

## Python Pandas 库数据分析

### 一. 创建表格

```
import pandas as pd
引入库 import numpy as np # 数据计算库
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots # 画图表
time_range = pd.date_range(start='1/1/2020', end='12/31/2020')
时间序列 # date_range 生成一个固定频率的时间序列
```

```
order-table = pd.DataFrame({
```

"日期": time\_range,

"水果": fruit\_list,

})

pd.DataFrame 用于生成  
一个表格

需要依次指定每一列列名  
和对应数据来源

### 二. 合并表格

pd.merge

默认按照2张表格共有的列进行右并

参数 how: 设置合并的方式

outer: 求并集

```
final-table = pd.merge(order-table, fruit-table, how="outer")
```

### 三. 添加列

赋值方法

```
final-table["总价"] = final-table["购买量"] * final-table["单价"]
```

## 四、分析时间维度

分组:  $df1 = final\_table.groupby(\text{["年月"]}) \setminus$   
 同一组数据时...  $["购买量"].sum().reset\_index()$   
 求和

画图 ①  $fig1 = px.bar(df1, x="年月", y="购买量", color="购买量")$   
 柱状图  $fig1.show()$  显示黑点

折线图 ②  $fig2 = px.line(df2, x="年月", y="总价", markers=True)$

③  $df3 = final\_table.groupby("年份") \setminus$   
 $\setminus .agg(\{"购买量": "sum", "总价": "sum"\}).reset\_index()$   
 计算的  $\setminus$

个指标  $\setminus$  # agg 对数据作聚合操作 (max, sum, min, count)  
 求和 计数

④  $df5 = final\_table.groupby(["姓名"]) \setminus$   
 $\setminus .agg(\{"日期": "count", "总价": "sum"\}).reset\_index()$   
 $\setminus .rename(columns=\{"日期": "购买次数"\})$

⑤ 表格排序  $df6 = df5.sort\_values(["姓名", "购买次数"])$   
 $\setminus \dots ascending=[True, False]$

## 爬虫

(-) import requests

response = requests.get(url='网址')

print(response.text)

(=) selenium 库 模拟浏览器请求

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

# 初始化 webdriver

driver = webdriver.Chrome()

# 访问目标网站 url='网址' driver.get(url)

定位到目标所在前缀  
元素 element\_前缀元素 = driver.find\_element

(By.class\_name,

继续寻找此目录的下一级(可遍历) for in element\_前缀元素

所找元素

element\_元素2 = element\_前缀元素.find\_element(By.TAG\_NAME, '元素2')

☆ 若是网址, 则点击

element\_元素2.click()

或获取网址 href = element\_元素2.get\_attribute('href')

新开一个标签页 driver.execute\_script("window.open('');")

切换到标签页(driver.switch\_to.window(driver.window\_handles[1]))

打开网址 driver.get(href) (driver.window\_handles[1])

(可停留2秒) time.sleep(2)

关闭 driver.close

切回去原页 driver.switch\_to.window(driver.window\_handles[0])