**Getting Started with the FNNR-ABM Project**

Glossary

ABM – Agent-based Model

IDE – Integrated Development Environment

OS – Operating System

In order to access and download this project (for Python beginners):

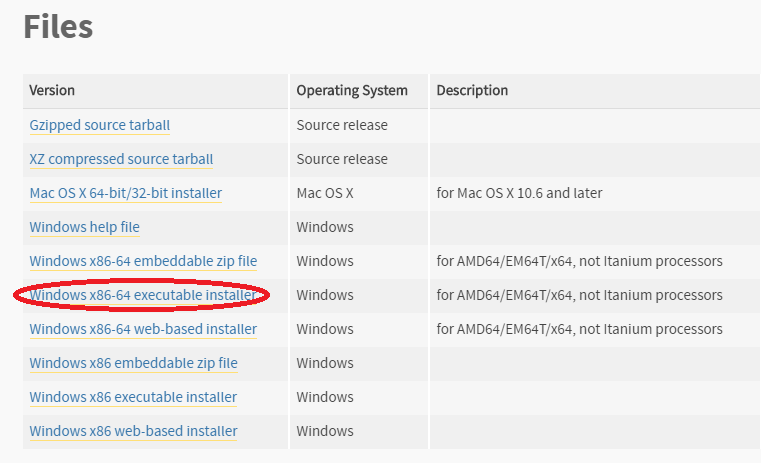
1. Have Python 3+ installed on your computer.

To download the latest version of Python, visit <https://www.python.org/>. At time of writing, [Python 3.6.1](https://www.python.org/downloads/release/python-361/) is the latest verison, though again, any version of Python 3.X.X should work. Python 2.X.X is more stable for use with older systems, but it differs in syntax from Python 3.X.X, so it is not compatible with code from the imported libraries we will use here (such as Mesa).

On the Python download page, scroll to the bottom and select the option that is best for you. For the most common configuration, refer to Figure 1.1; however, it may not apply to you. First, find out if you have a Mac, Linux or Windows OS, then figure out if your OS is 32-bit (x86) or 64-bit (x64). To find this out, view your computer properties (on Windows 10, search or find ‘This PC’ in File Explorer, right-click, and select ‘Properties’ from the menu; other versions of Windows might need you to right-click ‘My Computer’). Most standard newer computers will have the 64-bit version of Windows.

Once you download and run the installer (or configure the zip file/tarball; the installer is recommended), follow the installation steps to install Python 3.X.X onto your computer. If you are not sure what options to pick, do not change the default options. Keep note of where Python is installed on your computer. If it is convenient and fast to do so, restart your computer afterwards.

Figure 1.1 – The most common option. This option may not be right for you if you are not using a 64-bit version of Windows.



2. (optional) Download a Python IDE.

There are many different software programs that will run your Python code. IDEs are optional to download because Python comes with a default one named IDLE (and for shorter python functions, one can even run code straight from the command line). However, downloading a more sophisticated IDE will handle different versions of Python and different libraries more seamlessly, as well as provide debugging/testing tools and more detailed error messages. They may also provide other tools such as a built-in file system to manage multiple Python modules (files) more easily, the ability to open non-Python files, and more.

Once you have found an IDE (google to find different ones available; the one used in this tutorial is PyCharm), follow installation instructions, unzipping/extracting any files with 7zip (a free program) or Winzip as needed.

3. Download the Python libraries needed for the project (Mesa, openpyxl).

Python has many built-in frameworks and libraries (collections of pre-written functions and modules) that save users time and effort, as well as many more libraries available on the web to download; most common projects will use at least one external library (as opposed to being coded entirely from scratch). The two libraries we must download for the project are:  
  
Mesa – a Python 3+ framework for working with agent-based models

If you install Mesa through pip (covered later here), it will come installed along with the other libraries it depends on, such as Tornado (web framework), Pandas (data structure library), Numpy (for a variety of numerical expressions or generations), Six (for wrapping over differences between Python 2 and 3), Tqdm (progress meter), Matplotlib (for plotting, and more. The user will likely not directly access these libraries when working with Mesa, but they should be aware of what the libraries do.

Openpyxl – helps import data from cells in Microsoft Excel or Open/LibreOffice Calc files

The most common (and Pythonic) way to install external libraries is to open the Command Prompt on Windows (cmd.exe), or a similar terminal on whatever OS you’re using, and type in:

pip install mesa

and when that’s done,

pip install openpyxl

If you are using conda or miniconda (or another environment/package manager), replace ‘pip’ with ‘conda’ in the above commands.

**Troubleshooting**

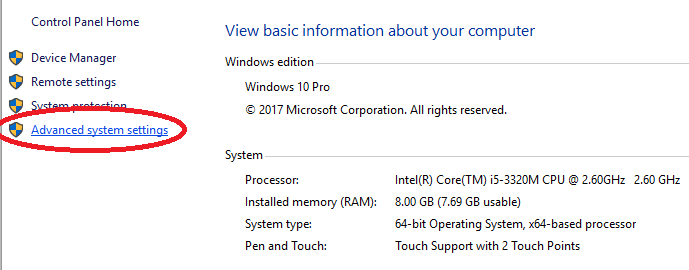
There are a number of possible error messages you can get. The instructions below diagnose them based on Windows 10.

