

数据可视化

DATA

创新应用技术类 — 数据分析技术

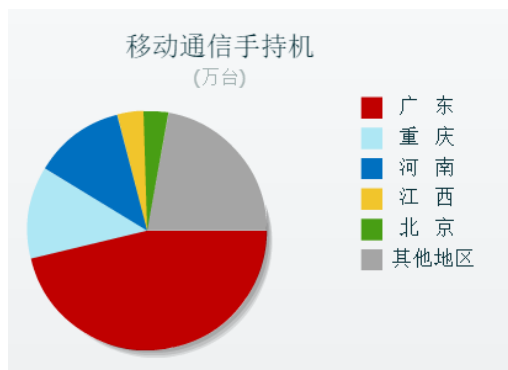
主讲教师：宋晖

数据可视化

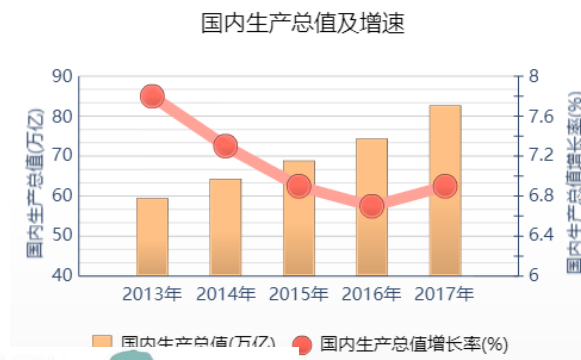
- 数据探索阶段的重要方法
 - 数据以图形图像形式表示
 - 揭示隐藏的数据特征，直观传达关键信息
- Matplotlib库
 - 专门用于开发二维（包括三维）图表的工具包
 - 实现图像元素精细化控制，绘制专业的分析图表
- Pandas封装了Matplotlib的主要绘图功能
 - Series和DataFrame提供绘图函数
 - 简便快捷地创建标准化图表

认识基本图形

- 按照数据值特性，可视图形大致可以分为3类
 - 展示离散数据：散点图、柱状图、饼图等；



- 展示连续数据：直方图、箱须图、折线图、半对数图等；



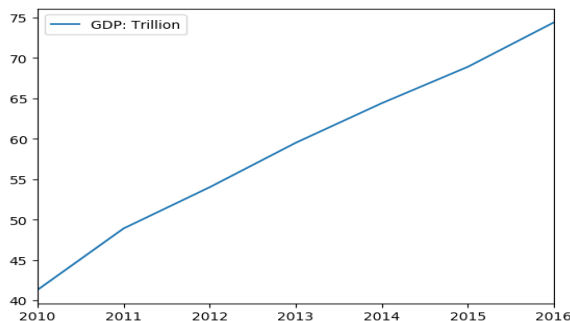
- 展示数据的区域或空间分布：统计地图、曲面图等



4.1.2 Pandas快速绘图

- 基本步骤
 - 导入matplotlib、Pandas
 - 准备数据
 - 使用Series或DataFrame封装数据
 - 绘图
 - 调用Series.plot()或DataFrame.plot()函数完成绘图

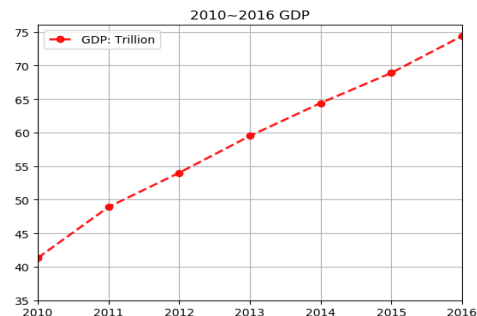
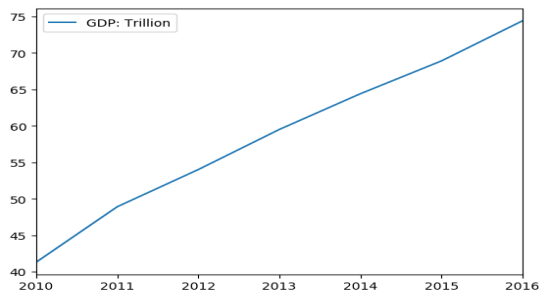
例4-1: 绘制2010-2016年我国GDP折线图



```
import matplotlib.pyplot as plt          #导入pyplot, 用于图形显示
from pandas import DataFrame
gdp = [41.3,48.9,54.0,59.5,64.4,68.9,74.4]
data = DataFrame({'GDP: Trillion':gdp},
                  index=['2010','2011','2012','2013','2014','2015','2016'])
data.plot()
plt.show()    #显示图形
```

