

### 第三周学习报告

#### 1. 配置 conda init

```
cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ conda init bash
no change      D:\Anaconda\Scripts\conda.exe
no change      D:\Anaconda\Scripts\conda-env.exe
no change      D:\Anaconda\Scripts\conda-script.py
no change      D:\Anaconda\Scripts\conda-env-script.py
```

- 初始化 bash

```
$ .bash_profile X
$ .bash_profile
1
2 # >>> conda initialize >>>
3 # !! Contents within this block are managed by 'conda init' !!
4 if [ -f '/d/Anaconda/Scripts/conda.exe' ]; then
5     eval "$('/d/Anaconda/Scripts/conda.exe' 'shell.bash' 'hook')"
6 fi
7 # <<< conda initialize <<<
```

- .bash\_profile: 启动的时候会自动运行此文件夹中的命令

```
10 . ~/.config/git/git-prompt.sh
11 else
12     PS1='\[\033]0;$TITLEPREFIX:$PWD\007\]' # set window title
13     # PS1="$PS1"\n' # new line
14     PS1="$PS1"'\[\033[32m\]' # change to green
```

```
6 fi
7 # <<< conda initialize <<<
8
9 PS1="\n$PS1"
10 |
```

```
(base) cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ ls -l
total 32798
```

- 把 (base) 命令行放在用户名前面

#### 2. 使用 conda info 命令查看本机 Conda 的配置信息

```
(base) cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ conda info

active environment : base
active env location : D:\Anaconda
shell level : 1
user config file : C:\Users\cherry\.condarc
populated config files : D:\Anaconda\.condarc
conda version : 24.9.2
conda-build version : 24.9.0
```

3. 使用 `conda env list` 命令查看已有的 Conda 环境的名称和路径

```
(base) cherry@LAPTOP-QR2UKG4V MINGW64 ~  
$ conda env list  
# conda environments:  
#  
base * D:\Anaconda
```

- 已有环境的名称和路径
4. `conda create` 命令创建两个 Conda 环境，一个里面安装 Python 3.12 和 `requests` 软件包，另一个里面安装 Python 3.9、`pandas` 和 `statsmodels` 软件包，能够在终端里切换 Conda 环境，验证 Python 和软件包的版本

```
(base) cherry@LAPTOP-QR2UKG4V MINGW64 ~  
$ conda create -n prj1 python=3.12 requests
```

环境名称

安装包名字

```
(base) cherry@LAPTOP-QR2UKG4V MINGW64 ~  
$ conda create -n prj2 python=3.9 pandas statsmodels
```

```
(base) cherry@LAPTOP-QR2UKG4V MINGW64 ~  
$ conda activate prj1  
(prj1)  
cherry@LAPTOP-QR2UKG4V MINGW64 ~
```

- 激活 `prj1` 环境

```
cherry@LAPTOP-QR2UKG4V MINGW64 ~  
$ python  
Python 3.12.9 | packaged by Anaconda, Inc. | (main, Feb 6 2025, 18:49:16) [MSC v.1929 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license" for more information.  
>>>
```

Python 的提示符

切换 conda 环境

```
>>> quit()  
(prj1)  
cherry@LAPTOP-QR2UKG4V MINGW64 ~  
$ conda activate prj2  
(prj2)  
cherry@LAPTOP-QR2UKG4V MINGW64 ~  
$
```

```
>>> pandas.__version__
'2.2.3'
>>> import statsmodels
>>> statsmodels.__version__
'0.14.4'
```

- 验证软件包版本

5. 使用 conda list 命令显示 Conda 环境里的软件包列表及其版本信息

```
$ conda list
# packages in environment at D:\Anaconda\envs\prj2:
#
# Name                        Version                      Build      Channel
blas                          1.0                         mkl
bottleneck                    1.4.2                       py39hc99e966_0
```

6. 使用 conda install 命令往 Conda 环境里安装更多的软件包，并验证版本

```
cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ conda install ipython
Channels:
```

- 安装 ipython 安装包

7. 配置 Anaconda 清华镜像，加快 conda install 安装软件包的速度，将 conda-forge 设置为默认 Channel，让 conda install 能够安装更多的软件包

```
cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ conda config --set show_channel_urls yes
(prj1)
cherry@LAPTOP-QR2UKG4V MINGW64 ~
```

- 通过清华的镜像服务器的地址安装软件

```
.condarc
1  channels:
2    - defaults
3  show_channel_urls: true
4  default_channels:
5    - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
6    - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r
7    - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/msys2
8  custom_channels:
9    conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
10   pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
```

```
1  channels:
2    - conda-forge
3    - defaults
```

由 defaults 更改为 conda-forge（社区驱动的组织），为 conda 提供大量软件包（应用商店），现在 conda-forge 仓库里面找，找不到了再从 defaults 仓库里面找

```
cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ conda install polars
Channels:
- conda-forge
- defaults
- https://repo.anaconda.com/pkgs/main
- https://repo.anaconda.com/pkgs/r
- https://repo.anaconda.com/pkgs/msys2
```

- 安装 polars，优先从 conda-forge 应用商店中寻找软件包，可以安装成功

```
pip                25.0                py312haa95532_0    https://repo.ana
polars              1.24.0              pypi_0             pypi
```

是从 pypi 上安装的

- 如果下载软件包后运行不了，不要删掉软件包后重新下载，因为软件包之间会有依赖性，可能会造成更大的瘫痪，将环境移除后重新创建
8. 使用 pip install 命令往 Conda 环境里安装 Python 软件包，并验证版本
  9. 根据文档配置 PyPI 清华镜像，加快 pip install 安装软件包的速度

```
cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ pip config set global.index-url https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple
Writing to C:\Users\cherry\AppData\Roaming\pip\pip.ini
(prj1)
```

在清华的镜像网站中安装软件包，速度会更快

```
>>> import tushare
>>> tushare.__version__
'1.4.19'
>>> tushare.__file__
'D:\Anaconda\envs\prj1\Lib\site-packages\tushare\__init__.py'
```

- 在镜像网站中安装 tushare 软件包
10. 能够导出 environment.yml Conda 环境配置文件，能够删除 Conda 环境，能够用 environment.yml 配置文件重建 Conda 环境

```
cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ conda env export
name: prj1
channels:
- conda-forge
- defaults
- https://repo.anaconda.com/pkgs/main
- https://repo.anaconda.com/pkgs/r
- https://repo.anaconda.com/pkgs/msys2
```

- 删除 conda 环境

```
cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ conda deactivate
(base)
```

首先要退出激活状态



```
cherry@LAPTOP-QR2UKG4V MINGW64 ~  
$ conda env remove -n prj1  
Remove all packages in environment D:\Anaconda\envs\prj1:
```

软件包的名字要放在-n 后面

```
cherry@LAPTOP-QR2UKG4V MINGW64 ~/repo/prj1  
$ mv ~/environment.yml ./  
(base)  
cherry@LAPTOP-QR2UKG4V MINGW64 ~/repo/prj1
```

相对路径

- 在 repo 下创建一个文件夹 prj1，把导出来的环境放进去

```
cherry@LAPTOP-QR2UKG4V MINGW64 ~/repo/prj1  
$ ls -l  
total 4  
-rw-r--r-- 1 cherry 197121 2515 3月 21 16:18 environment.yml  
(base)  
cherry@LAPTOP-QR2UKG4V MINGW64 ~/repo/prj1  
$ conda env create
```

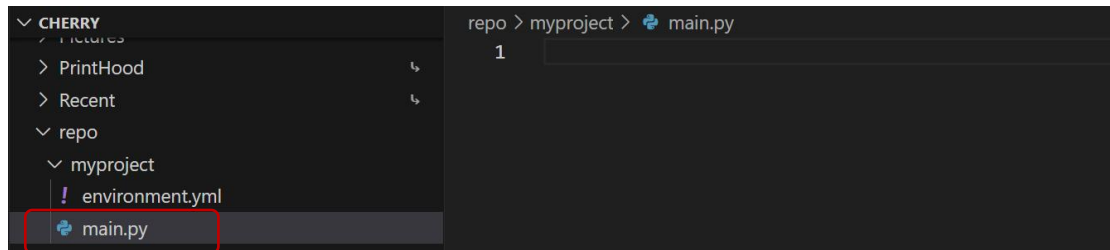
- 在此文件夹中重建 conda 环境，因为已经有导出来的环境，所以可以直接使用 conda env create 命令(在这一步中报错，可能是清华大学镜像网站高负载/正在维护导致响应时间过长无法连接)
11. 理解 Conda 与 Python 的关系，理解 Conda-Forge 与 Conda 的关系，理解 Python 解释器、第三方软件包、PyPI 软件仓库、以及程序/软件包的路径问题（使用 AI 大模型）
  12. 按照 [教程](#) 创建项目目录，在 VS Code 文本编辑器里安装一些支持 Python 开发的常用扩展，编写 main.py 脚本，创建该项目专用的 Conda 环境，在终端里激活该环境并成功运行该脚本

```
repo > myproject > ! environment.yml  
1 name: myproject  
2 channels:  
3 | - conda-forge  
4 dependencies:  
5 | - python=3.12
```

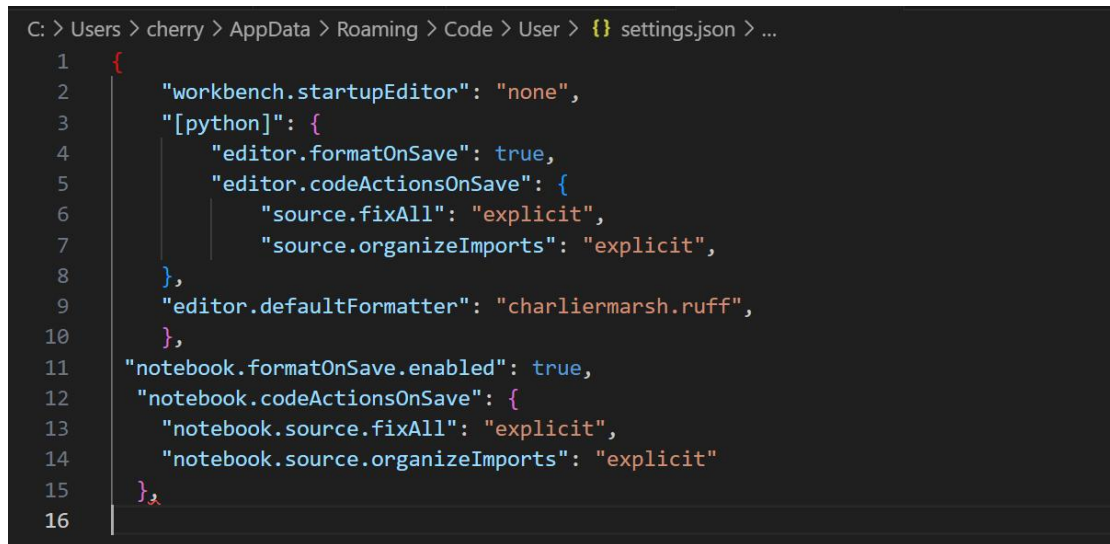
```
(base) cherry@LAPTOP-QR2UKG4V MINGW64 ~/repo/myproject  
$ ls -l  
total 1  
-rw-r--r-- 1 cherry 197121 75 3月 21 19:54 environment.yml  
(base) cherry@LAPTOP-QR2UKG4V MINGW64 ~/repo/myproject  
$ conda env create
```

- 创建新 conda 环境

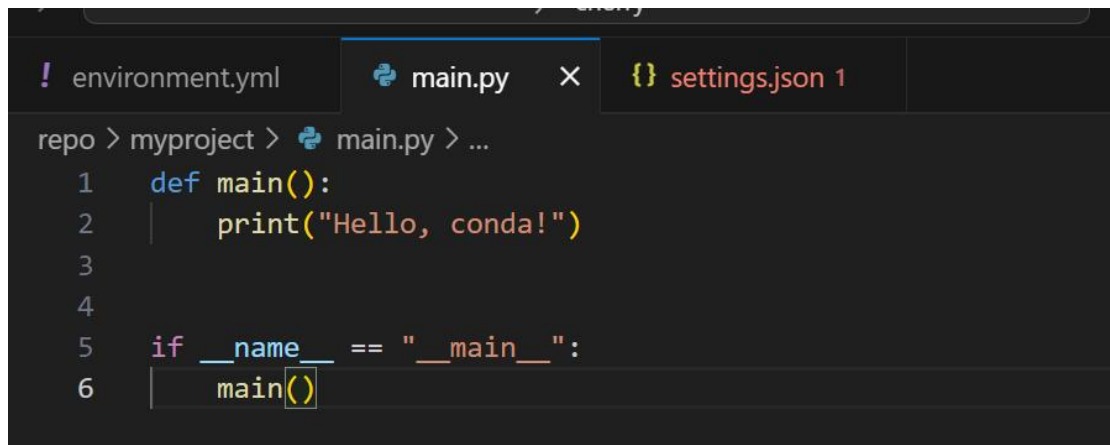
接下来创建新项目



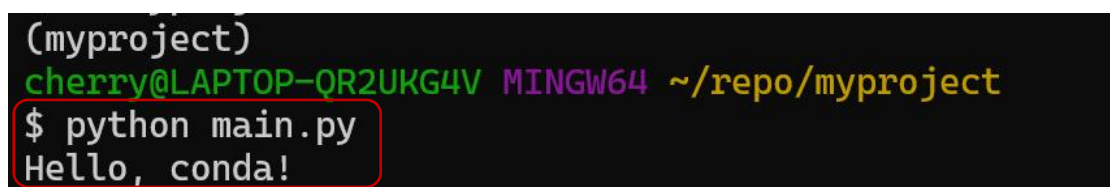
新建一个 main.py 用于写 python 代码



- 对 ruff 进行一些偏好设置



- 编写 main.py 脚本



- 在终端中运行脚本

```
! environment.yml X main.py
repo > myproject > ! environment.yml
1 name: myproject
2 channels:
3   - conda-forge
4 dependencies:
5   - python=3.12
6   - pandas
```

- 在 environment.yml 中编写命令在终端中运行并安装 pandas

```
repo > myproject > main.py > main
1 import pandas as pd
2
3
4 def main():
5     print("Hello, conda!")
6     print(pd.__version__)
7     print(pd.__file__)
8
9
10 if __name__ == "__main__":
11     main()
```

```
(myproject)
cherry@LAPTOP-QR2UKG4V MINGW64 ~/repo/myproject
$ python main.py
Hello, conda!
2.2.3
D:\Anaconda\envs\myproject\Lib\site-packages\pandas\__init__.py
```

- 在 vs code 中编写脚本并在终端中运行

```
$ curl -O https://edg.epa.gov/EPADDataCommons/public/OA/EPA_SmartLocationDatabase_V3_Jan_2021_Final.csv
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 192M 0 1904k 0 0 18609 0 3:00:31 0:01:44 2:58:47 15666
```

- 将这个网址的文件下载到此文件夹中

```

1 import pandas as pd
2
3
4 def main():
5     """
6     Answers the question:
7
8     What percentage of U.S. residents live highly walkable neighborhoods?
9
10    "15.26" is the threshold on the index for a highly walkable area.
11    """
12    csv_file = "./EPA_SmartLocationDatabase_V3_Jan_2021_Final.csv"
13    highly_walkable = 15.26
14
15    df = pd.read_csv(csv_file)
16
17    total_population = df["TotPop"].sum()
18    highly_walkable_pop = df[df["NatWalkInd"] >= highly_walkable]["TotPop"].sum()
19
20    percentage = (highly_walkable_pop / total_population) * 100.0
21
22    print(f"{percentage:.2f}% of U.S. residents live in highlywalkable neighborhoods.")
23
24
25 if __name__ == "__main__":
26     main()

```

- 在 vs code 中编写脚本，用于计算美国居民居住在高度适宜步行的区域

```

$ python main.py
12.64% of U.S. residents live in highlywalkable neighborhoods.
(myproject)

```

- 在终端中的运行结果