Getting Started with Python

Time Series Project

Author : KENEDY MBITI

Date: 26/3/2022

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 [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 [4]:
```
os.listdir()
```

Out[4]:
```
['.ipynb_checkpoints',
 '.png',
 'cleaned_stock.csv',
 'clean_stock_data.csv',
 'clean_stock_prices.csv',
 'desktop.ini',
 'KENNEDY-CV.doc',
 'microsoft.microsoftskydrive_8wekyb3d8bbwe!App',
 'new_daily_prices.csv',
 'project-time-series-workbook.ipynb',
 'R-4.1.3-win.exe',
 'receipt.pdf',
 'RStudio-2022.02.0-443.exe',
 'student_copy_pandas_workbook.ipynb',
 'student_workbook_stocks-Copy1.ipynb',
 'student_workbook_stocks.ipynb',
 'student_workbook_stocks.py',
 'Telegram Desktop',
 'top-5-regions.png',
 'Untitled.ipynb',
 'Untitled1.ipynb',
 'Untitled2.ipynb',
 'vehicle_data (1).csv',
 'vehicle_data (2).csv',
 'vehicle_data (3).csv',
 'vehicle_data (4).csv',
 'vehicle_data (5).csv',
 'vehicle_data (6).csv',
 'vehicle_data (7).csv',
 'vehicle_data.csv',
 'vehicle_dataset_project.ipynb',
 'vehicle_dataset_project.pdf',
 'vehicle_dataset_project.zip']
```

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 [2]:
```
# read in the necessary file ('clean_stock_prices.csv')
df = pd.read_csv('clean_stock_prices.csv',index_col=0)
df.head()
```

Out[2]:

Date	EGAD	KUKZ	LIMT	SASN	WTK	CGEN	ABSA	BKG	DTK	EQTY	...	BAT	CARB	EABL	EVRD	FT
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	10.80	151.50	0.96	

Date											...				
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	10.85	161.00	0.88
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	10.90	164.75	0.94
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	10.90	160.75	0.99
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	10.90	163.75	0.99

5 rows × 60 columns

In [47]:
```
df.tail()
```

Out[47]:

Date	EGAD	KUKZ	LIMT	SASN	WTK	CGEN	ABSA	BKG	DTK	EQTY	...	BAT	CARB	EABL	EVRD	FT
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	12.25	179.25	0.96	
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	12.25	179.00	0.98	
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	12.20	178.50	0.98	
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	12.00	179.75	0.98	
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	12.00	180.00	0.98	

