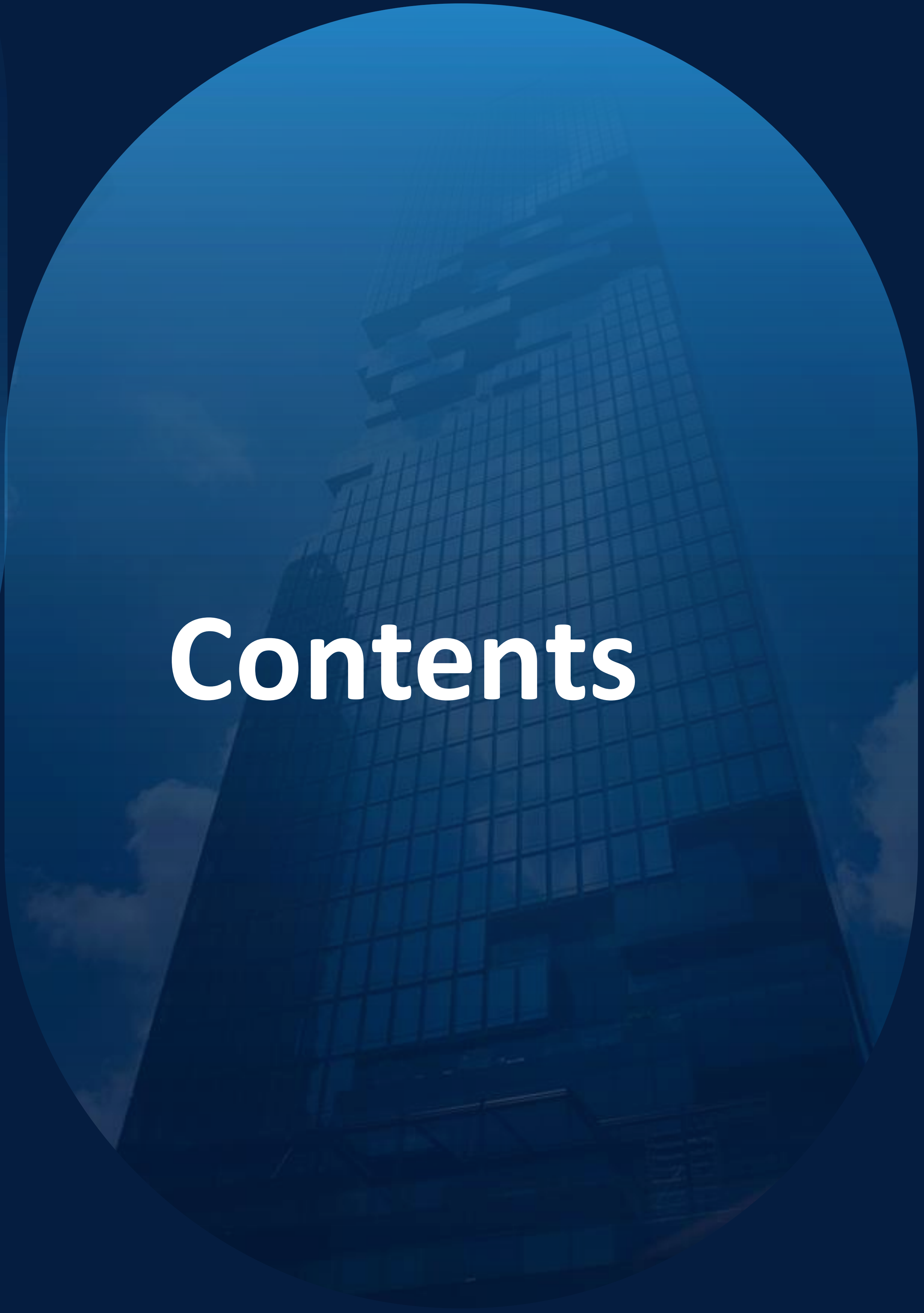
ETL Project

NASA Mars Rover Data Analysis

Presented by: Victoria Chueh



Contents

01 ETL Flow

Extract

NASA API

↓

Collection Python

↓

mars_rover_photos_summary.csv

↓

Daily Script

Transform

Cleaning Python

↓

mars_rover_photos_cleaned_summary.csv

Load

SQLite Database

↓

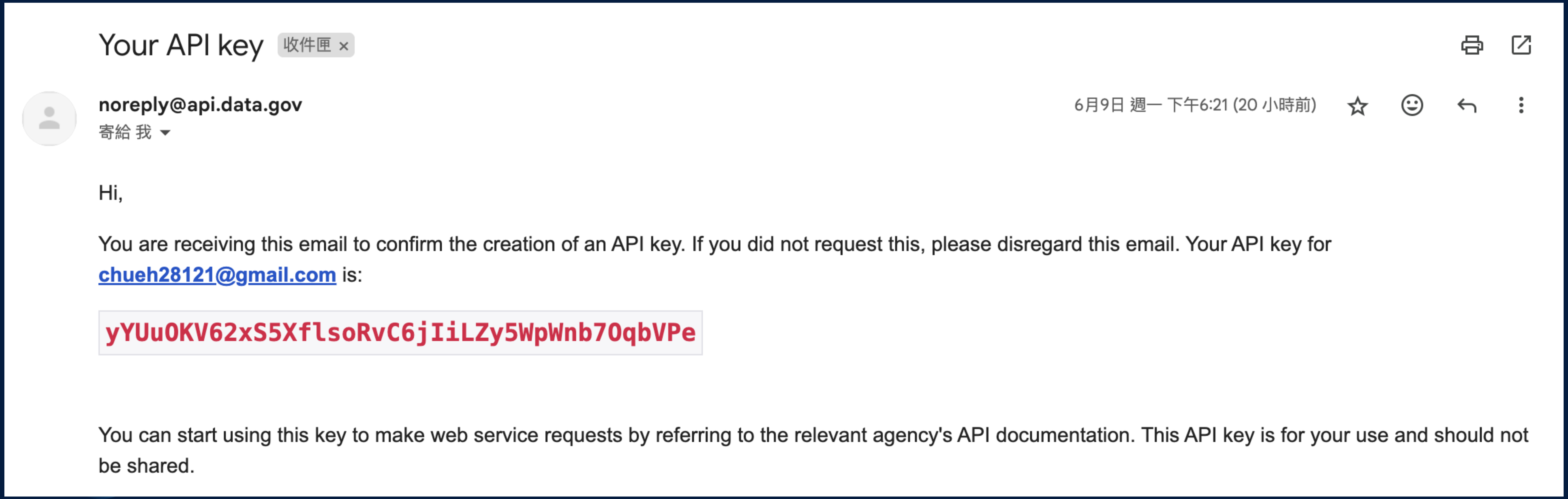
Export csv for visualization

↓

Tableau

02 Data Collection

NASA APIs – Imagery, astronomy, satellites, and more
https://api.nasa.gov/



Your API key 收件匣 ×

noreply@api.data.gov
寄給 我

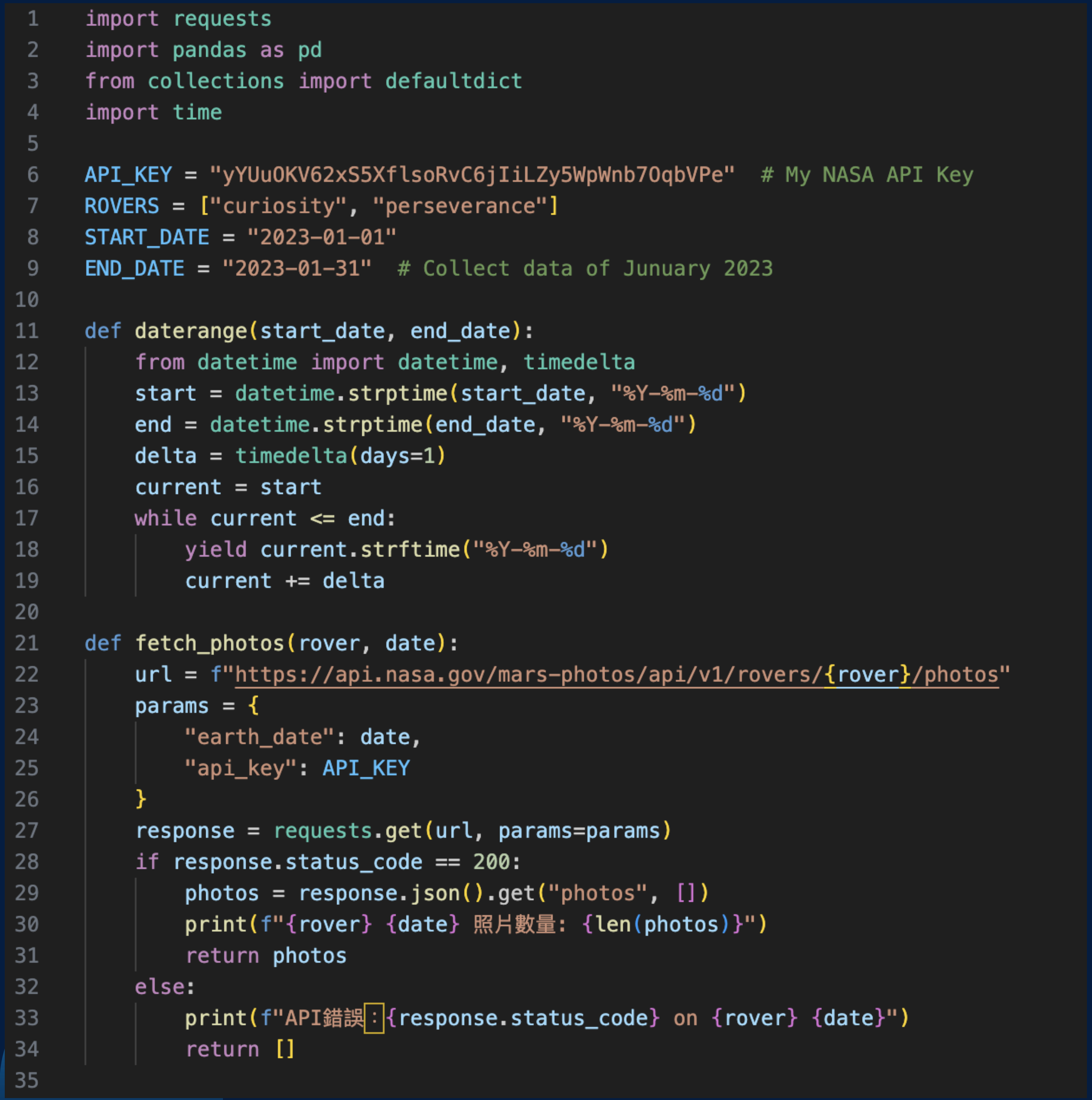
6月9日 週一 下午6:21 (20 小時前)

Hi,

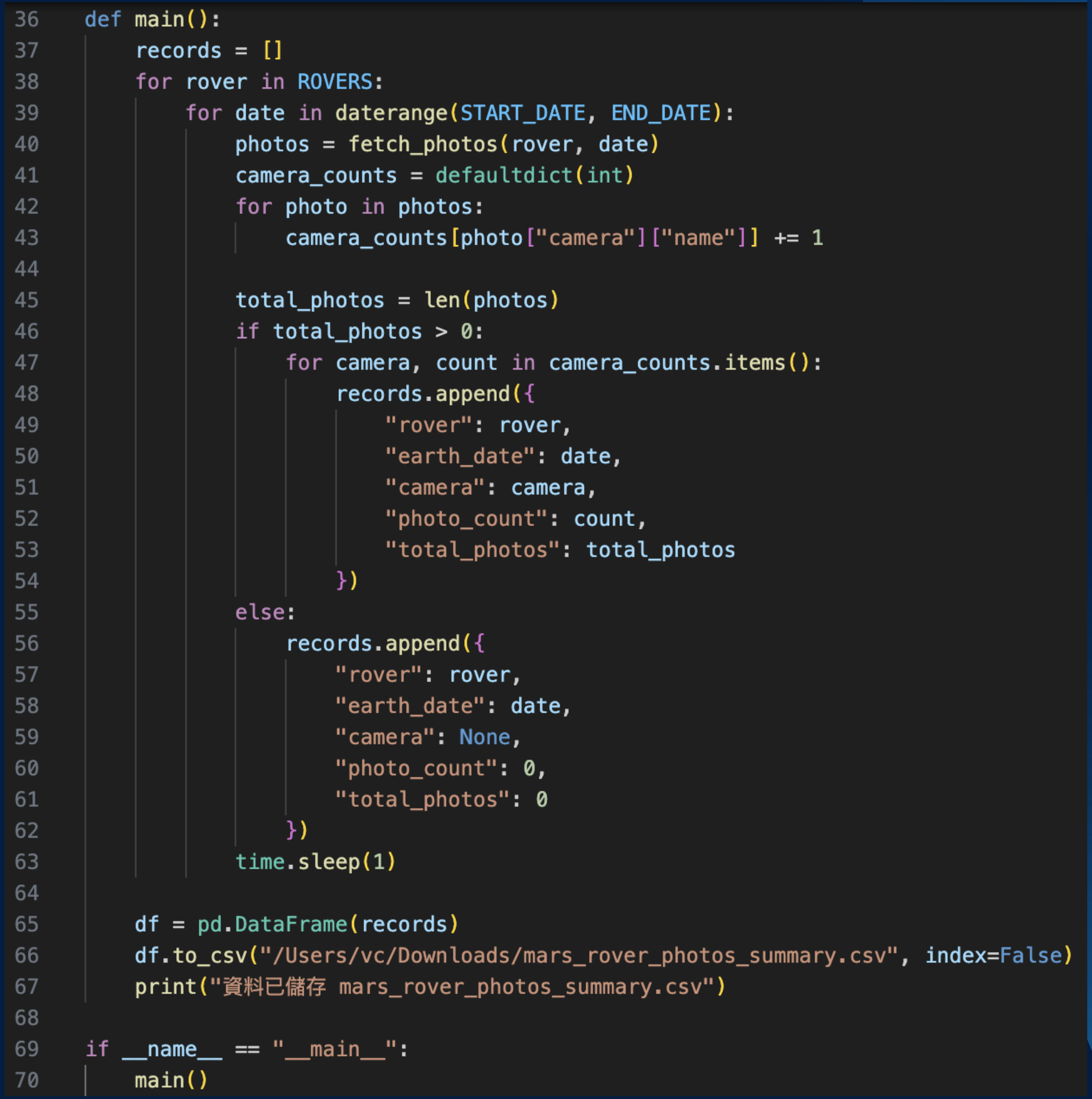
You are receiving this email to confirm the creation of an API key. If you did not request this, please disregard this email. Your API key for chueh28121@gmail.com is:

```
yYUuOKV62xS5XflsoRvC6jIiLZy5WpWnb7OqbVPe
```

You can start using this key to make web service requests by referring to the relevant agency's API documentation. This API key is for your use and should not be shared.



