

Collecting & Outputting Data from AnyLogic

Nathaniel Osgood

March 15, 2011

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 only supported by AnyLogic Professional*
- AnyLogic does 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

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

The screenshot displays the AnyLogic software interface. The main window shows a project named "SIR Agent Based Calibration" with a grid workspace containing several objects, including "nInfectious" and "InfectiousDS". A context menu is open over the workspace, showing the "New" option selected, which has opened a sub-menu. In this sub-menu, the "Experiment" option is highlighted. Other options in the sub-menu include "Model", "Active Object Class", "Dimension", "Java Class", "Java Interface", and "Library".

The interface includes a menu bar (File, Edit, View, Model, Window, Help), a toolbar, and a palette on the right side. The palette is divided into sections: "General" (Parameter, Event, Dynamic Event, Plain Variable, Collection, Function, Table Function, Port, Connector, Environment), "System Dynamics" (Statechart, Actionchart, Analysis, Presentation, 3D, Controls, Connectivity, Pictures, 3D Objects), "Enterprise Library", and "Pedestrian Library".

At the bottom, a properties panel for the "SIR Agent Based Calibration - Model" is visible, showing fields for Name, Package, and File.

Add an Experiment

AnyLogic University [EVALUATION USE ONLY]

File Edit View Model Window Help

Projects

- SIR Agent Based Calibration
 - Main
 - Person
 - Calibration: Main
 - MonteCarlo2DHistogram: Main
 - Analysis Data
 - Presentation

Person

envi

peop

Palette

- General
 - Parameter
 - Event
 - Dynamic Event
 - Plain Variable
 - Collection
 - Function
 - Table Function
 - Port
 - Connector
 - Environment
- System Dynamics
- Statechart
- Actionchart
- Analysis
- Presentation
- 3D
- Controls
- Connectivity
- Pictures
- 3D Objects
- Enterprise Library
- Pedestrian Library
- Palettes...

Problems

Description	Locati
-------------	--------

Properties Console

SIR Agent Based Calibration - Model

General Name: SIR Agent Based Calibration

Dependencies

Description Package: sir_agent_based_calibration

File: C:\Program Files (x86)\AnyLogic 6_5 University\plugins\com.xj.anylogic.examples_6.5.

New Experiment

Experiment

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

Name: SimpleExperiment

Main Active Object Class (root): Main

Experiment Type:

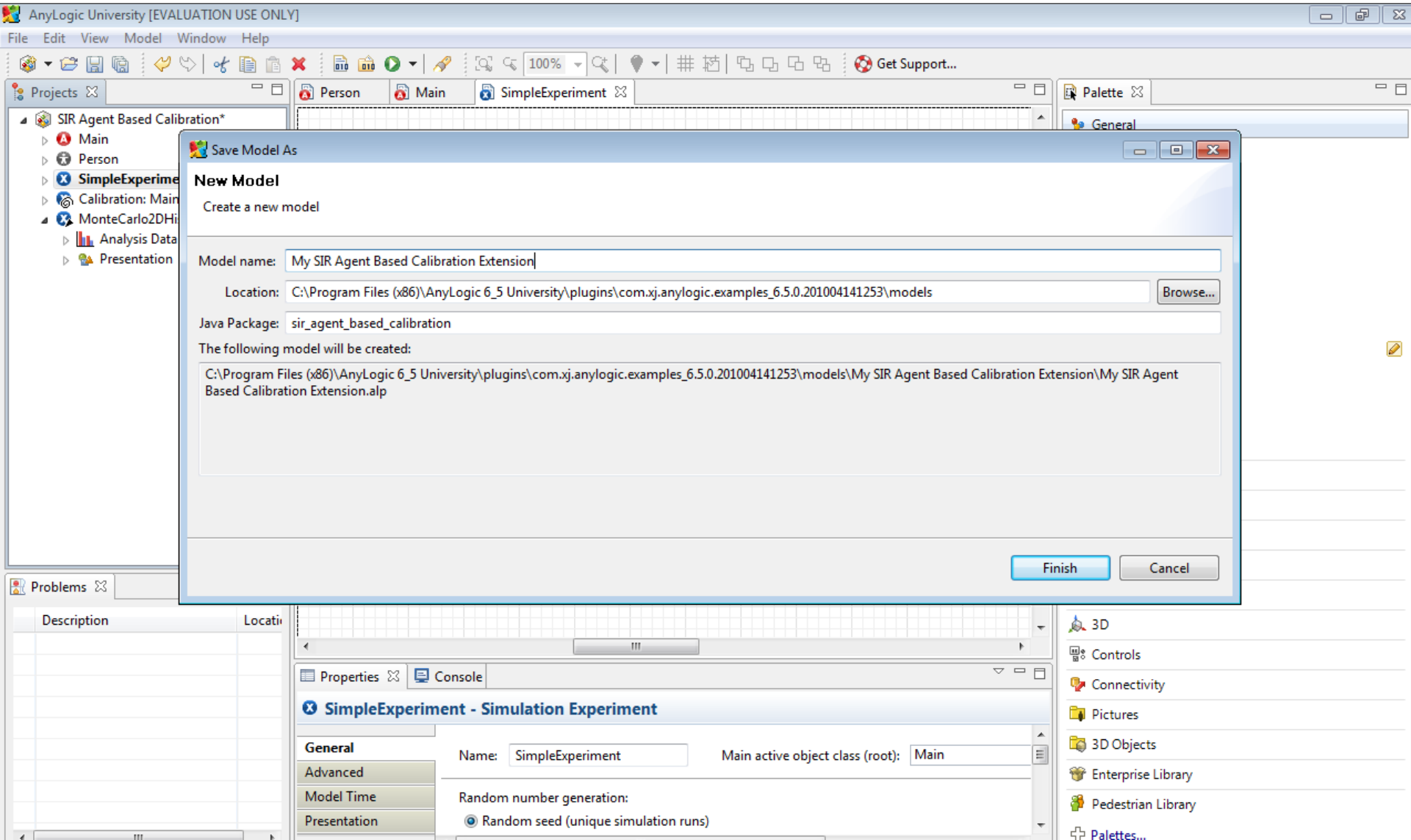
- Simulation
- Optimization
- Parameters Variation
- Compare Runs
- Monte Carlo
- Sensitivity Analysis
- Calibration
- Custom

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

Copy model time settings from: Calibration

< 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 University software interface, titled "AnyLogic University [EVALUATION USE ONLY]". The main workspace shows a simulation model with a grid background. A "Recent Experiment" pop-up menu is open, listing three experiments: "My SIR Agent Based Calibration Extension / SimpleExperiment", "My SIR Agent Based Calibration Extension / Calibration", and "My SIR Agent Based Calibration Extension / MonteCarlo2DHistogram". The "SimpleExperiment" is selected. The main workspace contains several objects: "AreaSide", "TotalPopulation", "AverageIllnessDuration", "ContactRate", "InfectionProbability", "nInfectious", "InfectiousDS", "environment", and "people [...]". The "Properties" panel at the bottom shows the "SimpleExperiment - Simulation Experiment" configuration. The "General" tab is active, displaying the "Name" as "SimpleExperiment" and the "Main active object class (root)" as "Main". The "Presentation" tab shows "Random number generation" set to "Random seed (unique simulation runs)". The "Palette" on the right lists various components: General (Parameter, Event, Dynamic Event, Plain Variable, Collection, Function, Table Function, Port, Connector, Environment), System Dynamics, Statechart, Actionchart, Analysis, Presentation, 3D, Controls, Connectivity, Pictures, 3D Objects, Enterprise Library, Pedestrian Library, and Palettes... The "Projects" panel on the left shows a tree view of the project structure, including "My SIR Agent Based Calibration Extensi", "Main", "Person", "SimpleExperiment: Main", "Calibration: Main", and "MonteCarlo2DHistogram: Main".

AnyLogic University [EVALUATION USE ONLY]

File Edit View Model Window Help

Projects

- My SIR Agent Based Calibration Extensi
 - Main
 - Person
 - SimpleExperiment: Main
 - Calibration: Main
 - MonteCarlo2DHistogram: Main
 - Analysis Data
 - Presentation

Person

Recent Experiment

- My SIR Agent Based Calibration Extension / SimpleExperiment
- My SIR Agent Based Calibration Extension / Calibration
- My SIR Agent Based Calibration Extension / MonteCarlo2DHistogram

AreaSide

TotalPopulation

AverageIllnessDuration

ContactRate

InfectionProbability

nInfectious

InfectiousDS

environment

people [...]

Properties Console

SimpleExperiment - Simulation Experiment

General

Name: SimpleExperiment Main active object class (root): Main

Advanced

Model Time

Presentation

Random number generation:

Random seed (unique simulation runs)

Palette

- General
 - Parameter
 - Event
 - Dynamic Event
 - Plain Variable
 - Collection
 - Function
 - Table Function
 - Port
 - Connector
 - Environment
- System Dynamics
- Statechart
- Actionchart
- Analysis
- Presentation
- 3D
- Controls
- Connectivity
- Pictures
- 3D Objects
- Enterprise Library
- Pedestrian Library
- Palettes...

Click on Variable “nInfectious”

The screenshot displays the AnyLogic software interface. The main workspace shows a simulation model with several variables and entities. The variable **nInfectious** is highlighted with a yellow tooltip, indicating its current value is **716**. The tooltip also shows a small 'x' icon and the text 'nInfectious [0, 715]'. The status bar at the bottom indicates the simulation is running, with a time of 49.97 and a simulation progress of 25%. The memory usage is shown as 11M of 63M.

Variables and their values:

- AreaSide: 100
- TotalPopulation: 10,000
- AverageIllnessDuration: 15
- ContactRate: 1
- InfectionProbability: 0.8

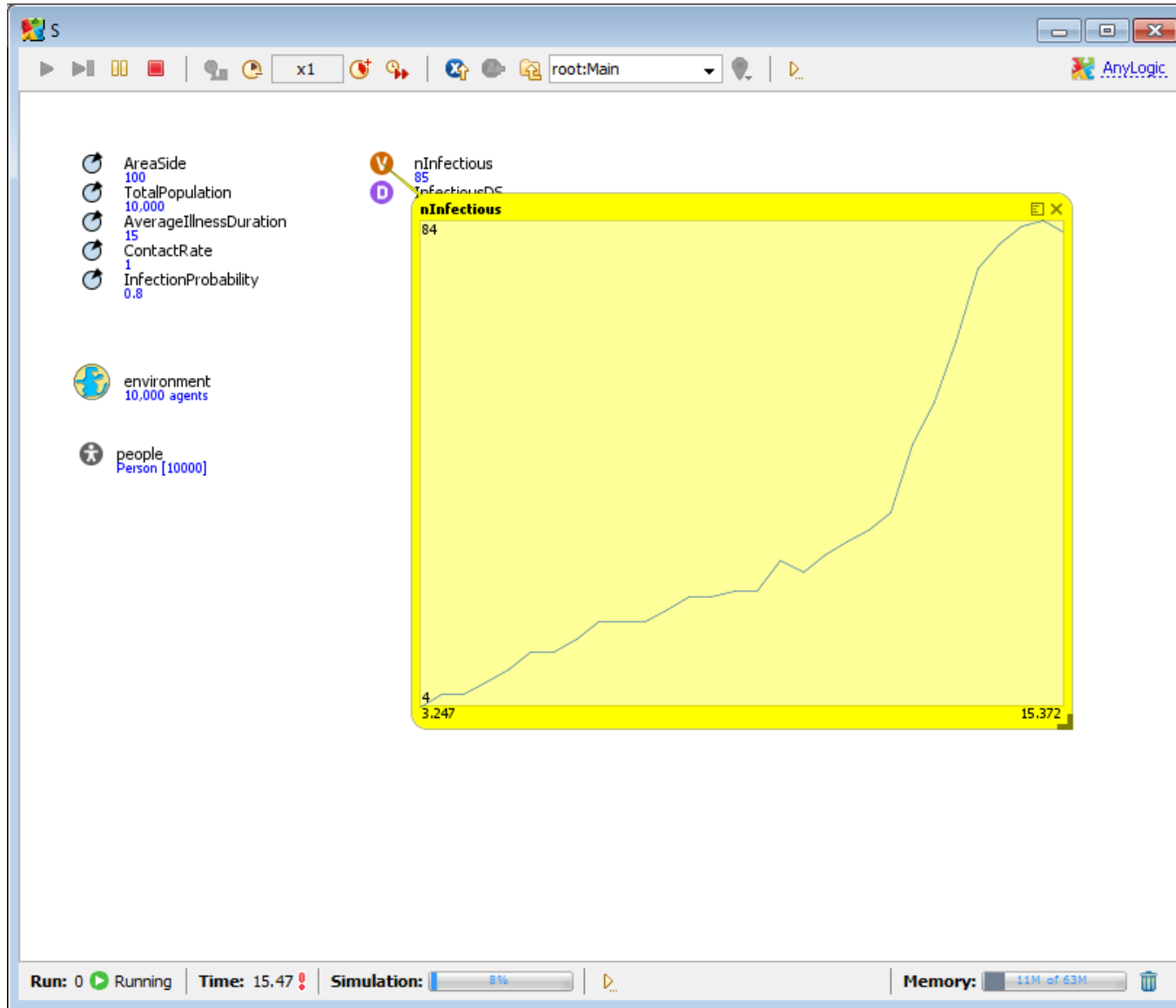
Entities:

- environment: 10,000 agents
- people: Person [10000]

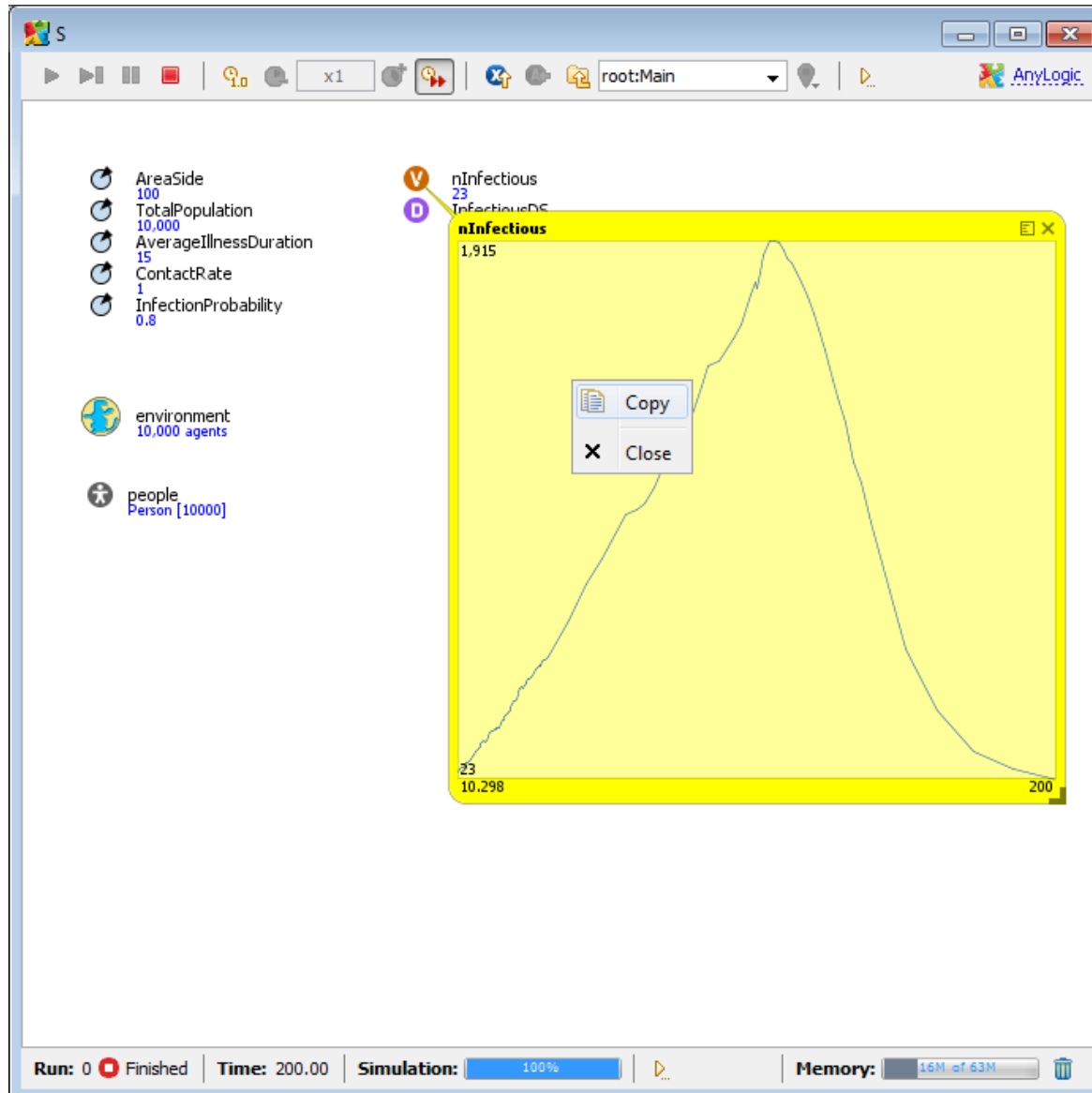
Simulation Status:

- Run: 0 (Running)
- Time: 49.97
- Simulation: 25%
- Memory: 11M of 63M

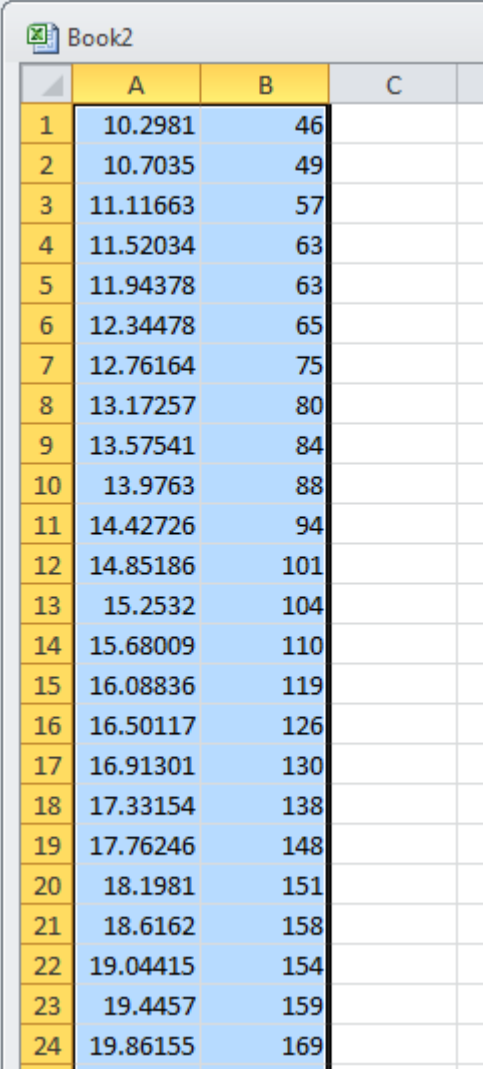
Graph of Variable



Right-Click to Copy the Numeric Data



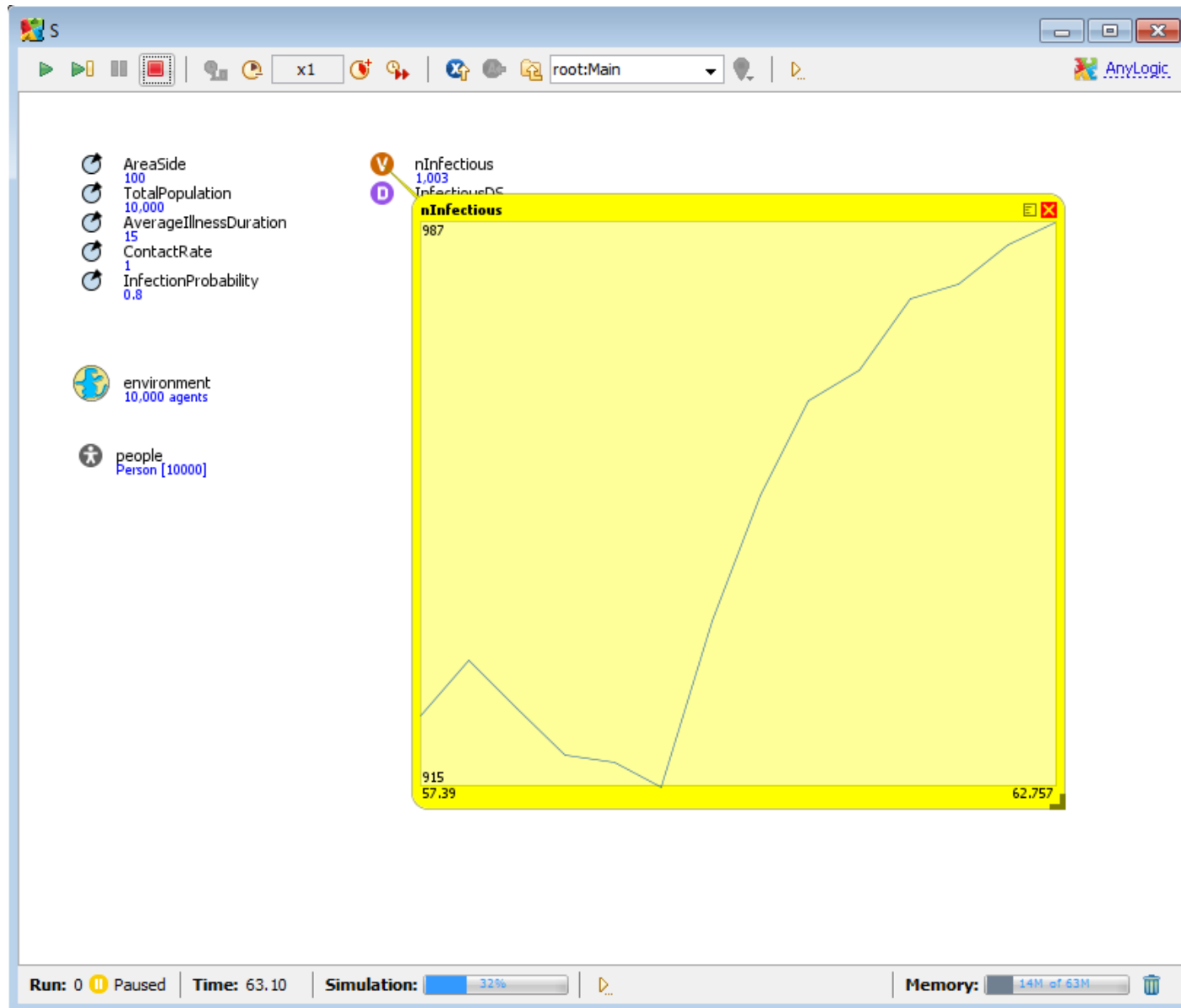
Pasting Into Excel



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 University software interface. The main workspace shows a grid with several objects: 'AreaSide', 'TotalPopulation', 'AverageIllnessDuration', 'ContactRate', 'InfectionProbability', 'environment', and 'people [...]'. The 'people' object is highlighted with a blue circle. The 'Analysis' palette on the right is open, showing various analysis options, with 'Statistics' selected. The 'Properties' console at the bottom shows the 'people - Person' object with a 'Statistics' tab selected, and an 'Add statistics' button is visible.

AnyLogic University [EVALUATION USE ONLY]

File Edit View Model Window Help

Projects

My SIR Agent Based Calibration Extensi

- Main
- Person
 - Statecharts
 - statechart
 - statechart
 - Susceptible
 - Infection
 - Infectious
 - Recovery
 - Recovered
 - Contact
- SimpleExperiment: Main
- Calibration: Main
- MonteCarlo2DHistogram: Main
 - Analysis Data
 - Presentation

Person Main SimpleExperiment

AreaSide nInfectious
TotalPopulation InfectiousDS
AverageIllnessDuration
ContactRate
InfectionProbability

environment

people [...]

Properties Console

people - Person

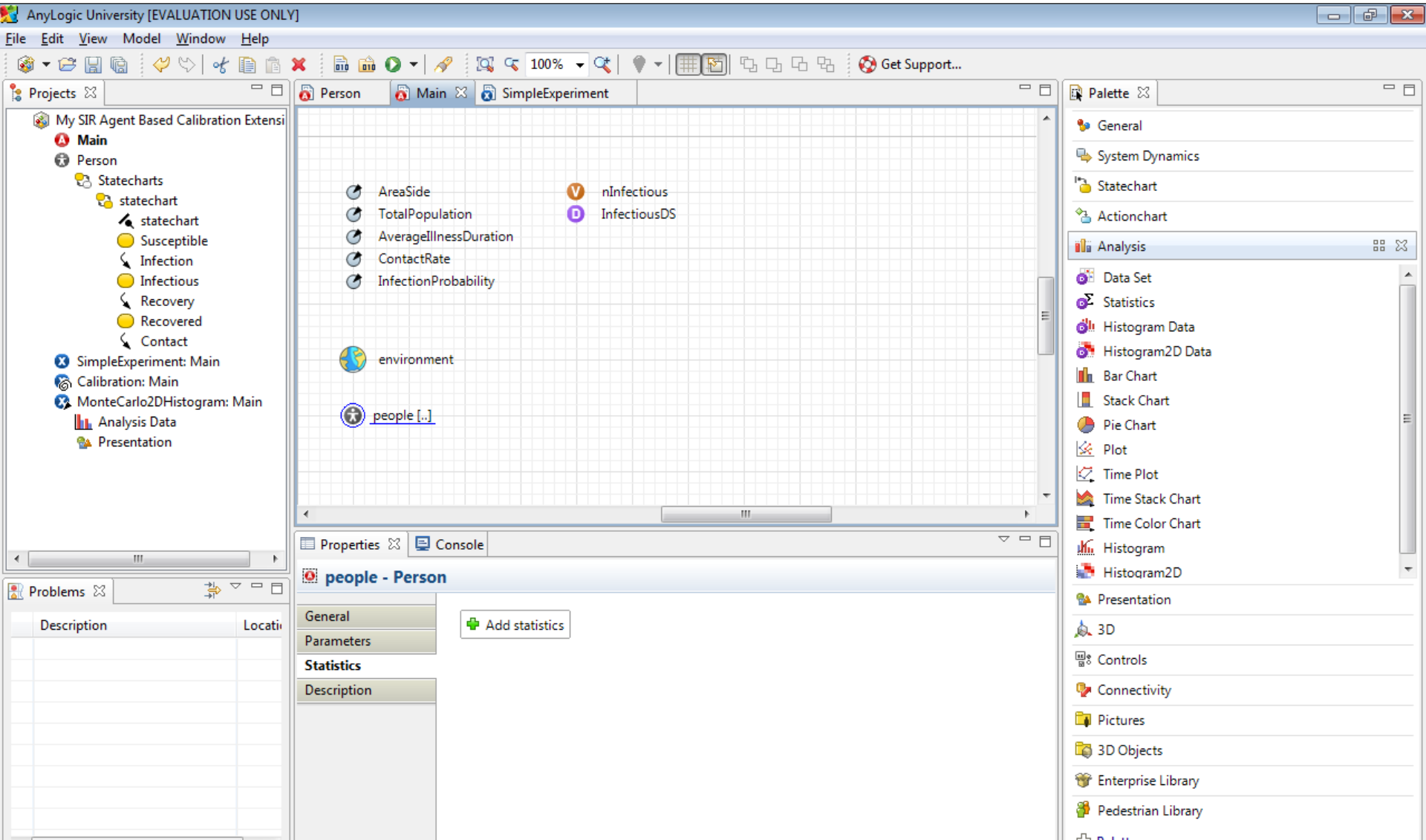
General Parameters Statistics Description

Add statistics

Palette

- General
- System Dynamics
- Statechart
- Actionchart
- Analysis
 - Data Set
 - Statistics
 - Histogram Data
 - Histogram2D Data
 - Bar Chart
 - Stack Chart
 - Pie Chart
 - Plot
 - Time Plot
 - Time Stack Chart
 - Time Color Chart
 - Histogram
 - Histogram2D
- Presentation
- 3D
- Controls
- Connectivity
- Pictures
- 3D Objects
- Enterprise Library
- Pedestrian Library

Also, Expand “Statechart” Under “Person”



Click “Add Statistics”

The screenshot displays the AnyLogic University software interface. The main workspace shows a grid with several variables: AreaSide, TotalPopulation, AverageIllnessDuration, ContactRate, InfectionProbability, environment, people [...], nInfectious, and InfectiousDS. The Properties panel at the bottom is open for the 'people - Person' object, showing tabs for General, Parameters, Statistics, and Description. A green 'Add statistics' button is visible in the Statistics tab. The Palette on the right lists various analysis tools, including Data Set, Statistics, Histogram Data, Histogram2D Data, Bar Chart, Stack Chart, Pie Chart, Plot, Time Plot, Time Stack Chart, Time Color Chart, Histogram, and Histogram2D.

AnyLogic University [EVALUATION USE ONLY]

File Edit View Model Window Help

Projects My SIR Agent Based Calibration Extensi

- Main
 - Person
 - Statecharts
 - statechart
 - statechart
 - Susceptible
 - Infection
 - Infectious
 - Recovery
 - Recovered
 - Contact
 - SimpleExperiment: Main
 - Calibration: Main
 - MonteCarlo2DHistogram: Main
 - Analysis Data
 - Presentation

Person Main SimpleExperiment

AreaSide TotalPopulation AverageIllnessDuration ContactRate InfectionProbability

nInfectious InfectiousDS

environment

people [...]

Properties Console

people - Person

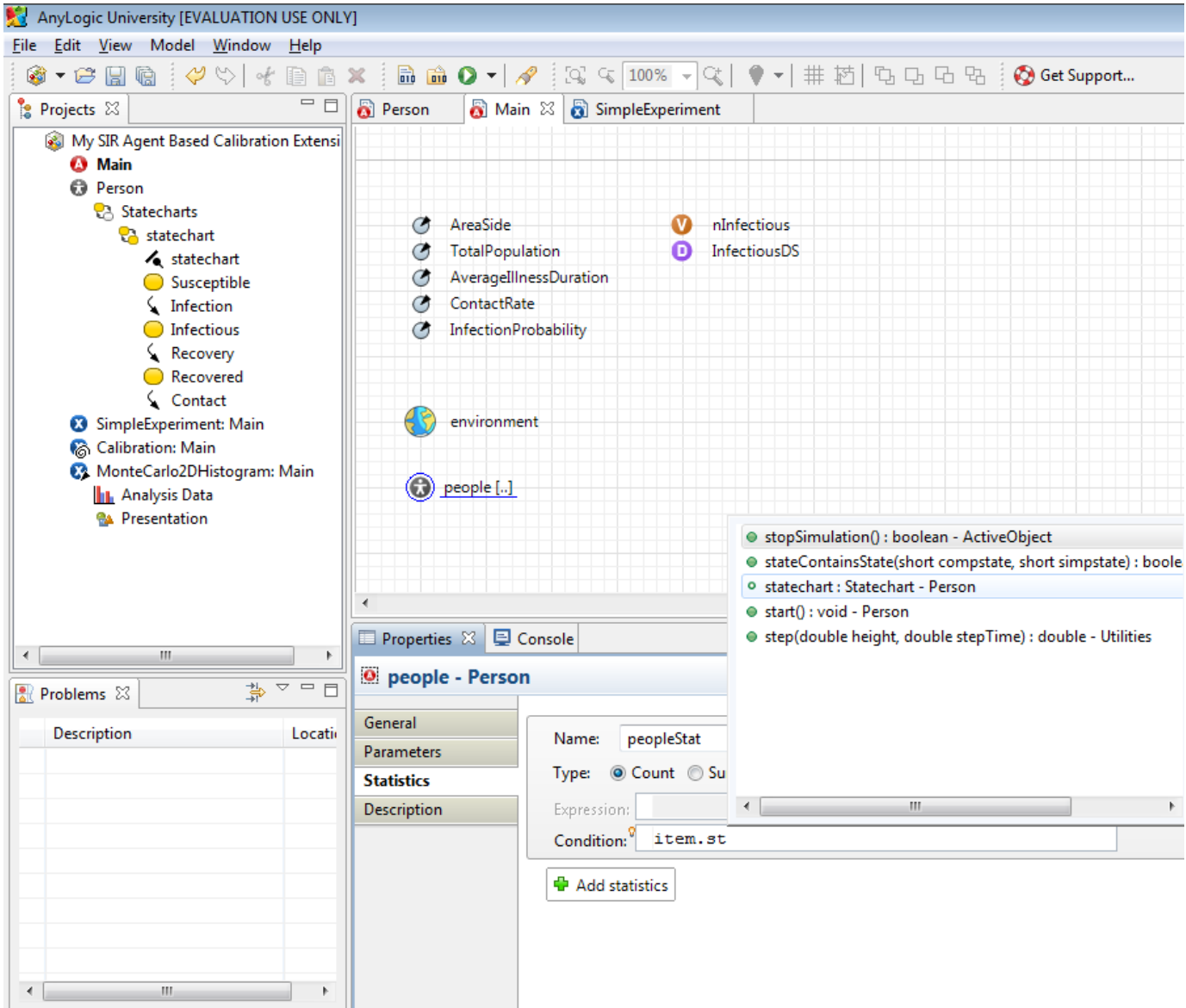
General Parameters Statistics Description

Add statistics

Palette

- General
- System Dynamics
- Statechart
- Actionchart
- Analysis
 - Data Set
 - Statistics
 - Histogram Data
 - Histogram2D Data
 - Bar Chart
 - Stack Chart
 - Pie Chart
 - Plot
 - Time Plot
 - Time Stack Chart
 - Time Color Chart
 - Histogram
 - Histogram2D
- Presentation
- 3D
- Controls
- Connectivity
- Pictures
- 3D Objects
- Enterprise Library
- Pedestrian Library
- Palettes...

Fill in the “Condition” (Predicate) on Person



Continue Typing

The screenshot displays the AnyLogic University interface for editing a statechart. The main workspace shows a statechart with several states: AreaSide, TotalPopulation, AverageIllnessDuration, ContactRate, InfectionProbability, nInfectious, and InfectiousDS. A tooltip is visible over the `isStateActive` method, providing its signature and documentation.

isStateActive

```
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

Press 'Tab' from proposal table or click for focus

Condition: `item.statechart.is`

Full Expression

The screenshot displays the AnyLogic University software interface. The main workspace shows a grid with several variables and their types: AreaSide (V), TotalPopulation (D), AverageIllnessDuration (V), ContactRate (D), InfectionProbability (V), nInfectious (V), and InfectiousDS (D). The Properties panel for the 'people - Person' agent is open, showing the 'Statistics' section. The 'Name' field is set to 'peopleStat', the 'Type' is 'Count', and the 'Condition' field contains the expression: `item.statechart.isStateActive(Person.Susceptible)`. The 'Add statistics' button is visible below the condition field.

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 statistic named 'CountSusceptible'. The 'Type' is set to 'Count', and the 'Condition' is 'item.TBProgressionStatechart.isStateActive(Person.TBSusceptible);'. A blue oval highlights the 'CountSusceptible' name and the 'Condition' field. A red arrow points from the text 'The population in which the statistics are to be calculated' to the 'person' state. A blue arrow points from the text 'What statistics we wish to calculate' to the 'CountSusceptible' name and the 'Condition' field.

What statistics we wish to calculate

person - Person

Name: CountSusceptible

Type: Count Sum Average Min Max

Expression:

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

Add Statistics

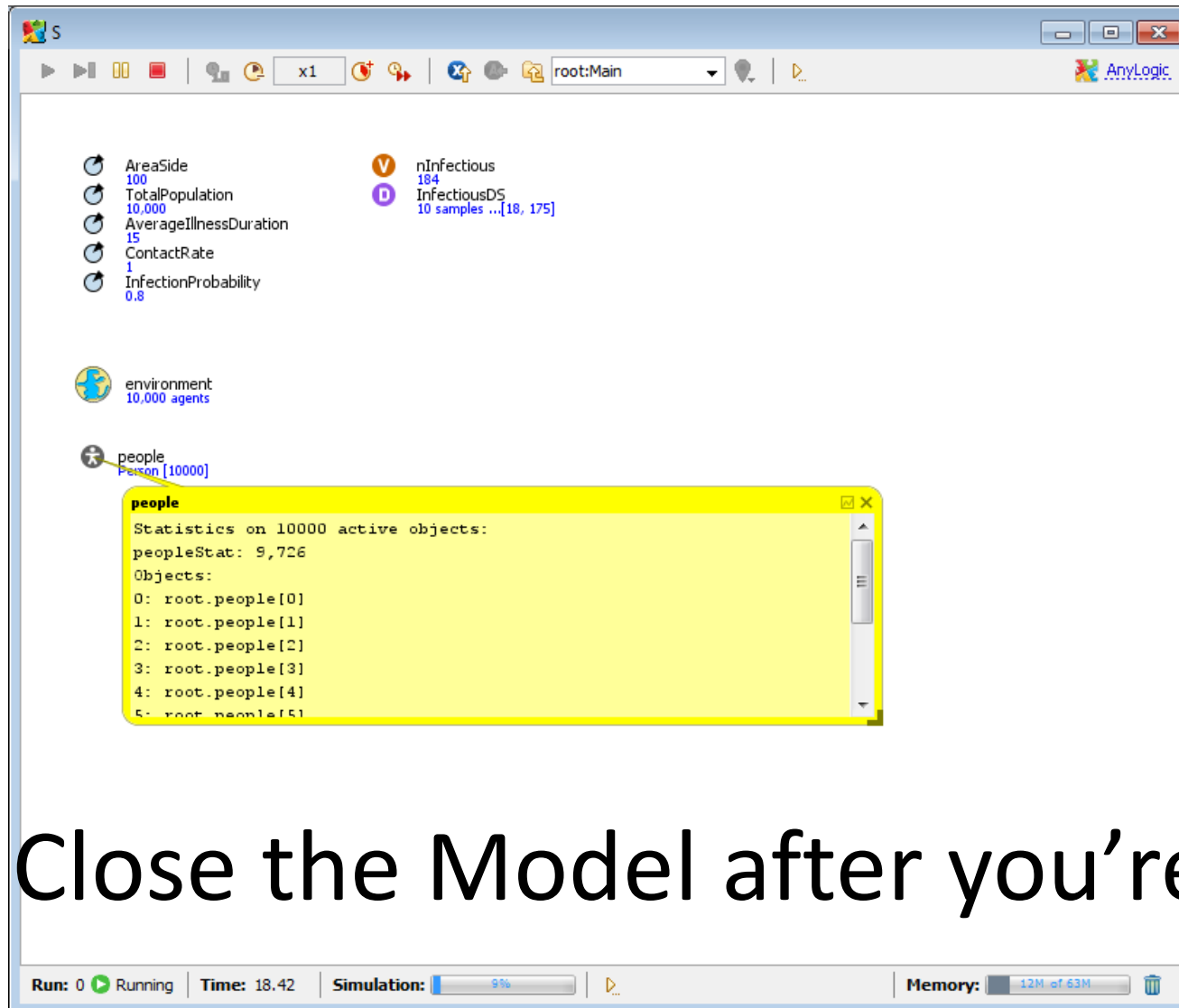
Name the Statistic “countSusceptible”

The screenshot displays the AnyLogic University software interface. The main workspace shows a model diagram with various components and statecharts. The 'Person' statechart is visible, containing states like 'Susceptible', 'Infection', 'Infectious', 'Recovery', and 'Recovered'. The 'SimpleExperiment' statechart is also visible, containing states like 'AreaSide', 'TotalPopulation', 'AverageIllnessDuration', 'ContactRate', and 'InfectionProbability'. The 'environment' and 'people [...]' components are also present.

The 'Properties' panel at the bottom shows the configuration for the 'people - Person' component. The 'Statistics' section is active, and a new statistic named 'countSusceptible' is being configured. The 'Name' field is set to 'countSusceptible', the 'Type' is set to 'Count', and the 'Condition' is set to 'item.statechart.isStateActive(Person.Susceptible)'. The 'Expression' field is empty.

The 'Palette' panel on the right shows the 'Analysis' category, which includes various data visualization options such as 'Data Set', 'Statistics', 'Histogram Data', 'Histogram2D Data', 'Bar Chart', 'Stack Chart', 'Pie Chart', 'Plot', 'Time Plot', 'Time Stack Chart', 'Time Color Chart', 'Histogram', and 'Histogram2D'.

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



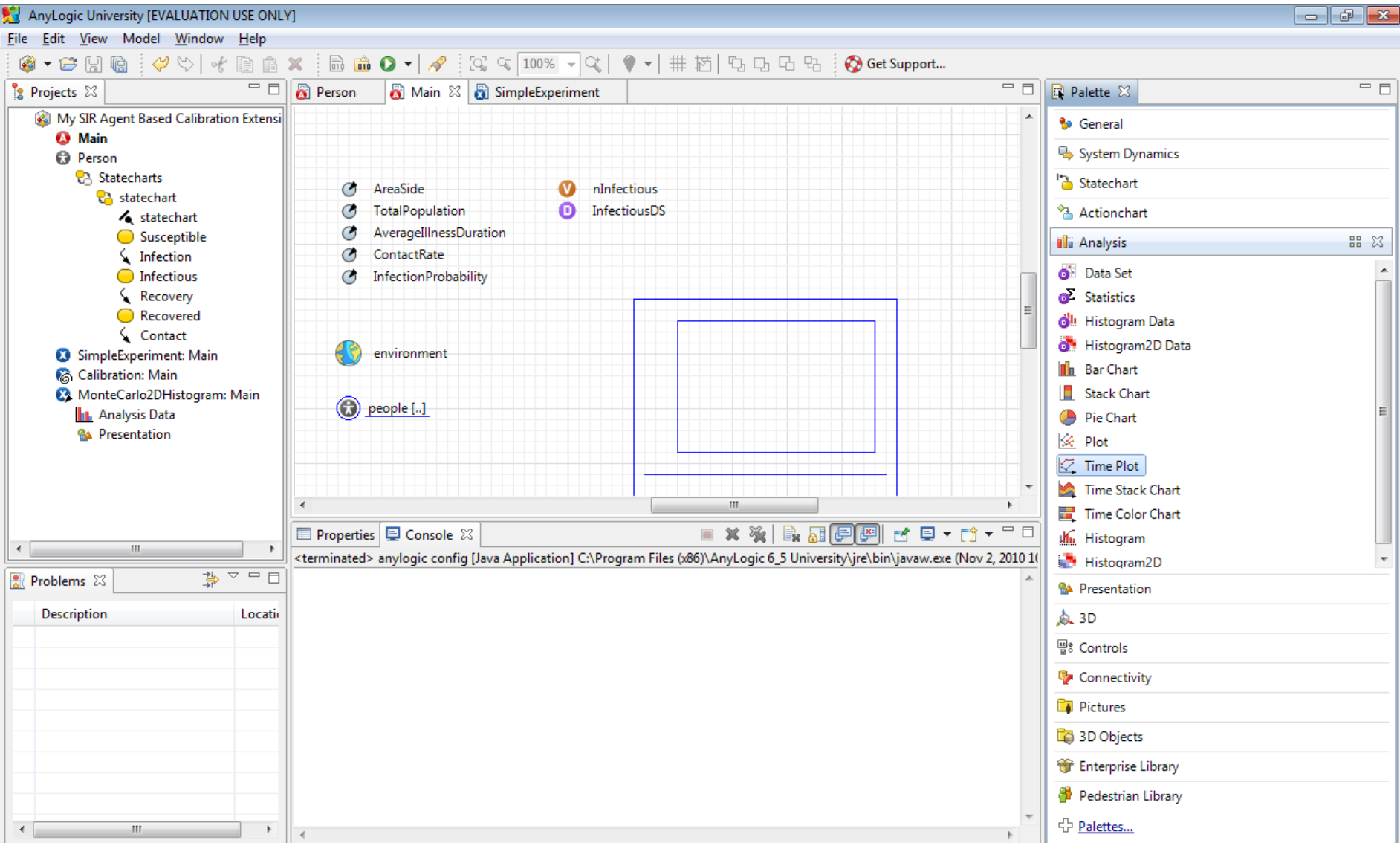
The screenshot displays the AnyLogic software interface during a simulation. The main workspace shows a hierarchical tree of objects on the left, including 'AreaSide', 'environment', and 'people'. The 'people' object is highlighted in yellow, and a yellow box is overlaid on it, displaying the following statistics:

```
people
Statistics on 10000 active objects:
peopleStat: 9,726
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]
```

The status bar at the bottom indicates the simulation is running, with a time of 18.42 and a memory usage of 12M of 63M.

Close the Model after you're done

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



Enlarge the Chart

The screenshot displays the AnyLogic University interface. The main workspace shows a time plot chart with the following data series:

- nInfectious**: A blue line with square markers, starting at 1 and dropping to 0 at time 10.
- InfectiousDS**: A purple line with diamond markers, starting at 0 and rising to 1 at time 10.

The chart's x-axis represents time from 0 to 100, and the y-axis represents values from -1 to 1. The plot is currently set to 'Auto' for the vertical scale.

The configuration panel for the 'plot - Time Plot' is visible at the bottom, with the following settings:

- Name:** plot
- Show name:**
- Ignore:**
- Public:**
- Time Window:** 100
- Vertical scale:** Auto
- From:** 0
- to:** 1
- Do not update automatically:**
- Update automatically:**
- Recurrence time:** 1

The interface also shows a Project Explorer on the left, a Palette on the right, and a Properties/Console panel at the bottom.

Click “Add Data Item”

The screenshot displays the AnyLogic University software interface. The main workspace shows a Time Plot with a vertical axis ranging from -1 to 1 and a horizontal axis from 0 to 100. The plot area is currently empty, with a blue border. To the left of the plot is a list of variables: AreaSide, TotalPopulation, AverageIllnessDuration, ContactRate, InfectionProbability, environment, and people [...]. Two variables, nInfectious (marked with a 'V') and InfectiousDS (marked with a 'D'), are highlighted with orange and purple circles respectively. Below the plot is the 'plot - Time Plot' configuration panel. The 'General' tab is active, showing the following settings:

- Name: plot
- Show name:
- Ignore:
- Public:
- Add data item:
- Time Window: 100
- Vertical scale: Auto
- From: 0 to: 1
- Do not update automatically:
- Update automatically:
- Recurrence time: 1

The right-hand side of the interface features a Palette with various chart types under the 'Analysis' category, including Data Set, Statistics, Histogram Data, Histogram2D Data, Bar Chart, Stack Chart, Pie Chart, Plot, Time Plot, Time Stack Chart, Time Color Chart, Histogram, and Histogram2D.

Put in “people.” and Press Ctrl-Space

The screenshot displays the AnyLogic University interface. On the left, a project tree shows a model named 'My SIR Agent Based Calibration Extension' with a 'Main' component containing 'Person' and 'Statecharts'. The 'Person' component includes statecharts for 'Susceptible', 'Infection', 'Infectious', 'Recovery', 'Recovered', and 'Contact'. The 'SimpleExperiment: Main' component includes 'Calibration: Main', 'MonteCarlo2DHistogram: Main', 'Analysis Data', and 'Presentation'.

The main workspace shows a time plot with a green line representing the 'people' variable. The plot has a y-axis ranging from 0.6 to 0.85 and an x-axis labeled 'Date'. The line starts at approximately 0.85, remains constant until about 0.84, then gradually decreases to about 0.7, and finally drops sharply to 0.65. A legend on the left lists variables: 'AreaSide', 'TotalPopulation', 'AverageIllnessDuration', 'ContactRate', 'InfectionProbability', 'environment', and 'people [...]'. A tooltip for the 'people' variable is visible, showing a list of methods such as 'sum(String fieldName, String triggerFieldName) : double', 'sum(String fieldName) : double - ActiveObjectCollection', 'min(Collection<? extends ActiveObject> activeObjects, String triggerFieldName) : double - ActiveObjectCollection', 'min(String fieldName) : double - ActiveObjectCollection', 'max(String fieldName, String triggerFieldName) : double - ActiveObjectCollection', 'max(String fieldName) : double - ActiveObjectCollection', 'average(String fieldName, String triggerFieldName) : double - ActiveObjectCollection', 'average(String fieldName) : double - ActiveObjectCollection', 'countSusceptible() : int - _people_Class', 'size() : int - ActiveObjectArrayList', and 'count(String triggerFieldName) : int - ActiveObjectCollection'.

The 'plot - Time Plot' properties window is open, showing the following configuration:

- Name: plot
- Value: people.
- Point style: [dropdown]
- Color: oliveDrab
- Draw line: [checked]
- Line width: 1 pt
- Interpolation: Linear

At the bottom of the properties window, there is an 'Add data item' button.

Choose “Count Susceptible”

The screenshot displays the AnyLogic software interface for an SIR model simulation. The main window shows a time plot with a green line representing the 'Count Susceptible' variable. The plot is titled 'plot - Time Plot' and is located in the 'Properties' panel. The 'Value' field is set to 'people.countSusceptible()'. The 'Color' is set to 'oliveDrab'. The 'Line width' is set to '1 pt' and the 'Interpolation' is set to 'Linear'. The 'Public' checkbox is checked.

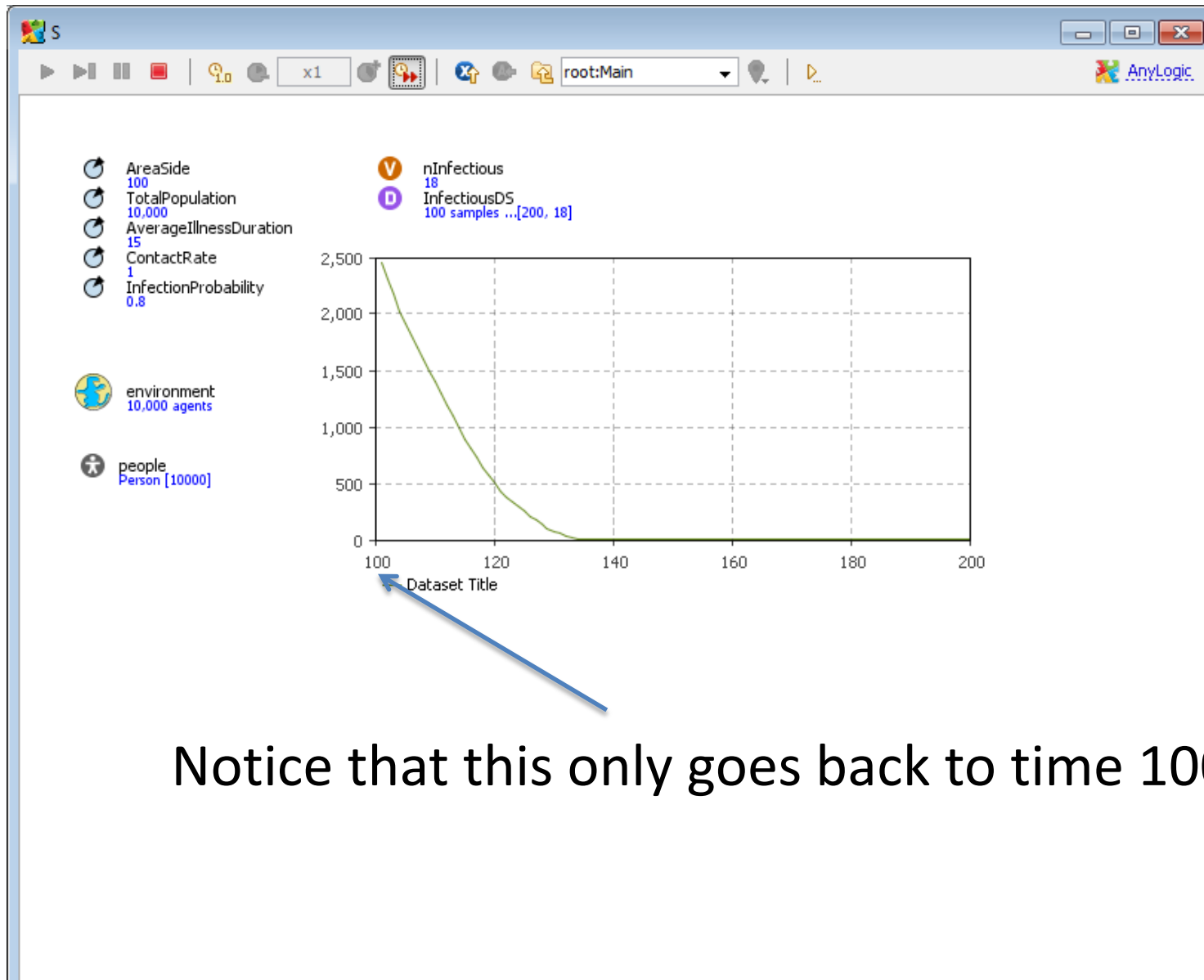
The plot shows the following data points (approximate values):

Time	Count Susceptible
0	0.85
20	0.85
40	0.80
60	0.75
75	0.70
80	0.65
100	0.65

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

- My SIR Agent Based Calibration Extensi
 - Main
 - Person
 - Statecharts
 - statechart
 - statechart
 - Susceptible
 - Infection
 - Infectious
 - Recovery
 - Recovered
 - Contact
- SimpleExperiment: Main
- Calibration: Main
- MonteCarlo2DHistogram: Main
 - Analysis Data
 - Presentation

Now Run the Model



Notice that this only goes back to time 100

Stop the Simulation, and Click on the Plot.

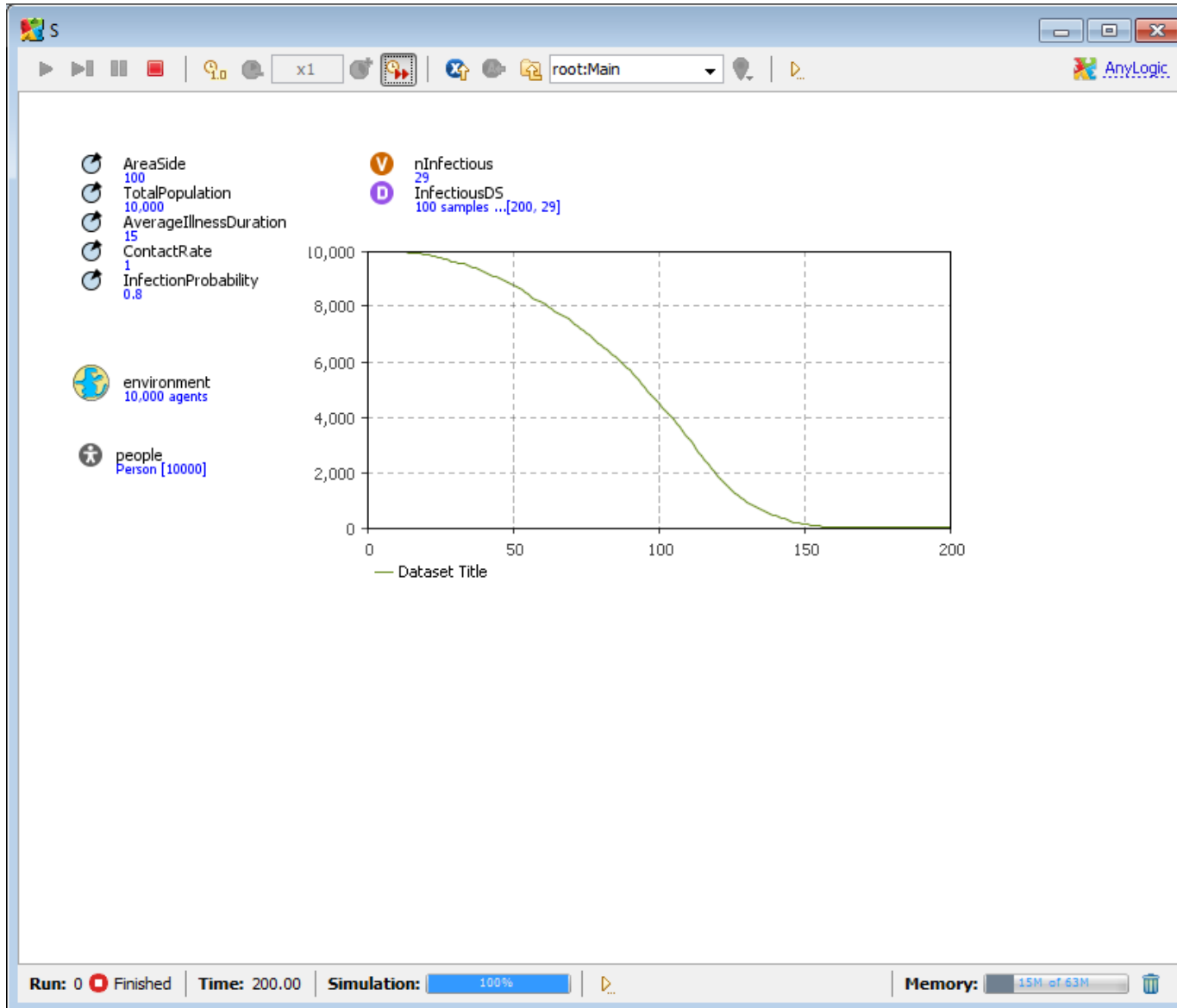
Change Time Window & Display Size to 200

The screenshot displays the AnyLogic University interface. The main window shows a plot titled "plot - Time Plot" with a green line representing the variable "nInfectious". The plot's time window is set to 200, and the vertical scale is set to "Auto". The plot shows a sharp increase in the number of infectious individuals around time 100, followed by a gradual decline. The plot is currently selected, and the "plot - Time Plot" properties panel is open at the bottom.

plot - Time Plot Properties:

- General:** Draw line, Line width: 1 pt, Interpolation: Linear
- Advanced:** Add data item
- Dynamic:** Time Window: 200
- Appearance:** Vertical scale: Auto, From: 0, to: 1
- Description:** Do not update automatically, Update automatically, Recurrence time: 1
- Display up to 200 latest samples (applies to "Value" data items only)

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 to string method

Output: Datasets

The screenshot displays the AnyLogic Advanced [EDUCATIONAL USE ONLY] interface. The top toolbar includes standard file operations and a 'Get Support' button. 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:

- Project: TBv1*
 - 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

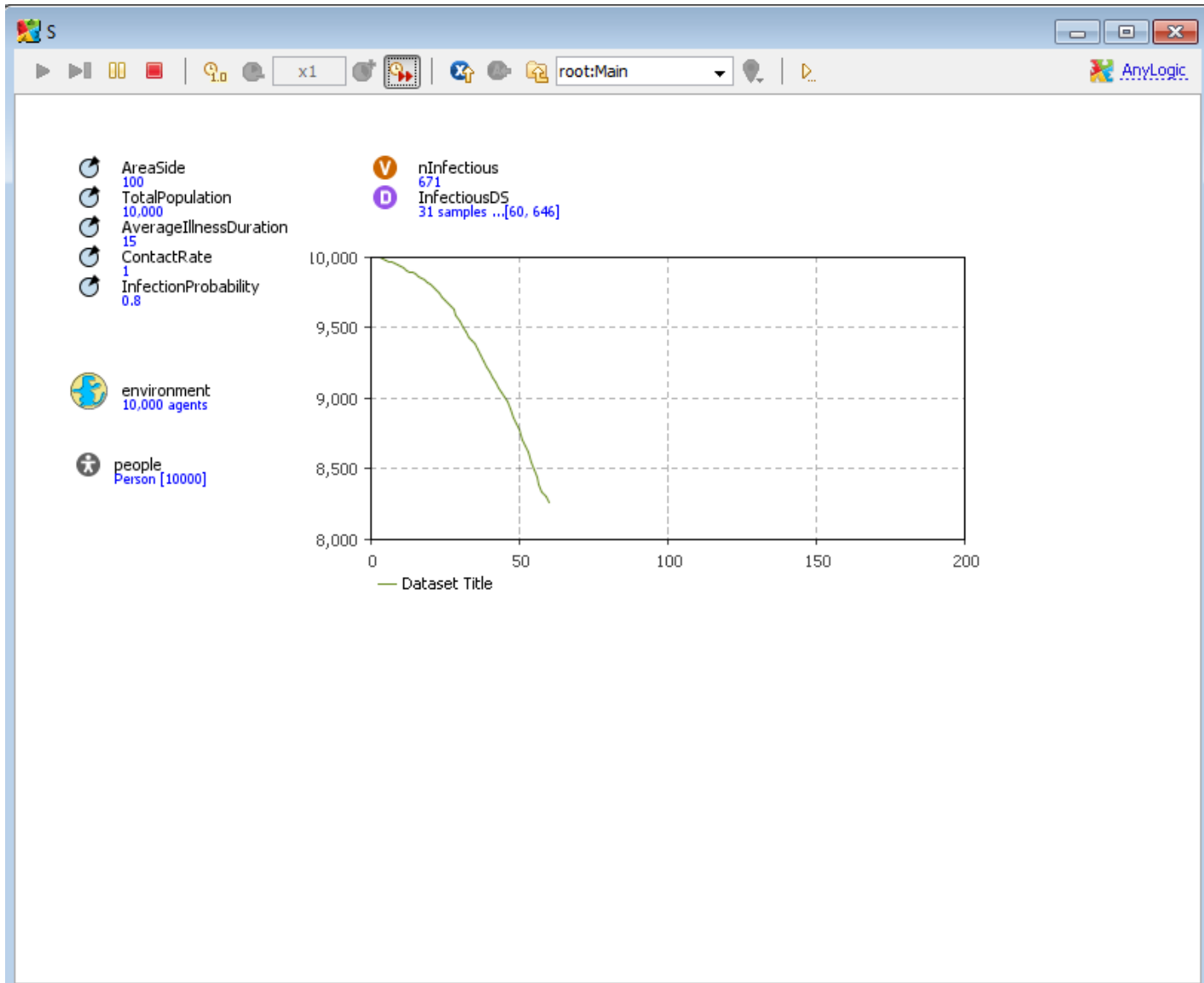
The bottom panel shows the configuration for the 'dsSusceptibleCount' Data Set:

- Name: dsSusceptibleCount
- General: 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
- Update options: Do not update automatically, Update automatically
- Begin at time: 0.0, Recurrence time: 1
- Optional date and time: October 29, 2009, 2:07:08 AM

The bottom-left corner features a 'Problems' table with columns for 'Description' and 'Location':

Description	Location

Run the Experiment & Click on “Infectious DS”



Click on “InfectiousDS” to See Data in Dataset

The screenshot shows the AnyLogic software interface. On the left, there are several simulation parameters and agents:

- AreaSide: 100
- TotalPopulation: 10,000
- AverageIllnessDuration: 15
- ContactRate: 1
- InfectionProbability: 0.8
- environment: 10,000 agents
- people: Person [10000]

In the center, there is a data table for 'InfectiousDS' with 32 samples. The table shows the following data:

Sample	Value
0	1
2	2
4	6
6	1
8	2
10	2
12	3
14	7
16	8
18	1

The status bar at the bottom indicates: Run: 0 Paused, Time: 63.10, Simulation: 32%, Memory: 13M of 63M.

Right Click and Select "Copy"

The screenshot shows the AnyLogic software interface. On the left, there is a list of variables and their values:

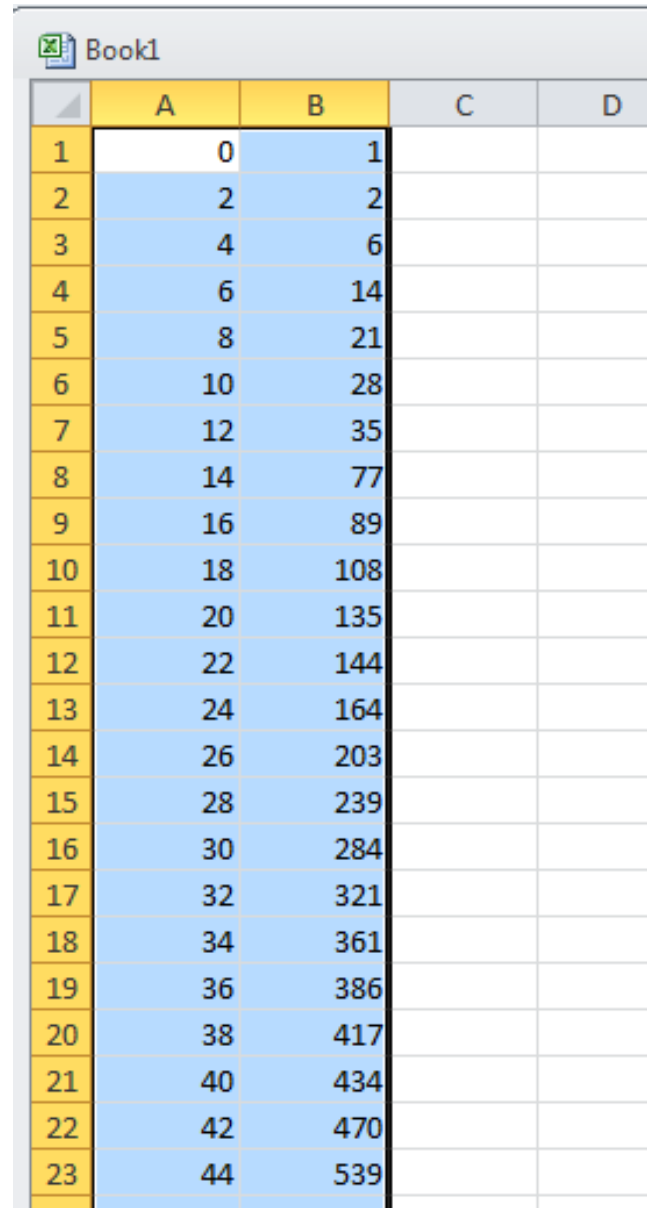
- AreaSide: 100
- TotalPopulation: 10,000
- AverageIllnessDuration: 15
- ContactRate: 1
- InfectionProbability: 0.8
- environment: 10,000 agents
- people: Person [10000]

In the center, a data table titled "InfectiousDS" is displayed. A context menu is open over the table, showing "Copy" and "Close" options. The table data is as follows:

Time	Value
0	1
2	2
4	6
6	14
8	21
10	27
12	33
14	39
16	45
18	51
20	57
22	63
24	69
26	75
28	81
30	87
32	93
34	99
36	105
38	111
40	117
42	123

At the bottom of the interface, the status bar shows: Run: 0 Paused, Time: 68.52, Simulation: 34%, and Memory: 14M of 63M.

Call Up Excel and Paste into It



The image shows a screenshot of an Excel spreadsheet titled "Book1". The spreadsheet has four columns labeled A, B, C, and D. Column A contains even numbers from 0 to 44, and column B contains a sequence of numbers from 1 to 539. Columns C and D are empty. The rows are numbered 1 through 23.

	A	B	C	D
1	0	1		
2	2	2		
3	4	6		
4	6	14		
5	8	21		
6	10	28		
7	12	35		
8	14	77		
9	16	89		
10	18	108		
11	20	135		
12	22	144		
13	24	164		
14	26	203		
15	28	239		
16	30	284		
17	32	321		
18	34	361		
19	36	386		
20	38	417		
21	40	434		
22	42	470		
23	44	539		

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 is labeled 'Time'.

The 'Properties' dialog for the 'datasetAbsolutePrevalence' data set is open, showing the following configuration:

- Name:** datasetAbsolutePrevalence
- Show Name
- Ignore
- Public
- Show As Line

Description

- Use time as horizontal axis value
- Horizontal axis value: [Empty]
- 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' window 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.

Chart Use of Datasets

The screenshot displays the AnyLogic Advanced [EDUCATIONAL USE ONLY] interface. The main window shows a time plot for the variable `dsSusceptibleCount` from the `person` dataset. The plot shows a line graph with data points connected by a line, starting at approximately 0.35 at time 0, rising to a peak of 0.7 at time 0.3, and then declining to about 0.58 at time 0.5. The x-axis represents time from 0 to 0.5, and the y-axis represents the count from 0.3 to 0.7.

The configuration panel for the `TimePlotAgentCount - Time Plot` is visible below the plot. It includes the following settings:

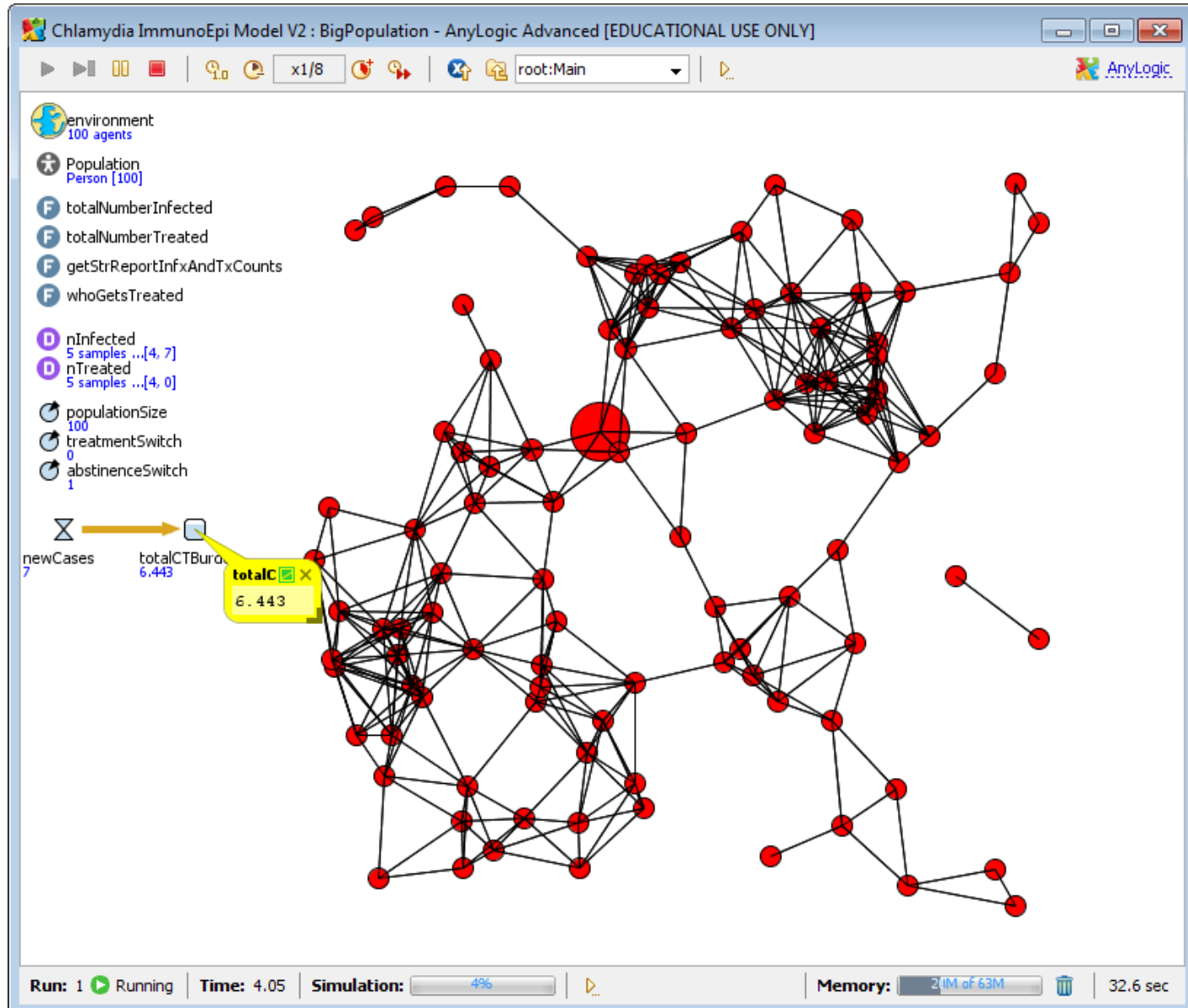
- Name: `TimePlotAgentCount`
- Show Name:
- Ignore:
- Public:
- Title: `Count Susceptibles`
- Data Set: `dsSusceptibleCount`
- Point Style:
- Color: `darkorange`
- Draw line:
- Line Width: `1 pt`
- Interpolation: `Linear`

The left sidebar shows the project structure for `TBv1*`, including `Main` (Parameters, Functions, Environments, Embedded Objects, Analysis Data, Presentation) and `Person` (Parameters).

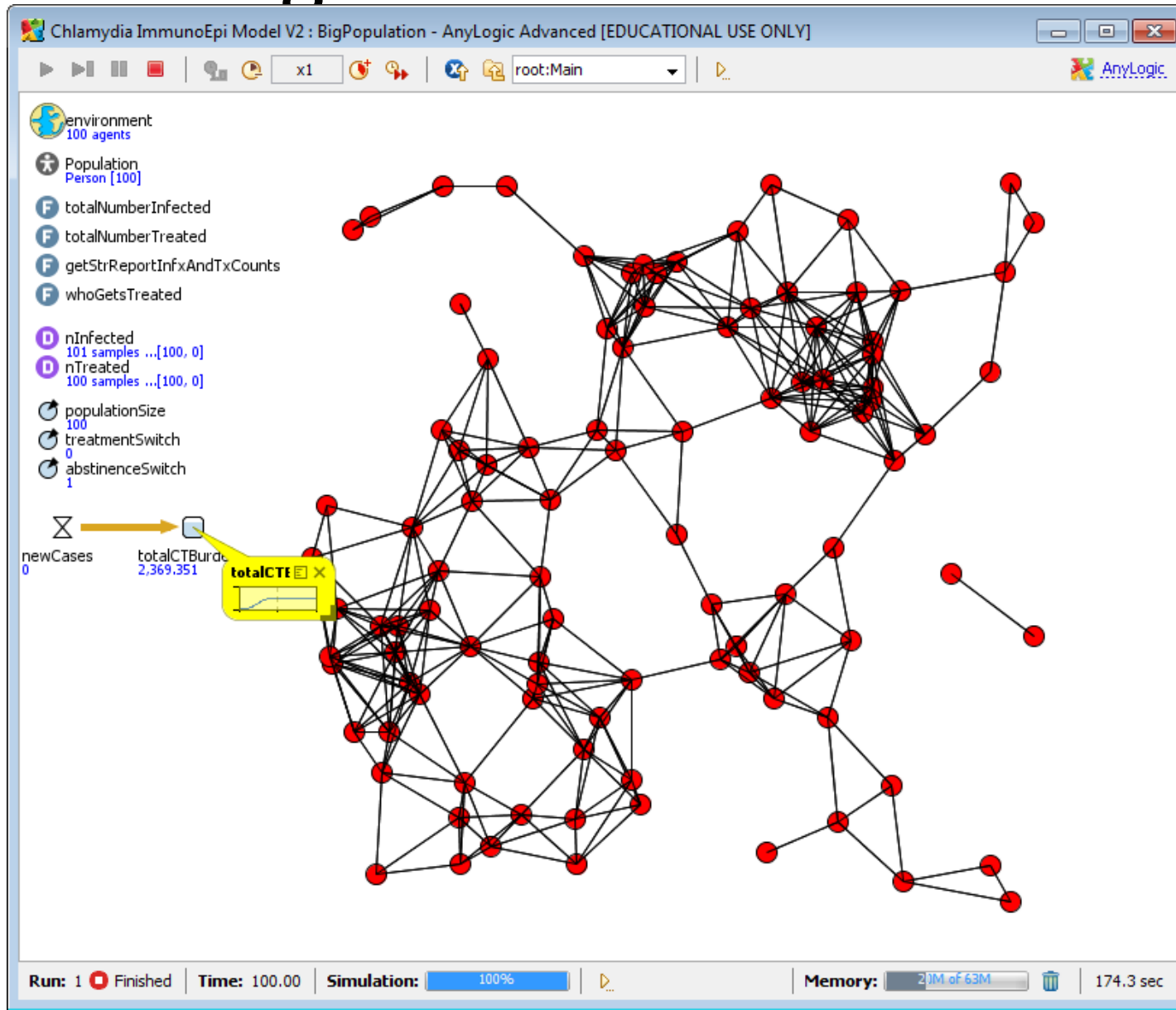
The bottom left shows the `Problems` panel with a table:

Description	Location

Ad-hoc Export



Begins as a Small Chart



Copying Data

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]

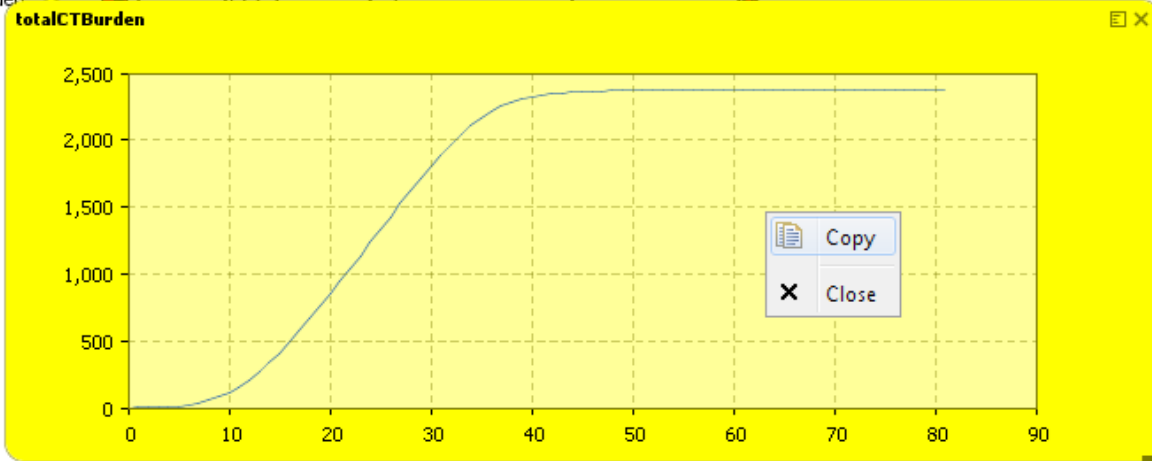
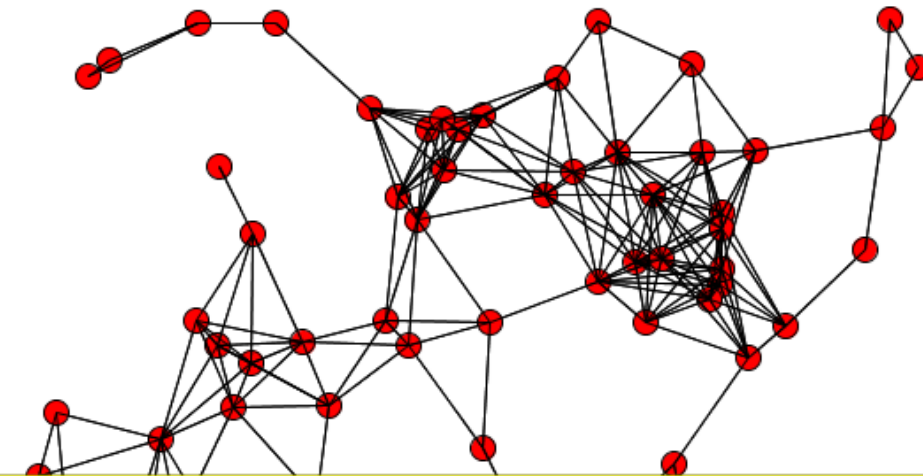
populationSize
100

treatmentSwitch
0

abstinenceSwitch
1

newCases
0

totalCTBurden
2,369.351



totalCTBurden

2,500

2,000

1,500

1,000

500

0

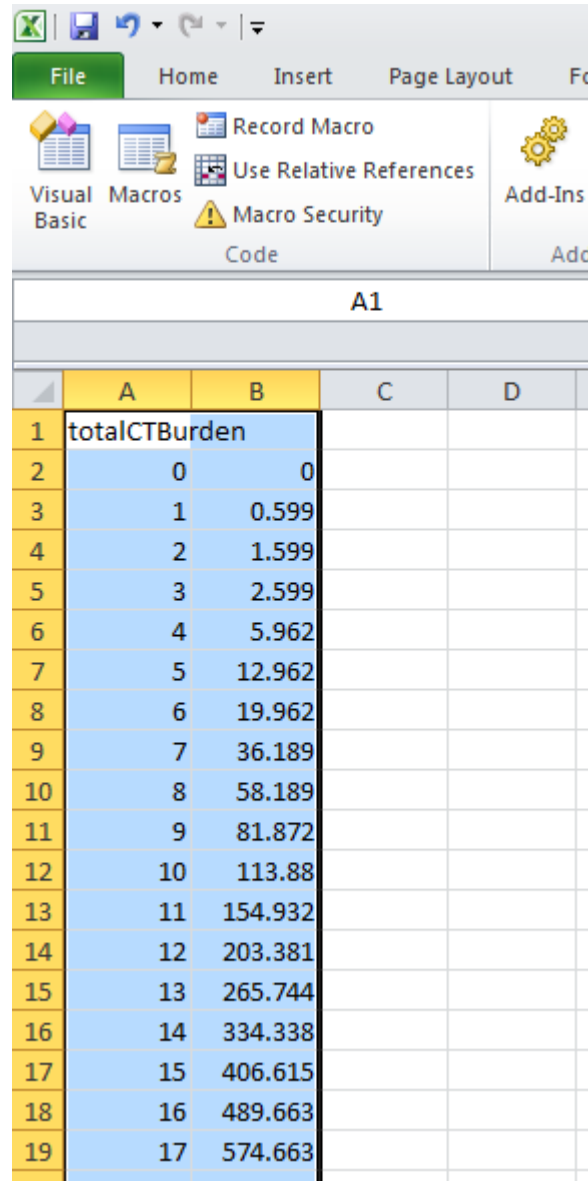
0 10 20 30 40 50 60 70 80 90

Copy

Close

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

Data Exported from Ad-Hoc Chart

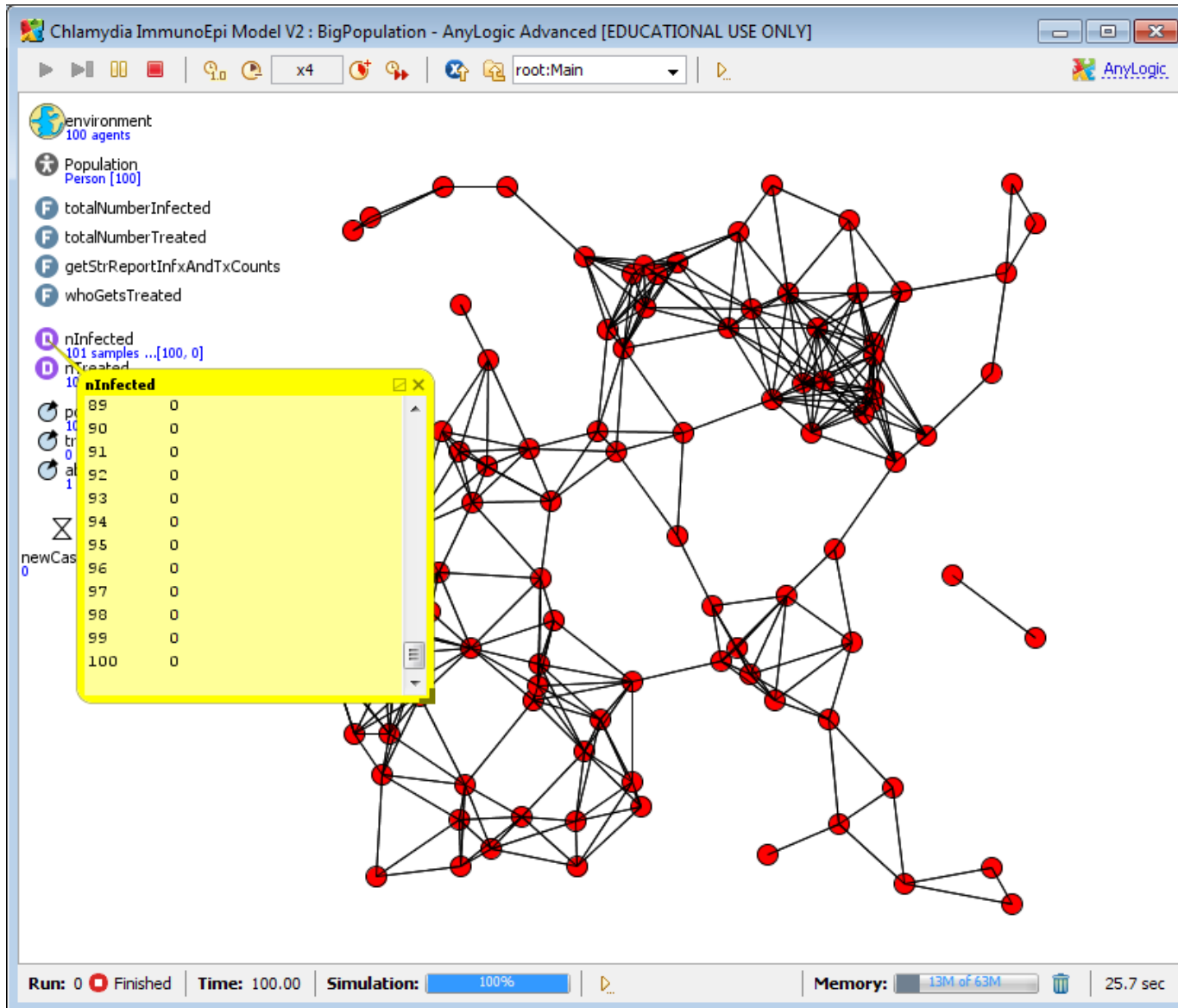


	A	B	C	D
1	totalCTBurden			
2	0	0		
3	1	0.599		
4	2	1.599		
5	3	2.599		
6	4	5.962		
7	5	12.962		
8	6	19.962		
9	7	36.189		
10	8	58.189		
11	9	81.872		
12	10	113.88		
13	11	154.932		
14	12	203.381		
15	13	265.744		
16	14	334.338		
17	15	406.615		
18	16	489.663		
19	17	574.663		

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 software interface for a simulation titled "Chlamydia ImmunoEpi Model V2 : BigPopulation - AnyLogic Advanced [EDUCATIONAL USE ONLY]". The main workspace shows a network diagram with red nodes and black edges. A context menu is open over the "nInfected" variable in the left-hand tree view. The menu contains a table of data and two buttons: "Copy" and "Close".

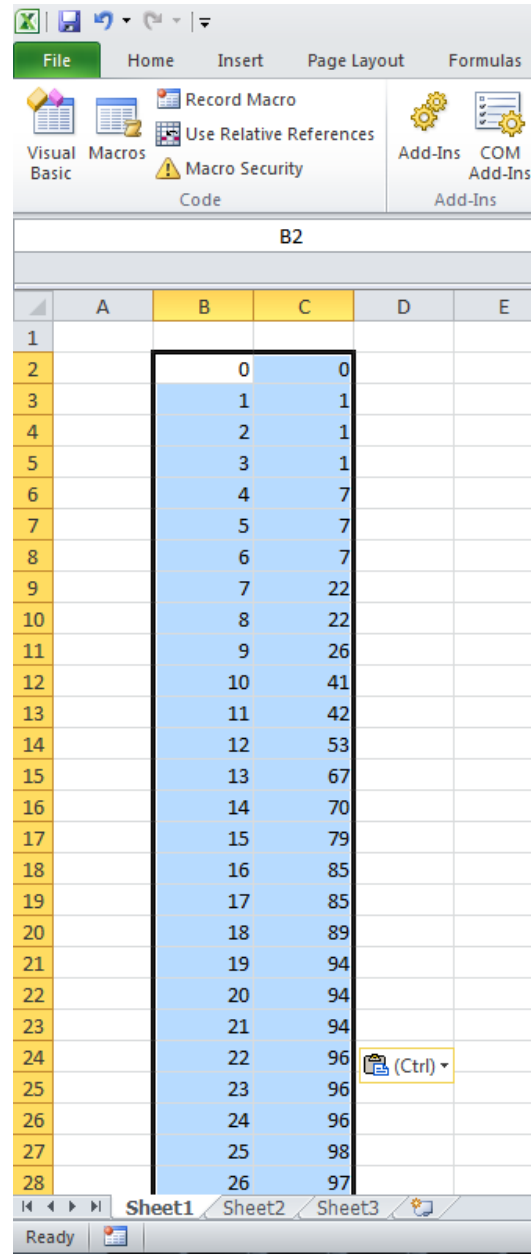
Environment: 100 agents

- Population: Person [100]
- totalNumberInfected
- totalNumberTreated
- getStrReportInfxAndTxCounts
- whoGetsTreated
- nInfected: 101 samples ...[100, 0]
- nTreated
- nInfected
- nTreated
- newCas: 0

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
- Export to files
- Writing to console
- [AnyLogic Professional] Dataset archiving
- Export to databases

Output to Console

- Pros

- Easy to program

- `ActiveObject.traceIn(Stringstr)` **outputs string to console**
- `System.out.println(Stringstr)`

- 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

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

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 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

Example code to Export Dataset to File

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

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

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
 - Add reference to database libraries
- Each time 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

Database Dependencies (MySQL database)

The screenshot shows the 'Properties' window of an IDE, specifically the 'Dependencies' tab for a model named 'RecreationOfAzizaModelCubeDataStorage_V4NoA...CGLTBI_NDOModificationsForTestingV2 - Model'. The window is divided into two main sections: 'AnyLogic libraries required to build the model:' and 'Jar files and class folders required to build the model:'. The first section contains an empty table with columns for Name, Version, and Location. The second section contains a table with a 'Location' column, listing 'mysql-connector-java-5.1.13-bin.jar' as a dependency.

Properties Console

RecreationOfAzizaModelCubeDataStorage_V4NoA...CGLTBI_NDOModificationsForTestingV2 - Model

General

Dependencies

Description

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.jar

Example Simple Database Class

```
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();
    }
}
```

Setup for Database Class

The screenshot shows the 'Properties' window for a variable named 'myConn'. The window has two tabs: 'Properties' and 'Console'. The 'Properties' tab is active, showing the following configuration:

- myConn - Plain Variable**
- General** (selected in the left sidebar)
- Name:** myConn
- Show Name
- Ignore
- Public
- Show At Runtime
- Access:** public
- Static
- Constant
- Save in snapshot
- Type:** boolean, int, double, String, Other: MyDB
- Initial Value:** new MyDB ()

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