In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
df = pd.read_csv('euro-daily-hist_1999_2020.csv')
```

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

Out[3]:

	Period\Unit:	[Australian dollar]	[Bulgarian lev]	[Brazilian real]	[Canadian dollar]	[Swiss franc]	[Chinese yuan renminbi]	[Cypriot pound]	[Czech koruna]	[Danish krone]	...
0	2021-01-08	1.5758	1.9558	6.5748	1.5543	1.0827	7.9184	NaN	26.163	7.4369	...
1	2021-01-07	1.5836	1.9558	6.5172	1.5601	1.0833	7.9392	NaN	26.147	7.4392	...
2	2021-01-06	1.5824	1.9558	6.5119	1.5640	1.0821	7.9653	NaN	26.145	7.4393	...
3	2021-01-05	1.5927	1.9558	6.5517	1.5651	1.0803	7.9315	NaN	26.227	7.4387	...
4	2021-01-04	1.5928	1.9558	6.3241	1.5621	1.0811	7.9484	NaN	26.141	7.4379	...

5 rows × 41 columns

In [4]:
```
df.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
```

In [5]:
```
df.rename(columns = {'[US dollar ]':'USD','Period\\Unit:':'Time'}, inplace=True)
```

In [6]:
```
df.Time = pd.to_datetime(df.Time)
```

In [7]:
```
df.sort_values('Time', inplace=True)
