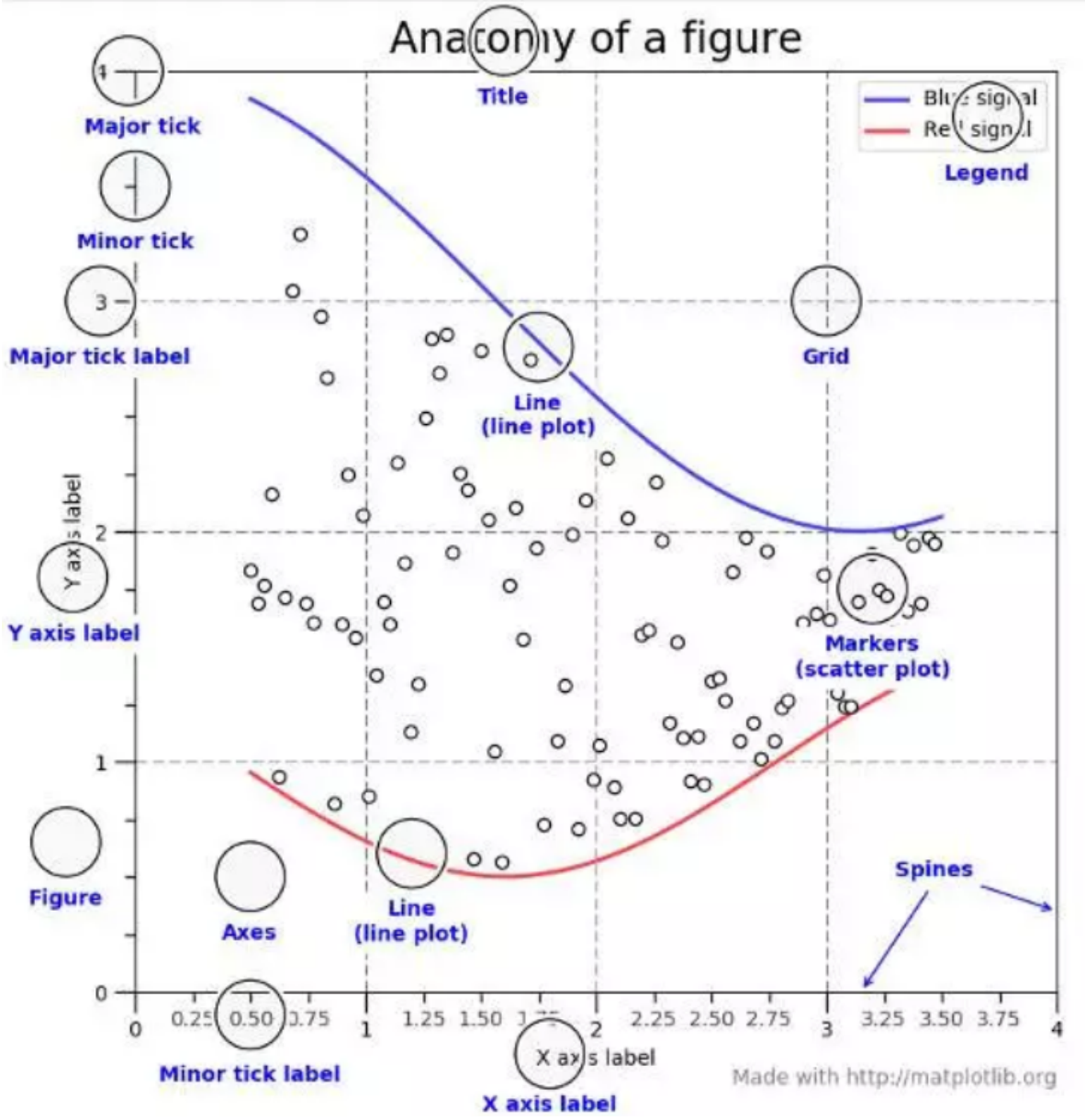
matplotlib——数据可视化

1.简介

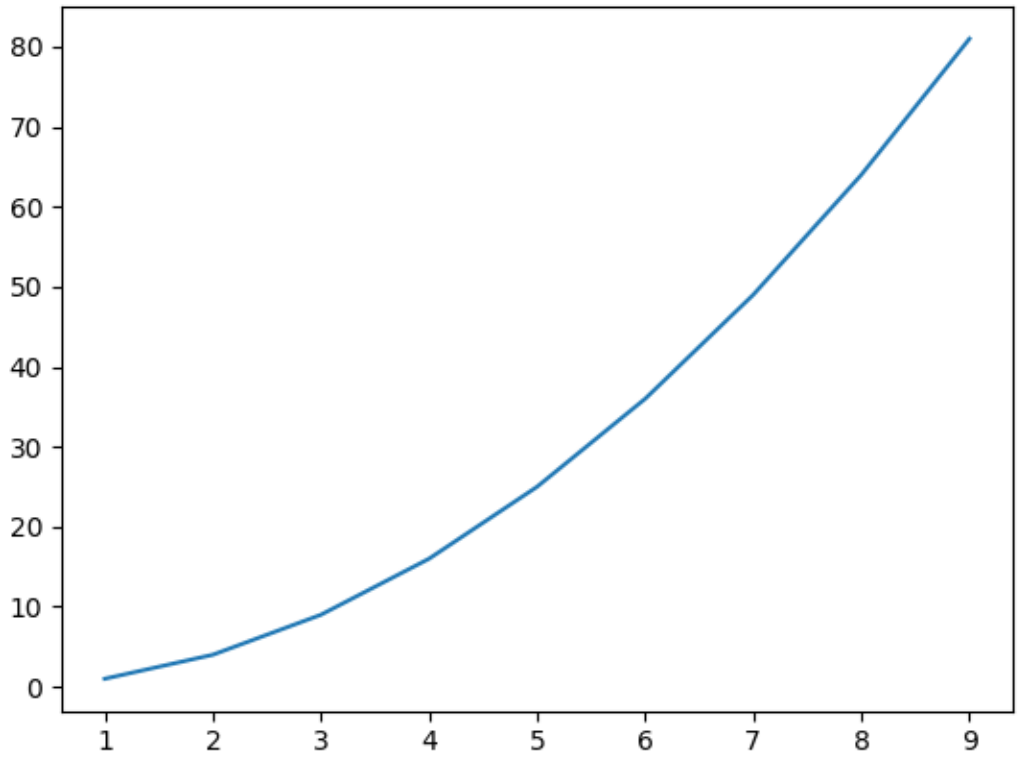
数据的处理、分析和可视化已经成为Python近年来最为重要的应用领域之一，其中数据的可视化指的是将数据呈现为漂亮的统计图表，然后进一步发现数据中包含的规律以及隐藏的信息。



2.安装与使用

- 安装：使用pip命令进行安装：`pip install matplotlib`
- 使用

```
from matplotlib import pyplot as plt

x=[x for x in range(1,10)]
y=[i**2 for i in x]
plt.plot(x,y)
plt.show()
```



3.参数详解

- 图片大小（绘图函数前设置）：`fig=plt.figure(figsize=(20,8),dpi=80)`
- 图片保存（绘图函数后设置）：`plt.savefig("路径")`
- 标题：`plt.title("")`
- 坐标轴标记

  `plt.xlabel("")`

  `plt.ylabel("")`
- 坐标轴刻度

  `plt.xticks(x)`

  `plt.yticks(y)`
- **字符串刻度**：

  `xtick=["{}".format(i) for i in range(x)]`

  `plt.xticks(x,xtick)`

  注：若刻度不需要全部显示，在设置xticks时，可以通过将列表切片xtick[::h]设定步长实现部分显示，此时需要将x做同样的切片

  x为指定刻度，xtick指定刻度上的文字

  `plt.xticks(x[::h],xtick[::h])`
- 坐标轴刻度旋转：`plt.xticks(rotation=45)`
- **坐标刻度显示中文**

```
import matplotlib.pyplot as plt
import random
import matplotlib
from matplotlib import font_manager

#这个字体设置为全局设置,只用在该处修改即可(windows和linux下有效)
# font = {'family' :"Microsoft Yahei",
#         'size'   : '10'}
# matplotlib.rc("font",**font)

……
……

#设置中文字体(指定具体的字体文件路径,然后在需要显示中文的地方添加fontproperties参数)
my_font = font_manager.FontProperties(fname="/System/Library/Fonts/PingFang.ttc")

plt.xticks(x[::5],_x_ticks[::5],rotation=90,fontproperties=my_font)
plt.show()
```

```
from matplotlib import font_manager
my_font=font_manager.FontProperties(fname="C:/Windows/Fonts/msyh.ttc")
在需要显示中文的方法(xlabel,xticks,title...)中添加缺省参数：fontproperties=my_font
在图例中（legend）添加缺省参数：prop=myfont
```

注意：图例的中文字体属性用prop，其余均用fontproperties

- 网格：`plt.grid(alpha=0.5)` alpha为缺省参数
- **图例**

  `plot(x,y,label="")`

  `plt.legend([prop=myfont,loc="upper left"])`

  同一个图中画多条曲线，直接多次plot即可，同时在plot中的label参数指定图例信息

  图例中文字体属性**prop**，其他中文属性**fontproperties**

  图例位置loc：upper/lower left/right/center

- **绘图函数的缺省参数**

  abel=""图例信息

  color="r"线条颜色

  linestyle='--'线形

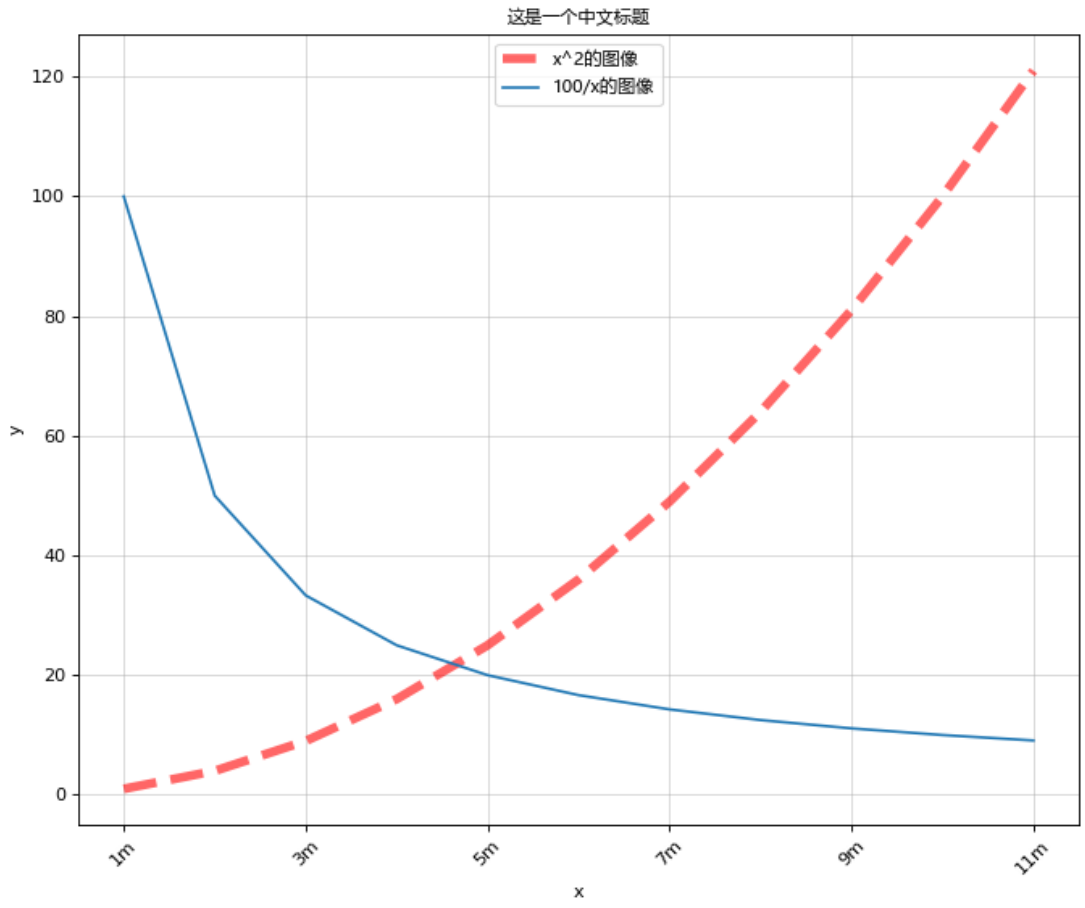
  linewidth=5线条粗细

  alpha=0.5透明度

颜色字符		**线型字符**
r红色	c青色	-实线
g绿色	m洋红色	--虚线、破折线
b蓝色	y黄色	-.点划线
w白色	k黑色	： 虚线、点虚线
		''留空或空格，无线条

```
# coding=utf-8
```

```
from matplotlib import pyplot as plt
from matplotlib import  font_manager

x=[x for x in range(1,12)]
y=[i**2 for i in x]
y2=[100/i for i in x]

# 中文字体
myfont=font_manager.FontProperties(fname="C:/Windows/Fonts/msyh.ttc")

plt.figure(dpi=80,figsize=(10,8))
# plt.title("fig1.1")
plt.title("这是一个中文标题",fontproperties=myfont)
plt.xlabel('x')
xtick=["{}m".format(i) for i in x]
plt.xticks(x[::2],xtick[::2])
plt.xticks(rotation=45)
plt.ylabel('y')
plt.grid(alpha=0.5)

plt.plot(x,y,label='x^2的图像',color='r',linestyle='--
',linewidth=5,alpha=0.6)
plt.plot(x,y2,label='100/x的图像')
plt.legend(prop=myfont,loc="upper center")
# plt.savefig("plt.jpg")
plt.show()
```

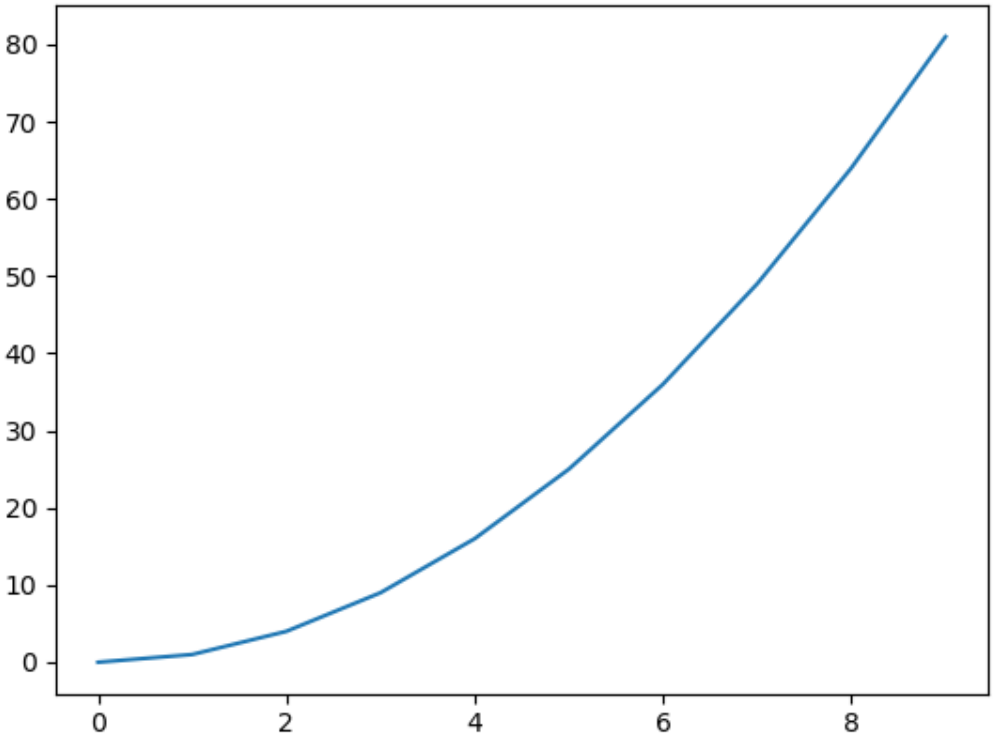


4.常用统计图

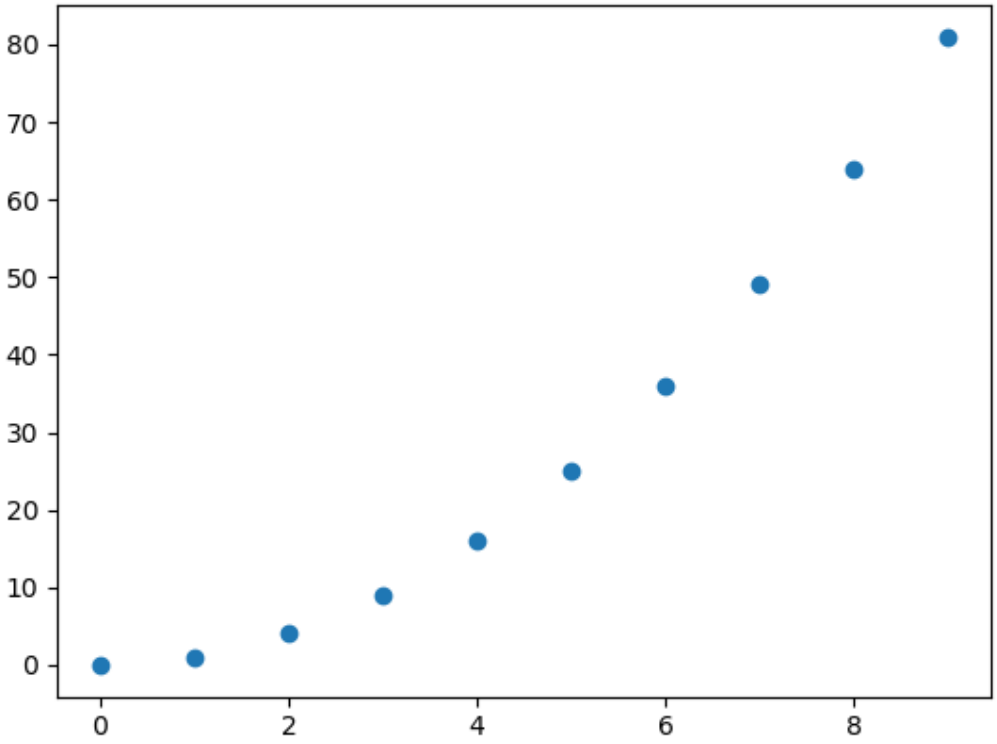
- 参数调整同plot

```
import matplotlib.pyplot as plt

x=[i for i in range(10)]
y=[j**2 for j in x]

plt.fun(x,y)
plt.show()
```

- 折线图
  - plot

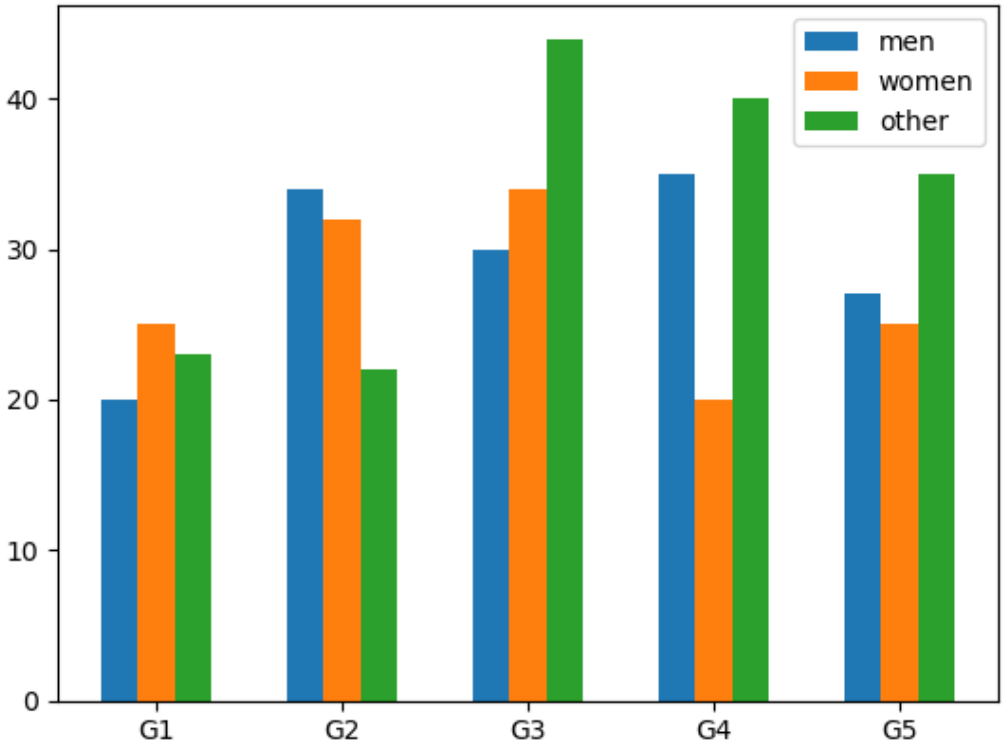


- 散点图
  - scatter



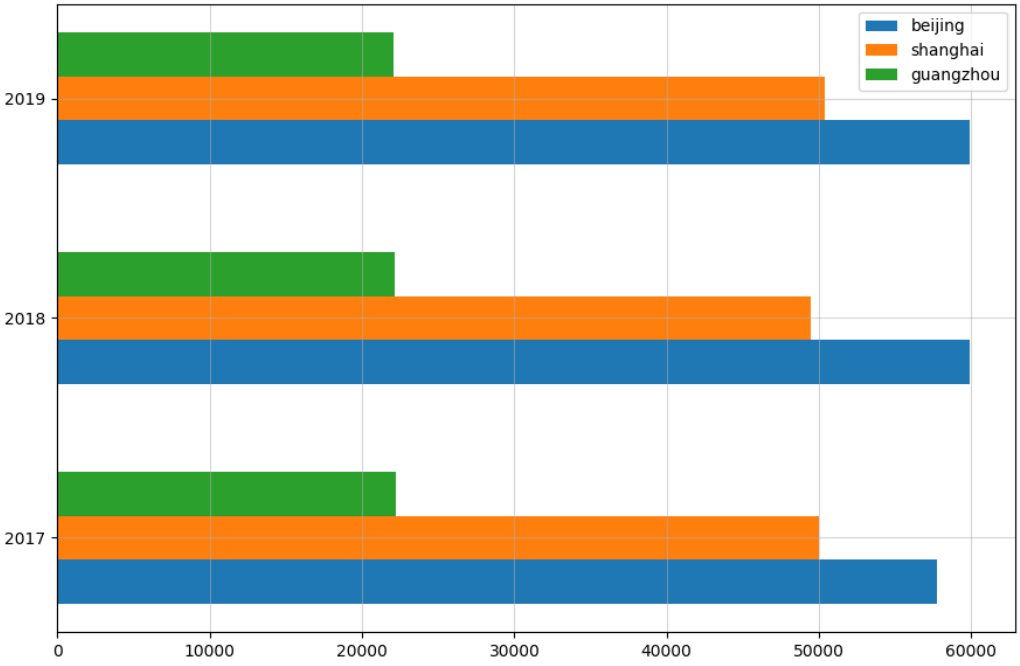
- 纵向条形图
  - 离散数据
  - `plt.bar(x,y,width=0.4)`
  - 绘制多个条形图通过多次bar即可，但是需要将偏移width，xtick和x都需要调整
  - m*n列的数据适合绘制条形图，类内和类间均可对比

```
import matplotlib.pyplot as plt

x=[i for i in range(1,6)]
label=["G{}".format(i) for i in x]
width=0.2
men_x=[i-width for i in x]
women_x=x
other_x=[i+width for i in x]
men_means = [20, 34, 30, 35, 27]
women_means = [25, 32, 34, 20, 25]
other_means=[23, 22, 44, 40, 35]

plt.bar(men_x,men_means,width=width,label="men")
plt.bar(women_x,women_means,width=width,label="women")
plt.bar(other_x,other_means,width=width,label="other")
plt.xticks(x,label)
plt.legend()

plt.show()
```



- 横向条形图
  - 离散数据，统计数据
  - `plt.barh(x,y,height=0.3)`

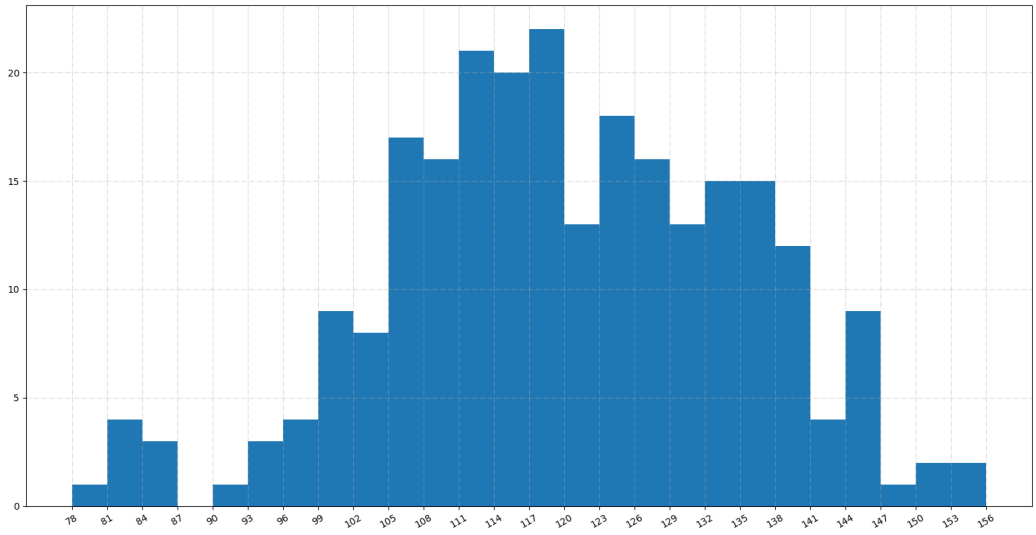
```
import matplotlib.pyplot as plt
height=0.2
x=[i for i in range(3)]
year=[i for i in range(2017,2020)]
print(year)
beijing=[57768,59868,59906]
shanghai=[50017,49446,50420]
guangzhou=[22205,22188,22055]

plt.barh([i-height for i in x],beijing,height=height,label="beijing")
plt.barh(x,shanghai,height=height,label="shanghai")
plt.barh([i+height for i in
x],guangzhou,height=height,label="guangzhou")
plt.yticks(range(len(year)),year)
plt.legend()
plt.grid(alpha=0.5)

plt.show()
```



- 直方图
  - 连续数据，原始数据
  - 频数直方图：`plt.hist(data,n)`
  - 频率直方图：`plt.hist(data,n, normed=True)`

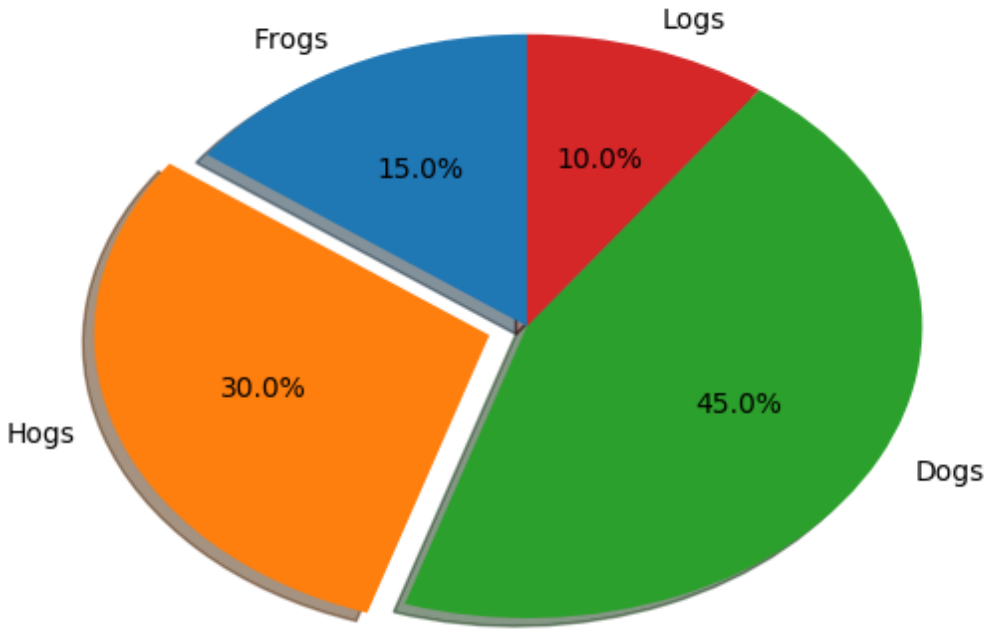
```
# coding=utf-8

'''
Author:Regent Wan
Email:wanrunjun@gmail.com
微信公众号：学术创客
Github:https://github.com/RegentWan

Date:2019/8/29 11:19
Desc:
'''

import matplotlib.pyplot as plt

# 直方图
time=
[131,98,125,131,124,139,131,117,128,108,135,138,131,102,107,114,119,128
,121,142,127,130,124,101,116,117,110,128,128,115,99,136,126,134,95,138,
117,111,78,132,124,113,150,110,117,86,95,144,105,126,130,126,130,126,11
6,123,106,112,138,123,86,101,99,136,123,117,119,105,137,123,128,125,104
,109,134,125,127,105,120,107,129,116,108,132,103,136,118,102,120,114,10
5,115,132,145,119,121,112,139,125,138,109,132,134,156,106,117,127,144,1
39,139,119,140,83,110,102,123,107,143,115,136,118,139,123,112,118,125,1
09,119,133,112,114,122,109,106,123,116,131,127,115,118,112,135,115,146,
137,116,103,144,83,123,111,110,111,100,154,136,100,118,119,133,134,106,
129,126,110,111,109,141,120,117,106,149,122,122,110,118,127,121,114,125
,126,114,140,103,130,141,117,106,114,121,114,133,137,92,121,112,146,97,
137,105,98,117,112,81,97,139,113,134,106,144,110,137,137,111,104,117,10
0,111,101,110,105,129,137,112,120,113,133,112,83,94,146,133,101,131,116
,111,84,137,115,122,106,144,109,123,116,111,111,133,150]
binWidth=3
```

```
binNum=int((max(time)-min(time))/binWidth)
plt.hist(time,binNum)
plt.xticks((range(min(time),max(time)+binWidth))
[::binWidth],rotation=30)
plt.grid(linestyle="-.",alpha=0.5)

plt.show()
```



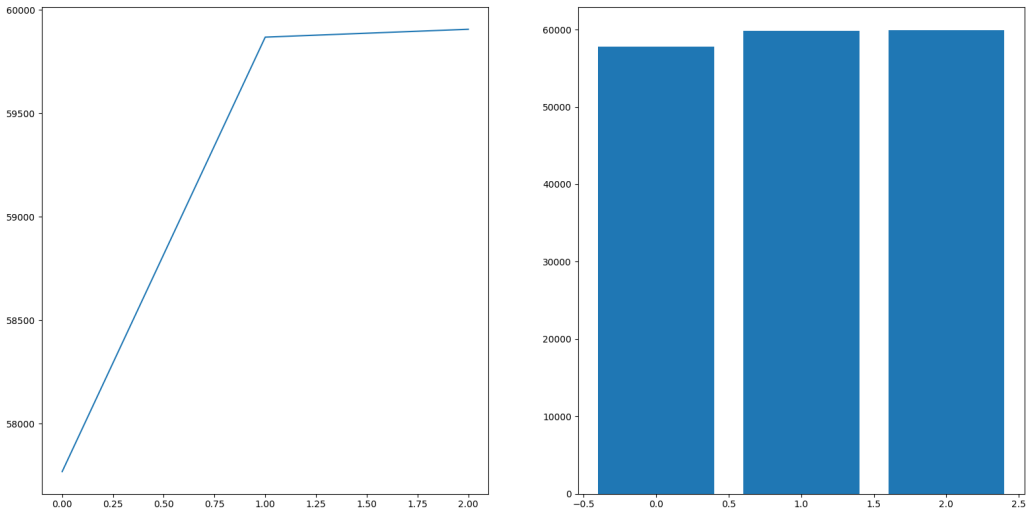
- 饼图
  - pie(size,explode=,labels=,startangle=)
  - 参数：各部分比例、explode突出部分、label各部分标记、startangle起始角度

```
import matplotlib.pyplot as plt

labels = 'Frogs', 'Hogs', 'Dogs', 'Logs'
sizes = [15, 30, 45, 10]
explode = (0, 0.1, 0, 0)
plt.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
shadow=True, startangle=90)

plt.show()
```



- 子图
  - 用plt.subplot(row, col, num)来指定子图位置
  - 各子图样式分别设置

```
import matplotlib.pyplot as plt

data=[57768,59868,59906]
plt.subplot(1,2,1)
plt.plot(range(len(data)),data)

plt.subplot(1,2,2)
plt.bar(range(len(data)),data)

plt.show()
```



- 等高线图
- 更多图形参考https://matplotlib.org/gallery/index.html