



**SCIENTIFIC TEST & ANALYSIS TECHNIQUES
CENTER OF EXCELLENCE**

Model Validation Level (MVL) R Tool User Guide

June 2026

Nicholas Jones, Ctr
Kyle Provost, Ctr
Corinne Stafford, Ctr
Chloe Young, Ctr



To develop and apply independent, tailored Scientific Test & Analysis Techniques solutions to Test and Evaluation that deliver insight to inform better decisions.

About this Publication:

This work was conducted by the Scientific Test & Analysis Techniques Center of Excellence under contract FA8075-18-D-0002, Task FA8075-21-F-0074.

For more information:

Visit, www.AFIT.edu/STAT

Email, AFIT.ENS.STATCOE@us.af.mil

Call, 937-255-3636 x4736

Technical Reviewers:

Steve Oimoen

Wayne Adams

Copyright Notice: No Rights Reserved

Scientific Test & Analysis Techniques Center of Excellence

2950 Hobson Way

Wright-Patterson Air Force Base, Ohio

The views expressed are those of the author(s) and do not necessarily reflect the official policy or position of the Department of the Air Force, the Department of Defense, or the U.S. government.

Version: 4, FY26

Table of Contents

1. Introduction.....	1
1.1. <i>Purpose</i>	<i>1</i>
1.2. <i>Background.....</i>	<i>1</i>
2. Package Dependencies	1
3. Scope Input File	2
3.1. <i>Scope Input File Contents.....</i>	<i>3</i>
3.2. <i>Scope Input File Formatting and Notation</i>	<i>3</i>
3.3. <i>Using Multiple Scope Input Files.....</i>	<i>4</i>
4. Model and Referent Data Files.....	4
4.1. <i>Model Data Input File.....</i>	<i>4</i>
4.2. <i>Referent Data Input File.....</i>	<i>6</i>
4.3. <i>Formatting of Response Data</i>	<i>6</i>
4.4. <i>Interpolator Data Input File.....</i>	<i>15</i>
5. Data Matching and Validation Point Selection.....	16
6. Running the MVL Tool.....	17
6.1. <i>Shiny Application</i>	<i>17</i>
6.2. <i>MVL Function.....</i>	<i>18</i>
6.3. <i>MVL_a Function</i>	<i>19</i>
7. MVL Output	21
7.1. <i>MVL Output Tables.....</i>	<i>21</i>
7.2. <i>MVL_a Output Tables.....</i>	<i>23</i>
7.3. <i>Additional Outputs.....</i>	<i>23</i>
7.4. <i>Vignette</i>	<i>24</i>
7.5. <i>Output Interpretation & Troubleshooting</i>	<i>27</i>
8. Contact Information.....	28
References	29
Appendix A: Key Definitions	30
Appendix B: Catapult Model Example.....	32

1. Introduction

1.1. Purpose

This document is a User Guide for the Scientific Test and Analysis Techniques Center of Excellence (STAT COE) Model Validation Level (MVL) tool, implemented in R. This User Guide will instruct users on how to format the necessary input files and interpret results output by the tool, with examples provided.

1.2. Background

An MVL is an objective, automatable metric scored from 0-9 that quantifies how much trust can be placed in the results of a model to represent the real world. MVLs aim to provide utility both for decision makers to quickly evaluate model risk and for model developers to identify areas to improve model trust. Additionally, the MVL framework supports digital engineering via automation to continuously perform validation and by quantifying trust in preexisting models for new use cases.

The STAT COE developed MVLs to provide an objective, automatable metric that quantifies the degree of trust that can be placed in the results of a model to represent the real world. The MVL framework defines validation in terms of three key pillars which must be carefully considered when determining the degree to which a model can be considered valid: fidelity, referent authority, and scope. Comprehensive documentation and guidance for MVLs and all underlying methods, as well as practical considerations for incorporating MVLs into validation plans, can be found in the companion paper, Model Validation Levels: Methods and Implementation (Stafford et al., 2024). This document provides instructions for how to automatically compute the MVL using the R-based MVL package, which implements the methods outlined in the companion paper. Common terms are defined in Appendix A.

2. Package Dependencies

The MVL tool has the following R package dependencies:

- dplyr v 1.1.4 or later
- purr v 1.0.2 or later
- magrittr v 2.0.3 or later
- tibble v 3.2.1 or later
- tidyselect, v 1.2.1
- readxl v1.4.3
- geometry v 0.4.7 or later
- LHS v 1.2.0 or later
- RANN v 2.6.1 or later
- knitr v 1.45 or later
- shiny
- shinyWidgets
- shinythemes
- markdown
- DT
- openxlsx

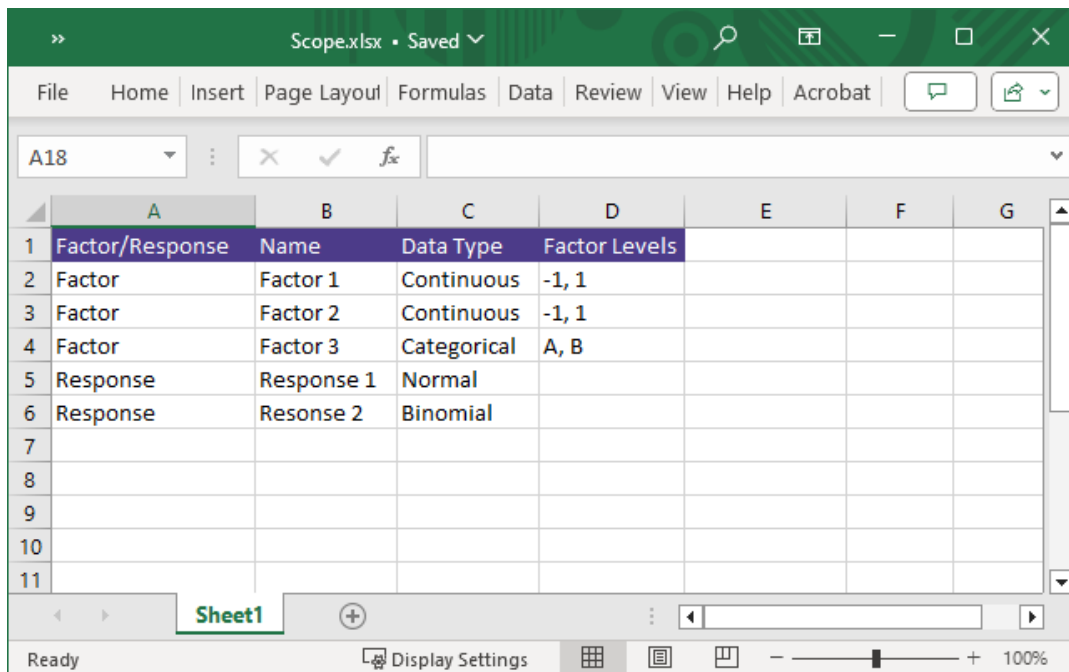
These packages can be installed from the CRAN library by running the following lines in the R

console.

```
install.packages("dplyr")
install.packages("purrr")
install.packages("magrittr")
install.packages("tibble")
install.packages("tidyselect")
install.packages("readxl")
install.packages("geometry")
install.packages("LHS")
install.packages("RANN")
install.packages("knitr")
install.packages("shiny")
install.packages("shinyWidgets")
install.packages("shinythemes")
install.packages("markdown")
install.packages("DT")
install.packages("openxlsx")
```

3. Scope Input File

The scope over which the model will be validated is read into the MVL tool from a scope input file. The MVL tool is designed to read scope data from an Excel (.xlsx) file, formatted as shown in Figure 1. Scope input files may have any valid file name. An Excel scope input file template is available within the MVL Shiny application (section 6.1).