```

In [8]:
```
df.reset_index(inplace=True, drop=True)
```

In [9]:
```
euro_to_dollar = df[['Time','USD']].copy()
```

In [10]:
```
euro_to_dollar
```

Out[10]:

	Time	USD
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 × 2 columns

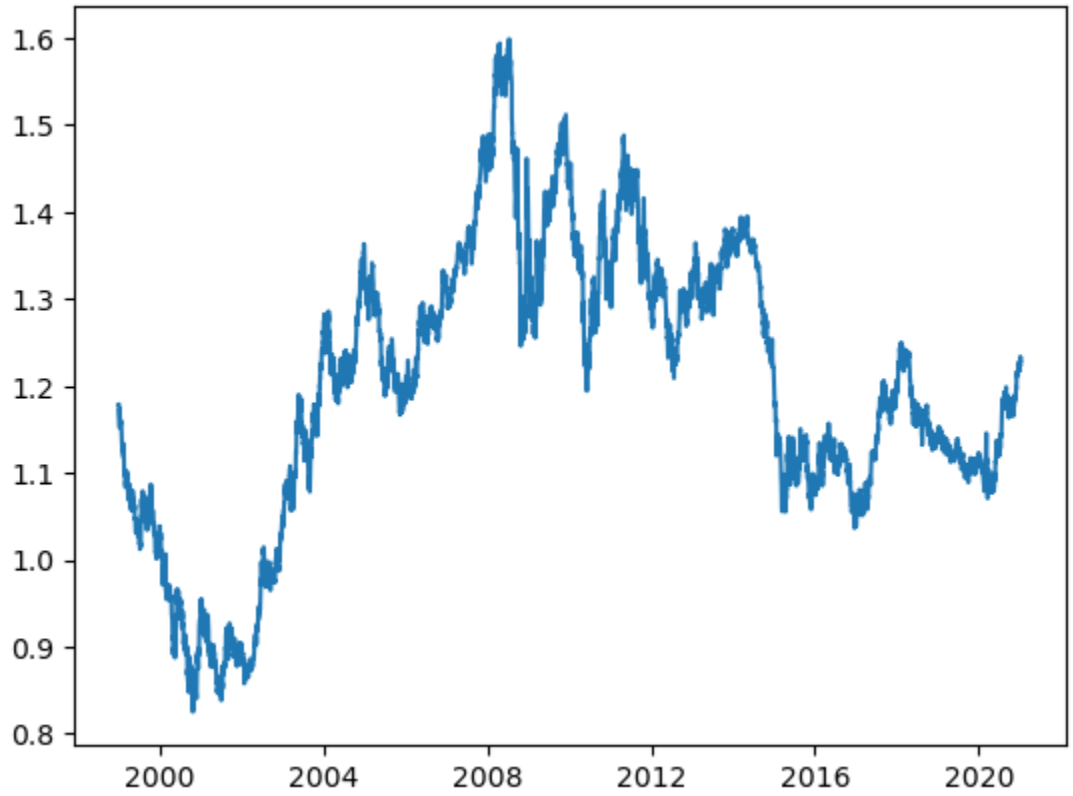
In [11]:
```
euro_to_dollar['USD'].value_counts()
```

Out[11]:
```
USD
-         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: count, Length: 3528, dtype: int64
```

In [12]:
```
euro_to_dollar = euro_to_dollar[euro_to_dollar.USD != '-']
```

In [13]:
```
euro_to_dollar['USD'] = euro_to_dollar['USD'].astype(float)
```

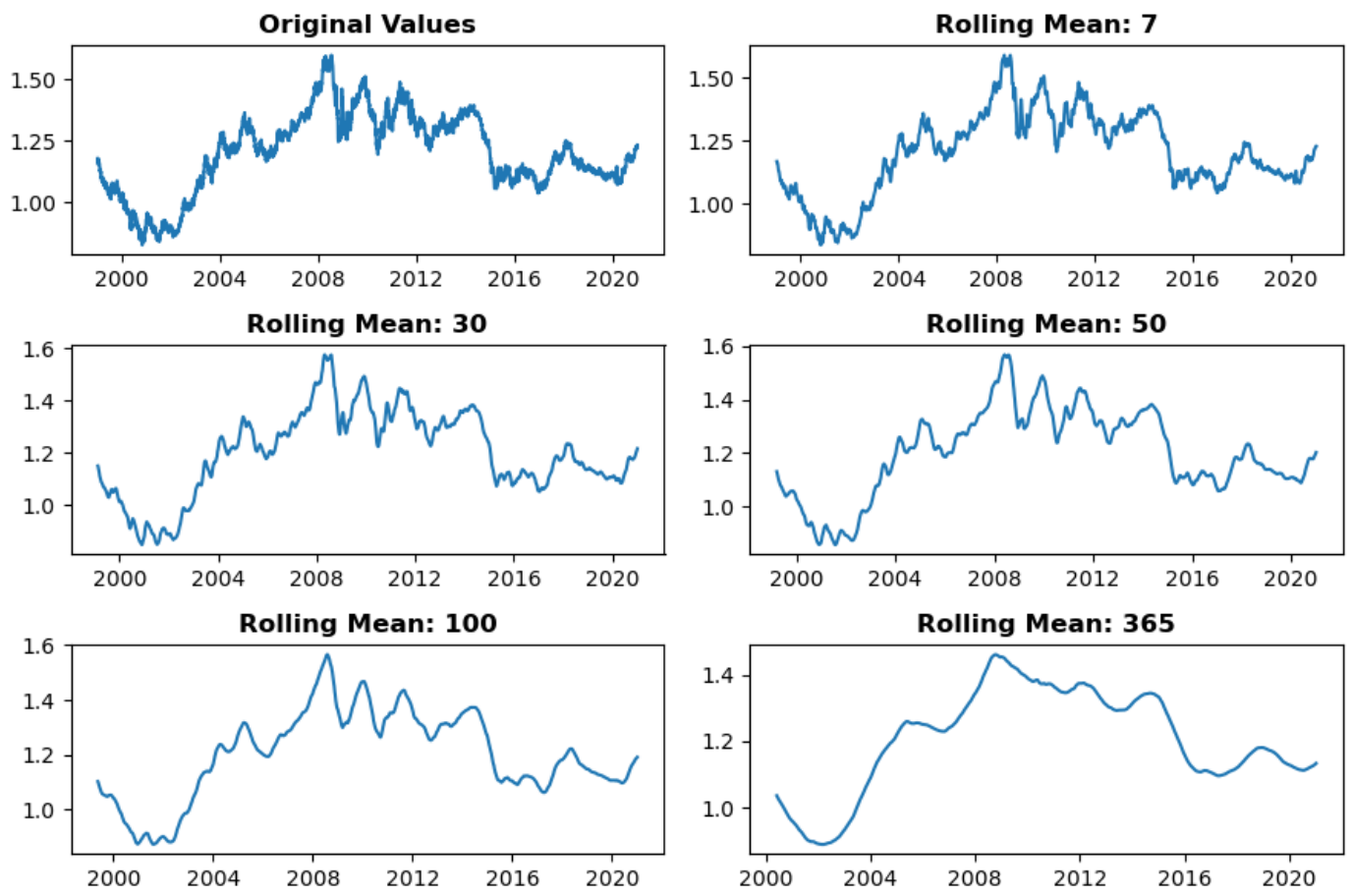
In [14]:
```
plt.plot(euro_to_dollar['Time'], euro_to_dollar['USD'])
plt.show()
```



In [15]:
```
euro_to_dollar['USD'].rolling(7).mean().head(8)
```

Out[15]:
```
0         NaN
1         NaN
2         NaN
3         NaN
4         NaN
5         NaN
6    1.167171
7    1.166529
Name: USD, dtype: float64
```

In [16]:
```
plt.figure(figsize=(9,6))

