Internship Project 2

Storytelling Data Visualization on Euro-US Exchange rates

There are two types of Data Visualizations:

- Exploratory Data Visualization
- Explanatory Dtaa Visualization

In this project, we are going to focus on explanatory data visualization and learn following things:

- How to use information design principles (familiarity and maximizing data-ink ratio) to create better grpahs for audience.
- About the elements of a story and how to create data visulaizations using Matplotlib.
- How to guide the attention of audience with pre-attentive attributes.
- How to use matplotlib built-in styles -- with a casestudy oN FiveThirtyEight style.

In [1]:
```
import pandas as pd
import warnings

warnings.filterwarnings("ignore")
```

In [2]:
```
exh_rates = pd.read_csv("euro-daily-hist_1999_2022.csv")
```

In [3]:
```
exh_rates.head()
```

Out[3]:

	Period\Unit:	[Australian dollar]	[Bulgarian lev]	[Brazilian real]	[Canadian dollar]	[Swiss franc]	[Chinese yuan renminbi]	[Cypriot pound]	[Czech koruna]	[Da kr
0	2023-12-15	1.6324	1.9558	5.4085	1.4653	0.9488	7.7812	NaN	24.477	7.
1	2023-12-14	1.6288	1.9558	5.3349	1.4677	0.949	7.7866	NaN	24.408	7.
2	2023-12-13	1.6452	1.9558	5.3609	1.4644	0.9452	7.7426	NaN	24.476	7.
3	2023-12-12	1.6398	1.9558	5.3327	1.4656	0.9443	7.7447	NaN	24.42	7.
4	2023-12-11	1.642	1.9558	5.3169	1.4609	0.9478	7.7206	NaN	24.367	7.

5 rows × 41 columns

In [4]:
```
exh_rates.shape
```

Out[4]:
```
(6456, 41)
```

In [5]:
```
exh_rates.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6456 entries, 0 to 6455
Data columns (total 41 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Period\Unit:               6456 non-null   object 
 1   [Australian dollar ]       6456 non-null   object 
 2   [Bulgarian lev ]           6054 non-null   object 
 3   [Brazilian real ]          6188 non-null   object 
 4   [Canadian dollar ]         6456 non-null   object 
 5   [Swiss franc ]             6456 non-null   object 
 6   [Chinese yuan renminbi ]   6188 non-null   object 
 7   [Cypriot pound ]           2346 non-null   object 
 8   [Czech koruna ]            6456 non-null   object 
 9   [Danish krone ]            6456 non-null   object 
 10  [Estonian kroon ]          3130 non-null   object 
 11  [UK pound sterling ]       6456 non-null   object 
 12  [Greek drachma ]           520 non-null    object 
 13  [Hong Kong dollar ]        6456 non-null   object 
 14  [Croatian kuna ]           5941 non-null   object 
 15  [Hungarian forint ]        6456 non-null   object 
 16  [Indonesian rupiah ]       6456 non-null   object 
 17  [Israeli shekel ]          6188 non-null   object 
 18  [Indian rupee ]            6188 non-null   object 
 19  [Iceland krona ]           4049 non-null   float64
 20  [Japanese yen ]            6456 non-null   object 
 21  [Korean won ]              6456 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 ]            6456 non-null   object 
 26  [Malaysian ringgit ]       6456 non-null   object 
 27  [Norwegian krone ]         6456 non-null   object 
 28  [New Zealand dollar ]      6456 non-null   object 
 29  [Philippine peso ]         6456 non-null   object 
 30  [Polish zloty ]            6456 non-null   object 
 31  [Romanian leu ]            6394 non-null   float64
 32  [Russian rouble ]          5994 non-null   object 
 33  [Swedish krona ]           6456 non-null   object 
 34  [Singapore dollar ]        6456 non-null   object 
 35  [Slovenian tolar ]         2085 non-null   object 
 36  [Slovak koruna ]           2608 non-null   object 
 37  [Thai baht ]               6456 non-null   object 
 38  [Turkish lira ]            6394 non-null   float64
 39  [US dollar ]               6456 non-null   object 
 40  [South African rand ]      6456 non-null   object 
dtypes: float64(3), object(38)
memory usage: 2.0+ MB
```

In [7]:

```python
exh_rates.rename(columns={'[US dollar ]':'US_dollar', 'Period\\Unit:': 'Time'}, inplac
```

