heatmap

November 1, 2024

[1]:
```
#https://towardsdatascience.com/heatmap-basics-with-pythons-seaborn-fb92ea280a6c
#The idea is straightforward, replace numbers with colors.
#Now, this visualization style came a long way from simple color-coded
#tables, it became widely used with geospatial data,
#and its commonly applied for describing density or intensity of variables,
#visualize patterns, variance, and even anomalies.
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
import numpy as np
```

[2]:
```
# read file
df = pd.read_csv('Foreign_Exchange_Rates.csv')
#print(df)
df = pd.read_csv('Foreign_Exchange_Rates.csv',
                 usecols=[1,7], names=['DATE', 'CAD_USD'],
                 skiprows=1, index_col=0, parse_dates=[0])
df
```

[2]:
```
            CAD_USD
DATE
2000-01-03   1.4465
2000-01-04   1.4518
2000-01-05   1.4518
2000-01-06   1.4571
2000-01-07   1.4505
…              …
2019-12-25       ND
2019-12-26   1.3124
2019-12-27   1.3073
2019-12-30   1.3058
2019-12-31   1.2962

[5217 rows x 1 columns]
```

[3]:
```
df['CAD_USD'] = pd.to_numeric(df.CAD_USD, errors='coerce')
df.dropna(inplace=True)
```

```
print(df)
```

```
            CAD_USD
DATE
2000-01-03   1.4465
2000-01-04   1.4518
2000-01-05   1.4518
2000-01-06   1.4571
2000-01-07   1.4505
…             …
2019-12-24   1.3160
2019-12-26   1.3124
2019-12-27   1.3073
2019-12-30   1.3058
2019-12-31   1.2962

[5019 rows x 1 columns]
```

[4]:
```
# create a copy of the dataframe, and add columns for month and year
df_m = df.copy()
df_m['month'] = [i.month for i in df_m.index]
df_m['year'] = [i.year for i in df_m.index]
# group by month and year, get the average
df_m = df_m.groupby(['month', 'year']).mean()
print(df_m)
```

```
              CAD_USD
month year
1     2000  1.448600
      2001  1.503200
      2002  1.599714
      2003  1.541448
      2004  1.295755
…              …
12    2015  1.371255
      2016  1.333919
      2017  1.276870
      2018  1.343611
      2019  1.316895

[240 rows x 1 columns]
```

[5]:
```
df_m = df_m.unstack(level=0)
print(df_m)
```

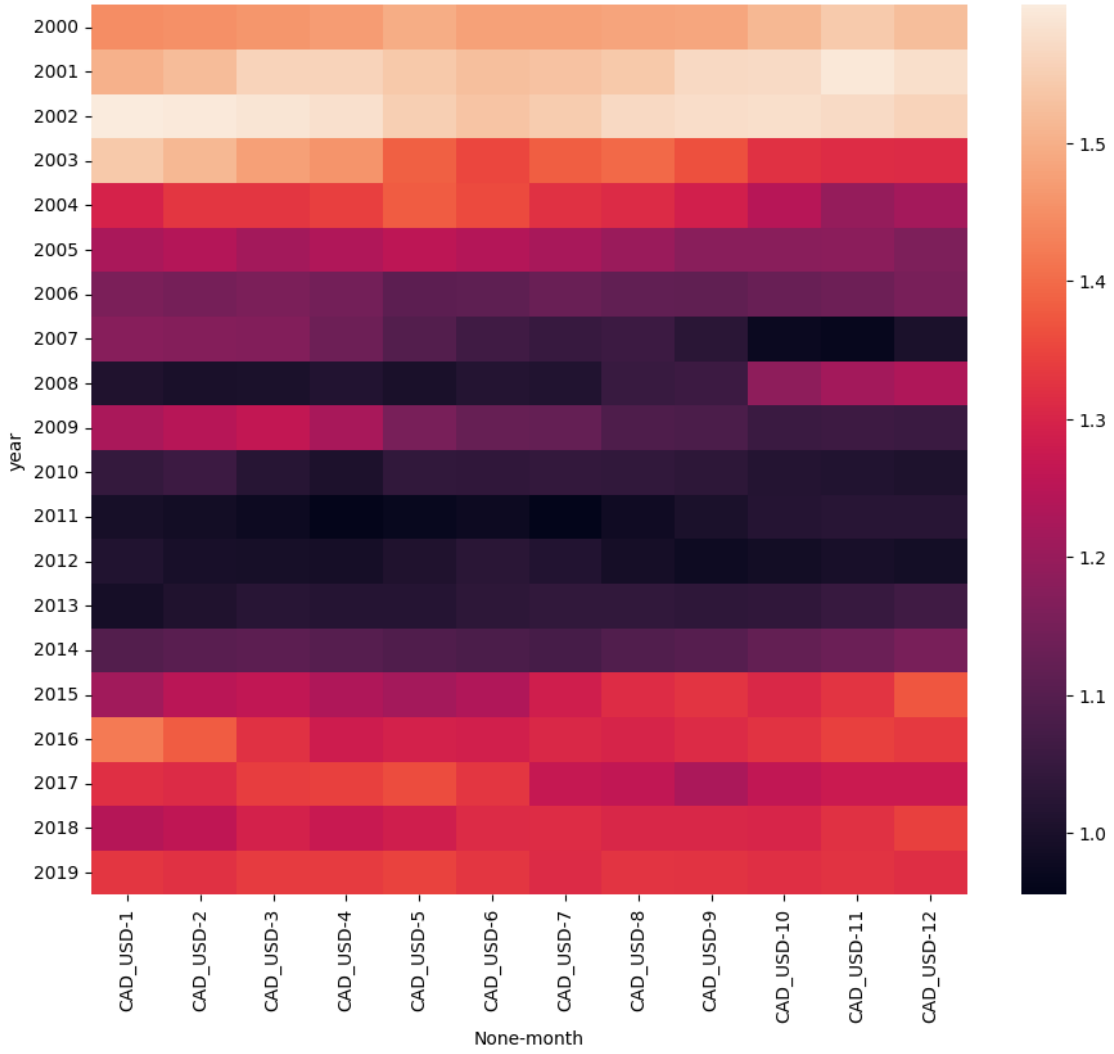
```
        CAD_USD                                                        \
month         1         2         3         4         5         6         7
year
```

2000	1.448600	1.451210	1.460774	1.468875	1.495736	1.477045	1.477785
2001	1.503200	1.521563	1.558741	1.557767	1.541050	1.524538	1.530790
2002	1.599714	1.596400	1.587743	1.581486	1.550155	1.531840	1.545550
2003	1.541448	1.512147	1.476081	1.458205	1.383957	1.352510	1.382091
2004	1.295755	1.329895	1.328578	1.341973	1.378860	1.357841	1.322505
2005	1.224835	1.240053	1.216026	1.235900	1.255529	1.240168	1.222855
2006	1.157165	1.148895	1.157309	1.144105	1.109991	1.113727	1.129445
2007	1.176262	1.170989	1.168159	1.134986	1.095086	1.065105	1.050186
2008	1.009943	0.998555	1.002943	1.013718	0.999305	1.016624	1.012964
2009	1.224820	1.245200	1.264518	1.224182	1.152785	1.126355	1.122861
2010	1.043811	1.057211	1.022900	1.005209	1.040280	1.037623	1.042229
2011	0.993945	0.987637	0.976561	0.957952	0.968043	0.976645	0.955315
2012	1.012985	0.996745	0.993773	0.992824	1.009732	1.028000	1.014200
2013	0.992057	1.009784	1.024424	1.018673	1.019559	1.031400	1.040214
2014	1.094010	1.105442	1.110681	1.099209	1.089386	1.083038	1.073918
2015	1.212190	1.249905	1.261832	1.233682	1.217640	1.236495	1.286314
2016	1.420811	1.379690	1.322639	1.281814	1.294529	1.289405	1.305235
2017	1.318305	1.310916	1.338700	1.343705	1.360573	1.329486	1.269040
2018	1.242905	1.258821	1.293255	1.273162	1.286627	1.312452	1.313343
2019	1.330045	1.320872	1.337052	1.337814	1.345977	1.328870	1.310523

