Getting Started with Python

Time Series Project

Author :Lawrence Nzomo

Date: 22/3/2022

Resources

1. #### Clean Dataset
2. #### Submission Portal

If you are having problems please refer to this document:

1. #### Time Series Notebook

Instructions

Import all the libraries listed in the first cell. Make sure all modules are installed.

Use the provided data set to answer the following:

1. a) What is the lowest price for Safaricom (*SCOM*) b) What was the date when Safaricom had the lowest price?
2. a) What is the highest price Safaricom stock reached in the data b) What was the date when Safaricom stock recorded the highest price?
3. Create a line plot for Safaricom stock and verify if the information provided above is indeed correct.
4. Select **one** of the sectors provided (agric, comm, bank, const, energy, insur, invest, manu)

   a) Use **pandas** to create a subset containing all the rows of the dataframe and only companies in your selected sector. Rename this dataframe to the **sector_name_df**

   b) Using the subset for the sector, use **matplotlib** subplot to create subplots to fit all the sector stocks in one plot. One row can have a maximum of 3 charts.

   c) Using your sector DataFrame use the `corr()` DataFrame method to come up with a correlogram. Create a DataFrame for these correlations

   d) Use **Seaborn** to plot the **correlation plot** for your sector stocks.

Key performance Metrics:

- Go an extra step to produce charts that are visually appealing
- Ensure all the plots have a Title
- Ensure all plots have x labels and y labels where applicable
- Your plots should be clearly visible. Change the size of your plot to a comfortable width and height.
- Save all your plots

In [12]:
```python
import os
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
```

In [2]:
```python
os.listdir()
```

Out[2]:
```
['.ipynb_checkpoints',
 'APPLIED STOCHASTIC PROCESSES  (1).ppt',
 'APPLIED STOCHASTIC PROCESSES .ppt',
 'BluetoothDriverInstaller_x64.exe',
 'branching process (1).ppt',
 'branching process (2).ppt',
 'branching process.ppt',
 'BT-22.110.2-32-64UWD-Win10-Win11.exe',
 'clean_stock_prices.csv',
 'data-analysis-with-python-pandas.html',
 'data-analysis-with-python-pandas_files',
 'denaco-manu-python-curriculum - Sheet1.pdf',
 'desktop.ini',
 'loops and iterations in python.html',
 'loops and iterations in python_files',
 'mysql-installer-web-community-8.0.28.0.msi',
 'New folder',
 'new_daily_prices.csv',
 'PDFConverterSetup_x64.zip',
 'POISSON PROCESS (1).ppt',
 'POISSON PROCESS (2).ppt',
 'POISSON PROCESS.ppt',
 'project-time-series-workbook.ipynb',
 'project-time-series-workbook.ipynb - Shortcut.lnk',
 'project-time-series.ipynb',
 'python-3.9.10-amd64.exe',
 'Python_Lecture3.pptx',
 'R-4.1.3-win.exe',
 'RStudio-2022.02.0-443.exe',
 'Semester_20212022_I63_4_2__I6337902018.pdf',
 'Student-Internship.pdf1_.zip',
 'student_copy_pandas_workbook.ipynb',
 'student_workbook_stocks.ipynb',
 'subaru.csv',
 'telegram-for-desktop-3-6-1.exe',
 'top-10-brands.png',
 'vehicle_data (1).csv',
 'vehicle_data.csv',
 'vehicle_data.xls',
 'vehicle_dataset_project (1).ipynb',
 'vehicle_dataset_project.ipynb',
 'vehicle_dataset_project1',
 'vehicle_dataset_project1.ipynb',
 'vehicle_dataset_project1.zip',
 'WhatsAppSetup.exe']
```

In [16]:
```python
# read in the necessary file ('clean_stock_prices.csv')
df = pd.read_csv('clean_stock_prices.csv')
df.head()
```

Out[16]:

	Date	EGAD	KUKZ	LIMT	SASN	WTK	CGEN	ABSA	BKG	DTK	...	BAT	CARB	EABL	EVRD	FTGH	ORCH	MSC	UNGA	SCOM
0	2022-01-13	12.90	385.0	320.0	22.20	130.00	54.00	11.80	30.00	59.00	...	440.0	10.80	151.50	0.96	1.34	10.4	0.27	27.10	37.95
1	2022-01-11	13.80	385.0	320.0	20.55	134.75	44.75	11.90	30.75	59.50	...	445.0	10.85	161.00	0.88	1.31	10.4	0.27	27.65	38.45
2	2022-01-07	13.80	420.0	320.0	21.25	132.00	37.05	11.80	29.05	60.00	...	442.0	10.90	164.75	0.94	1.30	10.4	0.27	27.65	39.90
3	2022-01-06	13.80	420.0	320.0	20.25	130.75	33.70	11.80	29.30	60.00	...	442.0	10.90	160.75	0.99	1.29	10.4	0.27	27.65	40.00
4	2022-01-05	12.85	420.0	320.0	19.95	130.75	30.60	11.75	29.50	59.75	...	442.0	10.90	163.75	0.99	1.26	10.4	0.27	27.65	39.55

5 rows × 61 columns

In [17]:
```python
df.tail()
```

Out[17]:

	Date	EGAD	KUKZ	LIMT	SASN	WTK	CGEN	ABSA	BKG	DTK	...	BAT	CARB	EABL	EVRD	FTGH	ORCH	MSC	UNGA	SCOM
97	2021-08-09	12.15	415.0	300.00	19.50	134.5	35.0	9.80	32.40	65.75	...	445.5	12.25	179.25	0.96	1.32	10.4	0.27	31.0	42.9
98	2021-08-06	12.15	415.0	300.00	20.00	134.5	35.0	9.80	32.40	65.75	...	454.0	12.25	179.00	0.98	1.32	10.4	0.27	31.0	42.1
99	2021-08-05	12.30	415.0	320.00	20.00	134.5	35.0	9.82	31.85	65.00	...	450.0	12.20	178.50	0.98	1.31	10.4	0.27	31.0	41.2
100	2021-08-04	12.00	415.0	320.00	19.95	135.0	35.0	9.76	29.75	64.00	...	455.0	12.00	179.75	0.98	1.30	10.4	0.27	31.0	41.1
101	2021-08-03	11.80	415.0	304.75	19.95	134.5	35.0	9.82	29.50	65.00	...	450.0	12.00	180.00	0.98	1.31	10.4	0.27	28.3	41.0

5 rows × 61 columns

In [18]:
```python
# a) What is the lowest price for Safaricom (SCOM)
# b) What was the date when Safaricom had the lowest price?
df.nsmallest(1,'SCOM')[['Date','SCOM']]
```

Out[18]:

	Date	SCOM
22	2021-12-07	36.5

In [20]:
```python
# a) What is the highest price Safaricom stock reached in the data
# b) What was the date when Safaricom stock recordedthe highest price?
df.nlargest(1,'SCOM')[['Date','SCOM']]
```

Out[20]:

	Date	SCOM
87	2021-08-24	44.95

In [8]:
```python
df.SCOM
```

Out[8]:
```
Date
2022-01-13    37.95
2022-01-11    38.45
2022-01-07    39.90
2022-01-06    40.00
2022-01-05    39.55
              ...
2021-08-09    42.95
2021-08-06    42.10
2021-08-05    41.25
2021-08-04    41.10
2021-08-03    41.05
Name: SCOM, Length: 102, dtype: float64
```

In [22]:
```python
DATA=df[['Date','SCOM']]
DATA.head()
```

Out[22]:

	Date	SCOM
0	2022-01-13	37.95
1	2022-01-11	38.45
2	2022-01-07	39.90
3	2022-01-06	40.00
4	2022-01-05	39.55

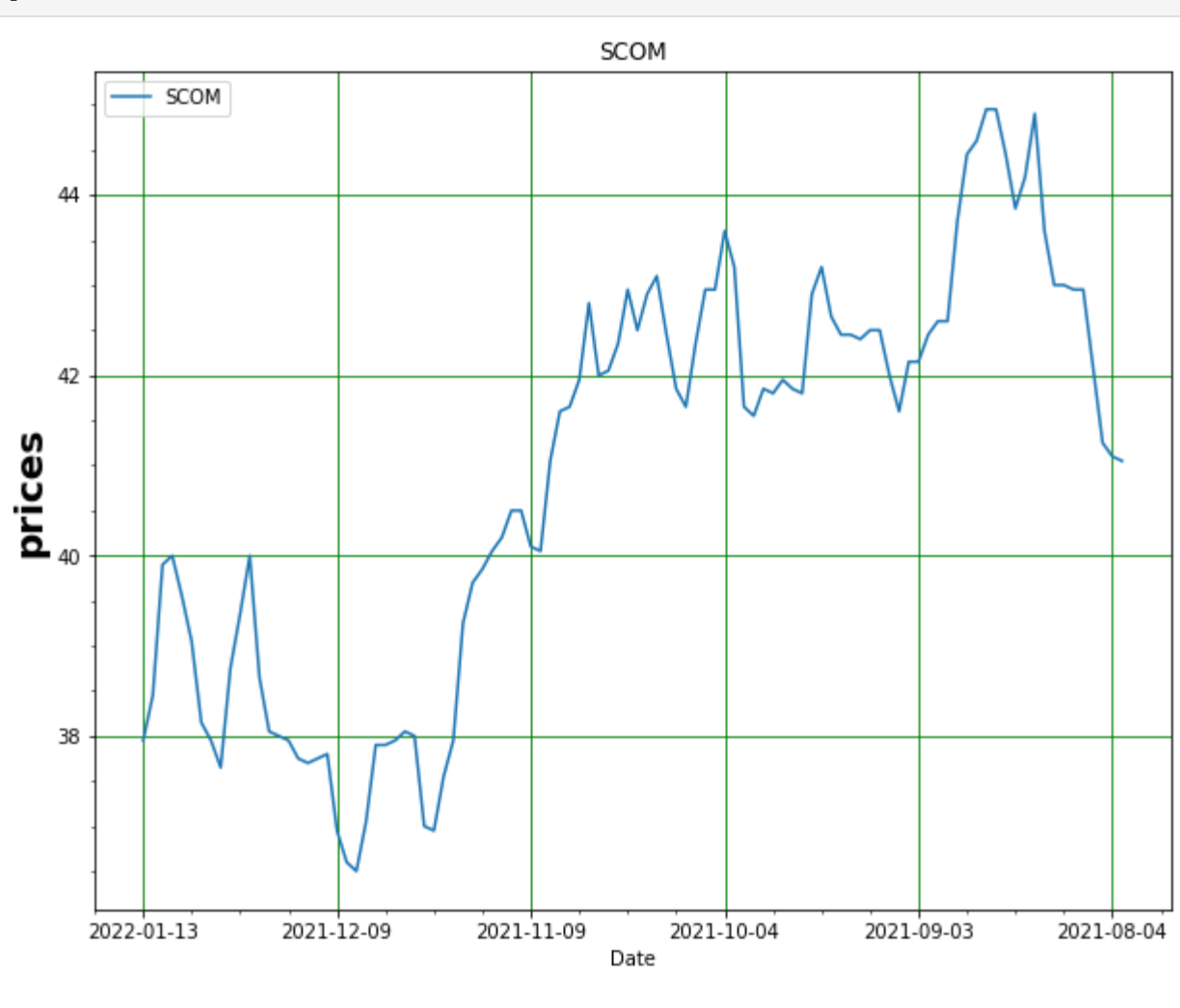
In [23]:
```python
DATA.tail()
```

Out[23]:

	Date	SCOM
97	2021-08-09	42.95
98	2021-08-06	42.10
99	2021-08-05	41.25
100	2021-08-04	41.10
101	2021-08-03	41.05

In [24]:
```python
DATA.plot('Date','SCOM')

plt.title('SCOM')
plt.ylabel('prices',fontsize=20, fontweight='bold')
plt.grid(color='g')
plt.minorticks_on()
fig = plt.gcf()
fig.set_size_inches(10,8)
fig.savefig('SCOM stocks')
# fig = plt.gcf()
# fig.savefig('SCOM ')
plt.show()
```



Use this part to answer question 4

In [29]:
```python
# agriculatural companies
agric = ['EGAD','KUKZ','LIMT','SASN','WTK']

# commercial companies
comm = ['XPRS','KQ','LKL','NBV', 'NMG','SMER','SCAN','SGL','TPSE','UCHM']

# banking companies
bank = ['ABSA','BKG','DTK','EQTY','HFCK','IMH', 'KCB','NBK', 'NCBA','SBIC','SCBK','COOP']

# construction sector
const = ['ARM','BAMB','CRWN','CABL','PORT']

# energy sector
energy = ['KEGN','KPLC','TOTL','UMME']

# insurance sector
insur = ['BRIT','CIC','JUB','KNRE','LBTY', 'SLAM']

# investement sector
invest = ['CTUM','HAFR','KURV','OCH','TCL','NSE']


# manufacturing sector
manu = ['BOC','BAT','CARB','EABL','EVRD','FTGH','ORCH', 'MSC','UNGA']
```

In [10]:
```python
df.index
```

Out[10]:
```
Index(['2022-01-13', '2022-01-11', '2022-01-07', '2022-01-06', '2022-01-05',
       '2022-01-04', '2022-01-03', '2021-12-31', '2021-12-30', '2021-12-29',
       ...
       '2021-08-17', '2021-08-16', '2021-08-13', '2021-08-12', '2021-08-11',
       '2021-08-09', '2021-08-06', '2021-08-05', '2021-08-04', '2021-08-03'],
      dtype='object', name='Date', length=102)
```

In [25]:
```python
df2 = df.loc['2022-08-03':]
df2
```

Out[25]:

	Date	EGAD	KUKZ	LIMT	SASN	WTK	CGEN	ABSA	BKG	DTK	...	BAT	CARB	EABL	EVRD	FTGH	ORCH	MSC	UNGA	SCO
21	2021-12-08	13.80	423.5	300.00	18.40	126.00	35.00	10.95	26.35	56.50	...	421.5	10.80	150.50	1.02	1.35	10.4	0.27	27.65	36.
22	2021-12-07	13.80	423.5	300.00	18.40	126.00	35.00	10.85	24.30	57.00	...	421.5	10.80	150.00	1.02	1.35	10.4	0.27	27.65	36.
23	2021-12-03	13.95	423.5	300.00	19.00	129.25	35.00	11.05	27.00	55.25	...	427.0	11.05	150.00	0.96	1.30	10.4	0.27	27.55	37.
24	2021-12-02	13.95	423.5	300.00	19.00	129.25	31.95	10.80	27.00	55.50	...	428.5	11.15	149.75	0.94	1.28	10.4	0.27	27.55	37.
25	2021-12-01	12.80	423.5	300.00	19.00	129.00	35.50	10.90	26.60	54.75	...	430.0	11.00	152.00	0.92	1.28	10.4	0.27	27.55	37.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
97	2021-08-09	12.15	415.0	300.00	19.50	134.50	35.00	9.80	32.40	65.75	...	445.5	12.25	179.25	0.96	1.32	10.4	0.27	31.00	42.
98	2021-08-06	12.15	415.0	300.00	20.00	134.50	35.00	9.80	32.40	65.75	...	454.0	12.25	179.00	0.98	1.32	10.4	0.27	31.00	42.
99	2021-08-05	12.30	415.0	320.00	20.00	134.50	35.00	9.82	31.85	65.00	...	450.0	12.20	178.50	0.98	1.31	10.4	0.27	31.00	41.
100	2021-08-04	12.00	415.0	320.00	19.95	135.00	35.00	9.76	29.75	64.00	...	455.0	12.00	179.75	0.98	1.30	10.4	0.27	31.00	41.
101	2021-08-03	11.80	415.0	304.75	19.95	134.50	35.00	9.82	29.50	65.00	...	450.0	12.00	180.00	0.98	1.31	10.4	0.27	28.30	41.

81 rows × 61 columns

In [26]:
```python
df2.columns
```

Out[26]:
```
Index(['Date', 'EGAD', 'KUKZ', 'LIMT', 'SASN', 'WTK', 'CGEN', 'ABSA', 'BKG',
       'DTK', 'EQTY', 'HFCK', 'IMH', 'KCB', 'NBK', 'NCBA', 'SBIC', 'SCBK',
       'COOP', 'XPRS', 'KQ', 'LKL', 'NBV', 'NMG', 'SMER', 'SCAN', 'SGL',
       'TPSE', 'UCHM', 'ARM', 'BAMB', 'CRWN', 'CABL', 'PORT', 'KEGN', 'KPLC',
       'TOTL', 'UMME', 'BRIT', 'CIC', 'JUB', 'KNRE', 'LBTY', 'SLAM', 'CTUM',
       'HAFR', 'KURV', 'OCH', 'TCL', 'NSE', 'BOC', 'BAT', 'CARB', 'EABL',
       'EVRD', 'FTGH', 'ORCH', 'MSC', 'UNGA', 'SCOM', 'FAHR'],
      dtype='object')
```

In [27]:
```python
df2.loc[:,'ABSA':'COOP']
```

Out[27]:

	ABSA	BKG	DTK	EQTY	HFCK	IMH	KCB	NBK	NCBA	SBIC	SCBK	COOP
21	10.95	26.35	56.50	46.95	3.99	21.00	42.75	4.12	23.50	87.25	127.50	11.95
22	10.85	24.30	57.00	45.15	3.99	21.00	41.00	4.12	23.15	87.00	128.75	11.80
23	11.05	27.00	55.25	46.60	4.10	20.60	41.15	4.12	24.00	90.50	129.00	11.80
24	10.80	27.00	55.50	47.35	4.17	20.75	41.10	4.12	24.25	90.75	129.00	11.80
25	10.90	26.60	54.75	48.25	4.20	20.90	42.25	4.12	24.60	91.75	129.00	12.05
...	...	...	...	...	...	...	...	...	...	...	...	...
97	9.80	32.40	65.75	50.25	3.70	22.50	46.60	4.12	25.85	86.50	131.50	13.65
98	9.80	32.40	65.75	50.00	3.75	22.55	46.75	4.12	26.20	92.50	131.00	13.65
99	9.82	31.85	65.00	49.40	3.71	22.30	46.45	4.12	26.20	90.00	130.00	13.55
100	9.76	29.75	64.00	49.10	3.66	22.70	46.20	4.12	26.25	90.00	132.00	13.45
101	9.82	29.50	65.00	49.00	3.74	22.75	45.95	4.12	26.65	88.50	130.00	13.55

81 rows × 12 columns

In [28]:
```python
bank_df = df2.loc[:,'ABSA':'COOP']
bank_df.head()
```

Out[28]:

	ABSA	BKG	DTK	EQTY	HFCK	IMH	KCB	NBK	NCBA	SBIC	SCBK	COOP
21	10.95	26.35	56.50	46.95	3.99	21.00	42.75	4.12	23.50	87.25	127.50	11.95
22	10.85	24.30	57.00	45.15	3.99	21.00	41.00	4.12	23.15	87.00	128.75	11.80
23	11.05	27.00	55.25	46.60	4.10	20.60	41.15	4.12	24.00	90.50	129.00	11.80
24	10.80	27.00	55.50	47.35	4.17	20.75	41.10	4.12	24.25	90.75	129.00	11.80
25	10.90	26.60	54.75	48.25	4.20	20.90	42.25	4.12	24.60	91.75	129.00	12.05

In [29]:
```python
bank_df.columns
```

Out[29]:
```
Index(['ABSA', 'BKG', 'DTK', 'EQTY', 'HFCK', 'IMH', 'KCB', 'NBK', 'NCBA',
       'SBIC', 'SCBK', 'COOP'],
      dtype='object')
```

In [30]:
```python
bank_cols = bank_df.columns
for bank in bank_cols:
    print(bank)
```

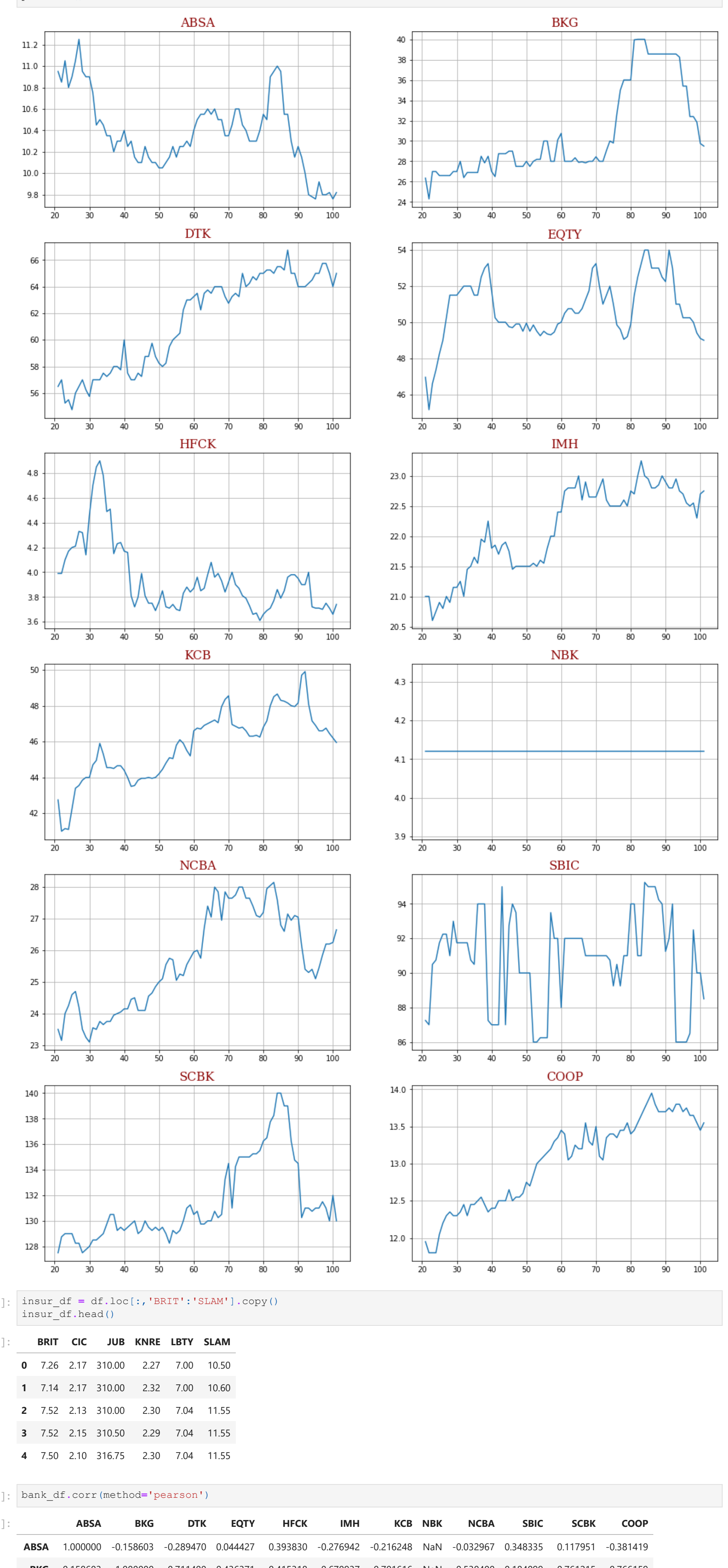
```
ABSA
BKG
DTK
EQTY
HFCK
IMH
KCB
NBK
NCBA
SBIC
SCBK
COOP
```

In [31]:
```python
bank_cols = bank_df.columns

font = {'family': 'serif',
        'color':  'darkred',
        'weight': 'normal',
        'size': 16,
        }

for idx,bank in enumerate(bank_cols,start=1):
    plt.subplot(6,2,idx)
    plt.title(bank,fontdict=font)
    plt.grid()
    plt.plot(bank,data=df2)

fig = plt.gcf()
fig.set_size_inches(16,30)
plt.show()
```



In [32]:
```python
insur_df = df.loc[:,'BRIT':'SLAM'].copy()
insur_df.head()
```

Out[32]:

	BRIT	CIC	JUB	KNRE	LBTY	SLAM
0	7.26	2.17	310.00	2.27	7.00	10.50
1	7.14	2.17	310.00	2.32	7.00	10.60
2	7.52	2.13	310.00	2.30	7.04	11.55
3	7.52	2.15	310.50	2.29	7.04	11.55
4	7.50	2.10	316.75	2.30	7.04	11.55

In [33]:
```python
bank_df.corr(method='pearson')
```

Out[33]:

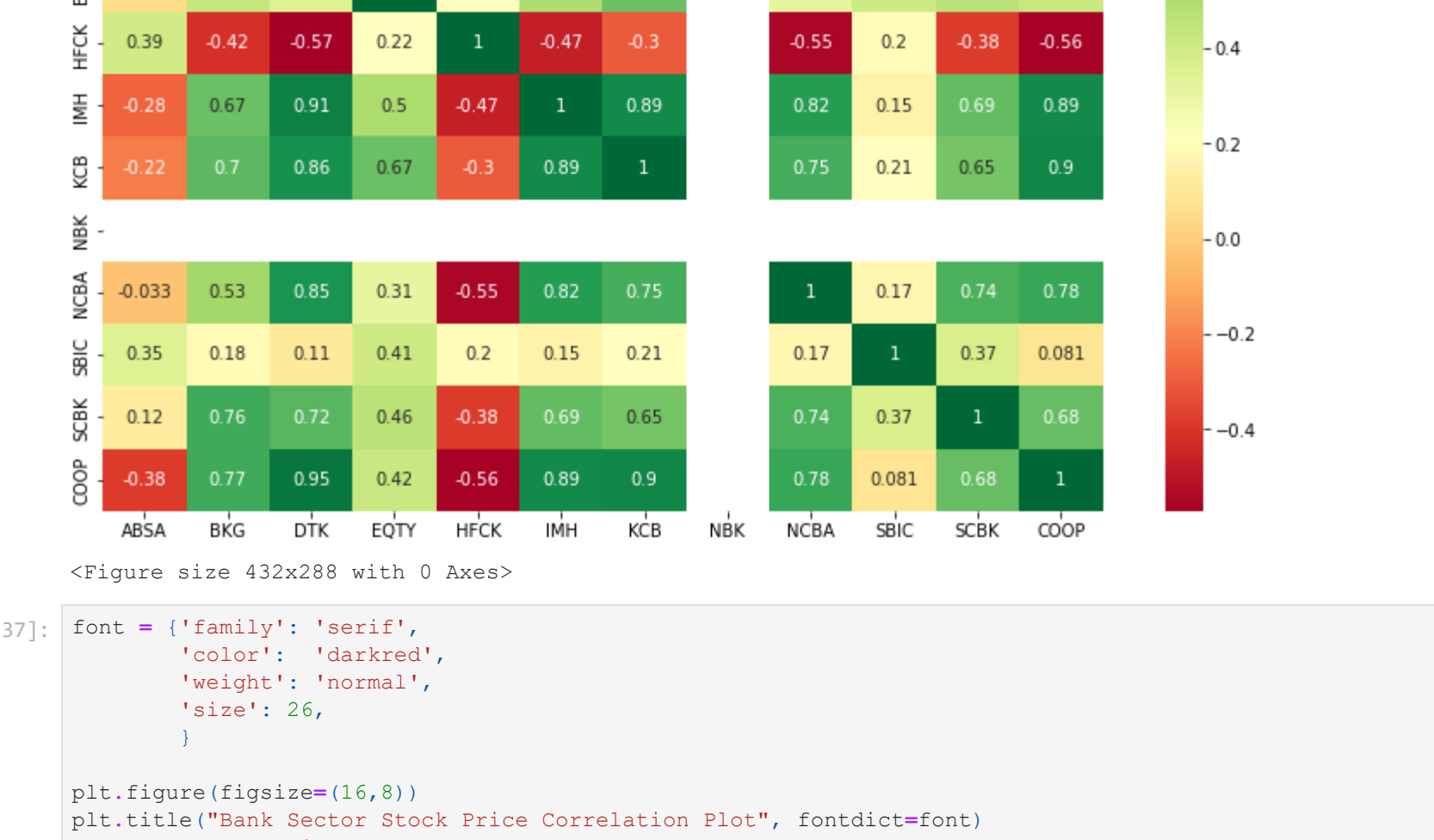
	ABSA	BKG	DTK	EQTY	HFCK	IMH	KCB	NBK	NCBA	SBIC	SCBK	COOP
ABSA	1.000000	-0.158603	-0.289470	0.044427	0.393830	-0.276942	-0.216248	NaN	-0.032967	0.348335	0.117951	-0.381419
BKG	-0.158603	1.000000	0.711400	0.436371	-0.415318	0.670937	0.701616	NaN	0.530400	0.184099	0.761215	0.766159
DTK	-0.289470	0.711400	1.000000	0.344141	-0.571338	0.914820	0.861130	NaN	0.847686	0.106406	0.716745	0.950264
EQTY	0.044427	0.436371	0.344141	1.000000	0.219571	0.504339	0.674137	NaN	0.308285	0.406240	0.462389	0.422881
HFCK	0.393830	-0.415318	-0.571338	0.219571	1.000000	-0.470142	-0.301448	NaN	-0.550816	0.195543	-0.384900	-0.559312
IMH	-0.276942	0.670937	0.914820	0.504339	-0.470142	1.000000	0.887178	NaN	0.819075	0.152025	0.689314	0.888288
KCB	-0.216248	0.701616	0.861130	0.674137	-0.301448	0.887178	1.000000	NaN	0.754208	0.210085	0.647696	0.901175
NBK	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NCBA	-0.032967	0.530400	0.847686	0.308285	-0.550816	0.819075	0.754208	NaN	1.000000	0.168577	0.741653	0.782317
SBIC	0.348335	0.184099	0.106406	0.406240	0.195543	0.152025	0.210085	NaN	0.168577	1.000000	0.373323	0.081002
SCBK	0.117951	0.761215	0.716745	0.462389	-0.384900	0.689314	0.647696	NaN	0.741653	0.373323	1.000000	0.683280
COOP	-0.381419	0.766159	0.950264	0.422881	-0.559312	0.888288	0.901175	NaN	0.782317	0.081002	0.683280	1.000000

In [34]:
```python
corr_df = bank_df.corr(method="pearson")
```

In [35]:
```python
import seaborn as sns
```

In [36]:
```python
plt.figure(figsize=(13, 8))
sns.heatmap(corr_df, annot=True, cmap='RdYlGn')
plt.figure()
```

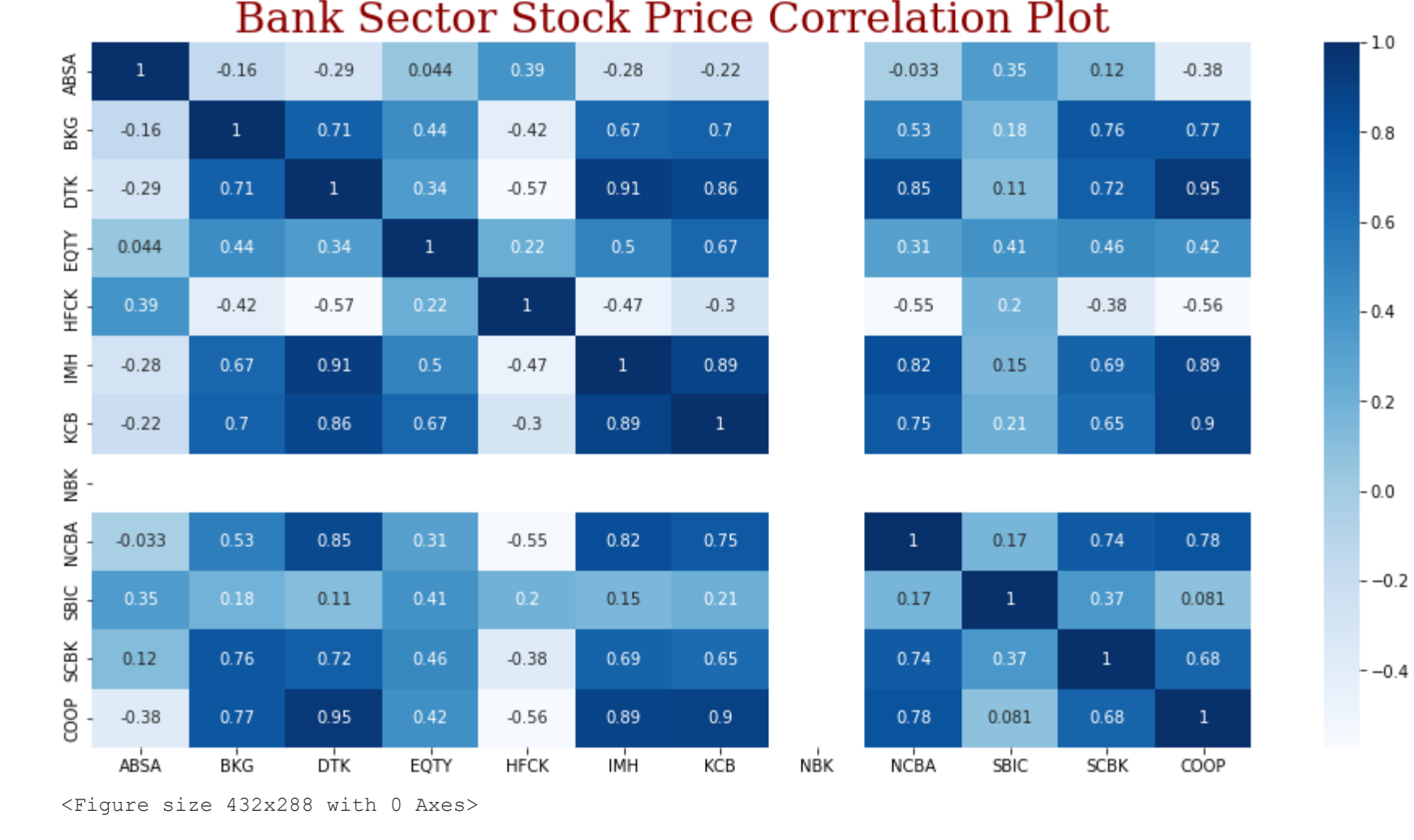
Out[36]: `<Figure size 432x288 with 0 Axes>`



`<Figure size 432x288 with 0 Axes>`

In [37]:
```python
font = {'family': 'serif',
        'color':  'darkred',
        'weight': 'normal',
        'size': 26,
        }

plt.figure(figsize=(16,8))
plt.title("Bank Sector Stock Price Correlation Plot", fontdict=font)
cmap = ["mako","PiYG","YlGnBu","Blues"]
sns.heatmap(corr_df, annot=True, cmap=cmap[np.random.randint(len(cmap))])
plt.figure()
plt.show()
```



`<Figure size 432x288 with 0 Axes>`