



北京航空航天大学
BEIHANG UNIVERSITY

大学计算机基础 (理科类)

第11讲 数据可视化

北京航空航天大学



目 录

11.1 数据可视化基础

11.2 使用Matplotlib绘制数据图形



本讲重点和难点

重点

- Matplotlib的使用方法
- 掌握折线图的绘制
- 掌握柱状图的绘制

难点

- 如何选择合适的可视化图形？





北京航空航天大学
BEIHANG UNIVERSITY



11.1 数据可视化基础

北京航空航天大学



数据可视化技术

- **数据可视化技术：** 将数据集中每一个数据项作为单个图元元素（如柱形、折线、气泡、圆点、扇面）表示，同时将数据的各个属性值以多维数据的形式表示，可以从不同的维度观察数据，从而对数据进行更深入的观察和分析

- **数据可视化技术应用领域**

- ◆ 数据采集
- ◆ 数据分析
- ◆ 数据管理
- ◆ 数据挖掘
- ◆ 科学计算结果展示
- ◆ 科技论文撰写

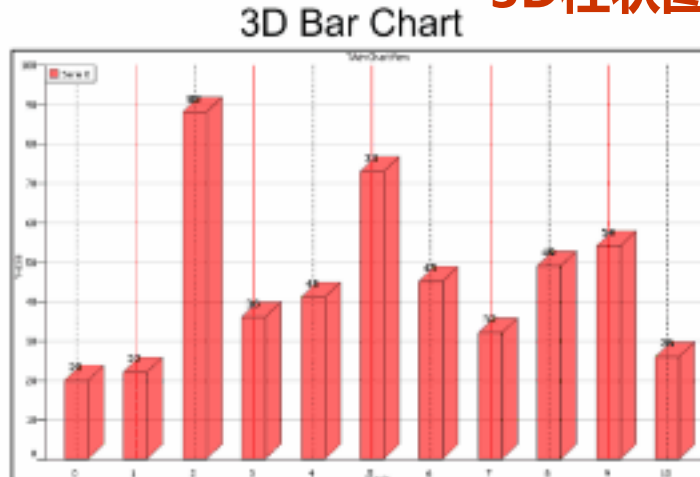


典型图形展示方式

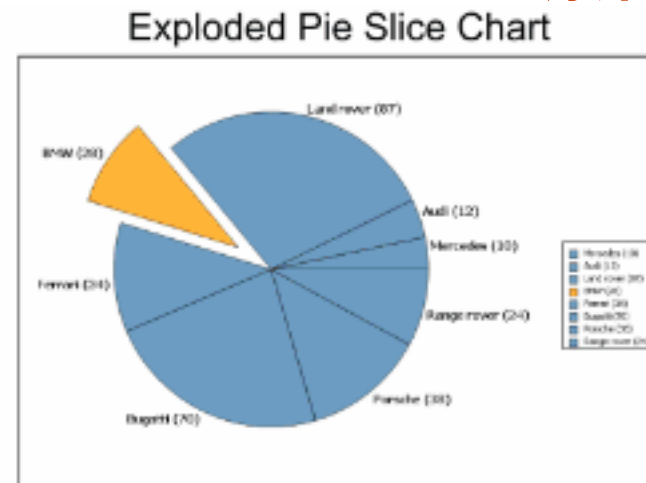
高低图



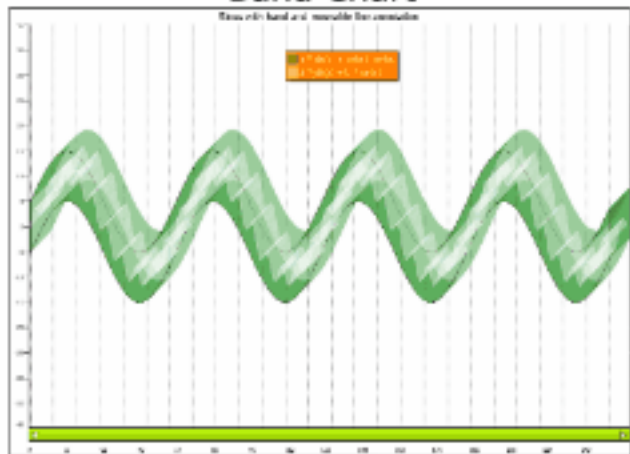
3D柱状图



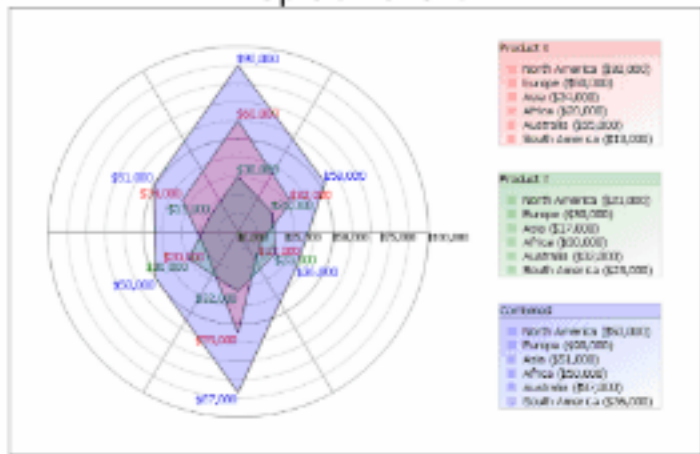
爆炸饼切片图



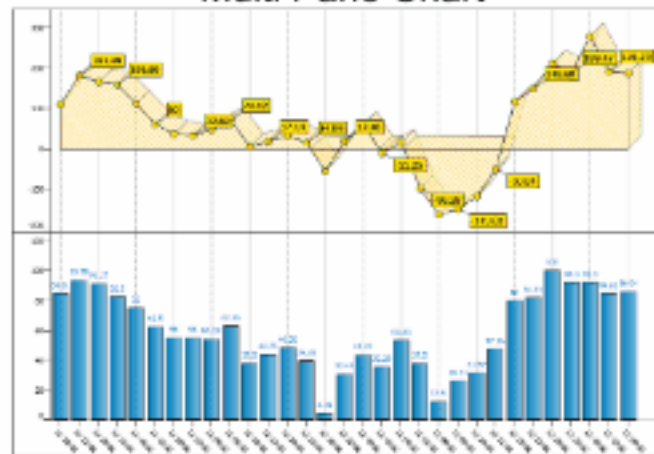
带状图



蛛网图 (雷达图)



多窗格图



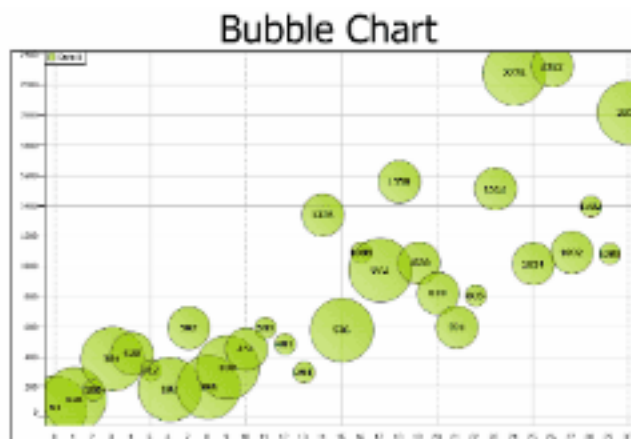
带状图

蛛网图 (雷达图)

