Trade&Ahead Project

Context

The stock market has consistently proven to be a good place to invest in and save for the future. There are a lot of compelling reasons to invest in stocks. It can help in fighting inflation, create wealth, and also provides some tax benefits. Good steady returns on investments over a long period of time can also grow a lot more than seems possible. Also, thanks to the power of compound interest, the earlier one starts investing, the larger the corpus one can have for retirement. Overall, investing in stocks can help meet life's financial aspirations.

It is important to maintain a diversified portfolio when investing in stocks in order to maximise earnings under any market condition. Having a diversified portfolio tends to yield higher returns and face lower risk by tempering potential losses when the market is down. It is often easy to get lost in a sea of financial metrics to analyze while determining the worth of a stock, and doing the same for a multitude of stocks to identify the right picks for an individual can be a tedious task. By doing a cluster analysis, one can identify stocks that exhibit similar characteristics and ones which exhibit minimum correlation. This will help investors better analyze stocks across different market segments and help protect against risks that could make the portfolio vulnerable to losses.

Objective

Trade&Ahead is a financial consultancy firm who provide their customers with personalized investment strategies. They have hired you as a Data Scientist and provided you with data comprising stock price and some financial indicators for a few companies listed under the New York Stock Exchange. They have assigned you the tasks of analyzing the data, grouping the stocks based on the attributes provided, and sharing insights about the characteristics of each group.

Data Dictionary

- Ticker Symbol: An abbreviation used to uniquely identify publicly traded shares of a particular stock on a particular stock market
- Company: Name of the company
- GICS Sector: The specific economic sector assigned to a company by the Global Industry Classification Standard (GICS) that best defines its business operations
- GICS Sub Industry: The specific sub-industry group assigned to a company by the Global Industry Classification Standard (GICS) that best defines its business operations
- Current Price: Current stock price in dollars
- Price Change: Percentage change in the stock price in 13 weeks
- Volatility: Standard deviation of the stock price over the past 13 weeks
- ROE: A measure of financial performance calculated by dividing net income by shareholders' equity (shareholders' equity is equal to a company's assets minus its debt)
- Cash Ratio: The ratio of a company's total reserves of cash and cash equivalents to its total current liabilities
- Net Cash Flow: The difference between a company's cash inflows and outflows (in dollars)

- Net Income: Revenues minus expenses, interest, and taxes (in dollars)
- Earnings Per Share: Company's net profit divided by the number of common shares it has outstanding (in dollars)
- Estimated Shares Outstanding: Company's stock currently held by all its shareholders
- P/E Ratio: Ratio of the company's current stock price to the earnings per share
- P/B Ratio: Ratio of the company's stock price per share by its book value per share (book value of a company is the net difference between that company's total assets and total liabilities)

Importing necessary libraries and data

In [1]:
```
# suppress all warnings
import warnings
warnings.filterwarnings("ignore")

#import libraries needed for data manipulation
import pandas as pd
import numpy as np

pd.set_option('display.float_format', lambda x: '%.3f' % x)

#import libraries needed for data visualization

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# unlimited number of displayed columns, limit of 200 for displayed rows
pd.set_option("display.max_columns", None)
pd.set_option("display.max_rows", 200)


# to scale the data using z-score
from sklearn.preprocessing import StandardScaler

# to compute distances
from scipy.spatial.distance import cdist, pdist

# to perform k-means clustering and compute silhouette scores
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# to visualize the elbow curve and silhouette scores
from yellowbrick.cluster import KElbowVisualizer, SilhouetteVisualizer

# to perform hierarchical clustering, compute cophenetic correlation, and create dendrograms
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram, linkage, cophenet
```

Data Overview

- Observations
- Sanity checks

In [2]:

```
#import dataset named 'stock_data.csv'

stock = pd.read_csv('stock_data.csv')

# read first five rows of the dataset

stock.head()
```

Out[2]:

	Ticker Symbol	Security	GICS Sector	GICS Sub Industry	Current Price	Price Change	Volatility	ROE	Cash Ratio	Net Cash Flow	Net Income	Earnings Per Share	Estimated Shares Outstanding	P/E Ratio	I Ra
0	AAL	American Airlines Group	Industrials	Airlines	42.350	10.000	1.687	135	51	-604000000	7610000000	11.390	668129938.500	3.718	-8.7
1	ABBV	AbbVie	Health Care	Pharmaceuticals	59.240	8.339	2.198	130	77	51000000	5144000000	3.150	1633015873.000	18.806	-8.7
2	ABT	Abbott Laboratories	Health Care	Health Care Equipment	44.910	11.301	1.274	21	67	938000000	4423000000	2.940	1504421769.000	15.276	-0.3
3	ADBE	Adobe Systems Inc	Information Technology	Application Software	93.940	13.977	1.358	9	180	-240840000	629551000	1.260	499643650.800	74.556	4.2
4	ADI	Analog Devices, Inc.	Information Technology	Semiconductors	55.320	-1.828	1.701	14	272	315120000	696878000	0.310	2247993548.000	178.452	1.0

In [3]:

```
stock.shape
```

Out[3]:

```
(340, 15)
```

In [4]:

```
stock.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 340 entries, 0 to 339
Data columns (total 15 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Ticker Symbol                 340 non-null    object
 1   Security                      340 non-null    object
 2   GICS Sector                   340 non-null    object
 3   GICS Sub Industry             340 non-null    object
 4   Current Price                 340 non-null    float64
 5   Price Change                  340 non-null    float64
 6   Volatility                    340 non-null    float64
 7   ROE                           340 non-null    int64
```

```
 8   Cash Ratio                    340 non-null    int64
 9   Net Cash Flow                 340 non-null    int64
 10  Net Income                    340 non-null    int64
 11  Earnings Per Share            340 non-null    float64
 12  Estimated Shares Outstanding  340 non-null    float64
 13  P/E Ratio                     340 non-null    float64
 14  P/B Ratio                     340 non-null    float64
dtypes: float64(7), int64(4), object(4)
memory usage: 40.0+ KB
```

In [5]:

```
stock.sample(n=10, random_state=1)
```

Out[5]:

	Ticker Symbol	Security	GICS Sector	GICS Sub Industry	Current Price	Price Change	Volatility	ROE	Cash Ratio	Net Cash Flow	Net Income	Earnings Per Share	Estimated Shares Outstanding	P/E Ratio
102	DVN	Devon Energy Corp.	Energy	Oil & Gas Exploration & Production	32.000	-15.478	2.924	205	70	830000000	-14454000000	-35.550	406582278.500	93.089
125	FB	Facebook	Information Technology	Internet Software & Services	104.660	16.224	1.321	8	958	592000000	3669000000	1.310	2800763359.000	79.893
11	AIV	Apartment Investment & Mgmt	Real Estate	REITs	40.030	7.579	1.163	15	47	21818000	248710000	1.520	163625000.000	26.336
248	PG	Procter & Gamble	Consumer Staples	Personal Products	79.410	10.661	0.806	17	129	160383000	636056000	3.280	491391569.000	24.070
238	OXY	Occidental Petroleum	Energy	Oil & Gas Exploration & Production	67.610	0.865	1.590	32	64	-588000000	-7829000000	-10.230	765298142.700	93.089
336	YUM	Yum! Brands Inc	Consumer Discretionary	Restaurants	52.516	-8.699	1.479	142	27	159000000	1293000000	2.970	435353535.400	17.682
112	EQT	EQT Corporation	Energy	Oil & Gas Exploration & Production	52.130	-21.254	2.365	2	201	523803000	85171000	0.560	152091071.400	93.089
147	HAL	Halliburton Co.	Energy	Oil & Gas Equipment & Services	34.040	-5.102	1.966	4	189	7786000000	-671000000	-0.790	849367088.600	93.089
89	DFS	Discover Financial Services	Financials	Consumer Finance	53.620	3.654	1.160	20	99	2288000000	2297000000	5.140	446887159.500	10.432
173	IVZ	Invesco Ltd.	Financials	Asset Management & Custody Banks	33.480	7.067	1.581	12	67	412000000	968100000	2.260	428362831.900	14.814

In [6]:

```
stock.describe().T
```

Out[6]:

	count	mean	std	min	25%	50%	75%	max
Current Price	340.000	80.862	98.055	4.500	38.555	59.705	92.880	1274.950
Price Change	340.000	4.078	12.006	-47.130	-0.939	4.820	10.695	55.052
Volatility	340.000	1.526	0.592	0.733	1.135	1.386	1.696	4.580
ROE	340.000	39.597	96.548	1.000	9.750	15.000	27.000	917.000
Cash Ratio	340.000	70.024	90.421	0.000	18.000	47.000	99.000	958.000
Net Cash Flow	340.000	55537620.588	1946365312.176	-11208000000.000	-193906500.000	2098000.000	169810750.000	20764000000.000
Net Income	340.000	1494384602.941	3940150279.328	-23528000000.000	352301250.000	707336000.000	1899000000.000	24442000000.000
Earnings Per Share	340.000	2.777	6.588	-61.200	1.558	2.895	4.620	50.090
Estimated Shares Outstanding	340.000	577028337.754	845849595.418	27672156.860	158848216.100	309675137.800	573117457.325	6159292035.000
P/E Ratio	340.000	32.613	44.349	2.935	15.045	20.820	31.765	528.039
P/B Ratio	340.000	-1.718	13.967	-76.119	-4.352	-1.067	3.917	129.065

In [7]:

```
stock.isnull().sum()
```

Out[7]:

```
Ticker Symbol                   0
Security                        0
GICS Sector                     0
GICS Sub Industry               0
Current Price                   0
Price Change                    0
Volatility                      0
ROE                             0
Cash Ratio                      0
Net Cash Flow                   0
Net Income                      0
Earnings Per Share              0
Estimated Shares Outstanding    0
P/E Ratio                       0
P/B Ratio                       0
dtype: int64
```

In [8]:

```
stock.duplicated().sum()
```

Out[8]: 0

In [9]:

```
# create a copy of the data so that the original dataset is not changed.

df = stock.copy()
```

Observations

- 0 null or duplicate values in the dataset.
- Ticker Symbol identifies stock individual is investing in.
- GICS (Global Industry Classification Standard) Sector & Sub Industry, as well as Security and Ticker Symbol are object type. Remaining variables are numeric.
- Net Income averages at ~ 1.5 billion, Net Cash Flow averages at ~ 55 million.

Exploratory Data Analysis (EDA)

- EDA is an important part of any project involving data.
- It is important to investigate and understand the data better before building a model with it.

Leading Questions: *Done within Bivariate analysis section*

1. What does the distribution of stock prices look like?
2. The stocks of which economic sector have seen the maximum price increase on average?
3. How are the different variables correlated with each other?
4. Cash ratio provides a measure of a company's ability to cover its short-term obligations using only cash and cash equivalents. How does the average cash ratio vary across economic sectors?
5. P/E ratios can help determine the relative value of a company's shares as they signify the amount of money an investor is willing to invest in a single share of a company per dollar of its earnings. How does the P/E ratio vary, on average, across economic sectors?

Univariate Analysis

In [10]:
```
# define a function to plot a boxplot and a histogram along the same scale

def histbox(data, feature, figsize=(12, 7), kde=False, bins=None):
    """
    Boxplot and histogram combined
    data: dataframe
    feature: dataframe column
    figsize: size of figure (default (12,7))
    kde: whether to show the density curve (default False)
    bins: number of bins for histogram (default None)
    """
    f2, (box, hist) = plt.subplots(
        nrows=2,                                          # Number of rows of the subplot grid = 2
                                                              # boxplot first then histogram created below
        sharex=True,                                      # x-axis same among all subplots
        gridspec_kw={"height_ratios": (0.25, 0.75)},      # boxplot 1/3 height of histogram
        figsize=figsize,                                  # figsize defined above as (12, 7)
    )
    # defining boxplot inside function, so when using it say histbox(df, 'cost'), df: data and cost: feature
```

```
    sns.boxplot(
        data=data, x=feature, ax=box, showmeans=True, color="chocolate"
    )  # showmeans makes mean val on boxplot have star, ax =
    sns.histplot(
        data=data, x=feature, kde=kde, ax=hist, bins=bins, color = "darkgreen"
    ) if bins else sns.histplot(
        data=data, x=feature, kde=kde, ax=hist, color = "darkgreen"
    )  # For histogram if there are bins in potential graph

    # add vertical line in histogram for mean and median
    hist.axvline(
        data[feature].mean(), color="purple", linestyle="--"
    )  # Add mean to the histogram
    hist.axvline(
        data[feature].median(), color="black", linestyle="-"
    )  # Add median to the histogram
```

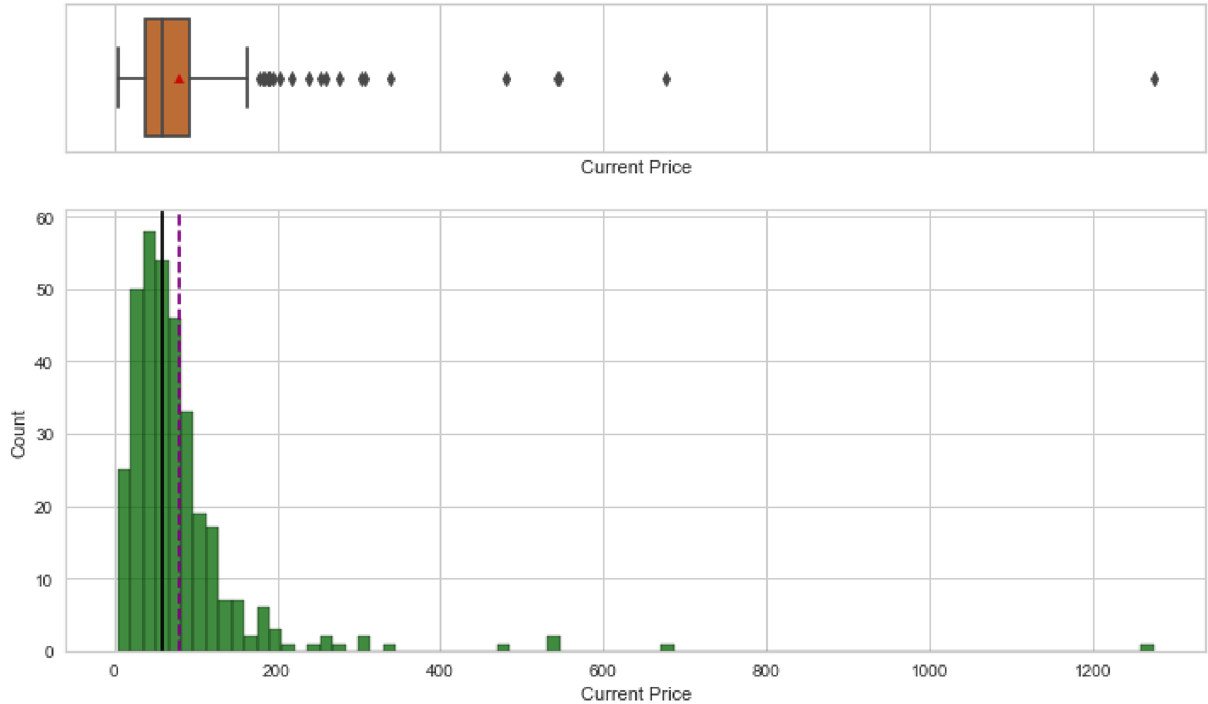
In [11]:

```
# define a function to create labeled barplots

def bar(data, feature, perc=False, n=None):
    """
    Barplot with percentage at the top

    data: dataframe
    feature: dataframe column
    perc: whether to display percentages instead of count (default is False)
    n: displays the top n category levels (default is None, i.e., display all levels)
    """

    total = len(data[feature])  # length of the column
    count = data[feature].nunique()
    if n is None:
        plt.figure(figsize=(count + 1, 5))
    else:
        plt.figure(figsize=(n + 1, 5))

    plt.xticks(rotation=90, fontsize=15)
    ax = sns.countplot(
        data=data,
        x=feature,
        palette="Paired",
        order=data[feature].value_counts().index[:n].sort_values(),
    )

    for p in ax.patches:
        if perc == True:
            label = "{:.1f}%".format(
                100 * p.get_height() / total
            )  # percentage of each class of the category
        else:
            label = p.get_height()  # count of each level of the category
```

```
        x = p.get_x() + p.get_width() / 2  # width of the plot
        y = p.get_height()  # height of the plot

        ax.annotate(
            label,
            (x, y),
            ha="center",
            va="center",
            size=12,
            xytext=(0, 5),
            textcoords="offset points",
        )  # annotate the percentage
plt.show()  # show the plot
```

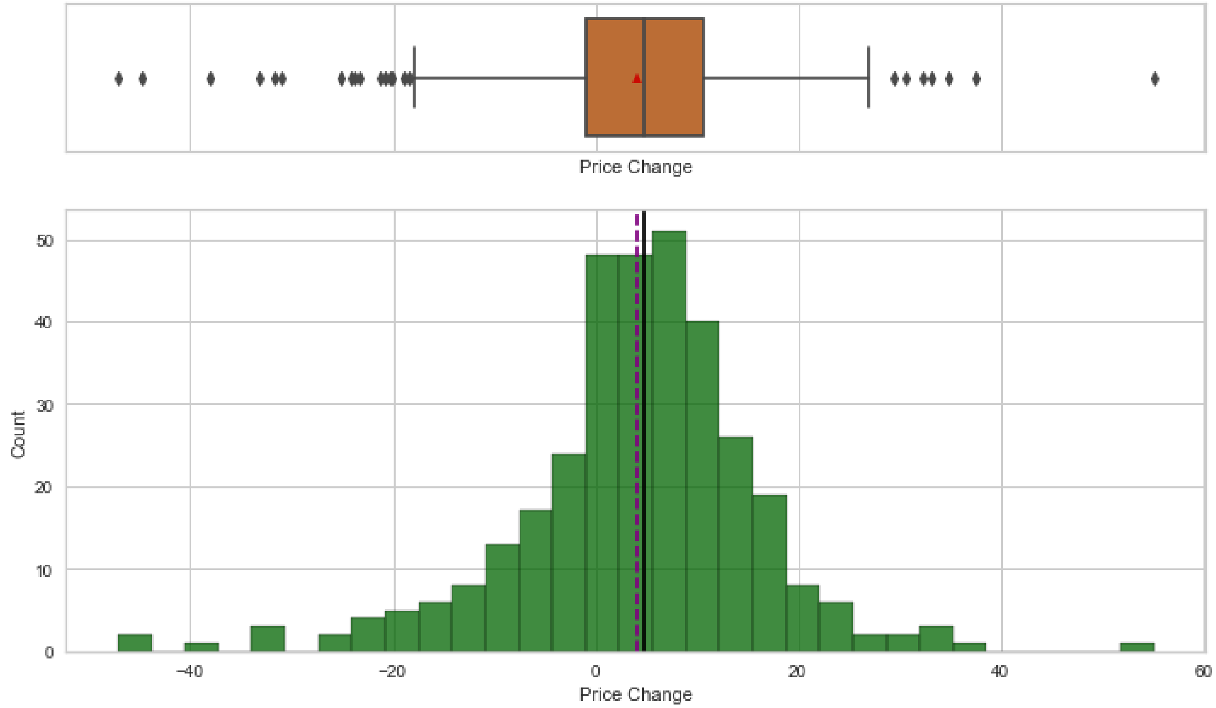
In [12]:

```
histbox(df, 'Current Price')
```

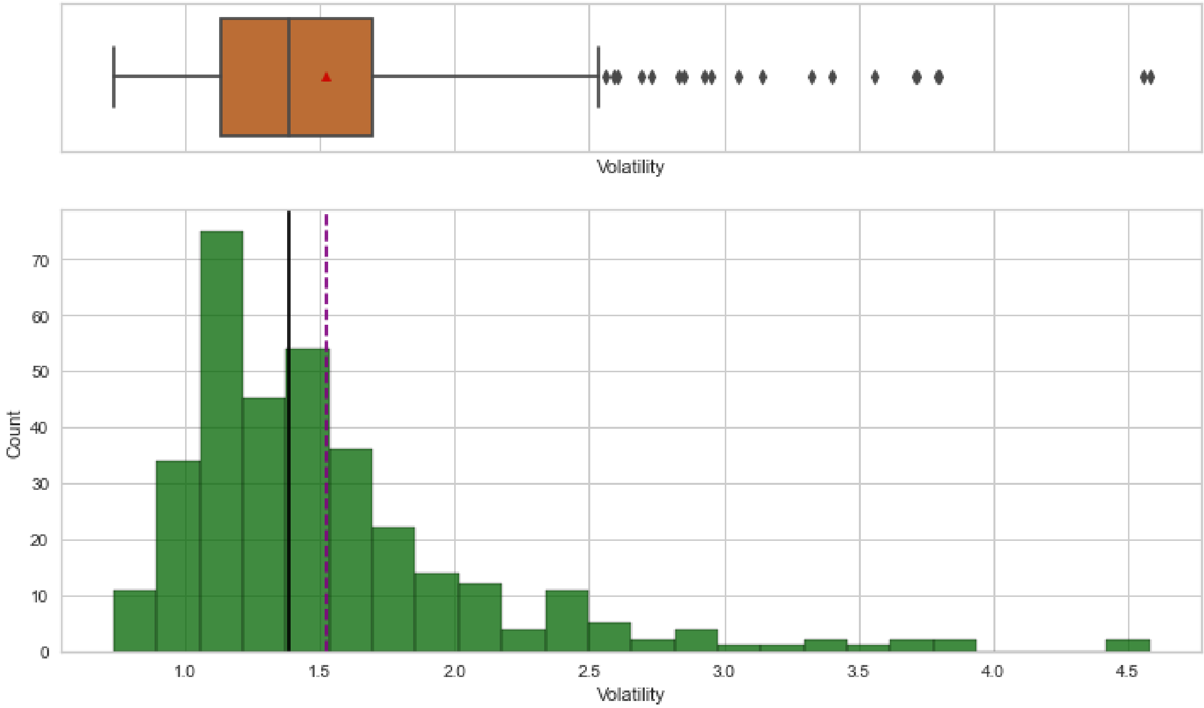


In [13]:

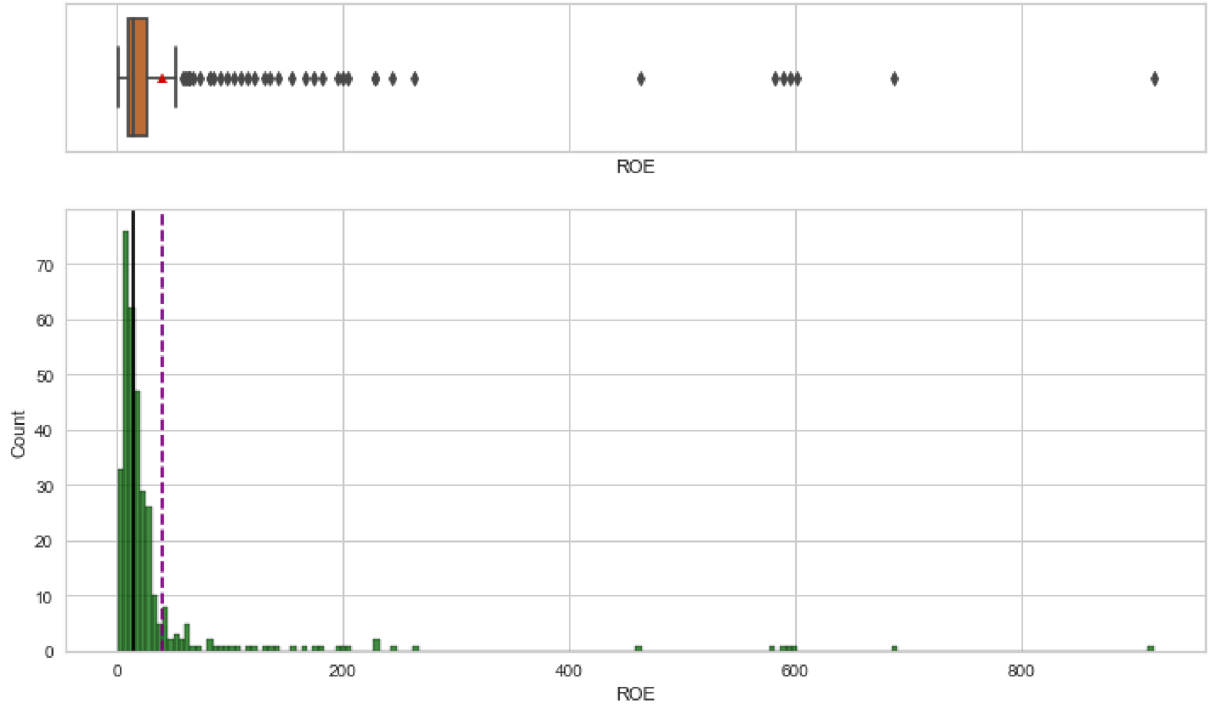
```
histbox(df, 'Price Change')
```



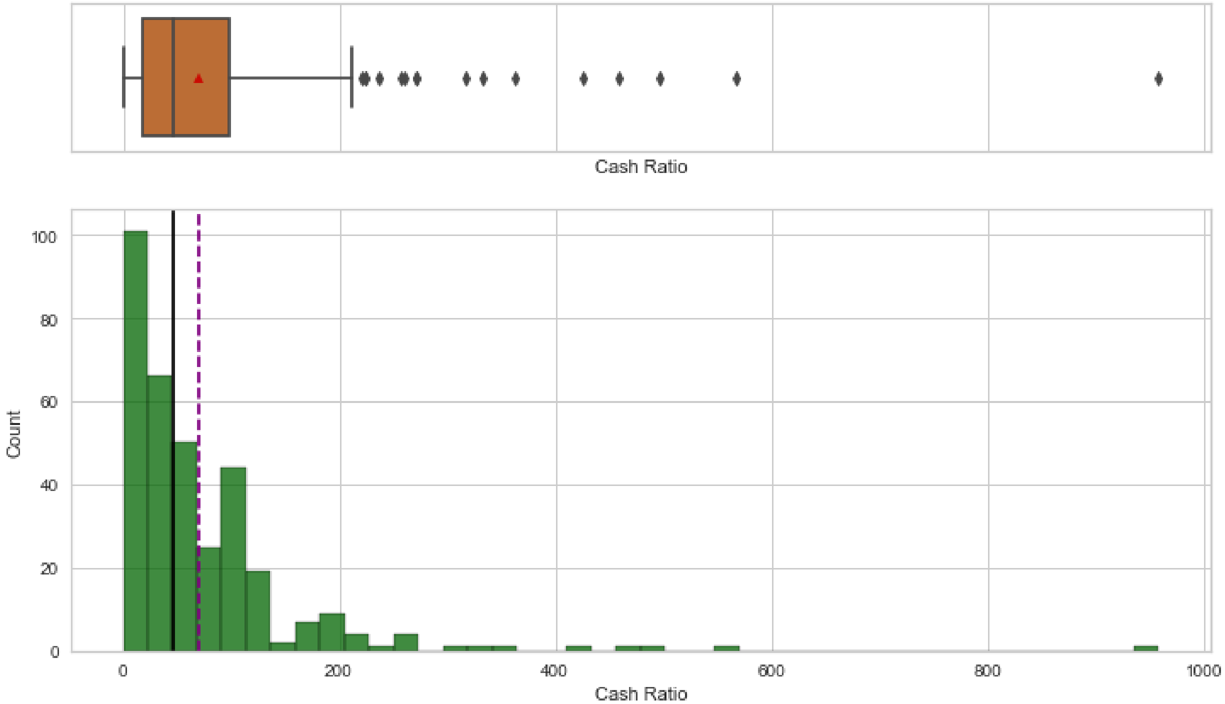
In [14]:
```
histbox(df, 'Volatility')
```



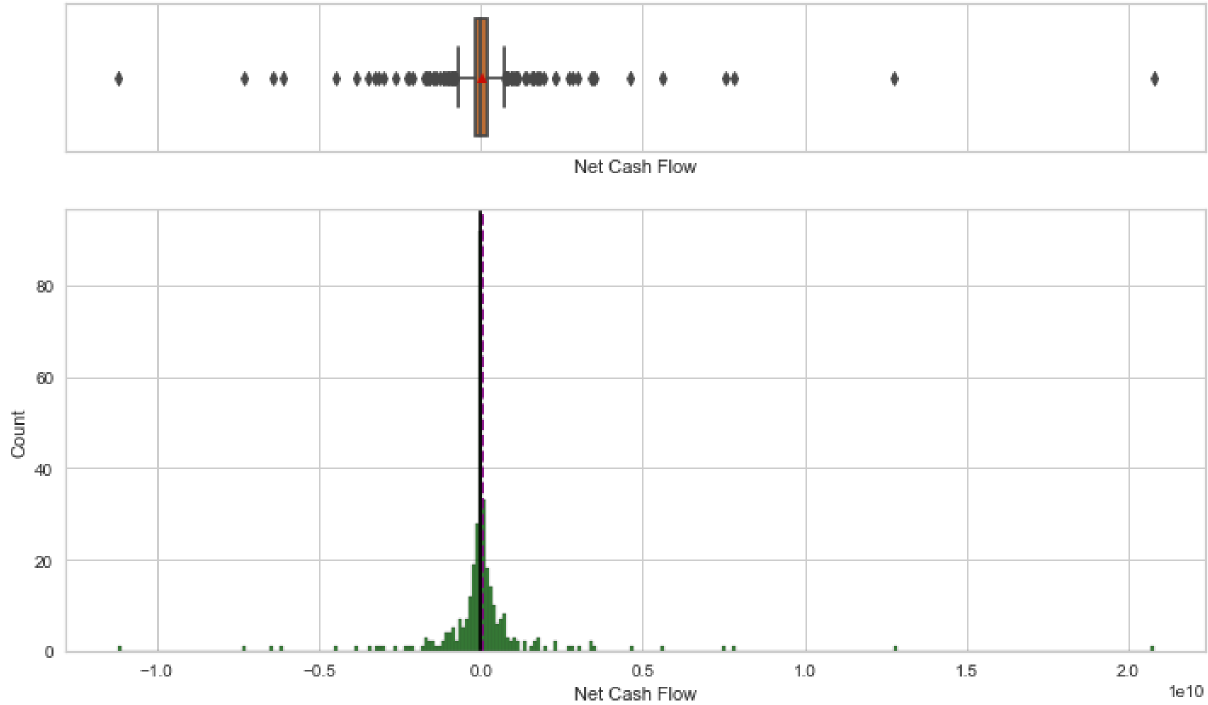
In [15]:
```
histbox(df, 'ROE')
```



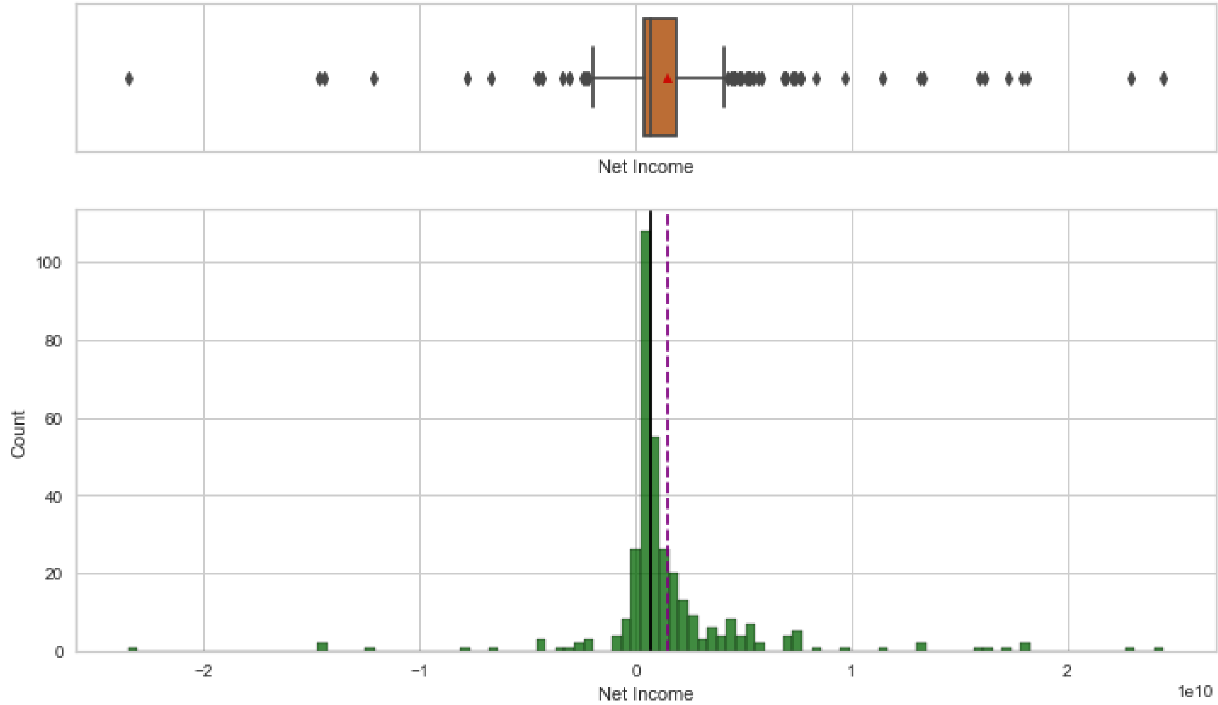
In [16]:
```
histbox(df, 'Cash Ratio')
```



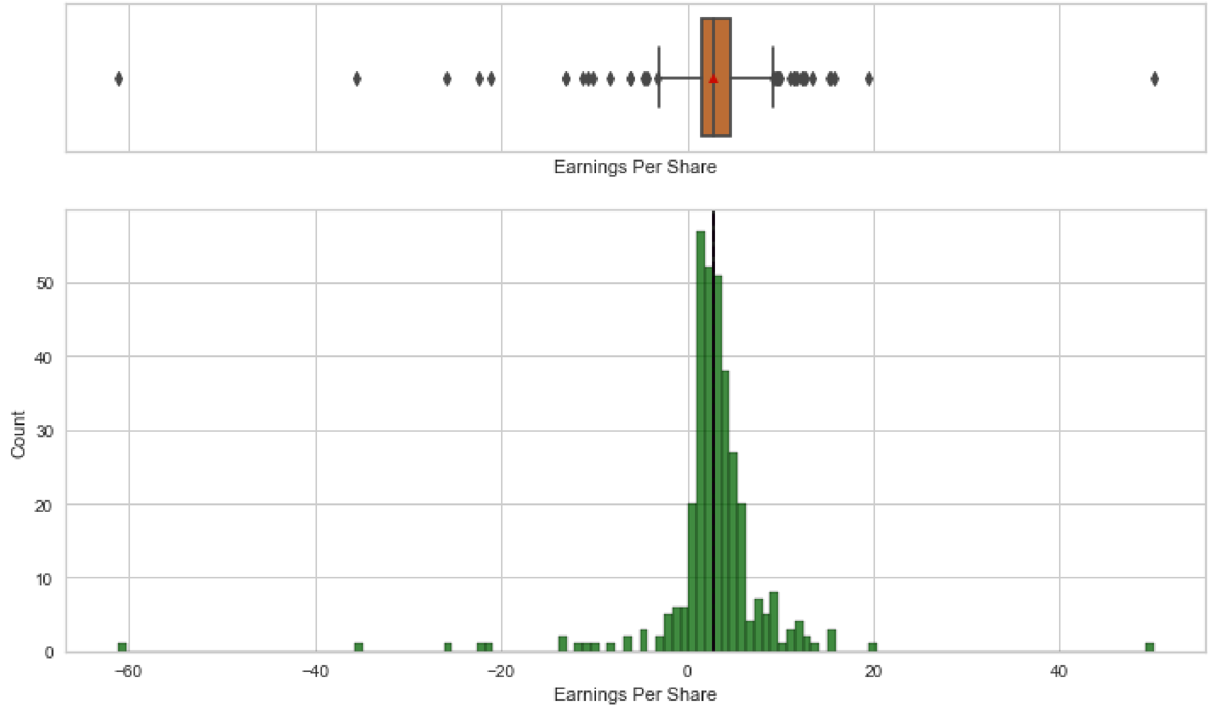
In [17]:
```
histbox(df, 'Net Cash Flow')
```



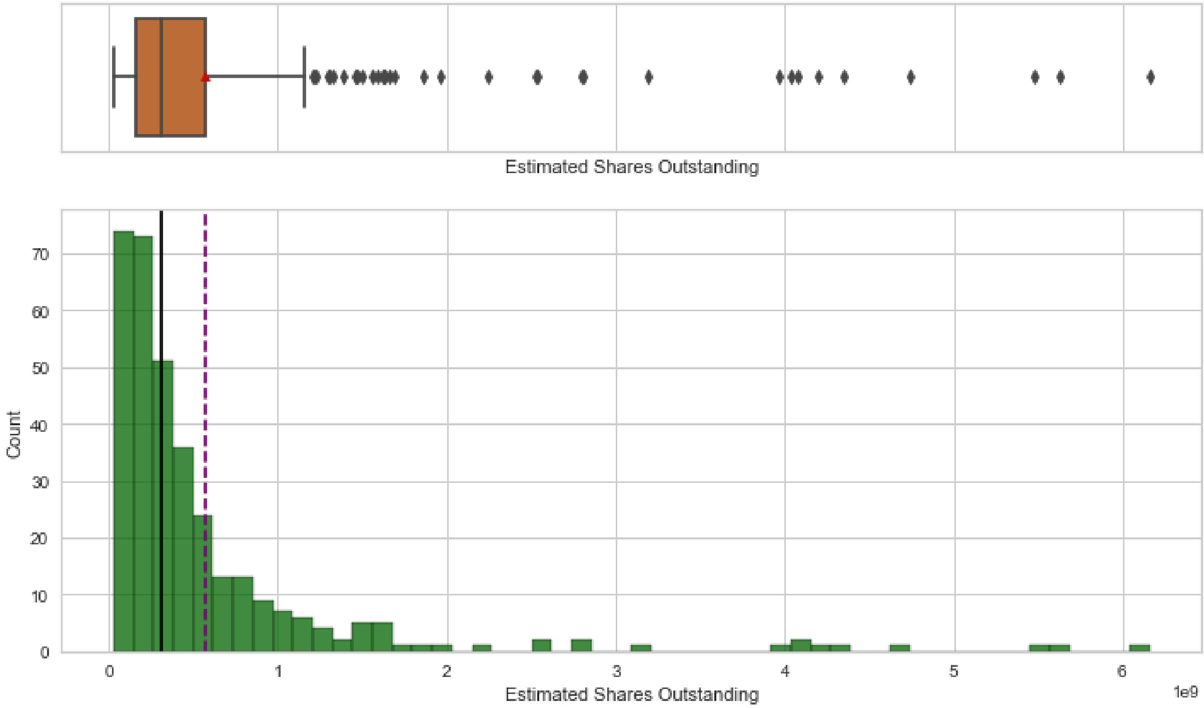
In [18]:
```
histbox(df, 'Net Income')
```



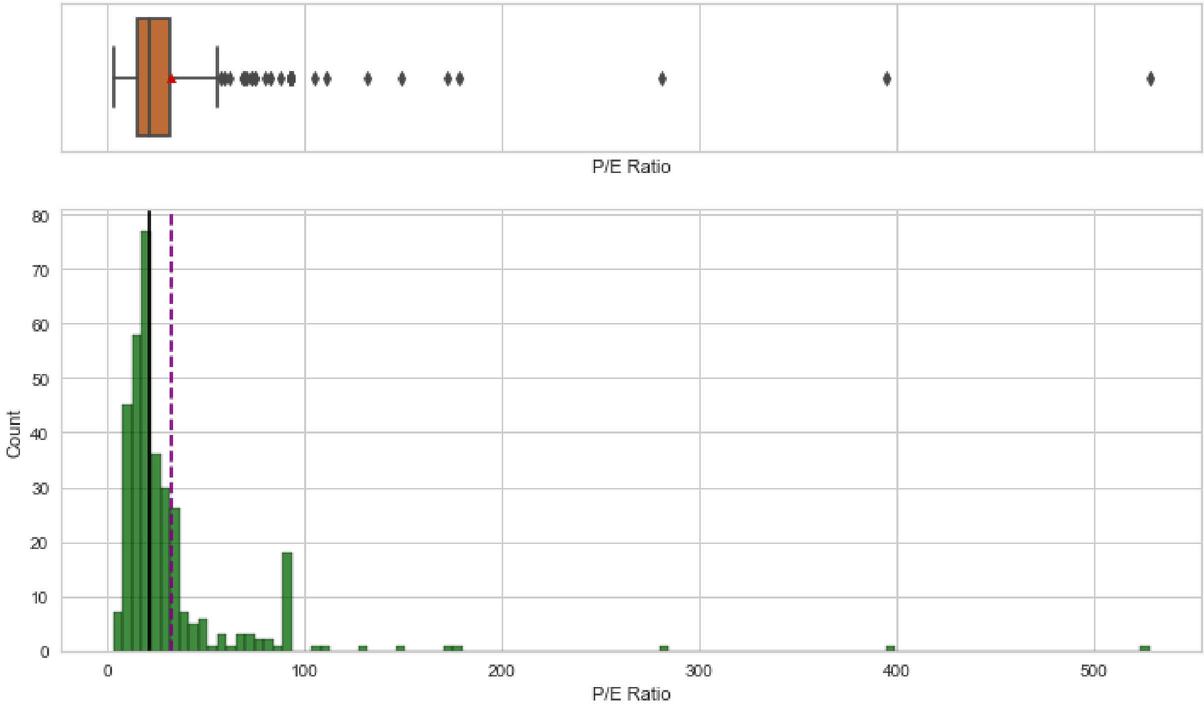
In [19]:
```
histbox(df, 'Earnings Per Share')
```



In [20]:
```
histbox(df, 'Estimated Shares Outstanding')
```

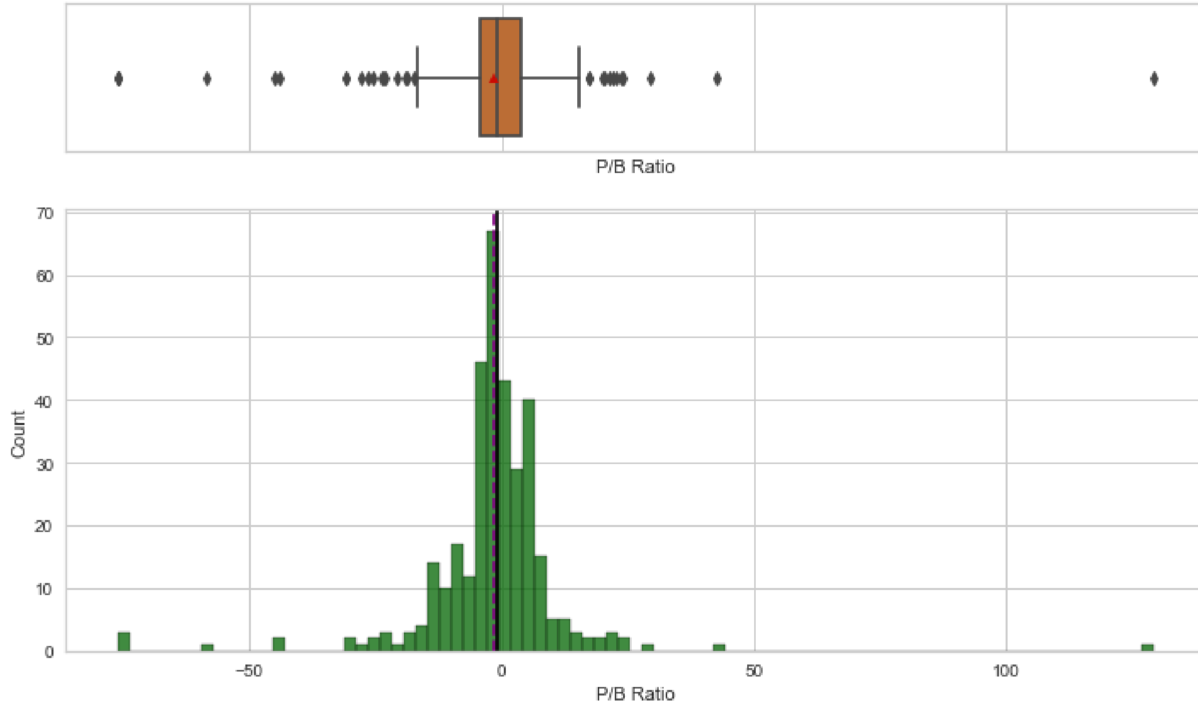


In [21]:
```
histbox(df, 'P/E Ratio')
```

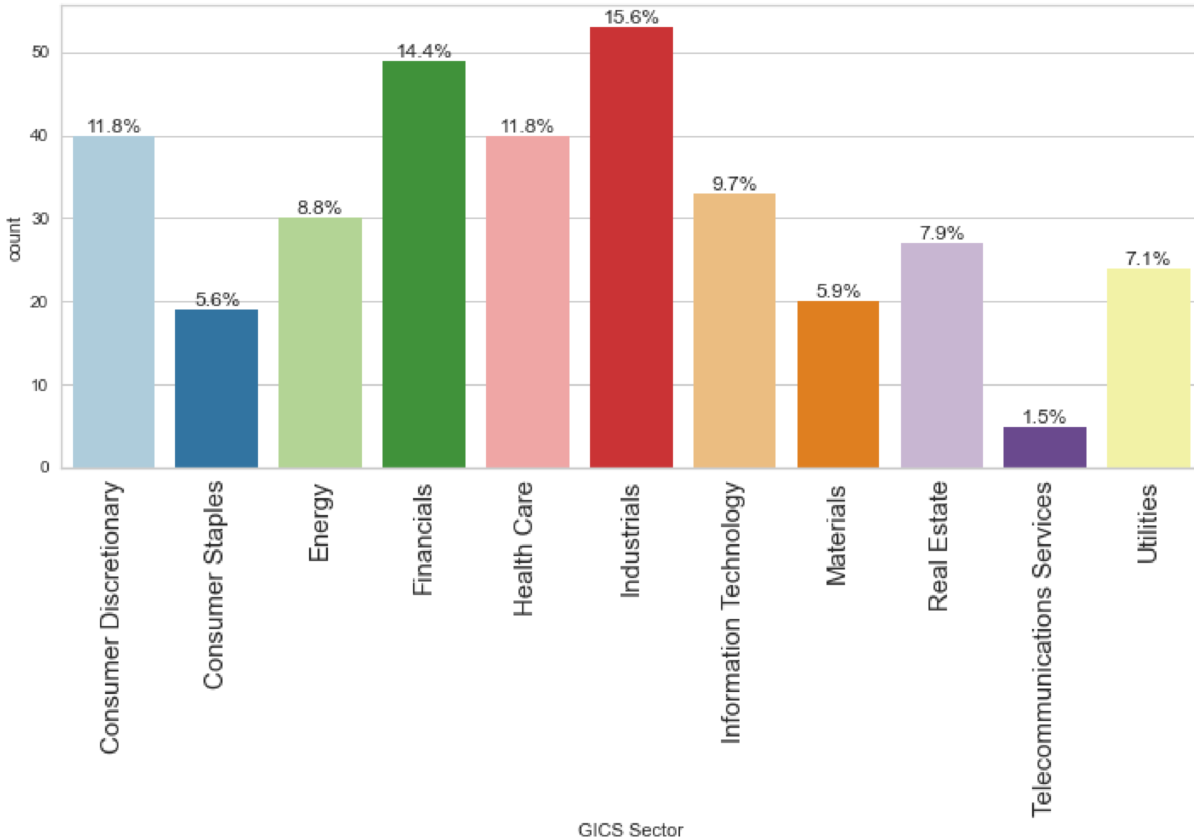


In [22]:

```
histbox(df, 'P/B Ratio')
```

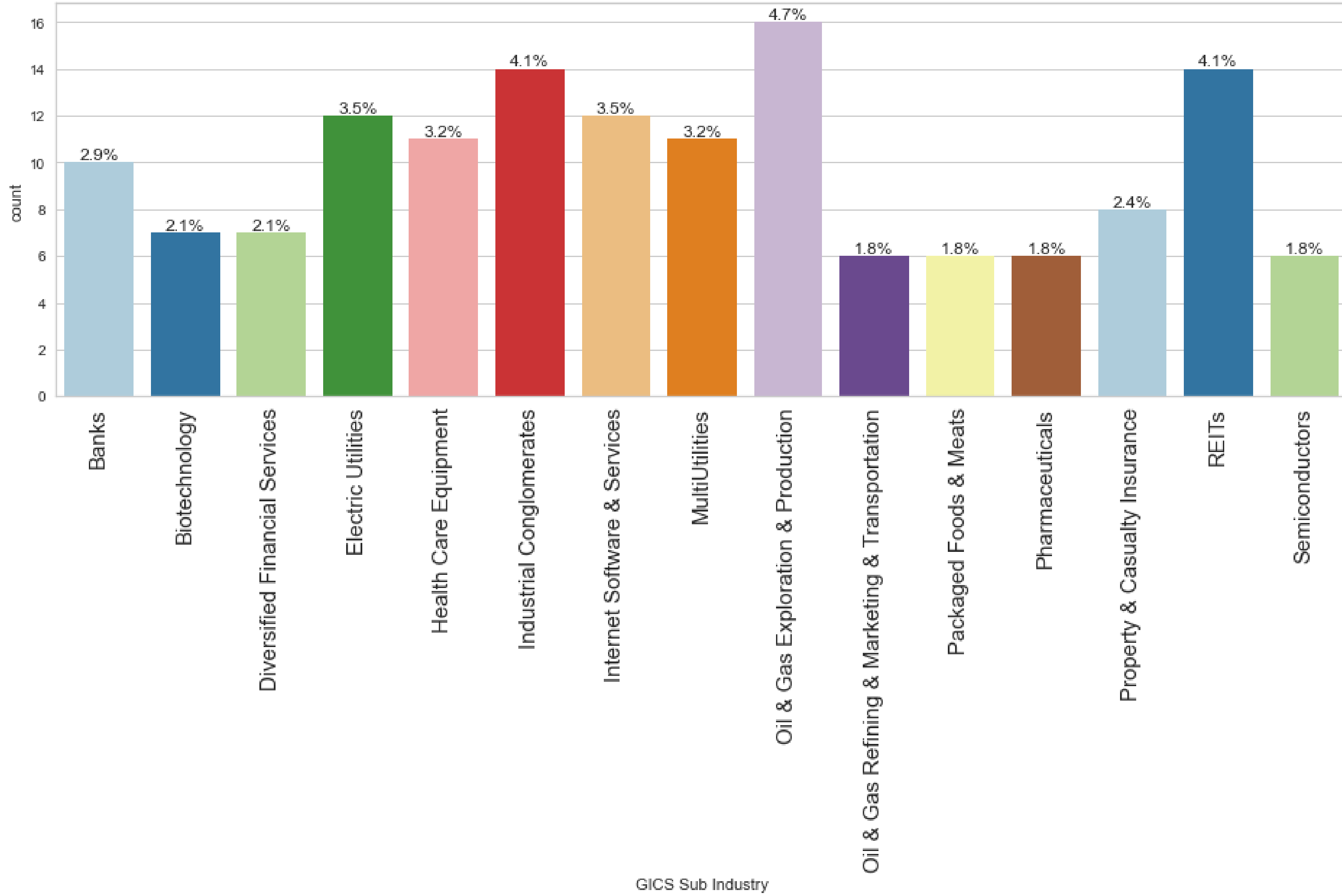


In [23]:
```
bar(df,'GICS Sector', perc=True)
```



In [24]:

```
bar(df, 'GICS Sub Industry', perc=True, n=15)
```



Observations

- Current Price: right skewed distribution, average around 80
- Price Change: near normal distribution, average around 4
- Volatility: right skew, high mode around 1.0, mean 1.5
- ROE: heavy right skew, average around 40
- Cash Ratio: right skew, average around 70
- Net Cash Flow/Net Income: similar normal distributions
- Earnings Per Share: near normal distribution, average 2.77
- Estimated Shares Outstanding: right skew, average around 577 million

- P/E Ratio: average at 32.6
- P/B Ratio: average at -1.72

- GICS most common sector is Industrials

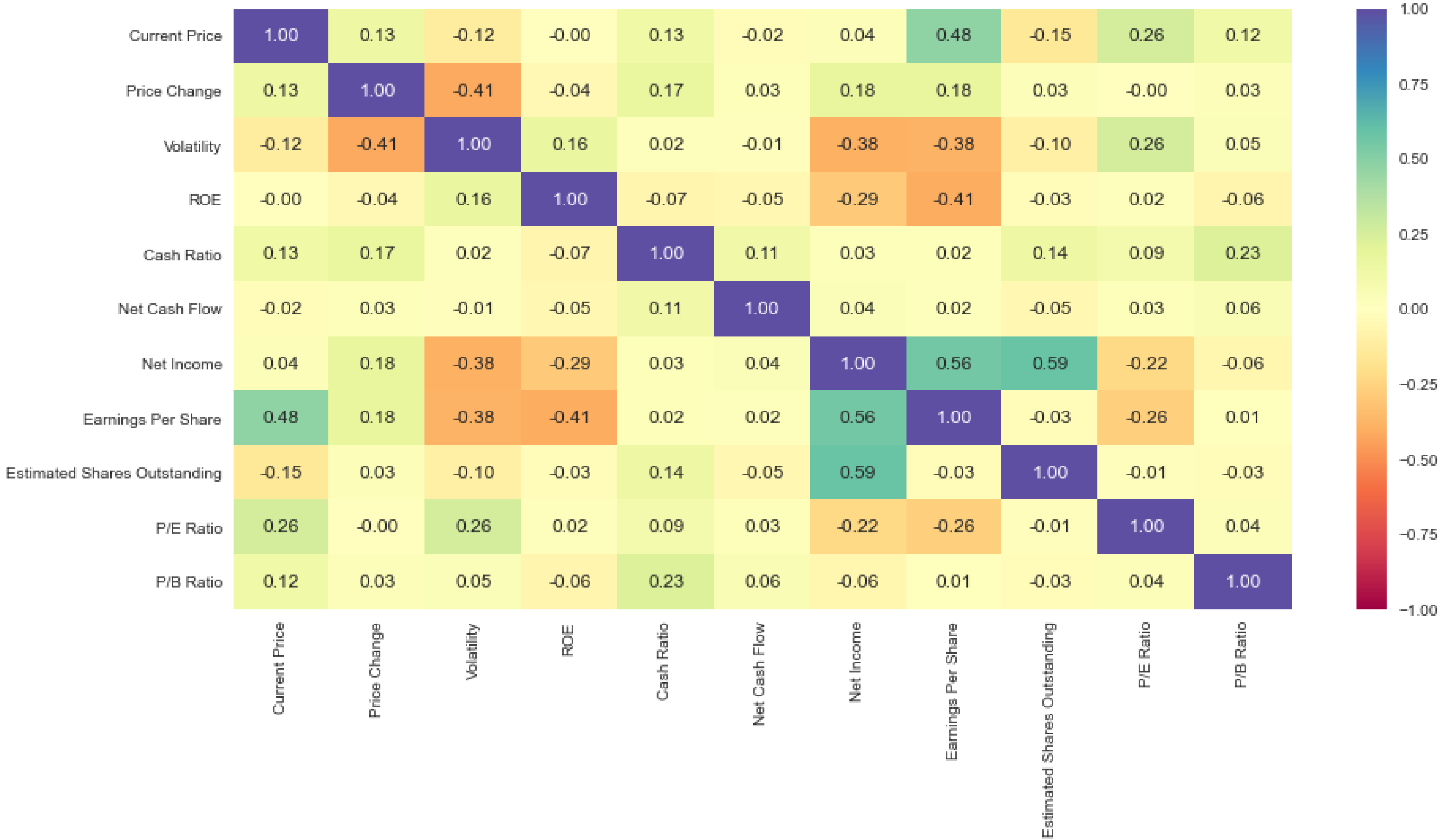
- GICS most common sub-industry is Oil & Gas Exploration & Production

Bivariate Analysis

Leading Question 3: How are the different variables correlated with each other?

In [25]:

```
# correlation check
plt.figure(figsize=(15, 7))
sns.heatmap(
    df.corr(), annot=True, vmin=-1, vmax=1, fmt=".2f", cmap="Spectral"
)
plt.show()
```



- Medium/high correlations:
  - Current Price - Earnings Per Share
  - Net Income - Estimated Shares Outstanding
  - Net Income - Earnings Per Share

Leading Question 1: What does the distribution of stock prices look like?

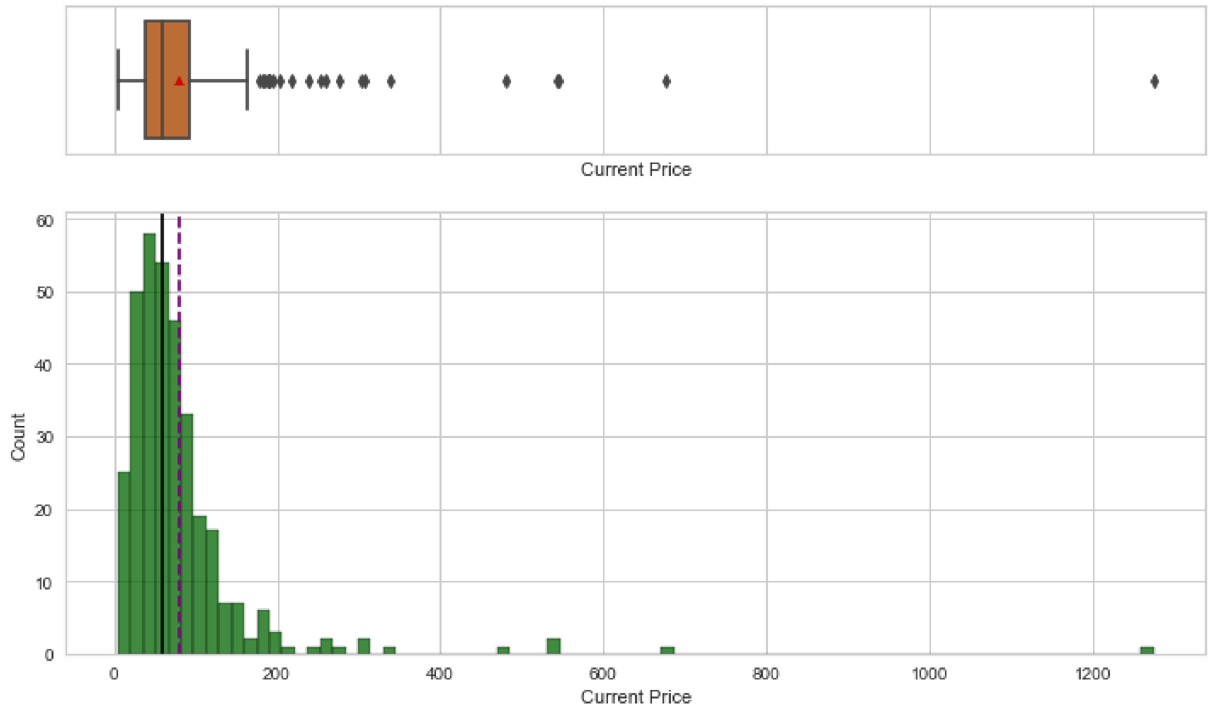
In [26]:
```
df['Current Price'].describe()
```

Out[26]:
```
count    340.000
mean      80.862
std       98.055
min        4.500
25%       38.555
50%       59.705
75%       92.880
```

```
max      1274.950
Name: Current Price, dtype: float64
```

In [27]:

```
histbox(df, 'Current Price')
```



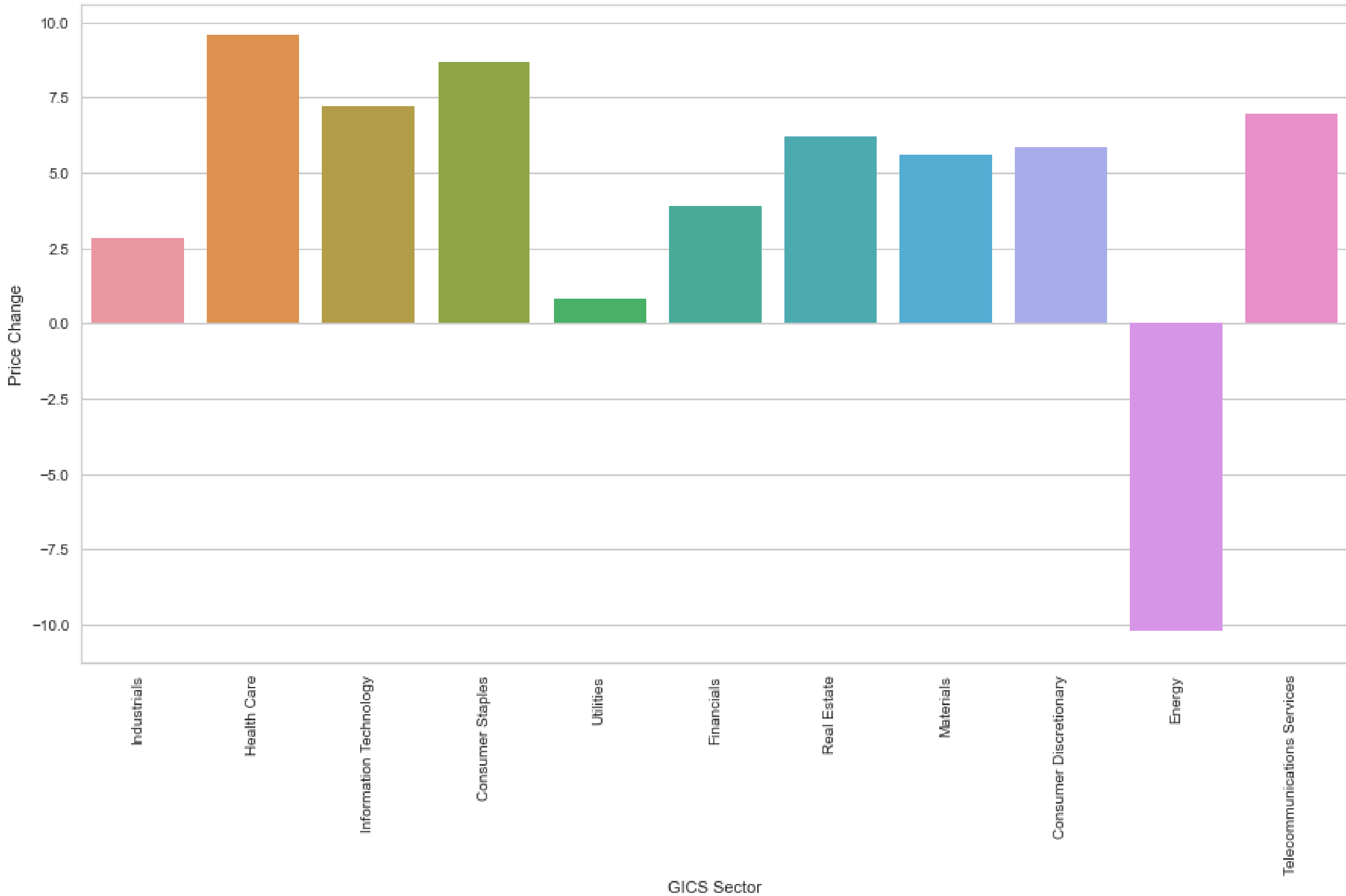
Observations

- Majority of prices fall between 38 and 92 dollars.
- Many outliers, highest being over 1200 dollars.

Leading Question 2: The stocks of which economic sector have seen the maximum price increase on average?

In [28]:

```
plt.figure(figsize=(15,8))
sns.barplot(data=df, x='GICS Sector', y='Price Change', ci=False)
plt.xticks(rotation=90)
plt.show()
```



In [29]:

```
df.groupby("GICS Sector")["Price Change"].describe()
```

Out[29]:

GICS Sector	count	mean	std	min	25%	50%	75%	max
Consumer Discretionary	40.000	5.846	13.291	-33.131	1.228	3.544	12.249	34.804
Consumer Staples	19.000	8.685	8.795	-12.017	5.427	6.977	12.605	24.496
Energy	30.000	-10.228	16.939	-47.130	-20.668	-9.245	2.959	17.342
Financials	49.000	3.865	6.024	-14.293	-0.362	3.910	7.697	15.463
Health Care	40.000	9.586	9.849	-12.532	1.528	10.324	16.776	33.177

GICS Sector	count	mean	std	min	25%	50%	75%	max
Industrials	53.000	2.833	9.922	-23.244	-2.798	3.953	10.105	20.433
Information Technology	33.000	7.217	14.463	-23.791	-1.828	7.497	14.035	55.052
Materials	20.000	5.590	15.284	-31.685	-1.307	4.906	15.450	37.490
Real Estate	27.000	6.206	5.624	-13.067	4.198	7.579	9.140	15.574
Telecommunications Services	5.000	6.957	10.589	-2.301	0.159	5.942	6.277	24.708
Utilities	24.000	0.804	4.471	-8.231	-2.188	1.282	3.661	8.597

In [30]:

```
# Check volatility (standard deviation of stock price) to better explain certain sector price changes behavior

df.groupby("GICS Sector")["Volatility"].describe()
```

Out[30]:

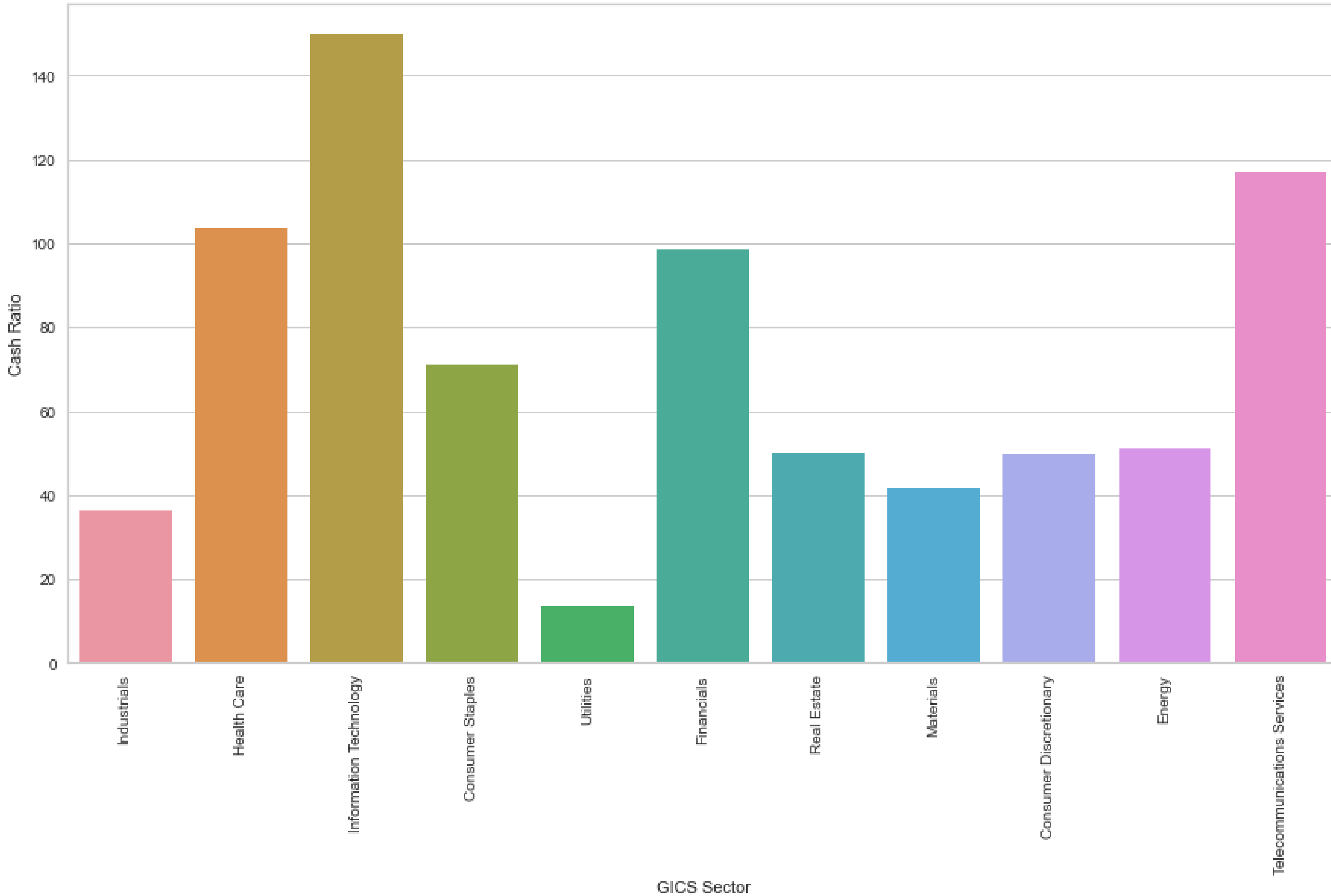
GICS Sector	count	mean	std	min	25%	50%	75%	max
Consumer Discretionary	40.000	1.595	0.486	0.733	1.347	1.558	1.696	3.795
Consumer Staples	19.000	1.153	0.300	0.805	0.893	1.078	1.404	1.718
Energy	30.000	2.569	0.847	1.370	1.956	2.413	3.023	4.580
Financials	49.000	1.267	0.280	0.900	1.081	1.189	1.438	2.231
Health Care	40.000	1.541	0.381	1.007	1.246	1.493	1.700	2.457
Industrials	53.000	1.417	0.416	0.826	1.143	1.349	1.552	2.954
Information Technology	33.000	1.660	0.546	0.904	1.273	1.578	1.886	3.400
Materials	20.000	1.817	0.674	1.079	1.400	1.579	2.167	3.796
Real Estate	27.000	1.206	0.140	0.960	1.112	1.169	1.300	1.595
Telecommunications Services	5.000	1.342	0.499	0.843	0.859	1.457	1.522	2.027
Utilities	24.000	1.118	0.127	0.890	1.039	1.112	1.190	1.390

Observations

- Highest postive percentage change was in the Health Care sector, well over 9.5 in the 13 week period.
- Only negative percentage change was in Energy sector, with the highest volatility (standard dev).
- Lowest positive percentage change was in Utilities sector.

Leading Question 4: Cash ratio provides a measure of a company's ability to cover its short-term obligations using only cash and cash equivalents. How does the average cash ratio vary across economic sectors?

In [31]:
```
plt.figure(figsize=(15,8))
sns.barplot(data=df, x='GICS Sector', y='Cash Ratio', ci=False)
plt.xticks(rotation=90)
plt.show()
```



In [32]:
```
df.groupby("GICS Sector")["Cash Ratio"].describe()
```

Out[32]:

GICS Sector	count	mean	std	min	25%	50%	75%	max
Consumer Discretionary	40.000	49.575	69.208	0.000	11.500	25.000	35.500	260.000
Consumer Staples	19.000	70.947	125.833	9.000	18.000	33.000	63.000	568.000

	count	mean	std	min	25%	50%	75%	max
GICS Sector								
Energy	30.000	51.133	55.939	0.000	7.000	38.500	68.500	201.000
Financials	49.000	98.592	17.907	51.000	99.000	99.000	99.000	183.000
Health Care	40.000	103.775	104.118	3.000	41.500	70.000	128.250	425.000
Industrials	53.000	36.189	29.127	1.000	15.000	31.000	44.000	130.000
Information Technology	33.000	149.818	174.231	16.000	45.000	126.000	180.000	958.000
Materials	20.000	41.700	50.396	2.000	10.000	25.000	49.500	198.000
Real Estate	27.000	50.111	28.251	12.000	47.000	47.000	47.000	164.000
Telecommunications Services	5.000	117.000	213.083	3.000	11.000	14.000	61.000	496.000
Utilities	24.000	13.625	17.277	0.000	3.000	8.500	14.250	74.000

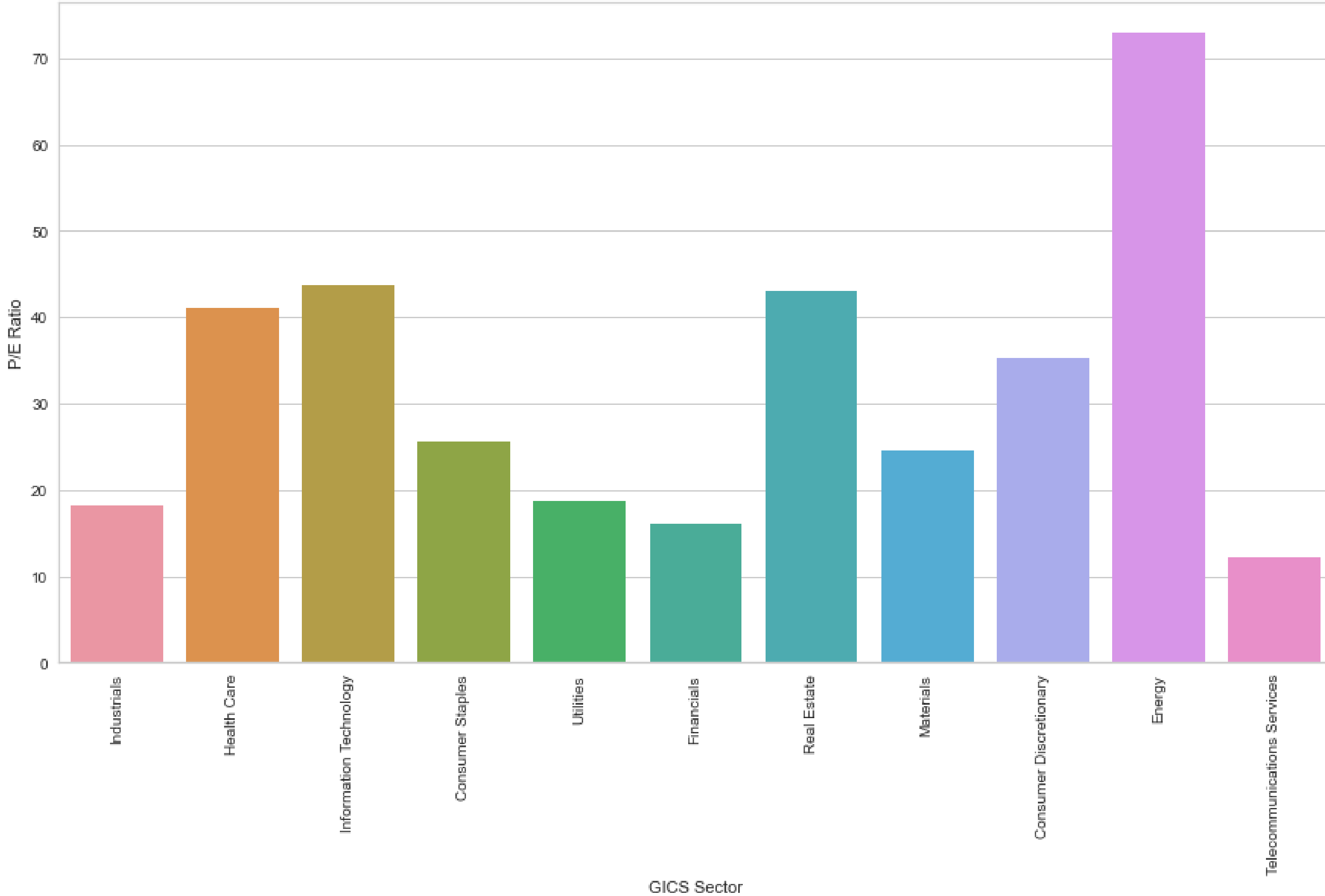
Observations

- Cash Ratio: Company's total reserves of cash and cash equivalents : total current liabilities
- Highest is in IT Sector, lowest is in Utilities Sector

Leading Question 5: P/E ratios can help determine the relative value of a company's shares as they signify the amount of money an investor is willing to invest in a single share of a company per dollar of its earnings. How does the P/E ratio vary, on average, across economic sectors?

In [33]:

```
plt.figure(figsize=(15,8))
sns.barplot(data=df, x='GICS Sector', y='P/E Ratio', ci=False)
plt.xticks(rotation=90)
plt.show()
```



Observations

- P/E Ratio: Stock price : Earnings Per Share
- Highest by a large margin is in Energy sector, lowest is Telecommunications Services

Data Preprocessing

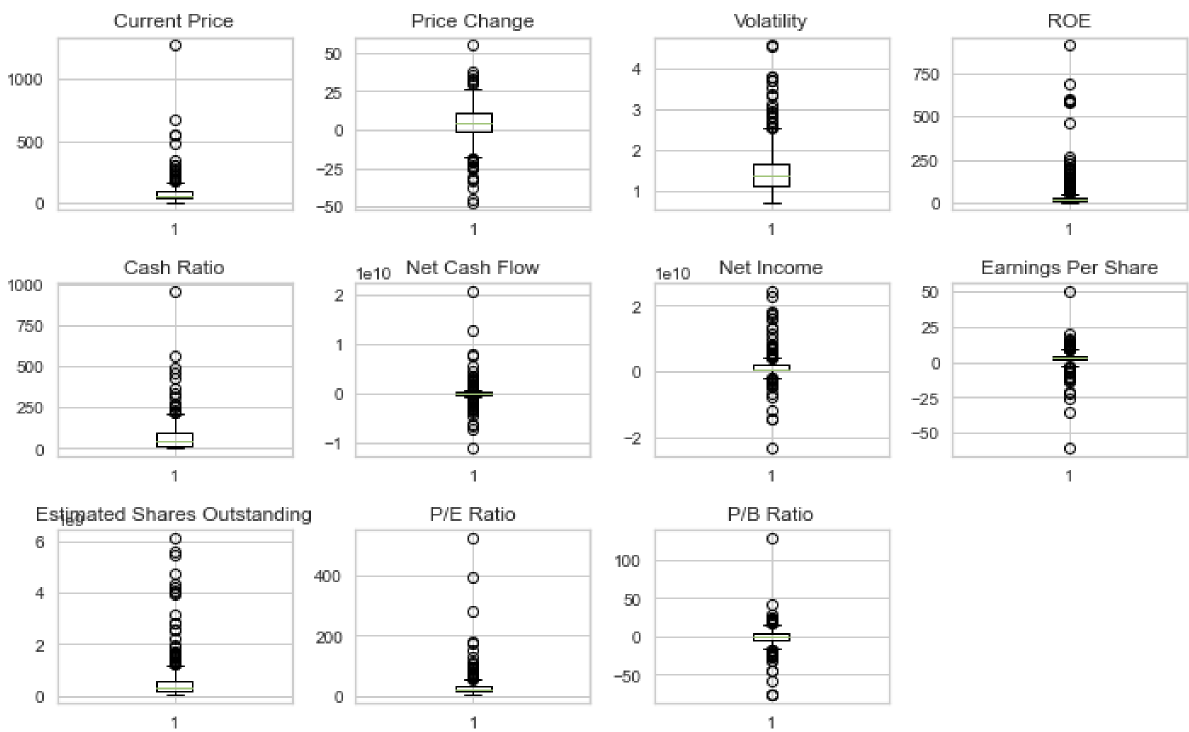
- Missing value treatment (not needed, no missing values)
- Feature engineering (if needed)
- Outlier detection and treatment
- Preparing data for modeling

- Any other preprocessing steps (if needed)

Outlier Detection and treatment

In [34]:

```
# outlier detection using boxplot

num_cols = df.select_dtypes(include=np.number).columns.tolist()
plt.figure(figsize=(10, 8))

for i, variable in enumerate(num_cols):
    plt.subplot(4, 4, i + 1)
    plt.boxplot(df[variable], whis=1.5)
    plt.tight_layout()
    plt.title(variable)

plt.show()
```



Observations:

- There are quite a few outliers in the data, notably in net income and estimated shares outstanding.
- However, they are proper values and reflect the market. We will scale the data before proceeding with clustering.

