Getting Started with Python

Vehicle Analysis Project

Author : Emmanuel Eddy Bakit

Date: 20/3/2022

Resources

1. #### Vehicle Dataset

2. #### Submission Portal

If you are having problems please refer to this document:

1. #### Data Analysis with Python Pandas 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:

Use `pandas` to come up with:

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

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

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

Plotting

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

<u>Key performance Metrics</u>:

- 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
import matplotlib.pyplot as plt
```

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

Out[2]:
```
['.ipynb_checkpoints',
 'ACFrOgDNOpXraTWAiHYnIJsSZj9PkjkTe9jHRVpW_lhLt1GyJe-iUVQT_0bi4gXfb2XUhOrNac1A_1cuyp9TaPgufO_VFO1k4Dp_dcrT6SV0-3MpU8Ry7bQjP1YKOpM=.pdf',
 'Anaconda3-2021.05-Windows-x86_64.exe',
 'arduino-ide_2.0.0-rc4_Windows_64bit.exe',
 'AutoCAD_2022_English_Win_64bit_di_en-US_setup_webinstall.exe',
 'AutoCAD_2022_English_Win_64bit_dlm.sfx.exe',
 'banking_stock_prices.png',
 'BRILLIANT KCPE  SCIENCE STD 8.pdf',
 'ChromeSetup.exe',
 'Class-8-Science-End-Term-1-2020..pdf',
 'clean_stock_data.csv',
 'Click_me_to_install_SnapTube_tube_organic_search.apk',
 'Click_me_to_install_SnapTube_tube_snaptubecom.apk',
 'codeblocks-20.03mingw-setup.exe',
 'Course Syllabus.docx',
 'desktop.ini',
 'MacGyver 2016 Season 1 Complete HDTV x264 [i_c]',
 'Major Lazer - Light it Up Remix (feat. Nyla & Fuse ODG) (Music Video) by Method Studios.webm',
 'MathcadPrime1.0',
 'MathcadPrime1.0.zip',
 'MATHS-4.pdf',
 'matlab_R2021b_win64.exe',
 'musicmatters_yearmix_2019_2020_tracklist_h264_35953.mp4',
 'mysql-installer-web-community-8.0.28.0.msi',
 'new_daily_prices.csv',
 'Politics and Policy Final Project Dataset (1).xlsx',
 'Politics and Policy Final Project Dataset.xlsx',
 'project-time-series-workbook (1).ipynb',
 'project-time-series-workbook.ipynb',
 'PTC Mathcad Prime 6.0.0.0 [FileCR].zip',
 'python-3.10.2-amd64.exe',
 'R-4.1.2-win.exe',
 'rarreg.key',
 'RStudio-2021.09.2-382.exe',
 'Snaptube_v5.18.0.5183310_apkpure.com.apk',
 'SOCIAL-3.pdf',
 'SpotifySetup.exe',
 'Star Wars All Movies Collection (1977-2015) 720p Bluray x264 Dual Audio [Hindi-English] - KartiKing',
 'Statistical tables-20211109.zip',
 'Student workbook',
 'student_copy_pandas_workbook.ipynb',
 'student_workbook_stocks.ipynb',
 'Telegram Desktop',
 'tsetup.2.8.4.exe',
 'uTorrent.exe',
 'vehicle_data.csv',
 'vehicle_dataset_project (1).ipynb',
 'vehicle_dataset_project (2).ipynb',
 'vehicle_dataset_project.ipynb',
 'videoplayback (1).mp4',
 'videoplayback.mp4',
 'VSPER theory and molecular shapes.edited (1) 2.docx',
 'WhatsApp Image 2021-07-21 at 3.43.38 PM.jpeg',
 'WhatsApp Image 2021-07-21 at 3.43.39 PM.jpeg',
 'WhatsApp Image 2022-02-05 at 20.35.30.jpeg',
 'WhatsAppSetup (1).exe',
 'WhatsAppSetup.exe',
 'winrar-x64-602.exe',
 '[1337x.buzz]__Lucifer.S05.480p.x264-2MNT.torrent',
 '_Getintopc.com_PTC.Mathcad.Prime.6.0.0.0.Win64',
 '_Getintopc.com_PTC.Mathcad.Prime.6.0.0.0.Win64.rar',
 '_temp_matlab_R2021b_win64']
```

vehicle_data.csv should be listed in your output from the above cell

In [3]:
```
df = pd.read_csv('vehicle_data.csv')
df.head()
```

Out[3]:

	title	category	region	parent_region	condition	attrs	brand	color	model	yom	mileage	body_type	fuel	drive_train	trans	seat	registered	price
0	Toyota Land Cruiser Prado 2016 Black	Cars	Mvita	Mombasa	Foreign Used	First registration, No faults	Toyota	Black	Land Cruiser Prado	2016.0	87000.0	NaN	NaN	NaN	Automatic	NaN	NaN	6500000
1	Mazda Demio 2014 Brown	Cars	Langata	Nairobi	Foreign Used	First owner, No faults	Mazda	Brown	Demio	2014.0	92000.0	NaN	NaN	NaN	Automatic	NaN	Yes	970000
2	Clean NV300 Caravan 2014 Model Dielsel 16 Seater	Buses & Microbuses	Kilimani	Nairobi	Foreign Used	Nissan	Nissan	NaN	Caravan (Urvan)	2014.0	180000.0	NaN	NaN	NaN	NaN	NaN	NaN	2550000
3	Toyota Crown 2014 Pearl	Cars	Kilimani	Nairobi	Foreign Used	No faults	Toyota	Pearl	Crown	2014.0	75000.0	NaN	NaN	NaN	Automatic	NaN	No	2100000
4	Honda Fit 2014 Black	Cars	Mvita	Mombasa	Foreign Used	No faults	Honda	Black	Fit	2014.0	58000.0	NaN	NaN	NaN	Automatic	NaN	Yes	880000

In [ ]:

As an example I have shown the top 10 most expensive vehicles that are in _parent*region* Mombasa

In [4]:
```
# filter only rows that have Mombasa as their region
df['parent_region'] == 'Mombasa'
```

Out[4]:
```
0       True
1      False
2      False
3      False
4       True
```

