CMSE 202 - Spring 2019
Software Setup Guide

As this is a course in computational modeling and data science, you will be completing all of your assignments using your computer! However, in order to do so there are a number of things you need to set up before the course starts. If you run into issues during this setup process make sure to document the error you encountered and send an email to your Professor to let them know that you ran into a problem.

MAKE SURE TO COMPLETE ALL OF THE SECTIONS LISTED IN THIS DOCUMENT BEFORE YOU COME TO CLASS

Installing Python for this course

You need to have a functioning and **current** Anaconda Python installation on your computer for this course. If you have a past installation, you are expected to make sure it is up-to-date, which you can do by opening up a command line interface and running `conda update --all`. In addition to making sure your installation is updated, you should also ensure that the Anaconda installation is in your default path. If you don't know how to do the update or ensure that Anaconda is in your path, you are encouraged to install a fresh version of Anaconda following the directions below.

If you don't having a fully functioning up-to-date installation...

Follow the instructions below to get the Anaconda distribution of Python installed on your computer.

Instructions for downloading Anaconda (Python 3.7.x):

1. Go to the Anaconda Download webpage: https://www.anaconda.com/download/
2. Use the “Jump to: Windows | OS X | Linux” to pick your operating system.
3. Download the Python 3.7 version (64 bit recommended).
4. Follow the online documentation to install Python for your specific operating system: https://docs.anaconda.com/anaconda/install/
   Important: When you're asked if you want to add Anaconda to your path choose **yes**.
5. Open the command line program on your computer
   - On Windows, type CMD in the run box in the Start menu.
   - On Mac, type “terminal” in the spotlight search and run the "Terminal" application

- On Linux, open up the "Console" application

6. Type “jupyter notebook” in the command line and hit enter

If everything goes correctly, a browser window should open up with the Jupyter interface running. If things don’t work, don’t worry, we will help you get started.

MSU's JupyterHub Interface

From time to time, you might run into issues with your computer. When this happens, you should use the web-hosted JupyterHub server managed by MSU. It creates a virtual environment that allows you to run simple commands and host Jupyter notebooks. To make sure that you have access to this backup option, follow the directions below. Note that there are extra steps involved that require that you upload and download you Jupyter notebooks to and from JupyterHub in order to turn them in on D2L.

Instructions for connecting to the engineering JupyterHub server:

Every student enrolled in this class will be given an engineering computing account. If this is your first time using your Engineering account you will need to activate the account by going to the following website:

https://www.egr.msu.edu/decs/myaccount/?page=activate

Enter your MSU NetID. The initial password will be your APID with an @ on the end (example: A12345678@) and then you have to set a password that meets the requirements listed on the page. Verify the password. Then agree to the terms and Activate.

Once your account is activated you can access the classroom Jupyterhub server using the following instructions:

1. Open up a web browser and go to the following URL: https://jupyterhub.egr.msu.edu
2. Type your engineering login name. This will be your MSU NetID.
3. Type your engineering password.

If everything is working properly you will see the main “Files” windows in the Jupyter interface.

If you ever end up working on your assignments using JupyterHub, the remaining directions should serve as a reference for how you can go about uploading and downloading Jupyter notebooks and turning them in.

Instructions for getting Jupyter notebook files into JupyterHub:

IPython notebooks (also referred to as Jupyter notebooks) are files that end with the .ipynb extension. We will

give you these files for all of your assignments, you will edit them and turn in the edited files in using the class Website.

You can download the ipynb assignment files from the class website (http://d2l.msu.edu). Once you have an ipynb file you can load it into Jupyter using the “upload” button on the main “Files” tab in the JupyterHub web interface. Hitting this button will cause a file browser window to open. Just navigate to your ipynb file, select it and hit the open button.

Once you see your filename in the jupyter window you can just click on that name to start using that file.

Instructions for making a copy of Jupyter notebooks from JupyterHub and turning them in:

When you are finished editing your IPython notebook and are ready to turn it in you will need to download the ipynb file from the JupyterHub interface.

1. With the notebook file open in Jupyter, go to the “File” menu, select the “Download as” menu option and then choose “iPython Notebook (.ipynb)”
2. Pick a place to save the file (The desktop is a good choice).
3. Make sure you make a copy of the .ipynb file for your own records.
4. Go to the Desire 2 Learn (http://d2l.msu.edu) class website and upload the .ipynb file into the assignment folder.

NOTE: Video versions of these instructions are available on the CMSE YouTube channel here and here.

Course Communication with Slack

We will be using Slack (http://slack.com) as our means of communicating about course content as the semester progresses. We believe that this will provide an excellent avenue to have discussions not only with course instructors, TAs, and LAs, but also between you and your fellow classmates. In order to join the Slack team that we’ve created for the course you should complete the following steps:

1. Go to https://cmse-courses.slack.com/signup
2. Once there, sign up using your @msu.edu email address. **Important**: When you create your account, use your MSU NetID as your user name. This will make it easier for your instructors to recognize you within the Slack channels.

Once you've joined the CMSE Courses Slack team, you'll need to **add yourself to two channels**. The first

channel you should add yourself to is **"cmse202-s19-help"**. The second one you should add yourself to is **"cmse202-###-s19"** where "###" corresponds to the section of the course that you are enrolled in. To add yourself to these channels, click on "Channels" and search for the appropriate channel.

The "help" channel will be the place to go for any questions about assignments in the course or issues you're having with your computer or Python. We encourage you to help out other classmates when you can! The section-specific channel will be used by your instructor for important messages relevant only to your section of the course.

Slack usage rules

In order to ensure that Slack is a useful tool that does not become overly time-consuming for the course instructors, TAs, or LAs, we have a list of rules for how we expect you to use Slack. They are:

1. Before you ask a question, be sure to check the other section channels to see if the question has already been answered.
2. The Slack team is primarily for you, the students, so help each other.
3. The TAs and LAs will monitor the channels, but will defer to the students to work through things. They will only enter a conversation if students are going down the wrong path and/or there are too few other students involved. However, you should not expect that the TAs or LAs will always be available. The TAs and LAs will spend a limited amount of time “logged in” to Slack and we ask that you be respectful of their time.
4. Slack is meant to be used to help you when you are stuck with a minor issue. If you are having major issues or trouble understanding the concept, go to office/helproom hours. Office/helproom hours are meant for more in-depth discussions of course content.
5. Course instructors will rarely check Slack, only to examine progress. While they may offer help, do not rely on it. Instructors will not respond to the same student twice within a 30 minute time interval.
6. Only in rare cases should you contact an instructor through a private channel. But, if you are struggling, feel free to use this option.
7. **Do not** post your solutions to out-of-class assignments directly into Slack unless prompted by an instructor.
8. Be courteous to everyone on Slack. Students who are being rude or who are excessively posting might be banned from posting on the course Slack channel.

Using Git for version control

We'll be using Git in this course to keep track of changes to our code. Keeping track of the revision history of code called "version control", we'll be discussing this in more detail in class. You'll need to make sure that you have Git installed on your computer. Follow the steps below based on the type of computer you're using.

If you have a Windows computer:

Go to https://git-scm.com/downloads and download the most recent version for Windows. Run the installer and during the installation process make sure to do the following:

- Make sure to install the GitBash software along with Git (it should already be pre-selected).
- Choose a text editor other than vim for the default, unless you prefer using vim. If you don't have a preference, choose nano for now.
- Install the Git commands into the command prompt (not Unix commands though unless you specifically want that functionality).
- Choose OpenSSL when given the option.
- When asked about configuring the line ending conversions, keep 'Checkout Windows-style, commit Unix-style endings.'
- Leave it set to use MinTTY.

Once you finish the installation process, you should be able to open the CMD prompt and run `git --version` to see if Git has been successfully installed.

If you have a Mac computer:

You may already have Git installed on your computer. You can check if this is the case by opening the "Terminal" application (search for it in Spotlight if you don't know where it is) and typing `git --version` and hitting enter. If you already have Git installed, it should tell you what version you have. If Git isn't installed, it will either tell you that it can't find Git, or it will ask if you want to install the "Command Line Tools" to install Git. You're welcome to use the version on you machine, or the version that Command Line Tools installs, but if you want the newest version of Git you'll have to install it yourself.

If you want to install the newest version, go here and download the version for Mac: https://git-scm.com/downloads. Once you've downloaded the installer, follow the directions. *Warning*: you may run into issues with the installer that is provided on the Git website, if this is the case, just stick to the version that comes with your Mac.

Once you finish the installation process, you should be able to open the Terminal and run `git --version` to see if Git has been successfully installed.

If you have a Linux computer:

In the "Console" application, use the appropriate command from this page: https://git-scm.com/download/linux

Once you finish the installation process, you should be able to open the Console and run `git --version` to see if Git has been successfully installed.

Setting up a GitHub account

Now that you have Git installed, you need to set up an account on GitHub. Go to https://github.com/ and sign up for an account **using your "@msu.edu" email address**. It is important that you sign up using your "@msu.edu" account so that you can get the *Student Pack*, which gives you special access to software and unlimited private repositories. The student pack can be found here. **You are expected to request access to the Student Pack before you come to class.** Once you have a GitHub account, the instructors will be able to give you access to the course GitHub repository, which will serve as the mechanism for distributing assignments once you learn how to use Git.

Installing the Atom text editor

In this course you'll be work outside of Jupyter notebooks to write Python scripts that you'll run from the Command Line Interface. In order to write these scripts, you'll need to be comfortable using a text editor. If you don't already have a good text editor that you prefer using, it is recommended that you install Atom. To do this, go to https://atom.io/ and download the version appropriate for your computer.

You may wish to read through the installation information specific to your operating system:

Windows: http://flight-manual.atom.io/getting-started/sections/installing-atom/#platform-windows

Mac: http://flight-manual.atom.io/getting-started/sections/installing-atom/#platform-mac

Linux: http://flight-manual.atom.io/getting-started/sections/installing-atom/#platform-linux

That's it! We look forward to seeing you in class!