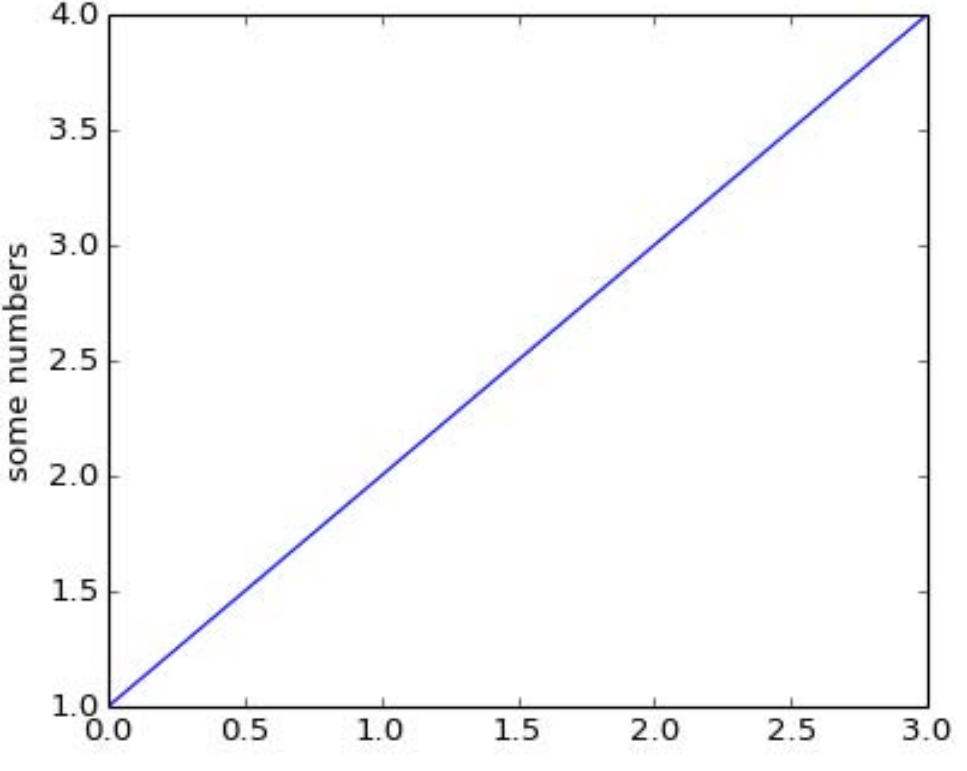
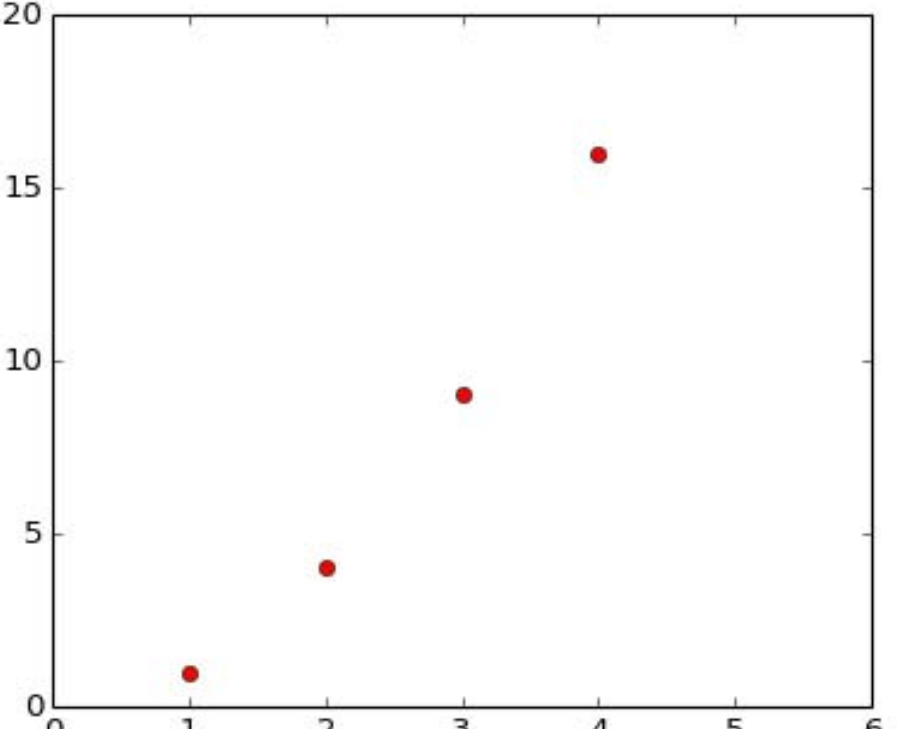
