SPEAKER 1: Welcome to this Vensim tutorial. As a group of students engaged in the system dynamics seminar at MIT Sloan, we will present how to estimate model parameters and the confidence interval in a dynamic model using Maximum Likelihood Estimation, MLE, Likelihood Ratio, LR method.

First, the basics of MLE are described, as well as the advantages and underlying assumptions. Next, the different methods available for finding confidence intervals of the estimated parameters are discussed. Then, a step-by-step guide to a parameter estimation using MLE-- an assessment of the uncertainty around parameter estimates using Univariant Likelihood Ratio Method in Vensim-- is provided.

This video is presented to you by William, George, Sergey, and Jim under the guidance of Professor Hazhir Rahmandad. The literature, Struben, Sterman, and Keith, highlight that estimating model parameters and the uncertainty of these parameters are central to good dynamic modeling practice. Models must be grounded in data if modelers are to provide reliable advice to policymakers.

Ideally, one should estimate model parameters using data that are independent of the model behavior. Often, however, such direct estimation from independent data is not possible. In practice, modelers must frequently estimate at least some model parameters using historical data itself, finding the set of parameters that minimize the difference between the historical and simulated time series.

Only then can estimations of model parameters and uncertainties of these estimations serve to test model hypotheses and quantify important uncertainties, which are crucial for decision-making based on modeling outcome. Therefore, when modeling a system in Vensim, if the purpose of this model involves numerical testing or projection, robust statistical parameter estimation is necessary.

Confidence intervals also serve as an important tool for decision-making based on modeling outcomes. Various approaches are available to estimate parameters in confidence intervals in dynamic model, including, for estimation, the generalized methods of moments in maximum likelihood, and for confidence intervals, likelihood-based methods and bootstrapping.

Maximum Likelihood Estimation is becoming increasingly important for nonlinear models when estimating nonlinear parameters that consist of non-normal, autocorrelated errors, and heteroscedasticity. It is simpler to understand the construct, yet at the same time, requires relatively little computational power.

MLE is best suitable for using historical data to generate parameter estimation and confidence intervals as long as errors of estimation are independent and identically distributed. When using MLE in complex systems where these assumptions are violated, and/or the analytical likelihood function might be difficult to find, one should use more advanced methods. This tutorial will not address these cases.

As the demonstration will show, the average laptop in use in 2014 is capable of running the analysis in a few minutes or less. This tutorial is based on the following four references-- "Bootstrapping for confidence interval estimation and hypothesis testing for parameters of system dynamics models" by Dogan, "A behavioral approach to feedback loop dominance analysis" by Ford, "Modeling Managerial Behavior," also known as "The Beer Game" by Sterman, and a soon-to-be published text by Struben, Sterman, and Keith, "Parameter and Confidence Interval Estimation in Dynamic Models-- Maximum Likelihood and Bootstrapping Methods."

More background and explanation on the theory of MLE can be found in these works. This tutorial will focus on the application of MLE in Vensim. The literature, Struben, Sterman, and Keith, stresses that modelers must not only estimate parameter values, but also the uncertainty in the estimates so they and others can determine how much confidence to place in the estimates and select appropriate ranges for sensitivity analysis in order to assess the robustness of the conclusions.

Estimating confidence intervals can be thought of as finding the shape of the inverted bowl, Figure 2. If for a given data set, the likelihood function for a set of parameters falls off very steeply for even small departures from the best estimate, then one can have confidence that the true parameters are close to the estimated value. As always, assuming the model is correctly specified, another maintained hypothesis are satisfied.

If the likelihood falls off only slowly, other values of the parameters are nearly as likely as the best estimates and one cannot have much confidence in the estimated values. MLE methods provides two major approaches to constructing confidence intervals or confidence regions. The first is the asymptotic method, AM, which assumes that the likelihood function can be approximated by a parabola around the estimated parameter. An assumption that is valid for a very large sample.

The second is the likelihood ratio, LR method. The LR is the ratio of the likelihood for a given set of parameter values to the likelihood for the MLE values. The LR method involves searching the actual likelihood surface to find values of the likelihood function that yield a particular value for the LR. That value is derived for the probability distribution of the LR and the confidence level desired, such as 95% chance that the true parameter value lies within the confidence interval.

This tutorial will use the Univariate Likelihood Ratio for determining the MLE competence interval in Vensim. The estimated parameter and competence interval mean that for a specific percentage of probability-- usually 95 or 99% that the real parameter falls within the confidence interval with the designated percent possibility. This is consistent with general applications of statistics and probability.

The LR our method of confidence interval estimation compared to the likelihood for the estimated parameter, theta hat, with that of an alternative set, theta star. The likelihood ratio is determined in equation one as R equals L theta hat divided by L theta star. Asymptotically, the likelihood ratio falls at chi square distribution, equation two.

This is valuable, because the univariate method requires no new optimizations once the MLE has been found. The critical parameter value for all parameters is then simply using equation three. A disadvantage of univariate confidence interval estimates, however, is that the parameter space is not fully explored. Hence, the effect of interaction between the parameters on LL is ignored,

The tutorial now switches to a real simulation using Vensim 6.1. It will show how the theory just described can be applied to estimating parameters of decision-making in the Beer Distribution Game.

SPEAKER 2: Hello. This tutorial is now going to explore analytics, by estimating the parameters for a well-analyzed model of decision-making used in the Beer Distribution Game. The model is described in the paper, "Modeling Managerial Behavior--Misperception of Feedback in a Dynamic Decision-Making Experiment" by Professor Sterman from 1989.

Participants in the beer game choose how much beer to order each period in a simulated supply chain. The challenge is to estimate the parameters of a proposed decision rule for participant orders. The screen shows the simple model of beer game decision-making. The ordering decision rule proposed by Professor Sterman is the following. The orders placed every week are given by the maximum of zero and the sum of expected customer orders-- the orders that participants expect to receive next period from their immediate customer and inventory discrepancy.

The inventory discrepancy is the difference between total desired inventory and some of actual on-hand inventory and supply line of on-order inventory. There are four parameters that are used in the model-- [INAUDIBLE], the weight on incoming orders in demand forecasting, S prime, the desired on-hand and on-order inventory, the fraction of the gap between desired and actual on-hand and on-order inventory ordered each week, and the fraction of the supply line the subject accounts for.

As modelers, we don't know what the parameters of the actual game are. But we have the data for actual orders placed by participants. And this data is read from the

Excel spreadsheet into the variable, actual orders. Let's start from some guesstimates and assign the following values. [INAUDIBLE], fraction of discrepancy between desired and actual inventory, and supply line fraction are all equal to 0.5. S prime, the total desired inventory, is 20 cases of beer.

With these parameters, the model runs and generates some alias of the order placed, which can be seem on this graph. Let's compare them against the actual orders observed in the beer game. We can see that although the trend is generally correct at the high level, the shape is totally different.

This necessitates the question, how can we effectively measure the fit of the data? The typical way is to calculate the sum of squared errors, which are differences between simulated and actual data points. Square, to make sure negative values don't reduce the total sum. The basic statistics of any variable can be found by using statistics to from either object output or bench tool.

For this tutorial, I have already defined it. And in this case, we can see [INAUDIBLE] of sum of squares of residuals is 1700, with a mean of about 35. This is pretty far from a good fit, and we can confirm it officially. Now let's see how to run the optimization to find the parameters that will bring the values of orders placed as close to the actual orders as possible.

The optimization control panel is invoked by using optimized tool on the toolbar. When you first open, you have to specify the file name here. Also, it is necessary to specify what type of optimization you're going to do. There are two types that can be used. They differ in the way they interpret and calculate the payoff value. A payoff is a single number that summarize the simulation. In the case of the simulation, it will be a measure of fit, which can be a sum of squared errors between the actual and the simulated values, or it can be a true electrical function.

If you are only interested in finding the best fit without worrying about confidence intervals, you can use the calibration mode. For calibration, it is not necessary to define the payoff functional mode. Instead, choose the model variable and the data variable with which the model variable will be compared.

In Vensim, then at each time step, the difference between the data and the model variable is multiplied by the weight specified. And this product is then squared. This number, which is always positive, is then subtracted from the payoff, so that the payoff is always negative. Maximizing the payoff means getting it to be as close to zero as possible. However, this is not a true log-likelihood function, so the results cannot be used to find the confidence intervals, which we need. Therefore, we're going to use the policy mode.

For policy mode, the payoff function must be specified explicitly. At each time step, the value of the variable or presenting a payoff function is multiplied by the weight. Then it is multiplied by time step and added to the payoff. The optimizer is designed to maximize the payoff. So variables, for which more is better, should be given positive weights, and those, for which less is better, should be given negative weights.

Here, [INAUDIBLE] are specified in the sum of squared errors. Residuals are calculated as the difference between orders placed and actual orders at each time step. This squared value is then accumulated in the stock sum of residual square. The tricky part is the variable, total sum of residual square. Please note that because Vensim, in the policy mode, multiplies the values of the payoff function by the time step, the payoff value is the weighted combination of the different payoff elements integrated over the simulation.

This is not what we're looking for. We are interested in the final value of a variable instead of the integrated value, because the final value is exactly the total sum of squared errors the way we defined it. Here's an example to illustrate the concept. In this line is a stock variable level. It can be seen that, if we just use the stock of sum of residual squared in the payoff function, we get the value, which is equal to the area under the curve, instead of the final value that we need.

[INAUDIBLE] look at only the final value. In a model, it is necessary to introduce new variable for the payoff function and use the following equation that makes its value to be zero at each time step, except for the final step. Add the current value of the residual squared to the stock level to account for the value of the current time step.

This effectually ignores all the intermediate values and looks only at the final value. So far, this is nothing different from what Vensim was doing in the calibration mode. We simply replicated what it would be doing automatically. In order to get the true likelihood function-- assuming normal distribution of errors-- we need to divide the sum of squared errors by two variances.

This is what you can see in the true low likelihood function variable. So far, in this demonstration, we don't know what the standard deviation of errors is going to be, because we haven't optimized anything yet, so we're going to leave this variable as 1. We are going to change it to the actual value of the standard deviation of errors later by the confidence interval.

Now let's change the run name and go back to Optimization Setup. And let's use our log-likelihood function as the payoff element. Because this is policy mode, we don't need to specify anything in Compare To field. For the weight, because this is policy mode, it is necessary to assign a negative value. This tells Vensim that we're looking to minimize the payoff function. If there are multiple payoff functions, the weight value would have been important.

For one function, it doesn't matter. But we don't want to change the values of the log-likelihood function to calculate true confidence internal values. Therefore, the log-likelihood function will not be scaled, and the weight is going to be minus 1. Next screen, the Optimization Control Panel, we have to specify the file name again. Then, make sure to use multiple starts. This means the analysis is less likely going to be stuck at a local optimum if the surface is not complete.

If multiple starts is random, starting points for any optimizations are picked randomly and uniformly over the range of each parameter. Next field, optimizer, tells Vensim to use probable method to find local minimum function at each start. Technically, you could ignore this parameter and just rely on random starts to find the values of a payoff function. However, using multiple starts together with optimization of each time step produces better and faster convergence.

The other values are left at the default levels. You can read more about these parameter in Vensim help, but generally, they control the optimization and are important for complicated models where simulation time is significant. Let's add now the parameters that Vensim will vary in order to minimize the payoff function.

Next, make sure Payoff Report is selected and hit Finish. For random or other random values of multiple start, the optimizer will never stop unless you click on the Stop button to interrupt it. If you don't choose multiple starts, the next few steps are not necessary as Vensim would be able to complete optimization and sensitivity analysis, which is performed in the final step of optimization. However, as I mentioned, the optimization can be stuck at a local optimum, thus producing some optimal results.

So it is recommended that you use multiple starts unless you know the shape of the payoff function and are sure that local optimum is equal to the global optimum. So having chosen random multiple starts, the question is, when to stop the optimization. This requires some experiments and depends on the shape of your payoff function.

In this case, the values of the payoff have been changing quite fast in the beginning and are now at the same level for quite some time. So it's a good time to attempt to interrupt the optimization and see if you like the results. As you can see, the shape of the orders placed is much closer to the actual order values.

The statistics shows a much better fit, with the total sum of squared error significantly lower at the level of just 279, with a mean of 5.813. The values of the rest of the optimized parameter can be looked up in the file with the same name as the run name and the extension out. We can open this file from Vensim using the Edit File command from File menu.

So far, we only know the optimal values, but not the confidence intervals. We need now to tell Vensim to use the optimal values found and calculate the confidence intervals around them. We are going to do it by using the output of the optimization and changing optimization parameters. To do this, first, we have to modify the optimization control parameters and turn of multiple starts, and optimize them, and save it as the optimization control type file, BOC.

Also, we need to change the value of standard deviation to the actual value to make sure we're looking at the true values of the log-likelihood function. This value can be found in the statistics [INAUDIBLE] again. We use it to change the variable in the model.

Open the Optimization Control Panel again, and leave the payoff function as it is. But on the Optimization Control Screen, open the file that was just created and check that multiple starts and optimizer are turned off. In addition, specify [INAUDIBLE] by choosing payoff value and entering the value by which Vensim will change the optimal likelihood function in order to find the constants controls.

Because for likelihood ratio method, the likelihood ratio is approximated by the chi-square distribution, it is necessary to find the value of chi-square distribution [INAUDIBLE] for 95th percentile for one degree of freedom, because we are doing the univariate analysis. A disadvantage of univariate confidence interval estimation is that the parameter space is not fully explored. Hence, the effect of interaction between parameters and log-likelihood function is ignored.

The value of chi-squared, in this case, is approximately 3.84146, which we're going to use in the sensitivity field. Now, the [INAUDIBLE] button. And as you see, this time, we don't have to wait as Vensim doesn't do any optimization. The results, allocated in the file that has the name of the run-- and the word sensitive with the extension tab.

Let's open this file and see our estimated parameters and the confidence intervals. Since we are using too much more likelihood function, the confidence bounds are not necessarily symmetric. This is a very simple tutorial. I hope you are now more familiar with [INAUDIBLE] for estimating parameters and confidence intervals for your models. Thank you.

SPEAKER 3: A summary of the seven key steps involved in determining the MLE in Vensim is contained on the slide. It can be printed and serve as a useful reference and check for those wishing to use the method.

MLE has its advantages and disadvantages in estimating parameters and their confidence intervals when performing dynamic modeling. In conclusion, when the condition of errors being independent and identically distributed are met, then MLE is a simple and straightforward method for estimating parameters in determining their statistical fit in system dynamics models developed in Vensim.