

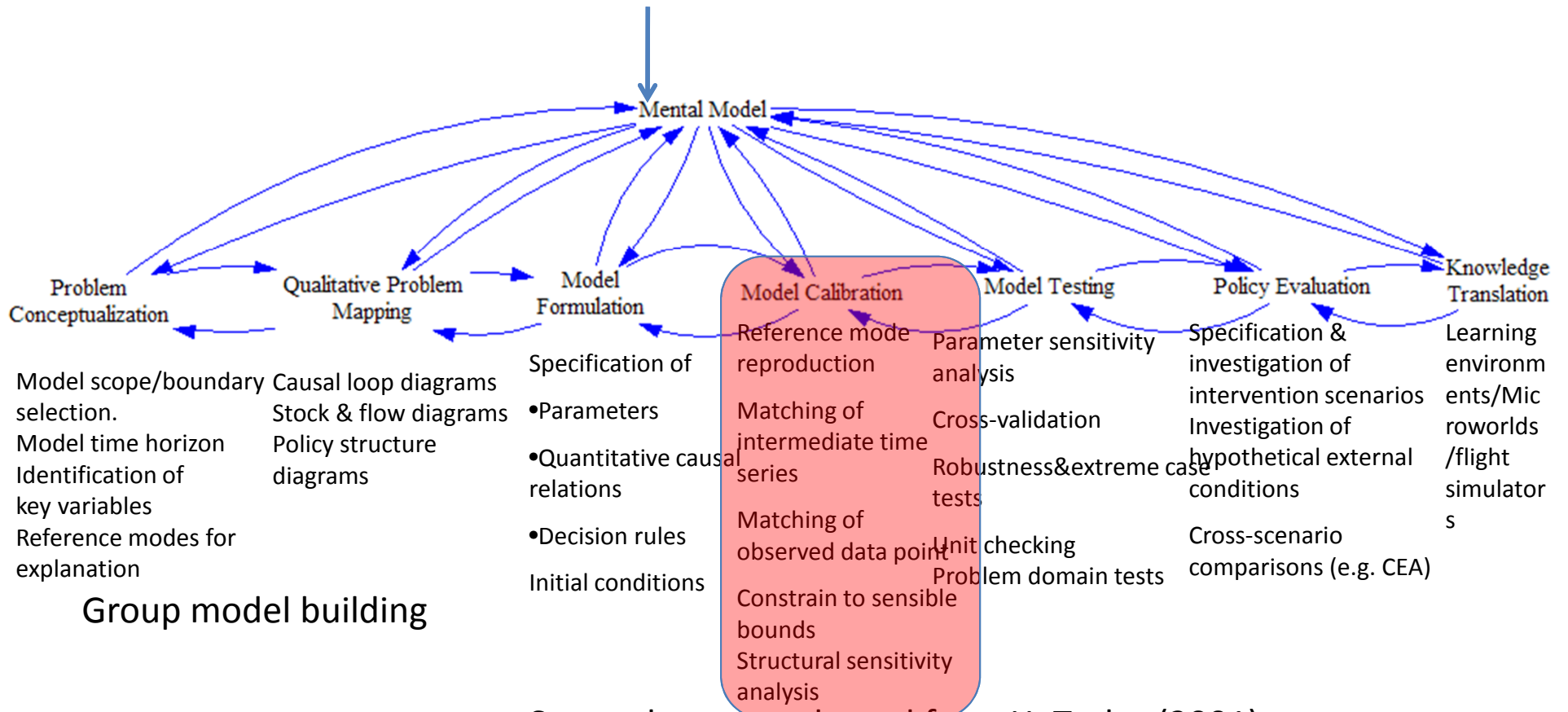
Dealing with Data Gradients: Calibration 2

Nathaniel Osgood

CMPT 858

March 31, 2011

A Key Deliverable!



Group model building

Some elements adapted from H. Taylor (2001)

Recall: Dealing with Data Gradients

- Often we don't have reliable information on *some* parameters, but do have other data
 - Some parameters may not be observable, but some closely related observable data is available
 - Sometimes the data doesn't have the detailed breakdown needed to specifically address one parameter
 - Available data could specify sum of a bunch of flows or stocks
 - Available data could specify some function of several quantities in the model (e.g. prevalence)
- Some parameters may implicitly capture a large set of factors not explicitly represented in model
- There are two big ways of dealing with this: manually “backing out”, and automated calibration

Recall: Calibration: “Triangulating” from Diverse Data Sources

- Calibration involves “tuning” values of less well known parameters to best match observed data
 - Often try to match against *many* time series or pieces of data at once
 - Idea is trying to get the software to answer the question: “What must these (less known) parameters be in order to explain all these different sources of data I see”
- Observed data can correspond to complex combination of model variables, and exhibit “emergence”
- Frequently we learn from this that our model structure just can’t produce the patterns!

Recall: Calibration: A Bit of the How

- Calibration uses a (global) optimization algorithm to try to adjust unknown parameters so that it automatically matches an arbitrarily large set of data
- The data (often in the form of time series) forms constraints on the calibration
- The optimization algorithm will run the model many (minimally, thousands, typically 100K or more) times to find the “best” match for all of the data

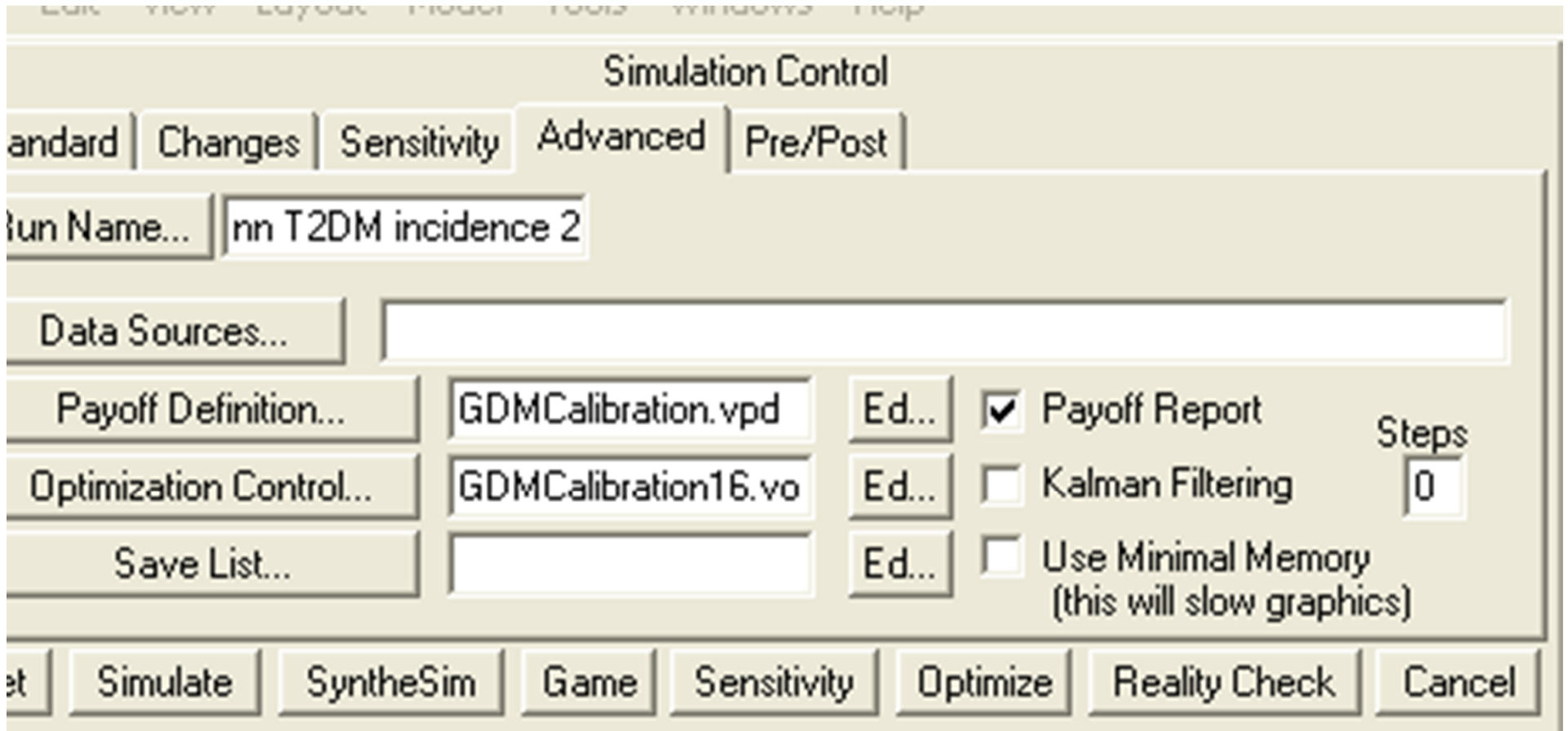
Recall: Required Information for Calibration

- Specification of what to match (and how much to care about each attempted match)
 - Involves an “error function” (“penalty function”, “energy function”) that specifies “how far off we are” for a given run (how good the fit is)
 - Alternative: specify “payoff function” (“objective function”)
- A statement of what parameters to vary, and over what range to vary them (the “parameter space”)
- Characteristics of desired tuning algorithm
 - Single starting point of search?

Recall: Example Global Optimization Algorithm

- Starts at random position, tries to improve match (minimize error) by
 - Adjusting parameters
 - Running Model
 - Recording error function
- Keeps on improving until reaches “local minimum” in error of fit
 - May add some randomness to knock out of local minima

Running Calibrations in Vensim: (Under Model/Simulate Commands)



Optimization Control

Optimization Control. Edit the filename to save changes to a different control file

Filename:

Output Level Trace Sensitivity =

Multiple Start Random type Seed

#Restart Optimizer Max Iterations Max Sims

Pass Limit Fractional Tolerance Tolerance Multiplier

Absolute Tolerance Scale Absolute Vector Points

Currently active parameters (drag to reorder)

- 0<=Net RI Emigration Weight Coefficient for Age Sex[Reproductive]
- 0<=Net RI Emigration Weight Coefficient for Age Sex[PostReproduc
- 0<=Net RI Emigration Weight Coefficient for Age Sex[Child,Male]=.5
- 0<=Net RI Emigration Weight Coefficient for Age Sex[Reproductive
- 0<=Net RI Emigration Weight Coefficient for Age Sex[PostReproduc
- 0<=Net GP Emigration Weight Coefficient for Age Sex[Child,Female]
- 0<=Net GP Emigration Weight Coefficient for Age Sex[Reproductive
- 0<=Net GP Emigration Weight Coefficient for Age Sex[PostReprodu

<= <=

Model value of constant 1 =

Payoff Definition

Payoff Definition. Edit the filename to save changes to a different control file

Filename:

Type Calibration Policy

Payoff Elements

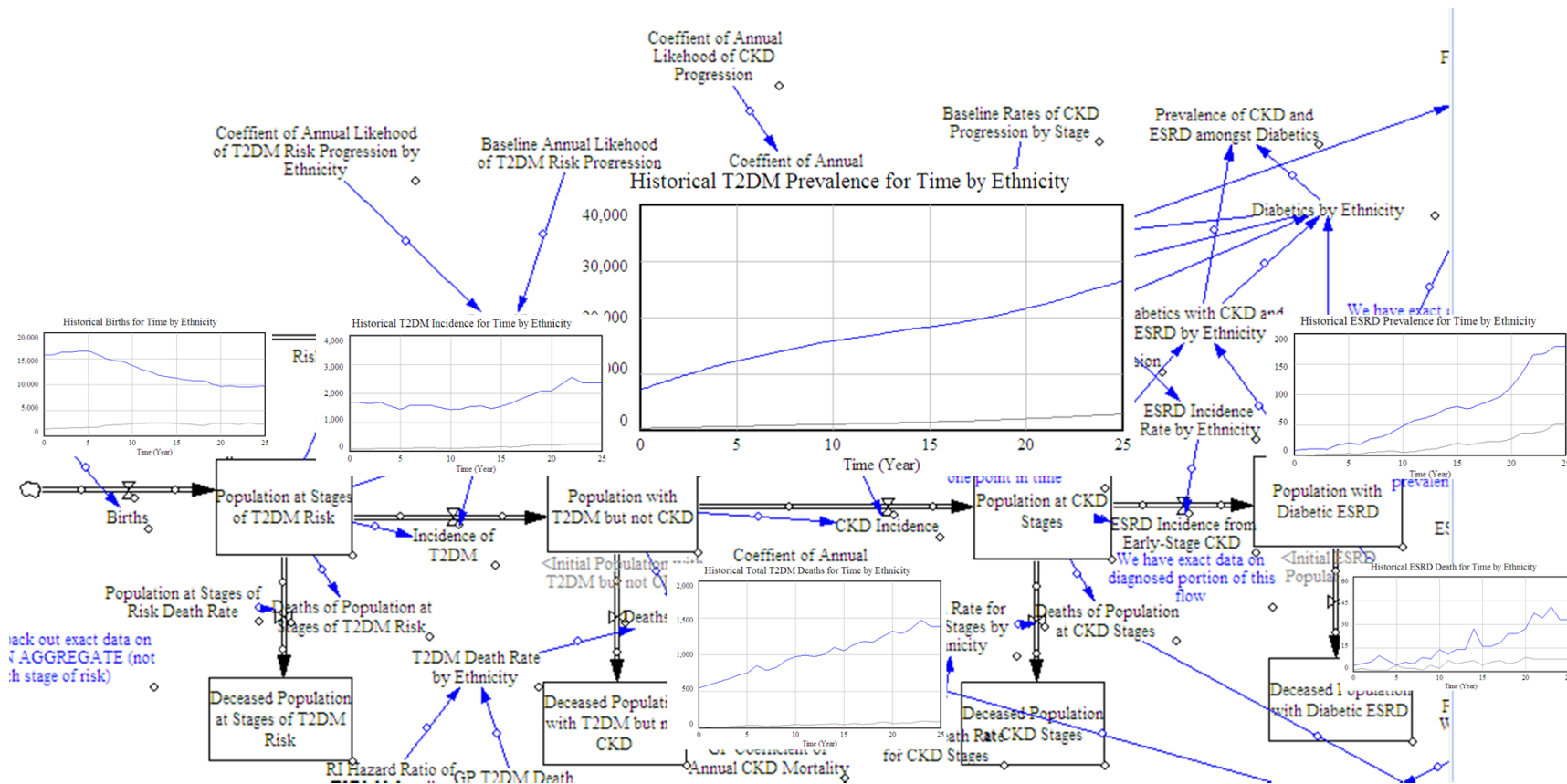
Total Weighted Absolute Value of All Discrepancy for Current Time/-1

Variable	<input type="text"/>	<input type="button" value="Sel"/>	Compare to is used only for calibration payoffs
Compare to	<input type="text"/>	<input type="button" value="Sel"/>	
Weight	<input type="text"/>	<input type="button" value="Sel"/>	

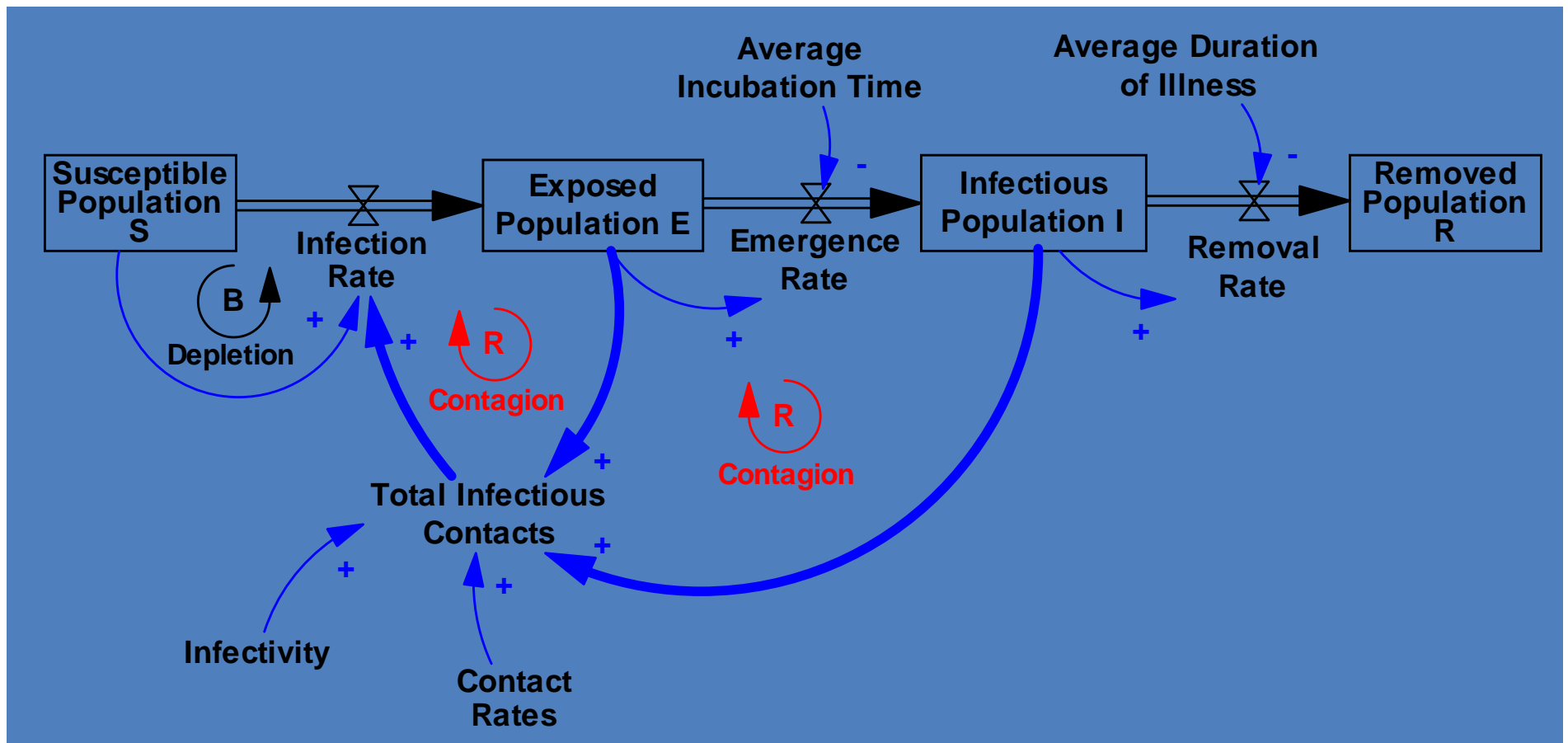
The weight should be positive for calibration. For policy optimizations use a positive number when more is better and a negative number when less is better.

The Pieces of the Elephant

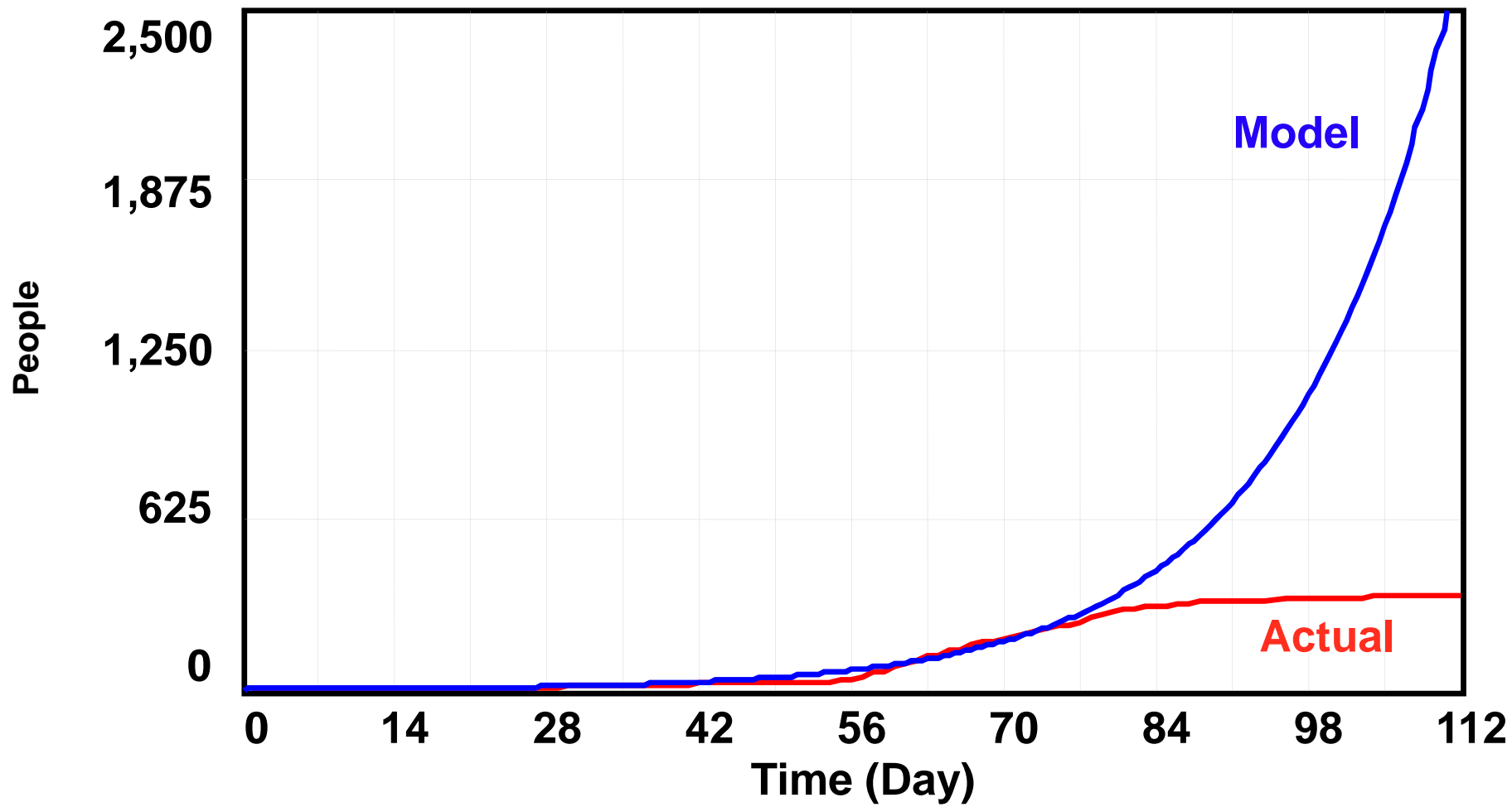
Example Model of Underlying Process & Time Series It Must Match



Example: Iteration & Calibration

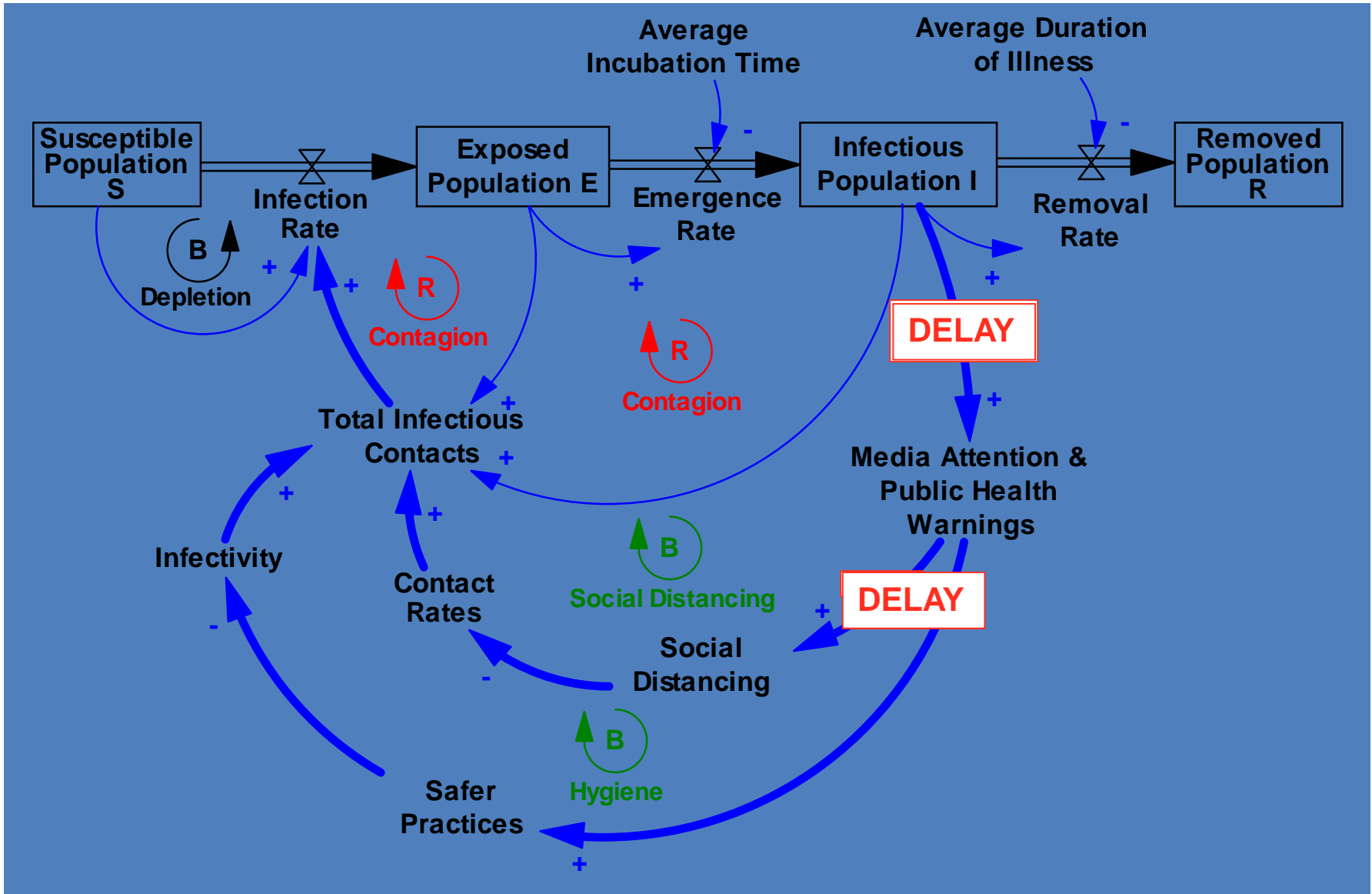


From Sterman



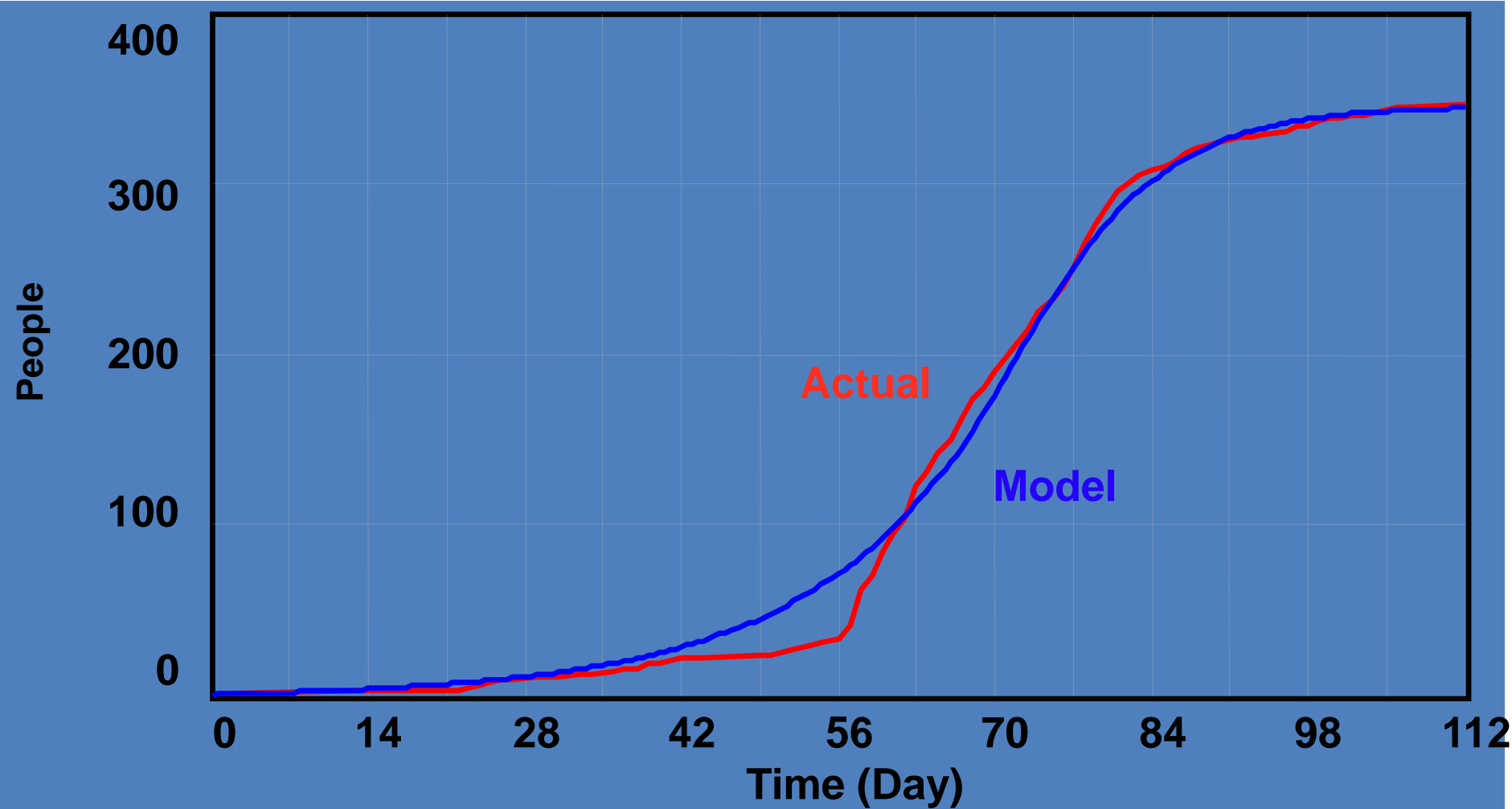
From Sterman

Expanding the Boundary: Behavioral Feedbacks

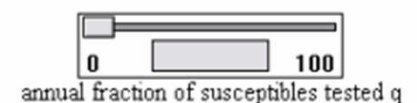
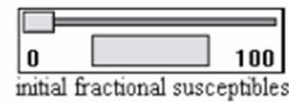
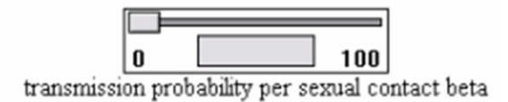
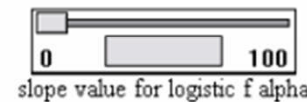
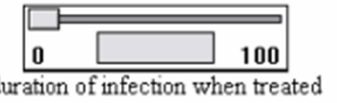
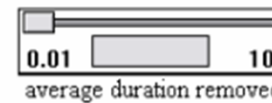
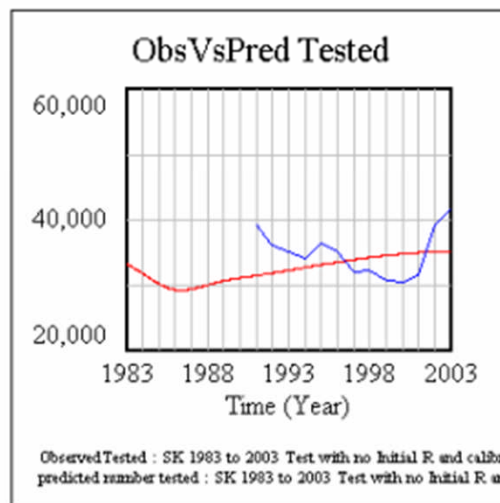
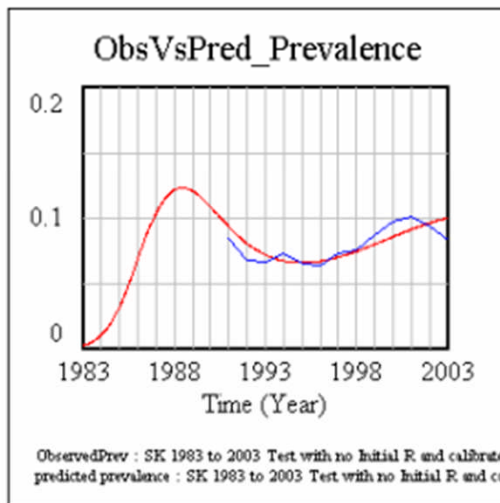
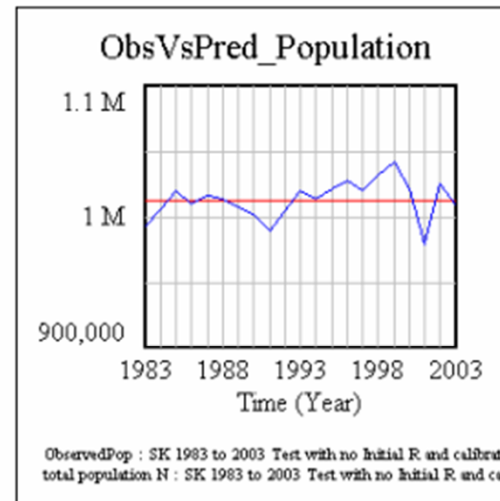
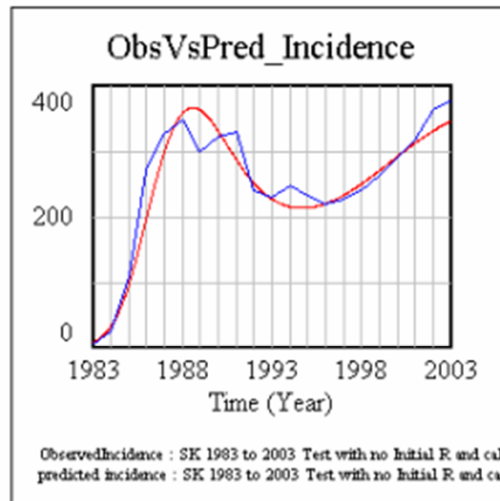
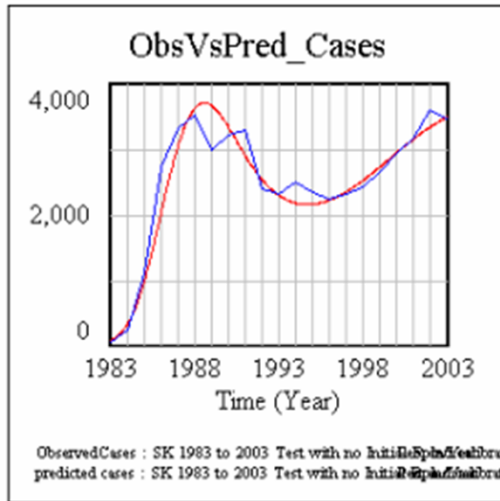


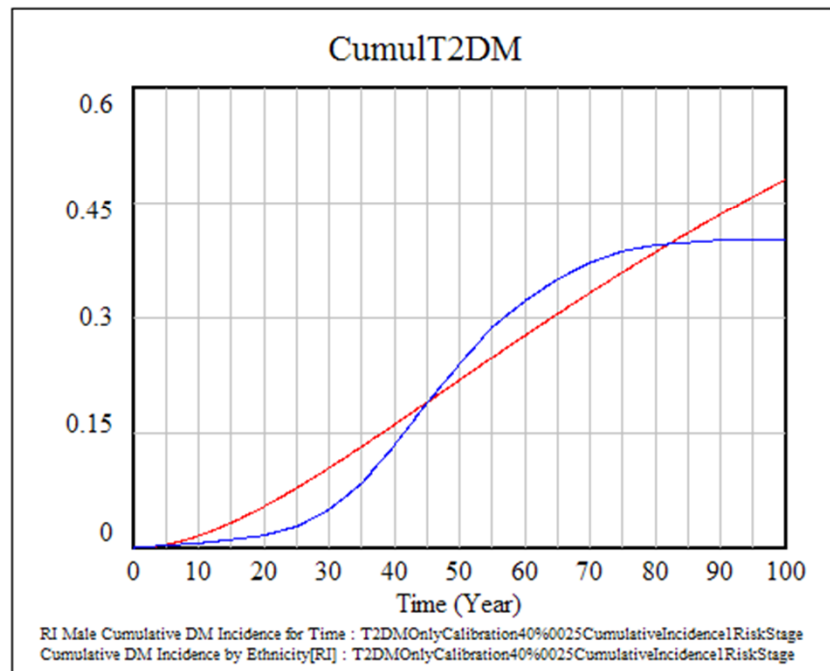
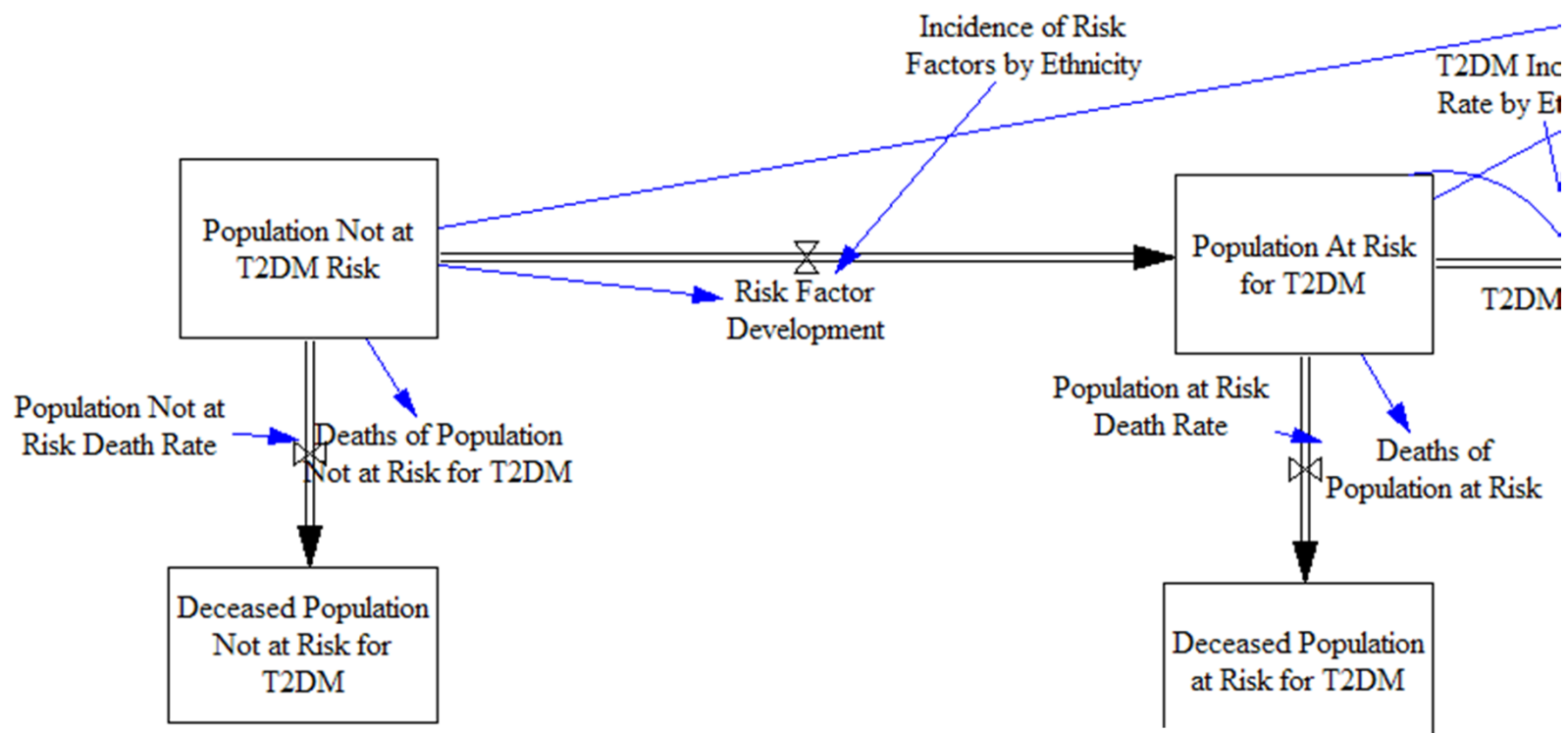
From Sterman

Cumulative Cases



Pieces of the Elephant: STI







Hands on Model Use Ahead



Load Sample Model:

SIR Agent Based Calibration

(Via “Sample Models” under “Help” Menu)

An Optimization Experiment in AnyLogic

Stops after best objective ceases to significantly improve
Caveat Modelor: May prematurely terminate the optimization

Stops after 500 optimization iterations

Varying these parameters

Calibration of Agent Based SIR Model

Run calibration

	Current	Best
Iteration:	infeasible	?
Objective: ↓	?	?

Parameters

ContactRate	?	?
InfectionProbability	?	?

Copy the best solution to the clipboard

copy

Calibration progress

Properties Console

Calibration - Optimization Experiment

Objective: minimize maximize

difference ()

Optimization stop conditions

Iteration count: 500

Automatic Stop

Parameters:

parameter	type	value	min	max	step
AverageI...uration	fixed	15			
ContactRate	continuous	0.5	0	3	0
Infection...bability	continuous	0.1	0	0.8	0
AreaSide	fixed	100			
TotalPopulation	fixed	10000			

Defining a Payoff Function

Caveat: Non-Analytic, Non-Concave

The screenshot shows the AnyLogic Advanced software interface. The main workspace displays a calibration model for an SIR agent-based model. The 'difference' function is highlighted in the code editor, and a red arrow points to the calculation of the discrepancy between historic and model values.

Computing discrepancy between (historic & model values at this point during the run)

```
int diff = 0;
for( int i=0; i<dsInfectiousCurrent.size(); i++ )
    diff += abs( dsInfectiousCurrent.getY(i) - dsInfectiousHistoric.getY(i) );
return diff;
```

Historic Data Captured via Table Function

The screenshot displays the AnyLogic Advanced software interface. The main workspace shows a calibration setup with a 'Run calibration' button and a 'Current' status. A tooltip indicates that the data corresponds to ContactRate = 1.5 and InfectionProbability = 0.4, with a link to 'InfectiousHistoric'. The 'Properties' panel for 'InfectiousHistoric - Table Function' is open, showing the following configuration:

- Name: InfectiousHistoric
- Access: public
- Static: checked
- Interpolation: Linear (highlighted with a yellow box)
- Out of Range: Error
- Value: 0.0

The 'Table Data' section contains the following data:

Argument	Function
2	11.0
4	73.0
6	171.0
8	314.0
10	502.0
12	692.0
14	893.0
16	1082.0
18	1403.0

A yellow arrow points from the text 'How to interpolate ("fill in") between data points' to the 'Interpolation' dropdown menu.