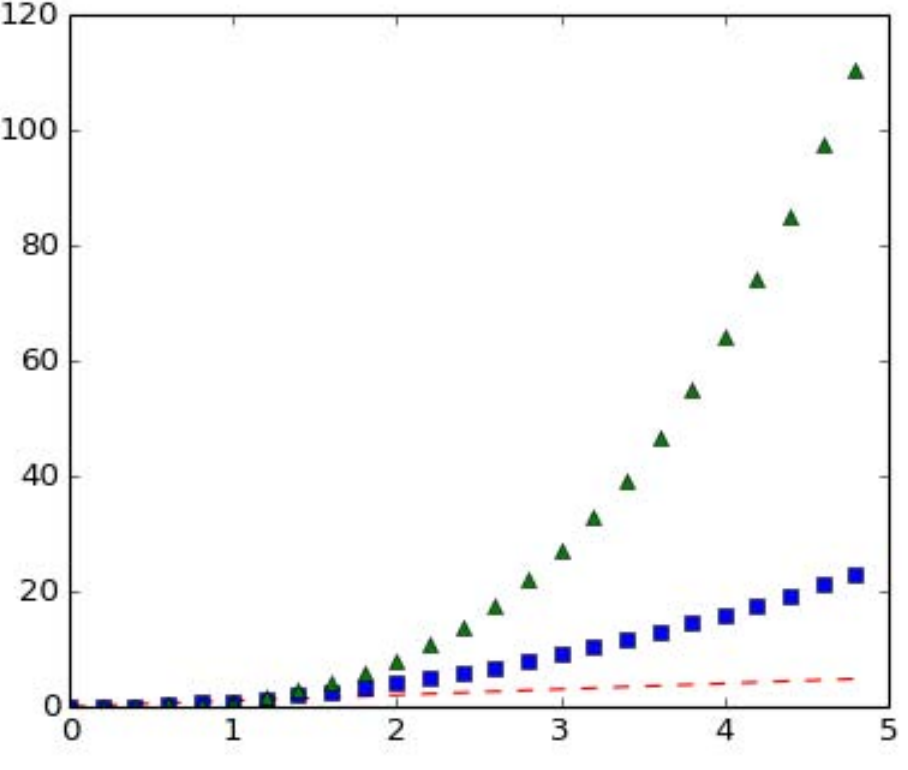
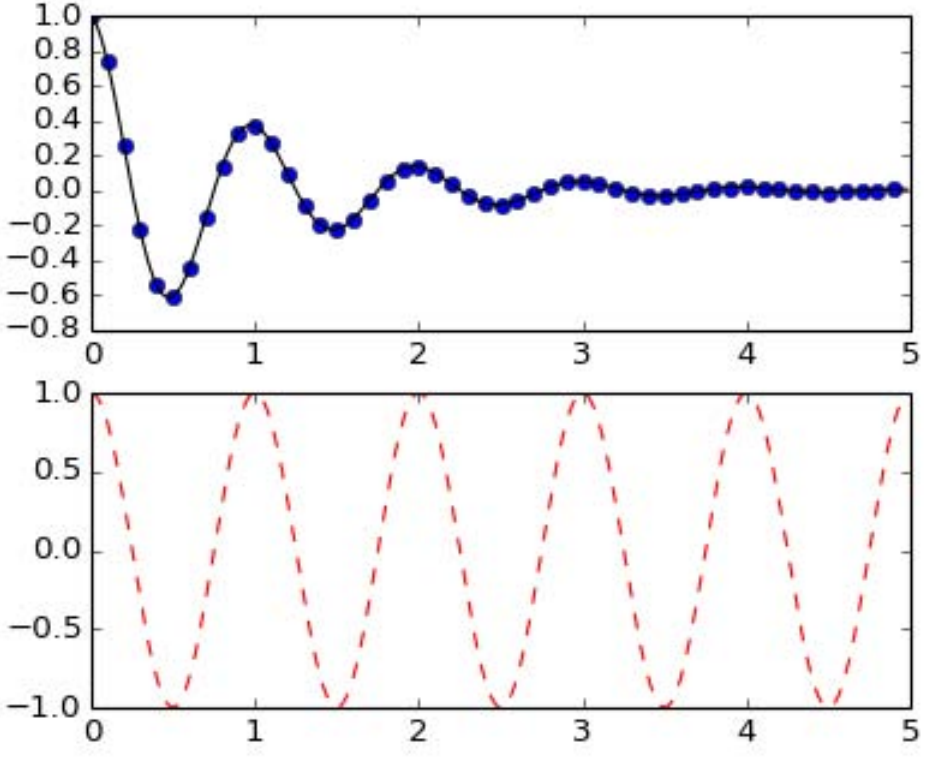
