

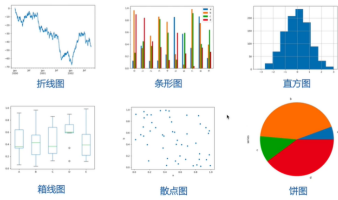
...Pandas的基本绘图函数（常用参数表、图形类型、样式参数设置表、条形图、直方图、箱线图、散点图、气泡图、六边箱图、饼图）

一、基于Matplotlib的Pandas绘图方法

Pandas绘制图形相较于Matplotlib来说更为简洁，基础函数为df.plot(x,y)

```
例：
>>>df.plot('time','Money')
```

二、基本数据图形类型



通过kind可以设置图形的类型，df.plot()默认绘制折线图，df.plot(kind='')用于设置各类图形，如下表所示

df.plot() 常用参数表		通过kind设置图形类型		注意：若在绘制图形时，对象是Series，那么Series对象的索引会自动会设置为x轴，但是不会显示刻度标签
参数	说明	参数kind	图形类型	
x	x轴数据	line	折线图	linestyle: 设置线的样式（简写：style） marker: 设置标记样式
y	y轴数据	bar/barh	条形图	
kind	设置图表类型（具体可见右表）	hist	直方图	color: 设置线的颜色（简写：c）
subplots	判断图片是否有子图	box	箱线图	linewidth: 设置显得粗细（简写：lw）
figsize	图片尺寸大小	scatter	散点图	样式参数设置,总结如下表:
label	用于图例的标签	pie	饼图	
style	设置风格，传入字符串（如'ko-'）			
alpha	设置透明度（越靠近0越透明）			
rot	旋转刻度标签（rotation缩写，取值0-360）			
fontsize	设置轴刻度的字体大小			
xticks	用作x轴刻度的值			
yticks	用作y轴刻度的值			
xlim	x轴的范围			
ylim	y轴的范围			
grid	显示刻度背景网格			
colormap	设置图形颜色			
layout	设置子图排列格式			

linestyle	marker	color
- 实线	^ 菱形	b 蓝色
-- 虚线	v 向下三角形点	r 红色
-. 点划线	< 向左三角形点	c 青色
: 点状虚线	> 向右三角形点	m 洋红色
- 实心小圆点	x 向上三角形点	y 黄色
· 像素点	s 正方形	k 黑色
	D 圆形	w 白色
	d 小菱形	g 绿色
	o 圆圈	blueviolet 蓝紫色
	p 五边形	darkgreen 深绿色

三、条形图

1、Series.plot(kind = 'bar')

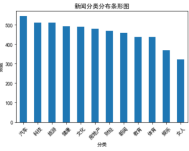
```
Series绘制条形图时，通常结合value_counts()显示各值的出现频率

除了传入kind参数外，也可以简写为data.plot.bar()的形式，此类方法也适用于其他图形。

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus']=False
data = pd.read_csv('C:/Users/xhl/Desktop/input/train_sample_utf8.csv')
cg = data['分类'].value_counts()

cg
分类  544
科技  511
旅游  510
健康  492
文化  491
房地产  480
财经  469
新闻  458
体育  437
教育  437
娱乐  370
女人  322
Name: 分类, dtype: int64

#两种绘图方式均可
#eg.plot(kind='bar',rot=45)
cg.plot.bar(rot=45)
plt.xlabel('分类')
plt.ylabel('频数')
plt.title('新闻分类分布条形图')
```



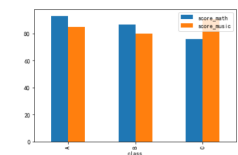
2、DataFrame.plot.bar()

DataFrame绘制条形图时，会将每一行的值分为一组，各列名称作为图例

```
df = pd.read_excel('C:/Users/xhl/Desktop/input/class.xlsx')
df
  class  sex  score_math  score_music
0    A  male          95           79
1    A female          96           90
2    B female          85           85
3    C  male          93           92
4    B female          84           90
5    B  male          88           70
6    C  male          59           89
7    A  male          88           86
8    B  male          89           74

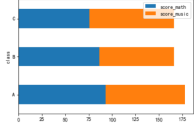
df_class = df.groupby('class').mean()
df_class
      score_math  score_music
class
A             93.0         85.00
B             86.5         79.75
C             76.0         90.50

df_class.plot.bar()
```



3、堆积条形图

设置参数`stacked=True`，每行的值就堆积起来，更易于观察比较各组的分数
`df_class.plot.barh(stacked=True)`

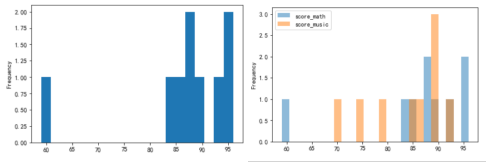


四、直方图

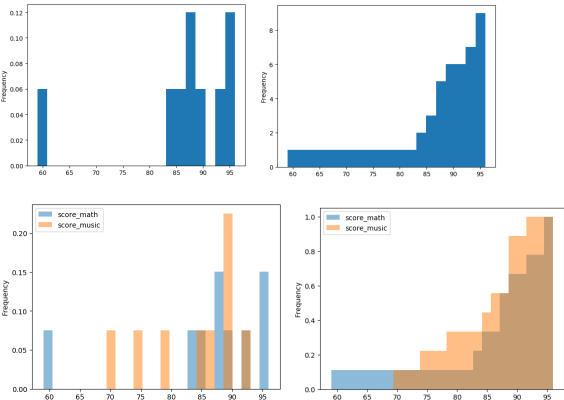
1、Series.plot.hist()

2、DataFrame.plot.hist()

通过`bins`设置面元，表示将数据分为多少组：
当有多种类型进行数据可视化时（堆叠情况发生时）可以设置透明度`alpha`，将数据的可视化度加明显
`df['score_math'].plot.hist(bins=20)`
`df[['score_math','score_music']].plot.hist(bins=25,alpha=0.5)`

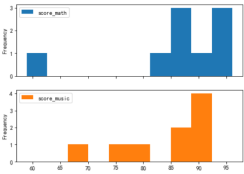


默认画的是频数图，如果想画频率图，可以修改参数 `density = True`
调整参数 `cumulative = True` 绘制累积直方图



3、绘制子图

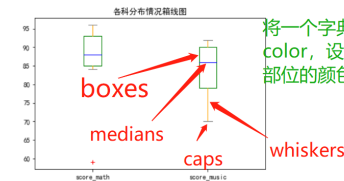
设置参数`subplots=True`，可将各列分布直方图分别绘制在不同的子图中
`df.plot.hist(subplots=True,figsize = (7,5))`



五、箱线图

1、常规箱线图

箱线图是一种用作显示数据分数情况的统计图
用于考察数据之间的分布状况，同时又用于考察数据之间的离散和分布程度，离散程度高表明数据之间的差异较大：
现绘制各科分数分布箱线图：
`color = dict(boxes='Green',whiskers='Orange',medians='Blue',caps='Gray')`
`df.plot.box(color=color,sym='r+')`//sym表示离群点
`plt.title('各科分布情况箱线图')`



将一个字典传入参数
`color`，设置箱线图个
部位的颜色

可以通过修改箱线图的 `return_type` 参数为 `dict` 使得绘图函数返回一个字典，这可以让我们方便地从图中提取一些数据。
`>>> box = df.plot.box(color=color,sym='r+',return_type='dict')`

```
>>>box
['whiskers': [matplotlib.lines.Line2D object at 0x00000186969E8D08], <matplotlib.lines.Line2D object at 0x00000186969F3438>, <matplotlib.lines.Line2D object at 0x00000186969BA5F8>, <matplotlib.lines.Line2D object at 0x00000186960D2C18>],
'caps': [matplotlib.lines.Line2D object at 0x00000186969F378D], <matplotlib.lines.Line2D object at 0x00000186969F3AC8>, <matplotlib.lines.Line2D object at 0x00000186969FDC88>, <matplotlib.lines.Line2D object at 0x00000186969FDFD0>],
'boxes': [matplotlib.lines.Line2D object at 0x00000186969E8C88], <matplotlib.lines.Line2D object at 0x00000186969F04E0>],
'medians': [matplotlib.lines.Line2D object at 0x00000186969F3E10], <matplotlib.lines.Line2D object at 0x0000018696A09358>],
'fliers': [matplotlib.lines.Line2D object at 0x00000186969F3EB8], <matplotlib.lines.Line2D object at 0x0000018696A096A0>],
'means': []]

>>> box.keys()
dict_keys(['whiskers', 'caps', 'boxes', 'medians', 'fliers', 'means'])
```

其中：'boxes' 对应的是箱体部分，'fliers' 对应的是异常值点，'whiskers' 对应的是两条须，可以通过提取列表的第一个元素获得相应的图块，然后使用 `get_xydata()` 获取其对应的数据点

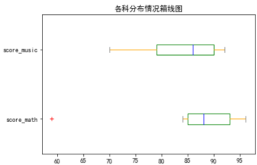
例：查看异常值部分，使用 `get_xydata()` 获取对应的异常值

```
>>> box['fliers'][0].get_xydata()
array([[1., 59.]])
```

2. 水平箱线图

通过 `vert = False` 可设置箱线图为水平方向展示

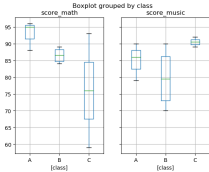
```
color = dict(boxes='Green',whiskers='Orange',medians='Blue',caps='Gray')
df.plot.box(color=color,sym='r+',vert=False)
plt.title('各科分布情况箱线图')
```



3. 分组绘图by

箱线图也可以使用 `df.boxplot()` 的方法，设置参数 `by` 根据某列的唯一值将数据进行分组绘图；子图先列进行分组，然后按照班级分类进行分组（即子图的个数 = 列的个数）；当类别较多时，可以设定 `columns`，也就是要分析的列

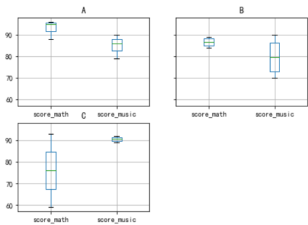
如按照班级分组：
`df.boxplot(by='class',sym='r+')`



当 `boxplot` 默认绘制了两个标题时，可以通过 `suptitle` 和 `title` 进行调整，并设定 `grid` 参数为 `False` 不显示刻度背景网格

4. 分组绘图groupby

使用 `df.groupby().boxplot()` 的方法，子图先按照班级分类进行分组，然后每个子图再按照各列进行分组（即子图的个数 = 班级分类的个数）
`df.groupby('class').boxplot(sym='r+',figsize=(7,5))`



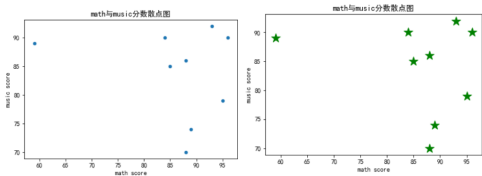
六、散点图

1、基本散点图

`df.plot.scatter(x,y)`，绘制散点图需要传入参数 `x` 和 `y`，分别设置 `x` 轴与 `y` 轴的数据。

基本的散点图主要考察两个散点图之间的相互依存关系（相关关系）

```
df.plot.scatter(x='score_math',y='score_music')
plt.title('math与music分数散点图')
plt.xlabel('math score')
plt.ylabel('music score')
```



设置点的样式：

- s：设置点的大小
- marker：设置点的形状
- c：设置点的颜色

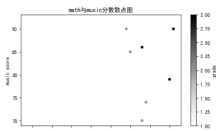
```
df.plot(kind='scatter',x='score_math',y='score_music',c='g',marker='w',s=200)
```

说明：在散点图的绘制过程中，有些数据可能会重叠，可通过设置透明度 `alpha` 解决此类问题，但是散点图本就是分析趋势，所以一般情况下问题不大。

2、加入影响因素（分类散点图）

将班级等级进行数值编码，并存与新列 `grade` 中：
`df['grade'] = df['class'].replace(['A','B','C'],[3,2,1])`
`df.plot(kind='scatter',x='score_math',y='score_music',c='grade',figsize=(7,4))`

注意：参数 `c` 既可以传入颜色，也可以传入列名，不同值的颜色会发生渐变

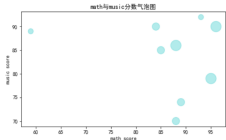


3、气泡图（散点图的变种）

散点图可用于展示三个变量（不是单纯的三个变量之间的关系，数据分析一般分析两两变量之间的关系）之间的关系，现使用气泡图的形式加入班级等级影响因素。

为了便于观察，将grade的编码替换为等比数列。

```
df['grade'] = df['grade'].replace([1,2,3],[1,2,4])
df.plot.scatter(x='score_math',y='score_music',figsize=(7,4),c='c',alpha=0.3,s=100*df['grade'])
plt.title('math与music分数散点图')
plt.xlabel('math score')
plt.ylabel('music score')
```

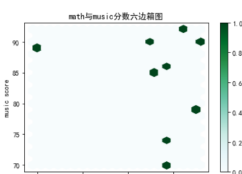


加入plt.colorbar(), 可以给子图添加colorbar（颜色条或渐变色条）

4、六边箱图

六边箱图又称为高密度散点图，df.plot.hexbin()，若数据点太密集，该图结果优于散点图

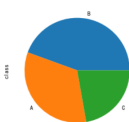
```
df.plot.hexbin(s='score_math',y='score_music',gridsize=20)
plt.title('math与music分数六边箱图')
plt.xlabel('math score')
plt.ylabel('music score')
```



注意：其中参数gridsize用于设置x轴方向的六边形数量，默认为100个，使用时根据自身需求进行调整

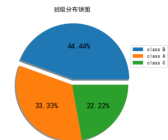
查看某特征的分布情况，df.plot.pie() 会自动添加标题和各部分的名称

```
df['class'].value_counts().plot.pie()
```



如何去除默认文字、添加图例、添加每部分百分比数值、设置凸出部分、添加阴影？

```
labels = None: 可去除饼图外侧每部分的名称；
plt.ylabel(''): 设置y轴标签为空可以去掉默认标题；
legend = True: 添加图例（或者plt.legend()）；
bbox_to_anchor: 用来移动图例的位置，其中0.9表示左右，0.8表示上下；
labels: 表示显示的图例文字；
autopct: 控制图内百分比设置，'%d' 的表示输出出一个百分号，前一个%是转义字符；
explode: 设置每一块饼图离开中心的距离，如将比重最大的一块凸出；
shadow = True: 添加阴影。
explode=(0.1,0,0) 表示默认不离开
df['class'].value_counts().plot.pie(figsize=(5,5),labels=None,
                                autopct='%1.2f%%',fontSize=15,
                                explode = explode,shadow = True)
plt.ylabel('')
plt.legend(bbox_to_anchor=[0.9,0.8],labels=['class B','class A','class C'])
plt.title('班级分布饼图')
```



七、饼图