Getting Started with Python

Time Series Project

Author : Otieno Cindy Gloria Aluoch

Date: 27/3/2022

Resources

1. **Clean Dataset (some-link)**
2. **Submission Portal (https://dasclab.uonbi.ac.ke/analytics/projects)**

If you are having problems please refer to this document:

3. **Time Series Notebook (https://dasclab.uonbi.ac.ke/dstraining/time-series-data.html)**

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 [2]:
```
import os
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
```

Ensure that you have the ***clean_stock_prices.csv*** file in your working directory

If you can see the *clean_stock_prices.csv* as an output in the above cell, read the data into a DataFrame using pandas

In [1]:
```
# read in the necessary file ('clean_stock_prices.csv')
df = pd.read_csv('clean_stock_prices.csv',index_col=0)
df.head()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [1], in <cell line: 2>()
      1 # read in the necessary file ('clean_stock_prices.csv')
----> 2 df = pd.read_csv('clean_stock_prices.csv',index_col=0)
      3 df.head()

NameError: name 'pd' is not defined
```

In [8]:
```
df.tail()
```

Out[8]:

Date	EGAD	KUKZ	LIMT	SASN	WTK	CGEN	ABSA	BKG	DTK	EQTY	...	BAT	CAI
2021-08-09	12.15	415.0	300.00	19.50	134.5	35.0	9.80	32.40	65.75	50.25	...	445.5	12.
2021-08-06	12.15	415.0	300.00	20.00	134.5	35.0	9.80	32.40	65.75	50.00	...	454.0	12.
2021-08-05	12.30	415.0	320.00	20.00	134.5	35.0	9.82	31.85	65.00	49.40	...	450.0	12.
2021-08-04	12.00	415.0	320.00	19.95	135.0	35.0	9.76	29.75	64.00	49.10	...	455.0	12.
2021-08-03	11.80	415.0	304.75	19.95	134.5	35.0	9.82	29.50	65.00	49.00	...	450.0	12.

5 rows × 60 columns

Use this part to answer questions 1, 2 and 3

In [19]:
```
# lowest price for Safaricom
df['SCOM'].nsmallest(1,)
```

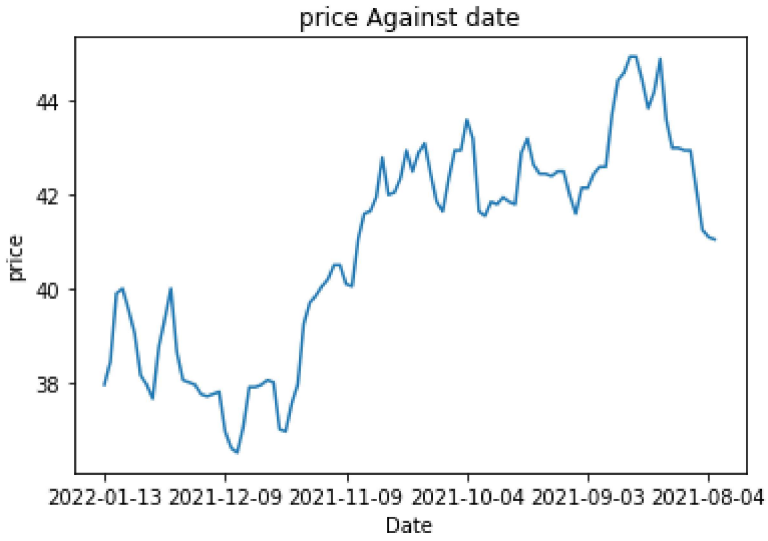
Out[19]:
```
Date
2021-12-07    36.5
Name: SCOM, dtype: float64
```

In [20]:
```
# highest price for Safaricom
df['SCOM'].nlargest(1,)
```

Out[20]:
```
Date
2021-08-24    44.95
Name: SCOM, dtype: float64
```

In [22]:
```
# Plot SCOM to confirm above observations
df['SCOM'].plot()
plt.ylabel("price")
plt.title("price Against date")
```

Out[22]:
```
Text(0.5, 1.0, 'price Against date')
```



Use this part to answer question 4

In [23]:
```
# 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','S

# 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']
```

To subset a sector simply use the **slice** notation. For example if I choose the Insurance sector, i will use the **insur** list

In [25]:
```
manu_df = df.loc[:,'BOC':'UNGA'].copy()
manu_df.head()
```

Out[25]:

Date	BOC	BAT	CARB	EABL	EVRD	FTGH	ORCH	MSC	UNGA
2022-01-13	72.5	440.0	10.80	151.50	0.96	1.34	10.4	0.27	27.10
2022-01-11	73.0	445.0	10.85	161.00	0.88	1.31	10.4	0.27	27.65
2022-01-07	73.0	442.0	10.90	164.75	0.94	1.30	10.4	0.27	27.65
2022-01-06	72.0	442.0	10.90	160.75	0.99	1.29	10.4	0.27	27.65
2022-01-05	70.0	442.0	10.90	163.75	0.99	1.26	10.4	0.27	27.65

In [26]:
```
sector_name_df = manu_df.copy()
sector_name_df.head()
```

Out[26]:

Date	BOC	BAT	CARB	EABL	EVRD	FTGH	ORCH	MSC	UNGA
2022-01-13	72.5	440.0	10.80	151.50	0.96	1.34	10.4	0.27	27.10
2022-01-11	73.0	445.0	10.85	161.00	0.88	1.31	10.4	0.27	27.65
2022-01-07	73.0	442.0	10.90	164.75	0.94	1.30	10.4	0.27	27.65
2022-01-06	72.0	442.0	10.90	160.75	0.99	1.29	10.4	0.27	27.65
2022-01-05	70.0	442.0	10.90	163.75	0.99	1.26	10.4	0.27	27.65

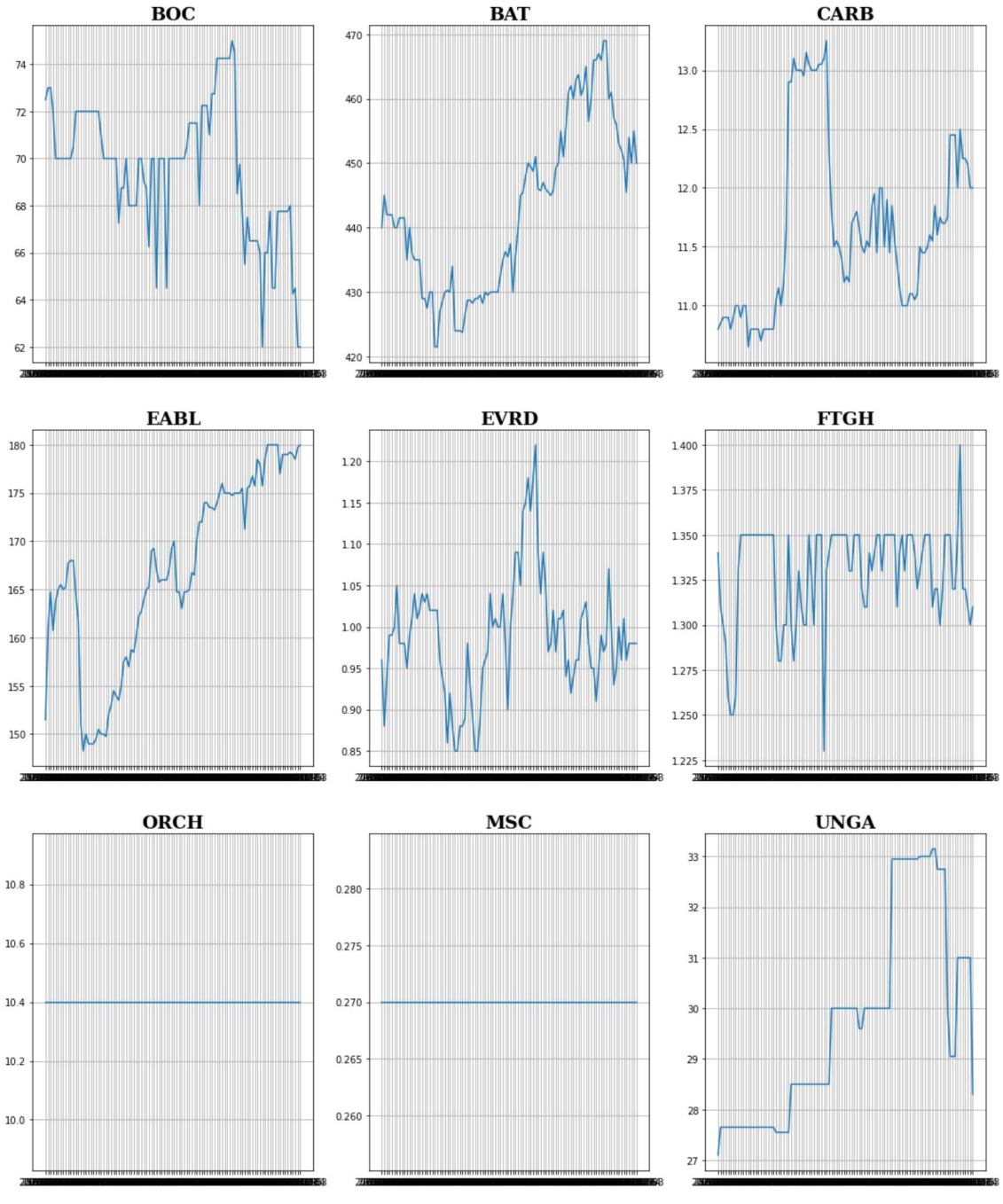
In [28]:

```
manu_cols = sector_name_df.columns

font = {'family': 'serif',
        'color':  'black',
        'weight': 'heavy',
        'size': 19,
        }

for idx,manu in enumerate(manu_cols,start=1):
    plt.subplot(4,3,idx)
    plt.title(manu,fontdict=font)
    plt.grid()
    plt.plot(manu,data=sector_name_df)

fig = plt.gcf()
fig.set_size_inches(18,30)
plt.show()
```


BOC
BAT
CARB
EABL
EVRD
FTGH
ORCH
MSC
UNGA

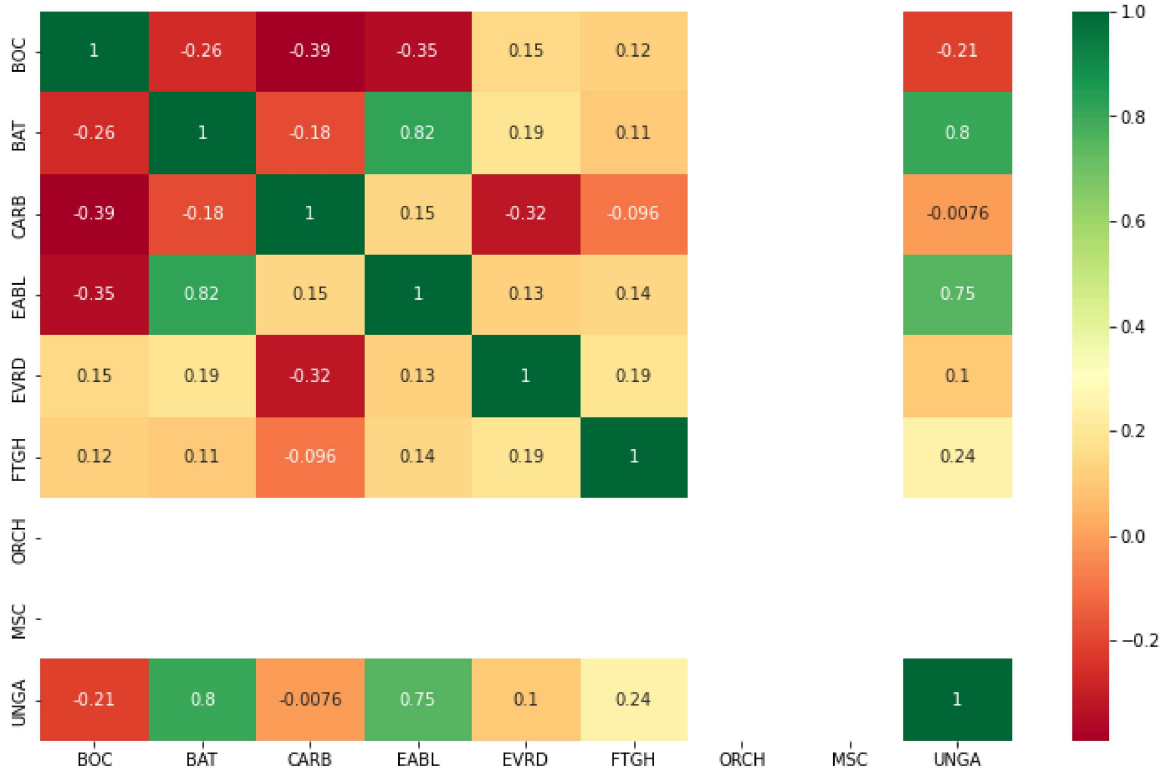
In [29]:
```
sector_name_df.corr(method='pearson')
```

Out[29]:

	BOC	BAT	CARB	EABL	EVRD	FTGH	ORCH	MSC	UNG
BOC	1.000000	-0.264702	-0.390657	-0.351283	0.147241	0.124104	NaN	NaN	-0.21394
BAT	-0.264702	1.000000	-0.184494	0.816630	0.187274	0.109096	NaN	NaN	0.80258
CARB	-0.390657	-0.184494	1.000000	0.146387	-0.317406	-0.095847	NaN	NaN	-0.00765
EABL	-0.351283	0.816630	0.146387	1.000000	0.128904	0.141361	NaN	NaN	0.75381
EVRD	0.147241	0.187274	-0.317406	0.128904	1.000000	0.187224	NaN	NaN	0.10124
FTGH	0.124104	0.109096	-0.095847	0.141361	0.187224	1.000000	NaN	NaN	0.24374
ORCH	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
MSC	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
UNGA	-0.213949	0.802586	-0.007650	0.753815	0.101246	0.243746	NaN	NaN	1.00000

In [30]:
```
plt.figure(figsize=(13, 8))
sns.heatmap(sector_name_df.corr(method='pearson'), annot=True, cmap='RdYlGn')
plt.figure()
```

Out[30]: <Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>