How to interpolate ("fill in") between data points

Stochastics in Agent-Based Models

- Recall that ABMs typically exhibit significant stochastics
 - Event timing within & outside of agents
 - Inter-agent interactions
- When calibrating an ABM, we wish to avoid attributing a good match to a particular set of parameter values simply due to chance
- To reliably assess fit of a given set of parameters, we need to repeatedly run model realizations
 - We can take the mean fit of these realizations

Distinction

- Replication/”Run”: One realization
 - Particular random number seed
- Iteration: Evaluation of a particular parameter set
 - This can contain many realizations (“replications”)
- Confusingly, the term “simulation” appears to sometimes be used for either of the above

Populating the Appropriate Datasets

The screenshot displays the AnyLogic Advanced interface for a calibration experiment. The left sidebar shows a project tree with 'InfectiousHistoric' and 'dsInfectiousHistoric' highlighted. The main workspace shows a 'Run calibration' button and a 'Current' iteration status of 'infeasible'. The 'Parameters' section shows 'InfectionProbability' with a value of 0.4. The 'Console' window shows the following code:

```
Additional Class Code:  
  
Initial Experiment Setup:  
dsInfectiousHistoric.fillFrom( InfectiousHistoric );  
  
Before Each Experiment Run:  
datasetCurrentObjective.reset();  
datasetBestFeasibleObjective.reset();  
  
Before Simulation Run:  
  
After Simulation Run:  
dsInfectiousCurrent.fillFrom( root.dsInfectious );  
  
After Iteration Code:  
if ( getCurrentIteration() == getBestIteration() )  
    dsInfectiousBest.fillFrom( dsInfectiousCurrent );
```

Annotations and callouts:

- Populates historic data up front from table fn** (Red text, arrow pointing to the 'Initial Experiment Setup' code block).
- These datasets are within the experiment Persist beyond the simulation** (Yellow text, arrow pointing to the 'dsInfectiousHistoric' and 'dsInfectiousCurrent' datasets in the project tree).
- Saves away best simulation Within in iteration** (Green text, arrow pointing to the 'After Iteration Code' block).
- Retaining the Current value After the realization (Simulation run)** (Blue text, arrow pointing to the 'After Simulation Run' code block).

Additional callouts include a box stating: 'These data correspond to ContactRate = 1.5 InfectionProbability = 0.4' and a 'Run calibration' button.

Running Calibration in AnyLogic

Calibration of Agent Based SIR Model

Run calibration

	Current	Best
Iteration:	5	3
Objective: ↓	120,500	3,895

Parameters

ContactRate	2.756	3
InfectionProbability	0.119	0.8

Copy the best solution to the clipboard

In this applet OptQuest optimizer is used to calibrate an agent based model of epidemic spread developed with AnyLogic. In that model each person is represented as a active object (agent) with 4 possible states: Susceptible, Exposed, Infectious and Recovered (SEIR). Initially all but few people are susceptible, and few – exposed. A person can contact another person, and in case one is susceptible and another – exposed or infectious, the first may get infected with a certain probability. The objective is to find the parameters of the agents (contact frequencies and infection probabilities) so that the output of the simulation model fits best with the historical data (in this case – the dynamics of infectious population). As the model is stochastic, the optimization is done under uncertainty, and simulation replications are used.

Calibration progress

Best payoff (objective) yet reached (lower is better)

Historic data, best fitting and current simulation output

Values of parameters being calibrated at best calibration thus far

Run: 4 Running Experiment: 1% Simulation: 5% 11.1 sec

Optimization Constraints – Tests on Legitimacy of Parameter Values

The screenshot displays the AnyLogic Advanced software interface for calibrating an Agent Based SIR Model. The main window is titled "Calibration of Agent Based SIR Model" and features a "Run calibration" button. Below this, a table compares "Current" and "Best" values for "Iteration" and "Objective". The "Current" iteration is marked as "infeasible". A "Calibration progress" graph shows a vertical red line at approximately 0.8 on the y-axis. The "Parameters" table lists "ContactRate" with a yellow highlight and a question mark.

	Current	Best
Iteration:	infeasible ?	?
Objective: ↓	? ?	?

Parameters

Parameter	Value
ContactRate	? ?

The bottom console window, titled "Calibration - Optimization Experiment", shows two tables for constraints and requirements. The "Constraints on simulation parameters" table is currently empty, with a red background. The "Requirements" table is also empty.

enabled	expression	type	bound
<input type="checkbox"/>			

Requirements (are tested after a simulation run to determine whether the solution is feasible):

enabled	expression	type	bound
<input type="checkbox"/>			

Optimization Requirements – Tests on Emergent results to Sense Validity

The screenshot displays the AnyLogic Advanced software interface for the "Calibration of Agent Based SIR Model". The main workspace shows a "Run calibration" button and a "Calibration progress" graph. The graph plots the progress of the calibration process, with a red vertical line indicating the current iteration. The current iteration is labeled "infeasible" and the objective is shown as a downward arrow.

The "Parameters" table shows the current and best values for the ContactRate parameter:

Parameter	Current	Best
ContactRate	?	?

The "Constraints" table shows the constraints on simulation parameters (are tested before a simulation run):

enabled	expression	type	bound
<input type="checkbox"/>			

The "Requirements" table (highlighted in red) shows the requirements (are tested after a simulation run to determine whether the solution is feasible):

enabled	expression	type	bound
<input type="checkbox"/>			

The interface also includes a "Palette" on the right with various model elements like Parameter, Flow Aux Variable, Stock Variable, Event, Dynamic Event, Plain Variable, Collection Variable, Function, Table Function, Port, Connector, Entry Point, State, Transition, Initial State Pointer, Branch, History State, Final State, and Environment. The bottom panel shows "Selection" and "More Libraries..." options.

Enabling Multiple Realizations ("Replications", "Runs") per Iteration

The screenshot displays the AnyLogic Advanced [EDUCATIONAL USE ONLY] interface. The main workspace shows a calibration experiment titled "Calibration - Optimization Experiment". The interface is divided into several panels:

- Project Panel (Left):** Lists model components including variables (InfectionProbability: 0.8, TotalPopulation: 10000), plain variables (nInfectious), environments, embedded objects (people), analysis data (dsInfectious), presentation (people_presentation), person, and calibration (Main). Under "Calibration: Main", it lists functions (InfectiousHistoric, difference), analysis data (datasetCurrentObjective, datasetBestFeasibleObjective, dsInfectiousHistoric, dsInfectiousCurrent, dsInfectiousBest), and presentation (MonteCarlo2DHistogram: Main).
- Main Workspace:** Contains a grid with data points and a text box stating: "These data correspond to ContactRate = 1.5 InfectionProbability = 0.4". Below this, several data series are listed: InfectiousHistoric, dsInfectiousHistoric, dsInfectiousCurrent, difference, and dsInfectiousBest.
- Properties Panel (Right):** Shows the "Current" state with "Iteration: ?" and "Objective: ↓ ?" (marked as "infeasible"). It lists parameters: ContactRate and InfectionProbability. A button "Copy the best solution to the clipboard" is visible. A text box at the bottom explains: "In this applet OptQuest optimizer i calibrate an agent based model o spread developed with AnyLogic. I each person is represented as a : (agent) with 4 possible states: Su Exposed, Infectious and Recovere".
- Properties Panel (Bottom):** Titled "Calibration - Optimization Experiment", it has tabs for General, Advanced, Model Time, Presentation, Window, Constraints, and Replications. The "Replications" tab is active, showing a "Description" field and a checkbox for "Use replications" which is currently unchecked.
- Problems Panel (Bottom Left):** A table with columns "Description" and "Location".
- Palette (Right):** Lists various model components such as Parameter, Flow Aux Variable, Stock Variable, Event, Dynamic Event, Plain Variable, Collection Variable, Function, Table Function, Port, Connector, Entry Point, State, Transition, Initial State Pointer, Branch, History State, Final State, Environment, Action, Analysis, Presentation, Connectivity, and Enterprise Library.

Fixed Number of Replications per Iteration

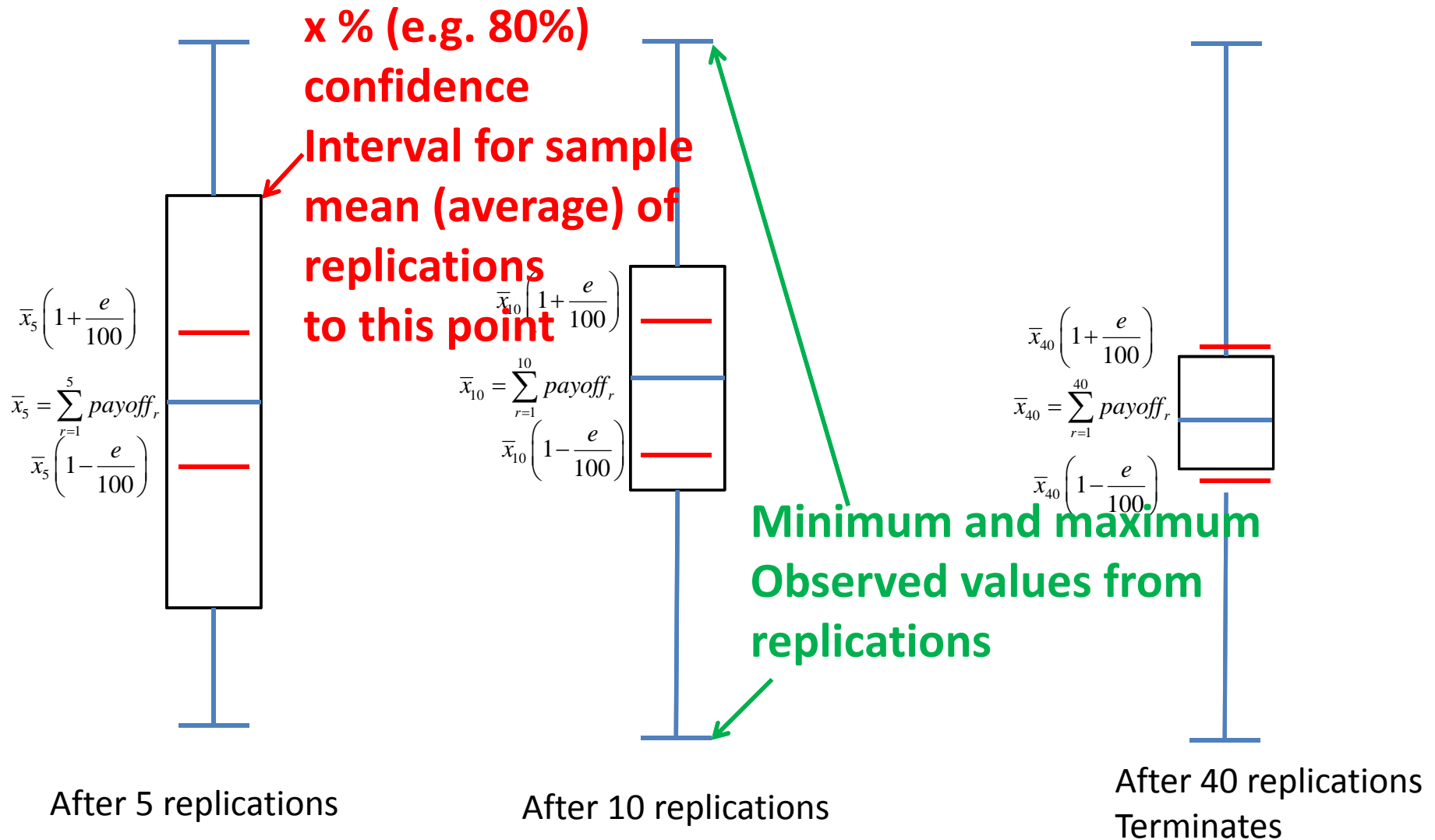
The screenshot displays the AnyLogic Advanced software interface. The main workspace shows a calibration model with various variables and functions. A red text overlay reads: "Specifies stopping Condition once minimum replications have been run. Indicates that the X% confidence interval around the mean is within 'Error percent' of the iteration mean obtained as of the most recent replication".

The Properties window is open to the "Replications" section, showing the following settings:

- Use replications
- Fixed number of replications
 - Replications per iteration: 10
- Varying number of replications (Stop replications after minimum replications, when confidence level is reached)
 - Minimum replications: 10
 - Maximum replications: 10
- Confidence level: 80%
- Error percent: 0,5

The "Error percent" field is highlighted with a red oval, and a red arrow points from the text overlay to it.

Example



Automatic Throttling of Replications Based on Empirical Fractiles for the Average of the Differences between Best and Current

The screenshot displays the AnyLogic Advanced software interface, specifically the 'Calibration - Optimization Experiment' window. The interface is divided into several panes:

- Project Explorer (Left):** Shows a hierarchical tree of model components, including 'InfectionProbability: 0.8', 'TotalPopulation: 10000', 'Plain Variables', 'nInfectious', 'Environments', 'Embedded Objects', 'people', 'Analysis Data', 'dsInfectious', 'Presentation', 'people_presentation', 'Person', 'Calibration: Main', 'Functions', 'InfectiousHistoric', 'difference', 'Analysis Data', 'datasetCurrentObjective', 'datasetBestFeasibleObjective', 'dsInfectiousHistoric', 'dsInfectiousCurrent', 'dsInfectiousBest', 'Presentation', and 'MonteCarlo2DHistogram: Main'.
- Main Canvas (Center):** Displays a grid with several data points and a text box. The text box contains: "These data correspond to ContactRate = 1.5, InfectionProbability = 0.4". Below this, there are several data points with labels like 'InfectiousHistoric', 'dsInfectiousHistoric', 'dsInfectiousCurrent', 'difference', and 'dsInfectiousBest'.
- Properties Panel (Bottom Left):** Shows the 'Calibration - Optimization Experiment' settings. The 'General' tab is active, showing options for 'Use replications' (checked), 'Fixed number of replications' (unchecked), and 'Varying number of replications (Stop replications after minimum replications, when confidence level is reached)' (checked). Other settings include 'Replications per iteration: 10', 'Minimum replications: 10', 'Maximum replications: 100', 'Confidence level: 80%', and 'Error percent: 0.5'.
- Console (Bottom Center):** Displays the title 'Calibration - Optimization Experiment'.
- Palette (Right):** Shows a list of model components and their icons, including 'Parameter', 'Flow Aux Variable', 'Stock Variable', 'Event', 'Dynamic Event', 'Plain Variable', 'Collection Variable', 'Function', 'Table Function', 'Port', 'Connector', 'Entry Point', 'State', 'Transition', 'Initial State Pointer', 'Branch', 'History State', 'Final State', and 'Environment'.

The 'Current' status bar at the top right indicates 'Iteration: ?' and 'Objective: ?' with a red 'infeasible' label. A yellow vertical bar is visible on the right side of the main canvas.

Enabling Random Variation Between Realizations (“Replications”)

The screenshot displays the AnyLogic Advanced [EDUCATIONAL USE ONLY] interface. The main workspace shows a calibration experiment named "Calibration" with the following parameters and data:

- Current:** Iteration: **infeasible**, Objective: **?**
- Parameters:** ContactRate: **?**, InfectionProbability: **?**
- Best Solution:** ContactRate = 1.5, InfectionProbability = 0.4
- Dataset:** dsInfectiousHistoric
- Other Datasets:** dsInfectiousHistoric, dsInfectiousCurrent, difference, dsInfectiousBest

The "Calibration - Optimization Experiment" configuration panel is visible at the bottom, showing the following settings:

- Name:** Calibration
- Main active object class (root):** Main
- Random number generation:** Random seed (unique simulation runs), Fixed seed (reproducible simulation runs) with Seed Value: 1
- Objective:** minimize, maximize
- Objective Function:** difference()
- Optimization stop conditions:** Iteration count: 500, Automatic Stop

The left sidebar shows the project structure, including variables like InfectionProbability (0.8) and TotalPopulation (10000), and the "Calibration: Main" section containing the optimization experiment. The right sidebar contains the "Palette" with various modeling elements like Parameter, Flow Aux Variable, Stock Variable, Event, Dynamic Event, Plain Variable, Collection Variable, Function, Table Function, Port, Connector, Entry Point, State, Transition, Initial State Pointer, Branch, History State, Final State, and Environment.

Understanding Replications: Report Results for Each Replication!

The screenshot displays the AnyLogic Advanced [EDUCATIONAL USE ONLY] interface. The main workspace shows a calibration experiment titled "Calibration - Optimization Experiment". The interface is divided into several panels:

- Project Panel (Left):** Lists model components including "InfectionProbability: 0.8", "TotalPopulation: 10000", "Plain Variables", "nInfectious", "Environments", "Embedded Objects", "people", "Analysis Data", "dsInfectious", "Presentation", "people_presentation", "Person", "Calibration: Main", "Functions", "InfectiousHistoric", "difference", "Analysis Data", "datasetCurrentObjective", "datasetBestFeasibleObjective", "dsInfectiousHistoric", "dsInfectiousCurrent", "dsInfectiousBest", "Presentation", and "MonteCarlo2DHistogram: Main".
- Main Workspace:** Contains a grid with a text box stating "These data correspond to ContactRate = 1.5 InfectionProbability = 0.4" and a list of datasets: "datasetBestFeasibleObjective", "InfectiousHistoric", "dsInfectiousHistoric", "dsInfectiousCurrent", "difference", and "dsInfectiousBest".
- Properties Panel (Right):** Shows the "Current" state of the calibration. It indicates "Iteration: ?" and "Objective: ?" with a red "infeasible" status. Parameters listed include "ContactRate" and "InfectionProbability". A button "Copy the best solution to the clipboard" is visible. A note at the bottom explains: "In this applet OptQuest optimizer i calibrate an agent based model o spread developed with AnyLogic. I each person is represented as a (agent) with 4 possible states: Su Exposed, Infectious and Recovere".
- Code Editor (Bottom):** Shows the "Advanced" configuration for the experiment. The code includes:

```
dsInfectiousHistoric.fillFrom( InfectiousHistoric );  
  
Before Each Experiment Run:  
datasetCurrentObjective.reset();  
datasetBestFeasibleObjective.reset();  
  
Before Simulation Run:  
  
After Simulation Run:  
dsInfectiousCurrent.fillFrom( root.dsInfectious );  
println("For this particular simulation, the difference is\t" + difference());  
  
After Iteration Code:  
if( getCurrentIteration() == getBestIteration() )  
    dsInfectiousBest.fillFrom( dsInfectiousCurrent );
```
- Palette (Far Right):** Lists various model elements such as "Parameter", "Flow Aux Variable", "Stock Variable", "Event", "Dynamic Event", "Plain Variable", "Collection Variable", "Function", "Table Function", "Port", "Connector", "Entry Point", "State", "Transition", "Initial State Pointer", "Branch", "History State", "Final State", "Environment", "Action", "Analysis", "Presentation", "Connectivity", "Enterprise Library", and "More Libraries...".

During First Several Realizations (“Replications”, “Runs”), No Results Appear

Calibration of Agent Based SIR Model

Run calibration

	Current	Best
Iteration:	1	
Objective: ↓	3,343.5	

Parameters

ContactRate	1.75
InfectionProbability	0.45

Copy the best solution to the clipboard

In this applet OptQuest optimizer is used to calibrate an agent based model of epidemic spread developed with AnyLogic. In that model each person is represented as a active object (agent) with 4 possible states: Susceptible, Exposed, Infectious and Recovered (SEIR). Initially all but few people are susceptible, and few – exposed. A person can contact another person, and in case one is susceptible and another – exposed or infectious, the first may get infected with a certain probability. The objective is to find the parameters of the agents (contact frequencies and infection probabilities) so that the output of the simulation model fits best with the historical data (in this case – the dynamics of infectious population). As the model is stochastic, the optimization is done under uncertainty, and simulation replications are used.

Calibration progress

Legend: ■ Current objective, — Best feasible objective

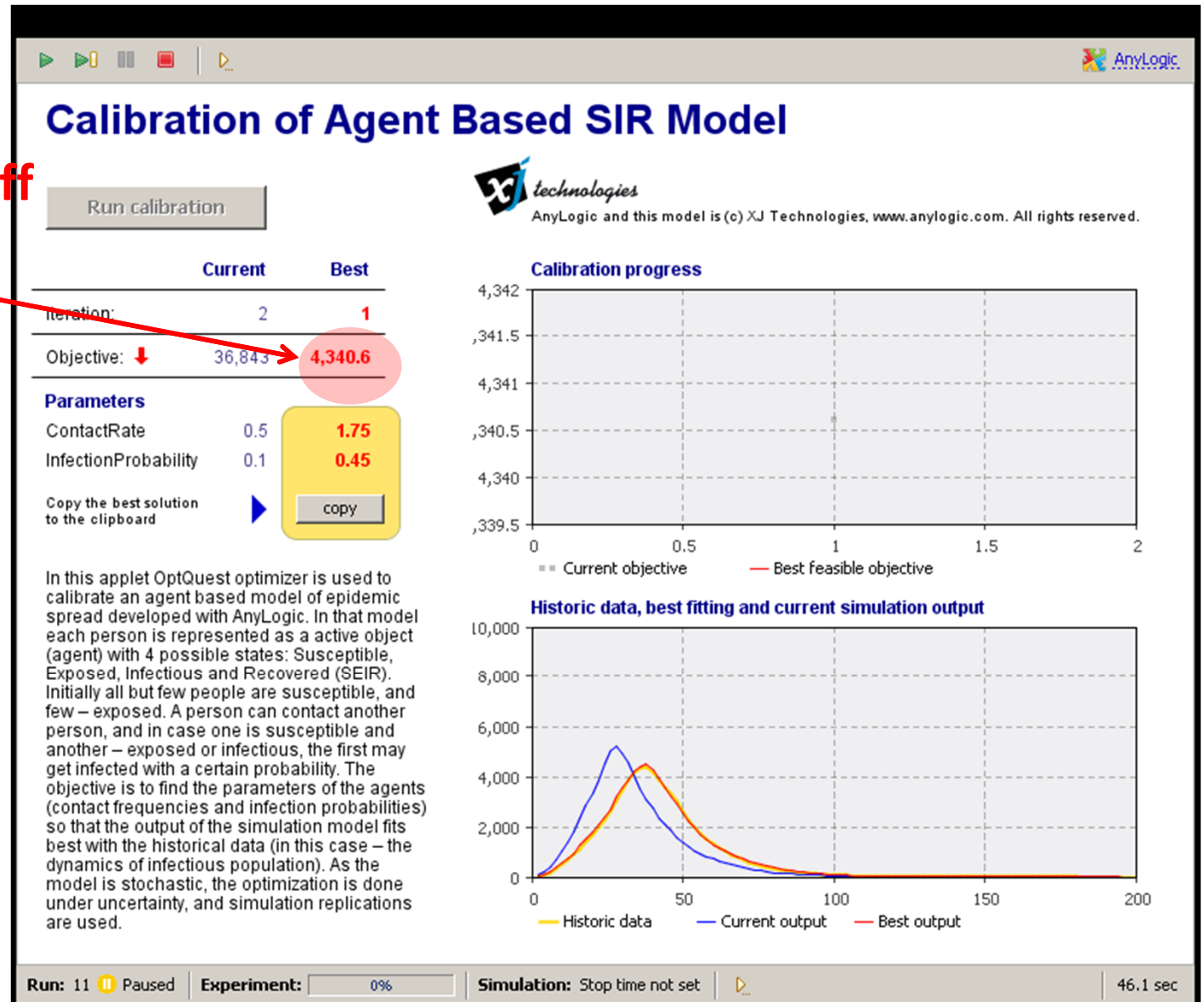
Historic data, best fitting and current simulation output

Legend: — Historic data, — Current output, — Best output

Run: 2 Running Experiment: Simulation: 4.6 sec

Report on Iteration 1 Appears after a Count of Runs Equal to Replications per Iteration

Reports best payoff (objective) yet reached (lower is better), but from where did this number come?



Output

The screenshot displays the AnyLogic Advanced software interface. The main window shows a simulation model with various components like 'InfectiousHistoric', 'difference', and 'dsInfectiousBest'. A console window at the bottom shows the following output:

```
anylogic config [Java Application] C:\Program Files (x86)\AnyLogic 6\jre\bin\javaw.exe (Mar 31, 2011 4:44:06 AM)
For this particular simulation, the difference is 3324
For this particular simulation, the difference is 3363
For this particular simulation, the difference is 8866
For this particular simulation, the difference is 2052
For this particular simulation, the difference is 2447
For this particular simulation, the difference is 3552
For this particular simulation, the difference is 6079
For this particular simulation, the difference is 6082
For this particular simulation, the difference is 4775
For this particular simulation, the difference is 2866
For this particular simulation, the difference is 36843
```

A red callout box highlights the values 3324, 3363, 8866, 2052, 2447, 3552, 6079, 6082, 4775, and 2866, which are the difference values for each replication. A red arrow points from the text below to this box.

The reported payoff for the iteration is the average of the payoffs for each replication within the replication

Average of Results for Replications is the Reported Score for the Iteration!

The screenshot shows the Microsoft Excel interface with the following data and formula:

Row	Column D
3	3324
4	3363
5	8866
6	2052
7	2447
8	3552
9	6079
10	6082
11	4775
12	2866
14	4340.6

The formula bar shows: `=AVERAGE(D3:D12)`

The status bar at the bottom indicates: Ready, Sheet1, Sheet2, Sheet3, 100%