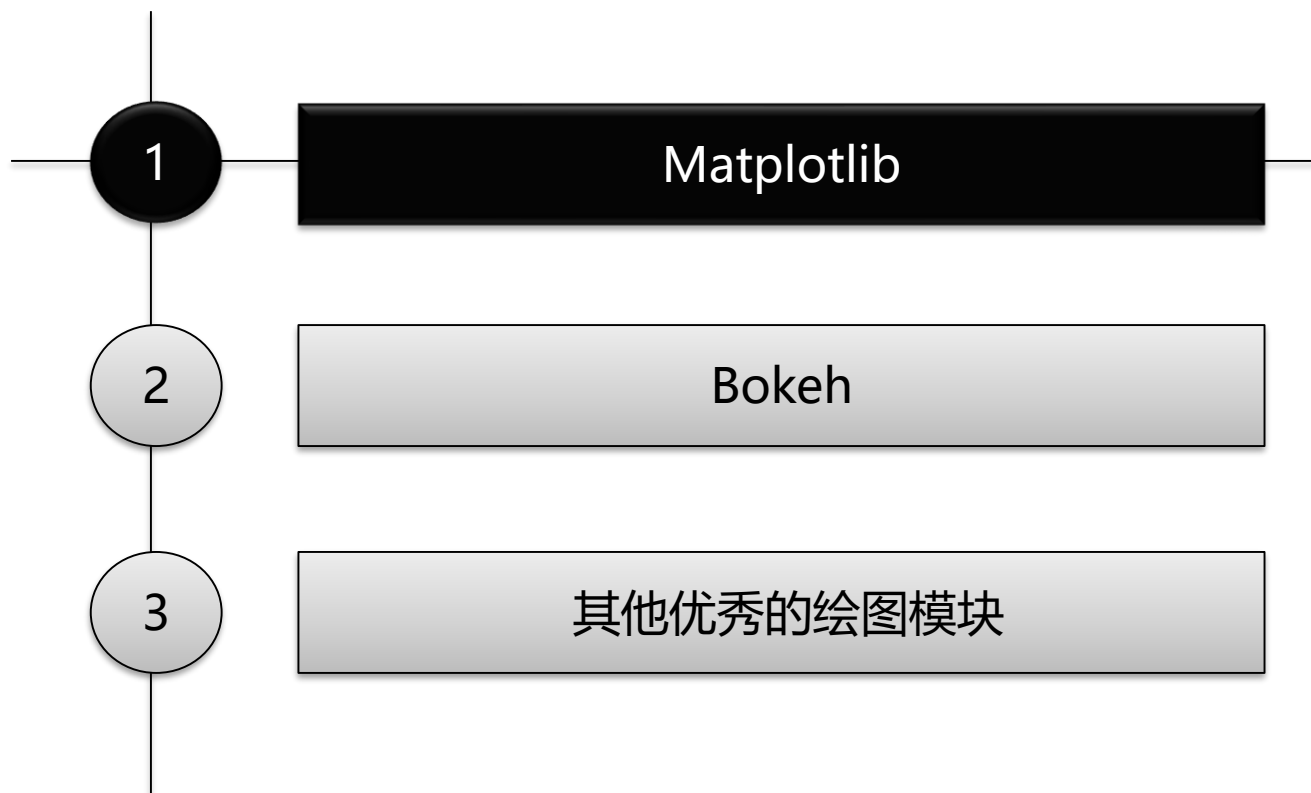


《Python与数据挖掘》

第6章 图表绘制入门

讲师：武永亮



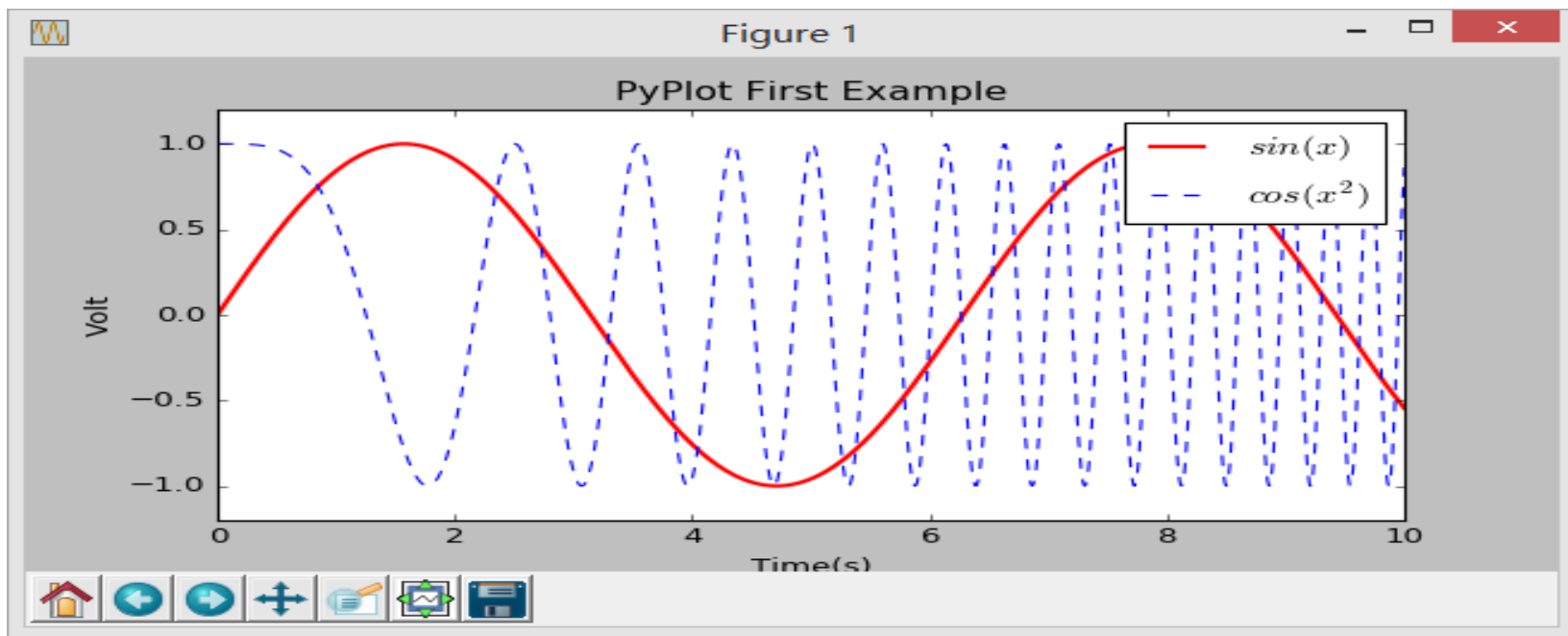


图表绘制介绍

- 读到这里，相信已经掌握了Python的语言基础，包括基本概念和数据结构。
- Python作为开源语言有一种魔力。那就是吸引众多开发者搭建**第三方模块**，使其能充分适应复杂现实的挑战，在众多诉求不同的领域中取得出色表现。
- 图表绘制对于数据分析和可视化环节有不可替代的作用和意义，它能给人带来直观的视觉冲击，快速把握数据的分布和规律。
- 在第7章，我们将重点介绍Matplotlib和Bokeh模块，见识下Python这个多面手的图表绘制能力。

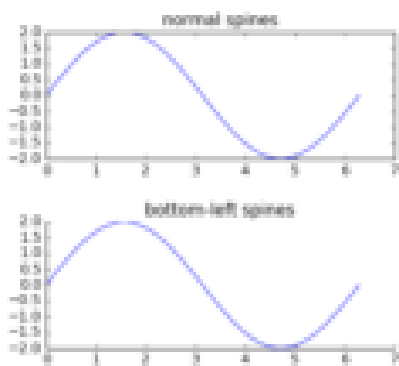
Matplotlib

- Matplotlib是Python中最著名的绘图库。其子库pyplot包含大量与MATLAB相似的函数调用接口，这种函数式编程的思想非常适合进行交互式制图。条形图、扇形图、散点图、等高线图等二维或三维图形都是它的拿手好戏。
- 函数式绘图如下图：

