# CHAPTER 3

# 数据分析和可视化

主讲人：吴春彪

2025年5月18日

# 01

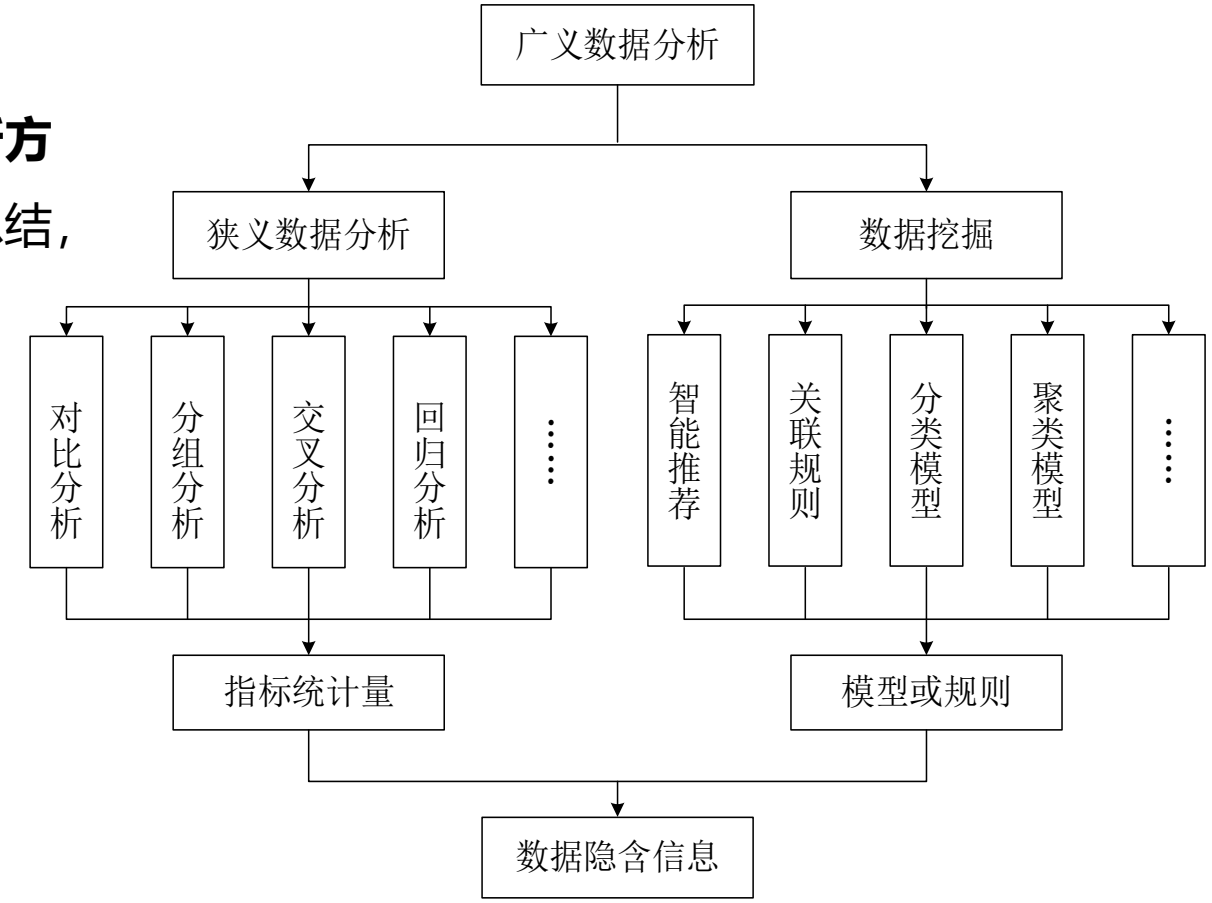
# 数据分析生态

# 数据分析生态

## 一、数据分析的概念

- 数据分析（Data Analysis）：采用合适的**统计分析方法**对收集来的海量的历史数据进行分析、概括和总结，**揭示数据隐藏的规律**，提取有效信息。

- 随着计算机技术的全面发展，企业生产、收集、存储和处理数据的能力大大提高，数据量与日俱增。而在现实生活中，需要将这些繁多、复杂的数据通过统计分析进行提炼，以此研究出数据的发展规律，进而帮助企业管理层做出决策。

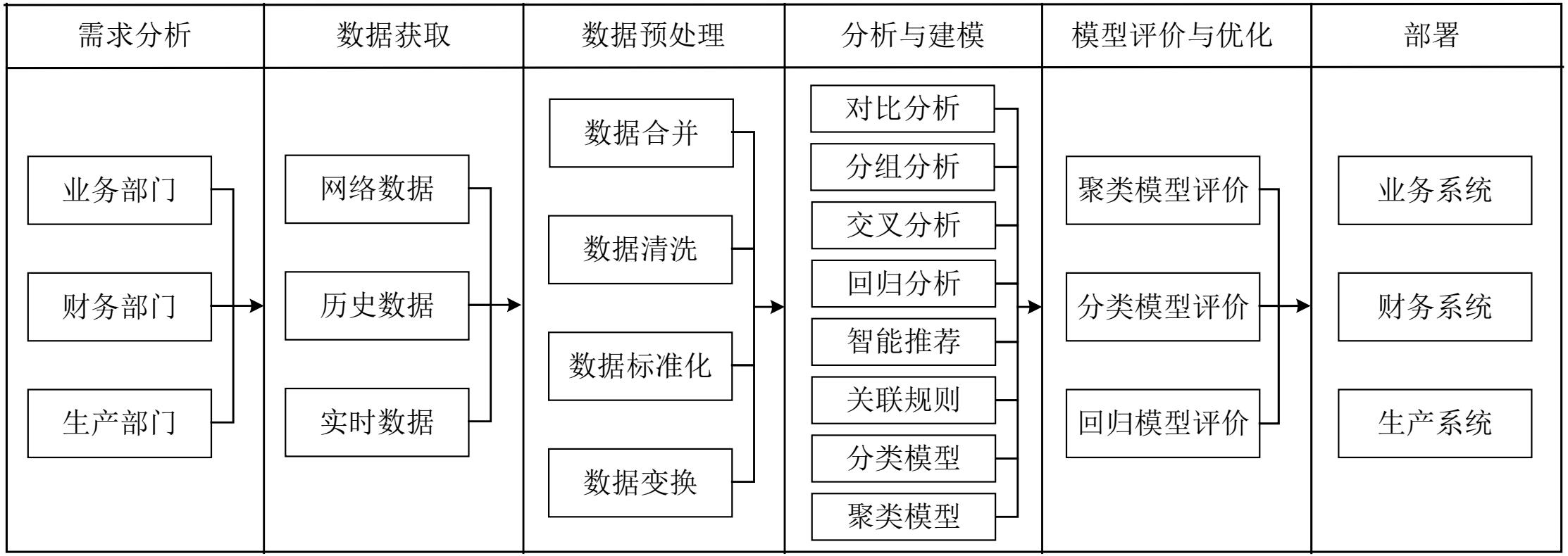


# 数据分析生态

## 二、数据分析的流程

- 数据分析已经逐渐演化为一种解决问题的求解过程/方法论。

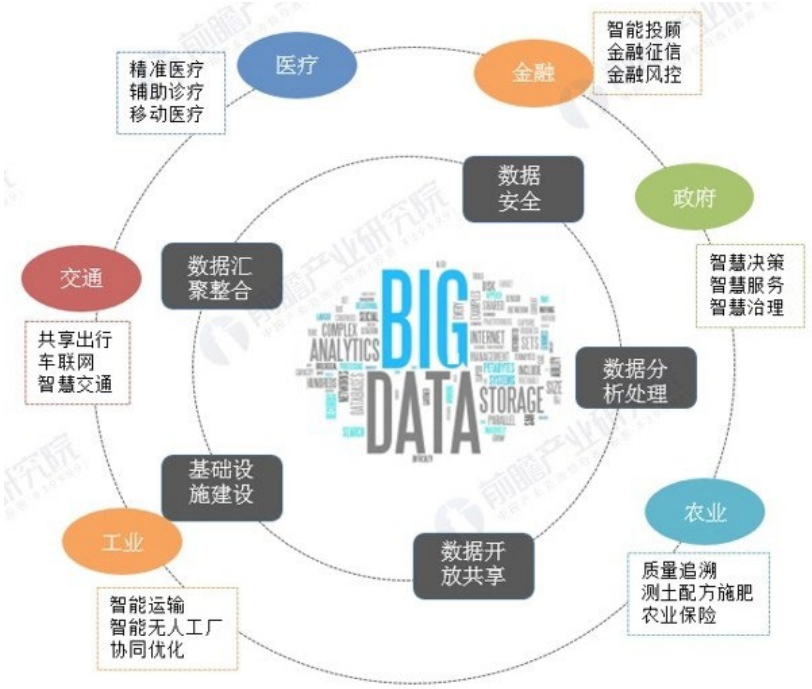
数据获取 →数据清洗 →数据统计 →数据可视化 →数据挖掘 →人工智能



# 数据分析生态

## 三、数据分析应用场景

- 企业使用数据分析解决不同的问题，实际应用的数据分析场景主要分为客户分析、营销分析、社交媒体分析、网络安全、设备管理、交通物流分析和欺诈行为检测等诸多领域。

# 数据分析生态

## 四、数据分析生态

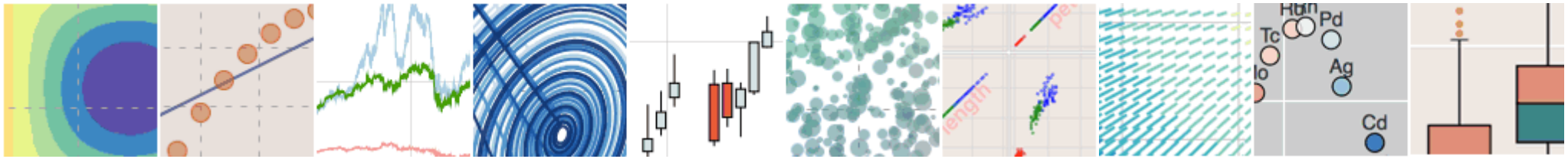
- <数据分析库>：Numpy、 Pandas、 SciPy

① Numpy： N维数组/矩阵存储和处理

② Pandas：提供数据分析、数据挖掘、数据清洗

③ SciPy： 提供数学算法和工程数据运算（优化算法、傅里叶变换、 信号处理等应用）

- <数据可视化库>：Matplotlib、 Seaborn、 Mayavi




- <机器学习库>：Scikit-learn、 TensorFlow、 Pytorch

① Scikit-learn： 提供聚类、分类、回归、强化学习等计算功能

② TensorFlow/Pytorch： 基于Python的深度学习框架（DNN 、 CNN 、 RNN）

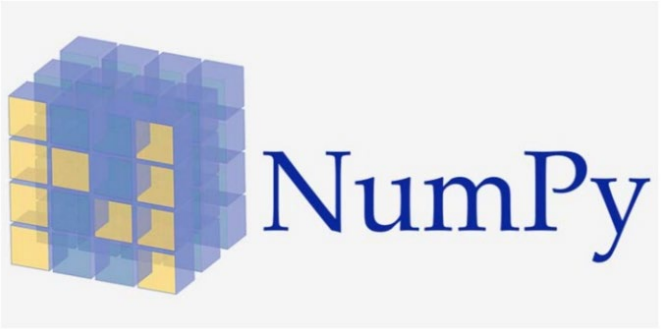
# 02

# 科学计算库：Numpy

# 科学计算库：Numpy

■ 多维数组： Numpy库 → 科学计算

- Numpy 库是Python中用于科学计算的基础工具包，提供用于数组进行快速操作的多维数组对象，拥有大量用于数据分析/处理的函数和方法，高效完成数学运算、逻辑运算、排序、选择、I/O （输入/输出）、傅里叶变换、线性代数、统计运算、随机模拟等操作




- <官网>：https://numpy.org/
- <安装>：anaconda模式 → conda install numpy
- <调用>：import numpy as np # 缩写模式
- Numpy是数据分析计算生态的基础库，是Pandas 、Matplotlib等支撑依赖库

# 科学计算库：Numpy

## ■ 数组对象 --- ndarray

- 数组对象ndarray是Numpy用来存储若干数据的数据存储器，实现数据的批量运算。和其他的编程语言一样，Python要求数组中的每个元素的类型相同。
- ndarray是同种元素的一维或多维数组对象

- 为什么使用 ndarray?

ndarray和Python内置列表数据类型类型，但ndarray的计算效率明显优于列表。

| |
|---|
| ① 列表100000个元素平均排序时间：0.05611551秒<br>② ndarray 数组100000个元素平均排序时间：0.01017438秒 |
| ① 列表100000个元素平均求和时间：0.00557586秒<br>② ndarray 数组100000个元素平均求和时间：0.00012212秒 |

# 科学计算库：Numpy

- 创建ndarray数组的方法：

| 函数 | 说明 |
| --- | --- |
| **array()** | 将输入数据（列表、元组等序列类型）转换为一维或多维数组 |
| **arange()** | 类似于range函数，返回一个一维数组，*arange([start, ]stop, [step, ])* |
| **linspace()** | 设置起始和结束区间，均匀地产生指定个数的数字，组成一维数组<br>*linspace ( start, stop, num= )* |
| **zeros((m, n))** | 创建一个m行n列全1的二维数组(矩阵)，dtype控制数据类型 |
| **ones((m, n))** | 创建一个m行n列全0的二维数组(矩阵)，dtype控制数据类型 |
| **empty()** | 空数组，只申请空间，不初始化 |
| **diag()** | 创建对角矩阵　np.diag([1,2,3]) □ 对角是1-2-3的矩阵 |
| **identity()** | 创建单位矩阵　np.diag(4) □ 4×4 单位矩阵 |