	A	B	C	D	E	F	G
1	Factor/Response	Name	Data Type	Factor Levels			
2	Factor	Factor 1	Continuous	-1, 1			
3	Factor	Factor 2	Continuous	-1, 1			
4	Factor	Factor 3	Categorical	A, B			
5	Response	Response 1	Normal				
6	Response	Response 2	Binomial				
7							
8							
9							
10							
11							

Figure 1
Example Scope Data Input File

3.1. Scope Input File Contents

The scope input file contains the fields shown in Table 1.

Table 1
Scope Input File Field Descriptions

Scope Input File Field:	Factor/Response	Name	Data Type	Factor Levels
Data Type:	String Literal	String	String Literal	String
Description:	Denotes whether the supplied field is a factor (input) or response (output) of the system	Name of the factor or response. Must exactly match the name(s) of related fields in Referent and Model data files (see Section 4)	The data type of the factor or expected distribution of the response data (see Section 4.3 for response type descriptions)	“Continuous” Data Type: list of high and low values bounding the factor range of interest. “Categorical” Data Type: explicit, unordered list of all allowed values as strings.
Allowed Values for Responses:	“Response”	Any valid variable name in R, must be unique	“Continuous”, “Normal”, “Exponential”, “Binomial”, “Poisson”	No value needed for responses, entries here will be ignored
Allowed Values for Factors:	“Factor”	Any valid variable name in R, must be unique	“Categorical”, “Continuous”	Comma and space separated string, all-numeric for continuous

3.2. Scope Input File Formatting and Notation

The scope input file defines the scope for model validation in terms of factors (inputs) and responses (outputs), and their allowed ranges and values. All factor and response names must be unique. Note, all fields are required for factors, but factor levels are not required for responses. Scope input files may contain any number of factors and responses. Response and factor level ranges may be given in any convenient units, so long as the units are consistent across all files where that response or factor appears. Ranges for categorical factors are specified by an explicit list of all allowed levels. Categorical responses with more than two levels (binomial) are not currently supported. When specifying the expected distribution type for responses, if the distribution is unknown, select “Continuous”. Any known factor constraints or

explicit assumptions, such as factors held constant or assumed to have a specific value, should be entered in the scope input file as “Factor” with the appropriate name and data type, and only one value specified for “Factor Levels.”

Disallowed factor combinations or non-rectangular constraints on the scope factor space are not currently supported.

3.3. Using Multiple Scope Input Files

For complex scopes of interest or systems that must perform multiple missions, it may be advantageous to define regions of the overall operating scope which pertain to specific missions or operating regions of interest, so that the MVL can be determined relative to each region or mission. This can be accomplished by defining multiple scope input files, one for each mission or operating region within the overall scope of interest. The complete set of scope files may be passed as input parameters for a single run of the MVL tool, which will calculate an independent MVL for each scope region specified.

4. Model and Referent Data Files

Data from the model to be validated and from validation referents are passed to the MVL tool using data input files in Excel (.xlsx) format. Data input file templates for models, referents, and interpolators are available within the MVL Shiny application (section 6.1). The file formats are similar for model, referent, and interpolator data. Formatting characteristics specific to each type of data are discussed in the rest of this section.

The companion paper, Model Validation Levels: Methods and Implementation (Stafford et al., 2024), contains details for the data required to calculate an MVL (see Figure 3). For example, referent points must be replicated and the model and referent data must contain matched inputs to calculate an MVL without interpolator methods.

4.1. Model Data Input File

Model data is provided to the MVL tool via a model data input file. An example model data input file is shown in Figure 2.

	A	B	C	D	E	F	G
1	Name	Level	Deterministic (TRUE / FALSE)				
2	Model	M	TRUE				
3	Factor 1	Factor 2	Response 1	Response 1			
4			Observation	Resolution			
5	1	1		2	0.01		
6	-1	1		0	0.01		
7	-1	-1		-2	0.01		
8							
9							
10							
11							

Figure 2
Example Model Data Input File

The model data input file consists of rectangular data in two sections:

The first section is a table of metadata contained in the first two rows of the file, which consists of the name, the level tag, and the deterministic flag. The name field identifies the name of the model. For model data, the level flag should be set to “M” to distinguish the model data from referent data, which will be input in a similar format. The deterministic flag identifies whether the model generates deterministic (“TRUE”) or stochastic (“FALSE”) performance predictions and is critical for correct handling of the model data.

The second section of the model data input file is a table of the model prediction data at specific factor combinations, contained after the first two rows of the file and using the leftmost columns. The data table consists of columns with headers for each factor and response. Two rows of column headers are used. The first column header row contains the factor and response names, which should exactly match the names supplied in the scope input file. Any factors or responses that are identified in the scope input file as being part of the scope of intended use but are not accounted for in the model should not be included in the model data table. If factors and/or responses are supplied in the model data input file but are not part of the scope of intended use, they will be ignored by the MVL R tool.

Responses are specified using multiple columns, and a second header row below the first is used to distinguish the data columns for each response. Table 2 in Section 4.3 describes different data sub-columns that should be used for different response types. For factors whose data types are given as “Categorical” in the scope input file, provided data must exactly match

one of the specified factor levels for that field, from the scope input file, to be counted toward the MVL.

4.2. Referent Data Input File

Referent data is provided to the MVL tool via a referent data input file. A basic example of a referent data input file is shown in Figure 3. When calculating an MVL using multiple referents, a separate referent data input file is required for each referent.

	A	B	C	D	E	F	G	H	I
1	Name	Level							
2	Referent	5							
3	Factor 1	Factor 2	Response 1	Response 1					
4			Observation	Resolution					
5	1	1	2.1	0.01					
6	1	1	2.29	0.01					
7	-1	1	0.36	0.01					
8	-1	1	0.73	0.01					
9	-1	-1	-1.55	0.01					
10	-1	-1	-1.629	0.01					
11									

Figure 3
Example Referent Data Input File

Referent data input files consist of the same two sections as model data input files. As with model data input files, the metadata section for the referent data input file contains a name field which contains the name of the referent, and a level field, where the level should be an integer value between 1 and 9, inclusive, representing the authority level of the referent. Unlike model data, the deterministic flag is not used for referent data and will be ignored if specified. As with model data input files, the data section of referent input files contains two header rows, one containing the names of the factors and responses, and a second distinguishing the multiple columns associated with each response. The factor and response names provided in the first header row should exactly match those provided in the scope input file. Any factors or responses in a referent file which are not in the scope file will be considered outside of the scope of interest and not included in the MVL calculation.

4.3. Formatting of Response Data

Formatting differs based on the data type, but a response is often captured across multiple

columns. Data type considerations are the expected distribution of data and whether rows contain summarized data or raw data (e.g. single observations). A quick reference for the data formats is given in Table 2, with referenced figures containing examples of each different format type.

For models and referents with multiple responses, a single file should be provided with a set of response columns labeled for each response. The responses do not need to have the same distribution type. The column sets for each response type should be specified as shown in Figures 4-10. File formatting templates for each response type are available from the STAT COE.

Table 2
Response Data Formats for Model, Referent, and Interpolator Files

Response Distribution	Format Description	Format Example
Continuous	Use Normal Option 1 or Option 2 formats.	Figure 4, Figure 5
Normal	Option 1: Raw data with one row for each data point. An optional “Resolution” column specifies the resolution of the response value. If resolution is not specified it will be assumed to be zero. Option 2: Summarized data with average response value at each factor combination. Must include standard deviation. Must include number of data points (trials) at each factor combination. An optional “Resolution” column specifies the resolution of the response value. If resolution is not specified it will be set to zero.	Figure 4, Figure 5
Binomial	Option 1: Raw data with one row for each data point, with factor and response values specified for each run. Entries must be either 1 (indicating success) or 0 (indicating failure). Non-success/fail responses must be mapped to either 0 or 1. Option 2: Summarized data with total number of trials and successes specified for each factor combination. Examples: Success/Fail , Hit/Miss, Classifiers	Figure 6, Figure 7
Exponential	Option 1: Raw data with one row for each event (e.g., failure event), with factor values specified for each, and response values specified as time to event. An optional “Censor” column contains a Boolean flag (1 for censored or 0 for non-censored) that denotes whether the time is for a censored run (stopped before the event could be observed). Option 2: Summarized data where each row summarizes multiple events for a specified factor combination, listing the number of events and the total time observe all events. An optional “Censor” column denotes the number of censored events (stopped before the event could be observed). Examples: Failure Analysis, time until event	Figure 8, Figure 9
Poisson	Summarized data with each row giving the number of events observed over a given time with those factor values, with response columns containing times and event counts. Examples: Event counts over time	Figure 10

	A	B	C	D	E	F	G
1	Name	Level					
2	Normal Referent Option 1	5					
3	Factor 1	Factor 2	Response 1	Response 1			
4			Observation	Resolution			
5	1	1	2.1	0.01			
6	1	1	2.29	0.01			
7	-1	1	0.36	0.01			
8	-1	1	0.73	0.01			
9	-1	-1	-1.55	0.01			
10	-1	-1	-1.629	0.01			
11							

Figure 4
Format Example for Normal Response Option 1

In Normal Response Option 1, each row is a single recorded response (or a single observation), under the heading “Observation”. Optionally, the “Resolution” can be recorded in an additional column. If no resolution column is provided, the resolution will be assumed to be zero.

	A	B	C	D	E	F
1	Name	Level				
2	Normal Referent Option 2	5				
3	Factor 1	Factor 2	Response 1	Response 1	Response 1	Response 1
4			Mean	Standard Deviation	Trials	Resolution
5	1	1	2.195	0.13435	2	0.01
6	-1	1	0.545	0.26163	2	0.01
7	-1	-1	-1.5895	0.05586	2	0.01
8						
9						
10						
11						

Figure 5
Format Example for Normal Response Option 2

In Normal Response Option 2, each row is a summary of multiple observations. Figure 4 and Figure 5 contain the same data, and either option can be chosen to provide the data to the tool. The average observed result is captured under the heading “Mean”. The quantity of observations is captured under the heading “Trials”. Data must include a “Standard Deviation” column. Optionally, the “Resolution” can be recorded in an additional column. This assumes all observations in the row carry the same resolution.

When using Normal Option 2 for a deterministic model which predicts mean and standard deviation (e.g., a regression model), the number of trials column must be present and “2” should be entered for all rows.

	A	B	C	D	E	F	G
1	Name	Level					
2	Binomial Referent Option 1	5					
3	Factor 1	Factor 2	Response 2				
4			Observation				
5		1	1	1			
6		1	1	1			
7		-1	1	0			
8		-1	1	1			
9		-1	-1	0			
10		-1	-1	0			
11							

Figure 6
Format Example for Binomial Response Option 1

In Binomial Response Option 1, each row is a single recorded categoric response under the heading “Observation”. Examples include single recordings from success/fail or hit/miss data, though any two-category response is assessable.

Data must be coded as 1 or 0. It does not matter which category is labeled as a 1 and which is a 0. However, the coding must be consistent between all referent and model files.

This format may also be used for a model that predicts probability of an outcome. In this case, the probability of the outcome of interest can be entered as a number between 0 and 1.

	A	B	C	D	E	F	G
1	Name	Level					
2	Binomial Referent Option 2	5					
3	Factor 1	Factor 2	Response 2	Response 2			
4			Successes	Trials			
5		1	1	2	2		
6		-1	1	1	2		
7		-1	-1	0	2		
8							
9							
10							
11							

Figure 7
Format Example for Binomial Response Option 2

In Binomial Response Option 2, each row is a summary of multiple categoric responses. Figure 6 and Figure 7 contain the same data, and either option can be chosen to provide the data to the tool. One response must be defined as a “Success”, with the total count of successes captured under the heading “Successes”. The total count of observations, regardless of result, is captured under the heading “Trials”. Examples include single recordings from success/fail or hit/miss data, though any two-category response is assessable.

While a category must be designated as a “Success”. It does not matter which category receives this designation. However, the coding must be consistent between all referent and model files. If using both binomial option 1 and option 2, “Successes” in option 2 should correspond with “1” in option 1.

	A	B	C	D	E	F	G
1	Name	Level					
2	Exponential Referent Option 1	5					
3	Factor 1	Factor 2	Response 3	Response 3			
4			Failure Time	Censor			
5		1	1	200	1		
6		1	1	46.45	0		
7		-1	1	6.3	0		
8		-1	1	45.28	0		
9		-1	-1	9.81	0		
10		-1	-1	0.78	0		
11							

Figure 8
Format Example for Exponential Response Option 1

In Exponential Response Option 1, the response is time until an event (often some sort of failure as in reliability analysis), or a time-like variable such as “miles driven” until a car needs repairs. Each row is a single recorded value of time since last event, under the heading “Failure Time”.

Optionally, the response can include censored data where a record may signify a time at which observation concluded with no event. If censored data is included, a “Censor” column must signify which rows are censored data and which are observed events using entries of 1 and 0 respectively.

	A	B	C	D	E	F	G
1	Name	Level					
2	Exponential Referent Option 2	5					
3	Factor 1	Factor 2	Response 3	Response 3	Response 3		
4			Time	Failures	Censors		
5		1	1	246.45	2	1	
6		-1	1	51.58	2	0	
7		-1	-1	10.59	2	0	
8							
9							
10							
11							

Figure 9
Format Example for Exponential Response Option 2

In Exponential Response Option 2, each row is a summary of time until event for multiple events occurring under the same factor inputs. Figure 8 and Figure 9 contain the same data, and either option can be chosen to provide the data to the tool. The total time is recorded under the heading “Time”, and the total number of events (censored or non-censored) is recorded under the heading “Failures”.

Optionally, the response can include censored data where a portion of events in the “Failures” column correspond to times at which observation concluded without observing an event. If censored data is included, a “Censors” column must signify the quantity of events which are censor events.

	A	B	C	D	E	F	G	H
1	Name	Level						
2	Poisson Referent	5						
3	Factor 1	Factor 2	Response 4	Response 4				
4			Time	Count				
5		1	1	5	43			
6		-1	1	4	24			
7		-1	-1	4	39			
8								
9								
10								
11								

Figure 10
Format Example for Poisson Response

In the Poisson response format, the response is a summary of the number of events counted over some time period. The quantity of events recorded is recorded under the heading “Count”. The Time duration is recorded under the heading “Time”. It does not matter what units (minutes/hours/days/etc.) time is recorded with, but all data files must be in the same units for both referents and models.

4.4. Interpolator Data Input File

If using interpolator methods to calculate an MVL as described by Stafford et al. (2024) in Appendix F, interpolator data file(s) must be provided to the MVL R tool. An interpolator serves to predict referent behavior where referent data was not collected and enable direct comparison between the model and referent. The user is responsible for constructing the interpolator from referent data and accepting any assumptions required to construct it. Both the interpolator predictions and the referent data used to construct the interpolator must be provided as separate input files to the MVL R tool. The referent data should use the appropriate referent data format according to Table 2.

A simple example of an interpolator data input file is shown in Figure 11. As with referent data input files, the metadata section for the interpolator data file contains a name field and a level field. The entry in the name field MUST exactly match the entry in the name field for the referent

data input file containing the data used to build the interpolator. This name matching enables the MVL R tool to link the two files together so that the MVL can be calculated. When using multiple referent files, please enter the unique entries for the name filed in each referent, so that the interpolator is linked only to the referent used to build the interpolator. The level field in the interpolator data file must contain “Interpolator” to appropriately designate to the MVL R tool that the file contains interpolator data. The deterministic flag is also not used for interpolator data and will be ignored if specified.

	A	B	C	D	E	F	G	H
1	Name	Level						
2	Referent	Interpolator						
3	Factor 1	Factor 2	Response 1	Response 1	Response 1			
4			Mean	Standard Deviation	Trials			
5	1	1	2.29	0.1	2			
6	-1	1	2.15	0.1	2			
7	-1	-1	2.25	0.1	2			
8								
9								
10								
11								

Figure 11
Example Data Input File for an Interpolator

The data section of interpolator input file also contains two header rows, one containing the names of the factors and responses, and a second distinguishing the multiple columns associated with each response. Again, the factor and response names provided in the first header row should exactly match those provided in the scope input file. The interpolator responses should be the same as those contained in the referent file used to build the interpolator; however, the sub-columns for that response may differ between the two files (e.g., use Normal Option 2 instead of Normal Option 1). The rows of data below the column headers should contain factor combinations matched to those in the model file. This allows model data to be directly compared to interpolator predictions.

5. Data Matching and Validation Point Selection

The referent data provided by the user is combined using Bayesian methods (Stafford et.al., 2024). These methods are applied separately at each referent point. Referent points are selected as unique combinations of factor settings represented in the referent data files. The referent data at each referent point is pooled to determine the pooled referent distribution and

referent authority at that point. If “Continuous” is selected as the response type and more than one referent is provided, the distribution type will be treated as normal for the purposes of referent pooling.

After pooling the referent data, fidelity is calculated at each validation point by the method described in MVL Methods (Stafford et.al., 2024). Validation points are selected as unique factor setting combinations which are present in both the model data file and one or more referent data files (i.e., records resulting from an “inner join” of the model performance data and the combined referent performance data).

Scope volume coverage and density coverage are calculated when continuous factors are present as described in MVL Methods (Stafford et al., 2024). For the calculation of volume coverage, volume is determined using the convex hull covering all validation points, selected as described above. For the calculation of density coverage, the validation points are compared to a reference grid of points generated using a Latin Hypercube method covering the scope of factors. If only categorical factors are present, coverage is calculated as the fraction of combinations covered. If both categorical and continuous factors are present, categorical coverage, average volume coverage, and average density coverage are calculated.

6. Running the MVL Tool

6.1. Shiny Application

Given the MVLs R package has been installed and loaded into R using `library(MVLs)`, the MVL shiny app may be launched from the R instance with the following code:

```
MVL_app()
```

This code launches the app, as pictured in Figure 12. This interactive app has multiple tabs which walk the user through the step-by-step process to upload data, calculate MVLs, and visualize results. Templates are available on the Input Setup tab.

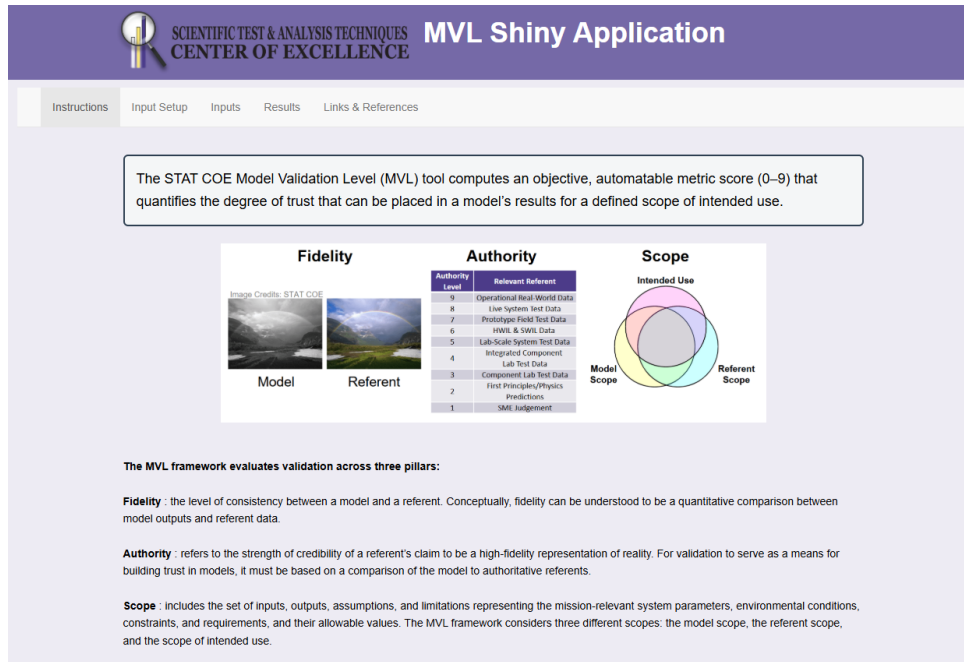


Figure 12
MVL R Shiny Application

6.2. MVL Function

Alternatively to the Shiny app, MVLs may be calculated using the `MVL()` and `MVLa()` functions provided with the R package.

To determine the type of applicable MVL, reference the flowchart in Figure 3 in the MVL Methods and Implementation paper (Stafford et al., 2024). This flowchart can help identify if an MVL or MVL_a (accuracy only MVL) is more appropriate and can help identify if a referent interpolator would be required to calculate the MVL. Typically, an MVL is appropriate for stochastic models and an MVL_a is appropriate for deterministic models. An MVL_a should be calculated with the understanding that the model is not assessed for how well it reflects variability in the referent data.

To calculate an MVL, enter the following code in the R console. Substitute the example file names in quotations with the names of data files, scope files, or file location where appropriate:

```
data_files <- c(
  "model_data_filename.xlsx" ,
  "referent_data_filename_1.xlsx" ,
  "referent_data_filename_2.xlsx" ,
  "referent_data_filename_3.xlsx"
)

scope_files <- c(
  "scope_filename_1.xlsx" ,
  "scope_filename_2.xlsx"
)
```

```
setwd("C:/example_filepath")
result = MVL(data_files,scope_files)
result
```

The example data file names, which include referent, model, and interpolator (if applicable) data files data, are the lines:

```
"model_data_filename.xlsx" ,
"referent_data_filename_1.xlsx" ,
"referent_data_filename_2.xlsx" ,
"referent_data_filename_3.xlsx"
```

When substituting file names, keep quotation marks around file names and commas separating each entry (shown here at the end of each line except the final entry). The file name should include the filetype extension “.xlsx”.

If you have a different number of files than the example, lines can be added or removed so long as quotations and comma format are maintained. At least one model file and one referent file is required to calculate the MVL.

The example scope file names are the lines:

```
"scope_filename_1.xlsx" ,
"scope_filename_2.xlsx"
```

As with data files, substitute scope file names in place of the examples. Keep quotation marks and the filetype extension “.xlsx”. Add commas after all but the final entry. Only one scope file is required to calculate the MVL, but more may be included if desired.

The line below tells R where the files listed above are located (called the working directory). All files (model, referent, interpolator, and scope) should be in this folder.

```
setwd("C:/example_filepath")
```

Replace the example filepath, **C:/example_filepath**, with the location of the folder containing your files, including the name of the folder itself at the end of the file path. Do not remove quotation marks. Alternatively, in RStudio, you can set the working directory under Session (located at the top of the window) > Set Working Directory > Choose Directory.

Once all file names are updated, run all code and an MVL report will be generated.

6.3. *MVL_a Function*

To calculate an MVL_a, enter the following code in the R console. Notably, the only difference between this code and the code in Section 6.1 is in the second-to-last line where the MVL_a function is used instead of the MVL function. Substitute the example file names in quotations with the names of data files, scope files, or file location where appropriate:

```
data_files <- c(
  "model_data_filename.xlsx" ,
  "referent_data_filename_1.xlsx" ,
  "referent_data_filename_2.xlsx" ,
  "referent_data_filename_3.xlsx"
)

scope_files <- c(
  "scope_filename_1.xlsx" ,
  "scope_filename_2.xlsx"
)

setwd("C:/example_filepath")
result = MVL(data_files,scope_files)
result
```

The example data file names, which include referent, model, and interpolator (if applicable) data files data, are the lines:

```
"model_data_filename.xlsx" ,
"referent_data_filename_1.xlsx" ,
"referent_data_filename_2.xlsx" ,
"referent_data_filename_3.xlsx"
```

When substituting file names, keep quotation marks around file names and commas separating each entry (shown here at the end of each line except the final entry). The file name should include the filetype extension “.xlsx”

If you have a different number of files than the example, lines can be added or removed so long as quotations and comma format are maintained. At least one model file and one referent file is required to calculate the MVL.

The example scope file names are the lines:

```
"scope_filename_1.xlsx" ,
"scope_filename_2.xlsx"
```

As with data files, substitute your scope file names in place of the examples. Keep quotation marks and the filetype extension “.xlsx”. Add commas after all but the final entry. Only one scope file is required to calculate the MVL, but more may be included if desired.

The line below tells R where the files listed above are located (called the working directory). All files (model, referent, interpolator, and scope) should be in this folder.

```
setwd("C:/example_filepath")
```

Replace the example file path, **C:/example_filepath**, with the location of the folder containing your files, including the name of the folder itself at the end of the file path. Do not remove

quotation marks. Alternatively, in RStudio, you can set the working directory under Session (located at the top of the window) > Set Working Directory > Choose Directory.

Once all file names are updated, run all code and an MVL_a report will be generated.

7. MVL Output

At execution, the MVL tool reads the provided scope, model, and referent data files. It identifies the factors and ranges provided in the scope file as the scope of intended use and builds a corresponding data frame. The referent data pooling, point matching, and calculation of fidelity, referent authority, and scope coverage and density are performed as described above. For more information on the calculation methodology, see Stafford et al. (2024).

Note that an MVL does not evaluate a model alone, but evaluates a model's validity for a given response, over a given scope. The MVL tool provides separate outputs for each scope and each response (if multiple scopes/responses are provided).

The MVL tool will display two tables summarizing the results for each response and scope combination: a summary table and an improvement table. The summary table displays the resulting MVL as well as lower-level metrics grading fidelity, referent authority, and scope coverage. The improvement table shows the maximum achievable MVL if a given lower-level metric were to be "perfect", as well as how much improvement such an MVL would represent over current results. This table can help guide which actions would have the biggest impact for improving the MVL.

7.1. MVL Output Tables

Example output tables from calculating an MVL are shown in Table 3 and Table 4.

Table 3
MVL Summary Table

Metric	Value
MVL	2.72
MVL _a	3.95
MVL _v	2.96
No. of Validation Points	8.00
Average Fidelity	0.49
Average fa	0.90
Average fv	0.55
Average authority level	5.00
Coverage	0.66
Cv	1.00
Cd	0.66

Table 4

MVL Improvement Table

Metric	Value	Improvement
Fidelity	4.16	1.43
fa	2.96	0.24
fv	3.95	1.23
Authority	6.72	4.00
Coverage	3.57	0.84
Cv	2.72	0.00
Cd	3.57	0.84

The summary table in Table 3 contains the model's overall MVL as well as various descriptive statistics supporting the MVL calculation.

MVL: The Model Validation Level of the model for a given response and scope. This is the main output of the MVL tool.

MVL_a: An "accuracy only" MVL which only accounts for the accuracy element of fidelity.

MVL_v: A "variability only" MVL which only accounts for the variability element of fidelity.

No. of Validation Points: The number of distinct combinations of factor settings which contain sufficient data to calculate fidelity and validate the model.

Average Fidelity: The average fidelity score among validation points, on a scale from 0 to 1.

Average fa: The average "accuracy only" fidelity score among validation points, on a scale from 0 to 1.

Average fv: The average "variability only" fidelity score among validation points, on a scale from 0 to 1.

Average authority level: The average authority level among validation points, on a 9 point scale.

Coverage: The coverage score of the model, on a scale from 0 to 1.

Cv: The volume component of the model's coverage score, on a scale from 0 to 1. Reported only when all factors are continuous.

Cd: The density component of the model's coverage score, on a scale from 0 to 1. Reported only when all factors are continuous.

Ct: The categorical coverage for t-way combinations. E.g., C2 reports the fraction of 2-way categorical combinations that are covered with validation points. Reported when any factors are categorical.

Average Cv: The average of C_v scores calculated for each categorical combination, on a scale from 0 to 1. Reported only when factors are a mix of continuous and categorical.

Average Cd: The average of C_d scores calculated for each categorical combination, on a scale from 0 to 1. Reported only when factors are a mix of continuous and categorical.

The improvement table in Table 4 shows how the MVL might have been higher if a perfect score would have been obtained for different lower-level metrics (Stafford et al., 2024). The improvement column shows the amount of improvement from the original score and can be used to help prioritize MVL improvement options based on which could have the most impact.

7.2. MVL_a Output Tables

Example output tables from calculating an MVL_a are shown in Table 5 and Table 6.

Table 5
 MVL_a Summary Table

Metric	Value
MVL_a	3.95
No. of Validation Points	8.00
Average fa	0.90
Average authority level	5.00
Coverage	0.66
C_v	1.00
C_d	0.66

Table 6
 MVL_a Improvement Table

Metric	Value	Improvement
fa	4.16	0.20
Authority	7.95	4.00
Coverage	4.80	0.84
C_v	3.95	0.00
C_d	4.80	0.84

These outputs are the same elements as detailed in section 7.1 except for the exclusion of the MVL and the elements specific to variability, “ MVL_v ” and “Average fv ”.

7.3. Additional Outputs

When using the `MVL()` or `MVLa()` functions directly, type the following into the console to display additional information.

```
result$
```

A dropdown box will appear showing the calculated elements organized by scope and response. Select the desired entry and type \$ again like the following:

```
result$`SCOPE: scope1.xlsx / RESPONSE: Response 2`$
```

This will display an additional dropdown box containing information regarding scope, coverage, data type, and the fidelity at each validation point. Any of these elements can be selected and then displayed separately, as in the example below.

```
result$`SCOPE: scope1.xlsx / RESPONSE: Response 2`$fidelity_table
```

7.4. Vignette

This section shows a simple vignette of an MVL_a calculation with the required files. See Appendix B for an additional example for a model of a toy catapult.

Scenario: Consider a notional intercontinental ballistic missile system called Next-Generation Long Range Missile, where the response of interest is the time of flight in minutes and the only two factors believed to affect this response are the distance to the target in miles and the launch angle in degrees. Model developers construct a deterministic physics-based simulation of the system.

The scope of intended use file is shown in Figure 13, the model data file is shown in Figure 14, and the referent data file is shown in Figure 15. The model is deterministic, so the model data file only contains one observation for each factor combination (using normal option 1). Since the model is deterministic and contains no replicates, the MVL_a will be calculated. The referent data file (normal option 2) contains the same input combinations as the model file and contains replicates at each point, meaning no interpolator methods need to be used.

Scope.xlsx • Saved

File Home Insert Page Layout Formulas Data Review View Help Acrobat

A14

	A	B	C	D	E	F
1	Factor/Response	Name	Data Type	Factor Levels		
2	Factor	Distance to Target	Continuous	-1, 1		
3	Factor	Launch Angle	Continuous	-1, 1		
4	Response	Flight Time	Normal			
5						
6						
7						
8						
9						
10						
11						

Sheet1

Ready Display Settings 100%

Figure 12
Vignette Scope File

Model.xlsx • Saved

File Home Insert Page Layout Formulas Data Review View Help Acrobat

A15

	A	B	C	D	E
1	Name	Level	Deterministic (TRUE / FALSE)		
2	Model Data	M	TRUE		
3	Distance to Target	Launch Angle	Flight Time		
4			Observation		
5		-1	-1	20.22	
6		1	-1	29.82	
7		-1	1	19.68	
8		1	1	30.04	
9					
10					
11					

Normal 1

Ready Display Settings 100%

Figure 13
Vignette Model File

	A	B	C	D	E	F
1	Name	Level				
2	Referent Data	8				
3	Distance to Target	Launch Angle	Flight Time	Flight Time	Flight Time	
4			Mean	Standard Deviation	Trials	
5	-1	-1	20.54	3.2	5	
6	1	-1	25.39	2.11	5	
7	-1	1	21.03	3.49	5	
8	1	1	26.74	2.52	5	
9						
10						
11						

Figure 14
Vignette Referent File

The results obtained for the MVL_a using the R code below (or equivalently with the Shiny app) are shown in Table 7 and Table 8.

```
result$`SCOPE: scopel.xlsx / RESPONSE: Response 2`$fidelity_table
library(MVLpackage)

data_files<-c(
  "Model.xlsx",
  "Referent.xlsx"
)

scope_files<-c(
  "Scope.xlsx"
)

result=MVL(a)(data_files,scope_files)

result
```

Table 7
Vingette MVL_a Summary Table

Metric	Value
MVL _a	6.19
No. of Validation Points	4.00
Average f_a	0.61
Average authority level	8.00
Coverage	0.66
Cv	1.00
Cd	0.66

Table 8
MVL_a Improvement Table

Metric	Value	Improvement
f_a	7.17	0.97
Authority	7.19	1.00
Coverage	7.03	0.83
Cv	6.19	0.00
Cd	7.03	0.83

7.5. Output Interpretation & Troubleshooting

Some common questions that may arise based on results are described below, including possible causes of issues.

Why is the MVL 0?

- *The number of validation points is zero:* The R package did not find any validation points where the fidelity could be calculated, so the MVL is zero.
 - *Is the average f_v zero?* If the f_a is non-zero, but f_v is zero, it's likely that you don't have model variability quantified in your data. For example, if the model is deterministic or if there are no replicates for stochastic models. For deterministic models, try calculating the MVL_a instead of the MVL. For stochastic models, we recommend obtaining more replicates.
 - *Are the average f_a and f_v both zero?* In this case, it's likely that you don't have referent variability quantified or your referent and model points do not overlap. If this is unexpected, check data files to ensure that factor and response names match exactly. Otherwise, if points do not overlap, you can try rerunning the

model at the same points where referent data is available or construct a referent interpolator and calculate an MVL with the interpolator.

- *The coverage is zero*: the validation points found by the package do not sufficiently cover the space to produce a non-zero coverage score, so the MVL is also zero.
 - Check data files to ensure that factor and response names match exactly.
 - Check that at least some validation points lie within the scope of intended use.
 - If C_V is zero but not C_D , it may be that you do not have enough validation points to have a non-zero interpolation volume. The number of validation points must be at least $d + 1$, where d is the number of continuous factors. This also applies to continuous factor domains within a categorical factor combination. If the number of validation points is greater than or equal $d + 1$, check that points are not colinear or coplanar. For example, three points in a straight line in 2 dimensions does not produce a shape with a non-zero area.
- *The average fidelity, scope, and authority are all nonzero*: the MVL can be reduced to zero with sufficiently poor scores across fidelity, scope, and authority. For example, if the highest authority referent is 2, and the scope coverage and average fidelity scores are 0.5, the MVL would be zero. In this case, lower-level metrics can still be used to assess model risk, and the improvement table shows the impact of model improvement actions.

Why is the MVL/Fidelity/Authority 'NaN'?

- NaN in R refers to “Not a Number.”
- NaN results usually occur due to a divide by zero error. For the MVL calculation, divide by zero errors usually occur when calculating fidelity and can propagate to produce NaN scores elsewhere. If the average f_a is NaN, it's likely because the variability of the referent could not be calculated and was set to zero, resulting in a divide by zero error and NaN fidelity scores. Referent standard deviation cannot be calculated for a referent if no replicates are available at a validation point or if a standard deviation is not provided. If the average f_v is zero, either the model or referent variability could not be calculated. If the average f_a is not NaN, but average f_v is NaN, try calculating an MVL_a instead. If both the average f_a and f_v are NaN, an interpolator or more data collection may be required. More data collection could include obtaining replicates of the same referent authority, or augmenting higher-level data with easier to obtain low-authority data.

8. Contact Information

While we have made every effort to ensure the accuracy of this tool, we DO NOT guarantee that this tool is free from defects. This tool is provided ‘as is’, and you use the tool at your own risk. We make no warranties as to performance, merchantability, fitness for a particular purpose, or any other warranties whether expressed or implied. Under no circumstances shall we be liable for direct, indirect, special, incidental, or consequential damages resulting from the use, misuse, or inability to use this tool. For any questions about the software, MVLs, or for information on planned feature updates, please contact the STAT COE at afit.ens.statcoe@us.af.mil.

References

- Helton, J. C., & Johnson, J. D. (2011). Quantification of margins and uncertainties: Alternative representations of epistemic uncertainty. *Reliability Engineering & System Safety*, 96(9), 1034–1052. <https://doi.org/10.1016/j.ress.2011.02.013>
- Modeling and Simulation Enterprise. (2021, August). Department of Defense Modeling and Simulation Glossary. Retrieved from Department of Defense Modeling and Simulation Enterprise: <https://www.msco.mil/MSReferences/Glossary>
- Pace, D. K. (2015). Fidelity, Resolution, Accuracy, and Uncertainty. Modeling and Simulation in the Systems Engineering Life Cycle, Simulation Foundations, Methods and Applications, 369. Retrieved from https://doi.org/10.1007/978-1-4471-5634-5_10
- Provost, K., Stafford, C., & Jones, N. (2024). MVL R Tool. Tool. Scientific Test & Analysis Techniques Center of Excellence.
- Stafford, C., Provost, K., & Jones, N. (2024). *Model Validation Levels: Methods and Implementation*. Best Practice, Scientific Test & Analysis Techniques Center of Excellence.
- US Department of Defense. (2018). Department of Defense Instruction 5000.61.

Appendix A: Key Definitions

To ensure a common understanding of the subject, the following definitions are used throughout this paper:

binomial data: data consisting of two possible values or a count of how often an event occurs out of a fixed number of trials, following the binomial distribution.

categorical variable: a variable that can take on one of a limited number of possible values, assigned based on some qualitative property.

continuous variable: a variable which can take an uncountable or infinite set of values.

epistemic uncertainty: uncertainty derived from a lack of knowledge about the appropriate value to use for a quantity that is assumed to have a fixed value in the context of a particular analysis (Helton, 2011).

exponential data: data following the exponential distribution, a skewed probability distribution commonly used to model constant-failure-rate reliability.

fidelity: the level of consistency between a model and a referent, defined in the three dimensions of accuracy, repeatability, and resolution.

model: a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process (US Department of Defense, 2018).

model validation level (MVL): an objective, automatable metric that quantifies how much trust can be placed in the results of a model to represent the real world.

modeling and simulation (M&S): the use of models, including emulators, prototypes, simulators, and stimulators, either statically or over time, to develop data as a basis for making managerial or technical decisions (Modeling and Simulation Enterprise, 2021).

normal data: data following a normal distribution, a symmetric distribution with no skew.

Poisson data: data following a Poisson distribution, consisting of the number of events occurring in a fixed interval of time.

referent: a codified body of knowledge representing real system behavior.

referent authority: the strength of credibility of a referent's claim to be a high-fidelity representation of reality.

resolution: the degree of granularity with which a parameter or variable can be determined (Pace, 2015); the width of an interval expressing epistemic uncertainty around an observed data point.

scope: the set of inputs, outputs, assumptions, and limitations representing the mission-relevant system parameters, environmental conditions, constraints, and requirements, and their allowable values.

scope of intended use: the set of dimensions, ranges, and assumptions of the model inputs and outputs needed to represent a system's relevant mission parameters, environmental conditions, constraints, and requirements, combined with the additional constraints imposed by the target modeling environment.

simulation: a method for implementing a model over time (US Department of Defense, 2018).

validation: the process that determines the degree to which a model has fidelity relative to an appropriate referent(s) for a specific intended use.

validation point: a factor combination where both model and referent data are available to validate a response.

validity: the fidelity of a model over a pre-specified scope relative to an appropriate referent(s).

Appendix B: Catapult Model Example

This appendix walks through a complete example of MVL calculation for a physics-based model of a toy catapult. The files required to replicate the results shown here can be requested by emailing afit.ens.statcoe@us.af.mil.

The toy catapult system uses rubber band tension to launch a foam ball and is pictured in Figure B1, which shows settings to control the launch behavior (launch angle, stop position, and tension). The catapult model is a stochastic, physics-based model implemented in Python. The model uses projectile motion physics equations from first principles combined with material properties of the catapult (e.g., rubber band elasticity) to predict the trajectory of the launched ball from the release point to first impact with the ground, assuming the catapult is on level terrain. The model is made stochastic by adding a randomly generated, normally distributed error term to a deterministic launch distance prediction. The objective of the model is to estimate the catapult launch distance under different launch conditions.

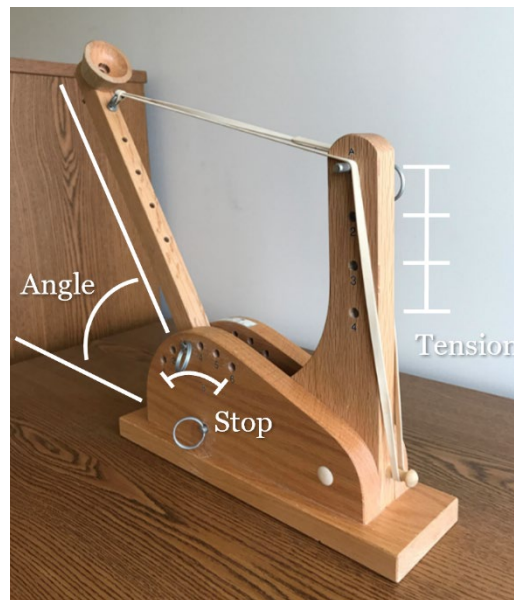


Figure B1
Toy Catapult with Three Launch Factors

The Excel file for scope of intended use is pictured in Figure B2.

	A	B	C	D
1	Factor/Response	Name	Data Type	Factor Levels
2	Factor	Stop	Continuous	1, 4
3	Factor	Tension	Continuous	1, 4
4	Factor	Launch Angle	Continuous	160, 180
5	Response	Distance	Normal	
6				

Figure B2
Scope of Intended Use for Catapult Model

The scope file in Figure B2 names all the factors (stop, tension, launch angle), the factor type for each factor (continuous), and the ranges of those factors for which the model validity will be assessed. All factors are considered continuous here. While Figure B1 shows discrete numeric settings for stop angle and tension, there is an associated physical measurement with each factor setting and they can therefore be treated as continuous variables.

In addition, the scope file in Figure B2 indicates the response of interest (only one response in this case). This response is noted as being normally distributed. As is seen in the data files below, since multiple referents are present and are pooled together, a distributional assumption is required.

Figure B3 shows the model file, while Figures B4 and B5 show the referent files. The figures picture only a subset of the data (first several rows).

The model was used to predict launch distance at all combinations of stop and tension settings for 5° increments of launch angle within the scope of intended use (80 unique combinations), with six replicates at each combination.

For this example, two different referents were used for validation. This toy catapult was used as a teaching tool in design of experiment courses, so data collected by students during course served as one referent, with referent authority level 8 (live system test data). The test data was collected at 53 unique factor combinations, each with 1-15 replicates. Additionally, course instructors (referent authority level 1, subject matter expert judgement) made estimates of catapult launch distance to augment student test data at the 9 factor combinations that only contained one replicate.

1	Name	Level (M or 1-9)	Deterministic (True / False)	
2	Catapult Model	M	FALSE	
3	Stop	Tension	Launch Angle	Distance
4				Observation
5		1	2	160 35.8
6		1	2	160 20.4
7		1	2	160 34.5
8		1	2	160 51.3
9		1	2	160 28.8
10		1	2	160 20
11		1	2	170 65.6
12		1	2	170 77.8
13		1	2	170 52.2
14		1	2	170 88.3
15		1	2	170 66.3
16		1	2	170 69.8
17		1	2	180 121.5

Figure B3
Model Data File for Catapult Model

1	Name	Level (M or 1-9)		
2	Student Test Data	8		
3	Stop	Tension	Launch Angle	Distance
4				Observation
5		1	2	160 30
6		1	2	160 30
7		1	2	160 38
8		1	2	160 20
9		1	2	160 30
10		1	2	160 18
11		1	2	160 19
12		1	2	160 11
13		1	2	160 10
14		1	2	160 9
15		1	2	160 20
16		1	2	160 29
17		1	2	160 20

Figure B4
Student Test Data Referent File for Catapult Model

	A	B	C	D
1	Name	Level (M or 1-9)		
2	Instructor Estimates	1		
3	Stop	Tension	Launch Angle	Distance
4				Observation
5	1	3	170	41
6	2	1	170	99
7	2	3	175	92
8	3	2	175	145
9	3	3	165	97
10	3	3	185	159
11	3	4	175	111
12	4	1	170	169
13	4	3	175	164

Figure B5
Instructor Estimates Referent File for Catapult Model

In each of the model and referent data files, data is provided as a raw observation, though summary statistics (columns for the mean, standard deviation, and number of trials) could also have been used. Additionally, note that model and referent resolution is assumed to be zero for simplicity.

Inputting these files into the MVL Shiny app produces the output shown in Figure B6.

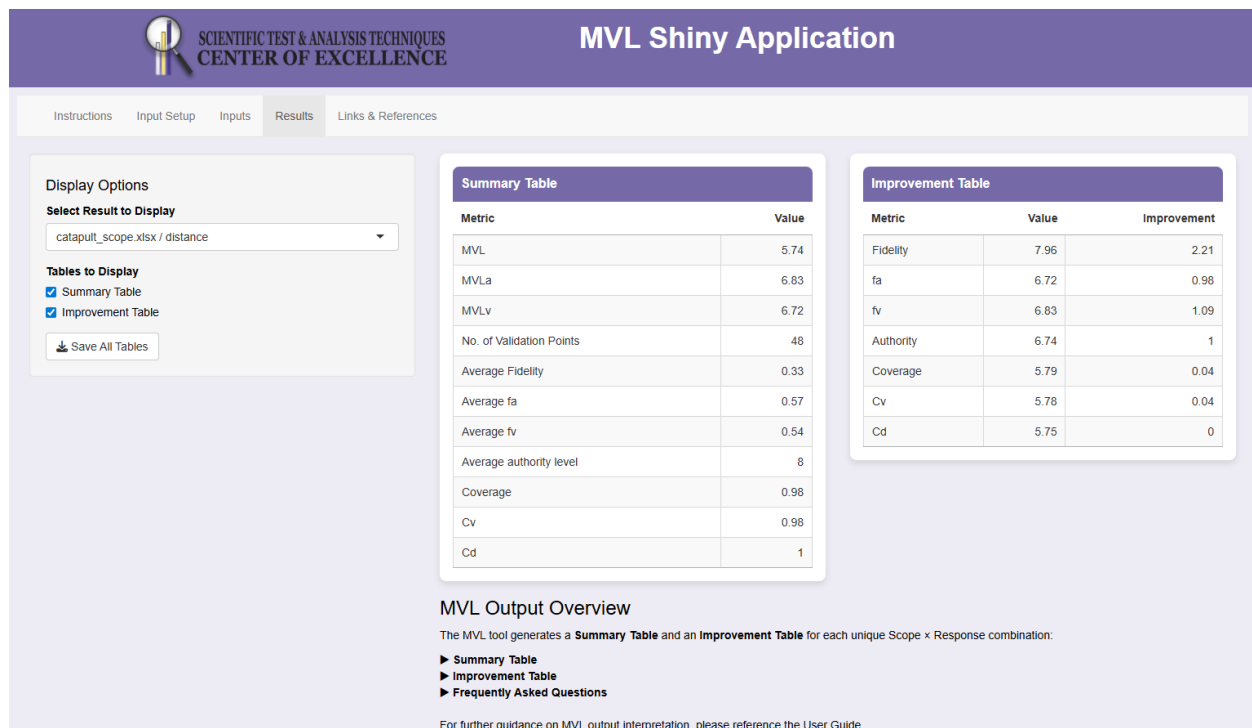


Figure B6
MVL Results for Catapult Model in Shiny Application

Equivalently, the below code may be used to calculate the MVL.

```
data_files <- c("catapult_model.xlsx",
"catapult_referent_1.xlsx",
"catapult_referent_2.xlsx")

scope_file <- c("catapult_scope.xlsx")

result <- MVL(data_files, scope_file)
result
```

Running the above code results in the following output to the R console, producing equivalent results to the Shiny app.

```
> result
$`SCOPE: catapult_scope.xlsx / RESPONSE: Distance`
$`SCOPE: catapult_scope.xlsx / RESPONSE: Distance`$summary_table
```

Metric	Value
MVL	5.74
MVL _a	6.83
MVL _v	6.72
No. of Validation Points	48.00
Average Fidelity	0.33
Average fa	0.57
Average fv	0.54
Average authority level	8.00
Coverage	0.98
Cv	0.98
Cd	1.00

```
$`SCOPE: catapult_scope.xlsx / RESPONSE: Distance`$improvement_table
```

Metric	Value	Improvement
Fidelity	7.96	2.21
fa	6.72	0.98
fv	6.83	1.09
Authority	6.74	1.00
Coverage	5.79	0.04
Cv	5.78	0.04
Cd	5.75	0.00