

A Practitioner's Perspective

GUI Development

Mark Kotanchek
Evolved Analytics LLC

Ariel Sepúlveda
Pronto Analytics Inc.

Abstract

- ❖ Although **Manipulate** is great for quick-and-dirty interactive interfaces, non-trivial interfaces require migrating towards **DynamicModule** and finer-grained control of the **Dynamic** evaluations. The Evolved Analytics' *DataModeler* GUI (Graphical User Interface) is one of the most sophisticated systems developed in 100% pure Mathematica but we have also developed other GUI systems which leverage Enterprise *Mathematica* to be able to provide clients with solutions which only require CDFPlayer and hide the complexity of the Wolfram Language.
- ❖ In this talk we will address the implementation best practices which have emerged over the years as well as some of the more subtle functional forms and option settings for dynamics and notebooks which contribute to success.

GUI vs. Notebook Interface

Mathematica

File

Edit

Insert

Format

Cell

Graphics

Evaluation

Palettes

Window

Help

Launch Project

Select Variables

Explore Data

Generate Models

Analyze Models

Test & Validate

What If?

Reports

Target Response

Death Age Gap

232

Variable Set

Round 4 Inputs

Data File Summary

Exploring Methods

Max Explorers

1

☒ DataSummaryTable

☐ View Data to Explore

☐ Imported Data Report

☐ UnivariatePlot

☐ DataDistributionPlot

☐ DataCompletenessMap

☐ DataCompletenessPlot

☐ BivariatePlot

☐ CorrelationChart

☐ CorrelationMatrixPlot

DataModeler 9.1 - [DeathAgeGapAnalysis]

Page 1

Page 2

DataSummaryTable

Col	Label	Type	Uniformity	Class	Unique	Distribution Plot	Zero-Cross	Min	Mean	Median	Max
3	AdultPopulation	123	96%		232			3508.0	2.0×10^7	3.5×10^6	9.9×10^8
20	DeathRateFraction	123	94%		202			0.0	0.0	0.0	0.0
35	FemaleLiteracyFraction	123	86%		142			0.1	0.8	0.9	1.0
42	GDPPerCapita	123	96%		231			137.6	15952.0	5514.7	211500.0
61	LifeExpectancy	123	96%		228			45.6	71.5	74.0	84.4
69	MaleLiteracyFraction	123	86%		125			0.3	0.9	0.9	1.0
98	PopulationGrowth	123	99%		231			-0.0	0.0	0.0	0.2
125	Male Smoking Deaths	123	75%		138			0.0	13.2	11.0	37.6
126	Female Smoking Deaths	123	75%		91			0.0	5.5	3.7	24.9
150	Total	123	79%		107			0.1	6.2	6.5	17.5
158	Prisoner Rate	123	91%		165			0.0	176.8	140.0	868.0
159	Communicable Disease Death Rate	123	79%		191			1.5	26.5	16.5	81.3
160	Injury Death Percent	123	79%		191			1.7	8.2	7.3	31.4

Initialization

Load DataModeler

Needs["DataModeler`"]

\$VersionOfDataModeler

DataModeler 9.0 (24 July 2016)

Load the data

AbsoluteTiming[

{countryHeaders, countryData} =

ImportDataMatrix[

"AugmentedCountryData_MultipleSources_2016-07-20-22-20-59.xls"];

Dimensions@countryData

]

{2.19387, {240, 232}}

Load All Developed Models

AbsoluteTiming@Length[

allModels = RetrieveModelSets["*Round*"]

]

{9.92449, 15364}

Testing

AbsoluteTiming[

ParetoFrontPlot[allModels]

]

Pareto Front Plot — 10107 models

{0.409862,

$-5.07 - (5.24 \times 10^{-2}) \text{communicable} \cdot \text{Disease} \cdot \text{Death} \cdot \text{Rate} - (1.58 \times 10^{-2}) \text{female} \cdot \text{Labor} \cdot \text{Fraction} + (4.75 \times 10^{-2}) \left(4 + \frac{\text{elderlyPopulation}}{\text{maleElderlyPopulation}}\right)^3$

AbsoluteTiming@Length[

interestingModels = SelectModels[allModels, QualityBox -> {100, 0.4}]

]

{0.859082, 2265}

AbsoluteTiming@Length[

interestingModels = SelectModels[allModels, QualityBox -> {100, 0.4}]

]

{0.859082, 2265}

Welcome to Evolved Analytics' DataModeler

Quick Start Guide

Start New Project

Open Project

Product Tour

Contact Us

01

02

03

04

05

Explore the Quick Start Tutorial to hit the ground running

Start a New Project from scratch

Open Existing Project to explore, modify or continue analysis

Take a quick product tour to see all the cool features

Contact us to share feedback or suggest improvements

GUI vs. Notebook Interface

Nonlinear interaction
Hides complexity of Mathematica
Design limits functionality

Linear view
Natural audit trail
Requires Mathematica awareness
Long-distance interactions
Execution order matters
No limits

Agenda

- ❖ Motivations
- ❖ Manipulate — Advantages & Limitations
- ❖ GUI Layout Tools
- ❖ Development Environment
- ❖ Interface Elements
- ❖ Design Review of Deployed GUIs
- ❖ Dynamics
- ❖ Issues
- ❖ Deployment
- ❖ Best Practices
- ❖ IP Protection

Why Build a GUI?

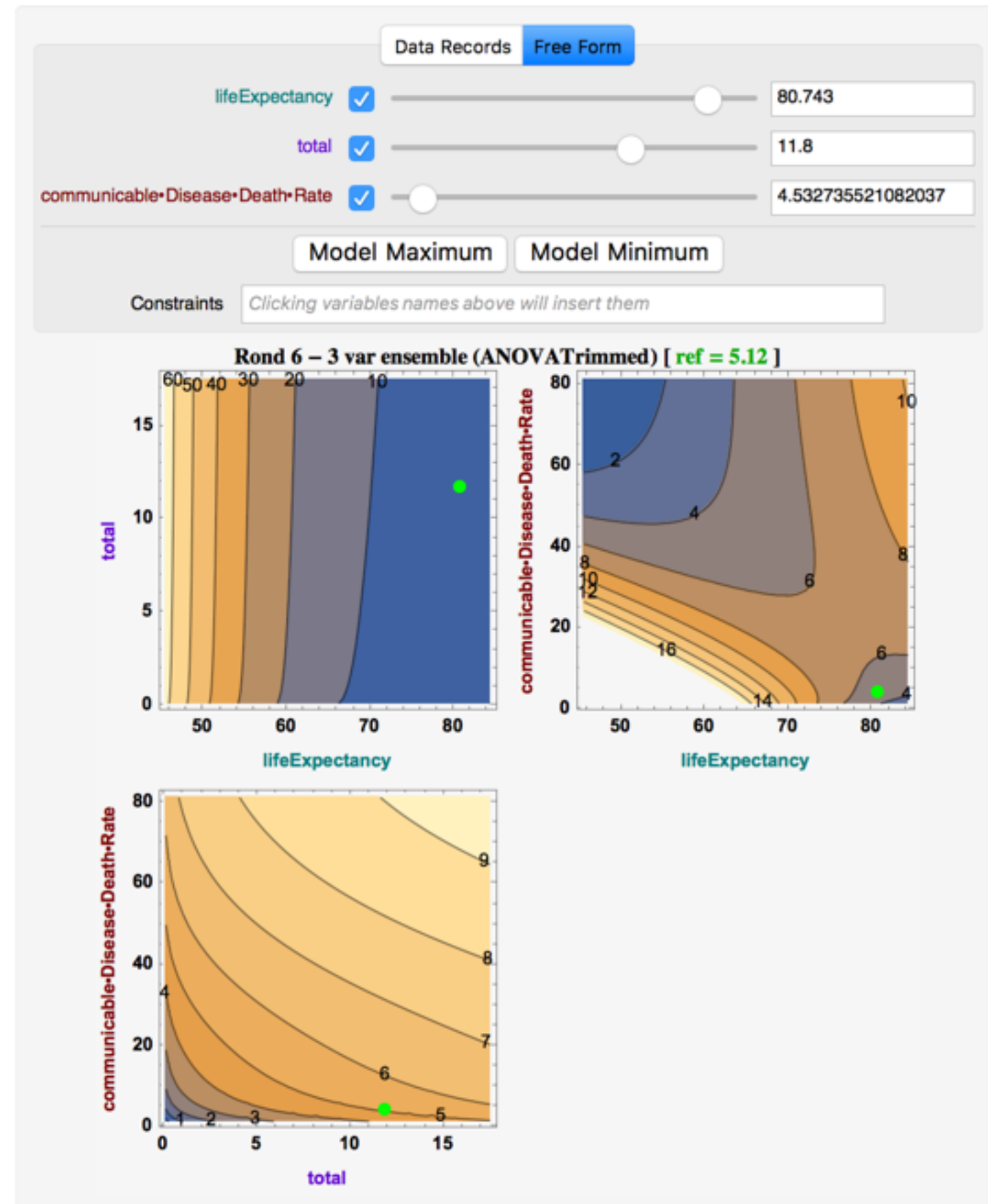
- ❖ Nonlinear Workflow
 - ❖ Iterative Analysis
 - ❖ Design option exploration
- ❖ Control the Environment
 - ❖ Hide Complexity
 - ❖ Constrain behaviors
- ❖ Ease-of-Use
 - ❖ No Mathematica knowledge required!

A GUI can have a prohibitive advantage relative to the linear and no-rules nature of a conventional Mathematica notebook

Credentials

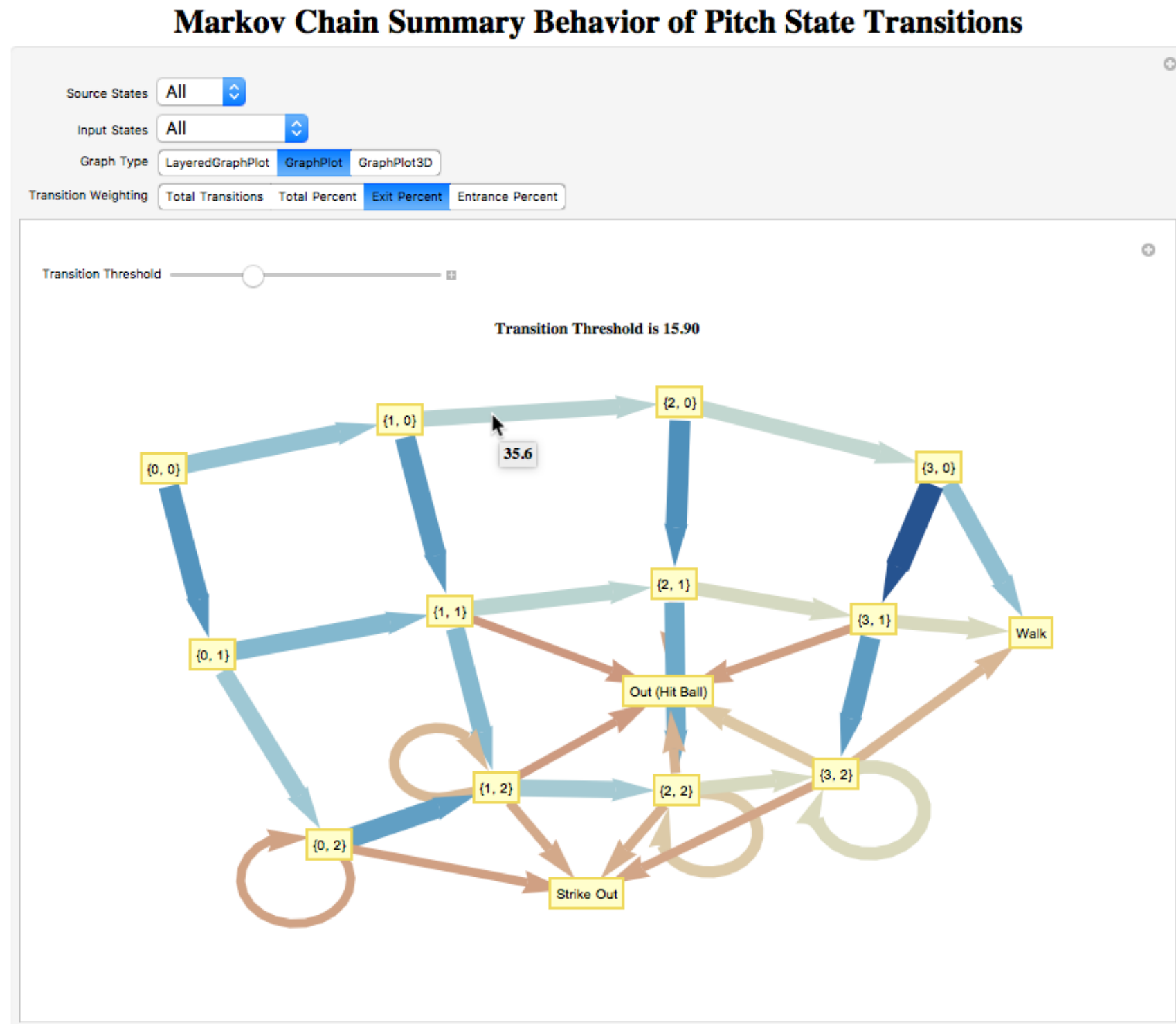
- ❖ Mark
 - ❖ DataModeler Response Explorers
 - ❖ Image Analyzer
 - ❖ Knitting Pattern Design
- ❖ Ariel
 - ❖ DataModeler GUI
 - ❖ BESTViewpoints
 - ❖ BEST DB Editor
 - ❖ OEE RADAR
 - ❖ consulting