4.1.2 Pandas快速绘图

例4-1（续）：绘制2010-2016年我国GDP折线图



参 数 名	说 明
x	x轴数据，默认值为None
y	y轴数据，默认值为None
kind	绘图类型。'line': 折线图，默认值；'bar': 垂直柱状图；'barh': 水平柱状图；'hist': 直方图；'box': 箱形图；'kde': Kernel核密度估计图；'density' 与kde相同；'pie': 饼图；'scatter': 散点图
title	图形标题，字符串
color	画笔颜色。用颜色缩写，如'r'、'b'，或者RGB值，如'#CECECE'。主要颜色缩写：'b': blue、'c': cyan、'g': green、'k': black、'm': magenta、'r': red、'w': white、'y': yellow
grid	图形是否有网格，默认值为None
fontsize	坐标轴（包括x轴和y轴）刻度的字体大小。整数，默认值为None
alpha	图表的透明度，值为0~1，值越大颜色越深
use_index	默认为True，用索引作为x轴刻度
linewidth	绘图线宽
linestyle	绘图线型。'-': 实线；'--': 破折线；'-.': 点画线；':': 虚线
marker	标记风格。'.': 点；':': 像素（极小点）；'o': 实心圈；'v': 倒三角；'^': 上三角；'>': 右三角；'<': 左三角；'1': 下花三角；'2': 上花三角；'3': 左花三角；'4': 右花三角；'s': 实心方形；'p': 实心五角；'*': 星形；'h'/ 'H': 竖/横六边形；' ': 垂直线；'+': 十字；'x': x；'D': 菱形；'d': 瘦菱形
xlim, ylim	x轴、y轴的范围，二元组表示最小值和最大值
ax	axes对象

4.1.3 Matplotlib精细绘图

- 基本步骤

- 导入matplotlib、Pandas, 导入matplotlib的pyplot模块
- 创建figure对象,matplotlib的图像都位于figure对象内
- 绘图: 利用pyplot的绘图函数plot() 或pandas绘图
- 设置图元: plt的图元设置函数, 实现图形精细控制

例4-1 (续): 绘制2010-2016年我国GDP折线图

```
import matplotlib.pyplot as plt    #导入绘图库
plt.figure()    #创建绘图对象
GDPdata=[[41.3,48.9,54.0,59.5,64.4,68.9,74.4]]    #准备绘图的序列数据
plt.plot(GDPdata,color="red",linewidth=2,linestyle='dashed',marker='o',label='GDP')    #绘图
#精细设置图元
plt.title('2010~2016 GDP: Trillion')
plt.xlim(0,6)    #x轴绘图范围
plt.ylim(35,75)    #y轴绘图范围
plt.xticks(range(0,7),('2010','2011','2012','2013','2014','2015',
    '2016'))    #将x轴刻度映射为字符串
plt.legend(loc='upper right')    #在右上角显示图例说明
plt.grid()    #显示网格线
plt.show()    #显示并关闭绘图
```

图元添加完后,再调用show()

- 显示图像
- 图形绘制过程关闭

多子图绘制

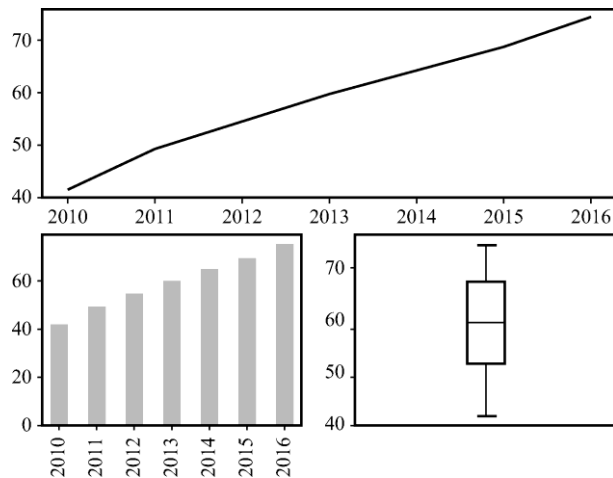
- figure对象内可绘制多个子图
 - 创建子图对象axes, 在子图上绘制图
 - 可使用pyplot或axes对象提供的绘图
 - 可pandas绘图
- 创建子图

figure.add_subplot(numRows, numCols, plotNum)

参数说明:	
numRows	绘图区域被分成numRows行
numCols	绘图区域被分成numCols列
plotNum	创建的axes对象所在的区域

多子图绘制实例

例4-2：用多个子图绘制2010~2016年GDP状况



创建1个子图，绘制
再创建，再绘制
创建过程中，子图总数是可变的

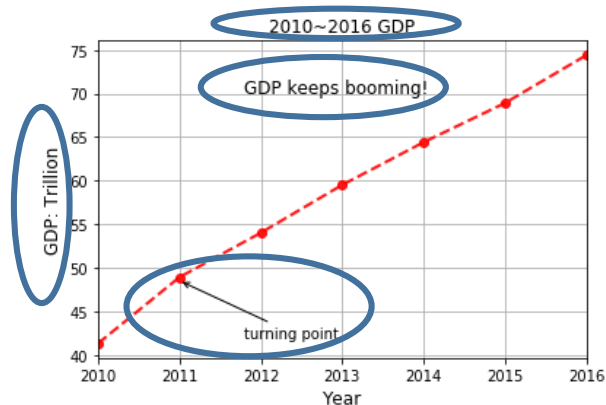
```
from pandas import Series
data=Series([4.13,4.89,5.4,5.95,6.44,6.89,7.44],
index=['2010','2011','2012','2013','2014','2015','2016'])
fig=plt.figure(figsize=(6,6)) #figsize定义图形大小
ax1=fig.add_subplot(2,1,1)    #创建子图1
ax1.plot(data)                #用AxesSubplot绘制折线图
ax2=fig.add_subplot(2,2,3)    #创建子图2
data.plot(kind='bar',use_index=True,fontsize='small',ax=ax2) #用pandas绘柱状图
ax3=fig.add_subplot(2,2,4)    #创建子图3
data.plot(kind='box',fontsize='small',xticks=[],ax=ax3) #用pandas绘柱状图
```


设置图元和说明

函数	说明
plt.title	设置图标题
plt.xlabel、plt.ylabel	设置x、y轴标题
plt.xlim、plt.ylim	设置x、y轴刻度范围
plt.xticks、plt.yticks	设置x、y轴刻度值
plt.legend	添加图例说明
plt.grid	显示网格线
plt.text	添加注解文字
plt.annotate	添加注释

例4-3：为图4-2增加注解、坐标轴标题

```
data.plot(title='2010~2016 GDP',LineWidth=2, marker='o',  
          linestyle='dashed',color='r',grid=True,alpha=0.9)  
plt.annotate('turning point',xy=(1,48.5),xytext=(0.5,42),  
            arrowprops=dict(arrowstyle='->'))  
plt.text(1.8,70,'GDP keeps booming!',fontsize='larger')  
plt.xlabel('Year',fontsize=12)  
plt.ylabel('GDP: Trillion',fontsize=12)
```



保存图表到文件

- 保存函数

figure.savefig(filename,dpi,bbox_inches)

plt.savefig(filename,dpi,bbox_inches)

参数说明:	
filename	文件路径及文件名，文件类型可以是jpg、png、pdf、svg、ps等
dpi	图片分辨率，每英寸点数，默认100
bbox_inches	图表需保存的部分，设置为“tight”可以剪除当前图表周围的空白部分

- 将例4-2绘制图形保存到当前文件夹

```
fig.savefig('2010-2012GDP.jpg',dpi=400,bbox_inches='tight')
```

思考与练习

1. 2012~2017年我国人均可支配收入为[1.47, 1.62, 1.78, 1.94, 2.38, 2.60](单位：万元)。按照要求绘制以下图形。

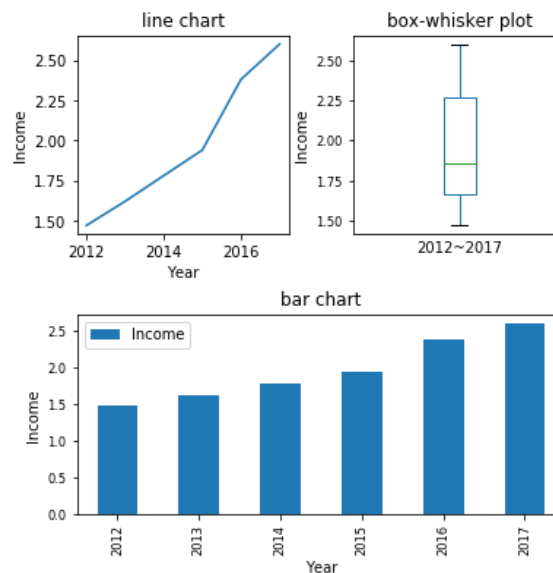
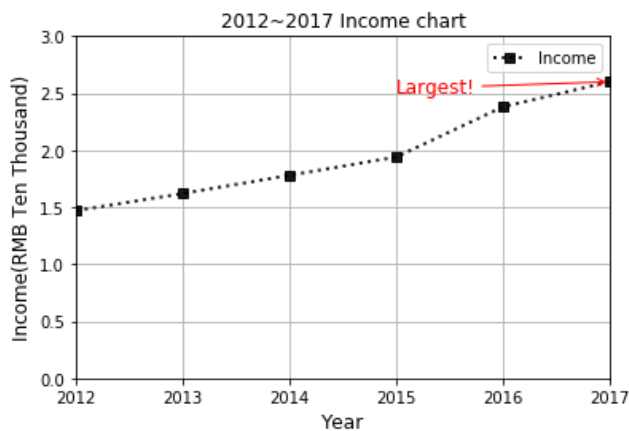
1) 模仿例4-1和4-3，绘制人均可支配收入折线图。用小矩形标记数据点，红色虚线，用注解标注最高点，图标题“Income chart”，设置坐标轴标题，最后将图形保存为JPG文件。一维数组访问。

