经济数据数量化分析 课程教案

授课题目：实验一：数据采集及其清洗			
教学时数：	4 学时	授课类型：	☐ 理论课 ☑ 实践课

教学目的、要求：

1. 熟悉 Python 基本的环境配置。
2. 了解基本容器结构以及第三方数据形式。
3. 熟悉常见的数据采集方法。
4. 掌握常见的数据清洗方法。

教学重点：

Numpy 和 pandas 创建数据容器、数据采集、数据清洗。

教学难点：

数据采集中的数据提取。

教学方法和手段：

教学方法：采用启发式教学、情境教学法以及与问题式教学法相结合的方法。
教学手段：多媒体教学。

教学条件：

多媒体教室

参考资料：

1.《经济数据数量化分析》，朱顺泉等主编，清华大学出版社，2022 年。
2.《Python 金融大数据分析应用》，甘晓丽主编，上海交通大学出版社，2023 年。
3.《Python 数据分析与应用》，黄红梅主编，人民邮电出版社出版社，2018 年。
4 .知乎网

教学内容及过程 旁批

教学引入：

引导性提问需要教师根据教材内容和学生实际水平，提出问题，启发引导学生去解决问题，提问，从而达到理解、掌握知识，发展各种能力和提高思想觉悟的目的。

（1）环境变量是什么？
（2）现实生活中数据怎么存储的？
（3）该如何发现数据蕴藏的规律？
（4）你所了解的有哪些数据采集方法？
（5）Python 实现数据清洗的常用方法？

教学内容与教学设计：

第一章 Python 基本环境配置

（插入：
Python 和编辑器两者之间的关系？环境变量设置的作用是什么？
）

一、Python 安装

1. 操作系统的位数
可通过以下操作确定：右击此电脑 -> 点击属性 -> 查看位数
2. 添加环境变量
3. 手动配置环境变量
右击此电脑
-> 单击属性
-> 点击左侧的"高级系统设置"
点击 "环境变量"
单击选中"Path" -> 单击编辑
在“编辑环境变量”选项卡
单击新建

添加环境变量是 Python 安装过程中的关键步骤。

二、Pip 安装第三方库

不追求速度方法：
pip install 库名

追求速度方法：
pip install 库名 -i https://pypi.tuna.tsinghua.edu.cn/simple

三、编辑器

Anaconda
Pycharm
VScode

两种方式新建文件，任意输入简单内容演示，让学生动手

四、运行文件

temp.ipynb
Temp.py

第二章 基本容器结构以及第三方数据形式

（插入：
数据容器是用以盛纳东西的容器，一是 Python 内置的容器，二是第三方库的容器。

一、字符串

1. 定义

理论部分

字符串是 Python 语言中的一个基本数据类型，用来表示一串字符。Python 中的字符串是一串由引号包围的字符。这些引号可以是单引号 (') 或双引号 (")，另一种创建字符串的方法是使用 str 函数，它接收一个对象并返回该对象的字符串表示。
（引入：案例 1：str 函数创建
案例 2：索引（要求掌握索引的具体位置）
案例 3：拼接
）

代码部分

```
############创建字符串
str_temp1 = '武汉工商学院'
str_temp2 = ''
print(str_temp1)

############索引
print('############正向索引############')
```

```
for i,j in enumerate(str_temp1):
    print(i,j)
print('#############反向索引#############')
for i,j in enumerate(str_temp1):
    print(i-len(str_temp1),j)
print(str_temp1[0:2])
print(str_temp1[-4:])

############拼接
print('hello'+' '+'world!')
```

练习部分

自己上网任意找一家上市公司的股票代码（比如 000001.SZ），

(1) 通过索引和拼接方法，将 000001.SZ 处理为 SZ.000001
(2) 将股票代码的六位数索引出来

2. 常用函数

代码部分

演示完，让学生复习回顾基本操作。

```
str_temp = Wuhan University of Technology and Business
str_temp.upper( )              #所有字母大写
str_temp.lower( )              #所有字母小写

print(' '.join(['武汉','工商','学院']))    #前面的字符串作为连接符
print(','.join(['武汉','工商','学院']))

print('武,汉,工,商,学,院'.split(','))#以括号中的内容为分隔符

print('武汉工商学院'.startswith('武汉'))
```

练习部分

:

自己上网任意找一家上市公司的股票代码（比如 000001.SZ），

(1) 股票代码的字母全部小写（或者大写字母小写）
(2) 通过字符串中的 split 方法，将股票代码中的代码和字母隔开
(3) 判断股票代码是 6 开头，还是 0 开头

二、列表

1. 定义

理论部分

```
list1 = ['武汉市','黄石市','十堰市','宜昌市','襄阳市','鄂州市',
         '荆门市','孝感市','荆州市','黄冈市','咸宁市','随州市']
#湖北省所有地级市
```

案例 1：
通过索引方法，将武汉市，黄石市、十堰市索引出来
案例 2：拼接江苏十三个地级市

代码部分

```
############索引
print('#############正向索引#############')
for i,j in enumerate(list1):
    print(i,j)
print('#############反向索引#############')
for i,j in enumerate(list1):
    print(i-len(list1),j)
print(list1[:6])

##########拼接
print(list1+['宿迁市'])
print(list1*2)
```

练习部分

自己上网任意找三家上海交易所的上市公司股票代码和三家深证交易所上市公司的股票代码，

(1) list_temp1,list_temp2 分别装两个交易所上市公司的股票代码
(2) 通过拼接方法，将两个交易所的股票代码放在一个列表容器 list_temp3

2. 常用函数

理论部分

```
list1.append(元素)                    #添加元素
list1.extend(list2)                          #添加列表
```

```
list1.count(元素)                    #某个元素在列表中出现的次数
list1.index(元素)                    #返回某个元素索引位置
list1.reverse( )                     #列表中的元素反向存放

list1.pop()                          #移除列表中的一个元素（默认最后一个）
list1.remove(元素)                   #移除列表中某一个值的第一个匹配项
```

代码部分

```
list1.extend(['南京市','苏州市','无锡市','常州市','南通市','泰州市','扬州市','淮安市','淮安市','盐城市','徐州市','连云港市','宿迁市'])
list1.append('宿迁市')
print(list1)
print(list1.count('宿迁市'))

list1.reverse()
list1.pop()
print(list1)
```

练习部分

自己上网任意找三家上海交易所的上市公司股票代码和三家深证交易所上市公司的股票代码，
再继续找一家任意的上市公司股票代码，

(1) 分别通过列表的 append 和 extend 方法将新的上市公司股票代码，加入到已有的列表 list_temp3.
(2)
通过 pop 方法，将添加的最后一家上市公司股票代码移除（金融）
通过 reverse 和 pop 方法，将列表 list_temp3 中的第一家上市公司股票代码移除（数字经济）

三、字典

1. 定义
理论部分

字典是使用键-值存储，用冒号分割键和值，用逗号分割不用的的项。
d = { ‘湖北省’ : ’武汉市’, ‘江苏省’ : ’南京市’ }
#创建字典的方式一

items = [(‘湖北省’,’武汉市’) , (‘江苏省’,’南京市’)]
dict(items)
#创建字典的方式二，dict 函数

练习部分

自己上网任意找三家上海交易所的上市公司股票代码和三家深证交易所上市公司的股票代码（list_temp3)
创建一字典 dict_temp1，键为上述上市公司名称，值为上述上市公司股票代码

2. 常用函数
理论部分

d_up = { ‘浙江省’ : ‘杭州市’ }
d.update(d_up)
{ ‘湖北省’ : ’武汉市’, ‘江苏省’ : ’南京市’, ‘浙江省’ : ‘杭州市’ }

案例 1：增加新的项
字典[键名]=值名；update 函数
案例 2：查找某个项是否在字典
键名 in 字典
案例 3：获取字典中的所有键和值
keys 函数（）；values 函数（）

代码部分

```
############创建字典
d = { '湖北省':'武汉市','江苏省':'南京市'}
d_up = {'浙江省':'杭州市'}
d.update( d_up )
############常用函数
'浙江省' in d
d.keys()
d.values()
```

练习部分

自己上网任意找三家上海交易所的上市公司股票代码和三家深证交易所的上市公司股票代码，再继续找一家任意的上市公司股票代码，

(1) 字典增加项方法，在 dict_temp1 基础上增加新项（新寻找的上市公司代码）
(2) 查找贵州茅台是否在 dict_temp1 数据库中
(3) 获取字典 dict_temp1 中所有上市公司名称和上市公司代码

四、集合

集合是唯一元素组成的无序集，因为是无序集，不记录元素位置，因此不支持索引、分片等常见类似序列的操作，支持并、交、差等运算。

```
set1 = {1,2,3,8,9}
set2 = {2,4,5,6,8}
set3 = set1&set2   #交集
set4 = set1|set2   #并集
```

五、控制语句

理论部分

for 循环语句

```
d = { '湖北省':'武汉市','江苏省':'南京市'}
for i in d.keys():
    print(i)
```

if 语句语句

```
for i,j in d.items():
    if i == '湖北省':
        print('武汉市')
    else:
        print('南京市')
```

练习部分

沪市主板股票代码：600、601、603、605
深市主板股票代码：000
创业板股票代码：300
科创板股票代码：688

自己上网任意找三家上海交易所的上市公司股票代码和三家深证交易所上市公司的股票代码（list_temp3），通过 for 循环遍历和 if 语句判断每只股票是沪市还是深市？（提醒：判断字符串开头常用函数 startswith）

六、常用第三方数据格式

理论部分

数组：

```
import numpy as np
np.array([1,2,3])
np.asarray([[1,2,3],[2,3,4]])
```

Series 和 DataFrame 对象：

```
import pandas as pd
pd.Series([1,2,3])
pd.DataFrame({'A':[1,2,3],'B':[2,3,4]})
```

练习部分

自己上网任意找三家上海交易所的上市公司股票代码和三家深证交易所上市公司的股票代码（list_temp3），创建一个两列的 DataFrame，第一列为上述全部上市公司的股票代码，第二列为上市公司名称。

第三章 基础的数据采集

一、挖地兔财经数据采集

理论部分

Tushare 运行三年多以来，数据从广度和深度都得到了提升，Pro 版正是在此基础上做了更大的改进。数据内容将扩大到包含股票、基金、期货、债券、外汇、行业大数据，为各类金融投资和研究人员提供适用的数据和工具。

```
import tushare as ts
pro = ts.pro_api('token')
pro = ts.pro_api()
df = pro.daily(ts_code='000001.SZ', start_date='20180701', end_date='20180718')
```

二、证券宝财经数据采集

理论部分

证券宝 www.baostock.com 是一个免费、开源的证券数据平台（无需注册）。提供大量准确、完整的证券历史行情数据、上市公司财务数据

等。

通过 python API 获取证券数据信息，满足量化交易投资者、数量金融爱好者、计量经济从业者数据需求。
返回的数据格式：pandas DataFrame 类型，以便于用 pandas/NumPy/Matplotlib 进行数据分析和可视化。同时支持通过 BaoStock 的数据存储功能，将数据全部保存到本地后进行分析。
支持语言：目前版本 BaoStock.com 目前只支持 Python3.5 及以上(暂不支持 python 2.x)。

三、爬虫采集

理论部分

请求方式：Get 是向服务器发索取数据的一种请求，而 Post 是向服务器提交数据的一种请求

```
headers = {'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML,
                          like Gecko) Chrome/100.0.4896.75 Safari/537.36'}
response = requests.post('http://quote.stockstar.com/stock/sha.shtml',headers=headers)
response.encoding='gb2312'

with open('html.txt','w') as f:
    f.write(response.text)
with open('html.txt','r') as f:
    f.readline()
```

代码部分

```
import re
import requests
from lxml import etree
import pandas as pd

def get_one_page(url):                    #爬取网页源代码
    headers = {'.....'}
    response = requests.post(url,headers=headers)
    response.encoding='gb2312'
    if response.status_code ==200:
        return response.text

def parse_pne_page1(html):                #解析页面，提取想要的信息
    pattern = re.compile('<.....',re.S)
    #使.可以匹配包括换行在内的所有字符
```

```
    items = re.findall(pattern,html)
    return items

def parse_pne_page2(html): #解析页面，提取想要的信息
    html = etree.HTML(html)
    items = html.xpath('....')
    return items

def write_to_file(items):#保存内容
    dataframe_temp = pd.DataFrame(
        {...}
    )
    print(dataframe_temp)
    return dataframe_temp

def main():#组合函数
    url1 = 'http://quote.stockstar.com/stock/sha.shtml'
    html = get_one_page(url1)
    items = parse_pne_page2(html)
    write_to_file(items).to_csv('/Users/xuliang/Desktop/temp.csv')
main()
```

