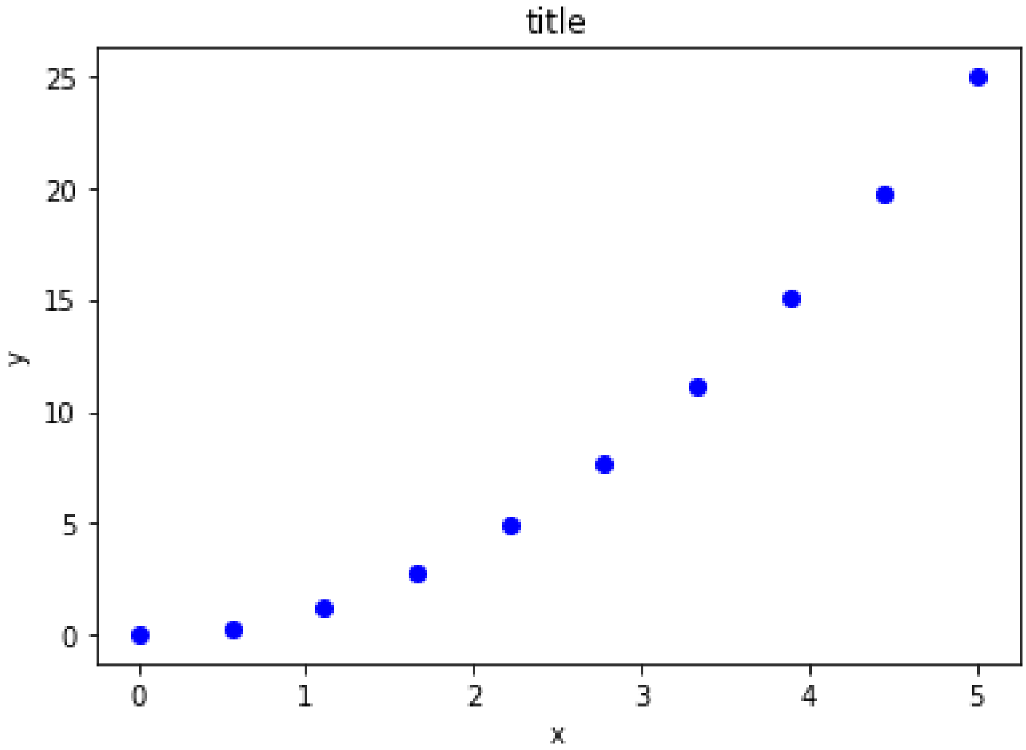
matplotlib2

December 4, 2018

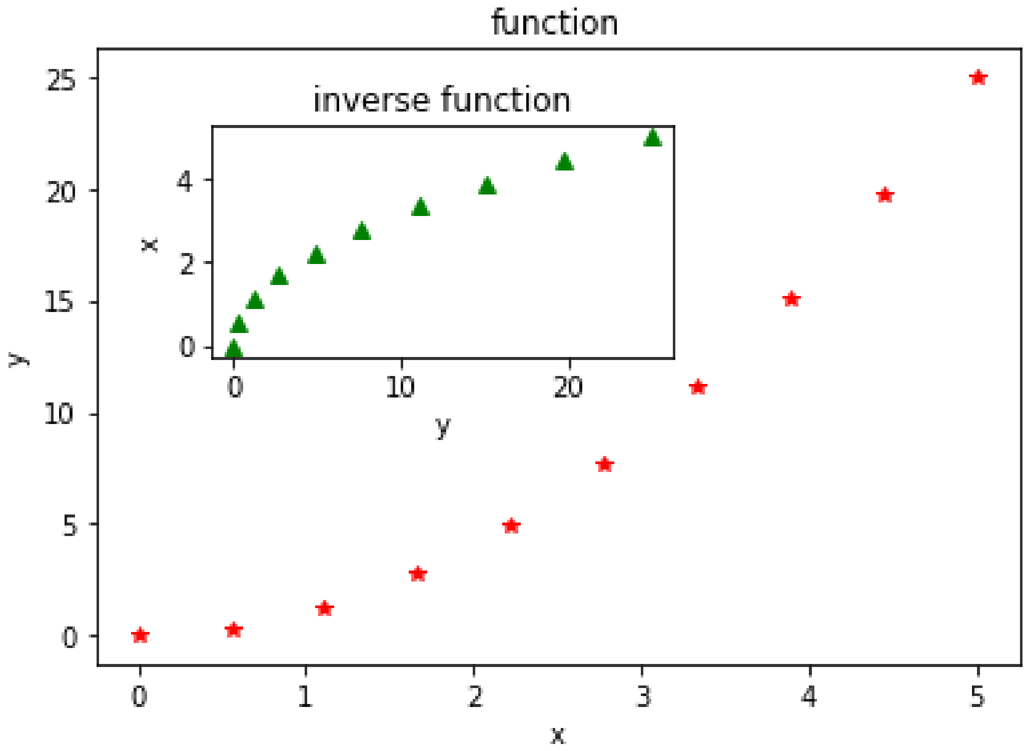
```
In [2]: import numpy as np
        import matplotlib
        import matplotlib.pyplot as plt
```

```
In [8]: x = np.linspace(0, 5, 10)
        y = x ** 2
```

```
In [9]: fig = plt.figure()

        axes = fig.add_axes([0.1, 0.1, 0.8, 0.8]) # left, bottom, width, height (range 0 to 1)

        axes.plot(x, y, 'bo')

        axes.set_xlabel('x')
        axes.set_ylabel('y')
        axes.set_title('title');
```

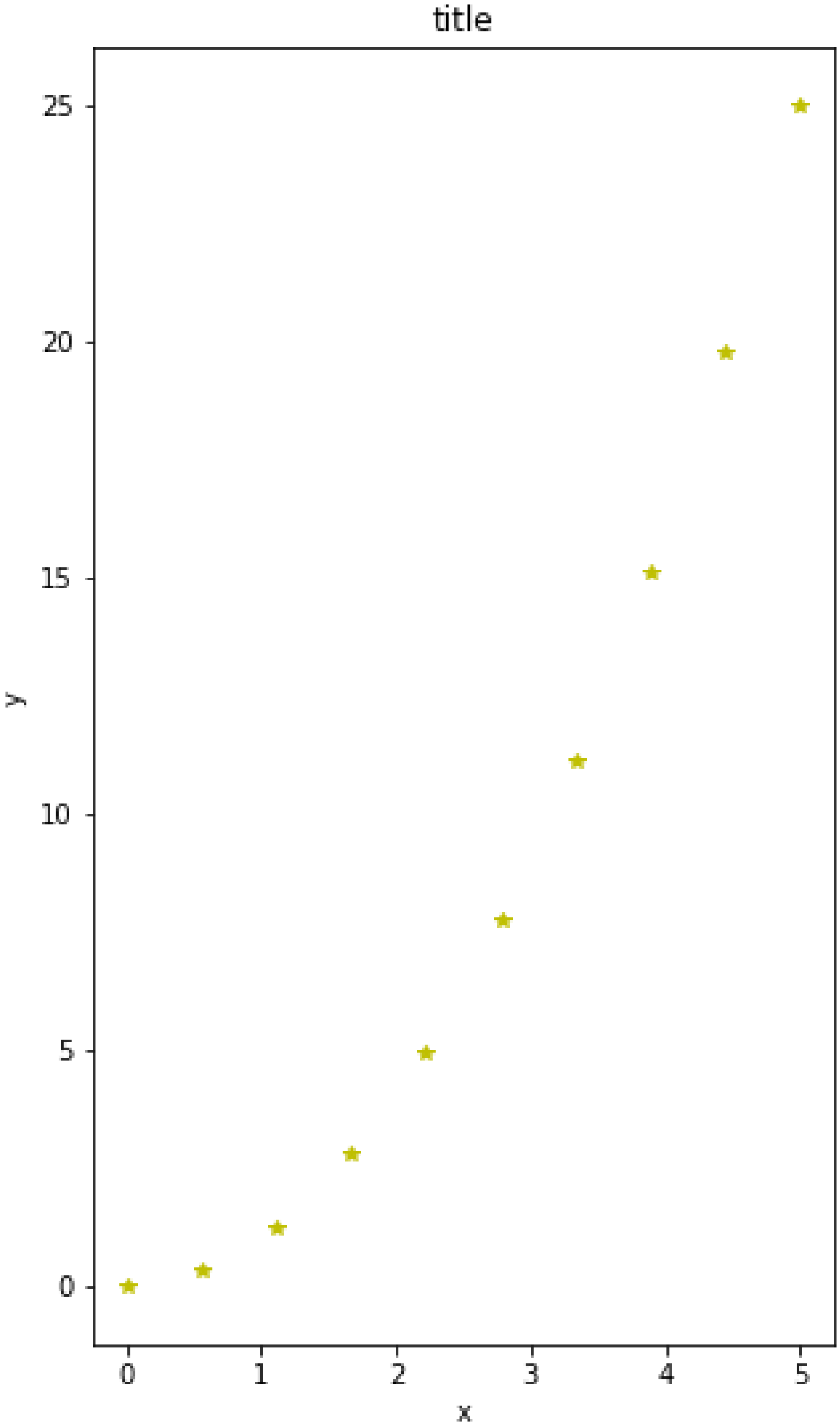


```
In [10]: fig = plt.figure()

         axes1 = fig.add_axes([0.1, 0.1, 0.8, 0.8]) # main axes
         axes2 = fig.add_axes([0.2, 0.5, 0.4, 0.3]) # inset axes

         # main figure
         axes1.plot(x, y, 'r*')
         axes1.set_xlabel('x')
         axes1.set_ylabel('y')
         axes1.set_title('function')

         # insert
         axes2.plot(y, x, 'g^')
         axes2.set_xlabel('y')
         axes2.set_ylabel('x')
         axes2.set_title('inverse function');
```



