VaR FXI Model[1] User Manual

Zhuohui Chen[2]
Adeleke Hakeem Adeyemi[3]

1 Introduction

This document outlines the usage of the VaR FXI model developed by the IMF's Monetary and Capital Markets Department. The model is
meant to be used to generate daily forecast on the conational density and related value at risk thresholds.

For theoretical background, see *Foreign Exchange Intervention Rules for Central Banks: A Risk-Based Framework*[4]. Below, descriptions of how to set up the data and configuration parameters, as well as which scripts to run and what outputs to expect are described.

This document does not cover how to use Python or Spyder. For instructions on this, please check online (both Python and Spyder are open source and freely available for download, as are all libraries used in this framework).

2 Model Structure and Python Scripts

This section provides an explanation of the structure of the model package and each stage of the process. For each stage, there is one script to be run, with the behavior of the script being controlled by the parameters set in each script, and possibly by the output of previous stages.

The model packager contains:
- data [folder]

[1] The most crucial part of the code is authored by Mr. Romain Lafarguette
[2] zchen4@imf.org
[3] AAdeyemi@imf.org
[4] https://www.imf.org/en/Publications/WP/Issues/2021/02/12/Foreign-Exchange-Intervention-Rules-for-Central-Banks-A-Risk-based-Framework-50081

- html_report [folder]
- intermediary results [folder]
- modules [folder]
- output [folder]
- step000_descriptive_plot.py
- step001_exogenous_variable_exploration.py
- step002_data_treatment.py
- step003_model_cross_validation.py
- step004A_best_model_combination.py
- step004B_best_density_model.py
- step005_create_html.py

And the whole process of the modelling will be as follows:
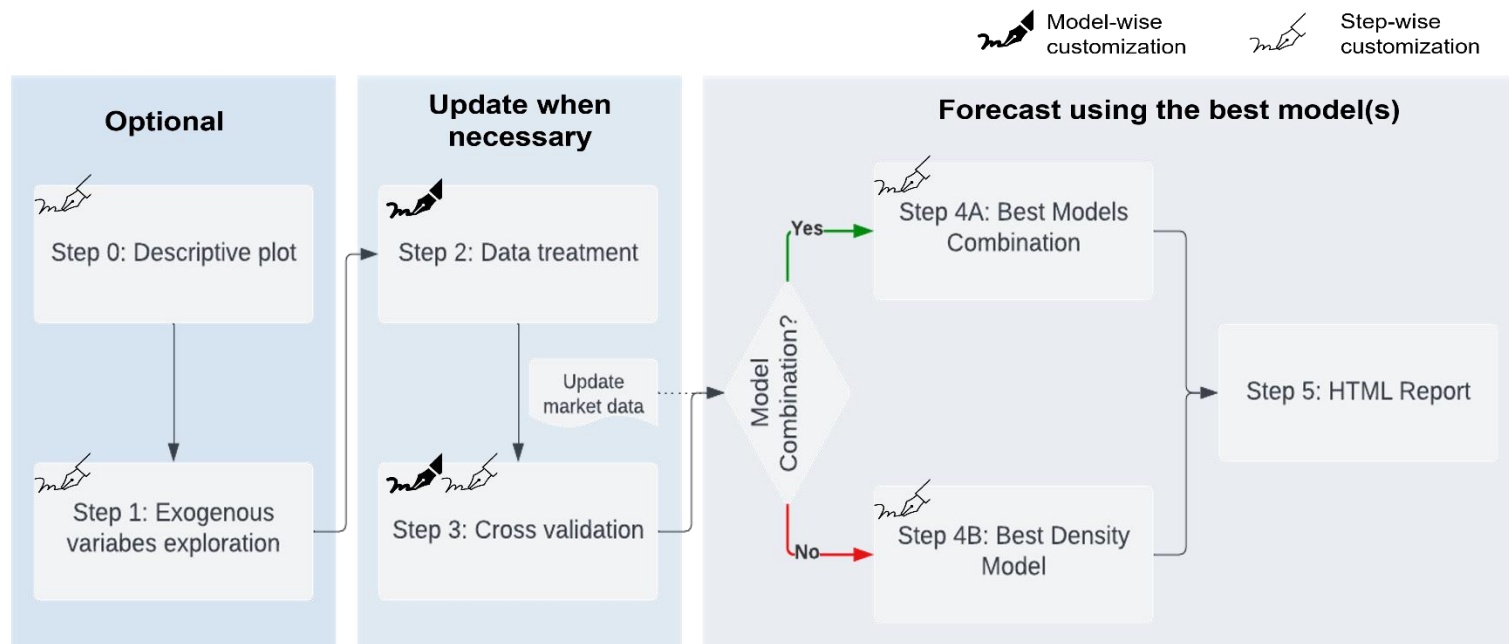
2.1 data [folder]

The folder "data" contains a csv file of underlying data. You can either add into the dataset untreated or treated series. Please align all series with the date columns and do not change the name of the files and the type of the file.

