

Collecting, Outputting & Inputting Data in AnyLogic

Nathaniel Osgood

Using Modeling to Prepare for Changing
Healthcare Needs

Duke-NUS

April 16, 2014

Recording of Results

- A frequent modeler need is to record some components of model state over time
 - State variables (e.g. stocks)
 - States of agents
 - Summaries of model state
 - We informally term this a “trajectory file”
- *Trajectory recording is supported in higher AnyLogic versions*
- All versions of AnyLogic allow for
 - Definition of *DataSets* that record recent values of parameters
 - Statistics summarizing model state
 - Reporting on values of data sets as a graph or table

What to Record?

- Current model state
 - Aggregated (“Stocks”) or disaggregated
- Changes in model state (flow statistics)
- History
- Model version (model structure)
- Assumptions (parameter values)
 - Interventions

Common Ways to Accumulate Flow Statistics

- Have a variable that
 - is zeroed out at the beginning of each time interval of interest
 - Is accumulated across that interval of interest
 - Is stored away at the end of that interval interest
- Option2
 - Have a variable that gives the cumulative number of events that have occurred
 - Subtract off the value of that variable from the beginning of the interval and store away the difference

Techniques for Outputting Data

- Ad-Hoc Exports from variables
- Manual copies from visible datasets
- Export to files
- Writing to console
- Export to databases
- [AnyLogic Professional] Dataset archiving
- Capturing images of graphs

Cross-Method Output Tips

- A convenient mechanism is to periodically output data using events (e.g. every time unit)
- Beyond output, be sure to save information on context of run
 - Model version (Use unique id that increment whenever change model)
 - Parameter assumptions
 - Intention
- Think carefully about whether want to save away intermediate data



Hands on Model Use Ahead



Load Sample Model:

SIR Agent Based Calibration

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

Techniques for Collecting & Outputting Data

- Ad-Hoc Exports from variables
- Pre-Prepared methods
 - Statistics
 - Charts
 - Manual copies from visible datasets
 - Export to files
 - Writing to console
 - Export to databases
 - [AnyLogic Professional] Dataset archiving
 - Capturing images of graphs

Add an Experiment

Experiment
Select an experiment type, specify a name and choose a root (top-level) active object.

Name:

Main active object class (root):

Experiment Type:

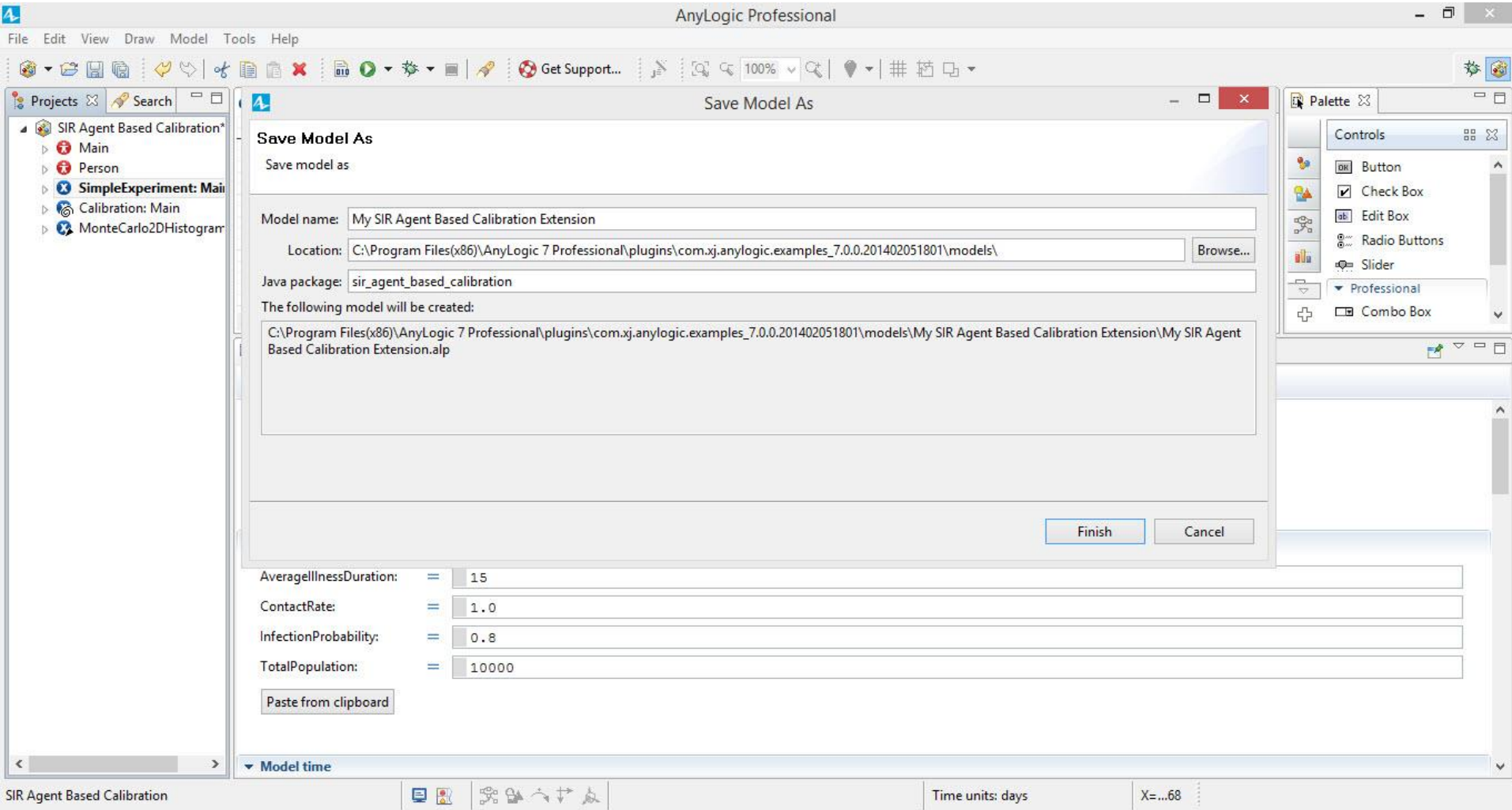
- Simulation
- Optimization
- Parameter Variation
- Compare Runs
- Monte Carlo
- Sensitivity Analysis
- Calibration
- Custom

Performs model runs with specified parameters, supports virtual and real-time modes, animation, and model debugging

Copy model time settings from:

< Back Next > **Finish** Cancel

Save the Resulting Model (To Avoid Overwriting the Other Model)



Run the Experiment (To Verify Functionality)

The screenshot displays the AnyLogic Professional software interface. The main window is titled "AnyLogic Professional" and contains several panels:

- Projects:** A tree view on the left showing the project structure for "My SIR Agent Based Calibration Extension", including sub-projects like "Main", "Person", "SimpleExperiment: Main", "Calibration: Main", and "MonteCarlo2DHistogram: Main".
- Recent Experiment:** A list of recent experiments in the center, including "My SIR Agent Based Calibration Extension / SimpleExperiment", "My SIR Agent Based Calibration Extension / Calibration", and "My SIR Agent Based Calibration Extension / MonteCarlo2DHistogram". A "Run" button is visible below this list.
- Palette:** A panel on the right containing various controls such as "Button", "Check Box", "Edit Box", "Radio Buttons", "Slider", and "Combo Box".
- Properties:** A panel at the bottom showing the properties for the selected model, "My SIR Agent Based Calibration Extension - Model". It includes fields for "Name" (My SIR Agent Ba:), "Model time units" (days), and "System of measurement" (Metric).
- Dependencies:** A section below the properties panel titled "Dependencies" with a table for "AnyLogic libraries required to build the model".

Name	Version	Location

At the bottom of the interface, the status bar shows "Time units: days" and "X=...18".

Click on Variable “nInfectious”

The screenshot shows the AnyLogic Professional interface for a simulation titled "SIR Agent Based Calibration : SimpleExperiment". The main workspace displays a list of variables on the left and a central workspace with a variable "nInfectious" highlighted by a yellow tooltip. The tooltip shows the variable name "nInfer" and its value "4,143".

Variables listed on the left:

- TotalPopulation: 10,000
- AverageIllnessDuration: 15
- ContactRate: 1
- InfectionProbability: 0.8

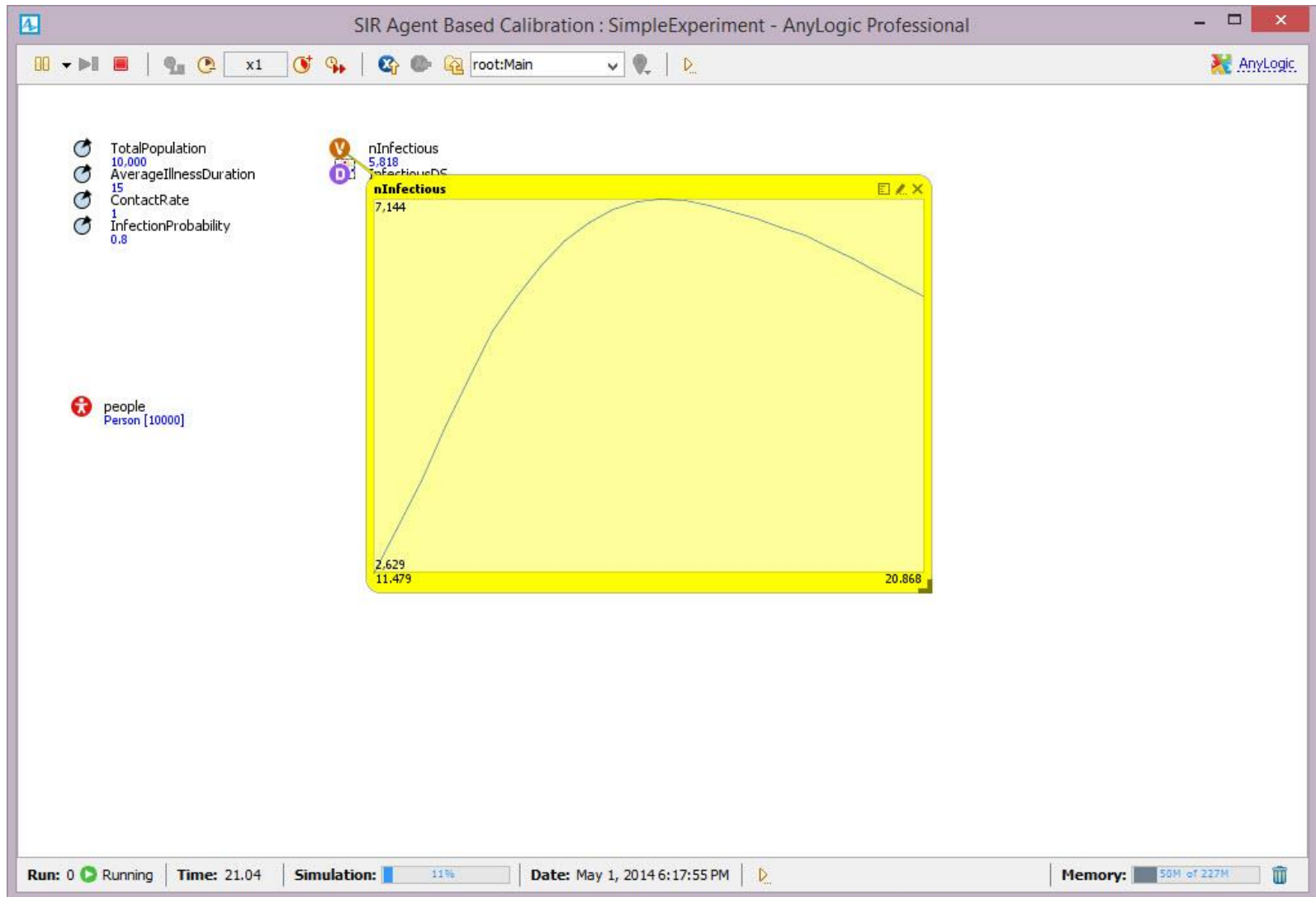
Agent-based model element:

- people: Person [10000]

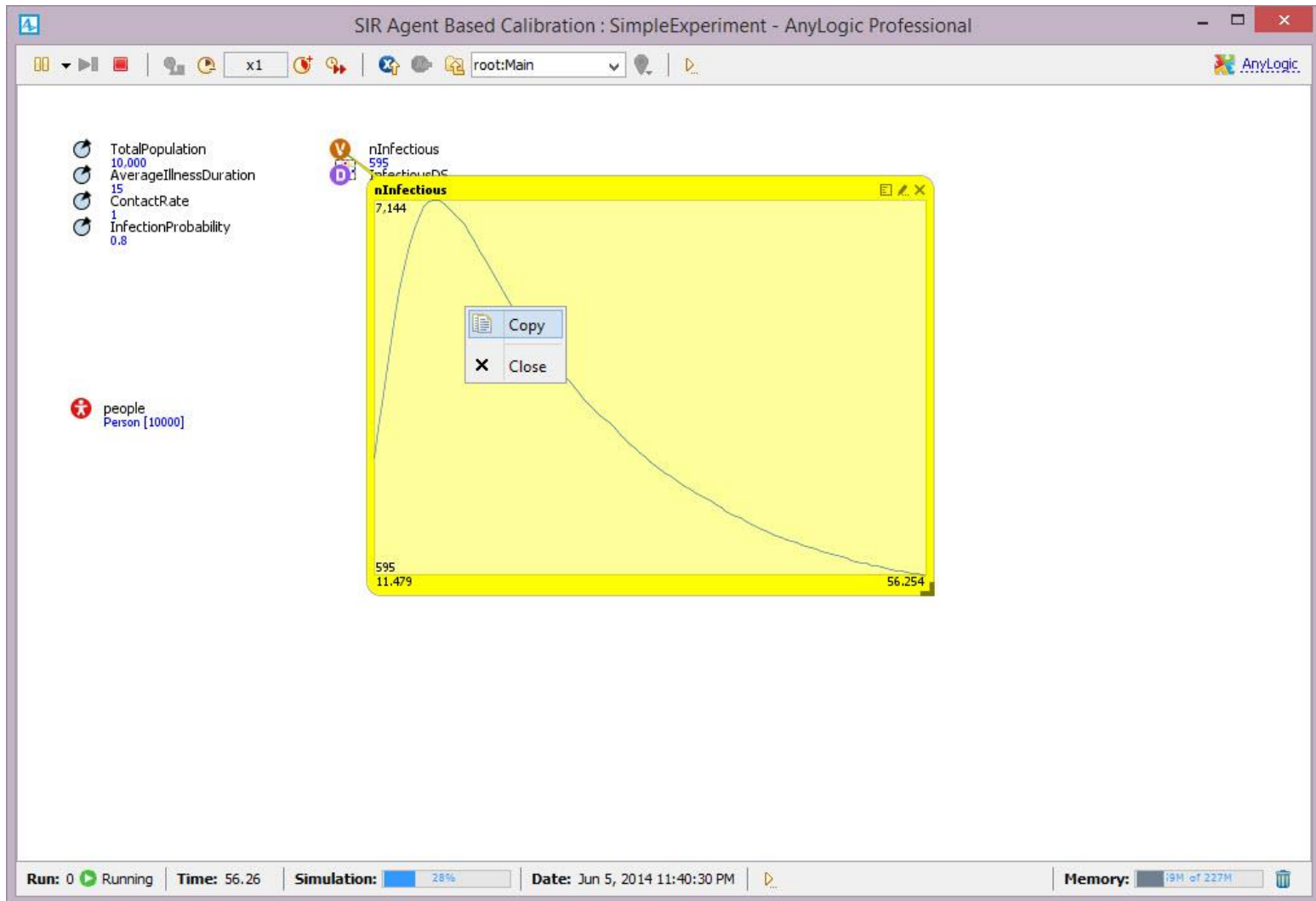
Simulation status bar:

- Run: 0
- Running: [Green circle]
- Time: 12.63
- Simulation: [Progress bar at 6%]
- Date: Apr 23, 2014 8:38:36 AM
- Memory: 39M of 227M

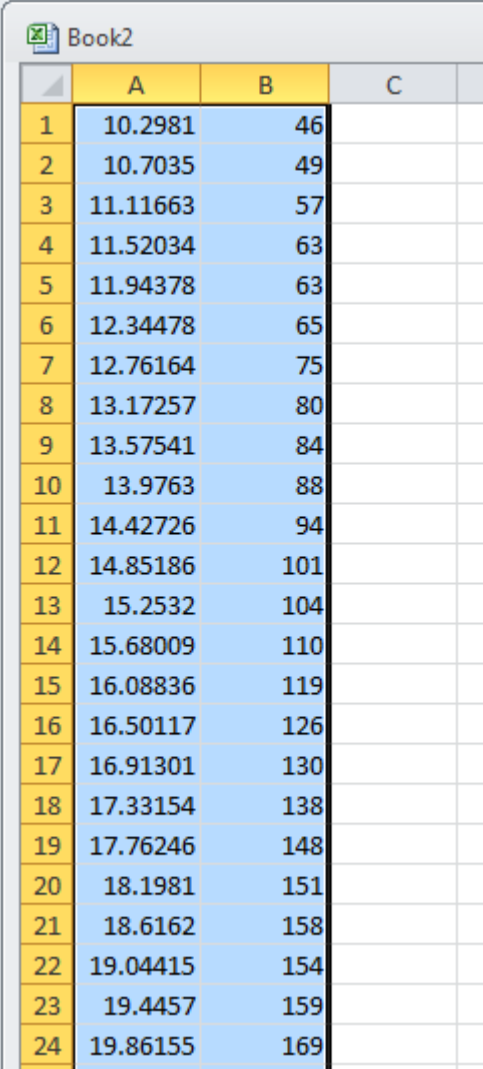
Graph of Variable



Right-Click to Copy the Numeric Data



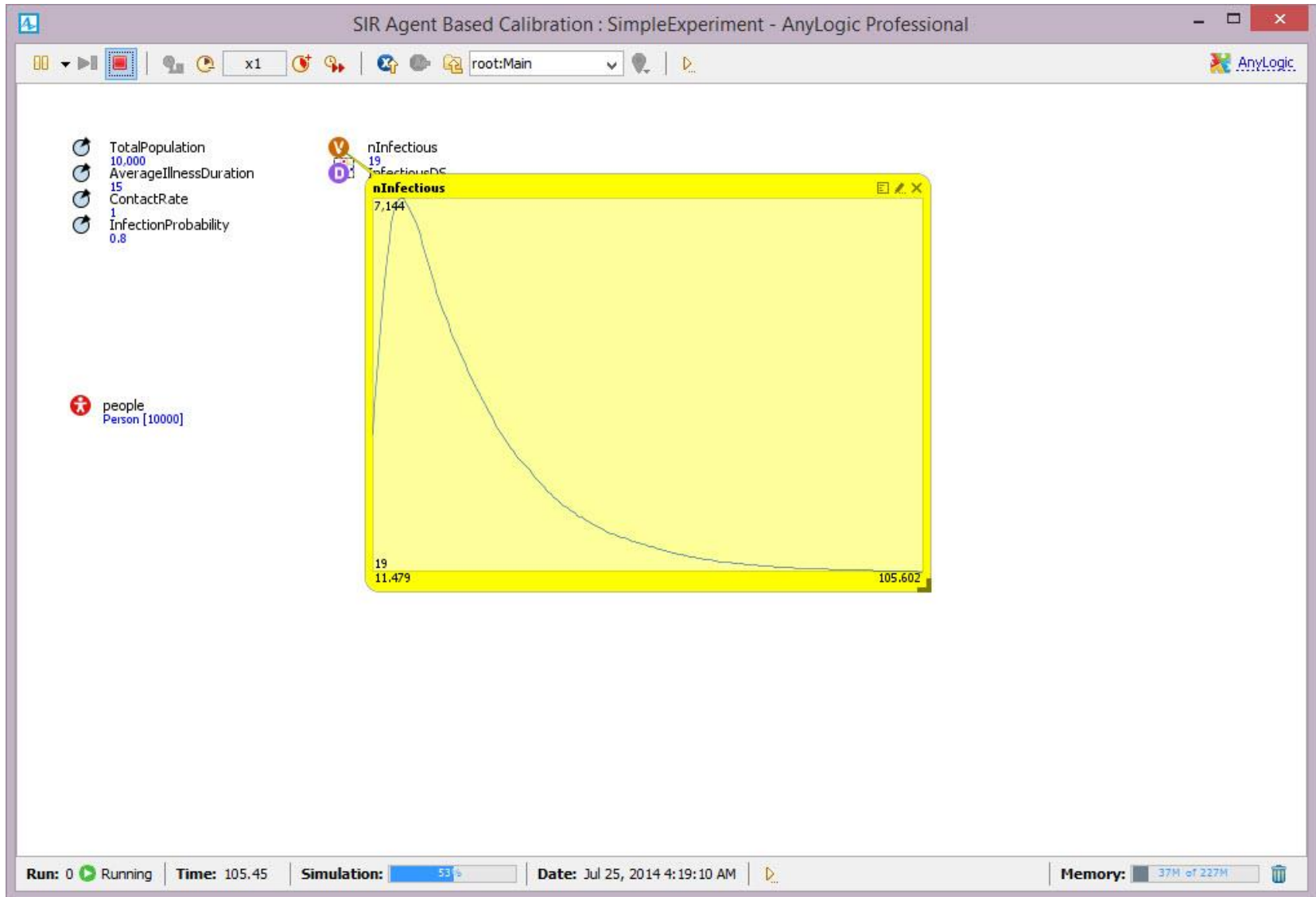
Pasting Into Excel



Book2

	A	B	C
1	10.2981	46	
2	10.7035	49	
3	11.11663	57	
4	11.52034	63	
5	11.94378	63	
6	12.34478	65	
7	12.76164	75	
8	13.17257	80	
9	13.57541	84	
10	13.9763	88	
11	14.42726	94	
12	14.85186	101	
13	15.2532	104	
14	15.68009	110	
15	16.08836	119	
16	16.50117	126	
17	16.91301	130	
18	17.33154	138	
19	17.76246	148	
20	18.1981	151	
21	18.6162	158	
22	19.04415	154	
23	19.4457	159	
24	19.86155	169	

Press Red "Stop" Button to Terminate Execution



Techniques for Collecting & Outputting Data

- Ad-Hoc Exports from variables
- Pre-Prepared methods
 - Statistics
 - Charts
 - Manual copies from visible datasets
 - Export to files
 - Writing to console
 - Export to databases
 - [AnyLogic Professional] Dataset archiving
 - Capturing images of graphs

Statistics & Charts

- A population of agents can have associated statistics that calculate values
- Examples of things that can be computed with using AnyLogic's statistics
 - Count of agents in the population for which certain condition (“predicate”) evaluates to true
 - Function of the values of some expression over the population
 - Maximum value
 - Minimum value
 - Average value
 - Sum (total) over population
 - Statistics can be defined as properties of the population

Select “People”, and Choose “Statistics”

The screenshot displays the AnyLogic Professional software interface. The main workspace shows a grid with several variables: TotalPopulation, AveragellnessDuration, ContactRate, InfectionProbability, nInfectious, and InfectiousDS. A blue arrow points from the 'people' agent in the bottom left towards the 'nInfectious' variable. The 'Properties' panel at the bottom is open to the 'people - Person' configuration, showing Y and Z coordinates set to 0. The 'Statistics' section is expanded, showing an 'Add statistics' button. The 'Palette' on the right lists various controls like Button, Check Box, Edit Box, Radio Buttons, Slider, Combo Box, List Box, File Chooser, and Progress Bar. The bottom status bar indicates 'Time units: days'.

Also, Expand “Statechart” Under “Person”

The screenshot displays the AnyLogic Professional software interface. The main workspace shows a statechart diagram with a state named "people [...]" at the bottom left. A blue arrow points from this state to a list of variables: TotalPopulation, AveragellnessDuration, ContactRate, InfectionProbability, nInfectious, and InfectiousDS. The "people [...]" state is currently selected, and its properties are shown in the bottom panel. The properties panel includes a "Y:" field with the value "0", a "Z:" field with the value "0", and a "Statistics" section with an "Add statistics" button. The "Advanced" and "Description" sections are also visible but collapsed. The top menu bar includes File, Edit, View, Draw, Model, Tools, and Help. The top toolbar contains various icons for file operations, navigation, and simulation. The right side of the interface features a "Palette" window with a "Controls" section containing various UI elements like Button, Check Box, Edit Box, Radio Buttons, Slider, Combo Box, List Box, File Chooser, and Progress Bar. The bottom status bar shows "Time units: days".

AnyLogic Professional

File Edit View Draw Model Tools Help

Projects Search

SimpleExperiment Person Main

My SIR Agent Based Calibration Extension

- Main
- Person
- SimpleExperiment: Main
- Calibration: Main
- MonteCarlo2DHistogram: Main

TotalPopulation nInfectious

AveragellnessDuration InfectiousDS

ContactRate

InfectionProbability

people [...]

