Python programming and data analytics.

PROJECT

Author: Dennis Omari

Bio:

Date: 2022-03-22

Questions

Section A : General

1. **Name 5 Python Modules in the Standard Library and describe what they are mainly used for.**

- **Os module**

  This module is used to interact with the operating system to get the working directory renaming of the directory, making new directories.

- **DateTime module**

  this module provides many tools for working with date and time

  allows one to get the time and date in python and perform operations on the dates and time.

- **Regular expressions module**

  RegEx is used to check if a string contains the specified search pattern

- **Comma separated values module**

  This is used for transfer of information which is structured as a table

- **Math module**
  This module allows to perform mathematical operations on numbers

2. Name 5 external modules of Python and describe the main use cases of each of these modules.

I. Seaborn

This is a data visualization module that serves as a useful Python machine learning tool for visualizing statistical models – heat maps and other types of visualizations that summarize data and depict the overall distributions.

II. Pandas

Pandas is a machine learning library in Python that provides data structures of high-level and a wide variety of tools for analysis.

It provides fast, expressive, and flexible data structures to easily work with structured and time-series data.

III. Matplotlib

Matplotlib helps with data analyzing, and is a numerical plotting library.

It helps to generate data visualizations such as two-dimensional diagrams and graphs.

IV. NumPy

Numpy is fundamentally used scientific computing in python and basic array operations.

It is useful in linear algebra and random number capabilities with broadcasting functions.

It is also used in integrating c and c++ languages.

V. Tensorflow

This module is used for machine learning and deep learning. It is used for object identification, speech recognition and many other functions.

It helps in working with artificial neural networks that need to handle multiple data sets.

Section B : Data Analysis

1. Vehicle Dataset.

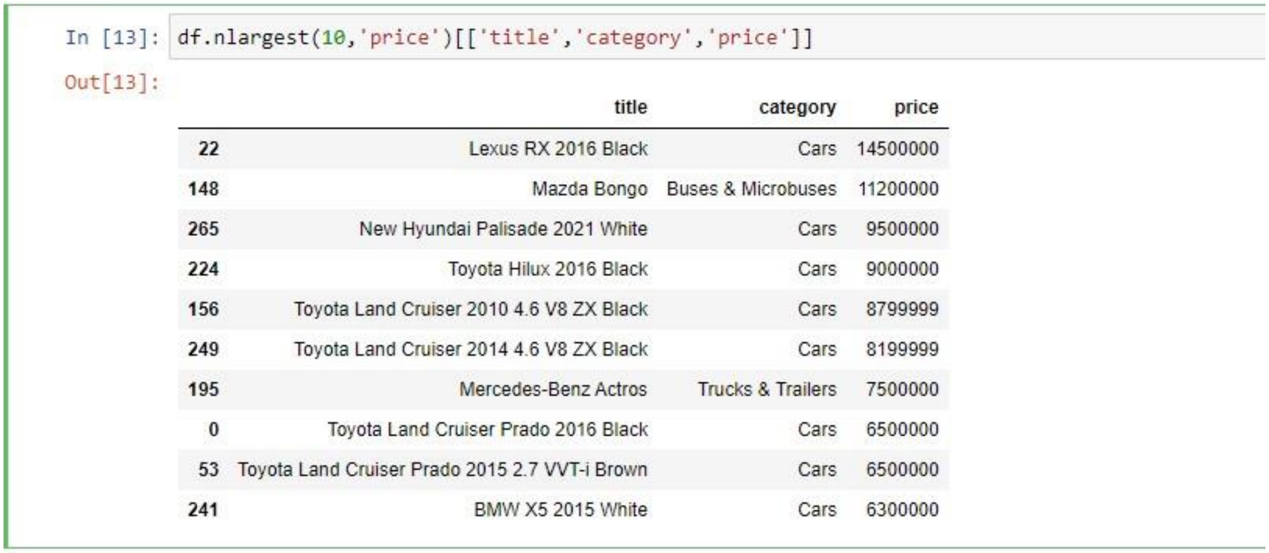
Instructions

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

Use the provided data set to answer the following:

Use pandas to come up with:

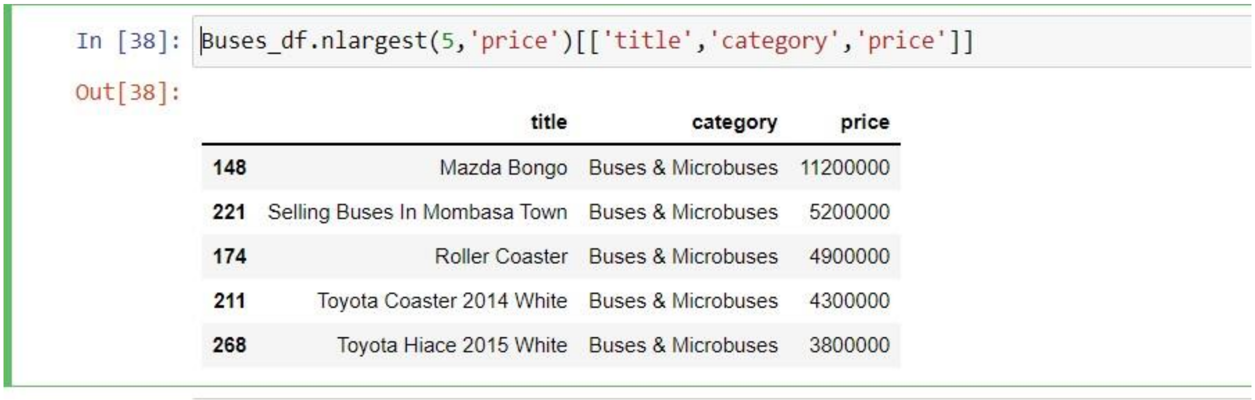
1. The titles and prices of **10** Cars with highest price



```
In [13]: df.nlargest(10,'price')[['title','category','price']]
Out[13]:
```

	title	category	price
22	Lexus RX 2016 Black	Cars	14500000
148	Mazda Bongo	Buses & Microbuses	11200000
265	New Hyundai Palisade 2021 White	Cars	9500000
224	Toyota Hilux 2016 Black	Cars	9000000
156	Toyota Land Cruiser 2010 4.6 V8 ZX Black	Cars	8799999
249	Toyota Land Cruiser 2014 4.6 V8 ZX Black	Cars	8199999
195	Mercedes-Benz Actros	Trucks & Trailers	7500000
0	Toyota Land Cruiser Prado 2016 Black	Cars	6500000
53	Toyota Land Cruiser Prado 2015 2.7 VVT-i Brown	Cars	6500000
241	BMW X5 2015 White	Cars	6300000

2. The titles and prices of 5 Buses & Microbuses with highest price

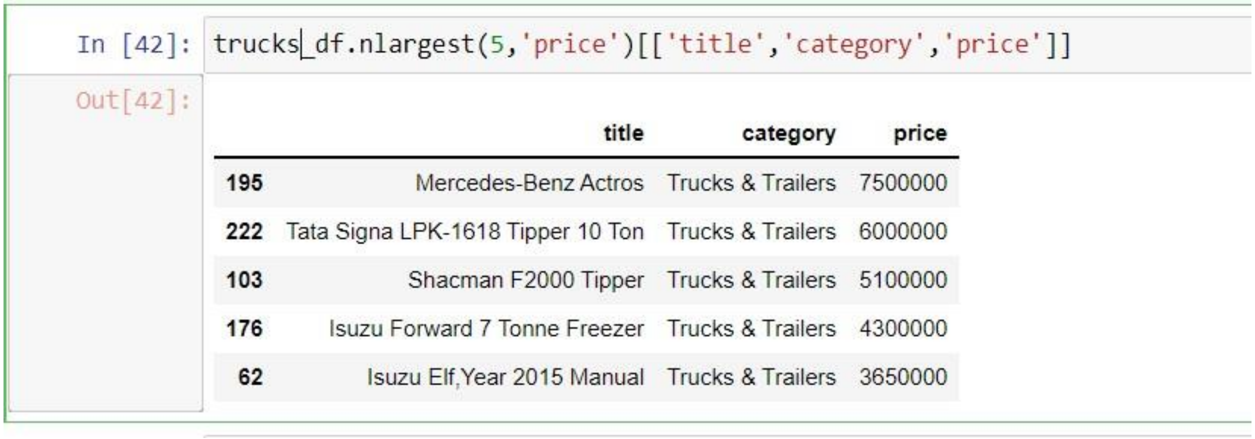


```
In [38]: Buses_df.nlargest(5,'price')[['title','category','price']]
```

Out[38]:

	title	category	price
148	Mazda Bongo	Buses & Microbuses	11200000
221	Selling Buses In Mombasa Town	Buses & Microbuses	5200000
174	Roller Coaster	Buses & Microbuses	4900000
211	Toyota Coaster 2014 White	Buses & Microbuses	4300000
268	Toyota Hiace 2015 White	Buses & Microbuses	3800000

3. The titles and prices of 5 Trucks & Trailers with highest price



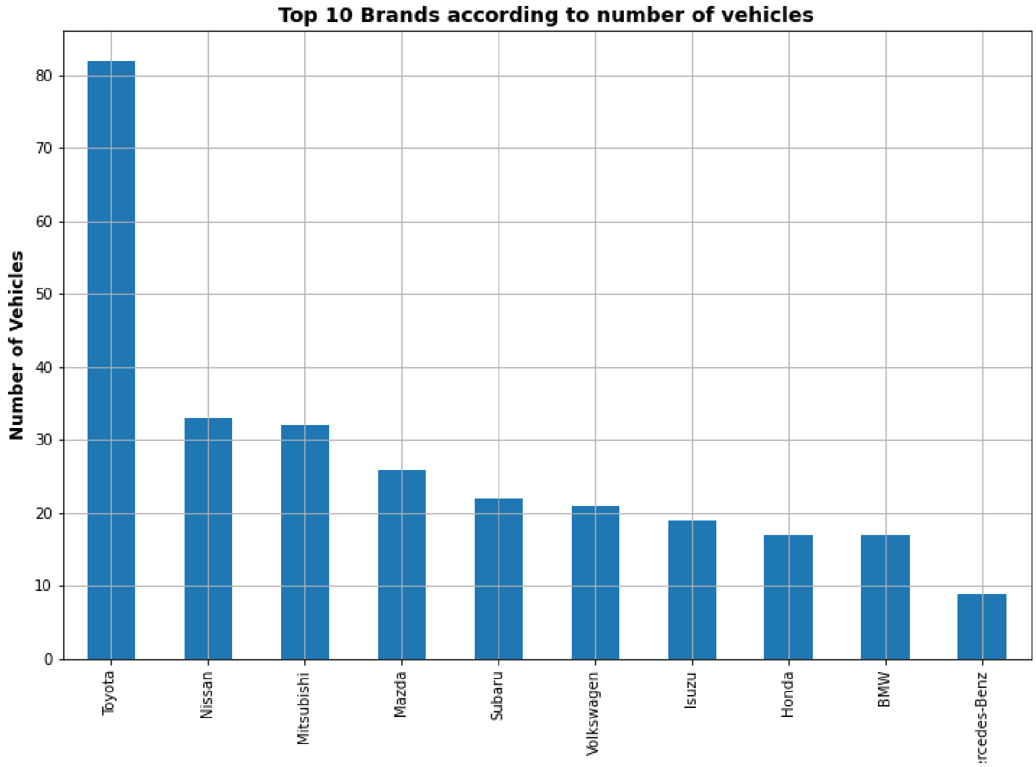
```
In [42]: trucks_df.nlargest(5,'price')[['title','category','price']]
```

Out[42]:

	title	category	price
195	Mercedes-Benz Actros	Trucks & Trailers	7500000
222	Tata Signa LPK-1618 Tipper 10 Ton	Trucks & Trailers	6000000
103	Shacman F2000 Tipper	Trucks & Trailers	5100000
176	Isuzu Forward 7 Tonne Freezer	Trucks & Trailers	4300000
62	Isuzu Elf,Year 2015 Manual	Trucks & Trailers	3650000

Plotting

Use matplotlib to come up with a plot indicating the **top 10 brands** that we have in the vehicle_dataset.

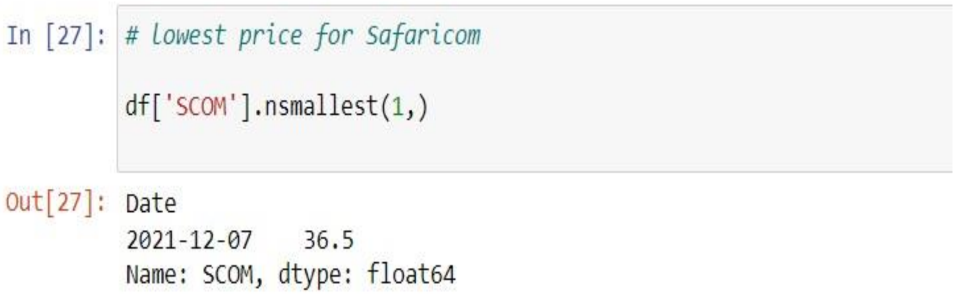


2. Time Series Data.

Instructions

Import all the libraries listed in the first cell. Make sure all modules are installed.
Use the data set provided to answer the following:

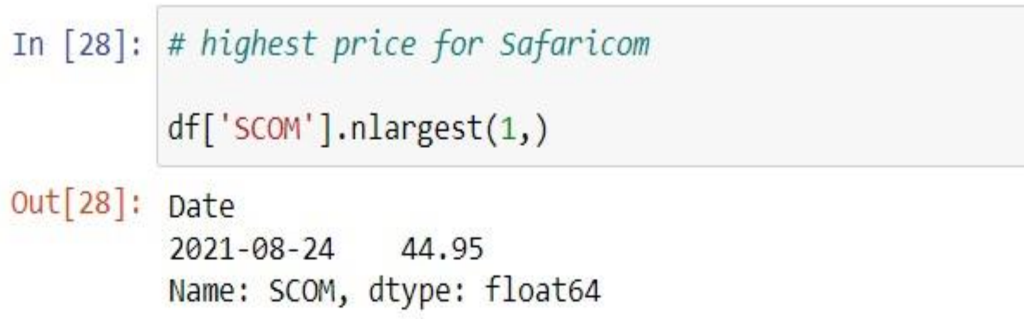
a) What is the lowest price for Safaricom (*SCOM*). b) What was the date when Safaricom had the lowest price?



```
In [27]: # Lowest price for Safaricom

         df['SCOM'].nsmallest(1,)

Out[27]: Date
         2021-12-07    36.5
         Name: SCOM, dtype: float64
```

The lowest price of SCOM was Ksh. 36.5 on 2021-12-07.

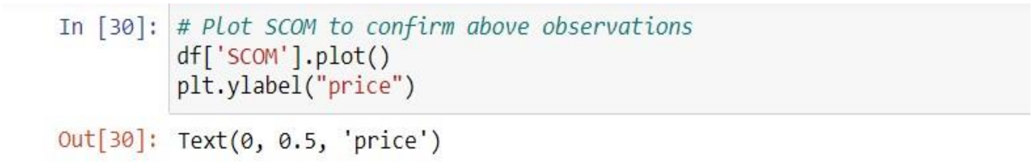
1. a) What is the highest price Safaricom stock reached in the data b) What was the date when Safaricom stock recorded the highest price?



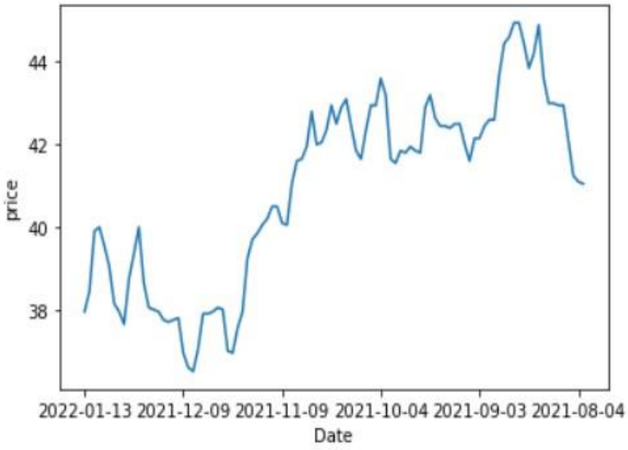
```
In [28]: # highest price for Safaricom

         df['SCOM'].nlargest(1,)

Out[28]: Date
         2021-08-24    44.95
         Name: SCOM, dtype: float64
```

The highest price of SCOM was Ksh. 44.95 on 2021-08-24.

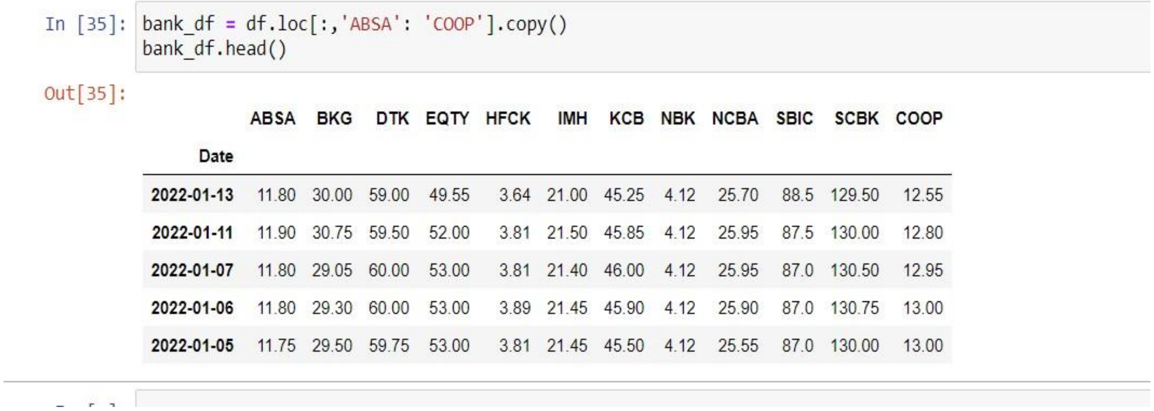
2. Create a line plot for Safaricom stock and verify if the information provided above is indeed correct.



```
In [30]: # Plot SCOM to confirm above observations
         df['SCOM'].plot()
         plt.ylabel("price")

Out[30]: Text(0, 0.5, 'price')
```



3. Select **one** of the sectors provided (agric, comm, bank, const, energy, insur, invest, manu)

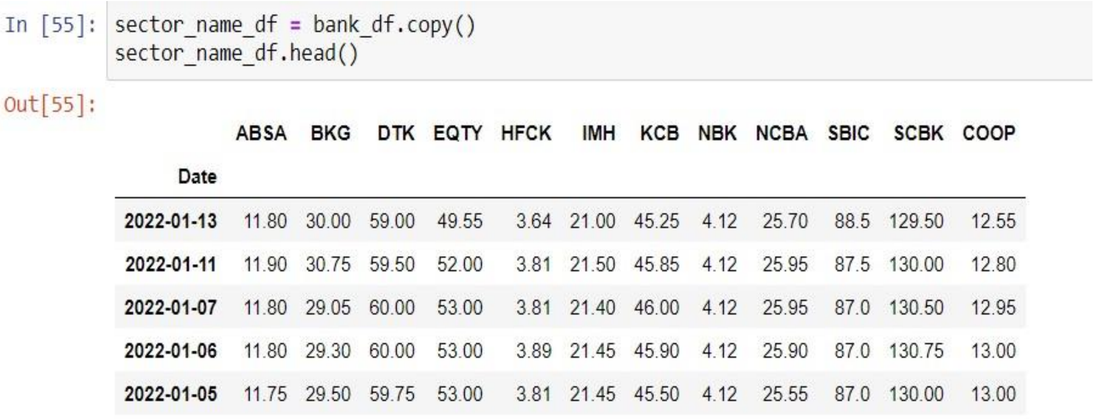


```
In [35]: bank_df = df.loc[:,'ABSA': 'COOP'].copy()
         bank_df.head()
```

Out[35]:

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

4. 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**

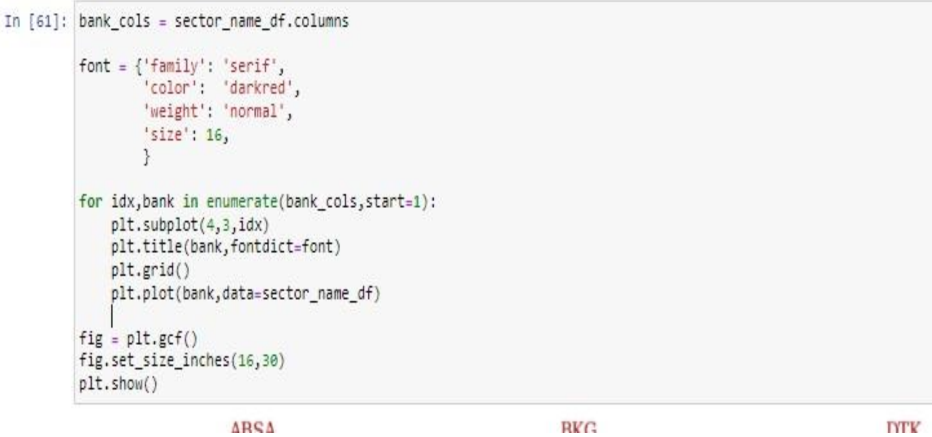


```
In [55]: sector_name_df = bank_df.copy()
         sector_name_df.head()
```

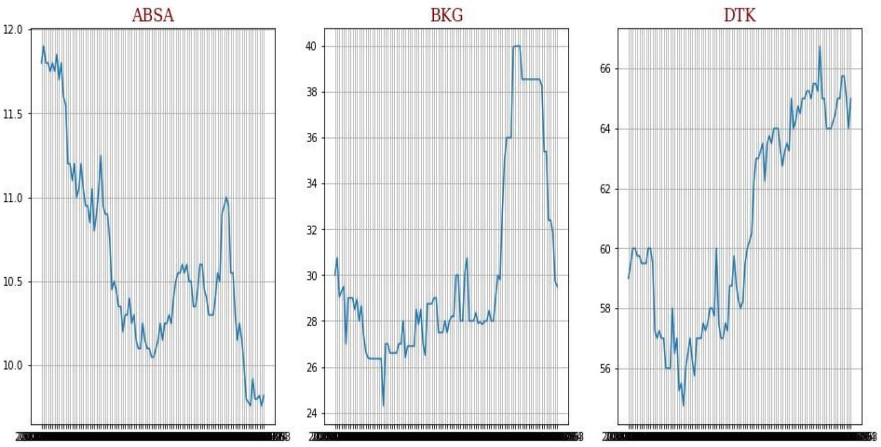
Out[55]:

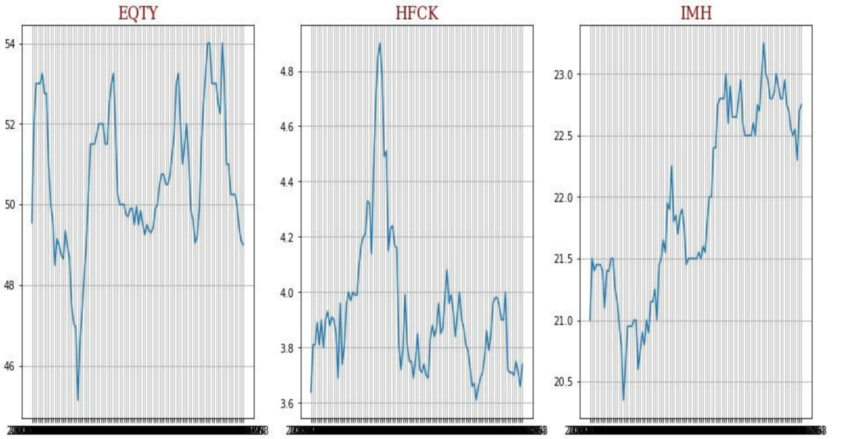
	ABSA	BKG	DTK	EQTY	HFCK	IMH	KCB	NBK	NCBA	SBIC	SCBK	COOP
Date												
2022-01-13	11.80	30.00	59.00	49.55	3.64	21.00	45.25	4.12	25.70	88.5	129.50	12.55
2022-01-11	11.90	30.75	59.50	52.00	3.81	21.50	45.85	4.12	25.95	87.5	130.00	12.80
2022-01-07	11.80	29.05	60.00	53.00	3.81	21.40	46.00	4.12	25.95	87.0	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

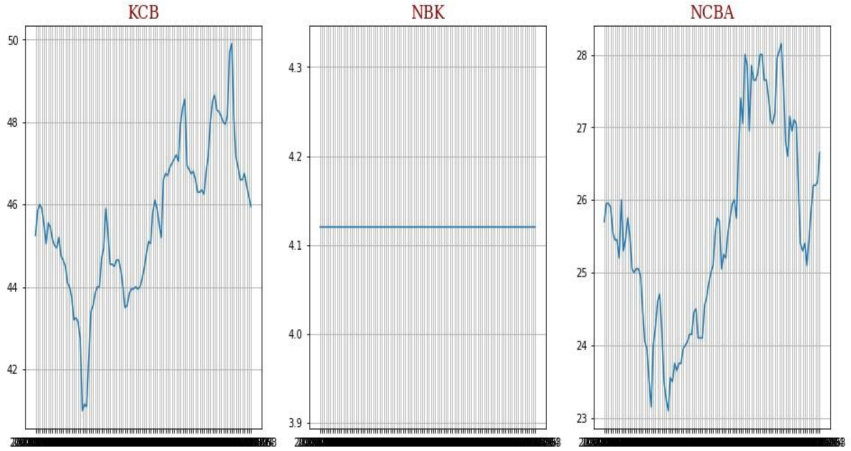
b) Using the subset for the sector, use **matplotlib** subplot to create subplots to fit all the sector stocks in one plot. One row can have a maximum of 3 charts.

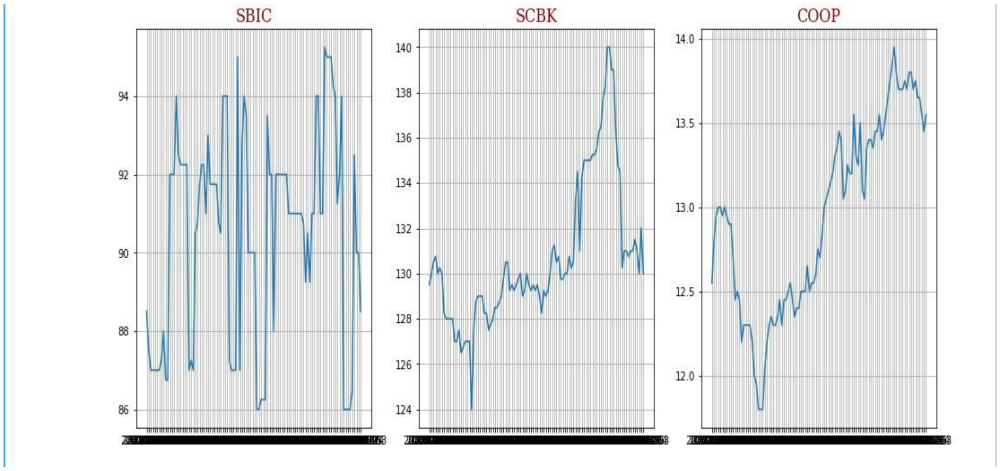


```
In [61]: bank_cols = sector_name_df.columns

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

         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(16,30)
         plt.show()
```

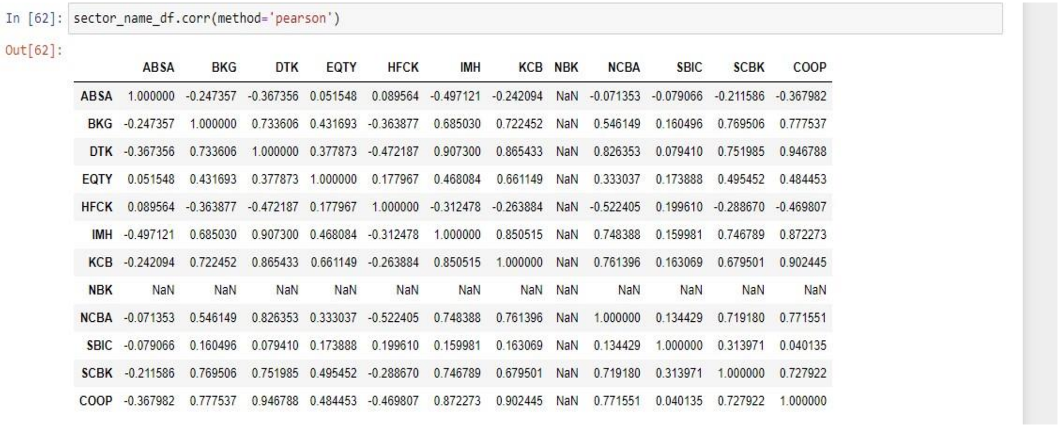

ABSA
12.0
11.5
11.0
10.5
10.0
BKG
40
38
36
34
32
30
28
26
24
DTK
66
64
62
60
58
56


EQTY
54
52
50
48
46
HFCK
4.8
4.6
4.4
4.2
4.0
3.8
3.6
IMH
23.0
22.5
22.0
21.5
21.0
20.5


KCB
50
48
46
44
42
NBK
4.3
4.2
4.1
4.0
3.9
NCBA
28
27
26
25
24
23



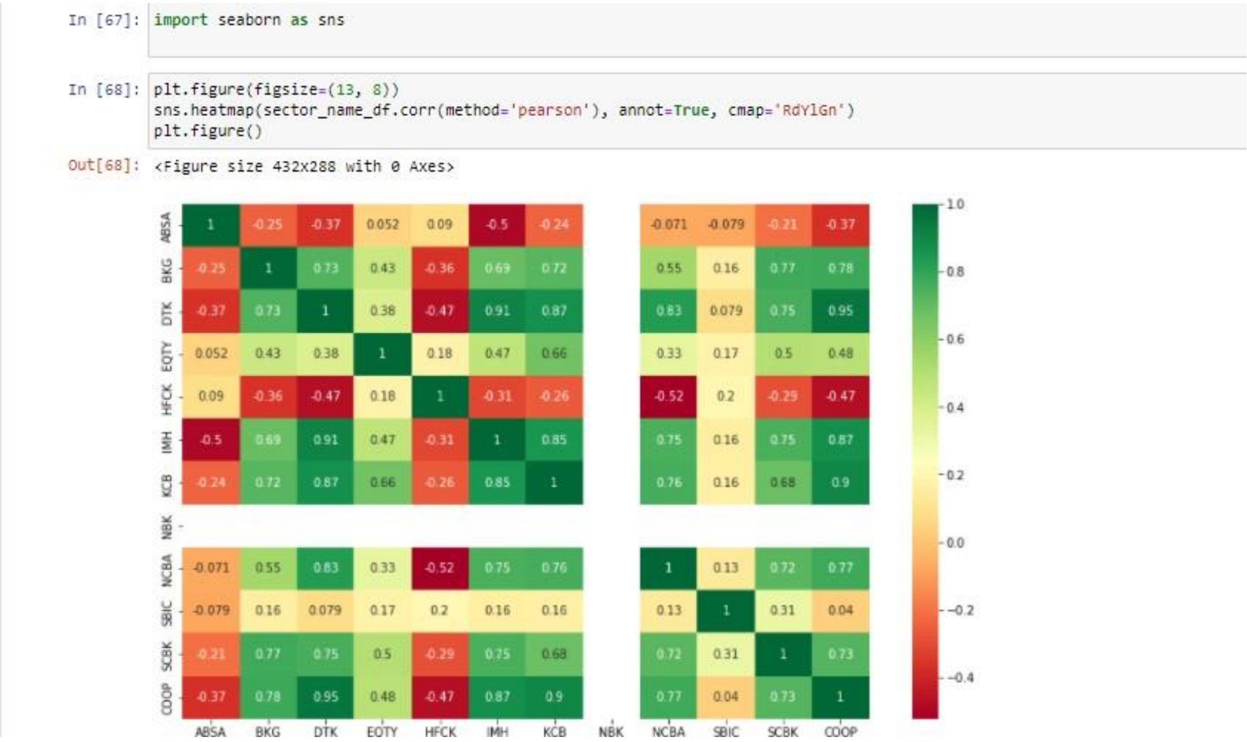
c) Using your sector DataFrame use the `corr()` DataFrame method to come up with a correlogram. Create a Data Frame for these correlations.


```
In [62]: sector_name_df.corr(method='pearson')
```

Out[62]:

	ABSA	BKG	DTK	EQTY	HFCK	IMH	KCB	NBK	NCBA	SBIC	SCBK	COOP
ABSA	1.000000	-0.247357	-0.367356	0.051548	0.089564	-0.497121	-0.242094	NaN	-0.071353	-0.079066	-0.211586	-0.367982
BKG	-0.247357	1.000000	0.733606	0.431693	-0.363877	0.685030	0.722452	NaN	0.546149	0.160496	0.769506	0.777537
DTK	-0.367356	0.733606	1.000000	0.377873	-0.472187	0.907300	0.865433	NaN	0.826353	0.079410	0.751985	0.946788
EQTY	0.051548	0.431693	0.377873	1.000000	0.177967	0.468084	0.661149	NaN	0.333037	0.173888	0.495452	0.484453
HFCK	0.089564	-0.363877	-0.472187	0.177967	1.000000	-0.312478	-0.263884	NaN	-0.522405	0.199610	-0.288670	-0.469807
IMH	-0.497121	0.685030	0.907300	0.468084	-0.312478	1.000000	0.850515	NaN	0.748388	0.159981	0.746789	0.872273
KCB	-0.242094	0.722452	0.865433	0.661149	-0.263884	0.850515	1.000000	NaN	0.761396	0.163069	0.679501	0.902445
NBK	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NCBA	-0.071353	0.546149	0.826353	0.333037	-0.522405	0.748388	0.761396	NaN	1.000000	0.134429	0.719180	0.771551
SBIC	-0.079066	0.160496	0.079410	0.173888	0.199610	0.159981	0.163069	NaN	0.134429	1.000000	0.313971	0.040135
SCBK	-0.211586	0.769506	0.751985	0.495452	-0.288670	0.746789	0.679501	NaN	0.719180	0.313971	1.000000	0.727922
COOP	-0.367982	0.777537	0.946788	0.484453	-0.469807	0.872273	0.902445	NaN	0.771551	0.040135	0.727922	1.000000

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



```
In [67]: import seaborn as sns
```

```
In [68]: plt.figure(figsize=(13, 8))
         sns.heatmap(sector_name_df.corr(method='pearson'), annot=True, cmap='RdYlGn')
         plt.figure()
```

```
Out[68]: <Figure size 432x288 with 0 Axes>
```

	ABSA	BKG	DTK	EQTY	HFCK	IMH	KCB	NBK	NCBA	SBIC	SCBK	COOP
ABSA	1	-0.25	-0.37	0.052	0.09	-0.5	-0.24		-0.071	-0.079	-0.21	-0.37
BKG	-0.25	1	0.73	0.43	-0.36	0.69	0.72		0.55	0.16	0.77	0.78
DTK	-0.37	0.73	1	0.38	-0.47	0.91	0.87		0.83	0.079	0.75	0.95
EQTY	0.052	0.43	0.38	1	0.18	0.47	0.66		0.33	0.17	0.5	0.48
HFCK	0.09	-0.36	-0.47	0.18	1	-0.31	-0.26		-0.52	0.2	-0.29	-0.47
IMH	-0.5	0.69	0.91	0.47	-0.31	1	0.85		0.75	0.16	0.75	0.87
KCB	-0.24	0.72	0.87	0.66	-0.26	0.85	1		0.76	0.16	0.68	0.9
NBK												
NCBA	-0.071	0.55	0.83	0.33	-0.52	0.75	0.76		1	0.13	0.72	0.77
SBIC	-0.079	0.16	0.079	0.17	0.2	0.16	0.16		0.13	1	0.31	0.04
SCBK	-0.21	0.77	0.75	0.5	-0.29	0.75	0.68		0.72	0.31	1	0.73
COOP	-0.37	0.78	0.95	0.48	-0.47	0.87	0.9		0.77	0.04	0.73	1