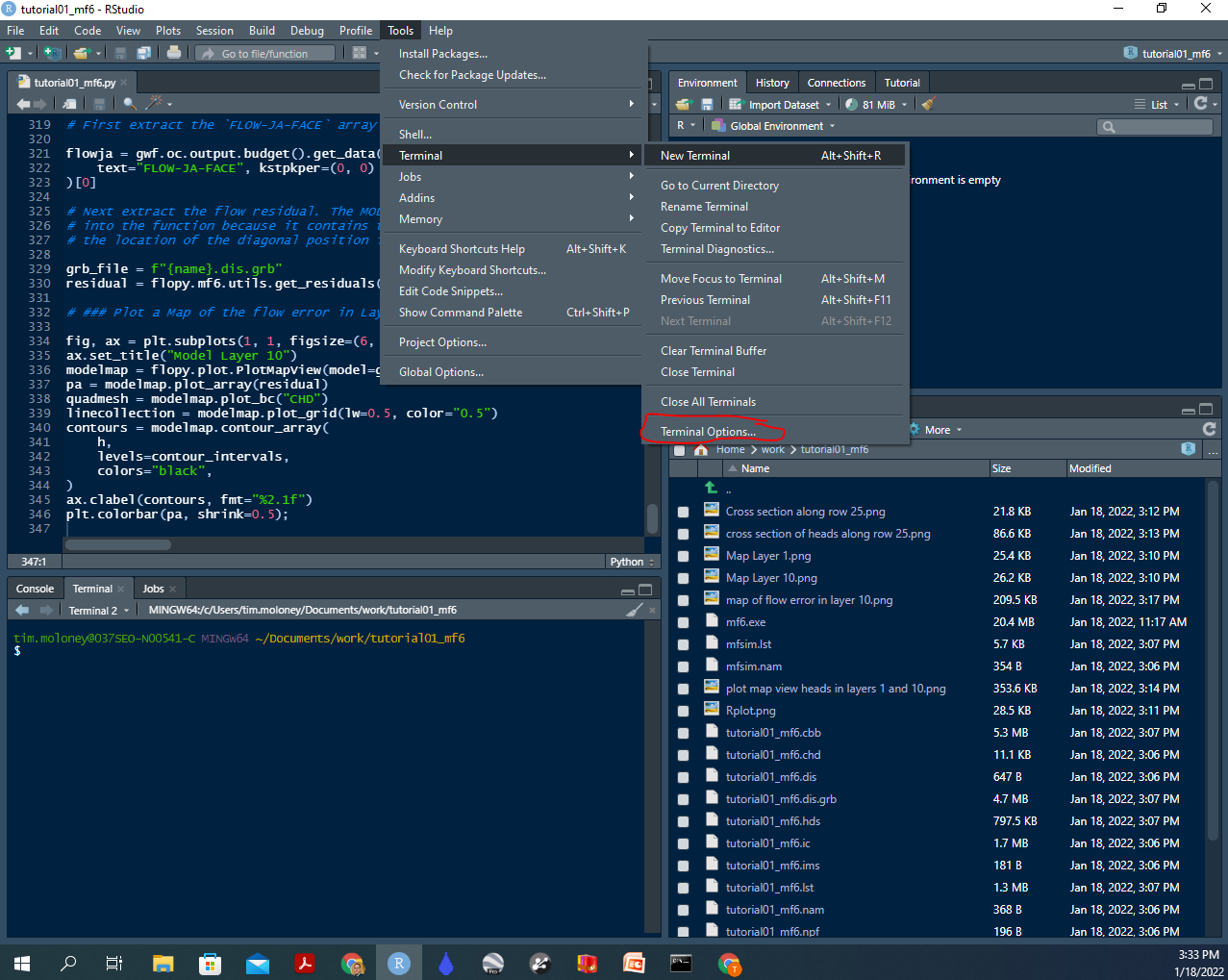
Python GW modeling

From R, library(reticulate)

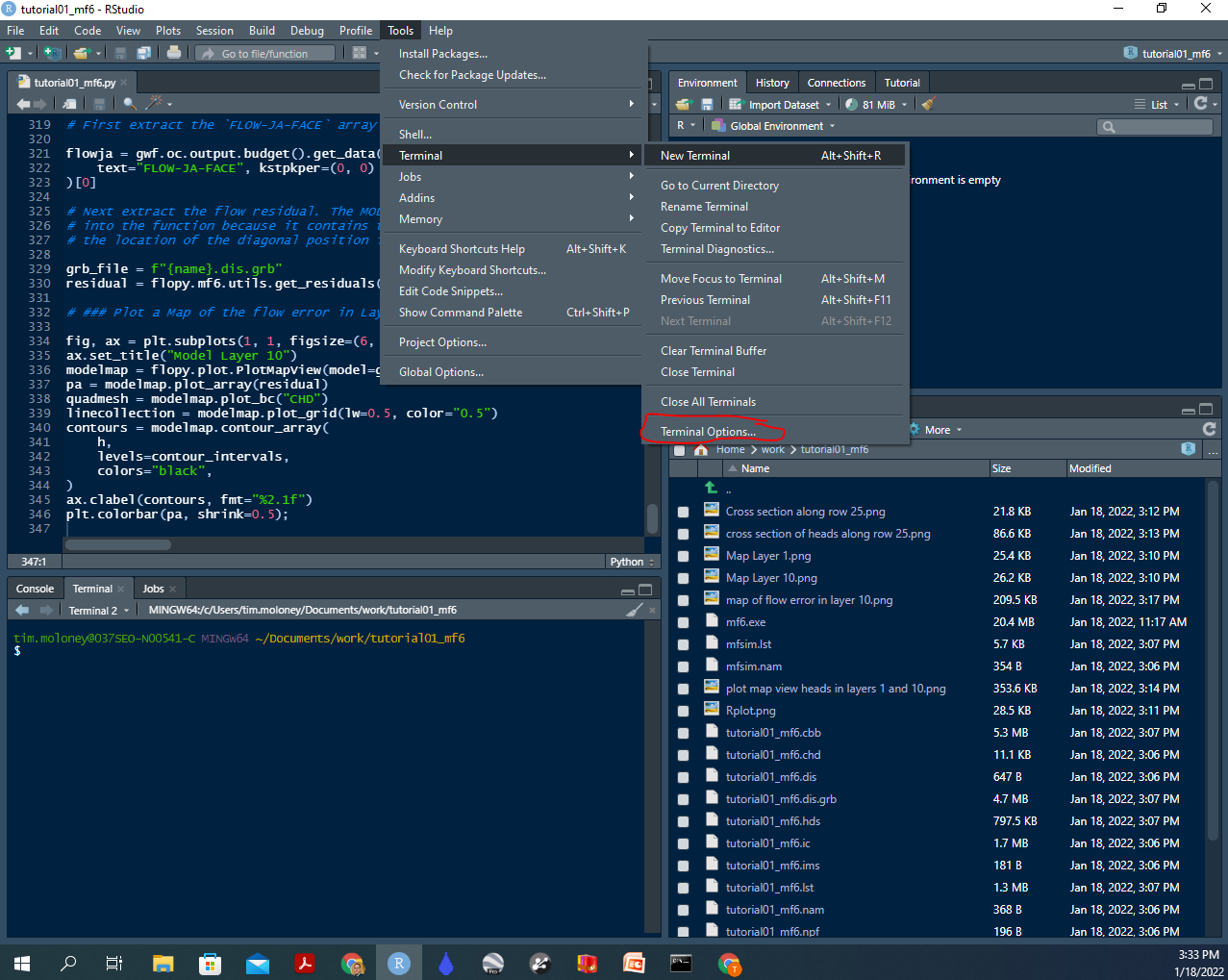
Go file new Python script and enter some python code to start the python interpreter. You can type exit to return to R.

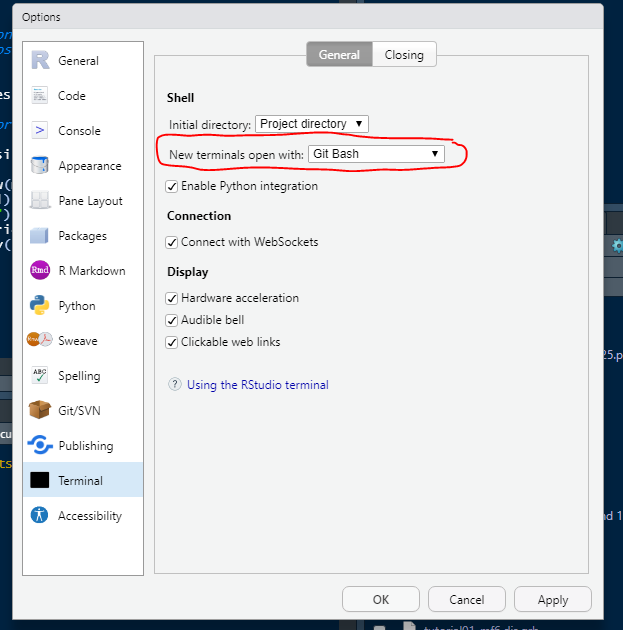
Import all relevant python modules, e.g., numpy, pandas, matplotlib, and perhaps most importantly, flopy.

If you need something else installed use “pip install …” from a git bash shell in the terminal window to install what you need



If you are not seeing the git bash shell check under Terminal options



Make sure that new Terminals open with Git Bash, and then open a new Terminal if necessary.

You can use “pip list” in a git bash shell to view a list of all the python modules and versions in your environment.

Once all relevant modules are installed and imported to the python session, Flopy should work fine to create your model input files in a directory of your choice. Make sure a copy of the model executable you are trying to run (e.g., mf6.exe) is in the same working directory with your files.

Note that the run\_simulation command to execute Modflow 6 uses a separate python module called subprocess.Popen to call the mf6 executable. It does not work with Rstudio on windows and throws an Error which I haven’t been able to overcome. OSError: [WinError 6] The handle is invalid. It is probable that this is also the case for other model versions, but that is untested.

You can work around this by running mf6 from the command line, open a windows shell or command prompt (not git bash shell) on the working directory and type mf6.exe. Alternatively just open command prompt and type the whole path to the working directory + mf6.exe, e.g., "C:\Users\tim.moloney\Documents\work\tutorial01\_mf6\mf6.exe" It is probable that you can work around this by calling the exe file from R, but I have not yet figured out how to do this.

Assuming Flopy ran correctly to write the simulation name file (must always be named mfsim.nam) and other files, mf6 should run correctly. The Flopy results importing commands can then be used to import and visualize results of the model run.

You will need to run all the code supplied in tutorial documents to generate these types of plots and then type plt.show() to get the figure to draw in Rstudio.

