

In []:

```
sns.catplot(x = 'time', y = 'index1', col = 's0', data = ccss,  
            kind = "box")
```

In []:

```
sns.catplot(x = 'time', y = 'index1', col = 's0', data = ccss,  
            kind = "boxen")
```

In []:

```
sns.catplot(x = 'time', y = 'index1', col = 's0', data = ccss,  
            kind = "violin")
```

In []:

```
sns.catplot(x = 'time', y = 'index1', col = 's0', hue = 's2',  
            data = ccss, kind = "bar")
```

In []:

```
sns.catplot(x = 's0', y = 'index1', col = 'time', hue = 's2',  
            data = ccss, kind="bar", col_wrap = 2, legend_out = False)
```

6.5 实战练习

尝试在同一张图形中给出分受教育程度的年龄均数和总信心指数均数的信息，要求：

年龄均数用线图表示。

总信心指数用带CI的条图表示。

对颜色、数轴等作必要的设定，使得图形显示效果达到最佳。

比较seaborn中的catplot()命令和pairplot()命令在使用目的、参数设定等各方面的异同，自行思考这两个命令应当适用于哪些数据可视化的需求。

7 子图与图形网格

7.1 图形叠加/图中图

在matplotlib中，所有的绘图操作实际上都是以Axes对象为独立的绘图区域进行，这里称为子图。

在一个Figure对象中可以有多个子图，这些子图对象可以叠加存在，从而形成图中图的效果。

matplotlib.figure.Figure.add_axes(# 在已有Figure对象中按照指定范围添加子图

rect : 代表插入子图对象大小的序列。

[left, bottom, width, height]

projection : 子图使用的坐标体系。

['aitoff' | 'hammer' | 'lambert' |

'mollweide' | 'polar' | 'rectilinear']

polar : boolean, 为True时等价于projection = 'polar'

) 返回: Axes对象

In []:

```
sns.pointplot(x = ccss.time, y = ccss.index1, capsize = .1)
```

In []:

```
sns.pointplot(x = ccss.time, y = ccss.index1, capsize = .1)
fig = plt.gcf() # gcf即'get current figure'的缩写
fig.add_axes([0.15,0.2,0.25,0.2])
sns.distplot(ccss.index1, kde = False)
```

In []:

```
sns.pointplot(x = ccss.time, y = ccss.index1, capsize = .1)
plt.xlabel("月份")
plt.ylabel("总信心指数均值")

fig = plt.gcf() # gcf即'get current figure'的缩写
fig.add_axes([0.14,0.18,0.25,0.2])
sns.distplot(ccss.index1, kde = False, color = 'c')
plt.xlabel('总信心指数')
plt.xticks([])
plt.yticks([])
```

7.2 分区绘制多个子图

可以将Figure对象分为多个区域, 每个区域分别放置一个Axes对象进行绘图。

7.2.1 subplot命令

subplot命令可以用来直接指定具体的子图位置。

matplotlib.pyplot.subplot(

nrows / ncols = 1 : 图组的行/列数。

整个Figure对象区域被划分为nrows行 * ncols列。

然后按照从左到右、从上到下的顺序对每个区域进行编号, 左上区域的编号为1。

index : 指定所创建Axes对象所在的区域。

)

如果numRows、numCols和Index三个参数都小于10, 则可以把它们缩写成一个整数。

subplot(323)和subplot(3,2,3)的含义相同。

如果新创建的子图和之前创建的子图区域有重叠的部分, 则之前的子图将被删除。

In []:

```
plt.subplot(221) # 第一行的左图
plt.subplot(222) # 第一行的右图
plt.subplot(212) # 第二整行
plt.plot([1,2,2,1])
plt.show()
```

7.2.2 切换使用的子图对象

`matplotlib.pyplot.sca(ax)` : Set the current Axes instance to `ax`

子图对象所在的图表对象也会被自动设定为当前图表对象。

In []:

```
plt.subplot(221) # 第一行的左图
ax1 = plt.subplot(222) # 第一行的右图
plt.subplot(212) # 第二整行
plt.sca(ax1)
plt.plot([1,2,2,1])
plt.show()
```

