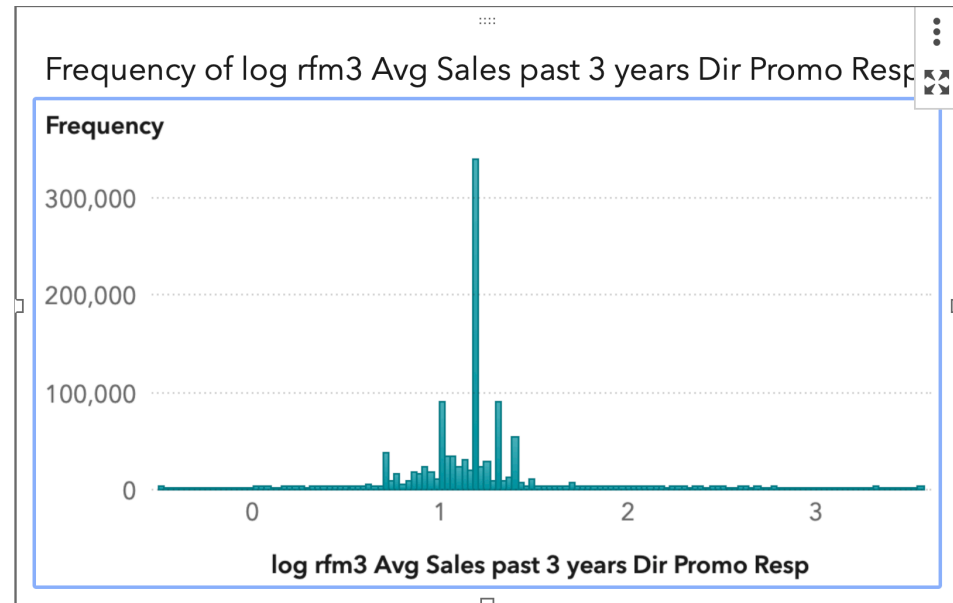
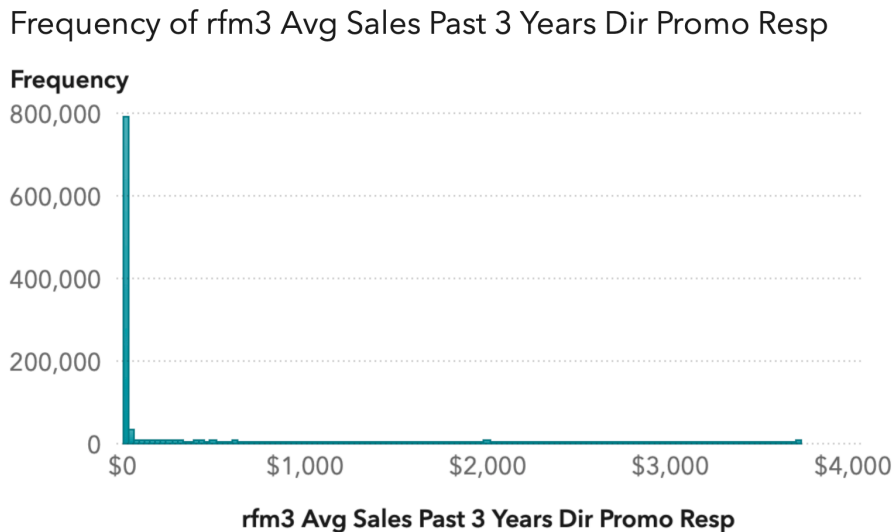
- ▶ Numeric (simple)
- ▶ Comparison
- ▶ Boolean
- ▶ Numeric (advanced)

i_rfm3 Avg Sales
Past 3 Years Dir
Promo Resp

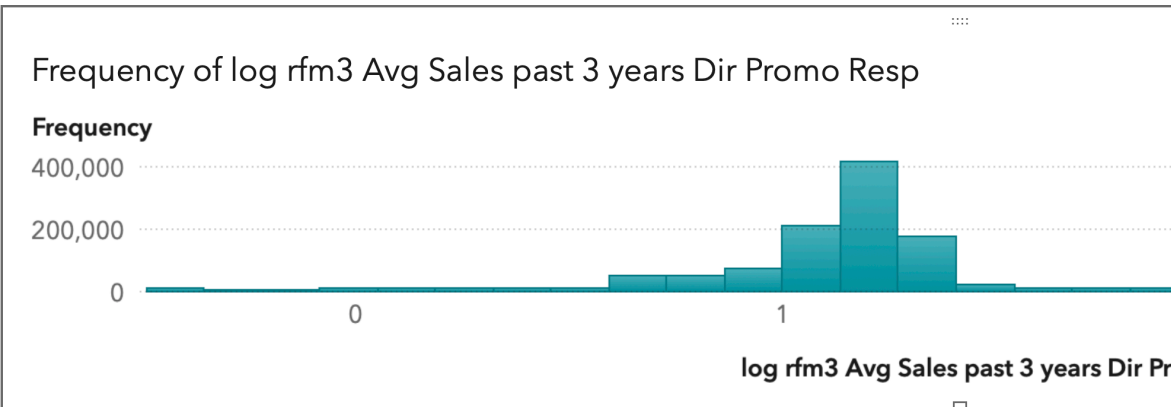
Log 10

Side-by-side Comparison

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



Log Transform



Options

Histogram - log rfm3 Avg Sales past 3... ▼

Object

► Style

► Layout

► Graph Frame

▼ Histogram

Direction:



Transparency:

0%



Bin range:

Measure values ▼

☒ Set a fixed bin count

Bin count (2-100): *

30

Options



Roles



Actions



Rules



Filters



Ranks

Data

Create a new page

VS_BANK_PARTITION

Filter

+ New data item

- ▼ 11117 MONTHS SINCE LAST PURCHASE
- log_rfm3 Avg Sales past 3 years Di...
- logi_rfm1 Average Sales Past ...
- logi_rfm10 Count Total Prom...
- logi_rfm11 Count Direct Prom...
- logi_rfm12 Customer Tenure
- logi_rfm2 Average Sales Lifeti...
- logi_rfm3 Avg Sales Past 3 Ye...
- logi_rfm4 Last Product Purcha...
- logi_rfm5 Count Purchased Pa...
- logi_rfm6 Count Purchased Lif...
- logi_rfm7 Count Prchsd Past 3...
- logi_rfm8 Count Prchsd Lifeti...
- logi_rfm9 Months Since Last P...

Visualize Multiple Relationships at once

Drag all variables
to chart.

Visualize Multiple Relationships at once

Measures

8

6

4

2

0

logi_rfm1 Average Sales Past 3 Years

logi_rfm12 Customer Tenure

logi_rfm4 Last Product Purchase Amount

logi_rfm10 Cou

logi_rfm2 Average Sales Past 3 Years

logi_rfm5 Count Purchased Past 3 Years

List Table

Scatter Plot

Word Cloud

Automated Analysis

Cluster

Decision Tree

Forest

Generalized Additive M...

Generalized Linear Model

Gradient Boosting

Linear Regression

Neural Network

Box Plot

Bubble Change Plot

Bubble Plot

Butterfly Chart

Dot Plot

Remove all role assignments

Remove title

Delete

Duplicate

Duplicate as

Move to

New object from selection

Save image

Export data...

Print object...

Share object...






Save to Objects pane

Change Correlation Matrix to

Group by Target Values

Add Data Item

 Filter

-  Account ID - 1.1M
-  category 1 Account Activity Level - 3
-  category 2 Customer Value Level - 5
-  **tgt Binary New Product - 2**
-  Validation Partition - 2

Data Roles

Box - logi_rfm1 Average Sales Past 3 ... ▼

Category

+ Add

Measures

 logi_rfm1 Average Sales Past 3 ...

 logi_rfm10 Count Total Promos ...



Options



Roles

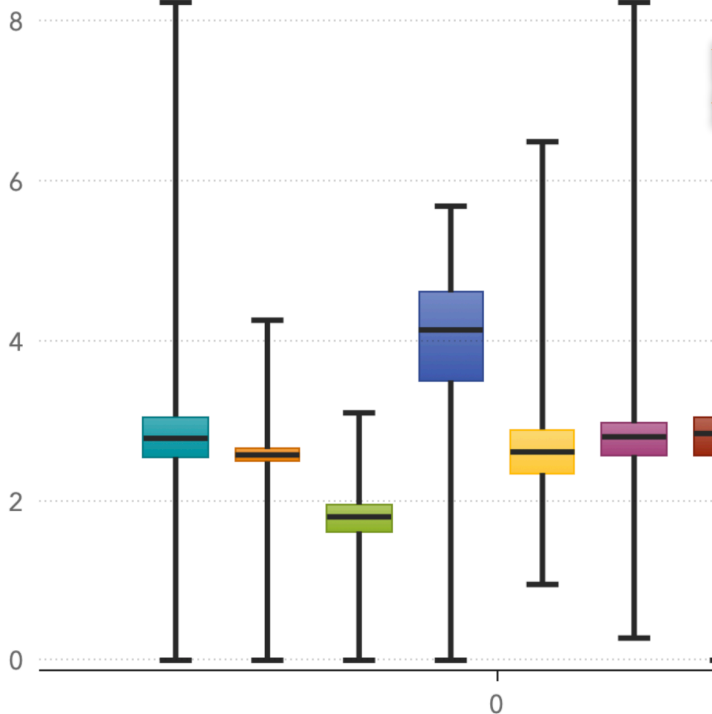


Actions



Rules

Explore with Logistic Reg.



Add title



Delete

Duplicate

Duplicate as



Move to



New object from selection



Add link



Save image

Export data...

Print object...

Share object...

Save to Objects pane

Change Box Plot to



 Automated Analysis

 Cluster Decision Tree

 Forest

Gradient Boosting

Logistic Regression

- Neural Network

 Support Vector Mac...

Bubble Change Plot

Bubble Plot

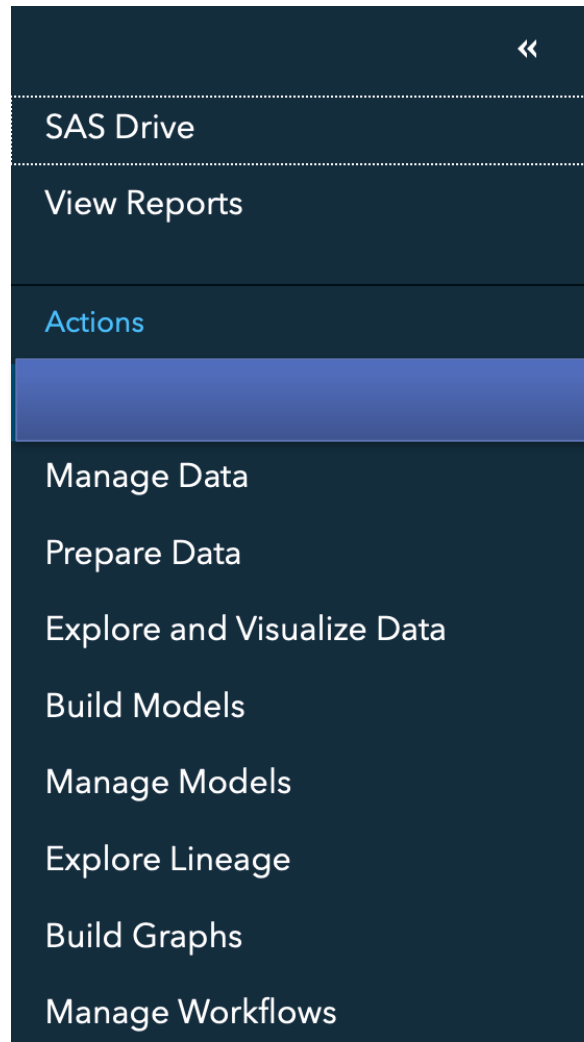
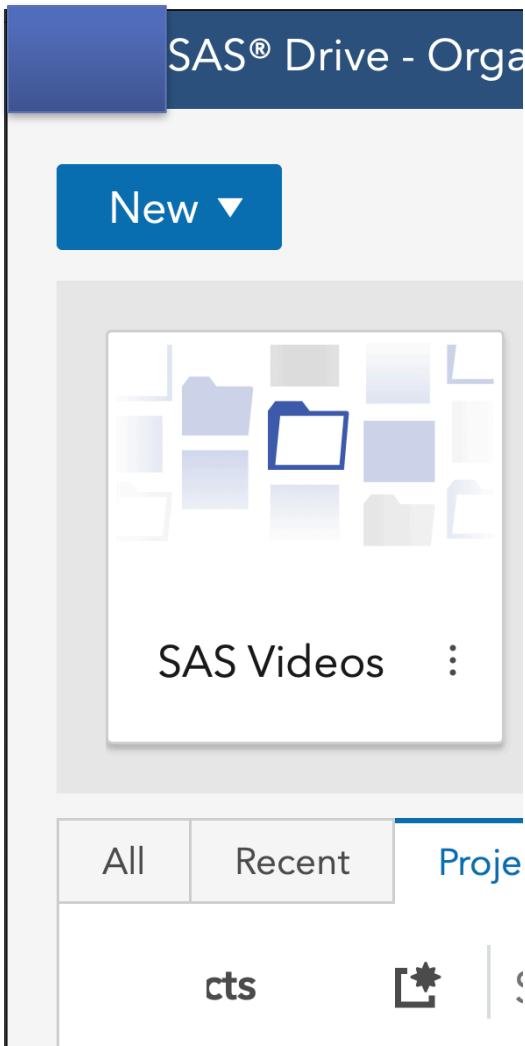
Butterfly Chart

Dot Plot

||| Dual Axis Bar Chart

Dual Axis Bar-Line C...

Imputation Task



Imputation Task

Tasks



Filter



My Tasks



Summary



Transform Data



Variable Selection



Sampling



Partitioning



Binning

Imputation Task

▼ DATA

PUBLIC.VS_BANK_PARTITION

Filter: (...)

▼ ROLES

▼ Interval Variables

Replace missing values with the mean:

☐ ☒ ri_demog_homeval

DATA OUTPUT INFORMATION

▼ OUTPUT DATA

The following table must use a CAS engine libref:

☒ Save imputed data

Specify a CAS table: *

☐ Overwrite data

casuser.test

Include variables from the input CAS table:

☒ All variables

☐ Variables used in the analysis

☐ No variables

Code Log

```
1 /*
2 *
3 * Task code generated by SAS® Studio 5.
4 *
5 * Generated on '9/29/19, 2:40 PM'
6 * Generated by 'slrace'
7 * Generated on server 'sasviyal'
8 * Generated on SAS platform 'Linux LIN
9 * Generated on SAS version 'V.03.04M0P0
10 * Generated on browser 'Mozilla/5.0 (Ma
11 * Generated on web client 'https://sasv
12 */
13
14 ods noproctitle;
15
16 proc varimpute data=PUBLIC.VS_BANK_PARTI
```