Addressing parameter uncertainty in SD models with fit-to-history and Monte-Carlo sensitivity methods

By

Wayne Wakeland, PhD, Systems Science Graduate Program, Portland State University, PO Box 751, Portland, OR 97207, USA.

Jack Homer, Homer Consulting, 72 Station Hill Road, Barrytown, NY, 12507, USA.

Abstract

We present a practical guide, including a step-by-step flowchart, for establishing uncertainty intervals for key model outcomes in the face of uncertain parameters. The process starts with Powell optimization (e.g., using Vensim^{TM}) to find a set of uncertain parameters (the "optimum" parameter set or OPS) that minimize the model fitness error relative to available reference behavior data. The optimization process also helps in refinement of assumed parameter uncertainty ranges. Next, Markov Chain Monte Carlo (MCMC) or conventional Monte Carlo (MC) randomization is used to create a sample of parameter sets that fit the reference behavior data nearly as well as the OPS. Under the MC method, the entire parameter space is explored broadly (with a very large number of runs), and the results are sorted for selection of qualifying parameter sets (QPS) based on goodness-of-fit criteria. The statistical properties of the QPS parameter distributions are analyzed to ensure their centrality relative to the uncertainty ranges. Also, the full set of QPS outputs are graphed (as sensitivity graphs or box-and-whisker plots) for comparison with the reference behavior data. Finally, alternative policies and scenarios are run against the OPS and all QPS, and confidence intervals are found for key model outcomes. We demonstrate the method with a substantial model that is in the process of being published, and we discuss how such analyses can be used by policy/decision makers.

1. Introduction

System dynamics (SD) models frequently employ parameters for which solid empirical data is not available. Modelers often use expert judgment, either informally or formally, to provide estimates of parameter values for which empirical data is not available. When several experts are available, and a formal process such as a Delphi survey is used to obtain their judgments, model developers and users may have high confidence in the values of these parameters despite the lack of data. Nevertheless, these parameter values remain uncertain to a degree.

To address parameter uncertainty, SD modelers experiment extensively with alternative parameter values in order to understand their degree of influence in the model. Formal sensitivity analyses can (and should) be run using features provided in popular SD software packages. Results can be displayed in a table or portrayed a graphically as a “Tornado” diagram (c.f. Wakeland and Hoarfrost 2005).

Once the modeler has identified influential parameters, additional effort is applied as time and budget allow to increase confidence that the values of these influential parameters are well supported. When reliable empirical data are available, ideally from multiple sources, the parameter value may be fixed and used with confidence. Usually, however, some, perhaps even many, parameter values remain uncertain. This paper describes how the model analysis features available in the Vensim ^{TM} software, and likely other software, can be used to represent and incorporate that uncertainty into the analysis of SD model results, including, potentially, providing outcome intervals informed by parameter uncertainty.

A brief background section reviews the use of these methods in recent literature. A methods section provides a detailed diagram describing the process for incorporating parameter uncertainty in SD models, followed by a results section that provides an illustration of the use of the method. The paper closes with a discussion section demonstrating how model results under uncertainty can be described for interested parties such as policy makers or other end users of the analyses.

2. Background

Helton et al. (2006) provide a general overview of sampling-based methods for sensitivity analysis, including Monte Carlo (MC). Cheun and Beck (2009) explain Bayesian updating in MC processes, and background material on the mathematics of Markov Chain Monte Carlo (MCMC) is readily available (cf. ter Braak 2006 and Vrugt J.A. et al. 2009). Background on sensitivity analysis methods for dynamic modeling, and SD in particular, including the MC and MCMC methods being employed in this paper, may be found in Houghten et al. (2014), Houghten and Siegel (2015), Osgood (2013), and Osgood and Liu (2015). Using conventional MC methods to search a parameter space is sometimes considered to be a brute force or totally random search, whereas the MCMC method, uses a Bayesian update process to guide the search of parameter space. In at least some contexts, it has been shown that MCMC optimally creates statistically valid samples (ter Braak 2006).

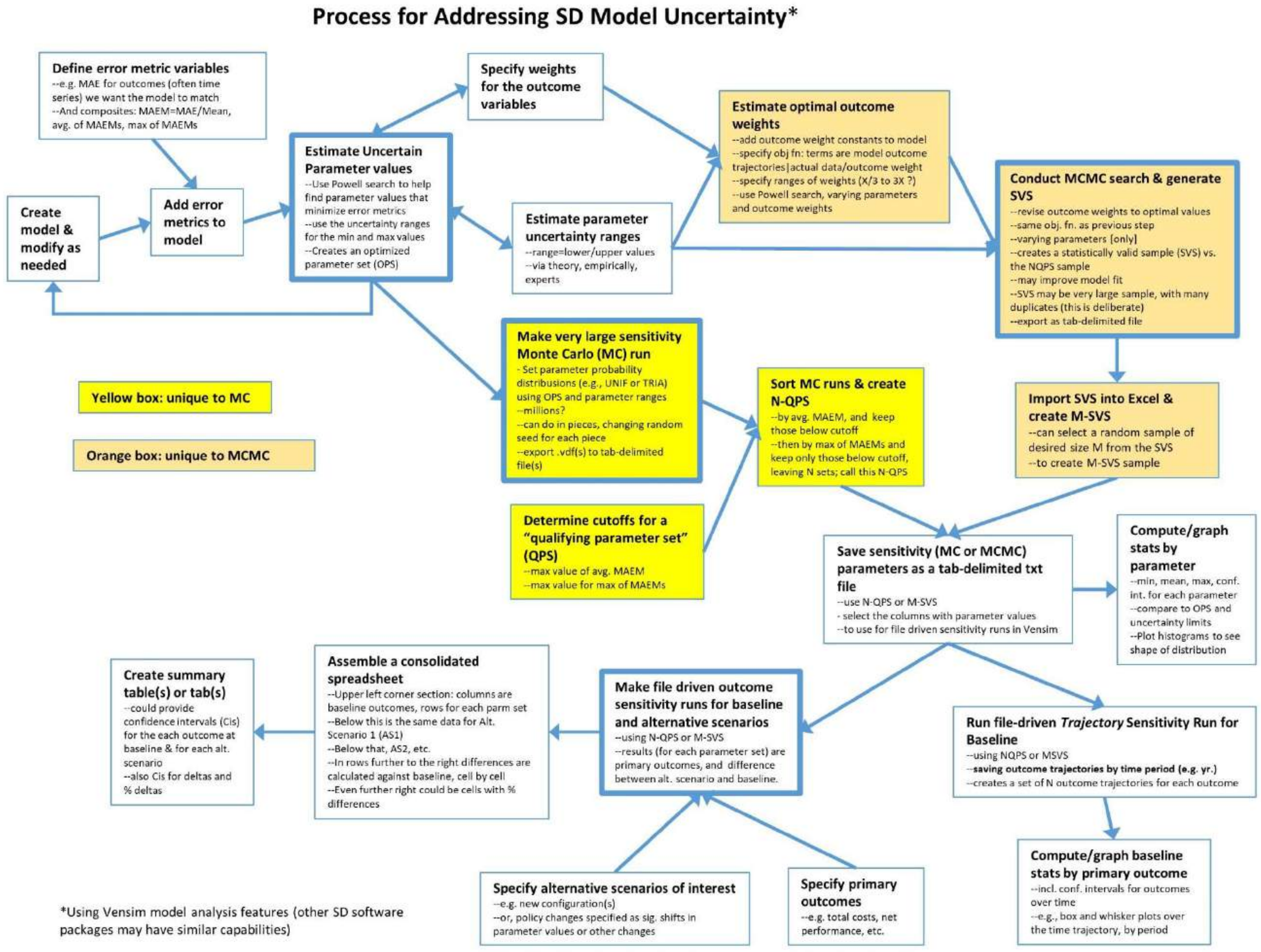
Although publications describing the application of these methods are plentiful is some scientific and engineering disciplines, publications featuring their use with SD models are scant. A search for “system dynamics” AND (“MCMC” or “monte carlo”) return only a handful of publications. Two examples are Jeon and Chin (2014), who describe their use of MC with an SD model applied to a renewable energy study, and Sterman et al. (2018), who describe their use of MCMC in the evaluation of an SD forestry industry model. Conference papers are no more numerous. A search of the SD conference proceeds for MCMC or Monte return just one result for the period from 2015-2018, a methods paper (Fiddaman and Yeager, 2015).

3. Method

Figure 1 summarizes a detailed process for incorporating uncertainty analysis into SD models. Two alternative approaches are included. One approach employs a conventional MC-based search of parameter space, whereas the other uses the MCMC approach.

In the following text, ***bold italics*** are used to highlight the name of the block in Figure 1 being discussed. The process starts at ***Create model & modify as needed***. This would be a model that employs uncertain parameters and includes dynamic outcome variables that strive to match the dynamics seen in real world reference behavior data. To use the methods described in Figure 1, we need to ***Define error metric variables*** and ***Add error metrics to model***. A useful example could be to compute the mean absolute error (MAE) between the model calculated time series and the reference behavior time series for each outcome variable. Care must be taken to consider how to compute MAE when reference data is incomplete so as not to distort results. When different outcomes have very different scales, it is useful to use MAEM which is MAE over M, where M is the mean value of the metric. In addition, composite error statistics are added to the model, such as the average of the MAEMs over all the outcomes, and the maximum value of the individual MEAMs. These are used later in the process for identifying well-fitting parameter sets. There are statistical macros available for Vensim to help with this.

Another key step at the outset is to ***Estimate parameter uncertainty ranges***, a lower and upper bound for each uncertain parameter, and to ***Specify weights for the outcome variables.*** These are needed for the algorithm used in a key part of the process: ***Estimate Uncertain Parameter values***. This process employs a Powell optimization process that uses an objective function consisting of model outcomes vs. reference data. Each term of the objective function represents one of the outcome time series of interest. The algorithm strives to minimize the differences between model and reference data. Weights are needed because outcomes may have very different scales and variances. This tends to be an iterative process, so Figure 1 contains a feedback loop, and some of the connections are bidirectional. The end result of this step is a set of optimized parameter values for the uncertain parameters (OPS). Its average MAEM might be .1 and its max MAEM (worst MAEM for any one of the outcomes) might be .2.



Process for Addressing SD Model Uncertainty*

Define error metric variables
--e.g. MAE for outcomes (often time series) we want the model to match
--And composites: MAEM=MAE/Mean, avg. of MAEMs, max of MAEMs

Create model & modify as needed

Add error metrics to model

Specify weights for the outcome variables

Estimate Uncertain Parameter values
--Use Powell search to help find parameter values that minimize error metrics
--use the uncertainty ranges for the min and max values
--Creates an optimized parameter set (OPS)

Estimate parameter uncertainty ranges
--range=lower/upper values
--via theory, empirically, experts

Estimate optimal outcome weights
--add outcome weight constants to model
--specify obj fn: terms are model outcome trajectories|actual data/outcome weight
--specify ranges of weights (X/3 to 3X ?)
--use Powell search, varying parameters and outcome weights

Conduct MCMC search & generate SVS
--revise outcome weights to optimal values
--same obj. fn. as previous step
--varying parameters (only)
--creates a statistically valid sample (SVS) vs. the NQPS sample
--may improve model fit
--SVS may be very large sample, with many duplicates (this is deliberate)
--export as tab-delimited file

Make very large sensitivity Monte Carlo (MC) run
- Set parameter probability distributions (e.g., UNIF or TRIA) using OPS and parameter ranges
--millions?
--can do in pieces, changing random seed for each piece
--export .vdf(s) to tab-delimited file(s)

Yellow box: unique to MC

Orange box: unique to MCMC

Sort MC runs & create N-QPS
--by avg. MAEM, and keep those below cutoff
--then by max of MAEMs and keep only those below cutoff, leaving N sets; call this N-QPS

Import SVS into Excel & create M-SVS
--can select a random sample of desired size M from the SVS
--to create M-SVS sample

Determine cutoffs for a "qualifying parameter set" (QPS)
--max value of avg. MAEM
--max value for max of MAEMs

Save sensitivity (MC or MCMC) parameters as a tab-delimited txt file
--use N-QPS or M-SVS
- select the columns with parameter values
--to use for file driven sensitivity runs in Vensim

Compute/graph stats by parameter
--min, mean, max, conf. int. for each parameter
--compare to OPS and uncertainty limits
--Plot histograms to see shape of distribution

Create summary table(s) or tab(s)
--could provide confidence intervals (Cis) for the each outcome at baseline & for each alt. scenario
--also Cis for deltas and % deltas

Assemble a consolidated spreadsheet
--Upper left corner section: columns are baseline outcomes, rows for each parm set
--Below this is the same data for Alt. Scenario 1 (AS1)
--Below that, AS2, etc.
--In rows further to the right differences are calculated against baseline, cell by cell
--Even further right could be cells with % differences

Make file driven outcome sensitivity runs for baseline and alternative scenarios
--using N-QPS or M-SVS
--results (for each parameter set) are primary outcomes, and difference between alt. scenario and baseline.

Run file-driven *Trajectory* Sensitivity Run for Baseline
--using NQPS or MSVS
--**saving outcome trajectories by time period (e.g. yr.)**
--creates a set of N outcome trajectories for each outcome

Compute/graph baseline stats by primary outcome
--incl. conf. intervals for outcomes over time
--e.g., box and whisker plots over the time trajectory, by period

Specify alternative scenarios of interest
--e.g. new configuration(s)
--or, policy changes specified as sig. shifts in parameter values or other changes

Specify primary outcomes
--e.g. total costs, net performance, etc.

*Using Vensim model analysis features (other SD software packages may have similar capabilities)

Figure 1. Process for addressing SD model uncertainty

At this point, the user may elect to use conventional Monte Carlo (MC) or Markov Chain Monte Carlo (MCMC). MC is discussed first.

Make very large sensitivity Monte Carlo (MC) run is another key step of the process. It employs Vensim's sensitivity feature to make a very large number of model runs, millions if there are many uncertain parameters. Using this feature requires the user to specify how many runs to make, the seed to start with, the type of sampling (e.g., multivariate, Latin hypercube, etc.), and which parameters to vary and how. We used multi-variate, which is a totally random search process. Latin hypercube strives to cover a large parameter space more efficiently. Both may be valid choices. We experimented with uniform, but settled on Triangular with the mode specified as the value from the OPS. Since we are focused on overall error, we changed the SAVEPER to be the length of the run. This keeps the output file size manageable. We also specified some additional variables to be saved (all the parameters being varied are automatically saved). These were the "avgMAEM" and "maxMAEM" for the run.

Note that when >1M runs are needed, we found it practical to make one million at a time and to change the seed for each run. At the end of a sensitivity run, the .vdf file is exported to a tab-delimited file for further analysis.

Next, we ***Determine cutoffs for a "qualified parameter set (QPS)*** which involves determining how well-fitting a parameter set needs to be in order for it to be kept in the sample. The vast majority of the MC runs will not be well-fitting like the OPS. One rational approach would be to accept parameter sets that performed nearly as well as the OPS, say, for example average MAEM < .12 and max MAEM < .25.

Sort MC runs and create N-QPS employs Excel's data/import external data from file to read in the MC results. The result will be M columns x N rows, where M is the number parameters being varied + K, and N is the number of runs. K is the number of saved variables times the number of saved times per variable, which could be one, two (start time and end time), or more. The results are then sorted by avgMAEM and all rows > avgMAEM cutoff are discarded. The remainder is sorted by maxMAEM and all rows > maxMAEM are discarded. This will likely leave a very tiny fraction of runs, perhaps a few hundred out of a million. These N runs are the qualified parameter sets (N-QPS). ***Save sensitivity (MC or MCMC) parameters as a tab-delimited txt file*** creates the file needed for file-driven sensitivity runs.

Within Excel, the user can proceed to ***Compute/graph stats by parameter*** to see what the values in N-QPS file look like for each parameter. For example, to see if the entire range of possible values for a given parameter represented in N-QPS, and if the mean of this sample is near the value of the parameter in the OPS. Also to determine the confidence interval of the estimate for the mean of the parameter based on this sample and examine the distribution (shape) of the sample? Does this information raise any red flags with respect to the OPS or N-QPS? Or, does this information indicate that the N-QPS may be representative of the entire parameter space?

Moving on, the N-QPS file that can be used in two primary ways. One is the ***Run file-driven Trajectory Sensitivity Run for Baseline***, for which we change the SAVEPER to a useful time-period such as by year in order to create a series of N trajectories for each outcome. The result is exported to a tab-delimited file that can be imported into Excel to ***Compute/graph baseline stats by primary outcome,*** including uncertainty intervals for the model calculated outcome at each time period, since there is now a sample of N model-calculated values at each time point. Excel can also be used to create box and whisker plots at each time point, with or without outliers. However, we found it more expedient to make these plots using Python.

For the other primary use of the N-QPS file (or the M-SVS file, which is explained later), we first need to: ***Specify primary high level outcomes of interest*** such as total cost and net performance. The previous sensitivity run was focused on behavior over time for all outcome time series, but to compare alternative scenarios in an overall sense, a few key end-of-run metrics are needed. And, we need to ***Specify alternative scenarios of interest;*** these might different configurations or policy changes that could be implemented in the model as switches that are used in conjunction with "magnitude of impact" parameters and timing parameters linked to specific model constants.

Using the N-QPS or M-SVS, we change the SAVPER back to the End of Run and then ***Make file driven sensitivity runs for baseline and each alt. scenario.*** The results, for the baseline and for each scenario are matched pairs of data points (where all of the uncertain parameter values are the same for baseline and the alternative). This means that the distributions of the differences can be used to determine in a statistically valid fashion the confidence interval estimates for the mean of the difference by outcome between baseline and the alternative that can be attributed to parameter uncertainty.

We suggest to ***Assemble a consolidated spreadsheet*** to do these analyses. In the upper left corner, is O x N, where O is the number of overall outcomes and N is the number of runs. Note that the raw sensitivity results file has columns for all the parameters as well as columns for each outcome. You will be selecting and copying into the consolidated sheet just the end columns for each outcome. The data for Alternative 1 will be placed below the Baseline results; Alternative 2 below Alternative 1, etc. Then, starting at the first row of Alternative 1, columns will be added that compute the differences between the Alternative 1 numbers and baseline number, one column for each outcome. Similarly for Alternative 2 vs. baseline, and so forth. Columns for percentage differences can also be added.

Finally, ***Create summary table(s) or tab(s***) is used to provide the results of relevant calculations, such as means and their confidence intervals for each outcome at baseline and for each alternative. Also for the differences by outcome between baseline and Alternative 1, baseline and Alternative 2, etc. Similarly for the percentage differences.

A very different approach for creating a set of runs to be used to evaluate the impacts of parameter uncertainty is to use the Markov Chain Monte Carlo (MCMC) approach. This begins with ***Estimate Optimal Outcome Weights.*** First, one adds to the model a weight variable for each outcome instead of specifying the weights numerically as in the MC process, and then one specifies the objective function, which is similar to the prior approach except that the weights are variables within the model rather than numeric values. One also needs to specify the search range for outcome weight constants. The search ranges for the uncertain parameters can be the same as for the MC approach. Or, these ranges could be broader than those used with the prior approach. No sampling distribution such as Uniform or Triangular is specified because the search of parameter space is guided by a heuristic not random sampling. Guidance regarding how broad to set the ranges for MCMC varies by expert. Informally we've heard the comment that very broad ranges will to give the algorithm "room to work." Others have suggested, again, informally, that ranges should not include implausible values. Both comments are sensible, suggesting that a careful study of this question is needed. To proceed, first a Powell search is done that optimizes the uncertain parameters and the weights simultaneously. In our experience, allowing larger ranges may allow the algorithm to find an optimum that achieves slightly lower avgMAEM, but higher maxMAEM. The reason is that some of the individual outcome are "sacrificed" to achieve lower overall error. However, these results may be less realistic.

Next, is to ***Conduct MCMC search& generate SVS***, which, like MC, varies the uncertain parameters over the specified ranges, but uses in the objective function the outcome weights that were found in previous step. The Markov Chain-driven search algorithm is designed to create a statistically valid sample. It can be quite large and contain many duplicates (which apparently helps to assure a sample with the correct properties). Model fitness may be further improved, but be less plausible, as mentioned earlier. Results are then exported as a tab-delimited file.

One then must ***Import results into Excel***, and, if the population of runs produced by the MCMC search is excessively large, one can select a random sample of desired size M from the SVS. This subsample of size M, which is still a statistically valid sample (M-SVS), is comparable to the N-QPS set of runs created by the conventional MC process described earlier. One can proceed to computing stats by parameter using the M-SVS and/or use it to run file-driven sensitivity runs, analyze alternative scenarios, etc.

4. Results, an illustration

To illustrate the process and results, the MC method was applied to an SD model of significant complexity referred to as the Opioid Epidemic Model that explores the opioid epidemic in the United States from 1990 to 2030 (Homer and Wakeland 2020, Homer 2020). SD has been applied to drug abuse and other areas of public health and social policy (Levin et al 1975, Homer 1993, Homer and Hirsch 2006, Homer 2012, Wakeland et al 2013, Wakeland et al 2015, Homer 2017). The current model adopts the basic scientific approach and some of the same elements as these forerunner models, but the current model was completely redeveloped to address current needs.

A committee of the National Academies of Sciences/Engineering/Medicine considered the complexities of the opioid epidemic and stated that for informed decision making “a true systems model, not just simple statistics” was needed because “decisions made about complex systems with endogenous feedback can be myopic in the absence of a formal model” (Bonnie et al 2017). The NAS cited the system dynamics model by Wakeland et al (2015) as an example a true systems model.

Best practices for model development, testing, and reporting are well documented (Homer 1996, Sterman 2000, Richardson 2011, Rahmandad and Sterman 2012). The model building process for the Opioid Epidemic Model involved amassing evidence from many different sources and developing a dynamic structure that reproduces historical trends and is sufficiently parsimonious to allow explanation. Uncertainties are addressed through sensitivity testing and the multivariate Monte Carlo testing described previously that allows results to be reported with rigorous uncertainty intervals.

A complete list of the approximately 340 interacting model equations (including about 80 input constants, 17 time series for input and validation, and 240 output variables including 16 stock variables) is available in Homer (2020). Homer (2020) also provides stock and flow diagrams for each sector of the model and describes how model constant and time series were estimated, including data sources. The model time horizon is 1990-2030. Figure 1 shows the basic model structure.