2.2 html_report [folder]

The folder "html_report" stores the output from step005. The "master.html" file works as a landing page to connect different parts of the report.

2.3 intermediary results [folder]

The folder "intermediary results" shelves intermediary outputs from several steps, which will be fed as an input to next steps or the HTML report generation. This is not a folder of users' interests.

2.4 modules [folder]

The folder "modules" hosts the most essentials packages used for the modelling. Please do not change anything in this folder.

2.5 output [folder]

The folder "output" saves all numeric or graphical results generated from step000, step001, step003, step004A and step004B. All documents are prefixed with their step number. And all graphical outputs are saved in both pdf and png format for users' convenience.

2.6 step000_descriptive_plot.py

This is an optional step. Run it when you need a descriptive summary on the FX data. This step will produce a 4-panel chart consisting of a line chart for FX level data, a line chart for FX log return data, a Kernel Density Estimation plot and a Cumulative Distribution Function plot with historic intervention points.

2.7 step001_exogenous_variable_exploration.py

This is an optional step. Run it when you need an exploration on different exogenous variables. This step will produce a regression table showing coefficient, significant status, R squared and other information.

2.8 step002_data_treatment.py

This step should be updated whenever you want to change the treatment on your data. No output will be produced on this step and you do not necessarily run this step. But this step is crucial since the treatment you set in this step will be applied on all steps afterwards. You can also directly include the treated column into the underlying dataset.

2.9 step003_model_cross_validation.py

This step should be updated whenever you think a new model selection exercise is desired. This step will perform the cross validation for various combinations of exogenous variables specifications, volatility models and distributions. The output will be predictive performance tables and the combination pools of models.

2.10 step004A_best_model_combination.py

This step should be executed daily after you update the underlying dataset with new daily entries. This step will forecast the conditional density and conditional mean using the model combination technique. The final forecast on conditional mean and density will be weighted averages of models in each combination pool. Two plots will be generated at this step.

2.11 step004B_best_density_model.py

This step is parallel to step004A_best_model_combination.py and should be executed daily after you update the underlying dataset with new daily entries. This step will forecast the conditional density using the best density model [with the lowest MIS] from the cross-validation step. Several plots, distribution plot, pit plot, fan chart, etc. will be produced on this step.

2.12 step005_create_html.py

This step should be run daily after you finish step004A and step00B unless an HTML report is not desired. 4 HTML files will be stored in the "html_report" folder and the "master.html" file works as a landing page to connect different parts of the report.

3 Parameters Configuration

This section will present all customizable parts of each step. However, details on each customizable parameters will not be explained here but are included as comments inside the Python scripts.

Step	Purpose and Output	Customizable Block	Notes
step000_descriptive_plot.py	• Optional • To generate a 4-panel descriptive plot of the FX series • Output contains a plot	################ #%% Configuration ################	• Step-wise configuration
step001_exogenous_variable_exploration.py	• Optional • To explore exogenous variables with assumed volatility and distribution model • Output contains a regression table	################ #%% Configuration ################	• Step-wise configuration
		############################ #%% Configuration: Data treatment ############################	• Step-wise configuration • You can also include treated columns into the underlying dataset
		################################ #%% Configuration: Model specification ################################	• Step-wise configuration • Steps to create a new model: model7 = ['Correct column name'] models_l = [model7] labels_l = ['A readable label for the model']
step002_data_treatment.py	• Update when necessary • To treat data for all following steps • No output	############################ #%% Configuration: Data treatment ############################	• Model-wise configuration • Please indent your code
step003_model_cross_validation.py	• Update when necessary • To run cross validation for all exogenous specification, volatility models and distributions	################ #%% Configuration ################	• Step-wise configuration
		################################ #%% Configuration: Model specification ################################	• Model-wise configuration: this step will determine the cross

	• Outputs contains predictive performance tables and combination pools		<mark>validation results and the model used for forecast</mark>

Add new daily entries into the underlying dataset

step004A_best_model_co mbination.py	• <mark>Run daily</mark> • To generate forecast using model combination technique • Outputs contain several plots	################ #%% Configuration ################	• <mark>Step-wise configuration</mark>
step004B_best_density_ model.py	• <mark>Run daily</mark> • To generate forecast using the best model in terms of conditional density • Outputs contain several plots	################ #%% Configuration ################	• <mark>Step-wise configuration</mark>
step005_create_html.py	• <mark>Run daily</mark> • To generate HTML reports • Outputs contains serval html reports		• <mark>Please make sure you run both step004A and step004B before running the step005</mark>