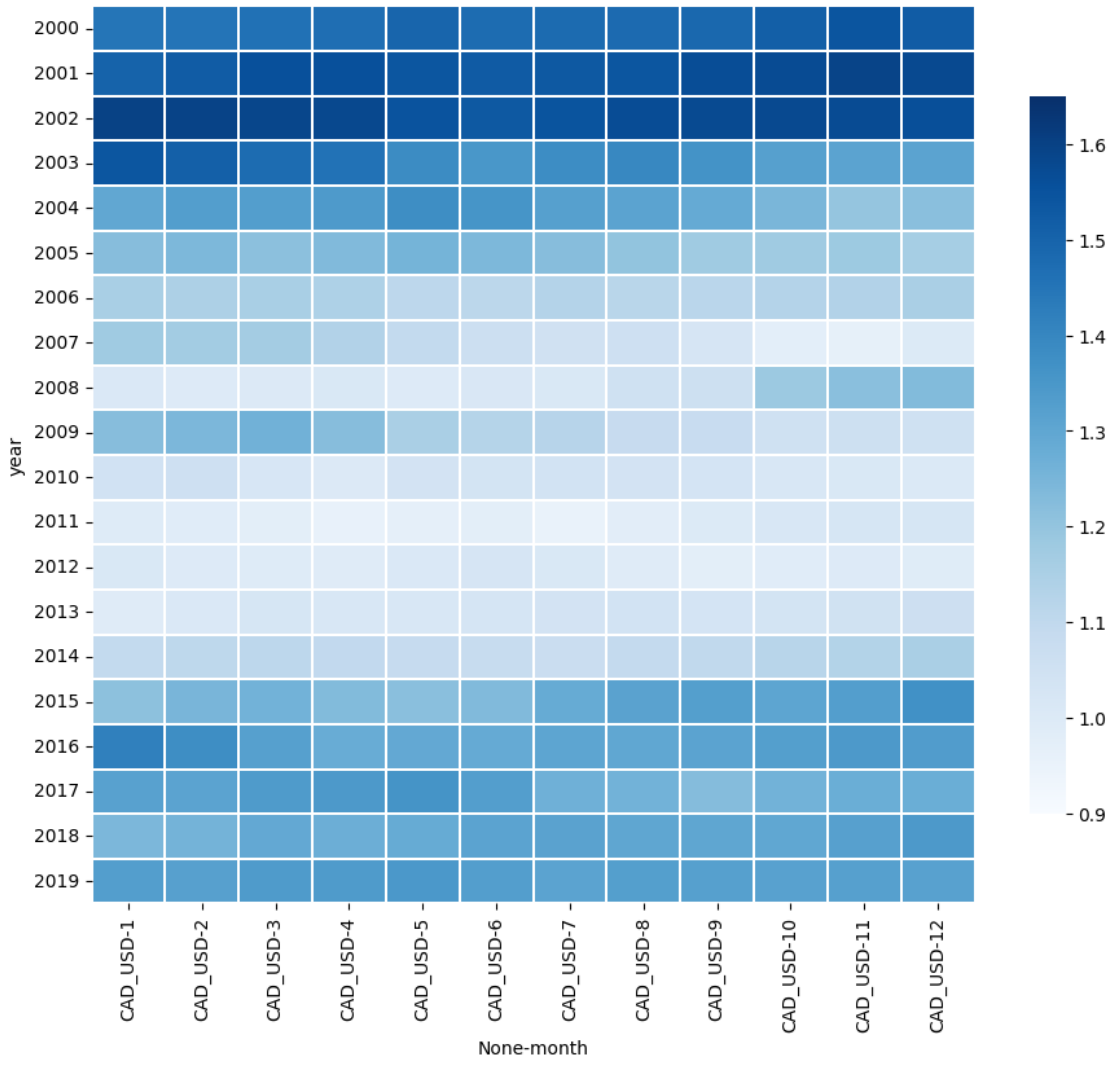
month	8	9	10	11	12
year					
2000	1.482813	1.486430	1.512476	1.542638	1.521875
2001	1.539857	1.567939	1.571677	1.592245	1.578755
2002	1.569418	1.576135	1.578009	1.571453	1.559219
2003	1.396271	1.363371	1.322095	1.313044	1.312755
2004	1.312677	1.288095	1.246935	1.196770	1.218883
2005	1.204283	1.177681	1.177415	1.181545	1.161481
2006	1.118213	1.116120	1.128538	1.135881	1.153235
2007	1.057852	1.026745	0.975413	0.967238	1.002070
2008	1.053457	1.058205	1.184695	1.217094	1.233695
2009	1.087238	1.081638	1.054676	1.059300	1.053691
2010	1.040395	1.032957	1.017900	1.012900	1.008062
2011	0.981709	1.002500	1.019800	1.024755	1.023524
2012	0.992383	0.978300	0.987155	0.996970	0.989820
2013	1.040718	1.034235	1.036282	1.048642	1.063919
2014	1.092633	1.101052	1.121155	1.132539	1.153162
2015	1.314724	1.326581	1.307224	1.327853	1.371255
2016	1.299783	1.310776	1.325095	1.343415	1.333919
2017	1.260770	1.227875	1.260690	1.277335	1.276870
2018	1.304248	1.303400	1.300441	1.320480	1.343611
2019	1.327314	1.324050	1.318923	1.323658	1.316895

