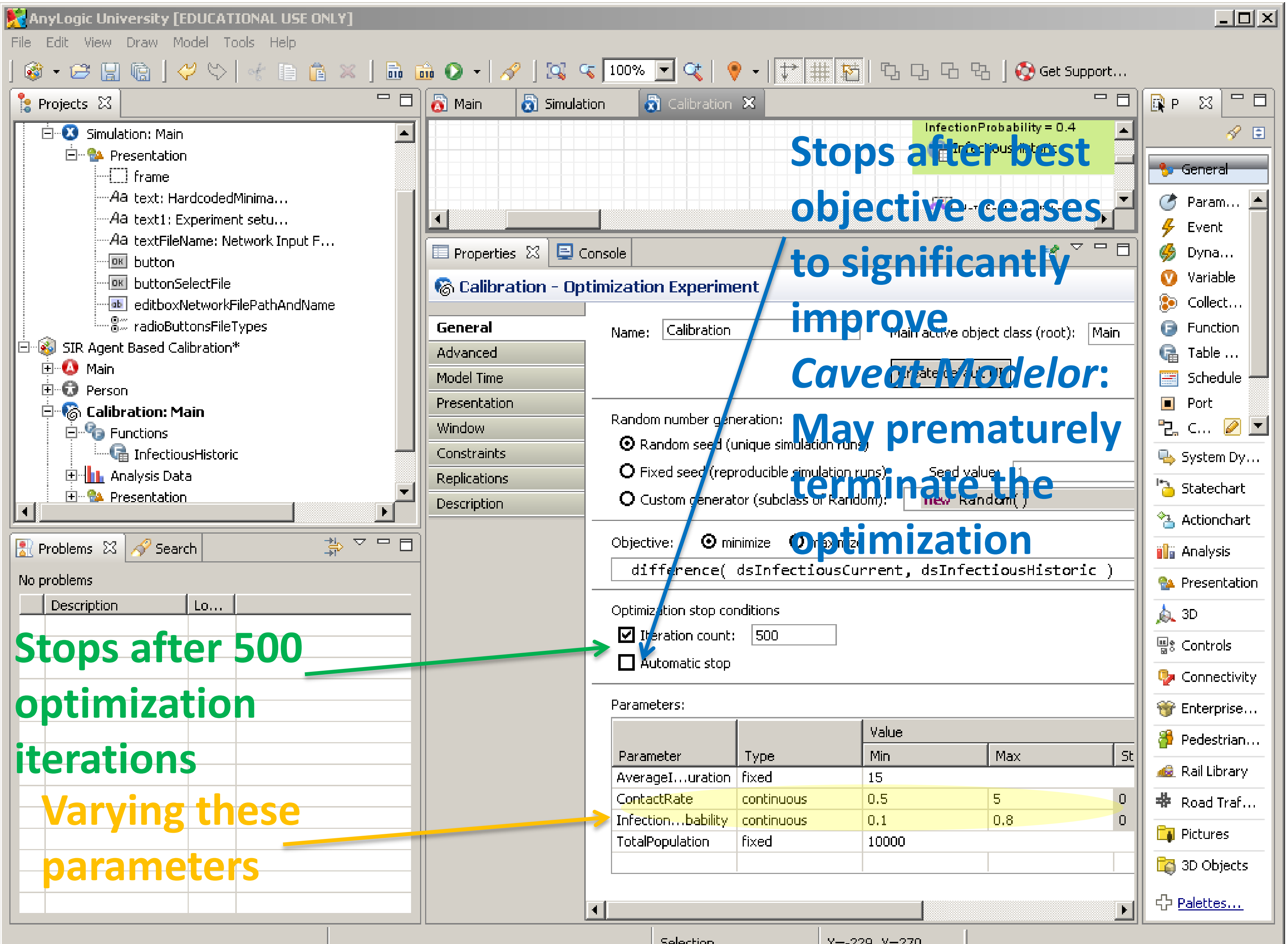
Recall: Optimization Experiment in AnyLogic



AnyLogic University [EDUCATIONAL USE ONLY]

Projects

- Simulation: Main
  - Presentation
    - frame
    - text: HardcodedMinima…
    - text1: Experiment setu…
    - textFileName: Network Input F…
    - button
    - buttonSelectFile
    - editboxNetworkFilePathAndName
    - radioButtonsFileTypes
- SIR Agent Based Calibration*
  - Main
  - Person
  - Calibration: Main
    - Functions
      - InfectiousHistoric
    - Analysis Data
    - Presentation

InfectionProbability = 0.4

Problems

No problems

Description	Lo…

Calibration - Optimization Experiment

General | Advanced | Model Time | Presentation | Window | Constraints | Replications | Description

Name: Calibration

Main active object class (root): Main

Random number generation:

● Random seed (unique simulation runs)

○ Fixed seed (reproducible simulation runs) Seed value: 1

○ Custom generator (subclass of Random): new Random( )

Objective: ● minimize ○ maximize

```
difference( dsInfectiousCurrent, dsInfectiousHistoric )
```

Optimization stop conditions

☑ Iteration count: 500

□ Automatic stop

Parameters:

Parameter	Type	Value		
		Min	Max	St
AverageI…uration	fixed	15		
ContactRate	continuous	0.5	5	0
Infection…bability	continuous	0.1	0.8	0
TotalPopulation	fixed	10000		

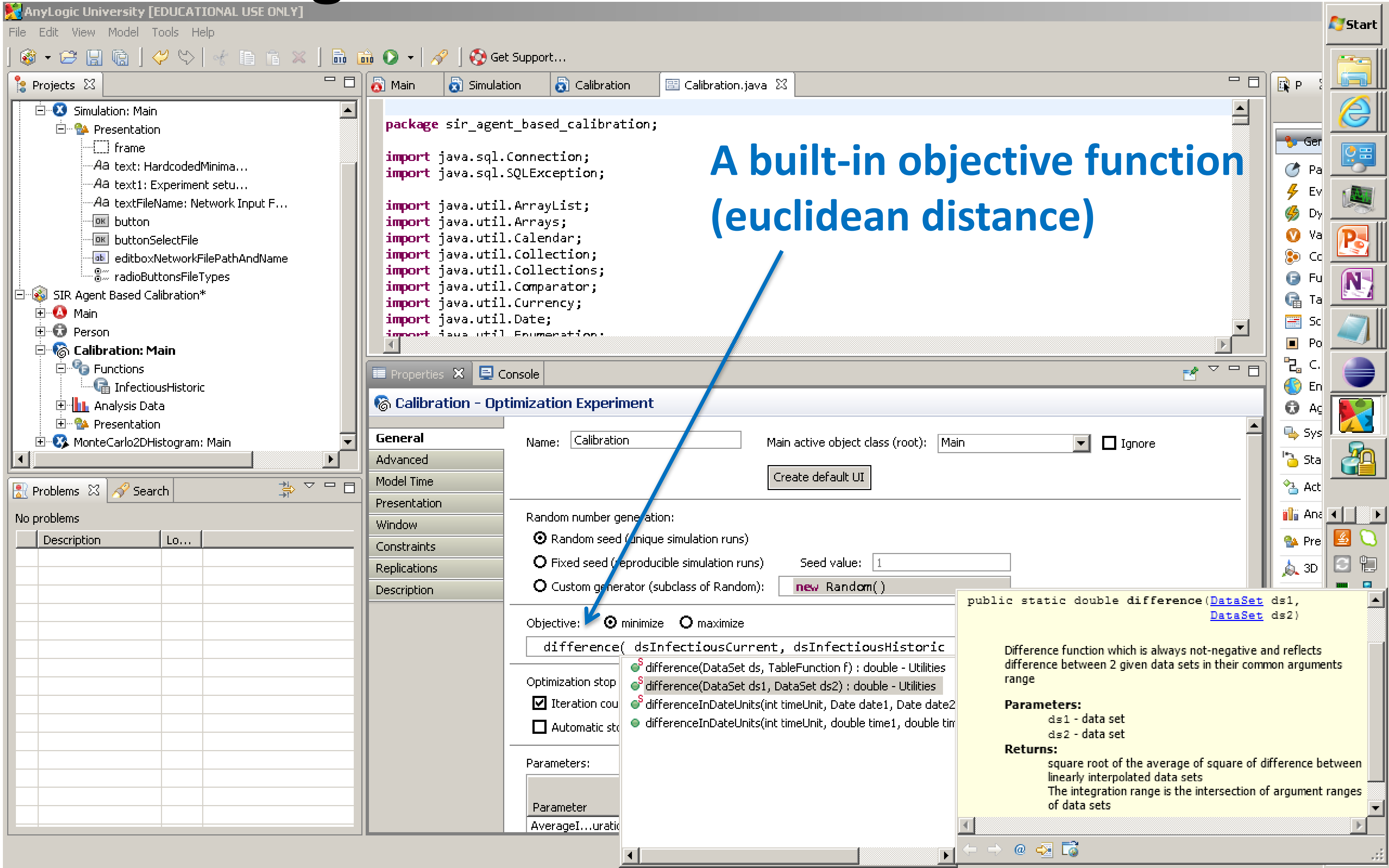
Stops after best objective ceases to significantly improve

Caveat Modelor: May prematurely terminate the optimization

Stops after 500 optimization iterations

Varying these parameters

An Optimization Experiment in AnyLogic Using Built-in Difference Function



```
package sir_agent_based_calibration;

import java.sql.Connection;
import java.sql.SQLException;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Calendar;
import java.util.Collection;
import java.util.Collections;
import java.util.Comparator;
import java.util.Currency;
import java.util.Date;
import java.util.Enumeration;
```

A built-in objective function (euclidean distance)

Calibration - Optimization Experiment

Name: Calibration

Main active object class (root): Main □ Ignore

Create default UI

Random number generation:

● Random seed (unique simulation runs)

○ Fixed seed (reproducible simulation runs) Seed value: 1

○ Custom generator (subclass of Random): new Random()

Objective: ● minimize ○ maximize

```
difference( dsInfectiousCurrent, dsInfectiousHistoric
```

Optimization stop

☑ Iteration cou

□ Automatic sto

Parameters:

Parameter

AverageI...uratio

- difference(DataSet ds, TableFunction f) : double - Utilities
- difference(DataSet ds1, DataSet ds2) : double - Utilities
- differenceInDateUnits(int timeUnit, Date date1, Date date2
- differenceInDateUnits(int timeUnit, double time1, double tim

```
public static double difference(DataSet ds1,
                                DataSet ds2)
```

Difference function which is always not-negative and reflects difference between 2 given data sets in their common arguments range

