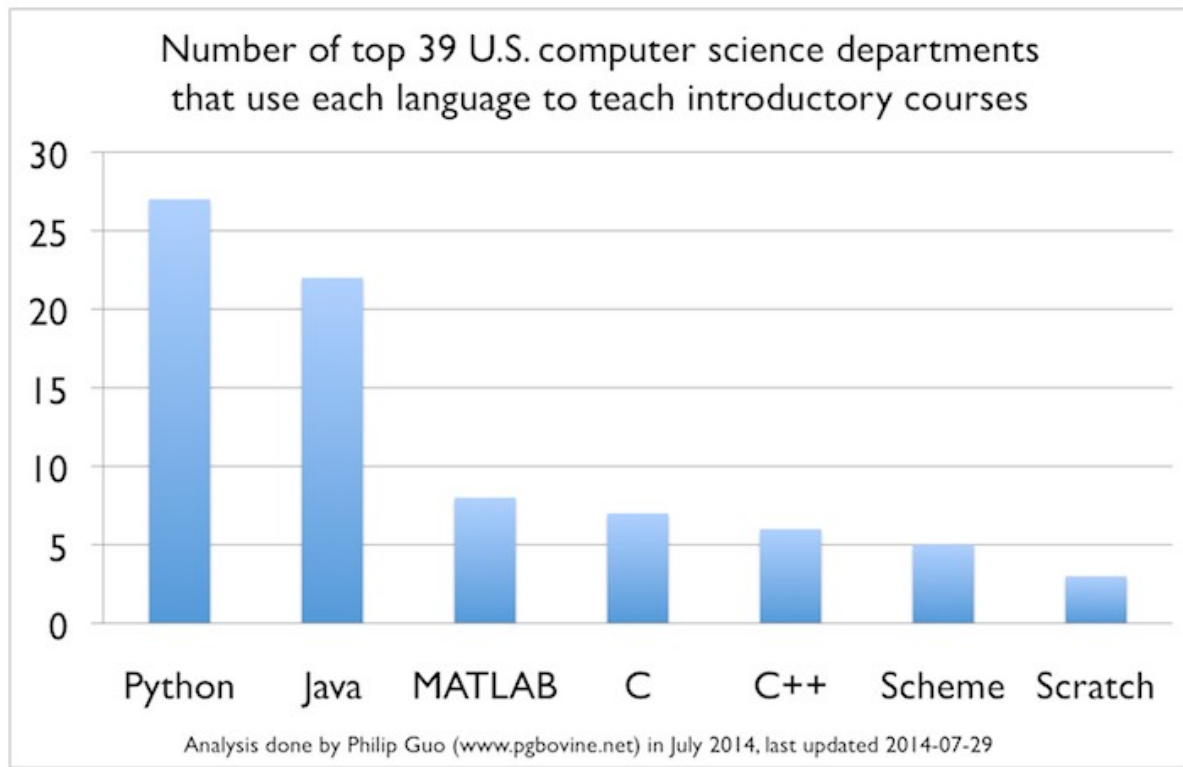
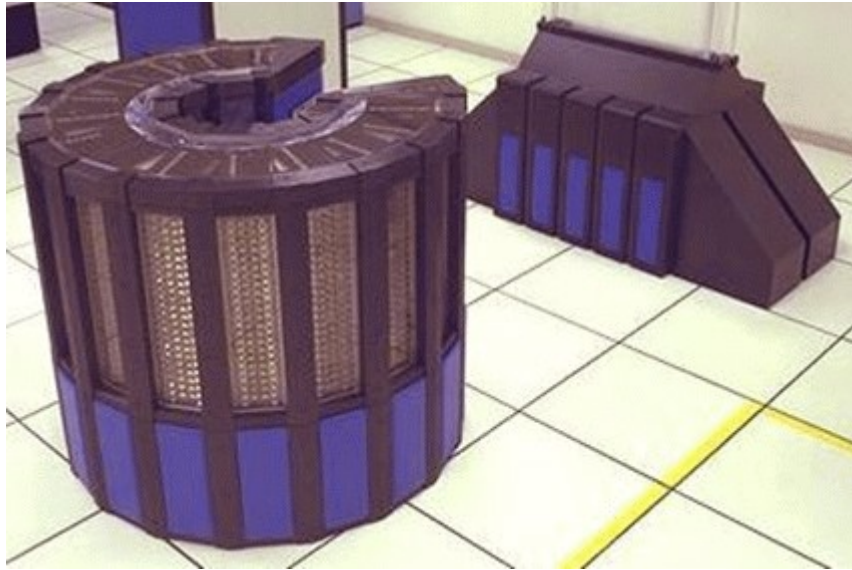
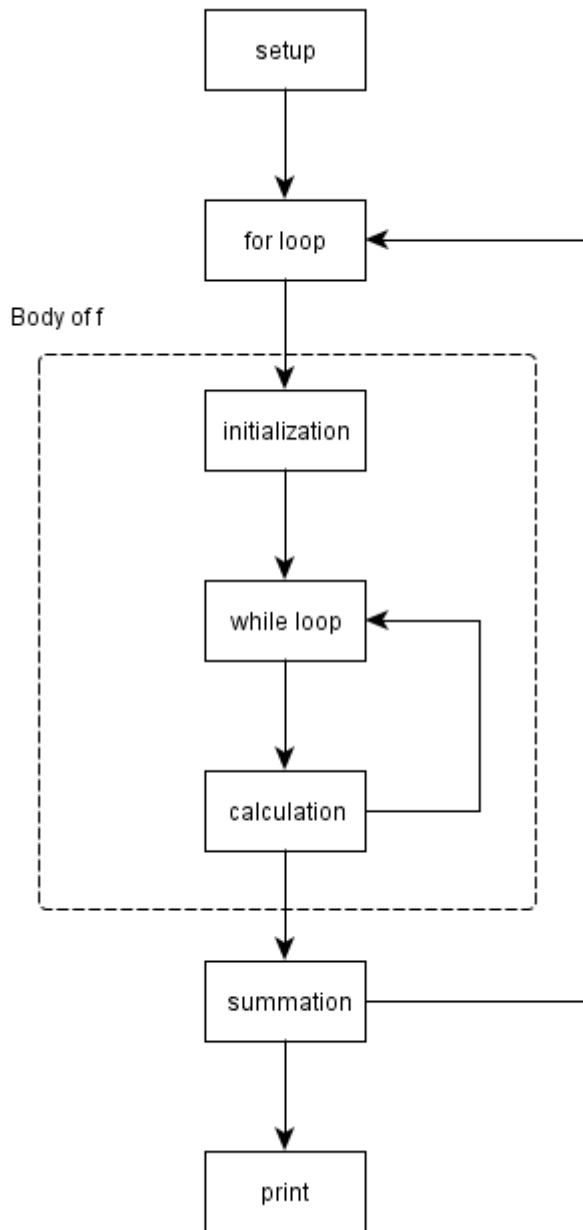


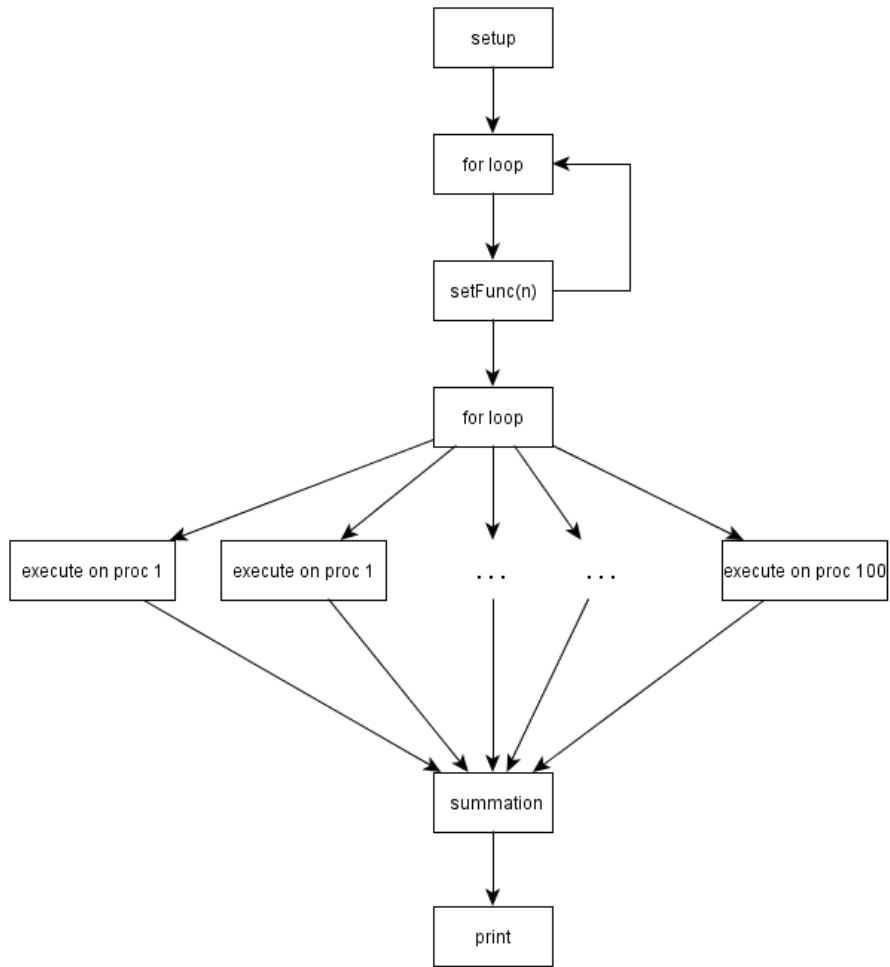
Chapter 1: Using IPython for HPC



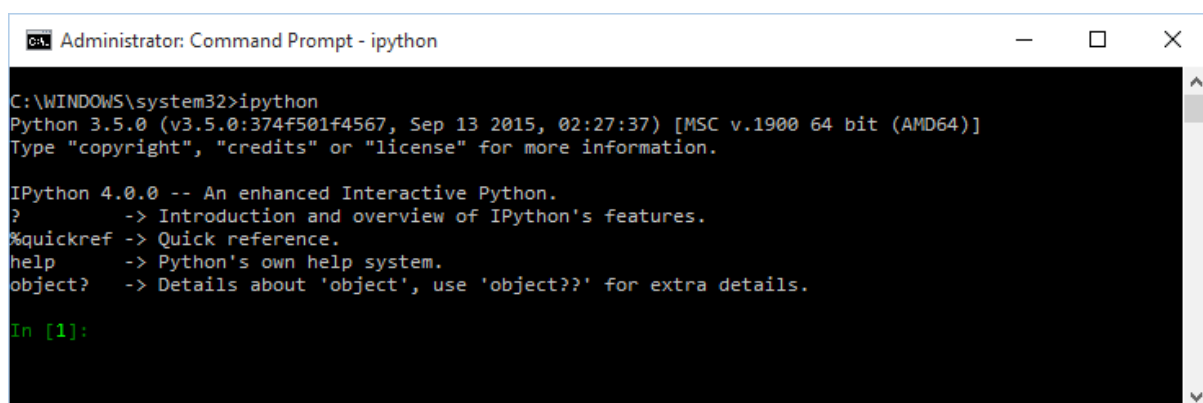
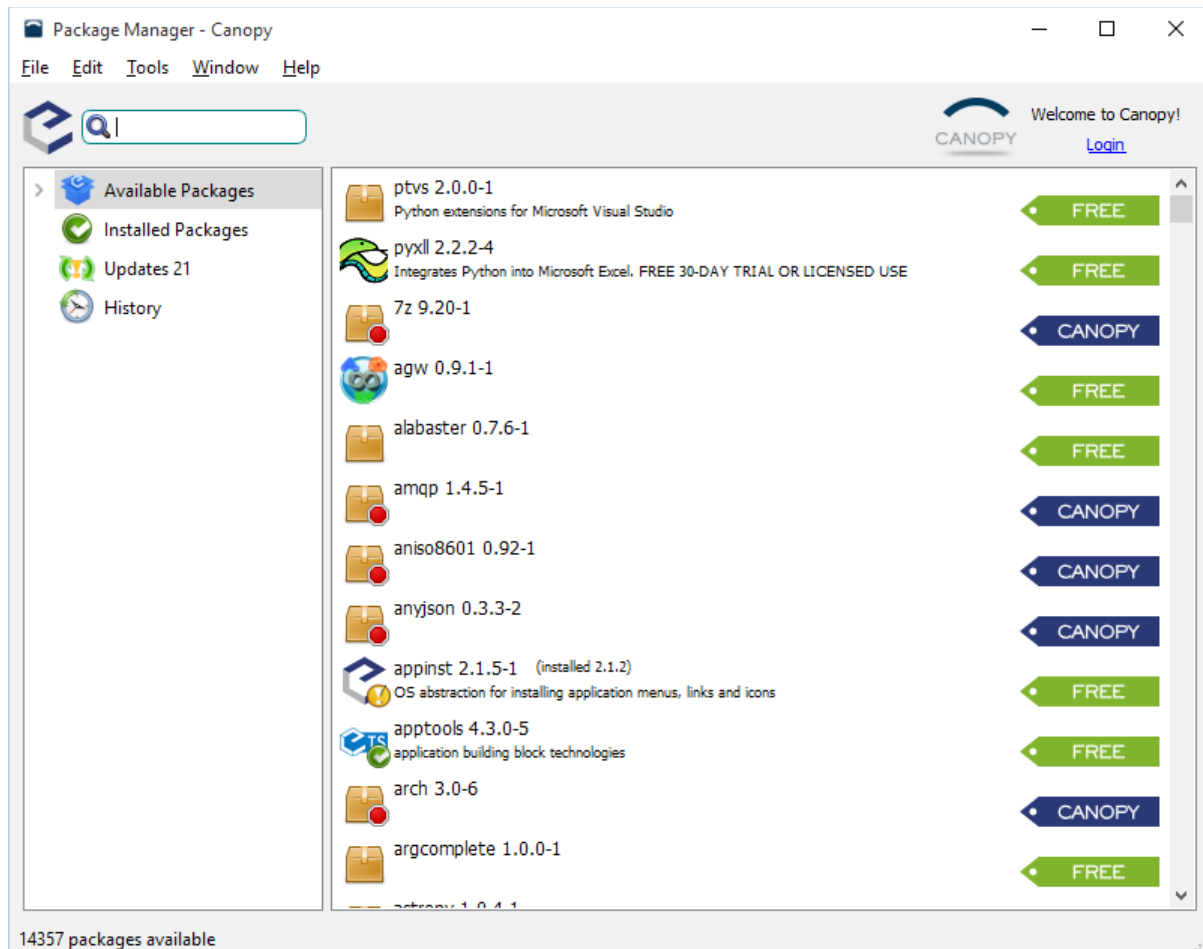
Language Rank	Types	Spectrum Ranking	Spectrum Ranking
1. Java	🌐 📱 🖥️	100.0	100.0
2. C	📱 🖥️ 🖨️	99.9	99.3
3. C++	📱 🖥️ 🖨️	99.4	95.5
4. Python	🌐 🖥️	96.5	93.5
5. C#	🌐 📱 🖥️	91.3	92.4
6. R	🖥️	84.8	84.8
7. PHP	🌐	84.5	84.5
8. JavaScript	🌐 📱	83.0	78.9
9. Ruby	🌐 🖥️	76.2	74.3
10. Matlab	🖥️	72.4	72.8







Chapter 2: Advanced Shell Topics



```

=====
l list ll longlist

ipdb> ?

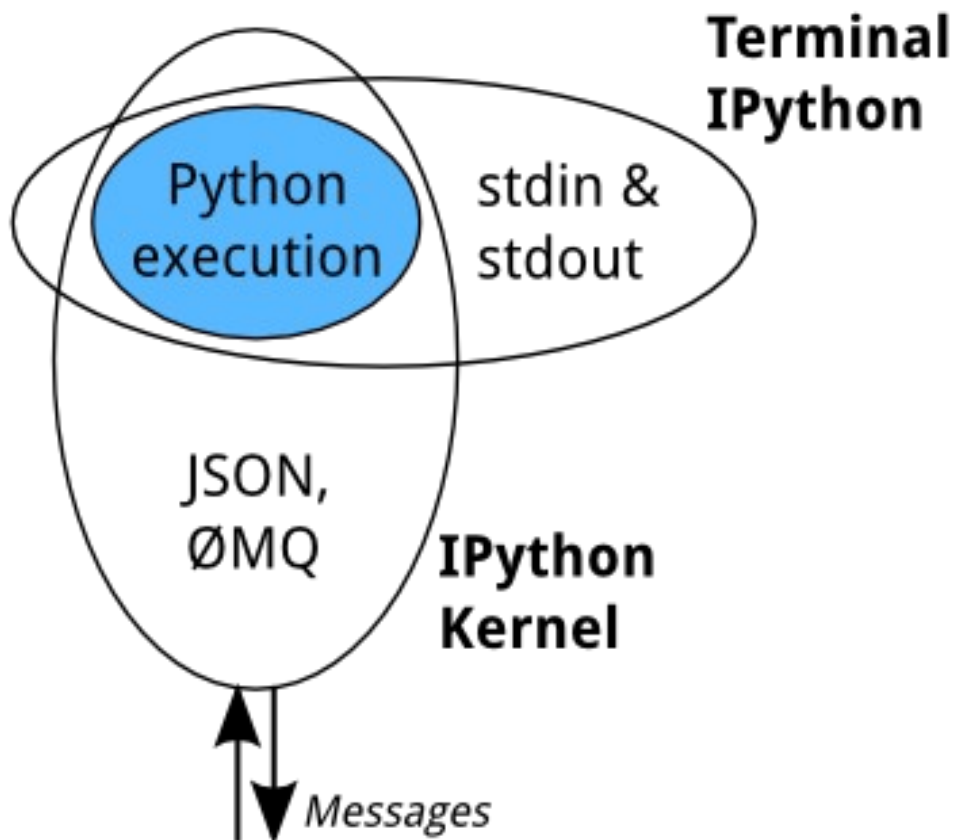
Documented commands (type help <topic>):
=====
EOF      c          d          h          next      pp         retval    u          whatis
a        cl         debug     help       p         psource   run       unalias   where
alias    clear     disable   ignore     pdef      q         rv        undisplay
args     commands  display   interact   pdoc      quit      s         unt
b        condition down       j          pfile     r         source    until
break    cont      enable    jump       pinfo     restart   step      up
bt       continue  exit      n          pinfo2    return    tbreak    w

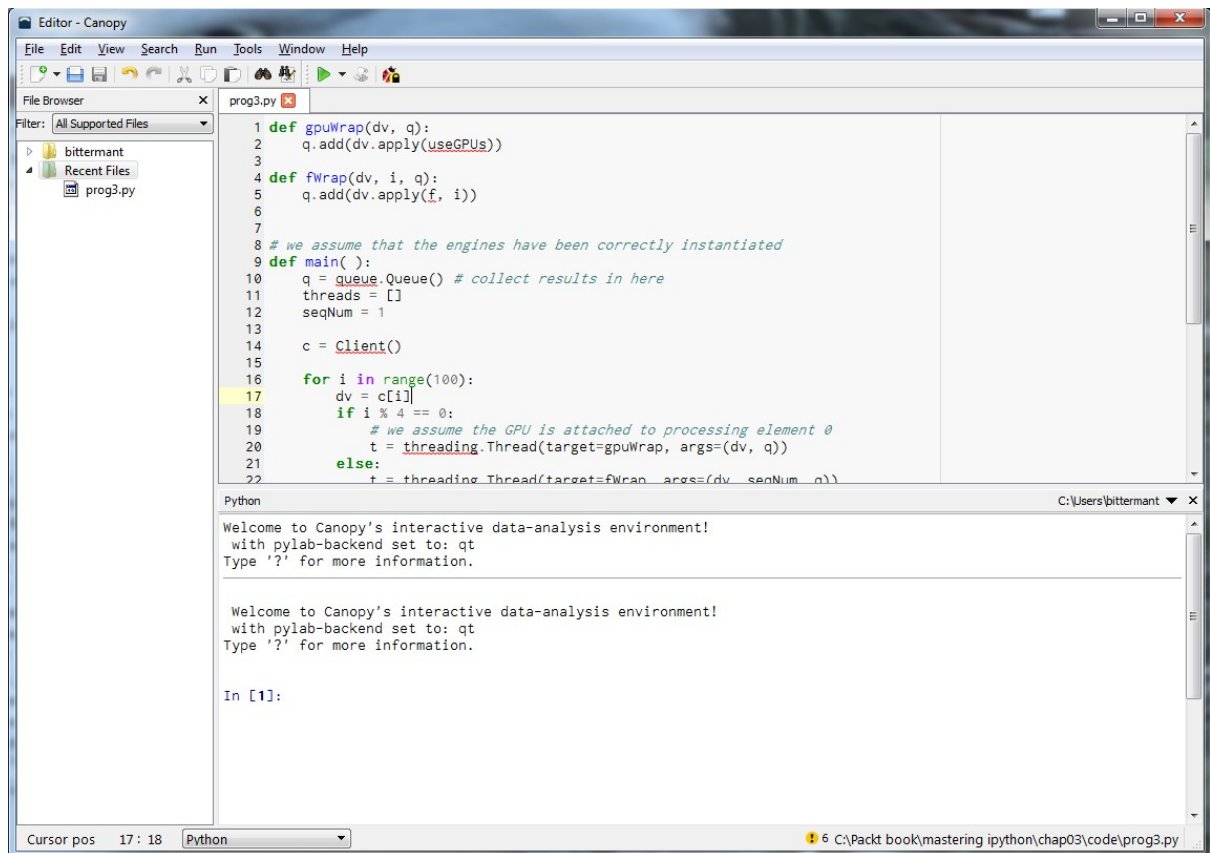
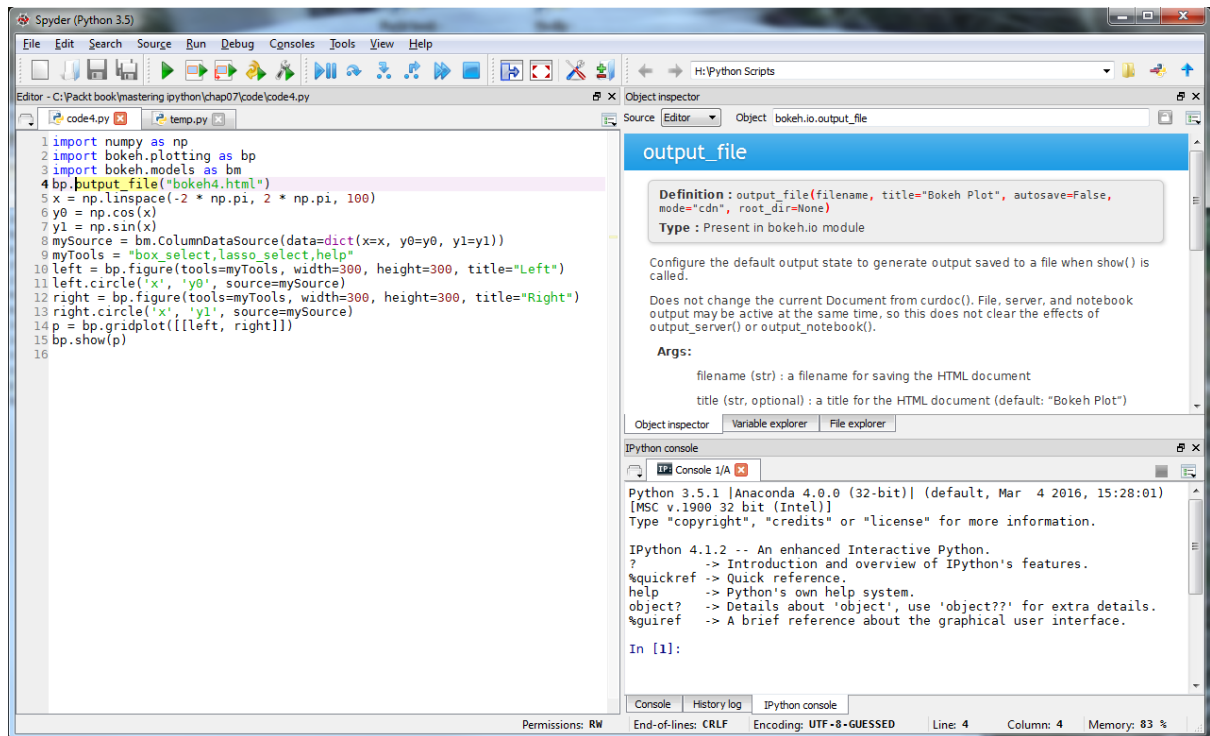
Miscellaneous help topics:
=====
exec     pdb

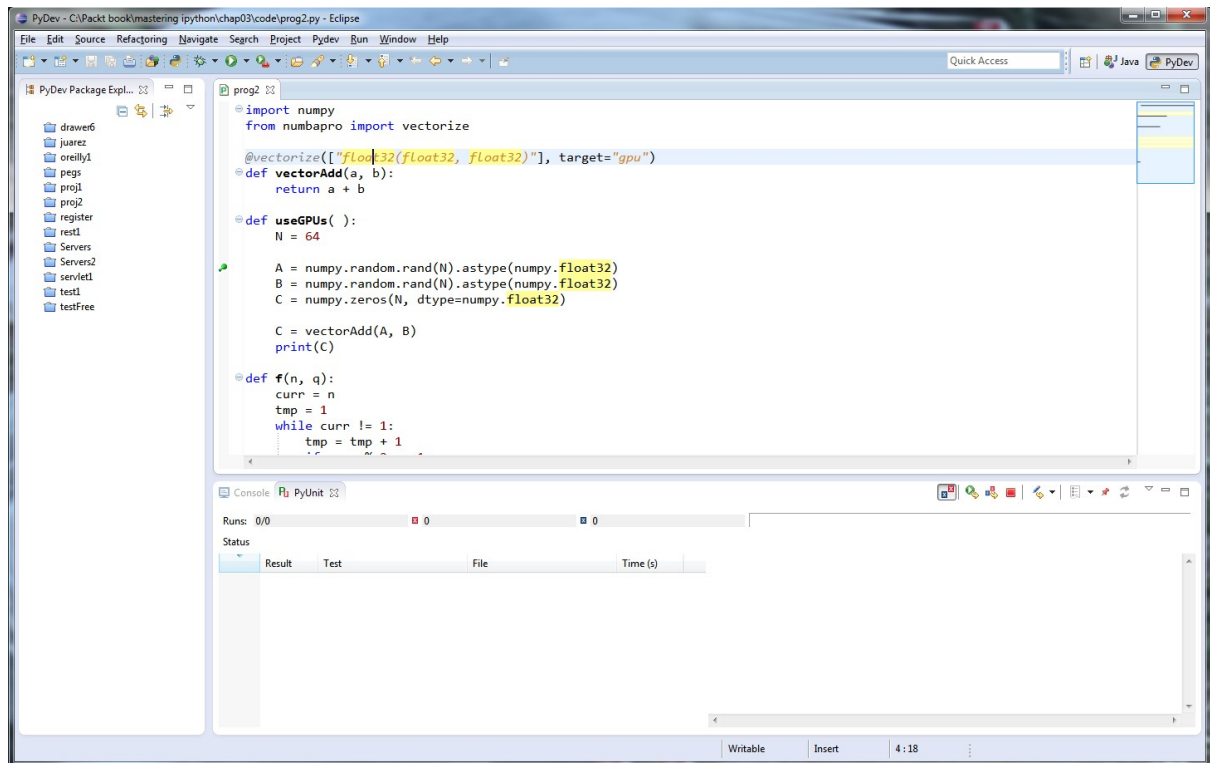
Undocumented commands:
=====
l list ll longlist

ipdb> |

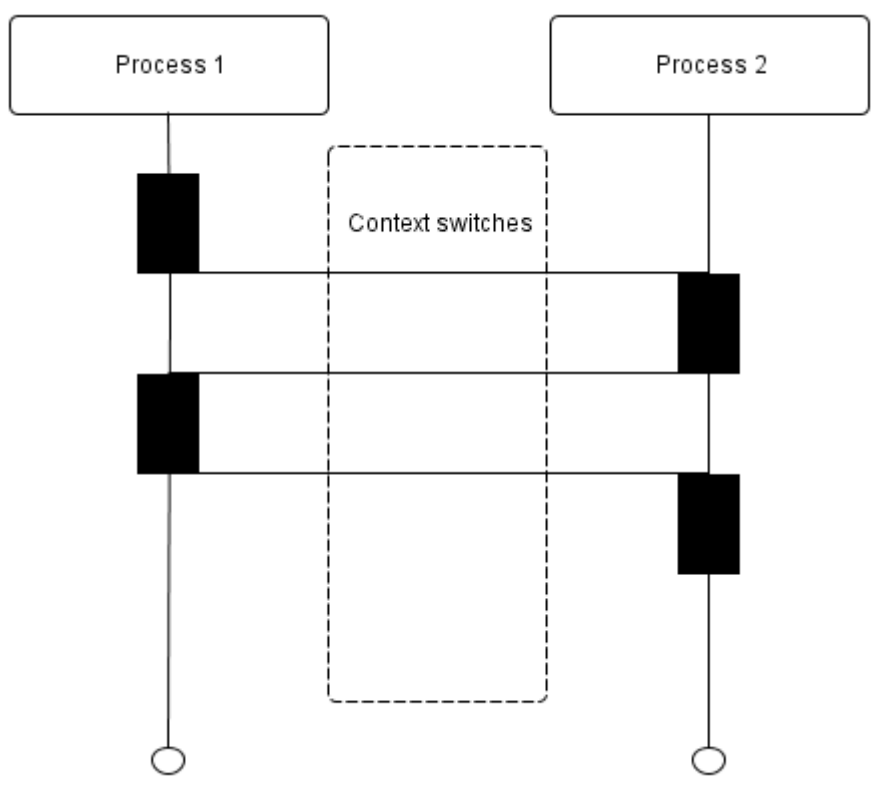
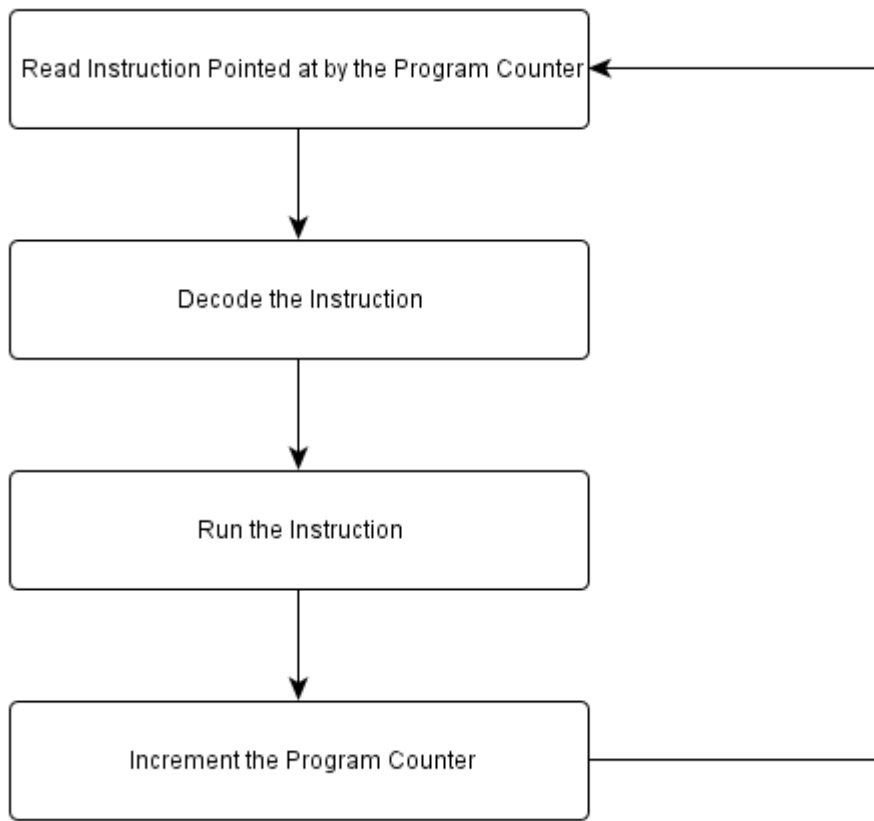
```



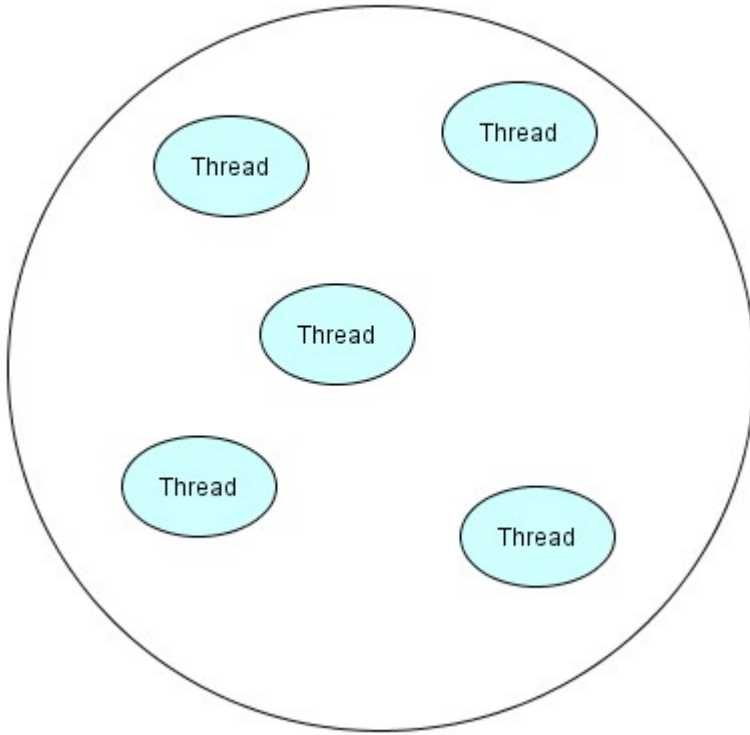


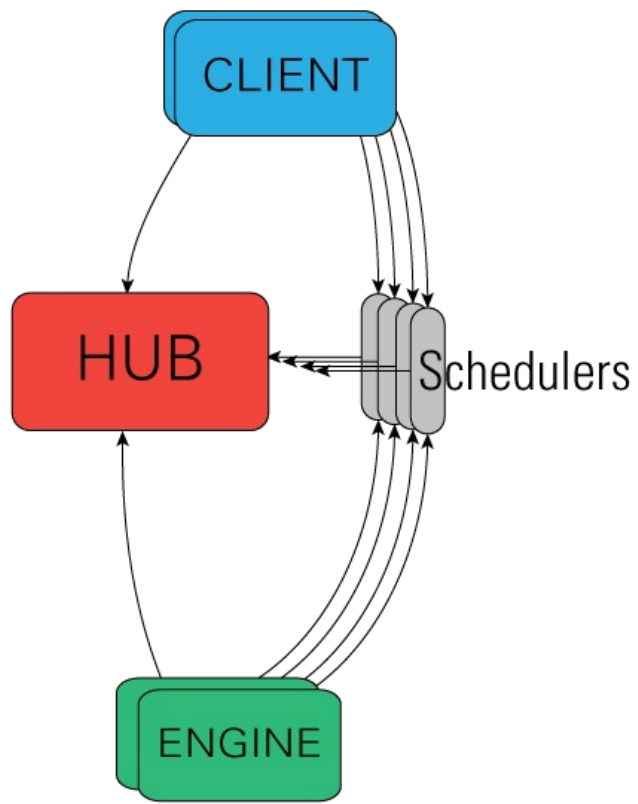


Chapter 3: Stepping Up to IPython for Parallel Computing

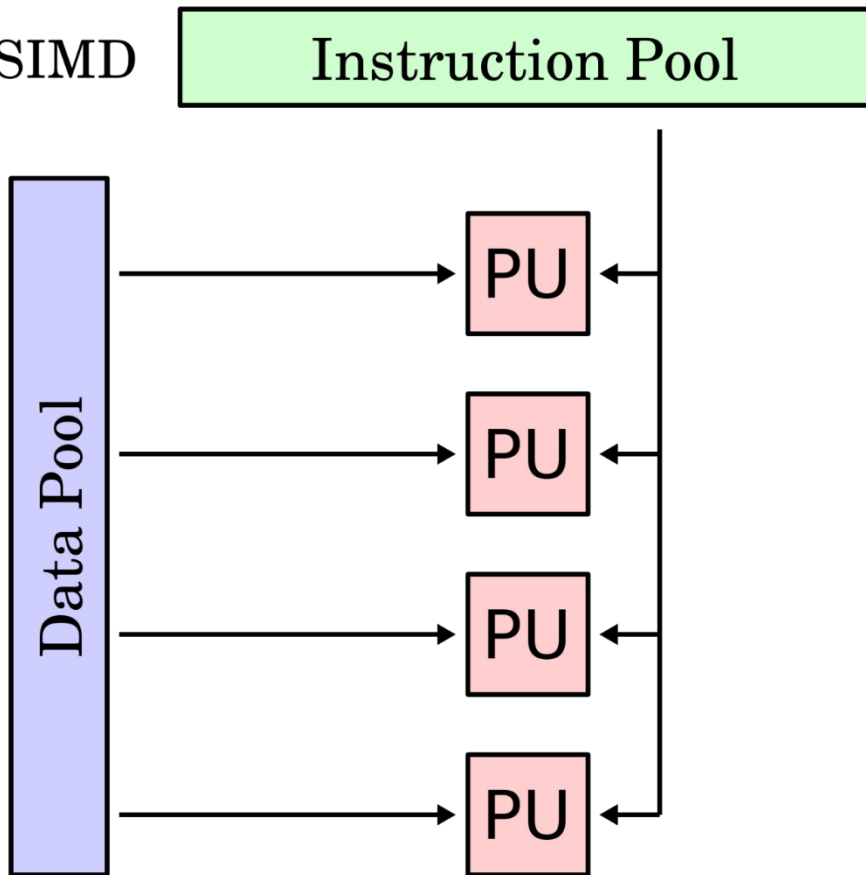


Process



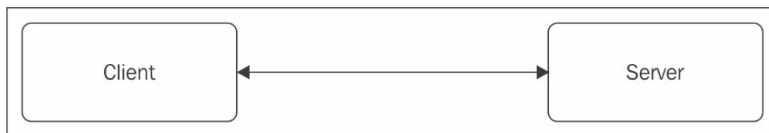
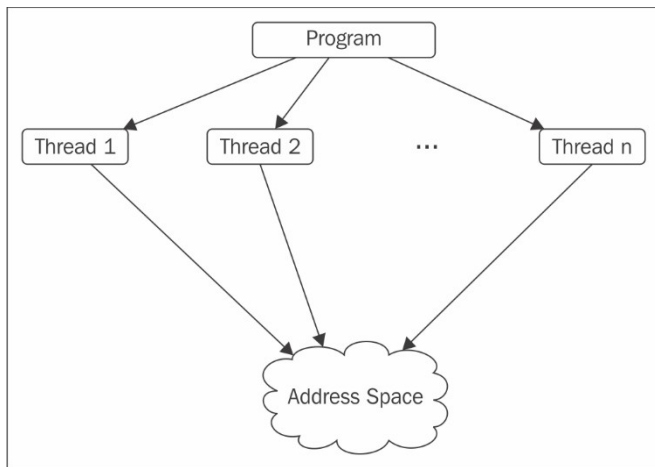
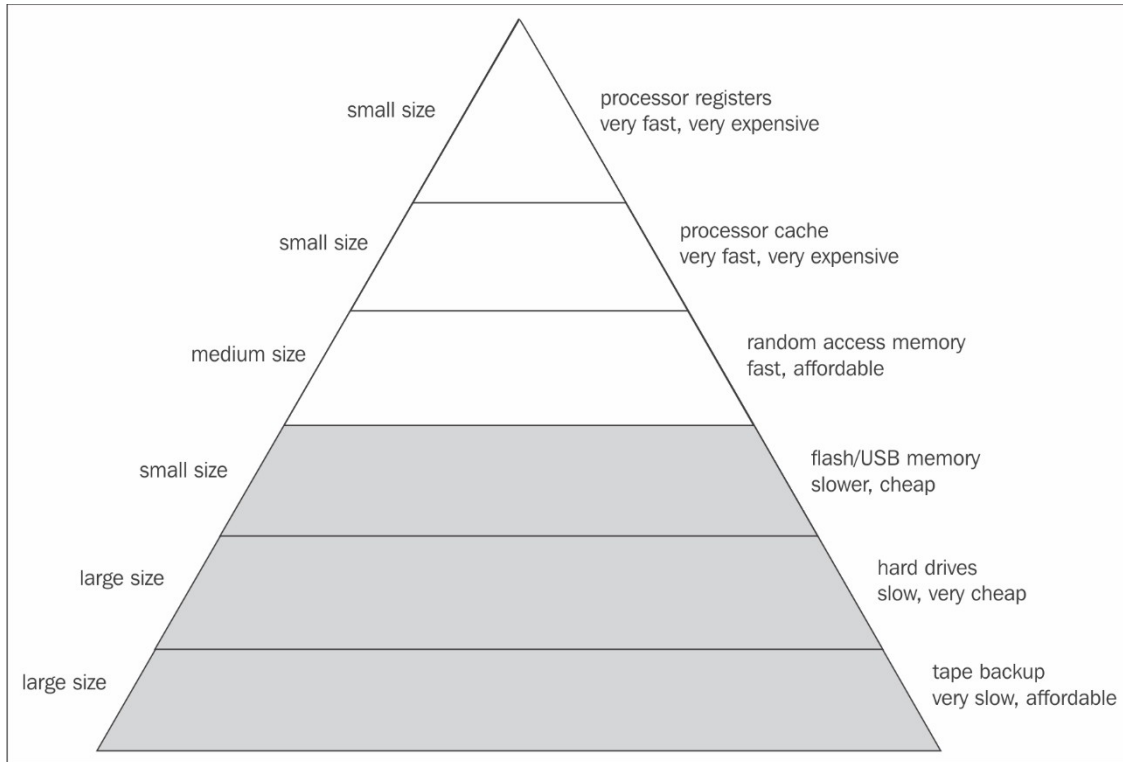


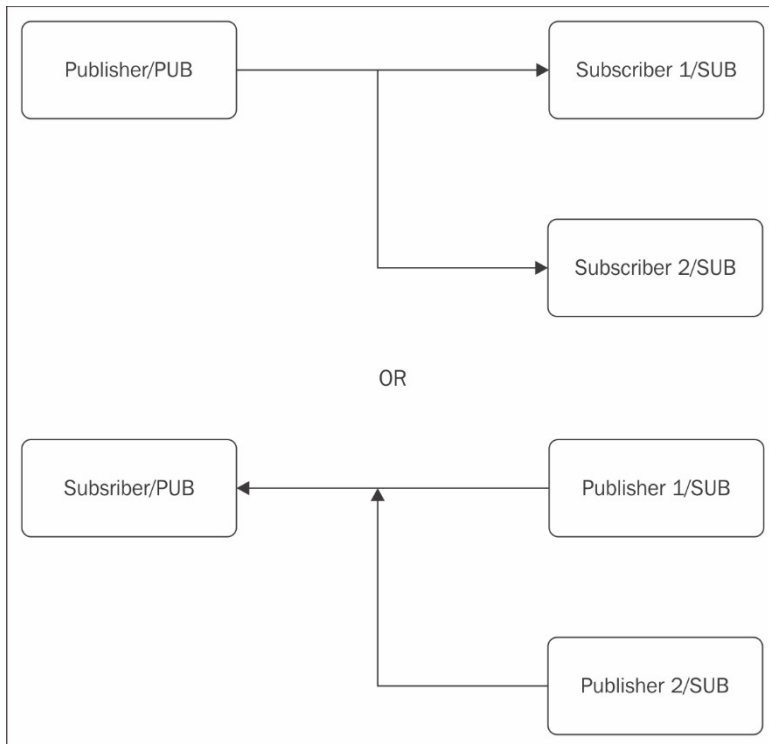
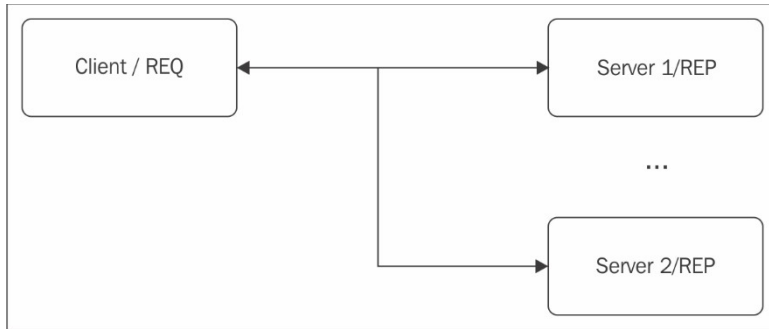
SIMD