2) 模仿例4-2，使用多个子图分别绘制人均可支配收入的折线图、箱须图以及柱状图。

【提示：】

1) 创建3个子图分别使用 (2,2,1)、(2,2,2) 和 (2,1,2) 作为参数。

2) 使用plt.subplots_adjust()函数调整子图间距离，以便添加图标题。



4.2.1 绘制常用图形

- 函数绘图
- 散点图
- 柱状图
- 折线图
- 直方图
- 密度图
- 饼图
- 箱须图

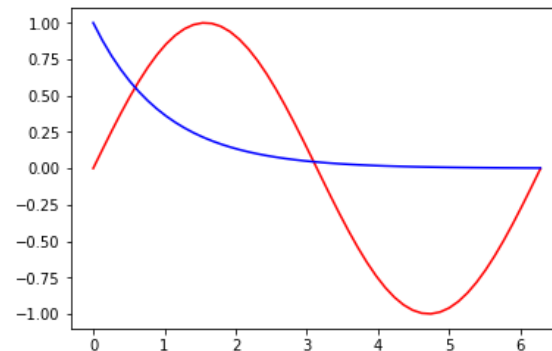
函数绘图

- 函数 描述了变量 y 随自变量 x 的变化过程
- `plt.plot()`根据给定的 x 、 y 坐标值绘图

例4-4：绘制 $y = \sin(x)$ 和 $y = e^{-x}$ 的函数图

- 给定 x 的范围采样生成 x 列表
- 计算对应 y 值

```
import numpy as np                #导入numpy
#生成x数组
x = np.linspace(0,6.28,50)        #start, end, num-points
y=np.sin(x)                       #计算y=sin(x)数组
plt.plot(x,y, color='r')          #用红色绘图y=sin(x)
plt.plot(x,np.exp(-x),c='b')      #用蓝色绘图y=exp(-x)
```



散点图 (Scatter diagram)

- 描述两个一维数据序列之间的关系
 - 将两组数据分别作为点的横坐标和纵坐标

DataFrame.plot(kind='scatter',x,y,title, grid,xlim,ylim,label,...)

DataFrame.plot.scatter(x,y,title, grid,xlim,ylim,label,...)

参数说明:	
x	DataFrame中x轴对应的数据列名
y	DataFrame中y轴对应的数据列名
label	图例标签

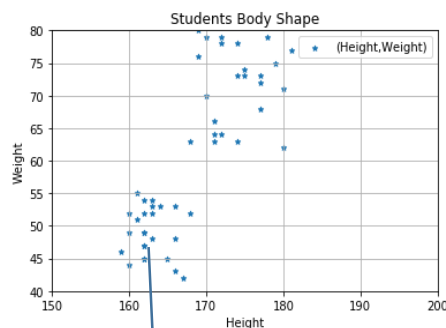
- Matplotlib的scatter函数也可以绘制散点图
 - 图元的设置需要采用独立的语句

plt.scatter(x,y,...)

散点图绘制

例4-5：绘制散点图观察学生身高和体重之间的关系

```
stdata = pd.read_csv('data\students.csv')    #读文件
stdata.plot(kind='scatter',x='Height',y='Weight',title='Students
Body Shape', marker='*',grid=True, xlim=[150,200],
ylim=[40,80], label='(Height,Weight)')    #绘图
```



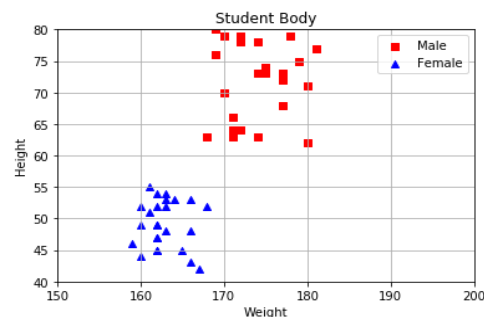
男女生身高、体重明显存在差异性

- 分组散点图清晰显示数据聚集特性

```
#将数据按男生和女生分组
data1= data[data['Gender'] == 0] #筛选出男生
data2= data[data['Gender'] == 1] #筛选出女生
#分组绘制男生、女生的散点图
plt.figure()
plt.scatter(data1['Height'],data1['Weight'],c='r',marker='s',label='Male')
plt.scatter(data2['Height'],data2['Weight'],c='b',marker='^',label='Female')
plt.xlim(150,200) #x轴范围
plt.ylim(40,80) #y轴范围
plt.title('Students Body Shape') #标题
plt.xlabel('Weight') #x轴标题
plt.ylabel('Height') #y轴标题
plt.grid() #网格线
plt.legend(loc='upper right') #图例显示位置
```

使用不同的图例标识分组

学生的身高与体重具有正相关性，但不显著



散点图矩阵

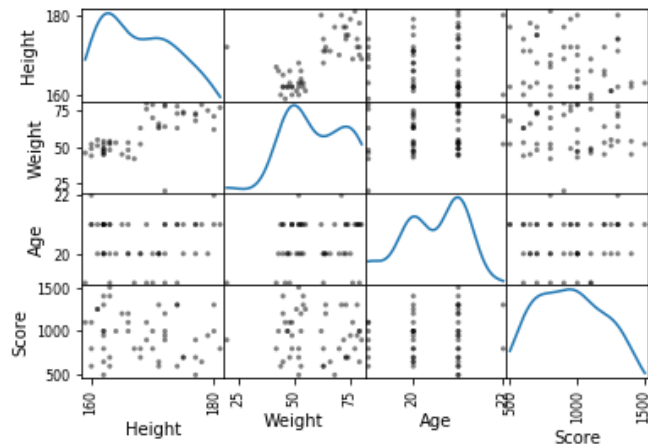
- 同时观察多组数据之间的关系

pd.plotting.scatter_matrix(data,diagonal,...)

参数说明:	
data	包含多列数据的DataFrame对象
diagonal	对角线上的图形类型。通常放置该列数据的密度图或直方图

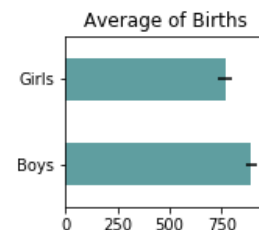
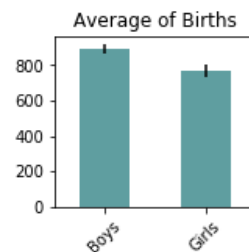
例4-6：绘制散点图矩阵观察学生各项信息之间的关系
身高、体重、年龄、成绩

```
data = stdata[['Height', 'Weight', 'Age', 'Score']] #准备数据  
pd.plotting.scatter_matrix(data,diagonal='kde',color='k') #绘图
```



柱状图 (Bar Chart)

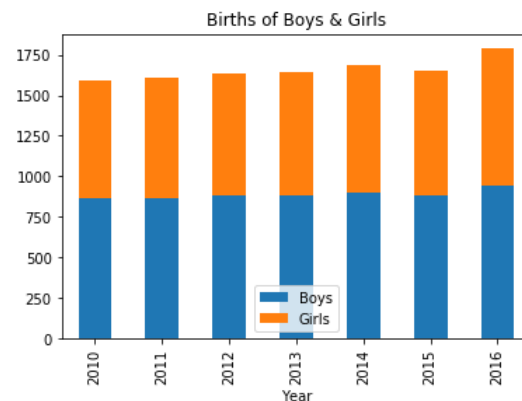
- 用多个柱体描述单个总体处于不同状态的数量
 - 柱体高度或长度与该状态下的数量成正比
 - 分为垂直柱状形图和水平柱状图
- 堆叠柱状图
 - 多个总体同一状态的直条叠加



Series.plot(kind,xerr,yerr,stacked,...)

DataFrame.plot(kind,xerr,yerr,stacked,...)

参数说明:	
kind	bar: 垂直柱状图; barh: 水平柱状
xerr,yerr	x、y轴向误差线
stacked	是否为堆叠图, 默认为False
rot	刻度标签旋转度数, 值0~360

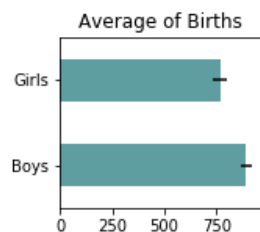
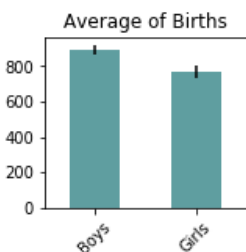


柱状图绘制

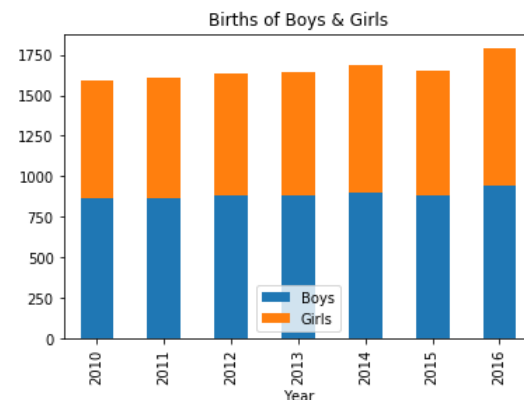
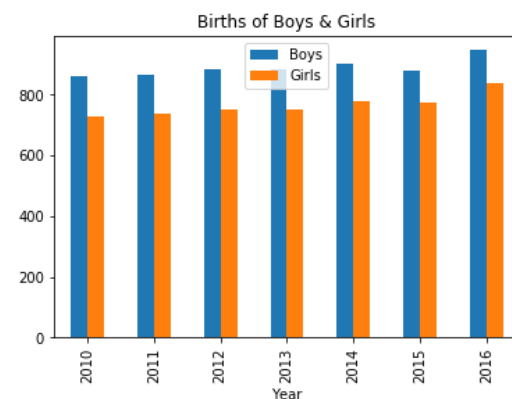
Year	Total	Boys	Girls	Ratio
年度	出生人口总数	男孩数	女孩数	男女比例

- 从population.csv文件中读取人口数据，绘制各性别的出生人口比较图

```
#读取数据
data = pd.read_csv('data\population.csv', index_col = 'Year')
data1 = data[['Boys', 'Girls']]
mean = np.mean(data1, axis=0)          #计算均值
std = np.std(data1, axis=0)            #计算标准差
#创建图
fig = plt.figure(figsize = (6,2)) #设置图片大小
plt.subplots_adjust(wspace = 0.6) #设置两个图之间的纵向间隔
#绘制均值的垂直和水平柱状图，标准差使用误差线来表示
ax1 = fig.add_subplot(1, 2, 1)
mean.plot(kind='bar', yerr=std, color='cadetblue', title = 'Average
      of Births', rot=45)
ax2 = fig.add_subplot(1, 2, 2)
mean.plot(kind='barh', xerr=std, color='cadetblue', title = 'Average
      of Births')
#绘制复式柱状图和堆叠柱状图
data1.plot(kind='bar', title = 'Births of Boys & Girls')
data1.plot(kind='bar', stacked=True, title = 'Births of Boys &
      Girls')
```



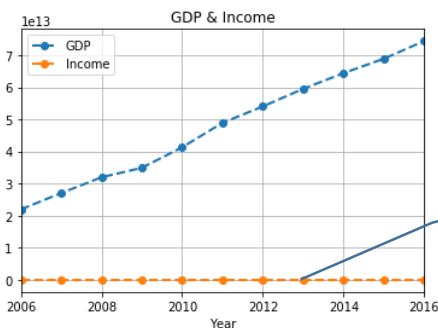
堆叠柱状图



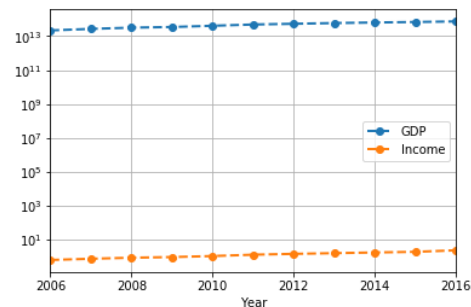
折线图

- 用线条描述事物的发展变化及趋势
 - 普通折线图：横、纵坐标轴上都使用算术刻度
 - 半对数折线图：横、纵坐标分别使用算术刻度与对数刻度
 - 比较的两种或多种事物的数据值域相差较大
 - 指标“相对增长量”的变化关系
- 从GDP.csv文件中读取数据，绘制国民经济生产总值GDP和居民人均可支配收入Income的折线图与半对数折线图

```
data = pd.read_csv('GDP.csv', index_col = 'Year')      #读取数据
#绘制GDP和Income的折线图
data.plot(title='GDP & Income', LineWidth=2, marker='o', linestyle='dashed',
          grid=True, use_index=True)
#绘制GDP和Income的半对数折线图
data.plot(logy=True, LineWidth=2, marker='o', linestyle='dashed', color='G')
```



无法反应Income
的变化



直方图 (Histogram)

- 描述总体的频数分布情况
 - 将横坐标按区间个数等分
 - 每个区间上长方形的高度表示该区间样本的频率, 面积表示频数

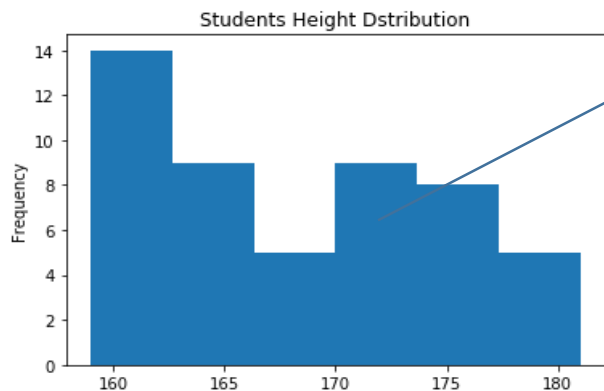
Series.plot(kind='hist',bins,normed,...)

参数说明:	
bins	横坐标区间个数
normed	是否标准化直方图, 默认值False

直方图绘制

例4-9：从student.csv文件中读取学生信息，绘制身高分布直方图。
将身高155~185划分为6个区间

```
stdata = pd.read_csv('data\students.csv')    #读文件  
stdata['Height'].plot(kind='hist',bins=6,title='Students Height Dstribution') #绘图
```



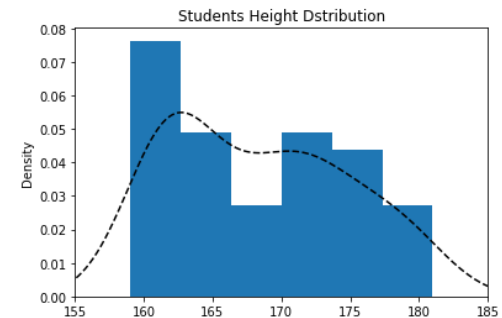
分箱的数量与数据集大小和分布本身相关，通过改变分箱bins的数量，可以改变分布的离散化程度

密度图 (Kernel Density Estimate)

- 基于样本数据拟合概率密度函数
 - 采用平滑的峰值函数：核函数
 - 常用高斯核
 - 模拟真实的概率分布曲线
 - 与直方图（标准化后）一起绘制，对比

Series.plot(kind='kde',style,...)

参数说明:	
style	风格字符串，包括颜色和线型，如'ko—','r-'



在例4-9基础上，增加密度图

```
stdata['Height'].plot(kind='hist',bins=6,normed=True,title='Students Height Distribution') #绘图
stdata['Height'].plot(kind='kde',title='Students Height Distribution', xlim=[155,185],
style = 'k--') #绘制密度图
```

饼图 (Pie Chart)

- 描述总体的样本值构成比
 - 扇形图
 - 反映部分与部分、部分与整体之间的数量关系

Series.plot(kind='pie', explode,shadow,startangle,autopct,...)

参数说明:	
explode	列表, 表示各扇形块离开中心的距离
shadow	扇形块是否有阴影, 默认False
startangle	起始绘制角度, 默认从x轴正方向逆时针开始
autopct	百分比格式, 可用format字符串或者format function, '%1.1f%%'指小数点前后各1位(不足空格补齐)

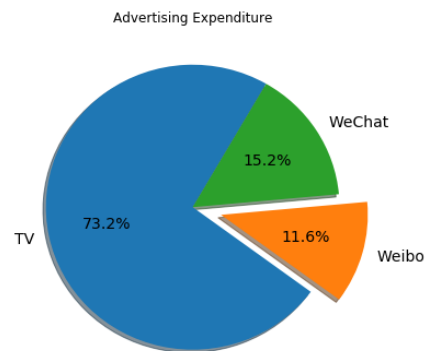
饼图绘制

例4-10：从advertising.csv中读取营销数据，绘制各类广告投入占比的饼图

	TV	Weibo	WeChat	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3

计算各类渠道的广告总投入，绘制饼图表示各类广告占比

```
#准备数据，计算各类广告投入费用总和
data = pd.read_csv('data/advertising.csv')
piedata = data[['TV','Weibo','WeChat']]
datasum = piedata.sum()
#绘制饼图
datasum.plot( kind='pie', figsize=(6,6), title='Advertising
    Expenditure',fontsize=14,
    explode=[0,0.2,0],shadow=True,startangle=60,
    autopct='%1.1f%%')
```



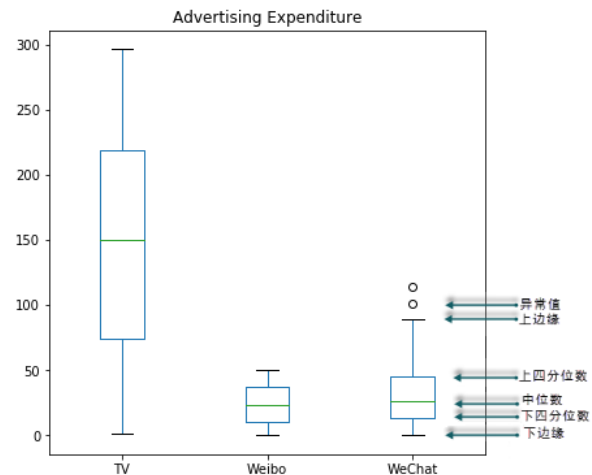
箱须图 (Box plot)

- 表达数据的分位数分布，观察异常值
 - 将样本居中的50%值域用一个长方形表示
 - 较小和较大的四分之一值域各用一根线表示
 - 异常值用“o”表示

Series.plot(kind='box', ...)

例4-10：从advertising.csv中读取营销数据，绘制各类广告投入投入的箱须图

```
data = pd.read_csv('data\Advertising.csv')
advdata = data[['TV', 'Weibo', 'WeChat']]
#绘制各类经费投入的箱须图
advdata.plot(kind='box', figsize=(6,6),
             title='Advertising Expenditure')
```



箱须图 (Box plot)

- Pandas提供专门绘制箱须图的函数boxplot
 - 方便将观察样本按照其他特征进行分组对比

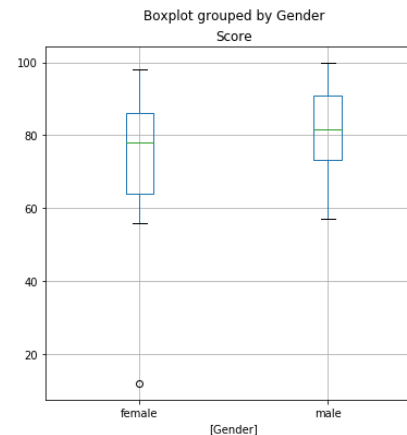
DataFrame.boxplot(by, ...)

参数说明:	
by	用于分组的列名

例4-10： 从students.csv中读取学生数据，按性别绘制学生成绩的箱须图

```
stdata = pd.read_csv('data\students.csv')
stdata1 = stdata[['Gender', 'Score']]
stdata1.boxplot(by='Gender', figsize=(6, 6))
```

Dataframe对象要包括绘制列和分组列

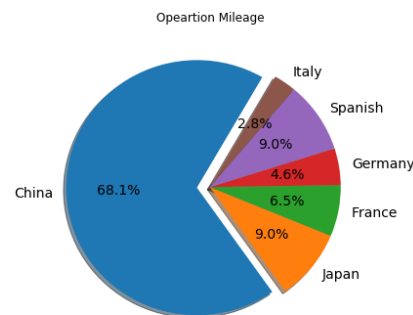
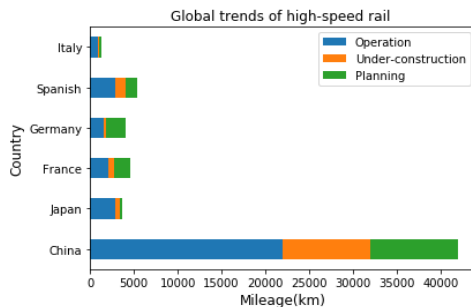
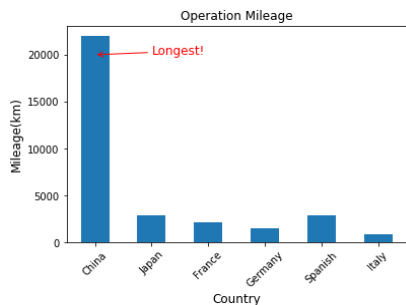


思考与练习

1. 数据文件high-speed rail.csv存放着世界各国高速铁路的情况

Country	Operation	Under-construction	Planning
国家	运营里程（公里）	在建里程（公里）	计划里程（公里）

- 1) 各国运营里程对比柱状图，标注China为“Longest”
- 2) 各国运营里程现状和发展堆叠柱状图
- 3) 各国运营里程占比饼图，China扇形离开中心点



【提示】：

从文件中读取数据时，使用第一列数据作为index

`data = pd.read_csv('High-speed rail.csv', index_col='Country')`，获取中国对应的数据行，使用`data ['China']`

课后作业

文件bankpep.csv存放着银行储户的基本信息

id	age	sex	region	income	married	children	car	save_act	current_act	mortgage	pep
编号	年龄	性别	区域	收入	婚否	孩子数	有车否	存款账户	现金账户	是否抵押	接受新业务

请通过绘图对这些客户数据进行探索性分析。

- 1) 客户年龄分布的直方图和密度图
- 2) 客户年龄和收入关系的散点图
- 3) 绘制散点图观察账户（年龄，收入，孩子数）之间的关系，对角线显示直方图
- 4) 按区域展示平均收入的柱状图，并显示标准差
- 5) 多子图绘制：账户中性别占比饼图，有车的性别占比饼图，按孩子数的账户占比饼图
- 6) 各性别收入的箱须图

