# Background and Scope

Manitoba Agriculture Weather Program operates and maintains a network of 108 weather stations that provides weather variables such as air temperature, wind speed and direction, humidity, precipitation, solar radiation and soil temperature and moisture at 5, 20, 50 and 100 cm depths.

The program supports many partners and the data has a wide variety of applications such as generating current condition maps, crop thermos-physiological growth stage (e.g. Growing Degree Days, Corn Heat Units etc.) and crop disease risk mapping (e.g. Fusarium Head Blight). The data query process occurs in the database that sits in the managed environment. This job aid provides information required to upload data into the managed environment (ME) using the AgAuto python program.

# Software Installation

In order to use the AgAuto program, the necessary software must first be installed. The software that AgAuto needs are Anaconda, and PyCharm. AgAuto also needs the folder containing the AgAuto script and configuration files.

**Installing Anaconda**

Anaconda is an open-source code python software distribution and is how the necessary package(s) will be installed.

1. First navigate to <https://www.anaconda.com/distribution/>
2. As shown in Figure 1, select the windows icon to select Anaconda for windows.

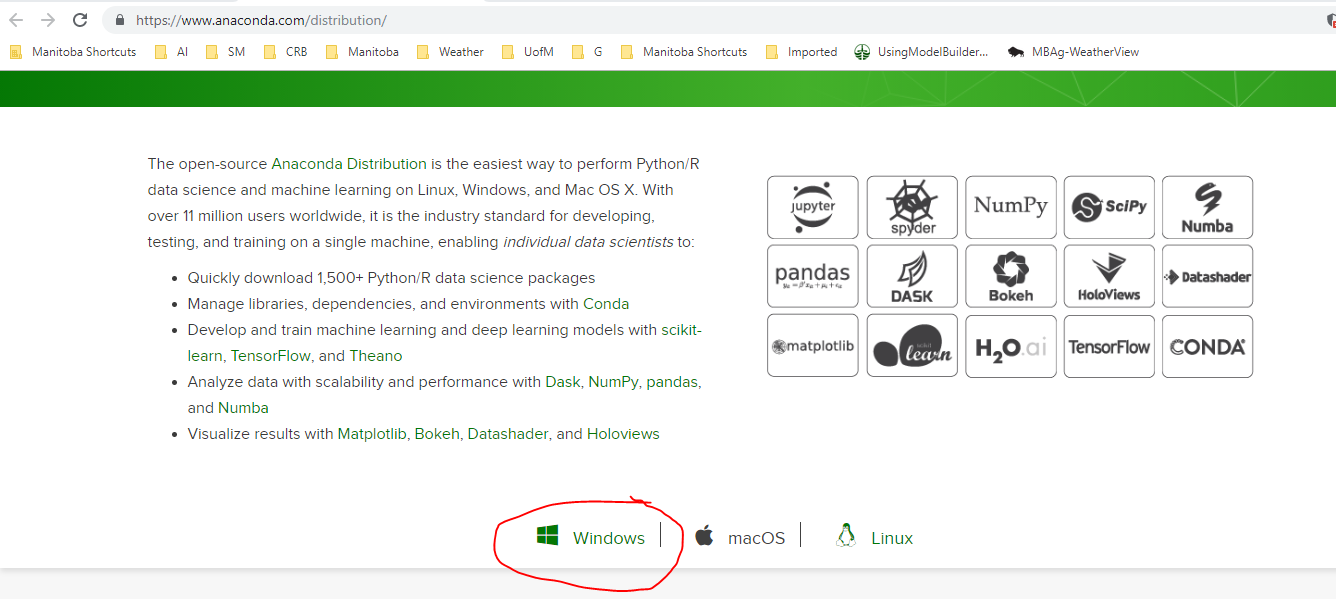


Figure 1: Selecting proper Anaconda version for computer.

1. As shown in Figure 2, select the latest python version (Python 3.7 as of May 2019) for the 64-bit version of windows.

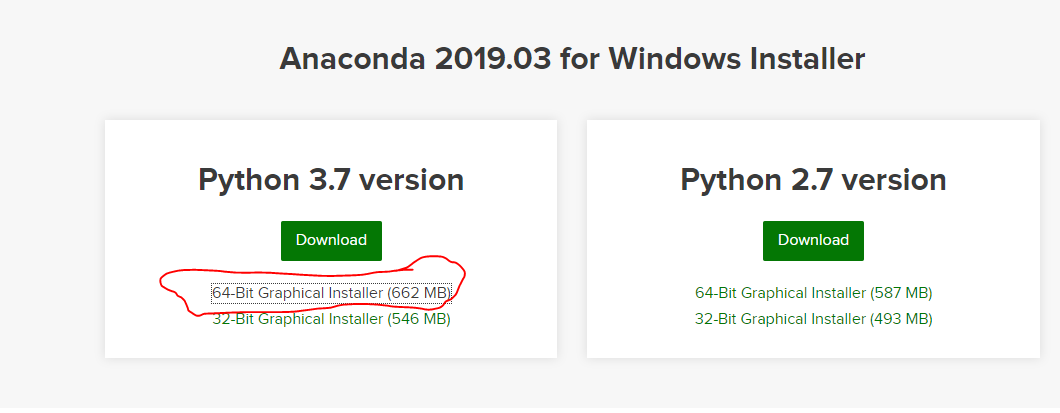


Figure 2: Selecting the latest 64-bit Anaconda version.

1. After the download has completed, go through the installation process. The installation could take up to 30 minutes.
   1. If prompted to select an installation folder, just use the default location.
   2. If prompted, do not check the box ‘add to path variable’.

**Setting Up the Virtual Environment**

Although we have previously installed Anaconda for Python 3.7, the AgAuto program is written on Python 3.6. The newest Anaconda was installed in order to easily install new packages in the future. A future project might be to port the AgAuto script into Python 3.7 but for now, we can create a virtual environment for Python 3.6 to run Python 3.6 code.

