Supercharge Your User Interfaces in JSL

JMP Discovery 2018 Conference October, 2018

Peter Mroz
Statistical Programming
Janssen Research & Development

Justin Chilton
JMP Development Testing
SAS

Martin Freeman, *Untitled*
Diagnosed with AIDS in 1990,
Martin lives in San Francisco where
he continues to create new pieces.

Agenda

- Introduction
- Col Boxes
- Tabs
- Too Many Tabs
- Associative Arrays
- Tree Nodes and Tree Boxes
- Filtering Long Picklists

Introduction

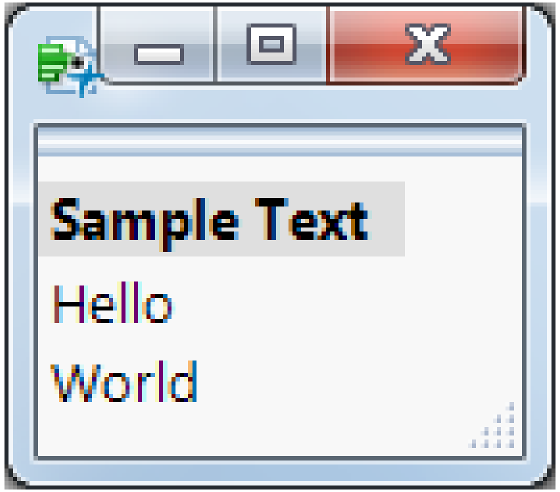
- Application user interfaces should be
  - Easy to use
  - Easy to understand
  - Transparent to the user
- Good user interfaces result in
  - Engaged users
  - Fewer frustrations
  - Great user experiences
- This talk:
  - Variety of ways to supercharge your JMP user interfaces

Col Boxes

- Special type of column object that can contain any other display box
- Contained inside a Table Box
- Allows you to display
  - Text in different fonts, styles, sizes, foreground/background colors in a Table Box grid
  - A column of clickable buttons
  - A column of icons representing the status of a row
  - A column of mini-graphs
  - A column of pictures

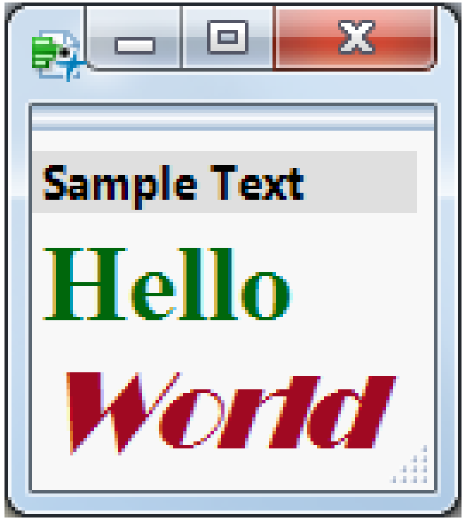
Simple Col Box Example (1)

```
// Simple Col Box Example.jsl
nw = new window("Col Box Example",
    tb = table box(
        cb = col box("Sample Text",
            tb1 = text box("Hello"),
            tb2 = text box("World")
        )
    )
);
```


Sample Text
Hello
World

Simple Col Box Example (2)

```
// Format the text boxes
tb1 << font color("Dark Green")
    << set font("Times New Roman", 24, "bold");
tb2 << font color("Dark Red")
    << set font("Broadway", 24, "italic");
```

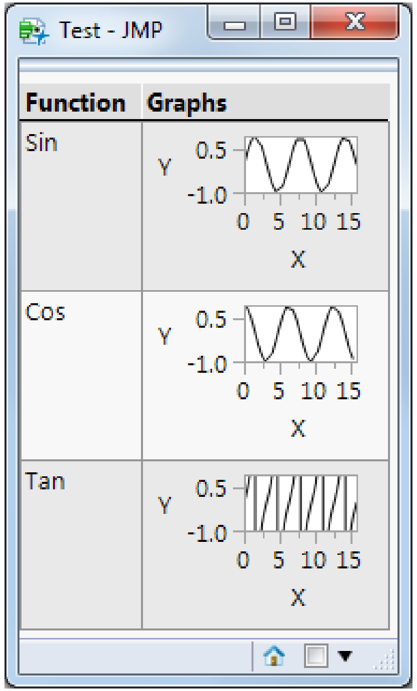


Sample Text

Hello

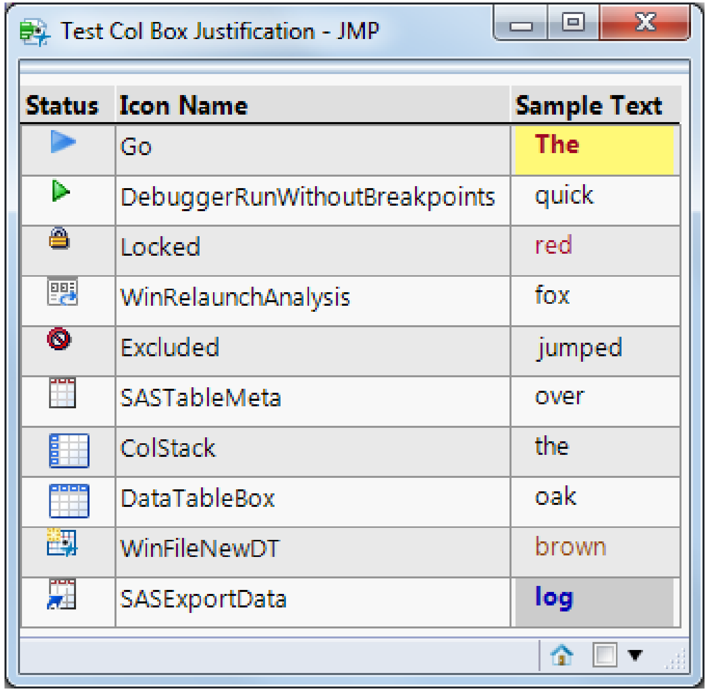
World

Col Box Examples



Function	Graphs
Sin	
Cos	
Tan	

Col Box Graph Function Example.jsl

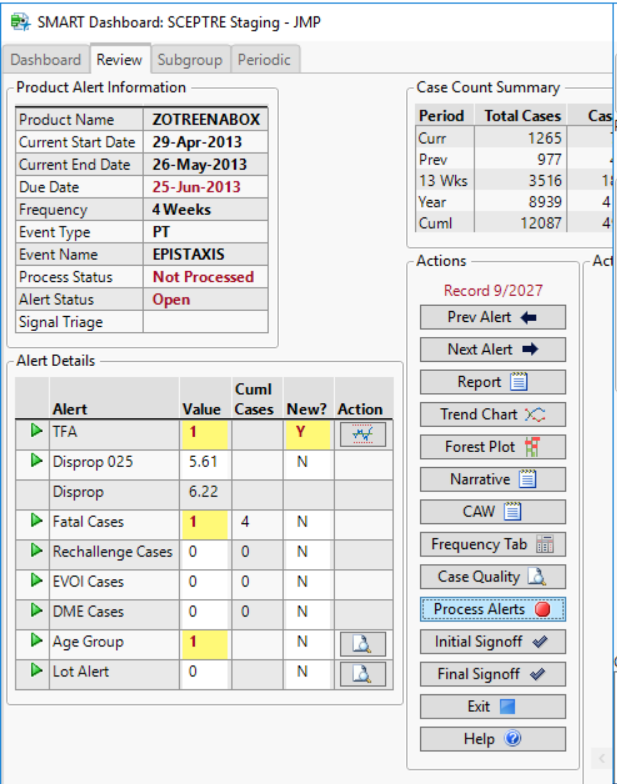


Status	Icon Name	Sample Text
▶	Go	**The**
▷	DebuggerRunWithoutBreakpoints	quick
	Locked	red
	WinRelaunchAnalysis	fox
⊘	Excluded	jumped
	SASTableMeta	over
	ColStack	the
	DataTableBox	oak
	WinFileNewDT	brown
	SASExportData	**log**

Col Box icons formatted text Example.jsl

Real World Example – Adverse Event App

- Review adverse events that have alerted



Product Alert Information

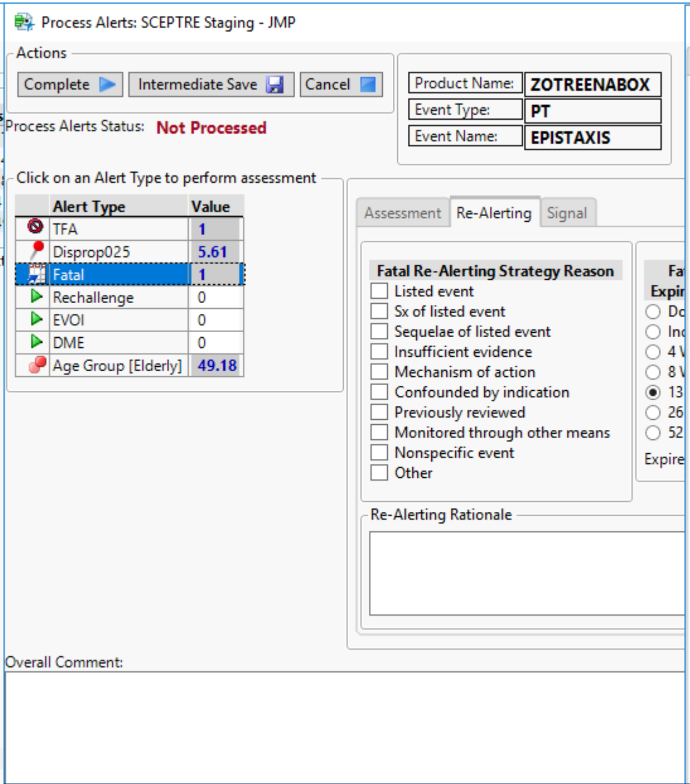
Product Name	ZOTREENABOX
Current Start Date	29-Apr-2013
Current End Date	26-May-2013
Due Date	25-Jun-2013
Frequency	4 Weeks
Event Type	PT
Event Name	EPISTAXIS
Process Status	Not Processed
Alert Status	Open
Signal Triage	

Case Count Summary

Period	Total Cases	Cas
Curr	1265	
Prev	977	
13 Wks	3516	1
Year	8939	4
Cuml	12087	4

Alert Details

Alert	Value	Cuml Cases	New?	Action
TFA	1		Y	
Disprop 025	5.61		N	
Disprop	6.22			
Fatal Cases	1	4	N	
Rechallenge Cases	0	0	N	
EVOI Cases	0	0	N	
DME Cases	0	0	N	
Age Group	1		N	
Lot Alert	0		N	



Product Name:	ZOTREENABOX
Event Type:	PT
Event Name:	EPISTAXIS

Process Alerts Status: Not Processed

Click on an Alert Type to perform assessment

Alert Type	Value
TFA	1
Disprop025	5.61
Fatal	1
Rechallenge	0
EVOI	0
DME	0
Age Group [Elderly]	49.18

Fatal Re-Alerting Strategy Reason

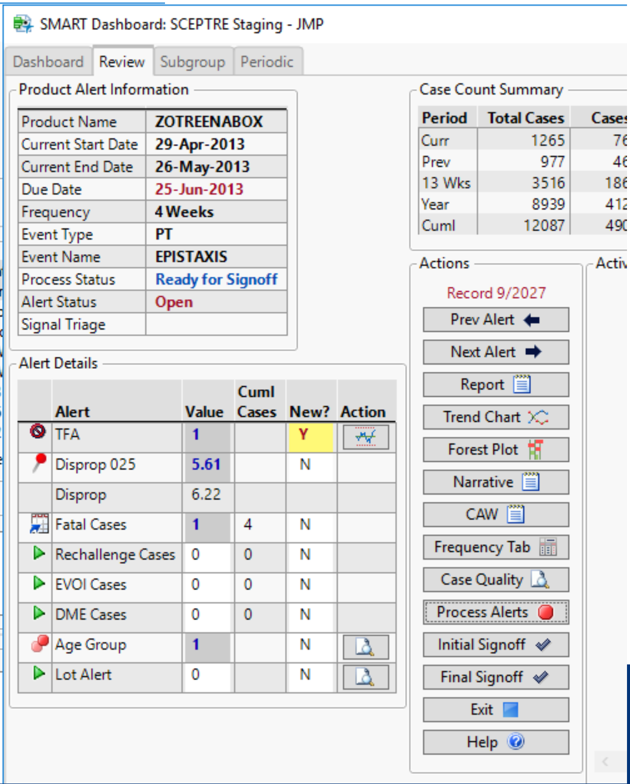
- ☐ Listed event
- ☐ Sx of listed event
- ☐ Sequelae of listed event
- ☐ Insufficient evidence
- ☐ Mechanism of action
- ☐ Confounded by indication
- ☐ Previously reviewed
- ☐ Monitored through other means
- ☐ Nonspecific event
- ☐ Other

Fa
Expir
- ○ Do
- ○ Inc
- ○ 4 W
- ○ 8 W
- ● 13
- ○ 26
- ○ 52

Expire

Re-Alerting Rationale

Overall Comment:



Product Alert Information

Product Name	ZOTREENABOX
Current Start Date	29-Apr-2013
Current End Date	26-May-2013
Due Date	25-Jun-2013
Frequency	4 Weeks
Event Type	PT
Event Name	EPISTAXIS
Process Status	Ready for Signoff
Alert Status	Open
Signal Triage	

Case Count Summary

Period	Total Cases	Cases
Curr	1265	76
Prev	977	46
13 Wks	3516	186
Year	8939	412
Cuml	12087	490

Alert Details

Alert	Value	Cuml Cases	New?	Action
TFA	1		Y	
Disprop 025	5.61		N	
Disprop	6.22			
Fatal Cases	1	4	N	
Rechallenge Cases	0	0	N	
EVOI Cases	0	0	N	
DME Cases	0	0	N	
Age Group	1		N	
Lot Alert	0		N	

Table Box Tip

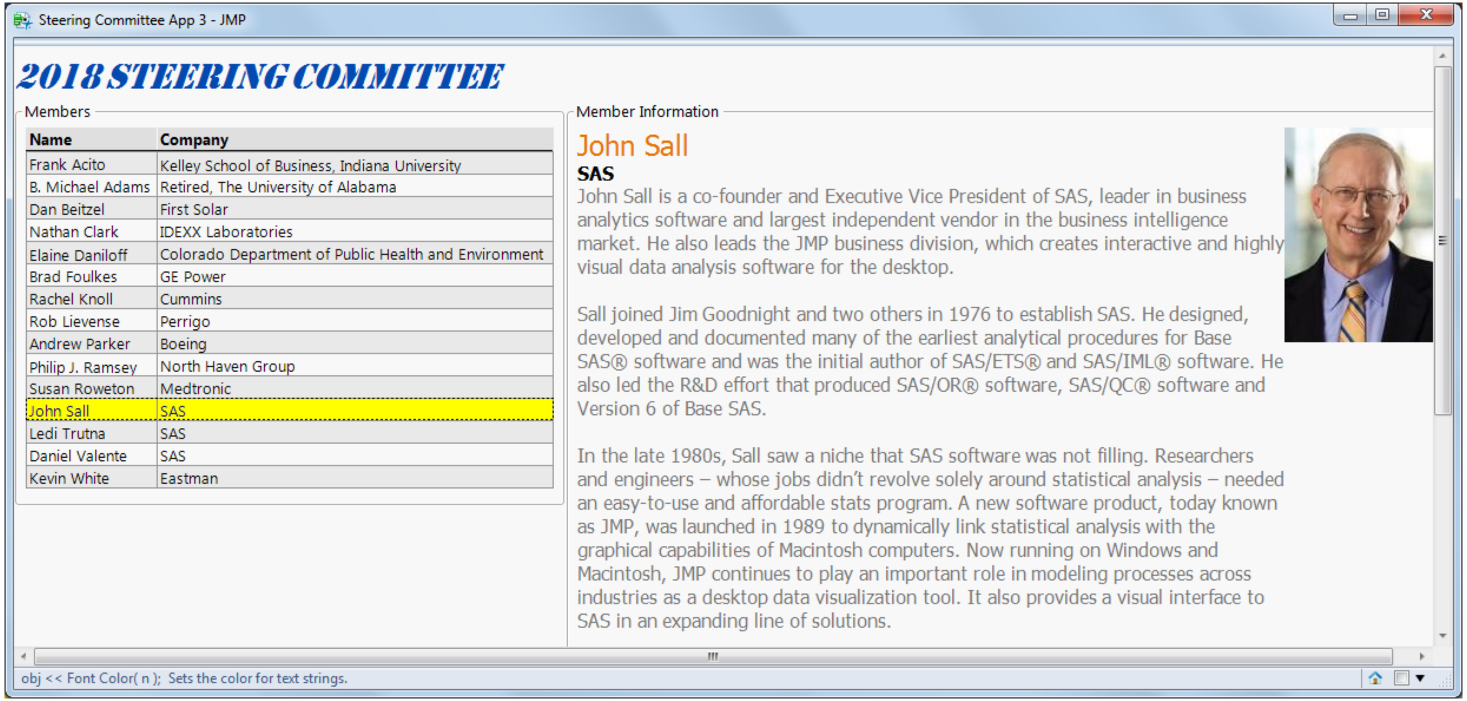
- Make rows clickable

```
    << set selectable rows(1);
```

