Setting up Python 3.4 and numpy and matplotlib  
on your own Windows PC or laptop



**CS-1004, Introduction to Programming for Non-Majors, C-Term 2015**

Hugh C. Lauer[[1]](#footnote-2)©  
Adjunct Professor  
Worcester Polytechnic Institute

Programming assignments in CS-1004 will be in 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* on *Windows 7* and *Windows 8* platforms. The second part of the document provides instructions on how to install additional *Python* packages, such as *numpy* and *matplotlib*.[[2]](#footnote-3)

Public laboratory computers at WPI will have *Python 3.4*, *numpy*, and *matplotlib* installed on them for C- and D-terms of the academic year 2014-2015.

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.*

## Installing Python 3.4 on Windows Systems[[3]](#footnote-4), [[4]](#footnote-5)

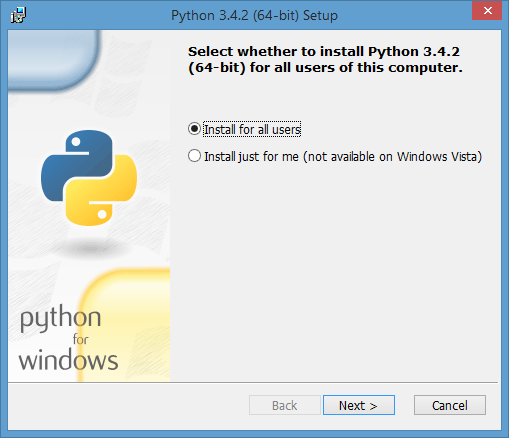
There are two variants of *Python 3.4* for Windows — a 32-bit version and a 64-bit version. Obviously, the 64-bit version requires a 64-bit Windows computer. Almost all Windows PCs sold over the past few years are 64-bit, and most of them have the 64-bit version of the *Windows 7* or *Windows 8* installed. Although the 32-bit version of *Python* can run on *both* 64-bit Windows PCs *and* 32-bit Windows PCs, these instructions are for installing the 64-bit version. *If you have a 32-bit Windows operating system, see the Professor or TAs for assistance.*

To obtain the correct version of *Python*, click on this link — [python-3.4.2.amd64.msi](http://www.cs.wpi.edu/~cs1004/c15/Resources/Windows/python-3.4.2.amd64.msi)— and download the resulting file to a convenient folder or directory. Alternatively, you may browse to

<http://www.cs.wpi.edu/~cs1004/c15>

and download it from there.

Double-click on the file python-3.4.2.amd64.msi to start the installation. You should be greeted by a dialog box resembling the following:–



Figure

If, instead, you see a dialog box resembling Figure 2 below for any version of *Python 3*, select *Remove Python* for that version. This will remove an old, stale version, including any 32-bit versions of *Python 3*. Removing *Python* will take several minutes and may require you to confirm in one or more additional dialog boxes.

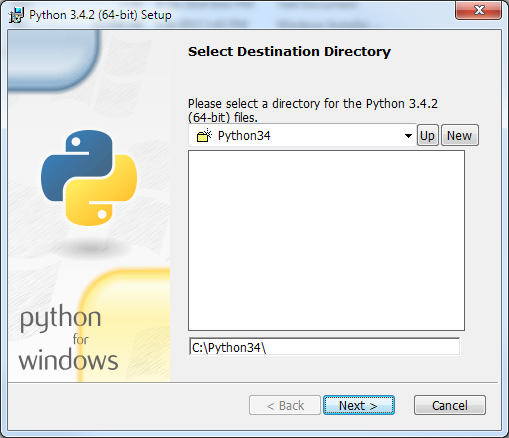
Note: Even if you don’t see the dialog box of Figure 2, if there is a previous version of *Python* installed on your computer, you should uninstall it before continuing.



Figure

After you have removed your previous version of *Python*, click *Finish* and start over at Figure 1. In Figure 1, whether you choose to install “for all users” or just for yourself is a matter of personal preference.

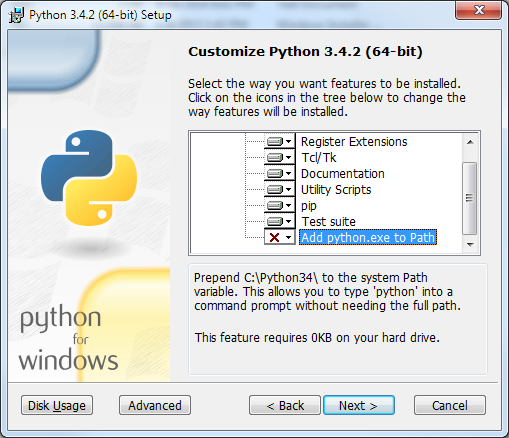
Click *Next* to bring up the following dialog box.



Figure

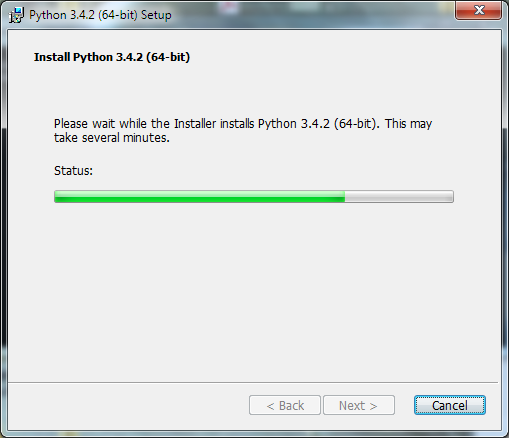
Click *Next* to select the default directory. If it tells you the directory already exists and asks if you are sure that you want to overwrite existing files, click *Yes*.

In the next dialog box (Figure 4 below), you need to customize the installation. Click on the “X” at the bottom under *Add python.exe to Path*. It will expand this line to several options. Select “Will be installed on local hard drive*.”* This facility lets you run *Python* and related programs from command prompts, something that you will need to do later on when installing other packages.



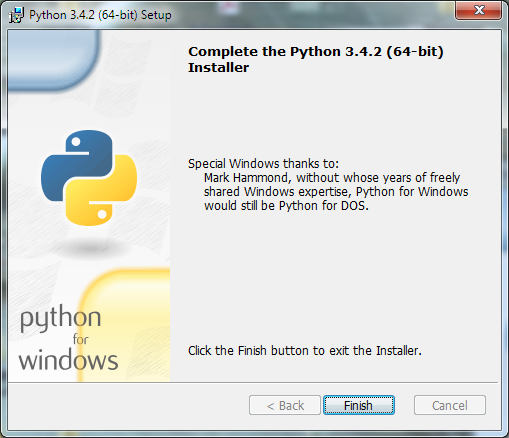
Figure

Click *Next* to begin the installation. The progress of the installation is shown in a dialog box resembling Figure 5 below.



Figure

The installation should begin, will take several minutes, and may require confirmation in additional dialog boxes.[[5]](#footnote-6) When it completes, you should see the final dialog box, below.

