Dealing with Data Gradients: “Backing Out” & Calibration

Nathaniel Osgood

Using Modeling to Prepare for Changing Healthcare Needs

Duke-NUS

April 16, 2014

ABM Modeling Process Overview

A Key Deliverable!

ODD: Overview & high-level design components

ODD: Design components & details

Mental Model

Problem Conceptualization | Qualitative Problem Mapping | Model Formulation | Model Calibration | Model Testing | Policy Evaluation | Knowledge Translation

Problem Conceptualization

Problem/research question articulation

Patterns for explanation

Model scope/boundary selection (endogeous, exogenous,ignored)

Key entities & their relationships
- Agents (&collectives)
- Environment
- Nesting hierarchy or network diagrams

Output of interest

Qualitative Problem Mapping

State charts
Parameter & state variables
Qualitative State charts
Influence & Causal loop diagrams
Multi-agent interaction diagrams
Process flow structure
Key events
Rough model time & spatial extent
Observers to measure outputs

Model Formulation

Specification of
- Parameters
- Quantitative causal relations
- Decision/behavior rules
  - Transitions
- Interactions
  - Messaging &handlers
  - Resources
- Relationship dynamics
- Mobility dynamics
- Initial conditions

Model Calibration

Reference mode reproduction

Matching of intermediate time series

Matching of observed data points

Constrain to sensible bounds

Structural sensitivity analysis

Model Testing

Parameter sensitivity analysis

Cross-validation
- Formal
- w/Discovered
  - Patterns

Robustness tests

Extreme value tests

Unit checking

Problem domain tests

Policy Evaluation

Specification & investigation of intervention scenarios

Investigation of hypothetical external conditions

Cross-scenario comparisons (e.g. CEA)

Knowledge Translation

Learning environments (e.g. DISimS)

Visualizations

Common Sources for Parameter Estimates

- Surveillance data
- Controlled trials
- Outbreak data
- Clinical reports data
- Intervention outcomes studies
- Calibration to historic data
- Expert judgement
- Meta-analyses

Material Adapted from External Source

Redacted from Public PDF for Copyright Reasons

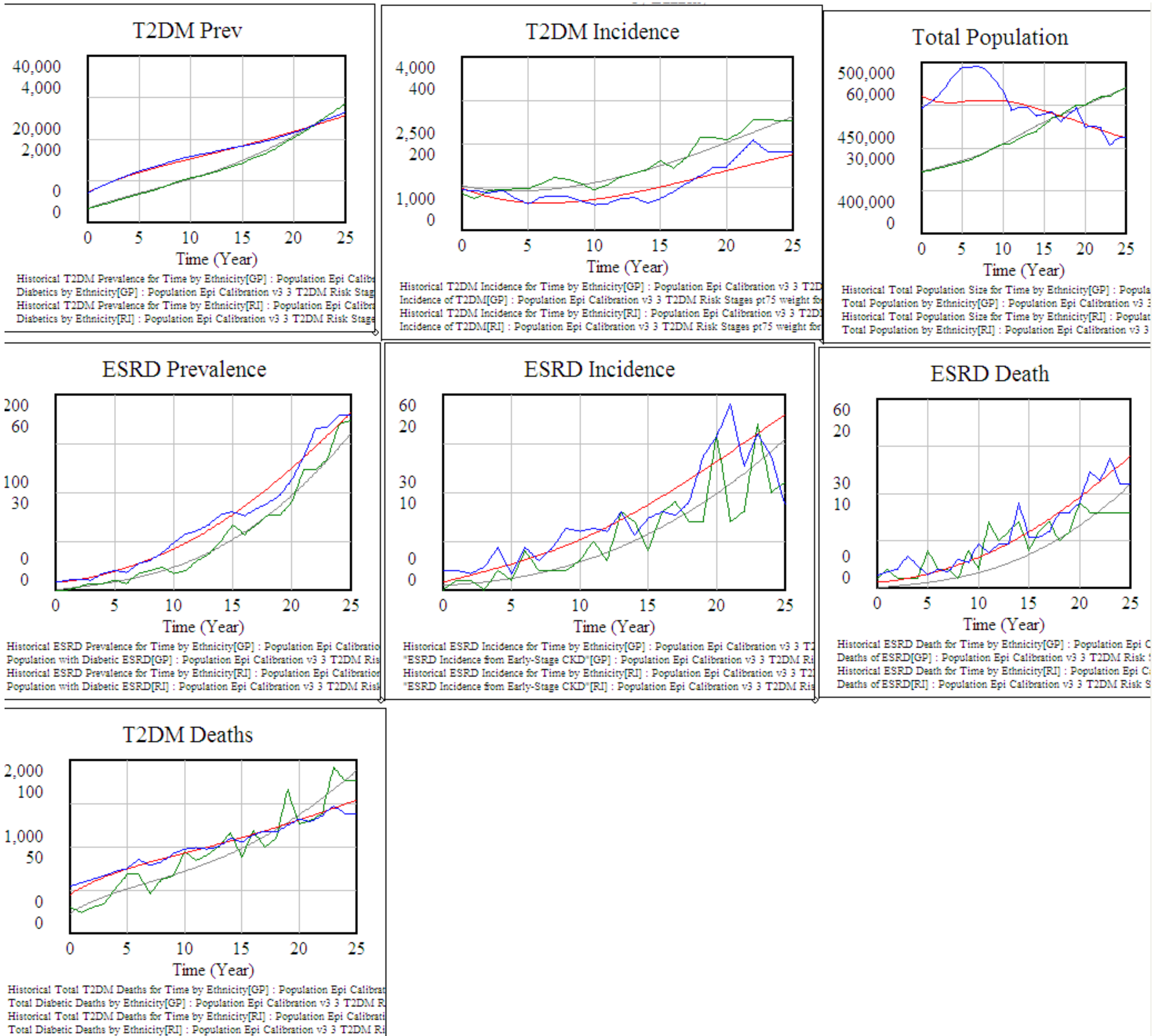
Sensitivity Analyses

- Same relative or absolute uncertainty in different parameters may have hugely different effect on outcomes or decisions
- Help identify parameters that strongly affect
  - Key model results
  - Choice between policies
- We place more emphasis in parameter estimation into parameters exhibiting high sensitivity

Dealing with Data Gradients

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

Recall: Single Model Matches Many Data Sources



one of

“Backing Out”

- Sometimes we can manually take several aggregate pieces of data, and use them to collectively figure out what more detailed data might be
- Frequently this process involves imposing some (sometimes quite strong) assumptions
  - Combining data from different epidemiological contexts (national data used for provincial study)
  - Equilibrium assumptions (e.g. assumes stock is in equilibrium – deriving prevalence from incidence)
  - Independence of factors (e.g. two different risk factors convey independent risks)

Example

- Suppose we seek to find out the sex-specific prevalence of diabetes in some population
- Suppose we know from published sources
  - The breakdown of the population by sex (c_M, c_F)
  - The population-wide prevalence of diabetes (p_T)
  - The prevalence rate ratio of diabetes in women when compared to men (rr_F)
- We can "back out" the sex-specific prevalence from these aggregate data (p_F, p_M)
- Here we can do this "backing out" without imposing assumptions

Backing Out

\# male diabetics + # female diabetics = # diabetics

$(p_M * c_M) \quad + \quad (p_F * c_F) \quad = p_T*(c_M+c_F)$

- Further, we know that $p_F / p_M = rr_F \Rightarrow p_F = p_M * rr_F$
- Thus

$(p_M * c_M) \quad + \quad ((p_M * rr_F) * c_F) \quad = p_T*(c_M+c_F)$

$p_M*(c_M + rr_F * c_F) = p_T*(c_M+c_F)$

- Thus
  - $p_M = p_T*(c_M+c_F) / (c_M + rr_F * c_F)$
  - $p_F = p_M * rr_F = rr_F * p_T*(c_M+c_F) / (c_M + rr_F * c_F)$

Disadvantages of “Backing Out”

- Backing out often involves questionable assumptions (independence, equilibrium, etc.)
- Sometimes a model is complex, with several related known pieces
  - Even thought we may know a lot of pieces of information, it would be extremely complex (or involve too many assumptions) to try to back out several pieces simultaneously

Another Example: Joint & Marginal Prevalence

	Rural	Urban	
Male	p_{MR}	p_{MU}	p_M
Female	p_{FR}	p_{MU}	p_F
	p_R	p_U	

Perhaps we know

- The count of people in each { Sex, Geographic } category
- Each marginal prevalence (p_R, p_U , p_M , p_F)

We need at least one more constraint (one possibility: assume $p_{MR} / p_{MU} = p_R / p_U$)
We can then derive the prevalence in each { Sex, Geographic } category

Calibration: “Triangulating” from Diverse Data Sources

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

Calibration

- Calibration helps us find a reasonable (specifics for) “dynamic hypothesis” that explains the observed data
  - Not necessarily the truth, but probably a reasonably good guess – at the least, a consistent guess
- Calibration helps us leverage the large amounts of diffuse information we may have at our disposal, but which cannot be used to directly parameterize the model
- Calibration helps us falsify models

Calibration: A Bit of the How

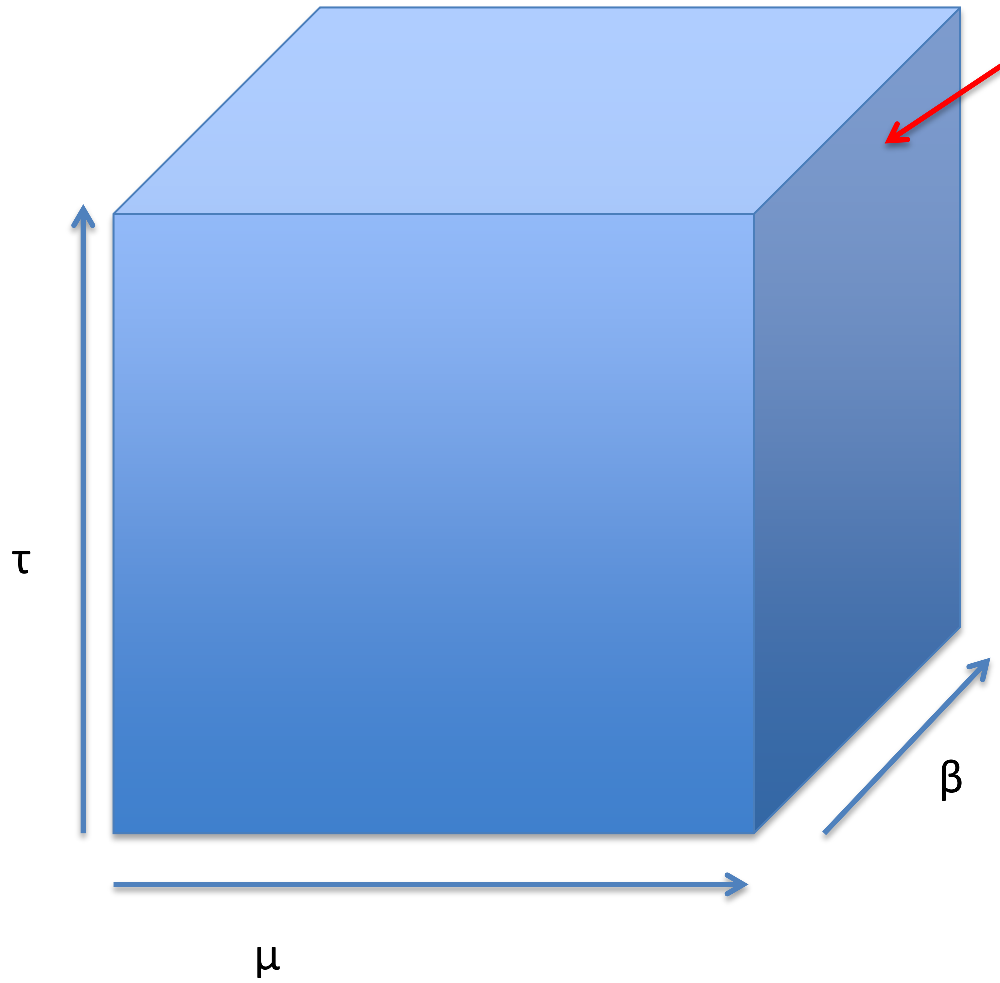
- Calibration uses a (global) optimization algorithm to try to adjust unknown parameters so that it automatically matches an arbitrarily large set of data
- The data (often in the form of time series) informs the objective function of the calibration
- The optimization algorithm will run the model many (thousands or more) times to find the "best" match for all of the data

Required Information for Calibration

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

Envisioning “Parameter Space”

For each point in this space, there will be a certain “goodness of fit” of the model to the collective data



Assessing Model "Goodness of Fit"

- To improve the "goodness of fit" of the model to observed data, we need to provide some way of quantifying it!
- Within the model, we
  - For each historic data, calculate discrepancy of model
    - Figure out absolute value of discrepancy from comparing
      - Historic Data
      - The model's calculations
    - Convert the above to a fractional value (dividing by historic data)
  - Sum up these discrepancy

Characteristics of a Desirable Discrepancy Metric

- **Dimensionless**: We wish to be able to add discrepancies together, regardless of the domain of origin of the data
- **Weighted**: Reflecting different pedigrees of data, we'd like to be able to weigh some matches more highly than others
- **Analytic**: We should be able to differentiate the function one or more times
- **Concave**: Two small discrepancies of size *a* should be considered more desirable than having one big discrepancy of size 2*a* for one, and no discrepancy at all for the other.
- **Symmetric**: Being off by a factor of two should have the same weight regardless of whether we are 2x or ½x
- **Non-negative**: No discrepancy should cancel out others!
- **Finite**: Finite inputs should yield finite discrepancies

A Good Discrepancy Function (Assuming non-negative h & m)

Exponent
>1 ⇒ concave with respect to h-m

Taking average in denominator (together w/squaring of result) ensures symmetry with respect to h&m

$$w \cdot \left(\frac{h-m}{average(h,m)} \right)^2 = w \cdot \left(\frac{h-m}{\left(\frac{h+m}{2}\right)} \right)^2$$

Division ⇒ Dimensionless
(Judging by *proportional* error, not absolute)

Only zero if h=m=0.
Denominator is only very small if numerator is as well!

Considerations for Weighting

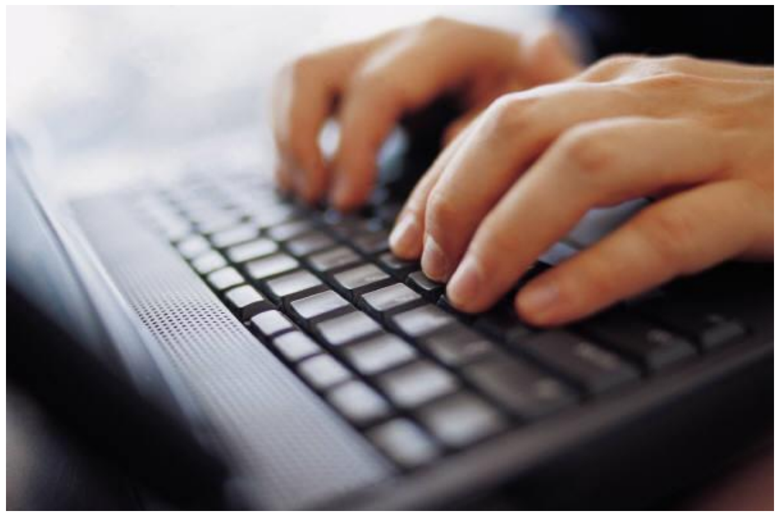
- **Purpose of model**: If we "care" more about a match with respect to some variables, we can more heavily weight matches for those variables
- **Uncertainty in estimate**: The more uncertain the estimate of the quantity, the lower the weight
- **Whether data exists**: no data => weight should be zero

Example (Simplistic) Global Optimization Algorithm

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

Many more sophisticated "global optimization" algorithms are available and can improve the outcome & speed of optimization (e.g. genetic algorithms, swarm-based methods)

Hands on Model Use Ahead

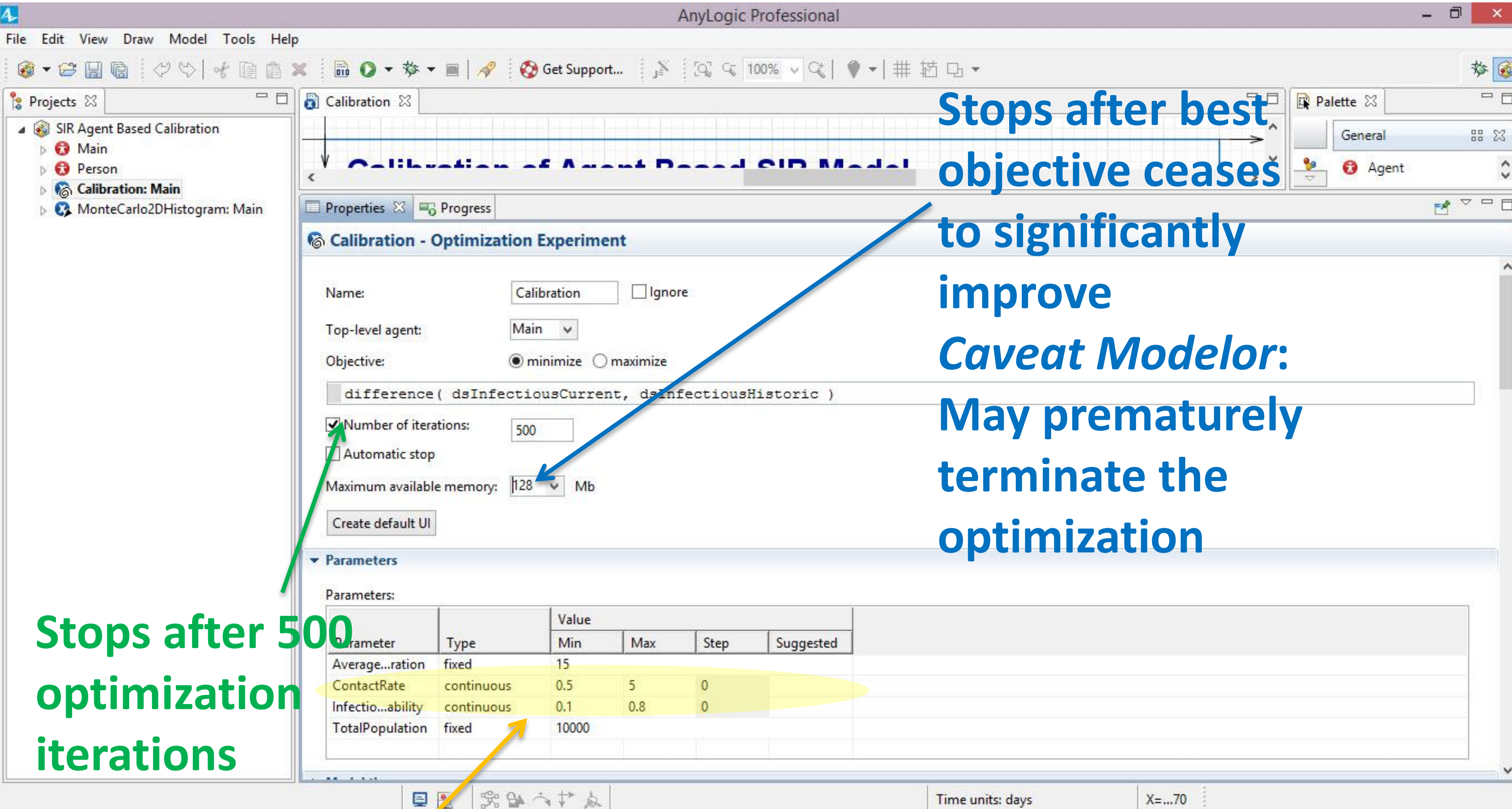


Load Sample Model:

SIR Agent Based Calibration

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

Recall: Optimization Experiment in AnyLogic



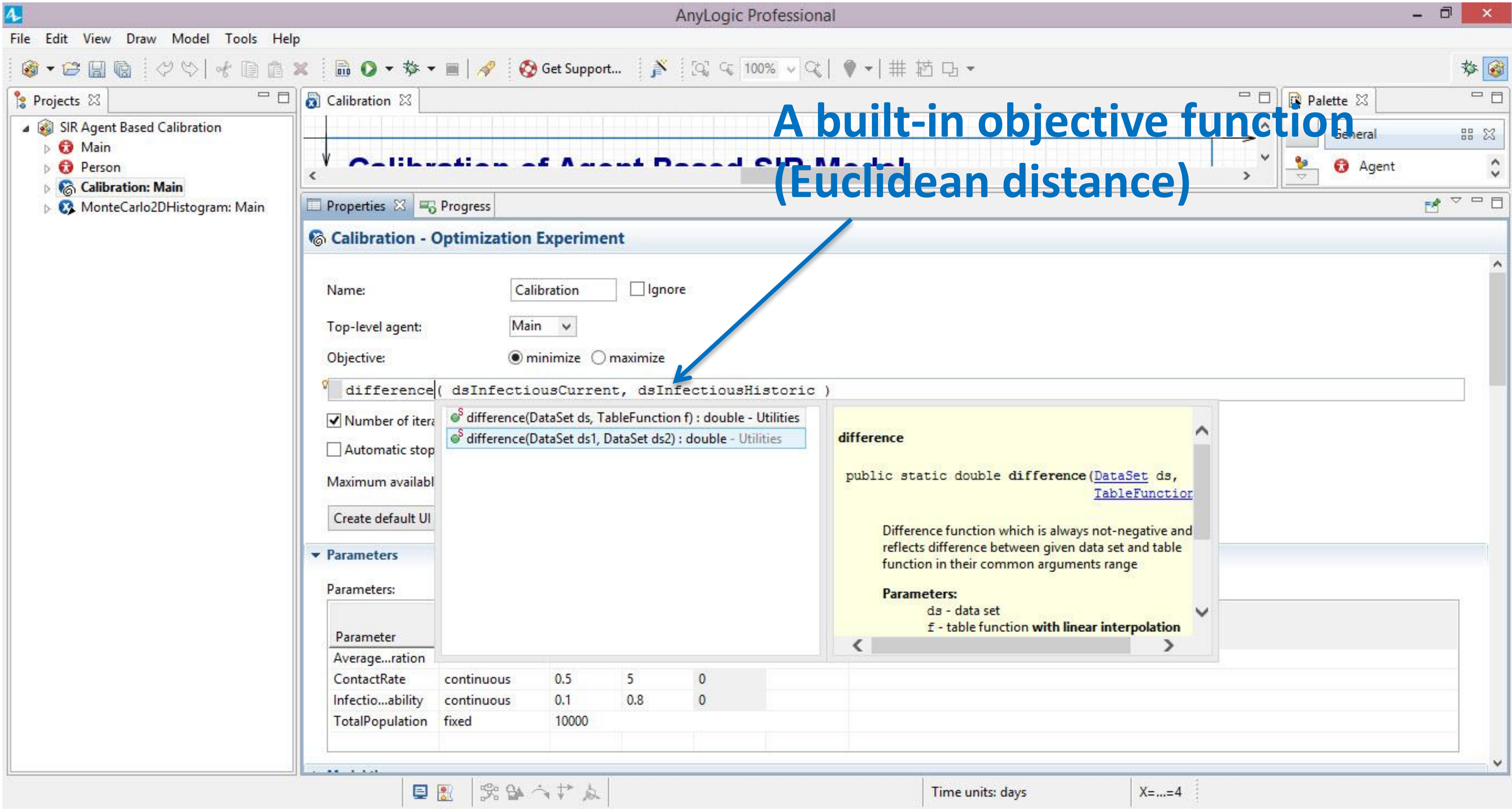
Stops after best objective ceases to significantly improve

***Caveat Modelor*: May prematurely terminate the optimization**

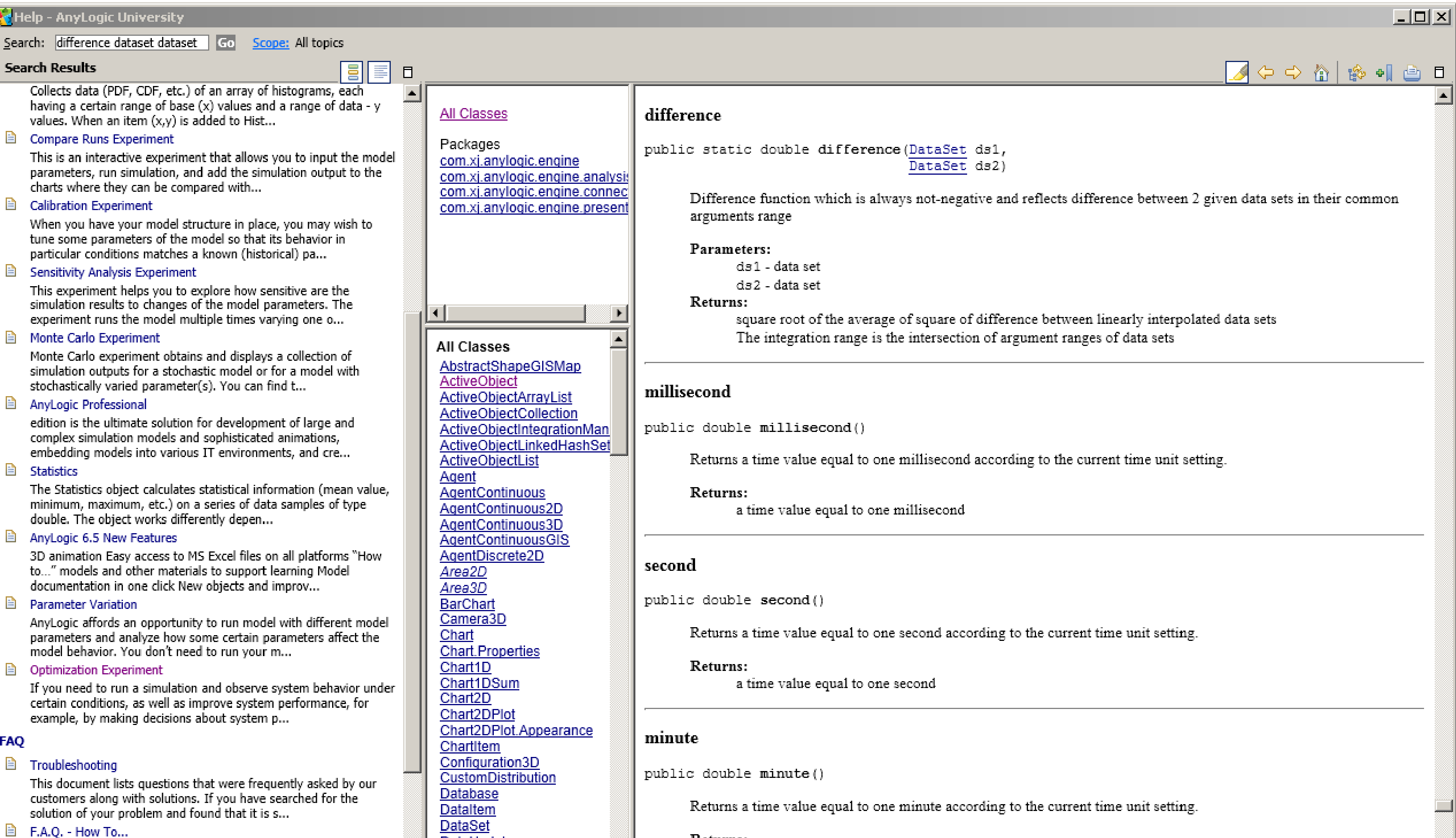
Stops after 500 optimization iterations

Varying these parameters

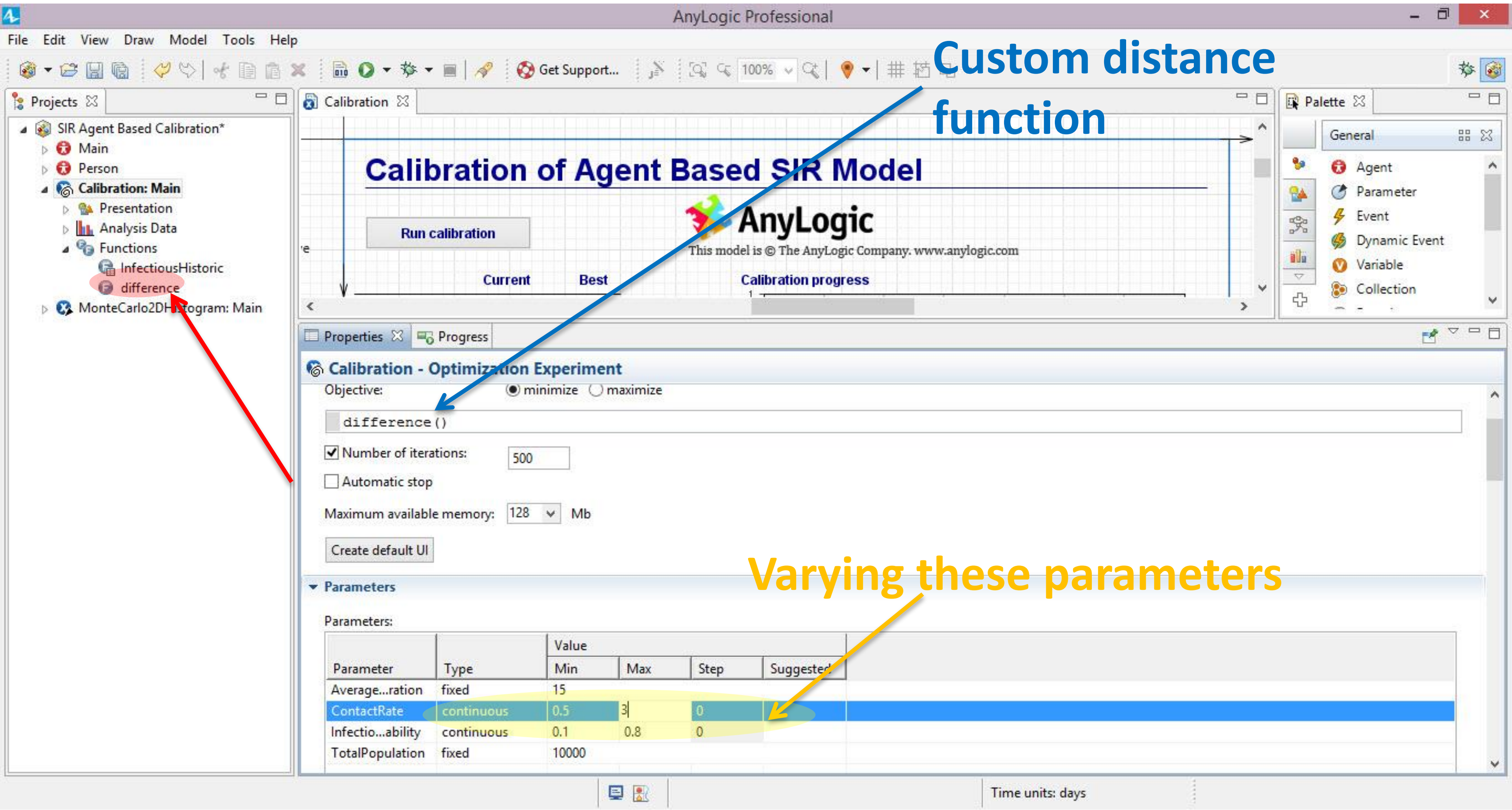
An Optimization Experiment in AnyLogic Using Built-in Difference Function



Finding the Definition

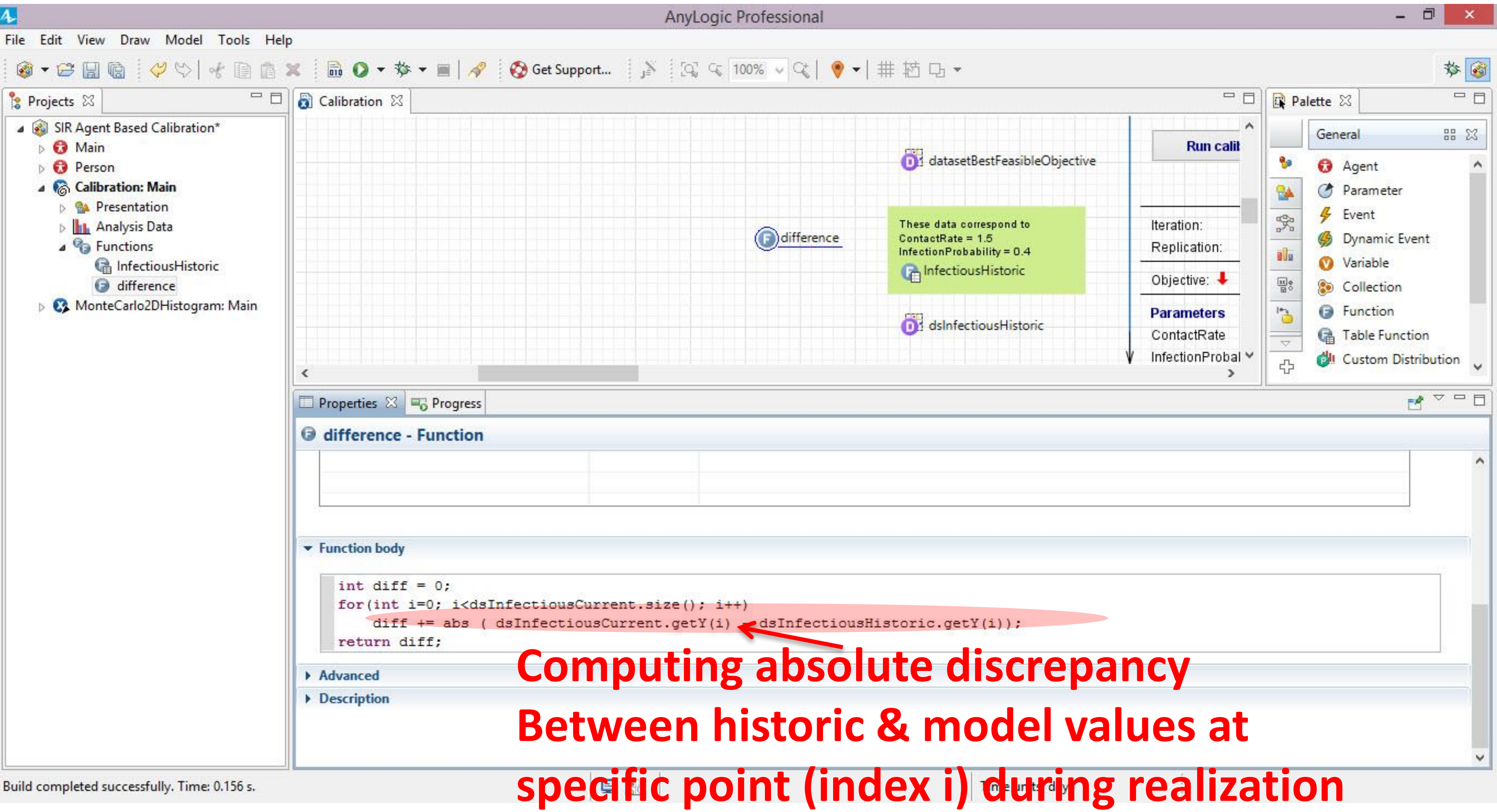


An Optimization Experiment in AnyLogic with a custom difference function



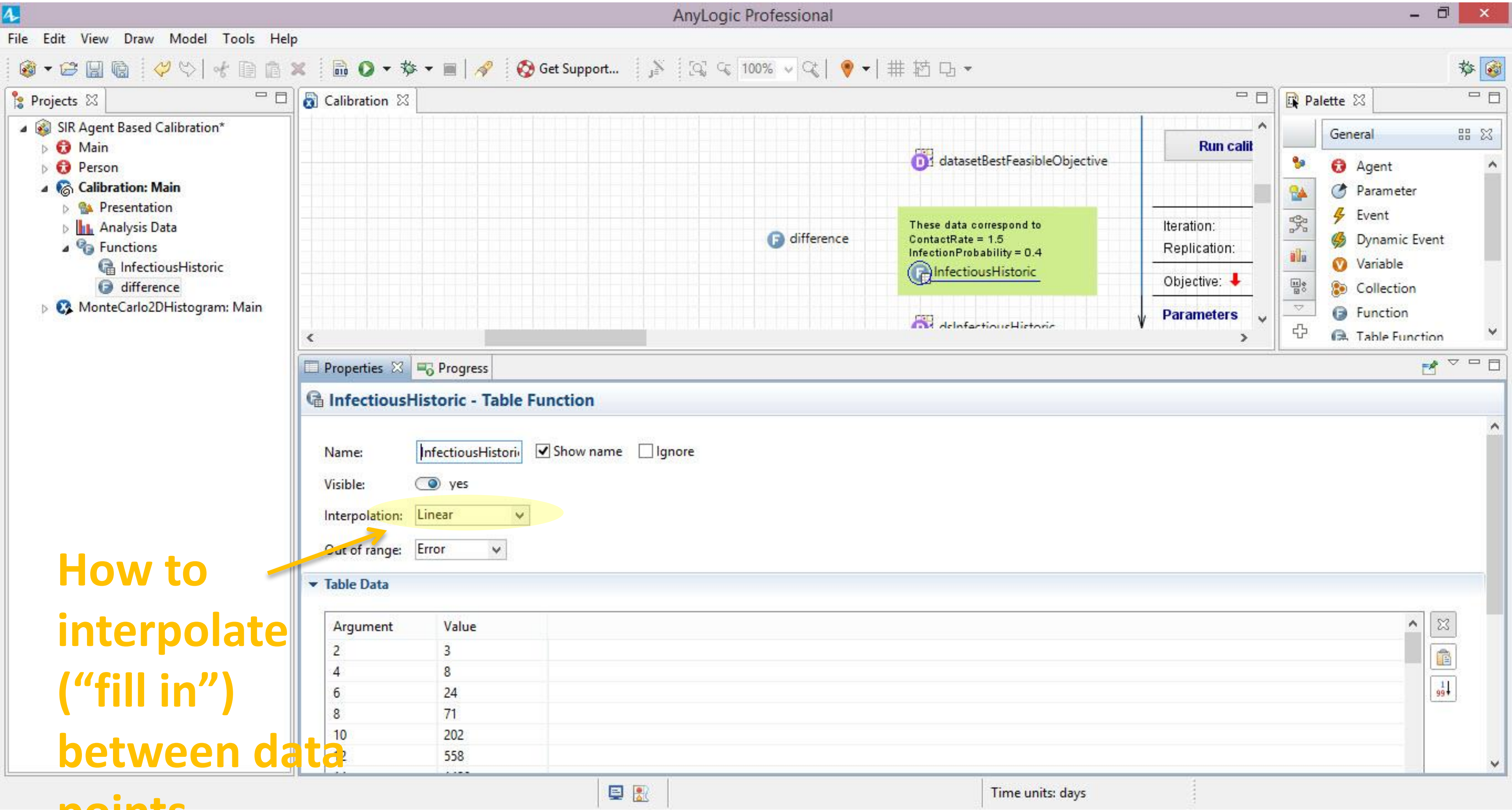
Defining a Payoff Function

Caveat: Here, Non-Analytic, Non-Concave



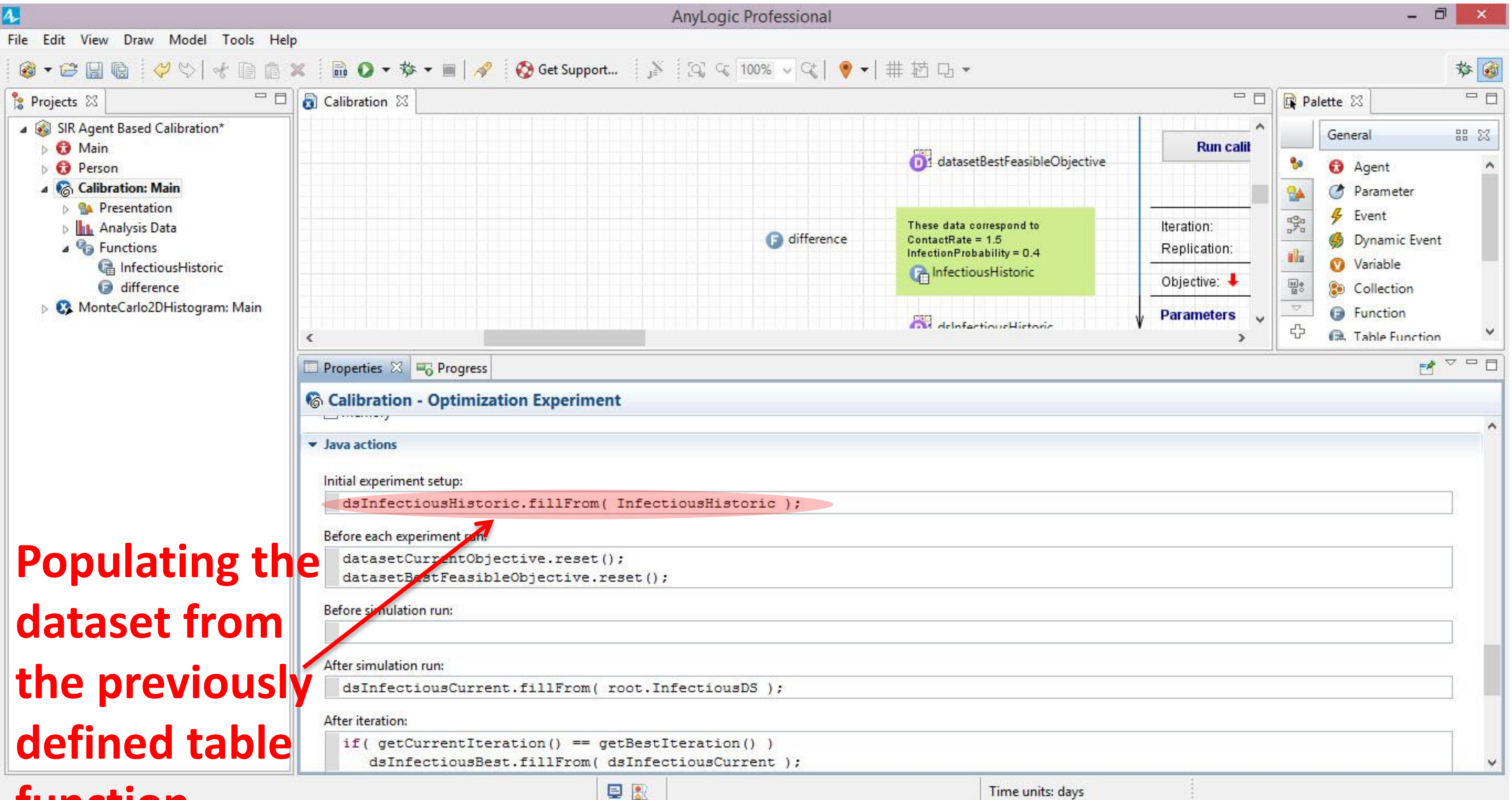
Computing absolute discrepancy Between historic & model values at specific point (index i) during realization

Historic Data Captured via Table Function



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

Populating a Dataset with Historic Data



Populating the dataset from the previously defined table function

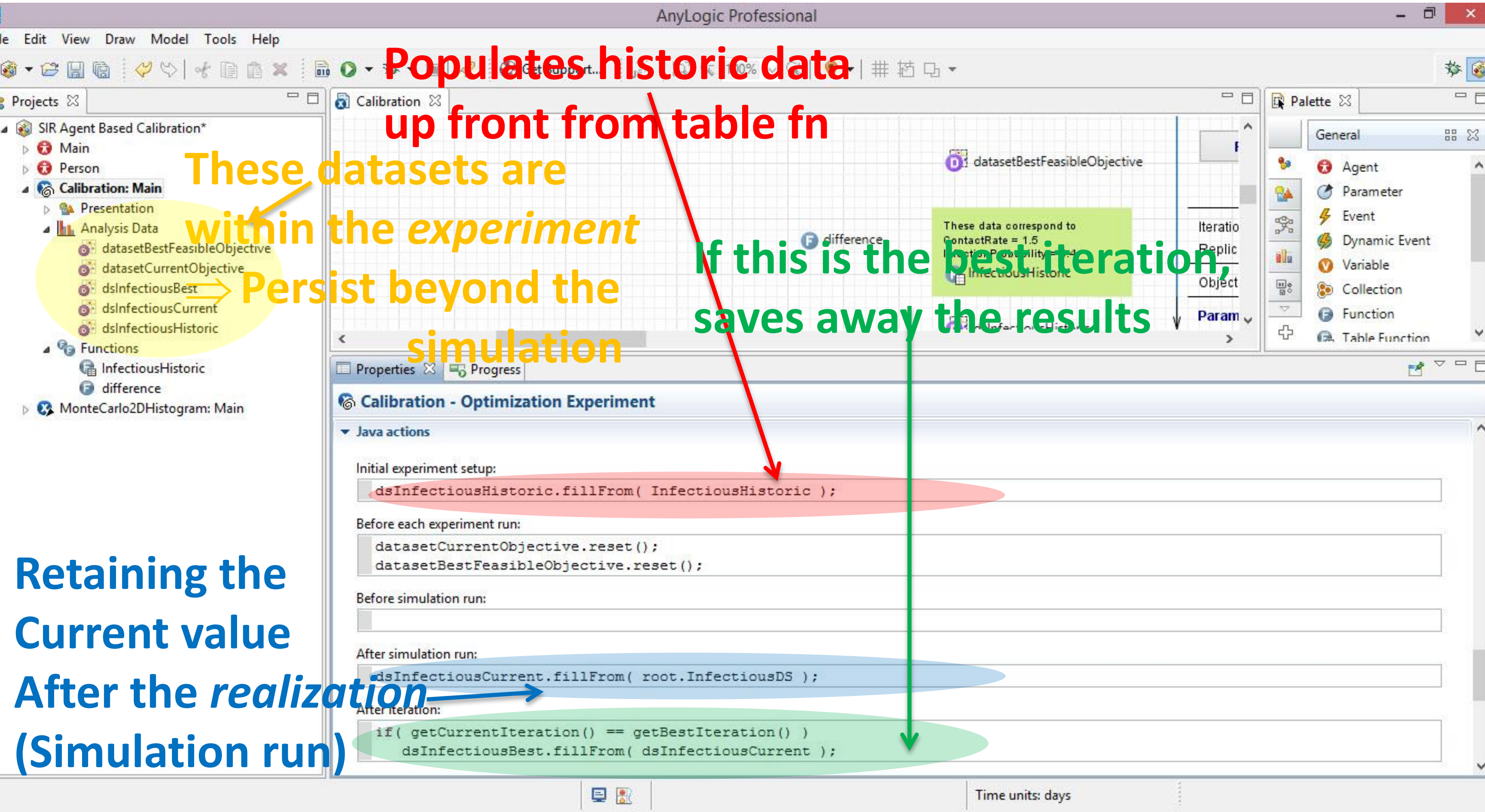
Stochastics in Agent-Based Models

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

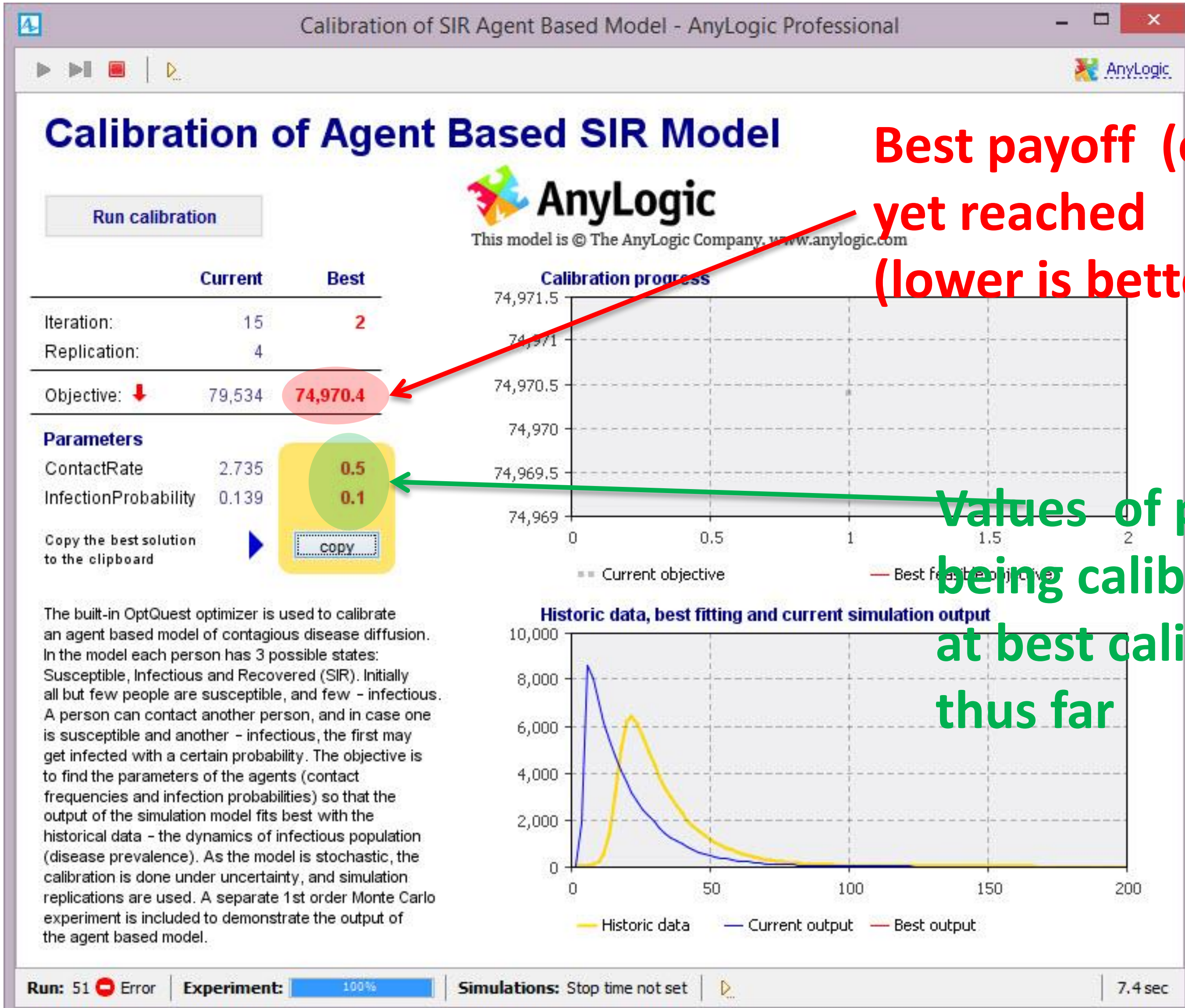
Recall: Important Distinction (Declining Order of Aggregation)

- Experiment
  – Collection of simulations
- Simulation (i.e., Scenario)
  – Collection of replications that can yield findings across set of replications (e.g. mean value)
- Replication (i.e., Realization)
  – A Single realization (“run”) of the model, with a unique random number seed

Populating the Appropriate Datasets



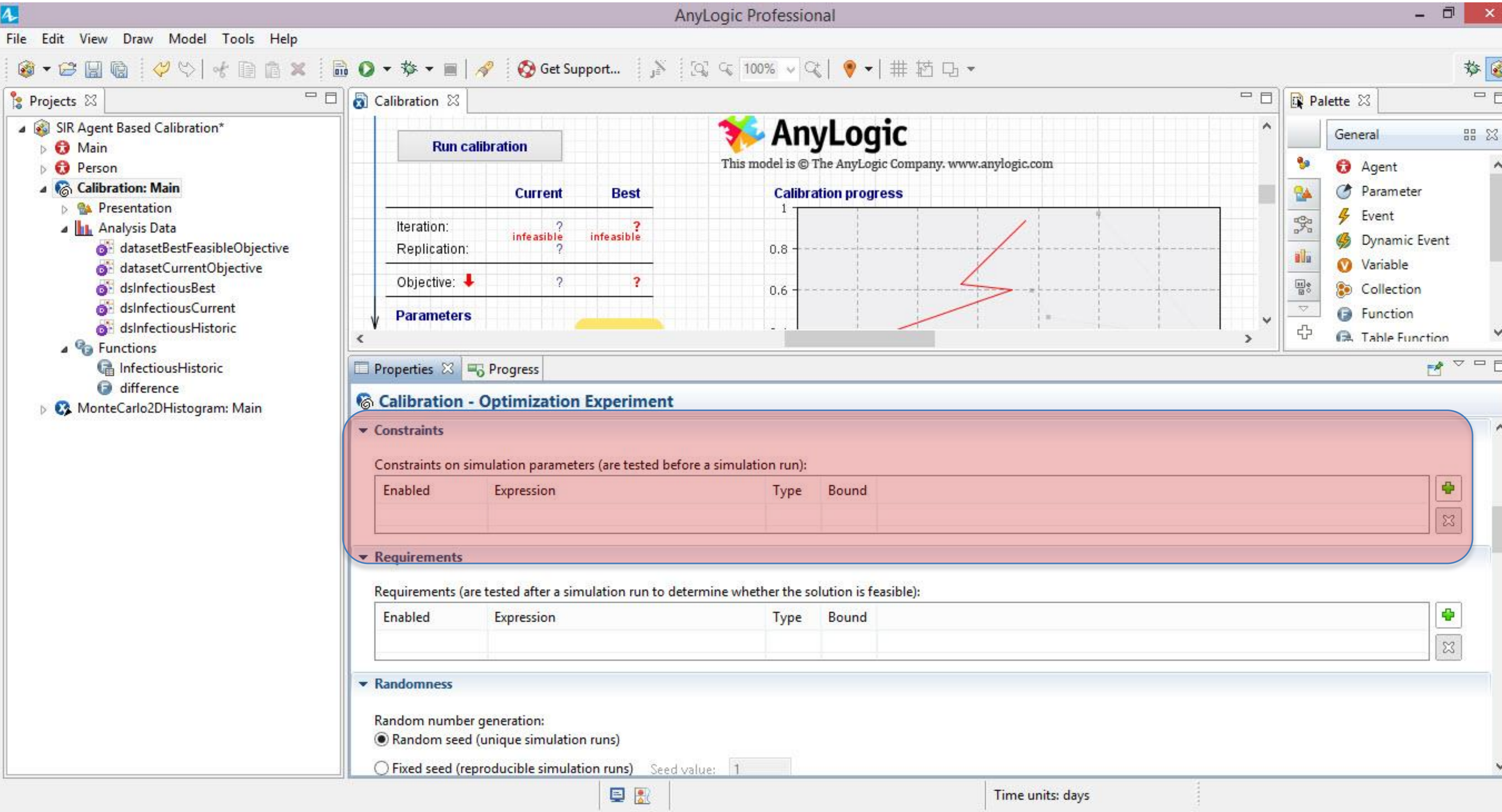
Running Calibration in AnyLogic



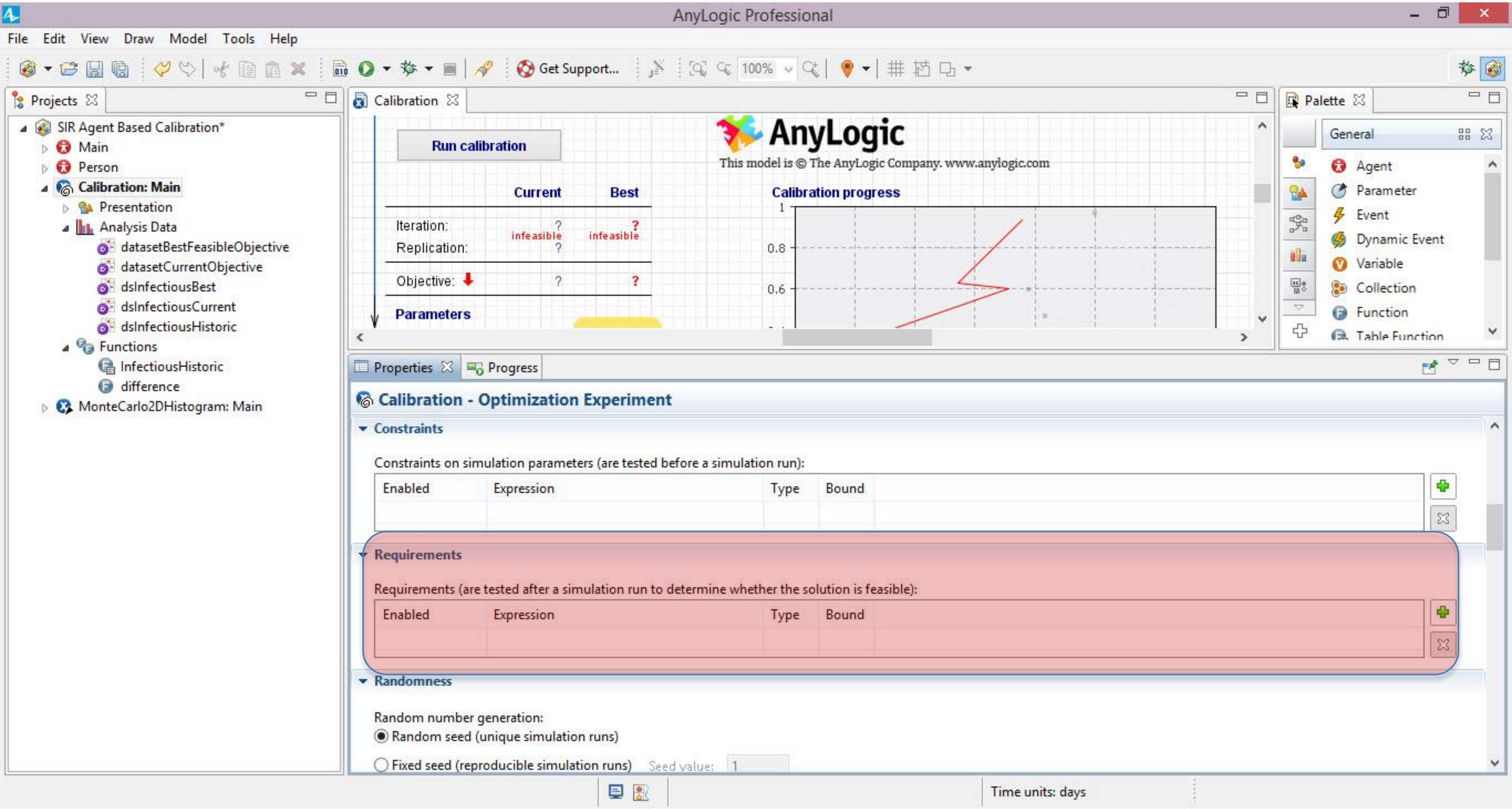
Best payoff (objective) yet reached (lower is better)

Values of parameters being calibrated at best calibration thus far

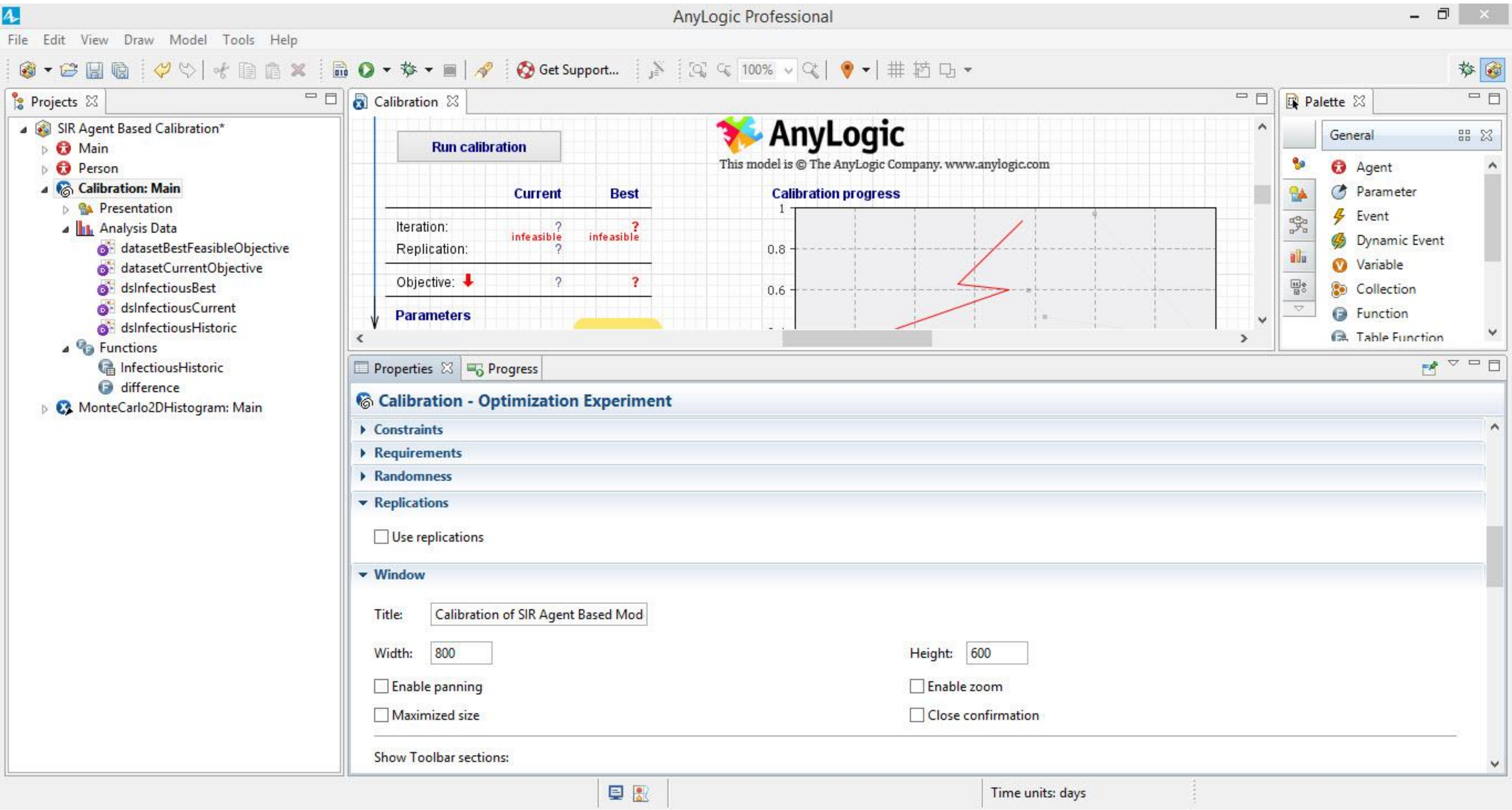
Optimization Constraints – Tests on Legitimacy of Parameter Values



Optimization Requirements – Tests to Sense Validity of Emergent Results

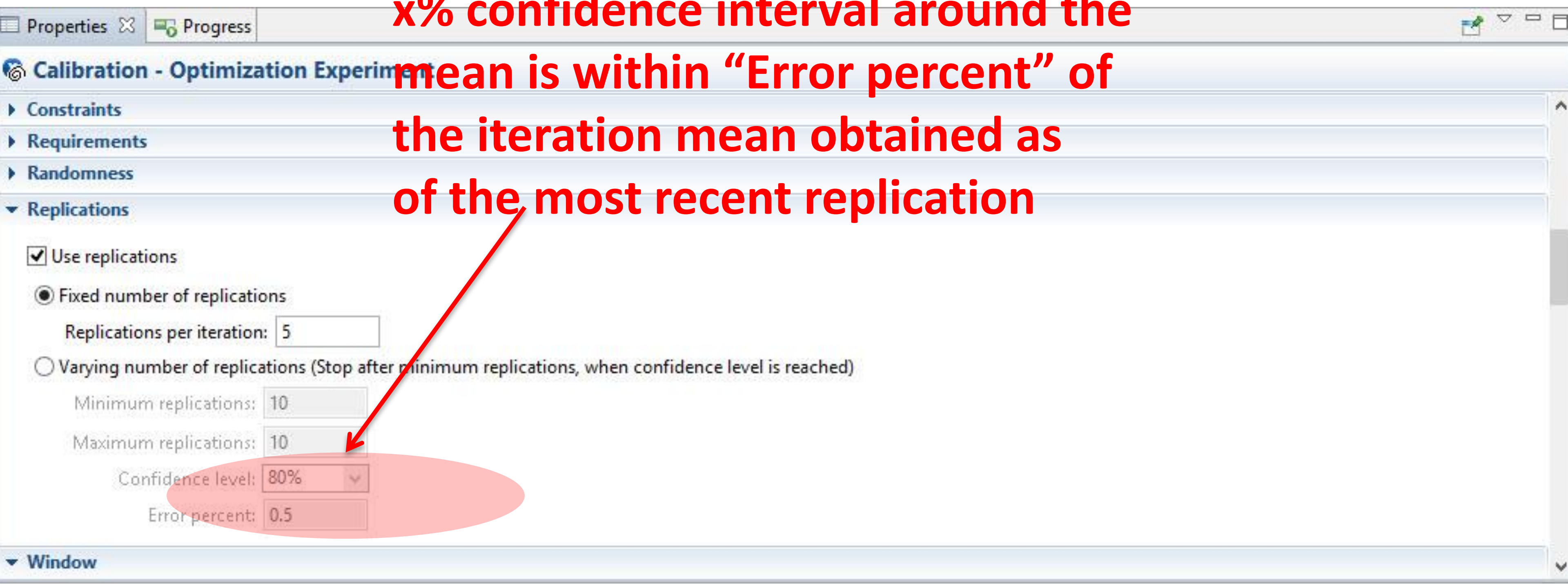


Enabling Multiple Realizations (“Replications”,”Runs”) per Iteration

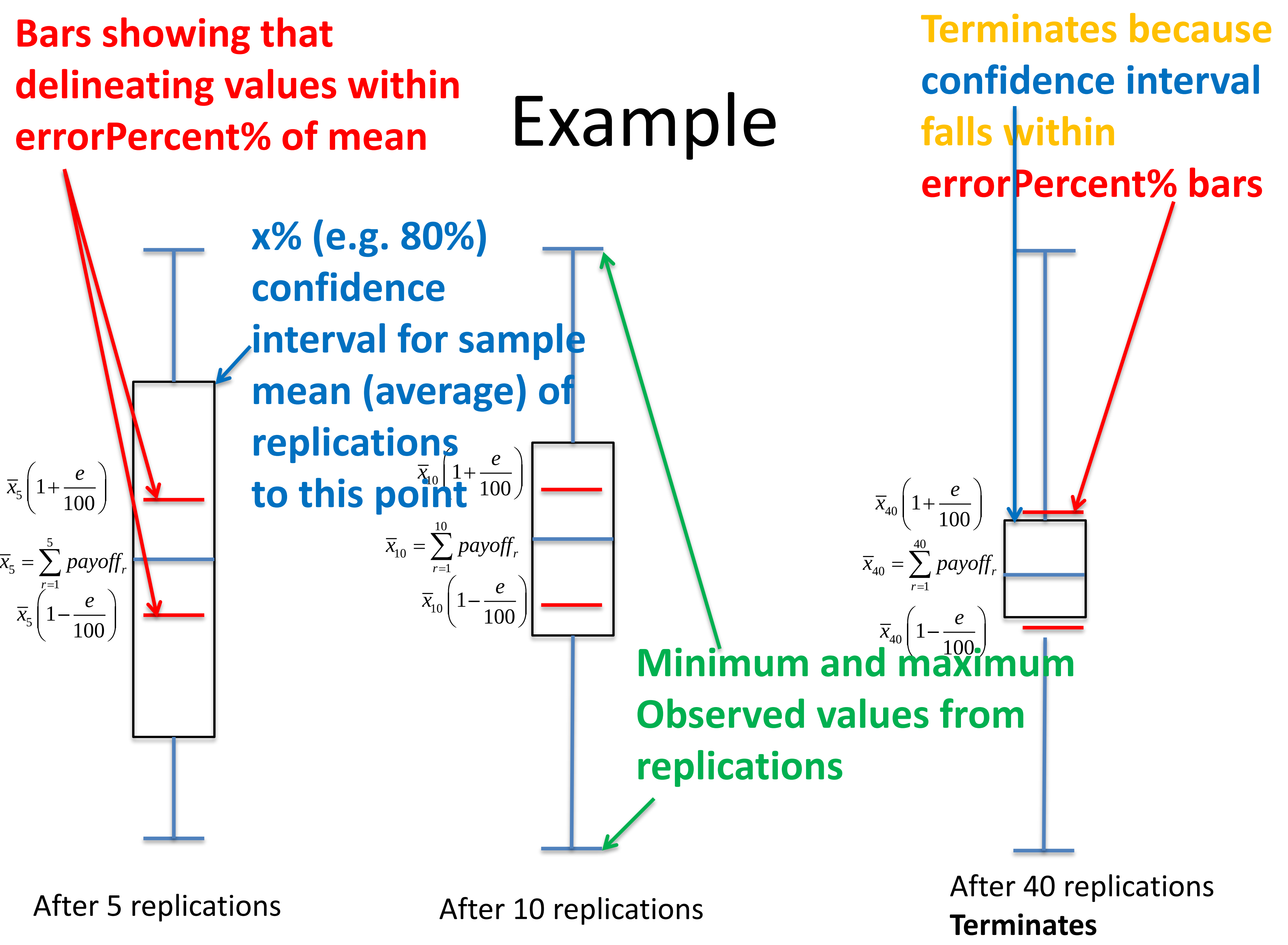


Fixed Number of Replications per Iteration

Specifies stopping Condition once minimum replications have been run. Indicates that the x% confidence interval around the mean is within "Error percent" of the iteration mean obtained as of the most recent replication

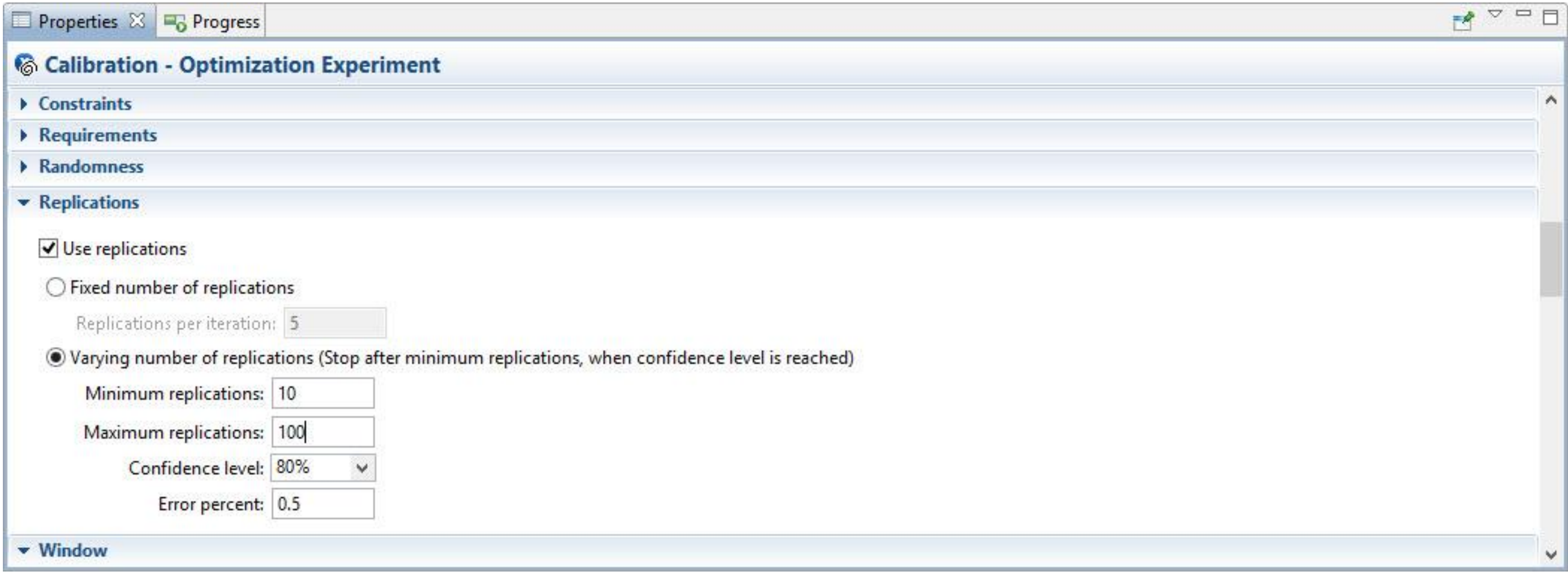


Example

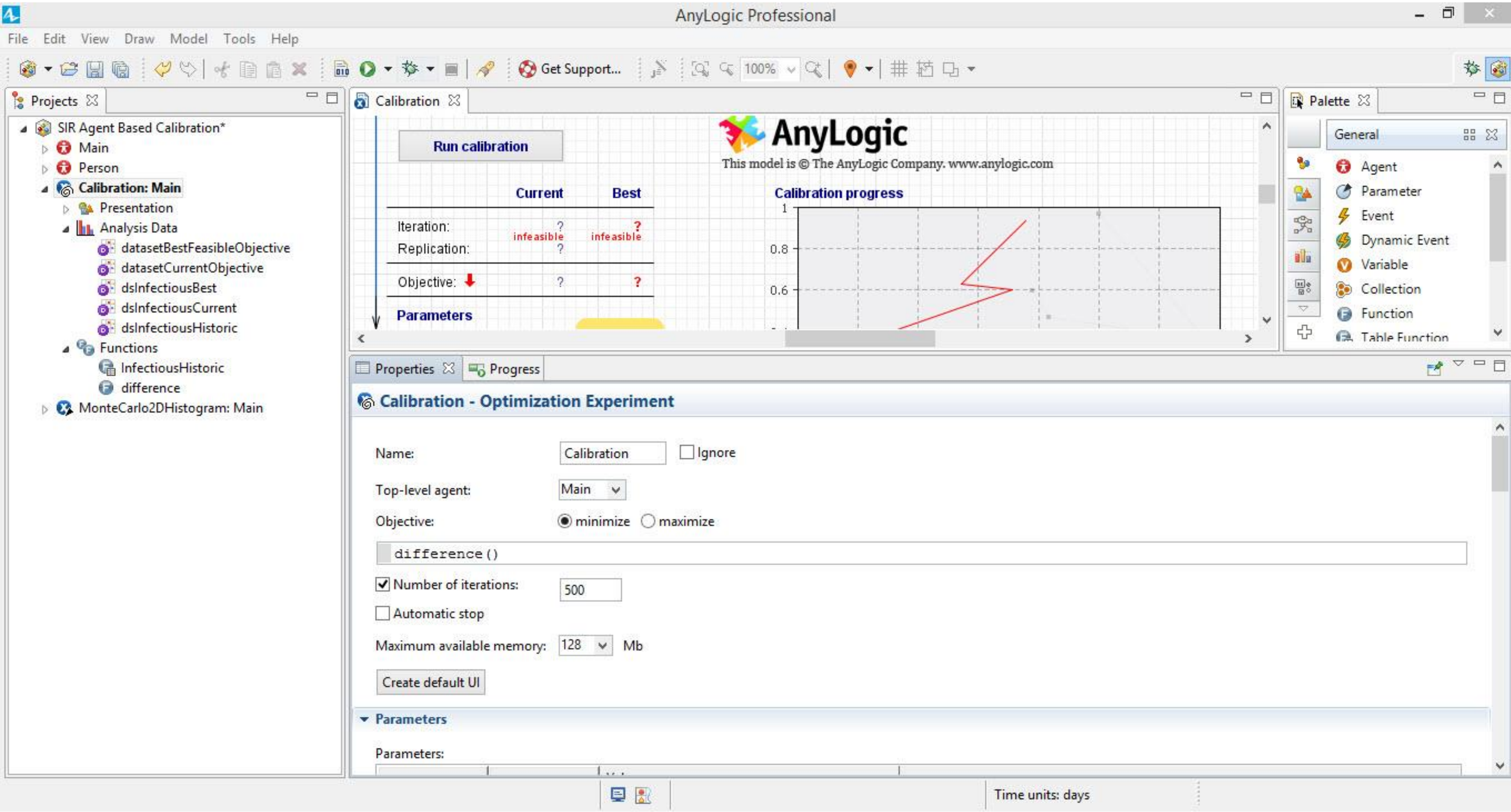


Bars showing that delineating values within errorPercent% of mean
x% (e.g. 80%) confidence interval for sample mean (average) of replications to this point
Terminates because confidence interval falls within errorPercent% bars
Minimum and maximum Observed values from replications
$\bar{x}_5\left(1+\frac{e}{100}\right)$
$\bar{x}_5 = \sum_{r=1}^{5} payoff_r$
$\bar{x}_5\left(1-\frac{e}{100}\right)$
$\bar{x}_{10}\left(1+\frac{e}{100}\right)$
$\bar{x}_{10} = \sum_{r=1}^{10} payoff_r$
$\bar{x}_{10}\left(1-\frac{e}{100}\right)$
$\bar{x}_{40}\left(1+\frac{e}{100}\right)$
$\bar{x}_{40} = \sum_{r=1}^{40} payoff_r$
$\bar{x}_{40}\left(1-\frac{e}{100}\right)$
After 5 replications
After 10 replications
After 40 replications
Terminates

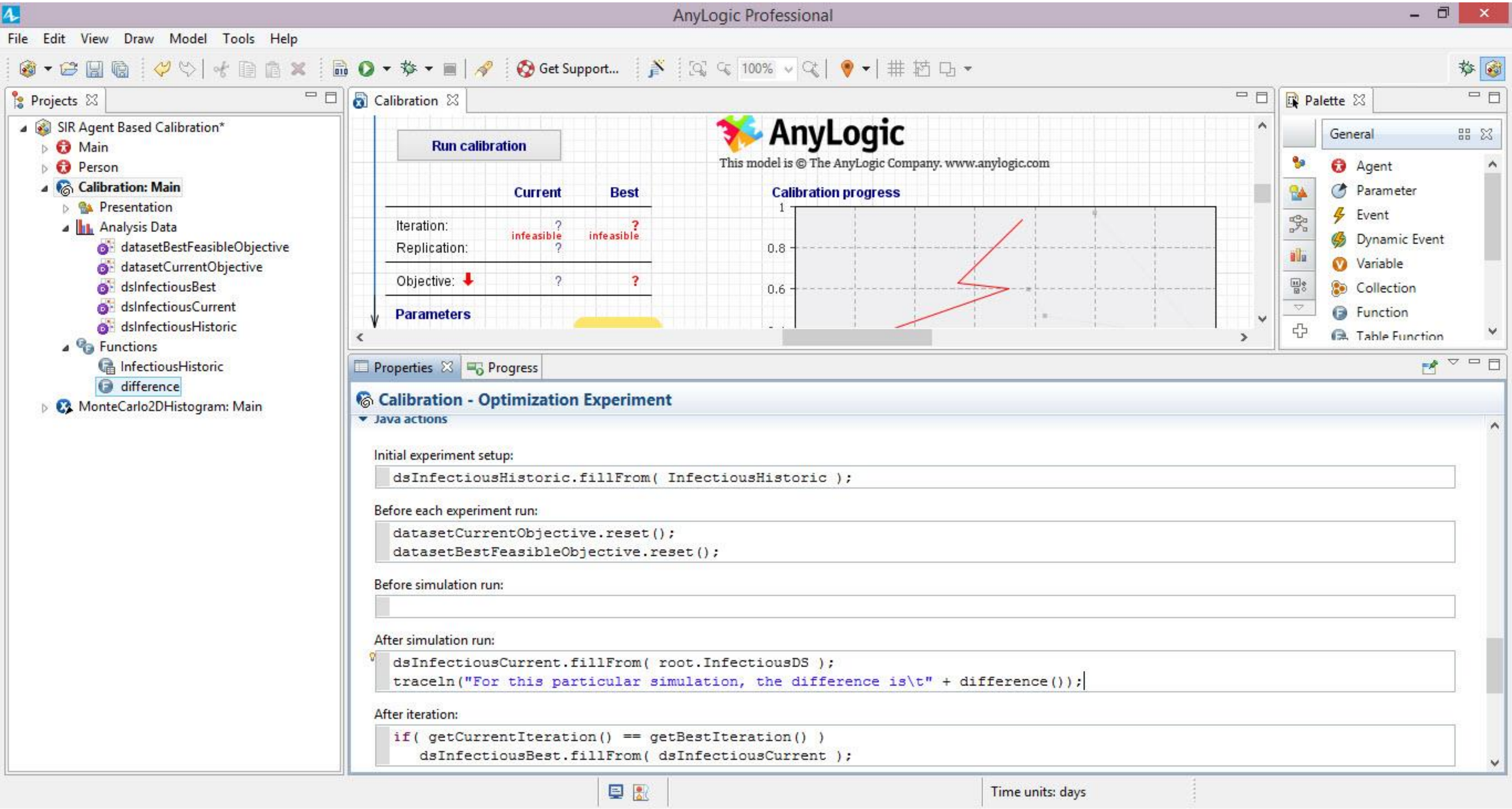
Automatic Throttling of Replications Based on Confidence Intervals for the Average of the Differences between Best and Current



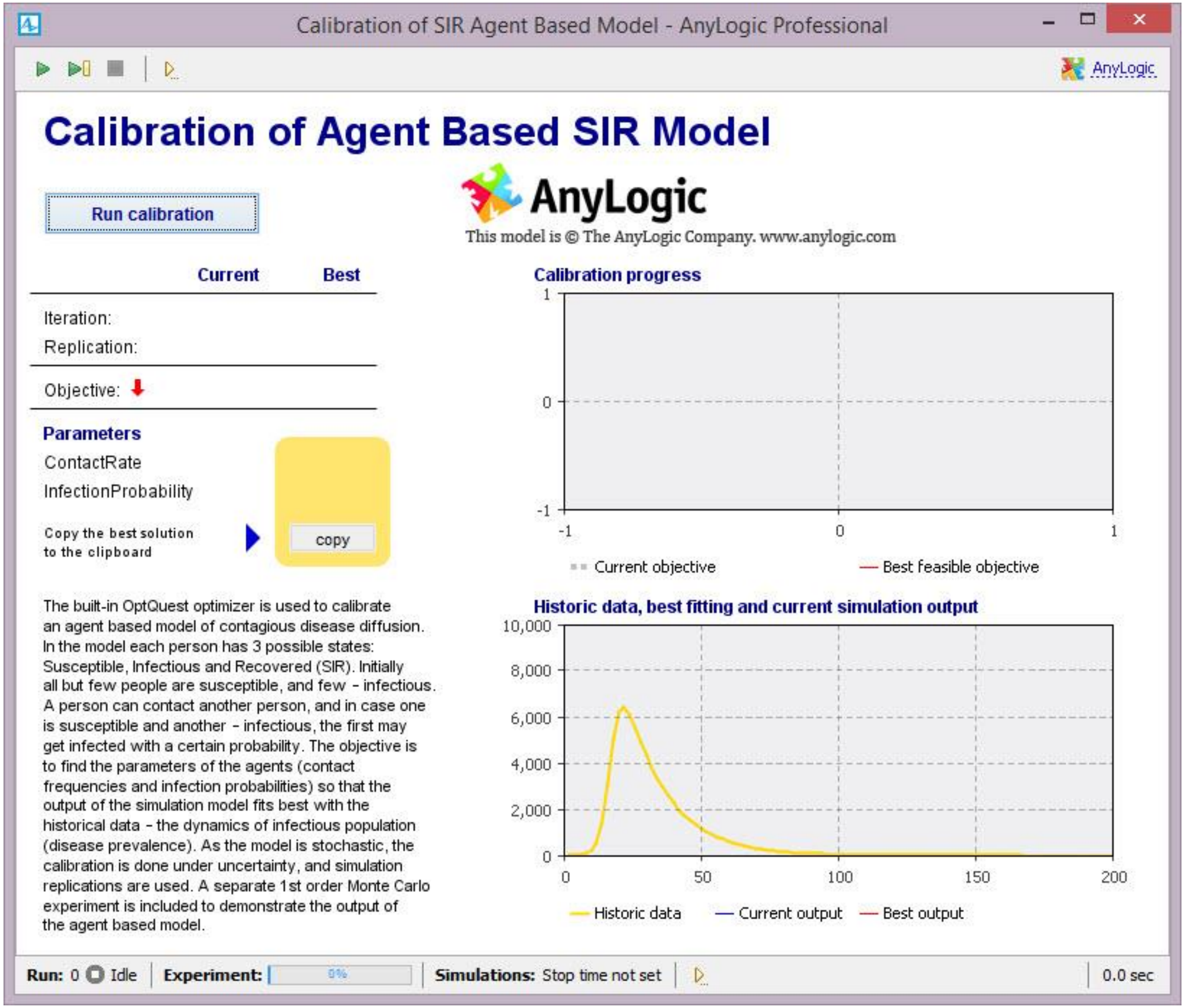
Enabling Random Variation Between Realizations ("Replications")



Understanding Replications: Report Results for Each Replication!

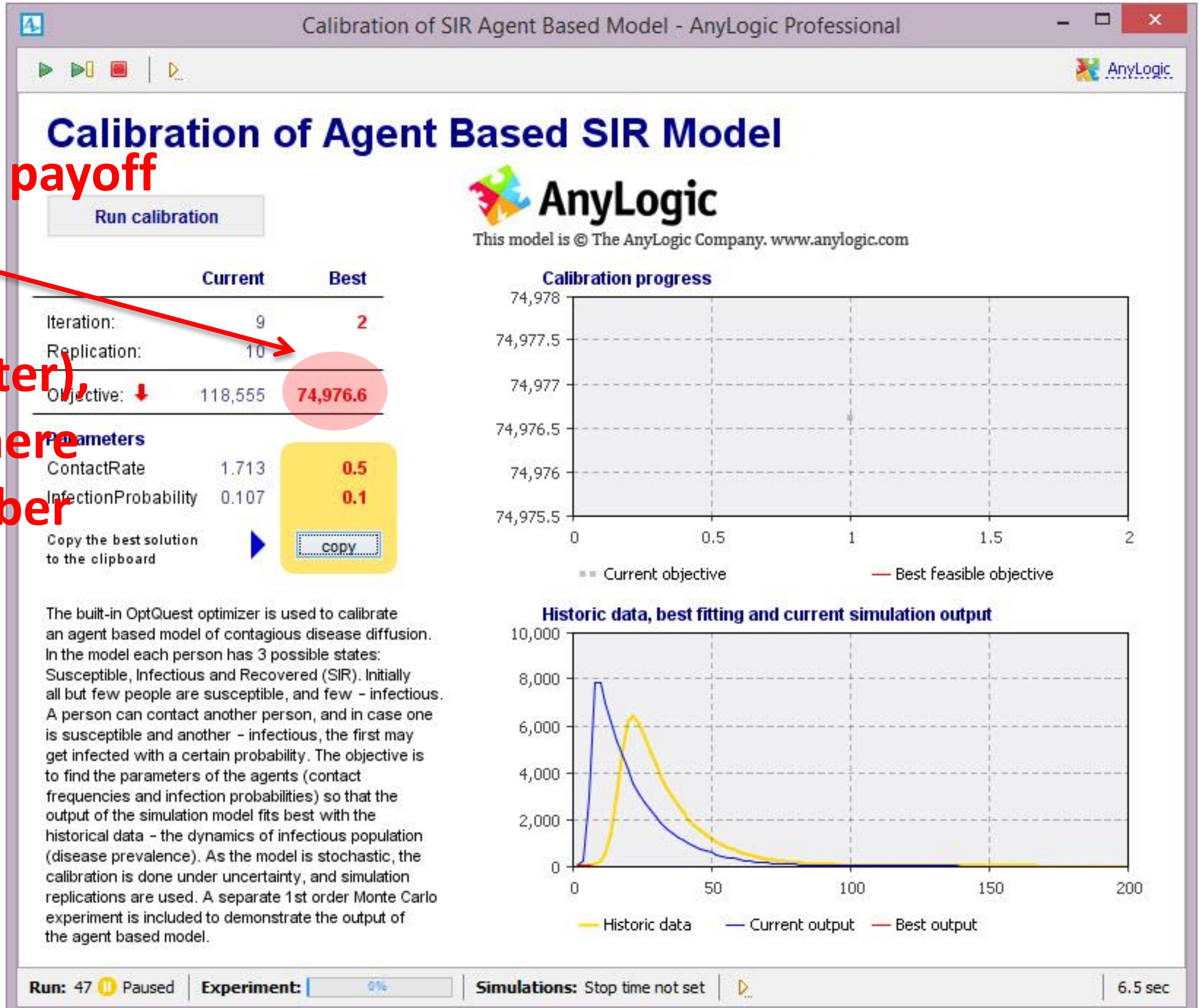


During First Several Realizations ("Replications", "Runs"), No Results Appear

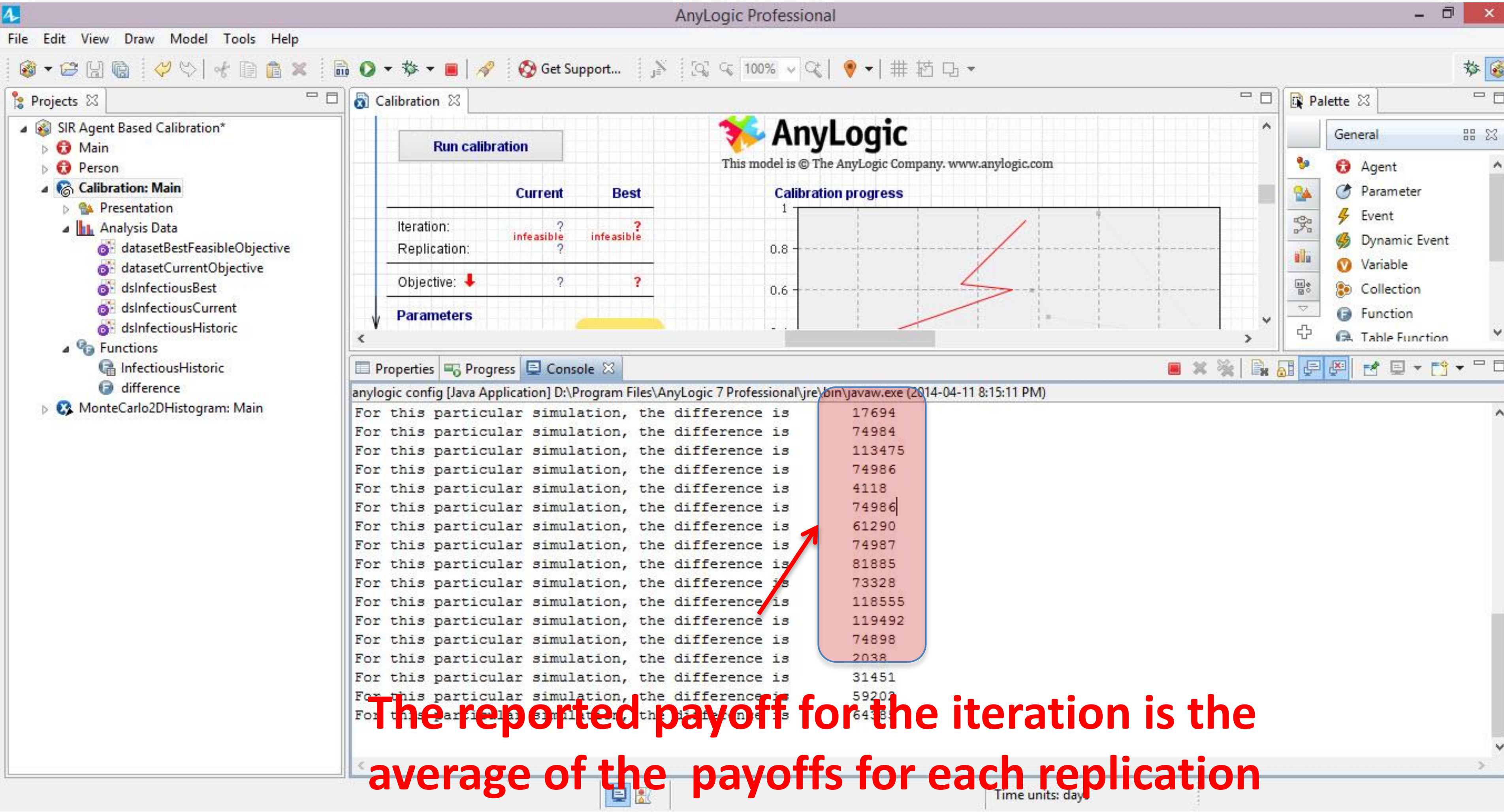


Report on Iteration 2 Appears after a Count of Runs Equal to Replications per Iteration

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

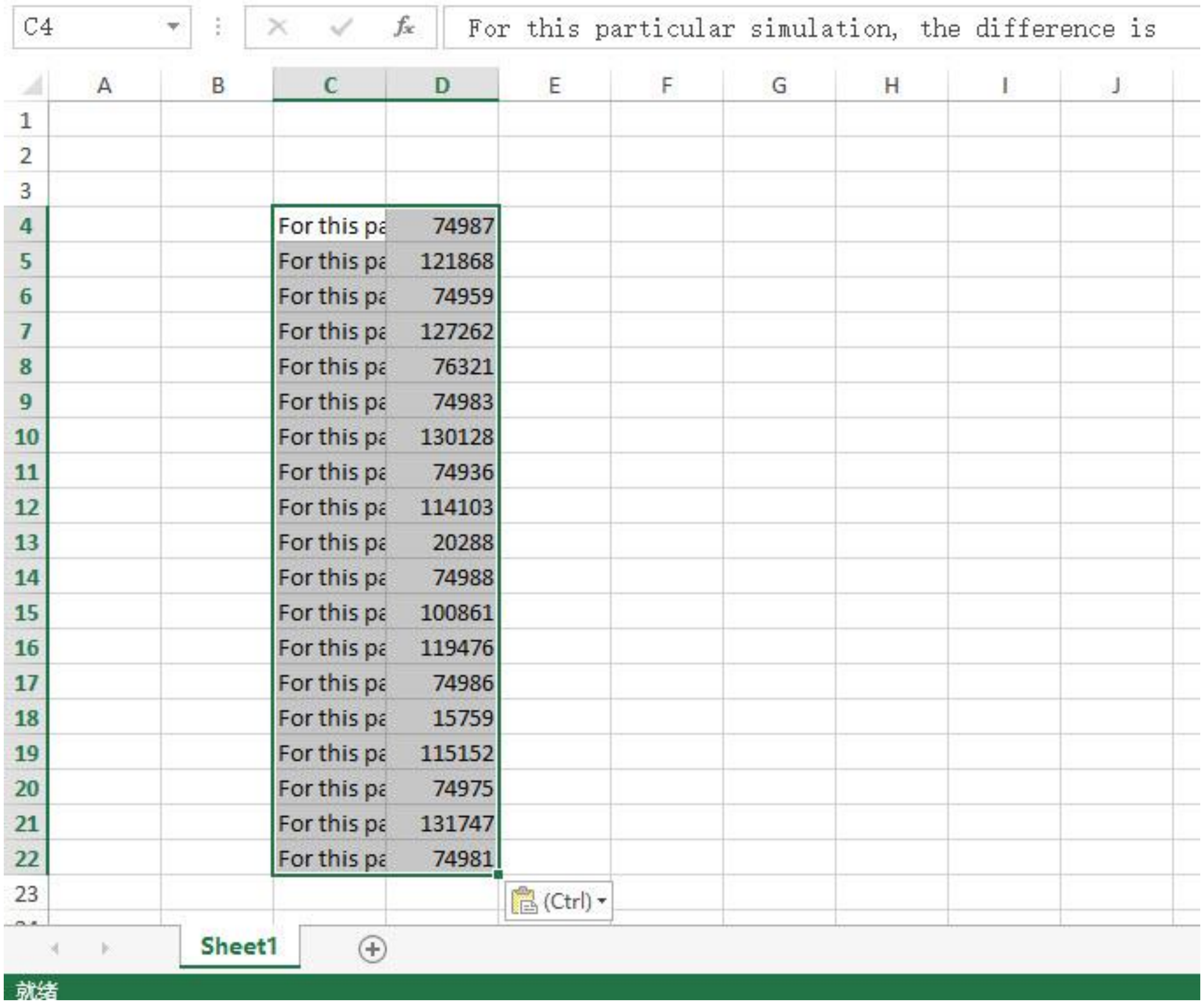


Output



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

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


C4 For this particular simulation, the difference is

	C	D
4	For this pa	74987
5	For this pa	121868
6	For this pa	74959
7	For this pa	127262
8	For this pa	76321
9	For this pa	74983
10	For this pa	130128
11	For this pa	74936
12	For this pa	114103
13	For this pa	20288
14	For this pa	74988
15	For this pa	100861
16	For this pa	119476
17	For this pa	74986
18	For this pa	15759
19	For this pa	115152
20	For this pa	74975
21	For this pa	131747
22	For this pa	74981

Considerations

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

Dealing with Calibration Problems: Experiments

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

Dealing with Calibration Problems: Additional Experiments

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

Important Cross-Checks: Uniqueness

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

Important Cross-Checks: Binding Constants

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

Capturing Parameter Interdependencies in Calibration

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

Calibrating Initial Conditions

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

Calibration & Regression: Similarities & Differences

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