1. In Windows search, type ‘Anaconda Prompt’ and run as shown in Figure 3.

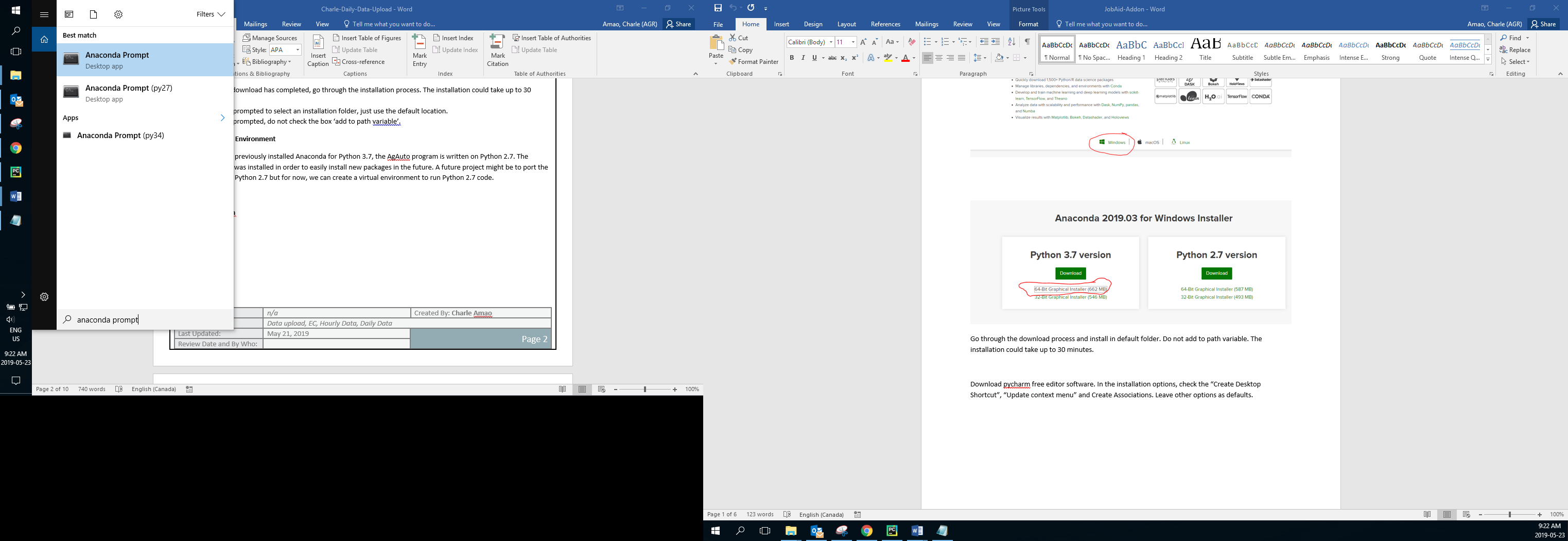


Figure 3: Running the Anaconda prompt.

1. At the prompt, type *‘conda create -n py36 python=3.6 anaconda’* as shown in Figure 4. This process might take 30-40 minutes.
   1. If prompted to install packages (y/n), type ‘y’ to install the necessary packages.

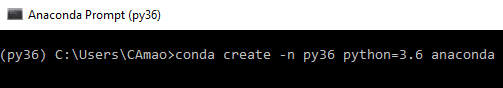


Figure 4: Typing this command creates a virtual Python 3.6 environment named 'py36'. We will use this to run AgAuto in later steps.

1. Once the creation process is complete, type ‘conda activate py36’ and press Enter in order to activate the newly created virtual environment.
2. While inside the prompt, navigate to the AgAuto’s working directory. This is simply inside the folder that contains the AgAuto script and its configuration files. In order to navigate to the directory simply type *cd filepath*, where *filepath* in this case is *C:\Users\CAmao\Documents\AgWeather\AgAuto, filepath* will change depending on where you place the AgAuto directory.

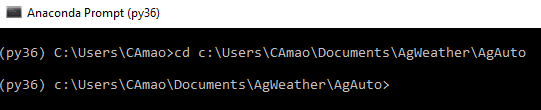


Figure 5: After entering the command, the prompt now sees everything within the AgAuto folder.

1. In the prompt, type *pip install REQUIREMENTS.txt* as shown in Figure 6. This looks through the REQUIREMENTS.txt file within the AgAuto folder to install the necessary packages listed there.

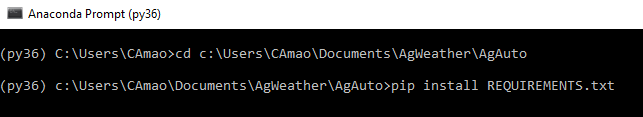
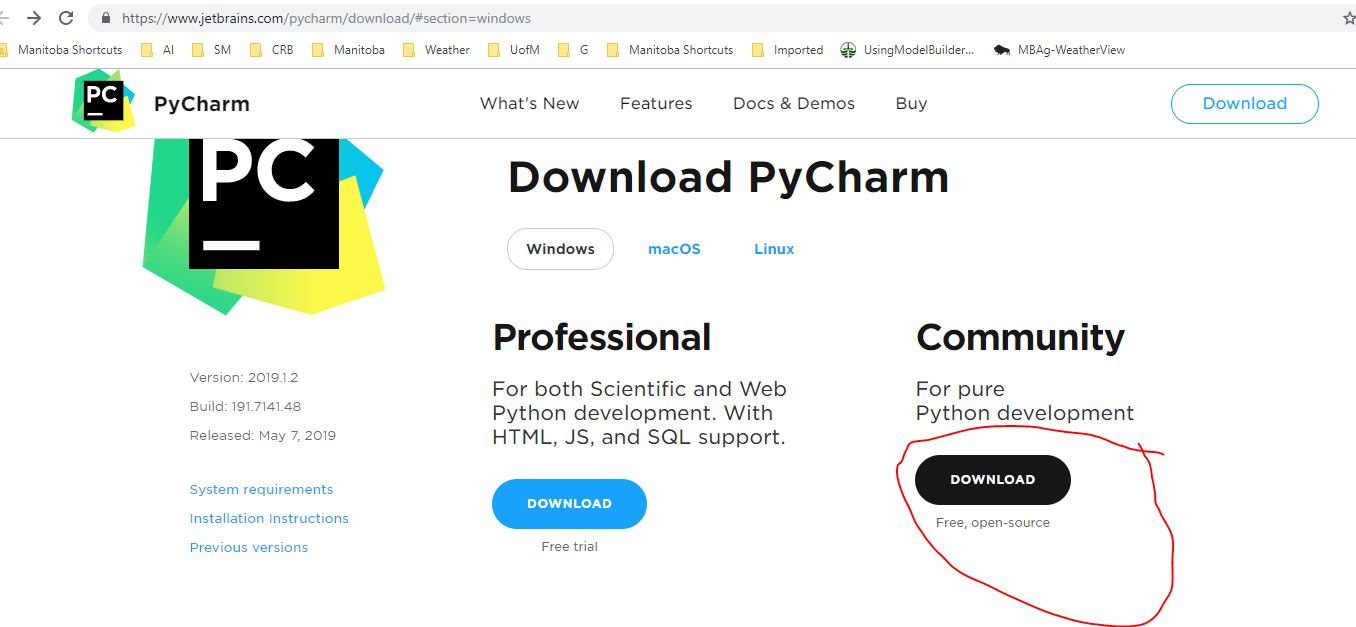


Figure 6: Using pip install command to install necessary packages within REQUIREMENTS.txt. Anaconda is now properly setup for AgAuto.

**Installing Pycharm**

Once Anaconda is properly set-up, the Pycharm IDE must be installed in order to run the program and edit the code should you wish to change it in the future.

1. Navigate to <https://www.jetbrains.com/pycharm/download/#section=windows> and download the Community version for Windows.



1. Once PyCharm has been downloaded, proceed with the installation.
   1. If prompted, check the “Create Desktop Shortcut”, “Update context menu” and “Create Associations for .py”. Leave other options as defaults.
2. Once PyCharm has been successfully installed, navigate to the folder containing *AgAuto.py* and open *AgAuto.py* using PyCharm.
3. Once AgAuto has been created, press ‘Configure Python Interpreter’ in the top right corner.

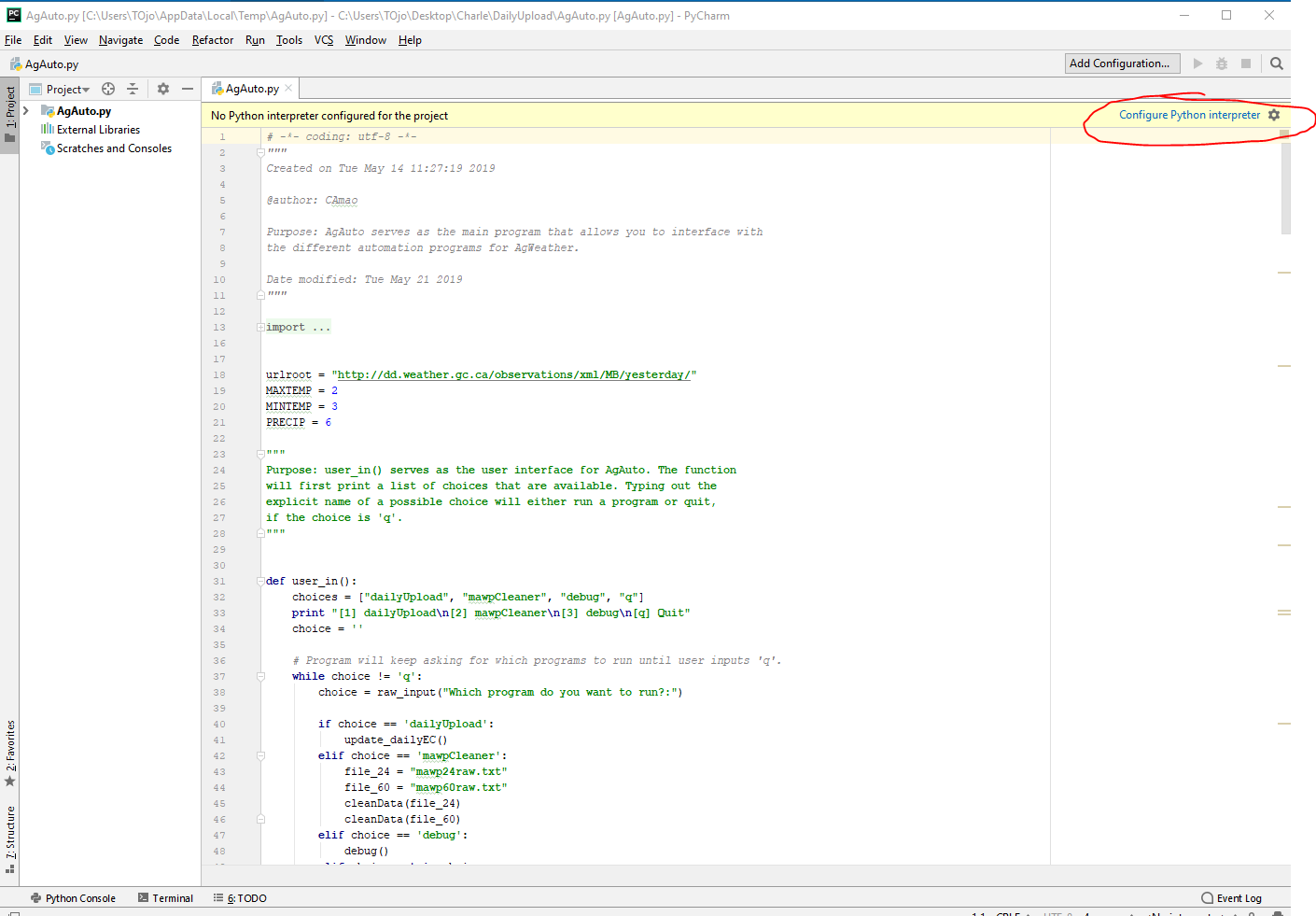


Figure 7: PyCharm needs to know which version of Python to run. In this case, we need to run the py27 virtual environment we created earlier.

1. Select the gear icon in the top right corner. Then select ‘Conda Environment’ as shown in Figure 8.

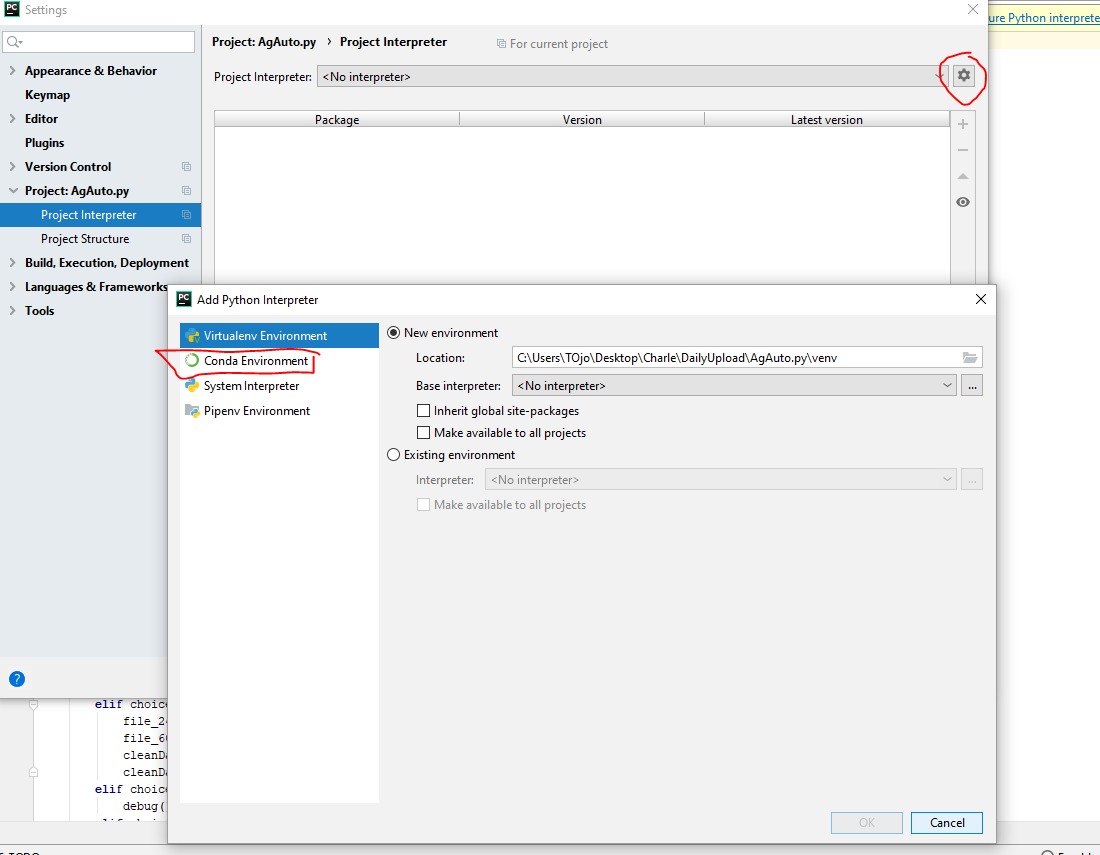


Figure 8: We need to tell PyCharm that we wish to run a conda virtual environment.

1. Navigate to where the python executable for py36 is located, as shown in Figure 9. Then press OK.

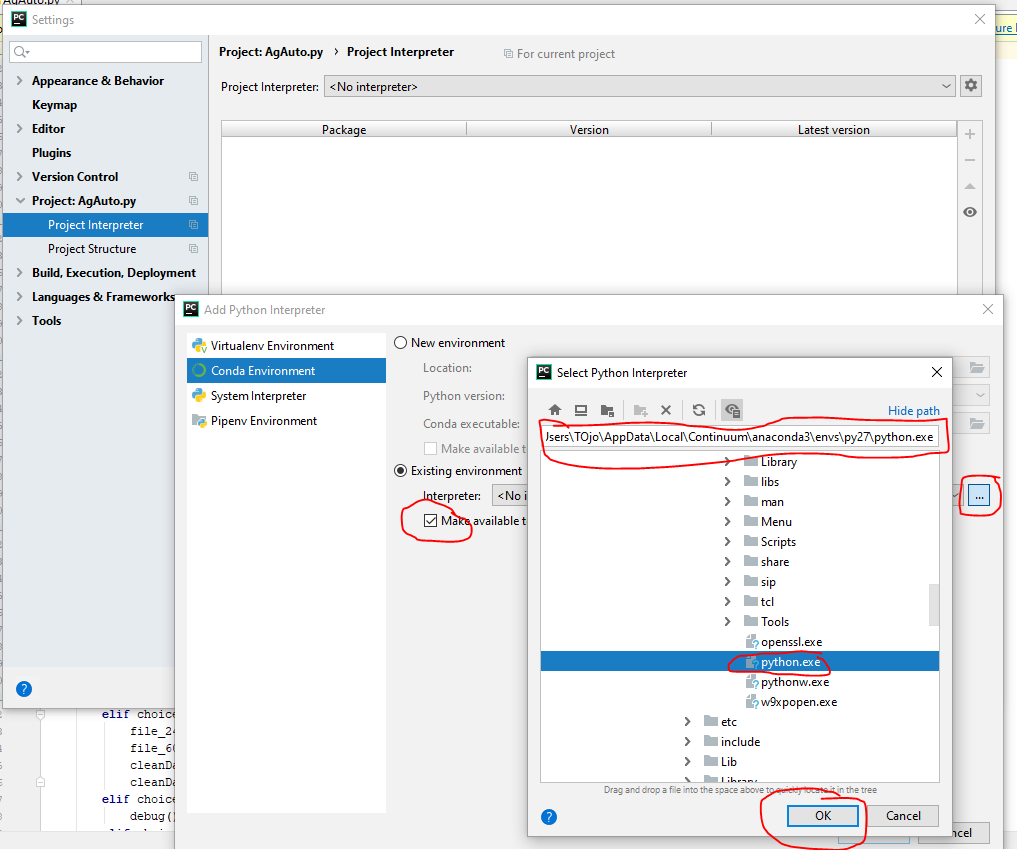


Figure 9: Locate the py36 python executable. Usually it's located in the link shown above.

1. Finally, press Apply and then OK to complete the process. It might take PyCharm 10 minutes to fully load the packages.

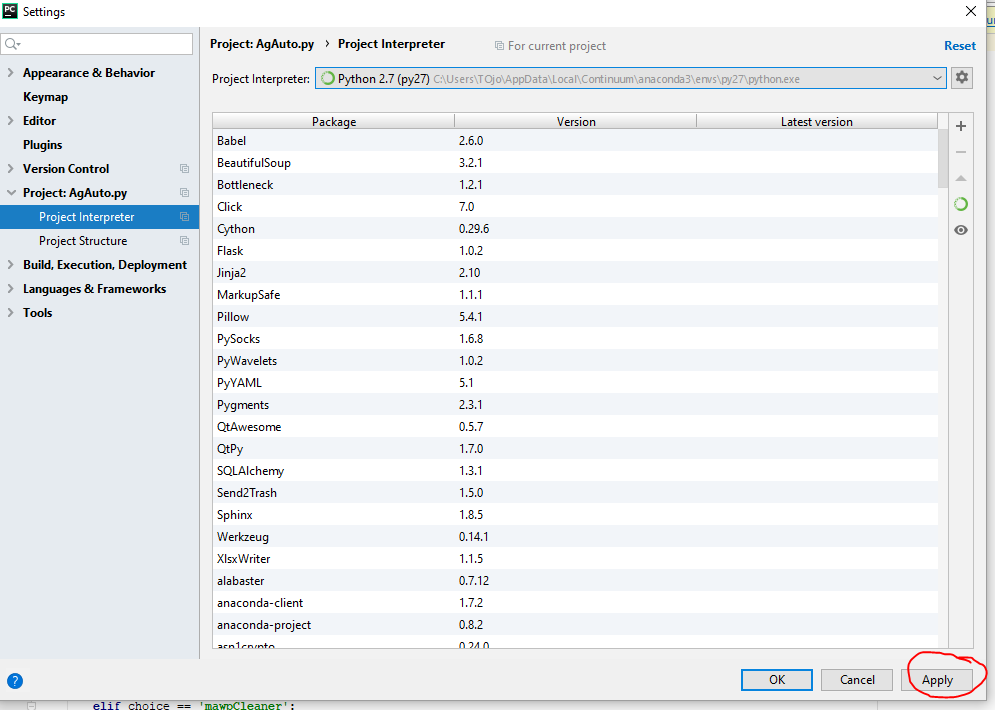


Figure 10: Finalize the interpreter selection process.

# Procedure

Three data files are uploaded into the ME every morning: DailyEC.csv, mawp24raw.txt and mawp60raw.txt

NOTE: In order to run the script, you must ensure that your computer is outside of the managed internet environment; connecting to your phones WiFi will suffice. However, in order to place the new files into the upload folder you must switch back to the managed environment.

1. Access the latest ‘DailyEC.csv’ file from the archive folder: [\\MBPApp0964P\Shared\_Data\AgWeather\archive](file:///\\MBPApp0964P\Shared_Data\AgWeather\archive) and copy into the script’s working directory (i.e. make sure that DailyEC.csv and AgAuto.py are in the same folder).
   1. Note: If you are just beginning to gather daily summary data for the start of the growing season, make sure that **‘DailyEC.csv’ does not exist in the scripts working directory** and proceed to step 2.

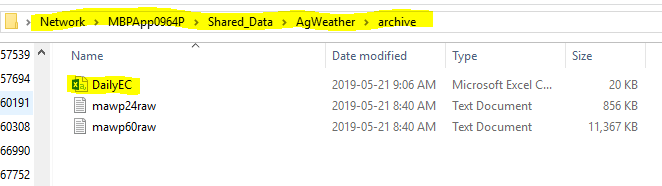


Figure 11: Directory containing latest DailyEC.csv file.

