Data Engineer (Python) Test

**Scenario:**

Your Business Intelligence team has been requested to create a visualization illustrating the comparative performance of soccer teams in 2020, 2021, 2022 and 2023 in the English Premier League.

The KPIs they are looking for are derived from games won, drawn, lost, goals for and goals against, for the EPL teams in the desired timeframe.

They have reached out to you so that you can help summarize and download the necessary data from <https://api.football-data.org/> , a third party open data API. Your manager was able to get a set of credentials for you to use (refer git repo).

The data needs to be in a tabular format (csv/ excel), so that it can be readily consumed.

**Suggested tasks:**

1. Fork and clone the github repository at: <https://github.com/datatest999/test>
2. On your local machine, create a virtual environment for Python
3. Identify the necessary end-points to use for the task at hand for <https://api.football-data.org/>
   * The API credentials can be found in the Readme file.
   * Further details about the available end-points and filters is available here: <https://www.football-data.org/documentation/quickstart/>
4. Create Python script(s) that are able to extract the data needed, and transform it to csv format.
5. Optimize the code and create unit tests in python.
6. Update dependencies (requirements.txt, gitignore, etc) files as needed.
7. Generate final csv output for submission.
8. Share the csv files created by the script as well the scripts to the git repository through a pull request against the original repository for review.

**Evaluation criteria:**

1. Ability of the code to get the job done.
2. Adherence to coding standards, including but not limited to commenting, use of functional approach, modularity, exception handling,
3. Management of dependencies and unit tests.

Answer:

**Programs**

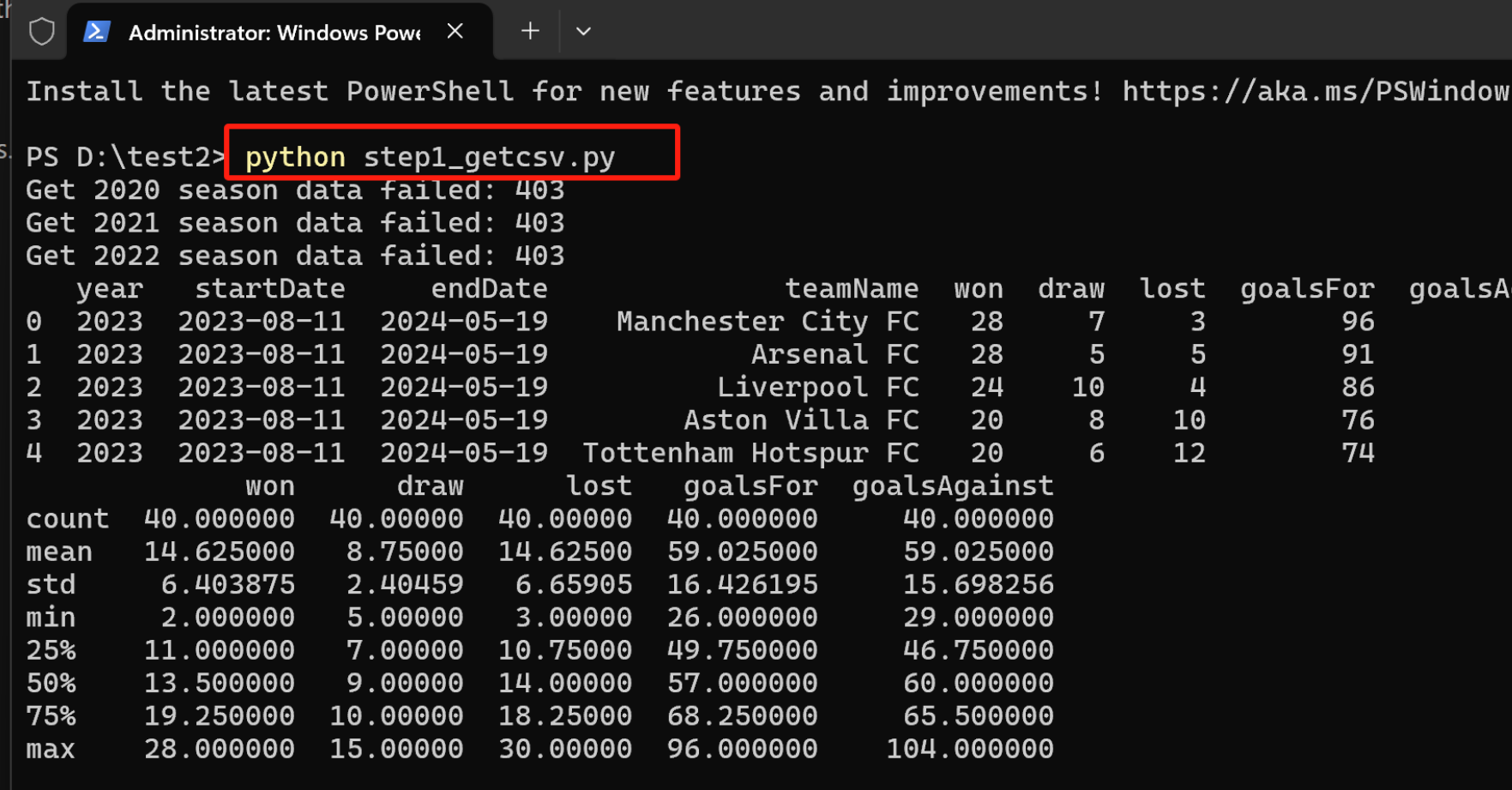
I created two programs to finish this task.

One Named step1\_getcsv.py, Get data from API then save it as CSV.

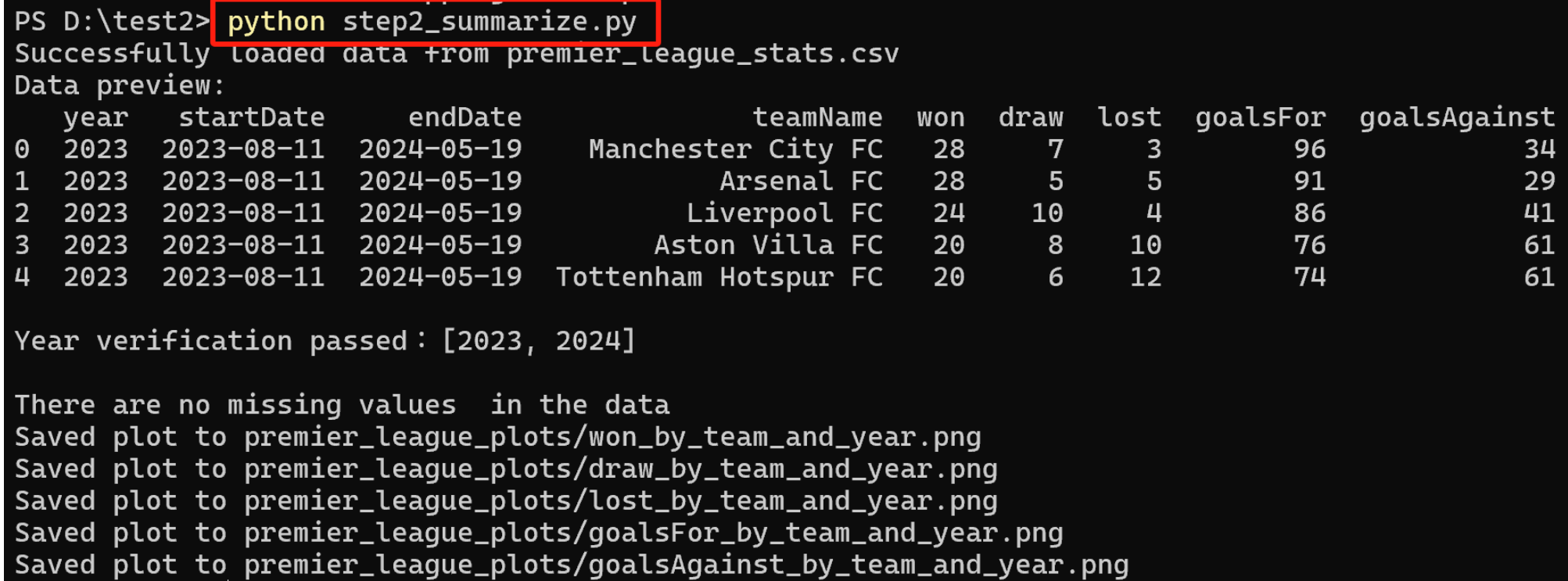
The other one named step2\_summarize.py, Clean the data and save the summaries as graphs.

Run “python step1\_getcsv.py”

The script retrieves data from football-data.org and saves the raw data as a CSV file named premier\_league\_stats.csv in the root directory.



Run “python step2\_summarize.py”



As a result of the data summary, six graphs have been generated with the following names:

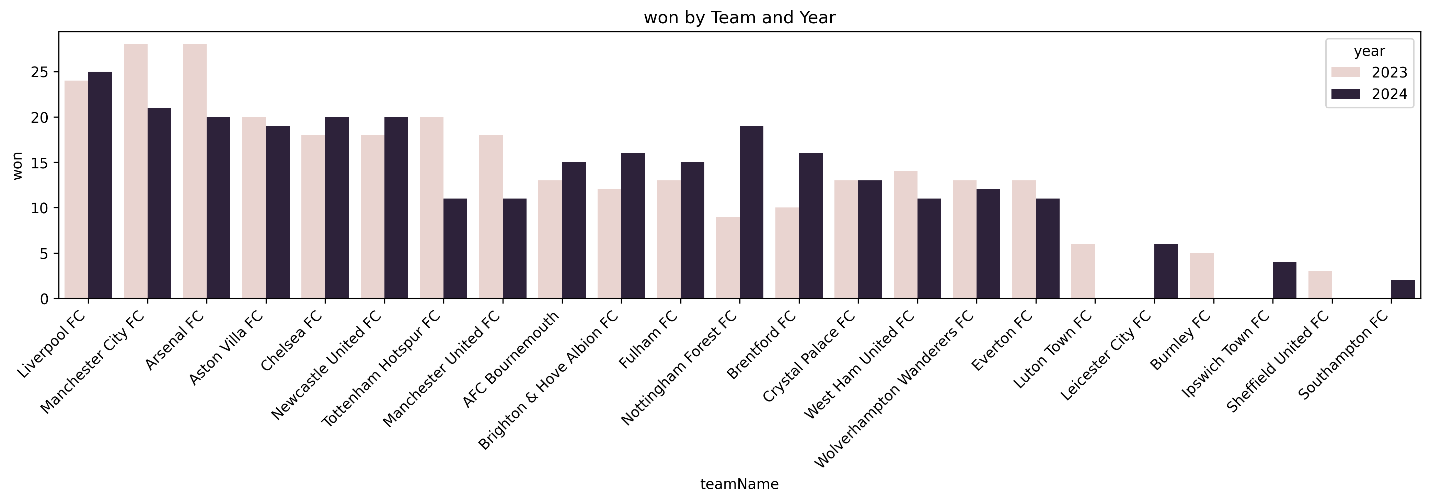
"premier\_league\_plots\won\_by\_team\_and\_year.png",

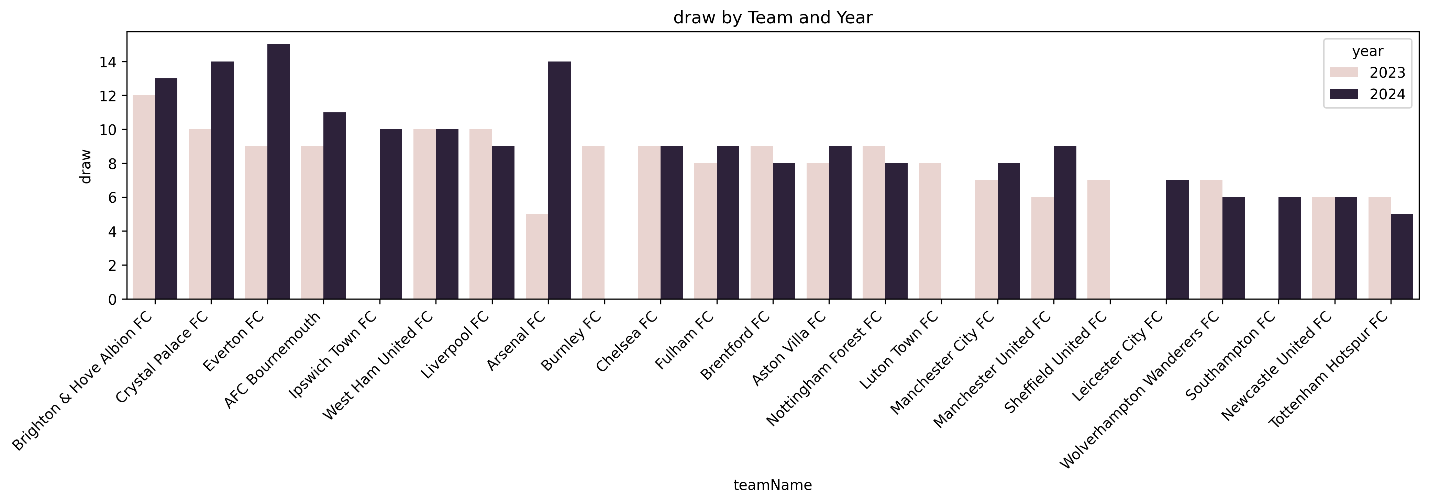
"premier\_league\_plots\draw\_by\_team\_and\_year.png",

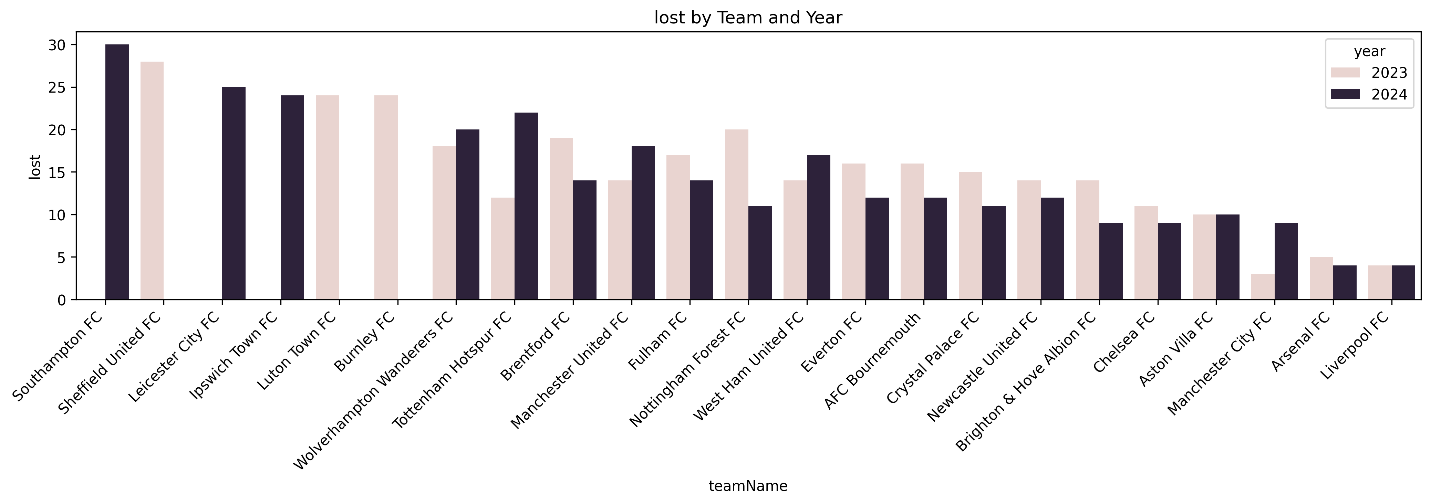
"premier\_league\_plots\lost\_by\_team\_and\_year.png",

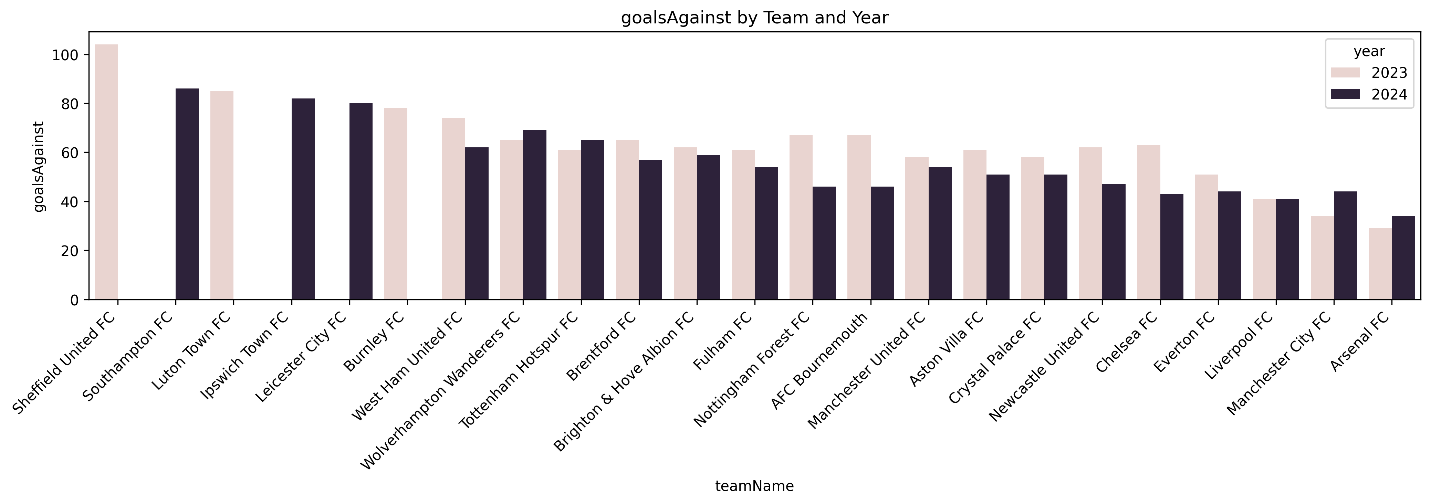
"premier\_league\_plots\goalsFor\_by\_team\_and\_year.png",

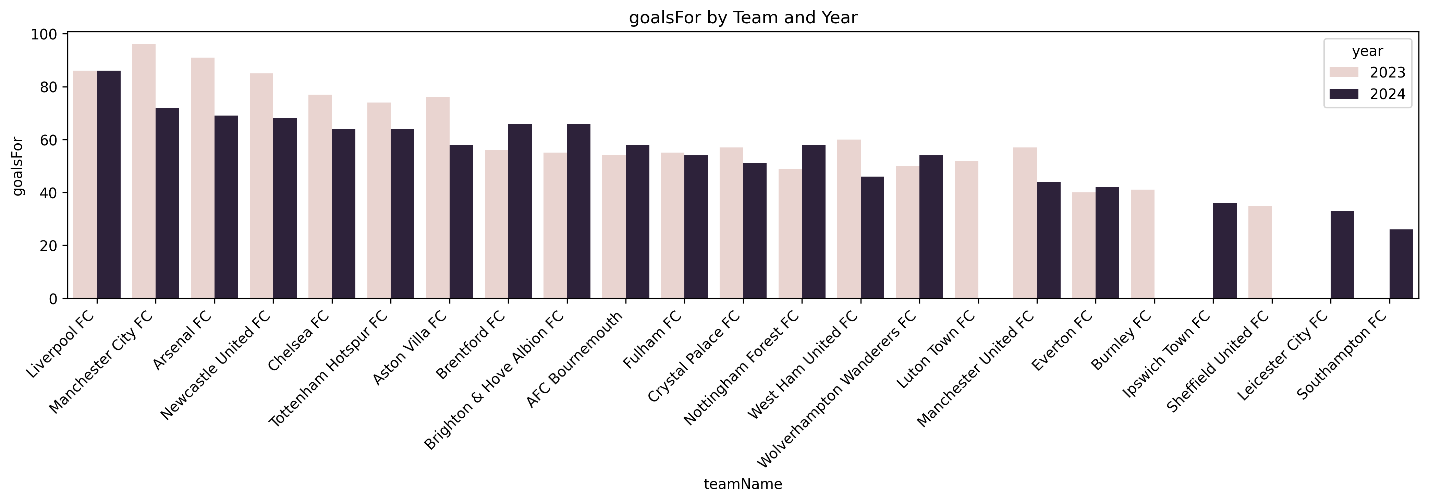
"premier\_league\_plots\goalsAgainst\_by\_team\_and\_year.png".









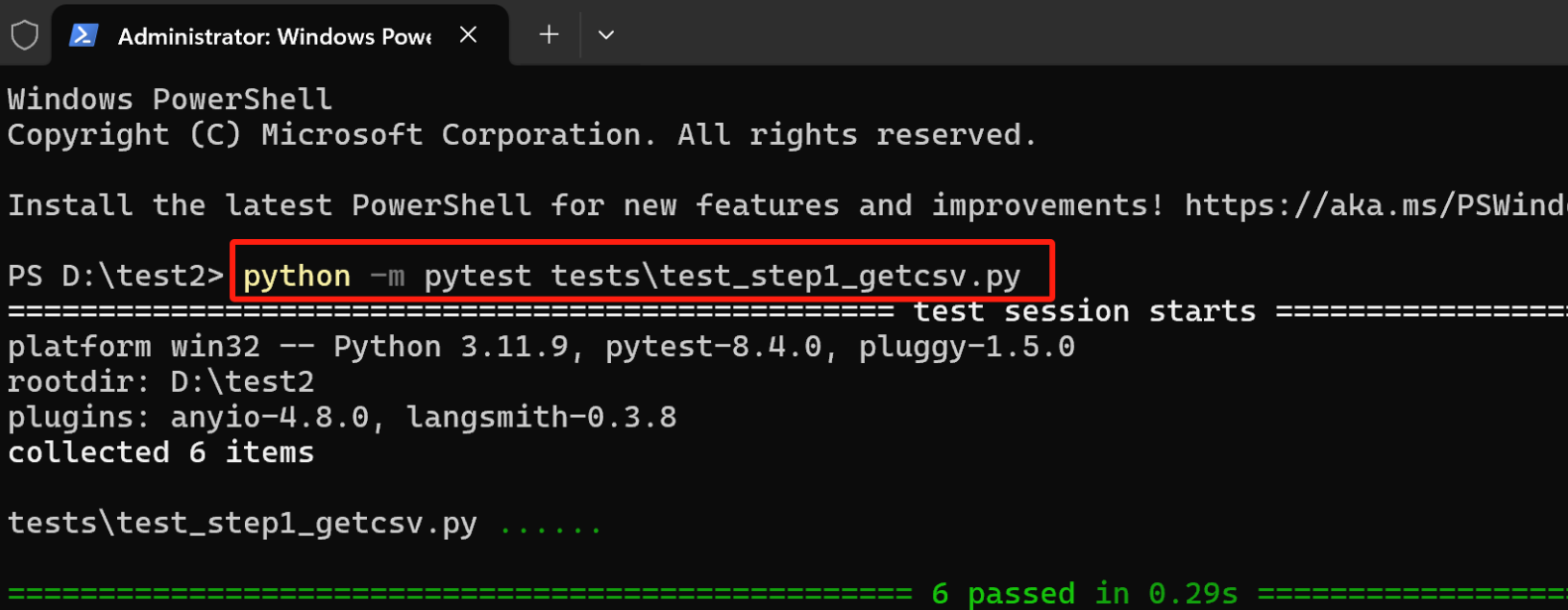


**Unit tests**

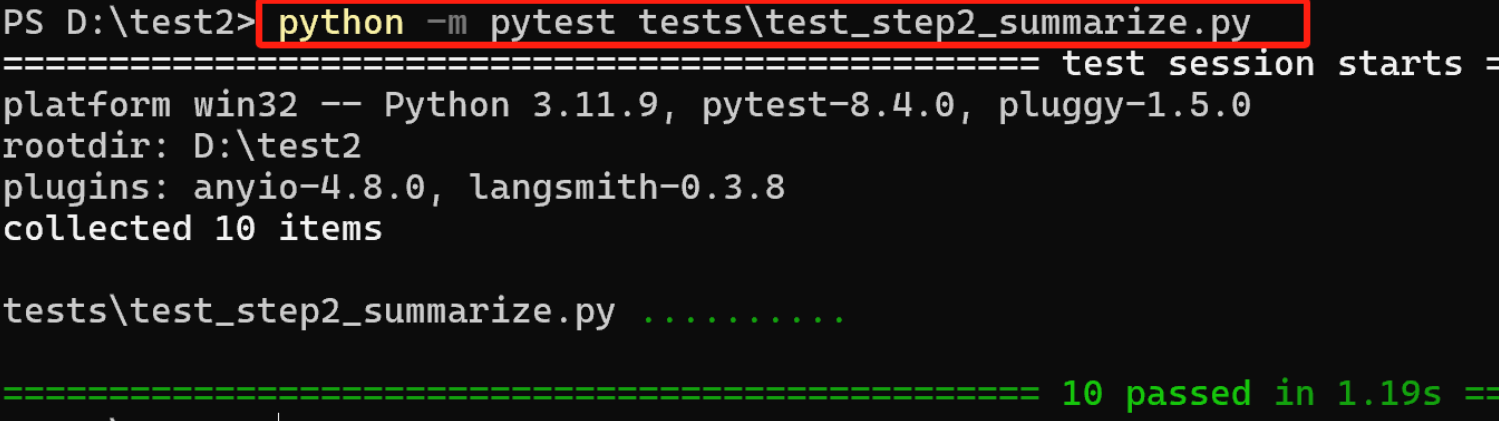
I use pytest.

There are two test files named tests\test\_step1\_getcsv.py and tests\test\_step2\_summarize.py.

You can run python -m pytest tests\test\_step1\_getcsv.py.



You can run python -m pytest tests\test\_step2\_summarize.py



**Program limitation**

This token may be a token for the free package, so only data for the past two years can be obtained, and the requested data for 2020-2022 cannot be obtained.

If we have a token of higher-level packagem, we will access to the requested data for 2020 to 2023.

**Github Readme**

Github Readme has a detailed explanation.

