VENTANA systems, inc.

Vensim® Software

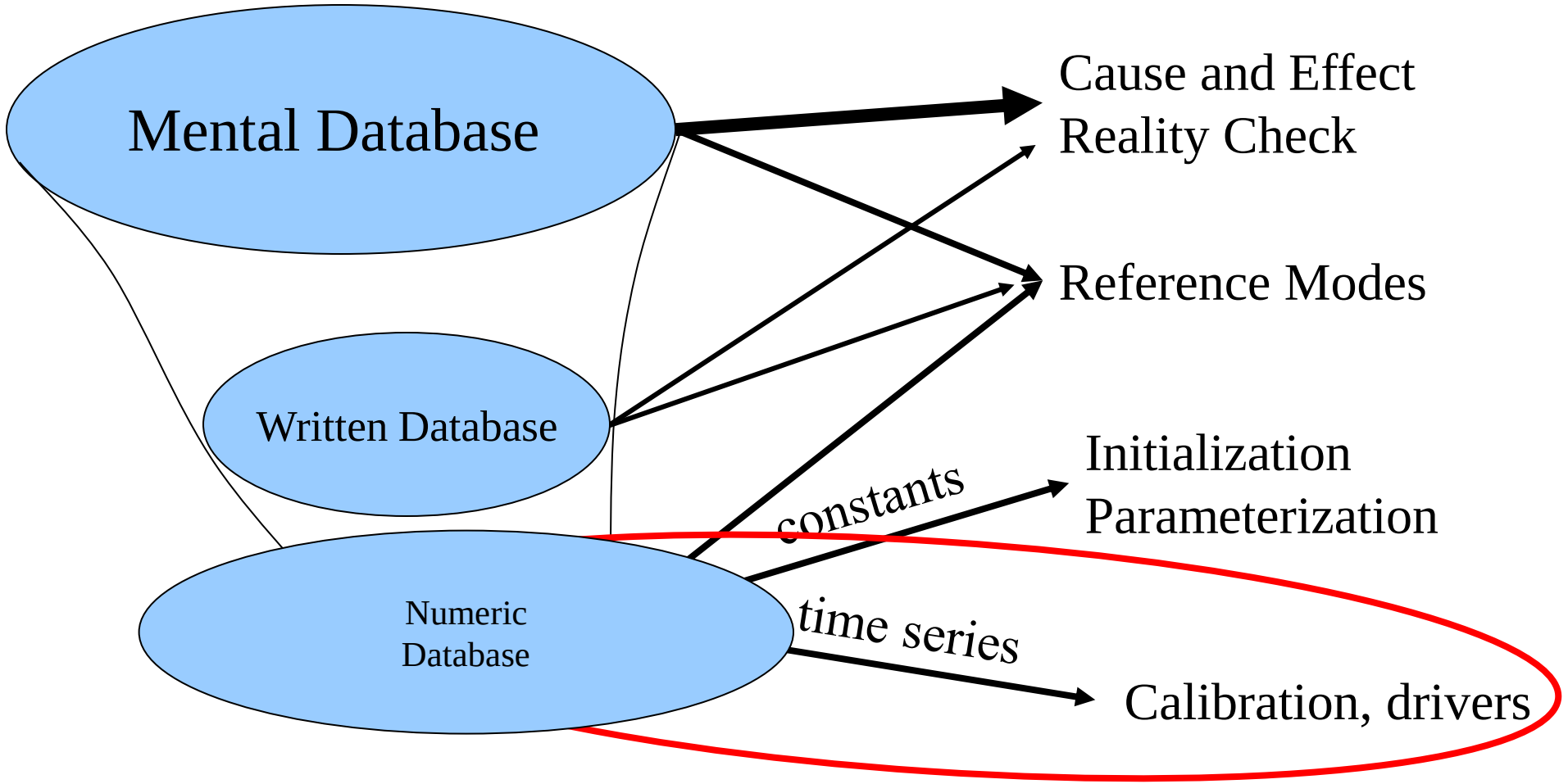
Linking systems thinking to powerful dynamic models

