

CPC Zürich 2025 Practical Tutorial J

Regression DCM using TAPAS

Installation Guide

Authors/Tutors: Imre Kertesz (ikertesz@biomed.ee.ethz.ch), Herman Galiouline (galiouline@biomed.ee.ethz.ch), Computational Psychiatry Course 2025, Zürich, Switzerland.
Revision and testing: Florian Schönleitner (schoenleitner@biomed.ee.ethz.ch)

This document guides you through the installation of necessary tools and data needed for the tutorial. In case additional files are needed, we will inform you in time, so that you can also download them before the tutorial.

1 Julia

In this tutorial we will work with the [rDCM toolbox](#) which is written in the [Julia](#) programming language. The following steps describe the installation of Julia and necessary Julia packages.

1. Download and install Julia for your operating system as described [here](#). We recommend Julia version 1.9 or higher.
2. Download the file `cpc2025_rdc_m_tutorial.zip` from [here](#) (password: `cpc2025`) and unzip it. Place the folder in a location you will use later for the tutorial (e.g., `Documents/CPC/cpc2025_rdc_m_tutorial`).

The folder `cpc2025_rdc_m_tutorial` contains a file called `Project.toml`. This file is needed to set up the environment for the tutorial (similar to `requirements.txt` in Python). The following steps show you how to instantiate the environment using the `Project.toml` file:

3. Open the Julia REPL.

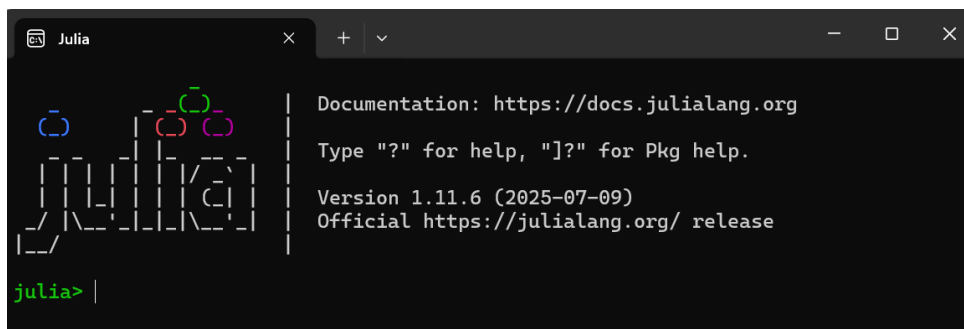
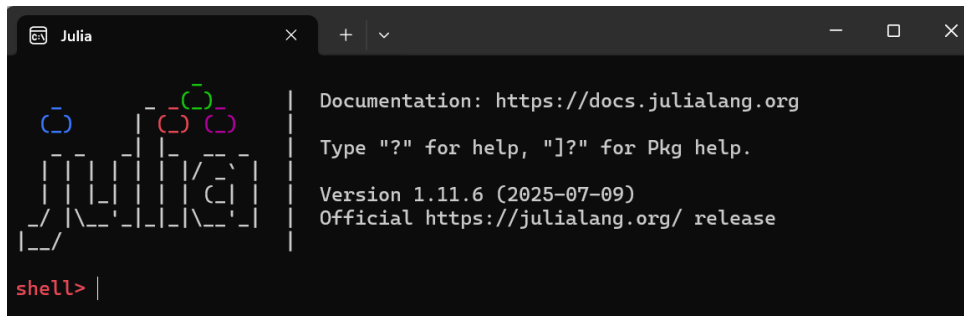


Figure 1: Julia REPL (in Julian mode)

4. Type `;` to switch to shell mode. The prompt will change to `shell>`.

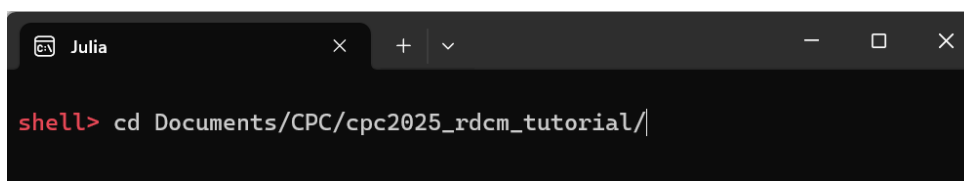


```
Documentation: https://docs.julialang.org
Type "?" for help, "]? " for Pkg help.
Version 1.11.6 (2025-07-09)
Official https://julialang.org/ release

shell> |
```

Figure 2: Julia REPL (in shell mode)

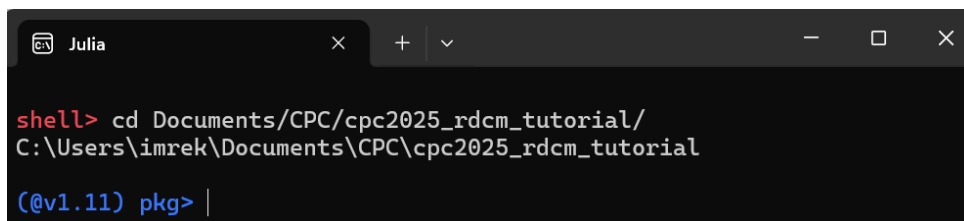
5. Change directory to `cpc2025_rdc_m_tutorial` using the `cd` command (e.g. `cd Documents/CPC/cpc2025_rdc_m_tutorial/`).



```
shell> cd Documents/CPC/cpc2025_rdc_m_tutorial/|
```

Figure 3: cd into tutorial folder

6. Press backspace to get back to Julian mode and then type `]` (closing square bracket) to switch to the package manager mode. You should see the following:

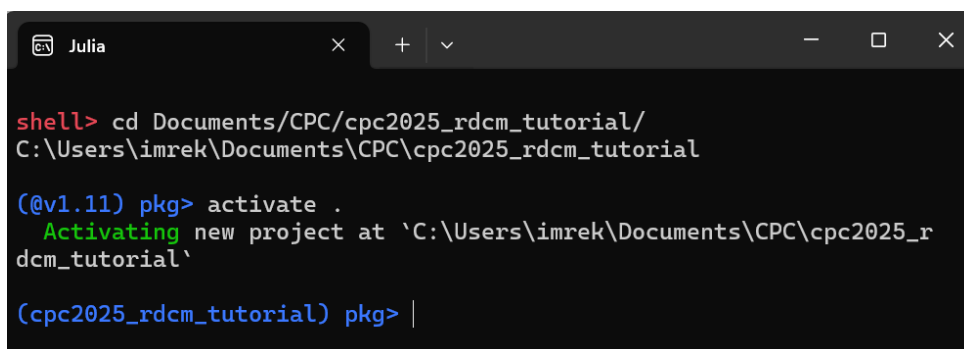


```
cd Documents/CPC/cpc2025_rdc_m_tutorial/
C:\Users\imrek\Documents\CPC\cpc2025_rdc_m_tutorial

(@v1.11) pkg> |
```

Figure 4: Julia REPL (package manager mode)

7. Type `activate .` This will change the environment name to `cpc2025_rdc_m_tutorial`.



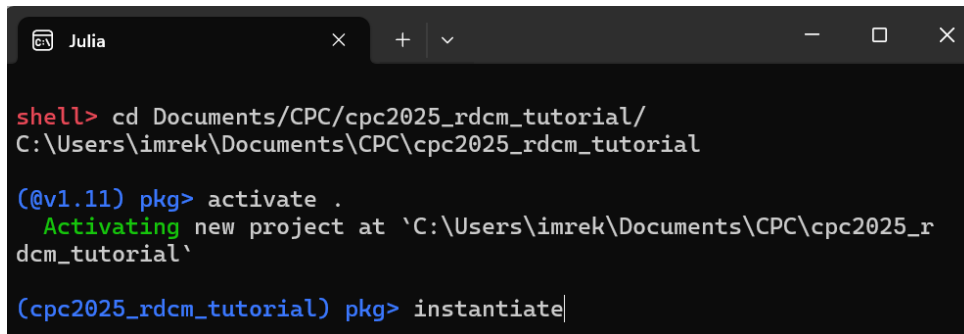
```
cd Documents/CPC/cpc2025_rdc_m_tutorial/
C:\Users\imrek\Documents\CPC\cpc2025_rdc_m_tutorial

(@v1.11) pkg> activate .
Activating new project at `C:\Users\imrek\Documents\CPC\cpc2025_rdc_m_tutorial`

(cpc2025_rdc_m_tutorial) pkg> |
```

Figure 5: Activating the environment.

8. In order to install the dependencies, type `instantiate` (the installation of the packages might take a while).

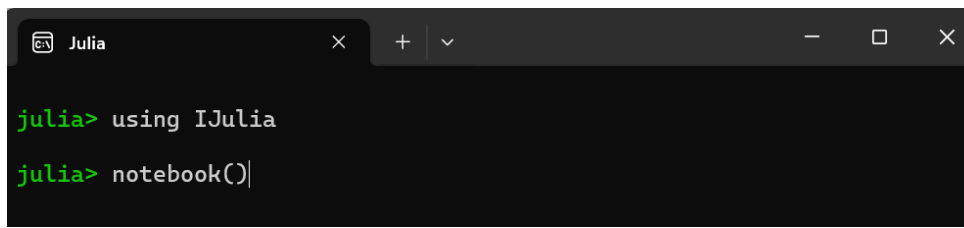
A terminal window titled 'Julia' with standard window controls. The prompt is 'shell>'. The user enters 'cd Documents/CPC/cpc2025_rdc_m_tutorial/' and the shell shows the full path 'C:\Users\imrek\Documents\CPC\cpc2025_rdc_m_tutorial'. The prompt changes to '(@v1.11) pkg>'. The user enters 'activate .' and the terminal shows 'Activating new project at `C:\Users\imrek\Documents\CPC\cpc2025_rdc_m_tutorial`'. The prompt changes to '(cpc2025_rdc_m_tutorial) pkg>'. The user enters 'instantiate|' and the terminal shows a cursor at the end of the command.

```
shell> cd Documents/CPC/cpc2025_rdc_m_tutorial/  
C:\Users\imrek\Documents\CPC\cpc2025_rdc_m_tutorial  
  
(@v1.11) pkg> activate .  
Activating new project at `C:\Users\imrek\Documents\CPC\cpc2025_rdc_m_tutorial`  
  
(cpc2025_rdc_m_tutorial) pkg> instantiate|
```

Figure 6: Instantiate the environment.

This concludes the setup of the Julia environment. You can test if the setup was successful by typing the following commands into the terminal (switch back from package manager mode to Julian mode by pressing back space):

- `using IJulia`
- `notebook()`

A terminal window titled 'Julia' with standard window controls. The prompt is 'julia>'. The user enters 'using IJulia' and the terminal shows the command being executed. The prompt changes to 'julia>'. The user enters 'notebook(|' and the terminal shows a cursor at the end of the command.

```
julia> using IJulia  
  
julia> notebook(|
```

Figure 7: Test Julia environment by launching Jupyter.

This should launch Jupyter in a browser which will be needed for the tutorial.

2 SPM12 (Optional)

The installation of Matlab and SPM12 is optional. In order to apply the **rDCM** toolbox to empirical data, one needs to extract BOLD signal time-series from fMRI data. The tutorial briefly covers how this can be done with the **Statistical Parametric Mapping** (SPM) software package. We will provide the extracted BOLD signal time-series, but if you want to extract them yourself, please install Matlab and SPM12 as described below:

1. **Download MATLAB** at: (<https://www.mathworks.com/products/get-matlab.html>). We recommend using MATLAB R2016a or newer. If you don't have access to a valid license, please have a look at the [CPC website](#) for more information.
2. **Download SPM12** at: <https://www.fil.ion.ucl.ac.uk/spm/software/spm12/>
3. Put the code and the material in a folder which you will use for the practical tutorial (e.g., Documents/CPC/cpc2025_rdcmtutorial). **Make sure you do not have any spaces in the titles of your folders!**
4. Add the `spm12` folder to your MATLAB path. For this, in MATLAB, navigate to the folder you prepared (e.g., `cpc2025_rdcmtutorial`). Then right-click on the folder and select *Add to Path* → *Selected Folders*. **Important: Do not add the subfolders for SPM12.**

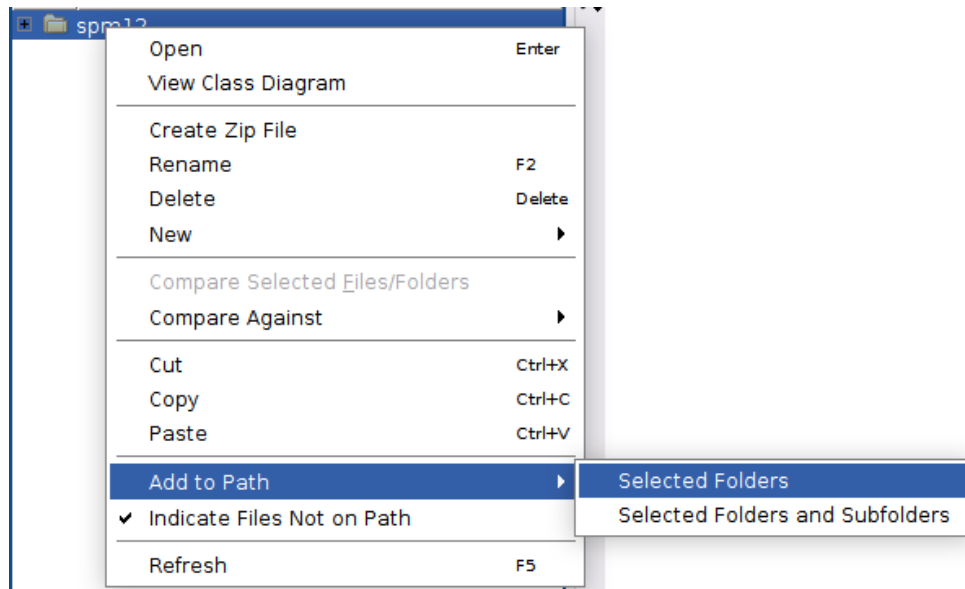


Figure 8: Add SPM12 to path.

5. Type `"spm"` into the command window of MATLAB and press Enter. If the installation was successful, this will open the SPM interface:

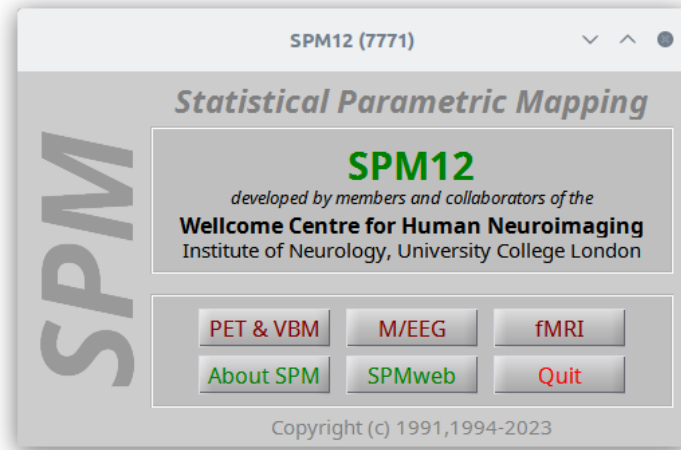


Figure 9: Illustration of the SPM interface.

If you have trouble getting to this point before the Practical Tutorial Session, please consult the **#tutorial-helpdesk** channel on Discord. You will be given access to the CPC Discord workspace at the beginning of the course. Check if anyone has had the same issue and has managed to solve it. If no one else has encountered the same problem, post your question. The team will be monitoring the channel and providing support. In addition, given the volume of attendees this year, we would be grateful if you could assist us by answering queries on Discord yourself if you come across a problem you know and have solved.

For those who need more personalized help, we will be offering support hours. More information on exact time and location will follow.

We look forward to seeing you all at the CPC Zürich 2025!

If you have the following issues with MEX files on macOS Catalina, Big Sur, Monterey, Ventura:

```
"*.mexmaci64" cannot be opened because the developer cannot be verified. macOS
cannot verify that this app is free from malware
```

or

```
Code signature not valid for use in process using Library Validation: library
load disallowed by system policy,
```

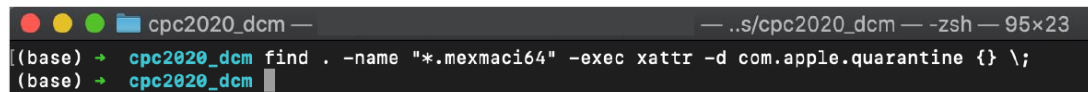
open a Terminal and navigate to the folder where you placed your **spm12** folder. In the following example, the **spm12** folder was placed in a folder named **cpc2020_dcm** in the **Downloads** folder.



```
cpc2020_dcm — ..s/cpc2020_dcm — zsh — 95x23
(base) → ~ cd Downloads/cpc2020_dcm
```

Then type the following command:

```
find . -name "*.mexmaci64" -exec xattr -d com.apple.quarantine {} \;
```



```
cpc2020_dcm — ..s/cpc2020_dcm — zsh — 95x23
(base) → cpc2020_dcm find . -name "*.mexmaci64" -exec xattr -d com.apple.quarantine {} \;
(base) → cpc2020_dcm
```

This should solve the problem and allow you to run the Matlab scripts. This solution was taken from the SPM [Wiki](#).