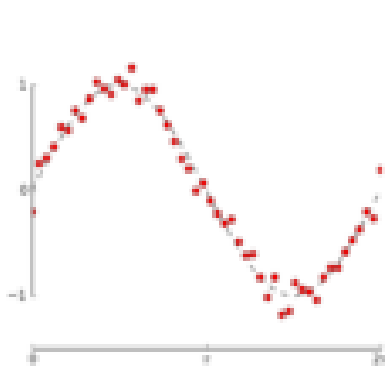


Matplotlib绘图

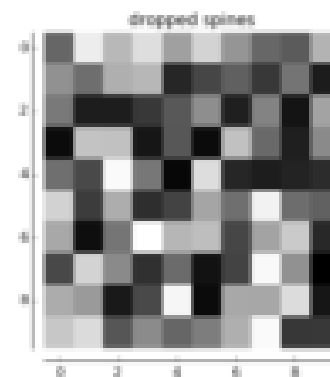
- Matplotlib的官方文档多达几百页，相当完备，并在“画廊(Gallery)”中附有上百幅示例图及对应源代码。这对于新手非常友好。可以在其中找到同个类型的图片，并尝试修改对应代码进行创作。
- Matplotlib Gallery网址为：<http://matplotlib.org/gallery.html>
- Matplotlib官方文档剪影如下图：



spines_demo



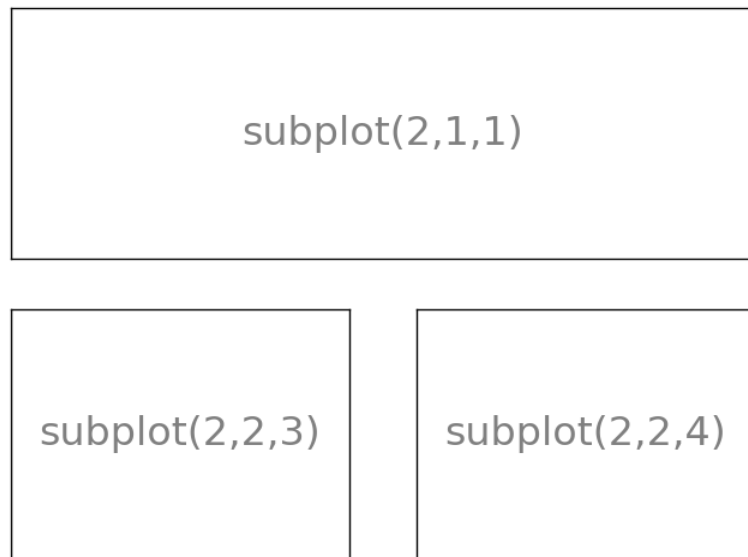
spines_demo_bounds



spines_demo_dropped

Matplotlib绘图

- Matplotlib这一小节作为Matplotlib的入门介绍，将通过一个综合绘图示例来理解和学习Matplotlib函数式绘图中所涉及的基本概念。
- 首先介绍的概念是“子图”。它允许用户将多幅图同时绘制到一个图片窗口之中。这能节省空间，同时允许用户从多个角度展示和解读数据的同时，在数据可视化任务中非常实用，如下图：



- 在函数式绘图中，任何的绘图对象都被看作是一条函数产生的结果。因此，达到这个效果的代码非常简单，子图的声明方法如下：

```
import matplotlib.pyplot as plt
import numpy as np
# 第一部分
plt.subplot(2,1,1) # 参数依次为：行，列，第几项
# 第二部分
plt.subplot(2,2,3)
# 第三部分
plt.subplot(2,2,4)
plt.show()
```

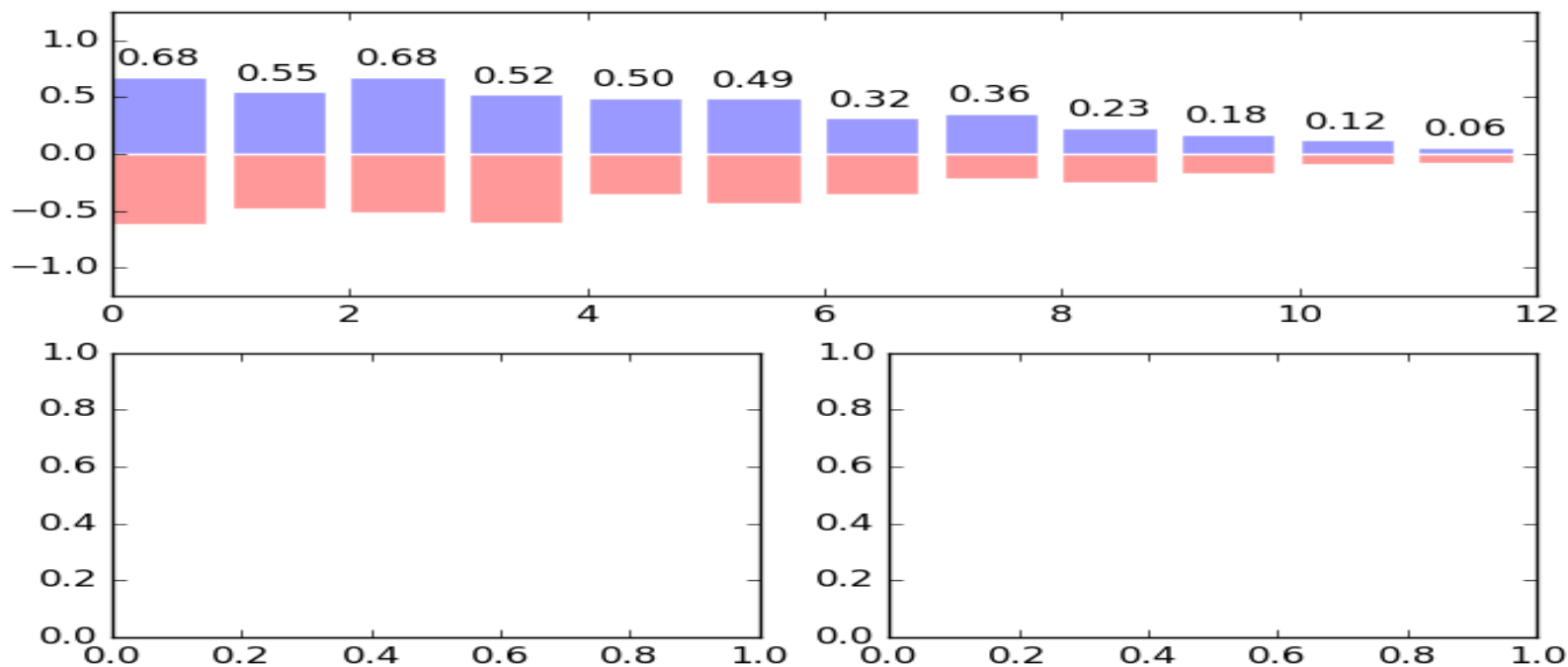
- 接下来，只需要将绘图代码插入两个部分之间，图像就会在用户指定的位置出现。准确地说，插入子图绘制方法plt.subplot()之间。

Matplotlib绘图

```
import matplotlib.pyplot as plt
import numpy as np
# 第一部分
plt.subplot(2,1,1) # 参数依次为：行，列，第几项
n = 12
X = np.arange(n)
Y1 = (1-X/float(n)) * np.random.uniform(0.5,1.0,n)
Y2 = (1-X/float(n)) * np.random.uniform(0.5,1.0,n)
# 利用plt.bar(x, y)绘制柱状图，并指定柱状图颜色，柱子边框颜色
plt.bar(X, +Y1, facecolor='#9999ff', edgecolor='white')
plt.bar(X, -Y2, facecolor='#ff9999', edgecolor='white')
for x, y in zip(X,Y1):
    # 利用plt.text()指定文字出现的坐标和内容
plt.text(x+0.4, y+0.05, '%.2f' % y, ha='center', va= 'bottom')
# 利用plt.ylim(y1, y2)限制图形打印时对应的纵坐标范围
plt.ylim(-1.25,+1.25)
```

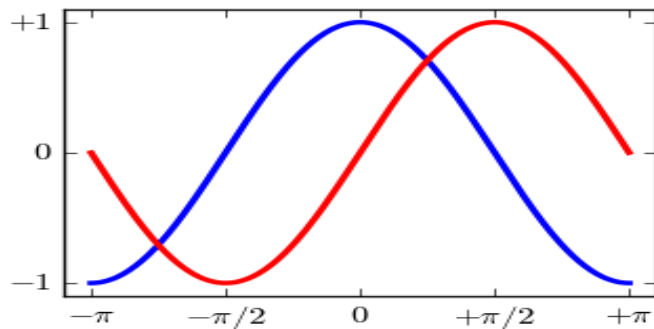
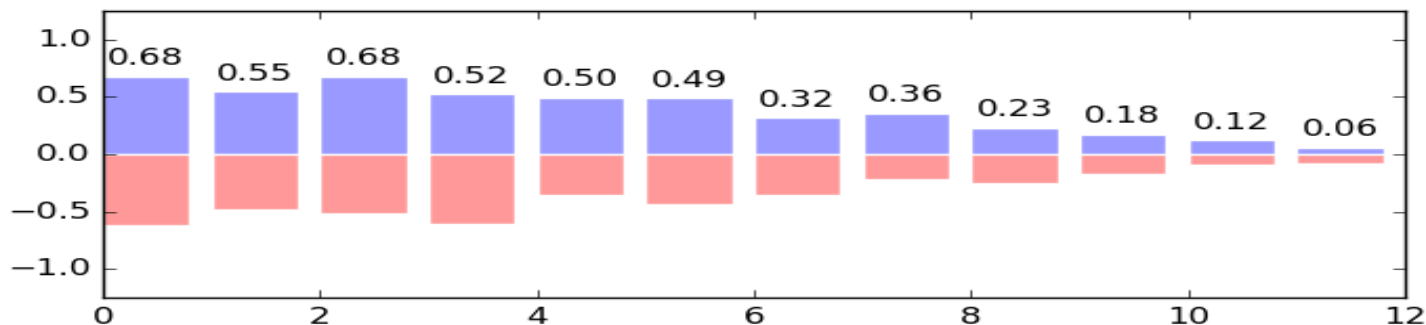

Matplotlib绘图

- 利用Matplotlib的子库pyplot绘制图形时，与MATLAB中函数式绘图的风格非常相似。无论你需要的是一个柱状图，还是显示在图片上的文字，甚至是控制坐标轴的范围，都通过传递参数给对应的绘图函数的方式来实现。此时，图形的表现力更加丰富了，如下图：



Matplotlib绘图

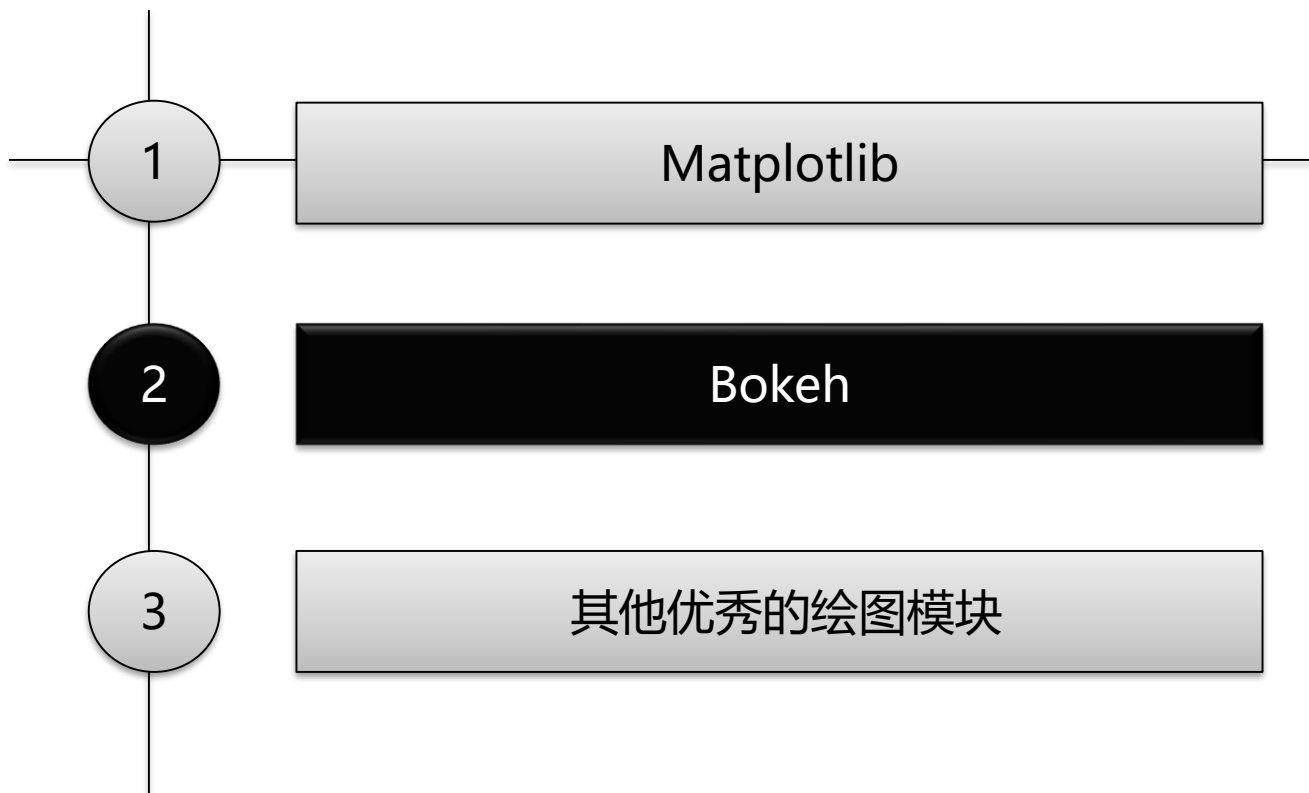
- 类似地，我们继续加入饼状图绘制代码和三角函数曲线绘制代码。值得一提的是：`plt.xticks()`, `plt.yticks()`能够改变坐标轴的刻度文字。通常情况下，绘制三角函数曲线时，我们更加关心 及其倍数的对应取值，而非原始的坐标刻度1,2,3...



Matplotlib的不足

- 尽管函数式绘图能快速出图，但有以下缺点需要指出：
 - a) 函数调用的方法影响效率。
 - b) 图形与内容之间的从属关系被传递函数的方式所掩盖，降低代码的可读性。
 - c) 对于开发者而言，不能直接接触对象，操作对象的数据是致命的。
- 在上面提及的内容中，它们至少涉及了以下四个类：Figure类，FigureCanvas类，Axes类和Line2D类。有能力的应该朝着这个方向，继续探索。

目录

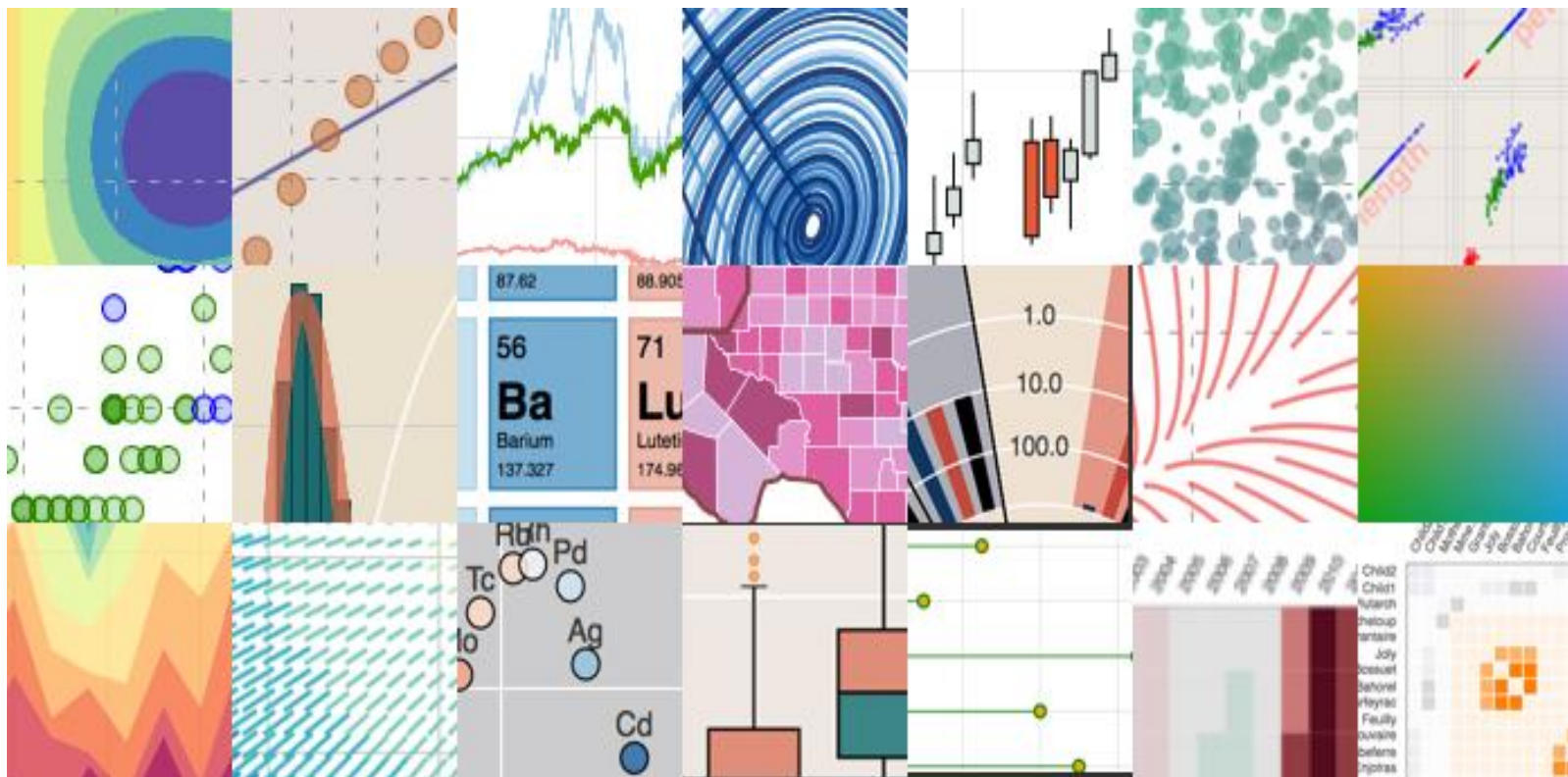


Bokeh绘图

- 与Matplotlib不同，Bokeh是一款针对浏览器中图形演示的交互式绘图工具。
- 它的目标是使用 [d3.js](#) 样式提供优雅，简洁新颖的图形化风格，同时提供大型数据集的高性能交互功能。
- Bokeh 支持用户快速创建交互式的绘图，仪表盘和数据应用。这对于喜爱d3.js的可视化效果，但不熟悉JavaScript的用户有莫大的帮助。因此，在使用IPython Notebook进行编程时，能将Bokeh的交互体验提升至最大。

Bokeh绘图

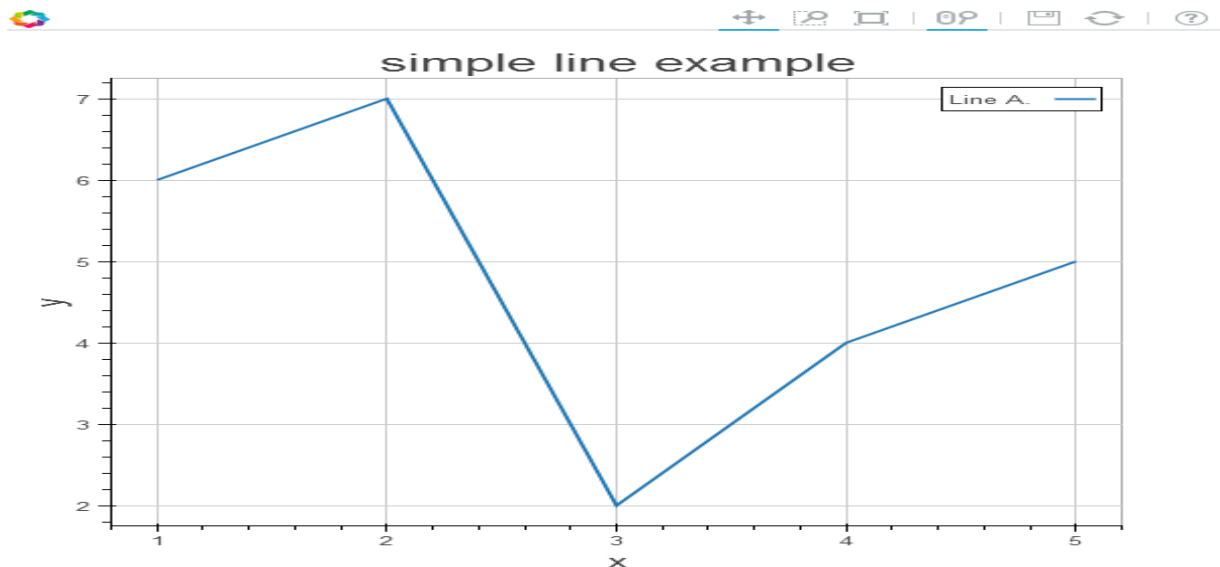
- 其最新的官方文档为<http://bokeh.pydata.org/en/latest/index.html>同样，它也为用户提供一个精彩的画廊(Gallery)以展示基础的例子。
- Bokeh画廊剪影如下图：



Bokeh绘图

```
from bokeh.plotting import figure, output_file, show
x = [1, 2, 3, 4, 5]
y = [6, 7, 2, 4, 5]
output_file("lines.html", title="line plot example") # 输出为静态文件
# 创建一个figure对象，附带标题和坐标轴标记
p = figure(title="simple line example", x_axis_label='x', y_axis_label='y')
p.line(x, y, legend="Line A.", line_width=2) # 添加一条线，设置图例
show(p)
```

- 运行得出结果如下图:



Bokeh的应用

- 在画廊页面中，有非常多生动的交互式例子。例如，交互式的电影检索工具。 Bokeh画廊网页

<http://bokeh.pydata.org/en/latest/docs/gallery.html>

Minimum number of reviews:
10

Dollars at Box Office
(millions):
0

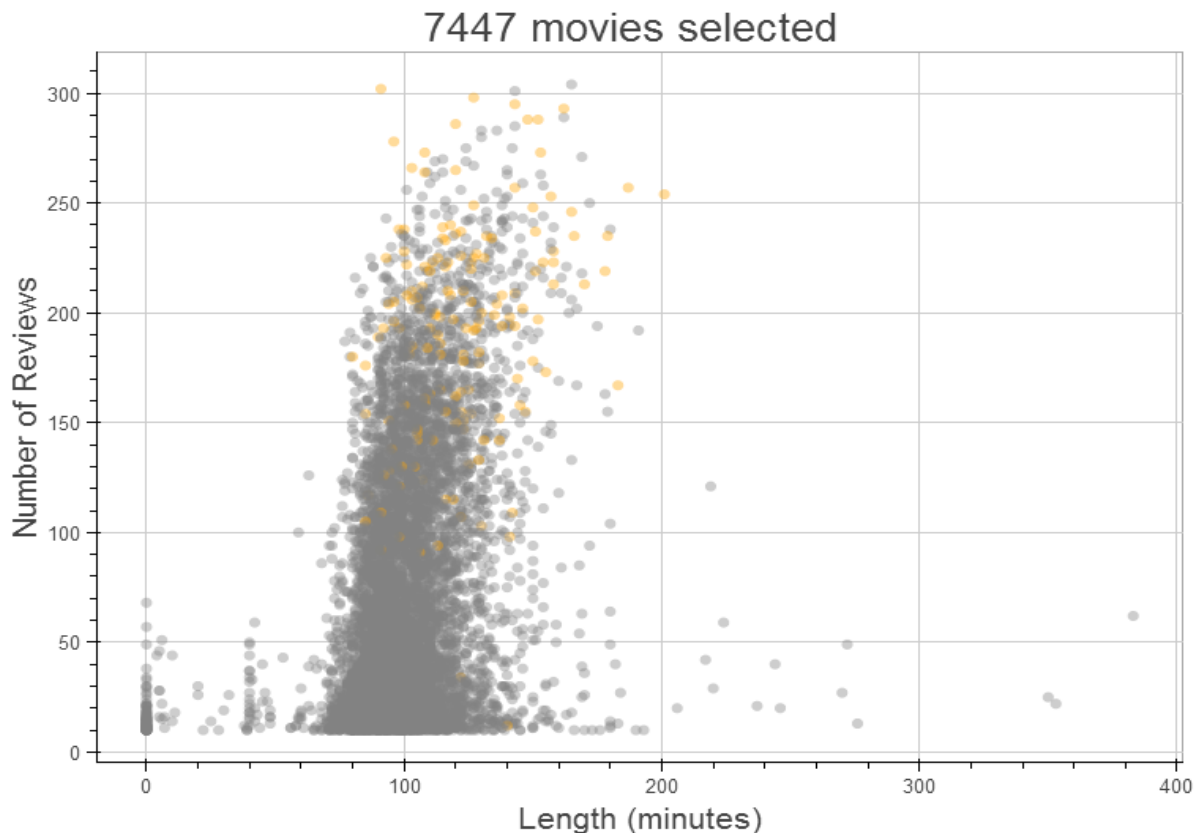
Genre
All

Year released:
2000

End Year released:
2014

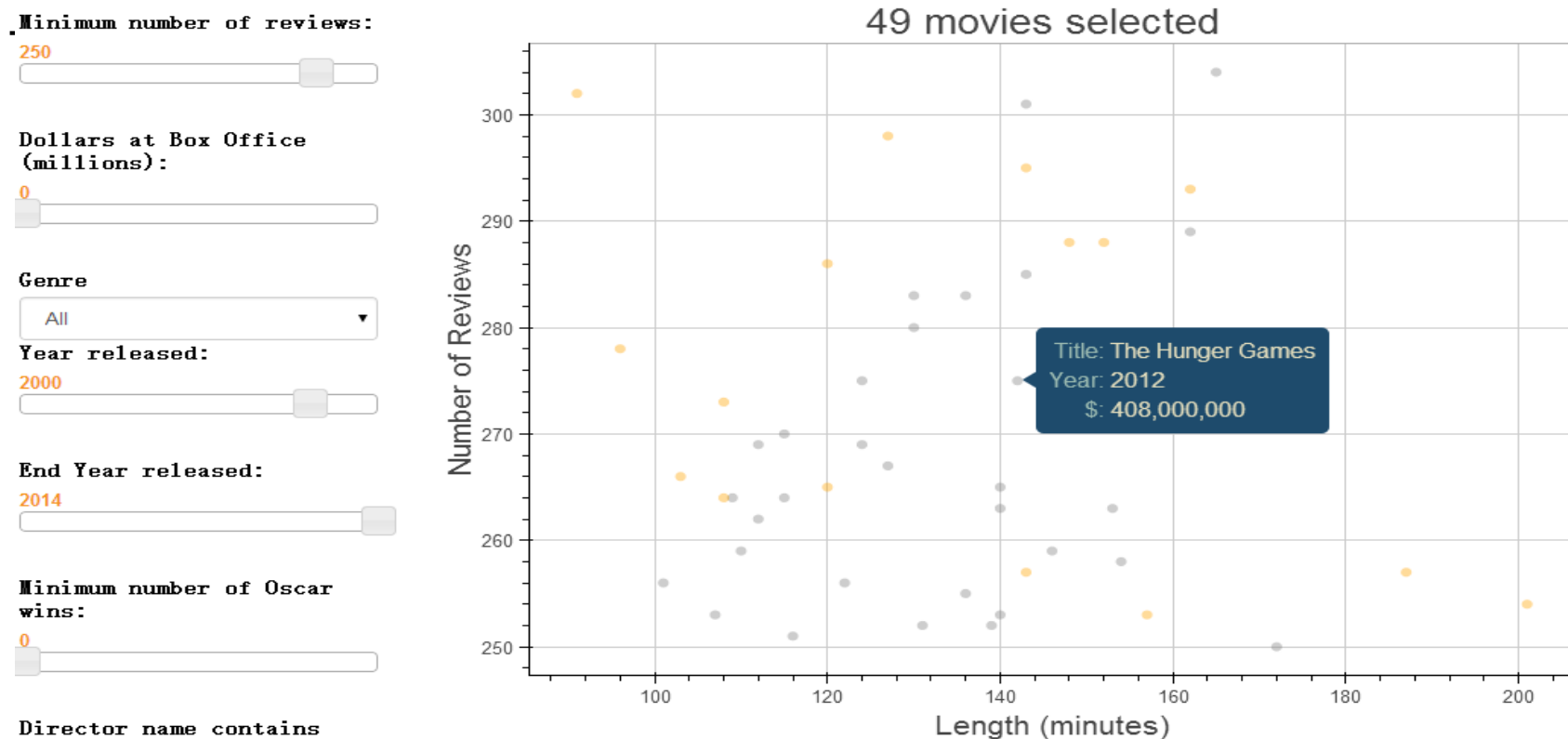
Minimum number of Oscar
wins:
0

Director name contains



Bokeh的应用

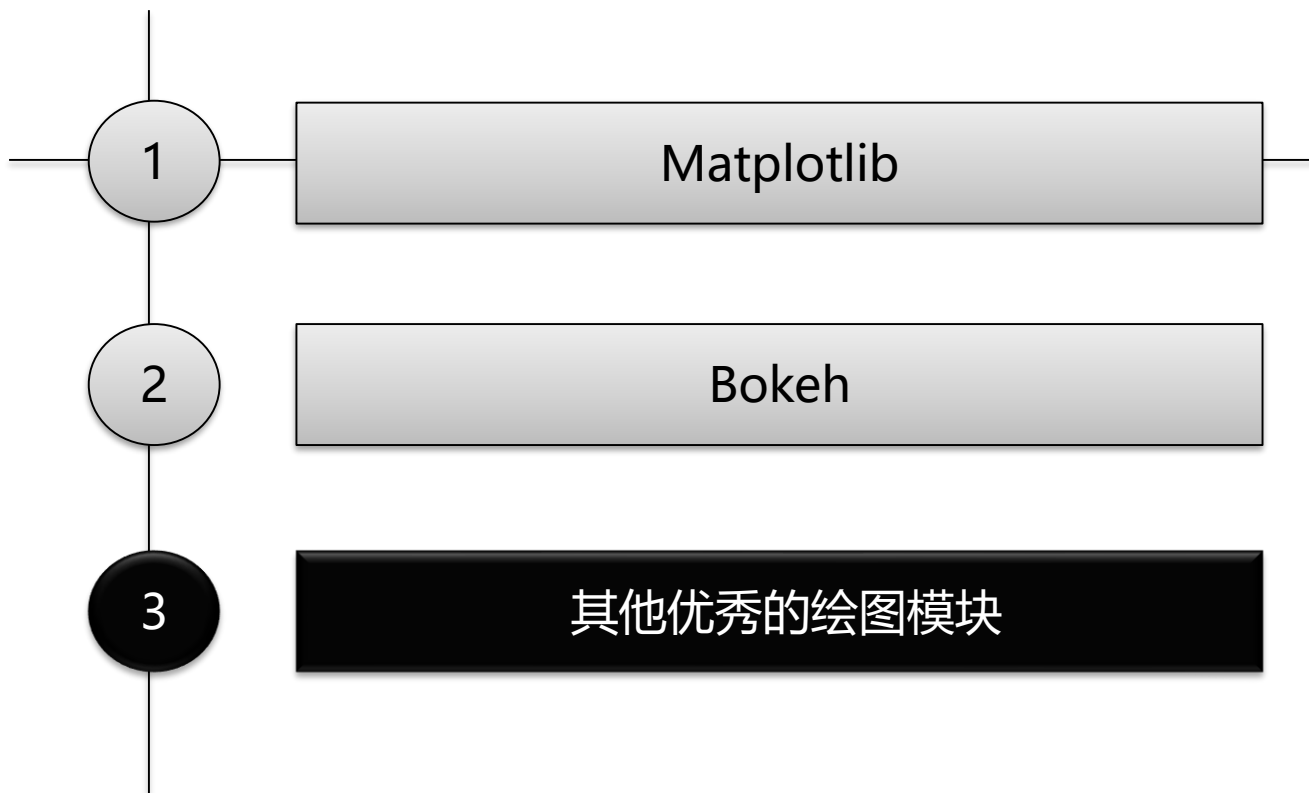
- 可以通过左边预设的过滤器（filter）来改变右边图像的样式和内容。过滤器包括：最小电影评论数、票房、首映时间、奥斯卡奖杯数等。可见，当前有7447部电影被展示在图上。我们将“最小电影评论数”提高到250，以寻找一些经典好片，得到2000年到2014年的49部经典电影，如



Bokeh的应用

- 将鼠标悬停到某一个具体的点上，交互式响应将显示出这个点对应的电影名称、首映时间和票房信息。上图中的是2012年上映的《饥饿游戏》，官方票房统计高达4亿美元。
- 可视化的目的是汇总数据，展示信息。而交互式绘图能够让信息在合适的时机才出现。这种交互体验优于Matplotlib，但这意味着开发者要进行更多的准备工作，以支持用户可能的行为。如果仅为绘制简单的统计图表，Matplotlib将使你更加高效

目录



其他优秀的绘图模块

- 以下附一张可视化任务中优秀模块的功能简介，希望能够帮助提升视野，在需要深入学习数据可视化时能更有方向感。
- 需要指出的是，大部分场合，Matplotlib和Bokeh都能胜任具体的可视化任务，这也是它们成为Python可视化中最出色模块的理由之一。

模块名称	用途
VisPy	简单快速、可拓展性强的交互式科学（天文、物理等）绘图工具
Glumpy	VisPy的姐妹项目，专注于2D/3D的高性能数据可视化工具
Seaborn	基于Matplotlib和NumPy等，用于制作表现力强且美观的信息图表
Kivy	快速开发应用程序中创新的用户交互界面，如多点触控
Folium	提供Leaflet.js的Python编程接口，方便地将数据可视化于地图之上
NetworkX	用于创造、操作、研究和绘制复杂网络的结构图和机理

Thank You!