CS-1004, Introduction to Programming for Non-Majors, C-Term 2017

Setting up Python 3.5, numpy, and matplotlib on your Macintosh or Linux computer

Hugh C. Lauer©
Adjunct Professor
Worcester Polytechnic Institute

Programming assignments in CS-1004 will be carried out using the programming language *Python* — specifically, version 3.5 of *Python*. In addition, you will need several *Python* packages, including one called *numpy* (meaning "Numerical Python") and one called *matplotlib*, a *Python* version of the popular *Matlab* system. The first part of this document provides instructions for installing *Python 3.5* and packages such as *numpy* and *matplotlib* on Macintosh platforms. Click here to skip forward to the Linux section of this document. [1]

In general, it is expected that assignments will be compatible among Windows, Macintosh, and Linux systems, provided that they all use compatible versions of *Python* and *numpy*.

> **Note:** There are two different, incompatible sets of versions of *Python* in general use around the world — *Python 2.7* and *Python 3.5*. Significant changes to the *Python* language were made between *Python 2.x* and *Python 3.y* (for all values of *x* and *y*). The *Python 3* language is cleaner, more self-consistent, and more user-friendly. Programs written for versions of *Python 2* will not necessarily run on *Python 3* installations; if they do run, they may get different answers to the same problem.

That being said, a lot of legacy *Python 2* code is still in use, and new *Python 2.7* code is still being written and distributed by organizations that have not yet upgraded to *Python 3*. Not all *Python 2* packages have been ported to *Python 3*.

> **Note 2:** Both Mac OS X and Linux come with Python 2.7 already installed for their own needs. That version of Python must not be deleted or disturbed when installing Python 3.5 and its packages.

You also should have "Xcode" installed on your Mac laptop. To install "Xcode" on your Mac laptop, go to App Store and search for "xcode". After you find the app, click on the "Get" button to install "Xcode" on your machine. After the installation is completed, click on "Open" button to start "Xcode". Your "Xcode" installation is not fully completed until you open "Xcode" to register it for the first time.

---

[1]   If you have a Windows computer or laptop, please refer to this documents instead:– docx, pdf
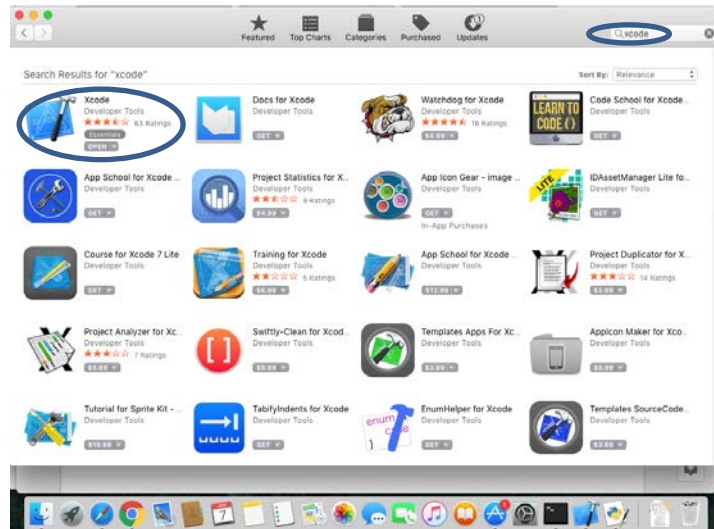
Figure 1

Installing Python 3.5 on Macintosh[2, 3]

Installing *Python 3.5* on Mac OS X presents two challenges:–

- The *IDLE* tool of *Python 3.5* has a dependency on *Tcl/Tk*, an open-source toolkit for building user interfaces. However, *Python 3.5* requires a more recent version than is already installed on Mac OS X.
- Packages such as *numpy 1.11.0* and *matplotlib 1.5.1* are delivered in forms that require installation via commands in a *Terminal* window for installation.

First, create a folder on your desktop — for example, *Python-files* — and download the following two installers to that folder by clicking on these links:– ActiveTcl 8.5.18.0, and python-3.5.1-macosx10.6.pkg.

ActiveTcl 8.5.18.0

> This step requires you to have an Administrative account and password for your Macintosh. Have that ready before you start.

Open the **ActiveTcl 8.5.18.0** file that you just downloaded. You will be asked to confirm your action, and then it displays the following dialog box:–

---

2   It is useful to print out the relevant section of this document. If you read it on-screen, the dialog boxes of the installation tend to obscure the instructions of the document, just when you need them the most!

3   These instructions have been tested on MacOS X 10.10.3 Yosemite. There may be slight differences with other versions of MacOS.

Figure 2

Double-click on the icon that looks like a shipping carton. Under most circumstances,[4] you will next be presented with the following dialog box of the Macintosh *Gatekeeper:*–
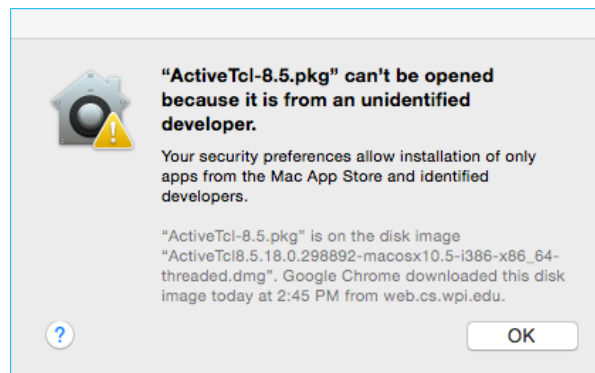

Figure 3

Click *OK,* and then open *System Preferences* under the *Apple* menu. In the *System Preferences* folder, open the *Security and Privacy* icon, shown circled in Figure 4, to bring up the *Security & Privacy* control panel of Figure 5.

---

[4] If your Macintosh already recognizes you as an administrator, it may skip this and subsequent steps and continue the installation of **ActiveTcl**.
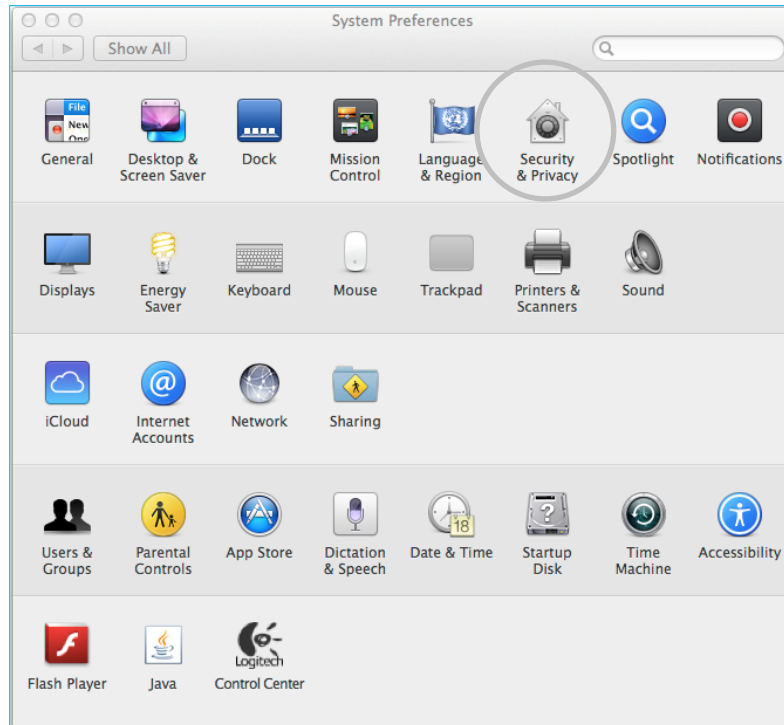
Figure 4
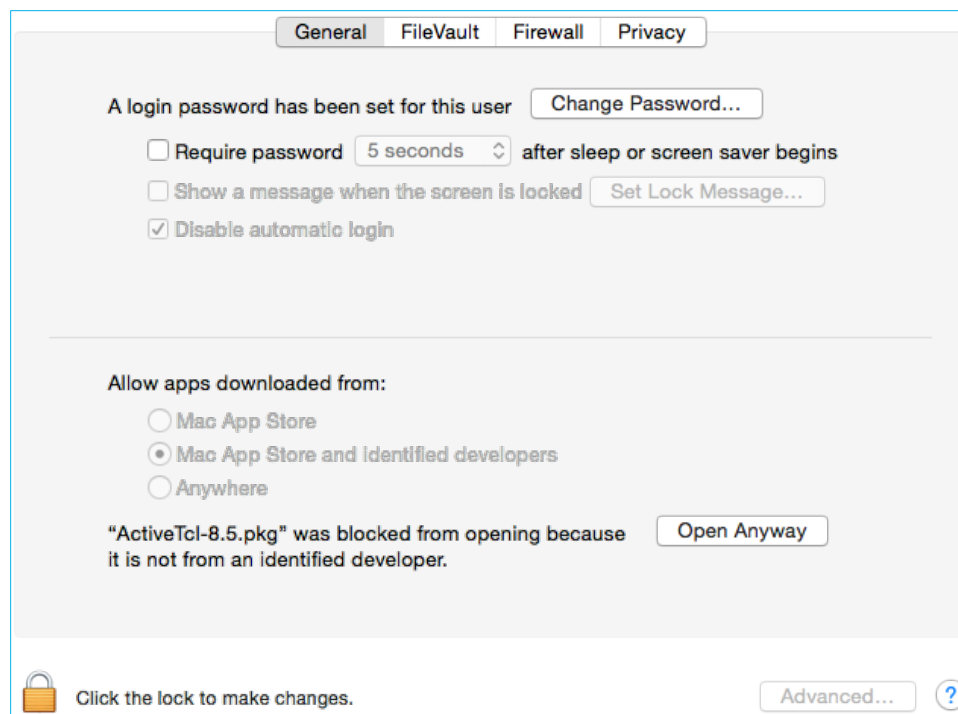
Figure 5

Notice near the bottom of this panel the message that *"ActiveTcl-8.5.pkg" was block from opening because it is not from an identified developer*. Click on *Open Anyway* to proceed with the installation. This will involve several more dialog boxes, including an agreement to the licensing terms. You may need to enter the user ID and password of the Administrator of your Mac-

intosh. You may also have to click on the lock icon at the lower left corner of the window before you can make the change.

The installation of *ActiveTcl* should now proceed to completion and display with the screen in Figure 6. It will finish with a window entitled *ActiveTCL User Guide*. For CS-1004, it is safe to ignore this user guide.
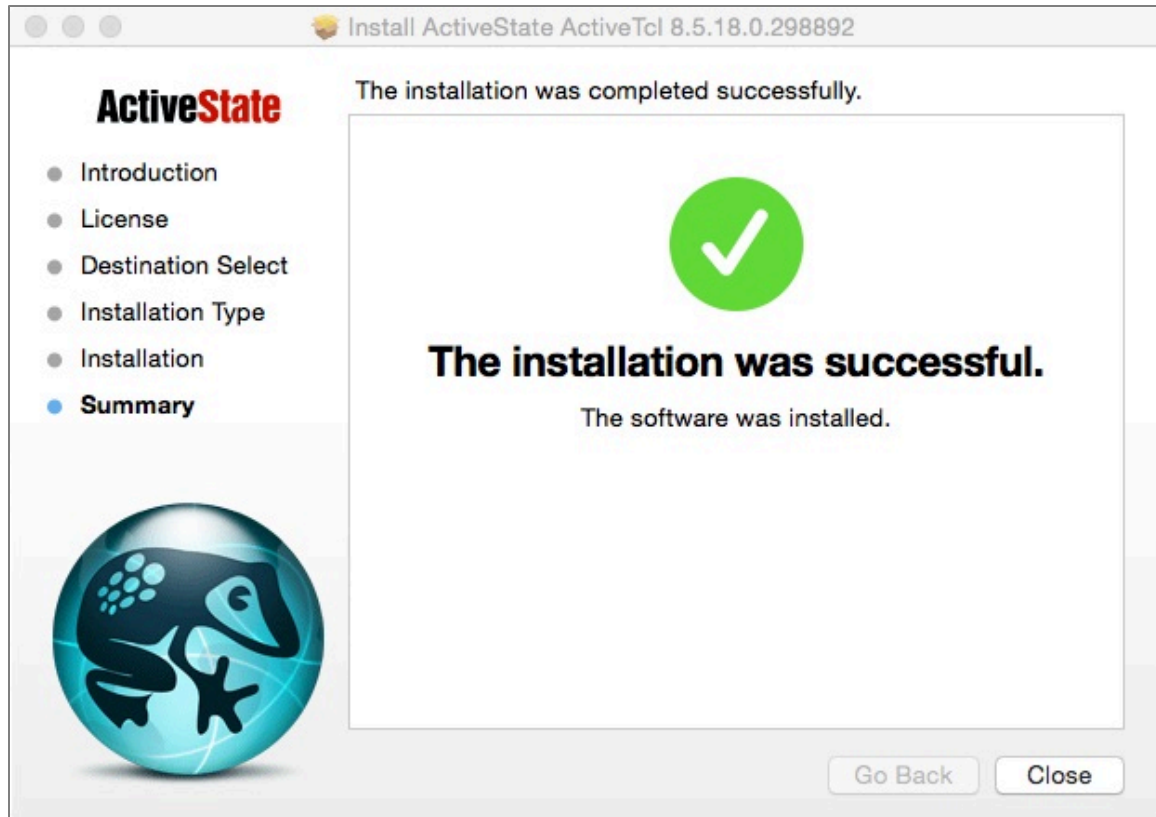
Figure 6

Installing Python 3.5

Next, you will install *Python 3.5.1*. Open the **python-3.5.1-macosx10.6.pkg** file that you downloaded on Page 2 above. This should bring up a dialog similar to Figure 6 below.

Figure 7

The message in the dialog box includes information about specific versions of *Tcl/Tk*. We have already addressed this issue with the installation of *ActiveTcl 8.5.18.0*.

If the installer complains that there is already a version of *Python 3* installed on your computer, please seek help or contact the Professor. You must have *Python 3.5.1* for this course. Earlier versions should be upgraded. Click *Continue* to bring up another dialog:–
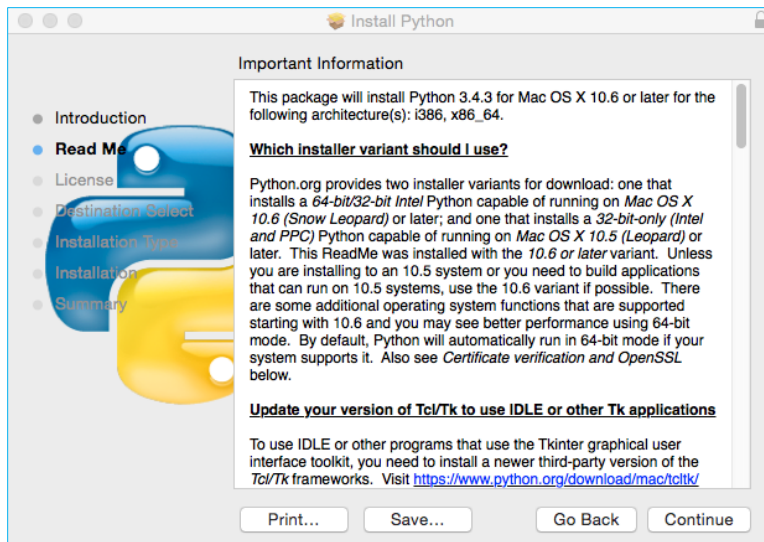
Figure 8

This dialog box is scrollable and contains information about what is new in this release. It also has another reminder to update *Tcl/Tk*, which we just did in Figure 2 and Figure 5. Click *Continue* to bring up additional dialog boxes until the one in Figure 9 appears.

6
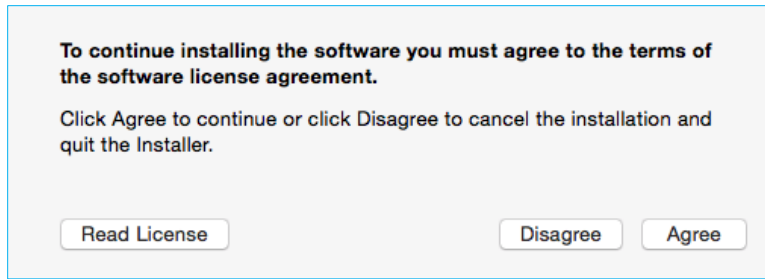
Figure 9

Click *Agree* to continue the installation. The installation itself will take several minutes and may show one or more dialog boxes. It will finish with a panel resembling Figure 10.
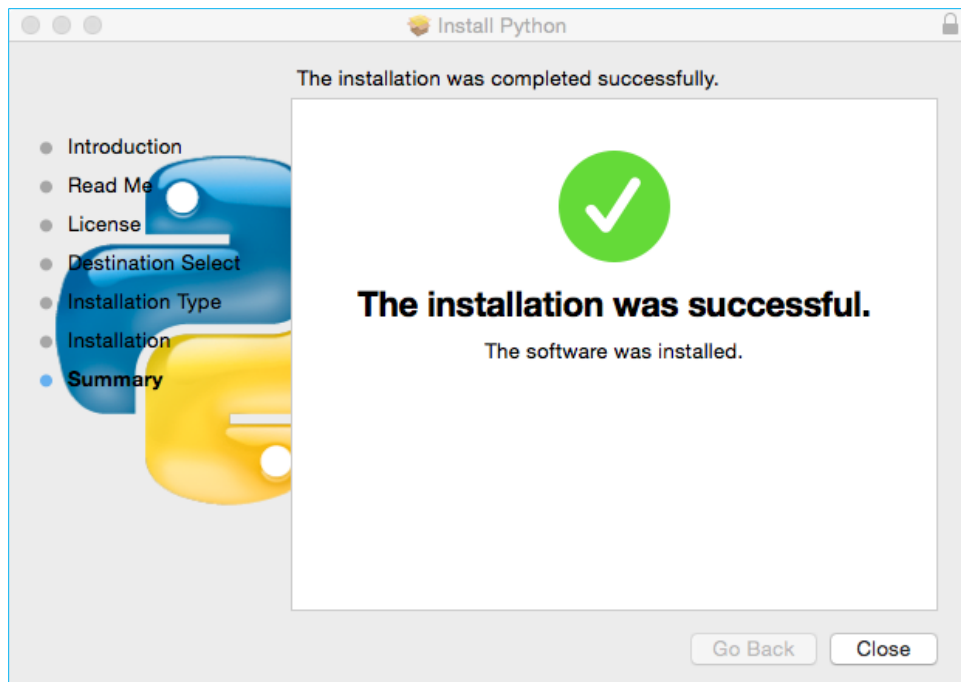
Figure 10

After it has completed, you should find a folder named *Python 3.5* in your *Applications* folder, as shown circled in solid black below.
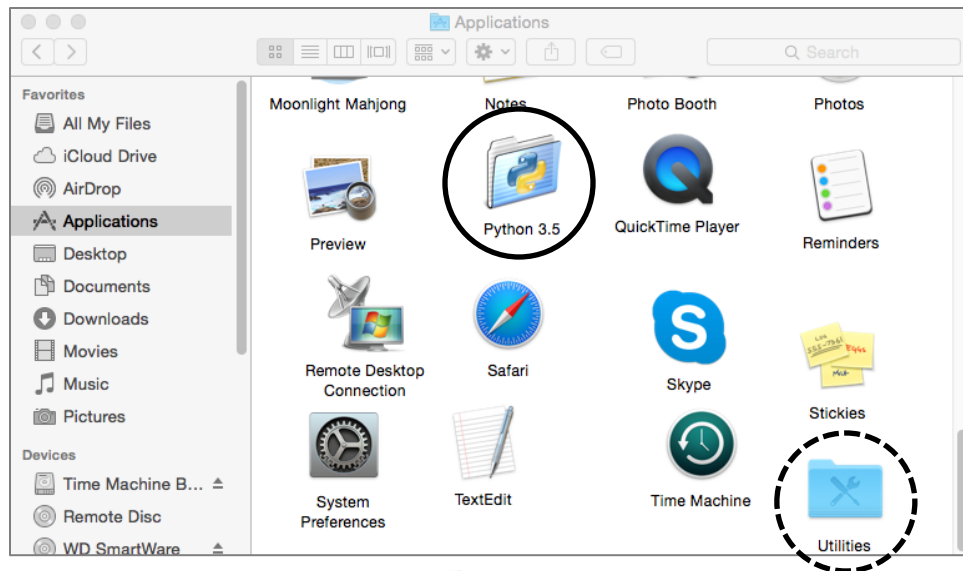
7

Figure 11

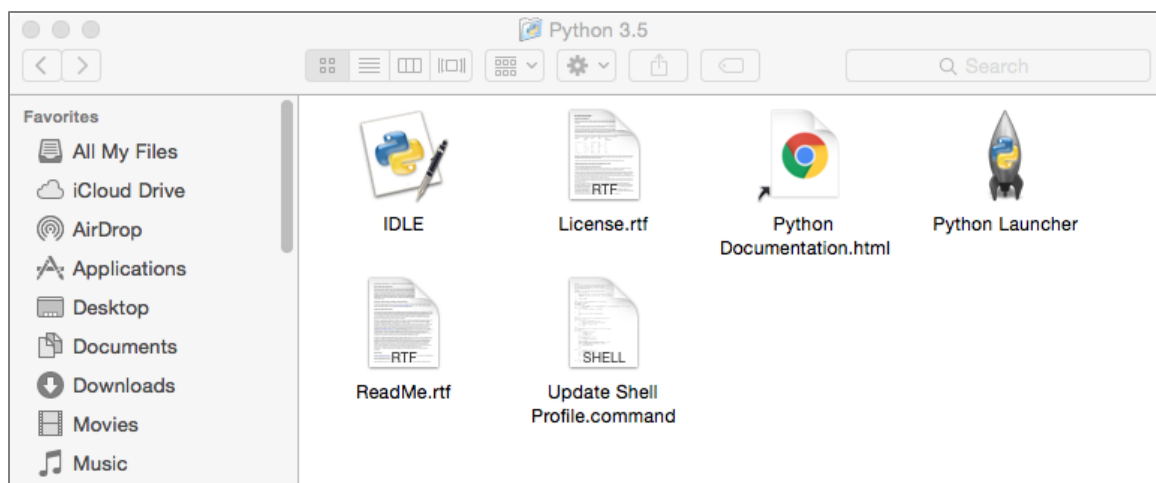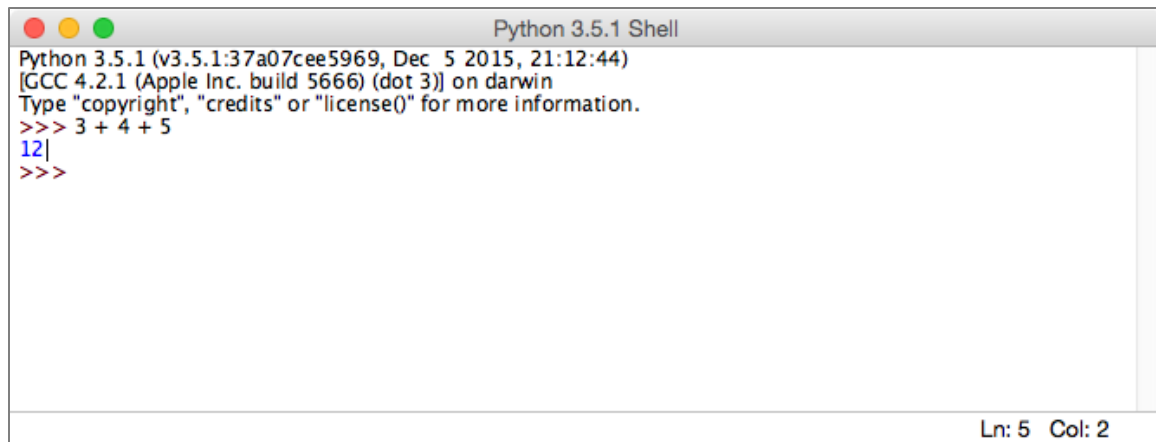Open the *Python 3.5* folder to show Figure 12 below.

Figure 12

Create an *Alias* for *IDLE* and put it in a convenient place so that you can access it easily — for example, on your desktop. Double-click this alias to bring up the following window:–

Figure 13

This is *IDLE*, the *Python* command prompt and graphical user interface. *This is where we will start all programs and projects in this course.*

For now, simply type any *Python* statement or expression after the ">>>" prompt. For example, Figure 13 shows the expression *3 + 4 + 5* as typed, followed by *Python's* response with the value *12*. Continue testing by typing out the code on pages 10-11 of the textbook, just to make sure that your installation works as expected.

Note: IDLE is essentially the same across *Windows*, *Macintosh*, and *Linux* platforms. It will be used the same way in all three for this course.

Installing *matplotlib*, *numpy*, and other packages

One of the many benefits of *Python* is the vast number of third-party packages that can be downloaded and used by your *Python* programs. Many of these are open-source and free. For this course, we will use at least the following:–

- *matplotlib* (a package for creating 2D plots and graphs similar to *Matlab*),

- *numpy* (meaning "Numerical Python," a package for efficient handling of large arrays of numerical data), and

- *graphics.py*, a simple tool written in *Python 3* and created by the textbook author for making simple drawings.

Installing *matplotlib*, *numpy*, and other packages on a Macintosh requires you to type Linux-like commands in a *Terminal* window.

Open the *Utilities* folder, which is indicated by the dashed circle in Figure 11. This will bring up the window of Figure 14. Select *Terminal* (highlighted with a red circle) and open it.

9

Figure 14

This will bring up a terminal window (also known as a *command shell* in Unix and Linux terminology). An example terminal window is shown in Figure 15. In this window, in this window, the system prints a *prompt* starting at the beginning of a line and ending in '$'. After the prompt, you type a *command*, consisting of a command name followed by zero or more *operands*, which control what the command does. When you terminate the command with the *Enter* key, the system performs the command.

A *command* may work silently and then type out its results in the same window, or it may engage in a textual conversation with you, requiring you to respond, or it may open its own window with its own graphical user interface. When the command has completed, the system prints a new prompt for the next command.

Installing *numpy* and *matplotlib*
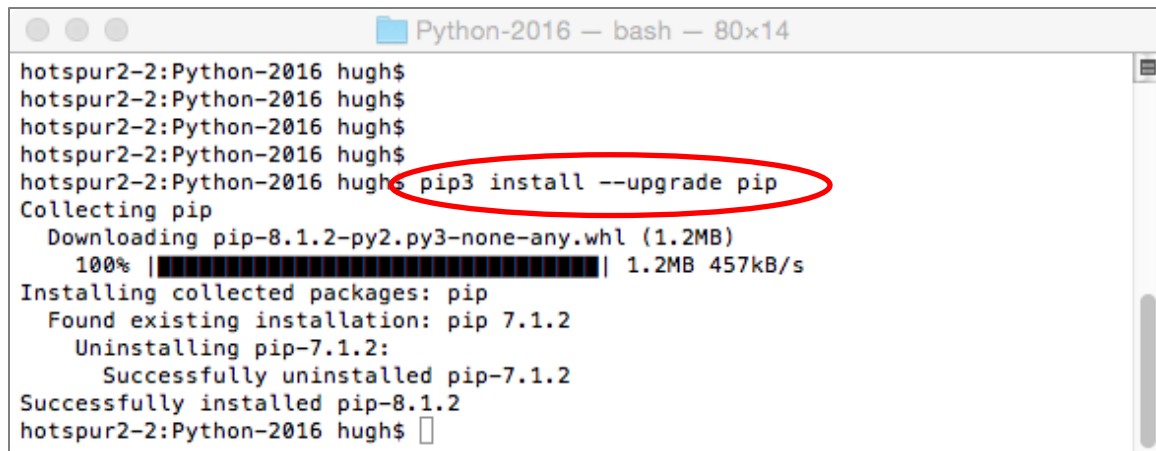
> This step requires you to have a working internet connection.

Upgrading pip

To install *numpy* and *matplotlib*, we will use a command named **pip3** — "pip" means *Python Installation Program*, and "3" refers to the *Python 3.x* versions of *Python*.

In Figure 15, we use the **pip3** command to upgrade itself.[5]

---

[5] It is not strictly necessary to upgrade **pip**; however, if you don't, you will get prompts from **pip** complaining that it is out of date on all subsequent commands.
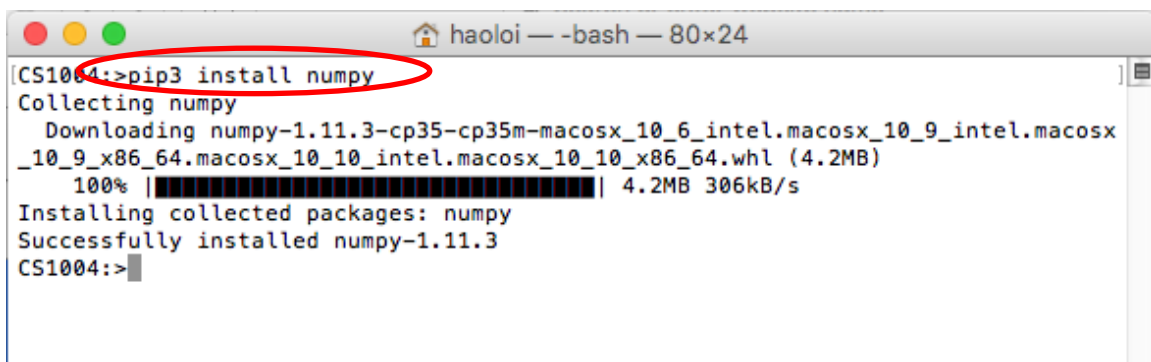
Figure 15

numpy

To install *numpy*, issue the following command at your terminal,

```
pip3 install numpy
```

This command will install *numpy* as shown in Figure 16 below. If necessary, it will resolve any dependencies on other packages. In this case, there were none.

Figure 17

You can test your installation of *numpy* by opening an *IDLE* window, as in Figure 12. Type or paste the following commands into IDLE, one line at a time, *exactly* as written.

> **Note:** In these lines, the word **version** is preceded by *two* underscore characters and followed by two more underscore characters.
>
> The word **arange** is spelled with one "r" — it is a variation of the range function that we will learn about in the first week of class.
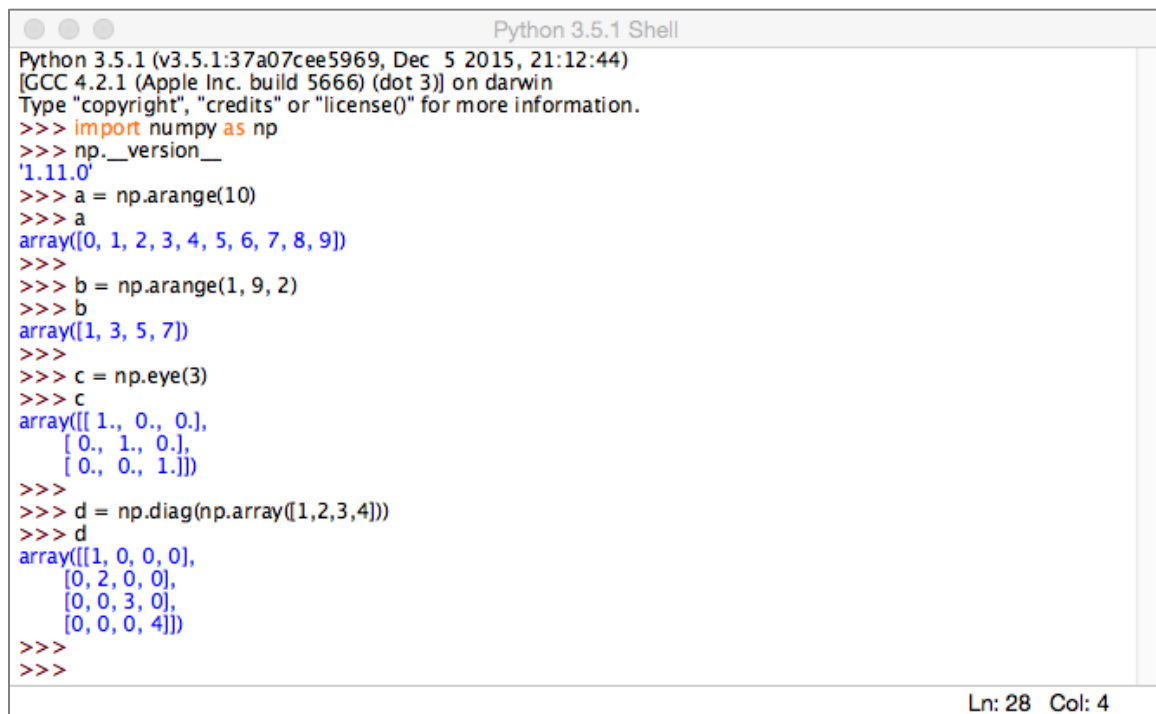
```
import numpy as np

np.__version__

a = np.arange(10)
a

b = np.arange(1, 9, 2)
b

c = np.eye(3)
c

d = np.diag(np.array([1, 2, 3, 4]))
d
```

After each line **a**, **b**, **c**, and **d**, *Python* will print the values of these variables. The result should resemble Figure 18 below:–
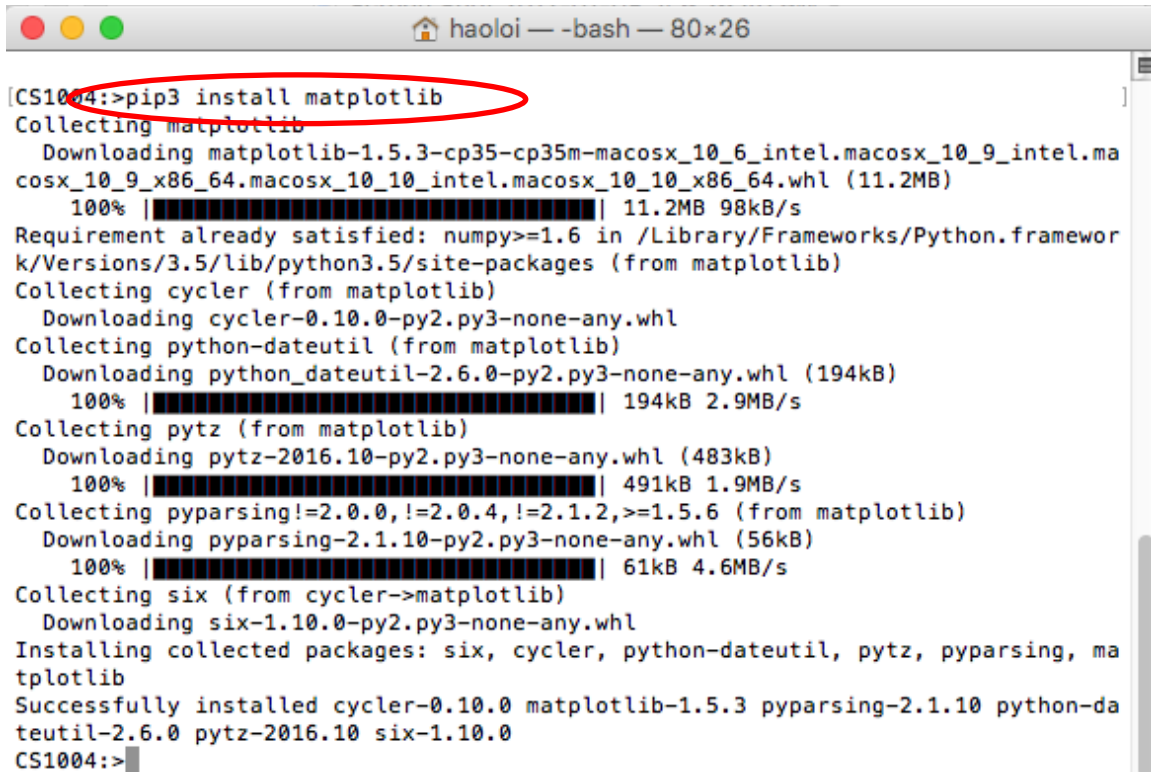

Figure 28

Congratulations! You have successfully installed and tested *numpy 1.11.0*. However, we will run another test below.

matplotlib

Installation of *matplotlib* is done the same way. Type the command into the *Terminal* window, all on one line:–
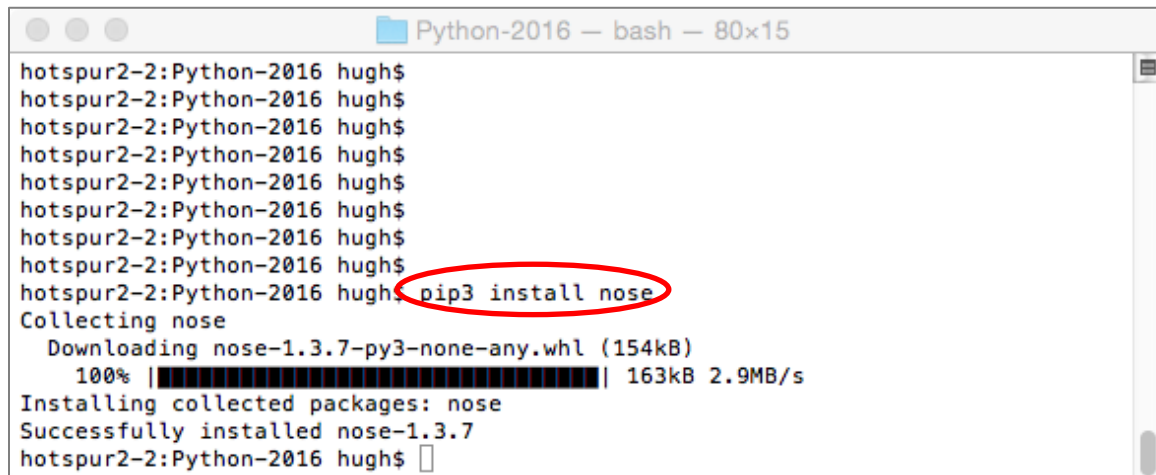
```
pip3 install matplotlib
```

This is shown in Figure 19 below.

12

Figure 19

In addition, you will need one other package — **nose** — that will help test your installation. The installation of **nose** is shown in Figure 20.

Figure 20

To test your *matplotlib* installation, type or paste the following commands into IDLE, *one line at a time*, exactly as written:–

```
from matplotlib import pyplot
pyplot.plot([1, 2, 3, 4], [1, 4, 9, 16])
pyplot.show()
```

You may get some warning messages for the first time. You can just ignore the warning messages. The IDLE window should look something like the following:–


```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  5 2015, 21:12:44)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> from matplotlib import pyplot
>>> pyplot.plot([1, 2, 3, 4],[1, 4, 9, 16])
[<matplotlib.lines.Line2D object at 0x10a83e7f0>]
>>> pyplot.show()
```

Figure 21

After you type the **ENTER** key following the last line, the following window should appear:–
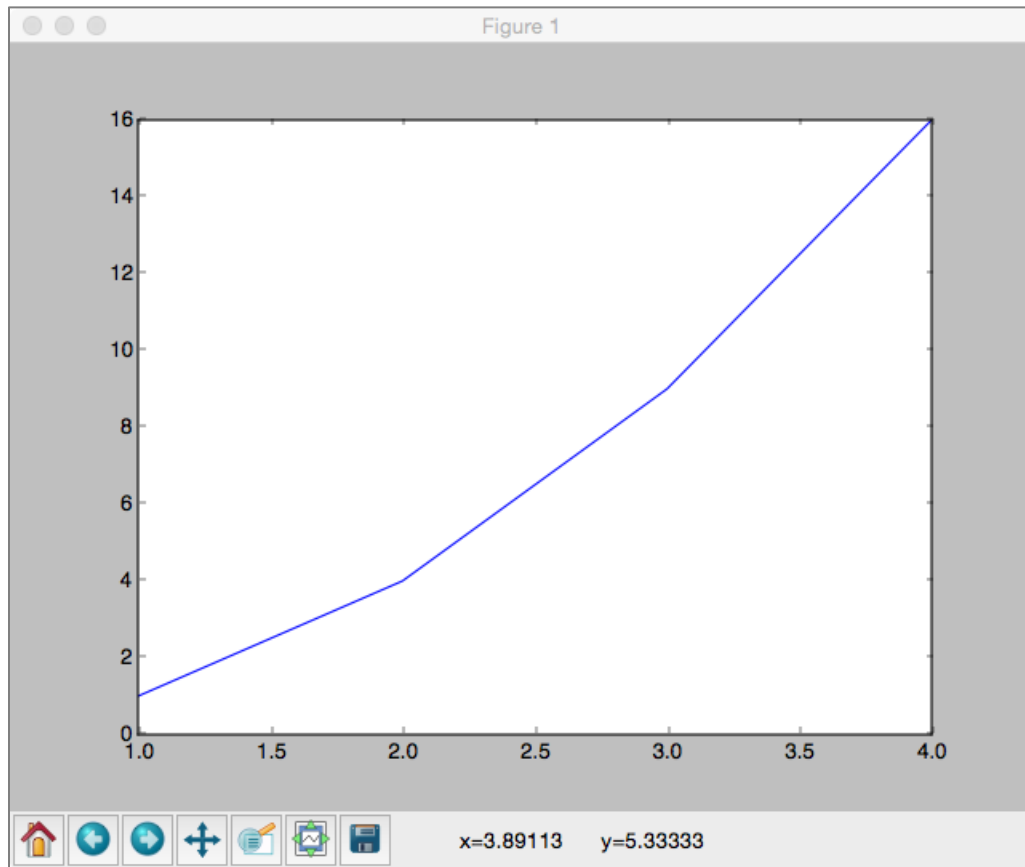

Figure 22

To close this window, click on the Macintosh "close" button in the upper left corner.

For a more interesting test, download the following file to a convenient folder and save it as a **.py** file:–

TestMatplotlib2.py

Then, use the *File* menu in the *IDLE* window to open this file in its own window, as shown in Figure 23:–

```
#  TestMatplotlib2.py

from pylab import *

def testPyplot(low, high, incr):
    t = arange(low, high, incr)
    s = sin(2*pi*t)
    plot(t, s)
    xlabel('time (s)')
    ylabel('voltage (mV)')
    title('About as simple as it gets, folks')
    grid(True)
    savefig("test.png")
    show()
    return
if __name__ == '__main__':
    testPyplot(-1.0, 1.0, 0.001)
```

Figure 23

Click on this window to make it the active window, and then select the the *Run > Run Module* command in the IDLE menu at the top of the Macintosh screen. This will cause the *Python* program to run and to produce the window of *figure 24*.

This tests not only *matplotlib* but also *numpy* (in the background). The result should be a window like this:–
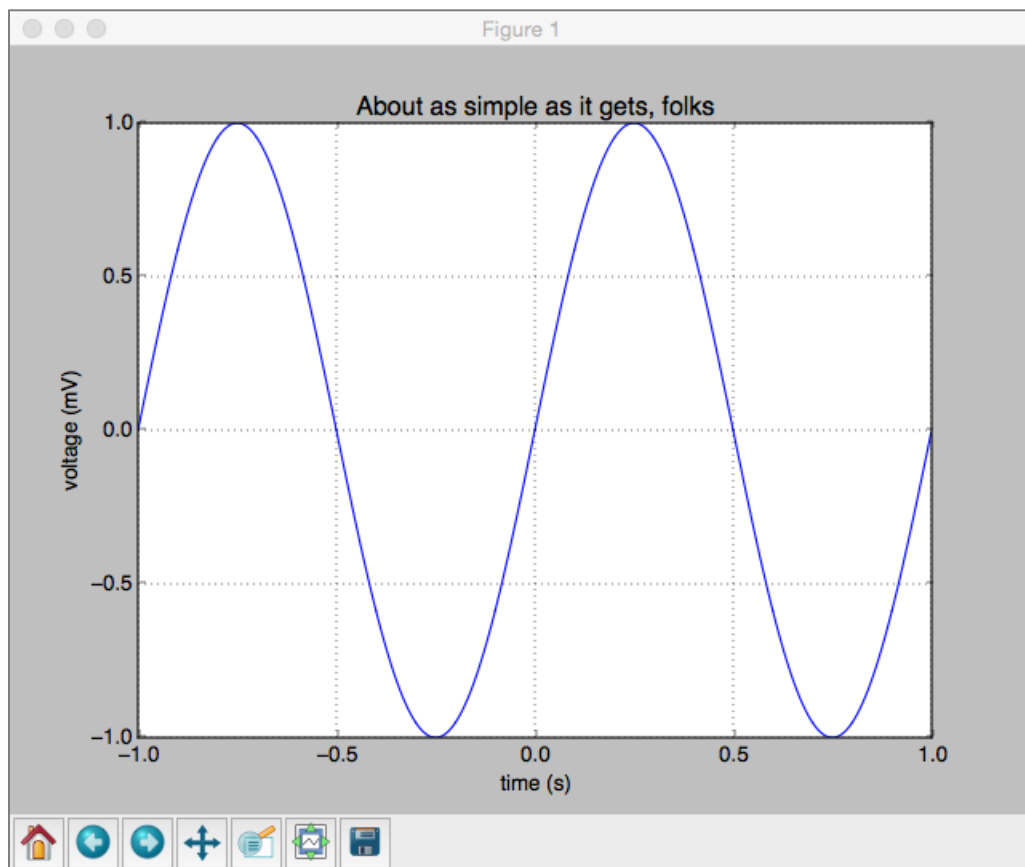

Figure 24

Congratulations! You now have a working version of *matplotlib* installed. As part of this test, a copy of the picture was saved under the name **test.png** in the current directory.
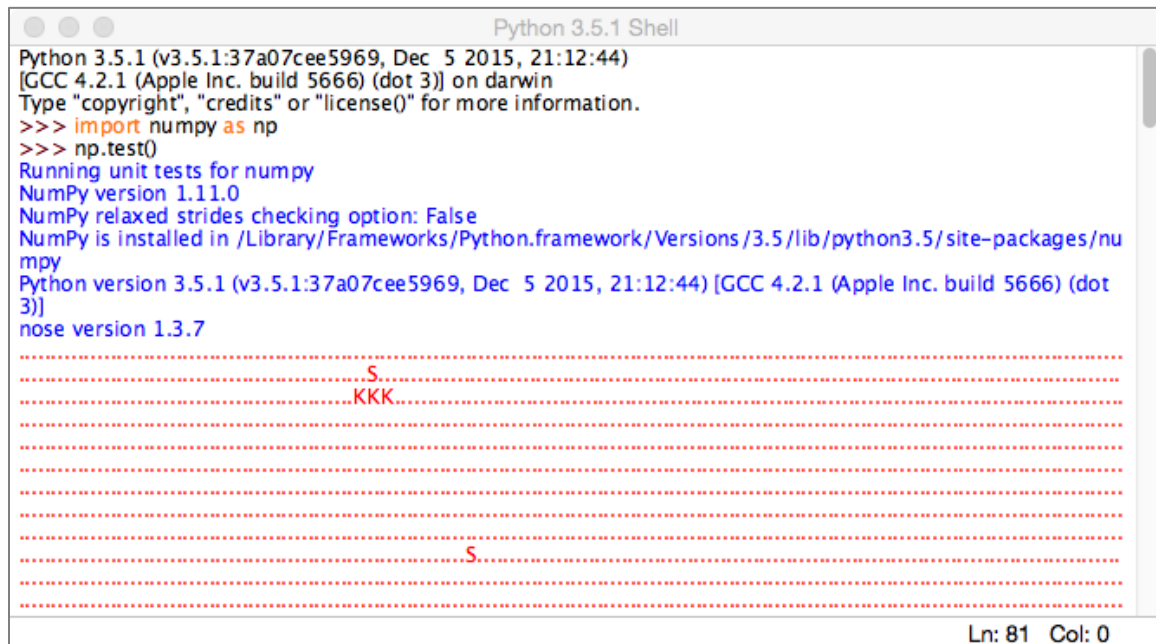
Note also that *matplotlib* used *numpy* in the background to build up the array of points making up the graph.

For a final test, type the following into an *IDLE* window:–

```
import numpy as np
np.test()
```

This will run a complete set of tests of the *numpy* package for several minutes, producing a lot of output. The output begins with the lines in Figure 25 and continues for some time. At times, it may seem like it has paused or stopped. Please be patient.

> On some systems, it may ask to install or update gcc (the compiler for the C programming language). If it asks, say "no" or "later."

Figure 25

Eventually, it will finish with some "apparent" errors as shown in Figure 26 below. You may ignore these errors.

```
..................................................................................
..................................................................................
..................................................................................
....................F
==================================================================================
FAIL: test_scripts.test_f2py
----------------------------------------------------------------------------------
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-pac
kages/nose/case.py", line 198, in runTest
    self.test(*self.arg)
  File "/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-pac
kages/numpy/testing/decorators.py", line 147, in skipper_func
    return f(*args, **kwargs)
  File "/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-pac
kages/numpy/tests/test_scripts.py", line 99, in test_f2py
    assert_(success, msg)
  File "/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-pac
kages/numpy/testing/utils.py", line 77, in assert_
    raise AssertionError(smsg)
AssertionError: Warning: neither f2py nor f2py3 nor f2py3.5 found in path

----------------------------------------------------------------------------------
Ran 5780 tests in 615.727s

FAILED (KNOWNFAIL=7, SKIP=10, failures=1)
<nose.result.TextTestResult run=5780 errors=0 failures=1>
>>>
```

Figure 26

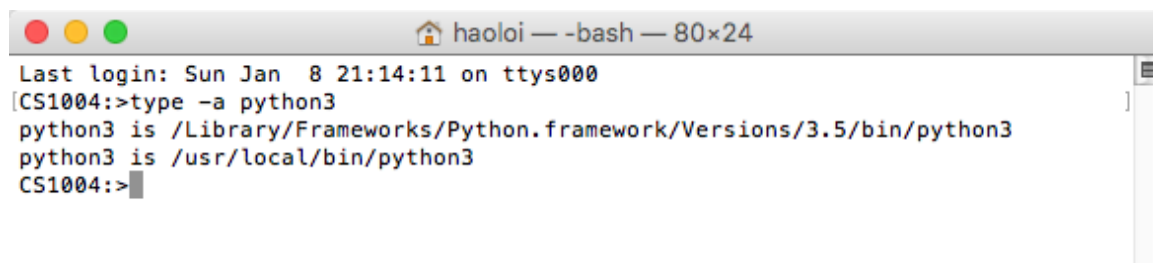The last line shows that 5780 tests were run with seven errors and one failure. This is considered success.

Congratulations! You now have both *matplotlib 1.5.1* and *numpy 1.11.0* working.

Cleaning up

Note that **dmg** files in Apple parlance are really virtual disk images. Opening the files to install *ActiveTcl 8.5.18.0* leaves behind a disk icon on your desktop. You can remove these it right-clicking and selecting *Eject*.

Installing Graphics.py

To install *graphics.py*, click on this link — graphics.py — and download the file to the folder where you keep your *Python* programs. You can type in the following command on the terminal to find out where your Python is installed on your Mac laptop.

```
● ● ●                    ⌂ haoloi — -bash — 80×24
Last login: Sun Jan  8 21:14:11 on ttys000
[CS1004:>type -a python3
python3 is /Library/Frameworks/Python.framework/Versions/3.5/bin/python3
python3 is /usr/local/bin/python3
CS1004:>
```

Figure 27

Save your graphics.py to the following directory:

/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5
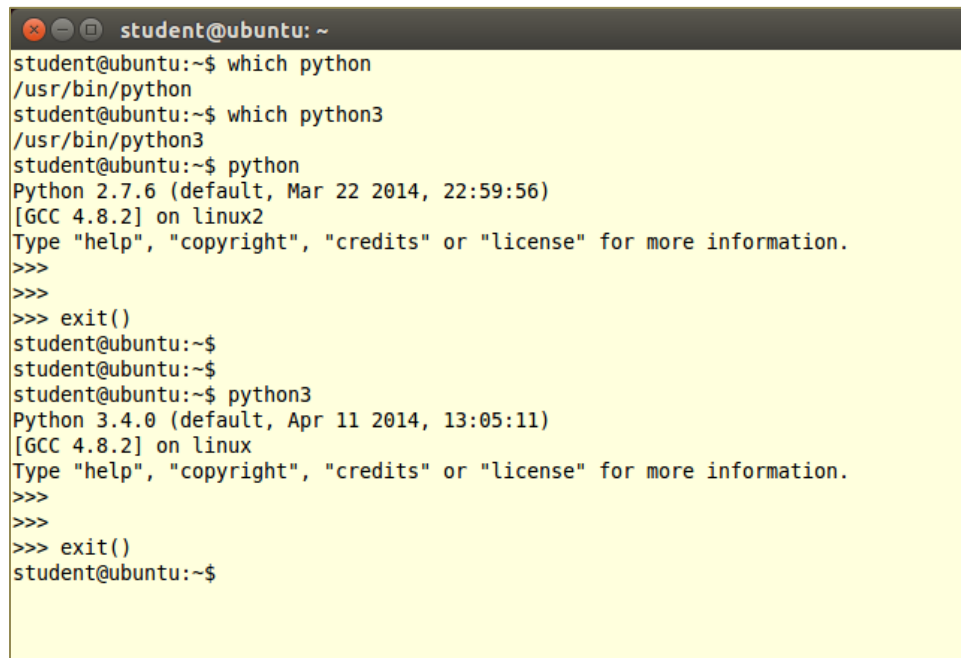
Installing Python 3 on Linux

Linux comes in many flavors supported by different organizations with different approaches to software installation. The desktop/laptop versions of Linux typically package the most common applications — including *Python* — in their automated download and installation systems. The following describes the installation on *Ubuntu 14.04*, a popular Linux version in use on the WPI campus, also known by the code name *Trusty Tahr*.[6, 7]

To determine whether or not *Python* is installed in your Linux (for any version or distribution of Linux), open a *Terminal Window* (also called a *Command Shell*) and type the following two commands (on separate lines):–

```
which python
```

```
which python3
```

If either returns the name of an executable file, then execute it to start that version of *Python*. The version number should be the first thing that *Python* says, as illustrated in the following figure for **python** (line 5) and **python3** (line 15):–

Figure 3

The first two commands (lines 1 and 3) ask which file contains the programs *python* and *python3*, respectively. In this case, both programs are present in the indicated files. If either had not been present, the command would have reported nothing.

---

[6]   It is not clear whether up-to-date versions of *Python* are available for download on older releases of *Ubuntu* or other versions of Linux. If you have an older Linux and are not able to upgrade, please see the Professor or Teaching Assistants for advice.

[7]   The WPI Computer Science Department maintains a mirror distribution site for Ubuntu releases at http://ubuntu.cs.wpi.edu.

In line 5, the *python* program is run. Its first action is to print a preamble announcing its version and other (possibly useful) information; in this case, it is *Python 2.7.6*. The *exit()* command following the traditional *Python* prompt of ">>>" merely tells *Python 2.7.6* to finish what it is doing and to quit.

In line 14, the *python3* program is run. You can see from its preamble that it is *Python 3.4.0*.[8]

> **Note:** Both Mac OS X and Linux come with Python 2.7 already installed for their own needs. That version of Python must not be deleted or disturbed when installing Python 3.4 and its packages.

Installing IDLE

Figure 26 applies to any version Linux from any distributor to determine whether or not *Python* is installed and, if so, what version it is.

The following is specific to *Ubuntu* Linux. Users of other Linux distributions should carry out similar actions on those distributions. Select and open the *Ubuntu Software Center* icon

in the toolbar on the left of the *Ubuntu* desktop. In the upper right, enter the term "Python3" in the search box. A search should produce the results shown in the figure below.
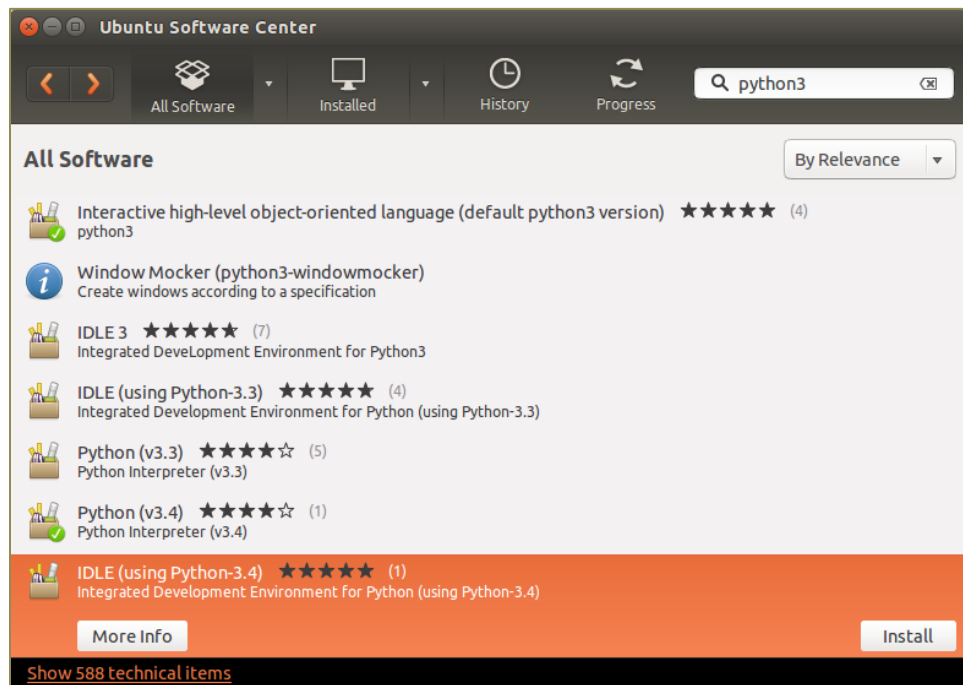

Figure 4

This shows that the *Ubuntu Software Center* knows about several versions of *Python 3* and several versions of *IDLE*. The last line is highlighted and shows an *Install* button for "IDLE using Python 3.4). Click on this *Install* button to cause IDLE to be downloaded installed.

---

[8]   This is one minor release older than the versions that we have installed on Windows or the Macintosh. However, we will live with that difference for CS-1004.

When the installation has completed, a new icon will have been added to the toolbar on the left, namely, the IDLE icon . *IDLE* has now been installed.

Testing your Linux installation of Python 3.4

Follow the same instructions as with Figure 12 in order to validate that you can run *Python*.

Installing *numpy* on Linux

In *Ubuntu 14.04*, open the *Ubuntu Software Center* again and search for the term "python3-numpy". This will bring up one or more results, including the *numpy* package for *Python 3*.

> **Note:** Be sure that you specified "**python3**" as part of the search term. Otherwise, the search will return information about *numpy* for *Python 2.7*.
>
> **Note2:** If you use a different version of Linux, the search term in your software installation system should still be "**python3-numpy**."
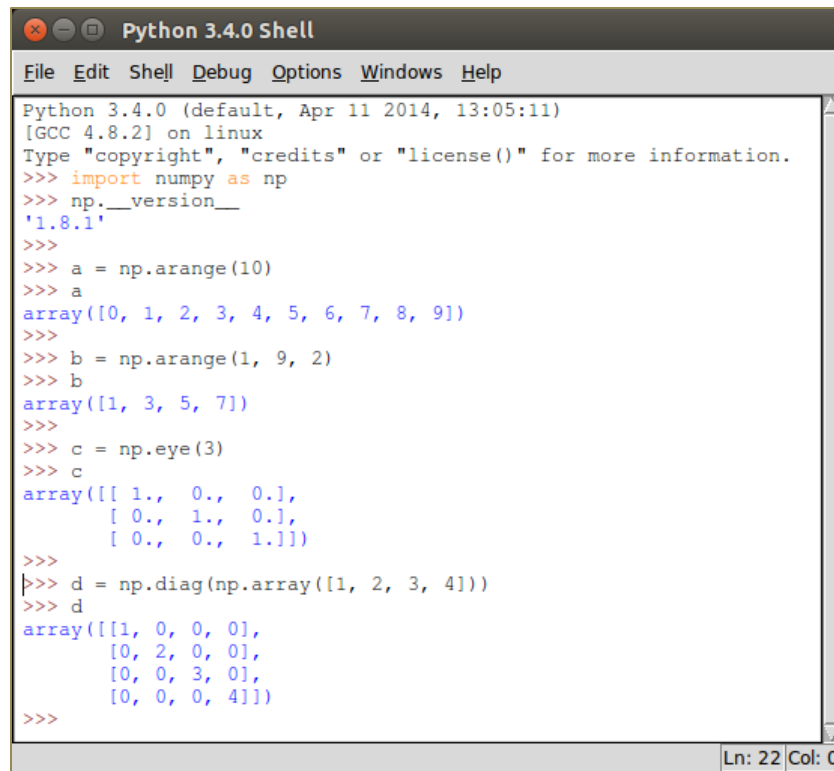
Select on the *More Info* button to bring up the following window:–

Figure 5

Although it does not specify which version of *numpy* is provided, click on the *Install* button to install it. It will automatically install any required dependencies. You do not need to select other options for this course.

Finally, open an *IDLE* window by clicking on the *IDLE* icon and carry out the tests of Figure 15. Results of these tests are shown in Figure 29 below. The version of *numpy* is 1.8.1, as indicated by line 5 of this window, but your version may be different
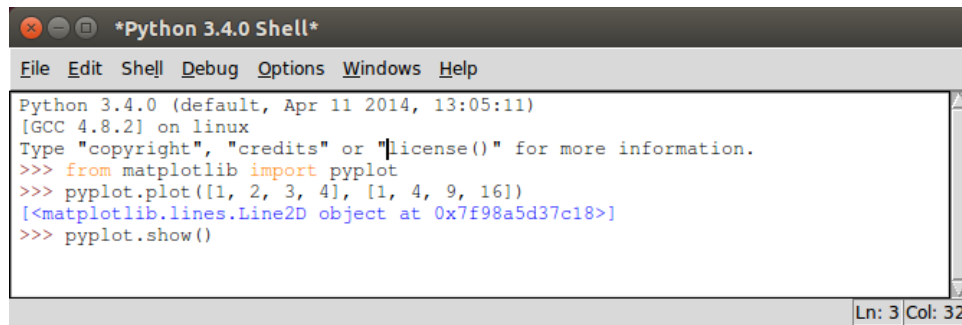

Figure 6

Installing *matplotlib* on Linux

To install *matplotlib* on your version of Linux, repeat the same steps as you used for *numpy* but search instead for "**python3-matplotlib**." In *Ubuntu* , you would see the following:–

Figure 7

Install this, just as you installed *numpy*. To test your installation, carry out the tests of Figure 20 and of Figure 22. This should produce the *IDLE* window of Figure 31 plots shown in those figures

Figure 8

Conclusion

Congratuations! You have successfully used *Python 3.5.1* as already installed and have added *IDLE*, *numpy*, and *matplotlib*.

If you are brave enough (and savvy enough) to use another distribution of *Linux*, good luck. The guidance here should help you get started. The official *Python* download page is at

<div align="center">

https://www.python.org/download/

</div>

Source "tarballs" are available to compile for your own environment. Alternatively, a list of other *Python* distributors is also provided. Good luck!