法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

关注 **小象学院**

第三讲

数据展示及可视化

--Robin

目录

- 数据可视化的重要性
- 基本图表的绘制及应用场景
- 数据分析常用图表的绘制
- Pandas及Seaborn绘图
- 其他常用的可视化工具
- 实战案例：YouTube视频趋势分析

目录

数据可视化的重要性

Anscombe's quartet (安斯库姆四重奏)

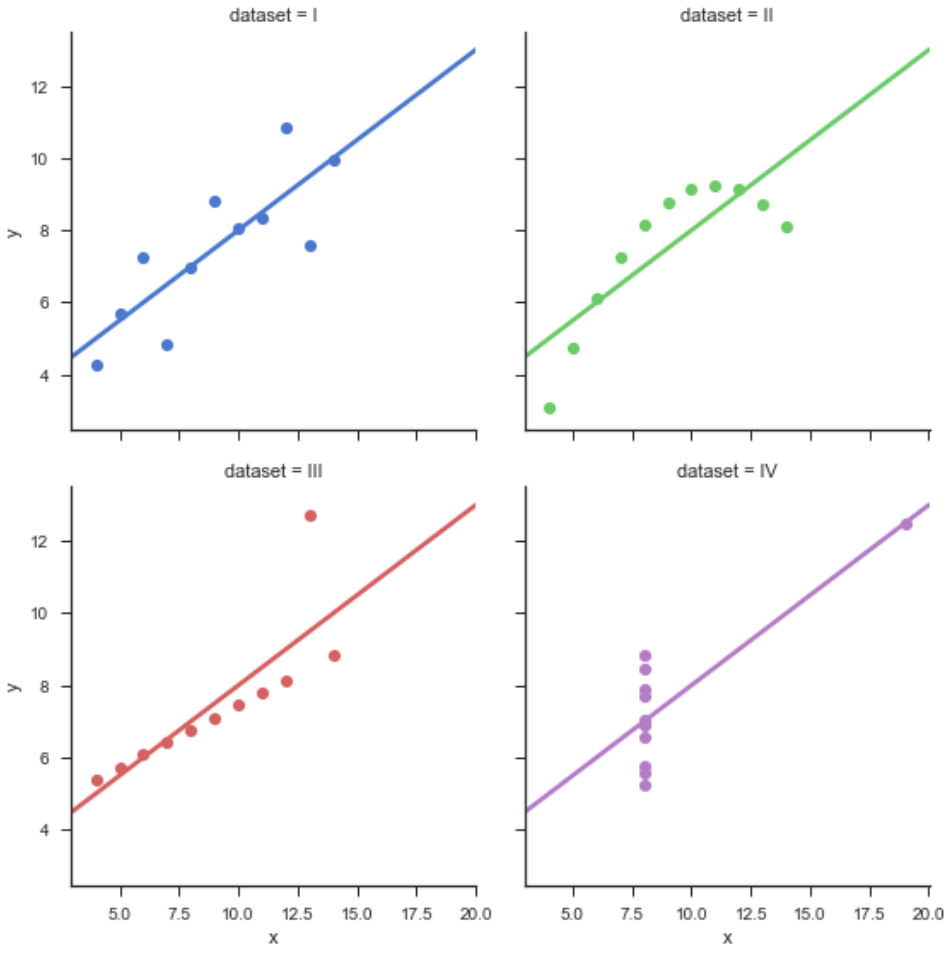
- 是四组基本的统计特性一致的数据，但由它们绘制出的图表则截然不同。
- 每一组数据都包括了11个(x,y)点
- 这四组数据由统计学家弗朗西斯·安斯库姆（Francis Anscombe）于1973年构造，他的目的是用来说明在分析数据前先绘制图表的重要性，以及离群值对统计的影响之大。
- 这四组数据的共同统计特性如下：

性质	数值
x的平均数	9
x的方差	11
y的平均数	7.50（精确到小数点后两位）
y的方差	4.122或4.127（精确到小数点后三位）
x与y之间的相关系数	0.816（精确到小数点后三位）
线性回归线	$y = 3.00 + 0.500x$（分别精确到小数点后两位和三位）

	x		y	
	mean	var	mean	var
dataset				
I	9.0	11.0	7.500909	4.127269
II	9.0	11.0	7.500909	4.127629
III	9.0	11.0	7.500000	4.122620
IV	9.0	11.0	7.500909	4.123249

数据可视化的重要性

Anscombe's quartet (安斯库姆四重奏)