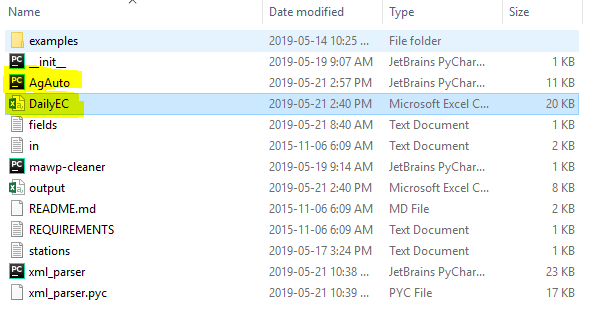


Figure 12: User defined directory containing AgAuto.py script.

1. Open the AgAuto.py script using a python editor (e.g. PyCharm, Spyder, Eclipse, etc.)
2. Figure 3 below shows the PyCharm editor after opening AgAuto.py. In order to run the script, press the green triangle on the top right corner.

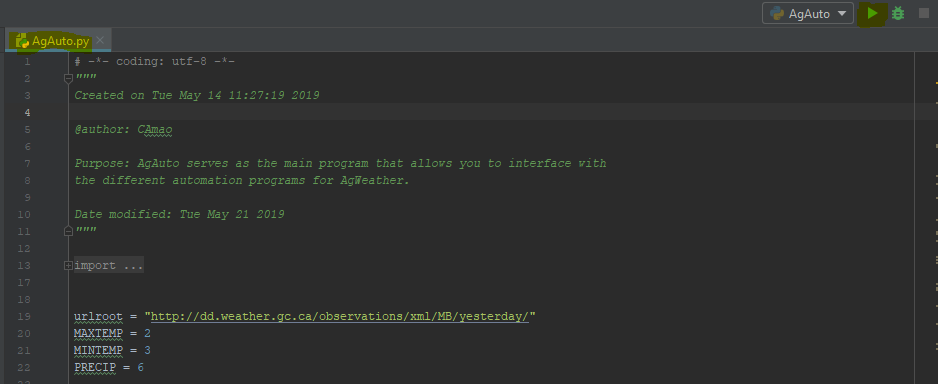


Figure 13: PyCharm editor after opening AgAuto.py. Note the 'Run' button on the top right corner.

1. Figure 4 shows the user-interface of the AgAuto program. The program lists 4 options: dailyUpload, mawpCleaner, debug, and Quit. At the prompt “Which program do you want to run?:” type dailyUpload or 1 and press ENTER.

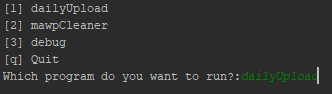


Figure 14: The different options of the AgAuto program at the time of writing this document.

* 1. Figure 5 shows what the console looks like after the program successfully cleans the DailyEC, mawp24raw.txt, and mawp60raw.txt files.

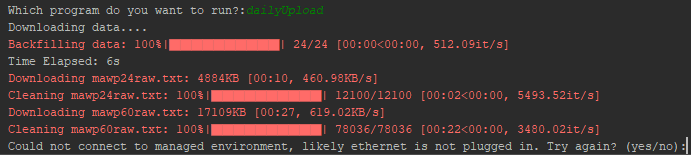


Figure 15: Successful execution of the dailyUpload program.

1. However the program prompts you to reconnect to the managed environment and type ‘yes’ when done. After typing ‘yes’ and pressing ENTER, the program then copies the file onto the ME ‘upload’ folder.

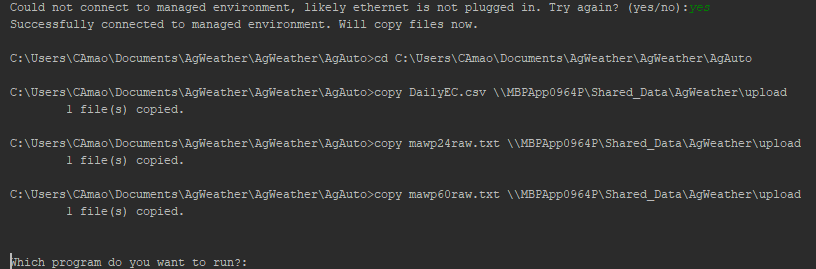


Figure 16: Successful upload to the 'Upload' Folder.

1. Type ‘q’ and press ENTER to stop the script.