EA-MD-QD data processing

Last update: November 30, 2025

Overview

The aim of this short document is to explain the methodology used by the Matlab file *routine_data.m* to process the data contained in the EA-MD-QD dataset.
NEW (11.2025) Also python codes available!

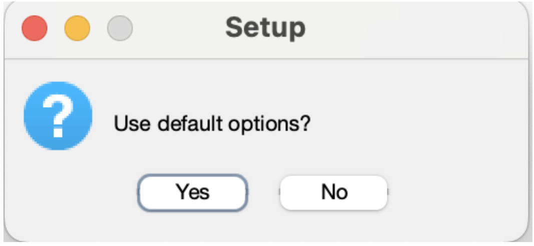
The complete procedure takes 4 steps to run, regarding:

1. Country of interest
2. Frequency of the data
3. Transformations
4. Imputation of outliers/missing values

The user is guided through all of these choices with pop-up windows appearing once the code is run. All subroutines used to perform each of the aforementioned steps is included within the master file. All subroutines employ, if not necessary otherwise, standard functions included in the basic Matlab licence.

The code also provides a default set of options which allow the user to skip all the passages above.

Figure 1: *Default options*



Setup

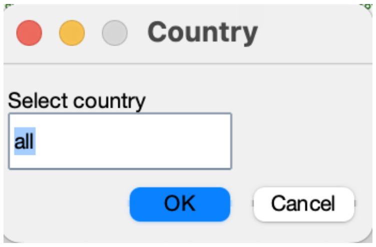
Use default options?

Figure 1 shows the very first pop-up window showing up once the code is run. By choosing *Yes*, the code runs automatically producing a .xlsx file containing data according to the default setup, which will be explained throughout the various sections of this short document. All pop-up windows in the following sections appear whenever default options are overridden by the user, by choosing *No*.

Country choice

If default options are not selected, the first choice the user must undertake is for which country data should be downloaded. Using *default options*, data for all countries are downloaded simultaneously.

FIGURE 2: *Country choice*



Select country
all

Figure 2 shows the pop-up window appearing to the user if default options are not selected. The user can choose to select all countries by manually writing *all*, or can select a specific country, or the Euro Area, by inserting the appropriate 2-digit code. The codes, in alphabetic order, are: *AT*, *BE*, *DE*, *EA*, *EL*, *ES*, *FR*, *IE*, *IT*, *NL*, *PT*.

Frequency choice

If default options are not selected, the user must further choose the frequency at which data must be delivered.

Figure 3 shows the pop-up window appearing to the user. Currently, there are three choices available:

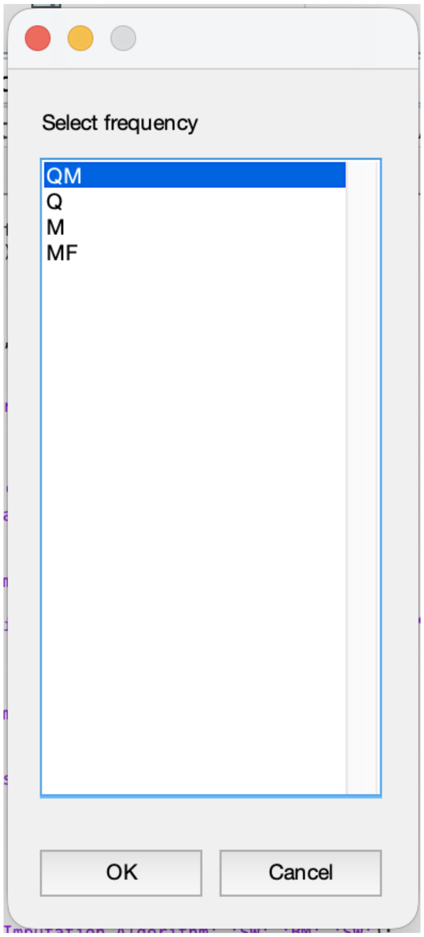
1. **Quarterly-aggregated panel** (*QM*). All series included in the final dataset are at the quarterly level. Monthly series are transformed to quarterly series by standard aggregation, i.e. taking simple averages (for flows) and sums (for stocks) over the months of the corresponding quarter, provided all months related to a quarter are available as data points.

2. **Quarterly panel** (*Q*). Includes only the subset of data originally recorded at the quarterly frequency.

3. **Monthly panel** (*M*). Includes only the subset of data originally recorded at the monthly frequency.

4. **NEW** (11.2025) **Mixed Frequencies** (*MF*). Keep the original frequencies of the series, with quarterly data recorded in the first month of the quarter. (**!** with mixed-frequencies, as the moment, imputation of outliers and missing values is not available).

Figure 3: *Frequency choice*



By choosing default options, the code returns the dataset with all series, where monthly series are aggregated at the quarterly level (QM).

Transformation choice

Updated: 11.2025. If default options are not selected, the user must further choose the kind of transformations to apply to the data.

Figure 4: *Transformation choice*

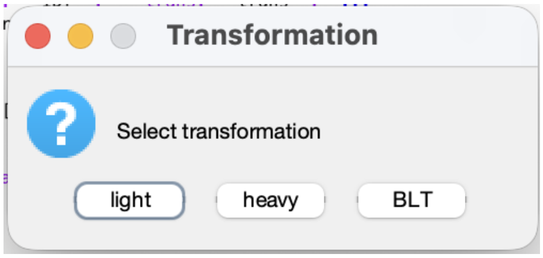


Figure 4 shows the pop-up window appearing to the user. There are three choices available, which correspond to the transformations outlined in Barigozzi et al. (2024). Specifically:

1. **All rates in levels** (*light*). All series are differenced according to standard statistical tests for unit roots with the exception of interest rates and the unemployment rate, hence keeping all rates in levels. Whenever a series is $I(2)$, it is differenced twice.

2. **Statistical transformations** (*heavy*). All series, with no exception, are differenced according to standard statistical tests.

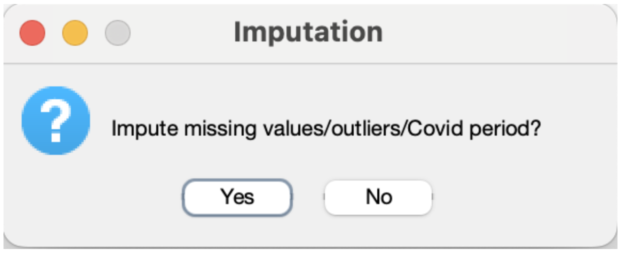
3. **Benchmark transformations** in Barigozzi et al. (2024) (BLT). Replicates the transformations adopted in the paper, where all variables are differenced according to standard statistical tests with the exception of interest rates.

By choosing default options, the code returns the dataset with all series transformed using the set of *light* transformations.

Outliers/missing values imputation

Updated: 11.2025. If default options are not selected, the user must finally choose whether to impute outliers and/or missing values and with which methodology.

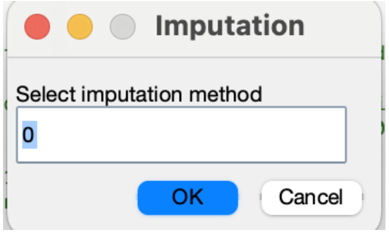
FIGURE 5: *Imputation choice*



Impute missing values/outliers/Covid period?

Figure 5 shows the pop-up window appearing to the user. If *Yes* is selected, another pop-up window will appear, in order to select the methodology for imputation. If *No* is selected, the code runs automatically with no imputation whatsoever.

FIGURE 6: *Methodology choice*



Select imputation method

0

Figure 6 shows the pop-up window appearing to the user for the methodology choice. There are several choices available, which are number-coded. Specifically:

1. Method *-1*. No imputation whatsoever, data are returned with the missing values corresponding to ragged edges, missing data points and observations lost due to transformations,

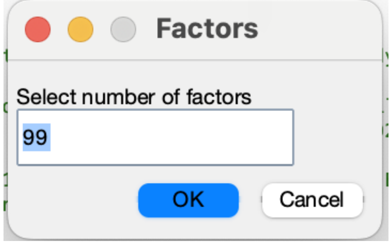
2. Method *0*. No adjustment for outliers, only missing values as the beginning and at the end of the sample (whenever present) are imputed (i.e., ragged edges).

3. Method *1*. Impute both outliers and ragged edges.

4. Method *2*. Impute outliers and missing values via the EM algorithm, by treating the Covid period, i.e. 2020 and 2021 (regardless of the frequency) as missing values for all real variables (R in column **Class** in the data description). Using this procedure, the Covid period is imputed for real variables using only the information contained in other variables, mostly financial and nominal variables.

By choosing default options, method *0* is employed, and only ragged edges are imputed.
A second pop-up box will appear to the user if any imputation is performed. As imputation is based on the EM algorithm, which implies an underlying factor model, the number of factors used for imputation must be chosen.

FIGURE 7: *Number of factors*



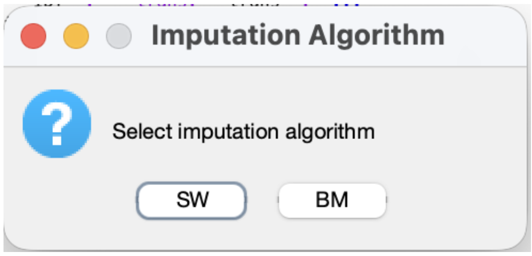
Factors

Select number of factors

99

By choosing $q = 99$ (default) the number of factors is automatically chosen via the criterion of Bai and Ng (2002). Otherwise, if the user has any pre-acquired knowledge on the factor structure for a specific dataset, or if she is willing to assume it, the number of factors can be set directly in the appropriate dialogue box.

FIGURE 8: *Algorithm for imputation*



Imputation Algorithm

Select imputation algorithm

NEW 11.2025 A final pop-up, shown in Figure box will appear to the users if imputation of missing values and/or outliers is selected. This requires the choice of the algorithm to perform imputation. Two choices are now available:

1. The EM algorithm of Stock and Watson (2002). This algorithm, which is the same one employed by (McCracken and Ng, 2016), is based on a *static* factor model, where convergence is based on the common component of the data.

2. **(Beta)** The EM algorithm of Bańbura and Modugno (2014). This algorithm is based on a *dynamic* factor model, and convergence is likelihood based.

Some remarks:

1. The EM algorithm of Bańbura and Modugno (2014) is newly implemented, and still under testing. Although the algorithm yields a superior performance with respect to the algorithm by Stock and Watson (2002), due to the limited presence of missing values in the data results are virtually identical, at least for the transformations and subsets we used in our works.

2. The algorithm of Stock and Watson (2002) is considerably faster, as it only involves principal components, while the algorithm of Bańbura and Modugno (2014) also entails a filtering/smoothing step.

3. The EM algorithm of Bańbura and Modugno (2014) requires an additional hyper-parameter to be set, namely the number of lags p for the VAR on the common factors, necessary to run the Kalman Filter. In the code, this number is not chosen, but rather set to 1. The reason is simple: with some data (e.g., quarterly data) and many factors, the number of parameters could increase considerably with higher order VARs. For this reason, we fix this parameters. This choice can be freely amended by the users with minor changes the matlab codes provided.

4. In running the algorithm of Bańbura and Modugno (2014), we make two other arbitrary choices. First, we set the maximum number of factor to be chosen by test of Bai and Ng (2002) to 6, rather than 15. This, again, since the number of parameters to be estimated is much larger, and higher numbers of factors with few data points would lead to a quick loss of degrees of freedom. Second, we estimate the parameters of the model using only data pre-Covid. It is widely known that adjustments for Covid are needed when estimating a VAR model (see, e.g., Lenza and Primiceri, 2022). This however goes beyond the scope of these codes.

Python Codes

NEW (11.2025). As of November 2025, Python codes to perform all the steps described above are also available to the users. Implementation follows exactly the same steps outlined above for the matlab code. The only difference is that **no pop-up windows will appear**. All the choices will be displayed directly in the Python prompt.

! This first release of the codes has been tested thoroughly, but it is still a beta version. The only difference with respect to the matlab coded is that the only imputation available is through the

EM algorithm by Stock and Watson (2002). We will keep testing and expanding these codes in the following releases.

References

Bai, J. and S. Ng (2002). Determining the number of factors in approximate factor models. *Econometrica 70*, 191–221.

Bańbura, M. and M. Modugno (2014). Maximum likelihood estimation of factor models on datasets with arbitrary pattern of missing data. *Journal of applied econometrics 29*(1), 133–160.

Barigozzi, M., C. Lissona, and L. Tonni (2024). Large datasets for the euro area and its member countries and the dynamic effects of the common monetary policy. *arXiv preprint arXiv:2410.05082*.

Lenza, M. and G. E. Primiceri (2022). How to estimate a vector autoregression after march 2020. *Journal of Applied Econometrics 37*(4), 688–699.

McCracken, M. W. and S. Ng (2016). FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics 34*, 574–589.

Stock, J. H. and M. W. Watson (2002). Macroeconomic forecasting using diffusion indexes. *Journal of business & economic statistics 20*(2), 147–162.