[6]:
```
fig, ax = plt.subplots(figsize=(11, 9))
sb.heatmap(df_m)
```

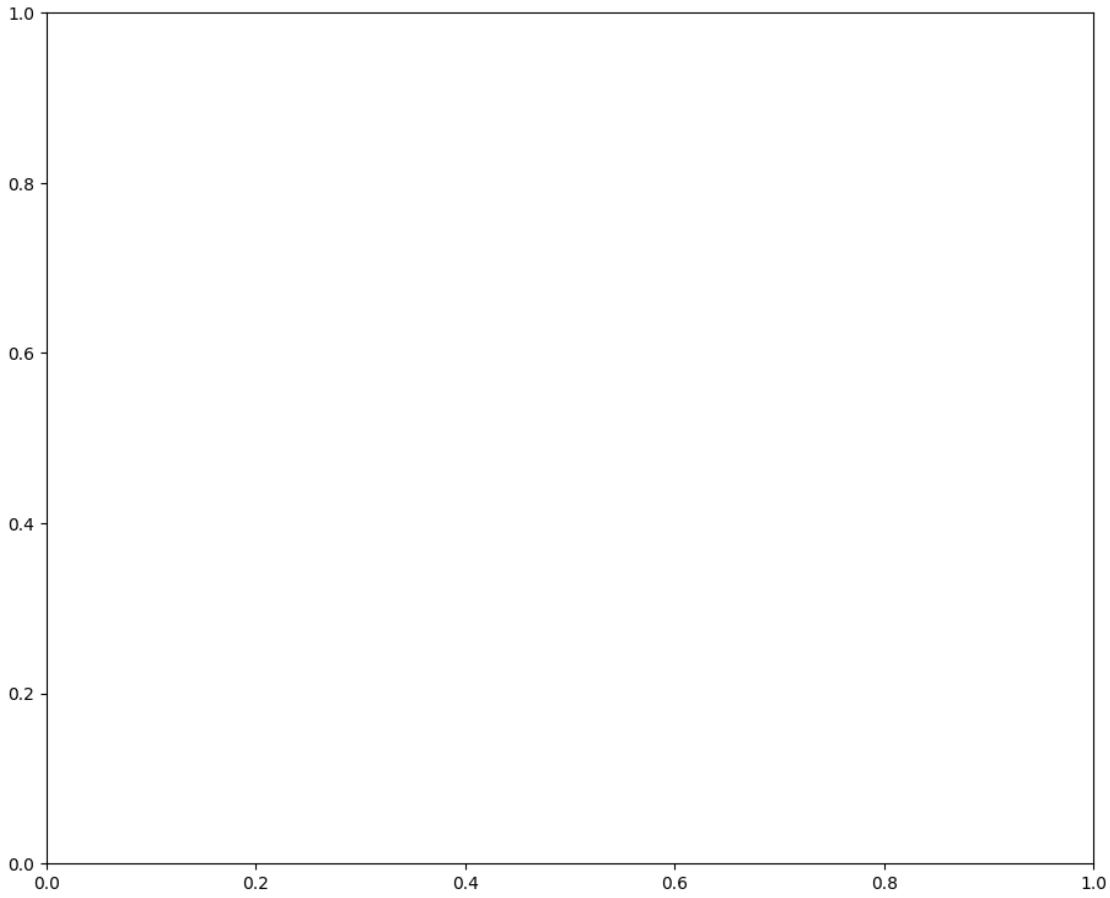
```
plt.show()
```



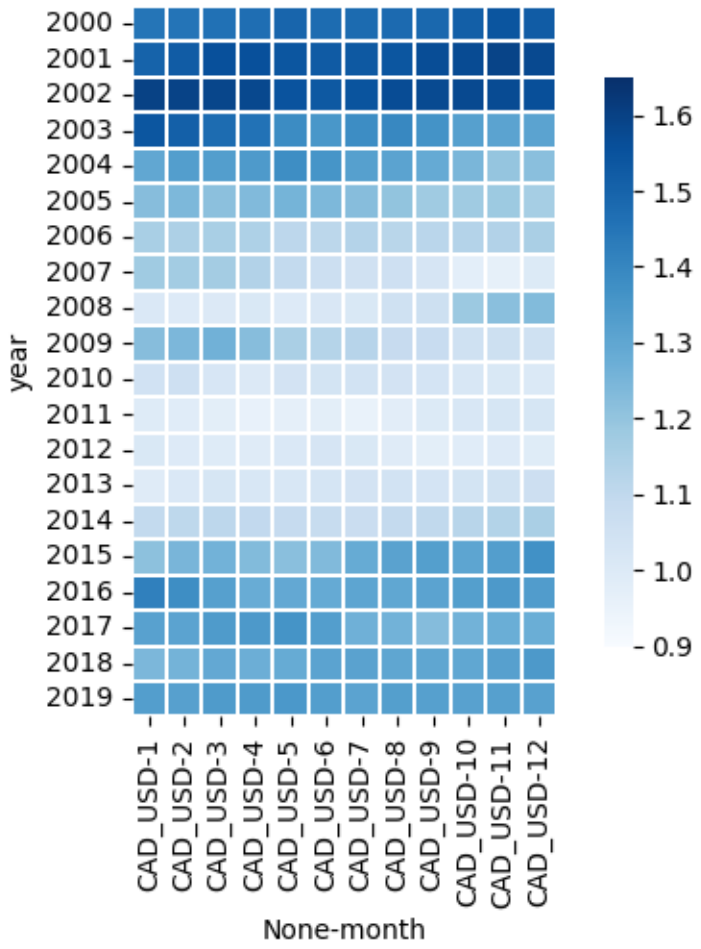
[7]:
```
fig, ax = plt.subplots(figsize=(11, 9))
# plot heatmap
sb.heatmap(df_m, cmap="Blues", vmin= 0.9, vmax=1.65,
           linewidth=0.3, cbar_kws={"shrink": .8})