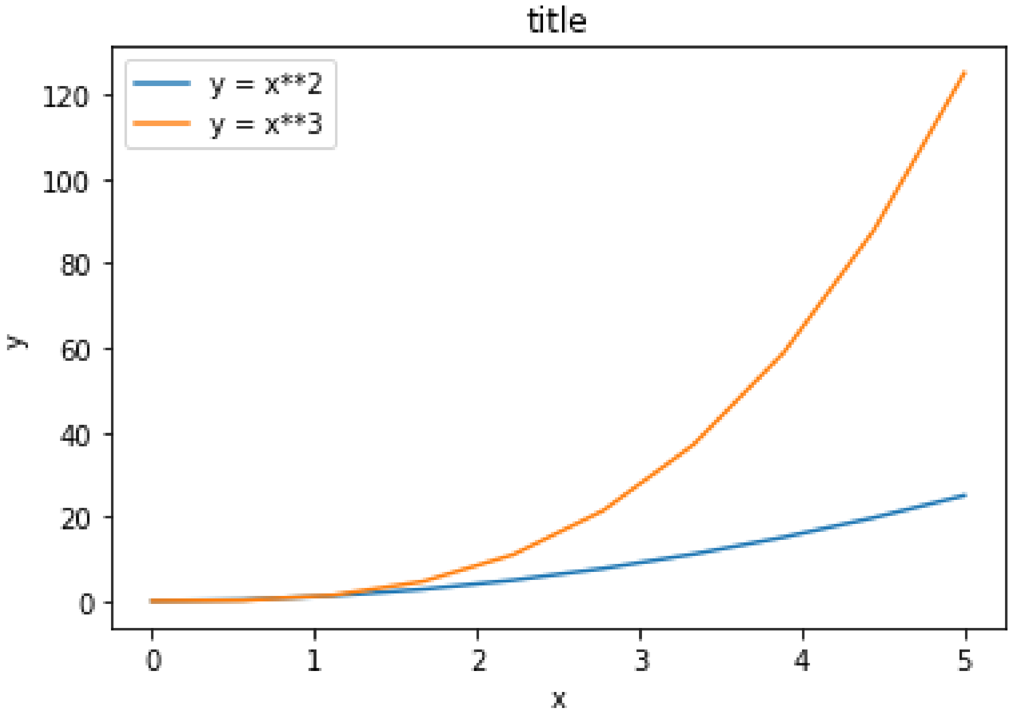
```
In [23]: fig, axes = plt.subplots(figsize=(5,9))

         axes.plot(x, y, 'y*')
```

```
#axes.axis('tight')
axes.set_xlabel('x')
axes.set_ylabel('y')
axes.set_title('title');
```