*If ‘pip’ is not recognized in the command window:*

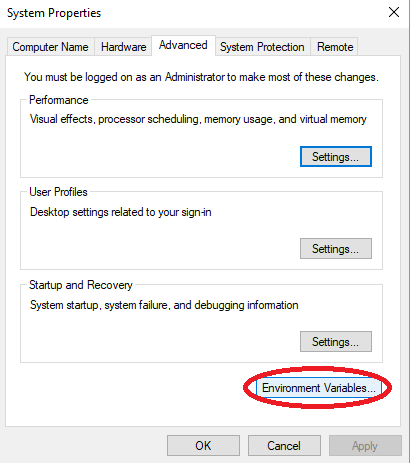
1. Set the Environment Path.

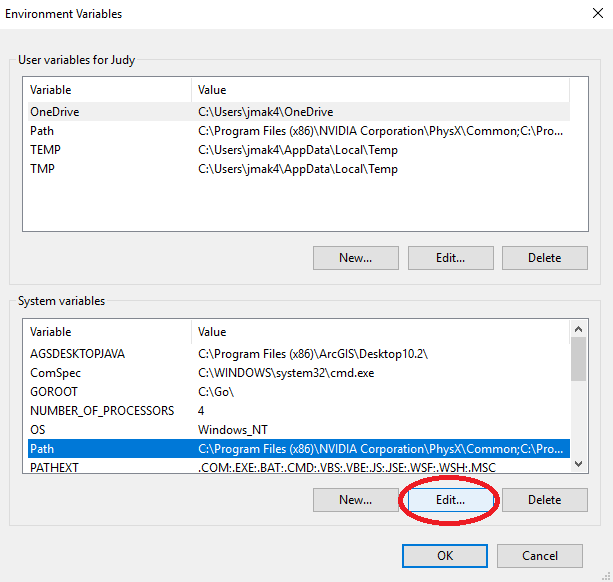
a. This PC or My Computer > Right-Click > Properties

b. Click on ‘Advanced system settings’ on the left tab.



c. Select ‘Environment Variables...’.

d. Select ‘Path’ under ‘System Variables’ (near bottom of the window, not the first ‘Path’ near the top), then ‘Edit...’.

e. Select ‘New’, then ‘Browse...’ to find where your Python installation is. Common filepaths to add here include (depending on where you’ve installed Python):

C:\Python27

C:\Users\<YOUR USERNAME>

C:\Users\<YOUR USERNAME>\Downloads

C:\Users\<YOUR USERNAME>\AppData\Local\Programs\Python\Python36

Basically, add the directory that contains the same version of python.exe that you want to run. Make sure that when adding new filepaths, you do not overwrite old ones.

\*NOTE: If you have multiple versions of Python installed, make sure that the Python version you want is moved up above the other version(s). To do this, select the ‘Move Up’ button in the Path > Edit… window. Now you should be able to run pip in the command line to install the necessary libraries.

2. Change the CMD Directory.

For example, if you have Python installed under C:\Users\<YOUR USERNAME>\AppData\Local\Programs\Python\Python36, then in cmd.exe, you may want to type:

cd C:\Users\

in order for cmd.exe to look for pip in the right drive.

*If it installs successfully in the wrong directory, or if your IDE does not recognize the library after installation:*

3. If you are using Anaconda/Miniconda and the library has installed the library in the wrong environment (such as one for a version of Python 2.X.X), set up a new environment; otherwise, skip to Step 4.

To set up a new environment, type the following into the Command window:

conda create --name 3point6 python 3.6

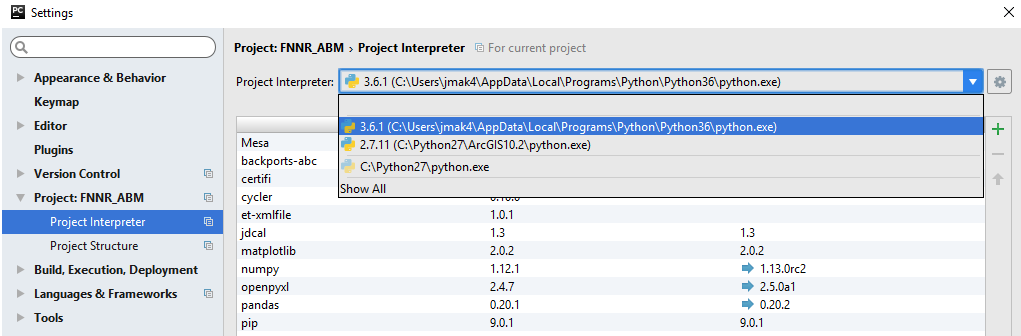
Note: ‘3point6’ here can be any name you wish, and ‘3.6’ can be changed to another version of Python.

Then activate the env in the Command window:

activate 3point6 (or whatever you named it)

You should be able to use the pip command to install the needed libraries under this new environment now. Proceed to Step 4.

4. Change your IDE/project interpreter.



This varies per IDE, but in PyCharm, you will want to go to File > Settings to set the Project Interpreter. The Project Interpreter should either be wherever you have the desired version of Python installed, or in a Conda environment that has the desired version of Python installed (see Step 3). You will know you’ve selected the correct environment when:

- the Python version shown is 3.X.X, and

- the libraries shown in the table under the Interpreter selection dropdown box include pip, matplotlib, openpyxl, pandas, Mesa, and more (assuming that you have successfully used pip or conda in the command line to install the libraries at this point).

Finally, if your libraries appear to have successfully installed to the same directory that runs the desired version of Python as configured in your IDE, but you still get import errors, you may need to restart your computer.

4. Download the FNNR-ABM project files from Github.