User Installation Instructions Manual for `dynr`

Michael D. Hunter, Sy-Miin Chow, Lu Ou, Sukruth N. Reddy,
Meng Chen, Lyinying Ji,& Xiaoyue Xiong

October 16, 2022

The `dynr` software is a package for R written in R and C. It has utilities which allow users to create C code for linear and nonlinear dynamic models, including regime-switching models, without ever having to actually write C code. Various functions in R are called by the user to specify the model desired. These functions in turn write code in C based on the user's input. These functions are then compiled and shared with the rest of the `dynr` C code to estimate the model, obtaining free parameter estimates, standard errors, and latent variable estimates.

1 Instructions for Installing and Using `dynr` on a Windows Computer

Because the `dynr` package compiles C code in response to user input, more setup is required for the `dynr` package than for many others. The general requirements are as follows:

1. Rtools must be installed so that C code can be compiled on Windows. See Section 1.1.

2. Additional GSL libraries must be installed so that the C code can use GSL for matrix multiplication routines. See Section 1.2.

3. The environment variable for the system Path must be modified to include R and Rtools. See Section 1.4.

4. An environment variable for GSL called LIB_GSL must be created. See Section 1.3.

Each of these steps is detailed below. We acknowledge that this additional setup can be bothersome, but we believe the ease of use for the rest of the package and the wide variety of models it is possible to fit with it will compensate for this initial burden. Hopefully you will agree!

1.1 Instructions for Installation of R-tools

1. If you already have R-tools on your machine, make sure the version of R-tools matches with your R version. R-tools versions newer than 3.0 should work fine with `dynr`.

2. Install R-tools through https://cran.r-project.org/bin/windows/Rtools/. Install the latest version of Rtools.

3. The following instructions are based on rtools40.

4. Click RTools 4.0, then click "rtools40-x86_64.exe" in the new webpage. You may be asked if you want to save or run a file "rrtools40-x86_64.exe". Choose "Save" and save the file on the Desktop. Then double-click on the icon for the file to run it.

5. (If you are asked what language to install it in during installation - choose English.)

6. The next page says "Select Destination Location" at the top. By default, it will suggest to install R in "C:\rtools40" on your computer. You may also install R-tools in *any other directory where there are* **no spaces** *in the words describing the directory*. Click "Next" at the bottom of the R tools Setup wizard window.

7. The next page says "Select additional tasks" at the top. Check the box to edit the system PATH. Click "Next" again.

8. The next page says "Ready to install" at the top. Click "install".

9. Rtools should now be installed. This will take about a minute. Click "Finish".

1.2 Instructions for Installation of GSL

1. If you already have GSL libraries installed on your machine, you may skip this step and go to Section 1.3.

2. To install GSL libraries, go to http://www.stats.ox.ac.uk/pub/Rtools/libs.html.

3. Download "local323.zip" (or the latest version) by clicking it. The 323 refers to R 3.2.3.

4. Extract it into a new folder and copy and paste the extracted new folder into the same directory as R ("C:\") so that there would be two folders in that directory. One containing R-3.2.3 (or whatever version of R you have installed) and another local323.

1.3   Setting up the GSL Environment Variable

1. Open - Control Panel\System and Security\System

2. Click on Advanced system settings and then click on "Environment Variables"

3. Add a new *system* variable by clicking on New. Note that this should be a *System* environment variable, not a *User* environment variable.

4. Name the new variable as LIB_GSL and set the variable value to the directory containing the local323 files downloaded in Section 1.2 or the directory where your GSL libraries are installed. Example C:/R/local323 but please note that the direction of these slashes is important. This slash / will work but *not* this one \.

1.4   Setting up R and Rtools Environment Variable

1. Check if R can be run through CMD as follows. Put a simple R script (e.g., save a simple R command: print ("hello") to a .r document as "Hello.r") to a known directory. Open the command prompt window. If you can't find it just do a search from the Start Menu for "cmd" and open "Command Prompt.exe". Then change directory to the location containing "Hello.r" (e.g., cd C:\myfiles). Run the script by typing "Rscript Hello.r". If the file runs correctly, it should print out "hello" in the command prompt console.

2. Close your command prompt window.

3. If the file did not run correctly, follow the next 4 steps.

4. Open - Control Panel\System and Security\System

5. Click on Advanced system settings

6. Check to make sure that your path variable specified during Rtools installation is specified correctly. In Windows 10 do so by clicking on "Environment Variables" and then in the "System variables" panel click on "Path" and then "Edit". Check that the following paths have been specified at the front of your *system* path variable or just move them all to way up to the top in the following order.
C:\R\R-4.1.1\bin\x64
`C:\rtools40\usr\bin`
`C:\rtools\mingw64\bin`

7. Repeat (1) and see if you can now run R from CMD. Make sure that you open a new CMD window after you have completed the steps above. Try typing in "PATH" in CMD to make sure that the newly added paths are indeed shown in the output.

1.5 Wrapping up the Installation Procedure for Windows

1. Open RGui, Rstudio or whatever editor you use to run R. Please type the following code to check whether the gsl commands can be found correctly:
`shell("echo %LIB_GSL%")`
If this returns something like "C:/R/local323", then everything worked fine.

2. If that command returns something like "%LIB_GSL%", then something might be wrong with the GSL installation (Return to Section 1.2) or GSL path (Return to Section 1.3).

3. If the command worked fine, follow the steps in section 3 to finish the installation process in R.

2 Instructions for Installing and Using `dynr` On Mac

2.1 Instructions for Installing Prerequisites on Mac

1. Install Xcode with the command line tools from your Apple store. For this and all the subsequent steps, please read the installation instructions on the individual websites carefully to pick the correct software versions for your operating system.

   - **NOTE:** If you have Big Sur as your Operating System, you do not need to install the command line tools. In fact, you need to remove it by navigating to /Library/Developer and remove Command Line Tools.
   - Then, edit /Library/Frameworks/R.framework/Resources/etc/Makeconf and remove all compilers' explicit link to a particular OS; e.g., change "CC=clang -mmacosx-version-min=10.16" to just "CC=clang." Please see the FAQ section on the dynr Github webpage for further updates.

2. Install Macports by going to https://www.macports.org/install.php

3. Open the terminal window. In the terminal window install the gsl library by typing: "sudo port install gsl".

4. Follow the steps in section 2.2 to verify that these steps worked properly.

2.2 Checking the Installation for Mac

1. Open RGui, Rstudio or whatever editor you use to run R. Please type the following code to check whether the gsl commands can be found correctly:
`system("gsl-config -cflags", intern=TRUE)`
When the command can not be found, and you know where it is stored

(e.g., "/opt/local/bin"), we could then set the PATH variable by typing:
`Sys.setenv(PATH=paste0(Sys.getenv("PATH"), ":", "/opt/local/bin"))`
and then check again.

2. If the above failed, then something went wrong with one or several of the steps in Section 2.1. Please go back and try repeating or checking that section.

3. Follow the steps in section 3 to finish the installation process in R.

3   Getting dynr from CRAN

1. Open RGui, Rstudio, or whatever editor you use to run R.

2. Install `dynr` from CRAN as usual by typing install.packages('`dynr`') and then require('`dynr`').

3. To test if `dynr` is installed correctly, run one of the demo examples. For instance, type:
`demo('LinearSDE',package='dynr')`
`Press <Return> to start the demo.`
Note that this specific demo requires the packages *Matrix* and *mvtnorm* so make sure that these have been installed.

4. If you're on Windows and everything worked fine until you tried to run the model, something probably went wrong with installing Rtools (Section 1.1) or installing GSL (Section 1.2). Please refer to those sections for troubleshooting.