

Basics

June 5, 2023

1 European Union Exchange Rates Analysis

1.1 1. Dataset Introduction

```
[3]: import pandas as pd
exchange_rates= pd.read_csv('euro-daily-hist_1999_2020.csv')
```

```
[4]: exchange_rates.head(5)
```

```
[4]:  Period\Unit: [Australian dollar ] [Bulgarian lev ] [Brazilian real ] \
0    2021-01-08                1.5758                1.9558                6.5748
1    2021-01-07                1.5836                1.9558                6.5172
2    2021-01-06                1.5824                1.9558                6.5119
3    2021-01-05                1.5927                1.9558                6.5517
4    2021-01-04                1.5928                1.9558                6.3241

    [Canadian dollar ] [Swiss franc ] [Chinese yuan renminbi ] [Cypriot pound ] \
0                1.5543                1.0827                7.9184                NaN
1                1.5601                1.0833                7.9392                NaN
2                1.5640                1.0821                7.9653                NaN
3                1.5651                1.0803                7.9315                NaN
4                1.5621                1.0811                7.9484                NaN

    [Czech koruna ] [Danish krone ] ... [Romanian leu ] [Russian rouble ] \
0                26.163                7.4369 ...                4.8708                90.8000
1                26.147                7.4392 ...                4.8712                91.2000
2                26.145                7.4393 ...                4.8720                90.8175
3                26.227                7.4387 ...                4.8721                91.6715
4                26.141                7.4379 ...                4.8713                90.3420

    [Swedish krona ] [Singapore dollar ] [Slovenian tolar ] [Slovak koruna ] \
0                10.0510                1.6228                NaN                NaN
1                10.0575                1.6253                NaN                NaN
2                10.0653                1.6246                NaN                NaN
3                10.0570                1.6180                NaN                NaN
4                10.0895                1.6198                NaN                NaN

    [Thai baht ] [Turkish lira ] [US dollar ] [South African rand ]
```

0	36.8480	9.0146	1.2250	18.7212
1	36.8590	8.9987	1.2276	18.7919
2	36.9210	9.0554	1.2338	18.5123
3	36.7760	9.0694	1.2271	18.4194
4	36.7280	9.0579	1.2296	17.9214

[5 rows x 41 columns]

```
[5]: exchange_rates.tail(5)
```

```
[5]:      Period\Unit: [Australian dollar ] [Bulgarian lev ] [Brazilian real ] \
5694    1999-01-08                1.8406                NaN                NaN
5695    1999-01-07                1.8474                NaN                NaN
5696    1999-01-06                1.8820                NaN                NaN
5697    1999-01-05                1.8944                NaN                NaN
5698    1999-01-04                1.9100                NaN                NaN
```

```
      [Canadian dollar ] [Swiss franc ] [Chinese yuan renminbi ] \
5694                1.7643                1.6138                NaN
5695                1.7602                1.6165                NaN
5696                1.7711                1.6116                NaN
5697                1.7965                1.6123                NaN
5698                1.8004                1.6168                NaN
```

```
      [Cypriot pound ] [Czech koruna ] [Danish krone ] ... [Romanian leu ] \
5694                0.58187                34.938                7.4433 ...                1.3143
5695                0.58187                34.886                7.4431 ...                1.3092
5696                0.58200                34.850                7.4452 ...                1.3168
5697                0.58230                34.917                7.4495 ...                1.3168
5698                0.58231                35.107                7.4501 ...                1.3111
```

```
      [Russian rouble ] [Swedish krona ] [Singapore dollar ] \
5694                27.2075                9.1650                1.9537
5695                26.9876                9.1800                1.9436
5696                27.4315                9.3050                1.9699
5697                26.5876                9.4025                1.9655
5698                25.2875                9.4696                1.9554
```

```
      [Slovenian tolar ] [Slovak koruna ] [Thai baht ] [Turkish lira ] \
5694                188.8400                42.560                42.5590                0.3718
5695                188.8000                42.765                42.1678                0.3701
5696                188.7000                42.778                42.6949                0.3722
5697                188.7750                42.848                42.5048                0.3728
5698                189.0450                42.991                42.6799                0.3723
```

```
      [US dollar ] [South African rand ]
5694                1.1659                6.7855
```

5695	1.1632	6.8283
5696	1.1743	6.7307
5697	1.1790	6.7975
5698	1.1789	6.9358

[5 rows x 41 columns]

```
[6]: exchange_rates.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5699 entries, 0 to 5698
Data columns (total 41 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Period\Unit:                          5699 non-null   object
1   [Australian dollar ]                  5699 non-null   object
2   [Bulgarian lev ]                     5297 non-null   object
3   [Brazilian real ]                    5431 non-null   object
4   [Canadian dollar ]                   5699 non-null   object
5   [Swiss franc ]                       5699 non-null   object
6   [Chinese yuan renminbi ]              5431 non-null   object
7   [Cypriot pound ]                     2346 non-null   object
8   [Czech koruna ]                      5699 non-null   object
9   [Danish krone ]                      5699 non-null   object
10  [Estonian kroon ]                    3130 non-null   object
11  [UK pound sterling ]                 5699 non-null   object
12  [Greek drachma ]                     520 non-null    object
13  [Hong Kong dollar ]                 5699 non-null   object
14  [Croatian kuna ]                    5431 non-null   object
15  [Hungarian forint ]                 5699 non-null   object
16  [Indonesian rupiah ]                 5699 non-null   object
17  [Israeli shekel ]                    5431 non-null   object
18  [Indian rupee ]                     5431 non-null   object
19  [Iceland krona ]                    3292 non-null   float64
20  [Japanese yen ]                     5699 non-null   object
21  [Korean won ]                       5699 non-null   object
22  [Lithuanian litas ]                  4159 non-null   object
23  [Latvian lats ]                      3904 non-null   object
24  [Maltese lira ]                     2346 non-null   object
25  [Mexican peso ]                     5699 non-null   object
26  [Malaysian ringgit ]                 5699 non-null   object
27  [Norwegian krone ]                   5699 non-null   object
28  [New Zealand dollar ]                 5699 non-null   object
29  [Philippine peso ]                   5699 non-null   object
30  [Polish zloty ]                      5699 non-null   object
31  [Romanian leu ]                     5637 non-null   float64
32  [Russian rouble ]                    5699 non-null   object
```

```

33 [Swedish krona ]          5699 non-null  object
34 [Singapore dollar ]      5699 non-null  object
35 [Slovenian tolar ]       2085 non-null  object
36 [Slovak koruna ]        2608 non-null  object
37 [Thai baht ]            5699 non-null  object
38 [Turkish lira ]         5637 non-null  float64
39 [US dollar ]            5699 non-null  object
40 [South African rand ]   5699 non-null  object
dtypes: float64(3), object(38)
memory usage: 1.8+ MB

```

- There are 5 rows and 41 columns in this data set.
- There are null values in the data set.
- Most columns are of object (probably string) data type, but 3 are of float64.

1.2 2. Dataset Cleaning

```

[7]: exchange_rates.rename(columns={'[US dollar ]': 'US_dollar',
                                   'Period\\Unit': 'Time'},
                           inplace=True)
exchange_rates['Time'] = pd.to_datetime(exchange_rates['Time'])
exchange_rates.sort_values('Time', inplace=True)
exchange_rates.reset_index(drop=True, inplace=True)

```

```

[8]: euro_to_dollar = exchange_rates[['Time', 'US_dollar']]

```

```

[9]: print(euro_to_dollar)

```

```

      Time US_dollar
0  1999-01-04    1.1789
1  1999-01-05    1.1790
2  1999-01-06    1.1743
3  1999-01-07    1.1632
4  1999-01-08    1.1659
...
5694 2021-01-04    1.2296
5695 2021-01-05    1.2271
5696 2021-01-06    1.2338
5697 2021-01-07    1.2276
5698 2021-01-08    1.2250

```

[5699 rows x 2 columns]

```

[10]: euro_to_dollar['US_dollar'].value_counts()

```

```

[10]: -          62
      1.2276      9
      1.1215      8

```

```

1.1305    7
1.1797    6
..
1.2571    1
1.2610    1
1.2651    1
1.2632    1
1.2193    1
Name: US_dollar, Length: 3528, dtype: int64

```

```
[11]: euro_to_dollar = euro_to_dollar[euro_to_dollar["US_dollar"] != "-"]
```

```
[12]: euro_to_dollar["US_dollar"] = pd.to_numeric(euro_to_dollar["US_dollar"])
```

```
[13]: print(euro_to_dollar.dtypes)
```

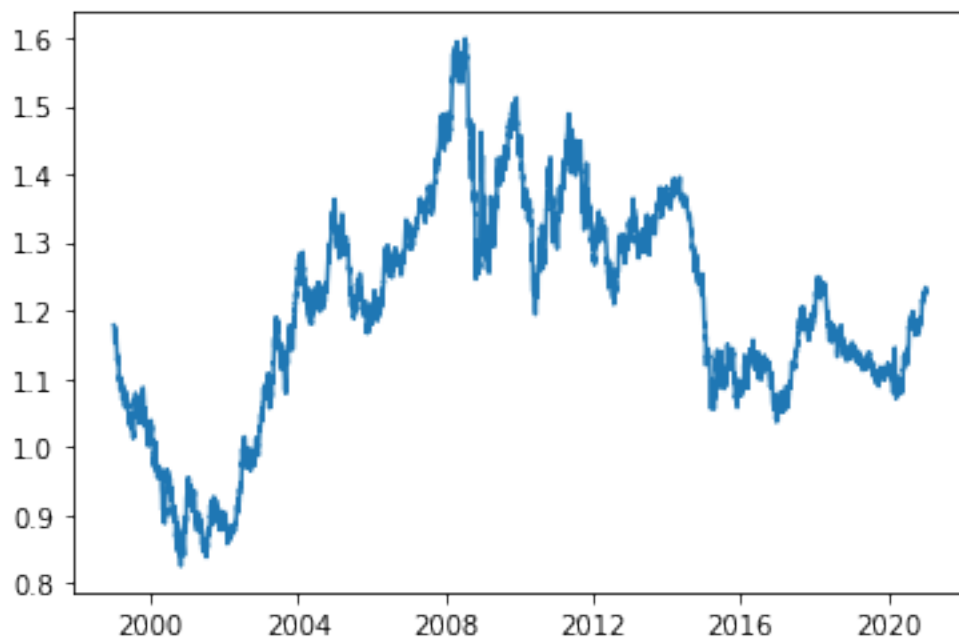
```

Time          datetime64[ns]
US_dollar      float64
dtype: object

```

1.3 3. Rolling Mean

```
[14]: import matplotlib.pyplot as plt
      %matplotlib inline
      # Enables Jupyter to display graphs
      plt.plot(euro_to_dollar['Time'],
               euro_to_dollar['US_dollar'])
      plt.show()
```



```
[15]: euro_to_dollar['rolling_mean'] = euro_to_dollar['US_dollar'].rolling(30).mean()
```

```
[16]: ##matplotlib inline  
# Enables Jupyter to display graphs  
#plt.plot(euro_to_dollar['Time'],  
#         euro_to_dollar['rolling_mean'])  
#plt.show()
```

1.4 4. 2007-2008 financial crisis

Now, we will analyze the change of the U.S. to euro exchange rate during the 2007-2008 financial crisis. We'll show the data from 2006 and 2009 for comparison.

1.5 5. Graph description

- We will employ a line plot. We will plot the “Time” and “rolling_mean” columns of the euro_to_dollar dataframe.
- There will be three graphs. Each graph will highlight the time periods of 2006, 2007-2008, and 2009.
- The section of the “Time” vs “US_dollar” curve of the line plot corresponding to those periods will be solid while the rest of the curve will have transparency.
- The time period will be annotated above the curve-section.
- The color of the curves will stay at the default blue color.

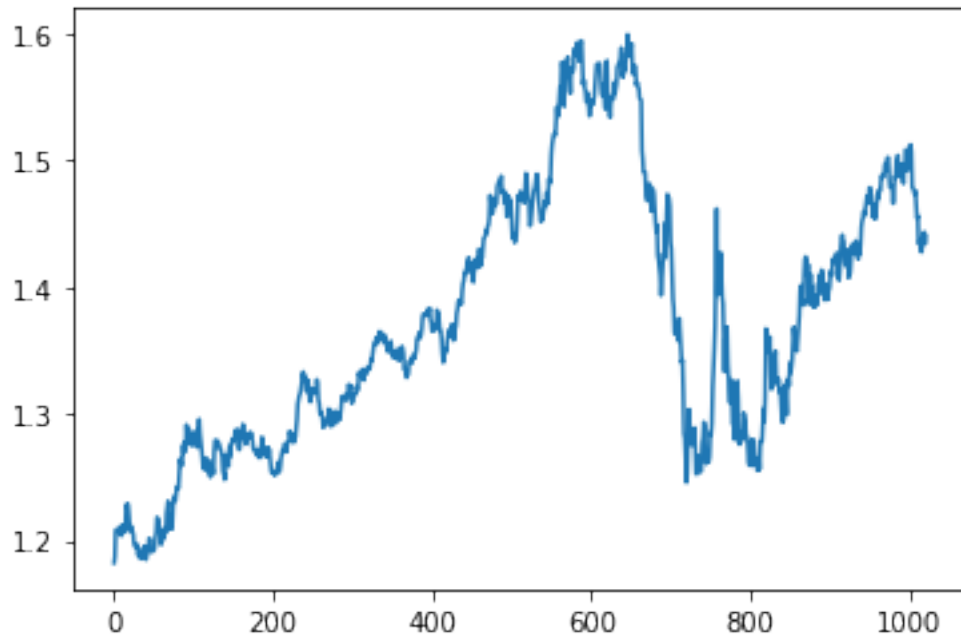
1.6 6. Graph

```
[17]: euro_to_dollar_2006_to_2009_data = euro_to_dollar[(euro_to_dollar["Time"] <=   
↪ '2010-01-01') & ('2006-01-01' <= euro_to_dollar["Time"])]
```

```
[18]: euro_to_dollar_2006_to_2009_data.reset_index(drop=True, inplace = True)
```

```
[19]: plt.plot(euro_to_dollar_2006_to_2009_data.  
↪ index, euro_to_dollar_2006_to_2009_data["US_dollar"])
```

```
[19]: [
```



1.6.1 Now we plot the graph:

```
[20]: import matplotlib.pyplot as plt
      %matplotlib inline
      # Enables Jupyter to display graphs

      #fig, (ax1, ax2, ax3) = plt.subplots(nrows=3,ncols=1,figsize=(10,12))
      #fig, (ax1, ax2, ax3) = plt.subplots(nrows=3, ncols=1, figsize=(6, 8), dpi=100)
      fig, (ax1, ax2, ax3) = plt.subplots(nrows=3, ncols=1, figsize=(10,12))

      axes = [ax1,ax2,ax3]
      for ax in axes:
          ax.plot(
              euro_to_dollar_2006_to_2009_data.index,
              euro_to_dollar_2006_to_2009_data["US_dollar"],
              color='#af0b1e',
              alpha=0.1
          )
          ax.set_xticks([])
          ax.set_yticks([])
          for location in ['left','right','bottom','top']:
              ax.spines[location].set_visible(False)
```

```

ax1.plot(
    euro_to_dollar_2006_to_2009_data.
    →index[(euro_to_dollar_2006_to_2009_data["Time"] < '2007-01-01') &
              ('2006-01-01' <=
    →euro_to_dollar_2006_to_2009_data["Time"])]],
    □
    →euro_to_dollar_2006_to_2009_data["US_dollar"][(euro_to_dollar_2006_to_2009_data["Time"]
    →< '2007-01-01') &
              ('2006-01-01' <=
    →euro_to_dollar_2006_to_2009_data["Time"])]],
    color='#af0b1e')
ax1.text(x=-25,y=1.37,s="January 1, 2006-December 31, 2006",size =
    →7,color='#af0b1e', weight='bold')
ax1.text(x=0,y=1.24,s="1.23",size = 7)
ax1.text(x=530,y=1.58,s="1.58",size = 7,alpha=0.5)
ax1.text(x=930,y=1.5,s="1.5",size = 7,alpha=0.5)
ax1.text(x=130,y=1.7,s="U.S. to euro exchange rate between 2006 to 2009",
    size=14, weight='bold')
ax1.text(x=130,y=1.65,s="Exchange rate peaked during the financial crisis of
    →2007-2008",
    size=12)

ax2.plot(
    euro_to_dollar_2006_to_2009_data.
    →index[(euro_to_dollar_2006_to_2009_data["Time"] < '2009-01-01') &
              ('2007-01-01' <=
    →euro_to_dollar_2006_to_2009_data["Time"])]],
    □
    →euro_to_dollar_2006_to_2009_data["US_dollar"][(euro_to_dollar_2006_to_2009_data["Time"]
    →< '2009-01-01') &
              ('2007-01-01' <=
    →euro_to_dollar_2006_to_2009_data["Time"])]],
    color='#af0b1e')
ax2.text(x=400,y=1.28,s="January 1, 2007-December 31, 2008",size =
    →7,color='#af0b1e', weight='bold')
ax2.text(x=0,y=1.24,s="1.23",size = 7,alpha=0.5)
ax2.text(x=530,y=1.58,s="1.58",size = 7,alpha=0.5)
ax2.text(x=930,y=1.5,s="1.5",size = 7,alpha=0.5)

ax3.plot(
    euro_to_dollar_2006_to_2009_data.
    →index[(euro_to_dollar_2006_to_2009_data["Time"] < '2010-01-01') &
              ('2009-01-01' <=
    →euro_to_dollar_2006_to_2009_data["Time"])]],

```



```

    ↪euro_to_dollar_2006_to_2009_data["US_dollar"][(euro_to_dollar_2006_to_2009_data["Time"]
    ↪< '2010-01-01') &
                                ('2009-01-01' <=
    ↪euro_to_dollar_2006_to_2009_data["Time"])],
                                color='#af0b1e')
ax3.text(x=780,y=1.55,s="January 1, 2009-December 31, 2009",size =
    ↪7,color='#af0b1e', weight='bold')
ax3.text(x=0,y=1.24,s="1.23",size = 7,alpha=0.5)
ax3.text(x=530,y=1.58,s="1.58",size = 7,alpha=0.5)
ax3.text(x=930,y=1.5,s="1.5",size = 7,alpha=0.5)

plt.show()

```

U.S. to euro exchange rate between 2006 to 2009

Exchange rate peaked during the financial crisis of 2007-2008