Properties Progress

people - Person

Y: 0

Z: 0

Statistics

Add statistics

Advanced

Description

Time units: days

Click “Add Statistics”

The screenshot displays the AnyLogic Professional software interface. The main workspace shows a simulation model with a grid background. A blue arrow points from the top right towards the 'people' agent, which is located at the bottom left of the workspace. The 'people' agent is represented by a red circle with a white plus sign and the text 'people [...]'. To the right of the 'people' agent, there are several variables: 'TotalPopulation', 'AverageIllnessDuration', 'ContactRate', 'InfectionProbability', 'nInfectious', and 'InfectiousDS'. The 'nInfectious' and 'InfectiousDS' variables are highlighted with a yellow and purple icon respectively. The 'Properties' panel at the bottom shows the 'people - Person' agent selected. Under the 'Statistics' section, there is a green plus icon and the text 'Add statistics'. The 'Advanced' and 'Description' sections are also visible but collapsed. The 'Palette' panel on the right side of the interface shows various controls like Button, Check Box, Edit Box, Radio Buttons, Slider, Combo Box, List Box, File Chooser, and Progress Bar.

Fill in the “Condition” (Predicate) on Person

The screenshot displays the AnyLogic Professional software interface. The main workspace shows a statechart for a **Person** agent. The statechart includes variables: **TotalPopulation**, **AverageIllnessDuration**, **ContactRate**, **InfectionProbability**, **nInfectious**, and **InfectiousDS**. A state named **people [...]** is visible. A blue arrow points from the **people [...]** state to the **person** agent icon.

The **Properties** panel on the left shows the **person - Person** agent. The **Statistics** section is expanded, showing a configuration for a statistic named **peopleStat**. The **Condition** field is set to **item.st**.

A **statechart : Statechart - Person** window is open, listing the following methods:

- stop() : void - Agent
- stopSimulation() : boolean - Agent
- step(double height, double stepTime) : double - Utilities

The **Palette** on the right shows various controls, including **Button**, **Check Box**, **Edit Box**, **Radio Buttons**, **Slider**, **Combo Box**, **List Box**, **File Chooser**, and **Progress Bar**.

The bottom status bar indicates **Time units: days**.

Continue Typing

The screenshot displays the AnyLogic Professional software interface. The main workspace shows a statechart diagram with a state named 'people' and several variables: TotalPopulation, AveragellnessDuration, ContactRate, InfectionProbability, nInfectious, and InfectiousDS. A blue arrow points from the 'people' state to the 'Main' state.

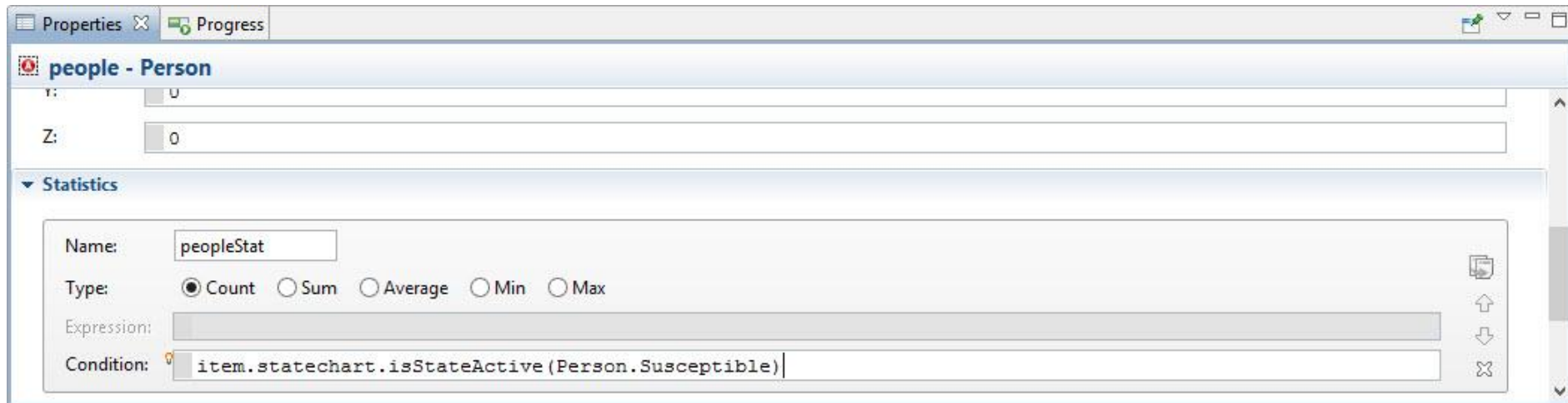
The Properties panel on the left shows the 'people - Person' object with the following statistics:

- Name: peopleStat
- Type: Count Sum Ave
- Expression:
- Condition: item.statechart.is

The Code Editor on the right shows the implementation of the `isStateActive` method:

```
isStateActive(short state) : boolean - Statechart  
  
public boolean isStateActive(short state)  
  
Returns true if the statechart is at the specified  
state, i.e. exactly in the state for a simple state and in  
one of its inner states for a composite state.  
  
Parameters:  
state - the state  
  
Returns:  
true if state is currently active
```


Full Expression



Expression: `item.statechart.isStateActive(Person.Susceptible)`

Example Statistics

The population in which the statistics are to be calculated

The screenshot displays the AnyLogic Advanced software interface. On the left, a project tree shows a statechart named 'TBProgressionStatechart' with various states and transitions. The main workspace shows a statechart diagram with a 'person' state highlighted in a red oval, indicating the population for statistics. Below the workspace, the 'Properties' panel for the 'person - Person' state is open, showing the configuration for a 'Count' statistic named 'CountSusceptible'. The expression for this statistic is 'item.TBProgressionStatechart.isStateActive(Person.TBSusceptible);'. A blue oval highlights the 'CountSusceptible' name and the expression field, with a blue arrow pointing to it from the text 'What statistics we wish to calculate'. A red arrow points from the text 'The population in which the statistics are to be calculated' to the 'person' state in the diagram.

AnyLogic Advanced [EDUCATIONAL USE ONLY]

Project

- Plain Variables
- Dynamic Variables
 - Age
 - Aging
 - Weight
- Statecharts
 - TBProgressionStatechart
 - TBProgressionStatechart
 - TBSusceptible
 - TBInfectiousContact
 - WhetherInfected
 - TBTransmission
 - WhetherPrimaryProgression
 - PrimaryProgression
 - UnDiagnosedActiveTB
 - NaturalTBRecovery
 - LTBI
 - Reactivation
 - DeathFromUndiagnosedTB
 - Death
 - UndiagnosedTBInfectionContact
 - Diagnosis
 - DiagnosedActiveTB
 - TreatmentMediatedTBRecovery

Person Main Person Main Main Main Main Main

DaysFromDiagnosisUntilRecovery

LikelihoodOfPrimaryProgression

environment

person [..]

Model

- Parameter
- Flow Aux ...
- Stock Vari...
- Event
- Dynamic ...
- Plain Vari...
- Collectio...
- Function
- Table Fun...
- Port
- Connector
- Entry Point
- State
- Transition
- Initial Stat...
- Branch
- History St...
- Final State
- Environ...

Console Properties

person - Person

General

Parameters

Statistics

Description

Name: CountSusceptible

Type: Count Sum Average Min Max

Expression:

Condition: `item.TBProgressionStatechart.isStateActive(Person.TBSusceptible);`

Add Statistics

Problems

Description	Location

Action

Analysis

Presentati...

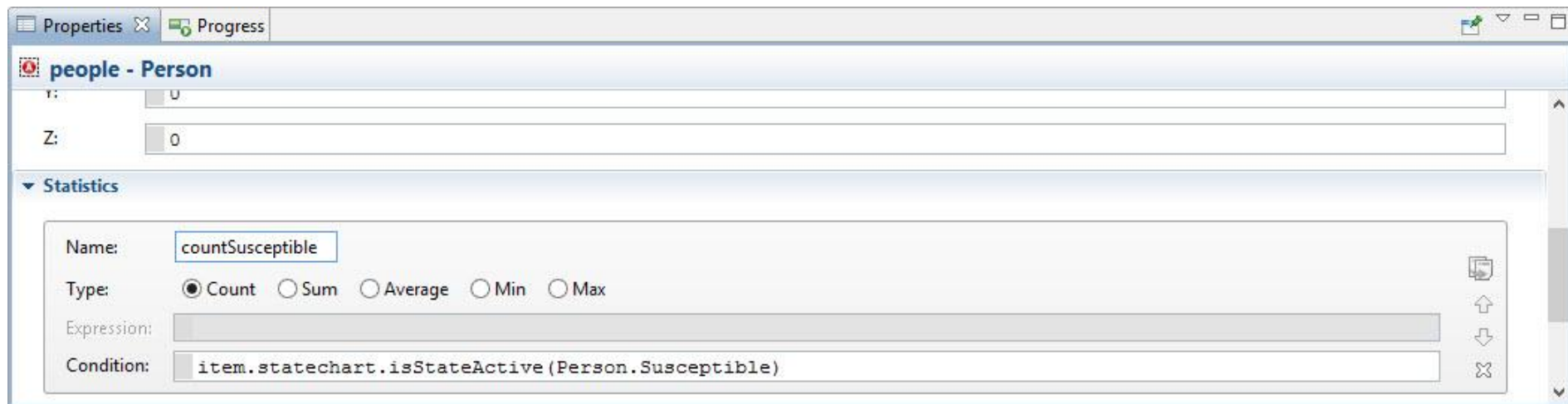
Connectiv...

Enterpris...

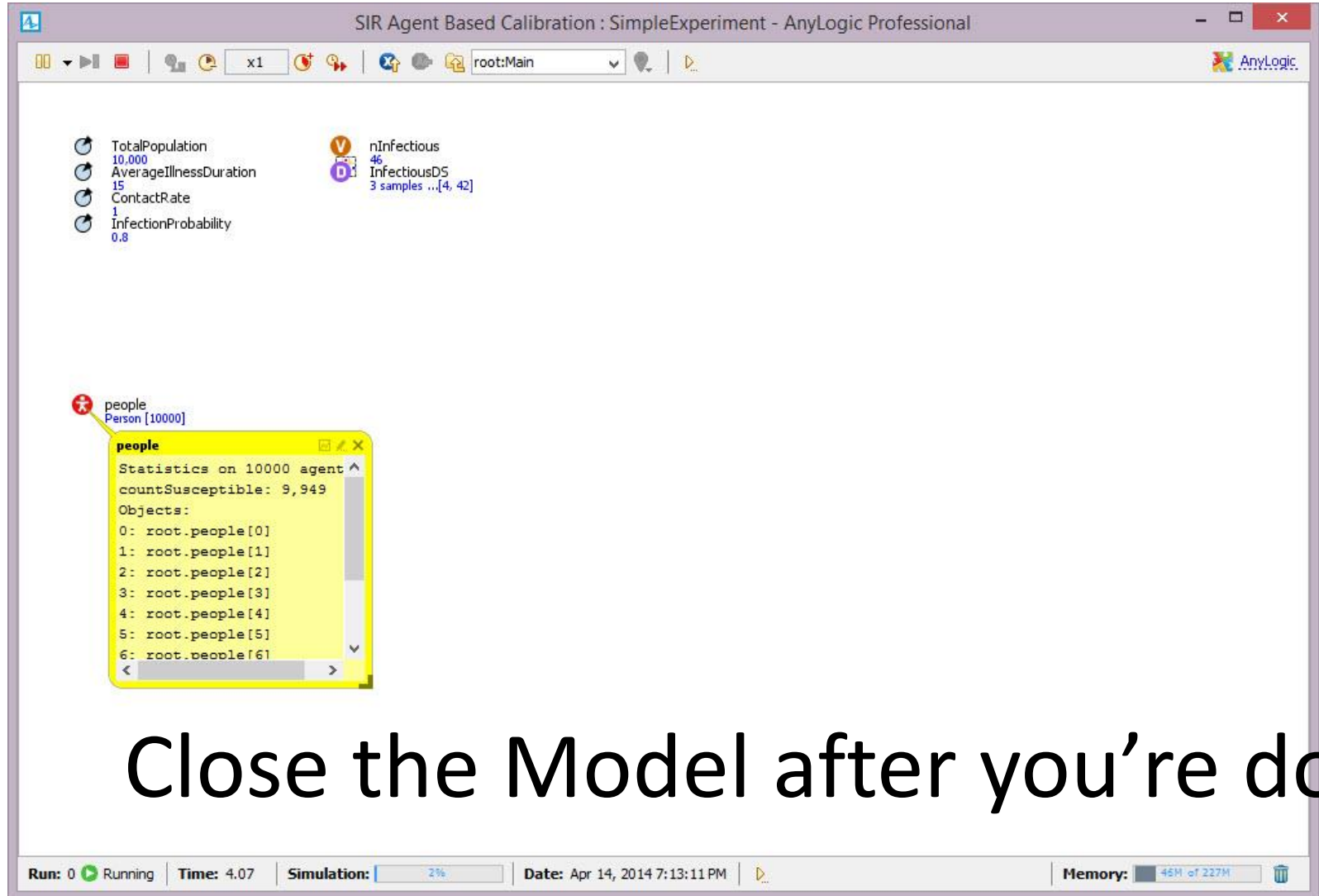
More Libraries...