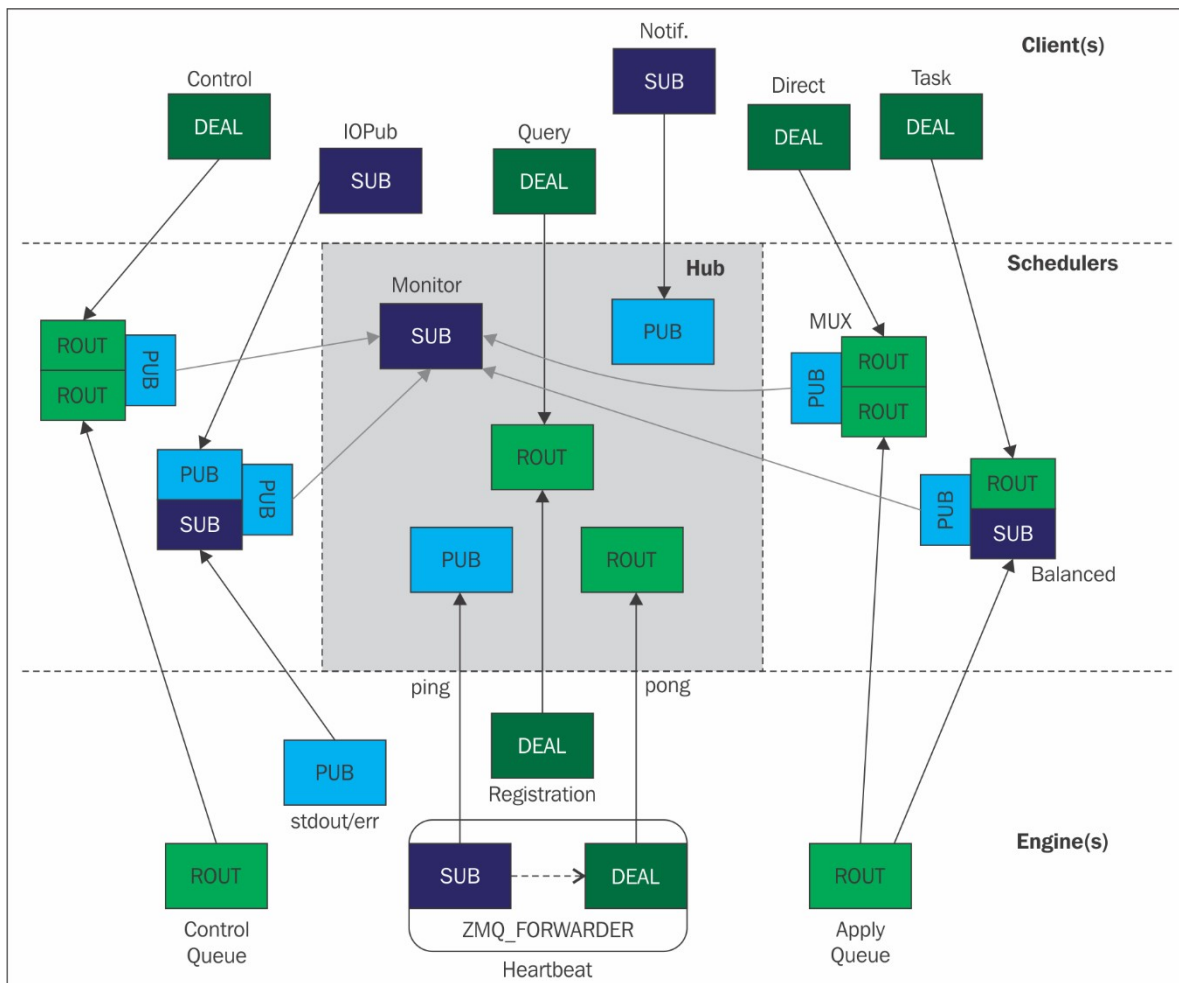
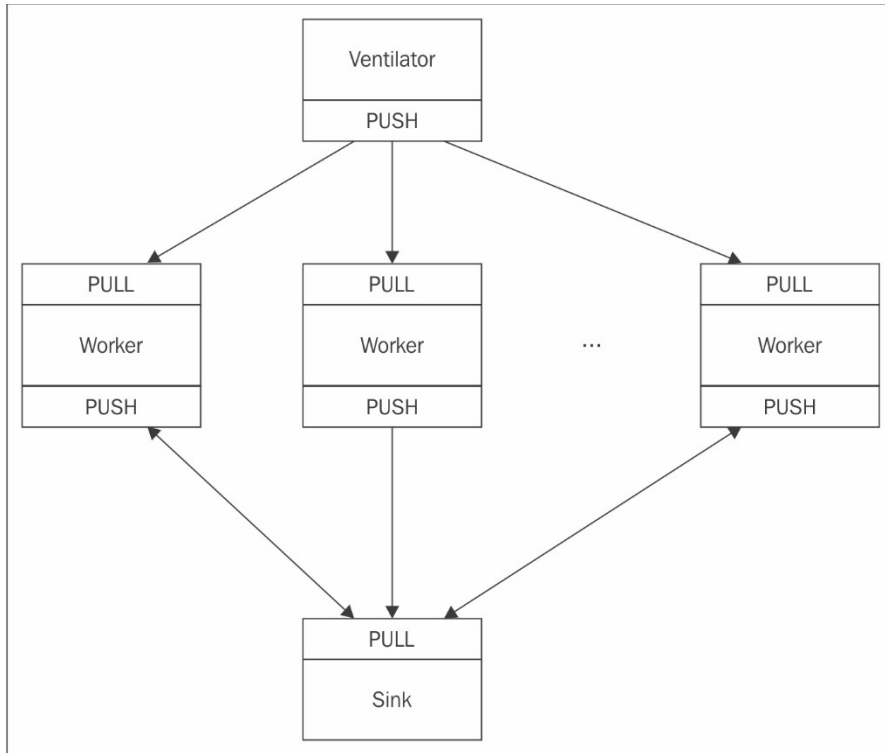


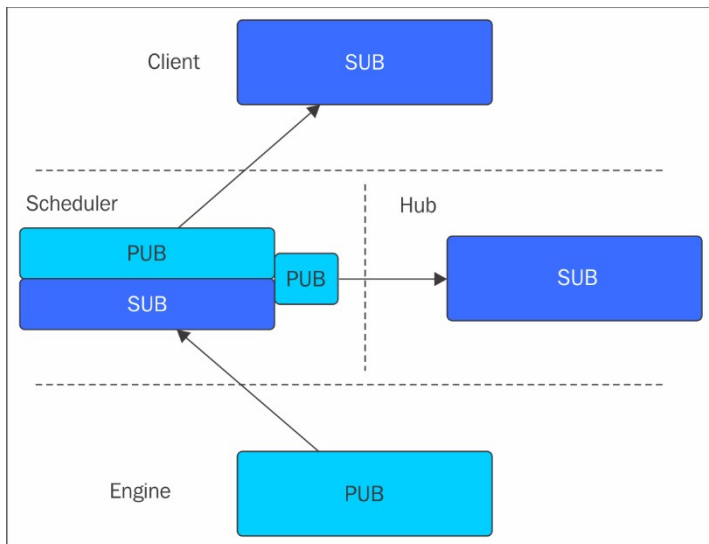
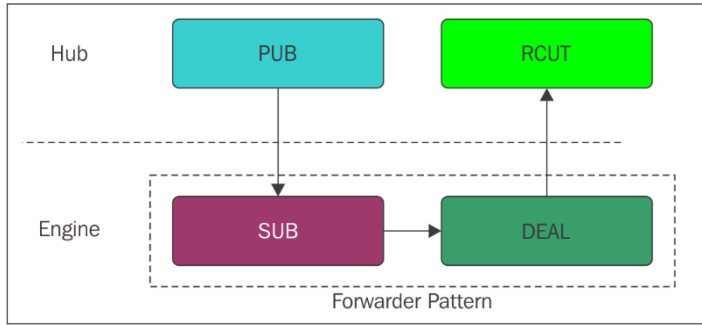
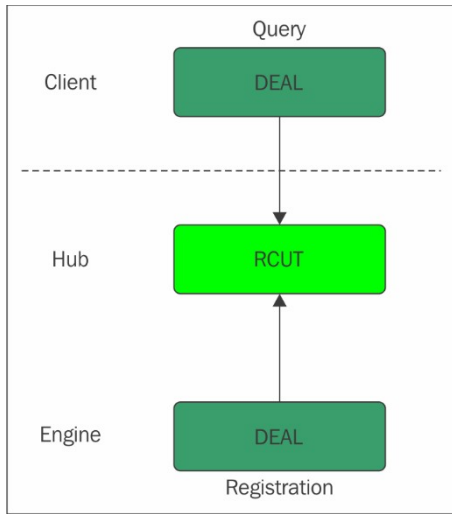
Chapter

