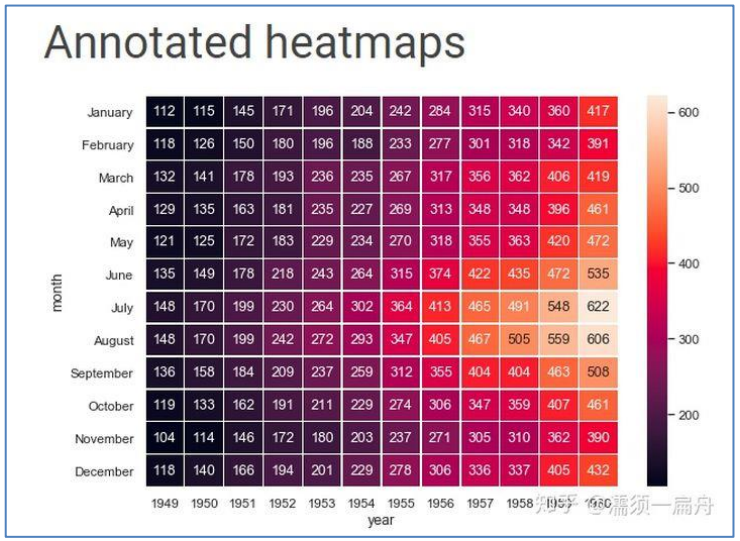
数据可视化

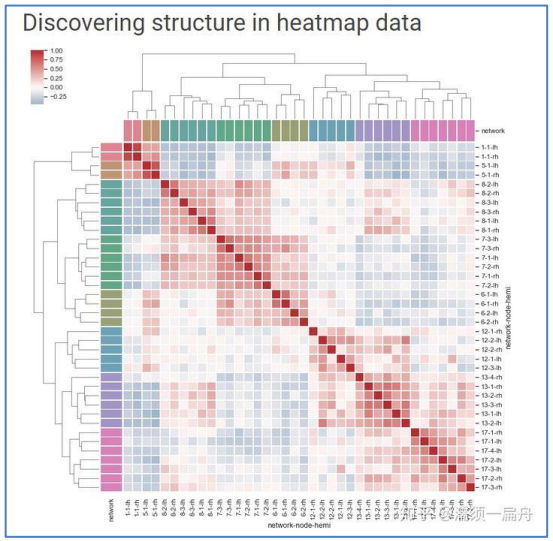
目录

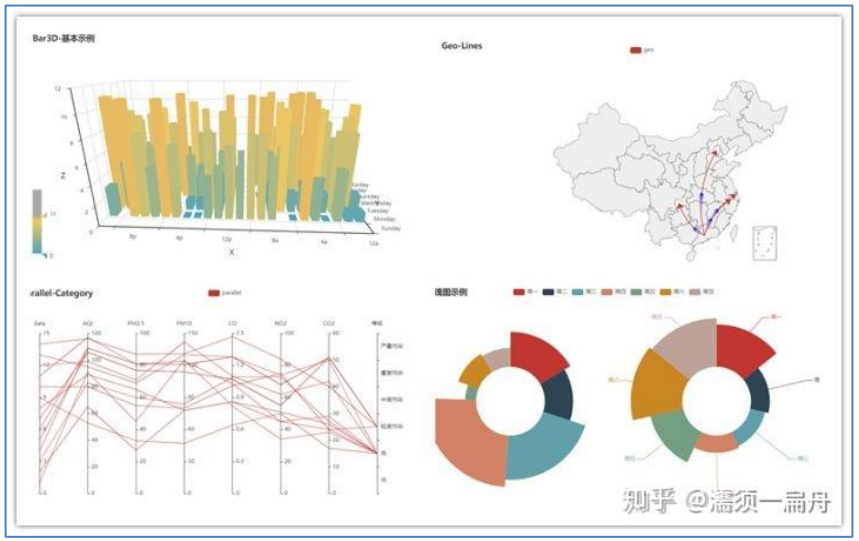
Python数据可视化

信息可视化是数据分析中最重要的任务之一，可视化可能是探索过程的一部分，对部分人来说，构建网络交互式可视化可能是终极目标。Python有很多附加库可以用来制作静态或动态的可视化文件。其中matplotlib属于python比较底层的可视化库，可定制性强、图表丰富、简单易用。 pandas绘图和Seaborn是基于matplotlib开发的更加简便的绘图方式。

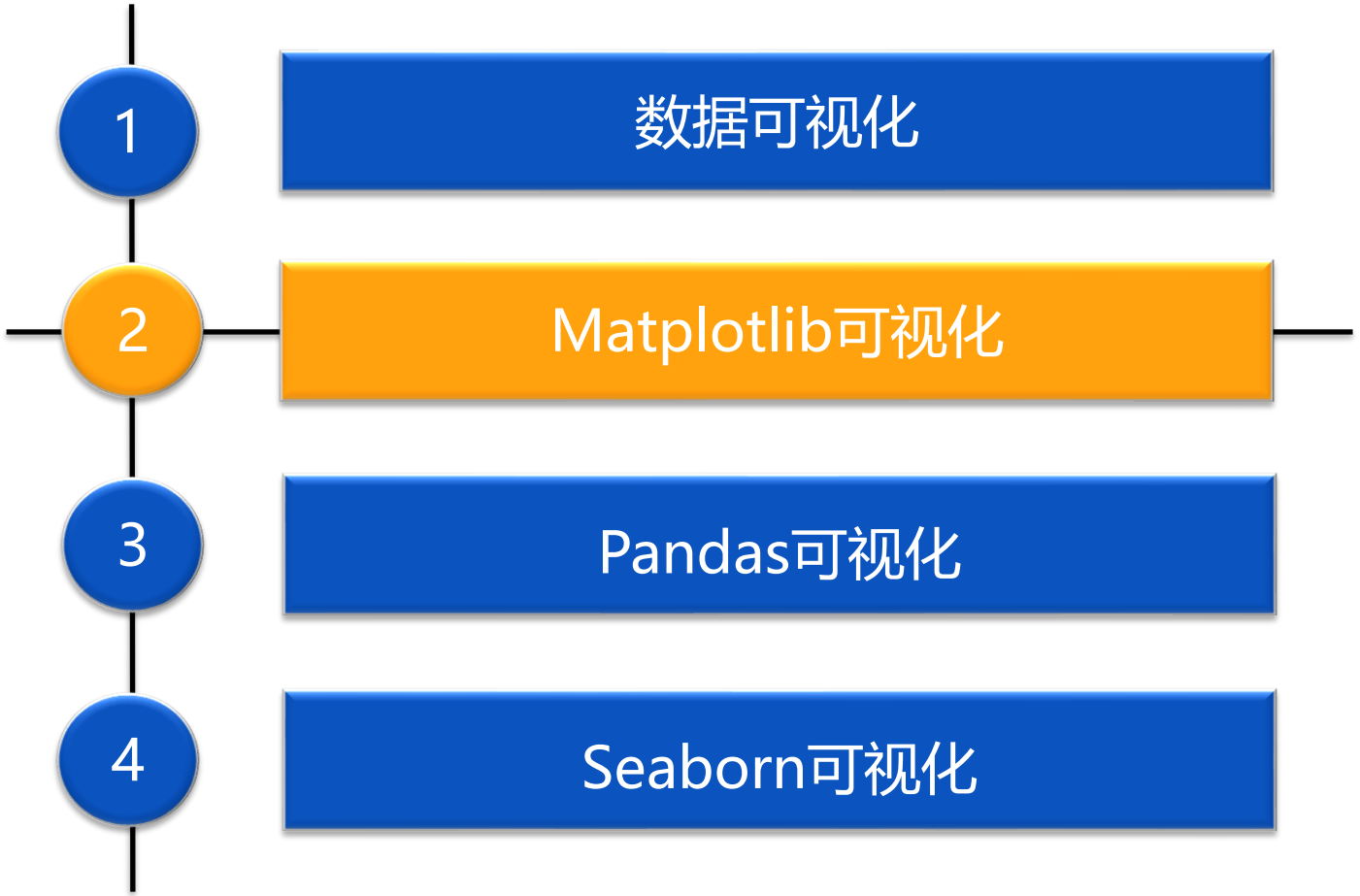
其他可视化库有：pyecharts、ggplot、plotnine、holoviews、basemap等等







目录



matplotlib 简介

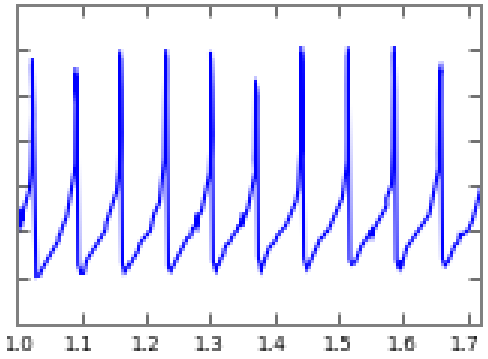
Matplotlib是Python中最常用的可视化工具之一，可以非常方便地创建海量类型地2D图表和一些基本的3D图表，可根据数据集（DataFrame，Series）自行定义x,y轴，绘制图形（线形图，柱状图，直方图，密度图，散布图等等），能够解决大部分的需要。

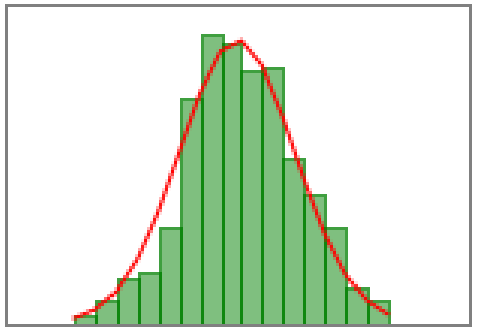
John Hunter

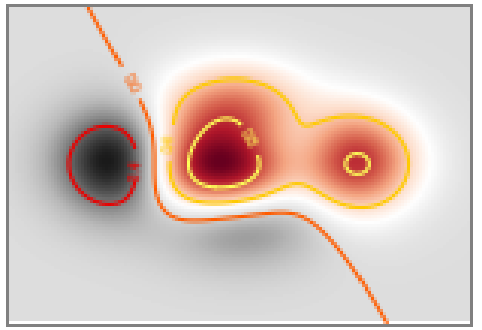
Matplotlib的优点有哪些？

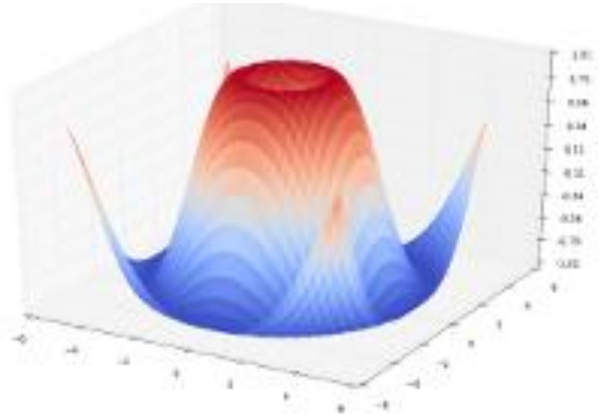
- 具有良好的操作系统兼容性和图形显示接口兼容性
- 支持几十种图形显示接口与输出格式

*由于matplotlib中参数较多后续会用到*args（可变参数），**kwargs（关键字参数）表示参数设置*



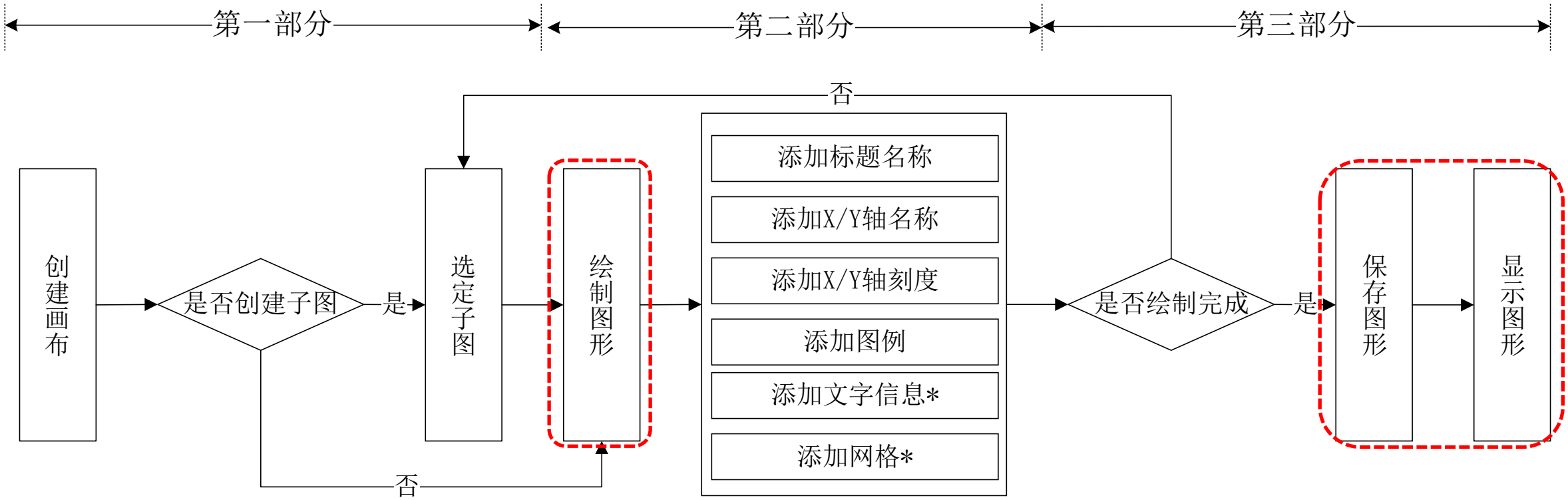






Matplotlib使用预备知识

Matplotlib画图流程



- Matplotlib导入设置，一般导入matplotlib.pyplot，并设置为**plt**
- 画图结果用**plt.show( )**查看, *一个Python会话中一般只使用一次**plt.show( )**，放在最后

创建画布

figure函数详解

matplotlib.pyplot.figure(*num=None, figsize=None, dpi=None, facecolor=None, edgecolor=None, frameon=True, FigureClass=<class 'matplotlib.figure.Figure'>, clear=False, **kwargs***)**

num： 图片的编号，不设置则默认增长

figsize： 可选参数。整数元组，默认是无。提供整数元组则会以该元组为长宽，若不提供，默认为rc fiuguer.figsize。例如(4,4)即以长4英寸 宽4英寸的大小创建一个窗口

dpi： 可选参数，整数。表示该窗口的分辨率，如果没有提供则默认为 figure.dpi

***color*： 控制指定部位的颜色

Python画图不支持中文，如果图片中需要添加中文字体，需要提前设置

plt.rcParams['*font.sans-serif*'] = ['*SimHei*']

plt.rcParams['*axes.unicode_minus*'] = *False*

子图画布设置及选定

子图画布subplot函数详解

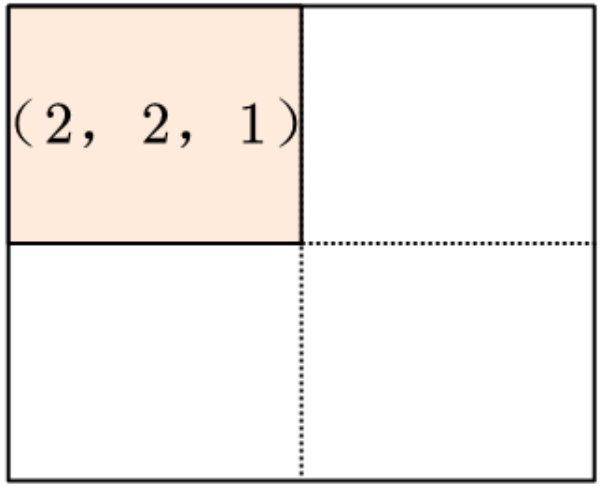
matplotlib.pyplot.subplot(*nrows, ncols, index, **kwargs***)**

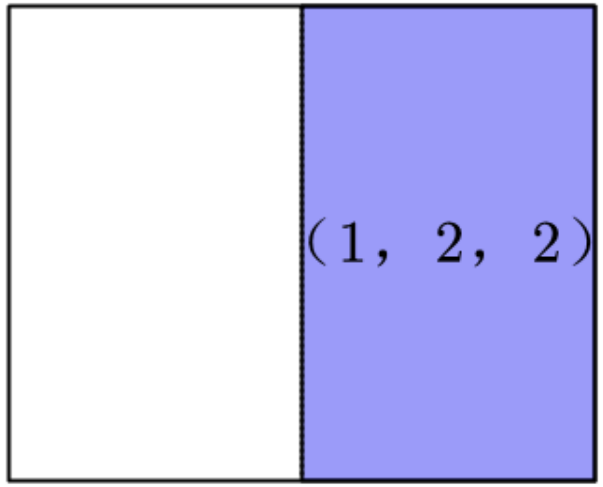
nrows： 行数

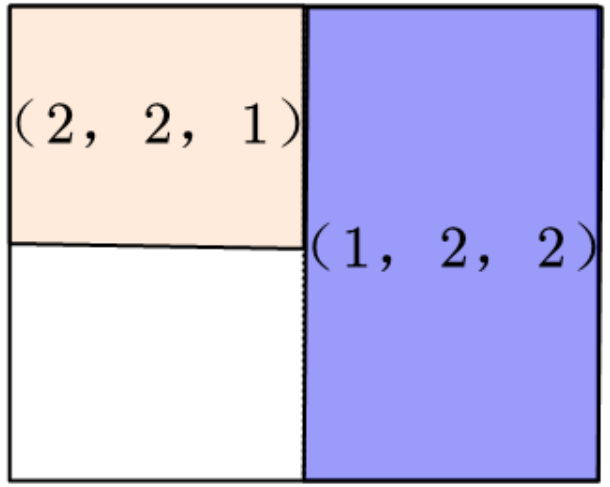
ncols： 列数

index： 索引

异形子图拼合







关于子图间距的调节函数

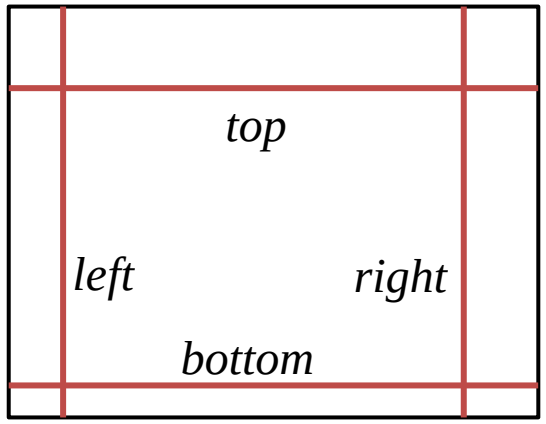
子图之间距离较小时无法显示全部信息，可以使用subplots_adjust函数调节

subplots_adjust函数详解

plt.subplots_adjust(*left=None, bottom=None, right=None, top=None, wspace=None, hspace=0.5***)**

left, right, bottom, top： 子图所在区域的边界。当值大于1.0的时候子图会超出figure的边界从而显示不全；值不大于1.0时，子图会自动分布在一个矩形区域（下图灰色部分）

要保证left < right, bottom < top



绘制一个简单的函数并保存

以 $y=\sin(x)$ $y=\cos(x)$ 为例

plot函数详解

matplotlib.pyplot.plot(*[x], y, [fmt], *, data=None, **kwargs***)**

matplotlib.pyplot.plot(*[x], y, [fmt], [x2], y2, [fmt2], *, data=None, **kwargs***)**

[x]，y： 数组/张量，表示数据点的水平/垂直坐标。*x*值是可选的，默认为range(len(y))

[fmt]： str,可选格式字符串，可以用简写(eg:'ro')，也可以使用color=''，linestyle=''

*plot函数可以一次绘制多个函数图像

savefig函数详解

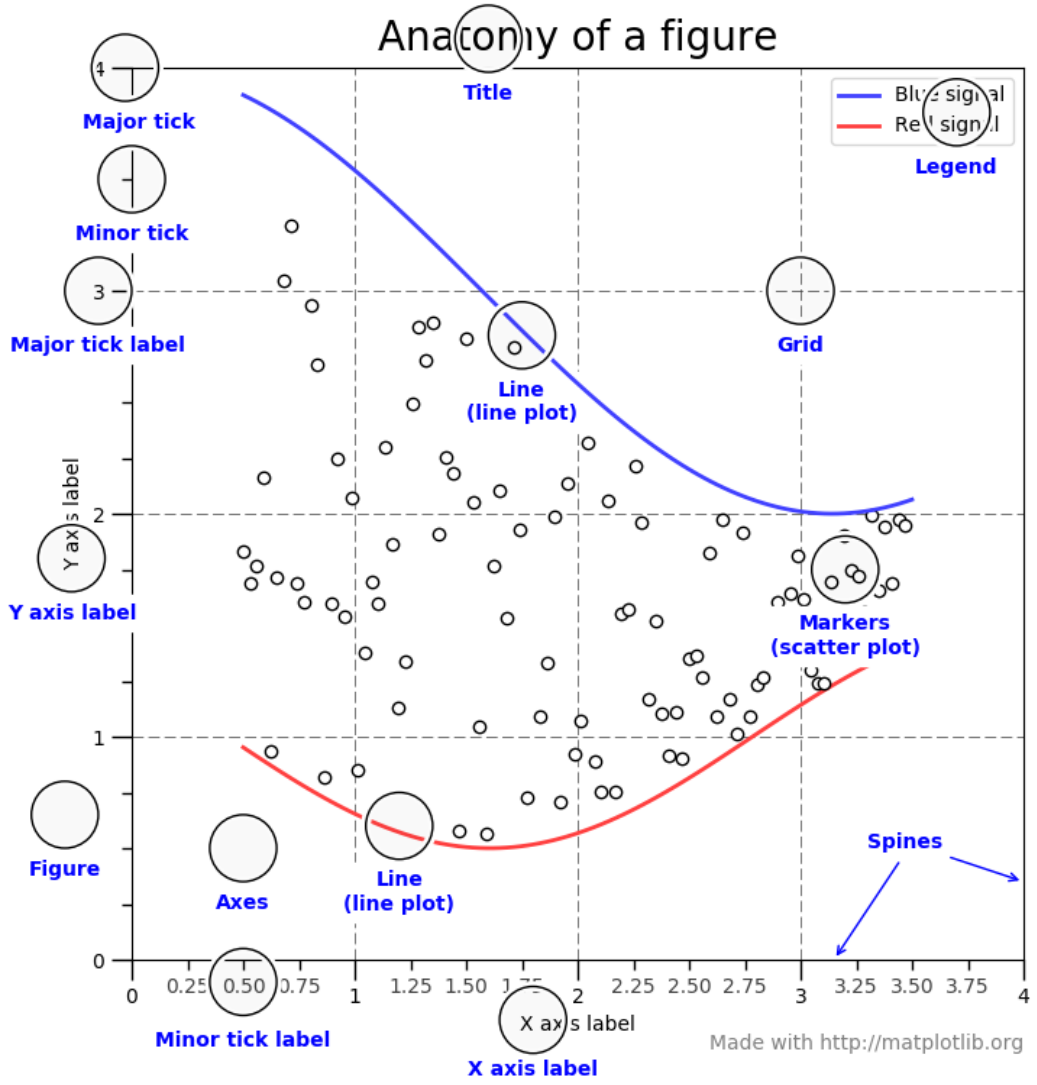
matplotlib.pyplot.savefig(*fname,**kwargs***)** **plt.savefig(***'my_picture.png'***)**

*函数支持：eps、jepg、jpg、pdf、png,ps,raw,tgba,svg,svgz,tif,tiff，默认png

给图片添加关键信息

关键信息说明

- 题目
- 图例
- X/Y轴标签
- X/Y轴刻度
- 网格
- 文字标记



关键信息之标题

title函数详解

matplotlib.pyplot.title(*label, fontdict=None, loc=None, pad=None, *, y=None,* **)**

label： 类型为字符串，即标题文本

fontdict： 类型为字典，控制文本的字体属性

loc： 取值范围为{'left', 'center', 'right’}

子图标题

ax.set_title(*str,**kwargs***)**

matplotlib.pyplot.title(*str, **kwargs***)**

总标题

matplotlib.pyplot.suptitle(*str, **kwargs***)**

子图添加名称可以使用title()和set_title()两个函数，有子图时添加总标题要使用suptitle()

关键信息之图例

表示不同图形的文本标签图案叫图例，一般使用legend函数进行标注

legend函数

matplotlib.pyplot.legend(**args, **kwargs* **)**

plt.legend( )

Keyword： 图形的标注名称

图例还可以通过在函数中添加label参数完成，此时使用legend函数不需要添加keyword参数，如果添加参数会直接覆盖前面的标记。

关键信息之轴信息

坐标轴一般包含三种信息

X/Y轴命名

matplotlib.pyplot.xlabel(**args, **kwargs* **)** **plt.xlabel(***'str'***)**

matplotlib.pyplot.ylabel(**args, **kwargs* **)** **plt.ylabel(***'str'***)**

X/Y轴范围

matplotlib.pyplot.xlim(**args, **kwargs* **)** **plt.xlim(***num1, num2***)**

matplotlib.pyplot.ylim(**args, **kwargs* **)** **plt.ylim(***num1, num2***)**

X/Y轴刻

matplotlib.pyplot. xticks(*ticks, labels, *args, **kwargs* **)**

matplotlib.pyplot. yticks(*ticks, labels, *args, **kwargs* **)**

刻度的使用可以均匀分布也可以指定

网格、文本及子图函数

添加网格

matplotlib.pyplot.grid()

给图片指定位置添加文本信息

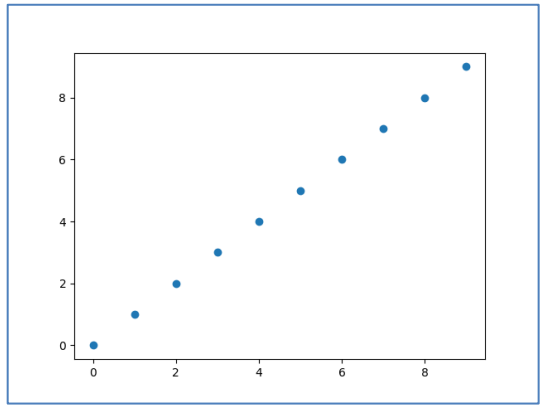
matplotlib.pyplot.text(*x, y, str,*args, **kwargs***)**

功能	整图函数	子图函数
创建画布	**plt.figure(** **args, **kwargs* **)**	**plt. subplot(** **args, **kwargs* **)**
设置标题	**plt.title(** **args, **kwargs* **)**	**ax.set_title(** **args, **kwargs* **)**
绘图函数	**plt.function(** **args, **kwargs* **)**	**ax.function(** **args, **kwargs* **)**
设置X/Y轴名称	**plt.xlabel(** **args, **kwargs* **)**	**ax.set_xlabel(** **args, **kwargs* **)**
设置X/Y轴刻度	**plt.xticks(** **args, **kwargs* **)**	**ax.set_xticks(** **args, **kwargs* **)**

其他图形绘制

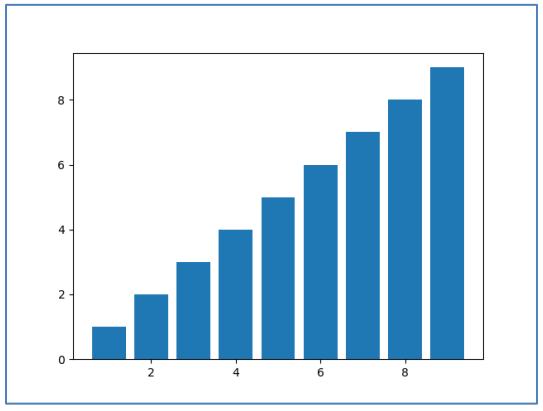
散点图

matplotlib.pyplot.scatter(*x*, *y*, *s=None, c=None, marker=None, cmap=None, norm=None, vmin=None, vmax=None, **kwargs***)**



柱状图

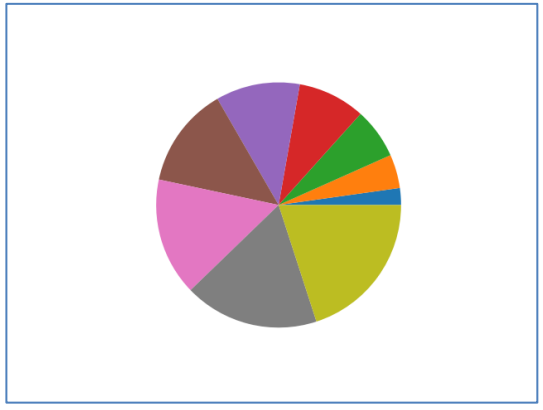
matplotlib.pyplot.bar(*x*, *height, width=0.8, bottom=None, *, align='center', data=None, **kwargs***)**



饼状图

matplotlib.pyplot.pie(*x*, *explode=None, labels=None, colors=None, autopct=None, pctdistance=0.6, ,**kwargs***)**

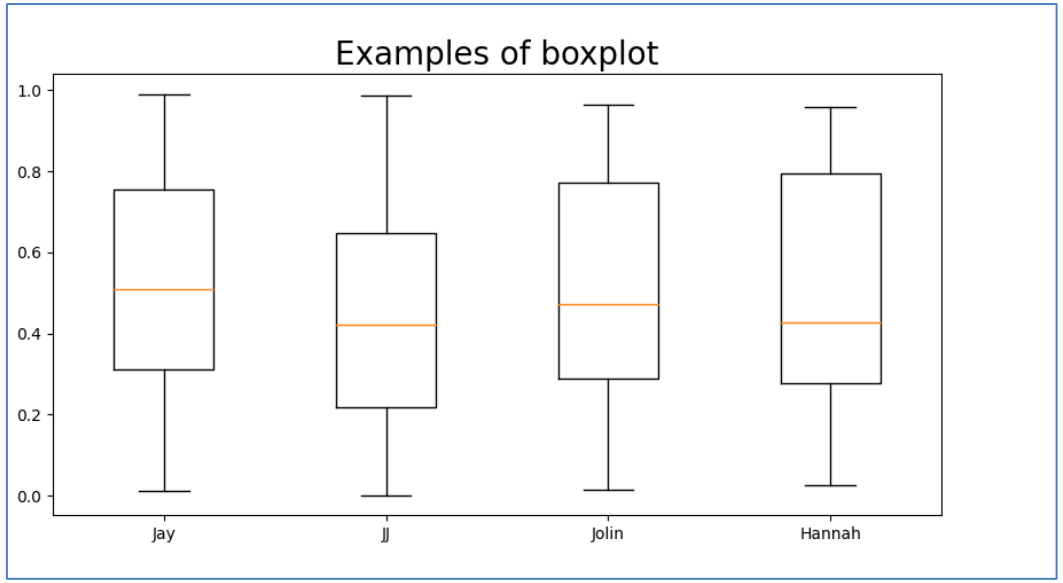
***饼状图输入值不得为负**



其他图形绘制

箱线图函数

matplotlib.pyplot.boxplot(*x*, **args*, ***kwargs*)

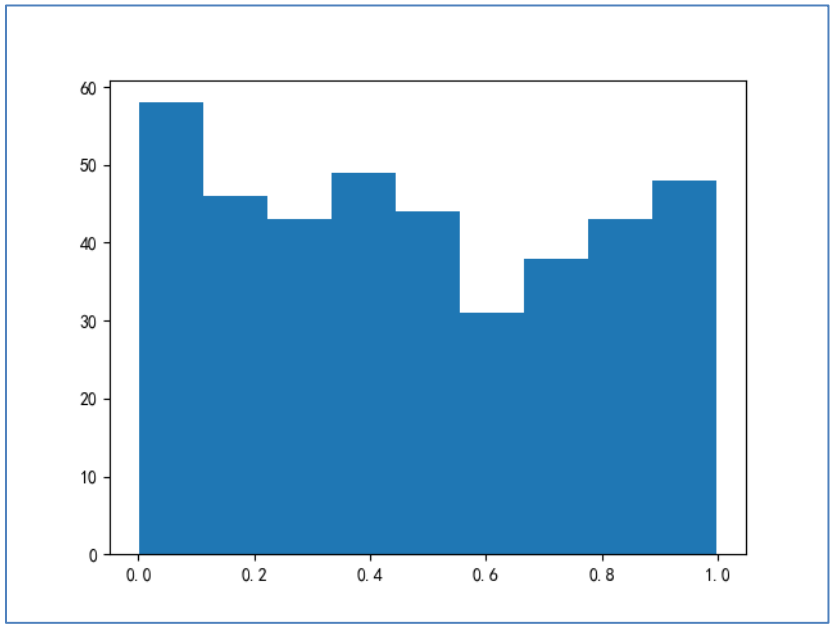


直方图函数

matplotlib.pyplot.hist(*x*,*bins=10*,*normed=False*,* *args*, ***kwargs*)

bins： 直方图的柱数，可选项，默认为10

normed： 是否将得到的直方图向量归一化。默认为0



3D绘图了解

3D绘图与2D绘图区别并不大，生成具有三维格式的对象Axes3D，有两种方法供同学们选择

子图画布3D化直接通过参数指定即可

fig.add_subplot(*224, projection='3d'***)**

无子图图片通过创建三维坐标轴对象Axes3D实现画布3D化，有两个办法可以做到

- 使用关键字生成3D画图

  fig.add_subplot(*111， projection='3d'***)**

- 定义图像和三维格式坐标轴，此时需要导入函数

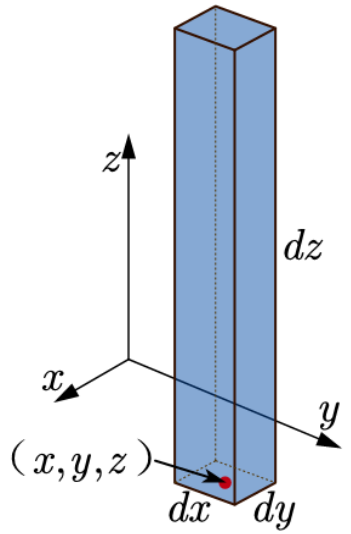
  from matplotlib import pyplot as plt

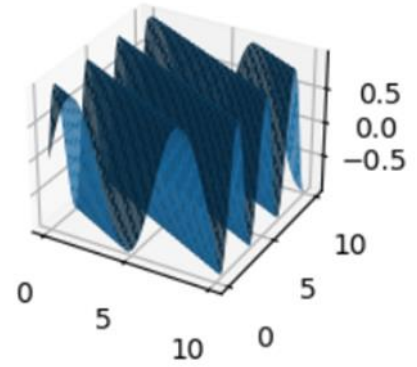
  from mpl_toolkits.mplot3d import Axes3D

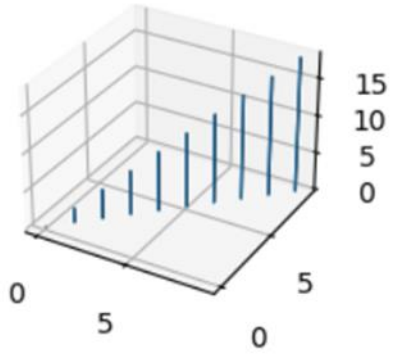
3D图像的参数调节

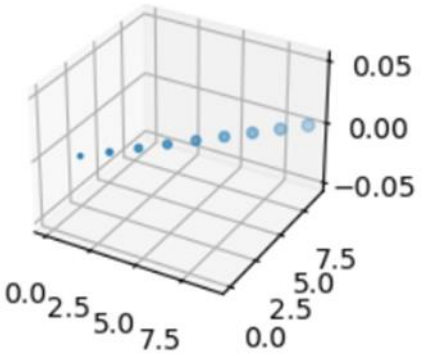
3D图像调节原理与2D图像类似

功能	函数
添加表标题	与2D图一致
设置坐标轴范围	ax.set_xlim3d(min,max)
设置坐标轴名称	ax.set_xlabel('str')
图中添加文本	ax.text(x,y,z,str,*args,**kwargs)
3D折线图	plt.plot3D(*X,Y,Z,*args,**kwargs*)
3D柱状图	plt.bar3d(*x,y,z,dx,dy,dz,*args,**kwargs*)
3D散点图	plt.scatter3D(*x,y,z,*args,**kwargs*)
3D曲面图	ax.plot_surface(*x,y,z*)
图例	暂时无法添加



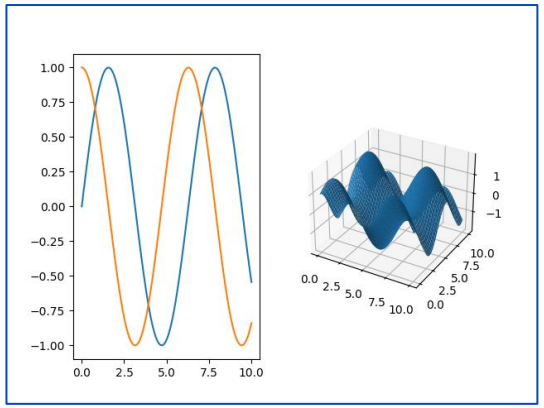


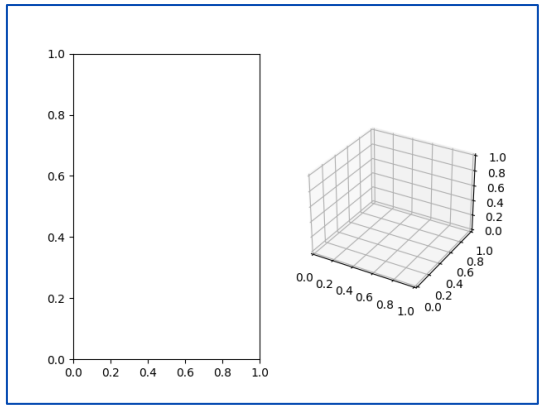


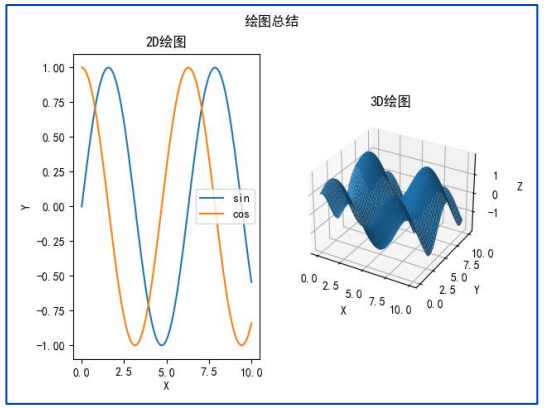


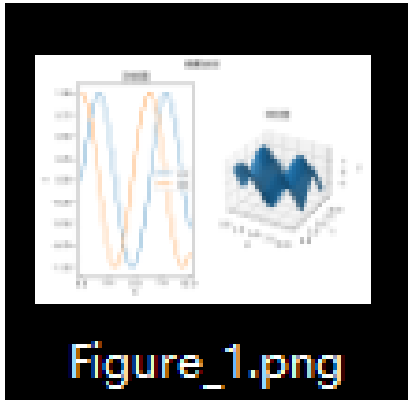
matplotlib小结

创建画布	绘制图形	图片美化	保存显示
• 创建完整画布 • 创建子图画布 • 创建3D画布	• 2D图形 • 3D图形	• 添加关键信息 • 子图间距调节	• 保存图片 • 显示图片

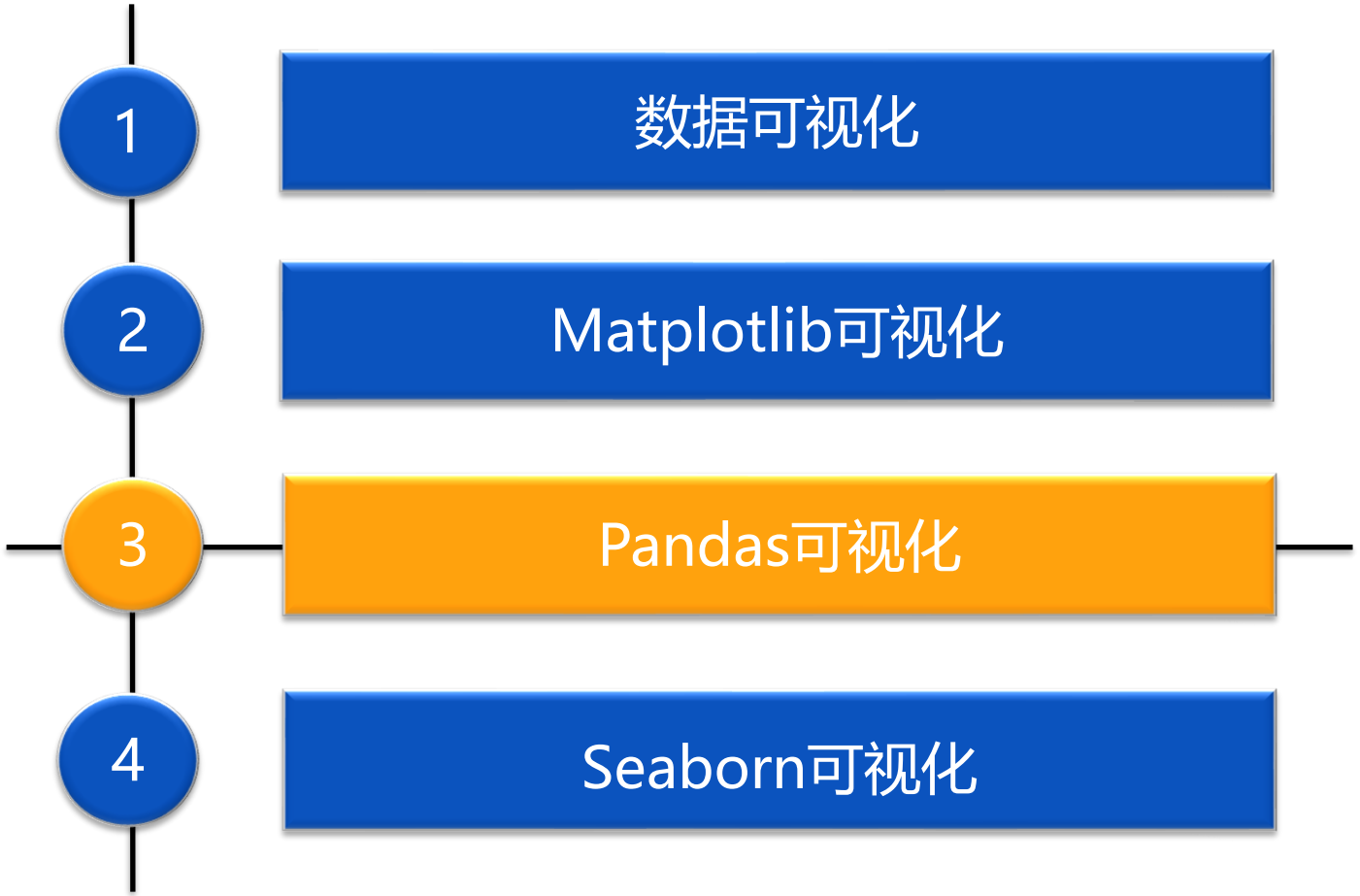








目录

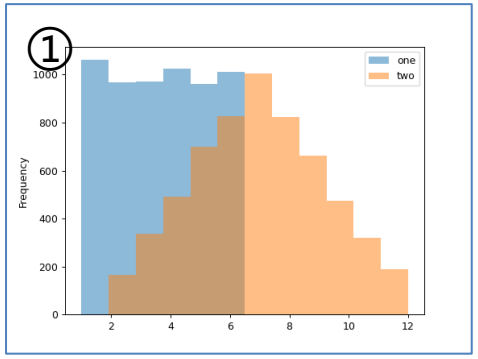


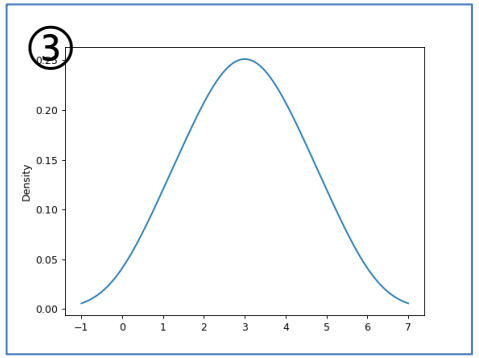
Pandas绘图模块

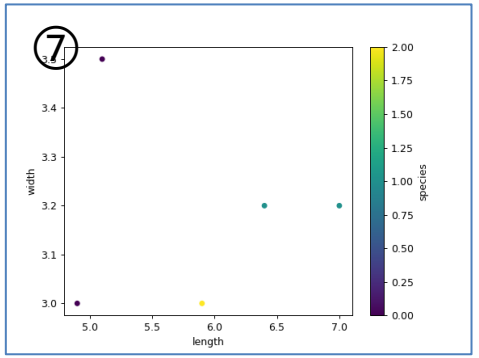
pands自身又很多内建方法可以简化DataFrame和Series对象生成可视化的过程。

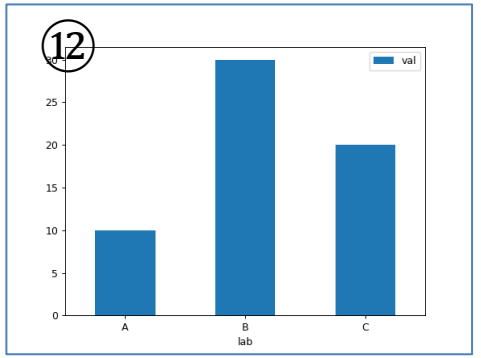
1. pandas.DataFrame.plot.hist
2. pandas.DataFrame.hist
3. pandas.DataFrame.plot.kde
4. pandas.DataFrame.plot.density
5. pandas.DataFrame.plot.line
6. pandas.DataFrame.plot.pie
7. pandas.DataFrame.plot.scatter
8. pandas.DataFrame.boxplot
9. pandas.DataFrame.plot.box
10. pandas.DataFrame.attrs
11. pandas.DataFrame.plot.area
12. pandas.DataFrame.plot.bar
13. pandas.DataFrame.plot.barh
14. pandas.DataFrame.plot.hexbin

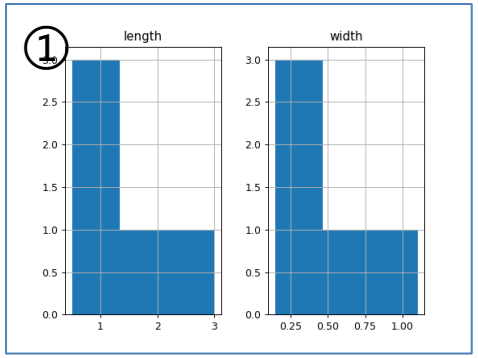
图片示例

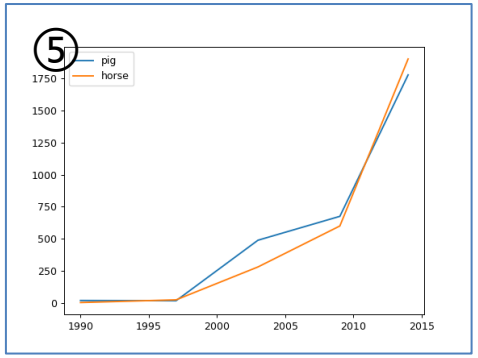

①
one
two
Frequency

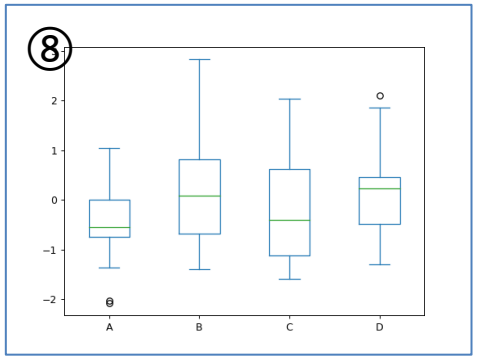

③
Density

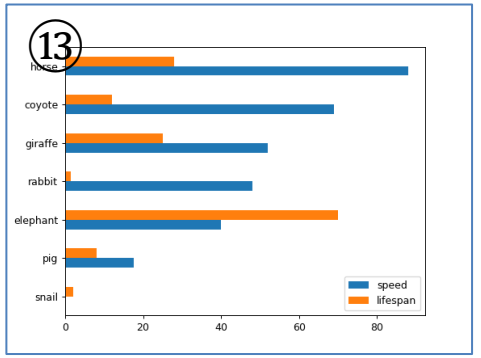

⑦
width
length
species

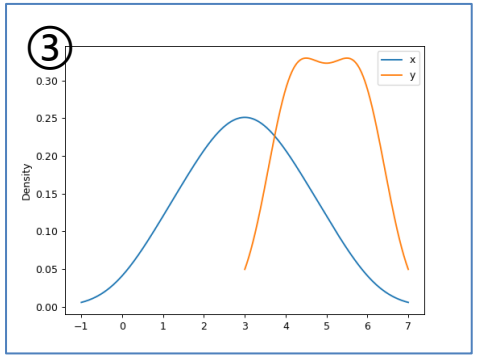

⑫
val
A
B
C
lab

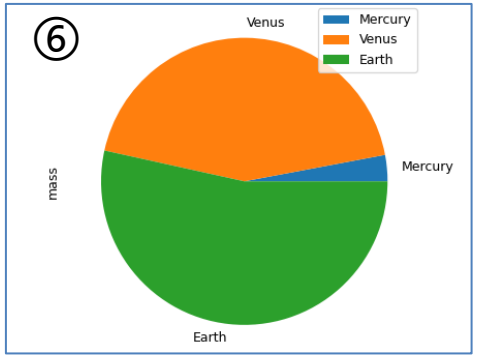

①
length
width

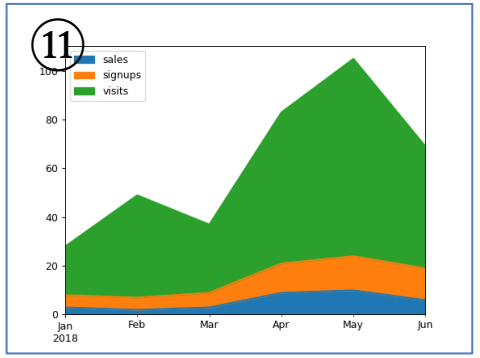

⑤
pig
horse

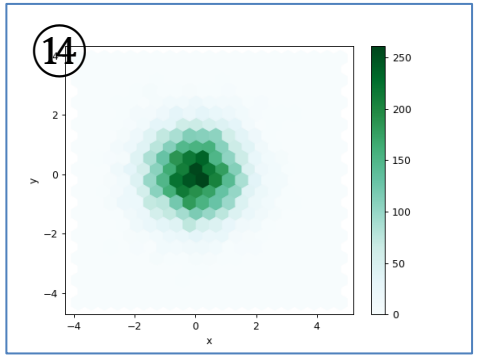

⑧
A
B
C
D


⑬
horse
coyote
giraffe
rabbit
elephant
pig
snail
speed
lifespan


③
x
y
Density


⑥
Venus
Mercury
Venus
Earth
mass
Mercury
Earth


⑪
sales
signups
visits
Jan
2018
Feb
Mar
Apr
May
Jun

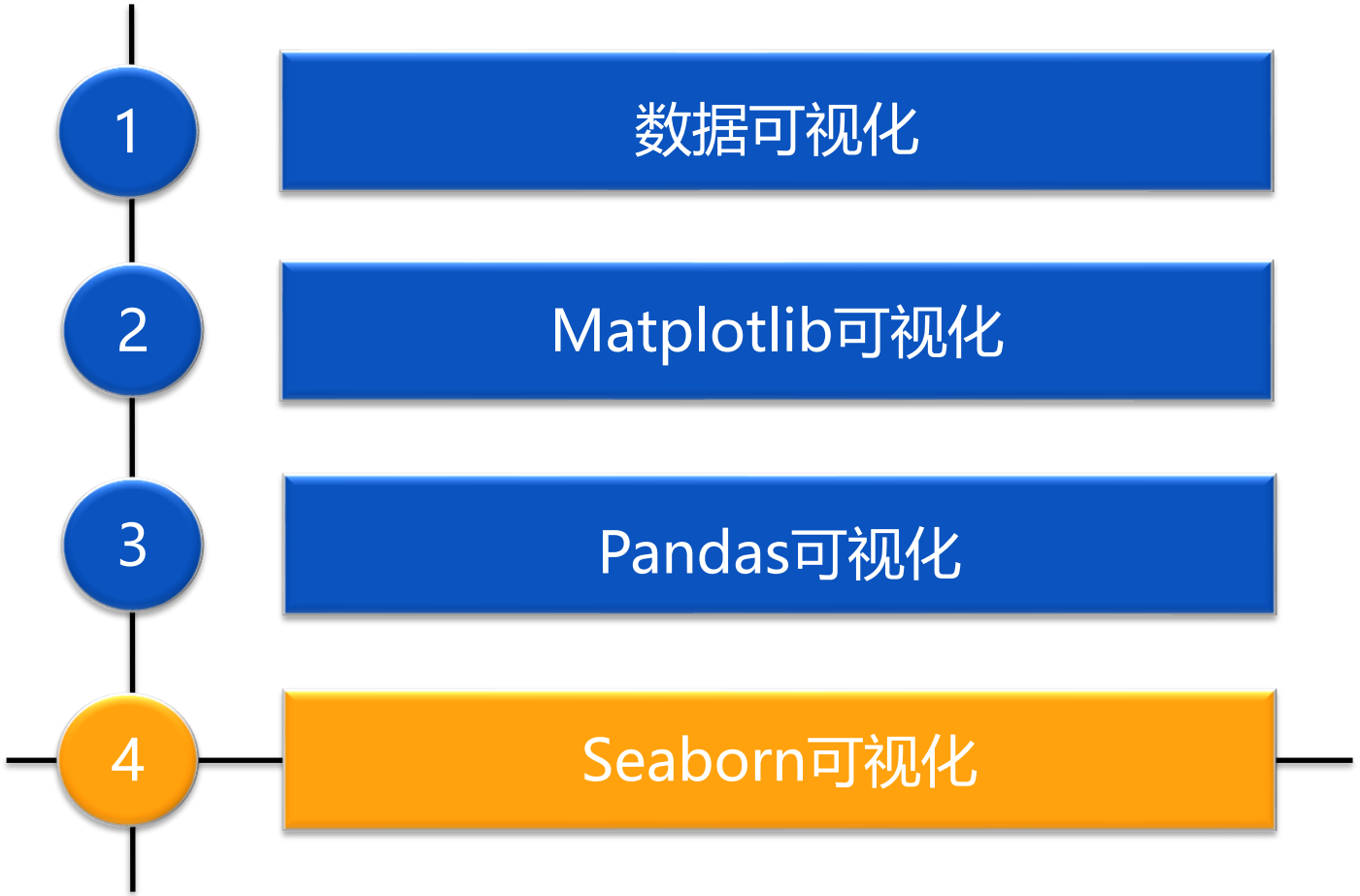

⑭
y
x

常见参数调节

参数	描述
label	图例标签
ax	绘图所用的matplotlib子图对象，如果没有传值，使用当前活动的matplotlib子图
style	传给matplotlib的样式字符串，比如 ‘bo--’
alpha	图片不透明度，（0到1）
logy	在y轴上使用对数缩放
use_index	使用对象索引刻度标签
rot	刻度标签旋转（0到360）
xticks	用于x轴刻度
yticks	用于y轴刻度
xlim	x轴范围
ylim	y轴范围
grid	展示轴网格（默认打开）

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.plot.html

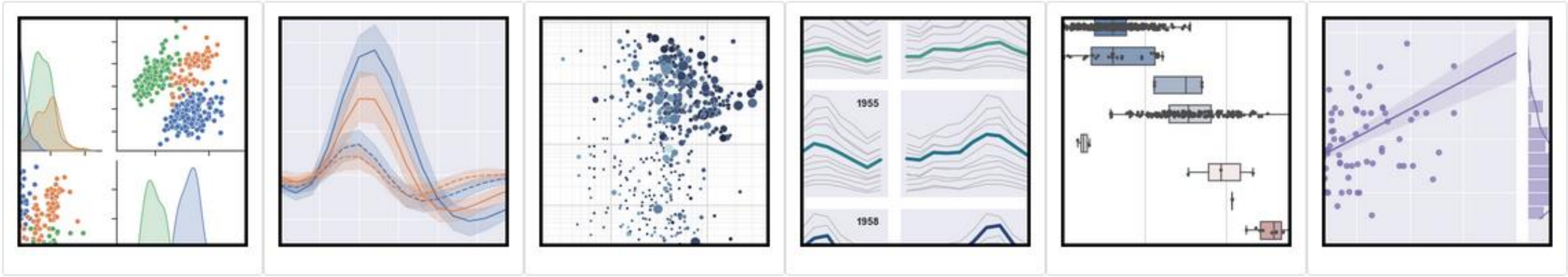
目录



Seaborn

Seaborn是基于matplotlib的图形可视化python包。它提供了一种高度交互式界面，便于用户能够做出各种有吸引力的统计图表。

Seaborn是在matplotlib的基础上进行了更高级的API封装，从而使得作图更加容易，在大多数情况下使用Seaborn能做出很具有吸引力的图，而使用matplotlib就能制作具有更多特色的图。**应该把Seaborn视为matplotlib的补充，而不是替代物**。同时它能高度兼容numpy与pandas数据结构以及scipy与statsmodels等统计模式。



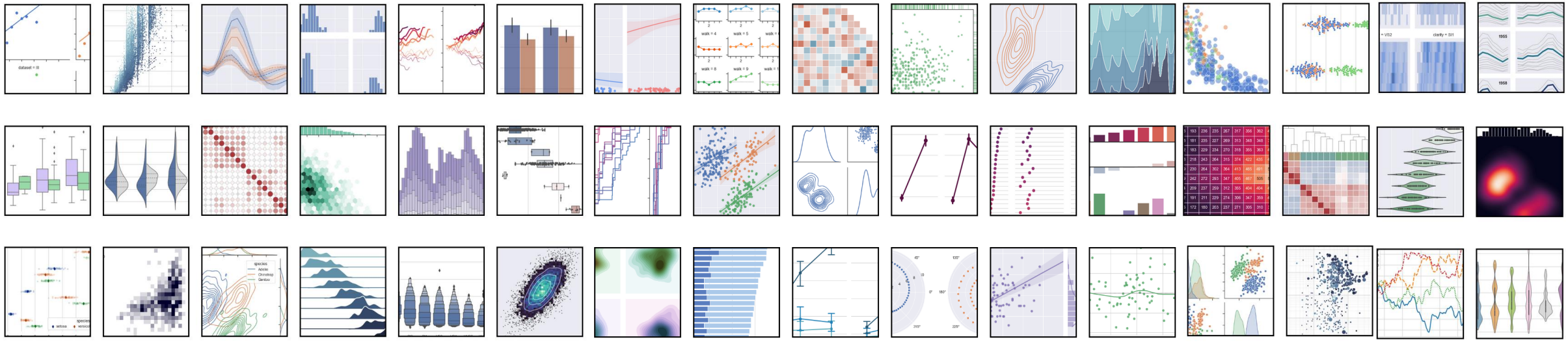
Seaborn使用预备知识

Seaborn通过在绘图函数中指定自变量、因变量和分类数据进行绘图。产生的图形、图形的构成、图形的样式等修饰信息都可以在函数中指定。由于Seaborn是基于Matplotlib开发的，所以在绘图中，二者的语法可以混合使用。

对于函数选择有“困难症”的同学，推荐查看官网示例：

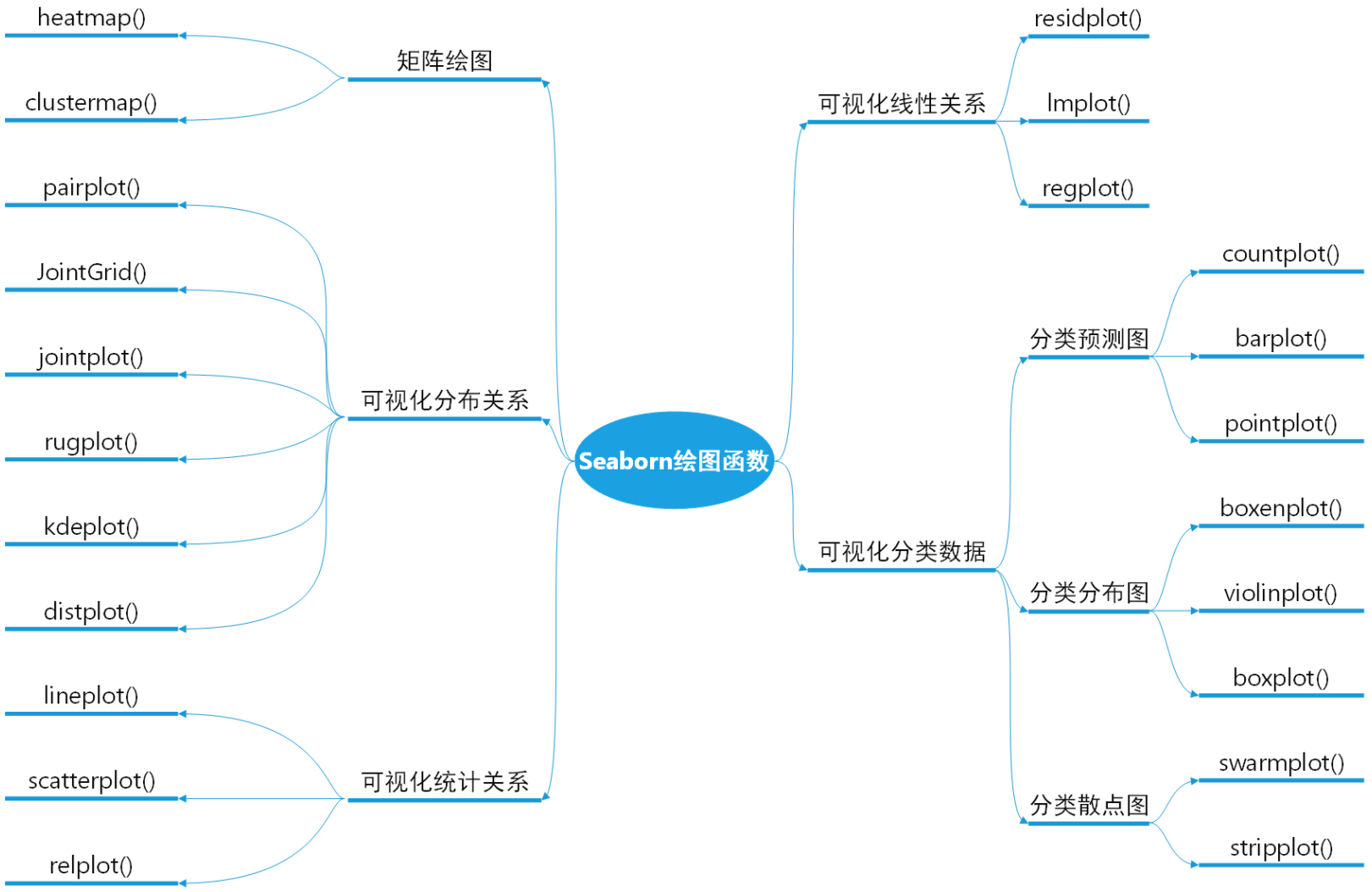
http://seaborn.pydata.org/examples/index.html

Example gallery



Seaborn 常见函数分类

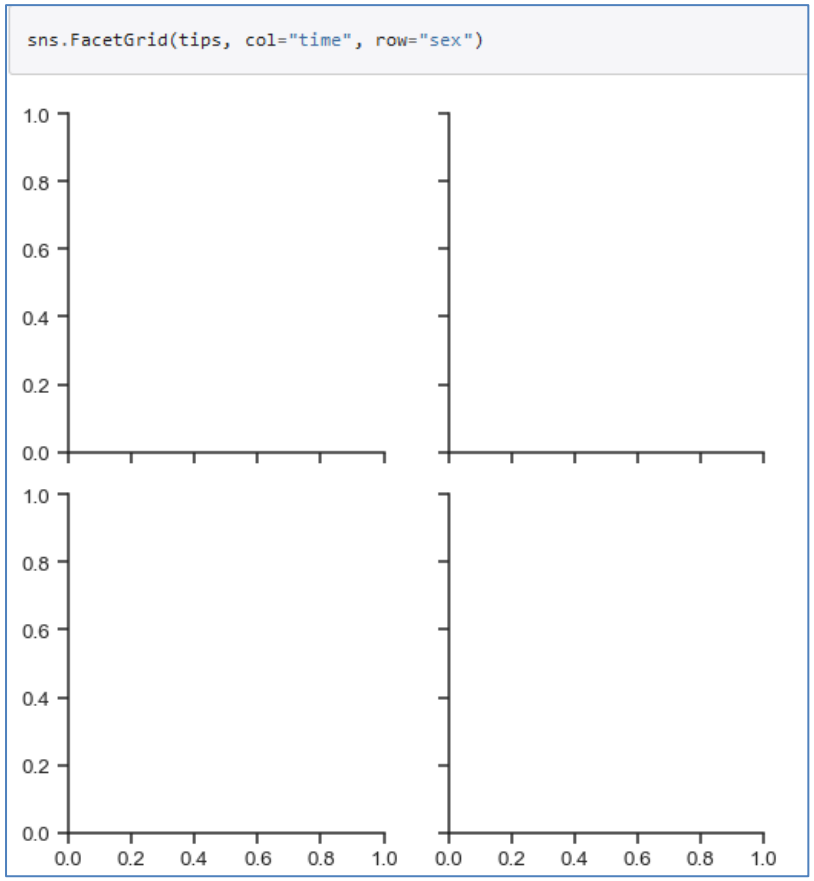
Seaborn的部分函数可以通过指定参数达到相同的效果，在使用函数时需要注重的是函数实现的**功能**，而不是单个函数本身。



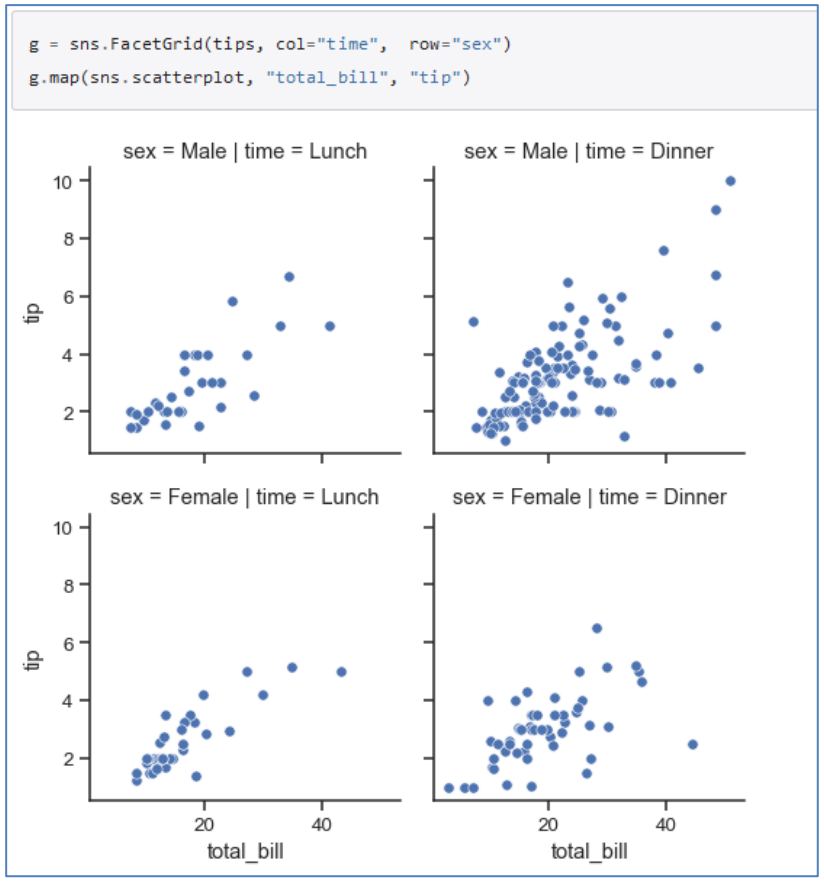
Seaborn 绘制子图

Seaborn的子图绘制与Matplotlib不同，Seaborn主要根据分类数据进行子图生成。通过指定行列数的安排规定子图布局。在生成子图布局后借助map、或者map系列函数讲其他数据的绘图映射到子图中。

```
sns.FacetGrid(tips, col="time", row="sex")
```

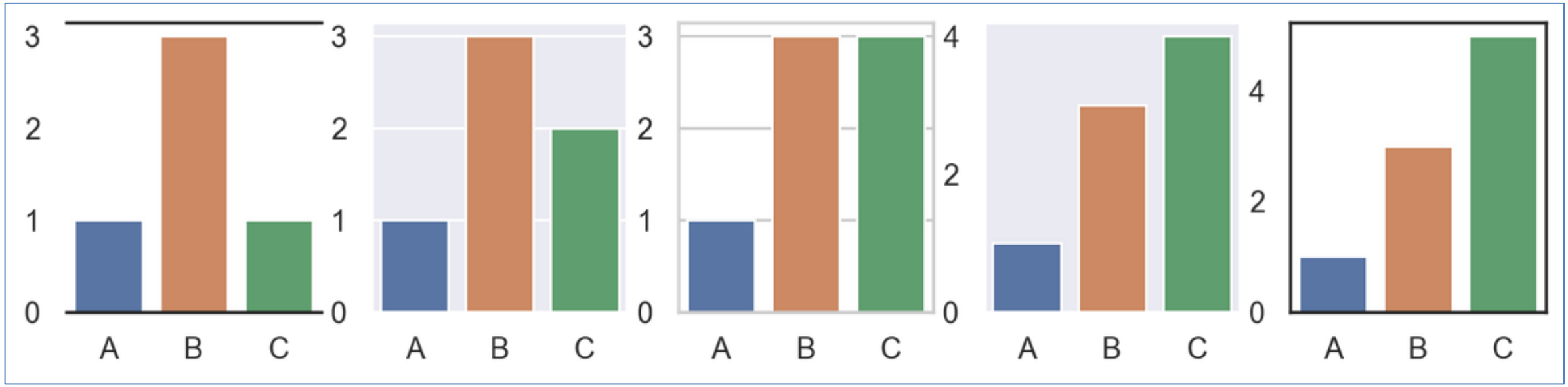



```
g = sns.FacetGrid(tips, col="time",  row="sex")
g.map(sns.scatterplot, "total_bill", "tip")
```



Seaborn的图形美化

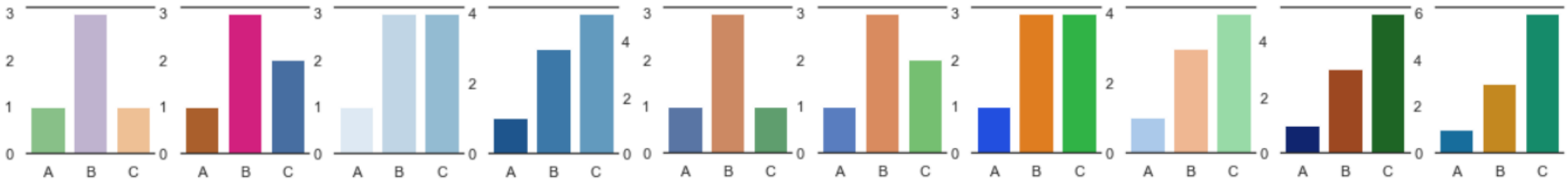
matplotlib的绘图风格朴实无华，Seaborn内置多种绘图风格可供选择。每种绘图风格又可以通过修改具体参数改变原始的绘图。在sns.set_style函数中可以修改图形的背景、网格线、边框、坐标轴、字体等等信息。



Seaborn调色板设置

Seaborn色彩调节可以通过palette参数实现，该参数可以接受176种预定参数，包括

- Seaborn 自定义的主题色板
- matplotlib的色板
- husl或者hls的色彩分类
- cubehelix（使用函数生成的）色板
- light、dark主题的色板
- matplotlib接受任意格式的颜色序列



Thank you!