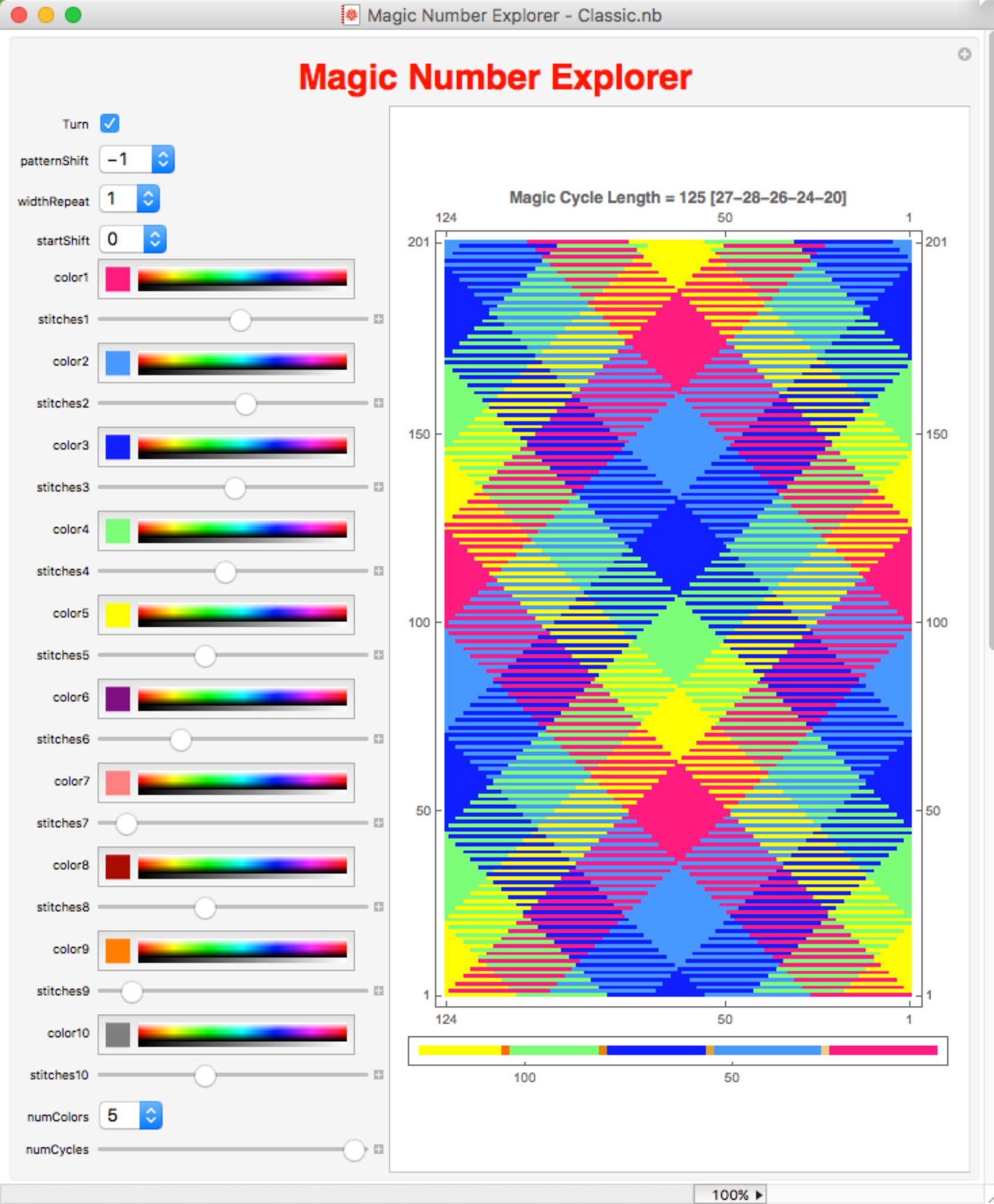
```
ResponseContourPlotExplorer[ensemble3Vars, countryData,  
  DataRecordLabels → "Country"  
]
```



What's Great about Manipulate?

- ❖ Great for quick-and-dirty interactive exploration
- ❖ Can achieve reasonably decent interfaces
- ❖ Intelligent defaults on behaviors

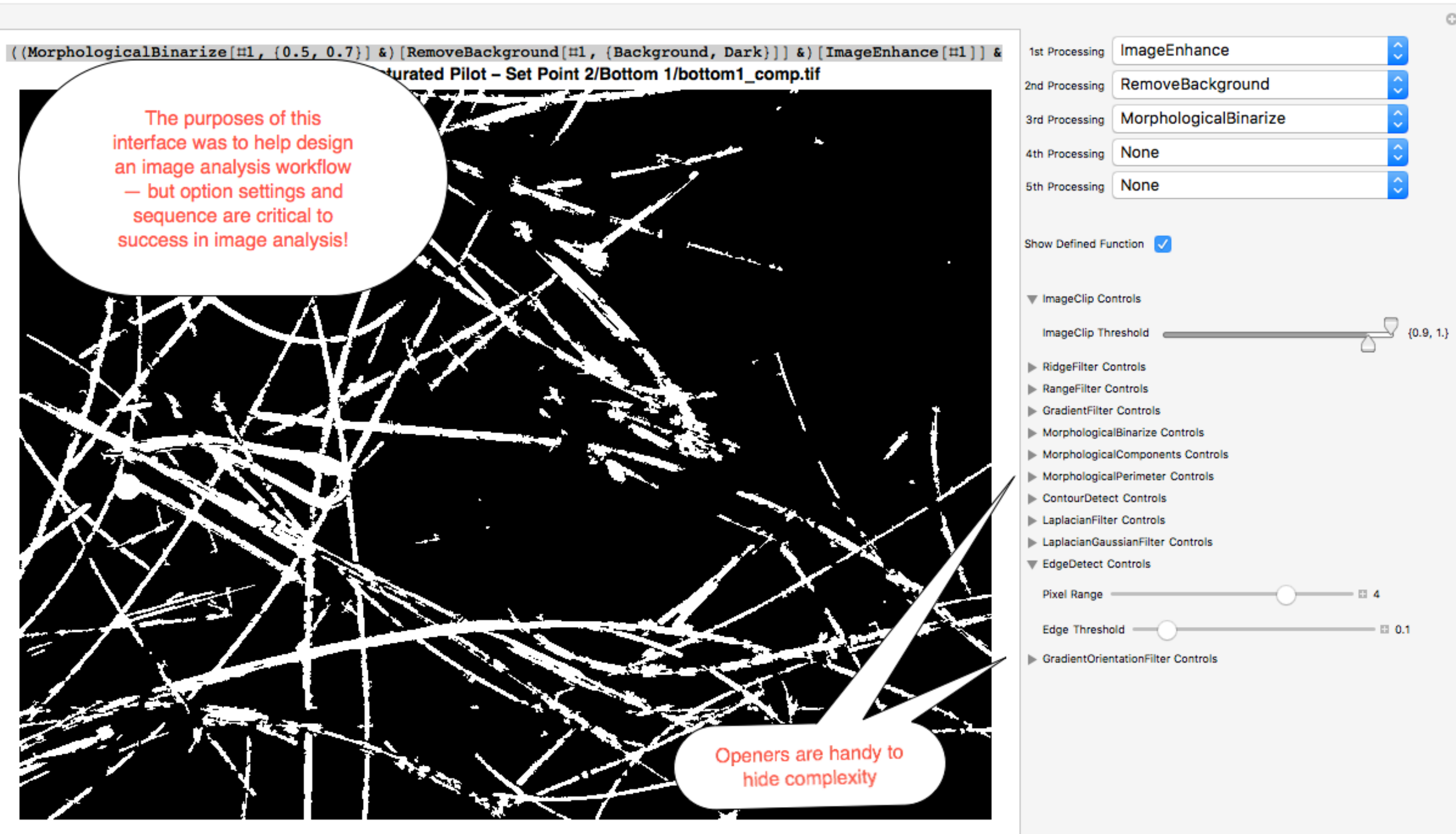




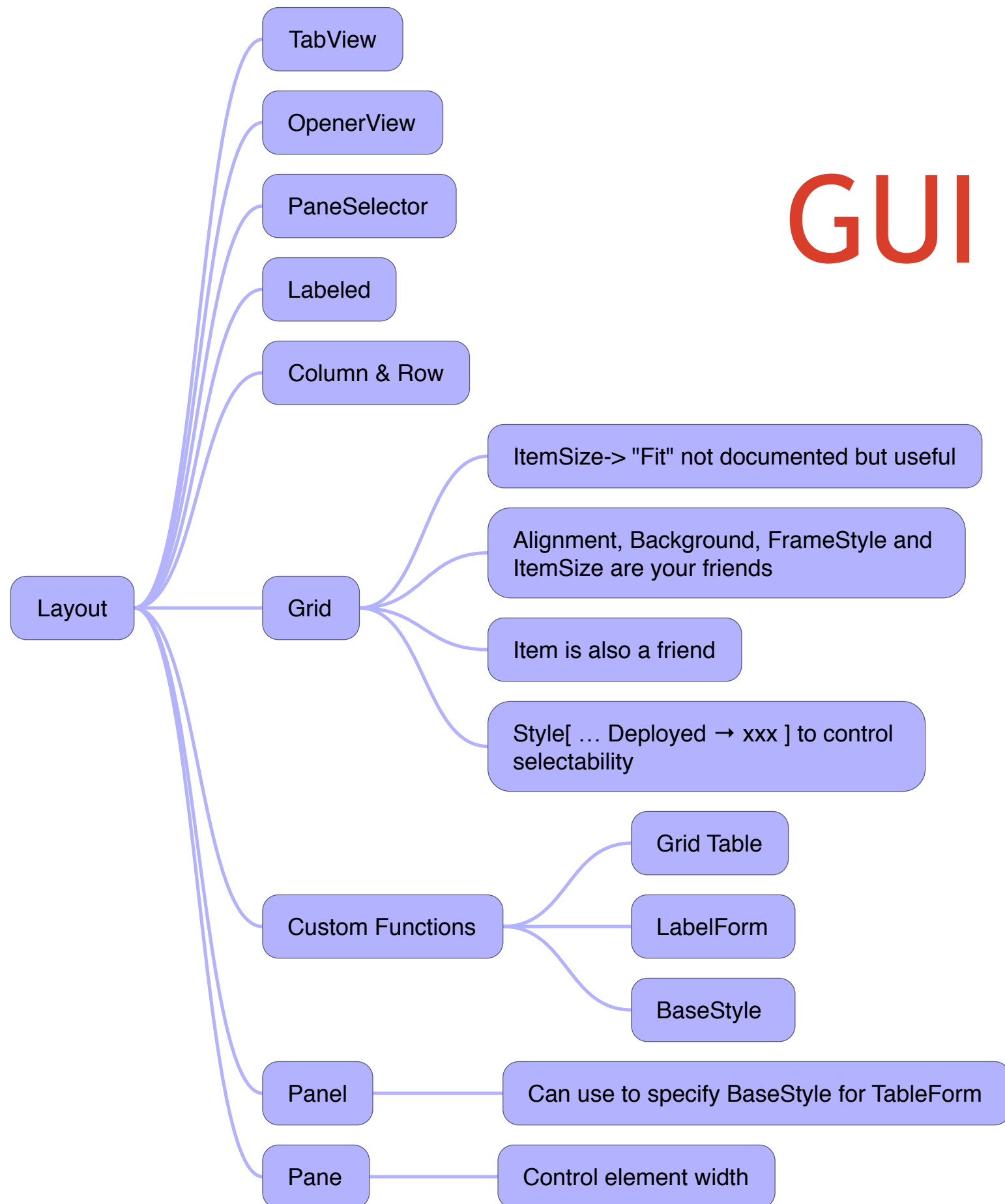
What's Wrong with Manipulate?

- ❖ Doesn't scale well to complicated interfaces
- ❖ Difficult to precisely control the firing of variable updates

Manipulate is Handy!



GUI Layout



- ❖ These are the functions I most use for structuring a GUI
- ❖ You want to layer the complexity – TabView & OpenerView most useful
- ❖ Respect different screen resolutions!

The ability to easily kill the kernel running the interface is very important

GUI code tends to be very verbose so indentations and being able to easily scan for instances of variables is huge

DataModeler has ~30,000 lines of code to implement 470 functions. Another 15,000 is needed to implement the GUI.

