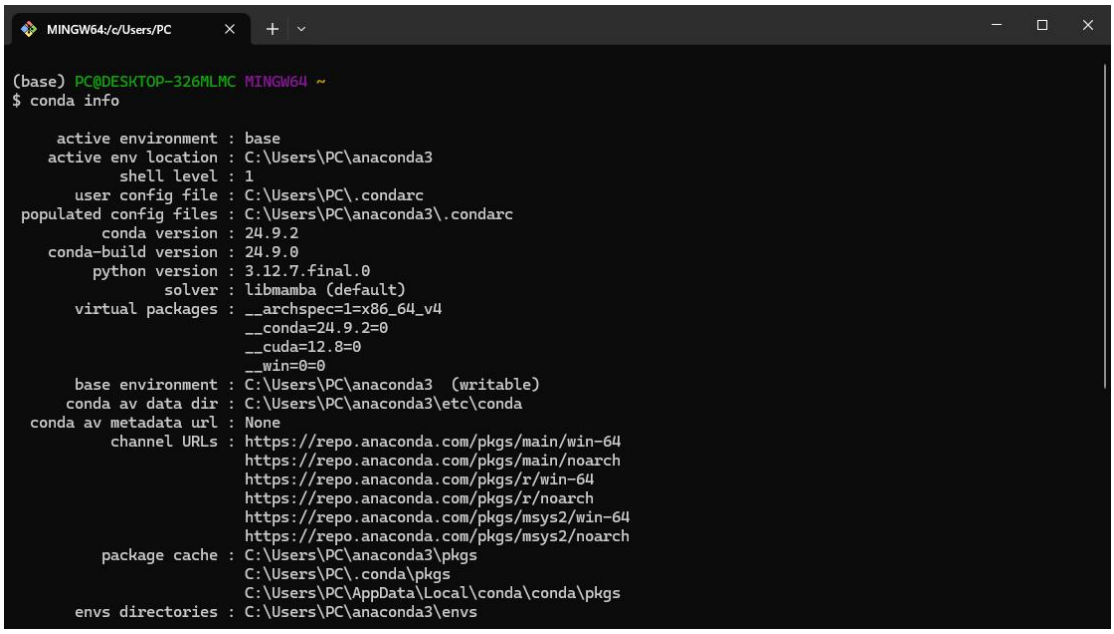
1. 使用 conda info 命令查看电脑 Conda 的配置信息。



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

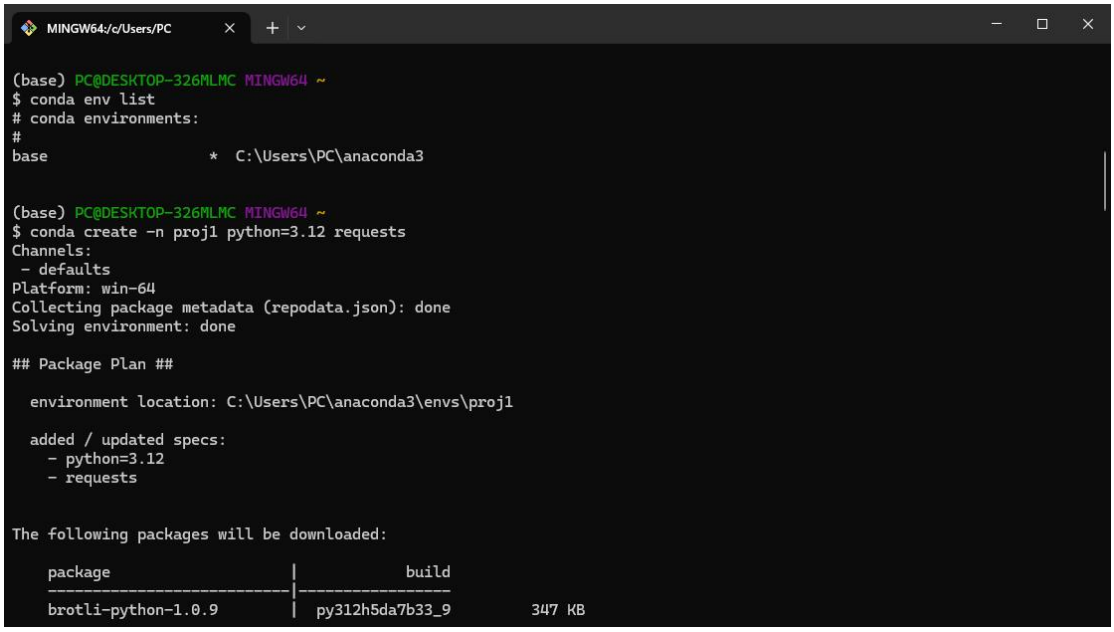
每个环境可以安装独立的 Python 版本、库及其依赖项，不同环境之间互不干扰。例如：项目 A 需要 Python 3.8 和 Pandas 1.2.0，项目 B 需要 Python 3.9 和 Pandas 2.0.0，两个项目可分别运行在不同的 Conda 环境中，互不影响。

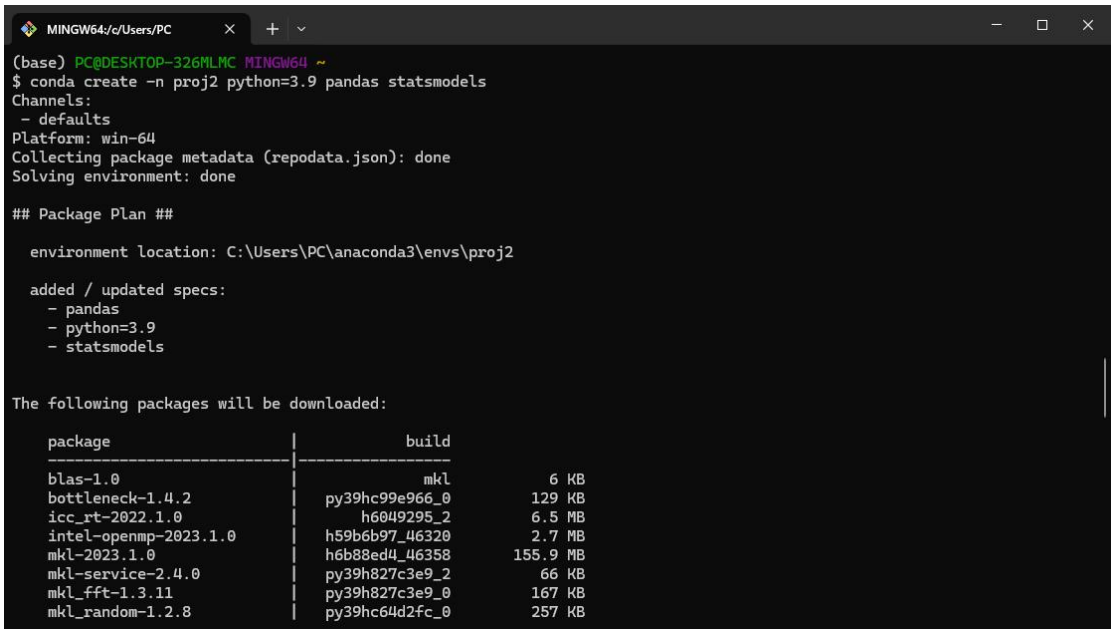
（2）跨平台兼容性

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。

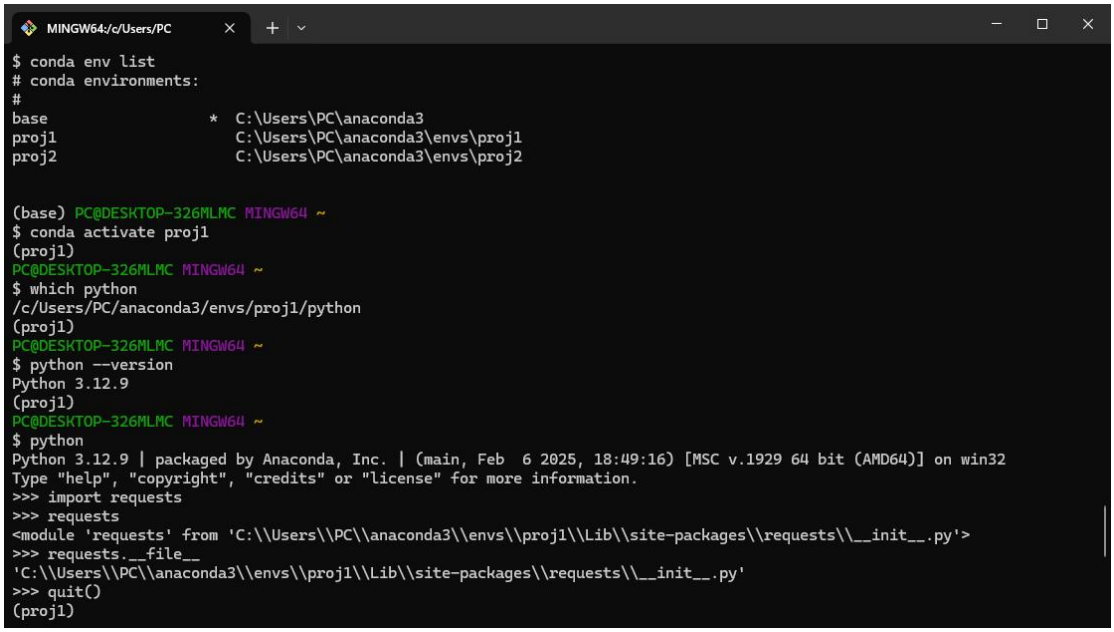
（3）环境共享

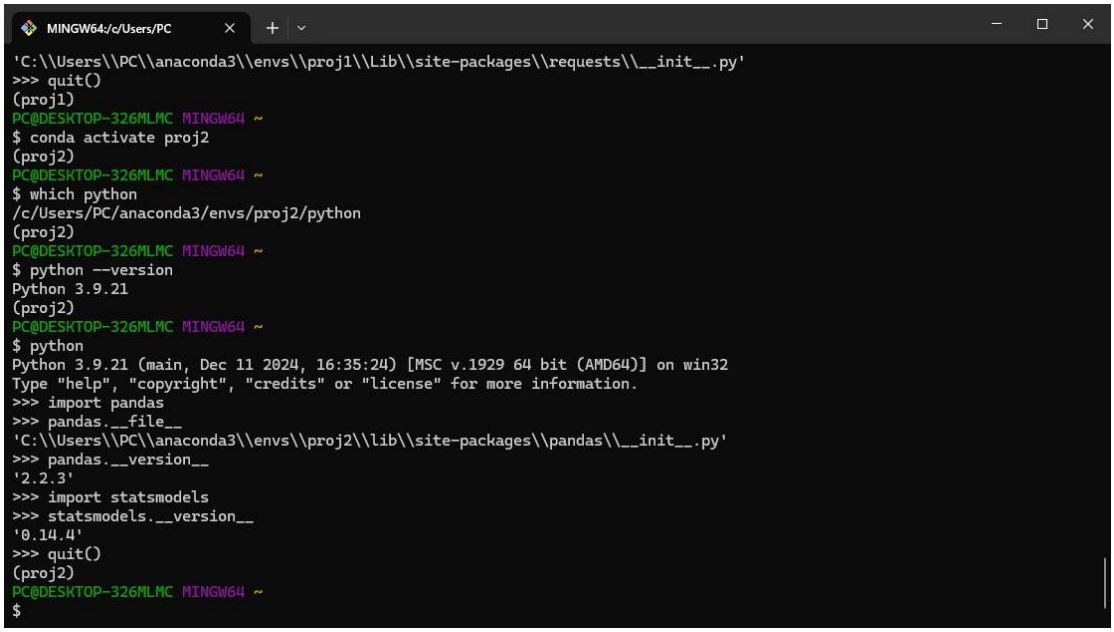
可以通过导出环境配置文件（如 environment.yml），快速复现环境，方便协作和部署。


MINGW64:/c/Users/PC
(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

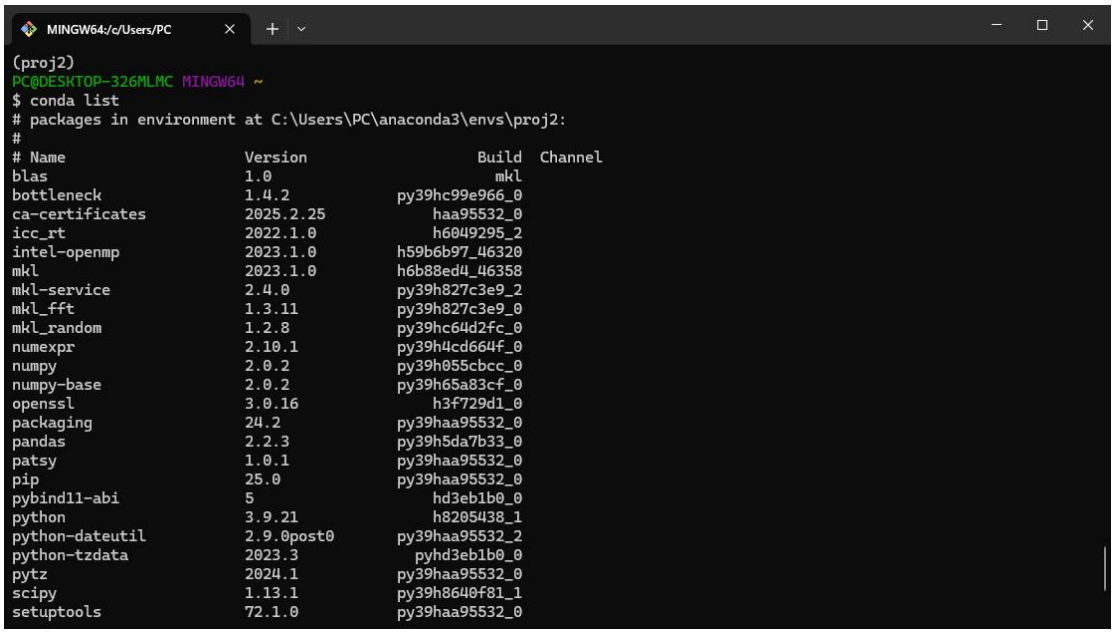

MINGW64:/c/Users/PC
(base) PC@DESKTOP-326MLMC MINGW64 ~
$ conda create -n proj2 python=3.9 pandas statsmodels
Channels:
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done
Package Plan
environment location: C:\Users\PC\anaconda3\envs\proj2
added / 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

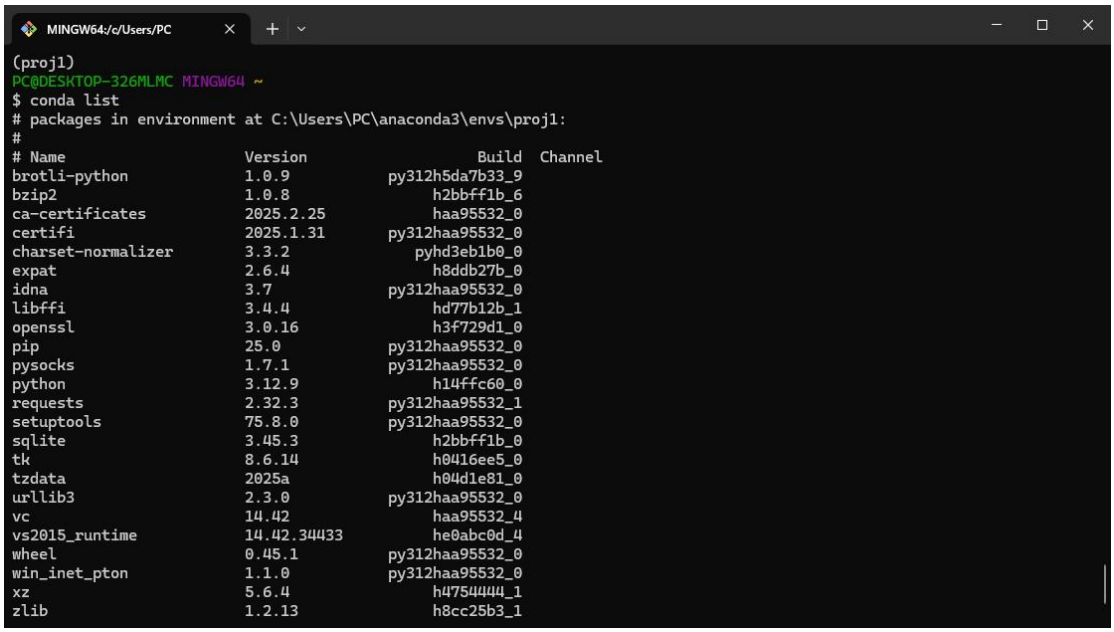
3. 使用 conda env list 命令查看已安装的 Conda 环境的名称和路径，并在终端里切换 Conda 环境，验证 Python 和软件包的版本。



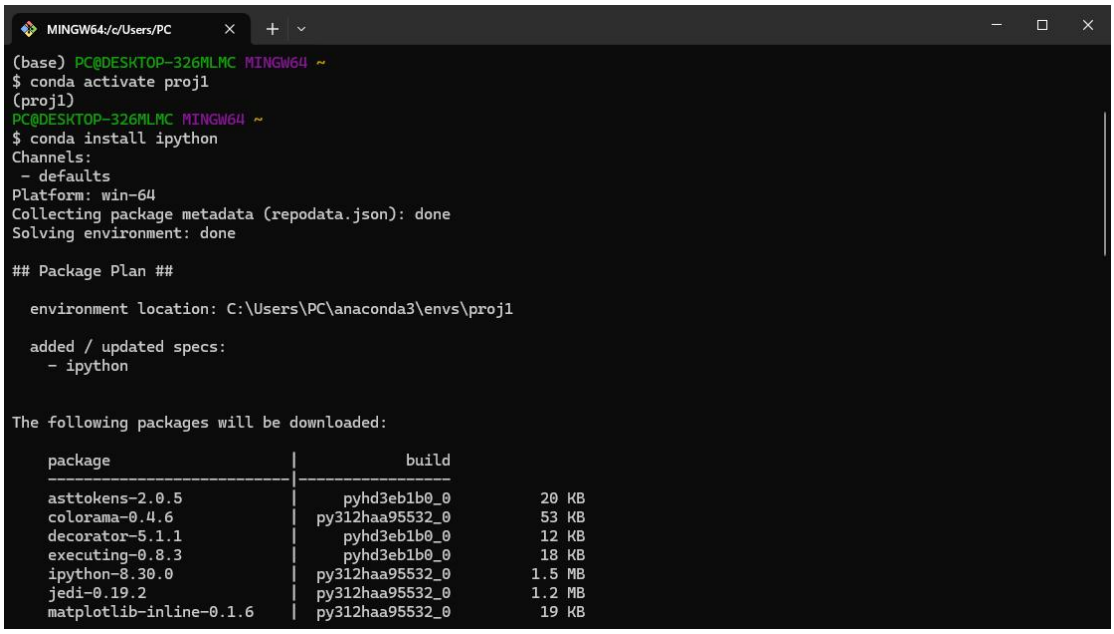


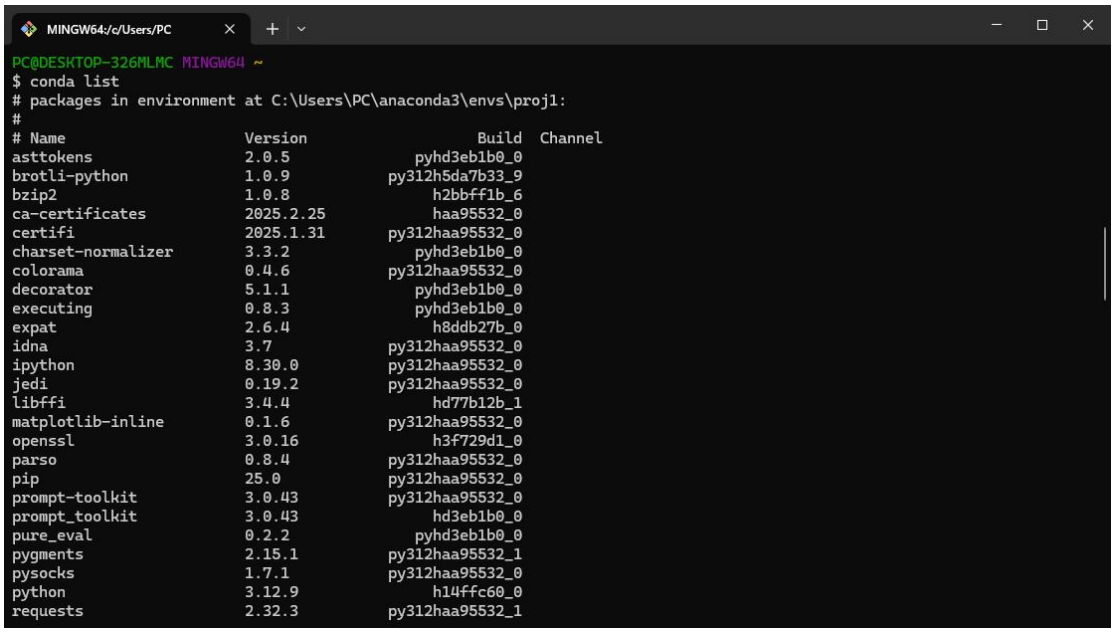
4. 使用 conda list 命令显示 Conda 环境里的软件包列表及其版本信息。



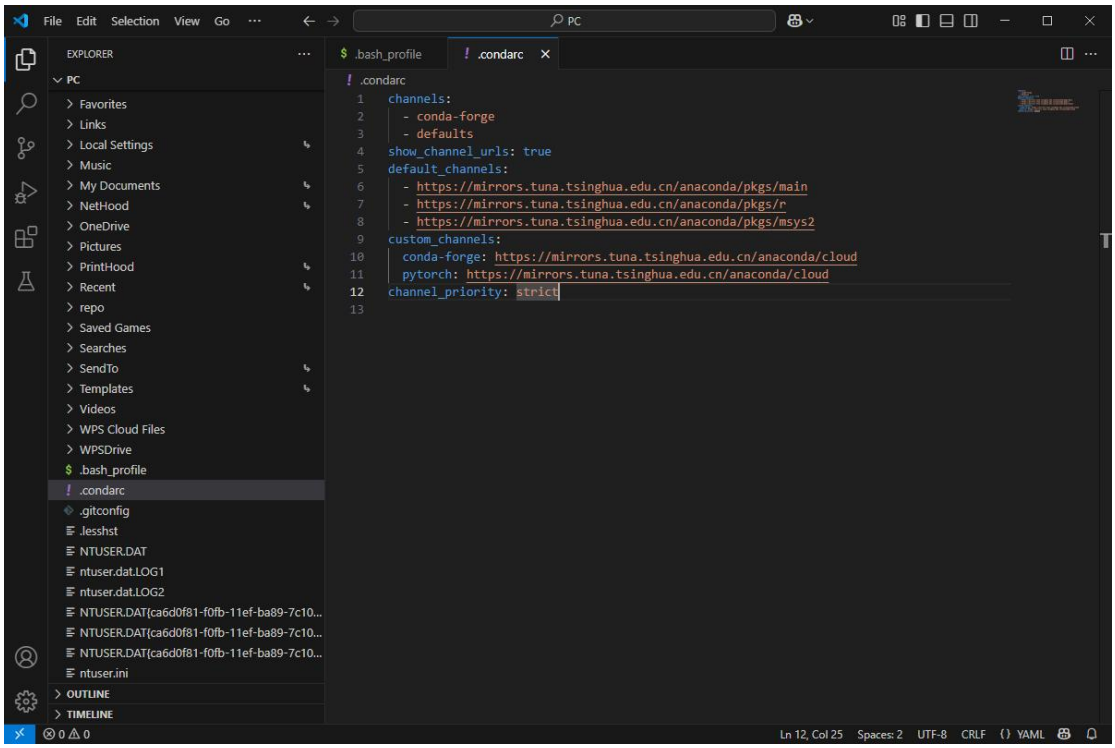


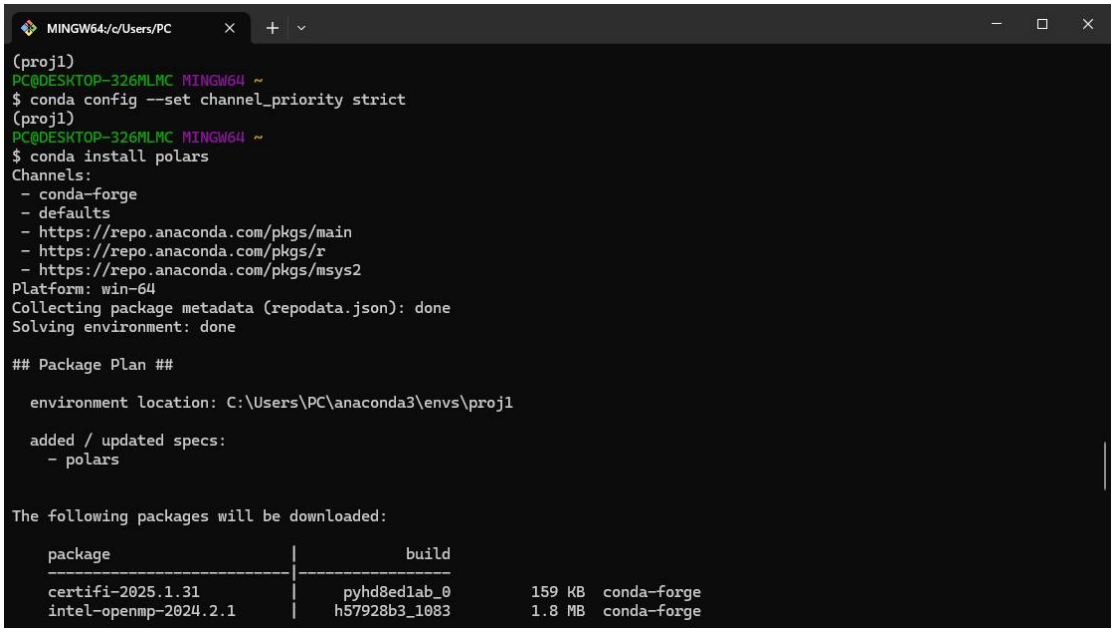
5. 使用 conda install 命令往 Conda 环境里安装 ipython 软件包，并验证版本。



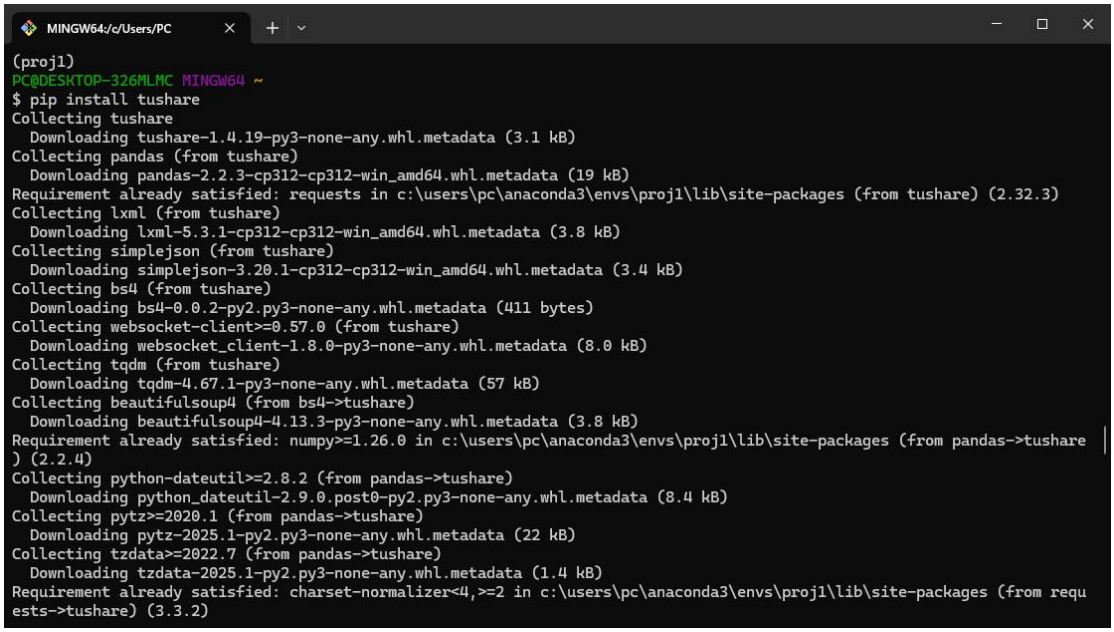


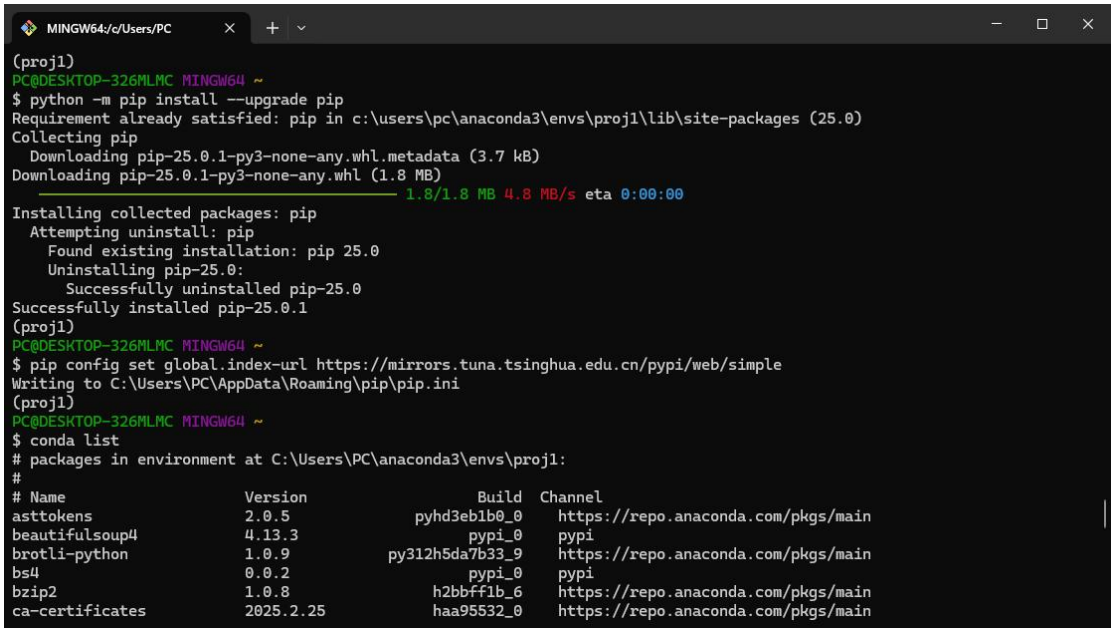
6. 配置 Anaconda 清华镜像，将 conda-forge 设置为默认 Channel，让 conda install 能够安装 polars 软件包。





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。





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/p

(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。

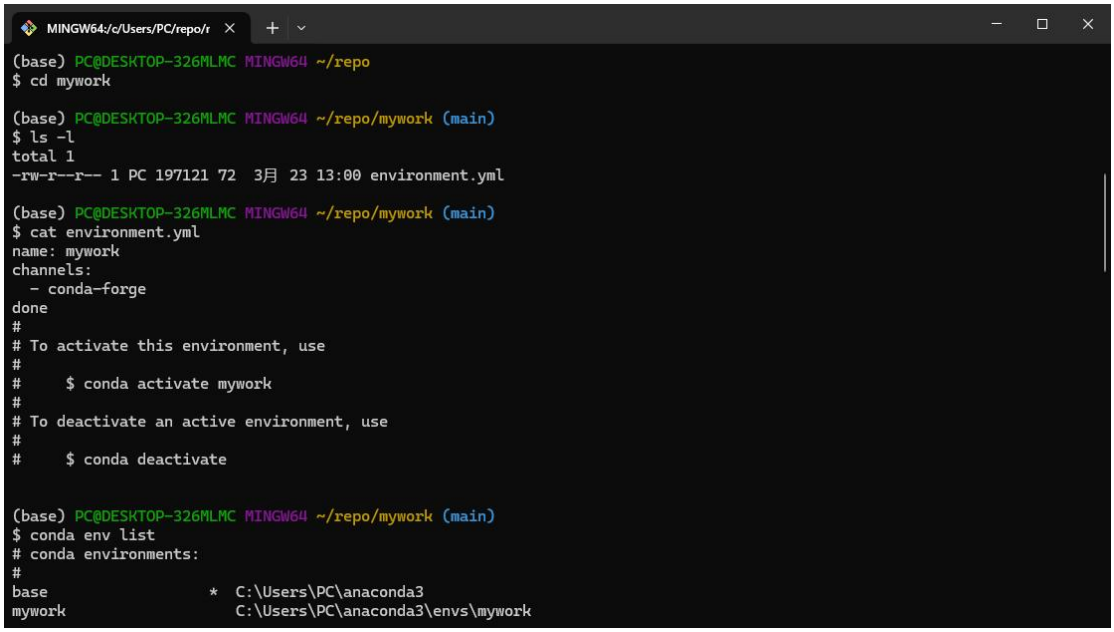
（2）Conda-forge 与 Conda 的关系

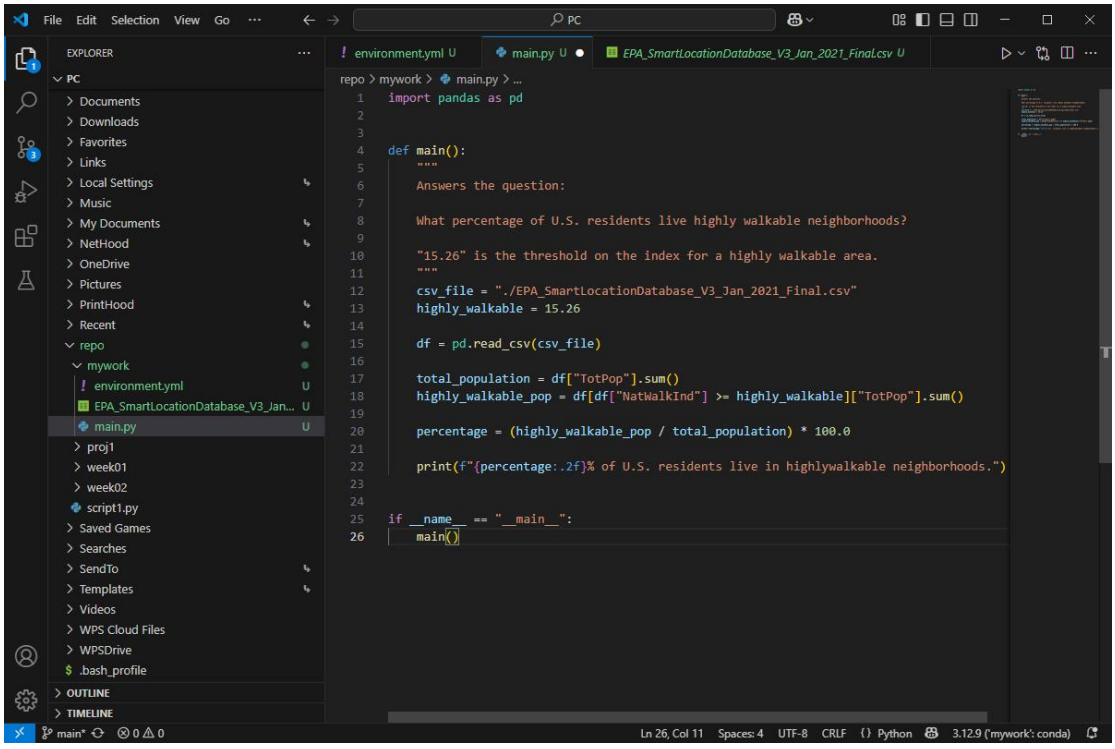
Conda-forge 是一个由社区驱动的项目，它为 Conda 提供了大量的软件包。Conda 本身支持从多个包源（channel）下载和安装包，Conda-forge 就是其中一个重要的包源。Conda 默认的包源可能没有包含所有你需要的包，而 Conda-forge 社区维护了大量的开源软件包，涵盖了科学计算、机器学习、数据处理等多个领域。

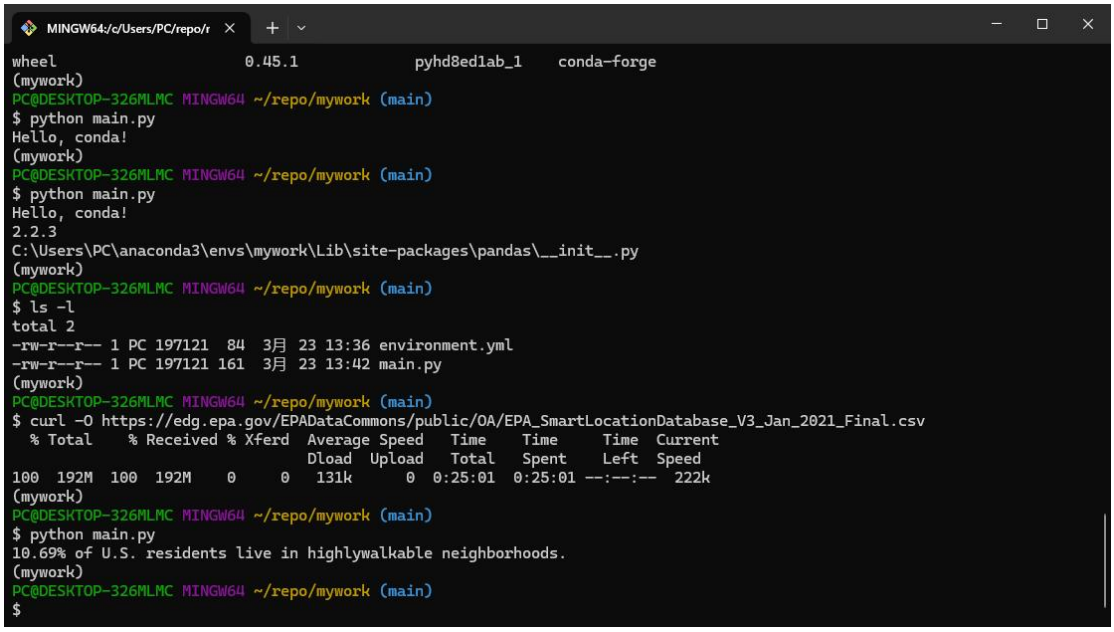
Conda-forge 是由社区驱动的，社区成员会不断地更新和维护包。这意味着 Conda-forge 上的包通常能够及时更新到最新版本，并且会修复已知的问题。相比之下，Conda 默认的包源可能更新速度较慢。

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
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)
$