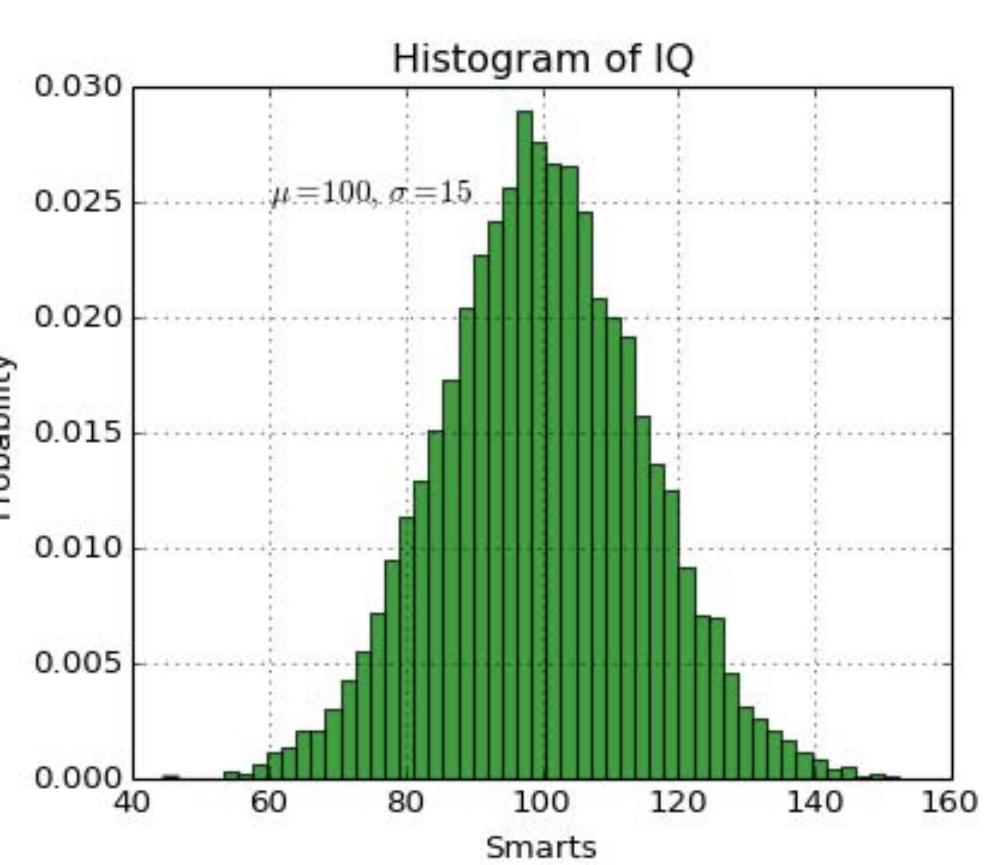
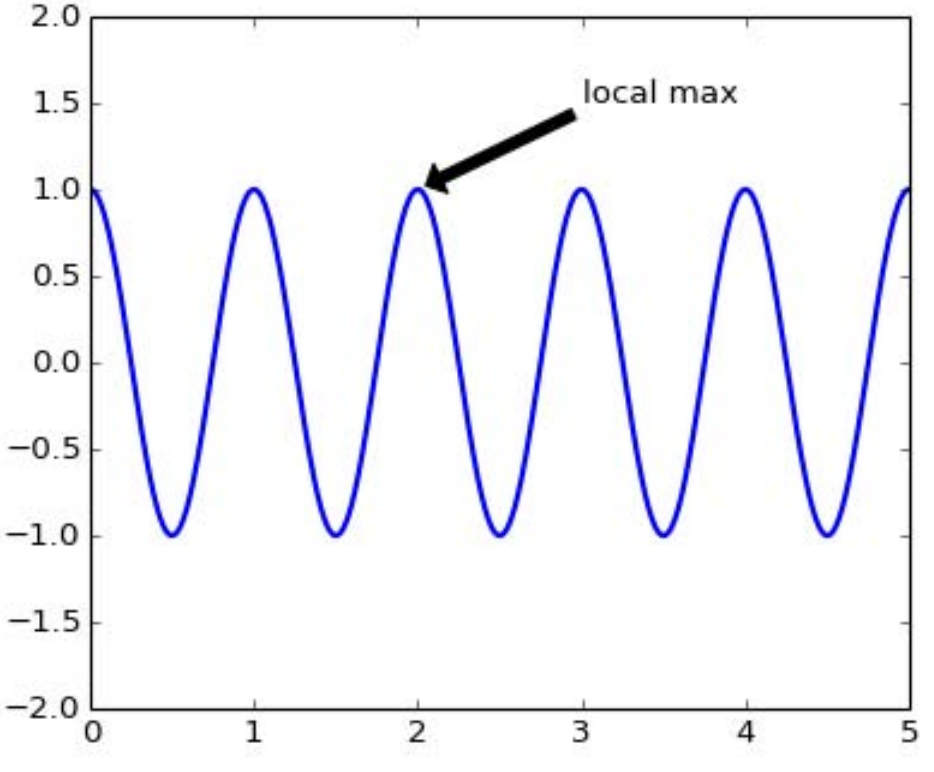