In [8]:

```python
exh_rates["Time"] = pd.to_datetime(exh_rates["Time"])
```

In [9]:

```python
exh_rates.sort_values('Time', inplace=True)
```

In [10]:

```python
exh_rates.head()
```

Out[10]:

	Time	[Australian dollar]	[Bulgarian lev]	[Brazilian real]	[Canadian dollar]	[Swiss franc]	[Chinese yuan renminbi]	[Cypriot pound]	[Czech koruna]	[Danis krone
6455	1999-01-04	1.9100	NaN	NaN	1.8004	1.6168	NaN	0.58231	35.107	7.45
6454	1999-01-05	1.8944	NaN	NaN	1.7965	1.6123	NaN	0.58230	34.917	7.44
6453	1999-01-06	1.8820	NaN	NaN	1.7711	1.6116	NaN	0.58200	34.850	7.44
6452	1999-01-07	1.8474	NaN	NaN	1.7602	1.6165	NaN	0.58187	34.886	7.44
6451	1999-01-08	1.8406	NaN	NaN	1.7643	1.6138	NaN	0.58187	34.938	7.44

5 rows × 41 columns

In [12]:

```python
euro_to_dollar=exh_rates[['Time', 'US_dollar']].copy()
euro_to_dollar['US_dollar'].value_counts()
```

Out[12]:

```
US_dollar
-         62
1.2276     9
1.1215     8
1.0888     7
1.0868     7
          ..
1.4304     1
1.4350     1
1.4442     1
1.4389     1
1.0804     1
Name: count, Length: 3769, dtype: int64
```

In [13]:

```python
euro_to_dollar=euro_to_dollar[euro_to_dollar['US_dollar']!= '-']
euro_to_dollar['US_dollar'] = euro_to_dollar['US_dollar'].astype(float)
euro_to_dollar.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 6394 entries, 6455 to 0
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype         
---  ------     --------------  -----         
 0   Time       6394 non-null   datetime64[ns]
 1   US_dollar  6394 non-null   float64       
dtypes: datetime64[ns](1), float64(1)
memory usage: 149.9 KB
```