Advanced Vensim
Data

Tom Fiddaman

2022

Context – Types of Data



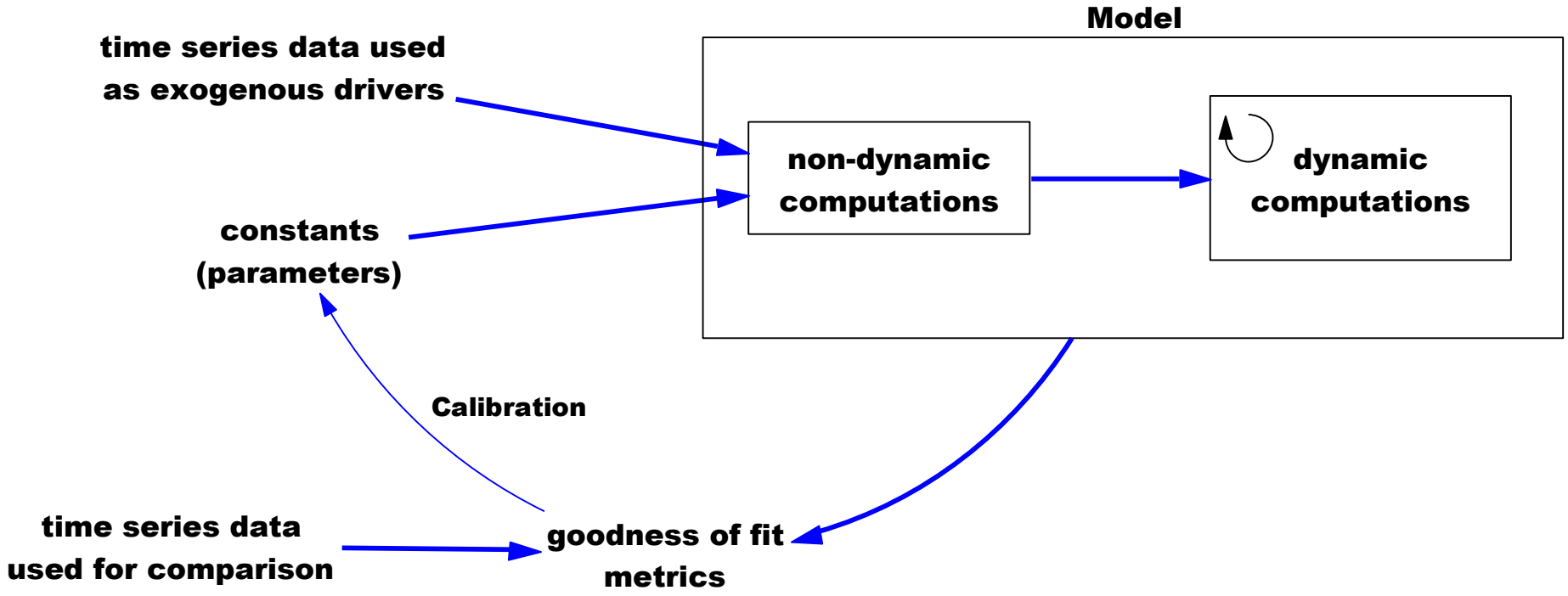
Motivation

- **Time series data defines behavior**

- **We can use that behavior to …**
  - Frame the problem to be addressed
  - Look for patterns and relationships
  - Compare the model with what happened
    - Qualitatively
    - Quantitatively
  - Drive the model with outside influences
    - For example, interest rates in a housing model
  - Convince people with compelling stories

- **Data availability does not bound a problem!**

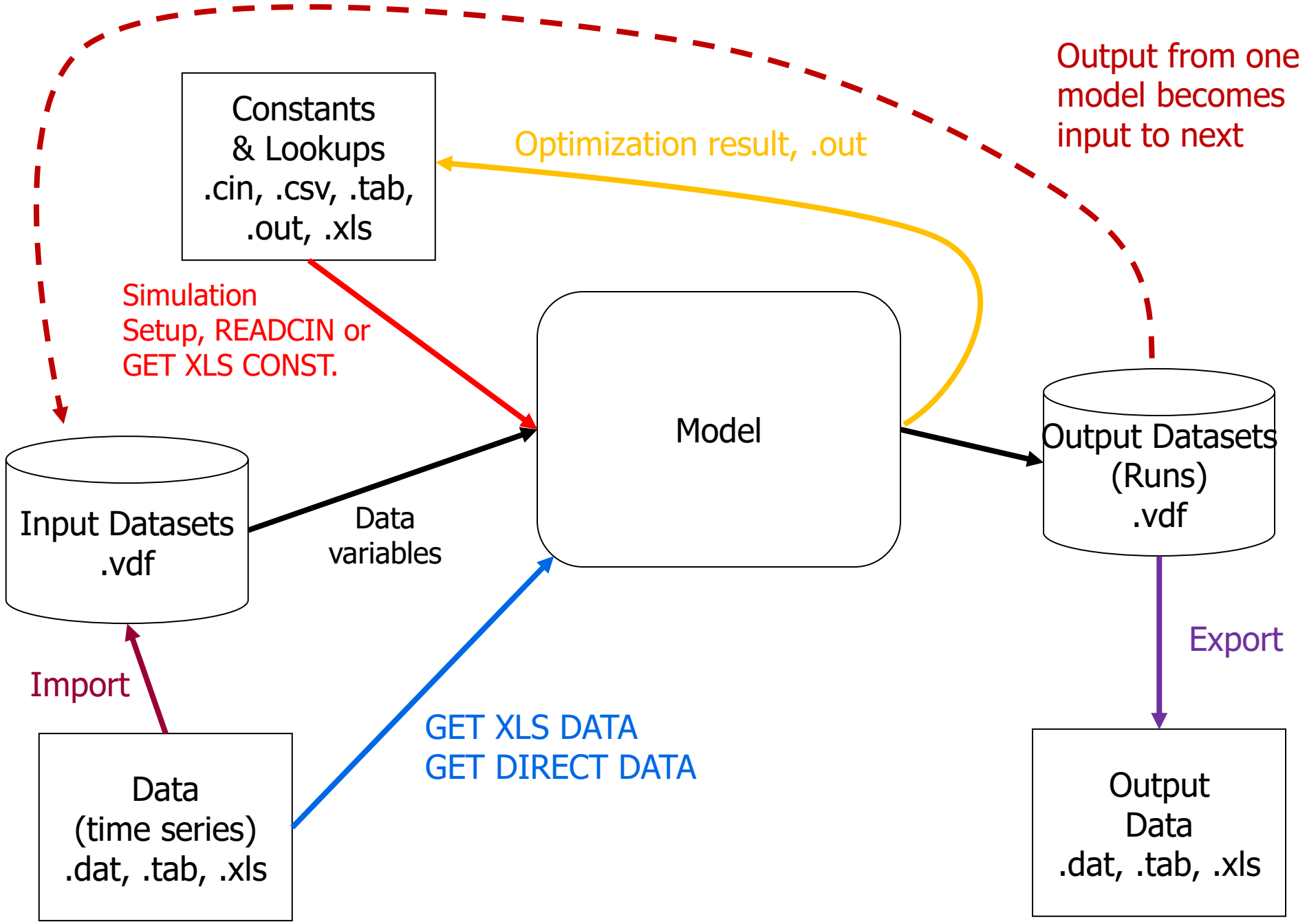
Perspective



- Classic SD models have no drivers
- Spreadsheet, statistical and econometric models often lack dynamic computation
- Comparison is not always performed

Getting Data In

1. **Lookups (table functions) - common but not usually recommended**
   - Not good for calibration
   - More work to enter data
   - Unnecessary model structure
2. **Reading from .vdf files**
   - Importing from files (text or spreadsheets, DAT2VDF, XLS2VDF, etc.)
   - Output from one model may be used as input to another
   - GET VDF DATA
3. **Connecting to spreadsheets & text files**
   - Excel .xls & .xlsx: GET XLS
   - Text files, or .xls & .xlsx without Excel: GET DIRECT
4. **Connecting to databases: ODBC**

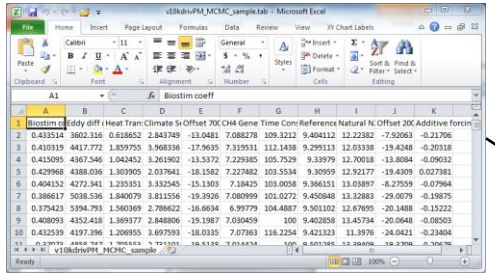


File Types

Extension	Description	
.vdfx	Vensim binary dataset (since v8, double precision)	
.vdf	Vensim binary dataset (older versions – still works)	
.dat	Vensim's native text format for data	
.tab, .csv, .txt	Delimited text files that can be imported to .vdfx or called via GET DIRECT	Can have various layouts – time across, time down, relational, Tidy data
.xls, .xlsx	Excel, can be imported to .vdfx or called via GET XLS or GET DIRECT	
.vdi, .vdo	Control files for ODBC input and output	

Organization – Simple File Import

.dat, .tab, .csv, .xlsx



Import via Model>Import Dataset

Data
.vdf file

Model

Run
.vdf file

Data Variable Naming

When you use file import to .vdf, you have a choice:

- **Give data the same name as model variables**
  - Useful for comparison graphs and tables – if the data is loaded from a dataset, model and data automatically show up on the same axes
  - Can't use for driving data, due to name conflict

- **Give data different names**
  - Ex: "price of cheese" and "data price of cheese"
  - Necessary for driving the model from data
  - Makes comparisons slightly less convenient, because plots, calibration payoffs, etc. must specify two variable names

Importing Data (Import\import0)

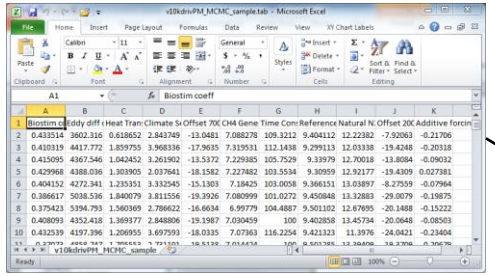
- **Simplest input format is .dat**
  - Can also use tab delimited and spreadsheet formats
  - This format is unique to Vensim
- **Import to a .vdf file**
  - Format the same as simulations
  - Can use same Analysis Tools to examine
  - Easy to compare
- **Importing other text and spreadsheet formats is conceptually the same, but might involve more options**

Data Importing (Import\import1)

- **From .dat file**
  - Simple layout
    - Variable followed by time value pairs
  - No options for importing
- **From tab delimited file**
  - Dialog to control importing
- **From other models**
  - Just use matching variable names

Organization – Simple GET XLS

.xlsx



Import via GET XLS functions

Model

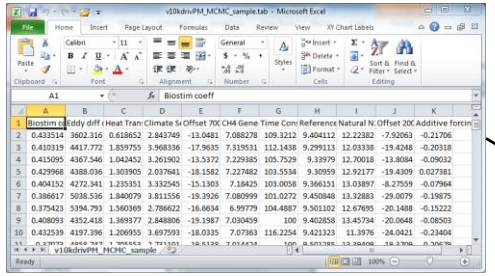
Run
.vdf file

Linking Data from Excel (Import\import2)

- **Via Excel Spreadsheet**
  - Works the same way within a model
  - Connects live to Excel and use Excel to connect to other things
- **Easiest for driving input**
  - More work for comparison

Organization – Simple GET DIRECT

.tab, .csv, .txt

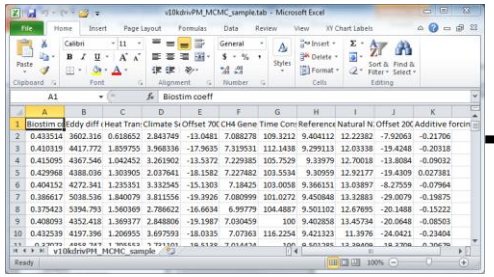


Import via GET DIRECT functions

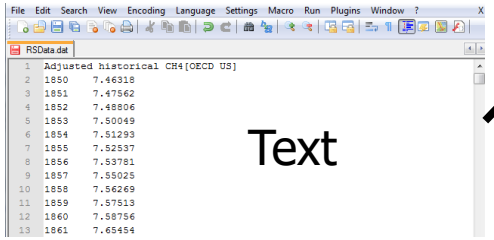
Model

Run
.vdf file

Organization – Complex Case, Many Sources



GET XLS



Text

Single Source .vdf files

Data Model

Consolidate, clean, normalize units, aggregate

All Data .vdf file

Dynamic Model

Run .vdf file

Advantages:

- All the data ends up in one place
- Lower computational overhead in the dynamic model
- Fewer files to distribute
- Division of labor

Interpolation Patterns (Interpolation\connect.mdl)

- **:Interpolate:**
  - Straight line between points
- **:Hold Backward:**
  - Hold a value until you get the next
- **:Look Forward:**
  - Look ahead to the next available value
- **:Raw:**
  - Special number :NA: except when data

Data Equations
dataeq.mdl

- **Provides manipulation of data in the model**
- **Assignment is := rather than the usual = (this is handled by changing the type in the equation editor)**
- **Generally the same as model equations, but …**
  - Constraints
    - Only data and constants on right
    - Not changed for optimization
  - Intersection of available points
    - If A is available at times 1,2,3 and B is available at times 2,3,4, a data equation C := A*B will be defined at times 2,3 only
- **Can shift these variables in time**
- **Equations all computed before the model is simulated**

Data Equation Examples

Time	0	1	2	3
Data X	2		8	12
Data Y	4	8		10
Auxiliary Z=X+Y	6	13	17	22
Data DZ := X+Y	6			22
Auxiliary V1=Z	6	13	17	22
Auxiliary V2=DZ	6	11.33	16.67	22

Data Equations – Special Functions

Time	0	1	2	3	4	5
X	4		1	3		
Y	7	6		5		
X IF MISSING(X+Y,0)	11	6	1	8		
TIME SHIFT(X,2)			4		1	3
CUMULATE(X)	4		5	8		
CUMULATEF(X)				8		

Special Functions /Functions folder

- **Get statistics of data:**
  - GET DATA FIRST TIME, GET DATA MEAN, etc.
  - Usually, set variable type to Initial to wrap with INITIAL() (avoids slowdown)
- **Get single point values**
  - GET DATA AT TIME, GET DATA BETWEEN TIMES
  - Usually, wrap with INITIAL() to avoid slowdown
- **Manipulate data**
  - CUMULATE
  - TIME SHIFT
- **See Help system, Reference Manual, Functions chapter for examples**