

Getting Started with Python

Time Series Project

Author : Meshack Muli

Date: 26/3/2022

Resources

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

If you are having problems please refer to this document:

3. [Time Series Notebook \(https://dasclab.uonbi.ac.ke/dstraining/time-series-data.html\)](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 [1]:  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

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

```
Out[2]:  ['.ipynb_checkpoints',
'clean_stock_prices.csv',
'project-time-series-workbook.ipynb',
'vehicle_data.csv',
'vehicle_dataset_project.ipynb']
```

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

Out[3]:

	EGAD	KUKZ	LIMIT	SASN	WTK	CGEN	ABSA	BKG	DTK	EQTY	...	BAT	CAI
Date													
2022-01-13	12.90	385.0	320.0	22.20	130.00	54.00	11.80	30.00	59.00	49.55	...	440.0	10.
2022-01-11	13.80	385.0	320.0	20.55	134.75	44.75	11.90	30.75	59.50	52.00	...	445.0	10.
2022-01-07	13.80	420.0	320.0	21.25	132.00	37.05	11.80	29.05	60.00	53.00	...	442.0	10.
2022-01-06	13.80	420.0	320.0	20.25	130.75	33.70	11.80	29.30	60.00	53.00	...	442.0	10.
2022-01-05	12.85	420.0	320.0	19.95	130.75	30.60	11.75	29.50	59.75	53.00	...	442.0	10.

5 rows × 60 columns



In [14]: `df.tail()`

Out[14]:

	EGAD	KUKZ	LIMIT	SASN	WTK	CGEN	ABSA	BKG	DTK	EQTY	...	BAT	CAI
Date													
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 [6]: `# Lowest price for Safaricom`
`df['SCOM'].nsmallest(1,)`

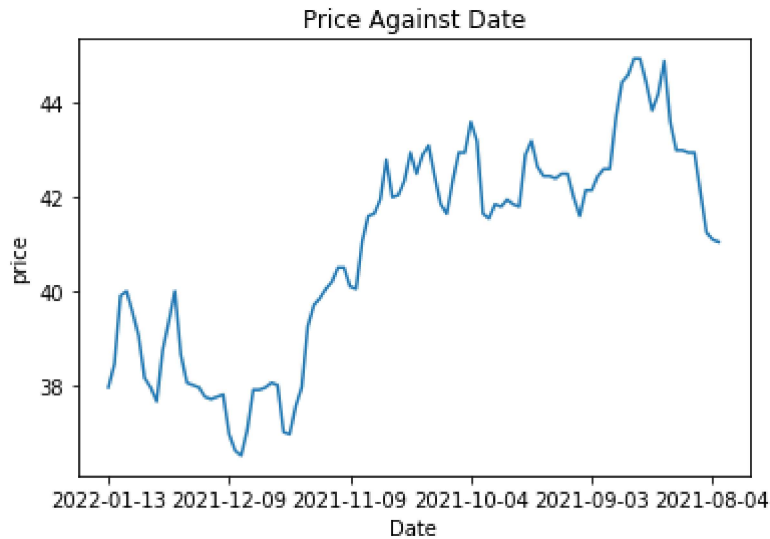
Out[6]: Date
 2021-12-07 36.5
 Name: SCOM, dtype: float64

In [7]: `# highest price for Safaricom`
`df['SCOM'].nlargest(1,)`

Out[7]: Date
 2021-08-24 44.95
 Name: SCOM, dtype: float64

```
In [15]: # Plot SCOM to confirm above observations
df['SCOM'].plot()
plt.ylabel("price")
plt.title('Price Against Date')
```

```
Out[15]: Text(0.5, 1.0, 'Price Against Date')
```



Use this part to answer question 4

```
In [ ]:  # agricultural 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 [16]:  bank_df = df.loc[:, 'ABSA': 'KCB'].copy()
bank_df.head()
```

Out[16]:

	ABSA	BKG	DTK	EQTY	HFCK	IMH	KCB
Date							
2022-01-13	11.80	30.00	59.00	49.55	3.64	21.00	45.25
2022-01-11	11.90	30.75	59.50	52.00	3.81	21.50	45.85
2022-01-07	11.80	29.05	60.00	53.00	3.81	21.40	46.00
2022-01-06	11.80	29.30	60.00	53.00	3.89	21.45	45.90
2022-01-05	11.75	29.50	59.75	53.00	3.81	21.45	45.50

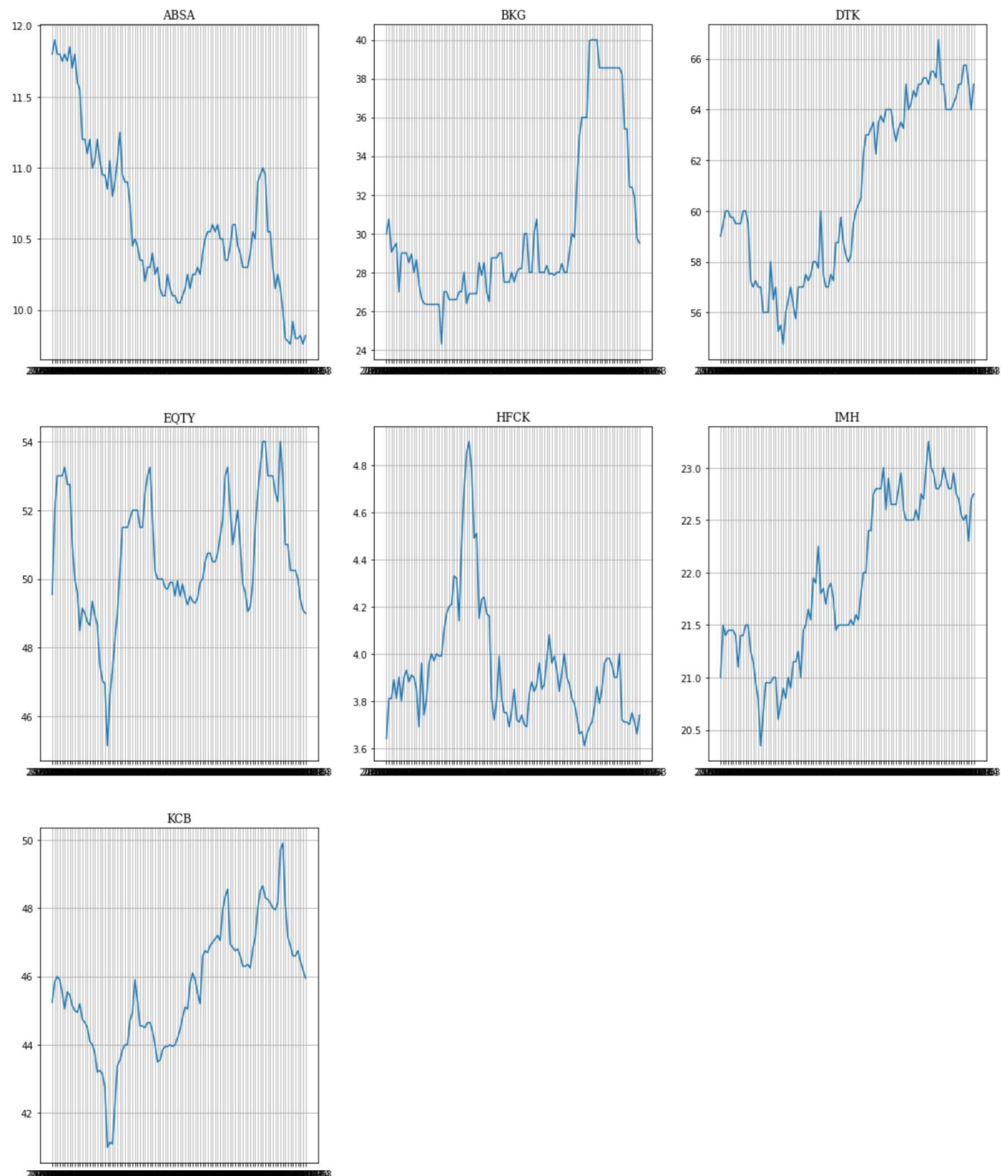
```
In [17]: sector_name_df=bank_df.copy()  
sector_name_df.head()
```

Out[17]:

	ABSA	BKG	DTK	EQTY	HFCK	IMH	KCB
Date							
2022-01-13	11.80	30.00	59.00	49.55	3.64	21.00	45.25
2022-01-11	11.90	30.75	59.50	52.00	3.81	21.50	45.85
2022-01-07	11.80	29.05	60.00	53.00	3.81	21.40	46.00
2022-01-06	11.80	29.30	60.00	53.00	3.89	21.45	45.90
2022-01-05	11.75	29.50	59.75	53.00	3.81	21.45	45.50

```
In [39]: bank_cols=sector_name_df.columns
font={'family':'serif',
      'color':'black',
      'weight':'normal'
      }
for idx, bank in enumerate (bank_cols, start=1):
    plt.subplot(4,3,idx)
    plt.title(bank,fontdict=font)
    plt.grid()
    plt.plot(bank,data=sector_name_df)

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



```
In [31]: ▶ sector_name_df.corr(method='pearson')
```

Out[31]:

	ABSA	BKG	DTK	EQTY	HFCK	IMH	KCB
ABSA	1.000000	-0.247357	-0.367356	0.051548	0.089564	-0.497121	-0.242094
BKG	-0.247357	1.000000	0.733606	0.431693	-0.363877	0.685030	0.722452
DTK	-0.367356	0.733606	1.000000	0.377873	-0.472187	0.907300	0.865433
EQTY	0.051548	0.431693	0.377873	1.000000	0.177967	0.468084	0.661149
HFCK	0.089564	-0.363877	-0.472187	0.177967	1.000000	-0.312478	-0.263884
IMH	-0.497121	0.685030	0.907300	0.468084	-0.312478	1.000000	0.850515
KCB	-0.242094	0.722452	0.865433	0.661149	-0.263884	0.850515	1.000000

```
In [38]: ▶ plt.figure(figsize=(13,8))
sns.heatmap(sector_name_df.corr(method="pearson"), annot=True, cmap='RdYlGn')
plt.figure()
```

Out[38]: <Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>

In []: ▶