



CS-1004, Introduction to Program-
ming for Non-Majors, A-Term 2015

Setting up Python 3.4, 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 3.4 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 3.4* and packages such as *numpy* and *matplotlib* on Macintosh platforms. 1616 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 versions of *Python* in general use around the world — *Python 2.7* and *Python 3.4*. 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 3.4* and its packages.

[©] Copyright 2015, Hugh C. Lauer. All rights reserved. Permission is given for use in courses at Worcester Polytechnic Institute, Worcester, Massachusetts.

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

Installing Python 3.4 on Macintosh^{2, 3}

Installing *Python 3.4* on Mac OS X presents two challenges:—

- Packages such as *numpy 1.9.2* and *matplotlib 1.4.3* are delivered in forms that require installation via commands in a *Terminal* window for installation.
- The *IDLE* tool of *Python 3.4* has a dependency on *Tcl/Tk*, an open-source toolkit for building user interfaces. However, *Python 3.4* requires a more recent version than is already installed on Mac OS X.

First, create a folder on your desktop — for example, *Python-files* — and download the following two installers to that by clicking on these links:— [ActiveTcl 8.5.18.0](#), and [python-3.4.3-macosx10.6](#). Alternatively, you may browse to

<http://www.cs.wpi.edu/~cs1004/a15/Resources/>

and download the files from there.

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:—

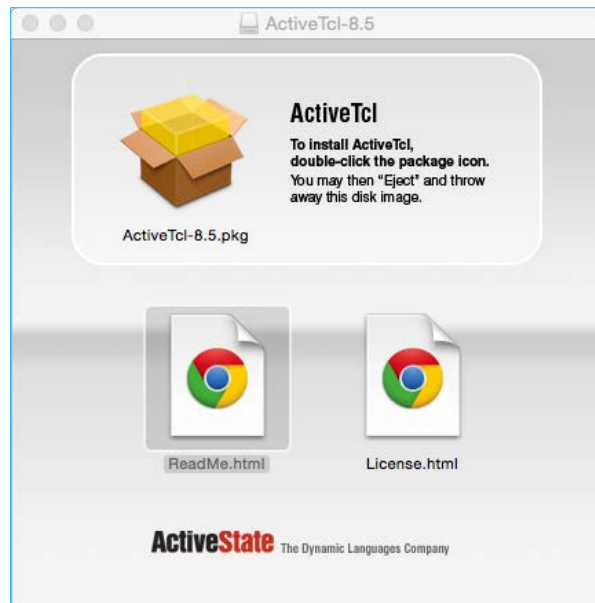


Figure 1

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.

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

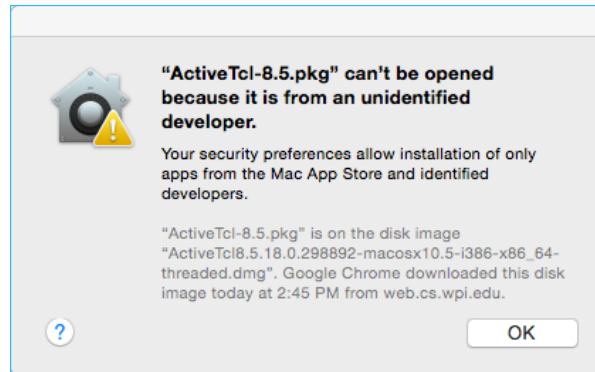


Figure 2

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 3, to bring up the *Security & Privacy* control panel of Figure 4.

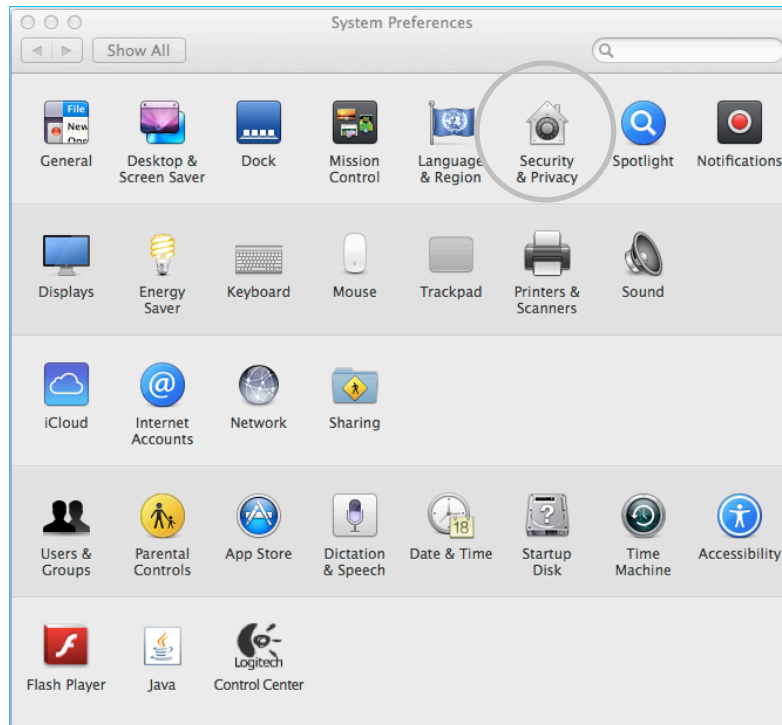


Figure 3

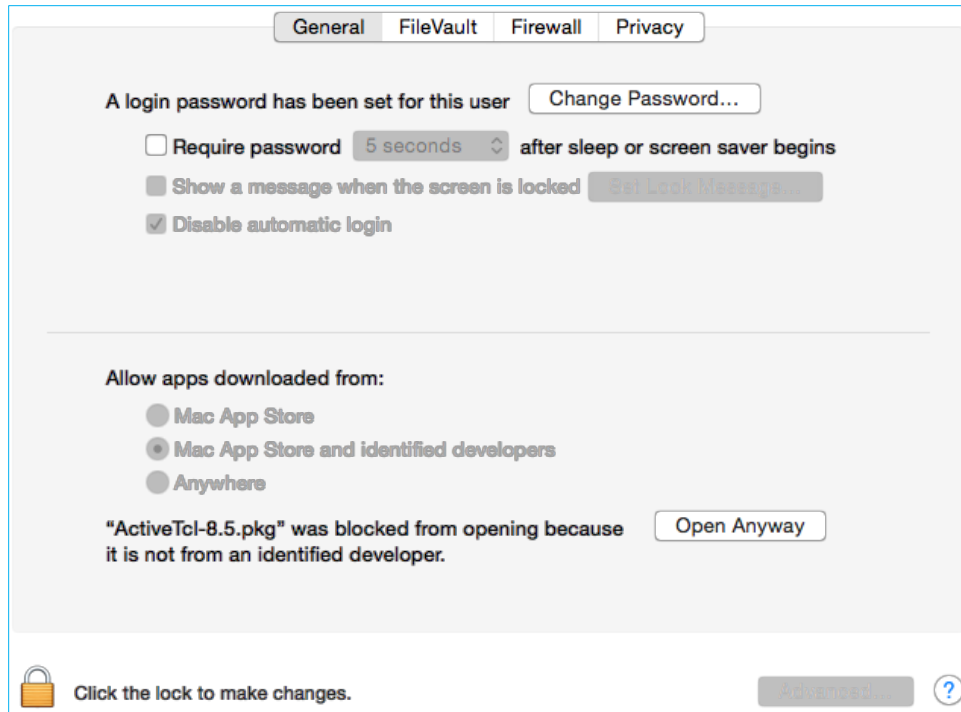


Figure 4

Notice near the bottom of this panel the message that *"ActiveTcl-8.5.pkg" was blocked 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. It will also require you to enter the user ID and password of the Administrator of your Macintosh.

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

Installing Python 3.4

Next, you will install *Python 3.4*. Open the **python-3.4.3-macosx10.6.pkg** file that you downloaded on Page 2 above. This should bring up a dialog resembling Figure 5 below.



Figure 5

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 3.4.3* for this course. Earlier versions should be upgraded. Click *Continue* to bring up another dialog:–

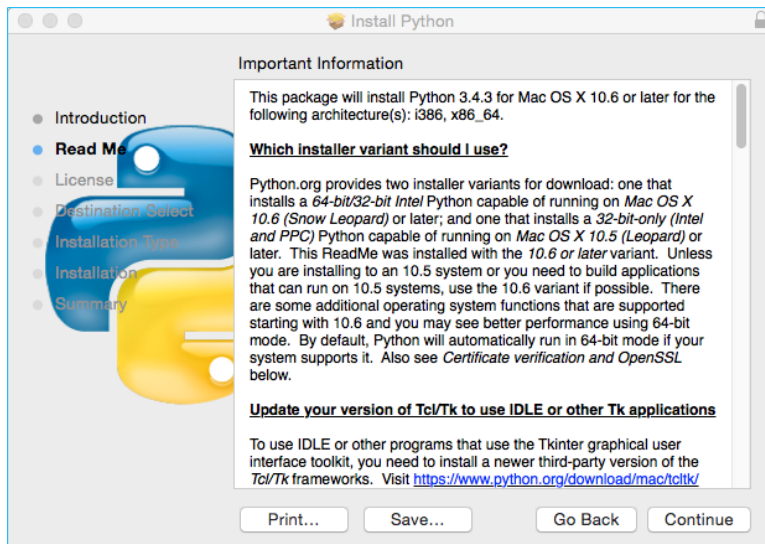


Figure 6

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 1 and Figure 4. Click *Continue* to bring up additional dialog boxes until the one in Figure 7 appears.

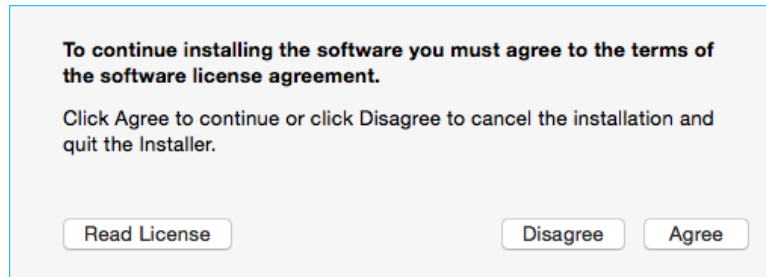


Figure 7

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 8.

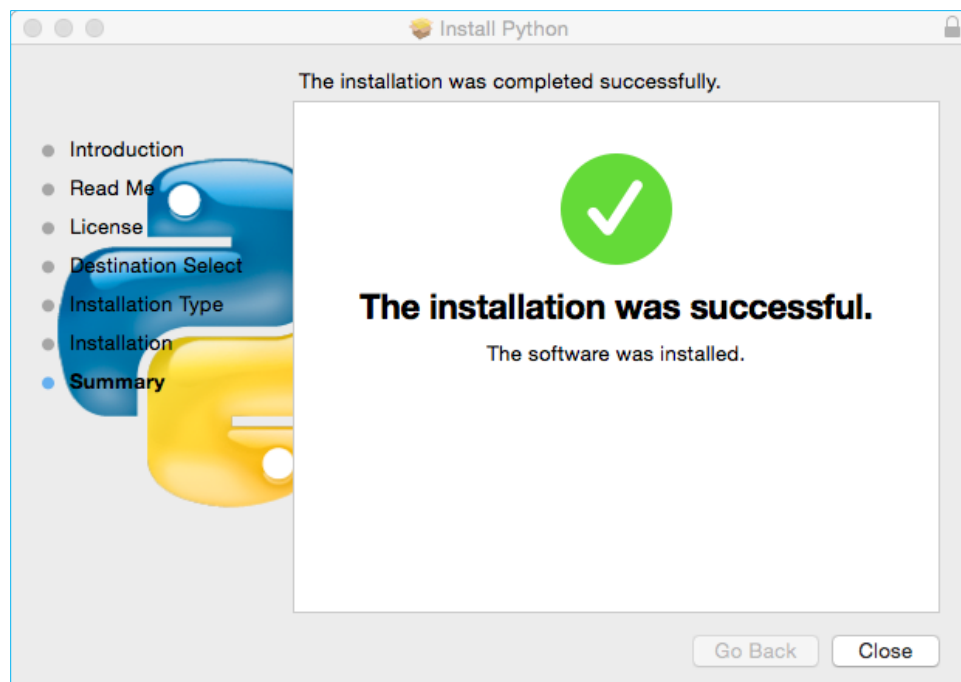


Figure 8

After it has completed, you should find a folder named *Python 3.4* in your *Applications* folder, as shown circled in solid black below.

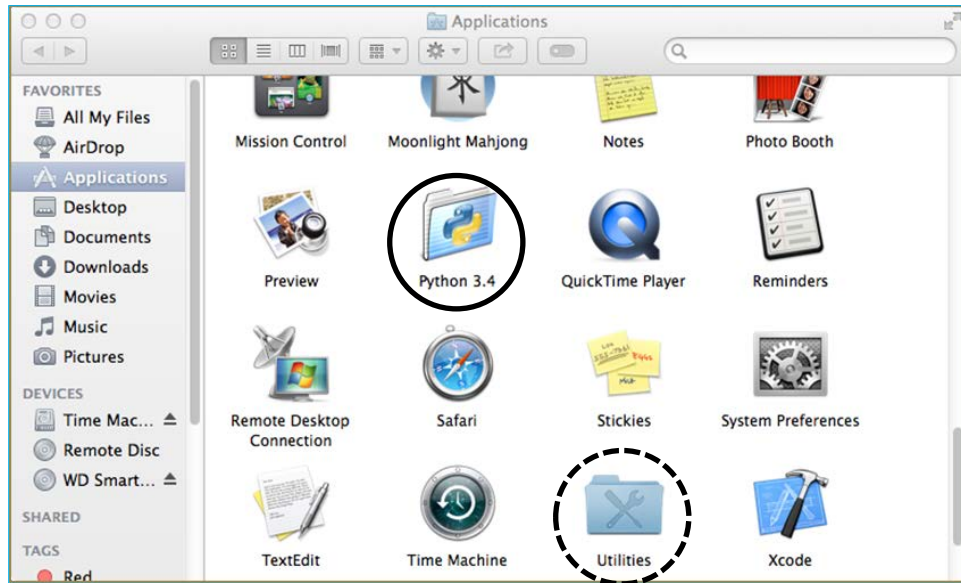


Figure 9

Open the *Python 3.4* folder to show Figure 10 below.

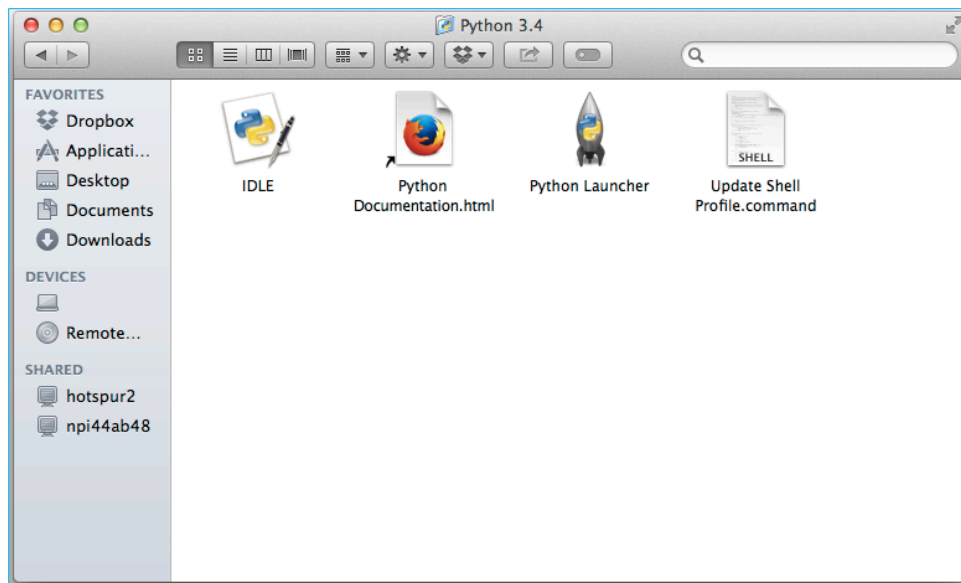
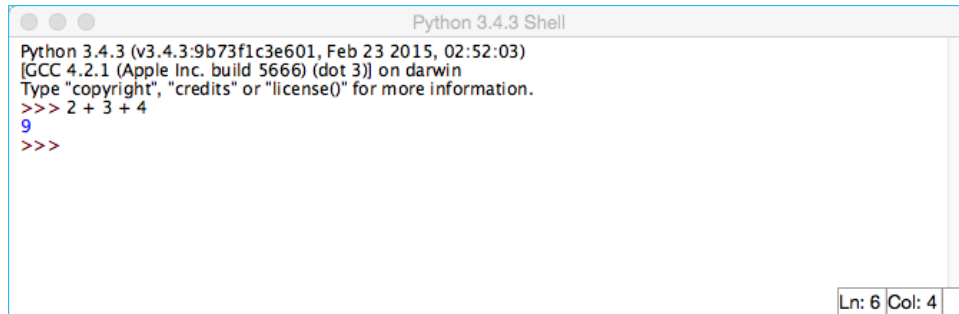


Figure 10

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:—

A screenshot of a Python 3.4.3 Shell window. The title bar says "Python 3.4.3 Shell". The text inside the window reads: "Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 23 2015, 02:52:03) [GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin Type 'copyright', 'credits' or 'license()' for more information." Below this, the prompt ">>>" is followed by the expression "2 + 3 + 4", which is followed by the response "9". Another prompt ">>>" is shown on the next line. In the bottom right corner, there is a status bar that says "Ln: 6 Col: 4".

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 23 2015, 02:52:03)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> 2 + 3 + 4
9
>>>
```

Figure 11

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 11 shows the expression $2 + 3 + 4$ as typed, followed by *Python*’s response with the value 9. 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 is used the same way in all three.

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 9. This will bring up the window of Figure 12. Select *Terminal* (highlighted with a red circle) and open it.



Figure 12

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 13. 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.

Click on the following links to download two files to the *Python-files* folder that you created on Page 2:—

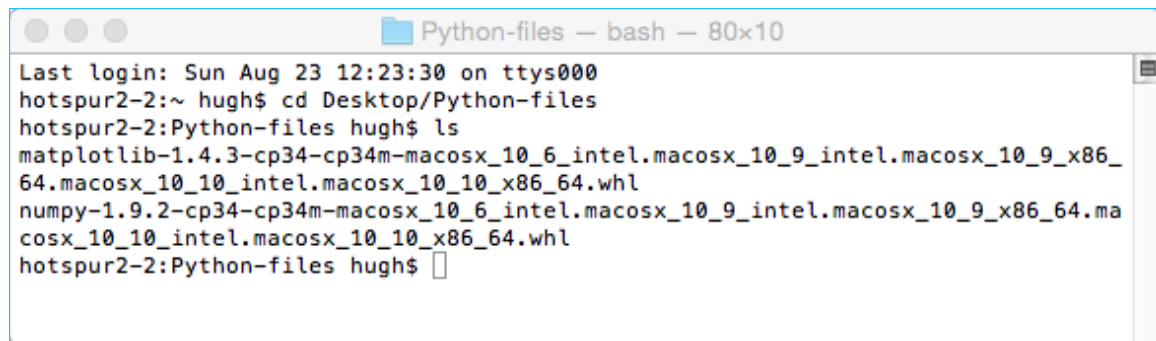
- [numpy-1.9.2-for-MacOS](#)
- [matplotlib-1.4.3-for-MacOS](#)

Note that the actual names of these two files are really long — too long to remember or to type correctly. Instead, it is better to copy and paste them in the Terminal window.

First, type

cd Desktop/Python-files

(or whatever you called the folder to which you downloaded the two files). This is shown in the second line of Figure 13.



```
Python-files — bash — 80x10
Last login: Sun Aug 23 12:23:30 on ttys000
hotspur2-2:~ hugh$ cd Desktop/Python-files
hotspur2-2:Python-files hugh$ ls
matplotlib-1.4.3-cp34-cp34m-macosx_10_6_intel.macosx_10_9_intel.macosx_10_9_x86_
64.macosx_10_10_intel.macosx_10_10_x86_64.whl
numpy-1.9.2-cp34-cp34m-macosx_10_6_intel.macosx_10_9_intel.macosx_10_9_x86_64.ma
cosx_10_10_intel.macosx_10_10_x86_64.whl
hotspur2-2:Python-files hugh$
```

Figure 13

Next, type the command

ls

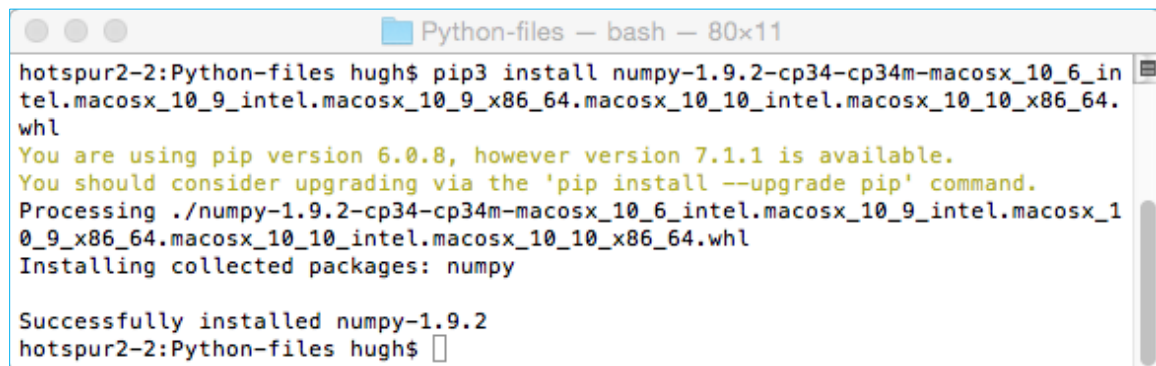
meaning “list directory.” Figure 13 shows, in the fourth and sixth lines the names of the two files, each of which overflows to the next line.

numpy

To install *numpy*, first select the entire name of the *numpy* installation file, right click, and select the *copy* command from the menu. Then type the following line at the prompt, and paste what you copied in lieu of the placeholder

pip3 install <paste numpy file name here>

The entire command should be pasted onto one line, even though it will automatically wrap to the next line due to its length. This command will install *numpy* and all of its dependencies, as shown in Figure 14 below.



```
Python-files — bash — 80x11
hotspur2-2:Python-files hugh$ pip3 install numpy-1.9.2-cp34-cp34m-macosx_10_6_in
tel.macosx_10_9_intel.macosx_10_9_x86_64.macosx_10_10_intel.macosx_10_10_x86_64.
whl
You are using pip version 6.0.8, however version 7.1.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Processing ./numpy-1.9.2-cp34-cp34m-macosx_10_6_intel.macosx_10_9_intel.macosx_1
0_9_x86_64.macosx_10_10_intel.macosx_10_10_x86_64.whl
Installing collected packages: numpy

Successfully installed numpy-1.9.2
hotspur2-2:Python-files hugh$
```

Figure 14

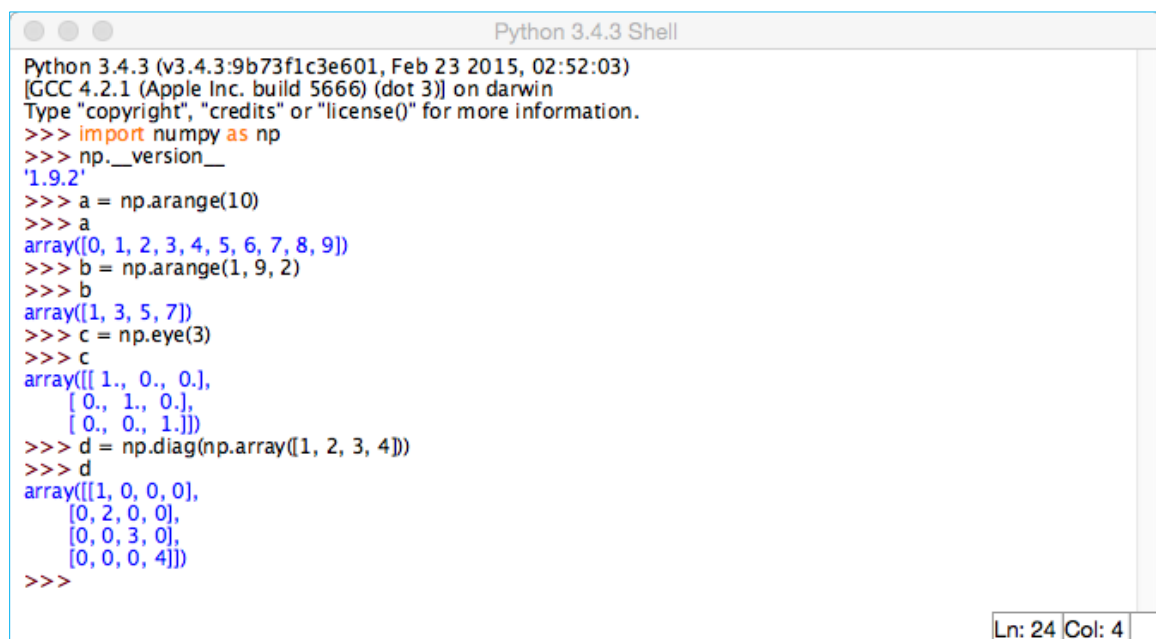
You can test your installation of *numpy* by opening an *IDLE* window, as in Figure 10. Type or paste the following commands into *IDLE*, one line at a time, *exactly* as written:—