Ariel would put the commas at the front of the line

```
Grid[{{
Column[{
TabView[{
"Yarn" -> OpenerView[{
Style["Yarn Details", FontSize->16, FontWeight->Bold, FontColor->Magenta],
Grid[
{
{"Brand", InputField[Dynamic@yarnBrand, String]},
{"Name", InputField[Dynamic@yarnName, String]},
{"Color #", InputField[Dynamic@yarnNumber, String]},
{"Lot #", InputField[Dynamic@yarnLot, String]},
{"Design Notes", InputField[Dynamic@designNotes, String]}
}, Alignment -> {{Right, Left}, Center}
, Dividers -> {False, { {Gray, False}}}
}],
True],
"Gauge" -> OpenerView[{
Style["Gauge Details", FontSize->16, FontWeight->Bold, FontColor->Magenta],
Dynamic[
Refresh[
Grid[{
{"Stitch Pattern", InputField[Dynamic@stitchPattern, String]},

{StringJoin["Stitches/", stitchUnits], InputField[Dynamic@stitchesPerStitchUnit, Number]},
{StringJoin["Rows/", stitchUnits], InputField[Dynamic@rowsPerStitchUnit, Number]},

{"Preferred Units", RadioButtonBar[Dynamic@stitchUnits, {"cm", "inch"}]},
{"Needle Type", InputField[Dynamic@needleType, String]},
{"Needle Size", InputField[Dynamic@needleSize, String]}
}, Alignment->{{Right, Left}, Center}
, Dividers->{False,False}
}
}
```

-
- The screenshot shows the Wolfram Language Development interface. The top toolbar includes icons for file operations, debugging, and execution. The main window displays a notebook titled 'FiberglassImageAnalysis.nb [Wolfram Workbench]'. The code editor shows a snippet of Mathematica code for a GUI. Three callout boxes provide context:
- Top Left:** GUI code tends to be very verbose so indentations and being able to easily scan for instances of variables is huge.
 - Top Right:** The ability to easily kill the kernel running the interface is very important.
 - Center:** *DataModeler* has ~30,000 lines of code to implement 470 functions. Another 15,000 is needed to implement the GUI.
- The code in the editor is as follows:
- ```
Grid[{{
 Column[{
 TabView[{
 "Yarn" -> OpenerView[{
 Style["Yarn Details", FontSize->16, FontWeight->Bold, FontColor->Magenta],
 Grid[
 {
 {"Brand", InputField[Dynamic@yarnBrand, String]},
 {"Name", InputField[Dynamic@yarnName, String]},
 {"Color #", InputField[Dynamic@yarnNumber, String]},
 {"Lot #", InputField[Dynamic@yarnLot, String]},
 {"Design Notes", InputField[Dynamic@designNotes, String]}
],
 Alignment -> {{Right, Left}, Center},
 Dividers -> {False, { {Gray, False}}}
],
 True,
],
 "Gauge" -> OpenerView[{
 Style["Gauge Details", FontSize->16, FontWeight->Bold, FontColor->Magenta],
 Dynamic[
 Refresh[
 Grid[{
 {"Stitch Pattern", InputField[Dynamic@stitchPattern, String]},
 {StringJoin["Stitches/", stitchUnits], InputField[Dynamic@stitchesPerStitchUnit, Number]},
 {StringJoin["Rows/", stitchUnits], InputField[Dynamic@rowsPerStitchUnit, Number]},
 {"Preferred Units", RadioButtonBar[Dynamic@stitchUnits, {"cm", "inch"}]},
 {"Needle Type", InputField[Dynamic@needleType, String]},
 {"Needle Size", InputField[Dynamic@needleSize, String]}
],
 Alignment->{{Right, Left}, Center},
 Dividers->{False, False}
],
 1
]
]
 }
]
}],
 True,
],
True,
```

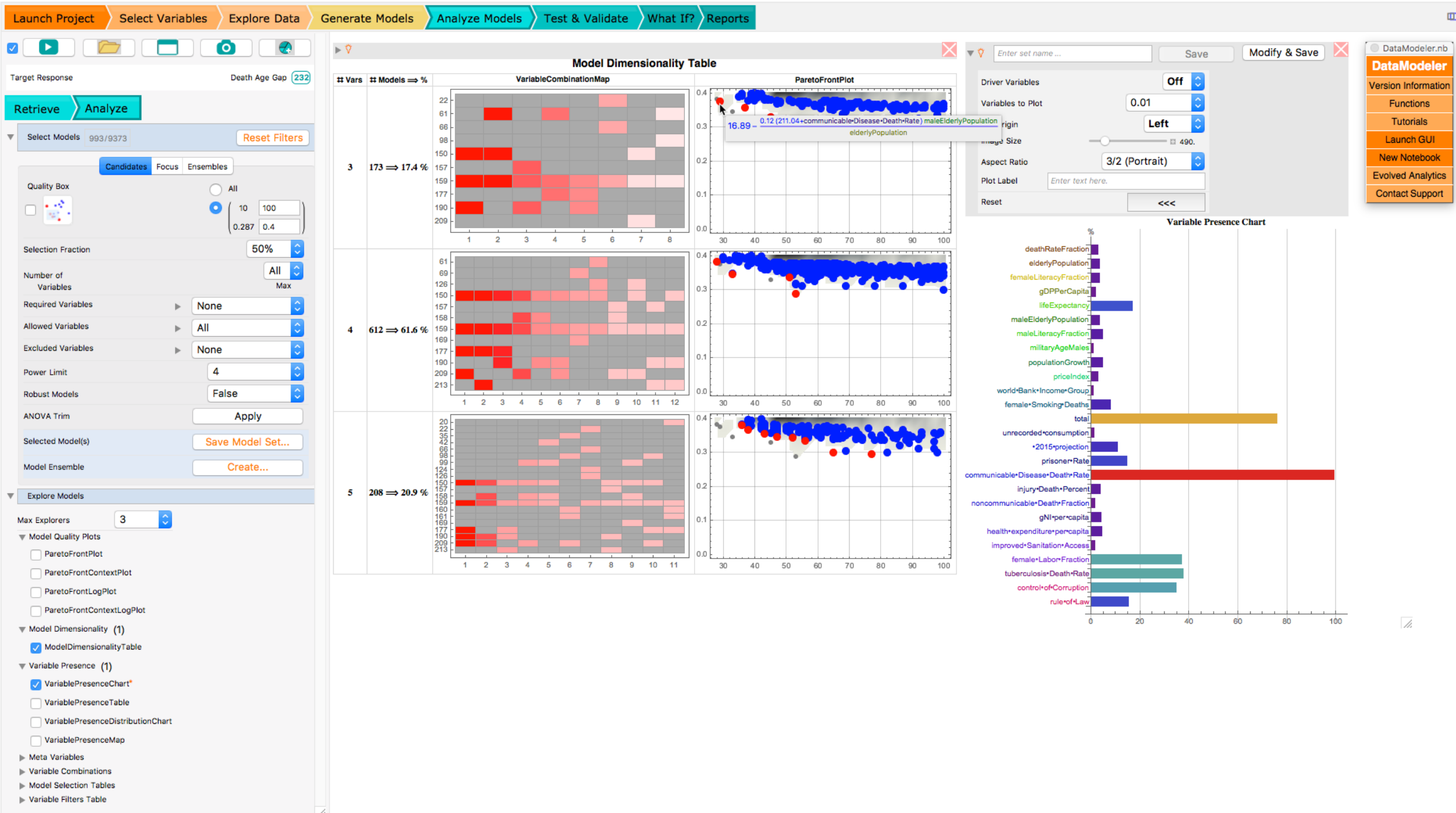
|                    |                                                                          |
|--------------------|--------------------------------------------------------------------------|
| <b>Buttons</b>     | Button<br>Setter<br>SetterBar<br>ButtonBar<br>ActionMenu                 |
| <b>Sliders</b>     | Slider<br>Slider2D<br>VerticalSlider<br>IntervalSlider                   |
| <b>Gauges</b>      | HorizontalGauge<br>VerticalGauge<br>BulletGauge<br>AngularGauge          |
| <b>Switches</b>    | Checkbox<br>RadioButton<br>RadioButtonBar<br>TogglerBar                  |
| <b>Togglers</b>    | Toggler<br>TogglerBar                                                    |
| <b>Colors</b>      | ColorSetter<br>ColorSlider                                               |
| <b>Pickers</b>     | PopupMenu<br>ListPicker<br>CheckboxBar                                   |
| <b>Actions</b>     | EventHandler<br>LocatorPane<br>ClickPane                                 |
| <b>Annotations</b> | Tooltip<br>Mouseover<br>Annotation<br>MouseAnnotation<br>MouseAppearance |
| <b>View</b>        | TabView<br>SlideView<br>MenuView<br>FlipView<br>OpenerView<br>PopupView  |
| <b>Files</b>       | FileNameSetter<br>SystemDialogInput                                      |
| <b>Dialogs</b>     | DialogInput<br>CreateDialog<br>MessageDialog<br>ChoiceDialog             |

# Interface Elements

- ❖ There is never only one way implement in Mathematica!
- ❖ Good enough trumps analysis paralysis
- ❖ However, aesthetics matter!
- ❖ The right controls AND the right layout AND the right option settings are key ingredients to success



# The DataModeler GUI

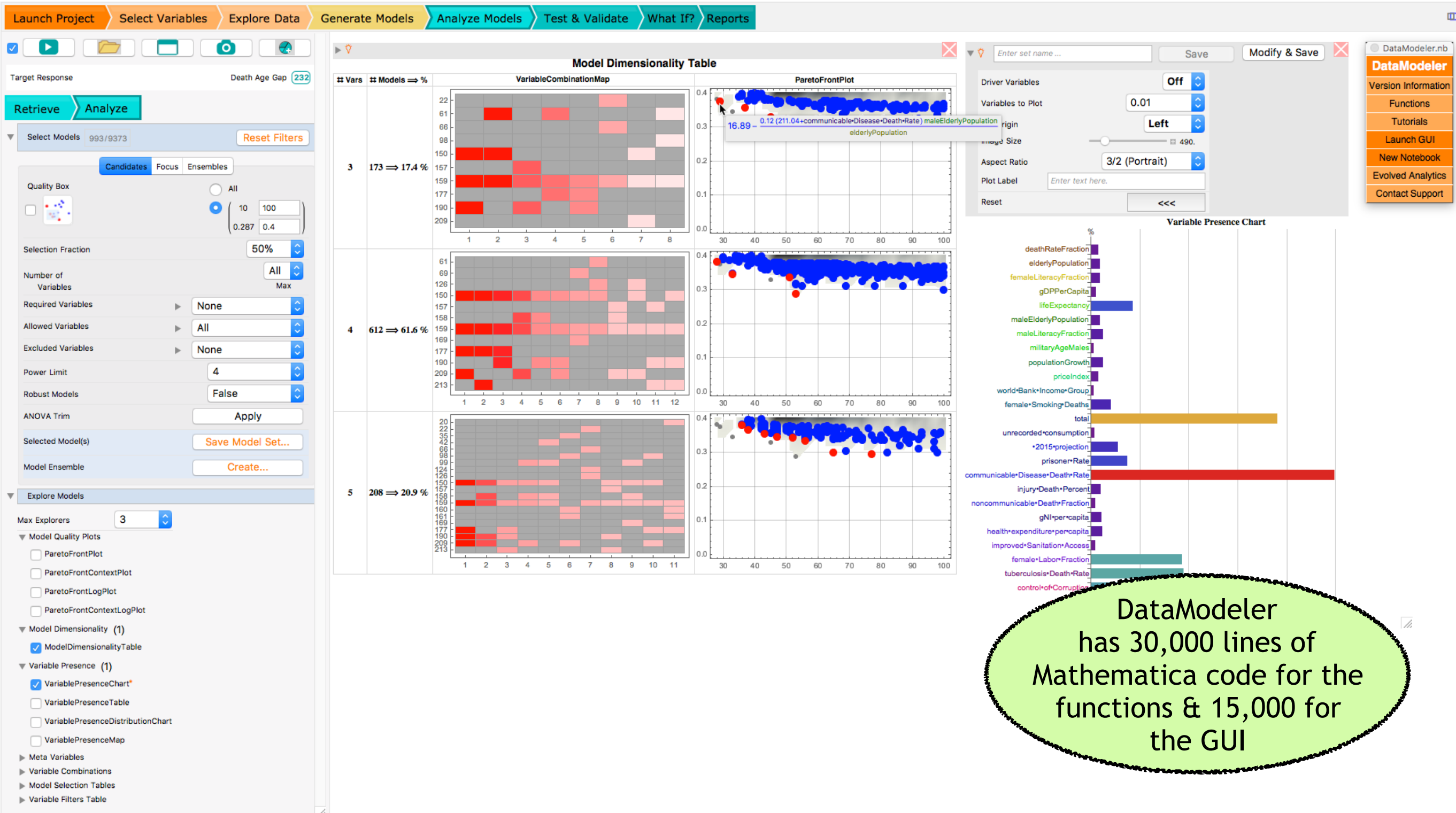




# The DataModeler GUI

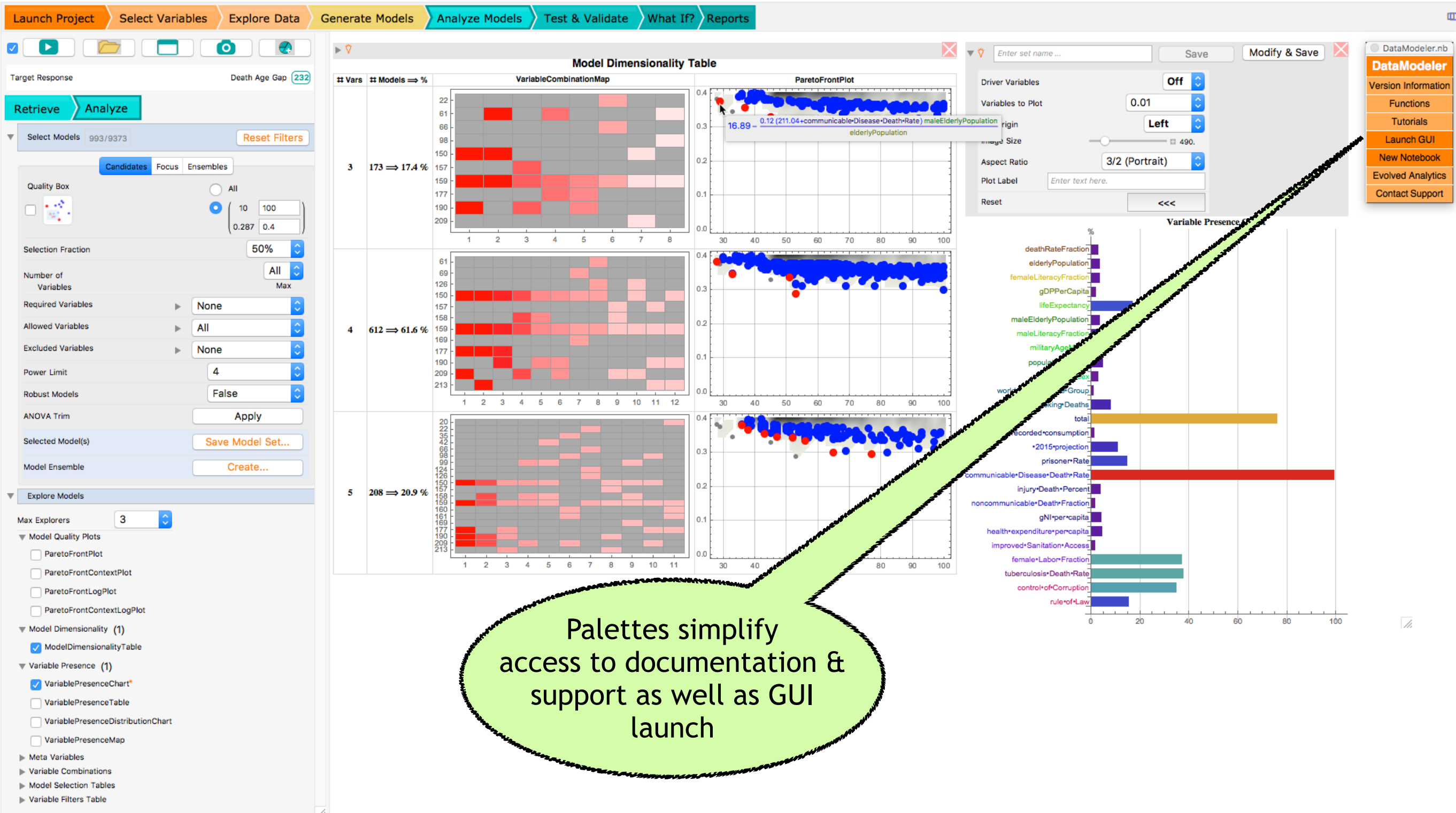


# The DataModeler GUI

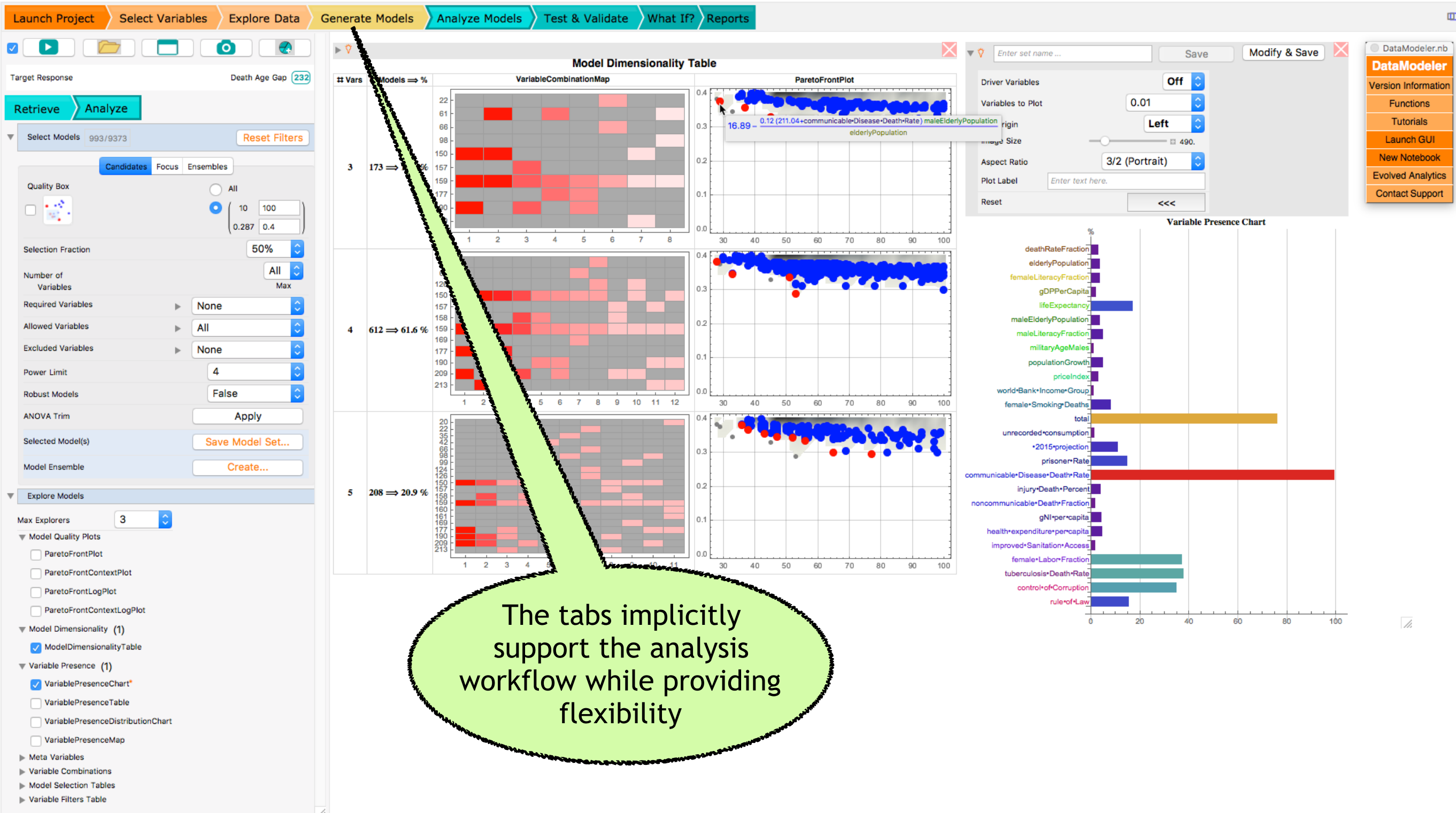


DataModeler  
has 30,000 lines of  
Mathematica code for the  
functions & 15,000 for  
the GUI

