****

REPORT ON FINANCIAL MARKET

ACFI827 - Introduction to Python

Submitted by

ID: 201669904  
MSc in Financial Technology  
University of Liverpool Management School

**January 2023**

Submitted to

DR. Swati Sachan  
University of Liverpool Management School

Table of Contents

[Table of figures 2](#_Toc124502877)

[TASK 1 3](#_Toc124502878)

[1.1. Import the financial dataset and display first eight rows 3](#_Toc124502879)

[1.2. Shape of the financial data set 3](#_Toc124502880)

[TASK 2 3](#_Toc124502881)

[2.1. Process missing values in the financial data set 3](#_Toc124502882)

[2.2. Basic statistical descriptions of numerical columns in the financials data set 5](#_Toc124502883)

[2.3. Basic statistical descriptions of categorical columns in the financials data set 7](#_Toc124502884)

[TASK 3 7](#_Toc124502885)

[3.1. Plot the ranks of the percentage of stocks in each sector 7](#_Toc124502886)

[3.2. Correlation between all features 9](#_Toc124502887)

[3.3. Find the name of the company which has Maximum or Minimum value 10](#_Toc124502888)

[3.4. Calculate the probability of dividend yield greater than 2.181 11](#_Toc124502889)

[3.5. Barplot Earnings/Share, Dividend Yield and Market Cap for each market sector 12](#_Toc124502890)

[TASK 4 14](#_Toc124502891)

[Appendices 19](#_Toc124502892)

[Apendix A. Treemaps of all sectors of financials data set 19](#_Toc124502893)

[Apendix B. Codes of the report 22](#_Toc124502894)

[References 25](#_Toc124502895)

# **Table of figures**

[Figure 1. First 8 rows of financials data set 3](#_Toc124502054)

[Figure 2. Financials data set’s shape 3](#_Toc124502055)

[Figure 3. The proportion of missing values in each column of financials data set 4](#_Toc124502056)

[Figure 4. Standard deviation of each feature by sectors 4](#_Toc124502057)

[Figure 5. The number of missing values in each column of financials data set 5](#_Toc124502058)

[Figure 6. Checking result of missing values in the financials data set after processing 5](#_Toc124502059)

[Figure 7. Basic statistical descriptions of numerical columns in the financials data set 6](#_Toc124502060)

[Figure 8. Basic statistical descriptions of categorical columns in the financials data set 7](#_Toc124502061)

[Figure 9. Treemap (Heatmap) of the percentage of stocks by sectors 8](#_Toc124502062)

[Figure 10. Treemap (Heat map) of the percentage of stocks in Tech Firm sector 8](#_Toc124502063)

[Figure 11. Correlation heatmap of financial data set features 9](#_Toc124502064)

[Figure 12. Companies with maximum and minimum value of Dividend Yield, Earnings/Share and Price/Sales 10](#_Toc124502065)

[Figure 13. Probability of dividend yield greater than 2.181 for Real Estate, Financials and Energy 11](#_Toc124502066)

[Figure 14. Barplot of Earnings/Share for each market sector 12](#_Toc124502067)

[Figure 15. Barplot of Dividend yield for each market sector 13](#_Toc124502068)

[Figure 16. Barplot of Market\_Cap for each market sector 14](#_Toc124502069)

[Figure 17. Power Consumption and Carbon Emission Estimates in Previous Studies 15](#_Toc124502070)

[Figure 18. Ranks of Bitcoin mining emissions with countries 15](#_Toc124502071)

[Figure 19. Treemap (Heat map) of the percentage of stocks in Tech Firm sector 19](#_Toc124502072)

[Figure 20. Treemap (Heat map) of the percentage of stocks in Financials sector 19](#_Toc124502073)

[Figure 21. Treemap (Heat map) of the percentage of stocks in Healthcare sector 19](#_Toc124502074)

[Figure 22. Treemap (Heat map) of the percentage of stocks in Consumer Discretionary sector 20](#_Toc124502075)

[Figure 23. Treemap (Heat map) of the percentage of stocks in Industrial sector 20](#_Toc124502076)

[Figure 24. Treemap (Heat map) of the percentage of stocks in Consumer Staples sector 20](#_Toc124502077)

[Figure 25. Treemap (Heat map) of the percentage of stocks in Energy sector 21](#_Toc124502078)

[Figure 26. Treemap (Heat map) of the percentage of stocks in Material sector 21](#_Toc124502079)

[Figure 27. Treemap (Heat map) of the percentage of stocks in Real Estate sector 21](#_Toc124502080)

[Figure 28. Treemap (Heat map) of the percentage of stocks in Utilities sector 21](#_Toc124502081)

# **TASK 1**

## 1.1. Import the financial dataset and display first eight rows

A data set in .csv file can be imported in Python using Pandas. The process of importing file and displaying first eight rows of the data set is:

Step 1. ﻿Get path of the file

Step 2. Import the financials dataset in Python using pandas DataFrame (pd.read\_csv). Now the data set is saved to Python and ready for using.

Step 3. Use df.head(8) command to display the first 8 rows of the data set.

Results of displaying the first eight rows of the data set are all available features of the first eight companies (Cement Company, AT industries, life Laboratories…), in this slice of the dataset, there seems to be only missing data of the Price of Blizzard stock and its EBITDA. Codes and results of the process are provided in the following figure:

**Figure 1. First 8 rows of financials data set**

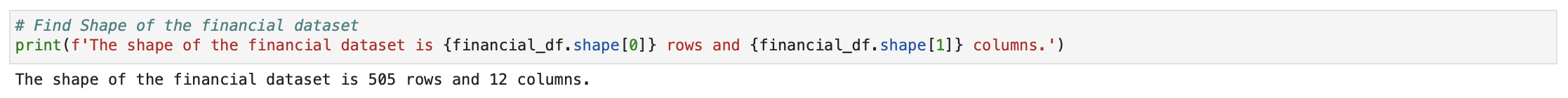


Source: Author’s work using Jupyter Lab IDE

## 1.2. Shape of the financial data set

It is quite simple to find shape of a data frame in Pandas via the command of df.shape(0) for number of rows and df.shape(1) for the number of columns. The output shows that the financial data frame has a shape of 505 rows and 12 columns.

**Figure 2. Financials data set’s shape**

****

Source: Author’s work using Jupyter Lab IDE

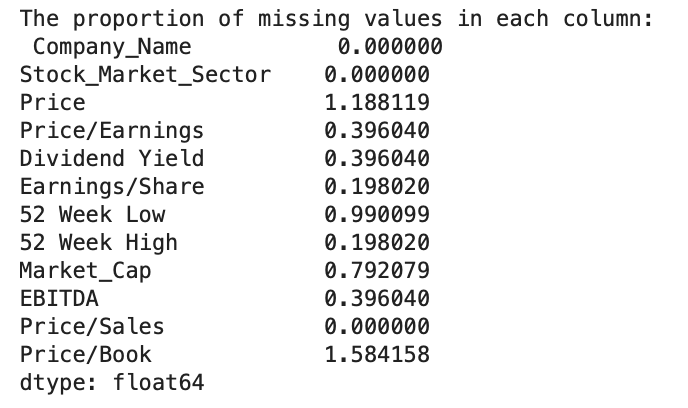
# **TASK 2**

## 2.1. Process missing values in the financial data set

The financial data set is not a time series of a single feature (index) but a cross-sectional data set which means the use of backward/forward fill and interpolation would not be appropriate here and can reduce the accuracy of the data. These ways of using neighbor data of the same column (from other companies of different industry) are not appreciated since companies from different industries would significantly differ in stock prices, so do the other indices.

Looking into provided indices, statistics indicate that the proportion of missing data is quite small in comparison to the total number of observations (mostly below 1.5%, see Figure 3) so it is grounded to use univariate methods of replacing missing values with mean, median or mode of the same sector data. With high degree of volatility (high standard deviation and outliers), median of each feature categorised by sectors is used to fill in missing values’ places.

**Figure 3. The proportion of missing values in each column of financials data set**



Source: Author’s work using Jupyter Lab IDE

**Figure 4. Standard deviation of each feature by sectors**



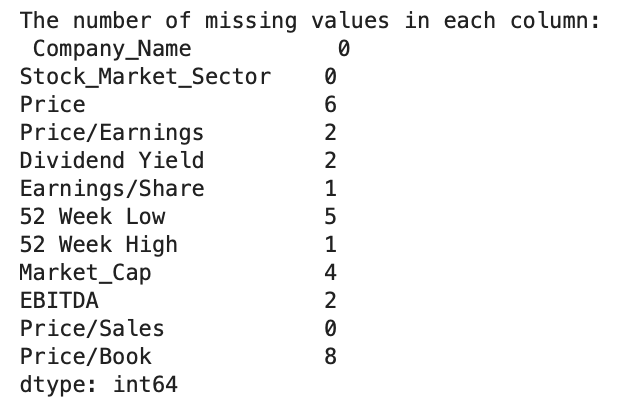
Source: Author’s work using Jupyter Lab IDE

With the abovementioned method, the process of detecting missing values and replacing them in Python is described as following:

﻿ Step 1. Loop over columns index 2 to 11 of the financial data set to convert these values into numerical data. ﻿From observation, data in columns index 0 and 1 (Company\_Name and and Stock\_Market\_Sector) have no missing value and processing of such missing values (if any) is both meaningless and complexible. Therefore, missing values in columns index 2 to 11 will be processed.

Step 2. Any value which cannot be converted into numerical will be automatically dectected as a missing value by Python. The number of missing values in each column is presented in Figure 5 It can be seen that the most missing values belong to Price/Book and there is no missing value in Company\_Name, Stock\_Market\_Sector, Price/Sales.

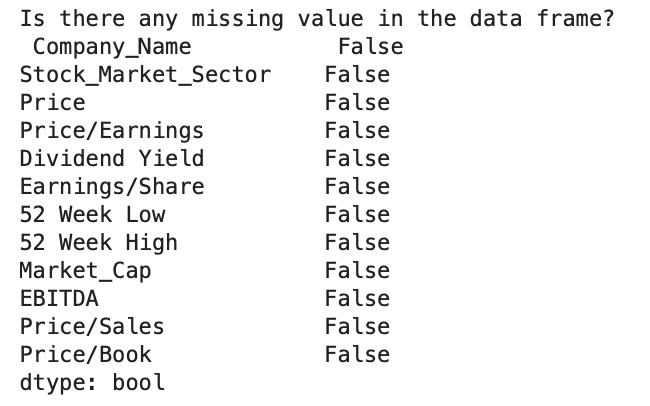
**Figure 5. The number of missing values in each column of financials data set**



Source: Author’s work using Jupyter Lab IDE

Step 3. Replace missing values with median of each feature categorised by sector. The checking result shows that there is no missing value left in the data frame. Now, the data set is ready for analysis.

