样式管理

管理图表的艺术

画一个吸引人注意的图表相当重要。当你探索一个数据集，需要画图表，图表看起来令人愉悦是件很高兴的事。在与你的观众交流观点时，可视化同样重要，同时，也很有必要去让图表吸引[注意力](http://www.wredian.com/tags-注意力-0.html)和印入脑海里。

Matplotlib自动化程度非常高，但是，掌握如何设置系统以便获得一个吸引人的图是相当困难的事。为了控制matplotlib图表的外观，Seaborn模块自带许多定制的[主题](http://www.07net01.com/tags-主题-0.html)和高级的[接口](http://www.07net01.com/tags-接口-0.html)。

%matplotlib inline

**import** **numpy** **as** **np**

**import** **matplotlib** **as** **mpl**

**import** **matplotlib.pyplot** **as** **plt**

np.random.seed(sum(map(ord, "aesthetics")))

让我们先定义一个函数用来画正弦函数，这将帮助我们了解我们可以控制的不同风格的参数

**def** sinplot(flip=1):

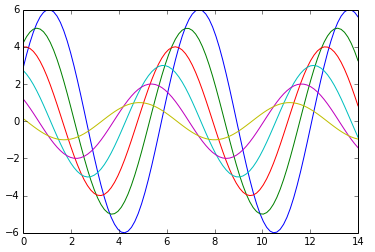
x = np.linspace(0, 14, 100)

**for** i **in** range(1, 7):

plt.plot(x, np.sin(x + i \* .5) \* (7 - i) \* flip)

默认情况下matplotlib的画的图是这样的：

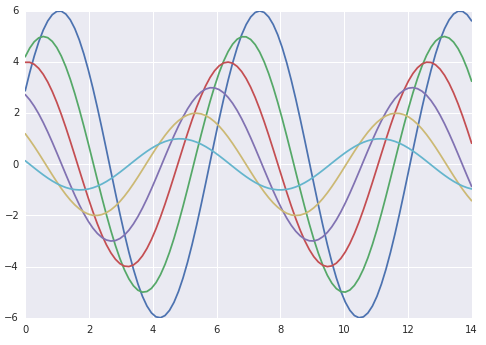
sinplot()



转换成Seaborn模式画图，只需要引入seaborn模块。

**import** **seaborn** **as** **sns**

sinplot()



seaborn默认浅灰色背景与白色[网络](http://www.wredian.com/tags-网络-0.html)线的灵感来源于matplotlib，却比matplotlib的颜色更多柔和。我们发现，网络线对于传播信息很有用，几乎在所有情况下，人们喜欢图甚于表。默认情况下白灰网格的形式可以避免过于刺眼。在多面作图的情况下，网络形式显得相当的有利，提供了一种作图结构，这对模块中的一些复杂工具非常重要。

seaborn将matplotlib的参数划分为两个组。第一组控制图表的样式，第二组控制图的度量尺度[元素](http://www.07net01.com/tags-元素-0.html)，这样就可以轻易在纳入到不同的上下文中。

操控这些参数由两个函数提供接口。控制样式，用axes\_style()和set\_style()这两个函数。度量图则用plotting\_context()和set\_context()这两个函数。在这两种情况下，第一组函数返回一系列的参数，第二组则设置matplotlib的默认属性。

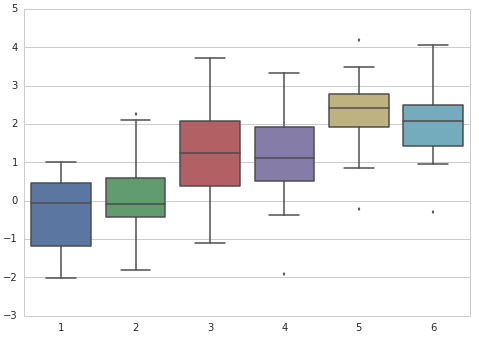
**使用axes\_style()和set\_style()控制图表样式**

seaborn目前有五种预设的样式：darkgrid（灰色网格）、whitegrid（白色网格）、dark（灰色）、white（白色）和ticks。它们根据不同人的爱好被应用于不同的使用场景。默认的样式为darkgrid。如上面所述，网格线对于传播信息很有用，而白灰背景样式有助于更好的展示数据。whitegrid样式类似，但它更适用于绘制带有复杂数据元素的图表，比如箱型图：

sns.set\_style("whitegrid")

data = np.random.normal(size=(20, 6)) + np.arange(6) / 2

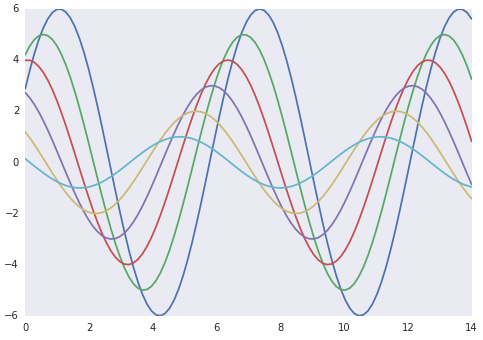
sns.boxplot(data)



对于大多数的图表来说，网格线有些时候并不是必要的。

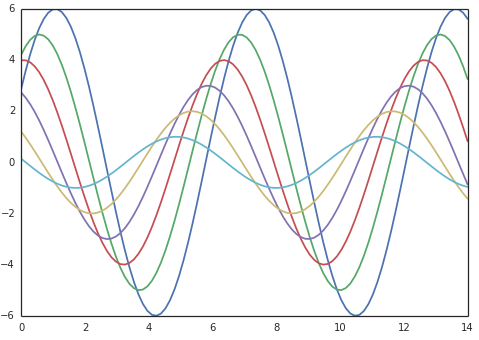
sns.set\_style("dark")

sinplot()



sns.set\_style("white")

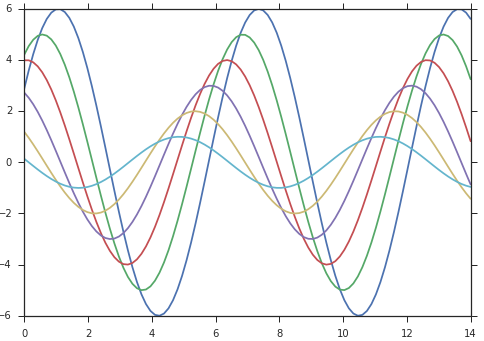
sinplot()



有些时候，你想突出显示图表的刻度线，此时ticks样式会更合适：

sns.set\_style("ticks")

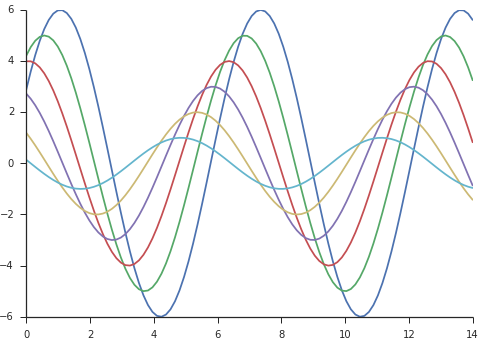
sinplot()



**使用despine()删除图表上边框和右边框**

white和ticks样式图表都会包含没有必要的上边框和右边框线，通过matplotlib的参数想去掉这两个边框是不可能的，但是你可以通过seaborn的despine()函数来删除它们：

sns.despine(sinplot())



你还可以通过附加的参数来控制需要删除哪些边框线：