Data and Code for "Interest Rates Under Falling Stars"

Michael Bauer and Glenn Rudebusch

December 27, 2019

Summary

This file describes the contents of the data and code repository for our article "Interest Rates Under Falling Stars" in the American Economic Review (working paper version available here). The repository is available at openICPSR under project ID `openicpsr-115622` (link here). We document the data and code, and provide detailed instructions for replication of all tables and figures in the paper. Full references to the literature are in the paper.

Data

Table 1 describes the files in the folder `data`. Table 2 describes the files with r^* estimates in the folder `data/rstar`; details are in the Online Appendix.

Table 1: Data files

File	Description	Frequency	Source
`PCEPILFE.csv`	Price index for personal consumption expenditures excluding food and energy	Monthly	FRED
`TB3MS.csv`	Three-month Treasury bill rate	Monthly	FRED
`pistar_PTR.csv`	Inflation trend in the Fed's FRB/US model, the "Perceived Target Rate"	Quarterly	Board's website
`yields.csv`	Treasury yields from Grkaynak, Sack, Wright (2007), combined with three-month and six-month Treasury bill rates	Quarterly	GSW yields: Board's website, 3m TB rate: FRED, 6m TB rate: FRED
`bluechip_10y.csv`	Not included – Long-range forecasts for the ten-year Treasury yield from the Blue Chip Financial Forecasts. Expected file format is described in function `loadBlueChip()` in `R/data_fns.R`	Semi-annual	Blue Chip Financial Forecasts

Table 2: Files with r^* estimates

File	Source	Type
delnegro.csv	Del Negro et al. (2017)	Smoothed
delnegro_realtime.csv	Del Negro et al. (2017)	Real-time
jm_smoothed.csv	Johannsen and Mertens (2016)	Smoothed
jm_realtime.csv	Johannsen and Mertens (2016)	Real-time
lw.csv	Laubach and Williams (2016)	Filtered & smoothed
hlw.csv	Holston, Laubach, Williams (2017)	Filtered
kiley.csv	Kiley (2015)	Filtered & smoothed
uc.csv	Unobserved-components model	Smoothed
uc_realtime.csv	Unobserved-components model	Real-time
proxies.csv	Proxies model	Smoothed
proxies_realtime.csv	Proxies model	Real-time
ssm.csv	State-space model	Smoothed
ssm_realtime.csv	State-space model	Real-time

Code

The folder `R` contains the R code required to replicate all tables and figures in the paper. Each code file begins with a short description of its purpose and output.

Computational requirements

- The code was last run with R version 3.5.0.
- The following R packages are required: `KFAS`, `MCMCpack`, `mvtnorm`, `Rcpp`, `RcppArmadillo`, `numDeriv`, `sandwich`, `xtable`, `urca`, `dynlm`, `dplyr`, `VAR.etp`.
  - The file `setup.R` lists the dependencies and installs all missing packages (latest version).
  - To use the package `Rcpp` a C++ compiler is needed. On Windows it is necessary to install Rtools. See here for the Rcpp documentation and here for unofficial Rtools Windows installation instructions.
- The computational costs of the estimation of the *ESE* model are relatively high, but they can be substantially reduced if the MCMC sampling is run in parallel.
  - Running the MCMC sampler for a single chain (100,000 iterations) takes about 10-20 hours on a typical modern computer, depending on its processing power. The results in the paper are based on the combination of five separate chains.
  - To keep the run time manageable, the sampling the MCMC chains can be done with parallel computation on multiple cores (see instructions below).
  - The parallel MCMC chains were run on a Linux high-performance computing (HPC) cluster, specifically on a 16-core Intel server with 300 GB of RAM.

Instructions

1. Ensure that the data files referenced above are stored in the folder `data`, and that the folders `figures` and `tables` exist.
2. Run `setup.R` to install all required R packages and to create folders for output.
3. Run `master.R` to generate all tables and figures in the article and in the Online Appendix.
   - Make sure that your working directory is the root folder, where `master.R` is located.
   - To run the entire script from R, use the command `source("master.R")`. Or from the command line, use the command `Rscript master.R`.

The dynamic term structure models are estimated as follows:

- The *OSE* model is estimated in `R/ose.R`, and the results are saved in `results/ose.RData`.
- The *ESE* model is estimated with a two-step process:

- First, `R/ese.R` is run five times with different random seeds to produce five MCMC chains from different starting values. The default random seed is 616, other random seeds (1-4) are set using a numeric command line argument. Each MCMC chain runs for 100,000 iterations, which takes about 10-20 hours, and the first half of the chain is discarded as a burn-in sample. The resulting chains are saved in the five files `results/ese_SEED.RData` where `SEED` is the random seed.

  In `R/master.R` the five chains are sampled sequentially. But if multiple cores are available, then it is highly recommended to run the chains in parallel. A very simple way to do this is to open up five command line windows, and in each issue one of the following commands:

  ```
  Rscript R/ese.R
  Rscript R/ese.R 1
  Rscript R/ese.R 2
  Rscript R/ese.R 3
  Rscript R/ese.R 4
  ```

- Second, `R/ese_combine.R` is used to combine the five parallel chains into one chain, which is saved as `results/ese.RData`. In addition, some convergence diagnostics are produced, such as a plot of cumulative ergodic means across chains.

Table 3 lists the code files to produce each figure and table in the paper and the required estimation results, if any. Figures are saved in folder `figures`, tables are saved in folder `tables`, and the code is in folder `R`.

The full out-of-sample (OOS) forecast exercise for Table 5 requires the Blue Chip Financial Forecasts, which are not included in the replication files due to licensing restrictions. Thus, `R/oos.R` will not execute unless the user provides these data in `data/bluechip_10y.csv` in the format described in function `loadBlueChip()` in `R/data_fns.R`. To produce only the OOS forecasting results that do not require Blue Chip data—in the top panel of Table 4—the code in `R/oos_nobluechip.R` can be used.

Table 3: Tables and figures

Table/Figure	File	Code
Table 1	coint.tex	coint.R
Table 2	returns.tex returns_subsample.tex	predict.R
Table 3	returns_cycles.tex	predict_cycles.R
Table 4	dtsm_returns.tex	sim_returns.R
Table 5	oos_10y_rmse.tex oos_10y_bluechip_rmse.tex	oos.R
Figure 1	data1.pdf	plot_data.R
Figure 2	data2.pdf	plot_data.R
Figure 3	cycles.pdf	plot_cycles.R
Figure 4	dtsm_istar.pdf	plot_dtsm_figures.R (requires *OSE* and *ESE* results)
Figure 5	dtsm_y10.pdf	same as Figure 4
Figure 6	dtsm_loadings.pdf	plot_ose_loadings.R (requires *OSE* results)
Figure 7	dtsm_tp.pdf	same as Figure 4
Online Appendix		
OA Table 2	persistence.tex	persistence.R
OA Table 3	coint_level.tex	coint.R with command line argument "level"
OA Table 4	returns_mats.tex	predict_mats.R
OA Table 5	returns_R2.tex	predict_R2.R
OA Figure 1	rstar_external.pdf	plot_rstar.R
OA Figure 2	rstar_internal.pdf	plot_rstar.R
OA Figure 3	istar_loadings_humpshape.pdf	plot_ose_loadings.R