**Figure 6. Checking result of missing values in the financials data set after processing**



Source: Author’s work using Jupyter Lab IDE

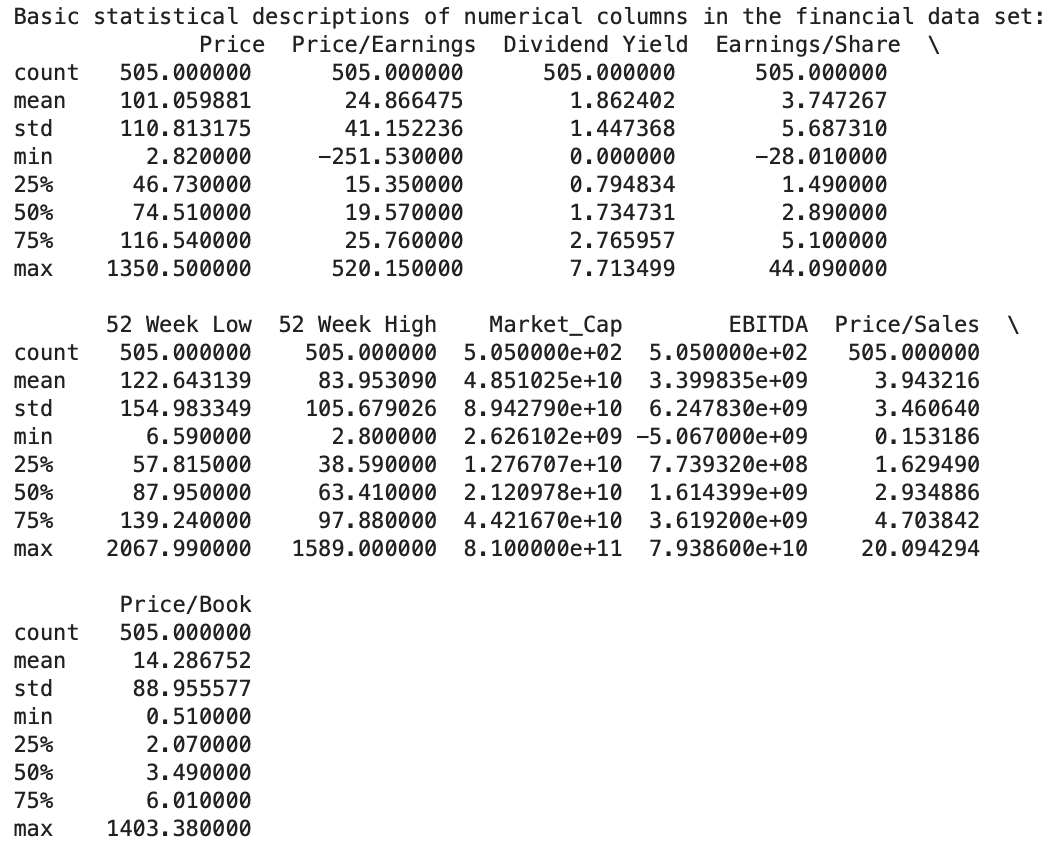
## 2.2. Basic statistical descriptions of numerical columns in the financials data set

Using Pandas’s df.describe(), all fundamental statistical characteristics of numerical features of the data set are generated. There are 10 numerical columns in the data set and the number of observations across all features are 505 which matches the number of rows in the dataset. Most of the values are positive except for the min of Price/Earnings, Earnings/Share and EBITDA. Negative min of Price/Earnings, Earnings/Share and EBITDA can be explained by poor performances of some companies. In addition, these indices positively correlate as Earnings are calculated from EBITDA by deducting interest, taxes, depreciation and amortization.

The feature with the highest mean (48,510,025,000), max value (810,000,000,000), median value (2,120,978,000) and standard deviation (89,427,900,000) is obviously Market\_Cap as this index equals to price multiplied by number of shares which are huge. Furthermore, companies of different industries are far more distinct in their scale of capitalization which results in a high degree of standard deviation. This enormous standard deviation also indicates the diversity in company size that this data set takes into account.

In contrast, Dividend Yield has the smallest mean, median value and standard deviation, which are 1.86, 1.73 and 1.45 respectively, due to two reasons. Firstly, this index is small in nature in comparison with other indices as it is just a small amount from one company’s retained earnings which is paid to the investors. Besides, it is common for companies not to pay dividends to their investors yearly. This might come from the financial situation of the companies and their plan of funding other projects. The lowest min value is of EBITDA which is -5,067,000,000. This number indicates the alarming financial situation of a surveyed company. One strange point of this data set is the mean of 52 week high mean is lower than mean of 52 week low, this might arise from data preparation and processing which requests more work on data validation and consolidation.

**Figure 7. Basic statistical descriptions of numerical columns in the financials data set**

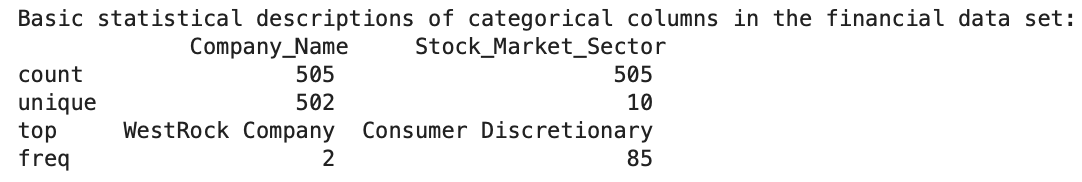


Source: Author’s work using Jupyter Lab IDE

## 2.3. Basic statistical descriptions of categorical columns in the financials data set

There are only two columns classified as categoricalin the financials data set and they are Company\_Name and Stock\_Market\_Sector with the number of observations is 505. The Company\_Name category has 502 unique values (different company names) while Stock\_Market\_Sector has only 10 unique industries. The number of unique company names implies that there are 3 company names that appears twice in the data set, one of which is displayed in the statistical description table (WestRock Company). Simply checking the WestRock Company in the data file, it is a duplication of observation which might result from data processing. The industry with the most number of companies (85 companies) in the data set is Consumer Discretionary.

**Figure 8. Basic statistical descriptions of categorical columns in the financials data set**

****

Source: Author’s work using Jupyter Lab IDE

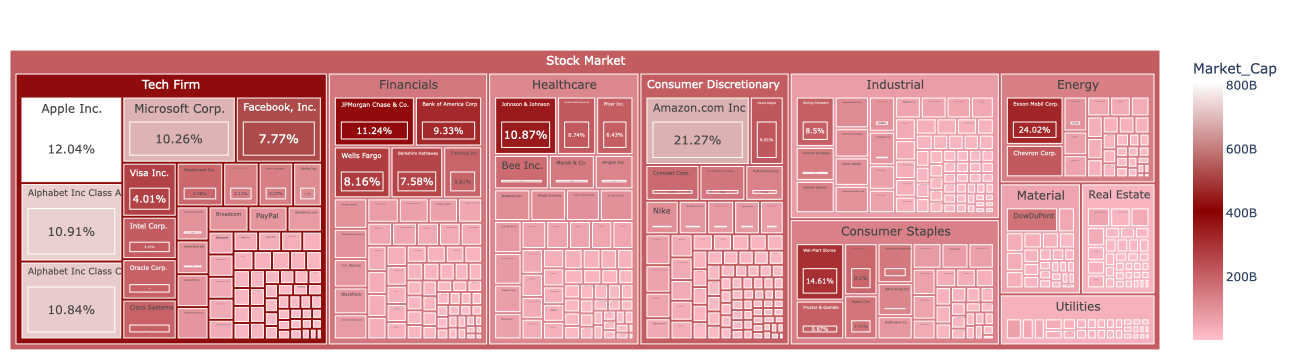
# **TASK 3**

## 3.1. Plot the ranks of the percentage of stocks in each sector

In stock market, one of the most important indices of a company is its market capitalization which enables investors to have a comparative sense of the relative size of one company versus others. Furthermore, market capitalization is the indicator of one company’s stage of performance which is more stable in the short to medium run while price and other indices basing on price are daily changeable. Therefore, the use of market capitalization for plotting ranks of stocks in each sector is much more meaningful and practical for a cross sectional data set (at only one point of time) like the financials file.

On the basis that there are 505 observations (stocks) listed in the data set and the difference between market capitalization of companies are huge, it is fairly not informative to use plots of pie chart, bar chart or boxplot. The most commonly used type of plot in stock market is the heatmap which facilitates an informative and comparative comprehension of stocks’ data. Nested heatmap of different sectors stocks is even more rich in data and visuality, which is called a treemap. A treemap presents not only data of stocks in the same sector but also relative scales of different sectors in the market. For these reasons, a treemap of rankings of the percentage of stocks in each sector is used here to illustrated comparative information among companies in the same sector and cross-sector. In addition, this type of map using px.treemap library in python is interactive which can expand to display further information of one sector and one company.

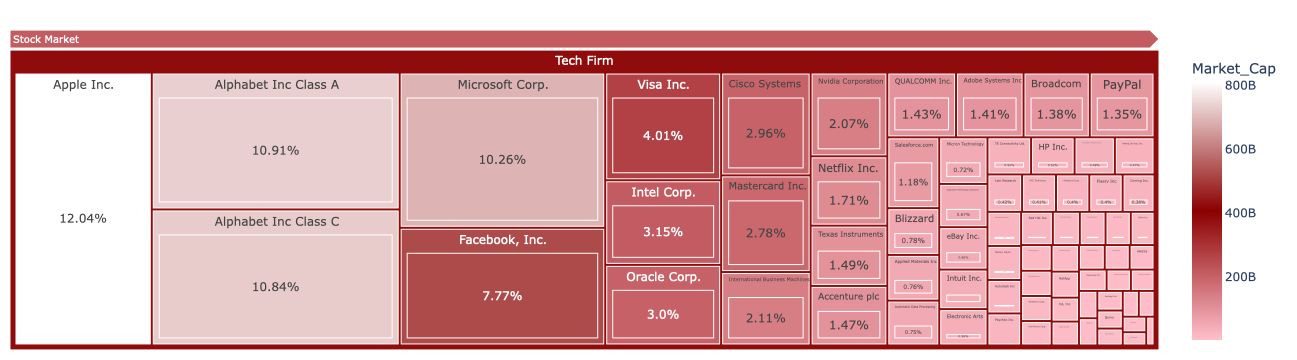
**Figure 9. Treemap (Heatmap) of the percentage of stocks by sectors**



Source: Author’s work using Jupyter Lab IDE

The treemap of the overall stock market gives the total value of market cap at approximately 24,497.7 billion. From the treemap of the overall stock market, it is undoubtful that Tech Firm (6,727.3 billion) is the biggest sector by market capitalization with its highest rank is Apple Inc. which comprises of 12.04% of the total sector capital. Apple is followed by three giants in technology which are Alphabet Inc. Class A and Class C (widely known as Google), Microsoft Corp. with their market cap occupy 10.91%, 10.84% and 10.26% of the sector respectively. The smallest company of this sector is CSRA Inc. with just over 0.08%.