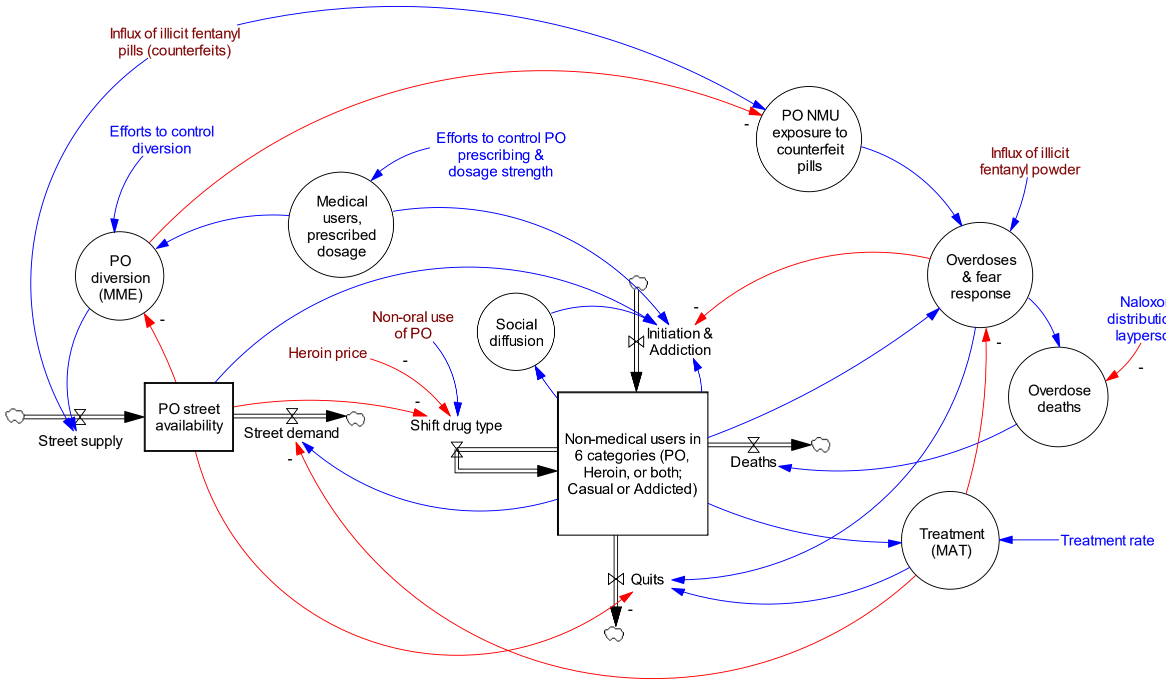
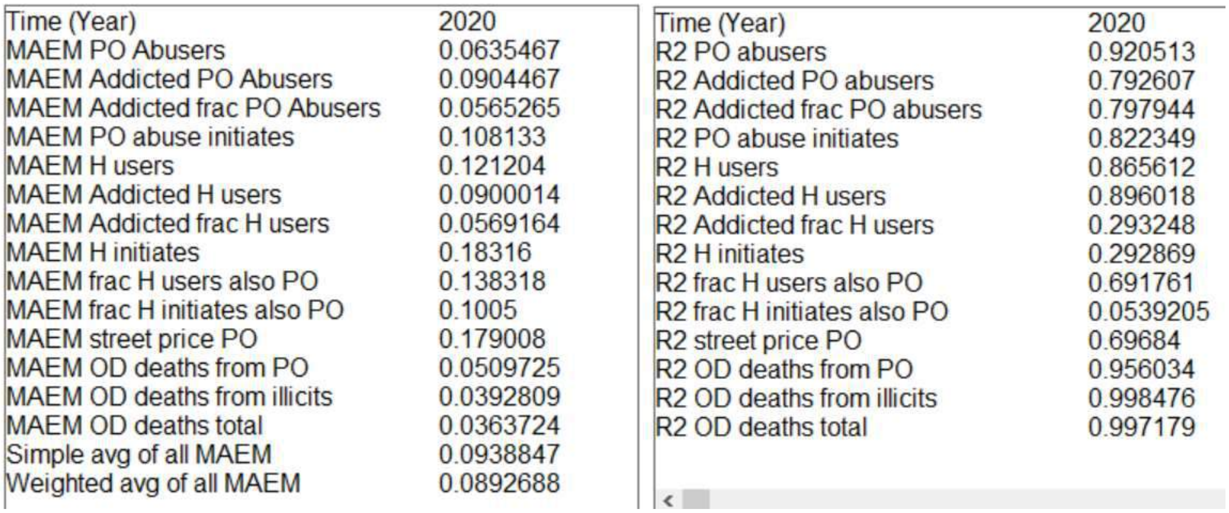

Influx of illicit fentanyl pills (counterfeits)
Efforts to control diversion
Efforts to control PO prescribing & dosage strength
PO NMU exposure to counterfeit pills
Influx of illicit fentanyl powder
Medical users, prescribed dosage
PO diversion (MME)
Overdoses & fear response
Naloxone distribution to laypersons
Social diffusion
Non-oral use of PO
Heroin price
Initiation & Addiction
Overdose deaths
PO street availability
Street supply
Street demand
Shift drug type
Non-medical users in 6 categories (PO, Heroin, or both; Casual or Addicted)
Deaths
Treatment (MAT)
Treatment rate
Quits

Figure 2. Model diagram showing high level stocks, flows, and primary feedback structures. Red lines are links with negative valence, and blue lines are links with positive valence

To apply the parameter uncertainty MC analysis method, error metrics are created, in this case MAEM (mean absolute error over mean) for the 12 outcome time series including Total opioid addicts, Opioid overdoses seen in ED, and Total opioid OD deaths. We used the Vensim SSTATS macro (author unknown) attached herewith as Appendix 1. This code is inserted via text editor (we used Notepad) at the top of the model file starting at the second line. Appendix 2 shows the model code for calculating the statistics, which is also inserted into the model file via text editor, typically after the "Control" block, which is located after the user-defined model variables/constants section. We utilized an Excel workbook that includes a worksheet that has Time in Row 1 and RBP time series data in the rows below. In some cases, there was data only for some of the historical years. The SSTATS macro calculations are designed to handle this correctly. An example Vensim equation for reading one these data time series is "GET XLS DATA('model RBP data.xlsx','RBP','1','B2')"

Another step to get the model ready for model analysis is to add a custom graphs file via the Control Panel and add custom tables that display the variables calculated by the SSTATS macro (R2, MAPE, MAEM, RMSE, Um, Us, Uc, and Count), one table for each Outcome variable. I/O Objects were added to a model view to display the custom tables. Figure 3 shows a sample of these statistical parameter displays created using the SSTATS macro.



Time (Year)	2020
MAEM PO Abusers	0.0635467
MAEM Addicted PO Abusers	0.0904467
MAEM Addicted frac PO Abusers	0.0565265
MAEM PO abuse initiates	0.108133
MAEM H users	0.121204
MAEM Addicted H users	0.0900014
MAEM Addicted frac H users	0.0569164
MAEM H initiates	0.18316
MAEM frac H users also PO	0.138318
MAEM frac H initiates also PO	0.1005
MAEM street price PO	0.179008
MAEM OD deaths from PO	0.0509725
MAEM OD deaths from illicits	0.0392809
MAEM OD deaths total	0.0363724
Simple avg of all MAEM	0.0938847
Weighted avg of all MAEM	0.0892688

Time (Year)	2020
R2 PO abusers	0.920513
R2 Addicted PO abusers	0.792607
R2 Addicted frac PO abusers	0.797944
R2 PO abuse initiates	0.822349
R2 H users	0.865612
R2 Addicted H users	0.896018
R2 Addicted frac H users	0.293248
R2 H initiates	0.292869
R2 frac H users also PO	0.691761
R2 frac H initiates also PO	0.0539205
R2 street price PO	0.69684
R2 OD deaths from PO	0.956034
R2 OD deaths from illicits	0.998476
R2 OD deaths total	0.997179

Figure 3: Selected SSTATS displays (MAEM and R-squared) for outcome variables

Besides the weighted average MAEM of 8.9%, it is clear that OD death-related trajectories are reproduced more accurately than for example Heroin-related trajectories. The STATS bear this out: the MAEM for OD deaths total is 3.6%, whereas for H initiates it is 18%.

Before using optimization to estimate uncertain parameter values, the user must specify an appropriate range for each parameter to be used by the Powell search algorithm. Appendix 3 provides an example Vensim optimization control file (.voc) which specifies the uncertain parameters to be varied. Table 1 shows the first few rows of the model parameter spreadsheet by parameter, including the min and max values. While many of the 80 model parameters have some degree of empirical support, many of these, along with those without empirical support, were included in the optimization process. The ranges for parameters with empirical support were smaller.

Table 1: Partial model parameter spreadsheet showing parameter name, units, value, source (or optimized), and the min and max values used during optimization runs

Parameter	**units**	**Value**	**Sources**	min value	max value
ESTIMATED CONSTANTS					
Addicted frac of H users initial	fraction	0.65	Optimized; our NSDUH analysis shows 60.8% 2000, 61.1% 2005.	0.6	0.7
Addicted frac of PONHA initial	fraction	0.123	Optimized; our NSDUH analysis shows 11.4% 2000, 14.2% 2005.	0.1	0.15
Addicted H user OD death rate initial	1/year	0.010	Optimized	0.005	0.015
Addicted H user quit rate initial	1/year	0.138	Optimized	0.07	0.21
Addicted opioid abuser misc death rate	1/year	0.0045	Ray et al 2016 gives mortality hazard ratio of 1.94 vs general popn for "high dose users" (>60 mg ME). Multiply by general popn: average of NVSR death rates for [age 25-34, 35-44, 45-54] = .0023 for 2000-2010 x 1.94 = .0045.		
Addicted PONHA move to heroin rate initial	1/year	0.021	Optimized	0.01	0.03
Addicted PONHA OD death rate initial	1/year	0.0059	Optimized	0.004	0.007
Addicted PONHA quit rate initial	1/year	0.149	Optimized	0.08	0.22

In addition, weights must be specified for the objective function, informed by the relative magnitudes and variances of the outcome variables. In this example, weights were set so that metrics with solid empirical data were given more weight than those for which empirical data was scant or less reliable.

The optimization runs using the CG (calibration Gaussian) option ran for several minutes on a laptop computer. A blend of modeler judgment and optimization results was used iteratively to select the values for the “optimized” parameter set. The MEAMs for many of the outcomes were reduced compared to a purely manual calibration process. The weighted (based on the amount of data available for each outcome) average of the MAEMs was 8.9%, with the largest individual MAEM being 17.9%.

Figure 4 shows the trajectories for two example outcomes calculated by the model using the optimized parameter set compared the available RBP data for these metrics. The MAEMs for the two outcomes in Figure 4 are 9% and 4%.

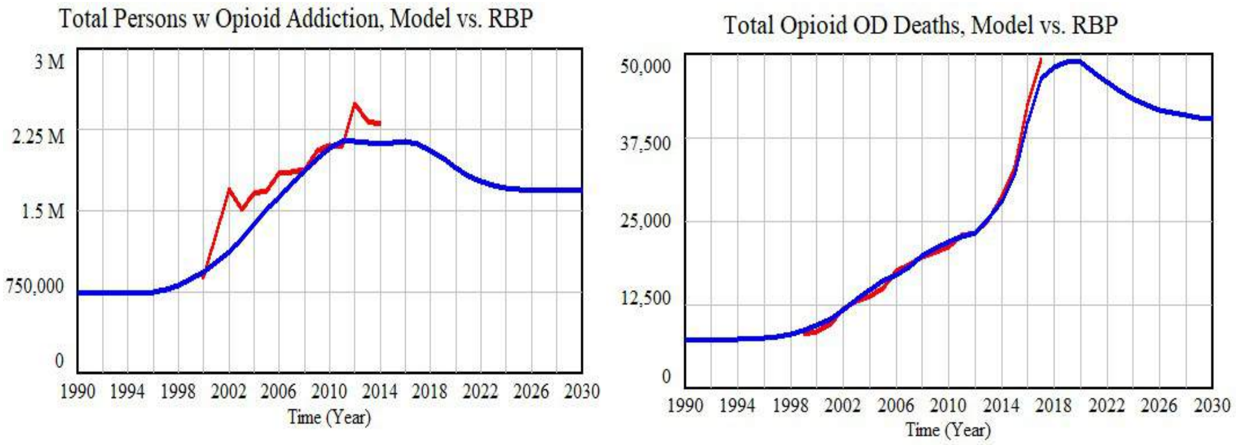


Figure 4: Model fit to key outcomes using the optimized parameter set

The calculated Persons with Addiction trajectory is somewhat problematic, whereas the OD Deaths trajectory matches very well. One possible reason could be the amount of data available, meaning that the standardized errors for the first variable would carry somewhat less weight than the second one. And, one must ask ask if the fluctuations in an outcome around its trend are meaningful and consistent with theory, or, perhaps due to measurement uncertainty?

Making a large Monte Carlo (MC) run is easy using the sensitivity feature in Vensim, which uses or creates a .vsc file. One option for exploring parameter space could be to use Uniform distributions between thoughtfully chosen minima and maxima. These values could be the same as those used during calibration or perhaps narrowed somewhat. Another option could be to use a Triangular distribution with the optimum value as the mode, which would increase samples near the optima. For this illustration, Triangular distributions were used. The user also need to specify a .lst file that lists which variables are to be saved for each run. Sensitivity runs automatically save the varied parameters used for each run, and in addition, it is useful to know the maxMAEM and average MAEM (to be used to select which runs are qualified). Since we don't need to know the time trajectories, we set the model SAVEPER to 40. This keeps the output file modest in size. Appendix 4 shows the .vsc and .lst files.

The simulation was set to create 1M runs, which took about seven hours on a laptop. Ten such runs were made, changing the random for each run, creating a total sample of 10M runs. Each resulting .vdf (vensim data file) was exported (via the model menu) to a tab-delimited file and read into Excel using data/get external data/from text (switch to "all files" to see .tab files). This was sorted by weighted average of MAEMs. The weighted average MAEM was below .11 for 300-600 runs in each of the 1M runs, and these rows were kept. The other rows were deleted. The file was then sorted by max MAEM, and 100-130 runs were less than .20 and kept. The rest were deleted. These ten spreadsheets were combined, yielding a sample of 1119 runs that accomplish <11% weighted average error and <20% max error. Rows were added below the sample to calculate useful stats about the sample, regarding both the distributions of the input parameters and the how well each run performed. Table 2 shows the first and last few rows of the spreadsheet.

Table 2: First and last few rows and columns of the file used to create the N-QPS

Simulation #	Addicted frac of H users initial	Addicted frac of PONHA initial	Addicted H user OD death rate initial	...	maxofMAEMs	Simple avg of all MAEM	Weighted avg of all MAEM
681526	0.630252	0.126897	0.012121		0.199446	0.100208	0.095847
376905	0.691254	0.118593	0.012554		0.197496	0.101902	0.09691
131761	0.645974	0.118036	0.009821		0.19669	0.105488	0.097978
67350	0.684089	0.117152	0.007821		0.171332	0.101261	0.098184
726864	0.650072	0.124617	0.010844		0.183783	0.101797	0.098296
736791	0.653768	0.123594	0.010931		0.190377	0.115002	0.109998
358518	0.688724	0.122379	0.009984		0.184925	0.114706	0.109999
MIN of sims	0.6012	0.1003	0.005936		0.1612	0.1002	0.0958
MAX of sims	0.6998	0.1488	0.014533		0.2000	0.1191	0.1100
MIN allowed	0.6	0.1	0.005				
MAX allowed	0.7	0.15	0.015				
Optimized value	0.65	0.123	0.010		*0.1795*	*0.0994*	*0.0935*
Sample Mean value	0.6487	0.1247	0.010454				
Sample std. dev.	0.0204	0.0100	0.001473				

After saving the spreadsheet, one can delete Col A, the metrics columns at the right, and the rows at the bottom used to check its properties. The file is then saved as a .txt (tab-delimited file); this is the N-QPS. The N-QPS is used to run file-driven sensitivity runs in two ways. The first way is to make a sensitivity run to obtain outcome trajectories for the different well-fitting runs and examine whether they are tightly grouped or spread out. The second usage is to capture final outcome values for key system metrics for the base case and for alternative configurations or policies being tested.

But first, one must examine the properties of the parameter samples contained in the N-QPS. The rows at the bottom of Figure 2 can show their range compared to the allowed (min to max) range, their mean value, and other statistics. This is easy, so there is no need to demonstrate this. One can also quickly create histogram plots. Figure 5 shows examples. The parameter shown on the left was limited to be in the .01 to .03 range, so none of the well-fitting runs was below the lower limit, but some were at the upper limit, so nearly all of the allowed range is included in the N-QPS. The mode is about .0235 and, which is slightly higher than its value of .021

in the optimized parameter set. The parameter on the right was limited to be in the range of 2000 to 5000, and its values in the N-QPS were inside well inside of these limits (2092 to 4566). The mode is 3330, very slightly higher than its value of 3200 in the optimized parameter set. One can examine all of the parameter in this fashion to build confidence in the sample.

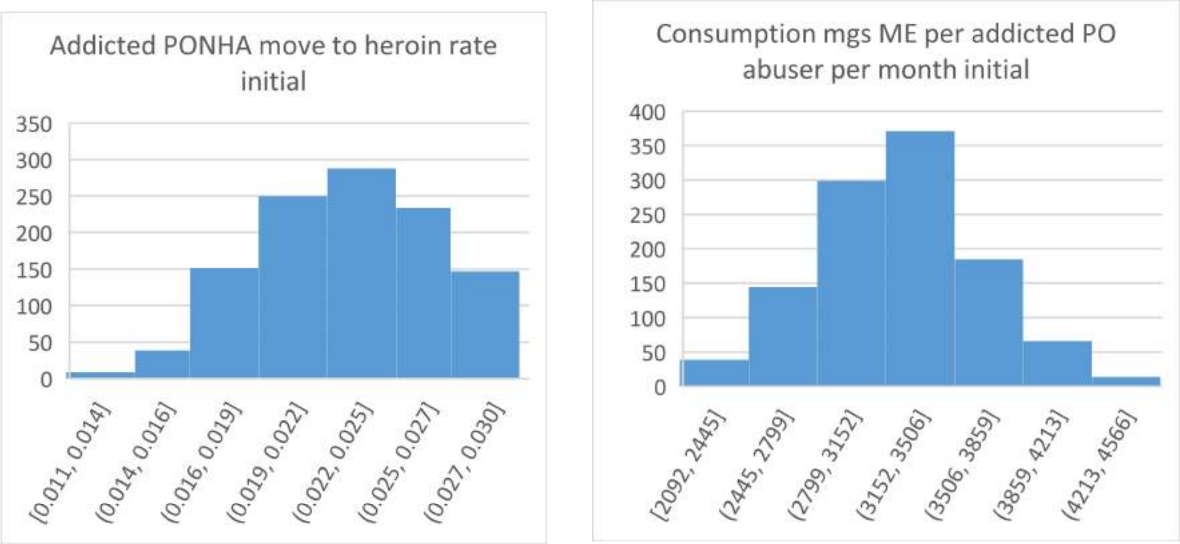


Figure 5: Example histograms for the uncertain parameters in the N-QPS

To put the N-QPS sample to use, file-driven sensitivity runs are performed. The SAVEPER is changed to 40, and the Sensitivity tool is selected. Previous settings are cleared, and the Type of sensitivity run is changed to “File.” The N-QPS text file is selected as the file. Next a save list is created. The previous save list (with the statistical variables) is cleared and three primary outcome variables were entered, Total opioid addicts, Opioid overdoses seen in ED, and Total opioid OD deaths. This run took just a minute to complete, and the .vdf file it produced was exported via Vensim and imported into Excel.

The first 50 columns of this trajectory spreadsheet provide the run number (1-1119) and the values of the ~49 uncertain parameter values used for each specific run. The remaining 41 columns are values from time 1990 to 2030 for the three outcome variables. Amazingly, if one selects the 1119 rows and 41 columns for one of the outcome variables, and then requests a Box and Whisker plot, it works, although Excel does not know the time periods. Figure 6 shows the trajectory under uncertainty for the Overdoses Seen at ED outcome metric using a box and whiskers plot. One can also use the data in the spreadsheet to calculate useful sample statistics.

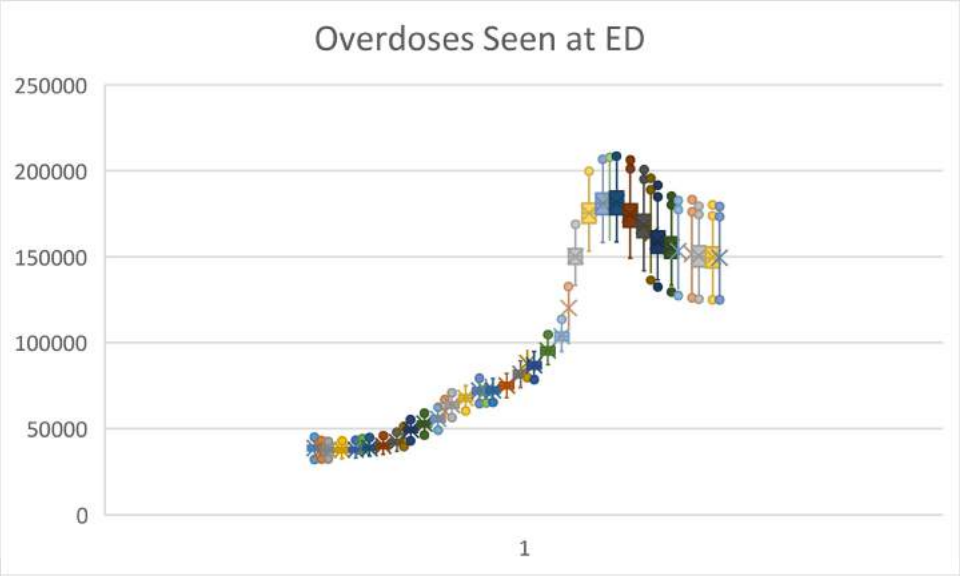
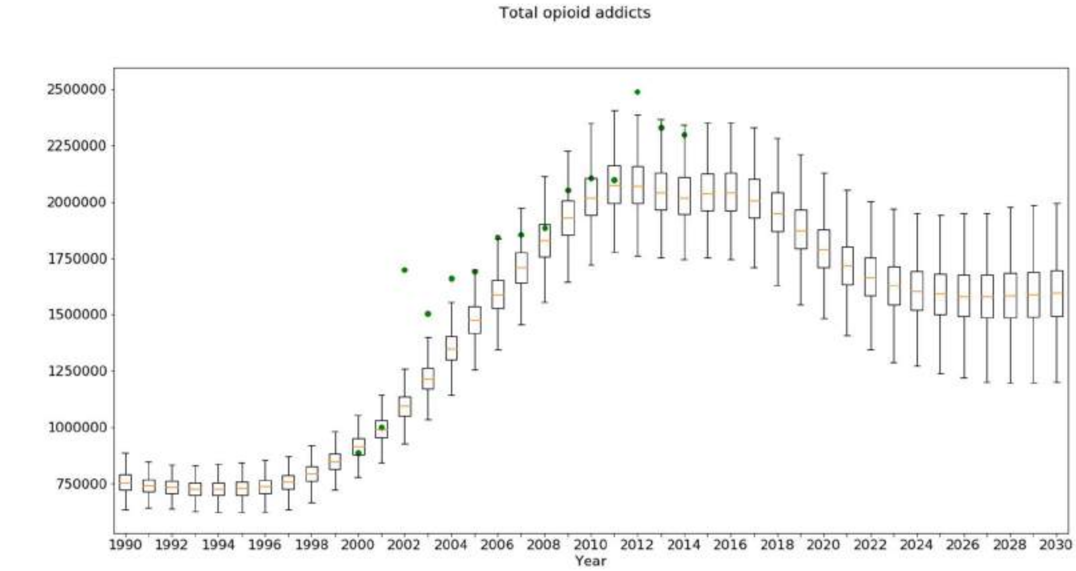
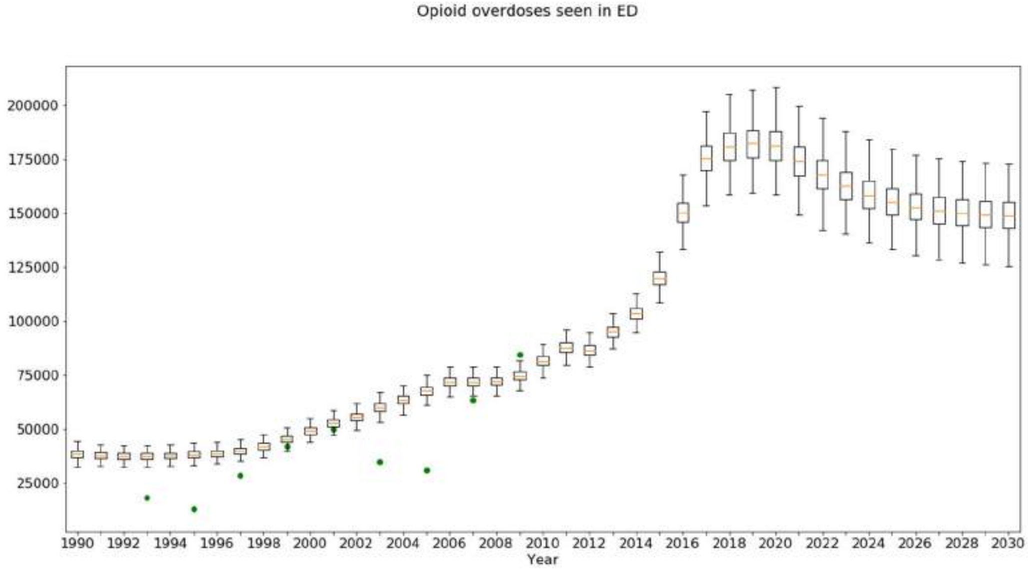


Figure 6: Box and whisker time series plot using Excel

There is no easy way using Excel to superimpose the RBP on this plot, which would be very helpful. Python can be used create better formatted and more complex charts as needed. Figure 7 shows the plots produced using Python for the three outcome variables which show the uncertainty interval trajectories overlaid with the RBP data.

To prepare the file needed by Python, a version of the trajectory spreadsheet was created, named “Outcome Trajectory Data for Python Plotting” (OTDPP), in which the columns for run number and the parameter values were deleted. Also, two rows were added below the first row. Row 2 contains the actual data to be overlaid. For each outcome variable, the appropriate row of the RBP spreadsheet is copied above its corresponding section in OTDPP. Since the RBP data spreadsheet contains up to 30 values of historical data for each outcome trajectory, and since the data sections for each outcome in OTDPP is oriented horizontally, it is a simple copying process, even for several outcomes. Row 3 in OTDPP provides the time values, which is easily created in Excel or can be copied from RBP spreadsheet (41 cells copied once and pasted N times in the OTDPP, where N is the number of outcome variable, 3 for this example).


Total opioid addicts
2500000
2250000
2000000
1750000
1500000
1250000
1000000
750000
1990
1992
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018
2020
2022
2024
2026
2028
2030
Year


Opioid overdoses seen in ED
200000
175000
150000
125000
100000
75000
50000
25000
1990
1992
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018
2020
2022
2024
2026
2028
2030
Year

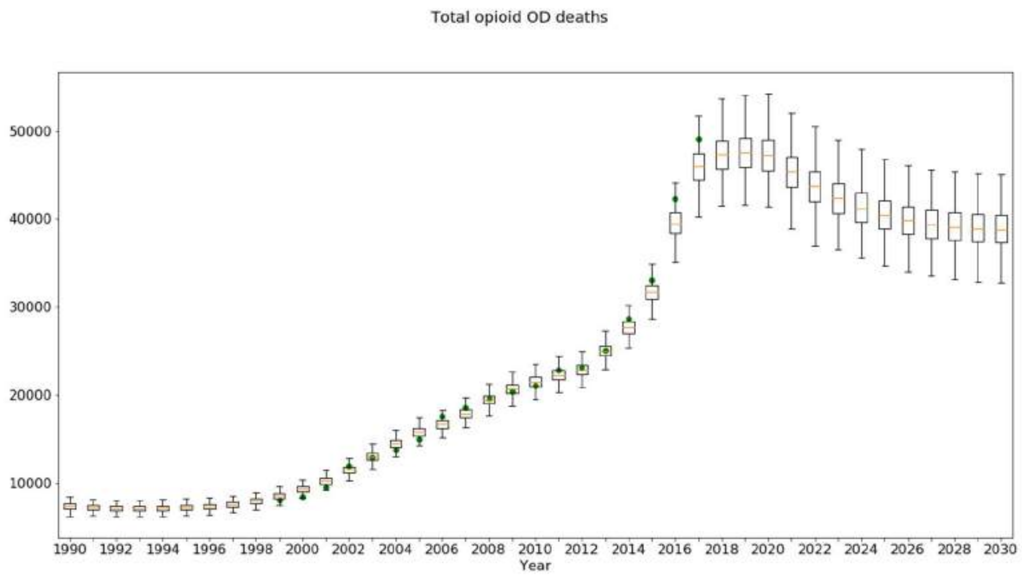


Figure 7: Outcome trajectories with uncertainty Intervals based on N-QPS, with actual data (dots)

Another use of the N-QPS is to compare a baseline case with policy runs. For example, how much impact might a policy of reducing opioid prescription strength have with respect to the most important outcome indicator(s)? That indicator could be the number of people with opioid addition, opioid overdose events, or deaths. These metrics are saved for each of the parameter sets in the N-QPS, for the baseline condition and for each alternative. And the net change in each metric is calculated, run by run, in both absolute and percentage terms

To demonstrate, the Opioid Epidemic Model was run using the file-driven sensitivity tool reading in the 1119-QPS parameter sets. The final values for the three key metrics were saved for each of the 1119 runs, and a spreadsheet was created to compute the differences in outcome metrics, by run. This spreadsheet has 1119 rows and 48 columns. The first three columns are the final baseline values for each of the three metrics. The next nine are values for the first policy run for each metric, plus three columns that calculate the change, and three more that calculate the percentage change. The next 36 columns are four more groups of nine each for the other four policy alternatives. A row is added at the bottom to compute the mean percentage change in each outcomes due to each policy. Two more rows compute minimum change in each metric and the maximum change in each metric. Table 3 shows a summary of the parameter uncertainty analysis of the impacts of the various policies.

Table 3. Impact of parameter uncertainty on the results of policy changes

		Optimized parameter set		**QPS 1119 MC Result**			**QPS 1119 MC, % change vs baseline**		
OUTCOME MEASURE	**TEST CONDITION**	**Result**	**% change vs Baseline**	**Mean**	***Range (min, max)***		**Mean %Δ**	***Range (min, max)***	
Persons with OUD (thou)	Baseline	1,694		1,593	*1,111*	*2,084*			
	Avg MME dose down 20%	1,510	-10.9%	1,416	*1,035*	*1,823*	-11.1%	-25.7%	-3.4%
	Diversion Control 30%	1,428	-15.7%	1,339	*1,007*	*1,716*	-15.9%	-37.4%	-4.6%
	Treatment rate 65% (from 45%)	1,713	1.1%	1,585	*1,054*	*2,130*	-0.5%	-9.0%	5.0%
	Naloxone lay use 20% (from 4%)	1,728	2.0%	1,624	*1,150*	*2,111*	1.9%	1.3%	2.3%
	All 4 policies combined	1,285	-24.1%	1,189	*905*	*1,560*	-25.4%	-60.2%	-6.5%
Overdoses seen at ED	Baseline	154,710		149,450	*124,745*	*179,297*			
	Avg MME dose down 20%	152,686	-1.3%	145,473	*118,491*	*176,363*	-2.7%	-8.2%	3.8%
	Diversion Control 30%	153,076	-1.1%	144,442	*116,080*	*174,701*	-3.4%	-11.6%	6.0%
	Treatment rate 65% (from 45%)	150,095	-3.0%	143,994	*118,277*	*171,260*	-3.7%	-11.3%	-0.3%
	Naloxone lay use 20% (from 4%)	159,228	2.9%	154,143	*128,151*	*186,870*	3.1%	2.2%	5.1%
	All 4 policies combined	148,395	-4.1%	138,581	*110,651*	*167,730*	-7.3%	-19.6%	6.1%
Overdose deaths	Baseline	40,323		38,952	*32,513*	*46,732*			
	Avg MME dose down 20%	39,796	-1.3%	37,916	*30,883*	*45,967*	-2.7%	-8.2%	0.6%
	Diversion Control 30%	39,897	-1.1%	37,647	*30,255*	*45,534*	-3.4%	-11.6%	6.0%
	Treatment rate 65% (from 45%)	39,120	-3.0%	37,530	*30,827*	*44,637*	-3.7%	-11.3%	-0.3%
	Naloxone lay use 20% (from 4%)	35,302	-12.5%	34,174	*28,412*	*41,434*	-12.3%	-18.6%	-8.1%
	All 4 policies combined	32,900	-18.4%	30,724	*24,532*	*31,186*	-21.1%	-36.4%	-6.9%

As one can see by examining the percentage change vs. Baseline column with the QPS % change vs. baseline columns at the right, even though the mean of the QPS-driven change is fairly close to the results using the optimized parameter set, seeing the uncertainly interval provides a richer interpretation. Notice that for the outcome “Persons with OUD” and the “increase treatment” policy, one sees a modest unfavorable impact for the optimized parameter set and a modest favorable impact for the QPS. The QPS sample interval contains zero, so this policy should perhaps be considered not to have an appreciable impact on persons with OUD. For “Overdoses seen at ED,” and “all 4 policies combined,” a beneficial outcome is predicted by both, but the uncertainty also includes zero, indicating that the net effect of all 4 policies on overdose events could be neutral. For “OD deaths,” and the “diversion control policy,” the uncertainty interval also includes zero, suggesting that for a significant number of the qualified parameters sets, the impact is unfavorable. This is likely due to persons switching to more dangerous drugs. This hypothesis can be examined directly by examining the results, obviating the need for speculation.

Discussion

This section uses a narrative adapted from a similar, but proprietary, model development and application that employed the uncertainty methods discussed in this paper. The narrative began by explaining that it accompanies a series of tables and graphs that lay out the assumptions and results of a complex system dynamics model, and that most of the graphs in the package are time plots covering a specific multi-year time period. It specified that the purpose of the model was to determine the extent to which specific events over time contributed to negative outcomes.

Part one: baseline modeling and parameter estimation

The model was developed and tested using the Vensim^{TM} simulation software (version 7.2 Professional dated 2017). Model development was carried out according to established practices of system dynamics, which include information gathering, model formulation, model testing, and revising of model assumptions as necessary to improve model realism and usefulness. Information came from published studies as well as online data sources. During the process, the model progressed through many iterations.

The model contains hundreds of interacting equations, eighty input constants and time series, and over two hundred primary and secondary outputs. “Primary outputs” refers to a dozen variables for which historical time-series data was available that covered many aspects of the system. A few time series included all of the historical data, but many provided historical data at only a few time points, so that on average half of the historical period was available.

Many of the model’s input constants and time series were estimated with high confidence from empirical data, but half of them must be considered to have a wider range of uncertainty. The model’s fixed constants were listed in a table, and the input time series were presented as figures. These materials also included data sources and derivations (associated logic).

We used Vensim’s optimization feature (utilizing a standard Powell hill-climbing routine) to test the uncertain constants over their plausible ranges and found the best fit to the primary output data. The best-fit metric was the Mean Absolute Error as a percent of the Mean (MAEM), for each of the primary output variables and for all of them combined. From this testing came the "optimized" version of the model.

We also used Vensim's sensitivity testing feature (using standard Monte Carlo randomization) to perform millions of runs. In these runs, we allowed the uncertain constants to vary over their plausible (or allowed) ranges according to triangular probability distributions. We set each triangular peak at the constant's optimized value, the lower extreme at the minimum allowed value, and the upper extreme at the maximum allowed value.

From these runs we found N "qualifying parameter sets" (QPS) with a good fit to the primary output data, a fit comparable to that of the optimized run. In particular, we selected as QPS only those runs for which the maximum of the individual MAEMs for each of the primary outputs was no greater than XX% (slightly above value achieved by the optimized run), and where the average across the MAEMs was no greater than YY%. The average MAEM was ZZ% for the optimized run, and ZZ was somewhat less than YY.

The table listing the model's uncertain constants, along with relevant sources and notes, their optimized values, the min and max allowed values for each, was updated with summary statistics for the values of these constants in the QPS. These summary statistics included, for each constant, the QPS mean, the sample interval based on the QPS (the min value and max value), and the sample range as a percentage of the min-to-max allowed range. These summary statistics indicate whether the assumed triangular distributions are adequately wide enough to capture the universe of well-fitting parameter sets. If the QPS mean values strayed away from the optimized value, or if the sample range were close to 100% of the allowed range, it would indicate that the assumed distributions may be skewed or insufficiently wide.

In fact, only a handful of the uncertain constants has a QPS mean value that differed by more than X% from the optimized value (and only one [or none] by more than YY%), and all of the CI ranges were appropriately within the allowed range. <u>We concluded that the assumed distributions for the uncertain constants are appropriate for finding the universe of well-fitting parameter sets.</u>

Another table lists the MAEM values for the outcomes in the optimized run, including MAEMs for each of the primary outputs; for all data points combined; for the average of the outputs; and the maximum MAEM across all of the outputs. This table also lists MAEM summary statistics for the N QPS runs; those statistics include, for each of the primary output variables, the mean of the N QPS runs and its range (sample interval).

The optimized run provides a better fit to the data (a lower MAEM) than the mean of the N QPS runs for most of the outputs. Moreover, the average of the MAEMs (A%) is lower for the optimized run than the mean of average MAEMs for the N QPS runs (B%). On the other hand, the maximum of the MAEMs is a little greater for the optimized run (C%) than it is for the mean of the N QPS runs (D%).

From these results, we conclude that our Monte Carlo-based approach provided N qualifying parameter sets that yield a good fit to the data, nearly as good as that of the optimized run.

The good fits to history of the optimized run may be seen in several figures, where we show line graphs for the simulated trajectories for each of the outcomes superimposed on the historical data (vertical bars), along with data sources and derivations.

The good fits to history of the N QPS runs may be seen in additional figures which present box-and-whisker plots (see http://mathworld.wolfram.com/Box-and-WhiskerPlot.html) for the N QPS, one plot for each year over the modeled time period, superimposed on the historical data (green dots).

Part two: testing alternative situations

We defined several conditions representing alternatives to be tested using the optimized parameter set, as well as the N QPS. Each alternative represents a change to the system, often at a particular point in time. There changes are often parameterized (made controllable) by adding a switch or flag that controls alternative logic (invokes alternative logic when the switch is "on." Often, alternatives are simply parameter changes at particular points in time, with constants being added to the model to represent the magnitude of the change and the timing of the change.

We selected several outcome metrics for comparing alternatives against the baseline run. Some of these metrics are key cumulative (end of run) statistics for key results of interest. Others are the state of the system at the end of the time-period of interest. The results are presented in a series of tables, one for each outcome metric. These tables also describe what the alternative configuration represents. Each table presents the outcome metric for the baseline and for each of the alternatives. They also show the difference between each alternative outcome and the baseline outcome ("change vs baseline"), as well as the percentage difference ("% change vs baseline"). These results and calculations are shown for the optimized parameter set; and also for the QPS (summarized in terms of mean and the sample interval).

When changes against baseline have a negative sign, it means that the net impact of the alternative being tested was beneficial (reduced adverse outcomes). When these changes have a positive sign, it means the net impact of the alternative was harmful.

The final section of the report from which this documentation sample was extracted discussed what the results (described here in completely generic terms) actually mean in real context. Of course, this is exactly how any report of simulation results should end: explaining what all the modeling and analysis reveals about the system of interest that was previously unknown and very important to know!

In conclusion, it is possible to evaluate objectively the degree of uncertainty in model outcomes due to uncertainty in the parameters used in the model. This paper has provided a flowchart that summarizes two ways to do this, and demonstrates one of these approaches with a concrete example. A template for documenting results was also provided.

Acknowledgements

The authors gratefully acknowledge Kevin Stoltz who wrote the Python code for graphing confidence interval trajectories.

References

Bonnie RJ, Ford MA, Phillips JK, editors (2017). *Pain Management and the Opioid Epidemic: Balancing Societal and Individual Benefits and Risks of Prescription Opioid Use*. National Academies Press: Washington, DC.

Cheung SH, Beck JL. (2009). Bayesian Model Updating Using Hybrid Monte Carlo Simulation with Application to Structural Dynamic Models with Many Uncertain Parameters. *Journal of Engineering Mechanics* 135(4), 243–255.

Fiddaman T, Yeager L. (2015). Vensim Calibration and Markov Chain Monte Carlo. *Proc 33rd Int Conf Syst Dyn Soc*.

Helton JC, Johnson JD, Sallaberry CJ, Storlie CB. (2006). Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliab Eng Syst Saf* 91(10-11):1175-1209.

Homer JB (1993). A system dynamics model of national cocaine prevalence. *Sys Dyn Rev* 9(1):49-78.

Homer JB (1996). Why we iterate: scientific modeling in theory and practice. *Sys Dyn Rev* 12(1):1-19.

Homer JB (2012). *Models that matter: selected writings on system dynamics 1985-2010*. Grapeseed Press: Barrytown, NY.

Homer JB (2017). *More models that matter: system dynamics writings 2011-2017*. Grapeseed Press: Barrytown, NY.

Homer J (2020). Reference guide for the opioid epidemic simulation model (version 2u). February 2020. Available at: https://pdxscholar.library.pdx.edu/cgi/viewcontent.cgi?filename=0&article=1154&context=sysc_fac&type=additional

Homer J, Hirsch G (2006). System dynamics modeling for public health: background and opportunities. *Am J Public Health* 96(3):452-458.

Homer J, Wakeland W (2020). A Dynamic Model of the Opioid Drug Epidemic with Implications for Policy. *Am J Drug Alcohol Abuse* (under review).

Houghton J, Siegel M. (2015). Advanced data analytics for system dynamics models using PySD. *revolution 3*(4).

Houghton JA, Siegel M, Wirsch A, Moulton A, Madnick S, Goldsmith D. (2014). A survey of methods for data inclusion in System Dynamics models: methods, tools and applications. *Composite Information Systems Laboratory (CISL) Working Paper CISL* 3:2014-03.

Jeon C, Shin J (2014). Long-term renewable energy technology valuation using system dynamics and Monte Carlo simulation: Photovoltaic technology case. *Energy* 66:447-457.

Levin G, Roberts EB, Hirsch GB (1975). *The persistent poppy: a computer-aided search for heroin policy*. Ballinger: Cambridge, Mass.

Osgood N. (2013). Bayesian Parameter Estimation of System Dynamics Models Using Markov Chain Monte Carlo Methods: An Informal Introduction. *Proc 31st Int Conf Syst Dyn Soc*. #1391.

Osgood ND, Liu J. (2015). Combining Markov Chain Monte Carlo approaches and dynamic modeling. In: *Analytical Methods for Dynamic Modelers*, eds. Rahmandad H, Oliva R, Osgood ND. Cambridge MA: MIT Press; Chapter 5, pp. 125-169.

Rahmandad H, Sterman JD (2012). Reporting guidelines for system dynamics modeling. *Sys Dyn Rev* 8(3):251-261.

Richardson GP (2011). Reflections on the foundations of system dynamics. *Sys Dyn Rev* 27(3):219-243.

Sterman JD (2000). *Business dynamics: systems thinking and modeling for a complex worl*d. Irwin McGraw-Hill: Boston, MA.

Sterman JD et al. (2018). Does replacing coal with wood lower CO2 emissions? Dynamic lifecycle analysis of wood bioenergy? *Environ. Res. Lett.* Vol 13(1).

Vrugt JA, ter Braak CJF, Diks CGH, Robinson BA, Hyman JM, Higdon D. (2009). Accelerating Markov Chain Monte Carlo Simulation by Differential Evolution with Self-Adaptive Randomized Subspace Sampling. *International Journal of Nonlinear Sciences & Numerical Simulation* 10(3):271-288.

ter Braak CJF. (2006). A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces. *Stat Comput* 16:239–249.

Wakeland W, Hoarfrost M. (2005). The Case for Thoroughly Testing Complex System Dynamic Models. *Proc. 23rd Int Conf of the Syst Dyn Soc*.

Wakeland W, Nielsen A, Schmidt T, McCarty D, Webster L, Fitzgerald J, Haddox JD (2013). Modeling the impact of simulated educational interventions in the use and abuse of pharmaceutical opioids in the United States: a report on initial efforts. *Health Educ Behav* 40:74S-86S.

Wakeland W, Nielsen A, Geissert P (2015). Dynamic model of nonmedical opioid use trajectories and potential policy interventions. *Am J Drug Alcohol Abuse* 41(6):508-518.

Appendix 1: SSTATS Vensim Macro (need to give credit)

```
:MACRO: SSTATS(historical,simulated:R2,MAE,MAE over Mean,MAPE,RMSE,MSE,Um,Us,Uc,Count\
               )
SSTATS = residuals
        ~      simulated
        ~      Note that first argument (historical) must be data;
               second argument (simulation) can be data or a simulation,
               but if it is data, it will not be checked for existence.
               This could easily by changed by modifying the code below.
               Arguments following the : are outputs. They generate model
               variables that are visible in the listing on the Variable
               tab of the control panel, and can be used in equations
               and custom graphs/tables/reports. However, they cannot be
               made visible on diagrams.
        |

R2 = r*r
        ~      Dimensionless
        ~      Correlation coefficient squared
        |

MAE=
        ZIDZ(sum ae,Count)
        ~      simulated
        ~      mean absolute error
        |

MAE over Mean=
        ZIDZ(MAE,meanx)
        ~      Dimensionless
        ~      Mean Absolute Error as a fraction of the mean of the data
        |

MAPE = ZIDZ(Sum APE,Count)
        ~      Dimensionless
        ~      Mean Absolute Percent Error is reported as fraction rather than
               percentage (x100) for consistency with other the other metrics
        |

RMSE = SQRT(MSE)
        ~      simulated
        ~      Root Mean Square Error
        |

MSE = dif mea + dif var + dif cov
        ~      simulated*simulated
        ~      Mean Square Error. The addition of the three components
        |

Um = ZIDZ(dif mea,MSE)
        ~      Dimensionless
        ~      Bias inequality proportion
        |

Us = ZIDZ(dif var,MSE)
        ~      Dimensionless
        ~      Variance inequality proportion
        |

Uc = ZIDZ(dif cov,MSE)
        ~      Dimensionless
        ~      Covariance inequality proportion
        |

Count = INTEG(pick/dt,0)
        ~      Dimensionless
        ~      Counter for # of points
        |

residuals=
        IF THEN ELSE(pick, Yi - Xi, :NA:)
        ~      simulated
        ~       Errors
        |

r = MIN(1,ZIDZ(Mxy-(meanx*MeanY),Sx*Sy))
        ~       Dimensionless
        ~       Correlation coefficient. Calculated through the 'hand computation' formula.
                   Sterman (1984) pg. 63
        |

sum ae= INTEG (
        ABS(Xi - Yi)/dt,
                0)
        ~       simulated
        ~       Sum of Absolute Errors
        |

meanx = ZIDZ(Sum Xi,Count)
        ~       simulated
        ~       Mean of x (sum x)/n
        |

Sum APE = INTEG(ABS(ZIDZ(Xi-Yi,Yi))/dt,0)
        ~       Dimensionless
        ~       Sum of Absolute Percent Errors is reported as fraction rather than
                percentage (x100) for consistency with other the other metrics
        |

dif mea = (meanx-MeanY)*(meanx-MeanY)
        ~       simulated*simulated
        ~       Difference of Means (bias)
        |

dif var = (Sx-Sy)*(Sx-Sy)
        ~       simulated*simulated
        ~       Difference of variances
        |

dif cov = 2*Sx*Sy*(1-r)
        ~       simulated*simulated
        ~       Difference of covariances
        |

pick= IF THEN ELSE(Y = :NA: :OR: X = :NA:, 0, 1)
        ~       Dimensionless
        ~       Takes a value of one for every data point available, assuming the data are \
                available at intervals of Interval between the Start Time and End Time.
        |

dt = TIME STEP$
        ~       Time$
        ~               |

Yi = pick*Y
        ~       simulated
        ~       Sampled simulated variable
        |

Xi = pick*X
        ~       simulated
        ~       The historic data series
        |

Mxy = ZIDZ(SumXY,Count)
        ~       simulated*simulated
        ~       Mean of x*y (sum x*y)/n
        |

MeanY = ZIDZ(Sum Yi,Count)
        ~       simulated
        ~       Mean of y (sum y)/n
        |

Sx = SQRT(MAX(0,MX2-(meanx*meanx)))
```

```
        ~      simulated
        ~      Standard Deviation of x. Calculated using the 'hand computation'
               formula to calculate the standard deviation without prior knowledge of
               the mean.  Sterman (1984), pg. 64  MAX prevents spurious numerical
               problems from roundoff
        |

Sy = SQRT(MAX(0,MY2-(MeanY*MeanY)))
        ~      simulated
        ~      Standard Deviation of y. Calculated using the 'hand computation'
               formula to calculate the standard deviation without prior knowledge of
               the mean.  Sterman (1984), pg. 64
        |

Sum Xi = INTEG(Xi/dt,0)
        ~      simulated
        ~      Sum of x's (simulated)
        |

Y = simulated
        ~      simulated
        ~      The simulated data series
        |

X :RAW: := historical
        ~      simulated
        ~      The historical data input
        |

SumXY = INTEG(Xi*Yi/dt,0)
        ~      simulated*simulated
        ~      Sum of x*y
        |

Sum Yi = INTEG(Yi/dt,0)
        ~      simulated
        ~      Sum of y
        |

MX2=
         ZIDZ(SumX2,Count)
        ~      simulated*simulated
        ~      Mean of x^2
        |

MY2 = ZIDZ(SumY2,Count)
        ~      simulated*simulated
        ~      Mean of y^2
        |

SumX2 = INTEG(Xi*Xi/dt,0)
        ~      simulated*simulated
        ~      Sum of x^2
        |

SumY2 = INTEG(Yi*Yi/dt,0)
        ~      simulated*simulated
        ~      Sum of y^2
        |

:END OF MACRO:
```

Appendix 2: Vensim Code to Specify Error Variables

```
***************************************************
        .Stats
***************************************************~
                Summary Statistics
        |
Residuals addicted frac H users = SSTATS(Addicted frac of H users hist,Addicted frac of H users\
                :R2 Addicted frac H users,MAE Addicted frac H users,MAEM Addicted frac H users,MAPE Addicted frac H users\
                ,RMSE Addicted frac H users,MSE Addicted frac H users,Um Addicted frac H users,Us Addicted frac H users\
                ,Uc Addicted frac H users,Count Addicted frac H users)
        ~       dmnl
        ~               |
Residuals addicted frac PO abusers = SSTATS(Addicted frac of PO abusers hist,Addicted frac of PO abusers\
                :R2 Addicted frac PO abusers,MAE Addicted frac PO Abusers,MAEM Addicted frac PO Abusers\
                ,MAPE Addicted frac PO Abusers,RMSE Addicted frac PO Abusers,MSE Addicted frac PO Abusers\
                ,Um Addicted frac PO Abusers,Us Addicted frac PO Abusers,Uc Addicted frac PO Abusers\
                ,Count Addicted frac PO Abusers)
        ~       dmnl
        ~               |
Residuals addicted H users = SSTATS(Addicted H users hist,Addicted H users:R2 Addicted H users\
                ,MAE Addicted H users,MAEM Addicted H users,MAPE Addicted H users,RMSE Addicted H users\
                ,MSE Addicted H users,Um Addicted H users,Us Addicted H users,Uc Addicted H users,Count Addicted H users\
                )
        ~       popn
        ~               |
Residuals addicted PO abusers = SSTATS(Addicted PO abusers hist,Addicted PO abusers:R2 Addicted PO abusers\
                ,MAE Addicted PO Abusers,MAEM Addicted PO Abusers,MAPE Addicted PO Abusers,RMSE Addicted PO
Abusers\
                ,MSE Addicted PO Abusers,Um Addicted PO Abusers,Us Addicted PO Abusers,Uc Addicted PO Abusers\
                ,Count Addicted PO Abusers)
        ~       popn
        ~               |
Residuals addicts treated = SSTATS(Opioid addicts treated hist,Opioid addicts treated\
                :       R2 addicts treated,MAE addicts treated,MAEM addicts treated,MAPE addicts treated\
                ,RMSE addicts treated,MSE addicts treated       ,Um addicts treated,Us addicts treated\
                ,Uc addicts treated,Count addicts treated)
        ~       popn/Year
        ~               |
Residuals frac H initiates also PO = SSTATS(Frac of H initiates started with PO abuse hist\
                    ,Frac of H initiates started with PO abuse:R2 frac H initiates also PO,MAE frac H initiates also PO\
                ,MAEM frac H initiates also PO,MAPE frac H initiates also PO,RMSE frac H initiates also PO\
                    ,MSE frac H initiates also PO,Um frac H initiates also PO,Us frac H initiates also PO\
                ,Uc frac H initiates also PO,Count frac H initiates also PO)
        ~       dmnl
        ~               |
Residuals frac H users also PO = SSTATS(Frac of H users also PO abusers hist     \
                ,Frac of H users also PO abusers:R2 frac H users also PO,MAE frac H users also PO,MAEM frac H users also PO\
                ,MAPE frac H users also PO,RMSE frac H users also PO    ,MSE frac H users also PO\
                ,Um frac H users also PO,Us frac H users also PO,Uc frac H users also PO,Count frac H users also PO\
                )
        ~       dmnl
        ~               |
Residuals H initiates = SSTATS(H initiates hist,H initiates:R2 H initiates,MAE H initiates\
                ,MAEM H initiates,MAPE H initiates,RMSE H initiates,MSE H initiates,Um H initiates,\
                Us H initiates,Uc H initiates,Count H initiates)
        ~       popn/Year
        ~               |
Residuals H users = SSTATS(H users hist,H users:R2 H users,MAE H users,MAEM H users,MAPE H users\
                ,RMSE H users,MSE H users,Um H users,Us H users,Uc H users,Count H users)
        ~       popn
        ~               |
Residuals OD deaths from illicits = SSTATS(Illicit opioid OD deaths hist,Illicit opioid OD deaths\
                :       R2 OD deaths from illicits,MAE OD deaths from illicits,MAEM OD deaths from illicits\
                ,MAPE OD deaths from illicits       ,RMSE OD deaths from illicits,MSE OD deaths from illicits\
                ,Um OD deaths from illicits,Us OD deaths from illicits,Uc OD deaths from illicits,Count OD deaths from illicits\
                )
        ~       popn/Year
        ~               |
Residuals OD deaths from PO = SSTATS(OD deaths from authentic PO hist,OD deaths from authentic PO\
                :       R2 OD deaths from PO,MAE OD deaths from PO,MAEM OD deaths from PO,MAPE OD deaths from PO\
                    ,RMSE OD deaths from PO,MSE OD deaths from PO,Um OD deaths from PO,Us OD deaths from PO\
                ,Uc OD deaths from PO,Count OD deaths from PO)
        ~       popn/Year
        ~               |
Residuals OD deaths total = SSTATS(Total opioid OD deaths hist,Total opioid OD deaths\
                :       R2 OD deaths total,MAE OD deaths total,MAEM OD deaths total,MAPE OD deaths total\
                    ,RMSE OD deaths total,MSE OD deaths total,Um OD deaths total,Us OD deaths total\
                ,Uc OD deaths total,Count OD deaths total)
        ~       popn/Year
        ~               |
Residuals PO abuse initiates = SSTATS(PO abuse initiates hist,PO abuse initiates:R2 PO abuse initiates\
                ,MAE PO abuse initiates,MAEM PO abuse initiates,MAPE PO abuse initiates,RMSE PO abuse initiates\
                ,MSE PO abuse initiates,Um PO abuse initiates,Us PO abuse initiates,Uc PO abuse initiates\
                ,Count PO abuse initiates)
        ~       popn/Year
        ~               |
Residuals PO abusers = SSTATS(PO abusers hist,PO abusers:R2 PO abusers,MAE PO Abusers\
                ,MAEM PO Abusers,MAPE PO Abusers,RMSE PO Abusers,MSE PO Abusers,Um PO Abusers,Us PO Abusers\
                ,Uc PO Abusers,Count PO Abusers)
        ~       popn
        ~               |
Residuals street price PO = SSTATS(Avg street price PO hist       ,Avg street price PO\
                :R2 street price PO,MAE street price PO,MAEM street price PO,MAPE street price PO       \
                ,RMSE street price PO,MSE street price PO       ,Um street price PO,Us street price PO\
                ,Uc street price PO,Count street price PO)
        ~       dollars/mgME
        ~               |
Simple avg of all MAEM = (MAEM Addicted frac H users + MAEM Addicted frac PO Abusers \
                + MAEM Addicted H users  + MAEM Addicted PO Abusers  + MAEM frac H initiates also PO\
                 + MAEM frac H users also PO + MAEM H initiates + MAEM H users + MAEM OD deaths from illicits\
                 + MAEM OD deaths from PO  + MAEM OD deaths total + MAEM PO abuse initiates + MAEM PO Abusers\
                 + MAEM street price PO) / 14
        ~       dmnl
        ~               |
Total count across all data = Count Addicted frac H users + Count Addicted frac PO Abusers\
                 + Count Addicted H users + Count Addicted PO Abusers  + Count frac H initiates also PO\
                 + Count frac H users also PO + Count H initiates + Count H users + Count OD deaths from illicits\
                 + Count OD deaths from PO + Count OD deaths total + Count PO abuse initiates + Count PO Abusers\
                 + Count street price PO
        ~       dmnl
        ~               |
Weighted avg of all MAEM = ZIDZ( MAEM Addicted frac H users*Count Addicted frac H users\
                 + MAEM Addicted frac PO Abusers*Count Addicted frac PO Abusers + MAEM Addicted H users\
                *Count Addicted H users + MAEM Addicted PO Abusers*Count Addicted PO Abusers       \
                + MAEM frac H initiates also PO*Count frac H initiates also PO + MAEM frac H users also PO\
                *Count frac H users also PO + MAEM H initiates*Count H initiates + MAEM H users*Count H users\
                 + MAEM OD deaths from illicits*Count OD deaths from illicits+ MAEM OD deaths from PO\
                *Count OD deaths from PO + MAEM OD deaths total*Count OD deaths total + MAEM PO abuse initiates\
                *Count PO abuse initiates  + MAEM PO Abusers*Count PO Abusers+ MAEM street price PO\
                *Count street price PO, Total count across all data)
        ~       dmnl
        ~                 |
```

Appendix 3: Sample .voc and .vpd files for the “Estimate Uncertain Parameters” Step

Opioid2u.voc

```
:OPTIMIZER=Powell
:SENSITIVITY=Off
:MULTIPLE_START=Off
:RANDOM_NUMBER=Default
:OUTPUT_LEVEL=On
:TRACE=Off
:MAX_ITERATIONS=1000
:RESTART_MAX=0
:PASS_LIMIT=2
:FRACTIONAL_TOLERANCE=0.0003
:TOLERANCE_MULTIPLIER=21
:ABSOLUTE_TOLERANCE=1
:SCALE_ABSOLUTE=1
:VECTOR_POINTS=25
:MCINITMETHOD=0
:MCPAYOFFTYPE=0
:MCRECORD=0
:MCSCHEDULE=0
:MCLIMIT=0
:MCBURNIN=0
:MCNCHAINS=2
:MCOUTLIER=0.05
:MCGAMMA=1
:MCEPSILON=0.01
:MCDELTA=0.0001
:MCJUMP=0.05
:MCUPDATEPAIRS=2
:MCXOVER=0.2
:MCTEMP=1
:MCFTEMP=1
:MCCOOLING=1000
.45<=Addicted frac of H users initial=0.65<=0.65
0.09<=Addicted frac of PONHA initial=0.123<=0.16
0.01<=Addicted H user OD death rate initial=0.01<=0.02
0.05<=Addicted H user quit rate initial=0.138<=0.15
0.02<=Addicted PONHA move to heroin rate initial=0.021<=0.035
0.003<=Addicted PONHA OD death rate initial=0.0059<=0.007
0.08<=Addicted PONHA quit rate initial=0.149<=0.15
1.2<=Avg street price initial=1.6<=1.6
0.04<=Casual H user become addicted rate=0.096<=0.1
0.002<=Casual H user OD death rate initial=0.0081<=0.01
0.15<=Casual H user quit rate=.192<=0.3
0.01<=Casual PONHA become addicted rate initial=0.02<=0.03
0.002<=Casual PONHA move to heroin rate initial=0.0038<=0.01
0.0004<=Casual PONHA OD death rate initial=0.0004<=0.001
0.1<=Casual PONHA quit rate initial=0.209<=0.3
2000<=Consumption mgs MME per addicted PO abuser per month initial=3200<=3200
200<=Consumption mgs MME per casual PO abuser per month=200<=400
0.3<=Diverted frac of script gms MME initial=0.576<=0.7
0.5<=Exponent for med user addiction rate from script strength=.655<=1.5
1.5<=Fent boost of H user OD deaths 2017=3<=3
0.05<=Frac reduction H price per doubling of supply experience=0.45<=0.8
0.65<=Frequency of use multiplier for treated PO addicts=0.65<=0.8
0.03<=H users attract initiates rate=0.062<=0.1
0.0007<=HNPOU frac of adult popn initial=0.0009<=0.0009
0.0004<=HPPOU frac of adult popn initial=0.0005<=0.0005
0.05<=HPPOU drop PO use rate=0.091<=0.15
```

```
5<=Inflow of disguised Fent pill apparent mgs per adult 2016=26.1<=60
0.1<=PO abuse initiate rate initial=0.142<=0.25
2<=Quit multiplier for treated PO addicts=2<=3.5
0.1<=Reltv noninjecting nonoral frac for casual PONHA vs addicted=0.19<=0.5
2<=Reltv PONHA OD risk from disguised Fent pills=5.19<=20
0<=Slope for addicted H user quits from overdoses=0<=0.2
0.1<=Slope for addicted PONHA move to heroin from street price=.92<=2
0.05<=Slope for addicted PONHA quits from OD deaths=.2<=1
-1<=Slope for addicted PONHA quits from street supply=-0.28<=-0.05
-2<=Slope for avg street price from street supply=-0.61<=-0.2
0<=Slope for casual H user quits from overdoses=0<=0.2
0.1<=Slope for casual PONHA move to heroin from street price=.78<=1.5
0.1<=Slope for casual PONHA quits from OD deaths=0.28<=1
-1<=Slope for casual PONHA quits from street supply=-0.3<=-0.05
-0.7<=Slope for diversion of scripts from street supply=-0.38<=-0.1
-0.4<=Slope for H direct initiation from overdoses=-0.2<=0
0.1<=Slope for PONHA addiction rate from street supply=0.53<=1
-0.9<=Slope for PONHA initiation from OD deaths=-0.69<=-0.1
1.5<=Slope for PONHA initiation from street supply=2.43<=4
.08<=Street supply coverage adequate=.13<=.18
0.15<=Street supply coverage initial=0.15<=0.35
0.25<=Time to perceive OD deaths=.25<=2.25
```

Opioid2u.vpd

```
*C
PO abusers|PO abusers hist/2
*C
Addicted PO abusers|Addicted PO abusers hist/2
*C
H users|H users hist/10
*C
Addicted H users|Addicted H users hist/10
*C
PO abuse initiates|PO abuse initiates hist/10
*C
H initiates|H initiates hist/20
*C
Frac of H initiates started with PO abuse|Frac of H initiates started with PO abuse hist/10000000
*C
Avg street price PO|Avg street price PO hist/5000000
*C
Total opioid OD deaths|Total opioid OD deaths hist/1000
*C
OD deaths from authentic PO|OD deaths from authentic PO hist/1000
*C
Illicit opioid OD deaths|Illicit opioid OD deaths hist/1000
*C
Frac of H users also PO abusers|Frac of H users also PO abusers hist/20000000
```

Appendix 4: Demo .vsc and .lst files for the Sensitivity MC run steps

Opioid2U_MCrunTRI.vsc (for finding QPS candidates)

```
1000000,M,6592493,,0
Addicted frac of H users initial=RANDOM_TRIANGULAR(0.6,0.7,0.6,0.65,0.7)
Addicted frac of PONHA initial=RANDOM_TRIANGULAR(0.1,0.15,0.1,0.123,0.15)
Addicted H user OD death rate initial=RANDOM_TRIANGULAR(0.005,0.015,0.005,0.010,0.015)
Addicted H user quit rate initial=RANDOM_TRIANGULAR(0.07,0.21,0.07,0.138,0.21)
Addicted PONHA move to heroin rate initial=RANDOM_TRIANGULAR(0.01,0.03,0.01,0.21,0.03)
Addicted PONHA OD death rate initial=RANDOM_TRIANGULAR(0.004,0.007,0.004,0.0059,0.007)
Addicted PONHA quit rate initial=RANDOM_TRIANGULAR(0.08,0.22,0.08,0.149,0.22)
Avg street price initial=RANDOM_TRIANGULAR(1.4,1.8,1.4,1.6,1.8)
Casual H user become addicted rate=RANDOM_TRIANGULAR(0.05,0.15,0.05,0.096,0.15)
Casual H user OD death rate initial=RANDOM_TRIANGULAR(0.004,0.012,0.004,0.0081,0.012)
Casual H user quit rate initial=RANDOM_TRIANGULAR(0.13,0.25,0.13,0.192,0.25)
Casual PONHA become addicted rate initial=RANDOM_TRIANGULAR(0.01,0.03,0.01,0.02,0.03)
Casual PONHA move to heroin rate initial=RANDOM_TRIANGULAR(0.0025,0.005,0.0025,0.0038,0.005)
Casual PONHA OD death rate initial=RANDOM_TRIANGULAR(0.0002,0.0006,0.0002,0.0004,0.0006)
Casual PONHA quit rate initial=RANDOM_TRIANGULAR(0.1,0.3,0.1,0.209,0.3)
Consumption mgs ME per addicted PO abuser per month initial=RANDOM_TRIANGULAR(2000,5000,2000,3200,5000)
Consumption mgs ME per casual PO NMU per month=RANDOM_TRIANGULAR(100,300,100,200,300)
Diverted frac of script gms ME initial=RANDOM_TRIANGULAR(0.4,0.65,0.4,0.576,0.65)
Exponent for med user addiction rate from script strength=RANDOM_TRIANGULAR(0.3,0.9,0.3,0.655,0.9)
Fent boost of H user OD deaths 2017=RANDOM_TRIANGULAR(2,4,2,3.1,4)
Frequency of use multiplier for treated PO addicts=RANDOM_TRIANGULAR(0.55,0.85,0.55,0.65,0.85)
H users attract initiates rate initial=RANDOM_TRIANGULAR(0.04,0.1,0.04,0.073,0.1)
HNPOU frac of adult popn initial=RANDOM_TRIANGULAR(0.0007,0.0011,0.0007,0.0009,0.0011)
HPPOU drop PO use rate=RANDOM_TRIANGULAR(0.05,0.13,0.05,0.091,0.13)
HPPOU frac of adult popn initial=RANDOM_TRIANGULAR(0.0004,0.0006,0.0004,0.0005,0.0006)
Inflow of disguised Fent pill apparent mgs ME per adult 2016=RANDOM_TRIANGULAR(10,40,10,25,40)
PO abuse initiate rate initial=RANDOM_TRIANGULAR(0.05,0.25,0.05,0.142,0.25)
Quit multiplier for treated PO addicts=RANDOM_TRIANGULAR(1.5,3.5,1.5,2.0,3.5)
Reltv noninjecting nonoral frac for casual PONHA vs addicted=RANDOM_TRIANGULAR(0.1,0.3,0.1,0.19,0.3)
Reltv PONHA OD risk from disguised Fent pills=RANDOM_TRIANGULAR(3,7,3,5.19,7)
Slope for addicted H user quits from overdoses=RANDOM_TRIANGULAR(0,0.2,0,0,0.2)
Slope for addicted PONHA move to heroin from street price=RANDOM_TRIANGULAR(0.5,1.4,0.5,0.92,1.4)
Slope for addicted PONHA quits from overdoses=RANDOM_TRIANGULAR(0.1,0.4,0.1,0.25,0.4)
Slope for addicted PONHA quits from street supply=RANDOM_TRIANGULAR(-0.45,-0.15,-0.45,-0.28,-0.15)
Slope for avg street price from street supply=RANDOM_TRIANGULAR(-0.9,-0.3,-0.9,-0.61,-0.3)
Slope for casual H user quits from overdoses=RANDOM_TRIANGULAR(0,0.2,0,0,0.2)
Slope for casual PONHA move to heroin from street price=RANDOM_TRIANGULAR(0.4,1.2,0.4,0.78,1.2)
Slope for casual PONHA quits from overdoses=RANDOM_TRIANGULAR(0.1,0.6,0.1,0.35,0.6)
Slope for casual PONHA quits from street supply=RANDOM_TRIANGULAR(-0.45,-0.15,-0.45,-0.3,-0.15)
Slope for diversion of scripts from street supply=RANDOM_TRIANGULAR(-0.6,-0.2,-0.6,-0.38,-0.2)
Slope for H direct initiation from overdoses=RANDOM_TRIANGULAR(-0.4,0,-0.4,-0.2,0)
Slope for PONHA addiction rate from street supply=RANDOM_TRIANGULAR(0.25,0.8,0.25,0.53,0.8)
Slope for PONHA initiation from overdoses=RANDOM_TRIANGULAR(-1.2,-0.3,-1.2,-0.75,-0.3)
Slope for PONHA initiation from street supply=RANDOM_TRIANGULAR(1.2,3.6,1.2,2.43,3.6)
Street supply coverage adequate=RANDOM_TRIANGULAR(0.08,0.18,0.08,0.13,0.18)
Street supply coverage initial=RANDOM_TRIANGULAR(0.1,0.2,0.1,0.15,0.2)
Time to perceive overdoses=RANDOM_TRIANGULAR(0.125,1,0.125,0.25,1)
```

Opioid2UwithMax.lst (for the sensitivity runs to find the QPS)

```
Addicted frac of H users initial
Addicted frac of PONHA initial
Addicted H user OD death rate initial
Addicted H user quit rate initial
Addicted PONHA move to heroin rate initial
Addicted PONHA OD death rate initial
Addicted PONHA quit rate initial
Avg street price initial
Casual H user become addicted rate
Casual H user OD death rate initial
Casual H user quit rate initial
Casual PONHA become addicted rate initial
Casual PONHA move to heroin rate initial
Casual PONHA OD death rate initial
Casual PONHA quit rate initial
Consumption mgs ME per addicted PO abuser per month initial
Consumption mgs ME per casual PO NMU per month
Diverted frac of script gms ME initial
Exponent for med user addiction rate from script strength
Fent boost of H user OD deaths 2017
Frequency of use multiplier for treated PO addicts
H users attract initiates rate initial
HNPOU frac of adult popn initial
HPPOU drop PO use rate
HPPOU frac of adult popn initial
Inflow of disguised Fent pill apparent mgs ME per adult 2016
MAEM Addicted H users
MAEM Addicted PO Abusers
MAEM frac H initiates also PO
MAEM frac H users also PO
MAEM H initiates
MAEM H users
MAEM OD deaths from illicits
MAEM OD deaths from PO
MAEM OD deaths total
MAEM PO abuse initiates
MAEM PO Abusers
MAEM street price PO
maxofMAEMs
PO abuse initiate rate initial
Quit multiplier for treated PO addicts
Reltv noninjecting nonoral frac for casual PONHA vs addicted
Reltv PONHA OD risk from disguised Fent pills
Simple avg of all MAEM
Slope for addicted H user quits from overdoses
Slope for addicted PONHA move to heroin from street price
Slope for addicted PONHA quits from overdoses
Slope for addicted PONHA quits from street supply
Slope for avg street price from street supply
Slope for casual H user quits from overdoses
Slope for casual PONHA move to heroin from street price
Slope for casual PONHA quits from overdoses
Slope for casual PONHA quits from street supply
Slope for diversion of scripts from street supply
Slope for H direct initiation from overdoses
Slope for PONHA addiction rate from street supply
Slope for PONHA initiation from overdoses
Slope for PONHA initiation from street supply
Street supply coverage adequate
Street supply coverage initial
Time to perceive overdoses
Weighted avg of all MAEM
```

runbyfile.vsc (for running file-driven sens runs, where the .txt file is the QPS file)

```
200,F,1234,2uQPS.txt,0
```

Opioid2Uforscenarios.lst (for running the baseline QPS trajectories and policy comparisons)

```
Opioid overdoses seen in ED
Total opioid addicts
Total opioid OD deaths
```