In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import pandas_datareader.data as web
import datetime
```

1.1 Data

In [ ]:
```
#Reading in MSF File
path = r"C:\Users\mhlad\OneDrive\Desktop\Georgia Tech\Chava\Homework 3\msf_1926_202
share_code = "SHRCD"
code_filter = [10, 11]
try:
    df = pd.read_csv(path)
except FileNotFoundError:
    print("Could not find the file.")
    exit()

filtered_df = df[df[share_code].isin(code_filter)]
filtered_df['RET'] = pd.to_numeric(filtered_df['RET'], errors='coerce')
numeric_columns = filtered_df.select_dtypes(include=['float64', 'int64']).columns

# Fill missing values using forward and backward fills for numeric columns only
forward_filled = filtered_df[numeric_columns].fillna(method='ffill')
backward_filled = filtered_df[numeric_columns].fillna(method='bfill')

# Average the filled data
averaged_fill = (forward_filled + backward_filled) / 2

# Fill the original dataframe with averaged values for numeric columns only
filtered_df[numeric_columns] = filtered_df[numeric_columns].fillna(averaged_fill)
```

```
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\3507690399.py:6: DtypeWarning: Col
umns (9) have mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv(path)
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\3507690399.py:12: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_df['RET'] = pd.to_numeric(filtered_df['RET'], errors='coerce')
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\3507690399.py:23: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_df[numeric_columns] = filtered_df[numeric_columns].fillna(averaged_fill)
```

In [ ]:
```
#Reading in CPI file
path = r"C:\Users\mhlad\OneDrive\Desktop\Georgia Tech\Chava\Homework 3\CPIAUCNS.csv
try:
    cpi_df = pd.read_csv(path)
except FileNotFoundError:
    print("Could not find the file.")
    exit()
```

In [ ]:
```
filtered_df['market_cap'] = abs(filtered_df['PRC']) * filtered_df['SHROUT']
filtered_df['date'] = pd.to_datetime(filtered_df["date"])
filtered_df['Month'] = filtered_df['date'].dt.month
filtered_df['Year'] = filtered_df['date'].dt.year
filtered_df['MonthYear'] = filtered_df['Month'].astype(str) + "-" + filtered_df['Ye
filtered_df = filtered_df.drop("Month", axis = 1)
filtered_df = filtered_df.drop("Year", axis = 1)

cpi_df['DATE'] = pd.to_datetime(cpi_df["DATE"])
cpi_df['Month'] = cpi_df['DATE'].dt.month
cpi_df['Year'] = cpi_df['DATE'].dt.year
cpi_df['MonthYear'] = cpi_df['Month'].astype(str) + "-" + cpi_df['Year'].astype(str
cpi_df = cpi_df.drop("Month", axis = 1)
cpi_df = cpi_df.drop("Year", axis = 1)
```

```
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\4130001579.py:1: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_df['market_cap'] = abs(filtered_df['PRC']) * filtered_df['SHROUT']
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\4130001579.py:2: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_df['date'] = pd.to_datetime(filtered_df["date"])
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\4130001579.py:3: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_df['Month'] = filtered_df['date'].dt.month
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\4130001579.py:4: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_df['Year'] = filtered_df['date'].dt.year
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\4130001579.py:5: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_df['MonthYear'] = filtered_df['Month'].astype(str) + "-" + filtered_df['Y
ear'].astype(str)
```

In [ ]:

```
print(filtered_df)
```

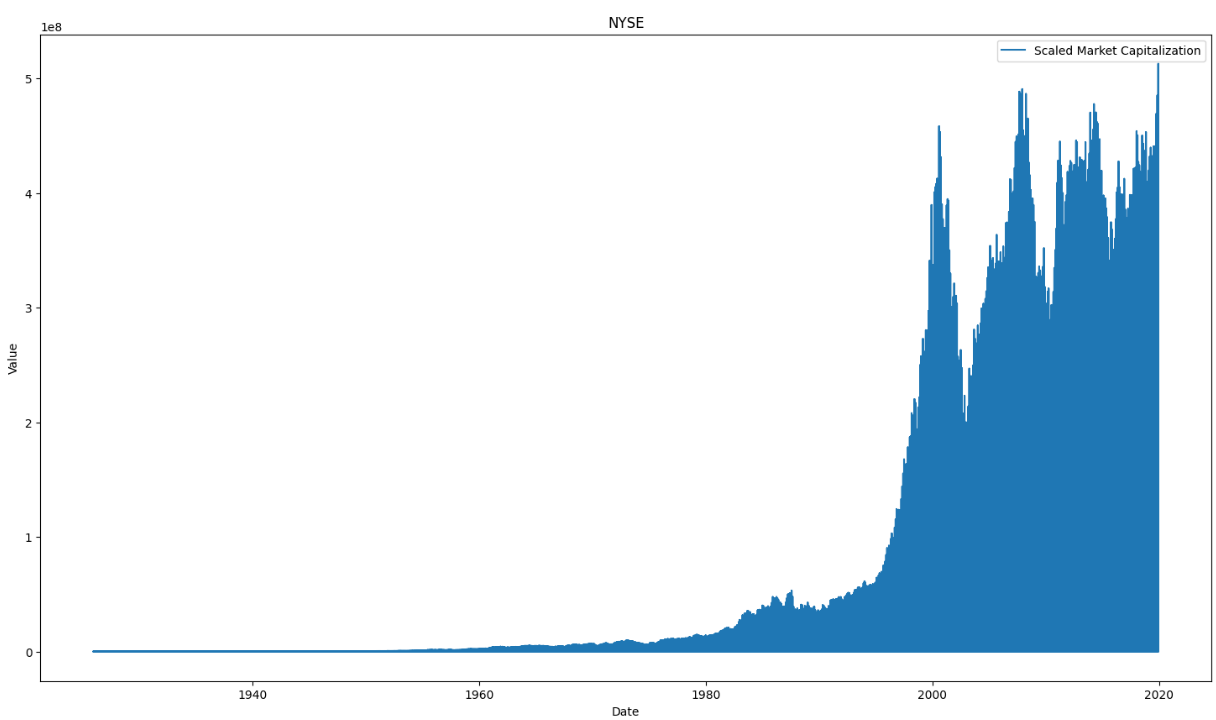
```
        PERMNO       date  SHRCD  EXCHCD TICKER                    COMNAM  
1        10000 1986-01-31   10.0     3.0  OMFGA  OPTIMUM MANUFACTURING INC   \
2        10000 1986-02-28   10.0     3.0  OMFGA  OPTIMUM MANUFACTURING INC   
3        10000 1986-03-31   10.0     3.0  OMFGA  OPTIMUM MANUFACTURING INC   
4        10000 1986-04-30   10.0     3.0  OMFGA  OPTIMUM MANUFACTURING INC   
5        10000 1986-05-30   10.0     3.0  OMFGA  OPTIMUM MANUFACTURING INC   
...        ...        ...    ...     ...    ...                        ...   
4927547  93436 2022-08-31   11.0     3.0   TSLA                  TESLA INC   
4927548  93436 2022-09-30   11.0     3.0   TSLA                  TESLA INC   
4927549  93436 2022-10-31   11.0     3.0   TSLA                  TESLA INC   
4927550  93436 2022-11-30   11.0     3.0   TSLA                  TESLA INC   
4927551  93436 2022-12-30   11.0     3.0   TSLA                  TESLA INC   

        PERMCO  ISSUNO  HEXCD HSICCD  ...     SHROUT  CFACPR  CFACSHR  
1         7952   10396      3   3990  ...     3680.0     1.0      1.0  \
2         7952   10396      3   3990  ...     3680.0     1.0      1.0   
3         7952   10396      3   3990  ...     3680.0     1.0      1.0   
4         7952   10396      3   3990  ...     3793.0     1.0      1.0   
5         7952   10396      3   3990  ...     3793.0     1.0      1.0   
...        ...     ...    ...    ...  ...        ...     ...      ...   
4927547  53453   66252      3   9999  ...  3133470.0     1.0      1.0   
4927548  53453   66252      3   9999  ...  3158000.0     1.0      1.0   
4927549  53453   66252      3   9999  ...  3157752.0     1.0      1.0   
4927550  53453   66252      3   9999  ...  3157752.0     1.0      1.0   
4927551  53453   66252      3   9999  ...  3157752.0     1.0      1.0   

           ALTPRC   SPREAD    ALTPRCDT       RETX    vwretd    market_cap  
1        -4.37500  0.25000  1986-01-31          C  0.009830  1.610000e+04  \
2        -3.25000  0.25000  1986-02-28  -0.257143  0.072501  1.196000e+04   
3        -4.43750  0.12500  1986-03-31   0.365385  0.053887  1.633000e+04   
4        -4.00000  0.25000  1986-04-30  -0.098592 -0.007903  1.517200e+04   
5        -3.10938  0.09375  1986-05-30  -0.222656  0.050847  1.179388e+04   
...           ...      ...         ...        ...       ...           ...   
4927547 275.60999      NaN  2022-08-31  -0.072489 -0.036240  8.636156e+08   
4927548 265.25000      NaN  2022-09-30  -0.037589 -0.091324  8.376595e+08   
4927549 227.53999      NaN  2022-10-31  -0.142168  0.077403  7.185149e+08   
4927550 194.70000      NaN  2022-11-30  -0.144326  0.052365  6.148143e+08   
4927551 123.18000      NaN  2022-12-30  -0.367334 -0.057116  3.889719e+08   

        MonthYear  
1          1-1986  
2          2-1986  
3          3-1986  
4          4-1986  
5          5-1986  
...           ...  
4927547    8-2022  
4927548    9-2022  
4927549   10-2022  
4927550   11-2022  
4927551   12-2022  

[3794913 rows x 28 columns]
```

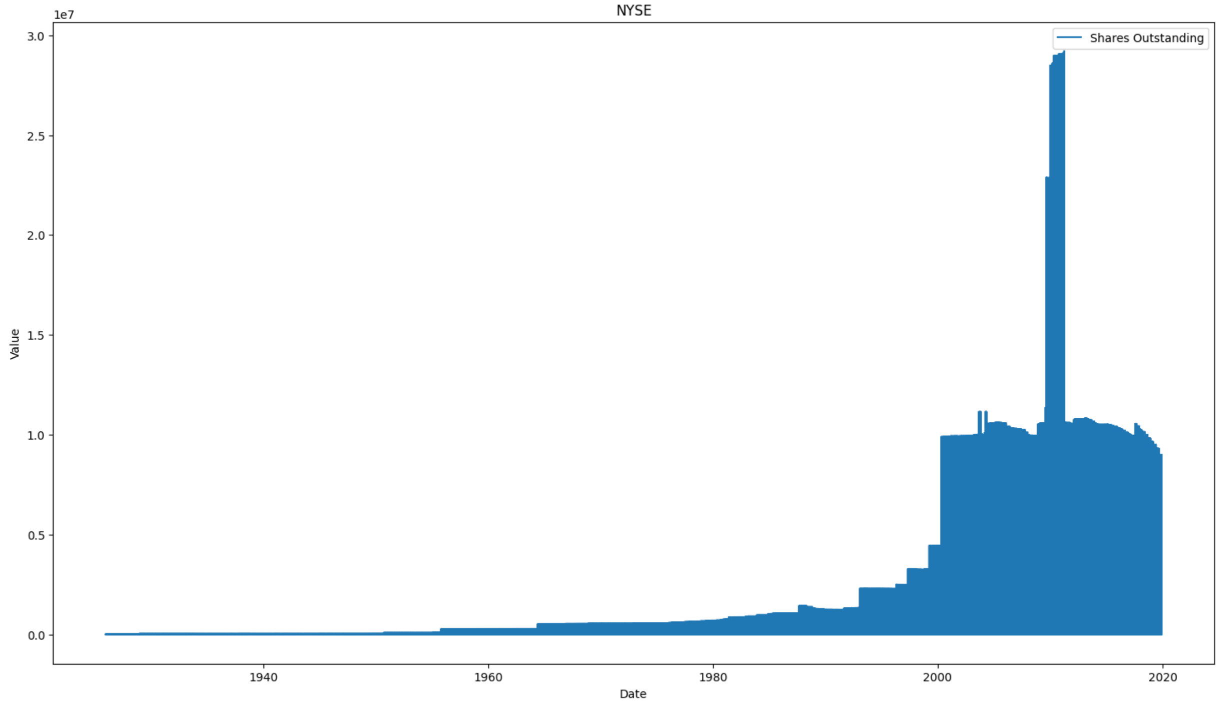
In [ ]:
```
merge_df = pd.merge(left = cpi_df, right = filtered_df, left_on='MonthYear', right_
```

```
merge_df["ScalingFactor"] = merge_df["CPIAUCNS"]/ merge_df.loc[merge_df['MonthYear'
merge_df['ScaledMarketCap'] = merge_df['market_cap'] * merge_df['ScalingFactor']
```

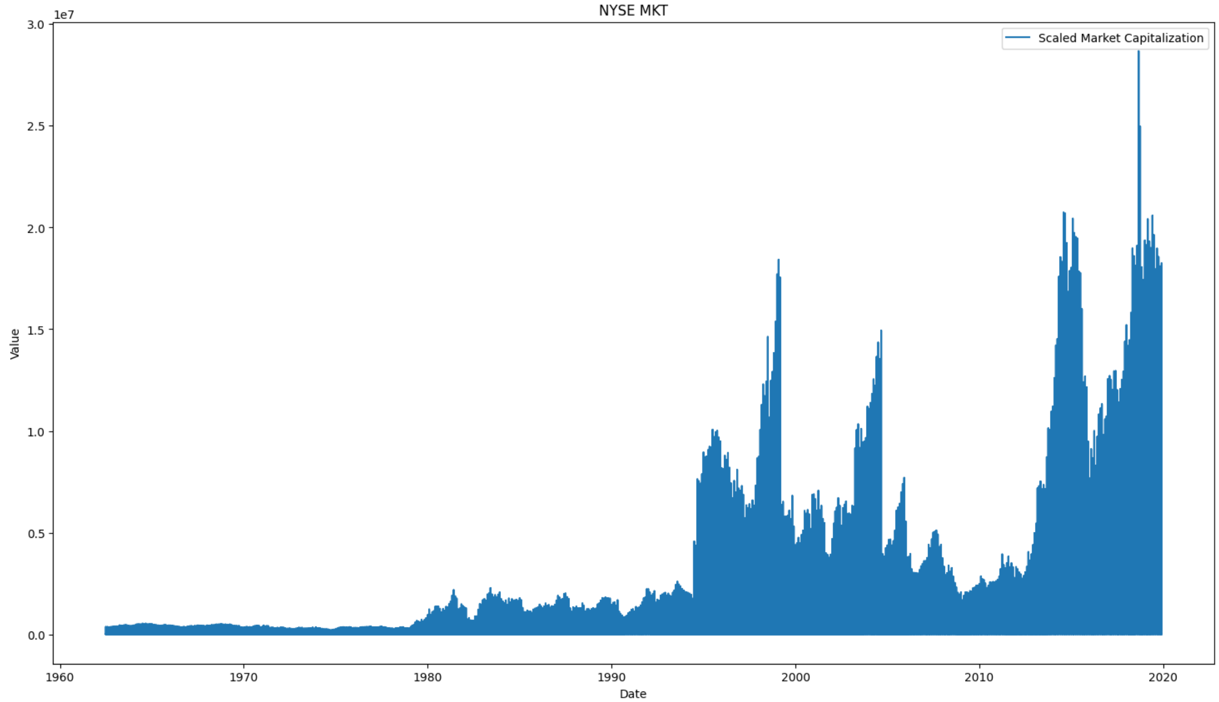
1.2 Plot by EXCHCD over time

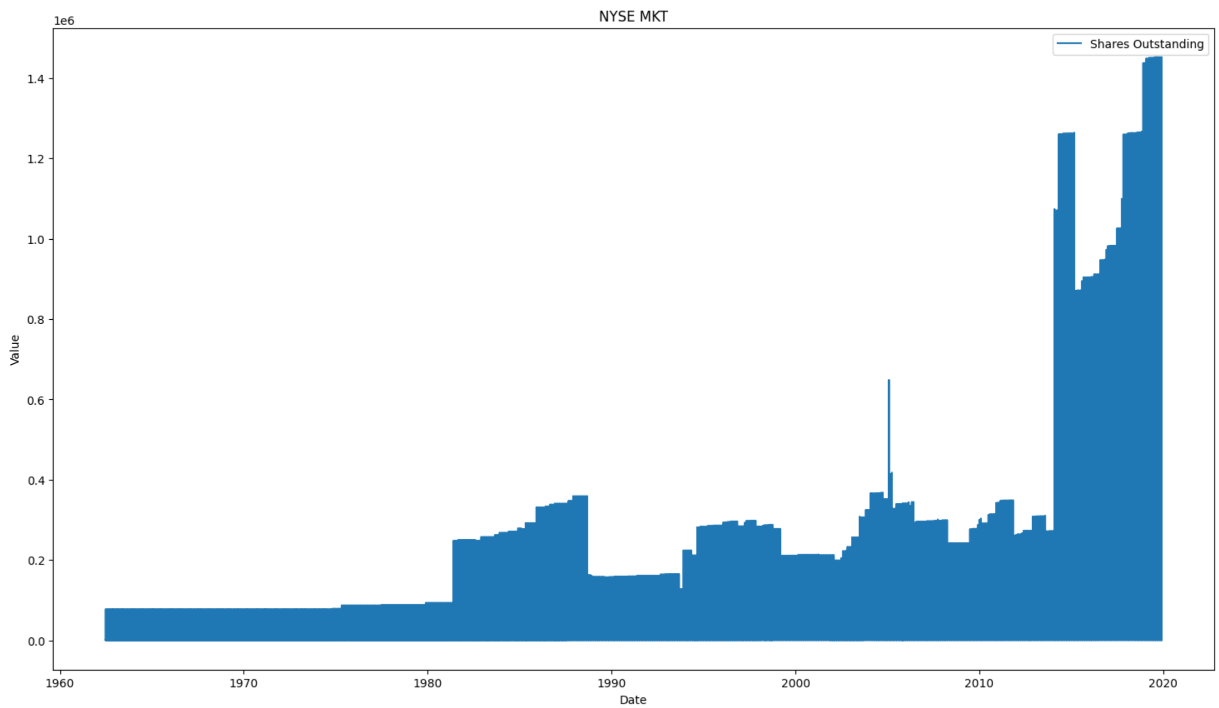
In [ ]:
```
start_date = '1925-12-01'
end_date = '2019-12-01'
exchcd_df = merge_df[(merge_df['DATE'] >= start_date) & (merge_df['DATE'] <= end_da