plt.show()
```



[8]:
```
# figure
fig, ax = plt.subplots(figsize=(11, 9))
plt.show()
```



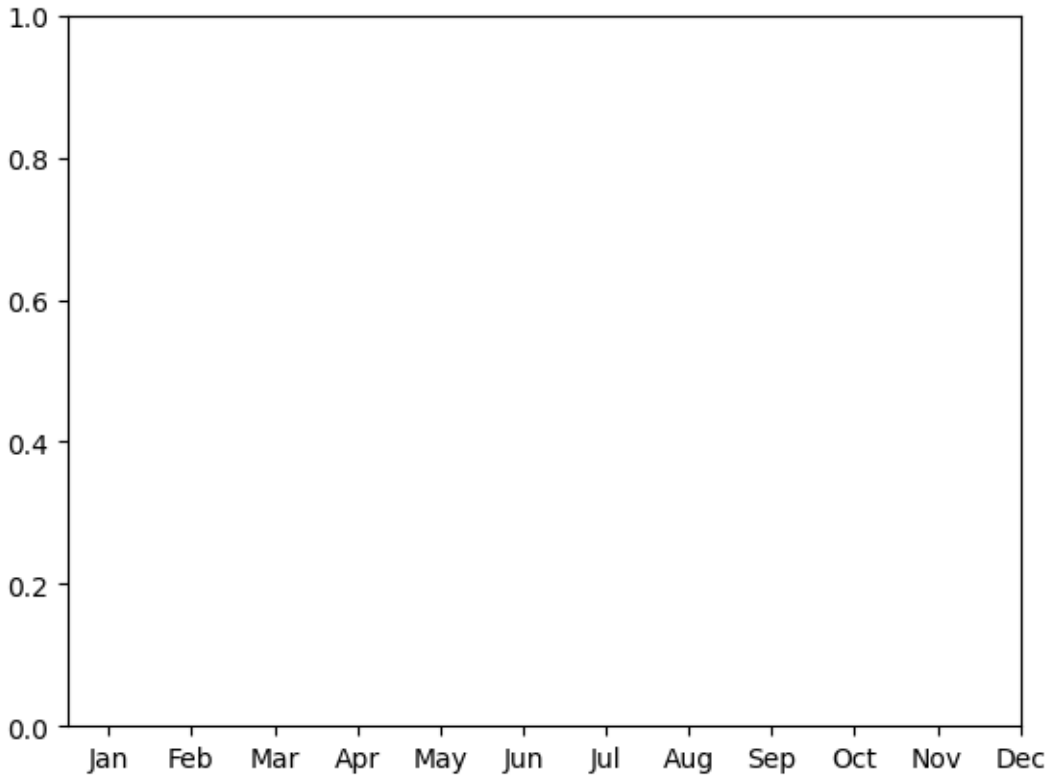
[9]:
```
# plot heatmap
sb.heatmap(df_m, cmap="Blues", vmin= 0.9, vmax=1.65, square=True,
           linewidth=0.3, cbar_kws={"shrink": .8})
plt.show()
```



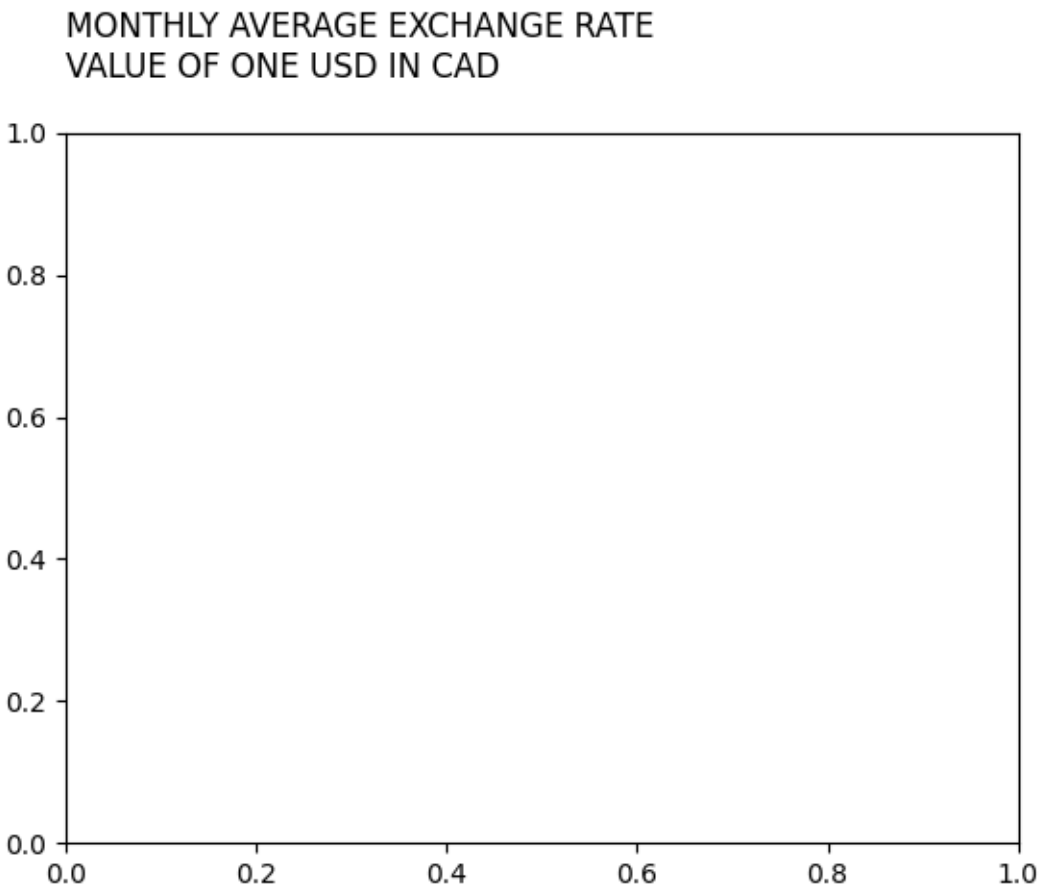
[ ]:

[10]:
```
ax.xaxis.tick_top()
labels = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
                'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
y = np.arange(0.5, 12)
plt.xticks(y, labels)    #optional to set the class names for the bars

plt.show()
```

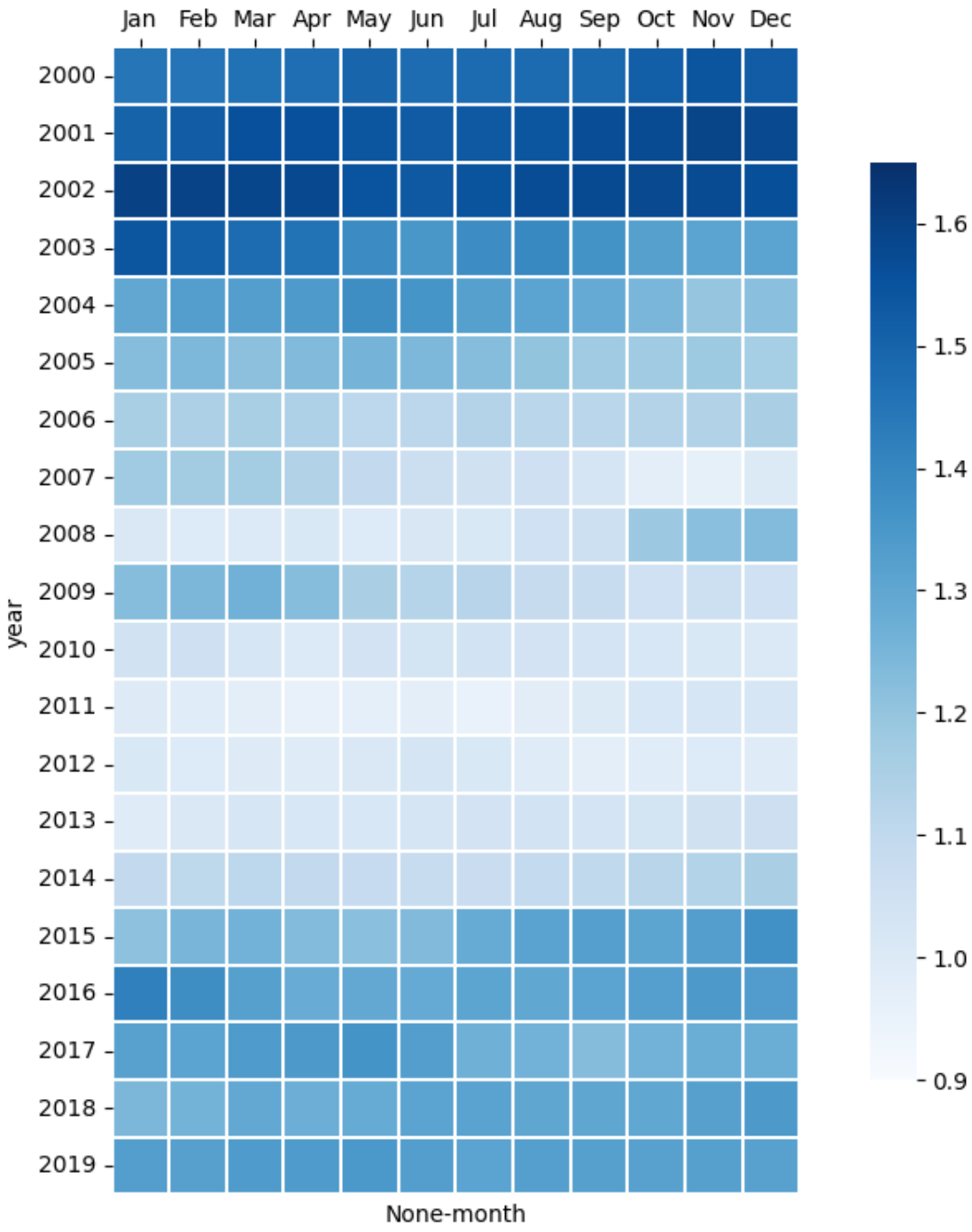


[11]:
```
# axis labels
plt.xlabel('')
plt.ylabel('')
# title
title = 'monthly Average exchange rate\nValue of one USD in CAD\n'.upper()
plt.title(title, loc='left')
plt.show()
```

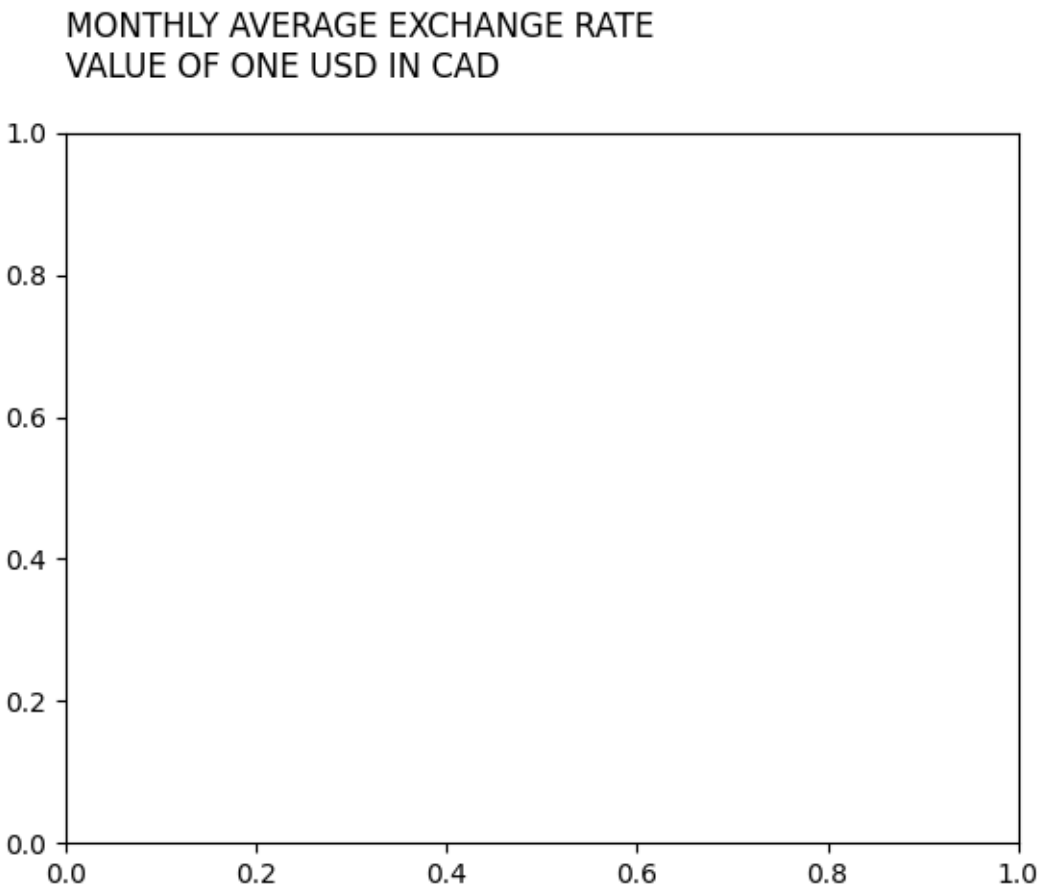


[12]:
```
# figure
fig, ax = plt.subplots(figsize=(11, 9))
# plot heatmap
sb.heatmap(df_m, cmap="Blues", vmin= 0.9, vmax=1.65, square=True,
           linewidth=0.3, cbar_kws={"shrink": .8})
# xticks
ax.xaxis.tick_top()
labels = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
                'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
y = np.arange(0.5, 12)
plt.xticks(y, labels)    #optional to set the class names for the bars

plt.show()
# axis labels
plt.xlabel('')
plt.ylabel('')
# title
title = 'monthly Average exchange rate\nValue of one USD in CAD\n'.upper()
plt.title(title, loc='left')
```



[12]: Text(0.0, 1.0, 'MONTHLY AVERAGE EXCHANGE RATE\nVALUE OF ONE USD IN CAD\n')


MONTHLY AVERAGE EXCHANGE RATE
VALUE OF ONE USD IN CAD
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

[ ]: