

```
1 %matplotlib inline
```

```
1 import matplotlib
2 matplotlib.__version__
```

## 第一个Matplotlib绘图程序

- 首先导入 Matplotlib 包中的 Pyplot 模块,并以 as 别名的形式简化引入包的名称。

```
1 from matplotlib import pyplot as plt
```

- 接下来,使用 NumPy 提供的函数 `arange()` 创建一组数据来绘制图像。

```
1 # 引入numpy包
2 import numpy as np
3 # 获得-50到50之间的ndarray对象
4 x = np.arange(-50,51)
```

- 上述所得 `x` 的值作用到 `x` 轴上,而该值对应的平方值,也就是 `y` 值,使用以下方式获取

```
1 y = x ** 2
```

- 使用 `plt` 的 `plot()` 函数对 `x`、`y` 进行绘制。

```
1 # plot() 绘制线性图表
2 plt.plot(x,y)
```

已上图还有很多需要修饰的地方,后期我们会慢慢学到

```
1 y = x^2
```

```
1
```

## matplotlib的基本方法

方法名	说明
<code>title()</code>	设置图表的名称
<code>xlabel()</code>	设置x轴名称
<code>ylabel()</code>	设置y轴名称
<code>xticks(ticks,label,rotation)</code>	设置x轴的刻度,rotation旋转角度
<code>yticks()</code>	设置y轴的刻度
<code>show()</code>	显示图表
<code>legend()</code>	显示图例
<code>text(x,y,text)</code>	显示每条数据的值 x,y值的位置

## 1. 图表名称 `plt.title()`

```

1 import numpy as np
2 # x轴获得-50到50之间的ndarray对象
3 x = np.arange(-50,50)
4 # y轴的值是x轴的平方
5 y = x **2
6 # 设置图名
7 plt.title("y= x^2")
8 # 绘制图形
9 plt.plot(x,y)

```

## 如果标题写成中呢?

```

1 plt.title("y等于x的平方")
2 plt.plot(x,y)

```

 默认不支持中文: missing from current font 字体丢失

## 修改字体配置 `plt.rcParams["font.sans-serif"]`

字体说明:

中文字体	说明
‘SimHei’	中文黑体
‘Kaiti’	中文楷体
‘LiSu’	中文隶书
‘FangSong’	中文仿宋
‘YouYuan’	中文幼圆
STSong	华文宋体

临时设置

```
1 plt.rcParams['font.sans-serif'] = ["SimHei"]
```

```
1 plt.title("y等于x的平方")
2 plt.plot(x,y)
```

字体设置时,字体名称不区别大小写

```
1 # 在尝试下别的字体
2 plt.rcParams['font.sans-serif'] = ["SimHei"]
```

```
1 plt.title("y等于x的平方")
2 plt.plot(x,y)
```

## 设置了中文字体后,负数又没办法正确显示了?

```
1 # 创建x为-10到10的整数
2 x = np.arange(-10,10)
3 # y是x的平方
4 y = x ** 2
5 plt.plot(x, y)
```



当字体设置支持中文后,必须设置负号,否则当数值中出现负值时,负号无法正常显示

```
1 # 解决方式: 修改轴中的负号编码 axis
2 plt.rcParams['axes.unicode_minus'] = False
```

```

1 # 创建x为-10到10的整数
2 x = np.arange(-10,10)
3 # y是x的平方
4 y = x ** 2
5 # 设置标题
6 plt.title('y=x^2 x取值范围为: [-10,10)')
7 plt.plot(x, y)

```

字体问题总结:

需要修改2个配置信息:

`plt.rcParams['font.sans-serif']=['SimHei']` # 用来设置字体样式以正常显示中文标签

`plt.rcParams['axes.unicode_minus']=False` # 默认是使用Unicode负号, 设置正常显示字符

## ✧ 2. x轴和y轴名称

`xlabel()` `ylabel()`

```

1 # 创建x为-10到10的整数
2 x = np.arange(-10,10)
3 # y是x的平方
4 y = x ** 2
5 # 设置标题
6 plt.title('y=x^2 x取值范围为: [-10,10)')
7 # 设置x轴名称
8 plt.xlabel("x 轴")
9 # 设置y轴名称
10 plt.ylabel("y 轴")
11
12 plt.plot(x, y)