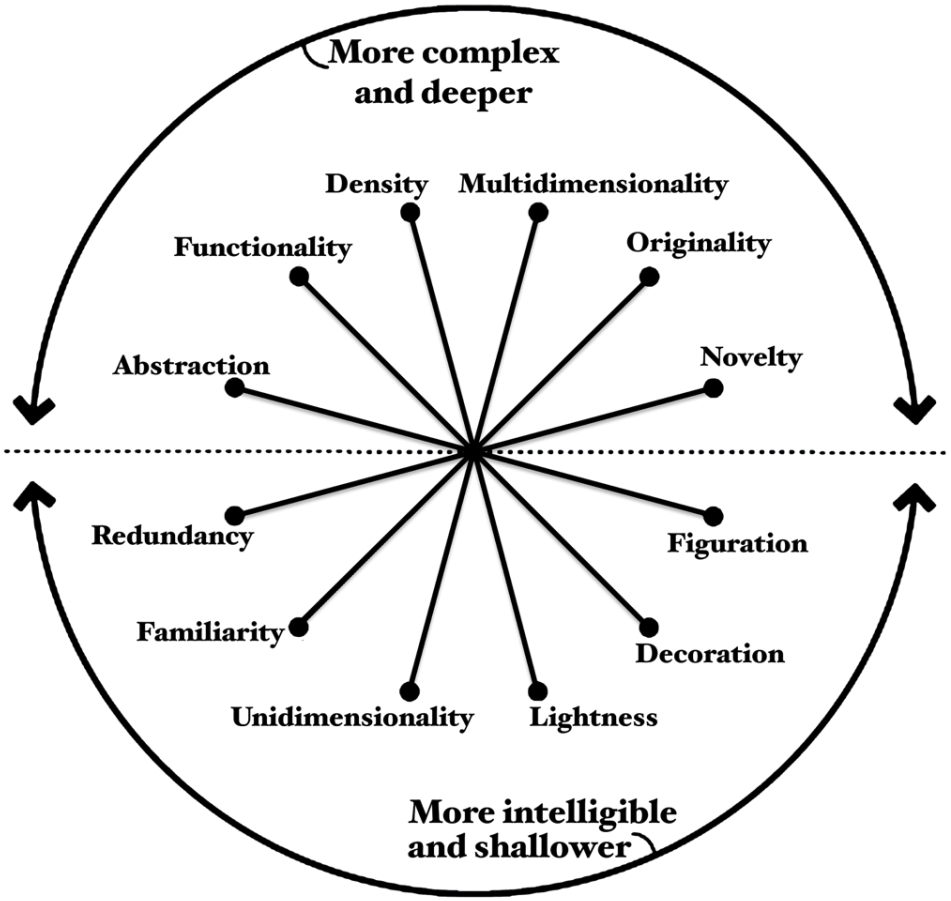


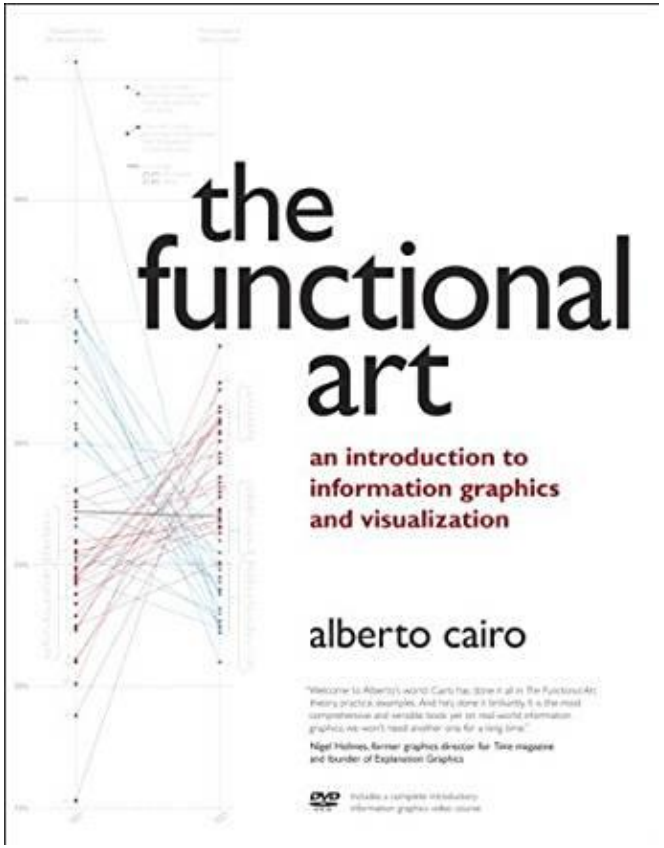
- **左上图**看起来最“正常”，可以看出两个随机变量之间的相关性
- **右上图**可以明显地看出两个随机变量间的关系是非线性的
- **左下图**虽然存在着线性关系，但由于一个离群值的存在，改变了线性回归线
- **右下图**尽管两个随机变量间没有线性关系，但仅仅由于一个离群值的存在就使得相关系数变得很高

lect03_eg01.ipynb

数据可视化的重要性

Visualization Wheel





数据可视化的重要性

Abstraction vs Figuration

- 抽象化，如盒形图
- 具体化，如现实中的物体

Functionality vs Decoration

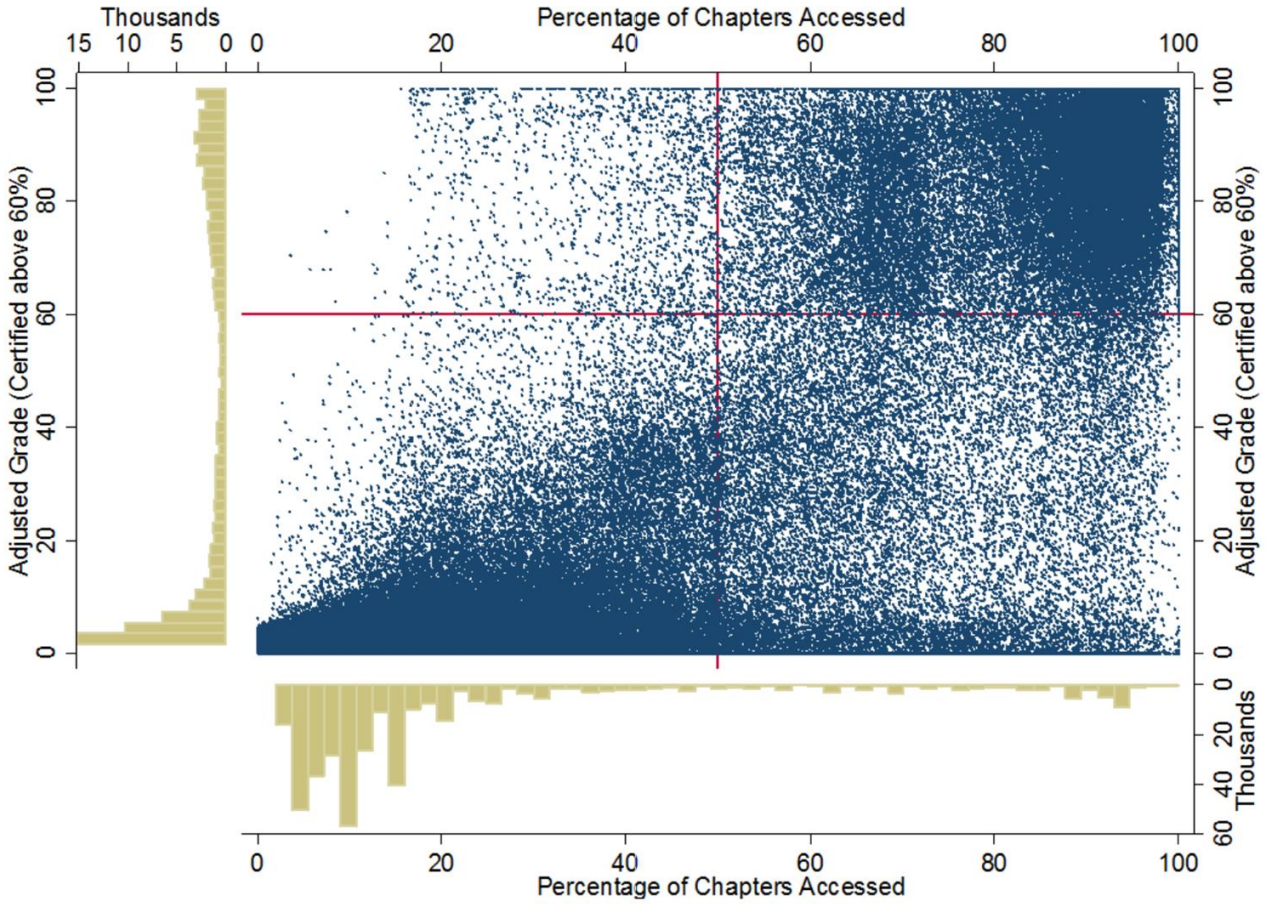
- 功能性：没有装饰及渲染
- 装饰性：包含艺术性、美学上的装饰

Density vs Lightness

- 深层表达：需要深层地研究探索数据
- 浅层表达：易于理解的、直观的表示

数据可视化的重要性

例子：带直方图的散点图



数据可视化的重要性

Multidimensional vs Unidimensional

- 多维度：数据或现象的多个层面
- 单一维度：数据的单一层面或统计

Originality vs Familiarity

- 创造性：全新的方式进行可视化及表达
- 熟悉性：被大众认知所熟悉的表达方式

Novelty vs Redundancy

- 新颖性：每个元素/现象只表述/解释一次
- 冗余性：每个元素/现象只表述/解释多次

数据可视化的重要性

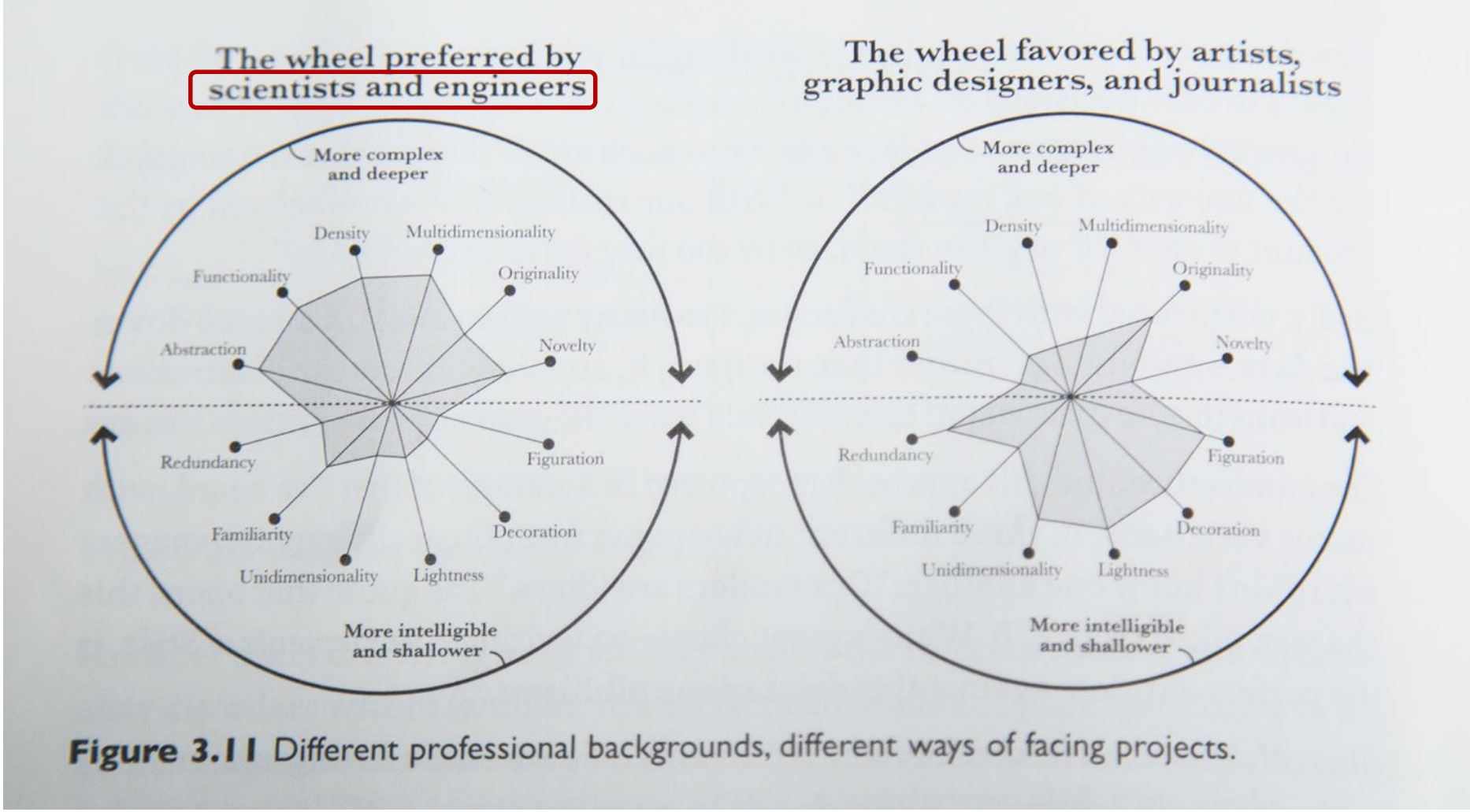
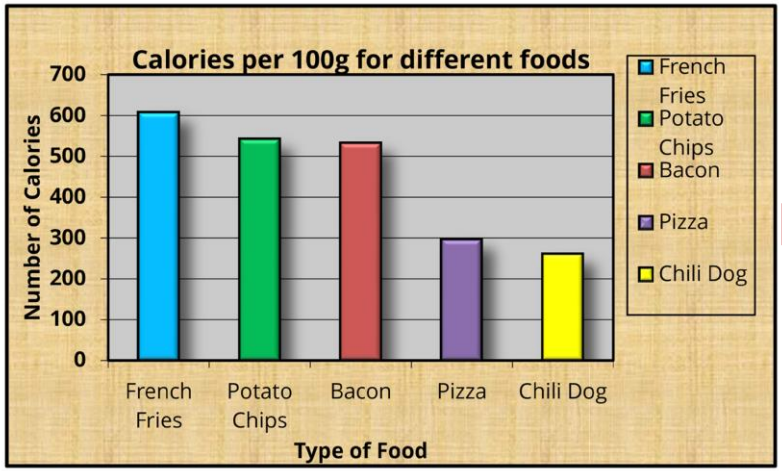


Figure 3.11 Different professional backgrounds, different ways of facing projects.

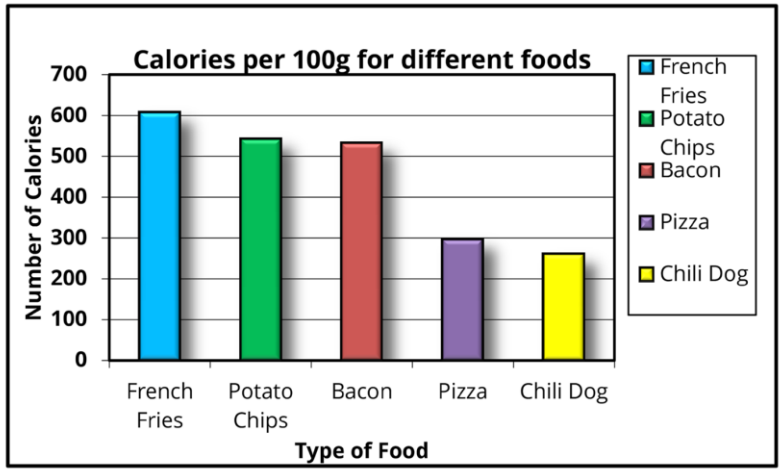
数据可视化的重要性

Remove to Improve

Remove backgrounds



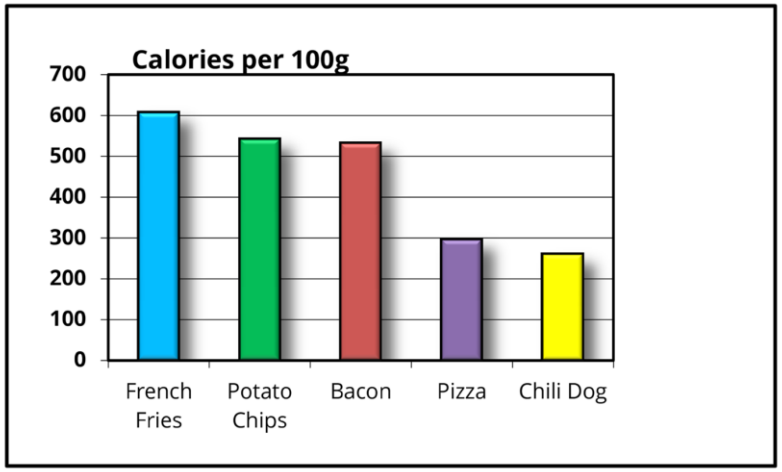
Remove redundant labels



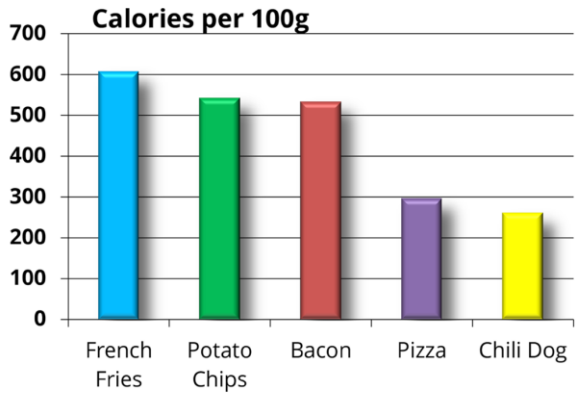
数据可视化的重要性

Remove to Improve

Remove borders



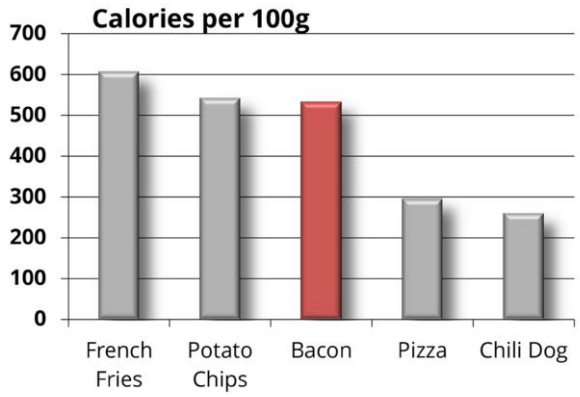
Reduce colors



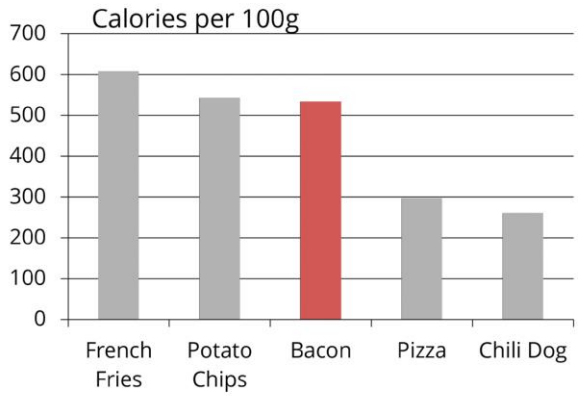
数据可视化的重要性

Remove to Improve

Remove special effects

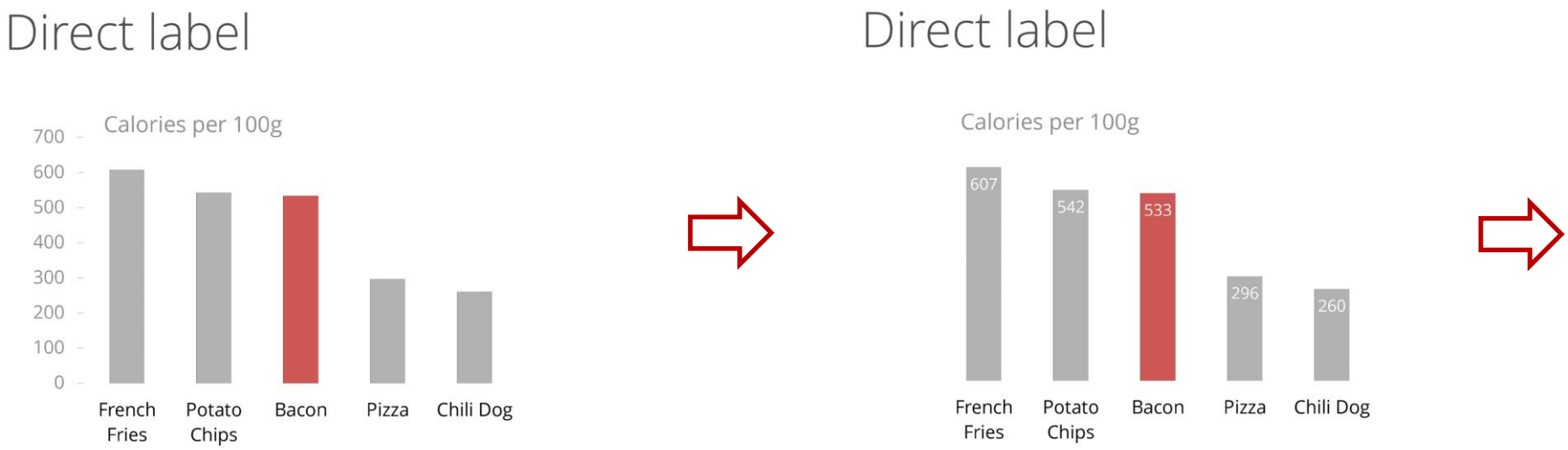


Lighten labels



数据可视化的重要性

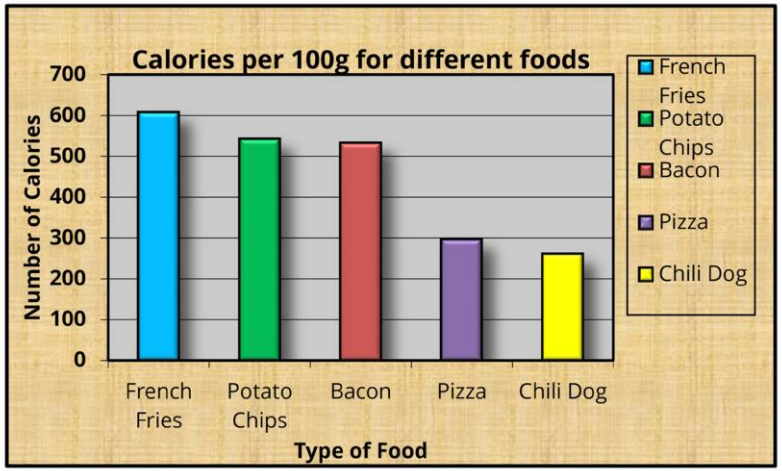
Remove to Improve



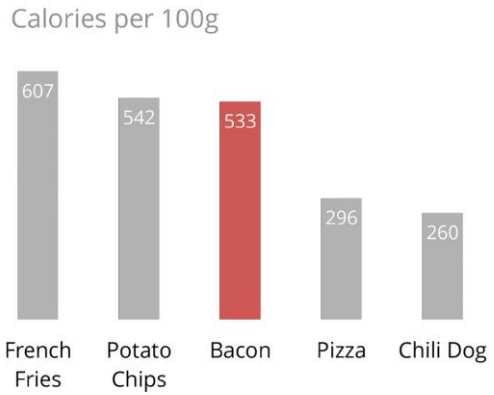
数据可视化的重要性

Remove to Improve

Remove backgrounds



Direct label



数据可视化的重要性

数据可视化准则

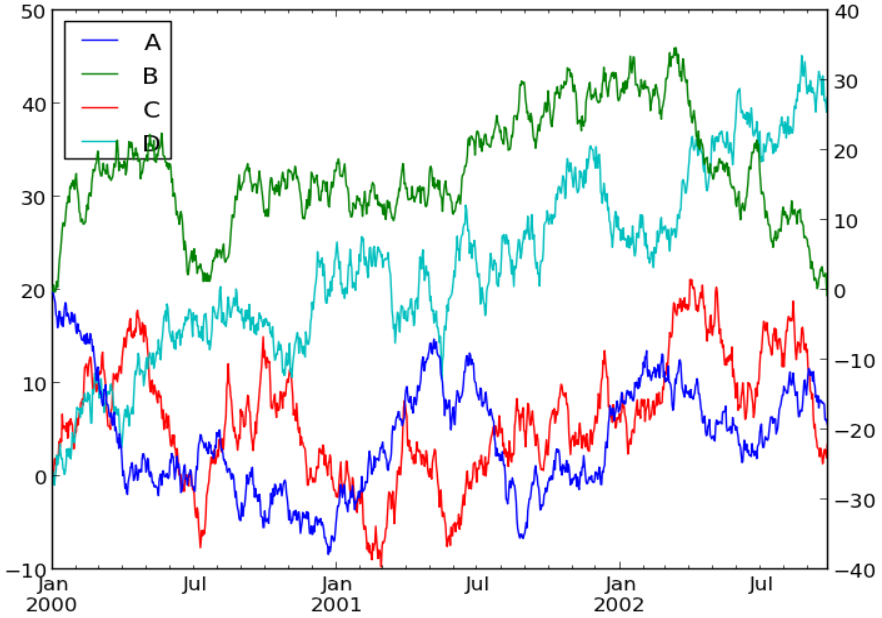
1. 真实性（Truthful）
2. 功能性（Functionality）
3. 美观（Beauty）
4. 深刻性（Insightful）
5. 启发性（Enlightening）

目录

基本图表的绘制及应用场景

Matplotlib

- 用于创建出版质量图表的绘图工具库
- 目的是为Python构建一个Matlab式的绘图接口
- import matplotlib.pyplot as plt
  - pyploy模块包含了常用的matplotlib API函数



lect03_eg02.ipynb

基本图表的绘制及应用场景

Matplotlib架构

- Backend层
  - 用于处理向屏幕或文件渲染图形
  - 在Jupyter中，使用inline backend
  - What is a backend?
- Artist层
  - 包含图像绘制的容器，如Figure，Subplot及Axes
  - 包含基本元素，如Line2D，Rectangle等
- Scripting层
  - 简化访问Artist和Backend层的过程

lect03_eg02.ipynb

基本图表的绘制及应用场景

pyplot

- https://matplotlib.org/users/pyplot_tutorial.html
- pyplot可通过gcf(get current figure)获取当前图像对象，gca(get current axis)获取当前坐标轴对象
- pyplot只是对axes对象的调用做了“镜像”，可以通过pyplot.plot()进行绘图，其底层调用的还是axes.plot()函数
- matplotlib的许多绘制函数包含许多开放的参数供使用，注意参考相关API文档

```
lect03_eg02.ipynb
```

基本图表的绘制及应用场景

散点图

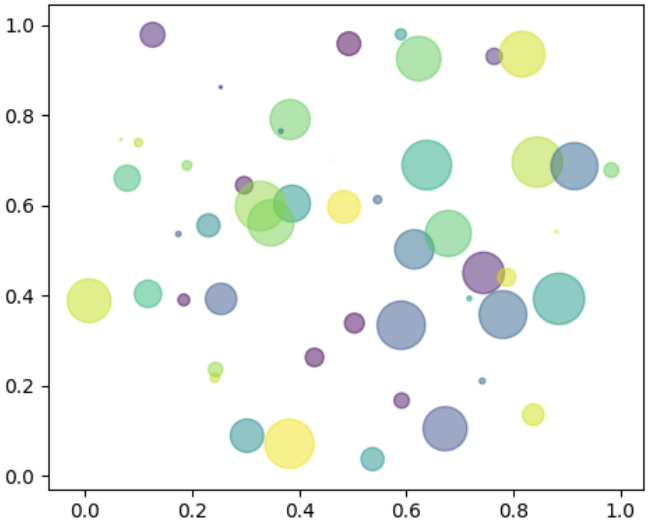
- plt.scatter()
- https://matplotlib.org/api/pyplot_api.html?highlight=matplotlib%20pyplot%20scatter#matplotlib.pyplot.scatter
- 知识点补充：zip封装及解包
- 坐标标签，标题，图例

  plt.xlabel()

  plt.ylabel()

  plt.title()

  plt.legend()



lect03_eg02.ipynb

基本图表的绘制及应用场景

颜色、标记、线型

- ax.plot(x, y, ‘r--’)
  - 等价于ax.plot(x, y, linestyle=‘--’, color=‘r’)

颜色

- b: blue
- g: green
- r: red
- c: cyan
- m: magenta
- y: yellow
- k: black
- w: white

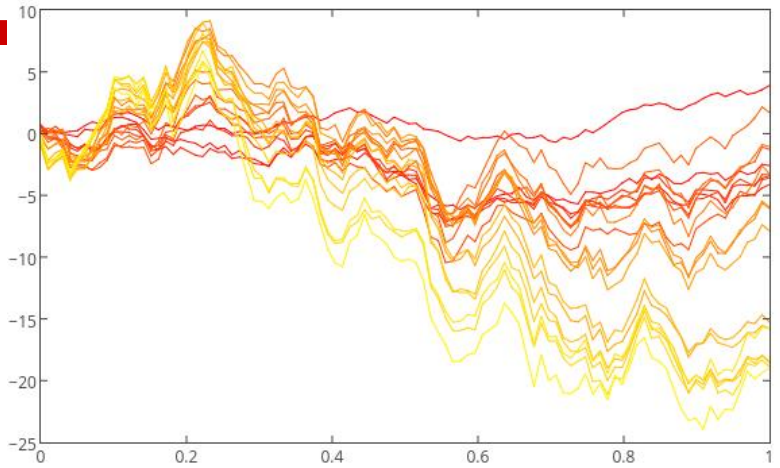
标记

marker	description
“.”	point
“,”	pixel
“o”	circle
“v”	triangle_down
“^”	triangle_up
“<”	triangle_left

线型

linestyle	description
'-' or 'solid'	solid line
'--' or 'dashed'	dashed line
'-.' or 'dashdot'	dash-dotted line
':' or 'dotted'	dotted line
'None'	draw nothing
' '	draw nothing
''	draw nothing

基本图表的绘制及应用场景

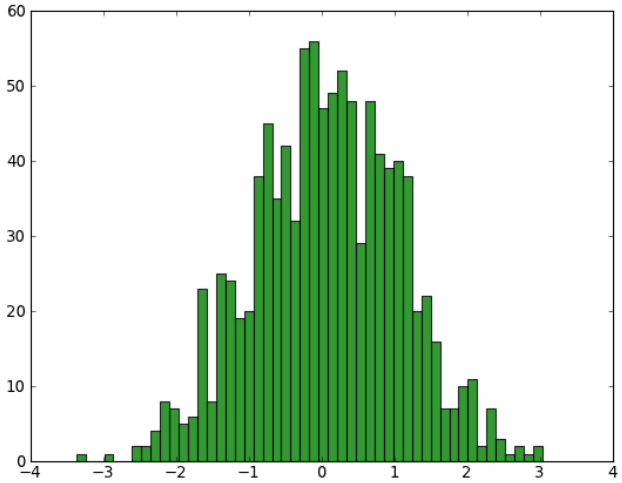


线图

- plt.plot()
- 填充线间的区域
  - plt.gca().fill_between()
- 知识点补充：np.array()生成时间数据
  - np.array(‘2017-01-01’, ‘2017-01-08’, dtype=‘datetime64[D]’)
- 绘制图像的坐标轴为时间数据时，可以借助pandas的to_datetime()完成
- 旋转坐标轴文字的方向
  - plt.xticks(rotation=) 或 遍历ticks进行set_rotation()
- 调整边界距离， plt.subplots_adjust()

```
lect03_eg02.ipynb
```

基本图表的绘制及应用场景

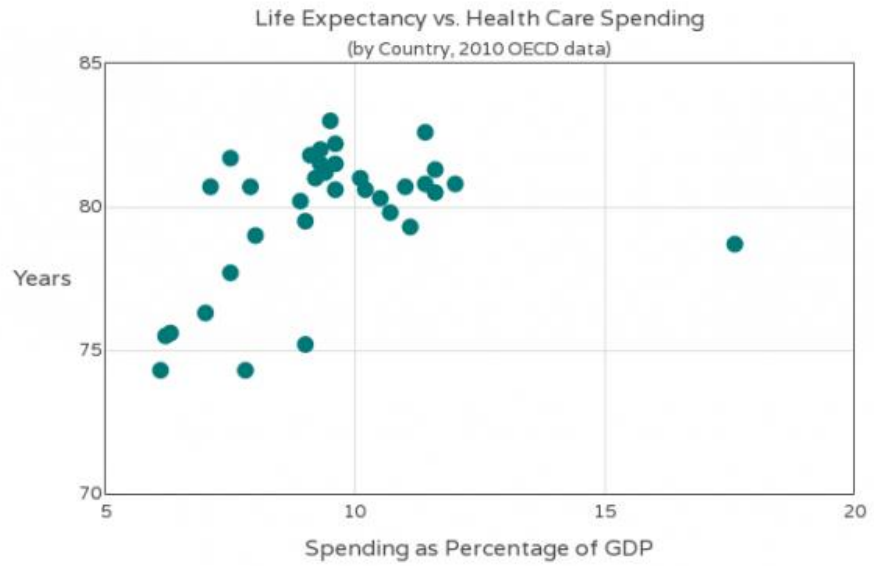


柱状图

- plt.bar()
- group bar chart
  - 同一副图中包含多个柱状图时，注意要对x轴的数据做相应的移动，避免柱状图重叠
- stack bar chart
  - 使用bottom参数
- 横向柱状图
  - barh
  - 相应的参数width变为参数height；bottom变为left

```
lect03_eg02.ipynb
```

基本图表的绘制及应用场景



应用场景

- 散点图，适用于二维或三维数据集，
  但其中只有两维需要比较。
  例子中每个数据点代表一个国家
- 线图，适用于二维数据集，
  适合进行趋势的比较
- 柱状图，适用于二维数据集，但只有一个维度需要比较。利用柱子的高度反映数据的差异。

lect03_eg02.ipynb

参考自：https://zhuanlan.zhihu.com/p/25069765

基本图表的绘制及应用场景

图表	维度	注意点
柱状图	二维	只需比较其中一维
折线图	二维	适用于较大的数据集
饼图	二维	只适用反映部分与整体的关系
散点图	二维或三维	有两个维度需要比较
气泡图	三维或四维	其中只有两维能精确辨识
雷达图	四维以上	数据点不超过6个

lect03_eg01.ipynb

参考自：https://zhuanlan.zhihu.com/p/25069765

目录

数据分析常用图表的绘制

Subplots

- plt.subplots()

直方图

- 直方图是一种对数据分布情况的图形表示
- 首先要对数据进行分组，然后统计每个分组内数据的数量。
- 作用：
  - 显示各分组频率或数量分布的情况
  - 易于显示各组之间频率或数量的差别
- plt.hist(data, bins)

  data: 数据列表

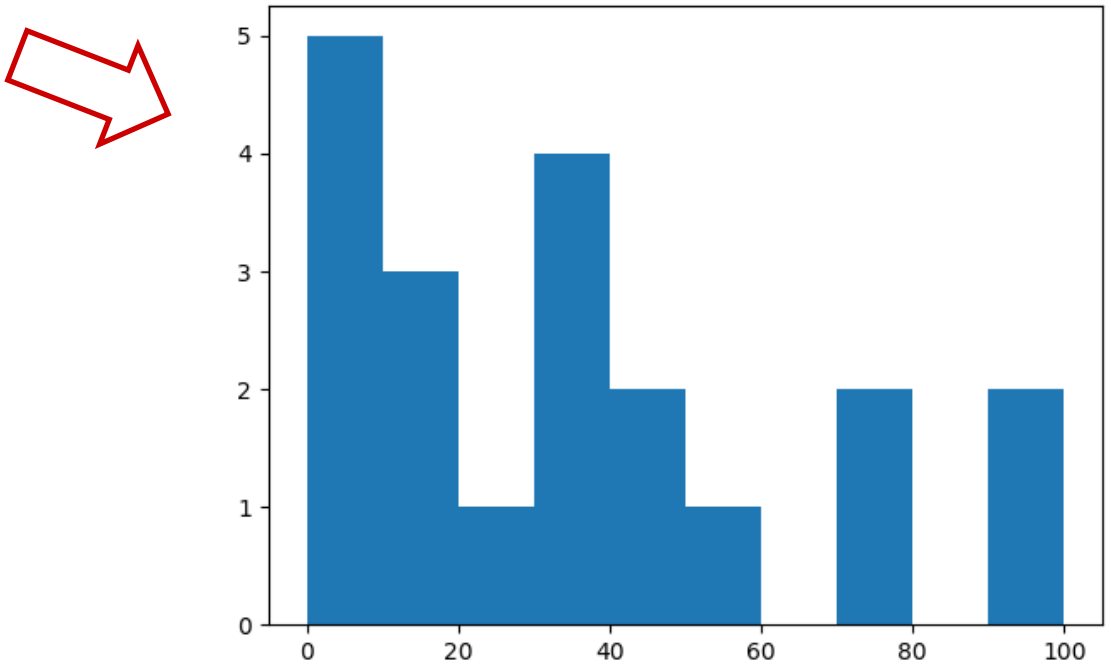
  bins: 分组边界或分组个数

lect03_eg03.ipynb

数据分析常用图表的绘制

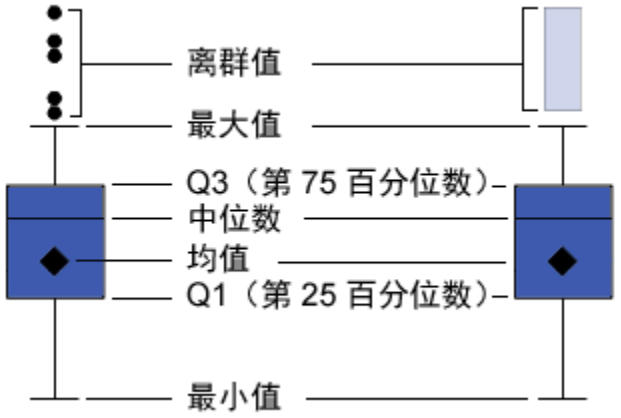
直方图

```
data = [20, 30, 33,  7, 76, 99, 31, 57, 33, 74,
        90,  2, 15, 11,  0, 41, 13, 7, 43,  6]
bins = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
plt.hist(data, bins)
```



数据分析常用图表的绘制

盒形图



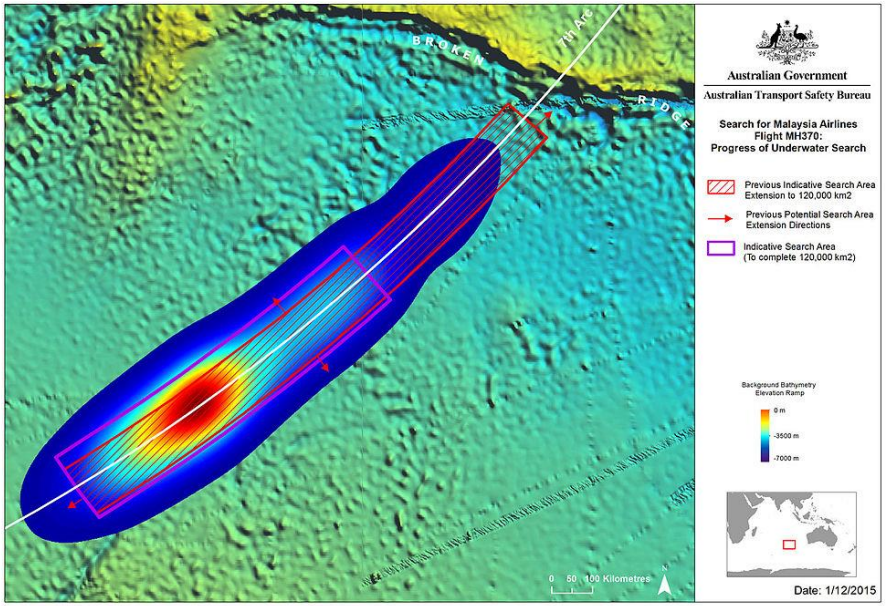
- 盒子的上边缘和下边缘指示四分位数间距 (IQR)，即介于第一个和第三个四分位数（第 25 百分位数和第 75 百分位数）之间的值范围。盒子内的标记指示均值。盒子内的线指示中位数值。
- 若不启用离群值，则须线延伸到图中的最大值和最小值。
- 若启用离群值，它们是与四分位间距的距离超过四分位间距大小 1.5 倍的数据点。
- plt.boxplot()
  - whis默认为1.5启用离群值；’range’为不启用离群值

参考自：
http://support.sas.com/documentation/cdl_alternate/zh/vaug/67500/HTML/default/n0kzo3n26iuhvhn1c1u3hwwa2qtt.htm

数据分析常用图表的绘制

热图

- 可用于三维数据的可视化
- plt.imshow(arr)
- plt.hist2d()
- plt.colorbar() 添加颜色条



基于贝叶斯模型预测马航370出现的可能地点

目录

Pandas及Seaborn绘图

Pandas 绘图

- df.plot(kind=)
  - kind用于指定绘图的类型
- pd.plotting.scatter_matrix()
- pd.plotting.parallel_coordinates()

lect03_eg04.ipynb

Pandas及Seaborn绘图

什么是Seaborn

- Python中的一个制图工具库，可以制作出吸引人的、信息量大的统计图
- 在Matplotlib上构建，支持numpy和pandas的数据结构可视化，甚至是scipy和statsmodels的统计模型可视化

特点

- 多个内置主题及颜色主题
- 可视化单一变量、二维变量用于比较数据集中各变量的分布情况
- 可视化线性回归模型中的独立变量及不独立变量

Pandas及Seaborn绘图

特点 (续)

- 可视化矩阵数据，通过聚类算法探究矩阵间的结构
- 可视化时间序列数据及不确定性的展示
- 可在分割区域制图，用于复杂的可视化

安装

- conda install seaborn
- pip install seaborn

Pandas及Seaborn绘图

数据集分布可视化

- 单变量分布 sns.distplot()
  - 直方图 sns.distplot(kde=False)
  - 核密度估计 sns.distplot(hist=False) 或 sns.kdeplot()
  - 拟合参数分布 sns.distplot(kde=False, fit=)
- 双变量分布
  - 散布图 sns.jointplot()
  - 二维直方图 Hexbin sns.jointplot(kind='hex')
  - 核密度估计 sns.jointplot(kind='kde')
- 数据集中变量间关系可视化 sns.pairplot()

Pandas及Seaborn绘图

类别数据可视化

- 类别散布图
  - sns.stripplot() 数据点会重叠
  - sns.swarmplot() 数据点避免重叠
  - hue指定子类别
- 类别内数据分布
  - 盒子图 sns.boxplot(), hue指定子类别
  - 小提琴图 sns.violinplot(), hue指定子类别
- 类别内统计图
  - 柱状图 sns.barplot()
  - 点图 sns.pointplot()

目录

其他常用的可视化工具

D3.js

- D3（Data-Driven Documents），被数据驱动的文档。是一个用动态图形显示数据的JavaScript库，一个数据可视化的工具。
- mpld3
  - http://mpld3.github.io/
  - 将Matplotlib和D3js结合起来的基于Python的可视化工具。
  - pip install mpld3

lect03_eg05.ipynb

其他常用的可视化工具

echarts

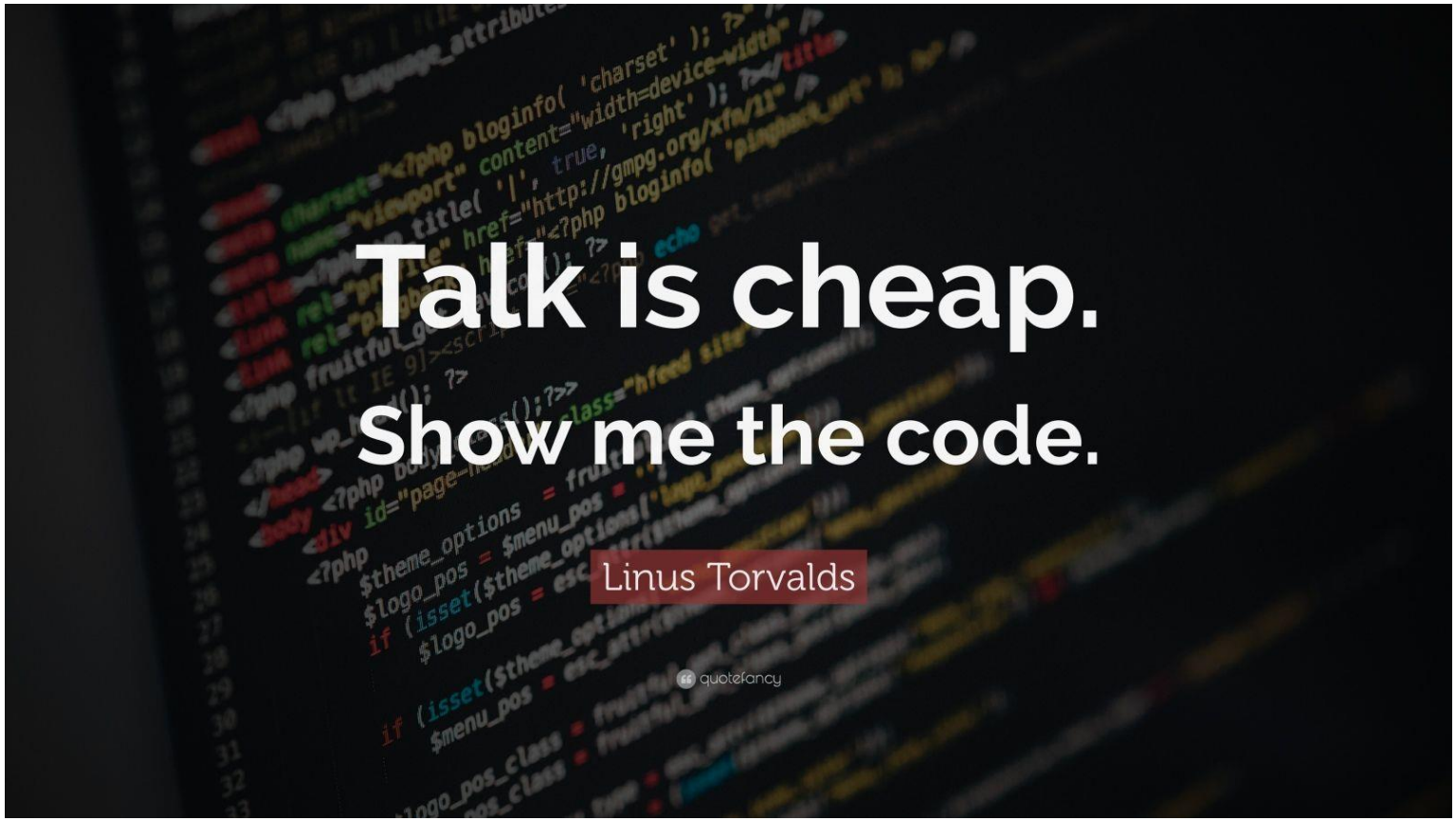
- 一个纯 Javascript 的图表库，可以流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器，底层依赖轻量级的 Canvas 类库 ZRender，提供直观，生动，可交互，可高度个性化定制的数据可视化图表。
- 2018年1月16日，ECharts （echarts.baidu.com）发布了最新大版本 4.0，新版本在产品的性能、功能、易用性等各个方面进行了全面提升。
- pyecharts
  - http://pyecharts.org/#/zh-cn/
  - pyecharts 是为了与 Python 进行对接，方便在 Python 中直接使用数据生成图
  - pip install pyecharts
  - 详细使用方法：http://pyecharts.org/#/zh-cn/prepare

目录

实战案例 2

项目名称：YouTube视频趋势分析

- 请参考相应的配套代码及案例讲解文档



疑问

- 问题答疑： http://www.xxwenda.com/
  - 可邀请老师或者其他人回答问题

小象问答邀请 @Robin_TY 回答问题

联系我们

小象学院：互联网新技术在线教育领航者

– 微信公众号：**小象学院**