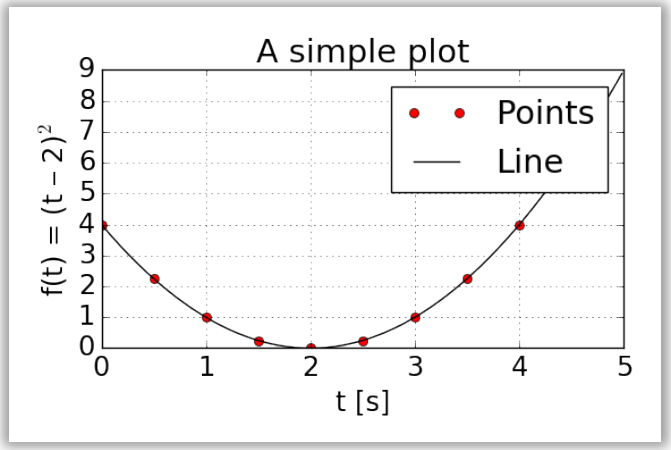
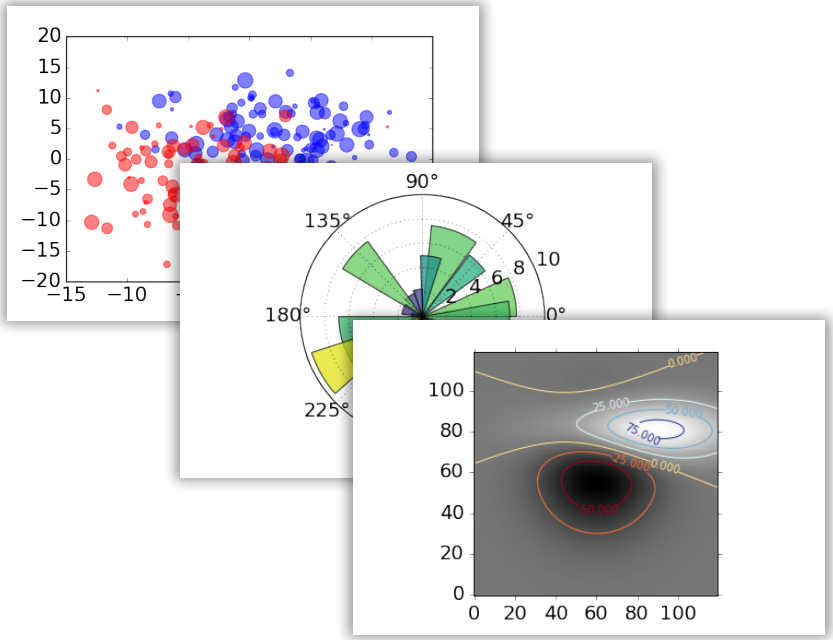
Python Bootcamp
Data Display with MatPlotLib

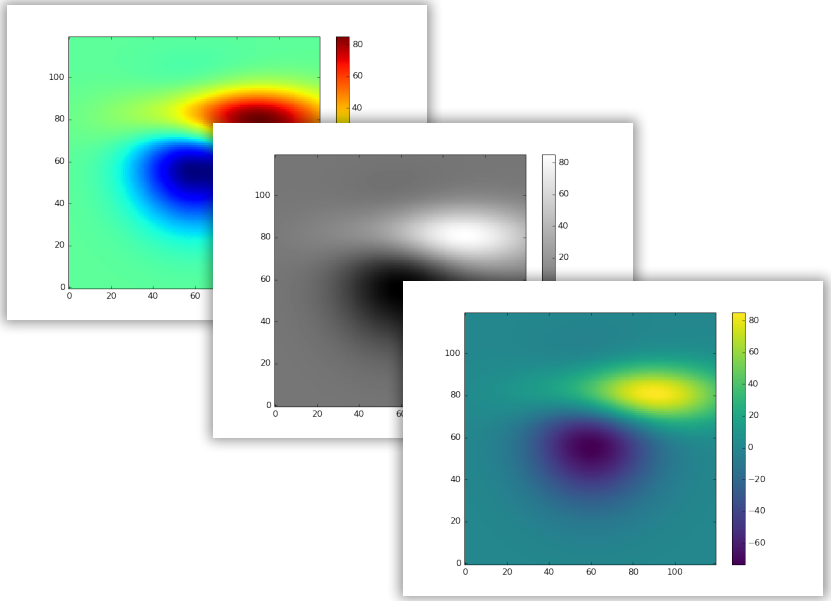
PhD Bruno C. Quint
bquint@ctio.noao.edu
Resident Astronomer at SOAR Telescope
https://github.com/b1quint/PythonBootcamp2017

Table of Contents

- A simple plot
- Types of plots
- Different styles

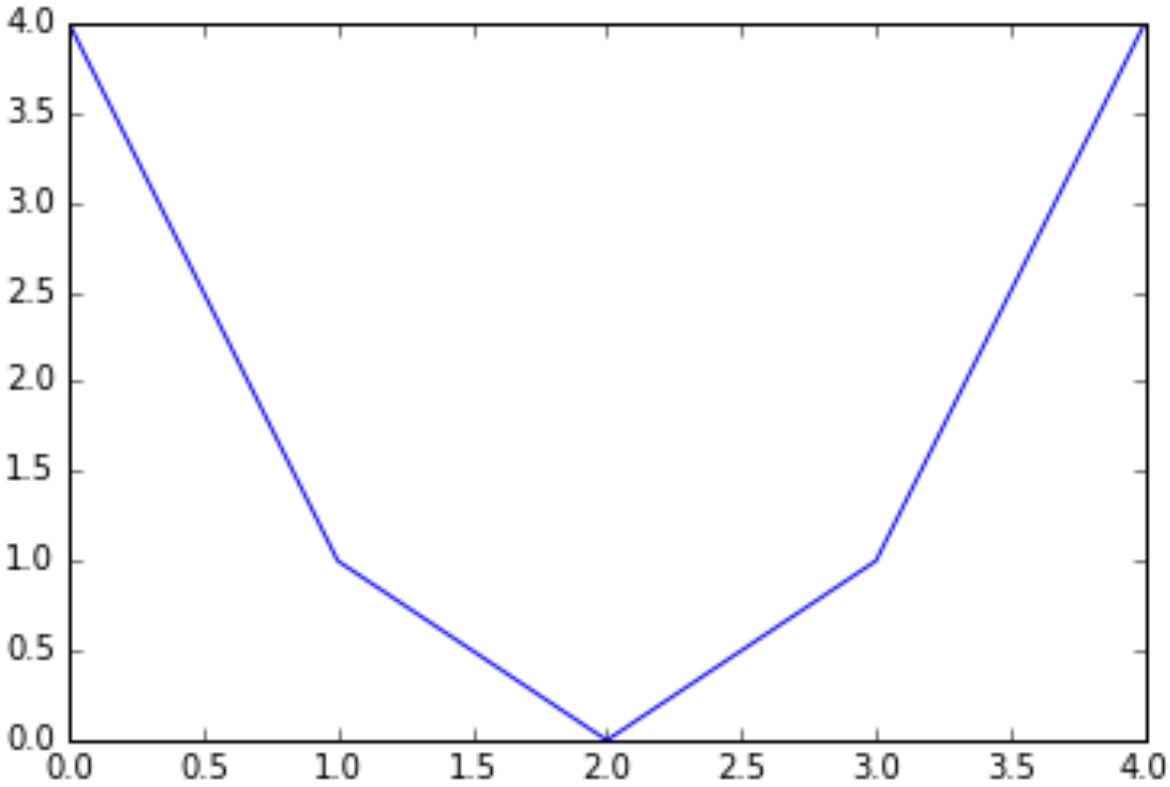






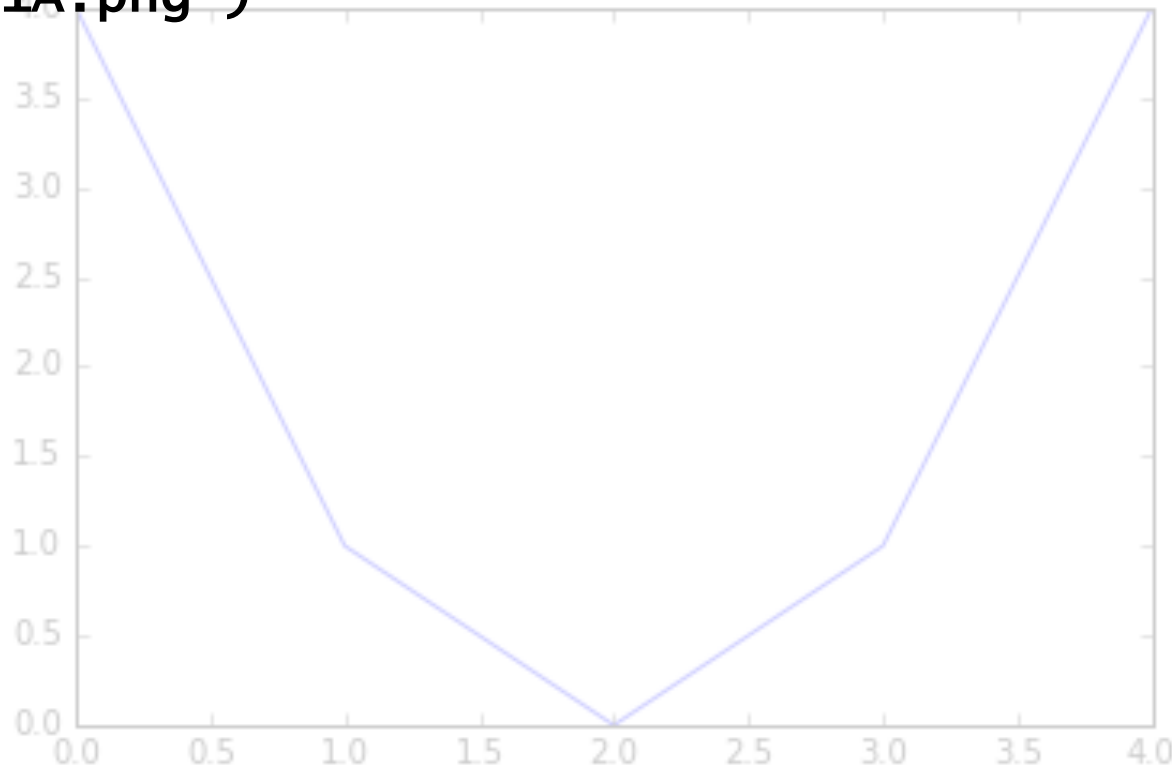
A simple plot My First Example

```
from matplotlib.pyplot import *
x = [0, 1, 2, 3, 4]
y = [4, 1, 0, 1, 4]
plot(x, y)
show()
```



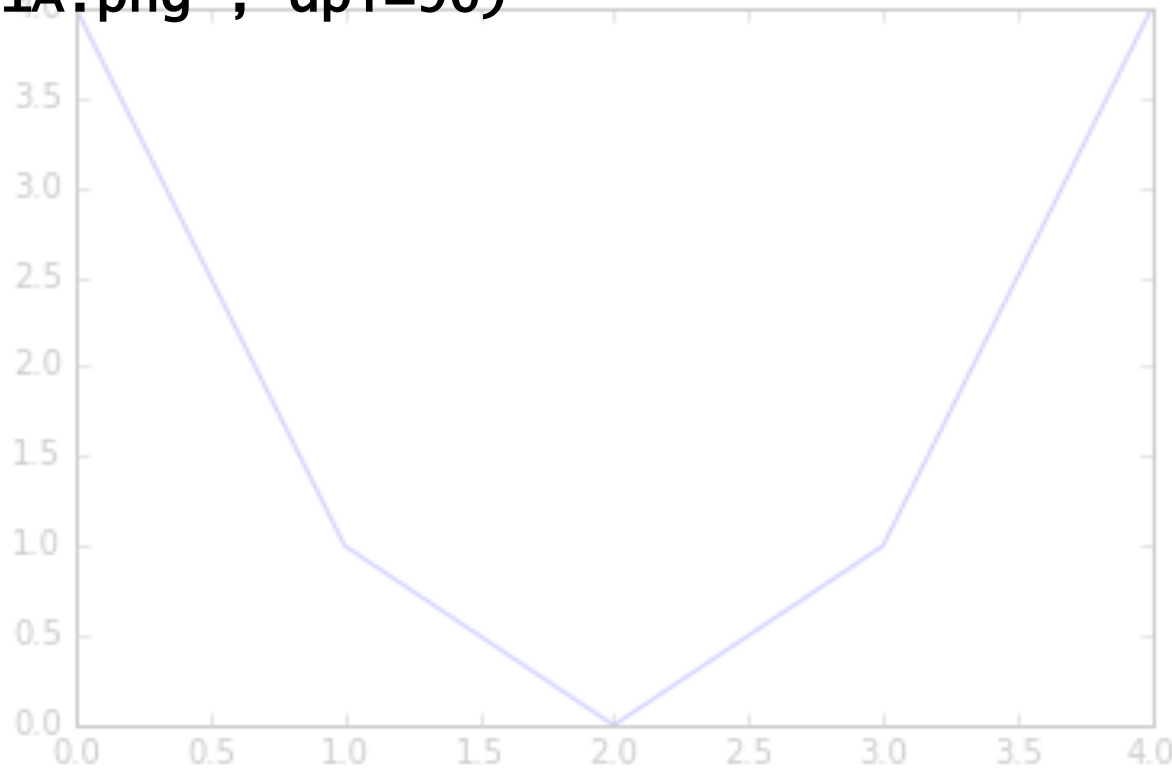
A simple plot Save the Image!

```
from matplotlib.pyplot import *
x = [0, 1, 2, 3, 4]
y = [4, 1, 0, 1, 4]
plot(x, y)
savefig('plot_001A.png')
```



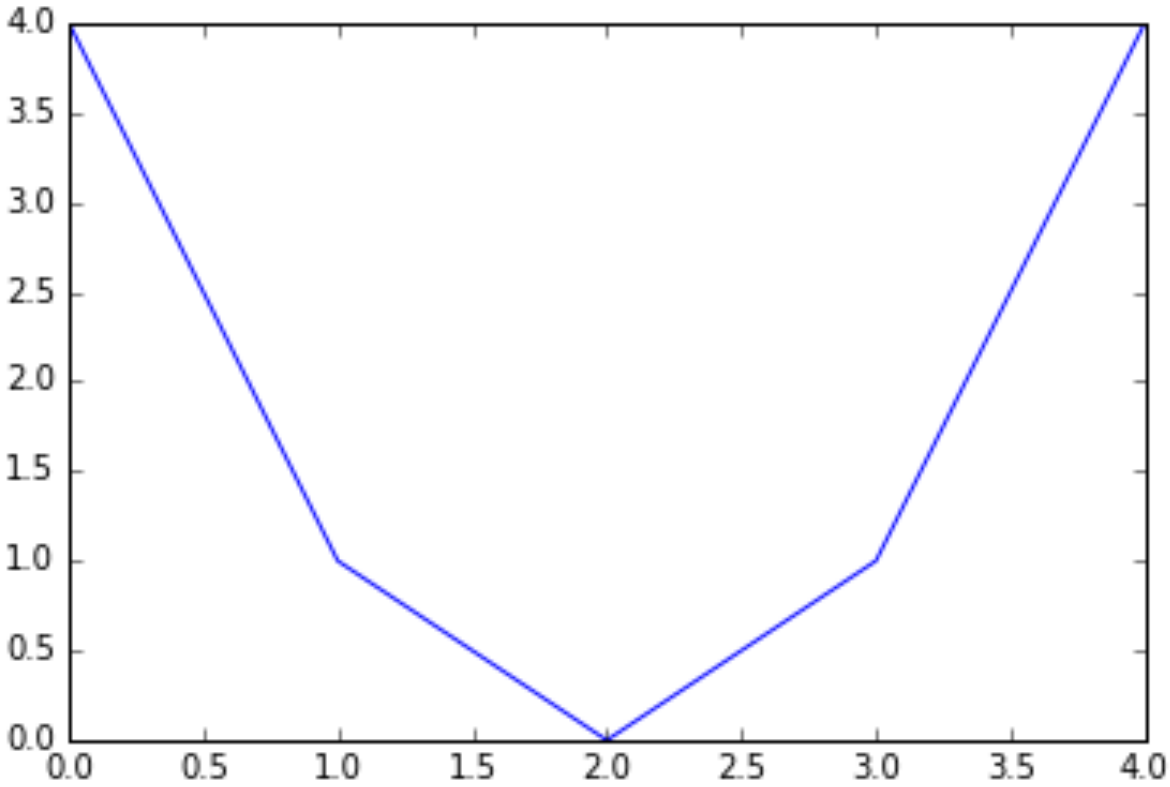
A simple plot Save the Image!

```
from matplotlib.pyplot import *
x = [0, 1, 2, 3, 4]
y = [4, 1, 0, 1, 4]
plot(x, y)
savefig('plot_001A.png', dpi=96)
```



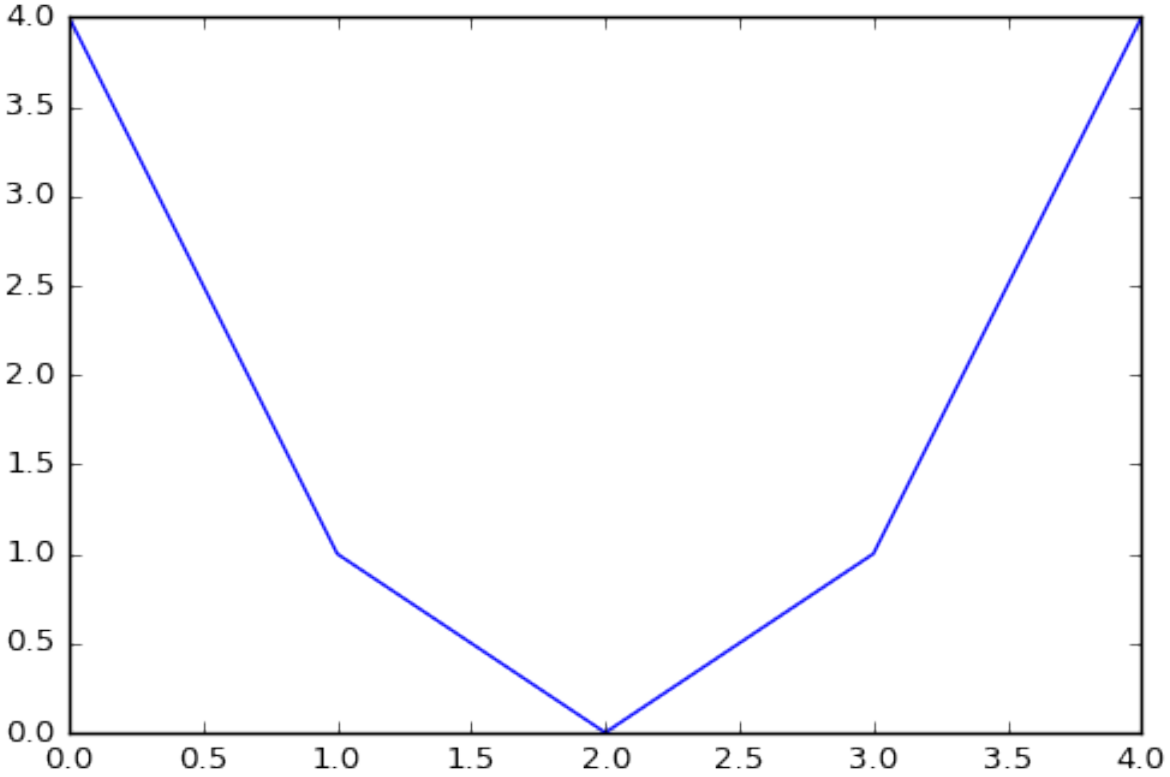
A simple plot Let's do it the "right" way

```
import matplotlib.pyplot as plt
x = [0, 1, 2, 3, 4]
y = [4, 1, 0, 1, 4]
plt.plot(x, y)
plt.show()
```

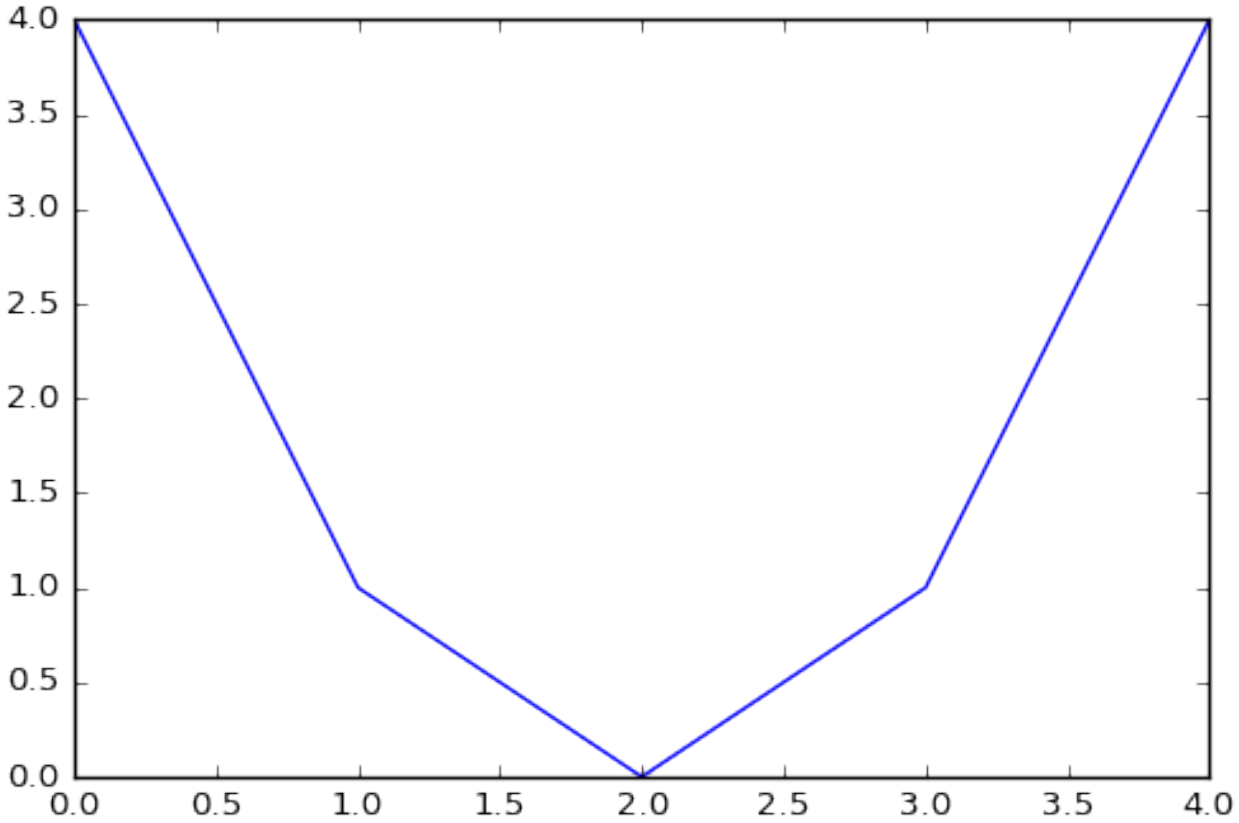


A simple plot Let's do it the "right" way

```
import matplotlib.pyplot as plt
import numpy as np

x = np.arange(5)
y = (x - 2) ** 2

plt.plot(x, y)
plt.show()
```



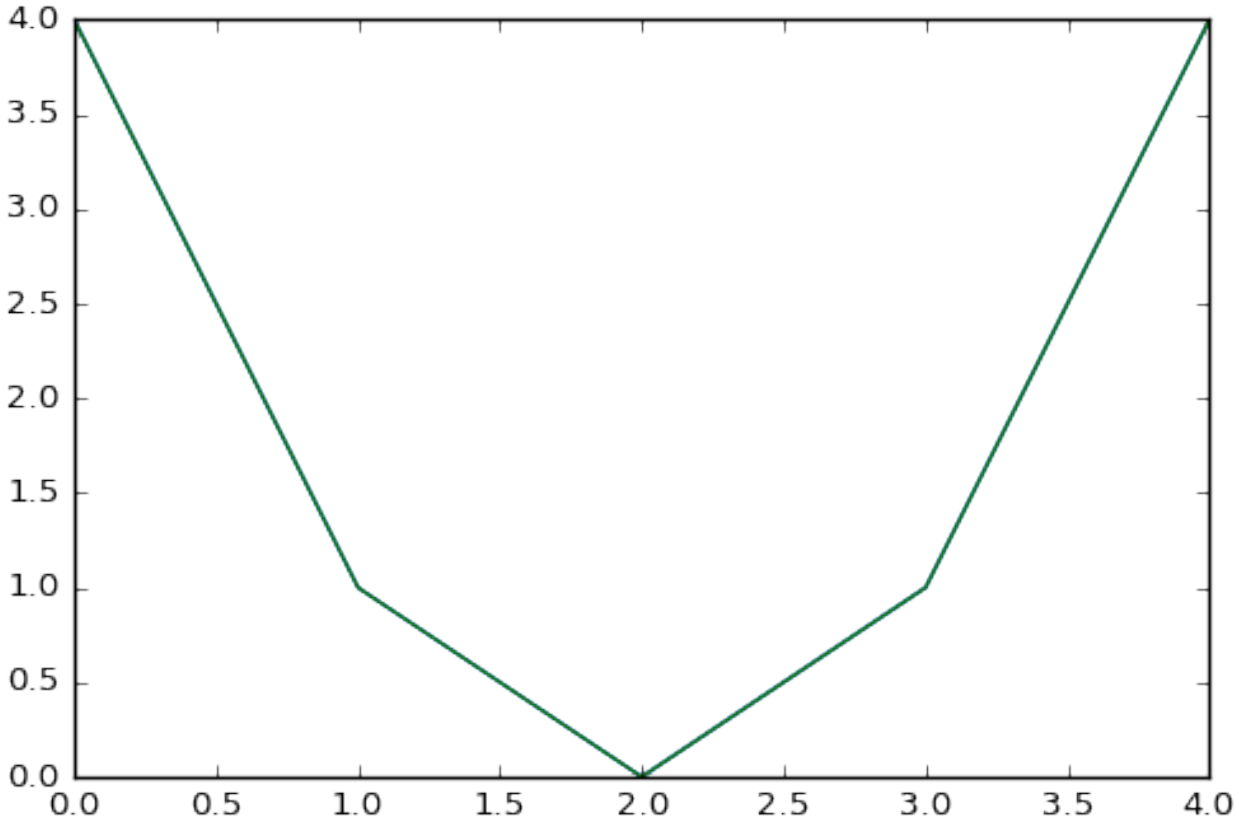
A simple plot Plot and plot again…




```
plt.plot(x, y)
```

```
x = np.arange(5)
y = x ** 2 - 4 * x + 4
```

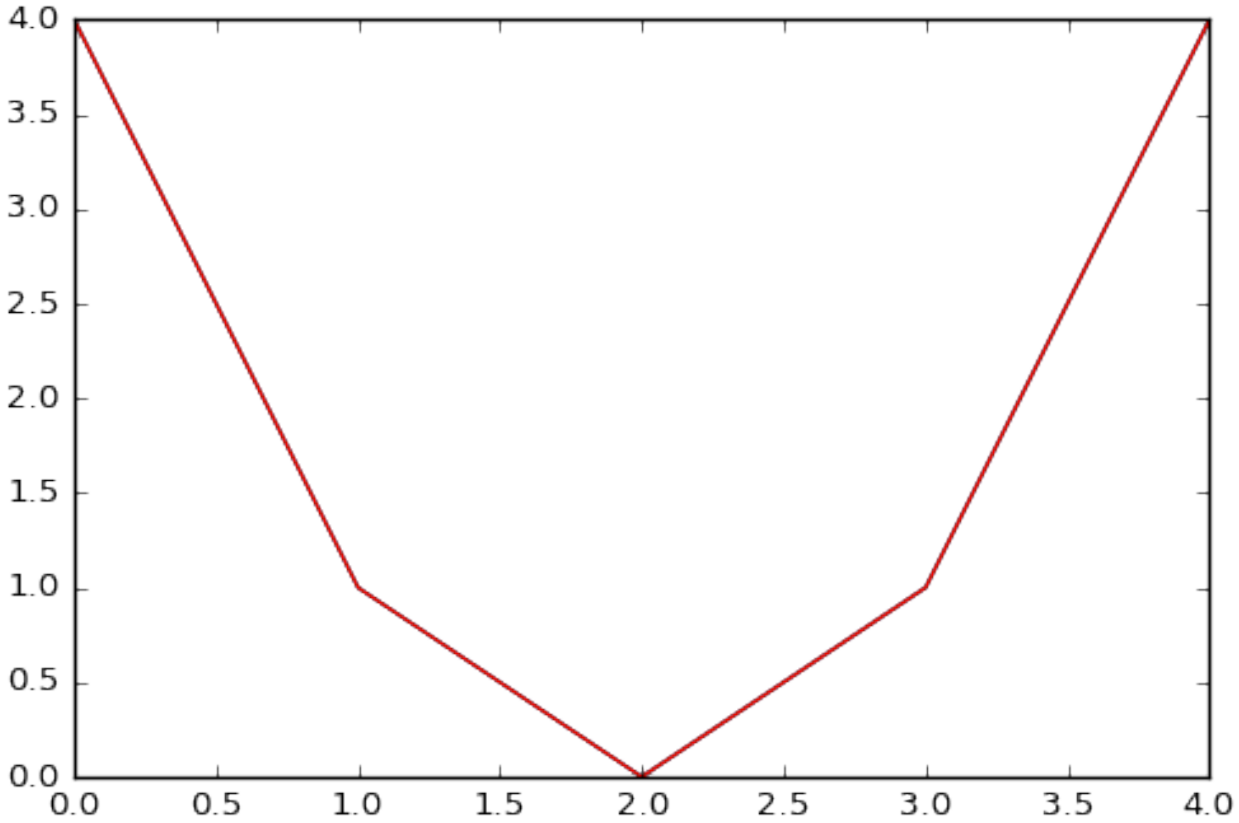
A simple plot Plot and plot again…




```
x = np.arange(5)
y = x ** 2 - 4 * x + 4
```

```
plt.plot(x, y)
plt.plot(x, y)
```

A simple plot Plot and plot again…

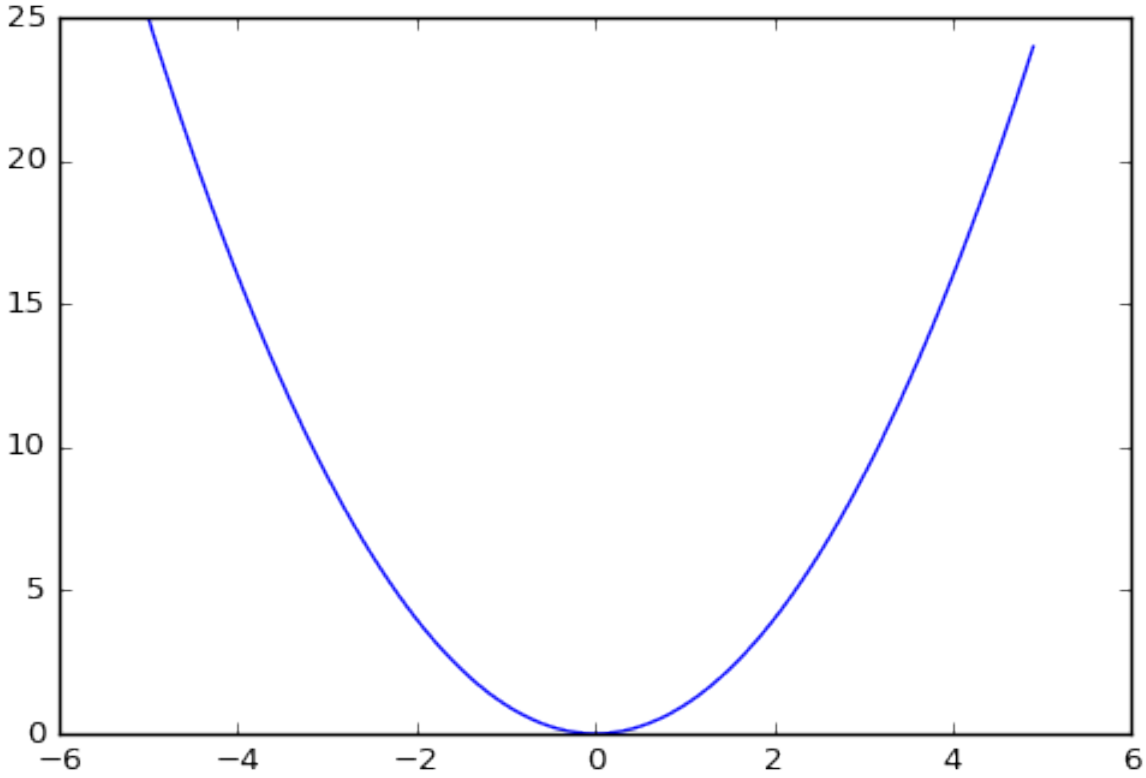



```
x = np.arange(5)
y = x ** 2 - 4 * x + 4
```

```
plt.plot(x, y)
plt.plot(x, y)
plt.plot(x, y)
```

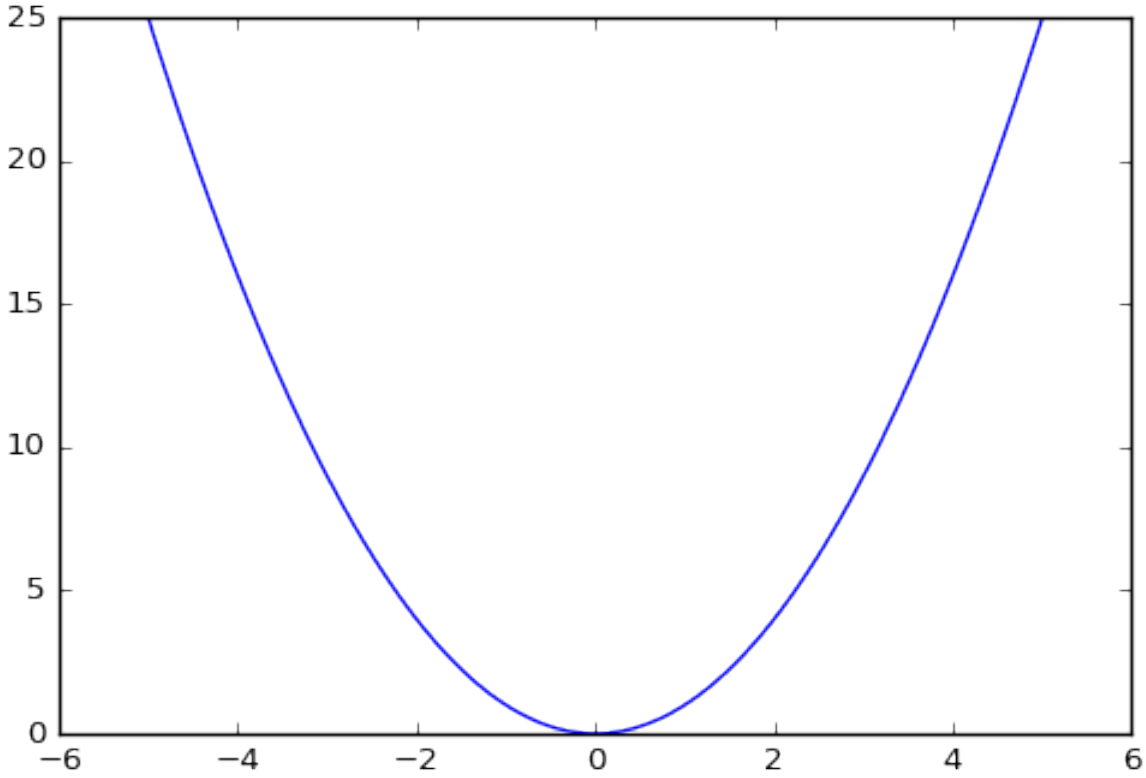
A simple plot Let's do it the "right" way

```
import matplotlib.pyplot as plt
import numpy as np

x = np.arange(-5, 5, 0.1)
y = (x - 2) ** 2

plt.plot(x, y)
plt.show()
```