- Add actions

```
    << set row change function()
```

Steering Committee Example



2018 STEERING COMMITTEE

Members

Name	Company
Frank Acito	Kelley School of Business, Indiana University
B. Michael Adams	Retired, The University of Alabama
Dan Beitzel	First Solar
Nathan Clark	IDEXX Laboratories
Elaine Daniloff	Colorado Department of Public Health and Environment
Brad Foulkes	GE Power
Rachel Knoll	Cummins
Rob Lievense	Perrigo
Andrew Parker	Boeing
Philip J. Ramsey	North Haven Group
Susan Roweton	Medtronic
John Sall	SAS
Ledi Trutna	SAS
Daniel Valente	SAS
Kevin White	Eastman

Member Information

John Sall

SAS

John Sall is a co-founder and Executive Vice President of SAS, leader in business analytics software and largest independent vendor in the business intelligence market. He also leads the JMP business division, which creates interactive and highly visual data analysis software for the desktop.

Sall joined Jim Goodnight and two others in 1976 to establish SAS. He designed, developed and documented many of the earliest analytical procedures for Base SAS® software and was the initial author of SAS/ETS® and SAS/IML® software. He also led the R&D effort that produced SAS/OR® software, SAS/QC® software and Version 6 of Base SAS.

In the late 1980s, Sall saw a niche that SAS software was not filling. Researchers and engineers – whose jobs didn't revolve solely around statistical analysis – needed an easy-to-use and affordable stats program. A new software product, today known as JMP, was launched in 1989 to dynamically link statistical analysis with the graphical capabilities of Macintosh computers. Now running on Windows and Macintosh, JMP continues to play an important role in modeling processes across industries as a desktop data visualization tool. It also provides a visual interface to SAS in an expanding line of solutions.

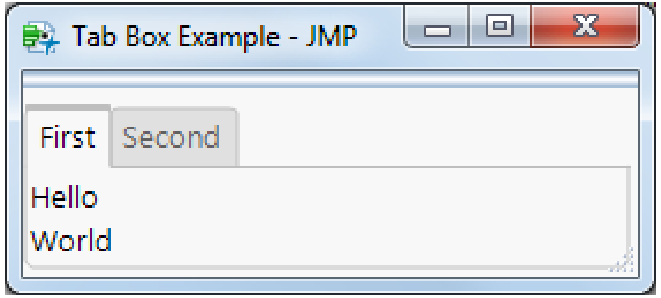
obj << Font Color(n); Sets the color for text strings.

Steering Committee.jsl + Steering Committee.jmp

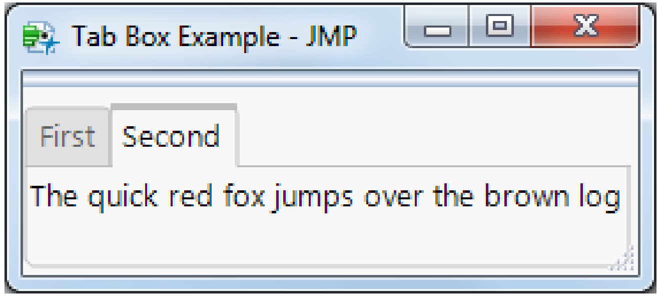
Tab Boxes

- Segment displays using a tabbed interface

```
// Example Tab Box.jsl
nw = new window("Tab Box Example",
    tb = tab box(
        "First Tab",
        vlistbox(
            text box("Hello"),
            text box("World")
        ),
        "Second Tab",
        text box("The quick red fox jumps over the brown log")
    )
);
```



Hello
World

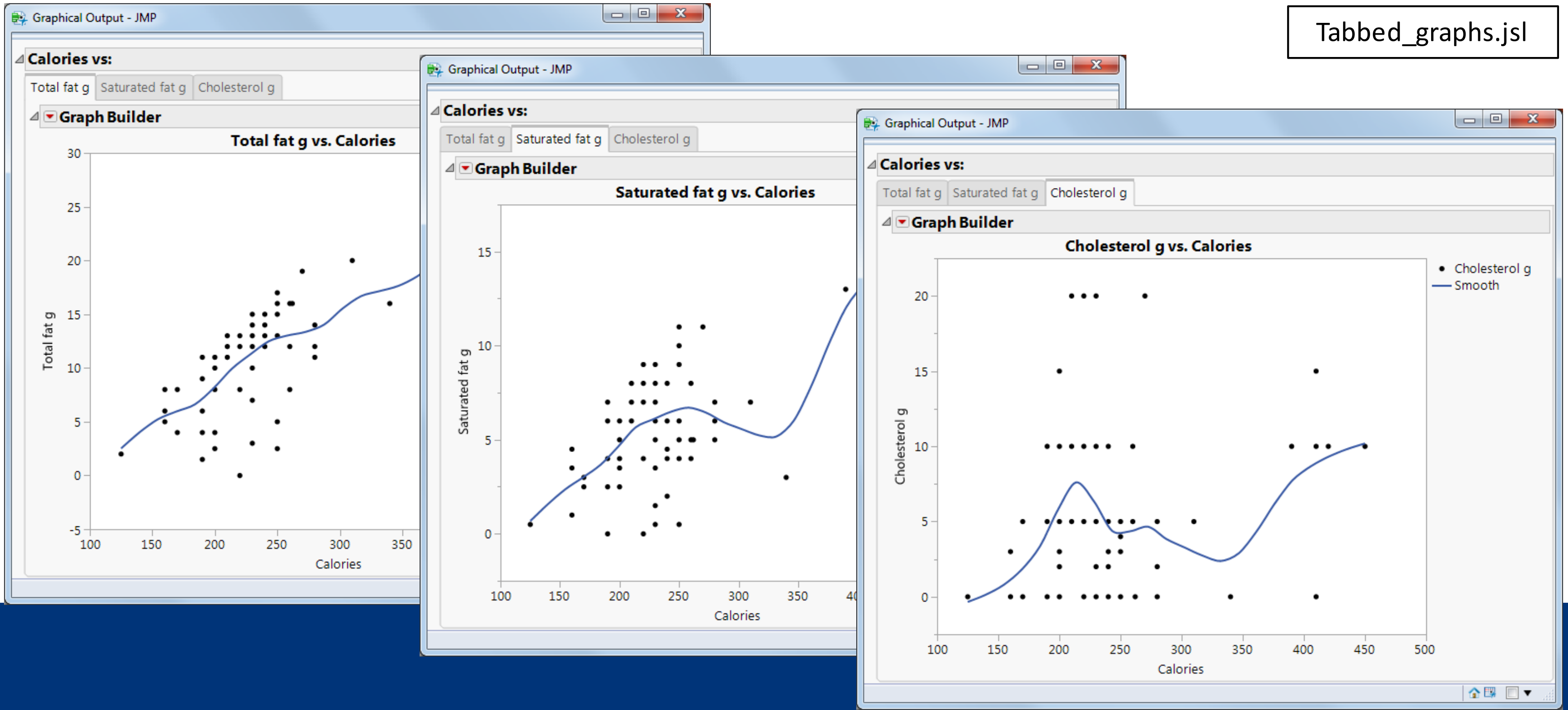


The quick red fox jumps over the brown log

Using Tab Boxes to Display Graphs

- Each tab shows a separate graph in a series

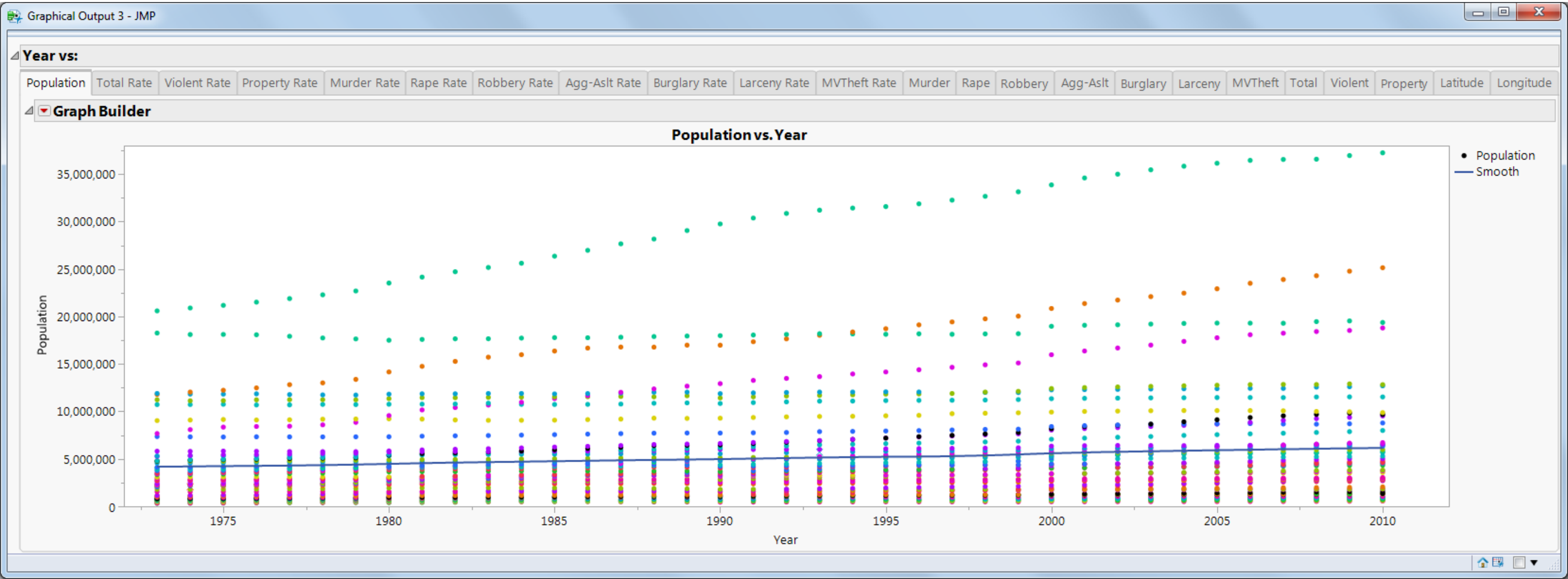
Tabbed_graphs.jsl



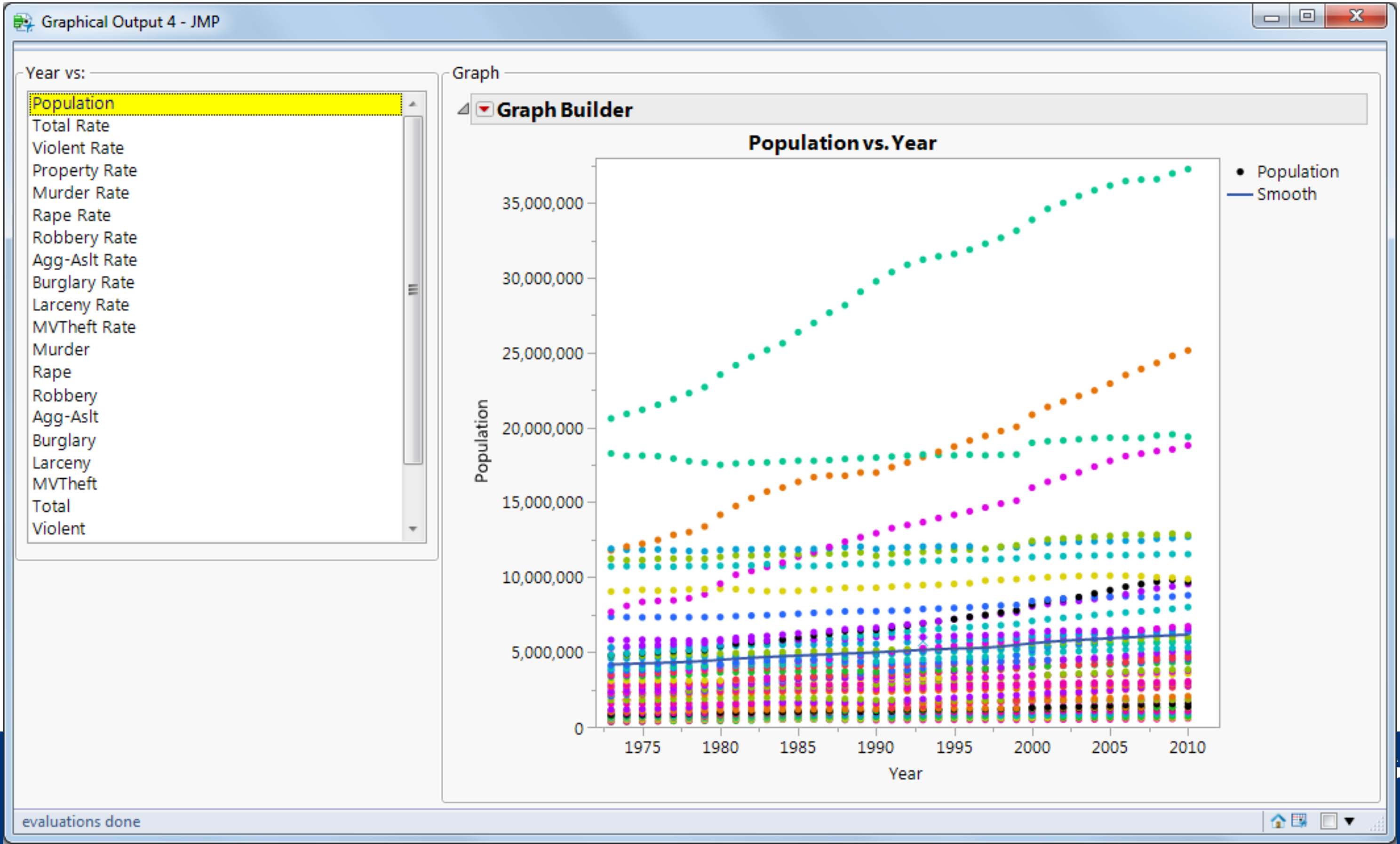
Tab Boxes are Great

- But what if there are too many tabs?

TooManyTabs_Graphs.jsl



Use a List Box, and Display One Graph at a Time


Graphical Output 4 - JMP

Year vs:

- Population
- Total Rate
- Violent Rate
- Property Rate
- Murder Rate
- Rape Rate
- Robbery Rate
- Agg-Aslt Rate
- Burglary Rate
- Larceny Rate
- MVTheft Rate
- Murder
- Rape
- Robbery
- Agg-Aslt
- Burglary
- Larceny
- MVTheft
- Total
- Violent

Graph

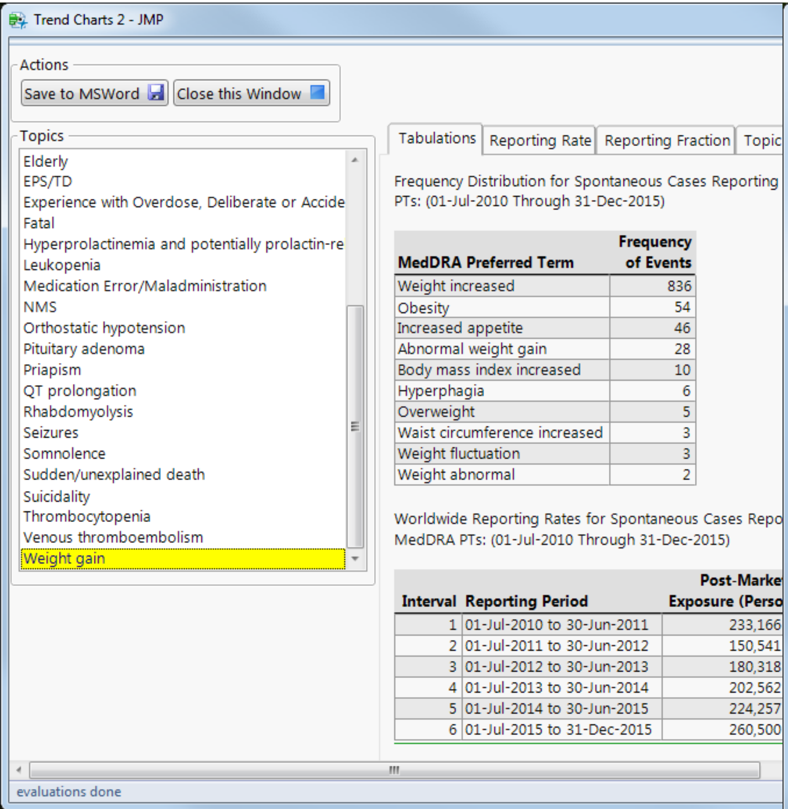
Graph Builder

evaluations done

Listbox_Graphs.jsl

Real World Example 2: Trending App

- Show trending information for many topics



Trend Charts 2 - JMP

Actions

Save to MSWord | Close this Window

Topics

Elderly
EPS/TD
Experience with Overdose, Deliberate or Accide
Fatal
Hyperprolactinemia and potentially prolactin-re
Leukopenia
Medication Error/Maladministration
NMS
Orthostatic hypotension
Pituitary adenoma
Priapism
QT prolongation
Rhabdomyolysis
Seizures
Somnolence
Sudden/unexplained death
Suicidality
Thrombocytopenia
Venous thromboembolism
Weight gain

Tabulations | Reporting Rate | Reporting Fraction | Topic

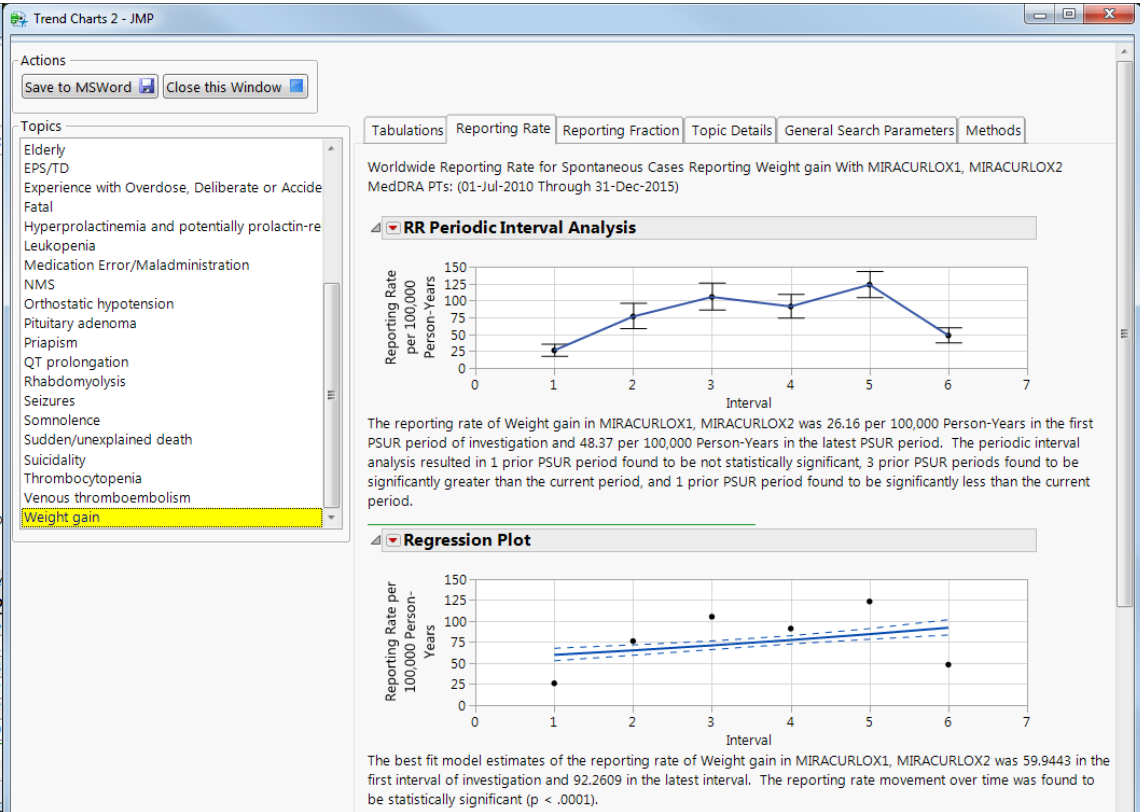
Frequency Distribution for Spontaneous Cases Reporting
PTs: (01-Jul-2010 Through 31-Dec-2015)

MedDRA Preferred Term	Frequency of Events
Weight increased	836
Obesity	54
Increased appetite	46
Abnormal weight gain	28
Body mass index increased	10
Hyperphagia	6
Overweight	5
Waist circumference increased	3
Weight fluctuation	3
Weight abnormal	2

Worldwide Reporting Rates for Spontaneous Cases Repo
MedDRA PTs: (01-Jul-2010 Through 31-Dec-2015)

Interval	Reporting Period	Post-Marke Exposure (Perso
1	01-Jul-2010 to 30-Jun-2011	233,166
2	01-Jul-2011 to 30-Jun-2012	150,541
3	01-Jul-2012 to 30-Jun-2013	180,318
4	01-Jul-2013 to 30-Jun-2014	202,562
5	01-Jul-2014 to 30-Jun-2015	224,257
6	01-Jul-2015 to 31-Dec-2015	260,500

evaluations done



Trend Charts 2 - JMP

Actions

Save to MSWord | Close this Window

Topics

Elderly
EPS/TD
Experience with Overdose, Deliberate or Accide
Fatal
Hyperprolactinemia and potentially prolactin-re
Leukopenia
Medication Error/Maladministration
NMS
Orthostatic hypotension
Pituitary adenoma
Priapism
QT prolongation
Rhabdomyolysis
Seizures
Somnolence
Sudden/unexplained death
Suicidality
Thrombocytopenia
Venous thromboembolism
Weight gain

Tabulations | Reporting Rate | Reporting Fraction | Topic Details | General Search Parameters | Methods

Worldwide Reporting Rate for Spontaneous Cases Reporting Weight gain With MIRACURLOX1, MIRACURLOX2 MedDRA PTs: (01-Jul-2010 Through 31-Dec-2015)

RR Periodic Interval Analysis

The reporting rate of Weight gain in MIRACURLOX1, MIRACURLOX2 was 26.16 per 100,000 Person-Years in the first PSUR period of investigation and 48.37 per 100,000 Person-Years in the latest PSUR period. The periodic interval analysis resulted in 1 prior PSUR period found to be not statistically significant, 3 prior PSUR periods found to be significantly greater than the current period, and 1 prior PSUR period found to be significantly less than the current period.

Regression Plot

The best fit model estimates of the reporting rate of Weight gain in MIRACURLOX1, MIRACURLOX2 was 59.9443 in the first interval of investigation and 92.2609 in the latest interval. The reporting rate movement over time was found to be statistically significant (p < .0001).

Associative Arrays

From the JSL Scripting book:

- Map unique keys to values that can be non-unique
- Also called a dictionary, a map, a hash map, or a hash table
- Keys are placed in quotes
- The value associated with a key can be a number, date, matrix, list, and so on.

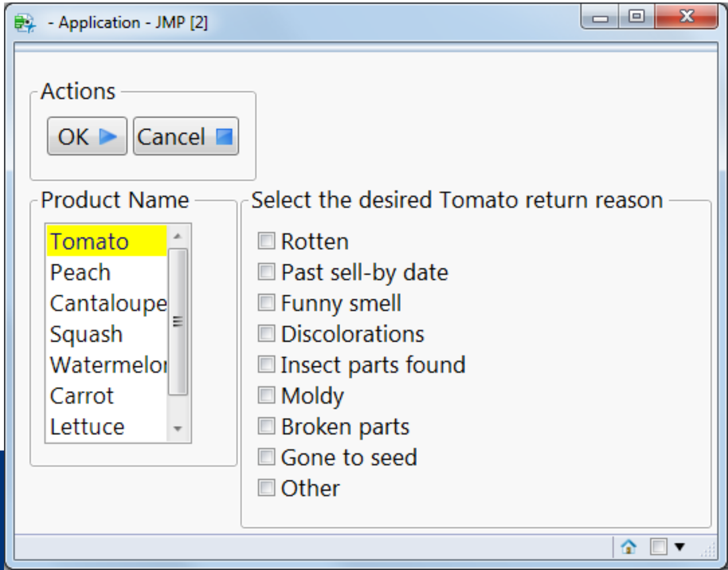
Example

```
aa = associative array();

aa["First"]  = {"Tom", "Jerry"};
aa["Second"] = {"Fred", "Wilma"};
aa["Third"]  = {"Pebbles", "Bam Bam"};

print(aa);

Associative Array({
    {"First", {"Tom", "Jerry"}},
    {"Second", {"Fred", "Wilma"}},
    {"Third", {"Pebbles", "Bam Bam"}}
})
```

Product Returns Application

- Click on product, select return reason(s)
- If another product selected
  - Store return reasons for previously selected product
  - Display return reasons for newly selected product

AssociativeArrayExample.jmpapp


- Application - JMP [2]

Actions

OK ▶ Cancel

Product Name

Tomato
Peach
Cantaloupe
Squash
Watermelon
Carrot
Lettuce

Select the desired Tomato return reason

☐ Rotten
☐ Past sell-by date
☐ Funny smell
☐ Discolorations
☐ Insect parts found
☐ Moldy
☐ Broken parts
☐ Gone to seed
☐ Other

Associative Arrays to the Rescue

```
// Use an associative array to store the return checkboxes for each product name

// Initialize associative array
return_aa     = associative array();
n_return      = nitems(return_cb << get items());

// Create a one-dimensional matrix of 0s
empty_list    = j(n_return, 1, 0);
```

Associative Arrays to the Rescue

```
// Use an associative array to store the return checkboxes for each product name

// Initialize associative array
return_aa     = associative array();
n_return      = nitems(return_cb << get items());

// Create a one-dimensional matrix of 0s
empty_list    = j(n_return, 1, 0);

// Get product names from product listbox
product_list = product_lb << get items;

for (i = 1, i <= nitems(product_list), i++,
    one_product = product_list[i];
    return_aa[one_product] = empty_list;
);
```

Initial Values for Associative Array

```
return_aa:
Associative Array({
{"Cantaloupe", [0, 0, 0, 0, 0, 0, 0, 0, 0]},
{"Carrot", [0, 0, 0, 0, 0, 0, 0, 0, 0]},
{"Cucumber", [0, 0, 0, 0, 0, 0, 0, 0, 0]},
{"Lettuce", [0, 0, 0, 0, 0, 0, 0, 0, 0]},
{"Peach", [0, 0, 0, 0, 0, 0, 0, 0, 0]},
{"Squash", [0, 0, 0, 0, 0, 0, 0, 0, 0]},
{"Tomato", [0, 0, 0, 0, 0, 0, 0, 0, 0]},
{"Watermelon", [0, 0, 0, 0, 0, 0, 0, 0, 0]}})
```

When checkboxes are checked or unchecked…

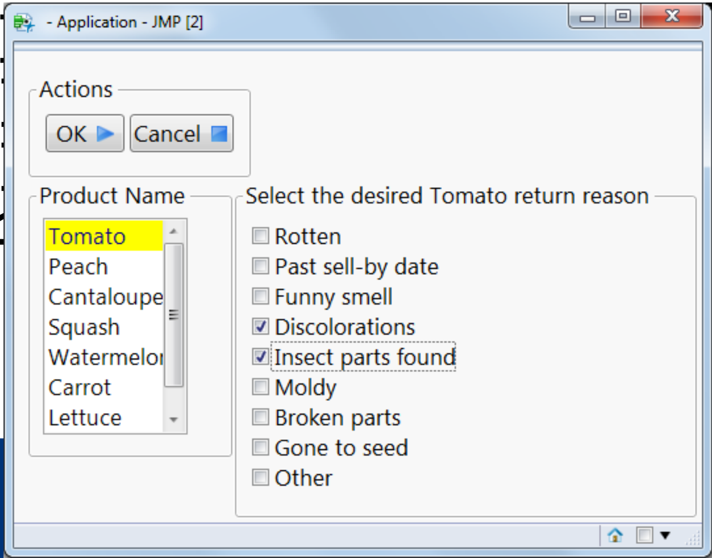
```
// Called when the return_cb check box selection changes
return_cbChange=Function({this, index},{selected},
    one_product_list = product_lb << get selected;

    if (nitems(one_product_list) > 0,
        one_product = one_product_list[1];
    );

// Get the status of the recently checked or unchecked checkbox
    one_checked = this << get(index);

// Save the checkbox status for this product name/return element
    return_aa[one_product][index] = one_checked;
);
```

When the product name changes…

```
// This function is called when the product_lb List Box selection changes
product_lbSelect=Function({this},{selectedIndex},

    one_product_list = this << get selected;
    if (nitems(one_product_list) > 0,
        one_product = one_product_list[1];

// Put the newly selected product name into some labels
        return_panel  << set title("Select the desired " ||
            one_product || " return reason");

// Set the return_cb checkboxes to this product’s values
        for (i = 1, i <= n_return, i++,
            return_cb << set(i, return_aa[one_product][i]);
        );
    );
);
```

Returns Associative Array with some Checkboxes Checked

```
Associative Array({
{"Cantaloupe", [0, 0, 0, 0, 0, 0, 0, 0, 0]},
{"Carrot", [0, 0, 0, 0, 0, 0, 0, 0, 0]},
{"Cucumber", [0, 0, 0, 0, 0, 0, 0, 0, 0]},
{"Lettuce", [0, 0, 0, 0, 0, 0, 0, 0, 0]},
{"Peach", [1, 1, 0, 0, 0, 0, 0, 0, 0]
{"Squash", [0, 0, 0, 0, 0, 0, 0, 0, 0
{"Tomato", [0, 0, 0, 1, 1, 0, 0, 0, 0
{"Watermelon", [0, 0, 0, 0, 0, 0, 1, 
```


- Application - JMP [2]

Actions

OK | Cancel

Product Name
- Tomato
- Peach
- Cantaloupe
- Squash
- Watermelon
- Carrot
- Lettuce

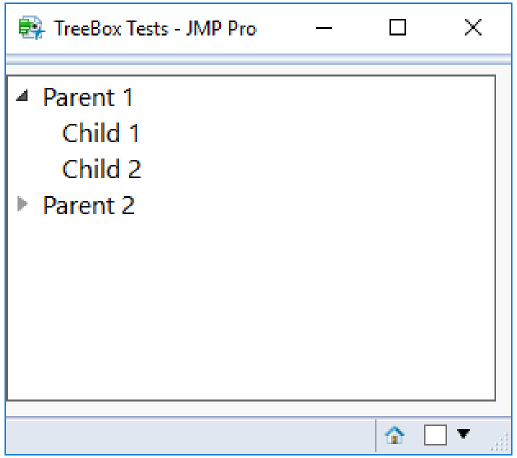
Select the desired Tomato return reason
- [ ] Rotten
- [ ] Past sell-by date
- [ ] Funny smell
- [x] Discolorations
- [x] Insect parts found
- [ ] Moldy
- [ ] Broken parts
- [ ] Gone to seed
- [ ] Other

Tree Nodes and Tree Boxes

- Tree Node
  - A tree data structure in JMP that can be displayed using a Tree Box.
  - Has a label, which appears in the Tree Box, but also can hold data (any JMP object).
- Tree Box
  - Shows Tree Nodes, allowing you to select and collapse the nodes as desired.
  - Can have various kinds of callback functions, which are useful when updating a window based on selection.

