# Software Setup: Python and JupyterLab

## Python

### Check for existing installation.

Open a **non-administrative** PowerShell and run the following command:

python --version

If that gives you an error, try running.

python3 --version

If both commands error, move on to [Installing Python](#_Installing_Python). If you get output from one of the commands that looks like



As long as you have a version number that begins with “3” (such as 3.11.0), you should be able to do. You can now continue with [Installing your Virtual Environment](#_Installing_your_Virtual)

### Installing Python

1. Download the installer from <https://www.python.org/downloads/> (at the time of writing this document version 3.11.2).
2. Run the installer, but carefully read each screen.
3. The first screen with have an option to “Add python.exe to PATH”. ***Make sure to check that box*** then select “Install Now”.

Graphical user interface, text, application, Teams

Description automatically generated

Open a **new** PowerShell (close the one from the previous step), and run

python --version



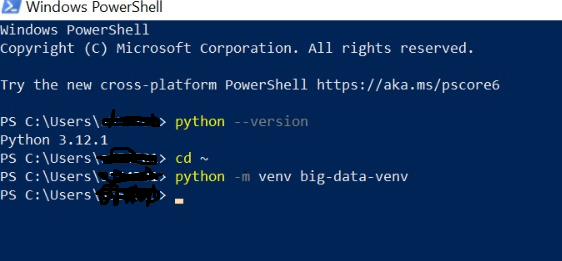
You should now see output similar to the screenshot above; move on to [Installing your Virtual Environment](#_Installing_your_Virtual)

### Installing your Virtual Environment

When python is installed, and you see a Python 3 version number in your shell, run the following commands (note: if you have a directory in your home directory called big-data-venv, you can choose a different directory name, but you will need to remember it when you are [Configuring Microsoft Terminal](#_Configuring_Windows_Terminal)):

cd ~

python -m venv big-data-venv



There should be no output from these commands. At this point, if you have already installed Microsoft Terminal, move on to [Configuring Windows Terminal](#_Configuring_Microsoft_Terminal); otherwise continue to [Installing Windows Terminal](#_Installing_Microsoft_Terminal)

### Installing Windows Terminal

1. Download the installer from the latest release from <https://github.com/microsoft/terminal/releases>.
2. Specifically, you want the one that has an extension of msixbundle (not .zip) and is for your version of Windows (either 10 or 11). At the writing of this document, the file name for Windows 10 is Microsoft.WindowsTerminal\_Win10\_1.16.10261.0\_8wekyb3d8bbwe.msixbundle

Graphical user interface, text, application

Description automatically generated

Run the installer; all default options should be fine.

### Configuring Windows Terminal

We will use Windows Terminal to make it easy for us to open a PowerShell with the virtual environment activated.

1. Open your newly installed Windows Terminal.
2. In the upper bar, click the little down arrow. We are going to configure the terminal to have a profile for our Virtual Environment.

Graphical user interface, text, application

Description automatically generated

1. Select the “Settings” option (note, you can also open the settings window with the keyboard shortcut “Ctrl+,”

Graphical user interface, text

Description automatically generated

Don’t worry about my other profiles, you will add to these as you want more specialized tools or install other shells (like WSL or Git Bash)

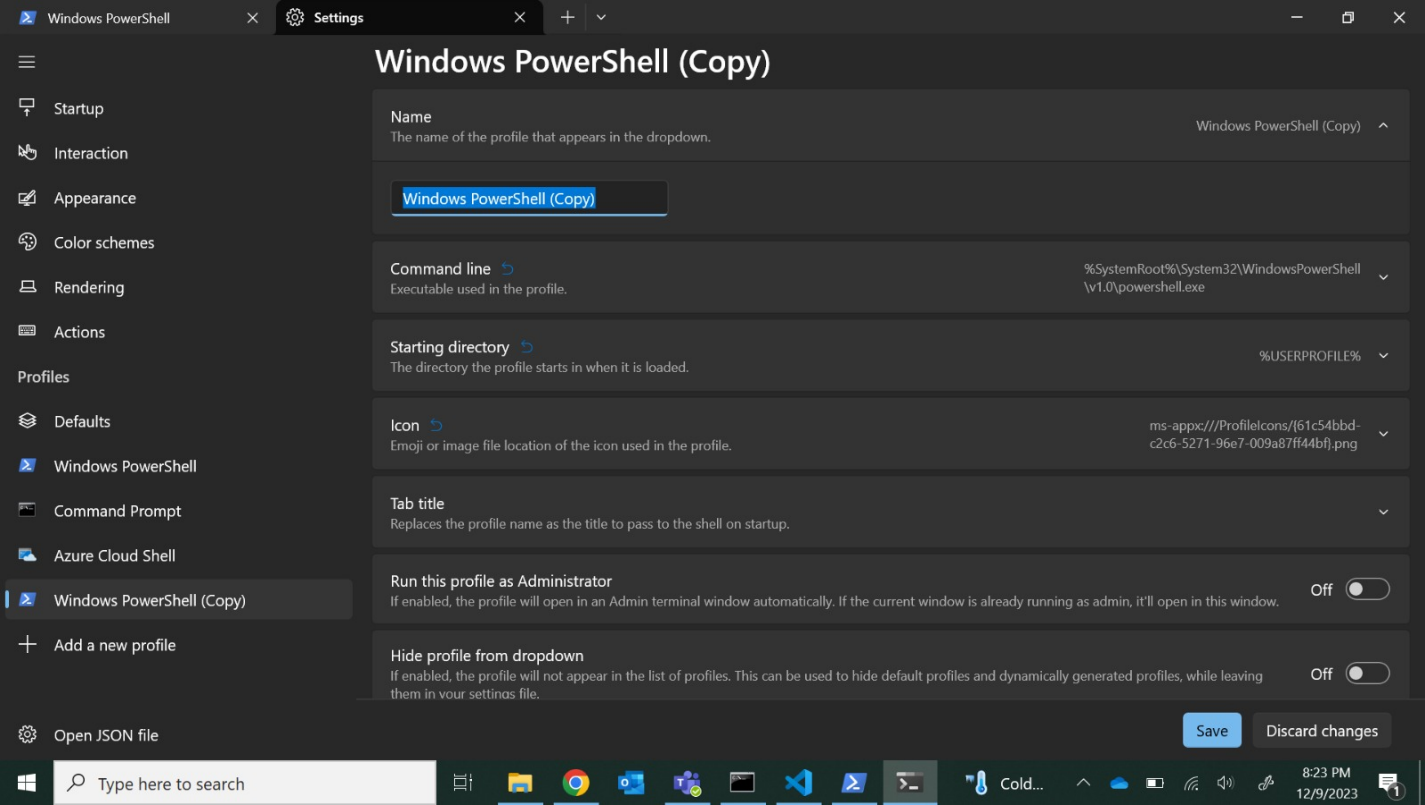
1. In the settings window, scroll down in the left pane until you see + Add a new profile, and select that option. We are going to choose to Duplicate the PowerShell profile.

Background pattern

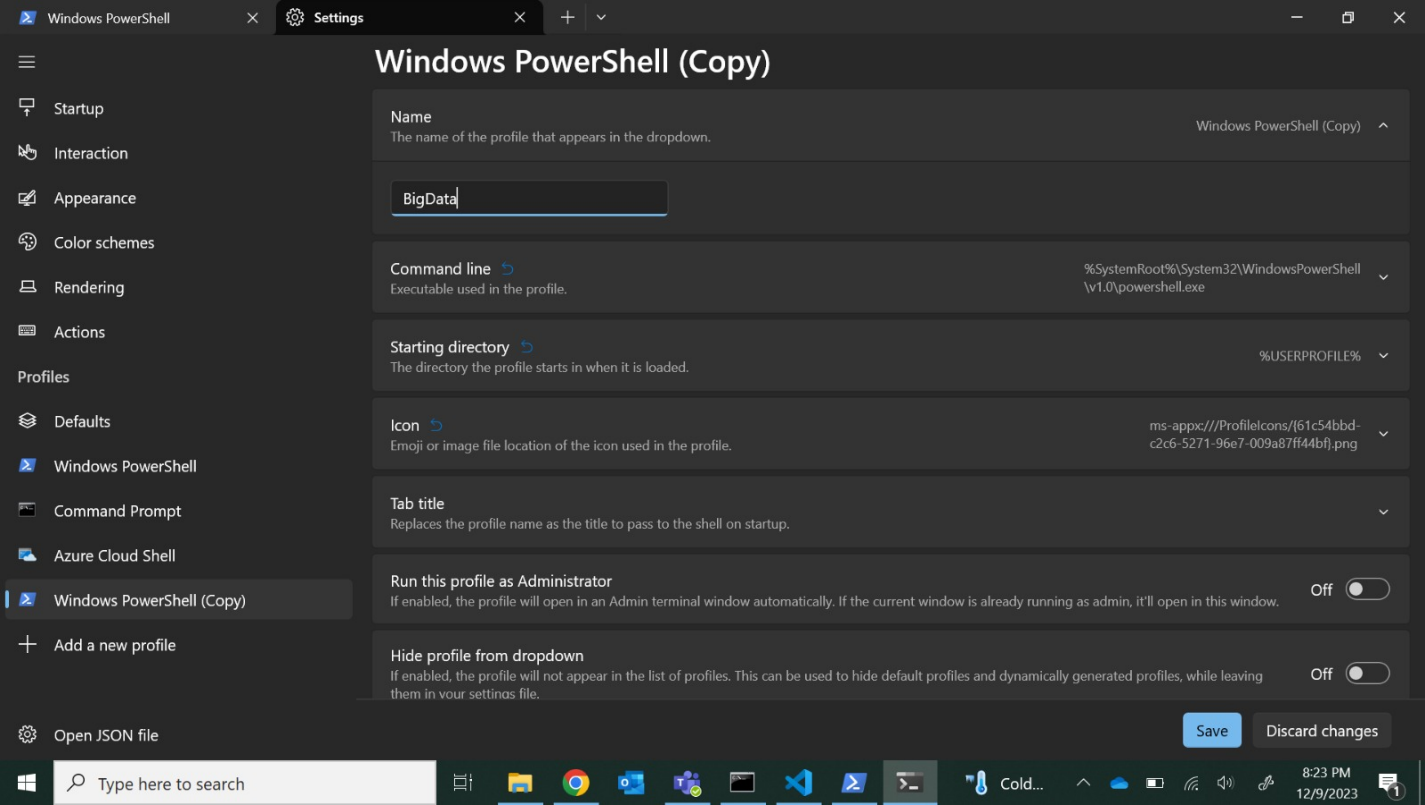
Description automatically generated with low confidence

1. Change the Name of the profile by clicking the name field and typing a descriptive Name, here I used BigData (notice that it starts as “Windows PowerShell (copy)”

**Before:**

****

**After:**

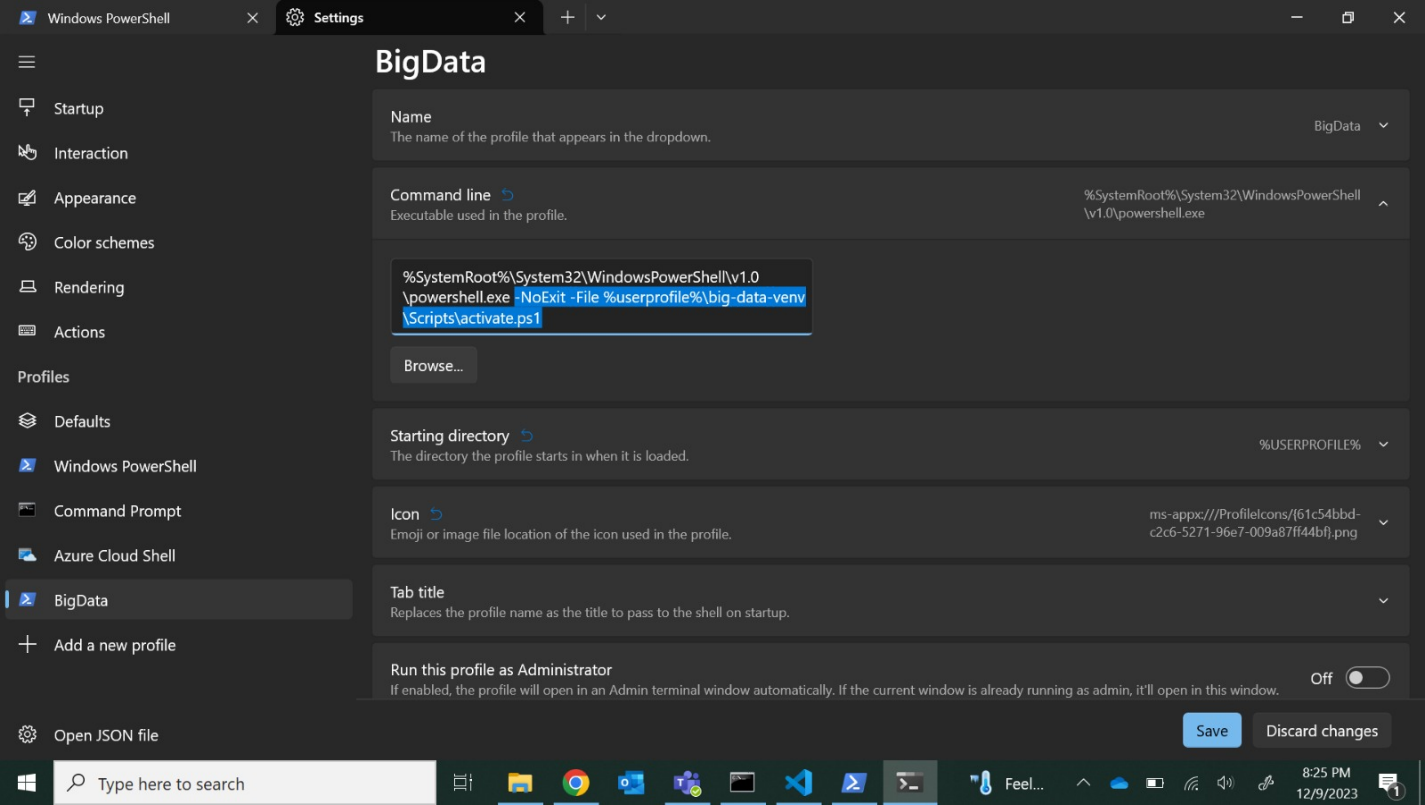
****

Open the Command line argument. Leave the powershell command, but add a space and

-NoExit -File %userprofile%\big-data-venv\Scripts\activate.ps1

To the end. The entire command should now be:

%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe -NoExit -File %userprofile%\big-data-venv\Scripts\activate.ps1



If you had to use a different folder name when creating the virtual environment, replace big-data-venv with that folder name. Save your settings.

### Installing JupyterLab

1. Open your new profile by clicking the down arrow in the top bar.
2. Select the profile with the name you gave it in the previous step. You should now see that the prompt is preceded by (big-data-venv).

We will install packages using the pip command. In the virtual environment run the following command

(**Note**: If pip install throws error, please add **python -m** before pip install.

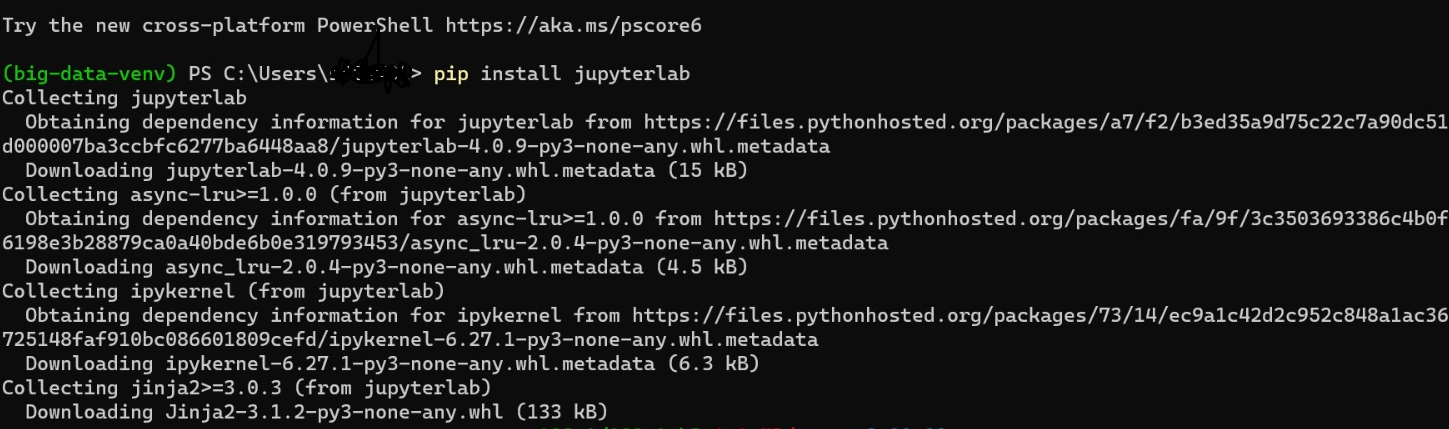
Ex: Python -m pip install package\_name)

pip install jupyterlab

If this fails because of a broken prerequisite, try installing an older version of jupyterlab:

pip install jupyterlab==3.5.3

you will be able to see similar output as here.



Open JupyterLab by running the following command

python -m jupyterlab

It should open a web browser with a notebook, but if not, navigate a web browser to localhost:8888/lab.

# Software Setup: PySpark

## Prerequisite software

This assignment assumes you have the following correctly installed:

* Hadoop/winutils (latest version available), with the HADOOP\_HOME environment variable set
* Java (at least 8, tested with Java 11 and 17)
* Python 3.x
* A Python virtual environment with Jupyter Lab installed

## Installing PySpark

1. Open your virtual environment in a terminal.
2. Install pyspark and pandas using pip by running  
   pip install pyspark pandas