

# Collecting, Outputting & Inputting Data in AnyLogic

Nathaniel Osgood

MIT 15.879

April 4, 2012

# 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 University software interface. The main window shows a project titled "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 software interface includes a menu bar (File, Edit, View, Model, Window, Help), a toolbar with various icons, and a palette on the right side. The palette is organized into categories: 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. The bottom of the screen shows a "Properties" window for the selected "SIR Agent Based Calibration" model, with 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

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 Performs model runs with specified parameters, supports virtual and real-time modes, animation, model debugging
- Optimization
- Parameters Variation
- Compare Runs
- Monte Carlo
- Sensitivity Analysis
- Calibration
- Custom

Copy model time settings from: Calibration

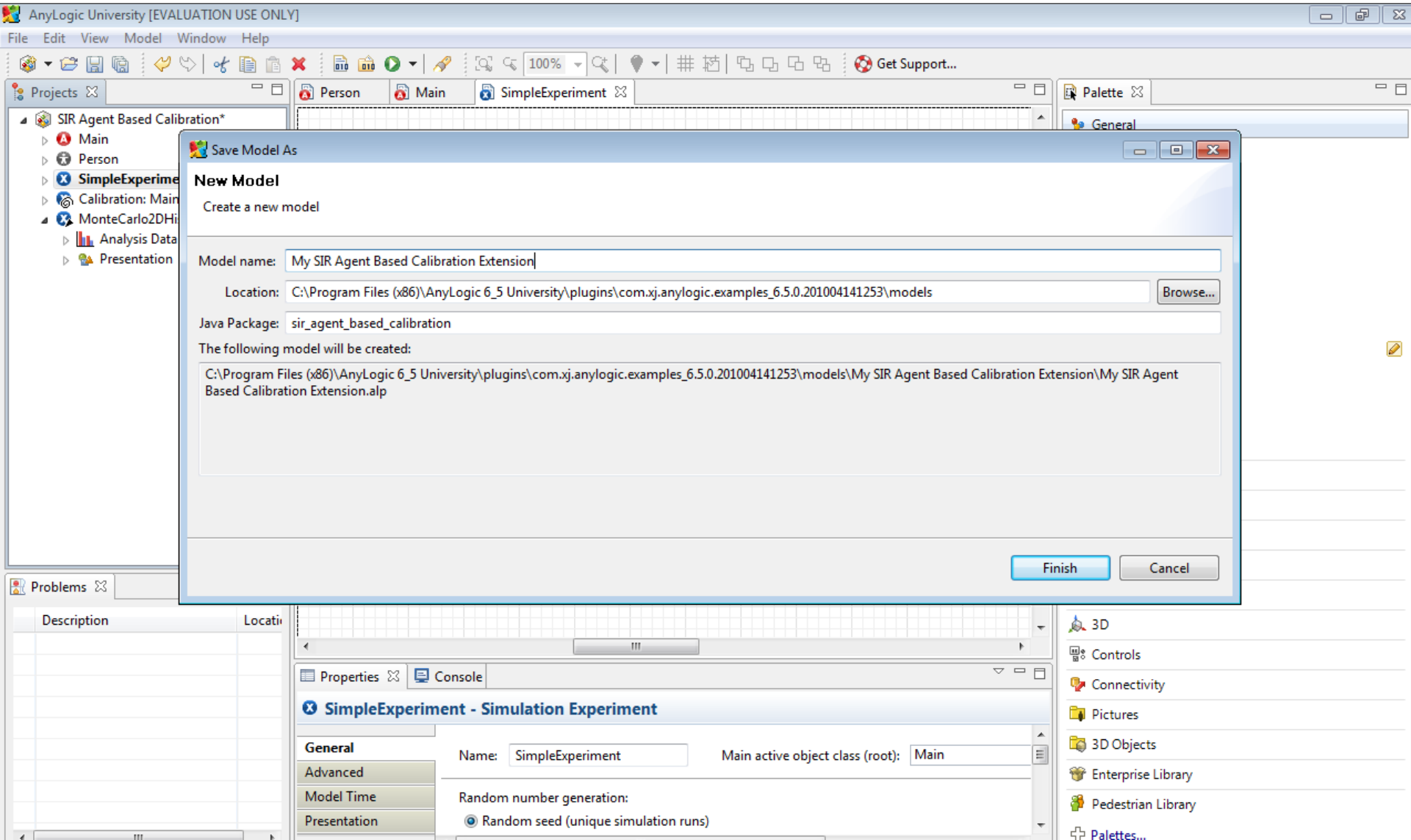
< Back Next > Finish Cancel

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...



# 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 for "My SIR Agent Based Calibration Extension". The model is structured as follows:

- My SIR Agent Based Calibration Extension
  - Main
    - Person
      - SimpleExperiment: Main (selected)
      - Calibration: Main
      - MonteCarlo2DHistogram: Main
        - Analysis Data
        - Presentation

The main workspace contains several variables and objects:

- AreaSide (clock icon)
- TotalPopulation (clock icon)
- AverageIllnessDuration (clock icon)
- ContactRate (clock icon)
- InfectionProbability (clock icon)
- nInfectious (orange circle with 'V')
- InfectiousDS (purple circle with 'D')
- environment (globe icon)
- people [...] (plus icon)

The "Recent Experiment" menu is open, showing the following options:

- My SIR Agent Based Calibration Extension / SimpleExperiment (selected)
- My SIR Agent Based Calibration Extension / Calibration
- My SIR Agent Based Calibration Extension / MonteCarlo2DHistogram

The "Properties" panel at the bottom shows the configuration for "SimpleExperiment - Simulation Experiment":

- General
  - Name: SimpleExperiment
  - Main active object class (root): Main
- Advanced
  - Random number generation:
    - Random seed (unique simulation runs) (selected)
- Model Time
- Presentation

The "Palette" on the right side of the interface lists various components available for the model:

- 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 hierarchical tree of variables and objects. The variable 'nInfectious' is highlighted with a yellow tooltip that displays its current value of 716. The tooltip also shows a small 'nInfer' icon and a close button. The background tree lists several variables: AreaSide (100), TotalPopulation (10,000), AverageIllnessDuration (15), ContactRate (1), and InfectionProbability (0.8). Below these are objects: environment (10,000 agents) and people (Person [10000]). 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.

AreaSide  
100

TotalPopulation  
10,000

AverageIllnessDuration  
15

ContactRate  
1

InfectionProbability  
0.8

environment  
10,000 agents

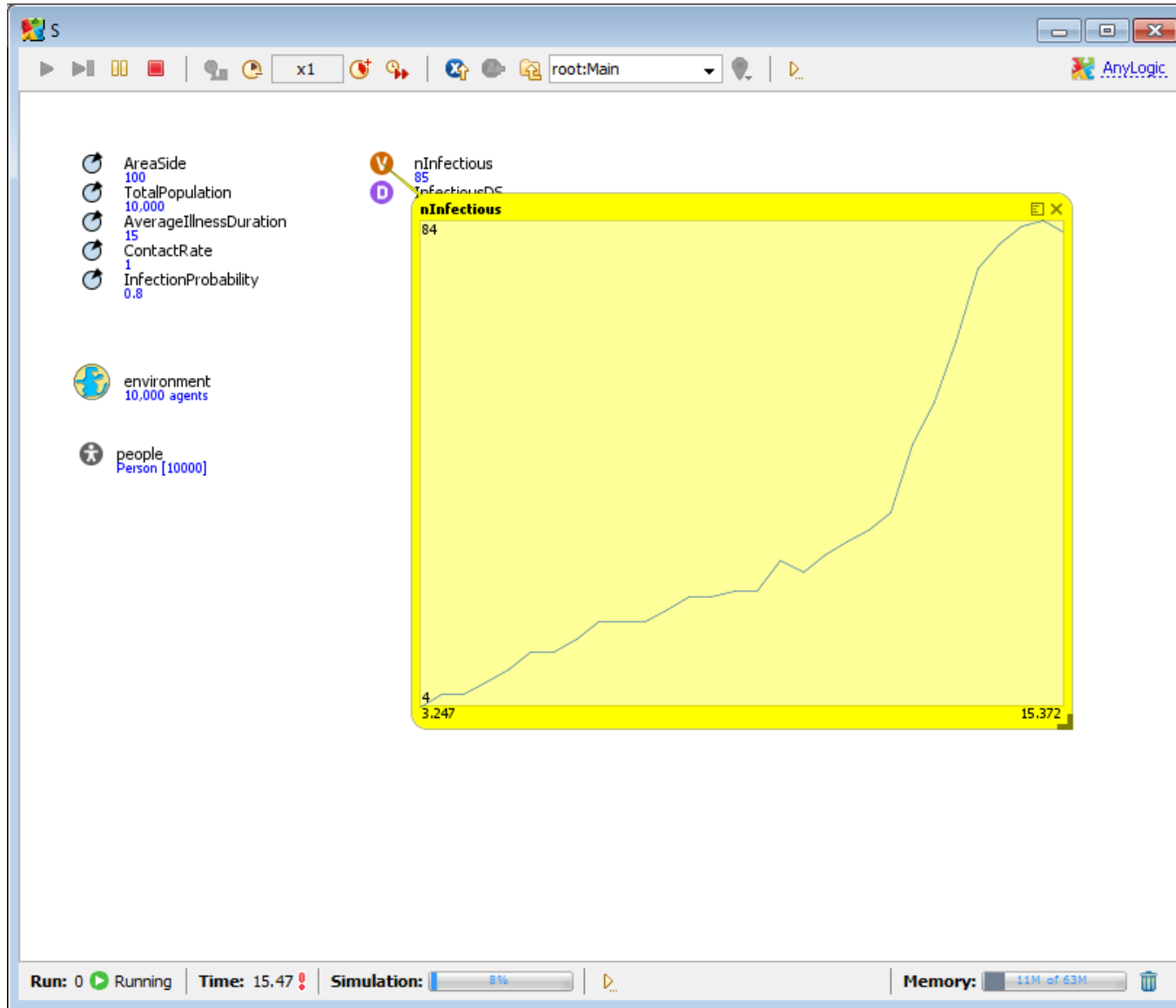
people  
Person [10000]

nInfectious  
716

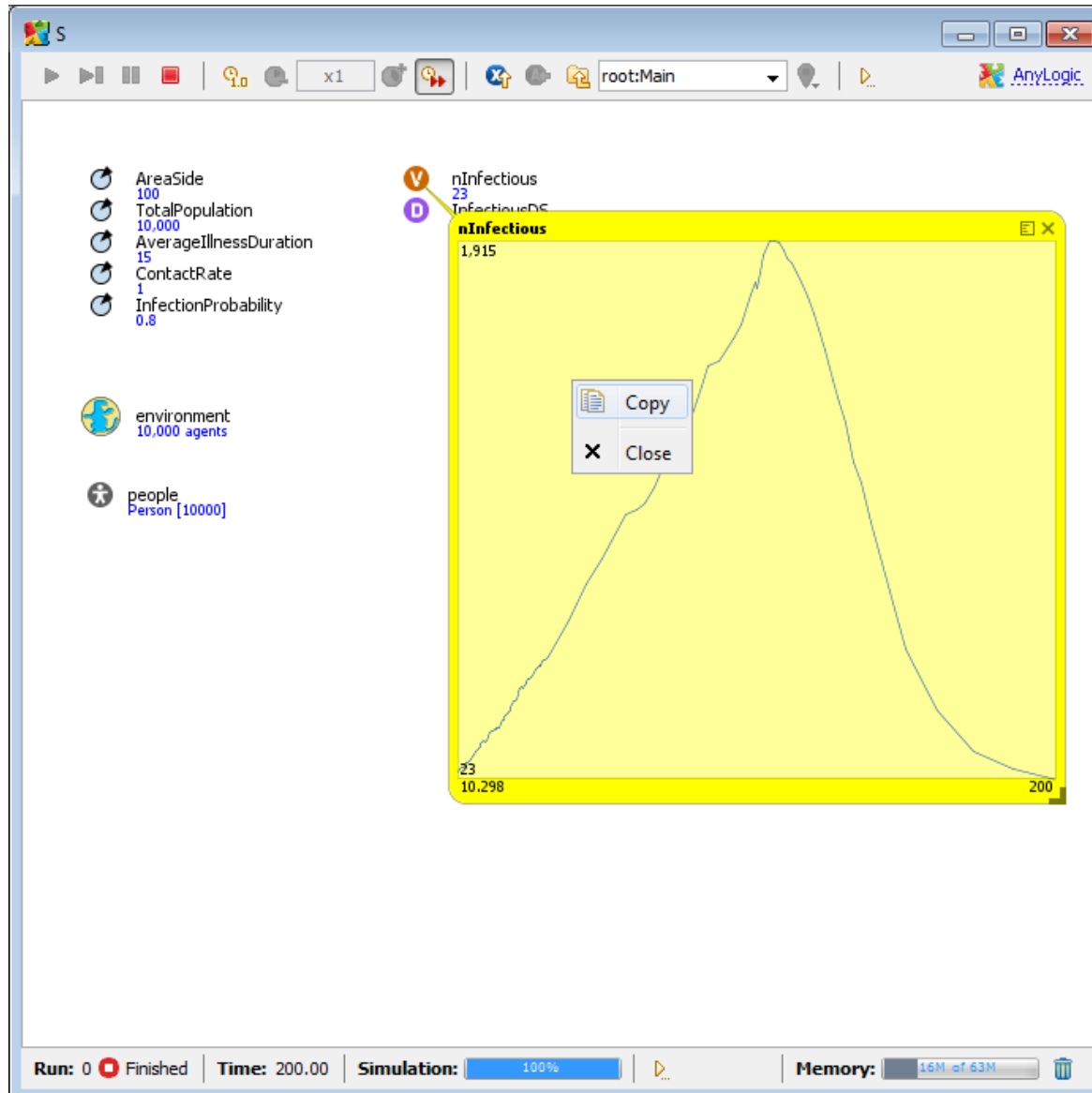
nInfer [X] [0, 715]  
716

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

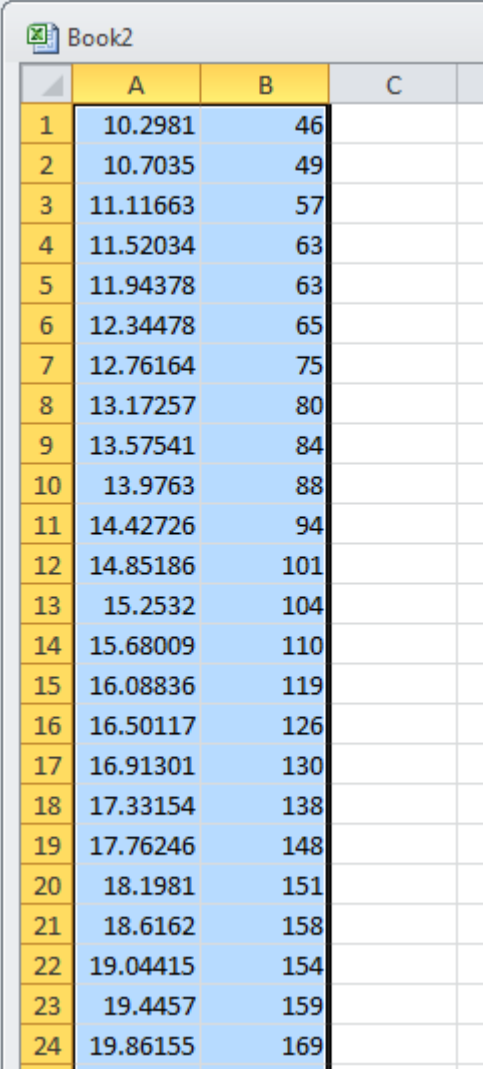
# Graph of Variable



# Right-Click to Copy the Numeric Data



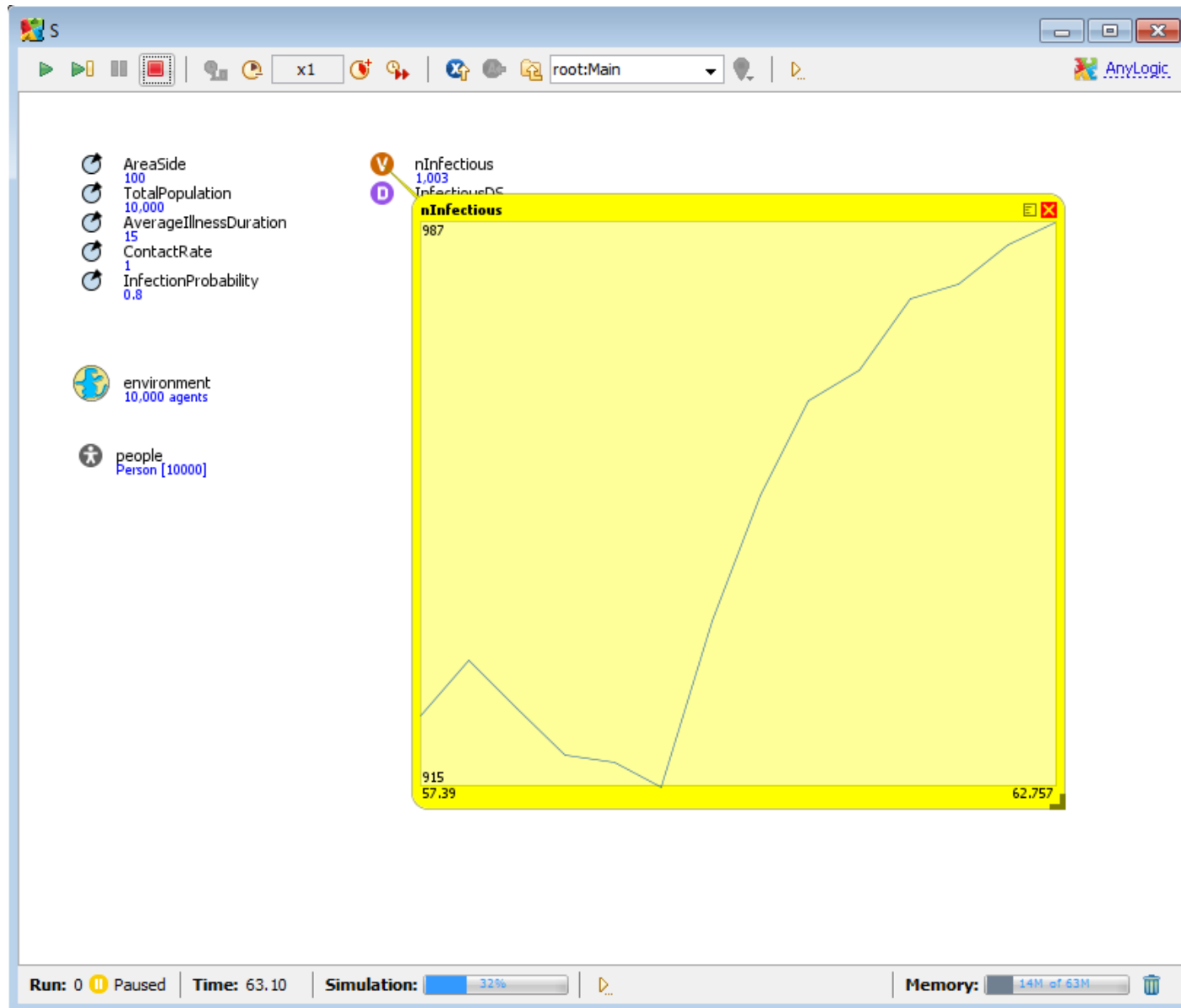
# Pasting Into Excel



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' category in the right-hand palette is selected, and the 'Statistics' option is highlighted. The 'Properties' window 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



# 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, nInfectious, and InfectiousDS. The 'people' object is highlighted in the workspace. The Properties console for 'people - Person' is open, showing a tabbed interface with 'General', 'Parameters', 'Statistics', and 'Description' tabs. The 'Statistics' tab is active, and a green 'Add statistics' button is visible. The Palette on the right side of the interface is open, showing 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. The Project Tree on the left shows the hierarchy of the project, including 'My SIR Agent Based Calibration Extensi', 'Main', 'Person', 'Statecharts', and 'SimpleExperiment: Main'.

# Fill in the “Condition” (Predicate) on Person

The screenshot displays the AnyLogic software interface for a simulation project titled "My SIR Agent Based Calibration Extensi". The main workspace shows a statechart for the "Person" agent with states: Susceptible, Infectious, Recovered, and Contact. Parameters include AreaSide, TotalPopulation, AverageIllnessDuration, ContactRate, and InfectionProbability. A "people [...]" agent is also visible.

The "people - Person" statistics configuration window is open, showing the following details:

- Name: peopleStat
- Type:  Count  Sum
- Expression: (empty)
- Condition: item.st

A tooltip for the "Condition" field lists several methods:

- stopSimulation() : boolean - ActiveObject
- stateContainsState(short compstate, short simpstate) : boole
- statechart : Statechart - Person
- start() : void - Person
- step(double height, double stepTime) : double - Utilities

# 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 objects: 'AreaSide', 'TotalPopulation', 'AveragellnessDuration', 'ContactRate', 'InfectionProbability', 'environment', and 'people [...]'. The 'people' object is selected, and its properties are shown in the bottom panel. The 'Statistics' section is active, showing a 'peopleStat' object with a 'Count' type and the following expression and condition:

Expression:

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

Below the screenshot, the full expression is written in red text:

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 pink oval, indicated by a red arrow. Below the workspace, the 'Properties' window for the 'person - Person' state is open, showing the configuration for a new statistic. The statistic is named 'CountSusceptible', has a type of 'Count', and its condition is 'item.TBProgressionStatechart.isStateActive(Person.TBSusceptible);'. A blue oval highlights this configuration area, with a blue arrow pointing to it from the text 'What statistics we wish to calculate'.

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 [..]

What statistics we wish to calculate

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

Description	Location
-------------	----------

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...

Action

Analysis

Presentati...

Connectiv...

Enterpris...

More Libraries...



# Name the Statistic “countSusceptible”

The screenshot displays the AnyLogic University software interface. The main workspace shows a grid with several variables and agents. The 'people' agent is selected, and its properties are visible in the bottom panel. The 'Statistics' section is active, showing the configuration for a new statistic named 'countSusceptible'.

**Statistics Configuration:**

- Name: countSusceptible
- Type:  Count  Sum  Average  Min  Max
- Expression: (empty)
- Condition: `item.statechart.isStateActive(Person.Susceptible)`

**Agent Properties (people - Person):**

- General
- Parameters
- Statistics
- Description

**Workspace Variables:**

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

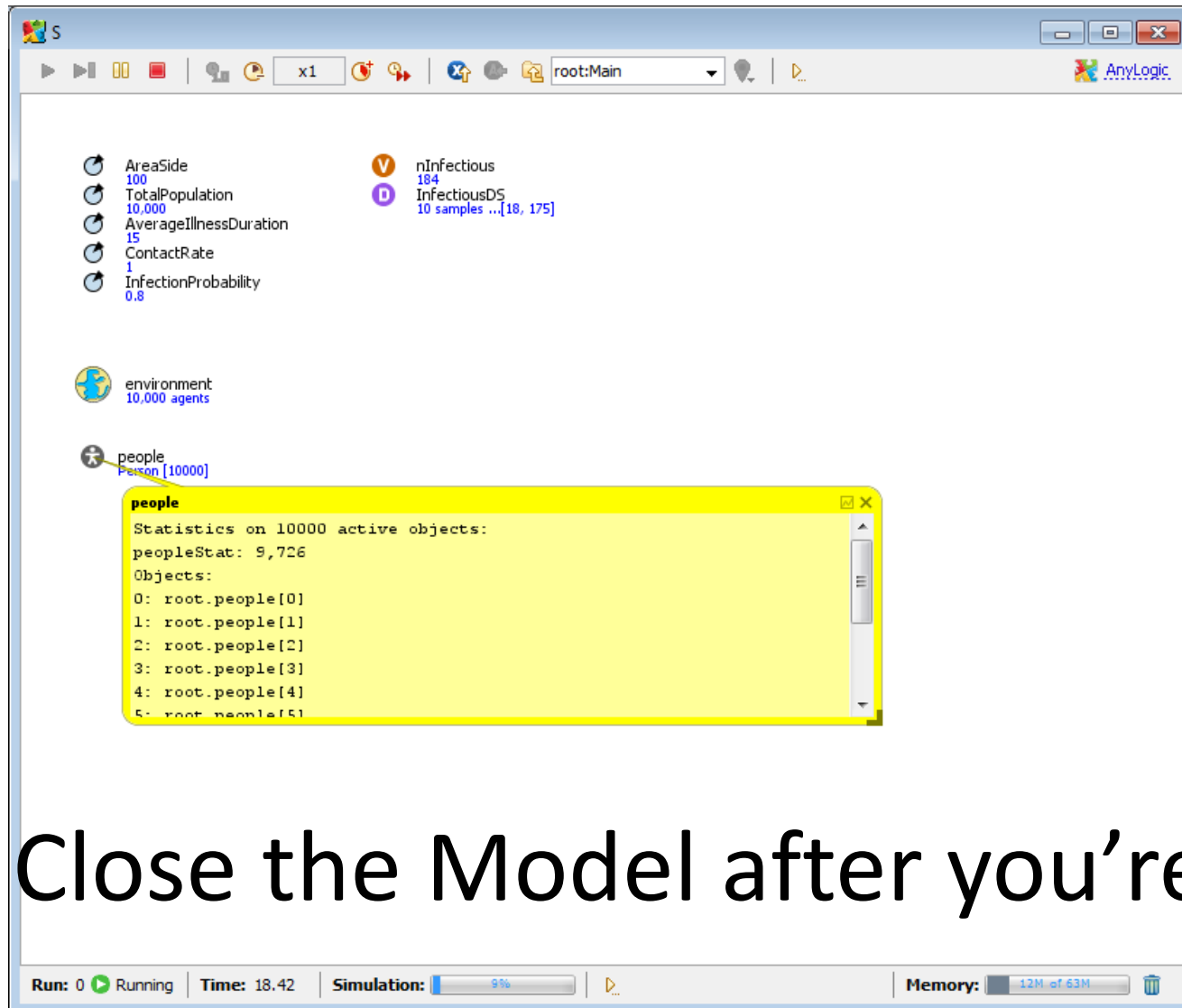
**Project Hierarchy (Left Panel):**

- 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

**Analysis Palette (Right Panel):**

- 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...

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



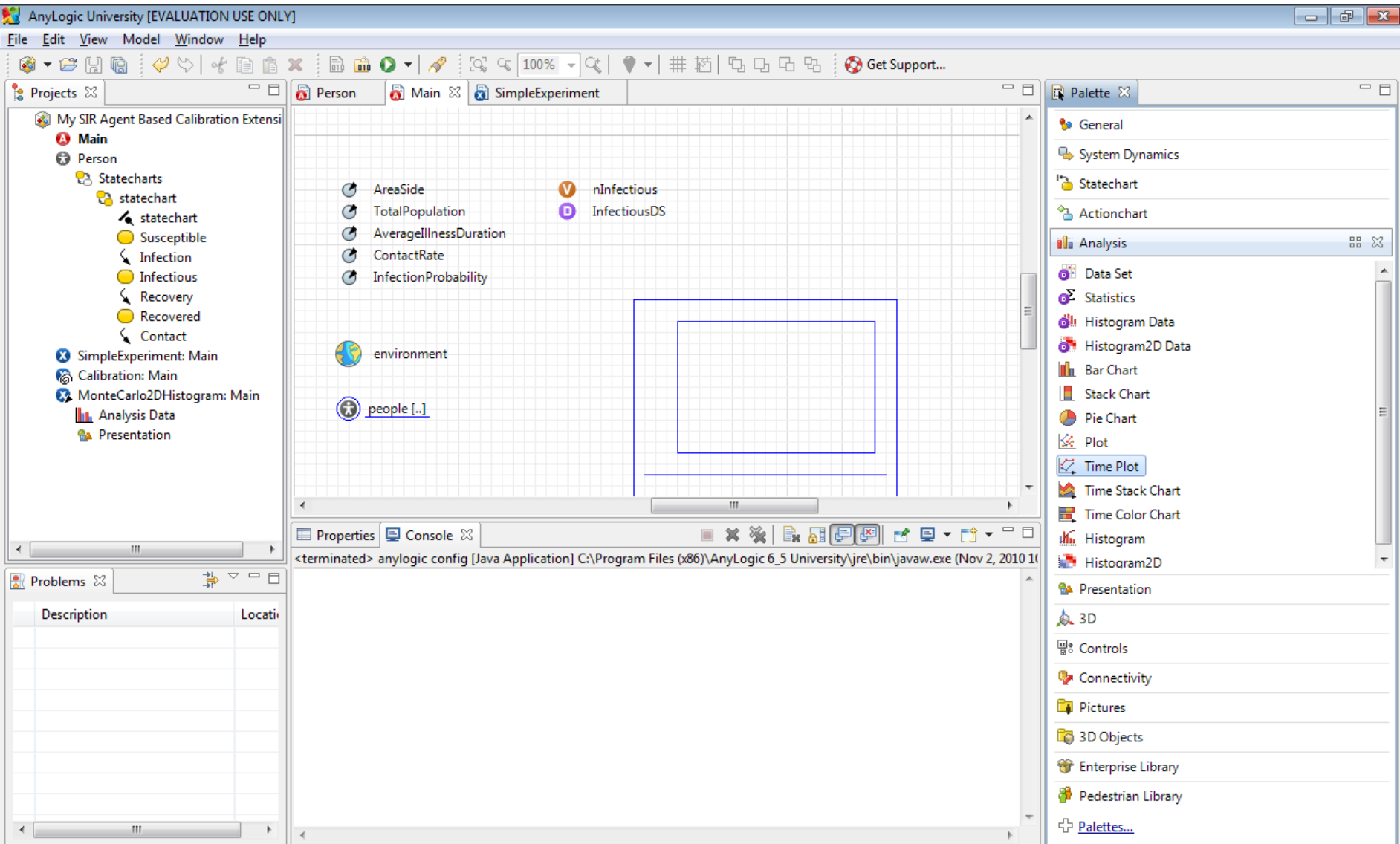
The screenshot shows the AnyLogic software interface. The main window displays a simulation environment with several objects and their statistics. The 'people' object is highlighted in yellow, and a yellow box is drawn around its statistics window. The statistics window shows the following information:

```
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 bottom status bar 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 with a time plot chart. The chart shows two data series: 'nInfectious' (represented by a blue line with a 'V' marker) and 'InfectiousDS' (represented by a purple line with a 'D' marker). Both series start at 0, rise sharply to 1 around time 5, and remain constant at 1 until time 100. The x-axis is labeled from 0 to 100, and the y-axis is labeled from -1 to 1. The chart is titled 'plot - Time Plot'.

The interface includes a menu bar (File, Edit, View, Model, Window, Help), a toolbar, and several panels:

- 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

- Properties:** plot - Time Plot
- General:** Name: plot,  Show name,  Ignore,  Public
- Advanced:** + Add data item
- Dynamic:** Time Window: 100
- Appearance:** Vertical scale: Auto, From: 0, to: 1
- Description:**  Do not update automatically,  Update automatically, Recurrence time: 1
- 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...

# 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 [...]. Above the plot, two data items are listed: nInfectious (marked with a 'V' icon) and InfectiousDS (marked with a 'D' icon).

The bottom panel, titled "plot - Time Plot", contains the following configuration options:

- General:** Name: plot,  Show name,  Ignore,  Public
- Dynamic:**
- Appearance:** Time Window: 100
- Description:** 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, 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' statechart containing states like 'Susceptible', 'Infection', 'Infectious', 'Recovery', 'Recovered', and 'Contact'. The main workspace shows a time plot with a green line representing the 'people' variable. The plot's y-axis ranges from 0.6 to 0.85. A tooltip is visible over the plot, listing various aggregation functions such as 'sum(String fieldName, String triggerFieldName) : double' and 'countSusceptible() : int - \_people\_Class'. At the bottom, the 'plot - Time Plot' properties window is open, showing the 'Value' set to 'people.', 'Color' set to 'oliveDrab', and 'Interpolation' set to 'Linear'. The 'Problems' window at the bottom left is empty.

# Choose “Count Susceptible”

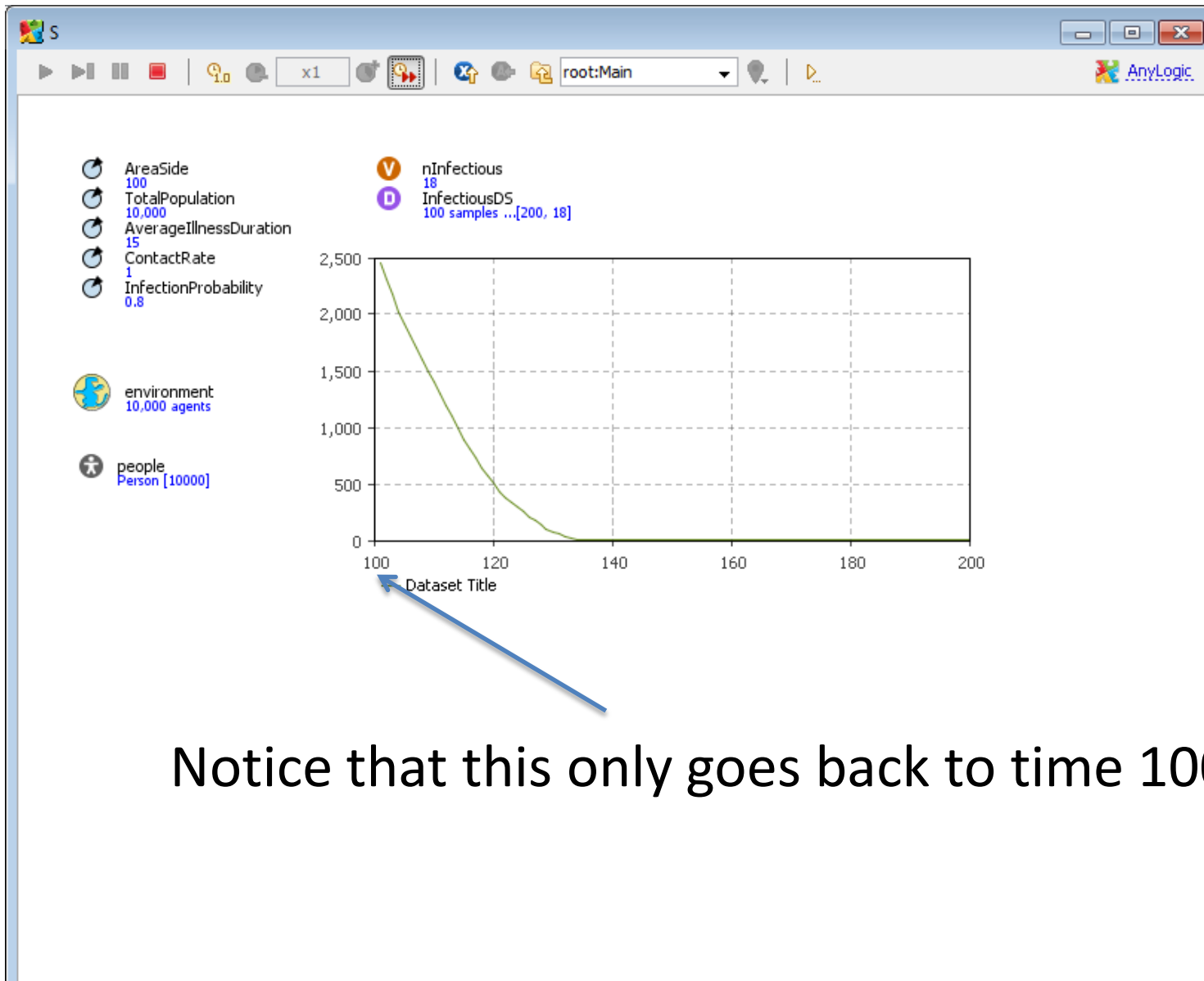
The screenshot displays the AnyLogic software interface. The main window shows a time plot with a green line representing the 'Count Susceptible' variable. The plot is titled 'plot - Time Plot' and is located in the 'Properties' pane. The 'Value' field is set to 'people.countSusceptible()'. The 'Color' 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	0.72
20	0.85
40	0.85
60	0.75
80	0.68
100	0.64

The interface also shows a 'Projects' pane on the left with a tree view of the model structure, including 'My SIR Agent Based Calibration Extensi', 'Main', 'Person', 'Statecharts', and 'SimpleExperiment: Main'. The 'Properties' pane at the bottom right shows the configuration for the 'plot' object, including 'Name', 'Show name', 'Ignore', 'Public', 'Value', 'Point style', 'Color', 'Draw line', 'Line width', 'Interpolation', and 'Add data item'.

# 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

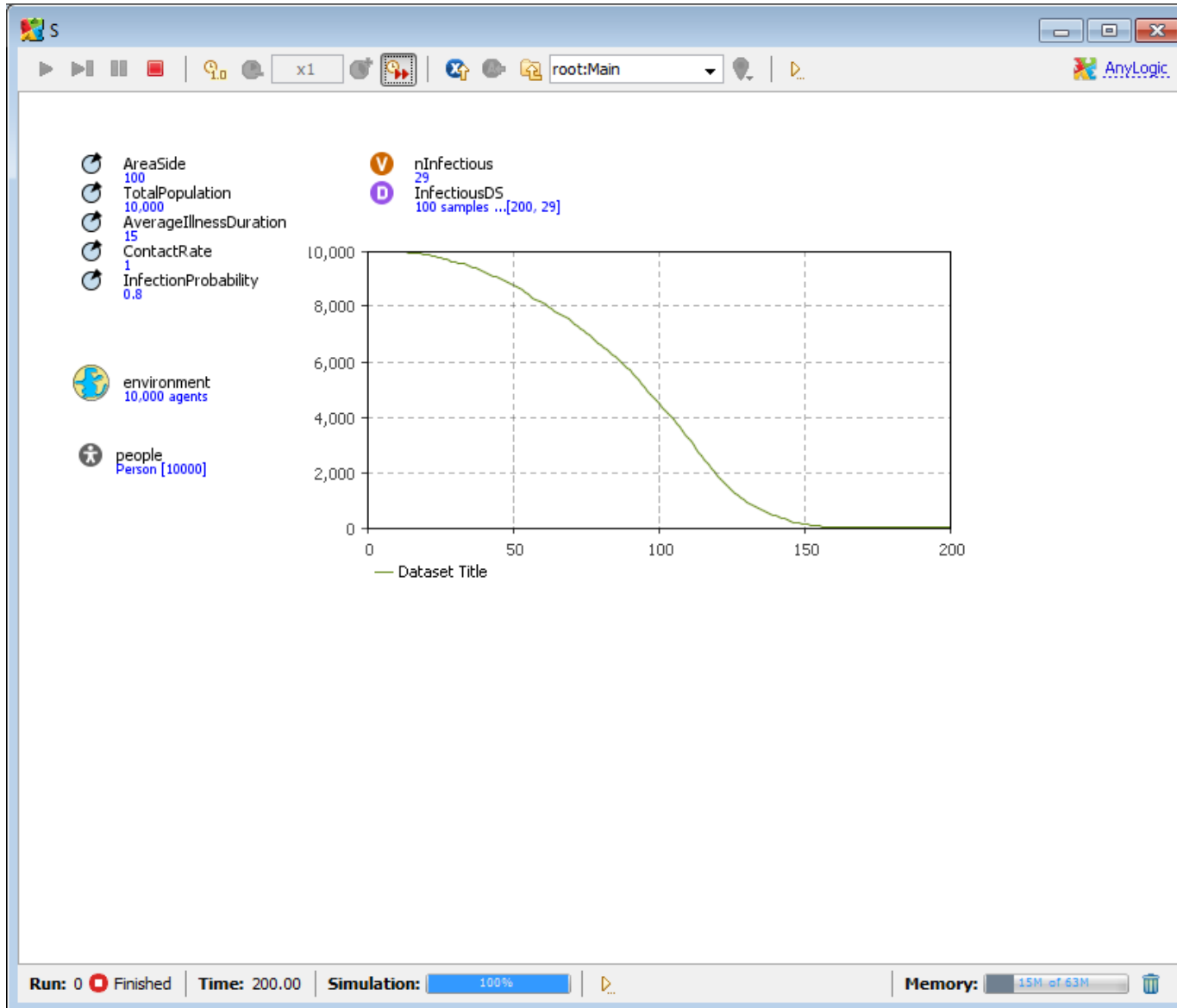
The screenshot displays the AnyLogic University interface. The main window shows a time plot with the following data series:

- nInfectious** (green line): Starts at approximately 0.85, decreases to about 0.7 at time 100, and then rises back to 0.85.
- InfectiousDS** (purple line): Starts at approximately 0.7, drops to about 0.65 at time 100, and then rises back to 0.7.

The plot is titled "plot - Time Plot" and has the following settings:

- General:** Draw line checked, Line width: 1 pt, Interpolation: Linear.
- Advanced:** Add data item button.
- Dynamic:** Time Window: 200.
- Appearance:** Vertical scale: Auto, From: 0, to: 1.
- Description:** Do not update automatically (unchecked), Update automatically (checked), 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 toString method

# Output: Datasets

The screenshot displays the AnyLogic Advanced [EDUCATIONAL USE ONLY] interface. The top toolbar includes icons for file operations, a search bar, and a 'Get Support' link. The main workspace shows a diagram with nodes for 'environment', 'person [...]', and 'dsSusceptibleCount'.

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

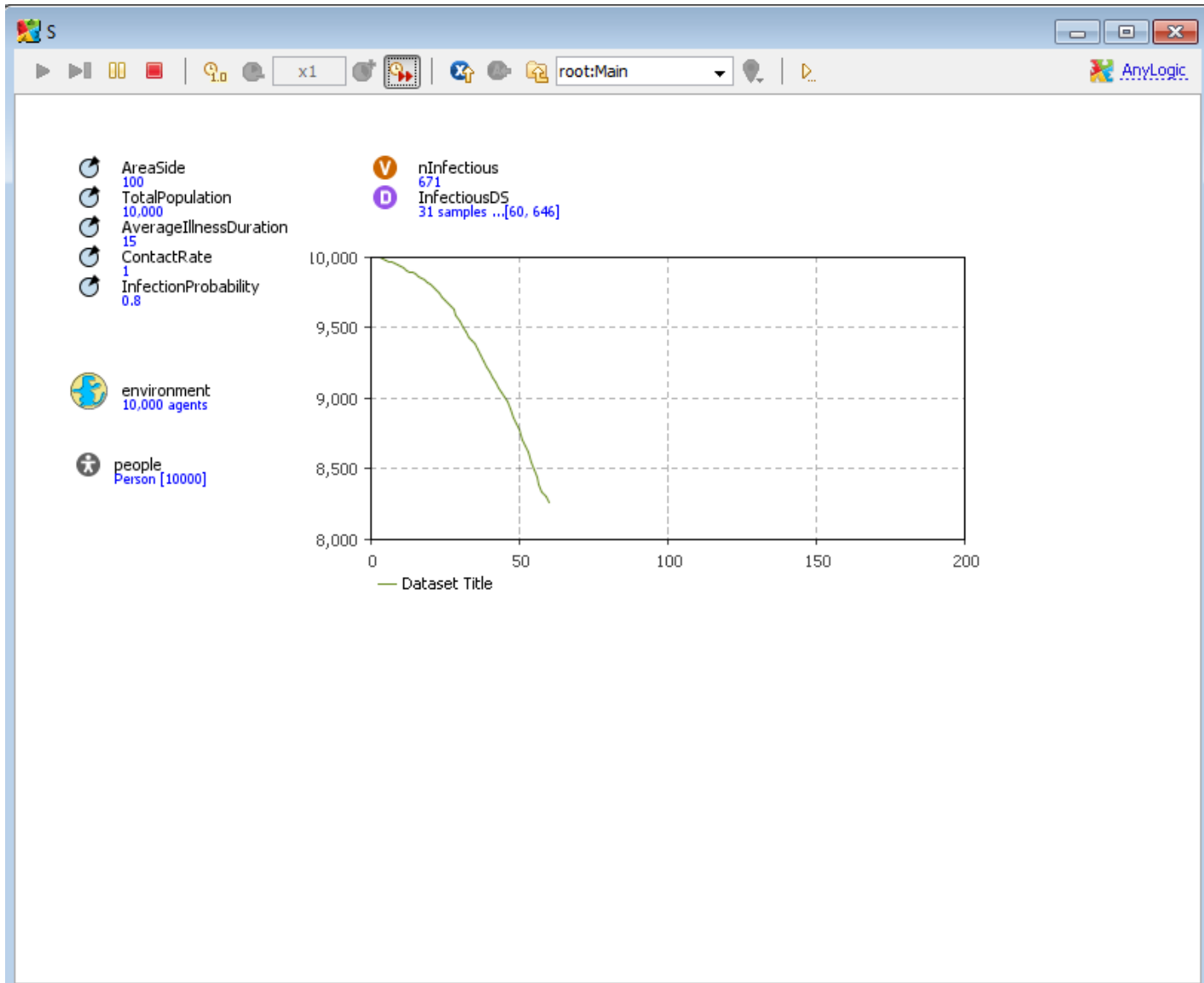
The bottom panel shows the 'dsSusceptibleCount - Data Set' configuration:

- 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 features a 'Problems' window with a table:

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:

- 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 is highlighted in yellow and 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 bottom status bar shows: 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 agents:

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

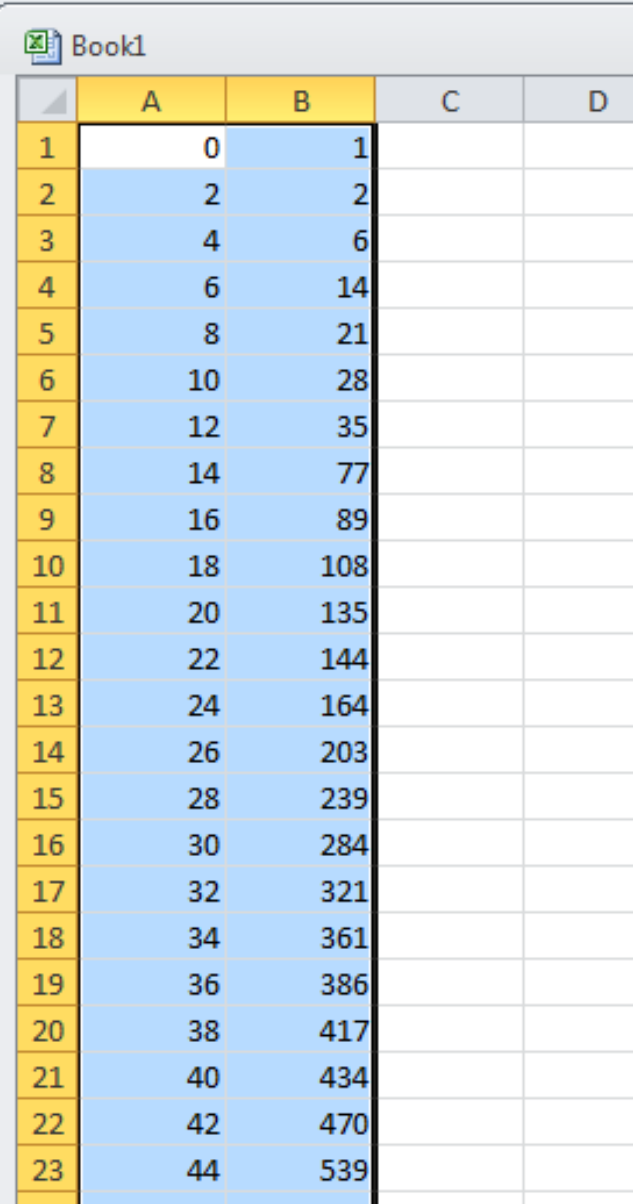
In the center, a data table titled "InfectiousDS" is displayed. The table has two columns: the first column represents time steps from 0 to 42, and the second column represents the number of infectious individuals. A context menu is open over the table, showing "Copy" and "Close" options.

Time	Infectious
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 contains a table with four columns labeled A, B, C, and D. The data is as follows:

	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  $((double) nInfectious) / ((double) TotalPopulation)$ . The Properties panel for this dataset is open, showing the following configuration:

- Name:** datasetAbsolutePrevalence
- Show Name
- Ignore
- Public
- Show As
- Use time as horizontal axis value
- Horizontal axis value:** [Empty field]
- 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
- 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 bottom-left 'Problems' panel lists several errors, including 'The constructor DataSet() is undefined' and 'Engine.log cannot be resolved'.

# 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, 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.

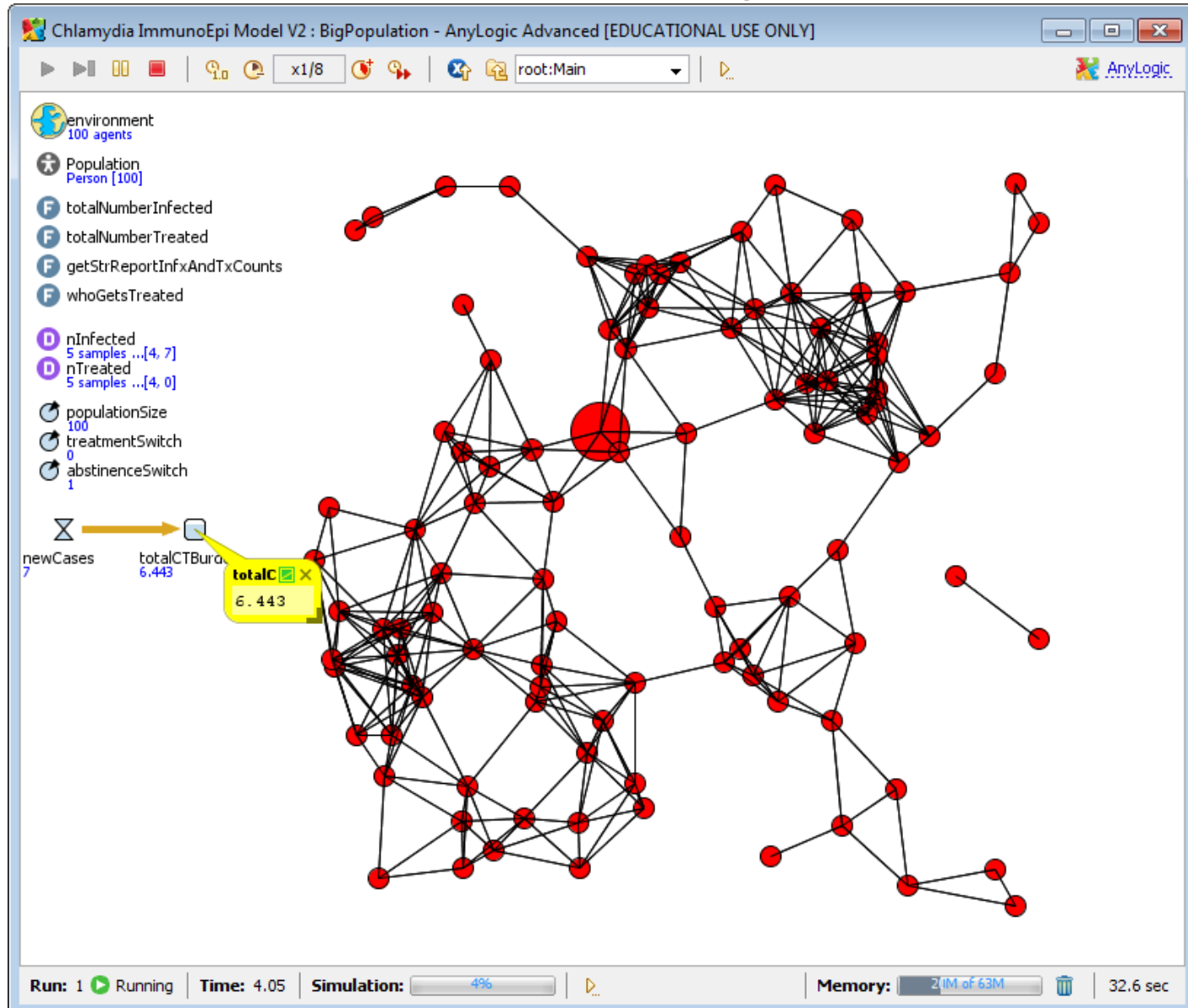
The configuration panel for the `TimePlotAgentCount - Time Plot` is visible below the plot. It includes the following settings:

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

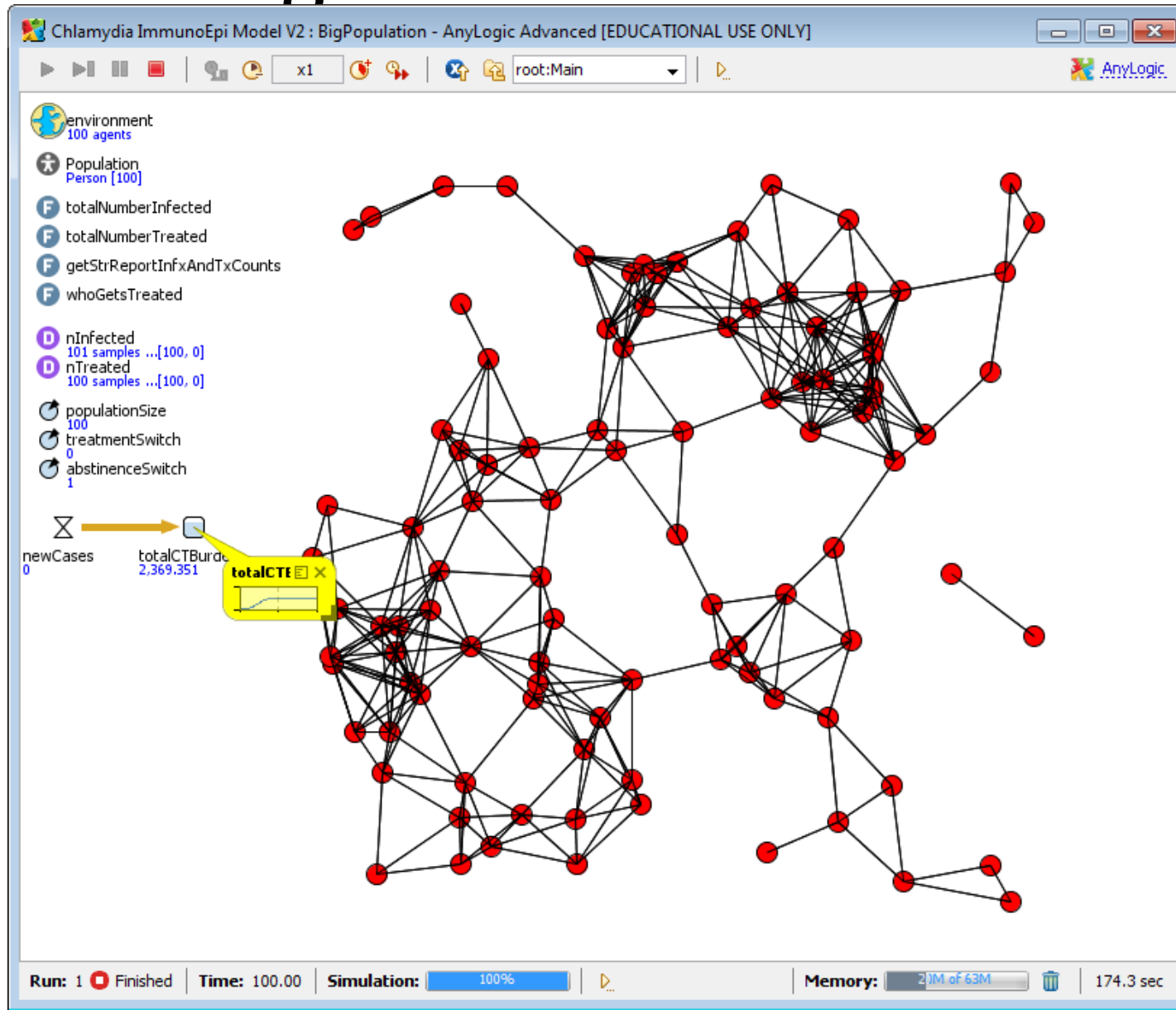
The left sidebar shows the project structure for `TBv1*`, including `Main` (Parameters, Functions, Environments, Embedded Objects, Analysis Data, Presentation) and `Person` (Parameters). The bottom-left pane shows a table with columns for `Description` and `Location`.

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]

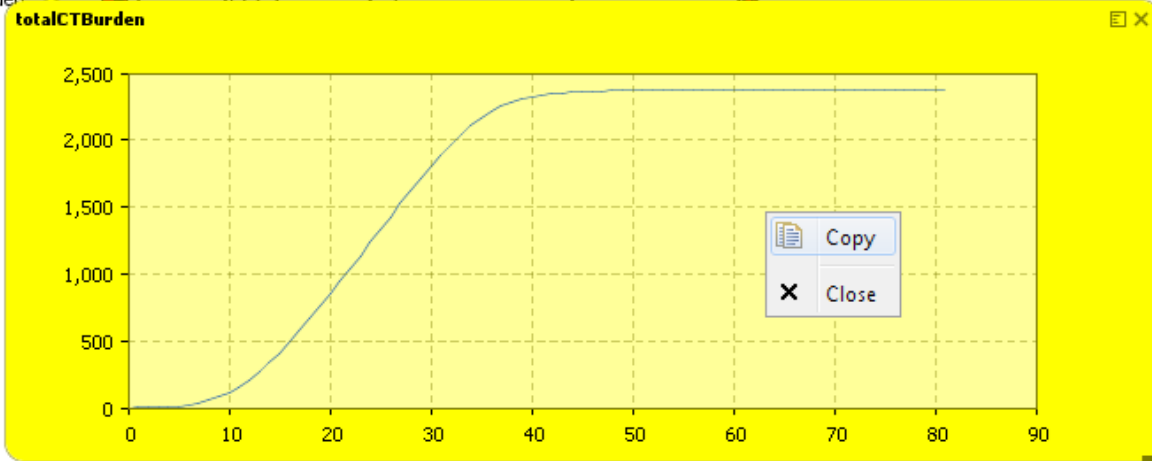
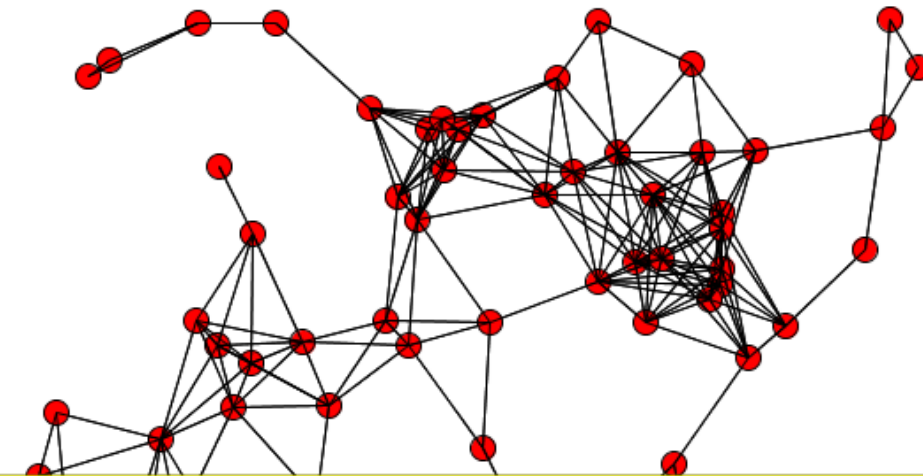
populationSize  
100

treatmentSwitch  
0

abstinenceSwitch  
1

newCases  
0

totalCTBurden  
2,369.351



totalCTBurden

2,500

2,000

1,500

1,000

500

0

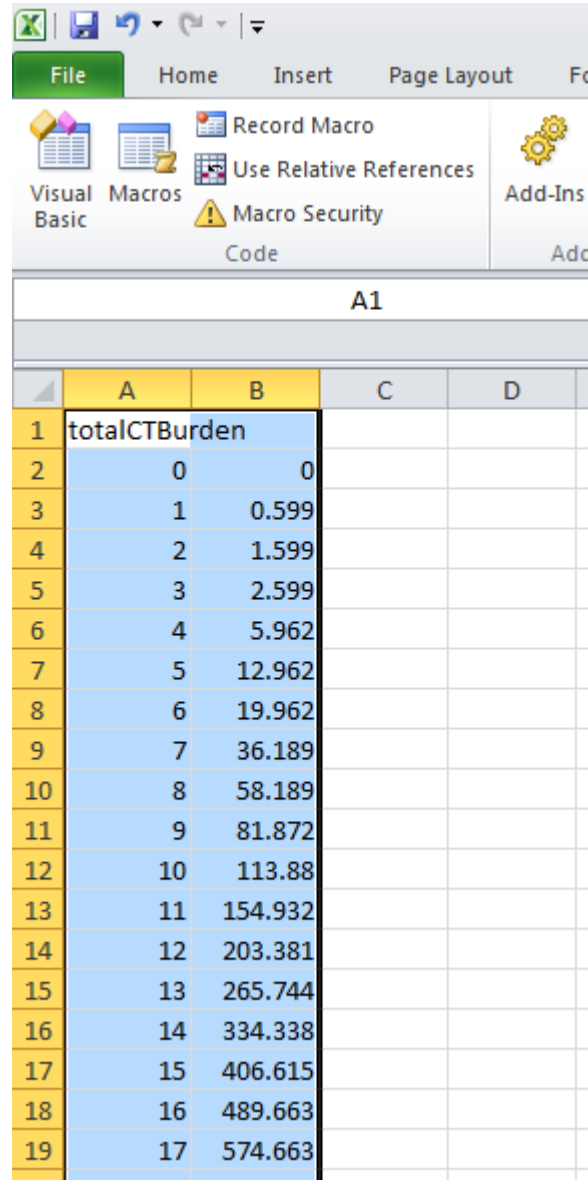
0 10 20 30 40 50 60 70 80 90

Copy

Close

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

# Data Exported from Ad-Hoc Chart



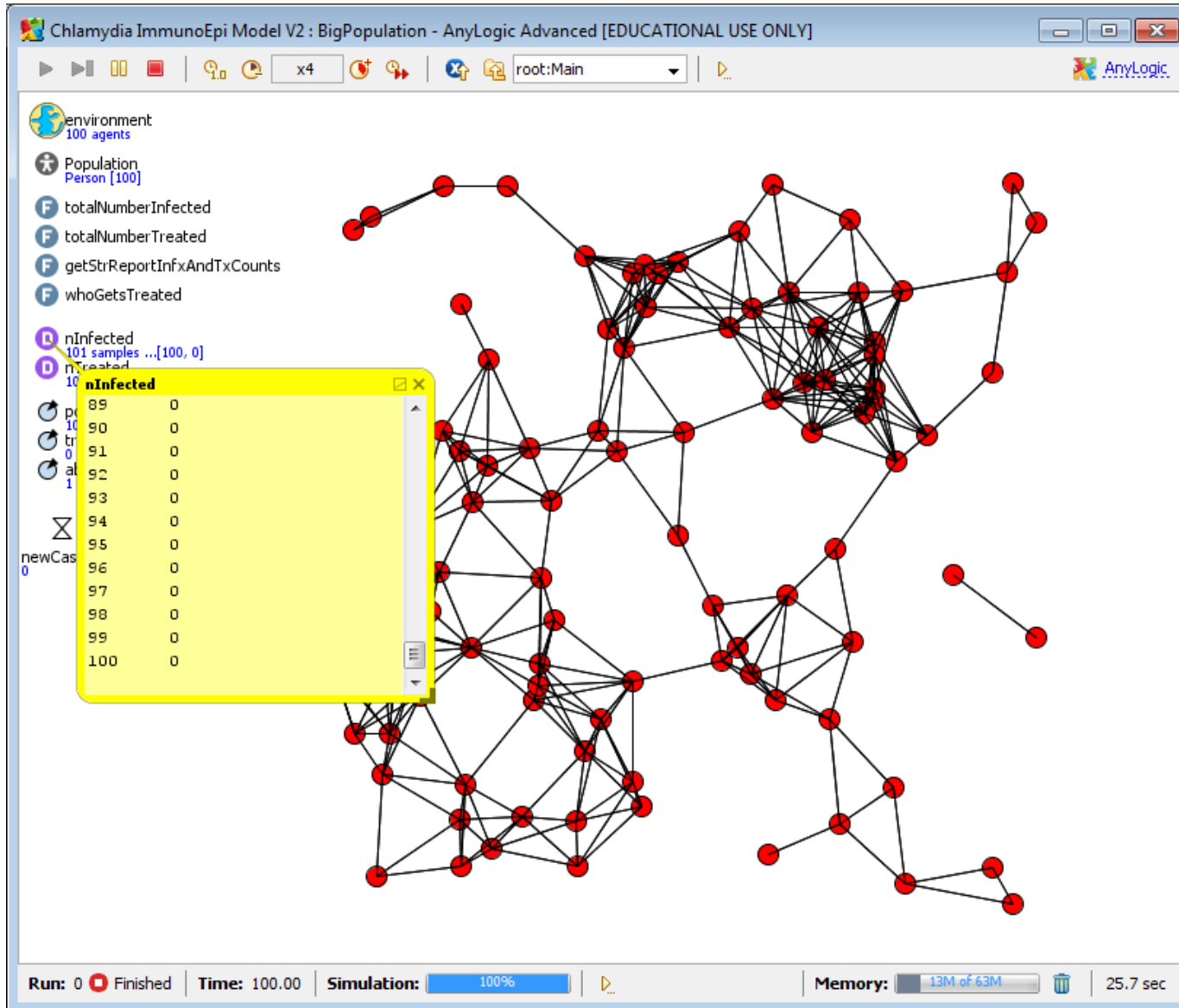
	A	B	C	D
1	totalCTBurden			
2	0	0		
3	1	0.599		
4	2	1.599		
5	3	2.599		
6	4	5.962		
7	5	12.962		
8	6	19.962		
9	7	36.189		
10	8	58.189		
11	9	81.872		
12	10	113.88		
13	11	154.932		
14	12	203.381		
15	13	265.744		
16	14	334.338		
17	15	406.615		
18	16	489.663		
19	17	574.663		

# 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 Advanced interface for the 'Chlamydia ImmunoEpi Model V2 : BigPopulation'. 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]

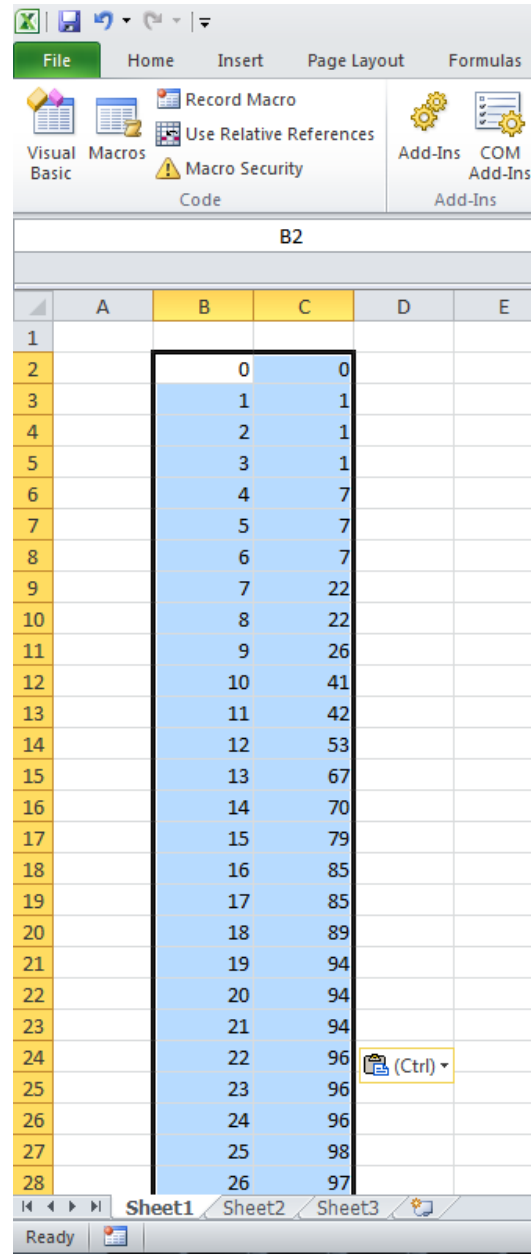
**Variables:** totalNumberInfected, totalNumberTreated, getStrReportInfxAndTxCounts, whoGetsTreated

**nInfected:** 101 samples ...[100, 0]

Year	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

**Simulation Status:** 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



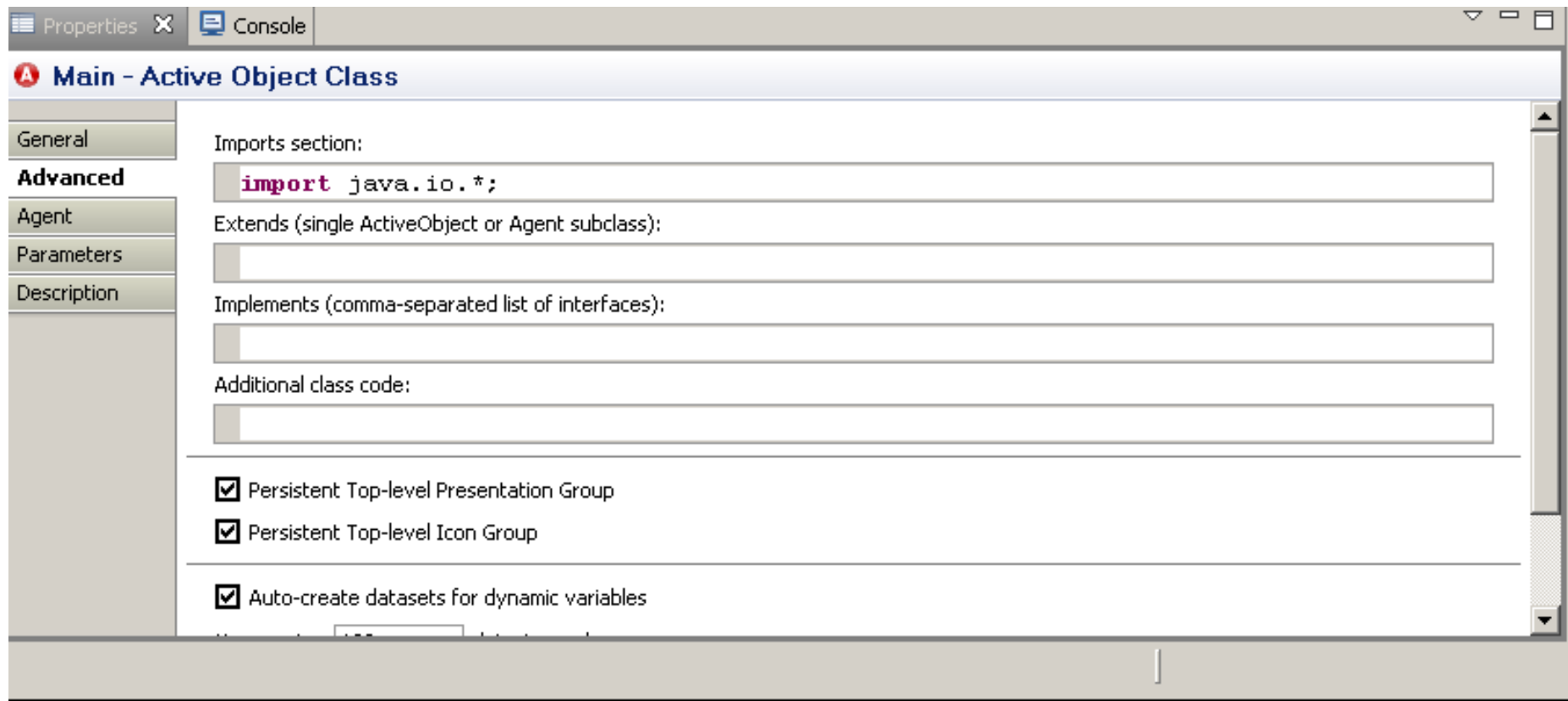
# Data Output to File

- Pros
  - Simple to perform
  - Relatively easy to import into e.g. Excel, R, etc.
  - Files can be readily archived
- Cons
  - Awkward to 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
    traceLn("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 displays the AnyLogic Advanced software interface. The main workspace shows a diagram of the 'Main' class with various components like 'Population [...]', 'environment', and 'datasetInfective'. The 'Main - Active Object Class' properties window is open, showing the 'Destroy Code' section with the following code:

```
try
{
    FileOutputStream fos = new FileOutputStream("Filename.tab");
    PrintStream p = new PrintStream(fos);
    p.println(datasetInfective.toString()); // outputs tab delimit
}
catch (Exception e)
```

The 'Problems' window at the bottom left is empty, and the status bar at the bottom indicates 'Build completed successfully. Time: 0.941 s.'

# Where to Put the Code to Output the Dataset

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

The screenshot displays the AnyLogic Advanced software interface. The main workspace shows a model diagram with various components and their connections. A specific event, 'writeToFile', is highlighted in the diagram. Below the diagram, the 'Properties' window is open, showing the configuration for the 'writeToFile - Event'. The event is set to trigger at a specific time (September 1, 2011, 4:12:49 PM) and occurs once. The action associated with this event is a Java code snippet that writes the dataset to a file.

**Project Structure:**

- PodSchedule
- SimplePersonEntity
- TestPodSchedule
- TriggerDebugger
- EclipseDebuggingSimulation: Main
- Simulation: Main
- UnitTest: Main
- ABMModelWithBirthDeath\*
- Main
  - Parameters
  - Functions
  - Events
  - Environments
  - Embedded Objects
  - Analysis Data
  - datasetInfective
  - Presentation

**Model Diagram Components:**

- Population [...]
- environment
- offspringDistanceFromMother
- initialPrevalenceOfInfection
- immigrantsPerYear
- MeanLifespan
- selectPeopleMatchingSpecification
- performActionOnPeopleMatchingPredicate
- countPeopleMatchingMultipleSpecification
- datasetInfective
- ImmigrantArrival
- writeToFile
- prevalenceOfInfectionAmongImmigrants

**writeToFile - Event Properties:**

- Name: writeToFile
- Show Name:
- Ignore:
- Public:
- Show At Runtime:
- Trigger Type: Timeout
- Mode: Occurs once
- Occurrence time (absolute):  0  September 1, 2011 4:12:49 PM

**Action Code:**

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

# 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
  - 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

# 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)

The screenshot shows an IDE's Properties window for a project named "RecreationOfAzizaModelCubeDataStorage\_V4NoA...CGLTBI\_NDOModificationsForTestingV2 - Model". The "Dependencies" tab is selected, showing a list of dependencies required to build the model. The dependencies are organized into two sections: "AnyLogic libraries required to build the model:" and "Jar files and class folders required to build the model:". The first section is empty, and the second section contains one entry: "mysql-connector-java-5.1.13-bin.jar".

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	

# 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

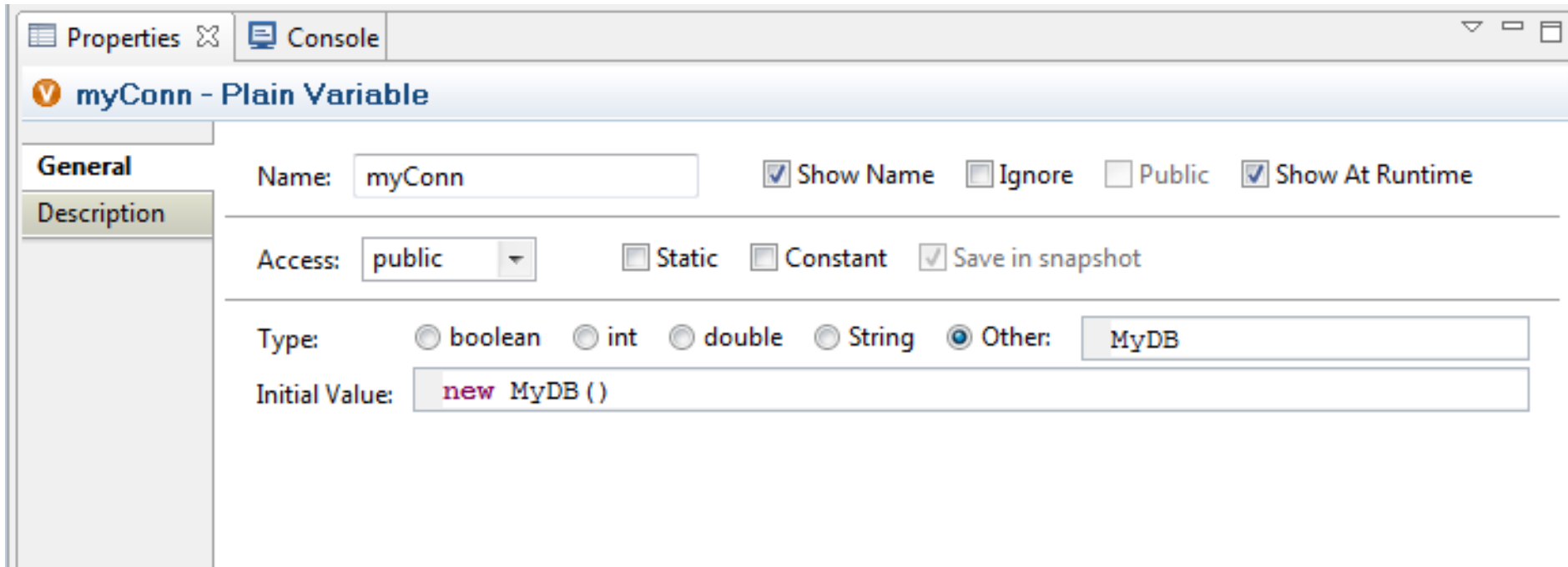
A Method is associated with each of  
Execute  
Query  
Insert

```
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



The screenshot shows the 'Properties' window for a variable named 'myConn'. The window has tabs for 'Properties' and 'Console'. The variable is identified as a 'Plain Variable'. The 'General' tab is selected, showing the following configuration:

- 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



# 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