5 rows × 60 columns

Use this part to answer questions 1, 2 and 3

In [3]:
```
# lowest price for Safaricom
df['SCOM'].min()
```

Out[3]:
```
36.5
```

In [4]:
```
# highest price for Safaricom
df['SCOM'].max()
```

Out[4]:
```
44.95
```

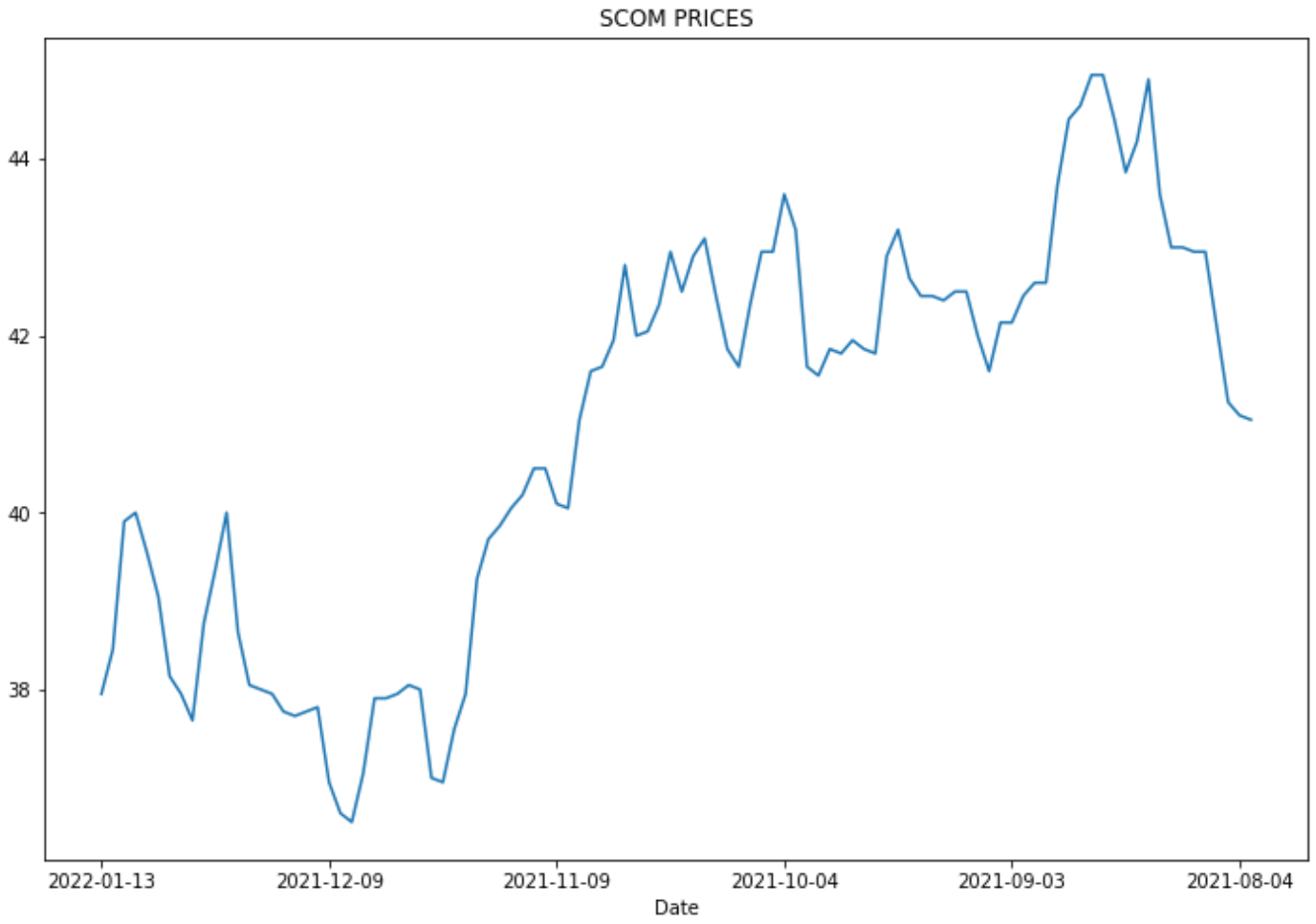
In [5]:
```
# Plot SCOM to confirm above observations
df['SCOM']
```

