Fortune Global 500 from 2019-2021

URL: https://www.kaggle.com/prasertk/fortune-global-500-from-20192021

Exploring the change in top Fortune global 500 companies from 2019-2021

In [71]:
```
# importing the libraries
import numpy as np
import scipy as sp
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

In [3]:
```
#Read the data from a csv file
data =pd.read_csv("./fortune_global_500_from_2019-2021.csv",squeeze=True)
```

In [4]:
```
# Reading the data in the dataframe
fortune_data=pd.DataFrame(data)
```

In [5]:
```
fortune_data.head()
```

Out[5]:

	year	rank	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity	hqstate	newcor
0	2019	1.0	Walmart	Retailing	General Merchandisers	514405.0	2.8	6670.0	-32.4	219295.0	2200000	Bentonville	Arkansas	
1	2019	2.0	Sinopec Group	Energy	Petroleum Refining	414649.9	26.8	5845.0	280.1	329186.3	619151	Beijing	NaN	
2	2019	3.0	Royal Dutch Shell	Energy	Petroleum Refining	396556.0	27.2	23352.0	79.9	399194.0	81000	The Hague	NaN	
3	2019	4.0	China National Petroleum	Energy	Petroleum Refining	392976.6	20.5	2270.5	NaN	601899.9	1382401	Beijing	NaN	
4	2019	5.0	State Grid	Energy	Utilities	387056.0	10.9	8174.8	-14.3	572309.5	917717	Beijing	NaN	

Exploring few data from the whole dataset in order to analyse them

In [6]:
```
# dropping the columns which not considering to analyse
data=fortune_data.drop(['year','rank','permalink','newcomer','hqstate','profitable','ceowoman','jobgrowth'], axis = 1)
```

In [7]:
```
# Final data to perform analysis.
data
```

Out[7]:

	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
0	Walmart	Retailing	General Merchandisers	514405.0	2.8	6670.0	-32.4	219295.0	2200000	Bentonville
1	Sinopec Group	Energy	Petroleum Refining	414649.9	26.8	5845.0	280.1	329186.3	619151	Beijing
2	Royal Dutch Shell	Energy	Petroleum Refining	396556.0	27.2	23352.0	79.9	399194.0	81000	The Hague
3	China National Petroleum	Energy	Petroleum Refining	392976.6	20.5	2270.5	NaN	601899.9	1382401	Beijing
4	State Grid	Energy	Utilities	387056.0	10.9	8174.8	-14.3	572309.5	917717	Beijing
...	...	...	...	...	...	...	...	...	...	...
1495	Truist Financial	Financials	Banks: Commercial and Savings	24427.0	66.6	4482.0	39.0	509228.0	53638	Charlotte
1496	China Reinsurance (Group)	Financials	Insurance: Property and Casualty (Stock)	24376.0	18.1	827.6	-5.5	69513.7	63914	Beijing
1497	Commonwealth Bank of Australia	Financials	Banks: Commercial and Savings	24362.0	-18.7	6457.1	5.4	698585.9	43585	Sydney
1498	Flex	Technology	Electronics, Electrical Equip.	24124.0	-0.4	613.0	599.9	15836.0	167201	Singapore
1499	Rite Aid	Food & Drug Stores	Food & Drug Stores	24043.4	9.6	-90.9	NaN	9335.4	50000	Camp Hill

1500 rows × 10 columns

In [8]:
```
#Removing the NAN value from the above dataset
clean_data=data.dropna()
```

In [8]:
```
# cleaned data
clean_data
```

Out[8]:

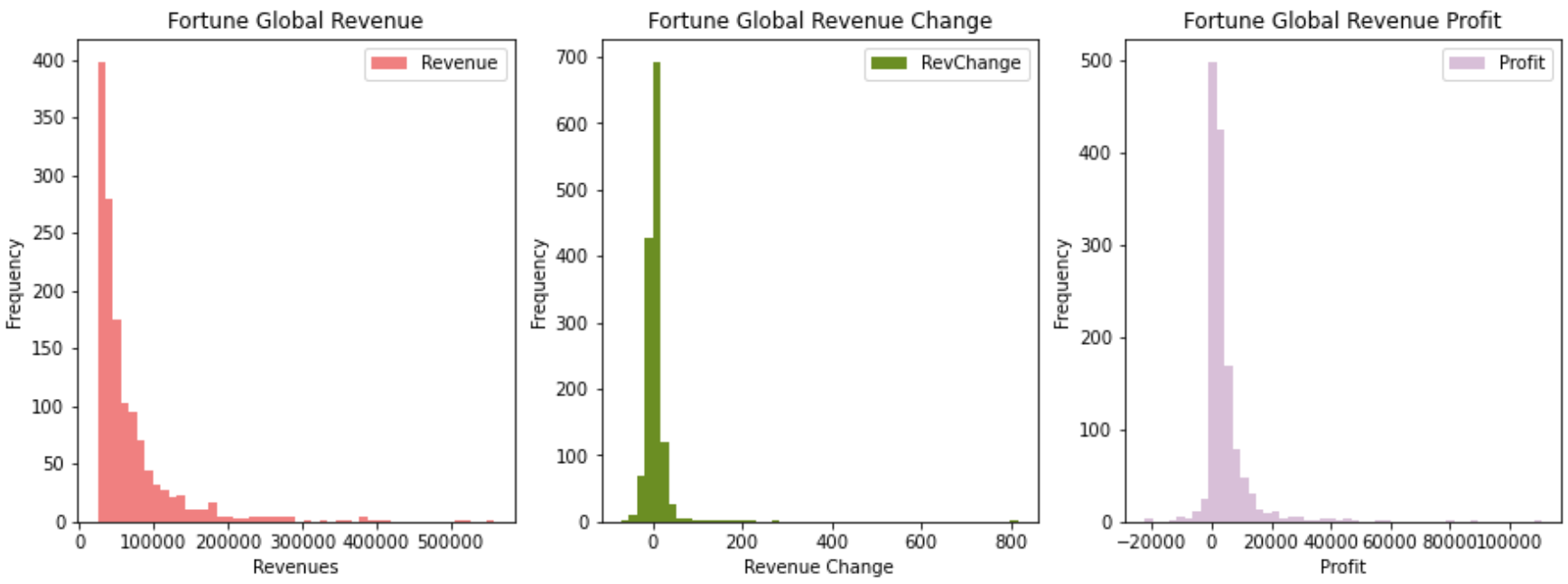
	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
0	Walmart	Retailing	General Merchandisers	514405.0	2.8	6670.0	-32.4	219295.0	2200000	Bentonville
1	Sinopec Group	Energy	Petroleum Refining	414649.9	26.8	5845.0	280.1	329186.3	619151	Beijing
2	Royal Dutch Shell	Energy	Petroleum Refining	396556.0	27.2	23352.0	79.9	399194.0	81000	The Hague
4	State Grid	Energy	Utilities	387056.0	10.9	8174.8	-14.3	572309.5	917717	Beijing
5	Saudi Aramco	Energy	Mining, Crude-Oil Production	355905.0	35.3	110974.5	46.9	358872.9	76418	Dhahran
...	...	...	...	...	...	...	...	...	...	...
1494	Eli Lilly	Health Care	Pharmaceuticals	24539.8	9.9	6193.7	-25.5	46633.1	35000	Indianapolis
1495	Truist Financial	Financials	Banks: Commercial and Savings	24427.0	66.6	4482.0	39.0	509228.0	53638	Charlotte
1496	China Reinsurance (Group)	Financials	Insurance: Property and Casualty (Stock)	24376.0	18.1	827.6	-5.5	69513.7	63914	Beijing
1497	Commonwealth Bank of Australia	Financials	Banks: Commercial and Savings	24362.0	-18.7	6457.1	5.4	698585.9	43585	Sydney
1498	Flex	Technology	Electronics, Electrical Equip.	24124.0	-0.4	613.0	599.9	15836.0	167201	Singapore

1369 rows × 10 columns

(a) Plot the histograms of primary (important) continuous variables and probability distributions of categorical variables.

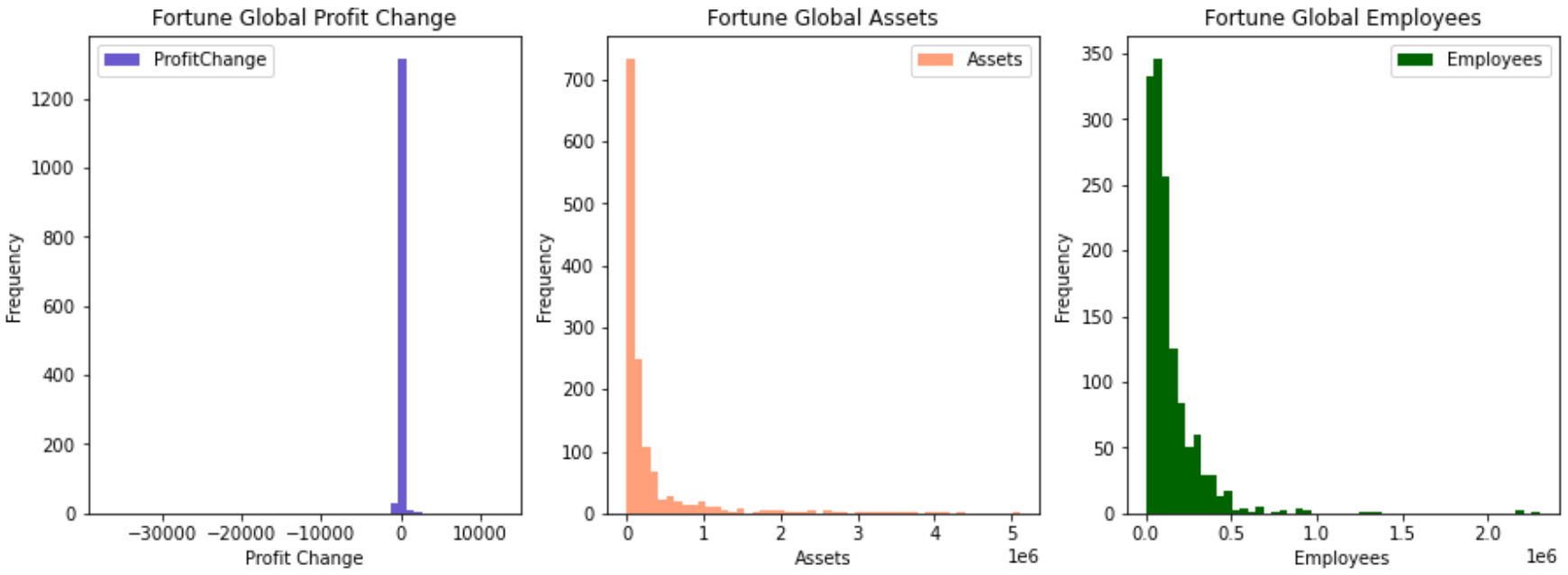
Plotting histograms for both continuos variables and categorical variables

In [9]:
```
# Plotting Histograms
fig, axis= plt.subplots(1,3, figsize=(15,5))


axis[0].hist (clean_data['revenues'], bins=50, density= False, color='LightCoral', label='Revenue')

axis[1].hist (clean_data['revchange'], bins=50, density= False, color='OliveDrab', label='RevChange')

axis[2].hist (clean_data['profits'], bins=50, density= False, color='Thistle', label='Profit')


axis[0].set(xlabel="Revenues", ylabel="Frequency", title='Fortune Global Revenue')

axis[1].set(xlabel="Revenue Change", ylabel="Frequency", title='Fortune Global Revenue Change')

axis[2].set(xlabel="Profit", ylabel="Frequency", title='Fortune Global Revenue Profit')

axis[0].legend()
axis[1].legend()
axis[2].legend()

plt.show()
```



In [10]:
```
# Plotting Histograms
fig, axis= plt.subplots(1,3, figsize=(15,5))

axis[0].hist (clean_data['prftchange'], bins=50, density= False, color='SlateBlue', label='ProfitChange')

axis[1].hist (clean_data['assets'], bins=50, density= False, color='LightSalmon', label='Assets')

axis[2].hist (clean_data['employees'], bins=50, density= False, color='DarkGreen', label='Employees')


axis[0].set(xlabel="Profit Change", ylabel="Frequency", title='Fortune Global Profit Change')
```

```
axis[1].set(xlabel="Assets", ylabel="Frequency", title='Fortune Global Assets')

axis[2].set(xlabel="Employees", ylabel="Frequency", title='Fortune Global Employees')

axis[0].legend()
axis[1].legend()
axis[2].legend()

plt.show()
```



In [9]:
```
# validating the dataset
clean_data.head()
```

Out[9]:

	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
0	Walmart	Retailing	General Merchandisers	514405.0	2.8	6670.0	-32.4	219295.0	2200000	Bentonville
1	Sinopec Group	Energy	Petroleum Refining	414649.9	26.8	5845.0	280.1	329186.3	619151	Beijing
2	Royal Dutch Shell	Energy	Petroleum Refining	396556.0	27.2	23352.0	79.9	399194.0	81000	The Hague
4	State Grid	Energy	Utilities	387056.0	10.9	8174.8	-14.3	572309.5	917717	Beijing
5	Saudi Aramco	Energy	Mining, Crude-Oil Production	355905.0	35.3	110974.5	46.9	358872.9	76418	Dhahran

In [10]:
```
# computing probabability distribution for different companies
freq_name=clean_data.groupby(['name']).size()
propor_name=freq_name/sum(freq_name)
propor_name.head()
```

Out[10]:
```
name
3M           0.002191
ABB          0.002191
ACS          0.002191
AEON         0.002191
AIA Group    0.002191
dtype: float64
```

In [11]:
```
probab_nam=pd.DataFrame(propor_name)
probab_nam.columns=['CompanyValues']
probab_nam
```

Out[11]:

	CompanyValues
name	
3M	0.002191
ABB	0.002191
ACS	0.002191
AEON	0.002191
AIA Group	0.002191
...	...
Zhejiang Geely Holding Group	0.002191
Zhejiang Hengyi Group	0.000730
Zhejiang Rongsheng Holding Group	0.000730
Zijin Mining Group	0.000730
Zurich Insurance Group	0.002191

580 rows × 1 columns

In [12]:
```
probab_nam.reset_index(inplace=True)
propor_names = probab_nam.rename(columns = {'name':'Name'})
```

```
propor_names
```

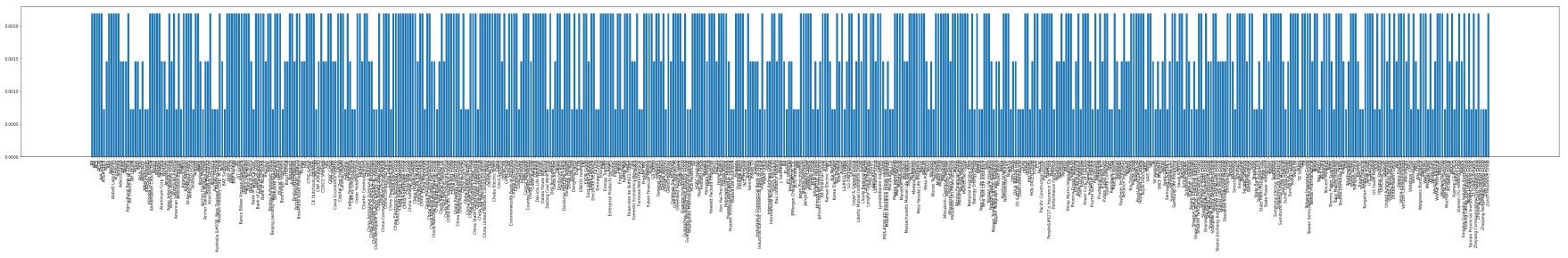
Out[12]:

	Name	CompanyValues
0	3M	0.002191
1	ABB	0.002191
2	ACS	0.002191
3	AEON	0.002191
4	AIA Group	0.002191
...	...	...
575	Zhejiang Geely Holding Group	0.002191
576	Zhejiang Hengyi Group	0.000730
577	Zhejiang Rongsheng Holding Group	0.000730
578	Zijin Mining Group	0.000730
579	Zurich Insurance Group	0.002191

580 rows × 2 columns

In [13]:

```
plt.figure(figsize=(70,7))
plt.bar(propor_names['Name'],propor_names['CompanyValues'])
plt.xticks(rotation=90)
plt.show()
```



As, we can see from the above plot that the probability distribution for categorical variable 'Name' is not much useful because we can't distinguish the company names.

In [14]:

```
# computing probabability distribution for different sector
freq_sector=clean_data.groupby(['sector']).size()
propor_sector=freq_sector/sum(freq_sector)
propor_sector.head()
```

Out[14]:

```
sector
Aerospace & Defense    0.027757
Apparel                0.005844
Business Services      0.005113
Chemicals              0.015340
Energy                 0.146822
dtype: float64
```

In [15]:

```
probab_distr_sector=pd.DataFrame(propor_sector)
probab_distr_sector.columns=['Sector_Values']
probab_distr_sector
```

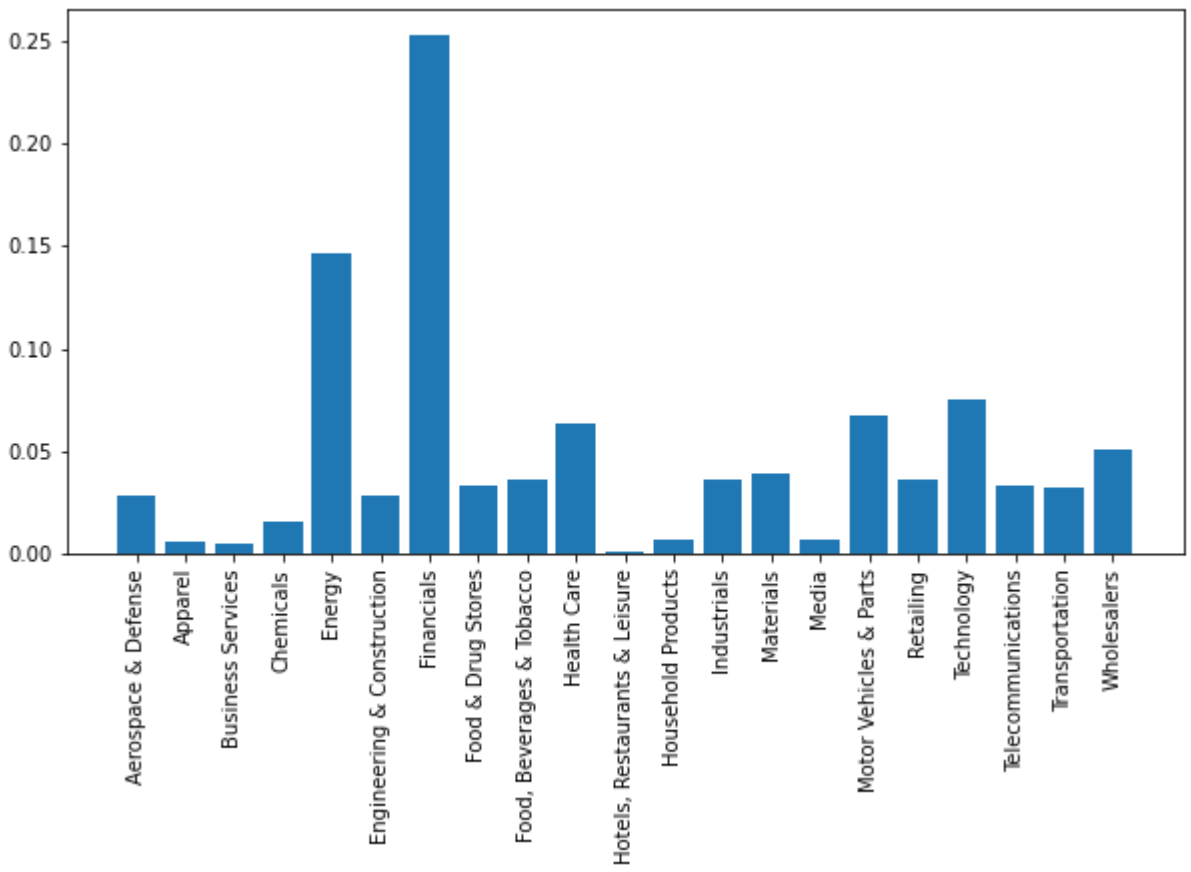
Out[15]:

sector	Sector_Values
Aerospace & Defense	0.027757
Apparel	0.005844
Business Services	0.005113
Chemicals	0.015340
Energy	0.146822
Engineering & Construction	0.028488
Financials	0.252739
Food & Drug Stores	0.033601
Food, Beverages & Tobacco	0.036523
Health Care	0.063550
Hotels, Restaurants & Leisure	0.000730
Household Products	0.006574
Industrials	0.036523
Materials	0.039445
Media	0.006574
Motor Vehicles & Parts	0.067202

sector	Sector_Values
Retailing	0.035793
Technology	0.075237
Telecommunications	0.032871
Transportation	0.032140
Wholesalers	0.051132

In [16]:
```
probab_distr_sector.reset_index(inplace=True)
probab_distr_sect = probab_distr_sector.rename(columns = {'sector':'Sector'})
```

In [17]:
```
plt.figure(figsize=(10,5))
plt.bar(probab_distr_sect['Sector'],probab_distr_sect['Sector_Values'])
plt.xticks(rotation=90)

plt.show()
```



In [18]:
```
# computing probabability distribution for different industries
freq_industry=clean_data.groupby(['industry']).size()
propor_industry=freq_industry/sum(freq_industry)
propor_industry.head()
```

Out[18]:
```
industry
Aerospace & Defense              0.017531
Aerospace and Defense            0.010226
Airlines                         0.008766
Apparel                          0.005844
Banks: Commercial and Savings    0.105917
dtype: float64
```

In [19]:
```
probab_distr_industry=pd.DataFrame(propor_industry)
probab_distr_industry.columns=['Industry_Values']
probab_distr_industry.head()
```

Out[19]:

industry	Industry_Values
Aerospace & Defense	0.017531
Aerospace and Defense	0.010226
Airlines	0.008766
Apparel	0.005844
Banks: Commercial and Savings	0.105917

In [20]:
```
probab_distr_industry.reset_index(inplace=True)
probab_distr_indust = probab_distr_industry.rename(columns = {'industry':'Industry'})
```

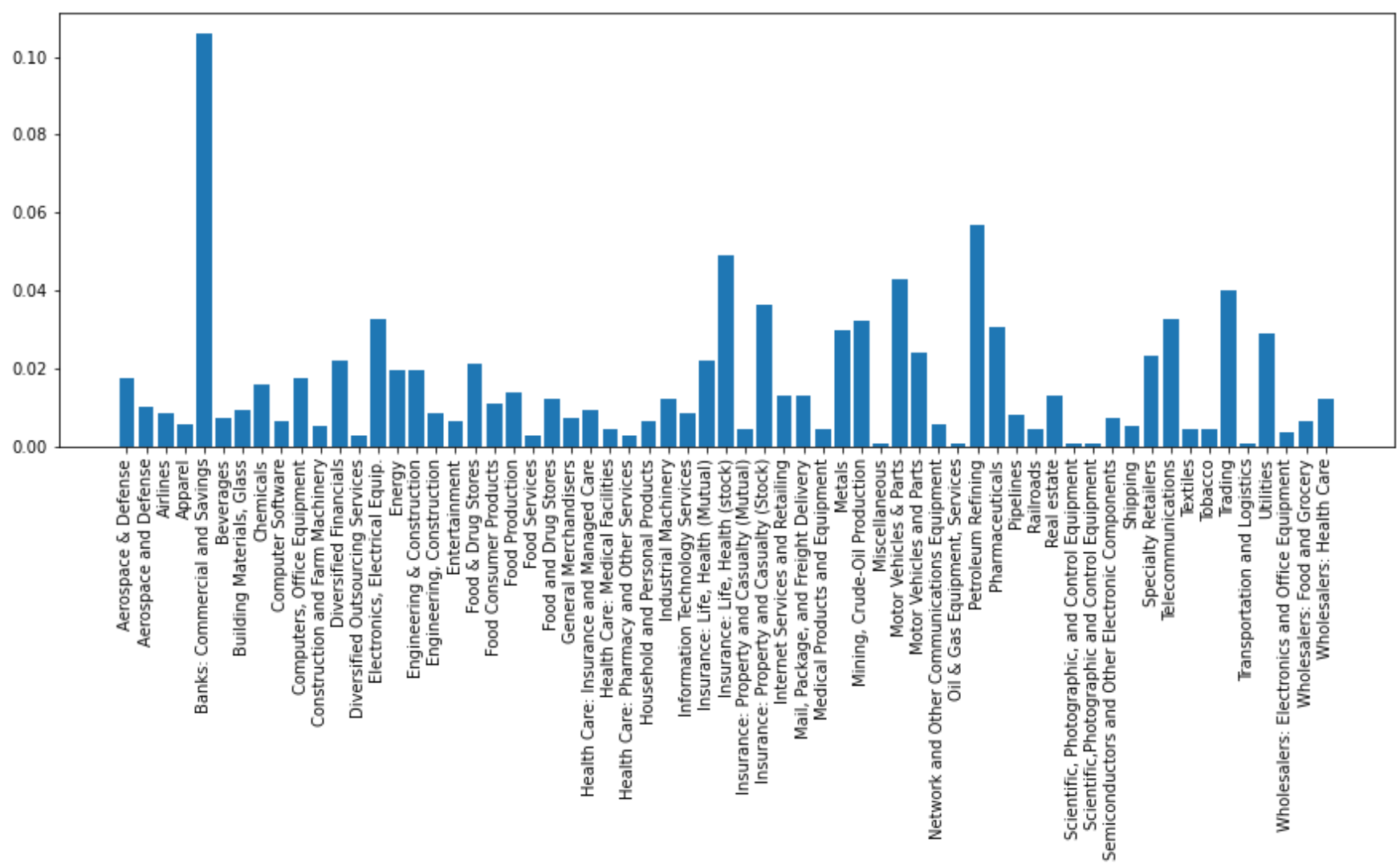
In [21]:
```
probab_distr_indust.head()
```

Out[21]:

	Industry	Industry_Values
0	Aerospace & Defense	0.017531

	Industry	Industry_Values
1	Aerospace and Defense	0.010226
2	Airlines	0.008766
3	Apparel	0.005844
4	Banks: Commercial and Savings	0.105917

In [22]:
```
plt.figure(figsize=(15,5))
plt.bar(probab_distr_indust['Industry'],probab_distr_indust['Industry_Values'])
plt.xticks(rotation=90)

plt.show()
```



In [27]:
```
# computing probabability distribution for different companies HeadQuaters
freq_hQcity=clean_data.groupby(['hqcity']).size()
propor_hqcity=freq_hQcity/sum(freq_hQcity)
propor_hqcity.head()
```

Out[27]:
```
hqcity
Abbott Park    0.002191
Amsterdam      0.007305
Anshan         0.000730
Armonk         0.002191
Arteixo        0.001461
dtype: float64
```

In [28]:
```
probab_distr_hqcity=pd.DataFrame(propor_hqcity)
probab_distr_hqcity.columns=['HeadQuater_Values']
probab_distr_hqcity
```

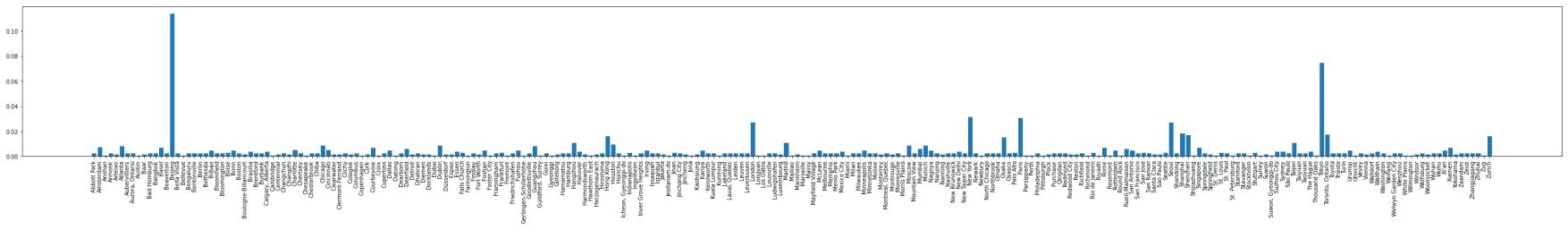
Out[28]:

hqcity	HeadQuater_Values
Abbott Park	0.002191
Amsterdam	0.007305
Anshan	0.000730
Armonk	0.002191
Arteixo	0.001461
...	...
Zeist	0.002191
Zhangjiagang	0.002191
Zhuhai	0.002191
Zug	0.000730
Zurich	0.016070

251 rows × 1 columns

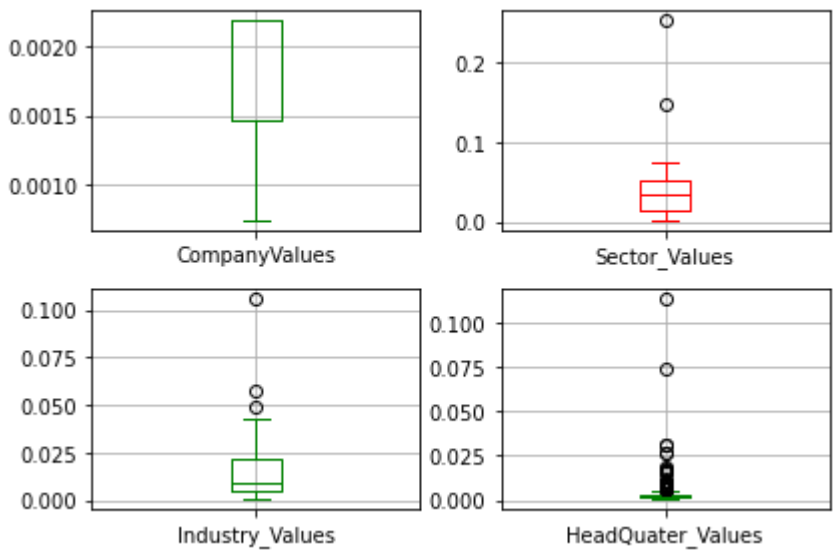
In [29]:
```
probab_distr_hqcity.reset_index(inplace=True)
probab_distr_hqcity = probab_distr_hqcity.rename(columns = {'hqcity':'Hqcity'})
```

In [136…
```
plt.figure(figsize=(50,5))
plt.bar(probab_distr_hqcity['Hqcity'],probab_distr_hqcity['HeadQuater_Values'])
plt.xticks(rotation=90)

plt.show()
```



(b) Plot the box plots of all the primary variables in the data. Identify and delete a couple of extreme outliers from the data if there are any.

In [31]:
```
fig=plt.figure()
ax1=fig.add_subplot(221)
ax2=fig.add_subplot(222)
ax3=fig.add_subplot(223)
ax4=fig.add_subplot(224)
propor_names.boxplot( ax=ax1, color='green')
probab_distr_sect.boxplot( ax=ax2, color='red')
probab_distr_indust.boxplot( ax=ax3, color='green')
probab_distr_hqcity.boxplot( ax=ax4, color='green')
plt.tight_layout()
```



From the above box plot it can be seen that in the **Sector** variable after **0.1** there is an outliers. Moreover, in Industry we can see that there is another outliers after **0.050**, whereas in **HeadQuater City** Variable we have lots of outliers. Moreover there is no outliers for the company values. Now, I will remove the oultiers from each variable.

Cleaning Sector Values

In [32]:
```
probab_distr_sect
```

Out[32]:

	Sector	Sector_Values
0	Aerospace & Defense	0.027757
1	Apparel	0.005844
2	Business Services	0.005113
3	Chemicals	0.015340
4	Energy	0.146822
5	Engineering & Construction	0.028488
6	Financials	0.252739
7	Food & Drug Stores	0.033601
8	Food, Beverages & Tobacco	0.036523
9	Health Care	0.063550
10	Hotels, Restaurants & Leisure	0.000730
11	Household Products	0.006574
12	Industrials	0.036523
13	Materials	0.039445
14	Media	0.006574
15	Motor Vehicles & Parts	0.067202
16	Retailing	0.035793
17	Technology	0.075237
18	Telecommunications	0.032871

	Sector	Sector_Values
19	Transportation	0.032140
20	Wholesalers	0.051132

From the above data we can see that Financials sector is the oultier which contains values as 0.252739 and Energy as 0.146822. So, we will delete this from the dataset.

In [33]:
```
#define list of values to be removed
values = ['Energy', 'Financials']
```

In [34]:
```
#drop all rows that have above values
df = clean_data[clean_data.sector.isin(values) == False]
```

In [35]:
```
df
```

Out[35]:

	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
0	Walmart	Retailing	General Merchandisers	514405.0	2.8	6670.0	-32.4	219295.0	2200000	Bentonville
8	Volkswagen	Motor Vehicles & Parts	Motor Vehicles and Parts	278341.5	7.0	14322.5	9.3	523672.3	664496	Wolfsburg
9	Toyota Motor	Motor Vehicles & Parts	Motor Vehicles and Parts	272612.0	2.8	16982.0	-24.6	469295.6	370870	Toyota
10	Apple	Technology	Computers, Office Equipment	265595.0	15.9	59531.0	23.1	365725.0	132000	Cupertino
12	Amazon.com	Retailing	Internet Services and Retailing	232887.0	30.9	10073.0	232.1	162648.0	647500	Seattle
...	...	...	...	...	...	...	...	...	...	...
1489	TD Synnex	Wholesalers	Wholesalers: Electronics and Office Equipment	24675.6	3.9	529.2	5.7	13468.6	277900	Fremont
1491	Holcim	Materials	Building Materials, Glass	24653.8	-8.3	1807.9	-20.0	60235.4	67409	Zug
1493	Alfresa Holdings	Health Care	Wholesalers: Health Care	24556.3	-1.1	231.1	-37.6	11904.7	12045	Tokyo
1494	Eli Lilly	Health Care	Pharmaceuticals	24539.8	9.9	6193.7	-25.5	46633.1	35000	Indianapolis
1498	Flex	Technology	Electronics, Electrical Equip.	24124.0	-0.4	613.0	599.9	15836.0	167201	Singapore

822 rows × 10 columns

In [37]:
```
# Re-computing the values again
freq_sector_new=df.groupby(['sector']).size()
propor_sector_new=freq_sector_new/sum(freq_sector_new)
propor_sector_new.head()
```

Out[37]:
```
sector
Aerospace & Defense           0.046229
Apparel                       0.009732
Business Services             0.008516
Chemicals                     0.025547
Engineering & Construction    0.047445
dtype: float64
```

In [38]:
```
probab_distr_sector_new=pd.DataFrame(propor_sector_new)
probab_distr_sector_new.columns=['Sector_Values']
probab_distr_sector_new
```

Out[38]:

sector	Sector_Values
Aerospace & Defense	0.046229
Apparel	0.009732
Business Services	0.008516
Chemicals	0.025547
Engineering & Construction	0.047445
Food & Drug Stores	0.055961
Food, Beverages & Tobacco	0.060827
Health Care	0.105839
Hotels, Restaurants & Leisure	0.001217
Household Products	0.010949
Industrials	0.060827
Materials	0.065693

sector	Sector_Values
Media	0.010949
Motor Vehicles & Parts	0.111922
Retailing	0.059611
Technology	0.125304
Telecommunications	0.054745
Transportation	0.053528
Wholesalers	0.085158

In [39]:
```
probab_distr_sector_new.reset_index(inplace=True)
probab_distr_sector = probab_distr_sector_new.rename(columns = {'sector':'Sector'})
```

In [40]:
```
probab_distr_sector
```

Out[40]:

	Sector	Sector_Values
0	Aerospace & Defense	0.046229
1	Apparel	0.009732
2	Business Services	0.008516
3	Chemicals	0.025547
4	Engineering & Construction	0.047445
5	Food & Drug Stores	0.055961
6	Food, Beverages & Tobacco	0.060827
7	Health Care	0.105839
8	Hotels, Restaurants & Leisure	0.001217
9	Household Products	0.010949
10	Industrials	0.060827
11	Materials	0.065693
12	Media	0.010949
13	Motor Vehicles & Parts	0.111922
14	Retailing	0.059611
15	Technology	0.125304
16	Telecommunications	0.054745
17	Transportation	0.053528
18	Wholesalers	0.085158

Removing outliers from HeadQuaterCity

In [41]:
```
# validating values above 0.008
prob=probab_distr_hqcity[probab_distr_hqcity['HeadQuater_Values']>0.008]
```

In [42]:
```
prob
```

Out[42]:

	Hqcity	HeadQuater_Values
5	Atlanta	0.008035
14	Beijing	0.113952
41	Chicago	0.008766
62	Dublin	0.008766
79	Guangzhou	0.008035
86	Hangzhou	0.010957
92	Hong Kong	0.016070
93	Houston	0.009496
118	London	0.027027
124	Madrid	0.010957
146	Moscow	0.008766
149	Munich	0.008766
157	New York	0.031410
163	Osaka	0.015340

	Hqcity	HeadQuater_Values
166	Paris	0.030679
193	Seoul	0.027027
195	Shanghai	0.018262
196	Shenzhen	0.016801
215	Taipei	0.010957
220	Tokyo	0.074507
221	Toronto, Ontario	0.017531
250	Zurich	0.016070

In [43]:
```
#define list of values to be removed
values = prob['Hqcity']
values
```

Out[43]:
```
5              Atlanta
14             Beijing
41             Chicago
62              Dublin
79           Guangzhou
86            Hangzhou
92           Hong Kong
93             Houston
118             London
124             Madrid
146             Moscow
149             Munich
157           New York
163              Osaka
166              Paris
193              Seoul
195           Shanghai
196           Shenzhen
215             Taipei
220              Tokyo
221    Toronto, Ontario
250             Zurich
Name: Hqcity, dtype: object
```

In [44]:
```
#drop any rows that have above values
data_hq= df[df.hqcity.isin(values) == False]
```

In [45]:
```
data_hq
```

Out[45]:

	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
0	Walmart	Retailing	General Merchandisers	514405.0	2.8	6670.0	-32.4	219295.0	2200000	Bentonville
8	Volkswagen	Motor Vehicles & Parts	Motor Vehicles and Parts	278341.5	7.0	14322.5	9.3	523672.3	664496	Wolfsburg
9	Toyota Motor	Motor Vehicles & Parts	Motor Vehicles and Parts	272612.0	2.8	16982.0	-24.6	469295.6	370870	Toyota
10	Apple	Technology	Computers, Office Equipment	265595.0	15.9	59531.0	23.1	365725.0	132000	Cupertino
12	Amazon.com	Retailing	Internet Services and Retailing	232887.0	30.9	10073.0	232.1	162648.0	647500	Seattle
...	...	...	...	...	...	...	...	...	...	...
1488	Gilead Sciences	Health Care	Pharmaceuticals	24689.0	10.0	123.0	-97.7	68407.0	13600	Foster City
1489	TD Synnex	Wholesalers	Wholesalers: Electronics and Office Equipment	24675.6	3.9	529.2	5.7	13468.6	277900	Fremont
1491	Holcim	Materials	Building Materials, Glass	24653.8	-8.3	1807.9	-20.0	60235.4	67409	Zug
1494	Eli Lilly	Health Care	Pharmaceuticals	24539.8	9.9	6193.7	-25.5	46633.1	35000	Indianapolis
1498	Flex	Technology	Electronics, Electrical Equip.	24124.0	-0.4	613.0	599.9	15836.0	167201	Singapore

452 rows × 10 columns

In [46]:
```
# Recomputing the probabilites again
freq_hq=data_hq.groupby(['hqcity']).size()
propor_hq_new=freq_hq/sum(freq_hq)
propor_hq_new.head()
```

Out[46]:
```
hqcity
Abbott Park    0.006637
Amsterdam      0.008850
Anshan         0.002212
Armonk         0.006637
```

```
Arteixo        0.004425
dtype: float64
```

In [47]:
```
probab_distr_hq=pd.DataFrame(propor_hq_new)
probab_distr_hq.columns=['HeadQuater_Values']
probab_distr_hq
```

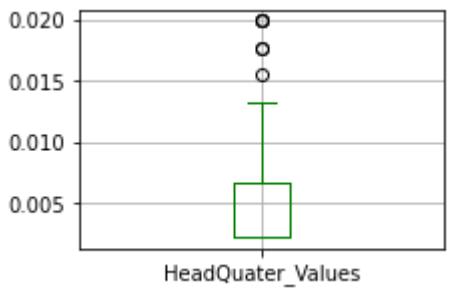
Out[47]:

hqcity	HeadQuater_Values
Abbott Park	0.006637
Amsterdam	0.008850
Anshan	0.002212
Armonk	0.006637
Arteixo	0.004425
...	...
Yokohama	0.004425
Zaandam	0.006637
Zhangjiagang	0.006637
Zhuhai	0.006637
Zug	0.002212

171 rows × 1 columns

In [48]:
```
probab_distr_hq.reset_index(inplace=True)
probab_distr_hcity = probab_distr_hq.rename(columns = {'hqcity':'HeadQuater'})
```

In [49]:
```
# Computing the boxplot to validate the result
fig=plt.figure()
ax1=fig.add_subplot(221)

probab_distr_hcity.boxplot( ax=ax1, color='green')

plt.tight_layout()
```



As I can see from the above box plot that there are still some outliers after 0.015 so we will remove the same.

In [50]:
```
# validating HeadQuater Values above 0.015
prob_new_hq=probab_distr_hcity[probab_distr_hcity['HeadQuater_Values']>0.015]
prob_new_hq
```

Out[50]:

	HeadQuater	HeadQuater_Values
8	Basel	0.019912
30	Cincinnati	0.015487
42	Deerfield	0.017699
128	Rueil-Malmaison	0.017699
135	Singapore	0.019912
165	Xiamen	0.019912

In [51]:
```
#define list of values to be removed
values = prob_new_hq['HeadQuater']
values
```

Out[51]:
```
8               Basel
30          Cincinnati
42           Deerfield
128    Rueil-Malmaison
135          Singapore
165             Xiamen
Name: HeadQuater, dtype: object
```

In [52]:
```
#drop any rows that have above values
data_hqc= data_hq[data_hq.hqcity.isin(values) == False]
```

In [53]:
```
data_hqc
```

Out[53]:

	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
0	Walmart	Retailing	General Merchandisers	514405.0	2.8	6670.0	-32.4	219295.0	2200000	Bentonville
8	Volkswagen	Motor Vehicles & Parts	Motor Vehicles and Parts	278341.5	7.0	14322.5	9.3	523672.3	664496	Wolfsburg
9	Toyota Motor	Motor Vehicles & Parts	Motor Vehicles and Parts	272612.0	2.8	16982.0	-24.6	469295.6	370870	Toyota
10	Apple	Technology	Computers, Office Equipment	265595.0	15.9	59531.0	23.1	365725.0	132000	Cupertino
12	Amazon.com	Retailing	Internet Services and Retailing	232887.0	30.9	10073.0	232.1	162648.0	647500	Seattle
...	...	...	...	...	...	...	...	...	...	...
1487	Gree Electric Appliances	Industrials	Electronics, Electrical Equip.	24709.7	-14.9	3213.8	-10.1	42792.0	83952	Zhuhai
1488	Gilead Sciences	Health Care	Pharmaceuticals	24689.0	10.0	123.0	-97.7	68407.0	13600	Foster City
1489	TD Synnex	Wholesalers	Wholesalers: Electronics and Office Equipment	24675.6	3.9	529.2	5.7	13468.6	277900	Fremont
1491	Holcim	Materials	Building Materials, Glass	24653.8	-8.3	1807.9	-20.0	60235.4	67409	Zug
1494	Eli Lilly	Health Care	Pharmaceuticals	24539.8	9.9	6193.7	-25.5	46633.1	35000	Indianapolis

402 rows × 10 columns

In [54]:
```
#Re-computing the values
freq_hq_new=data_hqc.groupby(['hqcity']).size()
propor_hqnew=freq_hq_new/sum(freq_hq_new)
propor_hqnew.head()
```

Out[54]:
```
hqcity
Abbott Park    0.007463
Amsterdam      0.009950
Anshan         0.002488
Armonk         0.007463
Arteixo        0.004975
dtype: float64
```

In [55]:
```
probabhq=pd.DataFrame(propor_hqnew)
probabhq.columns=['HeadQuater_Values']
probabhq
```

Out[55]:

hqcity	HeadQuater_Values
Abbott Park	0.007463
Amsterdam	0.009950
Anshan	0.002488
Armonk	0.007463
Arteixo	0.004975
...	...
Yokohama	0.004975
Zaandam	0.007463
Zhangjiagang	0.007463
Zhuhai	0.007463
Zug	0.002488

165 rows × 1 columns

In [56]:
```
probabhq.reset_index(inplace=True)
probabhqcity = probabhq.rename(columns = {'hqcity':'HeadQuater'})
```

Cleaning Industry Data

In [57]:
```
probab_distr_indust
```

Out[57]:

	Industry	Industry_Values
0	Aerospace & Defense	0.017531
1	Aerospace and Defense	0.010226
2	Airlines	0.008766
3	Apparel	0.005844

	Industry	Industry_Values
4	Banks: Commercial and Savings	0.105917
...	...	...
58	Transportation and Logistics	0.000730
59	Utilities	0.029218
60	Wholesalers: Electronics and Office Equipment	0.003652
61	Wholesalers: Food and Grocery	0.006574
62	Wholesalers: Health Care	0.012418

63 rows × 2 columns

In [58]:
```
# validating industry values above 0.040
prob_indus=probab_distr_indust[probab_distr_indust['Industry_Values']>=0.040]
prob_indus
```

Out[58]:

	Industry	Industry_Values
4	Banks: Commercial and Savings	0.105917
31	Insurance: Life, Health (stock)	0.048941
40	Motor Vehicles & Parts	0.043097
44	Petroleum Refining	0.056976
57	Trading	0.040175

In [59]:
```
#define list of values to be removed
indus_values = prob_indus['Industry']
indus_values
```

Out[59]:
```
4       Banks: Commercial and Savings
31    Insurance: Life, Health (stock)
40              Motor Vehicles & Parts
44                  Petroleum Refining
57                             Trading
Name: Industry, dtype: object
```

In [60]:
```
#drop any rows that have above values
data_indus= data_hqc[data_hqc.industry.isin(indus_values) == False]
data_indus
```

Out[60]:

	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
0	Walmart	Retailing	General Merchandisers	514405.0	2.8	6670.0	-32.4	219295.0	2200000	Bentonville
8	Volkswagen	Motor Vehicles & Parts	Motor Vehicles and Parts	278341.5	7.0	14322.5	9.3	523672.3	664496	Wolfsburg
9	Toyota Motor	Motor Vehicles & Parts	Motor Vehicles and Parts	272612.0	2.8	16982.0	-24.6	469295.6	370870	Toyota
10	Apple	Technology	Computers, Office Equipment	265595.0	15.9	59531.0	23.1	365725.0	132000	Cupertino
12	Amazon.com	Retailing	Internet Services and Retailing	232887.0	30.9	10073.0	232.1	162648.0	647500	Seattle
...	...	...	...	...	...	...	...	...	...	...
1487	Gree Electric Appliances	Industrials	Electronics, Electrical Equip.	24709.7	-14.9	3213.8	-10.1	42792.0	83952	Zhuhai
1488	Gilead Sciences	Health Care	Pharmaceuticals	24689.0	10.0	123.0	-97.7	68407.0	13600	Foster City
1489	TD Synnex	Wholesalers	Wholesalers: Electronics and Office Equipment	24675.6	3.9	529.2	5.7	13468.6	277900	Fremont
1491	Holcim	Materials	Building Materials, Glass	24653.8	-8.3	1807.9	-20.0	60235.4	67409	Zug
1494	Eli Lilly	Health Care	Pharmaceuticals	24539.8	9.9	6193.7	-25.5	46633.1	35000	Indianapolis

361 rows × 10 columns

In [61]:
```
# Re-computing the values
freq_indus_new=data_indus.groupby(['industry']).size()
propor_indus_new=freq_indus_new/sum(freq_indus_new)
propor_indus_new.head()
```

Out[61]:
```
industry
Aerospace & Defense      0.024931
Aerospace and Defense    0.016620
Airlines                 0.019391
Apparel                  0.013850
Beverages                0.019391
dtype: float64
```

In [62]:
```
probab_distr_indu=pd.DataFrame(propor_indus_new)
probab_distr_indu.columns=['Industry_Values']
probab_distr_indu.head()
```

Out[62]:

industry	Industry_Values
Aerospace & Defense	0.024931
Aerospace and Defense	0.016620
Airlines	0.019391
Apparel	0.013850
Beverages	0.019391

In [63]:
```
probab_distr_indu.reset_index(inplace=True)
probab_distr_industry = probab_distr_indu.rename(columns = {'industry':'Industry'})
```

In [64]:
```
prob_indus_new=probab_distr_industry[probab_distr_industry['Industry_Values']>=0.04]
prob_indus_new
```

Out[64]:

	Industry	Industry_Values
11	Electronics, Electrical Equip.	0.049861
15	Food & Drug Stores	0.044321
30	Metals	0.055402
32	Motor Vehicles and Parts	0.052632
34	Pharmaceuticals	0.055402
40	Specialty Retailers	0.060942
41	Telecommunications	0.044321

In [65]:
```
#define list of values to be removed
values_ind = prob_indus_new['Industry']
values_ind
```

Out[65]:
```
11    Electronics, Electrical Equip.
15                Food & Drug Stores
30                           Metals
32          Motor Vehicles and Parts
34                   Pharmaceuticals
40                Specialty Retailers
41                 Telecommunications
Name: Industry, dtype: object
```

In [66]:
```
#drop any rows that have above values
data_ind= data_indus[data_indus.industry.isin(values_ind) == False]
data_ind
```

Out[66]:

	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
0	Walmart	Retailing	General Merchandisers	514405.0	2.8	6670.0	-32.4	219295.0	2200000	Bentonville
10	Apple	Technology	Computers, Office Equipment	265595.0	15.9	59531.0	23.1	365725.0	132000	Cupertino
12	Amazon.com	Retailing	Internet Services and Retailing	232887.0	30.9	10073.0	232.1	162648.0	647500	Seattle
13	UnitedHealth Group	Health Care	Health Care: Insurance and Managed Care	226247.0	12.5	11986.0	13.5	152221.0	300000	Minnetonka
16	McKesson	Health Care	Wholesalers: Health Care	214319.0	2.9	34.0	-49.3	59672.0	70000	Irving
...	...	...	...	...	...	...	...	...	...	...
1482	Performance Food Group	Wholesalers	Wholesalers: Food and Grocery	25086.3	27.1	-114.1	-168.4	7719.7	20000	Richmond
1483	Netflix	Media	Entertainment	24996.1	24.0	2761.4	47.9	39280.4	9400	Los Gatos
1484	Nokia	Technology	Network and Other Communications Equipment	24899.2	-4.6	-2874.8	-36793.1	44290.5	92039	Espoo
1489	TD Synnex	Wholesalers	Wholesalers: Electronics and Office Equipment	24675.6	3.9	529.2	5.7	13468.6	277900	Fremont
1491	Holcim	Materials	Building Materials, Glass	24653.8	-8.3	1807.9	-20.0	60235.4	67409	Zug

230 rows × 10 columns

In [67]:
```
freq_ind=data_ind.groupby(['industry']).size()
propor_ind=freq_ind/sum(freq_ind)
propor_ind.head()
```

Out[67]:
```
industry
Aerospace & Defense      0.039130
Aerospace and Defense    0.026087
Airlines                 0.030435
Apparel                  0.021739
Beverages                0.030435
dtype: float64
```

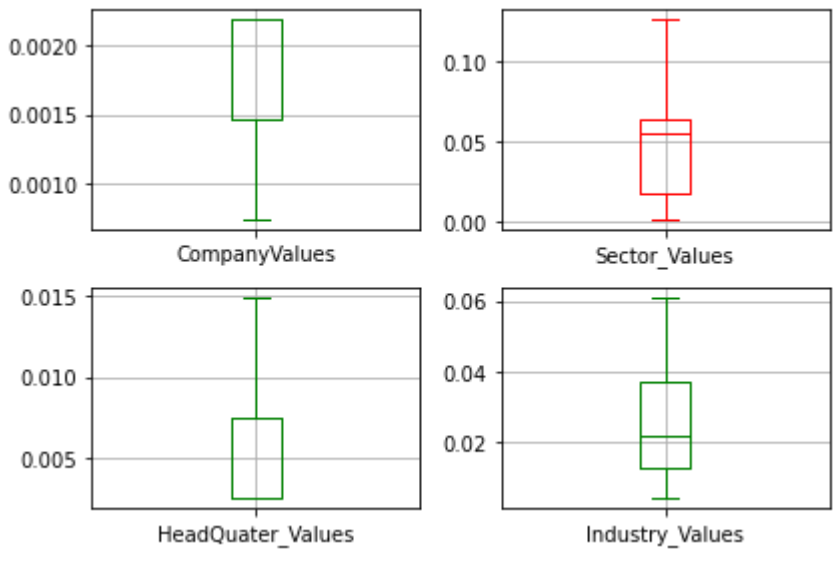
In [68]:
```
probab_distr_ind=pd.DataFrame(propor_ind)
probab_distr_ind.columns=['Industry_Values']
probab_distr_ind.head()
```

Out[68]:

industry	Industry_Values
Aerospace & Defense	0.039130
Aerospace and Defense	0.026087
Airlines	0.030435
Apparel	0.021739
Beverages	0.030435

In [69]:
```
probab_distr_ind.reset_index(inplace=True)
probab_distr_indust = probab_distr_ind.rename(columns = {'industry':'Industry'})
```

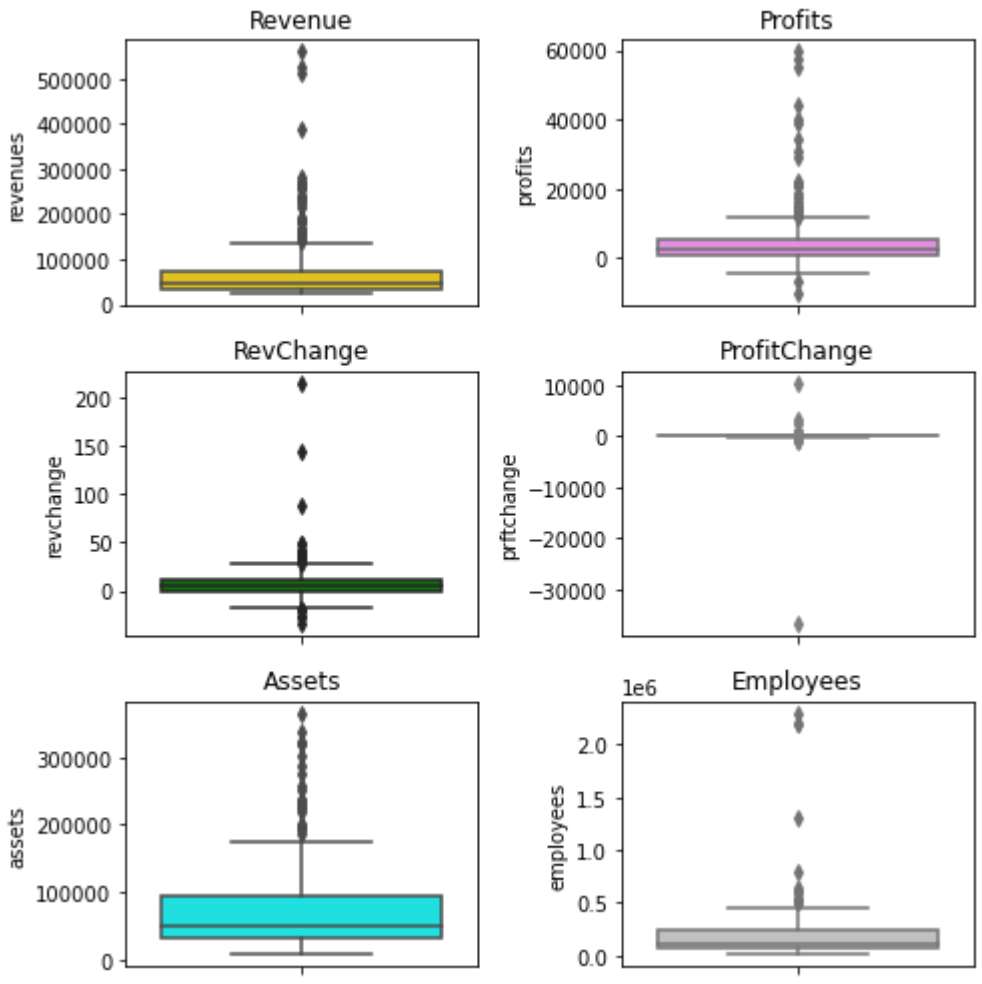
In [70]:
```
# Plotting box plot after removing all outliers
fig=plt.figure()
ax1=fig.add_subplot(221)
ax2=fig.add_subplot(222)
ax3=fig.add_subplot(223)
ax4=fig.add_subplot(224)

propor_names.boxplot( ax=ax1, color='green')
probab_distr_sector.boxplot( ax=ax2, color='red')
probabhqcity.boxplot( ax=ax3, color='green')
probab_distr_indust.boxplot( ax=ax4, color='green')
plt.tight_layout()
```



From the above plot its visible that all the outliers from the box plot has been removed

Performing the above steps with Numerical Data in order to validate the outliers and remove them

In [97]:
```
fig_dims = (7, 7)
fig, axes=plt.subplots(3,2,figsize=fig_dims)
sns.boxplot(data=data_ind, y='revenues',ax=axes[0,0], color='Gold').set(title='Revenue')
sns.boxplot(data=data_ind, y='profits',ax=axes[0,1], color='Violet').set(title='Profits')
sns.boxplot(data=data_ind, y='revchange',ax=axes[1,0], color='Green').set(title='RevChange')
sns.boxplot(data=data_ind, y='prftchange',ax=axes[1,1], color='Wheat').set(title='ProfitChange')
sns.boxplot(data=data_ind, y='assets',ax=axes[2,0], color='Aqua').set(title='Assets')
sns.boxplot(data=data_ind, y='employees',ax=axes[2,1], color='Silver').set(title='Employees')
plt.tight_layout()
```



Removing the oultiers from the numerical data

In [100…
```
# From Revenue
data_rev=data_ind[~(data_ind['revenues'] >55000)]
```

In [101…
```
data_rev.head()
```

Out[101…

	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
191	Anheuser-Busch InBev	Food, Beverages & Tobacco	Beverages	54619.0	-3.2	4368.0	-45.4	232103.0	172603	Leuven
194	Wesfarmers	Food & Drug Stores	Food and Drug Stores	53985.3	4.6	927.4	-57.2	27282.4	217000	Perth
196	Lockheed Martin	Aerospace & Defense	Aerospace and Defense	53762.0	5.3	5046.0	152.0	44876.0	105000	Bethesda
207	Deutsche Bahn	Transportation	Railroads	52004.1	8.1	623.1	-25.8	66896.4	318528	Berlin
209	Alimentation Couche-Tard	Food & Drug Stores	Food and Drug Stores	51394.4	35.6	1673.6	38.4	23140.6	130000	Laval, Quebec

In [102…
```
# For Revenue Change
data_revchng=data_rev[~(data_rev['revchange'] >15)]
data_revchng2=data_revchng[~(data_revchng['revchange'] <-10)]
data_revchng2.head()
```

Out[102…

	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
191	Anheuser-Busch InBev	Food, Beverages & Tobacco	Beverages	54619.0	-3.2	4368.0	-45.4	232103.0	172603	Leuven
194	Wesfarmers	Food & Drug Stores	Food and Drug Stores	53985.3	4.6	927.4	-57.2	27282.4	217000	Perth
196	Lockheed Martin	Aerospace & Defense	Aerospace and Defense	53762.0	5.3	5046.0	152.0	44876.0	105000	Bethesda
207	Deutsche Bahn	Transportation	Railroads	52004.1	8.1	623.1	-25.8	66896.4	318528	Berlin
218	JBS	Food, Beverages & Tobacco	Food Production	49709.7	-2.8	6.9	-95.9	29454.7	230086	São Paulo

In [103…
```
# For Profit
data_profit=data_revchng2[~(data_revchng2['profits'] >8000)]
data_profit2=data_profit[~(data_profit['profits'] <-5)]
data_profit2.head()
```

Out[103…

	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
191	Anheuser-Busch InBev	Food, Beverages & Tobacco	Beverages	54619.0	-3.2	4368.0	-45.4	232103.0	172603	Leuven
194	Wesfarmers	Food & Drug Stores	Food and Drug Stores	53985.3	4.6	927.4	-57.2	27282.4	217000	Perth
196	Lockheed Martin	Aerospace & Defense	Aerospace and Defense	53762.0	5.3	5046.0	152.0	44876.0	105000	Bethesda
207	Deutsche Bahn	Transportation	Railroads	52004.1	8.1	623.1	-25.8	66896.4	318528	Berlin

	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
218	JBS	Food, Beverages & Tobacco	Food Production	49709.7	-2.8	6.9	-95.9	29454.7	230086	São Paulo

In [104…

```
# For Profit Change
data_profit_chng=data_profit2[~(data_profit2['prftchange'] >50)]
data_profit_chng2=data_profit_chng[~(data_profit_chng['prftchange'] <-40)]
data_profit_chng2.head()
```

Out[104…

	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
207	Deutsche Bahn	Transportation	Railroads	52004.1	8.1	623.1	-25.8	66896.4	318528	Berlin
232	Woolworths Group	Food & Drug Stores	Food and Drug Stores	47842.1	3.6	1335.7	15.5	17402.3	201522	Bella Vista
251	SABIC	Chemicals	Chemicals	45096.4	12.9	5738.3	16.8	85231.2	33000	Riyadh
256	American Airlines Group	Transportation	Airlines	44541.0	5.5	1412.0	-26.4	60580.0	128900	Fort Worth
266	Metro	Food & Drug Stores	Food and Drug Stores	43466.5	6.1	409.3	14.2	17702.1	132293	Düsseldorf

In [105…

```
# For Assets
data_assets=data_profit_chng2[~(data_profit_chng2['assets'] >75000)]
data_assets.head()
```

Out[105…

	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
207	Deutsche Bahn	Transportation	Railroads	52004.1	8.1	623.1	-25.8	66896.4	318528	Berlin
232	Woolworths Group	Food & Drug Stores	Food and Drug Stores	47842.1	3.6	1335.7	15.5	17402.3	201522	Bella Vista
256	American Airlines Group	Transportation	Airlines	44541.0	5.5	1412.0	-26.4	60580.0	128900	Fort Worth
266	Metro	Food & Drug Stores	Food and Drug Stores	43466.5	6.1	409.3	14.2	17702.1	132293	Düsseldorf
283	Lufthansa Group	Transportation	Airlines	42302.0	5.5	2552.7	-4.2	43677.5	115882	Cologne

In [107…

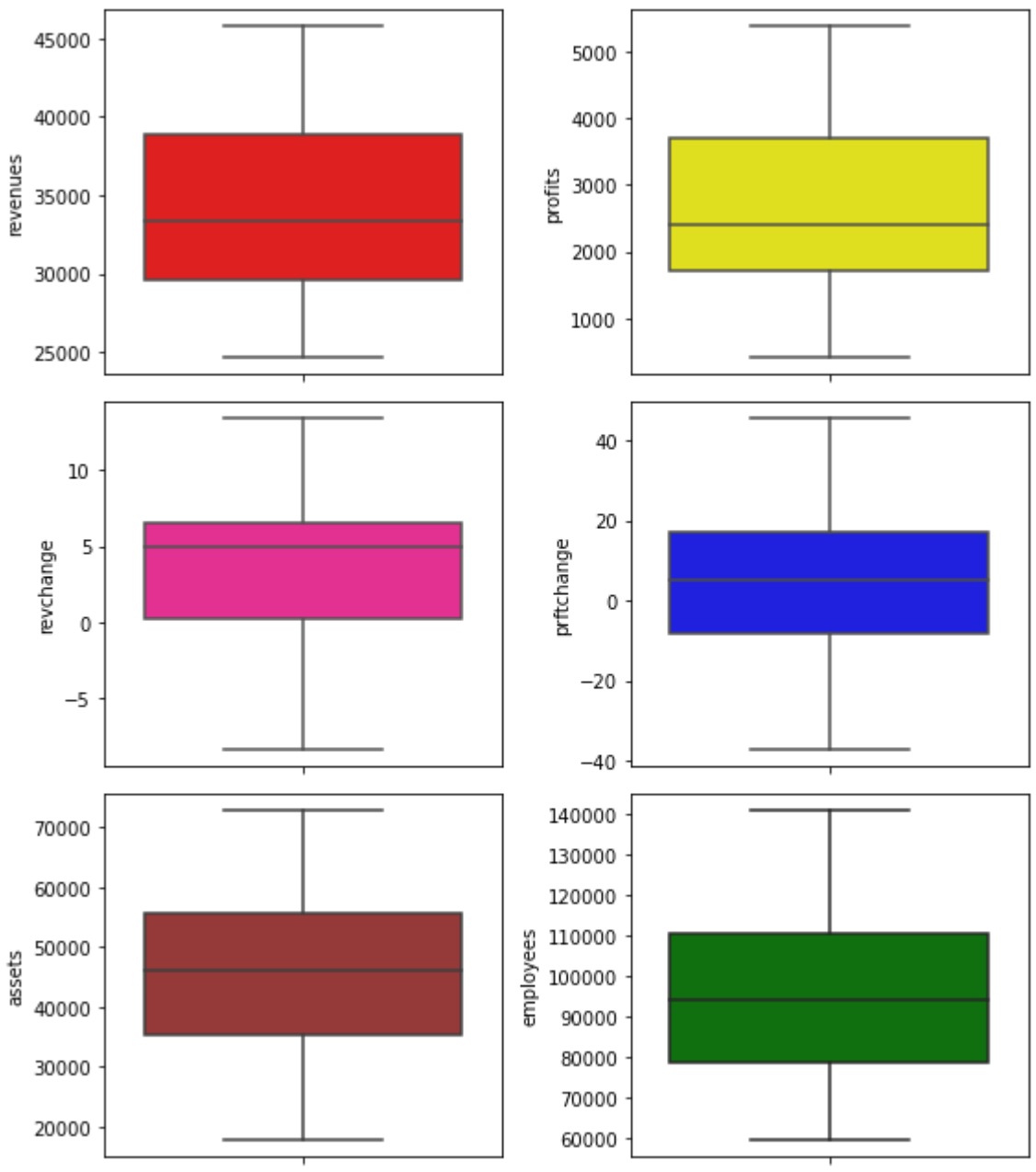
```
# For Employees
data_emp=data_assets[~(data_assets['employees'] >150000)]
final_data=data_emp[~(data_emp['employees'] <40000)]
final_data.head(5)
```

Out[107…

	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
256	American Airlines Group	Transportation	Airlines	44541.0	5.5	1412.0	-26.4	60580.0	128900	Fort Worth
266	Metro	Food & Drug Stores	Food and Drug Stores	43466.5	6.1	409.3	14.2	17702.1	132293	Düsseldorf
283	Lufthansa Group	Transportation	Airlines	42302.0	5.5	2552.7	-4.2	43677.5	115882	Cologne
364	Quanta Computer	Technology	Computers, Office Equipment	34102.6	1.6	501.5	6.2	21455.9	112421	Taoyuan
379	3M	Industrials	Miscellaneous	32765.0	3.5	5349.0	10.1	36500.0	93516	St. Paul

In [129…

```
fig_dims = (8, 9)
fig, axes=plt.subplots(3,2,figsize=fig_dims)

sns.boxplot(data=final_data, y='revenues',ax=axes[0,0], color='red')
sns.boxplot(data=final_data, y='profits',ax=axes[0,1], color='yellow')
sns.boxplot(data=final_data, y='revchange',ax=axes[1,0], color='DeepPink')
sns.boxplot(data=final_data, y='prftchange',ax=axes[1,1], color='blue')
sns.boxplot(data=final_data, y='assets',ax=axes[2,0], color='Brown')
sns.boxplot(data=final_data, y='employees',ax=axes[2,1], color='Green')
plt.tight_layout()
```



From the above box plot it can be seen that now all the outliers are been removed.

(c) Compute the matrix of sample correlation coefficients between every pair of variables.

In [131…
```
#Computing the correlation matrix
correlation=final_data.corr()
correlation
```

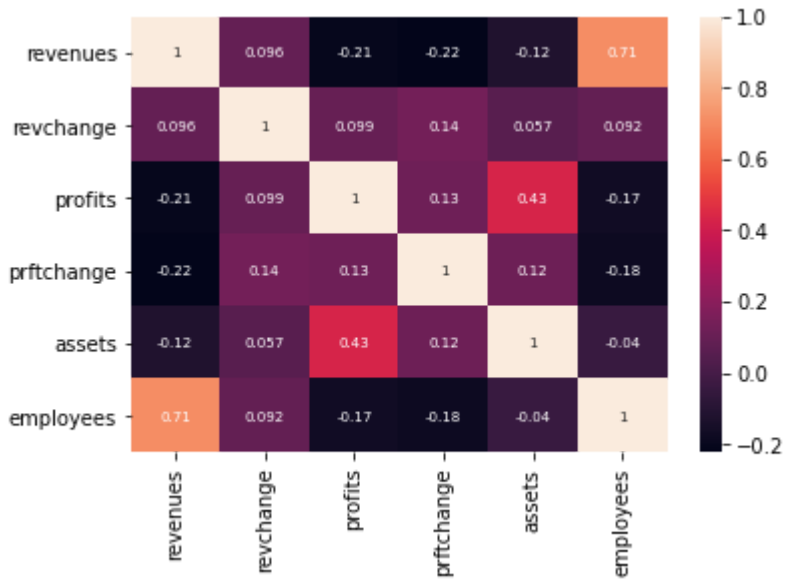
Out[131…

	revenues	revchange	profits	prftchange	assets	employees
revenues	1.000000	0.096410	-0.207680	-0.222908	-0.121076	0.709210
revchange	0.096410	1.000000	0.098674	0.136277	0.057242	0.092304
profits	-0.207680	0.098674	1.000000	0.125054	0.428269	-0.168903
prftchange	-0.222908	0.136277	0.125054	1.000000	0.116492	-0.180512
assets	-0.121076	0.057242	0.428269	0.116492	1.000000	-0.039650
employees	0.709210	0.092304	-0.168903	-0.180512	-0.039650	1.000000

Heatmap is basically defined as a graphical representation of data using colors to visualize the value of the matrix.Plotting the heatmap below in order to visualize the above correlation matrix

In [132…
```
# Plotting heatmap for above correlation data
sns.heatmap(correlation, annot=True, annot_kws={"size": 7})
```

Out[132… <AxesSubplot:>



(d) Choose two variables that are highly correlated (positively or negatively) and plot their scatter plot and the best regression line.

In [134…
```
# Plotting a scatter plot
sns.scatterplot(x=final_data['revenues'],y= final_data['employees'],color='g')
```

```
plt.title('Employees VS Revenues')
m, b = np.polyfit(final_data['revenues'], final_data['employees'], 1)
#Best Regression Line
plt.plot(final_data['revenues'], b+m*final_data['revenues'],color='red')
plt.xlabel('Revenues')
plt.ylabel('Employees')
```

Out[134…

```
Text(0, 0.5, 'Employees')
```

From the above plot I can see that the employees and revenues have positive relationship which is not so strong as it contains lot of residuals points which are far from the regression line.

In [137…

```
clean_data
```

Out[137…

	name	sector	industry	revenues	revchange	profits	prftchange	assets	employees	hqcity
0	Walmart	Retailing	General Merchandisers	514405.0	2.8	6670.0	-32.4	219295.0	2200000	Bentonville
1	Sinopec Group	Energy	Petroleum Refining	414649.9	26.8	5845.0	280.1	329186.3	619151	Beijing
2	Royal Dutch Shell	Energy	Petroleum Refining	396556.0	27.2	23352.0	79.9	399194.0	81000	The Hague
4	State Grid	Energy	Utilities	387056.0	10.9	8174.8	-14.3	572309.5	917717	Beijing
5	Saudi Aramco	Energy	Mining, Crude-Oil Production	355905.0	35.3	110974.5	46.9	358872.9	76418	Dhahran
…	…	…	…	…	…	…	…	…	…	…
1494	Eli Lilly	Health Care	Pharmaceuticals	24539.8	9.9	6193.7	-25.5	46633.1	35000	Indianapolis
1495	Truist Financial	Financials	Banks: Commercial and Savings	24427.0	66.6	4482.0	39.0	509228.0	53638	Charlotte
1496	China Reinsurance (Group)	Financials	Insurance: Property and Casualty (Stock)	24376.0	18.1	827.6	-5.5	69513.7	63914	Beijing
1497	Commonwealth Bank of Australia	Financials	Banks: Commercial and Savings	24362.0	-18.7	6457.1	5.4	698585.9	43585	Sydney
1498	Flex	Technology	Electronics, Electrical Equip.	24124.0	-0.4	613.0	599.9	15836.0	167201	Singapore

1369 rows × 10 columns

(e) Write a brief description of your data and summarize your findings on the variables. Please submit your data as a csv file

The Fortune Global 500, commonly referred to as the Global 500, is an annual ranking of the world's top 500 corporations. measured on the basis of there revenue.Every year,a variety of variables influence the Global 500 rankings, including the global economy, trade policies,mergers and acquisitions, and corporate instability. I have analysed the above dataset with specific columns (name,business sectors,industry, revenues generated, change in revenues,profits made, profit change,assets , number of employees and there headquater) respectively. Below are the observation been computed after cleaning and analysing the dataset:

Financial sector is one of the popular sector among the other sector whereas Hotels, Restaurnats & Leisure being the lowest.

Banks: Commercial and Savings being the popular industry over others industries

Most of the companies HeadQuaters are in Beijing and the second highest headquaters are present in Tokyo

From the box plot it was been seen that in the Sector variable after 0.1 there is an outliers. Moreover, in Industry we can see that there is another outliers after 0.050, whereas in HeadQuater City Variable we have lots of outliers. Moreover there were no outliers for the company values.