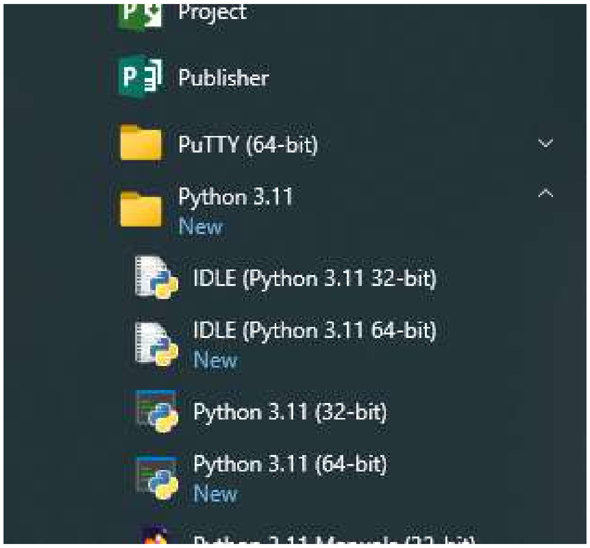
Download and install Python3:

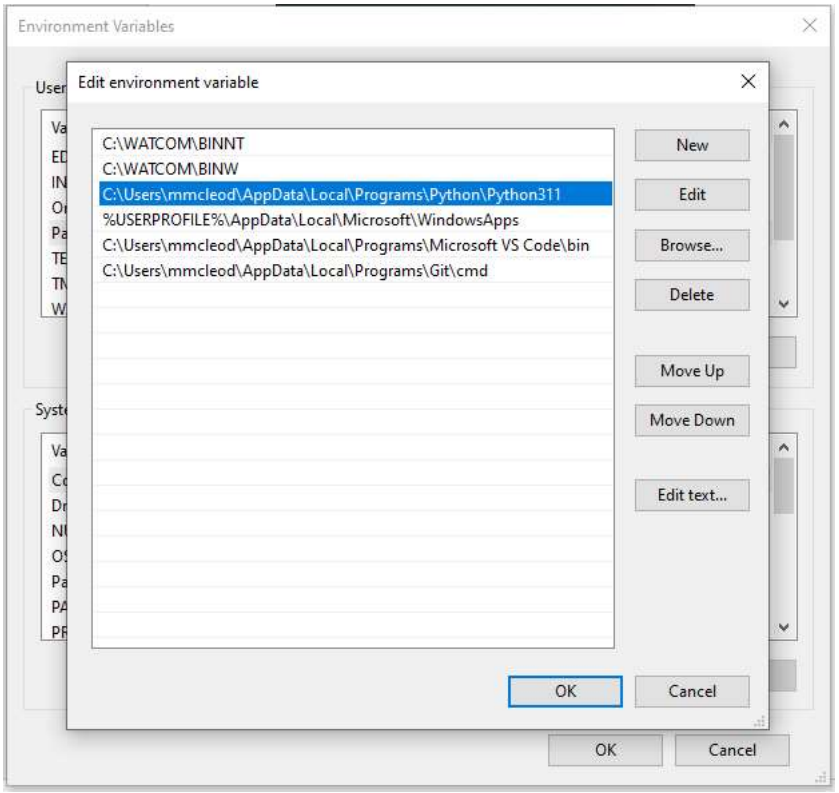
https://www.python.org/downloads/

I’m using 3.11.3 64-bit for Windows.

Installed into C:\Users\mmcleod\AppData\Local\Programs\Python\Python311



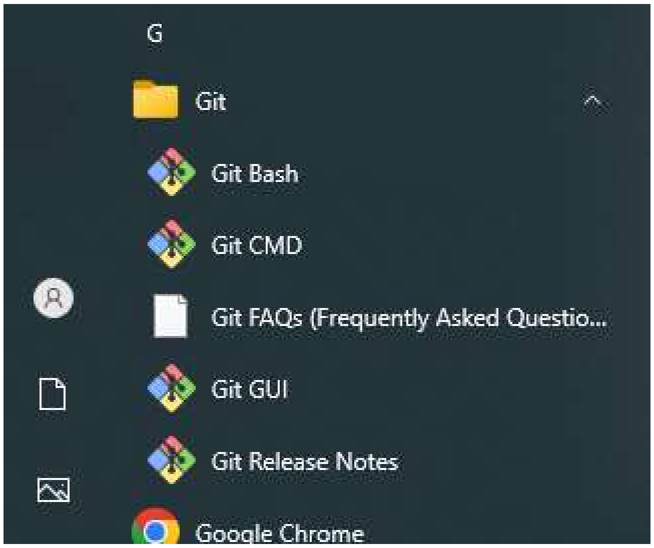
Add it to the path before Microsoft\Windows\Apps:



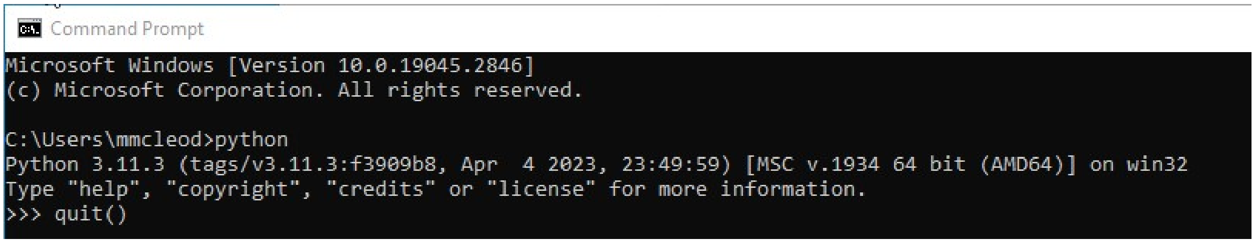
```
C:\WATCOM\BINNT
C:\WATCOM\BINW
C:\Users\mmcleod\AppData\Local\Programs\Python\Python311
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\mmcleod\AppData\Local\Programs\Microsoft VS Code\bin
C:\Users\mmcleod\AppData\Local\Programs\Git\cmd
```

Download and install Git for Windows:

https://gitforwindows.org/



Now in a command prompt, the “python” command should open a session with the version you just installed.



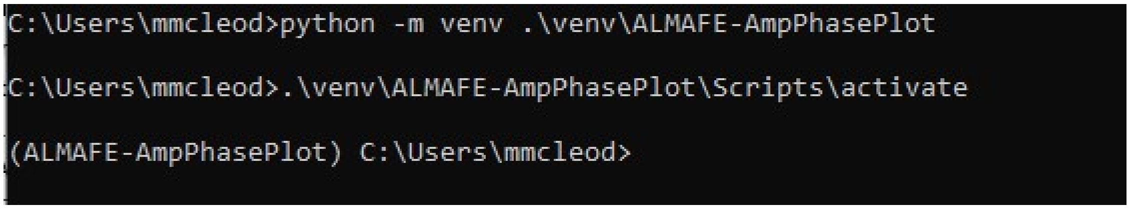
```
Microsoft Windows [Version 10.0.19045.2846]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mmcleod>python
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> quit()
```

(Recommended but optional) create and activate a virtual environment:

```
python -m venv .\venv\ALMAFE-AmpPhasePlot
```

```
.\venv\ALMAFE-AmpPhasePlot\Scripts\activate
```



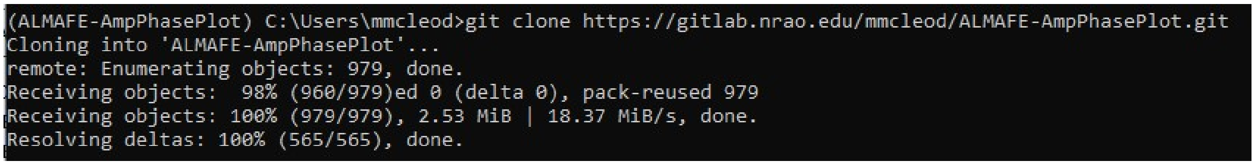
```
C:\Users\mmcleod>python -m venv .\venv\ALMAFE-AmpPhasePlot

C:\Users\mmcleod>.\venv\ALMAFE-AmpPhasePlot\Scripts\activate

(ALMAFE-AmpPhasePlot) C:\Users\mmcleod>
```

Get the code:

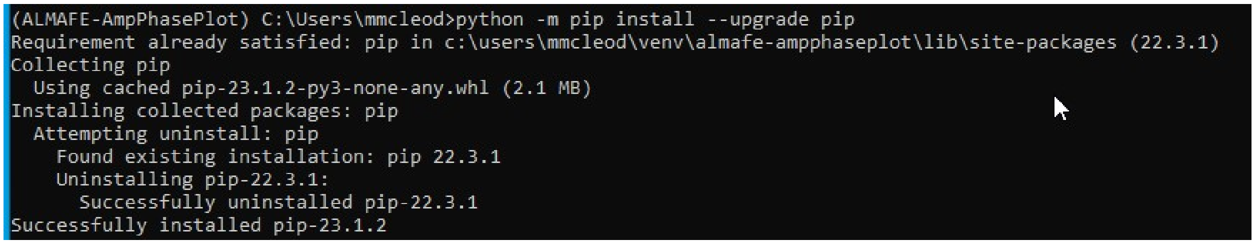
```
git clone https://gitlab.nrao.edu/mmcleod/ALMAFE-AmpPhasePlot.git
```



```
(ALMAFE-AmpPhasePlot) C:\Users\mmcleod>git clone https://gitlab.nrao.edu/mmcleod/ALMAFE-AmpPhasePlot.git
Cloning into 'ALMAFE-AmpPhasePlot'...
remote: Enumerating objects: 979, done.
Receiving objects:  98% (960/979)ed 0 (delta 0), pack-reused 979
Receiving objects: 100% (979/979), 2.53 MiB | 18.37 MiB/s, done.
Resolving deltas: 100% (565/565), done.
```

Best practice: Update PIP before using:

```
python -m pip install --upgrade pip
```

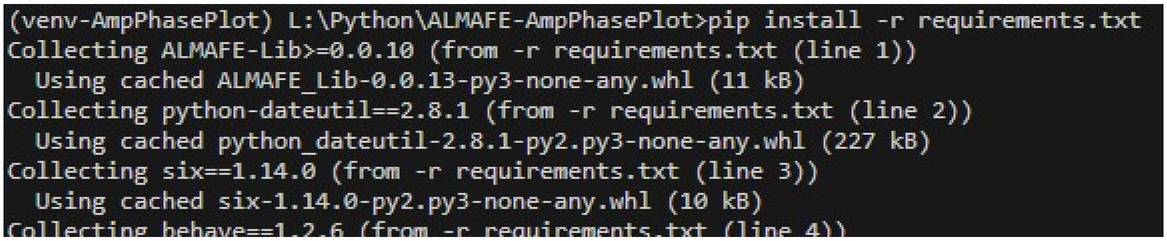


```
(ALMAFE-AmpPhasePlot) C:\Users\mmcleod>python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\mmcleod\venv\almafe-ampphaseplot\lib\site-packages (22.3.1)
Collecting pip
  Using cached pip-23.1.2-py3-none-any.whl (2.1 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.3.1
    Uninstalling pip-22.3.1:
      Successfully uninstalled pip-22.3.1
Successfully installed pip-23.1.2
```

Get the required Python packages:

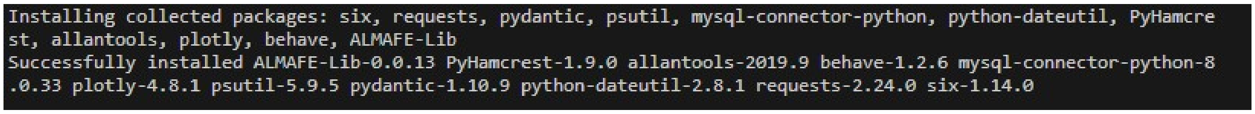
```
cd ALMAFE-AmpPhasePlot

pip install -r requirements.txt
```



```
(venv-AmpPhasePlot) L:\Python\ALMAFE-AmpPhasePlot>pip install -r requirements.txt
Collecting ALMAFE-Lib>=0.0.10 (from -r requirements.txt (line 1))
  Using cached ALMAFE_Lib-0.0.13-py3-none-any.whl (11 kB)
Collecting python-dateutil==2.8.1 (from -r requirements.txt (line 2))
  Using cached python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
Collecting six==1.14.0 (from -r requirements.txt (line 3))
  Using cached six-1.14.0-py2.py3-none-any.whl (10 kB)
Collecting behave==1.2.6 (from -r requirements.txt (line 4))
```

…eventually...



```
Installing collected packages: six, requests, pydantic, psutil, mysql-connector-python, python-dateutil, PyHamcre
st, allantools, plotly, behave, ALMAFE-Lib
Successfully installed ALMAFE-Lib-0.0.13 PyHamcrest-1.9.0 allantools-2019.9 behave-1.2.6 mysql-connector-python-8
.0.33 plotly-4.8.1 psutil-5.9.5 pydantic-1.10.9 python-dateutil-2.8.1 requests-2.24.0 six-1.14.0
```

Install ORCA:

Get the latest Windows build here: https://github.com/plotly/orca/releases

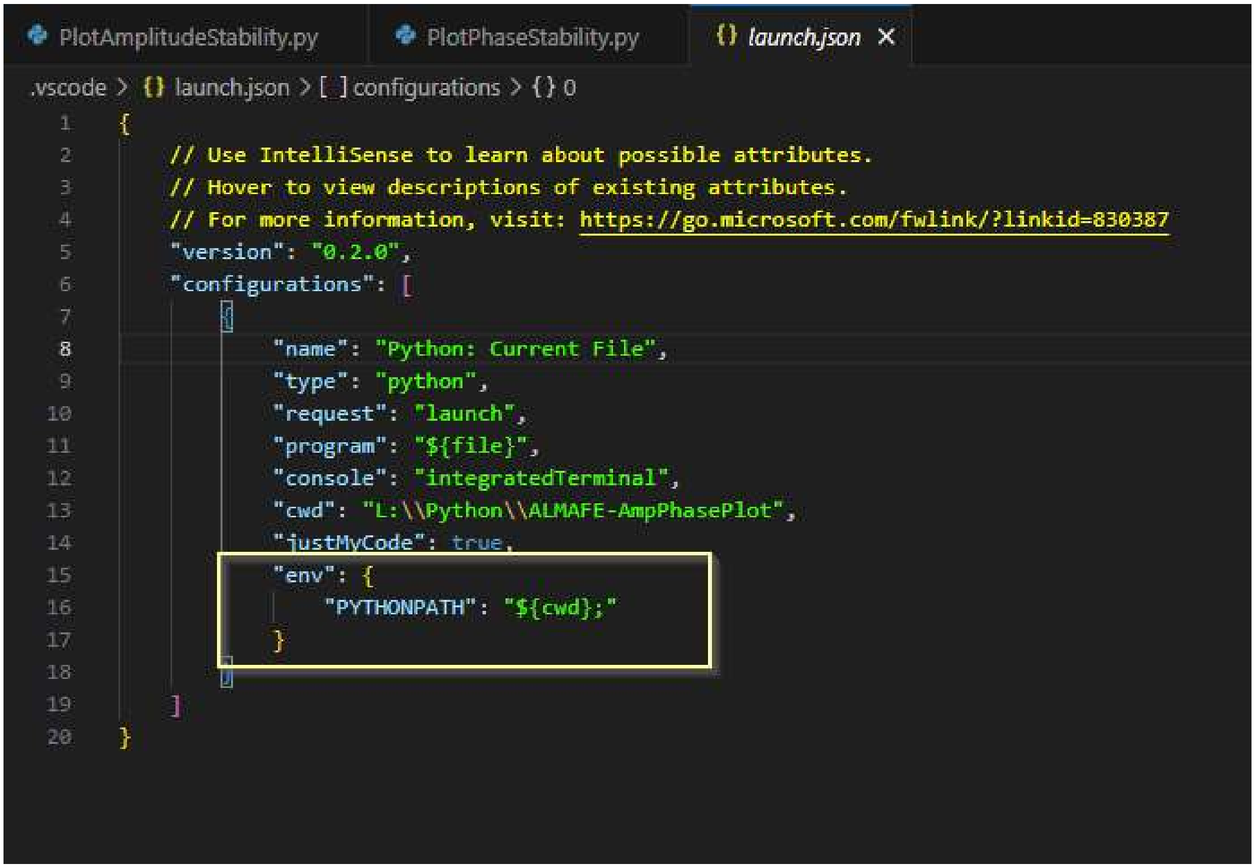
Follow the instructions here: https://github.com/plotly/orca

The top-level plotting scripts in the Python package are:

tests\Integration\PlotAmplitudeStability.py

tests\Integration\PlotPhaseStability.py

When running these, the top-level directory of the project needs to be on the PYTHONPATH for the scripts to work. In Microsoft Visual Studio Code this can be achieved by setting up your run configuration like this:


PlotAmplitudeStability.py
PlotPhaseStability.py
{} launch.json ×
.vscode > {} launch.json > [ ] configurations > { } 0

```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "name": "Python: Current File",
            "type": "python",
            "request": "launch",
            "program": "${file}",
            "console": "integratedTerminal",
            "cwd": "L:\\Python\\ALMAFE-AmpPhasePlot",
            "justMyCode": true,
            "env": {
                "PYTHONPATH": "${cwd};"
            }
        }
    ]
}
```

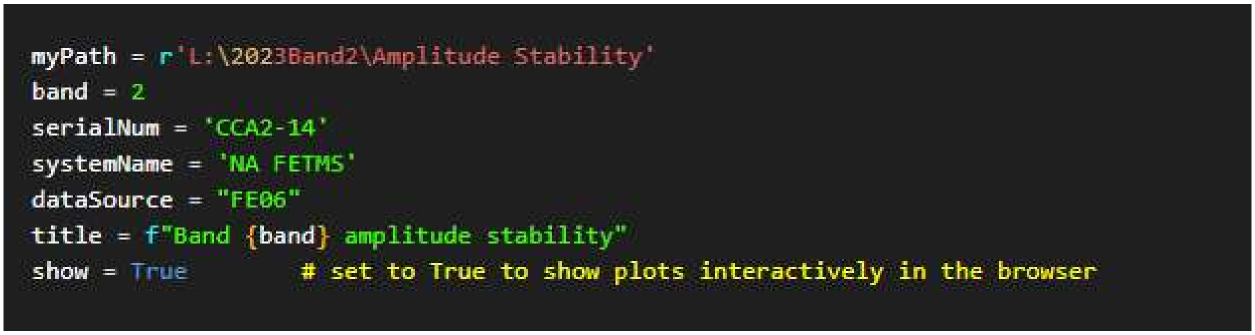
This link shows a way to do it with a Windows batch file:

https://stackoverflow.com/questions/4580101/python-add-pythonpath-during-command-line-module-run

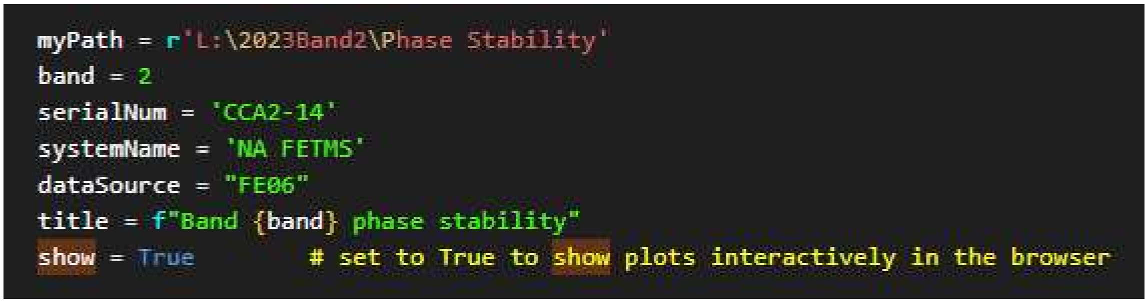
Or it seems you can simply add an environment variable.

In my examples above, PYTHONPATH would need to include L:\Python\ALMAFE-AmpPhasePlot

Finally, modify the variables at the top of those two scripts with the path your raw data files, and update the other metadata:



```
myPath = r'L:\2023Band2\Amplitude Stability'
band = 2
serialNum = 'CCA2-14'
systemName = 'NA FETMS'
dataSource = "FE06"
title = f"Band {band} amplitude stability"
show = True        # set to True to show plots interactively in the browser
```



```
myPath = r'L:\2023Band2\Phase Stability'
band = 2
serialNum = 'CCA2-14'
systemName = 'NA FETMS'
dataSource = "FE06"
title = f"Band {band} phase stability"
show = True        # set to True to show plots interactively in the browser
```

Then run either script.