# The DataModeler GUI

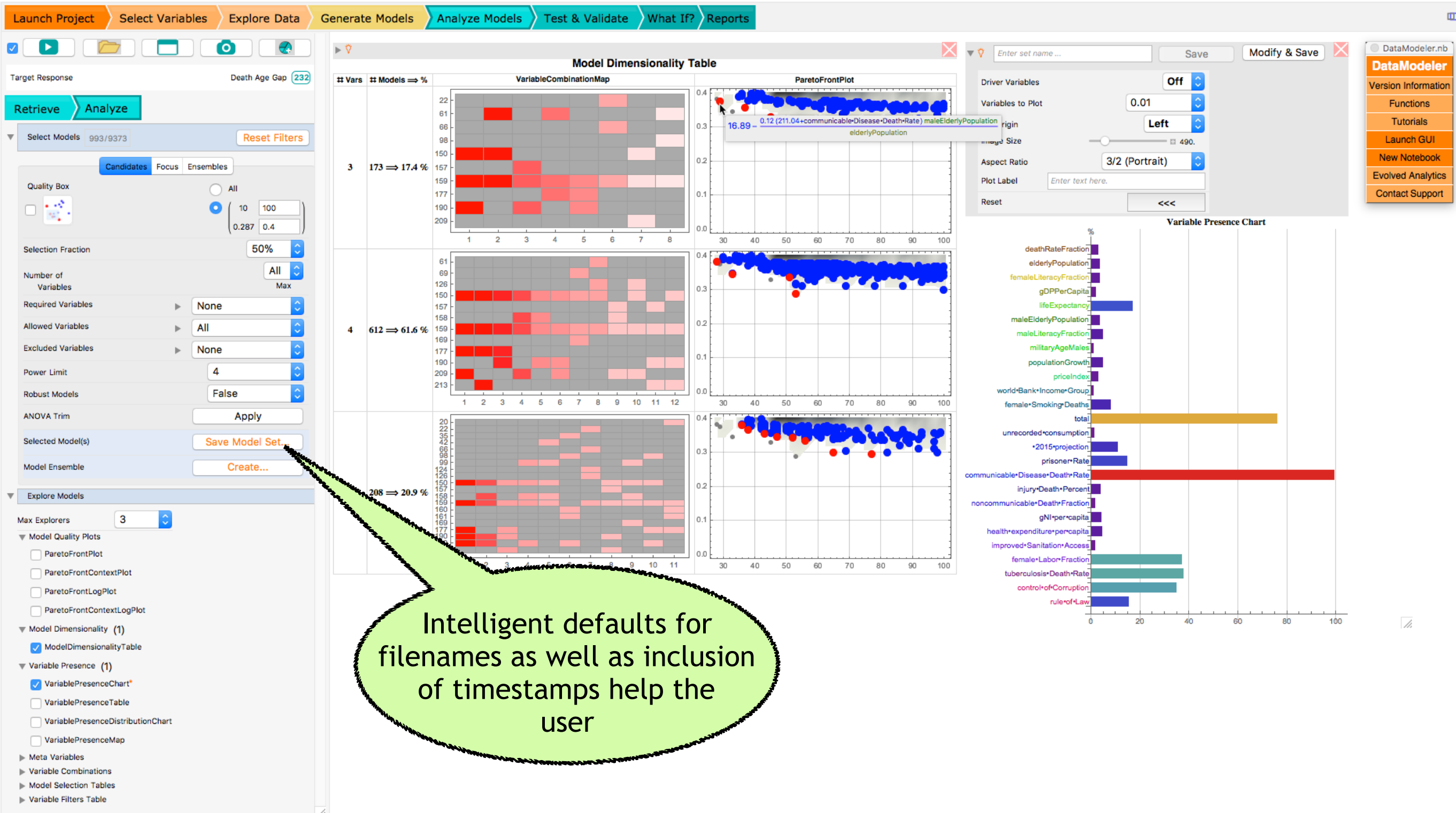


# The DataModeler GUI

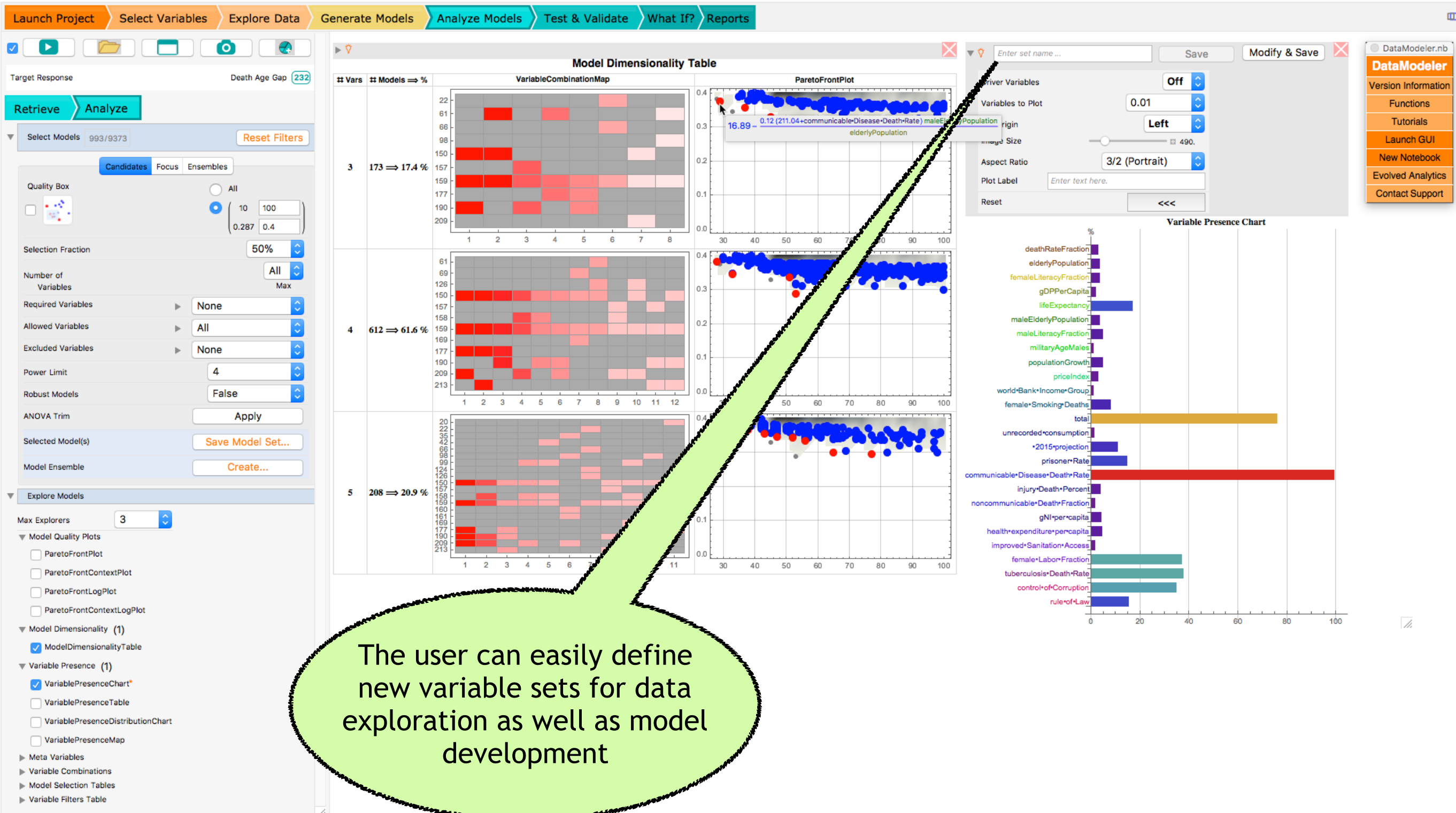




# The DataModeler GUI

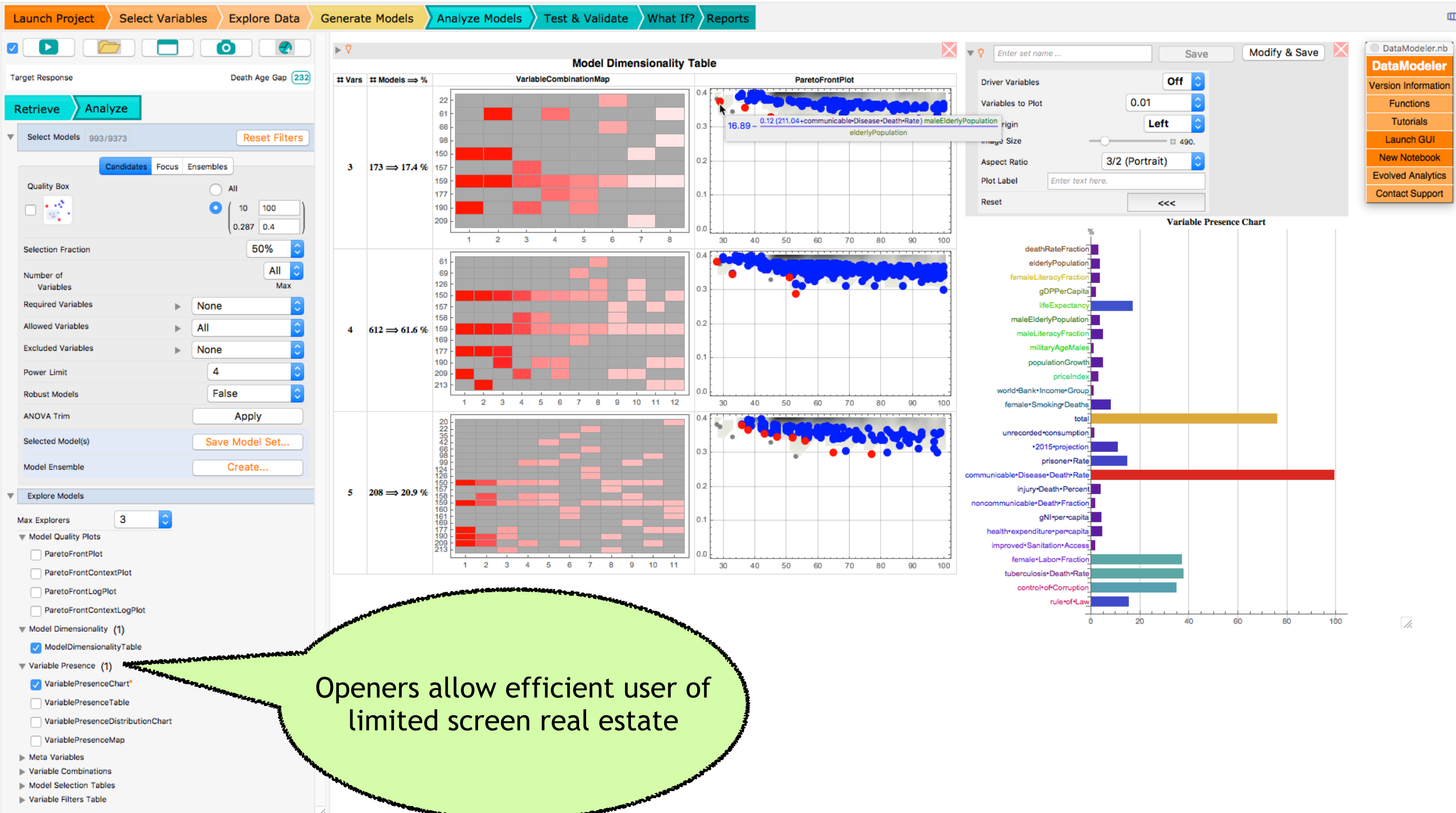


# The DataModeler GUI





# The DataModeler GUI



# The DataModeler GUI



# The DataModeler GUI

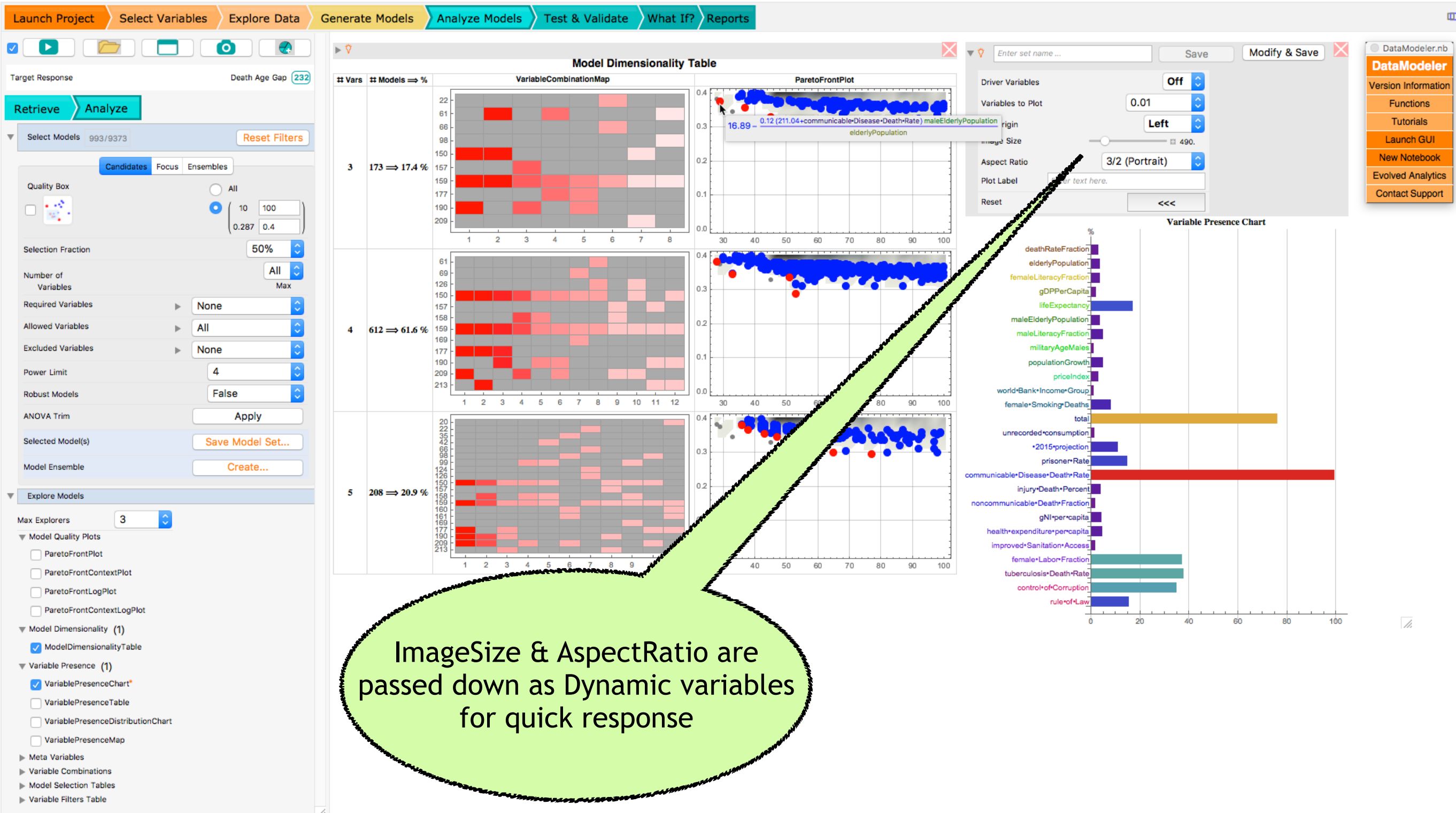


# The DataModeler GUI





# The DataModeler GUI



# The DataModeler GUI





# The DataModeler GUI

The screenshot displays the DataModeler GUI with the following components and callouts:

- Workflow Tabs:** Launch Project, Select Variables, Explore Data, Generate Models, Analyze Models, Test & Validate, What If?, Reports.
- Left Panel (Target Response):** Includes buttons for Retrieve, Analyze, and a list of models (818/5044). Callout: "Output can be captured for export or to include in reports".
- Quality Box:** Contains filters for Required Variables, Allowed Variables, Excluded Variables, Power Limit, Robust Models, ANOVA Trim, and Selected Model(s). Callout: "Intelligent defaults for filenames, locations and the use of timestamps help the user".
- Explore Models:** Includes a dropdown for Max Explorers (3) and a list of model quality plots (ParetoFrontPlot, ParetoFrontContextPlot, etc.). Callout: "Openers allow showing desired functionality when screen real estate is limited".
- Driver Variables Panel:** Includes controls for Driver Variables, Variables to Plot, Bar Origin, Image Size, Aspect Ratio, and Plot Label. Callouts: "ImageSize & AspectRatio are Dynamic for quick response" and "Controls for explorers are not exposed unless desired. Intelligent default behaviors are important!".
- Pareto Front Context Plot:** A scatter plot showing the relationship between variables. Callout: "The user can easily specify named variable subsets to focus the model development".
- Equation Editor:** Displays a mathematical model:  $5.54 - (9.57 \times 10^{-2}) \text{communicable} \cdot \text{Disease} \cdot \text{Death} \cdot \text{Rate} + (4.22 \times 10^{-2}) \text{female} \cdot \text{Labor} \cdot \text{Fraction} - 0.67 \text{rule-of-Law} + 0.16 \text{total} - 1.47 \sqrt{\frac{1}{\text{tuberculosis} \cdot \text{Death} \cdot \text{Rate}}}$ . Callout: "Palettes simplify access & application launch".
- Right Panel (DataModeler):** Includes Version Information, Functions, Tutorials, Launch GUI, New Notebook, Evolved Analytics, and Contact Support. Callout: "The tabs help to support a flexible modeling workflow".
- Callouts:**
  - "The tabs help to support a flexible modeling workflow"
  - "The user can easily specify named variable subsets to focus the model development"
  - "Palettes simplify access & application launch"
  - "Controls for explorers are not exposed unless desired. Intelligent default behaviors are important!"
  - "ImageSize & AspectRatio are Dynamic for quick response"
  - "Output can be captured for export or to include in reports"
  - "Intelligent defaults for filenames, locations and the use of timestamps help the user"
  - "Openers allow showing desired functionality when screen real estate is limited"
  - "Tooltips are heavily used to provide guidance as well as insight"
  - "DataModeler evolves models rewarding simplicity & accuracy. This identifies driving variables while simultaneously generating simple human-readable models. Here there were 230 input variables"
  - "A lot is possible in Mathematica when you move beyond Manipulate!"
  - "DataModeler requires 30,000 lines of Mathematica for the function set and another 15,000 to implement the GUI"



## Fiberglass Image Analysis System

Load Images

Image Review

Image Stack Explorer

Fiber Orientation

Solids Analysis

Fiber Orientation 3D

Color-based Analysis — Image Stack

Analysis Designer

3D Analysis Designer

Define Reference Color

Define Color Distance Threshold

Explore Component Morphology

Explore Intersection Angles

☒ Image Stack EnhancedImage Stack Layer: Swatch Size: 

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 32 33 34

Fully Saturated Pilot – Set Point 2/Bottom 3/bottom3

Set Reference Color

Color Swatch



Dominant Colors



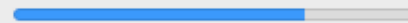
Reference Color



Lifesign to provide  
user feedback  
during long  
processing

**The Color Distance Image is Being Calculated**

2917200 of the 4080000 voxels have been analyzed



Each microscopy 3D image stack would  
produce 1,200 attributes which were,  
then, aggregated and fed into  
DataModeler for numeric analysis in  
complement with process variables.

5,700 lines of  
Mathematica code  
to implement  
analysis & GUI

Many (painful)  
lessons were learned  
in this exercise!



# Magic Number Designer

Save Design

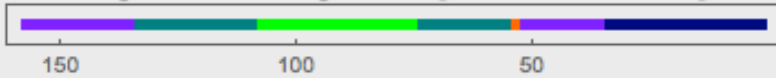
Retrieve Design

Restore Defaults

Reset  
buttons are  
good ideas!

## Skein View

Magic Number Length = 158 [34-19-21-34-26-24]



Rotate Skein  0

☐ Reverse Yarn Direction

## Yarn Colors

# of Colors

Yarn Length Yarn Data Swatch Data

Color Length Units: ☐ cm ☒ in.  
Yarn Length

| Color   | Yarn Length                 |
|---------|-----------------------------|
| color 1 | <input type="range"/> 13.75 |
| color 2 | <input type="range"/> 7.5   |
| color 3 | <input type="range"/> 8.25  |
| color 4 | <input type="range"/> 13.5  |
| color 5 | <input type="range"/>       |
| color 6 | <input type="range"/>       |

Controls are disabled  
if superseded by  
entries in other tabs

This version is 1,000 lines of  
code vs. the 50 lines of the  
earlier Manipulate version

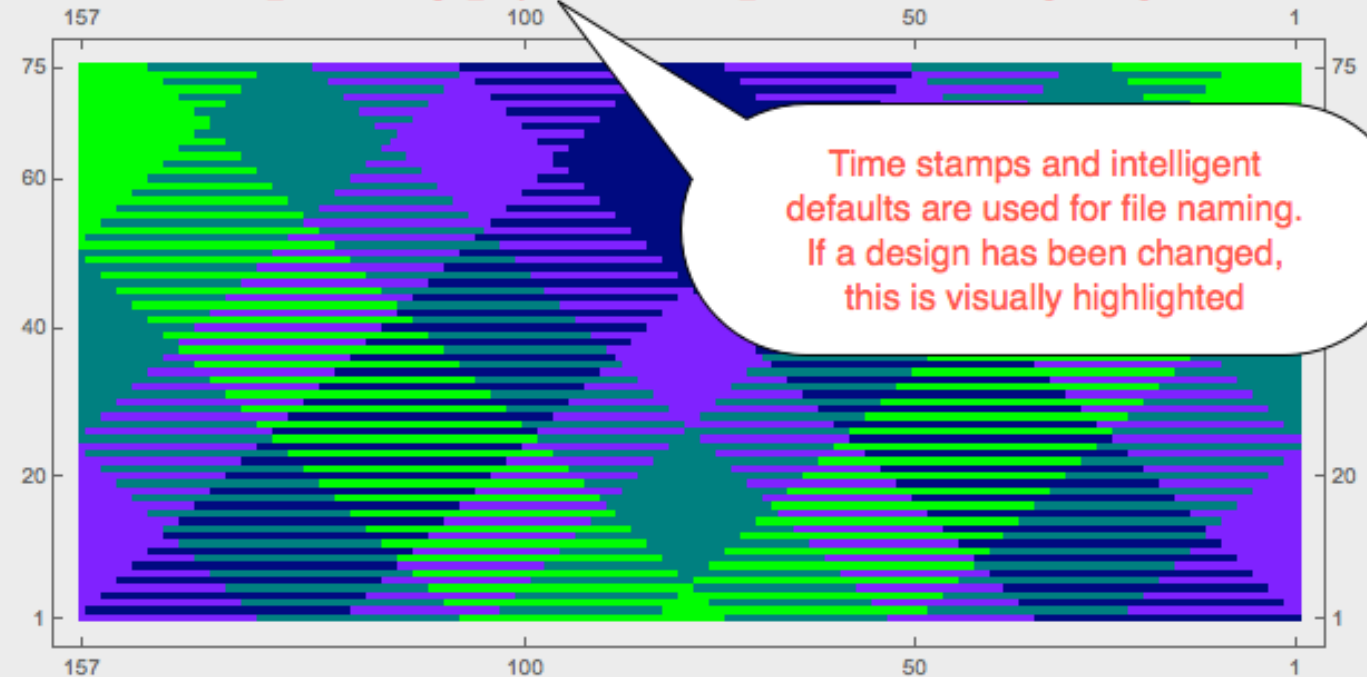
## Yarn Usage Rate

Yarn Usage Rate: 1 inch of yarn  $\leftrightarrow$  2.50794 stitches

Since the swatch data table has been populated, it is possible to calculate the usage rate directly.