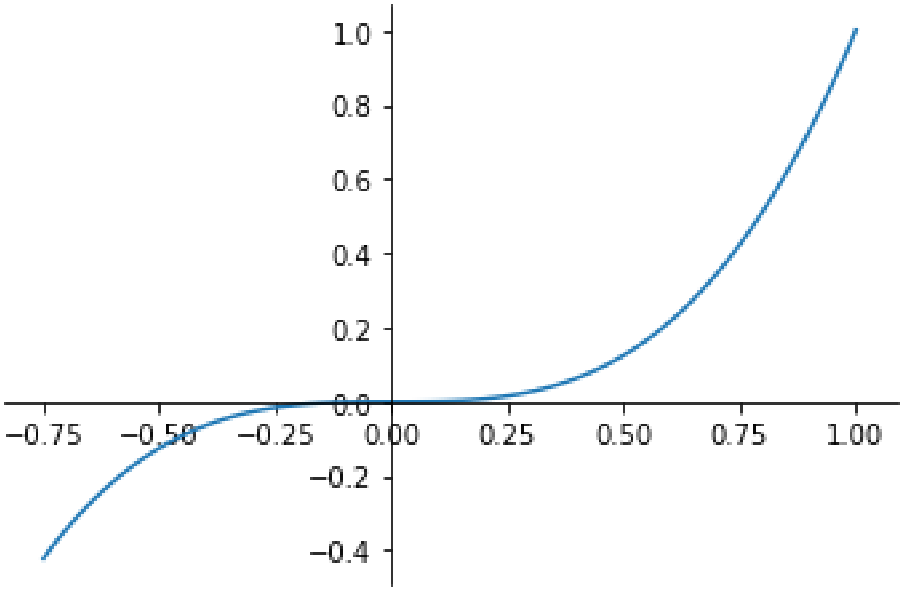


```
In [18]: fig.savefig("bigplot.pdf")
```

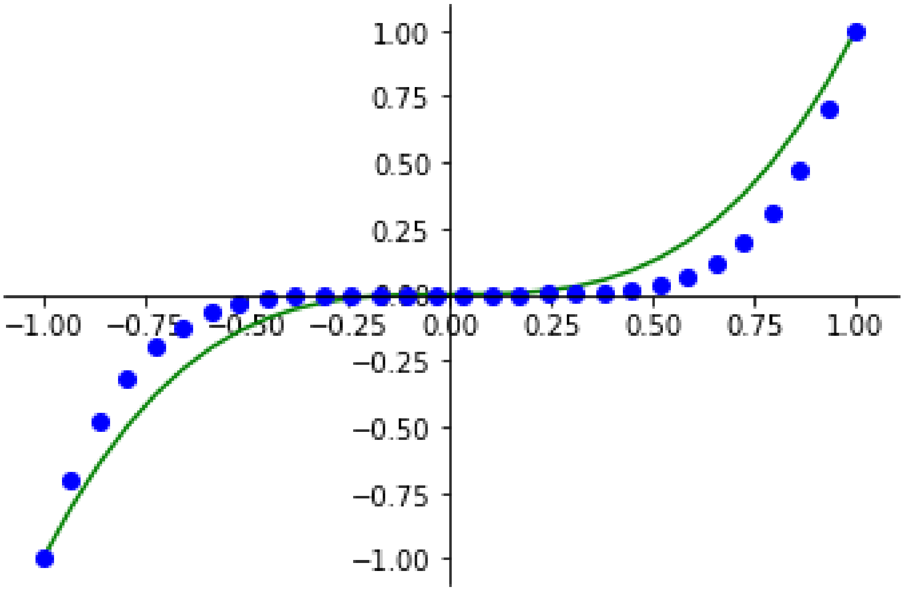
```
In [22]: fig, ax = plt.subplots()

         ax.plot(x, x**2, label="y = x**2")
         ax.plot(x, x**3, label="y = x**3")
         ax.legend(loc=0);
         ax.set_xlabel('x')
         ax.set_ylabel('y')
         ax.set_title('title');
```



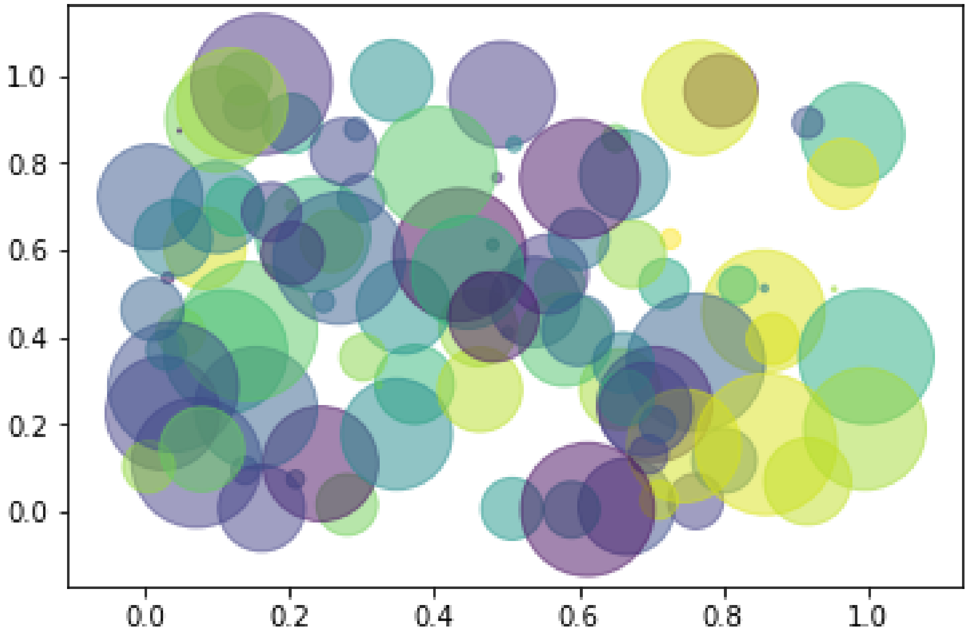
```
In [24]: fig, ax = plt.subplots()

         ax.spines['right'].set_color('none')
         ax.spines['top'].set_color('none')

         ax.xaxis.set_ticks_position('bottom')
         ax.spines['bottom'].set_position(('data',0)) # set position of x spine to x=0

         ax.yaxis.set_ticks_position('left')
         ax.spines['left'].set_position(('data',0))   # set position of y spine to y=0

         xx = np.linspace(-0.75, 1., 100)
         ax.plot(xx, xx**3);
```



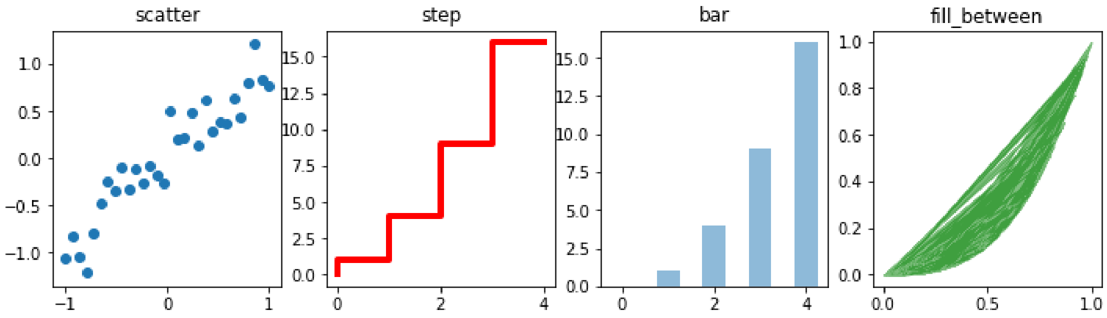
```
In [42]: fig, ax = plt.subplots()

         ax.spines['right'].set_color('none')
         ax.spines['top'].set_color('none')

         ax.xaxis.set_ticks_position('bottom')
         ax.spines['bottom'].set_position(('data',0)) # set position of x spine to x=0

         ax.yaxis.set_ticks_position('left')
         ax.spines['left'].set_position(('data',0))   # set position of y spine to y=0

         xx = np.linspace(-1., 1., 30)
         ax.plot(xx, xx**3, 'g-');
         ax.plot(xx, xx**5, 'bo');
```



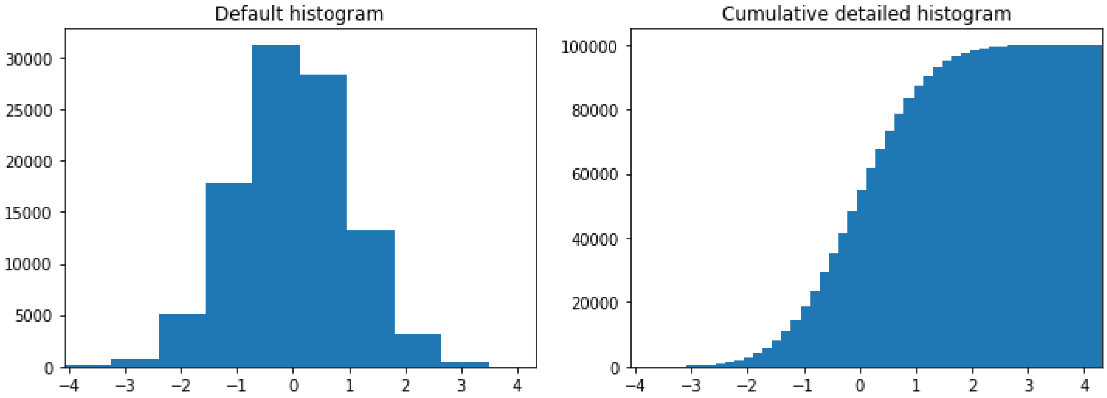
```
In [51]: #this one is fun
         N = 100
         x = np.random.rand(N)
         y = np.random.rand(N)
         colors = np.random.rand(N)
         area = np.pi * (30 * np.random.rand(N))**2  # 0 to 15 point radii

         plt.scatter(x, y, s=area, c=colors, alpha=0.5)
         plt.show()
```



```
In [55]: m = np.array([0, 1, 2, 3, 4])

In [66]: fig, axes = plt.subplots(1, 4, figsize=(12,3))

         axes[0].scatter(xx, xx + 0.25*np.random.randn(len(xx)))
         axes[0].set_title("scatter")

         axes[1].step(m, m**2, 'r', lw=4)
         axes[1].set_title("step")

         axes[2].bar(m, m**2, align="center", width=0.5, alpha=0.5)
         axes[2].set_title("bar")

         axes[3].fill_between(x, x**2, x**3, color="green", alpha=0.5);
         axes[3].set_title("fill_between");
```



```
In [69]: # A histogram
         n = np.random.randn(100000)
         fig, axes = plt.subplots(1, 2, figsize=(12,4))

         axes[0].hist(n)
         axes[0].set_title("Default histogram")
         axes[0].set_xlim((min(n), max(n)))

         axes[1].hist(n, cumulative=True, bins=50)
         axes[1].set_title("Cumulative detailed histogram")
         axes[1].set_xlim((min(n), max(n)));
```


Default histogram

Cumulative detailed histogram