FE5214 Assignment 1

Gaurav Agrawal - A0294856J

Question 1:

Obtain the number of stocks in each of the GICS sectors and industries for the S&P 500 universe at the beginning of 2014 and 2024 respectively. Calculate the total market cap of each sector and industry. **What are the top three sectors and the top three industries in terms of total market cap at the beginning of 2014 and at the beginning of 2024**?

Answer:

Code	Sector_Name	No. of Stocks 2014	No. of Stocks 2024	Market Cap of Stocks 2014	Market Cap of Stocks 2024
10	Energy	45	23	1,714,223.140000	1,617,214.102100
15	Materials	31	28	592,968.530000	983,547.433200
20	Industrials	64	78	1,851,276.045500	3,648,384.261600
25	Consumer Discretionary	62	53	1,505,581.206100	4,720,285.124300
30	Consumer Staples	41	38	1,787,059.533100	2,856,263.468800
35	Healthcare	55	64	2,259,201.379500	5,312,223.662000
40	Financials	67	72	2,783,578.085500	5,644,229.939100
45	Information Technology	50	64	2,318,796.187700	11,628,815.501800
50	Communication Services	26	22	1,595,571.071900	5,669,719.599600
55	Utilities	30	30	465,493.444200	949,389.018800
60	Real Estate	20	31	346,674.906500	1,035,240.743200
na	Unclassified	9	0	107,714.149400	0.000000

Table 1.1

The **GICS sector data frame** is as shown in Table 1.1. The table compares the number of stocks and market capitalization across different sectors in 2014 and 2024, showing growth or decline trends.

Code	Industry_Name	No. of Stocks 2014	No. of Stocks 2024	Market Cap of Stocks 2014	Market Cap of Stocks 2024
101010	Energy Equipment & Services	13	3	309,275.319500	140,460.189800
101020	Oil, Gas & Consumable Fuels	32	20	1,404,947.820500	1,476,753.912300
151010	Chemicals	16	16	437,059.870300	667,427.461900
151020	Construction Materials	1	2	7,627.782200	60,059.659600
151030	Containers & Packaging	7	6	57,084.064500	86,186.609900
...	...	...	...	...	...
601060	Residential REITs	3	7	37,820.255900	126,824.603400
601070	Retail REITs	4	5	81,762.950400	124,768.585200
601080	Specialized REITs	5	10	104,774.755800	466,794.907600
602010	Real Estate Management & Development	1	2	8,728.909400	63,311.518300
na	Unclassified	9	0	107,714.149400	0.000000

70 rows × 5 columns

Table 1.2 (Complete Table can be found with code in Appendix of this report)

The **GICS Industries data frame** is as shown in Table 1.2. The table compares the number of stocks and market capitalization across different industries in 2014 and 2024, showing growth or decline trends.

Top 3 Sectors by Market Cap at the beginning of 2014

Code	Sector_Name	Market Cap of Stocks 2014
40	Financials	2,783,578.085500
45	Information Technology	2,318,796.187700
35	Healthcare	2,259,201.379500

Top 3 Sectors by Market Cap at the beginning of 2024

Code	Sector_Name	Market Cap of Stocks 2024
45	Information Technology	11,628,815.501800
50	Communication Services	5,669,719.599600
40	Financials	5,644,229.939100

Top 3 Industries by Market Cap at the beginning of 2014

Code	Industry_Name	Market Cap of Stocks 2014
101020	Oil, Gas & Consumable Fuels	1,404,947.820500
401010	Banks	1,031,024.225700
352020	Pharmaceuticals	863,815.602900

Top 3 Industries by Market Cap at the beginning of 2024

Code	Industry_Name	Market Cap of Stocks 2024
502030	Interactive Media & Services	4,375,614.834600
451030	Software	4,364,983.241300
453010	Semiconductors & Semiconductor Equipment	3,171,491.261400

Steps:

1. Data Loading and Preprocessing
   - The universe DataFrame, containing stock tickers for the S&P 500 universe for different years, is read from a CSV file (univ_h.csv). The year column is set as the index to allow easy filtering based on specific years (2014 and 2024).
   - The tickers DataFrame is read from a CSV file (tickers.csv) and contains the mapping of stock tickers to their corresponding GICS codes. The tickers are set as the index.
   - The market_cap DataFrame, containing the market capitalization data for each ticker over time, is read and converted into a proper datetime format. Missing values are replaced with zeros.

2. Computing the result:
   - Two DataFrames (gics_sectors and gics_industries) are created to hold information about the sectors and industries.

- For both the years 2014 and 2024, the code iterates over the tickers in the S&P 500 universe for that year. The universe matrix (universe.loc[y, ticker]) is checked to ensure that the ticker is part of the S&P 500 universe for the specified year

  For each ticker that is part of the universe:

- The corresponding GICS sector and industry codes are retrieved from the tickers DataFrame.
- The sector code is extracted from the first two digits of the GICS code, and the industry code is extracted from the first six digits.
- The number of stocks in each sector and industry for the year is updated in the respective DataFrames.
- The market cap of the stock for the given date (January 2nd of the respective year) is retrieved from the market_cap DataFrame, and the total market cap for the respective sector and industry is updated accordingly.

Note: Detailed Python Code for above steps can be found in Appendix at end of this report

Data Visualization and Analysis:

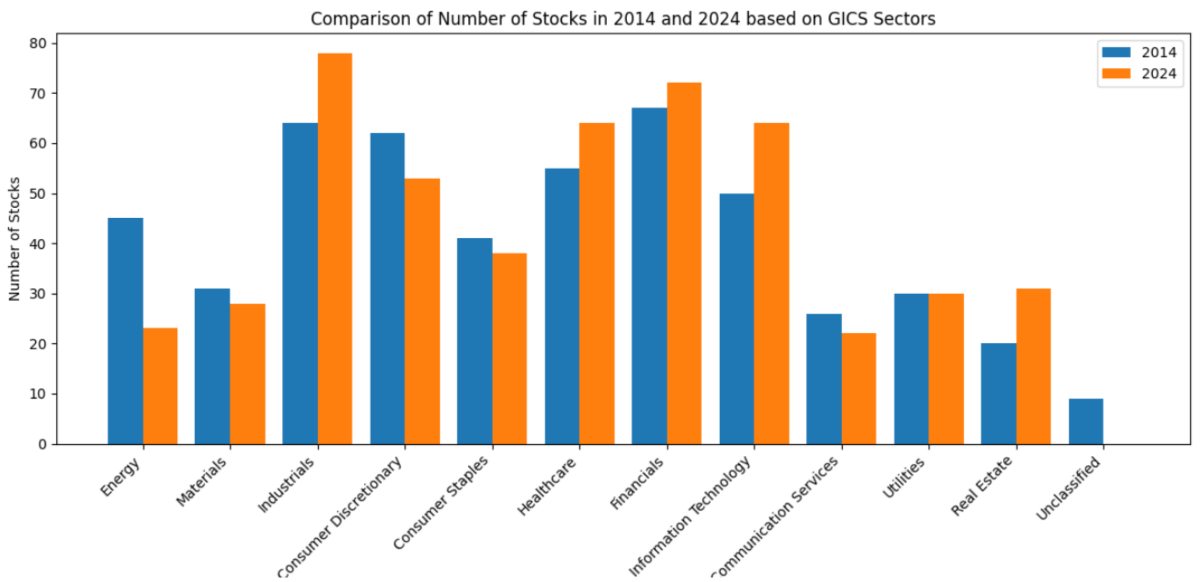


Figure 1.1 Comparison of No of Stocks in 2014 & 2024 based on GICS Sectors

The chart in Figure 1.1 shows sectoral shifts from 2014 to 2024. Energy and Communication Services declined, while Industrials and Information Technology grew. Consumer Discretionary and Healthcare saw slight reductions and Real Estate expanded.

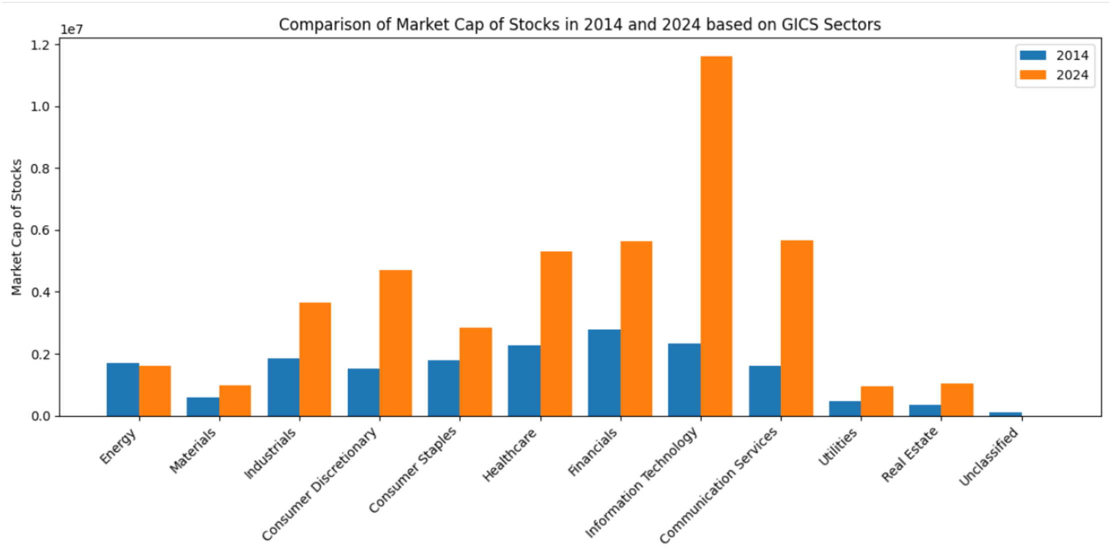


Figure 1.2 Comparison of Market Cap in 2014 & 2024 based on GICS Sectors

The chart in Figure 1.2 shows market cap growth across most sectors from 2014 to 2024. Information Technology and Communication Services saw massive increases, while Consumer Discretionary, Financials, and Healthcare also grew significantly. Energy slightly declined, despite fewer stocks and The Real Estate sector expanded.

