IST2334 - Web and Network Analytics - Practical #3

Analyse the browser usage data provided which encompasses information from July 2008 to October 2014 using time series and plot the results.

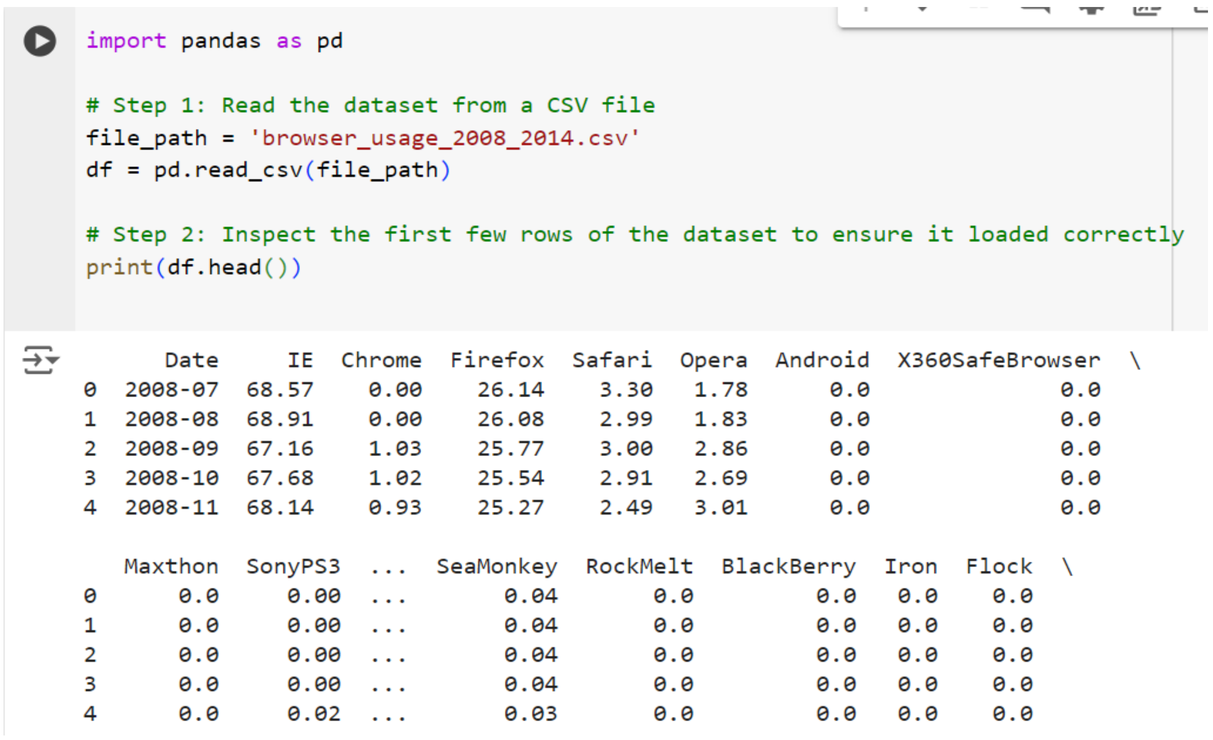
Steps to follow:

1. Read the data from the source file.
2. Define time series data objects.
3. Create a multiple time series object.
4. Plot multiple time series object using standard Python graphics.
5. Save the plot on a pdf file.

1. Read the data from the source file

To read the dataset using Python, you can use the pandas library.

If you don't have the ***pandas*** library installed, you can install it using:
pip install pandas

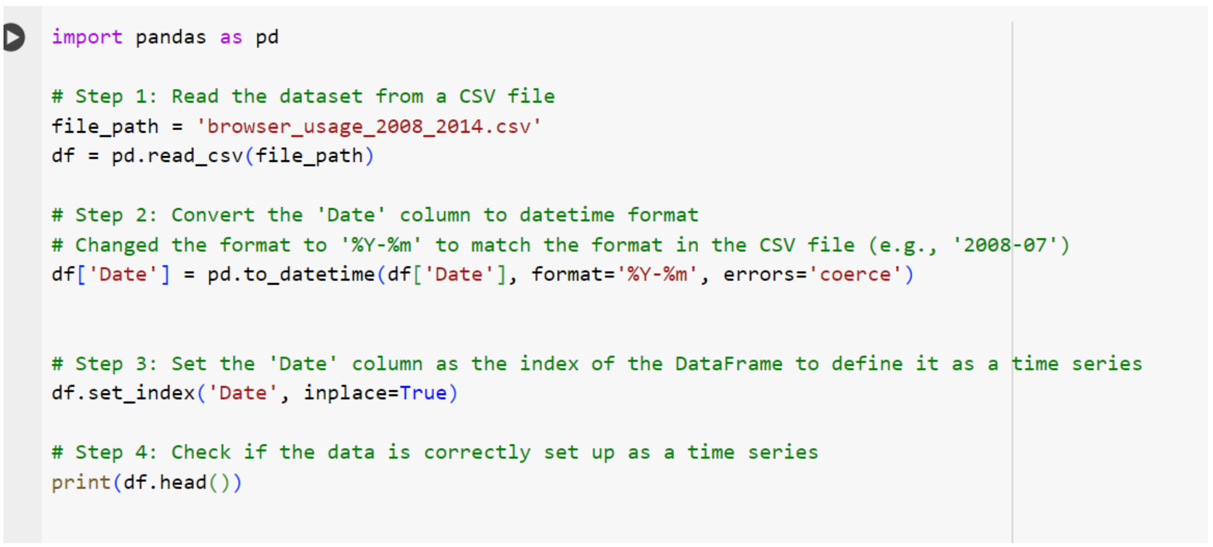


```
import pandas as pd

# Step 1: Read the dataset from a CSV file
file_path = 'browser_usage_2008_2014.csv'
df = pd.read_csv(file_path)

# Step 2: Inspect the first few rows of the dataset to ensure it loaded correctly
print(df.head())
```

```
      Date     IE  Chrome  Firefox  Safari  Opera  Android  X360SafeBrowser  \
0  2008-07  68.57    0.00    26.14    3.30   1.78      0.0              0.0
1  2008-08  68.91    0.00    26.08    2.99   1.83      0.0              0.0
2  2008-09  67.16    1.03    25.77    3.00   2.86      0.0              0.0
3  2008-10  67.68    1.02    25.54    2.91   2.69      0.0              0.0
4  2008-11  68.14    0.93    25.27    2.49   3.01      0.0              0.0

   Maxthon  SonyPS3  ...  SeaMonkey  RockMelt  BlackBerry  Iron  Flock  \
0      0.0     0.00  ...       0.04       0.0         0.0   0.0    0.0
1      0.0     0.00  ...       0.04       0.0         0.0   0.0    0.0
2      0.0     0.00  ...       0.04       0.0         0.0   0.0    0.0
3      0.0     0.00  ...       0.04       0.0         0.0   0.0    0.0
4      0.0     0.02  ...       0.03       0.0         0.0   0.0    0.0
```

2. Define time series data objects.

To define time series data in Python, we need to ensure the Date column in our dataset is correctly recognized as a datetime object and then set it as the index of your DataFrame.



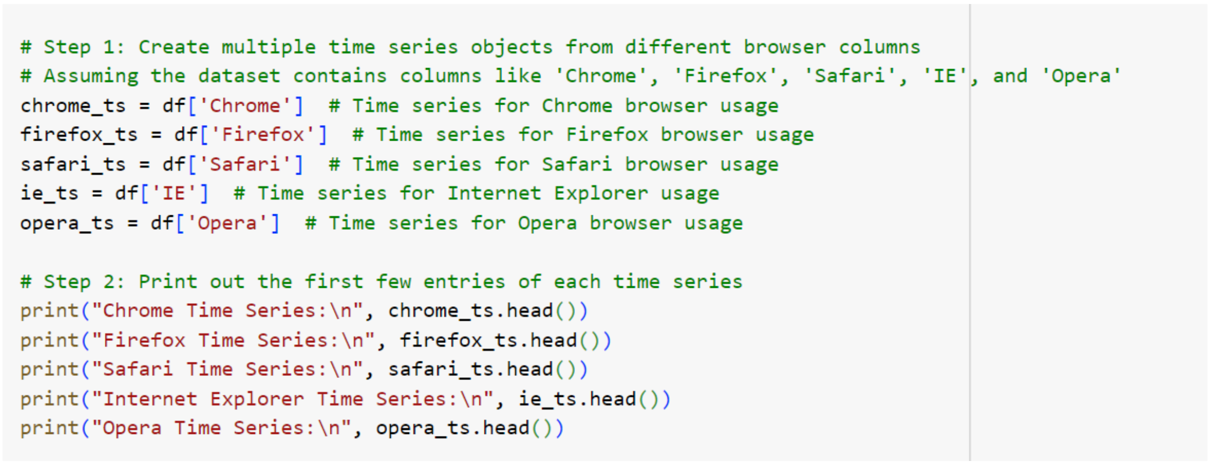
```
import pandas as pd

# Step 1: Read the dataset from a CSV file
file_path = 'browser_usage_2008_2014.csv'
df = pd.read_csv(file_path)

# Step 2: Convert the 'Date' column to datetime format
# Changed the format to '%Y-%m' to match the format in the CSV file (e.g., '2008-07')
df['Date'] = pd.to_datetime(df['Date'], format='%Y-%m', errors='coerce')


# Step 3: Set the 'Date' column as the index of the DataFrame to define it as a time series
df.set_index('Date', inplace=True)

# Step 4: Check if the data is correctly set up as a time series
print(df.head())
```

pd.to_datetime(df['Date']): Converts the 'Date' column into a datetime object that can be used for time series analysis.

df.set_index('Date', inplace=True): Sets the 'Date' column as the index of the DataFrame, which is necessary for time series data manipulation.

Inspect the Data: Use **print(df.head())** to check the first few rows and confirm that the **Date** column has been successfully set as the index.

3. Create a multiple time series object



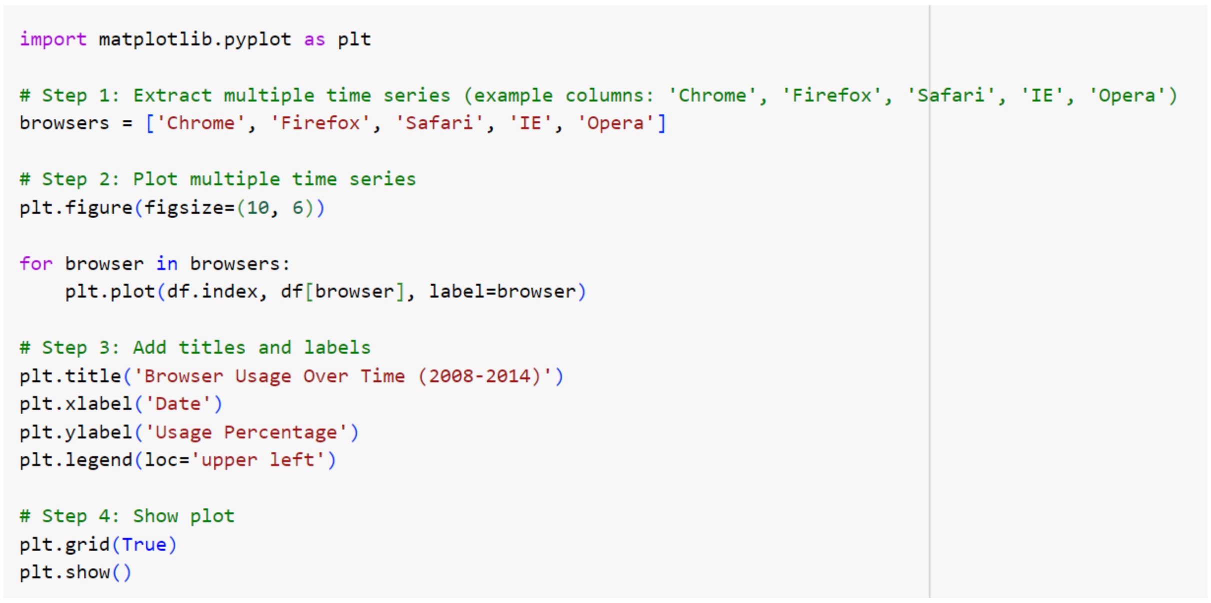
```
# Step 1: Create multiple time series objects from different browser columns
# Assuming the dataset contains columns like 'Chrome', 'Firefox', 'Safari', 'IE', and 'Opera'
chrome_ts = df['Chrome']  # Time series for Chrome browser usage
firefox_ts = df['Firefox']  # Time series for Firefox browser usage
safari_ts = df['Safari']  # Time series for Safari browser usage
ie_ts = df['IE']  # Time series for Internet Explorer usage
opera_ts = df['Opera']  # Time series for Opera browser usage

# Step 2: Print out the first few entries of each time series
print("Chrome Time Series:\n", chrome_ts.head())
print("Firefox Time Series:\n", firefox_ts.head())
print("Safari Time Series:\n", safari_ts.head())
print("Internet Explorer Time Series:\n", ie_ts.head())
print("Opera Time Series:\n", opera_ts.head())
```

Extracting Time Series: For each browser (e.g., Chrome, Firefox, Safari), we extract its corresponding column as a time series object. This allows you to work with each one individually.

Naming the Time Series: We create variables like **chrome_ts, firefox_ts, safari_ts,** etc., to store each browser's time series data.

Each of these variables **(chrome_ts, firefox_ts, etc.)** is now an individual time series object, which you can manipulate or plot separately.

4. Plot multiple time series object using standard Python graphics.



```python
import matplotlib.pyplot as plt

# Step 1: Extract multiple time series (example columns: 'Chrome', 'Firefox', 'Safari', 'IE', 'Opera')
browsers = ['Chrome', 'Firefox', 'Safari', 'IE', 'Opera']

# Step 2: Plot multiple time series
plt.figure(figsize=(10, 6))

for browser in browsers:
    plt.plot(df.index, df[browser], label=browser)

# Step 3: Add titles and labels
plt.title('Browser Usage Over Time (2008-2014)')
plt.xlabel('Date')
plt.ylabel('Usage Percentage')
plt.legend(loc='upper left')

# Step 4: Show plot
plt.grid(True)
plt.show()
```

browsers: A list containing the names of the columns representing each browser's time series.

plt.plot(): Loops through each browser's time series and plots them on the same graph.

Labels: Adds titles, x-axis, and y-axis labels for clarity.

plt.legend(): Displays a legend to differentiate between different time series (e.g., Chrome, Firefox, Safari).

plt.grid(True): Adds gridlines for better readability.

5. Save the plot on a pdf file.

plt.savefig('browser_usage_time_series.pdf')