| SpecEval | Equation, VAR, String Proc |
|---|---|

**Produce evaluation report for equation specification(s).**

`speceval` performs evaluation of (forecasting properties of) equation/VAR/identity object(s) and prepares report summarizing the results. It will return spool with tables and graphs that can be used for evaluation of (forecasting) performance of given equation, and/or compare forecasting performance across alternative specifications. See list of outputs with brief description in dedicated table, and separate document ('*Output objects for specification evaluation add-in.pdf*') for detailed explanation and illustration of add-in output objects. Consult separate document ('*SpecEval illlustrated.pdf*') for illustration of model building process which this add-in is meant to facilitate.

## Syntax

{%equation}.speceval(options)

## Options

Options are divided into 5 groups: general options (what for what should be performed), performance metric and graph options (adjusting settings of performance tables and graphs), forecast options (adjusting forecasting procedure), date options (adjusting samples of procedures, tables and graphs) and store options (controlling naming in report spools and workfile as well as what should be stored).

### General options

| | |
|---|---|
| EXEC_LIST=*arg(s)*<br>*(default="normal")* | Space-separated list of procedure components to executed. All possible arguments together with their meaning are listed in separate table below options table.<br>**Note**: Some arguments imply multiple different arguments. All arguments can be also combined with "-" at the beginning to indicate that given component should not be included despite being included as part of group of arguments (e.g. "all –scenarios" will execute all components except for conditional scenario forecasts components). |
| SPEC_LIST=*arg(s)* | List of specifications (=alternative equations ) for which the forecast performance should be evaluated (e.g. "e_mp1 eq mp2"). Used when multiple equations should be evaluated at the same time. If multiple equations are identified the program will produce cross-equation comparison of performance results in addition to individual equation performance results.<br>Can refer to pattern identifying equations by their names, e.g. "eq_*", or list of patterns separated by comma.<br>If multiple equations are evaluated then results for individual equations will be stored with alias (e.g |

| | |
|---|---|
| | "{name}_1").<br>**Note**: The list includes the equation from which the procedure is executed by default. |
| SAVE=*arg*<br>*(default="f ")* | Indicator whether the resulting report(s) should be saved to disk. Arg may be "t" (true), "f" (false), "d" (true with descriptions of outputs) and "user" (issue user dialog).<br>**Note**: Use option 'REPORT_NAME' for setting the name and path and for default names and paths. |
| PROMPT | Indicator that user dialog should be issued. Relevant when executed from program.<br>**Note**: Use 'NOPROMPT' if executing without any option and wish not to have dialog displayed. |

**Performance metric and graph options**

| | |
|---|---|
| HORIZONS_METRICS = *arg(s)* *(default=1,4,8,12,24)* | Comma-separated list of forecast-horizon lengths (number of steps in n-step-ahead forecasts) for which results will be provided. |
| HORIZONS_GRAPH=*ars(s)* *(default=8,24 or forecast horizon lengths if specified)* | Comma-separated list of graph-horizon lengths (number of steps in n-step-ahead forecasts) for which results will be provided. |
| HORIZONS_BIAS=*arg(s)* *(default=8,24)* | Comma-separated list of bias-horizon lengths (number of steps in n-step-ahead forecasts) for which results will be provided. |
| METRICS=*arg(s)* *(default=rmse)* | Comma-separated list of forecast performance metrics to be calculated and reported: “RMSE”, “MAE”, “BIAS”. |
| PERC=*arg* *(default=”auto”)* | Indicator if percentage errors should be used. Arg may be “t” (true), “f” (false) or “auto” (automatically selected based on significance of constant in first-difference equation). |
| | |
| TRANS=*arg* *(default=”level”)* | Transformation to be used in graphs: “level” (level of series = no transformation applied), “growth” (annualized growth rate of series), ”spread” (spread between series and benchmark series specified in option ‘graph_bench’), ”ratio” (ratio between series and benchmark series specified in option ‘graph_bench’), “index” (index of series, with index period specified in option ‘index_period’), “deviation” (deviation from baseline – applies only to scenario graphs, historical graphs report level of series). |
| GRAPH_BENCH=*arg* | Series to use as benchmark when graphing forecasts in terms of spread or ratio. |
| INDEX_PERIOD=*arg* *(default=start of backtest sample)* | Period to use as index when graphing forecasts in terms of index. |
| GRAPH_ADD_BACKTEST=*arg(s)* | Space-delimited list of additional series that should be added into backtesting graphs.<br>**Note**: Can include transformations.<br>**Note**: Can include ‘[R]’ at the end of series to indicate right axis assignment. |
| GRAPH_ADD_SCENARIOS=*arg(s)* | Space-delimited list of additional series that should be added into scenario forecast graphs. |

| | |
|---|---|
| | **Note**: Can include "[CSF]" indicator at the end of mnemonic to indicate that graph should use conditional scenario forecast for given variable. This is relevant when given variable is forecasted by the model due to it inclusion of its equation in 'eq_list_add'.<br>**Note**: Cannot include transformations. |
| INCLUDE_ORIGINAL =*arg (default="t")* | Indicator whether conditional scenario forecast graphs should include original forecast fopr the scenario. Arg may be "t" (true) or "f" (false).<br>**Note**: The original forecast has to be located in the workfile page. If it is not found the program will ignore this option. |
| INCLUDE_BASELINE =*arg (default="t")* | Indicator whether conditional scenario forecast graphs should include original forecast fopr the scenario. Arg may be "t" (true) or "f" (false).<br>**Note**: The original forecast has to be located in the workfile page. If it is not found the program will ignore this option. |
| ADD_SCENARIOS=*arg(s)* | Space-separated list of additional scenarios to be included in the scenario graphs.<br>**Note**: Conditional scenario forecasts can be referred to with prefix "csf" (e.g. "csfs2" will include conditional scenario forecast for scenario s2). The scenario has to be included among the scenarios. |

**Forecast options**

| | |
|---|---|
| BASE_VAR=*arg* | Base variable string.<br>**Note**: Relevant for VAR models. If not specified then the first endogenous variable will be used. |
| OOS = *arg*<br>*(default=*<br>*"t" (for command execution)*<br>*"f" (for GUI execution))* | Indicator if forecasts should be made out of sample. Arg may be "t" (true) or "f" (false). |
| SCENARIOS=*arg(s)* | Coma-separated list of scenario aliases for which conditional scenario forecasts should be created.<br>**Note**: By default first scenario is used as base scenario against which other scenarios are compared, when relevant.<br>**Note**: Additional equations from option 'eq_list_add' are included in the forecasting model. |
| NOSHOCK_SCENARIO=*arg* | Scenario alias for scenario that should be used for constructing no-shock path for shock response graphs.<br>**Note**: If no no-shock scenario is specified then actuals will be used. |
| SHOCK_SCENARIO=*arg* | Scenario alias for scenario that should be used for constructing shock path for shock response graphs.<br>**Note**: If no shock scenario is specified then independent variables/regressors will be shocked by 1 standard deviation. |
| SHOCK_TYPE=*arg*<br>*(default="permanent")* | Type of shock in shock response graphs. Arg may "transitory" or "permanent". |
| AUTO_SELECT=*arg*<br>*(default="f")* | Indicator whether automatic selection should be performed during re-estimation of backtesting equation. Applicable for equations created via AUTOARMA command, and to equations estimated via ARDL approach when removal of perfect collinearity issues should be performed. Applicable for VAR models when automatic lag selection should be performed. Arg may be "t" (true) or "f" (false).<br>In case of multiple specifications one can specify comma separated list with argument for each specification.<br>**Note**: For equations estimated by AUTOARMA the specifications for AUTOARMA have to be stored in as attribute of the equation with name "selection_info". The relevant info is |

| | |
|---|---|
| | "maxar","maxma" and "info" (e.g. "maxar=4,maxma=2,info="aic"). If not specified the default settings will be used.<br>**Note**: For VAR model the specifications for lag length have to be stored in as attribute of the equation with name "selection_info". The relevant info is "maxlag" and "info", with the latter equal to either of LR,FPE,AIC,SC/SIC or HQ. If not specified then maximum lag of 4 and LR will be used. The selection info can also include optional argument "stability", in which case the number of lags will be adjusted down until model which will not produce explosive forecast is found. |
| CUSTOM_REEST=*arg (default="f")* | Indicator whether custom reestiamtion program should be used. Arg may be "t" (true) or "f" (false).<br>**Note**: custom re-estimation should be performed in subroutine 'custom_reestimation' stored in 'custom_ reestimation.prg' program located in the folder from which the master program is executed. As a result the program has to create model object called 'm_speceval' which will be able to produce forecasts for given base variable. See minimal example of such program. By default, the custom re-estimation will produce forecasts from random walk without drift. In addition the program should store also the equations used in each period, stored as {st_spec_name}_reest{!fp}. |
| EQ_LIST_ADD=*arg(s)* | List of additional equations/identities for right-hand-side variables to be included in the forecasting model.<br>**Note**: This option is used for situation when variable of interested is being forecasted within system-of-equations and user wishes to obtain unconditional rather than conditional forecasts. The procedure will provide results only for the dependent variable in the main equation; results for the added variables will not be provided.<br>**Note**: When used in connection with multiple specifications user can specify to use different specifications also for additional equations; in such case the equation name should include "[ALIAS]" at the end of its name, and the alternative specifications for additional equations should by aliased according to alias used for the |

| | |
|---|---|
| | main equation.<br>**Note**: If no equation/identity is identified/located then the program will try to determine the equation/identity from separate model object specified in 'MODEL_NAME_ADD' option; in such situation the underlying variable should be specified.<br>**Note**: By default the equations will be treated as the main equation with respect to in-sample/out-of-sample forecasting. User can change that by appending "[OOS]" or "[IS]" at the end of each additional equation name. |
| MODEL_NAME_ADD=*arg* | Name model object which contains equations/identities for right-hand-side variables specified in 'eq_list_add' option (if any). |
| FORECASTED_IVARIABLES=*arg(s)* | Comma-separated list of independent variables for which forecasts should be used instead of history.<br>**Note**: Forecasts are provided in the workspace with alias '_F{FSTART}', where 'FSTART' is the date corresponding to first forecasts period (e.g. "MRR_F2008M01" for forecast for series MRR starting in January of 2008). |
| SCENARIO_DATALOAD=*arg (default="t")* | Indicator whether missing scenarios data should be loaded from databases. Arg may be "t" (true) or "f" (false).<br>**Note**: Dataload is performed via subroutine 'cfp_scenario_dataload' which should be included in the execution program. See minimal example of such program. |
| ELIMINATE_MULTICOL=*arg (default="t")* | Indicator whether perfect multicollinearity issues should be identified and eliminated. Arg may be "t" (true) or "f" (false). |
| IGNORE_ERRORS=*arg (default="f")* | Indicator whether reestimation error should be ignored. Arg may be "t" (true) or "f" (false). If equal to "t" then errors encountered in estimating equations will be ignored, and when encountered then previous equation will be used. |

**Date options**

| | |
|---|---|
| TFIRST_TEST=*arg* | User-specified first date of forecast performance evaluation.<br>**Note**: If empty the program will use first possible date given the availability of data for lhs and rhs variables. In case of out-of-sample forecasting is used then this first date will be increased by double of the number of estimated coefficients. |
| TLAST_TEST=*arg* | User-specified last date of forecast performance evaluation.<br>**Note**: If empty the program will use first possible date given the availability of data for lhs and rhs variables. |
| TFIRST_GRAPH=*arg (default=TFIRST_TEST)* | User-specified first date of forecast performance graphs. |
| TLAST_GRAPH=*arg (default=TLAST_TEST)* | User-specified last date of forecast performance graphs. |
| Subsamples=*arg(s)* | Comma separated list of sub-samples for which forecast performance should be analyzed in individual fashion. The format is "start_date-end_date", where start and end are dates conforming to the page frequency (e.g. "2008M01-2012M12"). |
| TFIRST_SCENARIOS=*arg* | User-specified first date of conditional scenario forecasts.<br>**Note**: If empty then first period after last historical observation for all regressors will be used.<br>**Note**: Only used when earlier then last historical observation. |
| TLAST_SCENARIOS=*arg* | User-specified last date of conditional scenario forecasts.<br>**Note:** If empty then end of workfile page will be used.<br>**Note**: Only used when earlier then last historical observation. |
| TFIRST_SGRAPH=*arg (default=TFIRST_SCENARIOS-12)* | User-specified first date of scenario graphs.<br>**Note**: Only used when earlier then last historical observation. |
| TFIRST_SHOCKS=*arg* | User-specified first date of shock response.<br>**Note**: If empty then it will be set to period 12 observations before last historical observation for all regressors. |
| TLAST_SHOCKS=*arg* | User-specified last date of shock response. |

| | |
|---|---|
| | **Note**: If empty then period of last historical observation for all regressors will be used. |
| TFIRST_SHOCKGRAPH=*arg (default=TFIRST_SHOCKS-12)* | User-specified first date of shock graphs. |

**Store options**

| | |
|---|---|
| ALIAS_LIST=*arg(s)* | Alias of equation under which the results should be stored. No alias will be used when empty.<br>In case of multiple equations this should be space-separated list of aliases. If it is empty or it has different length then number of identified equations then numerical aliases will be used.<br>Alternatively, user can input "name" as an argument, in which case the alias will be extracted from the name of the equation; this is applicable when specifications have identical name except for alias (e.g. "eq_static1, eq_static2,eq_static5", in which case the aliases will be "1,2 and 5 respectively). |
| USE_NAMES=*arg*<br>*(default="f")* | Indicator whether outputs should use equations names instead of aliases. Arg may be "t" (true) or "f" (false). |
| SPOOL_NAME=*arg*<br>*(default=sp_spec_evaluation)* | Name of report spool.<br>**Note**: In case when multiple specifications are being evaluated then the auxiliary spool is called '{spool_name}_specs'. |
| REPORT_FILE=*arg* | Name/full path for the pdf report file.<br>**Note**: By default the file will be stored in current working fodler under name '{specification_name}_specification_evaluation.pdf' when single equation is being evaluated, and '{variable}_specification_evaluation.pdf' when multiple equations for given variable are being evaluated. |
| KEEP_FORECASTS=*arg*<br>*(default="f")* | Indicator whether forecasts should be stored. Arg may be "t" (true) or "f" (false). |
| KEEP_EQS=*arg*<br>*(default="f")* | Indicator whether equations should be stored. Arg may be "t" (true) or "f" (false). |
| KEEP_OBJECTS=*arg*<br>*(default="f")* | Indicator whether intermediate objects should be stored. Arg may be "t" (true) or "f" (false). |
| KEEP_INFO=*arg*<br>*(default="f")* | Indicator whether objects with information about forecast evaluation process should be stored. Arg may be "t" (true) or "f" (false). |
| KEEP_SETTINGS=*arg*<br>*(default="f")* | Indicator whether setting objects (strings and sclars) should be stored. Arg may be "t" (true) or "f" (false). |

## Examples

The commands

```
equation eq_ip_static.ls dlog(ip) c dlog(gdp)

eq_ip_static.speceval(nodialog)
```

estimate equation object EQ01 and generate forecast performance report using the add-in default settings. The program will create spool called '*sp_forecast_perfromance*' that will contain tables and graphs with forecast performance results (see '*Outputs for specification evaluation add-in.pdf*' for detailed description of outputs).

You may wish to change the default settings. For example, the command:

```
eq_ip_static.speceval(horizons_forecast="1 2 4
8",horizons_graph="8", oos="f", trans="growth",
subsamples="2008q3-2009q4", scenarios="bl su sd")
```

will create results which include 4 forecast horizons (1,2,4 and 8 steps ahead), and one graph horizon (8 steps ahead); all of the forecast will be created in sample; the graphs will be made in terms of growth rate; and the spool will include graphs for one sub-sample (covering period from 2008q3 until 2009q4), as well as graph with 3 scenario forecasts (for scenarios *bl*, *su* and *sd)*.

Alternatively, you may want to compare results for multiple specifications. The commands

```
equation eq_ip_static_dummy.ls dlog(ip) c dlog(gdp)
dum_recess*dlog(gdp)
eq_ip_static.speceval(spec_list="eq_ip_static_dummy",alias=
"normal dummy")
```

will create two equations and then two spools, one with results for each of the two specification separately, and one with results for both specifications together. The second spool will refer to each specification by the specified alias (*normal* for first specification and *dummy* for second).

Document '*Developing forecasting models using SpecEval add-in for Eviews.pdf*' contains multiple additional examples in context of using the add-in as tool for developing forecasting models.

## Execution list arguments

The table below lists the execution list options together with their meaning. For meaning of the object names please see table in next section. Some arguments are equivalent to multiple other arguments as indicated in the table by italics font (e.g. "graphs" is equivalent to "graphs_summary graphs_ss graphs_bias"). Some arguments are applicable only if other settings are included, e.g. scenario graph options are applicable only if scenarios are specified.

| Argument | Function/output meaning |
|---|---|
| Forecasts | Creation of recursive forecasts, including equations |
| Metrics | Table(s) with forecast performance metrics |
| Graphs_summary | Forecast summary graphs |
| Graphs_SS | Graphs for sub-sample forecasts |
| Graphs_bias | Forecast bias graphs |
| *Graphs* | All types of backtest graphs<br>*Equivalent to "graphs_summary graphs_ss graphs_bias"* |
| Scenarios_individual | Individual scenario graphs<br>By default implies both scenarios_level and scenarios_trans |
| Scenarios_all | All conditional scenario graphs<br>By default implies both scenarios_level and scenarios_trans |
| Scenarios_level | Level conditional scenario graphs |
| Scenarios_trans | Transformation conditional scenario graphs |
| Scenarios_multispec | Multiple specification conditional scenario graphs |
| *Scenarios* | All types of conditional scenario graphs<br>*Equivalent to "scenarios_individual scenarios_all scenarios_level scenarios_trans scenarios_multispec"* |
| Shocks_variables | Shock-response graphs (independent variable shocks) |
| Shocks_regressors | Shock-response graphs (regressor shocks) |
| Shocks | Both types of shock-response graphs<br>*Equivalent to "shocks_variables shocks_regressors"* |
| Decomposition | Forecast decomposition graphs for subsample forecasts and/or conditional scenario forecasts |
| Reg_output | Table(s) with adjusted regression output |
| Stability | Graphs with model/coefficient stability |
| *All* | *Equivalent to all above arguments* |

In addition to above arguments there are also several arguments that both imply several of the above arguments and change the *default* settings for some other add-in options (if user specified non-default value then that value will be used.)

| Argument | Function/output meaning |
|---|---|
| Short | Equivalent to inclusion of arguments "*forecasts metrics graphs_summary reg_output*"<br>Sets following defaults:<br>• horizons_metrics="*4 8 24"*<br>• horizons_graph= "12*"* |
| Normal | Equivalent to inclusion of arguments "*forecasts metrics graphs_summary graphs_ss sceanrios_all reg_output*" |
| Long | Equivalent to inclusion of all arguments<br>Sets following defaults:<br>• Metrics="rmse mae bias"<br>• horizons_metrics="1 *2 4 8 16 24"*<br>• horizons_graph= "*4 8 16 24"* |
| Report | Equivalent to inclusion of arguments *"forecasts metrics graphs_summary scenarios_all scenarios_multispec reg_output"*<br>Sets following defaults:<br>• horizons_metrics="1 2 4 8 12 20"<br>• graph_horizons="12"<br>• tfirst_graph="2000Q1"<br>• scenarios="bl s4"<br>• tfirst_sgraph="2005q1"<br>• tlast_scenarios="2040q4"<br>• save="d" |

## List of outputs

All add-in outputs are explained in detail and illustrated in separate document ('*Output objects for specification evaluation add-in.pdf*'), which also contains discussion of the forecasting procedures. Below is table with full list and brief descriptions.

| Object name | Description |
|---|---|
| Regression output table | Adjusted regression output table |
| Coefficient stability graph | Graph with recursive equation coefficients |
| Model stability graph | Graph with recursive lag orders |
| Performance metrics tables | Table with values of forecast performance metrics |
| Performance metrics tables (multiple specifications) | Table with values of forecast performance metrics for given metric for all specifications |
| Forecast summary graph | Graph with all recursive forecasts with given horizons |
| Sub-sample forecast graph | Graph with forecast for given sub-sample |
| Subsample forecast decomposition graph | Graph with decomposition of sub-sample forecast |
| Forecast bias graph | Scatter plot of forecast and actual values for given forecast horizon (Minzer-Zarnowitz plot) |
| Individual conditional scenario forecast graph (level) | Graph with forecast for single scenario and specification |
| Individual conditional scenario forecast graph (transformation) | Graph with transformation of forecast for single scenario and specification |
| All conditional scenario forecast graph | Graph with forecasts for all scenarios for single specification |
| Multiple specification conditional scenario forecast graph | Graph with forecasts for single scenario for multiple specifications |
| Shock response graphs | Graphs with response to shock to individual independent variable/regressor |

## Execution through GUI

The add-in can be also executed through GUI. Due to large number of options the GUI is separated into main dialog, which controls basic options as well as whether dialogs with additional options should be displayed. The additional options are separated into dialog for date options, dialog for store options, and dialog for advanced options related to performance metric and graph options and forecast options.

To facilitate repeated execution thought GUI, there is option to store the current GUI settings (option is located in the store settings dialog). If this option will be chosen then the workfile will contain table 'tb_speceval_gui' will all settings parameters (ordered by appearance in dialogs) and indicator string 'st_load_gui_settings' set equal to "T". If the table will exist and the indicator string will be set to "T" then upon next execution the defaults for the dialog will be loaded from this table. If user wants to return to original defaults then he/she just needs to change the indicator string or delete either of the objects.

## Additional Notes

### Code structure

The code is created in modular fashion as far as my patience allowed. This means that individual tasks are performed by stand-alone subroutines. However, in many cases the subroutines rely on existence of objects created by previous chunk of the code. Similarly, the subroutines are currently undocumented. One day these issues will be fixed...

The add-in is separated into settings program file, execution program file and subroutines program file. This allows developers to repurpose the subroutine(s) outside of the add-in. Full list of subroutines contained in the subroutine program, together with their arguments, is at the top of the subroutine program.

### Partially duplicate add-ins

This procedure is somewhat duplicate with several other add-ins.

Most similar is the user add-in `TSCVAL`, which performs time-series cross-validation of equation. At heart the two procedures are similar, both focused on creation and measurement of precision of recursive forecasts from existing equation. The main difference is that this procedure is primarily focused on building/improving models by leveraging forecasting performance. Correspondingly, the procedure has inbuilt infrastructure to handle multiple equations at a time, facilitating direct comparison of forecasting performance across various specifications; and it focused on graphs of forecasts, rather than just numerical measures of forecast precision. Additionally, it also allows user to store forecasts. Apart from these structural differences due to slightly different focus, the procedure also allows for much more user customization, especially with respect to control of backtesting samples and horizons (including multiple subsamples); in contrast `TSCVAL` is built around the usual train/test samples approach. Apart from this customization, the procedure allows user to control also several other aspects of outputs (e.g. use of transformations in graphs; possibility of in-sample forecasting; custom re-estimation etc.), and includes number of functions not available in `TSCVAL` (e.g. inclusion of additional equations/identities in the forecasting model, possibility to create conditional scenario forecasts, and others). On the other hand, `TSCVAL`, includes more metrics of forecast precision.

Another partially duplicate add-in is add-in `Roll`. This add-in performs rolling or recursive regression estimation for an equation and stores the resulting equations coefficients or other statistics. Like this add-ins, `SpecEval` also performs recursive regression estiamtion, but unlike `Roll` add-in it does not have functionality to perform rolling regression estimation. While `SpecEval` does not store the regression coefficients or other regression statistics, it can store the resulting equation objects that allows user to extract this (and other) information from them via simple loop. In addition, `SpecEval` creates coefficient stability graph, that displays the recursive coefficients together with the error bands.

Finally, `SpecEval` also performs the functions of `SignifCoefs` add-in. Like this add-in the SpecEval creates adjusted regression output table that is color coded. There are two differences. First, `SignifCoefs` gives user flexibility in the coloring scheme, while `SpecEval` does not. On the other hand, `SpecEval` provides color coding also based on positive/negative coefficient values, not just based on significance. It also includes standardized coefficients and variable descriptions, as well as performing several formatting operations.

## Contact Information

Please send any questions, comments, criticisms, or complaints to the dedicated Eviews forum or privately to kamil.kovar@cerge-ei.cz. If you'd like to contribute to the project, please feel free to send a pull request to https://github.com/CrisisStudent/SpecEval.