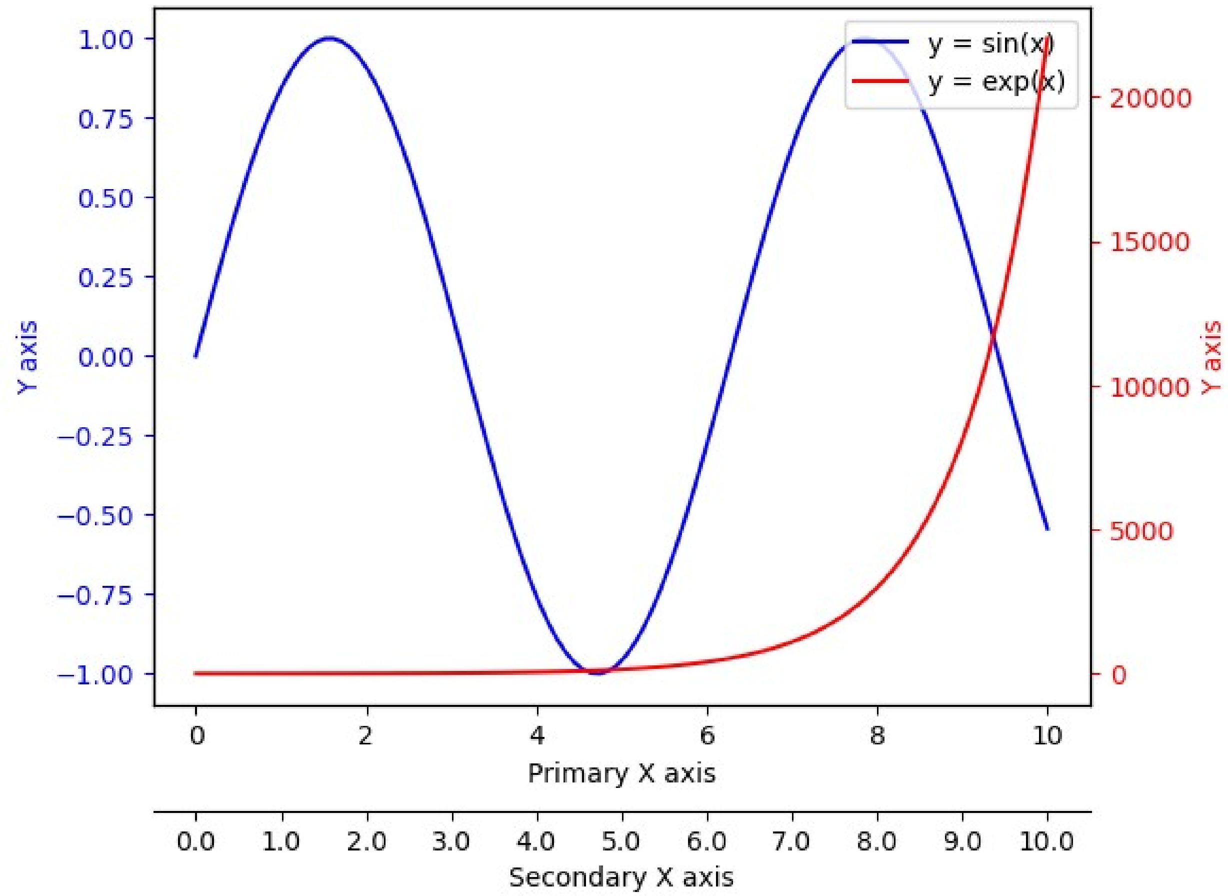
Q.1 Use Secondary Axis In matplotlib

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 10, 100)
y1 = np.sin(x)
y2 = np.exp(x)

fig, ax1 = plt.subplots()

ax1.plot(x, y1, 'b-', label='y = sin(x)')
ax1.set_xlabel('Primary X axis')
ax1.set_ylabel('Y axis', color='b')
ax1.tick_params(axis='y', colors='b')

ax2 = ax1.twinx()

ax2.plot(x, y2, 'r-', label='y = exp(x)')
ax2.set_ylabel('Y axis', color='r')
ax2.tick_params(axis='y', colors='r')

ax3 = ax1.twiny()

ax3.xaxis.set_ticks_position('bottom')
ax3.xaxis.set_label_position('bottom')

ax3.set_xlim(ax1.get_xlim())

ax3.set_xticks(np.linspace(0, 10, 11))
ax3.set_xticklabels(['{:.1f}'.format(i) for i in np.linspace(0, 10, 11)])

ax3.set_xlabel('Secondary X axis')

ax3.spines['bottom'].set_position(('outward', 40))

lines1, labels1 = ax1.get_legend_handles_labels()
lines2, labels2 = ax2.get_legend_handles_labels()
ax1.legend(lines1 + lines2, labels1 + labels2, loc='upper right')

plt.show()
```



Q.2 Use lambda operations on the dataframes

```
import pandas as pd

netflix_data = pd.read_csv("Netflix_Dataset.csv")
netflix_data.head()

                 Title                                          Genre  \
0     Lets Fight Ghost  Crime, Drama, Fantasy, Horror, Romance
1  HOW TO BUILD A GIRL                                         Comedy
2     The Con-Heartist                               Comedy, Romance
3         Gleboka woda                                          Drama
4        Only a Mother                                          Drama

                                                Tags         Languages
\
0  Comedy Programmes,Romantic TV Comedies,Horror ...  Swedish, Spanish

1      Dramas,Comedies,Films Based on Books,British            English

2  Romantic Comedies,Comedies,Romantic Films,Thai...              Thai

3   TV Dramas,Polish TV Shows,Social Issue TV Dramas            Polish
```

```
4  Social Issue Dramas,Dramas,Movies Based on Boo...           Swedish


                                Country Availability       Runtime  \
0                                           Thailand  < 30 minutes
1                                             Canada      1-2 hour
2                                           Thailand       > 2 hrs
3                                             Poland  < 30 minutes
4  Lithuania,Poland,France,Italy,Spain,Greece,Bel...      1-2 hour

          Director                                             Writer
\
0  Tomas Alfredson                              John Ajvide Lindqvist

1    Coky Giedroyc                                      Caitlin Moran

2    Mez Tharatorn  Pattaranad Bhiboonsawade, Mez Tharatorn, Thods...

3              NaN                                                NaN

4      Alf Sjöberg                                  Ivar Lo-Johansson


                                              Actors View Rating  ...
\
0  Lina Leandersson, Kåre Hedebrant, Per Ragnar, ...           R  ...

1  Cleo, Paddy Considine, Beanie Feldstein, Dónal...           R  ...

2  Kathaleeya McIntosh, Nadech Kugimiya, Pimchano...         NaN  ...

3  Katarzyna Maciag, Piotr Nowak, Marcin Dorocins...         NaN  ...

4  Hugo Björne, Eva Dahlbeck, Ulf Palme, Ragnar F...         NaN  ...


   Awards Nominated For   Boxoffice  Release Date Netflix Release Date
\
0                  57.0  $21,22,065     12-Dec-08             04-03-21

1                   NaN    $70,632      08-May-20             04-03-21

2                   NaN         NaN     03-Dec-20             03-03-21

3                   4.0         NaN     14-Jun-11             03-03-21

4                   1.0         NaN     31-Oct-49             03-03-21


                         Production House  \
0                Canal+, Sandrew Metronome
1  Film 4, Monumental Pictures, Lionsgate
2                                      NaN
3                                      NaN
4                                      NaN

                             Netflix Link  \
0  https://www.netflix.com/watch/81415947
1  https://www.netflix.com/watch/81041267
2  https://www.netflix.com/watch/81306155
3  https://www.netflix.com/watch/81307527
4  https://www.netflix.com/watch/81382068

                                             Summary Series or
Movie  \
0  A med student with a supernatural gift tries t...          Series

1  When nerdy Johanna moves to London, things get...           Movie

2  After her ex-boyfriend cons her out of a large...           Movie

3  A group of social welfare workers led by their...          Series

4  An unhappily married farm worker struggling to...           Movie


  IMDb Votes                                              Image
0   205926.0  https://occ-0-4708-64.1.nflxso.net/dnm/api/v6/...
1     2838.0  https://occ-0-1081-999.1.nflxso.net/dnm/api/v6...
2      131.0  https://occ-0-2188-64.1.nflxso.net/dnm/api/v6/...
3       47.0  https://occ-0-2508-2706.1.nflxso.net/dnm/api/v...
4       88.0  https://occ-0-2851-41.1.nflxso.net/dnm/api/v6/...

[5 rows x 22 columns]
```

```
import math
netflix_data['IMDb Score'] = netflix_data['IMDb Score'].apply(lambda
x: 6.6 if math.isnan(x) else x)
netflix_data['IMDb Score']
```

```
0       7.9
1       5.8
2       7.4
3       7.5
4       6.7
       ...
9398    6.2
9399    6.2
9400    7.3
9401    7.8
```

```
9402    6.8
Name: IMDb Score, Length: 9403, dtype: float64
```

Q.3 Code to check anagrams

```
def are_anagrams(str1, str2):
    return sorted(str1.lower().replace(" ", "")) ==
sorted(str2.lower().replace(" ", ""))

str1 = "listen"
str2 = "silent"
if are_anagrams(str1, str2):
    print(f"{str1} and {str2} are anagrams.")
else:
    print(f"{str1} and {str2} are not anagrams.")
```

```
listen and silent are anagrams.
```

Q.4 Regex example in python

```
import re

#Sample
sample = """
<email>user1@example.com</email>
<email>user2@example.com</email>
<email>invalid_email.com</email>
"""

# Regular expression pattern
email_pattern = r'<email>(.*?)</email>'

# Extract
emails = re.findall(email_pattern, sample)

for email in emails:
    print(email)
```

```
user1@example.com
user2@example.com
invalid_email.com
```

```
#sample
sample1 = """
[12131234]
[232344]
[invalid_phone_number]
"""

# Regular expression pattern
```

```
phone_pattern = r'\[(.*?)\]'

# Extract
phone_numbers = re.findall(phone_pattern, sample1)

for phone_number in phone_numbers:
    print(phone_number)
```

```
12131234
232344
invalid_phone_number
```

Q.5 Explain three types of merges in git

1)Fast-forward Merge:

a.When you're merging a branch with no new changes onto the current branch.

b.It moves the current branch pointer directly to the branch you're merging without creating a new commit.

2)Regular Merge:

a.Combines changes from two divergent branches

b.Creates a new merge commit to integrate the changes, even if there are no conflicts.

3)Squash Merge:

a.Combines multiple commits from a feature branch into a single commit.

b.Useful for keeping the commit history clean and concise.