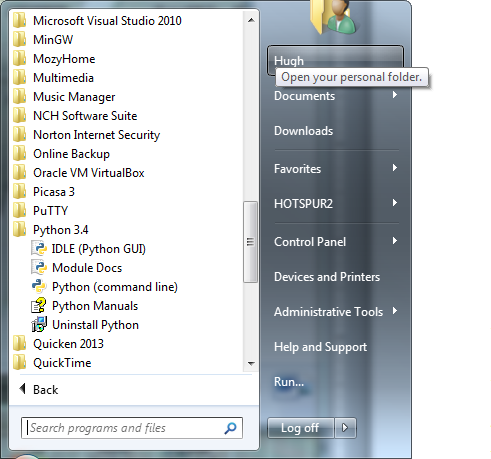


Figure

Click *Finish* to complete the installation of *Python 3.4.2.*

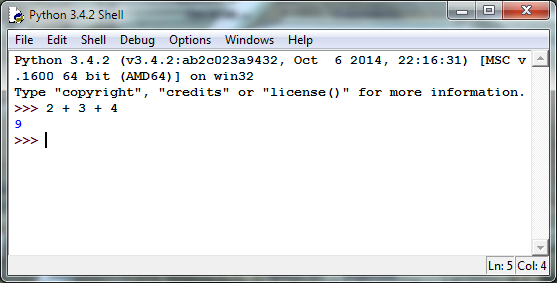
### Testing your installation on Windows 7

If you are running *Windows 7*, you may confirm your installation by clicking the *Start* button to bring up the Windows *Start* menu. Select *All Programs* and scroll down to *Python 3.4.* Open this folder to expose shortcuts similar to the following:–



Figure

Click on *IDLE (Python GUI)* to bring up the following window (only the upper part of which is shown here):–



Figure

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, in Figure 8, the expression *2 + 3 + 4* was typed and *Python* responded 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.

#### Testing your installation in Windows 8

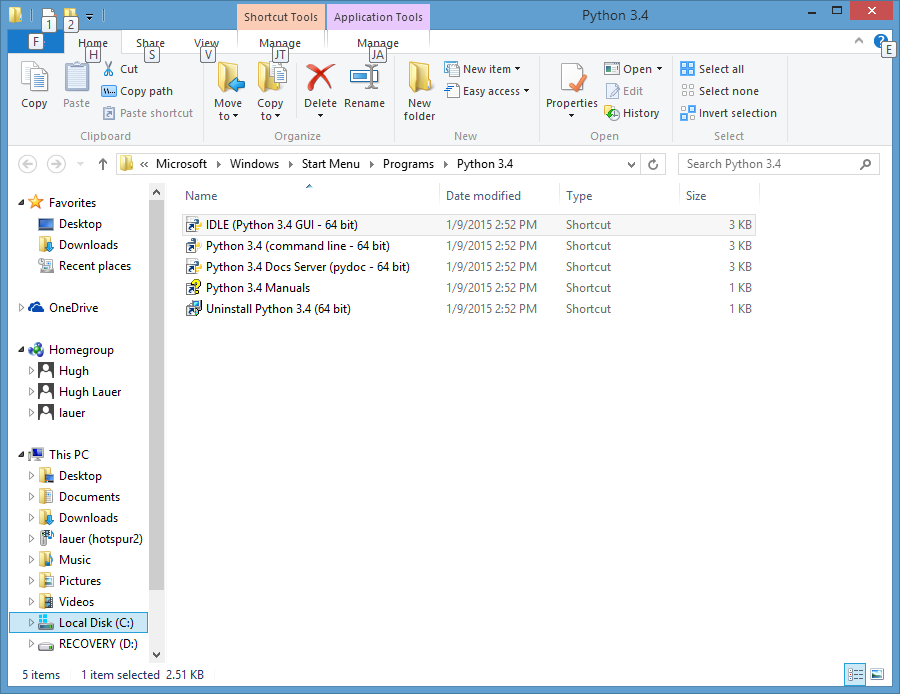
*Windows 8* does not have a *Start* button but rather a *Start* screen that is intended to make the user experience more like the smartphone experience.[[6]](#footnote-7) Unfortunately, when *Python* is installed as instructed above, its icon does not automatically appear on the *Start* screen. It also does not appear in the list of apps.

To find it, move the cursor to the upper-right or lower-right corner of the screen to expose the *Windows 8* pallet of “charms”. Select the *Search* charm to bring up a *Search* box. Type the word “Python.” This will bring up a list of matching items, shown in Figure 9 below.



Figure

Note that this list is similar to the *Python 3.4* folder in the Start Menu in Figure 7. *Right-click* on the item labeled *IDLE (Python 3.4 GUI).* From the menu, select “Pin to Start” to cause an icon to be added to the *Start* screen. You may also want to pin the item to the *Task bar* (i.e., the bar of tiny icons at the bottom of the screen). You may also select “Open file location,” which will bring up the following window:–



Figure

From this window, you can copy any or all of the *Python* links to the desktop.

To test your installation, double-click on the *IDLE (Python GUI)* icon and carry out the same tests as shown above under Figure 8.

## 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, also needed by *matplotlib*), and
* *graphics.py*, a simple tool written in *Python 3* and created by the textbook author for making simple drawings.

In addition, several small, utility packages are needed to support these. You can download the packages from <http://www.cs.wpi.edu/~cs1004/c15/Resources/Windows> , except for graphics.py, which is at <http://web.cs.wpi.edu/~cs1004/c15/Resources/graphics.py>.

Download the following packages to a convenient folder:–

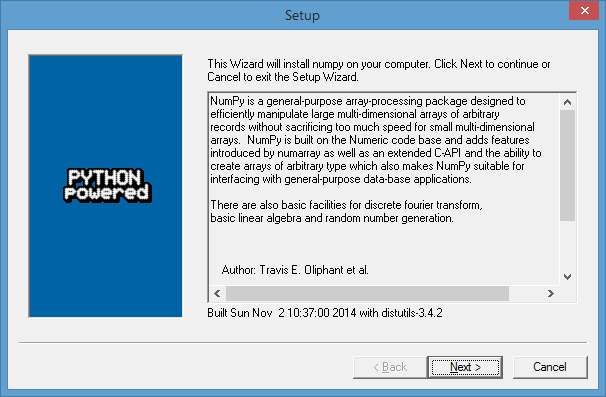
* [numpy-MKL-1.9.1.win-amd64-py3.4.exe](http://www.cs.wpi.edu/~cs1004/c15/Resources/Windows/numpy-MKL-1.9.1.win-amd64-py3.4.exe)
* [pyparsing-2.0.3-py2.py3-none-any.whl](http://www.cs.wpi.edu/~cs1004/c15/Resources/Windows/pyparsing-2.0.3-py2.py3-none-any.whl)
* [python\_dateutil-2.4.0-py2.py3-none-any.whl](http://www.cs.wpi.edu/~cs1004/c15/Resources/Windows/python_dateutil-2.4.0-py2.py3-none-any.whl)
* [matplotlib-1.4.2.win-amd64-py3.4.exe](http://www.cs.wpi.edu/~cs1004/c15/Resources/Windows/matplotlib-1.4.2.win-amd64-py3.4.exe)

### Installing Graphics.py

*Graphics.py* is a simple drawing package that we will use a lot in this course. To install it, click on this link — [graphics.py](http://web.cs.wpi.edu/~cs1004/c15/Resources/graphics.py) — and download the file *to the folder where you keep your Python programs*. Follow the instructions on p.488 of the textbook.

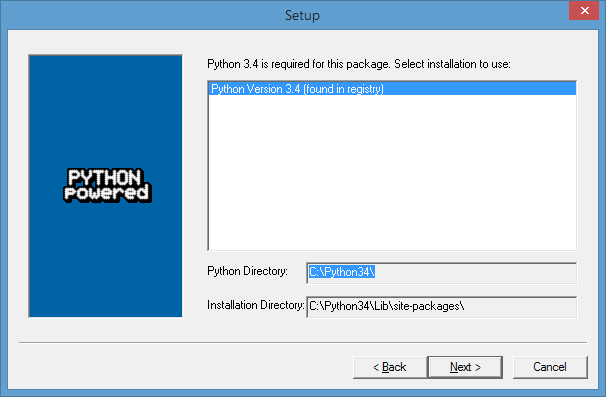
### Installing numpy 1.9.1

The numpy package needs to be installed immediately after you install *Python 3.4.2* itself. In either *Windows 7* or *Windows 8*, double-click (or open) the *numpy* installer that you downloaded above. After confirming that you do want to allow the system to install software, it will start the installation and show the following dialog box:–



Figure

Click *Next*. If your installation of *Python 3.4* is correct, you should get the following dialog:–



Figure

If instead, it complains that you do not have *Python 3.4* installed, ask for help. Such a complaint could arise if *Python* was not correctly installed or if you have an incompatible version.

After the installation completes, click *Finish*. Note that you might have to click somewhere in some window to get the *Finish* dialog box to pop up. Note also that this installer contains all of the dependencies of *numpy* — i.e., other packages needed by *numpy* to run. These are installed silently.

You can test your installation of *numpy* by opening an *IDLE* window, as in Figure 8. Type or paste the following commands into IDLE, one line at a time, *exactly* as written:–

**import numpy as np**

**np.\_\_version\_\_** [[7]](#footnote-8)

**a = np.arange(10)** [[8]](#footnote-9) **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 the following figure:–



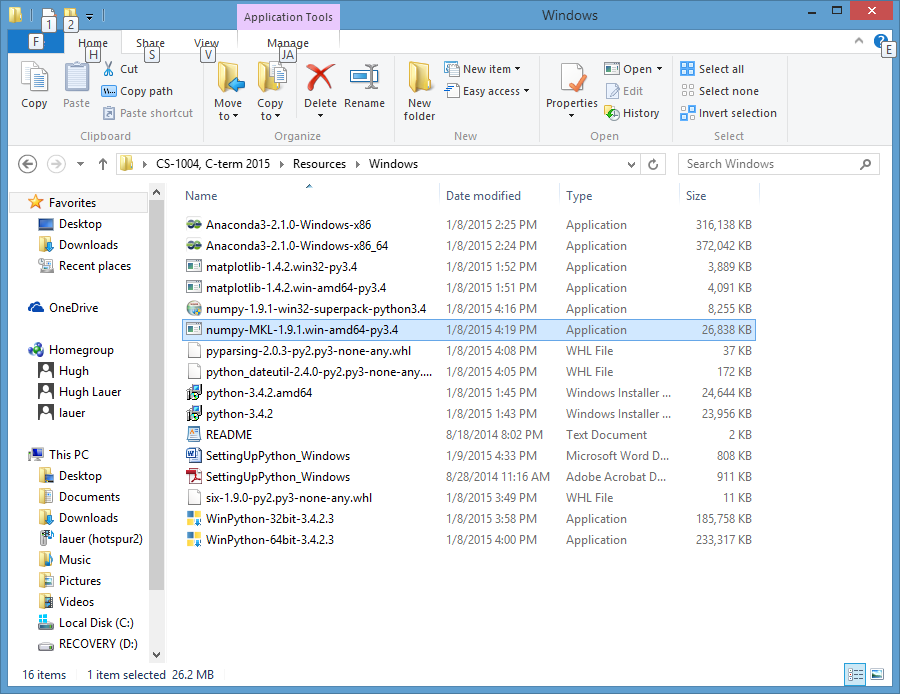
Figure

Congratulations! You have now installed a working versions *numpy 1.9.1.*

### Installing Matplotlib and its Dependencies on Windows

Installing *Matplotlib* is not nearly so straightforward. There is no officially released installer that captures all of the dependencies (as there is for the Macintosh). The only installer that the Professor could find at the time of this writing is the one downloaded above; this contains just *Matplotlib* itself, but not any of the packages upon which it depends. Therefore, you need to manually install each of those packages. We will do this first, using a *Windows* command prompt.

First, open a *Command Prompt* window. In *Window 7*, this can be found in the *Accessories* folder under the *Start Menu*. In *Windows 8*, you may have to search for it the same way that you searched for the *Python GUI* in Figure 9.[[9]](#footnote-10) In the command prompt, change your working directory to the folder where you downloaded the packages. This is not so easy in *Windows* as on the Macintosh or Linux. First, you need to get the full pathname of the folder. In the folder window, right-click on the *address line* as shown in Figure 14 below. A menu will pop up; select the menu command *Copy address as text*.



Figure

Next, go to the *Command Prompt* window that you just opened (above). Click in this window and type the command cd followed by a space. Then right click the mouse to paste the address of Figure 14 that you just copied as text. The result should be something resembling the following command, with the address of your folder substituted for this example:–

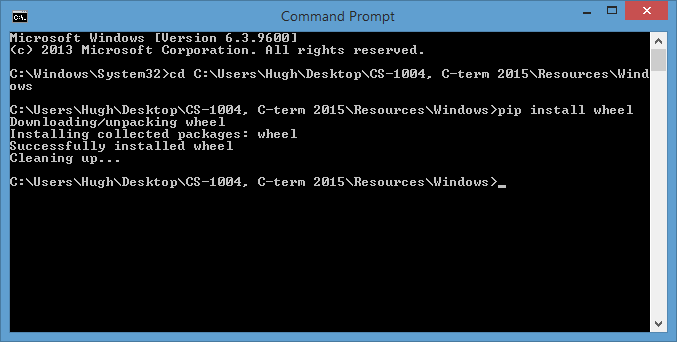
cd C:\Users\Hugh\Desktop\CS-1004, C-term 2015\Resources\Windows

This changes the current “working” directory to be that of your download folder. You can list the contents of the folder by typing the command dir, followed by the Enter key.

Next, type the following command (followed by the Enter key):–

pip install wheel

In this command, pip is the *Python Installation Program*, install is the command to install something, and wheel is the operand of the install command — in this case, the tool for opening files with the .whl extension. The result should resemble the following:–

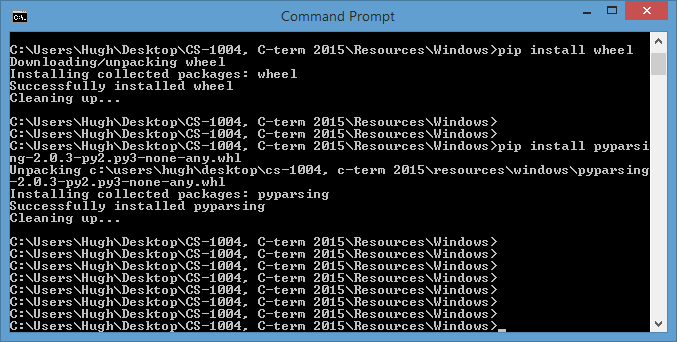


Figure

Next, install the pyparsing package by typing the command

pip install pyparsing-2.0.3-py2.py3-none-any.whl

The result should resemble the response in Figure 16 below.

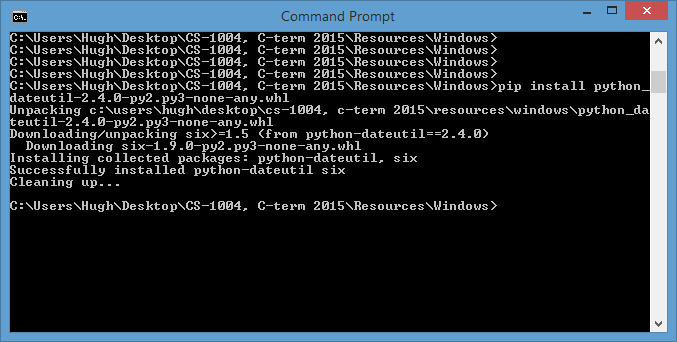


Figure

Next, install the python date utility by typing the following command:–

pip install python\_dateutil-2.4.0-py2.py3-none-any.whl

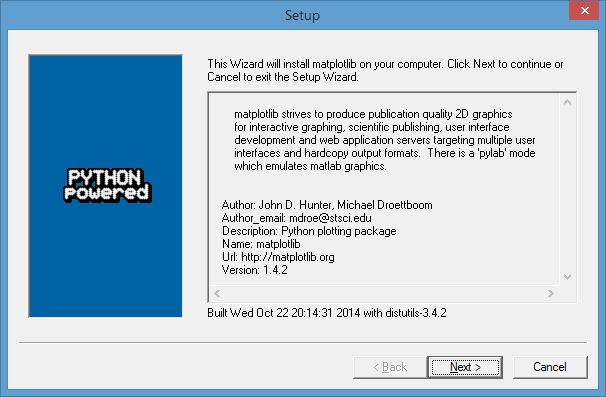
The result should resemble the following:–



Figure

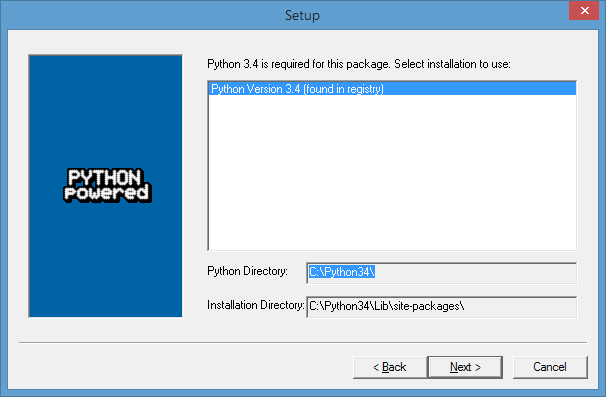
With these packages installed, you are now ready to install matplotlib, an important tool in this course. First, close the command prompt window by typing the exit command. Next, return to the window where you downloaded the installation files and double click on matplotlib-1.4.2.win-amd64-py3.4. This starts the matplotlib installer.

*Windows* may ask if you really want to run this program from an unknown source; click on “yes.” The installer then starts with the following window:–



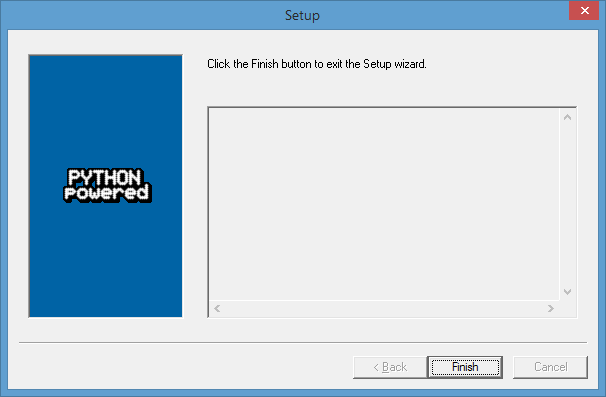
Figure

Click *Next* to continue with the installation.



Figure

After one more dialog box to confirm that you ready to install *matplotlib*, the installation begins. In a short time, it completes with the following dialog box:–