#plot NYSE - 1
code = 1
exchcd_df = exchcd_df[exchcd_df["EXCHCD"] == code]
exchcd_df.set_index('DATE', inplace = True)

plt.figure(figsize=(18, 10))
plt.plot(exchcd_df.index, exchcd_df["ScaledMarketCap"], label = 'Scaled Market Capi
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("NYSE")
plt.legend()
plt.show()

plt.figure(figsize=(18, 10))
plt.plot(exchcd_df.index, exchcd_df["SHROUT"], label = 'Shares Outstanding')
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("NYSE")
plt.legend()
plt.show()
```





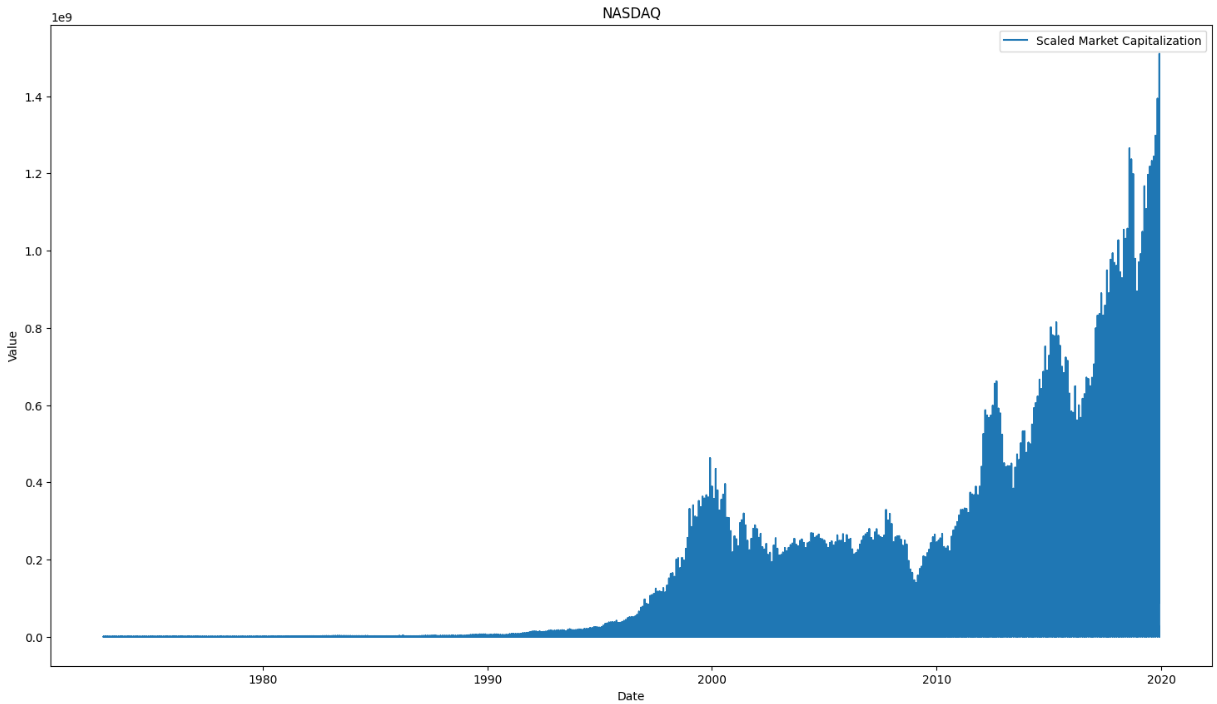
In [ ]:
```
start_date = '1925-12-01'
end_date = '2019-12-01'
exchcd_df = merge_df[(merge_df['DATE'] >= start_date) & (merge_df['DATE'] <= end_da

#plot NYSE MKT -2
code = 2
exchcd_df = exchcd_df[exchcd_df["EXCHCD"] == code]
exchcd_df.set_index('DATE', inplace = True)

plt.figure(figsize=(18, 10))
plt.plot(exchcd_df.index, exchcd_df["ScaledMarketCap"], label = 'Scaled Market Capi
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("NYSE MKT")
plt.legend()
plt.show()

plt.figure(figsize=(18, 10))
plt.plot(exchcd_df.index, exchcd_df["SHROUT"], label = 'Shares Outstanding')
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("NYSE MKT")
plt.legend()
plt.show()
```

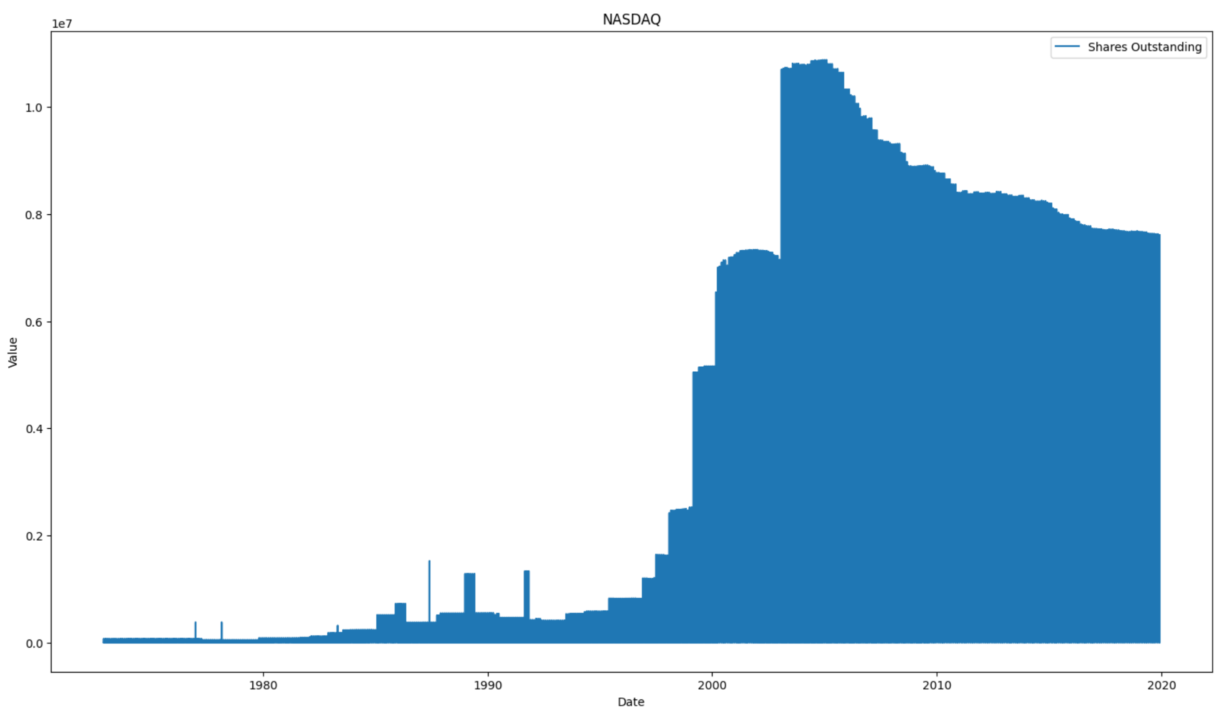




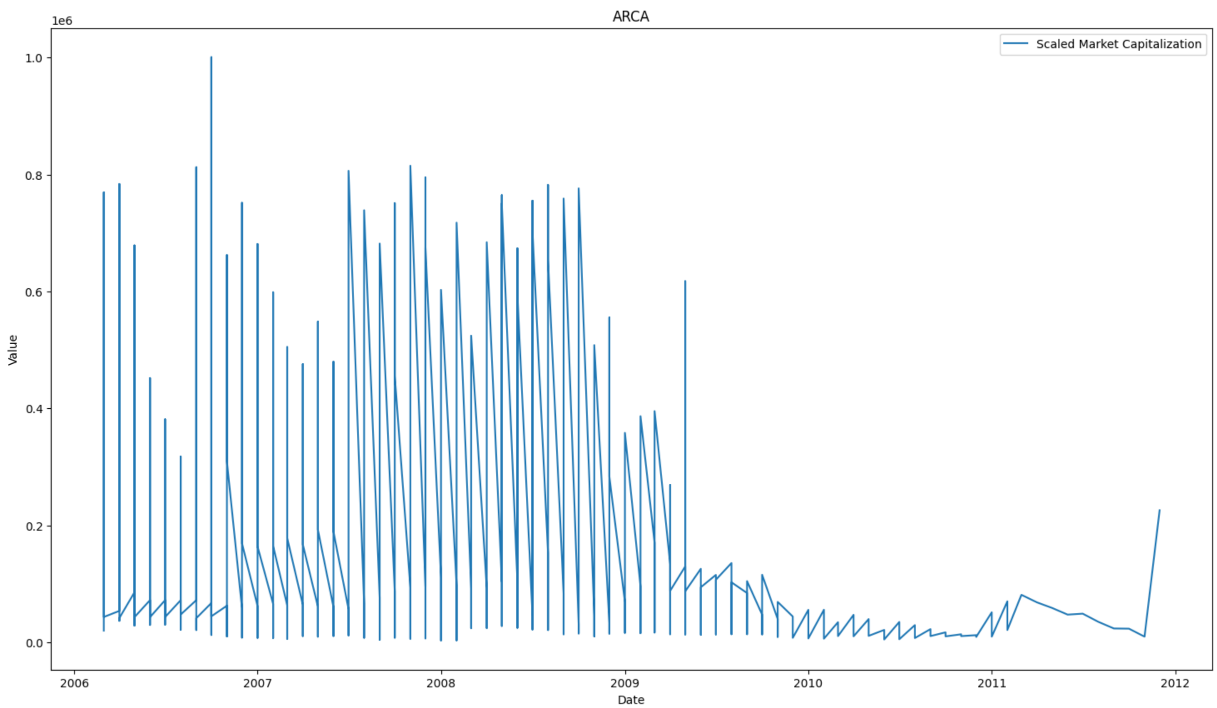
In [ ]:
```
start_date = '1925-12-01'
end_date = '2019-12-01'
exchcd_df = merge_df[(merge_df['DATE'] >= start_date) & (merge_df['DATE'] <= end_da

#plot NASDAQ - 3
code = 3
exchcd_df = exchcd_df[exchcd_df["EXCHCD"] == code]
exchcd_df.set_index('DATE', inplace = True)

plt.figure(figsize=(18, 10))
plt.plot(exchcd_df.index, exchcd_df["ScaledMarketCap"], label = 'Scaled Market Capi
plt.xlabel("Date")
plt.ylabel("Value")
```

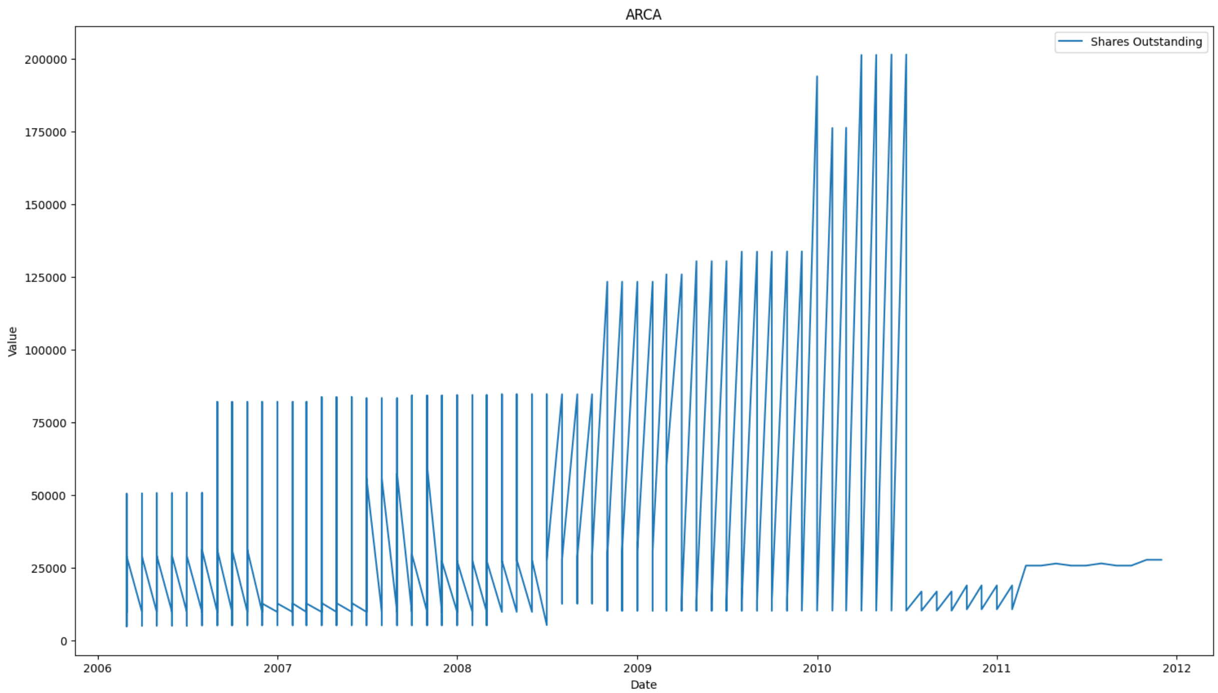
```
plt.title("NASDAQ")
plt.legend()
plt.show()

plt.figure(figsize=(18, 10))
plt.plot(exchcd_df.index, exchcd_df["SHROUT"], label = 'Shares Outstanding')
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("NASDAQ")
plt.legend()
plt.show()
```



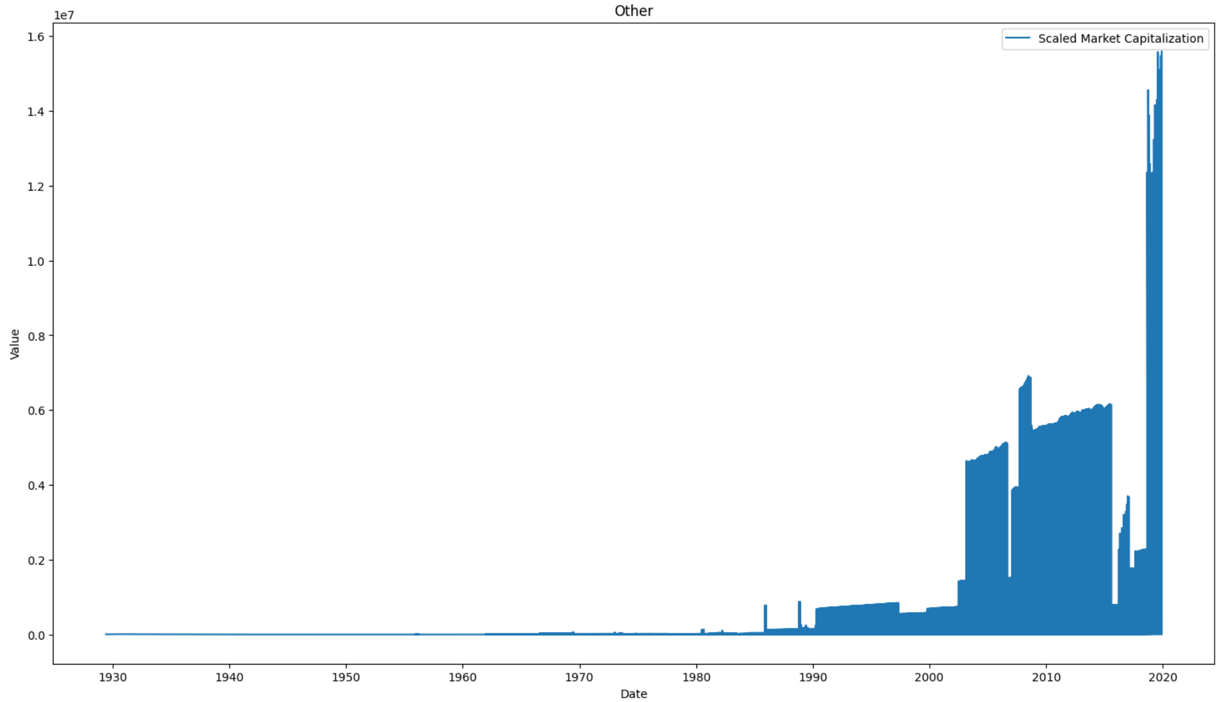


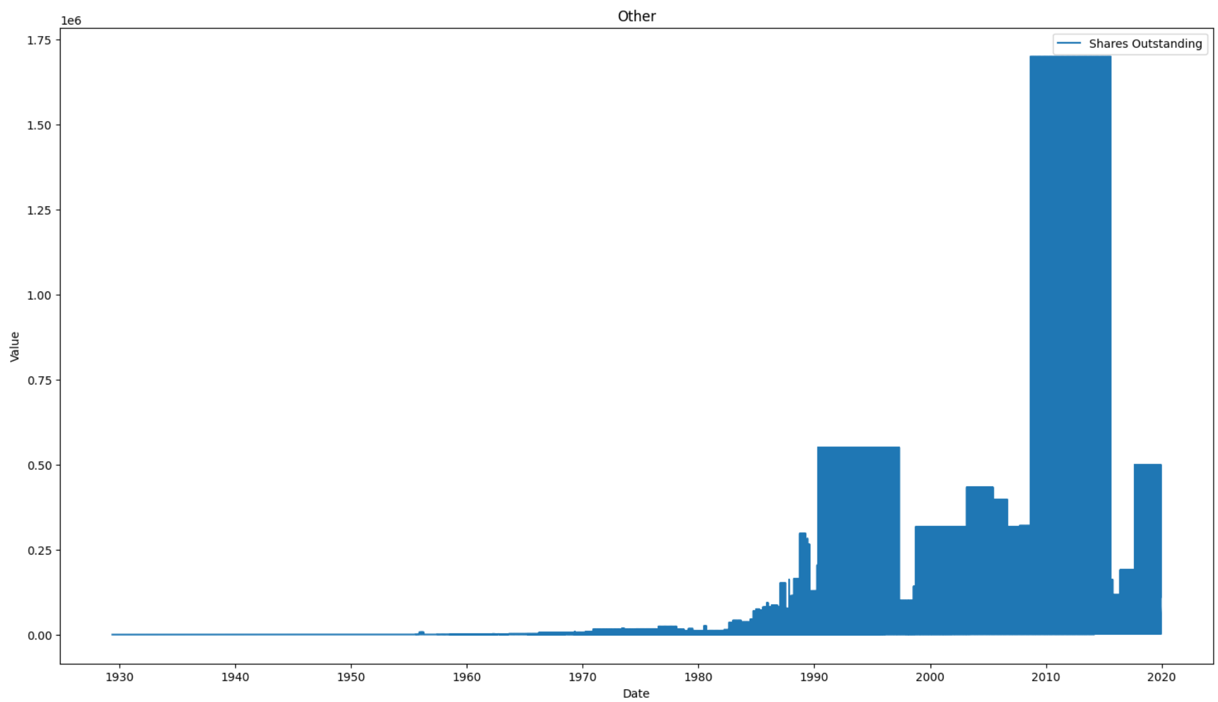
In [ ]:
```
start_date = '1925-12-01'
end_date = '2019-12-01'
exchcd_df = merge_df[(merge_df['DATE'] >= start_date) & (merge_df['DATE'] <= end_da

