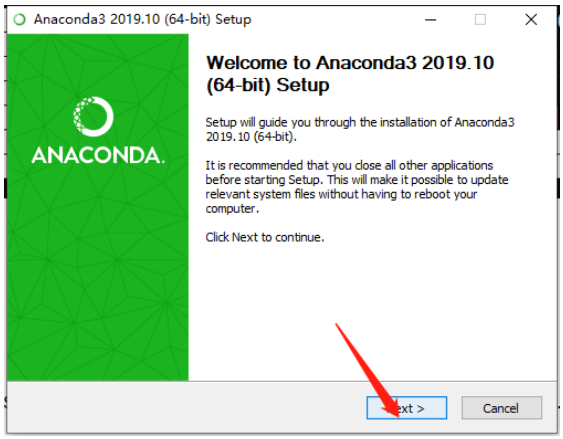
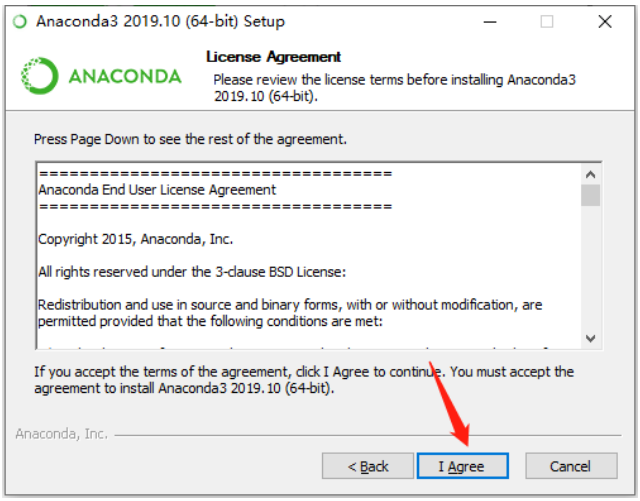
目录

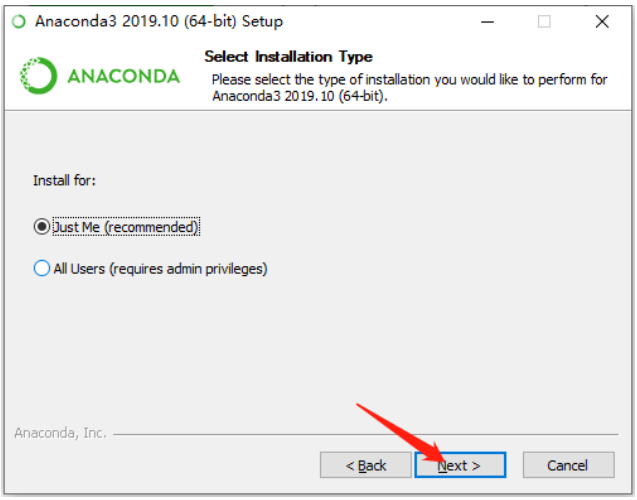
一、 工具安装

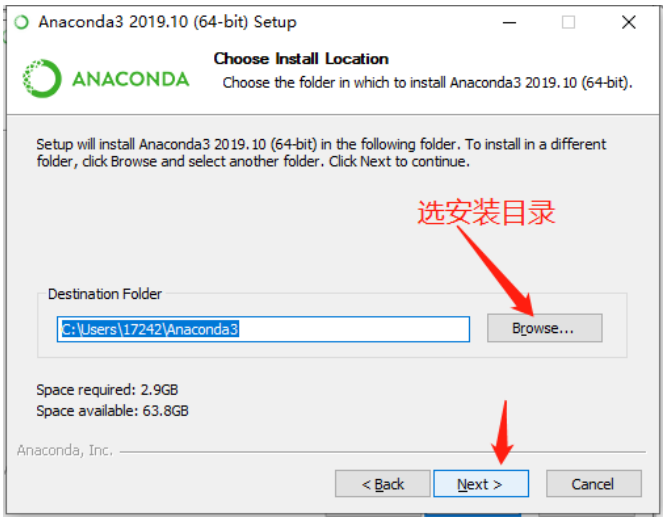
1.1 工具安装

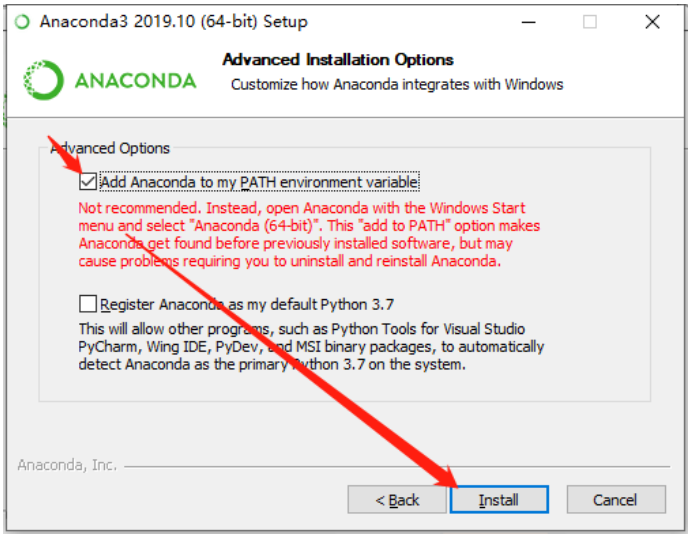
- Anaconda3:
https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2020.02-Windows-x86_64.exe






Anaconda3 2019.10 (64-bit) Setup
ANACONDA
Select Installation Type
Please select the type of installation you would like to perform for Anaconda3 2019.10 (64-bit).
Install for:
Just Me (recommended)
All Users (requires admin privileges)
Anaconda, Inc.
< Back
Next >
Cancel


Anaconda3 2019.10 (64-bit) Setup
ANACONDA
Choose Install Location
Choose the folder in which to install Anaconda3 2019.10 (64-bit).
Setup will install Anaconda3 2019.10 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
选安装目录
Destination Folder
C:\Users\17242\Anaconda3
Browse...
Space required: 2.9GB
Space available: 63.8GB
Anaconda, Inc.
< Back
Next >
Cancel


Anaconda3 2019.10 (64-bit) Setup
ANACONDA
Advanced Installation Options
Customize how Anaconda integrates with Windows
Advanced Options
Add Anaconda to my PATH environment variable
Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.
Register Anaconda as my default Python 3.7
This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.7 on the system.
Anaconda, Inc.
< Back
Install
Cancel

等待安装即可!!!

1.2pip 库安装

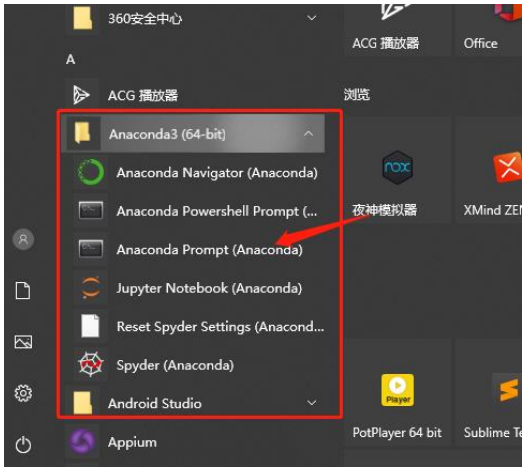
- `pip install -i https://pypi.tuna.tsinghua.edu.cn/simple/  xlrd`

```
C:\Users\17242>pip install -i https://pypi.tuna.tsinghua.edu.cn/simple/ xlrd
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple/
Collecting xlrd
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/b0/16/63576a1a001752e34
bf8ea62e367997530dc553b689356b9879339cf45a4/xlrd-1.2.0-py2.py3-none-any.whl (103
kB)
     |                                | 51kB 656kB/s eta 0:00:01
     |                                | 61kB 787kB/s eta 0:00
     |                                | 71kB 762kB/s eta 0
     |                                | 81kB 751kB/s e
     |                                | 92kB 845kB/
     |                                | 102kB 93
     |                                | 112kB 93
9kB/s
```

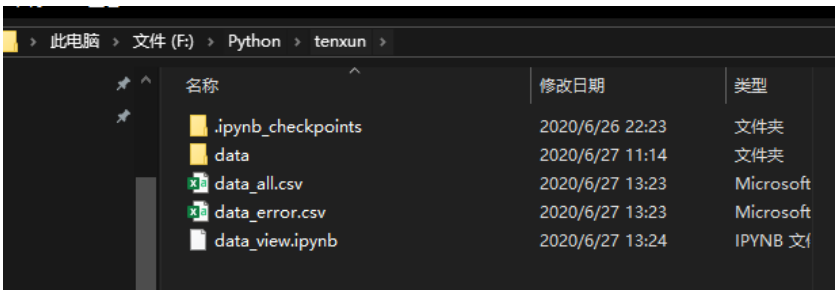
二、操作步骤

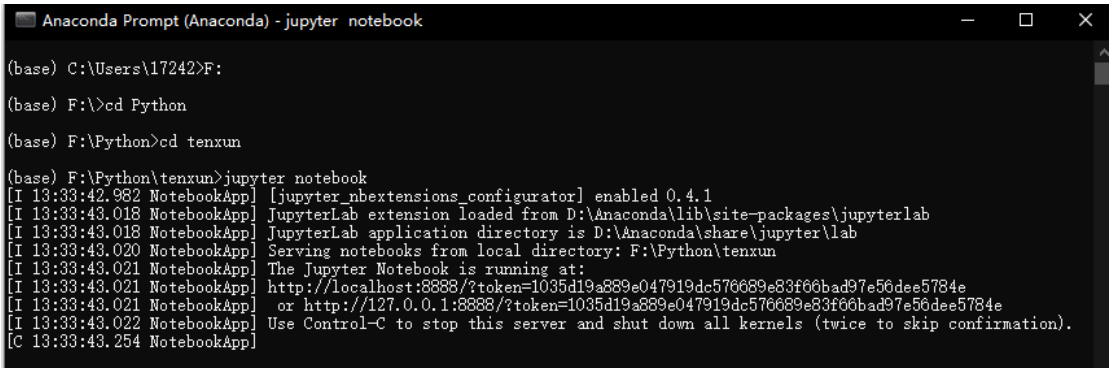
2.1 打开 jupyter notebook

打开命令行:



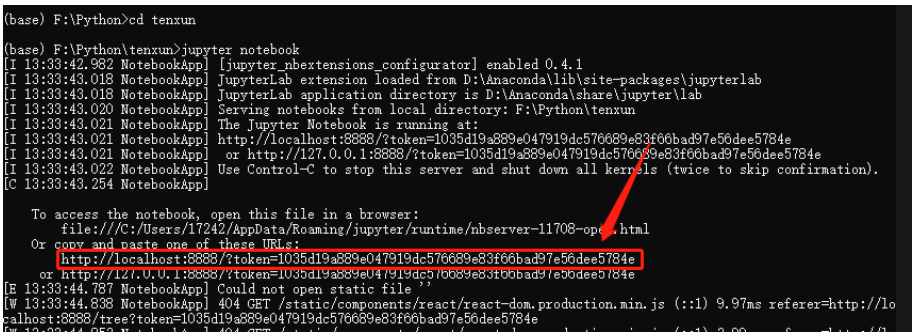
找到源文件位置:
例如:





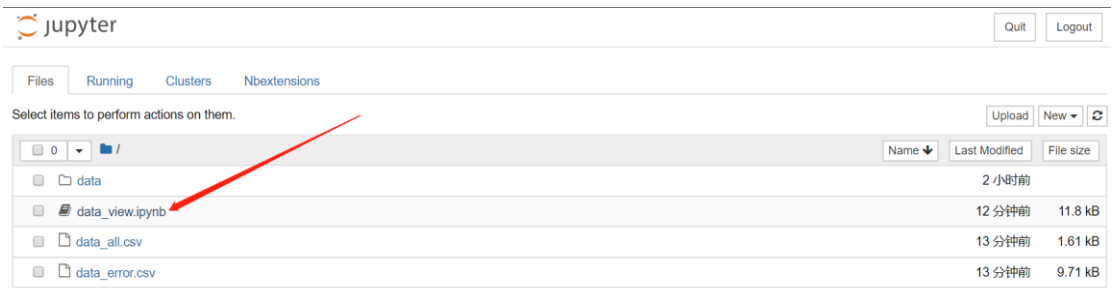
找到源文件目录，输入 **jupyter notebook** 会自动打开浏览器!!! 例如：如下图

备注：



如果浏览器未自动打开，则自己复制该链接在浏览器打开即可!!!

2.2 运行



每个框框里面让其执行一次(每选一个框框，按 **Shift + Enter** 键):

In [14]:
```
# coding=utf-8
import os
import re
import csv
import time
import xlrd
# 记录所有数据
data_all = {}
# 时长不足
data_no_time = []
# 没签到
data_no_sign = []
# 保存头部
csv_headers = ['学号', '姓名', '观看次数', '签到次数']
csv_one_headers = ['学号','名字','观看直播时长','签到','直播时间']
```

In [2]:
```
class Utils(object):
    """
    工具类
    """
    def select_number(self, text):
        """
        清洗出数字
        :text:文本
        """
        number_tmp = re.search(r'\d+', text) # 避免格式错误导致NoneType
        number = '' # 学号
        if number_tmp:
            number = number_tmp.group()
        return number
```

选中情况下，框框的边为绿色！！！

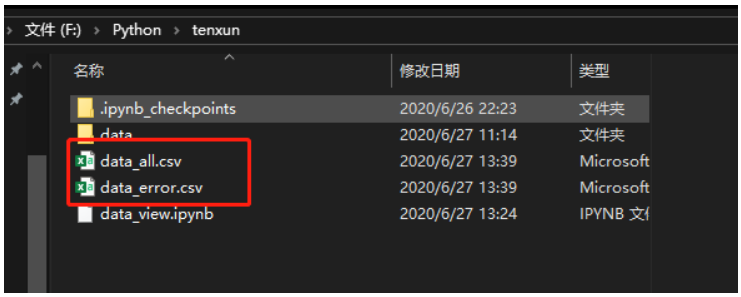
In [2]:
```
class Utils(object):
    """
    工具类
    """
    def select_number(self, text):
        """
        清洗出数字
        :text:文本
        """
        number_tmp = re.search(r'\d+', text) # 避免格式错误导致NoneType
        number = '' # 学号
        if number_tmp:
            number = number_tmp.group()
        return number
    def select_no_number(self, text):
        """
        清除数字的文本
        :text:文本
        """
        txt = re.sub('[^A-Za-z\u4e00-\u9fa5]', '', text).lstrip()
        return txt
    def select_bool(self, text):
        """
```


选中最后一个代码框之后，就会开始执行!!!

In [6]:
```
# 每次更新,防止追加
data_all= {}
data_no_sign = []
data_no_time = []
if __name__ == '__main__':
    student = Student()
    # 写入全部数据头部
    with open('data_all.csv',mode = 'a', newline='') as f:
            writer = csv.DictWriter(f, csv_headers)
            writer.writeheader()
    # 写入时长未到&没签到
    with open('data_error.csv',mode = 'a', newline='') as f:
            writer = csv.DictWriter(f, csv_one_headers)
            writer.writeheader()
    for path in os.listdir("data"):
        path = 'data/{}'.format(path)
        data , all_time = student.clean_one(path) # 单个excel文件数据ss
        student.stat_data(data,all_time) # 所有数据
    num = student.clean_error(data_all) # 清洗data_all学号格式错误的人员
    num_no_time = student.clean_error(data_no_time) # 清洗data_no_time学号格式错误的人员s
    num_no_sign = student.clean_error(data_no_sign) # 清洗data_no_sign学号格式错误的人员
    # 存储全部数据
    save = Save()
    save.csv_save(data_all, 'data_all.csv',  csv_headers)
    save.csv_save(data_no_time,'data_error.csv',csv_one_headers) # 时长未到
    save.csv_save(data_no_sign,'data_error.csv',csv_one_headers) # 未签到
```

```
这节课还没签到功能： 2020-03-11 09:50:31
这节课还没签到功能： 2020-03-18 09:42:12
这节课还没签到功能： 2020-03-19 07:41:54
```

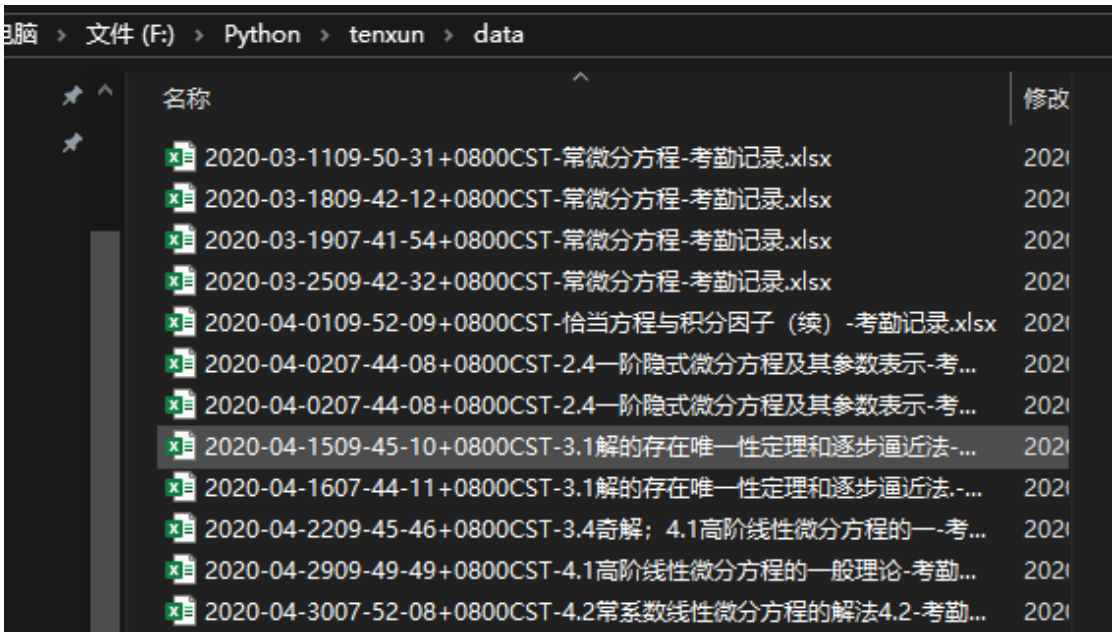
目录下就会生成一个文件:



三、说明

数据说明:

> *只要在 data 目录里面存放腾讯课堂数据文件即可对里面所有文件进行统计!!!*



注意：

每次生成两个数据文件之后，假如要更新，需要将两个数据文件给删除，如果不删除，每次执行之后会直接在两个文件后面追加数据!!!