4: Messaging with ZeroMQ and MPI

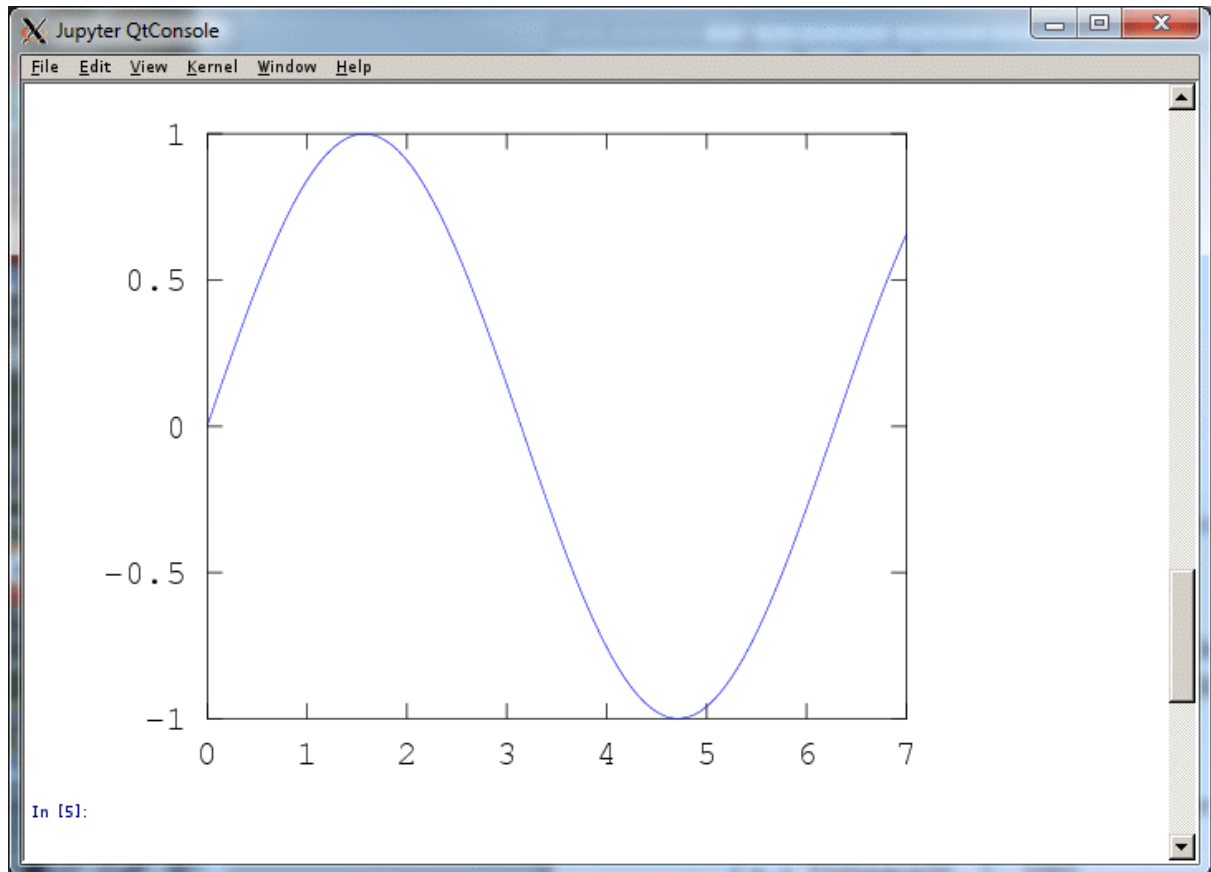


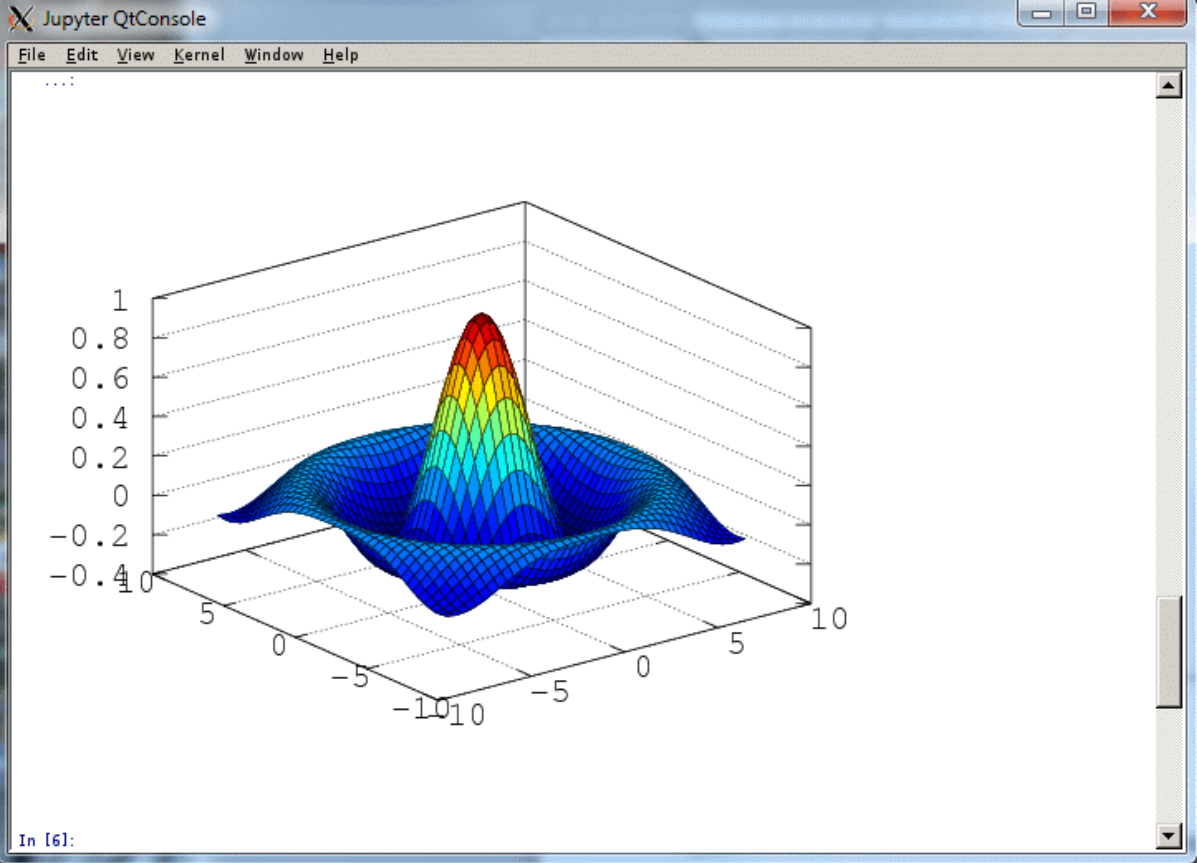


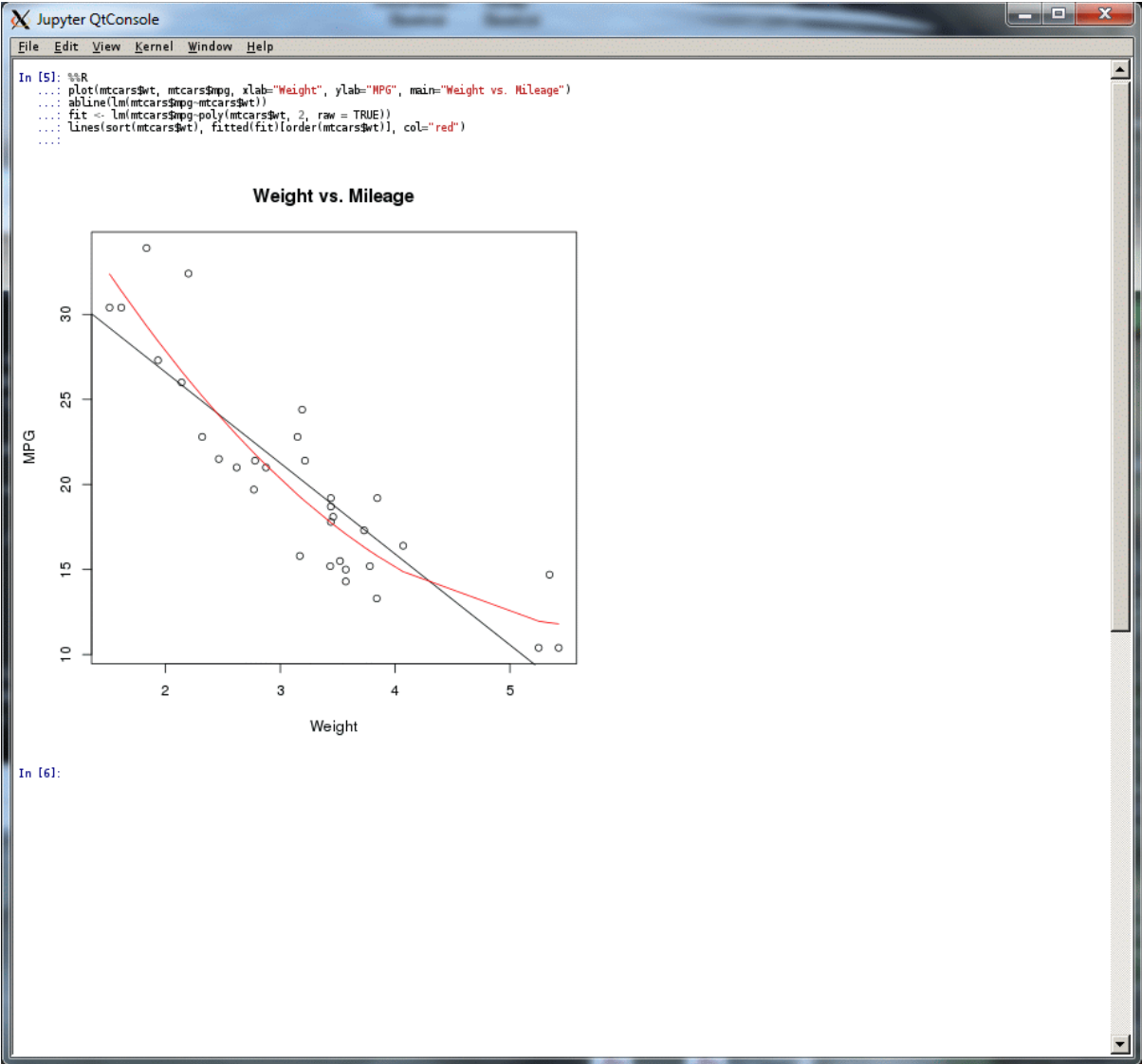


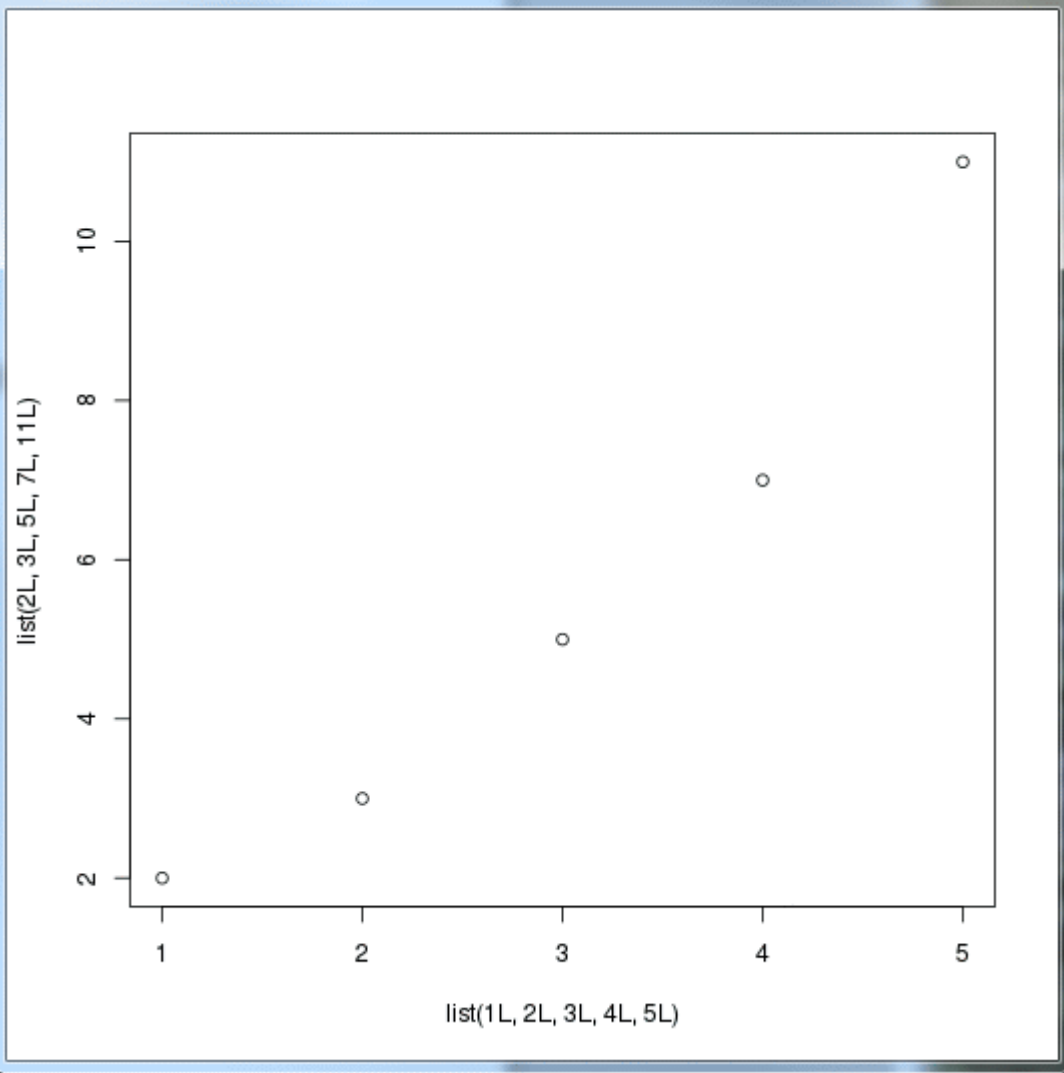


Chapter 6: Works Well with Others - IPython and Third-Party Tools









Chapter 7: Seeing Is Believing- Visualization

```
dipanjana@dipanjana-K53SD: ~
dipanjana@dipanjana-K53SD:~$ ipython --matplotlib
Python 2.7.11 |Anaconda 2.3.0 (64-bit)| (default, Dec 6 2015, 18:08:32)
Type "copyright", "credits" or "license" for more information.

IPython 3.2.0 -- An enhanced Interactive Python.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.
GLib-GIO-Message: Using the 'memory' GSettings backend. Your settings will not
be saved or shared with other applications.

(python:6628): Gtk-WARNING **: GModule (/usr/lib/x86_64-linux-gnu/gtk-2.0/2.10.0
/immodules/im-ibus.so) initialization check failed: GLib version too old (micro
mismatch)

(python:6628): Gtk-WARNING **: Loading IM context type 'ibus' failed

(python:6628): Gtk-WARNING **: GModule (/usr/lib/x86_64-linux-gnu/gtk-2.0/2.10.0
/immodules/im-ibus.so) initialization check failed: GLib version too old (micro
mismatch)

(python:6628): Gtk-WARNING **: Loading IM context type 'ibus' failed

(python:6628): Gtk-WARNING **: GModule (/usr/lib/x86_64-linux-gnu/gtk-2.0/2.10.0
/immodules/im-ibus.so) initialization check failed: GLib version too old (micro
mismatch)

(python:6628): Gtk-WARNING **: Loading IM context type 'ibus' failed
Using matplotlib backend: Qt4Agg

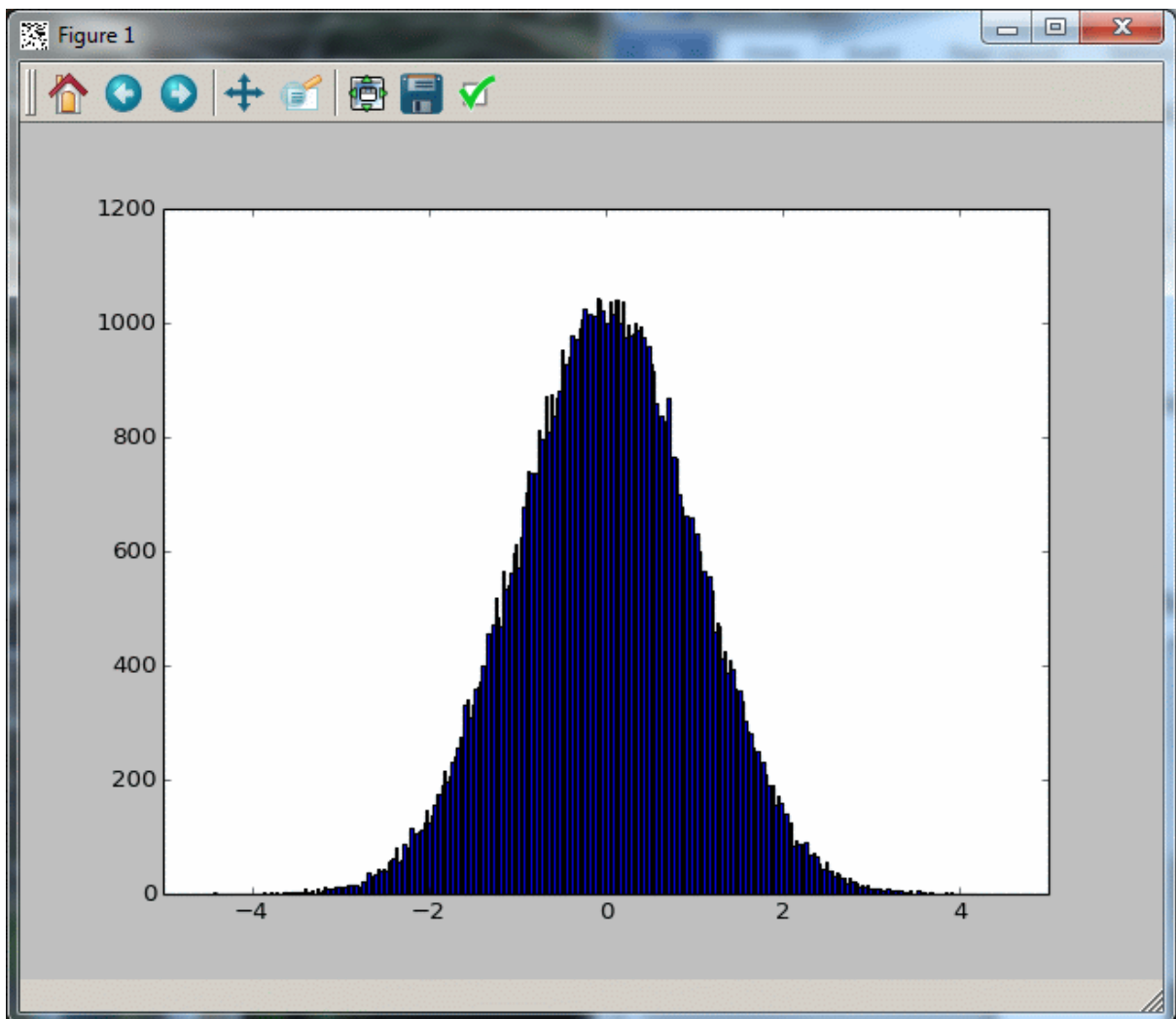
In [1]: █
```

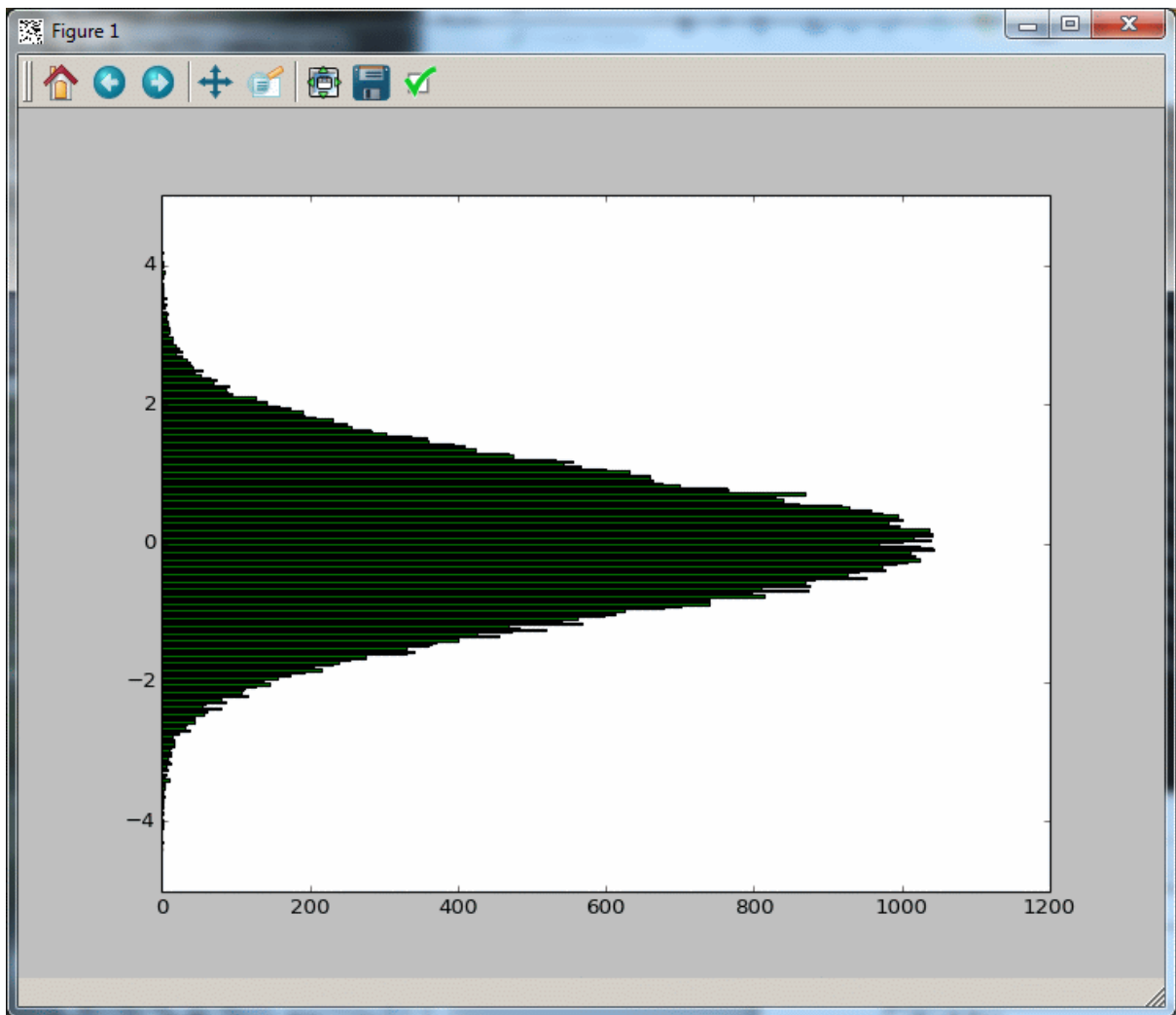
```
SmarTTY - oakley.osc.edu
File Edit View SCP Settings Help

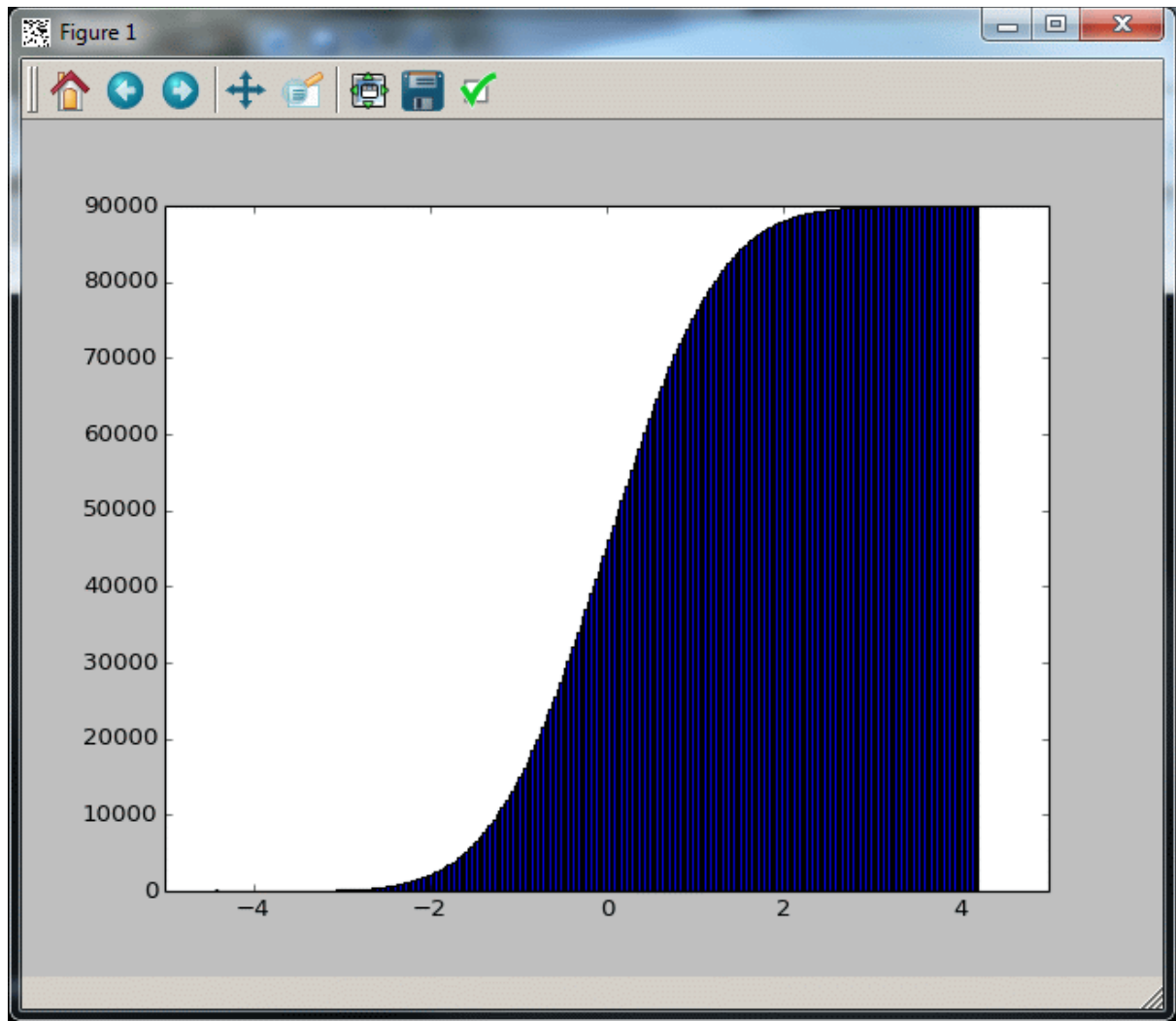
IPython 4.0.1 -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.

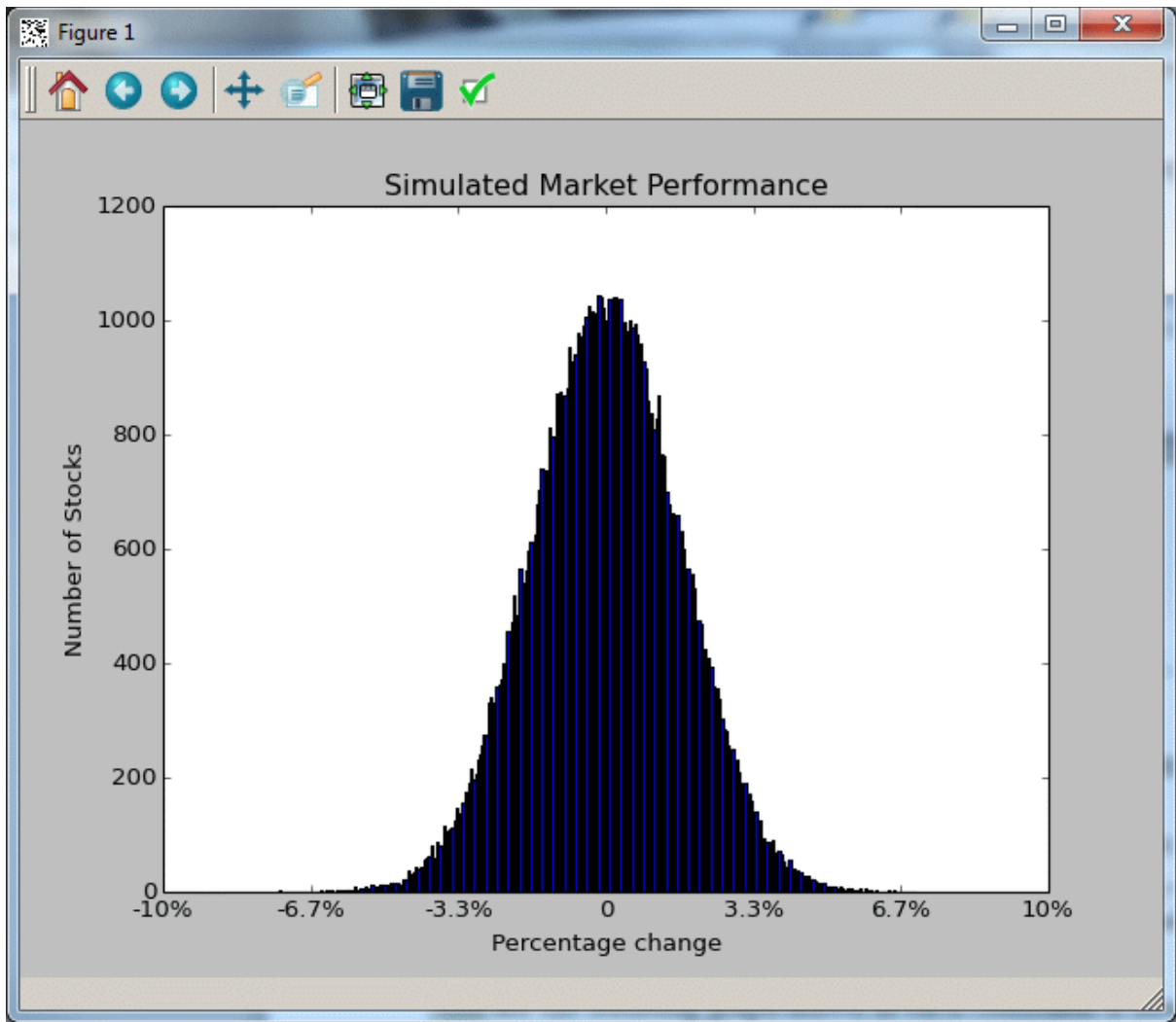
In [1]: import numpy
In [2]: import matplotlib
In [3]: from matplotlib import pylab, mlab, pyplot
In [4]: np = numpy
In [5]: plt = pyplot
In [6]: from IPython.display import display
In [7]: from IPython.core.pylabtools import figsize, getfigs
In [8]: from pylab import *
In [9]: from numpy import *
In [10]:

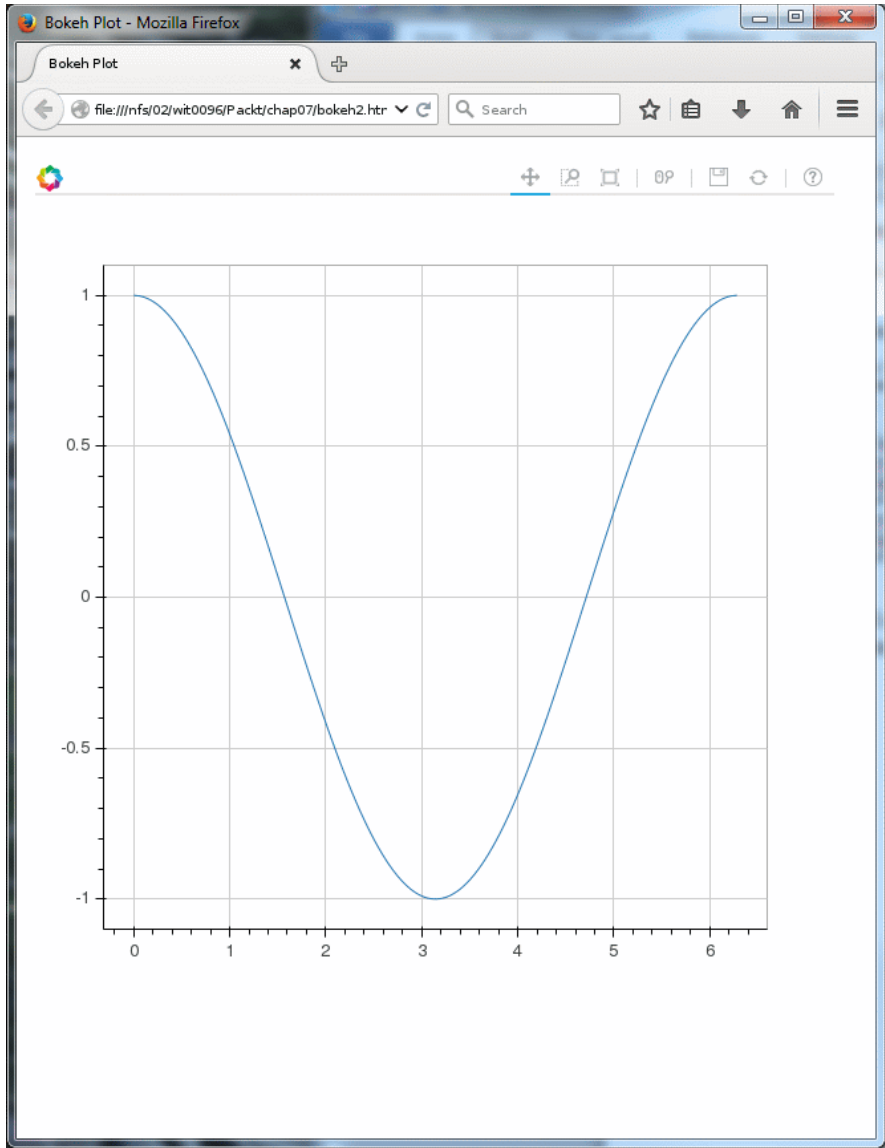
wit0096@oakley.osc.edu
SCP: No transfers 39KB sent, 23KB received
```