```
      ...
295     True
296     True
297     True
298    False
299     True
Name: parent_region, Length: 300, dtype: bool
```

In [10]:
```
df['category']=='Cars'
```

Out[10]:
```
0       True
1       True
2      False
3       True
4       True
      ...
295     True
296    False
297    False
298     True
299     True
Name: category, Length: 300, dtype: bool
```

In [14]:
```
mask = df['category']=='Cars'
```

In [13]:
```
cars_df=df[mask].copy()
cars_df.head()
```

Out[13]:

	title	category	region	parent_region	condition	attrs	brand	color	model	yom	mileage	body_type	fuel	drive_train	trans	seat	registered	price
0	Toyota Land Cruiser Prado 2016 Black	Cars	Mvita	Mombasa	Foreign Used	First registration, No faults	Toyota	Black	Land Cruiser Prado	2016.0	87000.0	NaN	NaN	NaN	Automatic	NaN	NaN	6500000
1	Mazda Demio 2014 Brown	Cars	Langata	Nairobi	Foreign Used	First owner, No faults	Mazda	Brown	Demio	2014.0	92000.0	NaN	NaN	NaN	Automatic	NaN	Yes	970000
3	Toyota Crown 2014 Pearl	Cars	Kilimani	Nairobi	Foreign Used	No faults	Toyota	Pearl	Crown	2014.0	75000.0	NaN	NaN	NaN	Automatic	NaN	No	2100000
4	Honda Fit 2014 Black	Cars	Mvita	Mombasa	Foreign Used	No faults	Honda	Black	Fit	2014.0	58000.0	NaN	NaN	NaN	Automatic	NaN	Yes	880000
5	Mitsubishi Delica 2013 White	Cars	Mvita	Mombasa	Foreign Used	First registration, No faults, Unpainted	Mitsubishi	White	Delica	2013.0	88000.0	NaN	NaN	NaN	Automatic	NaN	Yes	630000

To get the highest price I will use the `nlargest` function

In [15]:
```
cars_df.nlargest(10,'price')
```

Out[15]:

	title	category	region	parent_region	condition	attrs	brand	color	model	yom	mileage	body_type	fuel	drive_train	trans	seat	registered	price
22	Lexus RX 2016 Black	Cars	Mombasa CBD	Mombasa	Foreign Used	No faults	Lexus	Black	RX	2016.0	NaN	NaN	NaN	NaN	Automatic	NaN	Yes	14500000
265	New Hyundai Palisade 2021 White	Cars	Mombasa Road	Nairobi	Brand New	No faults	Hyundai	White	Palisade	2021.0	200.0	NaN	NaN	NaN	Automatic	NaN	Yes	9500000
224	Toyota Hilux 2016 Black	Cars	Mombasa CBD	Mombasa	Foreign Used	First registration	Toyota	Black	Hilux	2016.0	36000.0	NaN	NaN	NaN	Automatic	NaN	No	9000000
156	Toyota Land Cruiser 2010 4.6 V8 ZX Black	Cars	Runda	Nairobi	Foreign Used	No faults	Toyota	Black	Land Cruiser	2010.0	NaN	SUV	Petrol	4x4	Automatic	8.0	No	8799999
249	Toyota Land Cruiser 2014 4.6 V8 ZX Black	Cars	Karen	Nairobi	Foreign Used	Unpainted, Original parts, No faults	Toyota	Black	Land Cruiser	2014.0	30000.0	SUV	Petrol	4x4	Automatic	8.0	No	8199999
0	Toyota Land Cruiser Prado 2016 Black	Cars	Mvita	Mombasa	Foreign Used	First registration, No faults	Toyota	Black	Land Cruiser Prado	2016.0	87000.0	NaN	NaN	NaN	Automatic	NaN	NaN	6500000
53	Toyota Land Cruiser Prado 2015 2.7 VVT-i Brown	Cars	Mvita	Mombasa	Foreign Used	No faults	Toyota	Brown	Land Cruiser Prado	2015.0	45260.0	SUV	Petrol	All Wheel	Automatic	5.0	No	6500000
241	BMW X5 2015 White	Cars	Mombasa CBD	Mombasa	Foreign Used	First registration	BMW	White	X5	2015.0	68000.0	SUV	Petrol	All Wheel	Automatic	NaN	No	6300000
177	Toyota Land Cruiser Prado 2014 Blue	Cars	Nairobi Central	Nairobi	Foreign Used	First owner, Unpainted, Original parts	Toyota	Blue	Land Cruiser Prado	2014.0	42000.0	NaN	NaN	NaN	Automatic	NaN	No	6150000
8	BMW X4 2015 xDrive35i Black	Cars	Mombasa CBD	Mombasa	Foreign Used	No faults	BMW	Black	X4	2015.0	63128.0	NaN	NaN	NaN	Automatic	NaN	No	5800000

To get only the `titles`

In [17]:
```
# top 10 vehicles with highest price
cars_df.nlargest(10,'price')[['title','category','price']]
```

Out[17]:

	title	category	price
22	Lexus RX 2016 Black	Cars	14500000
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
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
177	Toyota Land Cruiser Prado 2014 Blue	Cars	6150000
8	BMW X4 2015 xDrive35i Black	Cars	5800000

In [10]:
```
df['category']=='Buses & Microbuses'
```

Out[10]:
```
0      False
1      False
2       True
3      False
4      False
      ...
295    False
296    False
297    False
298    False
299    False
Name: category, Length: 300, dtype: bool
```

In [7]:
```
df['category']=='Buses & Microbuses'
```

Out[7]:
```
0      False
1      False
2       True
3      False
4      False
      ...
295    False
296    False
297    False
298    False
299    False
Name: category, Length: 300, dtype: bool
```

In [9]:
```
Bus=df['category']=='Buses & Microbuses'
Bus.head()
```

Out[9]:
```
0    False
1    False
2     True
3    False
4    False
Name: category, dtype: bool
```

In [12]:
```
buses_df=df[Bus].copy()
buses_df.head()
```

Out[12]:

	title	category	region	parent_region	condition	attrs	brand	color	model	yom	mileage	body_type	fuel	drive_train	trans	seat	registered	price
2	Clean NV300 Caravan 2014 Model Dielsel 16 Seater	Buses & Microbuses	Kilimani	Nairobi	Foreign Used	Nissan	Nissan	NaN	Caravan (Urvan)	2014.0	180000.0	NaN	NaN	NaN	NaN	NaN	NaN	2550000
25	Nissan Van	Buses & Microbuses	Machakos Town	Machakos	Kenyan Used	Nissan	Nissan	NaN	Caravan (Urvan)	2013.0	267000.0	NaN	NaN	NaN	Manual	NaN	NaN	450000
26	Nissan NV200 2015 White	Buses & Microbuses	Changamwe	Mombasa	Foreign Used	Nissan	Nissan	NaN	NV200	2015.0	112000.0	NaN	NaN	NaN	NaN	NaN	NaN	1050000
34	Toyota Hiace Box Petrol on Sale	Buses & Microbuses	Nairobi Central	Nairobi	Kenyan Used	Toyota	Toyota	Grey	HiAce	2008.0	178769.0	NaN	NaN	NaN	Automatic	NaN	NaN	960000
55	Chopper 9L Hiace	Buses & Microbuses	Tudor	Mombasa	Foreign Used	Toyota	Toyota	NaN	Grand HiAce	2015.0	101000.0	NaN	NaN	NaN	NaN	NaN	NaN	3400000

In [23]:
```
buses_df.nlargest(5,'price')
```

Out[23]:

	title	category	region	parent_region	condition	attrs	brand	color	model	yom	mileage	body_type	fuel	drive_train	trans	seat	registered	price
148	Mazda Bongo	Buses & Microbuses	Ridgeways	Nairobi	Foreign Used	Unpainted, No faults	Mazda	White	Bongo	2014.0	127800.0	NaN	NaN	NaN	Automatic	NaN	NaN	11200000
221	Selling Buses In Mombasa Town	Buses & Microbuses	Tudor	Mombasa	Foreign Used	Mitsubishi	Mitsubishi	NaN	Fuso Rosa	2013.0	101903.0	NaN	NaN	NaN	NaN	NaN	NaN	5200000
174	Roller Coaster	Buses & Microbuses	Nairobi Central	Nairobi	Foreign Used	Toyota	Toyota	NaN	Coaster	2012.0	88000.0	NaN	NaN	NaN	NaN	NaN	NaN	4900000
211	Toyota Coaster 2014 White	Buses & Microbuses	Mombasa CBD	Mombasa	Foreign Used	Toyota	Toyota	NaN	Coaster	2014.0	189000.0	NaN	NaN	NaN	NaN	NaN	NaN	4300000
268	Toyota Hiace 2015 White	Buses & Microbuses	Ziwa la Ngombe	Nyali	Brand New	Toyota	Toyota	NaN	HiAce	2015.0	85472.0	NaN	NaN	NaN	NaN	NaN	NaN	3800000

In [24]:
```
buses_df.nlargest(5,'price')[['title','category','price']]
```

Out[24]:

	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

In [26]:
```
truck = df['category']=='Trucks & Trailers'
truck.head()
```

Out[26]:
```
0    False
1    False
2    False
3    False
4    False
Name: category, dtype: bool
```

In [27]:
```
truck_df=df[truck].copy()
truck_df.head()
```

Out[27]:

	title	category	region	parent_region	condition	attrs	brand	color	model	yom	mileage	body_type	fuel	drive_train	trans	seat	registered	price
14	Very Clean Isuzu FRR Truck 2015 Model	Trucks & Trailers	Thome	Nairobi	Used	Used	Isuzu	White	F SERIES	2015.0	NaN	NaN	Diesel	NaN	Manual	NaN	NaN	3520000
20	Truck Lorry	Trucks & Trailers	Shanzu	Mombasa	Used	Used	Ashok Leyland	NaN	1920	2014.0	NaN	NaN	Diesel	NaN	NaN	NaN	NaN	1600000
27	Volvo NL12 for Sale	Trucks & Trailers	Embakasi	Nairobi	Used	Used	Volvo	NaN	NaN	1994.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1200000
30	Mitsubishi Fuso Canter	Trucks & Trailers	Mombasa CBD	Mombasa	Used	Used	Mitsubishi	NaN	NaN	2014.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2600000
36	Isuzu Elf,Year 2014manual Transmission	Trucks & Trailers	Mombasa CBD	Mombasa	Brand New	Brand New	Isuzu	NaN	NaN	2013.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2150000

In [29]:
```
truck_df.nlargest(5,'price')
```

Out[29]:

	title	category	region	parent_region	condition	attrs	brand	color	model	yom	mileage	body_type	fuel	drive_train	trans	seat	registered	price
195	Mercedes-Benz Actros	Trucks & Trailers	Thome	Nairobi	Used	Used	Mercedes-Benz	NaN	NaN	2013.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	7500000
222	Tata Signa LPK-1618 Tipper 10 Ton	Trucks & Trailers	Nairobi Central	Nairobi	Brand New	Brand New	Tata	NaN	M&HCV Rigid Truck	2021.0	NaN	NaN	Diesel	NaN	NaN	NaN	NaN	6000000
103	Shacman F2000 Tipper	Trucks & Trailers	Municipality	Meru	Used	Used	Shacman	NaN	NaN	2018.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5100000
176	Isuzu Forward 7 Tonne Freezer	Trucks & Trailers	Tudor	Mombasa	Used	Used	Isuzu	NaN	F SERIES	2015.0	NaN	NaN	Diesel	NaN	NaN	NaN	NaN	4300000
62	Isuzu Elf,Year 2015 Manual	Trucks & Trailers	Mombasa CBD	Mombasa	Brand New	Brand New	Isuzu	NaN	NaN	2014.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	3650000

In [31]:
```
truck_df.nlargest(5,'price')[['title','category','price']]
```

Out[31]:

	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

In [ ]:

The above output is what the question is asking for. So take a screenshot.

In [ ]:

Plotting

I will demonstrate how to solve the plotting challenge using the following question:

Use **matplotlib** to come up with a plot indicating the **top 5 regions** that we have in the vehicle_dataset

In [14]:
```
# get number of rows with same region
df['region'].value_counts()
```

Out[14]:
```
Mombasa CBD                     92
Mvita                           28
Nairobi Central                 27
Kilimani                        23
Lavington                       16
Ridgeways                       15
Tudor                           13
Karen                            8
Langata                          7
Nairobi                          6
Kileleshwa                       4
Ganjoni                          4
Kasarani                         3
Nyali                            3
Embakasi                         3
Thome                            3
Kisauni                          3
Parklands/Highridge              3
Donholm                          3
Mombasa Road                     2
Nakuru Town West                 2
Westlands                        2
Runda                            2
Kikuyu                           2
Ruiru                            2
Mbaraki                          2
Nakuru Town East                 2
Makadara                         1
Ziwa la Ngombe                   1
Shanzu                           1
Imara Daima                      1
Majengo                          1
Kiambaa                          1
Ngara                            1
Ngong                            1
Kiambu / Kiambu                  1
Machakos Town                    1
Woodley/Kenyatta Golf Course     1
Thika                            1
Changamwe                        1
Kisumu Central                   1
Ruai                             1
Eldoret CBD                      1
Syokimau                         1
Kerugoya                         1
Municipality                     1
Roysambu                         1
Name: region, dtype: int64
```

In [16]:
```
# grab the top 5
df['region'].value_counts()[:5]
```

Out[16]:
```
Mombasa CBD        92
Mvita              28
Nairobi Central    27
Kilimani           23
Lavington          16
Name: region, dtype: int64
```

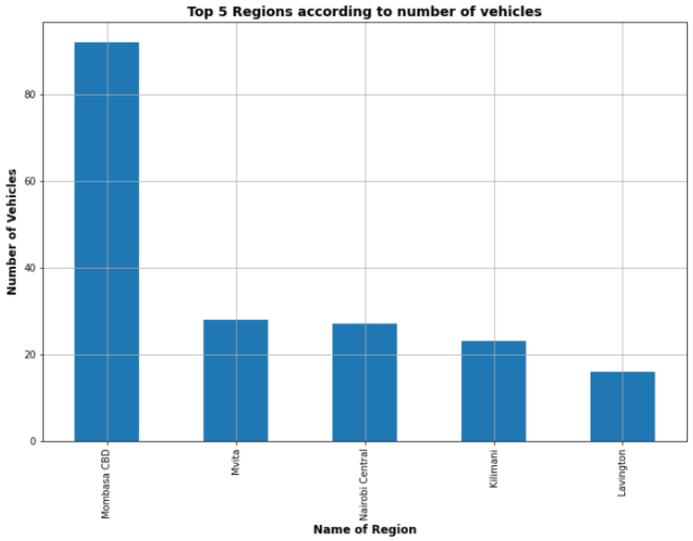
In [17]:
```
# make it a variable
top_5 = df['region'].value_counts()[:5]
```

Now to create a bar plot of the top 5 regions

In [23]:
```
plt.figure(figsize=(12,8))
plt.title("Top 5 Regions according to number of vehicles", fontsize=14, fontweight='bold')
top_5.plot.bar()
plt.xlabel('Name of Region',fontsize=12, fontweight='bold')
plt.ylabel('Number of Vehicles',fontsize=12, fontweight='bold')
plt.grid()

# save the plot to file
fig = plt.gcf()
fig.savefig('top-5-regions.png')

plt.show()
```



In [32]:
```
df['brand'].value_counts()
```

Out[32]:
```
Toyota           82
Nissan           33
Mitsubishi       32
Mazda            26
Subaru           22
Volkswagen       21
Isuzu            19
Honda            17
BMW              17
Mercedes-Benz     9
Suzuki            5
Lexus             4
Tata              2
Volvo             2
Ashok Leyland     2
Land Rover        1
Shacman           1
Other             1
Hyundai           1
Daihatsu          1
Audi              1
Name: brand, dtype: int64
```

In [34]:
```
df['brand'].value_counts()[:10]
```

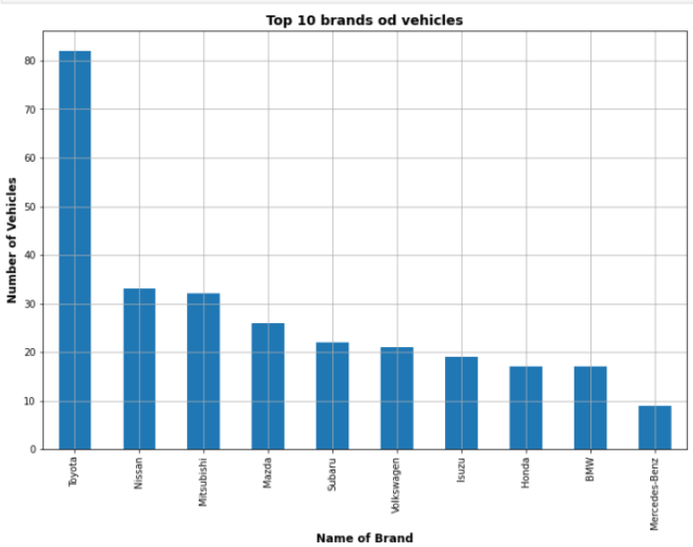
Out[34]:
```
Toyota           82
Nissan           33
Mitsubishi       32
Mazda            26
Subaru           22
Volkswagen       21
Isuzu            19
Honda            17
BMW              17
Mercedes-Benz     9
Name: brand, dtype: int64
```

In [35]:
```
top_10=df['brand'].value_counts()[:10]
```

In [36]:
```
plt.figure(figsize=(12,8))
plt.title("Top 10 brands od vehicles", fontsize=14, fontweight='bold')
top_10.plot.bar()
plt.xlabel('Name of Brand',fontsize=12, fontweight='bold')
plt.ylabel('Number of Vehicles',fontsize=12, fontweight='bold')
plt.grid()

# save the plot to file
fig = plt.gcf()
fig.savefig('top-10-brands.png')

plt.show()
```



In [ ]:

Getting Started with Python

Time Series Project

Author : Emmanuel Eddy Bakit

Date: 17/3/2022

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.

<u>Key performance Metrics</u>:

- 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 [8]:
```
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 [12]:
```
os.listdir()
```

Out[12]:
```
['.ipynb_checkpoints',
 'ACFrOgDNOpXraTWAiHYnIJsSZj9PkjkTe9jHRVpW_lhLtlGyJe-iUVQT_0bi4gXfb2XUhOrNaclA_1cuyp9TaPgufO_VFOlk4Dp_dcrT6SV0-3MpU8Ry7bQjPlYKOpM=.pdf',
 'Anaconda3-2021.05-Windows-x86_64.exe',
 'arduino-ide_2.0.0-rc4_Windows_64bit.exe',
 'AutoCAD_2022_English_Win_64bit_di_en-US_setup_webinstall.exe',
 'AutoCAD_2022_English_Win_64bit_dlm.sfx.exe',
 'banking_stock_prices.png',
 'BRILLIANT KCPE  SCIENCE STD 8.pdf',
 'ChromeSetup.exe',
 'Class-8-Science-End-Term-1-2020..pdf',
 'clean_stock_data.csv',
 'clean_stock_prices.csv',
 'Click_me_to_install_SnapTube_tube_organic_search.apk',
 'Click_me_to_install_SnapTube_tube_snaptubecom.apk',
 'codeblocks-20.03mingw-setup.exe',
 'Course Syllabus.docx',
 'desktop.ini',
 'MacGyver 2016 Season 1 Complete HDTV x264 [i_c]',
 'Major Lazer - Light it Up Remix (feat. Nyla & Fuse ODG) (Music Video) by Method Studios.webm',
 'MathcadPrime1.0',
 'MathcadPrime1.0.zip',
 'MATHS-4.pdf',
 'matlab_R2021b_win64.exe',
 'musicmatters_yearmix_2019_2020_tracklist_h264_35953.mp4',
 'mysql-installer-web-community-8.0.28.0.msi',
 'new_daily_prices.csv',
 'Politics and Policy Final Project Dataset (1).xlsx',
 'Politics and Policy Final Project Dataset.xlsx',
 'project-time-series-workbook (1).ipynb',
 'project-time-series-workbook (2).ipynb',
 'project-time-series-workbook.ipynb',
 'PTC Mathcad Prime 6.0.0.0 [FileCR].zip',
 'python-3.10.2-amd64.exe',
 'R-4.1.2-win.exe',
 'rarreg.key',
 'RStudio-2021.09.2-382.exe',
 'Snaptube_v5.18.0.5183310_apkpure.com.apk',
 'SOCIAL-3.pdf',
 'SpotifySetup.exe',
 'Star Wars All Movies Collection (1977-2015) 720p Bluray x264 Dual Audio [Hindi-English] - KartiKing',
 'Statistical tables-20211109.zip',
 'Student workbook',
 'student_copy_pandas_workbook.ipynb',
 'student_workbook_stocks.ipynb',
 'Telegram Desktop',
 'top-10-brands.png',
 'tsetup.2.8.4.exe',
 'uTorrent.exe',
 'vehicle_data (1).csv',
 'vehicle_data (2).csv',
 'vehicle_data (3).csv',
 'vehicle_data.csv',
 'vehicle_dataset_project (1).ipynb',
 'vehicle_dataset_project (2).ipynb',
 'vehicle_dataset_project.ipynb',
 'videoplayback (1).mp4',
 'videoplayback.mp4',
 'VSPER theory and molecular shapes.edited (1) 2.docx',
 'WhatsApp Image 2021-07-21 at 3.43.38 PM.jpeg',
 'WhatsApp Image 2021-07-21 at 3.43.39 PM.jpeg',
 'WhatsApp Image 2022-02-05 at 20.35.30.jpeg',
 'WhatsAppSetup (1).exe',
 'WhatsAppSetup.exe',
 'winrar-x64-602.exe',
 '[1337x.buzz]__Lucifer.S05.480p.x264-ZMNT.torrent',
 '_Getintopc.com_PTC.Mathcad.Prime.6.0.0.0.Win64',
 '_Getintopc.com_PTC.Mathcad.Prime.6.0.0.0.Win64.rar',
 '_temp_matlab_R2021b_win64']
```

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 [13]:
```
# read in the necessary file ('clean_stock_prices.csv')
df = pd.read_csv('clean_stock_prices.csv',index_col=0)
df.head()
```

Out[13]:

Date	EGAD	KUKZ	LIMT	SASN	WTK	CGEN	ABSA	BKG	DTK	EQTY	...	BAT	CARB	EABL	EVRD	FTGH	ORCH	MSC	UNGA	SCOM	FAHR
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	1.34	10.4	0.27	27.10	37.95	6.52
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	1.31	10.4	0.27	27.65	38.45	6.56
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	1.30	10.4	0.27	27.65	39.90	6.38
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	1.29	10.4	0.27	27.65	40.00	6.40
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	1.26	10.4	0.27	27.65	39.55	6.02

5 rows × 60 columns

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

Out[14]:

	EGAD	KUKZ	LIMT	SASN	WTK	CGEN	ABSA	BKG	DTK	EQTY	...	BAT	CARB	EABL	EVRD	FTGH	ORCH	MSC	UNGA	SCOM	FAHR
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	12.25	179.25	0.96	1.32	10.4	0.27	31.0	42.95	6.78
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	1.32	10.4	0.27	31.0	42.10	6.98
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	1.31	10.4	0.27	31.0	41.25	6.80
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	1.30	10.4	0.27	31.0	41.10	6.92
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	1.31	10.4	0.27	28.3	41.05	6.94

5 rows × 60 columns

Use this part to answer questions 1, 2 and 3

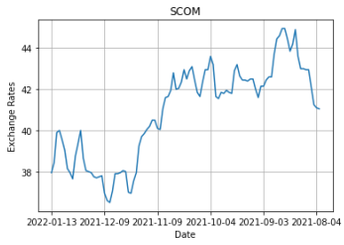
In [15]:
```
# lowest price for Safaricom
df['SCOM'].nsmallest(1)
```

Out[15]:
```
Date
2021-12-07    36.5
Name: SCOM, dtype: float64
```

In [16]:
```
# highest price for Safaricom
df['SCOM'].nlargest(1)
```

Out[16]:
```
Date
2021-08-24    44.95
Name: SCOM, dtype: float64
```

In [19]:
```
# Plot SCOM to confirm above observations
df['SCOM'].plot()
plt.grid()
plt.xlabel('Date')
plt.ylabel('Exchange Rates')
plt.title('SCOM')
plt.show()
```



Use this part to answer question 4

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','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 [22]:
```
insur_df = df.loc[:,'BRIT':'SLAM'].copy()
insur_df.head()
```

Out[22]:

	BRIT	CIC	JUB	KNRE	LBTY	SLAM
Date						
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 [23]:
```
insur_cols = insur_df.columns
for insurance in insur_cols:
    print(insurance)
```

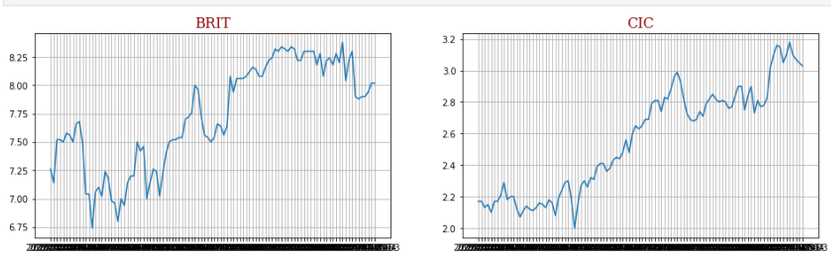
```
BRIT
CIC
JUB
KNRE
LBTY
SLAM
```

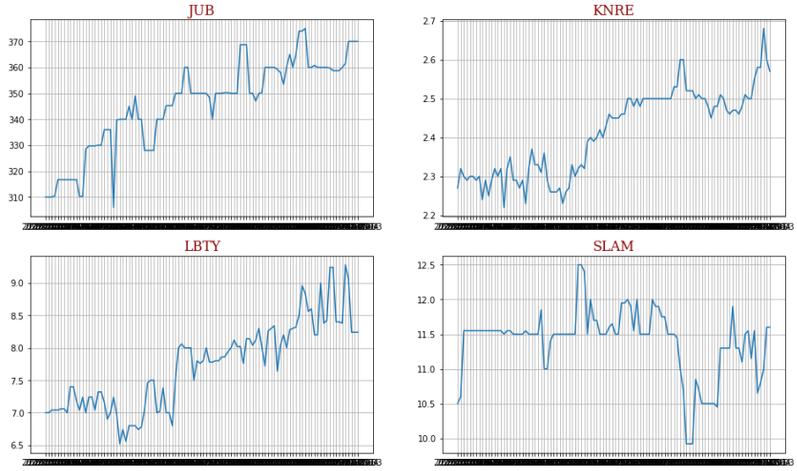
In [25]:
```
insur_cols = insur_df.columns

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

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

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





In [26]:
```
agric_df=df.loc[:,'EGAD':'WTK'].copy()
agric_df.head()
```

Out[26]:

	EGAD	KUKZ	LIMT	SASN	WTK
Date					
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 [27]:
```
agric_cols=agric_df.columns
for agriculture in agric_cols:
    print(agriculture)
```

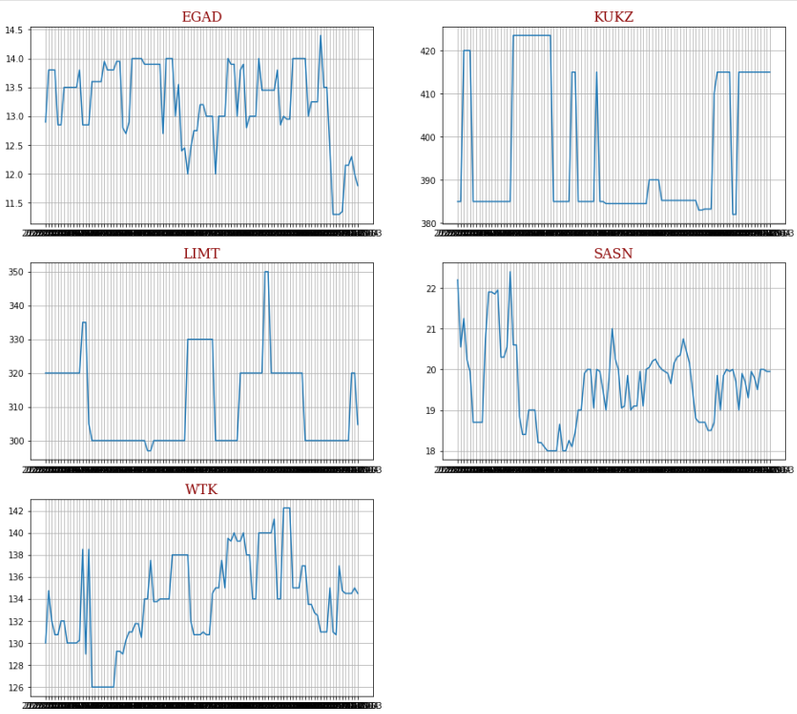
```
EGAD
KUKZ
LIMT
SASN
WTK
```

In [28]:
```
agric_cols = agric_df.columns

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

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

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



In [29]:
```
comm_df=df.loc[:,'XPRS':'UCHM'].copy()
comm_df.head()
```

Out[29]:

	XPRS	KQ	LKL	NBV	NMG	SMER	SCAN	SGL	TPSE	UCHM
Date										
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 [37]:
```
comm_cols=comm_df.columns
for commercial in comm_cols:
    print(commercial)
```

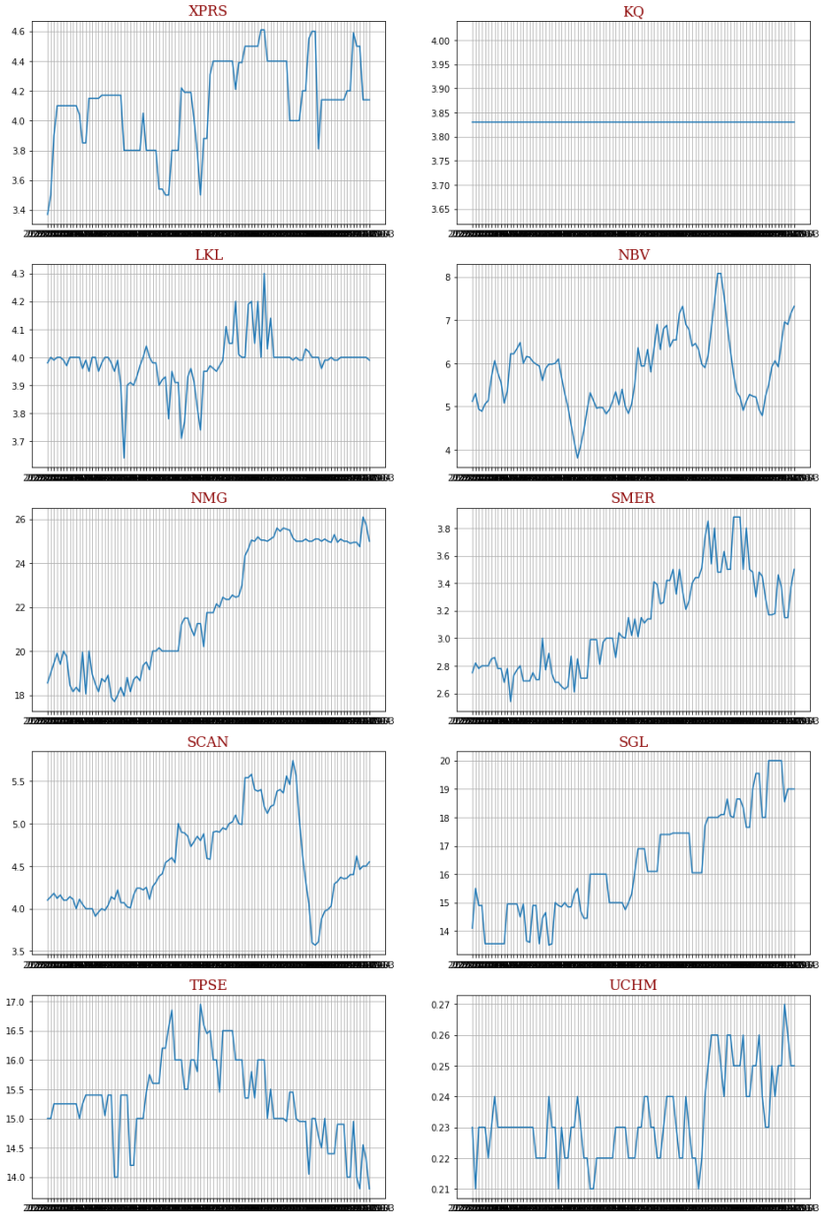
```
XPRS
KQ
LKL
NBV
NMG
SMER
SCAN
SGL
TPSE
UCHM
```

In [38]:
```
comm_cols = comm_df.columns
```

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

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

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



In [33]:
```
bank_df=df.loc[:,'ABSA':'COOP'].copy()
bank_df.head()
```

Out[33]:

	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

In [35]:
```
bank_cols=bank_df.columns
for banking in bank_cols:
    print(banking)
```

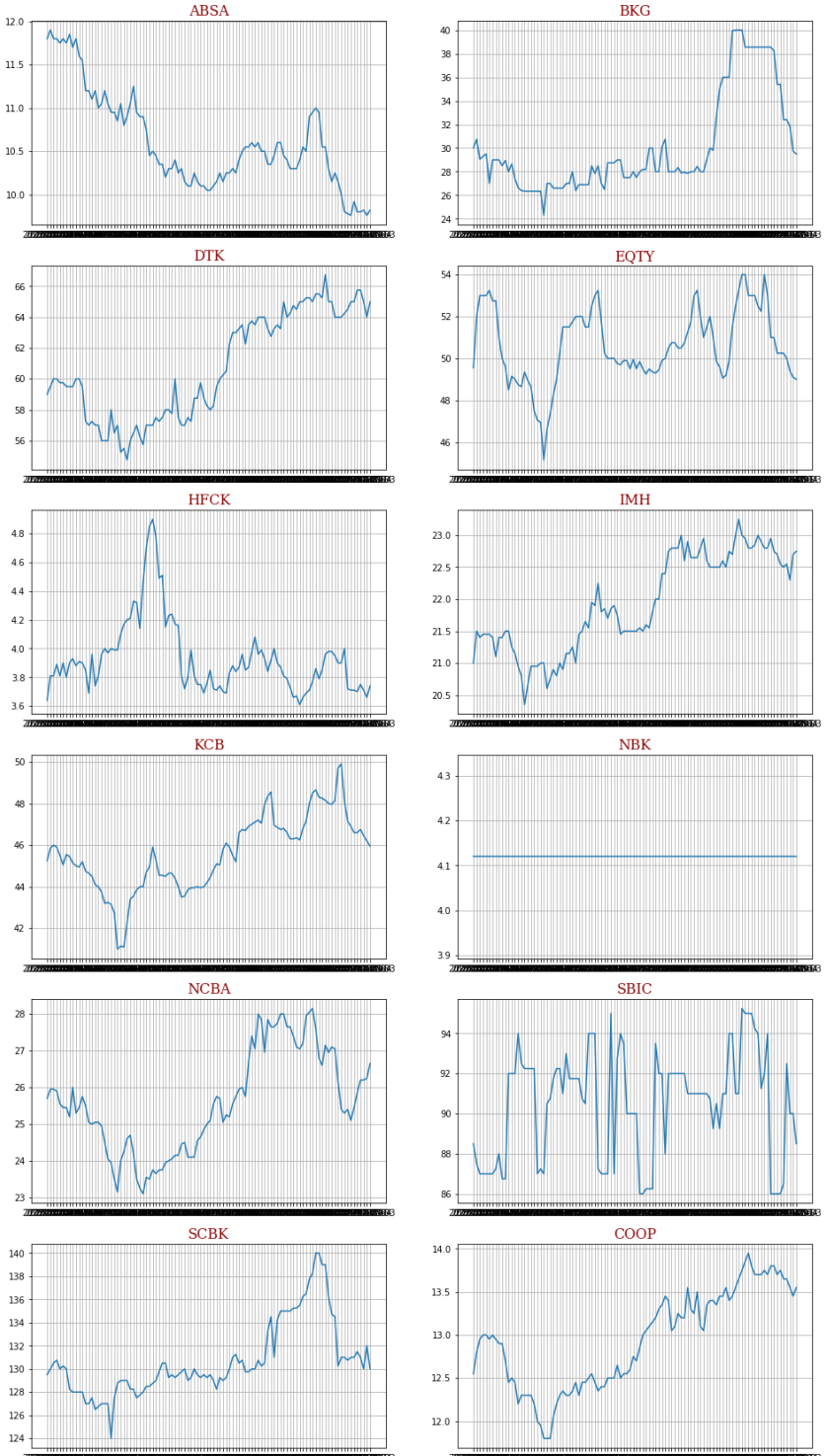
```
ABSA
BKG
DTK
EQTY
HFCK
IMH
KCB
NBK
NCBA
SBIC
SCBK
COOP
```

In [36]:
```
bank_cols = bank_df.columns

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

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

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



In [39]:
```
const_df=df.loc[:,'ARM':'PORT'].copy()
const_df.head()
```

Out[39]:

	ARM	BAMB	CRWN	CABL	PORT
Date					
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 [40]:
```
const_cols=const_df.columns
for construction in const_cols:
    print(construction)
```

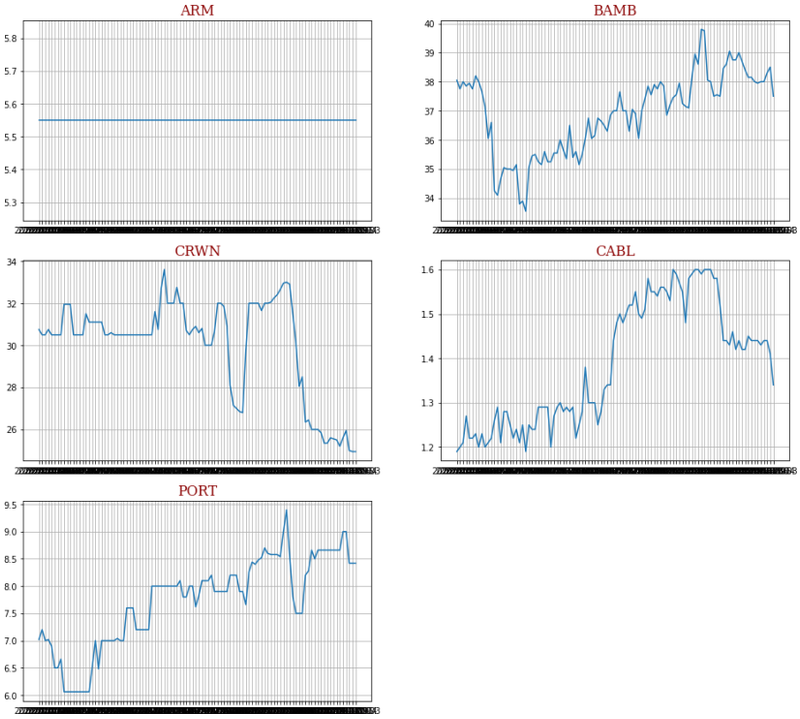
```
ARM
BAMB
CRWN
CABL
PORT
```

In [41]:
```
const_cols = const_df.columns

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

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

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



In [42]:
```
energ_df=df.loc[:,'KEGN':'UMME']
energ_df.head()
```

Out[42]:

	KEGN	KPLC	TOTL	UMME
Date				
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 [43]:
```
energ_cols=energ_df.columns
for energy in energ_cols:
    print(energy)
```

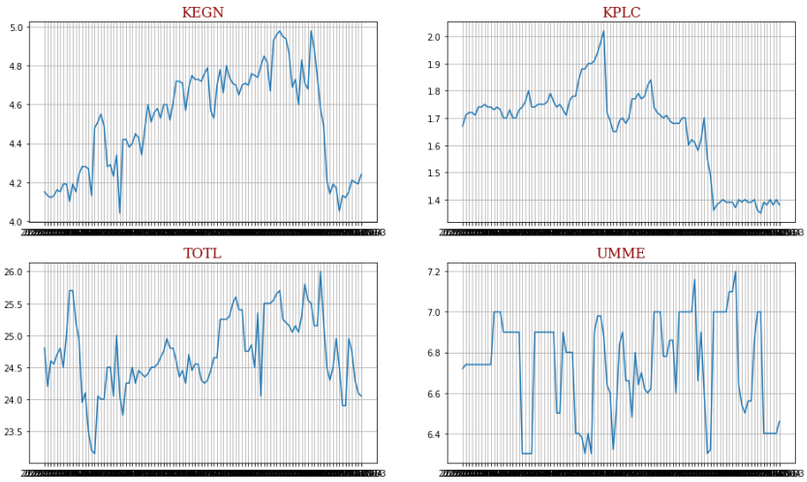
```
KEGN
KPLC
TOTL
UMME
```

In [44]:
```
energ_cols = energ_df.columns

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

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

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



In [45]:
```
invest_df=df.loc[:,'CTUM':'NSE'].copy()
invest_df.head()
```

Out[45]:

	CTUM	HAFR	KURV	OCH	TCL	NSE
Date						
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 [46]:
```
invest_cols=invest_df.columns
for investment in invest_cols:
    print(investment)
```

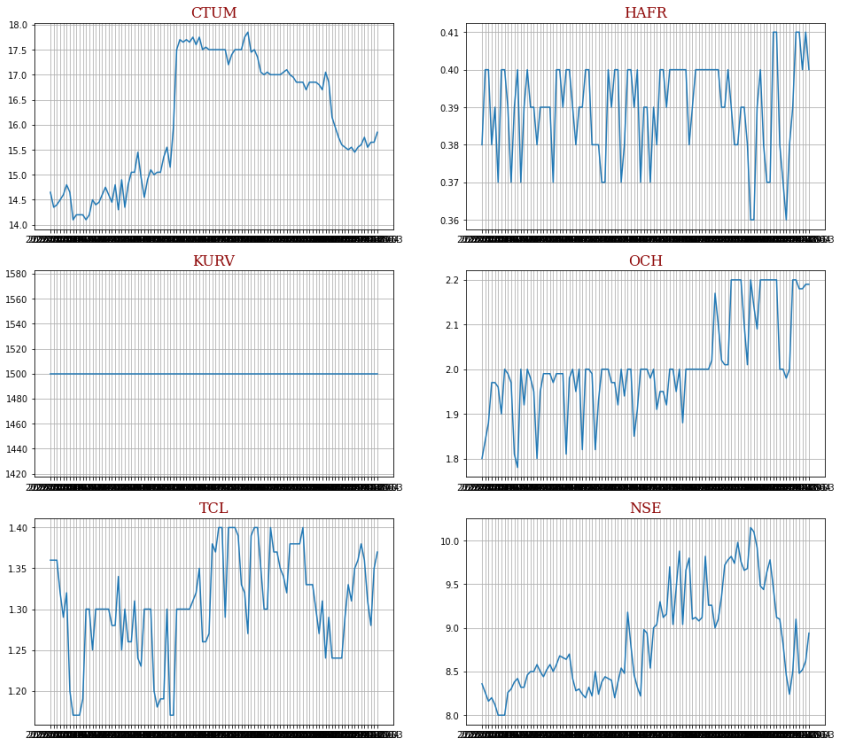
```
CTUM
HAFR
KURV
OCH
TCL
NSE
```

In [47]:
```
invest_cols = invest_df.columns
```

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

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

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



In [48]:
```
manu_df=df.loc[:,'BOC':'UNGA'].copy()
manu_df.head()
```

Out[48]:

	BOC	BAT	CARB	EABL	EVRD	FTGH	ORCH	MSC	UNGA
Date									
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 [49]:
```
manu_cols=manu_df.columns
for manufacturing in manu_cols:
    print(manufacturing)
```

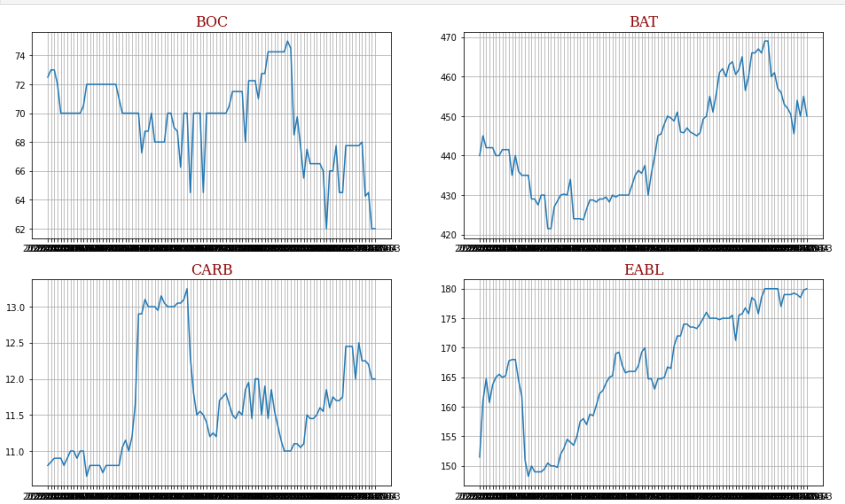
```
BOC
BAT
CARB
EABL
EVRD
FTGH
ORCH
MSC
UNGA
```

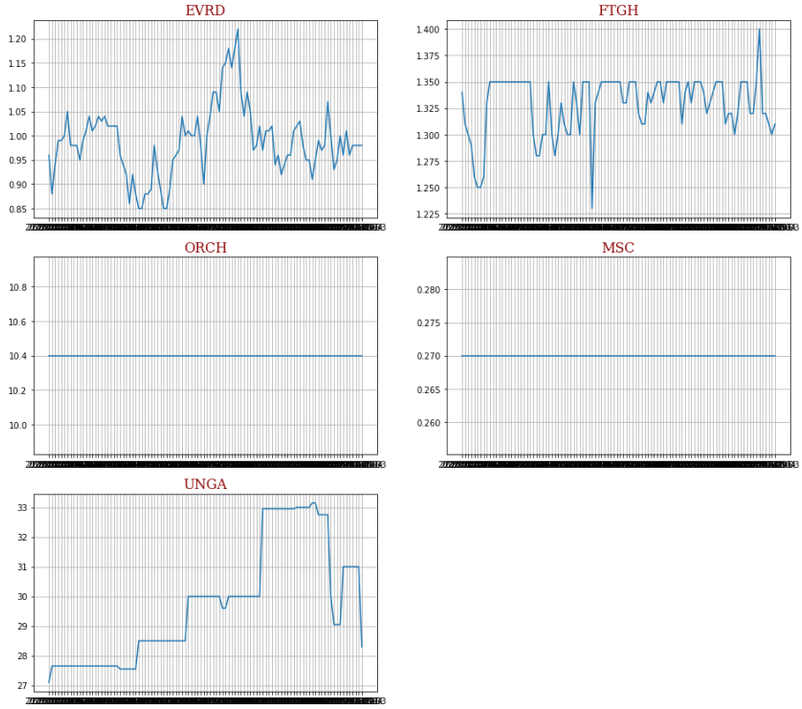
In [50]:
```
manu_cols = manu_df.columns

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

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

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





In [55]:
```
insur_df.corr(method='pearson')
```

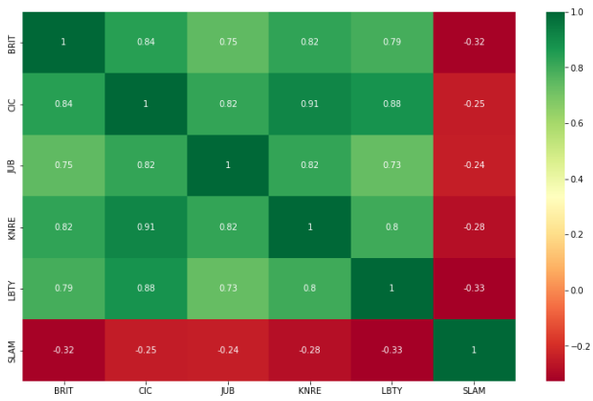
Out[55]:

	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
SLAM	-0.318103	-0.247342	-0.241677	-0.282419	-0.326399	1.000000

In [58]:
```
corr_df=insur_df.corr(method='pearson')
```

In [59]:
```
plt.figure(figsize=(13, 8))
sns.heatmap(corr_df, annot=True, cmap='RdYlGn')
plt.figure()
```

Out[59]:
```
<Figure size 432x288 with 0 Axes>
```




```
<Figure size 432x288 with 0 Axes>
```

In [60]:
```
agric_df.corr(method='pearson')
```

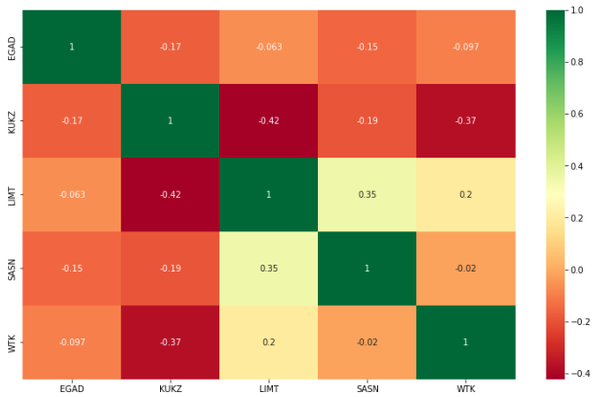
Out[60]:

	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 [61]:
```
corr_df=agric_df.corr(method='pearson')
```

In [62]:
```
plt.figure(figsize=(13, 8))
sns.heatmap(corr_df, annot=True, cmap='RdYlGn')
plt.figure()
```

Out[62]:
```
<Figure size 432x288 with 0 Axes>
```




```
<Figure size 432x288 with 0 Axes>
```

In [63]:
```
comm_df.corr(method='pearson')
```

Out[63]:

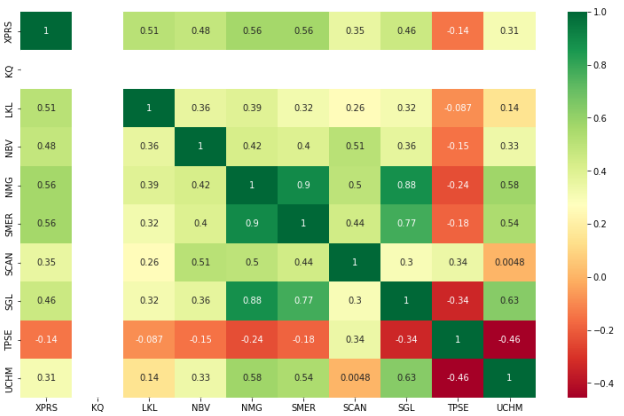
	XPRS	KQ	LKL	NBV	NMG	SMER	SCAN	SGL	TPSE	UCHM

XPRS	1.000000	NaN	0.505657	0.482448	0.564433	0.564130	0.353536	0.456735	-0.143478	0.310362
KQ	NaN	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	0.142561
NBV	0.482448	NaN	0.361295	1.000000	0.423817	0.398074	0.508718	0.363434	-0.153862	0.334953
NMG	0.564433	NaN	0.387344	0.423817	1.000000	0.896570	0.497322	0.879855	-0.235356	0.575009
SMER	0.564130	NaN	0.318353	0.398074	0.896570	1.000000	0.441286	0.774443	-0.182699	0.542615
SCAN	0.353536	NaN	0.257131	0.508718	0.497322	0.441286	1.000000	0.296204	0.344714	0.004806
SGL	0.456735	NaN	0.317583	0.363434	0.879855	0.774443	0.296204	1.000000	-0.340694	0.633963
TPSE	-0.143478	NaN	-0.087045	-0.153862	-0.235356	-0.182699	0.344714	-0.340694	1.000000	-0.455251
UCHM	0.310362	NaN	0.142561	0.334953	0.575009	0.542615	0.004806	0.633963	-0.455251	1.000000

In [64]:
```
corr_df=comm_df.corr(method='pearson')
```

In [65]:
```
plt.figure(figsize=(13, 8))
sns.heatmap(corr_df, annot=True, cmap='RdYlGn')
plt.figure()
```

Out[65]:
```
<Figure size 432x288 with 0 Axes>
```




```
<Figure size 432x288 with 0 Axes>
```

In [66]:
```
bank_df.corr(method='pearson')
```

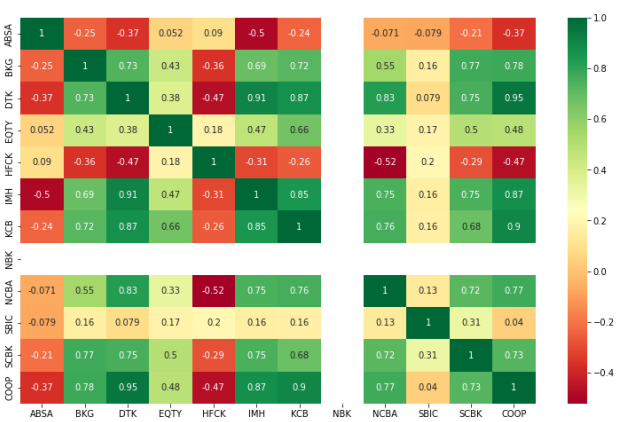
Out[66]:

	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

In [67]:
```
corr_df=bank_df.corr(method='pearson')
```

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

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




```
<Figure size 432x288 with 0 Axes>
```

In [69]:
```
const_df.corr(method='pearson')
```

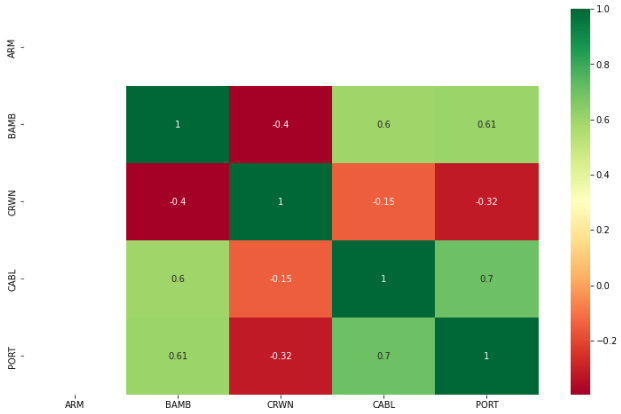
Out[69]:

	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 [70]:
```
corr_df=const_df.corr(method='pearson')
```

In [71]:
```
plt.figure(figsize=(13, 8))
sns.heatmap(corr_df, annot=True, cmap='RdYlGn')
plt.figure()
```

Out[71]:
```
<Figure size 432x288 with 0 Axes>
```




```
<Figure size 432x288 with 0 Axes>
```

In [72]:
```
energ_df.corr(method='pearson')
```

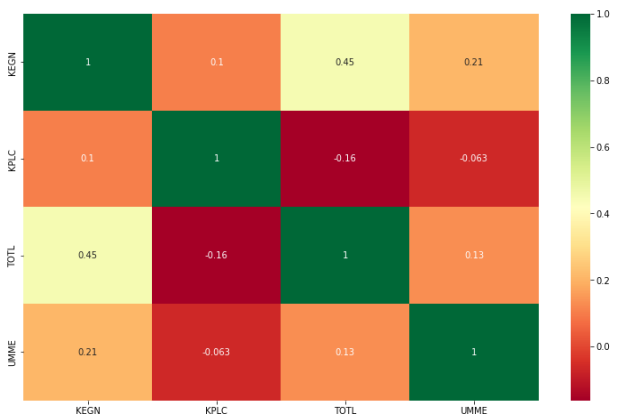
Out[72]:

	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 [73]:
```
corr_df=energ_df.corr(method='pearson')
```

In [74]:
```
plt.figure(figsize=(13, 8))
sns.heatmap(corr_df, annot=True, cmap='RdYlGn')
plt.figure()
```

Out[74]:
```
<Figure size 432x288 with 0 Axes>
```




```
<Figure size 432x288 with 0 Axes>
```

In [75]:
```
invest_df.corr(method='pearson')
```

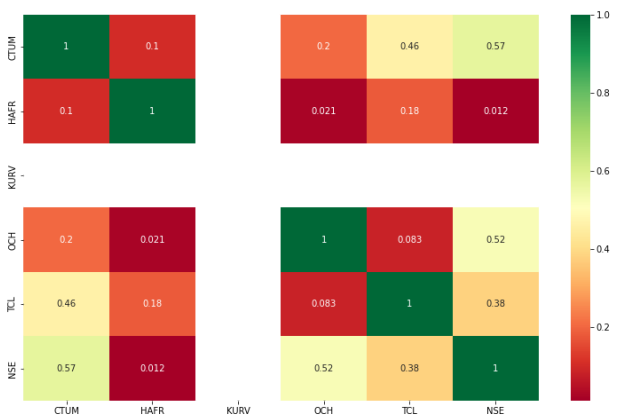
Out[75]:

	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 [76]:
```
corr_df=invest_df.corr(method='pearson')
```

In [77]:
```
plt.figure(figsize=(13, 8))
sns.heatmap(corr_df, annot=True, cmap='RdYlGn')
plt.figure()
```

Out[77]:
```
<Figure size 432x288 with 0 Axes>
```




```
<Figure size 432x288 with 0 Axes>
```

In [78]:
```
manu_df.corr(method='pearson')
```

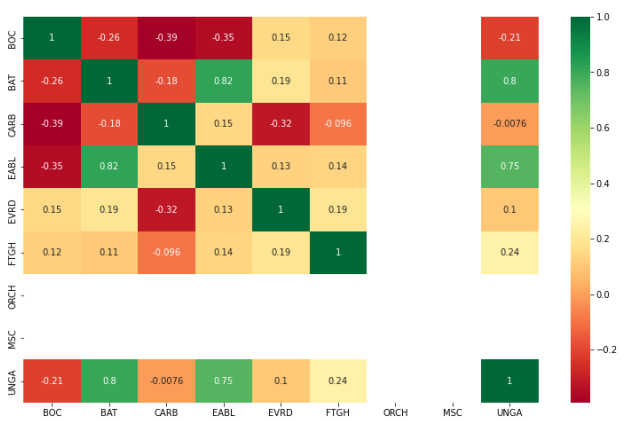
Out[78]:

	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
FTGH	0.124104	0.109096	-0.095847	0.141361	0.187224	1.000000	NaN	NaN	0.243746
ORCH	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
MSC	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
UNGA	-0.213949	0.802586	-0.007650	0.753815	0.101246	0.243746	NaN	NaN	1.000000

In [79]:
```
corr_df=manu_df.corr(method='pearson')
```

In [80]:
```
plt.figure(figsize=(13, 8))
sns.heatmap(corr_df, annot=True, cmap='RdYlGn')
plt.figure()
```

Out[80]:
```
<Figure size 432x288 with 0 Axes>
```




```
<Figure size 432x288 with 0 Axes>
```

In [ ]: