Data Visualization Project: Euro-dollar Rate Changes under the Last Three US Presidents

Introducing the Dataset

In [47]:
```
import pandas as pd
exchange_rates = pd.read_csv('euro-daily-hist_1999_2020.csv')
exchange_rates.head(5)
```

Out[47]:

	Period\Unit:	[Australian dollar]	[Bulgarian lev]	[Brazilian real]	[Canadian dollar]	[Swiss franc]	[Chinese yuan renminbi]	[Cypriot pound]	[Czech koruna]	[Danish krone]	...	[Romanian leu]	[Russian rouble]	[Swedish krona]	[Singapore dollar]	[Slovenian tolar]	[Slovak koruna]	[Thai baht]	[Turkish lira]	[US dollar]
0	2021-01-08	1.5758	1.9558	6.5748	1.5543	1.0827	7.9184	NaN	26.163	7.4369	...	4.8708	90.8000	10.0510	1.6228	NaN	NaN	36.8480	9.0146	1.2250
1	2021-01-07	1.5836	1.9558	6.5172	1.5601	1.0833	7.9392	NaN	26.147	7.4392	...	4.8712	91.2000	10.0575	1.6253	NaN	NaN	36.8590	8.9987	1.2276
2	2021-01-06	1.5824	1.9558	6.5119	1.5640	1.0821	7.9653	NaN	26.145	7.4393	...	4.8720	90.8175	10.0653	1.6246	NaN	NaN	36.9210	9.0554	1.2338
3	2021-01-05	1.5927	1.9558	6.5317	1.5651	1.0803	7.9315	NaN	26.227	7.4387	...	4.8721	91.6715	10.0570	1.6190	NaN	NaN	36.7760	9.0694	1.2271
4	2021-01-04	1.5928	1.9558	6.3241	1.5621	1.0811	7.9484	NaN	26.141	7.4379	...	4.8713	90.3420	10.0895	1.6198	NaN	NaN	36.7280	9.0579	1.2296

5 rows × 41 columns

In [48]:
```
exchange_rates.tail(5)
```

Out[48]:

	Period\Unit:	[Australian dollar]	[Bulgarian lev]	[Brazilian real]	[Canadian dollar]	[Swiss franc]	[Chinese yuan renminbi]	[Cypriot pound]	[Czech koruna]	[Danish krone]	...	[Romanian leu]	[Russian rouble]	[Swedish krona]	[Singapore dollar]	[Slovenian tolar]	[Slovak koruna]	[Thai baht]	[Turkish lira]	[US dollar]
5694	1999-01-08	1.8406	NaN	NaN	1.7643	1.6138	NaN	0.58187	34.938	7.4433	...	1.3143	27.2075	9.1650	1.9537	188.8400	42.560	42.5590	0.3718	1.160
5695	1999-01-07	1.8474	NaN	NaN	1.7602	1.6165	NaN	0.58187	34.886	7.4431	...	1.3092	26.9876	9.1800	1.9436	188.8000	42.765	42.1678	0.3701	1.160
5696	1999-01-06	1.8820	NaN	NaN	1.7711	1.6116	NaN	0.58200	34.850	7.4452	...	1.3168	27.4315	9.3050	1.9699	188.7000	42.778	42.6949	0.3722	1.174
5697	1999-01-05	1.8944	NaN	NaN	1.7965	1.6123	NaN	0.58230	34.917	7.4495	...	1.3168	26.5876	9.4025	1.9655	188.7750	42.848	42.5048	0.3728	1.179
5698	1999-01-04	1.9100	NaN	NaN	1.8004	1.6168	NaN	0.58231	35.107	7.4501	...	1.3111	25.2875	9.4696	1.9554	189.0450	42.991	42.6799	0.3723	1.178

5 rows × 41 columns

In [49]:
```
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 ]       3138 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 ]        5699 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(1), object(39)
memory usage: 1.6+ MB
```

Data Cleaning

In [50]:
```
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)
```

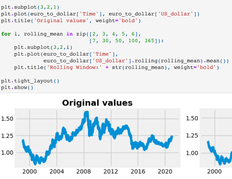
In [51]:
```
euro_to_dollar = exchange_rates[['Time','US_dollar']].copy()
euro_to_dollar['US_dollar'].value_counts()
```

Out[51]:
```
-       62
1.2276   9
1.1215   8
1.1305   7
1.1797   6
        ..
1.2926   1
0.8734   1
0.9775   1
0.9425   1
1.1959   1
Name: US_dollar, Length: 3528, dtype: int64
```

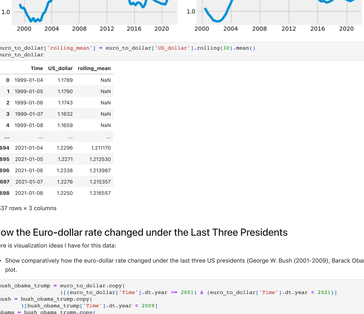
In [52]:
```
euro_to_dollar = euro_to_dollar[euro_to_dollar['US_dollar'] != '-']
euro_to_dollar['US_dollar'] = euro_to_dollar['US_dollar'].astype(float)
```

Rolling Mean

In [53]:
```
import matplotlib.pyplot as plt
%matplotlib inline

plt.plot(euro_to_dollar['Time'], euro_to_dollar['US_dollar'])
plt.show()
```



In [54]:
```
plt.figure(figsize=(9,6))

plt.subplot(3,2,1)
plt.plot(euro_to_dollar['Time'], euro_to_dollar['US_dollar'])
plt.title('Original values',weight='bold')

for i, rolling_mean in zip([2, 3, 4, 5, 6],
                           [7, 30, 50, 100, 365]):

    plt.subplot(3,2,i)
    plt.plot(euro_to_dollar['Time'],
             euro_to_dollar['US_dollar'].rolling(rolling_mean).mean())
    plt.title('Rolling Window: ' + str(rolling_mean), weight='bold')

plt.tight_layout()
plt.show()
```



In [55]:
```
euro_to_dollar['rolling_mean'] = euro_to_dollar['US_dollar'].rolling(30).mean()
euro_to_dollar
```

Out[55]:

	Time	US_dollar	rolling_mean
0	1999-01-04	1.1789	NaN
1	1999-01-05	1.1790	NaN
2	1999-01-06	1.1743	NaN
3	1999-01-07	1.1632	NaN
4	1999-01-08	1.1659	NaN
...	...	...	...
5694	2021-01-04	1.2296	1.211170
5695	2021-01-05	1.2271	1.212530
5696	2021-01-06	1.2338	1.213987
5697	2021-01-07	1.2276	1.215357
5698	2021-01-08	1.2250	1.216557

5637 rows × 3 columns

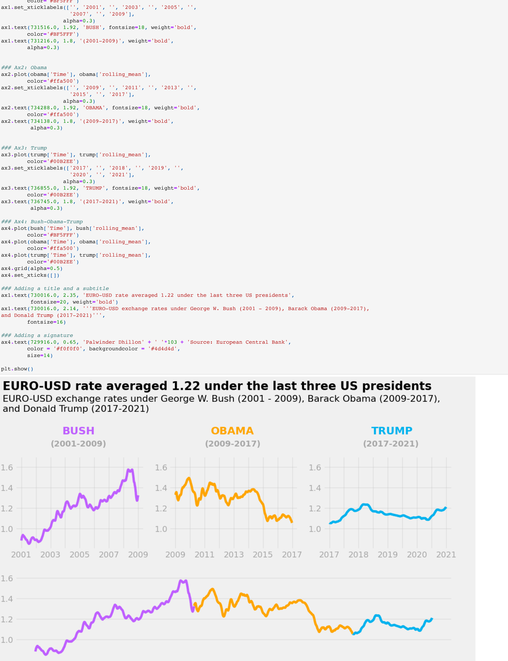
How the Euro-dollar rate changed under the Last Three Presidents

Here is visualization ideas I have for this data:

- Show comparatively how the euro-dollar rate changed under the last three US presidents (George W. Bush (2001-2009), Barack Obama (2009-2017), and Donald Trump (2017-2021)). Use a line plot.

In [56]:
```
bush_obama_trump = euro_to_dollar.copy(
                   )[(euro_to_dollar['Time'].dt.year >= 2001) & (euro_to_dollar['Time'].dt.year < 2021)]
bush = bush_obama_trump.copy(
    )[bush_obama_trump['Time'].dt.year < 2009]
obama = bush_obama_trump.copy(
    )[(bush_obama_trump['Time'].dt.year >= 2009) & (bush_obama_trump['Time'].dt.year < 2017)]
trump = bush_obama_trump.copy(
    )[(bush_obama_trump['Time'].dt.year >= 2017) & (bush_obama_trump['Time'].dt.year < 2021)]
```

In [57]:
```
import matplotlib.style as style
style.use('fivethirtyeight')

### Adding the subplots
plt.figure(figsize=(12, 6))
ax1 = plt.subplot(2,1,1)
ax2 = plt.subplot(2,2,2)
ax3 = plt.subplot(2,2,3)
ax4 = plt.subplot(2,2,4)
axes = [ax1, ax2, ax3, ax4]

### Changes to all the subplots
for ax in axes:
    ax.set_ylim(0.8, 1.7)
    ax.set_yticks([1.0, 1.2, 1.4, 1.6])
    ax.set_yticklabels(['1.0', '1.2', '1.4', '1.6'],
                       alpha=0.3)
    ax.grid(alpha=0.5)


### Ax1: Bush
ax1.plot(bush['Time'], bush['rolling_mean'],
         color='#BF5FFF')
ax1.set_xticklabels(['', '2001', '', '2003', '', '2005', '',
                     '2007', '', '2009'],
                    alpha=0.3)
ax1.text(731516.0, 1.92, 'BUSH', fontsize=18, weight='bold',
         color='#BF5FFF')
ax1.text(731216.0, 1.8, '(2001-2009)', weight='bold',
         alpha=0.3)


### Ax2: Obama
ax2.plot(obama['Time'], obama['rolling_mean'],
         color='#ffa500')
ax2.set_xticklabels(['', '2009', '', '2011', '', '2013', '',
                     '2015', '', '2017'],
                    alpha=0.3)
ax2.text(734288.0, 1.92, 'OBAMA', fontsize=18, weight='bold',
         color='#ffa500')
ax2.text(734138.0, 1.8, '(2009-2017)', weight='bold',
         alpha=0.3)


### Ax3: Trump
ax3.plot(trump['Time'], trump['rolling_mean'],
         color='#0192E8')
ax3.set_xticklabels(['2017', '', '2018', '', '2019', '',
                     '2020', '', '2021'],
                    alpha=0.3)
ax3.text(736745.0, 1.92, 'TRUMP', fontsize=18, weight='bold',
         color='#0192E8')
ax3.text(736745.0, 1.8, '(2017-2021)', weight='bold',
         alpha=0.3)

### Ax4: Bush-Obama-Trump
ax4.plot(bush['Time'], bush['rolling_mean'],
         color='#BF5FFF')
ax4.plot(obama['Time'], obama['rolling_mean'],
         color='#ffa500')
ax4.plot(trump['Time'], trump['rolling_mean'],
         color='#0192E8')
ax4.grid(alpha=0.5)
ax4.set_xticks([])

### Adding a title and a subtitle
ax1.text(730916.0, 2.25, 'EURO-USD rate averaged 1.22 under the last three US presidents',
         fontsize=20, weight='bold')
ax1.text(730916.0, 2.14, '''EURO-USD exchange rates under George W. Bush (2001 - 2009), Barack Obama (2009-2017),
and Donald Trump (2017-2021) ''',
         fontsize=16)

### Adding a signature
ax4.text(729916.0, 0.65, 'Palwinder Dhillon' + ' ' *16 + 'Source: European Central Bank',
         color = '#f0f0f0', backgroundcolor = '#464646',
         size=14)

plt.show()
```



Using the data about the Euro-USD exchange rate, I was able to create the visualization above that shows that the exchange rate averaged 1.22 under the last three US presidents using rolling means.

In [ ]:
```

```