#plot ARCA - 4
code = 4
exchcd_df = exchcd_df[exchcd_df["EXCHCD"] == code]
exchcd_df.set_index('DATE', inplace = True)

plt.figure(figsize=(18, 10))
plt.plot(exchcd_df.index, exchcd_df["ScaledMarketCap"], label = 'Scaled Market Capi
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("ARCA")
plt.legend()
plt.show()

plt.figure(figsize=(18, 10))
plt.plot(exchcd_df.index, exchcd_df["SHROUT"], label = 'Shares Outstanding')
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("ARCA")
plt.legend()
plt.show()
```





In [ ]:
```
start_date = '1925-12-01'
end_date = '2019-12-01'
exchcd_df = merge_df[(merge_df['DATE'] >= start_date) & (merge_df['DATE'] <= end_da

#plot other - 1
code = [1, 2, 3, 4,31, 32, 33, 34]
exchcd_df = exchcd_df[~exchcd_df["EXCHCD"].isin(code) ]
exchcd_df.set_index('DATE', inplace = True)

plt.figure(figsize=(18, 10))
plt.plot(exchcd_df.index, exchcd_df["ScaledMarketCap"], label = 'Scaled Market Capi
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Other")
plt.legend()
plt.show()

plt.figure(figsize=(18, 10))
plt.plot(exchcd_df.index, exchcd_df["SHROUT"], label = 'Shares Outstanding')
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Other")
plt.legend()
plt.show()
```





1.3 Plot Number of Stocks and Market Capitalization over time

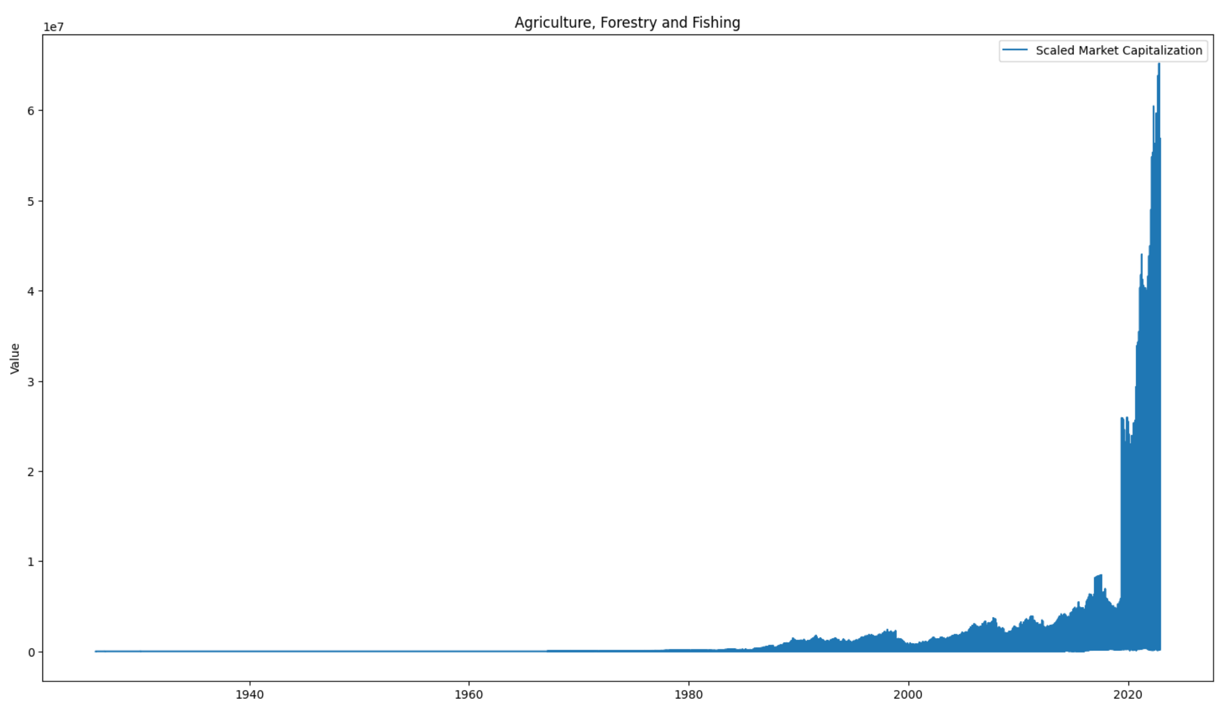
- Overall, market capitalization follows a similar distrubtion as the shares outstanding. Comparing these two indicators allows us to see the way market cap is affected by shares outstadning. When there are discrepancies, i.e., the graphs don't match up entirely, this can be because market cap does not account for factors such as a company's debt, future growth potential, aqcuistions or the health of its industry sector. These factors play a role in the number of share outstanding.
- In the 90's and early 2000's we see a large increase in shares outstanding/ market cap. This is due to the "dot com bubble" when shares were able to be bought and sold via

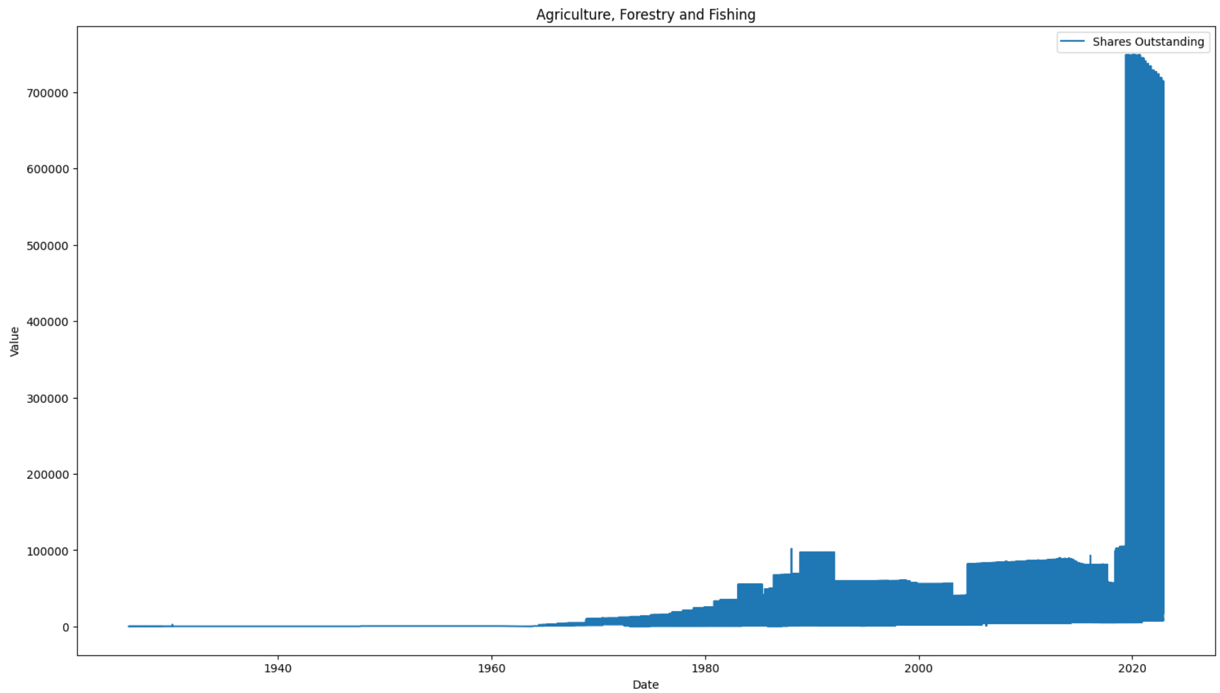
the internet.

- Of course the shares outstanding/ market cap drop during the 2008 finanical crisis.

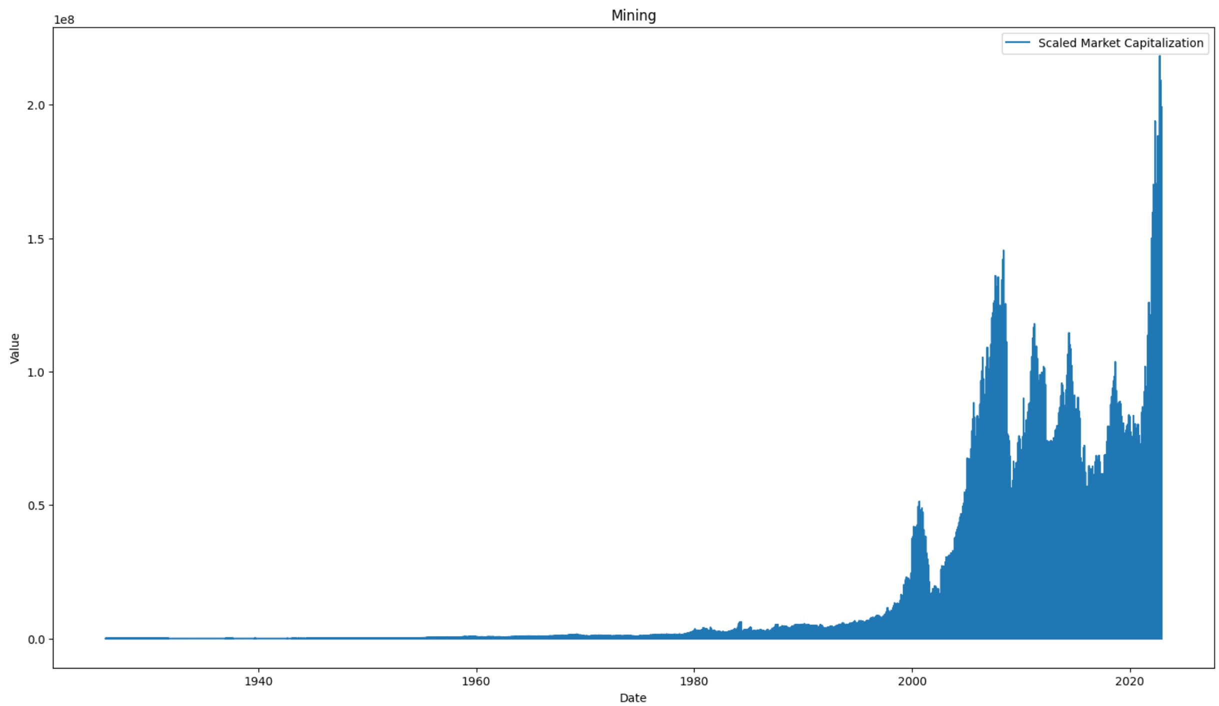
1.4 Plot by Industry (HSICCD) over time

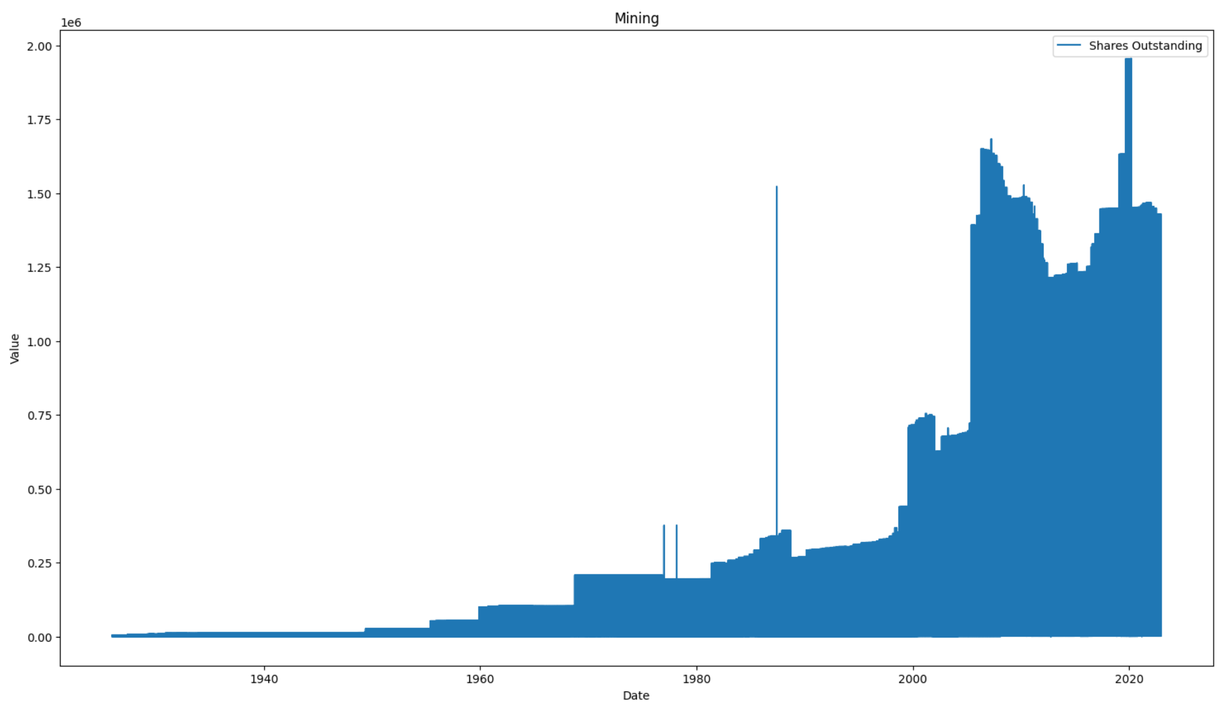
In [ ]:
```
start_date = '1925-12-01'
end_date = '2022-12-01'
industry_df = merge_df[(merge_df['DATE'] >= start_date) & (merge_df['DATE'] <= end_
initial_data_type = industry_df['HSICCD'].dtype
industry_df["HSICCD"] = industry_df["HSICCD"].replace('NaN', 0)
industry_df["HSICCD"] = pd.to_numeric(industry_df['HSICCD'], errors = 'coerce')
data_type = industry_df['HSICCD'].dtype

ag_industry_df = industry_df[(industry_df['HSICCD'] >= 1) & (industry_df["HSICCD"]

ag_industry_df.set_index('DATE', inplace = True)

plt.figure(figsize=(18, 10))
plt.plot(ag_industry_df.index, ag_industry_df["ScaledMarketCap"], label = 'Scaled M
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Agriculture, Forestry and Fishing")
plt.legend()
plt.show()

plt.figure(figsize=(18, 10))
plt.plot(ag_industry_df.index, ag_industry_df["SHROUT"], label = 'Shares Outstandin
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Agriculture, Forestry and Fishing")
plt.legend()
plt.show()
```





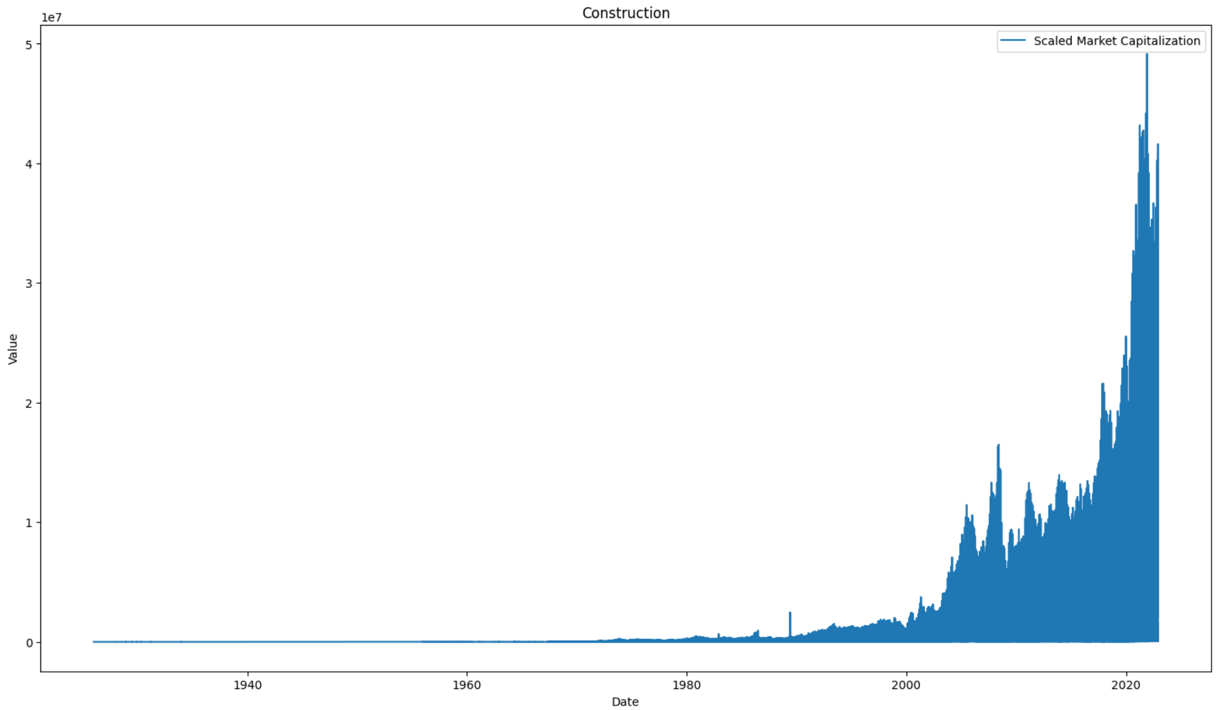
In [ ]:
```
mining_industry_df = industry_df[(industry_df['HSICCD'] >= 1000) & (industry_df["HS

mining_industry_df.set_index('DATE', inplace = True)

plt.figure(figsize=(18, 10))
plt.plot(mining_industry_df.index, mining_industry_df["ScaledMarketCap"], label = '
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Mining")
plt.legend()
plt.show()

plt.figure(figsize=(18, 10))
plt.plot(mining_industry_df.index, mining_industry_df["SHROUT"], label = 'Shares Ou
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Mining")
plt.legend()
plt.show()
```

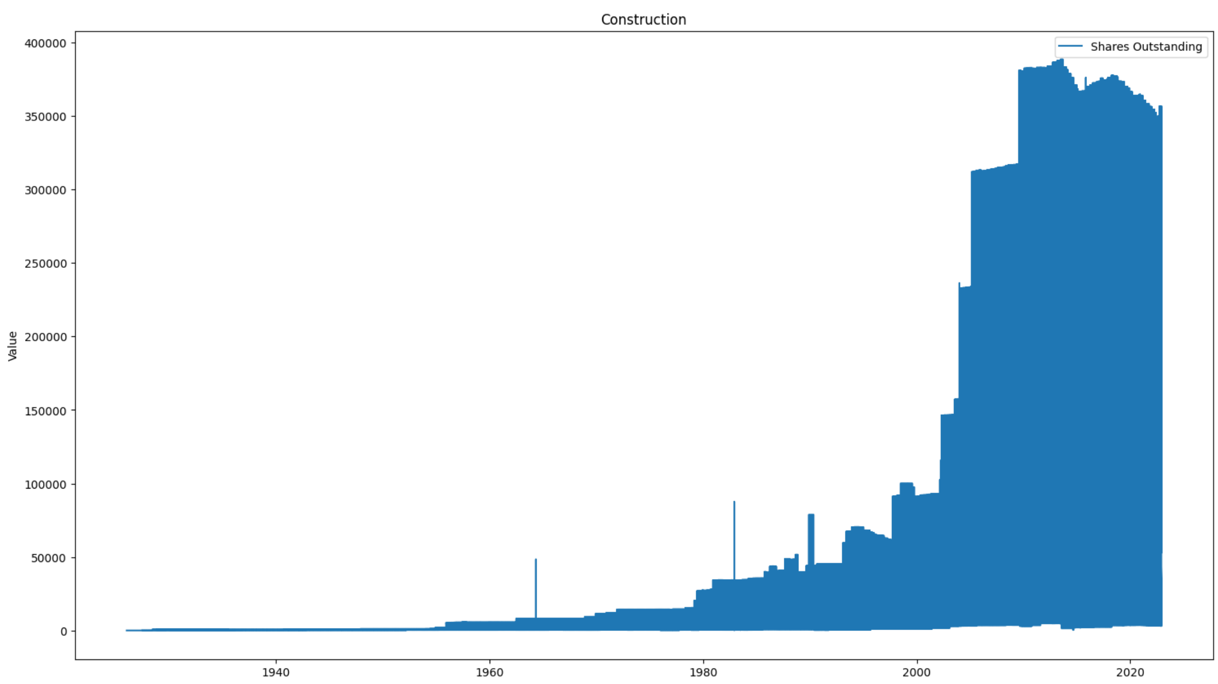




In [ ]:
```
construct_industry_df = industry_df[(industry_df['HSICCD'] >= 1500) & (industry_df[

construct_industry_df.set_index('DATE', inplace = True)

plt.figure(figsize=(18, 10))
plt.plot(construct_industry_df.index, construct_industry_df["ScaledMarketCap"], lab
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Construction")
plt.legend()
plt.show()

plt.figure(figsize=(18, 10))
```

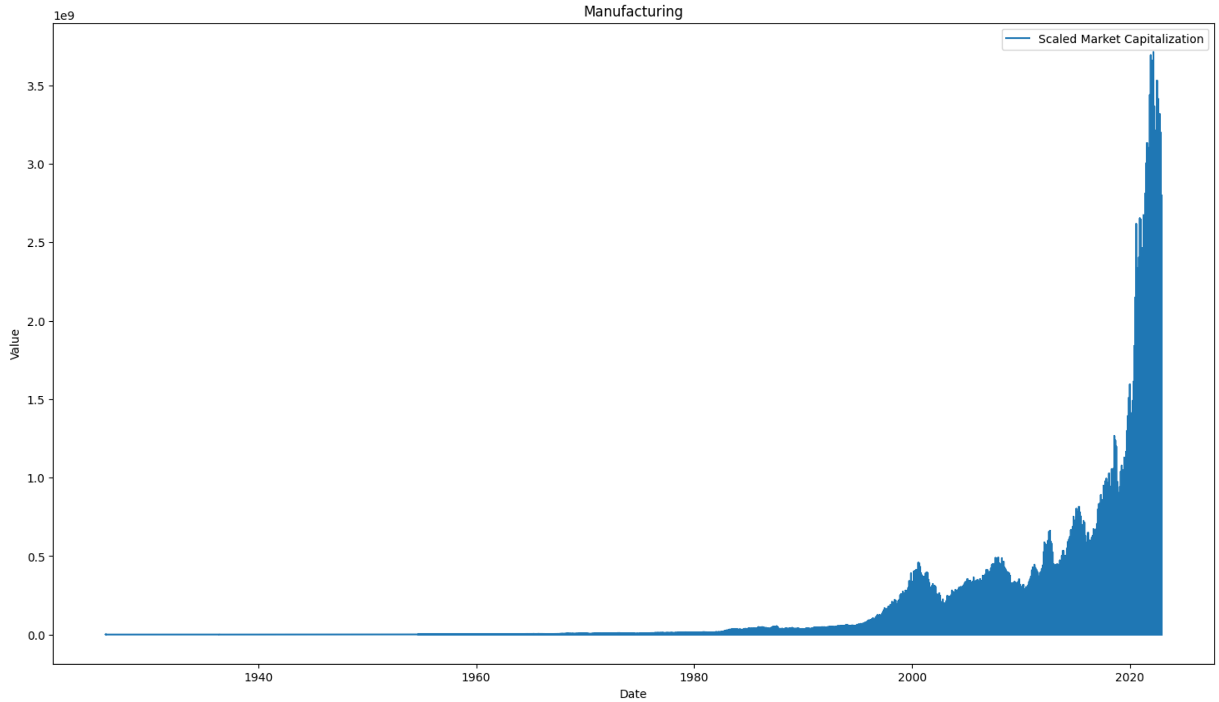
```
plt.plot(construct_industry_df.index, construct_industry_df["SHROUT"], label = 'Sha
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Construction")
plt.legend()
plt.show()
```

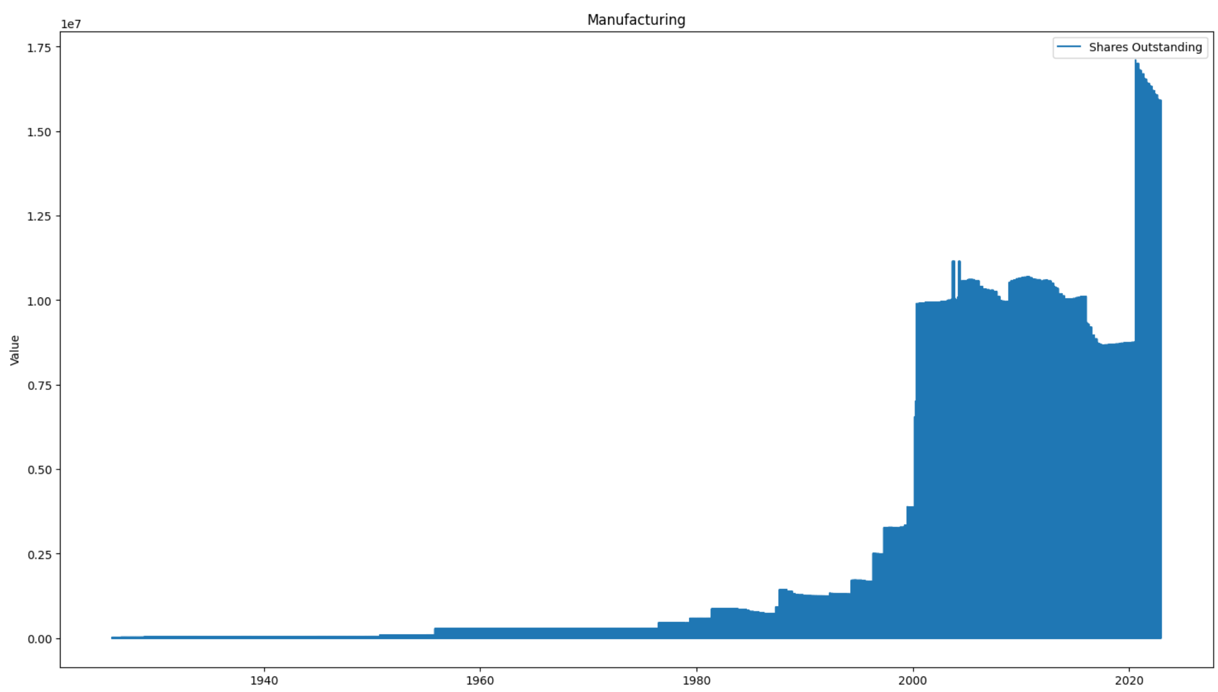




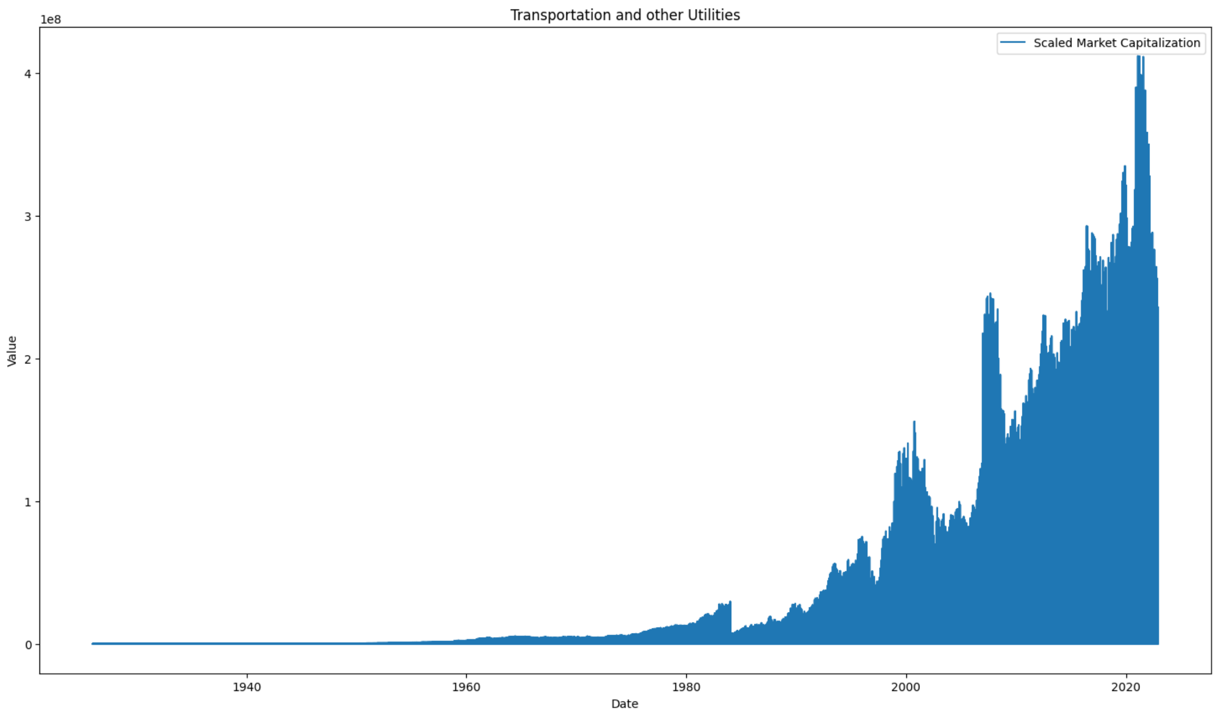
In [ ]:
```
man_industry_df = industry_df[(industry_df['HSICCD'] >= 2000) & (industry_df["HSICC

man_industry_df.set_index('DATE', inplace = True)

plt.figure(figsize=(18, 10))
plt.plot(man_industry_df.index, man_industry_df["ScaledMarketCap"], label = 'Scaled
```

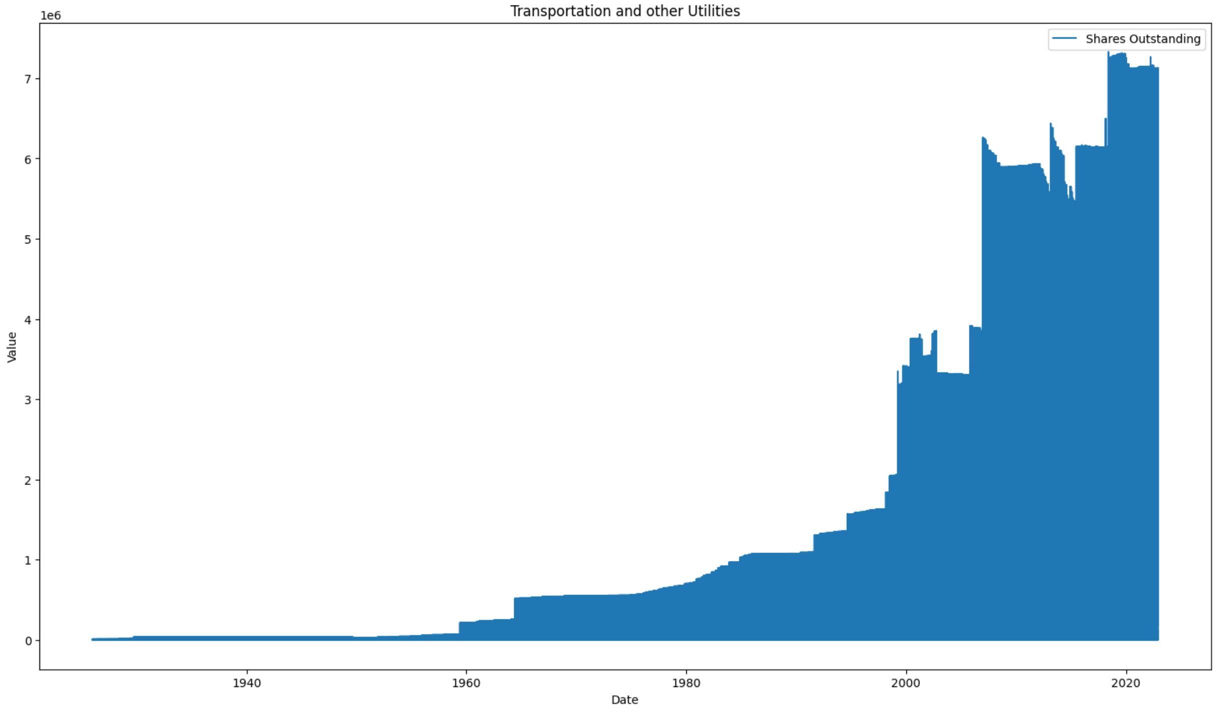
```
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Manufacturing")
plt.legend()
plt.show()

plt.figure(figsize=(18, 10))
plt.plot(man_industry_df.index, man_industry_df["SHROUT"], label = 'Shares Outstand
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Manufacturing")
plt.legend()
plt.show()
```





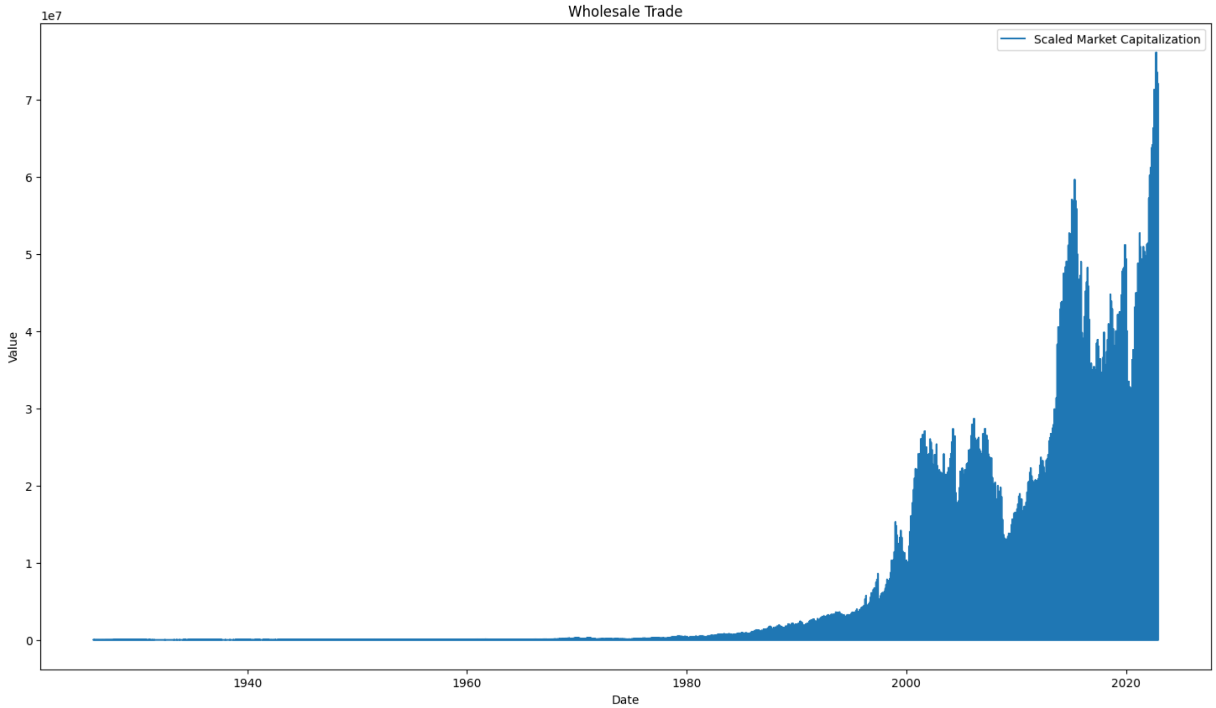
In [ ]:
```
trans_industry_df = industry_df[(industry_df['HSICCD'] >= 4000) & (industry_df["HSI

trans_industry_df.set_index('DATE', inplace = True)

plt.figure(figsize=(18, 10))
plt.plot(trans_industry_df.index, trans_industry_df["ScaledMarketCap"], label = 'Sc
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Transportation and other Utilities")
plt.legend()
plt.show()

plt.figure(figsize=(18, 10))
plt.plot(trans_industry_df.index, trans_industry_df["SHROUT"], label = 'Shares Outs
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Transportation and other Utilities")
plt.legend()
plt.show()
```

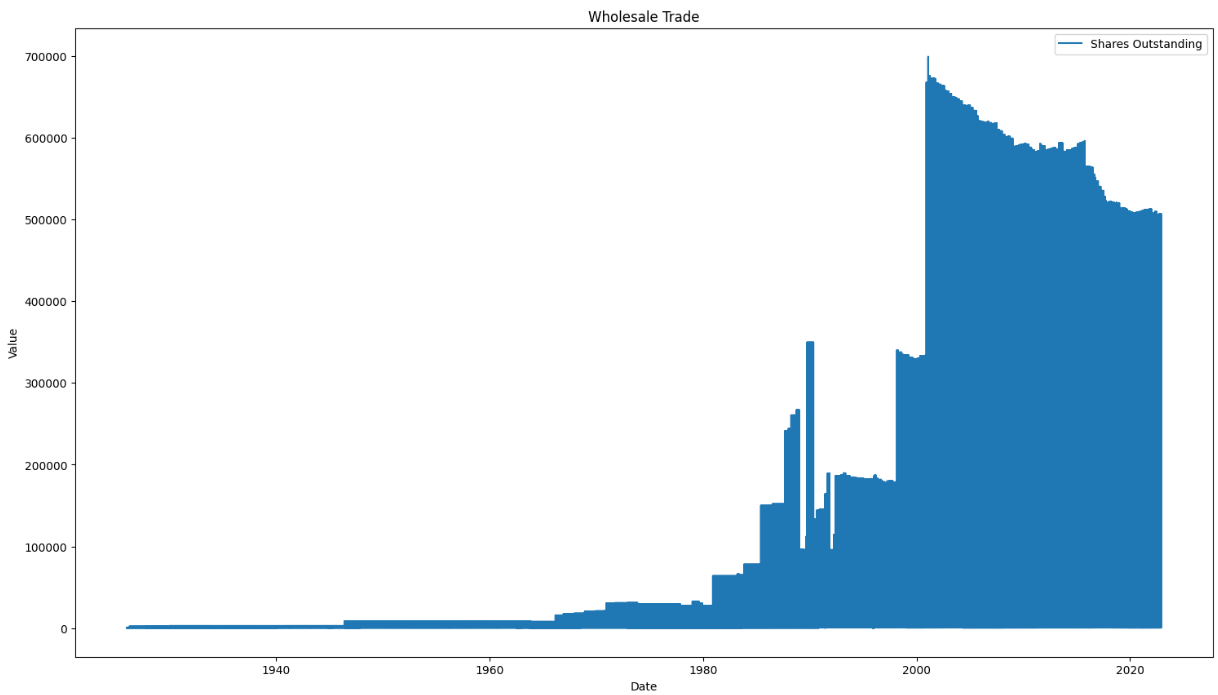




In [ ]:

```
whole_industry_df = industry_df[(industry_df['HSICCD'] >= 5000) & (industry_df["HSI

whole_industry_df.set_index('DATE', inplace = True)

plt.figure(figsize=(18, 10))
plt.plot(whole_industry_df.index, whole_industry_df["ScaledMarketCap"], label = 'Sc
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Wholesale Trade")
plt.legend()
plt.show()

plt.figure(figsize=(18, 10))
plt.plot(whole_industry_df.index, whole_industry_df["SHROUT"], label = 'Shares Outs
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Wholesale Trade")
plt.legend()
plt.show()
```

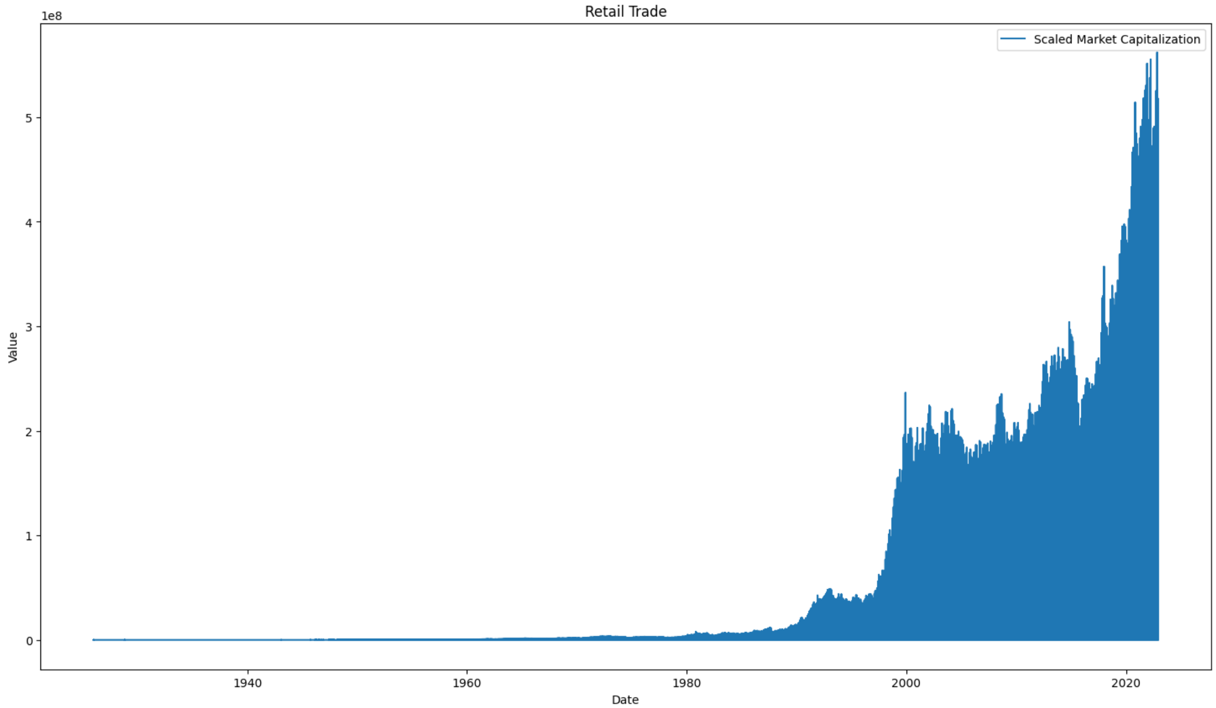


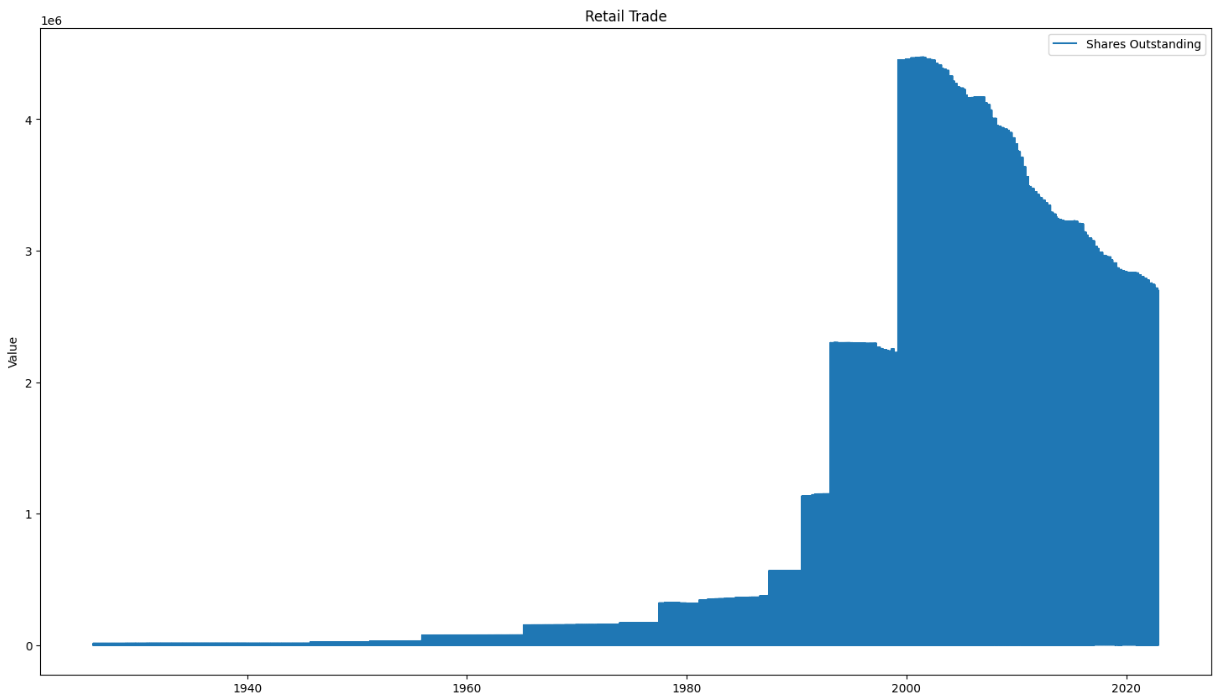


In [ ]:

```
re_industry_df = industry_df[(industry_df['HSICCD'] >= 5200) & (industry_df["HSICCD

re_industry_df.set_index('DATE', inplace = True)

plt.figure(figsize=(18, 10))
plt.plot(re_industry_df.index, re_industry_df["ScaledMarketCap"], label = 'Scaled M
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Retail Trade")
plt.legend()
plt.show()

plt.figure(figsize=(18, 10))
```

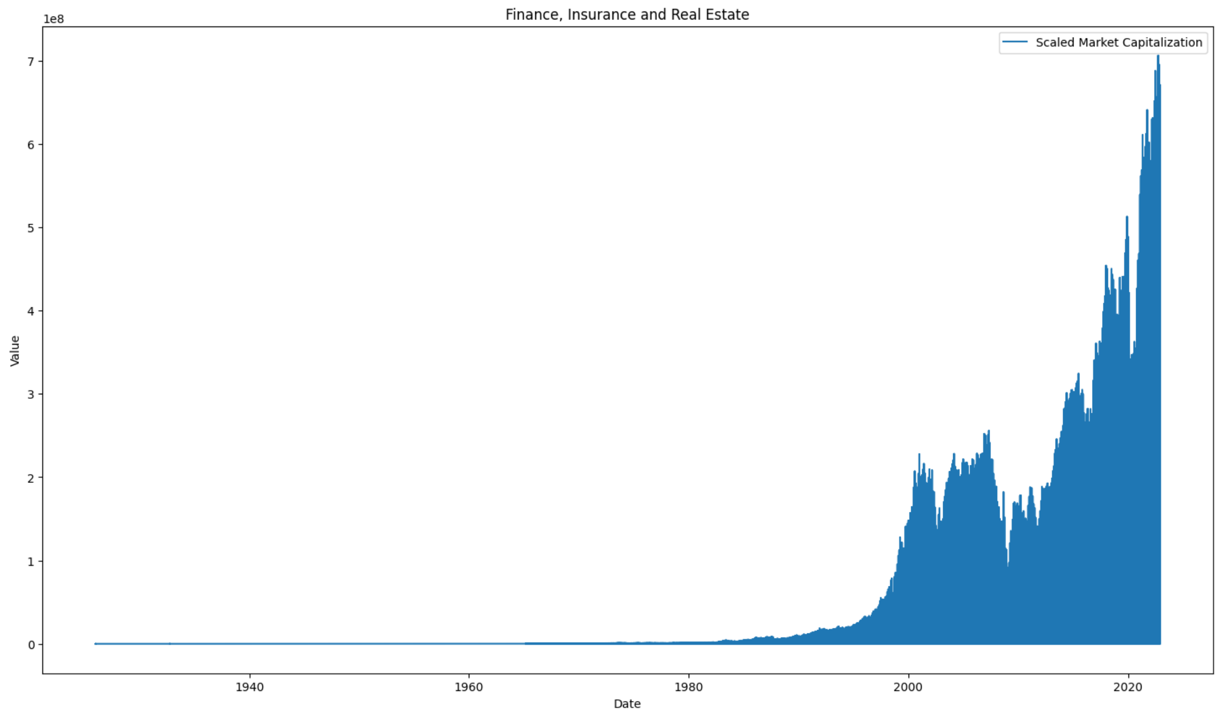
```
plt.plot(re_industry_df.index, re_industry_df["SHROUT"], label = 'Shares Outstandin
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Retail Trade")
plt.legend()
plt.show()
```

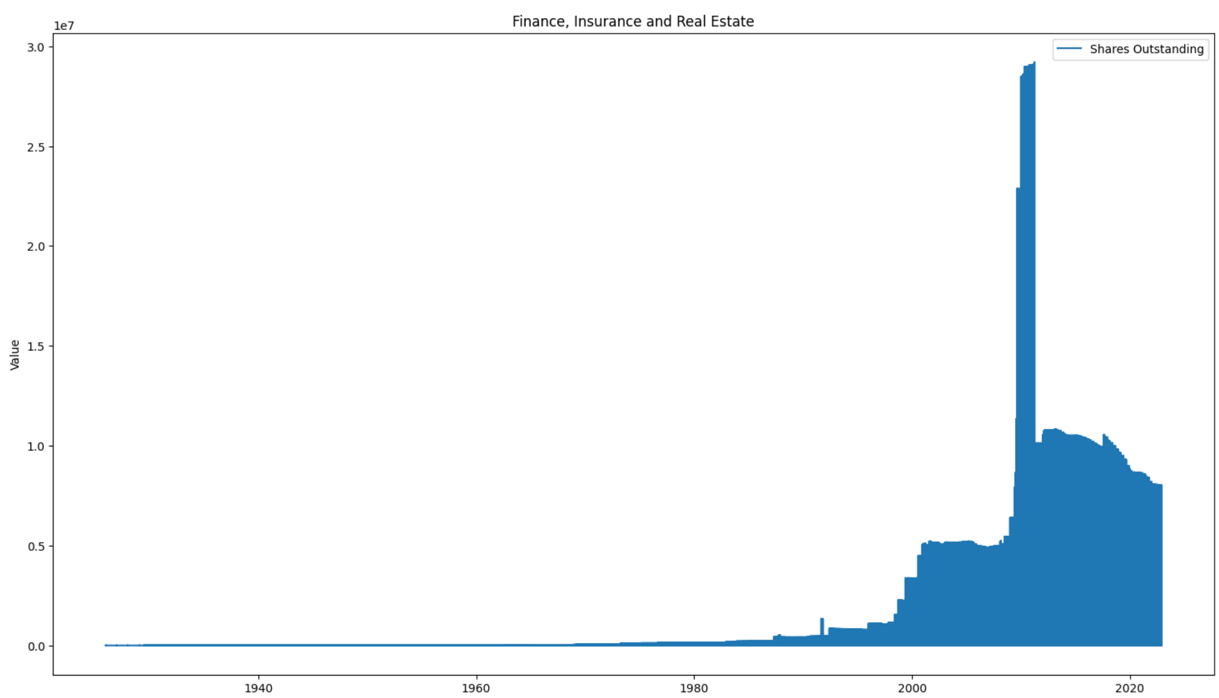




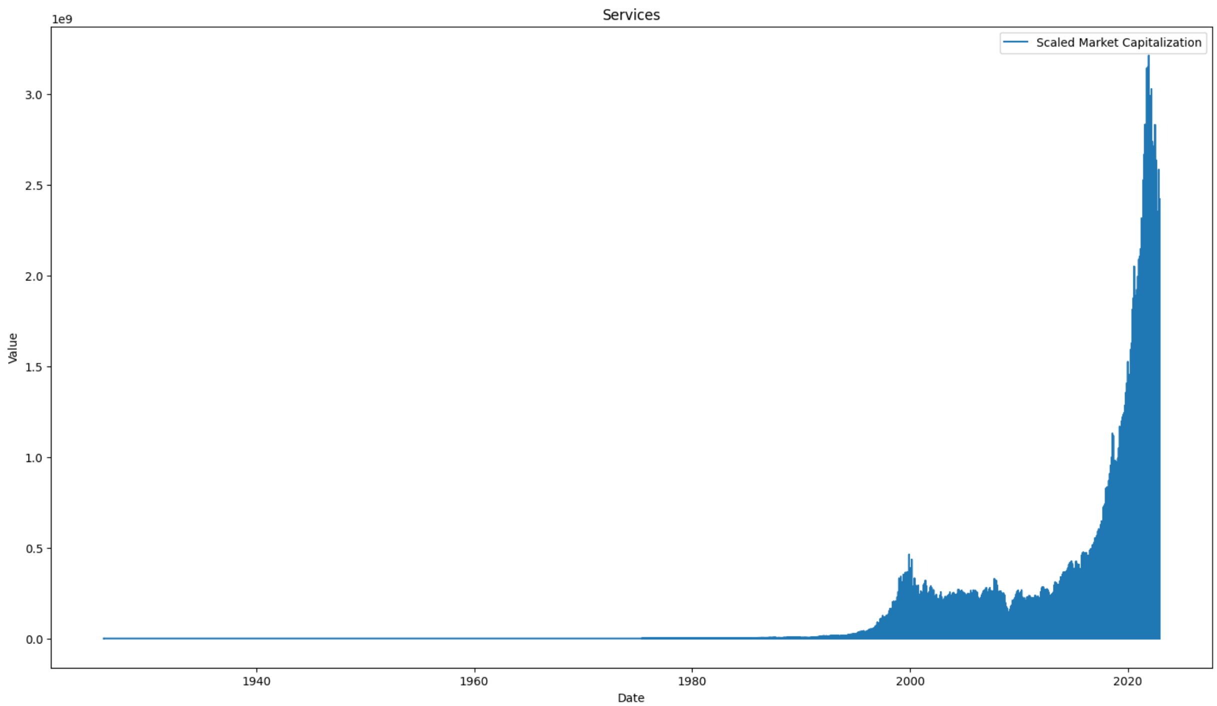
In [ ]:
```
fin_industry_df = industry_df[(industry_df['HSICCD'] >= 6000) & (industry_df["HSICC

fin_industry_df.set_index('DATE', inplace = True)