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

for i, r_mean in zip(range(2,7), [7,30,50,100,365]):
    plt.subplot(3,2,i)
    plt.plot(euro_to_dollar['Time'], euro_to_dollar['USD'].rolling(r_mean).mean())
    plt.title('Rolling Mean: ' + str(r_mean), weight='bold')
plt.tight_layout()
```



In [17]:
```
euro_to_dollar["rolling_mean"] = euro_to_dollar['USD'].rolling(30).mean()
euro_to_dollar
```

Out[17]:

	Time	USD	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

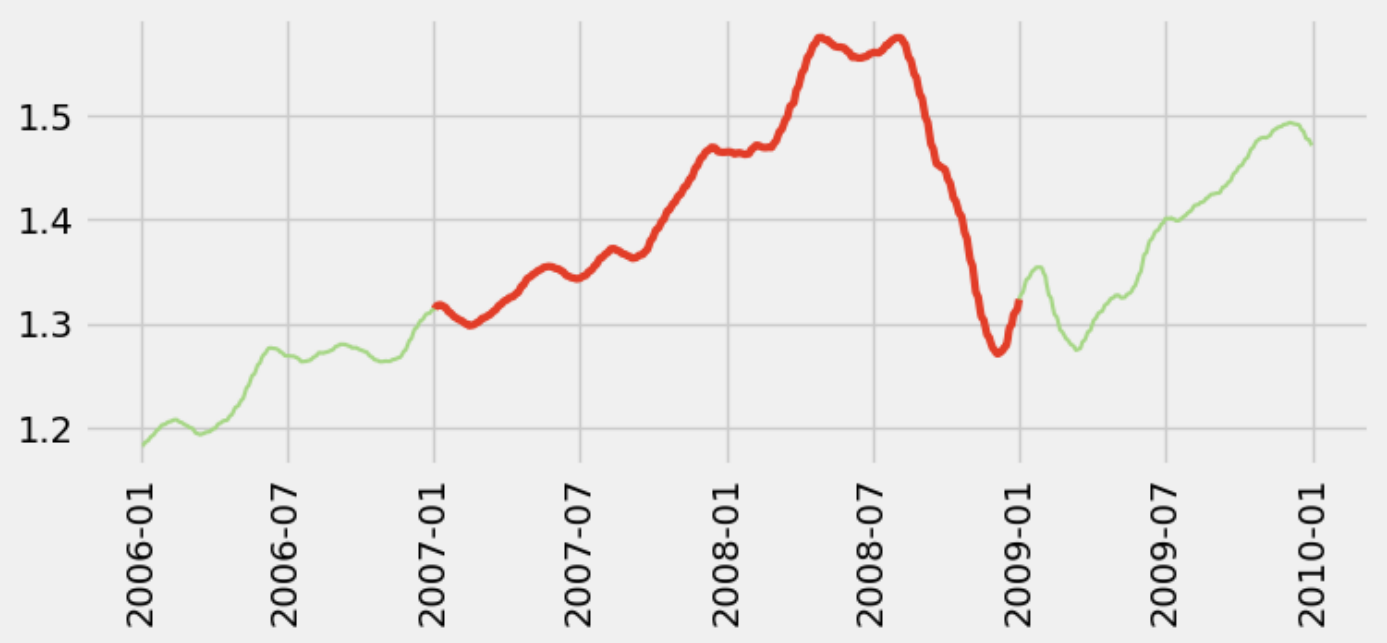
5637 rows × 3 columns

Let's Visualize the data of financial crisis occured in 2007-2008

In [18]:
```
financial_crisis = euro_to_dollar[(euro_to_dollar.Time.dt.year >= 2006) & (euro_to_dolla
financial_crisis_7_8 = euro_to_dollar[(euro_to_dollar.Time.dt.year >= 2007) & (euro_to_d
```

In [19]:
```
import matplotlib.style as style
style.use('fivethirtyeight')
```

In [20]:
```
# Adding a plot
fig, ax = plt.subplots(figsize = (8,3))

ax.plot(financial_crisis.Time, financial_crisis['rolling_mean'], linewidth = 1.5, color=

#Let's highlight the 2007-2008 period
ax.plot(financial_crisis_7_8.Time, financial_crisis_7_8['rolling_mean'], linewidth = 3,

plt.xticks(rotation=90)

plt.show()
```



In [21]:
```
fig, ax = plt.subplots(figsize = (8,3))

ax.plot(financial_crisis.Time, financial_crisis['rolling_mean'], linewidth = 1.5, color=
ax.plot(financial_crisis_7_8.Time, financial_crisis_7_8['rolling_mean'], linewidth = 3,

#plt.xticks(rotation=90)
ax.set_xticklabels(['2006','','2007','','2008','','2009','','2010'])

plt.show()
```

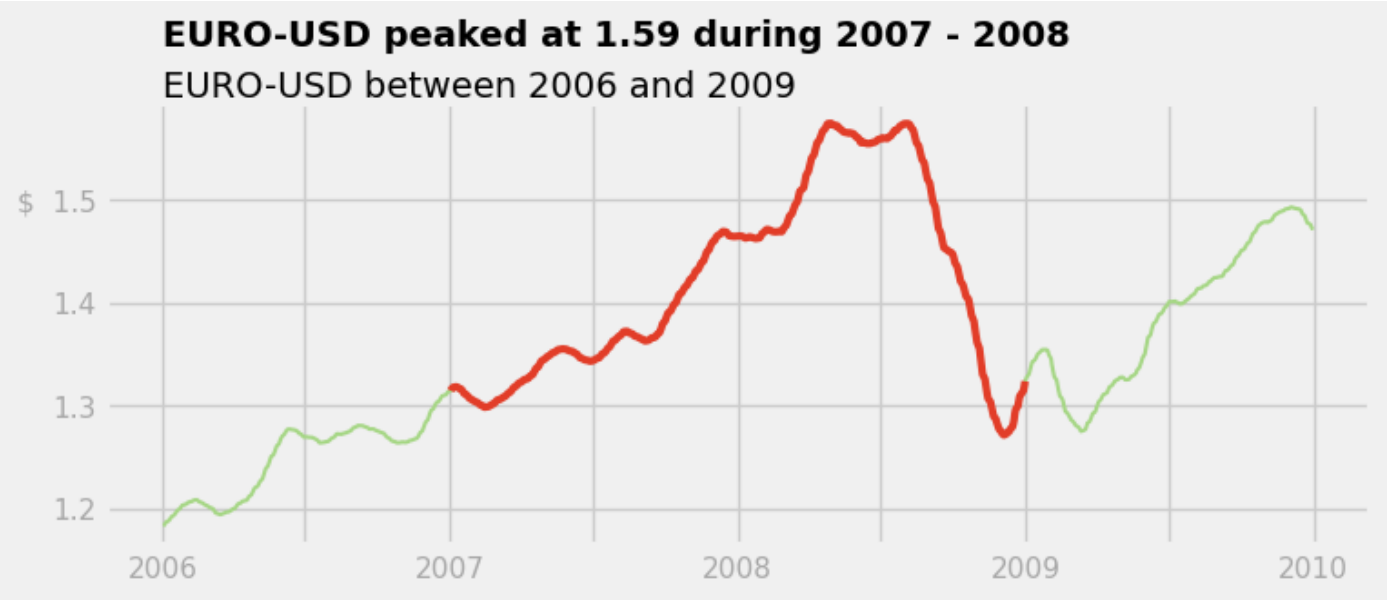
In [22]:
```
fig, ax = plt.subplots(figsize = (8,3))
```

```
ax.plot(financial_crisis.Time, financial_crisis['rolling_mean'], linewidth = 1.5, color=
ax.plot(financial_crisis_7_8.Time, financial_crisis_7_8['rolling_mean'], linewidth = 3,

#plt.xticks(rotation=90)
ax.set_xticklabels(['2006','','2007','','2008','','2009','','2010'], alpha=0.3, fontdict

plt.show()
```

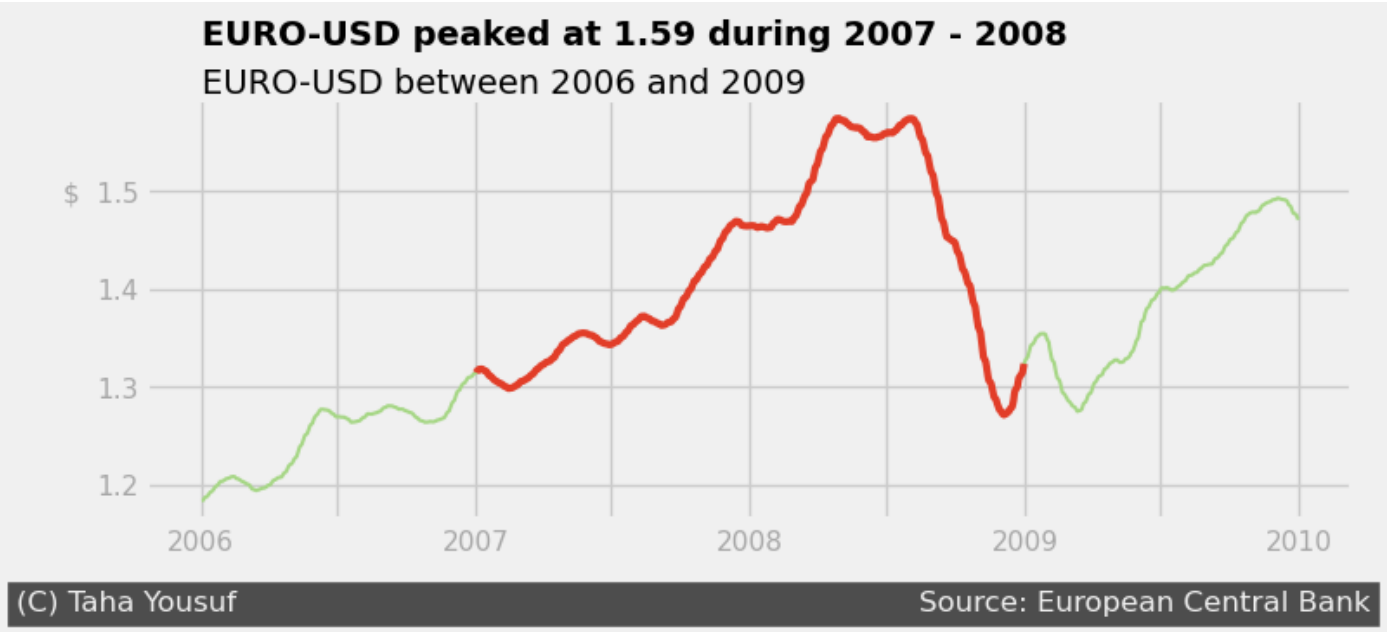
In [23]:
```
fig, ax = plt.subplots(figsize = (8,3))

ax.plot(financial_crisis.Time, financial_crisis['rolling_mean'], linewidth = 1.5, color=
ax.plot(financial_crisis_7_8.Time, financial_crisis_7_8['rolling_mean'], linewidth = 3,

