

zhuanlan.zhihu.com

Python可视化 | Seaborn5分钟入门(五)——Implot回归图

9-11 minutes

微信公众号：「Python读财」
如有问题或建议，请公众号留言

Seaborn是基于matplotlib的Python可视化库。 它提供了一个高级界面来绘制有吸引力的统计图形。
Seaborn其实是在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，不需要经过大量的调整就能使你的图变得精致。



注：所有代码均在IPython notebook中实现

Implot(回归图)

Implot是用来绘制回归图的，通过Implot我们可以直观地总览数据的内在关系

先总览一下Implot的API：

```
seaborn.Implot(x, y, data, hue=None, col=None, row=None, palette=None,
               col_wrap=None, size=5, aspect=1, markers='o', sharex=True, sharey=True,
               hue_order=None, col_order=None, row_order=None, legend=True, legend_out=True,
               x_estimator=None, x_bins=None, x_ci='ci', scatter=True, fit_reg=True, ci=95,
               n_boot=1000, units=None, order=1, logistic=False, lowess=False, robust=False,
               logx=False, x_partial=None, y_partial=None, truncate=False, x_jitter=None, y_jitter=None,
               scatter_kws=None, line_kws=None)
```

可以看到Implot的参数还是比较多的，接下来我们就挑一些常用的来讲一下，有一些参数会涉及到一些统计学的知识。

老套路，先导入相应的包：

```
import seaborn as sns
%matplotlib inline
sns.set(font_scale=1.5, style="white")
```

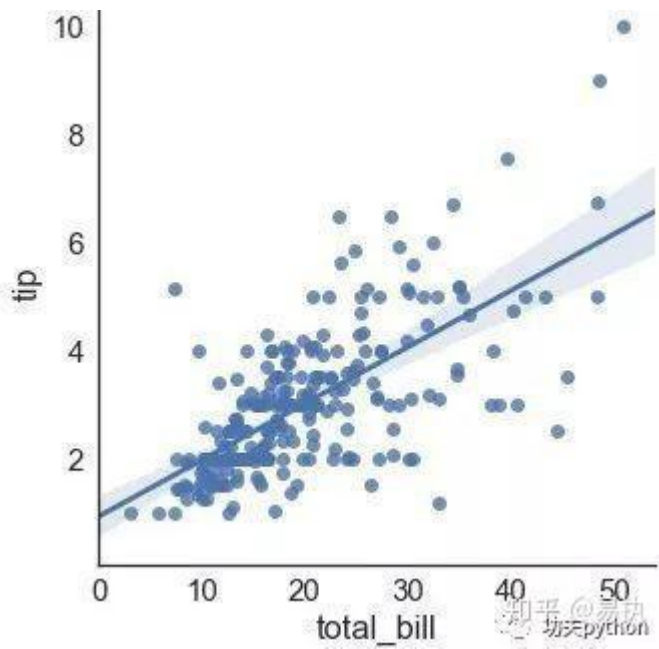
本次试用的数据集是Seaborn内置的tips小费数据集：

```
data=sns.load_dataset("tips")
data.head(5)
```

	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

我们先来看一下Implot是什么样的

```
sns.lmplot(x="total_bill",y="tip",data=data)
```



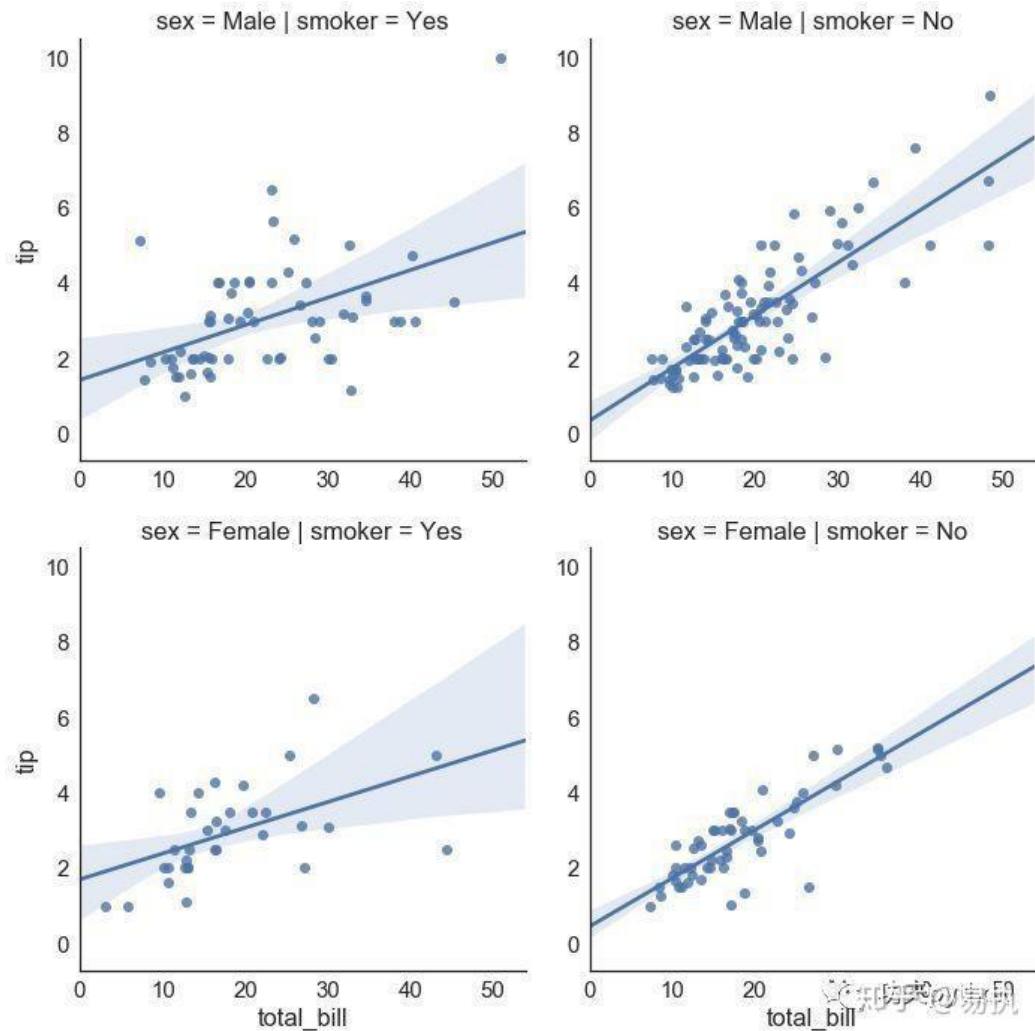
可以看到lmplot对所选数据集进行了一元线性回归，拟合出了一条最佳的直线，

接下来进入具体参数的演示。

col:根据所指定属性在列上分类

row:根据所指定属性在行上分类

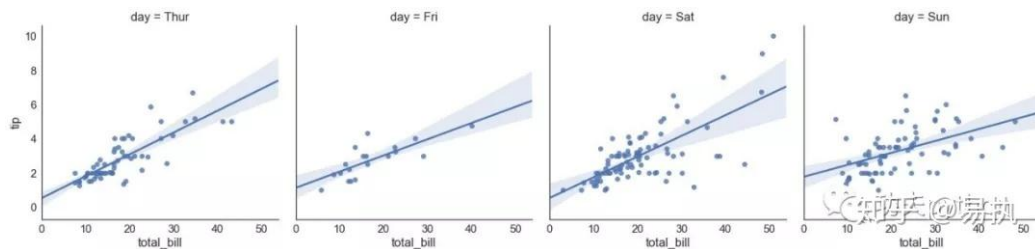
```
sns.lmplot(x="total_bill",y="tip",data=data,row="sex",col="smoker")
```



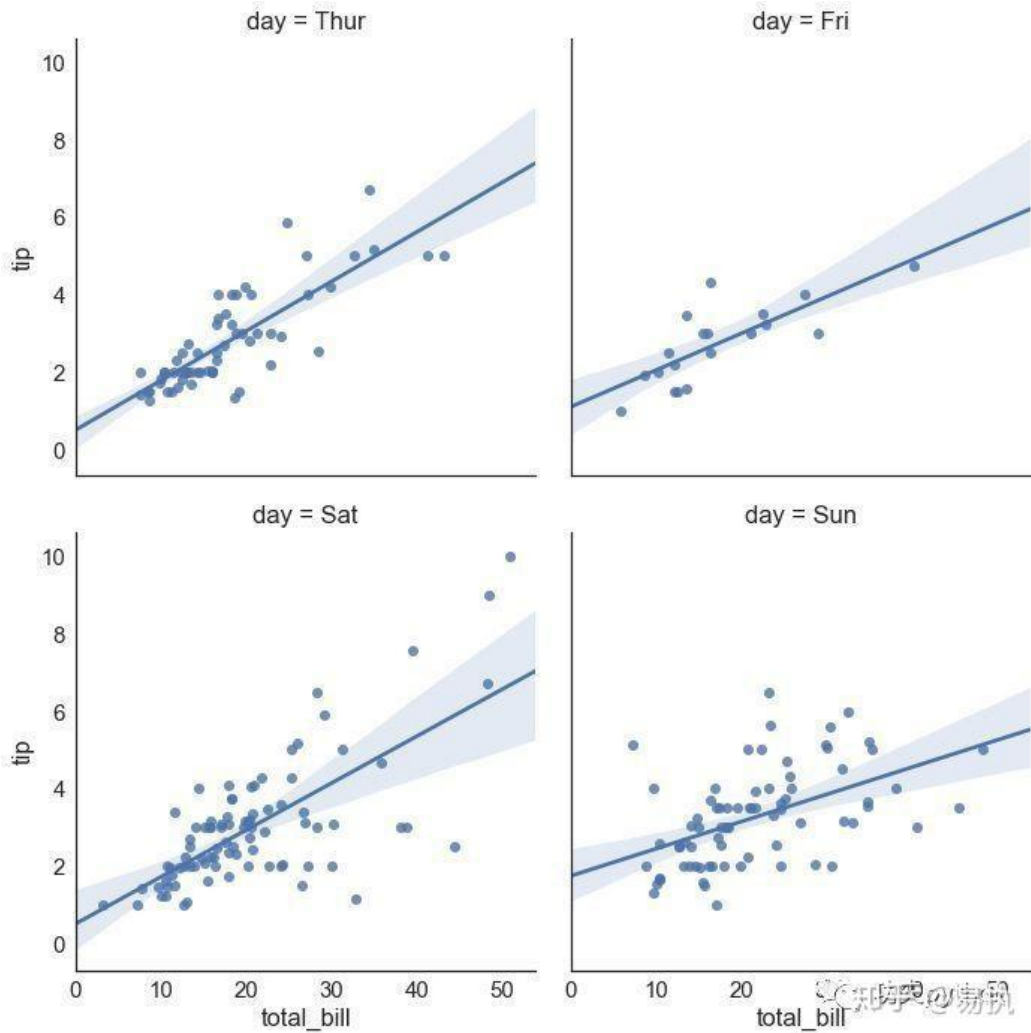
结合我们的数据集，看上图的横纵坐标就可以明白这两个参数的用法

`col_wrap`:指定每行的列数，最多等于`col`参数所对应的不同类别的数量

```
sns.lmplot(x="total_bill",y="tip",data=data,col="day",col_wrap=4)
```



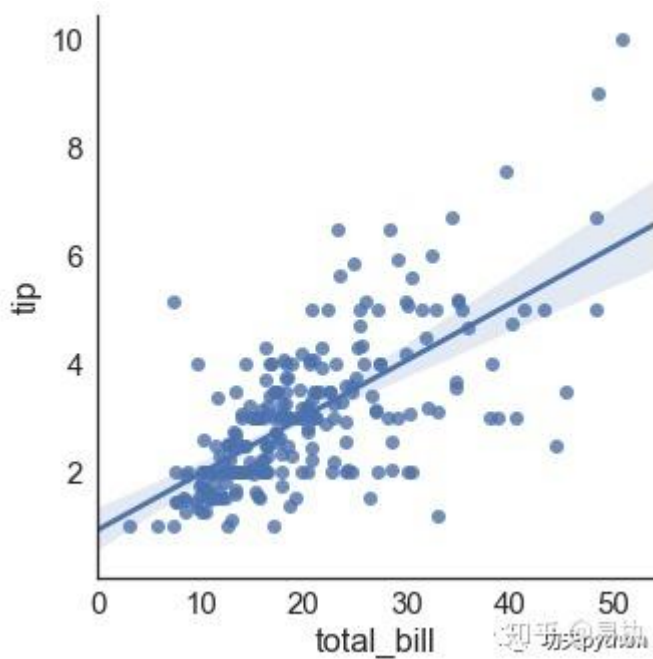
```
sns.lmplot(x="total_bill",y="tip",data=data,col="day",col_wrap=2)
```



aspect:控制图的长宽比

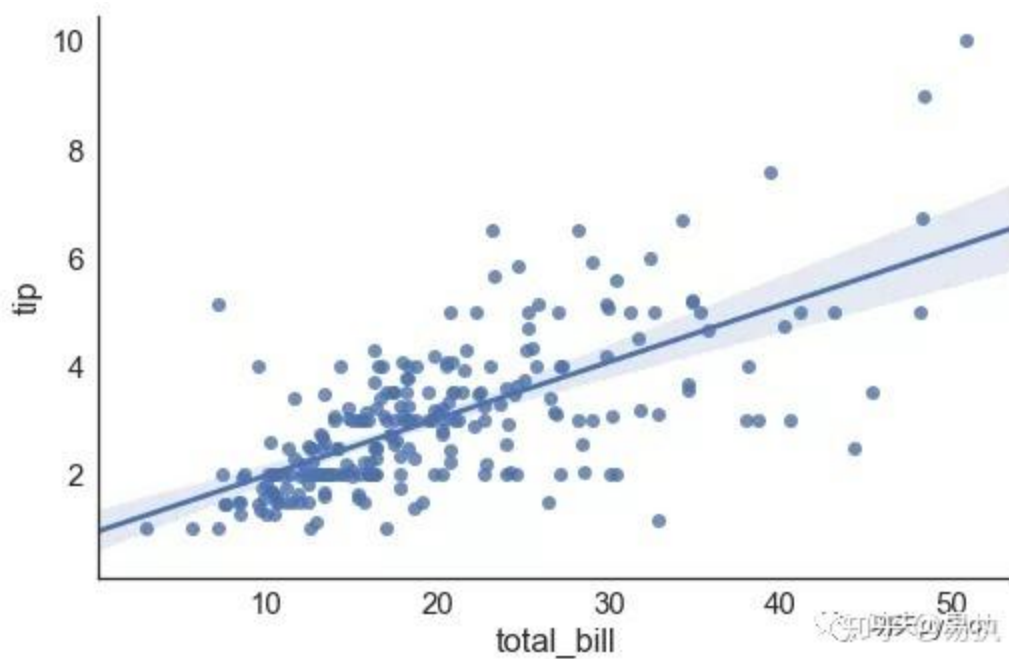
```
sns.lmplot(x="total_bill",y="tip",data=data,aspect=1)
```

#长度比宽度等于一比一，即正方形



```
sns.lmplot(x="total_bill",y="tip",data=data,aspect=1.5)
```

#长度比宽度等于1:1.5, 可以看到横轴更长一点

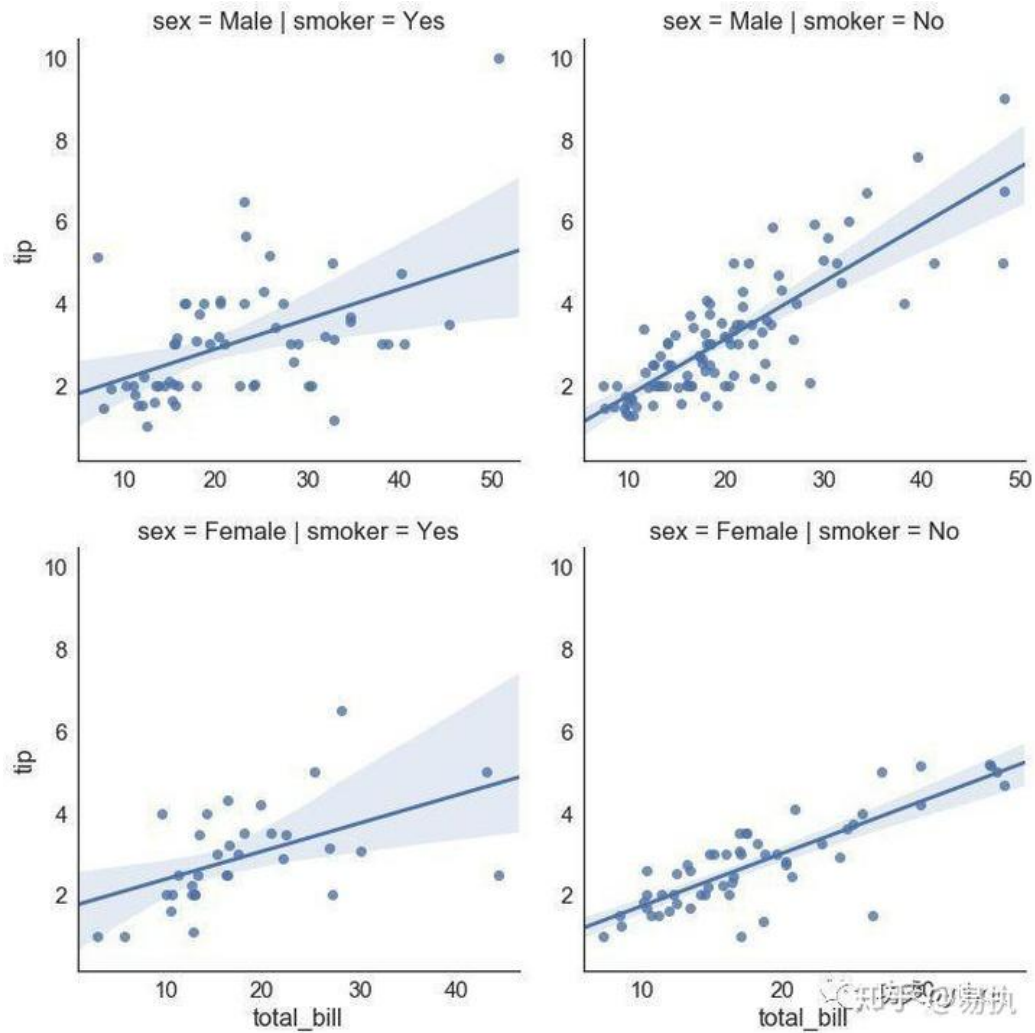


sharex: 共享x轴刻度 (默认为True)

sharey: 共享y轴刻度 (默认为True)

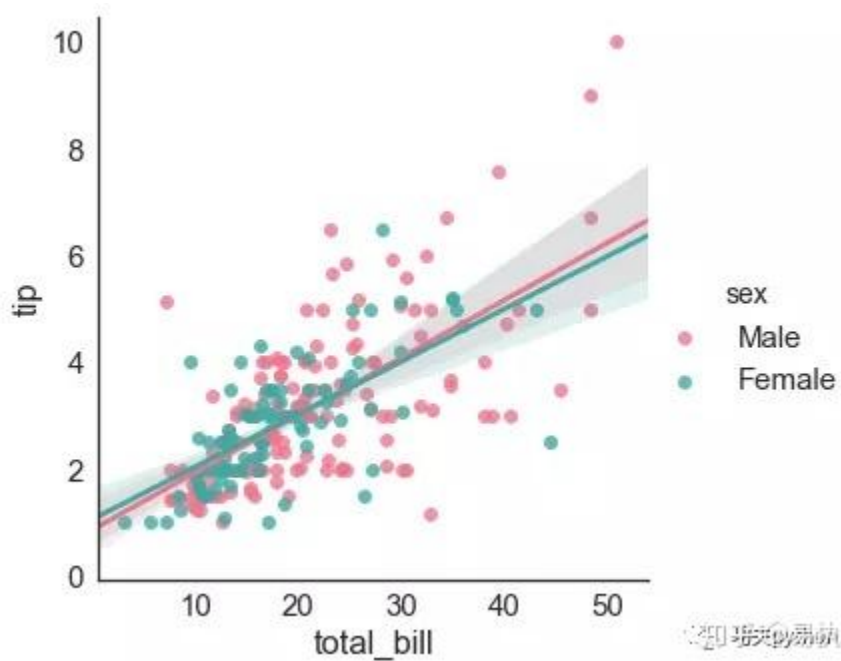
```
sns.lmplot(x="total_bill",y="tip",data=data,row="sex",col="smoker",sharex=False)
```

#可以看到设置为False时, 各个子图的x轴的坐标刻度是不一样的



hue:用于分类

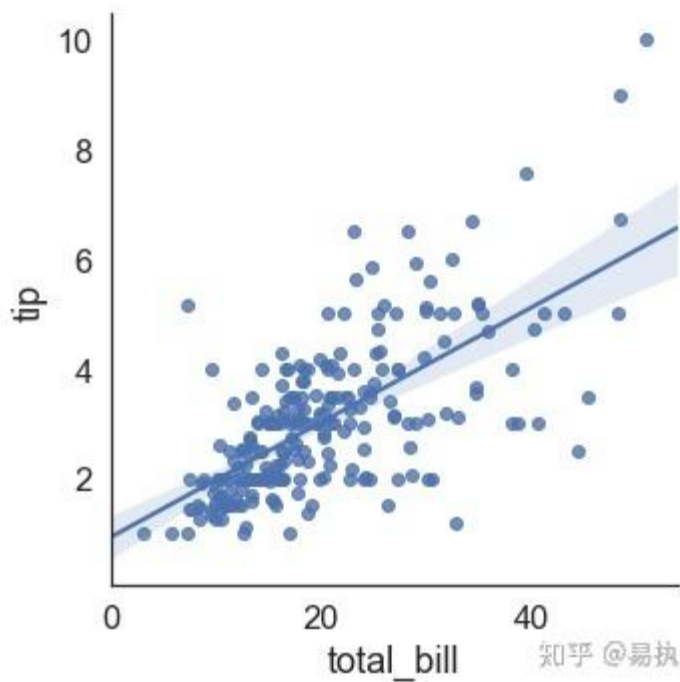
```
sns.lmplot(x="total_bill",y="tip",data=data,hue="sex",palette="husl")
```



ci:控制回归的置信区间 (有学过统计学的同学们应该都是知道滴)

#采用0.95的置信区间

```
sns.lmplot(x="total_bill",y="tip",data=data,ci=95)
```

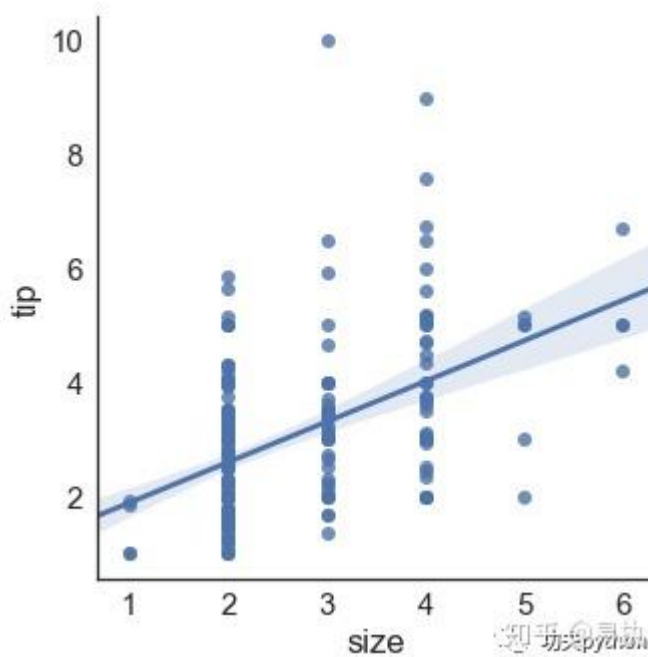


x_jitter:给x轴随机增加噪音点

y_jitter:给y轴随机增加噪音点

设置这两个参数不影响最后的回归直线

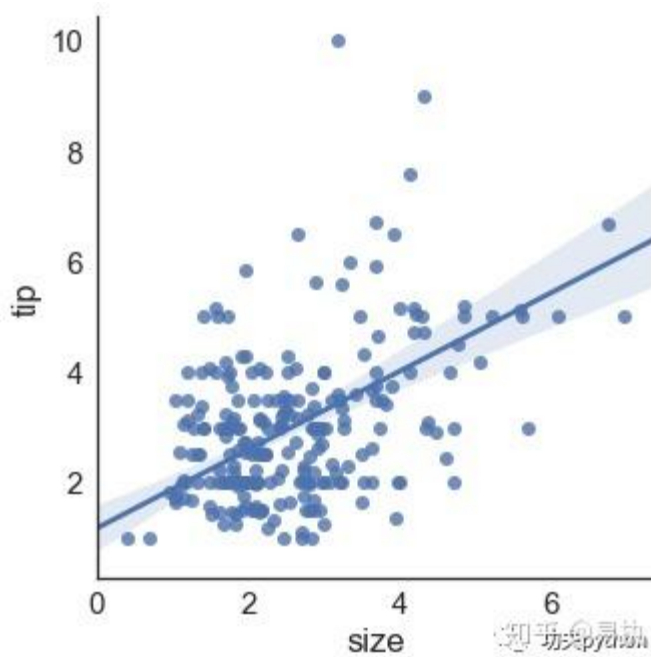
```
sns.lmplot(x="size",y="tip",data=data,x_jitter=False)
```




```
sns.lmplot(x="size",y="tip",data=data,x_jitter=True)
```

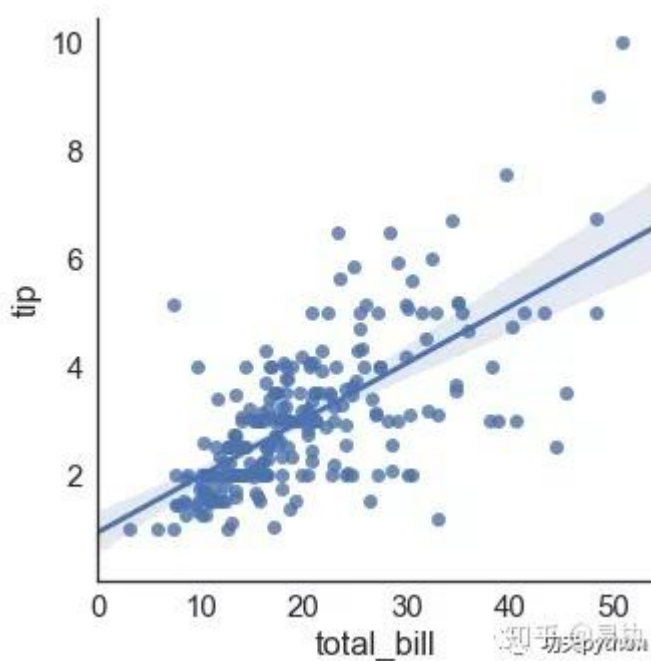
#可以看到刚才的一列一列的数据点被随机

#打乱了，但不会影响到最后的回归直线

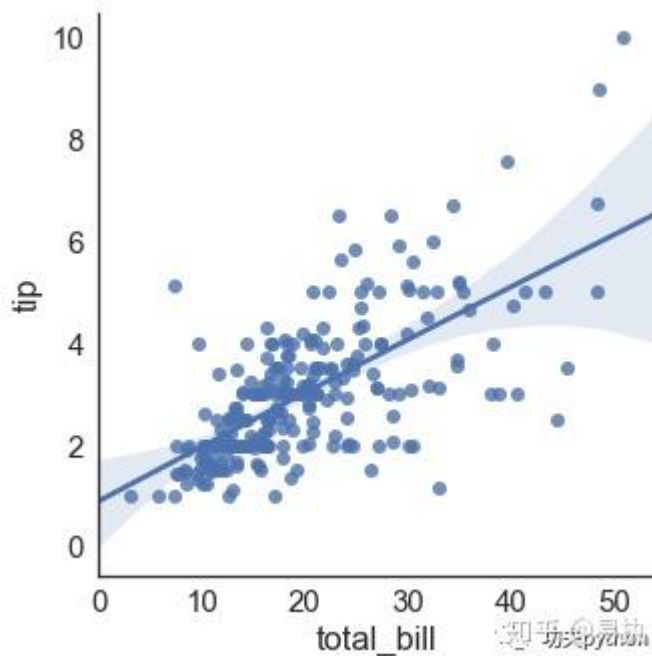


order:控制进行回归的幂次（一次以上即是多项式回归）

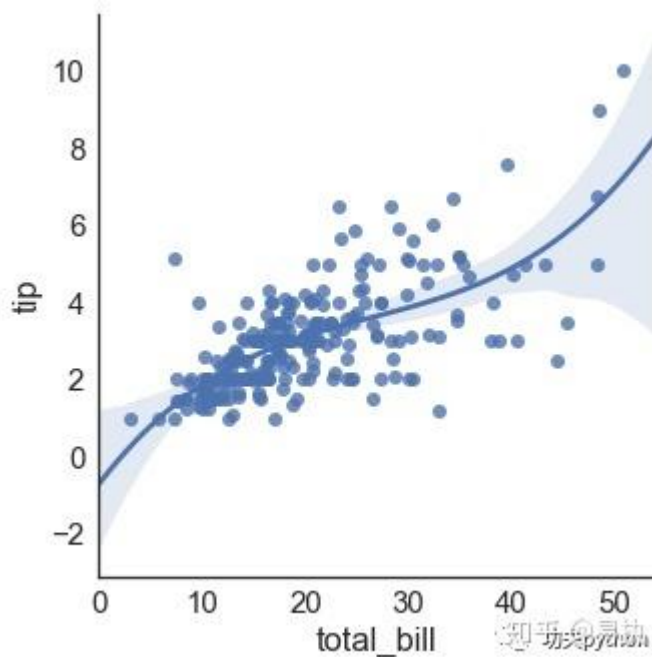
```
sns.lmplot(x="total_bill",y="tip",data=data,order=1) #一元线性回归
```



```
sns.lmplot(x="total_bill",y="tip",data=data,order=2) #次数最高为2
```



```
sns.lmplot(x="total_bill",y="tip",data=data,order=3) #次数最高为3
```



还有一些参数涉及到了更深入的统计学知识，在这里就不一一介绍了，有兴趣的可以自行查看官方文档！

原创不易，如果觉得有点用，希望可以随手点个赞，拜谢各位老铁。