In [35]:

```
# Scaling the data set before clustering

scaler = StandardScaler()
subset = df[num_cols].copy()
subset_scaled = scaler.fit_transform(subset)
```

In [36]:

```
subset_scaled_df = pd.DataFrame(subset_scaled, columns=subset.columns)
```

K-means Clustering

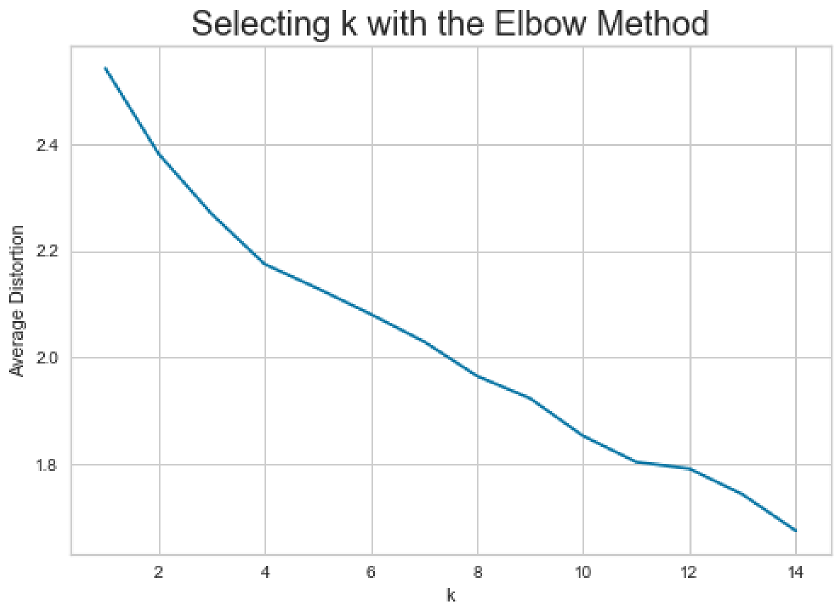
In [37]:

```
k_means = subset_scaled_df.copy()
```

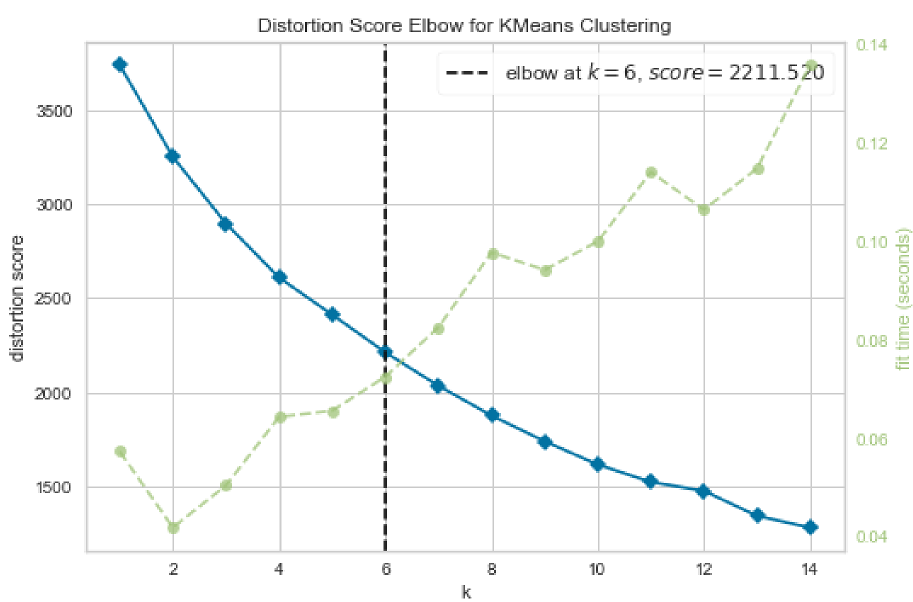
In [38]:

```
clusters = range(1, 15)
meanDistortions = []

for k in clusters:
    model = KMeans(n_clusters=k, random_state=1)
    model.fit(subset_scaled_df)
    prediction = model.predict(k_means)
    distortion = (
        sum(np.min(cdist(k_means, model.cluster_centers_, "euclidean"), axis=1))
        / k_means.shape[0]
    )

    meanDistortions.append(distortion)

    print("Number of Clusters:", k, "\tAverage Distortion:", distortion)

plt.plot(clusters, meanDistortions, "bx-")
plt.xlabel("k")
plt.ylabel("Average Distortion")
plt.title("Selecting k with the Elbow Method", fontsize=20)
plt.show()
```

```
Number of Clusters: 1   Average Distortion: 2.5425069919221697
Number of Clusters: 2   Average Distortion: 2.382318498894466
Number of Clusters: 3   Average Distortion: 2.2692367155390745
Number of Clusters: 4   Average Distortion: 2.1745559827866363
Number of Clusters: 5   Average Distortion: 2.128799332840716
Number of Clusters: 6   Average Distortion: 2.080400099226289
Number of Clusters: 7   Average Distortion: 2.0289794220177395
Number of Clusters: 8   Average Distortion: 1.964144163389972
Number of Clusters: 9   Average Distortion: 1.9221492045198068
Number of Clusters: 10  Average Distortion: 1.8513913649973124
Number of Clusters: 11  Average Distortion: 1.8024134734578485
Number of Clusters: 12  Average Distortion: 1.7900931879652673
Number of Clusters: 13  Average Distortion: 1.7417609203336912
Number of Clusters: 14  Average Distortion: 1.673559857259703
```



In [39]:
```
model = KMeans(random_state=1)
visualizer = KElbowVisualizer(model, k=(1, 15), timings=True)
visualizer.fit(k_means)
visualizer.show()
```



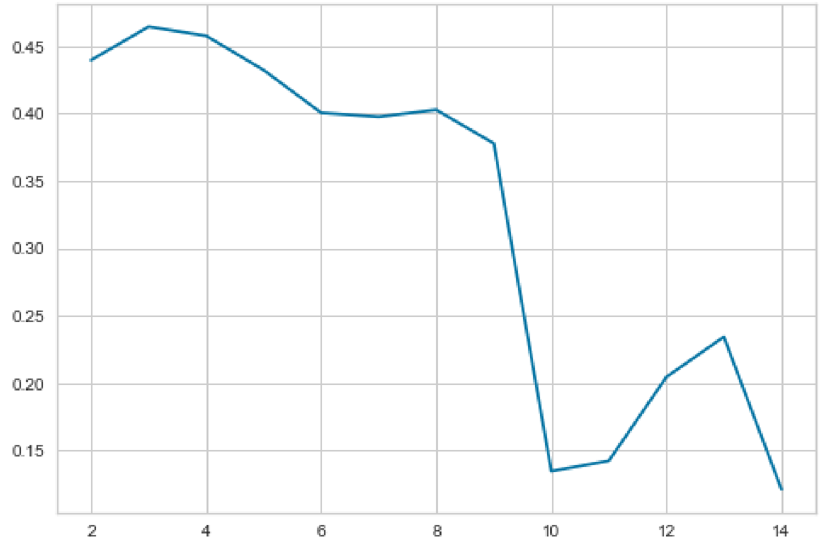
Out[39]: <AxesSubplot:title={'center':'Distortion Score Elbow for KMeans Clustering'}, xlabel='k', ylabel='distortion score'>

The appropriate value of k from the elbow curve is 6. Let's check the silhouette scores.

In [40]:

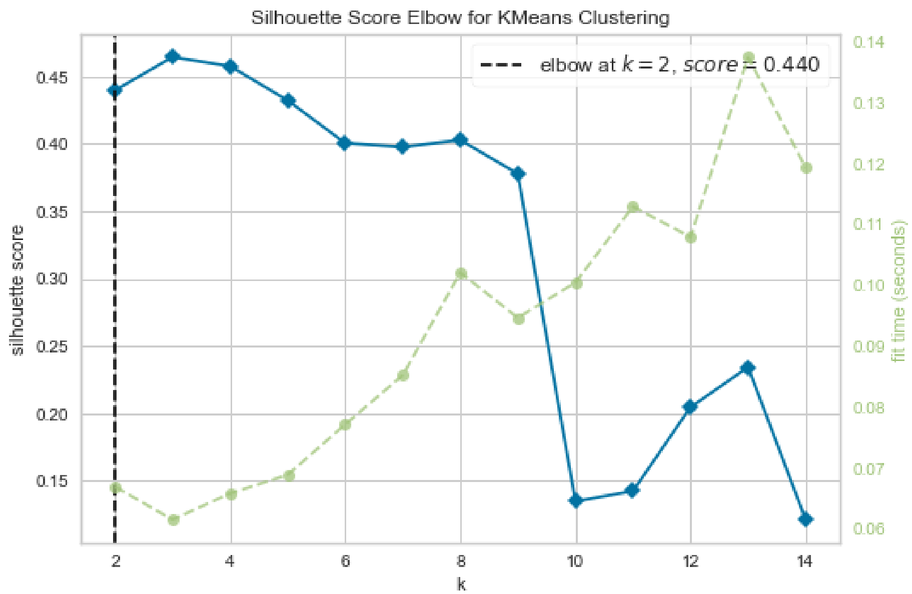
```
sil_score = []
cluster_list = range(2, 15)
for n_clusters in cluster_list:
    clusterer = KMeans(n_clusters=n_clusters, random_state=1)
    preds = clusterer.fit_predict((subset_scaled_df))
    score = silhouette_score(k_means, preds)
    sil_score.append(score)
    print("For n_clusters = {}, the silhouette score is {})".format(n_clusters, score))

plt.plot(cluster_list, sil_score)
plt.show()
```

```
For n_clusters = 2, the silhouette score is 0.43969639509980457)
For n_clusters = 3, the silhouette score is 0.4644405674779404)
For n_clusters = 4, the silhouette score is 0.4577225970476733)
For n_clusters = 5, the silhouette score is 0.43228336443659804)
For n_clusters = 6, the silhouette score is 0.4005422737213617)
For n_clusters = 7, the silhouette score is 0.3976335364987305)
For n_clusters = 8, the silhouette score is 0.40278401969450467)
For n_clusters = 9, the silhouette score is 0.3778585981433699)
For n_clusters = 10, the silhouette score is 0.13458938329968687)
For n_clusters = 11, the silhouette score is 0.1421832155528444)
For n_clusters = 12, the silhouette score is 0.2044669621527429)
For n_clusters = 13, the silhouette score is 0.23424874810104204)
For n_clusters = 14, the silhouette score is 0.12102526472829901)
```



In [41]:

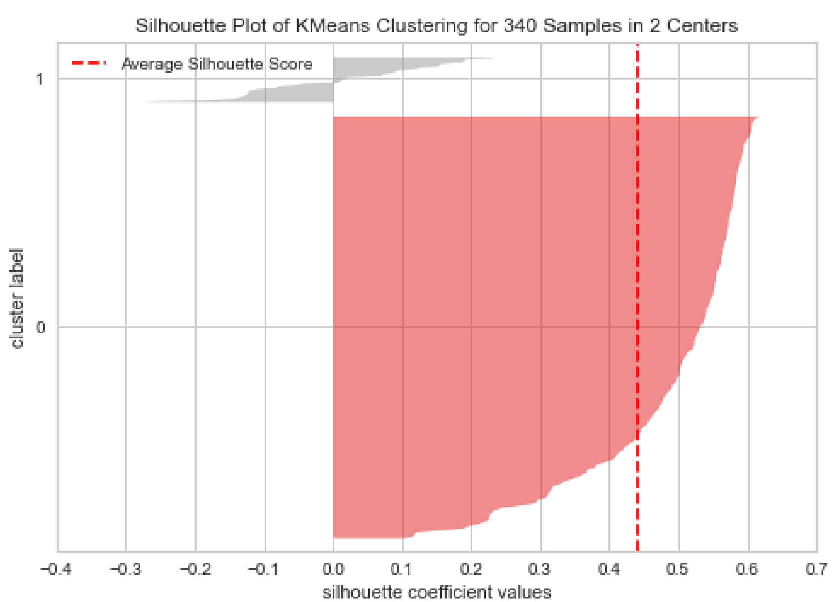
```
model = KMeans(random_state=1)
visualizer = KElbowVisualizer(model, k=(2, 15), metric="silhouette", timings=True)
visualizer.fit(k_means)
visualizer.show()
```



Out[41]: <AxesSubplot:title={'center':'Silhouette Score Elbow for KMeans Clustering'}, xlabel='k', ylabel='silhouette score'>

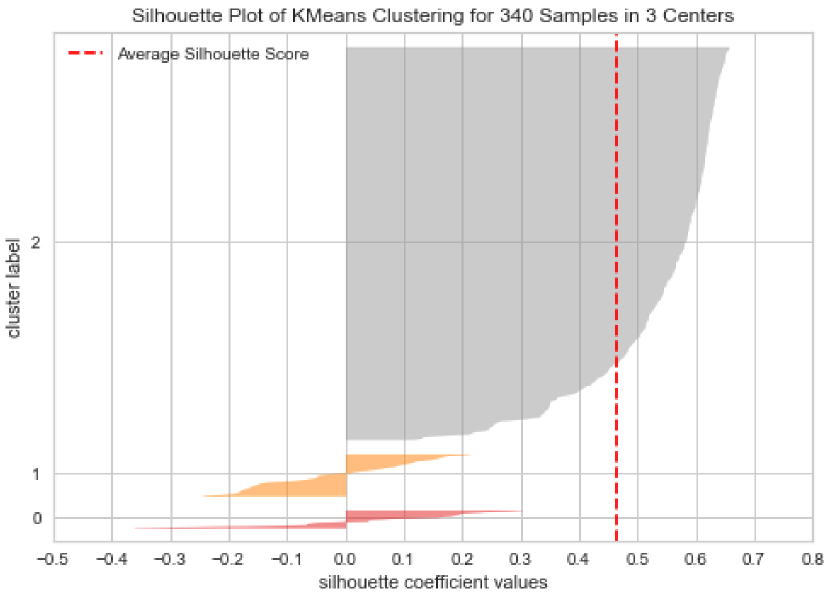
In [42]:

```
# finding optimal no. of clusters with silhouette coefficients
visualizer = SilhouetteVisualizer(KMeans(2, random_state=1))
visualizer.fit(k_means)
visualizer.show()
```



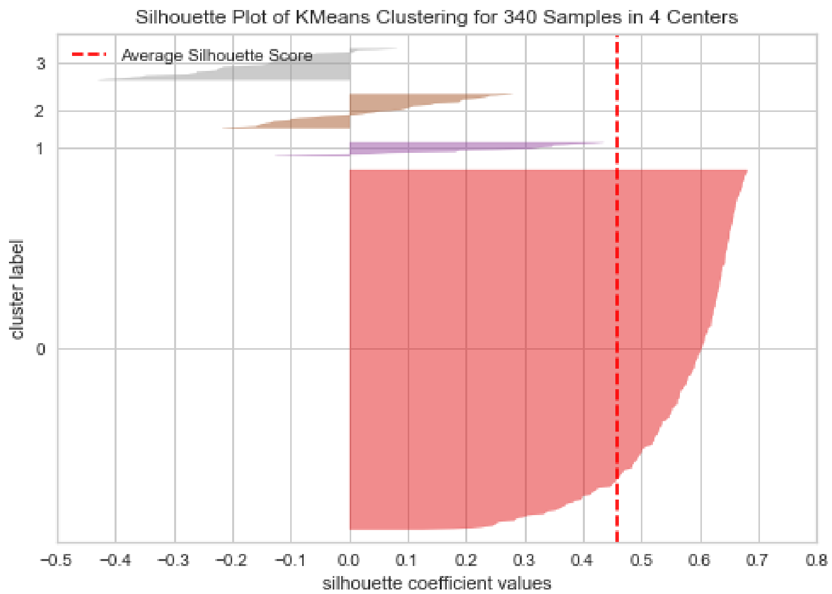
Out[42]: <AxesSubplot:title={'center':'Silhouette Plot of KMeans Clustering for 340 Samples in 2 Centers'}, xlabel='silhouette coefficient values', ylabel='cluster label'>

In [43]:
```
# finding optimal no. of clusters with silhouette coefficients
visualizer = SilhouetteVisualizer(KMeans(3, random_state=1))
visualizer.fit(k_means)
visualizer.show()
```



Out[43]: <AxesSubplot:title={'center':'Silhouette Plot of KMeans Clustering for 340 Samples in 3 Centers'}, xlabel='silhouette coefficient values', ylabel='cluster label'>

In [44]:
```
# finding optimal no. of clusters with silhouette coefficients
visualizer = SilhouetteVisualizer(KMeans(4, random_state=1))
visualizer.fit(k_means)
visualizer.show()
```



Out[44]: <AxesSubplot:title={'center':'Silhouette Plot of KMeans Clustering for 340 Samples in 4 Centers'}, xlabel='silhouette coefficient values', ylabel='cluster label'>

Let's take 3 as the appropriate no. of clusters as the silhouette score is high enough.

In [45]:
```
final = KMeans(n_clusters=3, random_state=0)
final.fit(k_means)
```

Out[45]: KMeans(n_clusters=3, random_state=0)

In [46]:
```
# creating a copy of the original data
df1 = df.copy()

# adding final k-means model cluster labels
k_means["KM_segments"] = final.labels_
df1["KM_segments"] = final.labels_
```

Cluster Profiles

In [47]:
```
cluster_profile = df1.groupby("KM_segments").mean()
```

In [48]:
```
cluster_profile["count_in_each_segment"] = (
    df1.groupby("KM_segments")["Security"].count().values  ## Complete the code to groupby the cluster labels
)
```

In [49]:
```python
cluster_profile.style.highlight_max(color="lightgreen", axis=0)
```

Out[49]:

KM_segments	Current Price	Price Change	Volatility	ROE	Cash Ratio	Net Cash Flow	Net Income	Earnings Per Share	Estimated Shares Outstanding	P/E Ratio	P/B I
0	84.250468	5.595187	1.402117	34.146758	66.815700	10741689.419795	1449597119.453925	3.902338	426357529.820239	24.416003	-2.0
1	52.142857	6.779993	1.175153	26.142857	140.142857	760285714.285714	13368785714.285715	3.769286	3838879870.871428	20.654832	-3.52
2	62.963940	-10.537087	2.774534	93.696970	68.757576	154287151.515152	-3145581545.454545	-7.639091	530986678.995151	110.461063	1.65

In [50]:
```python
# let's see the names of the companies in each cluster
for cl in df1["KM_segments"].unique():
    print("In cluster {}, the following companies are present:".format(cl))
    print(df1[df1["KM_segments"] == cl]["Security"].unique())
    print()
```

```
In cluster 0, the following companies are present:
['American Airlines Group' 'AbbVie' 'Abbott Laboratories'
 'Adobe Systems Inc' 'Archer-Daniels-Midland Co' 'Alliance Data Systems'
 'Ameren Corp' 'American Electric Power' 'AFLAC Inc'
 'American International Group, Inc.' 'Apartment Investment & Mgmt'
 'Assurant Inc' 'Arthur J. Gallagher & Co.' 'Akamai Technologies Inc'
 'Albemarle Corp' 'Alaska Air Group Inc' 'Allstate Corp' 'Allegion'
 'Applied Materials Inc' 'AMETEK Inc' 'Affiliated Managers Group Inc'
 'Amgen Inc' 'Ameriprise Financial' 'American Tower Corp A'
 'AutoNation Inc' 'Anthem Inc.' 'Aon plc' 'Amphenol Corp' 'Arconic Inc'
 'Activision Blizzard' 'AvalonBay Communities, Inc.' 'Broadcom'
 'American Water Works Company Inc' 'American Express Co' 'Boeing Company'
 'Baxter International Inc.' 'BB&T Corporation' 'Bard (C.R.) Inc.'
 'BIOGEN IDEC Inc.' 'The Bank of New York Mellon Corp.' 'Ball Corp'
 'Bristol-Myers Squibb' 'Boston Scientific' 'BorgWarner'
 'Boston Properties' 'Caterpillar Inc.' 'Chubb Limited' 'CBRE Group'
 'Crown Castle International Corp.' 'Carnival Corp.' 'Celgene Corp.'
 'CF Industries Holdings Inc' 'Citizens Financial Group' 'Church & Dwight'
 'C. H. Robinson Worldwide' 'Charter Communications' 'CIGNA Corp.'
 'Cincinnati Financial' 'Colgate-Palmolive' 'Comerica Inc.'
 'CME Group Inc.' 'Chipotle Mexican Grill' 'Cummins Inc.' 'CMS Energy'
 'Centene Corporation' 'CenterPoint Energy' 'Capital One Financial'
 'The Cooper Companies' 'CSX Corp.' 'CenturyLink Inc'
 'Cognizant Technology Solutions' 'Citrix Systems' 'CVS Health'
 'Chevron Corp.' 'Dominion Resources' 'Delta Air Lines' 'Du Pont (E.I.)'
 'Deere & Co.' 'Discover Financial Services' 'Quest Diagnostics'
 'Danaher Corp.' 'The Walt Disney Company' 'Discovery Communications-A'
 'Discovery Communications-C' 'Delphi Automotive' 'Digital Realty Trust'
 'Dun & Bradstreet' 'Dover Corp.' 'Dr Pepper Snapple Group' 'Duke Energy'
 'DaVita Inc.' 'eBay Inc.' 'Ecolab Inc.' 'Consolidated Edison'
 'Equifax Inc.' "Edison Int'l" 'Eastman Chemical' 'Equinix'
 'Equity Residential' 'Eversource Energy' 'Essex Property Trust, Inc.'
 'E*Trade' 'Eaton Corporation' 'Entergy Corp.' 'Edwards Lifesciences'
 'Exelon Corp.' "Expeditors Int'l" 'Expedia Inc.' 'Extra Space Storage'
 'Fastenal Co' 'Fortune Brands Home & Security' 'FirstEnergy Corp'
```

```
'Fidelity National Information Services' 'Fiserv Inc' 'FLIR Systems'
'Fluor Corp.' 'Flowserve Corporation' 'FMC Corporation'
'Federal Realty Investment Trust' 'First Solar Inc'
'Frontier Communications' 'General Dynamics'
'General Growth Properties Inc.' 'Corning Inc.' 'General Motors'
'Genuine Parts' 'Garmin Ltd.' 'Goodyear Tire & Rubber'
'Grainger (W.W.) Inc.' 'Hasbro Inc.' 'Huntington Bancshares'
'HCA Holdings' 'Welltower Inc.' 'HCP Inc.' 'Hartford Financial Svc.Gp.'
'Harley-Davidson' "Honeywell Int'l Inc." 'HP Inc.' 'Hormel Foods Corp.'
'Henry Schein' 'Host Hotels & Resorts' 'The Hershey Company'
'Humana Inc.' 'International Business Machines' 'IDEXX Laboratories'
'Intl Flavors & Fragrances' 'International Paper' 'Interpublic Group'
'Iron Mountain Incorporated' 'Intuitive Surgical Inc.'
'Illinois Tool Works' 'Invesco Ltd.' 'J. B. Hunt Transport Services'
'Jacobs Engineering Group' 'Juniper Networks' 'Kimco Realty'
'Kimberly-Clark' 'Kansas City Southern' 'Leggett & Platt' 'Lennar Corp.'
'Laboratory Corp. of America Holding' 'LKQ Corporation'
'L-3 Communications Holdings' 'Lilly (Eli) & Co.' 'Lockheed Martin Corp.'
'Alliant Energy Corp' 'Leucadia National Corp.' 'Southwest Airlines'
'Level 3 Communications' 'LyondellBasell' 'Mastercard Inc.'
'Mid-America Apartments' 'Macerich' "Marriott Int'l." 'Masco Corp.'
'Mattel Inc.' "McDonald's Corp." "Moody's Corp" 'Mondelez International'
'MetLife Inc.' 'Mohawk Industries' 'Mead Johnson' 'McCormick & Co.'
'Martin Marietta Materials' 'Marsh & McLennan' '3M Company'
'Monster Beverage' 'Altria Group Inc' 'The Mosaic Company'
'Marathon Petroleum' 'M&T Bank Corp.' 'Mettler Toledo' 'Mylan N.V.'
'Navient' 'NASDAQ OMX Group' 'NextEra Energy'
'Newmont Mining Corp. (Hldg. Co.)' 'Nielsen Holdings'
'Norfolk Southern Corp.' 'Northern Trust Corp.' 'Nucor Corp.'
'Newell Brands' 'Realty Income Corporation' 'Omnicom Group'
"O'Reilly Automotive" "People's United Financial" 'Pitney-Bowes'
'PACCAR Inc.' 'PG&E Corp.' 'Priceline.com Inc'
'Public Serv. Enterprise Inc.' 'PepsiCo Inc.' 'Principal Financial Group'
'Procter & Gamble' 'Progressive Corp.' 'Pulte Homes Inc.'
'Philip Morris International' 'PNC Financial Services' 'Pentair Ltd.'
'Pinnacle West Capital' 'PPG Industries' 'PPL Corp.'
'Prudential Financial' 'Phillips 66' 'Praxair Inc.' 'PayPal'
'Ryder System' 'Royal Caribbean Cruises Ltd' 'Regeneron'
'Robert Half International' 'Roper Industries' 'Republic Services Inc'
'SCANA Corp' 'Charles Schwab Corporation' 'Sealed Air' 'Sherwin-Williams'
'SL Green Realty' 'Scripps Networks Interactive Inc.' 'Southern Co.'
'Simon Property Group Inc' 'S&P Global, Inc.' 'Stericycle Inc'
'Sempra Energy' 'SunTrust Banks' 'State Street Corp.'
'Skyworks Solutions' 'Synchrony Financial' 'Stryker Corp.'
'Molson Coors Brewing Company' 'Tegna, Inc.' 'Torchmark Corp.'
'Thermo Fisher Scientific' 'TripAdvisor' 'The Travelers Companies Inc.'
'Tractor Supply Company' 'Tyson Foods' 'Tesoro Petroleum Co.'
'Total System Services' 'Texas Instruments' 'Under Armour'
'United Continental Holdings' 'UDR Inc' 'Universal Health Services, Inc.'
'United Health Group Inc.' 'Unum Group' 'Union Pacific'
'United Parcel Service' 'United Technologies' 'Varian Medical Systems'
'Valero Energy' 'Vulcan Materials' 'Vornado Realty Trust'
'Verisk Analytics' 'Verisign Inc.' 'Vertex Pharmaceuticals Inc'
'Ventas Inc' 'Waters Corporation' 'Wec Energy Group Inc'
'Whirlpool Corp.' 'Waste Management Inc.' 'Western Union Co'
'Weyerhaeuser Corp.' 'Wyndham Worldwide' 'Xcel Energy Inc' 'XL Capital'
'Dentsply Sirona' 'Xerox Corp.' 'Xylem Inc.' 'Yahoo Inc.'
'Yum! Brands Inc' 'Zimmer Biomet Holdings' 'Zions Bancorp' 'Zoetis']
```

```
In cluster 2, the following companies are present:
['Analog Devices, Inc.' 'Alexion Pharmaceuticals' 'Amazon.com Inc'
 'Apache Corporation' 'Anadarko Petroleum Corp' 'Baker Hughes Inc'
 'Chesapeake Energy' 'Cabot Oil & Gas' 'Concho Resources'
 'Devon Energy Corp.' 'EOG Resources' 'EQT Corporation'
 'Freeport-McMoran Cp & Gld' 'Halliburton Co.' 'Hess Corporation'
 'Hewlett Packard Enterprise' 'Kinder Morgan' 'Marathon Oil Corp.'
 'Murphy Oil' 'Noble Energy Inc' 'Netflix Inc.' 'Newfield Exploration Co'
 'National Oilwell Varco Inc.' 'ONEOK' 'Occidental Petroleum'
 'Quanta Services Inc.' 'Range Resources Corp.' 'Spectra Energy Corp.'
 'Southwestern Energy' 'Teradata Corp.' 'Williams Cos.' 'Wynn Resorts Ltd'
 'Cimarex Energy']

In cluster 1, the following companies are present:
['Bank of America Corp' 'Citigroup Inc.' 'Ford Motor' 'Facebook'
 'Gilead Sciences' 'Intel Corp.' 'JPMorgan Chase & Co.'
 'Coca Cola Company' 'Merck & Co.' 'Pfizer Inc.' 'AT&T Inc'
 'Verizon Communications' 'Wells Fargo' 'Exxon Mobil Corp.']
```

In [51]:

```
df1.groupby(["KM_segments", "GICS Sector"])['Security'].count()
```

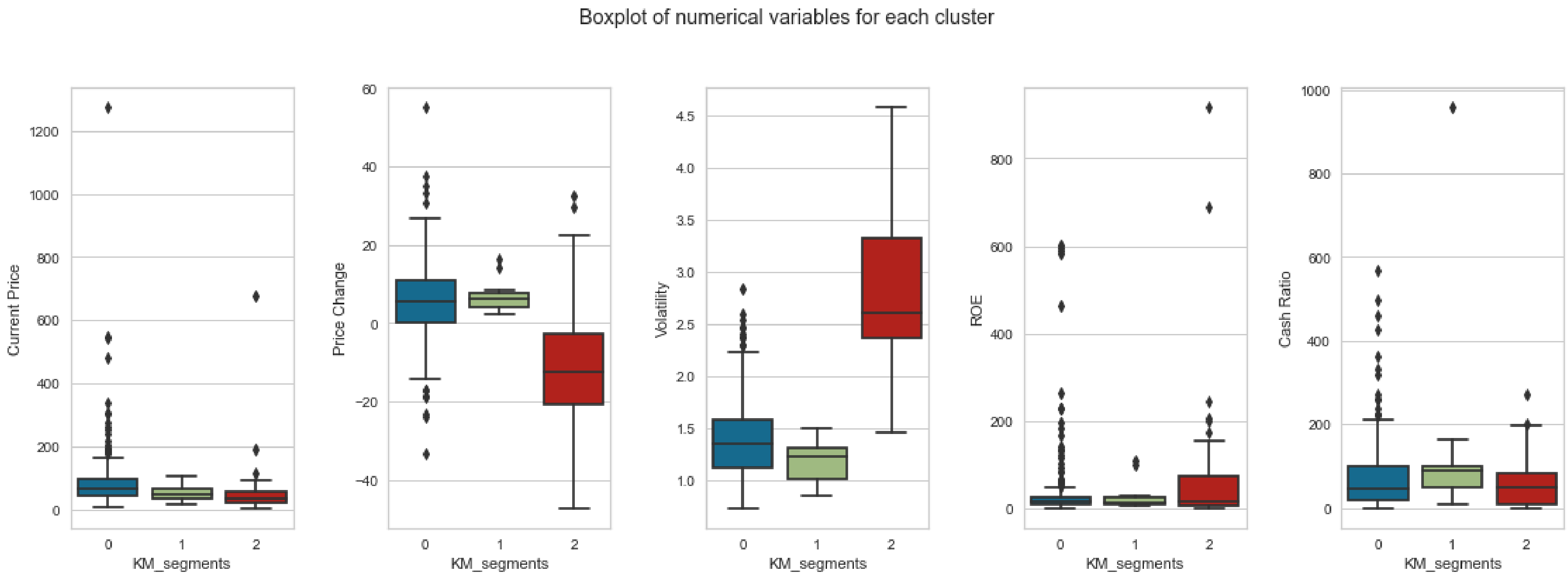
Out[51]:

```
KM_segments  GICS Sector
0            Consumer Discretionary         37
             Consumer Staples               18
             Energy                          5
             Financials                     45
             Health Care                    36
             Industrials                    52
             Information Technology         27
             Materials                      19
             Real Estate                    27
             Telecommunications Services     3
             Utilities                      24
1            Consumer Discretionary          1
             Consumer Staples                1
             Energy                          1
             Financials                      4
             Health Care                     3
             Information Technology          2
             Telecommunications Services     2
2            Consumer Discretionary          2
             Energy                         24
             Health Care                     1
             Industrials                     1
             Information Technology          4
             Materials                       1
Name: Security, dtype: int64
```

In [52]:

```
fig, axes = plt.subplots(1, 5, figsize=(16, 6))
fig.suptitle("Boxplot of numerical variables for each cluster")
counter = 0
for ii in range(5):
    sns.boxplot(ax=axes[ii], y=df1[num_cols[counter]], x=df1["KM_segments"])
```

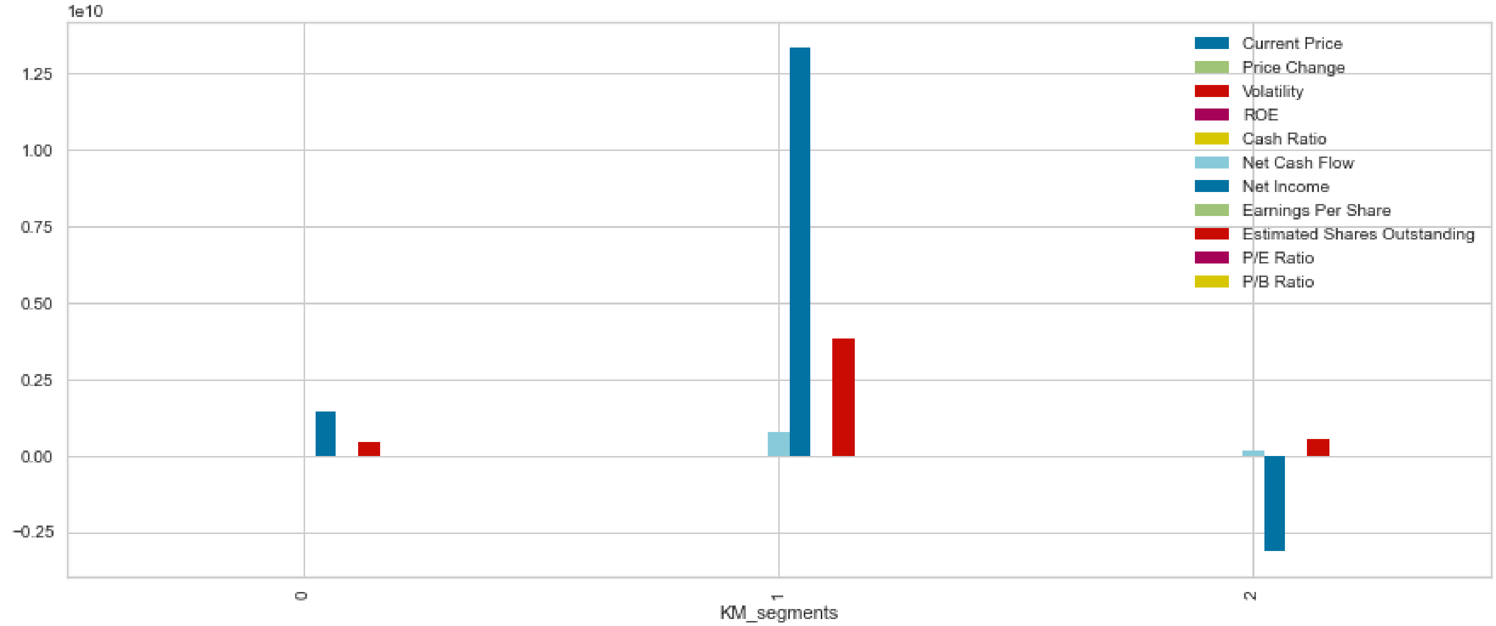
```
    counter = counter + 1

fig.tight_layout(pad=2.0)
```



In [53]:

```
df1.groupby("KM_segments").mean().plot.bar(figsize=(15, 6))
```

Out[53]: <AxesSubplot:xlabel='KM_segments'>



Insights

- Cluster 0:
  - Largest sectors are: Industrials, Financials, Consumer Discretionary
  - Largest number of companies (notable: several banks, petrol, holdings)
  - Current price has most outliers
  - Moderate price change
  - Low votality
  - Many outliers in ROE and Cash Ratio
- Cluster 1:
  - Largest sectors are: Financials, Health Care
  - Notable companies: Merck, Pfizer, Exxon, Verizon, Wells Fargo, Facebook
  - Current price is highest
  - Low price change
  - Low votality
  - Moderate ROE and Cash Ratio
- Cluster 2:
  - Largest sectors are: Energy, Information Technology
  - Notable companies: Amazon, Netflix, several oil corporations
  - Current price is lowest
  - Low price change

- Very high votality
- High ROE, Moderate Cash Ratio

Hierarchical Clustering

In [54]:

```
hc_df = subset_scaled_df.copy()
```

In [55]:

```
# list of distance metrics
distance_metrics = ["euclidean", "chebyshev", "mahalanobis", "cityblock"]

# list of linkage methods
linkage_methods = ["single", "complete", "average", "weighted"]

high_cophenet_corr = 0
high_dm_lm = [0, 0]

for dm in distance_metrics:
    for lm in linkage_methods:
        Z = linkage(hc_df, metric=dm, method=lm)
        c, coph_dists = cophenet(Z, pdist(hc_df))
        print(
            "Cophenetic correlation for {} distance and {} linkage is {}.".format(
                dm.capitalize(), lm, c
            )
        )
        if high_cophenet_corr < c:
            high_cophenet_corr = c
            high_dm_lm[0] = dm
            high_dm_lm[1] = lm
```

```
Cophenetic correlation for Euclidean distance and single linkage is 0.9232271494002922.
Cophenetic correlation for Euclidean distance and complete linkage is 0.7873280186580672.
Cophenetic correlation for Euclidean distance and average linkage is 0.9422540609560814.
Cophenetic correlation for Euclidean distance and weighted linkage is 0.8693784298129404.
Cophenetic correlation for Chebyshev distance and single linkage is 0.9062538164750717.
Cophenetic correlation for Chebyshev distance and complete linkage is 0.598891419111242.
Cophenetic correlation for Chebyshev distance and average linkage is 0.9338265528030499.
Cophenetic correlation for Chebyshev distance and weighted linkage is 0.9127355892367.
Cophenetic correlation for Mahalanobis distance and single linkage is 0.9259195530524591.
Cophenetic correlation for Mahalanobis distance and complete linkage is 0.7925307202850002.
Cophenetic correlation for Mahalanobis distance and average linkage is 0.9247324030159737.
Cophenetic correlation for Mahalanobis distance and weighted linkage is 0.8708317490180428.
Cophenetic correlation for Cityblock distance and single linkage is 0.9334186366528574.
Cophenetic correlation for Cityblock distance and complete linkage is 0.7375328863205818.
Cophenetic correlation for Cityblock distance and average linkage is 0.9302145048594667.
Cophenetic correlation for Cityblock distance and weighted linkage is 0.731045513520281.
```

In [56]:

```
# printing the combination of distance metric and linkage method with the highest cophenetic correlation
print(
```

```
    "Highest cophenetic correlation is {}, which is obtained with {} distance and {} linkage.".format(
        high_cophenet_corr, high_dm_lm[0].capitalize(), high_dm_lm[1]
    )
)
```

```
Highest cophenetic correlation is 0.9422540609560814, which is obtained with Euclidean distance and average linkage.
```

Let's explore different linkage methods with Euclidean distance only.

In [57]:

```python
# list of linkage methods
linkage_methods = ["single", "complete", "average", "centroid", "ward", "weighted"]

high_cophenet_corr = 0
high_dm_lm = [0, 0]

for lm in linkage_methods:
    Z = linkage(hc_df, metric="euclidean", method=lm)
    c, coph_dists = cophenet(Z, pdist(hc_df))
    print("Cophenetic correlation for {} linkage is {}.".format(lm, c))
    if high_cophenet_corr < c:
        high_cophenet_corr = c
        high_dm_lm[0] = "euclidean"
        high_dm_lm[1] = lm
```

```
Cophenetic correlation for single linkage is 0.9232271494002922.
Cophenetic correlation for complete linkage is 0.7873280186580672.
Cophenetic correlation for average linkage is 0.9422540609560814.
Cophenetic correlation for centroid linkage is 0.9314012446828154.
Cophenetic correlation for ward linkage is 0.7101180299865353.
Cophenetic correlation for weighted linkage is 0.8693784298129404.
```

In [58]:

```python
# printing the combination of distance metric and linkage method with the highest cophenetic correlation
print(
    "Highest cophenetic correlation is {}, which is obtained with {} linkage.".format(
        high_cophenet_corr, high_dm_lm[1]
    )
)
```

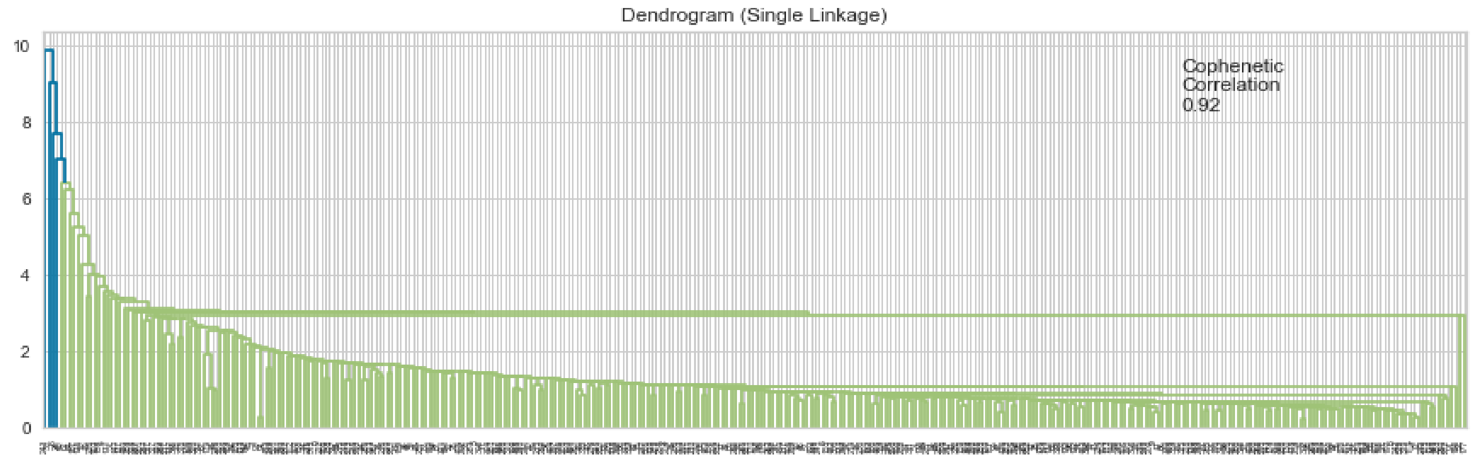
```
Highest cophenetic correlation is 0.9422540609560814, which is obtained with average linkage.
```

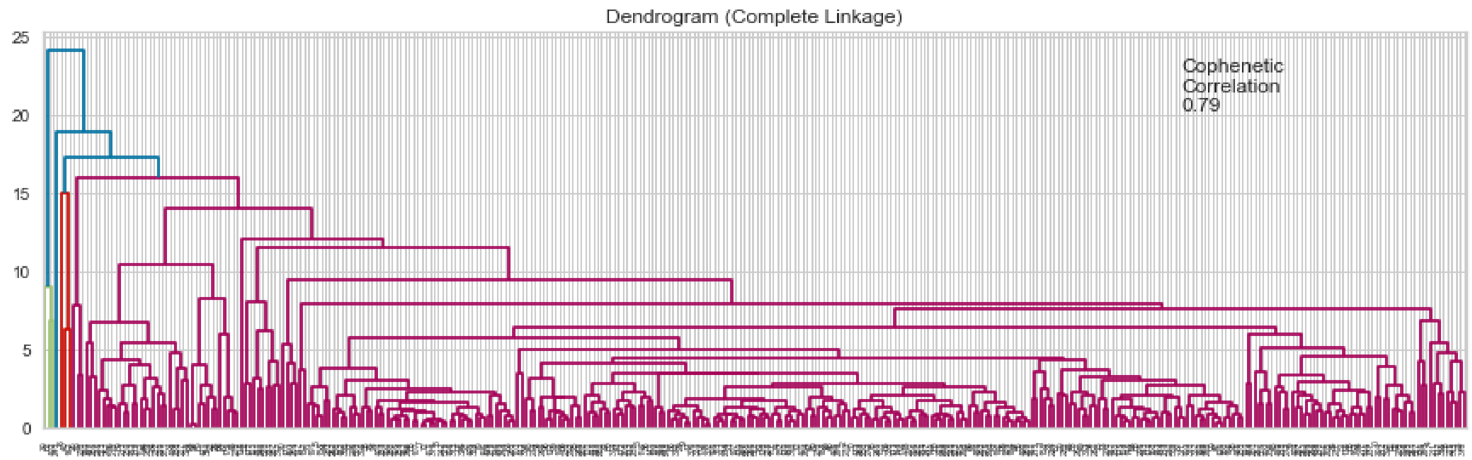
Let's view the dendrograms for the different linkage methods with Euclidean distance.

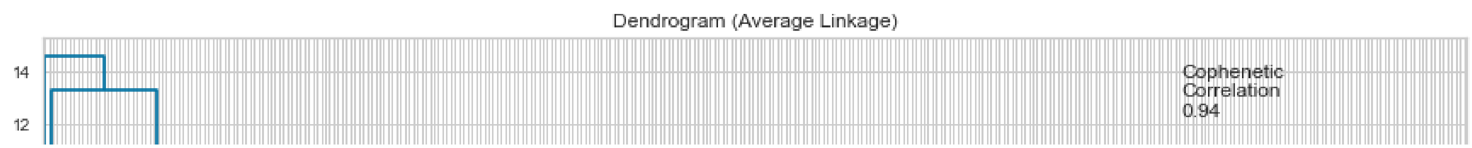
In [59]:

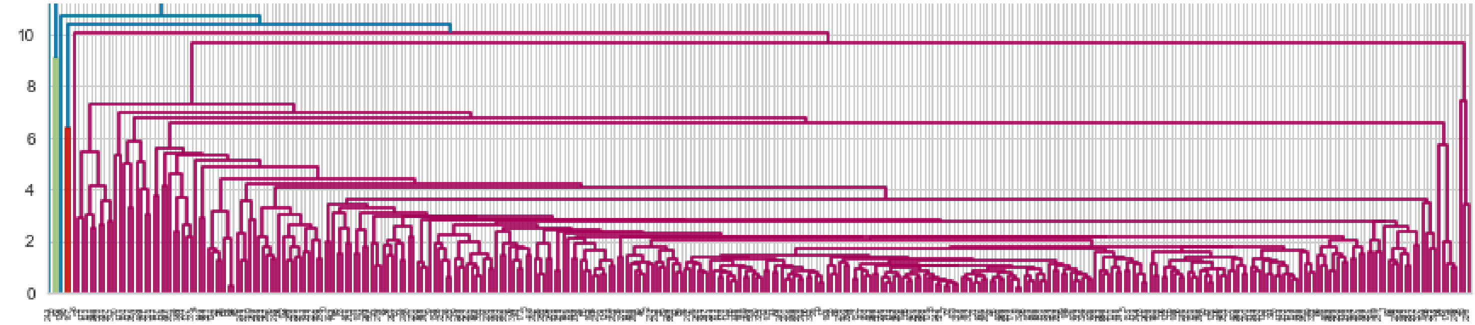
```python
# list of linkage methods
linkage_methods = ["single", "complete", "average", "centroid", "ward", "weighted"]

# lists to save results of cophenetic correlation calculation
compare_cols = ["Linkage", "Cophenetic Coefficient"]

# to create a subplot image
fig, axs = plt.subplots(len(linkage_methods), 1, figsize=(15, 30))

# We will enumerate through the list of linkage methods above
```

```
# For each linkage method, we will plot the dendrogram and calculate the cophenetic correlation
for i, method in enumerate(linkage_methods):
    Z = linkage(hc_df, metric="euclidean", method=method)

    dendrogram(Z, ax=axs[i])
    axs[i].set_title(f"Dendrogram ({method.capitalize()} Linkage)")

    coph_corr, coph_dist = cophenet(Z, pdist(hc_df))
    axs[i].annotate(
        f"Cophenetic\nCorrelation\n{coph_corr:0.2f}",
        (0.80, 0.80),
        xycoords="axes fraction",
    )
```

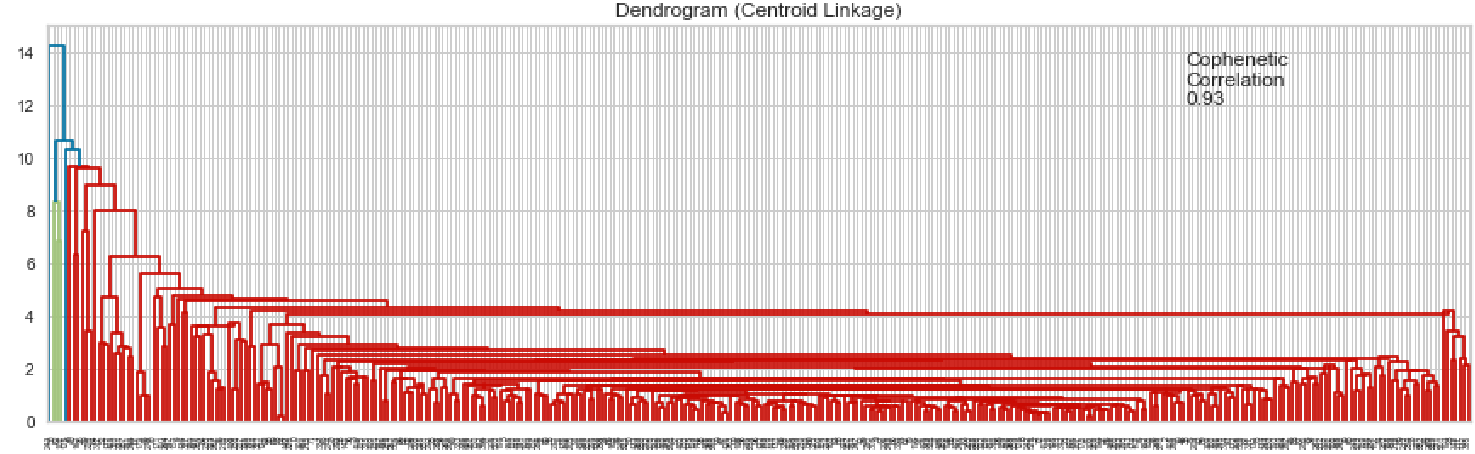




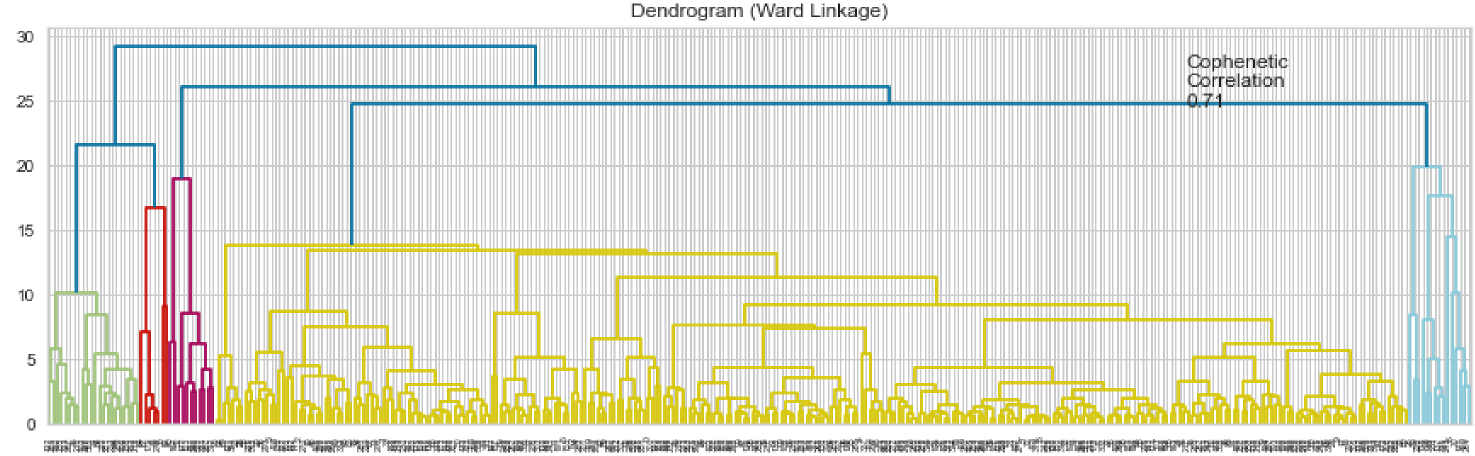




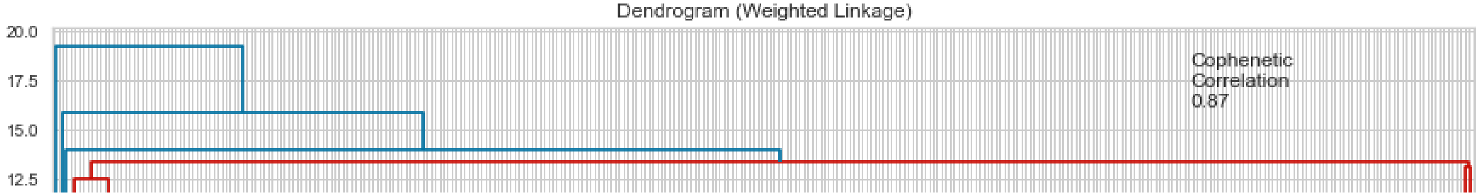
Dendrogram (Centroid Linkage)

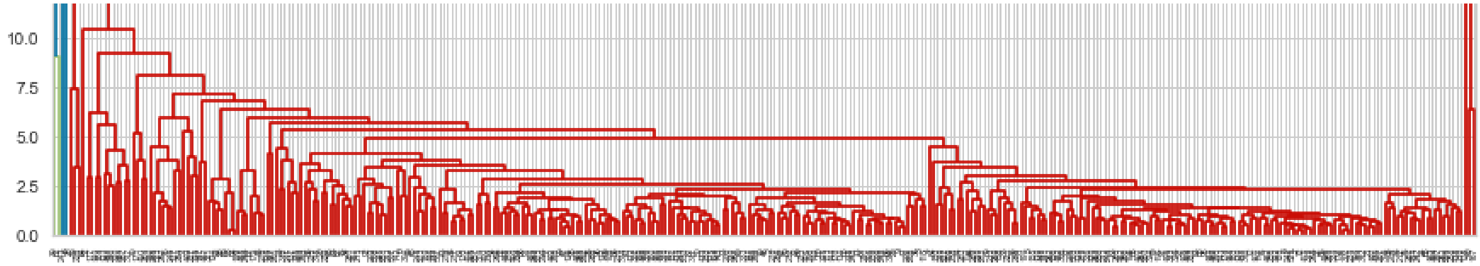


Dendrogram (Ward Linkage)



Dendrogram (Weighted Linkage)





Observations

- The cophenetic correlation is highest for average and centroid linkage methods, followeed by single and weighted.
- We will move ahead with average linkage.
- 6 appears to be the appropriate number of clusters from the dendrogram for average linkage.

In [60]:
```
HCmodel = AgglomerativeClustering(n_clusters=6, affinity="euclidean", linkage="average")
HCmodel.fit(hc_df)
```

Out[60]: AgglomerativeClustering(linkage='average', n_clusters=6)

In [61]:
```
# creating a copy of the original data
df2 = df.copy()

# adding hierarchical cluster labels to the original and scaled dataframes
hc_df["HC_segments"] = HCmodel.labels_
df2["HC_segments"] = HCmodel.labels_
```

Cluster Profiles

In [62]:
```
cluster_profile2 = df2.groupby("HC_segments").mean()
```

In [63]:
```
cluster_profile2["count_in_each_segment"] = (
    df2.groupby("HC_segments")["Security"].count().values
)
```

In [64]:
```
cluster_profile2.style.highlight_max(color="lightgreen", axis=0)
```

Out[64]:

	Current Price	Price Change	Volatility	ROE	Cash Ratio	Net Cash Flow	Net Income	Earnings Per Share	Estimated Shares Outstanding	P/E Rati
HC_segments										
0	77.287589	4.099730	1.518066	35.336336	66.900901	-33197321.321321	1538074666.666667	2.885270	560505037.293543	32.44170

HC_segments	Current Price	Price Change	Volatility	ROE	Cash Ratio	Net Cash Flow	Net Income	Earnings Per Share	Estimated Shares Outstanding	P/E Rati
1	25.640000	11.237908	1.322355	12.500000	130.500000	16755500000.000000	13654000000.000000	3.295000	2791829362.100000	13.64969
2	24.485001	-13.351992	3.482611	802.000000	51.000000	-1292500000.000000	-19106500000.000000	-41.815000	519573983.250000	60.74860
3	104.660004	16.224320	1.320606	8.000000	958.000000	592000000.000000	3669000000.000000	1.310000	2800763359.000000	79.89313
4	1274.949951	3.190527	1.268340	29.000000	184.000000	-1671386000.000000	2551360000.000000	50.090000	50935516.070000	25.45318
5	276.570007	6.189286	1.116976	30.000000	25.000000	90885000.000000	596541000.000000	8.910000	66951851.850000	31.04040

In [65]:

```
# let's see the names of the companies in each cluster
for cl in df2["HC_segments"].unique():
    print("In cluster {}, the following companies are present:".format(cl))
    print(df2[df2["HC_segments"] == cl]["Security"].unique())
    print()
```

```
In cluster 0, the following companies are present:
['American Airlines Group' 'AbbVie' 'Abbott Laboratories'
 'Adobe Systems Inc' 'Analog Devices, Inc.' 'Archer-Daniels-Midland Co'
 'Ameren Corp' 'American Electric Power' 'AFLAC Inc'
 'American International Group, Inc.' 'Apartment Investment & Mgmt'
 'Assurant Inc' 'Arthur J. Gallagher & Co.' 'Akamai Technologies Inc'
 'Albemarle Corp' 'Alaska Air Group Inc' 'Allstate Corp' 'Allegion'
 'Alexion Pharmaceuticals' 'Applied Materials Inc' 'AMETEK Inc'
 'Affiliated Managers Group Inc' 'Amgen Inc' 'Ameriprise Financial'
 'American Tower Corp A' 'Amazon.com Inc' 'AutoNation Inc' 'Anthem Inc.'
 'Aon plc' 'Anadarko Petroleum Corp' 'Amphenol Corp' 'Arconic Inc'
 'Activision Blizzard' 'AvalonBay Communities, Inc.' 'Broadcom'
 'American Water Works Company Inc' 'American Express Co' 'Boeing Company'
 'Baxter International Inc.' 'BB&T Corporation' 'Bard (C.R.) Inc.'
 'Baker Hughes Inc' 'BIOGEN IDEC Inc.' 'The Bank of New York Mellon Corp.'
 'Ball Corp' 'Bristol-Myers Squibb' 'Boston Scientific' 'BorgWarner'
 'Boston Properties' 'Citigroup Inc.' 'Caterpillar Inc.' 'Chubb Limited'
 'CBRE Group' 'Crown Castle International Corp.' 'Carnival Corp.'
 'Celgene Corp.' 'CF Industries Holdings Inc' 'Citizens Financial Group'
 'Church & Dwight' 'C. H. Robinson Worldwide' 'Charter Communications'
 'CIGNA Corp.' 'Cincinnati Financial' 'Colgate-Palmolive' 'Comerica Inc.'
 'CME Group Inc.' 'Chipotle Mexican Grill' 'Cummins Inc.' 'CMS Energy'
 'Centene Corporation' 'CenterPoint Energy' 'Capital One Financial'
 'Cabot Oil & Gas' 'The Cooper Companies' 'CSX Corp.' 'CenturyLink Inc'
 'Cognizant Technology Solutions' 'Citrix Systems' 'CVS Health'
 'Chevron Corp.' 'Concho Resources' 'Dominion Resources' 'Delta Air Lines'
 'Du Pont (E.I.)' 'Deere & Co.' 'Discover Financial Services'
 'Quest Diagnostics' 'Danaher Corp.' 'The Walt Disney Company'
 'Discovery Communications-A' 'Discovery Communications-C'
 'Delphi Automotive' 'Digital Realty Trust' 'Dun & Bradstreet'
 'Dover Corp.' 'Dr Pepper Snapple Group' 'Duke Energy' 'DaVita Inc.'
 'Devon Energy Corp.' 'eBay Inc.' 'Ecolab Inc.' 'Consolidated Edison'
 'Equifax Inc.' "Edison Int'l" 'Eastman Chemical' 'EOG Resources'
 'Equinix' 'Equity Residential' 'EQT Corporation' 'Eversource Energy'
```

```
'Essex Property Trust, Inc.' 'E*Trade' 'Eaton Corporation'
'Entergy Corp.' 'Edwards Lifesciences' 'Exelon Corp.' "Expeditors Int'l"
'Expedia Inc.' 'Extra Space Storage' 'Ford Motor' 'Fastenal Co'
'Fortune Brands Home & Security' 'Freeport-McMoran Cp & Gld'
'FirstEnergy Corp' 'Fidelity National Information Services' 'Fiserv Inc'
'FLIR Systems' 'Fluor Corp.' 'Flowserve Corporation' 'FMC Corporation'
'Federal Realty Investment Trust' 'First Solar Inc'
'Frontier Communications' 'General Dynamics'
'General Growth Properties Inc.' 'Gilead Sciences' 'Corning Inc.'
'General Motors' 'Genuine Parts' 'Garmin Ltd.' 'Goodyear Tire & Rubber'
'Grainger (W.W.) Inc.' 'Halliburton Co.' 'Hasbro Inc.'
'Huntington Bancshares' 'HCA Holdings' 'Welltower Inc.' 'HCP Inc.'
'Hess Corporation' 'Hartford Financial Svc.Gp.' 'Harley-Davidson'
"Honeywell Int'l Inc." 'Hewlett Packard Enterprise' 'HP Inc.'
'Hormel Foods Corp.' 'Henry Schein' 'Host Hotels & Resorts'
'The Hershey Company' 'Humana Inc.' 'International Business Machines'
'IDEXX Laboratories' 'Intl Flavors & Fragrances' 'International Paper'
'Interpublic Group' 'Iron Mountain Incorporated'
'Intuitive Surgical Inc.' 'Illinois Tool Works' 'Invesco Ltd.'
'J. B. Hunt Transport Services' 'Jacobs Engineering Group'
'Juniper Networks' 'JPMorgan Chase & Co.' 'Kimco Realty' 'Kimberly-Clark'
'Kinder Morgan' 'Coca Cola Company' 'Kansas City Southern'
'Leggett & Platt' 'Lennar Corp.' 'Laboratory Corp. of America Holding'
'LKQ Corporation' 'L-3 Communications Holdings' 'Lilly (Eli) & Co.'
'Lockheed Martin Corp.' 'Alliant Energy Corp' 'Leucadia National Corp.'
'Southwest Airlines' 'Level 3 Communications' 'LyondellBasell'
'Mastercard Inc.' 'Mid-America Apartments' 'Macerich' "Marriott Int'l."
'Masco Corp.' 'Mattel Inc.' "McDonald's Corp." "Moody's Corp"
'Mondelez International' 'MetLife Inc.' 'Mohawk Industries'
'Mead Johnson' 'McCormick & Co.' 'Martin Marietta Materials'
'Marsh & McLennan' '3M Company' 'Monster Beverage' 'Altria Group Inc'
'The Mosaic Company' 'Marathon Petroleum' 'Merck & Co.'
'Marathon Oil Corp.' 'M&T Bank Corp.' 'Mettler Toledo' 'Murphy Oil'
'Mylan N.V.' 'Navient' 'Noble Energy Inc' 'NASDAQ OMX Group'
'NextEra Energy' 'Newmont Mining Corp. (Hldg. Co.)' 'Netflix Inc.'
'Newfield Exploration Co' 'Nielsen Holdings'
'National Oilwell Varco Inc.' 'Norfolk Southern Corp.'
'Northern Trust Corp.' 'Nucor Corp.' 'Newell Brands'
'Realty Income Corporation' 'ONEOK' 'Omnicom Group' "O'Reilly Automotive"
'Occidental Petroleum' "People's United Financial" 'Pitney-Bowes'
'PACCAR Inc.' 'PG&E Corp.' 'Public Serv. Enterprise Inc.' 'PepsiCo Inc.'
'Pfizer Inc.' 'Principal Financial Group' 'Procter & Gamble'
'Progressive Corp.' 'Pulte Homes Inc.' 'Philip Morris International'
'PNC Financial Services' 'Pentair Ltd.' 'Pinnacle West Capital'
'PPG Industries' 'PPL Corp.' 'Prudential Financial' 'Phillips 66'
'Quanta Services Inc.' 'Praxair Inc.' 'PayPal' 'Ryder System'
'Royal Caribbean Cruises Ltd' 'Regeneron' 'Robert Half International'
'Roper Industries' 'Range Resources Corp.' 'Republic Services Inc'
'SCANA Corp' 'Charles Schwab Corporation' 'Spectra Energy Corp.'
'Sealed Air' 'Sherwin-Williams' 'SL Green Realty'
'Scripps Networks Interactive Inc.' 'Southern Co.'
'Simon Property Group Inc' 'S&P Global, Inc.' 'Stericycle Inc'
'Sempra Energy' 'SunTrust Banks' 'State Street Corp.'
'Skyworks Solutions' 'Southwestern Energy' 'Synchrony Financial'
'Stryker Corp.' 'AT&T Inc' 'Molson Coors Brewing Company'
'Teradata Corp.' 'Tegna, Inc.' 'Torchmark Corp.'
'Thermo Fisher Scientific' 'TripAdvisor' 'The Travelers Companies Inc.'
'Tractor Supply Company' 'Tyson Foods' 'Tesoro Petroleum Co.'
```

```
 'Total System Services' 'Texas Instruments' 'Under Armour'
 'United Continental Holdings' 'UDR Inc' 'Universal Health Services, Inc.'
 'United Health Group Inc.' 'Unum Group' 'Union Pacific'
 'United Parcel Service' 'United Technologies' 'Varian Medical Systems'
 'Valero Energy' 'Vulcan Materials' 'Vornado Realty Trust'
 'Verisk Analytics' 'Verisign Inc.' 'Vertex Pharmaceuticals Inc'
 'Ventas Inc' 'Verizon Communications' 'Waters Corporation'
 'Wec Energy Group Inc' 'Wells Fargo' 'Whirlpool Corp.'
 'Waste Management Inc.' 'Williams Cos.' 'Western Union Co'
 'Weyerhaeuser Corp.' 'Wyndham Worldwide' 'Wynn Resorts Ltd'
 'Cimarex Energy' 'Xcel Energy Inc' 'XL Capital' 'Exxon Mobil Corp.'
 'Dentsply Sirona' 'Xerox Corp.' 'Xylem Inc.' 'Yahoo Inc.'
 'Yum! Brands Inc' 'Zimmer Biomet Holdings' 'Zions Bancorp' 'Zoetis']

In cluster 5, the following companies are present:
['Alliance Data Systems']

In cluster 2, the following companies are present:
['Apache Corporation' 'Chesapeake Energy']

In cluster 1, the following companies are present:
['Bank of America Corp' 'Intel Corp.']

In cluster 3, the following companies are present:
['Facebook']

In cluster 4, the following companies are present:
['Priceline.com Inc']
```

In [66]:

```
df2.groupby(["HC_segments", "GICS Sector"])['Security'].count()
```

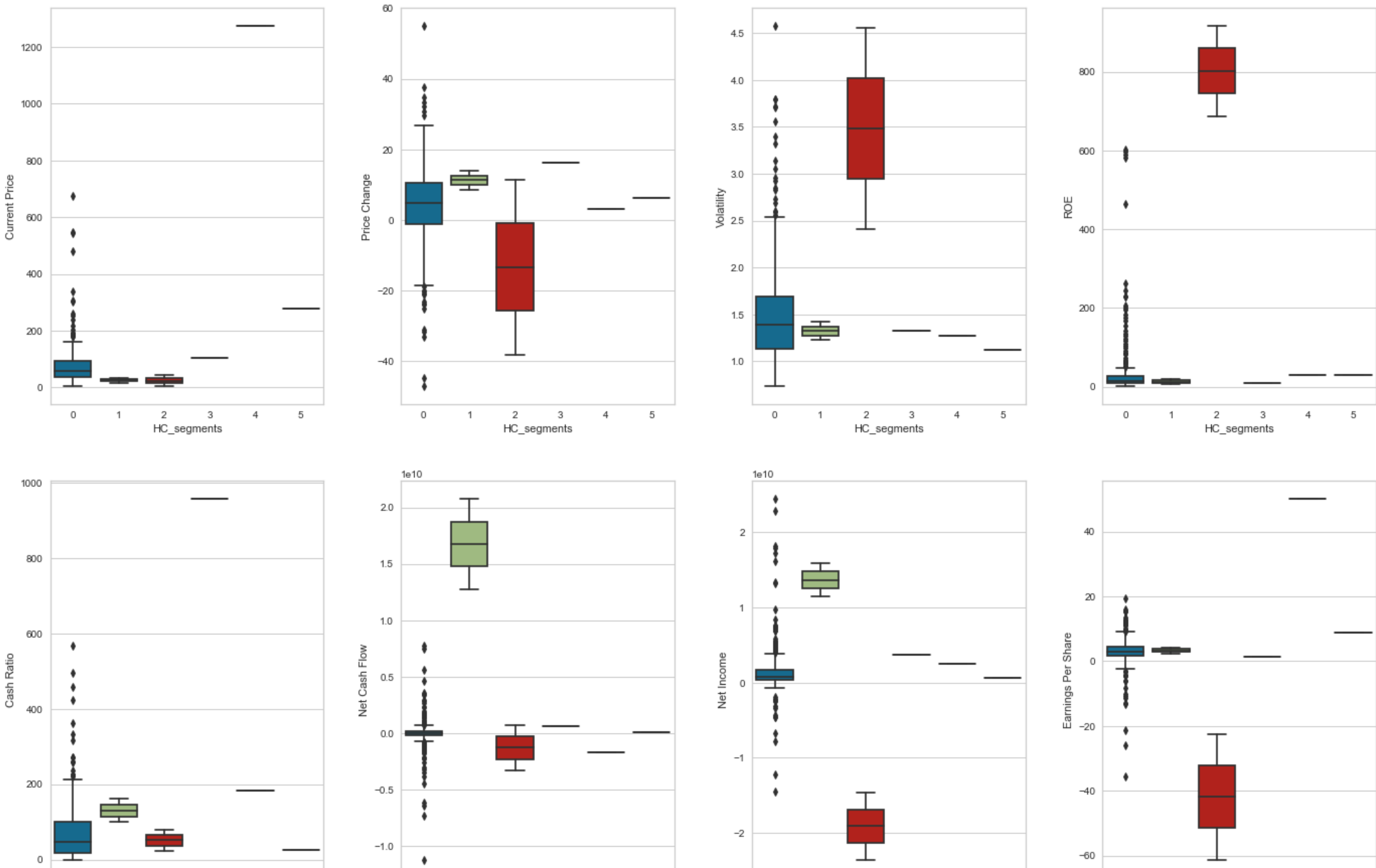
Out[66]:

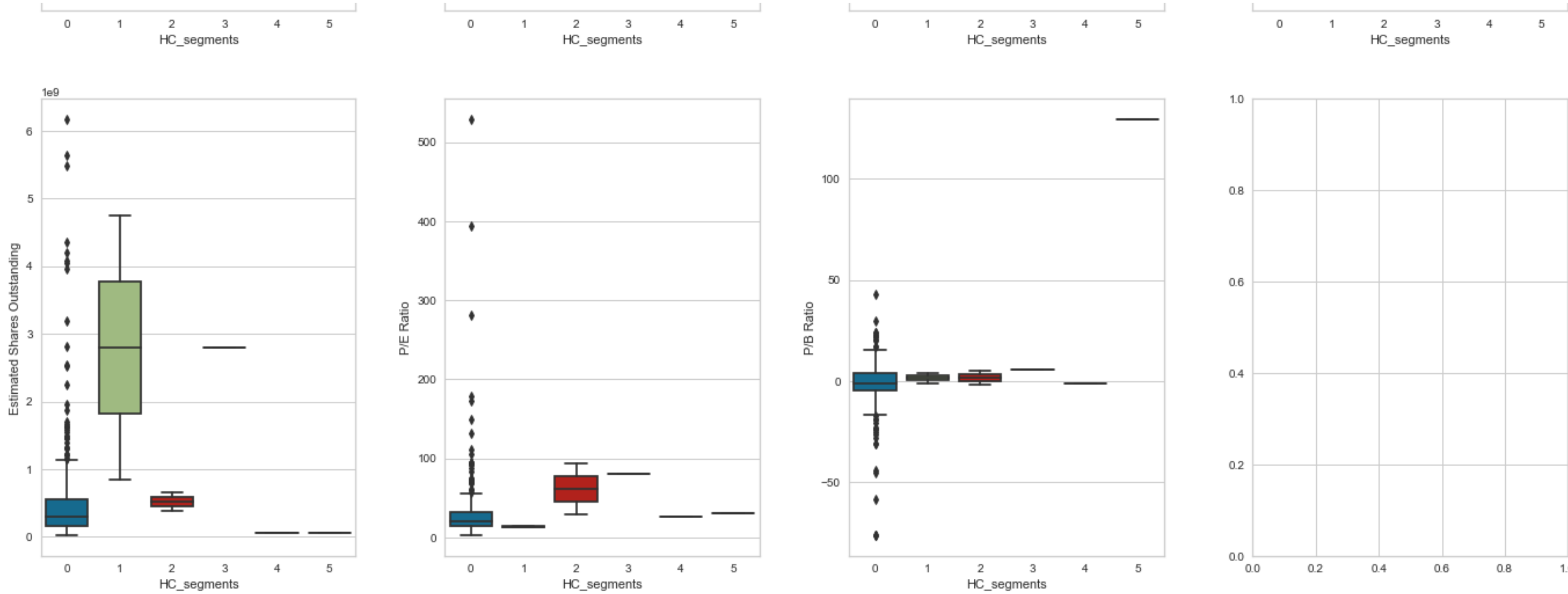
```
HC_segments  GICS Sector
0            Consumer Discretionary         39
             Consumer Staples               19
             Energy                         28
             Financials                     48
             Health Care                    40
             Industrials                    53
             Information Technology         30
             Materials                      20
             Real Estate                    27
             Telecommunications Services     5
             Utilities                      24
1            Financials                      1
             Information Technology          1
2            Energy                          2
3            Information Technology          1
4            Consumer Discretionary          1
5            Information Technology          1
Name: Security, dtype: int64
```

In [67]:

```
fig, axes = plt.subplots(3, 4, figsize=(20, 20))
counter = 0
```

```
for ii in range(3):
    for jj in range(4):
        if counter < 11:
            sns.boxplot(
                ax=axes[ii][jj],
                data=df2,
                y=df2.columns[4+counter],
                x="HC_segments",
            )
            counter = counter + 1

fig.tight_layout(pad=3.0)
```





Insights

- Looking at Clusters 0-2 since the rest are very small.
- Cluster 0
    - Largest number of companies by a large margin
    - Moderate current price
    - Moderate price change, volatility, ROE, Estimated shares outstanding
    - Many outliers in Net Cash Flow & Net Income
    - P/E Ratio on the lower end
- Cluster 1
    - Bank of America and Intel Corp.
    - Low current price
    - Moderate price change, volatility, ROE, net cash flow
- Cluster 2
    - Apache Corp and Chesapeake Energy
    - Very atypical compared to other clusters
    - High volatility, ROE
    - Low earnings per share, estimated shares outstanding, net cash flow, net income

K-means vs Hierarchical Clustering

- K Means executed immediately, compared to Hierarchial Clustering taking longer
- Appropriate number of clusters determined to be:
  - K Means: 3
  - Hierarchial: 6 (noted that the latter 3 of the 6 clusters have very little data)
- More distinct clusters obtained from K Means
- Cluster 0 in both styles, and Cluster 2 in both styles were very similar
  - Cluster 0: majority Consumer Staples and Consumer Discretionary
  - Cluster 2: majority Energy, most atypical when compared to other clusters (high volatility, low price change and net cash flow/income)

Actionable Insights and Recommendations

- Cluster 0 (K Means and Hierarchial): These are composed of companies that most consumers will encounter in their day to day lives. **Those looking to maximize earnings with little risk should approach this cluster to invest.**
- Cluster 1 (K Means): Composed of financial groups and health care. During a period of cyclical unemployment, avoid investing in banks and loan holdings. **On the upswing of market expansion, consumers can invest in these areas. With healthcare, look towards market trends and the latest innovations within these companies (notably Merck and Pfizer) to decide when to invest to maximize earnings.**
- Cluster 1 (Hierarchial): Composed of Bank of America and Intel. **Invest in periods when consumers can feasibly make a profit.**
- Cluster 2(K Means and Hierarchial): Composed of majority energy and oil corporations. With high volatility, and a high cash ratio (total cash reserves:total liabilities), **consumers that are more risk seeking may invest during times of a stable economy to maximize chances of making a profit.**