**Figure 10. Treemap (Heat map) of the percentage of stocks in Tech Firm sector**



Source: Author’s work using Jupyter Lab IDE

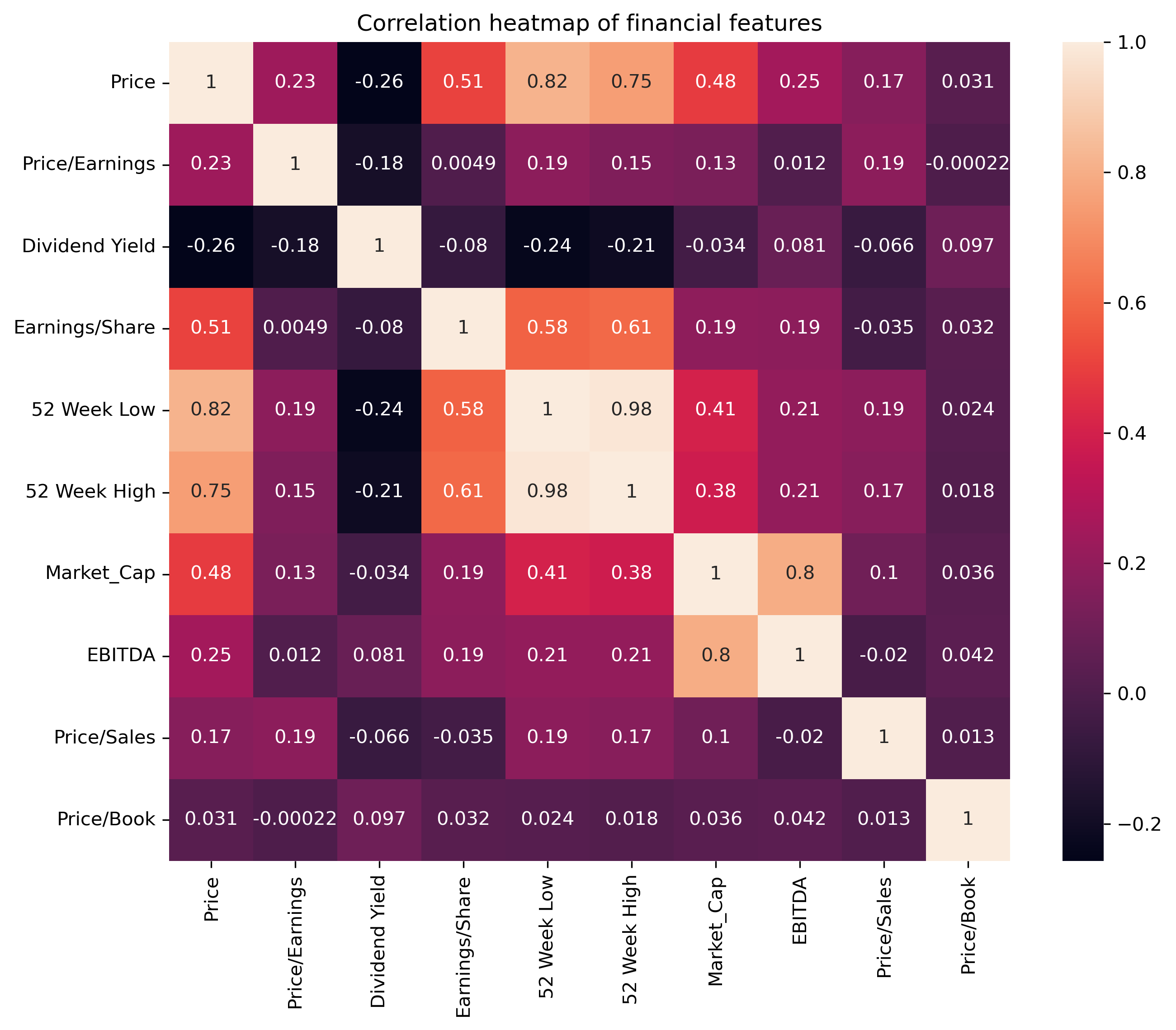
The second largest sector is Financials (3,441.2 billion) and its top rank stocks are JPMorgan Chase&Co (11.24%), Bank of America (9.33%) and Wells Fargo (8.16%). The humblest in market cap sector is Utilities with 625.2 billion in value. In Utilities, NexEra Energy and Duke Energy win the first and the second place with the value of 11.14% and 8.33% respectively. Further information of other sectors is given in treemap for each sector in the appendix.

In conclusion, investors can assess other factors to make their investment decision however market capitalization remains a trustworthy source of one company’s financial state. To this extent, Tech firm company stocks are in a good position in terms of market cap and this is also a core industry in the current innovation-driven world. However, this does not imply investing in lower ranked sector, i.e utilities sectors is not efficient as with high barriers of entry, the number of companies in this sector is clearly smaller than others which results in its low rank in market cap.

## 3.2. Correlation between all features

A statistical indicator of the strength of a linear link between two variables is the correlation coefficient. The Pearson correlation coefficient is calculated by dividing the covariance by the sum of the standard deviations of the two variables (Investopia, 2021). In stock market, knowing the relations between two indices of a stock helps predicting one on each other, especially the stock price. Using a heatmap facilitates a visual image of correlations between variables categorized by colors so that data analyst and researchers can choose potential indices for their model building process. In python, seaborn and matplotlib.pyplot libraries are commonly used to plot this type of map, details of approach and codes are given in the appendix.

**Figure 11. Correlation heatmap of financial data set features**



Source: Author’s work using Jupyter Lab IDE

The heatmap color scale marks higher correlations with lighter color so one feature which is perfectly correlated with itself is marked beige color. In contrast, the lower correlations are colored darker so some of the negative values are nearly black. As such, the most correlated features are 52 Week high and 52 Week low at 0.98 which can be interpreted that they near perfectly correlate in a positive manner, if the 52 Week high value goes up, this in 52 Week low will increase at high probability. This is reasonable in stock market as price of one stock peaks a higher level will pull along higher value of its lowest price. As outliers of Price, 52 Week high and 52 Week low obviously have high correlation values with price (0.75 and 0.82 respectively) and this can also be explained by the reason mentioned.

In orange colors, Earnings/Share correlates with Price, 52 Week high and 52 Week low with correlation values of more than 0.5 which are fairly strong. Market\_Cap also has a moderate links with Price, 52 Week high and 52 Week low of around 0.4. It is interesting to find out that since Price, 52 Week high and 52 Week low have nearly perfect correlations, they go together and share the nearly similar values of correlation when pairing with other variables. Fair correlation values between Earnings/Share or Market\_Cap and one of Price, 52 Week high or 52 Week low can be reasoned by the fact that these variables are computed based on price of each stock. Market\_Cap and EBITDA are also highly correlated at 0.8 as market cap is a factor in the calculator of EBITDA (CFI, 2022).

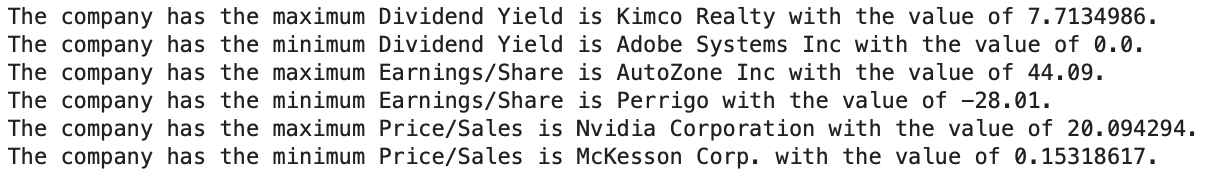
The negative and lowest values of correlations are between Dividend Yield and Price (-0.26). Consequently, Dividend Yield is also negatively correlated with 52 Week high and 52 Week low (-0.21 and -0.24). Negative correlations indicate that if dividend yield increases, Price, 52 Week high and 52 Week might go down. In fact, a stock dividend lowers the book value per common share and lowers the stock price since it increases the number of shares outstanding while the company's worth stays the same (Investopia, 2021).

From above observations, it might be meaningful to do further research into the relations among Price, 52 Week high, 52 Week low, Market\_Cap, EBITDA and Dividend Yield to construct models of data analysis, especially Price forecasting.

## 3.3. Find the name of the company which has Maximum or Minimum value

With Python, it is not difficult to find maximum and minimum value of a feature, however if associated data of that value is requested (in this case is the company name), more steps have to be involved. The approach is using ﻿df.idxmax() to get the row index of that value and use this index to access value of the column ‘Company\_Name’. Detail codes and explanation of each step are enclosed in the appendix.

**Figure 12. Companies with maximum and minimum value of Dividend Yield, Earnings/Share and Price/Sales**



Source: Author’s work using Jupyter Lab IDE

With Dividend Yield feature, Kimco Realty is the company which has the maximum value of Dividend Yield while the minimum value of 0 belongs to Adobe System Inc. This means recorded data point out Kimco Realty pays dividend of 7.713% of its current share price to the investors whereas Adobe System Inc did not pay any to theirs. As explained in the previous part, this can be interpreted as good and bad financial status of that company or in other case, Adobe is retaining earnings for their business plan. However, it can affect the appeal of their stocks in the market.

Concerning Earnings/Share, AutoZone Inc wins the first place with 44.09 which implies they make a net profit of 44.09 per share of their outstanding stock. Hence, they are the best profitable company in terms of net income per share and might be an attractive destination of investing. In contrast, Perrigo yields -28.01 net profit per share which means they are doing business at a loss. Again, this might be an indicator that investor should consider the decision of investing in this company.

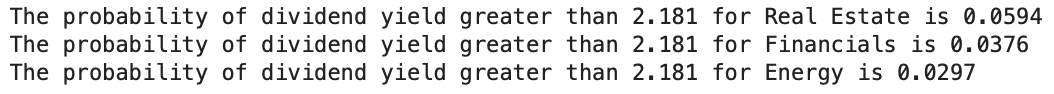
In terms of Price/Sales, Nvidia has the highest index of 20.094 which is above the average (mean) of that feature in the data set (3.94), thus it is questionable that if this stock is overpriced. On the other hand, McKensson Corp.’s stock is just priced at 0.153 per dollar of its sales and is possibly undervalued. Therefore, investing in McKensson Corp.’s stock might produce favourable investment results for investors.

## 3.4. Calculate the probability of dividend yield greater than 2.181

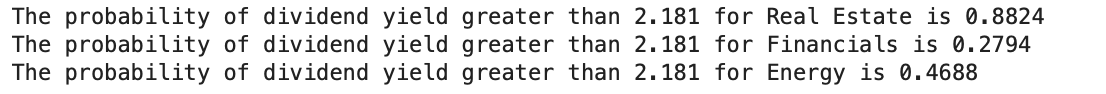
To find the probability of dividend yield greater than 2.181 for a sector, three necessary steps are taken. Firstly, categorising data frame into sectors using logical expression of df[‘Stock\_Market\_Sector’] == sector (this can also be done by df.groupby()). Secondly, comparing dividend yield value of each company in the sector with 2.181. Finally, count the number of satisfied companies and divide by the number of companies in the sector or the number of companies in the whole data set. To make this process easier and faster, function and loop are used. Codes and steps of executing this is provided in the appendix.

**Figure 13. Probability of dividend yield greater than 2.181 for Real Estate, Financials and Energy**

**Method 1: Probability with population of all companies in the data set**



**Method 2: Probability with population of companies in the same sector**



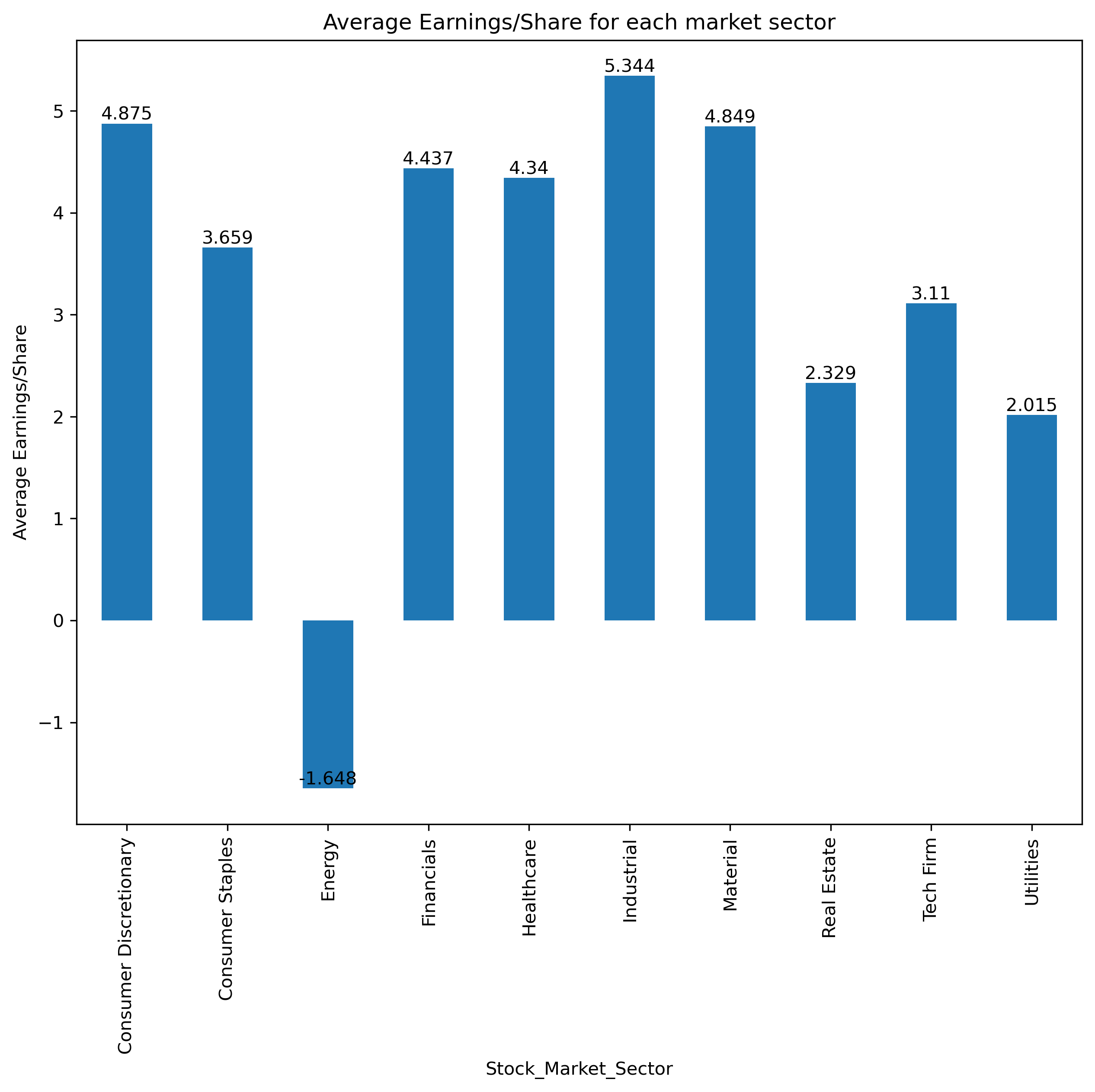
Source: Author’s work using Jupyter Lab IDE

If the population of all companies in the data set is used, the probability of Real Estate, Financials and Energy companies to yield dividend greater than 2.181 are only 0.0594, 0.0376 and 0.0297 respectively. This means it is only approximately 6% of chance, one random company in the data set is of Real Estate sector and yield dividend of more than 2.181. Similar interpretation can be made to Financials and Energy sector. The other way round, concerning dividend yield, the number of Real Estate companies are doing well is more than the number of Financials and Energy companies, which is an indicator of financial status. Using population of companies in the same sector, with less number of companies, the probability of Energy sector is higher than Financials. Anyways, these probabilities can be used by investors to decide sectors of interest for investing.

## 3.5. Barplot Earnings/Share, Dividend Yield and Market Cap for each market sector

Barplotting of a feature (column) for each market sector can be conducted by grouping by sector using df.groupby() and df1.plot.bar() of matplotlib.pyplot library. Using function can save much time of this process. However, the most complexible task is annotating these plots as the values of Market Cap is extensively big. A conditional statement inside the function is used here for annotating Earnings/Share, Dividend Yield and Market Cap separately. Details are given in appendix of codes.

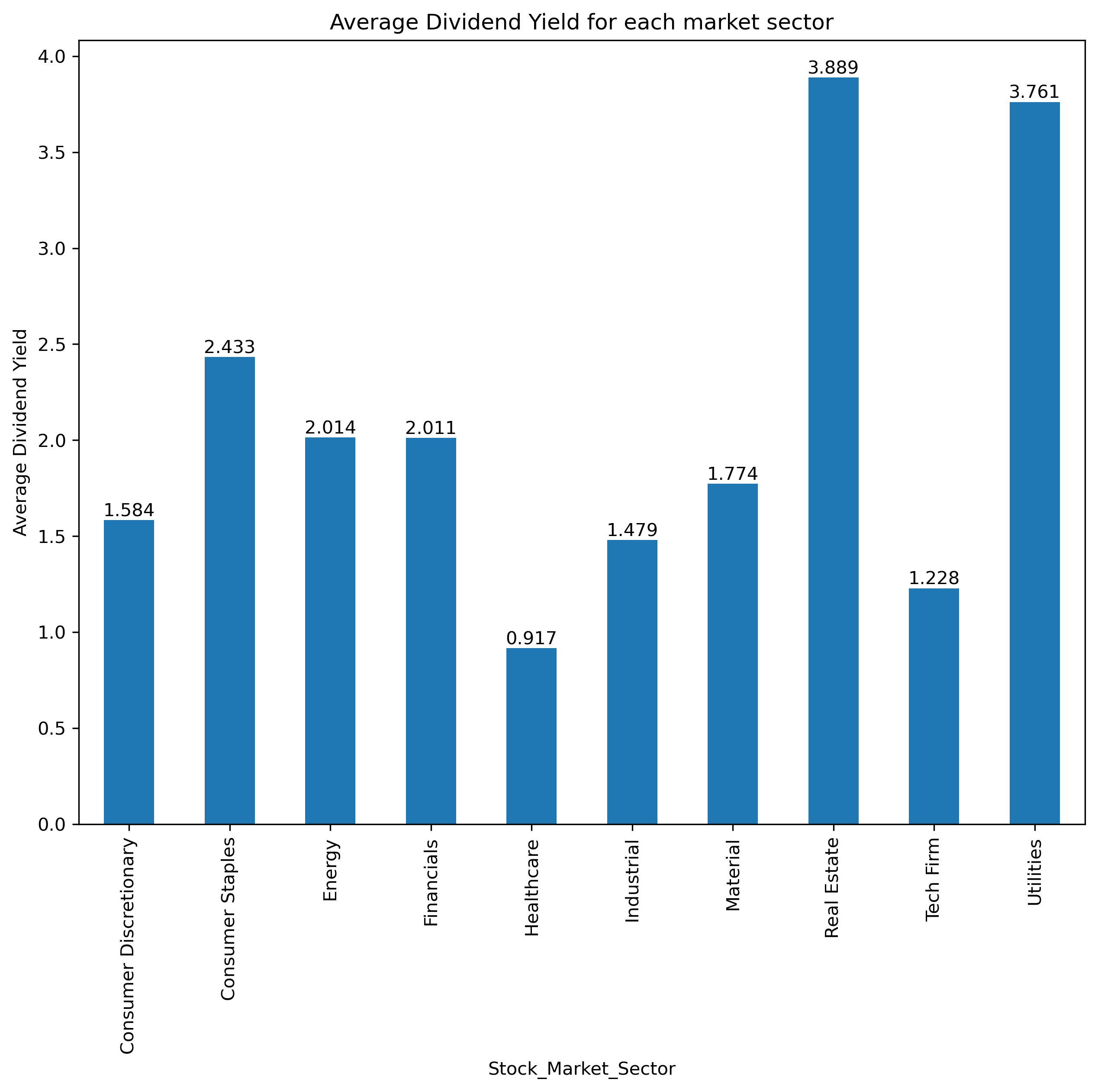
**Figure 14. Barplot of Earnings/Share for each market sector**



Source: Author’s work using Jupyter Lab IDE

From the plot of Earning/Share, it can be seen that most sectors produce positive earnings except for the energy sector. The highest Earnings/Share value is of Industrial sector which indicates that the net income of companies in this sector is high per their share. This might result from this sector’s high productivity and is a good indicator. Customer discretionary and Material have approximately same value of 4.8 which are the second most appealing sectors concerning the index of Earning/Share. Energy is the only sector that is making loss, this can pose a warning to investors about assessing investment in this sector.

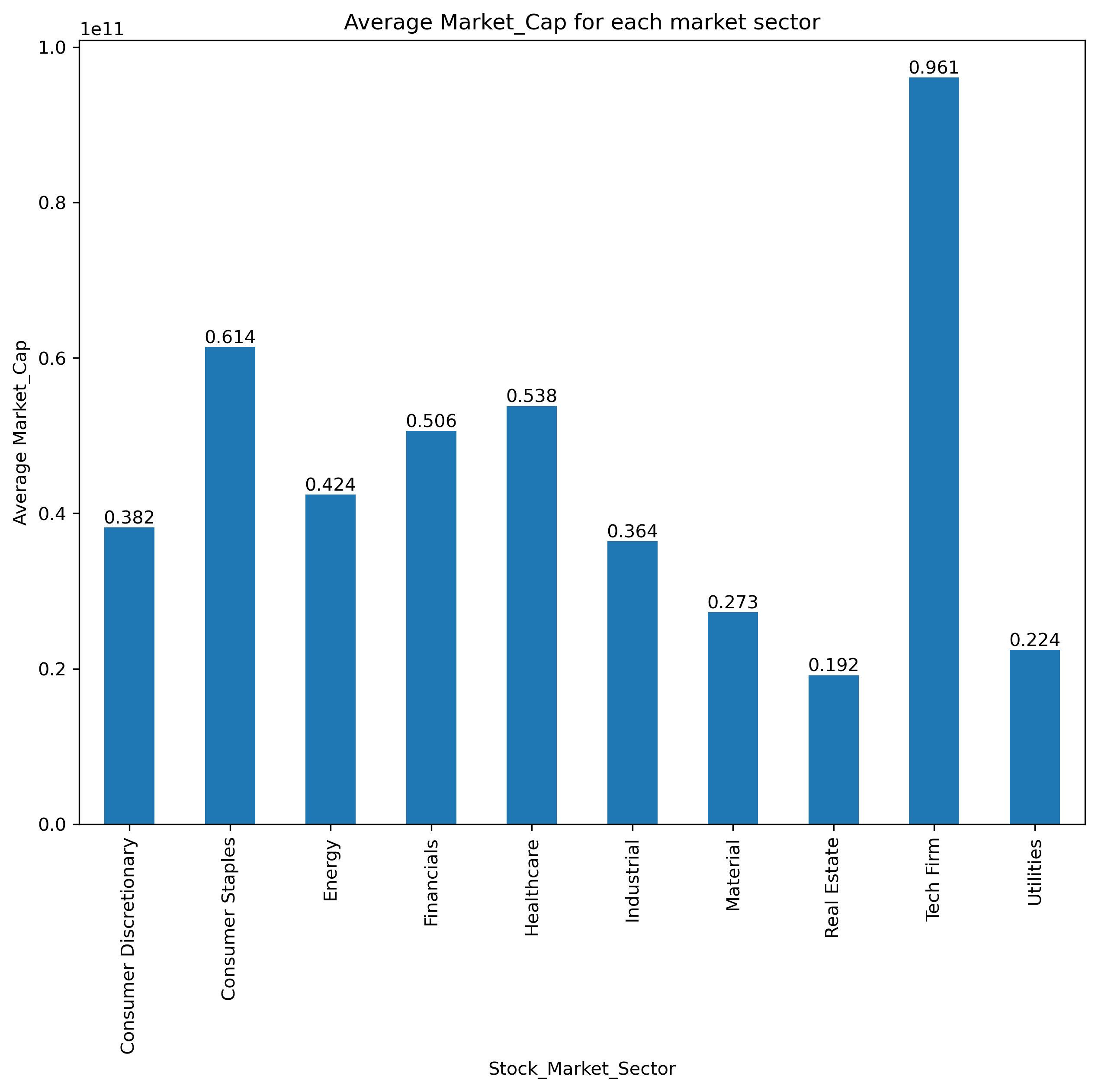
**Figure 15. Barplot of Dividend yield for each market sector**



Source: Author’s work using Jupyter Lab IDE

Barplot of Dividend yield for each market sector provides a comparison among sectors by their Dividend Yields. Real Estate pays out the most dividend of 3.889% of share price much higher than the lowest sector Healthcare with just 0.917%. Utilities is also high in percentage of stock price paid out as dividend. To name some sectors which are doing good in dividend payment, they are Real Estate, Utilities, Customer Staples, Financials and Energy. Investors considering this indices can take a deeper look at companies of these sectors.

**Figure 16. Barplot of Market\_Cap for each market sector**



Source: Author’s work using Jupyter Lab IDE

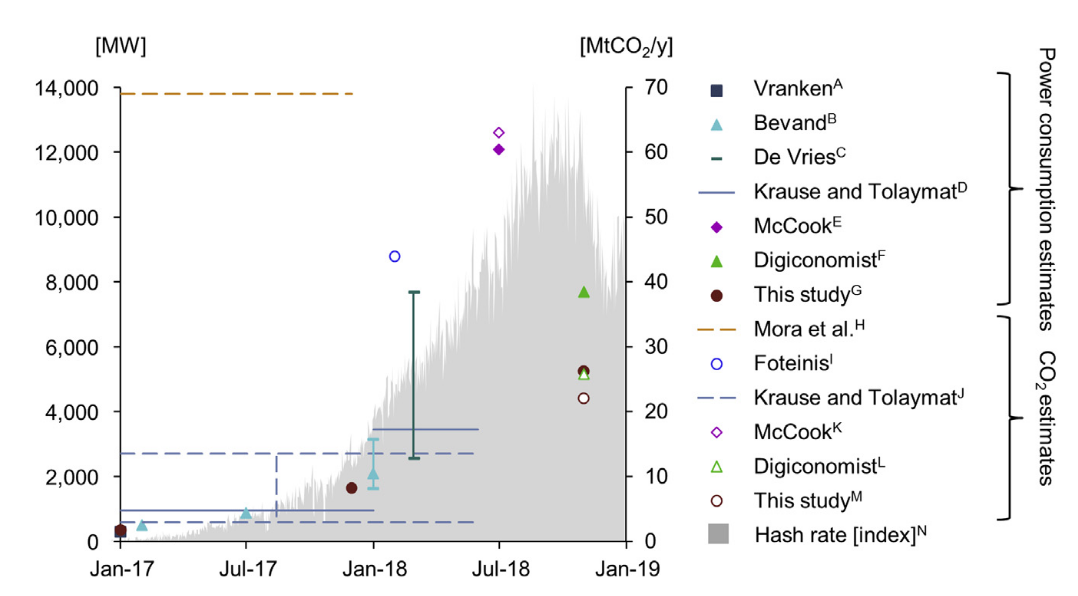
Plot of average Market\_Cap helps analysts to have a look at average size of companies in each sector in terms of market cap. Accordingly, Tech Firm has the biggest average market cap of 96.1 billion which is above 4 times higher than the smallest sector which is Utilities (22.4 billion). As mention in previous parts, Tech Firm is on the trend and is now core sector of the world with giants such as Apple, Google and Microsoft whereas Utilities with high barrier of entry and less booming motives are much less attractive and priced in the stock market. Customer Staples, Healthcare and Financials are sectors of average market cap over 50 billion.

# **TASK 4**

Blockchain is a decentralised, unchangeable database that makes it easier to track assets and record transactions in a corporate network. Blockchain is the best technology for delivering information of any kinds because it offers real-time, shareable, and entirely transparent data that is kept on an immutable ledger and accessible exclusively to members of a permissioned network. A sort of cryptographic evidence known as proof of work (PoW) involves one party (the prover) demonstrating to another (the verifiers) that a certain amount of computing effort has been applied. This protocol takes specialised technology and enormous quantities of power to participate in its validation process. This implies that it causes significant carbon dioxide (CO2) emissions and so has an impact on global warming. As a result, the biggest cryptocurrency using PoW, Bitcoin now uses the most energy among all public blockchain systems.

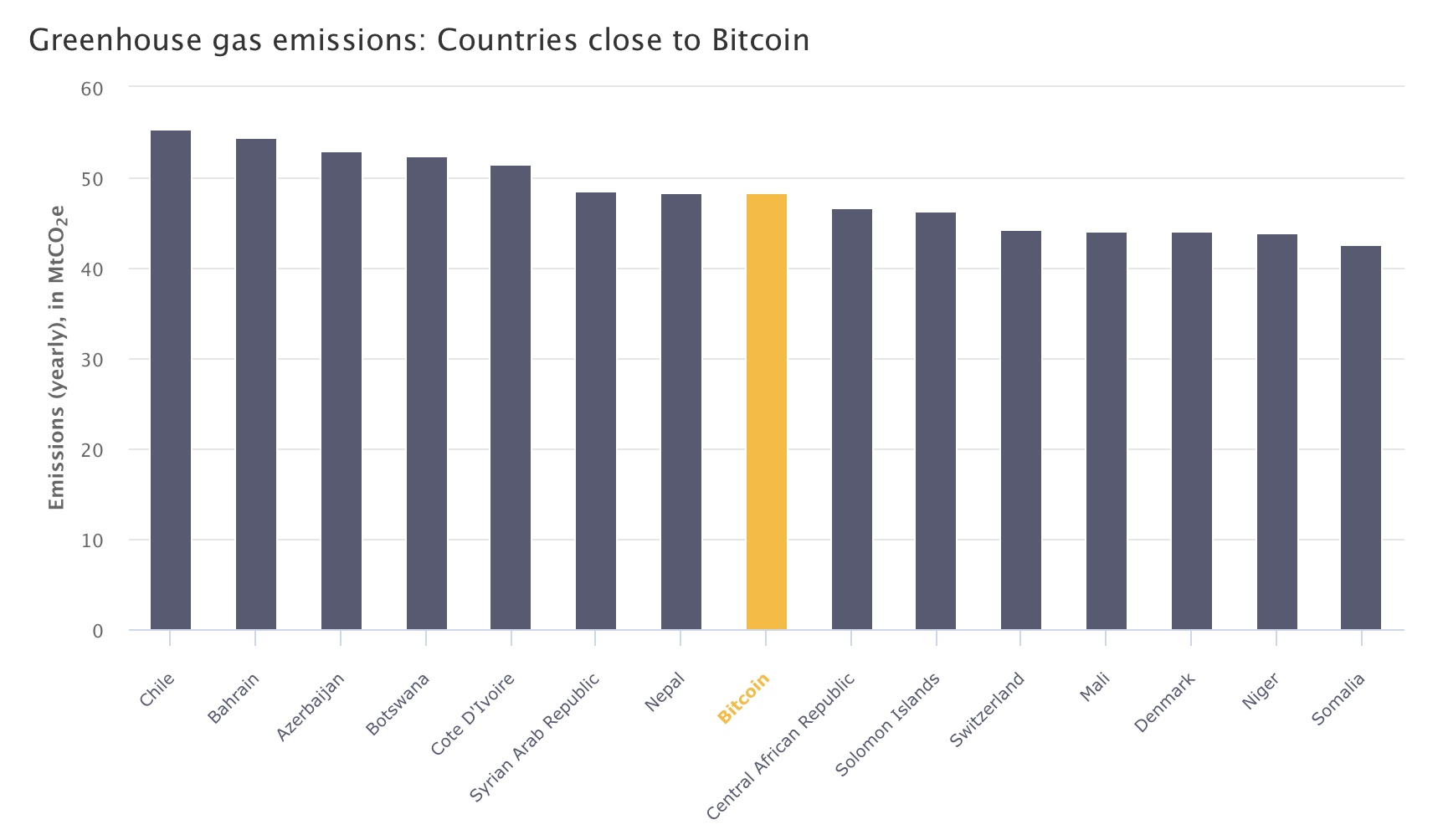
According to University of Cambridge estimations from 2018, the energy usage of Bitcoin is comparable to that of Nepal and three ranks higher than that of Switzerland. Prior academic research relied on imprecise estimates of power usage and lacked empirical support, such as projections of future carbon emissions or comparisons of bitcoin and metal mining. As a result, the estimations generated differ greatly between investigations. According to calculations made by Stoll, Klaaßen, and Gallersdörfer in 2019, the resulting yearly carbon emissions vary from 22.0 to 22.9 MtCO2.

**Figure 17. Power Consumption and Carbon Emission Estimates in Previous Studies**



Source: Stoll, Klaaßen and Gallersdörfer, 2019

**Figure 18. Ranks of Bitcoin mining emissions with countries**



Source: Climate Watch Historical GHG Emissions. 2022. Washington, DC: World Resources Institute. 2019 est.

For the above concerns about the significant effects of blockchain computational validation process, the issue of how to mitigate environment consequences from and by blockchain technology has become a central controversial topic. In general, empirical review shows that there are two considering solutions to this problem. The first one is to shift to other protocols of validation to resolve the issue of PoW energy consumption. The second key is to use blockchain to motivate and facilitate the creation of energy and exchange of emission scheme.

In extreme cases, authorities of countries are implementing restrictions and bans on PoW. In January 2022, Erik Thedéen, vice chairman of the European Securities and Markets Authority, urged the EU to replace the proof-of-work model with the proof-of-stake model in January 2022 because the latter has fewer energy emissions. The state of New York enacted a two-year ban on cryptocurrency mining in November 2022 for any operations that do not entirely rely on renewable energy. China, the largest Bitcoin mining pool, declared a ban on this activity in 2021.

The primary factors influencing power usage are the consensus procedures and the level of redundancy, or the quantity of nodes and the speed of the processes used to complete transactions. In order to use blockchains to increase sustainability, it is essential to apply alternate consensus processes to PoW. The power usage of alternative consensus blockchains protocols is lower than that of PoW blockchains. A Bitcoin transaction is reported to use 1 GJ, but each Ethereum transaction uses about 0.1 GJ because of its smaller market value. In contrast, public blockchains using a Proof of Stake (PoS) method only need 100 J for each transaction. With 1 J per transaction, private permissioned blockchains using Proof of Authority (PoA) as consensus algorithms use significantly less energy (Sedlmeir Et al., 2020 and Gil-Pulger, J., 2019).

Climate-conscious blockchain can also be used to motivate improvements on climate and emission issues. For instance, SolarCoin uses a blockchain network to reward solar energy providers with one free SolarCoin for every megawatt-hour of power they create. This digital award can be exchanged for other currencies or used as a medium of trade. The goal of initiatives like Earth Dollar is to connect blockchain currencies with carbon credits, which are environmental permits awarded for emissions averted elsewhere (representations of a particular asset or utility within the platform).

Developing blockchain-based apps is one way to address the current environmental concerns. The unique characteristics of the technology, which have a wide range of applications, can be crucial in protecting the environment and the climate. Blockchain's potential uses are listed by the United Nations Climate Change (2017) as safer and more effective for carbon emission trading and renewable energy trading platforms, the mobilisation of climate finance, and frameworks for monitoring greenhouse gas emissions.

According to Article 17 of the Kyoto Protocol, emissions trading enables countries who have emission units to spare emissions that are authorised but not "used" to sell this extra capacity to nations that are exceeding their objectives. The "Blockchain for Climate Foundation" creates an application with the goal of incorporating each country's NDC. By automating the verification procedure, the distributed consensus technique assures the accuracy of the recorded CO2 compensation data and lessens manual effort. In addition, the Swiss programme "Quartierstrom" offers neighborhood-level decentralised trading of locally produced power.

In Hongkong and Europe, similar programmes can also be found. Infinite Earth's Veridium Labs, a private firm with headquarters in Hong Kong, is integrating its Verde Pay payment system with carbon credits derived from Infinite Earth's forest reserve in Rimba Raya, Central Kalimantan. Blockchain tokens are being used by Ecosphere+, a division of Luxembourg-based Althelia, to distribute its carbon credits to middlemen. Using Ocean tokens, Poseidon's blockchain technology enables customers and merchants to monitor and offset their carbon footprints. Poseidon already has partnerships with the Liverpool City Council and the Ben & Jerry's store in London. Blockchain technology is used by Regen Network to exchange data, provide incentive payments to regional land stewards, and monitor and verify environmental performance.

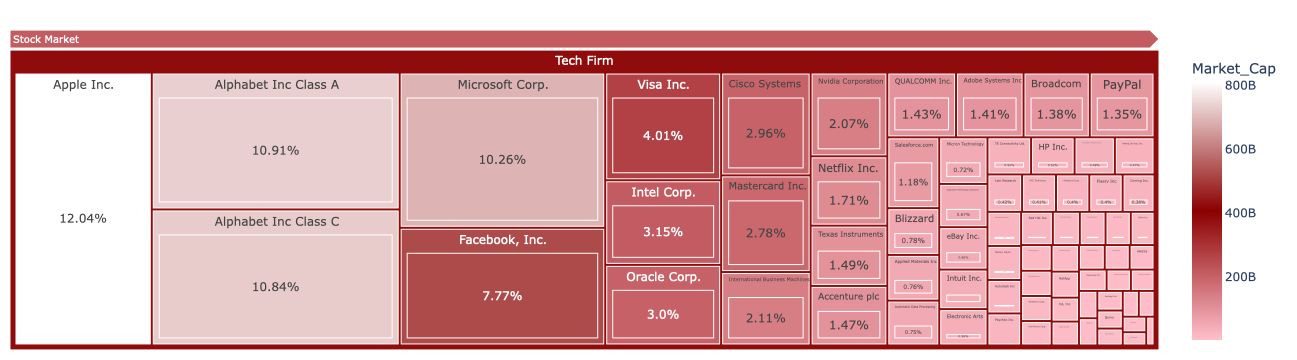
According to Howson (2019), although blockchain technology is enabling people and companies to monitor their carbon emissions, it is yet uncertain what the costs and benefits will be for society and the environment. Dorfleitner, Muck, and Scheckenbach (2021) discuss the application of blockchain technology for emission trading platforms and find evidence that the kind of activity strongly influences the likelihood of being operational. In addition, they demonstrate how important it is to choose the right consensus method. Additionally, the execution of an Initial Coin Offering (ICO) and the use of tokens have no bearing on the application's standing. Finally, they show that there are no differences between various blockchain types.

In conclusion, although there are many factors that are unknown, it is likely that the blockchain's underlying mechanism has a huge carbon footprint. To cut down on the level of carbon emissions from blockchain using and benefit from this novel technology, decision makers can choose among options of shifting to other validating protocols, using the blockchain itself to motivate the environment actions or to develop applications of carbon emission trading and regulating or in extreme case can impose restrictions on the highly energy-consuming protocols.

# **Appendices**

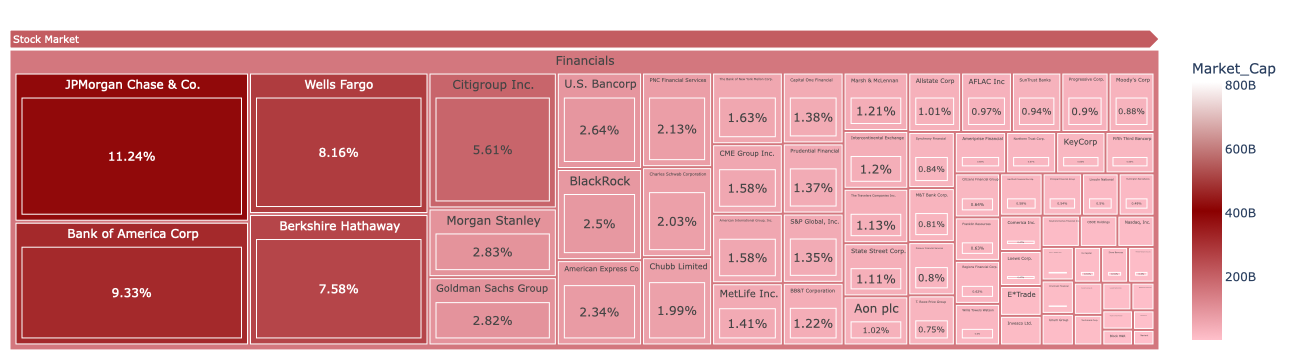
## Apendix A. Treemaps of all sectors of financials data set

**Figure 19. Treemap (Heat map) of the percentage of stocks in Tech Firm sector**



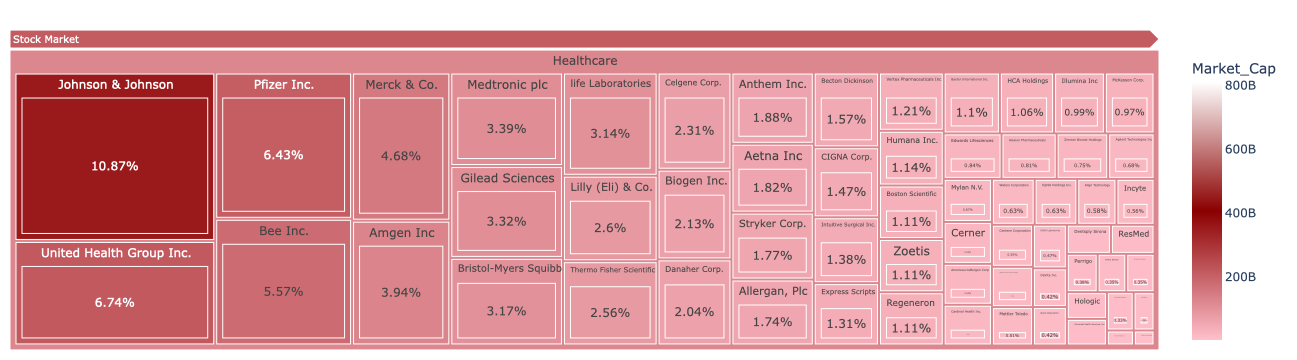
Source: Author’s work using Jupyter Lab IDE

**Figure 20. Treemap (Heat map) of the percentage of stocks in Financials sector**



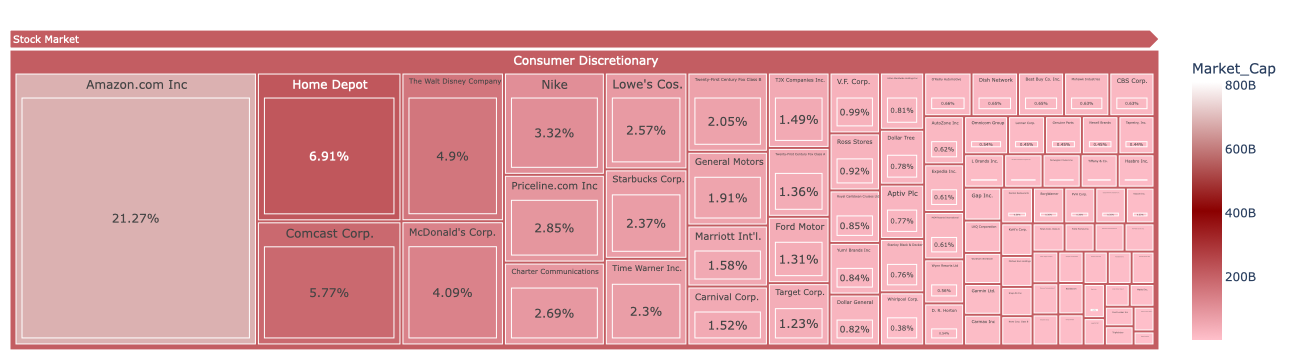
Source: Author’s work using Jupyter Lab IDE

**Figure 21. Treemap (Heat map) of the percentage of stocks in Healthcare sector**



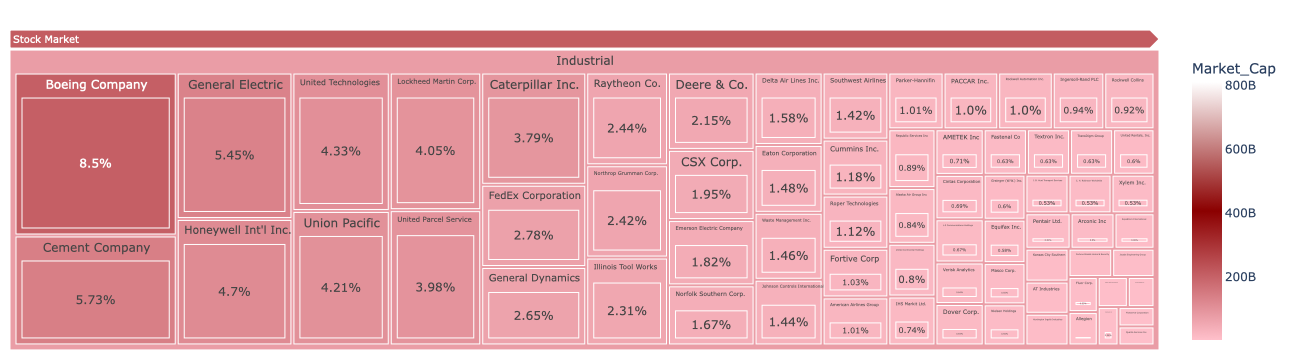
Source: Author’s work using Jupyter Lab IDE

**Figure 22. Treemap (Heat map) of the percentage of stocks in Consumer Discretionary sector**



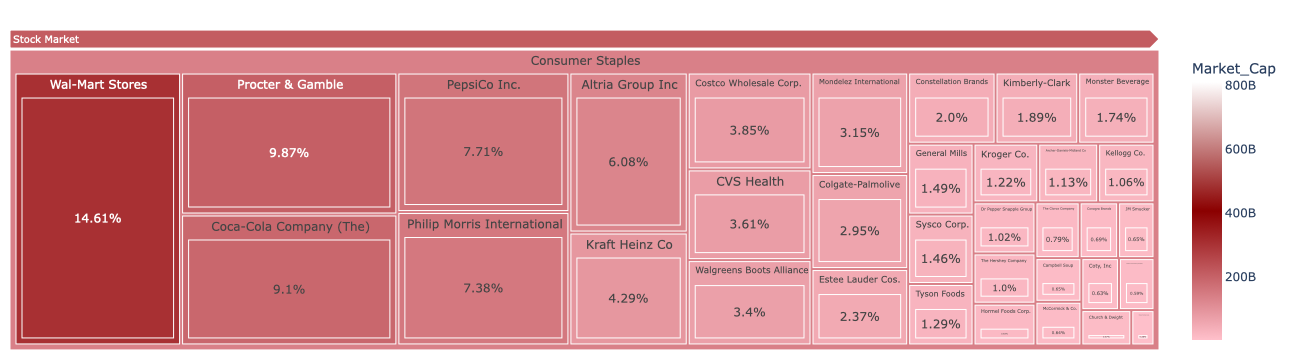
Source: Author’s work using Jupyter Lab IDE

**Figure 23. Treemap (Heat map) of the percentage of stocks in Industrial sector**



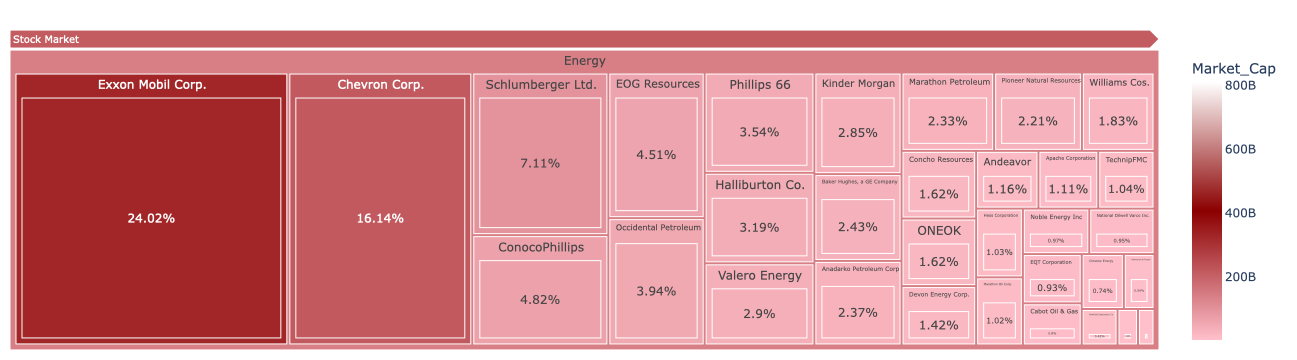
Source: Author’s work using Jupyter Lab IDE

**Figure 24. Treemap (Heat map) of the percentage of stocks in Consumer Staples sector**



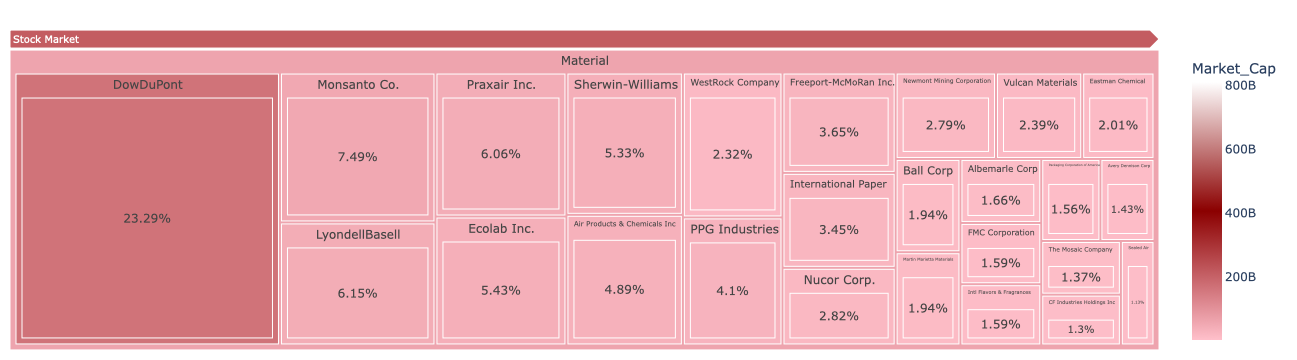
Source: Author’s work using Jupyter Lab IDE

**Figure 25. Treemap (Heat map) of the percentage of stocks in Energy sector**



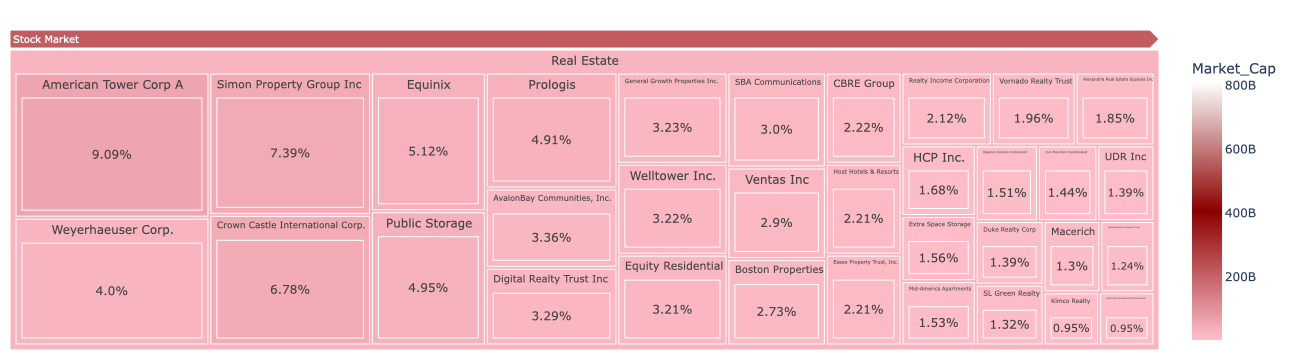
Source: Author’s work using Jupyter Lab IDE

**Figure 26. Treemap (Heat map) of the percentage of stocks in Material sector**



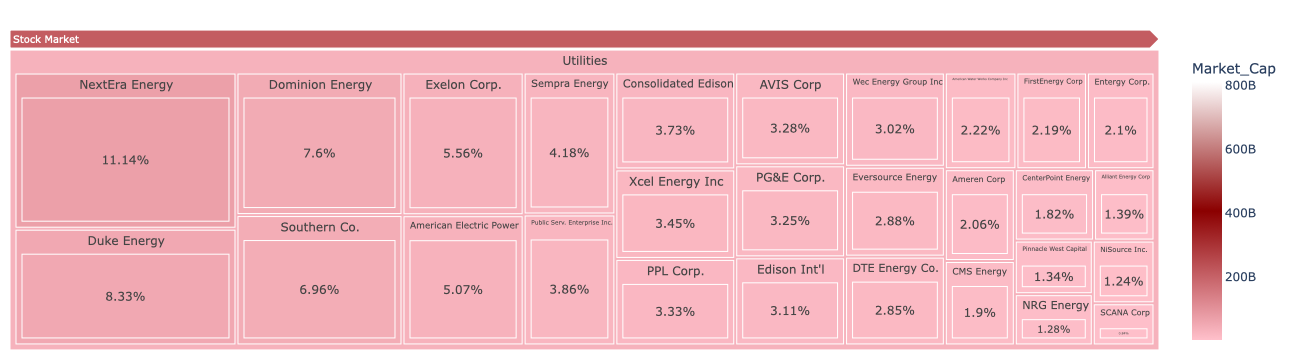
Source: Author’s work using Jupyter Lab IDE

**Figure 27. Treemap (Heat map) of the percentage of stocks in Real Estate sector**



Source: Author’s work using Jupyter Lab IDE

**Figure 28. Treemap (Heat map) of the percentage of stocks in Utilities sector**



Source: Author’s work using Jupyter Lab IDE

## Apendix B. Codes of the report

# =============================================================================

# Task 1

# =============================================================================

# 1.1. Import the dataset financials.csv in Python using pandas DataFrame

import pandas as pd

financial\_path = '/Users/chuchoanh/Documents/University of Liverpool 22:23/1. Programing (Python)/Assignment 2/'

financial\_df = pd.read\_csv(financial\_path + 'financials.csv')

# Display the first 8 rows of the financial dataframe

financial\_df.head(8) # This will be displayed using Jupyter Lab

# 1.2. Find Shape of the financial dataset

print(f'The shape of the financial dataset is {financial\_df.shape[0]} rows and {financial\_df.shape[1]} columns.')

# =============================================================================

# Task 2

# =============================================================================

# 2.1. Process missing values in the data set

# Loop over columns index 2 to 11 to convert these values into numerical data and detect missing values

for column in financial\_df.columns[2:]:

financial\_df[column] = pd.to\_numeric(financial\_df[column], errors='coerce')

# Print the number of missing values in each column

missing\_values = financial\_df.isna().sum()

print('The number of missing values in each column:\n', missing\_values)

# Calculate the proportion of missing values in each column

print('The proportion of missing values in each column:\n', (missing\_values/financial\_df.shape[0])\*100)

# Find standard deviation of each feature categorized by sectors:

sector\_df = financial\_df.groupby('Stock\_Market\_Sector') # Group data set by sectors

print ('Standard deviation of each feature categorized by sectors:')

sector\_df.std() # This will be displayed using Jupyter Lab

# Replace missing values with median of each feature catagorised by sector:

for column in financial\_df.columns[2:]: # Loop over columns that contain missing values

financial\_df[column] = financial\_df[column].fillna(sector\_df[column].median()) # Replace missing values with median of each feature catagorised by sector

print('Is there any missing value in the data frame?\n', financial\_df.isnull().any()) # Check if there is any missing value left in the data frame

# 2.2. Print basic statistical descriptions of numerical columns in the financial data set

print('Basic statistical descriptions of numerical columns in the financial data set:\n', financial\_df.describe())

# Print basic statistical descriptions of numerical columns in the financial data set

print('Basic statistical descriptions of categorical columns in the financial data set:\n', financial\_df.describe(include=["object", "bool"]))

# =============================================================================

# Task 3

# =============================================================================

# 3.1. Create the ranks of the percentage of stocks in each sector treemap using plotly.express library in Jupyter Lab

# Peter, M (2022) 'Create Your Own Wall Street HeatMap With Python' (Python 3.0) [Source code]. https://medium.com/@peterspage/create-your-own-wall-street-heatmap-with-python-27c0597df2db

df1 = round((financial\_df['Market\_Cap'] / financial\_df.groupby('Stock\_Market\_Sector')['Market\_Cap'].transform('sum'))\*100,2) # Finding percentage of stocks in each sector by dividing financial\_df['Market\_Cap'] by Market\_Cap sum of each sector

df2 = financial\_df[['Stock\_Market\_Sector','Company\_Name', 'Market\_Cap']] # Create new data frame including columns of 'Stock\_Market\_Sector','Company\_Name' and 'Market\_Cap'

df2 ['Market\_Cap\_Share'] = df1 # Append percentage of stocks in each sector to the data frame

df2 = df2.sort\_values(by=['Market\_Cap'], ascending=[False]) # Sort values of the data frame in descending order, this data frame is ready for ploting

import plotly.express as px # Import plotly.express library for ploting treemap

MSP = px.treemap(df2,

path=[px.Constant('Stock Market'), 'Stock\_Market\_Sector', 'Company\_Name', 'Market\_Cap\_Share'],

values='Market\_Cap', color='Market\_Cap',

color\_continuous\_scale=[ 'Pink', "#8b0000", 'White']) # Pass values into px.treemap for plotting, MSP is the name of the plot

MSP.data[0].texttemplate = "%{label}%" # Add % to the lable of percentage of stocks

MSP.update\_traces(textposition="middle center",

selector=dict(type='treemap')) # Format lables of stocks

MSP.update\_layout(margin = dict(t=50, l=10, r=10, b=10)) # Format margin of treemap

MSP.show() # Show the treemap

# 3.2. Import seaborn and matplotlib library to visualize the correlation matrix

import seaborn as sns

import matplotlib.pyplot as plt

# Find correlation matrix of columns index 2 to 11 using df.corr()

correlation\_mat = financial\_df.iloc[:,2:].corr()

# Visualize and render the plot with title

sns.heatmap(correlation\_mat, annot = True)

plt.title('Correlation heatmap of financial features')

plt.show()

# 3.3. Define a function to dectect and print names of companies which have maximum or minimum value of a column

def max\_min\_comp (column):

Company\_max = financial\_df.iloc[financial\_df[column].idxmax(), 0] # Find row index of the company with maximum value using df.idxmax() and access company name using df.iloc with column index of 0 (Column of Company\_Name)

Company\_min = financial\_df.iloc[financial\_df[column].idxmin(), 0] # Find row index of the company with minimum value using df.idxmin() and access company name using df.iloc with column index of 0 (Column of Company\_Name)

print(f'The company has the maximum {column} is {Company\_max} with the value of {financial\_df.loc[financial\_df[column].idxmax(), column]}.') # Print company name with maximum value

print(f'The company has the minimum {column} is {Company\_min} with the value of {financial\_df.loc[financial\_df[column].idxmin(), column]}.') # Print company name with minimum value

# Loop over the list of given columns (features) and pass columns to the defined function max\_min\_comp (column) to display results

for column in ('Dividend Yield', 'Earnings/Share', 'Price/Sales'):

max\_min\_comp (column)

# 3.4. Define a function to calculate the probability of dividend yield greater than 2.181 for a sector

def div\_prob (sector):

sector\_df = (financial\_df[(financial\_df['Stock\_Market\_Sector'] == sector)]) # Slicing (categorising) data frame into sectors using logical expression

i = 0

for item in sector\_df.iloc[:,4].values: # Turn the sector data frame in Dividend Yield column into array using df.values and loop over values in the array

if item > 2.181: # Count the number of values > 2.181 using conditional statement

i = i + 1

div\_prob = (i/len(sector\_df.iloc[:,4].values)) # Compute the probability by ratio of the number of satisfied values (i) and the total number of Dividend Yields in each sector

return div\_prob

# Loop over the list of given sectors and to each sector to function div\_prob (sector), print the result

for sector in ('Real Estate', 'Financials', 'Energy'):

print (f'The probability of dividend yield greater than 2.181 for {sector} is', round(div\_prob (sector),3))

# 3.5. Define a function to plot the data frame of a feature by sectors

def bar\_plot(column):

sector\_df1 = financial\_df.groupby('Stock\_Market\_Sector')[column].mean() # Group data frame by sector and feature

sector\_df1.plot.bar() # Bar plot the data frame

plt.title(f'Average {column} for each market sector') # Insert title for the bar plot

plt.ylabel(f'Average {column}') # Insert label for x-axis of the bar plot

for i in range(0, sector\_df1.shape[0]):

if sector\_df1[i] > 1e3:

plt.annotate(str(round(sector\_df1[i]/1e11,3)), xy = (i,sector\_df1[i]), ha = 'center', va = 'bottom') # Turn values greater than 1e3 into ratio of 1e11 for better annotations

else:

plt.annotate(str(round(sector\_df1[i],3)), xy = (i,sector\_df1[i]), ha = 'center', va = 'bottom') # Insert annotations of values less than 1e3

plt.show()

# Loop over list of given features and pass each feature to the bar\_plot(column) function to plot the requested data:

for sector in ('Earnings/Share', 'Dividend Yield', 'Market\_Cap'):

bar\_plot(sector)

# **References**

Becker, J., Breuker, D., Heide, T., Holler, J., Rauer, H.P., Böhme, R. (2013). Can We Afford Integrity by Proof-of-Work? Scenarios Inspired by the Bitcoin Currency. In: Böhme, R. (eds) The Economics of Information Security and Privacy. *Springer, Berlin, Heidelberg*. <https://doi.org/10.1007/978-3-642-39498-0_7>.

Coutinho, K., Wongthongtham, P., Abu-Salih, B., Abu Saleh, M.A. and Khairwal, N.K., 2022. Carbon emission and cost of blockchain mining in a case of peer-to-peer energy trading. *Frontiers in Built Environment*.

Claire, B (2021), ‘How Dividends Affect Stock Prices’, available at: <https://www.investopedia.com/articles/investing/091015/how-dividends-affect-stock-prices.asp> (accessed on 05/01/2023)

CFI (2022), ‘EBITDA Multiple’, available at: <https://corporatefinanceinstitute.com/resources/valuation/ebitda-multiple/> (accessed on 05/01/2023)

Climate Watch Historical GHG Emissions. 2022. Washington, DC: World Resources Institute. 2019 est.

Dorfleitner, G., Muck, F. and Scheckenbach, I. (2021) ‘Blockchain applications for climate protection: A global empirical investigation’, *Renewable and Sustainable Energy Reviews*, 149, p. 111378. Available at: https://doi.org/10.1016/j.rser.2021.111378.

European Parliament. July 2018. Cryptocurrencies and blockchain. Retrieved 29 October 2020.

Jiang, S., Li, Y., Lu, Q., Hong, Y., Guan, D., Xiong, Y. and Wang, S., 2021. Policy assessments for the carbon emission flows and sustainability of Bitcoin blockchain operation in China. *Nature communications*, *12*(1), pp.1-10.

Jason, F (2021), ‘The Correlation Coefficient: What It Is, What It Tells Investors’, Investopia, available at: <https://www.investopedia.com/terms/c/correlationcoefficient.asp> (accessed on 05/01/2023)

Gil-Pulger, J. 2019. ‘Ethereum 2.0’ PoS blockchain aims to cut energy use by 99%. *Bitcoinist*; https://bitcoinist.com/ ethereum-pos-blockchain-cut-energy/

Howson, P. (2019) ‘Tackling climate change with blockchain’, *Nature Climate Change*, 9(9), pp. 644–645. Available at: https://doi.org/10.1038/s41558-019-0567-9.

Kim, S.-K. and Huh, J.-H. (2020) ‘Blockchain of Carbon Trading for UN Sustainable Development Goals’, *Sustainability*, 12(10), p. 4021. Available at: https://doi.org/10.3390/su12104021.

Patel, D., Britto, B., Sharma, S., Gaikwad, K., Dusing, Y. and Gupta, M., 2020, February. Carbon credits on blockchain. In *2020 International Conference on Innovative Trends in Information Technology (ICITIIT)* (pp. 1-5). IEEE.

Stoll, C., Klaaßen, L. and Gallersdörfer, U. (2019) ‘The Carbon Footprint of Bitcoin’, *Joule*, 3(7), pp. 1647–1661. Available at: https://doi.org/10.1016/j.joule.2019.05.012.

Sedlmeir, J., Buhl, H.U., Fridgen, G. *et al.* Ein Blick auf aktuelle Entwicklungen bei Blockchains und deren Auswirkungen auf den Energieverbrauch. *Informatik Spektrum*43, 391–404 (2020). <https://doi.org/10.1007/s00287-020-01321-z>.

United Nations Climate Change. How blockchain technology could boost climate action. 2017, URL: https://unfccc.int/news/un-supports-blockchain-technologyfor-climate-action, [Accessed 12 June 2020].

**Wordcount: 2,456 words**