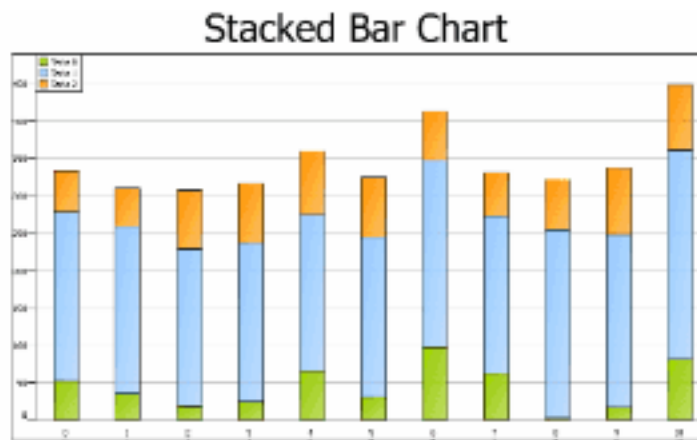
多窗格图

典型图形展示方式（续）

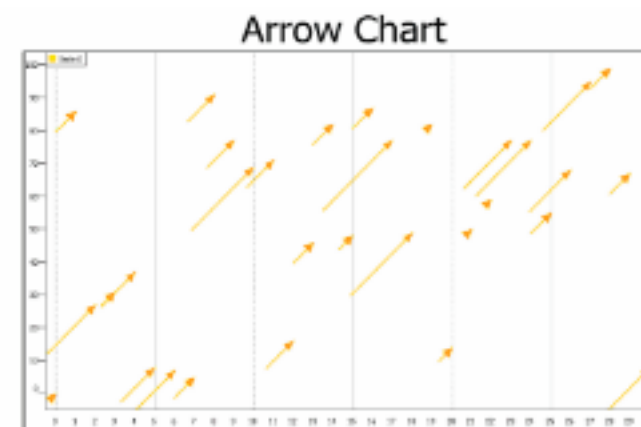
气泡图



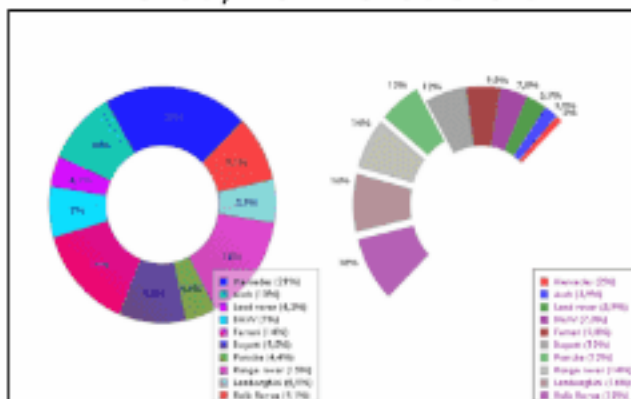
堆叠柱状图



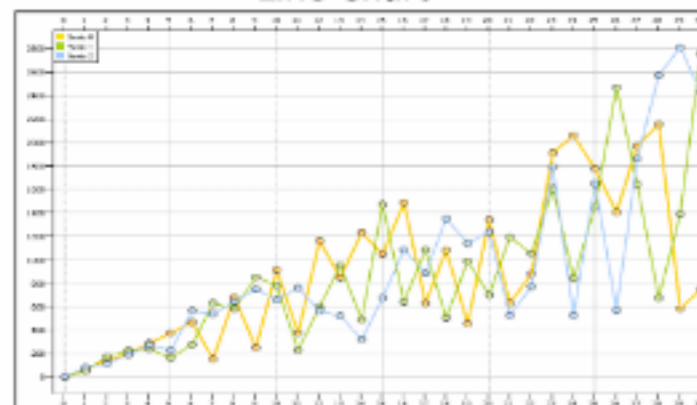
箭头图



圆环图 / 半圆环图

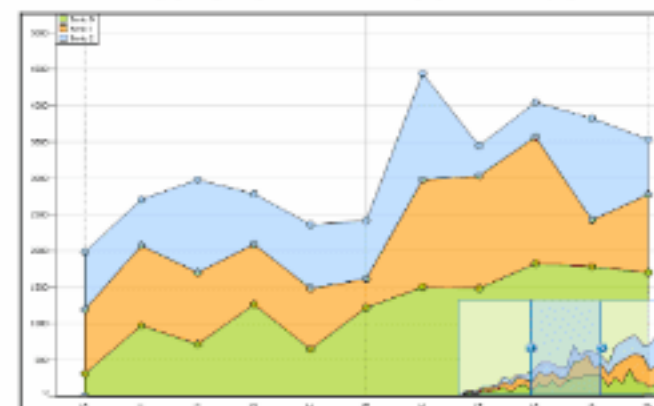


折线图



箭头图

带缩放窗口的区域图



圆环图/半圆环图

折线图

带缩放窗口的区域图

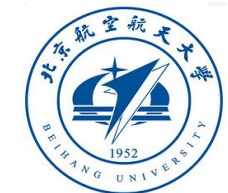




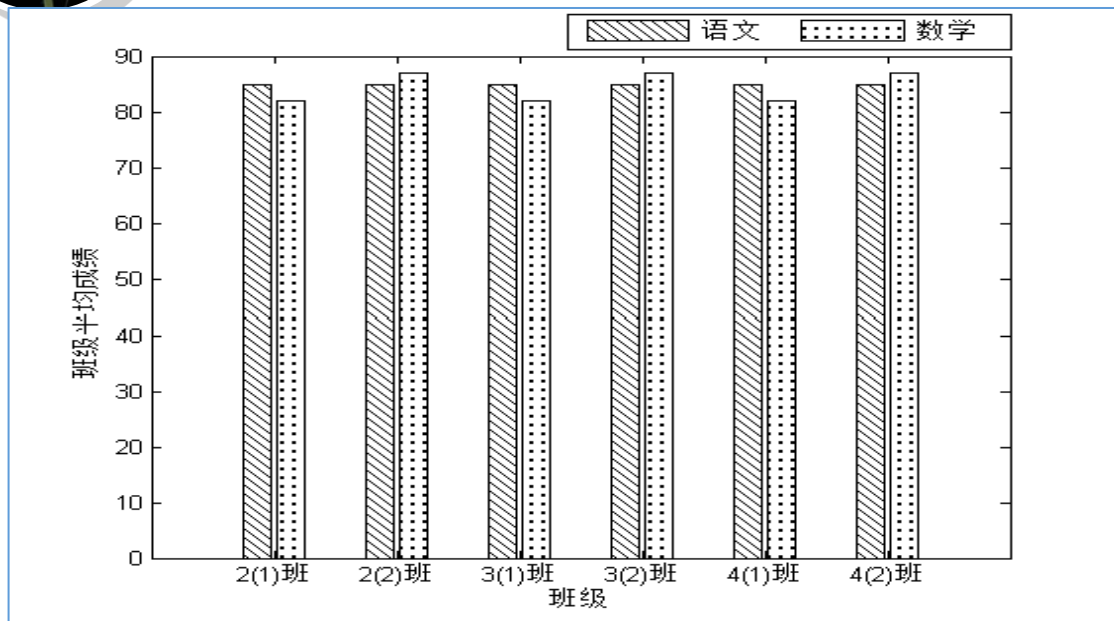
1、柱状图

1、柱状图 (Bar Chart)

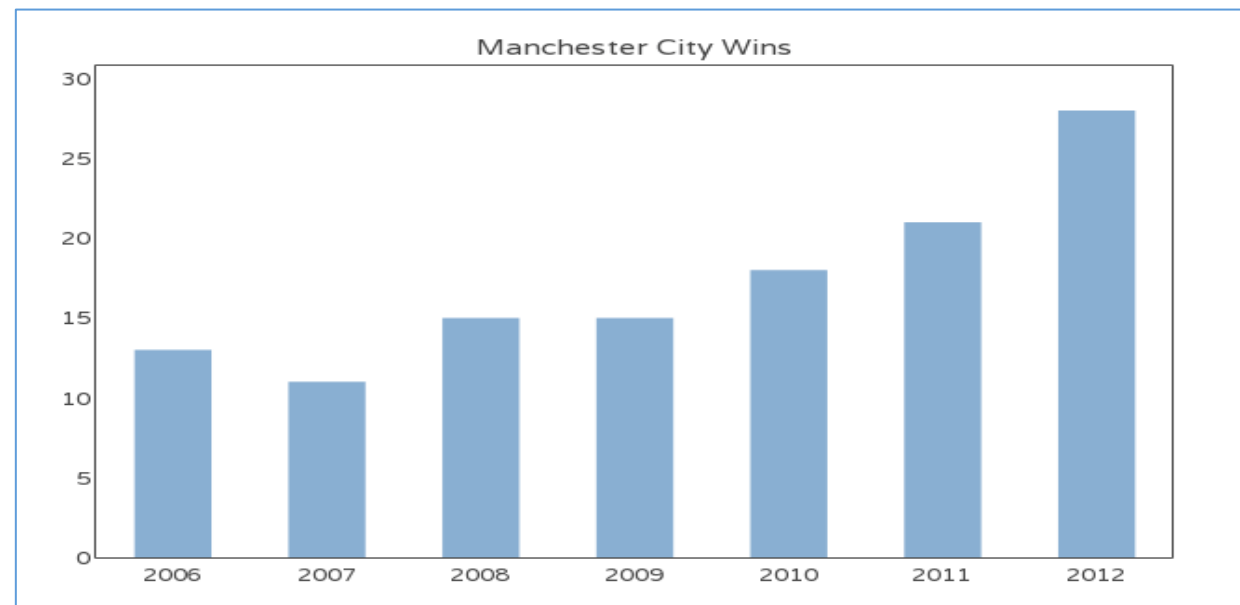
- ◆ **柱状图**是一种以长方形的**长度**为变量的表达图形的统计报告图，
由一系列高度不等的**纵向条纹**表示数据分布的情况，用来比较
两个或两个以上的数值（不同时间或者不同条件下）
- ◆ 通常用于**较小**的二维数据集分析，只比较其中**一维**数据
- ◆ **用途：**主要用于比较每个数据点的大小



柱状图示例



班级平均成绩的柱状图



各年销售额柱状图

- 如果有**多个数据集**，且是**黑白打印**，则每个数据集的柱状图最好选择不同的底纹显示，以便区分不同的数据集



2、折线图

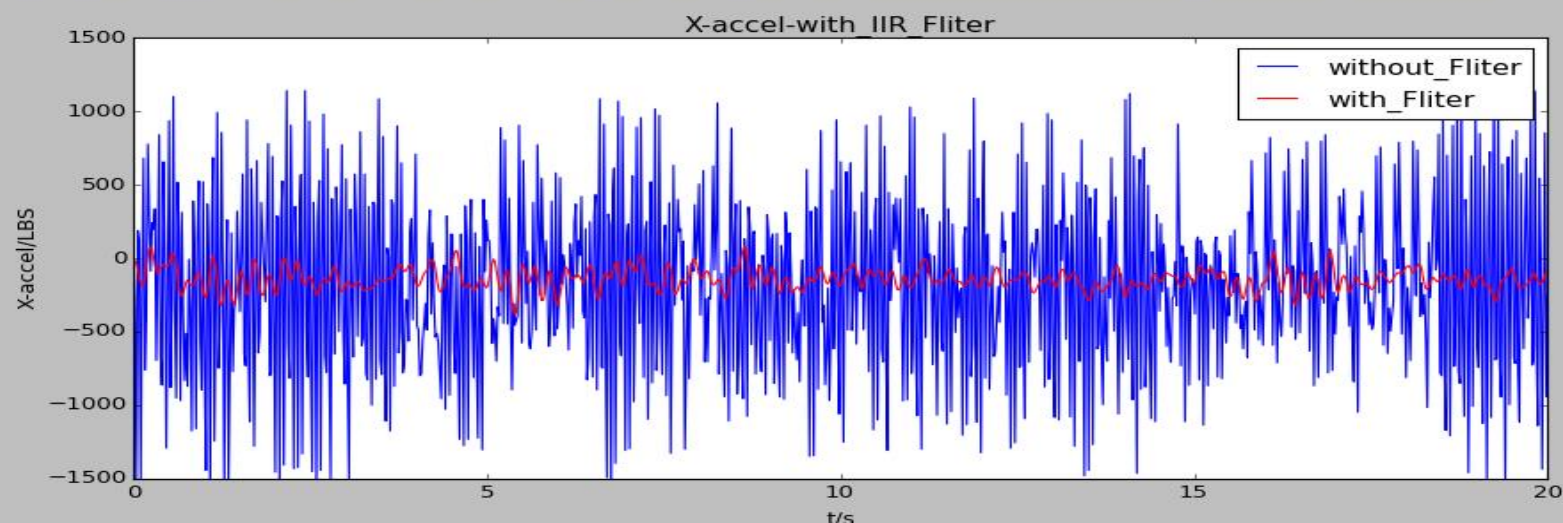
2、折线图 (Line Chart)

- ◆ **折线图**用一个**单位长度**表示一定的数值，根据数值的大小描出各数据点（每个数据点包括两个值：x和y）的位置；然后用线段把各点顺次连接起来，形成一条**连续的折线**
- ◆ 折线图不但可以表示数据项的具体数值，还能清楚地反映事物**变化**的情况
- ◆ 适合**二维**的**大**数据集，尤其是那些**趋势**比单个数据点更重要的场合

- ◆ **用途**：主要用于观察整个数据的**趋势**，或者比较**不同的数据集**的走势（如股票的K线图）

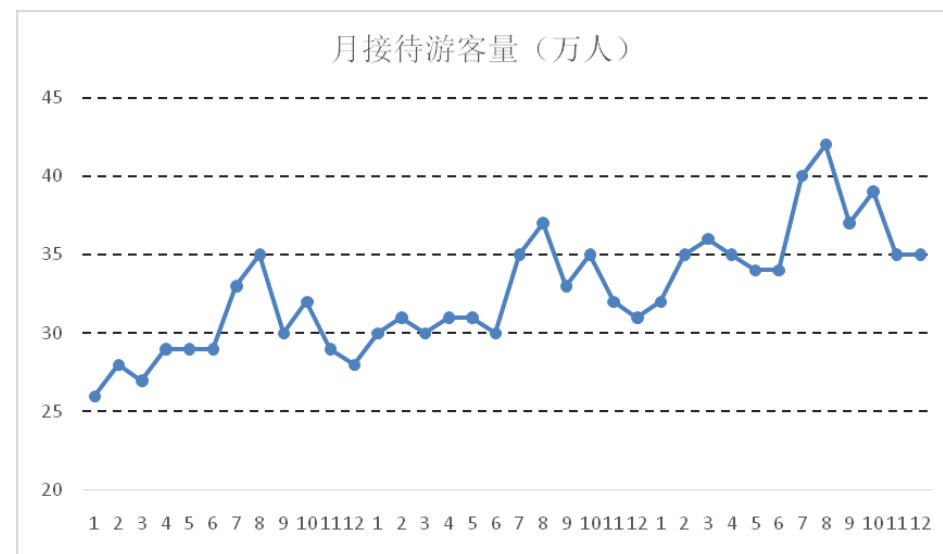


折线图示例



滤波前后实验数据比较折线图

某景点近三年
月接待游客量





3、饼图

3、饼图 (Sector Graph, Pie Graph)

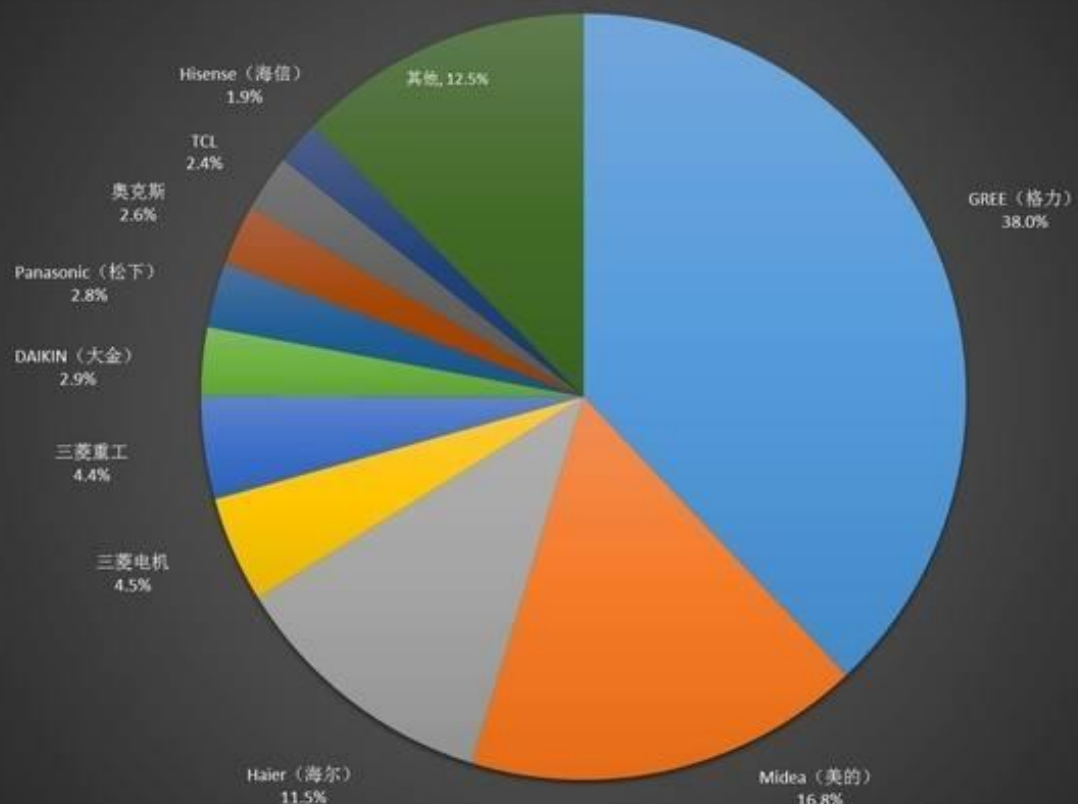
- ◆ **饼图**用于显示一个**数据系列**中各数据项的大小，以及每项数值与各项总和的比例关系。饼图中的数据点反映了某个部分占整体的**比重**，显示为整个饼图的**百分比**
- ◆ 饼图适合于**二维**的数据集
- ◆ 饼图以**二维**或**三维**格式显示每一数值相对于总数值的大小。可以手动拖出饼图的扇面来强调它们

◆ **用途**：主要用于观察每项数据在整个数据集里所占的比例



饼图示例

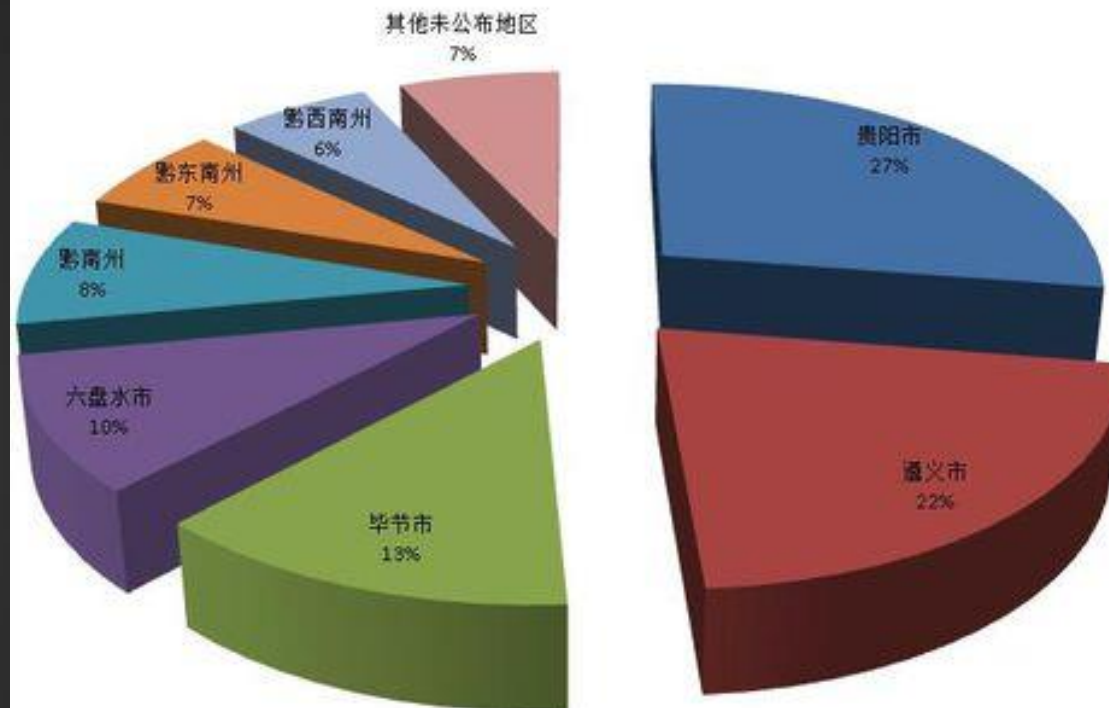
2017年Q3中国空调市场品牌关注比例分布



数据来源：互联网消费调研中心 (ZDC.zol.com.cn)



2012年贵州省前三季度GDP



2012年贵州省前三季度GDP饼图

空调品牌关注度比例



4、散点图

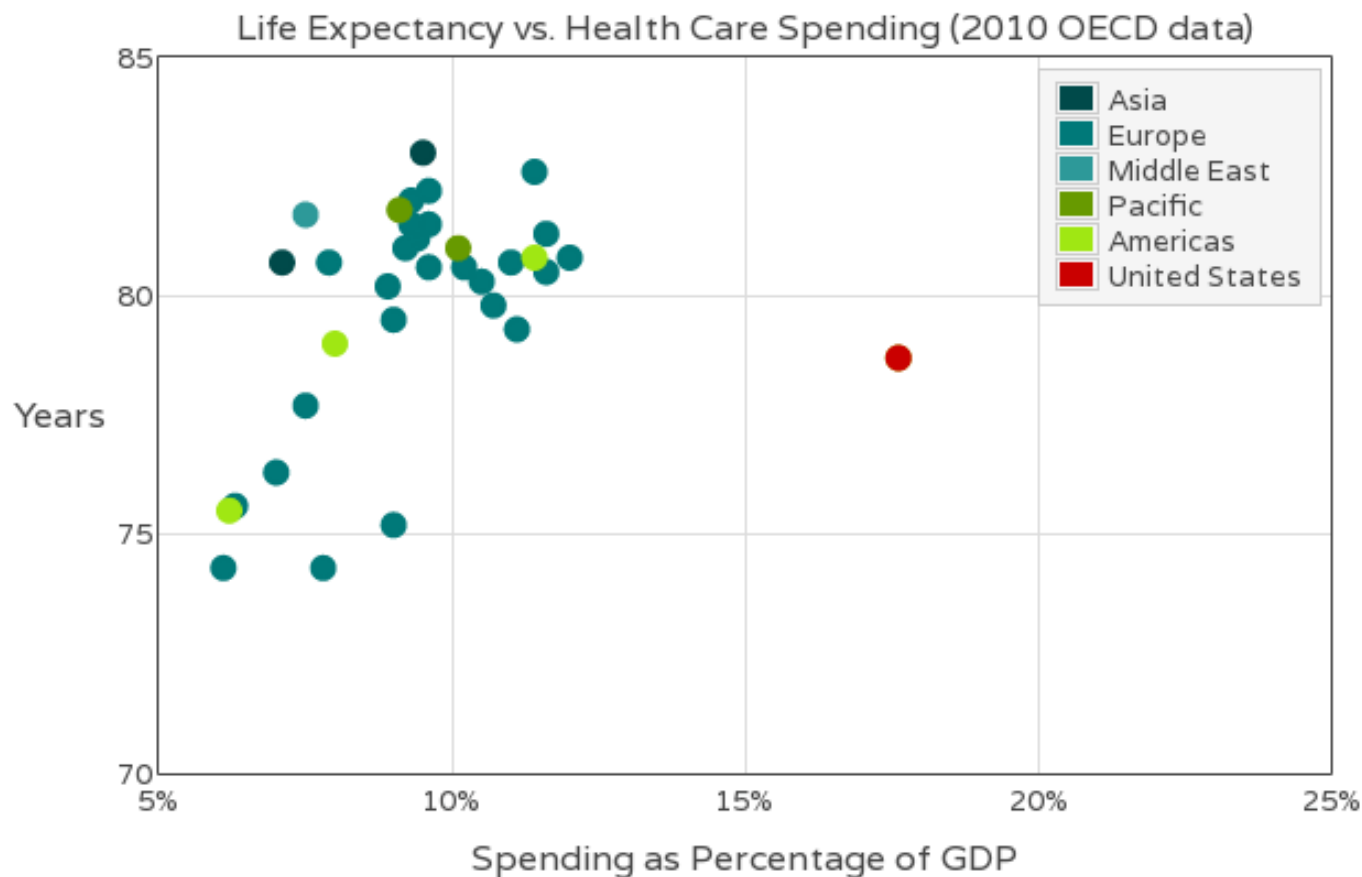
4、散点图 (Scatter Diagram)

- ◆ **散点图**将数据序列显示为一组点。值由点在图表中的位置表示，类别由图表中的不同标记（颜色或图形）表示
 - ◆ 当需要观察**因变量**随**自变量**而变化的大致**趋势**，或者大规模数据的**聚合特征**时，宜采用散点图
 - ◆ 适用于**二维**或**三维**数据集；当用于三维数据集时，其中只有两维需要比较，另一维用不同标记表示
- ◆ **用途：**大规模数据的比较，便于观察**聚合特征**



散点图示例

【例】分析各国居民的预期寿命与医疗支出是否存在线性关系。三个维度分别为**国家**、**医疗支出**、**预期寿命**，只有后两个维度需要比较





5、雷达图

5、雷达图（蛛网图）（Radar Chart , Spider Chart）

- ◆ **雷达图**将不同数据系列的各个**数据项**画在一个圆形图的不同轴向上，各数据项的数值大小通过**半径**的大小来区分，从而能够直观比较不同数据系列的同一个数据项的大小。
- ◆ 适用于多维数据（**四维以上**），且每个维度必须可以**排序**
- ◆ **局限**：数据系列最多不宜超过**7**个，否则无法辨别，因此适用场合有限

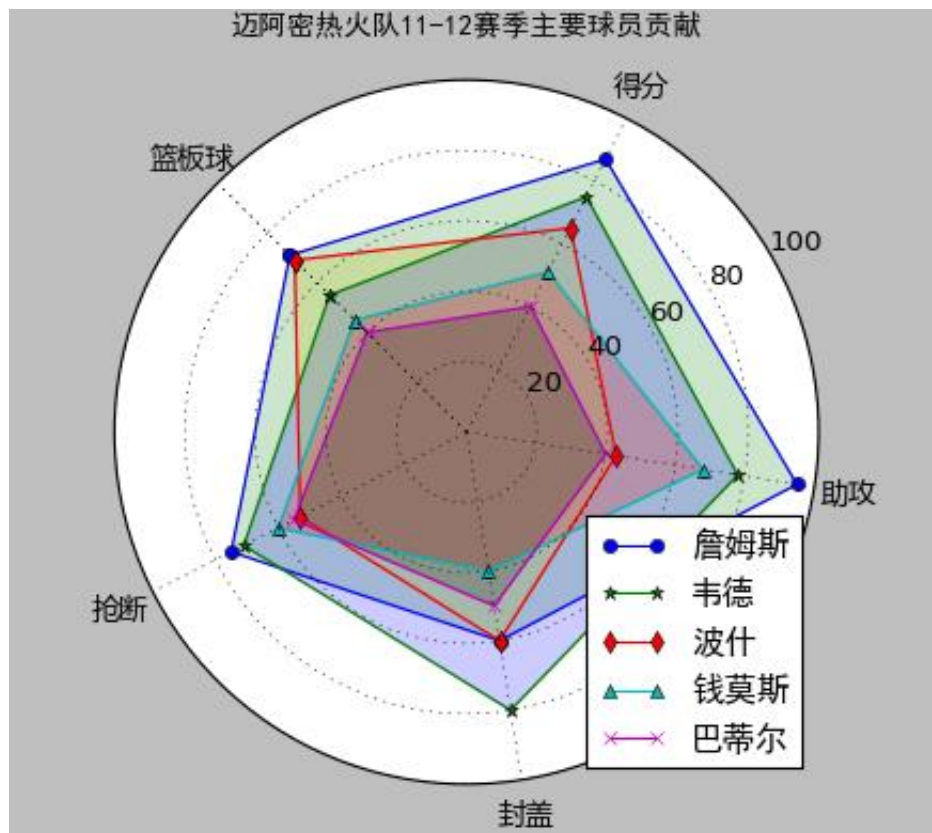
◆ **用途**：常用于比较**不同数据系列**的**多个属性值**



雷达图示例

◆ **【例】** 比较2011-12赛季迈阿密热火队首发的五名篮球选手在比赛对球队的贡献。除了姓名，每个数据系列有**五个维度**，分别是**得分、篮板球、助攻、抢断、封盖**。