What statistics we wish to calculate

Name the Statistic “countSusceptible”



Run the Model, and Click on “people” The Statistic should be Visible



The screenshot shows the AnyLogic Professional interface for a simulation titled "SIR Agent Based Calibration: SimpleExperiment". The main workspace displays several statistics:

- TotalPopulation: 10,000
- AverageIllnessDuration: 15
- ContactRate: 1
- InfectionProbability: 0.8
- nInfectious: 46
- InfectiousDS: 3 samples ..,[4, 42]

A yellow callout window is open over the "people" agent set, showing the following statistics:

```
people
Person [10000]
people
Statistics on 10000 agent ^
countSusceptible: 9,949
Objects:
0: root.people[0]
1: root.people[1]
2: root.people[2]
3: root.people[3]
4: root.people[4]
5: root.people[5]
6: root.people[6]
```

The bottom status bar indicates the simulation is running, with a time of 4:07 and a simulation progress of 2%. The date is Apr 14, 2014 7:13:11 PM, and the memory usage is 46M of 227M.

Close the Model after you're done

Drag a “Time Plot” from the Palette to the “Main” Canvas

The screenshot displays the AnyLogic Professional software interface. The main canvas, titled "Main", shows a grid with several variables: TotalPopulation, AverageIllnessDuration, ContactRate, InfectionProbability, nInfectious, and InfectiousDS. A "people [...]" agent is also visible. A blue rectangular box is drawn on the canvas, indicating the placement of a new element. The "Palette" window on the right side of the interface is open, showing a list of elements under the "Charts" category. The "Time Plot" element is highlighted in blue, indicating it is the selected element to be dragged onto the canvas.

AnyLogic Professional

File Edit View Draw Model Tools Help

Projects Search

SimpleExperiment Person Main

My SIR Agent Based Calibration Extension

- Main
- Person
- SimpleExperiment: Main
- Calibration: Main
- MonteCarlo2DHistogram: Main

Analysis

- Data
 - Data Set
 - Statistics
 - Histogram Data
 - Histogram2D Data
- Charts
 - Bar Chart
 - Stack Chart
 - Pie Chart
 - Plot
 - Time Plot**
 - Time Stack Chart
 - Time Color Chart

Properties Progress

My SIR Agent Based Calibration Extension - Model

Name: My SIR Agent Ba:

Model time units: days

System of measurement: Metric

Dependencies

AnyLogic libraries required to build the model:

Name	Version	Location

Time units: days

Enlarge the Chart

The screenshot displays the AnyLogic Professional interface for a simulation model. The main workspace shows a Time Plot chart with the following data series:

- TotalPopulation
- AverageIllnessDuration
- ContactRate
- InfectionProbability
- nInfectious
- InfectiousDS
- people [...]

The Time Plot properties panel is open, showing the following settings:

- Name: plot
- Ignore:
- Visible on upper level:
- Data: Add data item
- Data update: (empty)

The chart area shows a grid with the x-axis ranging from 0 to 100 and the y-axis ranging from -1 to 1. The data series are plotted as lines with markers. The 'people [...]' series is a horizontal line at y=0. The 'nInfectious' series is a horizontal line at y=1. The 'InfectiousDS' series is a horizontal line at y=0. The 'TotalPopulation' series is a horizontal line at y=1. The 'AverageIllnessDuration' series is a horizontal line at y=0. The 'ContactRate' series is a horizontal line at y=0. The 'InfectionProbability' series is a horizontal line at y=0.

Click “Add Data Item”

The screenshot displays the AnyLogic Professional software interface. The main workspace shows a simulation model with a grid background. A blue line represents the simulation flow, starting from a 'people' agent and moving through various components. A 'Time Plot' is visible, showing a step function with values 1 and 0 over time. The plot is titled 'plot - Time Plot' and has a name field set to 'plot'. The 'Data' section of the plot properties is expanded, showing an 'Add data item' button. The 'Data update' section is also visible. The 'Palette' on the right side of the interface lists various analysis and charting options, including 'Data Set', 'Statistics', 'Histogram Data', 'Histogram 2D Data', 'Bar Chart', 'Stack Chart', 'Pie Chart', 'Plot', 'Time Plot', 'Time Stack Chart', 'Time Color Chart', 'Histogram', and 'Histogram 2D'. The 'Properties' and 'Progress' panels are also visible at the bottom of the workspace.

AnyLogic Professional

File Edit View Draw Model Tools Help

Projects Search

SimpleExperiment Person Main

My SIR Agent Based Calibration Extension*

- Main
- Person
- SimpleExperiment: Main
- Calibration: Main
- MonteCarlo2DHistogram: Main

TotalPopulation
AveragellnessDuration
ContactRate
InfectionProbability

people [..]

nInfectious
InfectiousDS

1
0
-1

0 20 40 60 80 100

Properties Progress

plot - Time Plot

Name: plot Ignore Visible on upper level

Data

Add data item

Data update

Time units: days X=...91

Put in "people." and Press Ctrl-Space

The screenshot displays the AnyLogic Professional interface. The main workspace shows a model diagram with variables: TotalPopulation, AverageIllnessDuration, ContactRate, InfectionProbability, nInfectious, and InfectiousDS. A time plot is visible, showing a green line representing the number of people over time. The plot has a y-axis ranging from 0.5 to 0.9 and an x-axis representing time. The plot is currently set to display the value of 'people.'. A context menu is open over the plot, listing various methods for AgentList and AgentArrayList. The 'Value' radio button is selected in the plot properties, and the 'Value' field contains 'people.'. The 'Point style' is set to a solid circle, and the 'Color' is set to 'oliveDrab'. The 'Time units' are set to 'days'.

AnyLogic Professional

File Edit View Draw Model Tools Help

Projects Search

SimpleExperiment Person Main

My SIR Agent Based Calibration Extension*

- Main
- Person
- SimpleExperiment: Main
- Calibration: Main
- MonteCarlo2DHistogram: Main

TotalPopulation

AverageIllnessDuration

ContactRate

InfectionProbability

nInfectious

InfectiousDS

people [1]

Properties Progress

plot - Time Plot

Name: plot

Data

Value Data set

Value: people.

Point style: Line width: 1

Color: oliveDrab

average(...) : double - AgentList

contains(Object agent) : boolean - AgentArrayList

count(String triggerFieldName) : int - AgentList

get(int index) : Person - AgentArrayList

getEnvironment() : Agent - AgentList

getOwner() : Agent - AgentList

isEmpty() : boolean - AgentArrayList

iterator() : Iterator<Person> - AgentArrayList

max(...) : double - AgentList

min(...) : double - AgentList

onChange() : void - AgentList

random() : Person - AgentArrayList

Please Enter or double-click to see options...

Analysis

- Data
 - Data Set
 - Statistics
 - Histogram Data
 - Histogram2D Data
- Charts
 - Bar Chart
 - Stack Chart
 - Pie Chart
 - Plot
 - Time Plot
 - Time Stack Chart

Time units: days

Choose “Count Susceptible”

The screenshot displays the AnyLogic Professional interface. The main workspace shows a time plot with a green line representing the 'Count Susceptible' variable. The plot is titled 'plot - Time Plot' and is configured with the following settings:

- Value:** `people.countSusceptible()`
- Point style:** Solid line with circular markers
- Line width:** 1
- Color:** oliveDrab

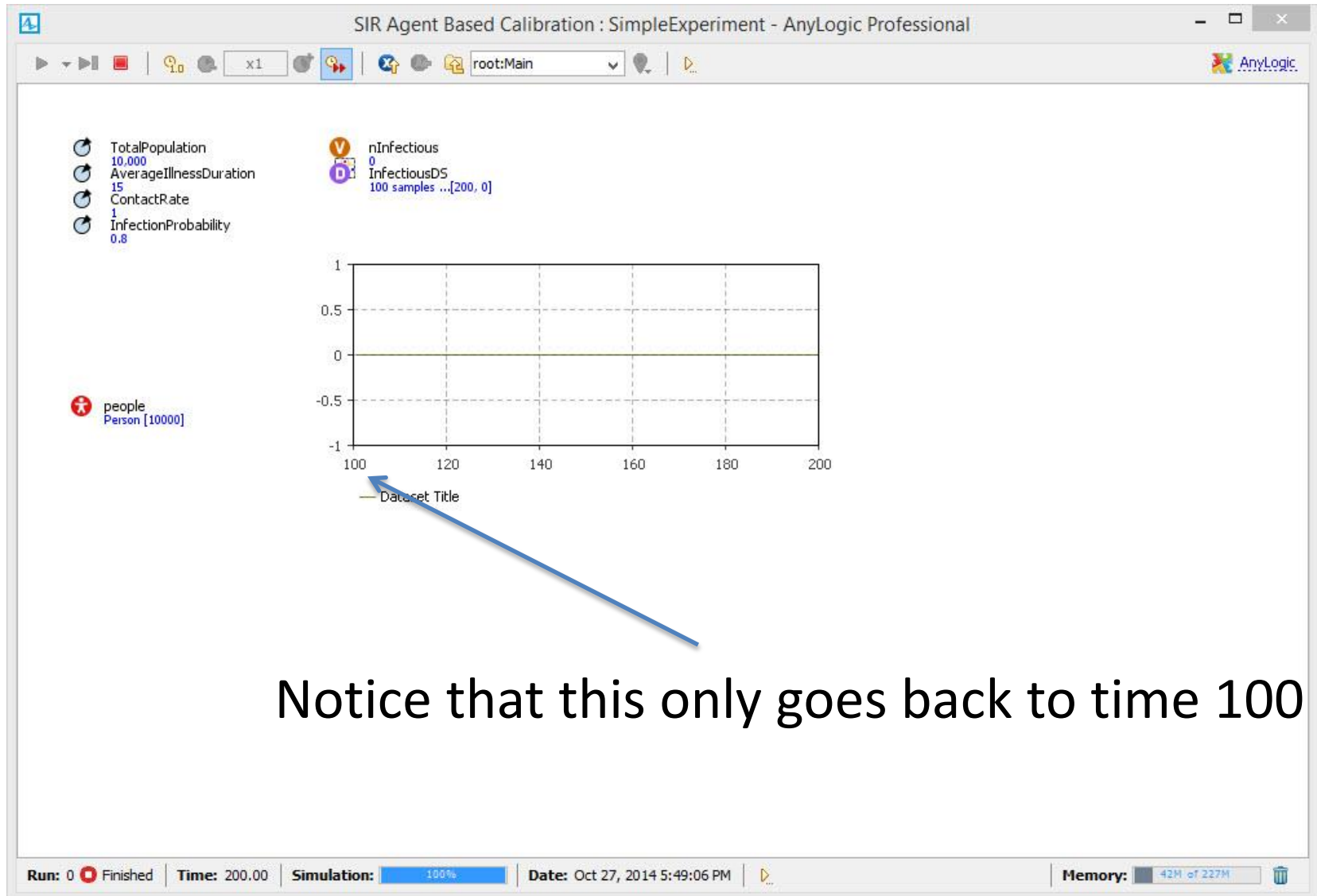
The plot shows a curve that starts at approximately 0.15, rises to a peak of about 0.7, and then declines to approximately 0.55. The y-axis ranges from 0 to 0.8, and the x-axis represents time in days.

The interface also shows a project tree on the left with the following structure:

- My SIR Agent Based Calibration Extension*
 - Main
 - Agents
 - Presentation
 - people_presentation
 - plot
 - Parameters
 - Variables
 - Analysis Data
 - Links to agents
 - Person
 - SimpleExperiment: Main
 - Calibration: Main
 - MonteCarlo2DHistogram: Main

The bottom status bar indicates 'Time units: days'.

Now Run the Model



Notice that this only goes back to time 100

Stop the Simulation, and Click on the Plot.

