# Application Menus

The UI comprises a series of drop down menus that provide access to a number of commonly used functions such as file handling, management of run cases, model setup, running and plotting of the results. In addition, Tabs are used to display set-up information of the Cases that have been run. In this manual text in *Red italic* refers to drop down menus and text in *Green italic* refers to Tab titles.

## File

*File>New*: clears any existing model (prompting to save if not already saved) and a popup dialog box prompts for Project name and Date (default is current date).

*File>Open*: existing models are saved as \*.mat files. User selects a model from dialog box.

*File>Save*: save a file that has already been saved.

*File>Save as*: save a file with a new or different name.

*File>Exit*: exit the program. The close window button has the same effect.

## Tools

*Tools>Refresh*: updates *Cases* tab.

*Tools>Clear all>Project*: deletes the current project, including setup parameters and all Cases.

*Tools>Clear all>Figures*: deletes all results plot figures (useful if a large number of plots have been produced).

*Tools>Clear all>Cases*: deletes all cases listed on the  *Cases* tab but does not affect the model setup.

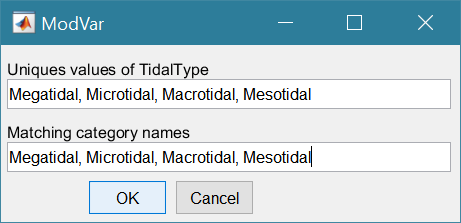
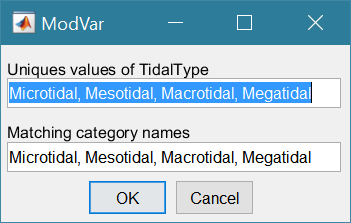
## Project

*Project>Project Info*: edit the Project name and Date.

*Project>Cases>Edit Description*: select a cases description to edit.

*Project>Cases>Edit DS properties*: edit the properties that define the meta-data for a dataset.

*Project>Cases>Edit Data Set*: edit a data set. Initialises a data selection UI to define the record to be edited and then lists the variable in a table so that values can be edited. The user can also limit the data set retrieved based on the variable range and the independent variable (X) or time. This can be useful in making specific edits (eg all values over a threshold or values within a date range). Using the Copy to Clipboard button also provides a quick way of exporting selected data.

*Project>Cases>Modify Variable Type*: select a variable and modify the data type of that variable. Used mainly to make data categorical or ordinal. First select a variable then the data type required. For text data the next UI lists the categories found for the selected data set. The first input can be modified to the required order. The second input can either match the first, or define alternative category names in the order defined for the first input.

If the data are already ordinal, selecting cardinal will remove the ordinal setting. Similarly, if already cardinal, selecting ordinal will set this property. If cardinal or ordinal and the ‘type’ selection matches the current type there are two options: (i) if the top box contains the word ‘order’, the definitions in the ‘Matching category names’ entry box are used to reorder the categories, otherwise (ii) the definitions in the ‘Matching category names’ entry box are used to rename the categories.

*Project>Cases>Save*: select the Case to be saved from the list of Cases, select whether to save the Case as a *dstable* or a *table* and name the file. The dataset *dstable* or *table* are saved to a mat file.

*Project>Cases>Delete*: select the Case(s) to be deleted from the list of Cases and these are deleted (model setup is not changed).

*Project>Cases>Reload*: select a previous model run and reload the input values as the current input settings.

*Project>Cases>View settings*: display a table of the model input parameters used for a selected Case.

*Project> Import/Export>Import*: load a Case class instance from a Matlab binary ‘mat’ file. Only works for data sets saved using Export.

*Project>Import/Export>Export*: save a Case class instance to a Matlab binary ‘mat’ file.

These last two functions can be used to move Cases between projects or models.

**NB**: to export the data from a Case for use in another application (eg text file, Excel, etc), use the *Project>Cases>Edit Data Set* option to make a selection and then use the ‘Copy to Clipboard’ button to paste the selection to the clipboard.

## Setup

The setup menu provides a series of menus to enable different components of the model to be defined.

*Setup>Import Data*: dialog with sub-menu options to Load, Add, Delete, Quality Control. The availability of these options may vary depending on what is defined in the data specific format file.

Select one or more files to load. Once added the current set of variables can be viewed using the *Inputs* tab. When the data has been loaded, the user is prompted to provide a description of the data set (case) and is listed on the *Cases* tab.

*Setup>Import data> Load data*: prompts for file format to be loaded. The options available vary with Data type. The data is then loaded and the user is prompted for a description (working title) for the data set.

*Setup>Import data > Add data*: prompts for file to be added (only one file at a time can be added) and the Case to use (if more than one Case). Only files with the format used to create the data set can be used to Add data to a data record and this is selected when the first file is loaded using the Load menu option.

*Setup>Import data > Delete data*: prompts for Case from which some part of the data is to be deleted.

*Setup>Import data > Data QC*: runs a series of checks on the data. This is only available if defined for the specific data format.

*Setup> Input Parameters*:enter and edit the specified model parameters.

*Setup>Input Data>Model Constants*: various constants are defined for use in models, such as the acceleration due to gravity, viscosity and density of sea water, and density of sediment. Generally, the default values are appropriate (9.81, 1.36e-6, 1025 , 2650 respectively) but these can be adjusted and saved with the project if required.

## Run

*Run> Run Model*: runs model, prompts for Case description which is added to the listing on the *Cases* tab.

Graphical user interface, text, application, email

Description automatically generated*Run> Derive Output*: data that has been added (either as data or modelled values) can be used to derive new variables. The UI allows the user to select data and use a chosen selection of data/variable/range to define either a Variable, XYZ dimension, or Time. Each data set is sampled for the defined data range. If the data set being sampled includes NaNs the default is for these to be included (button to right of Var-limits is set to ‘+N’). To exclude NaNs press the button so that it displays ‘-N’.

The selection is assigned by clicking one of the X, Y or Z buttons. The user is prompted to assign a Variable, XYZ dimension, or Time (the options available varies with the type of variable selected) – see Section 1.9 for details of how this works.

An equation is then defined in the text box below using the x, y, z or t variables[[1]](#footnote-1). Based on the user selection the routine applies the defined variable ranges to derive a new variable. In addition text inputs required by the call and the model object (mobj) can also be passed.

Adding the comment %time or %rows, allows the the row dimension to be added to the new dataset. For example if x and y data sets are timeseries, then a MatlabTM expresion, or function call, call can be used to create a new time series as follows:

x^2+y %time

The output from function calls can be figures or tables, a single numeric value, or a dataset to be saved (character vectors, arrays or dstables). External functions should return the table RowNames (e.g., time or location) as the first variable (or an empty first variable), followed by the other variables to be saved.

If there is no output to be passed back the function should return a string variable.

If varout = 'no output'; this suppresses the message box, which is used for single value outputs. For expressions that return a result that is the same length as one of the variables used in the call, there is the option to add the variable to the input dataset as a new variable. In all there are three ways in which results can be saved:

1. As a new dataset;
2. As an additional variable(s) to one of the input datasets;
3. As an additional variable(s) to some other existing dataset.

For options 2 and 3, the length of the new variables must be the same length (numbere of rows) as the existing dataset.

An alternative when calling external functions is to pass the selected variables as dstables, thereby also passing all the associated metadata and RowNames for each dataset selected. For this option up to 3 variables (plus time if defined for a selected variable) can be selected but they are defined in the call using dst, for example:

[time,varout] = myfunction(dst, 'usertext', mobj);

dst = myfunction(dst, 'usertext’, mobj);

This passes the selected variables as a struct array of dstables to the function. Using this syntax the function can return a dstable, or struct of dstables, or a number of variables. When a dstable, or struct of dstables is returned, it is assumed that the dsproperties have been defined in the function called and dstables are saved without the need to define the meta-data manually.

Some further details on using this option and the ‘**Function**’ library available are provided in Section XX

## Analysis

Plotting and Statistical Analysis both use the standard Data selection UI. These both require Case, Dataset and Variables to be selected from drop-down lists and assigned to a button. Further details of how this works are given in Section 1.9.

### Plotting

*Analysis>Plot menu*: initialises the Plot UI to select variables and produce several types of plot. The user selects the Case, Dataset, and Variable to be used and the plot Type from a series of drop-down lists. There are then buttons to create a New figure, or Add, or Delete variables from an existing figure for 2D plots, or simply a Select button for 3D and 4D plots. The following figures illustrate the options available.

|  |  |
| --- | --- |
|  | **2D plot**  For each selection choose theCase, Dataset and Variable to be used.  > Assign a variable, or a dimension, to the Var and X buttons to set the Y and X axes, respectively  Each selection can be scaled (log, normalised, etc) and the range to be plotted can be adjusted when assigning the selection to a button.  > Select plot type (line, bar, scatter, stem, etc)  Control Buttons:  → : updates the list of Cases  XY : swaps the X and Y axes  + : switches between cartesian and polar plot type  *If polar selected then Ind assumed to be in degrees.* |
|  | **3D plot**  For each selection choose theCase, Dataset and Variable to be used.  > Assign selections to the Var, X and Y buttons  Take care to ensure that the assignments to X and Y correctly match the dimensions selected for the variable (including any adjustment of the dimension ranges to be used).  > Select plot type.  Control Buttons: see 2D plot above. |
|  | **4D**  For each selection choose theCase, Dataset and Variable to be used.  > Assign selections to the Var, X, Y and Z buttons  Take care to ensure that the assignments to X, Y and Z correctly match the dimensions selected for the variable (including any adjustment of the dimension ranges to be used).  > Select plot type.  To produce a new plot, use the Clear button to remove the previous selection.  Control Buttons: see 2D plot above. |

For all plot types, when the data has more dimensions than the plot or animation the user is prompted to sub-select from the data (by selecting sampling values for the dimensions that are not being used).

Animations follow a similar workflow. There are buttons at the bottom of each tab to:

**Run** the selection and create an animation,

**Save** the animation to a file (the animation needs to have been run first) . There is also an option to save on the bottom left of the animation figure.

**Clear** the current selection.

|  |  |
| --- | --- |
|  | **2DT animation**  For each selection choose theCase, Dataset and Variable to be used.  > Assign a variable, or a dimension, to the Var, Time and X buttons.  Each selection can be scaled (log, normalised, etc) and the range to be plotted can be adjusted when assigning the selection to a button.  > Select plot type (line, bar, scatter, stem, etc)  Control Buttons:  → : updates the list of Cases  + : switches between cartesian and polar plot type  *If polar selected, then X assumed to be in degrees and when prompted select Polar and NOT Rose.* |
|  | **3DT animation**  For each selection choose theCase, Dataset and Variable to be used.  > Assign selections to the Var, Time, X and Y buttons  Take care to ensure that the assignments to Time, X and Y correctly match the dimensions selected for the variable (including any adjustment of the dimension ranges to be used).  > Select plot type.  Control Buttons: see 2DT plot above. |
|  | **4DT animation**  For each selection choose theCase, Dataset and Variable to be used.  > Assign selections to the Var, Time, X, Y and Z buttons  Take care to ensure that the assignments to Time, X, Y and Z correctly match the dimensions selected for the variable (including any adjustment of the dimension ranges to be used).  > Select plot type.  Control Buttons: see 2DT plot above. |

***Selection of User plot type***

Calls the user\_plot.m function, where the user can define a workflow, accessing data and functions already provided by the particular App or the muitoolbox. The sample code can be found in the psfunctions folder and illustrates the workflow to a simple line plot using x-y data from the 2D tab and a surface plot using x-y-z data from the 3D tab.

### Statistics

*Analysis> Statistics*: several statistical analysis options have been included within the Statistical Analysis GUI. The tabs are for **General** statistics, **Timeseries**statistics, model comparisons using a **Taylor** Plot, and the generation of a new record based on the statistics over the **Intervals** defined by another timeseries.

**General tab**

The General tab allows the user to apply the following statistics to data loaded in ModelUI:

1. **Descriptive for X**: general statistics of a variable (mean, standard deviation, minimum, maximum, sum and linear regression fit parameters). Only X needs to be defined. The range of the variable can be adjusted when it is assigned to the X button (see Section 1.9). If the variable being used is a multi-dimensional matrix (>2D), the user is prompted to define the range or each additional dimension, or select a value at which to sample. The function can return statistics for a vector or a 2D array.

The results are tabulated on the ***Stats>General*** tab and can be copied to the clipboard for use in other applications.

1. Graphical user interface, application

   Description automatically generated**Regression**: generates a regresion plot of the dependent variable, Y, against the independed variable, X. For time series data, the default data range is the maximum period of overlap of the two records. For other data types the two variables must have the same number of data points. After pressing the Select button, the user is prompted to select the type of model to be used for the regression. The results are output as a plot with details of the regression fit in the plot title.

Chart

Description automatically generated

1. **Cross**-**correlation**: generates a cross-corrleation plot of the reference variable, X, and the lagged variable, X (uses the Matlab ‘xcorr’ function). For time series data, the default data range is the maximum period of overlap of the two records. For other data types the two variables must have the same number of data points. This produces a plot of the cross-correlation as a function of the lag in units selected by the user.
2. **User**: calls the function user\_stats.m, in which the user can implement their own analysis methods and display results in the UI or add output to the project Catalogue. Currently implements an analysis of clusters as detailed for Timeseries data below.

**Timeseries tab**

The Timeseries tab allows the user to select a single Timeseries variable and apply any of the following statistics:

1. **Descriptive**: general statistics of a variable (mean, standard deviation, minimum, maximum, sum and linear regression fit parameters). The results are tabulated in a new window and can be copied to the clipboard for use in other applications.

Graphical user interface, application

Description automatically generatedVarious ‘seasonal’ sub-divisions can be defined. The required option is selected from the table in the UI, by selecting a Syntax cell and then closing the UI.

The next UI prompts for a threshold for calms (values below threshold are deemed to be “calm” conditions) and allows the selected ‘seasonal ‘divisions to be changed (if the desired option is not in the default list), or edited. The divisions can be expressed in several ways, as detailed below:

|  |  |
| --- | --- |
| **Script** | **Result** |
| 1 | Descriptive statistics for the full-time series |
| [1:1:12].’ | Descriptive statistics for the full-time series and monthly values (the .’ creates a column vector). |
| [12,1,2; 3,4,5; 6,7,8; 9,10,11] | Descriptive statistics for the full-time series and seasons based on groupings – Dec-Feb, Mar-May, Jun-Aug, Sep-Nov shown. |

When seasonal statistics are produced with more than 2 seasons a plot is generated. This can be a cartesian or polar plot of the mean values with error bars used to depict +/- one standard deviation. The polar plot maps the year as one revolution.

1. **Peaks**: generates a new timeseries of peaks over a defined threshold. There are three methods that can be selected:

1 = all peaks above the threshold;

2 = the peak value within each up-down crossing of threshold; and

3 = peaks that have a separation of at least ‘*tint*’ hours.

For option 3, the separation between peaks (‘*tint*’) is also be defined in the pop-up gui. This can be used to try and ensure that peaks are independent. The peaks are marked on a plot with the defined threshold. If rejected, new values can be defined. If accepted a new timeseries is added. This has the class of the Data Type that was used as the source timeseries but is not appended to that timeseries because the date/times are a subset of the source.

1. **Clusters**: The selection process is similar to peaks, where the user defines a threshold, selection method and time between peaks (for method 3). In addition, the cluster interval is defined in days. This is the period of time separating two peaks for them to be no longer considered part of a cluster (e.g. if a sequence of storms occurs every few days they will form a cluster. If there is then a gap of, say, 31 days to the next storm, with a cluster time interval of 30 days this would be considered as part of the next cluster). Once a selection has been made, a plot is generated that shows the peaks for each cluster with a different symbol. The user can either choose a different definition, or accept the definition. Once accepted, the results are added as a new timeseries, with the class of the Data Type that was used as the source timeseries. Two values are stored at the time of each peak in the clusters: the magnitude of the peak; and the number of the cluster to which it belongs (numbered sequentially from the start). This allows the data for individual clusters to be retrieved, if required.
2. **Extremes**: The selection process is similar to peaks, where the user defines a threshold, selection method and time between peaks (for method 3). A figure is generated with two plots. The left-hand plot shows the peaks for the defined threshold and the right hand plots shows the mean excess above the threshold (circles), the 95% confidence interval (dotted red lines) and the number of peaks (vertical bars + right hand axis) as a function of threshold. This plot can be used to help identify a suitable threshold for the peak-over-threshold extremes analysis method. The user can either choose a different definition, or accept the definition. Once accepted, the user is prompted to select a plot type. Options are: None; Type 1 – a single return period plot; Type 2 – a composite plot showing the probability, quantile, return period and density plots. See Coles (2001) for further details of the method used and the background to these plots. The results are tabulated on the *Stats/Extremes* tab and can be copied to the clipboard for use in other applications.
3. **Poisson Stats**: user is prompted to select a threshold, method and peak separation (see Peaks above) and the function generates a plot of the peak magnitude, time between peaks (interarrival time) and the duration above the threshold for each peak. The plot shows a histogram of each variable and the exponential pdf derived from the data, along with the value for the fit.
4. **Hurst Exponent**: user is prompted to select from one of 3 methods, which are based on different computation routines taken from the Matlab Forum, as follows:

1 = Chiarello matrix method,

2 = Abramov loop code,

3 = Aalok-Ihlen code and

4 = Aste using unweighted option.

Methods 1 and 2 are similar, whereas method 3 explores the effect of scale and method 4 derives the unweighted generalized Hurst exponent. The main difference between the first two methods is that Abramov uses a different form for S, rather than the Matlab standard deviation function (std).

The Hurst parameter H is a measure of the extent of long-range dependence in a time series (while it has another meaning in the context of self-similar processes). H takes on values from 0 to 1. A value of 0.5 indicates the absence of long-range dependence. The closer H is to 1, the greater the degree of persistence or long-range dependence. H less that 0.5 corresponds to a lack of persistence, which as the opposite of LRD indicates strong negative correlation so that the process fluctuates violently. H is also directly related to fractal dimension, D, where 1 < D < 2, such that D = 2 - H.

This is experimental code (for code see .../muitoolbox/psfunctions/hurst\_exponent.m, hurst\_aalok\_ihlen.m and genhurstw.m) and the user should refer to the background literature for further details. (Di Matteo *et al.*, 2003; Pacheco *et al.*, 2008; Ihlen, 2012; Morales *et al.*, 2012; Sutcliffe *et al.*, 2016; Abramov and Khan, 2017; Antoniades *et al.*, 2021; Brandi and Di Matteo, 2021).

1. **User**: calls user\_stats.m function, where the user can define a workflow, accessing data and functions already provided by the particular App, or the muitoolbox. The sample code can be found in the psfunctions folder and illustrates the workflow to produce a clusters plot. Some code in the header (commented out) shows how to get a time series using the handles passed to the function (obj and mobj). This code would get the same timeseries as the one passed to the function. However, by modifying the ‘options’ variable it is possible to access other timeseries variables.

**Taylor tab**

The Taylor tab allows the user to create a Taylor Plot using 1D or 2D data (e.g timeseries or grids):

A Reference dataset and a Test dataset are selected. Datasets need to be the same length if 1D, or same size if 2D. If the data are timeseries they are clipped to a time-period that is common to both, or any user defined interval that lies within this clipped period. The statistics (mean, standard deviation, correlation coefficient and centred root mean square error) are computed, normalized using the reference standard deviation and plotted on a polar Taylor diagram (Taylor, 2001).

[*The ModelSkill App provides additional tools to test data and the ModelSkill App manual provides further details of the methods used*.]

Selecting New generates a new Taylor Plot. Selecting the Add button adds the current selection to an existing plot and the Delete button deletes the current selection. The Clear button resets the UI to a blank selection.

Graphical user interface, text, application, email

Description automatically generatedOnce New or Add are selected, the user is asked whether they want to plot the skill score (Yes/No). If Yes, then the user is prompted to set the skill score parameters. As further points are added to the plot, this selection remains unchanged (i.e. the skill score is or is not included). To reset the option it is necessary to close and reopen the Statistics UI.

If the number of points in the Reference and Test datasets are not the same the user is prompted to select which of the two to use for interpolation.

Graphical user interface, text, application, email

Description automatically generated

This is the maximum achievable correlation (see Taylor (2001) for discussion of how this is used).

Exponent used in computing the skill score (see ModelSkill manual for details).

Number of points (+/-W) used to define a local window around the ith point. If W=0 (default) the local skill score is not computed.

Local skill score is computed for window around every grid cell (=1), or computes score for all non-overlapping windows (=0)

Window definition to sub-sample grid for the computation of the average **local** skill score. Format is [xMin, xMax, yMin, yMax].

|  |  |
| --- | --- |
| (a) time series skill score plot | (b) grid skill score plot |

The Taylor Plot shows the Reference point as a green cross and the Test points as coloured circles. The legend details the summary statistics and the Case List button generate a table figure listing all the results. These can be copied to the clipboard.

Chart, radar chart

Description automatically generated Taylor diagram legend includes: B – bias; E’’ – normalised RMS difference

Graphical user interface, text, application, email

Description automatically generatedThe normalised standard deviation and correlation coefficient are also given in the Case List table, along with the global skill score, Sg, and the average local skill score, Sl.

**Intervals tab**

The Intervals tab allows the user to compute selected simple statistics of a timeseries variable for the intervals between the times recorded in another timeseries. For example the mean wave energy between beach profile surveys.

The Reference dataset defines the time intervals to be used. As only the times are used it does not matter which variable is selected. The Sample dataset is the timeseries variable to be used. The analysis is run using the Select button and the user is prompted to define the statistical function.

Graphical user interface, application

Description automatically generated

The UI provides a list of the functions that can be used. [*Any function that simply requires a single variable as an input could be used. To modify the list, edit the variable ‘statoptions’ in muiStats.getIntervalStats*.]

Default properties are derived from the source variable and the selected statistic. The user is prompted to confirm, or edit these default properties. The results are saved as a new timeseries dataset in which the variable describes the statistical value for the interval preceding each time.

## Help

The help menu provides options to access the App documentation in the MatlabTM Supplemental Software documentation, or the App manual.

## Tabs

To examine what has been set-up the Tabs provide a summary of what is currently defined. Note: the tabs update when clicked on using a mouse but values displayed cannot be edited from the Tabs.

***Cases***: lists the cases that have been run with a case id and description. Clicking on the first column of a row generates a table figure with details of the variables for the case and any associated metadata. Buttons on the figure provide access the class definition metadata, source information (files input or models used) and any user data (e.g., tables of derived parameters) that is saved with the data set.

***Inputs***: tabulates the system properties that have been set (display only).

***Q-Plot***: displays a quick-plot defined for the class of the selected case (display only).

***Stats***: displays a table of results for any analyses that have been run (can be copied to clip board).

## UI Data Selection

Functions such as Derive Output (1.5), Plotting (1.6.1) and Statistics (1.6.2) use a standardised UI for data selection. The Case, Dataset and Variable inputs allow a specific dataset to be selected from drop down lists. One each of these has been set to the desired selection the choice is assigned to a button. The button varies with application and may be X, Y, Z, or Dependent and Independent, or Reference and Sample, etc. Assigning to the button enables further sub-sampling to be defined if required. Where an application requires a specific number of dimensions (e.g., a 2D plot), then selections that are not already vectors will need to be subsampled. At the same time, the range of a selected variable can be adjusted so that a contiguous window within the full record can be extracted. In most applications, any scaling that can be applied to the variable (e.g., linear, log, relative, scaled, normalised, differences) is also selected on this UI. The selection is defined in two steps:

**Graphical user interface, text, application, email

Description automatically generatedStep 1**.

Select the attribute to use. This can be the variable or any of its associated dimensions, which are listed in the drop-down list.

The range for the selection can be adjusted by editing the text box or using the Edit (Ed) button.

Any scaling to be applied is selected from the drop-down list.+

Press Select to go to the next step or Close to quit.

+ scaling options include Linear; Log; Relative (V-V(x=0)); Scaled (V/V(x=0)); Normalised; Normalised (=ve); Differences; Rolling mean.

The number of variables listed on the UI depends on the dimensions of the selected variable. For each one Select the attribute to use and the range to be applied.

**Graphical user interface, text, application, email

Description automatically generatedStep 2 - Variable only has dimension of time**.

No selection to be made.

Edit range if required.

**Graphical user interface

Description automatically generatedStep 2 - Variable has 3 dimensions but only 2 are needed for the intended use**.

Select the 1st variable to use as a dimension.

Edit range if required.

Select the 2nd variable to use as a dimension.

Edit range if required.

Use the slider or the Edit (Ed) button to set the value of the dimension to use. (A value of t=500 is selected in the example shown).

Press Select to accept the selection made.

[*NB: Only unused dimensions can be selected from the Select drop-down lists. To adjust from the default list this can sometimes require that the second Select list-box is set first to allow the first Select list-box to be set to the desired value.]*

The resulting selection is then detailed in full (including the ranges or values to be applied to all dimensions) in the text box alongside the button being defined.

Note where a variable is being selected as one property and a dimension as a second property, any sub-selection of range must be consistent in the two selections. This is done to allow variables and dimensions to be used as flexibly as possible.

## Accessing data from the Command Window

In addition to the options to save or export data provided by the *Project>Cases>Save* and *Project>Import/Export* options, data can also be accessed directly for use in MatlabTM, or to copy to other software packages. This requires use of the Command Window in MatlabTM, and a handle to the App being used. To get a handle, open the App from the Command Window as follows:

>> myapp = <AppName>; e.g., >> as = Asmita;

Simply typing:

>> myapp

Which displays the results shown in the left column below with an explanation of each data type in the right hand column.

|  |  |
| --- | --- |
| myapp =  <AppName> with properties: | **Purpose** |
| Inputs: [1×1 struct] | A struct with field names that match all the model parameter input fields currently |
| Cases: [1×1 muiCatalogue] | muuiCatalogue class with properties DataSets and Catalogue. The former holds the data the latter the details of the currently held records. |
| Info: [1×1 muiProject] | muiProject class with current project information such as file and path name. |
| Constants: [1×1 muiConstants] | muiConstants class with generic model properties (e.g. gravity, etc). |

To access current model settings, use the following:

>> myapp.Inputs.<InputClassName>

To access the listing of current data sets, use:

>> myapp.Cases.Catalogue

To access imported or model data sets, use:

>> myapp.Cases.DataSets.<DataClassName>

If there are more than one instance of the model output, it is necessary to specify an index. This then provides access to all the properties held by that data set. Two of these may be of particular interest, RunParam and Data. The former holds the input parameters used for that specific model run. RunParam is a struct with fields that are the class names required to run the model (similar to Inputs above). The Data property is a model specific stuct with field names defined in the code for the model class. If there is only a single table assigned this will be given the field name of ‘Dataset’. To access the *dstable* created by the model, use:

>> myapp.Cases.DataSets.<DataClassName>(idx).Data.Dataset

>> myapp.Cases.DataSets.<DataClassName>(idx).Data.<ModelSpecificName>

To access the underlying *table,* use:

>> myapp.Cases.DataSets.<DataClassName>(idx).Data.Dataset.DataTable

The result can be assigned to new variables as required. Note that when assigning *dstables* it may be necessary to explicitly use the copy command to avoid creating a handle to the existing instance and potentially corrupting the existing data.

Abramov V and Khan M K, 2017, A Practical Guide to Market Risk Model Validations (Part II - VaR Estimation). p. 70, <http://dx.doi.org/10.2139/ssrn.3080557>.

Antoniades I P, Brandi G, Magafas L and Di Matteo T, 2021, The use of scaling properties to detect relevant changes in financial time series: A new visual warning tool. Physica A: Statistical Mechanics and its Applications, 565, 10.1016/j.physa.2020.125561.

Brandi G and Di Matteo T, 2021, On the statistics of scaling exponents and the multiscaling value at risk. The European Journal of Finance, pp. 1-22, 10.1080/1351847x.2021.1908391.

Coles S, 2001, An Introduction to Statistical Modeling of Extreme Values, Springer-Verlag, London.

Di Matteo T, Aste T and Dacorogna M M, 2003, Scaling behaviors in differently developed markets. Physica A: Statistical Mechanics and its Applications, 324 (1-2), pp. 183-188, 10.1016/s0378-4371(02)01996-9.

Ihlen E A, 2012, Introduction to multifractal detrended fluctuation analysis in matlab. Front Physiol, 3, p. 141, 10.3389/fphys.2012.00141.

Morales R, Di Matteo T, Gramatica R and Aste T, 2012, Dynamical generalized Hurst exponent as a tool to monitor unstable periods in financial time series. Physica A: Statistical Mechanics and its Applications, 391 (11), pp. 3180-3189, <https://doi.org/10.1016/j.physa.2012.01.004>.

Pacheco J C R, Román D T and Vargas L E, 2008, R/S Statistic: Accuracy and Implementations, 18th International Conference on Electronics, Communications and Computers (conielecomp 2008), IEEE, 10.1109/conielecomp.2008.14.

Sutcliffe J, Hurst S, Awadallah A G, Brown E and Hamed K, 2016, Harold Edwin Hurst: the Nile and Egypt, past and future. Hydrological Sciences Journal, 61 (9), pp. 1557-1570, 10.1080/02626667.2015.1019508.

Taylor K E, 2001, Summarizing multiple aspects of model performance in a single diagram. Journal of Geophysical Research - Atmospheres, 106 (D7), pp. 7183-7192, 10.1029/2000JD900719.

1. Various pre-defined function templates can be accessed using the ‘Function’ button. Alternatively, text can be pasted into the equation box from the clipboard by right clicking in the text box with the mouse. [↑](#footnote-ref-1)