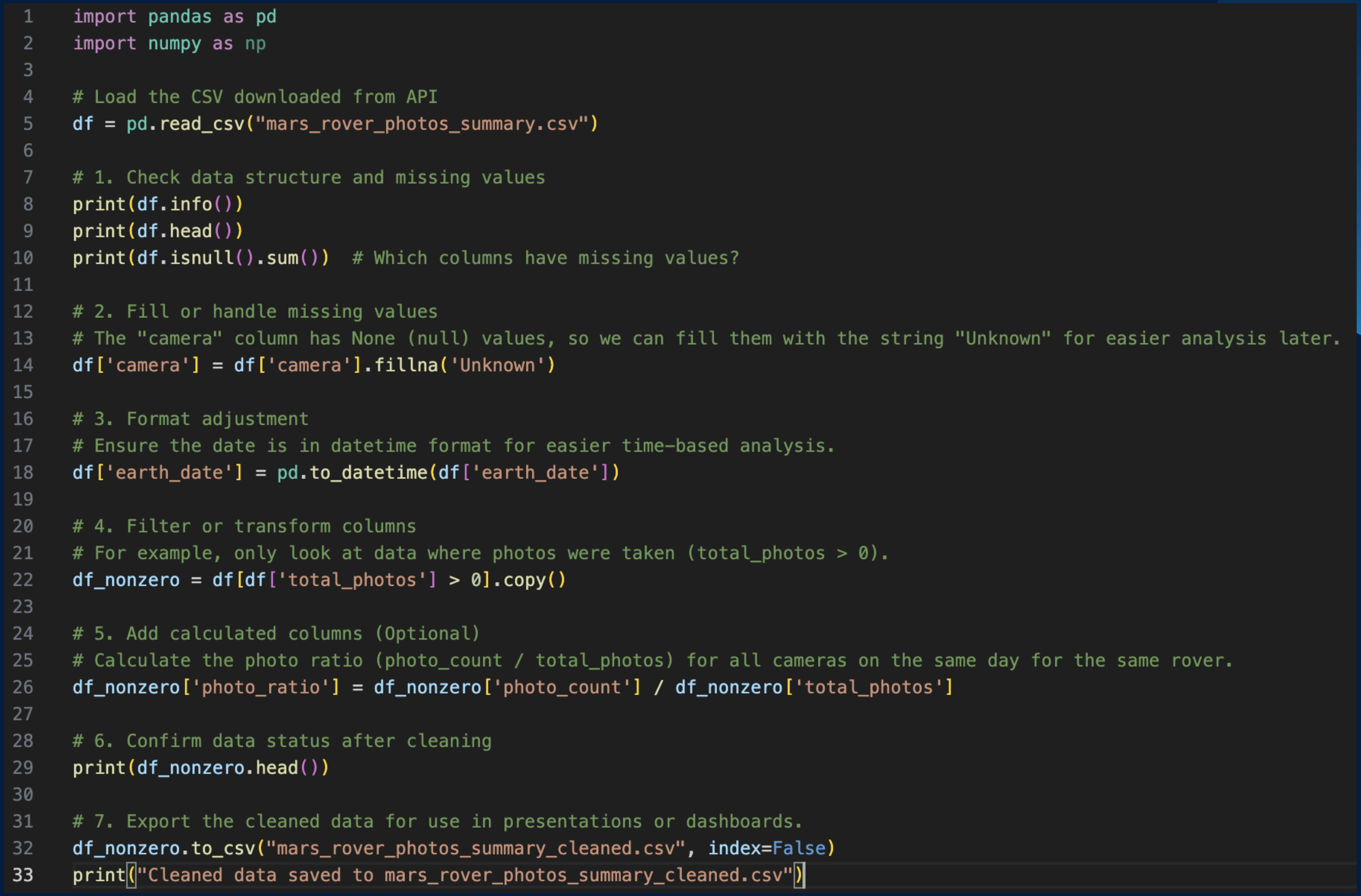
```python
import requests
import pandas as pd
from collections import defaultdict
import time

API_KEY = "yYUuOKV62xS5XflsoRvC6jIiLZy5WpWnb7OqbVPe"  # My NASA API Key
ROVERS = ["curiosity", "perseverance"]
START_DATE = "2023-01-01"
END_DATE = "2023-01-31"  # Collect data of Junuary 2023

def daterange(start_date, end_date):
    from datetime import datetime, timedelta
    start = datetime.strptime(start_date, "%Y-%m-%d")
    end = datetime.strptime(end_date, "%Y-%m-%d")
    delta = timedelta(days=1)
    current = start
    while current <= end:
        yield current.strftime("%Y-%m-%d")
        current += delta

def fetch_photos(rover, date):
    url = f"https://api.nasa.gov/mars-photos/api/v1/rovers/{rover}/photos"
    params = {
        "earth_date": date,
        "api_key": API_KEY
    }
    response = requests.get(url, params=params)
    if response.status_code == 200:
        photos = response.json().get("photos", [])
        print(f"{rover} {date} 照片數量: {len(photos)}")
        return photos
    else:
        print(f"API錯誤：{response.status_code} on {rover} {date}")
        return []

```



```python
def main():
    records = []
    for rover in ROVERS:
        for date in daterange(START_DATE, END_DATE):
            photos = fetch_photos(rover, date)
            camera_counts = defaultdict(int)
            for photo in photos:
                camera_counts[photo["camera"]["name"]] += 1

            total_photos = len(photos)
            if total_photos > 0:
                for camera, count in camera_counts.items():
                    records.append({
                        "rover": rover,
                        "earth_date": date,
                        "camera": camera,
                        "photo_count": count,
                        "total_photos": total_photos
                    })
            else:
                records.append({
                    "rover": rover,
                    "earth_date": date,
                    "camera": None,
                    "photo_count": 0,
                    "total_photos": 0
                })
            time.sleep(1)

    df = pd.DataFrame(records)
    df.to_csv("/Users/vc/Downloads/mars_rover_photos_summary.csv", index=False)
    print("資料已儲存 mars_rover_photos_summary.csv")

if __name__ == "__main__":
    main()
```

mars_rover_photos_summary.csv CSV 文件

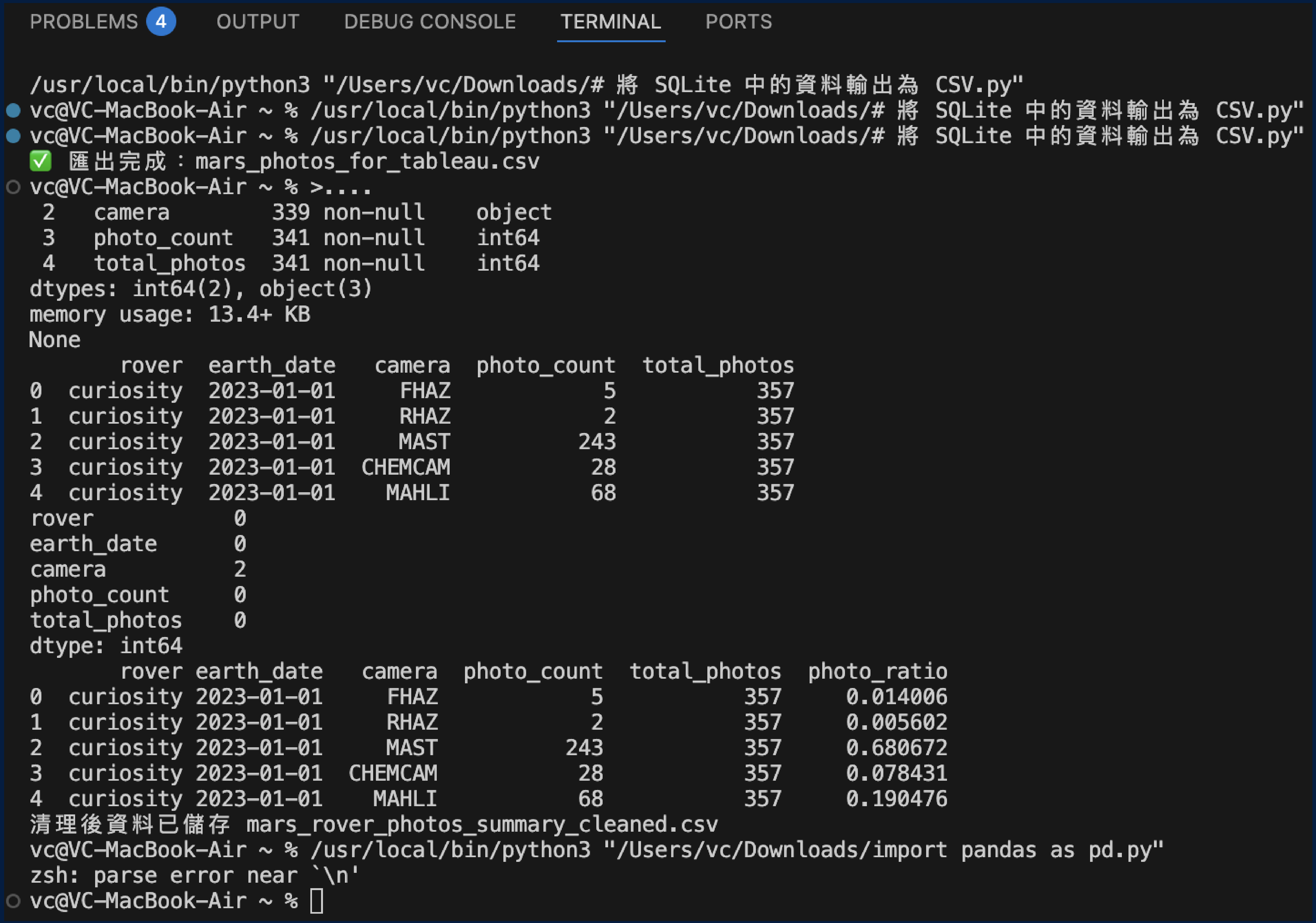
03 Data Cleaning & Preparation



```python
import pandas as pd
import numpy as np

# Load the CSV downloaded from API
df = pd.read_csv("mars_rover_photos_summary.csv")

# 1. Check data structure and missing values
print(df.info())
print(df.head())
print(df.isnull().sum())  # Which columns have missing values?

# 2. Fill or handle missing values
# The "camera" column has None (null) values, so we can fill them with the string "Unknown" for easier analysis later.
df['camera'] = df['camera'].fillna('Unknown')

# 3. Format adjustment
# Ensure the date is in datetime format for easier time-based analysis.
df['earth_date'] = pd.to_datetime(df['earth_date'])

# 4. Filter or transform columns
# For example, only look at data where photos were taken (total_photos > 0).
df_nonzero = df[df['total_photos'] > 0].copy()

# 5. Add calculated columns (Optional)
# Calculate the photo ratio (photo_count / total_photos) for all cameras on the same day for the same rover.
df_nonzero['photo_ratio'] = df_nonzero['photo_count'] / df_nonzero['total_photos']

# 6. Confirm data status after cleaning
print(df_nonzero.head())

# 7. Export the cleaned data for use in presentations or dashboards.
df_nonzero.to_csv("mars_rover_photos_summary_cleaned.csv", index=False)
print("Cleaned data saved to mars_rover_photos_summary_cleaned.csv")
```

mars_rover_photos_summary_cleaned.csv CSV 文件

03 Data Cleaning & Preparation



```
/usr/local/bin/python3 "/Users/vc/Downloads/# 將 SQLite 中的資料輸出為 CSV.py"
● vc@VC-MacBook-Air ~ % /usr/local/bin/python3 "/Users/vc/Downloads/# 將 SQLite 中的資料輸出為 CSV.py"
● vc@VC-MacBook-Air ~ % /usr/local/bin/python3 "/Users/vc/Downloads/# 將 SQLite 中的資料輸出為 CSV.py"
✅ 匯出完成：mars_photos_for_tableau.csv
○ vc@VC-MacBook-Air ~ % >....
  2   camera        339 non-null    object
  3   photo_count   341 non-null    int64
  4   total_photos  341 non-null    int64
dtypes: int64(2), object(3)
memory usage: 13.4+ KB
None
       rover  earth_date   camera  photo_count  total_photos
0  curiosity  2023-01-01     FHAZ            5           357
1  curiosity  2023-01-01     RHAZ            2           357
2  curiosity  2023-01-01     MAST          243           357
3  curiosity  2023-01-01  CHEMCAM           28           357
4  curiosity  2023-01-01    MAHLI           68           357
rover           0
earth_date      0
camera          2
photo_count     0
total_photos    0
dtype: int64
       rover earth_date   camera  photo_count  total_photos  photo_ratio
0  curiosity 2023-01-01     FHAZ            5           357     0.014006
1  curiosity 2023-01-01     RHAZ            2           357     0.005602
2  curiosity 2023-01-01     MAST          243           357     0.680672
3  curiosity 2023-01-01  CHEMCAM           28           357     0.078431
4  curiosity 2023-01-01    MAHLI           68           357     0.190476
清理後資料已儲存 mars_rover_photos_summary_cleaned.csv
vc@VC-MacBook-Air ~ % /usr/local/bin/python3 "/Users/vc/Downloads/import pandas as pd.py"
zsh: parse error near `\n'
○ vc@VC-MacBook-Air ~ % 
```

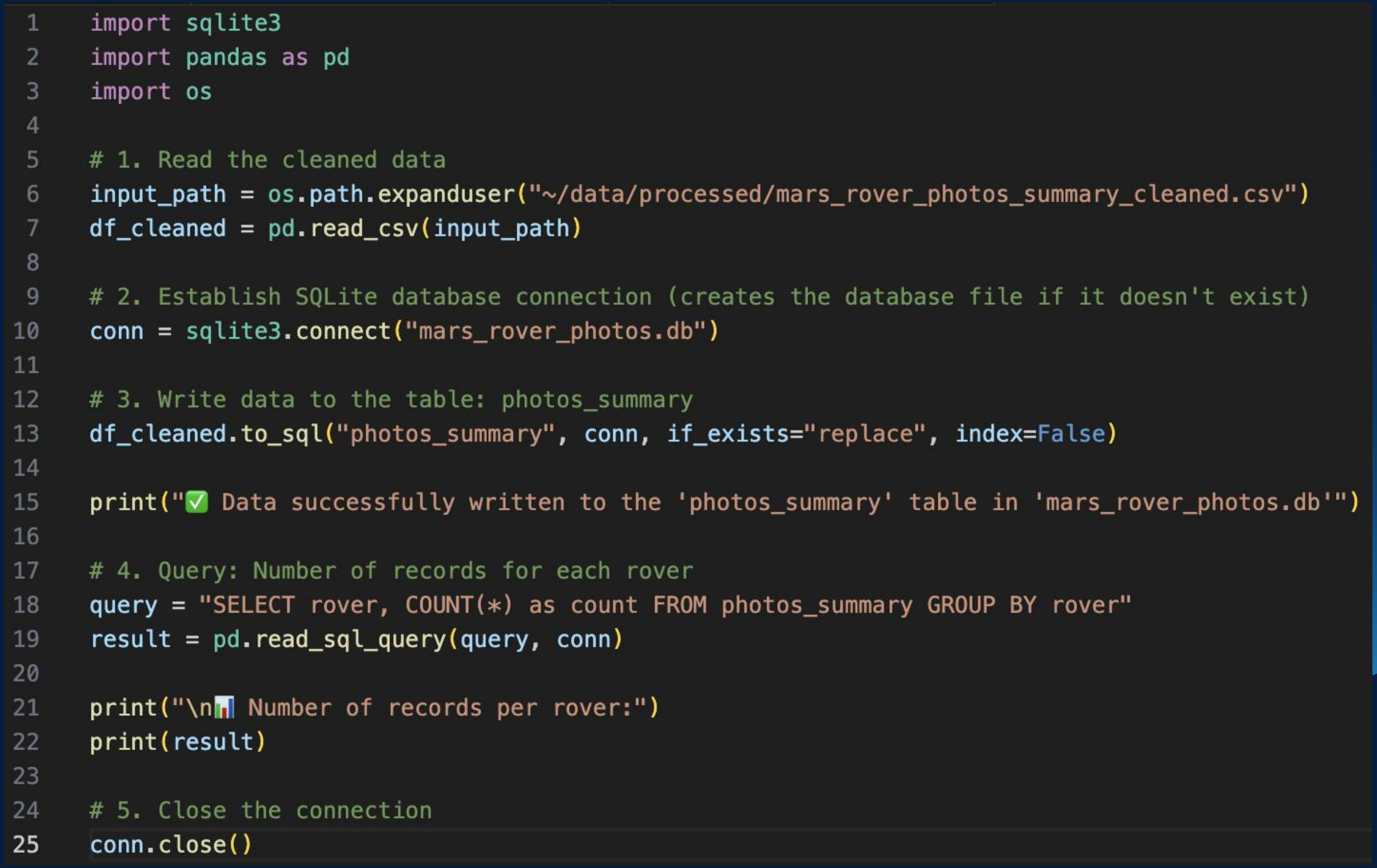
Filled the 2 missing values in the camera column using fillna()

The earth_date column has been correctly converted to datetime format.

Calculated the photo ratio, which represents the proportion of photos taken by each camera relative to the total photos

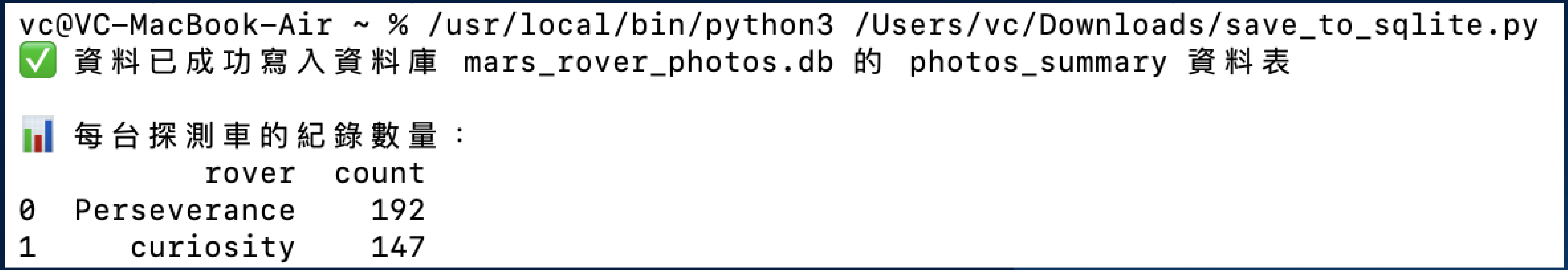
Saved cleaned data to mars_rover_photos_summary_cleaned.csv.

04 Data Storage



```
import sqlite3
import pandas as pd
import os

# 1. Read the cleaned data
input_path = os.path.expanduser("~/data/processed/mars_rover_photos_summary_cleaned.csv")
df_cleaned = pd.read_csv(input_path)

# 2. Establish SQLite database connection (creates the database file if it doesn't exist)
conn = sqlite3.connect("mars_rover_photos.db")

# 3. Write data to the table: photos_summary
df_cleaned.to_sql("photos_summary", conn, if_exists="replace", index=False)

print("✅ Data successfully written to the 'photos_summary' table in 'mars_rover_photos.db'")

# 4. Query: Number of records for each rover
query = "SELECT rover, COUNT(*) as count FROM photos_summary GROUP BY rover"
result = pd.read_sql_query(query, conn)

print("\n📊 Number of records per rover:")
print(result)

# 5. Close the connection
conn.close()
```

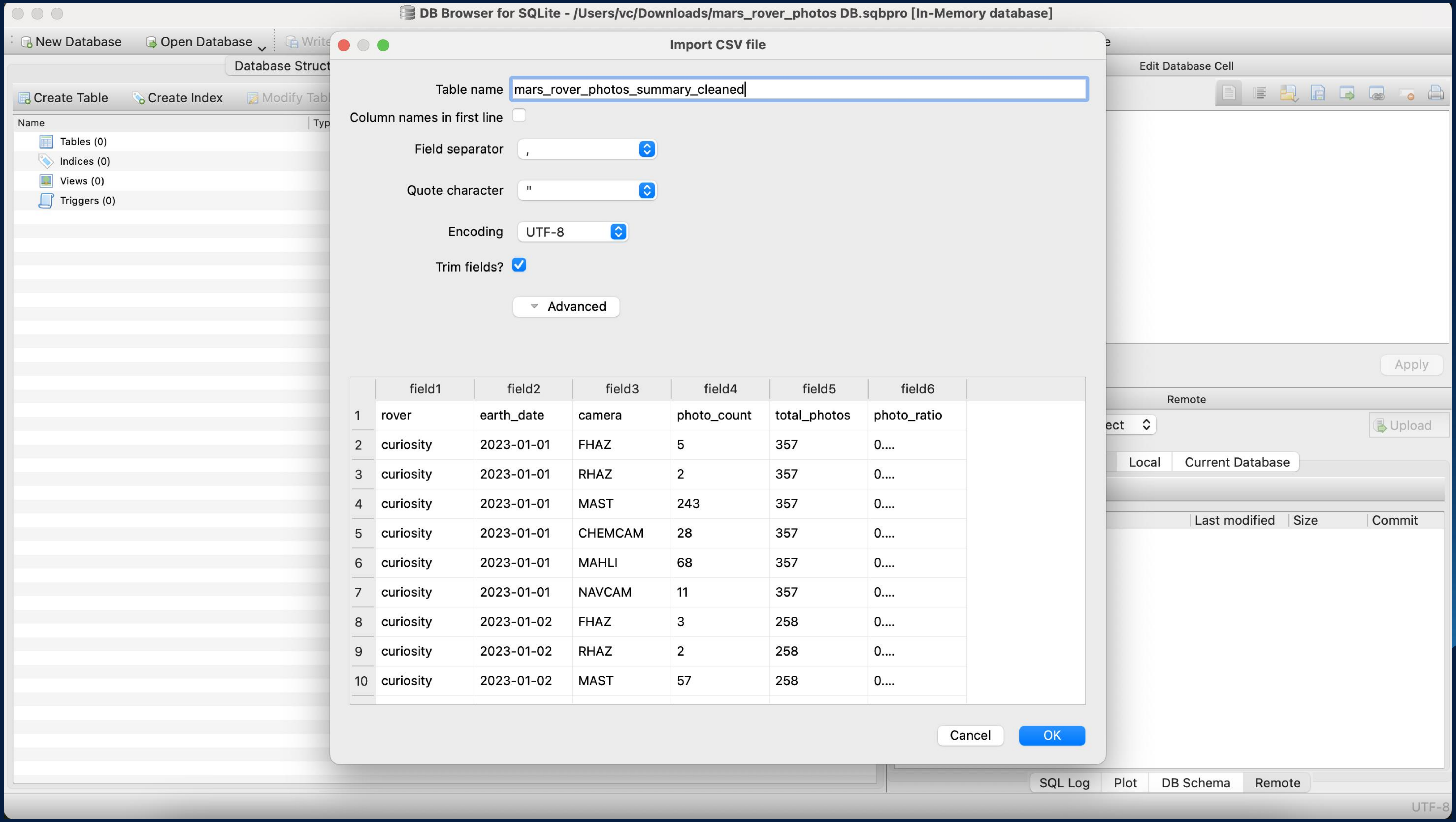
04 Data Storage



```
vc@VC-MacBook-Air ~ % /usr/local/bin/python3 /Users/vc/Downloads/save_to_sqlite.py
✅ 資料已成功寫入資料庫 mars_rover_photos.db 的 photos_summary 資料表

📊 每台探測車的紀錄數量：
          rover  count
0  Perseverance    192
1     curiosity    147
```

mars_rover_photos.db 文件

04 Data Storage


DB Browser for SQLite - /Users/vc/Downloads/mars_rover_photos DB.sqbpro [In-Memory database]

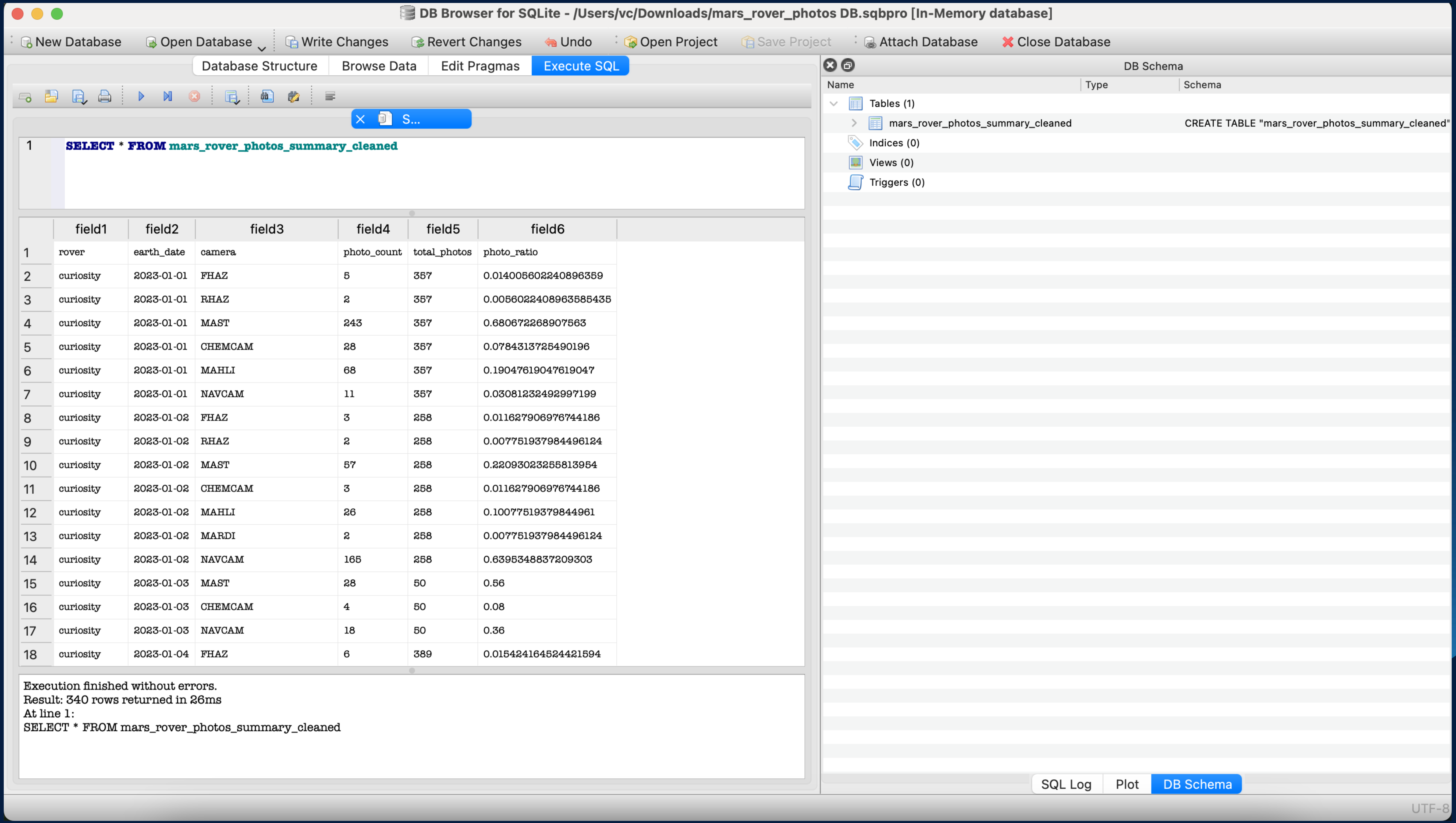
Import CSV file

Table name: mars_rover_photos_summary_cleaned
Column names in first line: ☐
Field separator: ,
Quote character: "
Encoding: UTF-8
Trim fields? ✓

	field1	field2	field3	field4	field5	field6
1	rover	earth_date	camera	photo_count	total_photos	photo_ratio
2	curiosity	2023-01-01	FHAZ	5	357	0....
3	curiosity	2023-01-01	RHAZ	2	357	0....
4	curiosity	2023-01-01	MAST	243	357	0....
5	curiosity	2023-01-01	CHEMCAM	28	357	0....
6	curiosity	2023-01-01	MAHLI	68	357	0....
7	curiosity	2023-01-01	NAVCAM	11	357	0....
8	curiosity	2023-01-02	FHAZ	3	258	0....
9	curiosity	2023-01-02	RHAZ	2	258	0....
10	curiosity	2023-01-02	MAST	57	258	0....

Tables (0)
Indices (0)
Views (0)
Triggers (0)

04 Data Storage

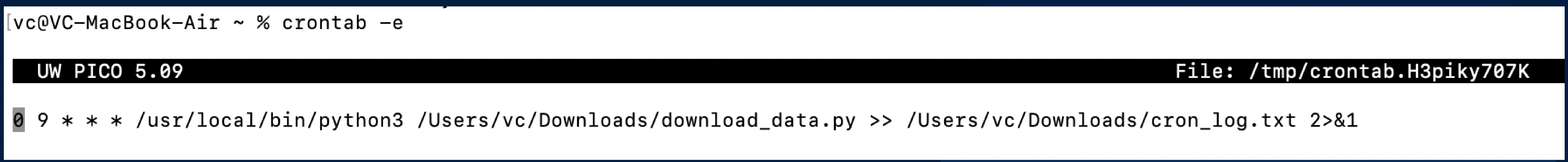


```
SELECT * FROM mars_rover_photos_summary_cleaned
```

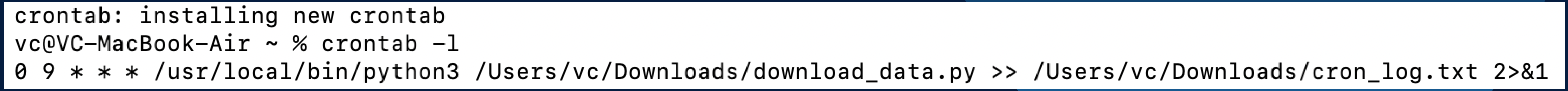
	field1	field2	field3	field4	field5	field6
1	rover	earth_date	camera	photo_count	total_photos	photo_ratio
2	curiosity	2023-01-01	FHAZ	5	357	0.014005602240896359
3	curiosity	2023-01-01	RHAZ	2	357	0.0056022408963585435
4	curiosity	2023-01-01	MAST	243	357	0.680672268907563
5	curiosity	2023-01-01	CHEMCAM	28	357	0.0784313725490196
6	curiosity	2023-01-01	MAHLI	68	357	0.19047619047619047
7	curiosity	2023-01-01	NAVCAM	11	357	0.03081232492997199
8	curiosity	2023-01-02	FHAZ	3	258	0.011627906976744186
9	curiosity	2023-01-02	RHAZ	2	258	0.007751937984496124
10	curiosity	2023-01-02	MAST	57	258	0.22093023255813954
11	curiosity	2023-01-02	CHEMCAM	3	258	0.011627906976744186
12	curiosity	2023-01-02	MAHLI	26	258	0.10077519379844961
13	curiosity	2023-01-02	MARDI	2	258	0.007751937984496124
14	curiosity	2023-01-02	NAVCAM	165	258	0.6395348837209303
15	curiosity	2023-01-03	MAST	28	50	0.56
16	curiosity	2023-01-03	CHEMCAM	4	50	0.08
17	curiosity	2023-01-03	NAVCAM	18	50	0.36
18	curiosity	2023-01-04	FHAZ	6	389	0.015424164524421594

```
Execution finished without errors.
Result: 340 rows returned in 26ms
At line 1:
SELECT * FROM mars_rover_photos_summary_cleaned
```

05 Workflow Orchestration

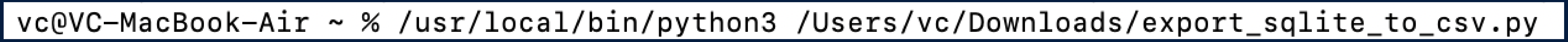


```
[vc@VC-MacBook-Air ~ % crontab -e

  UW PICO 5.09                                                                                         File: /tmp/crontab.H3piky707K

0 9 * * * /usr/local/bin/python3 /Users/vc/Downloads/download_data.py >> /Users/vc/Downloads/cron_log.txt 2>&1
```

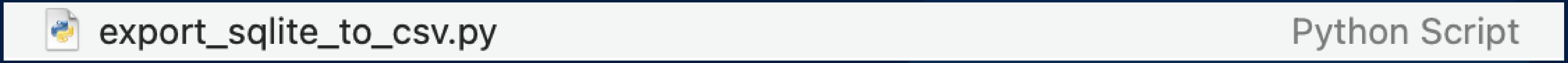


```
crontab: installing new crontab
vc@VC-MacBook-Air ~ % crontab -l
0 9 * * * /usr/local/bin/python3 /Users/vc/Downloads/download_data.py >> /Users/vc/Downloads/cron_log.txt 2>&1
```

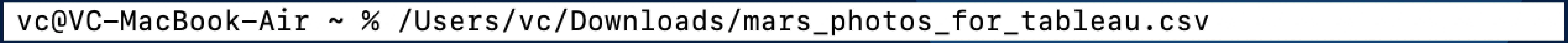
06 Data Analysis & Visualization



```
vc@VC-MacBook-Air ~ % /usr/local/bin/python3 /Users/vc/Downloads/export_sqlite_to_csv.py
```



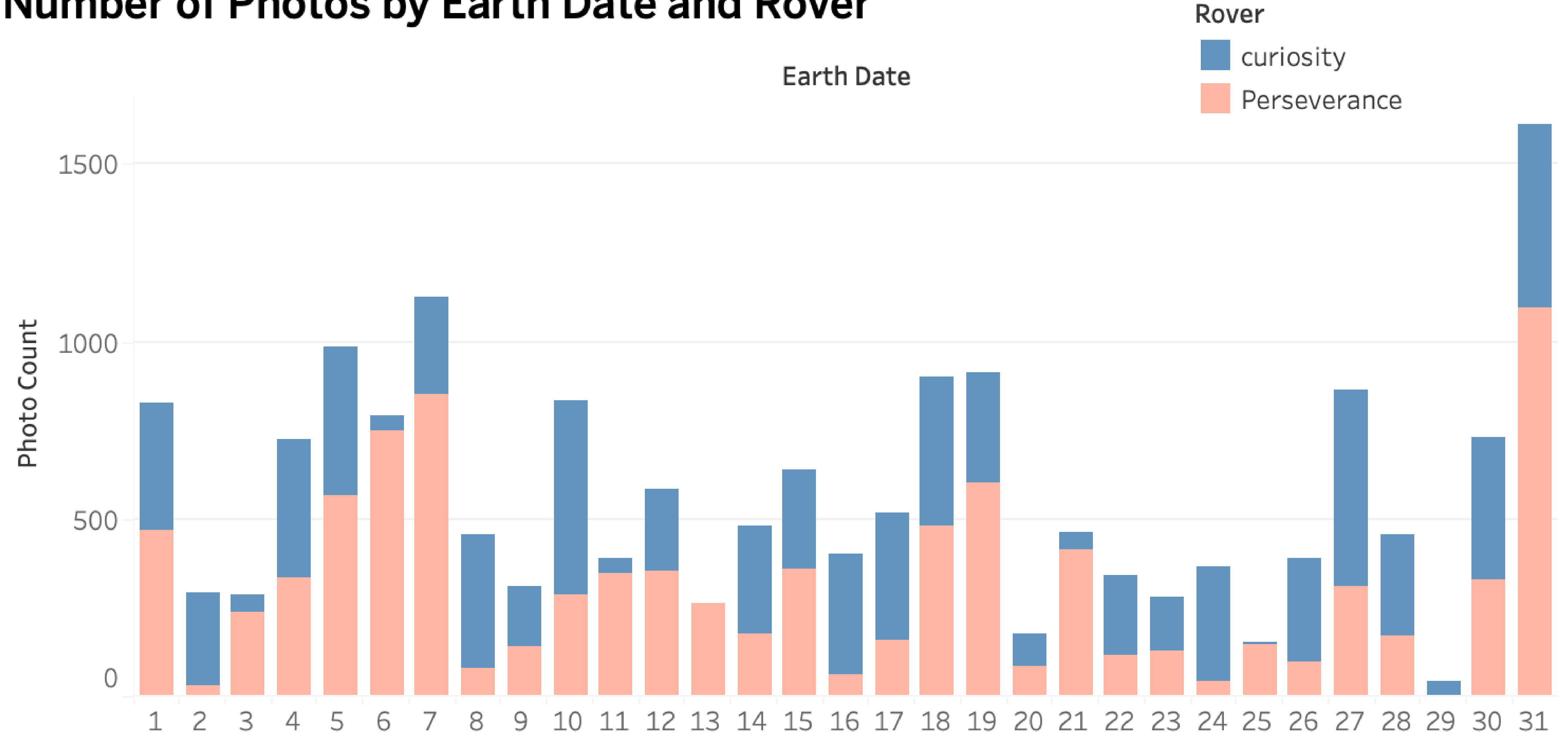
export_sqlite_to_csv.py Python Script



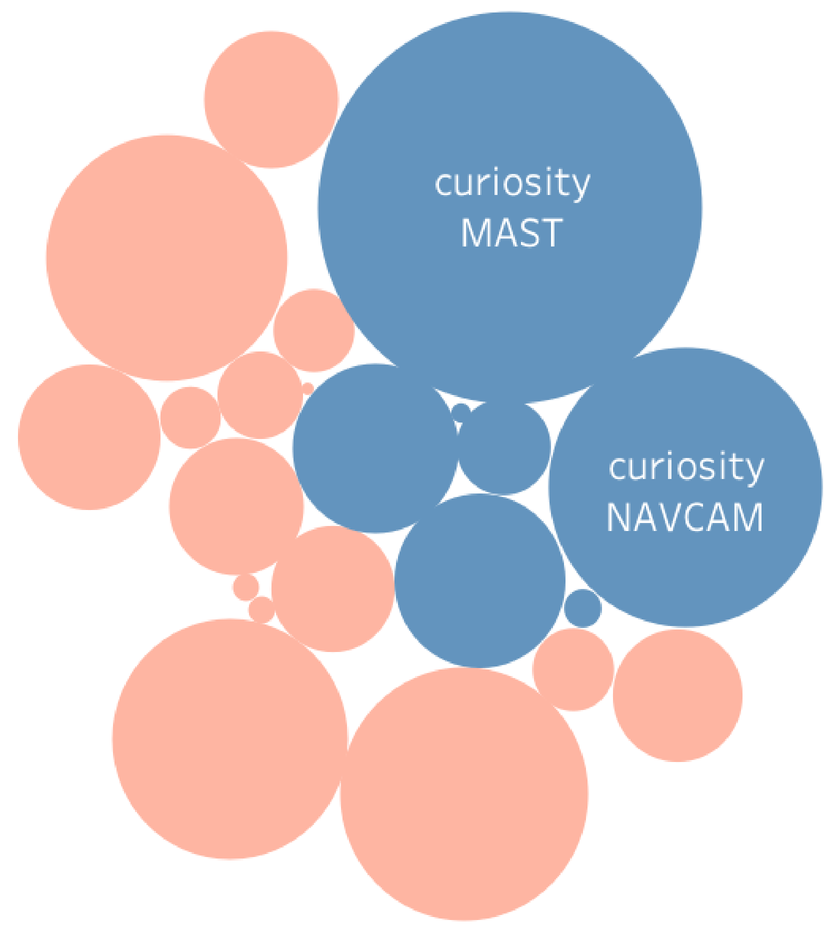
```
vc@VC-MacBook-Air ~ % /Users/vc/Downloads/mars_photos_for_tableau.csv
```

mars_photos_for_tableau.csv CSV 文件

Number of Photos by Earth Date and Rover



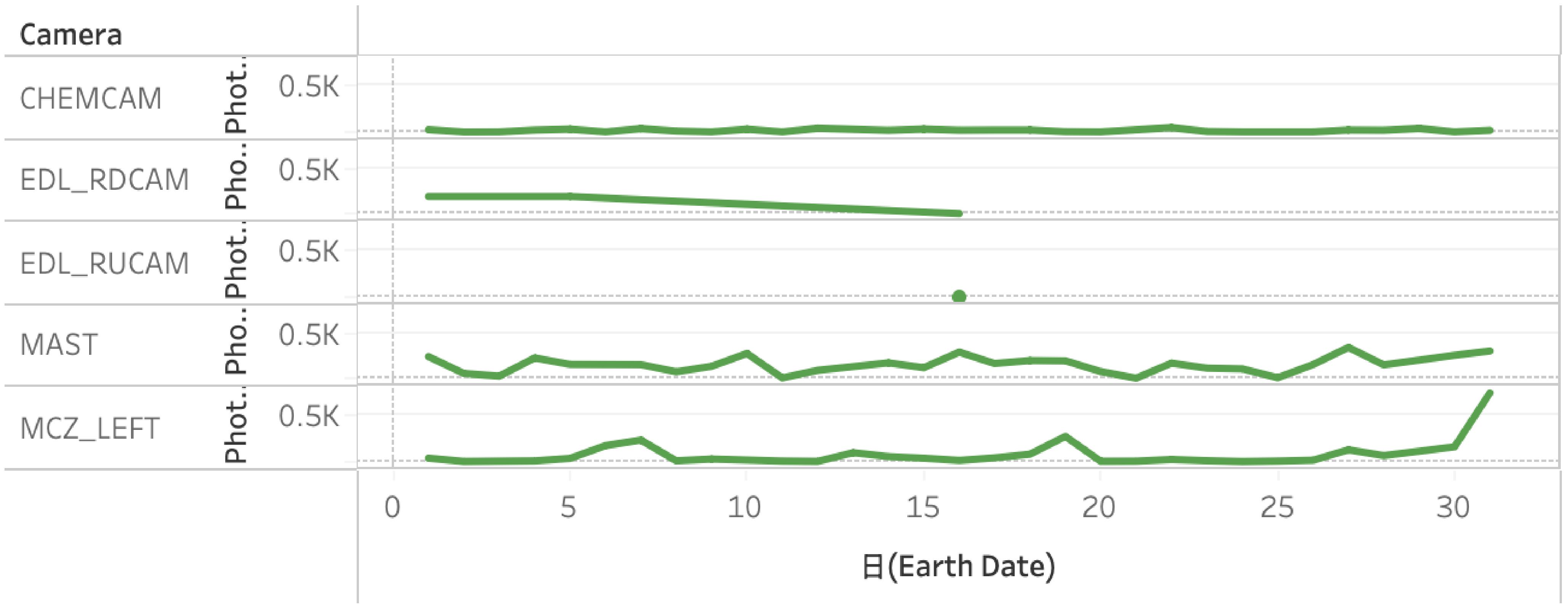
Proportion of Photos Taken by Each Camera



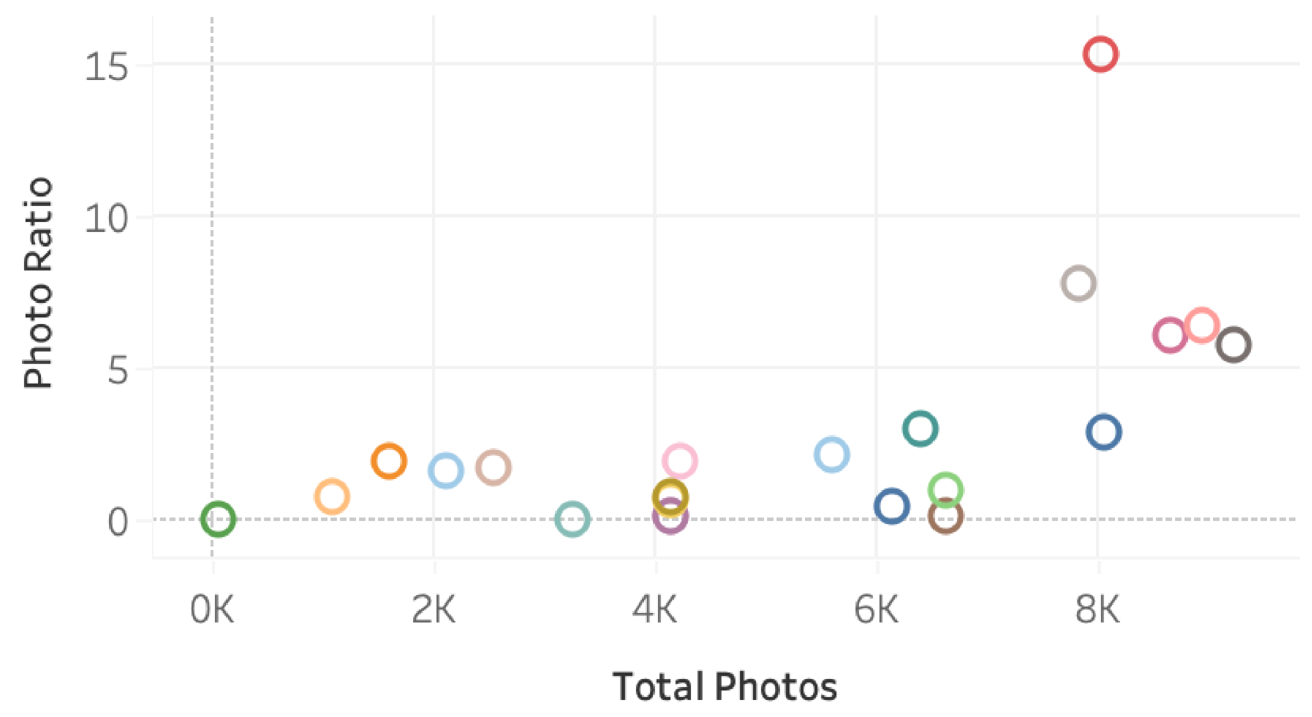
Camera

- 🟦 CHEMCAM
- 🟦 EDL_PUCA..
- 🟧 EDL_PUCA..
- 🟧 EDL_RDCAM
- 🟩 EDL_RUCAM
- 🟩 FHAZ
- 🟨 FRONT_HA..
- 🟨 FRONT_HA..
- 🟩 MAHLI
- 🟩 MARDI
- 🟥 MAST
- 🟥 MCZ_LEFT
- ⬛ MCZ_RIGHT
- ⬜ NAVCAM
- 🟪 NAVCAM_L..
- 🟪 NAVCAM_R..
- 🟪 REAR_HAZ..
- 🟪 REAR_HAZ..
- 🟫 RHAZ

Camera Performance Over Time



Daily Photo Count vs. Camera Ratio Correlation



THANK YOU

GitHub Link:
https://github.com/VictoriaChueh/Business-Intelligence-Final_ETL-Project.git