Parameters:

ds1 - data set

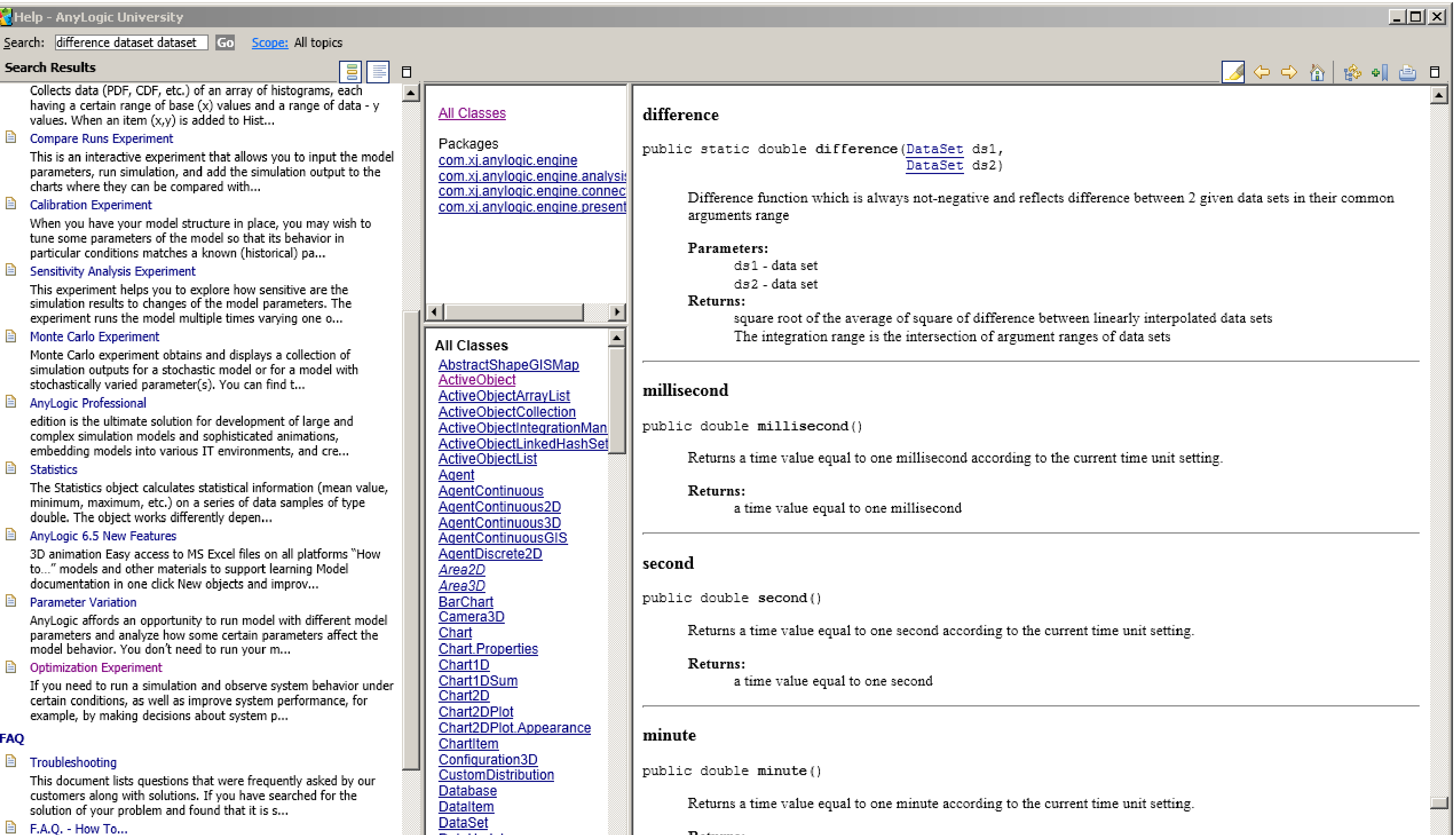
ds2 - data set

Returns:

square root of the average of square of difference between linearly interpolated data sets

The integration range is the intersection of argument ranges of data sets

Finding the Definition



Help - AnyLogic University
Search: difference dataset dataset
Go
Scope: All topics
Search Results

Collects data (PDF, CDF, etc.) of an array of histograms, each having a certain range of base (x) values and a range of data - y values. When an item (x,y) is added to Hist...

Compare Runs Experiment

This is an interactive experiment that allows you to input the model parameters, run simulation, and add the simulation output to the charts where they can be compared with...

Calibration Experiment

When you have your model structure in place, you may wish to tune some parameters of the model so that its behavior in particular conditions matches a known (historical) pa...

Sensitivity Analysis Experiment

This experiment helps you to explore how sensitive are the simulation results to changes of the model parameters. The experiment runs the model multiple times varying one o...

Monte Carlo Experiment

Monte Carlo experiment obtains and displays a collection of simulation outputs for a stochastic model or for a model with stochastically varied parameter(s). You can find t...

AnyLogic Professional

edition is the ultimate solution for development of large and complex simulation models and sophisticated animations, embedding models into various IT environments, and cre...

Statistics

The Statistics object calculates statistical information (mean value, minimum, maximum, etc.) on a series of data samples of type double. The object works differently depen...

AnyLogic 6.5 New Features

3D animation Easy access to MS Excel files on all platforms "How to..." models and other materials to support learning Model documentation in one click New objects and improv...

Parameter Variation

AnyLogic affords an opportunity to run model with different model parameters and analyze how some certain parameters affect the model behavior. You don't need to run your m...

Optimization Experiment

If you need to run a simulation and observe system behavior under certain conditions, as well as improve system performance, for example, by making decisions about system p...

FAQ

Troubleshooting

This document lists questions that were frequently asked by our customers along with solutions. If you have searched for the solution of your problem and found that it is s...

F.A.Q. - How To...

All Classes

Packages
com.xj.anylogic.engine
com.xj.anylogic.engine.analysis
com.xj.anylogic.engine.connec
com.xj.anylogic.engine.present

All Classes
AbstractShapeGISMap
ActiveObject
ActiveObjectArrayList
ActiveObjectCollection
ActiveObjectIntegrationMan
ActiveObjectLinkedHashSet
ActiveObjectList
Agent
AgentContinuous
AgentContinuous2D
AgentContinuous3D
AgentContinuousGIS
AgentDiscrete2D
Area2D
Area3D
BarChart
Camera3D
Chart
Chart.Properties
Chart1D
Chart1DSum
Chart2D
Chart2DPlot
Chart2DPlot.Appearance
ChartItem
Configuration3D
CustomDistribution
Database
DataItem
DataSet

difference

```
public static double difference(DataSet ds1,
                                DataSet ds2)
```

Difference function which is always not-negative and reflects difference between 2 given data sets in their common arguments range

Parameters:
`ds1` - data set
`ds2` - data set

Returns:
square root of the average of square of difference between linearly interpolated data sets
The integration range is the intersection of argument ranges of data sets

millisecond

```
public double millisecond()
```

Returns a time value equal to one millisecond according to the current time unit setting.

Returns:
a time value equal to one millisecond

second

```
public double second()
```

Returns a time value equal to one second according to the current time unit setting.

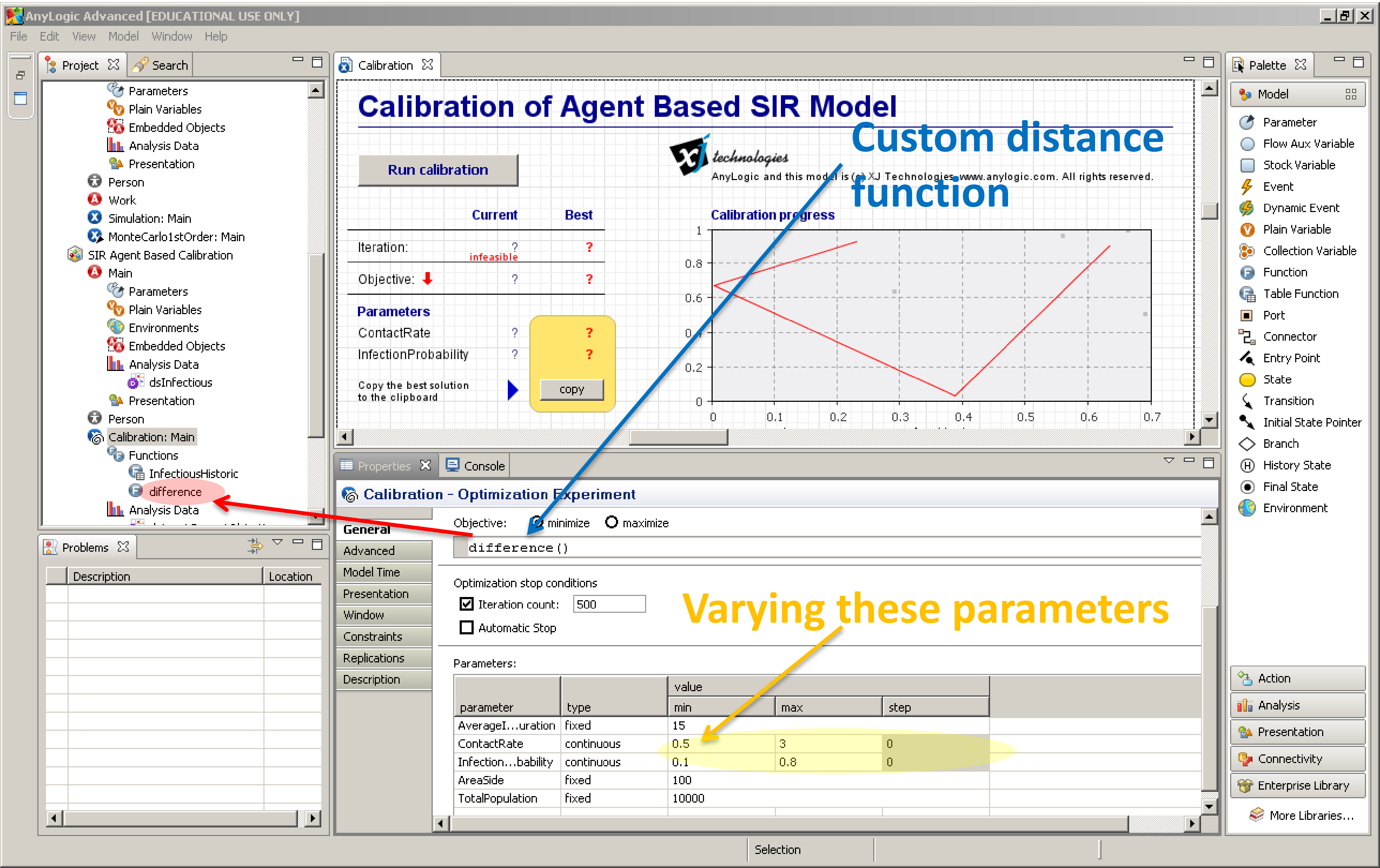
Returns:
a time value equal to one second

minute

```
public double minute()
```

Returns a time value equal to one minute according to the current time unit setting.

An Optimization Experiment in AnyLogic with a custom difference function



	Current	Best
Iteration:	? infeasible	?
Objective:	?	?
Parameters		
ContactRate	?	?
InfectionProbability	?	?

Calibration - Optimization Experiment

Objective: ● minimize ○ maximize

```
difference()
```

Optimization stop conditions

☑ Iteration count: 500

□ Automatic Stop

Parameters:

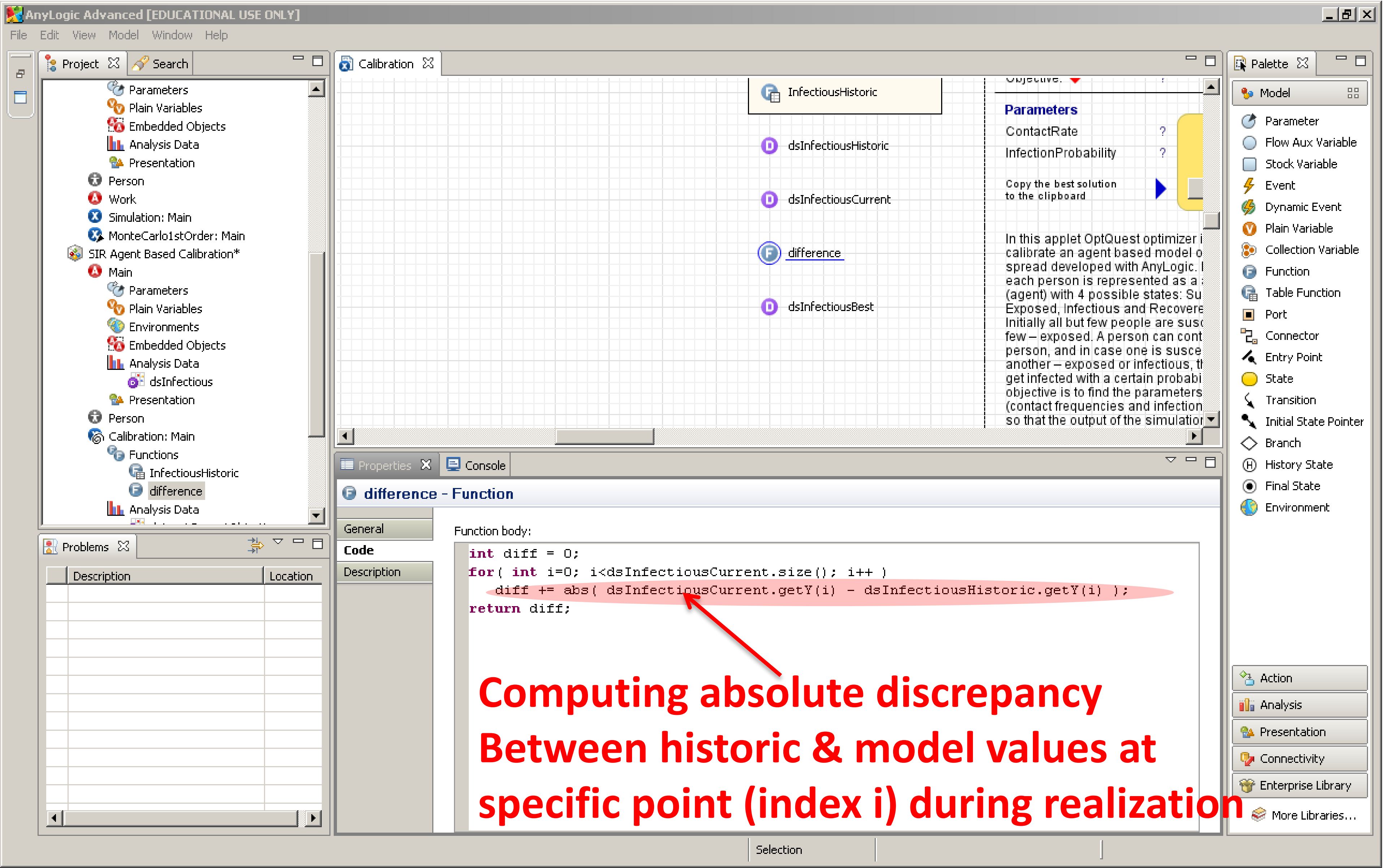
parameter	type	value		
		min	max	step
AverageI...uration	fixed	15		
ContactRate	continuous	0.5	3	0
Infection...bability	continuous	0.1	0.8	0
AreaSide	fixed	100		
TotalPopulation	fixed	10000		