```

import numpy as np
np.__version__ 4
a = np.arange(10) 5
a
b = np.arange(1, 9, 2)
b
c = np.eye(3)
c
d = np.diag(np.array([1, 2, 3, 4]))
d

```

The result should resemble Figure 15 below:–



```

Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 23 2015, 02:52:03)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> import numpy as np
>>> np.__version__
'1.9.2'
>>> a = np.arange(10)
>>> a
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> b = np.arange(1, 9, 2)
>>> b
array([1, 3, 5, 7])
>>> c = np.eye(3)
>>> c
array([[ 1.,  0.,  0.],
       [ 0.,  1.,  0.],
       [ 0.,  0.,  1.]])
>>> d = np.diag(np.array([1, 2, 3, 4]))
>>> d
array([[1, 0, 0, 0],
       [0, 2, 0, 0],
       [0, 0, 3, 0],
       [0, 0, 0, 4]])
>>>

```

Ln: 24 Col: 4

Figure 15

You have successfully installed *numpy* 1.9.2. However, we will run another test below.

matplotlib

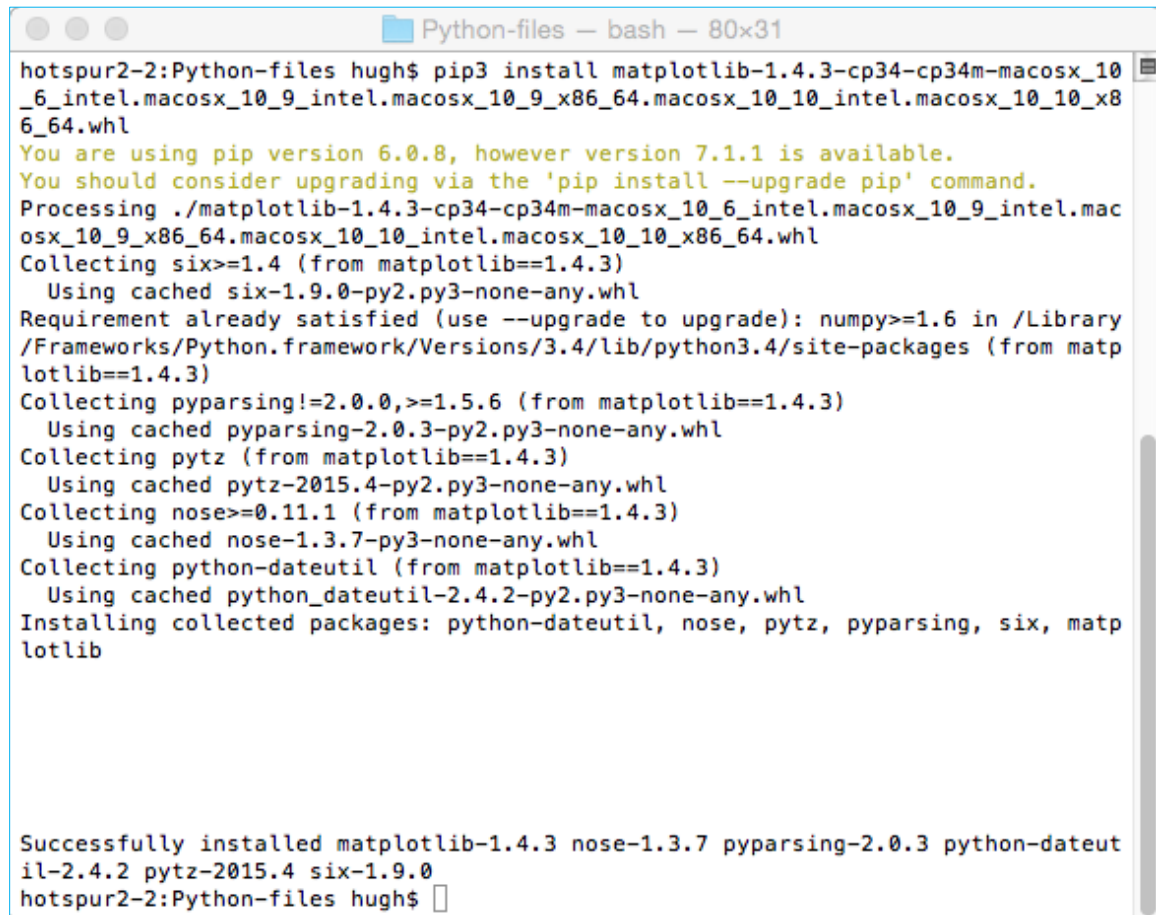
Installation of *matplotlib* is done the same way. Copy the name of the *matplotlib* installation file into the following command into the *Terminal* window, all on one line:–

pip3 install <paste matplotlib file name here>

This is shown in Figure 16 below. Note that, *matplotlib* depends upon several other software packages; the **.whl** file instructs the **pip3 install** command to retrieve them from the Internet and install them, also.

⁴ Note that the word “**version**” is preceded by *two* underscore characters and followed by *two* more underscore characters.

⁵ Note also that the word “**arange**” in several lines has only one ‘r’.



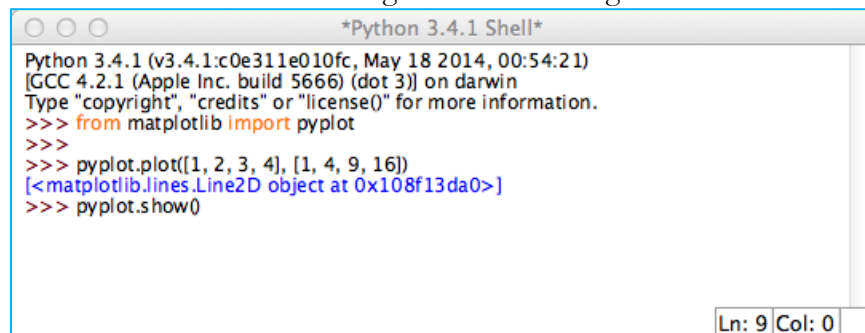
```
Python-files — bash — 80x31
hotspur2-2:Python-files hugh$ pip3 install matplotlib-1.4.3-cp34-cp34m-macosx_10_6_intel.macosx_10_9_intel.macosx_10_9_x86_64.macosx_10_10_intel.macosx_10_10_x86_64.whl
You are using pip version 6.0.8, however version 7.1.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Processing ./matplotlib-1.4.3-cp34-cp34m-macosx_10_6_intel.macosx_10_9_intel.macosx_10_9_x86_64.macosx_10_10_intel.macosx_10_10_x86_64.whl
Collecting six>=1.4 (from matplotlib==1.4.3)
  Using cached six-1.9.0-py2.py3-none-any.whl
Requirement already satisfied (use --upgrade to upgrade): numpy>=1.6 in /Library/Frameworks/Python.framework/Versions/3.4/lib/python3.4/site-packages (from matplotlib==1.4.3)
Collecting pyparsing!=2.0.0,>=1.5.6 (from matplotlib==1.4.3)
  Using cached pyparsing-2.0.3-py2.py3-none-any.whl
Collecting pytz (from matplotlib==1.4.3)
  Using cached pytz-2015.4-py2.py3-none-any.whl
Collecting nose>=0.11.1 (from matplotlib==1.4.3)
  Using cached nose-1.3.7-py3-none-any.whl
Collecting python-dateutil (from matplotlib==1.4.3)
  Using cached python_dateutil-2.4.2-py2.py3-none-any.whl
Installing collected packages: python-dateutil, nose, pytz, pyparsing, six, matplotlib
Successfully installed matplotlib-1.4.3 nose-1.3.7 pyparsing-2.0.3 python-dateutil-2.4.2 pytz-2015.4 six-1.9.0
hotspur2-2:Python-files hugh$
```

Figure 16

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()
```

The IDLE window should look something like the following:–



```
*Python 3.4.1 Shell*
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 00:54:21)
[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 0x108f13da0>]
>>> pyplot.show()
Ln: 9 Col: 0
```

Figure 17

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

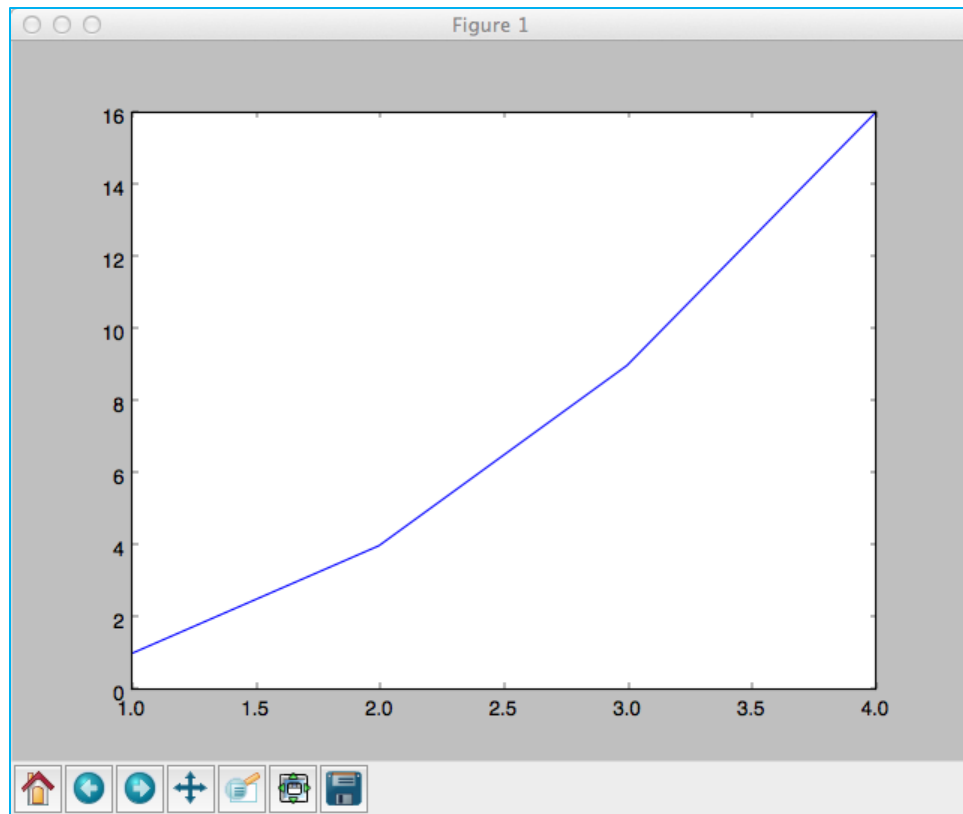


Figure 18

To close this window, click on the “close” button in the upper left.

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

[TestMatplotlib.py](#)

Then, use the *File* menu in the *IDLE* window to open this file, which contains the following lines:–

```
from pylab import *  
  
t = arange(0.0, 2.0, 0.01)  
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()
```

Click the *Run > Run Module* command in the menu at the top of the *IDLE* file window. This will cause the *Python* program to run and to produce the window of Figure 19.

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

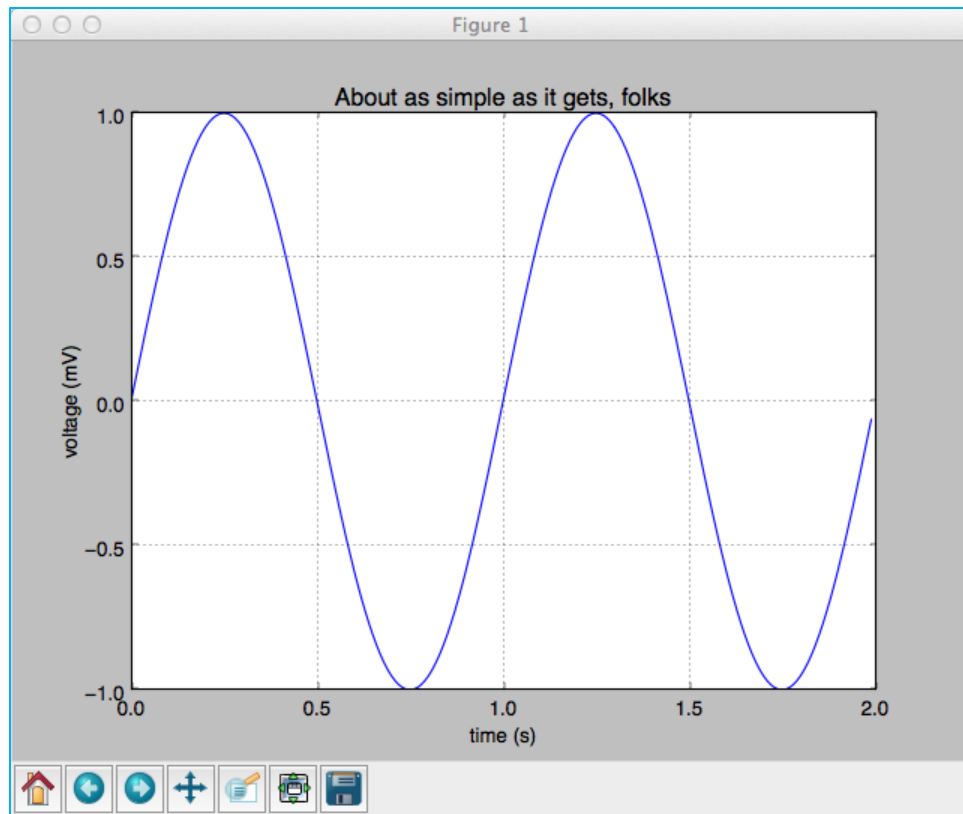


Figure 19

Congratulations! You now have a working version of *matplotlib* installed.

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 of the test ends with the lines in Figure 20.

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.”

```
Python 3.4.3 Shell
.....K.....
Ran 5198 tests in 76.553s
OK (KNOWNFAIL=6, SKIP=20)
<nose.result.TextTestResult run=5198 errors=0 failures=0>
>>>
```

Figure 20

The last line shows that 5198 tests were run with no errors or failures.

Congratulations! You now have both *matplotlib 1.4.3* and *numpy 1.9.2* 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. Follow the instructions on p.488 of the textbook.

Installing Python 3.4 on Linux

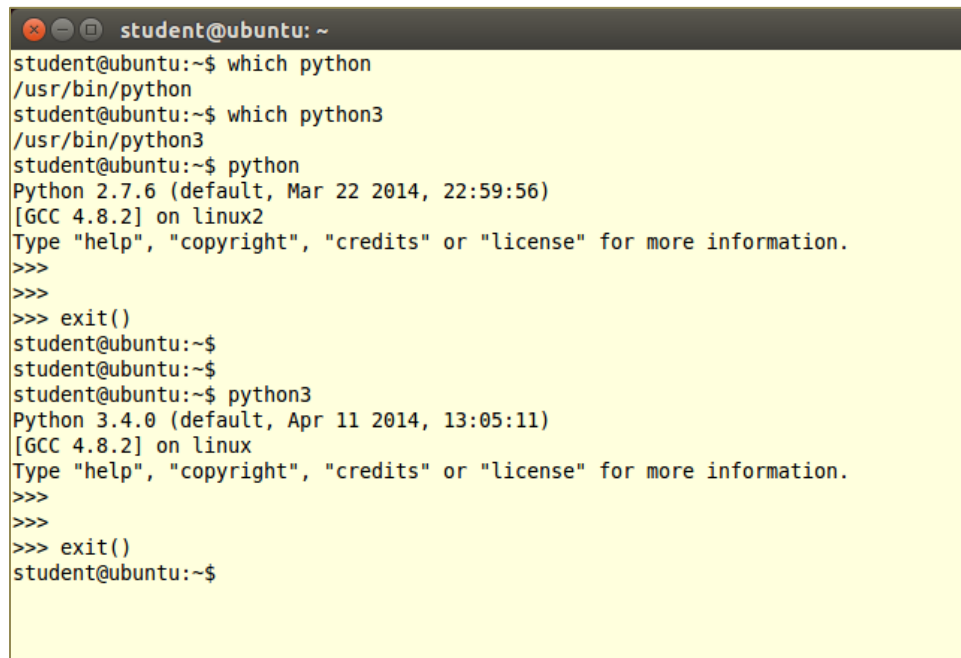
Linux comes in many flavors supported by different organizations with different approaches to software installation. The desktop/laptop versions of Linux typically package the most common applications — including *Python* — in their automated download and installation systems. The following describes 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):—



```
student@ubuntu: ~
student@ubuntu:~$ which python
/usr/bin/python
student@ubuntu:~$ which python3
/usr/bin/python3
student@ubuntu:~$ python
Python 2.7.6 (default, Mar 22 2014, 22:59:56)
[GCC 4.8.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> exit()
student@ubuntu:~$
student@ubuntu:~$ python3
Python 3.4.0 (default, Apr 11 2014, 13:05:11)
[GCC 4.8.2] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> exit()
student@ubuntu:~$
```

Figure 21

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 21 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.

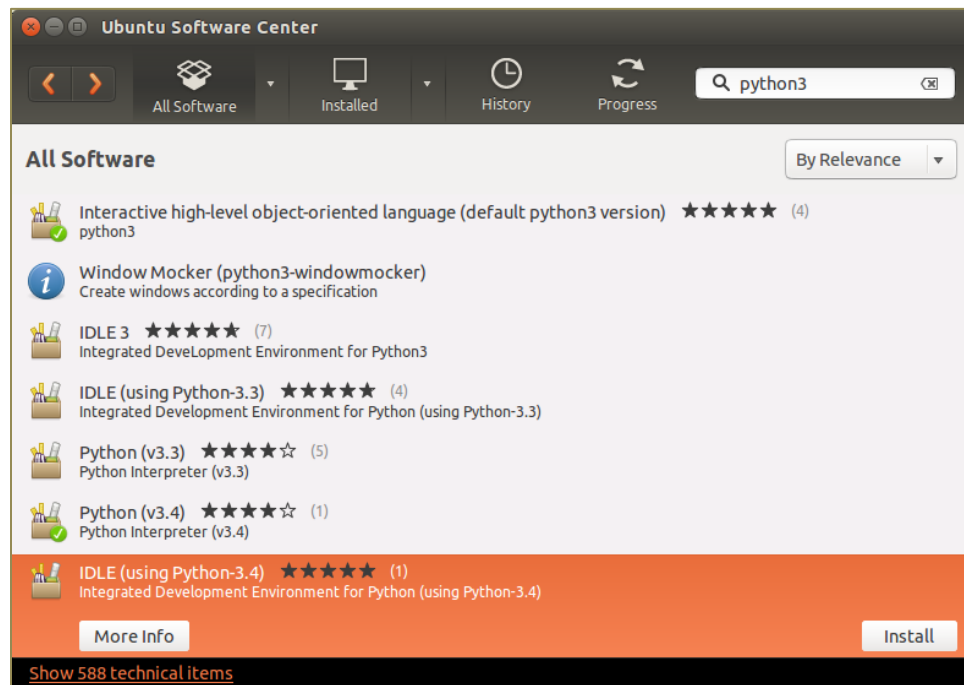
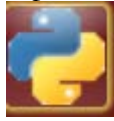


Figure 22

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 11 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.”

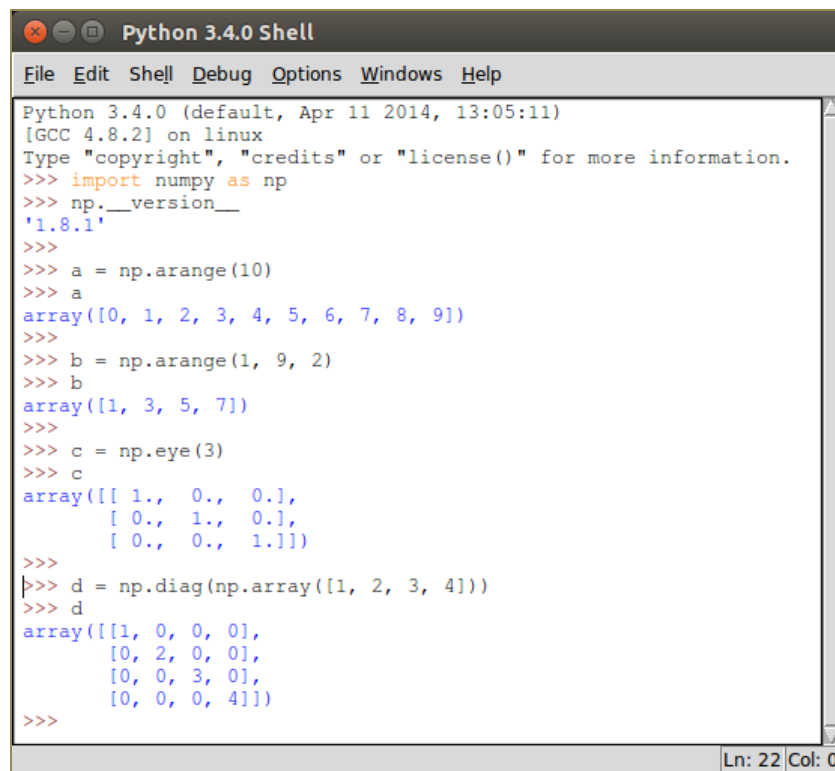
Select on the *More Info* button to bring up the following window:—



Figure 23

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 24 below. The version of *numpy* is 1.8.1, as indicated by line 5 of this window, but your version may be different

A screenshot of a Python 3.4.0 Shell window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Windows', and 'Help'. The main text area shows the following code and output:

```
Python 3.4.0 (default, Apr 11 2014, 13:05:11)
[GCC 4.8.2] on linux
Type "copyright", "credits" or "license()" for more information.
>>> import numpy as np
>>> np.__version__
'1.8.1'
>>>
>>> a = np.arange(10)
>>> a
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
>>>
>>> b = np.arange(1, 9, 2)
>>> b
array([1, 3, 5, 7])
>>>
>>> c = np.eye(3)
>>> c
array([[ 1.,  0.,  0.],
       [ 0.,  1.,  0.],
       [ 0.,  0.,  1.]])
>>>
>>> d = np.diag(np.array([1, 2, 3, 4]))
>>> d
array([[1, 0, 0, 0],
       [0, 2, 0, 0],
       [0, 0, 3, 0],
       [0, 0, 0, 4]])
>>>
```

The status bar at the bottom right shows 'Ln: 22 Col: 0'.

Figure 24

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:–

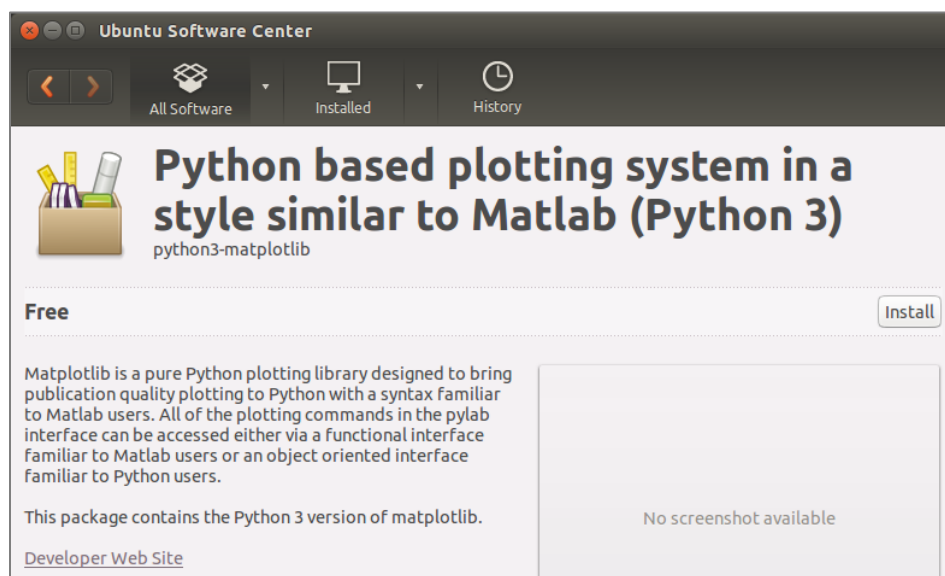
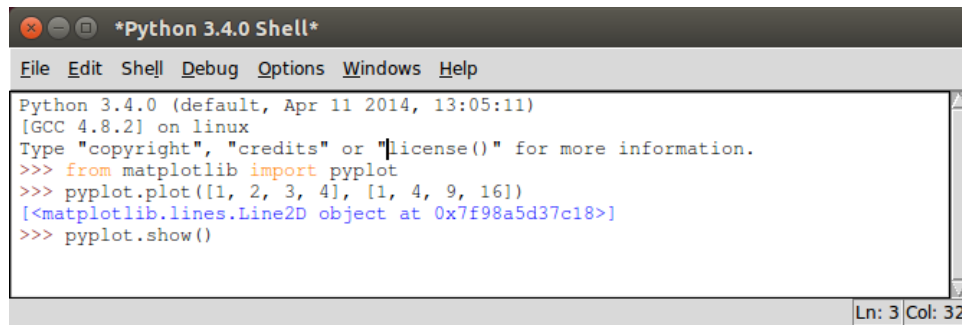


Figure 25

Install this, just as you installed *numpy*. To test your installation, carry out the tests of Figure 17 and of Figure 19. This should produce the *IDLE* window of Figure 26 plots shown in those figures



```
*Python 3.4.0 Shell*
File Edit Shell Debug Options Windows Help
Python 3.4.0 (default, Apr 11 2014, 13:05:11)
[GCC 4.8.2] on linux
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 0x7f98a5d37c18>]
>>> pyplot.show()
```

Figure 26

Conclusion

Congratulations! You have successfully used *Python 3.4* 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

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

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