Getting Started with Python

Time Series Project

Author : Ian odhiambo otieno

Date: 27/3/2022

Resources

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

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 [6]:
```
import os
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
```

In [7]:
```
os.listdir()
```

Out[7]:
```
['.ipynb_checkpoints',
 'access data',
 'banking_stock_prices.png',
 'clean_stock_data.csv',
 'clean_stock_prices.csv',
 'datatypes.py',
 'day 3(loops).py',
 'day3.py',
 'day4.py',
 'example12.py',
 'leap_year.py',
 'loop2.py',
 'loops.py',
 'manu dummy book 1 (Autosaved).xlsx',
 'manu dummy book 1.xlsx',
 'membership_operator.py',
 'multinest.py',
 'new_daily_prices.csv',
 'NJIRU HEALTH CENTER- COVID VACCINATION REPORT.xlsx',
 'power pivot.xlsx',
 'project-time-series-workbook.ipynb',
 'RUAI HEALTH CENTRE 2.xlsx',
 'RUAI HEALTH CENTRE(NEW).xlsx',
 'RUAI HEALTH CENTRE.xlsx',
 'student_copy_pandas_workbook.ipynb',
 'student_workbook_stocks.ipynb',
 'top-10-brands.png',
 'top-10-regions.png',
 'vehicle_data.csv',
 'vehicle_dataset_project.ipynb',
 'weeek2_looping.py',
 'WEEK ONE PROJECT.docx',
 'week one project.xlsx']
```

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 [8]:
```
# read in the necessary file ('clean_stock_prices.csv')
df = pd.read_csv('clean_stock_prices.csv',index_col=0)
df.head()
```

Out[8]:

Date	EGAD	KUKZ	LIMT	SASN	WTK	CGEN	ABSA	BKG	DTK	EQTY	...	BAT	CARB	
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	1
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	1
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	1
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	1
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	1

5 rows × 60 columns

In [9]:
```
df.tail()
```

Out[9]:

Date	EGAD	KUKZ	LIMT	SASN	WTK	CGEN	ABSA	BKG	DTK	EQTY	...	BAT	CARB	
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	1
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	1
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	1
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	1
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	1

5 rows × 60 columns

questions 1, 2 and 3

In [10]:
```
# lowest price for Safaricom
df["SCOM"].nsmallest(1)
```

Out[10]:
```
Date
2021-12-07    36.5
Name: SCOM, dtype: float64
```

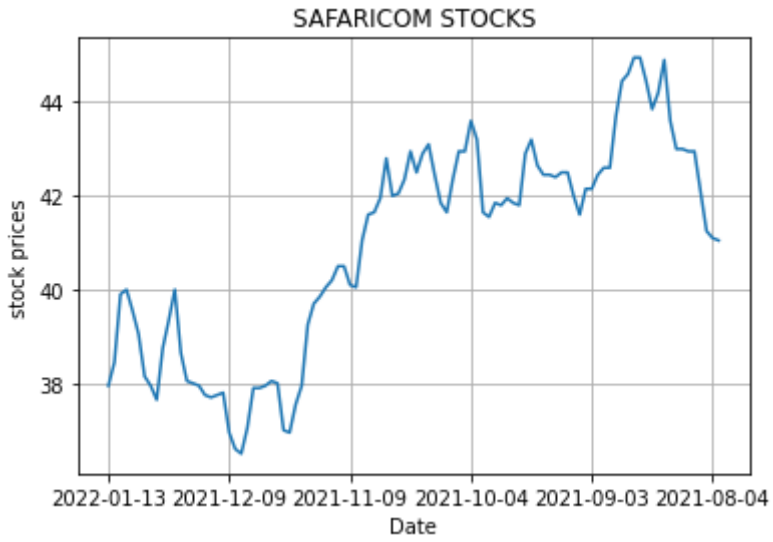
In [21]:
```
# highest price for Safaricom
df["SCOM"].nlargest(1)
```

Out[21]:
```
Date
2021-08-24    44.95
Name: SCOM, dtype: float64
```

In [94]:
```
# Plot SCOM to confirm above observations
df["SCOM"].plot()
```

```
plt.grid()
plt.title("SAFARICOM STOCKS")
plt.ylabel("stock prices")
```

Out[94]:
```
Text(0, 0.5, 'stock prices')
```



Sectors data

In [ ]:
```
# 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',

# 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 [13]:
```
insur_df= df.loc[:,'BRIT':'SLAM'].copy()
insur_df.head()
```

Out[13]:

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 [28]:
```
insur_cols = insur_df.columns
for insur in insur_cols:
    print(insur)
```

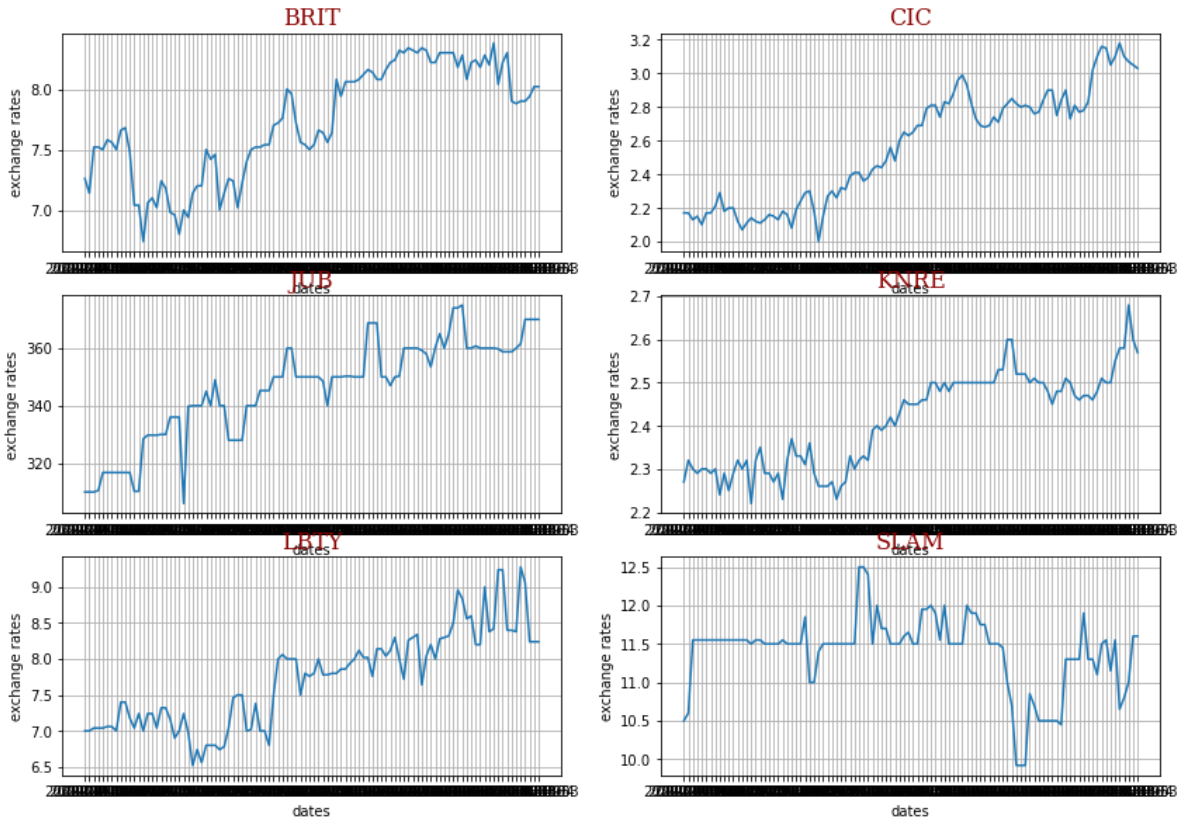
```
BRIT
CIC
JUB
KNRE
LBTY
SLAM
```

In [31]:
```
insur_cols = insur_df.columns

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

for idx,insur in enumerate(insur_cols,start=1):
    plt.subplot(6,2,idx)
    plt.title(insur,fontdict=font)
    plt.xlabel("dates")
    plt.ylabel("exchange rates")
    plt.grid()
    plt.plot(insur,data=df)

fig = plt.gcf()
fig.set_size_inches(14,20)
plt.show()
```



In [14]:
```
agric_df = df.loc[:,'EGAD':'WTK'].copy()
agric_df.head()
```

Out[14]:

Date	EGAD	KUKZ	LIMT	SASN	WTK
2022-01-13	12.90	385.0	320.0	22.20	130.00
2022-01-11	13.80	385.0	320.0	20.55	134.75
2022-01-07	13.80	420.0	320.0	21.25	132.00
2022-01-06	13.80	420.0	320.0	20.25	130.75
2022-01-05	12.85	420.0	320.0	19.95	130.75

In [32]:
```
agric_cols = agric_df.columns
for agric in agric_cols:
    print(agric)
```

```
EGAD
KUKZ
LIMT
SASN
WTK
```

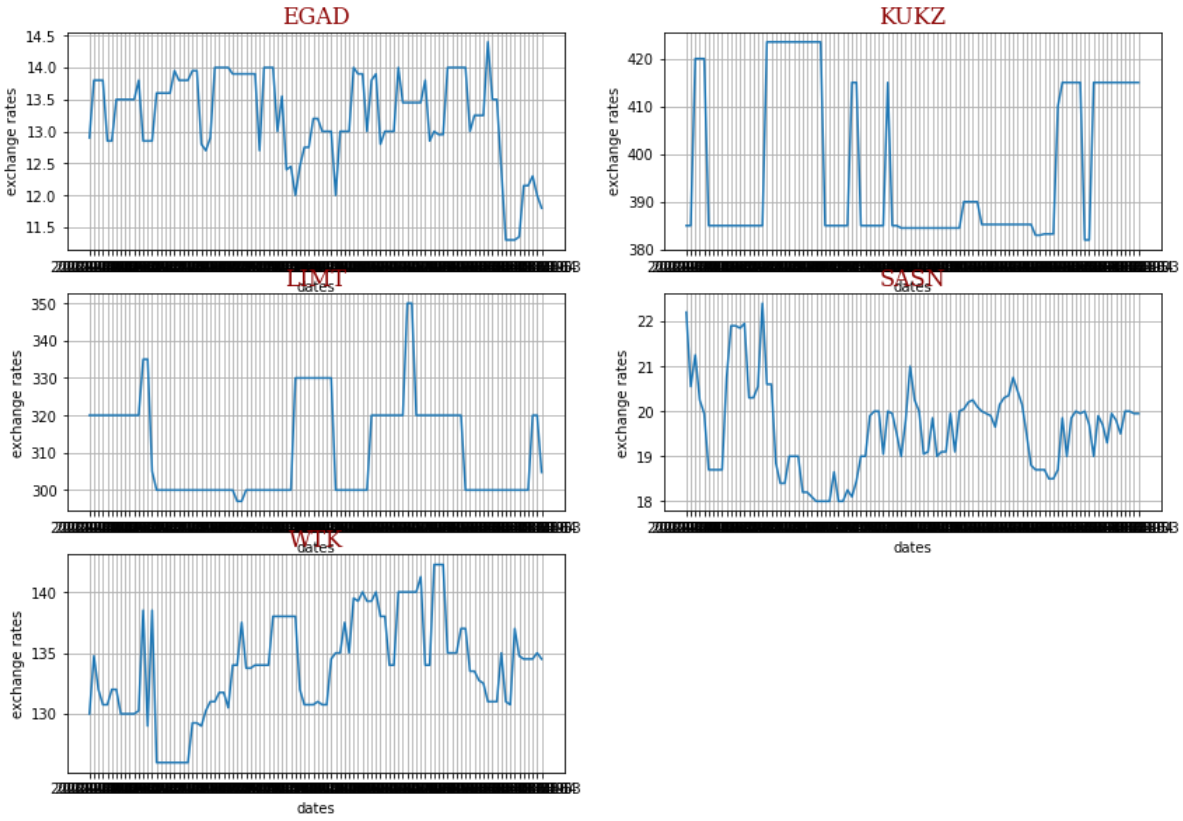
In [33]:
```
agric_cols = agric_df.columns

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

for idx,agric in enumerate(agric_cols,start=1):
    plt.subplot(6,2,idx)
    plt.title(agric,fontdict=font)
```

```
    plt.xlabel("dates")
    plt.ylabel("exchange rates")
    plt.grid()
    plt.plot(agric,data=df)

fig = plt.gcf()
fig.set_size_inches(14,20)
plt.show()
```



In [15]:
```
com_df = df.loc[:,'XPRS':'UCHM'].copy()
com_df.head()
```

Out[15]:

Date	XPRS	KQ	LKL	NBV	NMG	SMER	SCAN	SGL	TPSE	UCHM
2022-01-13	3.37	3.83	3.98	5.12	18.55	2.75	4.10	14.10	15.00	0.23
2022-01-11	3.50	3.83	4.00	5.30	19.00	2.82	4.14	15.50	15.00	0.21
2022-01-07	3.90	3.83	3.99	4.94	19.45	2.78	4.18	14.90	15.25	0.23
2022-01-06	4.10	3.83	4.00	4.89	19.90	2.80	4.12	14.90	15.25	0.23
2022-01-05	4.10	3.83	4.00	5.06	19.40	2.80	4.16	13.55	15.25	0.23

In [34]:
```
com_cols = com_df.columns
for com in com_cols:
    print(com)
```

```
XPRS
KQ
LKL
NBV
NMG
SMER
SCAN
SGL
TPSE
UCHM
```

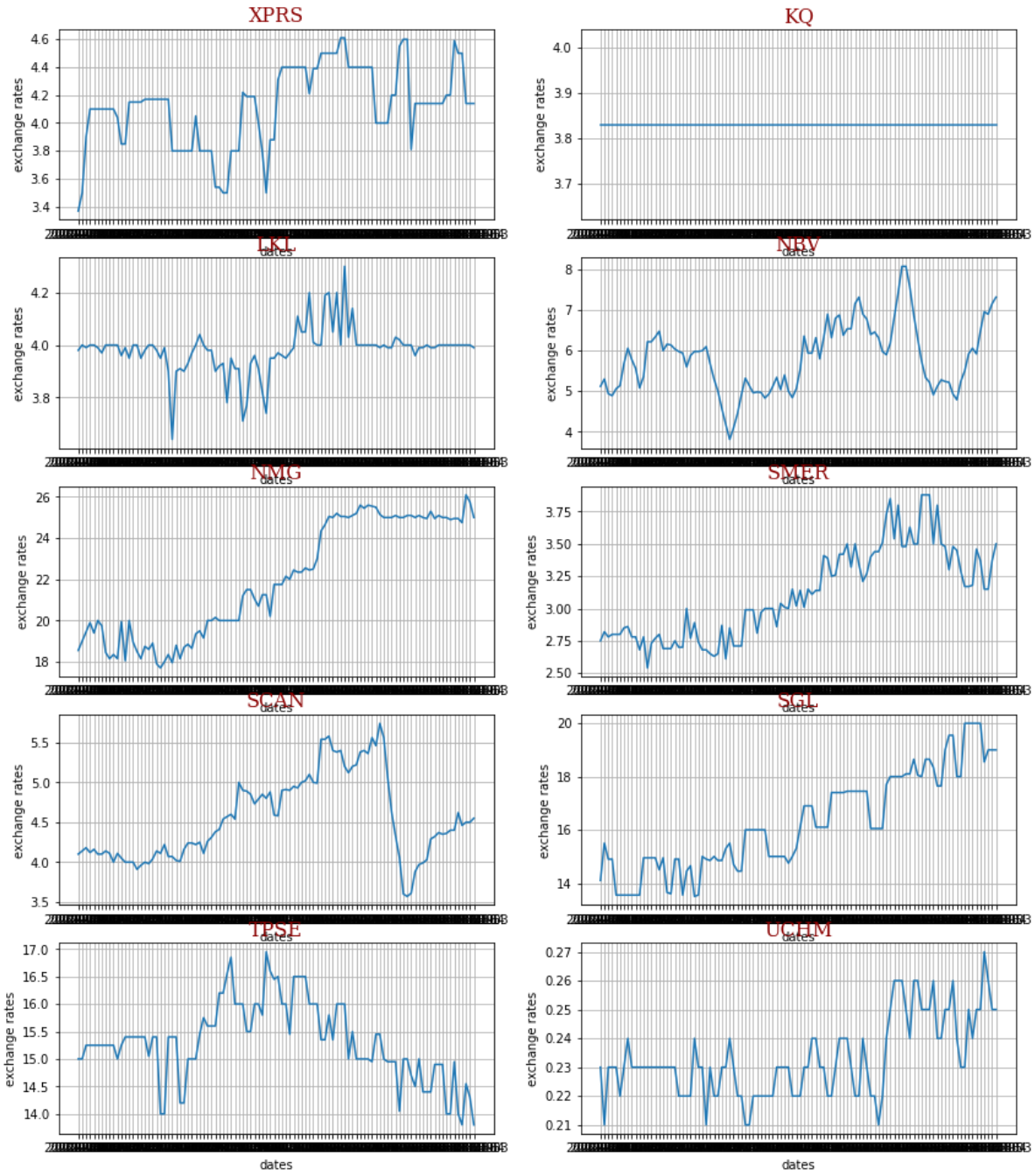
In [35]:

```
com_cols = com_df.columns

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

for idx,com in enumerate(com_cols,start=1):
    plt.subplot(6,2,idx)
    plt.title(com,fontdict=font)
    plt.xlabel("dates")
    plt.ylabel("exchange rates")
    plt.grid()
    plt.plot(com,data=df)

fig = plt.gcf()
fig.set_size_inches(14,20)
plt.show()
```



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

Out[16]:

Date	ABSA	BKG	DTK	EQTY	HFCK	IMH	KCB	NBK	NCBA	SBIC	SCBK	COOP
2022-01-13	11.80	30.00	59.00	49.55	3.64	21.00	45.25	4.12	25.70	88.5	129.50	12.55
2022-01-11	11.90	30.75	59.50	52.00	3.81	21.50	45.85	4.12	25.95	87.5	130.00	12.80
2022-01-07	11.80	29.05	60.00	53.00	3.81	21.40	46.00	4.12	25.95	87.0	130.50	12.95
2022-01-06	11.80	29.30	60.00	53.00	3.89	21.45	45.90	4.12	25.90	87.0	130.75	13.00
2022-01-05	11.75	29.50	59.75	53.00	3.81	21.45	45.50	4.12	25.55	87.0	130.00	13.00

In [36]:
```
bank_cols = bank_df.columns
for bank in bank_cols:
    print(bank)
```

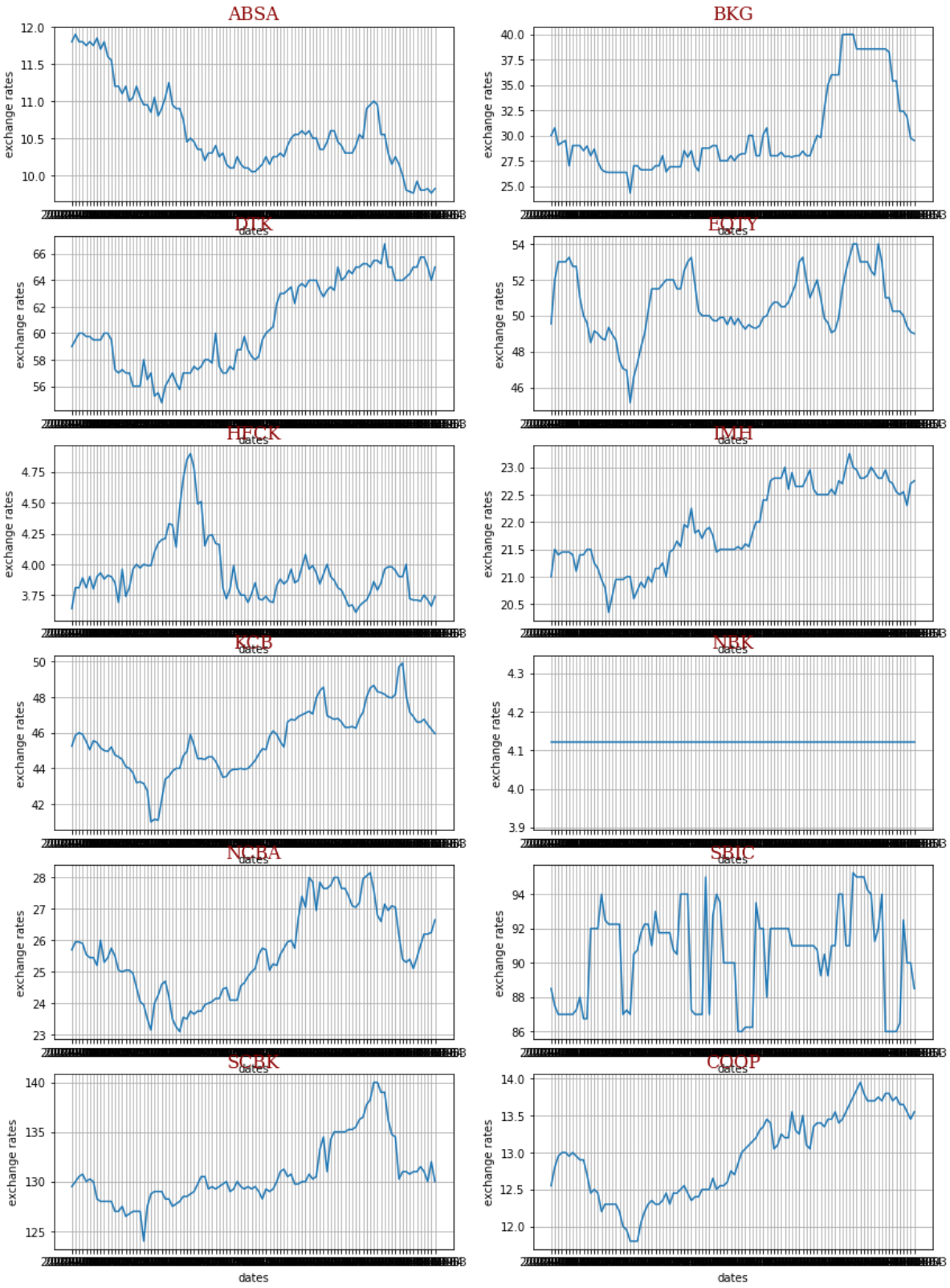
```
ABSA
BKG
DTK
EQTY
HFCK
IMH
KCB
NBK
NCBA
SBIC
SCBK
COOP
```

In [37]:
```
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.xlabel("dates")
    plt.ylabel("exchange rates")
    plt.grid()
    plt.plot(bank,data=df)

fig = plt.gcf()
fig.set_size_inches(14,20)
plt.show()
```



In [17]:
```
const_df = df.loc[:,'ARM':'PORT'].copy()
const_df.head()
```

Out[17]:

Date	ARM	BAMB	CRWN	CABL	PORT
2022-01-13	5.55	38.05	30.75	1.19	7.02
2022-01-11	5.55	37.75	30.50	1.20	7.20
2022-01-07	5.55	38.00	30.50	1.21	7.00
2022-01-06	5.55	37.85	30.75	1.27	7.02
2022-01-05	5.55	37.95	30.50	1.22	6.90

In [39]:
```
const_cols = const_df.columns
for const in const_cols:
    print(const)
```

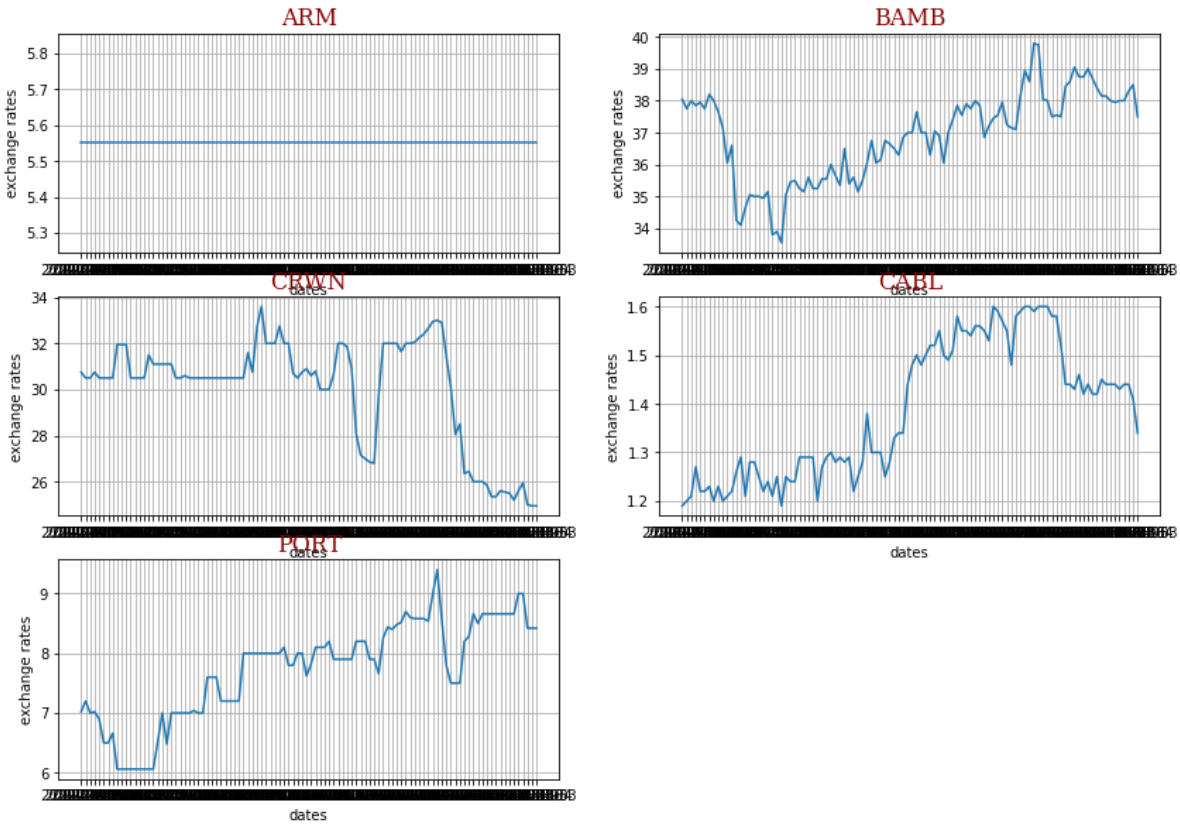
```
ARM
BAMB
CRWN
CABL
PORT
```

In [40]:
```
const_cols = const_df.columns

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

for idx,const in enumerate(const_cols,start=1):
    plt.subplot(6,2,idx)
    plt.title(const,fontdict=font)
    plt.xlabel("dates")
    plt.ylabel("exchange rates")
    plt.grid()
    plt.plot(const,data=df)

fig = plt.gcf()
fig.set_size_inches(14,20)
plt.show()
```



In [18]:
```
energy_df = df.loc[:,'KEGN':'UMME'].copy()
energy_df.head()
```

Out[18]:

Date	KEGN	KPLC	TOTL	UMME
2022-01-13	4.15	1.67	24.80	6.72
2022-01-11	4.13	1.71	24.20	6.74
2022-01-07	4.12	1.72	24.60	6.74
2022-01-06	4.13	1.72	24.55	6.74
2022-01-05	4.16	1.71	24.70	6.74

In [41]:
```
energy_cols = energy_df.columns
for energy in energy_cols:
    print(energy)
```

```
KEGN
KPLC
TOTL
UMME
```

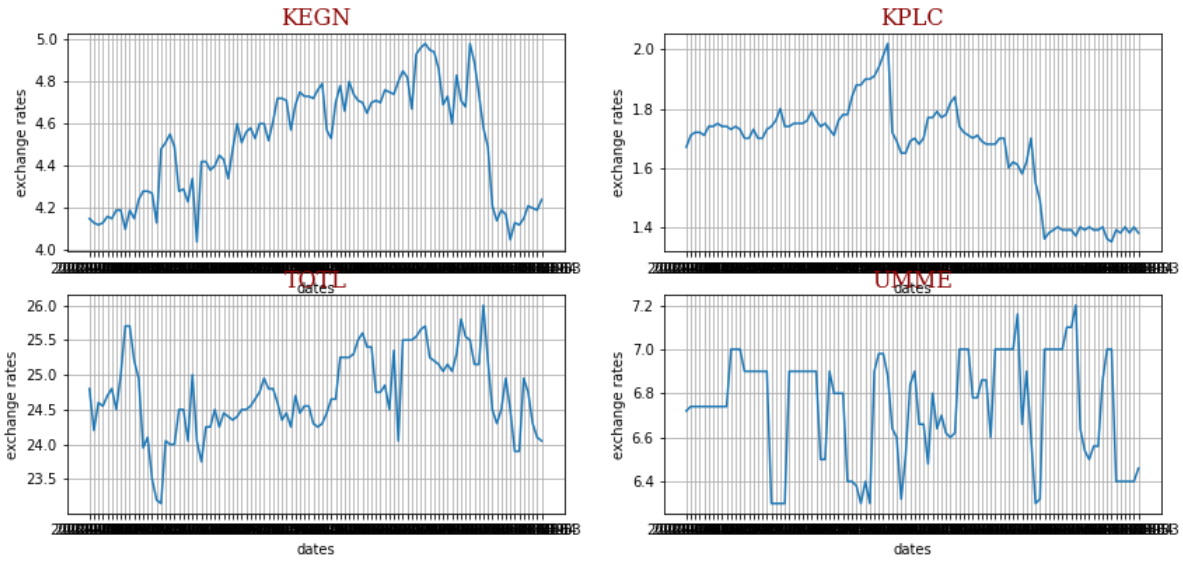
In [43]:
```
energy_cols = energy_df.columns

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

for idx,energy in enumerate(energy_cols,start=1):
    plt.subplot(6,2,idx)
    plt.title(energy,fontdict=font)
    plt.xlabel("dates")
```

```
    plt.ylabel("exchange rates")
    plt.grid()
    plt.plot(energy,data=df)

fig = plt.gcf()
fig.set_size_inches(14,20)
plt.show()
```



In [19]:
```
invest_df = df.loc[:,'CTUM':'NSE'].copy()
invest_df.head()
```

Out[19]:

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 [44]:
```
invest_cols = invest_df.columns
for invest in invest_cols:
    print(invest)
```

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

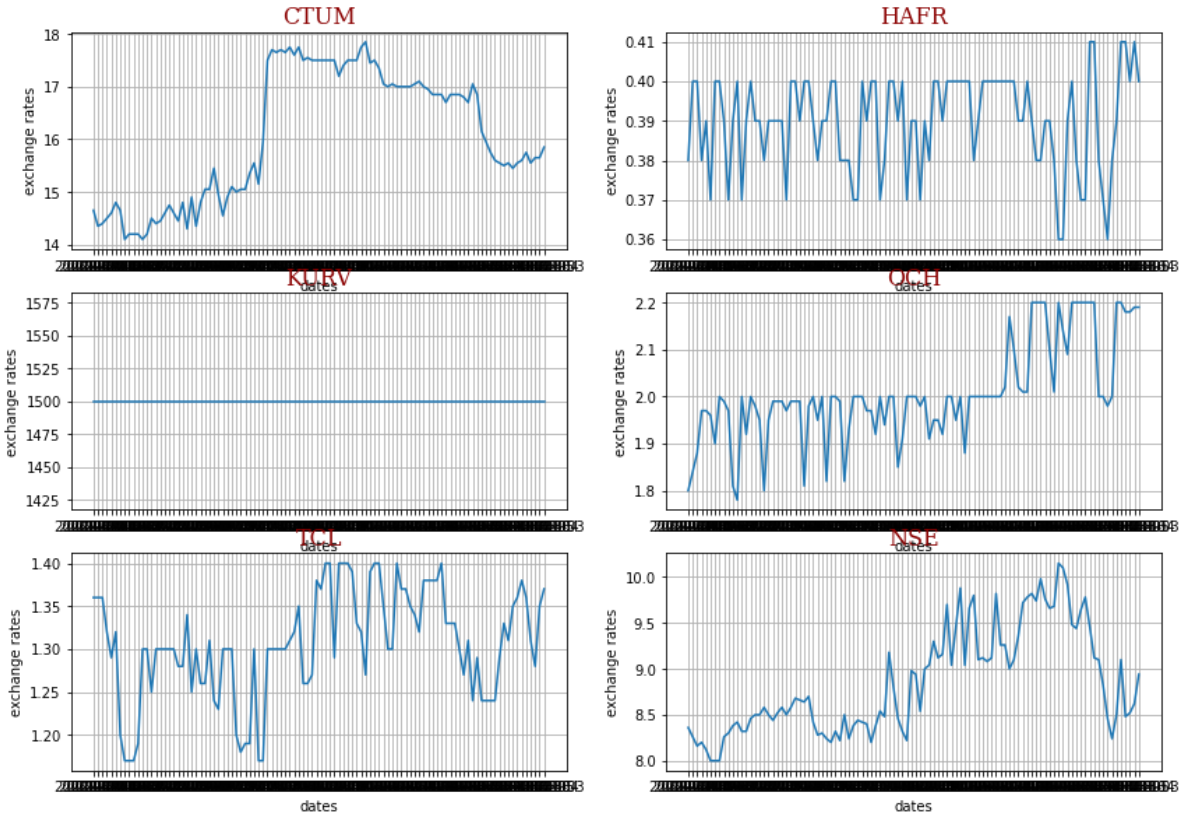
In [48]:
```
invest_cols = invest_df.columns

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

for idx,invest in enumerate(invest_cols,start=1):
    plt.subplot(6,2,idx)
```

```
    plt.title(invest,fontdict=font)
    plt.xlabel("dates")
    plt.ylabel("exchange rates")
    plt.grid()
    plt.plot(invest,data=df)

fig = plt.gcf()
fig.set_size_inches(14,20)
plt.show()
```



In [22]:
```
manu_df = df.loc[:,'BOC':'UNGA'].copy()
manu_df.head()
```

Out[22]:

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 [ ]:
```
```

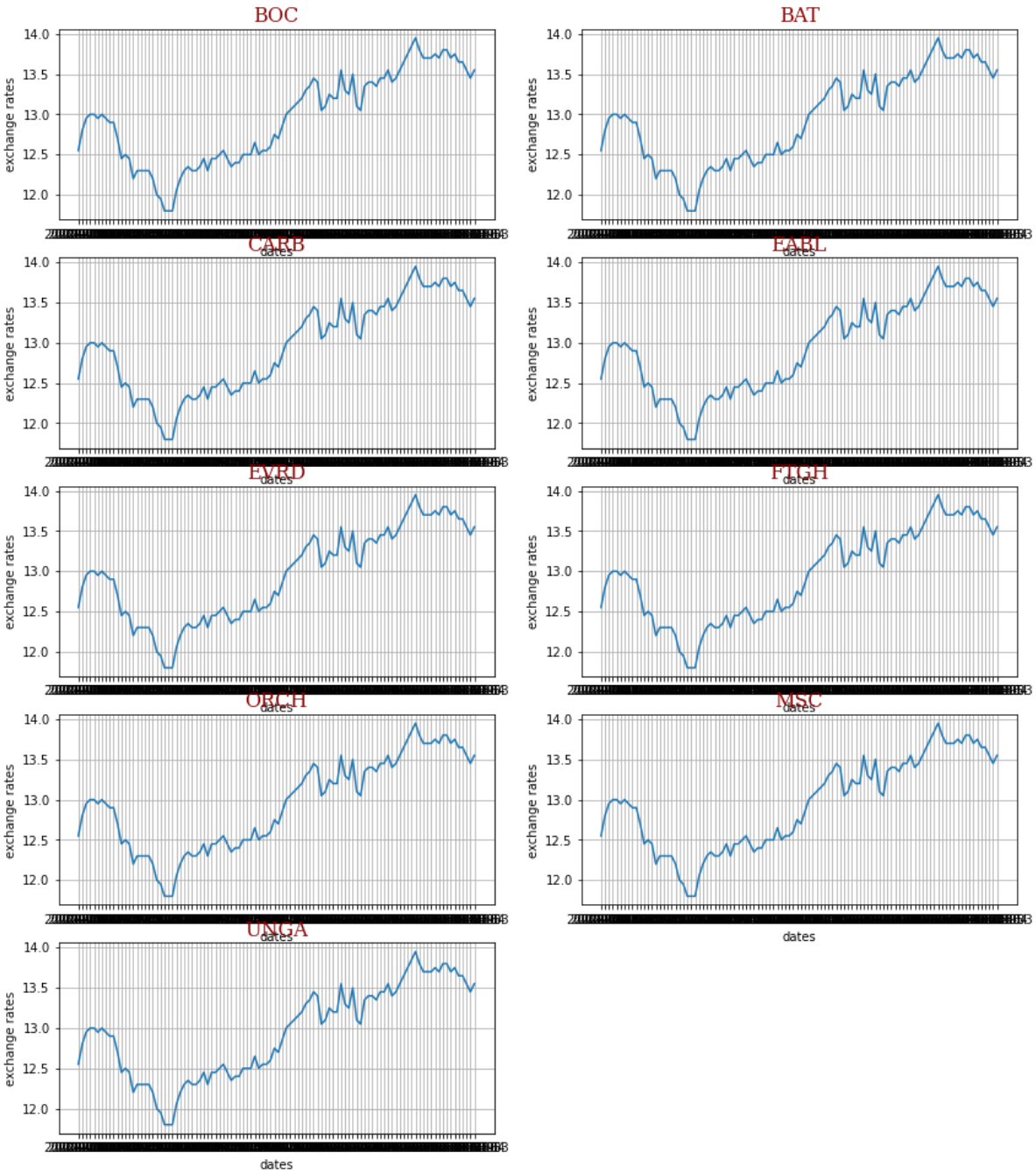
In [38]:
```
manu_cols = manu_df.columns

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

for idx,manu in enumerate(manu_cols,start=1):
```

```
    plt.subplot(6,2,idx)
    plt.title(manu,fontdict=font)
    plt.xlabel("dates")
    plt.ylabel("exchange rates")
    plt.grid()
    plt.plot(bank,data=df)

fig = plt.gcf()
fig.set_size_inches(14,20)
plt.show()
```



In [60]:
```
corr_df = bank_df.corr(method="pearson")
```

In [61]:
```
corr_df.head()
```

Out[61]:

	ABSA	BKG	DTK	EQTY	HFCK	IMH	KCB	NBK	NCBA
ABSA	1.000000	-0.247357	-0.367356	0.051548	0.089564	-0.497121	-0.242094	NaN	-0.071353
BKG	-0.247357	1.000000	0.733606	0.431693	-0.363877	0.685030	0.722452	NaN	0.546149
DTK	-0.367356	0.733606	1.000000	0.377873	-0.472187	0.907300	0.865433	NaN	0.826353
EQTY	0.051548	0.431693	0.377873	1.000000	0.177967	0.468084	0.661149	NaN	0.333037
HFCK	0.089564	-0.363877	-0.472187	0.177967	1.000000	-0.312478	-0.263884	NaN	-0.522405

In [64]:
```
# customize text
font = {'family': 'serif',
        'color':  'darkred',
        'weight': 'normal',
        'size': 26,
        }

plt.figure(figsize=(16,8))
plt.title("banking 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))],linewidth
plt.figure()
plt.show()
```

<Figure size 432x288 with 0 Axes>

In [66]:
```
insurance_df = insur_df.corr(method="pearson")
```

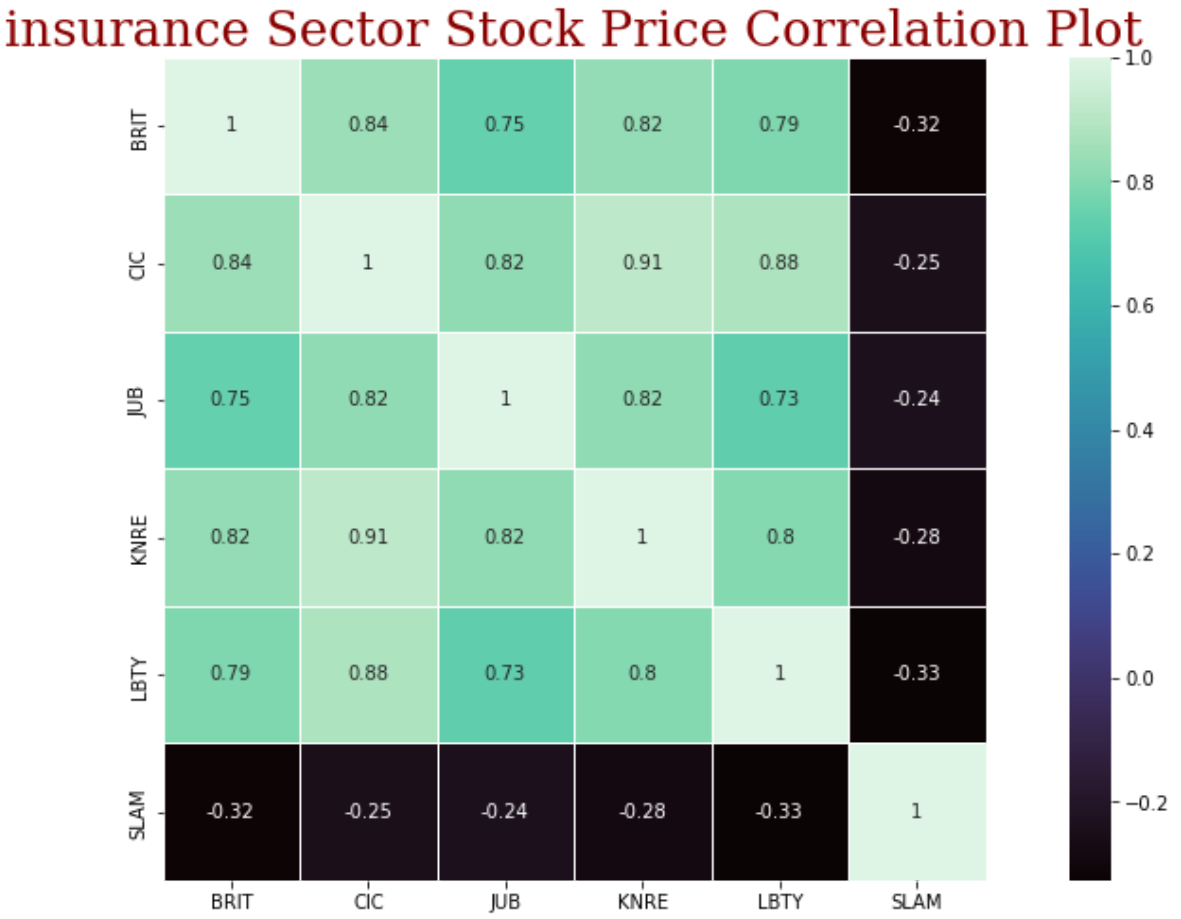
In [67]:
```
insurance_df.head()
```

Out[67]:

	BRIT	CIC	JUB	KNRE	LBTY	SLAM
BRIT	1.000000	0.843570	0.745239	0.824231	0.791978	-0.318103
CIC	0.843570	1.000000	0.819647	0.914090	0.878433	-0.247342
JUB	0.745239	0.819647	1.000000	0.819330	0.734525	-0.241677
KNRE	0.824231	0.914090	0.819330	1.000000	0.800351	-0.282419
LBTY	0.791978	0.878433	0.734525	0.800351	1.000000	-0.326399

In [68]:
```
font = {'family': 'serif',
        'color':  'darkred',
        'weight': 'normal',
        'size': 26,
        }

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



<Figure size 432x288 with 0 Axes>

In [69]:
```
agricultural_df=agric_df.corr(method="pearson")
```

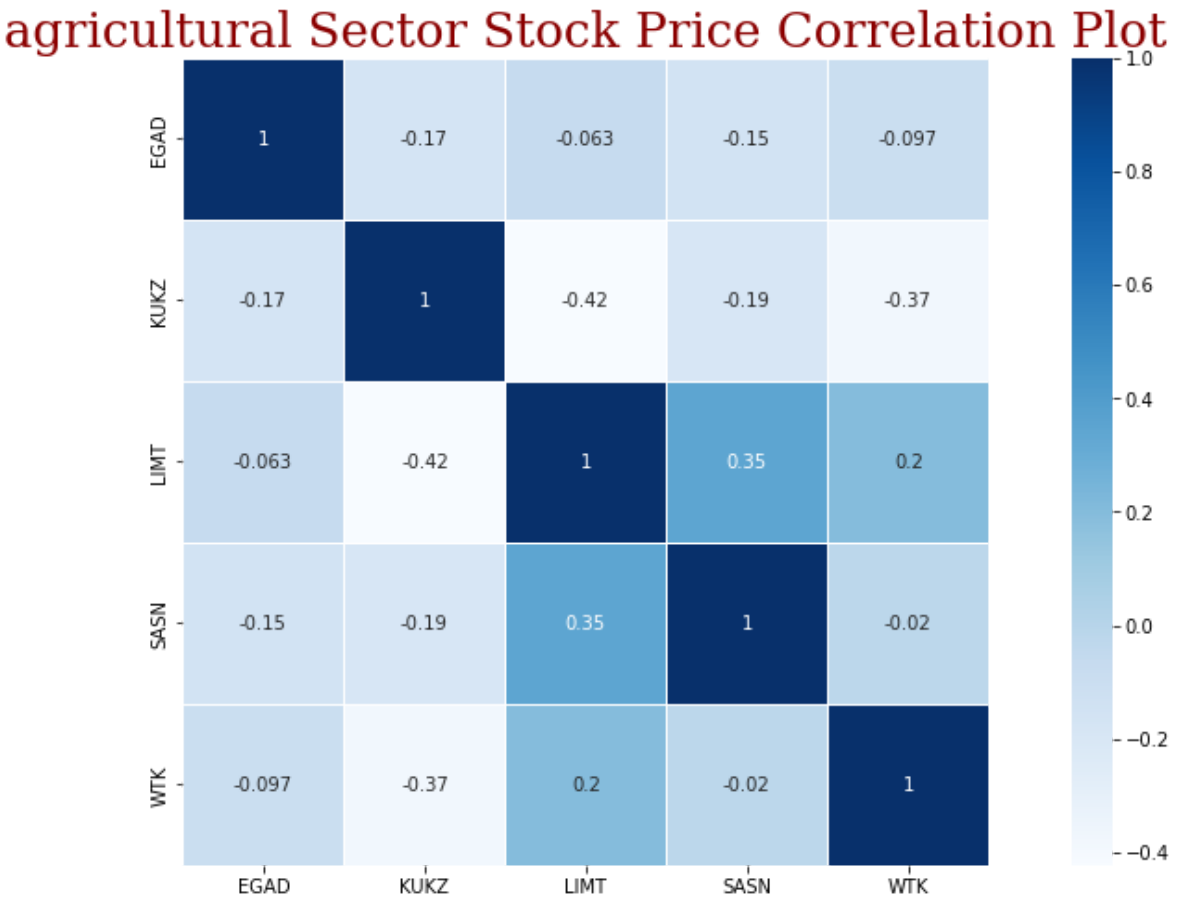
In [70]:
```
agricultural_df.head()
```

Out[70]:

	EGAD	KUKZ	LIMT	SASN	WTK
EGAD	1.000000	-0.168294	-0.063075	-0.152460	-0.097436
KUKZ	-0.168294	1.000000	-0.422992	-0.192463	-0.374117
LIMT	-0.063075	-0.422992	1.000000	0.345329	0.196757
SASN	-0.152460	-0.192463	0.345329	1.000000	-0.019554
WTK	-0.097436	-0.374117	0.196757	-0.019554	1.000000

In [71]:
```
font = {'family': 'serif',
        'color':  'darkred',
        'weight': 'normal',
        'size': 26,
        }

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



<Figure size 432x288 with 0 Axes>

In [73]:
```
investment_df=invest_df.corr(method="pearson")
```

In [74]:
```
investment_df.head()
```

Out[74]:

	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

In [75]:
```
font = {'family': 'serif',
        'color':  'darkred',
        'weight': 'normal',
        'size': 26,
        }

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

<Figure size 432x288 with 0 Axes>

In [77]:
```
manufacturing_df=manu_df.corr(method="pearson")
```

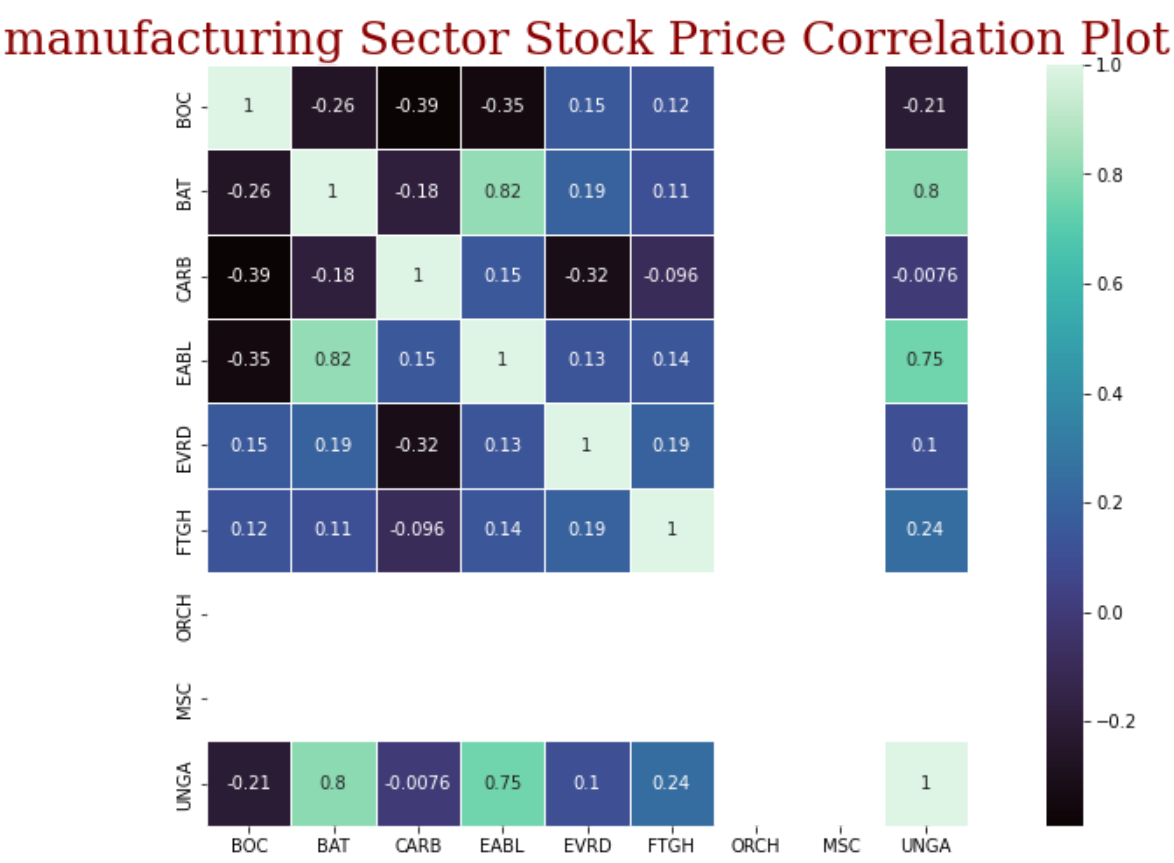
In [78]:
```
manufacturing_df.head()
```

Out[78]:

	BOC	BAT	CARB	EABL	EVRD	FTGH	ORCH	MSC	UNGA
BOC	1.000000	-0.264702	-0.390657	-0.351283	0.147241	0.124104	NaN	NaN	-0.213949
BAT	-0.264702	1.000000	-0.184494	0.816630	0.187274	0.109096	NaN	NaN	0.802586
CARB	-0.390657	-0.184494	1.000000	0.146387	-0.317406	-0.095847	NaN	NaN	-0.007650
EABL	-0.351283	0.816630	0.146387	1.000000	0.128904	0.141361	NaN	NaN	0.753815
EVRD	0.147241	0.187274	-0.317406	0.128904	1.000000	0.187224	NaN	NaN	0.101246

In [79]:
```
font = {'family': 'serif',
        'color':  'darkred',
        'weight': 'normal',
        'size': 26,
        }

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



<Figure size 432x288 with 0 Axes>

In [80]:
```
enrgy_df=energy_df.corr(method="pearson")
```

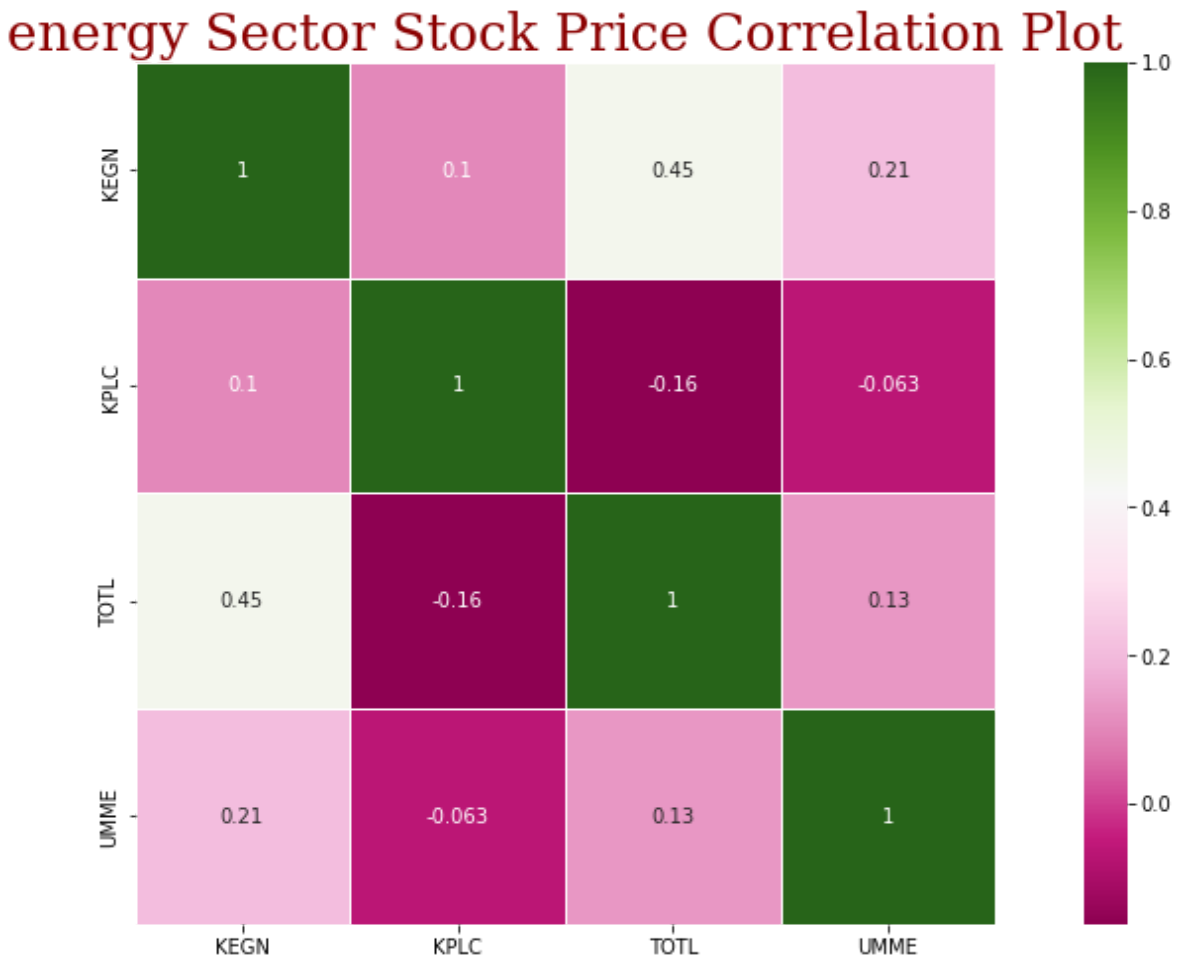
In [81]:
```
enrgy_df.head()
```

Out[81]:

	KEGN	KPLC	TOTL	UMME
KEGN	1.000000	0.104530	0.450142	0.206669
KPLC	0.104530	1.000000	-0.161972	-0.062976
TOTL	0.450142	-0.161972	1.000000	0.130735
UMME	0.206669	-0.062976	0.130735	1.000000

In [82]:
```
font = {'family': 'serif',
        'color':  'darkred',
        'weight': 'normal',
        'size': 26,
        }

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



<Figure size 432x288 with 0 Axes>

In [83]:
```
commercial_df=com_df.corr(method="pearson")
```

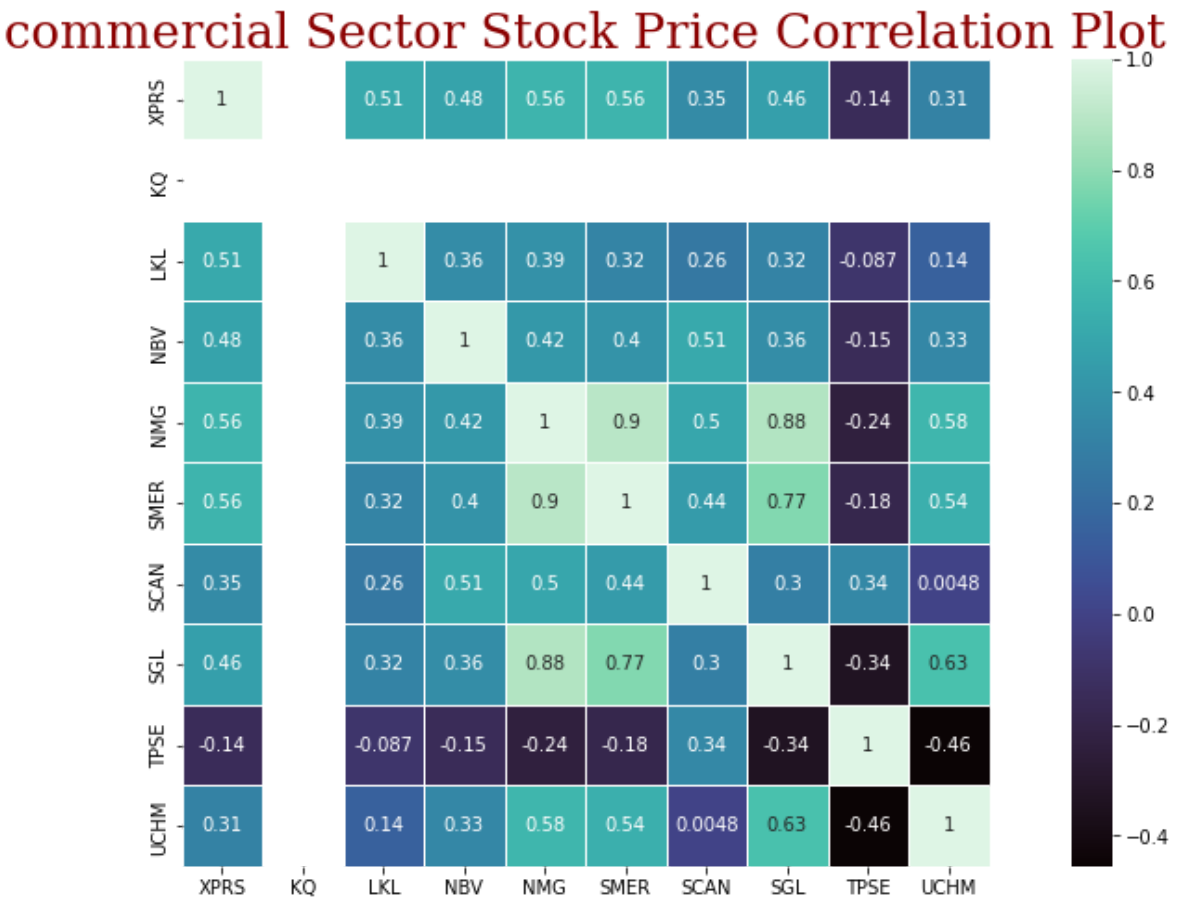
In [84]:
```
commercial_df.head()
```

Out[84]:

	XPRS	KQ	LKL	NBV	NMG	SMER	SCAN	SGL	TPSE	UC
XPRS	1.000000	NaN	0.505657	0.482448	0.564433	0.564130	0.353536	0.456735	-0.143478	0.31(
KQ	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
LKL	0.505657	NaN	1.000000	0.361295	0.387344	0.318353	0.257131	0.317583	-0.087045	0.142
NBV	0.482448	NaN	0.361295	1.000000	0.423817	0.398074	0.508718	0.363434	-0.153862	0.334
NMG	0.564433	NaN	0.387344	0.423817	1.000000	0.896570	0.497322	0.879855	-0.235356	0.575

In [85]:
```
font = {'family': 'serif',
        'color':  'darkred',
        'weight': 'normal',
        'size': 26,
        }

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



<Figure size 432x288 with 0 Axes>

In [89]:
```
construction_df=const_df.corr(method="pearson")
```

In [90]:
```
construction_df.head()
```

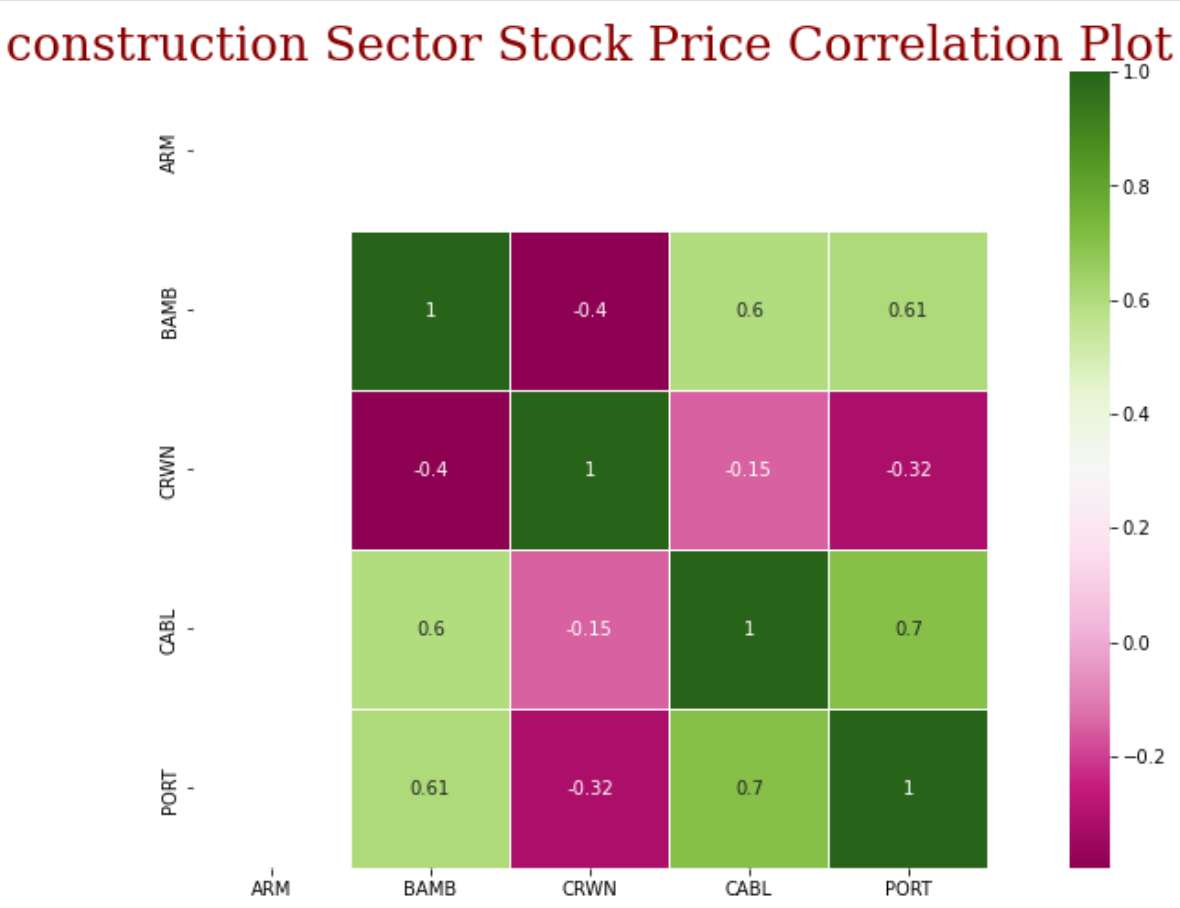
Out[90]:

	ARM	BAMB	CRWN	CABL	PORT
ARM	NaN	NaN	NaN	NaN	NaN
BAMB	NaN	1.000000	-0.395628	0.595477	0.607416
CRWN	NaN	-0.395628	1.000000	-0.147204	-0.315659
CABL	NaN	0.595477	-0.147204	1.000000	0.703981
PORT	NaN	0.607416	-0.315659	0.703981	1.000000

In [91]:

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

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



<Figure size 432x288 with 0 Axes>

In [ ]: