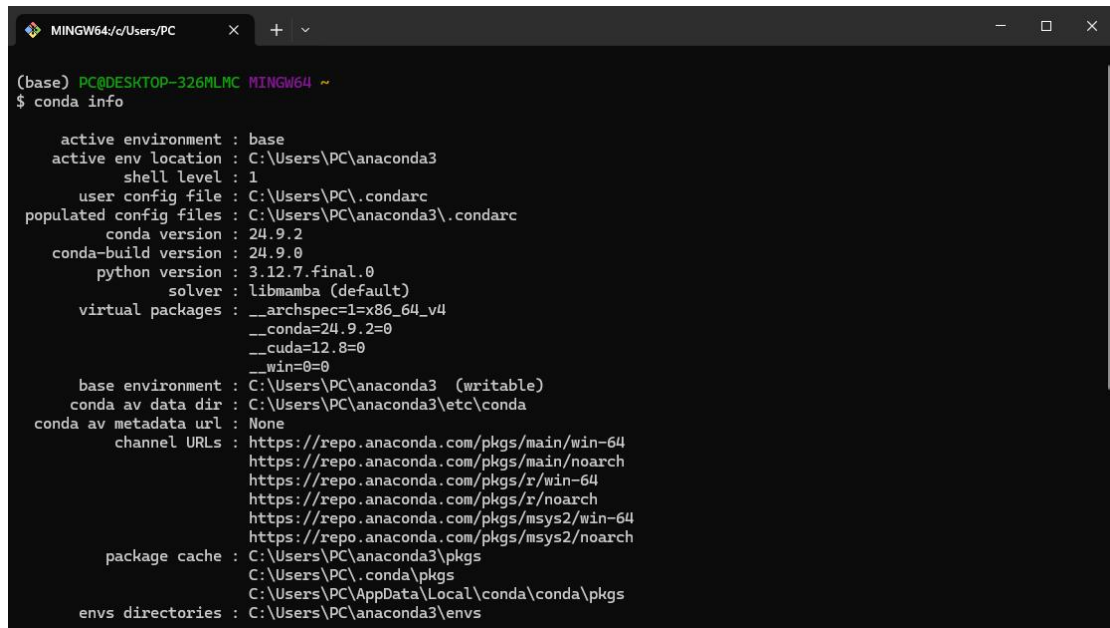


1. 使用 `conda info` 命令查看电脑 Conda 的配置信息。



```
(base) PC@DESKTOP-326MLMC MINGW64 ~
$ conda info

active environment : base
active env location : C:\Users\PC\anaconda3
shell level        : 1
user config file    : C:\Users\PC\.condarc
populated config files : C:\Users\PC\anaconda3\.condarc
conda version       : 24.9.2
conda-build version : 24.9.0
python version      : 3.12.7.final.0
solver              : libmamba (default)
virtual packages    : __archspec=1=x86_64_v4
                    __conda=24.9.2=0
                    __cuda=12.8=0
                    __win=0=0

base environment    : C:\Users\PC\anaconda3 (writable)
conda av data dir   : C:\Users\PC\anaconda3\etc\conda
conda av metadata url : None
channel URLs        : https://repo.anaconda.com/pkg/main/win-64
                    https://repo.anaconda.com/pkg/main/noarch
                    https://repo.anaconda.com/pkg/r/win-64
                    https://repo.anaconda.com/pkg/r/noarch
                    https://repo.anaconda.com/pkg/msys2/win-64
                    https://repo.anaconda.com/pkg/msys2/noarch
package cache       : C:\Users\PC\anaconda3\pkgs
                    C:\Users\PC\.conda\pkgs
                    C:\Users\PC\AppData\Local\conda\conda\pkgs
envs directories    : C:\Users\PC\anaconda3\envs
```

2. 使用 `conda create` 命令创建两个 Conda 环境，安装 Python 3.12 和 `requests` 软件包到 `proj1` 文件夹，而 `proj2` 文件夹里面安装 Python 3.9、`pandas` 和 `statsmodels` 软件包。

关于 Conda 环境概念的理解

(1) 依赖隔离

每个环境可以安装独立的 Python 版本、库及其依赖项，不同环境之间互不干扰。例如：项目 A 需要 Python 3.8 和 Pandas 1.2.0，项目 B 需要 Python 3.9 和 Pandas 2.0.0，两个项目可分别运行在不同的 Conda 环境中，互不影响。

(2) 跨平台兼容性

Conda 环境可以在 Windows、macOS、Linux 等系统上保持一致，确保代码在不同环境中的可移植性。

(3) 环境共享

可以通过导出环境配置文件（如 `environment.yml`），快速复现环境，方便协作和部署。

```
MINGW64/c/Users/PC x + v
(base) PC@DESKTOP-326MLMC MINGW64 ~
$ conda env list
# conda environments:
#
base * C:\Users\PC\anaconda3

(base) PC@DESKTOP-326MLMC MINGW64 ~
$ conda create -n proj1 python=3.12 requests
Channels:
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\PC\anaconda3\envs\proj1

added / updated specs:
- python=3.12
- requests

The following packages will be downloaded:

package | build
-----|-----
brotli-python-1.0.9 | py312h5da7b33_9 347 KB
```

```
MINGW64/c/Users/PC x + v
(base) PC@DESKTOP-326MLMC MINGW64 ~
$ conda create -n proj2 python=3.9 pandas statsmodels
Channels:
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\PC\anaconda3\envs\proj2

added / updated specs:
- pandas
- python=3.9
- statsmodels

The following packages will be downloaded:

package | build
-----|-----
blas-1.0 | mkl 6 KB
bottleneck-1.4.2 | py39hc99e966_0 129 KB
icc_rt-2022.1.0 | h6049295_2 6.5 MB
intel-openmp-2023.1.0 | h59b6b97_46320 2.7 MB
mkl-2023.1.0 | h6b88ed4_46358 155.9 MB
mkl-service-2.4.0 | py39h827c3e9_2 66 KB
mkl_fft-1.3.11 | py39h827c3e9_0 167 KB
mkl_random-1.2.8 | py39hc64d2fc_0 257 KB
```

3. 使用 `conda env list` 命令查看已安装的 Conda 环境的名称和路径，并在终端里切换 Conda 环境，验证 Python 和软件包的版本。

```
MINGW64/c/Users/PC
$ conda env list
# conda environments:
#
base                  * C:\Users\PC\anaconda3
proj1                  C:\Users\PC\anaconda3\envs\proj1
proj2                  C:\Users\PC\anaconda3\envs\proj2

(base) PC@DESKTOP-326MLMC MINGW64 ~
$ conda activate proj1
(proj1)
PC@DESKTOP-326MLMC MINGW64 ~
$ which python
/c/Users/PC/anaconda3/envs/proj1/python
(proj1)
PC@DESKTOP-326MLMC MINGW64 ~
$ python --version
Python 3.12.9
(proj1)
PC@DESKTOP-326MLMC 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.
>>> import requests
>>> requests.__file__
'C:\Users\PC\anaconda3\envs\proj1\Lib\site-packages\requests\__init__.py'
>>> quit()
(proj1)
```

```
MINGW64/c/Users/PC
'C:\Users\PC\anaconda3\envs\proj1\Lib\site-packages\requests\__init__.py'
>>> quit()
(proj1)
PC@DESKTOP-326MLMC MINGW64 ~
$ conda activate proj2
(proj2)
PC@DESKTOP-326MLMC MINGW64 ~
$ which python
/c/Users/PC/anaconda3/envs/proj2/python
(proj2)
PC@DESKTOP-326MLMC MINGW64 ~
$ python --version
Python 3.9.21
(proj2)
PC@DESKTOP-326MLMC MINGW64 ~
$ python
Python 3.9.21 (main, Dec 11 2024, 16:35:24) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import pandas
>>> pandas.__file__
'C:\Users\PC\anaconda3\envs\proj2\lib\site-packages\pandas\__init__.py'
>>> pandas.__version__
'2.2.3'
>>> import statsmodels
>>> statsmodels.__version__
'0.14.4'
>>> quit()
(proj2)
PC@DESKTOP-326MLMC MINGW64 ~
$
```

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

```
MINGW64/c/Users/PC x + v
(proj2)
PC@DESKTOP-326MLMC MINGW64 ~
$ conda list
# packages in environment at C:\Users\PC\anaconda3\envs\proj2:
#
# Name                      Version          Build          Channel
blas                        1.0              mkl
bottleneck                 1.4.2            py39hc99e966_0
ca-certificates            2025.2.25        haa95532_0
icc_rt                     2022.1.0         h6049295_2
intel-openmp               2023.1.0         h59b6b97_46320
mkl                        2023.1.0         h6b88ed4_46358
mkl-service                2.4.0            py39h827c3e9_2
mkl_fft                   1.3.11           py39h827c3e9_0
mkl_random                 1.2.8            py39hc64d2fc_0
numexpr                    2.10.1           py39h4cd664f_0
numpy                      2.0.2            py39h055cbcc_0
numpy-base                 2.0.2            py39h65a83cf_0
openssl                    3.0.16           h3f729d1_0
packaging                  24.2             py39haa95532_0
pandas                     2.2.3            py39h5da7b33_0
patsy                      1.0.1            py39haa95532_0
pip                        25.0             py39haa95532_0
pybind11-abi               5                hd3eb1b0_0
python                     3.9.21           h8205438_1
python-dateutil            2.9.0post0       py39haa95532_2
python-tzdata              2023.3           pyhd3eb1b0_0
pytz                       2024.1           py39haa95532_0
scipy                      1.13.1           py39h8640f81_1
setuptools                 72.1.0           py39haa95532_0
```

```
MINGW64/c/Users/PC x + v
(proj1)
PC@DESKTOP-326MLMC MINGW64 ~
$ conda list
# packages in environment at C:\Users\PC\anaconda3\envs\proj1:
#
# Name                      Version          Build          Channel
brotli-python              1.0.9            py312h5da7b33_9
bzip2                      1.0.8            h2bbff1b_6
ca-certificates            2025.2.25        haa95532_0
certifi                    2025.1.31        py312haa95532_0
charset-normalizer         3.3.2            pyhd3eb1b0_0
expat                      2.6.4            h8ddb27b_0
idna                       3.7              py312haa95532_0
libffi                     3.4.4            hd77b12b_1
openssl                    3.0.16           h3f729d1_0
pip                        25.0             py312haa95532_0
pysocks                    1.7.1            py312haa95532_0
python                     3.12.9           h14ffc60_0
requests                   2.32.3           py312haa95532_1
setuptools                 75.8.0           py312haa95532_0
sqlite                     3.45.3           h2bbff1b_0
tk                          8.6.14           h0416ee5_0
tzdata                     2025a            h04d1e81_0
urllib3                    2.3.0            py312haa95532_0
vc                          14.42            haa95532_4
vs2015_runtime             14.42.34433      he0abc0d_4
wheel                      0.45.1           py312haa95532_0
win_inet_pton              1.1.0            py312haa95532_0
xz                          5.6.4            h4754444_1
zlib                       1.2.13           h8cc25b3_1
```

5. 使用 `conda install` 命令往 Conda 环境里安装 `ipython` 软件包，并验证版本。

```
MINGW64/c/Users/PC
(base) PC@DESKTOP-326MLMC MINGW64 ~
$ conda activate proj1
(proj1)
PC@DESKTOP-326MLMC MINGW64 ~
$ conda install ipython
Channels:
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\PC\anaconda3\envs\proj1

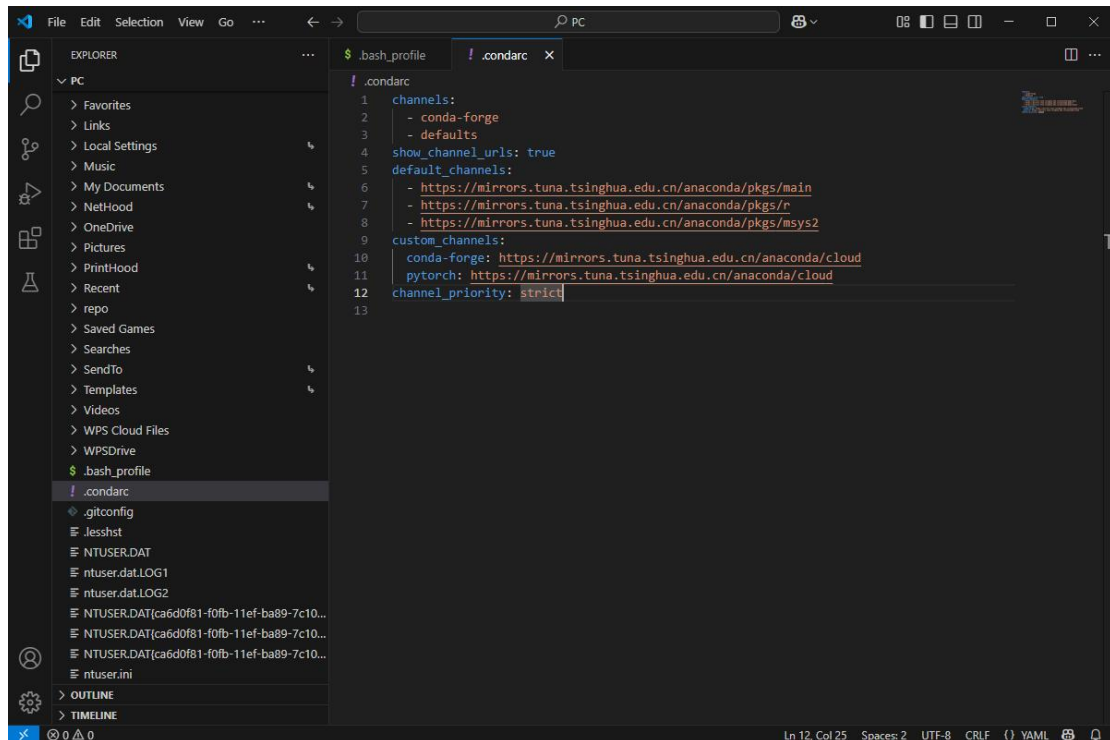
added / updated specs:
- ipython

The following packages will be downloaded:
```

package	build	
asttokens-2.0.5	pyhd3eb1b0_0	20 KB
colorama-0.4.6	py312haa95532_0	53 KB
decorator-5.1.1	pyhd3eb1b0_0	12 KB
executing-0.8.3	pyhd3eb1b0_0	18 KB
ipython-8.30.0	py312haa95532_0	1.5 MB
jedi-0.19.2	py312haa95532_0	1.2 MB
matplotlib-inline-0.1.6	py312haa95532_0	19 KB

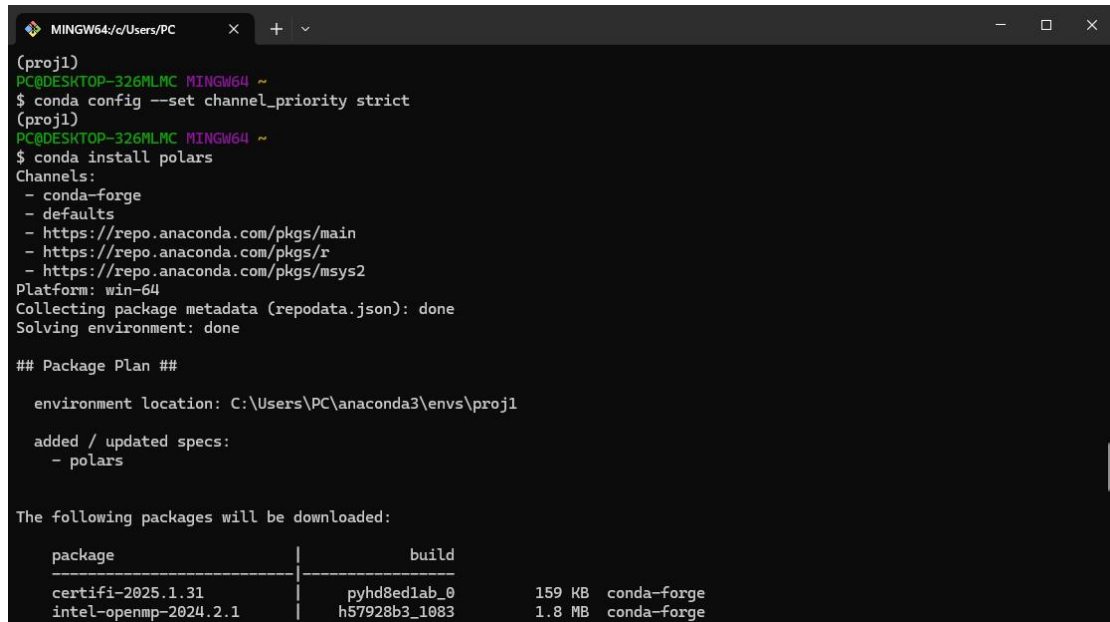
```
MINGW64/c/Users/PC
PC@DESKTOP-326MLMC MINGW64 ~
$ conda list
# packages in environment at C:\Users\PC\anaconda3\envs\proj1:
#
# Name          Version          Build          Channel
asttokens       2.0.5            pyhd3eb1b0_0
brotli-python   1.0.9            py312h5da7b33_9
bzip2           1.0.8            h2bbff1b_6
ca-certificates 2025.2.25        haa95532_0
certifi         2025.1.31        py312haa95532_0
charset-normalizer 3.3.2          pyhd3eb1b0_0
colorama        0.4.6            py312haa95532_0
decorator       5.1.1            pyhd3eb1b0_0
executing       0.8.3            pyhd3eb1b0_0
expat           2.6.4            h8ddb27b_0
idna            3.7              py312haa95532_0
ipython         8.30.0           py312haa95532_0
jedi            0.19.2           py312haa95532_0
libffi          3.4.4            hd77b12b_1
matplotlib-inline 0.1.6          py312haa95532_0
openssl         3.0.16           h3f729d1_0
parso           0.8.4            py312haa95532_0
pip             25.0            py312haa95532_0
prompt-toolkit  3.0.43           py312haa95532_0
prompt_toolkit  3.0.43           hd3eb1b0_0
pure_eval       0.2.2            pyhd3eb1b0_0
pygments        2.15.1           py312haa95532_1
pysocks         1.7.1            py312haa95532_0
python          3.12.9           h14ffc60_0
requests        2.32.3           py312haa95532_1
```

6. 配置 Anaconda 清华镜像，将 conda-forge 设置为默认 Channel，让 conda install 能够安装 polars 软件包。



The screenshot shows the VS Code editor with the `.condarc` file open. The file is configured to use the conda-forge channel as the default and to use the Tsinghua University mirrors for Anaconda packages. The configuration includes default channels, custom channels for conda-forge and pytorch, and a strict channel priority.

```
1 channels:
2   - conda-forge
3   - defaults
4 show_channel_urls: true
5 default_channels:
6   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/main
7   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/r
8   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/msys2
9 custom_channels:
10  conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
11  pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
12 channel_priority: strict
13
```



The screenshot shows a Windows command prompt window with the following commands and output:

```
(proj1)
PC@DESKTOP-326MLMC MINGW64 ~
$ conda config --set channel_priority strict
(proj1)
PC@DESKTOP-326MLMC MINGW64 ~
$ conda install polars
Channels:
- conda-forge
- defaults
- https://repo.anaconda.com/pkg/main
- https://repo.anaconda.com/pkg/r
- https://repo.anaconda.com/pkg/msys2
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\PC\anaconda3\envs\proj1

added / updated specs:
- polars

The following packages will be downloaded:
```

package	build	size	channel
certifi-2025.1.31	pyhd8ed1ab_0	159 KB	conda-forge
intel-openmp-2024.2.1	h57928b3_1083	1.8 MB	conda-forge

7. 使用 `pip install` 命令往 Conda 环境里安装 tushare 软件包，升级 pip 到最新的版本后配置配置 PyPI 清华镜像，可加快 `pip install` 安装软件包的速度。

```
MINGW64/c/Users/PC (proj1)
PC@DESKTOP-326MLMC MINGW64 ~
$ pip install tushare
Collecting tushare
  Downloading tushare-1.4.19-py3-none-any.whl.metadata (3.1 kB)
Collecting pandas (from tushare)
  Downloading pandas-2.2.3-cp312-cp312-win_amd64.whl.metadata (19 kB)
Requirement already satisfied: requests in c:\users\pc\anaconda3\envs\proj1\lib\site-packages (from tushare) (2.32.3)
Collecting lxml (from tushare)
  Downloading lxml-5.3.1-cp312-cp312-win_amd64.whl.metadata (3.8 kB)
Collecting simplejson (from tushare)
  Downloading simplejson-3.20.1-cp312-cp312-win_amd64.whl.metadata (3.4 kB)
Collecting bs4 (from tushare)
  Downloading bs4-0.0.2-py2.py3-none-any.whl.metadata (411 bytes)
Collecting websocket-client>=0.57.0 (from tushare)
  Downloading websocket_client-1.8.0-py3-none-any.whl.metadata (8.0 kB)
Collecting tqdm (from tushare)
  Downloading tqdm-4.67.1-py3-none-any.whl.metadata (57 kB)
Collecting beautifulsoup4 (from bs4->tushare)
  Downloading beautifulsoup4-4.13.3-py3-none-any.whl.metadata (3.8 kB)
Requirement already satisfied: numpy>=1.26.0 in c:\users\pc\anaconda3\envs\proj1\lib\site-packages (from pandas->tushare) (2.2.4)
Collecting python-dateutil>=2.8.2 (from pandas->tushare)
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pytz>=2020.1 (from pandas->tushare)
  Downloading pytz-2025.1-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas->tushare)
  Downloading tzdata-2025.1-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\pc\anaconda3\envs\proj1\lib\site-packages (from requests->tushare) (3.3.2)
```

```
MINGW64/c/Users/PC (proj1)
PC@DESKTOP-326MLMC MINGW64 ~
$ python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\pc\anaconda3\envs\proj1\lib\site-packages (25.0)
Collecting pip
  Downloading pip-25.0.1-py3-none-any.whl.metadata (3.7 kB)
  Downloading pip-25.0.1-py3-none-any.whl (1.8 MB)
    1.8/1.8 MB 4.8 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 25.0
    Uninstalling pip-25.0:
      Successfully uninstalled pip-25.0
  Successfully installed pip-25.0.1
(proj1)
PC@DESKTOP-326MLMC MINGW64 ~
$ pip config set global.index-url https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple
Writing to C:\Users\PC\AppData\Roaming\pip\pip.ini
(proj1)
PC@DESKTOP-326MLMC MINGW64 ~
$ conda list
# packages in environment at C:\Users\PC\anaconda3\envs\proj1:
#
# Name                    Version            Build                Channel
asttokens                 2.0.5              pyhd3eb1b0_0        https://repo.anaconda.com/pkgs/main
beautifulsoup4            4.13.3             pypi_0              pypi
brotli-python             1.0.9              py312h5da7b33_9     https://repo.anaconda.com/pkgs/main
bs4                       0.0.2              pypi_0              pypi
bzip2                     1.0.8              h2bbff1b_6          https://repo.anaconda.com/pkgs/main
ca-certificates           2025.2.25          haa95532_0          https://repo.anaconda.com/pkgs/main
```

8. 导出名为 `environment.yml` 的 Conda 环境配置文件，删除环境 `proj1`，利用 `environment.yml` 配置文件在文件夹 `repo` 里重建环境 `proj1`。

```
(proj1)
PC@DESKTOP-326MLMC MINGW64 ~
$ conda env export -f environment.yml
(proj1)
PC@DESKTOP-326MLMC MINGW64 ~
$ conda deactivate
(base)
PC@DESKTOP-326MLMC MINGW64 ~
```

```
(base)
PC@DESKTOP-326MLMC MINGW64 ~
$ conda env list
# conda environments:
#
base                * C:\Users\PC\anaconda3
proj1                C:\Users\PC\anaconda3\envs\proj1
proj2                C:\Users\PC\anaconda3\envs\proj2

(base)
PC@DESKTOP-326MLMC MINGW64 ~
$ conda env remove -n proj1

Remove all packages in environment C:\Users\PC\anaconda3\envs\proj1:

## Package Plan ##

  environment location: C:\Users\PC\anaconda3\envs\proj1

The following packages will be REMOVED:
```

```
MINGW64/c/Users/PC/repo/t  x  +  v  -  □  x

(base)
PC@DESKTOP-326MLMC MINGW64 ~
$ conda env list
# conda environments:
#
base                * C:\Users\PC\anaconda3
proj2                C:\Users\PC\anaconda3\envs\proj2

(base)
PC@DESKTOP-326MLMC MINGW64 ~
$ cd repo
(base)
PC@DESKTOP-326MLMC MINGW64 ~/repo
$ mkdir proj1
(base)
PC@DESKTOP-326MLMC MINGW64 ~/repo
$ cd proj1
(base)
PC@DESKTOP-326MLMC MINGW64 ~/repo/proj1
$ mv ~/environment.yml ./
(base)
PC@DESKTOP-326MLMC MINGW64 ~/repo/proj1
$ conda env create
C:\Users\PC\anaconda3\Lib\argparse.py:2006: FutureWarning: `remote_definition` is deprecated and will be removed in 25.9
. Use `conda env create --file=URL` instead.
  action(self, namespace, argument_values, option_string)
Warning: you have pip-installed dependencies in your environment file, but you do not list pip itself as one of your con
da dependencies. Conda may not use the correct pip to install your packages, and they may end up in the wrong place. P
lease add an explicit pip dependency. I'm adding one for you, but still nagging you.
```


9. Conda 与 Python 的关系，理解 Conda-forge 与 Conda 的关系

(1) Conda 与 Python 的关系

Conda 是一个跨平台、开源的包管理系统和环境管理系统，而 Python 是一种高级编程语言。

Conda 可以用来管理 Python 软件包，你能够借助 Conda 安装、更新和卸载 Python 软件包。

Conda 能够创建独立的 Python 环境。不同的项目可能需要不同版本的 Python 或者不同的 Python 包，Conda 可以为每个项目创建独立的环境，从而避免包版本冲突。

(2) Conda-forge 与 Conda 的关系

Conda-forge 是一个由社区驱动的项目，它为 Conda 提供了大量的软件包。Conda 本身支持从多个包源（channel）下载和安装包，Conda-forge 就是其中一个重要的包源。Conda 默认的包源可能没有包含所有你需要的包，而 Conda-forge 社区维护了大量的开源软件包，涵盖了科学计算、机器学习、数据处理等多个领域。

Conda-forge 是由社区驱动的，社区成员会不断地更新和维护包。这意味着 Conda-forge 上的包通常能够及时更新到最新版本，并且会修复已知的问题。相比之下，Conda 默认的包源可能更新速度较慢。

Conda-forge 提供了比 Conda 默认包源更多样化的包。无论你是在进行数据分析、机器学习还是其他领域的开发，都可能在 Conda-forge 上找到你需要的包。例如，一些流行的机器学习库如 scikit-learn、pytorch 等，在 Conda-forge 上都有很好的支持。

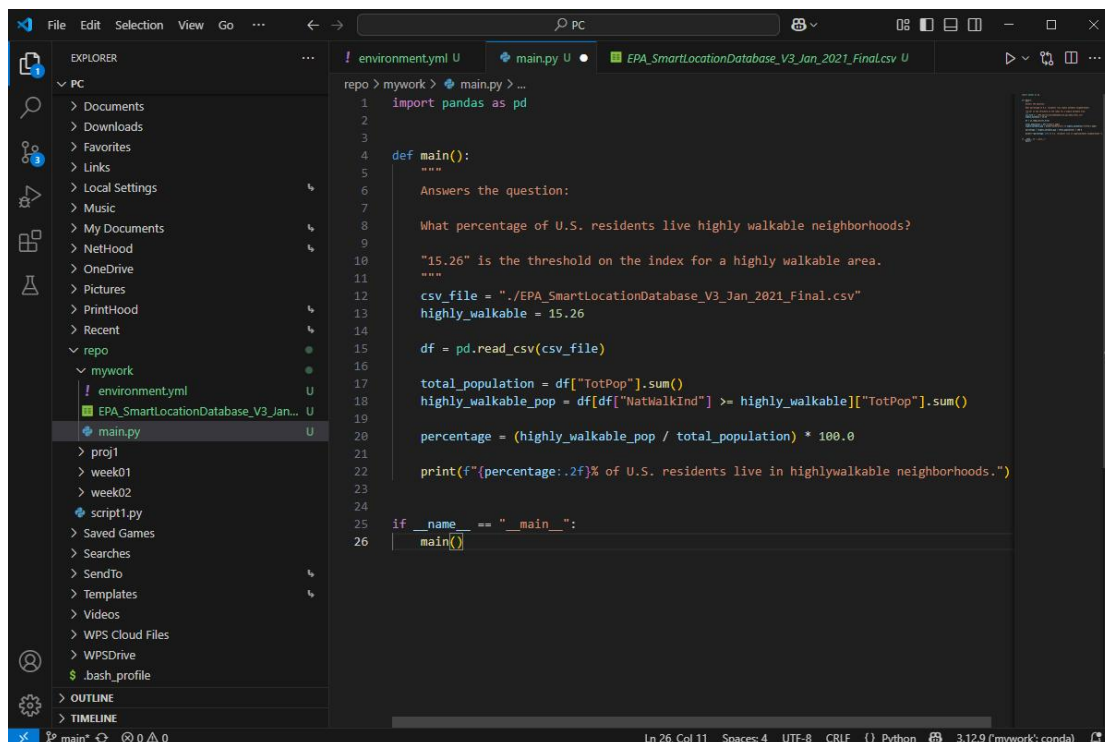
10. 创建 mywork 项目，在 VS Code 文本编辑器里安装常用扩展 Ruff, 以支持 Python 开发，编写 main.py 脚本，创建该项目专用的 environmental.yml 环境，在终端里激活该环境并成功运行该脚本。

```
MINGW64/c/Users/PC/repo/r x + -
(base) PC@DESKTOP-326MLMC MINGW64 ~/repo
$ cd mywork

(base) PC@DESKTOP-326MLMC MINGW64 ~/repo/mywork (main)
$ ls -l
total 1
-rw-r--r-- 1 PC 197121 72 3月 23 13:00 environment.yml

(base) PC@DESKTOP-326MLMC MINGW64 ~/repo/mywork (main)
$ cat environment.yml
name: mywork
channels:
  - conda-forge
done
#
# To activate this environment, use
#
#     $ conda activate mywork
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) PC@DESKTOP-326MLMC MINGW64 ~/repo/mywork (main)
$ conda env list
# conda environments:
#
base                  * C:\Users\PC\anaconda3
mywork                C:\Users\PC\anaconda3\envs\mywork
```



The screenshot shows the VS Code interface with the Explorer view on the left displaying the project structure. The main editor shows the main.py file with the following code:

```
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()
```

```
MINGW64/c/Users/PC/repo/r x + v
wheel 0.45.1 pyhd8ed1ab_1 conda-forge
(mywork)
PC@DESKTOP-326MLMC MINGW64 ~/repo/mywork (main)
$ python main.py
Hello, conda!
(mywork)
PC@DESKTOP-326MLMC MINGW64 ~/repo/mywork (main)
$ python main.py
Hello, conda!
2.2.3
C:\Users\PC\anaconda3\envs\mywork\Lib\site-packages\pandas\__init__.py
(mywork)
PC@DESKTOP-326MLMC MINGW64 ~/repo/mywork (main)
$ ls -l
total 2
-rw-r--r-- 1 PC 197121 84 3月 23 13:36 environment.yml
-rw-r--r-- 1 PC 197121 161 3月 23 13:42 main.py
(mywork)
PC@DESKTOP-326MLMC MINGW64 ~/repo/mywork (main)
$ curl -O https://edg.epa.gov/EPADataCommons/public/OA/EPA_SmartLocationDatabase_V3_Jan_2021_Final.csv
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 192M 100 192M 0 0 131k 0 0:25:01 0:25:01 --:--:-- 222k
(mywork)
PC@DESKTOP-326MLMC MINGW64 ~/repo/mywork (main)
$ python main.py
10.69% of U.S. residents live in highlywalkable neighborhoods.
(mywork)
PC@DESKTOP-326MLMC MINGW64 ~/repo/mywork (main)
$
```