# Installation Instructions for PyMC3

PyMC3 is a markov chain monte carlo (mcmc) sampler, which is computationally expensive even for reasonably trivial models.

While PyMC3 can run on python’s standard numpy library, but it isn’t optimised and frankly struggles for even modest problems.

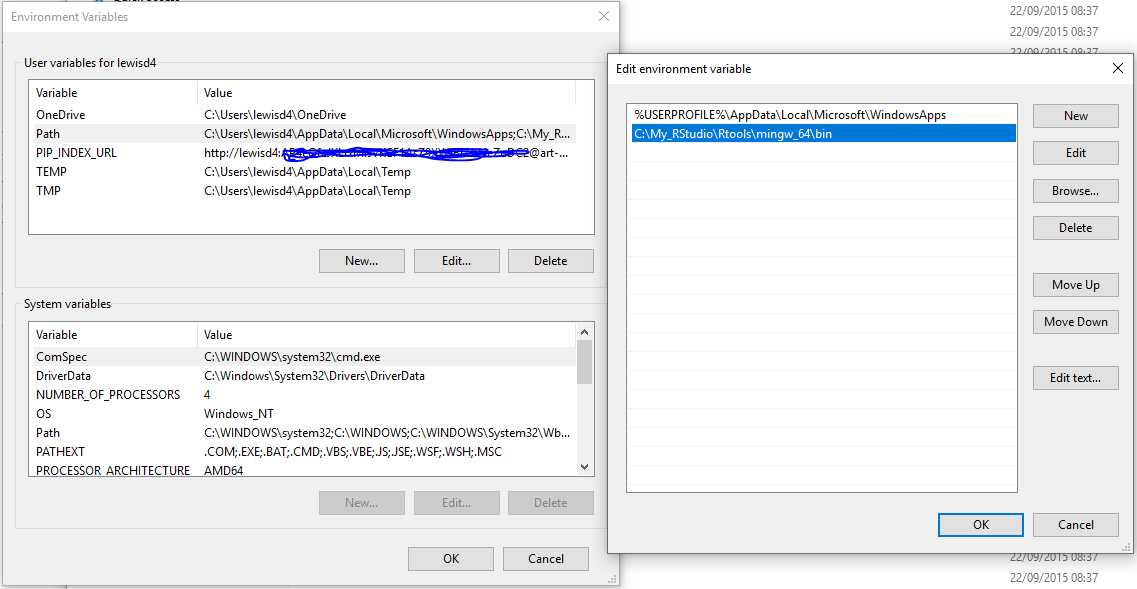
The way that PyMC3 would like to work is to compile models to C code, run that C code and pass the output back to python. This is much more efficient, however it requires that python be able to talk to a C compiler.

If you are lucky, you can set up python to talk to a C compiler, and hence run PyMC3 efficiently.

## Do I have a C compiler?

Recent installs of R within the ONS come with the R toolchain, and hence mingw tools for compiling C and fortran code. Check you’ve got this by exploring your file structure:

* Find your R install (or request R/update R if you don’t currently have it). Mine is located at: c:\My\_RStudio
* See whether you have the ‘Rtools’ toolchain, mine is at c:\My\_RStudio\Rtools
* We now want to locate the various command line tools for min\_gw 64bit, mine are located at: c:\My\_RStudio\Rtools\mingw\_64\bin
  + NB – what I’m looking for are things like gcc.exe and gfortran.exe
* Having found where your 64-bit mingw tools live, add this folder to your PATH environment variable.



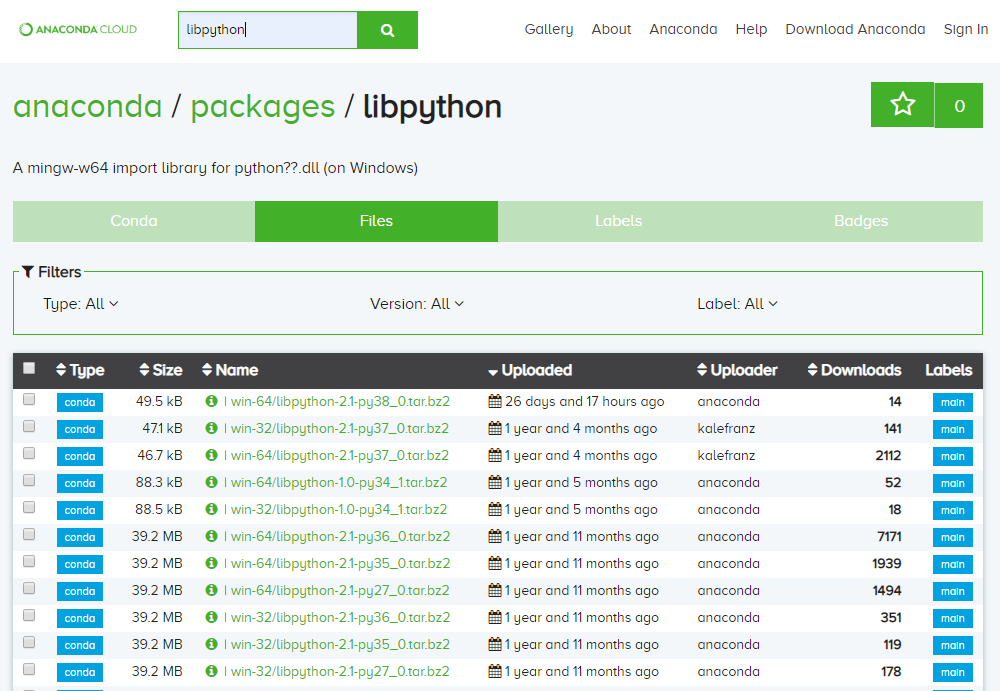
Above you can see I’ve accessed my environment variables for my account, and have added the mingw tools to the Path variable.

This establishes compiler functionality to your command line, now we need to make sure that python can make use of it.

## Making Python Talk to the Compiler

Python needs a couple of additional files to help it talk to the compiler. These files are commonly provided by a package called ‘Libpython’. As ONS uses the anaconda distribution for python, the best place to get Libpython for our python install is anaconda.

* Go to the package repository for anaconda: <https://anaconda.org/anaconda/repo>
* Search for libpython in the search bar
* Click the anaconda/libpython 2.1 link, go to the ‘Files’ tab.
* Download the relevant tar.bz2 file for your system – likely win-64 and py36, so something like:
  + win-64/libpython-2.1-py36\_0.tar.bz2





* Now, open “anaconda prompt” and cd to your downloads folder (or wherever you saved the libpython file).
* Use conda to install libpython, should look something like:
  + conda install --offline libpython-2.1-py36\_0.tar.bz2
  + if you start typing libpython and press tab cmd should autocomplete the filename.
  + offline is a flag and has two dashes in front of it.
* Conda will quietly install libpython and return a prompt when done.

Hopefully python should now be set up to use mingw compiler tools. You can check this by opening python and importing pymc3. If you don’t get the long warning about there not being a compiler, then you should be good to go!