7.2.3 subplots命令

`subplot`命令需要分别指定每个子图，如果希望批量快速生成子图的网格框架，则可以使用`subplots`命令。

`matplotlib.pyplot.subplots()`

`nrows / ncols = 1` : 图形网格的行/列数

`sharex, sharey = False` : 在图组中是否共用行/列坐标轴

`True` or `'all'` : 对应的单元格都将共用行/列坐标轴

`False` or `'none'` : 各单元格独立设定行/列坐标轴

`'row'` : 同一行的单元格将共用行/列坐标轴

`'col'` : 同一列的单元格将共用行/列坐标轴

`squeeze = True` : 是否尽量简化返回的Axes对象

`False`时即使只有一个单元格，也返回二维数组

`subplot_kw` : dict, 未来调用`add_subplot()`建立子图时需要传送的参数

)返回值:

`fig` : `matplotlib.figure.Figure`对象

`ax` : Axes对象/Axes数组对象

In []:

```
fig, axes = plt.subplots(2, 3)
```

In []:

```
fig, ax0 = plt.subplots()
sns.distplot(ccss.index1, ax = ax0) # 此处ax参数也可以不指定
ax0.set_title('Sample plot')
```

In []:

```
f, (ax1, ax2) = plt.subplots(1, 2, sharey = True)
ax1.hist(ccss.index1)
ax1.set_title('Sharing Y axis')
ax2.plot([0,200,300])
```

In []:

```
# 与seaborn联合使用
f, (ax1, ax2) = plt.subplots(1, 2)
sns.boxplot(y = ccss.index1, ax = ax1)
ax1.set_title('总信心指数')
sns.boxplot(y = ccss.s3, ax = ax2)
ax2.set_title('年龄')
```

In []:

```
# 使用数组方式指定子图
fig, axes = plt.subplots(2, 2)
sns.boxplot(y = ccss.index1, ax = axes[0, 0])
sns.boxplot(y = ccss.s3, ax = axes[1, 1])
```

In []:

```
# 多Figure对象, 多子图对象的联合应用
import numpy as np
import matplotlib.pyplot as plt

plt.figure(1) # 创建图表1
plt.figure(2) # 创建图表2
ax1 = plt.subplot(211) # 在图表2中创建子图1
ax2 = plt.subplot(212) # 在图表2中创建子图2

x = np.linspace(0, 3, 100)
for i in range(5):
    plt.figure(1) # 选择图表1
    plt.plot(x, np.exp(i*x/3))
    plt.sca(ax1) # 选择图表2的子图1
    plt.plot(x, np.sin(i*x))
    plt.sca(ax2) # 选择图表2的子图2
    plt.plot(x, np.cos(i*x))

plt.show()
```

7.2.4 调整子图间距

matplotlib.pyplot.subplots_adjust(

```

left  = 0.125 # the left side of the subplots of the figure
right = 0.9   # the right side of the subplots of the figure
bottom = 0.1  # the bottom of the subplots of the figure
top = 0.9     # the top of the subplots of the figure
wspace = 0.2  # the amount of width reserved between subplots,
               # as a fraction of the average axis width
hspace = 0.2  # the amount of height reserved between subplots,
               # as a fraction of the average axis height

)

```

In []:

```

# 使用数组方式指定子图
fig, axes = plt.subplots(1, 2, sharey = True)

sns.boxplot(y = ccss.index1a, ax = axes[0])
sns.boxplot(y = ccss.index1b, ax = axes[1])

```

In []:

```

# 使用数组方式指定子图
fig, axes = plt.subplots(1, 2, sharey = True)

# 进一步设定图形格式细节
plt.subplots_adjust(top = 0.7, wspace = 0.1)
sns.boxplot(y = ccss.index1a, ax = axes[0])
sns.boxplot(y = ccss.index1b, ax = axes[1])
plt.annotate('现状指数和预期指数的对比', xy = (-1.3, 220),
             annotation_clip = False, size = 20)

```

7.3 不等距复杂网格的设定

subplot/subplots命令主要用于指定等行/列数量的网格，对于不等行列的网格则操作较为繁琐，此时可以使用其他同类命令。

7.3.1 subplot2grid方法

当各行/列仍然是等量拆分，只是各行/列宽度不同时，用subplot2grid方法来设置网格很方便。

matplotlib.pyplot.subplot2grid(shape, loc, rowspan=1, colspan=1)

shape设定整体网格行列数

loc指定网格起点，而rowspan/colspan则用来指定网格跨度

In []:

```

plt.figure()
ax1 = plt.subplot2grid((3, 3), (0, 0), colspan = 3)
ax2 = plt.subplot2grid((3, 3), (1, 0), colspan = 2)

ax5 = plt.subplot2grid((3, 3), (2, 1))

```

In []:

```
plt.figure()
ax1 = plt.subplot2grid((3, 3), (0, 0), colspan = 3)
ax2 = plt.subplot2grid((3, 3), (1, 0), colspan = 2)
ax3 = plt.subplot2grid((3, 3), (1, 2), rowspan = 2)
ax4 = plt.subplot2grid((3, 3), (2, 0))
ax5 = plt.subplot2grid((3, 3), (2, 1))
```

7.3.2 GridSpec方法

可以说是指定各种不等距复杂网格的专用方法，操作最为便捷。

`matplotlib.gridspec.GridSpec(`

`nrows/ncols` : 网格的行/列数
`figure` : 指定网格将被放置的Figure对象
`left/bottom/right/top` : 网格四边所对应的图形长宽比例
`wspace/hspace` : 周边留空大小
`width_ratios/height_ratios` : 各行/列的大小比例

)

In []:

```
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec

f = plt.figure()

gs = gridspec.GridSpec(2, 2, width_ratios=[1, 2], height_ratios=[4, 1])
ax1 = plt.subplot(gs[0])
ax2 = plt.subplot(gs[1])
ax3 = plt.subplot(gs[2])
ax4 = plt.subplot(gs[3])

plt.show()
```

In []:

```
# 一个复杂的不等距网格案例
import matplotlib.pyplot as plt
from matplotlib.gridspec import GridSpec

# 自定义函数，隐藏数轴刻度
def make_ticklabels_invisible(fig):
    for i, ax in enumerate(fig.axes):
        ax.text(0.5, 0.5, "ax%d" % (i+1), va="center", ha="center")
        ax.tick_params(labelbottom=False, labelleft=False)

fig = plt.figure()

gs = GridSpec(3, 3)
ax1 = plt.subplot(gs[0, :])
# identical to ax1 = plt.subplot(gs.new_subplotspec((0, 0), colspan=3))
ax2 = plt.subplot(gs[1, :-1])
ax3 = plt.subplot(gs[1:, -1])
ax4 = plt.subplot(gs[-1, 0])
ax5 = plt.subplot(gs[-1, -2])

fig.suptitle("$GridSpec$")
make_ticklabels_invisible(fig)

plt.show()
```

7.4 实战练习

寻找手边做的比较好的图形叠加/图中图实例，并尝试在python中实现。

请尝试使用子图方式实现6.4节中的面板图形，并思考面板方式和子图方式各自的优缺点与适用环境是什么。

请自行设计一个面板，在其中同时呈现第5章中使用过的anscombe四个数据集各自拟合最佳回归模型的散点图，并使其显示效果达到最佳。

8 色彩搭配

8.1 色彩搭配的基本原则

8.2 如何自定义理想的色系

8.3 色板的指定方式

8.3.1 color_palette函数

除单独指定颜色外，seaborn的优势在于可以将颜色根据数据呈现的需求编组成恰当的色板/色环加以使用。

color_palette函数可以接受任何seaborn或者matplotlib颜色表中的颜色名称（除了jet），也可以接受任何有效的matplotlib形式的颜色列表（比如RGB元组，hex颜色代码，或者HTML颜色名称）。

seaborn.color_palette(