**Matplot绘图基本操作示例**

1.2 绘制人口数量折线图

　　下面绘制世界人口数量与年份的关系图。year和pop列表分别用于表示从2005年开始的年份和该年份对应的全球人口数量（单位为10亿），通过plt.plot函数绘制折线图，最后利用plt.show函数显示图像。

import matplotlib.pyplot as plt

year = [2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014]

pop = [6.49, 6.558, 6.656, 6.725, 6.804, 6.884, 6.965, 7.043, 7.125, 7.207]

plt.figure() # 创建图表

plt.plot(year,pop) # 绘制图像

plt.show() # 显示图像

　　操作步骤：将代码写入左边的文本区域，并点击绿色的执行按钮。弹出执行结果，即人口数量的折线图。  
　　之后的实验操作与此步骤相同，只显示实验结果。

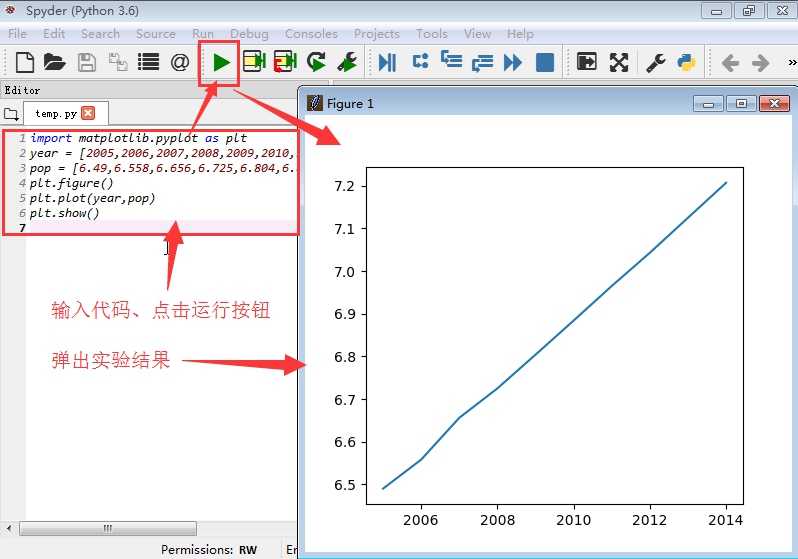


图2

　　1.3 自定义线的样式和颜色

　　在实际应用中，有时候需要在一张图中同时绘制多条线，那么为了区分它们常常使用不同的样式和颜色进行区分，而样式和颜色的定义只要通过plt.plot中的两个参数就能方便地定义。下面同时世界人口和中国人口数量的折线，其中世界人口用绿色实线，而中国人口用红色虚线加以表示。线的样式和颜色分别用linestyle和color两个属性进行设置。

plt.figure()

pop\_cn = [1.30756, 1.31448, 1.32129, 1.32802, 1.33450, 1.34091, 1.34735, 1.35404, 1.36072, 1.36782]

plt.plot(year,pop,color = "g")

plt.plot(year,pop\_cn,linestyle ="--", color = "r")

plt.show()



图3

　　1.4 添加图例

　　对于上图这种多条曲线的情况，可以通过添加图例更加清晰的表达各条曲线的含义。添加图例，首先要在添加图像时设置label参数，随后就能运用plt.legend()函数自动创建图例。

plt.figure()

plt.plot(year,pop,color = "g",label = "World population")

plt.plot(year,pop\_cn,linestyle ="--", color = "r",label = "China population")

plt.legend()

plt.show()

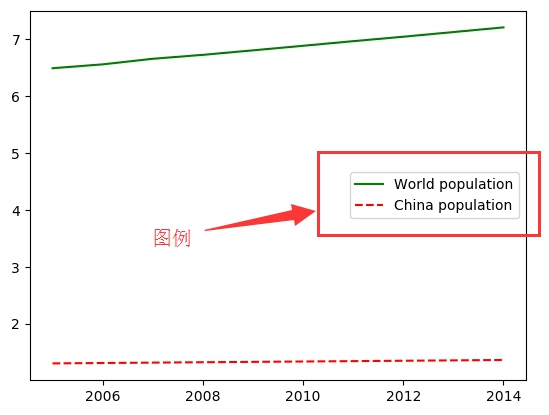


图4

　　1.5 添加标题和轴标签

　　用plt.title、plt.xlabel和plt.ylabel分别为世界人口数量图表设置标题和轴标签。

plt.plot(year, pop)

plt.title("World Population Summary")

plt.xlabel("Year")

plt.ylabel("Polulation")

plt.show()

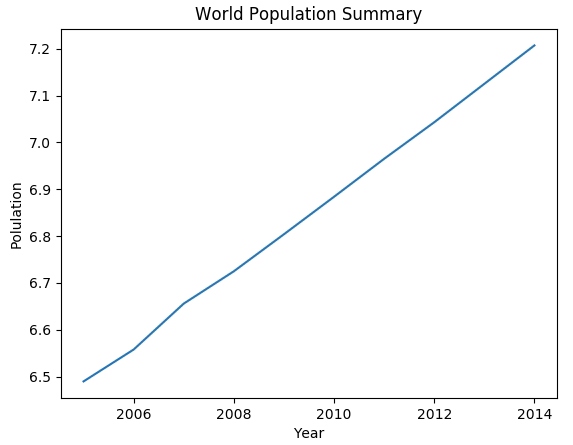


图5

　　1.6 设置刻度以及刻度标签

　　用pyplot作图时会根据数据的情况自动标识刻度和刻度标签，但是用户也可以自定义刻度以及刻度标签。下面运用plt.yticks对y轴的刻度和刻度标签进行设置，其中'6B'表示6 billion，即60亿。

plt.figure()

plt.plot(year, pop)

plt.title("World Population Summary")

plt.xlabel("Year")

plt.ylabel("Polulation")

plt.yticks([6, 6.5, 7, 7.5], ['6B', '6.5B', '7B', '7.5B'])

plt.show()

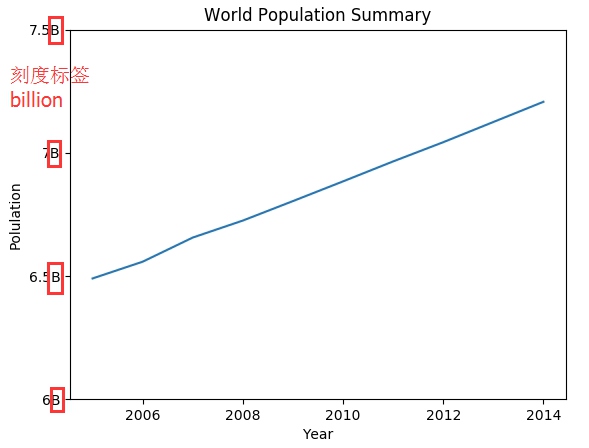


图6

　　1.7 绘制散点图

　　散点图即数据点在直角坐标系平面上的分布图。下面用函数plt.scatter绘制世界人口数量的散点图。

plt.figure()

plt.scatter(year, pop)

plt.title("World Population Summary")

plt.xlabel("Year")

plt.ylabel("Polulation(Billion)")

plt.show()

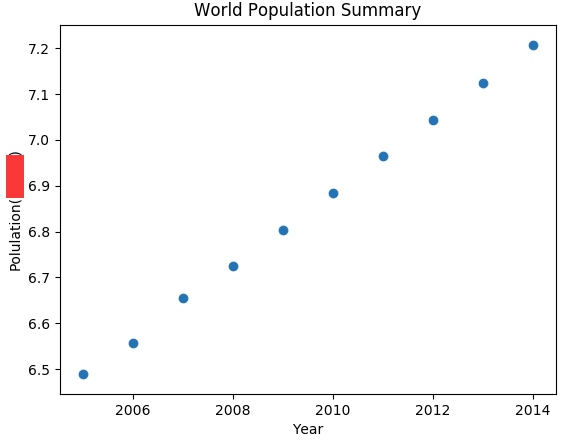


图7

　　1.8 给图标添加注解

　　对于下面用plt.text函数在2013年的人口数据点略上方标识"2013"字样，该数据点的坐标为(2013,7.125)，设置注解的坐标为(2013,7.13)。

plt.figure()

plt.scatter(year, pop)

plt.title("World Population Summary")

plt.xlabel("Year")

plt.ylabel("Polulation(Billion)")

plt.text(2013,7.13,'2013')

plt.show()

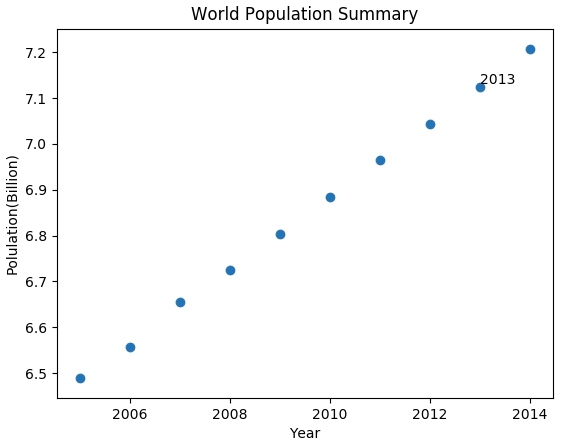


图8

　　1.9 绘制共享x轴的两个子图

　　利用plt.subplots函数创建包含多个子图的绘图对象，设置子图个数为2，并设置sharex为True来设定共享x轴，在两个子图中用plt.scatter分别绘制世界和中国的人口数量，并且用实例方法set\_title设置各子图对象的标题。

f, (ax1,ax2) = plt.subplots(2, sharex=True)

ax1.scatter(year, pop)

ax1.set\_title('World Population Summary')

ax2.scatter(year, pop\_cn)

ax2.set\_title('China Population Summary')

plt.show()

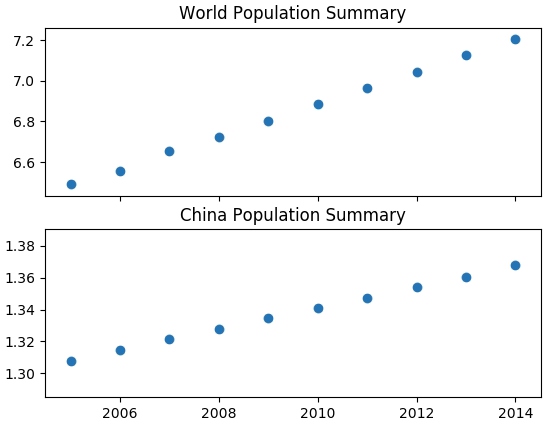


图9

　　1.10 绘制双y轴的图表

　　如果要比较两条曲线的关系，而两者的y轴取值范围又存在一定差别，则可以绘制双y轴的图表，用Axes.add\_subplot函数为绘图对象fig添加两个子图ax3和ax4，在子图ax3中用实例方法Axes.twinx，将ax4设为与ax3共享x轴，并且y轴在右侧。

fig = plt.figure()

ax3 = fig.add\_subplot(111)

ax3.plot(year,pop,color = "g",label = "World population")

ax3.set\_ylabel('World population')

ax3.set\_title("Double Y axis")

ax4 = ax3.twinx()

ax4.plot(year,pop\_cn,linestyle ="--", color = "r",label = "China population")

ax4.set\_ylabel('China population')

ax4.set\_xlabel('Year')

plt.show()

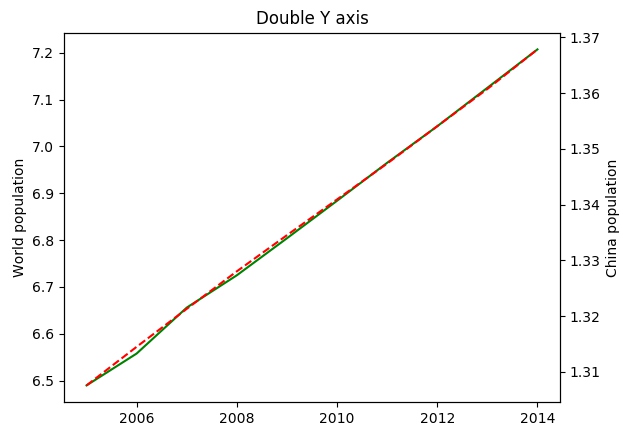


图10

　　1.11 将图片保存到文件

　　利用plt.savefig函数将最后的绘图对象保存为SVG文件，并且设置参数dpi即分辨率为400，设置参数bbox\_inches为tight，即尝试剪除图表周围的空白部分。  
　　使用上一步绘制的代码。

fig = plt.figure()

ax3 = fig.add\_subplot(111)

ax3.plot(year,pop,color = "g",label = "World population")

ax3.set\_ylabel('World population')

ax3.set\_title("Double Y axis")

ax4 = ax3.twinx()

ax4.plot(year,pop\_cn,linestyle ="--", color = "r",label = "China population")

ax4.set\_ylabel('China population')

ax4.set\_xlabel('Year')

plt.show()

plt.savefig('figure.svg', dpi=400, bbox\_inches = 'tight')

　　保存的结果可以在“C:\Users\Administrator\.spyder-py3”目录中看到。文件名为“figure.svg”

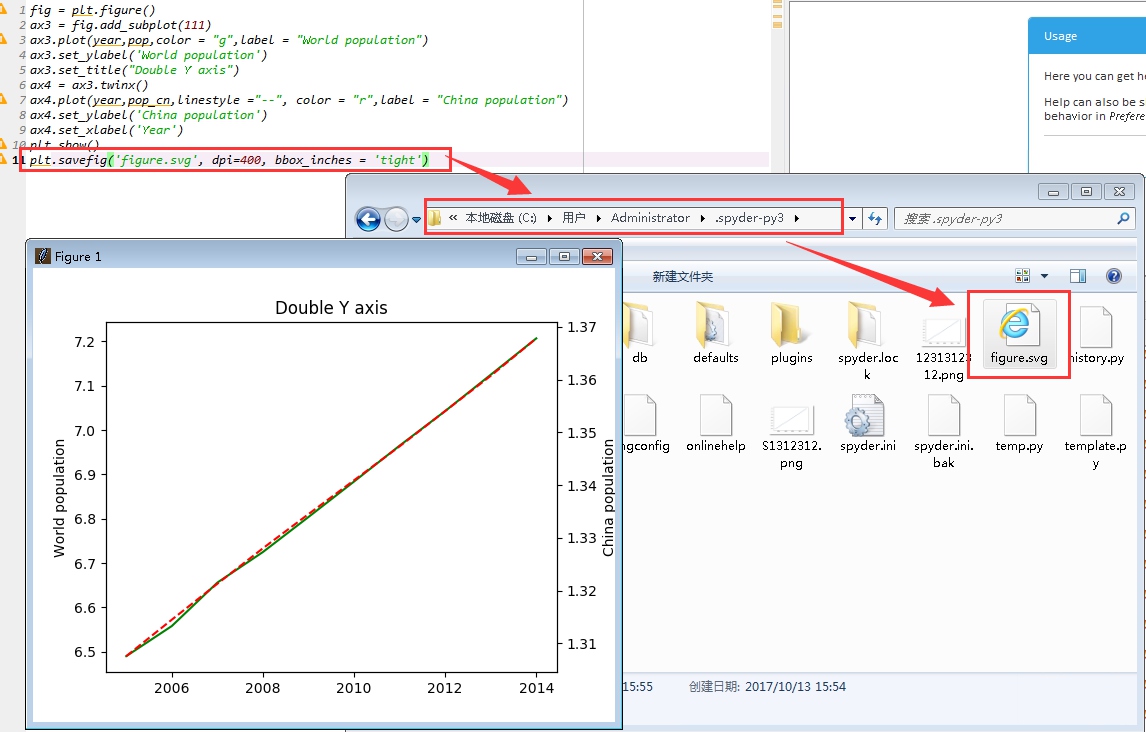


图11

　1.12 绘制柱状图  
　　柱状图是一种以长方形的长度为变量的表达图形的统计报告图，由一系列高度不等的纵向条纹表示数据分布的情况，用来比较两个或以上的价值（不同时间或者不同条件），只有一个变量，通常利用于较小的数据集分析。柱状图亦可横向排列，或用多维方式表达。  
　　依然使用上边的数据绘制柱状图。

fig = plt.figure()

plt.bar(year,pop,0.4,color="green")

plt.xlabel("class")

plt.ylabel("size")

plt.title("bar chart")

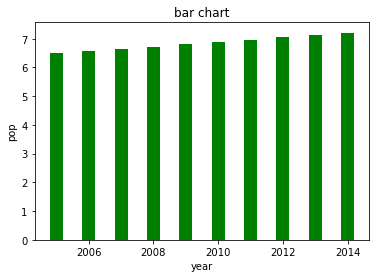


图12