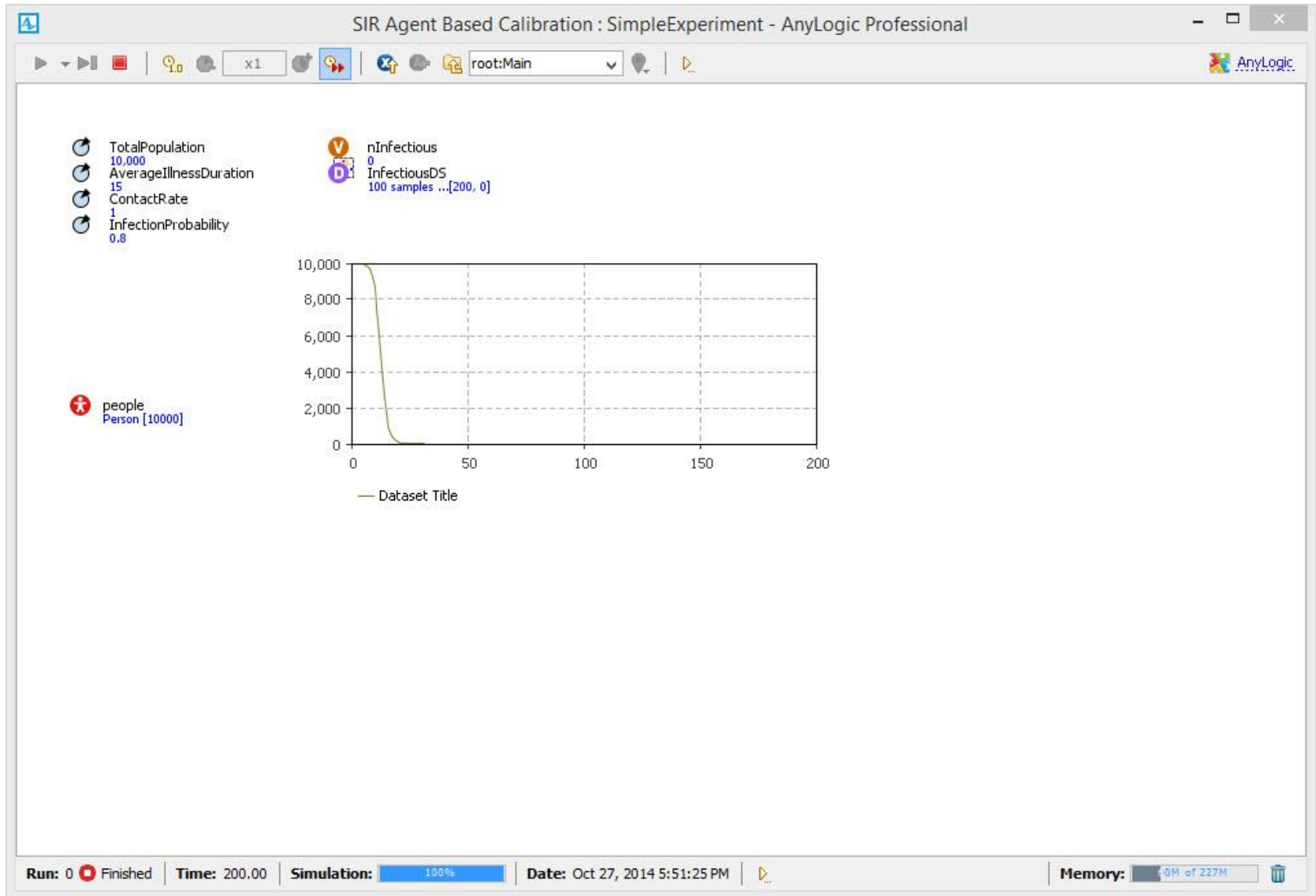
Change Time Window & Display Size to 200

The screenshot displays the AnyLogic Professional interface. The main window shows a simulation plot with a green line representing the 'Infectious' variable over time. The plot is titled 'plot - Time Plot' and is currently displaying data up to 200 samples. The configuration panel for the plot is open, showing the following settings:

- Data update:**
 - Update data automatically
 - Do not update data automatically
 - Recurrence time: 1
 - Display up to: 200 latest samples (applies to "Value" data items only)
- Scale:**
 - Time window: 200 model time units
 - Vertical scale: Auto Fixed

The status bar at the bottom indicates 'Time units: days'.

This Captures the Full Time Range



Techniques for Collecting & Outputting Data

- Ad-Hoc Exports from variables
- Pre-Prepared methods
 - Statistics
 - Charts
 - **Manual copies from visible datasets**
 - Export to files
 - Writing to console
 - Export to databases
 - [AnyLogic Professional] Dataset archiving
 - Capturing images of graphs

Datasets

- Datasets store recent values of some quantities from the model
- Datasets can be exported easily using custom code
 - This can simply call the dataset's toString method

Output: Datasets

The screenshot displays the AnyLogic Advanced [EDUCATIONAL USE ONLY] interface. The top toolbar includes standard file operations and a 'Get Support' link. The main workspace shows a diagram with an 'environment' object and a 'person' object, with a data set 'dsSusceptibleCount' associated with the person.

The left sidebar contains a project tree for 'TBv1*' with the following structure:

- Main
 - Parameters
 - DaysFromDiagnosisUntilRecovery: 30
 - DaysUntilDiagnosis: 60
 - DiagnosedPerDayTBContactRatePerNetworkContact: .
 - LikelihoodOfPrimaryProgression: .10
 - PerContactTBIInfectionProbability: .50
 - UndiagnosedPerDayTBContactRatePerNetworkContac
 - Functions
 - PersonWithMaxDegree
 - Environments
 - environment
 - Embedded Objects
 - person
 - Analysis Data
 - Presentation
 - person_presentation
 - TimePlotAgentCount
- Person
 - Parameters
 - DaysPerTimeUnit: 365.25
 - Ethnicity: 1
 - MeanDaysToNaturallyClearInfection: 180.00

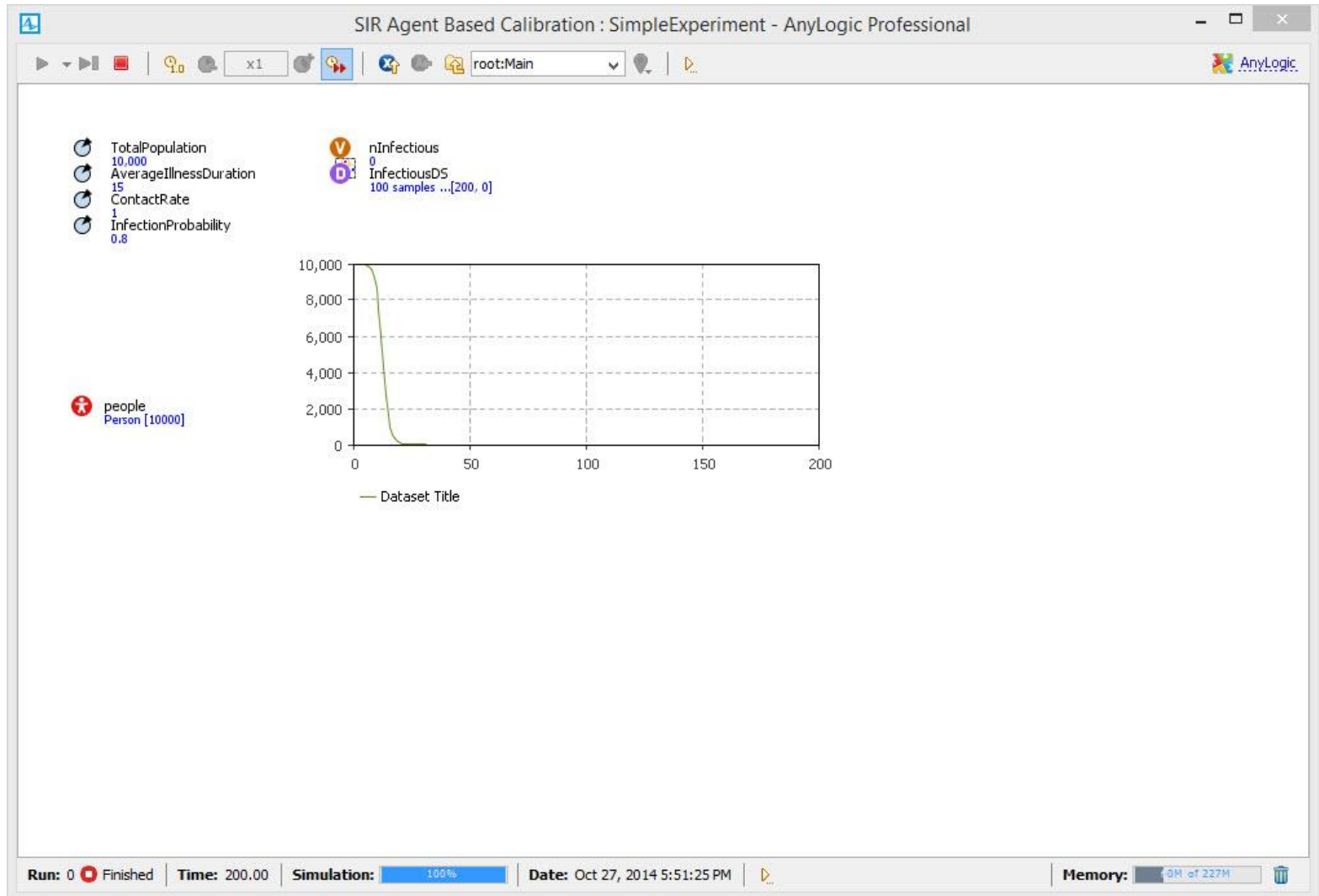
The bottom panel shows the 'dsSusceptibleCount - Data Set' configuration:

- General**
 - Name: dsSusceptibleCount
 - Show Name
 - Ignore
 - Public
 - Show At Runtime
- Description**
 - Use time as horizontal axis value
 - Horizontal axis value: []
 - Vertical axis value: person.CountSusceptible()
 - Keep up to 1000 latest samples
 - Do not update automatically
 - Update automatically
 - Begin at time: 0.0
 - Recurrence time: 1
 - October 29, 2009
 -

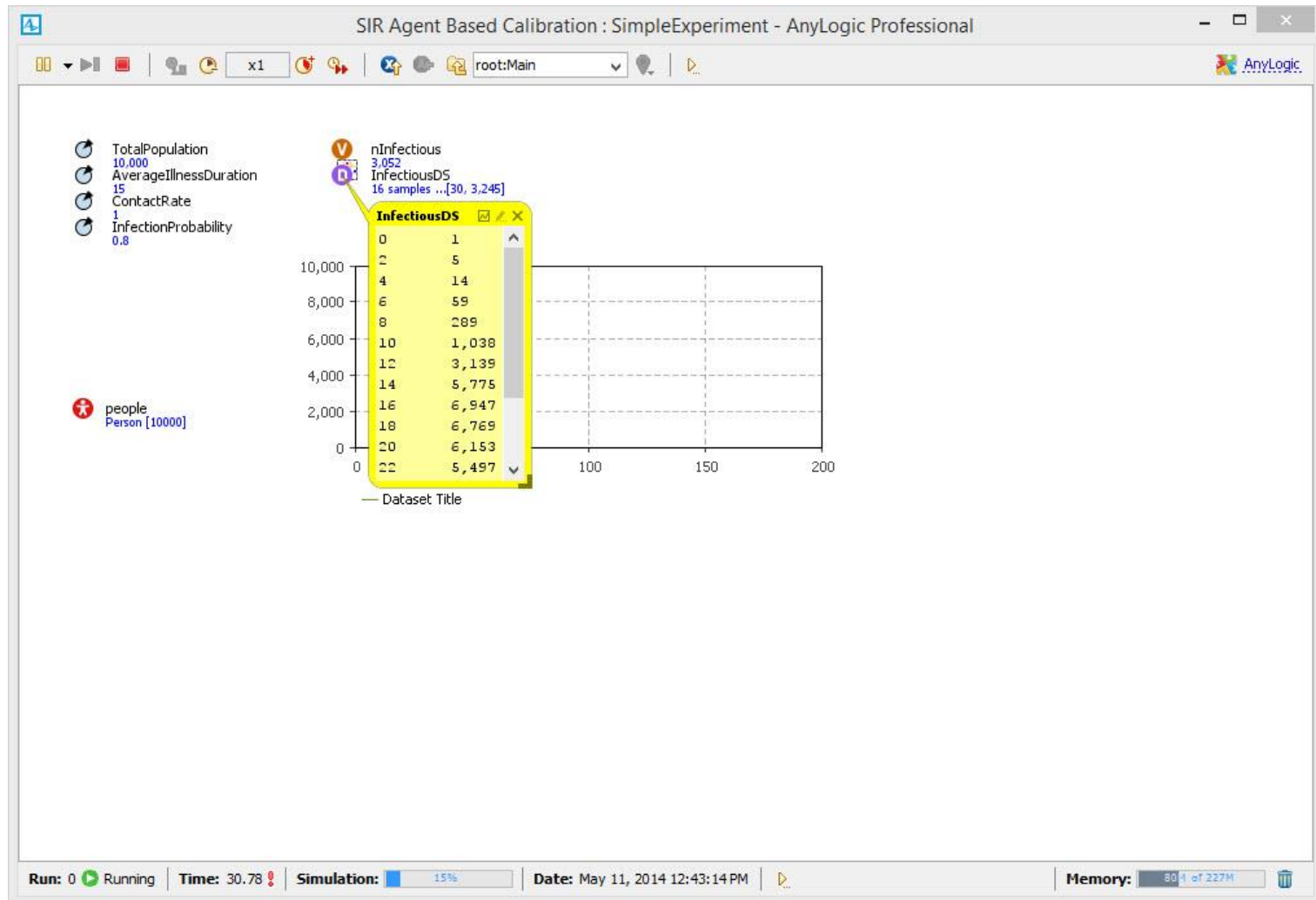
The bottom-left corner shows a 'Problems' table:

Description	Location

Run the Experiment & Click on “Infectious DS”



Click on “InfectiousDS” to See Data in Dataset



Right Click and Select “Copy”

The screenshot shows the AnyLogic Professional interface for a simulation titled "SIR Agent Based Calibration : SimpleExperiment". The main workspace displays a graph with a vertical axis labeled "people Person [10000]" ranging from 0 to 10,000 and a horizontal axis labeled "Dataset Title" ranging from 0 to 200. A data table titled "InfectiousDS" is overlaid on the graph, showing the following data:

Time	nInfectious
0	1
2	5
4	14
6	3,139
8	5,775
10	6,947
12	6,769
14	6,153
16	5,497
18	
20	
22	

A context menu is open over the data table, with the "Copy" option selected. The menu also includes a "Close" option. The simulation status bar at the bottom indicates: Run: 0, Running, Time: 67.90, Simulation: 34%, Date: Jun 17, 2014 3:36:04 PM, and Memory: 46M of 227M.

Call Up Excel and Paste into It

	A	B	C	D
1	0	1		
2	2	5		
3	4	14		
4	6	59		
5	8	289		
6	10	1,038		
7	12	3,139		
8	14	5,775		
9	16	6,947		
10	18	6,769		
11	20	6,153		
12	22	5,497		
13	24	4,845		
14	26	4,271		
15	28	3,761		
16	30	3,245		
17	32	2,834		
18	34	2,475		
19	36	2,154		

Dataset Properties

The screenshot displays the AnyLogic Advanced [EDUCATIONAL USE ONLY] interface. The main workspace shows a statechart with a state named 'datasetAbsolutePrevalence' and a line graph below it. The graph's vertical axis is labeled with the expression $((double) nInfectious) / ((double) TotalPopulation)$ and has a value of 0.8. The horizontal axis represents time.

The 'Properties' panel for the selected dataset is open, showing the following configuration:

- Name:** datasetAbsolutePrevalence
- Show Name
- Ignore
- Public
- Show A

Description

- Use time as horizontal axis value
- Horizontal axis value: []
- Vertical axis value: $((double) nInfectious) / ((double) TotalPopulation)$

Keep up to: 5000 latest samples

- Do not update automatically
- Update automatically

Begin at time: 0.0 **Recurrence time:** 1

Additional options: June 11, 2008, 2:54:05 AM

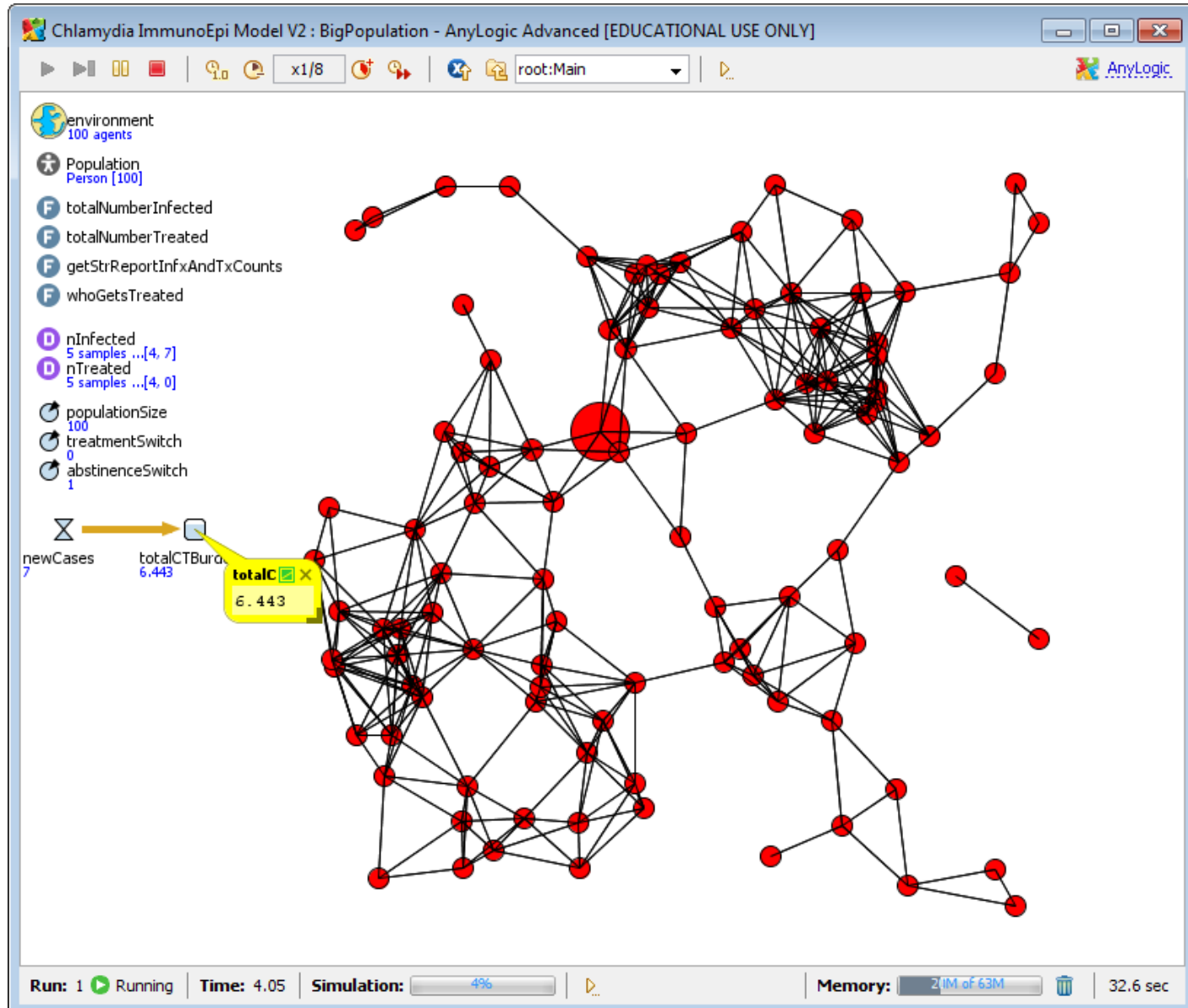
The left sidebar shows a project tree for 'AnqiModelV1' with folders for Main, Parameters, Plain Variables, Functions, Environments, Embedded Objects, Analysis Data, Presentation, and Person. The 'Person' folder is expanded, showing 'Plain Variables' (color), 'Statecharts' (statechart), and 'statechart'.

The bottom-left 'Problems' panel lists several errors:

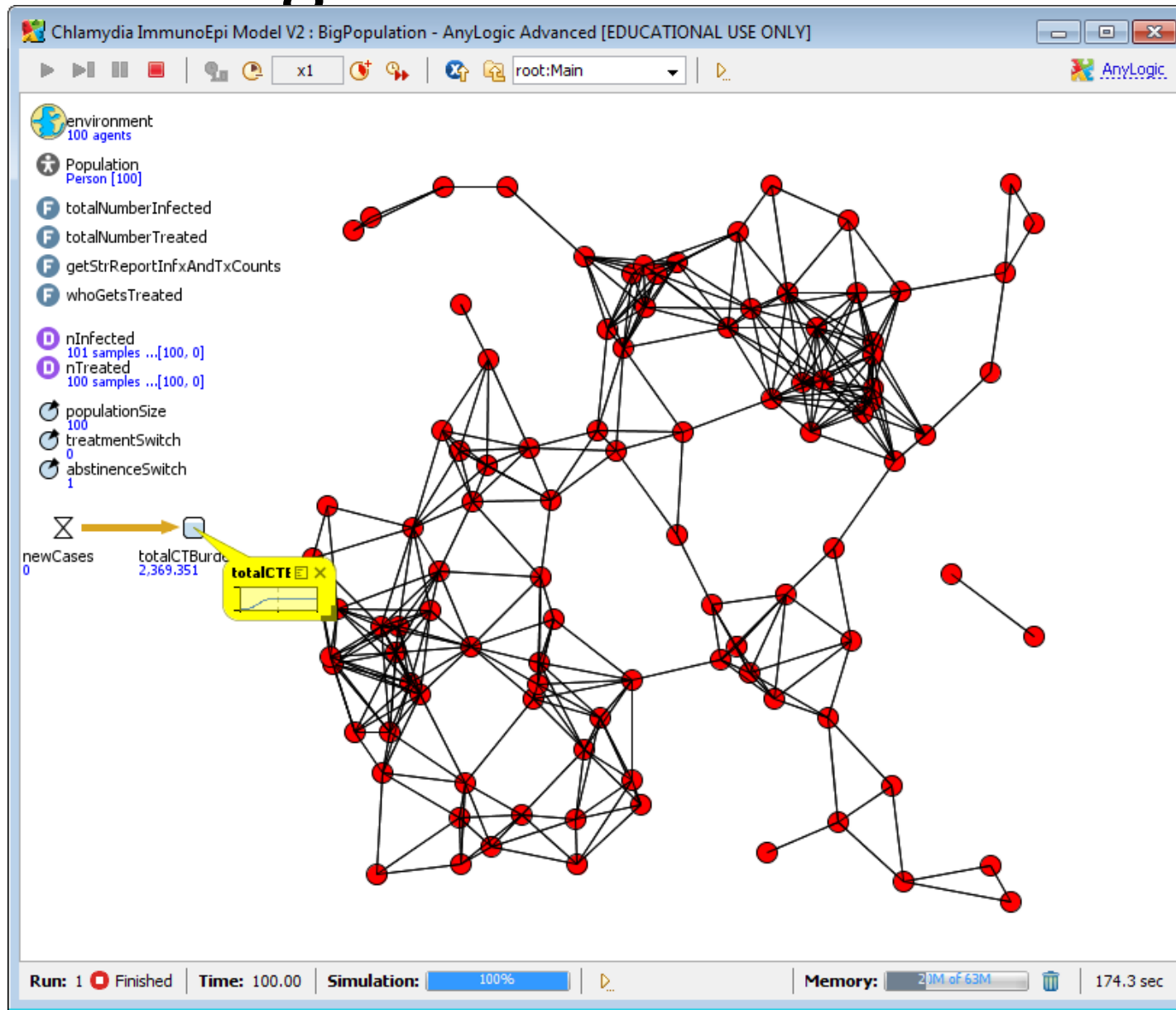
- The constructor DataSet() is undefined
- The constructor DataSet() is undefined
- The constructor DataSet() is undefined
- The constructor DataSet() is undefined
- Engine.log cannot be resolved
- mViewer cannot be resolved
- mViewer cannot be resolved

The right sidebar contains a 'Model' palette with various components like Parameter, Flow Aux, Stock Vari, Event, Dynamic, Plain Vari, Collectio, Function, Table Fun, Port, Connector, Entry Point, State, Transition, Initial Stat, Branch, History St, Final State, and Environm.

Ad-hoc Export



Begins as a Small Chart



Copying Data

The screenshot displays the AnyLogic Advanced interface for the 'Chlamydia ImmunoEpi Model V2 : BigPopulation'. The main workspace shows a network diagram with red nodes and black edges. On the left, a tree view lists model components: environment (100 agents), Population (Person [100]), and various attributes like totalNumberInfected, totalNumberTreated, and nInfected. A yellow arrow points from the 'totalCTBurden' variable in the tree to a graph window. The graph window, titled 'totalCTBurden', shows a line plot of the variable's value over time (0 to 90). The y-axis ranges from 0 to 2,500. The curve starts at 0, rises to a peak of approximately 2,369.351 around time 40, and then levels off. A context menu with 'Copy' and 'Close' options is visible over the graph.

Chlamydia ImmunoEpi Model V2 : BigPopulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]

environment
100 agents

Population
Person [100]

totalNumberInfected

totalNumberTreated

getStrReportInfxAndTxCounts

whoGetsTreated

nInfected
82 samples ...[81, 0]

nTreated
82 samples ...[81, 0]

populationSize
100

treatmentSwitch
0

abstinenceSwitch
1

newCases
0

totalCTBurden
2,369.351

totalCTBurden

2,500

2,000

1,500

1,000

500

0

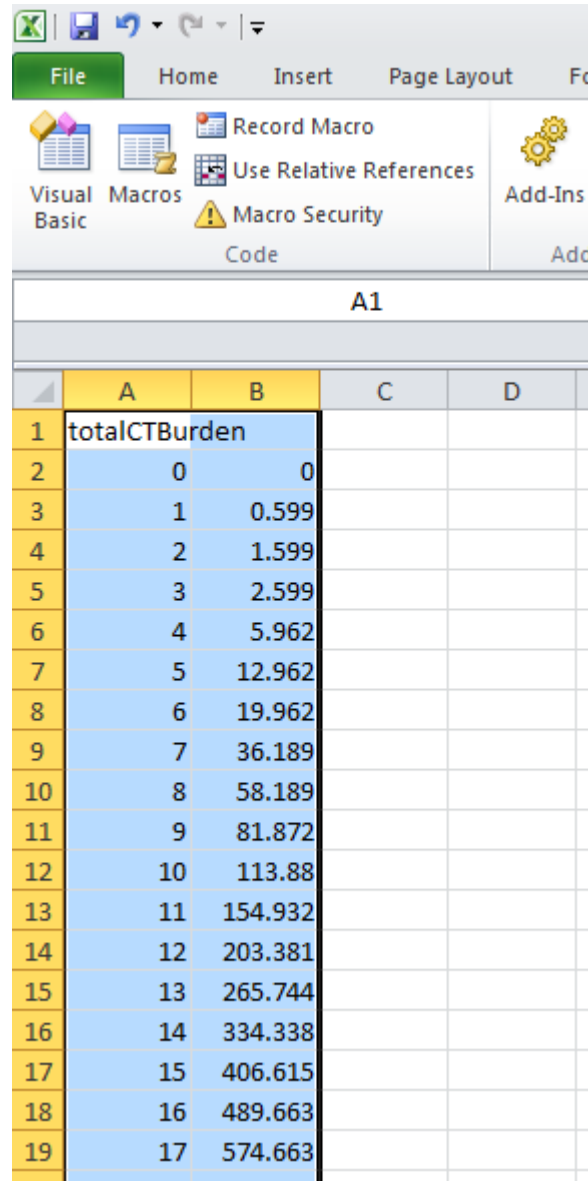
0 10 20 30 40 50 60 70 80 90

Copy

Close

Run: 1 Running Time: 81.45 Simulation: 81% Memory: 19M of 63M 154.6 sec

Data Exported from Ad-Hoc Chart

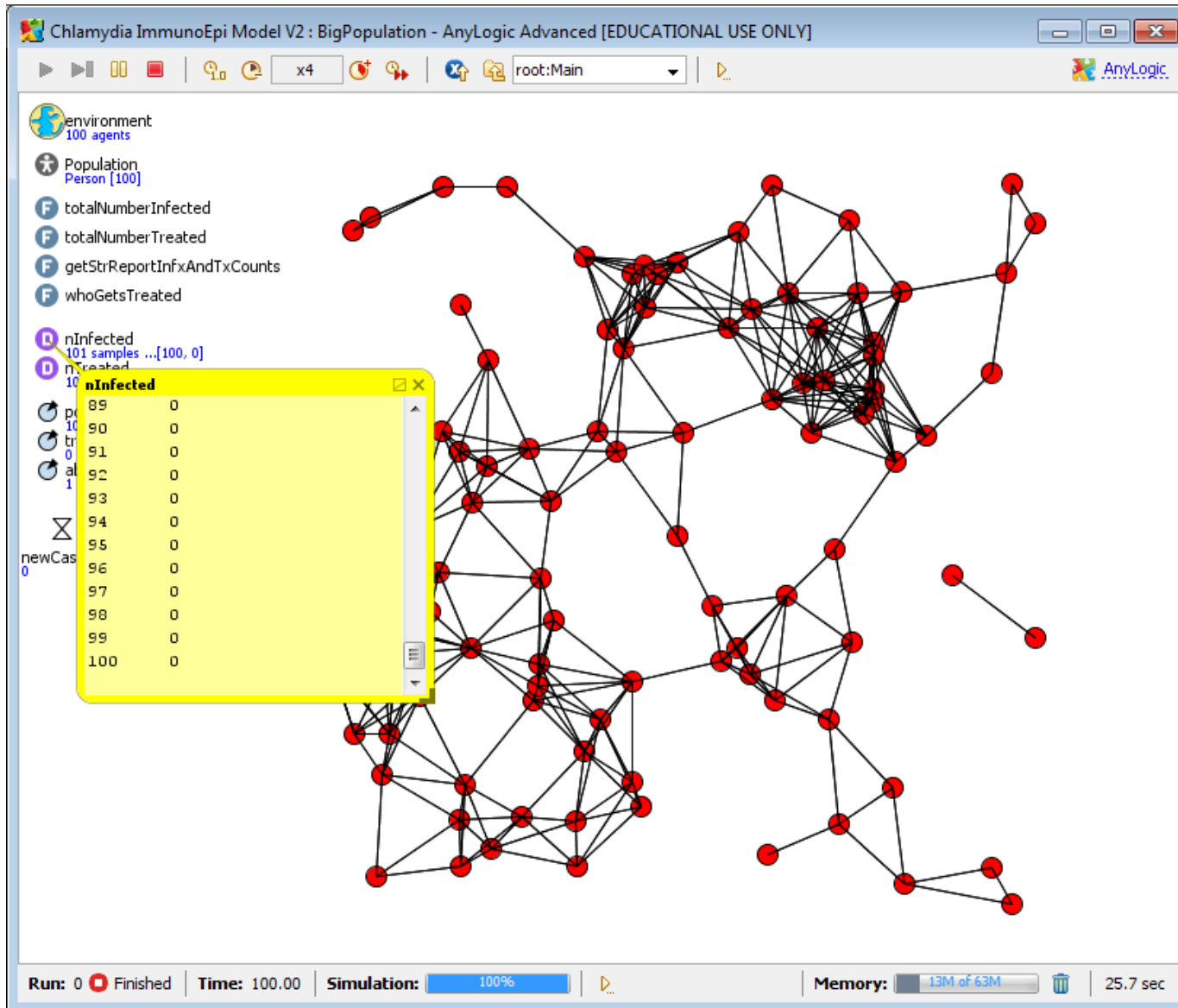


	A	B	C	D
1	totalCTBurden			
2		0		
3		1	0.599	
4		2	1.599	
5		3	2.599	
6		4	5.962	
7		5	12.962	
8		6	19.962	
9		7	36.189	
10		8	58.189	
11		9	81.872	
12		10	113.88	
13		11	154.932	
14		12	203.381	
15		13	265.744	
16		14	334.338	
17		15	406.615	
18		16	489.663	
19		17	574.663	

Techniques for Outputting Data

- Ad-Hoc Exports from variables
- Manual copies from visible datasets
- Capturing images of graphs
- Export to files
- Writing to console
- [AnyLogic Professional] Dataset archiving
- Export to databases

Manual Output from Datasets



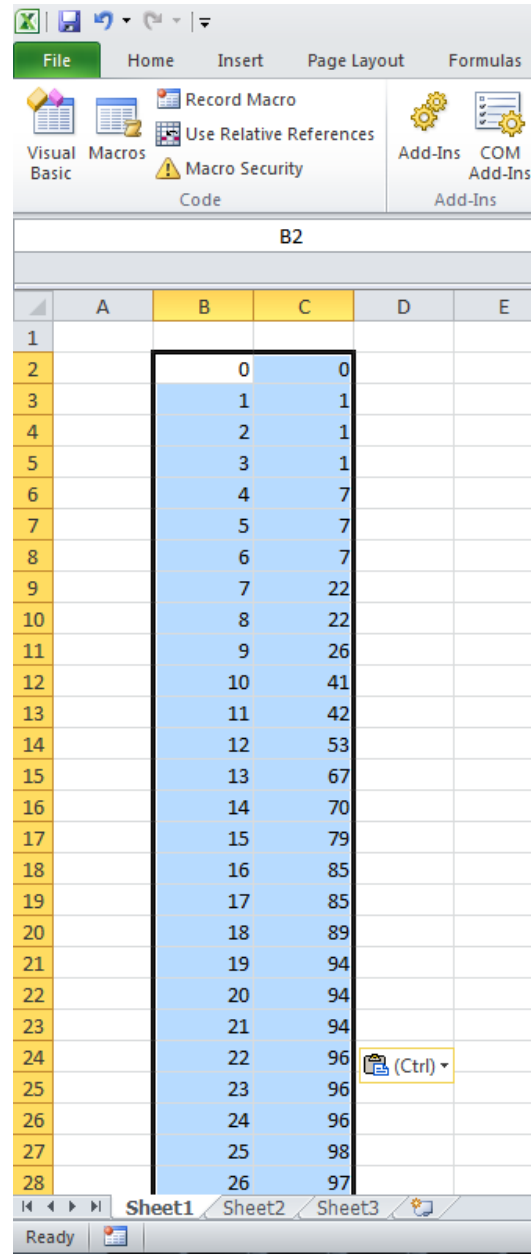
Right Clicking Gives Context Menu

The screenshot displays the AnyLogic Advanced interface for a simulation titled "Chlamydia ImmunoEpi Model V2 : BigPopulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]". The main workspace shows a complex network graph with red circular nodes and black connecting lines. On the left, a tree view lists the model's components and variables, including "environment" (100 agents), "Population" (Person [100]), and various statistics like "totalNumberInfected" and "nInfected". A yellow highlight is placed over the "nInfected" variable in the tree view, which has triggered a context menu. This menu contains a table of data and two buttons: "Copy" and "Close".

nInfected	
89	0
90	0
91	0
92	0
93	0
94	0
95	0
96	0
97	0
98	0
99	0
100	0

Run: 0 Finished | Time: 100.00 | Simulation: 100% | Memory: 12M of 63M | 25.7 sec

Copied Data Can be Pasted into Excel



Declaratively Specifying Datasets

The image shows a software interface with a grid area at the top and a configuration panel at the bottom. The grid area contains a list of variables:

- environment
- Population [..]
- totalNumberInfected
- totalNumberTreated
- getStrReportInfxAndTxCounts
- whoGetsTreated
- nInfected** (highlighted with a purple circle)
- nTreated** (highlighted with a purple circle)
- populationSize

The configuration panel is titled "nInfected - Data Set" and has the following settings:

- Name:** nInfected
- Show Name
- Ignore
- Public
- Show At Runtime
- Use time as horizontal axis value
- Horizontal axis value: [Empty text box]
- Vertical axis value: totalNumberInfected ()
- Keep up to: 1000 latest samples
- Do not update automatically
- Update automatically
- Begin at time: 0.0
- Recurrence time: 1
- March 4, 2010 9:31:57 PM

Supported Dataset Types

- Simple
 - holds values only -- no timestamps
- Timed
 - holds values and timestamps
- Phase
 - holds pairs of values but no timestamps
- Histogram
 - can define bins for data set
 - data set will record # falling in each bin

Techniques for Outputting Data

- Ad-Hoc Exports from variables
- Manual copies from visible datasets
- Capturing images of graphs
- **Output to console**
- Export to files
- [AnyLogic Professional] Dataset archiving
- Export to databases

Output to Console

- Pros

- Easy to program

- `ActiveObject.traceIn(String str)` **outputs string to console**
- `System.out.println(String str)` (Black)
- `System.err.println(String str)` (Red)

- Readily visible

- Copy & Paste to another document

- Cons

- May be mixed with other output (easy to miss other output)