```

对于上面的图,如果觉得字体偏小或线条太细,可以设置标签文字大小和线条粗细

`fontsize`参数: 设置文字大小

`linewidth`参数: 设置线条

```

1 # 创建x为-10到10的整数
2 x = np.arange(-10,10)
3 # y是x的平方
4 y = x ** 2
5 # 设置标题
6 plt.title('y=x^2 x取值范围为: [-10,10]', fontsize=16)
7 # 设置x轴名称
8 plt.xlabel("x 轴", fontsize=12)
9 # 设置y轴名称
10 plt.ylabel("y 轴")
11 # 画图
12 plt.plot(x, y)

```

```

1 # 设置标题
2 plt.title('y=x^2 x取值范围为: [-10,10]', fontsize=16)
3 # 设置x轴名称
4 plt.xlabel("x 轴", fontsize=12)
5 # 设置y轴名称
6 plt.ylabel("y 轴")
7 # 设置线条宽度
8 plt.plot(x, y, linewidth=5)

```

运行上面代码,除了线条变化外,大家有没有发现图像有什么变化?

1

```

1 # 我们可以在一张图片中绘制多个线条
2 # 创建x为-10到10的整数
3 x = np.arange(-10,10)
4 # y1是x的平方
5 y1 = x ** 2
6 # y2等于x
7 y2 = x
8 # 设置标题
9 plt.title('y=x^2 x取值范围为: [-10,10]', fontsize=16)
10 # 设置x轴名称
11 plt.xlabel("x 轴", fontsize=12)
12 # 设置y轴名称
13 plt.ylabel("y 轴")
14 # 绘图线条1
15 plt.plot(x, y1)
16 # 绘图线条2
17 plt.plot(x, y2)

```

### 3. 设置x轴和y轴的刻度

```
matplotlib.pyplot.xticks(ticks=None, labels=None, **kwargs)
```

- ticks: 此参数是xtick位置的列表。和一个可选参数。如果将一个空列表作为参数传递，则它将删除所有xticks
- labels: 此参数包含放置在给定刻度线位置的标签。它是一个可选参数。
- \*\*kwargs: 此参数是文本属性，用于控制标签的外观
- rotation: 旋转角度 如:rotation=45
- color: 颜色 如:color="red"



xticks到底有什么用，其实就是想把坐标轴变成自己想要的样子

```
1 # 每个时间点的销量绘图
2 times = ['2015/6/26', '2015/8/1', '2015/9/6', '2015/10/12',
3          '2015/11/17', '2015/12/23', '2016/1/28', '2016/3/4', '2016/4/9',
4          '2016/5/15', '2016/6/20', '2016/7/26', '2016/8/31', '2016/10/6', '2016/
5          11/11', '2016/12/17']
```

```
5 # 随机出销量
6 sales = np.random.randint(500,2000,size=len(times))
7
8 # 绘制图形
9 plt.plot(times,sales)
```

```
1 # 以上图形x轴存在的问题?
2
3 # 如果想只显示部分时间,或者按照某个规则展示,如何处理,这个时候就用到xticks
4 plt.xticks(range(1,len(times),2))
5 plt.plot(times,sales)
```

```
1 # 时间虽然少了,可不可以让横坐标中的字体倾斜,就可以放下了.
2 plt.xticks(range(1,len(times),2),rotation=45)
3 plt.plot(times,sales)
```

```
1 # 时间虽然少了,可不可以让横坐标中的字体倾斜,就可以放下了.
2 plt.xticks(range(1,len(times),2),labels=[1,2,3,4,],rotation=45)
3 plt.plot(times,sales)
```

```
1 # 创建x为-10到10的整数
2 x = np.arange(-50,51)
3 # y是x的平方
4 y = x ** 2
```

```

5 # 设置标题
6 plt.title('y=x^2 x取值范围为: [-10,10]', fontsize=16)
7 # 设置x轴名称
8 plt.xlabel("x 轴", fontsize=12)
9 # 设置y轴名称
10 plt.ylabel("y 轴")
11 x_ticks = [-50, -40, -30, -20, -10, 0, 10, 20, 30, 40, 50]
12 x_labels = ['%s度'%i for i in x_ticks]
13 plt.xticks(x_ticks, x_labels)
14 # 画图
15 plt.plot(x, y)

```

```
1
```

## 4.显示图表show()

- 显示所有打开的图形。



jupyter notebooks会自动显示图形

```

1 # 此处通过python交互模式演示
2

```

```

1 # 如果在jupyter中也想出现图形操作菜单,可以使用matplotlib中的魔术方法
2 %matplotlib notebook

```

```

1 plt.xticks(range(1, len(times), 2), rotation=45)
2 plt.plot(times, sales)

```

```

1 # 如果有想回去原先的展示,使用另一个 %matplotlib inline
2 %matplotlib inline

```

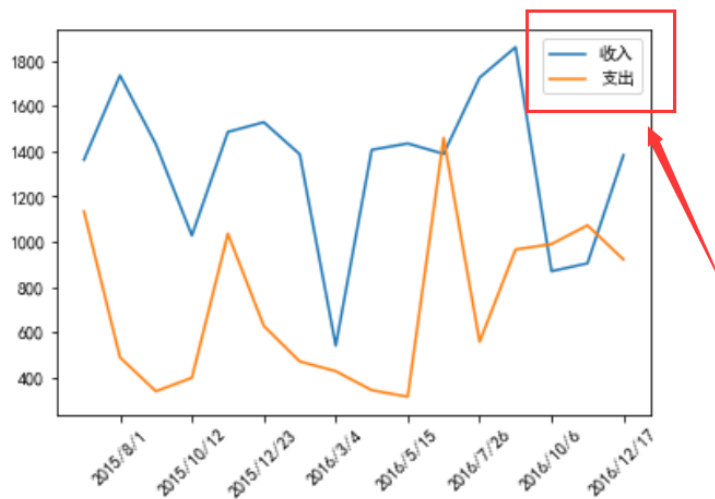
```

1 plt.xticks(range(1, len(times), 2), rotation=45)
2 plt.plot(times, sales)

```

## 5.图例 legend()

图例是集中于地图一角或一侧的地图上各种符号和颜色所代表内容与指标的说明，有助于更好的认识地图。



```

1 # 每个时间点的销量绘图
2 times = ['2015/6/26', '2015/8/1', '2015/9/6', '2015/10/12',
3         '2015/11/17', '2015/12/23', '2016/1/28', '2016/3/4', '2016/4/9',
4         '2016/5/15', '2016/6/20', '2016/7/26', '2016/8/31', '2016/10/6', '2016/11/11', '2016/12/17']
5
6 # 随机出收入
7 income = np.random.randint(500, 2000, size=len(times))
8 # 支出
9 expenses = np.random.randint(300, 1500, size=len(times))
10
11 # 绘制图形
12 plt.xticks(range(1, len(times), 2), rotation=45)
13 # 注意, 在使用图例前为每个图形设置label参数
14 plt.plot(times, income, label="收入1")
15 plt.plot(times, expenses, label="支出1")
16 # 默认会使用每个图形的label值作为图例中的说明
17 plt.legend()

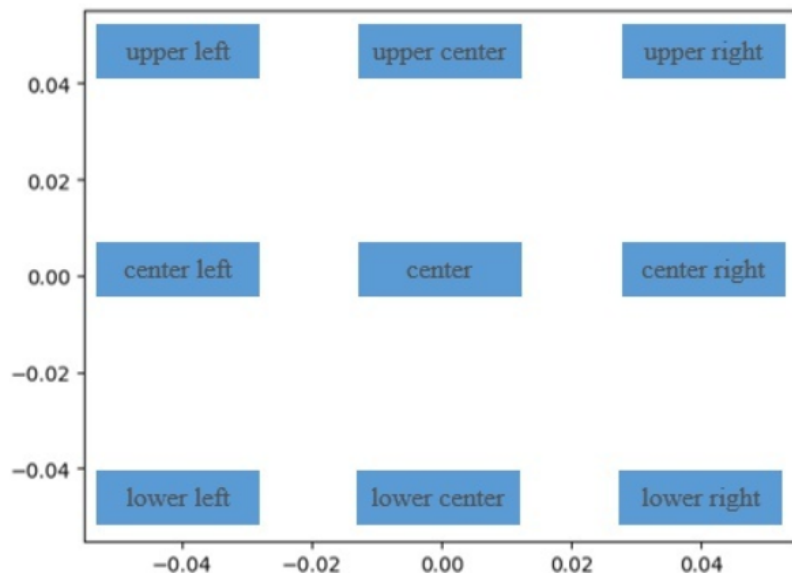
```

## 图例的图例位置设置

- loc代表了图例在整个坐标轴平面中的位置（一般选取'best'这个参数值）
- 第一种:默认是"best",图例自动‘安家’在一个坐标面内的数据图表最少的位置
- 第二种: loc = 'XXX' 分别有0: 'best'（自动寻找最好的位置）



位置字符串	位置值	备注
"best"	0	自动寻找最好的位置
"upper right"	1	右上角
"upper left"	2	左上角
"lower left"	3	左下角
"lower right"	4	右下角
"right"	5	右边中间
"center left"	6	左边中间
"center right"	7	右边中间
"lower center"	8	中间最下面
"upper center"	9	中间最上面
"center"	10	正中心



```

1 # 每个时间点的销量绘图
2 times = ['2015/6/26', '2015/8/1', '2015/9/6', '2015/10/12',
3         '2015/11/17', '2015/12/23', '2016/1/28', '2016/3/4', '2016/4/9',
4         '2016/5/15', '2016/6/20', '2016/7/26', '2016/8/31', '2016/10/6', '2016/11/11', '2016/12/17']

```

```

5 # 随机出收入
6 income =np.random.randint(500,2000,size=len(times))
7 # 支出
8 expenses =np.random.randint(300,1500,size=len(times))
9
10 # 绘制图形
11 plt.xticks(range(1,len(times),2),rotation=45)
12 # 注意,在使用图例前为每个图形设置label参数
13 plt.plot(times,income,label="收入")
14 plt.plot(times,expenses,label="支出")
15 # 默认会使用每个图形的label值作为图例中的说明
16 plt.legend(loc="center")

```

```

1 <matplotlib.legend.Legend at 0x23f147830c8>

```

```

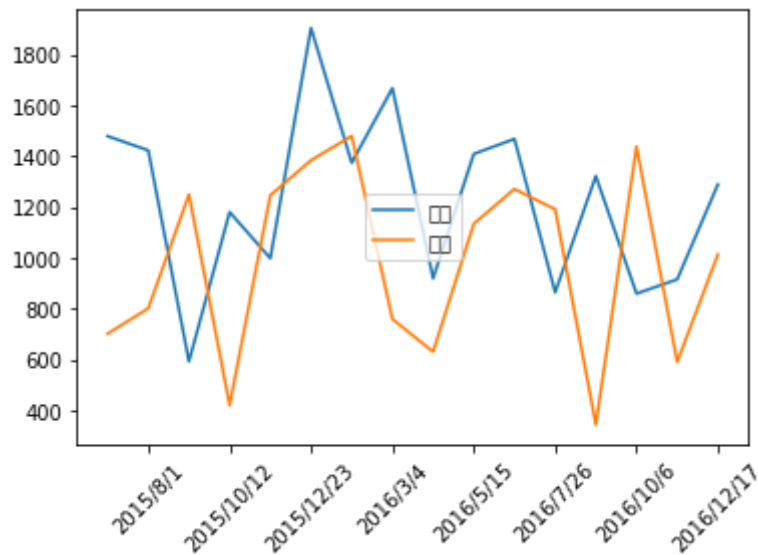
1 D:\Anaconda3\lib\site-
  packages\matplotlib\backends\backend_agg.py:211: RuntimeWarning:
  Glyph 25910 missing from current font.
2     font.set_text(s, 0.0, flags=flags)
3 D:\Anaconda3\lib\site-
  packages\matplotlib\backends\backend_agg.py:211: RuntimeWarning:
  Glyph 20837 missing from current font.
4     font.set_text(s, 0.0, flags=flags)
5 D:\Anaconda3\lib\site-
  packages\matplotlib\backends\backend_agg.py:211: RuntimeWarning:
  Glyph 25903 missing from current font.
6     font.set_text(s, 0.0, flags=flags)
7 D:\Anaconda3\lib\site-
  packages\matplotlib\backends\backend_agg.py:211: RuntimeWarning:
  Glyph 20986 missing from current font.
8     font.set_text(s, 0.0, flags=flags)
9 D:\Anaconda3\lib\site-
  packages\matplotlib\backends\backend_agg.py:180: RuntimeWarning:
  Glyph 25910 missing from current font.
10    font.set_text(s, 0, flags=flags)
11 D:\Anaconda3\lib\site-
  packages\matplotlib\backends\backend_agg.py:180: RuntimeWarning:
  Glyph 20837 missing from current font.
12    font.set_text(s, 0, flags=flags)
13 D:\Anaconda3\lib\site-
  packages\matplotlib\backends\backend_agg.py:180: RuntimeWarning:
  Glyph 25903 missing from current font.

```

```

14     font.set_text(s, 0, flags=flags)
15     D:\Anaconda3\lib\site-
packages\matplotlib\backends\backend_agg.py:180: RuntimeWarning:
Glyph 20986 missing from current font.
16     font.set_text(s, 0, flags=flags)

```



## 显示每条数据的值 x,y值的位置

```

plt.text(x,y, string, fontsize=15, verticalalignment="top",
horizontalalignment="right")

```

- x,y:表示坐标值上的值
- string:表示说明文字
- fontsize:表示字体大小
- verticalalignment: (va)垂直对齐方式，参数: [ 'center' | 'top' | 'bottom' | 'baseline' ]
- horizontalalignment: (ha)水平对齐方式，参数: [ 'center' | 'right' | 'left' ]

```

1     # 每个时间点的销量绘图
2     times = ['2015/6/26', '2015/8/1', '2015/9/6', '2015/10/12',
'2015/11/17', '2015/12/23', '2016/1/28', '2016/3/4', '2016/4/9',
3     '2016/5/15', '2016/6/20', '2016/7/26', '2016/8/31', '2016/10/6', '2016
/11/11', '2016/12/17']

```

```

4
5 # 随机出收入
6 income =np.random.randint(500,2000,size=len(times))
7 # 支出
8 expenses =np.random.randint(300,1500,size=len(times))
9
10 # 绘制图形
11 plt.xticks(range(1,len(times),2),rotation=45)
12 # 注意,在使用图例前为每个图形设置label参数
13 plt.plot(times,income,label="收入")
14 plt.plot(times,expenses,label="支出")
15 # 默认会使用每个图形的label值作为图例中的说明
16 plt.legend(loc="upper left")
17 for x,y in zip(times,income):
18     plt.text(x,y,'%s万'%y)
19
20 #for a,b in zip(times,expenses):
21 #    plt.text(a,b,b)

```

```

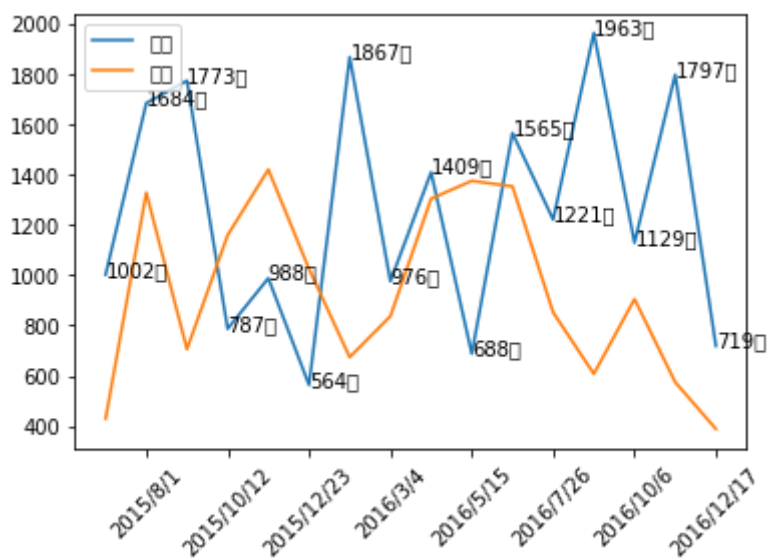
1 D:\Anaconda3\lib\site-
  packages\matplotlib\backends\backend_agg.py:211: RuntimeWarning:
  Glyph 19975 missing from current font.
2     font.set_text(s, 0.0, flags=flags)
3 D:\Anaconda3\lib\site-
  packages\matplotlib\backends\backend_agg.py:180: RuntimeWarning:
  Glyph 19975 missing from current font.
4     font.set_text(s, 0, flags=flags)
5 D:\Anaconda3\lib\site-
  packages\matplotlib\backends\backend_agg.py:211: RuntimeWarning:
  Glyph 25910 missing from current font.
6     font.set_text(s, 0.0, flags=flags)
7 D:\Anaconda3\lib\site-
  packages\matplotlib\backends\backend_agg.py:211: RuntimeWarning:
  Glyph 20837 missing from current font.
8     font.set_text(s, 0.0, flags=flags)
9 D:\Anaconda3\lib\site-
  packages\matplotlib\backends\backend_agg.py:211: RuntimeWarning:
  Glyph 25903 missing from current font.
10    font.set_text(s, 0.0, flags=flags)
11 D:\Anaconda3\lib\site-
  packages\matplotlib\backends\backend_agg.py:211: RuntimeWarning:
  Glyph 20986 missing from current font.
12    font.set_text(s, 0.0, flags=flags)

```

```

13 D:\Anaconda3\lib\site-
   packages\matplotlib\backends\backend_agg.py:180: RuntimeWarning:
   Glyph 25910 missing from current font.
14     font.set_text(s, 0, flags=flags)
15 D:\Anaconda3\lib\site-
   packages\matplotlib\backends\backend_agg.py:180: RuntimeWarning:
   Glyph 20837 missing from current font.
16     font.set_text(s, 0, flags=flags)
17 D:\Anaconda3\lib\site-
   packages\matplotlib\backends\backend_agg.py:180: RuntimeWarning:
   Glyph 25903 missing from current font.
18     font.set_text(s, 0, flags=flags)
19 D:\Anaconda3\lib\site-
   packages\matplotlib\backends\backend_agg.py:180: RuntimeWarning:
   Glyph 20986 missing from current font.
20     font.set_text(s, 0, flags=flags)

```



1

# 作业

一:绘制曲线图,要求如下:

- 1.显示图形名称

- 2.为x轴和y轴定义说明信息
- 3.使用中文显示

二: 已知数据为2000年到2020年20年之间的销量情况.

- `sales = [109, 150, 172, 260, 273, 333, 347, 393, 402, 446, 466, 481, 499, 504, 513, 563, 815, 900, 930, 961]`

要求:

- 1.绘制年度销量线形图.x轴为年份,y轴为销量.
- 2.显示图例

1

1