Custom distance function

Varying these parameters

Defining a Payoff Function

Caveat: Here, Non-Analytic, Non-Concave

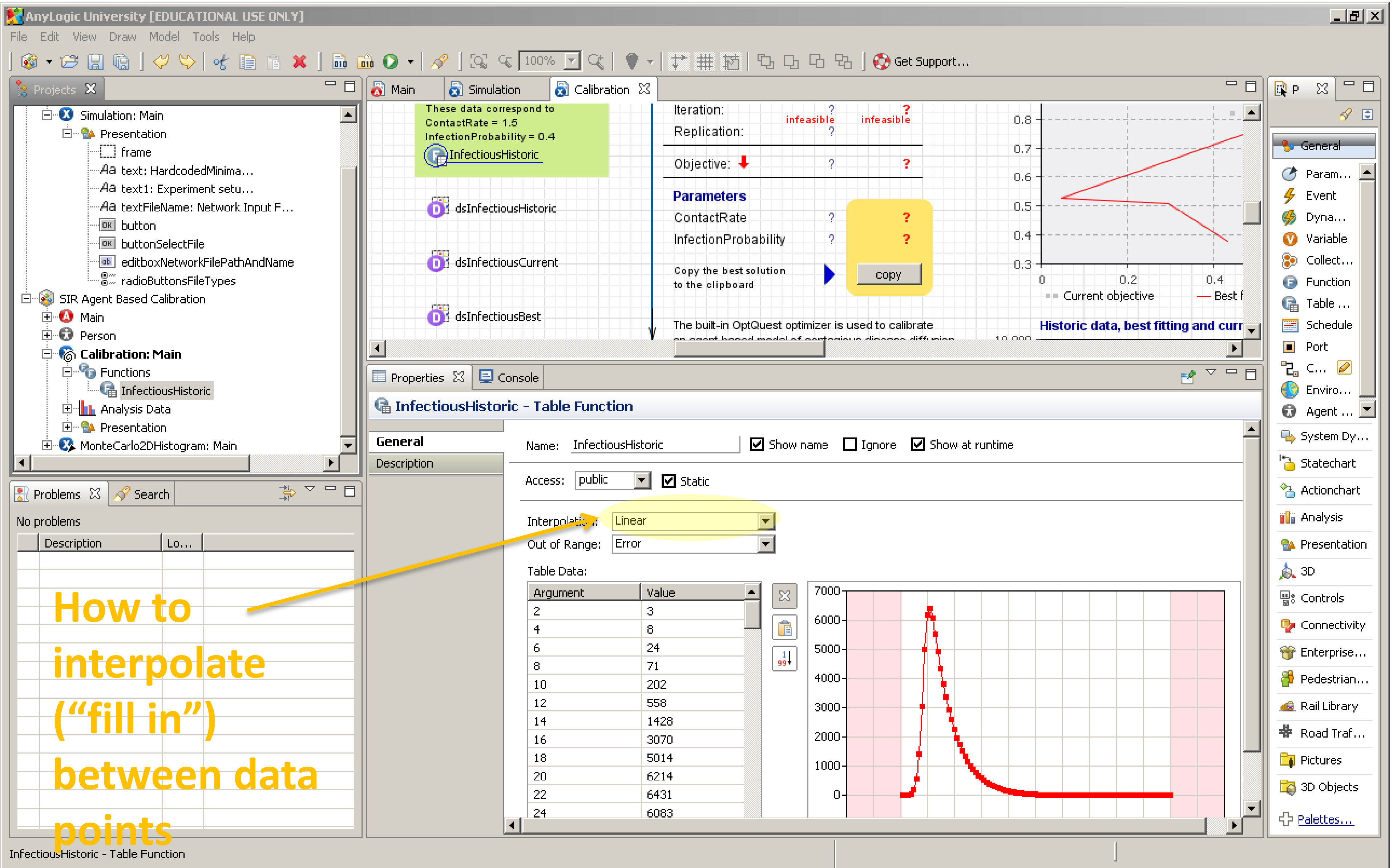


In this applet OptQuest optimizer i
calibrate an agent based model o
spread developed with AnyLogic. I
each person is represented as a
(agent) with 4 possible states: Su
Exposed, Infectious and Recovere
Initially all but few people are susc
few – exposed. A person can cont
person, and in case one is susce
another – exposed or infectious, t
get infected with a certain probabi
objective is to find the parameters
(contact frequencies and infection
so that the output of the simulatio

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

Computing absolute discrepancy
Between historic & model values at
specific point (index i) during realization

Historic Data Captured via Table Function



InfectiousHistoric - Table Function

Name: InfectiousHistoric ☑ Show name ☐ Ignore ☑ Show at runtime

Access: public ☑ Static

Interpolation: Linear

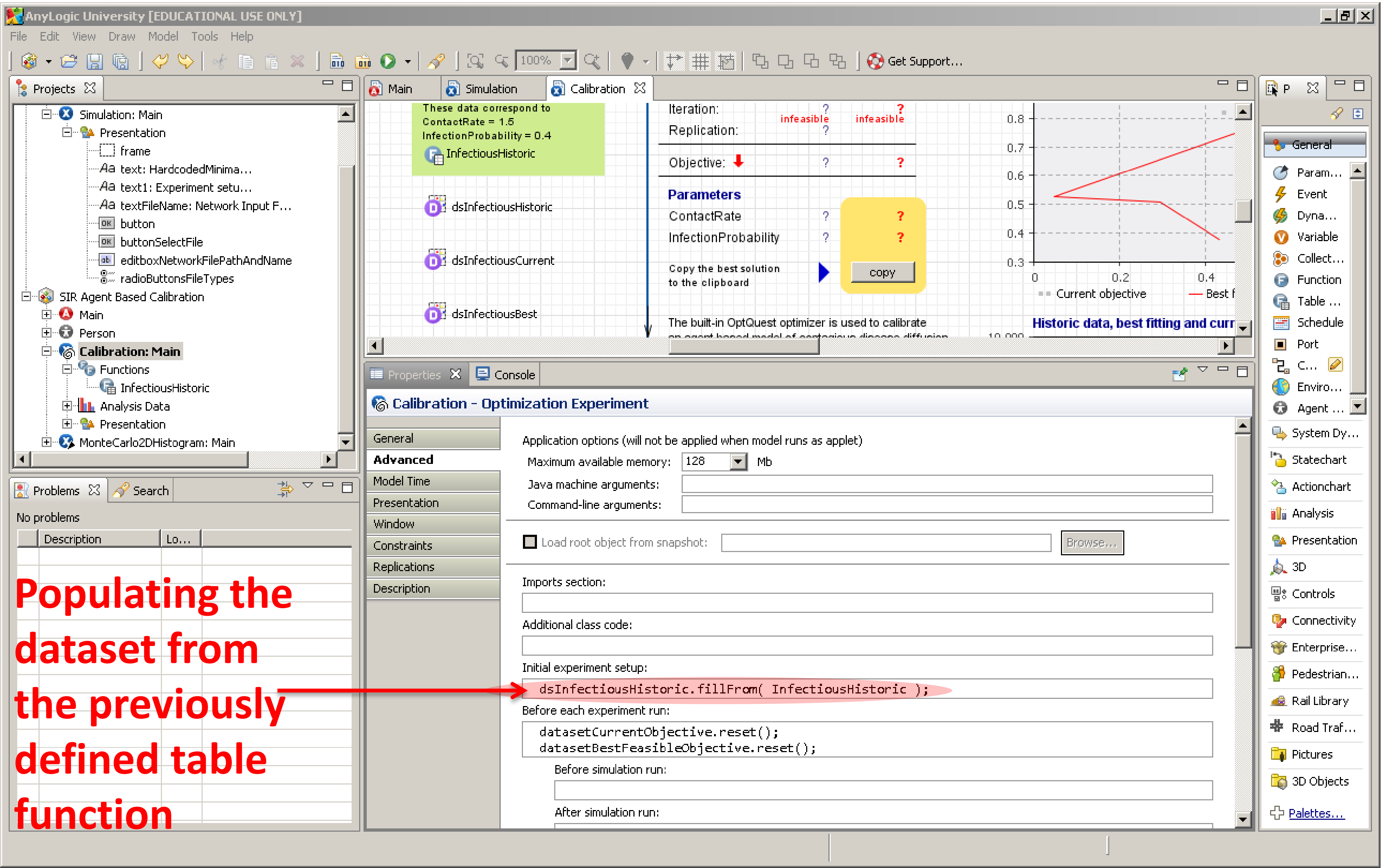
Out of Range: Error

Table Data:

Argument	Value
2	3
4	8
6	24
8	71
10	202
12	558
14	1428
16	3070
18	5014
20	6214
22	6431
24	6083

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

Populating a Dataset with Historic Data



Calibration - Optimization Experiment

General | **Advanced** | Model Time | Presentation | Window | Constraints | Replications | Description

Application options (will not be applied when model runs as applet)

Maximum available memory: 128 Mb

Java machine arguments:

Command-line arguments:

☐ Load root object from snapshot: Browse...

Imports section:

Additional class code:

Initial experiment setup:

```
dsInfectiousHistoric.fillFrom( InfectiousHistoric );
```

Before each experiment run:

```
datasetCurrentObjective.reset();
datasetBestFeasibleObjective.reset();
```

Before simulation run:

After simulation run:

Populating the dataset from the previously defined table function

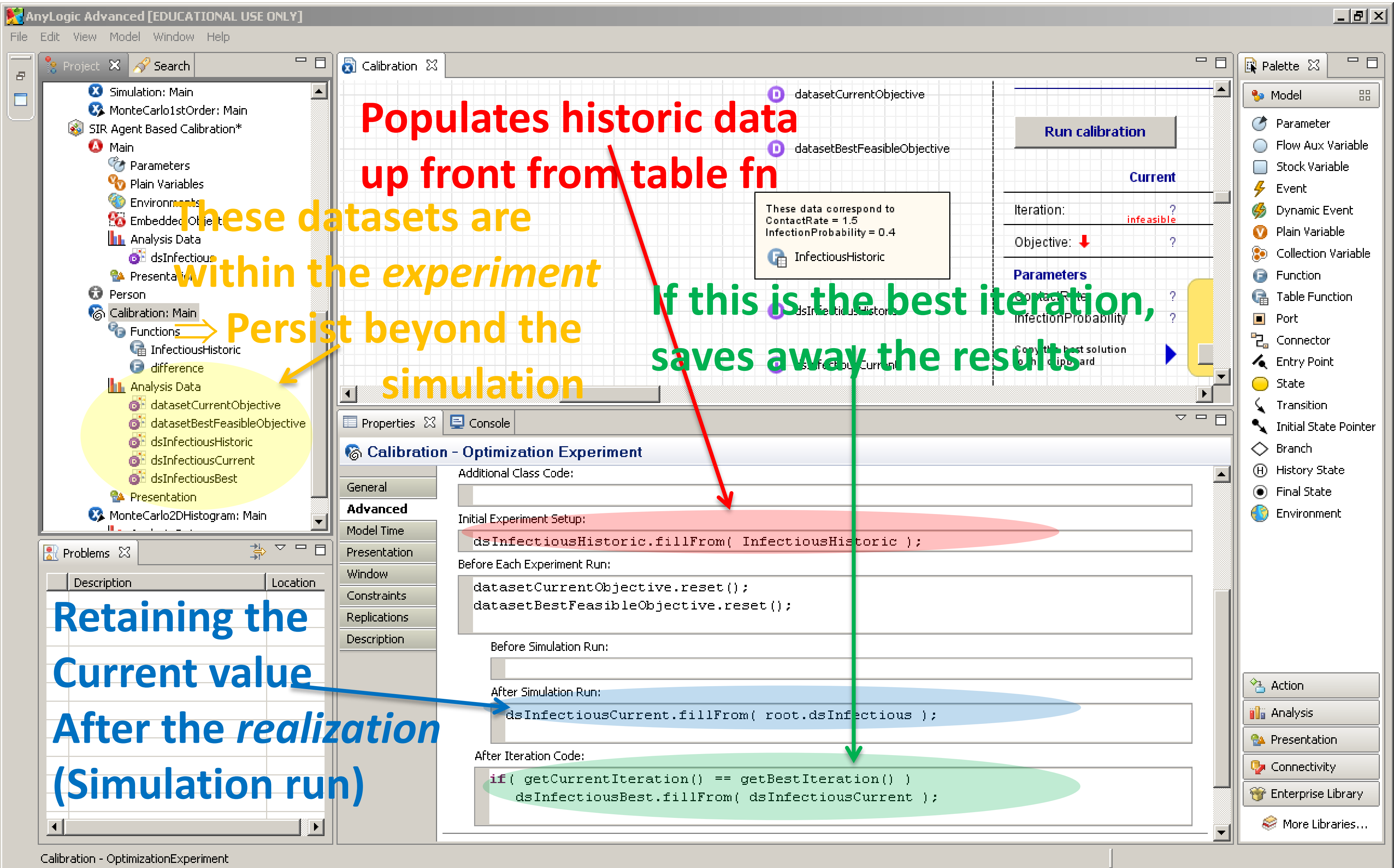
Stochastics in Agent-Based Models

- 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

Recall: Important Distinction (Declining Order of Aggregation)

- Experiment
  - Collection of simulations
- Simulation
  - Collection of replications that can yield findings across set of replications (e.g. mean value)
- Replication
  - One run of the model

Populating the Appropriate Datasets



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 );
```

Populates historic data up front from table fn

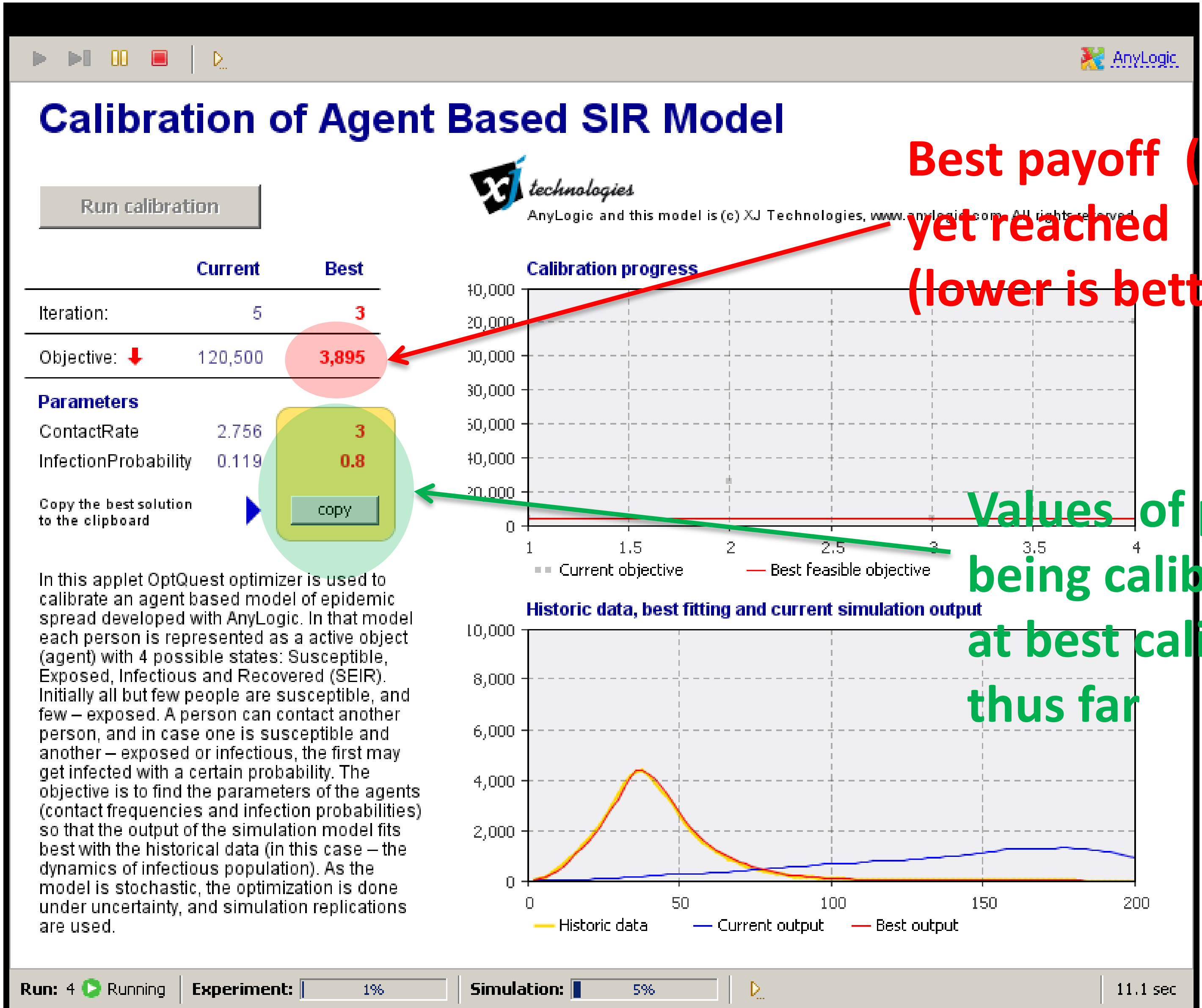
These datasets are within the *experiment*

Persist beyond the simulation

If this is the best iteration, saves away the results

Retaining the Current value After the *realization* (Simulation run)

Running Calibration in AnyLogic



Calibration of Agent Based SIR Model

AnyLogic and this model is (c) XJ Technologies, www.anylogic.com. All rights reserved.

	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 ▶ copy

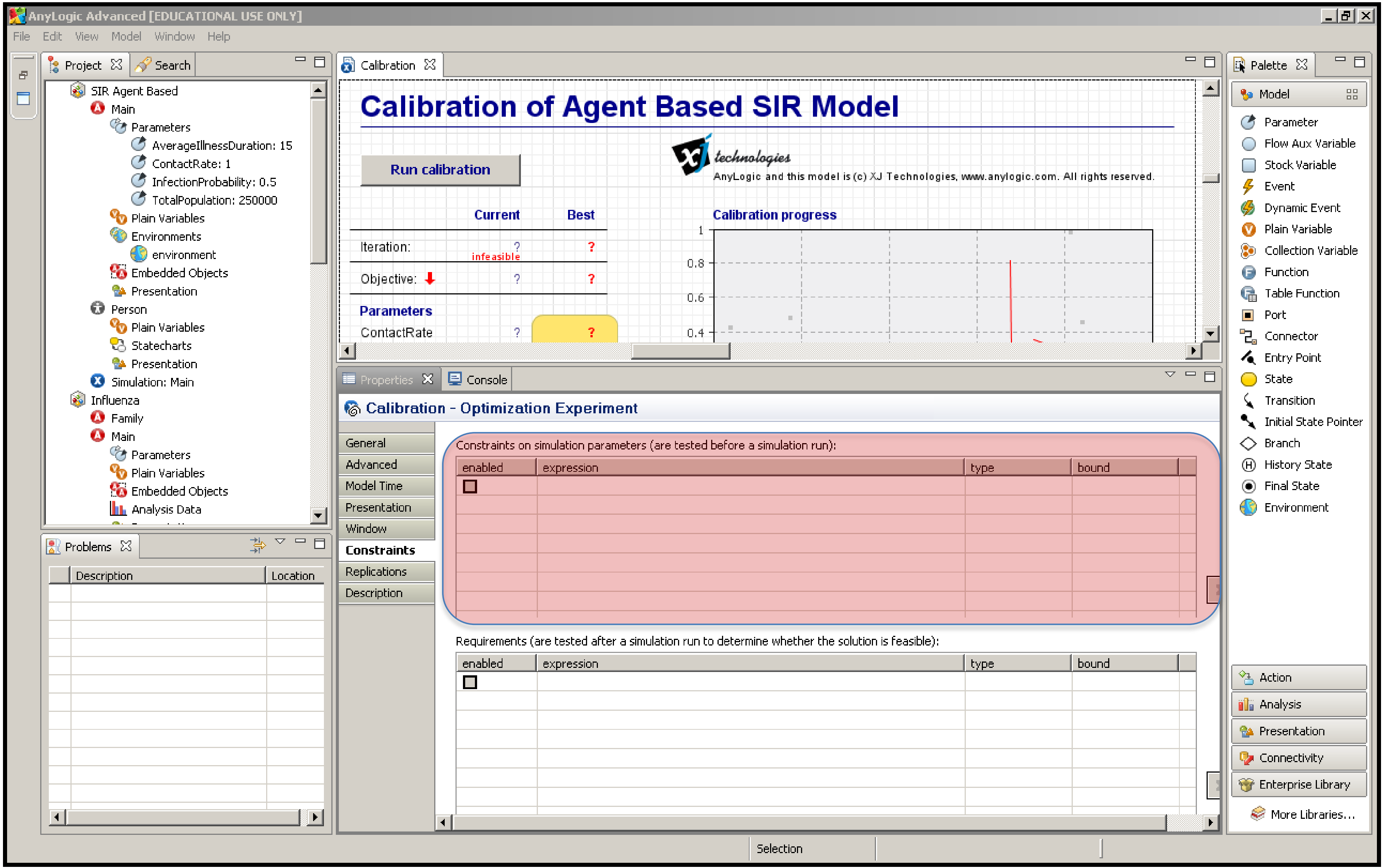
In this applet OptQuest optimizer is used to calibrate an agent based model of epidemic spread developed with AnyLogic. In that model each person is represented as a active object (agent) with 4 possible states: Susceptible, Exposed, Infectious and Recovered (SEIR). Initially all but few people are susceptible, 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.

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

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

Values of parameters being calibrated at best calibration thus far

Optimization Constraints – Tests on Legitimacy of Parameter Values



AnyLogic Advanced [EDUCATIONAL USE ONLY]

Calibration of Agent Based SIR Model

AnyLogic and this model is (c) XJ Technologies, www.anylogic.com. All rights reserved.

	Current	Best
Iteration:	? infeasible	?
Objective:	?	?

Parameters

ContactRate	?	?

SIR Agent Based > Main > Parameters: AverageIllnessDuration: 15; ContactRate: 1; InfectionProbability: 0.5; TotalPopulation: 250000

Calibration - Optimization Experiment

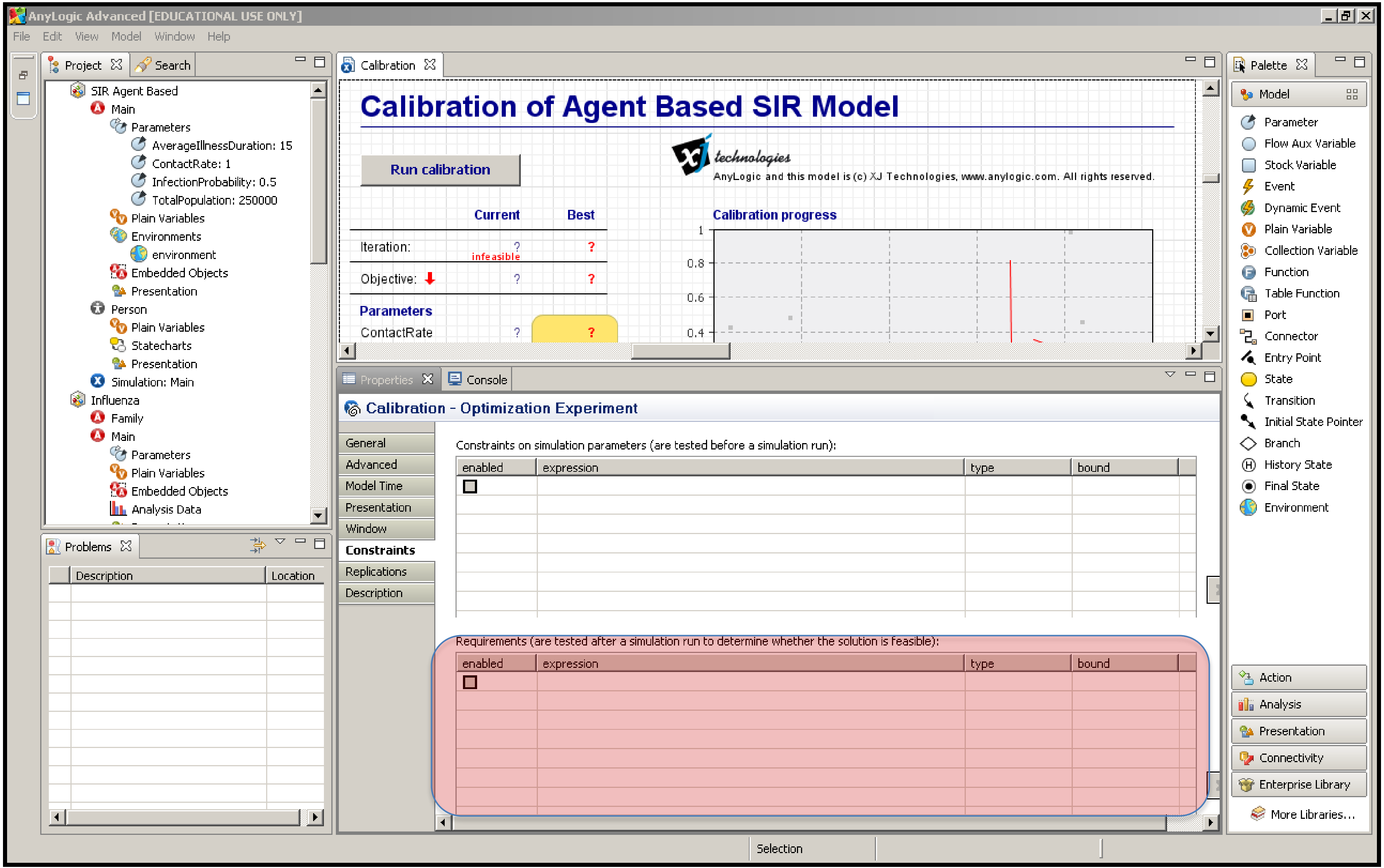
Constraints on simulation parameters (are tested before a simulation run):

enabled	expression	type	bound
□			

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

enabled	expression	type	bound
□			

Optimization Requirements – Tests to Sense Validity of Emergent Results



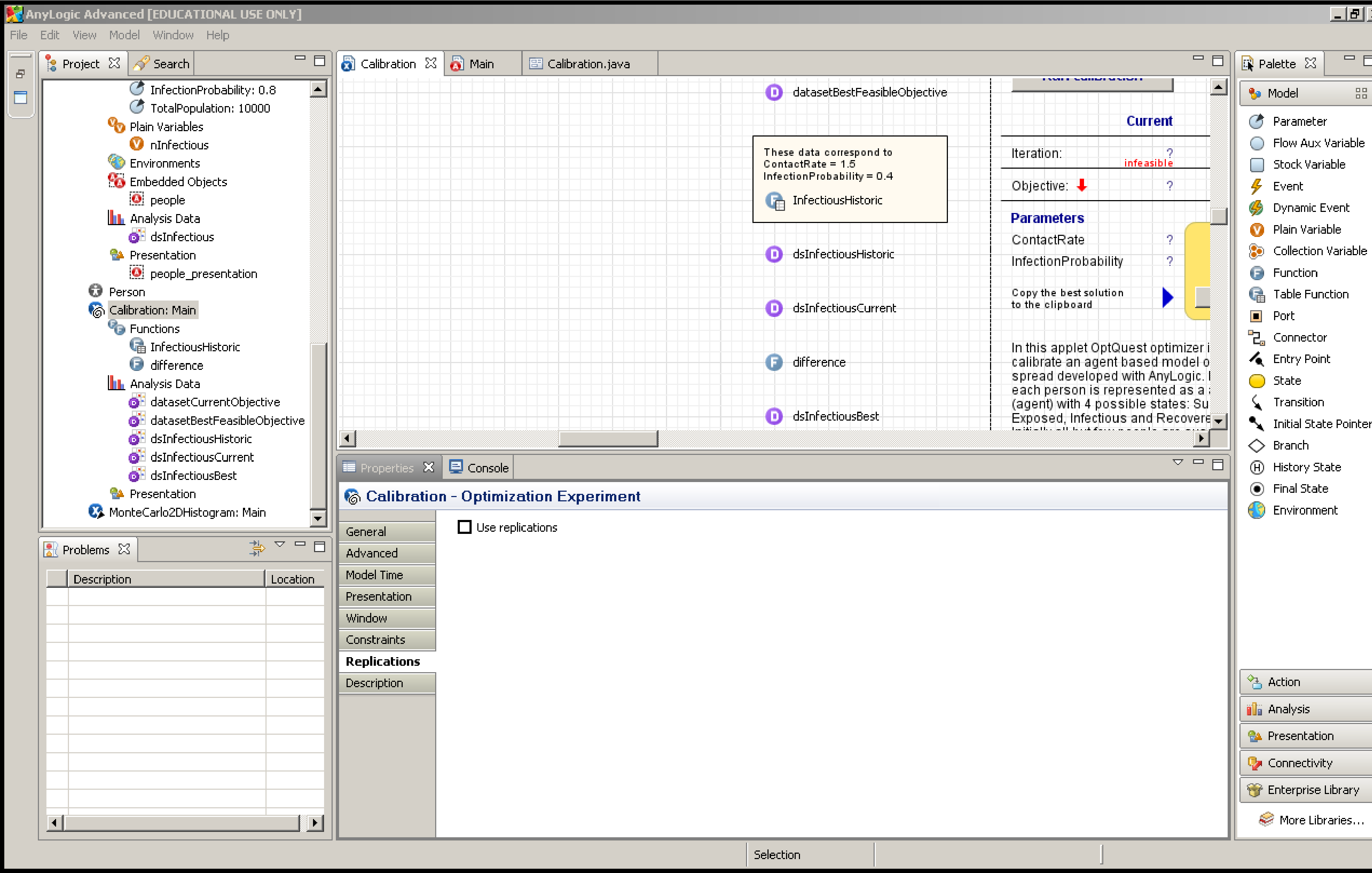
Constraints on simulation parameters (are tested before a simulation run):

enabled	expression	type	bound
□			

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

enabled	expression	type	bound
□			

Enabling Multiple Realizations (“Replications”,”Runs”) per Iteration



These data correspond to ContactRate = 1.5 InfectionProbability = 0.4

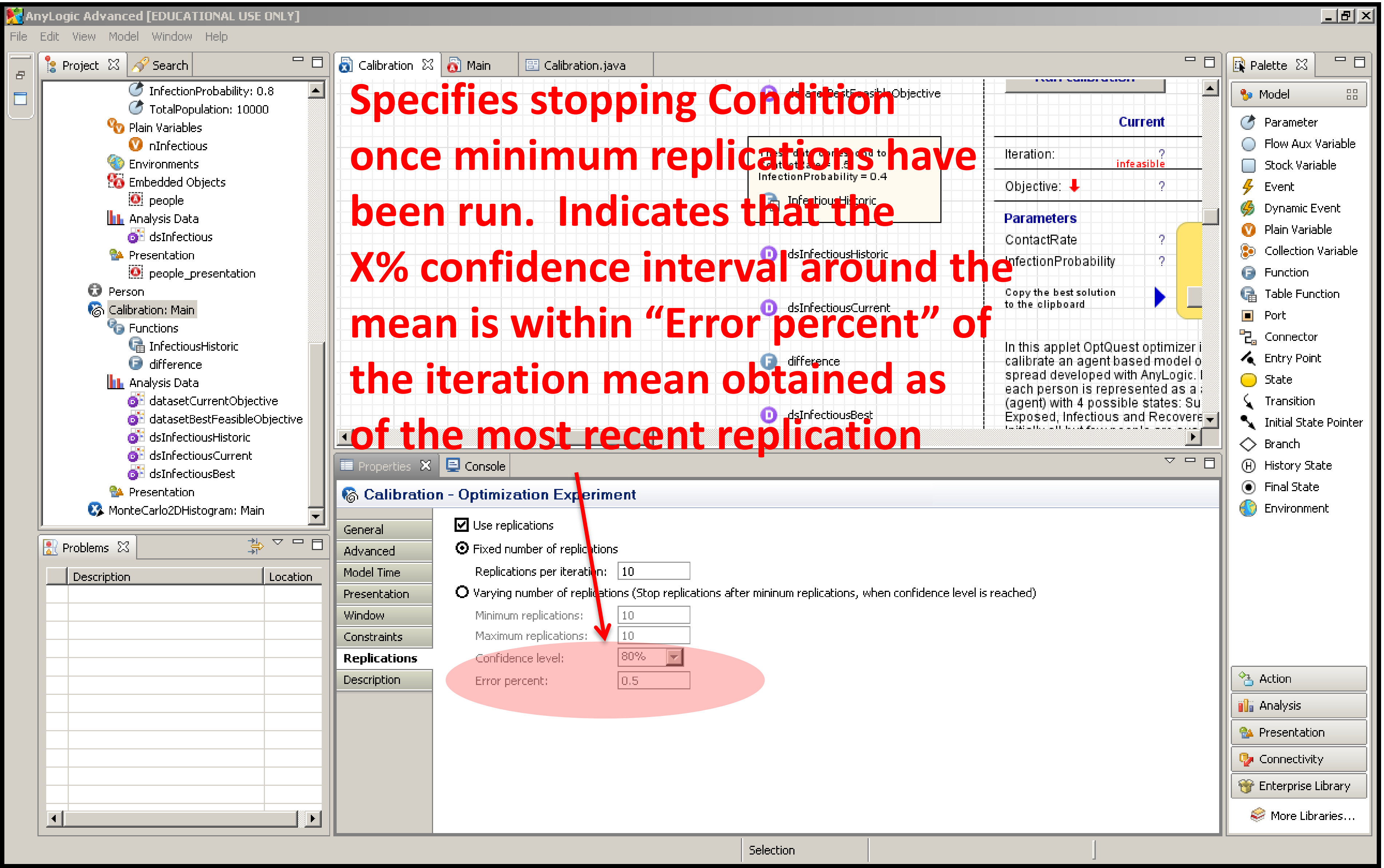
In this applet OptQuest optimizer i
calibrate an agent based model o
spread developed with AnyLogic. I
each person is represented as a
(agent) with 4 possible states: Su
Exposed, Infectious and Recovere

Calibration - Optimization Experiment

Replications

□ Use replications

Fixed Number of Replications per Iteration



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

Calibration - Optimization Experiment

☑ Use replications

● Fixed number of replications

Replications per iteration: 10

○ Varying number of replications (Stop replications after mininum replications, when confidence level is reached)

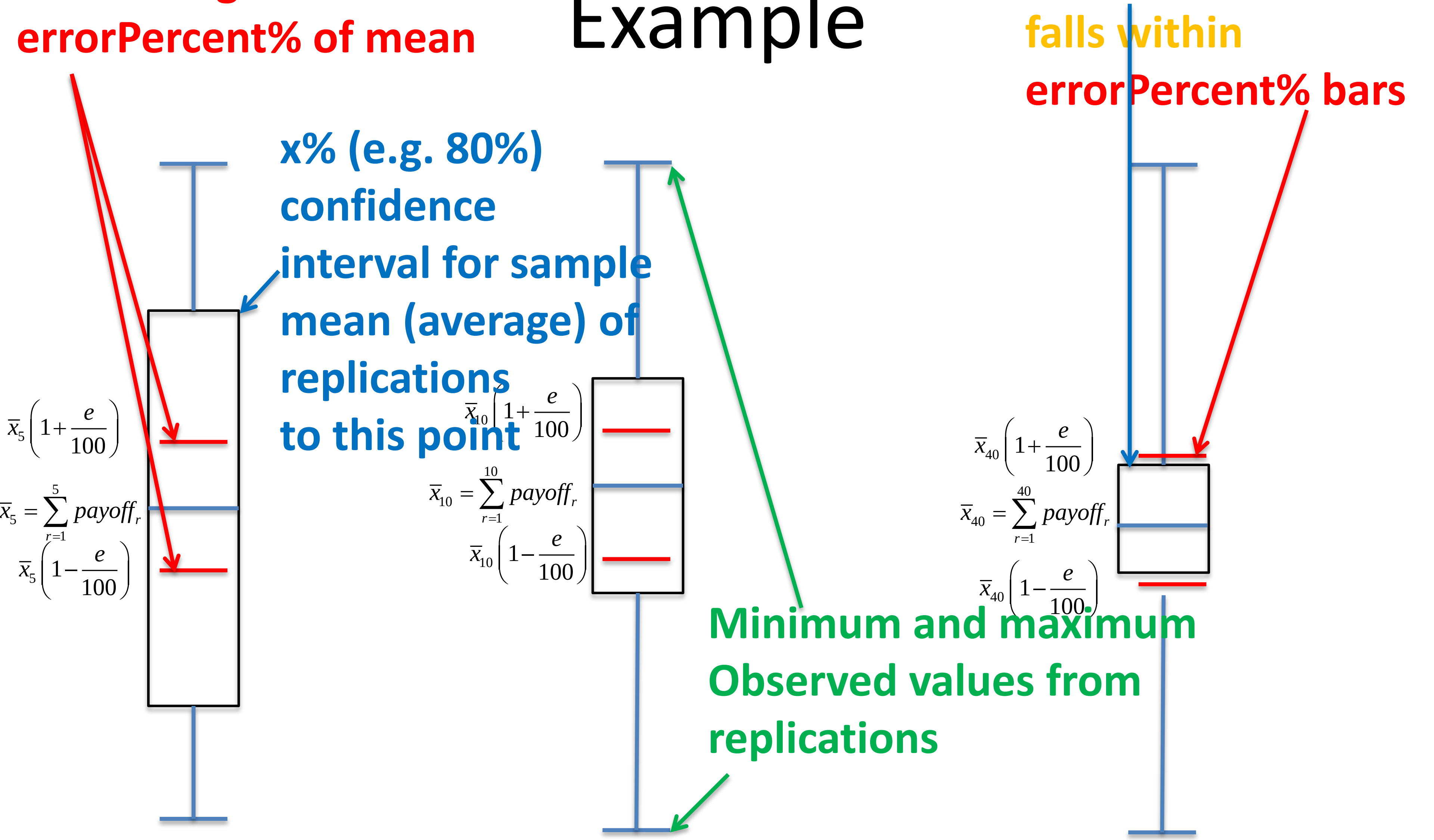
Minimum replications: 10

Maximum replications: 10

Confidence level: 80%

Error percent: 0.5

Example



Bars showing that delineating values within errorPercent% of mean

x% (e.g. 80%) confidence interval for sample mean (average) of replications to this point

Minimum and maximum Observed values from replications

Terminates because confidence interval falls within errorPercent% bars

$\overline{x}_5\left(1+\frac{e}{100}\right)$

$\overline{x}_5=\sum_{r=1}^{5} payoff_r$

$\overline{x}_5\left(1-\frac{e}{100}\right)$

$\overline{x}_{10}\left(1+\frac{e}{100}\right)$

$\overline{x}_{10}=\sum_{r=1}^{10} payoff_r$

$\overline{x}_{10}\left(1-\frac{e}{100}\right)$

$\overline{x}_{40}\left(1+\frac{e}{100}\right)$

$\overline{x}_{40}=\sum_{r=1}^{40} payoff_r$

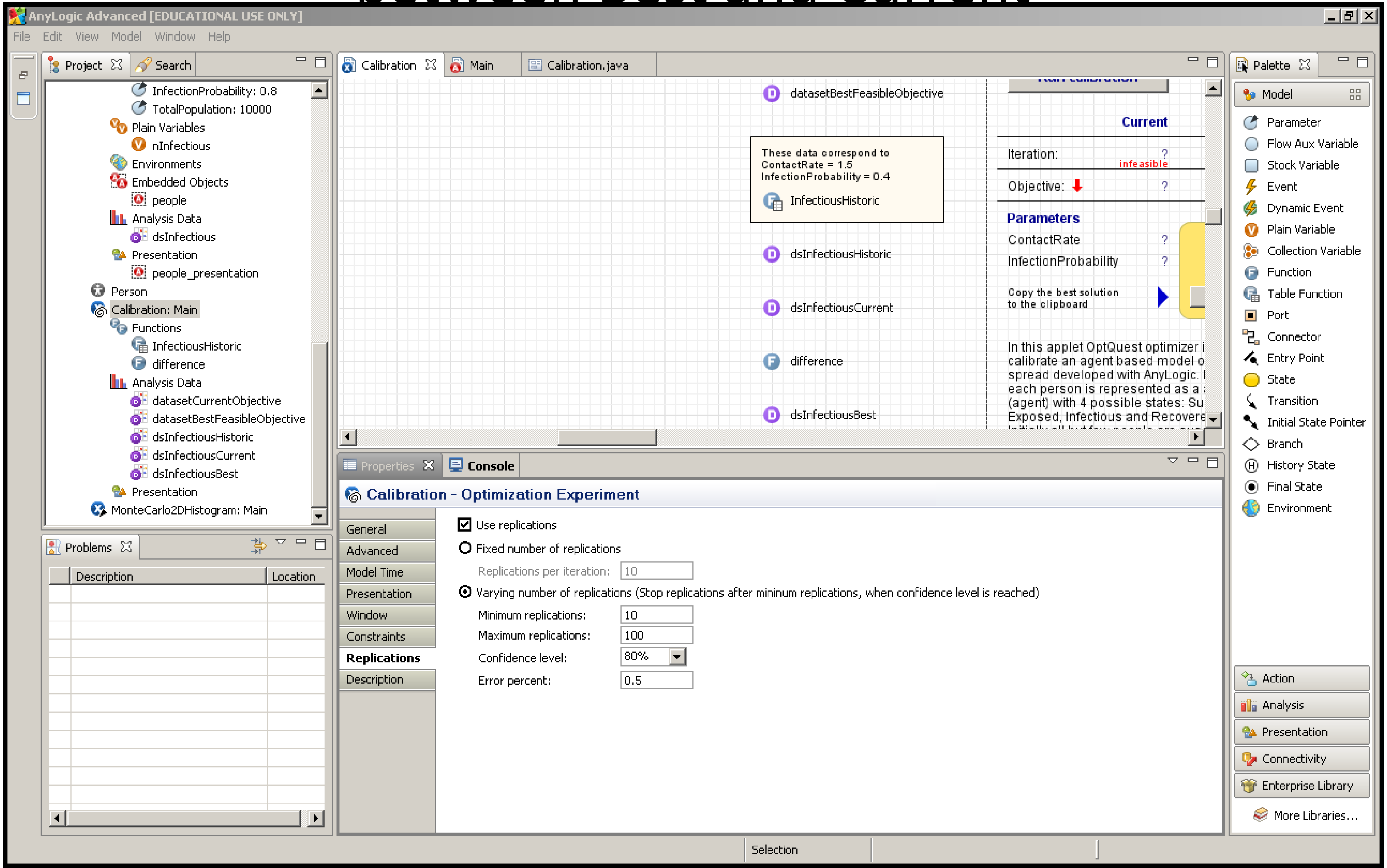
$\overline{x}_{40}\left(1-\frac{e}{100}\right)$

After 5 replications

After 10 replications

After 40 replications
Terminates

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



AnyLogic Advanced [EDUCATIONAL USE ONLY]

Calibration - Optimization Experiment

Replications

☑ Use replications

○ Fixed number of replications

Replications per iteration: 10

● Varying number of replications (Stop replications after mininum replications, when confidence level is reached)

Minimum replications: 10

Maximum replications: 100

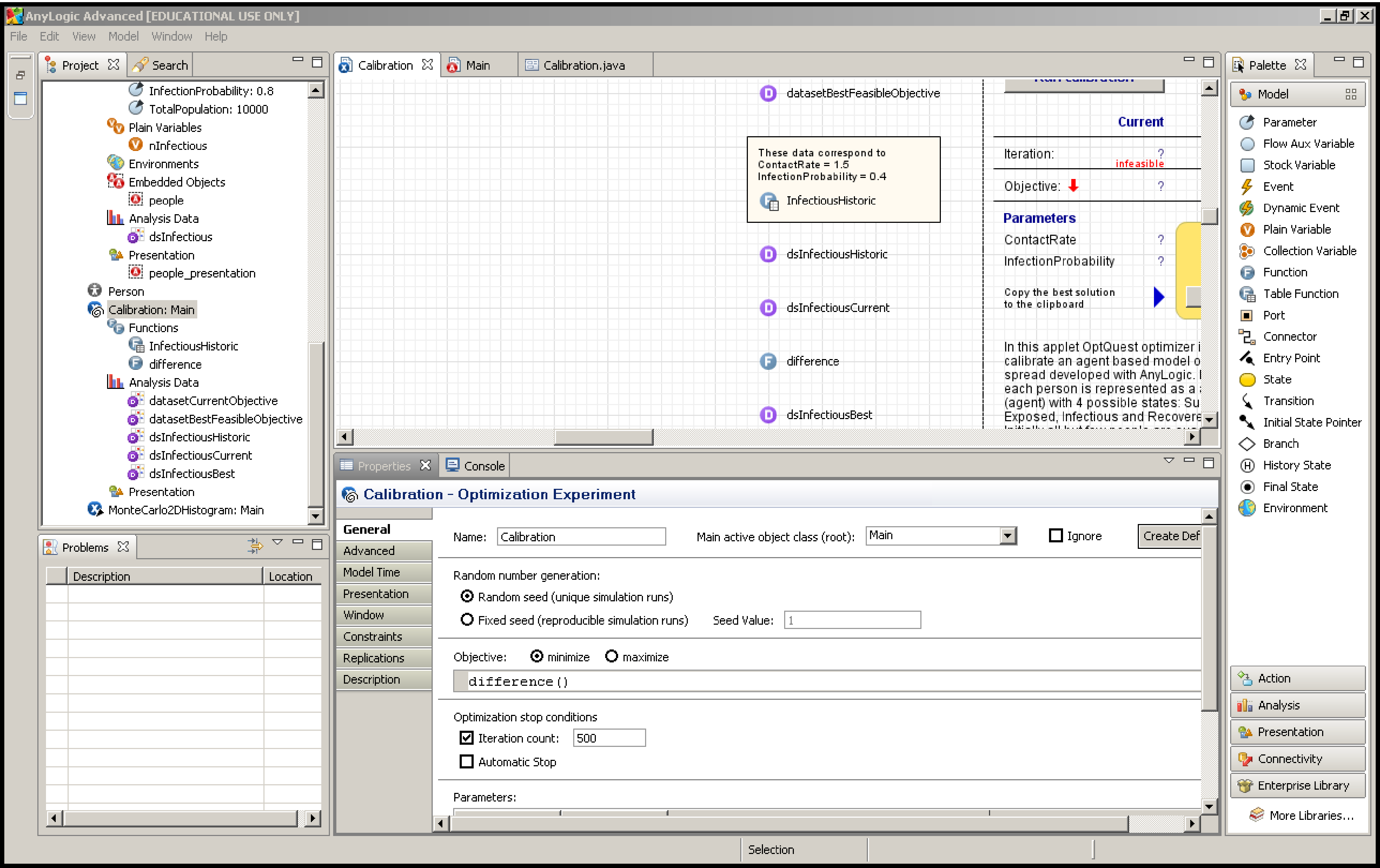
Confidence level: 80%

Error percent: 0.5

These data correspond to ContactRate = 1.5 InfectionProbability = 0.4

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

Enabling Random Variation Between Realizations (“Replications”)



These data correspond to
ContactRate = 1.5
InfectionProbability = 0.4

Copy the best solution to the clipboard

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

Calibration - Optimization Experiment

Name: Calibration   Main active object class (root): Main   ☐ Ignore

Random number generation:

● Random seed (unique simulation runs)

○ Fixed seed (reproducible simulation runs)   Seed Value: 1

Objective: ● minimize ○ maximize

```
difference()
```

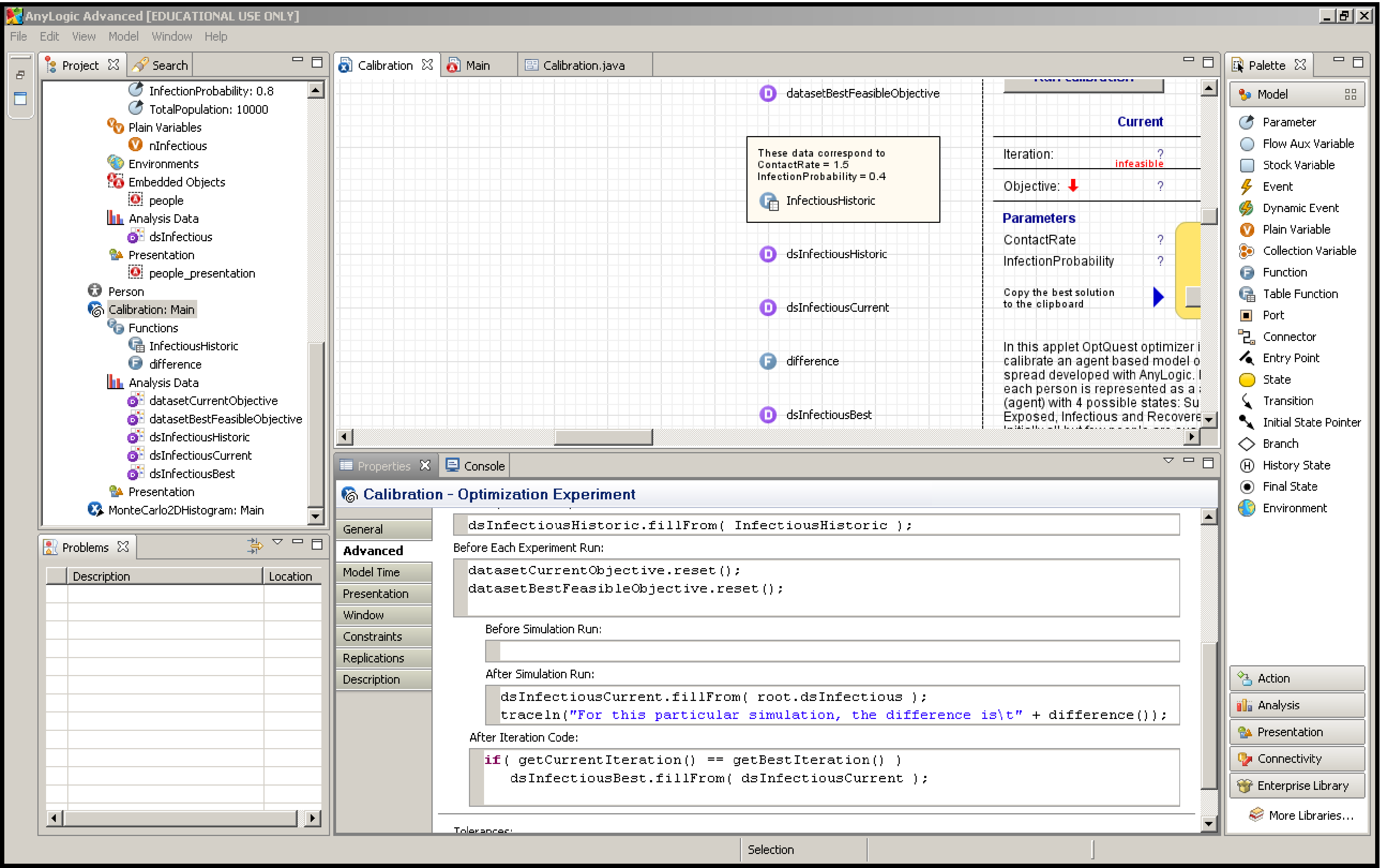
Optimization stop conditions

☑ Iteration count: 500

☐ Automatic Stop

Parameters:

Understanding Replications: Report Results for Each Replication!



Calibration - Optimization Experiment

```
dsInfectiousHistoric.fillFrom( InfectiousHistoric );
```

Before Each Experiment Run:

```
datasetCurrentObjective.reset();
datasetBestFeasibleObjective.reset();
```

Before Simulation Run:

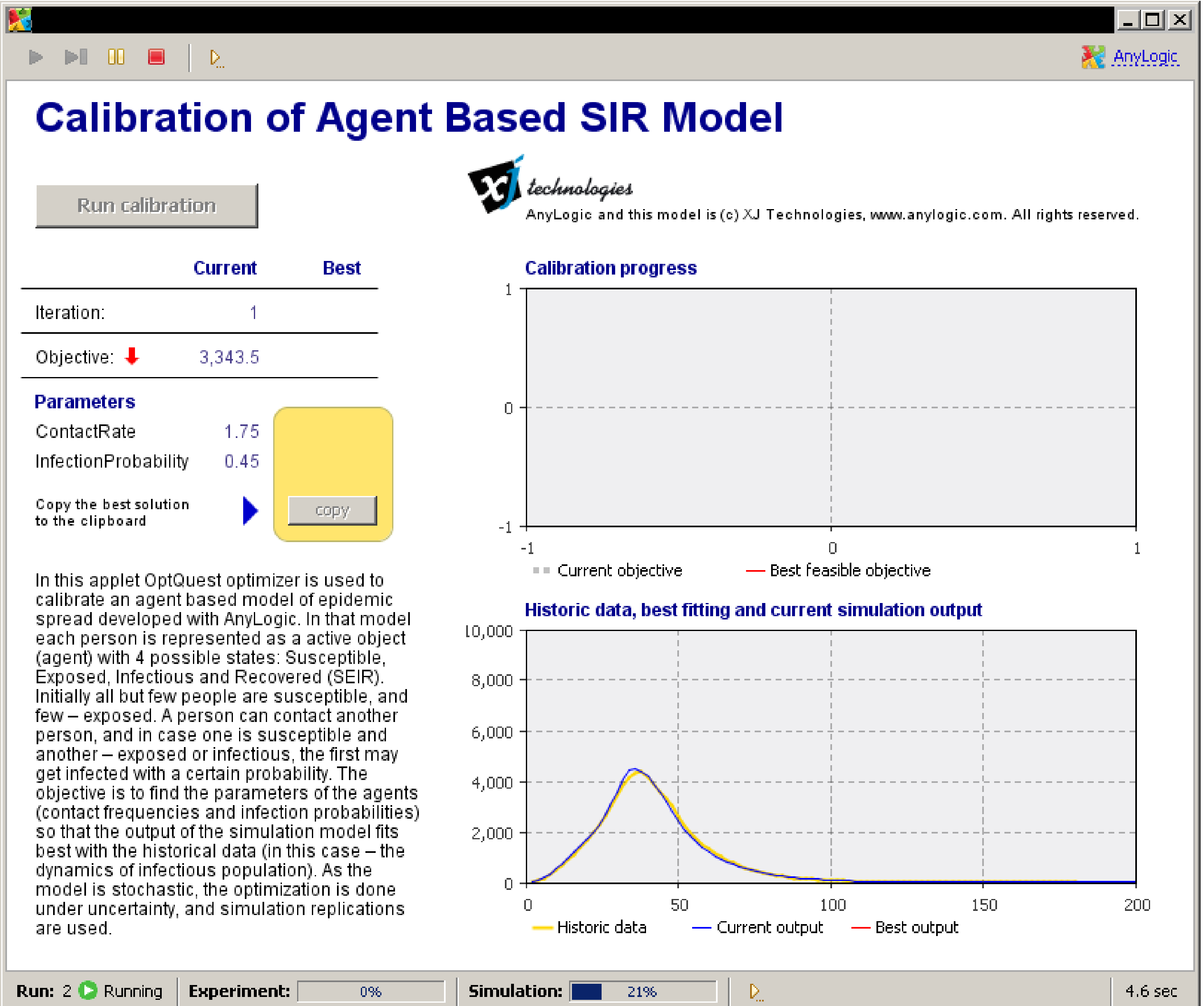
After Simulation Run:

```
dsInfectiousCurrent.fillFrom( root.dsInfectious );
traceln("For this particular simulation, the difference is\t" + difference());
```

After Iteration Code:

```
if( getCurrentIteration() == getBestIteration() )
  dsInfectiousBest.fillFrom( dsInfectiousCurrent );
```

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



Calibration of Agent Based SIR Model

AnyLogic and this model is (c) XJ Technologies, www.anylogic.com. All rights reserved.

	Current	Best
Iteration:	1	
Objective: ⬇	3,343.5	

Parameters

ContactRate	1.75
InfectionProbability	0.45

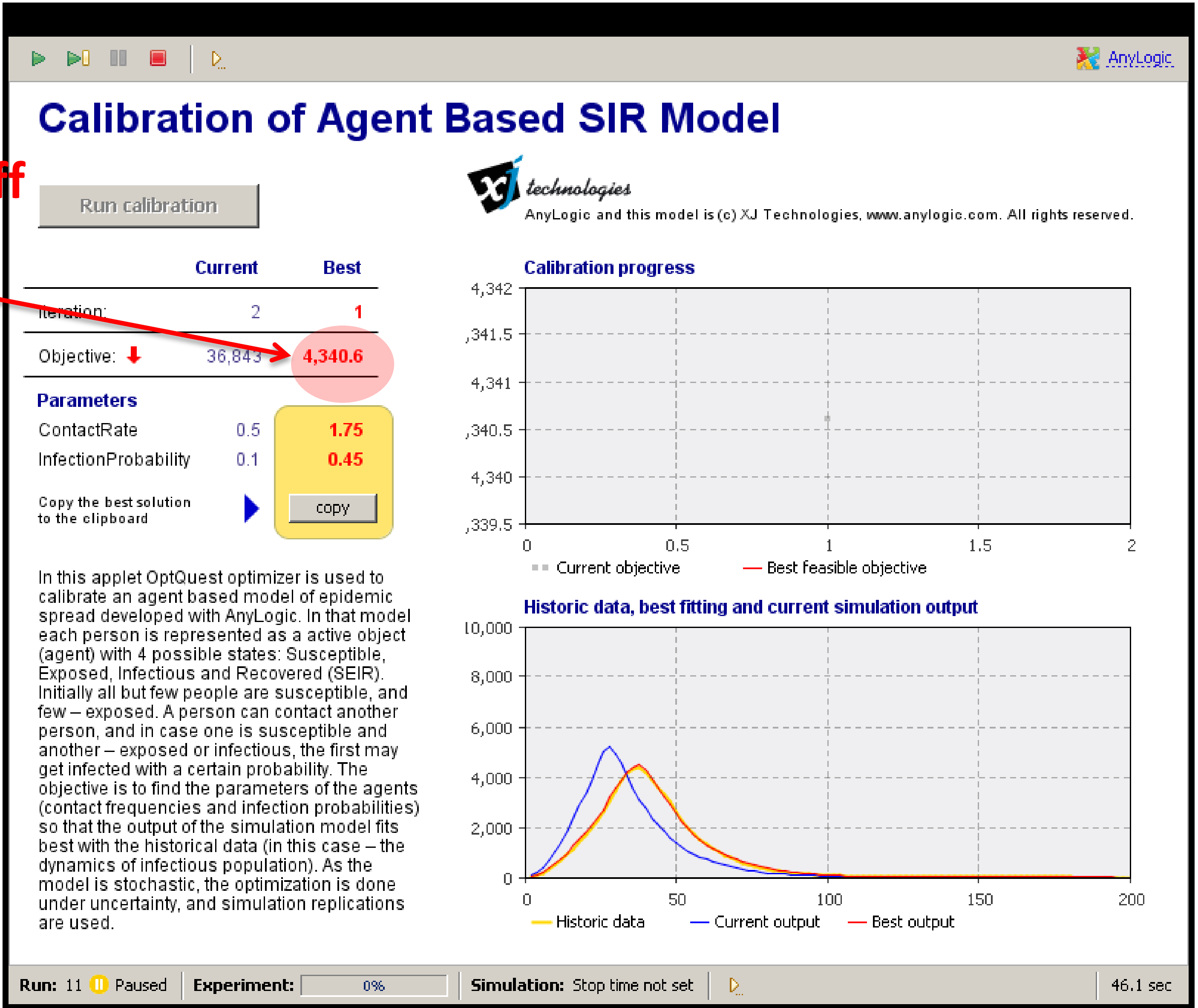
Copy the best solution to the clipboard ▶

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.

Run: 2 Running | Experiment: 0% | Simulation: 21% | 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?



Calibration of Agent Based SIR Model

Run calibration

AnyLogic and this model is (c) XJ Technologies, www.anylogic.com. All rights reserved.

	Current	Best
Iteration:	2	1
Objective: ↓	36,843	4,340.6

Parameters

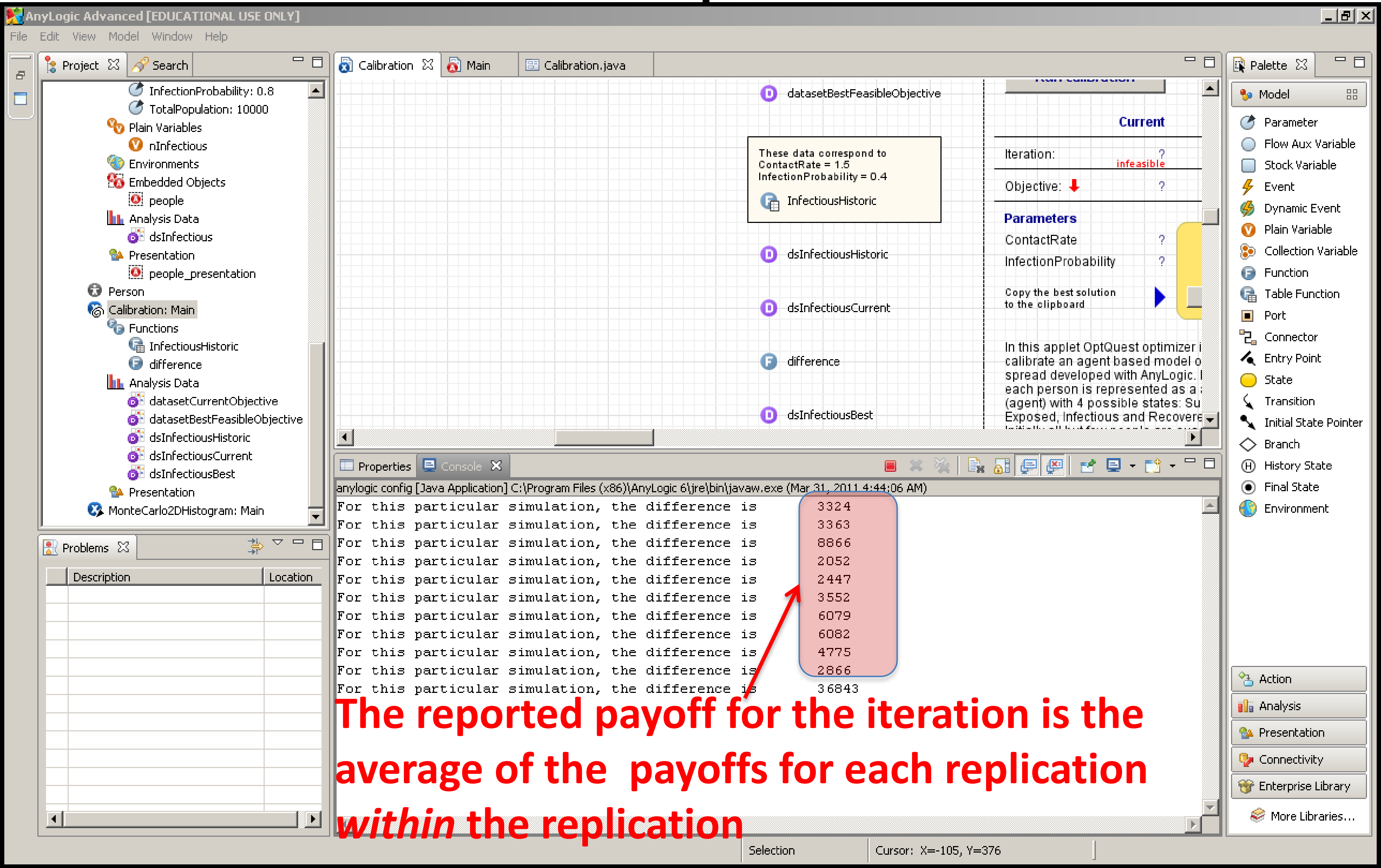
ContactRate	0.5	1.75
InfectionProbability	0.1	0.45

Copy the best solution to the clipboard ▶ copy

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.

Run: 11 Paused **Experiment:** 0% **Simulation:** Stop time not set 46.1 sec

Output



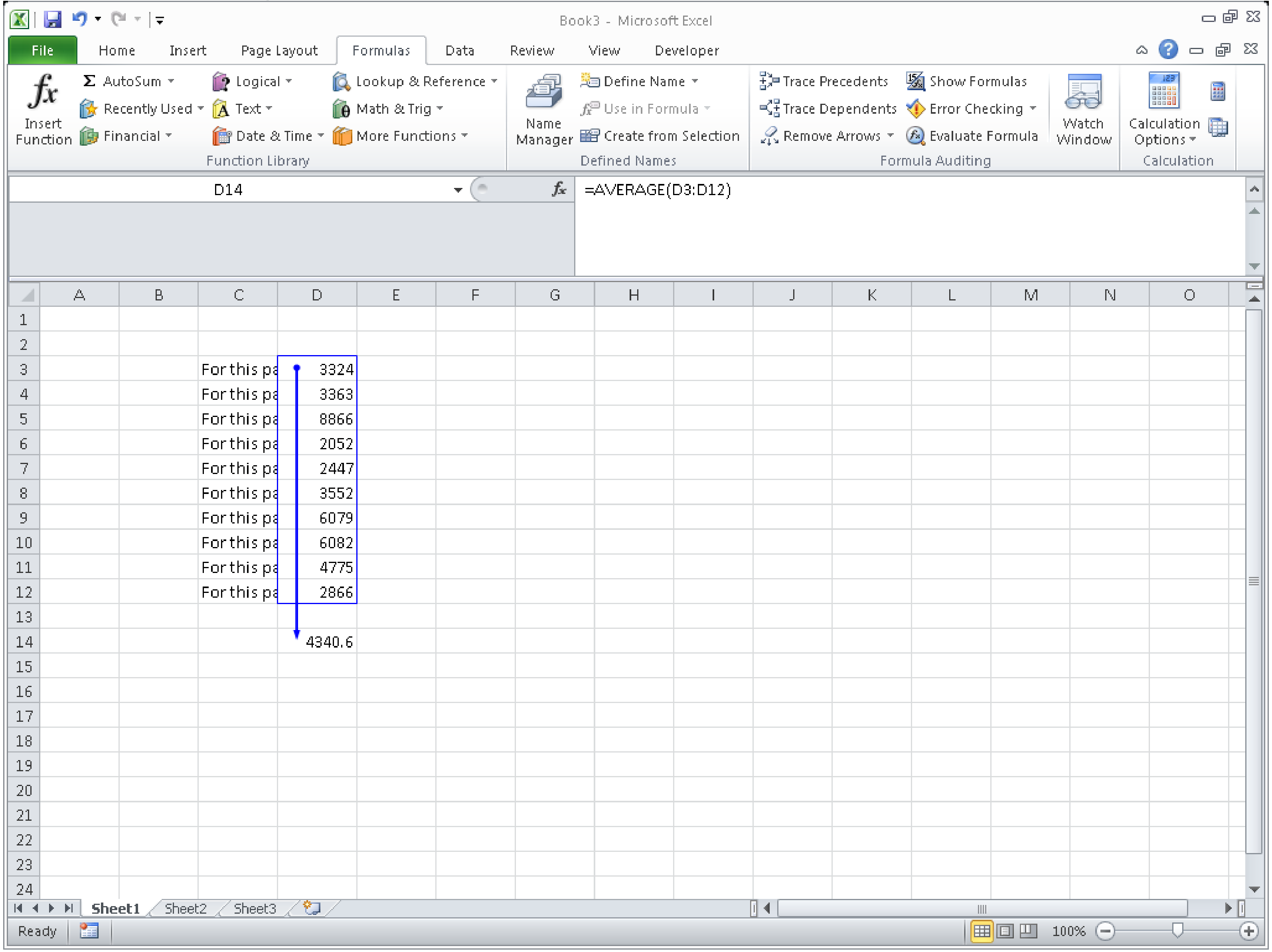
These data correspond to ContactRate = 1.5 InfectionProbability = 0.4

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

```
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
```

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!



D14 =AVERAGE(D3:D12)

	A	B	C	D
3			For this pa	3324
4			For this pa	3363
5			For this pa	8866
6			For this pa	2052
7			For this pa	2447
8			For this pa	3552
9			For this pa	6079
10			For this pa	6082
11			For this pa	4775
12			For this pa	2866
13				
14				4340.6

Considerations

- Adding constraints helps increase identifiability (selection of realistic best fit)
- Adding parameters to tune leads to larger space to explore
- Adding too many parameters to tune can lead to underdetermined situation
- All fits are within constraints of model

Dealing with Calibration Problems: Experiments

- Try to "outsmart" calibration
  - Adopt best parameter values from calibration
  - Try to adjust parameters to do better than calibration
    - If is better, it may be that the parameter space is too large, or that the range constraints are too tight
    - Typically this does not do as well: Opportunity to learn
      - Model not respond in the way that anticipated to parameter change
      - May just shift the discrepancy from one variable to another
        - Assumptions of model structure/values may not permit both variables to simultaneously match well!
- Set very high weight on thing that want to match, and see other matches
- Set all other weights to 0 (see if can possibly match)

Dealing with Calibration Problems: Additional Experiments

- Increase parameter range
- Increase # of parameters
- Examine impact of changed model structure
- Run for larger number of optimization runs
- Find other estimates for uncertain parameters

Important Cross-Checks: Uniqueness

- Are the calibration values Unique? If so, good; if not,
  - Do they give the same underlying interpretation?
  - Do the different interpretations lead to parameters that "trade off" in some structured way?
- Ways of addressing significantly different interpretations
  - Collect more primary data!
  - Impose additional constraints (in terms of time series, etc.)
  - Simplify model
  - Find other estimates for uncertain parameters

Important Cross-Checks: Binding Constants

- Look for calibrated parameter values that are at the edges of their permissible ranges
  - If “best” value is at the edge of the range, it may be that even better calibrations would have been possible if continuing in that direction
- To deal with those at the edge
  - Relax constraints
  - Collect more data on plausible values
  - Question model structure

Capturing Parameter Interdependencies in Calibration

- If we want parameter B adjusted during calibration to be at least as big as parameter A
  - In vensim, we can’t enforce this constraint using the typical calibration machinery, because the range limits for parameters must be constants
  - we can accomplish this by calibrating only parameter A, and a parameter representing the ratio B/A.
- If we want to adjust two or more parameters such that they still sum to 1 (e.g. fraction of initial population in each of *n* or more stocks), we can adjust each of *n* non-normalized weights, and then take the corresponding normalized amount to be frac. falling in that category

Calibrating Initial Conditions

- The initial conditions can be one of the best values to calibrate
- Sometimes need to divide a fixed population into several stocks

Calibration & Regression: Similarities & Differences

- Model calibration is similar to regression in that we are seeking to find the parameter values allowing the best match of model & data
  - As in non-linear regression, for non-linear simulation models no “closed form” solution of best parameter values is possible ⇒ optimization is required
- A big difference:
  - **Regression models**: the “functional form” (dependence of model output on par’ms/indep vars) is given explicitly
  - **Simulation models**: behavior is only *implicitly* specified (e.g. via giving differentials); model output is a complex resultant (even emergent) property of structure