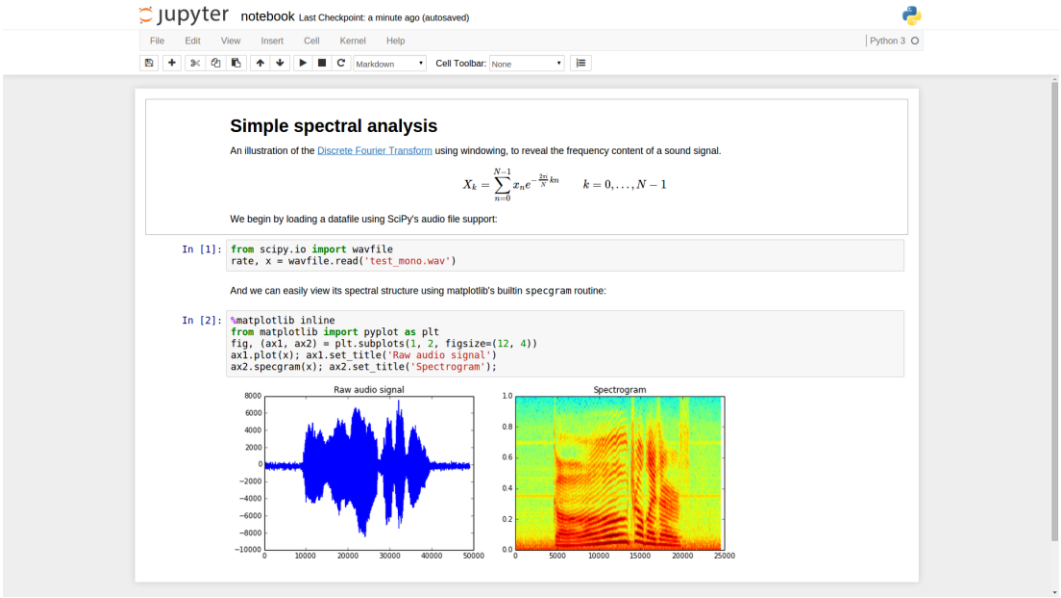
Chapter 1: Getting Started with IPython



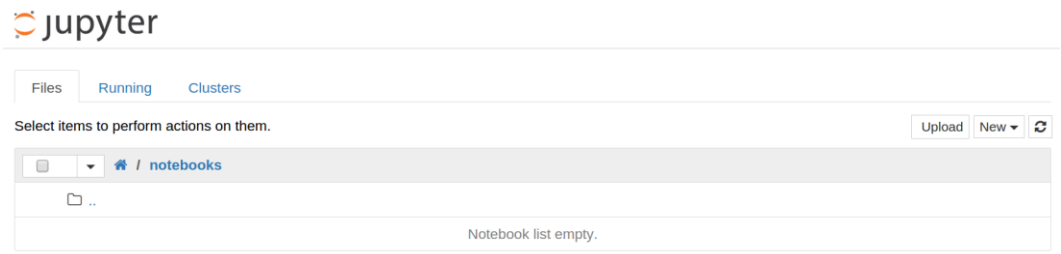

```
cyrille@gigabyte:~$ ipython
Python 3.4.3 |Anaconda 2.3.0 (64-bit)| (default, Jun  4 2015, 15:29:08)
Type "copyright", "credits" or "license" for more information.

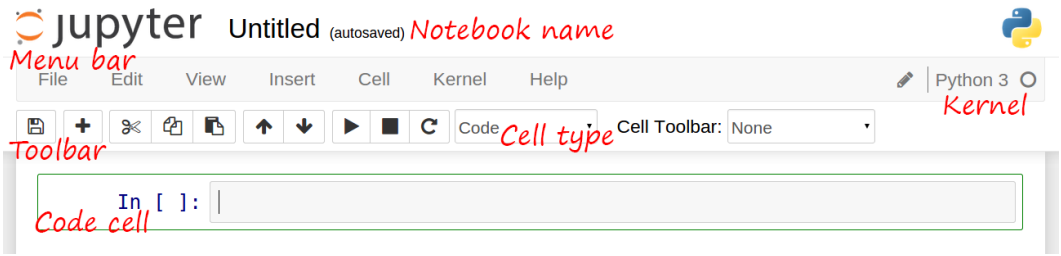
IPython 3.2.0 -- An enhanced Interactive Python.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: print("Hello world!")
Hello world!

In [2]: 2 * 3
Out[2]: 6

In [3]:
```






```
### New paragraph

This is *rich* **text** with [links](http://ipython.org), equations:

$$\hat{f}(\xi) = \int_{-\infty}^{+\infty} f(x)\, \mathrm{e}^{-i \xi x} dx$$

code with syntax highlighting:

```python
print("Hello world!")
```

and images:

![This is an image](http://jupyter.org/images/jupyter-sq-text.svg)
```

New paragraph

This is *rich* **text** with links, equations:

$$\hat{f}(\xi) = \int_{-\infty}^{+\infty} f(x)\, \mathrm{e}^{-i\xi x} dx$$

code with syntax highlighting:

```
print("Hello world!")
```

and images:

In [1]: Prompt number

Input area

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
print("Hello world!")
plt.imshow(np.random.rand(20, 20), interpolation='none');
from IPython.display import display_html
from IPython.html.widgets import FloatSlider
display_html('<table><tr><td>some</td><td>table</td></tr></table>', raw=True)
FloatSlider(value=70)
```

Widget area

70

Output area

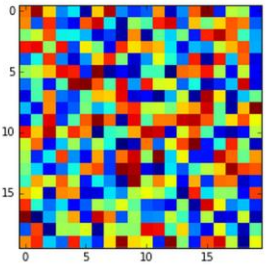
Hello world! Standard output

Error output

```
:0: FutureWarning: IPython widgets are experimental and may change in the future.
```

some	table

Rich output



jupyter Untitled1 (autosaved)

File Edit View Insert Cell Kernel Help Python 3

Code Cell Toolbar: None

```
In [1]: print("Hello world!")
        Hello world!
```

```
107.circles
107.edges
1684.circles
1684.edges
1912.circles
1912.edges
```

```
In [ ]: !head -n5 1
```

```
### New paragraph

This is *rich* **text** with [links](http://ipython.org), equations:

$$\hat{f}(\xi) = \int_{-\infty}^{+\infty} f(x)\, \mathrm{e}^{-i \xi x} dx$$

code with syntax highlighting:

```python
print("Hello world!")
```

and images:

![This is an image](http://jupyter.org/images/jupyter-sq-text.svg)
```

New paragraph

This is *rich* **text** with links, equations:

$$\hat{f}(\xi) = \int_{-\infty}^{+\infty} f(x)\, \mathrm{e}^{-i\xi x} dx$$

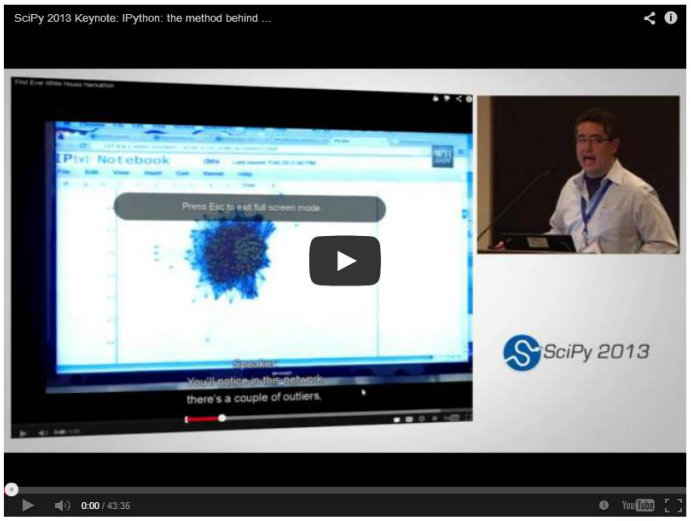
code with syntax highlighting:

```
print("Hello world!")
```

and images:

In [13]:
```
YouTubeVideo('j9YpkSX7NNM')
```

Out[13]:



```
In [2]: from IPython.html.widgets import interact
        @interact(x=(0, 10))
        def square(x):
            print("The square of %d is %d." % (x, x**2))
```

x 7

The square of 7 is 49.

```
In [3]: import networkx

In [4]: networkx.Graph?
```

```
Type:            type
String form:     <class 'networkx.classes.graph.Graph'>
File:            /home/cyrille/anaconda/envs/minibook/lib/python3.4/site-packages/n
etworkx/classes/graph.py
Init definition: networkx.Graph(self, data=None, **attr)
Docstring:
Base class for undirected graphs.

A Graph stores nodes and edges with optional data, or attributes.

Graphs hold undirected edges.  Self loops are allowed but multiple
(parallel) edges are not.

Nodes can be arbitrary (hashable) Python objects with optional
```

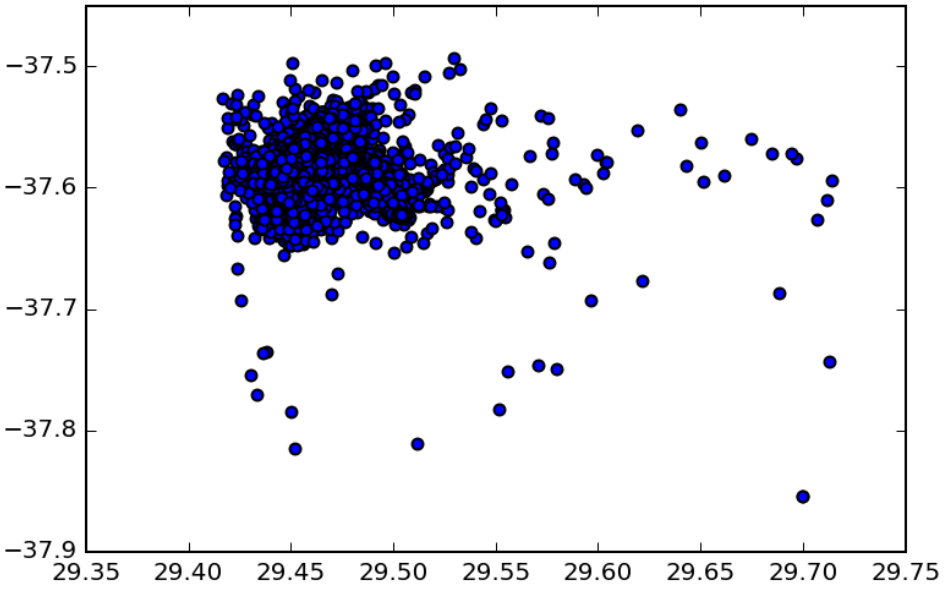
Chapter 2: Interactive Data Analysis with pandas

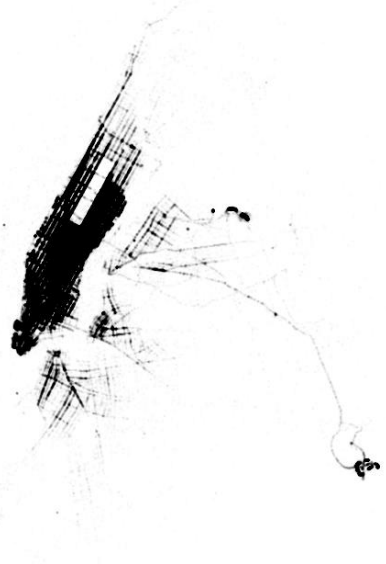
```
data.head(3)
```

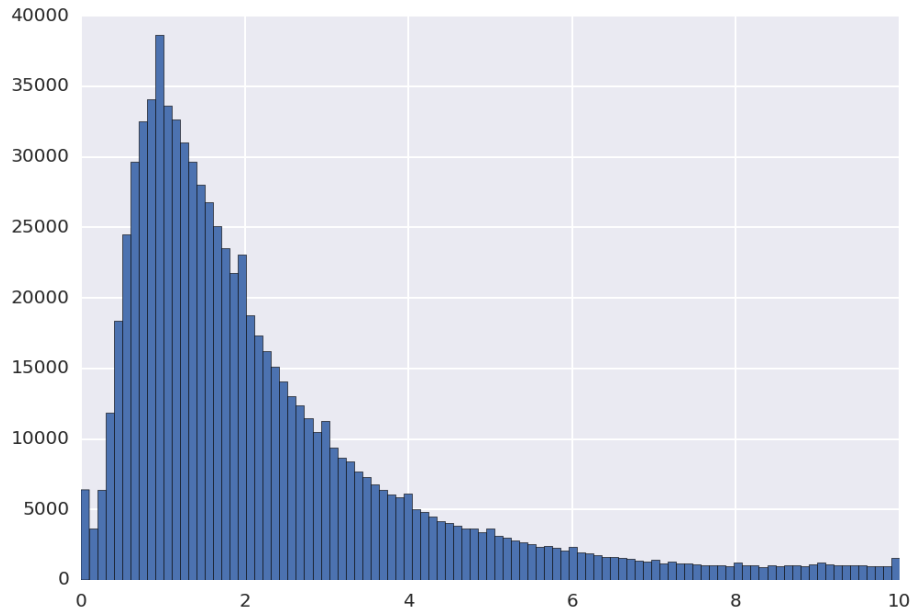
	medallion	hack_license	vendor_id	rate_code	store_and_fwd_flag	pickup_datetim
0	76942C3205E17D7E7FE5A9F709D16434	25BA06A87905667AA1FE5990E33F0E2E	VTS	1	NaN	2013-01-01 00:00:00
1	517C6B330DBB3F055D007B07512628B3	2C19FBEE1A6E05612EFE4C958C14BC7F	VTS	1	NaN	2013-01-01 00:05:00
2	ED15611F168E41B33619C83D900FE266	754AEBD7C80DA17BA1D81D89FB6F4D1D	CMT	1	N	2013-01-01 00:05:52

```
data.describe()
```

	rate_code	passenger_count	trip_time_in_secs	trip_distance	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latit
count	846945.000000	846945.000000	846945.000000	846945.000000	846945.000000	846945.000000	846945.000000	846945.0000
mean	1.026123	1.710272	812.523879	9.958211	-73.975155	40.750490	-73.974197	40.750967
std	0.223480	1.375266	16098.305145	6525.204888	0.035142	0.027224	0.033453	0.030766
min	0.000000	0.000000	-10.000000	0.000000	-74.098305	40.009911	-74.099998	40.009911
25%	1.000000	1.000000	361.000000	1.050000	-73.992371	40.736031	-73.991570	40.735207
50%	1.000000	1.000000	600.000000	1.800000	-73.982094	40.752975	-73.980614	40.753597
75%	1.000000	2.000000	960.000000	3.200000	-73.968048	40.767460	-73.965157	40.768227
max	6.000000	6.000000	4294796.000000	6005123.000000	-73.028473	40.996132	-73.027061	40.998592







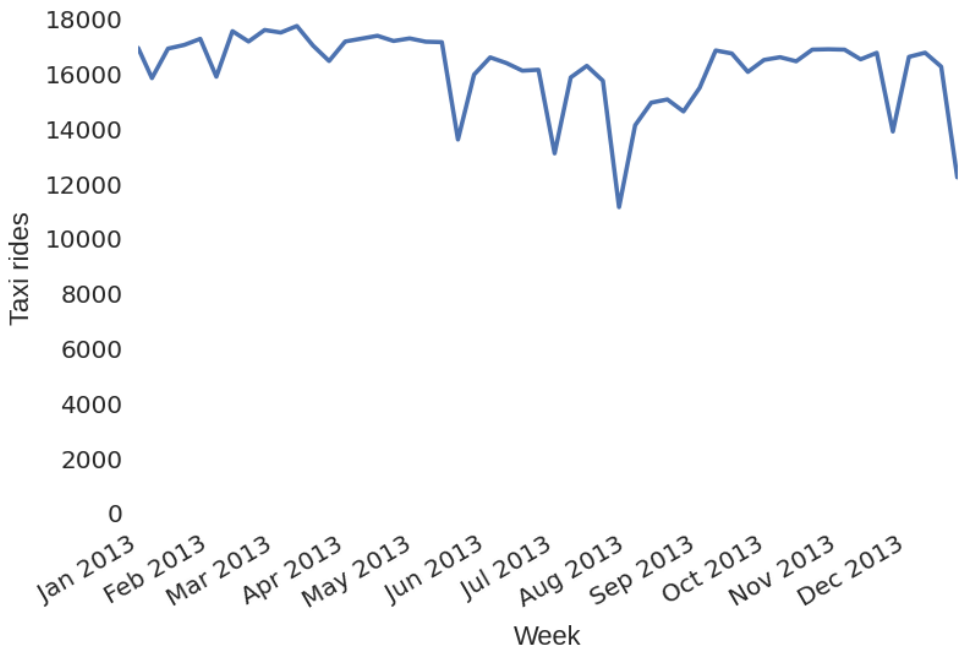
	medallion	hack_license	vendor_id	rate_code	store_and_fwd_flag	pickup_d
0	76942C3205E17D7E7FE5A9F709D16434	25BA06A87905667AA1FE5990E33F0E2E	VTS	1	NaN	2013-01-(00:00:00
100000	7461F7106D33D3A5775F4245724606FD	BACEA353BB4106A005BB7836BDCAC0C3	VTS	1	NaN	2013-02-1 18:10:00

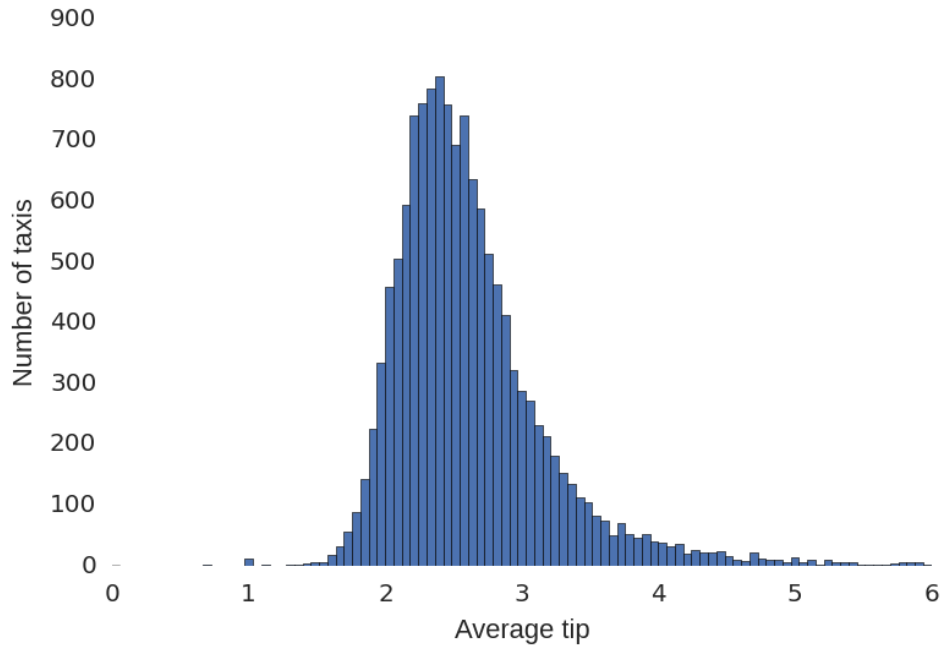
dropoff_datetime	passenger_count	trip_time_in_secs	trip_distance	pickup_longitude	pickup_latitude	dropoff_longitude
2013-01-01 21:56:37	1	934	52.20	-73.979576	40.743626	-73.941902
2013-01-04 07:17:14	1	1973	96.30	-73.959785	40.762497	-73.962440
2013-01-05 02:23:01	1	1913	52.90	-74.006119	40.735157	-73.958694
2013-01-12 03:24:47	1	1312	66.20	-73.966873	40.683315	-73.916885

distance_threshold 26

307

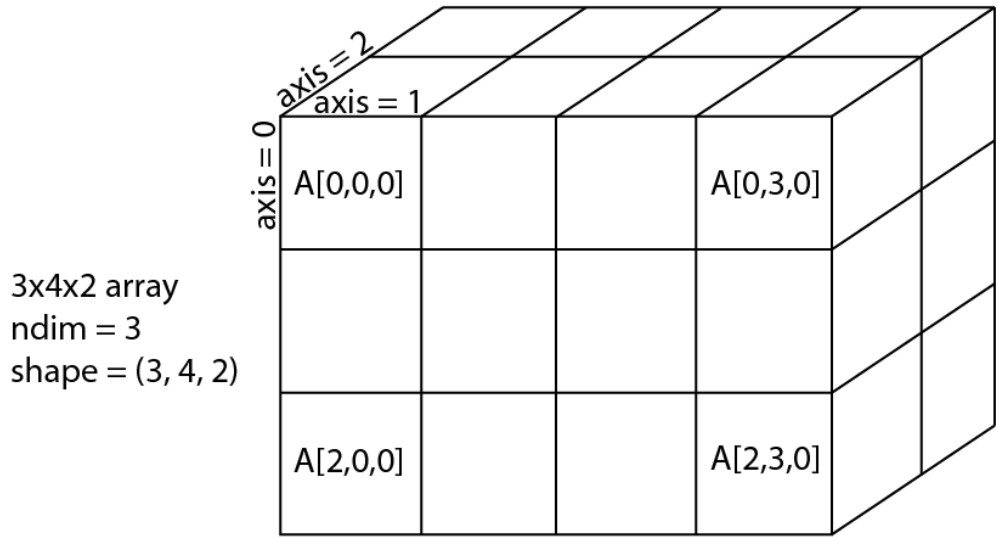
	vendor_id	rate_code	store_and_fwd_flag	pickup_datetime	dropoff_datetime	passenger_count	trip_time_in_secs	trip_distance
ED	VTS	1	NaN	2013-01-01 23:45:00	2013-01-02 00:03:00	1	1080	12.61
95	CMT	1	N	2013-01-01 23:46:22	2013-01-02 00:28:01	1	2498	16.10
49	CMT	1	N	2013-01-01 23:46:53	2013-01-02 00:03:33	1	1000	5.40

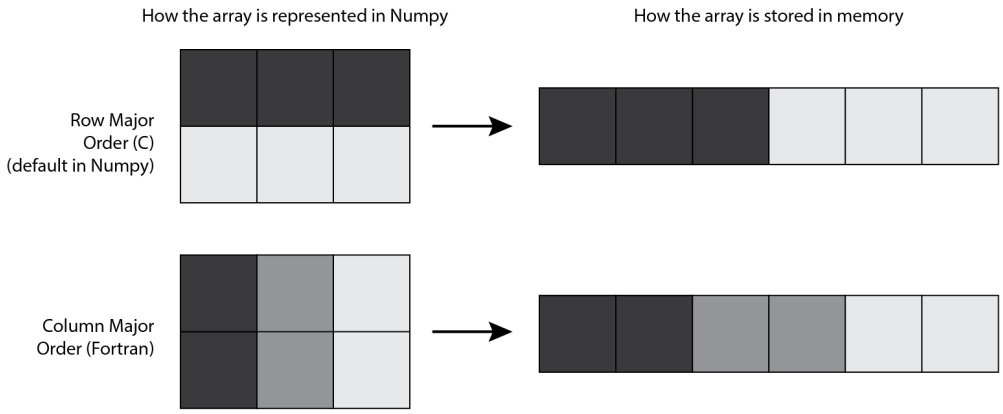


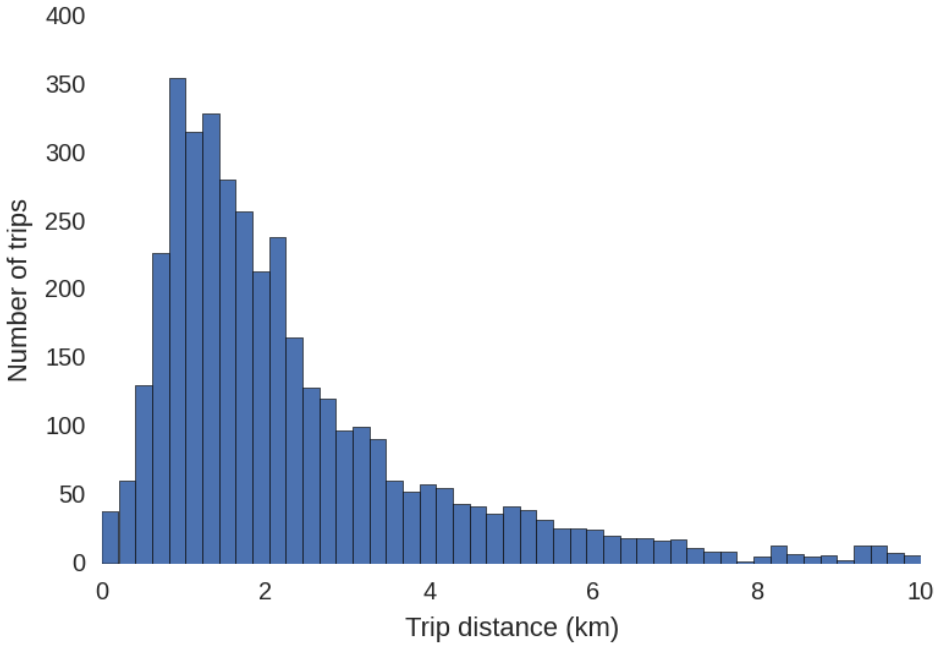


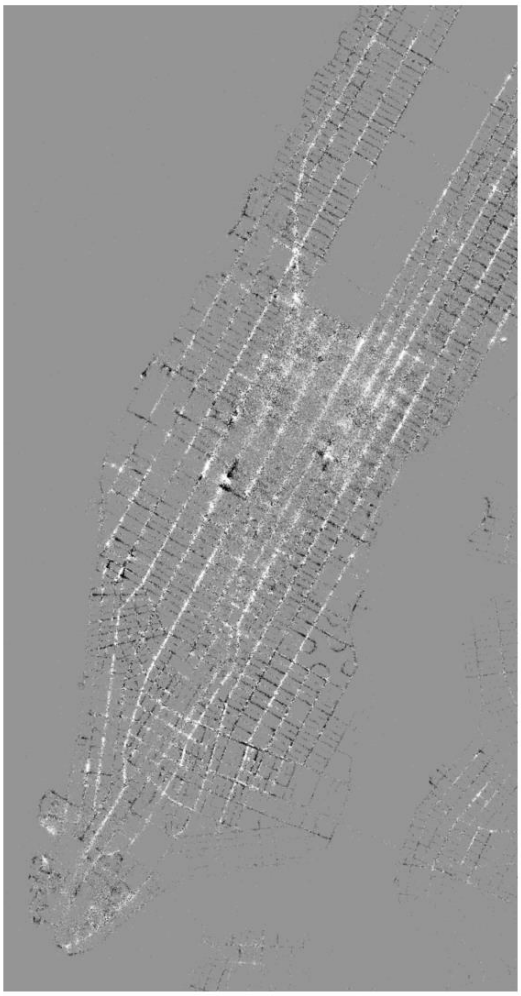
trip_distance	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	tip_amount
0.61	-73.955925	40.781887	-73.963181	40.777832	3.180417
3.28	-74.005501	40.745735	-73.964943	40.755722	2.863235
1.50	-73.969955	40.799770	-73.954567	40.787392	2.147143

Chapter 3: Numerical Computing with NumPy



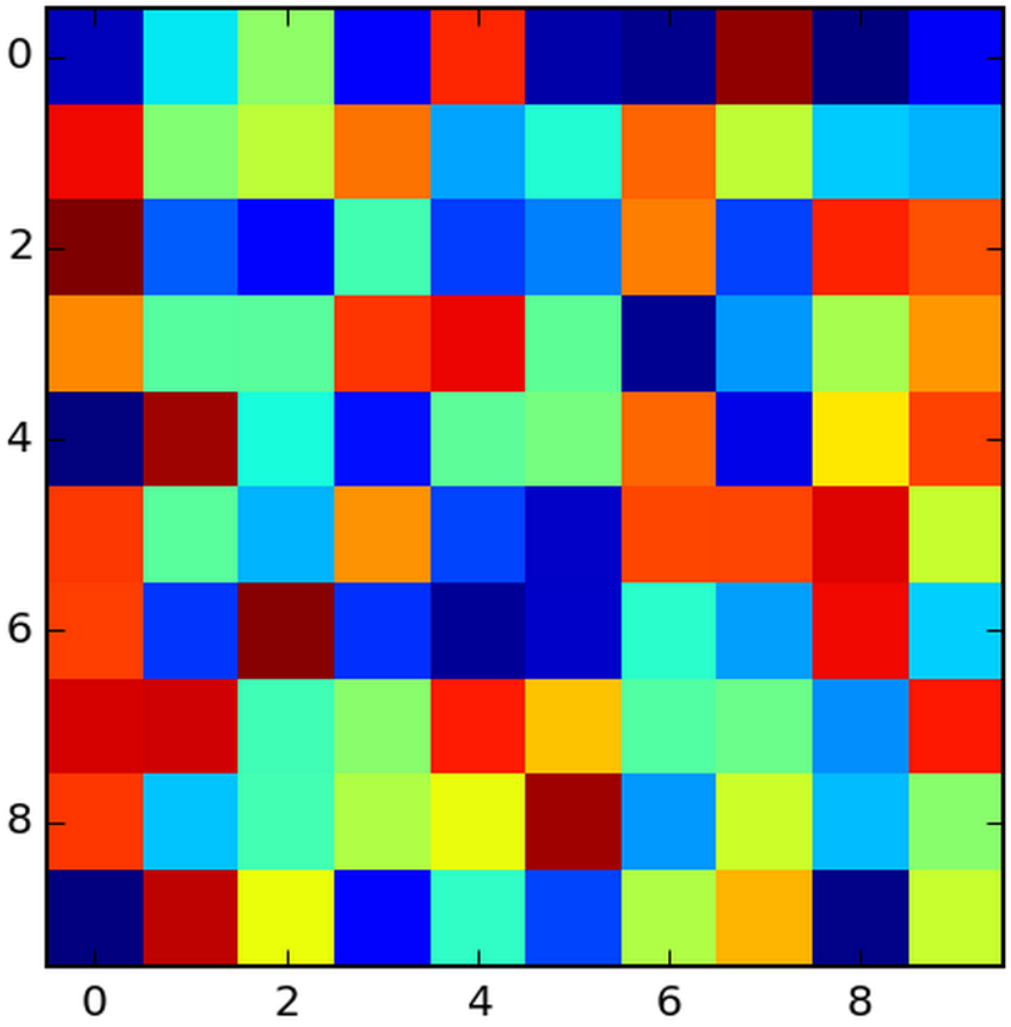



400
350
300
250
200
150
100
50
0
Number of trips
0
2
4
6
8
10
Trip distance (km)



Chapter 4: Interactive Plotting and Graphical Interfaces

```
plt.imshow(np.random.rand(10, 10), interpolation='none')
```



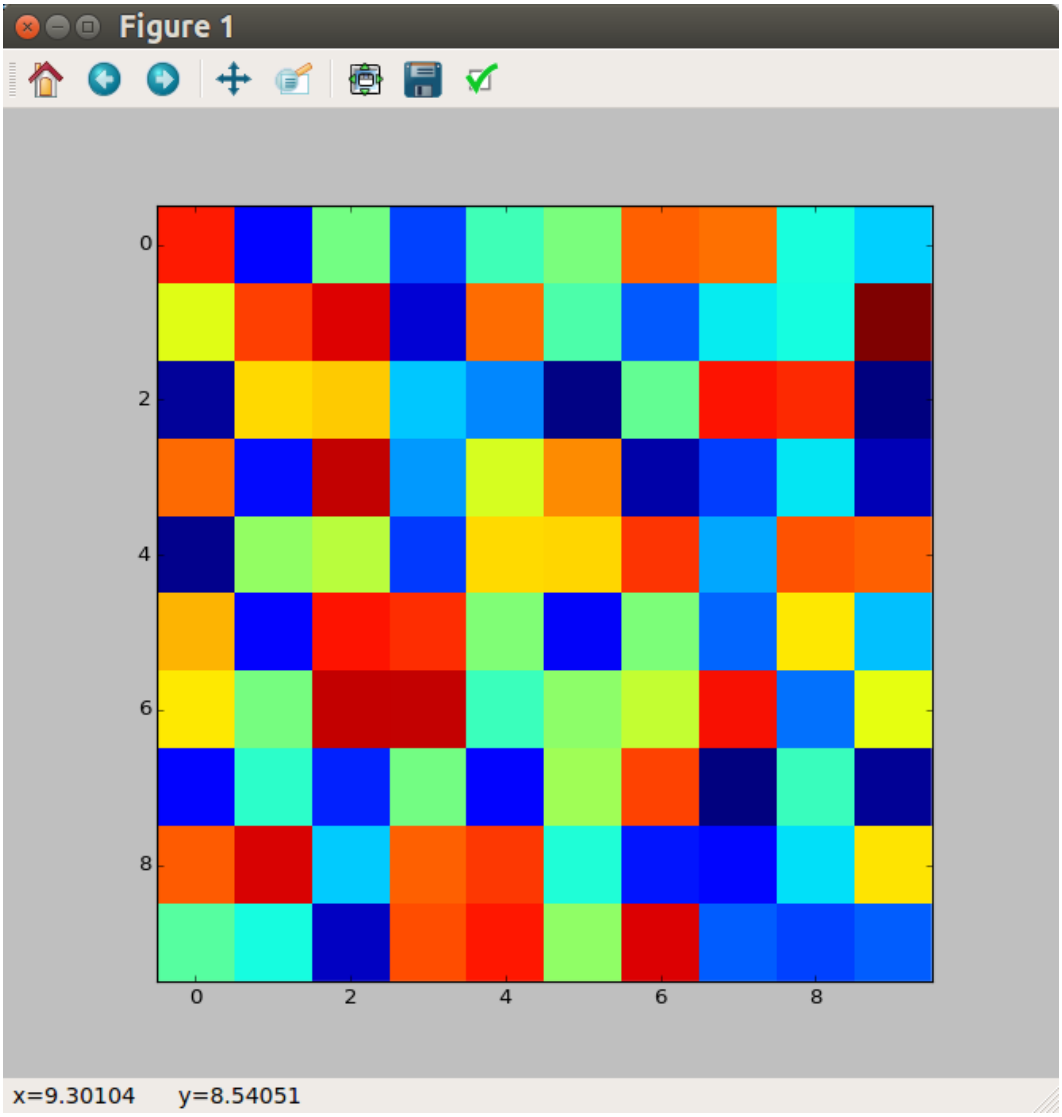
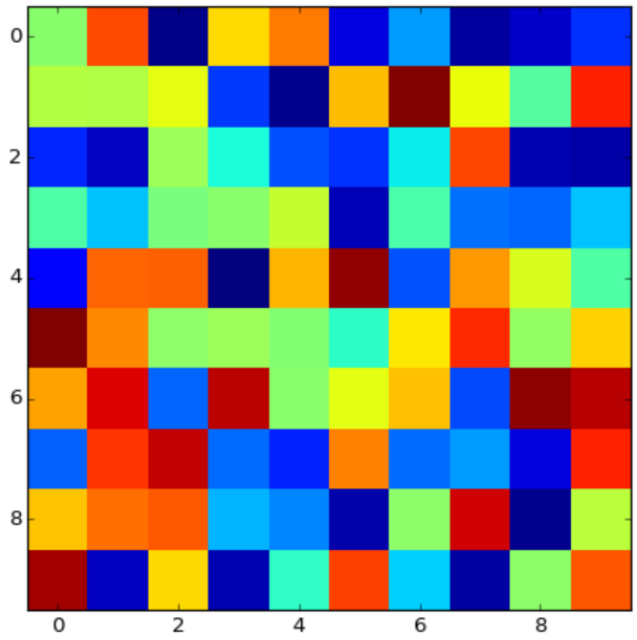


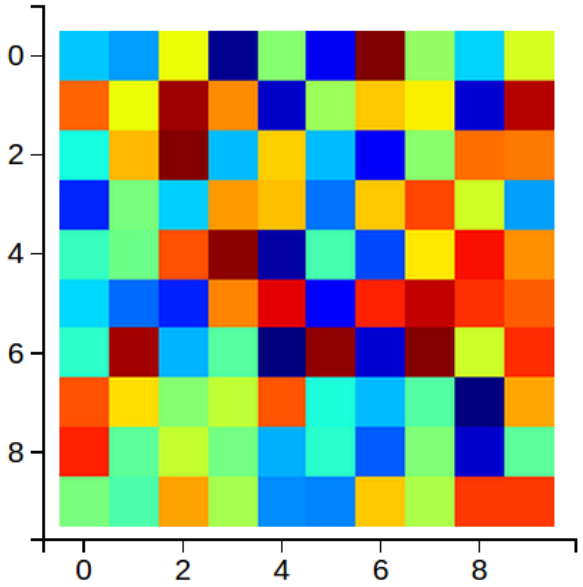
Figure 1
0
2
4
6
8
0
2
4
6
8
x=9.30104
y=8.54051

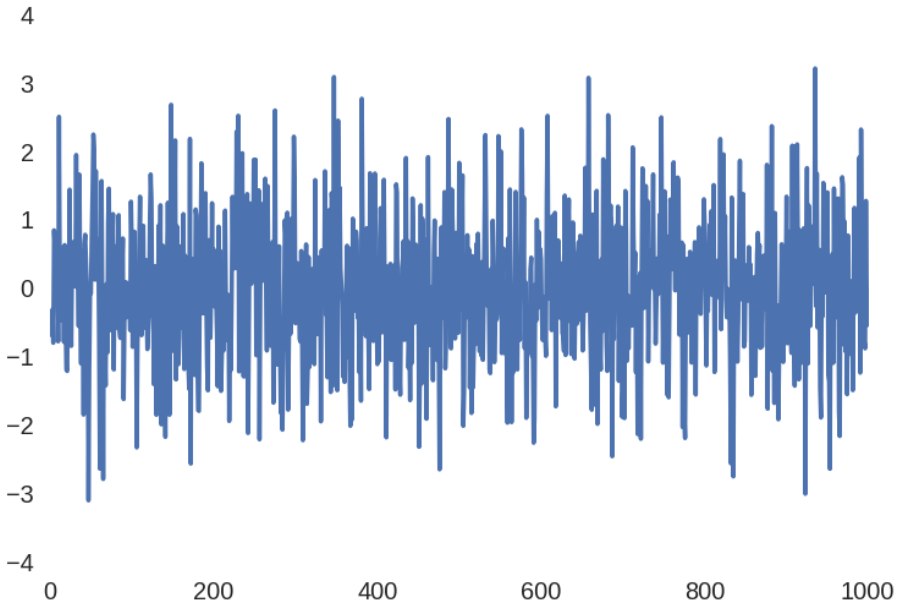
Figure 2

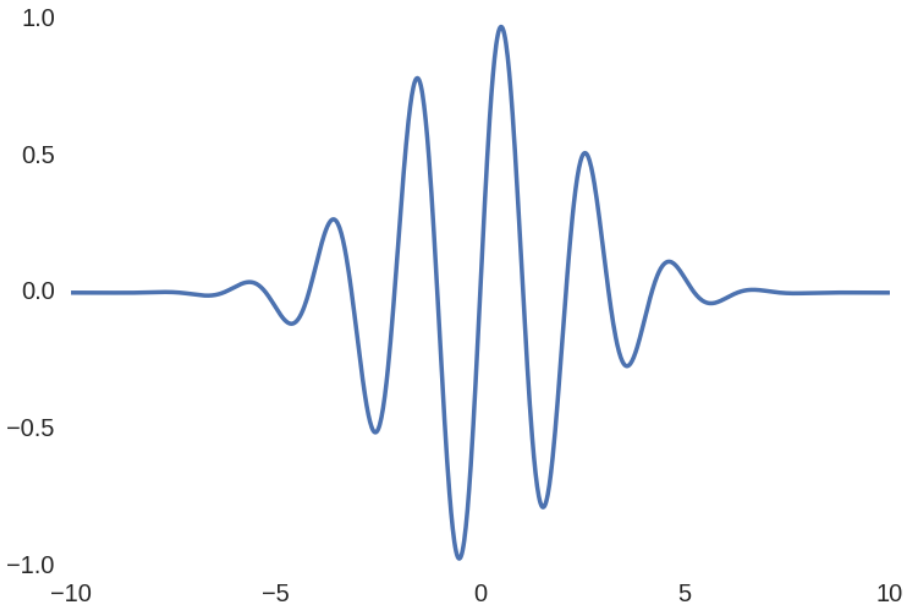


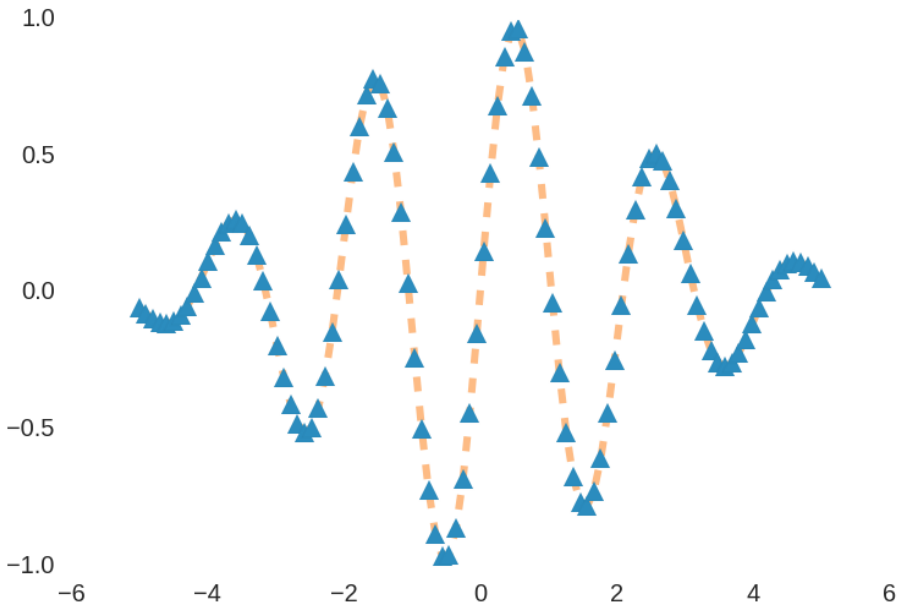
zoom rect, x=1.07308 y=1.76156

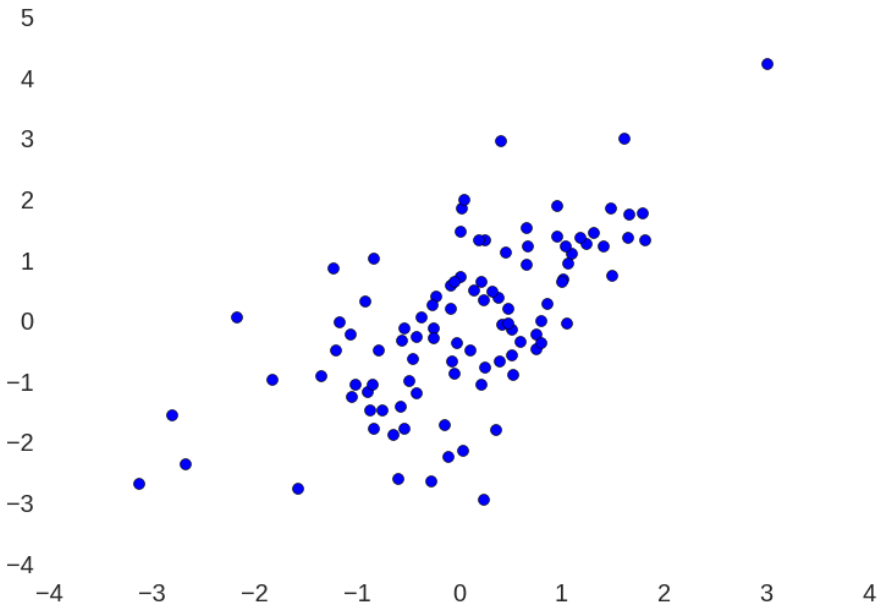
```
plt.imshow(np.random.rand(10, 10), interpolation='none')
```

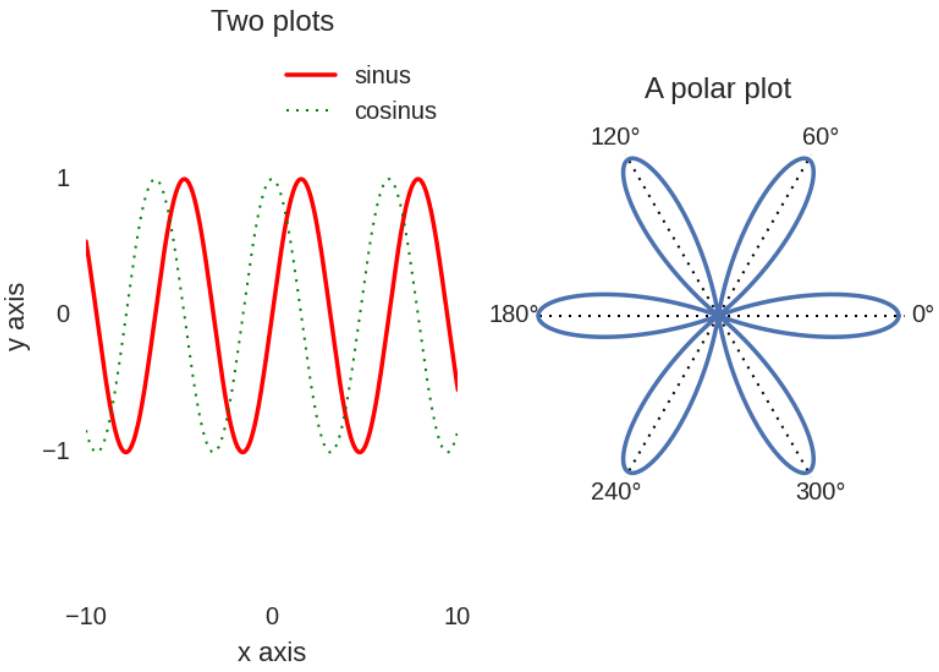


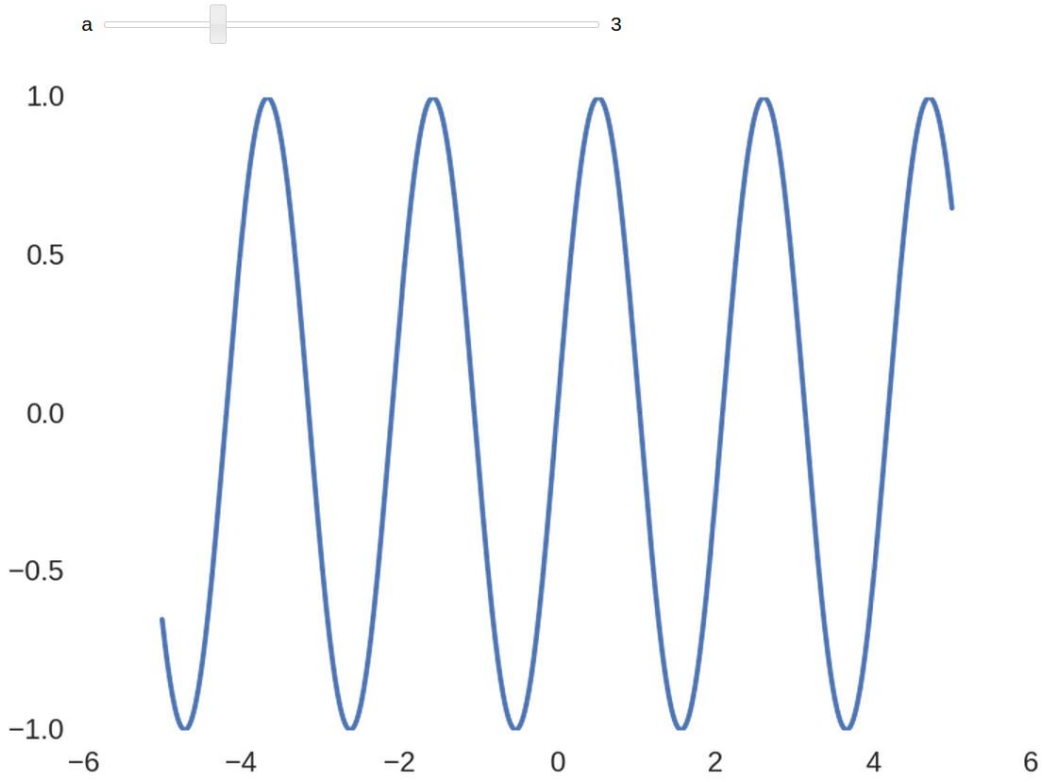

4
3
2
1
0
−1
−2
−3
−4
0
200
400
600
800
1000


1.0
0.5
0.0
−0.5
−1.0
−10
−5
0
5
10


1.0
0.5
0.0
−0.5
−1.0
−6
−4
−2
0
2
4
6


5
4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4


Two plots
sinus
cosinus
1
0
−1
y axis
−10
0
10
x axis
A polar plot
120°
60°
180°
0°
240°
300°


a
3
1.0
0.5
0.0
−0.5
−1.0
−6
−4
−2
0
2
4
6

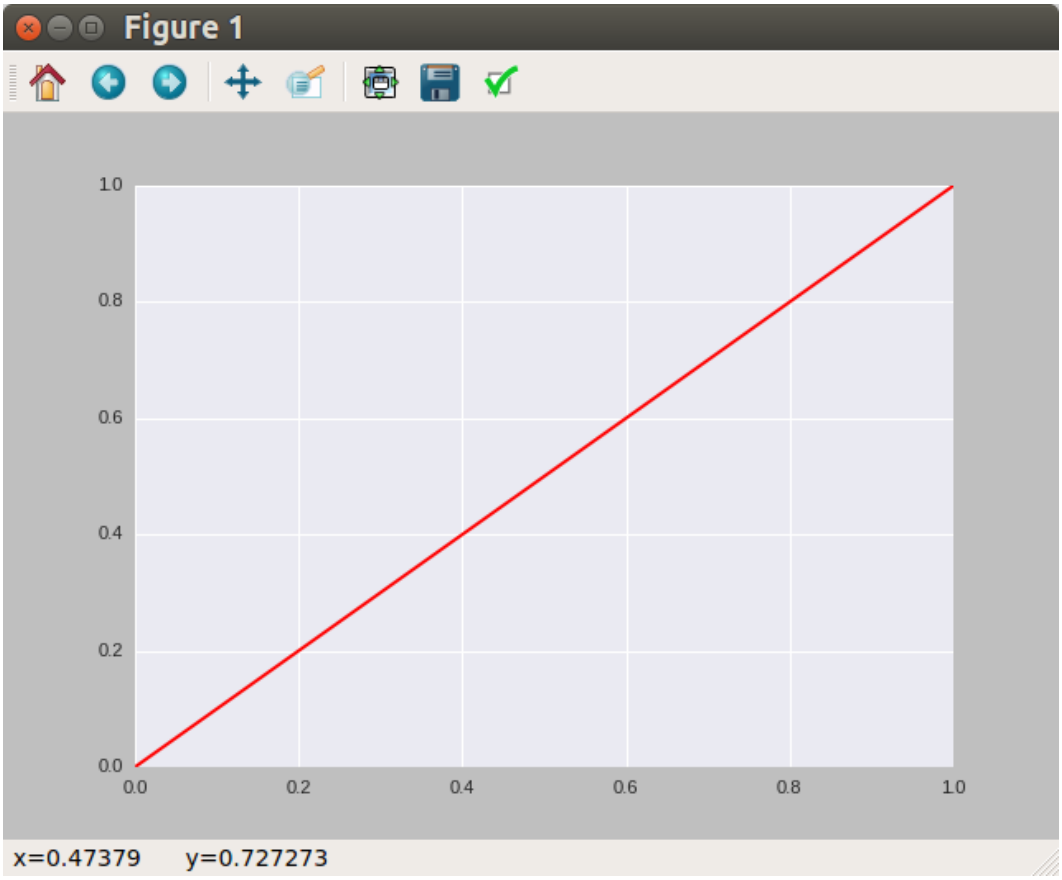
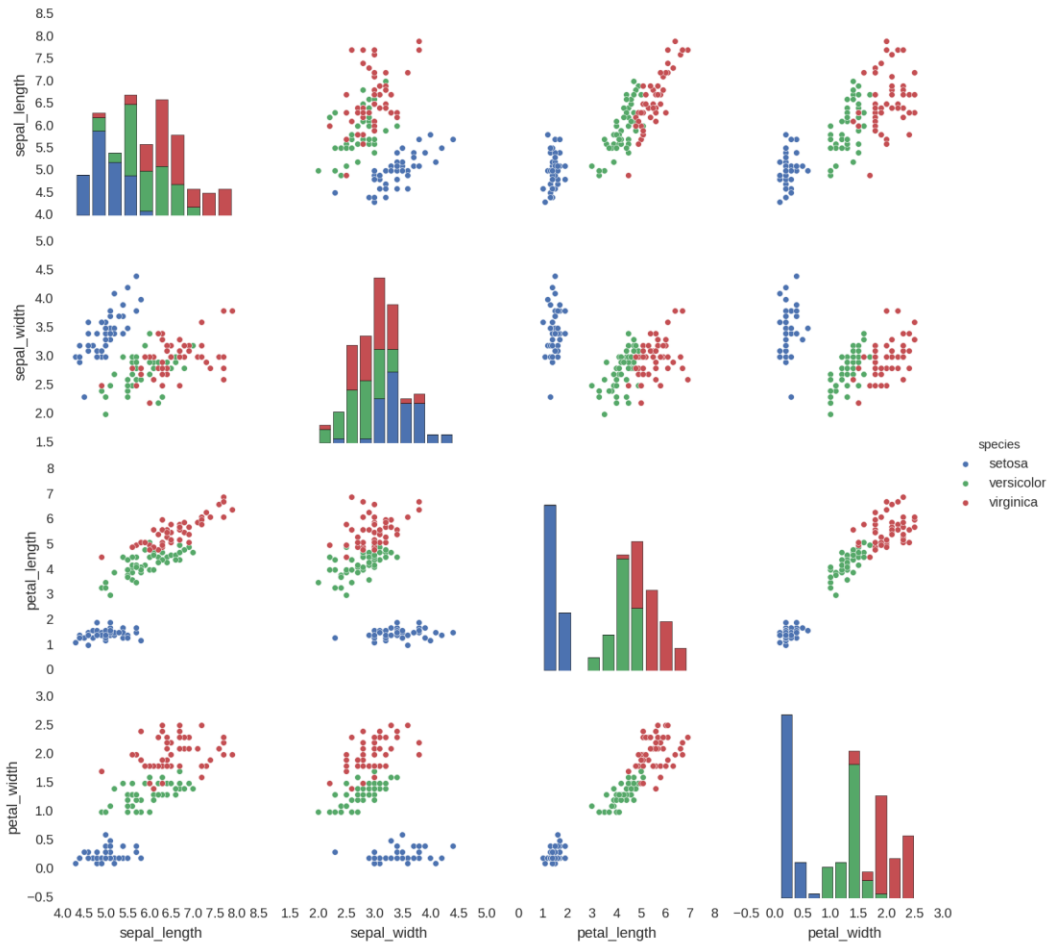
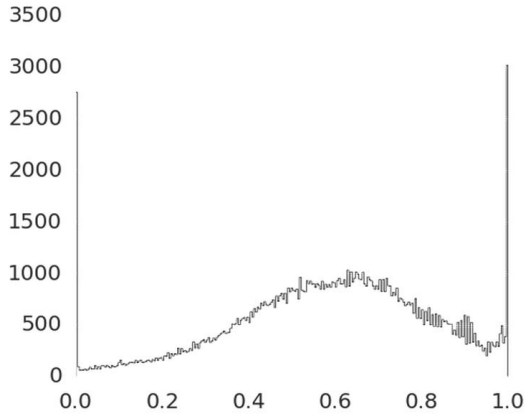


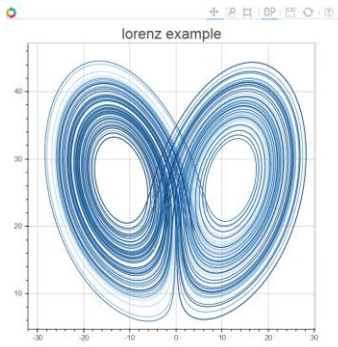
Figure 1
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
x=0.47379 y=0.727273

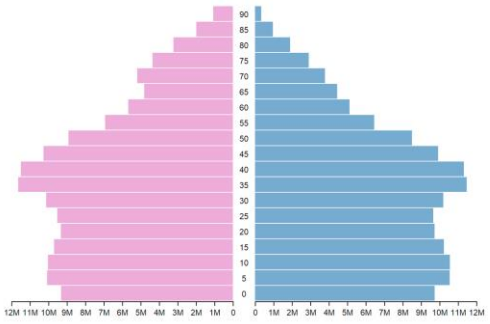

sepal_length
sepal_width
petal_length
petal_width
species
setosa
versicolor
virginica

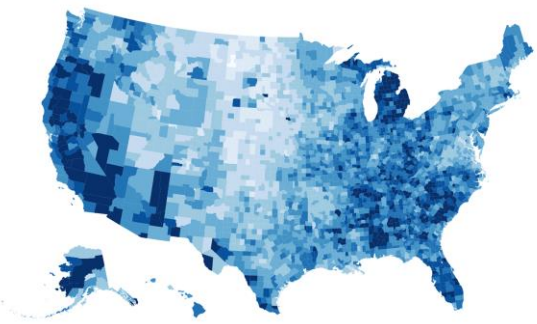
hist_type Contrast stretching

- Adaptive equalization
- Histogram equalization
- Contrast stretching



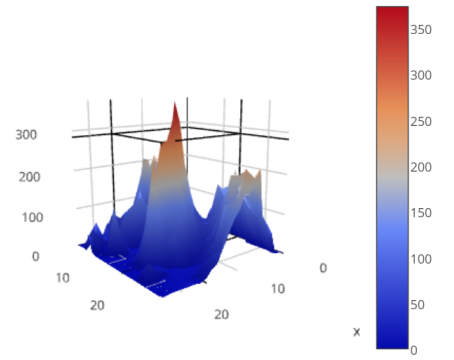




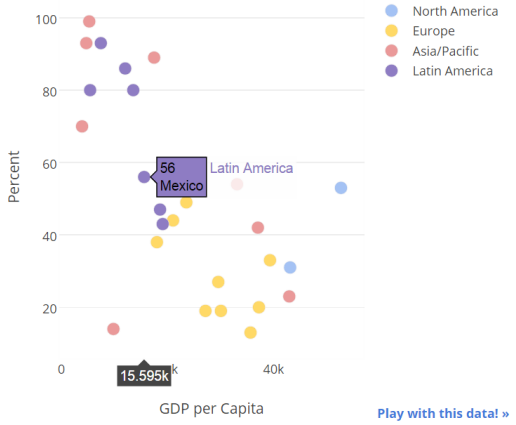


Mt Bruno Elevation

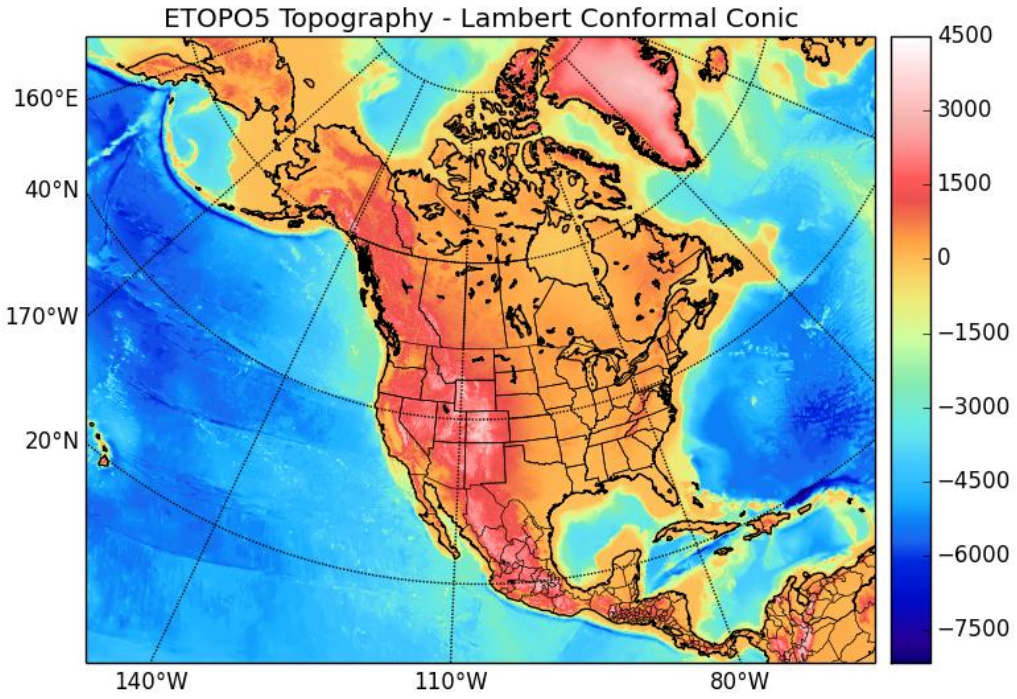
Quarter 1 Growth

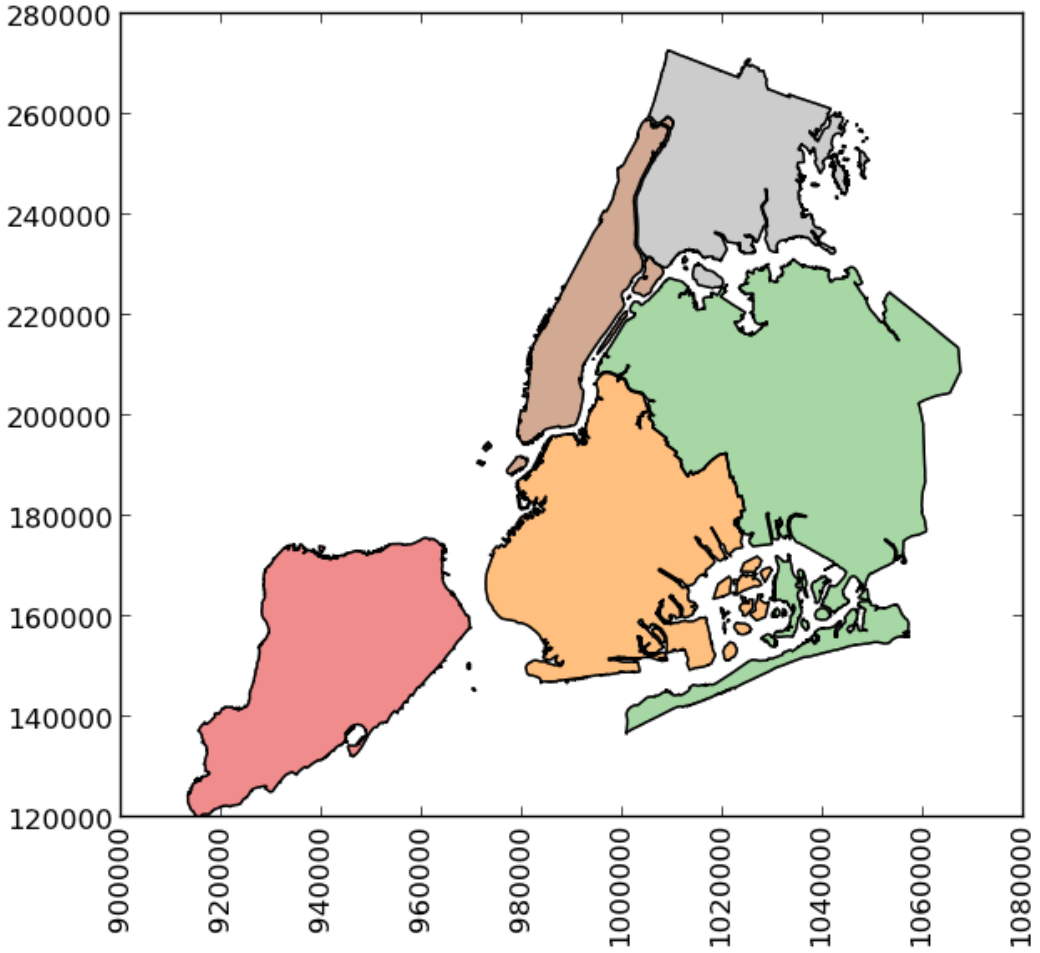


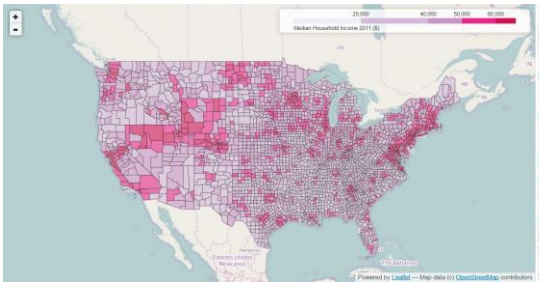
Play with this data! »

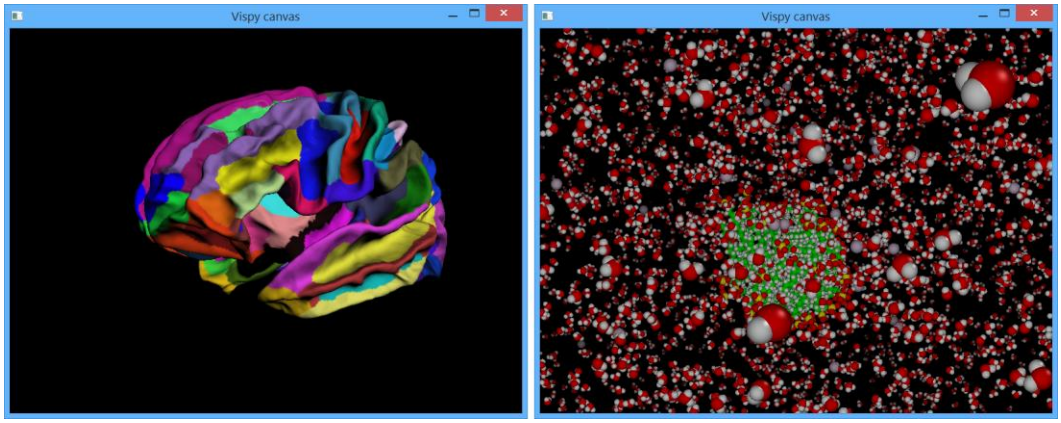


Play with this data! »

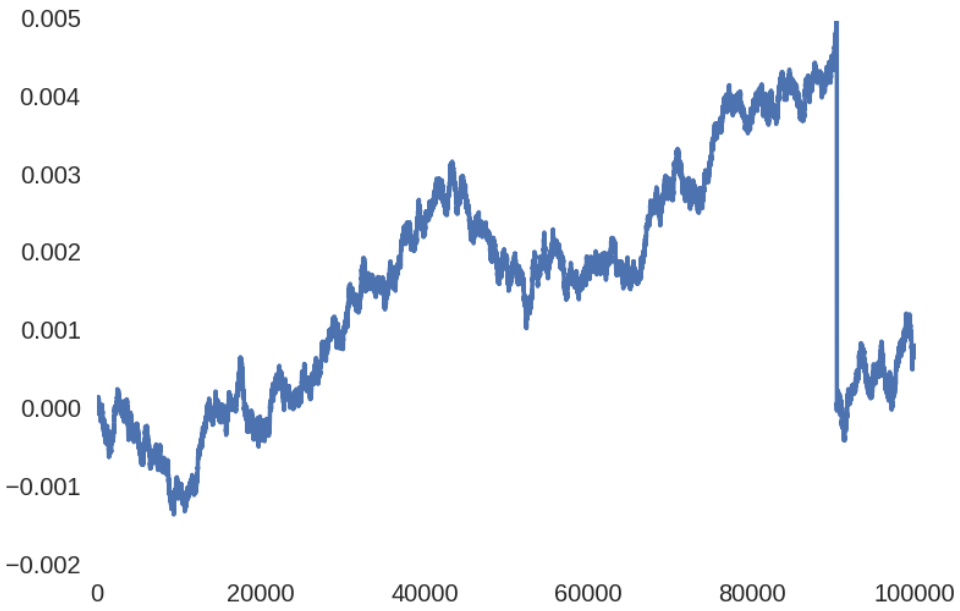

ETOPO5 Topography - Lambert Conformal Conic
160°E
40°N
170°W
20°N
140°W
110°W
80°W
4500
3000
1500
0
−1500
−3000
−4500
−6000
−7500


280000
260000
240000
220000
200000
180000
160000
140000
120000
900000
920000
940000
960000
980000
1000000
1020000
1040000
1060000
1080000



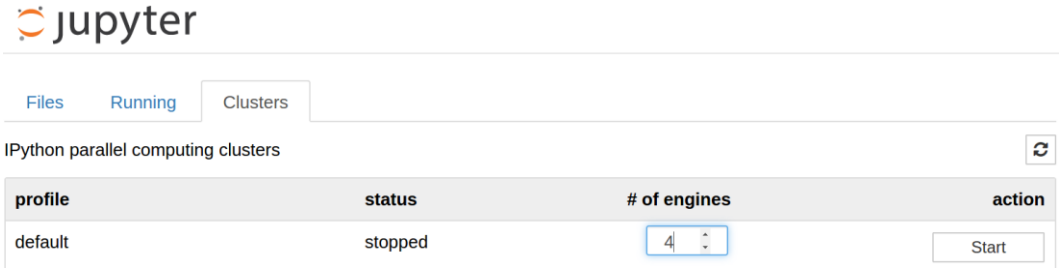


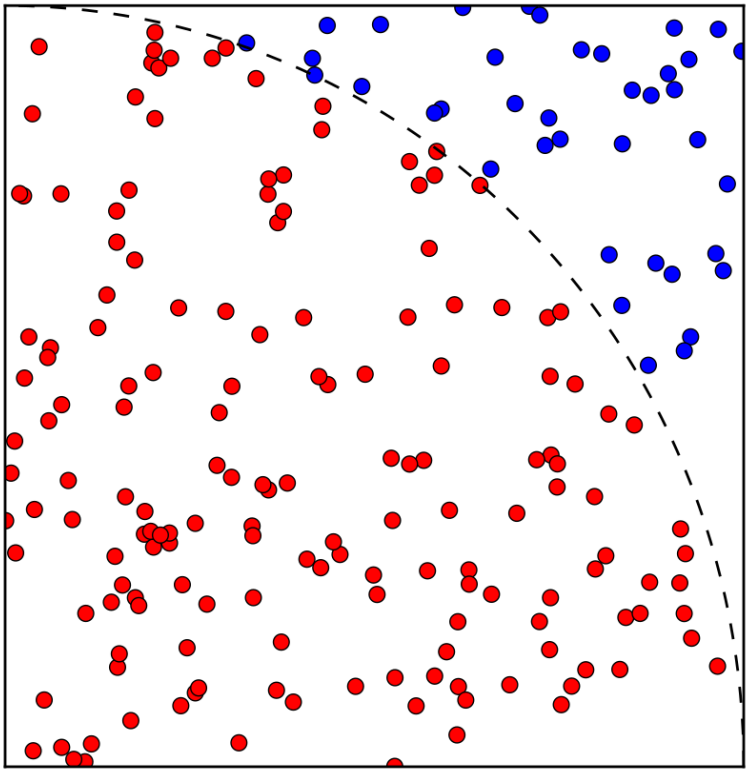
Chapter 5: High-Performance and Parallel Computing



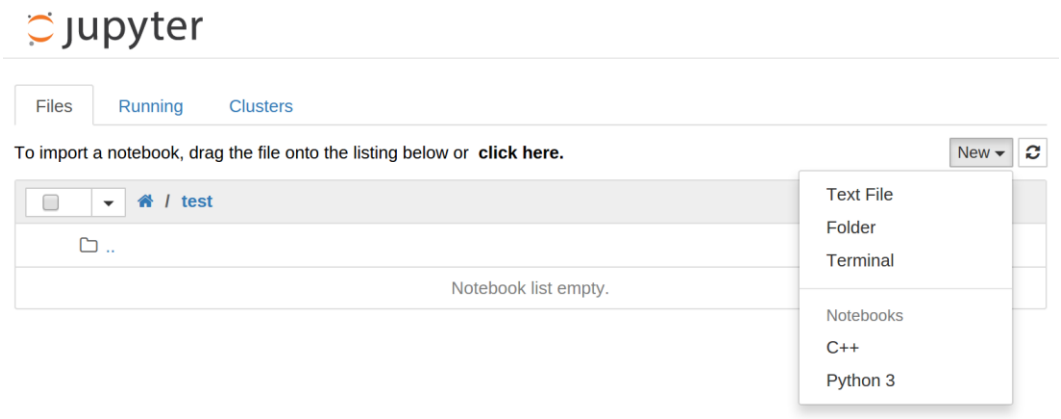
Generated by Cython 0.21

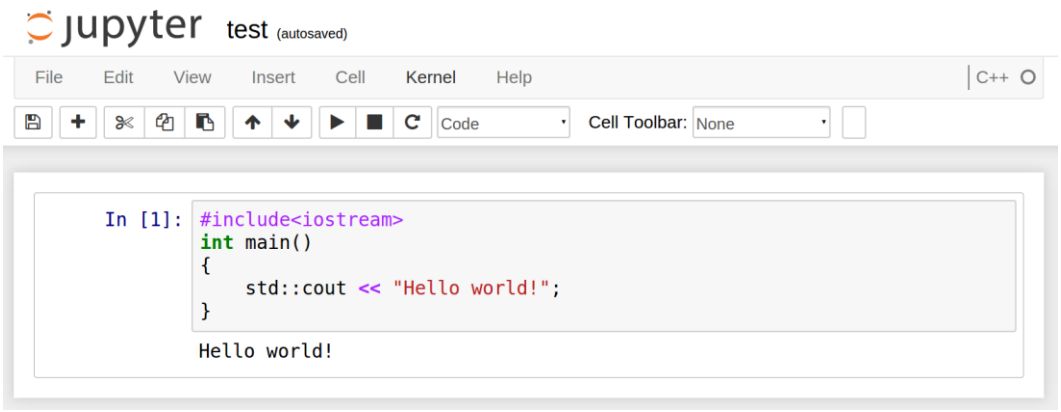
```
+01: def primes_cython_2(int n):
 02:     # Note the type declarations below:
+03:     cdef list primes = [False, False] + [True] * (n - 2)
+04:     cdef int i = 2
+05:     cdef int k = 0
 06:     # The rest of the function is unchanged.
+07:     while i < n:
 08:         # We do not deal with composite numbers.
+09:         if not primes[i]:
    __pyx_t_3 = __Pyx_GetItemInt_List(__pyx_v_primes, __pyx_v_i, int, 1, __Pyx_PyInt_From_int, 1, 1, 1); if (unlikely(__pyx_t_3 == NULL)) {__pyx_filename = __pyx_f[0]; __pyx_lineno = 9; __pyx_clineno = __LINE__; goto __pyx_L1_error;};
    __Pyx_GOTREF(__pyx_t_3);
    __pyx_t_4 = __Pyx_PyObject_IsTrue(__pyx_t_3); if (unlikely(__pyx_t_4 < 0)) {__pyx_filename = __pyx_f[0]; __pyx_lineno = 9; __pyx_clineno = __LINE__; goto __pyx_L1_error;}
    __Pyx_DECREF(__pyx_t_3); __pyx_t_3 = 0;
    __pyx_t_5 = ((!__pyx_t_4) != 0);
    if (__pyx_t_5) {
+10:             i += 1
+11:             continue
+12:         k = i * i
 13:         # We mark multiples of i as composite numbers.
+14:         while k < n:
+15:             primes[k] = False
+16:             k += i
+17:         i += 1
 18:     # We return all numbers marked with True.
+19:     return [i for i in range(2, n) if primes[i]]
```

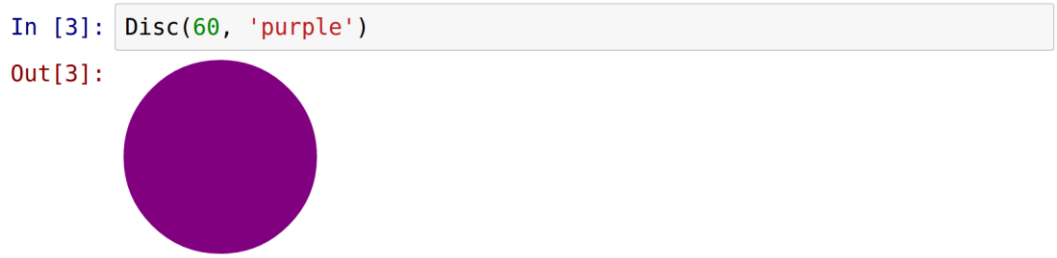


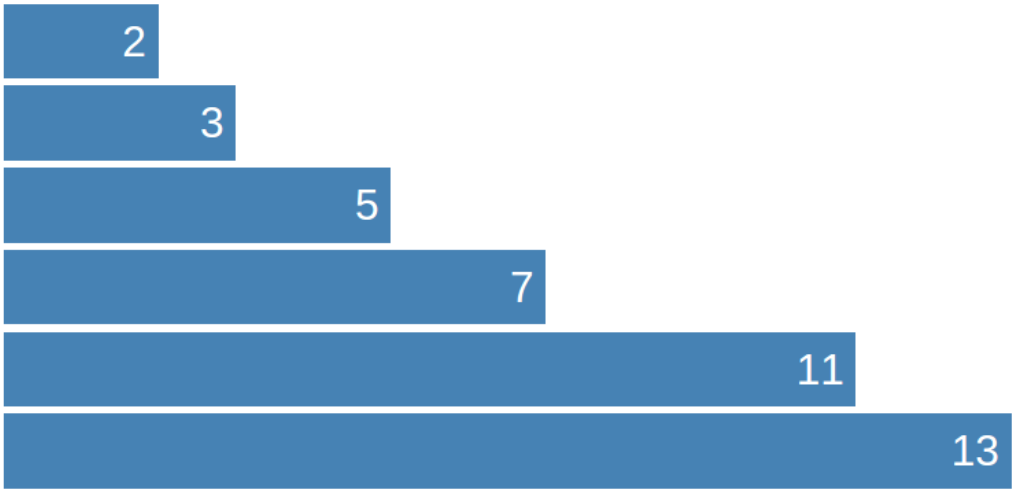


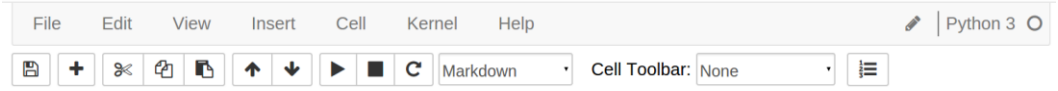
Chapter 6: Customizing IPython








2
3
5
7
11
13


File
Edit
View
Insert
Cell
Kernel
Help
Python 3
Markdown
Cell Toolbar:
None