Module 12 Challenge

Deliverable 2: Scrape and Analyze Mars Weather Data

In [6]:

```
# Import relevant libraries
from splinter import Browser
from bs4 import BeautifulSoup
import matplotlib.pyplot as plt
import pandas as pd
```

In [7]:

```
browser = Browser('chrome')
```

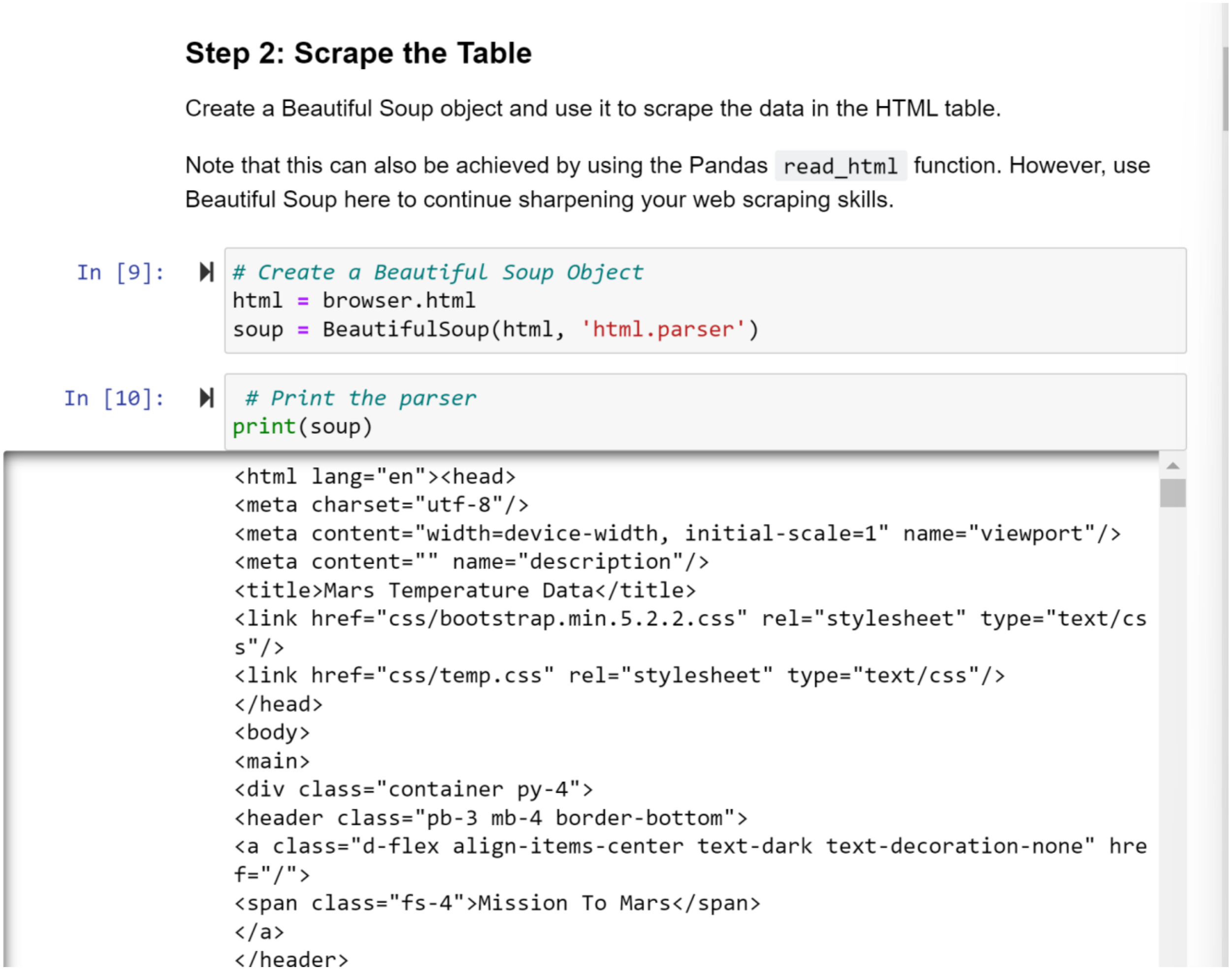
Step 1: Visit the Website

Use automated browsing to visit the Mars Temperature Data Site. Inspect the page to identify which elements to scrape.

> **Hint** To identify which elements to scrape, you might want to inspect the page by using Chrome DevTools to discover whether the table contains usable classes.

In [8]:

```
# Visit the website
# https://static.bc-edx.com/data/web/mars_facts/temperature.html
url = "https://static.bc-edx.com/data/web/mars_facts/temperature.html"
browser.visit(url)
```



Step 2: Scrape the Table

Create a Beautiful Soup object and use it to scrape the data in the HTML table.

Note that this can also be achieved by using the Pandas `read_html` function. However, use Beautiful Soup here to continue sharpening your web scraping skills.

In [9]:

```python
# Create a Beautiful Soup Object
html = browser.html
soup = BeautifulSoup(html, 'html.parser')
```

In [10]:

```python
 # Print the parser
print(soup)
```

```
<html lang="en"><head>
<meta charset="utf-8"/>
<meta content="width=device-width, initial-scale=1" name="viewport"/>
<meta content="" name="description"/>
<title>Mars Temperature Data</title>
<link href="css/bootstrap.min.5.2.2.css" rel="stylesheet" type="text/cs
s"/>
<link href="css/temp.css" rel="stylesheet" type="text/css"/>
</head>
<body>
<main>
<div class="container py-4">
<header class="pb-3 mb-4 border-bottom">
<a class="d-flex align-items-center text-dark text-decoration-none" hre
f="/">
<span class="fs-4">Mission To Mars</span>
</a>
</header>
```

In [11]:

```
# Extract all rows of data
data = soup.find_all('tr', class_="data-row")
```

Step 3: Store the Data

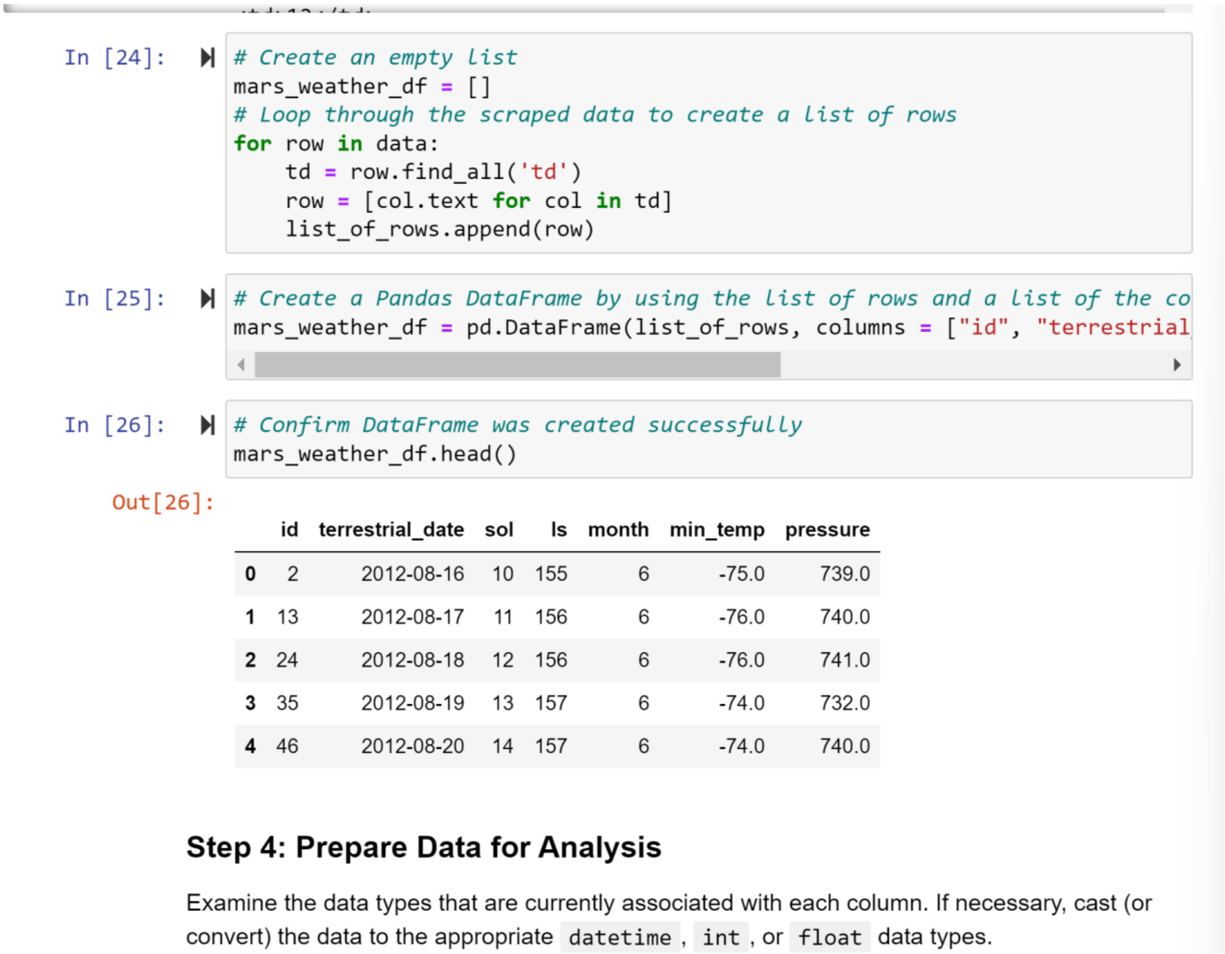
Assemble the scraped data into a Pandas DataFrame. The columns should have the same headings as the table on the website. Here’s an explanation of the column headings:

- `id` : the identification number of a single transmission from the Curiosity rover
- `terrestrial_date` : the date on Earth
- `sol` : the number of elapsed sols (Martian days) since Curiosity landed on Mars
- `ls` : the solar longitude
- `month` : the Martian month
- `min_temp` : the minimum temperature, in Celsius, of a single Martian day (sol)
- `pressure` : The atmospheric pressure at Curiosity's location

In [13]:

```
print(data)
```

```
[<tr class="data-row">
<td>2</td>
<td>2012-08-16</td>
<td>10</td>
<td>155</td>
<td>6</td>
<td>-75.0</td>
<td>739.0</td>
</tr>, <tr class="data-row">
<td>13</td>
<td>2012-08-17</td>
```



In [24]:

```python
# Create an empty list
mars_weather_df = []
# Loop through the scraped data to create a list of rows
for row in data:
    td = row.find_all('td')
    row = [col.text for col in td]
    list_of_rows.append(row)
```

In [25]:

```python
# Create a Pandas DataFrame by using the list of rows and a list of the co
mars_weather_df = pd.DataFrame(list_of_rows, columns = ["id", "terrestrial
```

In [26]:

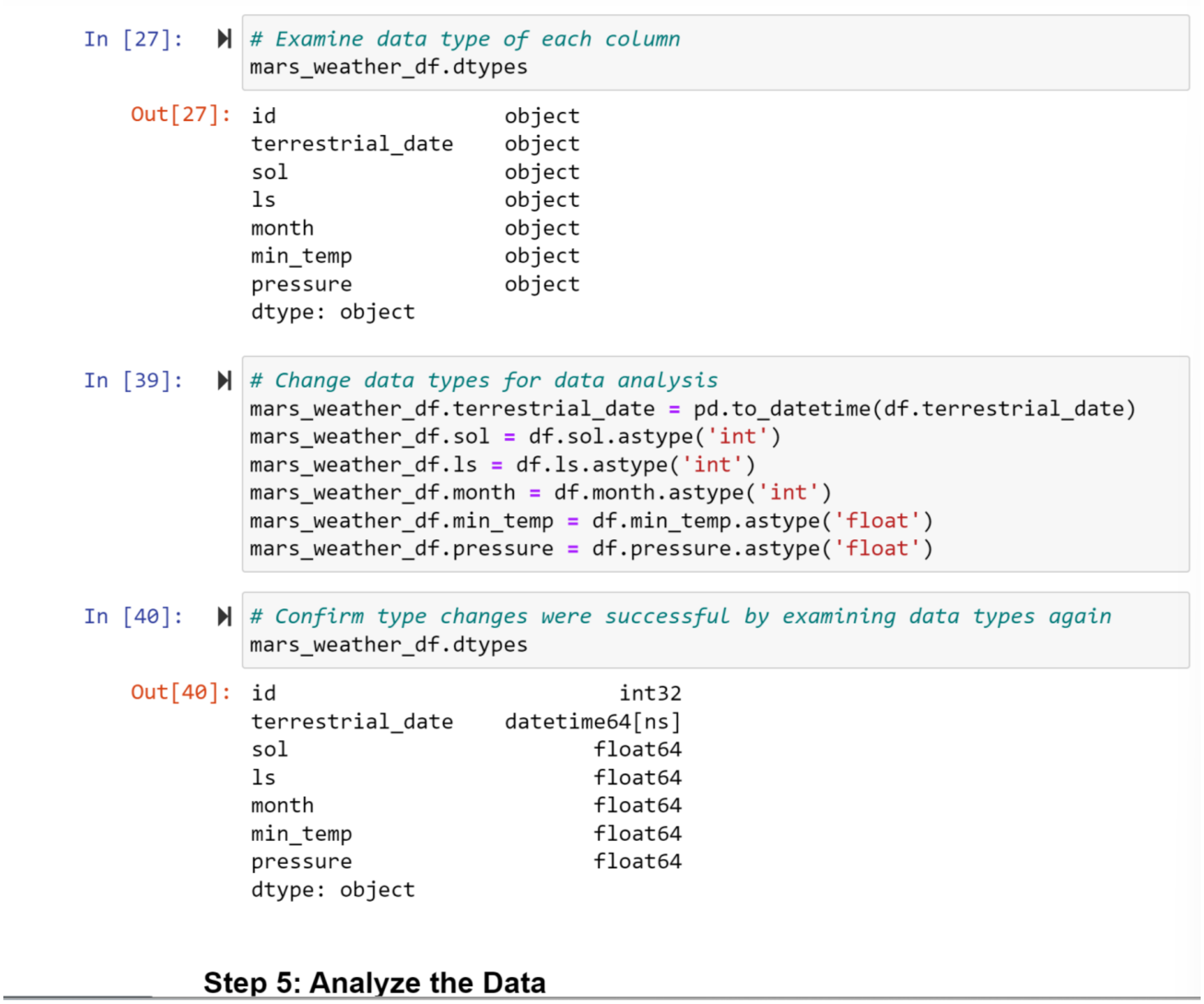
```python
# Confirm DataFrame was created successfully
mars_weather_df.head()
```

Out[26]:

	id	terrestrial_date	sol	ls	month	min_temp	pressure
0	2	2012-08-16	10	155	6	-75.0	739.0
1	13	2012-08-17	11	156	6	-76.0	740.0
2	24	2012-08-18	12	156	6	-76.0	741.0
3	35	2012-08-19	13	157	6	-74.0	732.0
4	46	2012-08-20	14	157	6	-74.0	740.0

Step 4: Prepare Data for Analysis

Examine the data types that are currently associated with each column. If necessary, cast (or convert) the data to the appropriate `datetime`, `int`, or `float` data types.



In [27]:

```
# Examine data type of each column
mars_weather_df.dtypes
```

Out[27]:

```
id                  object
terrestrial_date    object
sol                 object
ls                  object
month               object
min_temp            object
pressure            object
dtype: object
```

In [39]:

```
# Change data types for data analysis
mars_weather_df.terrestrial_date = pd.to_datetime(df.terrestrial_date)
mars_weather_df.sol = df.sol.astype('int')
mars_weather_df.ls = df.ls.astype('int')
mars_weather_df.month = df.month.astype('int')
mars_weather_df.min_temp = df.min_temp.astype('float')
mars_weather_df.pressure = df.pressure.astype('float')
```

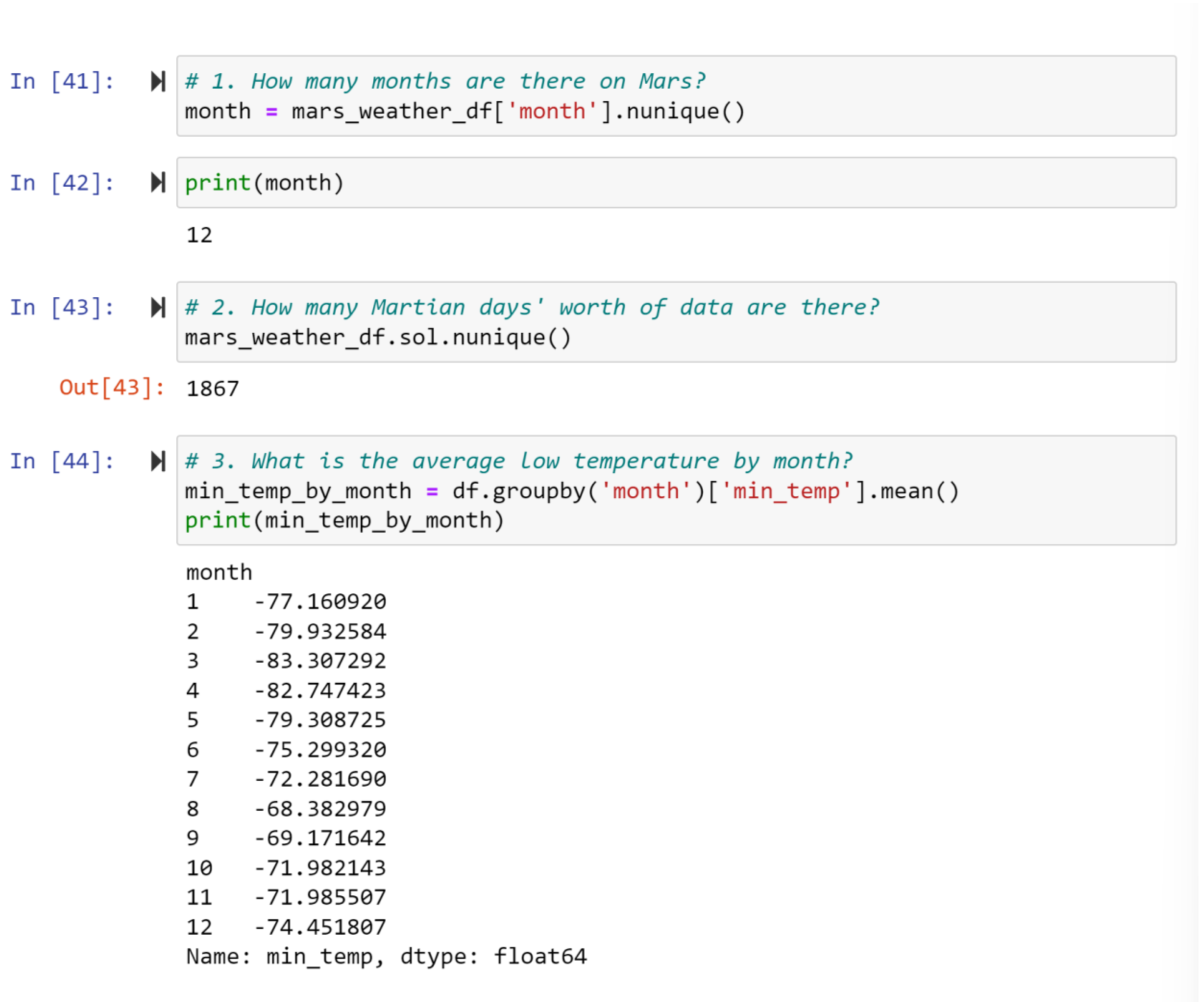
In [40]:

```
# Confirm type changes were successful by examining data types again
mars_weather_df.dtypes
```

Out[40]:

```
id                           int32
terrestrial_date    datetime64[ns]
sol                        float64
ls                         float64
month                      float64
min_temp                   float64
pressure                   float64
dtype: object
```

Step 5: Analyze the Data



In [41]:

```
# 1. How many months are there on Mars?
month = mars_weather_df['month'].nunique()
```

In [42]:

```
print(month)
```

```
12
```

In [43]:

```
# 2. How many Martian days' worth of data are there?
mars_weather_df.sol.nunique()
```

Out[43]:

```
1867
```

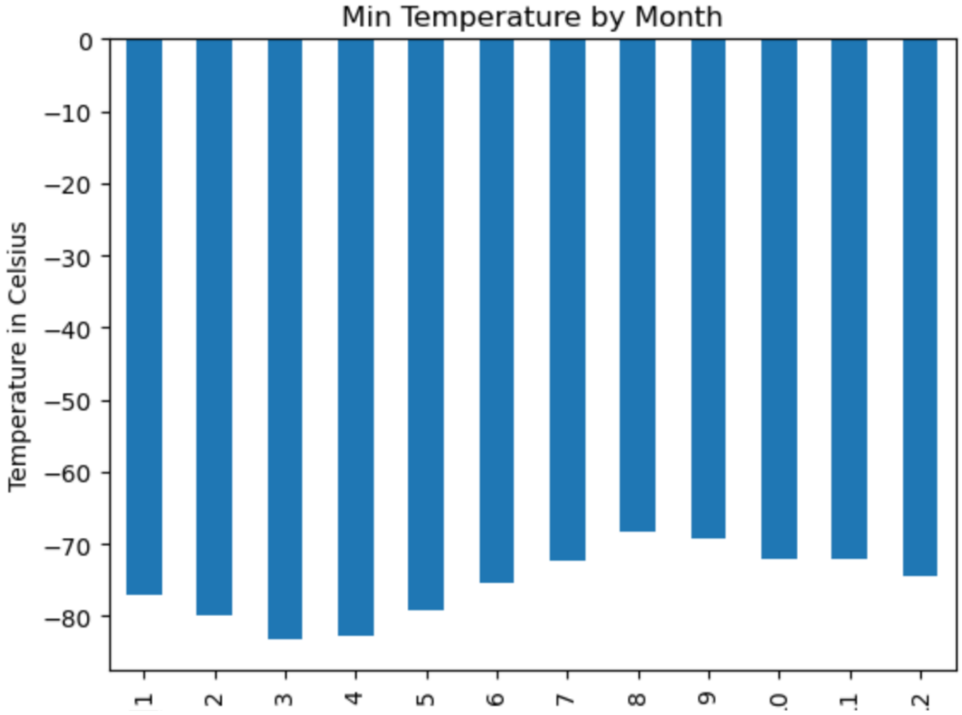
In [44]:

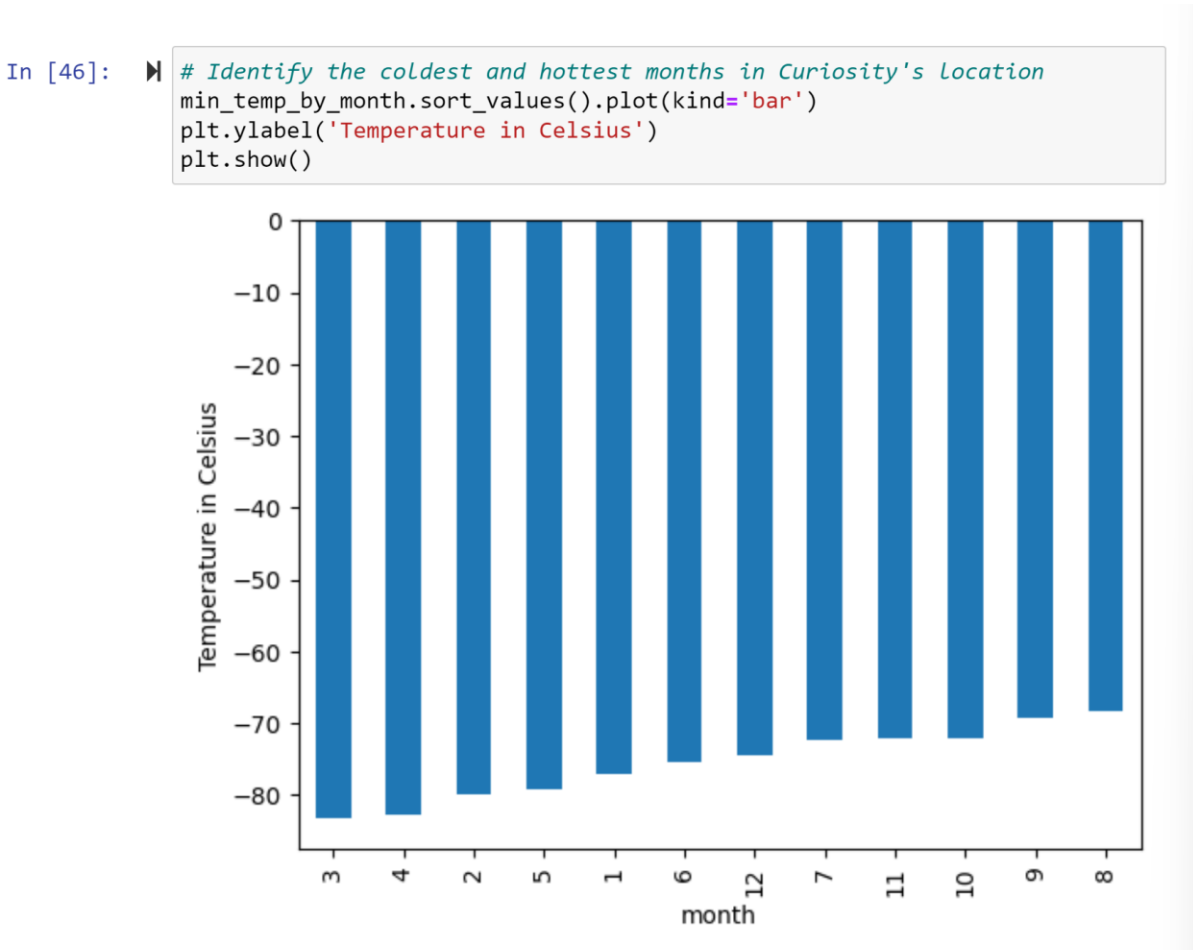
```
# 3. What is the average low temperature by month?
min_temp_by_month = df.groupby('month')['min_temp'].mean()
print(min_temp_by_month)
```

```
month
1    -77.160920
2    -79.932584
3    -83.307292
4    -82.747423
5    -79.308725
6    -75.299320
7    -72.281690
8    -68.382979
9    -69.171642
10   -71.982143
11   -71.985507
12   -74.451807
Name: min_temp, dtype: float64
```

In [50]:

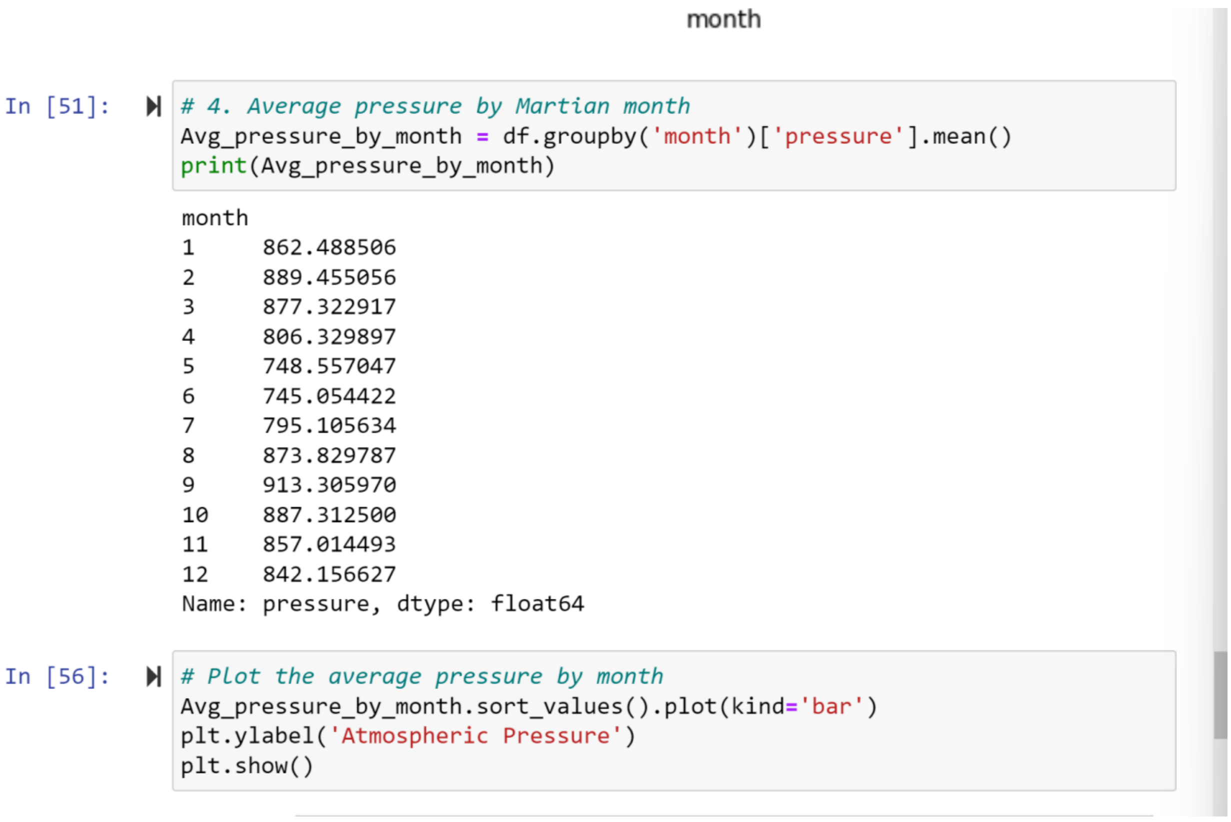
```
# Plot the average temperature by month
min_temp_by_month.plot(kind='bar')
plt.title('Min Temperature by Month')
plt.ylabel('Temperature in Celsius')
plt.show()
```





In [46]:

```
# Identify the coldest and hottest months in Curiosity's location
min_temp_by_month.sort_values().plot(kind='bar')
plt.ylabel('Temperature in Celsius')
plt.show()
```



month

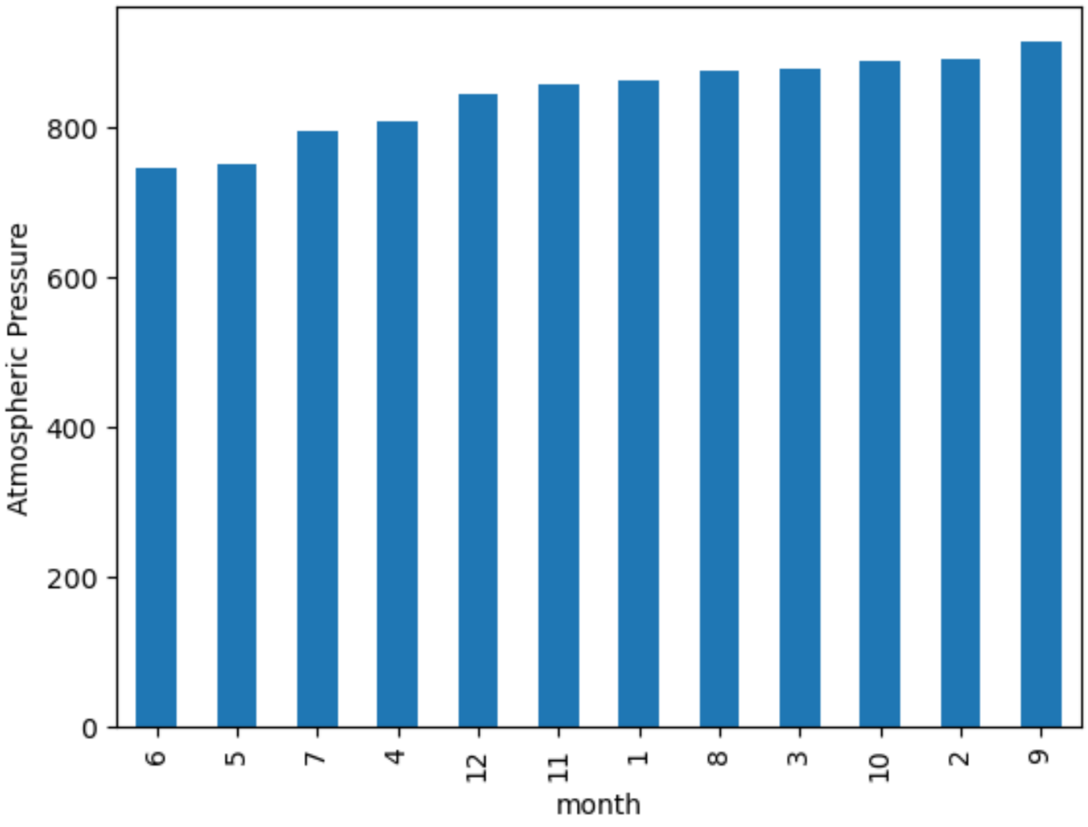
In [51]:

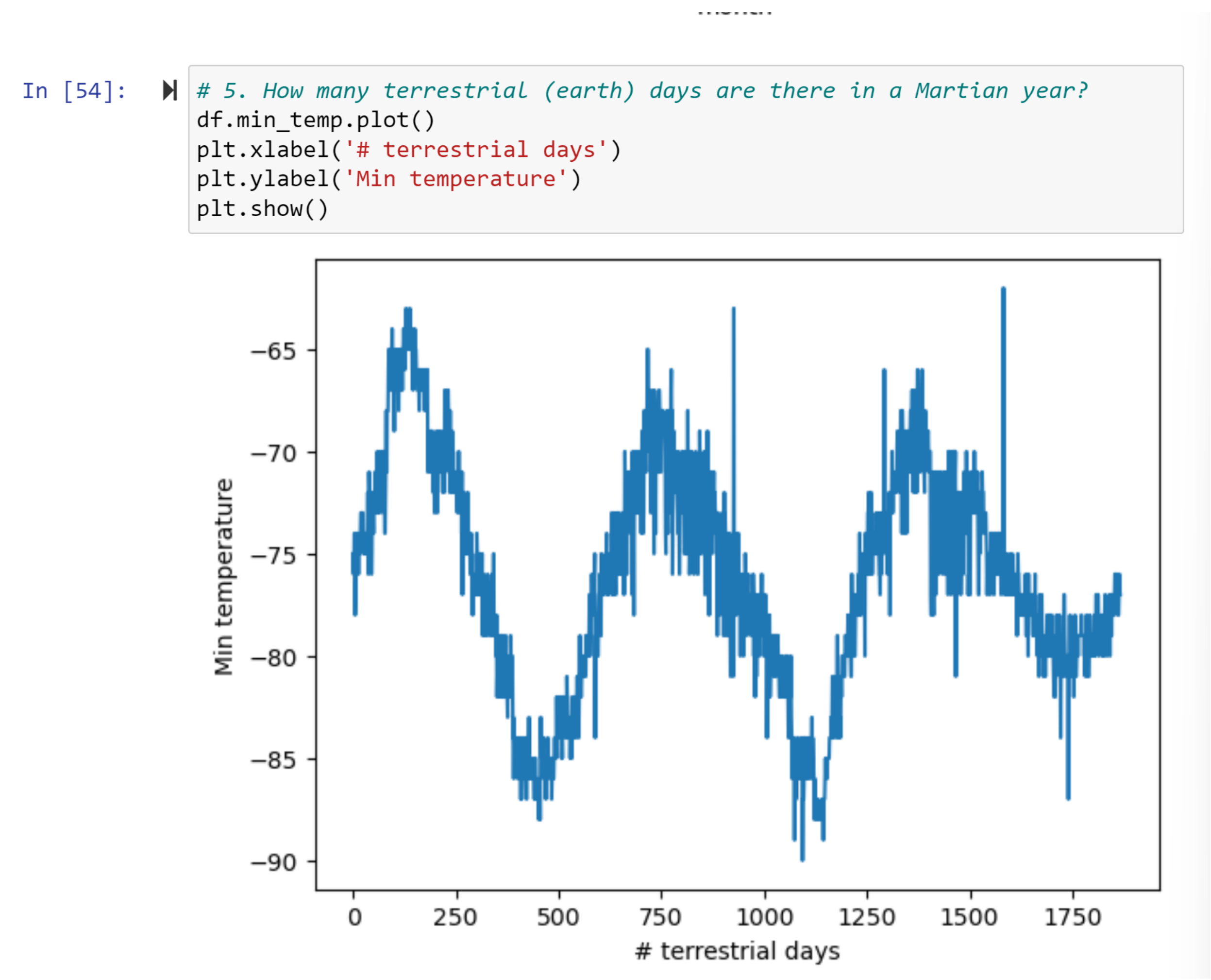
```
# 4. Average pressure by Martian month
Avg_pressure_by_month = df.groupby('month')['pressure'].mean()
print(Avg_pressure_by_month)
```

```
month
1     862.488506
2     889.455056
3     877.322917
4     806.329897
5     748.557047
6     745.054422
7     795.105634
8     873.829787
9     913.305970
10    887.312500
11    857.014493
12    842.156627
Name: pressure, dtype: float64
```

In [56]:

```
# Plot the average pressure by month
Avg_pressure_by_month.sort_values().plot(kind='bar')
plt.ylabel('Atmospheric Pressure')
plt.show()
```





```
# 5. How many terrestrial (earth) days are there in a Martian year?
df.min_temp.plot()
plt.xlabel('# terrestrial days')
plt.ylabel('Min temperature')
plt.show()
```

On average, the third month has the coldest minimum temperature on Mars, and the eighth month is the warmest. But it is always very cold there in human terms!

Atmospheric pressure is, on average, lowest in the sixth month and highest in the ninth.

The distance from peak to peak is roughly 1425-750, or 675 days. A year on Mars appears to be about 675 days from the plot. Internet search confirms that a Mars year is equivalent to 687 earth days.

Step 6: Save the Data

Export the DataFrame to a CSV file.

In [55]:

```
# Write the data to a CSV
df.to_csv('mars_data.csv', index=False)
```

In [21]:

```
browser.quit()
```

In [ ]:

```

```