Animal Disease Spread Model (ADSM)
Text Support Document for Training

The slide-based training was designed to optimize visual interest. This format does not always create a slide bank that is printer friendly. In some sections, there are many images and little text. This text support document is intended to be a printer-friendly version of the slides that can be used as a reference. This document is not intended to take the place of the main training slides.

Training 8 Detailed Evaluation of Results

Slide	Image	Text
1	Laying Hens	Animal Disease Spread Model Detailed Evaluation of Results
2	ADSM Application Sample Scenario with Outputs	Table of Contents Results Evaluation Verification Validation Example What's Next?
3	No Image	Document Conventions The following conventions are used throughout the training modules: ***TRAINING MODULES*** other than the one you are current in will use all capital letters, bold face, italics and underline. *Rhetorical questions* and *extra notes* will be in orange italics. Conventions applying to the ADSM application are: Navigation tabs on right and Admin panels on left are designated with an underline. Examples are Project Panel or Population tab. Items with an action on click, such as [Apply] Button or [Save As] icon are enclosed in square brackets. *Parameter fields* (inputs) are in blue italics and *Variables* **(**outputs) are in green italics. Navigation Tabs > *Parameter field* indicates to go to the given navigation tab to find the given field. Hyperlinks appear in bright green type with underline http://navadmc.github.io/ADSM/
4	Gear Section Break	Results Evaluation
5	Nosy cow	Once there is a simple understanding of what the results look like, it is important to evaluate those results. The training Model Output goes through all the main outputs from ADSM. It is critical to understand how the parameter inputs created the outputs. This allows you to determine if those outputs are a valid representation of the disease systems you are attempting to simulate. This could be called a "Sniff Test".
6	llama	The outcome of an *ADSM* simulation (as with any computer simulation model) depends heavily on the quality of the scenario input

		parameters; the assumptions of the modeler who created the scenario; and the capabilities and limitations of the model framework itself. The utility of disease models like those created with *ADSM* critically depends on participation and interpretation of experts familiar with the behavior of disease within populations, and with the limitations, assumptions, and output of the model. **Without such participation, modeling results can be seriously misleading.** While *ADSM* is available as a service to animal health communities, the *ADSM* team does not necessarily endorse results obtained with the *ADSM* application or any conclusions drawn from such results.
7	goat	It is important that the model be both accurate and credible. Creating a meaningful results dataset requires both verification and validation. We will discuss each of these concepts.
8	Gear Section Break	Verification
9	sheep	Verification of a model is the process of confirming that the software programming was correctly implemented with respect to the conceptual model. It means the simulation application is performing the calculations in the manner that is expected. In other words, the model does what it was supposed to do.
10	Word cloud with ADSM team names	Verification has been the job of many people who have played a part in the ADSM and NAADSM Development Teams as the applications have been created. As such, this training will focus on Validation.
11	Gear Section Break	Validation
12	hen	Validation of a model confirms the accuracy of the model's representation of the real system you are attempting to simulate.
13	Known vs unknown graphic	The ability to completely and accurately represent a real system is very complex. Are the exact parameters known or are they unknown? Can the parameters reproduce the exact population including the specifics of the animal management practices and every possible contact? If these things were possible, a model would not be necessary. Concept: Tariq Halasa
14	sheep	How do you go about checking that a software application accurately simulates a real-world system? This is especially difficult when the input values that were put into the model parameters range from highly scientific to scientific guesses.
15	Sunlit field	There are extensive writings on methods of validating models. Since each user will be exercising this model in a different way on a different disease with different parameters, it will be necessary for users to apply some of these techniques to determine if the model credibly represents the system they are modeling.
16	Silhouette of small ruminant	This training will go through some tools to help you understand first what your model did, and if your model did what you asked it to do. You will then have to decide if it realistically represented the real system that you were expecting to simulate.
17	No image	Validation: Some Helpful References

		1. Reeves A, Salman MA, Hill AE. Approaches for evaluating veterinary epidemiological models: verification, validation and limitations. *Rev Sci Tech*. 2011;30(2):499-512. doi:10.20506/rst.30.2.2053 2. Kotiadis K, Robinson S. Conceptual modelling: Knowledge acquisition and model abstraction. 2008 Winter Simulation Conference, Miami, FL, USA, 2008, pp. 951-958, doi: 10.1109/WSC.2008.4736161. 3. Sargent RG. Verification and validation of simulation models. Proceedings of the 2003 Winter Simulation Conference, 2003. New Orleans, LA, USA, 2003, pp. 27-48 Vol.1, doi: 10.1109/WSC.2003.1261406. 4. Sargent RG. An introduction to verification and validation of simulation models. 2013 Winter Simulations Conference (WSC), Washington, DC, 2013, pp. 321-327, doi: 10.1109/WSC.2013.6721430. 5. Garner MG, Hamilton SA. Principles of epidemiological modelling. *Rev Sci Tech*. 2011;30(2):407-416. doi:10.20506/rst.30.2.2045 6. Sanson RL, Harvey N, Garner MG, et al. Foot and mouth disease model verification and 'relative validation' through a formal model comparison. *Rev Sci Tech*. 2011;30(2):527-540. doi:10.20506/rst.30.2.2051
18	Reverse image of a cow through a lens	Recall from the ADSM Overview that simulations produce a representation of a complex system. "All models are wrong, but some are useful" George E.P. Box
19	Various ADSM outputs	We will use outputs provided by ADSM to evaluate if a scenario provided expected output based on the input parameters.
20	Gear Section Break	Example
21	Small ruminant herd in background	Evaluating the Sample Scenario This example will use the Sample Scenario, run with all Supplemental Outputs turned on and a Summary generated. It will cover: High Level information Exposures, Adequate Exposures, and Infection Parameters driving spread of infection Controls – Detection and Destruction
22	Goat looking over gate	Please note that we will review only a small subset of the ADSM output to demonstrate the research methods. You can apply the methods used in this training to any variable that is created from ADSM.
23	ADSM Results Home Form with call outs	Review Results Home Results Home is the best place to start evaluating the scenario. It is important to know how to look at your results at both a high level and at a detailed level. *The Data Dictionary can provide field level definitions, use the ? Panel in the ADSM application to find the Data Dictionary*.

24	ADSM Results Home Population Heat Map	Sample Scenario Population Heat Map The Population Heat Map gives you a quick visual summary of the scenario outcome. Recall that the Population Heat Map is a combination of all the iterations that were run. While it is a high-level view, it helps to understand the broad scope of the outbreak. When using zones, the darker blue color indicates those areas that were involved in most or all iterations. As the color gets lighter, it means those areas were in involved in fewer iterations. Each unit will have a status graph to indicate the frequency of the unit outcomes. If Zones were not used, then no zone circles are drawn. Instead, each unit will have a status graph showing the frequency of unit outcomes. On a large population, the resolution may not allow you to scroll into the units to see the detail. On the first run, it is hard to tell if this is a reasonable outcome. As you gain more experience, you will become more aware of population heat map changes in response to changes in the parameter input.
25	ADSM Results Home and summary file example	In addition to the Population Heat Map, the selected output variable and the summary file allow quick glances at results values at a high level.
26	ADSM Results Home Population Heat Map	Have you used the Sample Scenario several times and noticed that it gives you similar results every time? *This is on purpose.* The Random Seed is a set value in the Sample Scenario. This causes the randomly varying parameters to draw the same values every time the model is run, resulting in the same results every time. When a seed value is specified, model results will only change when parameter inputs are changed, which can be useful for evaluation. *For the training example, it is important to have an example that can be explained consistently. Therefore, we are using the Sample Scenario.*
27	ADSM Results Home and summary file example	High Level Indicators The median outbreak duration and median numbers of infected units and animals can indicate unexpected results that require further exploration.
28	ADSM SQL Explorer Form	We can also query the raw data and learn more details about the results. You can access SQLite Explorer through the Admin Panel.
29	List of table names	These are the main tables that hold the results, so our queries will connect to these tables.
30	Example SQL query and data representation in the background	Additional Helpful Tables Databases store information in a way that is most efficient and without redundancy. Sometimes efficiency creates an output that is difficult to understand. For example, Production Types are stored as numeric identifiers on the Results tables. As a user, you would not know that ID even existed. By connecting the table with the Production Type names in a query, it is easier to understand the data results. The queries in the Example Database Queries show how to make this connection. The following tables are helpful when a Production Type name or a Zone name is needed.

		ScenarioCreator_ProductionType ScenarioCreator_Zone
31	ADSM example data	A Helpful Hint The production_type_id field for the first record is blank (or null) because that record shows values for all production types combined. The example queries take advantage of this by using a *Where Clause* to return only the combined record. WHERE 1=1 AND production_type_id is null *The Where 1=1 clause is a logical true. This makes it easy to add additional clauses without having to rewrite. Simply add another AND clause if needed.*
32	ADSM example data showing completed order	Another Helpful Hint Databases do not store data in the order that is logical to you. Instead they store it in the order that it was created. Use Order By iteration and day in your query to create a logical order. In this image, the actual order iterations completed was 2, 1, 3, 5 then 4. *Order by Iteration, day, last_day*
33	ADSM SQL Explorer Form with SQL query	Raw Data for Duration and Infected at First Detection We will start at a high level to look at these results. You can cut and paste this query into your SQL Explorer window if you would like hands-on experience. SELECT iteration, Day, Last_day, Diseaseduration, Outbreakduration, firstDetUInf, firstDetAInf FROM Results_DailyControls WHERE 1=1 AND last_day <> 0 Order by 1
34	ADSM SQL Explorer Form with SQL query results	Raw Data for Duration and Infected at First Detection Here are the results from the previous query.
35	ADSM SQL Explorer Form with SQL query results with call outs	Raw Data for Duration and Infected at First Detection What can be learned from this result set? Since this is the first look at the data, it is still early in the investigation. (Call outs) There were a range of outcomes. The fewer units infected at first detection (firstDetUInf) the shorter the outbreak seems to be… BUT, The count of animals (firstDetAInf) also matters as in the case of iteration 10. Iteration 10 had 4 units with many animals infected at first detection.

		The Summary file agrees with the raw data for minimum and maximum values.
36	No image	Duration and Infected at First Detection Many things could influence the duration, including both the spread of the disease and the control measures taken in response to the disease. While duration is a high-level indicator of what the model is doing, it may not be the best place to begin evaluating what is happening. The data also returned 2 duration variables, *Disease duration* and *Outbreak Duration*. The difference between disease duration (diseaseDuration) and outbreak duration (outbreakDuration) is this: • Disease duration is the number of days that any unit was in an infected state. • Outbreak duration is the number of days that any unit was in an infected state, plus any additional days needed to complete the control measures that were applied. Let's move on to look at more details in the results, starting with count of exposure, count of exposures that are adequate to cause disease, and count of infections that happen because of those exposures.
37	ADSM Results Home Exposures	Understanding Exposures - **10** iterations The visualization shows the summary of exposures throughout the outbreak. Exposures are not always adequate to cause infection. Even when the exposure is adequate, it doesn't cause disease if the recipient unit is not susceptible to disease due to immunity.
38	Calves	*What situations could make a unit not susceptible to disease when the exposure was adequate?* If the unit was previously exposed, and is now in an active disease state, exposure will not cause an infection. If the unit is in an immune state, due to either vaccine immunity or natural immunity, exposure will not cause an infection. If the unit is in a susceptible state, there is still a probability that the adequate exposure will not result in disease transmission. The Infection Probability parameter controls infection probability.
39	ADSM SQL Explorer Form with SQL query	Raw Data for Exposure, Adequate Exposure, and Infection Copy and paste this query into your SQL window if you want hands-on experience. Remember to use the Sample Scenario with Outputs, or any scenario that has been run. SELECT iteration, Day, Last_day, production_type_id, -- not useful, use case to get name CASE WHEN name IS NULL THEN "ALL" ELSE name END as productiontype, expcU, adqcU, infcU FROM Results_DailybyProductionType r LEFT JOIN -- needed since one side of join can be null ScenarioCreator_productiontype pt

		ON r.production_type_id = pt.id WHERE 1=1 AND production_type_id IS NULL -- only pulling back combined production type records AND iteration = 1 -- just look at one iteration to start ORDER BY 1, 2 -- don't assume order is correct
40	ADSM SQL Explorer Form with SQL query results	Raw Data for Exposure, Adequate Exposure, and Infection The query requested results only from Iteration 1, starting on day 1 and counting forward. On day 5, an exposure happens. The exposure is adequate, and it causes an infection. On day 6, another exposure happens; it is adequate and also causes an infection. The variables in this query are the cumulative variables; they are a sum of the total as the days progress. In the query window, it is possible to scroll down and view each day of the outbreak.
41	ADSM SQL Explorer Form with SQL query results	Raw Data for Exposure, Adequate Exposure, and Infection In the query window, it is possible to scroll down and see all 66 days that happened in iteration 1. By the end of iteration 1, there were 208 total exposures, 21 of those were adequate, and 15 of the exposures caused disease.
42	ADSM Supplemental output exposure file example	Other Ways to Look at Exposure The results set seems clear, but we can look at the exposures in other ways to understand more. Using the Supplemental Output File Daily Exposures gives more details. This is daily_exposures_1, which matches iteration 1. The reason code “Ini” on day 0 refers to the initial infection of the index herd, Unit 19, is that this was specified by the user. On day 5, Unit 19 had direct contact with Unit 1808, causing infection. On day 6, Unit 1808 had direct contact with Unit 1818, causing infection. The data will continue if exposures and infections happen in the simulation. This is the network of disease spread. *A clarification on the Daily Exposures file -* *Where “infection” is noted, the meaning is actually adequate exposure.*
43	ADSM Supplemental output exposure file example	Learning More from Daily_Exposures Since we have a nice view of this data, there are a few more things to point out. Day 9 has many exposures and no infections. Why not? Perhaps the exposure was not adequate. Also, Unit 1808 and Unit 1818 are already infected, so those units won’t get infected again. The exposure count on Day 9 doesn’t match the query (shown on page 44). Why are there more exposures in the query? The Supplemental Output File is not going to show Airborne Spread unless it is adequate to cause disease. Airborne Spread creates a massive number of exposures and it would make huge output files. Instead, the next step will be looking at spread by contact method and that will show the details.

		Another hint from this file is that zone names do not appear until Day 11. That is a clue that detection didn't happen until Day 10 to trigger zone formation. There are ways you can double-check detection in other variables.
44	ADSM SQL Explorer Form with SQL query	Details for Routes of Exposure Copy and paste this query into your SQL window if you want hands-on experience. Remember to use the Sample Scenario with Outputs, or any scenario that has been run. *Note that Production_Type_id was dropped out. Having a field with no value doesn't tell us much once we understand why it is blank.* SELECT iteration, Day, Last_day, production_type_id, -- not useful, use case to get name CASE WHEN name IS NULL THEN "ALL" ELSE name END as productiontype, expcU, expcUDir, expcUInd, expcUAir, adqcU, infcU FROM Results_DailybyProductionType r LEFT JOIN -- needed since one side of join can be null ScenarioCreator_productiontype pt ON r.production_type_id = pt.id WHERE 1=1 AND production_type_id IS NULL -- only pulling back combined production type records AND iteration = 1 -- just look at one iteration to start ORDER BY 1, 2 -- don't assume order is correct
45	ADSM SQL Explorer Form with SQL query	Raw Data for Exposure with Cause, Adequate and Infection Methods of Spread The results from the previous page query look like this (image). You can determine which of your contact methods are causing the most spread. Airborne is now included, and the total count is more understandable. Between days 8 and 9, 6 exposures happened, but only one of those exposure was adequate. Also, no more infections happen, so the exposure must have been to the Unit that was already infected.
46	Cow in sunset scene	Note About Infection Understanding how infection is counted in the raw data is complicated. Since infection happens on one day, and the disease state transition occurs on the next day, there are opportunities for several situations that can add complexity. **Most of the cases are added in a straightforward fashion:** **day n:** one or more adequate exposures happen **day n+1:** unit changes to infected state

		This situation is clear: if there is one susceptible unit that became infected on day n, we add 1 to infcU. **However, there are some cases where an infection on day n does not lead to a state change on day n+1.** Specifically, the count varies when a unit is both infected and vaccinated on day n (with the days to immunity parameter set to zero-day delay) or both infected and destroyed on day n. In those cases, the change of state would never show up in the daily_states output on day n+1. In these situations, the simulation engine takes an action that is not visible. It "flips a coin" and may or may not add 1 to infcU.
47	Herd and sunset	Wait a minute! Something is missing When you created parameters, you decided: • the production types that can be contacted by other production types • How often the production types contact each other • The methods by which the production types come into contact There must be more details, right? The first three queries were designed to be preliminary steps to review the data, by collapsing the records so that only the combined production type record is showing. The next steps break down the results and show more details about production types.
48	ADSM SQL Explorer Form with SQL query	Query for Exposure, Adequate Exposure, and Infection Methods of Spread by Production Type Copy and paste this query into your SQL window if you want hands-on experience. Remember to use Sample Scenario with Outputs, or any scenario that has been run. SELECT iteration, Day, Last_day, production_type_id, -- not useful, use case to get name CASE WHEN name IS NULL THEN "ALL" ELSE name END as productiontype, expcU, -- adqcU, -- leaving out adq, because model doesn't return this detail infcU , infcUDir, infcUInd, infcUAir FROM Results_DailybyProductionType r LEFT JOIN -- needed since one side of join can be null ScenarioCreator_productiontype pt ON r.production_type_id = pt.id WHERE 1=1 AND production_type_id IS NOT NULL -- only pulling back specific production type records AND iteration = 1 -- just look at one iteration to start ORDER BY 1, 2 -- don't assume order is correct
49	ADSM SQL Explorer Form with SQL query results	Raw Data for Exposure, Adequate Exposure, and Infection Methods of Spread by Production Type

		This is the dataset from the previous page query. You can determine which of your contact methods are causing the most spread, and in which production types that spread is occurring. Notice the row count doubled, because there are two production types. Disease spread in this iteration occurs mainly in cattle, until Day 16 when it spreads to swine by airborne exposure.
50	ADSM disease spread Form	The Parameters Explain the Story In ADSM, all the parameters are in the individual tabs associated with each type of spread. It is possible to open each one of these and research every parameter block individually. There is no reason to open every one of these blocks when we have access to the data behind the application. The correct query will get us an answer with less hassle. The query is a little more complicated. It stacks results from direct spread and indirect spread together. *You've got this. You are a query professional at this point!*
51	ADSM SQL Explorer Form with SQL query results	Evaluation of Spread These tools provide a way to look at how spread is occurring, and which production types are being affected. Do these results make sense based on the Sample Scenario parameter inputs? We will check parameters next. (call outs) The last day of the outbreak is a good place to evaluate this question Most of the infection was caused by direct contact from Cattle to Cattle A small amount was cause by the other methods of contact
52	ADSM SQL Explorer Form with SQL query	Query for Direct and Indirect Disease Spread Parameters Copy and paste this query into your SQL window if you want hands-on experience. You can use this query on any database, as it is not looking at results. SELECT 'direct spread' as Spreadmethod, Name, CASE WHEN use_fixed_contact_rate = 0 THEN 'No' ELSE 'Yes' END use_fixed_contact_rate, Contact_rate, infection_probability , CASE WHEN latent_units_can_infect_others = 0 THEN 'No' ELSE 'Yes' END as latent_units_can_infect_others, CASE WHEN subclinical_units_can_infect_others = 0 THEN 'No' ELSE 'Yes' END as subclinical_units_can_infect_others FROM ScenarioCreator_directspread ds LEFT JOIN (SELECT id, name as distance_pdf FROM ScenarioCreator_probabilitydensityfunction) dd ON ds.distance_distribution_id = dd.id LEFT JOIN (SELECT id, name as movement_control_pdf FROM ScenarioCreator_probabilitydensityfunction) mc ON ds.movement_control_id = mc.id UNION SELECT 'indirect spread' Spreadmethod, Name, CASE WHEN use_fixed_contact_rate = 0 THEN 'No' ELSE 'Yes' END use_fixed_contact_rate,

		Contact_rate, infection_probability, 'Not possible', -- latent_units_can_infect_others, CASE WHEN subclinical_units_can_infect_others = 0 THEN 'No' ELSE 'Yes' END subclinical_units_can_infect_others FROM ScenarioCreator_indirectspread ids LEFT JOIN (SELECT id, name as distance_pdf FROM ScenarioCreator_probabilitydensityfunction) dd ON ids.distance_distribution_id = dd.id LEFT JOIN (SELECT id, name as movement_control_pdf FROM ScenarioCreator_probabilitydensityfunction) mc ON ids.movement_control_id = mc.id
53	ADSM SQL Explorer Form with SQL query results	Raw Data for Parameters for Direct and Indirect Spread This is the dataset from the previous page query. You can determine which of your contact methods were parameterized to cause the most spread, and in which production types. Now that we can see the parameters, it does make sense that Cattle > Cattle Direct Spread caused the most infections; the contact rate multiplied by the infection probability is highest for that route of spread. *Using a meaningful naming convention on the spread methods helps make this example clear. Name is user-defined.*
54	ADSM SQL Explorer Form with SQL query	Query for Airborne Disease Spread Parameters Copy and Paste this query into your SQL window if you want hands on experience. Remember to use Sample Scenario with Outputs, or any scenario that has been run. Here is where you can note the effect of turning on the airborne exponential decay parameter. SELECT 'airborne spread' as SpreadMethod, asp.name, Spread_1km_probability, max_distance as max_distance_km, 'and is', CASE WHEN Use_airborne_exponential_decay = 0 THEN 'in effect due to linear airborne decay' ELSE 'not in effect due to linear airborne decay' END as max FROM ScenarioCreator_airbornespread asp JOIN ScenarioCreator_disease d ON d.id = asp._disease_id
55	ADSM SQL Explorer Form with SQL query results	Raw Data for Parameters for Airborne Spread This is the dataset from the previous page query. You can determine how airborne spread was parameterized. From these parameters we would expect that cattle, as compared to swine, are more likely to contribute to airborne spread of disease to any susceptible production type and that the maximum distance that airborne spread can occur between an infectious and susceptible premises is 6 km.
56	ADSM SQL Explorer Form	Query for Exposure, Adequate Exposure and Infection Methods of Spread by Production Type for Last Day all iterations

	with SQL query	After walking through the steps for looking at one iteration, let's expand and look at the last day only but look across all 10 iterations Copy and Paste this query into your SQL window if you want hands on experience. Remember to use Sample Scenario with Outputs, or any scenario that has been run. SELECT iteration, Day, Last_day, CASE WHEN name IS NULL THEN "ALL" ELSE name END as productiontype, expcU,expcUDir, expcUInd, expcUAir, adqcU, infcU, infcUDir, infcUInd, infcUAir FROM Results_DailybyProductionType r LEFT JOIN -- needed since one side of join can be null ScenarioCreator_productiontype pt ON r.production_type_id = pt.id WHERE 1=1 AND last_day = 1 -- Last day is true AND production_type_id IS NOT NULL -- pulling back specific production type records -- AND iteration = 1 -- all iterations – turns off this clause ORDER BY 1, 2 -- don't assume order is correct
57	ADSM SQL Explorer Form with SQL query results	Raw Data for **10** iterations This is the dataset from the previous page query. Note that last_day now = True. There are a range of outcomes, as expected with the stochastic nature of the simulation. Does the evaluation hold true when looking at more iterations? Referring back to the Sample Scenario parameterization, Direct Spread (from Cattle > Cattle) caused most of the infection. In airborne spread, cattle do spread more disease than swine, as expected.
58	Cattle on a hillside	Validation Check in We have looked at exposure, adequate exposure and infection in several ways. We have also checked the parameters. So far, my simulation is providing the results I would expect from the parameters that I put in. In the next step, the Supplemental Output files will provide additional information.
59	File structure and example of Daily_States	Supplemental Output Files – **Daily States** Since we have been looking at the routes of infection, let's look at the Supplemental Output File with the daily disease state. In this case, we will look at states_1.csv to stay with the iteration 1 example.
60	Example of Daily_States	Supplemental Output Files – **Daily States** Unit 19 is the index herd. This is a good opportunity for a verification step. This view allows verification of the steps in the disease progression. The first thing I want to know is the production types of my units.

		Quick Hint – The production type information is on the Population tab, but instead just open Daily_events_1.csv file, because most of these units trigger events almost immediately. Image from Daily_Events_1 and all units are cattle.
61	Example of Daily_States and pdf	Supplemental Output Files – **Daily States** states_1 file probability density function Unit 19 is L (Latent) 8 days. On the 9th day it changes to B (subclinical). Unit 1808 is L (Latent) 1 day. On the 2nd day it becomes B (subclinical). Unit 1818 is L (Latent) 4 days. On the 5th day it becomes B (subclinical). Unit 1830 is L (Latent) 3 days. On the 4th day it becomes B (subclinical). Unit 458 is L (Latent) 3 days. On the 4th day it becomes B (subclinical). The probability density function assigned to the latent stage in cattle is named Latent period – cattle and is Triangular, 0, 3, 9. The values for the latent period days in cattle units (8, 1, 4, 3, 3) fall within the expected range of the probability density function (0 – 9 days) with most of the time lasting 3 days. This is a small example of making sure the model is doing what we expect.
62	ADSM Control Protocol, Detection showing pdf	Controls Now that we have a better understanding of how disease is spreading, let's look at how the control measures are behaving. Just a reminder: If destruction is checked in main Control Protocol, then destruction will happen for detected units. The additional settings in destruction put in additional units, either because of a trace or because of pre-emptive destruction in a ring. Note that Control Protocols are assigned to one or more production types.
63	SQL query code and Daily_events example	Assessing Detection There are several ways we can explore detection. At a high level, using the Results_DailyControls table it is possible to simply determine with a y/n flag the day detection occurred with the field detOccurred. At a daily level, using the Results_DailybyProductionType table there are multiple fields reporting on detection. At the herd and daily level, using the Supplemental Output File Daily_events, you can see a detailed list of detection events. SELECT iteration, Day, DetOccurred FROM Results_DailyControls WHERE 1=1 AND last_day = 1 ORDER BY 1, 2
64	ADSM SQL Explorer Form with SQL query	Query for Detection Copy and Paste this query into your SQL window if you want hands on experience. Remember to use Sample Scenario with Outputs, or any scenario that has been run. There's another thing we did in the SQL code. Using the keyword *as,* the field named Iteration was renamed to IT. This is called an a*lias.* You can alias field names and table names. We automatically did it

		on table names to reduce the amount of code needed in the ON statement. SELECT iteration as IT, Day, Last_day, CASE WHEN name IS NULL THEN "ALL" ELSE name END as productiontype, infcU, -- infection by Unit detcU, -- all detection by unit detcUClin, -- detection by clinical exam (default method of detection) detcUTest -- detection by laboratory testing (option method of detection) -- First Detection firstDetection, firstDetectionClin, firstDetectionTest FROM Results_DailybyProductionType r LEFT JOIN -- needed since one side of join can be null ScenarioCreator_productiontype pt ON r.production_type_id = pt.id WHERE 1=1 AND production_type_id IS NULL -- pulling back combined production type records AND iteration = 1 -- one iteration ORDER BY 1, 2 -- don't assume order is correct
65	ADSM SQL Explorer Form with SQL query results	Detection Raw Data Look at what happens in the raw data as the outbreak proceeds. (call outs) Day 5 Infection starts to spread Day 10 Detection happens Day 10 First Detection is stamped onto the record *Note some of the fieldnames were shortened to fit everything into one view*
66	ADSM SQL Explorer Form with SQL query results	Detection Raw Data By the last day, the raw data looks like this. Something seems wrong with this. *How are there more detections than infections?* After initial detection anywhere in the population, contact tracing may occur. Traced units may be examined for clinical signs and/or tested. Just as in real life, both of those processes could identify infection in the same unit. When this occurs, the model records both events as detections. This makes it appear that detections were over-counted. *FirstDetection* field is still showing the day of first detection. *Note some of the fieldnames were shortened to fit everything into one view*
68	ADSM SQL Explorer Form with SQL query	Query for Detection on Last Day *Is infection always detected?* Looking at 10 iterations provides a variety of results to see the stochastic nature of the model. In iteration 1, all infections appeared to be detected, but if we look at other iterations there are different outcomes. In this query, results are limited to the last day.

		Copy and Paste this query into your SQL window if you want hands on experience. Remember to use Sample Scenario with Outputs, or any scenario that has been run. SELECT iteration, Day, Last_day, CASE WHEN name IS NULL THEN "ALL" ELSE name END as productiontype, infcU, -- infection cumulative by Unit detcU -- detection cumulative by unit FROM Results_DailybyProductionType r LEFT JOIN -- needed since one side of join can be null ScenarioCreator_productiontype pt ON r.production_type_id = pt.id WHERE 1=1 AND production_type_id IS NULL -- pulling back combined production type records AND last_day = 1 ORDER BY 1, 2 -- don't assume order is correct
69	ADSM SQL Explorer Form with SQL query results	Detection Raw Data Last Day There were several iterations that had fewer detections than infections. *Why did iteration 9 have 1 detection when there were 0 infections?* The index unit was detected. *What happens to those units that are not detected?* The Supplemental Output file states_2 will show the state. Iteration 2 is an example. In states_2.csv on Day 64, Unit 1845 changes to N (Natural Immune) as it is never detected.
70	ADSM SQL Explorer Form with SQL query	Query for Destruction as a Control Measure Destruction is another common control measure used in animal disease outbreaks. An evaluation of depopulation's effectiveness may also reveal something about the scenario. Copy and paste this query into your SQL window if you want hands-on experience. Remember to use Sample Scenario with Outputs, or any scenario that has been run. SELECT iteration, Day, Last_day, CASE WHEN name IS NULL THEN "ALL" ELSE name END as productiontype, infcU, -- infection cumulative by Unit detcU, -- detection cumulative by unit FirstDestruction, descU -- destruction cumulative by Unit FROM Results_DailybyProductionType r LEFT JOIN -- needed since one side of join can be null ScenarioCreator_productiontype pt ON r.production_type_id = pt.id WHERE 1=1 AND production_type_id IS NULL -- pulling back combined production type records AND iteration = 1 ORDER BY 1, 2 -- don't assume order is correct

71	ADSM SQL Explorer Form with SQL query results And Daily_events file	Raw Data for Destruction as a Control Measure For Iteration 1 First detection happened on Day 10. On Day 16, destruction starts. Recall that detection must happen before the model knows to destroy the unit. Once a detection has occurred, there are 3 main options: 1) Destroy the detected unit 2) Destroy a trace in or out 3) Make a pre-emptive destruction ring The Supplemental Output file named Daily_events_1 shows exactly who was destroyed.
72	ADSM Destruction Global Form	Destruction Delay Verification This is another opportunity to verify that the parameters are guiding the model's action. Recall detection didn't happen until Day 10. On Day 16, destruction starts. Recall that detection must happen before the model knows to destroy the unit. The parameter **Destruction Program Delay** is set to 5 days. Therefore, a Day 10 detection with a Day 16 destruction makes sense in iteration 1.
73	Cow during oral exam	Summary of Evaluation Steps 1. At the beginning, we looked at duration and number of animals on infected premises at first detection 2. Then we ventured into Exposures - Exposure, Adequate Exposure, and Infection - Exposure, Adequate Exposure, and Infection by spread method - Exposure, Adequate, and Infection by production type 1. Spread parameters 2. Daily States 3. Detection 4. Destruction Depending on the specifics of your scenario there may be other variables, like those related to vaccination, that you should explore.
74	Gear Section Break	What's Next?
75	Flock of Sheep	**Join the flock!** **Learn more about ADSM or try an example** **ADSM is currently available** at https://github.com/NAVADMC/ADSM/releases/latest Try the sample scenario https://github.com/NAVADMC/ADSM/wiki/A-Quick-Start-Guide:-Running-the-sample-scenario Read the wiki pages link https://github.com/NAVADMC/ADSM/wiki
76	Goat on with green foliage	What's Next? Addition training material is posted at http://navadmc.github.io/ADSM/ Training includes: Overview Populations and Production Types Getting Started Disease Parameters Control Parameters

		Output Settings and Run Results Detailed Evaluation of Results - Verification and Validation Vaccination Strategy Administration
77	Cows grazing with blue sky and green grass	The outcome of an ADSM simulation (as with any computer simulation model) depends heavily on the quality of the scenario input parameters; the assumptions of the modeler who created the scenario; and the capabilities and limitations of the model framework itself. The utility of disease models like those created with ADSM critically depends on input and interpretation of experts familiar with the behavior of disease within populations, and with the limitations, assumptions, and output of the model. While ADSM is available as a service to animal health communities, the ADSM team does not necessarily endorse results obtained with the ADSM application or any conclusions drawn from such results. Note that the parameters provided in the Sample Scenario are simple examples to clarify concepts in the application. These parameters do not represent any real population or disease event.
78	Cattle image	This work was funded in whole through Cooperative Agreement AP18VSCEAH00C005 by the Animal and Plant Health Inspection Service, an agency of the United States Department of Agriculture. University of Tennessee Animal Science logo Photo credits Canva.com Pinecroft Farms, Woodstock CT, Mariah Chapman Dr. Melissa Ackerman Halasa, T Ali Abo Kareem Photography British Museum Mariposa Ranch Watusi Dr. Renee Dewell
	Metadata	Last Update: 1/2/2024 By: Schoenbaum Approved: Freifeld