Out[5]:
```
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 [6]:
```
plt.figure(figsize=(12,8))
plt.title('SCOM PRICES')
df['SCOM'].plot()
```

Out[6]:
```
<AxesSubplot:title={'center':'SCOM PRICES'}, xlabel='Date'>
```



Use this part to answer question 4

In [70]:
```
# 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']
```

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

In [9]:
```
insur_df = df.loc[:,'BRIT':'SLAM'].copy()
insur_df.head()
```

Out[9]:

Date	BRIT	CIC	JUB	KNRE	LBTY	SLAM
2022-01-13	7.26	2.17	310.00	2.27	7.00	10.50
2022-01-11	7.14	2.17	310.00	2.32	7.00	10.60
2022-01-07	7.52	2.13	310.00	2.30	7.04	11.55
2022-01-06	7.52	2.15	310.50	2.29	7.04	11.55
2022-01-05	7.50	2.10	316.75	2.30	7.04	11.55

In [10]:
```
invest_df = df.loc[:,'CTUM':'NSE'].copy()
invest_df.head()
```

Out[10]:

Date	CTUM	HAFR	KURV	OCH	TCL	NSE
2022-01-13	14.65	0.38	1500.0	1.80	1.36	8.36
2022-01-11	14.35	0.40	1500.0	1.84	1.36	8.26
2022-01-07	14.40	0.40	1500.0	1.88	1.36	8.16
2022-01-06	14.50	0.38	1500.0	1.97	1.32	8.20
2022-01-05	14.60	0.39	1500.0	1.97	1.29	8.12

In [12]:
```