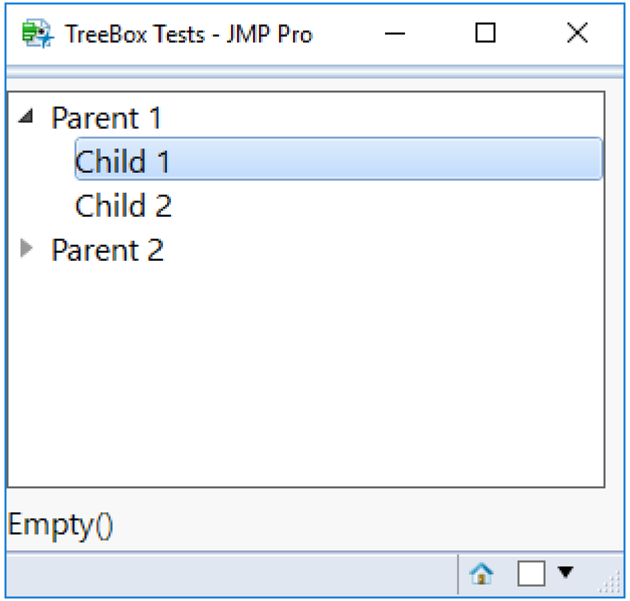
Simple Tree Box Example

```
root1 = Tree Node( "Parent 1" );
root2 = Tree Node( "Parent 2" );

c1 = Tree Node( "Child 1" );
c2 = Tree Node( "Child 2" );
c3 = Tree Node( "Child 3" );
c4 = Tree Node( "Child 4" );

root1 << Append( c1 );
root1 << Append( c2 );
root2 << Append( c3 );
root2 << Append( c4 );

nw = New Window( "TreeBox Tests",
    tree = Tree Box( {root1, root2}, Size( 300, 200 ) )
);

tree << Expand( root1 );
```



Adding Data to Tree Nodes

```
// add a text box to the window
nw << Append( tb = Text Box() );

// add some data to the root nodes
root1 << Set Data( "Welcome to Cary!" );
root2 << Set Data( "See you next year!" );

// add a callback function when selecting a node
tree << Set Node Select Script(
   Function({tree, node},
      If( !Is Empty( node ),
         tb <<  Set Text( Char( node << Get Data() ) )
      )
   )
);
```

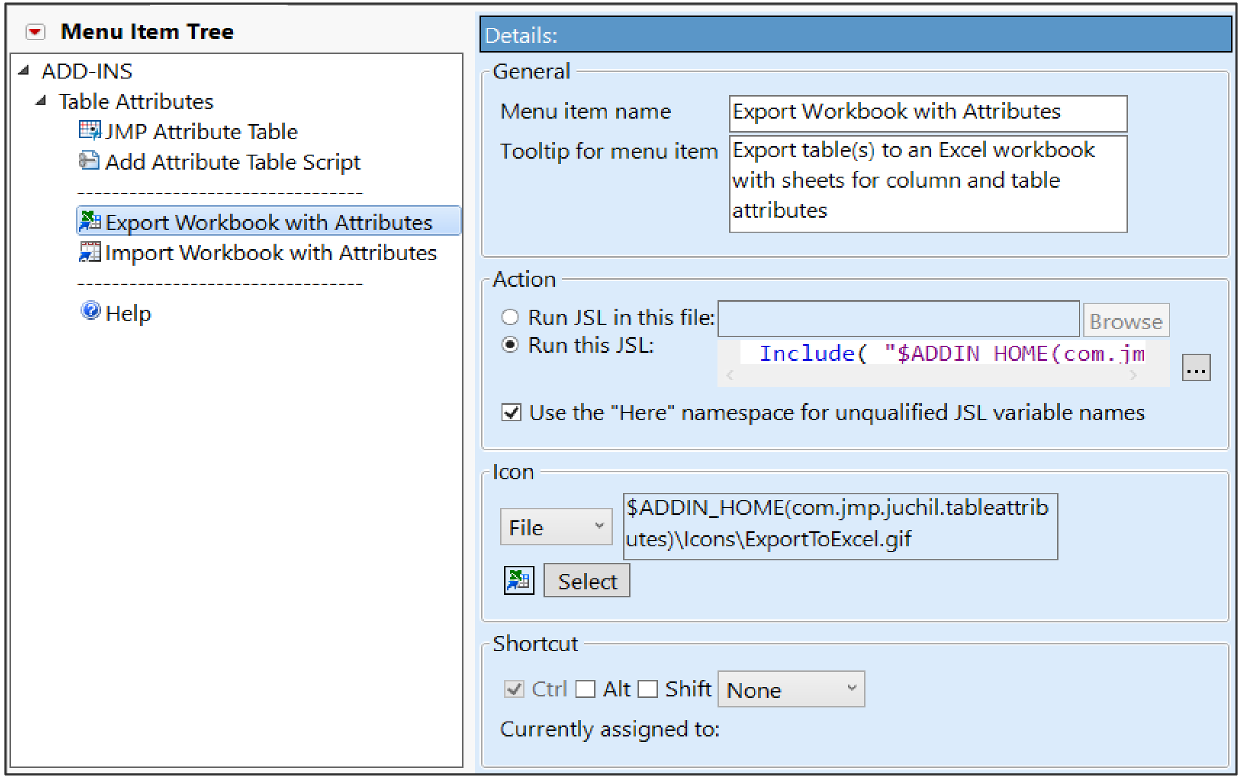


Empty()

Candy Bars Example - Demo

Candy Bar Nutrition.jsl

Real World Example – Add-In Manager



Menu Item Tree

- ADD-INS
  - Table Attributes
    - JMP Attribute Table
    - Add Attribute Table Script
    - ----------------------------------
    - Export Workbook with Attributes
    - Import Workbook with Attributes
    - ----------------------------------
    - Help

Details:

General

Menu item name	Export Workbook with Attributes
Tooltip for menu item	Export table(s) to an Excel workbook with sheets for column and table attributes

Action

( ) Run JSL in this file: Browse

(●) Run this JSL:

```
Include( "$ADDIN HOME(com.jm
```

[✓] Use the "Here" namespace for unqualified JSL variable names

Icon

File: $ADDIN_HOME(com.jmp.juchil.tableattributes)\Icons\ExportToExcel.gif

Select

Shortcut

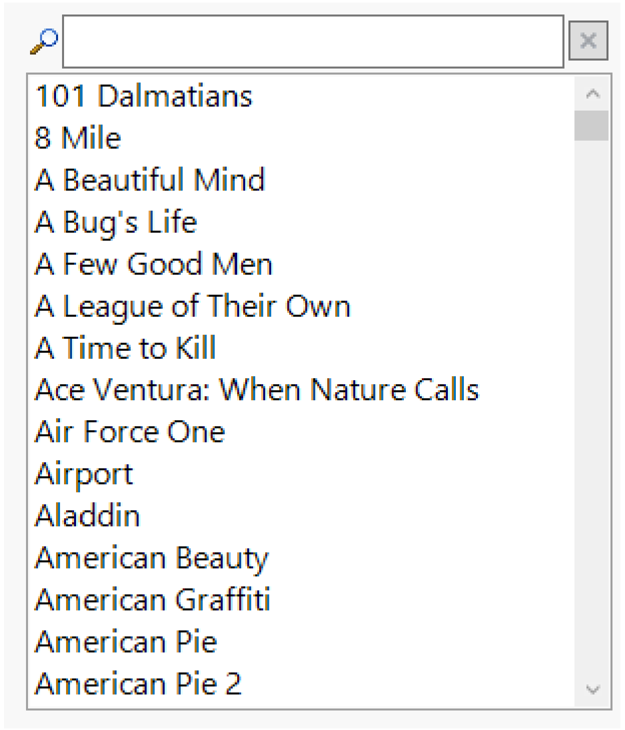
[✓] Ctrl [ ] Alt [ ] Shift None

Currently assigned to:

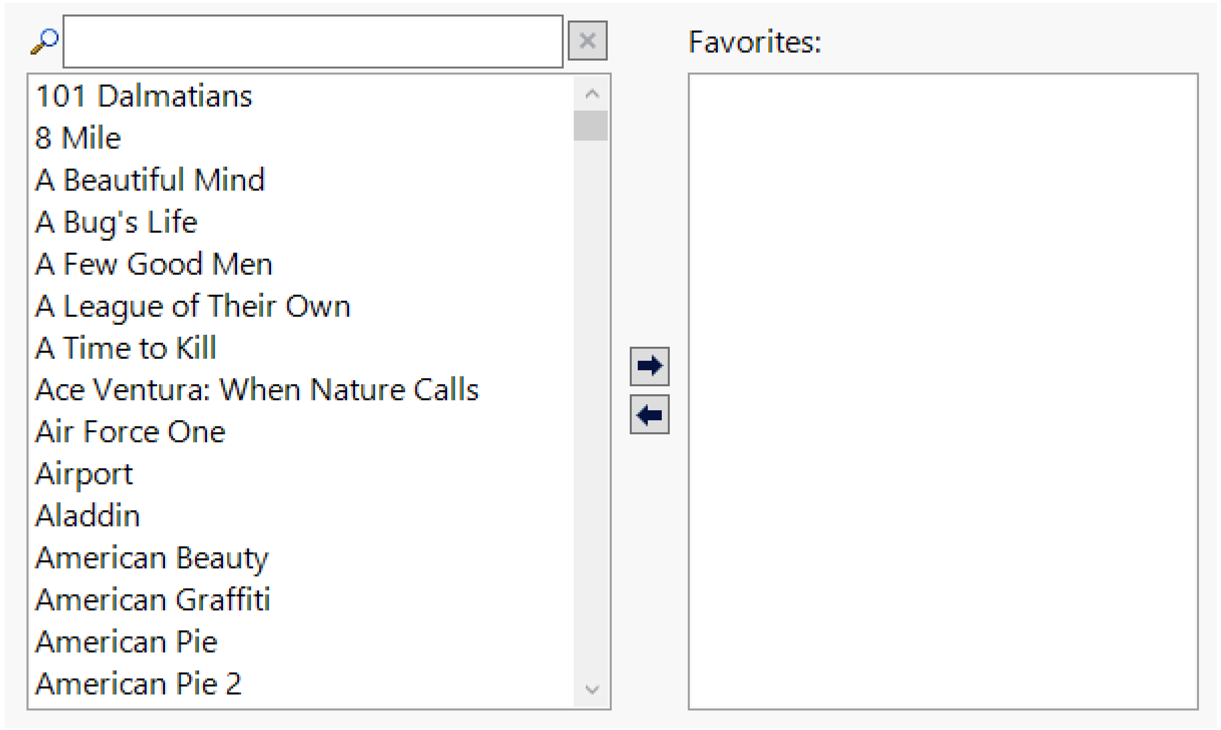
Filtering Long Picklists

- Sometimes there are just too many items in a list box to find what you are looking for.
- Implementing your own search box can help reduce the need for scrolling through these items.

Filtering Long Picklists – Single Select


101 Dalmatians
8 Mile
A Beautiful Mind
A Bug's Life
A Few Good Men
A League of Their Own
A Time to Kill
Ace Ventura: When Nature Calls
Air Force One
Airport
Aladdin
American Beauty
American Graffiti
American Pie
American Pie 2

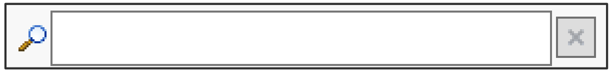
Filtering Long Picklists – Multiselect



101 Dalmatians
8 Mile
A Beautiful Mind
A Bug's Life
A Few Good Men
A League of Their Own
A Time to Kill
Ace Ventura: When Nature Calls
Air Force One
Airport
Aladdin
American Beauty
American Graffiti
American Pie
American Pie 2

Search Box

```
H List Box(
   Align( "Center" ),
   Icon Box( "SearchIndex" ),
   filter_teb = Text Edit Box( "",
      <<Set Width( 250 ),
      <<Set Text Changed( filterMovies )
   ),
   Button Box( "",
      <<Set Icon( "TabClose" ),
      <<Set Script(
         // clear the filter and call the text changed function
         filter_teb << Set Text( "" );
         filterMovies( filter_teb, "" );
      ),
      <<Set Tip( "Clear Filter" )
   )
)
```



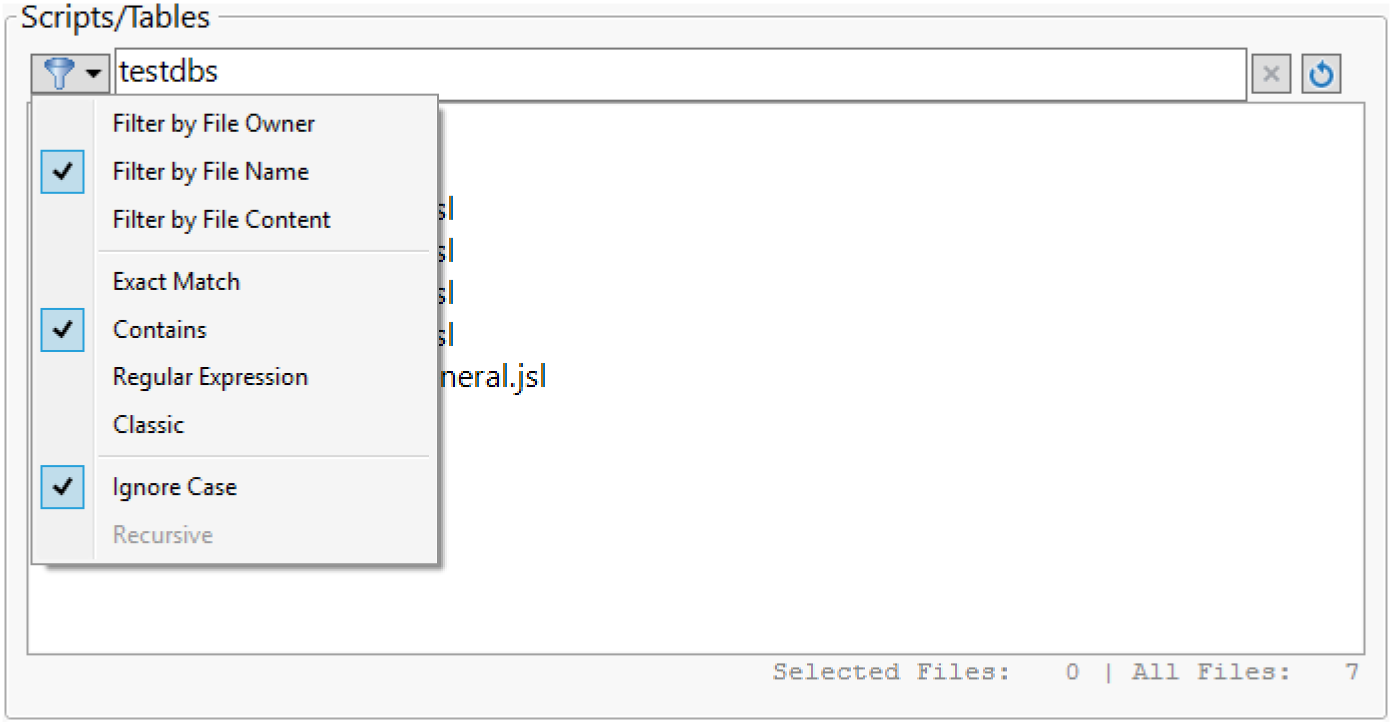
Filter Function

```
filterMovies = Function( {this, searchText},
   {filtered_movies, i},
   // only attempt to filter if there is any text
   If( searchText != "",
      // new list for movies that match searchText
      filtered_movies = {};
      // Check if each movie matches the given text
      For( i = 1, i <= N Items( all_movies_list ), i++,
         // Insert to our list if it contains our search text (case insensitive)
         If( Contains( Lowercase( all_movies_list[i] ), Lowercase( searchText ) ),
            Insert Into( filtered_movies, all_movies_list[i] );
         )
      );
   ,
      // else show all movies
      filtered_movies = all_movies_list;
   );
   nonFavMovies_lb << Set Items( filtered_movies );
);
```

Favorite Movies Example - Demo

Favorite Movies.jsl

Real World Example – JMP Testing Framework



Scripts/Tables

testdbs

Filter by File Owner
✓ Filter by File Name
Filter by File Content

Exact Match
✓ Contains
Regular Expression
Classic

✓ Ignore Case
Recursive

sl
sl
sl
sl
neral.jsl

```
Selected Files:    0 | All Files:    7
```

Conclusions

- Col boxes are a useful addition to a tablebox
- Tab boxes are great for segmenting displays
- Associative arrays are useful for storing complex state information
- Tree nodes and tree boxes are excellent for working with hierarchical data
- Filtering long picklists can easily be done in JSL

Key Learnings

- Listen to your users
- Listen some more
- Keep listening!
- Don’t say no right away
- Show prototypes
- Users don’t know what they want until they see what they don’t want

Thank you

Martin Freeman, *Untitled*
Diagnosed with AIDS in 1990,
Martin lives in San Francisco where
he continues to create new pieces.

Peter Mroz
pmroz@its.jnj.com

Justin Chilton
Justin.Chilton@jmp.com

janssen | PHARMACEUTICAL COMPANIES OF Johnson & Johnson