# 科学计算库：Numpy

## ndarray数组基本属性

- ndarray是同种元素的一维或多维数组对象，拥有如下基本属性：

| 属性名称 | 属性说明 |
| --- | --- |
| ndim | 返回int，表示数组的维数 |
| shape | 返回tuple，表示数组形状的阵列，对于n行m列的矩阵，形状为(n,m) |
| size | 返回int，表示数组的元素总数，等于数组形状的乘积 |
| dtype | 返回data-type，表示数组中元素的数据类型 |

# 科学计算库：Numpy

## ■ Numpy 生成随机数

- NumPy中，与随机数相关的函数都在random模块，包括生成服从多种概率分布随机数的函数。

| 函数 | 描述 |
| --- | --- |
| np.ramdom.random() | 生成指定形状的[0, 1)区间内的的随机数 |
| np.ramdom.rand() | 生成指定形状的[0, 1)区间内服从均匀分布的随机数 |
| np.ramdom.randn() | 生成指定形状的[0, 1)区间内服从正态分布的随机数 |
| np. ramdom.randint(a,b,size=[2,3]) | 生成一个2× 3的[a , b]区间内的随机整数。 |
| np.ramdom.normal() | 生成服从正态分布的随机数 |

# 科学计算库：Numpy

## ■ ndarray数组索引访问

- Numpy中通过数组的索引和切片进行数据元素的选取。

### 1. 一维数组的索引

➢ 一维数组的方法很简单，与Python中的list的索引方法一致。

arr [a : b ]

### 2. 多维数组的索引

➢ 多维数组的每一个维度都有一个索引，各个维度的索引之间用逗号隔开。

➢ 多维数组同样也可以使用整数序列和布尔值索引进行访问。

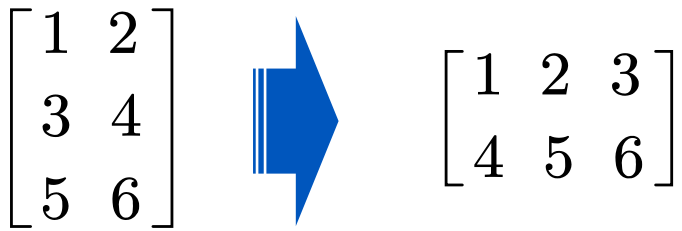
arr [ a : b , 1 : 2]

# 科学计算库：Numpy

## ndarray数组维度/形状变化

- 在NumPy中，常用reshape函数改变数组的“形状”，即改变数组的维度。
- reshape函数在改变原始数据的形状的同时不改变原始数据的值。如果指定的维度和数组的元素数目不吻合，那么函数将抛出异常。

- ndarrary. reshape (m.n)： 改变现有数组的维度，返回维度(m , n)数组。



# 科学计算库：Numpy

## ■ ndarray数组元素的修改

- Numpy中提供了多种修改数组元素值的方式。

（1）用下标的方式直接修改数组中一个或者多个元素的值。

（2）append() 函数是在数组末尾追加元素并返回新数组；

（3）insert () 函数是插入元素并返回新数组，可以设定插入的位置。

注意：append() 函数和 insert() 函数都是生成新数组，不会改变原数组。

# 科学计算库：Numpy

## ■ ndarray数组元素的排序

- Numpy的sort()方法和argsort()方法可以实现数组的排序。
- sort()方法直接对**原数组**进行排序，该方法**改变原始数组**。
- argsort()方法返回一个**新数组**，其中的每个元素是原数组中**元素的索引**，**不改变原始数组**。
- Numpy中还提供了argmax()和argmin()方法，用来返回数组中**最大值和最小值元素的下标**。

# 科学计算库：Numpy

## ndarray数组运算

- Numpy的数组支持与标量的加、减、乘、除和幂运算，计算结果为一个新数组，新数组的每个元素为标量与原数组中每个元素运算的结果。
- 两个等长的数组进行算术运算，可以得到一个新数组，新数组的元素为两数组对应位置的元素进行算术运算的结果。
- Numpy中还提供了dot()函数计算两个数组的内积（即两个等长数组中对应位置元素的乘积之和），

# 科学计算库：Numpy

- ndarray数组统计函数

| 函数 | 说明 |
| --- | --- |
| **sum** | 对数组中全部或某轴向的元素求和。零长度的数组的sum为0 |
| **mean** | 算术平均数。零长度的数组的mean为NaN |
| **std、var** | 计算标准差、方差，自由度可调 |
| **max、min** | 计算最大值、最小值 |
| **argmax、argmin** | 计算最大值、最小值的索引 |
| **cumsum** | 计算所有元素的累计和 |
| **cumprod** | 计算所有元素的累计积 |

# 03

# 数据分析库： Pandas

# 数据分析库： Pandas

## ■ 数据分析： Pandas库

- Pandas 库是基于Numpy的开源数据分析库，提供了快速、灵活的数据结构，旨在方便且直观地处理关系型和标记型数据，主要是执行数据清洗、数据分析等功能。
- Pandas是字典形式理解数据类型与索引的关系，操作索引即操作数据。
- 掌握Pandas提供的数据结构和数据分析工具

- <官网>：https://pandas.pydata.org/
- <安装> anaconda模式 → conda install pandas
- <调用>：import pandas as pd # 缩写模式

# 数据分析库： Pandas

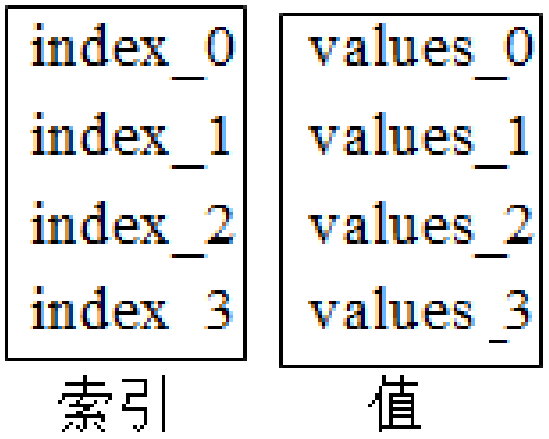
## ■ Pandas的数据结构

- Pandas库提供了一种扩展的数据类型，关注于数据与索引之间的关系，索引项是Pandas数据类型最独特的地方
- Pandas常用的数据类型包括表示一维数组结构的Series类型和二维数据结构的DataFrame类型。核心功能是执行各种操作，如增删、修改、数值计算等

# 数据分析库： Pandas

## （1）Series类型数组

- Series类型是Pandas提供的**一维数组**结构，由数据和与之相关的索引组成，其结构如图所示。




- Series类型是每个元素都有一个标签的一维表格，兼具列表和字典特点。

# 数据分析库： Pandas

## （1）Series类型数组

- Series数组的创建。使用列表、标量值、字典、ndarray对象等数据创建Series一维数组。

① 自动生成索引。在创建Series数组时如果没有给定索引，则可以自动生成从0开始的非负整数作为索引。

② 自定义索引。在创建Series数组时可以使用index参数自行设定索引。

③ 使用标量值创建Series数组。

④ 使用字典创建Series数组。

⑤ 使用ndarray对象创建Series数组。

# 数据分析库： Pandas

## （1）Series类型数组

- Series类型的常用操作。包括Series数据访问、数据的运算和对齐操作。

① Series类型由索引index和值values两部分组成，可以使用“对象名.index”获取Series对象的索引部分，使用“对象名.values”获取Series对象的数据部分。

② 使用索引访问Series对象

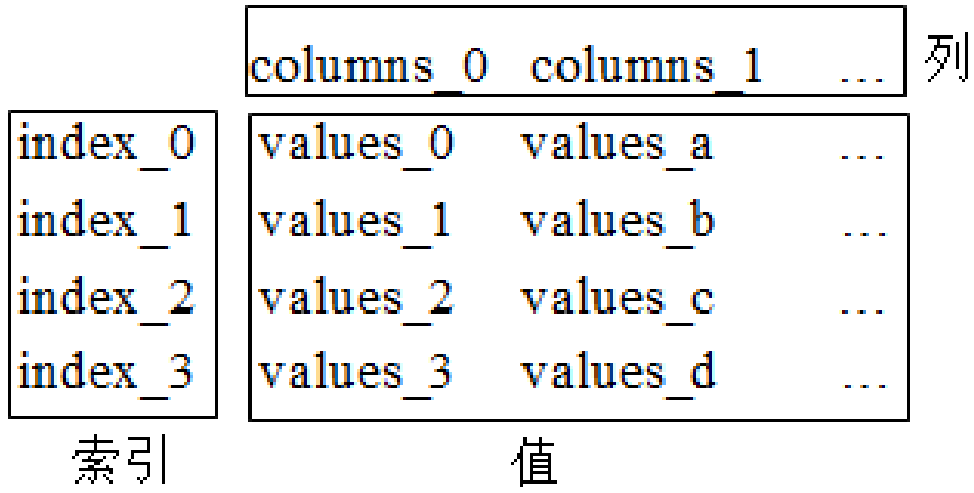
③ 使用Python内置函数或Series本身提供的方法操作Series对象。

④ Series类型的对齐操作

# 数据分析库： Pandas

## （2）DataFrame 类型数组

- DataFrame类型表示**二维数据**，可以看作一个**二维表格**，其结构由三部分组成：**索引index、列columns和值values**，如图所示。



# 数据分析库： Pandas

## （2）DataFrame 类型数组

- Pandas支持两种形式创建DataFrame类型数据：一是使用代码直接创建DataFrame数据，二是从外部数据读入到DataFrame类型

① 使用字典创建
② ndarray对象创建
③ 外部数据导入

| 数据类型 | 数据文件 | 读取方法 | 保存方法 |
| --- | --- | --- | --- |
| text | CSV | read _csv | to _ csv |
| text | JSON | read _json | to_json |
| text | XML | read _xml | to _xml |
| text | HTML | read _html | to _html |
| binary | MS Excel | read _excel | to_excel |
| SQL | SQL | read _ sql | to_ sql |

# 数据分析库： Pandas

## （2）DataFrame 类型数组

- Pandas使用read_csv()方法读取csv文件

```
df = pandas.read_csv( 'filepath_or_buffer' , header= 'infer' , names=NoDefault ,
index_col=None , usecols=[], skiprows=None , encoding= 'utf-8' )
```

① **filepath_or_buffer**：文件所在**路径**，可以是字符串形式的文件路径、URL或文件对象。这个参数是**唯一一个必传**的参数。

② **header**：指定由哪一行作为**列名**，默认为header=0，表示第一行作为列名。如果header=None，表示不从文件数据中指定行作为列名，这时Pandas会自动生成从零开始的序列作为列名。

③ **usecols**：指定需要读取原数据集中的哪些列，可以是列名，也可以是列序号。

④ **encoding**：表示文件的编码格式，常用的编码有UTF-8、UTF-16、GBK、GB2312、GB18030等，通常为UTF-8。如果文件中含有中文，有时需要指定字符编码。

⑤ **index_col**：设置原csv文件里哪一列的值作为DataFrame对象的index**索引值**。index默认从0开始的自动索引。

⑥ **names**：表示为读取的列加上指定的**列名**。

⑦ **skiprows**：数据读取时，指定需要跳过原数据集开头的行数。

# 数据分析库： Pandas

## （2）DataFrame 类型数组

- 数据查看与筛选：DataFrame类型支持行索引和列索引对行和列进行切片

① 使用**head()**方法查看前几行数据。

② 使用**tail()**方法查看最后几行数据

③ DataFrame类型的三个属性**index、columns和values**分别返回给定DataFrame的行索引、列名和值。

④ 使用**info()**方法获取 DataFrame 的**摘要**，包括索引、dtype、所有列的列名及其数据类型、每列中非空值的数量和内存使用情况。

# 数据分析库： Pandas

## （2）DataFrame 类型数组

- 数据查看与筛选：DataFrame类型支持行索引和列索引对行和列进行切片

a) loc访问器是基于“**标签**”选择数据的。

b) iloc通过**数字**选择某些行和列

c) at和iat访问器选择**某个位置**的值

d) 在DataFrame[]中给出**一定条件**筛选数据

# 数据分析库： Pandas

## （2）DataFrame 类型数组

- DataFrame 数据 统计分析

a) 常用的统计指标值：**计数、均值、标准差、最小值、最大值**

**b) 三个四分位数Q1、Q2、Q3**

✓ 第1四分位数 (Q1)，又称较小四分位数，等于该样本中所有数值由小到大排列后第25%的数字，即quantile(0.25)。

✓ 第2四分位数 (Q2)，又称中位数，等于该样本中所有数值由小到大排列后第50%的数字，即quantile(0.5)。

✓ 第3四分位数 (Q3)，又称较大四分位数，等于该样本中所有数值由小到大排列后第75%的数字，即quantile(0.75)。

# 数据分析库： Pandas

## （2）DataFrame 类型数组

- DataFrame 数据 数据预处理：重复值处理

➢ DataFrame.duplicated() 用来检测哪些行是重复的，它返回一个布尔序列，**唯一元素**为False，**重复元素**为True。其语法格式如下。

DataFrame.duplicated (subset=None, keep='first' )

参数说明：

subset：指定对哪些列检测是否存在重复值。

keep：指定如何考虑重复值，取值为**first （第一是唯一值）、last （最后是唯一值） 和 false （所有项都是重复值）** 。

# 数据分析库： Pandas

## （2）DataFrame 类型数组

- DataFrame 数据 数据预处理：重复值处理

➢ **DataFrame.drop_duplicates()**用来删除重复数据，其语法格式如下。

DataFrame. drop_duplicates（subset=None, keep='first',inplace=False)