some numbers
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0

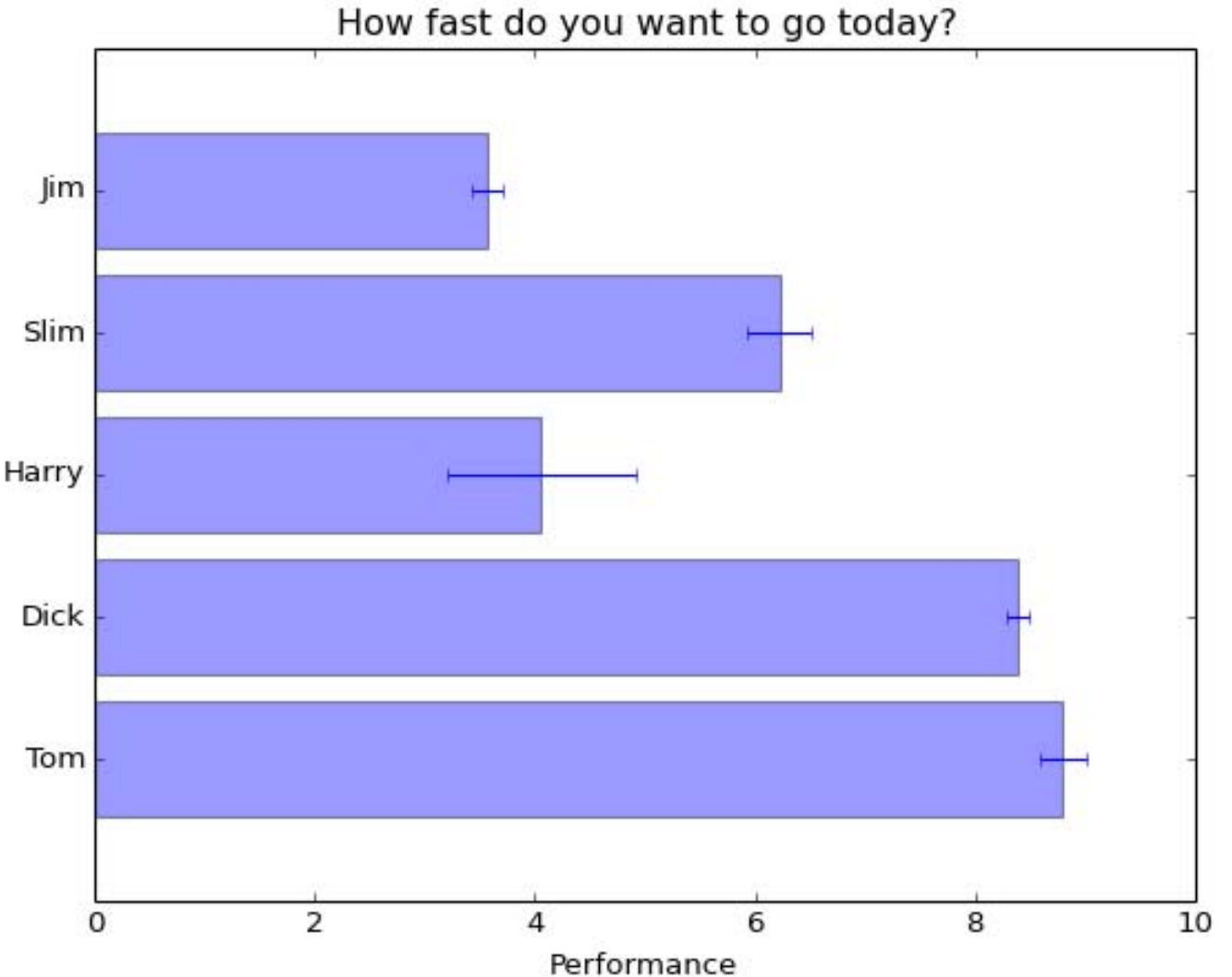

20
15
10
5
0
0
1
2
3
4
5
6

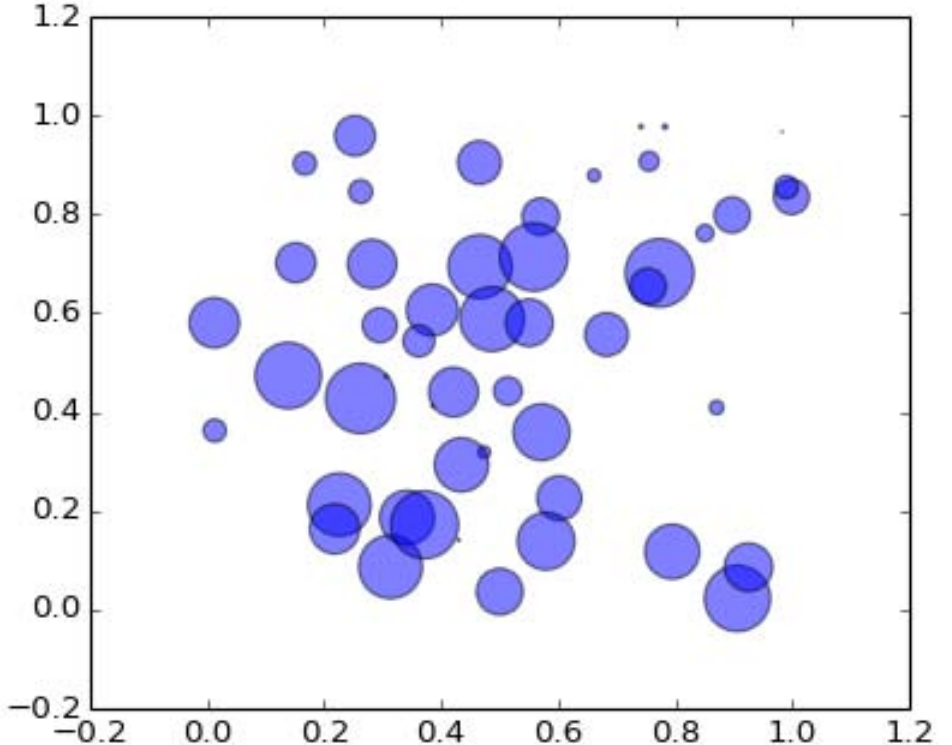

120
100
80
60
40
20
0
0
1
2
3
4
5

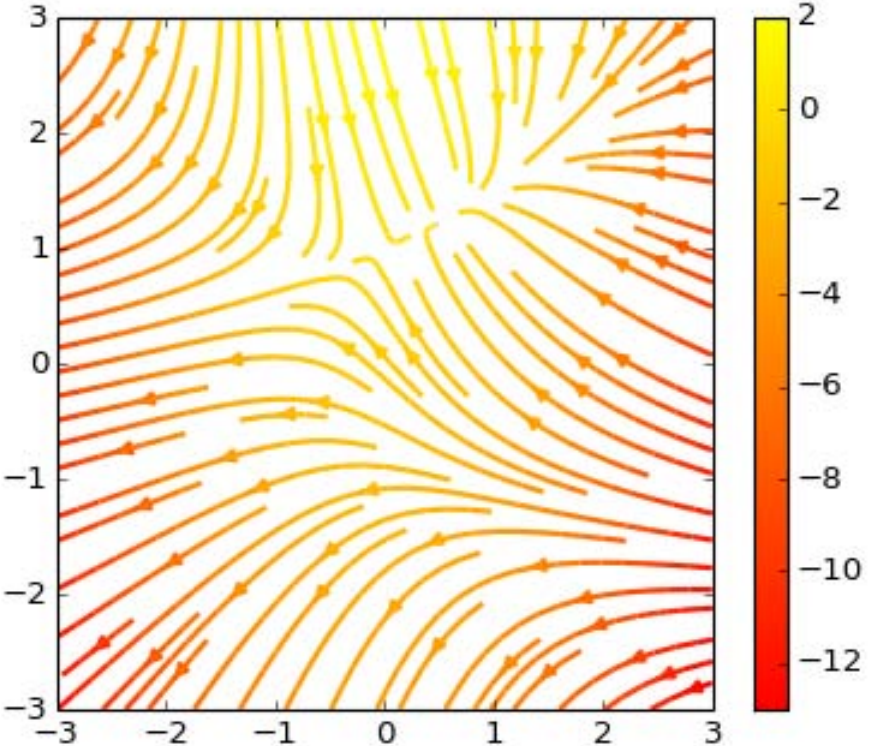

1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
0
1
2
3
4
5
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
4
5

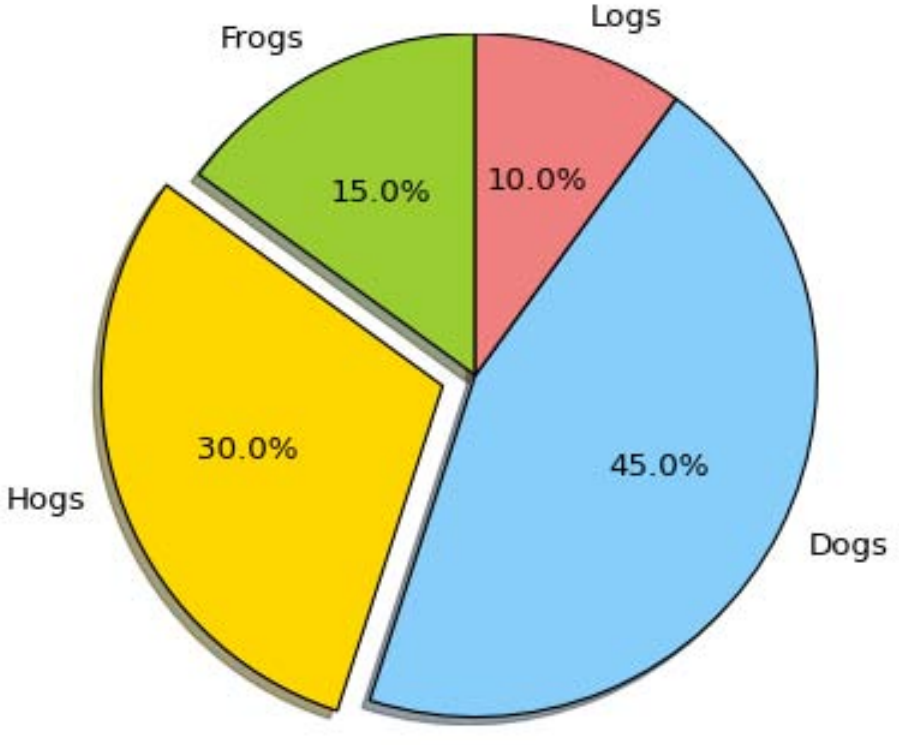

Histogram of IQ
$\mu=100, \sigma=15$
Probability
0.030
0.025
0.020
0.015
0.010
0.005
0.000
40
60
80
100
120
140
160
Smarts

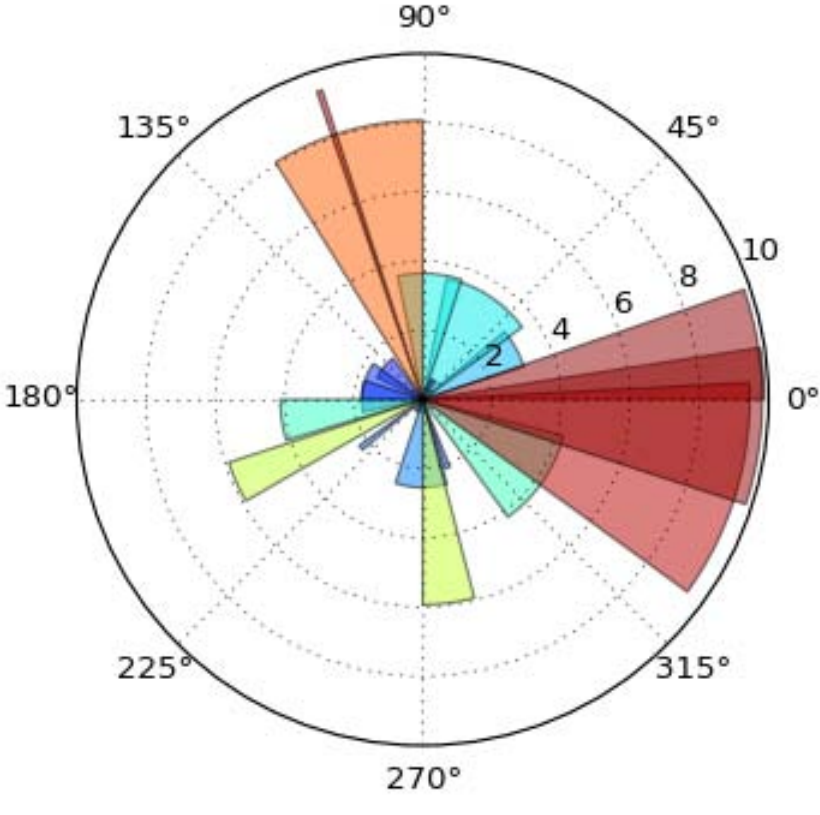

local max
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0
1
2
3
4
5

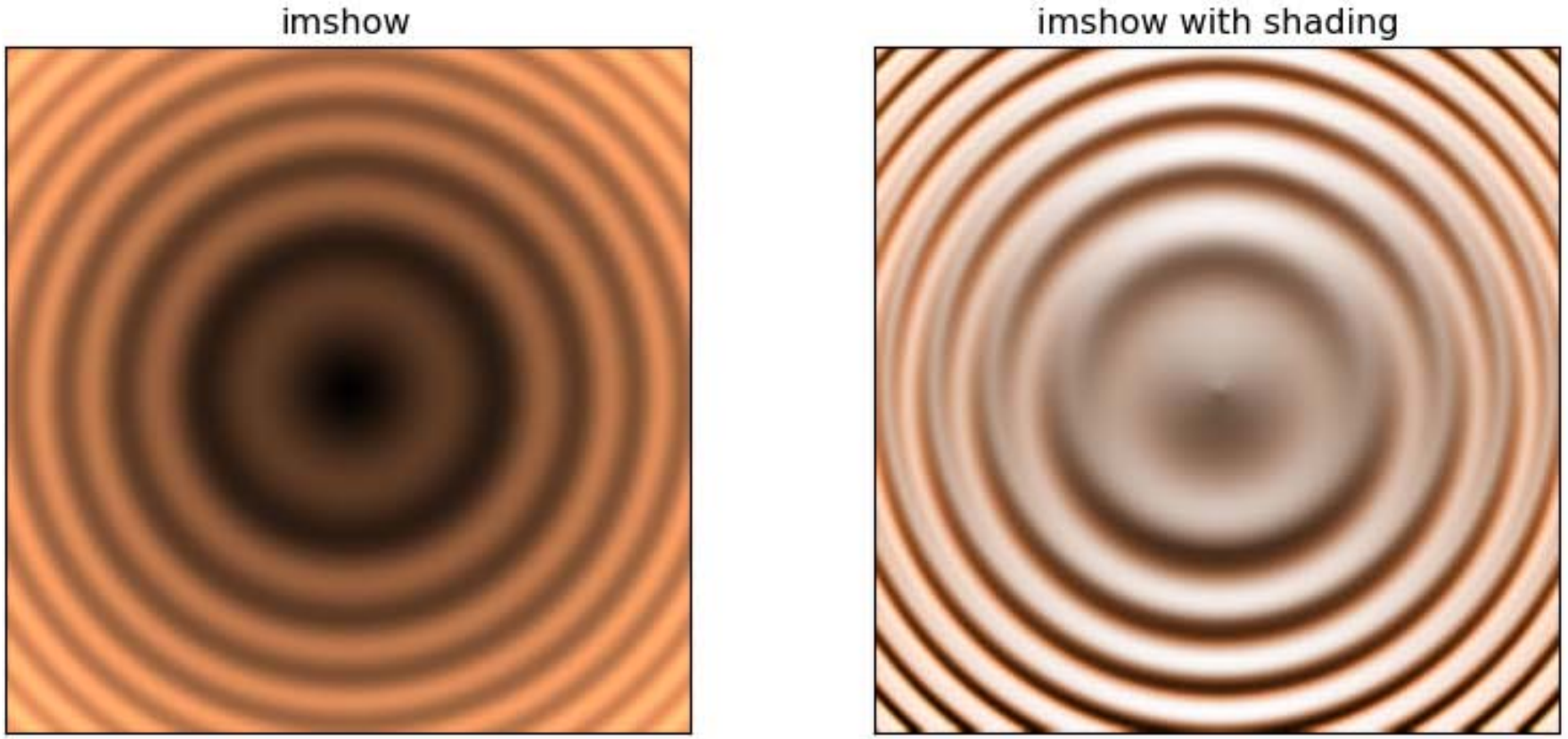

How fast do you want to go today?
Jim
Slim
Harry
Dick
Tom
0
2
4
6
8
10
Performance

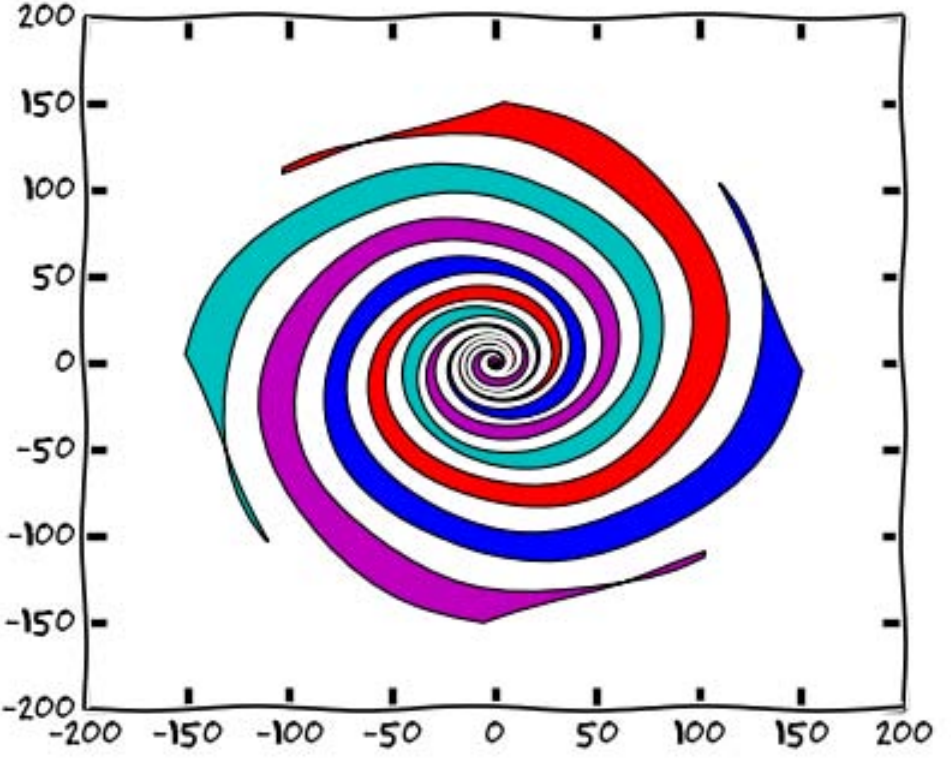

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2

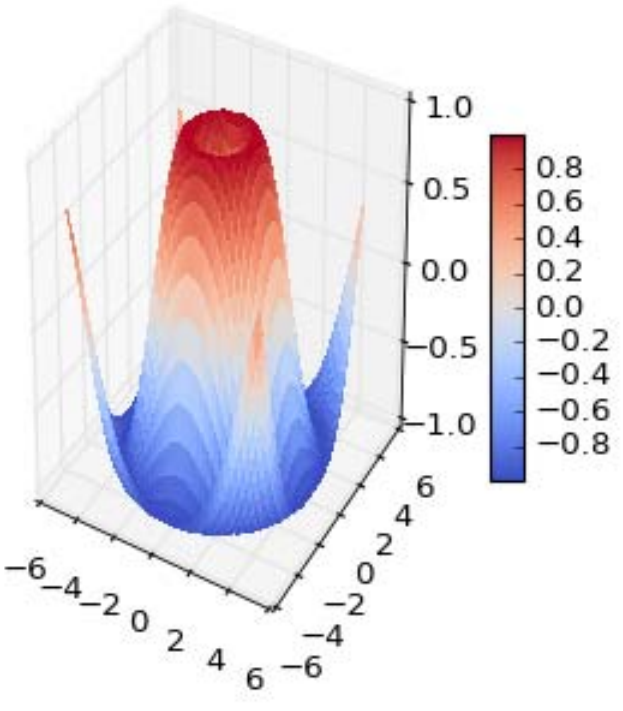

3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
2
0
−2
−4
−6
−8
−10
−12

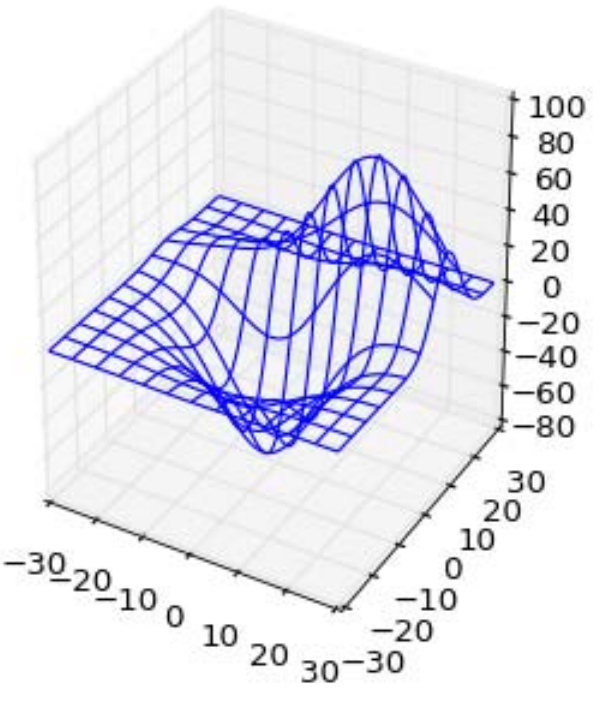

Frogs
Logs
15.0%
10.0%
30.0%
45.0%
Hogs
Dogs

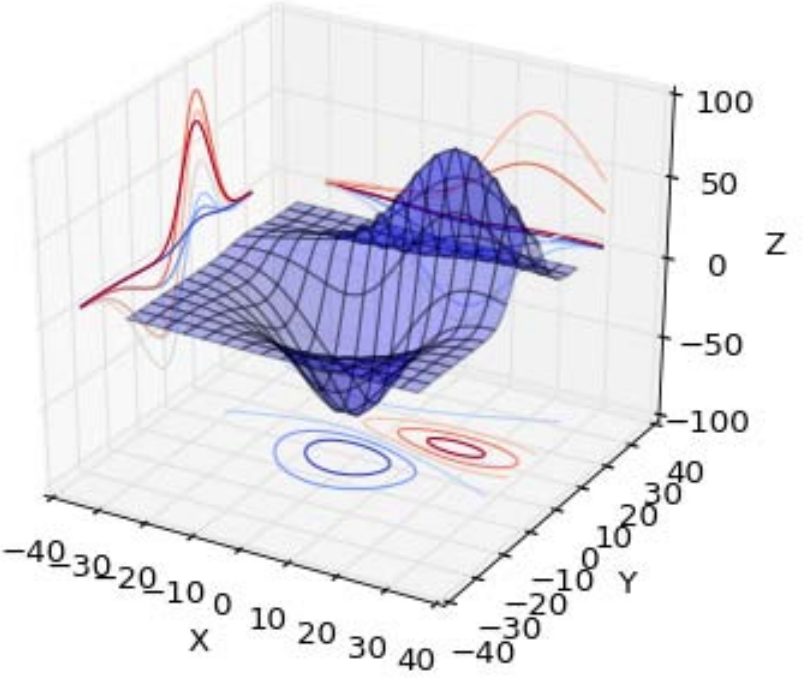

90°
135°
45°
180°
0°
225°
315°
270°
2
4
6
8
10


imshow
imshow with shading


200
150
100
50
0
-50
-100
-150
-200
-200
-150
-100
-50
0
50
100
150
200


1.0
0.5
0.0
−0.5
−1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−6 −4 −2 0 2 4 6
−6 −4 −2 0 2 4 6


100
80
60
40
20
0
−20
−40
−60
−80
−30 −20 −10 0 10 20 30
−30 −20 −10 0 10 20 30


100
50
0
−50
−100
Z
−40
−30
−20
−10
0
10
20
30
40
X
−40
−30
−20
−10
0
10
20
30
40
Y

IPy Home | IPy FirstNotebook

localhost:8890/notebooks/FirstNotebook.ipynb#

Apps | Google Maps | NYTimes | NYU | CSMR | Journals | Web of Science | Other Bookmarks

IP[y]: Notebook FirstNotebook (unsaved changes)

File Edit View Insert Cell Kernel Help

Code | Cell Toolbar: None

In [1]:
```
2+3
```

Out[1]: 5

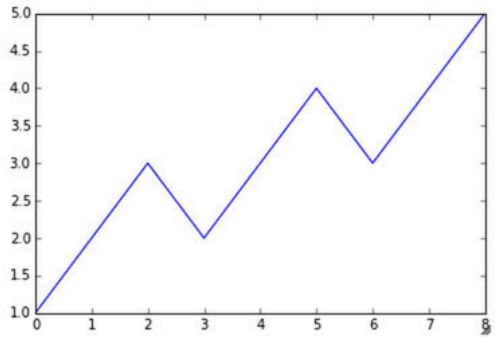
In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
np.sin(np.pi/6)
```

Out[3]: 0.49999999999999994

In [4]:
```
plt.plot([1,2,3,2,3,4,3,4,5])
```

Out[4]: [<matplotlib.lines.Line2D at 0x10cfc8b90>]



In [ ]:

Simple spectral analysis

An illustration of the Discrete Fourier Transform

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

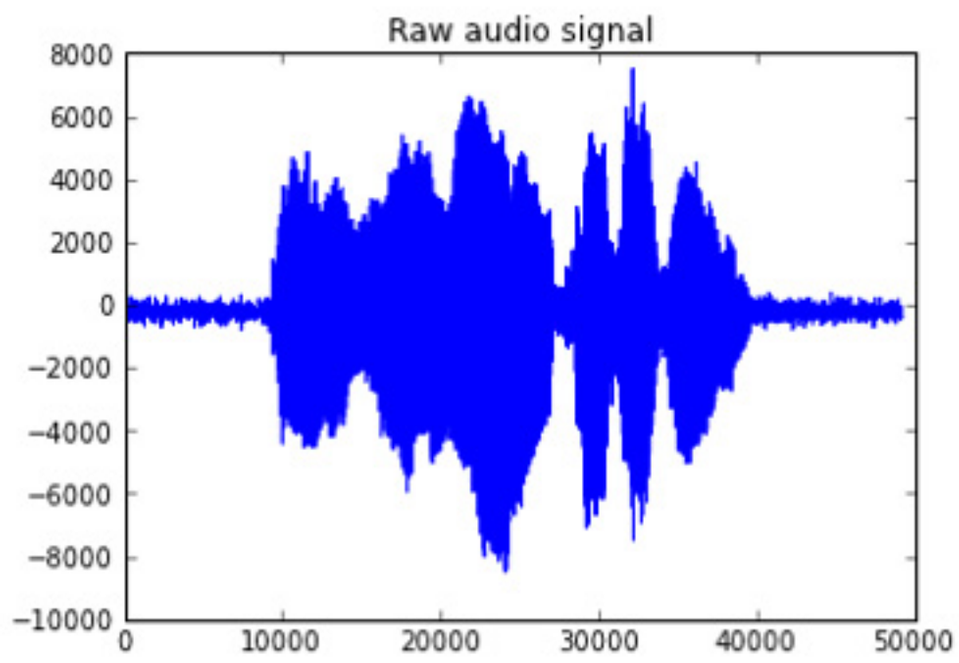
using windowing, to reveal the frequency content of a sound signal.

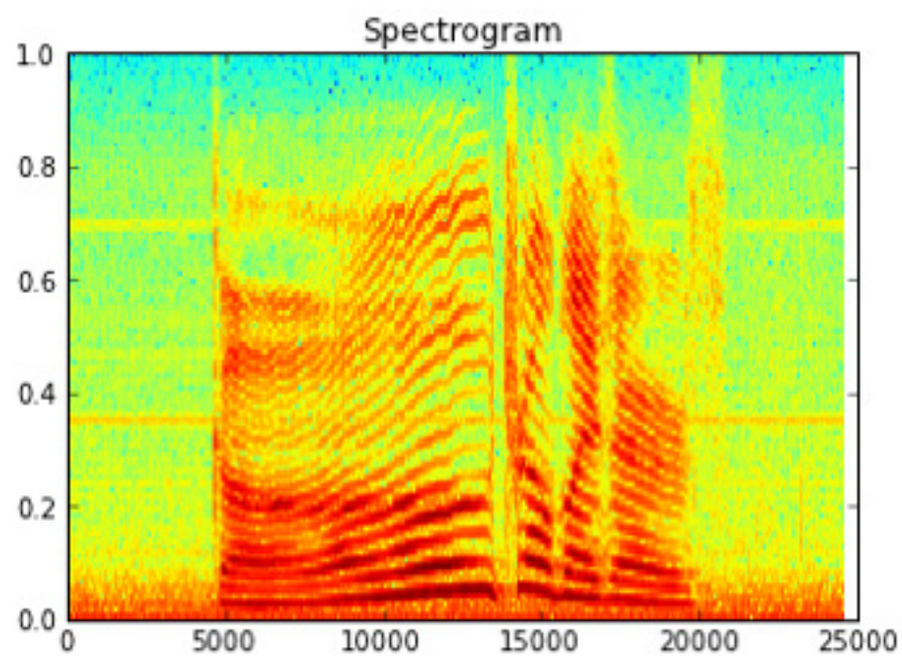
We begin by loading a datafile using SciPy's audio file support:

In [1]:
```
from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin `specgram` routine:

In [2]:
```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```





IPy examples/Interactive%20W × | IPy Exploring Graphs ×

127.0.0.1:8888/notebooks/examples/Interactive%20Widgets/Exploring%20Graphs.ip...

File Edit View Insert Cell Kernel Help

In [5]:
```
interact(plot_random_graph, n=(2,30), m=(1,10), k=(1,10), p=(0.0, 1.0, 0.001),
        generator={'lobster': random_lobster,
                   'power law': powerlaw_cluster,
                   'Newman-Watts-Strogatz': newman_watts_strogatz,
                   u'Erdős-Rényi': erdos_renyi,
                   });
```

n 16

m 5

k 5

p 0.527

generator Erdős-Rényi