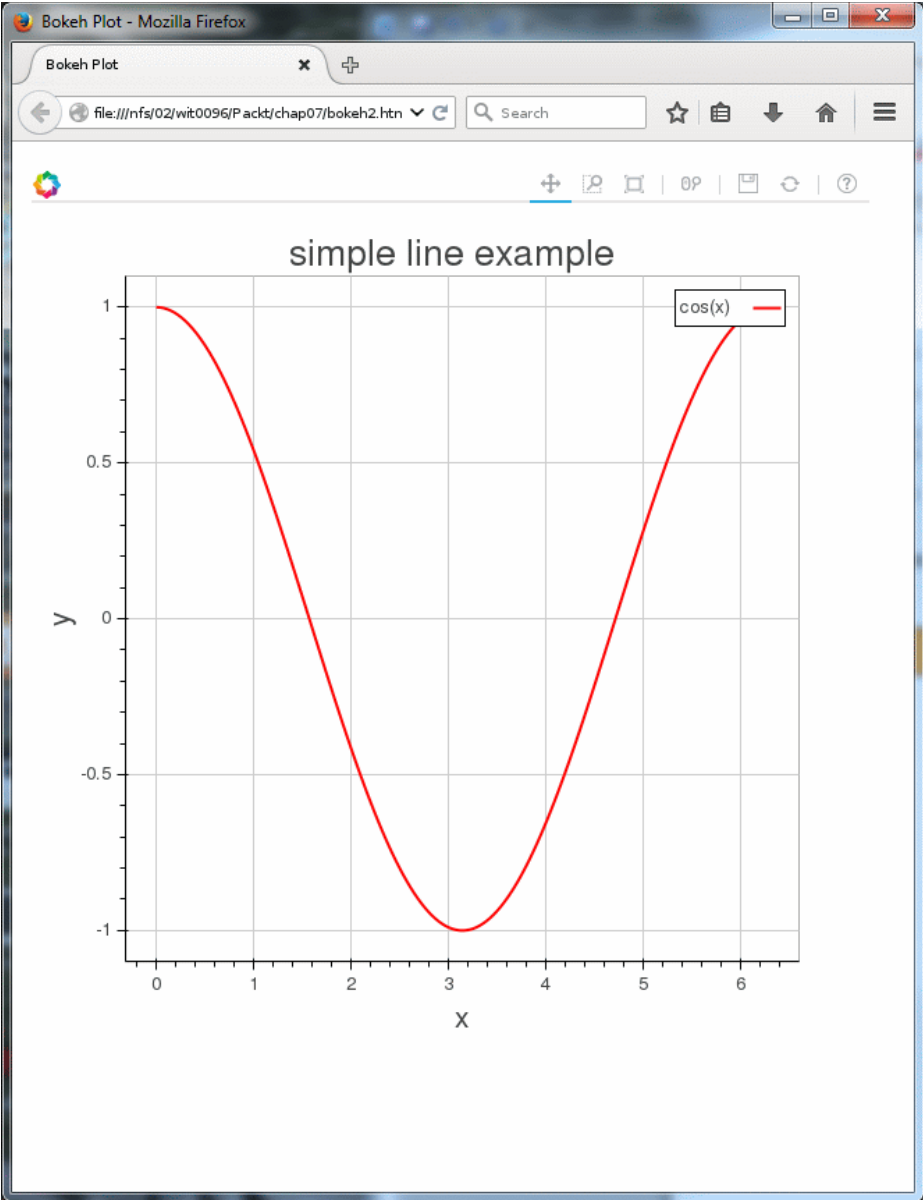


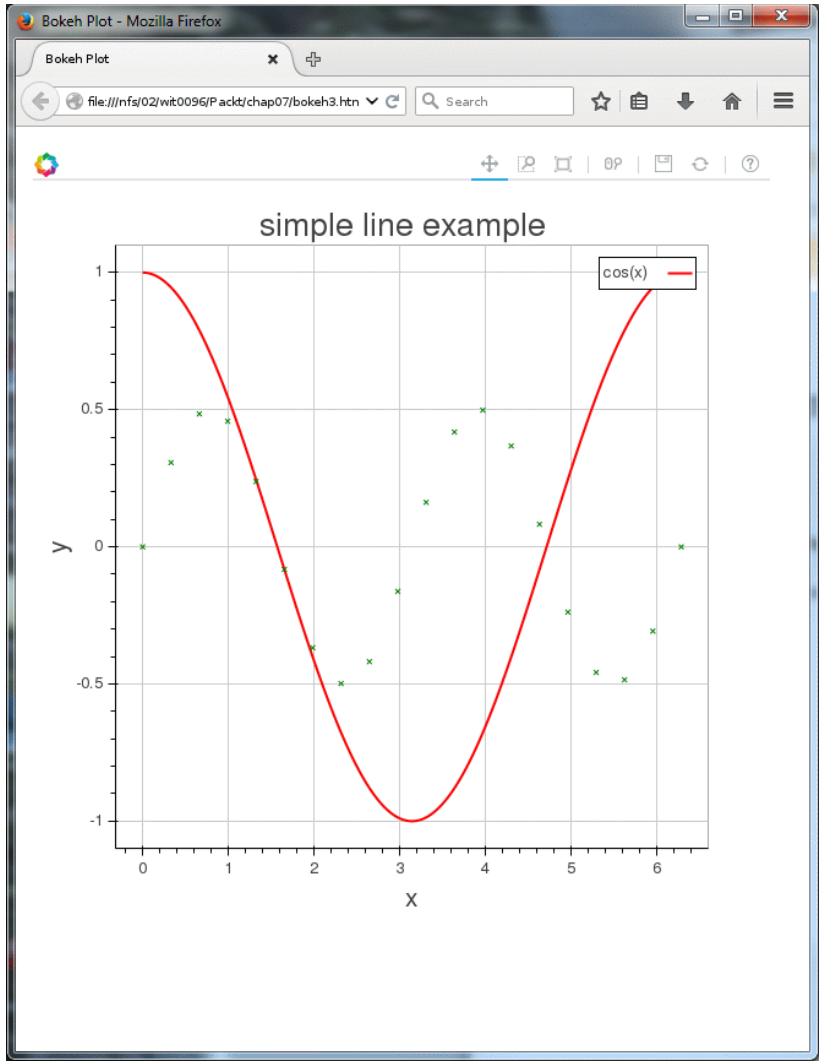


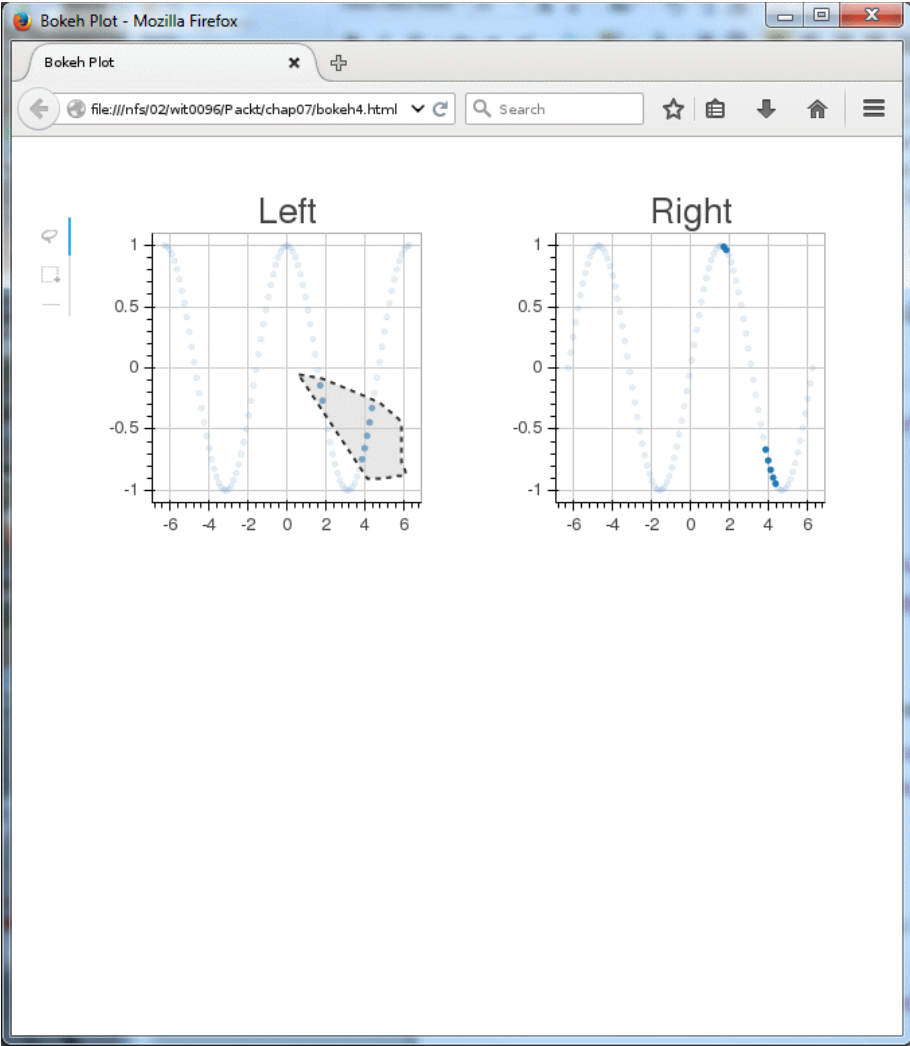


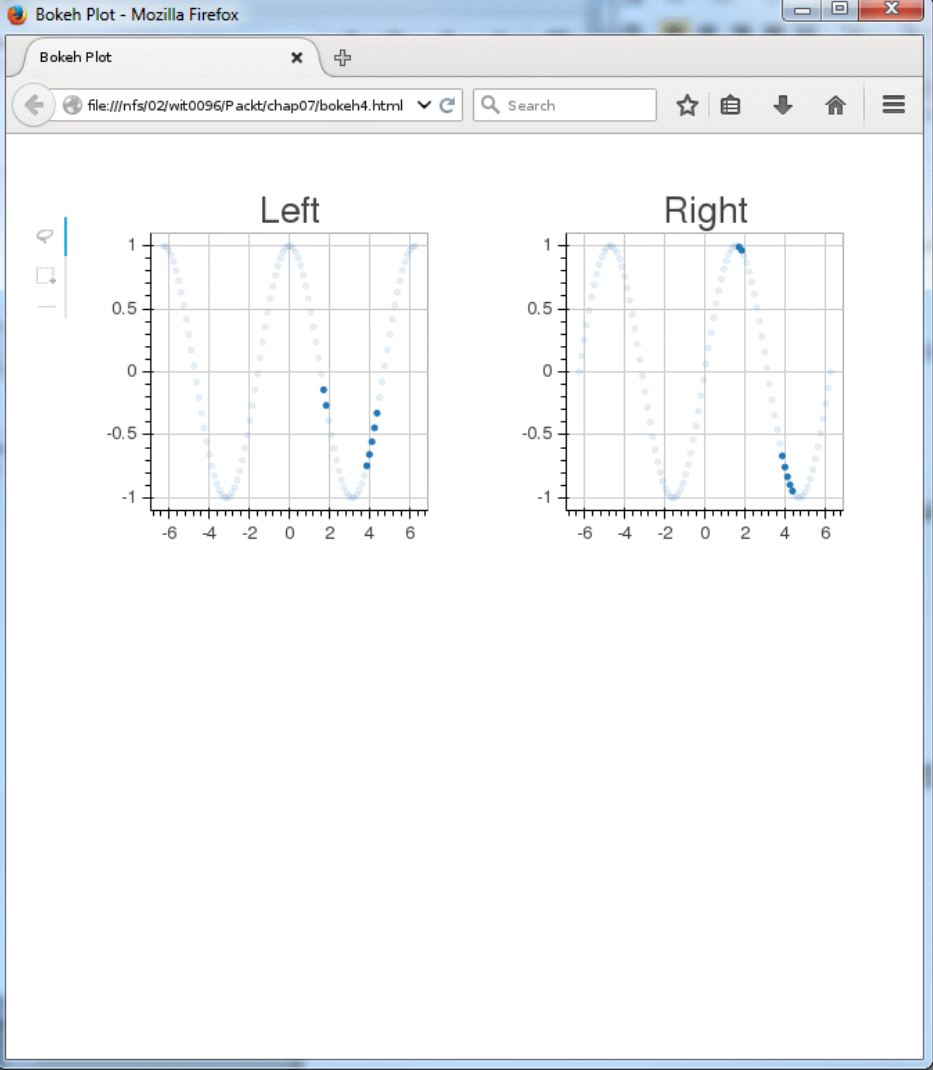


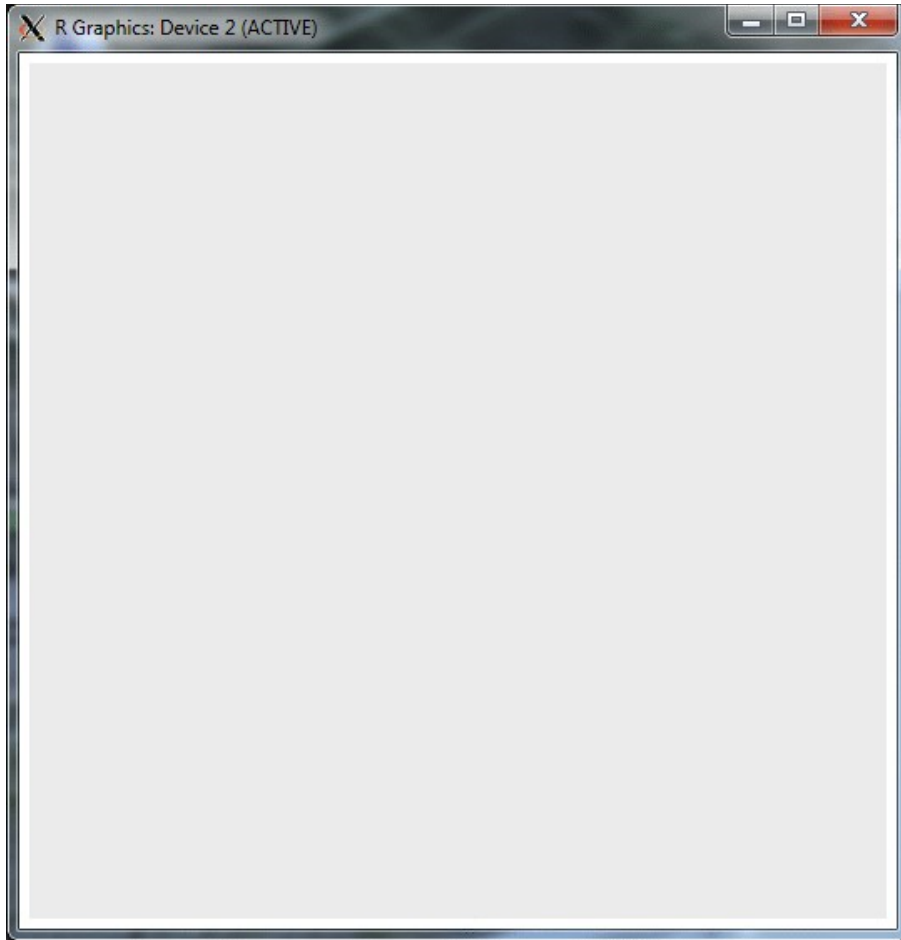


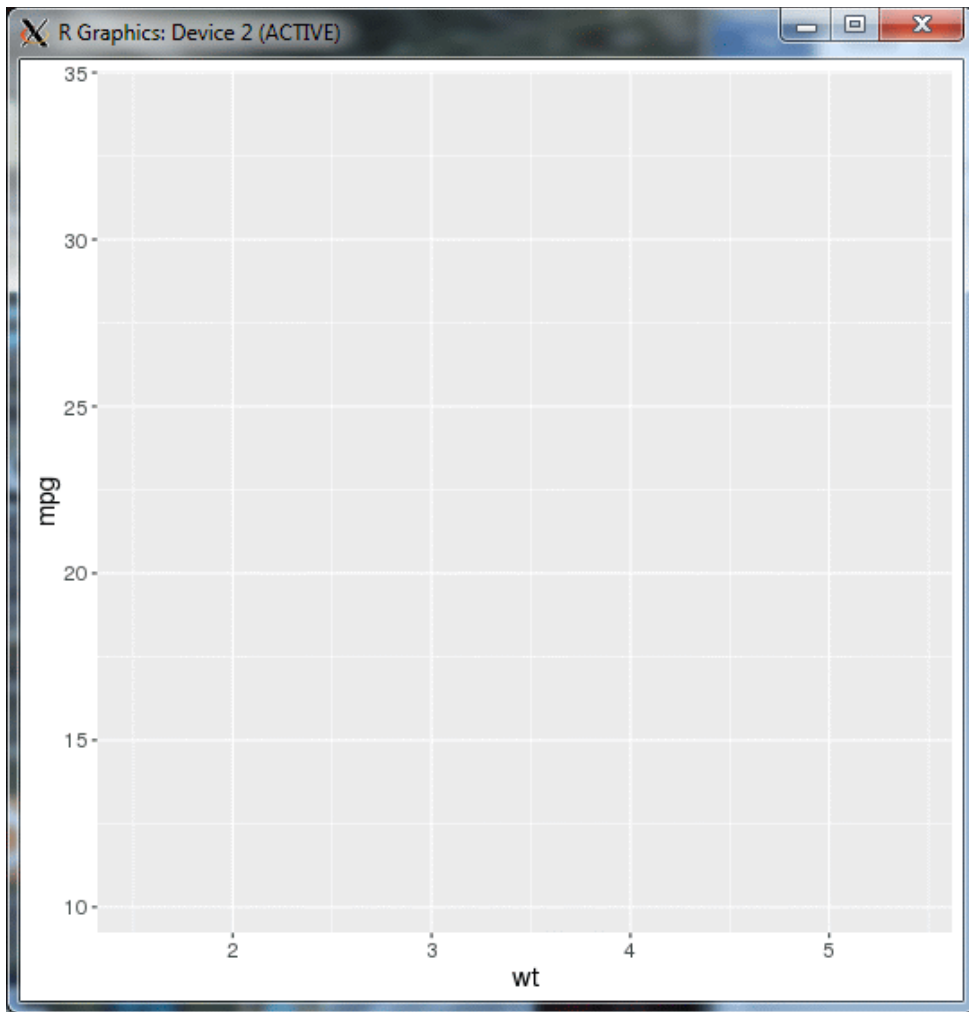


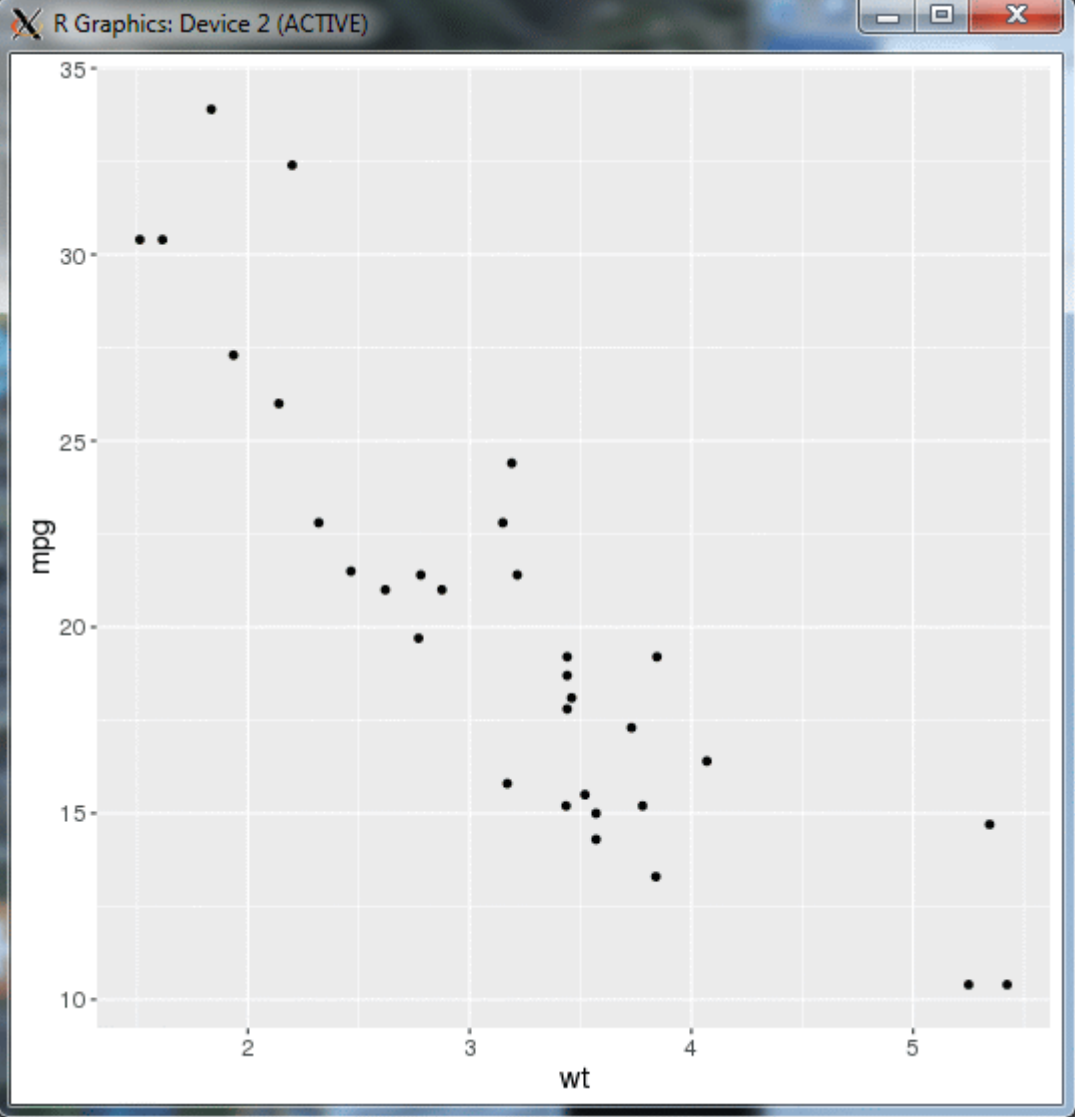


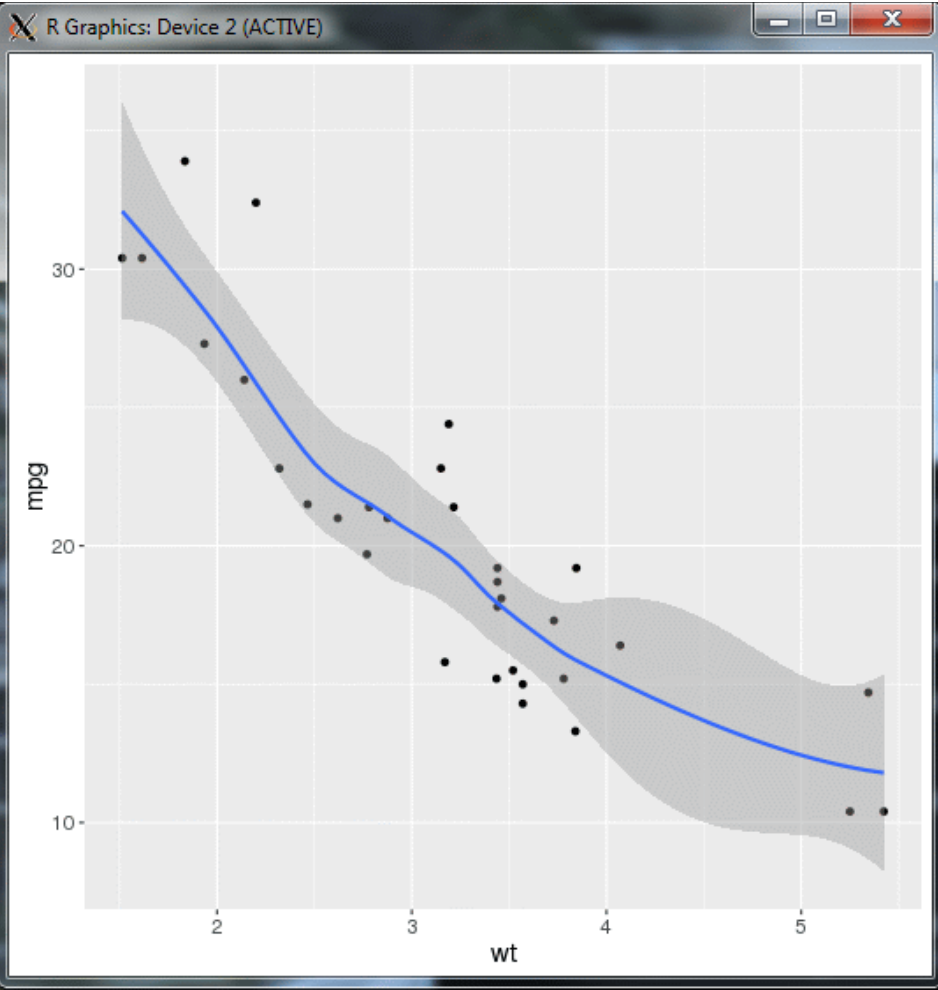


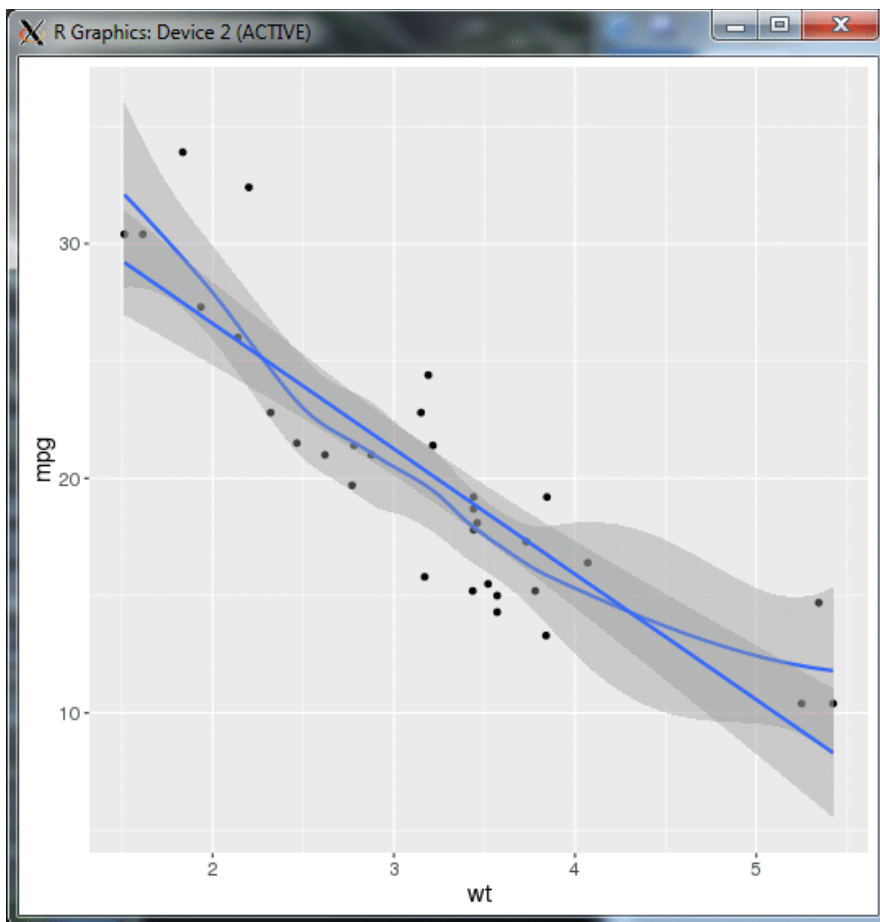


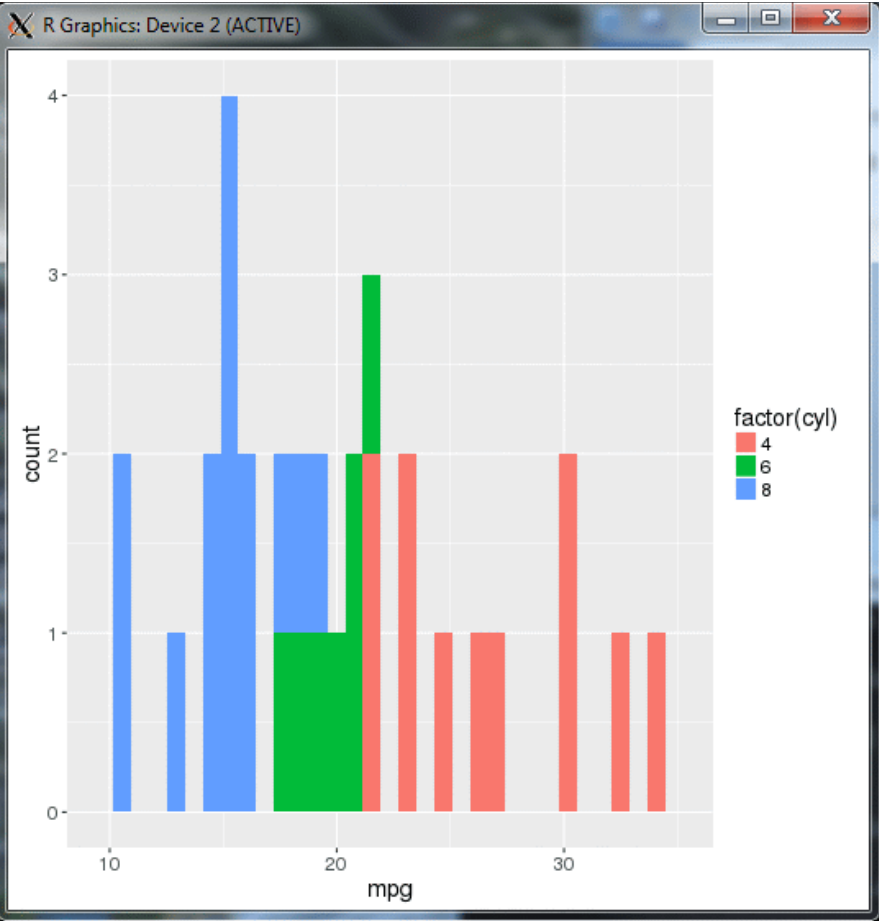


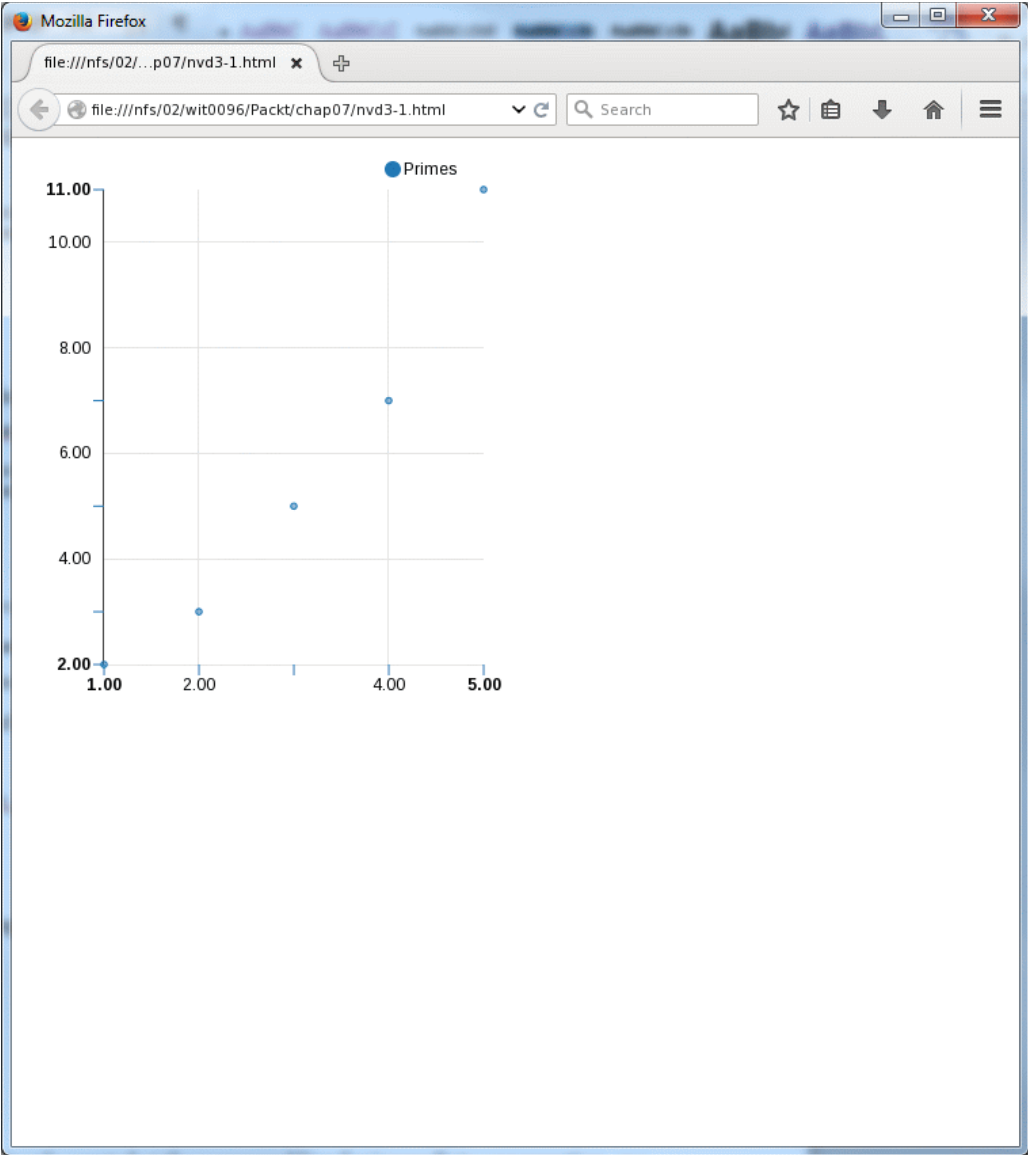


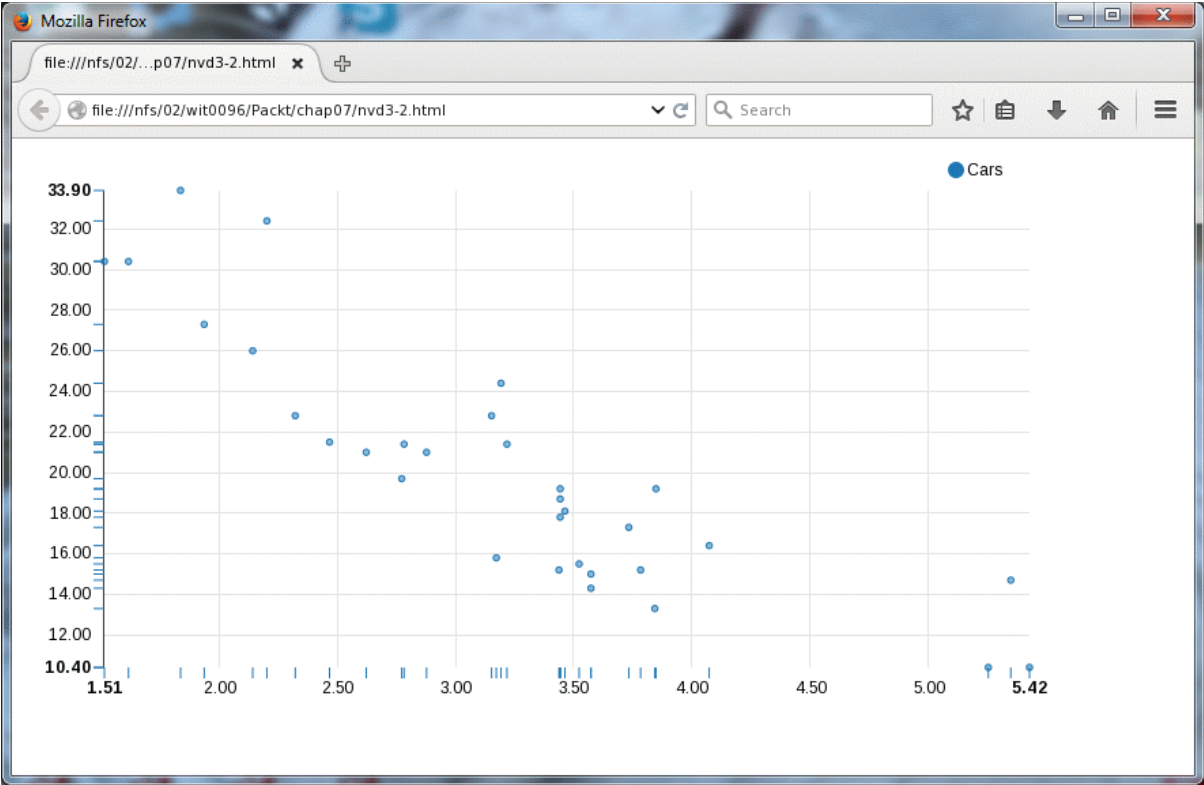


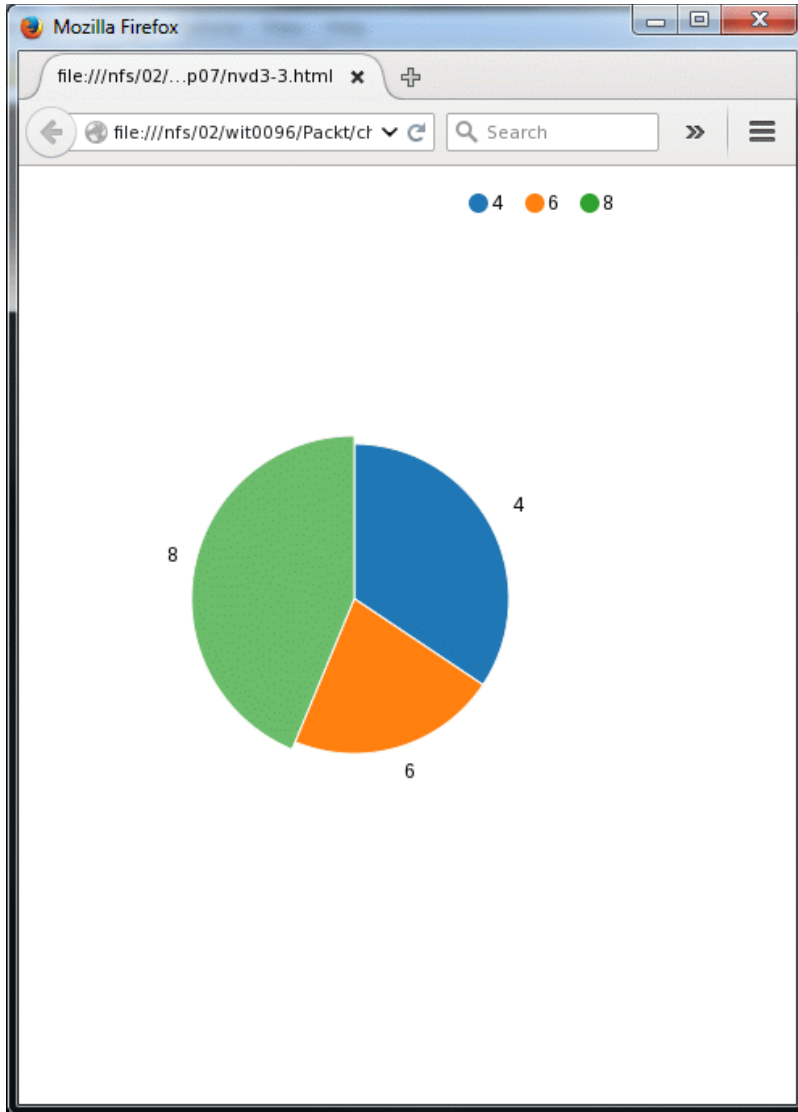












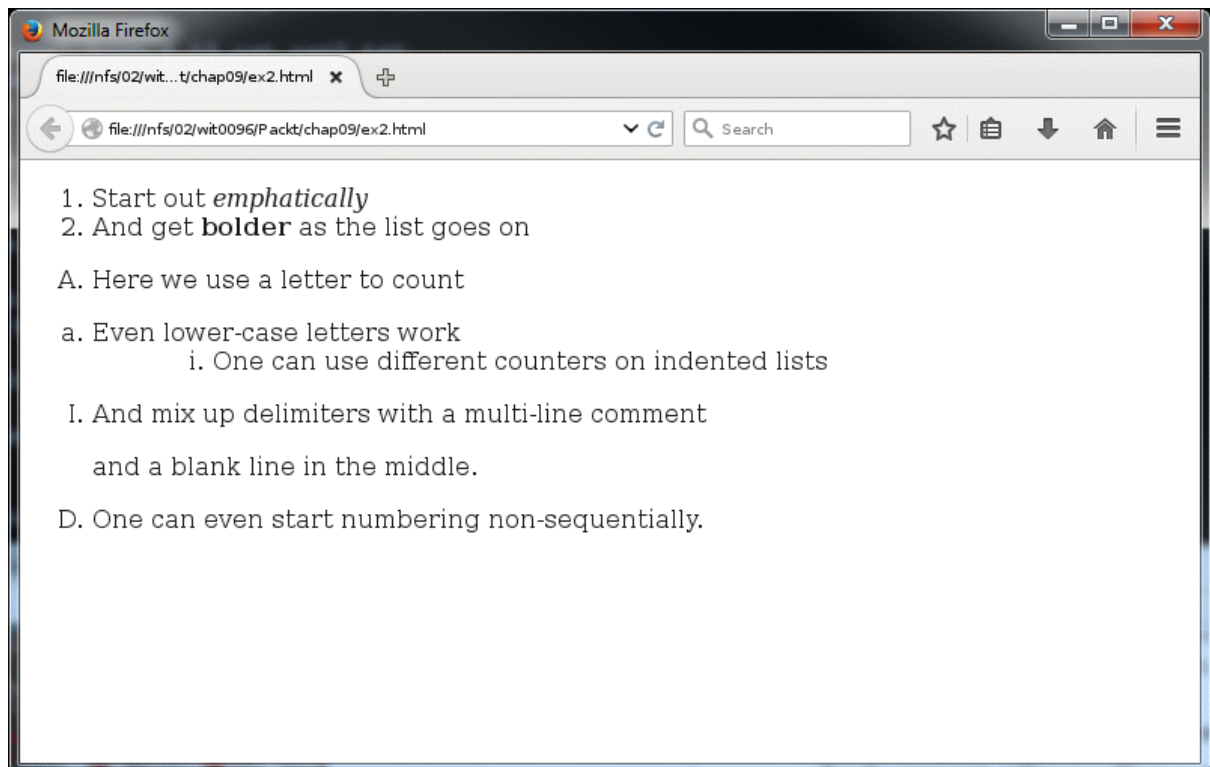
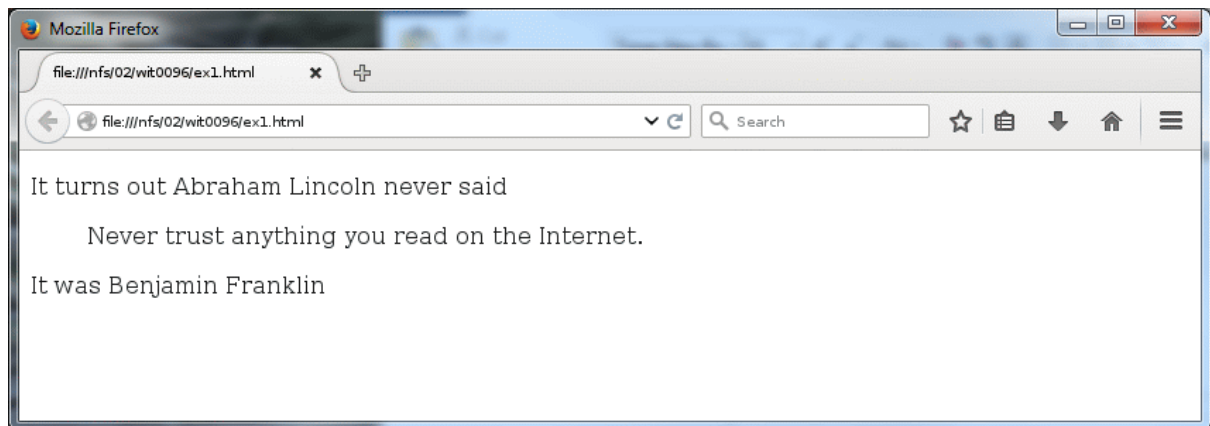
```
SmarTTY - oakley.osc.edu
File Edit View SCP Settings Help

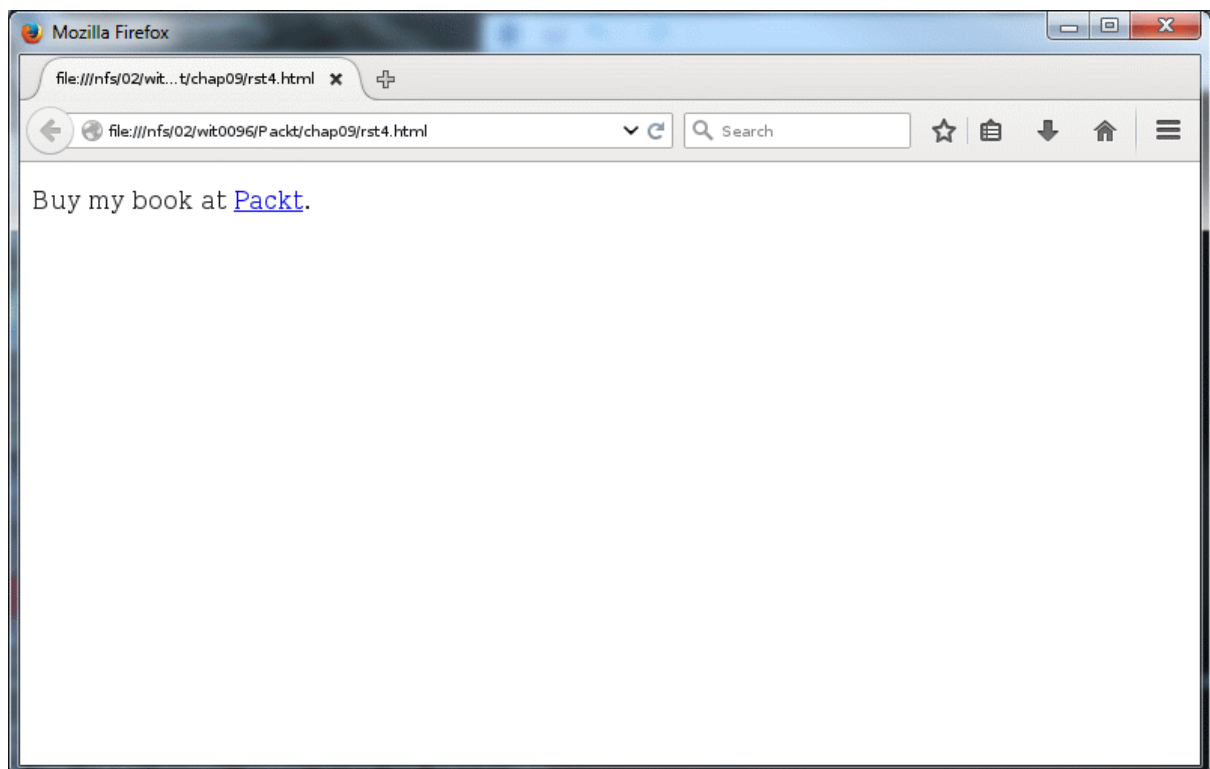
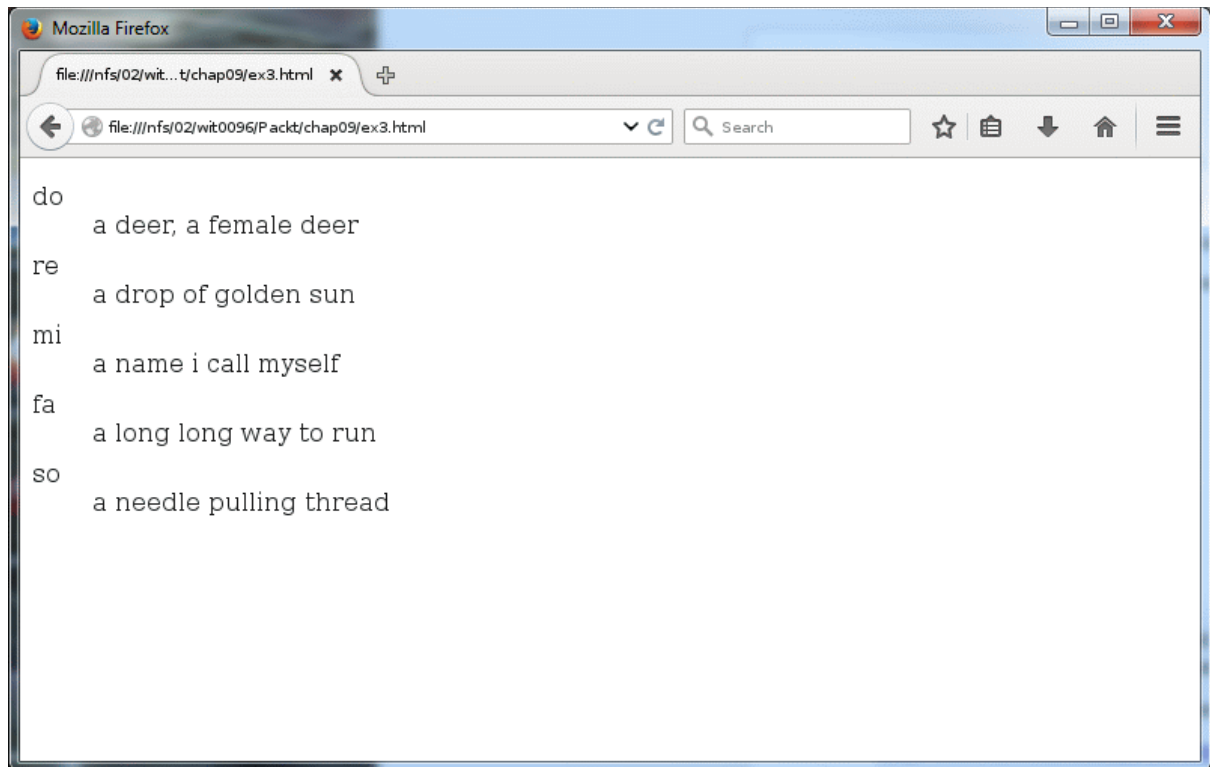
IPython 4.0.1 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.

In [1]: import numpy
In [2]: import matplotlib
In [3]: from matplotlib import pylab, mlab, pyplot
In [4]: np = numpy
In [5]: plt = pyplot
In [6]: from IPython.display import display
In [7]: from IPython.core.pylabtools import figsize, getfigs
In [8]: from pylab import *
In [9]: from numpy import *
In [10]:

wit0096@oakley.osc.edu
SCP: No transfers 39KB sent, 23KB received
```


Chapter 9: Documentation





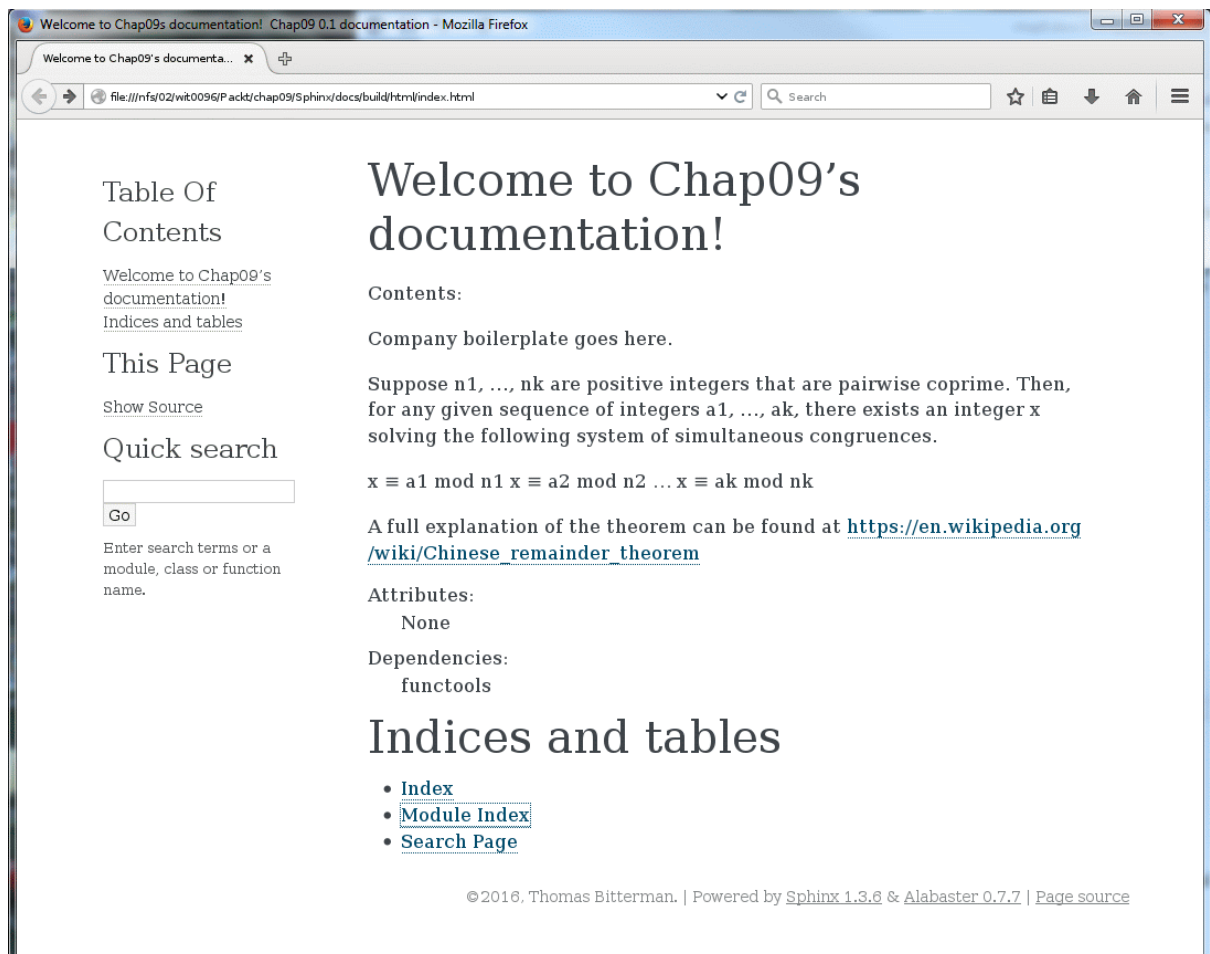
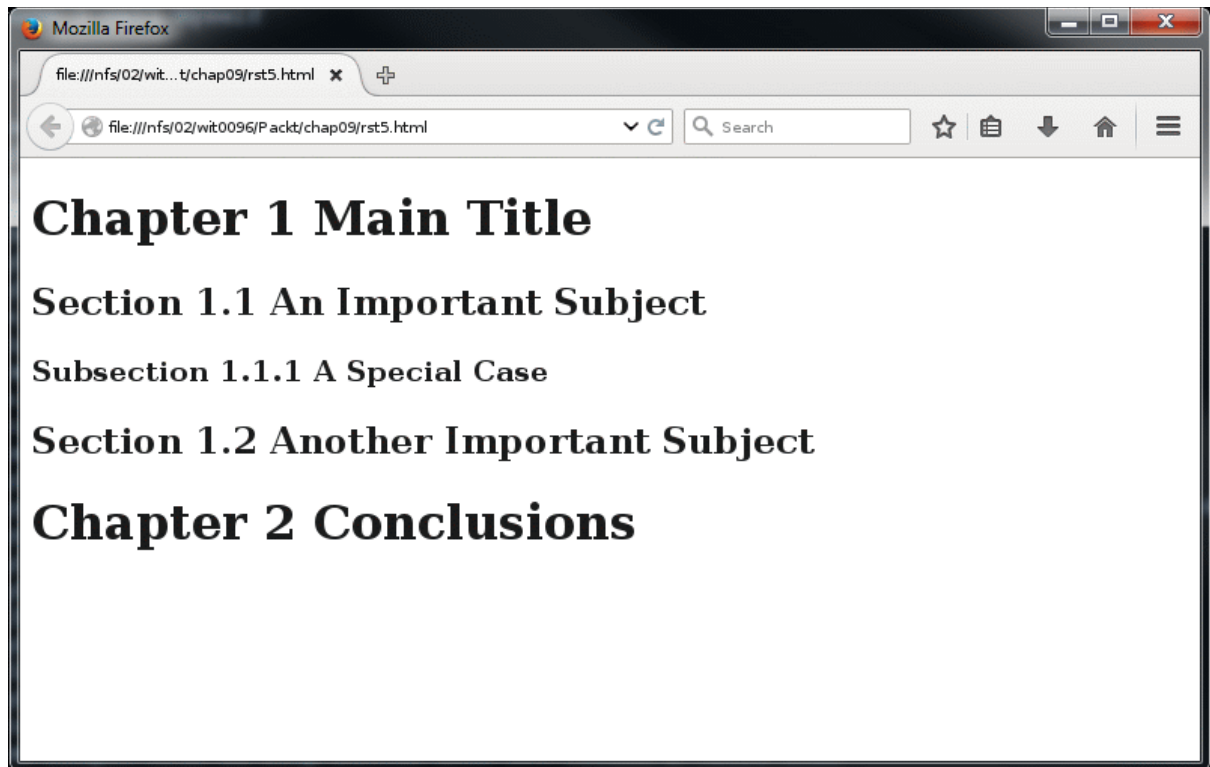


Table Of Contents

Welcome to Chap09's documentation!
[Indices and tables](#)

This Page

[Show Source](#)

Quick search

Enter search terms or a module, class or function name.

Welcome to Chap09's documentation!

Contents:

Company boilerplate goes here.

Suppose n_1, \dots, n_k are positive integers that are pairwise coprime. Then, for any given sequence of integers a_1, \dots, a_k , there exists an integer x solving the following system of simultaneous congruences.

$$x \equiv a_1 \pmod{n_1} \quad x \equiv a_2 \pmod{n_2} \quad \dots \quad x \equiv a_k \pmod{n_k}$$

A full explanation of the theorem can be found at https://en.wikipedia.org/wiki/Chinese_remainder_theorem

Attributes:

None

Dependencies:

`functools`

`class crt1.CRT`

The Chinese Remainder Theorem method for solving a system of linear congruences.

Notes:

This class has no `__init__` method. As such, there are no arguments required to construct an instance.

Without class or instance variables, all objects of this class are stateless.

Args:

None

Attributes:

None

Example:

An example is provided in the main guard corresponding to the system: $x \equiv 2 \pmod{3}$ $x \equiv 3 \pmod{5}$ $x \equiv 2 \pmod{7}$

Indices and tables

- [Index](#)
- [Module Index](#)
- [Search Page](#)

Table Of Contents

Welcome to Chap09's documentation!
Indices and tables

This Page

Show Source

Quick search

Go

Enter search terms or a module, class or function name.

Welcome to Chap09's documentation!

Contents:

Company boilerplate goes here.

Suppose n_1, \dots, n_k are positive integers that are pairwise coprime. Then, for any given sequence of integers a_1, \dots, a_k , there exists an integer x solving the following system of simultaneous congruences.

$$x \equiv a_1 \pmod{n_1} \quad x \equiv a_2 \pmod{n_2} \quad \dots \quad x \equiv a_k \pmod{n_k}$$

A full explanation of the theorem can be found at https://en.wikipedia.org/wiki/Chinese_remainder_theorem

Attributes:

None

Dependencies:

functools

class crt1.**CRT**

The Chinese Remainder Theorem method for solving a system of linear congruences.

Notes:

This class has no `__init__` method. As such, there are no arguments required to construct an instance.

Without class or instance variables, all objects of this class are stateless.

Args:

None

Attributes:

None

Example:

An example is provided in the main guard corresponding to the system: $x \equiv 2 \pmod{2}$ $x \equiv 3 \pmod{3}$ $x \equiv 5 \pmod{5}$ $x \equiv 2 \pmod{7}$

CRT.**chinese_remainder**(*n*, *a*)

Use the existence construction form of the CRT to compute the solution.

First, calculate the product of all the modulus (b_1, n_2, \dots, n_k) as *prod*
For each *i*, calculate $prod/n_i$ as *p* Sum up each $a_i * \text{the multiplicative inverse of } n_i \pmod{p}$

Args:

n: a list of modulus (n_1, n_2, \dots, n_k) *a*: a list of congruences (a_1, a_2, \dots, a_k)

Returns:

The smallest integer solution to the system of congruence equations defined by *a* and *n*.

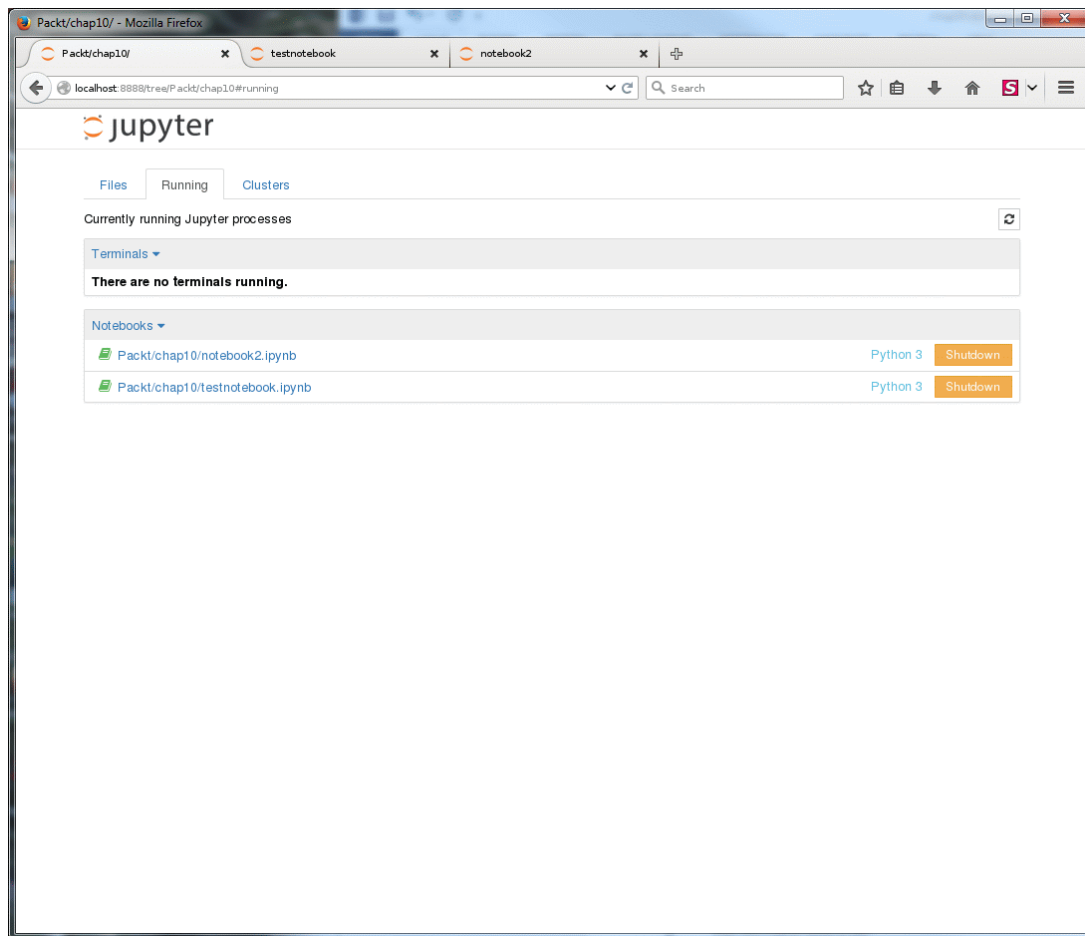
Error conditions:

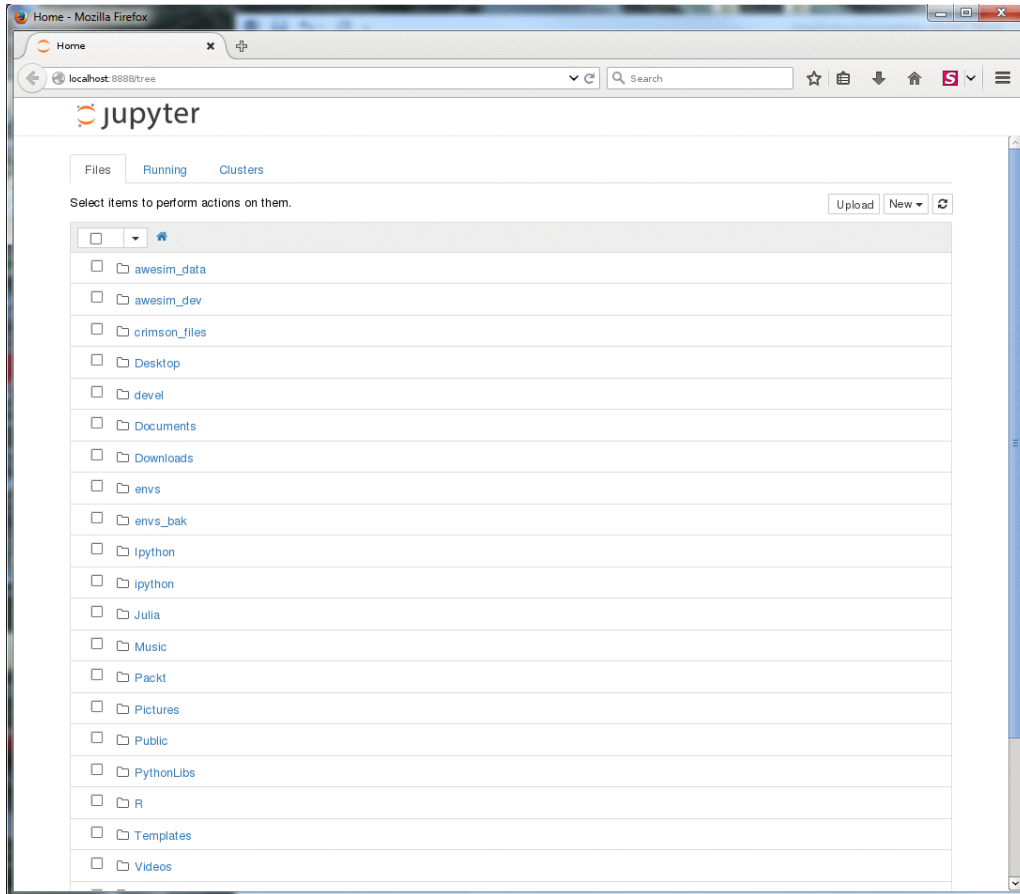
When $\text{len}(n) \neq \text{len}(a)$, a solution always exists. When $\text{len}(a) > \text{len}(n)$, any additional *a*'s are ignored. When $\text{len}(n) > \text{len}(a)$, behavior is deterministic but erroneous.