参数说明：

subset和keep与duplicated()方法类似。

inplace：指定该方法是原地修改还是返回新的DataFrame对象，inplace=True时表示**原地修改**，inplace=False时表示**返回新的DataFrame对象**而不对原来的DataFrame对象进行修改。

# 数据分析库： Pandas

## （2）DataFrame 类型数组

- DataFrame 数据 数据预处理： 缺失值处理

➢ Pandas中提供了一些用于检查或处理**空值和缺失值**的函数，其中，使用**isnull()和notnull()**方法可以判断数据集中**是否存在**缺失值，使用**dropna()和fillna()**方法对缺失值进行**删除和填充**。

# 数据分析库： Pandas

## （2）DataFrame 类型数组

- DataFrame 数据 数据预处理： 缺失值处理

➢ dropna()方法删除带有缺失值的数据行，其语法格式如下。

**DataFrame.dropna（axis=0, how='any', thresh=None, subset=None, inplace=False)**

- 参数说明：

axis：表示按**行或列**删除，axis=0表示删除某些带缺失值的行，axis=1按列丢弃。

how：表示**删除方式**，how='any'表示只要某行包含缺失值就被删除，how='all'表示某行全部为缺失值才被删除。

# 数据分析库： Pandas

## （2）DataFrame 类型数组

- DataFrame 数据 数据预处理： 缺失值处理

➢ fillna()方法对缺失值进行填充，其语法格式如下。

**DataFrame.fillna（value=None, method=None, axis=None, inplace=False, limit=None, downcast=None, **kwargs)**

- 参数说明：

value：用于填充的缺失值的**值**，可以是**标量、字典、Series或DataFrame**。

method：定义填充空值的**方法**，取值为**{'backfill', 'bfill', 'pad', 'ffill', None}**，默认为None。pad / ffill表示使用遇到缺失值之前的最后一个有效值来填充当前缺失值， backfill / bfill表示使用缺失值后面遇到的第一个有效值来填充。

limit：整数值，默认为None。如果method被指定，表示最多填充多少个连续的缺失值。

# 数据分析库： Pandas

## （2）DataFrame 类型数组

- DataFrame 数据 数据预处理： 分组与汇总

✓ Pandas提供了groupby()方法根据指定的一列或多列对数据进行分组，并对分组后的数据进行求和、求平均值等多种操作，并自动忽略非数值项。

**DataFrame . groupby ( by=None, axis=0, level=None, as_index=True, sort=True, …)**

✓ 其中参数by指定**分组依据**，可以是指定列名、用于行索引的函数或字典等。

✓ 另外，可以调用聚合函数**aggregate()对groupby()**方法的结果进行**汇总**，不同列使用不同的聚合函数。使用DataFrame结构的agg()也可以实现汇总操作。

# 04

# 数据可视化： matplotlib

# 数据可视化： matplotlib

## 一、matplotlib 库

matplotlib 库是一款提供二维数据绘图可视化功能的第三方模块。

- matplotlib是Python的一套基于Numpy的绘图工具包， 基于Numpy和标准库Tkinter开发，提供包括折线图、散点图、直方图、饼图、箱线图等超过100种数据可视化展示效果，是数据可视化的重要工具。
- matplotlib主要通过matplotlib.pyplot子库实现各种数据展示图形绘制
- <官网>： https://matplotlib.org/
- <安装> anaconda模式 → conda install matplotlib
- <调用>： import matplotlib.pyplot as plt # 缩写模式

# 数据可视化： matplotlib

## 1. 中文显示不乱码，以及正常显示负号

```
#调整字体设置
plt.rcParams['font.sans-serif']=['SimHei’]
# 默认是使用Unicode负号，设置正常显示字符
plt.rcParams['axes.unicode_minus'] =False
```

| 字体 | 字体名 |
| --- | --- |
| 黑体 | SimHei |
| 楷体 | KaiTi |
| 隶书 | LiSu |
| 幼圆 | YouYuan |
| 华文细黑 | STXihei |
| 华文楷体 | STKaiti |
| 华文宋体 | STSong |
| 华文中宋 | STZhongsong |
| 华文仿宋 | STFangsong |
| 方正舒体 | FZShuTi |
| 方正姚体 | FZYaoti |
| 华文彩云 | STCaiyun |
| 华文琥珀 | STHupo |
| 华文隶书 | STLiti |
| 华文行楷 | STXingkai |
| 华文新魏 | STXinwei |

# 数据可视化： matplotlib

## 2. 创建画布与创建子图

在pyplot中，创建画布及创建并选择子图的函数/方法

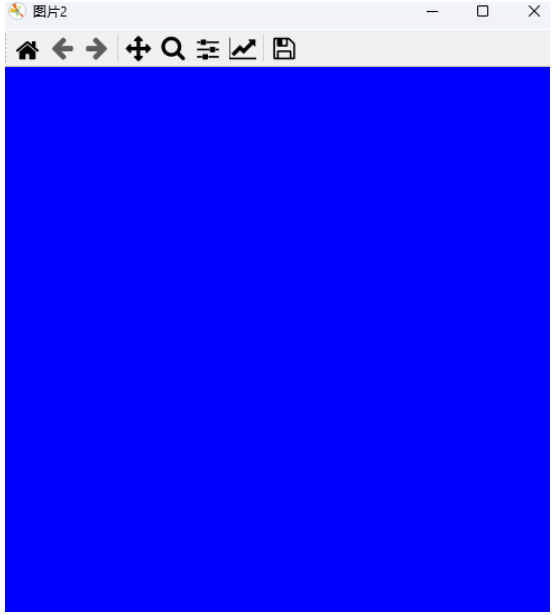
| 函数/方法名称 | 函数/方法作用 |
| --- | --- |
| plt.figure () | 创建一个空白画布，可以指定画布大小、像素 |
| plt.subplot() | 创建并选中子图，可以指定子图的行数、列数和选中图片的编号 |

# 数据可视化： matplotlib

## 2. 创建画布与创建子图

plt.figure (num=None, figsize=None, dpi=None, facecolor=None, edgecolor=None, frameon=True)

- num: 图像编号或名称，数字为编号 ， 字符串为名称
- figsize:指定figure的宽和高，单位为英寸；
- dpi参数指定绘图对象的分辨率
- facecolor:背景颜色
- edgecolor:边框颜色
- frameon:是否显示边框



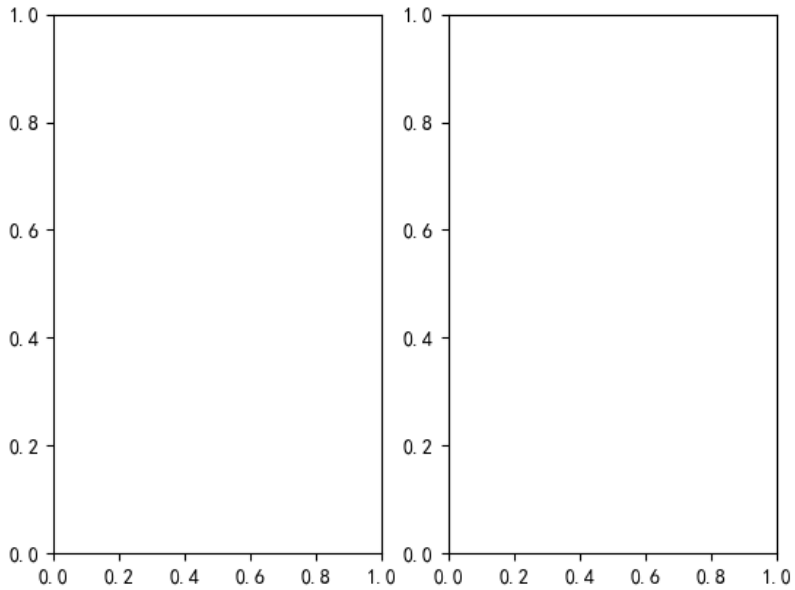
# 数据可视化： matplotlib

## 2. 创建画布与创建子图

plt. subplot (nrows, ncols, index) ： 一张画布上创建多个子图

- 行数 nrows, 列数 ncols, 索引号index
- 例如，121，第一个1表示1行，第二个2表示2列，第3位上的1表示第1个格子

```
>>>
    plt.subplot (121)
    plt.subplot (122)
```



# 数据可视化： matplotlib

## 3. 添加画布内容

添加标题、添加坐标轴名称

| 函数名称 | 函数作用 |
| --- | --- |
| plt.title() | 在当前图形中添加标题，可以指定标题的名称、位置、颜色、字体大小等参数 |
| plt.xlabel() | 在当前图形中添加x轴标签，可以指定位置、颜色、字体大小等参数 |
| plt.ylabel() | 在当前图形中添加y轴标签，可以指定位置、颜色、字体大小等参数 |

# 数据可视化： matplotlib

## 3. 添加画布内容

- plt. title ()

- fontsize设置字体大小，默认12，可选参数 [‘xx-small’, ‘x-small’, ‘small’, ‘medium’, ‘large’,‘x-large’, ‘xx-large’]
- fontweight设置字体粗细，可选参数 [‘light’, ‘normal’, ‘medium’, ‘semibold’, ‘bold’, ‘heavy’, ‘black’]
- fontstyle设置字体类型，可选参数[ ‘normal’ | ‘italic’ | ‘oblique’ ]，italic斜体，oblique倾斜
- verticalalignment设置水平对齐方式 ，可选参数： ‘center’ , ‘top’ , ‘bottom’ , ‘baseline’
- horizontalalignment设置垂直对齐方式，可选参数：left, right, center
- rotation(旋转角度)可选参数为:vertical,horizontal 也可以为数字

# 数据可视化： matplotlib

## 3. 添加画布内容

| 函数名称 | 函数作用 |
| --- | --- |
| plt.xlim() | 指定当前图形x轴的范围，只能确定一个数值区间，而无法使用字符串标识 |
| plt.ylim() | 指定当前图形y轴的范围，只能确定一个数值区间，而无法使用字符串标识 |
| plt.xticks() | 获取或设置x轴的当前刻度位置,标签,标签字体倾斜度和颜色等外观属性 |
| plt.yticks() | 获取或设置y轴的当前刻度位置和标签,标签字体倾斜度和颜色等外观属性 |
| plt.legend() | 指定当前图形的图例，可以指定图例的大小、位置、标签 |

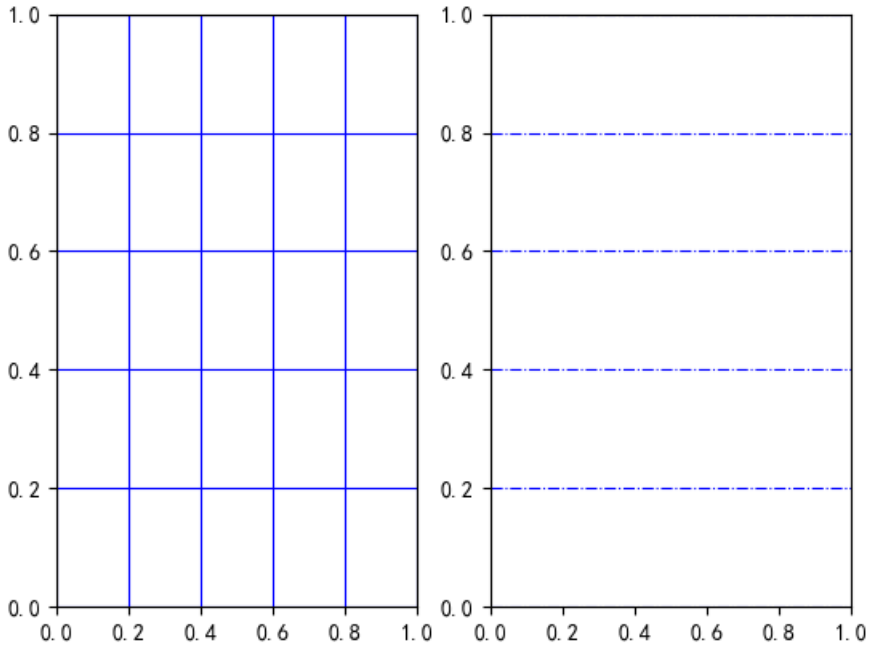
# 数据可视化： matplotlib

## 3. 添加画布内容

➢ 设置绘图网格

pyplot.grid ( axis, color, linestyle, linewidth)

- axis : 取值为 ‘both’， ‘x’， ‘y’ 是以什么轴为刻度生成网格。
- color : 设置网格线的颜色。
- linestyle :设置网格线的风格，| '-' | '--' | '-.' | ':' | ' ' |
- linewidth : 设置网格线的宽度



# 数据可视化： matplotlib

## 4. 保存与显示图形

| 函数名称 | 函数作用 |
| --- | --- |
| plt.savefig() | 保存绘制的图形，可以指定图形的分辨率、边缘的颜色等参数 |
| plt.show() | 在本机显示图形 |

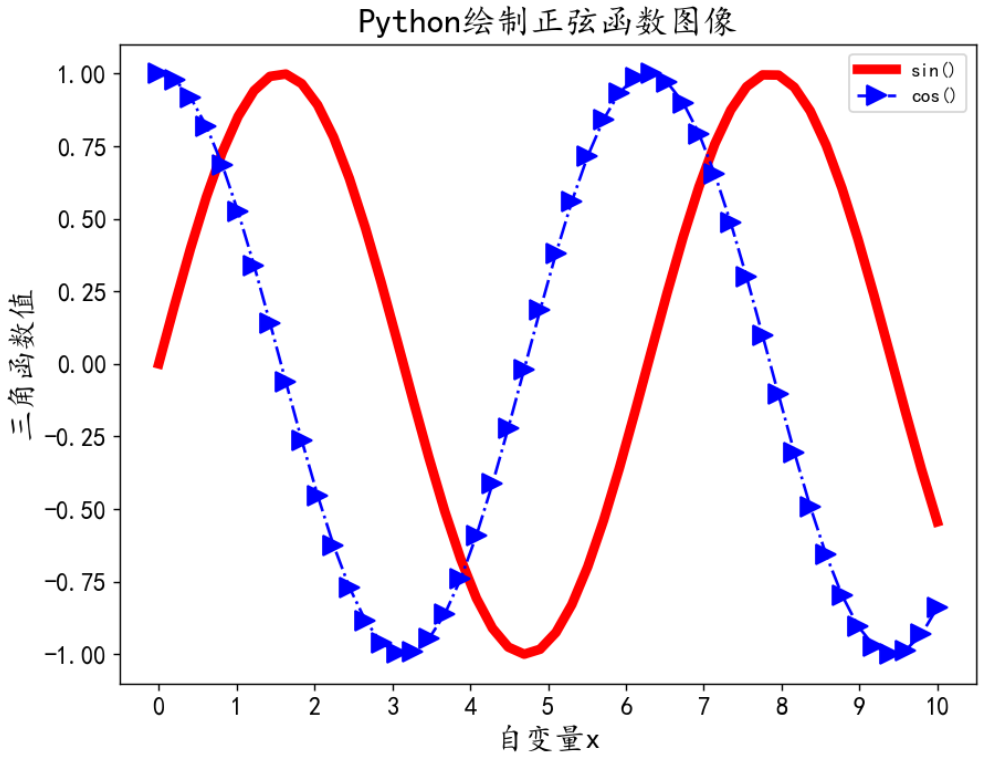
```
>>>
    plt. savefig ( ‘fig . png’ )
```

# 数据可视化： matplotlib

## (1) 绘制折线图 --- plot()

- 折线图（Line Chart）是一种将数据点按照顺序连接起来的图形，可以看作是将散点图按照x轴坐标顺序连接起来的图形。

- 如何清晰展示出图像结果，包括图像标题、图标、坐标设置、颜色、线形等？



# 数据可视化： matplotlib

- linewidth：指定折线的宽度
- markersize：设置点的大小
- label：为折线图添加标签

## 绘制折线图--- matplotlib.pyplot.plot()

plt.plot( x, y, '#color#linestyle#marker',linewidth= , markersize= ,label=)

必备参数

| 参数名称 | 解释 | 取值 |
| --- | --- | --- |
| linewidth | 线条宽度 | 取0～10之间的数值，默认为1.5 |
| linestyle | 线条样式 | 可取“-” “--” “-.” “：” 4种。默认为“-” |
| marker | 线条上点的形状 | 可取“o” “D” “h” “.” “,” “S”等20种，默认为None |
| markersize | 点的大小 | 取0～10之间的数值，默认为1 |

# 数据可视化： matplotlib

## 绘制折线图--- matplotlib.pyplot.plot()

plt.plot( x, y, ‘#color#linestyle#marker’,linewidth= , markersize= ,label=)

必备参数

- color：点、线条颜色

| 字符串 | 颜色 |
| --- | --- |
| ‘r’ | 红色 |
| ‘g’ | 绿色 |
| ‘b’ | 蓝色 |
| ‘y’ | 黄色 |
| ‘w’ | 白色 |
| ‘k’ | 黑色 |
| ‘m’ | 品红 |

- linestyle：线条样式

| 字符串 | 线形 |
| --- | --- |
| ‘-’ | 实线 |
| ‘--’ | 虚线 |
| ‘-.’ | 点划线 |
| ‘: ’ | 点虚线 |
| ‘ ’ | 空格 |
| ‘None’ / ‘’ | 无线 |

- marker：点的样式

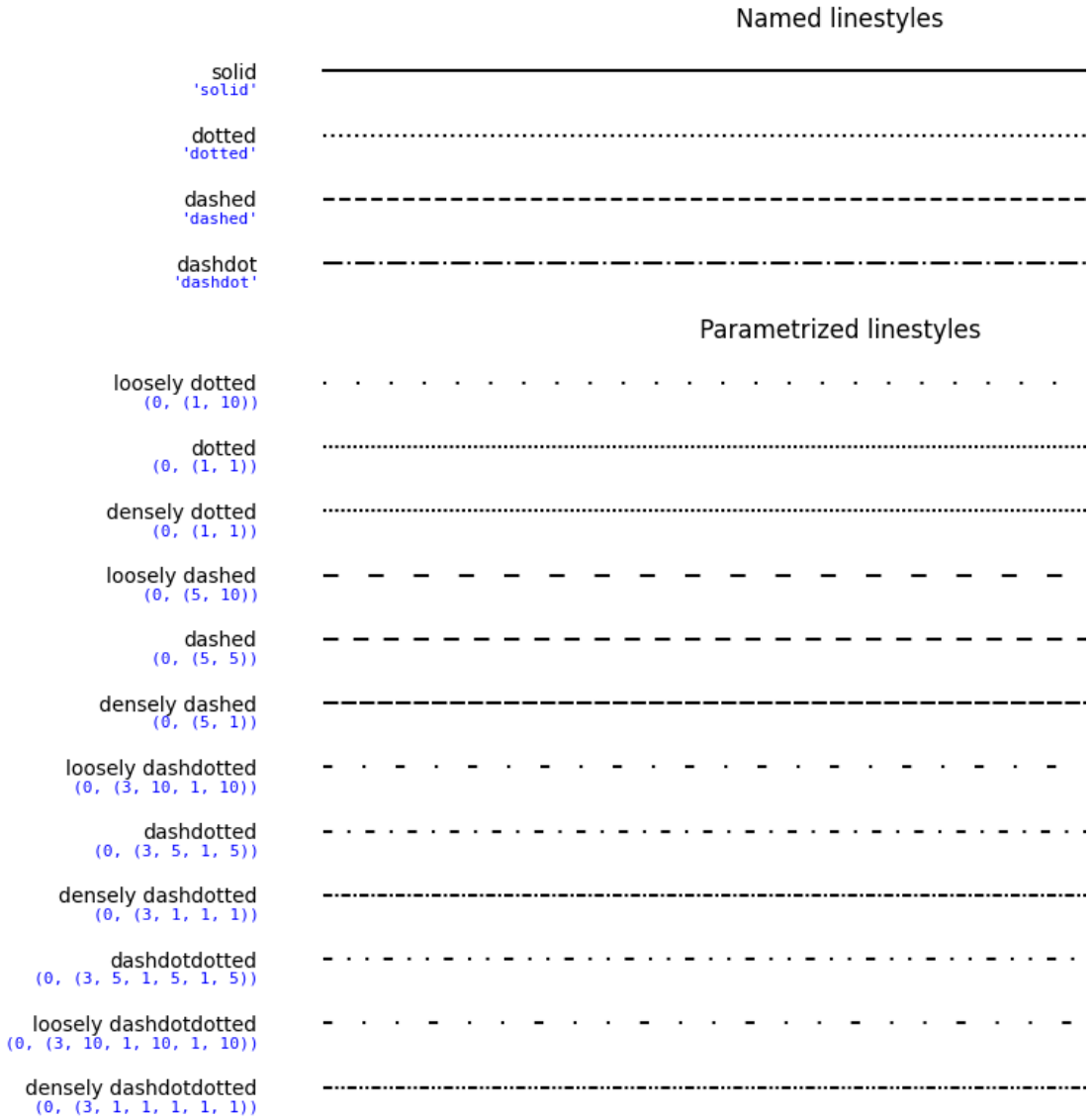
| 字符串 | 标记点 |
| --- | --- |
| ‘.’ | 点 |
| ‘^’‘<’‘>’ | 三角形 |
| ‘1’‘2’‘3’ | 三叉线 |
| ‘o’ | 圆形 |
| ‘s’‘D’ | 方形 |
| ‘p’ | 五边形 |
| ‘*’ | 五角星 |

# 数据可视化： matplotlib

| | | | |
|---|---|---|---|
| black | bisque | lightgreen | slategrey |
| k | darkorange | forestgreen | lightsteelblue |
| dimgray | burlywood | limegreen | cornflowerblue |
| dimgrey | antiquewhite | darkgreen | royalblue |
| grey | tan | green | ghostwhite |
| gray | navajowhite | g | lavender |
| darkgrey | blanchedalmond | lime | midnightblue |
| darkgray | papayawhip | seagreen | navy |
| silver | moccasin | mediumseagreen | darkblue |
| lightgray | orange | springgreen | mediumblue |
| lightgrey | wheat | mintcream | blue |
| gainsboro | oldlace | mediumspringgreen | b |
| whitesmoke | floralwhite | mediumaquamarine | slateblue |
| white | darkgoldenrod | aquamarine | darkslateblue |
| w | goldenrod | turquoise | mediumslateblue |
| snow | cornsilk | lightseagreen | mediumpurple |
| rosybrown | gold | mediumturquoise | blueviolet |
| lightcoral | lemonchiffon | azure | indigo |
| indianred | khaki | lightcyan | darkorchid |
| brown | palegoldenrod | paleturquoise | darkviolet |
| firebrick | darkkhaki | darkslategray | mediumorchid |
| maroon | ivory | darkslategrey | thistle |
| darkred | beige | teal | plum |
| red | lightyellow | darkcyan | violet |
| r | lightgoldenrodyellow | c | purple |
| mistyrose | olive | cyan | darkmagenta |
| salmon | y | aqua | m |
| tomato | yellow | darkturquoise | fuchsia |
| darksalmon | olivedrab | cadetblue | magenta |
| coral | yellowgreen | powderblue | orchid |
| orangered | darkolivegreen | lightblue | mediumvioletred |
| lightsalmon | greenyellow | deepskyblue | deeppink |
| sienna | chartreuse | skyblue | hotpink |
| seashell | lawngreen | lightskyblue | lavenderblush |
| chocolate | sage | steelblue | palevioletred |
| saddlebrown | lightsage | aliceblue | crimson |
| sandybrown | darksage | dodgerblue | pink |
| peachpuff | honeydew | lightslategrey | lightpink |
| peru | darkseagreen | lightslategray | |
| linen | palegreen | slategray | |

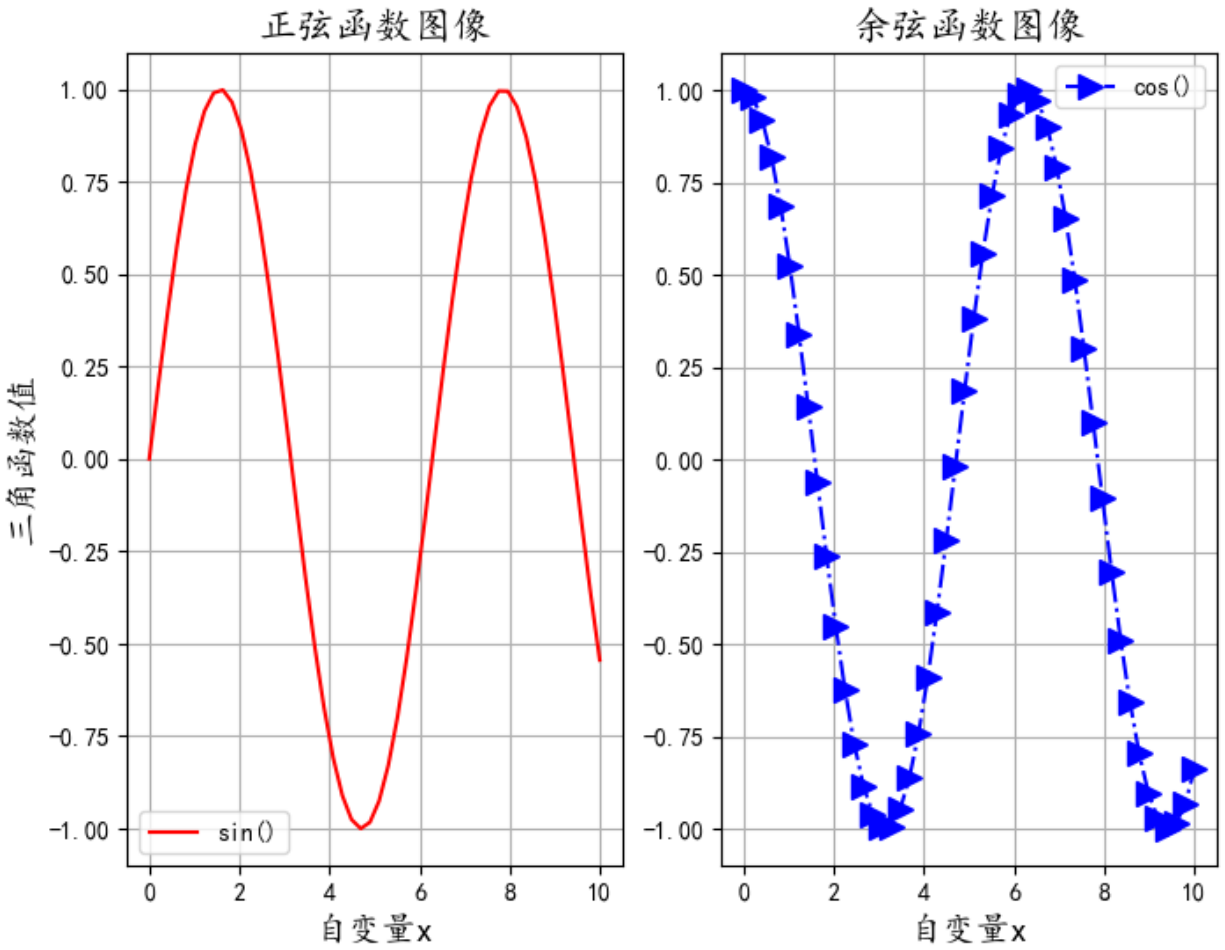
| | | | | | | |
|---|---|---|---|---|---|---|
| FFFFFF | #DDDDDD | #AAAAAA | #888888 | #666666 | #444444 | #000000 |
| #FFB7DD | #FF88C2 | #FF44AA | #FF0088 | #C10066 | #A20055 | #8C0044 |
| #FFCCCC | #FF8888 | #FF3333 | #FF0000 | #CC0000 | #AA0000 | #880000 |
| #FFC8B4 | #FFA488 | #FF7744 | #FF5511 | #E63F00 | #C63300 | #A42D00 |
| #FFDDAA | #FFBB66 | #FFAA33 | #FF8800 | #EE7700 | #CC6600 | #BB5500 |
| #FFEE99 | #FFDD55 | #FFCC22 | #FFBB00 | #DDAA00 | #AA7700 | #886600 |
| #FFFFBB | #FFFF77 | #FFFF33 | #FFFF00 | #EEEE00 | #BBBB00 | #888800 |
| #EEFFBB | #DDFF77 | #CCFF33 | #BBFF00 | #99DD00 | #88AA00 | #668800 |
| #CCFF99 | #BBFF66 | #99FF33 | #77FF00 | #66DD00 | #55AA00 | #227700 |
| #99FF99 | #66FF66 | #33FF33 | #00FF00 | #00DD00 | #00AA00 | #008800 |
| #BBFFEE | #77FFCC | #33FFAA | #00FF99 | #00DD77 | #00AA55 | #008844 |
| #AAFFEE | #77FFEE | #33FFDD | #00FFCC | #00DDAA | #00AA88 | #008866 |
| #99FFFF | #66FFFF | #33FFFF | #00FFFF | #00DDDD | #00AAAA | #008888 |
| #CCEEFF | #77DDFF | #33CCFF | #00BBFF | #009FCC | #0088A8 | #007799 |
| #CCDDFF | #99BBFF | #5599FF | #0066FF | #0044BB | #003C9D | #003377 |
| #CCCCFF | #9999FF | #5555FF | #0000FF | #0000CC | #0000AA | #000088 |
| #CCBBFF | #9F88FF | #7744FF | #5500FF | #4400CC | #2200AA | #220088 |
| #D1BBFF | #B088FF | #9955FF | #7700FF | #5500DD | #4400B3 | #3A0088 |
| #E8CCFF | #D28EFF | #B94FFF | #9900FF | #7700BB | #66009D | #550088 |
| #F0BBFF | #E38EFF | #E93EFF | #CC00FF | #A500CC | #7A0099 | #660077 |
| #FFB3FF | #FF77FF | #FF3EFF | #FF0 0FF | #CC00CC | #990099 | #770077 |

# 数据可视化： matplotlib




| marker | symbol | description |
|---|---|---|
| "." | • | point |
| "," | · | pixel |
| "o" | ● | circle |
| "v" | ▼ | triangle_down |
| "^" | ▲ | triangle_up |
| "<" | ◀ | triangle_left |
| ">" | ▶ | triangle_right |
| "1" | Y | tri_down |
| "2" | ⅄ | tri_up |
| "3" | ≺ | tri_left |
| "4" | ≻ | tri_right |
| "8" | ● | octagon |
| "s" | ■ | square |
| "p" | ⬟ | pentagon |
| "P" | ✚ | plus (filled) |
| "*" | ★ | star |
| "h" | ⬢ | hexagon1 |
| "H" | ⬣ | hexagon2 |
| "+" | + | plus |
| "x" | × | x |
| "X" | ✖ | x (filled) |
| "D" | ◆ | diamond |
| "d" | ♦ | thin_diamond |

# 数据可视化： matplotlib




如何实现？

# 数据可视化： matplotlib

## （2） 绘制散点图 ---scatter()

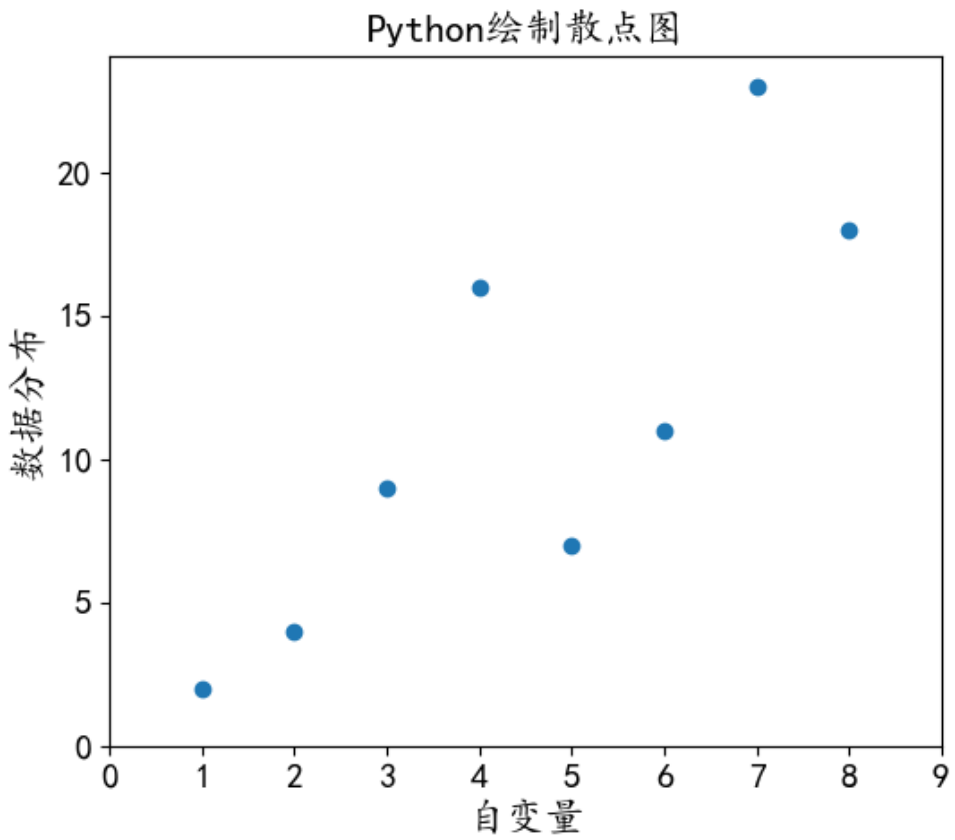
- 散点图（Scatter Diagram）又称为散点分布图，是以一个特征为横坐标，以另一个特征为纵坐标，利用坐标点（散点）的分布形态反映特征间的统计关系的一种图形。
- matplotlib.pyplot.scatter （x, y, s=None, c=None, marker=None, cmap=None, norm=None, vmin=None, vmax=None, alpha=None）

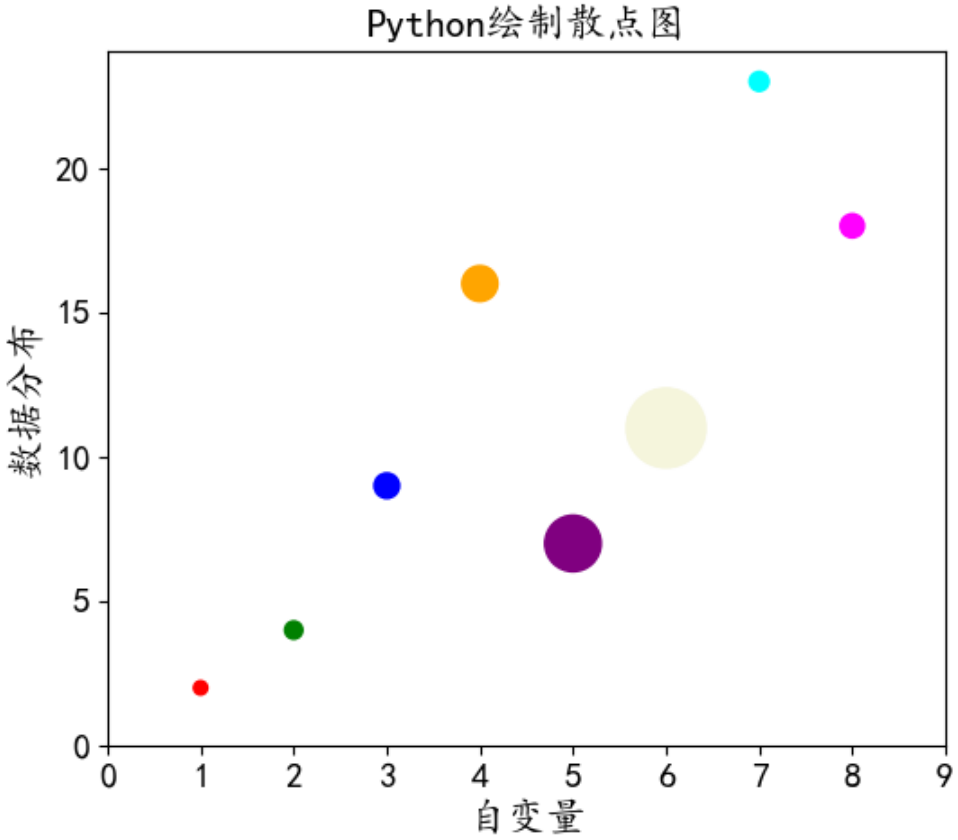
必备参数

  - **x，y**：长度相同的数组，也就是我们即将绘制散点图的数据点，输入数据。
  - **s**：点的大小，默认 20，也可以是个数组，数组每个参数为对应点的大小。
  - **c**：点的颜色，默认蓝色 'b'，也可以是个 RGB 或 RGBA 二维行数组。
  - **marker**：点的样式，默认小圆圈 'o'。
  - **cmap**：Colormap颜色映射集，默认 None，标量或者是一个 colormap 的名字，只有 c 是一个浮点数数组的时才使用。可取值’ viridis’、’ plasma’、’ inferno’、’ magma’等

# 数据可视化： matplotlib

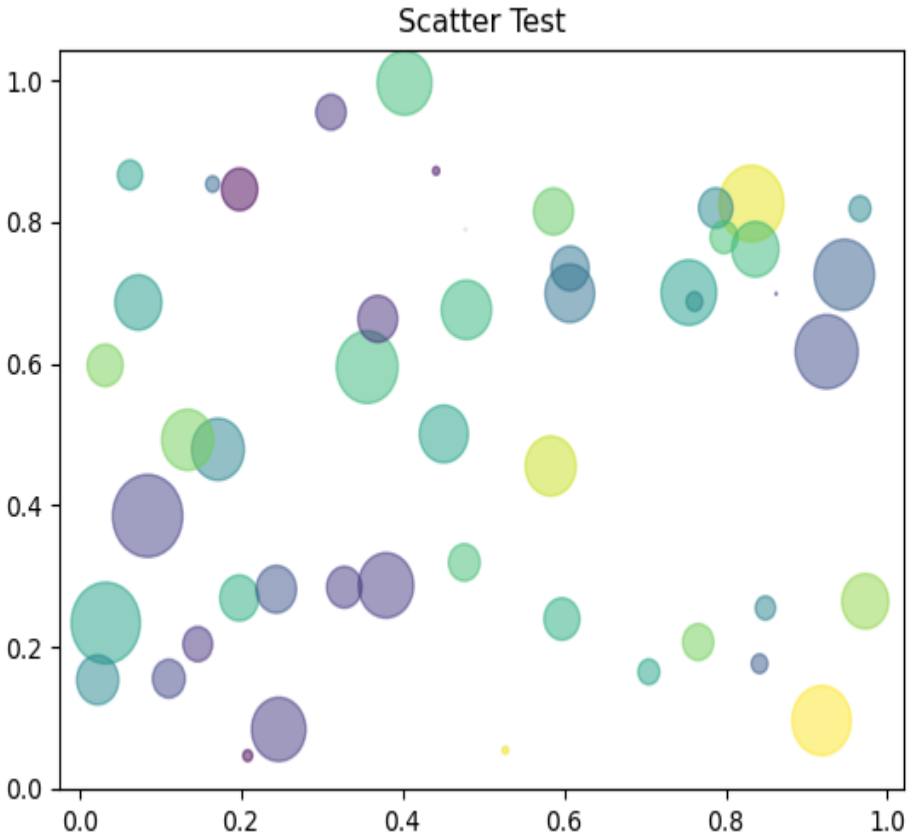
## 案例一

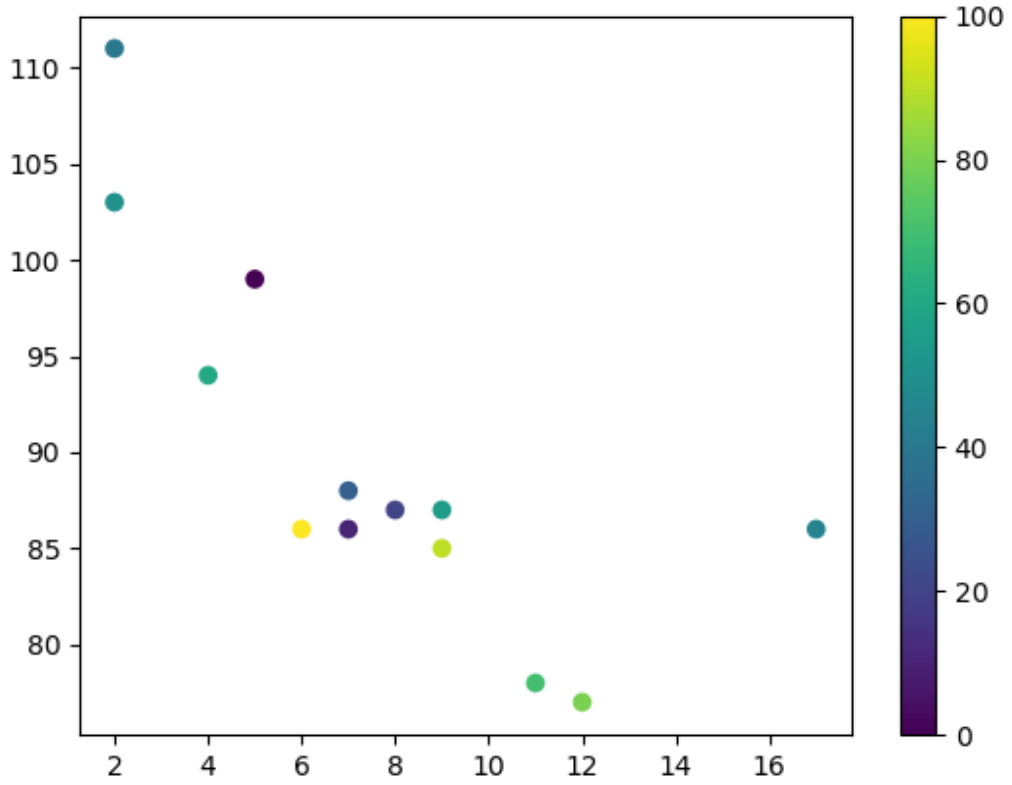


# 数据可视化： matplotlib

## 案例二



# 数据可视化： matplotlib

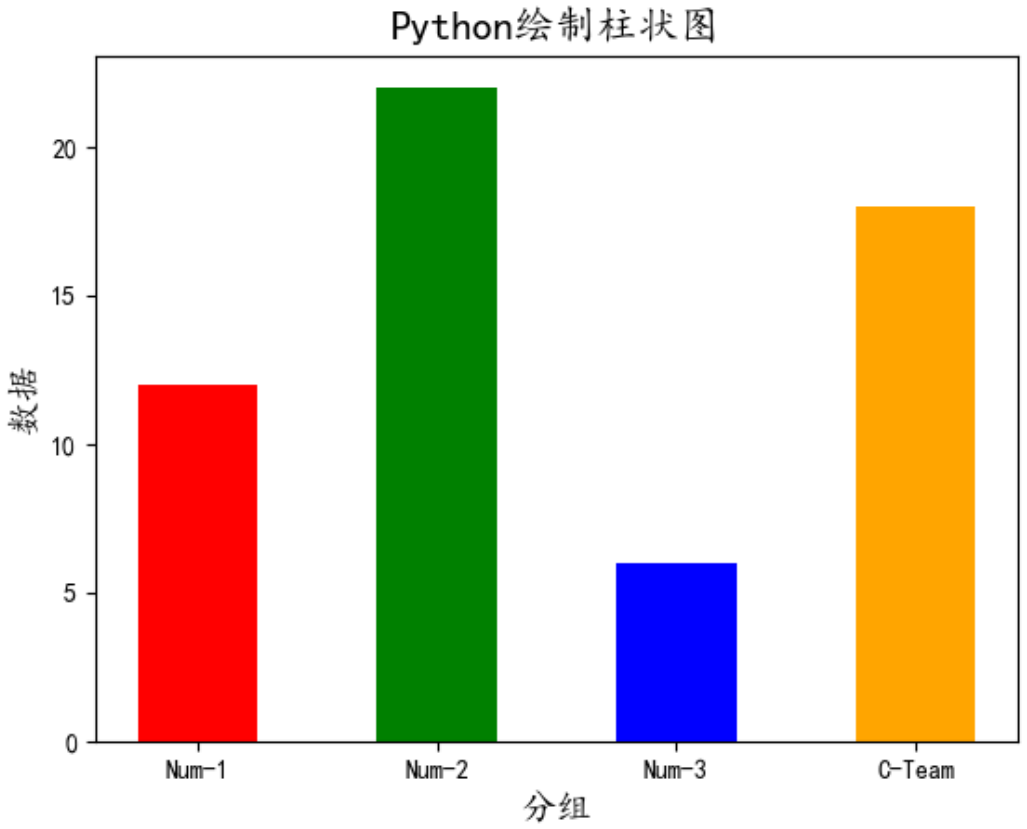
## （3） 绘制柱状图（Bar Chart） ---bar()

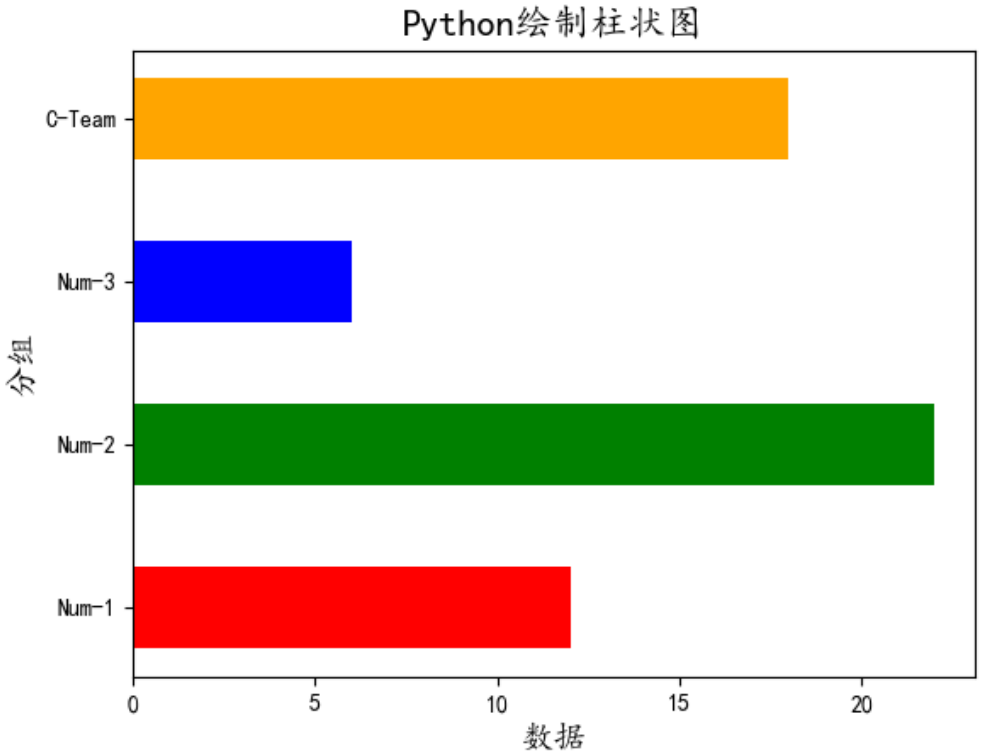
- *matplotlib.pyplot.bar ( x, height, width = 0.8, bottom = None, *, align = 'center', data = None, ** kwargs)*

| 参数名称 | 参数说明 |
| --- | --- |
| x | 接收array或float。表示x轴数据。无默认值 |
| height | 接收array或float。表示指定柱形图的高度/ y轴数值。无默认值 |
| width | 接收array或float。表示指定柱形图的宽度。默认为0.8 |
| align | 接收str。表示整个柱形图与x轴的对齐方式，可选center和edge。默认为center |
| color | 接收特定str或包含颜色字符串的list。表示柱形图颜色。默认为None |

# 数据可视化： matplotlib

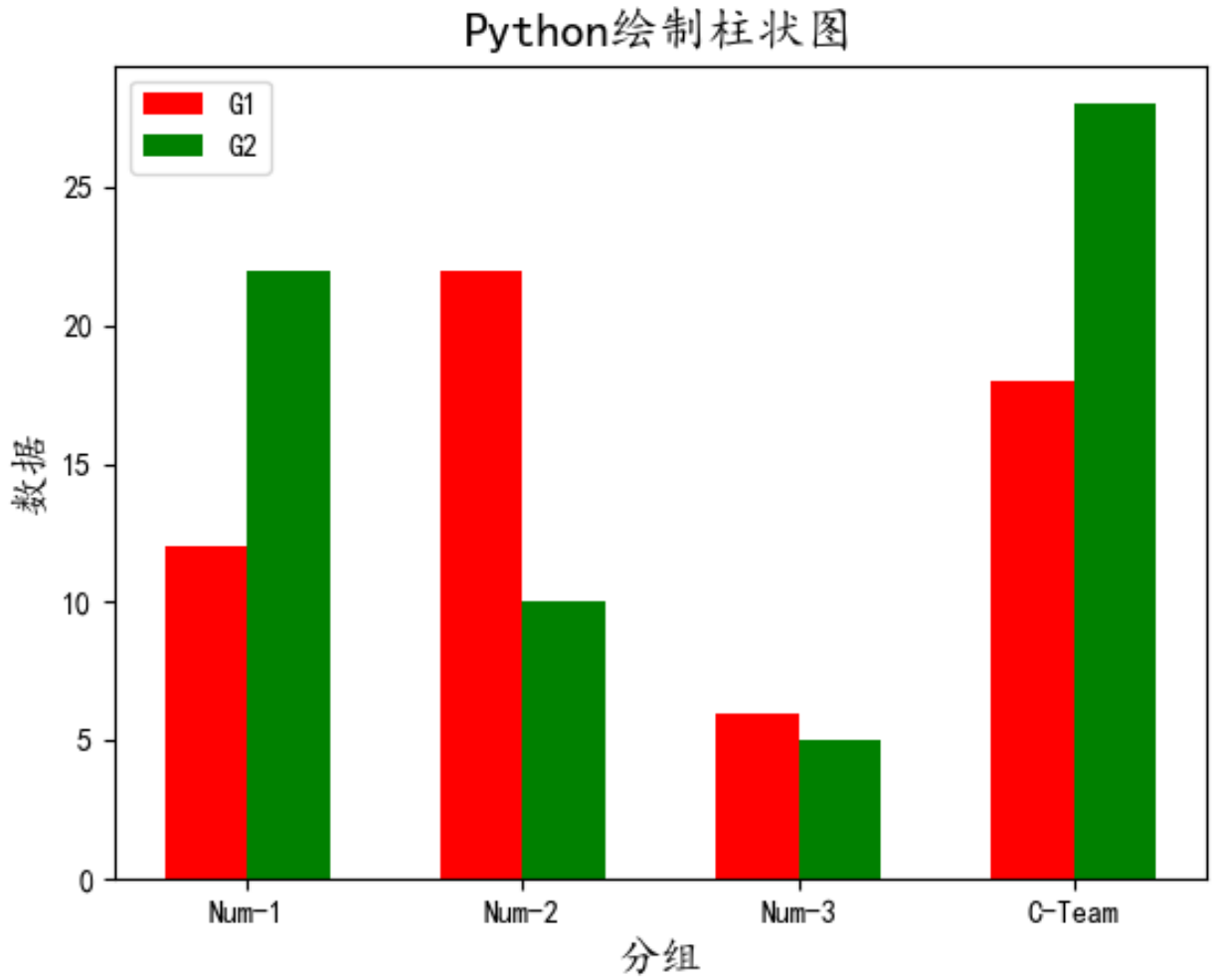
## 案例一



# 数据可视化： matplotlib

案例二



# 数据可视化： matplotlib

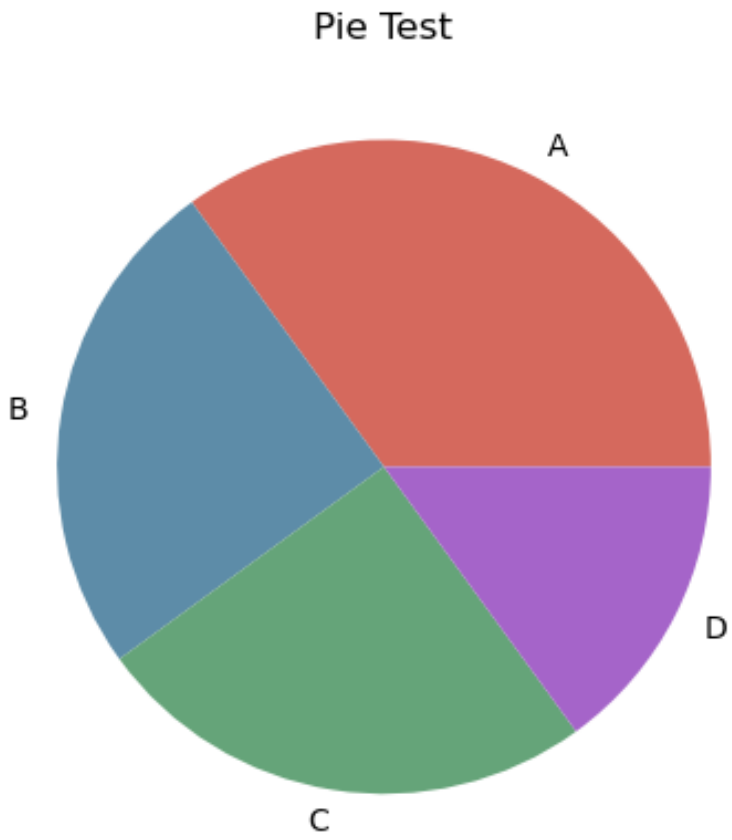
## （4） 绘制饼图（Pie Graph）-- -pie()

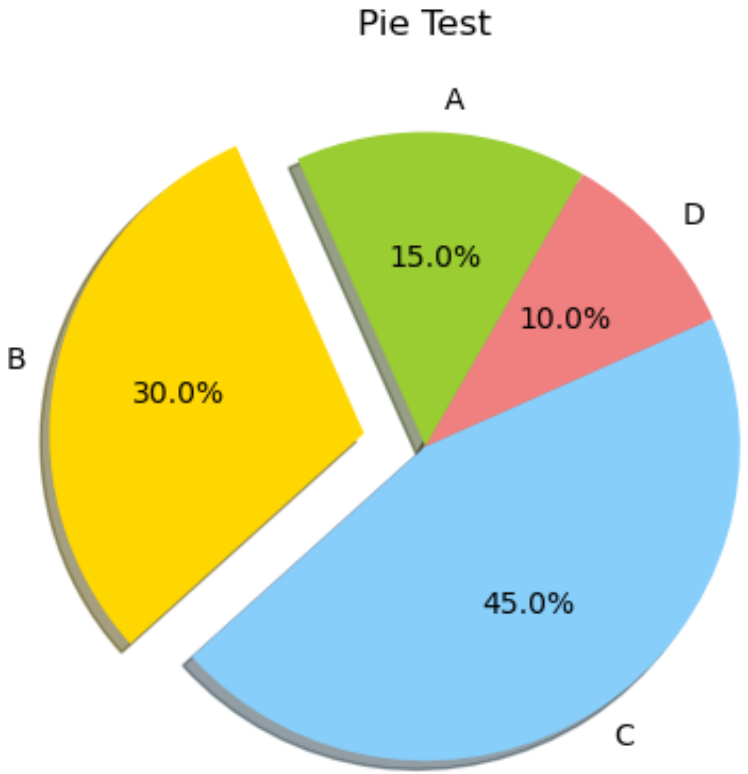
- matplotlib.pyplot.pie (x, explode=None, labels=None, colors=None, autopct=None, pctdistance=0.6, shadow=False, labeldistance=1.1, startangle=0, radius=1, counterclock=True, wedgeprops=None, textprops=None, center=0, 0, frame=False, rotatelabels=False, *, normalize=None, data=None)[source]

| 参数名称 | 参数说明 |
| --- | --- |
| x | 接收array。表示用于绘制饼图的数据。无默认值 |
| explode | 接收array。表示指定饼块距离饼图圆心的偏移距离。默认为None |
| labels | 接收list。表示指定每一项的标签名称。默认为None |
| color | 接收特定str或包含颜色字符串的array。表示饼图颜色。默认为None |
| autopct | 接收特定str。设置饼图内各个扇形百分比显示格式，%d%% 整数百分比，%0.1f 一位小数， %0.1f%% 一位小数百分比， %0.2f%% 两位小数百分比 |
| shadow | 布尔值 True 或 False，设置饼图的阴影，默认为 False，不设置阴影 |
| startangle | 用于指定饼图的起始角度，默认为从 x 轴正方向逆时针画起，如设定 =90 则从 y 轴正方向画起。 |
| radius | 接收float。表示饼图的半径。默认为1 |

# 数据可视化： matplotlib

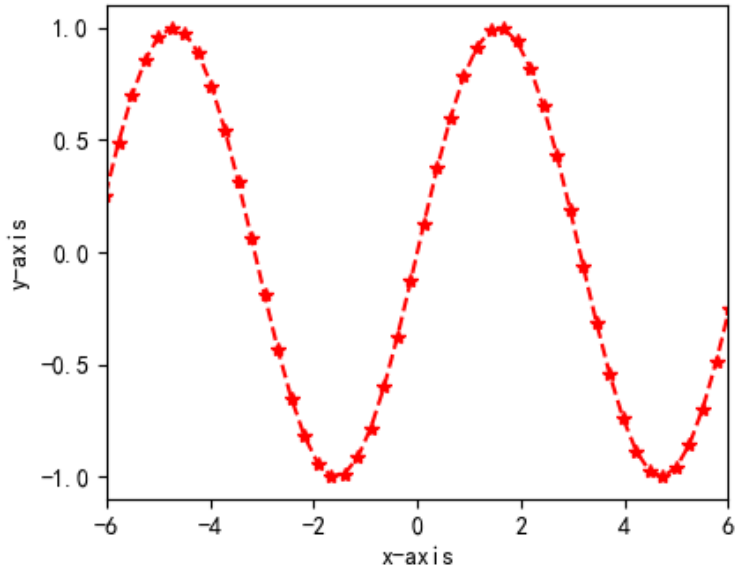
## 案例一

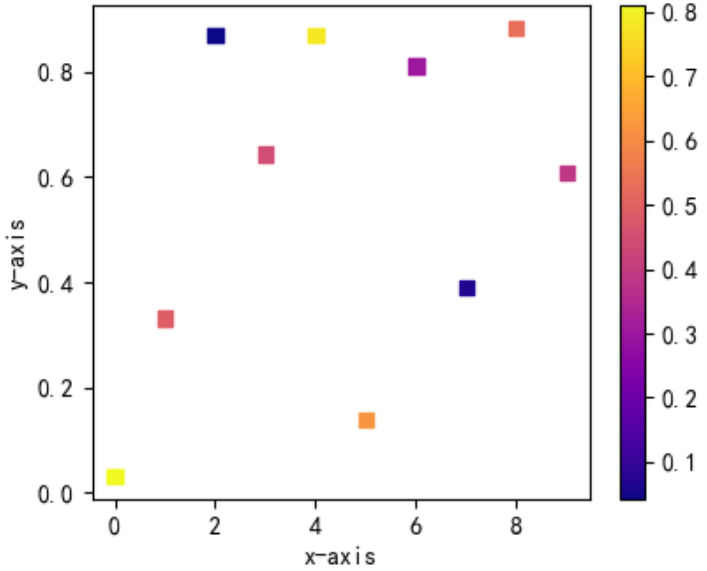


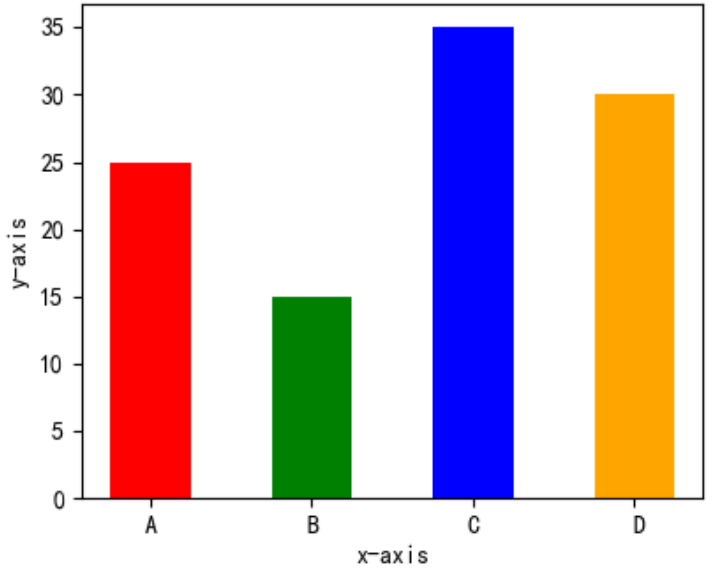



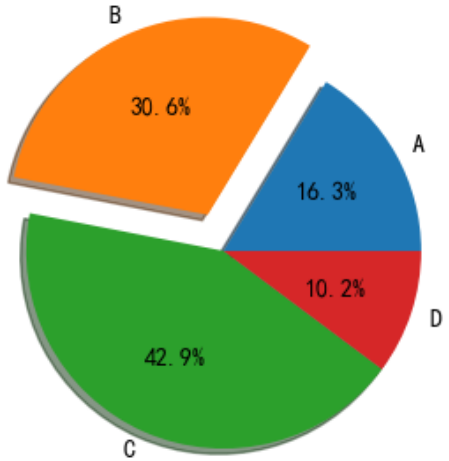
# 数据可视化： matplotlib

综合练习：



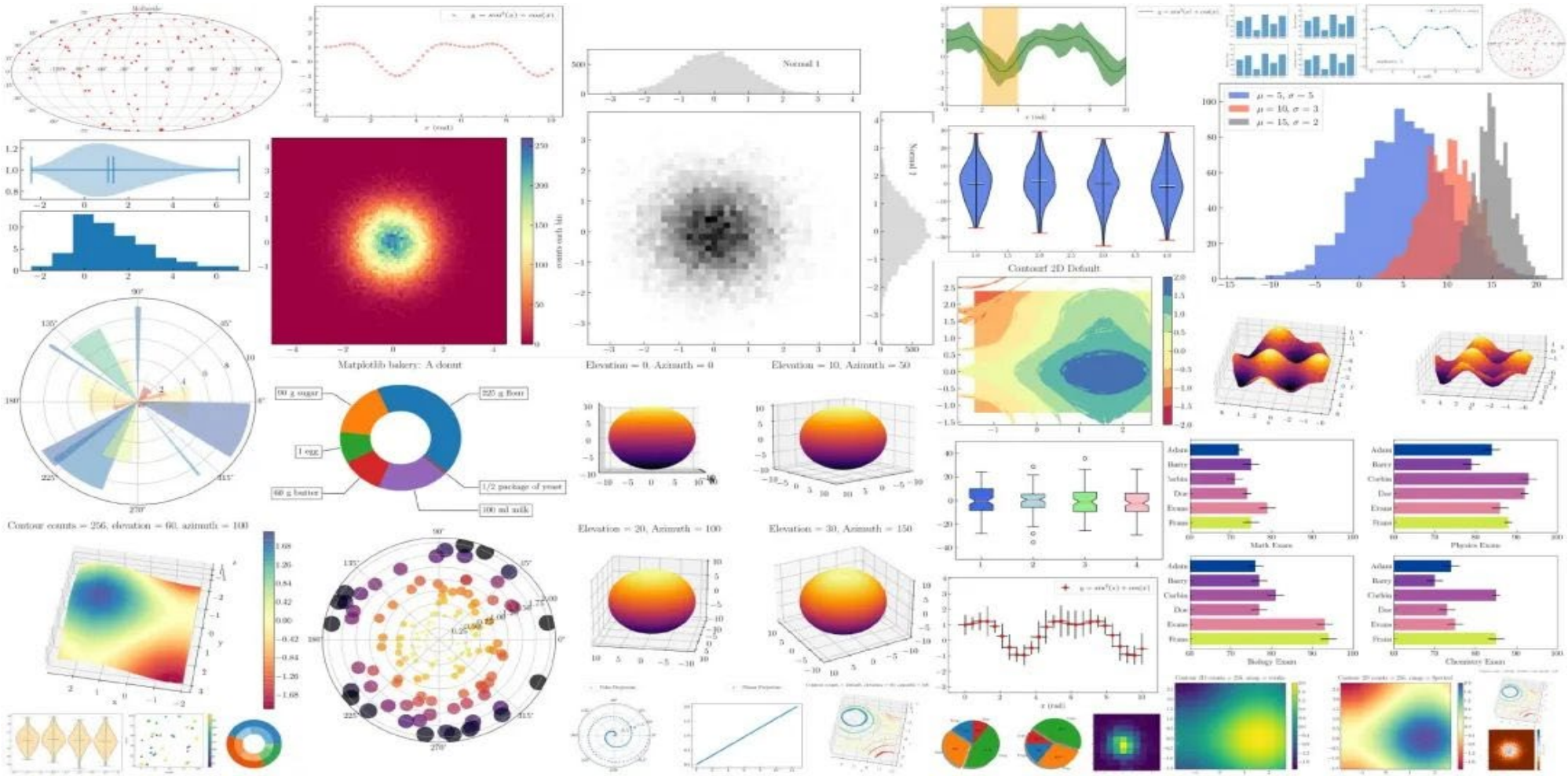



# 数据可视化： matplotlib

绘图手册：https://matplotlib.org/3.3.0/tutorials/introductory/sample_plots.html



# 05

# 词云图

# 词云图

## 一、 词云图

“词云”就是对文件文本中出现频率较高的“关键词”予以视觉上的突出，形成“关键词云层”或“关键词渲染”，从而过滤掉大量的文本信息，使浏览者一眼领略文本的主旨。依据权重，角度、字体、颜色等属性来控制生成的词云图 。



# 词云图

## 二、绘制词云图

◆ 绘制整体思路：

- <步骤 1>：文件数据管理，后缀.txt文件或其他格式存储
- <步骤 2>：数据清洗处理，提取文本中单词、词语、词组
- <步骤 3>：依据频率较高关键词，绘制词云图

- jieba库：第三方中文分词库 □ 提取文本词组
- wordcloud库：第三方词云展示库 □ 绘制词云图

# 词云图

## (1) jieba 库

jieba 库是一款优秀的 Python 第三方中文分词库。分词原理：利用一个中文词库，将待分词内容和分词词库比较，通过图结构和动态规划方法找到最大概率的词组。jieba 支持三种分词模式：精确模式、全模式和搜索引擎模式。

<安装>：conda install jieba

| 函数 | 描述 |
| --- | --- |
| jieba.lcut(s) | 精确模式：将语句最精确的切分，不存在冗余数据，适合做文本分析 |
| jieba.lcut(s，cut_all = Ture) | 全模式：将语句中所有可能的词语都切分出来，但存在冗余数据 |
| jieba.lcut_for_search(s) | 搜索引擎模式：在精确模式的基础上，对长词再次进行切分 |
| jieba.add_word(w) | 向分词词典中添加新词w |

# 词云图

## (1) jieba 库

实例： >>> Is= ‘中华人民共和国’

- >>> word = jieba.lcut (Is) <精确模式>

→ 运行结果: [‘中华人民共和国’]

- >>> word = jieba.lcut (Is, cut_all=Ture) <全模式>

→ 运行结果: [ '中华', '中华人民', '中华人民共和国', '华人', '人民','人民共和国', '共和', '共和国’]

- >>> word = jieba.lcut_for_search (Is) <搜索引擎模式>

→ 运行结果: ['中华', '华人', '人民', '共和', '共和国', '中华人民共和国']

# 词云图

## (2) wordcloud 库

wordcloud 库是一款优秀的 Python 第三方词云展示库。以词语为基本单位，通过图形可视化的方式，更加直观和艺术的展示文本。

<安装>: conda install -c conda-forge wordcloud

- 基本使用:

w= wordcloud.WordCloud(参数)　　#设置参数，构建绘制词云框架

→以 WordCloud对象为基础，加载文本、配置参数（形状、尺寸和颜色）、输出文件

| 方法 | 描述 |
| --- | --- |
| w. generate () | 向WordCloud对象中加载文本txt |
| w. to_file(filename) | 将词云输出为图像文件，.png或.jpg格式 |

# 词云图

w= wordcloud.WordCloud(参数)　　#设置参数，构建绘制词云框架

| 方法 | 描述 |
| --- | --- |
| width | 指定词云对象生成图片的宽度,默认400像 □ w=wordcloud.WordCloud(width=600) |
| height | 指定词云对象生成图片的高度,默认200像素 |
| min_font_size | 指定词云中字体的最小字号，默认4号 |
| max_font_size | 指定词云中字体的最大字号，根据高度自动调节 |
| font_step | 指定词云中字体的最大字号，根据高度自动调节 |
| font_path | 字体路径，词云图默认不支持中文，所以一般都要设置该参数<br>□ w=wordcloud.WordCloud(font_path="c:\Windows\Fonts\simfang.ttf") |
| max_words | 指定词云显示的最大单词数量,默认200 |
| stop_words | 指定词云的排除词列表，即不显示的单词列表 |
| background_color | 指定词云图片的背景颜色，默认为黑色 □ w=wordcloud.WordCloud(background_color="white") |
| mask | 指定词云形状，默认为长方形，需要引用imread()函数<br>mk=imread("pic.png")<br>w=wordcloud.WordCloud(mask=mk) |

# 词云图

## 三、《党的二十大报告》词云图

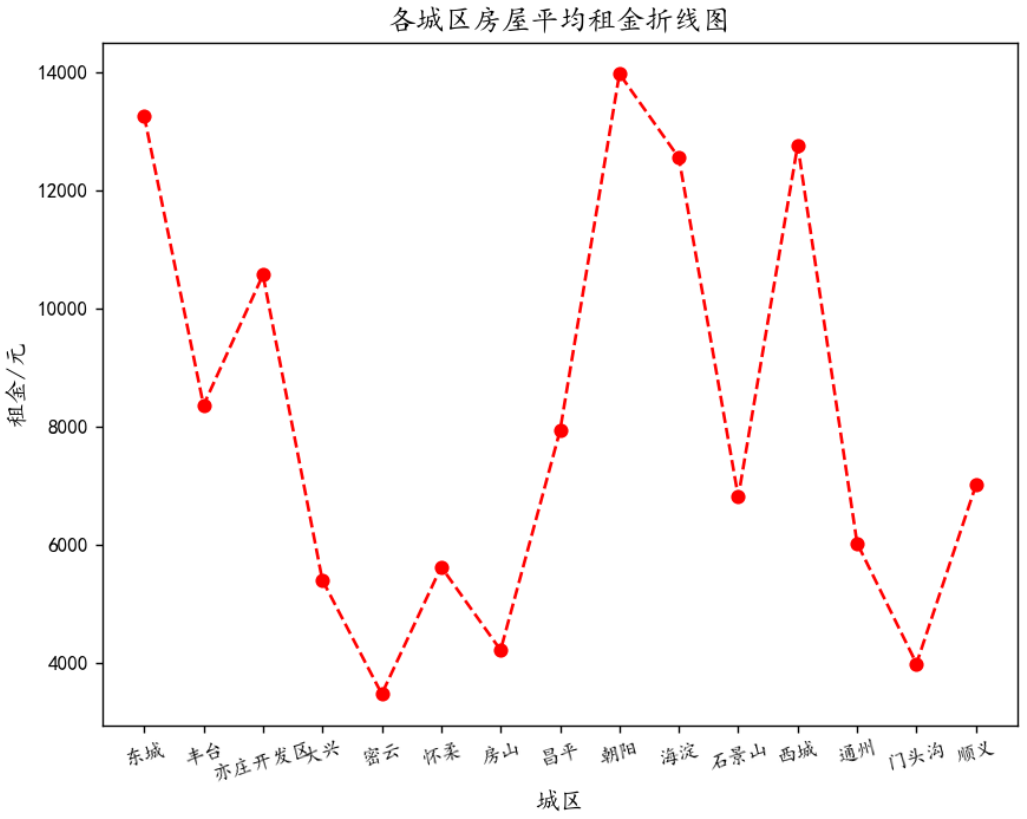
2022年10月16日中国共产党第二十次代表大会在北京胜利召开，习近平主席做了重要报告。《党的二十大报告》对未来一系列重大问题和决策作出整体部署和战略谋划。


06

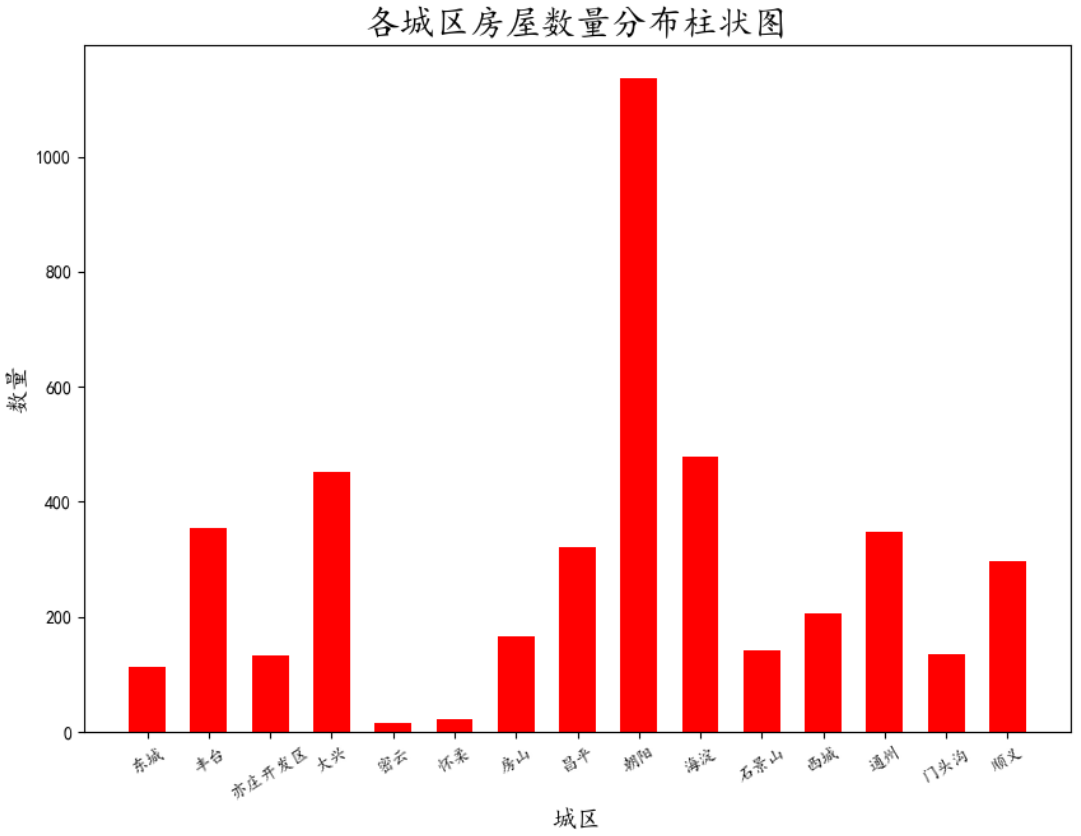
# 案例：房租价格可视化

# 案例：房租价格可视化

- 各地区平均房价

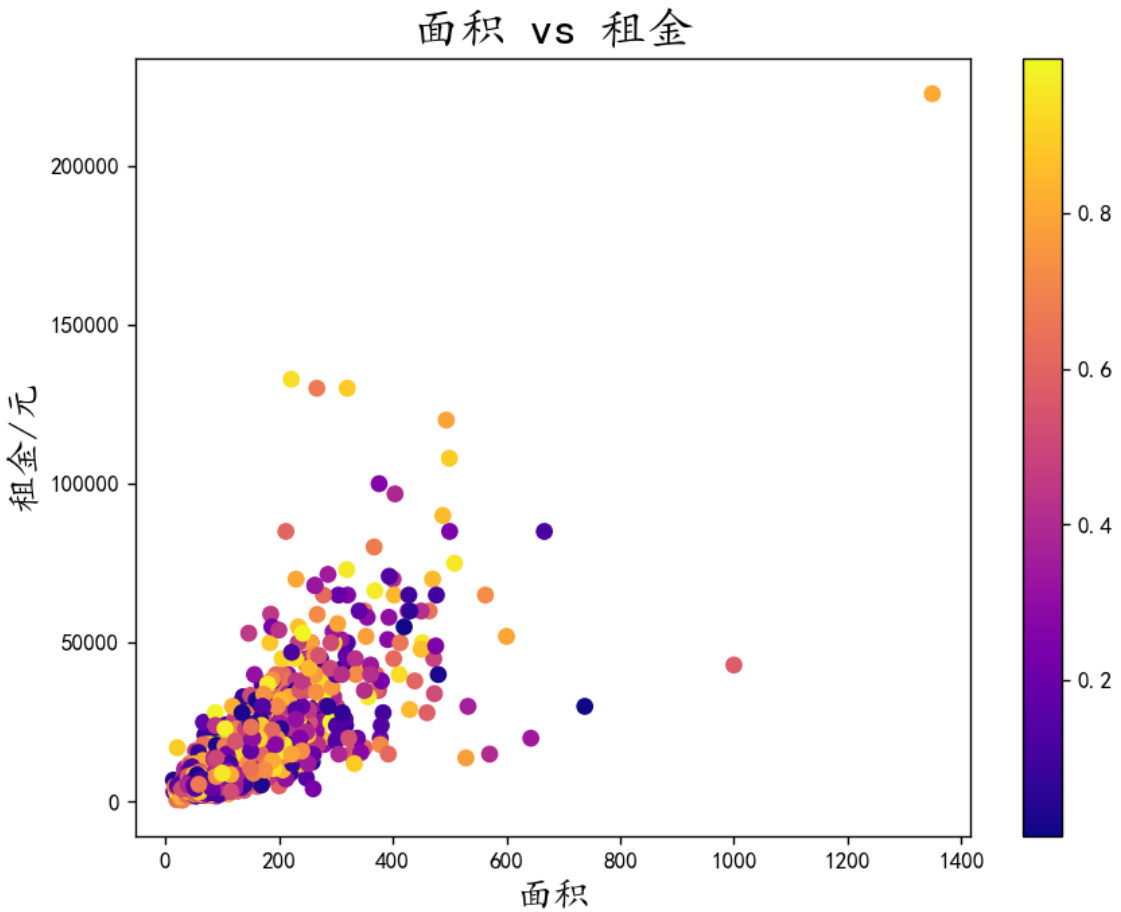



- 房屋数量分布

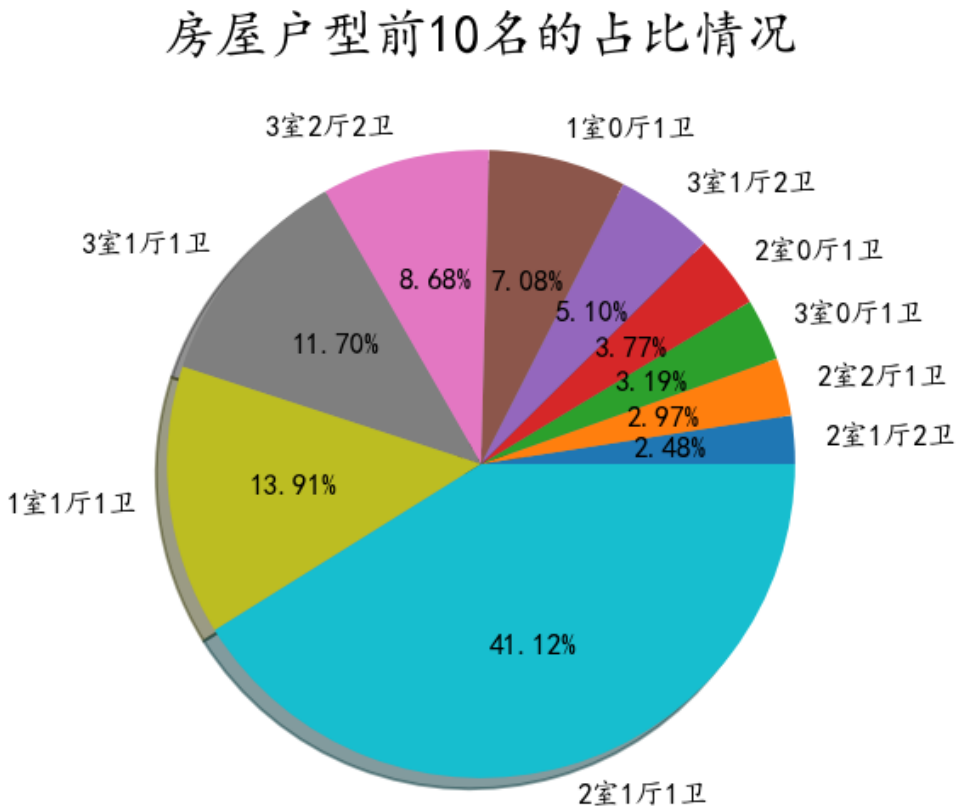


# 案例：房租价格可视化

- 面积 vs 价格




- 房屋户型前10名的占比



# 案例：房租价格可视化

- **词云图**