A simple plot Let's do it the "right" way

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-5, 5, 100)
y = (x - 2) ** 2

plt.plot(x, y)
plt.show()
```



A simple plot Plot with style

```
def f(t):
    return (t - 2) ** 2
```

A simple plot Plot with style

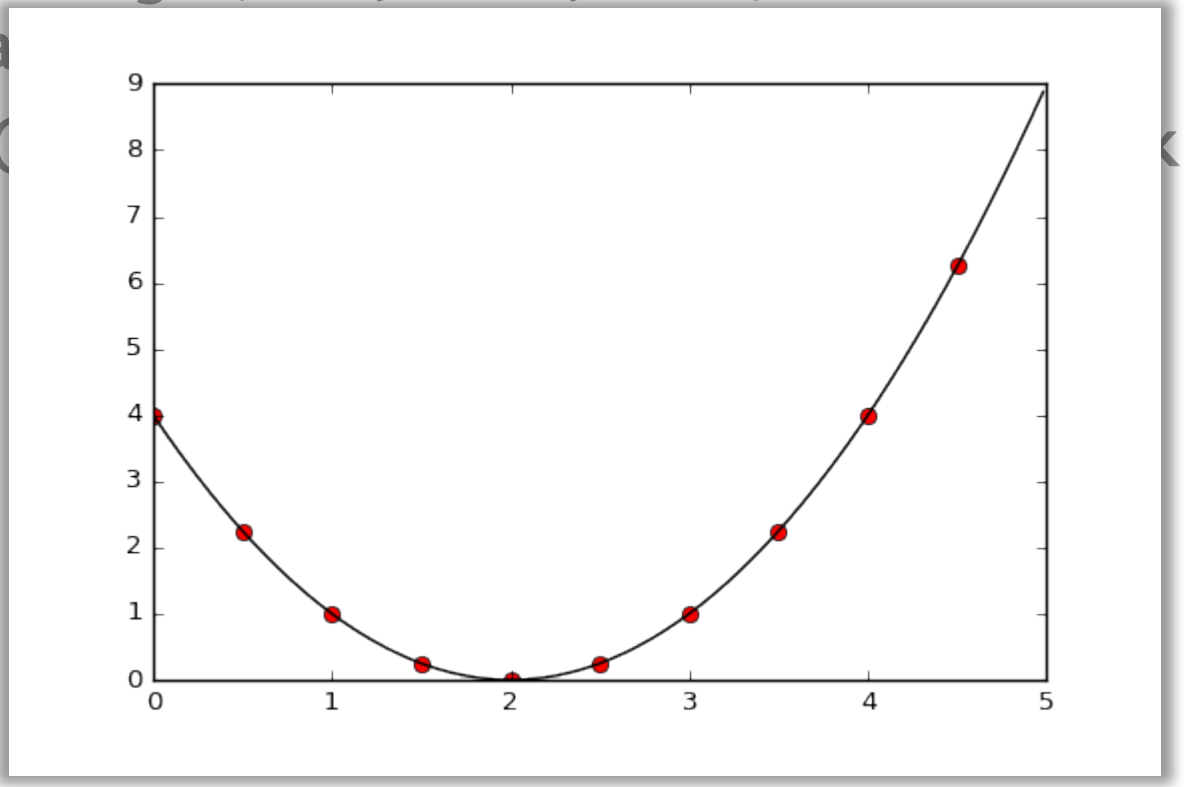
```
def f(t):
    return (t - 2) ** 2
t1 = np.arange(0.0, 5.0, 0.5)
t2 = np.arange(0.0, 5.0, 0.02)
```

A simple plot Plot with style

```
def f(t):
    return (t - 2) ** 2
t1 = np.arange(0.0, 5.0, 0.5)
t2 = np.arange(0.0, 5.0, 0.02)
plt.plot(t1, f(t1), 'ro’)
plt.plot(t2, f(t2), 'k')
```

A simple plot Plot with style

```
def f(t):
    return (t - 2) ** 2
t1 = np.arange(0.0, 5.0, 0.5)
t2 = np.a
plt.plot(                                    k')
```

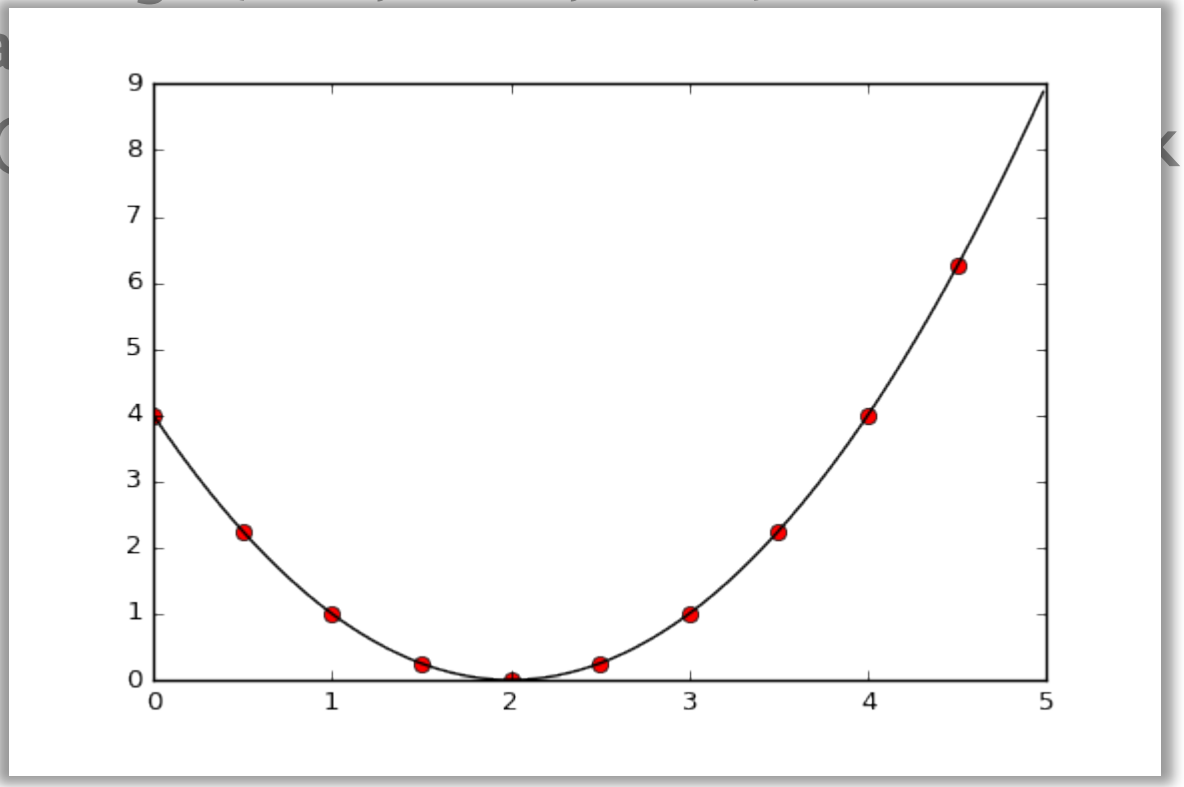


A simple plot Plot with style

```
def f(t):
    return (t - 2) ** 2
t1 = np.arange(0.0, 5.0, 0.5)
t2 = np.arange(0.0, 5.0, 0.02)
plt.plot(t1, f(t1), 'ro', t2, f(t2), 'k')
```

A simple plot Plot with style

```
def f(t):
    return (t - 2) ** 2
t1 = np.arange(0.0, 5.0, 0.5)
t2 = np.a
plt.plot(                                    k')
```

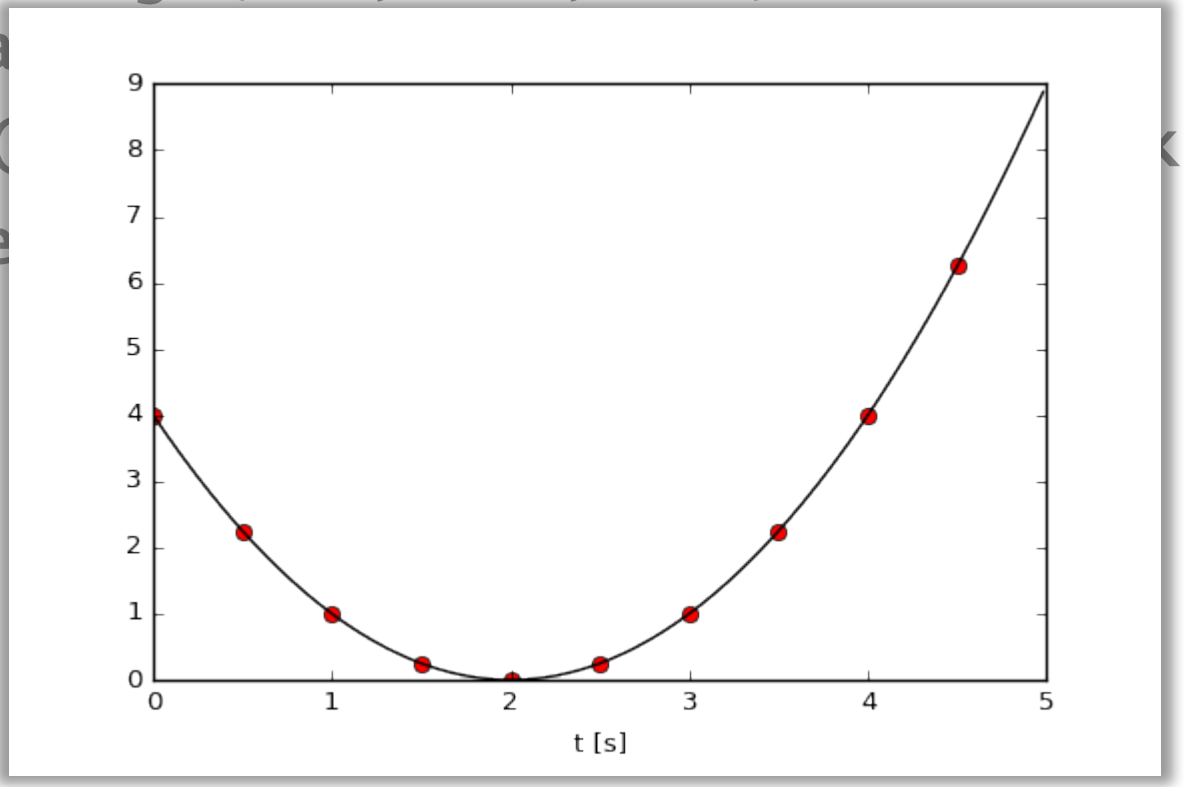


A simple plot Plot with style

```
def f(t):
    return (t - 2) ** 2
t1 = np.arange(0.0, 5.0, 0.5)
t2 = np.arange(0.0, 5.0, 0.02)
plt.plot(t1, f(t1), 'ro', t2, f(t2), 'k')
plt.xlabel("t [s]")
```

A simple plot Plot with style

```
def f(t):
    return (t - 2) ** 2
t1 = np.arange(0.0, 5.0, 0.5)
t2 = np.a
plt.plot(                                    k')
plt.xlabe
```

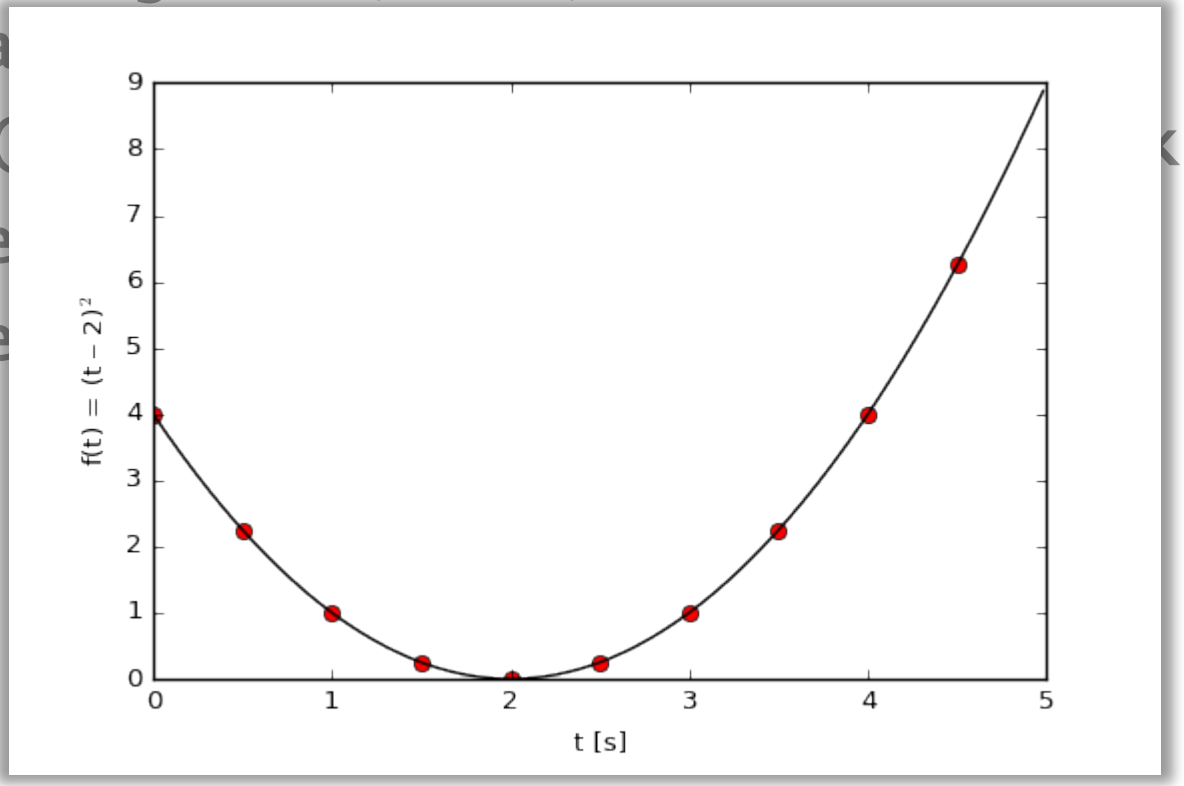


A simple plot Plot with style

```
def f(t):
    return (t - 2) ** 2
t1 = np.arange(0.0, 5.0, 0.5)
t2 = np.arange(0.0, 5.0, 0.02)
plt.plot(t1, f(t1), 'ro', t2, f(t2), 'k')
plt.xlabel(“t [s]”)
plt.ylabel(u"f(t) = (t - 2)$^2$")
```

A simple plot Plot with style

```
def f(t):
    return (t - 2) ** 2
t1 = np.arange(0.0, 5.0, 0.5)
t2 = np.a
plt.plot(                                    k')
plt.xlabe
plt.ylabe
```

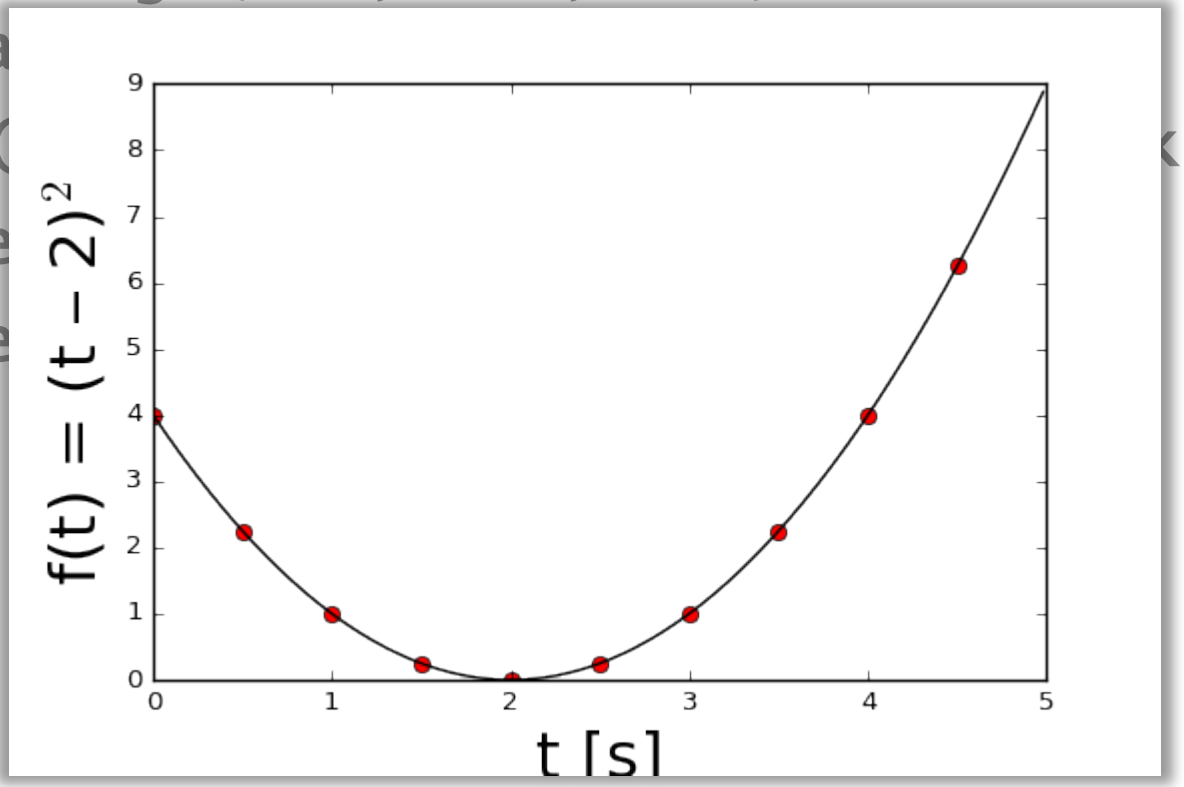


A simple plot Plot with style

```
def f(t):
    return (t - 2) ** 2
t1 = np.arange(0.0, 5.0, 0.5)
t2 = np.arange(0.0, 5.0, 0.02)
plt.plot(t1, f(t1), 'ro', t2, f(t2), 'k')
plt.xlabel("t [s]", fontsize=24)
plt.ylabel(u"f(t) = (t - 2)$^2$",
           fontsize=24)
```

A simple plot Plot with style

```
def f(t):
    return (t - 2) ** 2
t1 = np.arange(0.0, 5.0, 0.5)
t2 = np.a
plt.plot(                                    k')
plt.xlabe
plt.ylabe
```

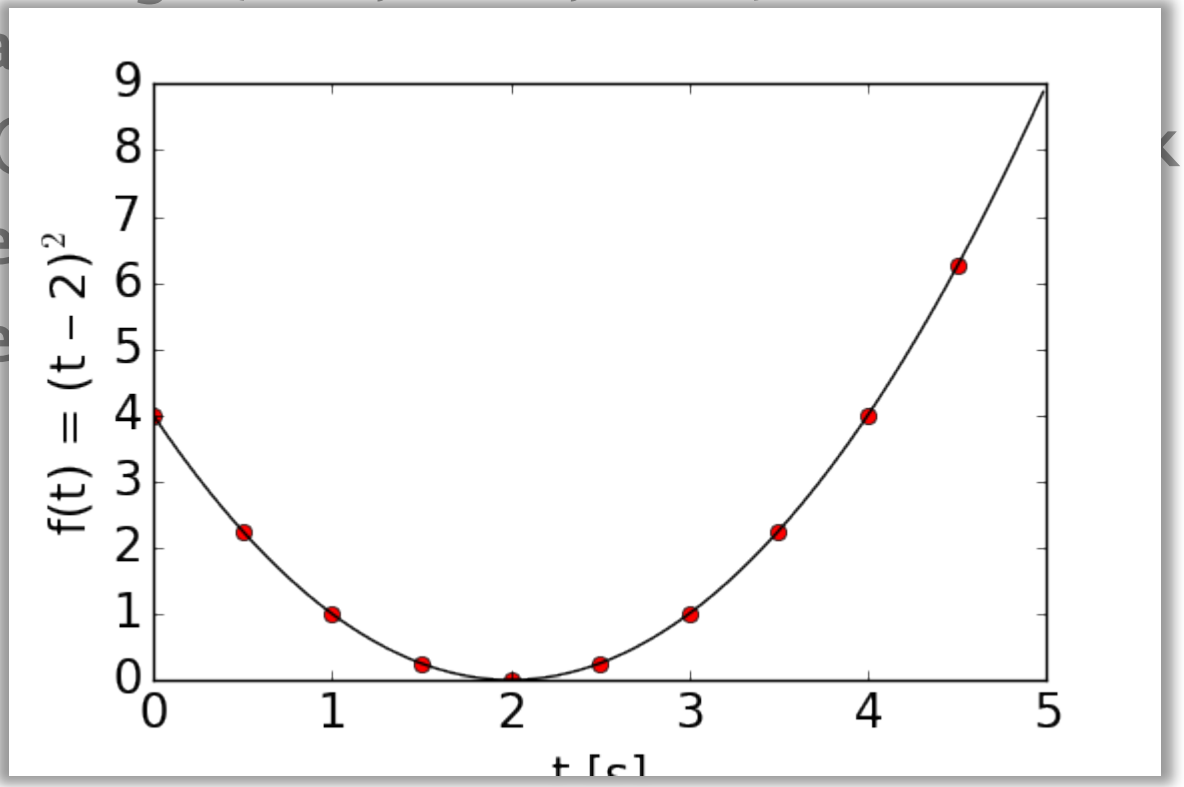


A simple plot Plot with style

```
def f(t):
    return (t - 2) ** 2
t1 = np.arange(0.0, 5.0, 0.5)
t2 = np.arange(0.0, 5.0, 0.02)
plt.plot(t1, f(t1), 'ro', t2, f(t2), 'k')
plt.xlabel("t [s]", fontsize=24)
plt.ylabel(u"f(t) = (t - 2)$^2$",
           fontsize=24)
plt.xticks(fontsize=18)
```

A simple plot Plot with style

```
def f(t):
    return (t - 2) ** 2
t1 = np.arange(0.0, 5.0, 0.5)
t2 = np.a
plt.plot(                                    k')
plt.xlabe
plt.ylabe
```

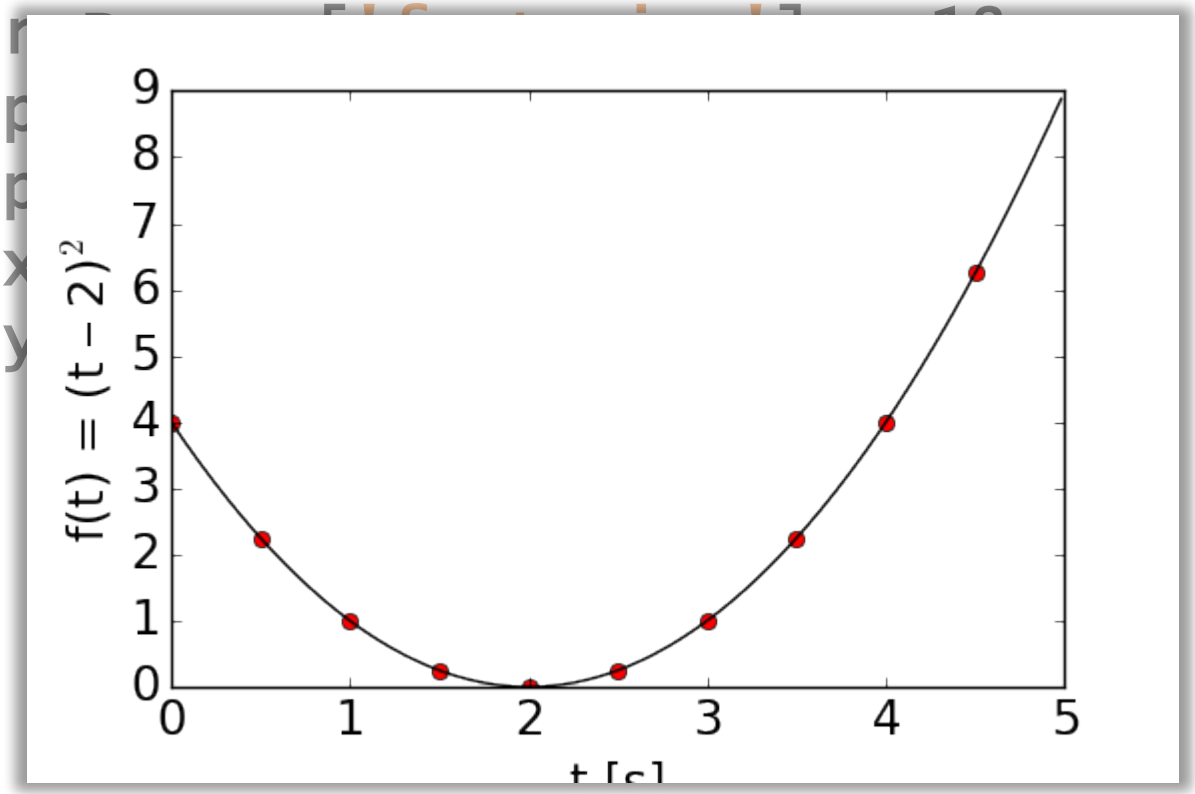


A simple plot Plot with style

```
>>> import matplotlib as mpl
>>> print mpl.rcParams['font.size']
10
>>> mpl.rcParams['font.size'] = 18
>>> plt.plot(t1, f(t1), 'ro')
>>> plt.plot(t2, f(t2), 'k')
>>> plt.xlabel('t [s]')
>>> plt.ylabel(u'f(t) = (t - 2)$^s$')
```

A simple plot Plot with style

```
>>> import matplotlib as mpl
>>> print mpl.rcParams['font.size']
10
>>> mpl.r
>>> plt.p
>>> plt.p
>>> plt.x
>>> plt.y
```

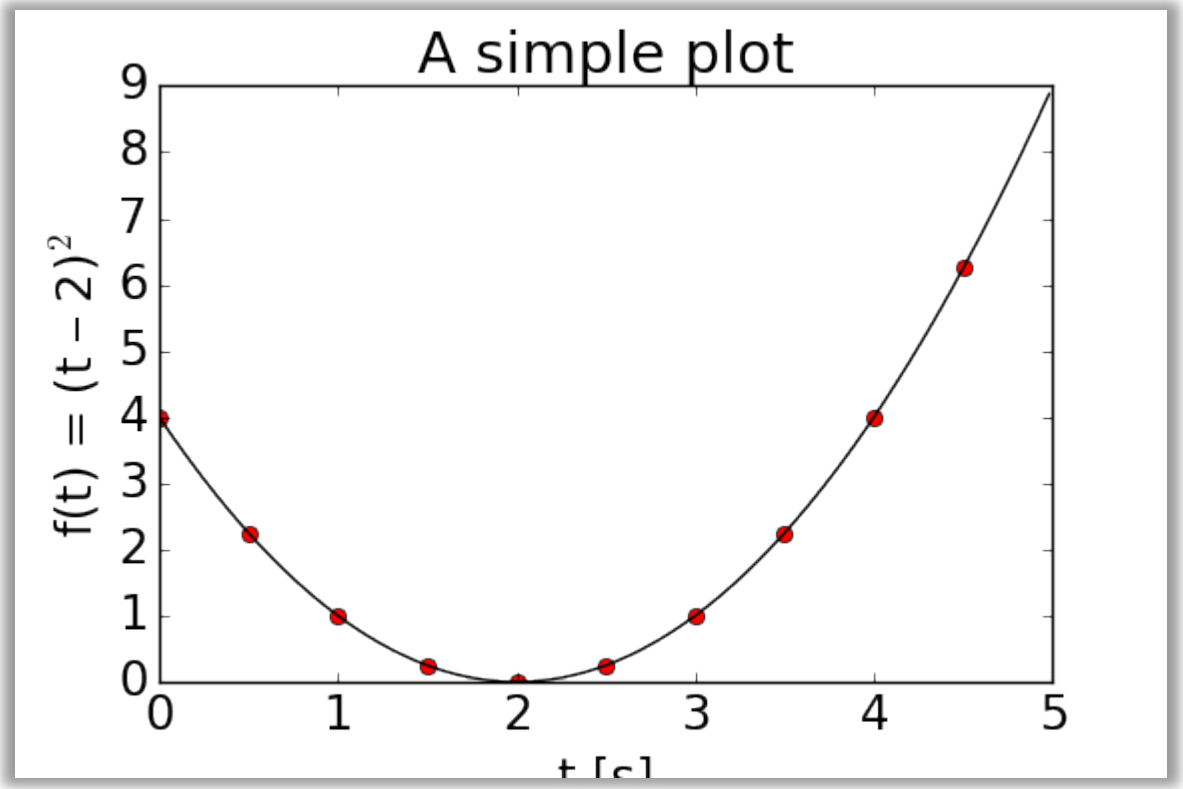


A simple plot Plot with style

```
>>> plt.title("A simple plot")
```

A simple plot Plot with style

```
>>> plt.title("A simple plot")
```

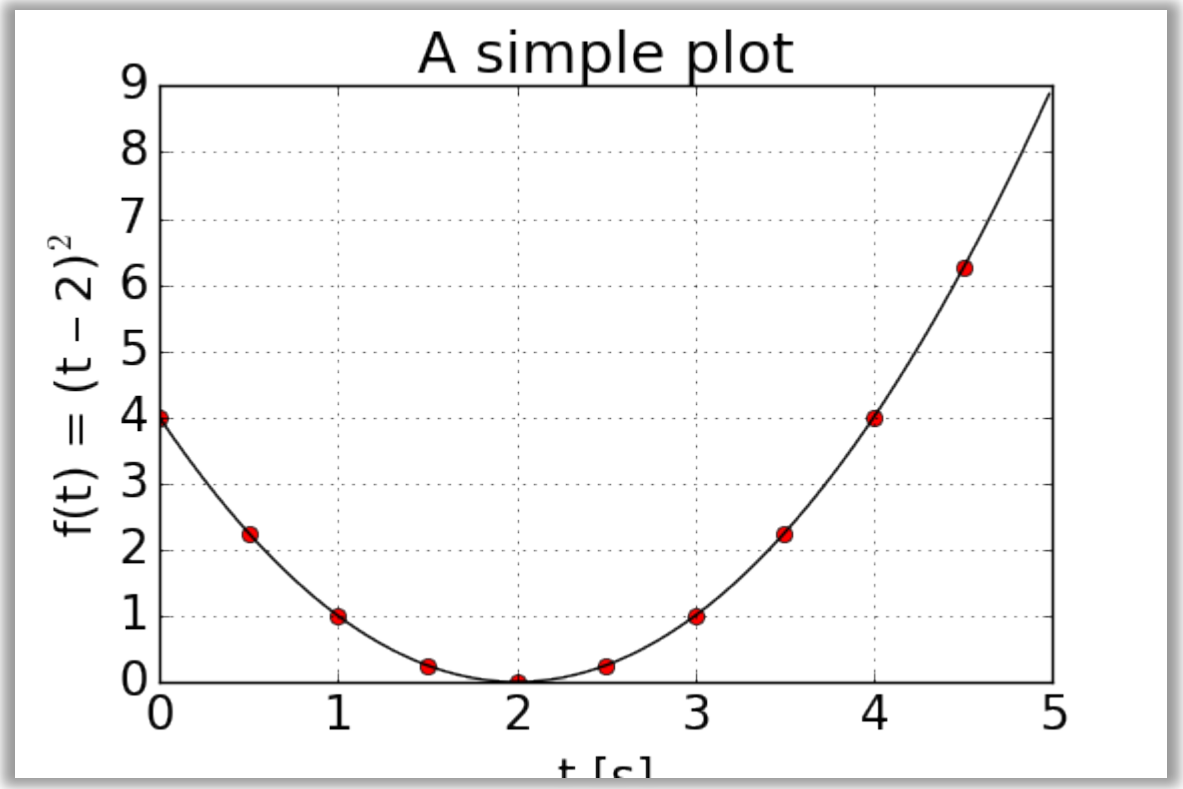


A simple plot Plot with style

```
>>> plt.grid()
```

A simple plot Plot with style

```
>>> plt.grid()
```

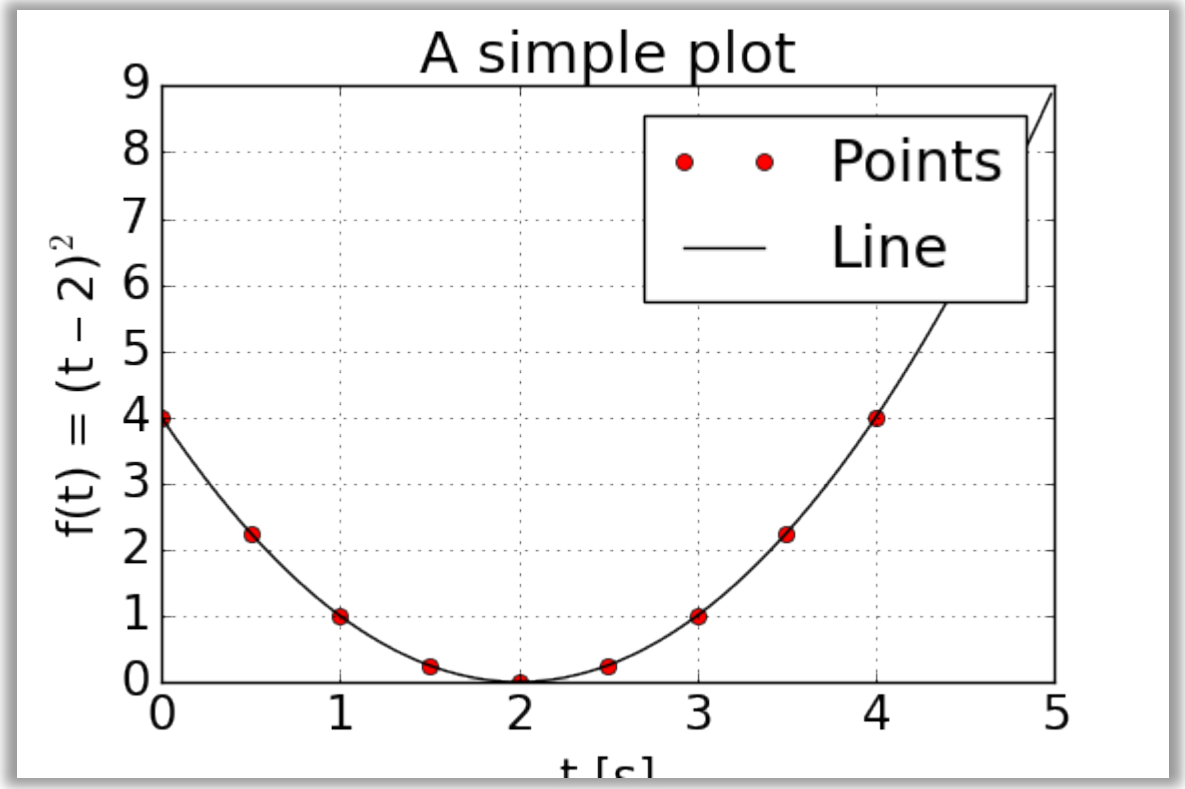


A simple plot Plot with style

