# Replication package for "Constructing Fan Charts from the Ragged Edge of SPF Forecasts"


- Todd E. Clark (Federal Reserve Bank of Cleveland)
- Gergely Ganics (Banco de España)
- Elmar Mertens (Deutsche Bundesbank; corresponding author: em@elmarmertens.com www.elmarmertens.com)


This README file describes the replication package for our paper "Constructing Fan Charts from the Ragged Edge of SPF Forecasts."

The replication package contains code and all input data in its raw form, obtained from the original sources as described below.

The materials provided do not necessarily reflect the views of the Federal Reserve Bank of Cleveland, the Federal Reserve System, the Banco de España, the Deutsche Bundesbank, or the Eurosystem.

A copy of this replication package is also maintained at https://github.com/elmarmertens/ClarkGanicsMertensSPFfancharts.

## Data files

Data used for this project comprises SPF survey responses as well as realized values for four different macroeconomic variables listed below, and with additional details described in Section 3 of the paper. All data were obtained from two, publicly available online sources:

1. The Real-Time Data Research Center (RTDRC) at the Federal Reserve Bank of Philadelphia (https://www.philadelphiafed.org/surveys-and-data/real-time-data-research).
2. The FRED database hosted by the Federal Reserve Bank of St. Louis (https://fred.stlouisfed.org) .

From the Real-Time Data Research Center (RTDRC) we obtained SPF mean responses for the following variables (with SPF mnemonics in parentheses as listed on the RTDRC's SPF page, https://www.philadelphiafed.org/surveys-and-data/data-files):

- Level of real GDP/GNP (RGDP)
- Level of the price index for GDP/GNP (PGDP)
- Civilian unemployment rate (UNEMP)
- CPI inflation rate (CPI)

In addition, for GDP growth and GDP inflation, we collected real-time measures of quarter t-1 data as these data were publicly available in quarter t from the quarterly files of real-time data in the Philadelphia Fed's Real-Time Data Set for Macroeconomists (RTDSM) (https://www.philadelphiafed.org/surveys-and-data/real-time-data-research/real-time-data-set-full-time-series-history).

From FRED, we collected realized values for CPI and UNRATE and (mnemonics: CPIAUCSL and UNRATE) using ”final” vintage data. In addition, from ALFRED, the archive site of FRED, we obtained vintage data for real GDP and the GDP deflator to patch in a missing observation in Real-Time Data Set for

Macroeconomists (RTDSM) for 1996Q1; in Real-Time Data Set for Macroeconomists (RTDSM), the observation is missing due to a government shutdown at the time of vintage collection, but became available shortly thereafter, and as collected by ALFRED. The ALFRED files for real GDP and its deflator are GDPC1_2.xls and GDPCTPI_2.xls, respectively.

For forecast evaluation, we measure the outcomes of GDP growth and GDP inflation with the RTDSM vintage published two quarters after the outcome date (that is, we use the quarterly vintage in t+h+2 to evaluate forecasts for t+h made in t; this is the second estimate available in the RTDSM's vintages). Because revisions to quarterly data are relatively small for the unemployment rate and CPI inflation, we simply use the historical time series available in the St. Louis Fed's FRED database to measure the outcomes and corresponding forecast errors for these variables.

The replication package includes copies of the raw input files in the folder rawdataKensington. Below we also describe code that transforms the input data before further processing by our main estimation routines.

## Code

All code used for this project was written in MATLAB. The code has been run on various, recent MATLAB versions (Versions 2022a through 2024a) as well as different operating systems (Linux, Windows and macOS) without the need for any particular adjustments across platforms. The code uses MATLAB's Statistics and Machine Learning Toolbox as well as (optionally) the Parallel Computing Toolbox. The main computational routines were last run on a 32-core Intel virtual client with Intel(R) Xeon(R) Gold 6248R CPU (3.0 GHz) and 112 GB of RAM using MATLAB 2023b.

If available, and prior to launching our routines, enable parallel processing of `parfor` loops to create a parallel pool in MATLAB, which requires the MATLAB Parallel Computing Toolbox; otherwise the loops will be executed sequentially. Choose a number of parallel workers suitable for your computing environment (in terms of available CPU and RAM memory). For example, to use the default setting for your system, simply use `parpool`; in order to use the code with 32 workers use `parpool(32)`. When no `parpool` is created, `parfor` steps will be executed one-by-one. Depending on system defaults in your MATLAB installation, it might also launch `parpool` automatically when encountering the first `parfor` (or `spmd`) command. In principle, it should suffice to launch `parpool` only once per session. But, depending on system defaults, please note that the parallel pool may automatically terminate when idle (and thus needs to be launched again for further use).

The MATLAB code also creates LaTeX files collecting tables and figures produced by the MATLAB code. If a LaTeX installation is present (and if the `pdflatex` command is available on the command line via MATLAB's "system" command), the LaTeX files will also be compiled into PDF files.

Replication of our results proceeds in three steps that are detailed further below:

1. data preparation
2. estimation
3. create tables and figures

The replication code is organized into the following subdirectories:

- rawdataKensington contains raw data files obtained from the sources described above as well as MATLAB files for transforming the raw inputs into a set of `.mat` files (one for each of the variables).

- matdataKensington is a placeholder, to be populated by the `.mat` files generated by the scripts in rawdataKensington.
- mcmcKensington contains various scripts and functions to perform MCMC estimation of our models.
- tablesandfiguresKensington contains various scripts to collect results (as generated by code provided in the mcmcKensington directory) and produce tables and figures.
- matlabtoolbox contains a GitHub submodule with various folders providing different auxiliary `.m`-files (MATLAB scripts and functions) also available at https://github.com/elmarmertens/em-matlabbox. The scripts used in this repository load the appropriate toolbox folders automatically to the MATLAB path as needed. We used the following commit hash of the toolbox: 1e9e6bedfc60b0862183cfafe83883cb744a8330; in GitHub the commit is also tagged as CGMreplicationPackage.

### To prepare input data files for estimation

The directory rawdataKensington contains all of the raw data files obtained from Real-Time Data Research Center (RTDRC) and FRED described above. In addition, the directory contains several `.m` scripts to transform the raw data input files for further processing by our main estimation routines contained in mcmcKensington.

**Replication Step 1:** To process raw data for all variables (RGDP, PGDP, CPI, and UNRATE) please run collectAllData.m. For each variable, a data file is created and stored in MATLAB's `.mat` format. The resulting `.mat` files should be copied into the matdataKensington directory for further use by the estimation routines described below. These `.mat` files are not provided by this replication package and replicators should run collectAllData.m at least once and copy the resulting `*.mat` files from rawdataKensington to matdataKensington.

When updating the data, please adapt the end of the sample in the appropriate scripts as indicated by comments therein (see variable `SPFsamstop` in the script collectData.m and its variants).

*Optional:* To download the most recent source files from RTDRC and FRED researchers can use the python notebook getKensingtonRawData.ipynb that is provided in rawdataKensington as well.

### To estimate the models

Our main estimation routines are contained in mcmcKensington.

**Replication Step 2:** To recompute our results, please execute the following programs.

- kensingtonMDStrendHScycleSVt2blockNoiseHS.m generates out-of-sample estimates from our MDS model with noise in measurement equations for annual forecasts.
- kensingtonVAR0trendHScycleSVt2blockNoiseHS.m generates out-of-sample estimates from our VAR model with noise in measurement equations for annual forecasts.
- kensingtonMDStrendHScycleSVt2block.m and kensingtonVAR0trendHScycleSVt2block.m generate out-of-sample estimates from the noise-free version of our VAR model.
- goMDS.m generates out-of-sample estimates from our VAR model with noise in measurement equations for annual forecasts.

These scripts store MCMC results as .mat files in a newly created subfolder foo. By default, `foo` is created inside the current working directory, which should be mcmcKensington when running the scripts above. To create `foo` in a different directory, please edit the contents of localtemp.m in the matlabtoolbox folder of the replication package).

*Optional:* Additional `go*.m` files in mcmcKensington allow to estimate MDS and VAR models over a single sample, or to compare output from estimates of two model versions.

*Optional:* mcmcKensington also contains the bash scripts goparbatch.sh and goseqbatch.sh that can be used to launch a sequence of multiple MATLAB scripts from the shell. Both shell scripts expect that the names of the MATLAB scripts to be executed should be passed as argument list. Both shell scripts execute the arguments in sequence *and in separate MATLAB sessions*. In case of goparbatch.sh, the MATLAB session opens a parallel pool. For example, the shell command `sh gobatch.sh kensingtonMDStrendHScycleSVt2blockNoiseHS.m kensingtonVAR0trendHScycleSVt2blockNoiseHS.m` will launch a command line session of MATLAB, start a parallel pool, and then execute kensingtonVAR0trendHScycleSVt2blockNoiseHS.m; afterwards, a new MATLAB session is reopened for execution of kensingtonVAR0trendHScycleSVt2blockNoiseHS.m etc. (The shell script works with as many command line arguments as supported by bash and has been written for use on macOS and Linux.) Alternatively, MATLAB scripts can, of course, also be called interactively on the MATLAB GUI's command line. As noted above: In Linux, these shell scripts assume that the command `matlab` is on the shell path. In macOS, each of these shell scripts defines an alias `matlab` that points to the installed MATLAB version. To adapt this setting to your environment, please edit line 12 in goparbatch.sh and goseqbatch.sh as needed.

### To generate tables and figures

The directory tablesandfiguresKensington provides scripts to generate LaTeX tables and figures (as shown in our paper and its appendix). These scripts assume that the `.mat` files with MCMC results generated by the code discussed in the previous section are stored in a common directory (indicated by the variable `resultsdir` in each script); by default, the estimation code discussed above stores results in `.mat` format in a newly created subfolder `foo` in mcmcKensington.

The scripts in tablesandfiguresKensington store their outputs (in `*tex` or `*.eps` format) in a newly created subfolder `foo` (by default `foo` is created inside the current working directory. To create `foo` in a different directory, please edit the contents of localtemp.m in the matlabtoolbox folder of the replication package).

**Replication Step 3:** To create the figures shown in our paper, run the following scripts and collect their outputs as detailed below:

| Tab/Fig | script | output file |
|---|---|---|
| Tab.1 | prettytableZfcst.m | table-ZhatZtp1-slopes-fullsampleSince1990.tex |
| Tab.2 | tabulateRelativeEvalstats.m | relative-RMSEandCRPS-MDStrendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4-vs-VAR0trendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4-fullsampleSince1990.tex |

| Tab/Fig | script | output file |
|---|---|---|
| Tab.3 | prettytableCoverageRates.m | table-CoverageRateBands-VAR0trendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4-fullsampleSince1990.tex |
| Fig.1a | mcmcKensington/goMDS.m | fanchartSEP-RGDP-MDStrendHScycleSV2blockNoiseHS-NgapBOP-samStart1968Q4-2024Q1-thin1-WITHLEGEND.pdf |
| Fig.1b | mcmcKensington/goMDS.m | fanchartSEP-UNRATE-MDStrendHScycleSV2blockNoiseHS-NgapBOP-samStart1968Q4-2024Q1-thin1.pdf |
| Fig.1c | mcmcKensington/goMDS.m | fanchartSEP-CPI-MDStrendHScycleSV2blockNoiseHS-NgapBOP-samStart1968Q4-2024Q1-thin1.pdf |
| Fig.2a | figuresFanCharts.m | YpredictivedensitySPF-RGDP-MDStrendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4-2024Q1.pdf |
| Fig.2b | figuresFanCharts.m | YpredictivedensitySPF-UNRATE-MDStrendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4-2024Q1-WITHLEGEND.pdf |
| Fig.2c | figuresFanCharts.m | YpredictivedensitySPF-CPI-MDStrendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4-2024Q1.pdf |
| Fig.2d | figuresFanCharts.m | YpredictivedensitySPF-PGDP-MDStrendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4-2024Q1.pdf |
| Fig.3a | figuresYtermstructures.m | termstructure-MDS-RGDP-Q1-Until2019Q4-WITHLEGEND.pdf |
| Fig.3b | figuresYtermstructures.m | termstructure-MDS-UNRATE-Q1-Until2019Q4.pdf |
| Fig.3c | figuresYtermstructures.m | termstructure-MDS-CPI-Q1-Until2019Q4.pdf |
| Fig.3d | figuresYtermstructures.m | termstructure-MDS-PGDP-Q1-Until2019Q4.pdf |
| Fig.4a | figuresYtermstructures.m | uncertainty-MDS-RGDP.pdf |
| Fig.4b | figuresYtermstructures.m | uncertainty-MDS-UNRATE.pdf |
| Fig.4c | figuresYtermstructures.m | uncertainty-MDS-CPI-pdf |
| Fig.4d | figuresYtermstructures.m | uncertainty-MDS-PGDP.pdf |
| Fig.5a | figuresPITcdf.m | PITs-RGDP-h4-fullsampleSince1990-y1q4-WITHLEGEND.pdf |
| Fig.5b | figuresPITcdf.m | PITs-UNRATE-h4-fullsampleSince1990-y1q4.pdf |

| Tab/Fig | script | output file |
|---|---|---|
| Fig.5c | figuresPITcdf.m | PITs-RGDP-h8-fullsampleSince1990-y1q4.pdf |
| Fig.5d | figuresPITcdf.m | PITs-UNRATE-h8-fullsampleSince1990-y1q4.pdf |
| Fig.5e | figuresPITcdf.m | PITs-RGDP-h12-fullsampleSince1990-y1q4.pdf |
| Fig.5f | figuresPITcdf.m | PITs-UNRATE-h12-fullsampleSince1990-y1q4.pdf |
| Fig.6a | figuresSEPfanchartsUncertainty.m | ERRORBANDsepfancharts-hh1-RGDP-MDStrendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4-WITHLEGEND.pdf |
| Fig.6b | figuresSEPfanchartsUncertainty.m | ERRORBANDsepfancharts-hh1-UNRATE-MDStrendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4.pdf |
| Fig.6c | figuresSEPfanchartsUncertainty.m | ERRORBANDsepfancharts-hh1-CPI-MDStrendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4.pdf |
| Fig.6d | figuresSEPfanchartsUncertainty.m | ERRORBANDsepfancharts-hh2-RGDP-MDStrendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4.pdf |
| Fig.6e | figuresSEPfanchartsUncertainty.m | ERRORBANDsepfancharts-hh2-UNRATE-MDStrendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4.pdf |
| Fig.6f | figuresSEPfanchartsUncertainty.m | ERRORBANDsepfancharts-hh2-CPI-MDStrendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4.pdf |
| Fig.6g | figuresSEPfanchartsUncertainty.m | ERRORBANDsepfancharts-hh3-RGDP-MDStrendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4.pdf |
| Fig.6h | figuresSEPfanchartsUncertainty.m | ERRORBANDsepfancharts-hh3-UNRATE-MDStrendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4.pdf |
| Fig.6i | figuresSEPfanchartsUncertainty.m | ERRORBANDsepfancharts-hh3-CPI-MDStrendHScycleSVt2blockNoiseHS-y1q4-NgapBOP-samStart1968Q4.pdf |

Note that the code to create Figure 1 can be found in mcmcKensington; all other scripts to create tables and figures are in tablesandfiguresKensington.