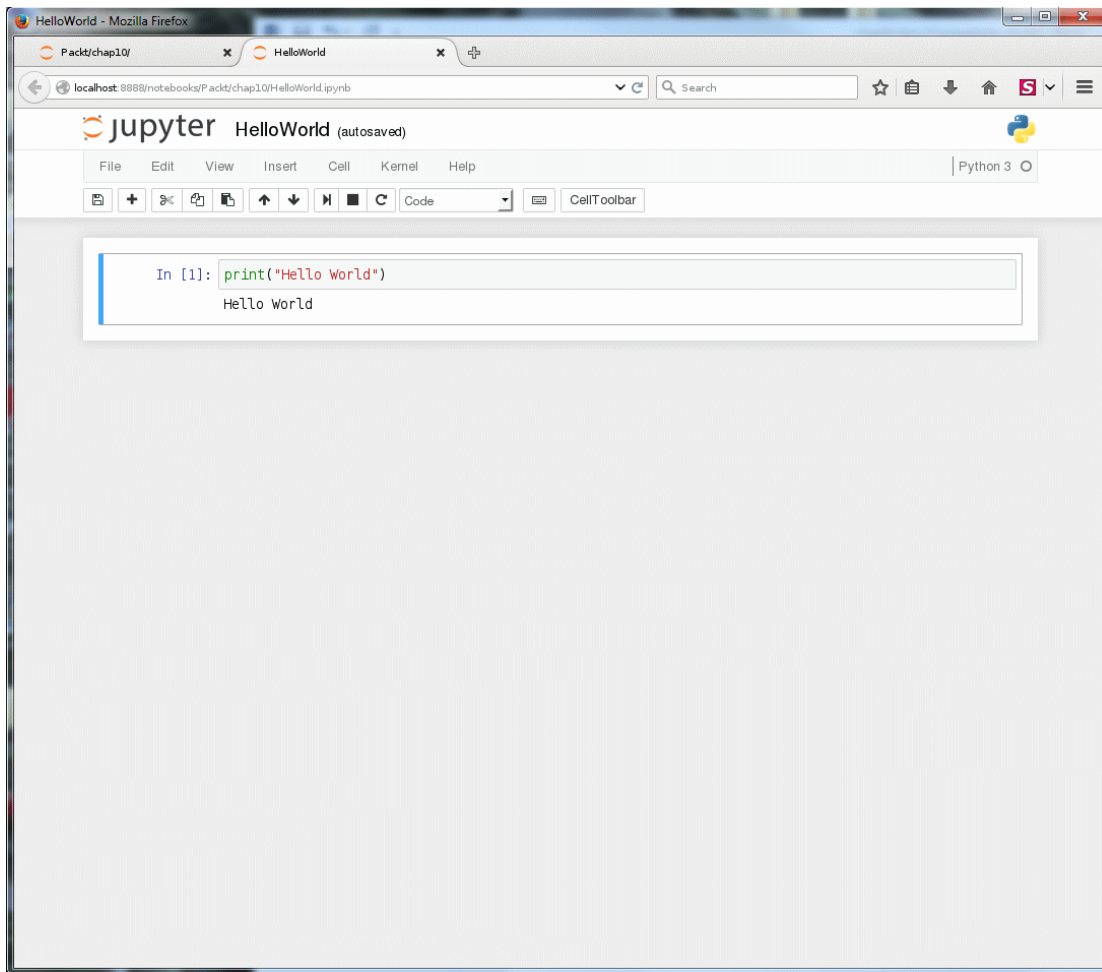
Indices and tables

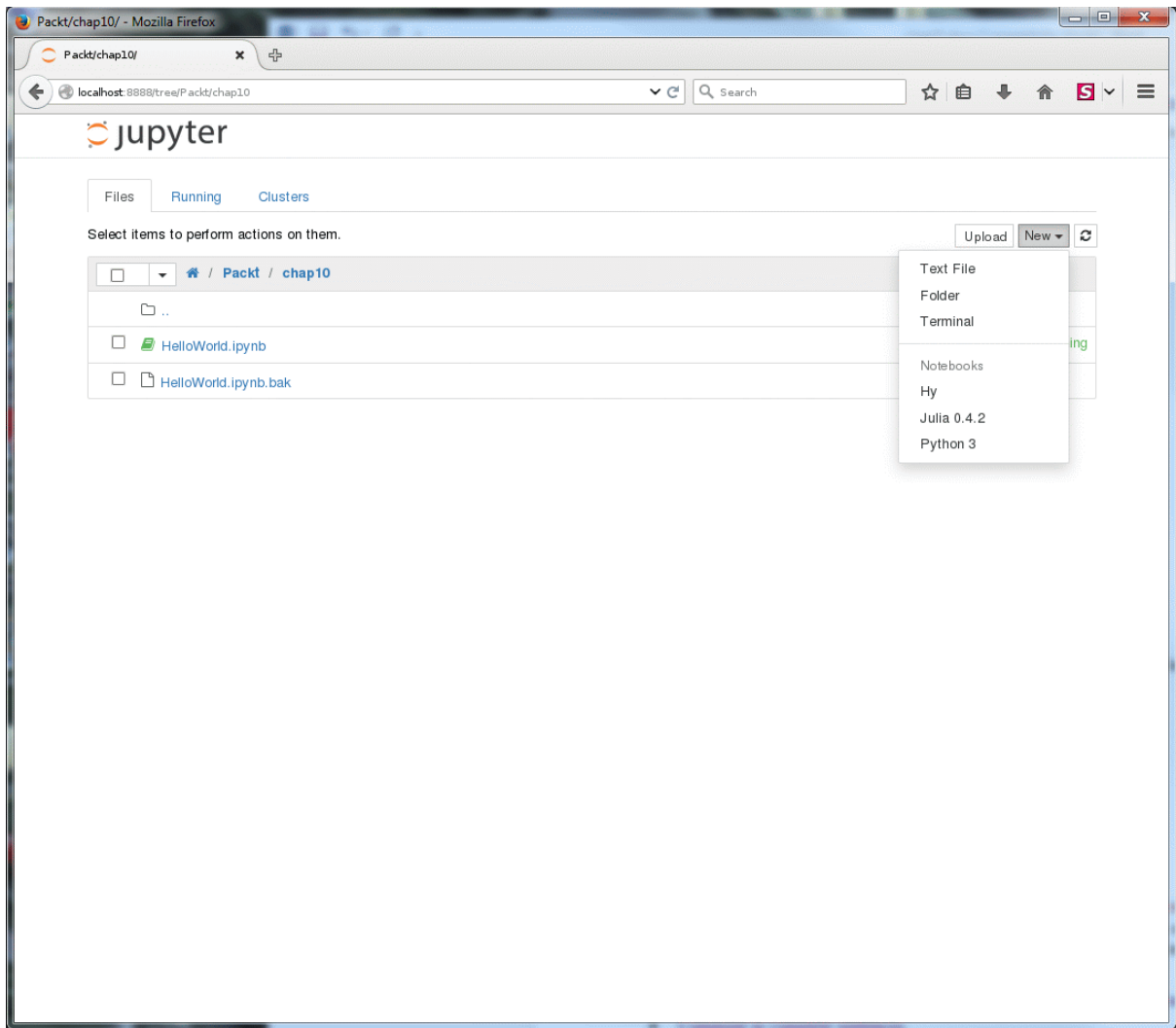
- [Index](#)
- [Module Index](#)
- [Search Page](#)

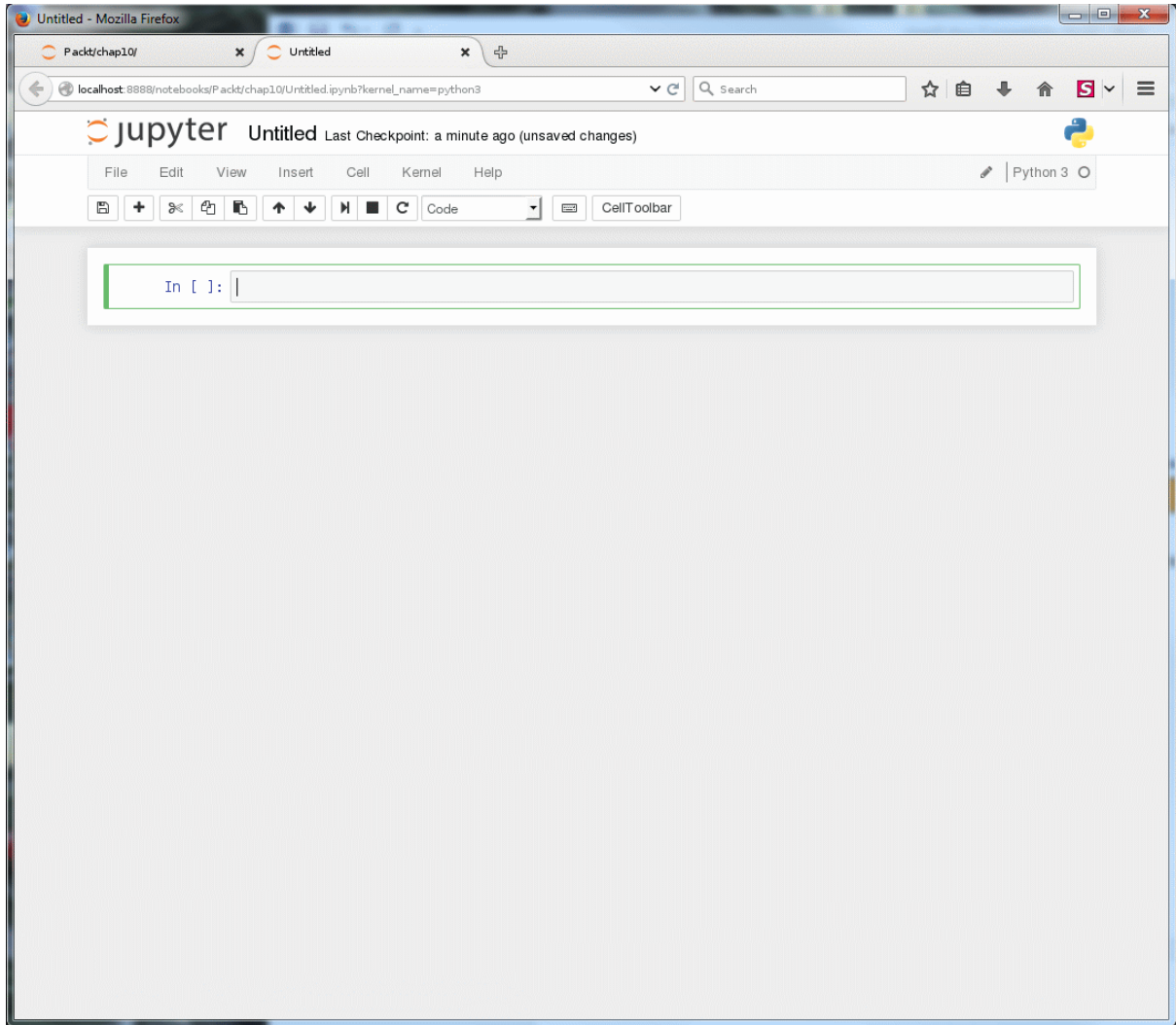
Chapter 10: Visiting Jupyter











testnotebook - Mozilla Firefox

Packt/chap10/ x testnotebook x Packt/chap09/ x notebook2 x

localhost:8888/notebooks/Packt/chap10/testnotebook.ipynb#

Search

jupyter testnotebook Last Checkpoint: Last Friday at 4:09 PM (unsaved changes)

File Edit View Insert Cell Kernel Help Python 3

Code CellToolbar

```
In [2]: # %load ../chap09/code1.py
"""
This is an abbreviated version of my random number generator test suite.
It uses the pytest framework. It does not do much in this form.
"""

import numpy as np
import scipy.stats
import random

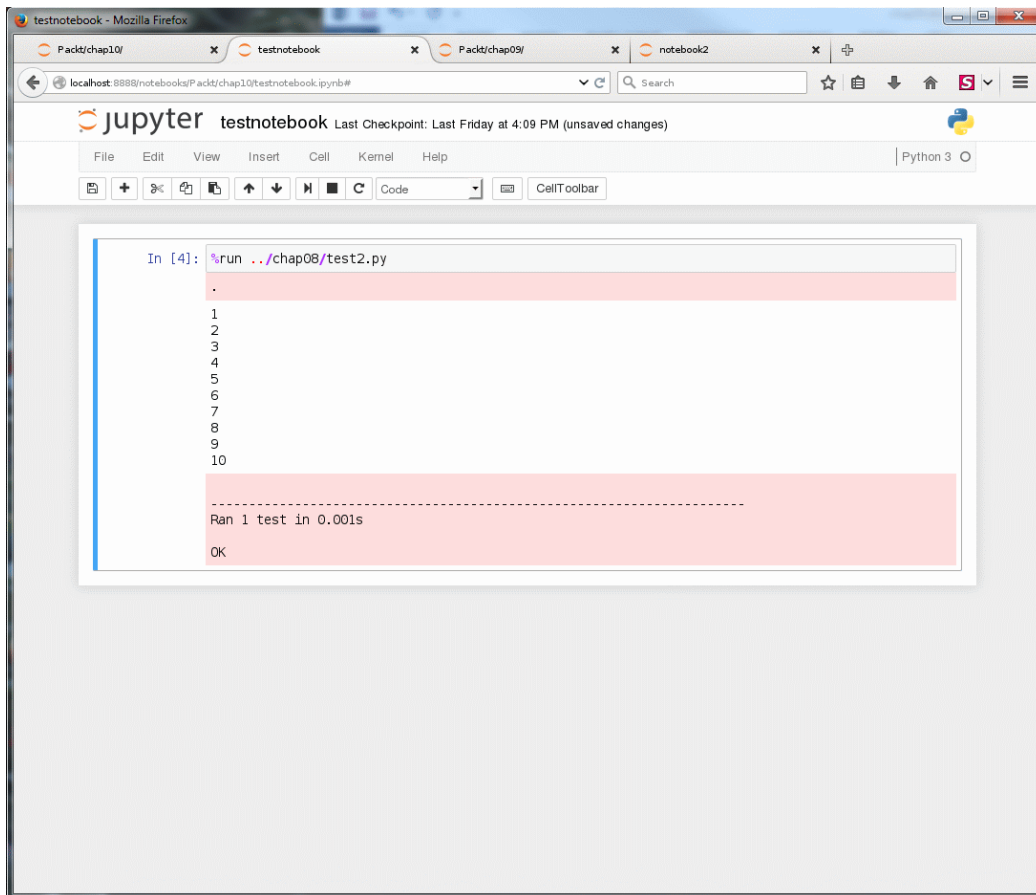
class TestRandoms( ):
    """
    This is the main class.

    Normally it would hold all the tests, plus and setup and teardown fixtures.
    """
    def test_builtin(self):
        """
        Test the built-in random number generator on 10000 numbers.
        """
        num_tests = 10000
        vals = [0 for i in range(10)]
        for i in range(num_tests):
            tmp = random.randint(0, 9)
            vals[tmp] = vals[tmp] + 1

        chi2, p = scipy.stats.chisquare(self.vals)
        assert p > 0.05

def foo():
    """ I just needed a function outside of a class as an example"""
    pass
```

In []:



testnotebook - Mozilla Firefox

localhost:8888/notebooks/Packt/chap10/testnotebook.ipynb#

jupyter testnotebook Last Checkpoint: an hour ago (unsaved changes)

File Edit View Insert Cell Kernel Help Python 3

Markdown CellToolbar

```
In [1]: x = 5
```

```
In [2]: print(x * 2)
```

10

```
In [3]: print("Something is missing")
```

File "<ipython-input-3-1aead959fd3b>", line 1
print("Something is missing")
 ^
SyntaxError: EOL while scanning string literal

```
In [ ]: import time
```

```
for i in range(10):
```

```
    print(i)
```

```
    time.sleep(1)
```

Here is an example of Markdown

This was a bold move

testnotebook - Mozilla Firefox

Pack/chap10/ testnotebook

localhost:8888/notebooks/Packt/chap10/testnotebook.ipynb#

jupyter testnotebook Last Checkpoint: an hour ago (unsaved changes)

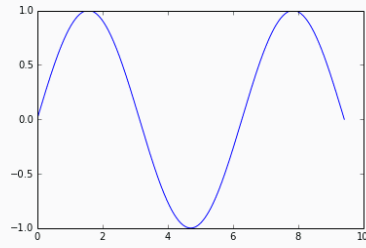
File Edit View Insert Cell Kernel Help Python 3

Code CellToolbar

Graphics are fun

```
In [1]: %matplotlib inline
import numpy
import matplotlib.pyplot as plt

x = numpy.linspace(0, 3*numpy.pi, 1000)
plt.plot(x, numpy.sin(x))
plt.show()
```



testnotebook - Mozilla Firefox

jupyter testnotebook Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Help Python 3

Code CellToolbar

```
In [7]: import numpy as np
import bokeh.plotting as bp
from bokeh.io import output_notebook

x = np.linspace(0, 2*np.pi, 1024)
y = np.cos(x)
fig = bp.figure()
fig.line(x, y)
output_notebook()
bp.show(fig)
```

BokehJS successfully loaded

Out[7]: <Bokeh Notebook handle for In[7]>

testnotebook - Mozilla Firefox

Packt/chap10/ testnotebook

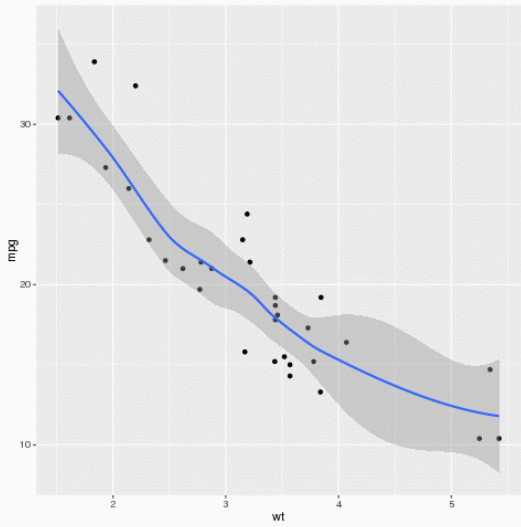
localhost:8888/notebooks/Packt/chap10/testnotebook.ipynb

jupyter testnotebook Last Checkpoint: 33 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Help Python 3

In [1]: `%load_ext rpy2.ipynthon`
.Rprofile: Setting UK repository

In [2]: `%%R
library(ggplot2)
p = ggplot(mtcars) + aes_string(x='wt', y='mpg') + geom_point() + geom_smooth(method="loess")
print(p)`



testnotebook - Mozilla Firefox

Packt/chap10/ testnotebook

localhost:8888/notebooks/Packt/chap10/testnotebook.ipynb

jupyter testnotebook Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Help Python 3

Code CellToolbar

Graphics are fun

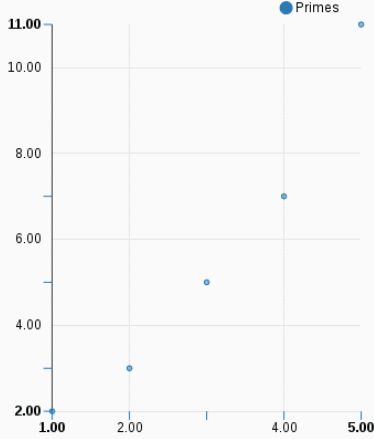
```
In [1]: import nvd3
nvd3.ipynb.initialize_javascript(use_remote=True)

from nvd3 import scatterChart

xs = [1, 2, 3, 4, 5]
ys = [2, 3, 5, 7, 11]
kwargs = {'shape': 'circle', 'size': '1'}
chart = scatterChart(name='The first few primes', height = 400, width=400)
chart.add_series(name="Primes", x=xs, y=ys, **kwargs)
chart
```

loaded nvd3 IPython extension
run nvd3.ipynb.initialize_javascript() to set up the notebook
help(nvd3.ipynb.initialize_javascript) for options

Out[1]:



Index	Prime Number
1	2
2	3
3	5
4	7
5	11

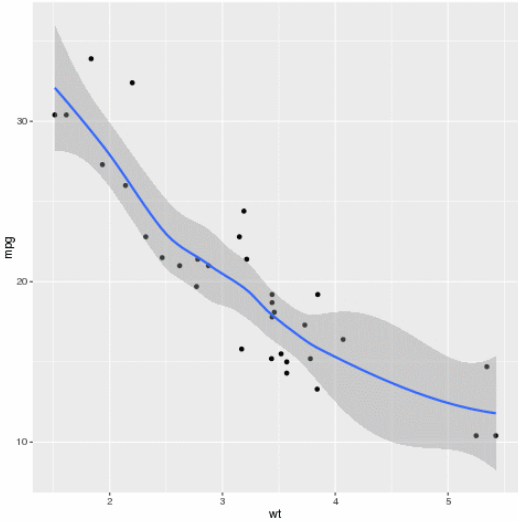
notebook2 - Mozilla Firefox

Packt/chap10/ testnotebook notebook2 Home notebook2

file:///nfs/02/wit0096/Packt/chap10/notebook2.html

```
In [1]: %load_ext rpy2.ipython
        .Rprofile: Setting UK repository

In [2]: %%R
        library(ggplot2)
        p = ggplot(mtcars) + aes_string(x='wt', y='mpg') + geom_point() + geom_smooth(method="loess")
        print(p)
```



Chapter 11: Into the Future

