第三周-Python项目

2025年3月29日　15:36

1. 在自己的终端（比如 Git Bash、Zsh 等）配置好 Conda Init，使得启动终端后，在提示符（比如 $、%）前能够看到（base）

```
DELL@DESKTOP-MOI26SB MINGW64 ~
$ conda init bash
no change     C:\Users\DELL\anaconda3\Scripts\conda.exe
no change     C:\Users\DELL\anaconda3\Scripts\conda-env.exe
no change     C:\Users\DELL\anaconda3\Scripts\conda-script.py
no change     C:\Users\DELL\anaconda3\Scripts\conda-env-script.p
y
no change     C:\Users\DELL\anaconda3\condabin\conda.bat
no change     C:\Users\DELL\anaconda3\Library\bin\conda.bat
no change     C:\Users\DELL\anaconda3\condabin\_conda_activate.b
at
no change     C:\Users\DELL\anaconda3\condabin\rename_tmp.bat
no change     C:\Users\DELL\anaconda3\condabin\conda_auto_activa
te.bat
no change     C:\Users\DELL\anaconda3\condabin\conda_hook.bat
no change     C:\Users\DELL\anaconda3\Scripts\activate.bat
no change     C:\Users\DELL\anaconda3\condabin\activate.bat
no change     C:\Users\DELL\anaconda3\condabin\deactivate.bat
no change     C:\Users\DELL\anaconda3\Scripts\activate
no change     C:\Users\DELL\anaconda3\Scripts\deactivate
no change     C:\Users\DELL\anaconda3\etc\profile.d\conda.sh
no change     C:\Users\DELL\anaconda3\etc\fish\conf.d\conda.fish
no change     C:\Users\DELL\anaconda3\shell\condabin\Conda.psm1
no change     C:\Users\DELL\anaconda3\shell\condabin\conda-hook.
ps1
no change     C:\Users\DELL\anaconda3\Lib\site-packages\xontrib\
conda.xsh
no change     C:\Users\DELL\anaconda3\etc\profile.d\conda.csh
modified      C:\Users\DELL\.bash_profile

==> For changes to take effect, close and re-open your current s
hell. <==
```

要在vs code中删除某行命令，可以在前面加#，将该行命令转化为注释。

```
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
```

由于配置原因，在配置了conda init之后，我在启动终端之后并没有显示出base，因此向chat gpt寻求解决办法并最终解决问题。

```
DELL@DESKTOP-MOI26SB MINGW64 ~
$ eval "$('/c/Users/DELL/anaconda3/Scripts/conda' "shell.bash" 'hook')"
(base)
DELL@DESKTOP-MOI26SB MINGW64 ~
$ conda activate base
(base)
DELL@DESKTOP-MOI26SB MINGW64 ~
$ which conda
/c/Users/DELL/anaconda3/Scripts/conda
(base)
DELL@DESKTOP-MOI26SB MINGW64 ~
```

2. 使用 conda info 命令查看本机 Conda 的配置信息

```
(base) DELL@DESKTOP-MOI26SB MINGW64 ~
$ conda info

     active environment : base
    active env location : C:\Users\DELL\anaconda3
            shell level : 1
       user config file : C:\Users\DELL\.condarc
 populated config files : C:\Users\DELL\anaconda3\.condarc
                          C:\Users\DELL\.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=12.5=0
                          __win=0=0
      base environment : C:\Users\DELL\anaconda3  (writable)
     conda av data dir : C:\Users\DELL\anaconda3\etc\conda
 conda av metadata url : None
          channel URLs : https://repo.anaconda.com/pkgs/main/win-64
                         https://repo.anaconda.com/pkgs/main/noarch
                         https://repo.anaconda.com/pkgs/r/win-64
                         https://repo.anaconda.com/pkgs/r/noarch
                         https://repo.anaconda.com/pkgs/msys2/win-64
                         https://repo.anaconda.com/pkgs/msys2/noarch
         package cache : C:\Users\DELL\anaconda3\pkgs
                         C:\Users\DELL\.conda\pkgs
                         C:\Users\DELL\AppData\Local\conda\conda\pkgs
```

3. 使用 conda env list 命令查看已有的 Conda 环境的名称和路径，理解 Conda **环境** 的概念

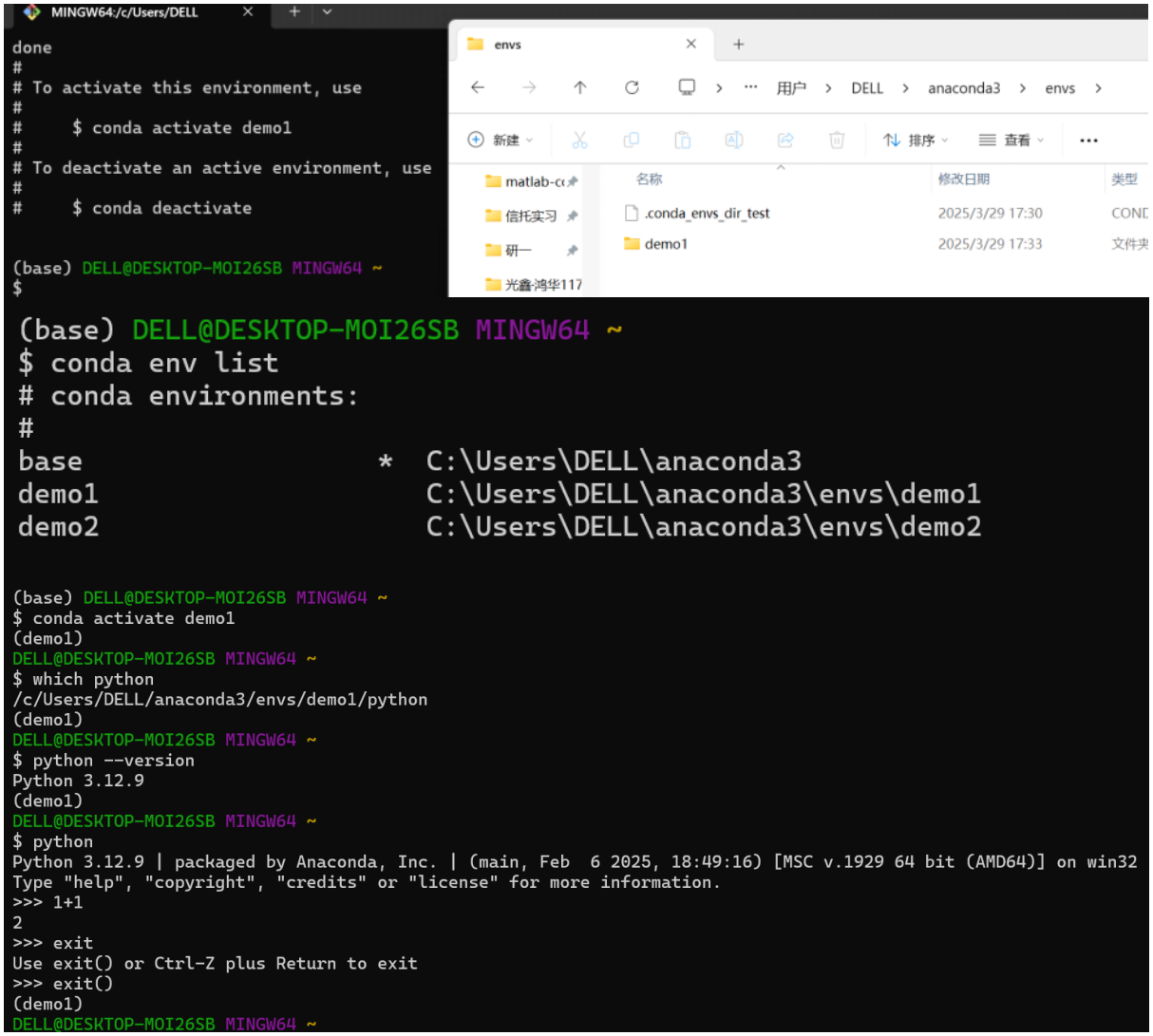
```
(base) DELL@DESKTOP-MOI26SB MINGW64 ~
$ conda env list
# conda environments:
#
base                  *  C:\Users\DELL\anaconda3
```

Conda 环境是 Anaconda 或 Miniconda 提供的一种工具，用于创建独立的、隔离的软件包和依赖项集合，使用户能够在同一台机器上为不同项目配置互不干扰的运行环境。通过 Conda 环境，用户可以为每个项目安装特定版本的编程语言（如 Python）、库（如 NumPy、Pandas）或其他工具，避免全局安装导致的版本冲突，同时支持跨平台（Windows/macOS/Linux）管理，并能方便地导出、共享或复现特定配置。

4. 使用 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

创建了两个conda环境，分别为demo1和demo2，并各自安装了软件包。

命令行conda activate demo1为激活python环境demo1，此时查询python的位置，会显示是在demo1下的python，通过“python --version“可以查看python版本为3.12.9。再次输入python则可以切换到python的模式，可以直接输入python的命令，quit（）或者exit()可以退出python模式，返回到demo1的目录下。



切换到demo2之后，查看python版本为3.9.21，然后用命令查看安装的软件包的版本

```
(demo2)
DELL@DESKTOP-MOI26SB MINGW64 ~
$ python --version
Python 3.9.21
(demo2)
DELL@DESKTOP-MOI26SB 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 requests
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'requests'
>>> import pandas
>>> pandas.__file__
'C:\\Users\\DELL\\anaconda3\\envs\\demo2\\lib\\site-packages\\pandas\\__init__.py'
>>> pandas.__version__
'2.2.3'
>>> import statsmodels
>>> statsmodels.__version__
'0.14.4'
>>>
```

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

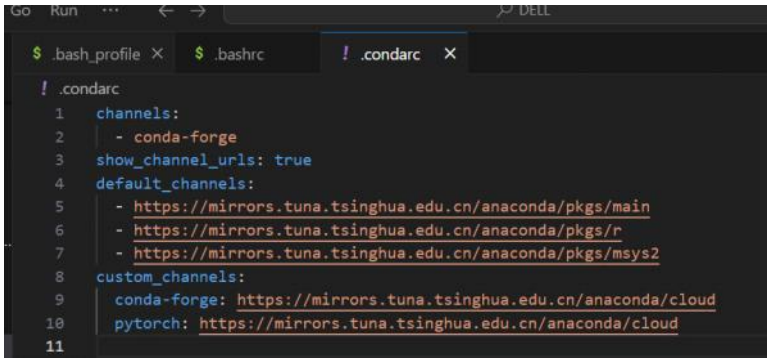
```
(demo2)
DELL@DESKTOP-MOI26SB MINGW64 ~
$ conda list
# packages in environment at C:\Users\DELL\anaconda3\envs\demo2:
#
# Name                    Version                   Build  Channel
blas                      1.0                         mkl
bottleneck                1.4.2           py39hc99e966_0
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
six                       1.16.0             pyhd3eb1b0_1
sqlite                    3.45.3               h2bbff1b_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

安装ipython，并用conda list查看

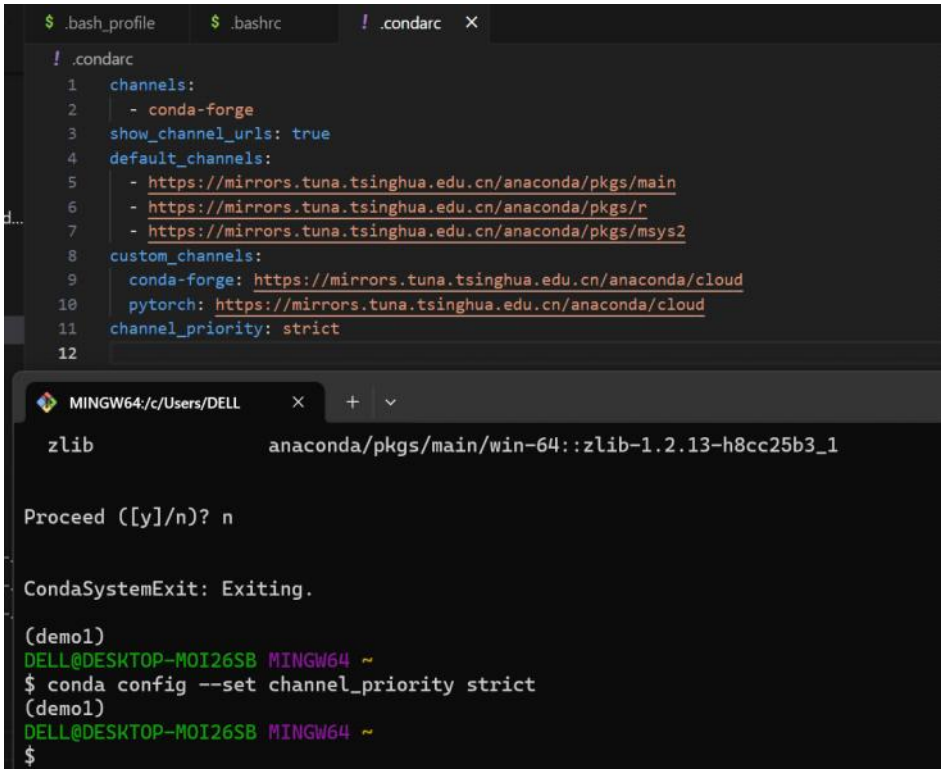
```
## Package Plan ##

  environment location: C:\Users\DELL\anaconda3\envs\demo1

  added / updated specs:
    - ipython


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    ipython-8.30.0             |  py312haa95532_0         1.5 MB
done
(demo1)
DELL@DESKTOP-MOI26SB MINGW64 ~
$ conda list
# packages in environment at C:\Users\DELL\anaconda3\envs\demo1:
#
# 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
```

7. 根据 文档，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

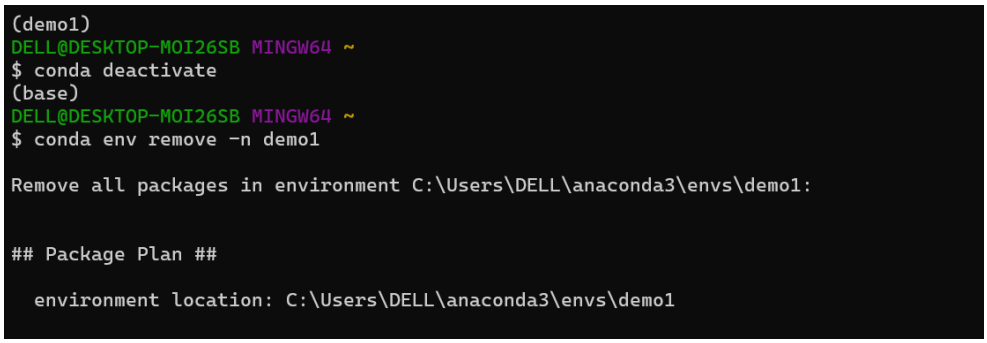


Conda-forge概述：conda-forge是由开源社区维护的Conda软件包通道，提供高质量、最新的开源软件包，支持跨平台（Windows/macOS/Linux）并拥有自动化CI/CD流程。截至2025年，其收录了超2.3万个软件包，每月下载量超1亿次，涵盖数据科学、机器学习等领域，是替代默认Anaconda通道的常用选择。

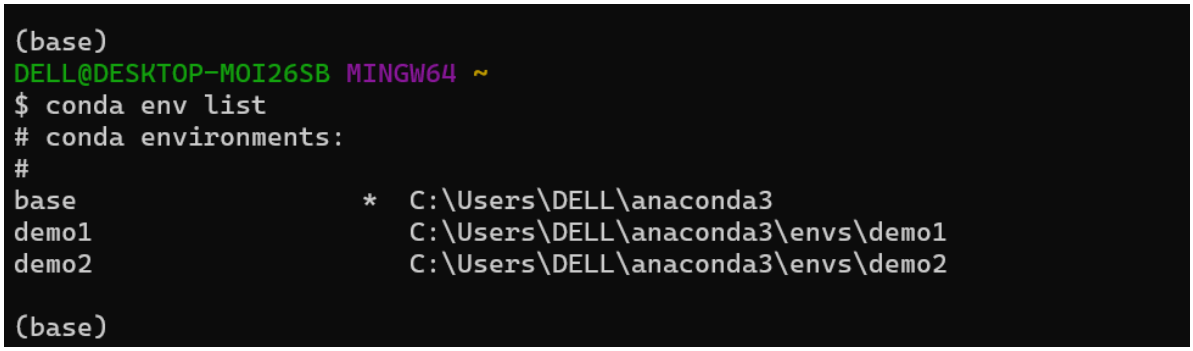
将 conda-forge 设置为默认 Channel之后，安装了polars，如下图所示。



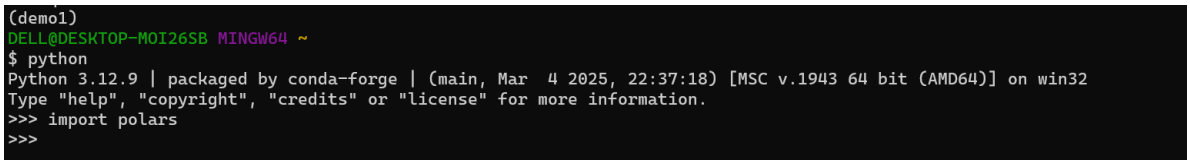
Conda install polars失败后，删除conda环demo1，如下图所示。



然后重新创建conda环境demo1，选择polars版本为polars-lts-cpu，demo1创建成功后如下图所示。



此时进入conda环境demo1中，成功导入软件包poalrs，如下图所示。



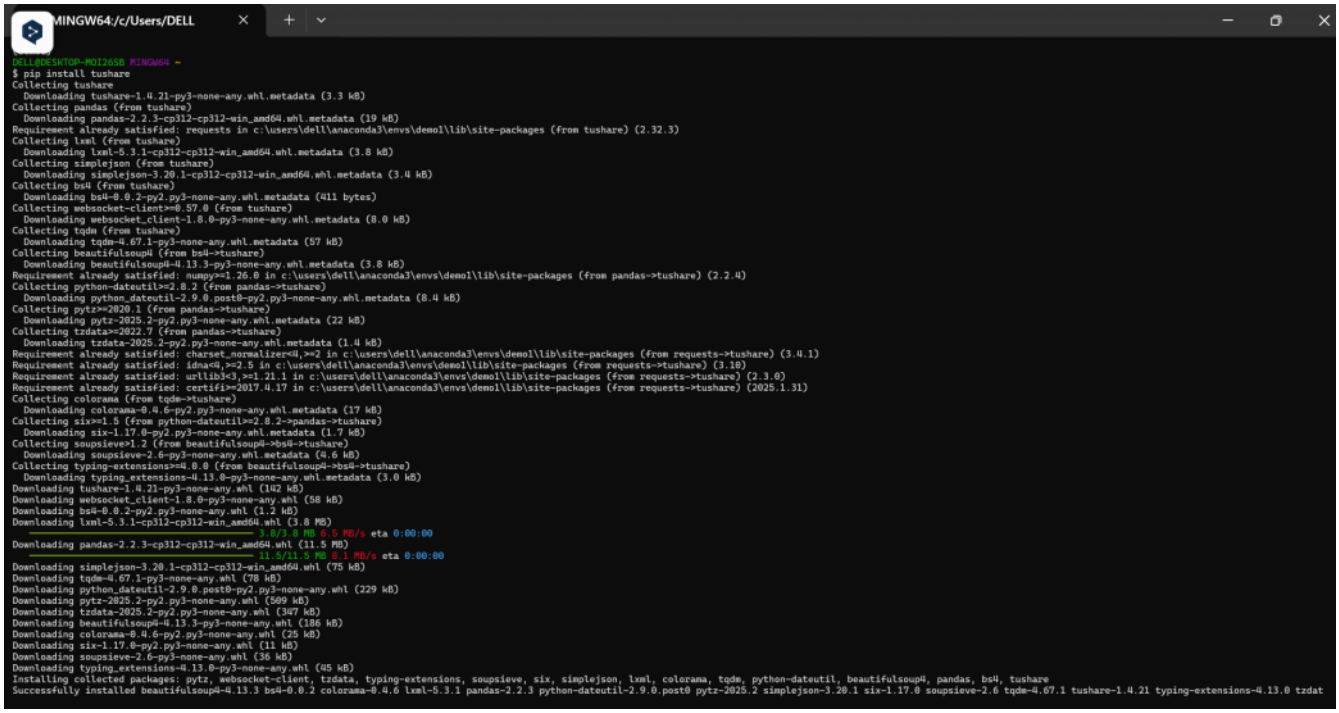
8. 使用 pip install 命令往 Conda 环境里安装 Python 软件包，并验证版本

此时使用conda install tushare命令下载软件包，下载失败，如下图所示。

```
done
Solving environment: failed

PackagesNotFoundError: The following packages are not available from current channels:

  - tushare

Current channels:

  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge
  - defaults
  - https://repo.anaconda.com/pkgs/main
  - https://repo.anaconda.com/pkgs/r
  - https://repo.anaconda.com/pkgs/msys2

To search for alternate channels that may provide the conda package you're
looking for, navigate to

    https://anaconda.org

and use the search bar at the top of the page.


(demo1)
DELL@DESKTOP-MOI26SB MINGW64 ~
$
```

可以看到下载用到的channels第一行为清华镜像，也说明之前的配置生效。

现在使用pip install tushare命令下载，执行结果如下图所示，软件包下载成功。



9. 根据 文档 配置 PyPI 清华镜像，加快 pip install 安装软件包的速度

首先使用命令python -m pip install --upgrade pip，然后使用命令pip config set global.index-url将清华镜像设为默认。指令执行结果如下图所示。

```
(demo1)
DELL@DESKTOP-MOI26SB MINGW64 ~
$ python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\dell\anaconda3\envs\demo1\lib\site-packages (25.0.1)
(demo1)
DELL@DESKTOP-MOI26SB MINGW64 ~
$ pip config set global.index-url https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple
Writing to C:\Users\DELL\AppData\Roaming\pip\pip.ini
```

由于前面已经成功下载了tushare，如下图所示，故不再重新使用清华镜像下载。

```
(demo1)
DELL@DESKTOP-MOI26SB MINGW64 ~
$ conda list | grep tushare
tushare                   1.4.21                   pypi_0    pypi
(demo1)
```

由于conda list的执行结果比较长，为清晰展示tushare下载结果，这里使用了grep搜索工具。

进入conda环境demo1的python，导入软件包tushare，查看版本及文件存放位置。三条指令的执行结果如下图所示。

```
(demo1)
DELL@DESKTOP-MOI26SB MINGW64 ~
$ python
Python 3.12.9 | packaged by conda-forge | (main, Mar  4 2025, 22:37:18) [MSC v.1943 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tushare
>>> tushare.__version__
'1.4.21'
>>> tushare.__file__
'C:\\Users\\DELL\\anaconda3\\envs\\demo1\\Lib\\site-packages\\tushare\\__init__.py'
>>>
```

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

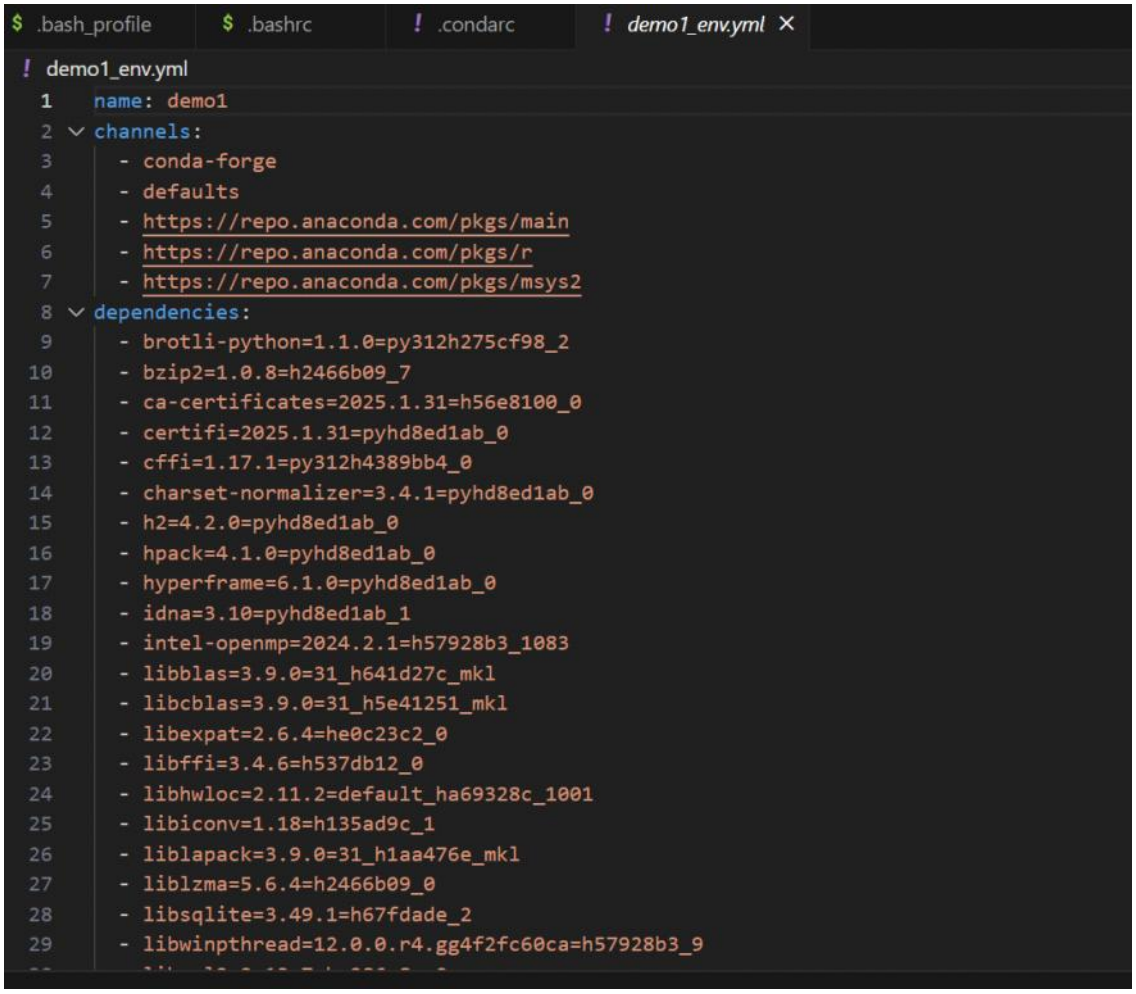
使用conda env export命令导出当前conda环境demo1的配置文件，如下图所示。

```
(demo1)
DELL@DESKTOP-MOI26SB MINGW64 ~
$ conda env export
name: demo1
channels:
  - conda-forge
  - defaults
  - https://repo.anaconda.com/pkgs/main
  - https://repo.anaconda.com/pkgs/r
  - https://repo.anaconda.com/pkgs/msys2
dependencies:
  - brotli-python=1.1.0=py312h275cf98_2
  - bzip2=1.0.8=h2466b09_7
  - ca-certificates=2025.1.31=h56e8100_0
  - certifi=2025.1.31=pyhd8ed1ab_0
  - cffi=1.17.1=py312h4389bb4_0
  - charset-normalizer=3.4.1=pyhd8ed1ab_0
  - h2=4.2.0=pyhd8ed1ab_0
  - hpack=4.1.0=pyhd8ed1ab_0
  - hyperframe=6.1.0=pyhd8ed1ab_0
  - idna=3.10=pyhd8ed1ab_1
  - intel-openmp=2024.2.1=h57928b3_1083
  - libblas=3.9.0=31_h641d27c_mkl
  - libcblas=3.9.0=31_h5e41251_mkl
  - libexpat=2.6.4=he0c23c2_0
  - libffi=3.4.6=h537db12_0
  - libhwloc=2.11.2=default_ha69328c_1001
  - libiconv=1.18=h135ad9c_1
  - liblapack=3.9.0=31_h1aa476e_mkl
  - liblzma=5.6.4=h2466b09_0
  - libsqlite=3.49.1=h67fdade_2
  - libwinpthread=12.0.0.r4.gg4f2fc60ca=h57928b3_9
  - libxml2=2.13.7=he286e8c_0
  - python_abi=3.12=5_cp312
  - requests=2.32.3=pyhd8ed1ab_1
  - setuptools=75.8.2=pyhff2d567_0
  - tbb=2021.13.0=h62715c5_1
  - tk=8.6.13=h5226925_1
  - ucrt=10.0.22621.0=h57928b3_1
  - urllib3=2.3.0=pyhd8ed1ab_0
  - vc=14.3=h2b53caa_26
  - vc14_runtime=14.42.34438=hfd919c2_26
  - wheel=0.45.1=pyhd8ed1ab_1
  - win_inet_pton=1.1.0=pyh7428d3b_8
  - zstandard=0.23.0=py312h4389bb4_1
  - pip:
      - beautifulsoup4==4.13.3
      - bs4==0.0.2
      - colorama==0.4.6
      - lxml==5.3.1
      - pandas==2.2.3
      - polars-lts-cpu==1.26.0
      - python-dateutil==2.9.0.post0
      - pytz==2025.2
      - simplejson==3.20.1
      - six==1.17.0
      - soupsieve==2.6
      - tqdm==4.67.1
      - tushare==1.4.21
      - typing-extensions==4.13.0
      - tzdata==2025.2
      - websocket-client==1.8.0
prefix: C:\Users\DELL\anaconda3\envs\demo1
```

为方便查找，重新将环境导出到demo1_env.yml中，如下图所示。

```
(demo1)
DELL@DESKTOP-MOI26SB MINGW64 ~
$ conda env export -f demo1_env.yml
```

在vscode中打开demo1_env.yml文件，如下图所示。



可以看到demo1的channels、dependencies、pip等信息都存在。

现在将demo1删除，删除前后使用conda env list命令查看当前的conda环境。删除命令执行前后，执行conda env list的结果如下图所示。

```
(base)
DELL@DESKTOP-MOI26SB MINGW64 ~
$ conda env list
# conda environments:
#
base                  *  C:\Users\DELL\anaconda3
demo1                    C:\Users\DELL\anaconda3\envs\demo1
demo2                    C:\Users\DELL\anaconda3\envs\demo2
```

```
(base)
DELL@DESKTOP-MOI26SB MINGW64 ~
$ conda env list
# conda environments:
#
base                  *  C:\Users\DELL\anaconda3
demo2                    C:\Users\DELL\anaconda3\envs\demo2
```

可以看到demo1已经被成功删除。

现在在我的项目文件夹my_repo中新建文件夹demo1，将前面导出的配置文件demo1_env.yml移动到文件夹demo1下。相关命令执行结果如下图所示。

```
(base)
DELL@DESKTOP-MOI26SB MINGW64 ~
$ cd my_repo/
(base)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo
$ ls -l
total 0
(base)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo
$ mkdir demo1
(base)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo
$ cd demo1/
(base)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ ls -l
total 0
(base)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$  mv ~/demo1_env.yml ./
(base)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ ls -l
total 4
-rw-r--r-- 1 DELL 197121 2055 Mar 29 20:52 demo1_env.yml
(base)
```

此时使用conda env list查看conda环境，如下图所示。

```
(base)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ conda env list
# conda environments:
#
base                  *  C:\Users\DELL\anaconda3
demo2                    C:\Users\DELL\anaconda3\envs\demo2
```

可以看到此时demo1还不是conda环境，只是普通目录。

现在根据配置文件demo_env.yml创建conda环境，命令执行结果如下图所示。

```
done
#
# To activate this environment, use
#
#     $ conda activate demo1
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

此时查看conda环境，如下图所示。

```
(base)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ conda env list
# conda environments:
#
base                  *  C:\Users\DELL\anaconda3
demo1                    C:\Users\DELL\anaconda3\envs\demo1
demo2                    C:\Users\DELL\anaconda3\envs\demo2
```

可以看到已经成功通过配置文件demo1_env.ym重新l创建了conda环境demo1。

激活demo1后可以看到polars也存在，如下图所示

```
(demo1)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ conda list | grep polars
polars-lts-cpu            1.26.0                   pypi_0    pypi
```

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

    通过deepseek简单了解了该问题。

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

    第一步，创建environment.yml文件，写入依赖项。

```
(demo1)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ mkdir my_python_project
(demo1)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ touch environment.yml
(demo1)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ cat environment.yml
(demo1)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ vim environment.yml
(demo1)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ cat environment.yml
name: my-project
channels:
  - defaults
dependencies:
  - python
(demo1)
```

    第二步，根据配置文件创建conda环境并激活。

```
(demo1)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ conda env list
# conda environments:
#
base                     C:\Users\DELL\anaconda3
demo1                 *  C:\Users\DELL\anaconda3\envs\demo1
demo2                    C:\Users\DELL\anaconda3\envs\demo2
my-project               C:\Users\DELL\anaconda3\envs\my-project

$ conda activate my-project
(my-project)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
```

    可以看到conda环境my-project创建成功并激活。

    第三步，编写python程序，打印一个七行的菱形，运行结果如下图所示。

```
(my-project)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ touch hello.py
(my-project)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ vim hello.py
(my-project)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ python ./
demo1_env.yml      environment.yml    hello.py           my_python_project/
(my-project)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ python ./
demo1_env.yml      environment.yml    hello.py           my_python_project/
(my-project)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$ python ./hello.py
   *
  ***
 *****
*******
 *****
  ***
   *
(my-project)
DELL@DESKTOP-MOI26SB MINGW64 ~/my_repo/demo1
$
```