

seaborn可视化(1) ——相关

类型	函数	主要参数
散点图	scatter(x,y,data)	hue,style,size,sizes
	relplot(x,y,data,kind = 'scatter')	
线条图	lineplot(x,y,date)	ci,estimator,hue,style
	relplot(x,y,data,kind = 'line')	
多子图	relplot(x,y,data,col,row,height,kind = 'line')	col,row,col_wrap,height

散点图相关

- scatterplot() or relplot() : 散点图可视化相关变量

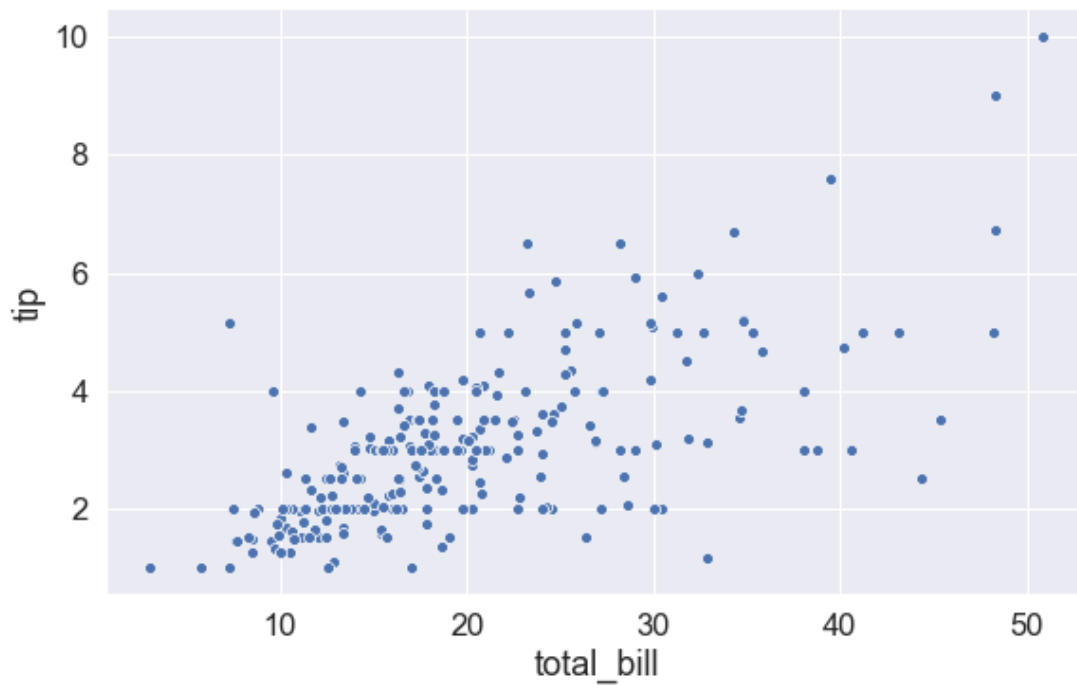
```
Signature:
sns.scatterplot(['x=None', 'y=None', 'hue=None', 'style=None', 'size=None', 'data=None',
'palette=None', 'hue_order=None', 'hue_norm=None', 'sizes=None', 'size_order=None',
'size_norm=None', 'markers=True', 'style_order=None', 'x_bins=None', 'y_bins=None',
'units=None', 'estimator=None', 'ci=95', 'n_boot=1000', "alpha='auto'", 'x_jitter=None',
'y_jitter=None', "legend='brief'", 'ax=None', '**kwargs'],
)

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="darkgrid")

#使用seaborn自带的小费数据集 tips
tips = sns.load_dataset('tips')
tips.head()
Out[42]:
   total_bill  tip  sex smoker  day  time  size
0      16.99  1.01 Female    No  Sun  Dinner     2
1      10.34  1.66   Male    No  Sun  Dinner     3
2      21.01  3.50   Male    No  Sun  Dinner     3
3      23.68  3.31   Male    No  Sun  Dinner     2
4      24.59  3.61 Female    No  Sun  Dinner     4
```

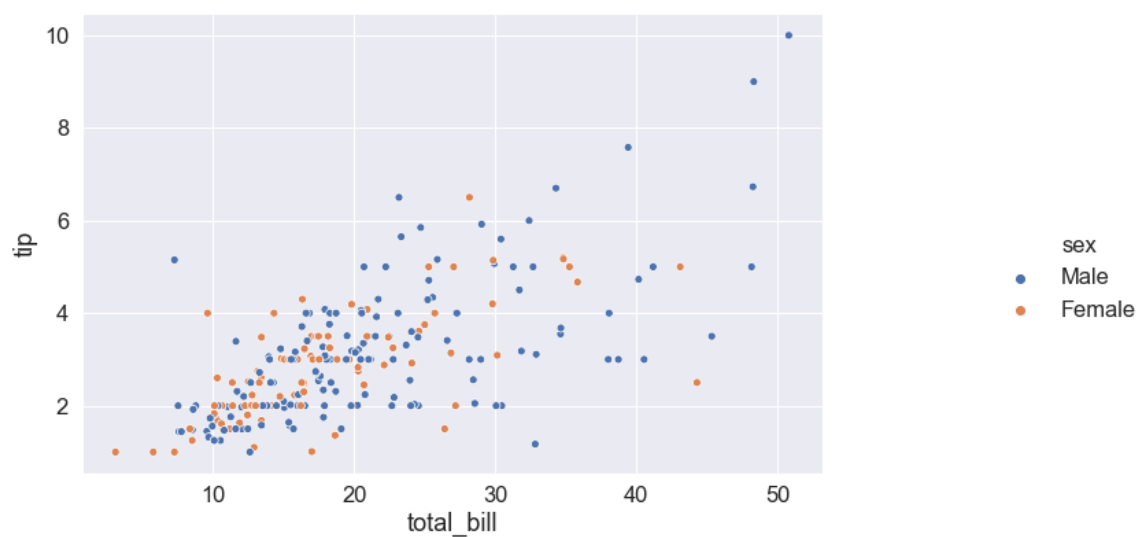
1:最基本的情况:

```
sns.scatterplot(x = 'total_bill',y = 'tip',data = tips)
or
sns.relplot(x="total_bill", y="tip", data=tips)
```



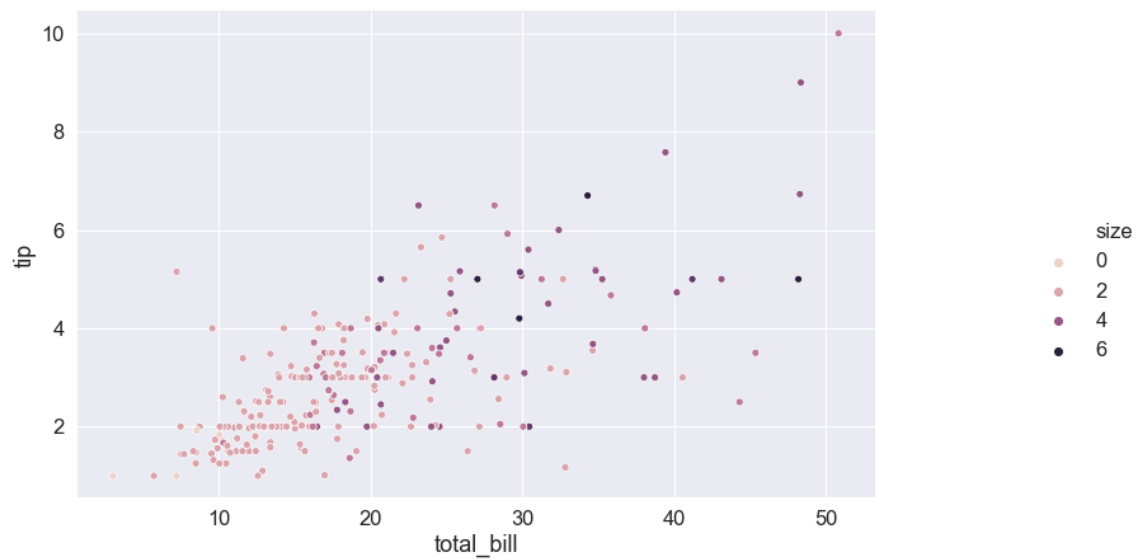
2. hue: 变量分组

```
sns.relplot(x = 'total_bill',y = 'tip',hue = 'sex',data = tips)
```



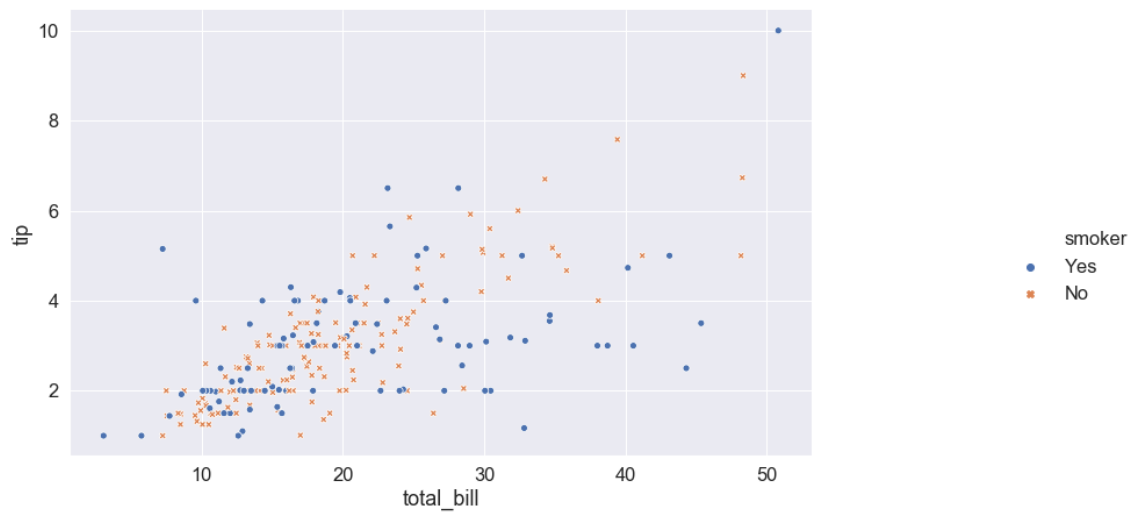
3. hue为数值型的情况

```
sns.relplot(x="total_bill", y="tip", hue="size", data=tips)
```



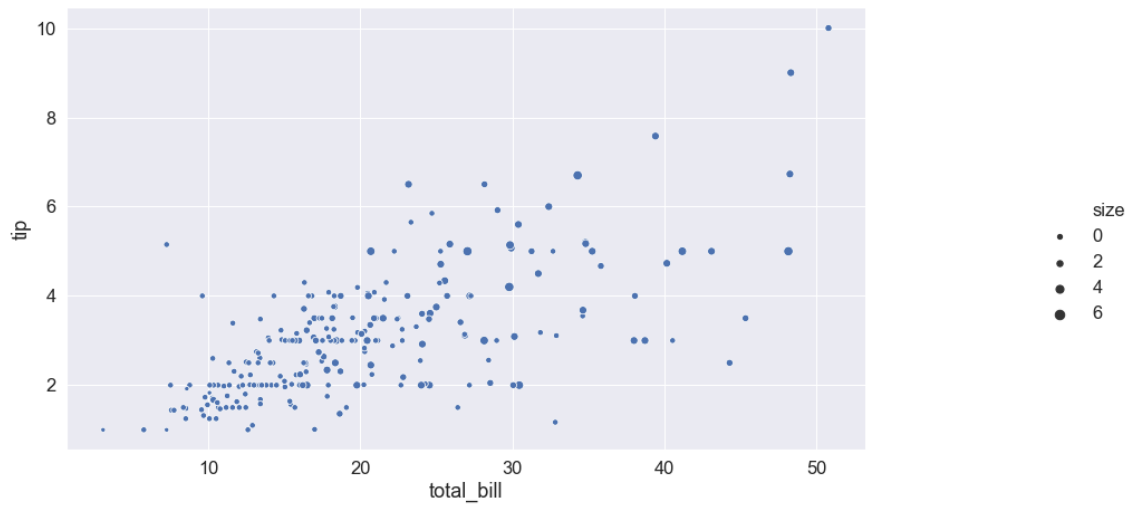
4. style: 为每个类使用不同的标记样式

```
sns.relplot(x = 'total_bill',y = 'tip',hue = 'smoker',style = 'smoker',data = tips)
```



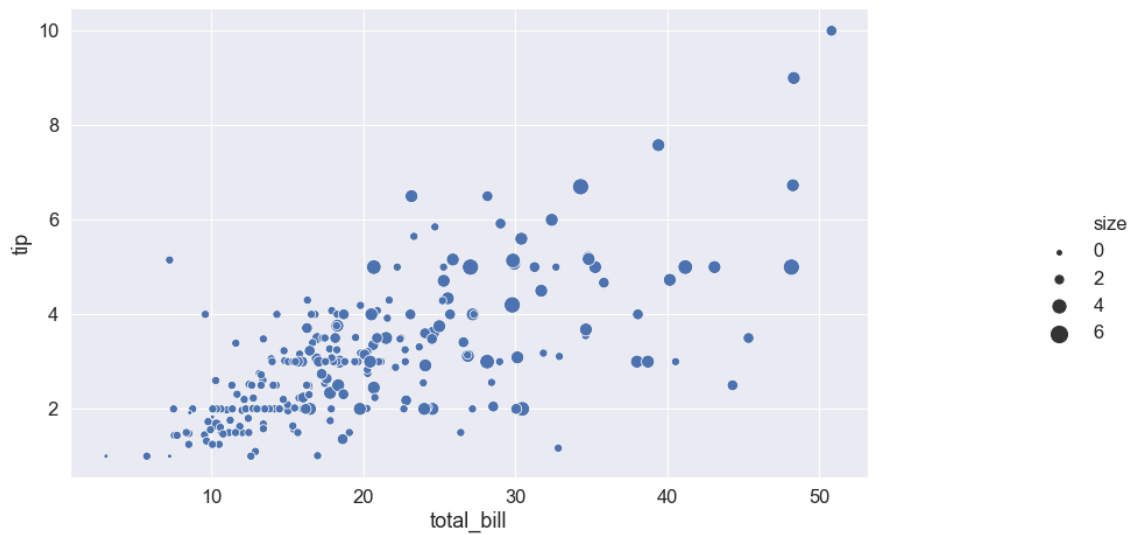
5.size : 改变点的大小

```
sns.relplot(x="total_bill", y="tip", size="size", data=tips)
```



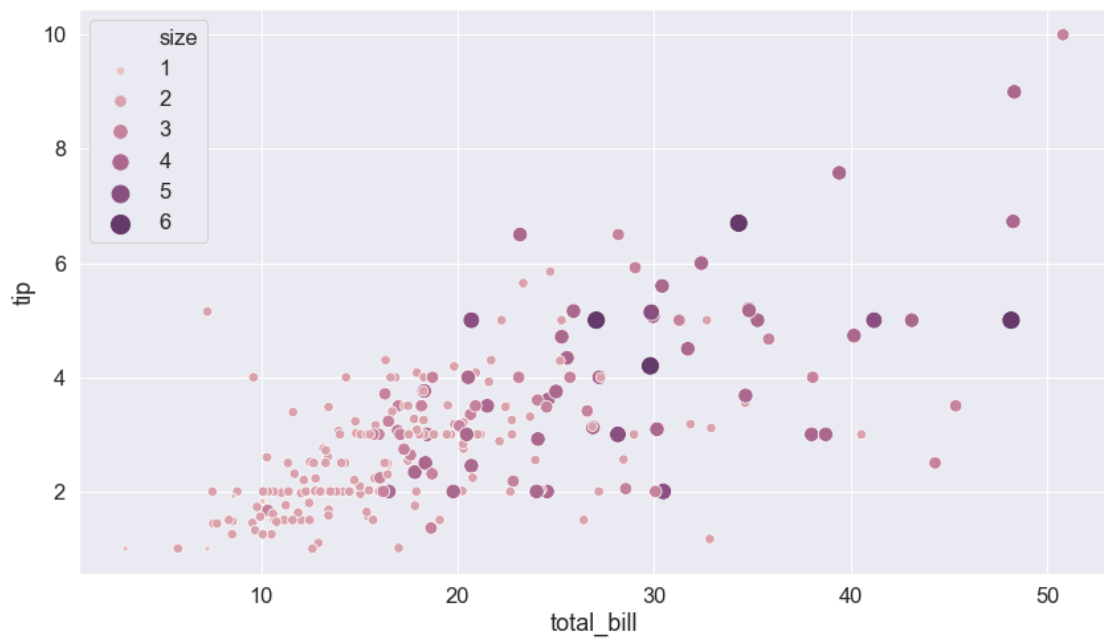
6.sizes : 设置点大小的范围

```
sns.relplot(x="total_bill", y="tip", size="size", sizes=(15, 200), data=tips)
```

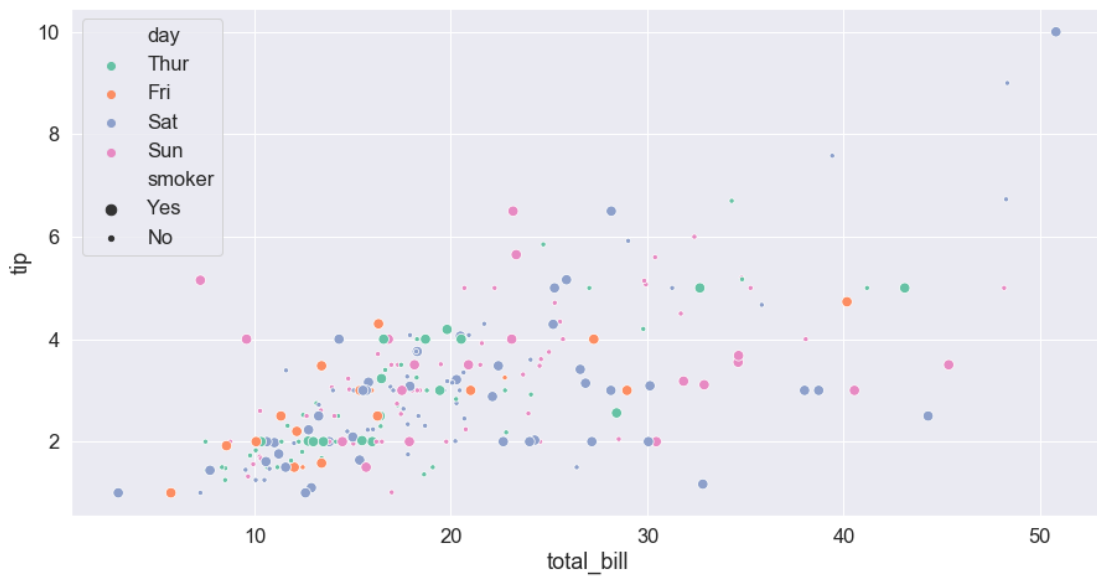


7.同时使用

```
cmap = sns.cubehelix_palette(dark=.3, light=.8, as_cmap=True)
sns.scatterplot(x="total_bill", y="tip",
...             hue="size", size="size",
...             sizes=(20, 200), palette=cmap,
...             legend="full", data=tips)
```



```
cmap = sns.cubehelix_palette(dark=.3, light=.8, as_cmap=True)
sns.scatterplot(x="total_bill", y="tip",
...             hue="day", size="smoker",
...             palette="Set2",
...             data=tips)
```



线条图相关

- `lineplot()` or `relplot(kind = 'line')` :线条图可视化相关变量

Signature:

```
sns.lineplot(['x=None', 'y=None', 'hue=None', 'size=None', 'style=None', 'data=None',  
'palette=None', 'hue_order=None', 'hue_norm=None', 'sizes=None', 'size_order=None',  
'size_norm=None', 'dashes=True', 'markers=None', 'style_order=None', 'units=None',  
"estimator='mean'", 'ci=95', 'n_boot=1000', 'sort=True', "err_style='band'",  
'err_kws=None', "legend='brief'", 'ax=None', '**kwargs'],)
```

```
fmri = sns.load_dataset("fmri")
```

```
fmri.head(10)
```

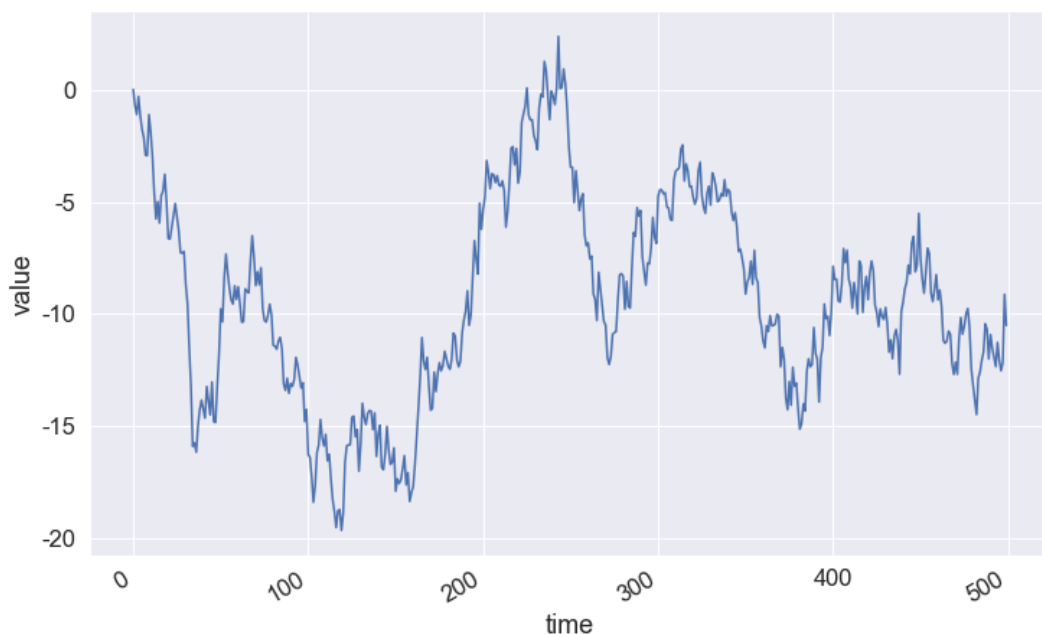
Out[84]:

	subject	timepoint	event	region	signal
0	s13	18	stim	parietal	-0.017552
1	s5	14	stim	parietal	-0.080883
2	s12	18	stim	parietal	-0.081033
3	s11	18	stim	parietal	-0.046134
4	s10	18	stim	parietal	-0.037970
5	s9	18	stim	parietal	-0.103513
6	s8	18	stim	parietal	-0.064408
7	s7	18	stim	parietal	-0.060526
8	s6	18	stim	parietal	-0.007029
9	s5	18	stim	parietal	-0.040557

1.默认情况

#模拟数据

```
df = pd.DataFrame(dict(time=np.arange(500),  
                        value=np.random.randn(500).cumsum()))  
g = sns.relplot(x="time", y="value", kind="line", data=df)
```



2. 聚合和表示不确定性

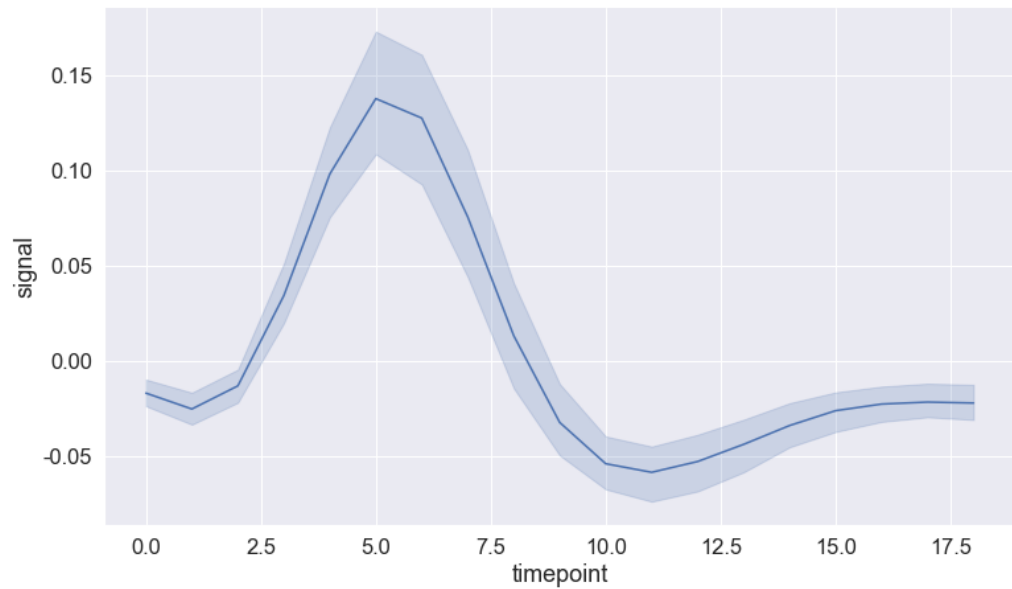
```
"""
```

更复杂的数据集将对相同的x变量值有多个测量值。

seaborn中的默认行为是通过绘制平均值和平均值周围的95%置信区间来聚合每个x值的多个度量值

```
"""
```

```
sns.relplot(x="timepoint", y="signal", kind="line", data=fmri)
```



3. ci = None

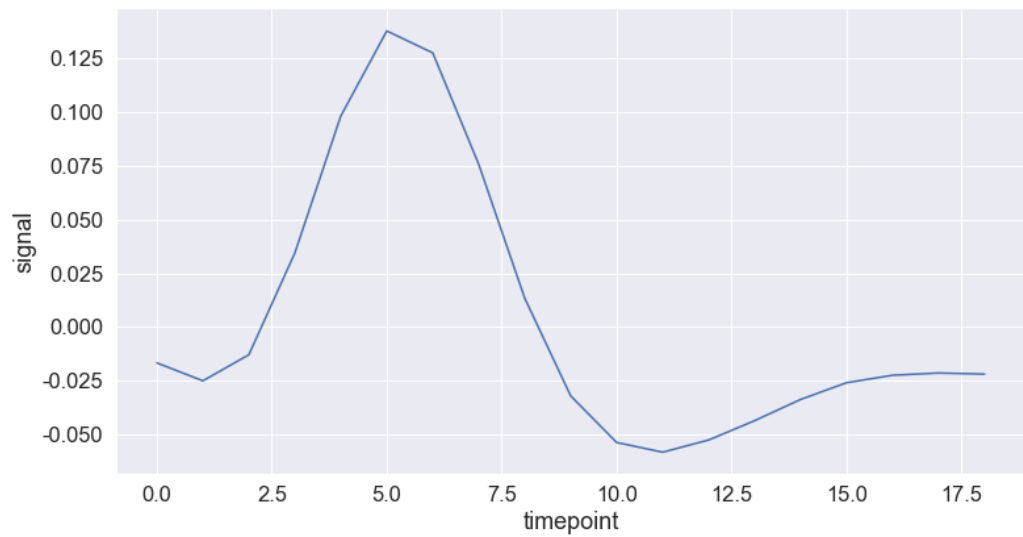
```
"""
```

```
ci = None
```

置信区间是使用bootstrapping计算的，对于较大的数据集，它可能是时间密集型的。因此，有可能禁用它们

```
"""
```

```
sns.relplot(x="timepoint", y="signal", ci=None, kind="line", data=fmri)
```



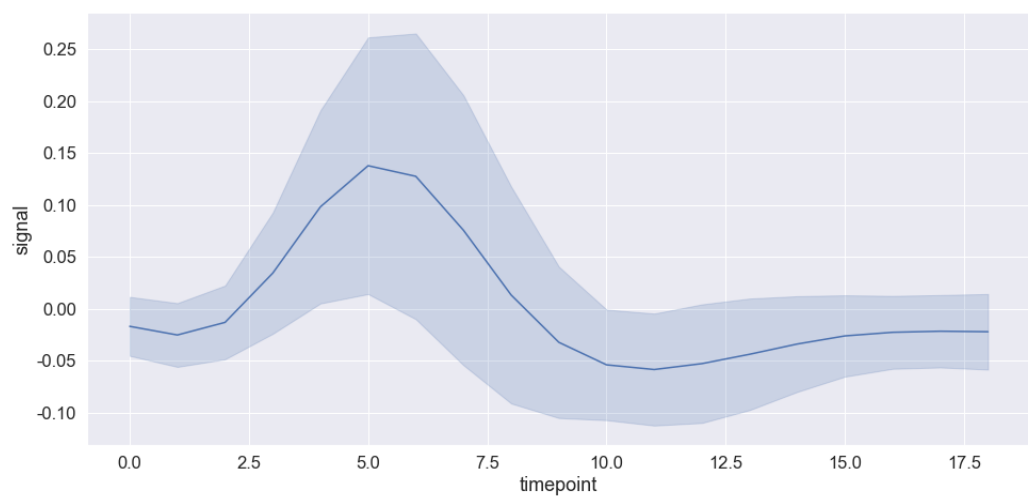
4.ci = sd

"""

另一个不错的选择，特别是对于较大的数据，
是通过绘制标准差而不是置信区间来表示分布在每个时间点的分布范围：

"""

```
sns.relplot(x="timepoint", y="signal", kind="line", ci="sd", data=fmri)
```



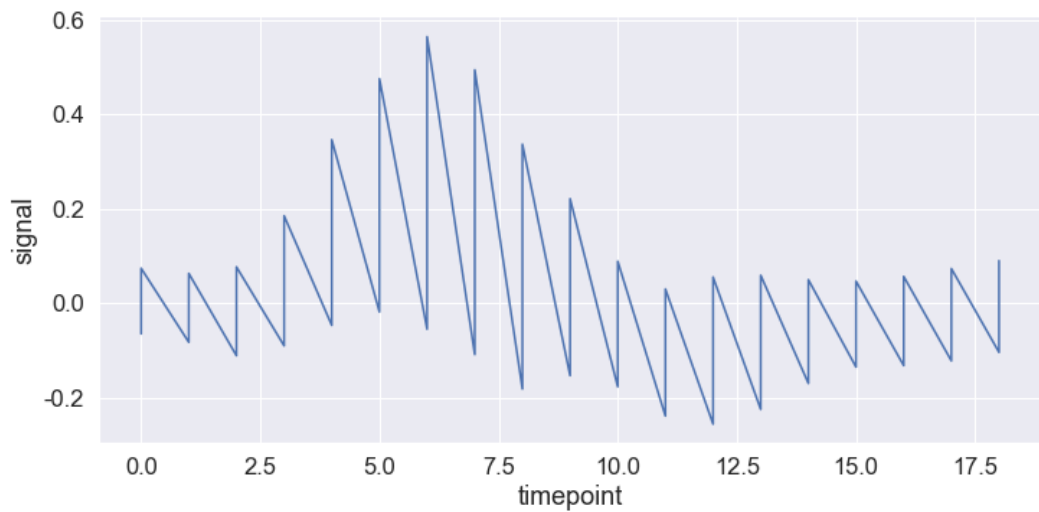
5.estimator = None

"""

要完全关闭聚合，将`estimator`参数设置为`None`。
当数据在每个点有多个观察值时，这可能会产生奇怪的效果。

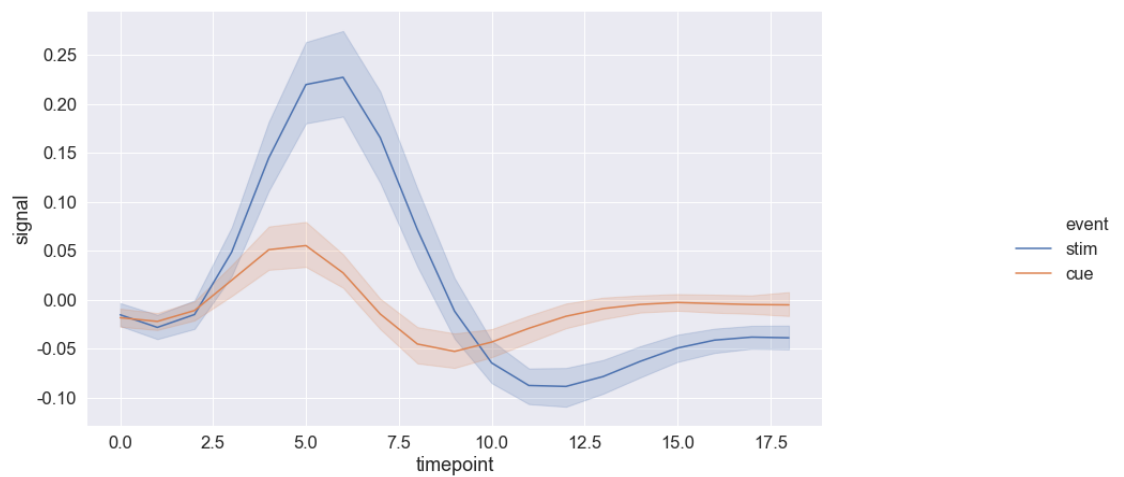
"""

```
sns.relplot(x="timepoint", y="signal", estimator=None, kind="line", data=fmri)
```

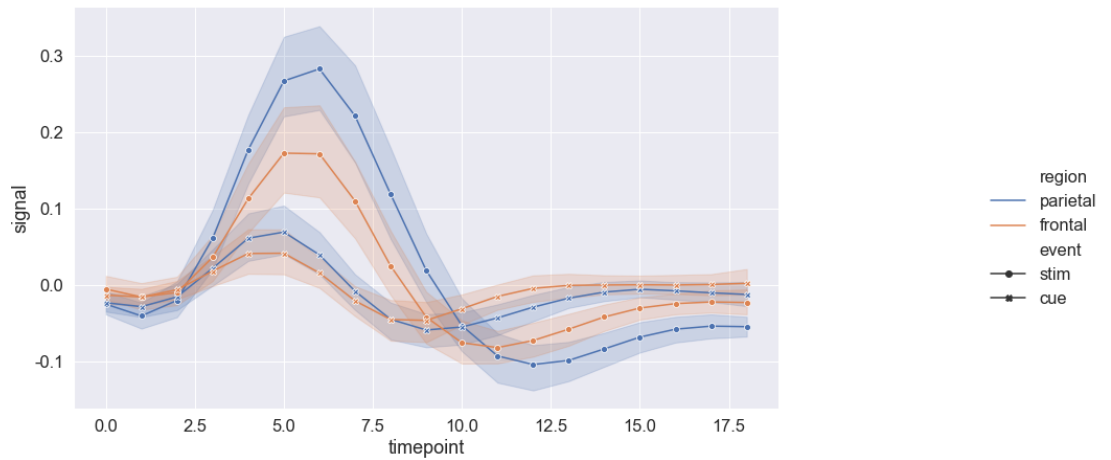
6. hue : 分组变量

```
"""hue
添加一个具有两个级别的hue语义
, 将图分割为两条线和错误带, 每条线用不同的颜色表示它们对应的数据子集
"""
sns.relplot(x="timepoint", y="signal", hue="event", kind="line", data=fmri)
```



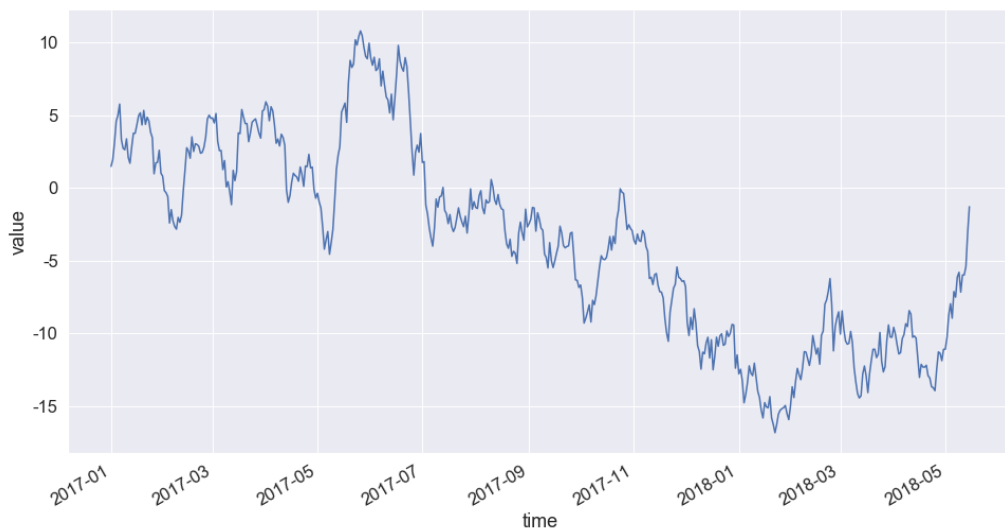
7. dashes,marker,style,hue

```
"""
多种标记和风格区分
"""
sns.relplot(x="timepoint", y="signal", hue="region", style="event",
            dashes=False, markers=True, kind="line", data=fmri)
```



8. 绘制日期格式

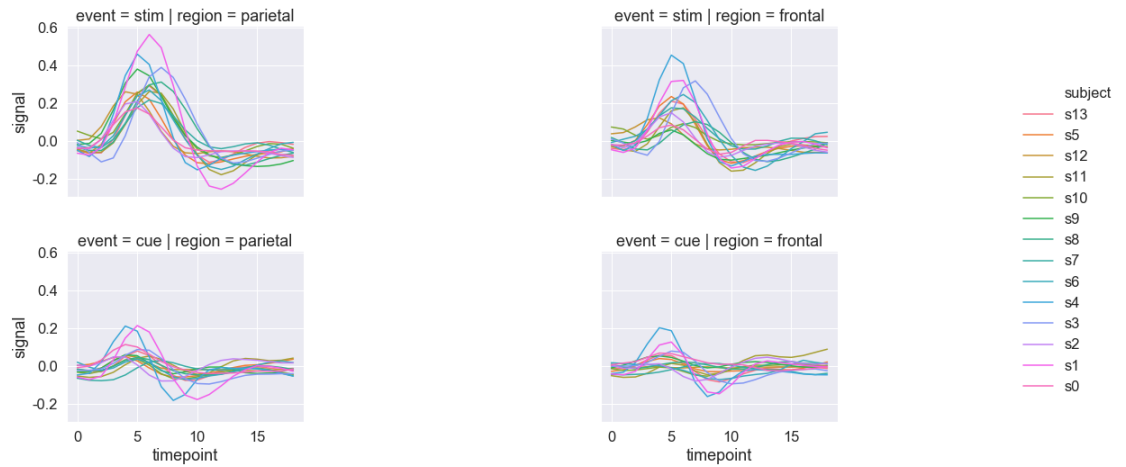
```
#绘制日期格式
df = pd.DataFrame(dict(time=pd.date_range("2017-1-1", periods=500),
                        value=np.random.randn(500).cumsum()))
g = sns.relplot(x="time", y="value", kind="line", data=df)
g.fig.autofmt_xdate()
```



多因素相关

1. 多子图

```
#多子图
sns.relplot(x="timepoint", y="signal", hue="subject",
            col="region", row="event", height=3,
            kind="line", estimator=None, data=fmri)
```



2. 单变量多子图

''''''

检查一个变量的多个级别的效果时，在列上对该变量进行切面处理，
然后将切面“包装”到行中是一个好主意

''''''

```
sns.relplot(x="timepoint", y="signal", hue="event", style="event",
            col="subject", col_wrap=5,
            height=3, aspect=.75, linewidth=2.5,
            kind="line", data=fmri.query("region == 'frontal'"))
```

