

# 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。



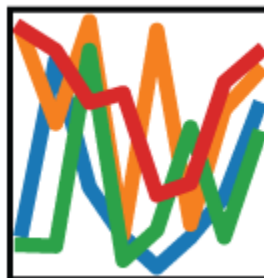
关注 小象学院

# 第二讲

---

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



## Pandas进阶及统计分析

--Robin

# 目录

---

- 基本数据对象及操作
- 数据清洗
- 数据合并及分组
- 透视表
- 实战案例1-2：中国五大城市PM2.5数据分析 (2)

# 目录

---

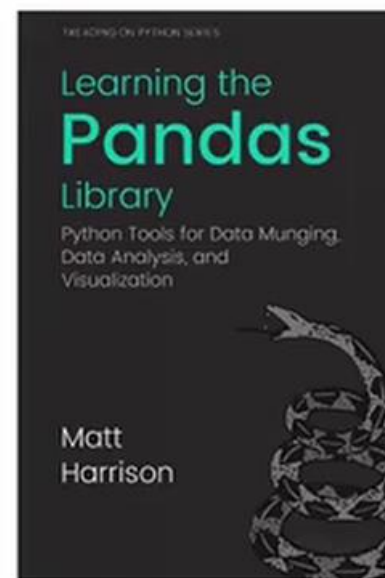
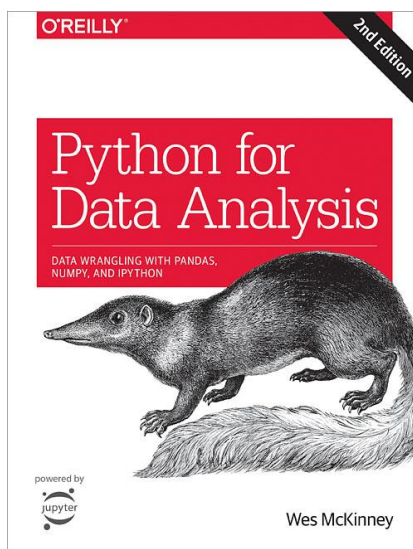
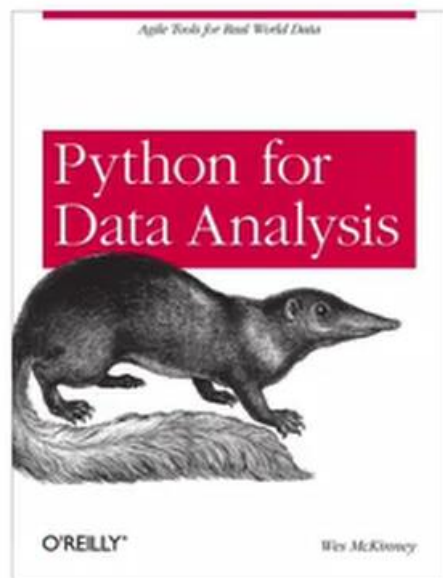
- 基本数据对象及操作
- 数据清洗
- 数据合并及分组
- 透视表
- 实战案例1-2：中国五大城市PM2.5数据分析 (2)

# 基本数据对象及操作

---

## Pandas

- 2008年由Wes McKinney创建
- 一个强大的分析结构化数据的工具集
- 基础是NumPy，提供了高性能矩阵的运算



# 基本数据对象及操作

## Series

- 类似一维数组的对象
- 通过list构建Series
  - `ser_obj = pd.Series(range(10))`
- 由数据和索引组成
  - 索引在左，数据在右
  - 索引是自动创建的
- 获取数据和索引
  - `ser_obj.index, ser_obj.values`
- 预览数据
  - `ser_obj.head(n)`

Animals	
0	Dog
1	Bear
2	Tiger
3	Moose
4	Giraffe
5	Hippopotamus
6	Mouse

`lect02_eg01.ipynb`

# 基本数据对象及操作

## Series (续)

- 通过索引名(字符串)获取数据, `ser_obj['idx_name']`, 或 `ser_obj.loc['idx_name']`
  - 如果索引名不存在, 则新建一项记录
- 通过`in`判断数据是否存在, Series也可看作定长、有序的字典
- 通过索引位置(整型数据)获取数据, `ser_obj.iloc[idx]`
- 索引与数据的对应关系仍保持在数组运算的结果中
- 通过dict构建Series
- name属性
  - `ser_obj.name`, `ser_obj.index.name`
- Pandas会根据数据类型自动处理缺失数据
  - 如object -> None, float -> NaN
- 仍然可以使用NumPy中的向量化操作

lect02\_eg01.ipynb

# 基本数据对象及操作

## DataFrame

- 类似**多维数组/表格数据** (如, excel, R中的data.frame)
- 每列数据可以是不同的类型
- 索引包括**行索引(index)**和**列索引(label)**

### Data Frame

columns

index	a	b
0	x	x
1	x	x
2	x	x
3	x	x
4	x	x

rows

lect02\_eg01.ipynb



# 基本数据对象及操作

---

## DataFrame

- 通过ndarray构建DataFrame
- 通过dict构建DataFrame
- 通过列索引获取列数据（Series类型）
  - `df_obj[label]` 或 `df_obj.label`
- 增加列数据，类似dict添加key-value
  - `df_obj[new_label] = data`
- 删除列
  - `del df_obj[col_idx]`

[lect02\\_eg01.ipynb](#)

# 基本数据对象及操作

---

## 索引操作

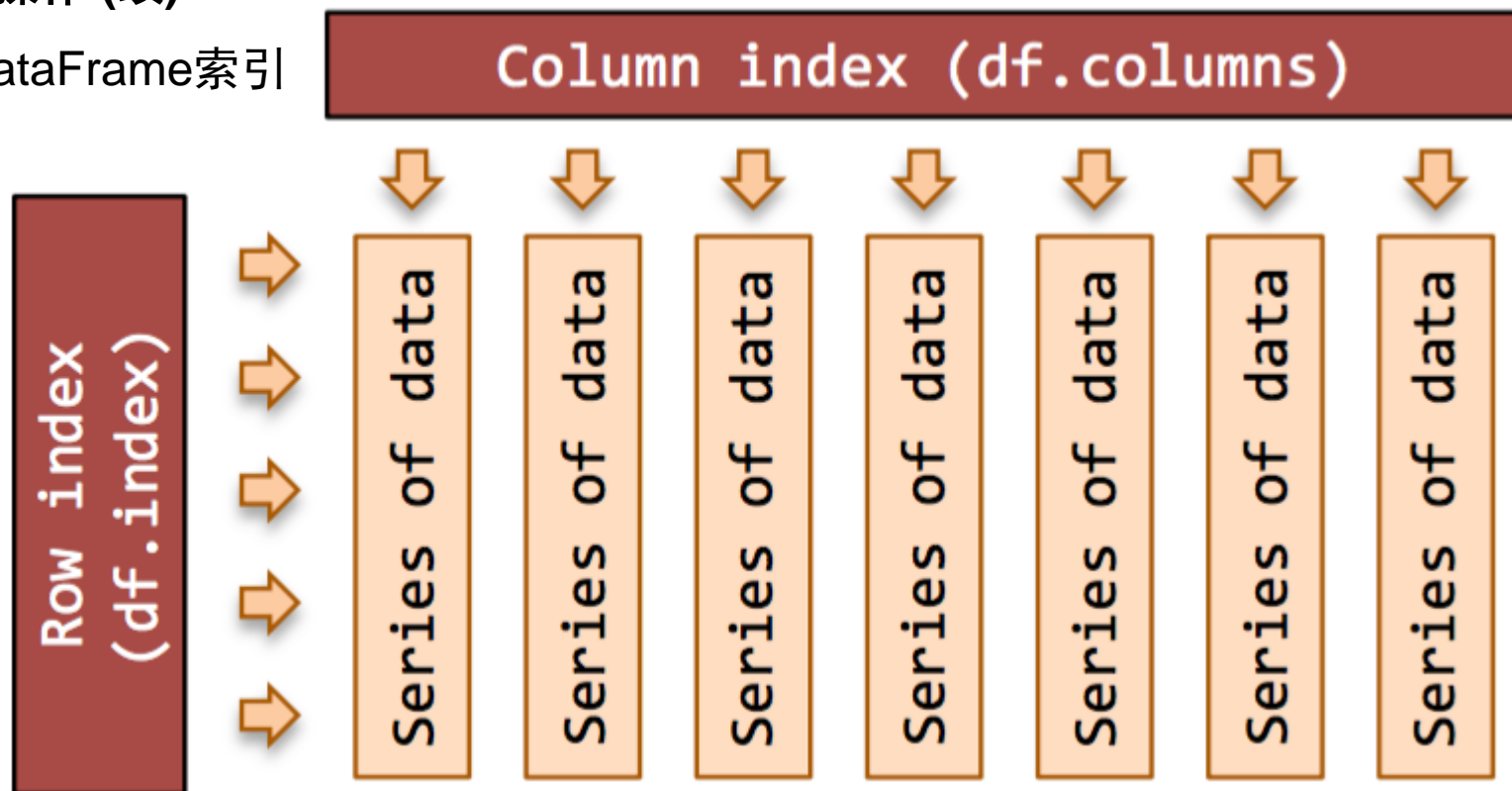
- Series索引
  - 行索引, `ser_obj['label']`, `ser_obj[pos]`
  - 切片索引, `ser_obj[2:4]`, `ser_obj['label1': 'label3']`
    - 注意, 按索引名切片操作时, 是包含终止索引的。
  - 不连续索引, `ser_obj[['label1', 'label2', 'label3']]`  
`ser_obj[[pos1, pos2, pos3]]`
  - 布尔索引

`lect02_eg01.ipynb`

# 基本数据对象及操作

## 索引操作 (续)

- DataFrame索引



lect02\_eg01.ipynb

# 基本数据对象及操作

## 索引操作 (续)

- DataFrame索引
  - 列索引: `df_obj['label']`
  - 不连续索引: `df_obj[['label1', 'label2']]`
- 注意从DataFrame中取出的数据进行操作后, 会对原始数据产生影响。为了保证不对原始数据产生影响, 应该使用`copy()`产生一个副本。在副本上进行操作。
- 数据读取
  - `pd.read_csv()`
  - `index_col`: 指定索引列
  - `usecols`: 指定需要读取的列

`lect02_eg01.ipynb`

# 基本数据对象及操作

---

## 索引操作总结

- Pandas的索引可归纳为3种
- .loc, 标签索引
- .iloc, 位置索引
- loc与iloc主要用于行索引
- .ix, 标签与位置混合索引
  - 先按标签索引尝试操作, 然后再按位置索引尝试操作
- 注意
  - DataFrame索引时可将其看作ndarray操作
  - 标签的切片索引是包含末尾位置的

lect02\_eg01.ipynb

# 基本数据对象及操作

---

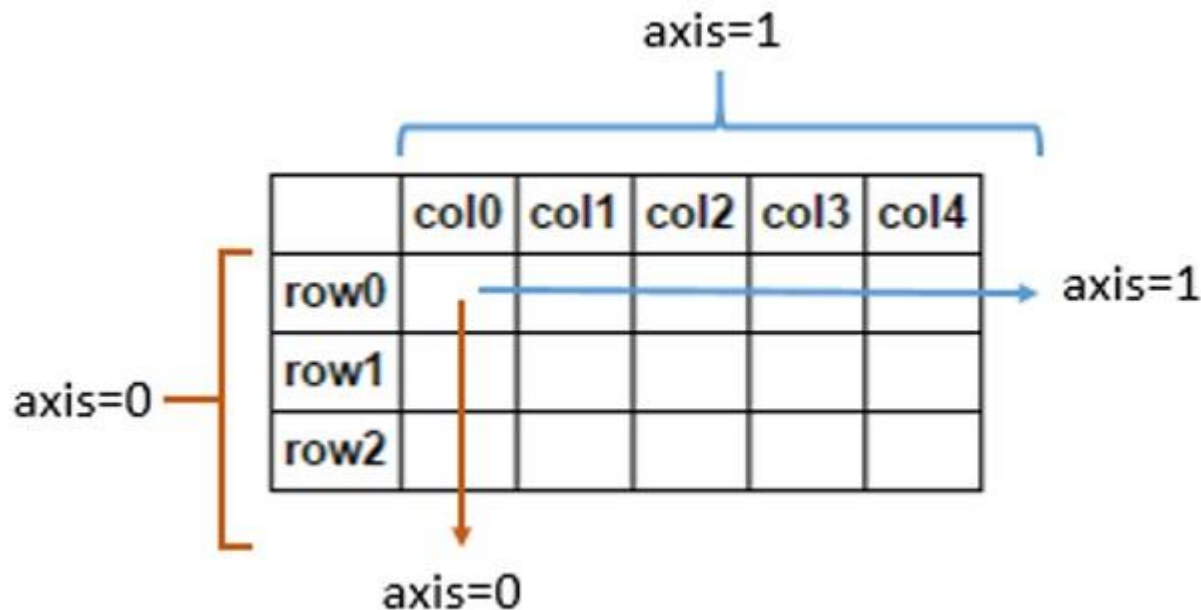
## 索引对象Index

- Series和DataFrame中的索引都是Index对象
- 不可变(**immutable**)
  - 保证了数据的安全
- 常见的Index种类
  - Index
  - Int64Index
  - MultiIndex, “层级”索引
  - DatetimeIndex, 时间戳类型
- 重置索引 `reset_index()`, 将索引重新赋值为0-1
- 重命名列名: `df.rename(columns={old_col: new_col}, inplace=True)`

# 基本数据对象及操作

## 轴的方向

- $axis=0$ , 表示纵向计算
- $axis=1$ , 表示横向计算



# 基本数据对象及操作

**Boolean mask**（布尔值遮罩 纯字面翻译 😊）

df		Boolean mask		result	
	Animals	Owners			
0	Dog	Chris	True	True	0 Dog Chris
1	Bear	Kevyn	True	True	1 Bear Kevyn
2	Tiger	Bob	False	False	
3	Moose	Vinod	True	True	3 Moose Vinod
4	Giraffe	Daniel	False	False	
5	Hippo	Fil	False	False	
6	Mouse	Stephanie	False	False	

lect02\_eg01.ipynb



# 基本数据对象及操作

## 层级索引 (hierarchical indexing)

- MultiIndex对象
- `set_index(['a','b'], inplace=True)`, 注意a, b的先后顺序
- 选取子集
  - 外层选取 `ser_obj.loc['outer_index']`
  - 内层选取 `ser_obj.loc['out_index', 'inner_index']`
- 常用于分组操作、透视表的生成等
- 交换分层顺序
  - `swaplevel()`
- 排序分层
  - `sort_index(level=)`

lect02\_eg01.ipynb

# 基本数据对象及操作

## 层级索引（续）

		0	1	2	3
bar	one	-1.133800	0.548640	1.109034	0.643708
	two	-0.792654	0.518681	-0.611958	0.913413
baz	one	0.775624	-2.520829	-0.472691	-0.557803
	two	0.190005	0.435193	1.635680	1.584821
foo	one	-0.592235	-0.361735	1.336444	-1.280014
	two	-1.016622	1.409086	0.114743	0.408211
qux	one	0.662941	-1.258482	-0.373214	-0.974658
	two	-0.931004	0.596507	0.148323	0.475039

lect02\_eg01.ipynb

# 目录

---

- 基本数据对象及操作
- **数据清洗**
- 数据合并及分组
- 透视表
- 实战案例1-2：中国五大城市PM2.5数据分析 (2)

# 数据清洗

---

## 数据清洗

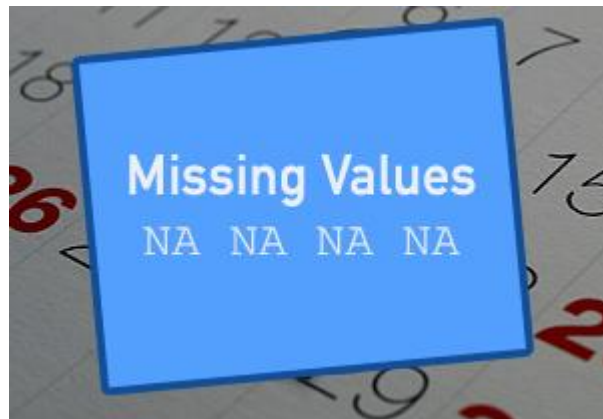
- 是数据分析**关键**的一步，直接影响之后的处理工作
- 数据需要修改吗？有什么需要修改的吗？数据应该怎么调整才能适用于接下来的分析和挖掘？
- 是一个**迭代**的过程，实际项目中可能需要不止一次地执行这些清洗操作



# 数据清洗

## 处理缺失数据

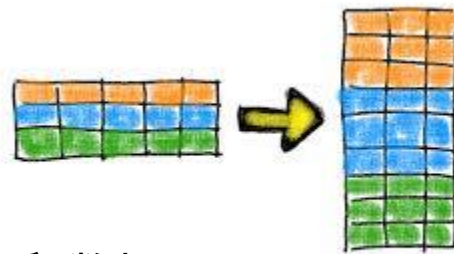
- 判断数据缺失, `ser_obj.isnull()`, `df_obj.isnull()`, 相反操作为`notnull()`
- 处理缺失数据
  - `df.fillna()`, `df.dropna()`
  - `df.ffill()`, 按之前的数据填充
  - `df.bfill()`, 按之后的数据填充
  - 项目中使用`ffill`或`bfill`时, 注意数据的排列顺序



# 数据清洗

## 数据变形

- 处理重复数据
  - 判断数据是否重复, `duplicated()`
  - 去除重复数据, `drop_duplicates()`, 可指定列及如何保留数据
- 使用函数或map转化数据, 通常根据字典进行数据转化
- 替换值, `replace()`
- 离散化和分箱操作, `pd.cut()`, 返回Categorical对象
- 哑变量操作, `pd.get_dummies()`
- 向量化字符串操作
  - 字符串列元素中是否包含子字符串, `ser_obj.str.contains()`
  - 字符串列切片操作, `ser_obj.str[a:b]`
  - ...



# 数据清洗

- 向量化字符串操作（续）

Method	Description
cat	Concatenate strings element-wise with optional delimiter
contains	Return boolean array if each string contains pattern/regex
count	Count occurrences of pattern
extract	Use a regular expression with groups to extract one or more strings from a Series of strings; the result will be a DataFrame with one column per group
endswith	Equivalent to <code>x.endswith(pattern)</code> for each element
startswith	Equivalent to <code>x.startswith(pattern)</code> for each element
findall	Compute list of all occurrences of pattern/regex for each string
get	Index into each element (retrieve <i>i</i> -th element)
isalnum	Equivalent to built-in <code>str.alnum</code>
isalpha	Equivalent to built-in <code>str.isalpha</code>
isdecimal	Equivalent to built-in <code>str.isdecimal</code>
isdigit	Equivalent to built-in <code>str.isdigit</code>
islower	Equivalent to built-in <code>str.islower</code>
isnumeric	Equivalent to built-in <code>str.isnumeric</code>
isupper	Equivalent to built-in <code>str.isupper</code>
join	Join strings in each element of the Series with passed separator
len	Compute length of each string
lower, upper	Convert cases; equivalent to <code>x.lower()</code> or <code>x.upper()</code> for each element

# 目录

---

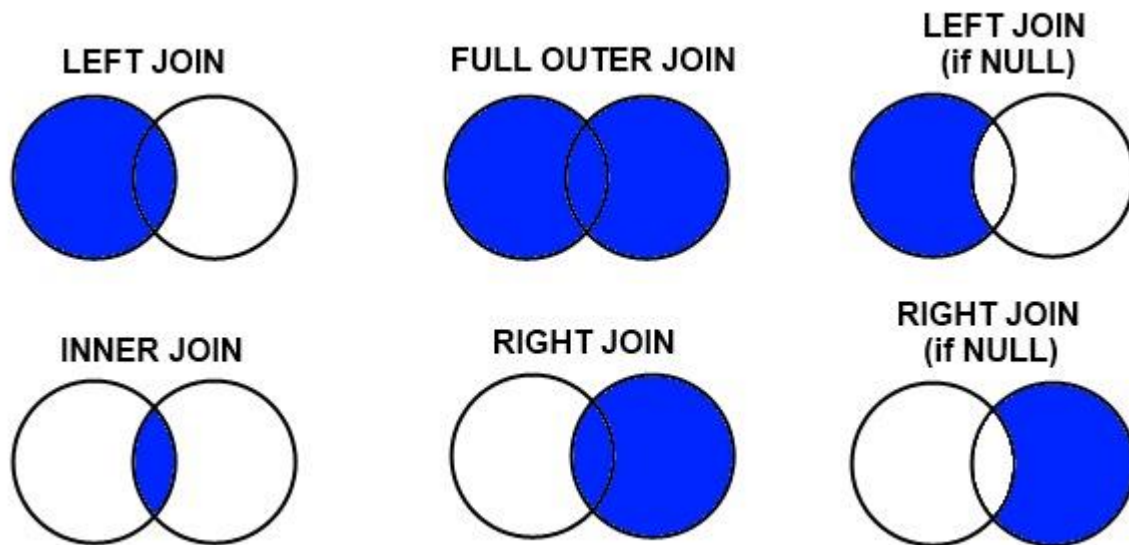
- 基本数据对象及操作
- 数据清洗
- 数据合并及分组
- 透视表
- 实战案例1-2：中国五大城市PM2.5数据分析 (2)



# 数据合并及分组

## 数据合并

- 维恩图或文氏图 (Venn diagram)



lect02\_eg03.ipynb

# 数据合并及分组

---

## pd.merge

- 根据单个或多个键将不同DataFrame的行连接起来
- 默认将重叠列的列名作为“外键”进行连接
  - on显示指定“外键”
  - left\_on, 左侧数据的“外键”
  - right\_on, 右侧数据的“外键”
- 默认是“内连接” (inner), 即结果中的键是交集

lect02\_eg03.ipynb

# 数据合并及分组

---

## pd.merge (续)

- how指定连接方式
- “外连接” (outer), 结果中的键是并集
- “左连接” (left)
- “右连接” (right)
- 处理重复列名
  - suffixes, 默认为\_x, \_y
- 按索引连接
  - left\_index=True或right\_index=True

lect02\_eg03.ipynb

# 数据合并及分组

---

## 函数应用

- 可直接使用NumPy的ufunc函数，如abs等
- 通过`apply`将函数应用到行或列上
  - 注意指定轴的方向，默认axis=0
- 通过`applymap`将函数应用到每个数据上
- `apply`的使用场景比`applymap`要多

lect02\_eg03.ipynb

# 数据合并及分组

---

## 分组 (groupby)

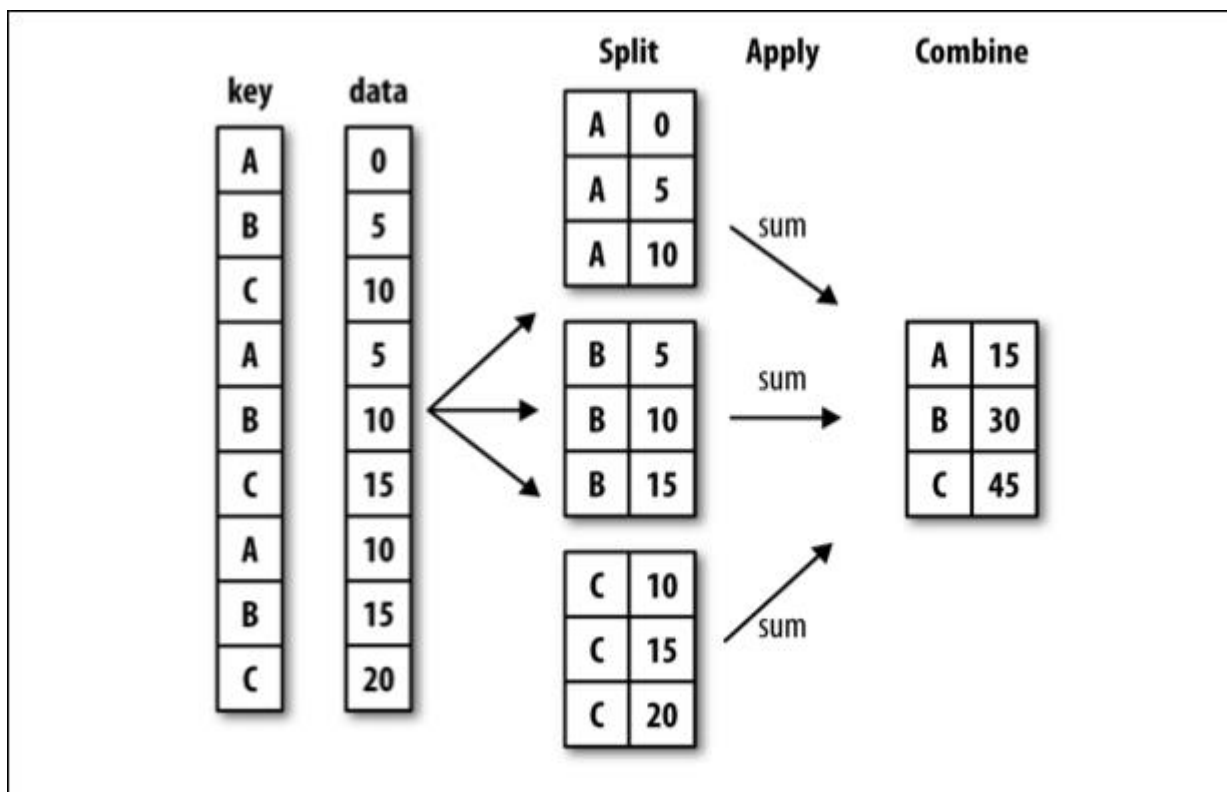
- 对数据集进行分组，然后对每组进行统计分析
- pandas能利用groupby进行更加复杂的分组运算
- 分组运算过程
  - split->apply->combine
    - 拆分：进行分组的根据
    - 应用：每个分组运行的计算规则
    - 合并：把每个分组的计算结果合并起来

lect02\_eg03.ipynb

# 数据合并及分组

## 分组 (续)

- 分组运算过程
  - split->apply->combine



# 数据合并及分组

## 分组 (续)

- GroupBy对象：DataFrameGroupBy, SeriesGroupBy
- GroupBy对象没有进行实际运算，只是包含分组的中间数据
- 对GroupBy对象进行分组运算/多重分组运算，如mean()
  - 非数值数据不进行分组运算
- size() 返回每个分组的元素个数
- 按列名分组，obj.groupby('label')
- 按列名多层分组，obj.groupby(['label1', 'label2'])->多层dataframe
- 按自定义的函数分组
  - 如果自定义函数，操作针对的是index
- 实际项目中，通常可以先人为构造出一个分组列，然后再进行groupby

lect02\_eg03.ipynb

# 数据合并及分组

---

## 分组 (续)

- GroupBy对象支持迭代操作
  - 每次迭代返回一个元组 (group\_name, group\_data)
  - 可用于分组数据的具体运算

## 聚合 (aggregation)

- grouped.agg(func), 数组产生标量的过程, 如mean()、count()等
- 常用于对分组后的数据进行计算
- 内置的聚合函数: sum(), mean(), max(), min(), count(), size(), describe()
- 可通过字典为每个列指定不同的操作方法
- 可自定义函数, 传入agg方法中

lect02\_eg03.ipynb



# 目录

---

- 基本数据对象及操作
- 数据清洗
- 数据合并及分组
- 透视表
- 实战案例1-2：中国五大城市PM2.5数据分析 (2)

# 透视表

原始数据

	A	B	C
1	月份	类别	金额
2	一月	交通	¥ 74.00
3	一月	日用杂货	¥ 235.00
4	一月	日常开销	¥ 175.00
5	一月	娱乐	¥ 100.00
6	二月	交通	¥ 115.00
7	二月	日用杂货	¥ 240.00
8	二月	日常开销	¥ 225.00
9	二月	娱乐	¥ 125.00
10	三月	交通	¥ 90.00
11	三月	日用杂货	¥ 260.00
12	三月	日常开销	¥ 200.00
13	三月	娱乐	¥ 120.00



透视表

金额汇总	列标签				
行标签	一月	二月	三月	总计	
娱乐	¥100	¥125	¥120	¥345	
日用杂货	¥235	¥240	¥260	¥735	
日常开销	¥175	¥225	¥200	¥600	
交通	¥74	¥115	¥90	¥279	
总计	¥584	¥705	¥670	¥1,959	

lect02\_eg04.ipynb

# 透视表

---

`df.pivot_table(values, index, columns, aggfunc, margins)`

- `values`: 透视表中的元素值（根据聚合函数得出的）
- `index`: 透视表的行索引
- `columns`: 透视表的列索引
- `aggfunc`: 聚合函数，可以指定多个函数
- `margins`: 表示是否对所有数据进行统计

`lect02_eg04.ipynb`

# 目录

---

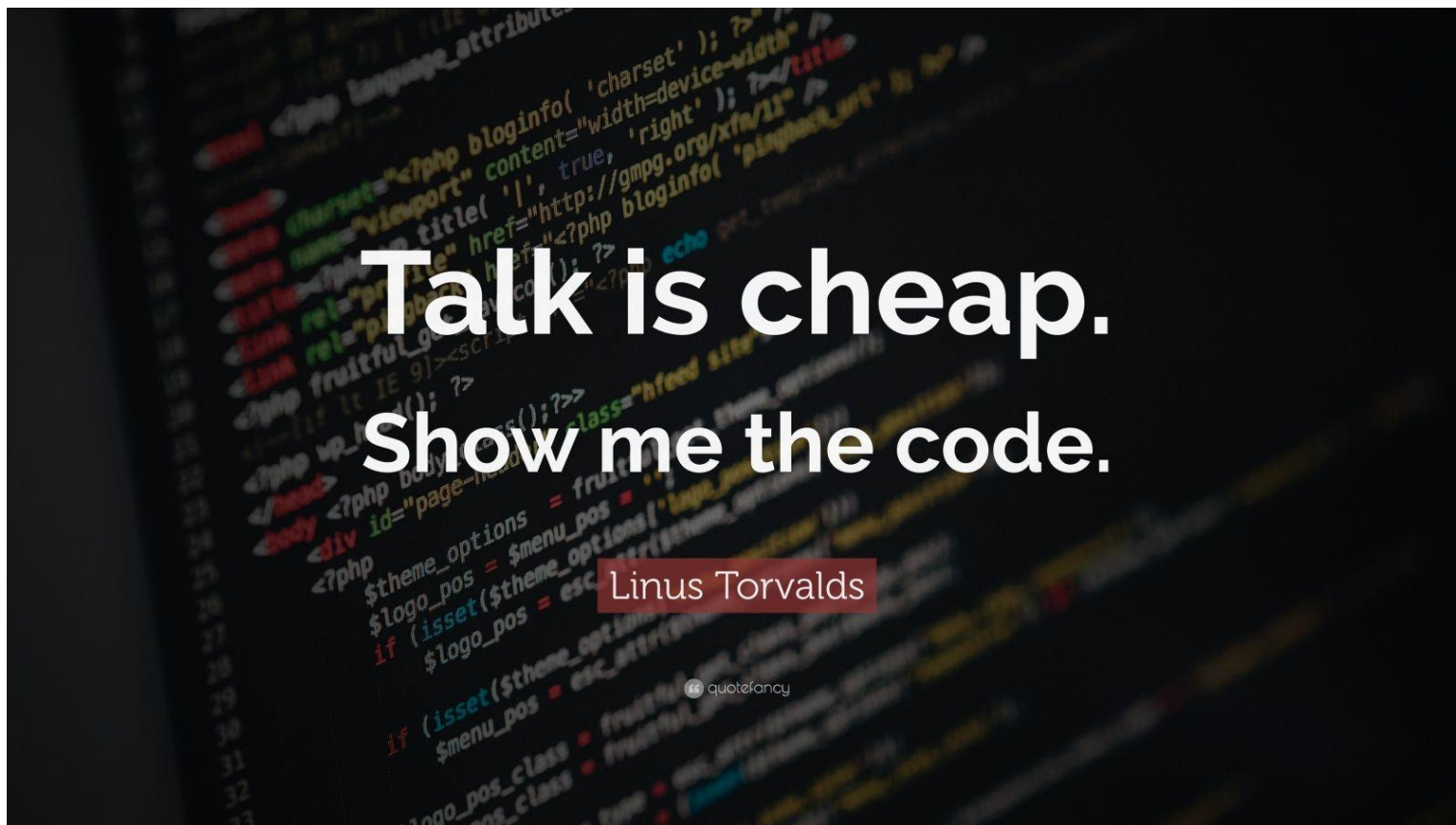
- 基本数据对象及操作
- 数据清洗
- 数据合并及分组
- 透视表
- 实战案例1-2：中国五大城市PM2.5数据分析 (2)

# 实战案例 1-2

---

项目名称：中国五大城市PM2.5数据分析（2）

- 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复问题

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**