sector_name_df = df.loc[:,'CTUM':'NSE'].copy()
sector_name_df
```

Out[12]:

Date	CTUM	HAFR	KURV	OCH	TCL	NSE
2022-01-13	14.65	0.38	1500.0	1.80	1.36	8.36
2022-01-11	14.35	0.40	1500.0	1.84	1.36	8.26
2022-01-07	14.40	0.40	1500.0	1.88	1.36	8.16
2022-01-06	14.50	0.38	1500.0	1.97	1.32	8.20
2022-01-05	14.60	0.39	1500.0	1.97	1.29	8.12
...	...	...	...	...	...	...
2021-08-09	15.75	0.41	1500.0	2.20	1.36	9.10
2021-08-06	15.55	0.41	1500.0	2.18	1.31	8.48
2021-08-05	15.65	0.40	1500.0	2.18	1.28	8.52
2021-08-04	15.65	0.41	1500.0	2.19	1.35	8.62

2021-08-03	15.85	0.40	1500.0	2.19	1.37	8.94

102 rows × 6 columns

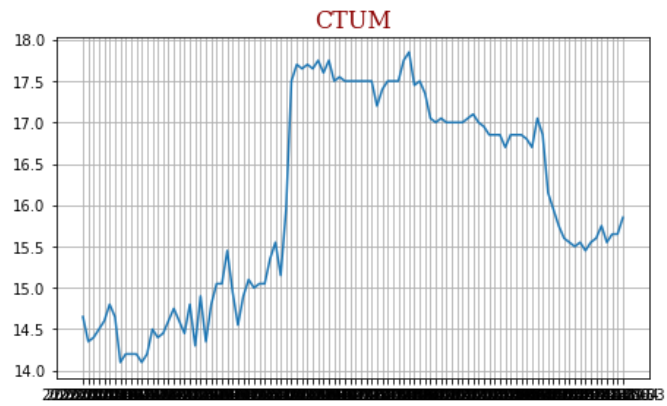
In [16]:
```
sector_name_df.columns
```

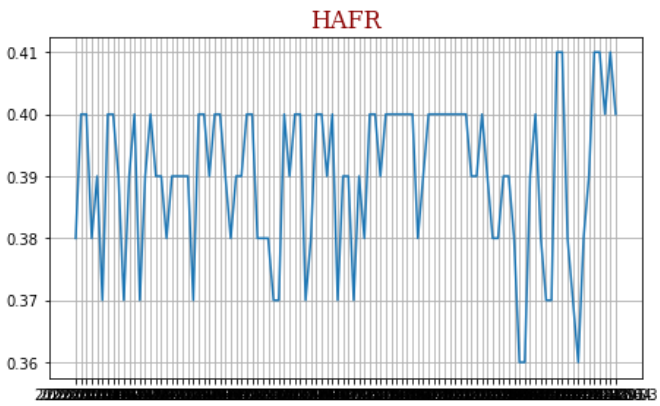
Out[16]:
```
Index(['CTUM', 'HAFR', 'KURV', 'OCH', 'TCL', 'NSE'], dtype='object')
```

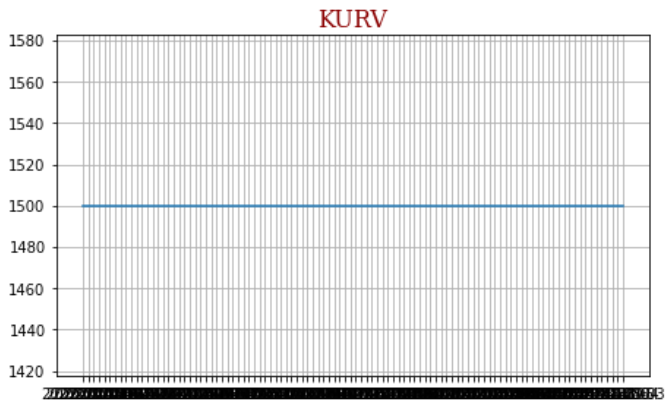
In [17]:
```
sector_cols = sector_name_df.columns
for sector in sector_cols:
    print(sector)
```

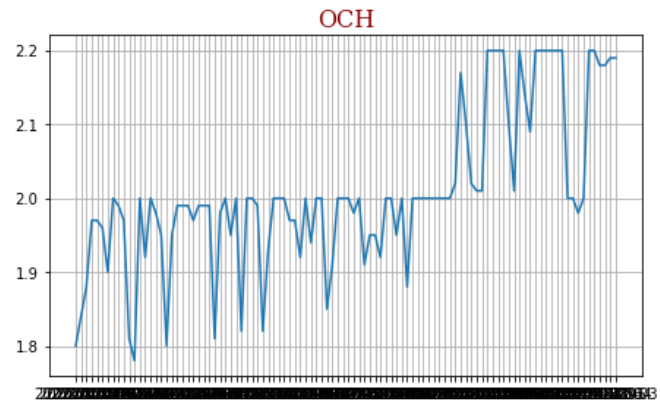
```
CTUM
HAFR
KURV
OCH
TCL
NSE
```

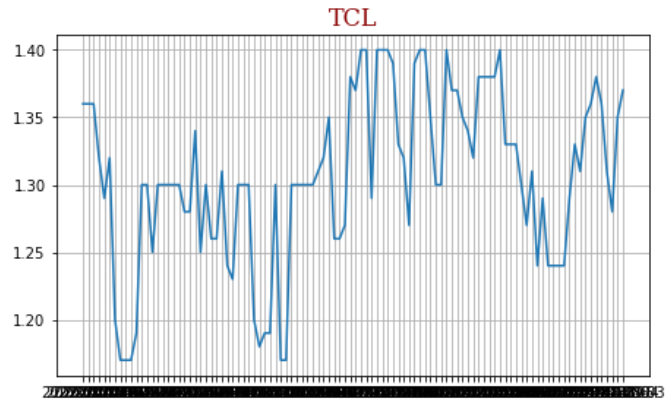
In [25]:
```
sector_cols = sector_name_df.columns

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

for idx,sector in enumerate(sector_cols,start=1):
    plt.subplot(6,2,idx)
    plt.title(sector,fontdict=font)
    plt.grid()
    plt.plot(sector,data=df)

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

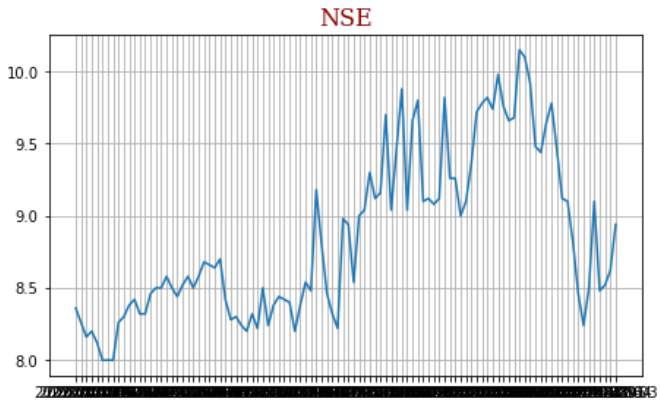












In [26]:
```
sector_name_df.corr(method='pearson')
```

Out[26]:

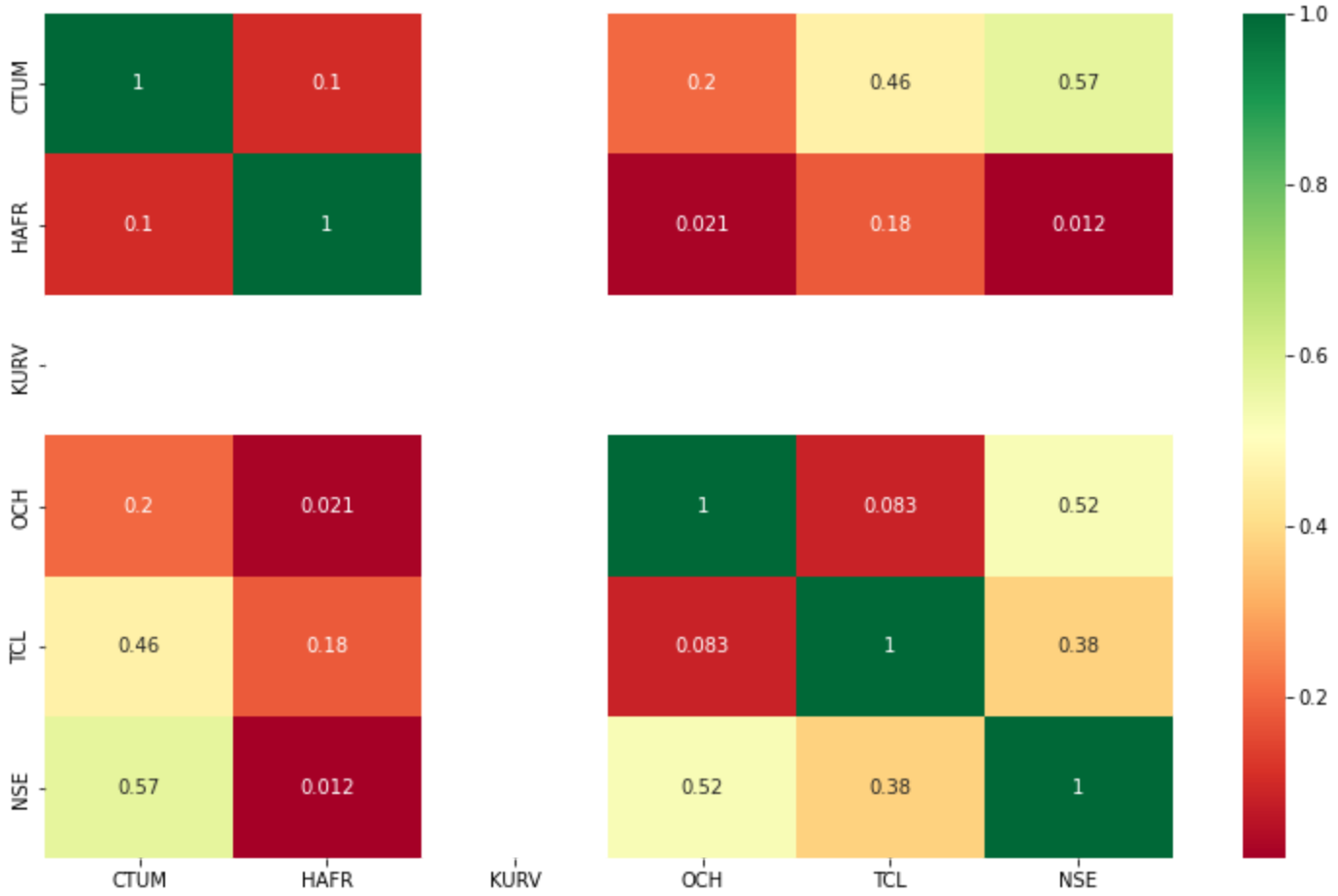
	CTUM	HAFR	KURV	OCH	TCL	NSE
CTUM	1.000000	0.102660	NaN	0.201916	0.461223	0.568061
HAFR	0.102660	1.000000	NaN	0.021208	0.181057	0.011877
KURV	NaN	NaN	NaN	NaN	NaN	NaN
OCH	0.201916	0.021208	NaN	1.000000	0.083212	0.524421
TCL	0.461223	0.181057	NaN	0.083212	1.000000	0.381206
NSE	0.568061	0.011877	NaN	0.524421	0.381206	1.000000

In [27]:
```
corr_df = sector_name_df.corr(method="pearson")
```

In [28]:
```
import seaborn as sns
```

In [29]:
```
plt.figure(figsize=(13, 8))
sns.heatmap(corr_df, annot=True, cmap='RdYlGn')
plt.figure()
```

Out[29]:
```
<Figure size 432x288 with 0 Axes>
```



<Figure size 432x288 with 0 Axes>

In [30]:
```
sector_name_df.head()
```

Out[30]:

Date	CTUM	HAFR	KURV	OCH	TCL	NSE
2022-01-13	14.65	0.38	1500.0	1.80	1.36	8.36
2022-01-11	14.35	0.40	1500.0	1.84	1.36	8.26
2022-01-07	14.40	0.40	1500.0	1.88	1.36	8.16
2022-01-06	14.50	0.38	1500.0	1.97	1.32	8.20
2022-01-05	14.60	0.39	1500.0	1.97	1.29	8.12

In [31]:
```
#drop KURV
sector_name_df.drop('KURV',axis="columns")
```

Out[31]:

Date	CTUM	HAFR	OCH	TCL	NSE
2022-01-13	14.65	0.38	1.80	1.36	8.36
2022-01-11	14.35	0.40	1.84	1.36	8.26
2022-01-07	14.40	0.40	1.88	1.36	8.16
2022-01-06	14.50	0.38	1.97	1.32	8.20
2022-01-05	14.60	0.39	1.97	1.29	8.12
...	...	...	...	...	...
2021-08-09	15.75	0.41	2.20	1.36	9.10

2021-08-06	15.55	0.41	2.18	1.31	8.48
2021-08-05	15.65	0.40	2.18	1.28	8.52
2021-08-04	15.65	0.41	2.19	1.35	8.62
2021-08-03	15.85	0.40	2.19	1.37	8.94

102 rows × 5 columns

In [32]:
```
# update the original df
sector_name_df.drop('KURV',axis="columns", inplace=True)
```

In [33]:
```
sector_name_df.head()
```

Out[33]:

Date	CTUM	HAFR	OCH	TCL	NSE
2022-01-13	14.65	0.38	1.80	1.36	8.36
2022-01-11	14.35	0.40	1.84	1.36	8.26
2022-01-07	14.40	0.40	1.88	1.36	8.16
2022-01-06	14.50	0.38	1.97	1.32	8.20
2022-01-05	14.60	0.39	1.97	1.29	8.12

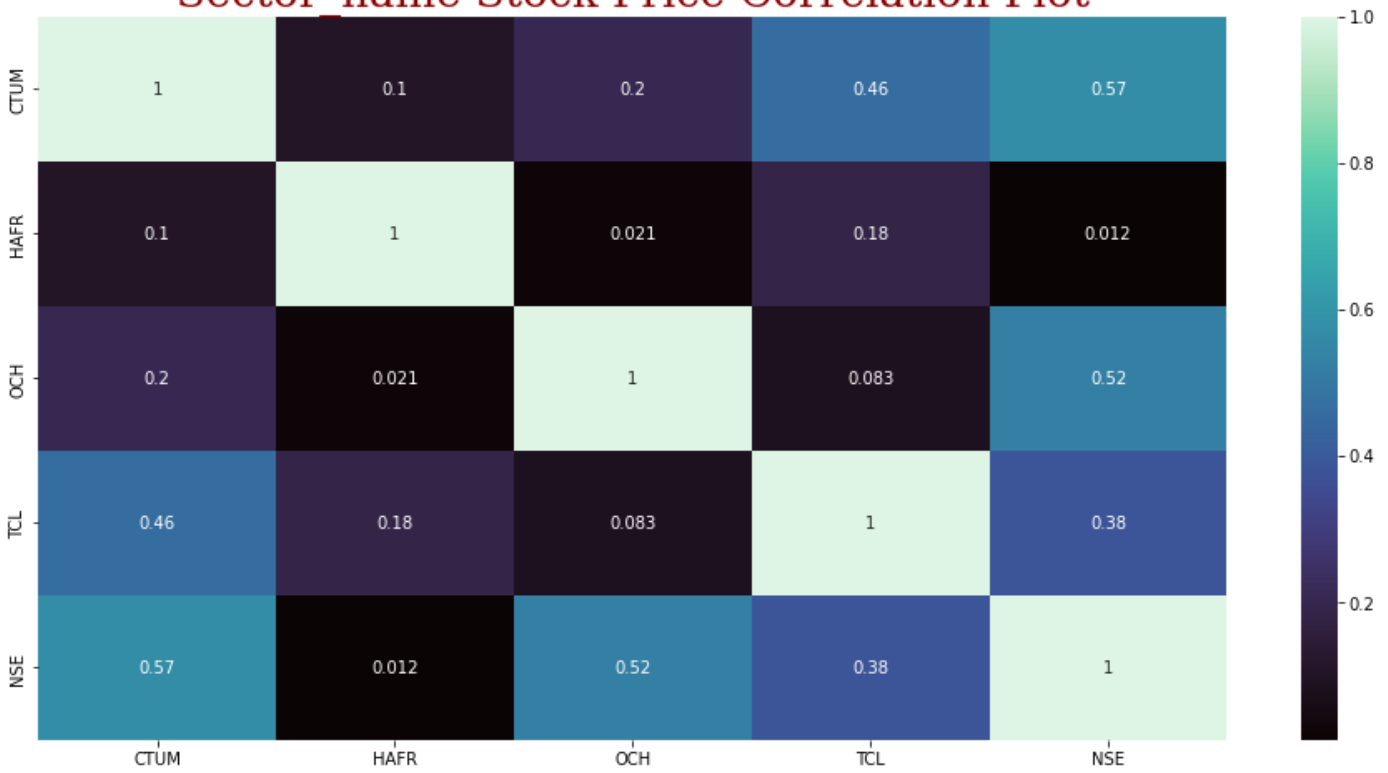
In [34]:
```
corr_df = sector_name_df.corr(method="pearson")
```

In [35]:
```
corr_df.head()
```

Out[35]:

	CTUM	HAFR	OCH	TCL	NSE
CTUM	1.000000	0.102660	0.201916	0.461223	0.568061
HAFR	0.102660	1.000000	0.021208	0.181057	0.011877
OCH	0.201916	0.021208	1.000000	0.083212	0.524421
TCL	0.461223	0.181057	0.083212	1.000000	0.381206
NSE	0.568061	0.011877	0.524421	0.381206	1.000000

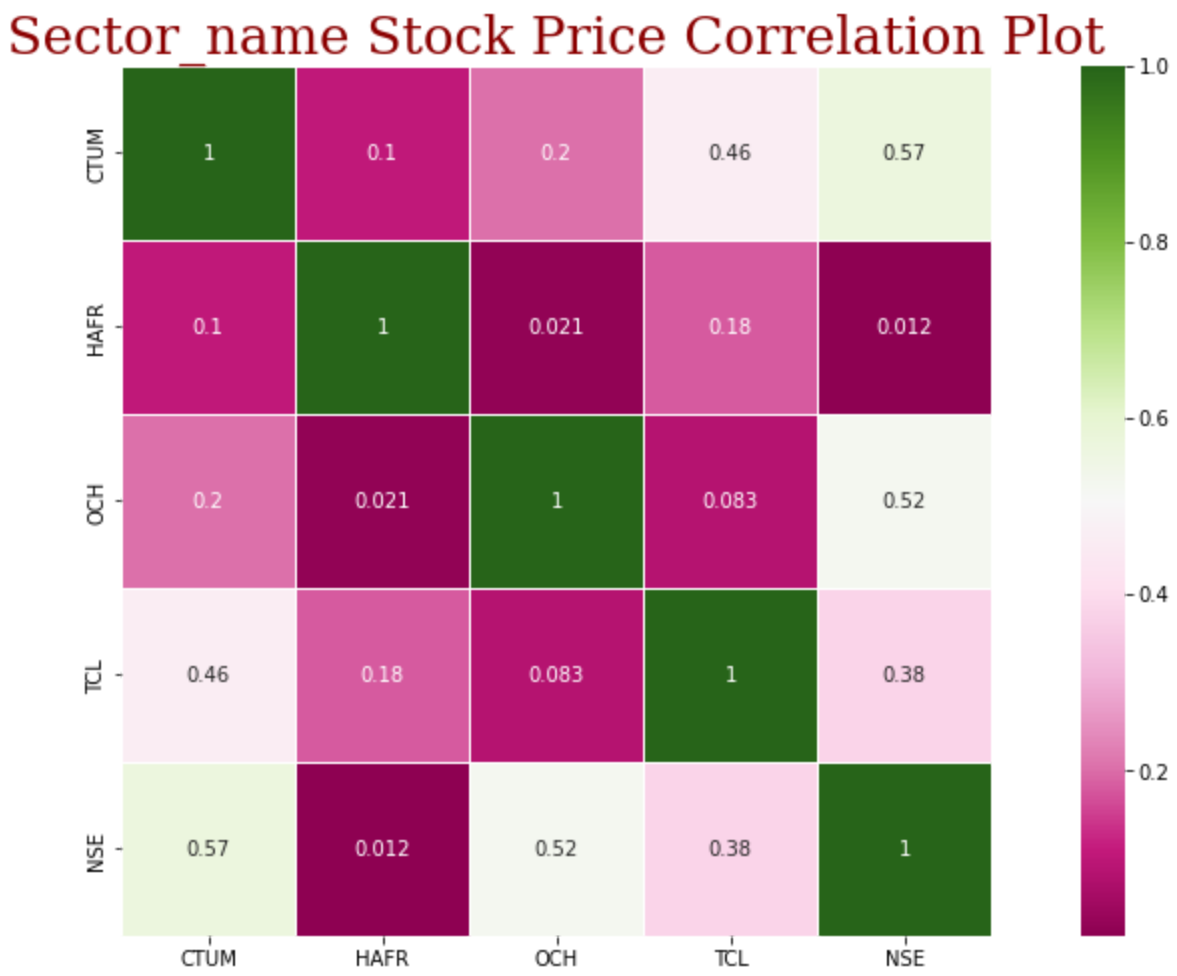
In [49]:
```
# customize text
font = {'family': 'serif',
        'color':  'darkred',
        'weight': 'normal',
        'size': 26,
        }

plt.figure(figsize=(16,8))
plt.title("Sector_name 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>
```

In [45]:

```
# customize text
font = {'family': 'serif',
        'color':  'darkred',
        'weight': 'normal',
        'size': 26,
        }

plt.figure(figsize=(16,8))
plt.title("Sector_name Stock Price Correlation Plot", fontdict=font)
cmap = ["mako","PiYG","YlGnBu","Blues"]
sns.heatmap(corr_df, annot=True, cmap=cmap[np.random.randint(len(cmap))],linewidth=1, li
plt.figure()
plt.show()
```



<Figure size 432x288 with 0 Axes>