同一数据序列的所有属性值所围成的图形用某种颜色表示——采用**不同的颜色**区分不同的**数据系列**





典型图形展示方式的比较

图表	维度	特点
柱状图	二维	适用于 小 数据集，侧重于比较每个数据点的 大小
折线图	二维	适用于 较大 的数据集，侧重于比较同一数据集整个数据走势或不同数据集的 趋势
饼图	二维	侧重于反映 部分与整体 的 比例 关系
散点图	二维或三维	有两个维度需要比较。适用于 大规模 数据集，便于观察 聚合 特征或因变量随自变量变化的趋势
雷达图	四维以上	适用于 小 数据集，常用于比较不同数据系列的 多个属性值 。数据系列最好不超过 7 个

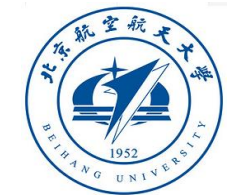


常用的数据可视化软件：（1）Excel

■ 常用的数据可视化软件有Excel、Matlab和Matplotlib等

（1）Excel

- ◆ 美国微软公司Office套件中软件之一，一个功能强大的电子表格软件
- ◆ 具有强大的**数据输入、编辑、计算与分析处理**功能
- ◆ **绘图功能**：将数据用**表格**及各种**统计图、透视图**的形式表示
- ◆ 柱形图、折线图、饼图、条形图、面积图、散点图、股价图、曲面图、圆环图、气泡图、雷达图
- ◆ **不足**：但Excel的**绘图功能**相对**弱小**，展示方式比较单调，难以满足科学计算中丰富的展示要求





(2) MATLAB

(2) MATLAB

- 美国**Mathworks**公司开发的商业数学软件
- 用于**算法开发**、**数据可视化**、**数据分析**以及**数值计算**的高级技术计算语言和交互式环境
 - ◆ **功能**：数值计算、绘图、符号计算、系统建模、仿真与分析
 - ◆ **优势**：强大的**矩阵计算**以及**仿真**能力
 - ◆ 丰富的图形绘制功能：可以绘制各种**二维曲线和图形**，**三维图形**（三维线图、网线图、曲面图）
 - ◆ **不足**：需要基于自身的**脚本语言**开发程序，给使用者提出了较高的使用要求





(3) Matplotlib

(3) Matplotlib

- 基于Python的**第三方绘图库**，它提供了匹敌（甚至超过）MATLAB的**专业级科学计算结果展示能力**
- 实际上是一套**面向对象**的绘图库，所绘制图表中的每个**绘图元素**（折线、柱形、x轴标签、y轴标签、图标题……），在内存中都有一个**对象**与之对应
- 通过**matplotlib.pyplot**模块提供了一套绘图API（应用程序接口）
 - ◆ 用户只需调用pyplot模块所提供的**函数**，就可以实现**快速绘图**以及**设置图表**的各种细节





Matplotlib 的功能

- Matplotlib风格与**Matlab**很相似，同时继承了Python简单明了的风格，可以很方便地绘制各种**二维**和**三维**图形
 - ◆ 二维曲线/折线图
 - ◆ 柱状图
 - ◆ 散点图
 - ◆ 等高线图
 - ◆ 灰度图
 - ◆ 饼图
 - ◆ 量场图 (Quiver Plots)
 - ◆ 极轴图
 - ◆ 三维图





NumPy: 科学计算基础库

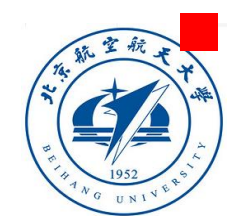
- 使用Matplotlib 绘图, 有时还需要利用NumPy库**生成数据**
- **NumPy的主要功能**
 - ◆ 可以创建一维、二维、N维数组, 支持**N维数组运算**
 - ◆ 内置许多ufunc (通用) 函数
 - ✓ 分为**一元func**, 对**一个数组**进行运算的函数。如**abs**: 计算绝对值, **sqrt**: 计算各元素的平方根, **square**: 计算各元素的平方, 三角函数**cos**, **sin**等
 - ✓ **二元func**, 对**两个数组**进行运算的函数。如**add**: 两个数组对应元素相加, **subtract**: 两个数组对应元素相减.....
 - ◆ 处理**大型矩阵**, 成熟的广播函数库, 矢量运算
 - ◆ 线性代数, 傅里叶变换、随机数生成
 - ◆ 提供用于读写硬盘上基于数组的数据集的工具
 - ◆ 提供用于将C、C++、Fortran代码集成到Python的工具





matplotlib.pyplot模块

- Matplotlib通过**matplotlib.pyplot**模块来提供绘图功能
- matplotlib.pyplot模块提供了一批**操作**和**绘图函数**（**方法**），如创建绘图区域、绘制折线图、添加标注或修改坐标轴等
 - ◆ **figure函数**用于创建一个全局绘图区域
 - ◆ **bar函数**用于绘制柱状图
 - ◆ **plot函数**用于绘制折线图
 - ◆ **xlabel函数**用于用户设置坐标系X轴的说明（x轴标签）
 - ◆ **ylabel函数**用于用户设置坐标系Y轴的说明（y轴标签）
- 从Matplotlib的官网，可以查阅所有其支持的绘制函数
http://matplotlib.org/api/pyplot_summary.html





matplotlib.pyplot模块的主要方法

序号	方法	描述
1	annotate	为指定数据点添加一段注解文字
2	axes	为figure对象添加一个axe，即添加一个新的子图
3	bar	绘制柱状图
4	figure	创建一个绘图对象，类似于创建一张画布
5	fill	在指定多边形内部填充颜色
6	grid	控制axe的栅格的显示与关闭
7	legend	生成图例说明
8	pie	绘制饼图
9	plot	在当前axe中绘制曲线或者离散数据点
10	polar	绘制极坐标图
11	sca	设置当前活动axe



matplotlib.pyplot模块的主要方法（续）

序号	方法	描 述
12	<code>subplot</code>	在全局绘图区域内创建子绘图区域
13	<code>text</code>	为当前axe添加文字描述
14	<code>title</code>	为当前axe添加标题
15	<code>xlabel</code>	为当前axe添加x轴标签
16	<code>xlim</code>	为当前axe添加x轴上限、下限
17	<code>xscale</code>	为当前axe设置x轴尺度
18	<code>xticks</code>	为当前axe设置x轴的刻度
19	<code>ylabel</code>	为当前axe添加y轴标签
20	<code>ylim</code>	为当前axe添加y轴上限、下限
21	<code>yscale</code>	为当前axe设置y轴尺度
22	<code>yticks</code>	为当前axe设置y轴的刻度





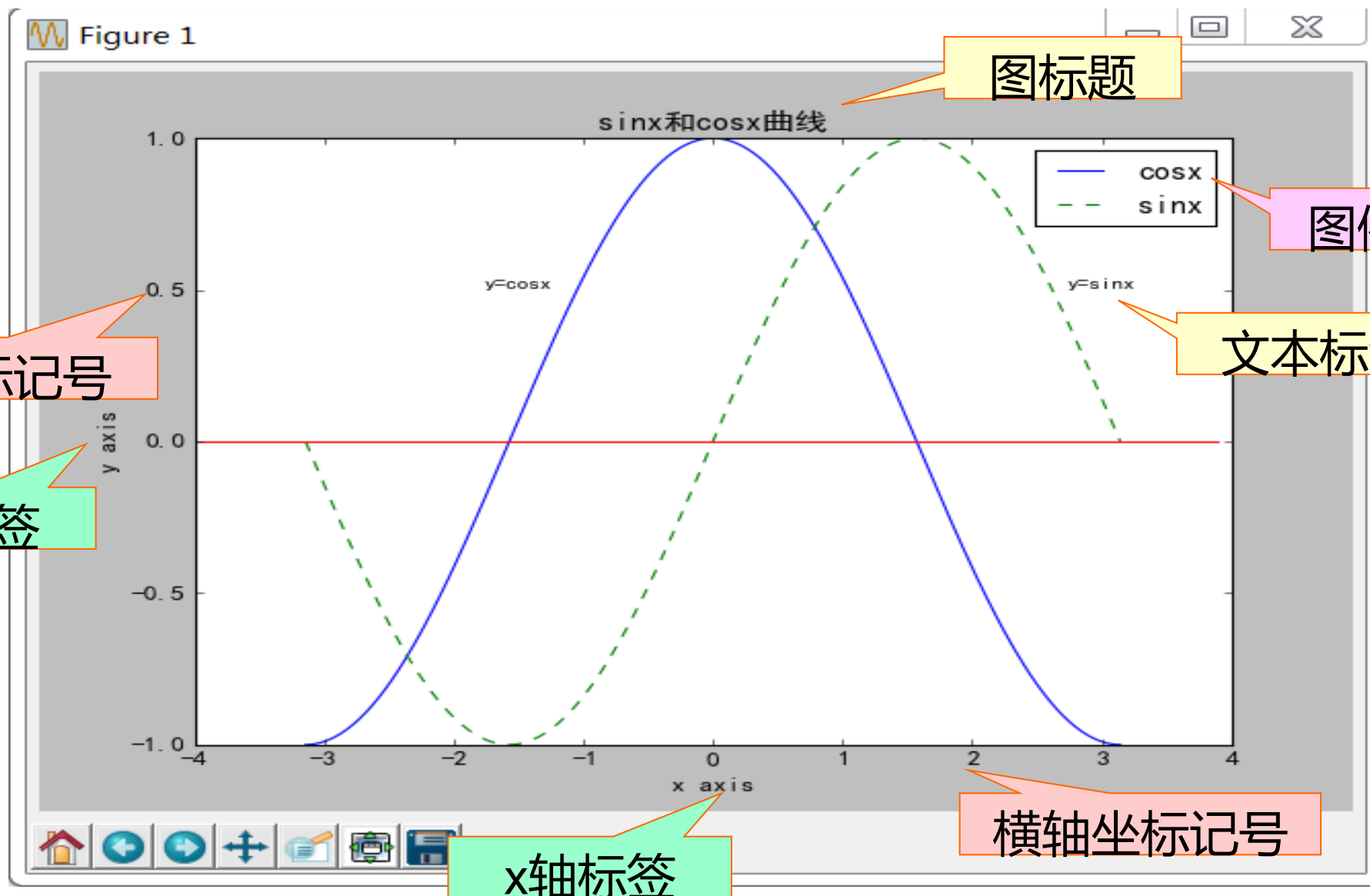
二维图形可能包含的元素

■ 一个图表中一般包括哪些元素？ 如何设置？

- ◆ 图标题 (title函数)
- ◆ x轴标签 (xlabel函数)
- ◆ y轴标签 (ylabel函数)
- ◆ x轴记号/刻度 (xticks函数)
- ◆ y轴记号/刻度 (yticks函数)
- ◆ 文本标注 (text函数)
- ◆ 颜色 (plot函数或bar函数中的color参数)
- ◆ 图例 (legend函数)



二维图表示例





北京航空航天大学
BEIHANG UNIVERSITY

11.2 使用Matplotlib 绘制数据图形

北京航空航天大学



11.2.1 绘制折线图

- 绘制折线图，调用matplotlib.pyplot模块中的plot函数
- **plot函数**用来绘制**二维折线/曲线**，或者绘制**离散的数据点**
- 其用法与MATLAB二维曲线绘图指令非常相似

格式 `plot(x,y,type)`

如： `plt.plot(x1, y1, 'ko-')` #绘制离散数据点，黑色，形状为实心圆点；绘制连续曲线，线形为实线

或： `plot(X, C, color="blue", linewidth=1.0, linestyle="-", label="cosx")` #label指定了图例的文字说明



plot函数的参数

■ 输入参数

x: 数据点的横坐标值, **列表、元组或数组**

y: 数据点的纵坐标值, **列表、元组或数组**

type: **绘图样式** (颜色, 离散数据点的形状, 曲线的线形、线宽等)

◆ **方法一**: 用**字母**和**符号**来表示绘图样式 (并用**单引号**或**双引号**括起来)

如: `plt.plot(x1, y1, 'ko-')` #黑色, 离散数据点形状为实心圆点, 曲线线形为实线

◆ **方法二**: 用**参数**指定绘图样式

color (颜色), **linewidth** (线宽), **linestyle** (曲线的线形), **label** (图例的文字说明)

如: `plt.plot(x1, y1, color='black', marker='o', linestyle='-')`





type参数指定绘图样式

■ type参数指定绘图样式

- ◆ **颜色**: b(蓝)、g(绿)、r(红)、c(青)、m(紫)、y(黄)、k(黑)、w(白)
- ◆ **离散数据点形状**: .(黑点)、+(加号)、*(星号)、o(实心圆点)、d(菱形)、p(五角星)、h(六角星)、x(X号)、s(方块)、...
- ◆ **连续线型**: -(实线)、:(虚线)、-.(点划线)、--(虚划线)

例如: 'ko-' : 黑色线, 圆形数据点, 实线





绘制折线图的应用场合

■ 两种情况

- ◆ (1) 已知一系列数据点 (x,y) ，要求绘制由所有数据点连成的折线图

- ✓ 直接使用plot函数绘制

- ◆ (2) 已知 x 的范围 $[a,b]$ 和某一函数 $f(x)$ ，要求绘制函数曲线

- ✓ 在 X 轴上 $[a,b]$ 区间内利用NumPy库提供的arange函数或linspace函数，生成一组 x 坐标值

- ✓ 根据函数，对应各 x 值计算 y 值，得到一组数据点 (x,y)

- ✓ 再利用plot函数绘制

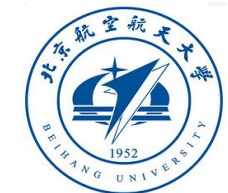




教材【微实例5.6】绘制折线图示例1

- ◆ (1) 已知一系列数据点 (x,y) ，要求绘制由所有数据点连成的折线图

- **教材【微实例5.6】** 某毕业班6个学期期末语文成绩绘制。
- 某学校统计了毕业班过去6个学期的期末考试成绩，某班1~6学期的语文成绩平均分为 $[78, 85, 80, 82, 79, 89]$ ，请绘制出该班的语文成绩的折线图。





【微实例5.6】程序

微实例5.6-line.py

(1) 导入三方库

import matplotlib.pyplot as plt #导入matplotlib.pyplot模块，别名取为plt

(2) 为了在生成的图像中正确显示中文，需要设置字体属性

from matplotlib.font_manager import FontProperties #导入FontProperties函数

#设置字体对象，本例选择的是简宋字体，字号是14

font = FontProperties(fname=r'c:\windows\fonts\simsum.ttc', size=14)

(3) 用列表存储(x,y)数据

用**元组**也可以

x1 = [1,2,3,4,5,6]

y1 = [78,85,80,82,79,89]

(4) 绘制折线图

plt.plot(x1, y1, 'ro-', linewidth=2.5)

使用**库名.函数名()**调用模块中的函数



【微实例5.6】程序（续）

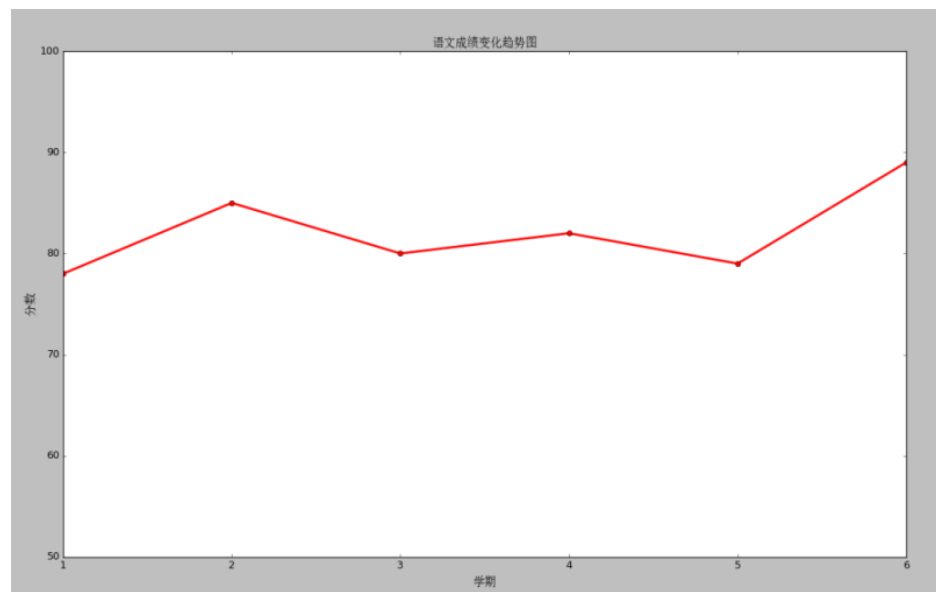
(5) 美化图表

必须指定字体属性

```
plt.xlabel('学期', fontproperties=font) #设置x轴标签 (x轴含义)
plt.ylabel('分数', fontproperties=font) #设置y轴标签 (y轴含义)
plt.title('语文成绩变化趋势图', fontproperties=font) #设置图标题

plt.ylim(50,+100) #设置纵轴的上下限

plt.show() #使图在屏幕上显示
```





绘制折线图——子图

- **问题：怎样在一张图纸上绘制多幅图形（每个有独立的坐标系），以进行比较？**

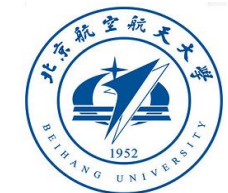




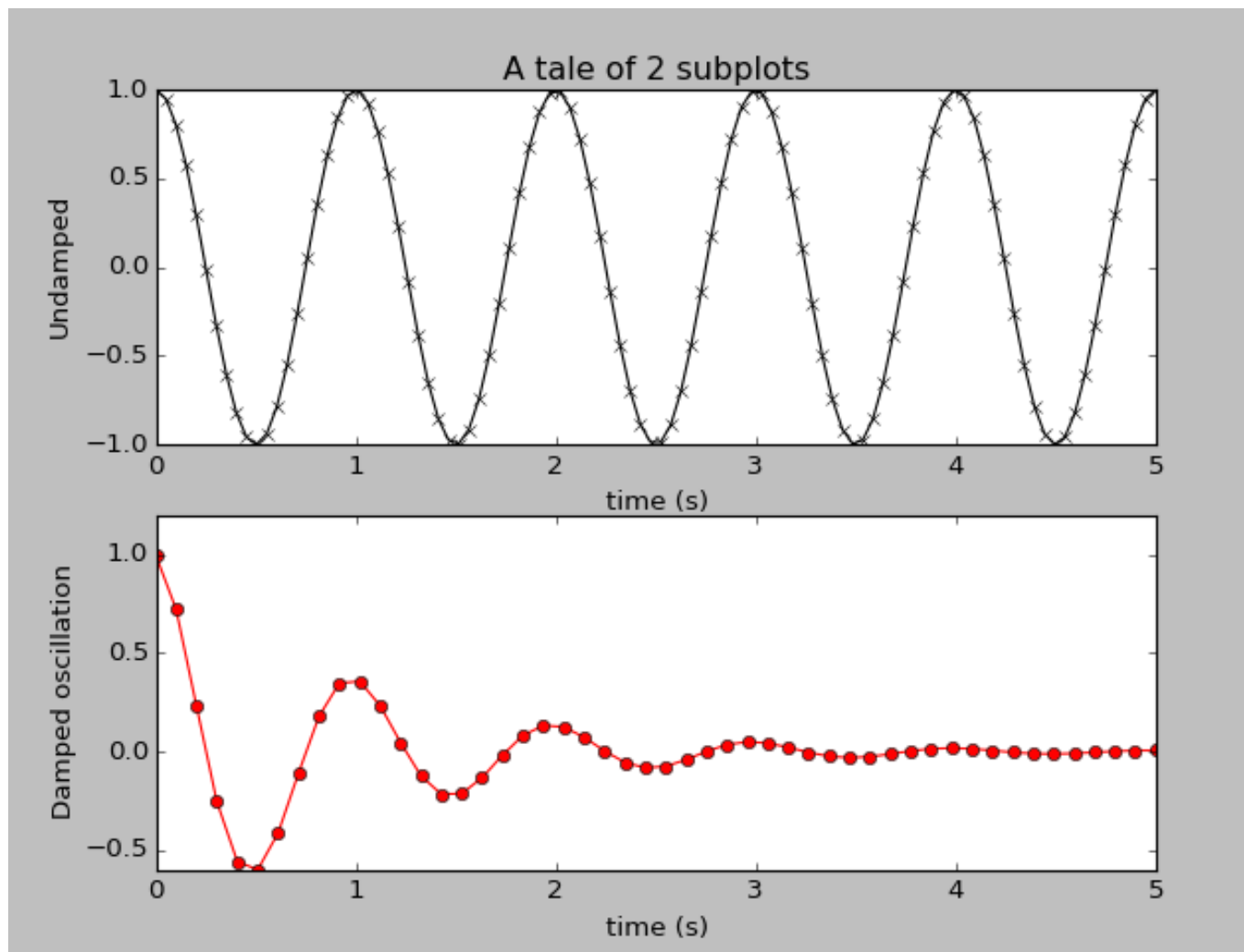
【例11.1】绘制折线图示例2

- ◆ (2) 已知 x 的范围 $[a,b]$ 和某一函数 $f(x)$ ，要求绘制函数曲线

【例11.1】 在2张子图上绘制函数 $y=\cos(2\pi x)$ 和 $y=\cos(2\pi x) * e^{-x}$ 在区间 $[0.0, 5.0]$ 内的曲线。分别在两条曲线的旁边添加文字标注。



【例11.1】子图样式





技巧：如何在一张图表中绘制多个子图？

- 有时候，需要在**同一图形窗**中绘制多幅不同坐标系中的图形，以进行对比
- 两种方法：使用plt的**subplot()函数**或**subplots()函数**
- subplot()函数将整个绘图区域等分为**numRows行*numCols列**个子区域，然后按照从左到右、从上到下的顺序对每个子区域进行编号，左上的子区域的编号为1
 - ◆ **创建子绘图区域后，即在该语句的下面编写绘制曲线的语句，就是在该子图中绘图**





subplot函数的用法

■ subplot函数的调用形式

subplot(*numRows, numCols, plotNum*)

subplot函数**每次**只创建**一张子图**，通过
plotNum参数指明子图的**编号**（即位置）

1	2	3
4	5	6

◆ 在**例11.1-subplot.py**程序中：

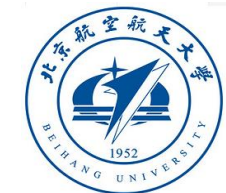
plt.subplot(2,1,1) # 创建**子图1**

plt.plot(x1, y1, 'kx-') #绘制离散点和连续曲线

.....

plt.subplot(2,1,2) # 创建**子图2**

plt.plot(x2, y2, 'ro-') #绘制离散点和连续曲线



subplot函数的用法（续）

■ subplot函数的参数

(1) 输入参数

numRows : 子图的行数, 默认值为1

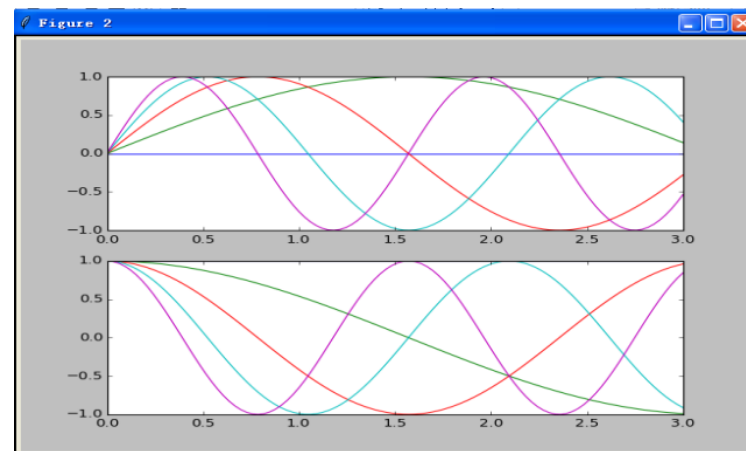
numCols : 子图的列数, 默认值为1

plotNum: 指明图样放在哪个子区域中

(2) 返回值: **(fig, ax)**元组

fig: matplotlib.figure.Figure对象 (**绘图区域**)

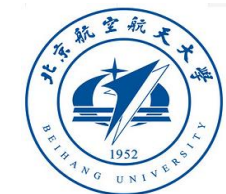
ax: 所创建的Axes对象 (**子图**), 可以将它用变量保存起来, 然后用sca()函数交替让它们作为当前Axes对象, 并调用plot()在其中绘图





技巧：如何生成一组x值？

- 利用Matplotlib绘制某个函数的图形，首先需要创建一个**一维数组**来存储一组x坐标值
 - ◆ **方法一**：利用numpy提供的**arange函数**，通过指定开始值、终值和**步长**（均可以是**浮点数**）来创建。**注意数组不包括终值！**
 - ◆ **方法二**：利用numpy提供的**linspace函数**，通过指定开始值、终值和**元素个数**来创建。通过endpoint关键字指定是否包括终值，缺省为包括终值





arange函数的用法

■ arange函数的用法

- ◆ 在区间[<开始值>,<终值>]之间以<步长>为步长生成一个包含等间距的若干个元素的**数组**

格式

arange(<开始值>,<终值>,<步长>)

```
>>> import numpy as np
```

```
>>> np.arange(0,1,0.1)
```

```
array([ 0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9])
```

此函数在区间[0,1]之间以0.1为步长生成一个**数组**。

- ◆ 第三个参数为**步长**，缺省时为1。也可以为负数

```
>>> np.arange(0,10)
```

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```





range函数

■ range函数

- ◆ range函数是Python**内建函数**，不必导入库
- ◆ 用于生成指定取值范围（**整数**）内的若干个**整数**，指定的范围包含下限，但不包括上限
- ◆ 下限为0时，可以省略。第三个参数为**步长**，缺省时为1。也可以**以为负数**

`range (<下限>, <上限>, <步长>)`

■ **注意：**上限、下限必须是整数，步长也必须是整数！否则出错！





linspace函数的用法

linspace函数的用法

(1) 在区间[<开始值>,<终值>]之间等间距生成包含<元素个数>指定个数的元素（为浮点数）的一个数组

格式 `linspace(<开始值>,<终值>,<元素个数>)`

```
>>> import numpy as np
>>> np.linspace(0, 1.2, 13.6)
array([ 0. ,  0.1,  0.2,  0.3,  0.4,  0.5,  0.6,  0.7,  0.8,  0.9,  1. ,  1.1,  1.2])
>>> np.linspace(0, 10, 11)
array([ 0.,  1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.]
```

linspace函数的用法 (续1)

- 开始值、终值可以为浮点数
- 若元素个数为浮点数，运算时向下取整
- 无论开始值、终值是否为整数，无论生成的元素的值是否为整数，显示时都以浮点数形式显示

(2) 第三个参数为元素个数，缺省时为50

>>> np.linspace(0,10) #缺省包括上限

```
>>> import numpy as np
>>> np.linspace(0,10)
array([ 0., 0.20408163, 0.40816327, 0.6122449 ,
        0.81632653, 1.02040816, 1.2244898 , 1.42857143,
        1.63265306, 1.83673469, 2.04081633, 2.24489796,
        2.44897959, 2.65306122, 2.85714286, 3.06122449,
        3.26530612, 3.46938776, 3.67346939, 3.87755102,
        4.08163265, 4.28571429, 4.48979592, 4.69387755,
        4.89795918, 5.10204082, 5.30612245, 5.51020408,
        5.71428571, 5.91836735, 6.12244898, 6.32653061,
        6.53061224, 6.73469388, 6.93877551, 7.14285714,
        7.34693878, 7.55102041, 7.75510204, 7.95918367,
        8.16326531, 8.36734694, 8.57142857, 8.7755102 ,
        8.97959184, 9.18367347, 9.3877551 , 9.59183673,
        9.79591837, 10.])
```



linspace函数的用法 (续2)

(3) 通过**endpoint**关键字指定是否包括终值，缺省为包括终值

>>> np.linspace(0,1,11,endpoint=True)

#生成0、0.1、0.2、.....、**1**共11个数

包括终值

```
>>> np.linspace(0,1,11)
array([ 0. ,  0.1,  0.2,  0.3,  0.4,  0.5,  0.6,  0.7,  0.8,  0.9,  1. ])
```

>>> np.linspace(0,1,11,endpoint=False)

#生成0、0.1、0.2、.....、**0.90909091**共11个数

不包括终值

```
>>> np.linspace(0,1,11,endpoint=False)
array([ 0. ,  0.09090909,  0.18181818,  0.27272727,  0.36363636,
        0.45454545,  0.54545455,  0.63636364,  0.72727273,  0.81818182,
        0.90909091])
```



【例11.1】的设计思路

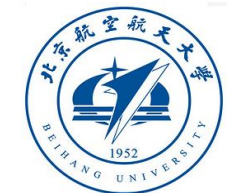
■ 程序包括4个部分

(1) 导入三方库

```
import numpy as np          #导入科学计算基础库，别名为np  
import matplotlib.pyplot as plt  #导入绘图库，别名为plt
```

(2) 生成绘制曲线所需的x、y坐标点

- ✓ 使用numpy库中`linspace`函数在指定]区间内等间隔产生100个或50个数据点的x值，数组形式
- ✓ 利用给定函数 $y = \cos(2 \pi x)$ 和 $y = \cos(2 \pi x) * e^{-x}$ 计算对应的y值





【例11.1】的设计思路（续）

(3) 创建子图，绘制曲线

先后创建2个子图，
并分别绘图

- ✓ 使用plt中的**subplot**函数创建子图（则成为当前绘图区域）
- ✓ 使用plt中的**plot**函数绘制曲线
- ✓ 使用plt中的**xlabel**函数创建x标签，**ylabel**函数创建y标签
- ✓ 使用plt中的**title**函数创建图标题（只需一个）

(4) 在屏幕上显示绘图对象

- ✓ 使用plt中的**show()**函数



【例11.1】程序

(1) 导入三方库

```
import numpy as np
import matplotlib.pyplot as plt
```

#导入科学计算基础库，别名为np
#导入绘图库，别名为plt

(2) 生成绘制曲线所需的x、y坐标点

```
x1 = np.linspace(0.0, 5.0, 100)
```

在指定区间等间隔产生100个数据点

```
x2 = np.linspace(0.0, 5.0)
```

#默认在指定区间等间隔产生50个数据点

```
y1 = np.cos(2 * np.pi * x1)
```

```
y2 = np.cos(2 * np.pi * x2) * np.exp(-x1)
```

(3) 创建子图1，绘制曲线

```
plt.subplot(2, 1, 1)
```

#子图1 (2行, 1列, 第1幅图)

```
plt.plot(x1, y1, 'kx-')
```

#绘图样式。黑色，离散数据点形状为“x”，曲线为实线

```
plt.xlabel('time (s)')
```

#x轴标签

```
plt.ylabel('Undamped')
```

#y轴标签

```
plt.title('A tale of 2 subplots') #图标题
```

例11.1-subplot.py





【例11.1】程序（续）

子图2（2行，1列，
第2幅图）

(3) 创建子图2，绘制曲线

plt.subplot(2, 1, 2) #子图2

plt.plot(x2, y2, 'ro-') #红色，离散数据点形状为实心圆点，曲线为实线

plt.xlabel('time (s)')

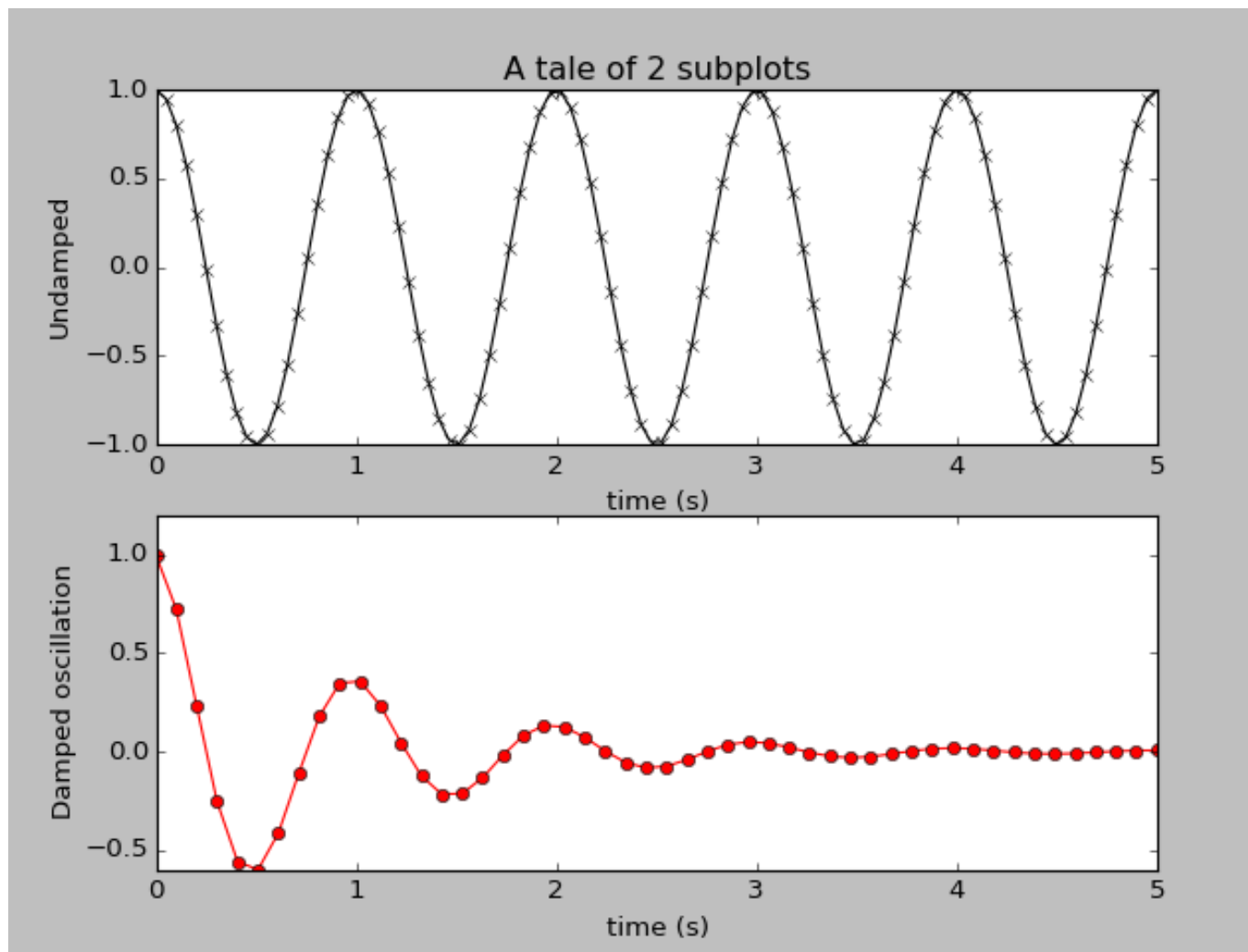
plt.ylabel('Damped oscillation')

plt.show()

#使图形显示

必须有此句！否则看不到图形

【例11.1】运行结果



绘制折线图小结

■ 绘制折线图的步骤

◆ 准备数据 (x,y)

- ✓ 已知实验数据存入**列表**或**元组**
- ✓ 绘制函数时, 采用arange函数或linspace函数生成一系列等间隔的数据 (**数组**)

◆ 根据需要创建子图

- ✓ subplt函数或subplots函数

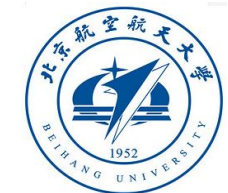
如果只有一个坐标系
则不必创建子图

◆ 使用plot()函数绘制折线

◆ 通过颜色 (**color** 参数) 、离散点形状 (**marker** 参数) 和线型 (**linestyle** 参数) 进行区分

◆ 辅以必要的坐标轴标签、图标题修饰

- ✓ xlabel、ylabel、title方法





11.2.2 绘制柱状图

■ 柱状图的要素

◆ 数据，误差，图例，坐标轴标签.....

■ matplotlib.pyplot模块中的**bar函数**用于绘制矩形柱状图

格式

*bar(left, height, width=0.8, bottom=None, hold=None, **kwargs)*

◆ **【例】** `rects1 = plt.bar(ind, menMeans, width)`





bar函数的用法

■ bar函数的常用参数

(1) 输入参数

left : 柱状图的**左边缘的x坐标**序列 (**列表或元组**)

height : 柱状图的**高度**序列 (列表或元组)

width : 柱状图的**宽度**, 默认值为**0.8**

bottom: bar 的**底部高度**, 默认值为None

****kwargs**: 其他可选参量, 参见Matplotlib的官网中

"matplotlib.patches.Polygon" (完全一致)





bar函数的用法（续）

(2) 返回值

返回：class（类）：`matplotlib.patches.Rectangle`的实例

- ◆ 如果一张图表中绘制有多个数据集的柱状图，当需要给每种柱形添加图例时，需要**使用bar函数的返回值作为legend函数的参数**





bar函数的参数

序号	参 数	含 义
1	left	--设置柱状图的左边界
2	height	--设置柱状图的高度
3	width	--设置柱状图的宽度， 默认为0.8
4	bottom	--设置柱状图底部的y轴坐标，默认值为None
5	color	--指定柱状图的填充显示颜色，默认值为None (蓝色)
6	edgecolor	--指定柱状图的边缘显示颜色，默认值为None
7	linewidth	--指定柱状图的边缘线宽，默认值是None





bar函数的参数（续）

序号	参 数	含 义
8	tick_label	--指定刻度显示，默认值为None
9	xerr	--指定柱状图的x轴的误差显示。默认值为None
10	yerr	--指定柱状图的y轴的误差显示。默认值为None
11	align	--设置柱状图的对齐方式，如果是left，当柱状图的显示为垂直方向时代表左边对齐，当柱状图的显示为水平方向时代表下边对齐
12	orientation	--设置柱状图的朝向，vertical或horizontal，默认值为 vertical
13	log	--指定坐标轴的尺度为对数方式，若为true，为对数轴。默认值为False





legend函数的用法

- matplotlib.pyplot模块的**legend函数**用来设置**图例**，其用法
legend(*args, **kwargs)

【例】：

```
rects1 = plt.bar(ind, menMeans, width, color = 'r', yerr=menStd)           #红色
rects2 = plt.bar(ind+width, womenMeans, width, color='y', yerr=womenStd) #黄色
plt.legend( (rects1, rects2), ('Men', 'Women'))
#给柱状图rects1、 rects2添加图例'Men'、 'Women'
```

◆ 当只有一种柱形时，不必添加图例





legend函数的参数

■ legend函数的参数

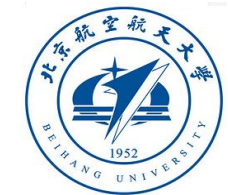
(1) 输入参数

args: 参数序列 (**图形对象**, 为bar函数的返回值)

kwargs: 参数所对应的**图例序列** (元组, **文字说明**, 每个图例用单引号括起来)

loc参数: 指定图例所在**位置**, loc='upper right': 右上角, 默认值, 可省略; loc='lower right', 右下角; loc='upper left': 左上角; loc='lower left', 左下角

(2) 无返回值





【例11.2】绘制柱状图示例

【例11.2】 (1) 已知有2组离散数据保存于列表menMeans、womenMeans中，假设每组包含5个数据点，试间隔一定的距离，绘制其相应的柱状图。

menMeans = (20, 35, 30, 35, 27)

womenMeans = (25, 32, 34, 20, 25)

(2) 假设两组数据存在一定的**统计误差**，试以柱状图的顶部为基准，标示正负误差。

menStd = (2, 3, 4, 1, 2)

womenStd = (3, 5, 2, 3, 3)

(3) **美化图表**，如添加图例，设置图的标题、x轴标签、y轴标签等。



【例11.2】 Python程序

```
import numpy as np
import matplotlib.pyplot as plt
```

(1) 生成绘制柱状图所需的数据

```
N = 5
```

```
menMeans = (20, 35, 30, 35, 27)
```

```
womenMeans = (25, 32, 34, 20, 25)
```

```
menStd = (2, 3, 4, 1, 2)
```

```
womenStd = (3, 5, 2, 3, 3)
```

```
ind = np.arange(N)
```

```
width = 0.35
```

#柱形的左边缘的x坐标

#柱的宽度

例11.2-bar_legend.py



【例11.2】Python程序（续）

(2) 绘制柱状图

```
rects1 = plt.bar(ind, menMeans, width, color = 'r', yerr=menStd)
rects2 = plt.bar(ind+width, womenMeans, width, color='y', yerr=womenStd)
```

误差

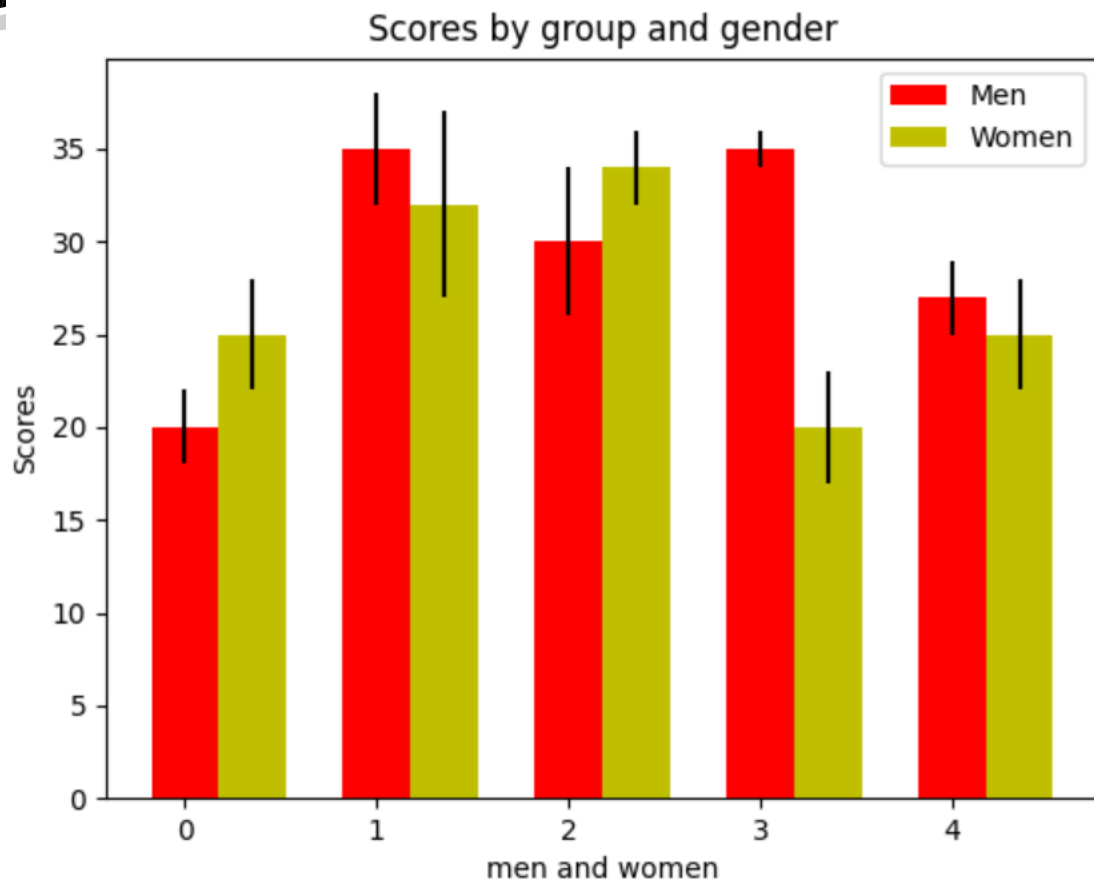
(3) 美化图表

添加图例

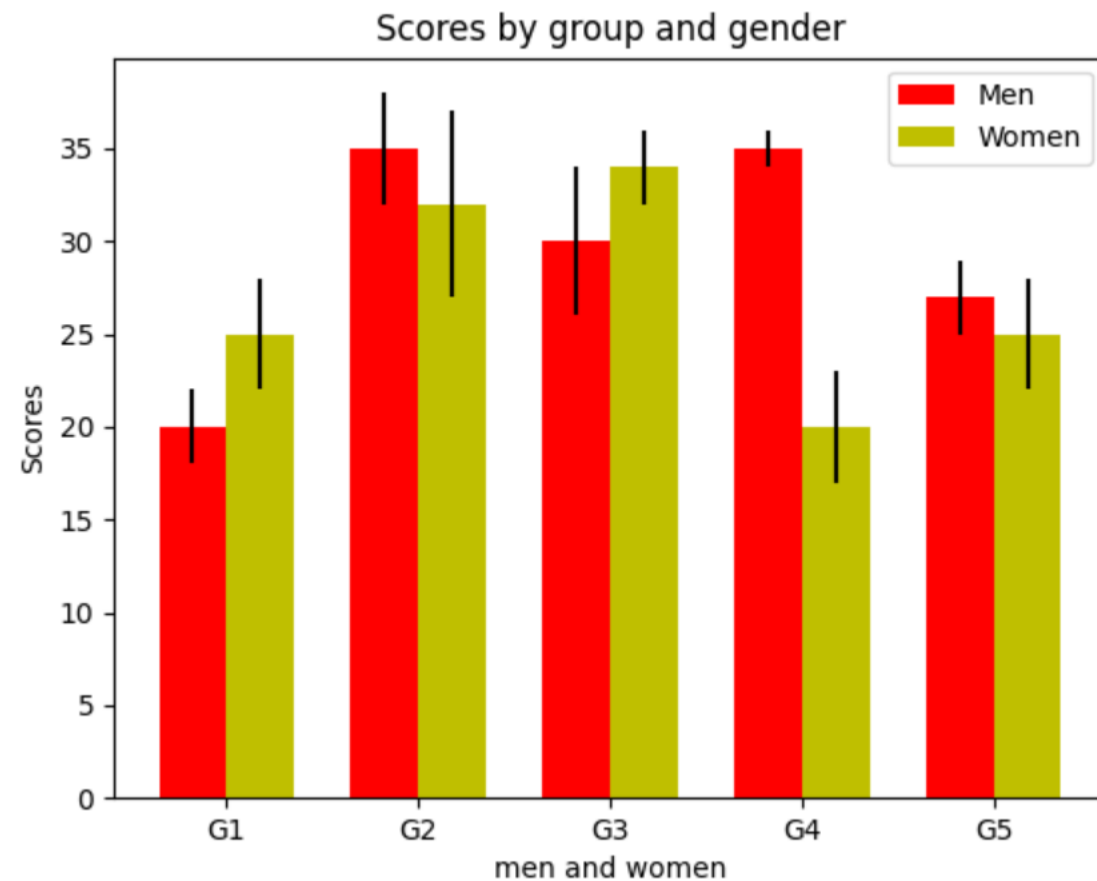
```
plt.legend( (rects1, rects2), ('Men', 'Women')) #默认右上角
plt.xlabel('men and women') #设置横坐标名称
plt.ylabel('Scores') #设置纵坐标名称
plt.title('Scores by group and gender') #设置图的标题
#plt.xticks(ind+ 1/2*width, ('G1', 'G2', 'G3', 'G4', 'G5')) #xticks函数设置当前坐标
#图的x轴刻度标记。第1个参数指定标记位置，第2个参数指定标记的符号
#如果用户不设置轴坐标标记号，系统会自动根据柱形的左边缘的x坐标按合适的间隔标记
```

```
plt.show()
```


【例11.2】运行结果



不设置轴坐标记号



设置当前坐标图的x轴刻度标记

`plt.xticks(ind+ 1/2*width, ('G1', 'G2',
'G3', 'G4', 'G5'))`

绘制柱状图小结

■ 绘制柱状图的步骤

◆ 准备数据 (x,y)

- ✓ 已知实验数据存入**列表**或**元组**
- ✓ y误差, 柱形的**左边缘的x坐标**

◆ 根据需要创建子图

- ✓ subplt函数或subplots函数

如果只有一个坐标系
则不必创建子图

◆ 使用bar()函数绘制柱状图

◆ 通过颜色 (**color** 参数)、底纹进行区分

◆ 辅以必要的坐标轴标签、图标题修饰

- ✓ xlabel、ylabel、title方法

◆ 当一幅图中有多种柱形时, 必须使用legend函数**添加图例**





绘制折线图时如何添加图例？

■ 当一幅图中有多种柱形时，必须使用**legend**函数添加图例

(1) 在调用plt.plot()绘制折线时，用**label**参数指定图例的**文字说明**

绘制正弦和余弦曲线

```
plt.plot(X, S, color='green', linewidth=1.0, linestyle='--', label='sinx')
```

```
plt.plot(X, C, color='blue', linewidth=1.0, linestyle='-', label='cosx')
```

(2) 调用**plt.legend()**函数，添加图例

#添加图例

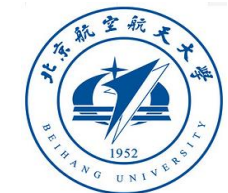
plt.legend()

right')效果相同

#**plt.legend(loc='lower right')**

#**添加图例**，默认是右上角，与plt.legend(loc='upper

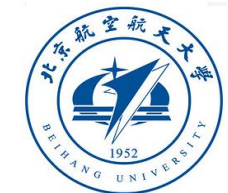
#在右下角添加图例





【例11.3】绘制正弦和余弦函数曲线

【例11.3】 在同一张图上绘制正弦函数和余弦函数在区间 $[-\pi, \pi]$ 内的曲线。分别在两条曲线的旁边添加文字标注 “ $y=\sin x$ ” ,
“ $y=\cos x$ ” 。添加图例。





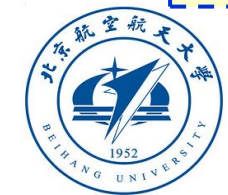
如何解决Matplotlib无法显示中文字体的问题？

■ 问题：如何解决Matplotlib无法显示中文字体的问题？

- ◆ 有时候，图像中图标题、x轴标签或y轴标签中中文显示为乱码
- ◆ 解决方法：在程序开始添加语句：

```
from pylab import mpl
```

```
mpl.rcParams['font.sans-serif'] = ['SimHei'] #指定默认字体为黑体  
mpl.rcParams['axes.unicode_minus'] = False #解决将负号 '-' 显示为方  
块的问题
```





技巧：如何在图表中添加文本标注？

- 当一张图表中包含多条曲线时，最好为每条曲线添加文本标注，使每条曲线意义明显。或者在柱状图中为每个柱形标注高度
- **text函数**用于在绘图区域添加文本，其用法

*text(x, y, s, fontdict=None, withdash=False, **kwargs)*

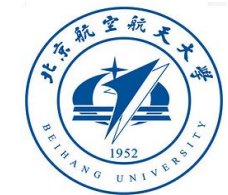
- **text函数**的参数

x, y：文本的x、y坐标值

s：要添加的文本（用**单引号**或**双引号**括起来）

fontdict：字典，可选参数，缺省：None。用于覆盖默认文本参数。若为None，则默认值由你的rc参数决定

withdash：布尔值，可选参数，缺省：False，创建一个带破折号的文本实例





text函数的用法

****kwargs** : 文本属性, 其它各种文本参数, **ha**表示text的**水平对齐**(horizontal align) 方式 (其值为**center**—x坐标位置在文字的中心, **left** — x坐标位置在文字的左边, **right**—x坐标位置在文字的右边); **va**表示**垂直对齐**(vertical align) 方式 (其值为**center** —y坐标位置在文字的中心, **top** — y坐标位置在文字的上面, **bottom** — y坐标位置在文字的下面); **fontsize**表示**字体大小** (缺省值为12)

from pylab import mpl

导入pylab的mpl模块

text(-1.5, 0.5, 'y=cosx', fontsize=10, ha='center', va='bottom')

#在(-1.5, 0.5)坐标位置添加文本'y=cosx', 字体为10号, 文字以x坐标位置为中心, y坐标位置在文字的下面

◆ 例



【例11.3】程序

例11.3-sin_cos.py

(1) 导入三方库

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
from pylab import mpl
```

#导入pylab的mpl模块。必须这么写

(2) 创建图表

```
plt.figure('绘制正弦和余弦函数曲线',figsize=(8,6), dpi=80)
```

#创建一个8*6 的图表，第一个参数为图表名称或序号；并设置分辨率为80

```
#plt.figure(1)
```

#也可以这样写，创建图表1，默认图表名称为 “Figure 1”

(3) 生成绘制曲线所需的数据

```
X = np.linspace(-np.pi, np.pi, 256,endpoint=True) #产生区间 $[-\pi, +\pi]$ 内等间隔的256个值
```

```
S,C = np.sin(X), np.cos(X)
```

#计算函数值

```
XX = np.arange(-4.0,4.0,0.1)
```

#产生区间 $[-4, +4]$ 内步长为0.1的X值，即0轴的X值

```
YY=XX*0
```

#产生0轴相应的Y值

【例11.3】Python程序（续1）

用于添加图例

(4) 绘制正弦和余弦曲线

`plt.plot(X, S, color='green', linewidth=1.0, linestyle='--', label='sinx')` #正弦曲线, 绿色, 宽度为1 (像素) 的虚划线, label指定了图例的文字说明

`plt.plot(X, C, color='blue', linewidth=1.0, linestyle='-', label='cosx')` #余弦曲线, 蓝色、宽度为1 (像素) 的实线

`plt.plot(XX, YY, color='red', linewidth=1.0, linestyle='-')` #绘制0轴

(5) 美化图表

`mpl.rcParams['font.sans-serif'] = ['SimHei']` #指定默认字体 (解决中文显示为乱码)

`mpl.rcParams['axes.unicode_minus'] = False` #解决将负号'-'显示为方块的问题

`plt.title('sinx和cosx曲线')` #设置图标题 (中文)

`plt.xlabel('x axis')` #设置x轴标签

`plt.ylabel('y axis')` #设置y轴标签



【例11.3】Python程序（续2）

#在曲线旁**添加文本标注**

```
plt.text(-1.5, 0.5, 'y=cosx', fontsize=10, withdash=True, ha='center', va='bottom')
```

```
plt.text(3, 0.5, 'y=sinx', fontsize=10, ha='center', va='bottom')#单引号或双引号都可以
```

```
plt.xlim(-4.0,4.0)
```

#设置横轴的上下限

```
plt.xticks(np.linspace(-4,4,9,endpoint=True))
```

#设置**横轴刻度标记**

```
plt.ylim(-1.0,1.0)
```

#设置纵轴的上下限

```
plt.yticks(np.linspace(-1,1,5,endpoint=True))
```

#设置**纵轴刻度标记**

```
plt.legend() #添加图例，默认是右上角，与plt.legend(loc='upper right')效果相同
```

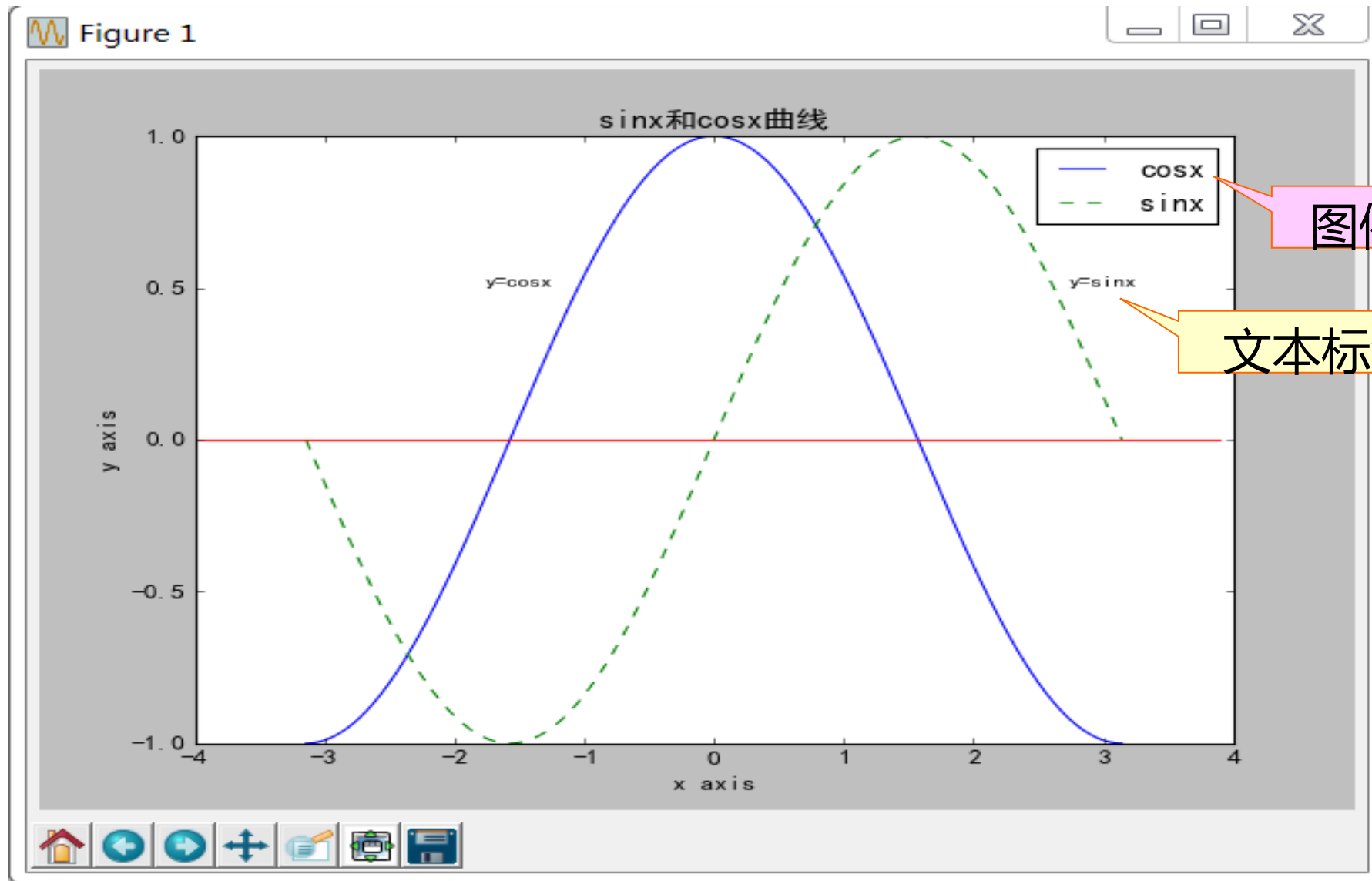
```
#plt.legend(loc='lower right')
```

#添加图例，右下角

```
plt.show()
```

#在屏幕上显示

【例11.3】程序运行结果



图例

文本标注