```
>>> plt.plot(t1, f(t1), 'ro', label="Points")
>>> plt.plot(t2, f(t2), 'k', label="Line")
>>> plt.legend()
```

A simple plot Plot with style

```
>>> plt.plot(t1, f(t1), 'ro', label="Points")
>>> plt.plot(t2, f(t2), 'k', label="Line")
>>> plt.legend()
```



A simple plot Plot with style

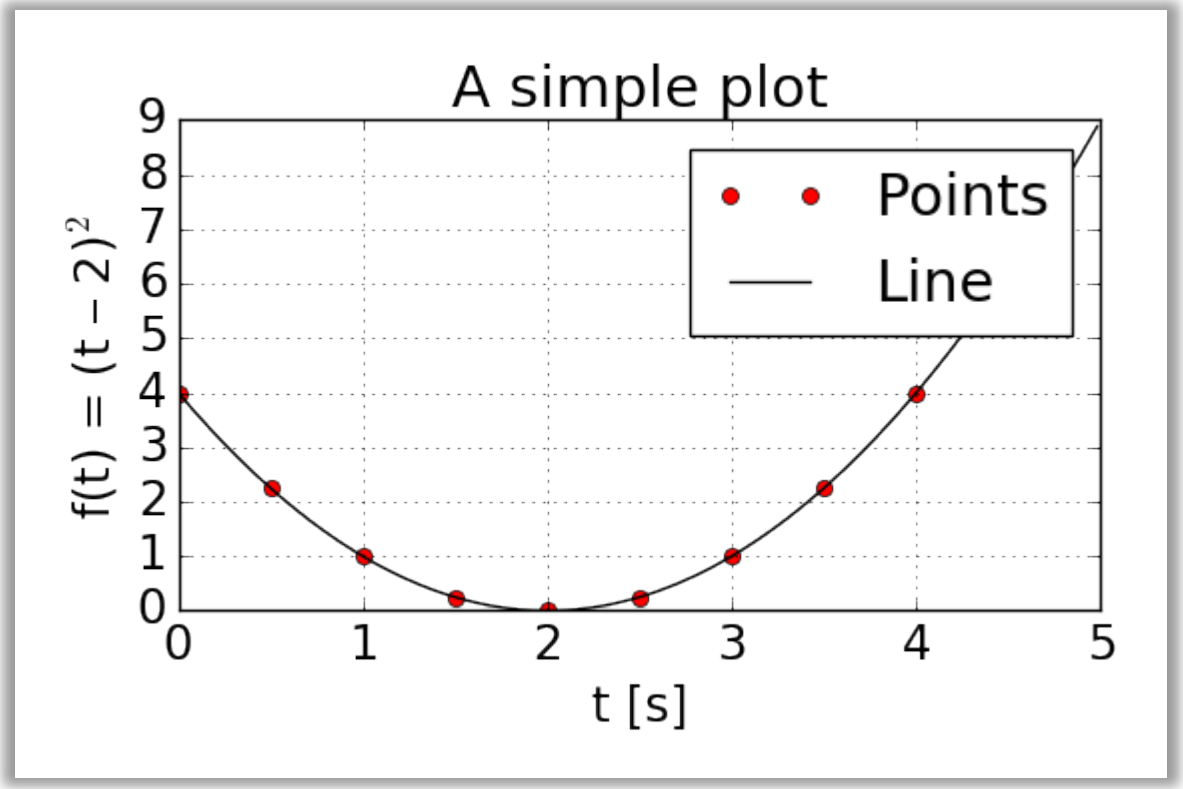
```
>>> plt.plot(t1, f(t1), 'ro', label="Points")
>>> plt.plot(t2, f(t2), 'k', label="Line")
>>> plt.legend()
```

A simple plot Plot with style

```
>>> plt.tight_layout()
```

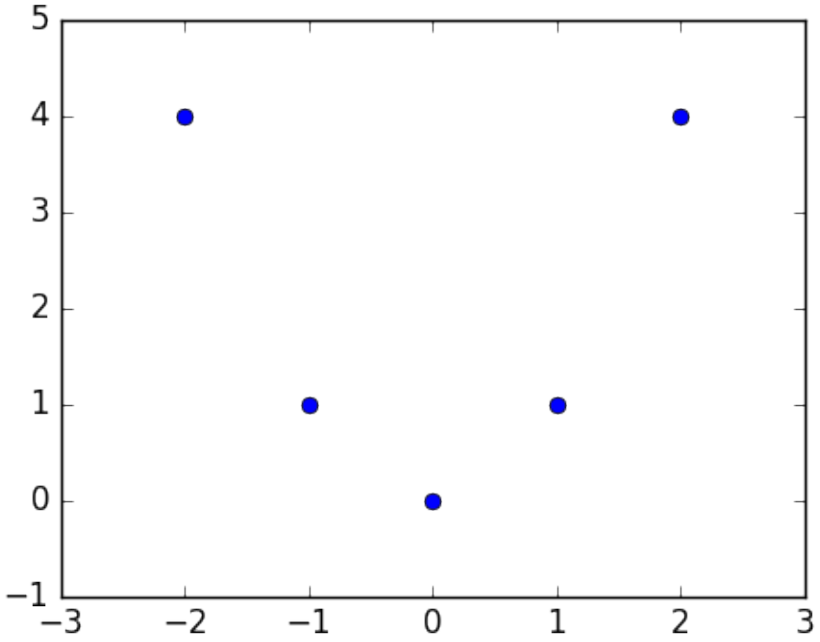
A simple plot Plot with style

```
>>> plt.tight_layout()
```



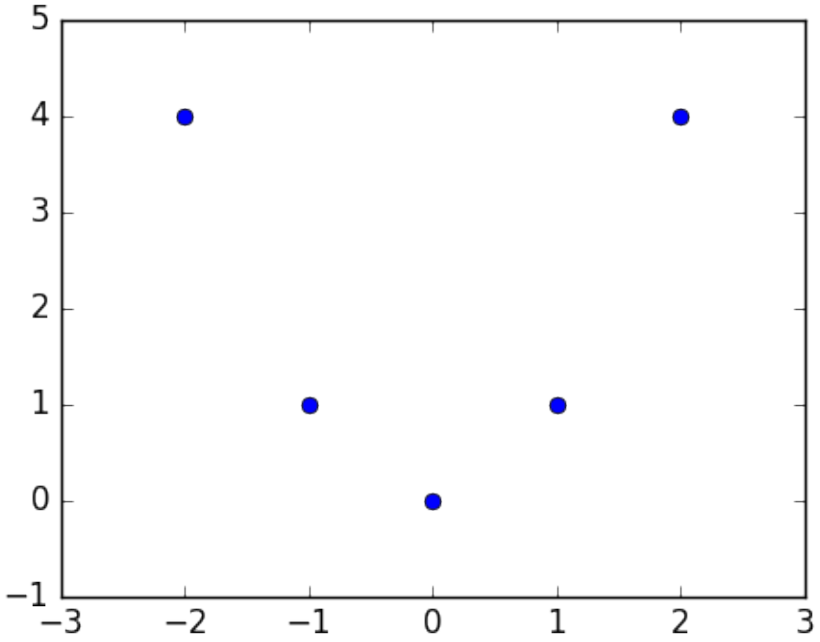
A good practice

```
x = np.arange(5) - 2
y = x ** 2

fig = plt.figure()
ax = fig.add_subplot(111)

ax.plot(x, y, 'o')
ax.set_xlim(-3, 3)
ax.set_xlim(-1, 5)

plt.show()
```



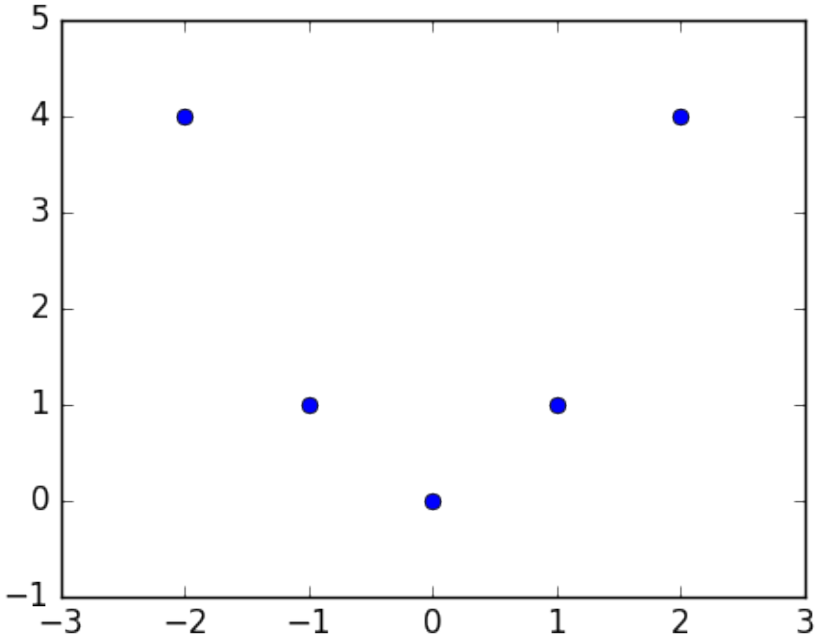
Some advanced MatPlotLib

```
x = np.arange(5) - 2
y = x ** 2

fig = plt.figure()
ax = fig.add_subplot(111)

ax.plot(x, y, 'o')
ax.set_xlim(-3, 3)
ax.set_xlim(-1, 5)

plt.show()
```



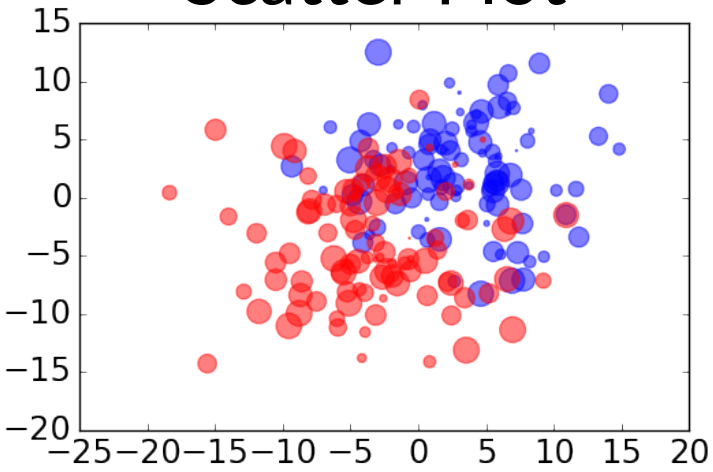
Some advanced MatPlotLib

```
x = np.arange(5) - 2
y = x ** 2

fig = plt.figure()
ax = fig.add_subplot(111)

ax.plot(x, y, 'o')
ax.set_xlim(-3, 3)
ax.set_xlim(-1, 5)

plt.show()
```

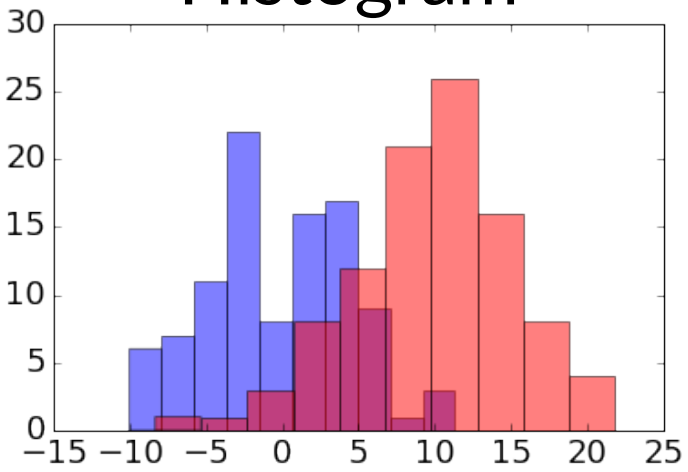


Types of plots

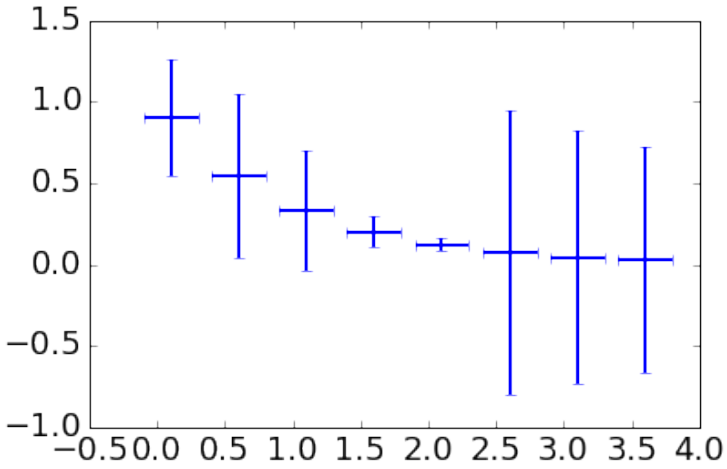
Scatter Plot



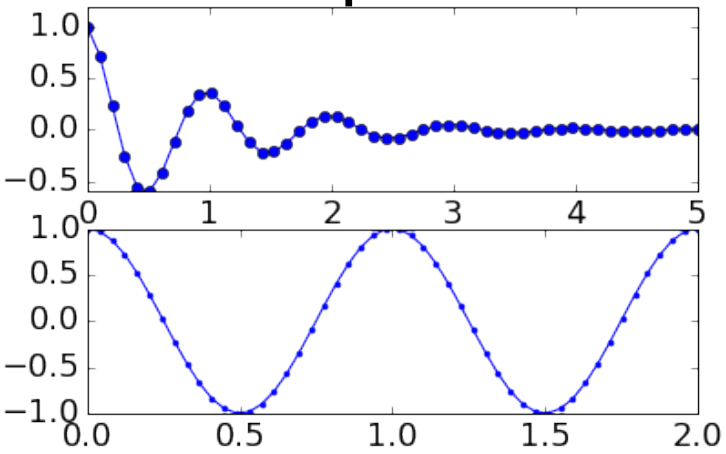
Histogram



Errorbar Plots

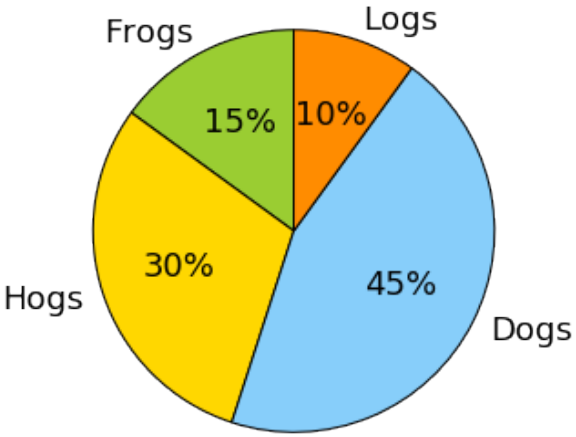


Multiple Axis

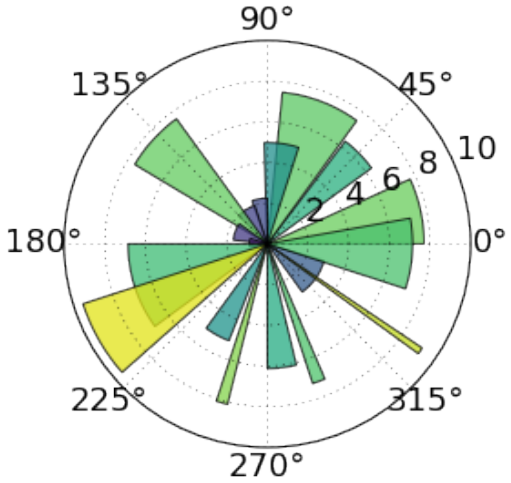


Types of plots

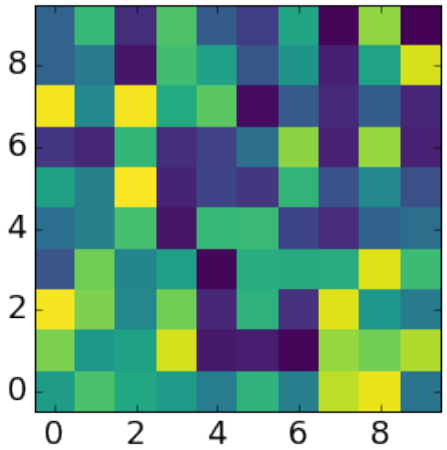
Pie Plots



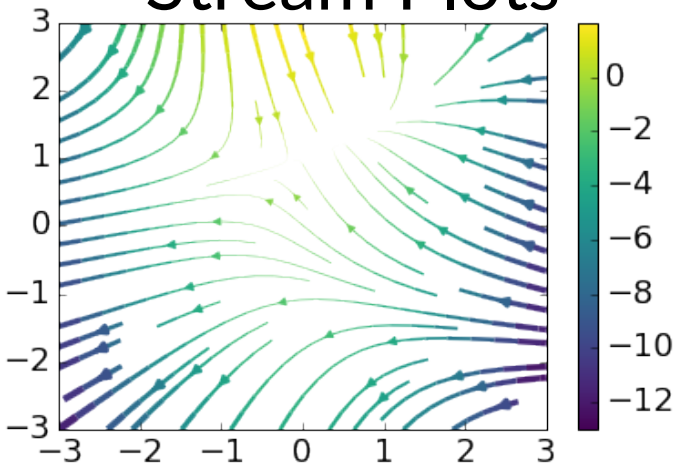
Polar Plots



Display Images

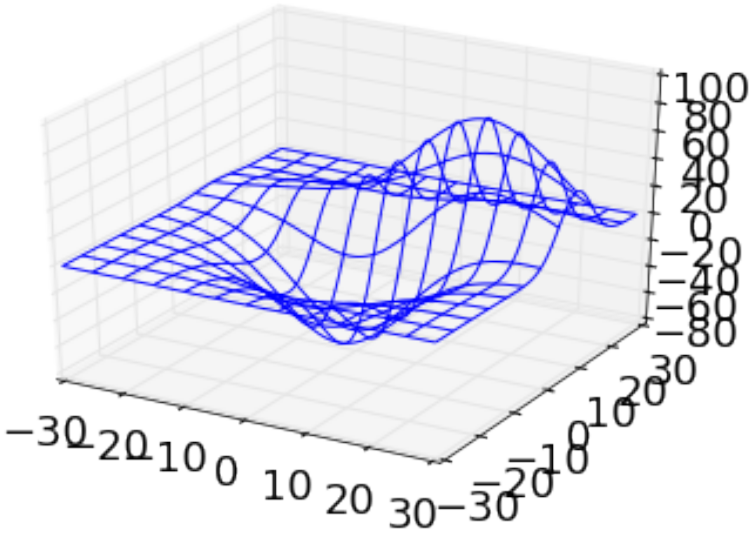


Stream Plots

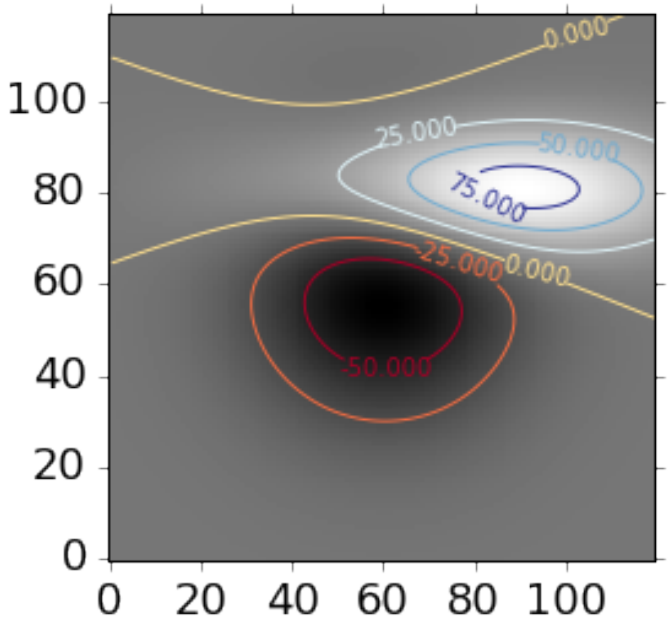


Types of plots

Plots 3D

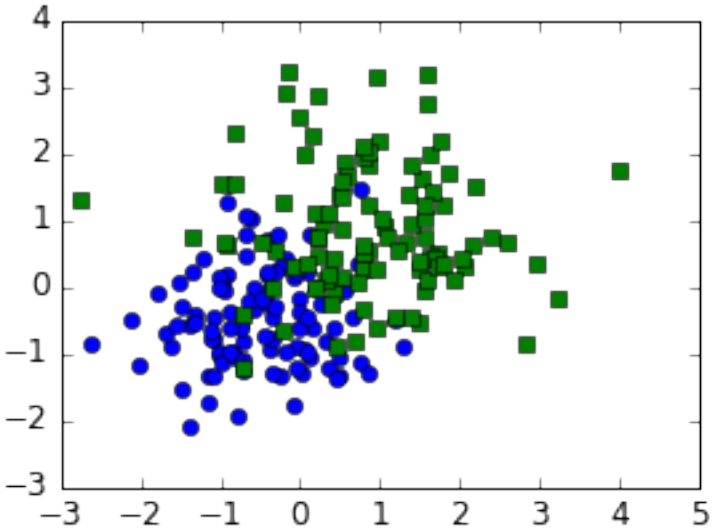


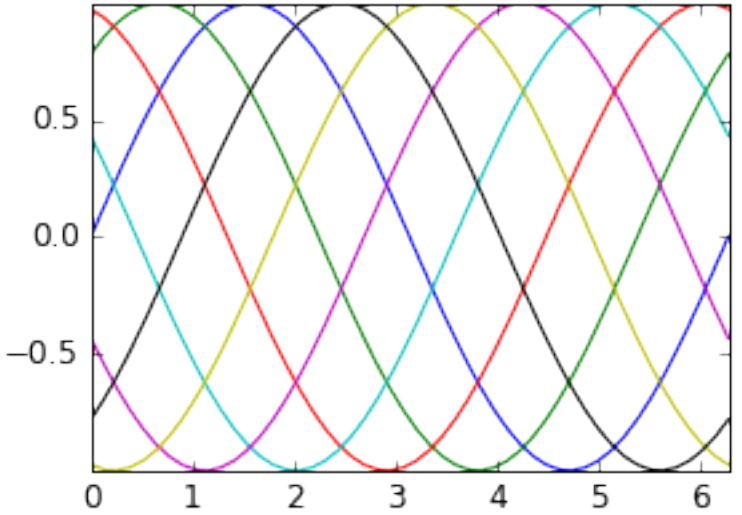
Contour Plots

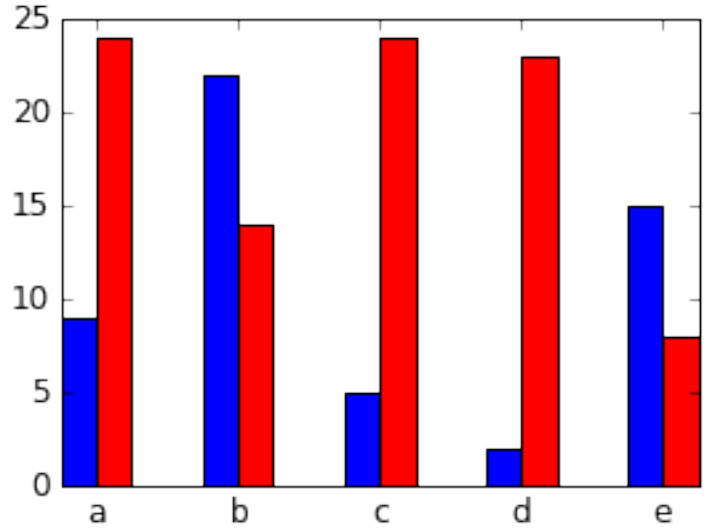


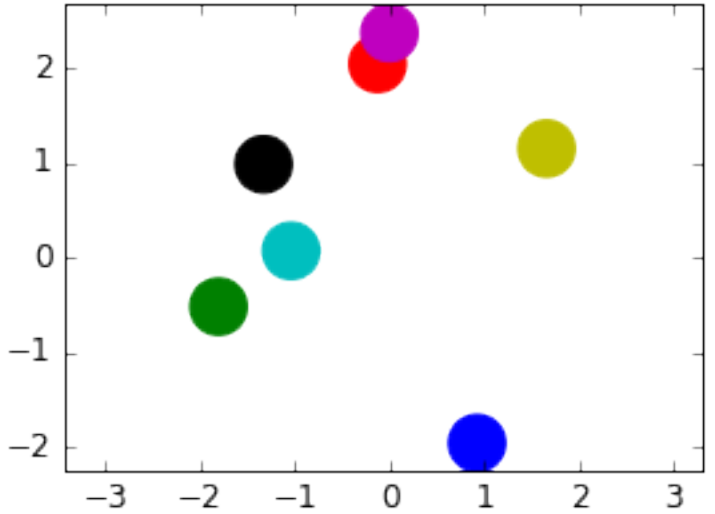
Different Styles

Default



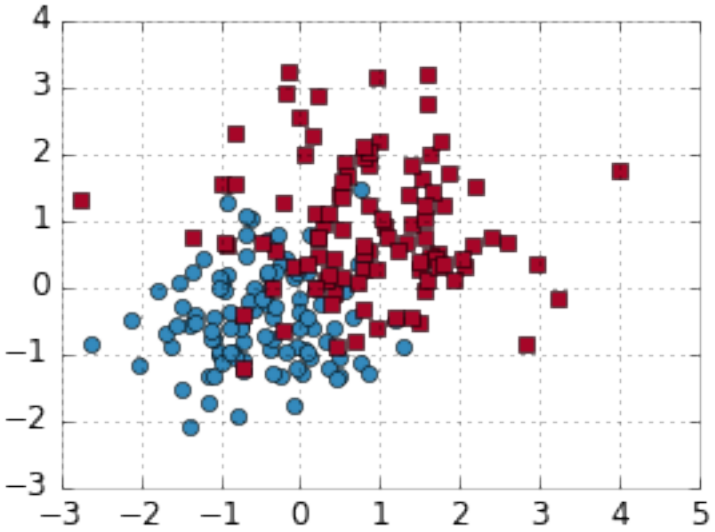


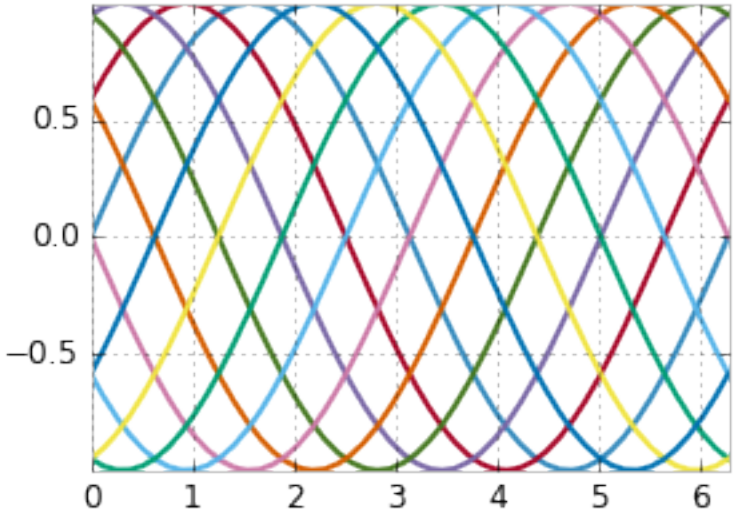


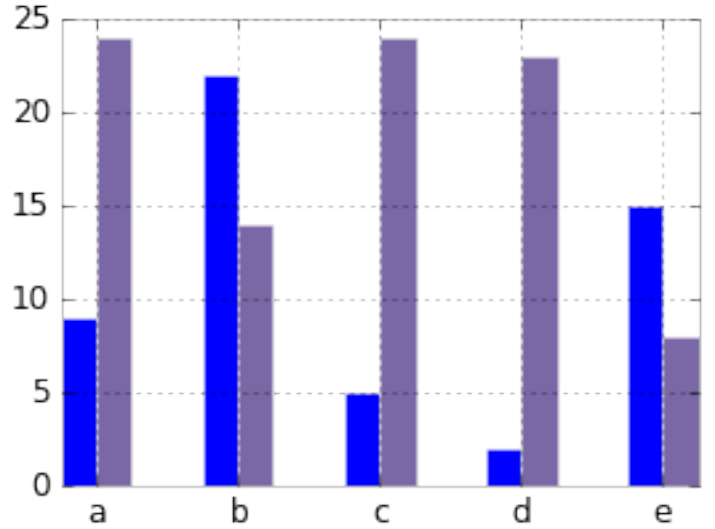


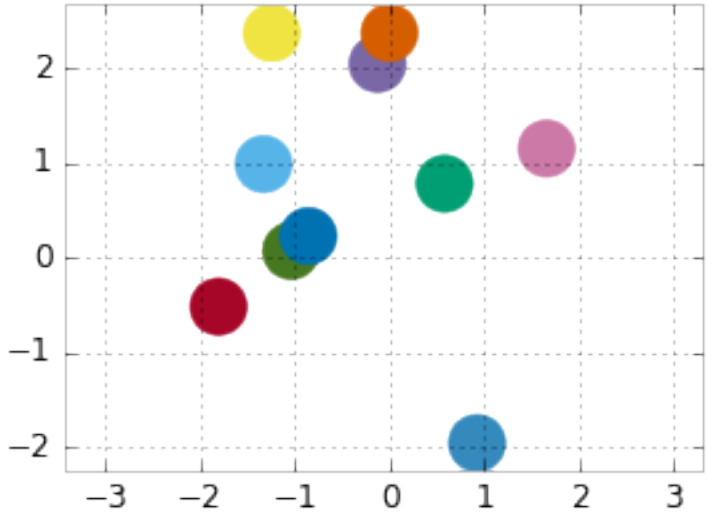
Different Styles

bmh



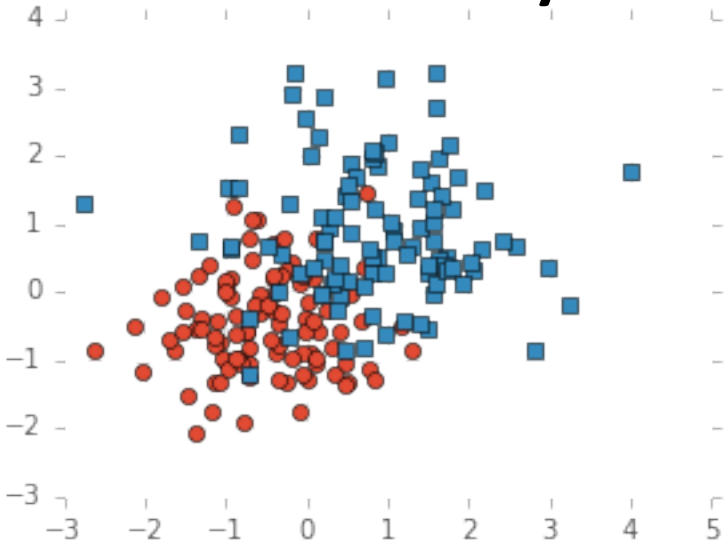


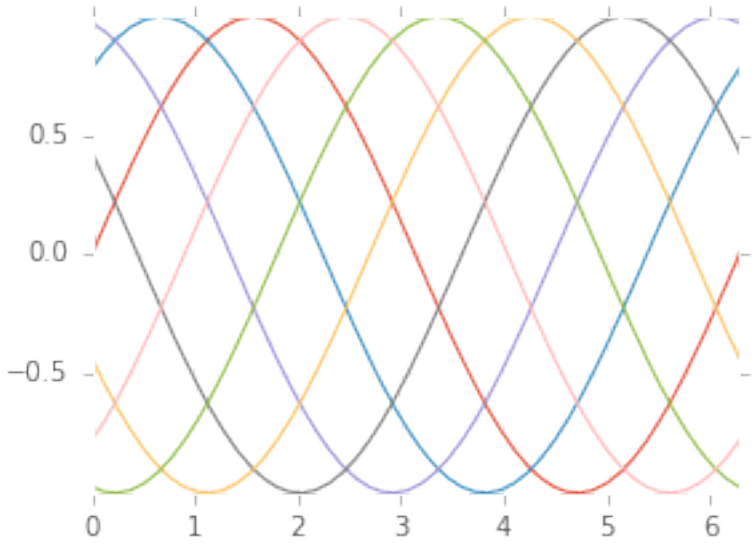


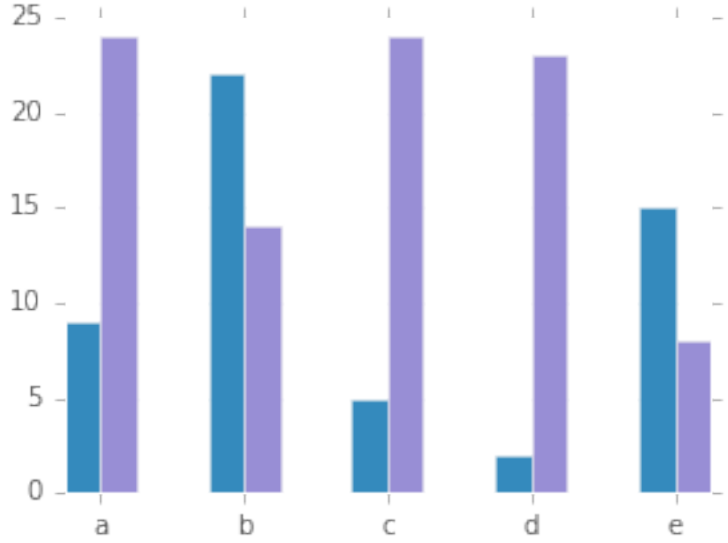


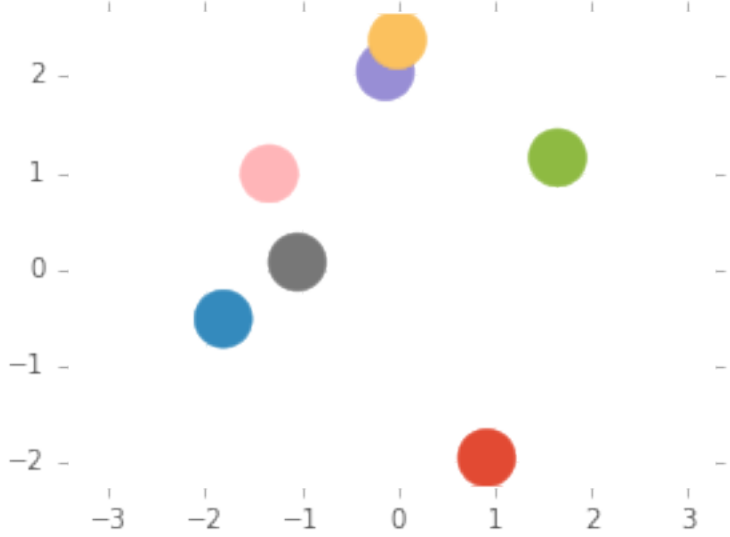
Different Styles

ggplot



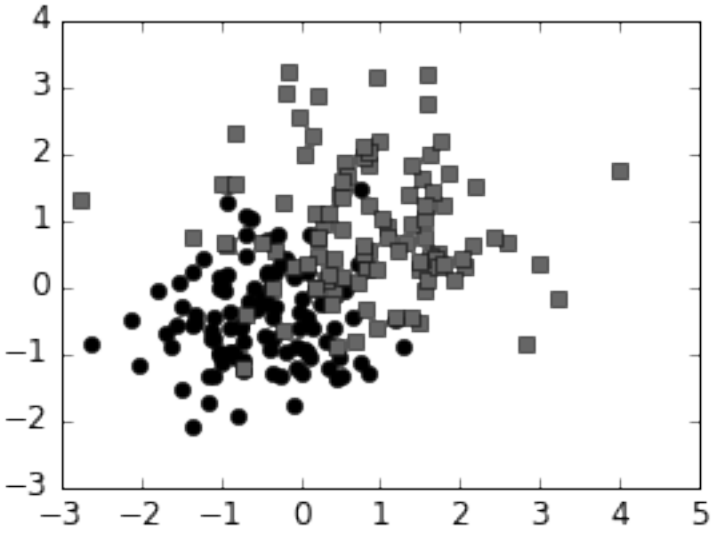


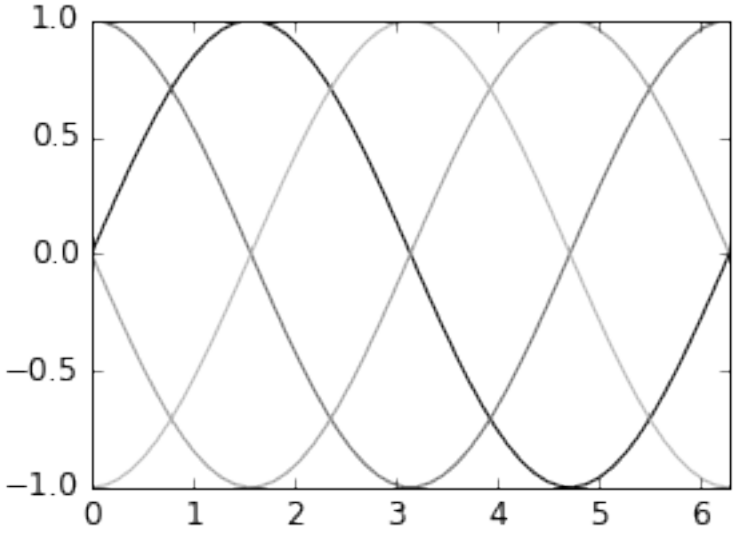


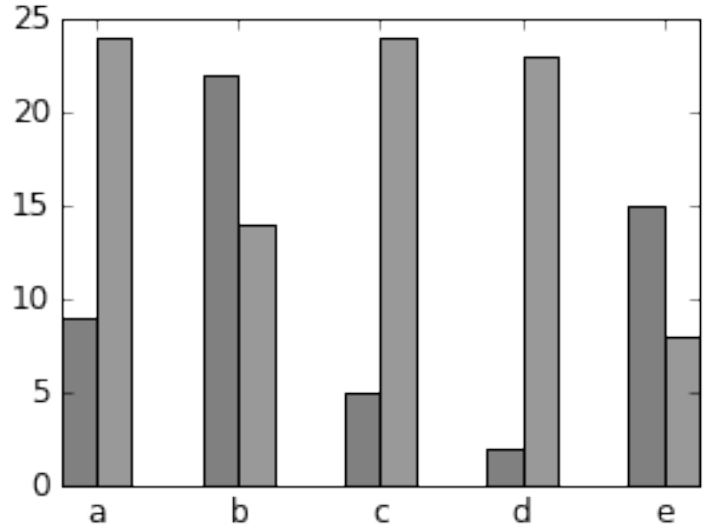


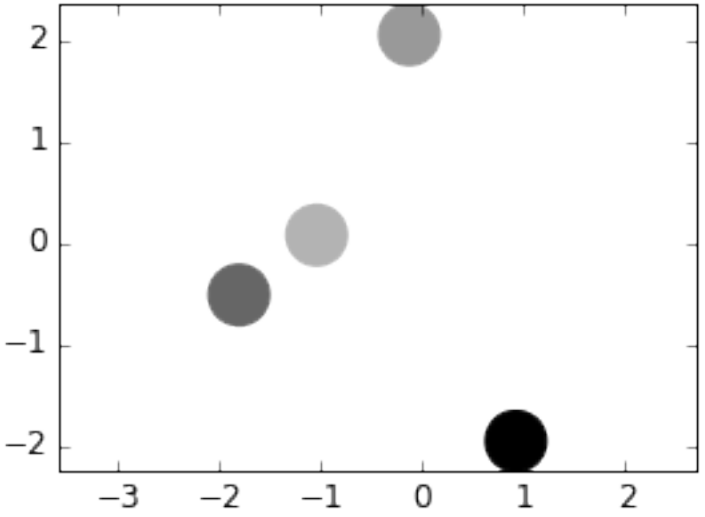
Different Styles

grayscale



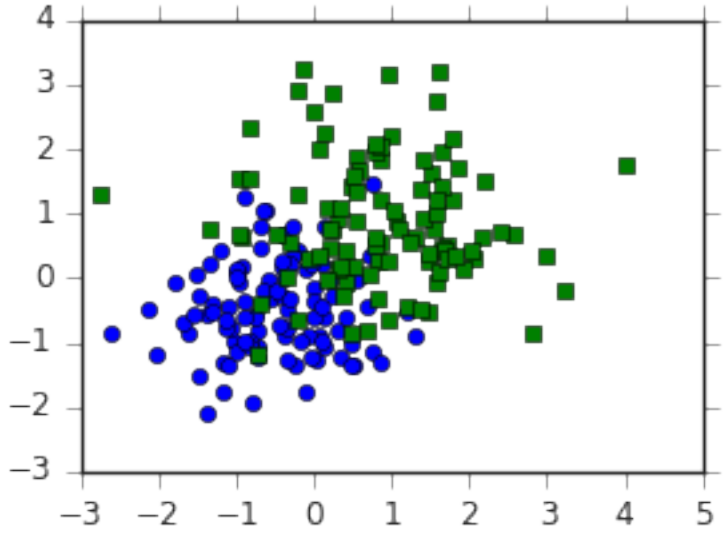


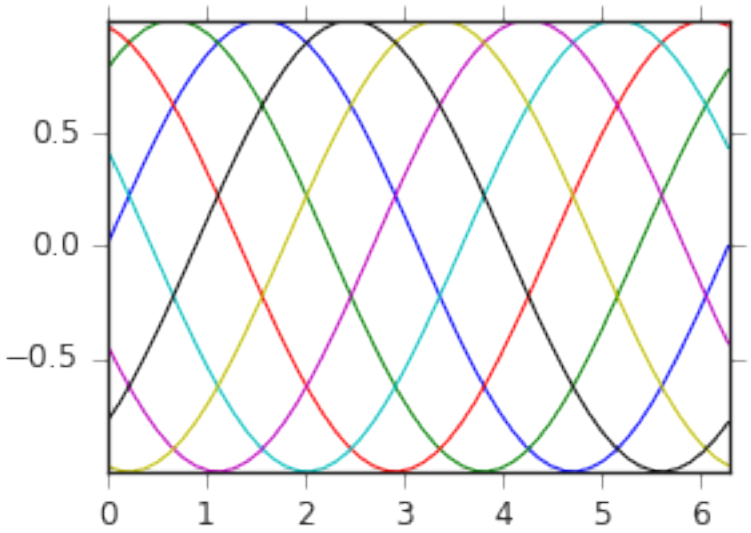


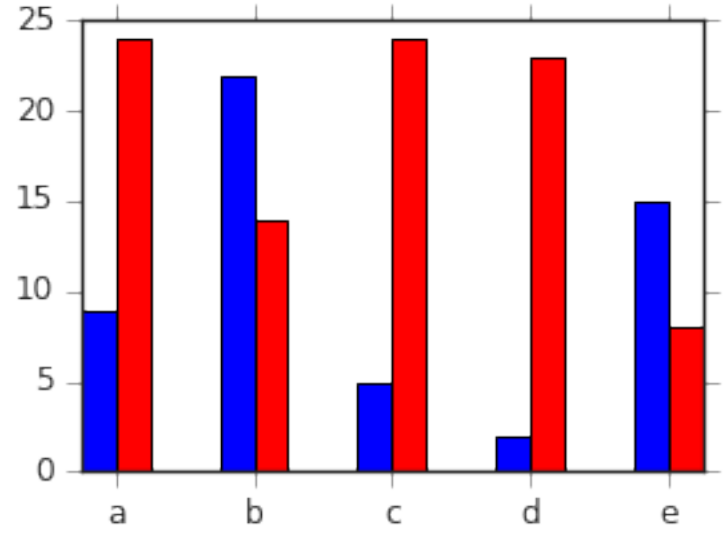


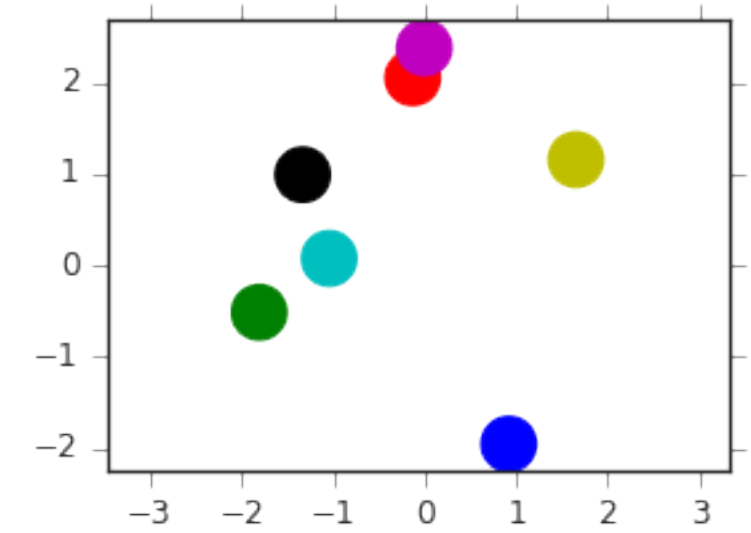
Different Styles

seaborn-ticks



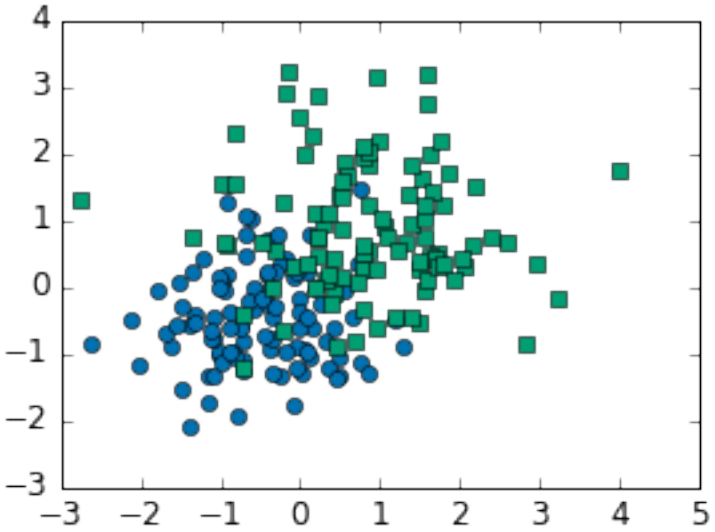


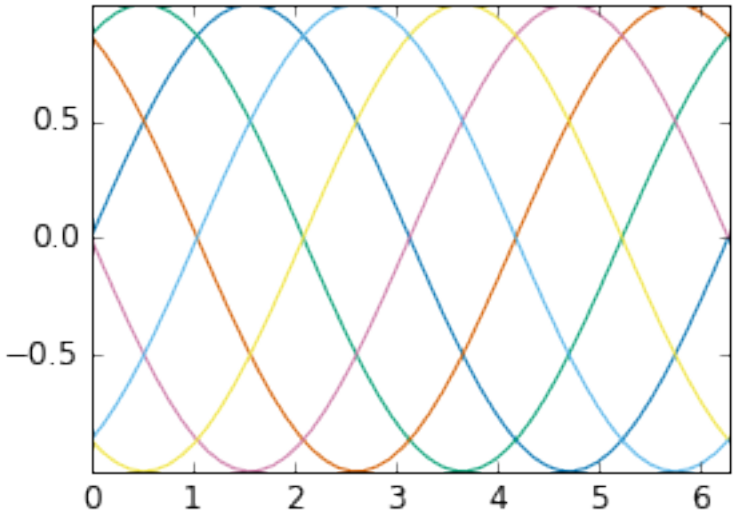


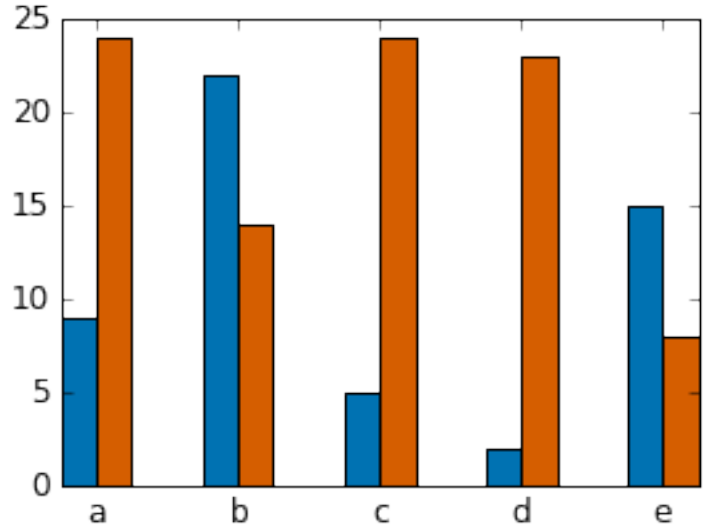


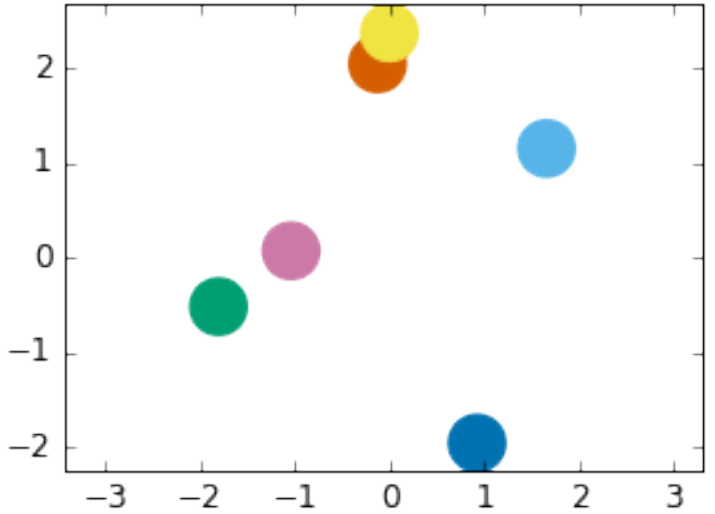
Different Styles

seaborn-colorblind



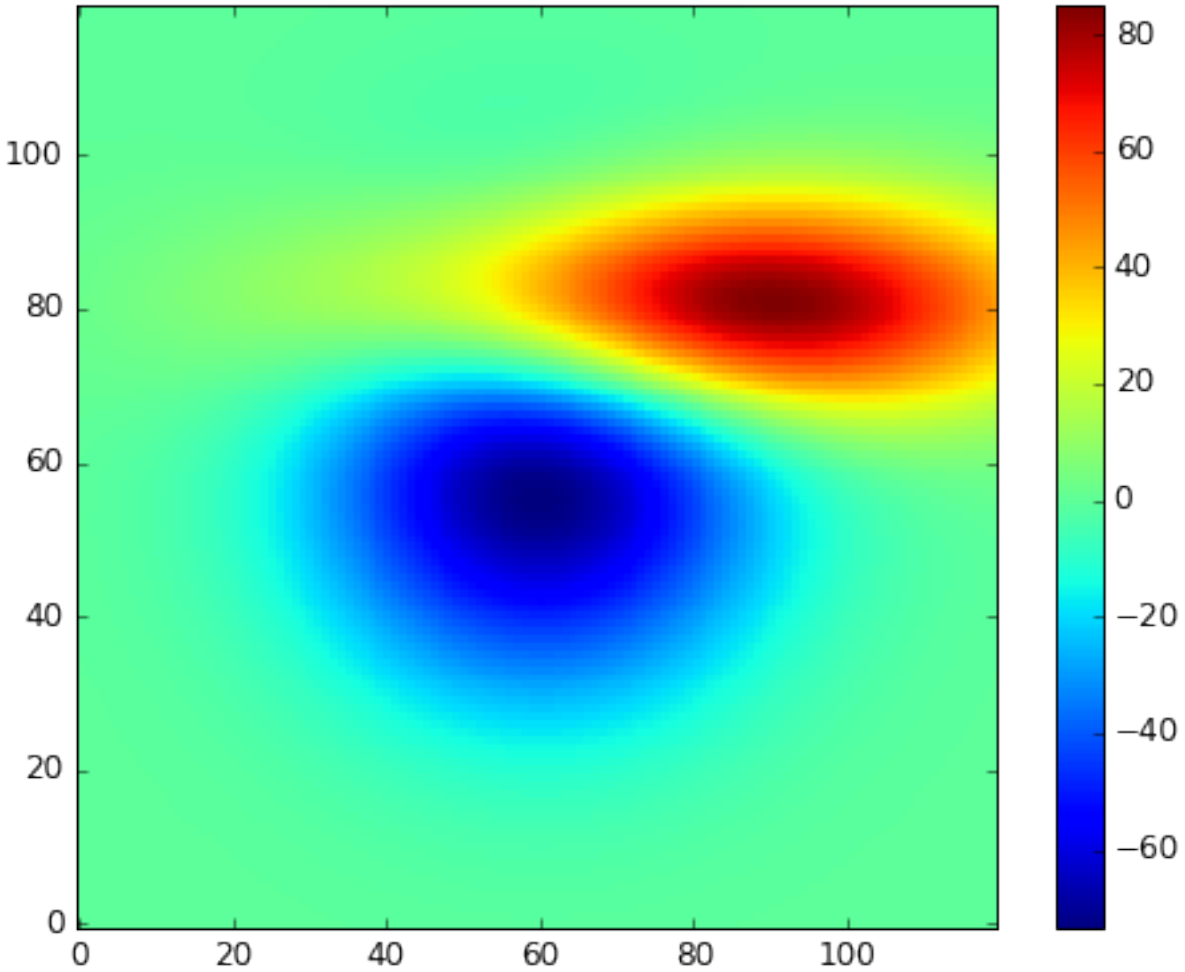






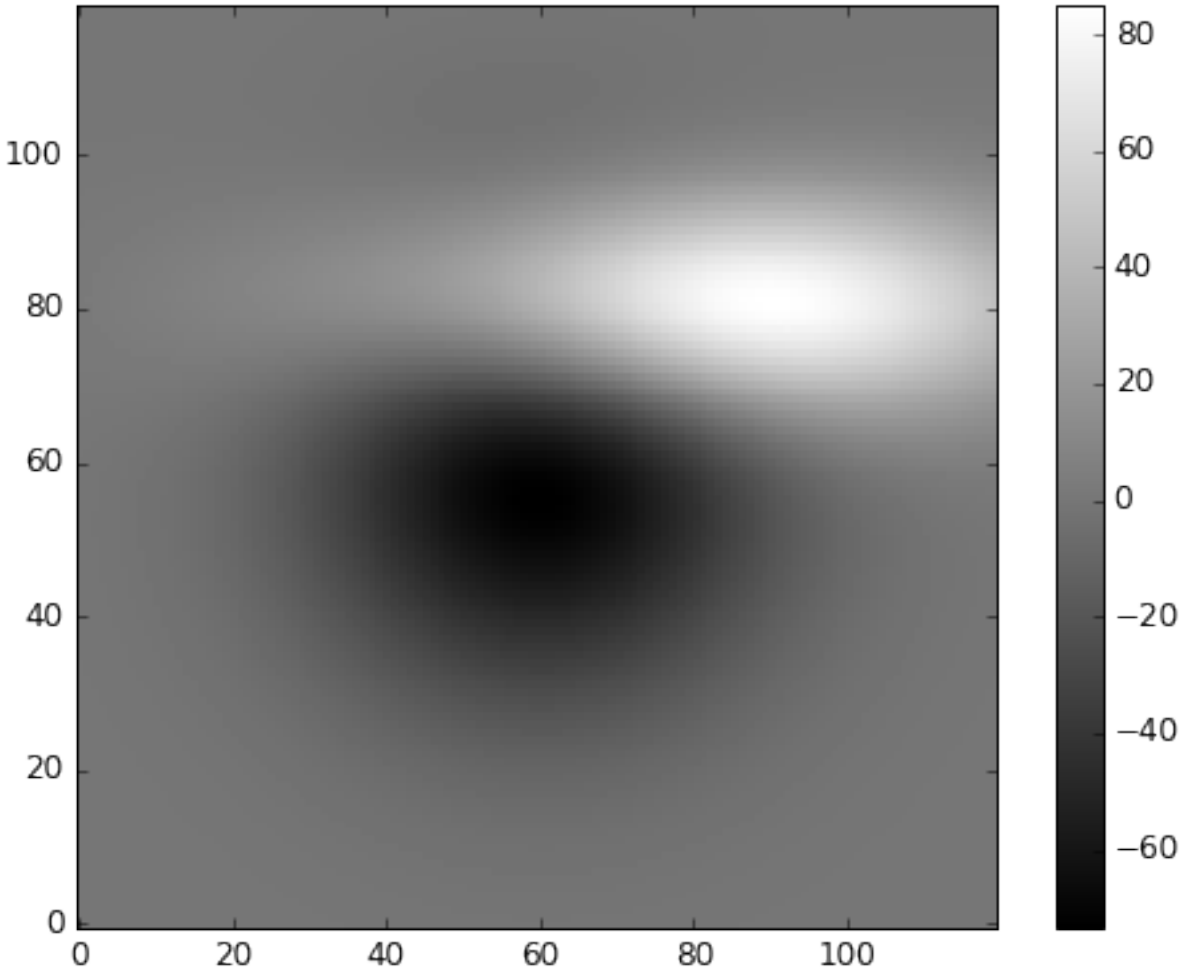
Colormaps

jet



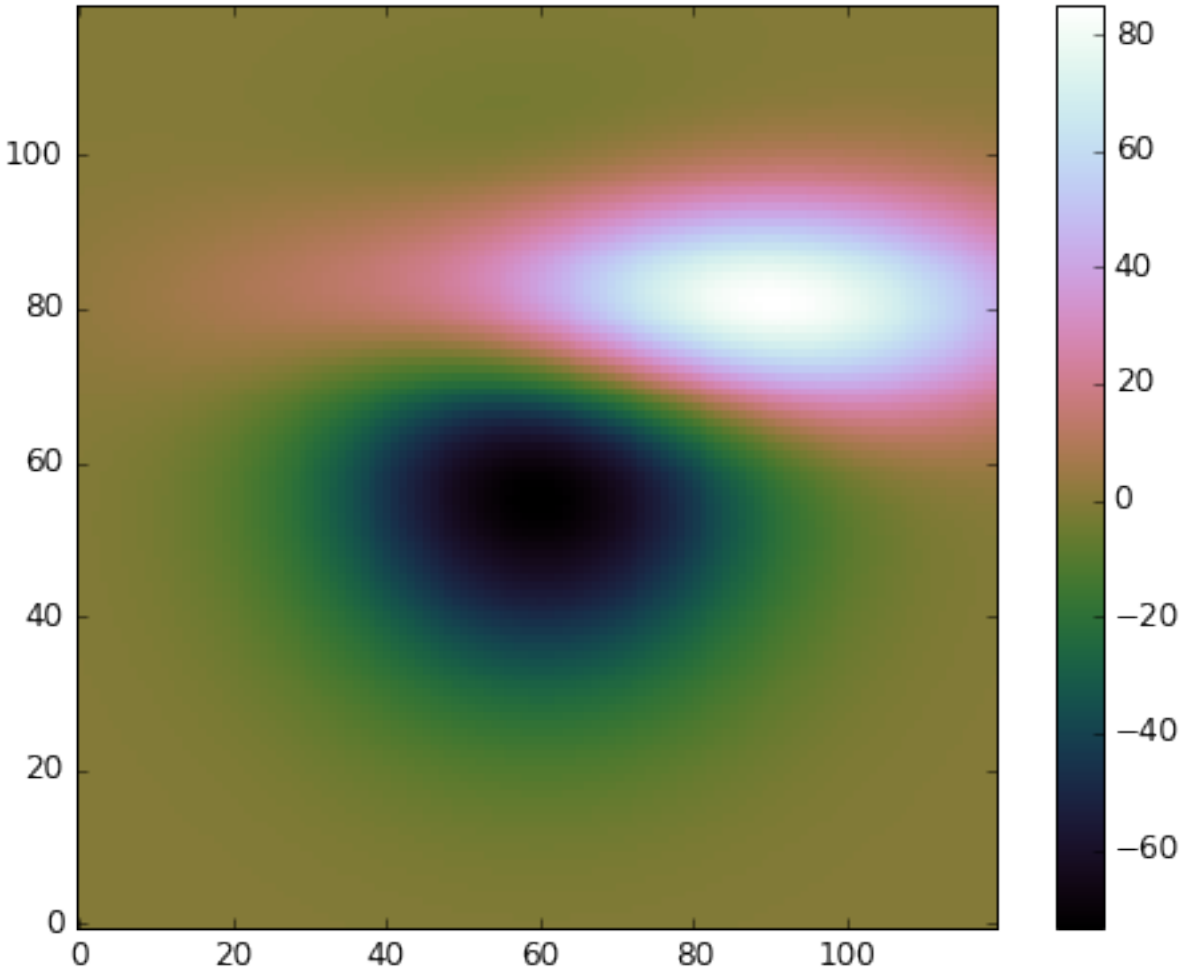
Colormaps

grayscale



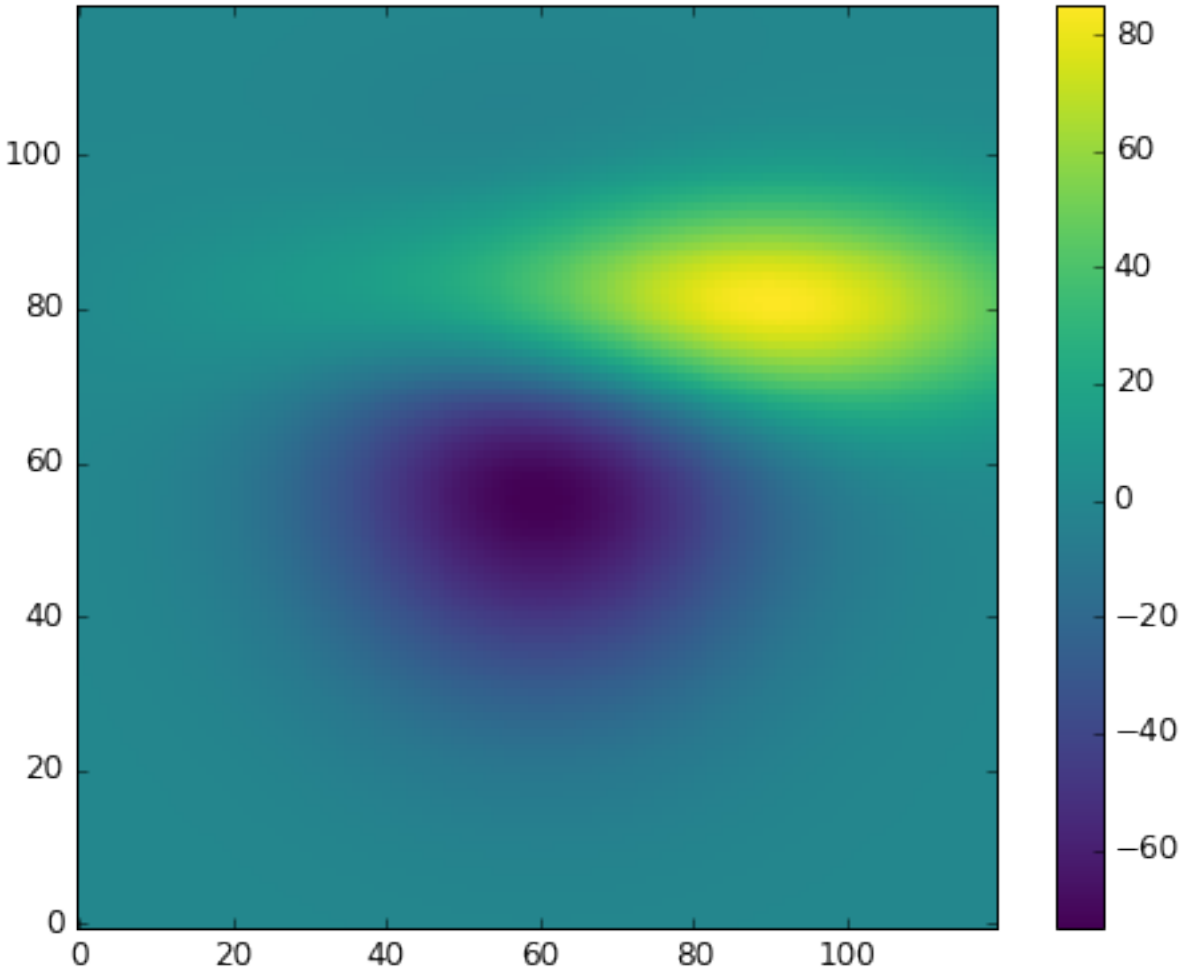
Colormaps

cubehelix



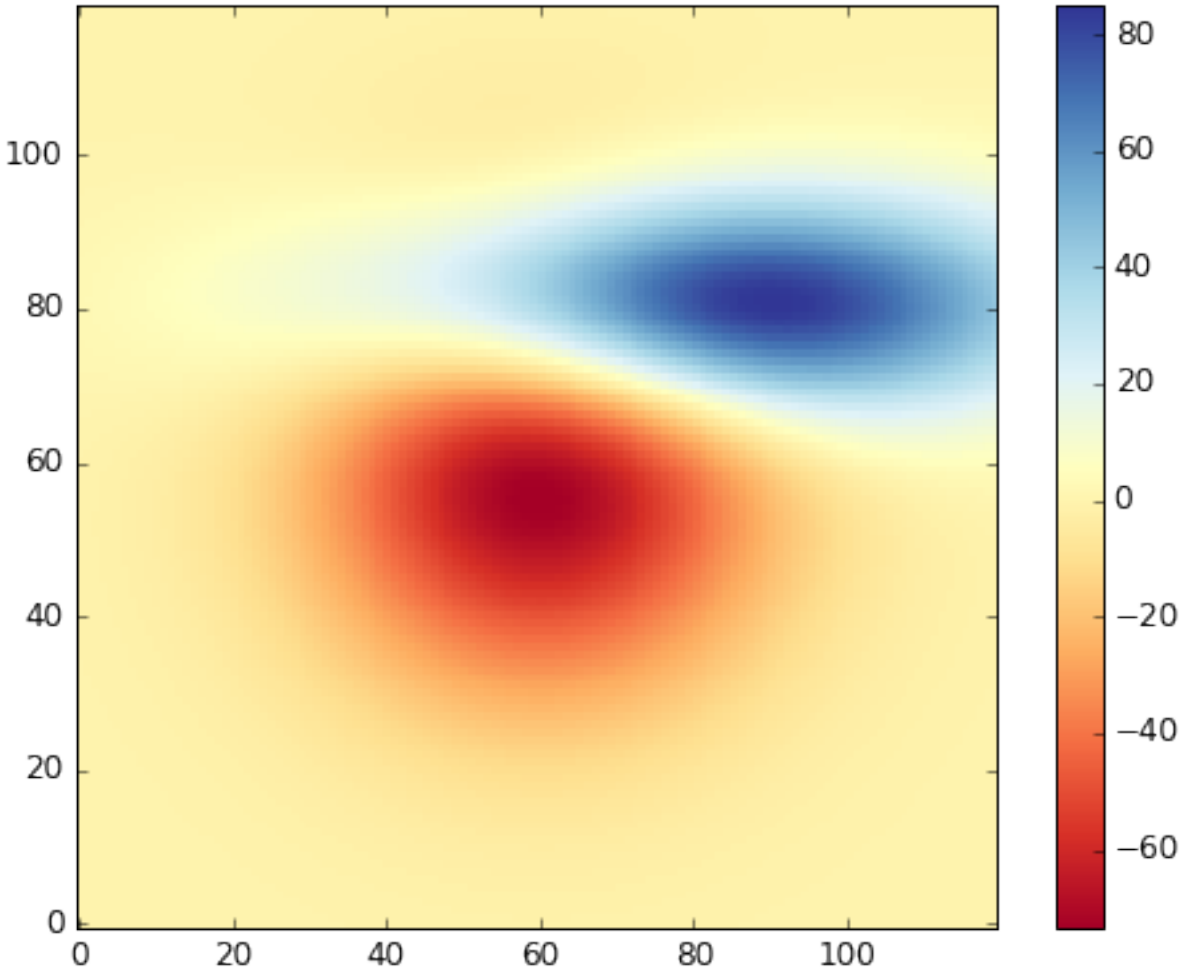
Colormaps

viridis



Colormaps

RdYlBu



Questions?