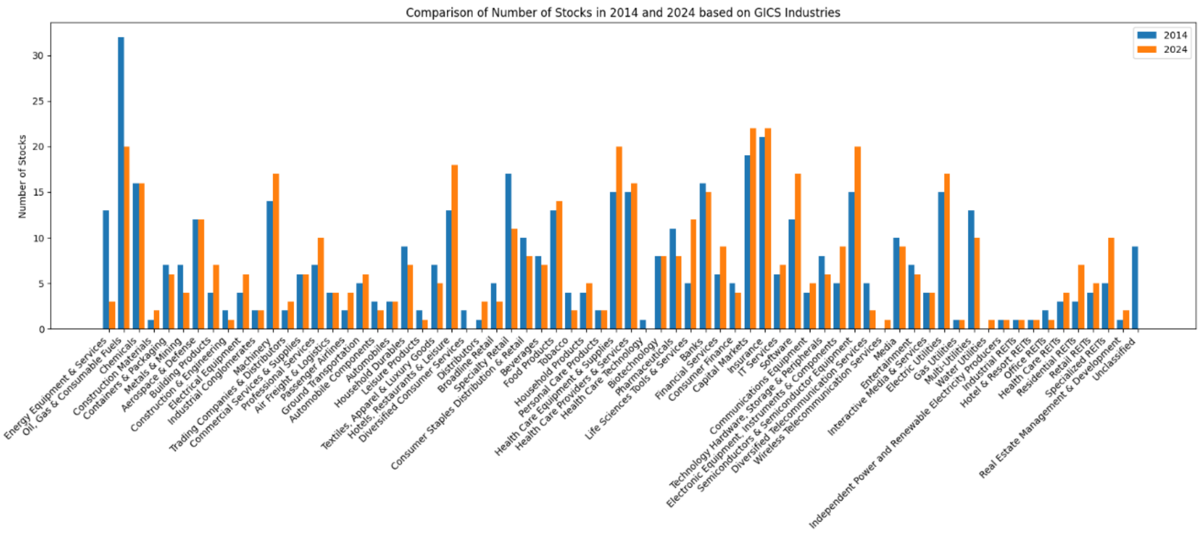


Figure 1.3 Comparison of No of Stocks in 2014&2024 based on GICS Industries

The chart in Figure 1.3 highlights industry shifts between 2014 and 2024. Technology Hardware, Health Care Equipment, and Financial Services saw notable growth in stock counts, indicating expansion and investor interest. Energy Equipment & Services and Oil, Gas & Consumable Fuels declined, possibly due to market exits or industry consolidation. Software, Biotechnology, and Capital Markets also show increases, reflecting innovation and economic shifts. Meanwhile, Retail, Consumer Services, and Real Estate Management exhibit mixed trends. The overall distribution appears more balanced in 2024, suggesting diversification and structural adjustments in various industries over the decade.

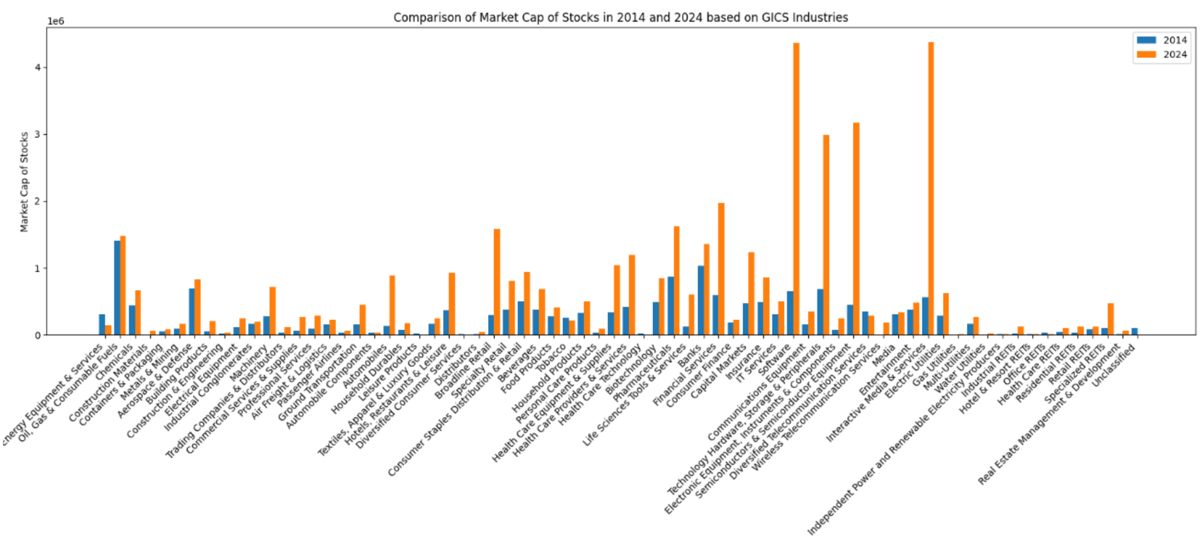


Figure 1.4 Comparison of Market Cap in 2014&2024 based on GICS Industries

The chart in Figure 1.4 shows a significant increase in market capitalization across most industries from 2014 to 2024. Technology Hardware, Interactive Media & Services, Software, and Financial Services saw substantial growth, reflecting strong investor confidence. Health Care Equipment and Capital Markets also expanded, indicating sectoral strength. In contrast, Energy Equipment & Services and Oil, Gas & Consumable Fuels showed slower growth, suggesting a decline in market influence. Overall, the data indicates economic expansion, shifting industry dominance, and evolving market trends over the decade.

Question 2:

List the stocks in the 2024 universe that are also in the industry NVDA belongs to.

Answer:

NVDA industry is: Code- 453010

Industry_Name- Semiconductors & Semiconductor Equipment

Stocks in the 2024 universe that are also in the industry NVDA belongs to:

['ADI', 'AMAT', 'AMD', 'AVGO', 'ENPH', 'FSLR', 'INTC', 'KLAC', 'LRCX', 'MCHP', 'MPWR', 'MU', 'NVDA', 'NXPI', 'ON', 'QCOM', 'QRVO', 'SWKS', 'TER', 'TXN']

Steps:

Determine NVDA's Industry

- The **GICS industry code** for NVDA is extracted from the tickers dataset using: nvda_industry_code = tickers.loc['NVDA']['GICS_Code'][:6]
- The first **six digits** of the GICS code represent the **industry-level classification**.

- The code iterates over all **tickers in the universe** (universe.columns). It checks if the stock exists in 2024 (universe.loc[2024, ticker] == 1). If the stock has a valid **GICS code**, the first six digits are extracted.
- If the extracted **industry code** matches nvda_industry_code, the stock is added to nvda_industry_tickers.

Note: Detailed Python Code for above steps can be found in Appendix at end of this report

Question 3:

Calculate the returns of the stocks in the list from Question 2 (the stocks that are in the 2024 universe and in the industry that NVDA belongs to) for the past five years (return time series, from 2020 to 2024). Use the adjusted prices to calculate the returns (you might use log-return, Calculate the industry return, by simply average over all the stock returns in the industry, where n is the number of stocks in that industry. Do the time-series regression Obtain the values of α and β. What's the R^2 of the regression?

Answer:

- Value of Alpha: 0.001820724872983911
- Value of Beta: 1.108654187377237
- R-squared of regression: 0.6251782477752315

Steps:

- Extracts **adjusted closing prices** for **stocks in NVDA's industry** from adjusted.
- Computes **log returns** for each stock using: log-return, $r_i(t) = \ln\left(\frac{P_i(t)}{P_i(t-1)}\right)$
- Selects data from **January 1, 2020, to December 31, 2024** for analysis.
- Averages the log returns of all stocks in NVDA's industry to obtain the **industry return**.

Run Linear Regression (Time-Series Analysis)

- Defines independent variable (**X**) as **industry return** with a constant term.
- Defines dependent variable (**y**) as **NVDA's log return**.
- Fits an **Ordinary Least Squares (OLS) regression**: $r = \alpha + \beta r_I + \epsilon$,where r is the return of NVDA.

Note: Detailed Python Code for above steps can be found in Appendix at end of this report

Data Visualization and Analysis:

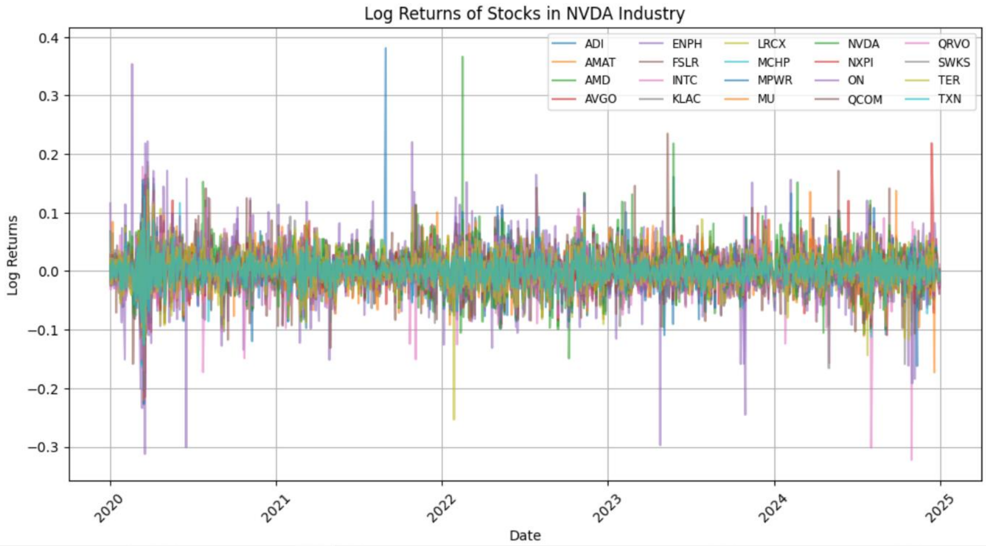


Figure 3.1 Log Returns of Stocks in NVDA Industry

The chart in Figure 3.1 illustrates the log returns of various stocks in NVIDIA's industry, showing significant volatility, especially around 2020, likely due to market turbulence from COVID-19. While fluctuations persist, volatility appears to stabilize post-2021, with occasional spikes indicating major market events. The stocks show strong correlation, suggesting industry-wide trends. Overall, returns cluster around zero, reflecting mean reversion, while periodic surges hint at earnings impacts or macroeconomic influences.

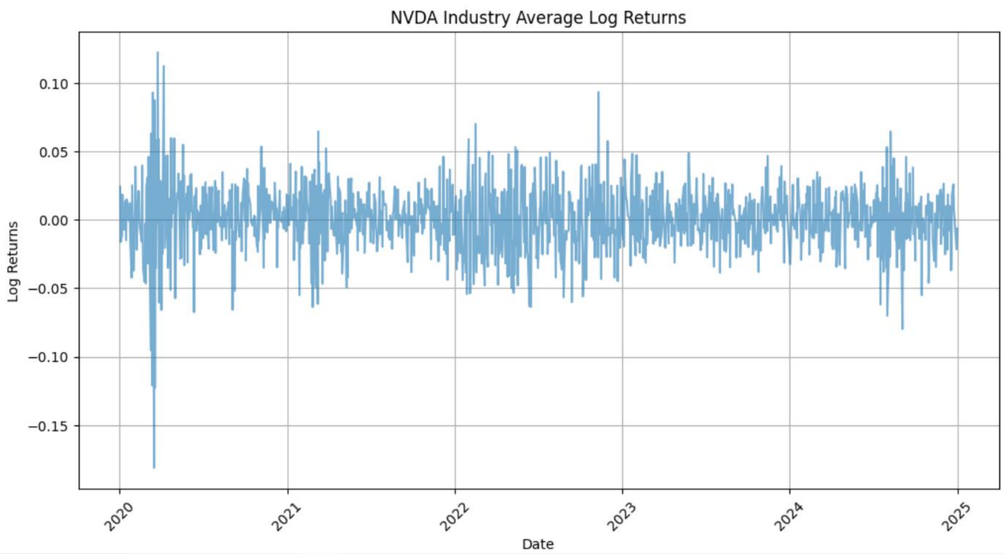


Figure 3.2 NVDA Industry Average Log Returns

This chart in Figure 3.2 shows the average log returns of the NVIDIA industry over time. There is high volatility around 2020, likely due to the COVID-19 market crash, followed by a gradual stabilization. While fluctuations persist, they are less extreme post-2021. Occasional spikes indicate significant industry events, such as earnings reports or macroeconomic shifts.

Question 4:

If the industry return, in Question 3, is calculated using the market-cap weighted average (instead of the simple average), **what will be your answers**?

Answer:

- Value of Alpha: 0.0007526980760657113
- Value of Beta: 1.2345535145383124
- R-squared of regression: 0.8447675448443841

Steps:

- Filter market cap data for the specific date range. Normalize the market cap for each stock by dividing it by the total market cap for all stocks in the industry on each day. This gives the weight of each stock.
- Multiply the log returns of each stock by its corresponding weight to get the weighted log returns.
- The industry return is then computed as the sum of the weighted returns.
- Regress NVDA's log returns against the weighted industry returns.

Note: Detailed Python Code for above steps can be found in Appendix at end of this report

Data Visualization and Analysis:

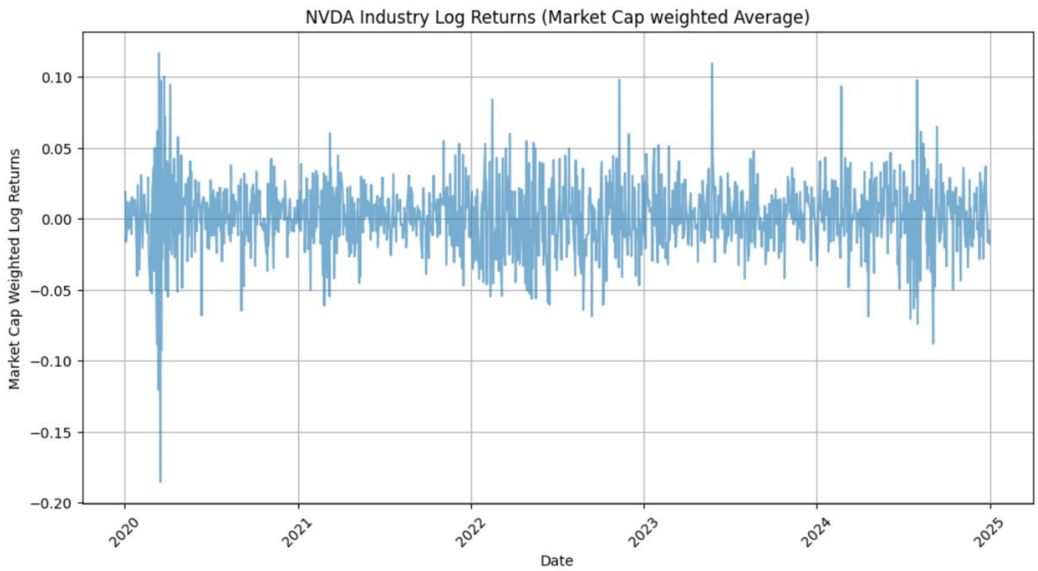


Figure 4.1 NVDA Industry Log Returns (Market Cap weighted Average)

This chart represents the market-cap-weighted average log returns of NVIDIA's industry over time. Similar to previous trends, high volatility is evident around 2020, likely due to the COVID-19 crash, followed by a stabilization phase. The weighted average approach emphasizes the impact of larger companies, making fluctuations more reflective of dominant firms

Appendix

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm

pd.set_option('display.max_rows', None) # Show all rows
pd.set_option('display.max_columns', None) # Show all columns
pd.options.display.float_format = '{:,.6f}'.format
```

In [2]:
```
gics_sectors_dict = {
    "10": "Energy",
    "15": "Materials",
    "20": "Industrials",
    "25": "Consumer Discretionary",
    "30": "Consumer Staples",
    "35": "Healthcare",
    "40": "Financials",
    "45": "Information Technology",
    "50": "Communication Services",
    "55": "Utilities",
    "60": "Real Estate",
    "na" : "Unclassified"
}
```

In [3]:
```
gics_industries_dict = {
    "101010": "Energy Equipment & Services",
    "101020": "Oil, Gas & Consumable Fuels",
    "151010": "Chemicals",
    "151020": "Construction Materials",
    "151030": "Containers & Packaging",
    "151040": "Metals & Mining",
    "151050": "Paper & Forest Products",
    "201010": "Aerospace & Defense",
    "201020": "Building Products",
    "201030": "Construction & Engineering",
    "201040": "Electrical Equipment",
    "201050": "Industrial Conglomerates",
    "201060": "Machinery",
    "201070": "Trading Companies & Distributors",
    "202010": "Commercial Services & Supplies",
    "202020": "Professional Services",
    "203010": "Air Freight & Logistics",
    "203020": "Passenger Airlines",
    "203030": "Marine Transportation",
    "203040": "Ground Transportation",
    "203050": "Transportation Infrastructure",
    "251010": "Automobile Components",
    "251020": "Automobiles",
    "252010": "Household Durables",
    "252020": "Leisure Products",
    "252030": "Textiles, Apparel & Luxury Goods",
    "253010": "Hotels, Restaurants & Leisure",
    "253020": "Diversified Consumer Services",
    "255010": "Distributors",
    "255030": "Broadline Retail",
    "255040": "Specialty Retail",
    "301010": "Consumer Staples Distribution & Retail",
    "302010": "Beverages",
    "302020": "Food Products",
    "302030": "Tobacco",
    "303010": "Household Products",
    "303020": "Personal Care Products",
    "351010": "Health Care Equipment & Supplies",
    "351020": "Health Care Providers & Services",
    "351030": "Health Care Technology",
    "352010": "Biotechnology",
    "352020": "Pharmaceuticals",
    "352030": "Life Sciences Tools & Services",
    "401010": "Banks",
```

```
    "402010": "Financial Services",
    "402020": "Consumer Finance",
    "402030": "Capital Markets",
    "402040": "Mortgage Real Estate Investment Trusts (REITs)",
    "403010": "Insurance",
    "451020": "IT Services",
    "451030": "Software",
    "452010": "Communications Equipment",
    "452020": "Technology Hardware, Storage & Peripherals",
    "452030": "Electronic Equipment, Instruments & Components",
    "453010": "Semiconductors & Semiconductor Equipment",
    "501010": "Diversified Telecommunication Services",
    "501020": "Wireless Telecommunication Services",
    "502010": "Media",
    "502020": "Entertainment",
    "502030": "Interactive Media & Services",
    "551010": "Electric Utilities",
    "551020": "Gas Utilities",
    "551030": "Multi-Utilities",
    "551040": "Water Utilities",
    "551050": "Independent Power and Renewable Electricity Producers",
    "601010": "Diversified REITs",
    "601025": "Industrial REITs",
    "601030": "Hotel & Resort REITs",
    "601040": "Office REITs",
    "601050": "Health Care REITs",
    "601060": "Residential REITs",
    "601070": "Retail REITs",
    "601080": "Specialized REITs",
    "602010": "Real Estate Management & Development",
    "na" : "Unclassified"
}
```

In [4]:

```
# Read the required CSV files and perform necessary preprocessing

universe = pd.read_csv('Data/data_us/univ_h.csv')
universe.set_index('year', inplace=True)

columns = ['Ticker', 'GICS_Code']
tickers = pd.read_csv('Data/data_us/tickers.csv', header=None, names = columns, dtype=str)
tickers.set_index('Ticker', inplace=True)

market_cap = pd.read_csv('Data/data_us/mktcap.csv')
market_cap['Date'] = pd.to_datetime(market_cap['Date'], format = '%Y%m%d')
market_cap.set_index('Date', inplace=True)
market_cap.replace(np.nan, 0, inplace=True)

adjusted = pd.read_csv('Data/data_us/mktcap.csv')
adjusted['Date'] = pd.to_datetime(adjusted['Date'], format = '%Y%m%d')
adjusted.set_index('Date', inplace=True)
```

In [5]:

```
universe.head()
```

Out[5]:

	0111145D	0202445Q	0203524D	0226226D	0544749D	0574018D	0772031D	0848680D	0867887D	0910150D	09
year											
2004	1	1	1	1	1	0	1	1	0	1	
2005	1	1	1	1	1	1	1	1	0	1	
2006	1	1	1	1	1	1	1	1	0	1	
2007	1	1	1	1	0	1	1	1	0	1	
2008	1	1	1	1	0	0	1	1	0	0	

In [6]:

```
tickers.head(5)
```

Out[6]:

	GICS_Code
Ticker	
0111145D	NaN
0202445Q	NaN
0203524D	NaN
0226226D	NaN
0544749D	NaN

In [7]:

```
market_cap.head()
```

Out[7]:

	0111145D	0202445Q	0203524D	0226226D	0544749D	0574018D	0772031D	0848680D	08678
Date									
2004-01-02	1,483.510000	2,098.220000	7,163.206600	6,489.253500	4,761.048100	5,316.218900	7,444.547300	14,761.537500	0.000
2004-01-05	1,491.440000	2,101.640000	7,426.450800	6,376.748200	4,851.811600	5,320.043500	7,992.720600	15,367.185400	0.000
2004-01-06	1,493.640000	2,074.310000	7,257.114100	6,545.195300	4,965.770200	5,316.218900	7,957.354600	15,981.872800	0.000
2004-01-07	1,478.670000	2,117.740000	7,148.506700	6,514.116500	4,958.710800	5,270.323500	7,851.256600	15,773.963800	0.000
2004-01-08	1,468.540000	2,127.510000	7,477.889900	6,177.222300	4,949.634400	5,266.498900	7,992.720600	15,999.951900	0.000

Question 1

In [8]:

```
# Create DataFrames for GICS sectors and industries from dictionaries and initialize relevant columns

gics_sectors = pd.DataFrame(list(gics_sectors_dict.items()), columns=["Code", "Sector_Name"])
gics_sectors.set_index('Code', inplace=True)
gics_sectors['No. of Stocks 2014'] = 0
gics_sectors['No. of Stocks 2024'] = 0

gics_industries = pd.DataFrame(list(gics_industries_dict.items()), columns=["Code", "Industry_Name"])
gics_industries.set_index('Code', inplace=True)
gics_industries['No. of Stocks 2014'] = 0
gics_industries['No. of Stocks 2024'] = 0

gics_sectors['Market Cap of Stocks 2014'] = 0.0
gics_sectors['Market Cap of Stocks 2024'] = 0.0

gics_industries['Market Cap of Stocks 2014'] = 0.0
gics_industries['Market Cap of Stocks 2024'] = 0.0
```

In [9]:

```
# Iterate over the years and tickers to update stock count and market cap for GICS sectors and industries

years = [2014, 2024]
for y in years:
    for ticker in universe.columns:
        if universe.loc[y, ticker] == 1:   # Check if the ticker is part of the universe in the given year
            curr_code = str(tickers.loc[ticker, 'GICS_Code'])
            if curr_code == 'nan':         # Handle missing GICS codes
                sector_code = 'na'
                industry_code = 'na'

            else:
                sector_code = curr_code[:2]    # Extract sector code (first 2 digits)
                industry_code = curr_code[:6]  # Extract industry code (first 6 digits)

            # Update stock count for the corresponding sector and industry
            gics_sectors.loc[sector_code, f'No. of Stocks {y}'] += 1
            gics_industries.loc[industry_code, f'No. of Stocks {y}'] += 1
```

```
        # Retrieve market cap for the given year and ticker
        date = pd.to_datetime(f"{y}0102", format='%Y%m%d')
        curr_market_cap = market_cap.loc[date, ticker]

        # Update market cap for the corresponding sector and industry
        gics_sectors.loc[sector_code, f'Market Cap of Stocks {y}'] += curr_market_cap
        gics_industries.loc[industry_code, f'Market Cap of Stocks {y}'] += curr_market_cap
```

In [10]:

```
gics_sectors
```

Out[10]:

Code	Sector_Name	No. of Stocks 2014	No. of Stocks 2024	Market Cap of Stocks 2014	Market Cap of Stocks 2024
10	Energy	45	23	1,714,223.140000	1,617,214.102100
15	Materials	31	28	592,968.530000	983,547.433200
20	Industrials	64	78	1,851,276.045500	3,648,384.261600
25	Consumer Discretionary	62	53	1,505,581.206100	4,720,285.124300
30	Consumer Staples	41	38	1,787,059.533100	2,856,263.468800
35	Healthcare	55	64	2,259,201.379500	5,312,223.662000
40	Financials	67	72	2,783,578.085500	5,644,229.939100
45	Information Technology	50	64	2,318,796.187700	11,628,815.501800
50	Communication Services	26	22	1,595,571.071900	5,669,719.599600
55	Utilities	30	30	465,493.444200	949,389.018800
60	Real Estate	20	31	346,674.906500	1,035,240.743200
na	Unclassified	9	0	107,714.149400	0.000000

In [11]:

```
# Remove industries with zero stocks in both 2014 and 2024
gics_industries.drop(gics_industries[(gics_industries['No. of Stocks 2014'] == 0) &
                    (gics_industries['No. of Stocks 2024'] == 0)].index, inplace = True)
```

In [12]:

```
# Compute total number of stocks across all sectors for 2014 and 2024

total_2014 = gics_sectors['No. of Stocks 2014'].sum()
print('Total number of stocks in 2014:',total_2014)

total_2024 = gics_sectors['No. of Stocks 2024'].sum()
print('Total number of stocks in 2024:',total_2024)
```

```
Total number of stocks in 2014: 500
Total number of stocks in 2024: 503
```

In [13]:

```
gics_industries
```

Out[13]:

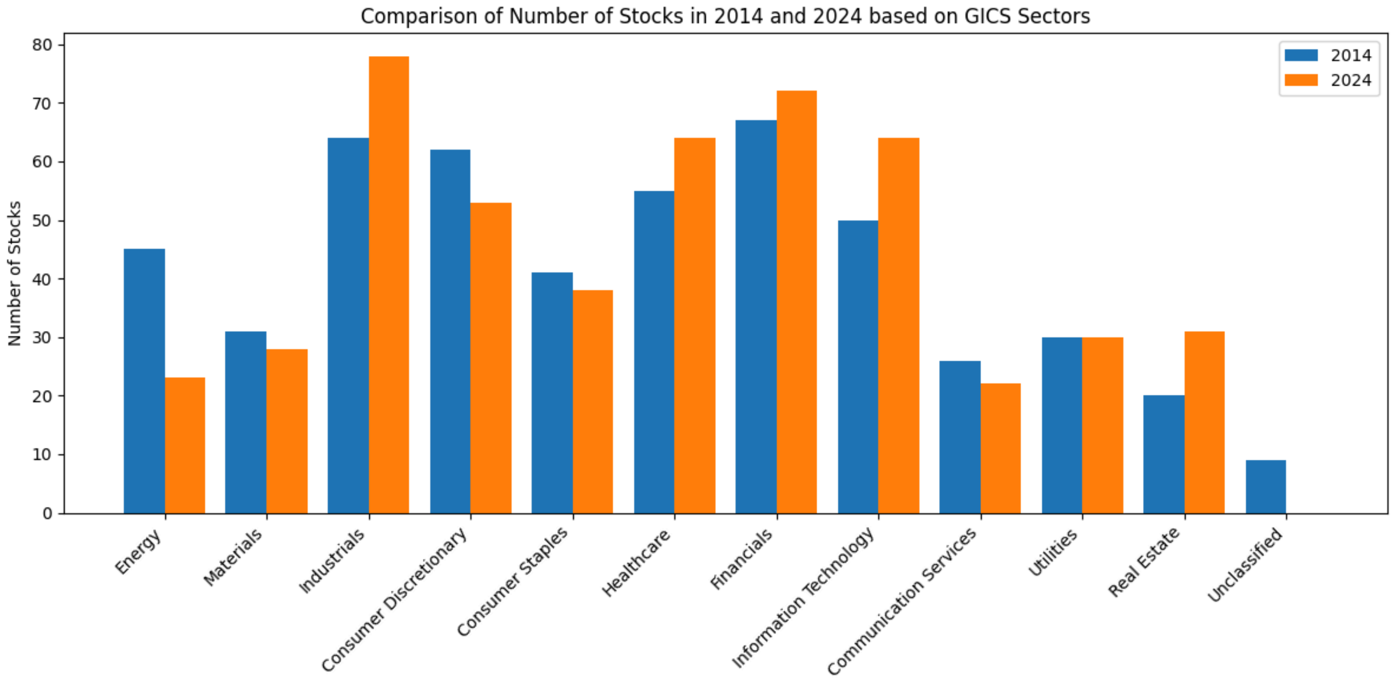
Code	Industry_Name	No. of Stocks 2014	No. of Stocks 2024	Market Cap of Stocks 2014	Market Cap of Stocks 2024
101010	Energy Equipment & Services	13	3	309,275.319500	140,460.189800
101020	Oil, Gas & Consumable Fuels	32	20	1,404,947.820500	1,476,753.912300
151010	Chemicals	16	16	437,059.870300	667,427.461900
151020	Construction Materials	1	2	7,627.782200	60,059.659600
151030	Containers & Packaging	7	6	57,084.064500	86,186.609900
151040	Metals & Mining	7	4	91,196.813000	169,873.701800
201010	Aerospace & Defense	12	12	693,070.520300	828,269.355900
201020	Building Products	4	7	48,860.804300	202,646.067300
201030	Construction & Engineering	2	1	19,637.884000	30,403.768900
201040	Electrical Equipment	4	6	113,542.573500	248,050.727600
201050	Industrial Conglomerates	2	2	162,761.502100	198,538.258800
201060	Machinery	14	17	275,880.427700	720,339.941500
201070	Trading Companies & Distributors	2	3	31,423.182500	115,116.643000
202010	Commercial Services & Supplies	6	6	62,260.388000	270,748.091600
202020	Professional Services	7	10	98,866.547700	288,759.552800
203010	Air Freight & Logistics	4	4	157,265.494200	226,566.680300
203020	Passenger Airlines	2	4	36,852.704800	65,145.895700
203040	Ground Transportation	5	6	150,854.016400	453,799.278200
251010	Automobile Components	3	2	36,596.026700	33,953.572100
251020	Automobiles	3	3	132,939.731800	887,753.375500
252010	Household Durables	9	7	73,072.873300	173,078.986000
252020	Leisure Products	2	1	23,120.414200	6,924.330900
252030	Textiles, Apparel & Luxury Goods	7	5	163,070.562500	251,372.893600
253010	Hotels, Restaurants & Leisure	13	18	373,486.835600	930,304.326000
253020	Diversified Consumer Services	2	0	12,892.972500	0.000000
255010	Distributors	1	3	12,644.941300	47,025.120700
255030	Broadline Retail	5	3	295,850.423300	1,581,858.762300
255040	Specialty Retail	17	11	381,906.424900	808,013.757200
301010	Consumer Staples Distribution & Retail	10	8	502,468.087500	939,347.478200
302010	Beverages	8	7	376,639.160100	689,623.808400
302020	Food Products	13	14	282,986.153900	412,421.947700
302030	Tobacco	4	2	258,828.162300	221,888.245200
303010	Household Products	4	5	330,141.829100	499,626.758300
303020	Personal Care Products	2	2	35,996.140200	93,355.231000
351010	Health Care Equipment & Supplies	15	20	338,649.767200	1,041,533.110900
351020	Health Care Providers & Services	15	16	420,085.122200	1,196,167.832800
351030	Health Care Technology	1	0	18,903.488400	0.000000
352010	Biotechnology	8	8	493,130.834200	845,880.001700
352020	Pharmaceuticals	11	8	863,815.602900	1,621,496.686500

Code	Industry_Name	No. of Stocks 2014	No. of Stocks 2024	Market Cap of Stocks 2014	Market Cap of Stocks 2024
352030	Life Sciences Tools & Services	5	12	124,616.564600	607,146.030100
401010	Banks	16	15	1,031,024.225700	1,353,965.734300
402010	Financial Services	6	9	588,754.246800	1,965,705.058500
402020	Consumer Finance	5	4	190,695.911800	231,343.270700
402030	Capital Markets	19	22	476,250.986400	1,236,383.665900
403010	Insurance	21	22	496,852.714800	856,832.209700
451020	IT Services	6	7	311,101.401300	505,337.808400
451030	Software	12	17	650,329.145700	4,364,983.241300
452010	Communications Equipment	4	5	153,037.124300	348,885.324400
452020	Technology Hardware, Storage & Peripherals	8	6	683,780.161800	2,990,330.868800
452030	Electronic Equipment, Instruments & Components	5	9	69,528.651900	247,786.997500
453010	Semiconductors & Semiconductor Equipment	15	20	451,019.702700	3,171,491.261400
501010	Diversified Telecommunication Services	5	2	352,911.607400	286,793.319500
501020	Wireless Telecommunication Services	0	1	0.000000	187,244.867200
502010	Media	10	9	306,384.898300	335,800.616200
502020	Entertainment	7	6	375,018.200600	484,265.962100
502030	Interactive Media & Services	4	4	561,256.365600	4,375,614.834600
551010	Electric Utilities	15	17	288,794.047200	625,486.663700
551020	Gas Utilities	1	1	5,474.966100	17,636.965600
551030	Multi-Utilities	13	10	160,915.187700	267,319.699900
551040	Water Utilities	0	1	0.000000	25,995.064100
551050	Independent Power and Renewable Electricity Pr...	1	1	10,309.243200	12,950.625500
601025	Industrial REITs	1	1	18,407.888200	124,370.302400
601030	Hotel & Resort REITs	1	1	14,566.007400	13,896.380000
601040	Office REITs	2	1	31,927.444200	11,315.314900
601050	Health Care REITs	3	4	48,686.695200	103,959.131400
601060	Residential REITs	3	7	37,820.255900	126,824.603400
601070	Retail REITs	4	5	81,762.950400	124,768.585200
601080	Specialized REITs	5	10	104,774.755800	466,794.907600
602010	Real Estate Management & Development	1	2	8,728.909400	63,311.518300
na	Unclassified	9	0	107,714.149400	0.000000

In [14]:

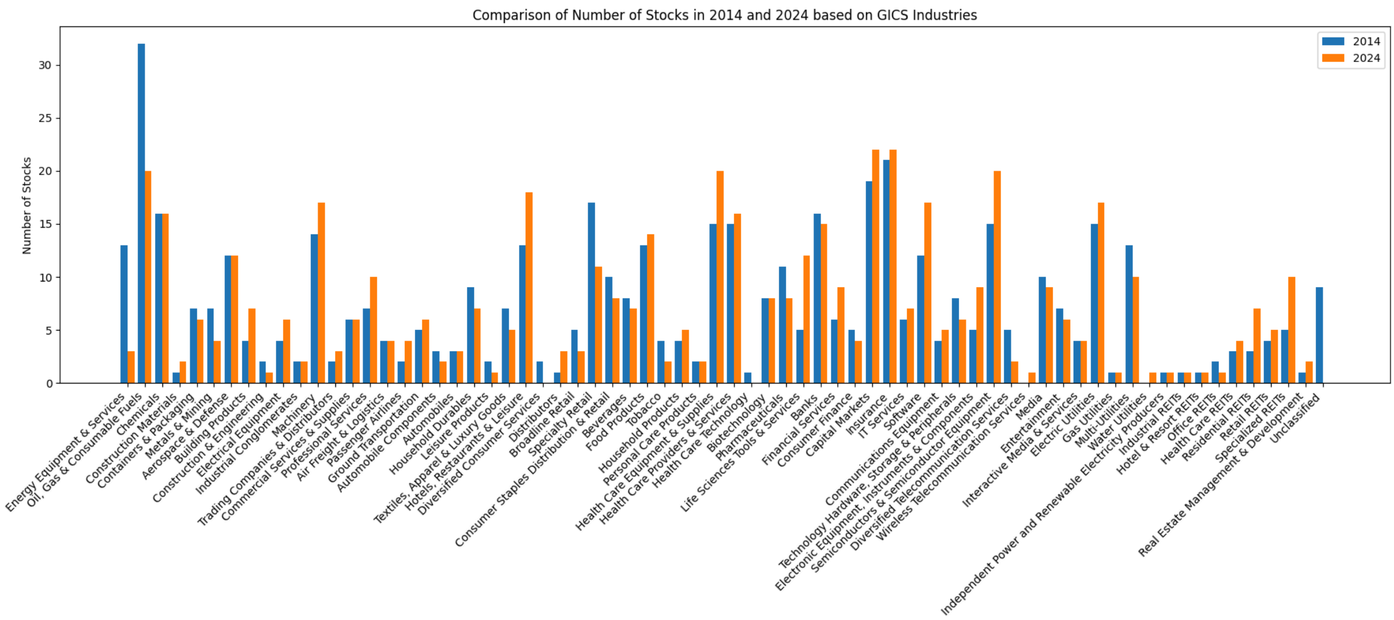
```
# Plot bar chart for Comparison of Number of Stocks in 2014 and 2024 based on GICS Sectors
fig, ax = plt.subplots(figsize=(12, 6))
bar_width = 0.4
x = range(len(gics_sectors))

# Plot bars for 2014 and 2024
ax.bar(x, gics_sectors['No. of Stocks 2014'], width=bar_width, label='2014', align='center')
ax.bar([i + bar_width for i in x], gics_sectors['No. of Stocks 2024'], width=bar_width, label='2024', align='cent
```

```python
# Set labels
ax.set_xticks([i + bar_width / 2 for i in x])
ax.set_xticklabels(gics_sectors['Sector_Name'], rotation=45, ha='right')
ax.set_ylabel('Number of Stocks')
ax.set_title('Comparison of Number of Stocks in 2014 and 2024 based on GICS Sectors')
ax.legend()
plt.tight_layout()
plt.show()
```

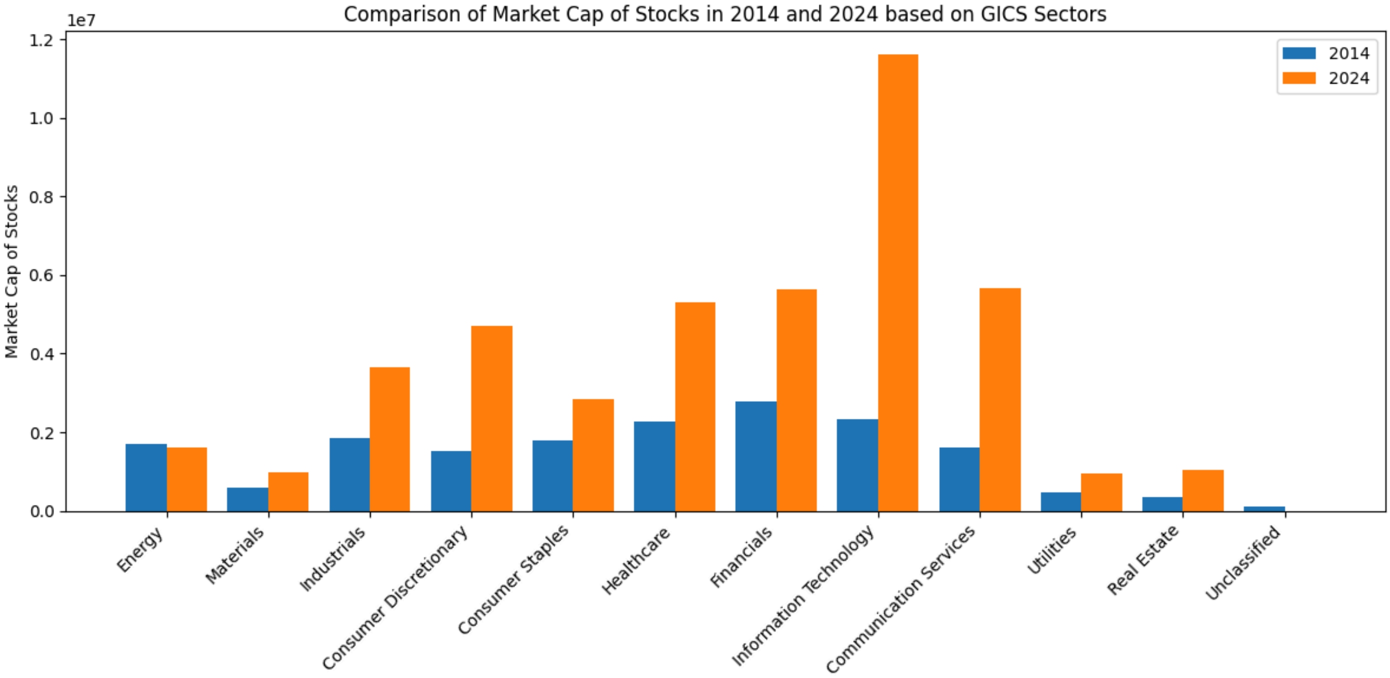


In [15]:

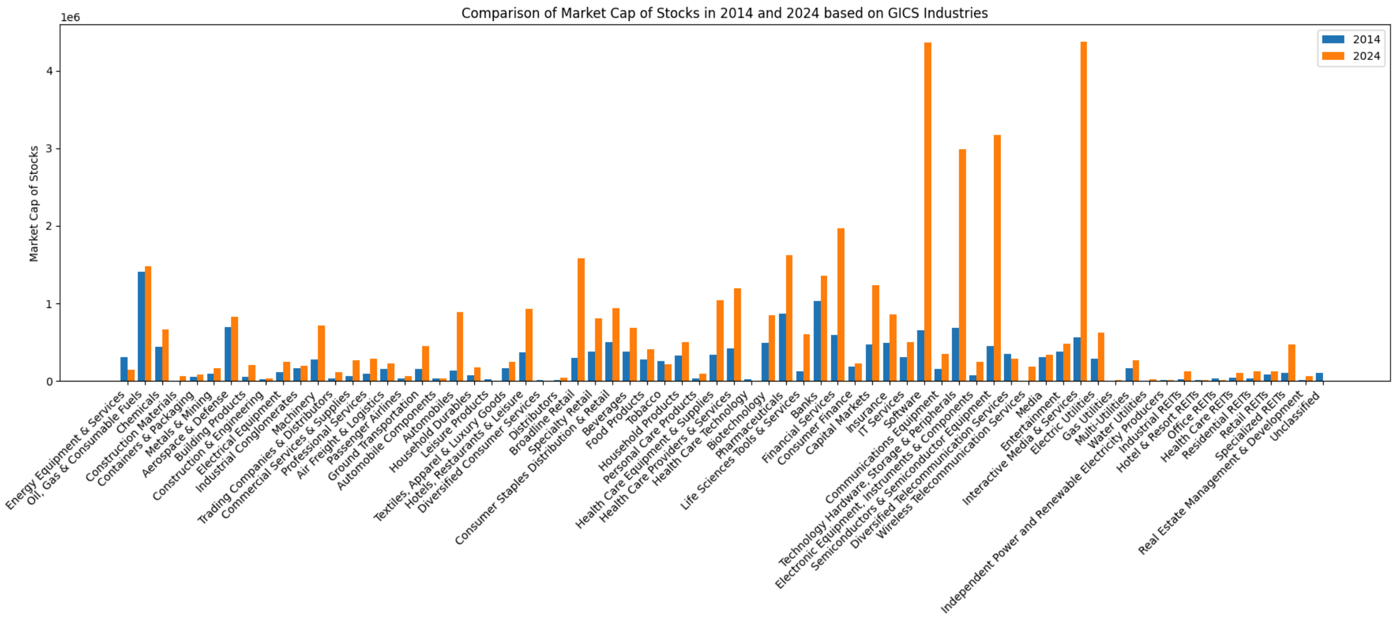
```python
# Plot bar chart for Comparison of Number of Stocks in 2014 and 2024 based on GICS Industries
fig, ax = plt.subplots(figsize=(18, 8))
bar_width = 0.4
x = range(len(gics_industries))

# Plot bars for 2014 and 2024
ax.bar(x, gics_industries['No. of Stocks 2014'], width=bar_width, label='2014', align='center')
ax.bar([i + bar_width for i in x], gics_industries['No. of Stocks 2024'], width=bar_width, label='2024', align='c

# Set labels
ax.set_xticks([i + bar_width / 2 for i in x])
ax.set_xticklabels(gics_industries['Industry_Name'], rotation=45, ha='right')
ax.set_ylabel('Number of Stocks')
ax.set_title('Comparison of Number of Stocks in 2014 and 2024 based on GICS Industries')
ax.legend()
plt.tight_layout()
plt.show()
```



In [16]:
```python
# Plot bar chart for Comparison of Market Cap of Stocks in 2014 and 2024 based on GICS Sectors
fig, ax = plt.subplots(figsize=(12, 6))
bar_width = 0.4
x = range(len(gics_sectors))

# Plot bars for 2014 and 2024
ax.bar(x, gics_sectors['Market Cap of Stocks 2014'], width=bar_width, label='2014', align='center')
ax.bar([i + bar_width for i in x], gics_sectors['Market Cap of Stocks 2024'], width=bar_width, label='2024', alig

# Set labels
ax.set_xticks([i + bar_width / 2 for i in x])
ax.set_xticklabels(gics_sectors['Sector_Name'], rotation=45, ha='right')
ax.set_ylabel('Market Cap of Stocks')
ax.set_title('Comparison of Market Cap of Stocks in 2014 and 2024 based on GICS Sectors')
ax.legend()
plt.tight_layout()
plt.show()
```



In [17]:
```python
# Plot bar chart for Comparison of Market Cap of Stocks in 2014 and 2024 based on GICS Industries
fig, ax = plt.subplots(figsize=(18, 8))
bar_width = 0.4
x = range(len(gics_industries))

# Plot bars for 2014 and 2024
ax.bar(x, gics_industries['Market Cap of Stocks 2014'], width=bar_width, label='2014', align='center')
ax.bar([i + bar_width for i in x], gics_industries['Market Cap of Stocks 2024'], width=bar_width, label='2024', a

# Set labels
ax.set_xticks([i + bar_width / 2 for i in x])  # Center tick labels
ax.set_xticklabels(gics_industries['Industry_Name'], rotation=45, ha='right')
ax.set_ylabel('Market Cap of Stocks')
ax.set_title('Comparison of Market Cap of Stocks in 2014 and 2024 based on GICS Industries')
ax.legend()
plt.tight_layout()
plt.show()
```



In [18]:

```
print('Top 3 sectors by Market Cap at the beginning of 2014')
top_3_sectors_2014 = gics_sectors.nlargest(3, 'Market Cap of Stocks 2014')
print(top_3_sectors_2014[['Sector_Name', 'Market Cap of Stocks 2014']])
print()

print('Top 3 sectors by Market Cap at the beginning of 2024')
top_3_sectors_2024 = gics_sectors.nlargest(3, 'Market Cap of Stocks 2024')
print(top_3_sectors_2024[['Sector_Name', 'Market Cap of Stocks 2024']])
```

```
Top 3 sectors by Market Cap at the beginning of 2014
                 Sector_Name  Market Cap of Stocks 2014
Code
40                Financials           2,783,578.085500
45    Information Technology           2,318,796.187700
35                Healthcare           2,259,201.379500

Top 3 sectors by Market Cap at the beginning of 2024
                 Sector_Name  Market Cap of Stocks 2024
Code
45    Information Technology          11,628,815.501800
50    Communication Services           5,669,719.599600
40                Financials           5,644,229.939100
```

In [19]:

```
print('Top 3 industries by Market Cap at the beginning of 2014')
top_3_industries_2014 = gics_industries.nlargest(3, 'Market Cap of Stocks 2014')
print(top_3_industries_2014[['Industry_Name', 'Market Cap of Stocks 2014']])
print()

print('Top 3 industries by Market Cap at the beginning of 2024')
top_3_industries_2024 = gics_industries.nlargest(3, 'Market Cap of Stocks 2024')
print(top_3_industries_2024[['Industry_Name', 'Market Cap of Stocks 2024']])
```

```
Top 3 industries by Market Cap at the beginning of 2014
                      Industry_Name  Market Cap of Stocks 2014
Code
101020  Oil, Gas & Consumable Fuels           1,404,947.820500
401010                        Banks           1,031,024.225700
352020              Pharmaceuticals             863,815.602900

Top 3 industries by Market Cap at the beginning of 2024
                                   Industry_Name  Market Cap of Stocks 2024
Code
502030                Interactive Media & Services           4,375,614.834600
451030                                    Software           4,364,983.241300
453010  Semiconductors & Semiconductor Equipment           3,171,491.261400
```

In [ ]:

In [ ]:

In [ ]:

Question 2

In [20]:
```
# Identify the industry code for NVDA and list stocks in the same industry for 2024

nvda_industry_code = tickers.loc['NVDA']['GICS_Code'][:6]
print ('NVDA industry is: Code-' , nvda_industry_code, ' Industry_Name-', gics_industries_dict[nvda_industry_code
nvda_industry_tickers = []

# Find all stocks in the 2024 universe that belong to the same industry as NVDA
for ticker in universe.columns:
    if universe.loc[2024, ticker] == 1:
        curr_code = str(tickers.loc[ticker, 'GICS_Code'])
        if curr_code != 'nan':
            industry_code = curr_code[:6]

        if nvda_industry_code == industry_code:
            nvda_industry_tickers.append(ticker)
print()
print('Stocks in the 2024 universe that are also in the industry NVDA belongs to:')
print(nvda_industry_tickers)
```

```
NVDA industry is: Code- 453010  Industry_Name- Semiconductors & Semiconductor Equipment

Stocks in the 2024 universe that are also in the industry NVDA belongs to:
['ADI', 'AMAT', 'AMD', 'AVGO', 'ENPH', 'FSLR', 'INTC', 'KLAC', 'LRCX', 'MCHP', 'MPWR', 'MU', 'NVDA', 'NXPI', 'ON',
'QCOM', 'QRVO', 'SWKS', 'TER', 'TXN']
```

Question 3

In [21]:
```
# Filter the adjusted close price data to include only stocks in NVDA's industry

adjusted = adjusted[nvda_industry_tickers]
adjusted.head()
```

Out[21]:

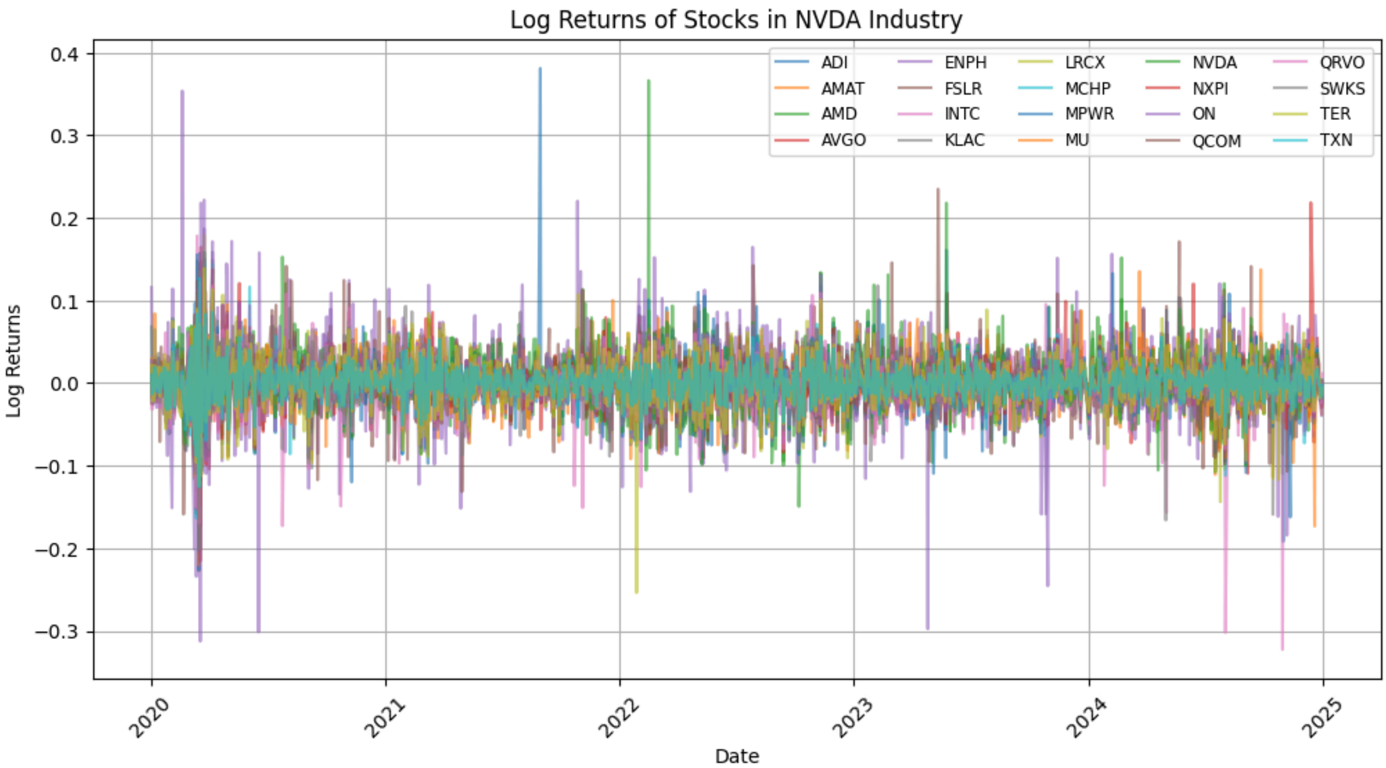
	ADI	AMAT	AMD	AVGO	ENPH	FSLR	INTC	KLAC	LRCX	M
Date										
2004-01-02	17,024.956200	36,755.459100	5,180.057800	NaN	NaN	NaN	210,069.120000	10,989.477600	4,137.879200	6,973.98
2004-01-05	17,549.201400	39,117.122100	5,298.578600	NaN	NaN	NaN	214,968.120000	11,534.667800	4,418.435500	7,083.92
2004-01-06	17,943.314800	39,782.379200	5,441.500800	NaN	NaN	NaN	214,902.800000	11,478.201700	4,394.947100	6,957.38
2004-01-07	18,121.781300	39,649.327800	5,458.930300	NaN	NaN	NaN	222,022.680000	11,406.158700	4,358.409500	7,056.95
2004-01-08	18,445.251700	40,381.110700	5,553.049800	NaN	NaN	NaN	223,655.680000	11,575.557100	4,419.740400	7,187.64

In [22]:
```
# Calculate log returns for each ticker in NVDA's industry
log_returns = adjusted.copy()
for ticker in nvda_industry_tickers:
    log_returns[ticker] = np.log(adjusted[ticker] / adjusted[ticker].shift(1))

# Filter the log returns data between the specified start and end dates
start_date = '2020-01-01'
end_date = '2024-12-31'
log_returns = log_returns.loc[start_date:end_date]
log_returns.head()
```

Out[22]:

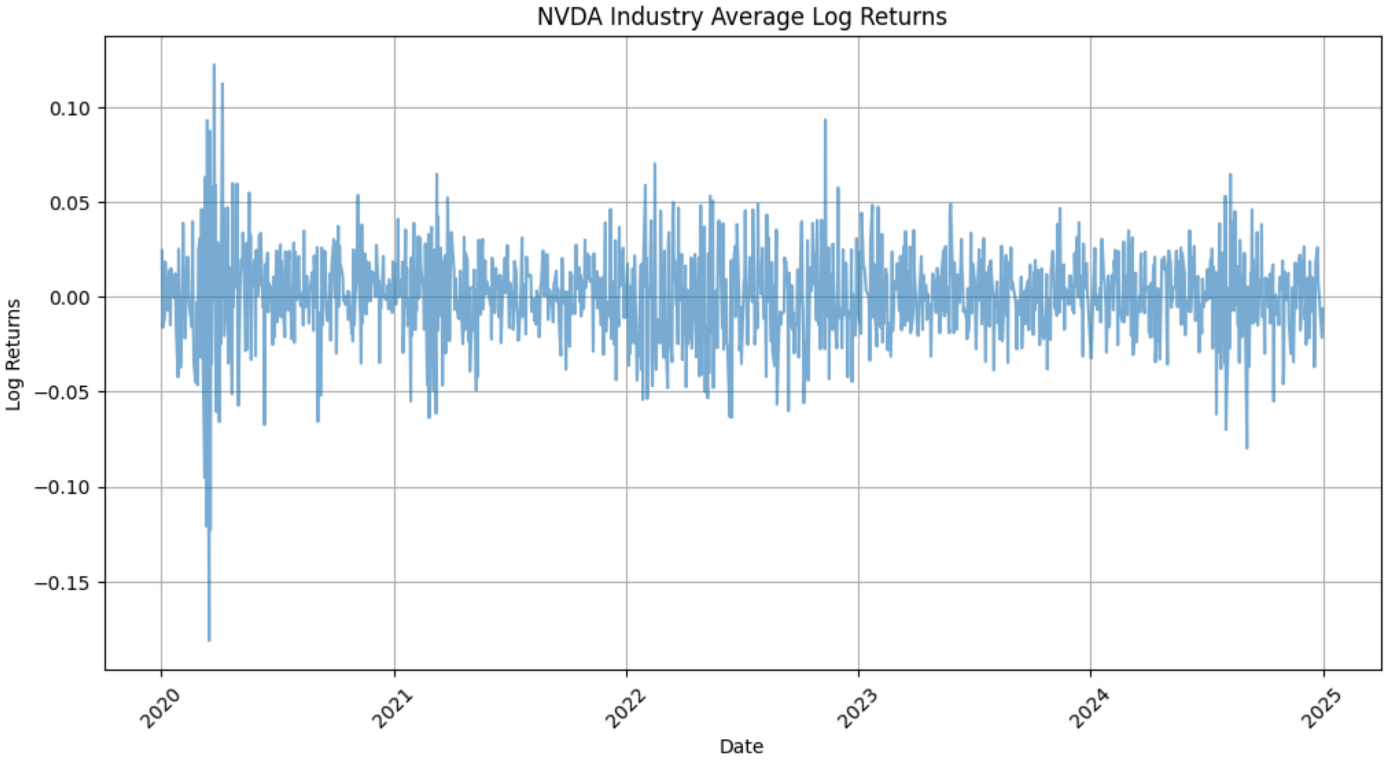
Date	ADI	AMAT	AMD	AVGO	ENPH	FSLR	INTC	KLAC	LRCX	MCHP	MPWR
2020-01-02	0.013291	0.018826	0.068266	0.019956	0.115868	0.022616	0.016406	0.024396	0.018602	0.027131	0.011117
2020-01-03	-0.017760	-0.016044	-0.010236	-0.025764	-0.001706	-0.003851	-0.012238	-0.019301	-0.010800	-0.022177	-0.018276
2020-01-06	-0.011818	-0.021801	-0.004330	-0.001497	0.012553	-0.021091	-0.002833	-0.022761	-0.018840	-0.014452	-0.010293
2020-01-07	0.022496	0.028477	-0.002897	-0.003448	0.011065	0.014934	-0.016827	0.007909	0.024086	0.064942	0.009274
2020-01-08	0.008991	-0.000649	-0.008743	-0.012553	0.033123	0.000529	0.000679	0.001303	0.012513	-0.012729	-0.001303

In [23]:
```python
# Plot the log returns of stocks in NVDA's industry over time

plt.figure(figsize=(12, 6))

# Plot each stock's log returns with a label
for ticker in nvda_industry_tickers:
    plt.plot(log_returns.index, log_returns[ticker], label=ticker, alpha=0.6)

# Set labels, title, and legend
plt.xlabel("Date")
plt.ylabel("Log Returns")
plt.title("Log Returns of Stocks in NVDA Industry")
plt.legend(loc="upper right", fontsize="small", ncol=5)
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



In [24]:
```python
# Calculate the average log return of all stocks in NVDA's industry and plot the result
log_returns['Industry Return'] = log_returns[nvda_industry_tickers].mean(axis=1)

# Plot the average industry log returns over time
plt.figure(figsize=(12, 6))
plt.plot(log_returns.index, log_returns['Industry Return'], alpha=0.6)
plt.xlabel("Date")
plt.ylabel("Log Returns")
plt.title("NVDA Industry Average Log Returns")
```

```python
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



In [25]:
```python
# Perform linear regression between NVDA's log returns and the industry average returns
X = sm.add_constant(log_returns['Industry Return'])
y = log_returns['NVDA']

model = sm.OLS(y, X).fit()

# Print the results of the regression
print("Value of Alpha:", model.params['const'])
print("Value of Beta:", model.params['Industry Return'])
print("R-squared of regression:", model.rsquared)
```

```
Value of Alpha: 0.001820724872983911
Value of Beta: 1.108654187377237
R-squared of regression: 0.6251782477752315
```

Question 4

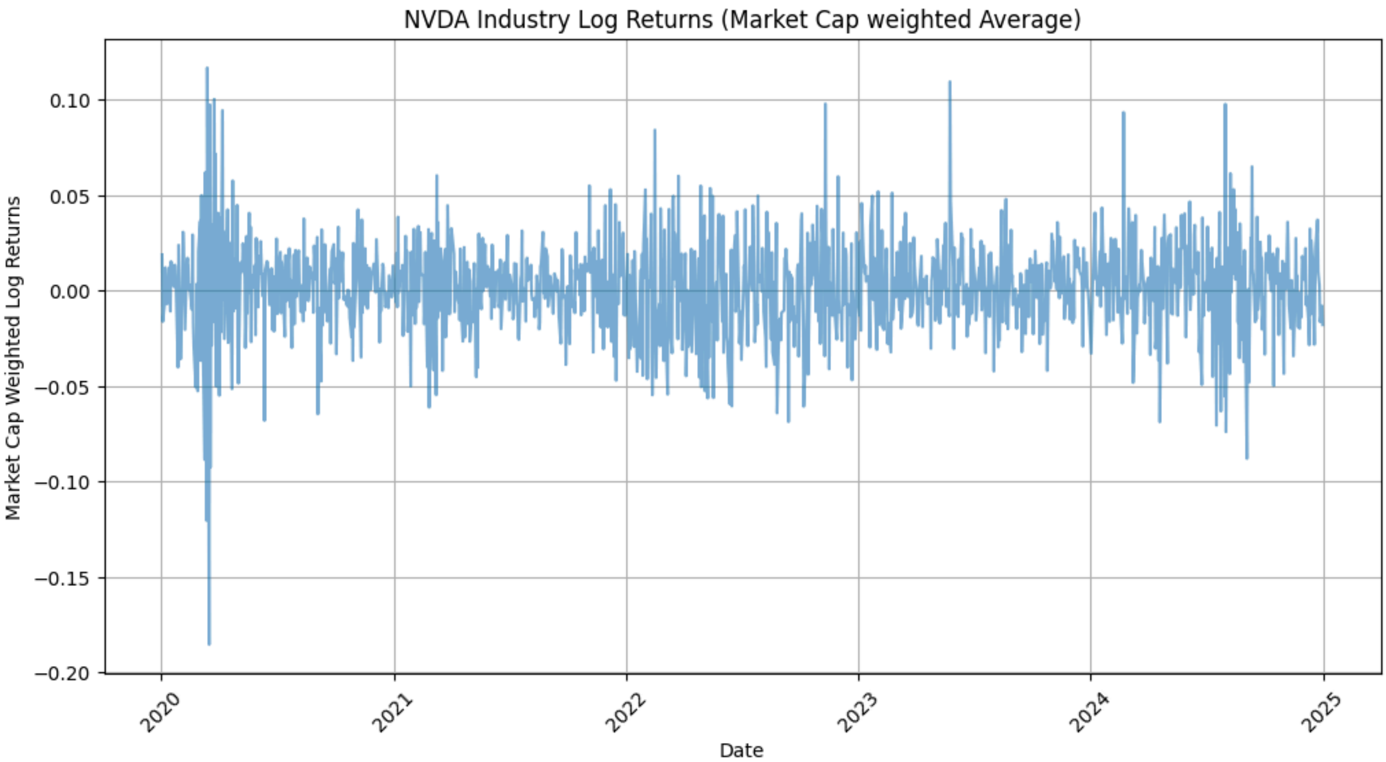
In [26]:
```python
# Filter market cap data for the given date range and calculate weighted log returns for NVDA's industry stocks
market_cap_filtered = market_cap.loc[start_date:end_date][nvda_industry_tickers]
weights = market_cap_filtered.div(market_cap_filtered.sum(axis=1), axis=0)

# Multiply log returns by weights to get weighted log returns
log_returns_weighted = log_returns * weights

# Calculate industry return as the weighted average of the stocks' returns
log_returns_weighted['Industry Return'] = log_returns_weighted[nvda_industry_tickers].sum(axis=1)
log_returns_weighted.head()
```

Out[26]:

Date	ADI	AMAT	AMD	AVGO	ENPH	FSLR	INTC	Industry Return	KLAC	LRCX	MCHP
2020-01-02	0.000497	0.000903	0.003145	0.002156	0.000351	0.000115	0.003658	0.018983	0.000592	0.000677	0.000588
2020-01-03	-0.000663	-0.000770	-0.000474	-0.002757	-0.000005	-0.000020	-0.002739	-0.016100	-0.000467	-0.000395	-0.000478
2020-01-06	-0.000439	-0.001030	-0.000201	-0.000161	0.000039	-0.000107	-0.000637	-0.006761	-0.000542	-0.000681	-0.000309
2020-01-07	0.000844	0.001368	-0.000133	-0.000365	0.000035	0.000076	-0.003674	0.012223	0.000188	0.000882	0.001463
2020-01-08	0.000341	-0.000031	-0.000397	-0.001314	0.000107	0.000003	0.000148	-0.000798	0.000031	0.000464	-0.000283

In [27]:

```
# Plot the market cap weighted average log returns for NVDA's industry
plt.figure(figsize=(12, 6))
plt.plot(log_returns_weighted.index, log_returns_weighted['Industry Return'], alpha=0.6)
plt.xlabel("Date")
plt.ylabel("Market Cap Weighted Log Returns")
plt.title("NVDA Industry Log Returns (Market Cap weighted Average)")
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



In [28]:

```
# Perform linear regression between NVDA's log returns and the market cap weighted industry returns
X = sm.add_constant(log_returns_weighted['Industry Return'])
y = log_returns['NVDA']

model = sm.OLS(y, X).fit()

# Print the results of the regression
print("Value of Alpha:", model.params['const'])
print("Value of Beta:", model.params['Industry Return'])
print("R-squared of regression:", model.rsquared)
```

```
Value of Alpha: 0.0007526980760657113
Value of Beta: 1.2345535145383124
R-squared of regression: 0.8447675448443841
```