# Setting Up Your Environment

In this class we will be using dockers for the environment in all our labs. The docker provided is already configured with Python3.6.9 and several popular packages used in data science. It works on both Linux and Windows system.

*Note: You should also be able to set up a similar environment on your machine without docker. However, the course staff will not support these configurations. So you’re on your own if you choose to go that route!*

## Download and Install

The installation and use of docker on Linux systems and Macs is pretty simple. You can follow the instructions [here](https://docs.docker.com/engine/install/ubuntu/).

For Windows, you need to use the subsystem for Linux (WSL) provided on Windows 10. The official instructions can be found [here](https://docs.docker.com/docker-for-windows/install/).

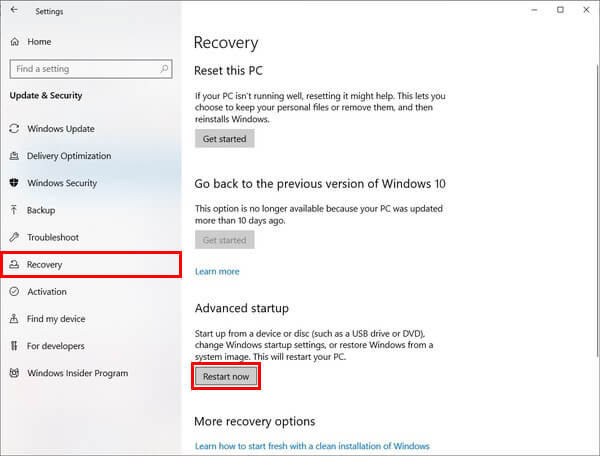
To create a container with the provided docker files:

To use Docker Desktop and Docker on Windows:

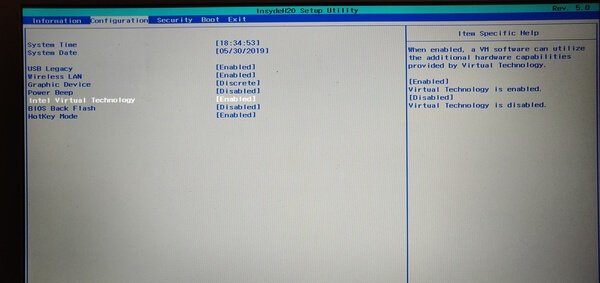
1. Download and install [Docker Desktop](https://docs.docker.com/docker-for-windows/install/)
2. Enable the [Hyper-V](https://mashtips.com/enable-virtualization-windows-10/) and Containers Windows features:
   1. (Win 11 please follow the instructions here: <https://www.makeuseof.com/windows-11-enable-hyper-v/>)
   2. Launch a command prompt or Windows PowerShell window.
   3. Enter the command: ***systeminfo.exe***
   4. Wait a few seconds. It takes a few seconds to gather information about your system before an output is displayed.
   5. Look under Hyper-V Requirements.



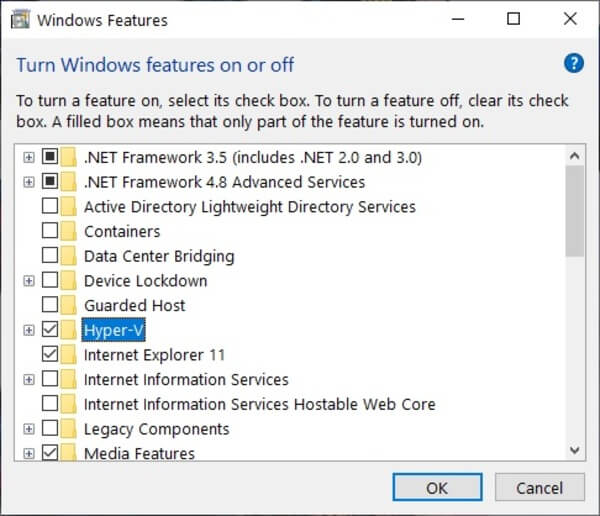
* 1. See that Virtualization Enabled In Firmware reads "Yes". (If it does, Hyper-V is enabled on your machine. If it’s not, you’ll have to enable this in the **BIOS** or **UEFI settings.**) An easy way for UEFI-boot machines is to do this from Windows 10 Settings:
     1. Go to **Windows Settings**
     2. Click on **Update & Security**
     3. Now click on **Recovery**
     4. Click **Restart,** now under Advanced startup.

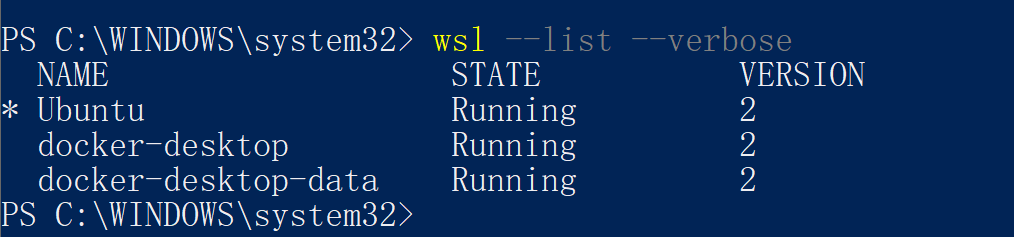
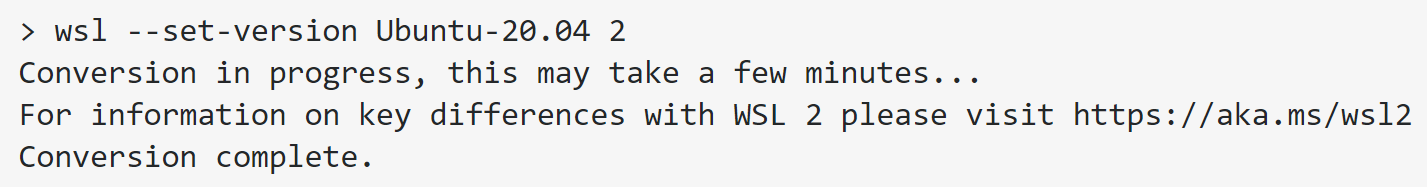


* + 1. Go to **Troubleshoot**
    2. Click on **Advanced Options**
    3. Now click on **UEFI Firmware Settings**
    4. Click **Restart**.



* 1. Look for the setting which has **Virtualization** in its name and turn it on.
  2. Save the change and launch your machine. Press the Windows key to get the Search box.
  3. Type “turn windows features on or off” and click on it to open it.
  4. Scroll down and check the box next to "Hyper-V". This step may differ for different versions of Windows. In other versions, "Windows hypervisor platform" may be displayed rather than "Hyper-V."



1. Download/update subsystems for Linux to WSL2:
   1. As in 2.h, Before installing WSL 2, you must enable the **Virtual Machine Platform** optional feature.
   2. Download the Linux kernel update package
      1. [WSL2 Linux kernel update package for x64 machines](https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi)
         * Note: If you're using an ARM64 machine, please download the ARM64 package instead. If you're not sure what kind of machine you have, open Command Prompt or PowerShell and enter: systeminfo | find "System Type".
      2. Run the update package downloaded in the previous step. (Double-click to run. You will be prompted for elevated permissions. Select ‘yes’ to approve the installation.)
   3. If you do not have the subsystem on your machine, follow i. If you already have one on your machine and want to update it, follow ii:
      1. Set WSL 2 as your default version and then install a subsystem:
         * Open PowerShell and run this command to set WSL 2 as the default version when installing a new Linux distribution: wsl --set-default-version 2
         * Open the [Microsoft Store](https://aka.ms/wslstore) and select your favorite Linux distribution. We recommend you install [Ubuntu 20.4](https://www.microsoft.com/store/apps/9n6svws3rx71)
         * After installation, a console will open and you will need to [create a user account and password for your new Linux distribution](https://docs.microsoft.com/en-us/windows/wsl/user-support).
      2. Change the default version to wsl2 for the previous subsystem:
         * Open PowerShell and run wsl --list --verbose which will give you a list of your wsl running processes: 
         * Then change the one you want to use with command: wsl --set-version <your proc> 2. This process may take a few minutes, depending on the size of you subsystem
2. Now open Docker Desktop and sign in to check all available docker images you have on Docker Hub. You should also be able to use the docker command in your sub-Linux-system. (If it still shows a WSL2 problem, just restart you machine to make the above changes take effect).
3. We also provide a Dockerfile from which you can automatically build the image. Create a directory to serve as your working directory if one doesn't already exist. Download the Dockerfile to your directory (link needed). To build the image, use docker build --tag cap4770:1.0 . You can now see your docker image by docker image ls.

After you have your image built, to test that everything is set up properly, create a container by running the following command in a subsystem terminal window:

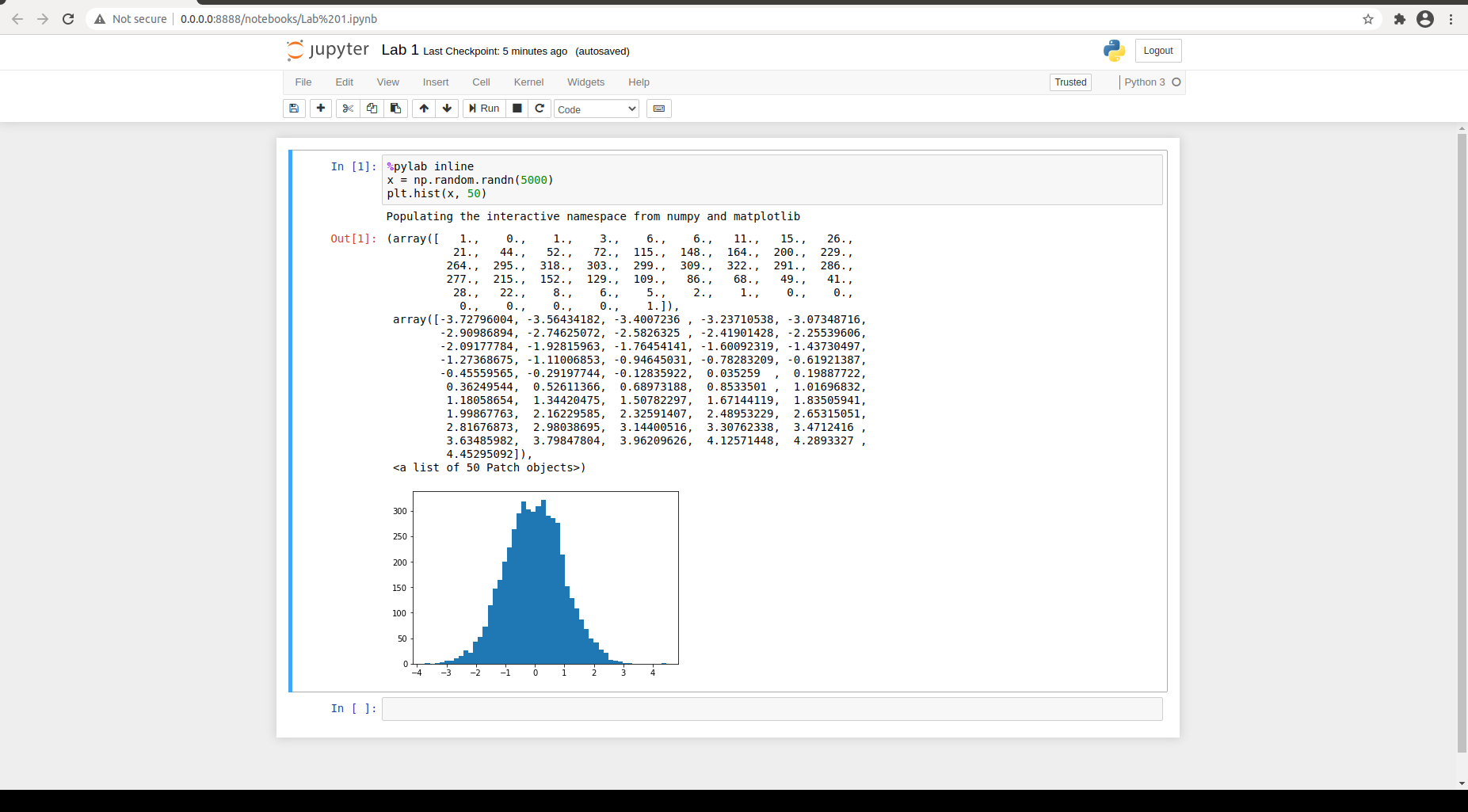
1. Create a container from your image with the command: docker run -v $(pwd):/home/datascience -p 8888:8888 -it cap4770:1.0
2. In the browser window that pops up (If the browser does not open automatically, open your browser and go to http://localhost:8888.), create a new notebook, and enter the following in the first cell

%pylab inline

x = np.random.randn(5000)

plt.hist(x, 50)

1. Hit the ‘Play’ button on the toolbar.
2. You should end up with something like this:



1. Close your browser. In the terminal hit Control-C and Y to exit.
2. Once you stop jupyter notebook, your container should be stopped as well. If you want to stop the container by hand, use docker stop [container\_name]. Check the status by docker container ls -a. To restart the container, you can use docker start -a [container\_name]. This will start the container and Jupyter Notebook again. If your container is not stopped and you want to enter it again, use docker exec -it [container\_name] bash which will let you enter the container with command line. Other Instructions about these command can be found on the official website.

## Credentials

The token for the notebook is displayed in the terminal. It changes each time the container is started.

## miscellaneous Information

Software installed in the Docker.

* jupyter-client (5.2.2)
* jupyter-core (4.4.0)
* pandas (0.22.0)
* numpy (1.13.3)
* python (3.6.9)
* scipy (0.19.1)
* matplotlib (2.1.1)
* scikit-learn (0.19.1)