## HDLPrepro: Reproducing Adamek, Smeekes, Wilms (2023)

This R package contains code and datasets to reproduce the simulations and applications in Adamek, Smeekes, Wilms (2023) Local Projection Inference in High-Dimensions.

## Installation

This package can be installed with the 'remotes" package:

```
install.packages("remotes")
```

You can then install the HDLPrepro package from the GitHub depository directly by running:

```
remotes::install_github("RobertAdamek/HDLPrepro")
```

Note that as the package contains C++ code, installing the package from source might require some additional tools depending on your system; see the details for Windows and Mac.

This package also makes heavy use of our package `desla`, which is available on CRAN. Alternatively, the latest version of this package can be installed from the GitHub depository by running

```
devtools::install_github("RobertAdamek/desla")
```

## Usage

In addition to a collection of functions and datasets, this package contains a several R scripts which reproduce different parts of our paper. These scripts can be found in the `inst/replication_scripts` folder of the package. They are easily accessible from the GitHub page https://github.com/RobertAdamek/HDLPrepro, or by unpacking the package tarball with `untar()`.

These scripts make use of several packages other than `HDLPrepro` and `desla`. For completeness, we list all used packages here, as well as the package versions with which we ran the code ourselves:

- `bigtime` (0.2.2)
- `desla` (0.3.0)
- `dplyr` (1.1.2)
- `ggpattern` (1.0.1)
- `ggplot2` (3.4.2)
- `ggpubr` (0.6.0)
- `HDLPrepro` (1.0.0)
- `readxl` (1.4.3)
- `reshape2` (1.4.4)
- `xtable` (1.8-4)

We expect these packages to remain backwards compatible, and the most up-to-date versions can be installed with

```
install.packages(pkgs = c("dplyr", "ggpattern", "ggplot2", "ggpubr", "readxl", "reshape2"))
```

Alternatively, it is possible to install the versions used in this replication package using the `remotes` package. For example, to install `ggpattern` v1.0.1, use

```
remotes::install_version("ggpattern", version = "1.0.1", repos = "https://cloud.r-project.org")
```

We also plan to maintain this package to be compatible with future versions of R, but if this is not the case, we ran all applications and simulations on R version 4.3.1. When this is no longer the most recent version of R, past versions of R can be obtained from https://cran.r-project.org/.

If one wants to run **all** the code related to the paper, the script `run_all.R` will execute all individual scripts in correct order and create a collection of files and plots. The user must specify a folder where these

outputs will be stored by changing the command `setwd("your/path/here")` to point to a local folder. An example of correct syntax on Windows is `setwd("D:/reproduction_files")`. The created files will take up approximately 75MB.

The individual scripts included are as follows:

- Simulations in Section 3.1: *Sparse Structural VAR Model*
  - `running_simulation3_1.R`: Runs the simulation and saves intermediate files in a user-chosen folder.
  - `plotting_simulation3_1.R`: Processes these files and generates the plots in our paper.
- Simulations in Section 3.2: *Empirically Calibrated Dynamic Factor Model*
  - `processing_FREDMD.R`: Processes the data obtained from the FRED-MD database. It includes our data transformations which differ from the "default" choices, and our classification of "fast" and "slow" variables.
  - `running_simulation3_2.R`: Uses the estimated DFMs to simulate data and estimate the impulse responses. Saves intermediate files in a user-chosen folder.
  - `plotting_simulation3_2.R`: Processes the simulation files and generates the plots in our paper.
- Application in Section 4.1: *Impulse Responses to a Shock in Monetary Policy*
  - `application4_1.R`: Estimates the models and plots the results, using the same FRED-MD processed data as that used in Section 3.2.
- Application in Section 4.2: *Impulse Responses to a Shock in Government Spending*
  - `processing_R&Z_data.R`: Processes the data obtained from the website of Valerie A. Ramey, relating to the paper Ramey and Zubairy (2018).
  - `application4_2.R`: Estimates the models and plots the results.

If the scripts are run individually, the command `setwd("your/path/here")` should be changed to a directory where the outputs are saved. Many scripts have the variable `load_sim_from_local_folder <- TRUE` defined near the beginning. When `TRUE`, the script will attempt to load data files from the chosen working directory, e.g. `application4_2.R` would attempt to load the file `dc.RData` created by the script `processing_R&Z_data.R`. When `load_sim_from_local_folder` is set to `FALSE`, the script will load the relevant file from inside the package, when we ran the code ourselves. Where necessary, scripts also include comments with more details on how to run them or change relevant settings.

## Runtime and Hardware Requirements

Running all applications and simulations can be time-intensive, and take multiple days to complete depending on the specifications of the machine running them. The specifications of the machine we used are:

- R version: 4.3.1
- Platform: x86_64-w64-mingw32/x64 (64-bit)
- Running under: Windows 10 x64 (build 19044)
- Processor: AMD Ryzen Threadripper PRO 3955WX 16-Cores 3.89 GHz
- Installed RAM: 128 GB (128 GB usable)
- System type: 64-bit operating system, x64-based processor

Runtimes:

- `running_simulation3_1.R`: 21 hours
- `plotting_simulation3_1.R`: 14 seconds
- `processing_FREDMD.R`: 1 second
- `running_simulation3_2.R`: 6 hours
- `plotting_simulation3_2.R`: 20 seconds

- `application4_1.R`: 5 hours
- `processing_R&Z_data.R`: 1 second
- `application4_2.R`: 2 minutes
- `run_all.R` : 31.5 hours

Installation of this package and other related packages requires approx. 730MB, and the files/plots produced by running the scripts take up around 75MB.

While we did not do extensive testing on the hardware requirements to run the code, we expect it to run on most commercially available computers with 8GB RAM or more.

## Data

We make use of external data in our applications and the simulation calibration in Section 3.2. For Sections 3.2 and 4.1, we use the FRED-MD database, which is freely available at https://research.stlouisfed.org/econ/mccracken/fred-databases/. We obtained the data from this site on 9/9/2021, and a copy of this data is included in the package under `inst/extdata/current.csv`. For more details about this dataset, see McCracken and Ng (2016a).

For Section 4.2, we use the data obtained freely from the webpage of Valerie A. Ramey: https://econweb.ucsd.edu/~vramey/research.html#govt, specifically from the link "Data and Programs" under the section "Government Spending Multipliers in Good Times and in Bad: Evidence from U.S. Historical Data". We ran the provided STATA code to define the variables as they do in their paper, then saved them into a new .xls file. A copy of this file is included under `inst/extdata/processed_data.xls`. For more details about this dataset, see Ramey and Zubairy (2018).

## References

Adamek, R., S. Smeekes, and I. Wilms (2023). Local Projection inference in High Dimensions. *The Econometrics Journal*, Forthcoming.

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

Ramey, V. A. and S. Zubairy (2018). Government spending multipliers in good times and in bad: evidence from US historical data. *Journal of Political Economy 126*, 850–901