第四章 基础的数据清洗

一、Numpy 库

理论部分

```
arr = np.arange(1,10).reshape(3,3)
print(arr*arr)
print(arr+arr)
print(arr/arr)
print(arr-arr)

print(np.eye(3))
print(np.square(arr))
print(np.sqrt(arr))
```

黑白图片（灰度图）通过 2 维向量（矩阵）来表达。2 个维度的长度分别代表了图片的高度和宽度（以像素为单位），向量元素记录着每一个像素的灰度（数值越大，颜色越浅）。
彩色图片通过 3 维向量来表达。3 个维度的长度分别代表了图片的高度、宽度（以像素为单位）和通道数。在 RGB 模式下，彩色图片有 3 个通道，保存着图片各个像素红色（Red）、绿色（Green）和蓝色（Blue）的量化数值。

代码部分

```
import matplotlib.pyplot as plt
import numpy as np
# 读取图像，指定格式为 PNG
image = plt.imread('/Users/xuliang/Desktop/img.jpeg', format='jpeg')
image_temp2 = image*[50,1,1,1]
image_temp3 = image*[1,50,1,1]
image_temp4 = image*[1,1,50,1]
image_temp5 = image*[1,1,1,0.4]
## 显示图像
plt.figure()
plt.imshow(image[100:-100,100:-100,:])
plt.figure()
plt.imshow(image_temp2)
plt.figure()
plt.imshow(image_temp3)
plt.figure()
plt.imshow(image_temp4)
plt.figure()
plt.imshow(image_temp5)
plt.show()
```

练习部分

可以自行查阅网上资料，尝试写一份 Python 代码，创建两个数组 arr，如何对上述两个数组进行线性代数中的两个矩阵乘法运算？

二、Pandas 库

理论部分

更改索引（行列）

```
data_temp = pd.DataFrame({'A':[1,2,3],'B':[2,3,4]},index=['a1','a2','a3'])
data_temp.index = ['a','b','c']
data_temp.columns = ['x','y']
print(data_temp
```

删除指定轴

```
data_temp = pd.DataFrame({'x':[1,2,3],'y':[2,3,4]},index=['a','b','c'])
data_temp.drop('a',axis=0)
data_temp.drop('x',axis=1)
```

索引

```
data_temp = pd.DataFrame({'x':[1,2,3],'y':[2,3,4]},index=['a','b','c'])
print(data_temp.loc['a',:])
print(data_temp.loc[:,'x'])
print(data_temp.iloc[:,0:1])
#注意 loc 和 iloc 两个函数用法区别

函数应用与映射
data_temp = pd.DataFrame({'x':[1,2,3],'y':[2,3,4]},index=['a','b','c'])
def func(x):
    return x.max()-x.min()
data_temp.apply(func,axis=0)
data_temp.apply(func,axis=1)
```

练习部分

自行创建 3 行 3 列的数据框 DataFrame，DataFrame 中全部都是数值，行索引为 x,y,z；列索引为 a,b,c。

（1）数据标准化，（x-均值）/标准差

（2）通过用 apply 方法进行数据标准化

三、Numpy、Pandas 库数据存取

理论部分

```
numpy_temp = np.arange(12).reshape(3,4)
np.save('...',numpy_temp)          #numpy 数据存储
np.load('.....')                   #numpy 数据读取
```

注意：文件位置如何确定？

```
dataframe_temp = pd.DataFrame(np.arange(12).reshape(3,4))
dataframe_temp.to_csv('....',index=None)
pd.read_csv('...')
```

练习部分

金融专业（批量数据文件存储）：

自行创建一个 DataFrame 文件，以此为基础批量生成 1000 个 csv 文件，保存在任意一个文件夹下。

数字经济专业（批量数据文件读取）：

data 文件夹下，读取全部六位数股票代码文件名称（0 开头和 6 开头）的 csv 文件，注意并不读取全部数据，每次仅读取每个 csv 文件的最后一行数据。

```
（提醒：os.listdir()               #显示某一文件夹下所有文件名
filename.endswith('.xlsx')         #判断文件名后缀格式）
```

实验小结

这一实验项目分为四个部分，第一部分内容要求大家客观看待这门专业课程、了解专业课程的内容安排以及学习要求；第二部分内容，给同学进行初步演示，引导大家对这门专业产生兴趣，并鼓励学生勤加动手。第三部分内容是数据采集部分，旨在通过理论内容和实践内容相结合，引导学生学习这部分新的知识，课堂中多加入学生自主动手环节，加深学生的学习体验。

课后思考

下一章节知识准备：是否已具备相应的知识储备
问题 1：如何通过 Python 第三方库进行数据存取？
问题 2：数据网络爬虫方式采集的基本流程？

重要名词：

Pandas 库、numpy 库、数据采集、数据检验和处理

教学内容及过程	旁批

教学内容与教学设计：

实验二 金融数据库搭建及使用

一、**数据存储概述**

1. 基本结构

理论部分

正式学习 SQL 语言之前，首先让我们对 SQL 语言有一个基本认识：

名称	定义
数据库	保存有组织的数据的容器，简单地想象为一个文件柜。
表	某种特定类型数据的结构化清单，相关资料应放入特定的文件。
主键	一行（一组列），其值能够唯一标识表中的每一行。

主键需满足以下条件：
任意两行都不具有相同的主键值；
每一行都必须具有一个主键值（主键列不允许 NULL 值）

2. 基本操作

代码部分

启动 MySQL：	mysql.server start
免密：	mysql -u root
退出 MySQL：	exit;
关闭 MySQL：	mysql.server stop

启动 MySQL：	net start mysql84
免密：	mysql -u root -p
密码：	123456
退出 MySQL：	exit;
关闭 MySQL：	mysql.server stop

备注：启动两种方式，一是运行输入 services.msc，手动启动 mysql；二是管理员身份运行命令行，启动具体内容就是上述代码。

3. 数据库操作

代码部分

```
show database;                    #显示全部数据库
```

退出命令后面加上分号。

```
create database 数据库名;                #创建数据库
use 数据库名;                            #连接数据库
select database();                       #显示当前选定的数据库
drop database 数据库名;                  #删除数据库
```

二、**Python 与数据库联系**

1. 数据预览（MySQL Workbench、Navicat Premium 可视化工具）

SQLyog 是一个易于使用的、快速而简洁的图形化管理 MYSQL 数据库的工具，它能够在任何地点有效地管理你的数据库。

SQLyog 是业界著名的 Webyog 公司出品的一款简洁高效、功能强大的图形化 MySQL 数据库管理工具。

使用 SQLyog 可以快速直观地让您从世界的任何角落通过网络来维护远端的 MySQL 数据库。

两种方式新建文件，任意输入简单内容演示，让学生动手

2. 数据读取

代码部分

```
import pymysql
import pandas as pd
from sqlalchemy import create_engine

###########读取数据库
db = pymysql.connect(host = 'localhost',user = 'root', password = '123456',
database = 'abcde', port=3306, charset = 'utf8')
sql = "select * from abc;"
# table_name 某个表名
df_data = pd.read_sql(sql,db)
print(df_data)
```

备注：

1. host：主机，port：端口，user：用户名，password：密码
2. 通过 pandas 和 pymysql 第三方库读取数据库中数据

3. 数据保存

代码部分

```
import pymysql
import pandas as pd
from sqlalchemy import create_engine

###########读取数据库
df_data = pd.DataFrame({'a':[1,2,3],'b':[2,3,4],'c':[3,4,5]})
engine = create_engine('mysql+pymysql://root:@localhost:3306/abcde')
con = engine.connect()
df_data.to_sql(name="students5", con=con, if_exists='append',
index=False)
# 写入数据库，如果已存在该表，则追加写入数据，不加索引
```

```
with engine.connect() as con:
    con.execute("""ALTER TABLE `{}`.`{}` \
                ADD COLUMN `id` INT NOT NULL AUTO_INCREMENT FIRST, \
                ADD PRIMARY KEY (`id`);"""
                    .format("abcde", "students5"))
#关闭数据库连接
con.close()
```

三、**数据库常用方法**

1. 创建数据表

```
create table temp(
a varchar(20) not null,
b char(10) not null,
primary key(a)
);
```