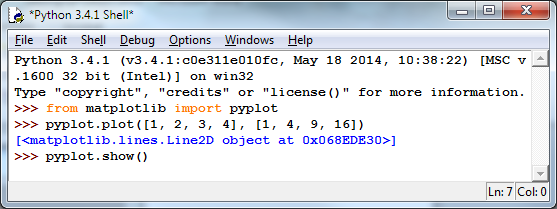


Figure

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

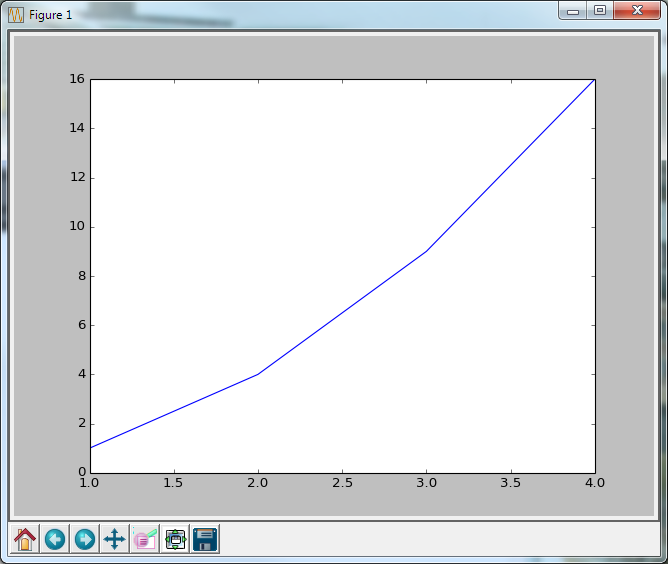
from matplotlib import pyplot  
pyplot.plot([1, 2, 3, 4], [1, 4, 9, 16])  
pyplot.show()

The IDLE window should look something like the following:–



Figure

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



Figure

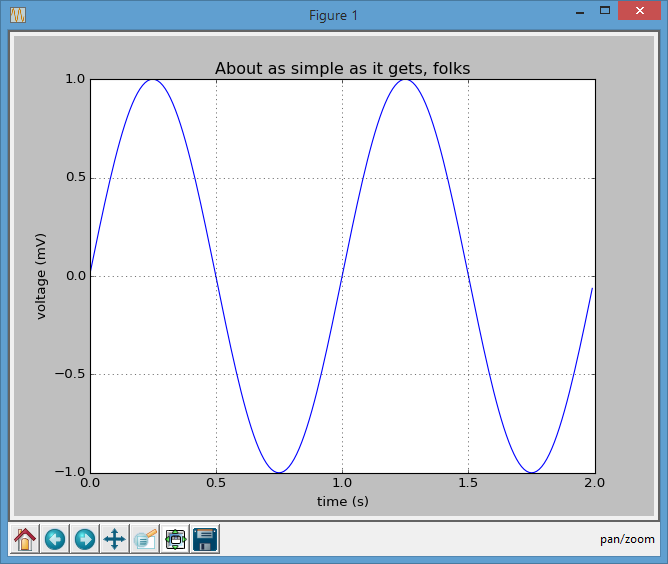
To close this window, click on the “close” button in the upper right.

Finally, to see a more interesting plot, copy and paste the following code, line-by-line, into the IDLE window:–

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()

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

\Figure

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

Note: Be sure to conduct these tests early in the term. There won’t be enough time to discover problems and fix them when a homework assignment is due the next day.

1. © Copyright 2015, Hugh C. Lauer. All rights reserved. Permission is given for use in courses at Worcester Polytechnic Institute, Worcester, Massachusetts. [↑](#footnote-ref-2)
2. If you have a Macintosh or Linux computer or laptop, please refer to this documents instead:– [docx](http://www.cs.wpi.edu/~cs1004/c15/Resources/Macintosh/SettingUpPython_Macintosh-Linux.docx), [pdf](http://www.cs.wpi.edu/~cs1004/c15/Resources/Macintosh/SettingUpPython_Macintosh-Linux.pdf) [↑](#footnote-ref-3)
3. 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! [↑](#footnote-ref-4)
4. These instructions have been tested on both *Windows 7* and *Windows 8*. [↑](#footnote-ref-5)
5. These additional dialog boxes are occasionally hidden behind other windows. If nothing seems to be happening, try clicking on or moving windows to look for such a dialog box. [↑](#footnote-ref-6)
6. In the Professor’s opinion, this was a mistake. [↑](#footnote-ref-7)
7. Note that the word “version” is preceded by *two* underscore characters and followed by *two* more underscore characters. [↑](#footnote-ref-8)
8. Note the spelling of “arange” with only one ‘r’. [↑](#footnote-ref-9)
9. Once you have found it, it is convenient to create a shortcut to it on your desktop or Task Bar. [↑](#footnote-ref-10)