- Limited length

- Depends on memory to copy

- Less structured

Techniques for Outputting Data

- Ad-Hoc Exports from variables
- Manual copies from visible datasets
- Capturing images of graphs
- Writing to console
- **Export to files**
- [AnyLogic Professional] Dataset archiving
- Export to databases

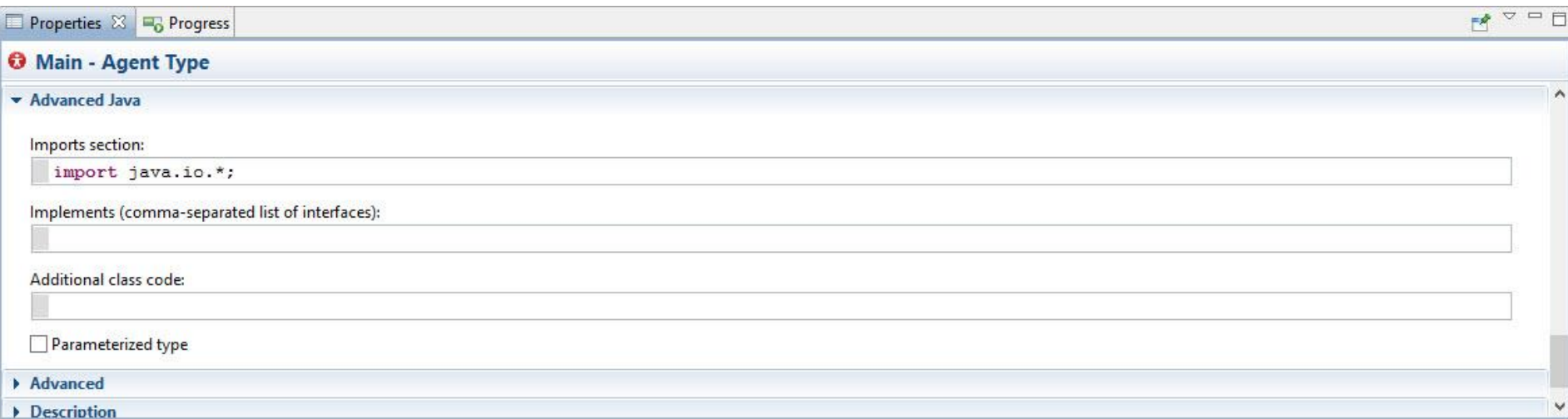
Data Output to File

- Pros
 - Simple to perform
 - Relatively easy to import into e.g. Excel, R, etc.
 - Files can be readily archived
- Cons
 - Awkward to have multiple data sets output to the same file
 - Awkward to draw combine from multiple files
 - Denormalization: Requires either
 - Duplication of scenario-wide information (e.g. parameter values) on each row
 - Separate header section & later section

Outputting a Dataset to a File Requires 2 Steps

- “Importing” (specifying how to find) the necessary Java code
- Defining the code

Step 1: Importing the Necessary Java Libraries



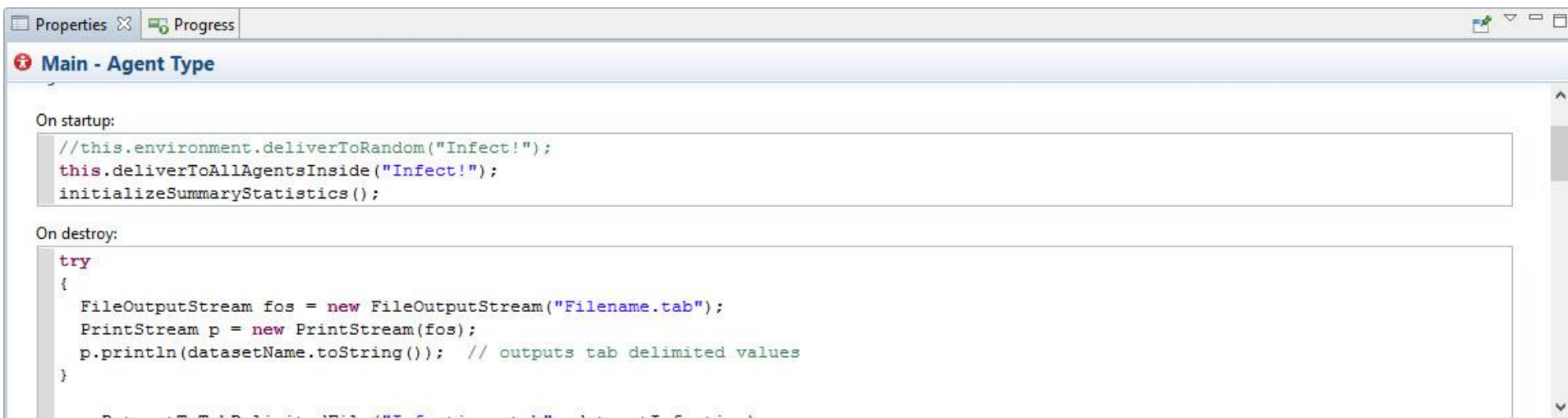
Step 2: Code to Export Dataset to File

```
try          Substitute whatever file name you wish to use
{           You may wish to put a "path" in front of this
    FileOutputStream fos = new
    FileOutputStream("Filename.tab");
    PrintStream p = new PrintStream(fos);
    p.println(datasetName.toString()); // outputs
    tab delimited values
}
catch (Exception e)
{           Substitute the name of the dataset
           You wish to output
    println("Could not write to file.");
}
```

Suggestion: For greater versatility, place this in a function that takes the file name as a parameter.

Where to Put the Code to Output the Dataset

Option 1: In “Destroy Code” for Main



The screenshot shows a NetLogo IDE window titled "Main - Agent Type". The window has tabs for "Properties" and "Progress". The main area contains two sections of code:

```
On startup:  
//this.environment.deliverToRandom("Infect!");  
this.deliverToAllAgentsInside("Infect!");  
initializeSummaryStatistics();
```

```
On destroy:  
try  
{  
  FileOutputStream fos = new FileOutputStream("Filename.tab");  
  PrintStream p = new PrintStream(fos);  
  p.println(datasetName.toString()); // outputs tab delimited values  
}
```

Where to Put the Code to Output the Dataset

Option 2: In “Action” for an Event Triggered at times

The screenshot displays the AnyLogic Professional software interface. The main workspace is divided into four panes: **System Parameters**, **Statistics Dimension**, **Main Events**, and **Model Entities**. The **Main Events** pane shows a list of events, including **writeToFile**. The **Properties** pane is open for the **writeToFile - Event**, showing the following configuration:

- Name: writeToFile
- Visible: yes
- Trigger type: Timeout
- Mode: Occurs once
- Use model time (selected) / Use calendar dates
- Occurrence time (absolute): 0
- Occurrence date: 2014-04-11 8:00:00 AM

The **Action** pane for the event contains the following code:

```
try
{
    FileOutputStream fos = new FileOutputStream("Filename.tab");
    PrintStream p = new PrintStream(fos);
    p.println(datasetName.toString()); // outputs tab delimited values
}
```

The bottom status bar indicates "Time units: days".

Techniques for Outputting Data

- Ad-Hoc Exports from variables
- Manual copies from visible datasets
- Capturing images of graphs
- Writing to console
- Export to files
- [AnyLogic Professional] Dataset archiving
- **Export to databases**

Output to Databases: Tradeoffs

- Pros
 - More flexible than string output to file
 - Can query from diverse tools (e.g. excel, R, SPSS, SAS, etc.)
 - Can easily clean up
 - For larger databases
 - Transactional (either writes entirely or not at all)
 - Can query from remote machines
- Cons
 - More programming
 - Need to set up a database

Output to Databases: Steps

- One Time:
 - Install database on computer
 - Setting up the database schema
 - Add reference to database libraries
- Each scenario during simulation
 - Open database connection at start of model
 - Optionally, “insert” model version & parameter information into the database
 - Periodically during simulation
 - “insert” values into databases
 - At end of model execution, close database connection

Relevant Databases

- Databases most oriented towards single users & single computers
 - MS Access
 - H2
 - These databases less robust => risk of corruption
 - These are often quite fast
- Databases oriented towards multiple users & multiple computers
 - Oracle
 - DB2
 - MS SQL Server
 - Open source
 - Postgres
 - Derby
 - MySQL
 - More robust
 - Support remote access

Database Dependencies (MySQL database)

Properties Progress

SKDiabeticESRDModel - Model

Dependencies

AnyLogic libraries required to build the model:

Name	Version	Location

Jar files and class folders required to build the model:

Location
mysql-connector-java-5.1.13-bin-1.jar

Options for Database Access

- AnyLogic Professional: Built-in visual database classes
 - Simplify the composition of database operations
- Direct calling of database operations in Java's "Java DataBase Connectivity" (JDBC) Library
 - Note ODBC "bridge" for windows database driver support
- Custom database classes
 - We would be happy to share our simple interface
 - More refined interfaces possible

Example Simple Database Class for SQL Relational Database Systems

A Method is associated with each of
Execute
Query
Insert

```
public class MyDB {
    private static String DriverName = "com.mysql.jdbc.Driver";
    private String dbURL = "jdbc:mysql://localhost:3306/mydb";
    private String dbuser = "root";
    private String dbpassword = "2005051146";
    //This is due to consideration of conflicts between database of AnyLogic and java.sql.* package.
    private java.sql.Connection conn = null;
    private java.sql.Statement stmt = null;
    private java.sql.ResultSet rs = null;
    /**
     * Default constructor
     */
    public MyDB(){
        try{
            Class.forName(DriverName);
        }catch(java.lang.ClassNotFoundException e){
            System.err.println(e.getMessage());
            System.out.println("Error with constructor!");
        }
    }
    /**
     *method name: executeQuery()
     *Query
     *return value: ResultSet
     */
    public java.sql.ResultSet executeQuery(String sql){
        try{
            conn = DriverManager.getConnection(dbURL,dbuser,dbpassword);
            stmt = conn.createStatement();
            rs=stmt.executeQuery(sql);
        }catch(SQLException ex){
            System.err.println(ex.getMessage());
            System.out.println("Error with executeQuery() method!");
        }
        return rs;
    }
    /**
     *method name: executeUpdate()
     *update, delete, and insert
     *return value: int
     */
    public int executeUpdate(String sql){
        int result=0;
        try{
            conn = DriverManager.getConnection(dbURL,dbuser,dbpassword);
            stmt=conn.createStatement();
            result=stmt.executeUpdate(sql);
        }catch(SQLException ex){
            result=0;
            System.err.println(ex.getMessage());
        }
        return result;
    }
    @Override
    public String toString() {
        return super.toString();
    }
}
```

Example: Execute Query

```
/**
 *method name: executeQuery()
 *Query
 *return value: ResultSet
 */
public java.sql.ResultSet executeQuery(String sql){
    try{
        conn = DriverManager.getConnection(dbURL,dbuser,dbpassword);
        stmt = conn.createStatement();
        rs=stmt.executeQuery(sql);
    }catch(SQLException ex){
        System.err.println(ex.getMessage());
        System.out.println("Error with executeQuery() method!");
    }
    return rs;
}
```


Setup for Database Class

Properties Progress

myConn - Variable

Name: Show name Ignore

Visible: yes

Type: Other

Initial value: `new MyDB ()`

▶ Advanced

▶ Description

Example Database Output Code

A database query language (SQL) statement

```
double simulated_time = time()+1975;
for(int k=0;k<Cube[0][0].length;k++){
    String sql = "INSERT INTO dataset (agegroup,ethnicity,state,modeltime,amount,simulation_id) VALUES (
        "+1+", "+0+", "+k+", "+simulated_time+", "+Cube[1][0][k]+", "+simulation_id+" ";
    int ret = myConn.executeUpdate(sql);
    if(ret == 0){
        println("Adding new record "+1+" "+0+" "+k+" in "+Cube[1][0][k]+" dataset TABLE failed!");
    }
}
```

Requesting that the database class execute the SQL statement

Checking to make sure that the insert worked properly

Database Output: Suggestions

- Maintain metadata
 - Purpose of run
 - Parameter settings
 - Model version (& possibly .alp file)
- Be mindful of performance & space burdens
 - Try to batch up data inserts
 - Be selective in what data to store, balancing pros & cons of storing more material
 - Pros: Analytic flexibility, greater understanding, less risk of having to re-run simulation
 - Cons: Mammoth database size, Impaired performance
 - Use a local database if possible

Database Input

- Database input can be desirable when “feeding in” certain data to model
 - Connection choreography
 - Agent movement patterns
 - Count of incident cases of a condition
 - Count of vaccinations over time
- Frequently this data is “quantized” into time units
 - In those cases, Dynamic Events can be helpful