课堂任务：

如何通过 pandas 实现类似的操作？（DataFrame 对象列名为 a 和 b）

2. 插入数据

```
insert into temp values(1,2,3);
#插入一行数据；
alter table temp add( d decimal(10,2) );
#插入一列数据；
select * from temp;
```

课堂任务：

如何通过 pandas 实现类似的操作？

给定 sz000001.csv,按照公式(close-open)/open 计算，增加新的一列，列名为 new。

3. 检索数据

```
Select prod_name from Products;
#检索单列；从表 Products 中检索 prod_name 列名
Select prod_id, prod_name, prod_price from Products;
#检索多列；从表 Products 中检索 prod_name 列名
select prod_name from Products limit 5;
#检索单列前五行；从表 Products 中检索 prod_name 列名
```

课堂任务：

如何通过 pandas 实现类似的操作？

4. 数据更新

update temp set d=1;
select * from temp;
#update 表名 set 列名=数值；
#插入一列数据

课堂任务：
如何通过 pandas 实现类似的操作？

5. 数据计算

select a,b,d,b*d as c from temp;
#select 检索新的列 c，不更改表 temp；
select * from temp;

课堂任务：
如何通过 pandas 实现类似的操作？（生成新的列）

6. 字段拼接与别名

update temp set c=concat(b,d);
select * from temp;
#b 和 d 拼接位列 c；

课堂任务：
如何通过 pandas 实现类似的操作？（生成新的列，提示：字符串拼接）

实验小结

启动 mysq：net start MySQL84
登录账号密码：MySQL -u root -p

Python 可以正常读取 sql 文件

退出 MySQL：exit;

关闭 MySQL：net stop MySQL84

课后思考

下一章节知识准备：是否已具备相应的知识储备
问题 1：如何通过 Python 第三方库进行数据存取？
问题 2：数据网络爬虫方式采集的基本流程？

重要名词：

Pandas 库、numpy 库、数据采集、数据检验和处理

教学内容及过程	旁批

教学内容与教学设计：

实验三 证券相关数据分析及挖掘

一、数据预处理

项目简介：
基于沪深 300 成分股数据，从中选取一定数量的股票（比如 100 股），旨在构造指数增强产品。假定是以下这一具体投资策略：每天实盘交易是从沪深 300 只成分股中选取 100 股涨幅靠前的股票，期望跑赢大盘？

1. **沪深 300 成分股数据**
下载沪深 300 成分股数据，预处理为收益率，如何拼接到一起？

任务一：如何拼接 300 只成分股数据？
注意事项：收益率计算（当日收盘价-上一个收盘价）/上一个收盘价

任务二：计算每只股票的收益率？
注意事项：需要下载每只股票至少两期以上

下载沪深 300 成分股数据，预处理为收益率，如何拼接到一起？

退出命令后面加上分号。

案例数据：

A	B
SZ01	2
SZ01	3

A	B
SZ02	2
SZ02	3

1. concat 函数拼接，如何使用？
dataframe = pd.concat([dataframe,result],axis=0)

2. 分组计算（收益率公式），如何使用？
data.groupby('A',group_keys=True).apply(lambda x: (x-x.shift(1))/(x.shift(1)))

2. **沪添加时间戳**

下载武进不锈股票数据，如何添加时间戳?

```
data = pd.read_csv('603878.csv',index_col='date')
data.index = pd.to_datetime(data.index)
```

两种方式新建文件，任意输入简单内容演示，让学生动手

二、数据筛选

1. str 用法

沪深 300 成分股数据，如何分别筛选出上海和深圳两市的股票？（单元格部分匹配）

```
data[data['A'].str.contains('SZ01')]
```

2. 条件索引

沪深 300 成分股数据，如何分别筛选出上海和深圳两市某只具体股票？

```
data[data['A']=='SZ01']
```

3. isin 用法

沪深 300 成分股数据，如何分别筛选出上海和深圳两市符合具体价格的股票？（单元格内容完全匹配）

```
data[ data['B'].isin([2]) ]
```

三、数据分组

沪深 300 成分股数据，将 300 只股票的收益率按照大小分为高、中、低三组，筛选出’高‘组：

```
a,b = pd.cut(x=data["close"],bins=3,right=True,retbins=True,labels=['低','中','高'])
```

四、数据排序

沪深 300 成分股数据，将 300 只股票的收益率进行排序，筛选出前 50 名：

code	ret	排序
sz000001	-0.03	6

sz000003	-0.02	5
sz000004	-0.01	4
sh000001	0.01	3
sh000002	0.02	2
sh000003	0.03	1

实验小结

数据处理操作：数据拼接、分组、排序

课后思考

下一章节知识准备：是否已具备相应的知识储备
问题 1：如何通过 pandas 第三方库进行数据处理？
问题 2：数据网络爬虫方式采集的基本流程？

重要名词：

Pandas 库、数据检验和处理

教学内容及过程	旁批

教学内容与教学设计：

实验四 Python 与常用统计计量方法的应用

一、概率统计分布的 Python 应用

（一）正态分布

1.背景

看涨看跌期权定价公式，其中其中，C 和 P 为欧式看涨期权和看跌期权价格，S0 为期权合约中标的资产的初始价格，X 为合约规定的执行价格，r 为无风险利率，T 为合约期限从期权合约距结束的时间长度。

线性回归模型中的基本假定

统计检验

单样本 t 检验：	广告策略是否影响销售人员的销售量
两个独立样本 t 检验	70、80 年代标准普尔指数月度平均收益率
配对样本 t 检验	同一指数两种基金平均收益率
单样本方差检验	某只股票某一个时间段的标准差
双样本方差检验	两只股票月收益率的标准差
方差分析	推进器和燃气两种因素显著影响射程

2.中心极限定理

在客观实际中有许多随机变量，它们是由大量的相对独立的随机因素的综合因素所形成的，而其中每一个别因素在总的影响中所起的作用都是微小的。这种随机变量往往近似地服从正态分布。

3.Python 应用

```
x = np.arange(-4,4,0.1)          #生成等差数列
y = stats.norm.pdf(x,0,1)        #计算概率密度
plt.plot(x,y)
data = stats.norm.rvs(loc=0.0,scale=1.0,size=10000)
plt.hist(data,bins=20)
#模拟 1000 个均值为 0，方差为 1 服从正态分布随机变量的样本点
```

退出命令后面加上分号。

#频数分布

两种方式新建文件，任意输入简单内容演示，让学生动手

（二）卡方分布

1.定义

设 X1,X2,...,Xn 是来自总体的样本，则称统计量

$\chi 2=X12+X22+...+Xn2$

服从自由度为 n 的 $\chi 2$ 分布。

2.Python 应用

```
x = np.arange(0,5,0.002)      #生成等差数列
y = stats.chi.pdf(x,3)        #计算概率密度
plt.plot(x,y)
data = stats.chi.rvs(1,size=10000)
plt.hist(data,bins=100)
#模拟 10000 个自由度为 1 服从卡方分布随机变量的样本点
#频数分布
```

（三）t 分布

1.定义

设 X~N(0,1)，Y~$\chi 2$(n)，且 X，Y 相互独立，则称随机变量

$t=X/\sqrt{Y/n}$

服从自由度为 n 的 t 分布。

2.Python 应用

```
x = np.arange(-4,4,0.004)     #生成等差数列
y = stats.t.pdf(x,5)          #计算概率密度
plt.plot(x,y)
data = stats.chi.rvs(1,size=10000)
plt.hist(data,bins=100)
#模拟 10000 个自由度为 1 服从卡方分布随机变量的样本点
#频数分布
```

（四）F 分布

1.定义

设 U~χ2(n1)，V~χ2(n2)，且 U，V 相互独立，则称随机变量

F=U/n_1/V/n_2

服从自由度为(n1,n2)的 F 分布。

2.Python 应用

```
x = np.arange(0,7,0.007)          #生成等差数列
y = stats.f.pdf(x,4,10)                  #计算概率密度
plt.plot(x,y)
data = stats.f.rvs(4,40,size=10000)
plt.hist(data,bins=100)
#模拟 10000 个自由度为(4,40)服从 F 分布随机变量的样本点
#频数分布
```

二、参数假设检验的 Python 应用

（一）单个样本 t 检验

通过单样本 t 检验，实现样本均值和总体均值的比较。检验步骤是：首先提出原假设，规定好显著性水平，然后确定合适的检验统计量，并计算检验统计量的值，最后依据计算值和临界值得比较做出统计决策。

科研研究:

某研究团队开发了一种新型降压药，并对其进行了临床试验。通过单样本 t 检验比较新药治疗后的患者血压均值与正常血压均值是否存在显著差异，结果表明新药具有显著的降压效果，且与正常血压均值无显著差异，从而为新药的推广和应用提供了科学依据。

市场调研：

一个手机制造商可能想要了解消费者对其新手机的平均满意度是否与设定的满意度目标有显著差异。制造商可以进行一次市场调研，收集消费者对新手机的满意度评分，并使用单样本 t 检验来进行分析。

质量控制：

生产线上的某个产品批次，其重量、尺寸或其他关键质量指标可能需要与标准值进行比较，以确定该批产品是否符合质量标准。

心理研究：

某研究者对一组接受心理治疗的焦虑症患者进行了单样本 t 检验，以比较其治疗前后的焦虑水平是否与正常人群的焦虑水平存在显著差异。结果表明，治疗后患

者的焦虑水平显著降低，与正常人群的焦虑水平无显著差异，从而验证了心理治疗的有效性

课堂练习：

1.基于 sz000001 的收盘价，计算其收益率。利用单个样本 t 检验，检验该公司的平均收益率 与 0 是否存在显著差异？

stats.ttest_1samp(data,popmean=0.0001)

2.某种元件的寿命服从正态分布 N，均值和方差未知，现测得 16 只元件的寿命，问是否有理由认为元件的平均寿命不大于 225h？（拓展部分）

（二）两个独立样本 t 检验

通过独立样本 t 检验，实现两个独立样本的均值比较。检验步骤是：首先提出原假设，规定好显著性水平，然后确定合适的检验统计量，并计算检验统计量的值，最后依据计算值和临界值得比较做出统计决策。

科教学研究:

比较两种不同教学方法对学生成绩的影响，可以收集两组学生的成绩数据，一组采用传统教学方法，另一组采用新的教学方法，然后使用两个独立样本 t 检验来比较两组学生的平均成绩是否存在显著差异。

医学研究：

比较两种不同药物对同一种疾病的治疗效果，可以收集两组患者的治疗数据，一组使用第一种药物，另一组使用第二种药物，然后使用两个独立样本 t 检验来比较两组患者的治疗效果是否存在显著差异。

市场调研：

分析不同营销策略对消费者购买行为的影响，可以分别收集接受不同营销策略的消费者数据，并使用两个独立样本 t 检验来比较两组消费者的购买意愿或购买量是否存在显著差异。

课堂练习：

基于 sz000001、sz000002 的收盘价，计算其收益率。假定两个总体方差未知且相等。利用两个独立样本 t 检验，检验两只股票 2020 年收益率有无明显的差别？

提示：

(1)添加时间戳

(2)独立 t 检验：stats.ttest_ind(data1,data2)

（三）配对样本 t 检验

通过配对样本 t 检验，实现成对数据的样本均值比较。检验步骤是：首先提出原假设，规定好显著性水平，然后确定合适的检验统计量，并计算检验统计量的值，最后依据计算值和临界值得比较做出统计决策。

课堂练习：

基于上证综指的收盘价，计算其收益率。第二次国九条实施（2013.12.27)前后，检验该政策是否显著影响股票市场价格。假定利用配对样本 t 检验，检验 2012-2013 年的收益率 与 2014-2015 年的收益率均值是否一致？

提示：

(1)下载指数数据

(2)添加时间戳

(3)配对 t 检验：stats.ttest_rel(data1,data2)

三、参数估计及其检验

（一）模型形式

（二）拟合优度检验

1. 总离差平方和（TSS)

被解释变量的样本观测值与其平均值的离差平方和

2. 回归平方和(ESS)

被解释变量的样本估计值与其平均值的离差平方和，这部分是模型回归线可以解释的部分

3. 残差平方和(RSS)

被解释变量观测值与估计值之差的平方和，这部分是模型回归线解释不了的部分

（三）显著性检验

```
from statsmodels.formula.api import ols
lm = ols('price ~ area + bedrooms + bathrooms', data=df).fit( )
lm.summary()
```

课堂练习：

基于 sz000001、sz000002、sz000004 中的收盘价和成交量两列，进行一元线性回归。收盘价为被解释变量，成交量为解释变量，说明两者之间是否存在显著的线性关系。