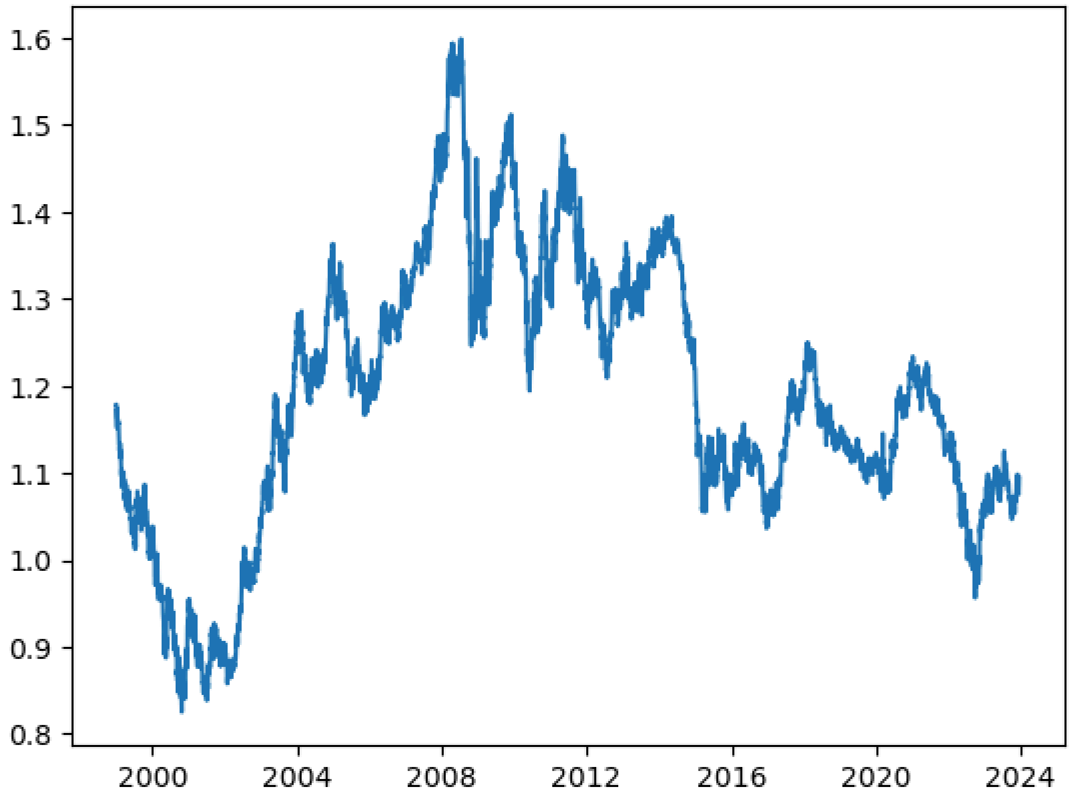
Rolling Mean

In [14]:

```python
import matplotlib.pyplot as plt
```

In [15]:
```
plt.plot(euro_to_dollar['Time'],euro_to_dollar['US_dollar'])
plt.show
```

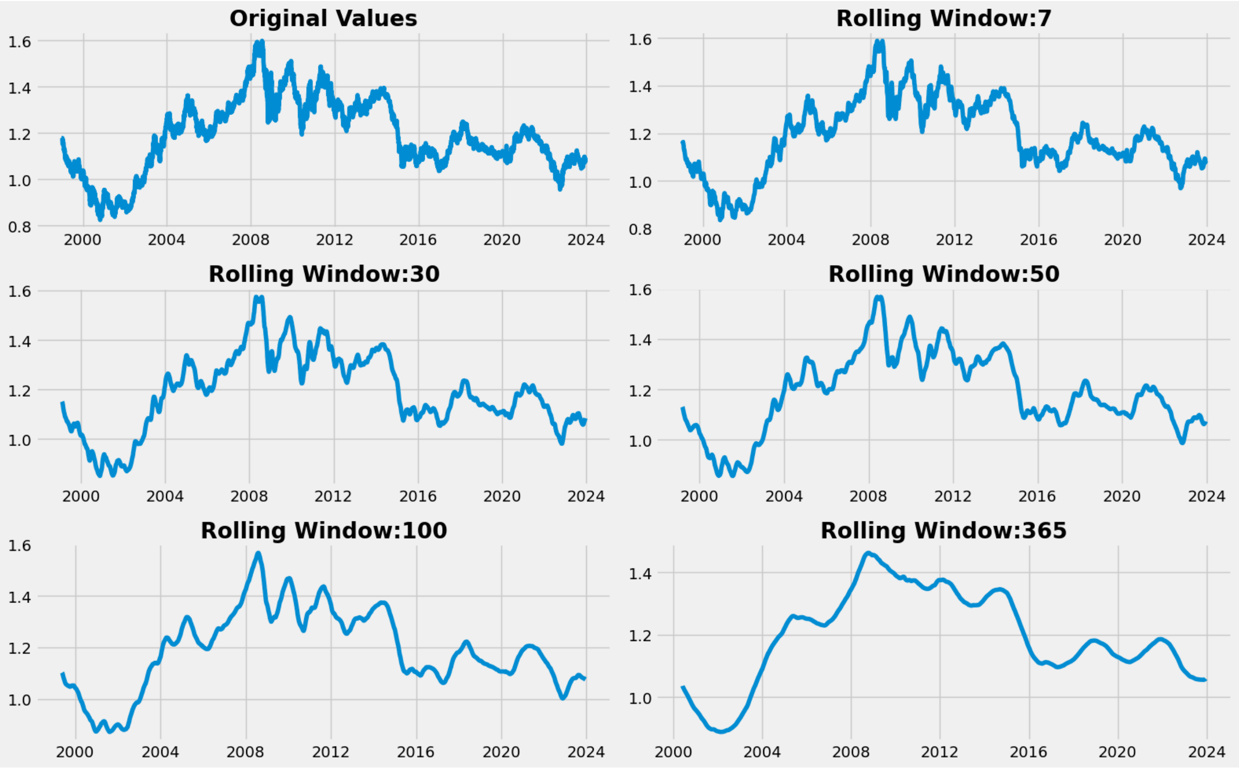
Out[15]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



If we look into the graph, we see that there are too many fluctuations in the graph -- tather than a smooth graph. These fluctuations are because of the fact that the graph represents daily variations in the exchange rate. As the rate fluctuates day to day; the graph also fluctuates.However, the graph indicates clear upward or downward trend when seen in longer run (months/years).

Here we will use concept of Rolling mean, where we wil focus on long-term trends while ignoring daily variations.

In [59]:
```
plt.figure(figsize=(16,10))

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()
```



#

Coming Up with an Idea

Here are few story ideas fo our data:

- We show how the Euro-dollar exchange rate has changed during the coronavirus pandemic. We can show the 2020 data and the 2016-2019 data as a baseline. We can use a line plot.
- We show how the euro-dollar exchange rate changed during the 2007-2008 financial crisis. We can also show the data for 2006 and 2009 for comparision. We can use a line plot.
- We show comparatively how the euro-dollar exchange rate changed under the last three US presidents(George W. Bush (2001-2009), Barack Obama (2009-2017) and Donald Trump (2017-2021).We can use a line plot.

In [17]:

```
euro_to_dollar["rolling_mean"]=euro_to_dollar["US_dollar"].rolling(30).mean()
euro_to_dollar
```

Out[17]:

	Time	US_dollar	rolling_mean
6455	1999-01-04	1.1789	NaN
6454	1999-01-05	1.1790	NaN
6453	1999-01-06	1.1743	NaN
6452	1999-01-07	1.1632	NaN
6451	1999-01-08	1.1659	NaN
...	...	...	...
4	2023-12-11	1.0757	1.080143
3	2023-12-12	1.0804	1.080760
2	2023-12-13	1.0787	1.081593
1	2023-12-14	1.0919	1.082453
0	2023-12-15	1.0946	1.083267

6394 rows × 3 columns

Storytelling Data Visualization

Financial crisis 2007-2008

In [18]:

```
financial_crisis = euro_to_dollar.copy()[(euro_to_dollar['Time'].dt.year >= 2006)
                                         & (euro_to_dollar['Time'].dt.year <= 2009)]
financial_crisis_7_8 = euro_to_dollar.copy()[(euro_to_dollar.Time.dt.year >= 2007)
                                             & (euro_to_dollar.Time.dt.year <= 2008)]
```

In [19]:

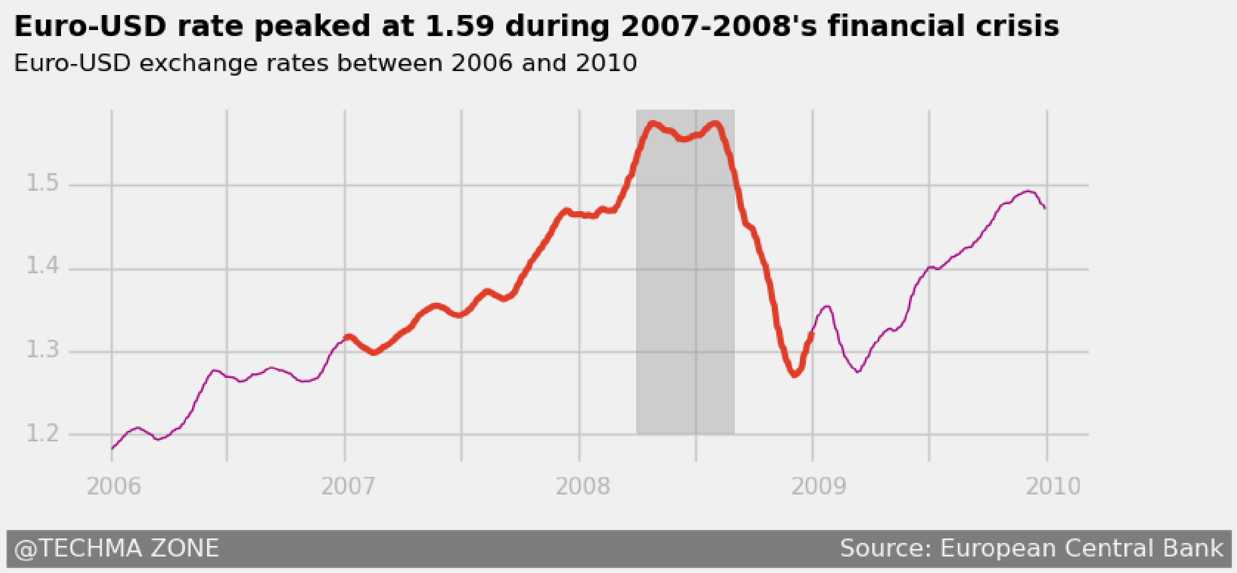
```
import matplotlib.style as style
```

In [20]:

```
style.use('fivethirtyeight')
```

In [64]:

```
## Adding the plot
fig,ax = plt.subplots(figsize= (8,3))
ax.plot(financial_crisis['Time'],
        financial_crisis['rolling_mean'],
        linewidth=1, color= '#A60785')

## Highlighting 2007-2008 period
ax.plot(financial_crisis_7_8['Time'], financial_crisis_7_8['rolling_mean'],
        linewidth= 3, color='#e23d28')

ax.set_xticklabels([])
x= 0.02
for year in ['2006' , '2007', '2008', '2009', '2010']:
    ax.text(x, -0.08, year, alpha = 0.5, fontsize=11, transform= plt.gca().transAxes,
    x += 0.22888

ax.set_yticklabels([])
y= 0.07
for rate in ['1.2', '1.3', '1.4','1.5']:
    ax.text(-.04, y, rate, alpha = 0.5, fontsize=11, transform= plt.gca().transAxes, c
    y += 0.2333

## Adding title and subtitle
ax.text(-0.05, 1.2, "Euro-USD rate peaked at 1.59 during 2007-2008's financial crisis"
        weight='bold', transform = plt.gca().transAxes)

ax.text(-0.05, 1.1, "Euro-USD exchange rates between 2006 and 2010",
        size=12, transform = plt.gca().transAxes)

## Adding a signature
ax.text(-0.05, -0.25, '@TECHMA ZONE' + ' '*80 + 'Source: European Central Bank',
        color='#f7f7f7', backgroundcolor='#7d7d7d',
        size='12', transform = plt.gca().transAxes)

ax.axvspan(xmin=pd.to_datetime("2008-04-1"), xmax = pd.to_datetime('2008-09-1'), ymin=
           alpha=0.3, color="gray")

plt.show()
```


Euro-USD rate peaked at 1.59 during 2007-2008's financial crisis

Euro-USD exchange rates between 2006 and 2010

@TECHMA ZONE Source: European Central Bank

Financial Crisis Impact on Euro-USD exchange rate:

The analysis of Exchnage rate during the 2007-2008 financial crisis revealed a notable fall in the USD worth, reflecting the uncertainty during that period. This underscores the interconnectedness of global financial markets and the impact of major economicdownturns on currency valuations.

COVID 19

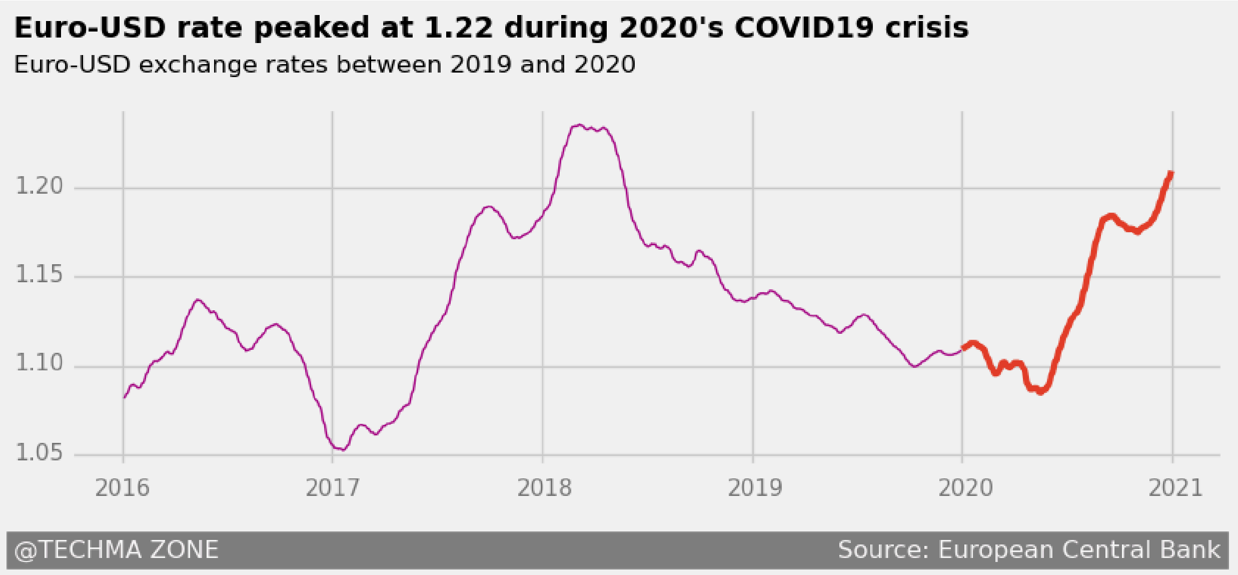
In [65]:

```
corona_crisis_20 = euro_to_dollar.loc[(euro_to_dollar['Time'] >= '2020-01-01')&
                                      (euro_to_dollar['Time']<='2020-12-31')]
corona_crisis= euro_to_dollar.loc[(euro_to_dollar['Time'] >= '2016-01-01')&
                                  (euro_to_dollar['Time']<='2019-12-31')]
```

In [68]:

```
## Adding plot
fig,ax = plt.subplots(figsize= (9,3))
```

```
ax.plot(corona_crisis['Time'],
        corona_crisis['rolling_mean'],
        linewidth=1, color= '#A60785')

ax.plot(corona_crisis_20['Time'],
        corona_crisis_20['rolling_mean'],
        linewidth= 3, color='#e23d28')

ax.set_xticklabels([])
x= 0.02
for year in ['2016' , '2017', '2018', '2019', '2020', '2021']:
    ax.text(x, -0.08, year, alpha = 0.5, fontsize=11, transform=plt.gca().transAxes)
    x += 0.183

ax.set_yticklabels([])
y= 0.02
for rate in ['1.05', '1.10', '1.15','1.20']:
    ax.text(-.05, y, rate, alpha = 0.5, fontsize=11, transform=plt.gca().transAxes)
    y += 0.248

## Adding title and subtitle
ax.text(-0.05, 1.2, "Euro-USD rate peaked at 1.22 during 2020's COVID19 crisis",
        weight='bold', transform = plt.gca().transAxes)

ax.text(-0.05, 1.1, "Euro-USD exchange rates between 2019 and 2020",
        size=12, transform = plt.gca().transAxes)

## Adding signature
ax.text(-0.05, -0.25, "@TECHMA ZONE" + ' '*80 + 'Source: European Central Bank',
        color='#f7f5f6', backgroundcolor='#7d7d7d',
        size='12', transform = plt.gca().transAxes)

plt.show()
```



COVID 19 impact on Euro-USD exchange rate:

Similarly, the storytelling of the COVID-19 crisis demonstrated another significant peak in the dollar rate, indicating the volatility and market responses during times of unprecedented global challenges. This emphasizes the importance of adaptability and resilience in navigating economic crises.

Comparing the three US presidential tenures

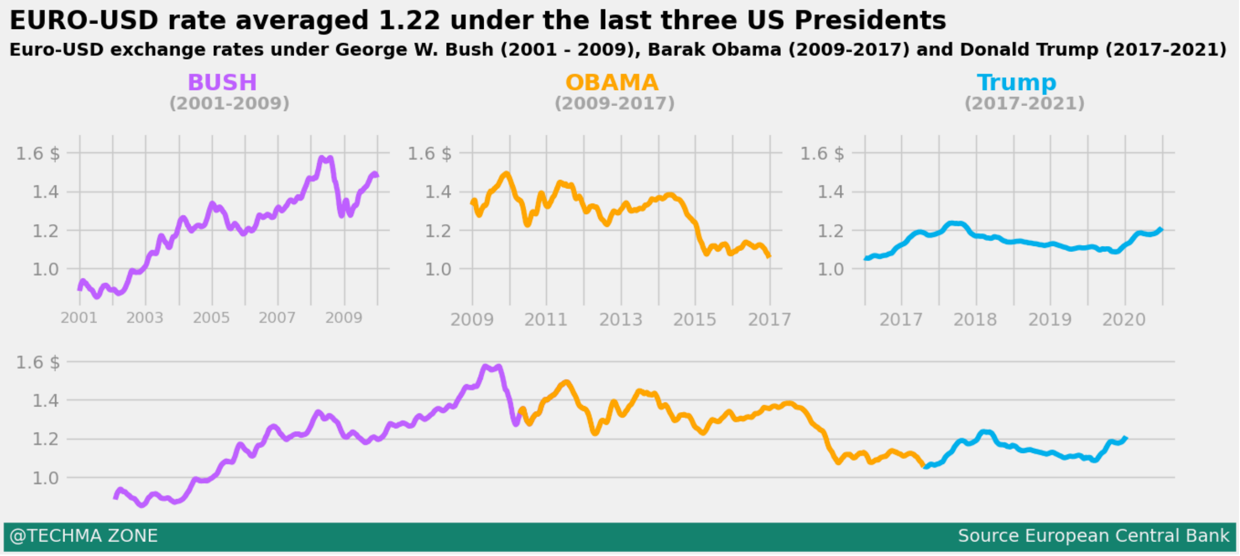
In [70]:

```
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 [71]:

```
plt.figure(figsize=(14, 8))
ax1 = plt.subplot(3,3,1)
ax2 = plt.subplot(3,3,2)
ax3= plt.subplot(3,3,3)
ax4=plt.subplot(3,1,2)
axes = [ax1, ax2, ax3, ax4]

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 =0.4)

## Ax1 Bush:
ax1.plot(bush['Time'], bush['rolling_mean'], color='#BF5FFF')
ax1.set_xticklabels(['', '2001','', '2003','','2005','','2007',
 '','2009'], alpha=0.3, size='12')

ax1.text(0.11, 2.45, 'BUSH', fontsize=18, weight='bold', color='#BF5FFF',
         transform = plt.gca().transAxes)
ax1.text(0.093, 2.34, '(2001-2009)', weight='bold', alpha=0.3,
         transform = plt.gca().transAxes)

##Ax2 Obama:
ax2.plot(obama['Time'], obama['rolling_mean'], color='#ffa500')
ax2.set_xticklabels(['', '2009','', '2011','','2013','','2015',
 '','2017'], alpha=0.3)
ax2.text(0.45, 2.45, 'OBAMA', fontsize=18, weight='bold', color='#ffa500',
         transform = plt.gca().transAxes)
ax2.text(0.44, 2.34, '(2009-2017)', weight='bold', alpha=0.3,
         transform = plt.gca().transAxes)

##Ax3 Trump:
ax3.plot(trump['Time'], trump['rolling_mean'], color='#00B2EE')
ax3.set_xticklabels(['', '2017','', '2018','','2019','','2020',
 '','2021'], alpha=0.3)
ax3.text(0.82, 2.45, 'Trump', fontsize=18, weight='bold', color='#00B2EE',
         transform = plt.gca().transAxes)
```

```
ax3.text(0.808, 2.34, '(2017-2021)', weight='bold', alpha=0.3,
         transform = plt.gca().transAxes)

##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='#00B2EE')
ax4.set_xticks([])

## Adding a title
ax1.text(-0.05, 2.8, 'EURO-USD rate averaged 1.22 under the last three US Presidents',
         fontsize=20, weight='bold', transform =plt.gca().transAxes)
##Adding a subtitle
ax1.text(-0.05, 2.65, "Euro-USD exchange rates under George W. Bush (2001 - 2009), Bar
         fontsize=14, weight='bold', transform =plt.gca().transAxes)

#Addign a signature
ax4.text(-0.05,-0.15, '@TECHMA ZONE' + ' '*143 +'Source European Central Bank',
 fontsize=14, color='#f7f5f6', backgroundcolor='#178272', transform = plt.gca().transA

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



EURO-USD rate averaged 1.22 under the last three US Presidents

Euro-USD exchange rates under George W. Bush (2001 - 2009), Barak Obama (2009-2017) and Donald Trump (2017-2021)

@TECHMA ZONE Source European Central Bank

Comparision bwtween three US Presidents' on basis of Euro-USD exchange rate:

The analysis of the Euro-USD Exchange rates suggests that the USD suffered increasing loss during the tenure of Bush, predominantly beacause of the War-on-terror waged by the Bush administration which resulted in serious loss in worth of dollars. The situation got worsened during the 2007-08 Financial crisis. Moving to Obama's tenure, the dollar gained some worth and got back to the early Bush administration's level. After that, the exchange rate didn't fluctuated much and pretty much remained same during the time of Trump's rule.

Conclusion:

Our analysis of Euro-USD Exchnage rate spans over two and a half decades. We developed three distinct perspectives through which we analyzed the Euro-USD Exchange rate; the 2007-2008 financial crisis, the COVID-19 pandemic and the presidencies of Trump, Obama, and Bush. After conducting thorough analysis and storytelling of Euro to USD conversion data from 1999 to 2023 we got useful insights and observed significant impact of geopolitical, economic and social events over the past two and a half decades on the Euro-USD Exchange rates.