#plt.xticks(rotation=90)
ax.set_xticklabels(['2006','','2007','','2008','','2009','','2010'], alpha=0.3, fontdict
ax.set_yticklabels(['','1.2','1.3','1.4','$  1.5'], alpha=0.3, fontdict={'fontsize':11})
plt.show()
```

In [24]:
```
fig, ax = plt.subplots(figsize = (8,3))

ax.plot(financial_crisis.Time, financial_crisis['rolling_mean'], linewidth = 1.5, color=
ax.plot(financial_crisis_7_8.Time, financial_crisis_7_8['rolling_mean'], linewidth = 3,

#plt.xticks(rotation=90)
ax.set_xticklabels(['2006','','2007','','2008','','2009','','2010'], alpha=0.3, fontdict
ax.set_yticklabels(['','1.2','1.3','1.4','$  1.5'], alpha=0.3, fontdict={'fontsize':11})

# Adding a title and a subtitle
ax.text(pd.to_datetime('2006-01-1'), 1.65, 'EURO-USD peaked at 1.59 during 2007 - 2008',
```

```
ax.text(pd.to_datetime('2006-01-1'), 1.6, 'EURO-USD between 2006 and 2009')
plt.show()
```

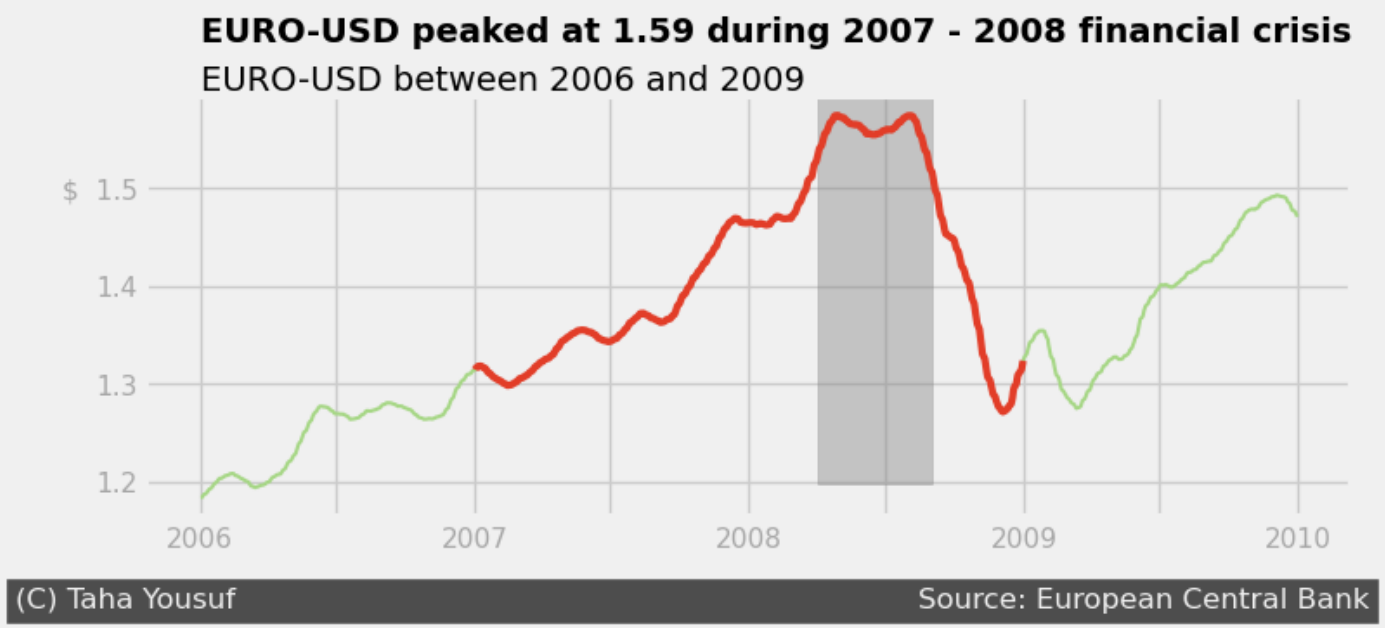


In [25]:
```
fig, ax = plt.subplots(figsize = (8,3))

ax.plot(financial_crisis.Time, financial_crisis['rolling_mean'], linewidth = 1.5, color=
ax.plot(financial_crisis_7_8.Time, financial_crisis_7_8['rolling_mean'], linewidth = 3,

#plt.xticks(rotation=90)
ax.set_xticklabels(['2006','','2007','','2008','','2009','','2010'], alpha=0.3, fontdict
ax.set_yticklabels(['','1.2','1.3','1.4','$  1.5'], alpha=0.3, fontdict={'fontsize':11})

ax.text(pd.to_datetime('2006-01-1'), 1.65, 'EURO-USD peaked at 1.59 during 2007 - 2008',
ax.text(pd.to_datetime('2006-01-1'), 1.6, 'EURO-USD between 2006 and 2009')

#Adding a signature
ax.text(pd.to_datetime('2005-4-30'), 1.07, '(C) Taha Yousuf'+ ' '*75 + 'Source: European

plt.show()
```



In [26]:
```
fig, ax = plt.subplots(figsize = (8,3))

ax.plot(financial_crisis.Time, financial_crisis['rolling_mean'], linewidth = 1.5, color=
ax.plot(financial_crisis_7_8.Time, financial_crisis_7_8['rolling_mean'], linewidth = 3,
```

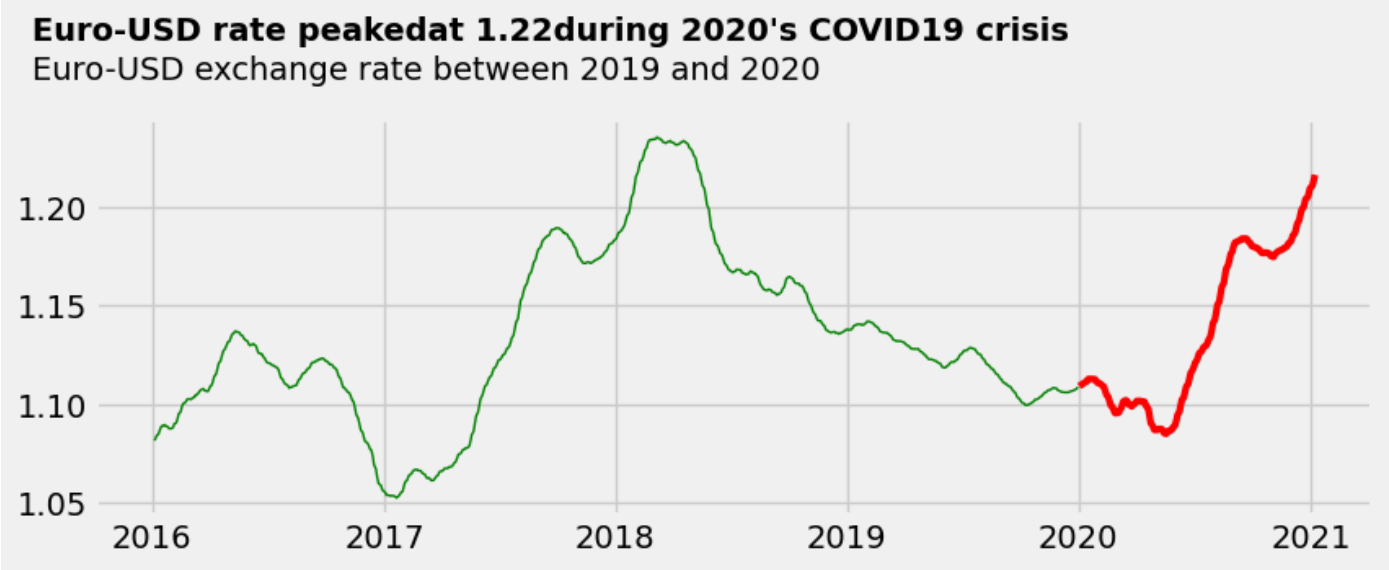
```
#plt.xticks(rotation=90)
ax.set_xticklabels(['2006','','2007','','2008','','2009','','2010'], alpha=0.3, fontdict
ax.set_yticklabels(['','1.2','1.3','1.4','$  1.5'], alpha=0.3, fontdict={'fontsize':11})

ax.text(pd.to_datetime('2006-01-1'), 1.65, 'EURO-USD peaked at 1.59 during 2007 - 2008 f
ax.text(pd.to_datetime('2006-01-1'), 1.6, 'EURO-USD between 2006 and 2009')

ax.text(pd.to_datetime('2005-4-30'), 1.07, '(C) Taha Yousuf'+ ' '*75 + 'Source: European
ax.axvspan(xmin =pd.to_datetime('2008-04-1'), xmax=pd.to_datetime('2008-09-1'), color='g

plt.show()
```



Now let's Visualize Covid 19 Period

In [27]:
```
corona_crisis_20 = euro_to_dollar.loc[(euro_to_dollar['Time'] >= '2020-01-01') & (euro_t
corona_crisis = euro_to_dollar.loc[(euro_to_dollar['Time'] >= '2016-01-01') & (euro_to_d
```

In [28]:
```
import matplotlib.style as style
style.use('fivethirtyeight')

# Adding the plot
fig, ax = plt.subplots (figsize=(9, 3))
ax.plot(corona_crisis [ 'Time'], corona_crisis['rolling_mean'], linewidth =1, color='gre

# Highlighting the 2007-2008 period
ax.plot(corona_crisis_20[ 'Time'], corona_crisis_20['rolling_mean'], linewidth =3, color

#Adding a title and subtitle
ax.text(-.05, 1.2, "Euro-USD rate peakedat 1.22during 2020's COVID19 crisis", weight="bo
ax.text(-.05, 1.1, "Euro-USD exchange rate between 2019 and 2020 ",  transform = plt.gca

plt.show()
```

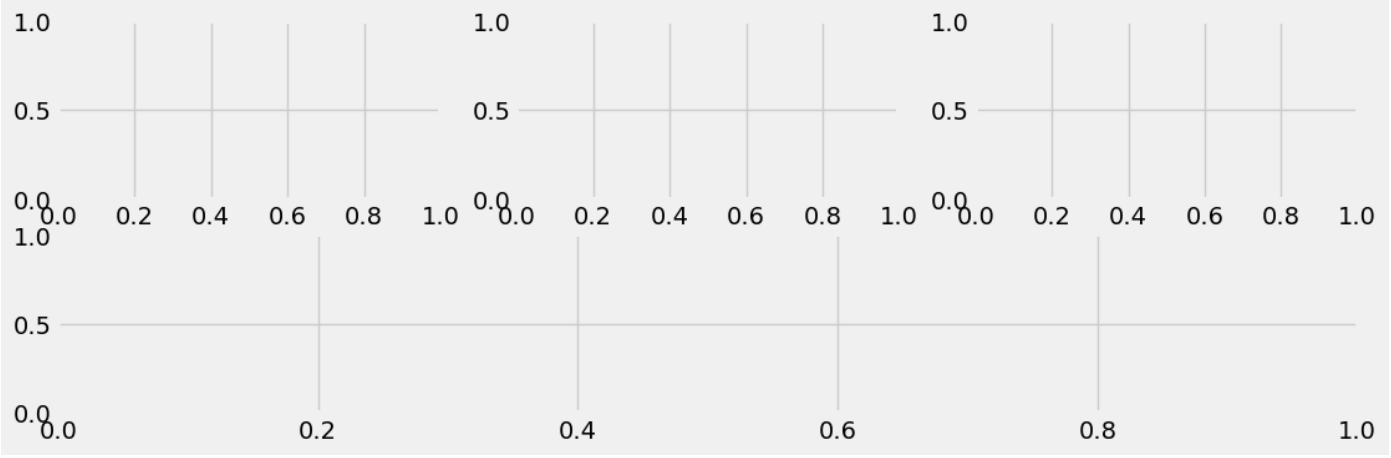


The Three US President Example

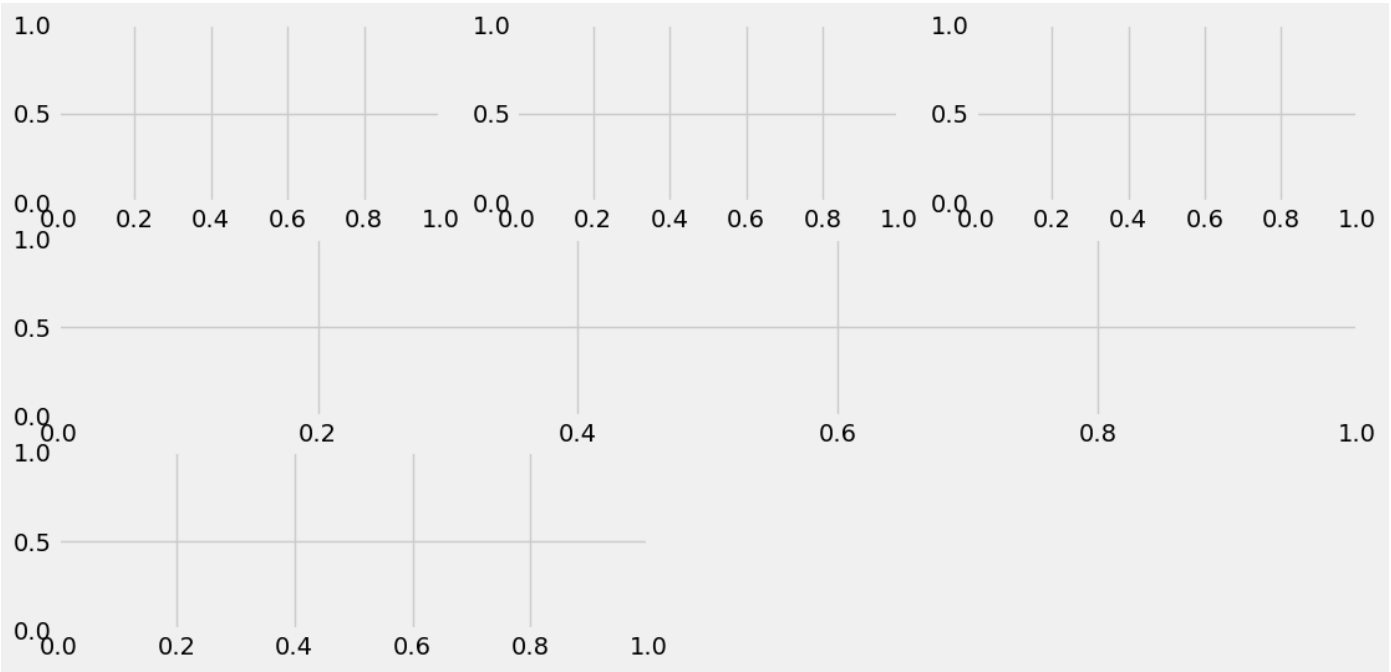
In [29]:
```
bush_obama_trump = euro_to_dollar.copy()[(euro_to_dollar['Time'].dt.year >= 2001) & (eur
#bush = bush_obama_trump.copy()[bush_obama_trump[ 'Time'].dt.year < 2009]

#obama = bush_obama_trump.copy()[(bush_obama_trump[ 'Time'].dt.year >= 2009) & (bush_oba

#trump = bush_obama_trump.copy()[(bush_obama_trump[ 'Time'].dt.year >= 2017) & (bush_oba


bush = euro_to_dollar.copy()[(euro_to_dollar['Time'].dt.year >= 2001) & (euro_to_dollar[
obama = euro_to_dollar.copy()[(euro_to_dollar.Time.dt.year >= 2009) & (euro_to_dollar.Ti
trump = euro_to_dollar.copy()[(euro_to_dollar.Time.dt.year >= 2017) & (euro_to_dollar.Ti
```

Below, you'll notice we used matplotlib's functional approach to build the graph. We use this approach beacuse it offers more flexibility in arranging the subplots: 1) We first build three of the graphs on a 2-by-3 grid(this grid should have six subplots, but we only build three, the bottom row remains empty) 2) We then build only the bottom graph of a 2-by-1 grid (this grid should have two subplots, The top row remains empty) 3) The two grids are merged and we end up with three graphs on the top row and one graph on the bottom row.

In [30]:
```
# Adding the FiveThirtyEight style
style.use('fivethirtyeight')
#Adding the sublpots
plt.figure(figsize=(12, 6))
#pattern1
ax1 = plt.subplot(3,3,1) #row,col, index
ax2 = plt.subplot(3,3,2)
ax3 = plt.subplot(3,3,3)
#pattern 2
ax4 = plt.subplot(3,1,2) #row,col, index
```



In [31]:
```
# Adding the FiveThirtyEight style
style.use('fivethirtyeight')
#Adding the sublpots
plt.figure(figsize=(12, 6))
#pattern1
ax1 = plt.subplot(3,3,1) #row,col, index
ax2 = plt.subplot(3,3,2)
ax3 = plt.subplot(3,3,3)
#pattern 2
ax4 = plt.subplot(3,1,2) #row,col, index
#pattern 3
ax5 = plt.subplot(3,2,5) #row,col, index
```



In [32]:
```
# Adding the FiveThirtyEight style
style.use('fivethirtyeight')
# Adding the subplots
plt.figure(figsize=(14, 8))

# Pattern 1:
ax1 = plt.subplot(3, 3, 1)
ax2 = plt.subplot(3, 3, 2)
ax3 = plt.subplot(3, 3, 3)
# Pattern 2
ax4 = plt.subplot(3, 1, 2)
axes = [ax1, ax2, ax3, ax4]
# Adjusting each subplot as needed
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=
ax1.text(0.11, 2.45, 'BUSH', fontsize=20, weight='bold', color='#BF5FFF', transform = pl
ax1.text(0.093, 2.34, '(2001-2009)', weight='bold', alpha=0.3, transform = plt.gca().tra

# Ax2: Obama
ax2.plot(obama['Time'], obama['rolling_mean'], color='#ffa500')
ax2.set_xticklabels([" ", '2009', " ", '2011', " ", '2013', " ", '2015', " ", '2017'], a
ax2.text(0.45, 2.45, 'Obama', fontsize=20, weight='bold', color='#ffa500', transform = p
ax2.text(0.44, 2.34, '(2009-2017)', weight='bold', alpha=0.3, transform = plt.gca().tran

# Ax3: Trump
ax3.plot(trump['Time'], trump['rolling_mean'], color='#00B2EE')
ax3.set_xticklabels([" ", '2017', " ", '2018', " ", '2019', " ", '2020', " ", '2021'], a
ax3.text(0.82, 2.45, 'Trump', fontsize=20, weight='bold', color='#00B2EE', transform = p
ax3.text(0.808, 2.34, '(2017-2021)', weight='bold', alpha=0.3, transform = plt.gca().tra

# 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')

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