Tooltips and FieldHints are  
used to provide guidance

MND\_Sweet Georgia\_Superwash Worsted\_2016-10-15-11-52.magicDesign



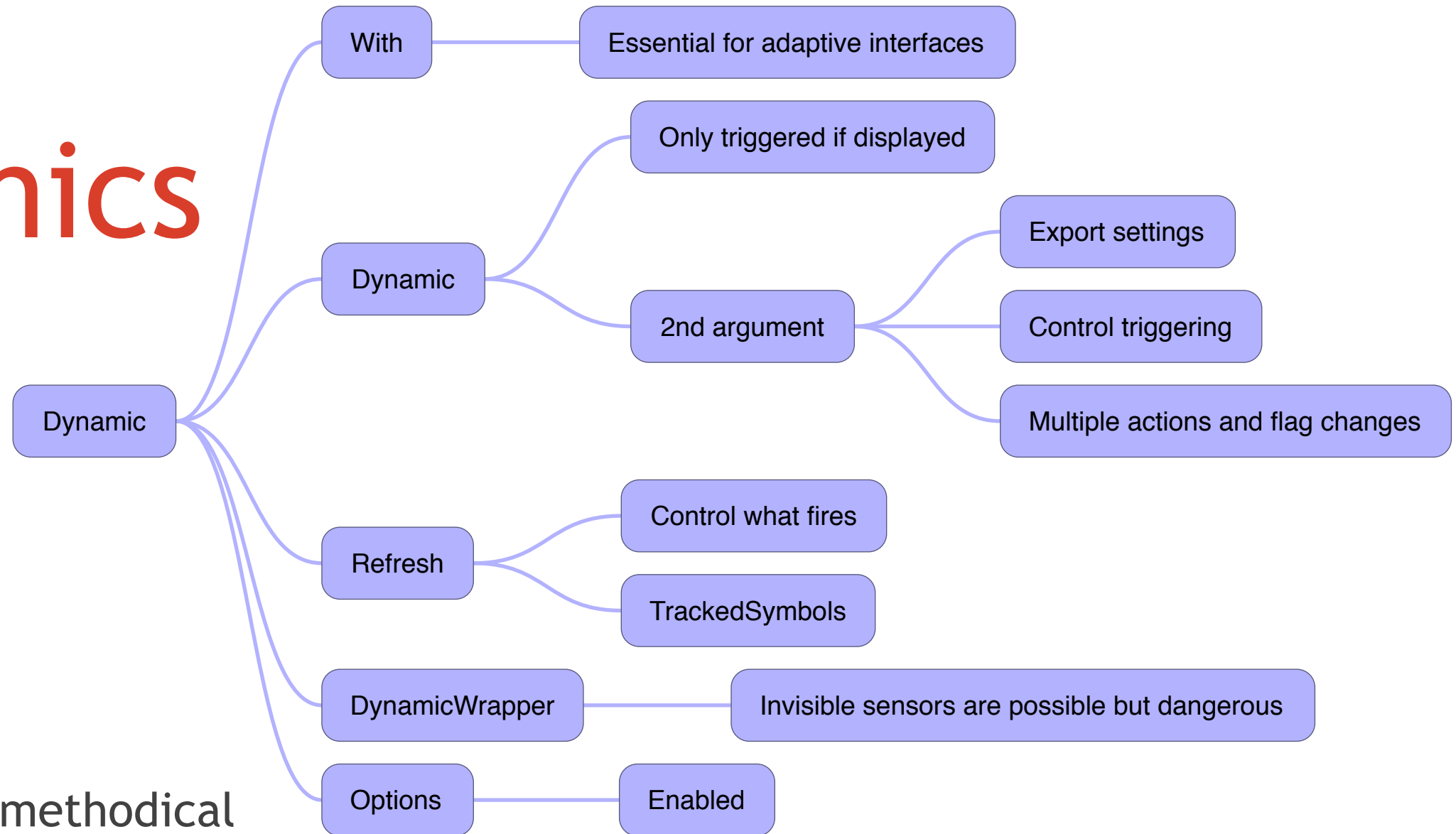
Time stamps and intelligent  
defaults are used for file naming.  
If a design has been changed,  
this is visually highlighted

Pattern Stitches: 157 wide \* 75 long  
Fabric Dimensions: 31.4 wide \* 15. long (inches)  
Yarn Used: 820.304 yards

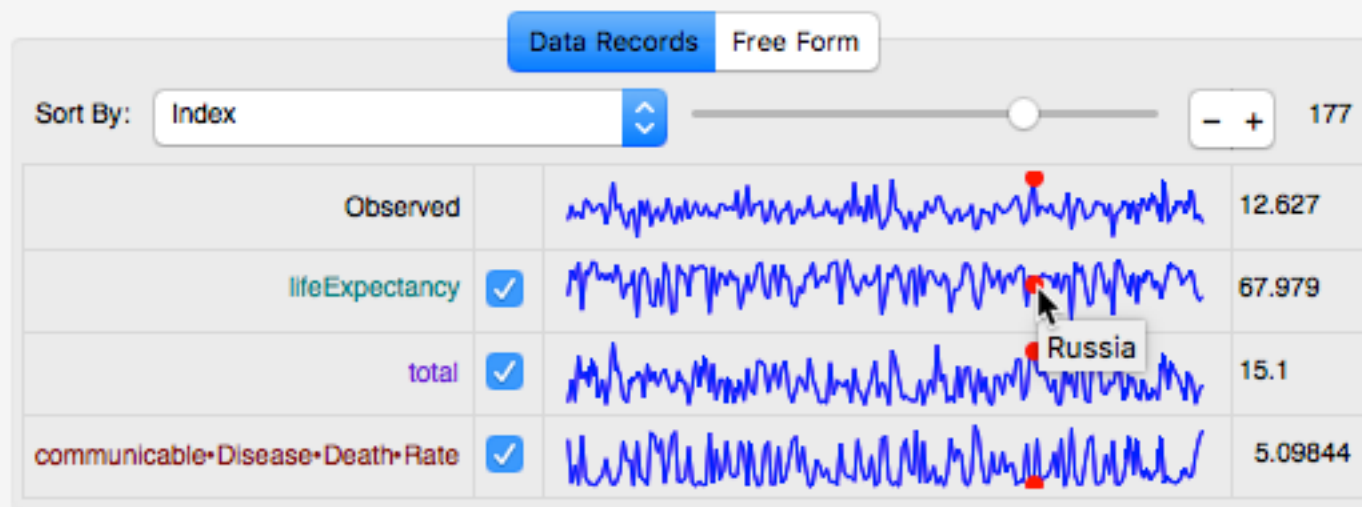
Enterprise Mathematica is  
needed to create CDFs able  
to read & write files (and to  
remove the Wolfram logo)

Image Size  600

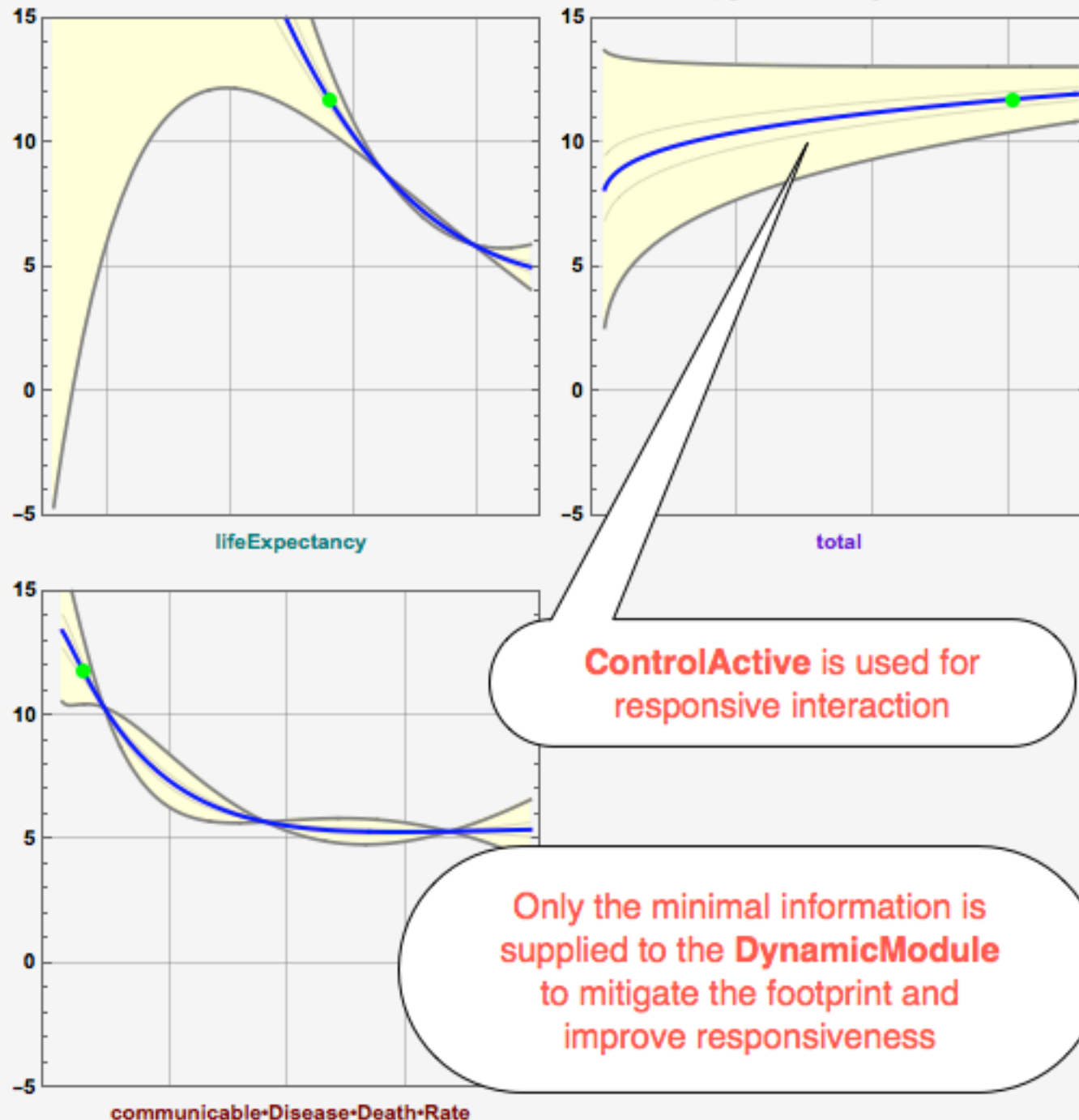
# Dynamics



- ❖ You have to be methodical
- ❖ Initialize ALL variables & have an initialization function
- ❖ Try to partition functionality & minimize interactions
- ❖ Do not **TrackSymbols** that are large (e.g., arrays or images) – use surrogates as triggers for updates. Counters also provide fine control of updates
- ❖ The 2nd argument for **Dynamic** is important to understand and embrace
- ❖ You WILL improve your ability to combine profanities in novel ways



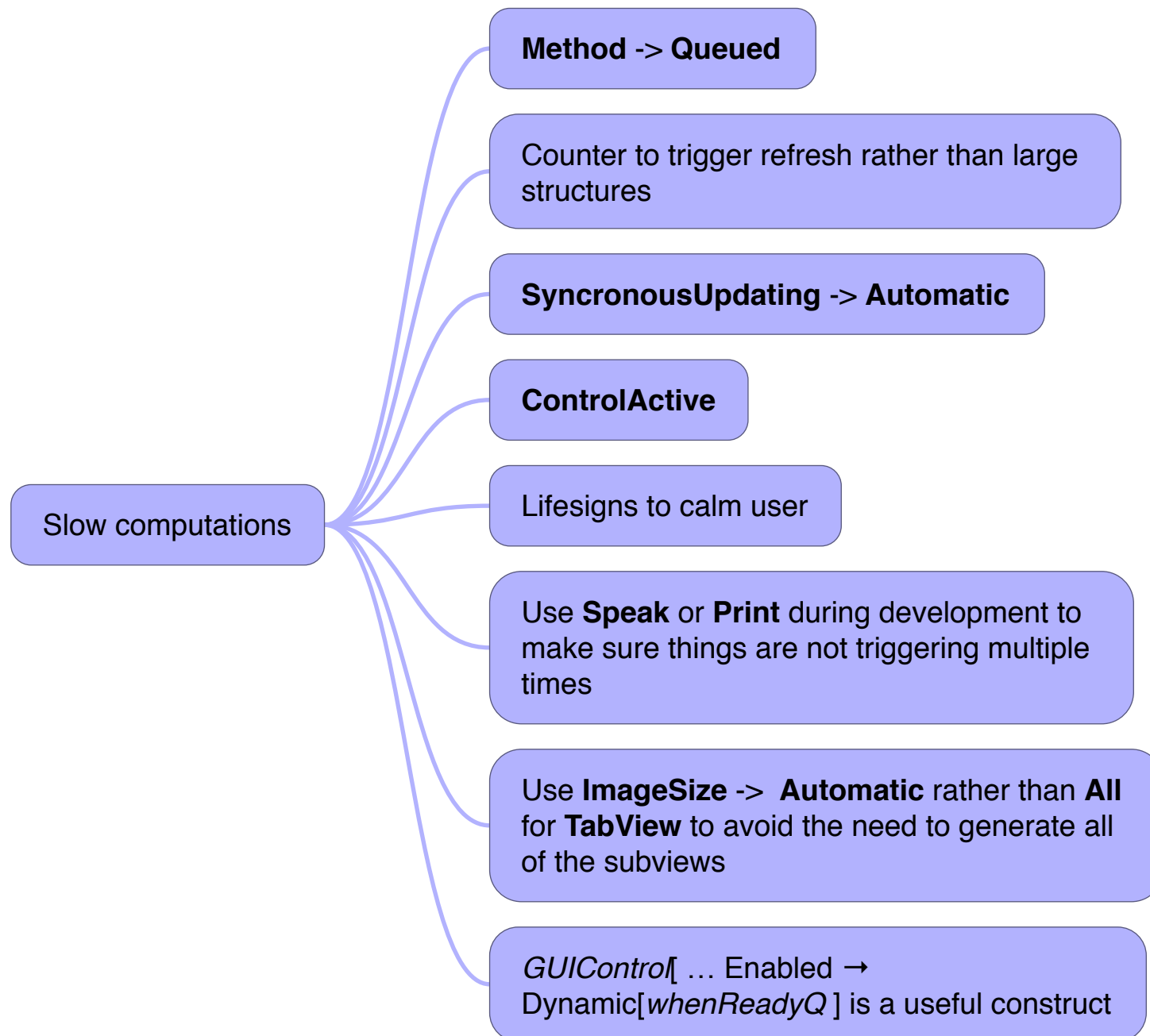
Rond 6 – 3 var ensemble (ANOVATrimmed) [ ref = 11.76 ]



# DynamicModule

- ❖ Do NOT nest **DynamicModule** functions (if it can be avoided)
- ❖ variables are maintained by default which can blow out memory footprints and hammer responsiveness
- ❖ If nesting needed, nonlocal variables and the **UnsavedVariables** option are useful
- ❖ Minimize the size of data transferred into **DynamicModule**

# Slow Computations



- ❖ Management of slow computations needs to be explicit to avoid timeouts on the part of Mathematica.
- ❖ Users need feedback that the system is alive and to keep them from clicking vigorously – which causes more problems!
- ❖ Synchronous vs Asynchronous updating is a big deal!



# Lifesign Panels for CDFs

```
CreateMonitorPanelMA[title_, templateString_String, vals_] := CreateDialog[
 Column[{
 LabelFormDM[title, FontSize -> 14],
 StringForm[templateString, vals],
 ProgressIndicator[#1 , {0, #2 }]&[vals],
 " "
], Center, 1.5]
];

Switch[showProgress,
 True, (
 Block[
 {mnbOld, mnbNew, fileIndex, numFiles = Length@fileNameSet},
 imageSet = {};
 fileIndex = 0;
 mnbNew = mnbOld = CreateMonitorPanelMA[
 "Image Stack Retrieval",
 "`1` of the `2` 3D images have been retrieved",
 fileIndex, numFiles
];
 imageSet = MapIndexed[(
 fileIndex = First@#2;
 mnbNew = CreateMonitorPanelMA[
 "Image Stack Retrieval",
 "`1` of the `2` 3D images have been retrieved",
 fileIndex, numFiles
];
 NotebookClose@mnbOld;
 mnbOld = mnbNew;
 Import[#1, "Graphics"]
) &,
 fileNameSet
];
 NotebookClose@mnbOld
],
 imageSet = Import[#, "Graphics"] & /@ fileNameSet
]
];
```

- ❖ Dialogs can be updated in Mathematica; however, such doesn't work when exported to CDFs
- ❖ One workaround is to create a new status panel over the prior one and delete the old one from behind
- ❖ This causes some flicker so it is suboptimal, albeit, functional

# Other Issues

```
x = 27
```

```
27
```

```
InputField[Dynamic@x, Number]
```

```
InputField[Dynamic@x, Number]
```

```
Dynamic@FullForm[x]
```

```
George
```

- ❖ Entering a nonnumeric into a numeric **InputField** will not display an entered nonnumeric; however the variable will still be defined!
- ❖ The 2nd argument to **Dynamic** can be used to trap this situation
- ❖ Many **Tooltips** are dangerous for the front-end
  - ❖ Limit the number of Tooltips simultaneously on-screen
  - ❖ Use the **TooltipDelay** option or bad things will happen
- ❖ **Workbench** is only marginally maintained
- ❖ Need an iPad solution

# Notebook Creation

```
 "
 ImageSize -> imageSize+50,
 Alignment -> Center,
 Background -> Lighter[Gray, 0.85]
],
 TrackedSymbols -> {
 numColors, segmentColors, segmentLengths, cy
 rowsPerGaugeUnit, stitchesPerGaugeUnit, gauge
 foldLocation, showBothSides, fileName, designMk
 }
]
}}, BaseStyle -> {FontFamily -> "Ariel", FontSize -> 12},
]
, SaveDefinitions -> False
],
WindowSize -> All,
ShowCellBracket -> False,
WindowFloating -> False,
Editable -> False,
Deployed -> True,
Saveable -> False,
WindowElements -> {"VerticalScrollBar", "StatusArea"},
PrivateNotebookOptions -> {"PluginToolbarEnabled" -> False}
]
```

- ❖ CreateDocument[  
guiDesignDynamicModule[ ... ]  
opts]
- ❖ The big one here is Saveable  
→ False so the user is not  
prompted when they close the  
app

# Deployment

- ❖ Mathematica/CDF
  - ❖ easiest path but overkill for many users
- ❖ CDF Player Pro
  - ❖ Pay per deployed system — ~\$200/seat
- ❖ Enterprise Mathematica
  - ❖ 10% of revenue (minimum of \$12,000)
- ❖ Cloud
  - ❖ Have not explored

# Best Practices

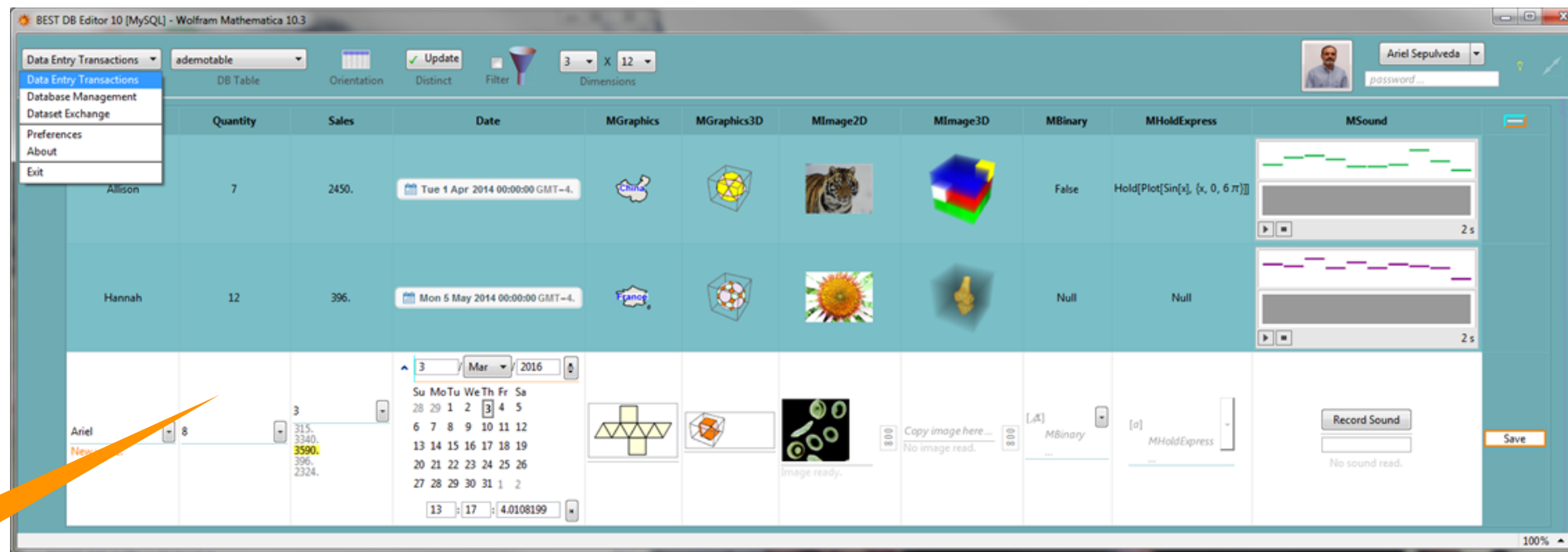


- ❖ Layer the complexity to keep the visual foundation clean & insightful
- ❖ Develop functions for common operations — e.g., the DataModeler explorer options
- ❖ Avoid ugly
  - ❖ use color judiciously
  - ❖ gridlines don't have to be black



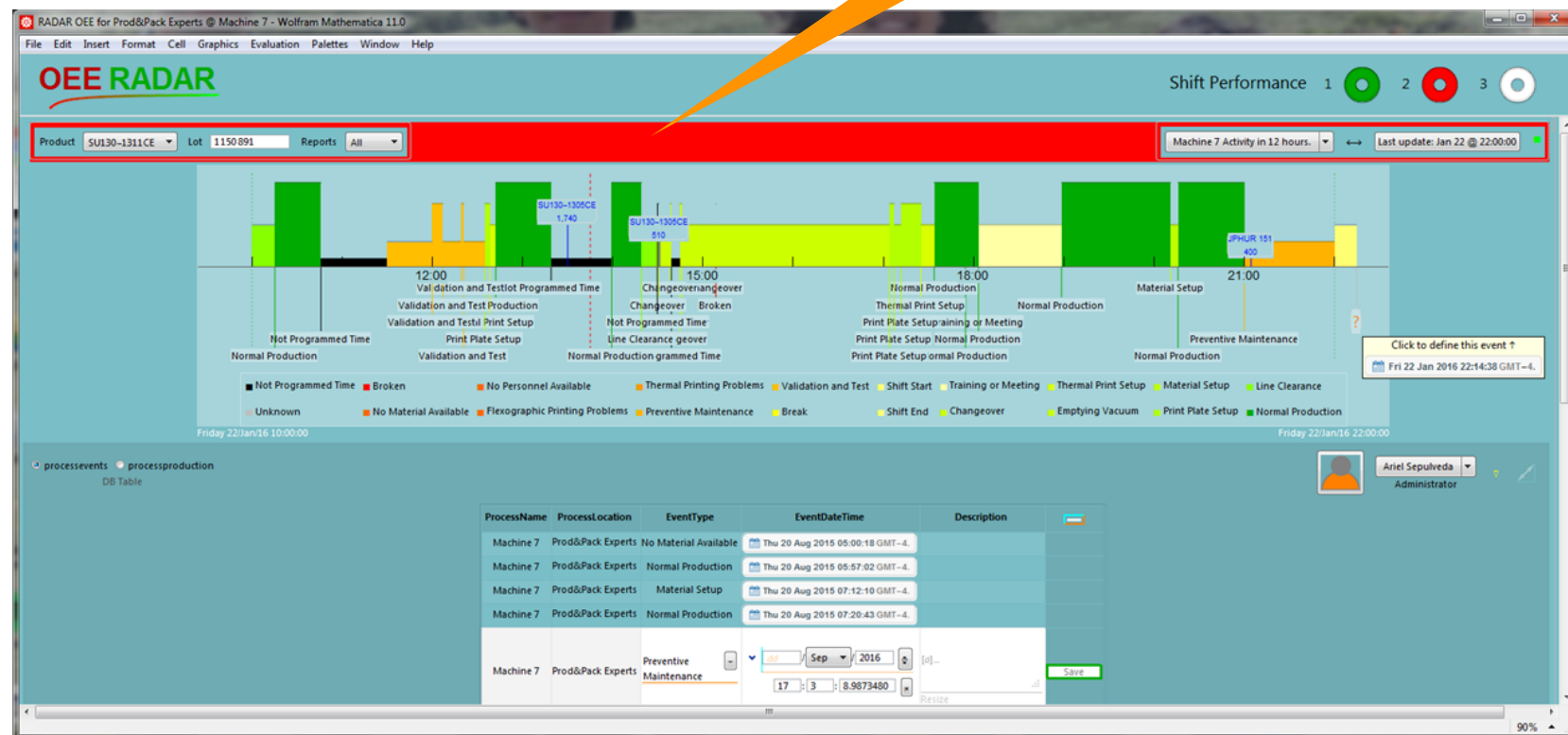
# BEST Practices Implemented

BEST DB Editor



More Info: [ariel.sepulveda@prontoanalytics.com](mailto:ariel.sepulveda@prontoanalytics.com)

OEE RADAR



- ❖ BEST DB Editor is a database editor which can be used to manage or edit database systems and tables.
- ❖ Works on Windows and Mac for database systems like MySQL, SQLServer, etc.
- ❖ Can be used to work with many data types like text, dates, numbers, Graphics, Images, Sound, etc.
- ❖ Can be used as a sub-component of other Mathematica GUIs like OEE RADAR which is a real-time data capture and analysis system.

```
Magic Number Designer 8.cdf
(* Content-type: application/vnd.wolfram.cdf.text *)

(***) Wolfram CDF File (***)
(* http://www.wolfram.com/cdf *)

(* CreatedBy='Mathematica 10.4' *)

(*****
*)
*)
*) This file was created under the Wolfram Enterprise licensing terms. *)
*)
*) For additional information concerning CDF licensing see: *)
*)
*) www.wolfram.com/cdf/adopting-cdf/licensing-options.html *)
*)
*)
*)
*)
(*****)

(*CacheID: 234*)
(* Internal cache information:
NotebookFileLineBreakTest|
NotebookFileLineBreakTest
NotebookDataPosition[1064, 20]
NotebookDataLength[160530, 3097]
NotebookOptionsPosition[160923, 3087]
NotebookOutlinePosition[161523, 3112]
CellTagsIndexPosition[161480, 3109]
WindowFrame->Normal*)

(* Beginning of Notebook Content *)
Notebook[{
Cell[BoxData[
DynamicModuleBox[{MagicNumber`Private`defaultSettings$,
MagicNumber`Private`yarnColorFunction$, MagicNumber`Private`cycle$$ =
False, MagicNumber`Private`reverseYarn$$ = False,
MagicNumber`Private`skeinRotation$$ = 0,
MagicNumber`Private`patternShift$$ = 0,
MagicNumber`Private`startLocation$$ = 0, MagicNumber`Private`widthRepeat$$ =
1, MagicNumber`Private`numColors$$ = 4,
MagicNumber`Private`yarnLengthUnits$$ = "inch",
MagicNumber`Private`maxColors$$ = 16, MagicNumber`Private`segmentColors$$ = {
RGBColor[1., 0.499474, 0.],
RGBColor[0.921184, 0.6542725, 0.0297604925],
RGBColor[0.7892615000000001, 0.6147505, 0.2668325],
RGBColor[0.8161955000000001, 0.41711224999999996, 0.62417325],
RGBColor[0.9952513749999999, 0.5463053124999999, 0.000994424375],
RGBColor[0.881750625, 0.667947375, 0.0622695625],
RGBColor[0.7719553125, 0.5767103125, 0.3573026875],
RGBColor[0.850505125, 0.3719416875, 0.6759001875],
RGBColor[0.99050275, 0.593136625, 0.00198884875],
RGBColor[0.8425987500000001, 0.66271225, 0.11674487500000001],
RGBColor[0.779648125, 0.523144, 0.45437249999999996],
RGBColor[0.885300625, 0.34233825, 0.6915035],
RGBColor[0.9608285, 0.626415125, 0.013726104375],
RGBColor[0.8065676875, 0.6527906874999999, 0.1763623125],
RGBColor[0.7015074375, 0.46600, 0.55254225]
}
}]]];
```

# Intellectual Property

- ❖ The default for a CDF is to store a plaintext version of the code
- ❖ The solution is to **Encode** a package file (optionally, tied to a **\$MachineID**)
- ❖ The GUI uses a DynamicModule with Initialization → Needs[“encodedPackage`”] to retrieve the encoded IP
- ❖ Reading & Writing files is restricted to CDFs created with Enterprise Mathematica or PlayerPro



# Overview

# Script the Encoding

The sequence of activities to create a *DataModeler* release package are:

- 1. Make sure the `$VersionOfDataModeler` variable in the `SymbolicRegression.m` source code is accurate and **update if needed**.
  - 2. Duplicate the `# DataModelerWorkbench/DataModeler/` folder structure and within a top-level folder as `DataModeler/` with a time-stamp suffix on the top-level folder
  - 3. ...
  - 4. The **Attributes** of the symbols created within package files should already be set to **ReadProtected** and **Locked** as the final cells within the notebook (and the `SymbolicRegression.m` file should be automatically created)
  - 5. Copy the installation GUI and readme files from this folder to the release folder and rename the latest version of the `DataModelerUIDate.m` file to be just `DataModelerUI.m`
  - 6. Rename `init.m` to be `initSource.m` and execute `Encode[initSource.m, init.m]` to create an encrypted version of the `init.m` file
  - 7. Rename `SymbolicRegression.m` to be `SymbolicRegressionSource.m` and execute `Encode[SymbolicRegressionSource.m, SymbolicRegression.m, key]` to create an encrypted and password-protected version of the critical symbolic regression functions. The `key`, of course, should be the same as that embedded within the now-encrypted `init.m` file
  - 8. Do the same for the non-`SymbolicRegression.m` source files and `Encode[ ]` them. Rename `other.m` to be `otherSource.m` and execute `Encode[otherSource.m, source.m]` to create an encrypted version of the `source.m` files. Then delete the `unencoded` source files
  - 9. **NOTE:** The `filenames` supplied to `Encode[ ]` must be different. If they are not, the original file will be destroyed — but a functional replacement will not be created.
  - 10. Copy the `SpreadsheetViewDM.m` file over to the target file since it is already encrypted, it will not need to be re-encrypted.
  - 11. Remove all `*.nb` files in the `DataModeler/` folder structure with the exception of those within the `DataModeler/Documentation/English/` subfolder.
  - 12. Remove the `initSource.m` and `SymbolicRegressionSource.m` files since they are in plaintext.
  - 13. Remove all files & folders within the `DataModeler/Documentation/` path which are NOT within the `English/` folder. (We are making an implicit assumption that the `BrowserIndex.nb` and `BrowserCategories.m` files are up-to-date and there are no superfluous files within the `English/` subfolder.)
- At this point, the created folder structure should be safe for release. We still have a couple of manual steps to create the release package
- 14. Zip up the `DataModeler` folder using `CleanArchiver` with settings so that the Mac OS X hidden files are deleted.
  - 15. Send out the update.



```
(*!1N!*)mcm
]rk;rDC6H&($a**wL/ix[R3Vj-0nuYJ@3t1EJj[1"#<[5Sp(6YEa(#6LL3Z[B~]]$3`Q@-
$| $Atb%]n;Z]GZVG-%$'m;!(JAMyE$a`n<[GJ+VR%p~Gk(.d@'1tb%]j)M+fCkd]\@o6]
blEs-YcyL,d;"1tb%]j)M+fCkd]\@okB~0-0R(-Eq5:~!A^t|taGiBjJz` ;U%;:L%DJa1
m8!(JAMyE$a`n<p|Ve"%?:2@(-Eq5:~!A^t|taGiBj!qN'hrbP*^oML\h|L0?`5:~!A^t|
taGiBjuEV0.bSTL(dwA1N!"#qH^K^50b(]\@oG.=ceeLM!LA#5Ef0iy]|"#qH^K^50b(]\@o
G.^$A^t|taGiBjuU@,$'m;..JuDq(GJl` ;)y(b5EsN.h&2ba)rp(6YEa(#6l!(C!eda{\
PFq2p'h$K&F>r#LRz1^H9D4tas2V]N9G1 H6lVI-%>wpK.h&2ba)rp(j)N-,ZfP!b$|M:
T6EI4!Zt!QTS5J?A<E6j#u#^LEN"bFL(p%>V;bI-96Si5Z%}MY6\:+e8!07,i !NL"!!\5
=+lH082>.)?B<E6j4&]5J@MyPB61s+JI'DN4lK$xs/+<4~fPij2q.kb(!)lSl'|DN@bna
0`cc-/m)qA e0Q*bC(p(IhV`-pCYi?'Dl/khtNBe"GM:PF;<=](<1.6th7?SL{l53*!6cB
d=Stop6Y 4-c^du!UE#x/c!S.b%CapDc)o(">)e\@n+:n_&pnS!|^&fHGZsT4N.UC*oU
(=pRZzh>a\9H!ckf^=B$5a$'o5ZI*#0Z;<n1'3U&0p!ot/\n.5U&=R`2@>YNjtbcp$/\P
b,"0^eoQ9UP4V:J.D)?>@>YN=C8$XH#u@|E!&em='A]i;2oX_krD^#RQG",T!gX7;-_!uw
6X:+C!QCE#b)N%Utc}a<um*?n1'3U&0p!ot/\n.5U&=R`2@>YNjtbcp$/\Pb,"0^eoQ9U
P4V:J.D)?>@>YN=C8$XHI[Xfw{k&:A+lCA@[cu^;g`2MZ(v'^qNj# `V#u@|E!&em='A
]i,,M41ibh0khhfHk(/=,4m<C; }EU#-[`q<0_h|Jn-sk46l0+v.?HbP!0>)DCEZ\rdVI6
uf$2+1:ycH<DH 't+PPM$W19@ BnuuiA10ps05m8t[(E6BU$PZXip /i7!2LmclYoZkW[q
EK@#<JTkAa2lpY8(H#TZuVsW?b h`Tc5b SA@!(%@@#FE V"/$ =)e~42S9u%'.+o@j]8
&q<M6d.^`j89e]RR%`NjEg&&>.5:pk@XqS<[0?LthU;I;_01E-Em#H Xhq`y@Bj:k[&p!1
K=.G3z)^JvoNhHd7)H8A($` ;Gk@6d>a:*wL/ix[R^aPe\u791EA,%|D_k{MU9de44U,3&~
RZ7X0woj6T[`reA%^(5D0Mk)\:.`#dSahm!Ro[6\iS9'D@TF_!6B8iDB,X@E$"/[s40o6
?@*d@Ra9Td^F GU0fWKsBe{pTAABg[8Fb[PKYgyd7=Bo0-0B-[8uj#}27f7PFq2p'h$Tu
sTJ=g`.m^%A^t|taGiRzZ5DLKBRGWKD]tRaWb]@&q"3>gn@E$j=7Ct`M0L2dY$@s/+<]{
$!VQ(+YYB~#cR4XpMU`@Qn[fa$V]<FP?R'6<|`p%0(\DNrljd>F^Cu,k;T6syf8H)CFR?
B^?vK0+ 19h|oC_4\mFPt`'A27q^V;p3fld>50b(]\@oG.kQs"rw?v[cluoWUyVao[1{S*
Mm%C<X/]e>c2l@p1!#]0C ('rU6</tVn$<d4e~\@a*H!ewayp-]0:-d<e~hLM`/'SY6^;8
W{[{JAG}#93>>=^L%43JD&1k$><d7e~MqA+A\,bLaT6rze{.`,"d<e~&d[N`#]sRa[fa$
e0h7h.o[t~t7fI7,3Xi-50ZeP(.N5<EX#-[`q<0_h|?S\Yj)q26;uV@=DWUh0>a211R2\0
e00.d#WDTiuLm|.Tik3*(\?K<"JqM*6PL$|;aiyM4kC0&MY^LNgs+$i"/R}S4`Ur$V,Ur
$| cS-h0^MqA+Vjt7Y8`S6T:+86"N-pL$,T.4+E9qYNjtbcp=[jx%k0~\}$3`Q@-$|-j^V
3v=0Ko.Go6Zij<WK;-_!uwFVGbocr\3iq|uKD99x&pdo]\a0fC(7Gb\usWMZ(h&h6n^a
0hRFFQ>qZ7j#GZt_;q=quaJ@p1`Pd6TQ6ToK`,6AD@GhlnLY@H?&&f'JGbK6IY\=X}&j72
!g\R_EMU.13v0h@ @sS; 6mnSx*!baPe\Q0!oKdB+zm:v2e06U[`reeUY0QuZr`X;3Q]:9
K)R|[fa$e07fa1onk6&:A+5|$,1IqB;Q@.!3!TSDMoB^f)ZcelIH7/h6SH7/JZ0XWMrDta
$m6B~8F6&4bq`6,Ub6Ky(8Bra%Wlm<!nN'hrbPE1sk3Ua@Pm9uJ>_ rD*_HZNDu5n?V`
)kHFKw_"(V,em1Lgt#dCe~'ASff.5%NL$/27J?`l?Hc>tt"Kf9PF89e^!;c,s!$.6^Hg/T
&lR-W2.9k4*FIR:9$zT3c]jqfLPfTs$wV<*W[>#PU~myMjr0e]'ASff.SBM=_.$]f)FA1
mro68Z 08A:+E#],+x&&A3I|` ;r$NA,]jstn9feuL~GbS5M{qZkA4#r&5bBsACQ:<KFZ\Z
JaK6 AEB^cM 7?7R-^nesxP2okk@i;h6<MJXnD&_b&Yyu17Cl<F%XBg2FZ\ZJa=<(/,{@
[@*b*B:fl<F%$s=g/0L07^a @^Kzuw<,X0Fo;11#Kja>!_uZX97e.K5<JM?I]d-'c%aQ]
bBl+Znj060d-'c%aQ]bBl+eYjRN'i!_42whl5yJ{Y1gB3GN!'6lDa.a"#)t~:=G^kGZpA
_J2whl5yJ{Y1gBZlr$GZD\E]Kwh7\=56#4h7\=56#4-R%CRaM=506|oN$~=j4bDse1;3Q]
-ynzTn\@eo?6?D3S;tg_3H6AuXtaGi\ 0DD9h6Q65u&:qp0U7+Vf)kv0?HbP!03>SrBi"G
M:P/q:&q~%M!#n|uU\ML>agc2sgNS&~EI4!EgmrFmkpn90Wefd@;2:r6BN0b!%x$w~%S*
Lw1Fq\00?Wt #FFmKs1K0&ks0Wlhu7efP16V0[1"0Wlhu7 05<reMP+YNNH0R(NMH0R(
```



# Conclusion

- ❖ I love GUIs
  - ❖ I hate GUI development
  - ❖ Being disciplined can minimize the pain
- 
- |                                                                              |                                                                                                |
|------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| ❖ Mark Kotanchek                                                             | ❖ Ariel Sepúlveda                                                                              |
| ❖ <a href="mailto:mark@evolved-analytics.com">mark@evolved-analytics.com</a> | ❖ <a href="mailto:ariel.sepulveda@prontoanalytics.com">ariel.sepulveda@prontoanalytics.com</a> |
| ❖ <a href="http://www.evolved-analytics.com">www.evolved-analytics.com</a>   | ❖ <a href="http://www.prontoanalytics.com">www.prontoanalytics.com</a>                         |