plt.figure(figsize=(18, 10))
plt.plot(fin_industry_df.index, fin_industry_df["ScaledMarketCap"], label = 'Scaled
```

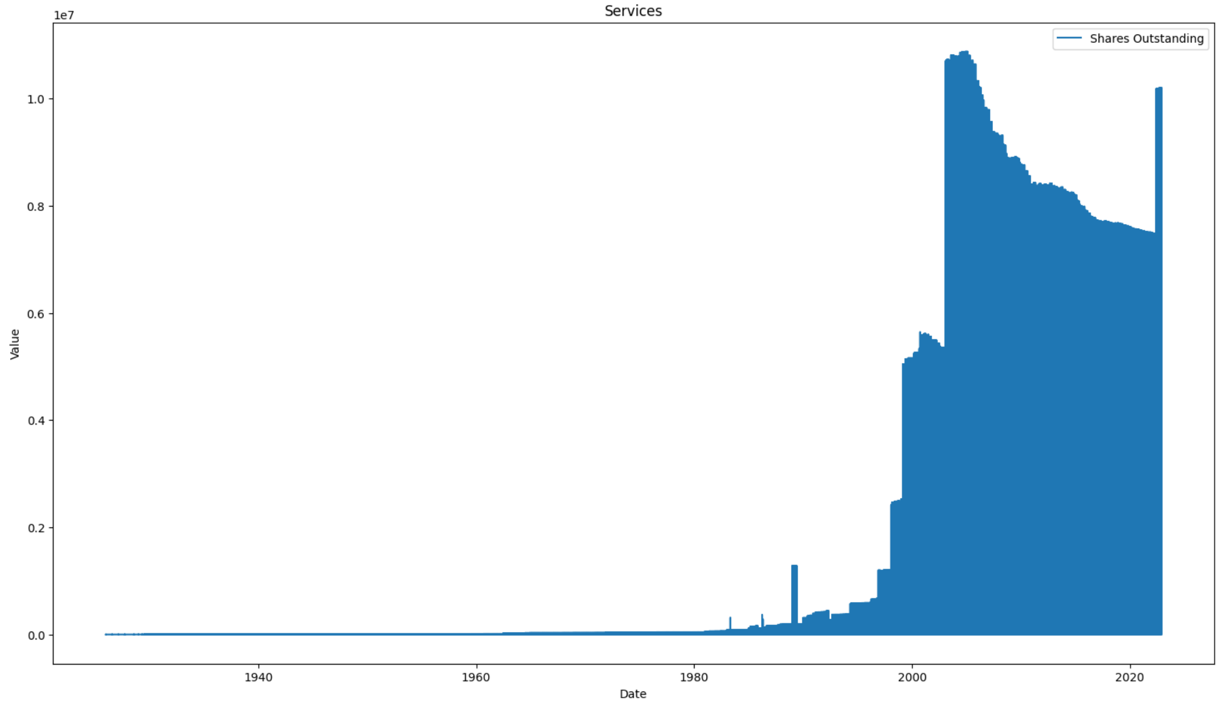
```
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Finance, Insurance and Real Estate")
plt.legend()
plt.show()

plt.figure(figsize=(18, 10))
plt.plot(fin_industry_df.index, fin_industry_df["SHROUT"], label = 'Shares Outstand
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Finance, Insurance and Real Estate")
plt.legend()
plt.show()
```



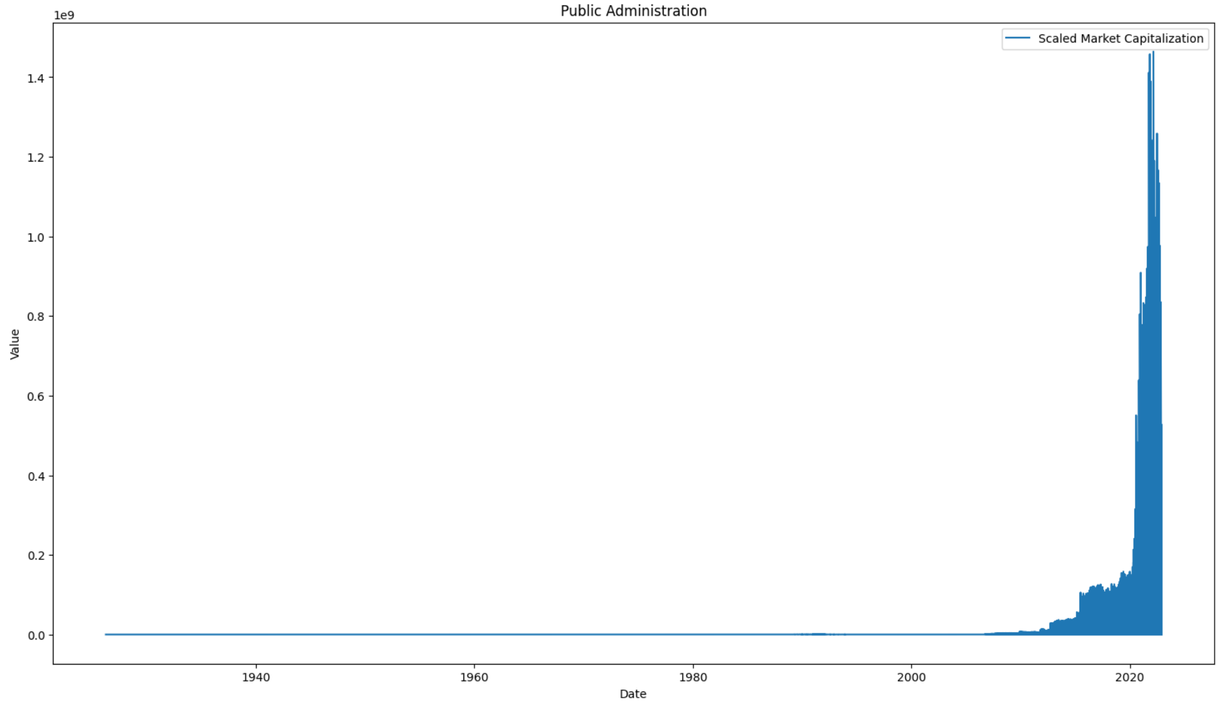


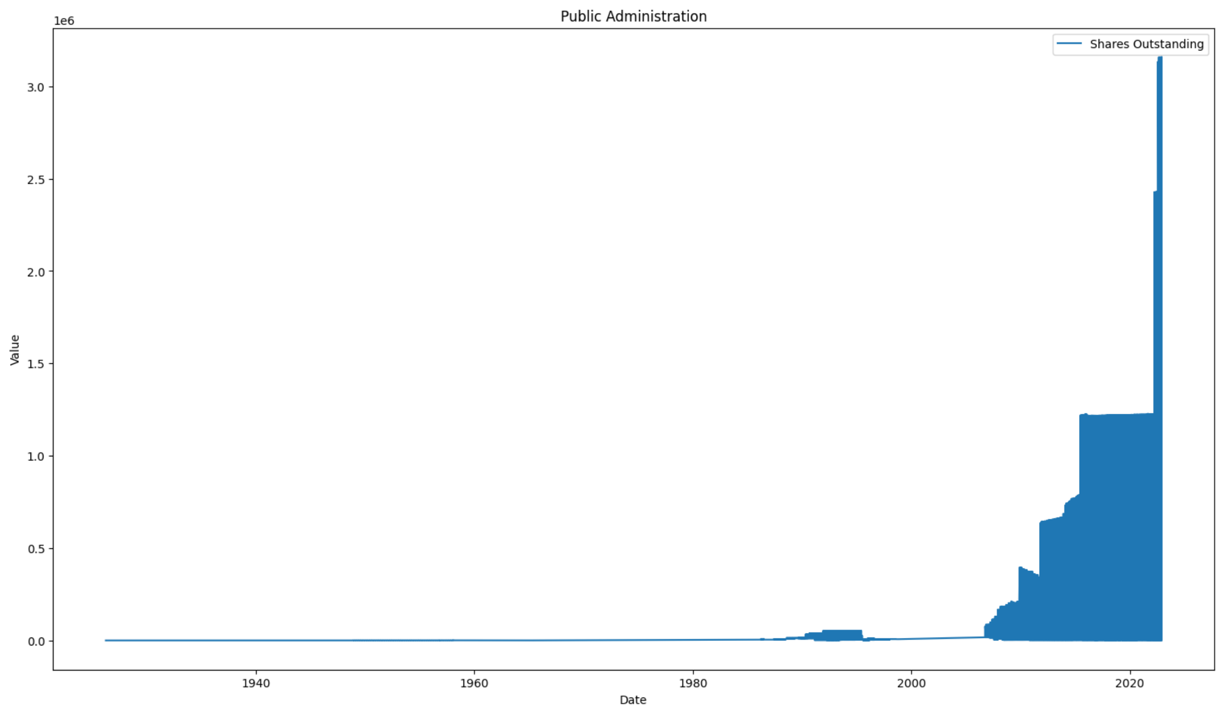
In [ ]:
```
ser_industry_df = industry_df[(industry_df['HSICCD'] >= 7000) & (industry_df["HSICC

ser_industry_df.set_index('DATE', inplace = True)

plt.figure(figsize=(18, 10))
plt.plot(ser_industry_df.index, ser_industry_df["ScaledMarketCap"], label = 'Scaled
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Services")
plt.legend()
plt.show()

plt.figure(figsize=(18, 10))
plt.plot(ser_industry_df.index, ser_industry_df["SHROUT"], label = 'Shares Outstand
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Services")
plt.legend()
plt.show()
```





In [ ]:
```
pub_industry_df = industry_df[(industry_df['HSICCD'] >= 9000) & (industry_df["HSICC

pub_industry_df.set_index('DATE', inplace = True)



plt.figure(figsize=(18, 10))
plt.plot(pub_industry_df.index, pub_industry_df["ScaledMarketCap"], label = 'Scaled
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Public Administration")
plt.legend()
plt.show()

plt.figure(figsize=(18, 10))
plt.plot(pub_industry_df.index, pub_industry_df["SHROUT"], label = 'Shares Outstand
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Public Administration")
plt.legend()
plt.show()
```





- Across industries, market cap/shares outstanding follow a similar growth pattern.
- During the finanical crisis of 2008, the finance industry had an inverse relationship between market cap/ shares outstanding. This tells us that there were many shares outstanding, but they were not being sold for a high price.

1.5 Compute excess return and log of return

In [ ]:
```
import urllib.request
import zipfile
ff_url = "https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/ftp/F-F_Research_
```

```
# Download the file and save it
# We will name it fama_french.zip file
urllib.request.urlretrieve(ff_url,'fama_french.zip')
zip_file = zipfile.ZipFile('fama_french.zip', 'r')
# Next we extact the file data
# We will call it ff_factors.csv
zip_file.extractall()
# Make sure you close the file after extraction
zip_file.close()
import pandas as pd
ff_factors = pd.read_csv('F-F_Research_Data_Factors.csv', skiprows = 3, nrows = 111
ff_factors.index = pd.to_datetime(ff_factors.index, format= '%Y%m')
ff_factors.index = ff_factors.index + pd.offsets.MonthEnd()
ff_factors = ff_factors.apply(lambda x: x/ 100)
```

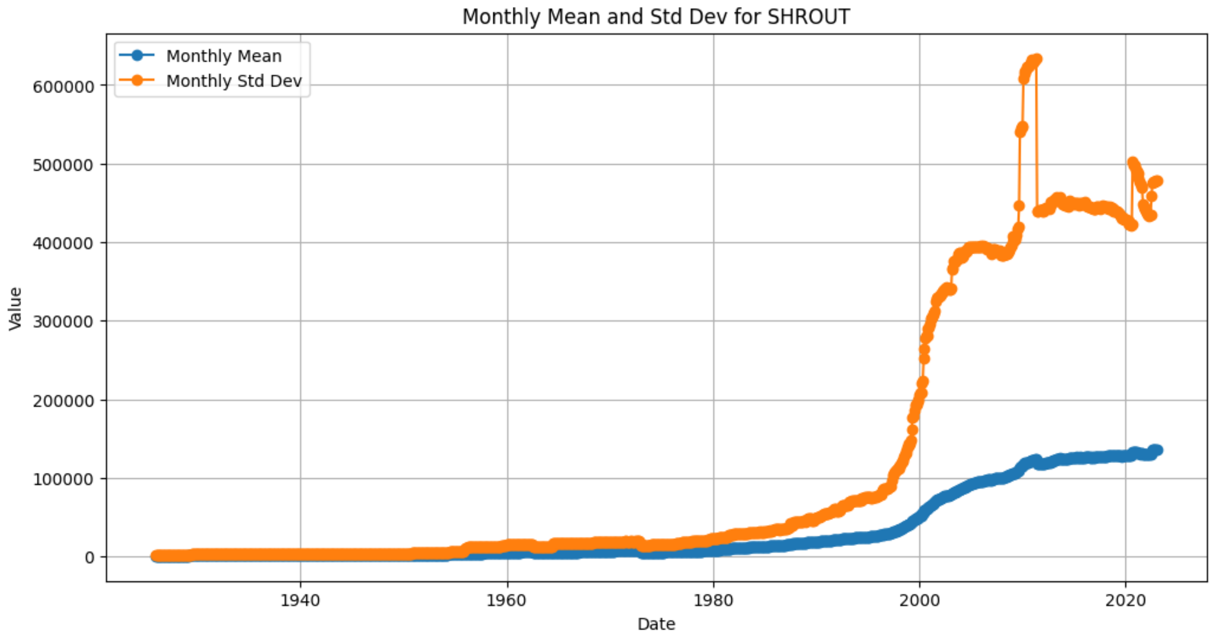
In [ ]:
```
filtered_df.index = pd.to_datetime(filtered_df['date'])

combined_df = filtered_df.merge(ff_factors, left_index=True, right_index=True, how=
combined_df = combined_df.dropna()

# Convert to float and coerce errors to NaN
combined_df['RET'] = pd.to_numeric(combined_df['RET'], errors='coerce')

# Drop rows containing NaN values in the 'RET' column
combined_df = combined_df.dropna(subset=['RET'])

combined_df['Excess_Return'] = combined_df['RET'] - combined_df['RF']
print(combined_df)
```

```
             PERMNO       date  SHRCD  EXCHCD TICKER  
date                                                    
1945-10-31  22402.0 1945-10-31   10.0     1.0     ST  \
1945-11-30  22402.0 1945-11-30   10.0     1.0     ST   
1945-12-31  22402.0 1945-12-31   10.0     1.0     ST   
1945-12-31  22517.0 1945-12-31   11.0     1.0    PPL   
1946-01-31  22402.0 1946-01-31   10.0     1.0     ST   
...             ...        ...    ...     ...    ...   
2019-04-30  93371.0 2019-04-30   11.0     2.0   CRMD   
2019-04-30  93372.0 2019-04-30   11.0     1.0   PLOW   
2019-04-30  93373.0 2019-04-30   11.0     1.0   EXPR   
2019-04-30  93374.0 2019-04-30   11.0     1.0    FAF   
2019-04-30  93384.0 2019-04-30   11.0     1.0   RRTS   

                                  COMNAM   PERMCO   ISSUNO  HEXCD  HSICCD  
date                                                                        
1945-10-31  CHICAGO MILWAUKEE ST PAUL & PAC  23142.0      0.0    1.0  4011.0  \
1945-11-30  CHICAGO MILWAUKEE ST PAUL & PAC  23142.0      0.0    1.0  4011.0   
1945-12-31  CHICAGO MILWAUKEE ST PAUL & PAC  23142.0      0.0    1.0  4011.0   
1945-12-31    PENNSYLVANIA POWER & LIGHT CO  21376.0      0.0    1.0  4813.0   
1946-01-31  CHICAGO MILWAUKEE ST PAUL & PAC  23142.0      0.0    1.0  4011.0   
...                                      ...      ...      ...    ...     ...   
2019-04-30                     CORMEDIX INC  53357.0  66161.0    3.0    2834   
2019-04-30             DOUGLAS DYNAMICS INC  53405.0      0.0    1.0    3531   
2019-04-30                      EXPRESS INC  53406.0      0.0    1.0    5621   
2019-04-30     FIRST AMERICAN FINL CORP NEW  53407.0      0.0    1.0    6361   
2019-04-30     ROADRUNNER TRANS SYSTEMS INC  53413.0      0.0    1.0    4731   

            ...    ALTPRCDT       RETX    vwretd  market_cap  MonthYear  
date        ...                                                           
1945-10-31  ...  1945-10-31          C  0.038749    53605.75    10-1945  \
1945-11-30  ...  1945-11-30   0.148515  0.054859    61567.00    11-1945   
1945-12-31  ...  1945-12-31   0.077586  0.013003    66343.75    12-1945   
1945-12-31  ...  1945-12-31          C  0.013003    60649.25    12-1945   
1946-01-31  ...  1946-01-31   0.192000  0.063402    79081.75     1-1946   
...         ...         ...        ...       ...         ...        ...   
2019-04-30  ...  2019-04-30  -0.142857  0.037893   192804.30     4-2019   
2019-04-30  ...  2019-04-30  -0.008143  0.037893   860739.20     4-2019   
2019-04-30  ...  2019-04-30  -0.140187  0.037893   244745.76     4-2019   
2019-04-30  ...  2019-04-30   0.107961  0.037893  6396368.94     4-2019   
2019-04-30  ...  2019-04-30   0.058096  0.037893   417391.59     4-2019   

            Mkt-RF     SMB     HML      RF  Excess_Return  
date                                                        
1945-10-31  0.0389  0.0238  0.0213  0.0003       0.081093  
1945-11-30  0.0539  0.0431  0.0396  0.0002       0.148315  
1945-12-31  0.0120  0.0210 -0.0228  0.0003       0.077286  
1945-12-31  0.0120  0.0210 -0.0228  0.0003      -0.198753  
1946-01-31  0.0624  0.0391  0.0248  0.0003       0.191700  
...            ...     ...     ...     ...            ...  
2019-04-30  0.0397 -0.0174  0.0215  0.0021      -0.144957  
2019-04-30  0.0397 -0.0174  0.0215  0.0021      -0.010243  
2019-04-30  0.0397 -0.0174  0.0215  0.0021      -0.142287  
2019-04-30  0.0397 -0.0174  0.0215  0.0021       0.105861  
2019-04-30  0.0397 -0.0174  0.0215  0.0021       0.055996  
```

```
[2193149 rows x 33 columns]
```

1.6 Compute CRSP_MSF descriptive stats and plot monthly ompute the descriptive stats - N, mean, standard deviation, skewness, kurtosis along with the minimum value, maximum value, 1%, 5%, 25%, 50%, 75%, 95%, 99% percentiles. • Compute these descriptive statistics for the following time periods – for 1925–2022 time period – for 1963–2022 time period – Plot the mean and standard deviation at a monthly frequency.

In [ ]:
```
filtered_df['date'] = pd.to_datetime(filtered_df['date'])
start_date = pd.to_datetime('1925-01-01')
end_date = filtered_df['date'].max()

period1_df = filtered_df[(filtered_df['date'] >= start_date) & (filtered_df['date']

# Compute descriptive statistics and percentiles for Shares Outstandng
column_name = 'SHROUT'
stats = period1_df[column_name].describe()
percentiles = period1_df[column_name].quantile([0.01, 0.05, 0.25, 0.50, 0.75, 0.95,
skewness = period1_df[column_name].skew()
kurt = period1_df[column_name].kurtosis()

period1_df.set_index('date', inplace=True)
monthly_stats = period1_df[column_name].resample('M').agg(['mean', 'std'])

# Plot the mean and standard deviation at a monthly frequency
plt.figure(figsize=(12, 6))
plt.plot(monthly_stats.index, monthly_stats['mean'], label='Monthly Mean', marker='
plt.plot(monthly_stats.index, monthly_stats['std'], label='Monthly Std Dev', marker
plt.xlabel('Date')
plt.ylabel('Value')
plt.title(f'Monthly Mean and Std Dev for {column_name}')
plt.legend()
plt.grid()
plt.show()

# Print the descriptive statistics and percentiles
print("Descriptive Statistics:")
print(stats)
print("skew: " + str(skewness))
print("kurtosis: " + str(kurt))
print("\nPercentiles:")
print(percentiles)
```

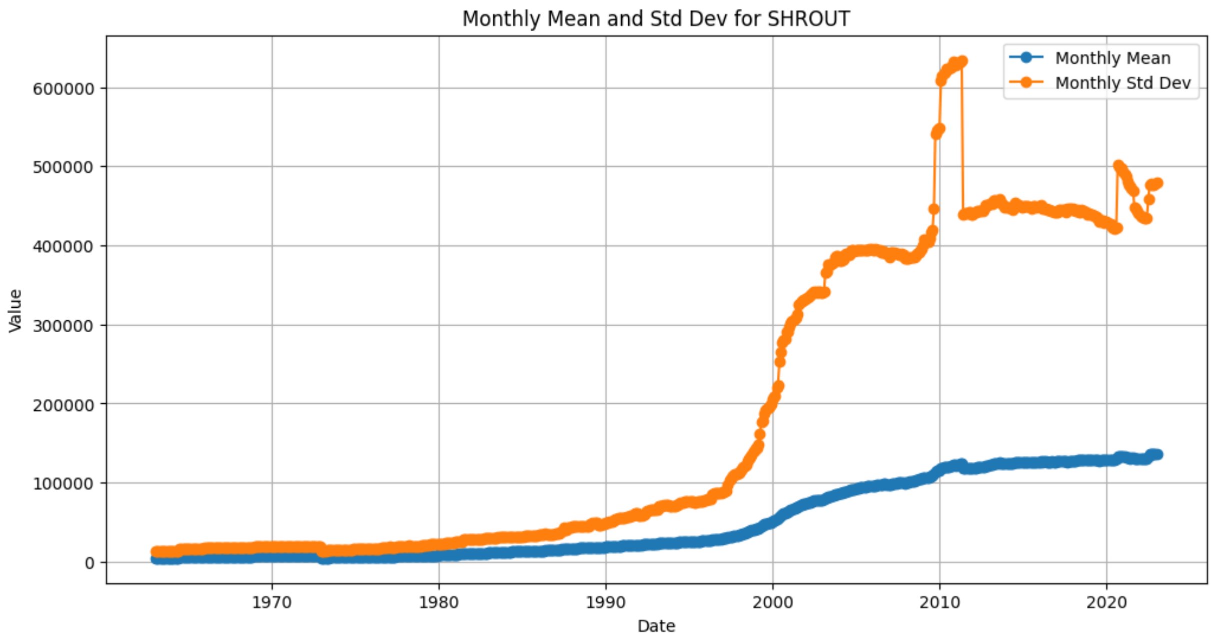



```
Descriptive Statistics:
count    3.794913e+06
mean     4.423087e+04
std      2.494665e+05
min      0.000000e+00
25%      2.288000e+03
50%      7.335000e+03
75%      2.558900e+04
max      2.920640e+07
Name: SHROUT, dtype: float64
skew: 28.40755764456632
kurtosis: 1529.0495524900014

Percentiles:
0.01       225.0
0.05       560.0
0.25      2288.0
0.50      7335.0
0.75     25589.0
0.95    145054.0
0.99    594402.0
Name: SHROUT, dtype: float64
```

In [ ]:

```
start_date = pd.to_datetime('1963-01-01')
end_date = filtered_df['date'].max()

period2_df = filtered_df[(filtered_df['date'] >= start_date) & (filtered_df['date']

# Compute descriptive statistics and percentiles for Shares Outstandng
column_name = 'SHROUT'
stats = period2_df[column_name].describe()
percentiles = period2_df[column_name].quantile([0.01, 0.05, 0.25, 0.50, 0.75, 0.95,
skewness = period2_df[column_name].skew()
kurt = period2_df[column_name].kurtosis()

period2_df.set_index('date', inplace=True)
monthly_stats = period2_df[column_name].resample('M').agg(['mean', 'std'])
```

```
# Plot the mean and standard deviation at a monthly frequency
plt.figure(figsize=(12, 6))
plt.plot(monthly_stats.index, monthly_stats['mean'], label='Monthly Mean', marker='
plt.plot(monthly_stats.index, monthly_stats['std'], label='Monthly Std Dev', marker
plt.xlabel('Date')
plt.ylabel('Value')
plt.title(f'Monthly Mean and Std Dev for {column_name}')
plt.legend()
plt.grid()
plt.show()

# Print the descriptive statistics and percentiles
print("Descriptive Statistics:")
print(stats)
print("skew: " + str(skewness))
print("kurtosis: " + str(kurt))
print("\nPercentiles:")
print(percentiles)
```



```
Descriptive Statistics:
count    3.408906e+06
mean     4.894859e+04
std      2.627831e+05
min      0.000000e+00
25%      3.026000e+03
50%      9.026000e+03
75%      2.932000e+04
max      2.920640e+07
Name: SHROUT, dtype: float64
skew: 27.00269838391608
kurtosis: 1380.2810552438398

Percentiles:
0.01       469.0
0.05       876.0
0.25      3026.0
0.50      9026.0
0.75     29320.0
0.95    160120.0
0.99    661413.0
Name: SHROUT, dtype: float64
```

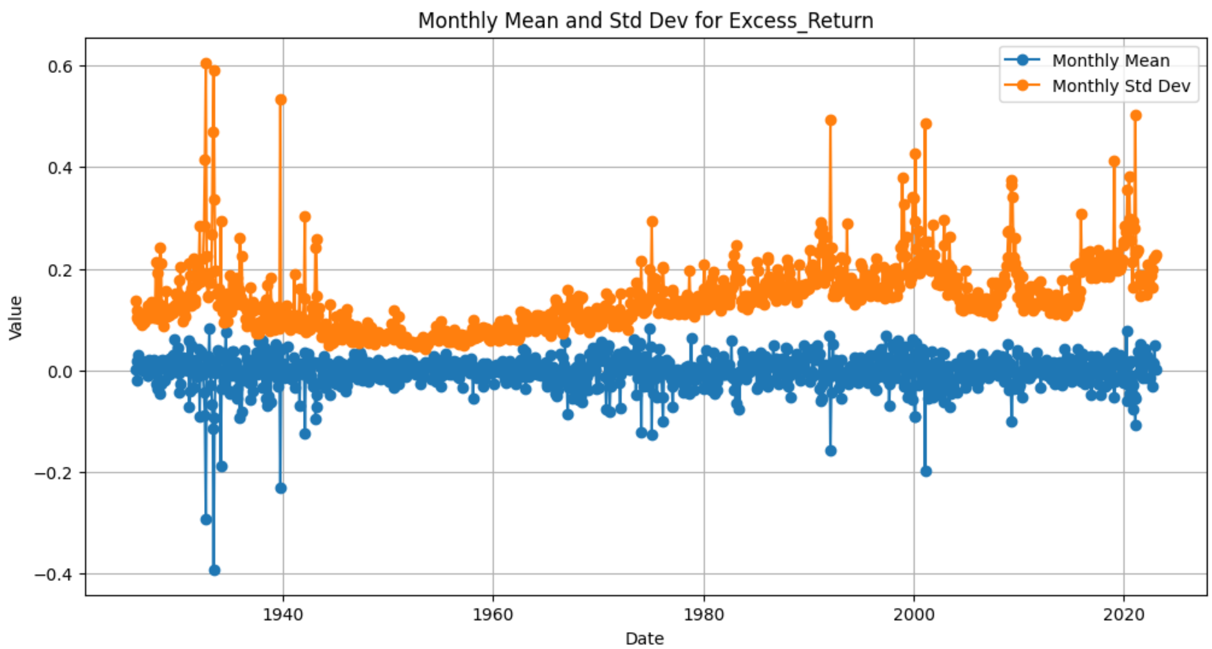
- Again, at the dawn of the internet, we see a large increase in the average shares outstanding.
- There are no large differences between these two time periods. The 2008 finanical crisis had a huge affect on the standard deviation.

1.7 Compute monthly VWRETD descriptive stats and plot monthly

In [ ]:
```
import matplotlib.dates as mdates

# Drop rows containing NaN values in the 'RET' column
filtered_df['Excess_Return'] = filtered_df['vwretd'] - filtered_df["RET"]
filtered_df['Log_Excess_Return'] = np.log(1 + filtered_df['Excess_Return'])

start_date = pd.to_datetime('1925-01-01')
end_date = filtered_df['date'].max()

period3_df = filtered_df[(filtered_df['date'] >= start_date) & (filtered_df['date']

# Compute descriptive statistics and percentiles for Shares Outstandng
column_name = 'Excess_Return'

monthly_stats = period3_df[column_name].resample('M').agg(['mean', 'std'])

# Plot the mean and standard deviation at a monthly frequency
plt.figure(figsize=(12, 6))
plt.plot(monthly_stats.index, monthly_stats['mean'], label='Monthly Mean', marker='
plt.plot(monthly_stats.index, monthly_stats['std'], label='Monthly Std Dev', marker
plt.xlabel('Date')
plt.ylabel('Value')
plt.title(f'Monthly Mean and Std Dev for {column_name}')
plt.legend()
```

```
plt.grid()
plt.show()




column_names = ['Excess_Return', 'Log_Excess_Return']
for i in column_names:
    stats = period3_df[i].describe()
    print("Descriptive Statistics for", i, "1925-2022", ":\n")
    print(stats)
    print("\n")


start_date = pd.to_datetime('1963-01-01')
end_date = filtered_df['date'].max()
period4_df = filtered_df[(filtered_df['date'] >= start_date) & (filtered_df['date']

column_names = ['Excess_Return', 'Log_Excess_Return']
for i in column_names:
    stats = period4_df[i].describe()
    print("Descriptive Statistics for", i, "1925-2022", ":\n")
    print(stats)
    print("\n")
```

```
C:\Users\mhlad\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfr
a8p0\LocalCache\local-packages\Python310\site-packages\pandas\core\arraylike.py:396:
RuntimeWarning: invalid value encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



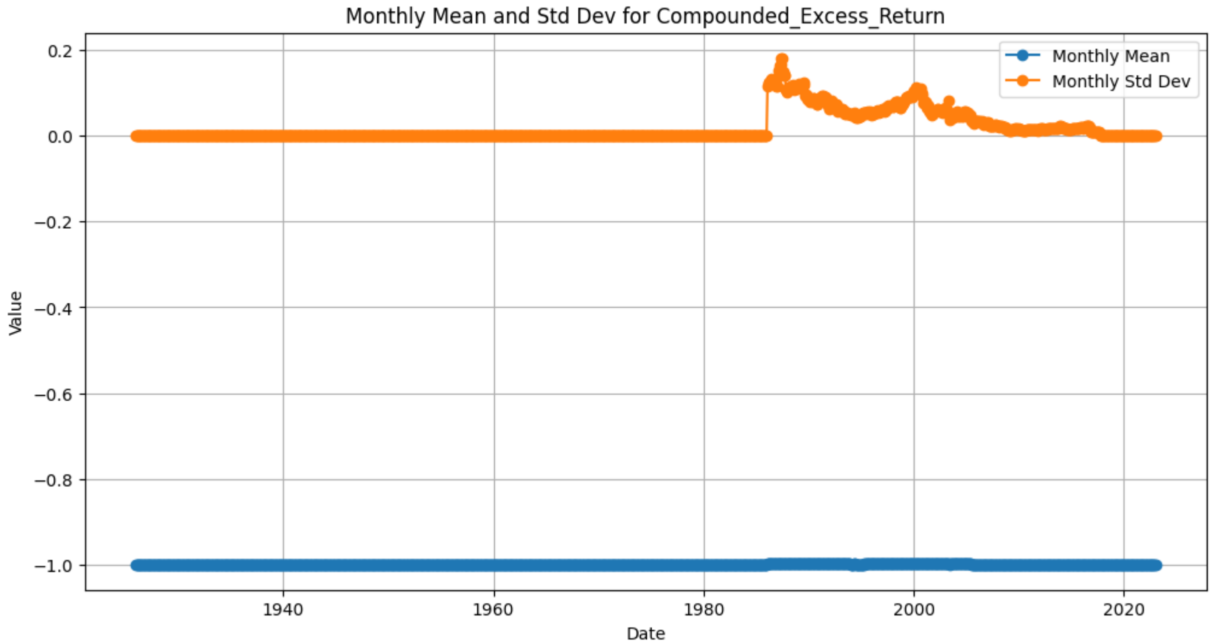
```
Descriptive Statistics for Excess_Return 1925-2022 :

count    3.794912e+06
mean    -1.630700e-03
std      1.754712e-01
min     -2.400116e+01
25%     -5.600400e-02
50%      8.520000e-03
75%      7.070200e-02
max      1.194143e+00
Name: Excess_Return, dtype: float64


Descriptive Statistics for Log_Excess_Return 1925-2022 :

count    3.786018e+06
mean    -1.185463e-02
std      1.927173e-01
min     -1.007784e+01
25%     -5.693973e-02
50%      8.728793e-03
75%      6.850198e-02
max      7.857915e-01
Name: Log_Excess_Return, dtype: float64


Descriptive Statistics for Excess_Return 1925-2022 :

count    3.408905e+06
mean    -1.481310e-03
std      1.806750e-01
min     -2.400116e+01
25%     -5.839600e-02
50%      8.868000e-03
75%      7.414400e-02
max      1.060084e+00
Name: Excess_Return, dtype: float64


Descriptive Statistics for Log_Excess_Return 1925-2022 :

count    3.400513e+06
mean    -1.226976e-02
std      1.980100e-01
min     -1.007784e+01
25%     -5.942469e-02
50%      9.116320e-03
75%      7.169721e-02
max      7.227468e-01
Name: Log_Excess_Return, dtype: float64
```

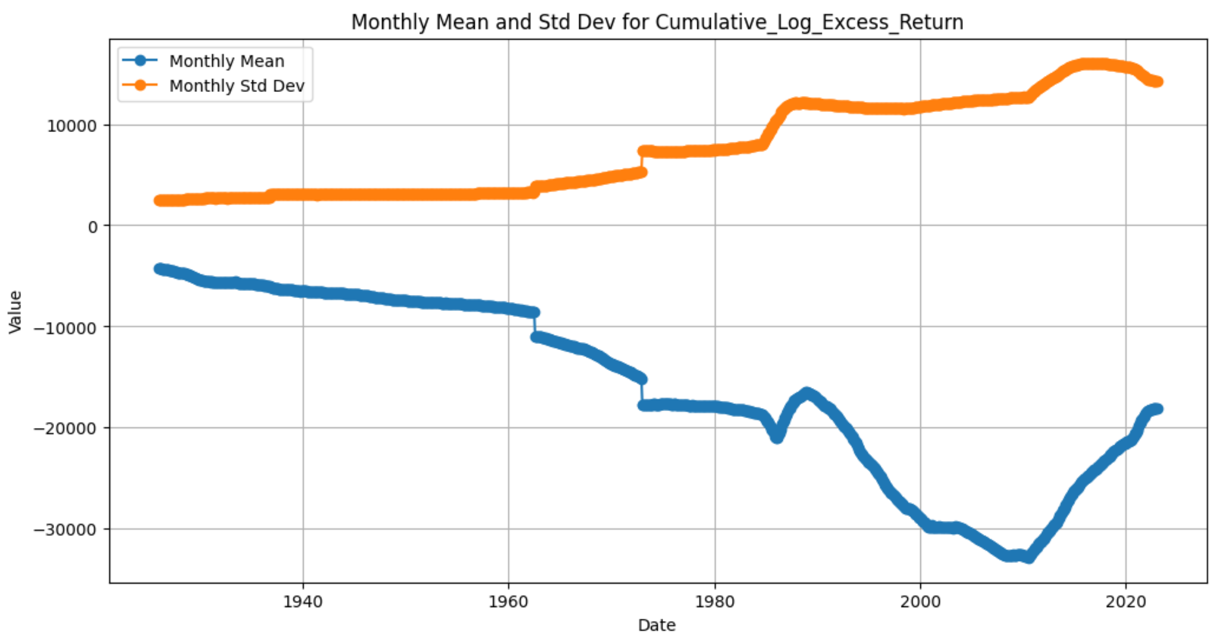
1.8 Plot the compounded excess returns and cumulative log excess returns

In [ ]:

```
filtered_df['Compounded_Excess_Return'] = (1 + filtered_df['Excess_Return']).cumpro
filtered_df['Cumulative_Log_Excess_Return'] = filtered_df['Log_Excess_Return'].cums




column_name = 'Compounded_Excess_Return'

monthly_stats = filtered_df[column_name].resample('M').agg(['mean', 'std'])

# Plot the mean and standard deviation at a monthly frequency
plt.figure(figsize=(12, 6))
plt.plot(monthly_stats.index, monthly_stats['mean'], label='Monthly Mean', marker='
plt.plot(monthly_stats.index, monthly_stats['std'], label='Monthly Std Dev', marker
plt.xlabel('Date')
plt.ylabel('Value')
plt.title(f'Monthly Mean and Std Dev for {column_name}')
plt.legend()
plt.grid()
plt.show()


column_name = 'Cumulative_Log_Excess_Return'

monthly_stats = filtered_df[column_name].resample('M').agg(['mean', 'std'])

# Plot the mean and standard deviation at a monthly frequency
plt.figure(figsize=(12, 6))
plt.plot(monthly_stats.index, monthly_stats['mean'], label='Monthly Mean', marker='
plt.plot(monthly_stats.index, monthly_stats['std'], label='Monthly Std Dev', marker
plt.xlabel('Date')
plt.ylabel('Value')
plt.title(f'Monthly Mean and Std Dev for {column_name}')
plt.legend()
plt.grid()
plt.show()


column_names = ['Compounded_Excess_Return', 'Cumulative_Log_Excess_Return']
for i in column_names:
    stats = filtered_df[i].describe()
    print("Descriptive Statistics for", i, "1925-2022", ":\n")
    print(stats)
    print("\n")


start_date = pd.to_datetime('1963-01-01')
end_date = filtered_df['date'].max()
period4_df = filtered_df[(filtered_df['date'] >= start_date) & (filtered_df['date']

column_names = ['Compounded_Excess_Return', 'Cumulative_Log_Excess_Return']
for i in column_names:
    stats = period4_df[i].describe()
    print("Descriptive Statistics for", i, "1925-2022", ":\n")
```

```
    print(stats)
    print("\n")
```





```
Descriptive Statistics for Compounded_Excess_Return 1925-2022 :

count    3.794912e+06
mean    -9.995143e-01
std      5.235812e-02
min     -2.950592e+00
25%     -1.000000e+00
50%     -1.000000e+00
75%     -1.000000e+00
max      1.097770e+01
Name: Compounded_Excess_Return, dtype: float64


Descriptive Statistics for Cumulative_Log_Excess_Return 1925-2022 :

count    3.786018e+06
mean    -2.098740e+04
std      1.301744e+04
min     -4.488249e+04
25%     -3.210556e+04
50%     -1.993579e+04
75%     -9.456406e+03
max      2.483047e+00
Name: Cumulative_Log_Excess_Return, dtype: float64


Descriptive Statistics for Compounded_Excess_Return 1925-2022 :

count    3.408905e+06
mean    -9.994589e-01
std      5.524270e-02
min     -2.950592e+00
25%     -1.000000e+00
50%     -1.000000e+00
75%     -1.000000e+00
max      1.097770e+01
Name: Compounded_Excess_Return, dtype: float64


Descriptive Statistics for Cumulative_Log_Excess_Return 1925-2022 :

count    3.400513e+06
mean    -2.256541e+04
std      1.276662e+04
min     -4.488249e+04
25%     -3.340144e+04
50%     -2.220452e+04
75%     -1.135217e+04
max      2.483047e+00
Name: Cumulative_Log_Excess_Return, dtype: float64
```

- We see spikes in the mean/standard deviation during the dotcom bubble, 2008 finanical crisis as well as during the 2020 pandemic.

- During the late 1930's we also see movement due to the aftermath of the great depression, it smooths out after due to various new economic policies, including banking reforms, securities regulations, and public works programs. The US also went of the gold standard in 1933, leading to changes in the value of the dollar.

2.1 Compute descriptive stats and range

In [ ]:
```
#Reading in MSF File

import pandas as pd
import numpy as np

path = r"C:\Users\mhlad\Downloads\dsf_2004_2022.csv"

sample = 0.01
data_sample1 = pd.read_csv(path, skiprows=lambda i: i>0 and np.random.rand() > samp
```

In [ ]:
```
import gc

path = r"C:\Users\mhlad\Downloads\dsf_2004_2022.csv"
chunk_size = 50000

reader = pd.read_csv(path, chunksize=chunk_size)

for chunk in reader:
    columns = ["BIDLO", "ASKHI", "PRC", "VOL", "RET", "BID", "ASK", "SHROUT", "CFAC

    for col in columns:
        chunk[col] = pd.to_numeric(chunk[col], errors='coerce')

    numeric_columns = chunk.select_dtypes(include=['float64', 'int64']).columns

    # Fill missing values using forward and backward fills for numeric columns only
    forward_filled = chunk[numeric_columns].fillna(method='ffill')
    backward_filled = chunk[numeric_columns].fillna(method='bfill')

    # Average the filled data
    averaged_fill = (forward_filled + backward_filled) / 2

    # Fill the original dataframe with averaged values for numeric columns only
    chunk[numeric_columns] = chunk[numeric_columns].fillna(averaged_fill)

    # Do any further processing with the chunk here if needed

    del chunk
    gc.collect()
```

```
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8,13,21) have mixed types. Specify dtype option on import or set low_memory=Fa
lse.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8,13,21) have mixed types. Specify dtype option on import or set low_memory=Fa
lse.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8,13,21) have mixed types. Specify dtype option on import or set low_memory=Fa
lse.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (7) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (7) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (7) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (7) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (7) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (7) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (7) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (7) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
```

```
umns (7) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (7) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (7) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (7) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (7) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (7) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (7) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8,13,21) have mixed types. Specify dtype option on import or set low_memory=Fa
lse.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
```

```
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8,13,21) have mixed types. Specify dtype option on import or set low_memory=Fa
lse.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8,13,21) have mixed types. Specify dtype option on import or set low_memory=Fa
lse.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
```

```
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
```

```
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8,13,21) have mixed types. Specify dtype option on import or set low_memory=Fa
lse.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
```

```
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
```

```
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
```

```
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8,13,21) have mixed types. Specify dtype option on import or set low_memory=Fa
lse.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8,13,21) have mixed types. Specify dtype option on import or set low_memory=Fa
lse.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8,13,21) have mixed types. Specify dtype option on import or set low_memory=Fa
lse.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
```

```
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
```

```
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8,13,21) have mixed types. Specify dtype option on import or set low_memory=Fa
lse.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
```

```
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8,13,21) have mixed types. Specify dtype option on import or set low_memory=Fa
lse.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
```

```
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
```

```
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (13,21) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
C:\Users\mhlad\AppData\Local\Temp\ipykernel_18644\828544239.py:10: DtypeWarning: Col
umns (8) have mixed types. Specify dtype option on import or set low_memory=False.
  for chunk in reader:
```

In [ ]:

```
print(df.head())
```

```
   PERMNO        date  SHRCD  EXCHCD  PERMCO  ISSUNO  HEXCD HSICCD     CUSIP  \
0   10001  2004-01-02   11.0     3.0    7953   10398      2   4925  36720410   
1   10001  2004-01-05   11.0     3.0    7953   10398      2   4925  36720410   
2   10001  2004-01-06   11.0     3.0    7953   10398      2   4925  36720410   
3   10001  2004-01-07   11.0     3.0    7953   10398      2   4925  36720410   
4   10001  2004-01-08   11.0     3.0    7953   10398      2   4925  36720410   

   BIDLO  ...       RET   BID   ASK  SHROUT  CFACPR  CFACSHR  OPENPRC  NUMTRD  \
0   6.00  ...  0.100840  6.44  6.59  2596.0     1.5      1.5     6.00    25.0   
1   6.44  ...  0.018321  6.70  6.95  2596.0     1.5      1.5     6.44    22.0   
2   6.65  ...  0.013493  6.65  6.74  2596.0     1.5      1.5     7.00    23.0   
3   6.75  ... -0.001479  6.75  6.80  2596.0     1.5      1.5     6.80    19.0   
4   6.65  ... -0.001482  6.72  6.83  2596.0     1.5      1.5     6.85    44.0   

       RETX    vwretd  
0  0.100840 -0.000784  
1  0.018321  0.012269  
2  0.013493  0.001951  
3 -0.001479  0.002716  
4 -0.001482  0.004865  

[5 rows x 23 columns]
```

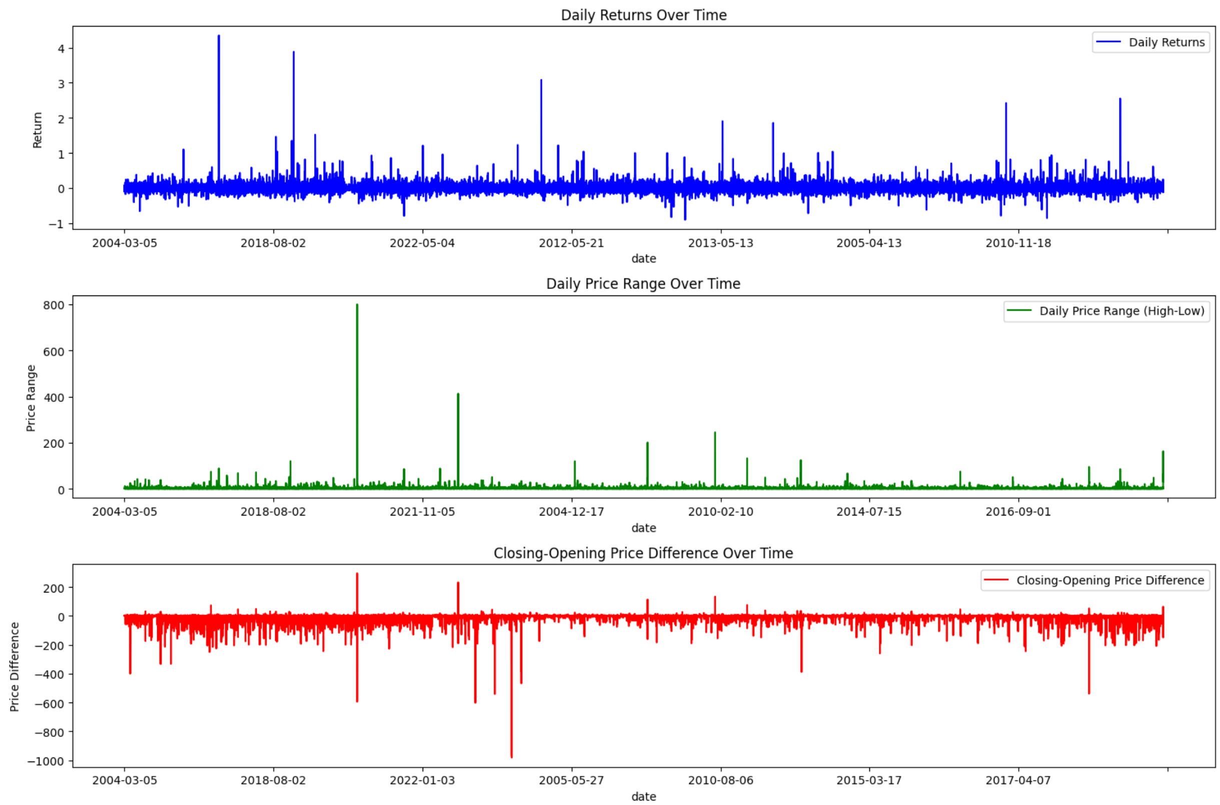
In [ ]:
```
columns = ["BIDLO", "ASKHI", "PRC", "VOL", "RET", "BID", "ASK", "SHROUT", "CFACPR",

for col in columns:
    data_sample[col] = pd.to_numeric(data_sample[col], errors='coerce')

numeric_columns = data_sample.select_dtypes(include=['float64', 'int64']).columns

# Fill missing values using forward and backward fills for numeric columns only
forward_filled = data_sample[numeric_columns].fillna(method='ffill')
backward_filled = data_sample[numeric_columns].fillna(method='bfill')

# Average the filled data
averaged_fill = (forward_filled + backward_filled) / 2

# Fill the original dataframe with averaged values for numeric columns only
data_sample[numeric_columns] = data_sample[numeric_columns].fillna(averaged_fill)
```

In [ ]:
```
# Handle negative closing prices
df['PRC'] = df['PRC'].abs()

# Compute bid-ask spreads
df['Bid-Ask Spread'] = df['ASK'] - df['BID']

# Compute descriptive statistics for the columns
descriptive_stats = df[['RET', 'PRC', 'Bid-Ask Spread', 'VOL']].describe()

print(descriptive_stats)
```

```
                 RET           PRC  Bid-Ask Spread           VOL
count  3.480404e+07  3.480405e+07    3.480405e+07  3.480405e+07
mean   3.367662e-04  3.345272e+01    1.023669e-01  1.041608e+06
std    3.919333e-01  4.712638e+02    1.470632e+00  6.341256e+06
min   -9.996920e-01  0.000000e+00   -7.660000e+00  0.000000e+00
25%   -1.051500e-02  8.300000e+00    1.000000e-02  1.795800e+04
50%    0.000000e+00  1.853000e+01    3.000000e-02  1.154000e+05
75%    1.015200e-02  3.671000e+01    7.000000e-02  5.418290e+05
max    2.299000e+03  3.383899e+05    1.980280e+03  2.144904e+09
```

In [ ]:

```
import matplotlib.pyplot as plt
data_sample.set_index('date', inplace=True)

# Calculate daily range and closing-opening difference
data_sample['Daily Range'] = data_sample['ASKHI'] - data_sample['BIDLO']
data_sample['Close-Open Diff'] = data_sample['PRC'] - data_sample['OPENPRC']

# Compute descriptive statistics


# Plotting
fig, ax = plt.subplots(3, 1, figsize=(15, 10))

data_sample['RET'].plot(ax=ax[0], label='Daily Returns', color='blue')
ax[0].set_title('Daily Returns Over Time')
ax[0].set_ylabel('Return')
ax[0].legend()

data_sample = data_sample[data_sample['Daily Range'] <= 800]
data_sample['Daily Range'].plot(ax=ax[1], label='Daily Price Range (High-Low)', col
ax[1].set_title('Daily Price Range Over Time')
ax[1].set_ylabel('Price Range')
ax[1].legend()

data_sample = data_sample[data_sample['Close-Open Diff'] >= -1000]
data_sample['Close-Open Diff'].plot(ax=ax[2], label='Closing-Opening Price Differen
ax[2].set_title('Closing-Opening Price Difference Over Time')
ax[2].set_ylabel('Price Difference')
ax[2].legend()

plt.tight_layout()
plt.show()
#Now, the x-axis of the plots will display the date, allowing you to see the tempor

descriptive_stats = data_sample[['RET', 'Daily Range', 'Close-Open Diff']].describe
print(descriptive_stats)
```



	RET	Daily Range	Close-Open Diff
count	348541.000000	348541.000000	348541.000000
mean	0.000417	0.770438	-1.324472
std	0.037829	3.474290	11.540687
min	-0.905089	0.000000	-981.635020
25%	-0.010455	0.150000	-0.170000
50%	0.000000	0.365000	0.000000
75%	0.010174	0.810000	0.130000
max	4.348276	800.000000	295.000000

- I had to remove some outliers to make the data more readable

2.2 Plot number of IPO shares vs VWRETD per month

In [ ]:
```python
# Reset index temporarily
data_reset = data_sample.reset_index()

# Convert the 'date' column to datetime format
data_reset['date'] = pd.to_datetime(data_reset['date'])

# Get IPO date for each stock
ipo_dates = data_reset.groupby('PERMNO')['date'].min()

# Calculate the 3 years end date for each IPO date
ipo_dates_3yrs = ipo_dates + pd.DateOffset(years=3)

# Extract the data
ipo_3yrs_data = pd.DataFrame()

for permno, ipo_date in ipo_dates.items():
    end_date = ipo_dates_3yrs[permno]
```

```
    subset_data = data_reset[(data_reset['PERMNO'] == permno) &
                             (data_reset['date'] >= ipo_date) &
                             (data_reset['date'] <= end_date)]
    ipo_3yrs_data = pd.concat([ipo_3yrs_data, subset_data])

# If you wish, set 'date' back as the index
ipo_3yrs_data.set_index('date', inplace=True)

print(ipo_3yrs_data)
```

```
            PERMNO  SHRCD  EXCHCD  PERMCO  ISSUNO  HEXCD  HSICCD     CUSIP  \
date
2004-08-06   10001   11.0     3.0    7953   10398      2  4925.0  36720410
2004-08-16   10001   11.0     3.0    7953   10398      2  4925.0  36720410
2005-01-12   10001   11.0     3.0    7953   10398      2  4925.0  36720410
2005-02-28   10001   11.0     3.0    7953   10398      2  4925.0  36720410
2005-05-23   10001   11.0     3.0    7953   10398      2  4925.0  36720410
...            ...    ...     ...     ...     ...    ...     ...       ...
2013-02-13   93436   11.0     3.0   53453   66252      3  9999.0  88160R10
2013-02-15   93436   11.0     3.0   53453   66252      3  9999.0  88160R10
2013-04-01   93436   11.0     3.0   53453   66252      3  9999.0  88160R10
2013-04-10   93436   11.0     3.0   53453   66252      3  9999.0  88160R10
2013-10-07   93436   11.0     3.0   53453   66252      3  9999.0  88160R10

                BIDLO    ASKHI  ...    SHROUT  CFACPR  CFACSHR    OPENPRC  \
date                            ...
2004-08-06    6.73000    6.880  ...    2599.0     1.5      1.5    6.88000
2004-08-16    6.88000    6.910  ...    2599.0     1.5      1.5    6.88000
2005-01-12    6.50000    6.700  ...    2595.0     1.5      1.5    6.64000
2005-02-28    6.32100    6.321  ...    2599.0     1.5      1.5    6.32100
2005-05-23    8.01000    8.750  ...    2625.0     1.5      1.5    8.40000
...               ...      ...  ...       ...     ...      ...        ...
2013-02-13   38.05000   39.000  ...  114518.0    15.0     15.0   38.30000
2013-02-15   36.95000   38.510  ...  114518.0    15.0     15.0   38.50000
2013-04-01   41.70000   46.680  ...  115161.0    15.0     15.0   42.36000
2013-04-10   40.61000   42.010  ...  115161.0    15.0     15.0   40.70000
2013-10-07  180.25999  186.730  ...  122566.0    15.0     15.0  182.46001

             NUMTRD      RETX    vwretd  Bid-Ask Spread  Daily Range  \
date
2004-08-06      6.0 -0.004215 -0.014952            0.02      0.15000
2004-08-16     13.0  0.007299  0.013976            0.01      0.03000
2005-01-12      3.0 -0.049708  0.004215            0.23      0.20000
2005-02-28      1.0 -0.012344 -0.005807            0.13      0.00000
2005-05-23     37.0  0.010892  0.004614            0.34      0.74000
...             ...       ...       ...             ...          ...
2013-02-13   5526.0  0.014780  0.001391            0.01      0.95000
2013-02-15  11447.0 -0.032898 -0.002011            0.02      1.56000
2013-04-01  66885.0  0.159409 -0.005396            0.07      4.98000
2013-04-10  11264.0  0.033580  0.011627            0.02      1.40000
2013-10-07  65222.0  0.011548 -0.008488            0.08      6.47001

            Close-Open Diff
date
2004-08-06           -0.029
2004-08-16            0.020
2005-01-12           -0.140
2005-02-28            0.000
2005-05-23           -0.140
...                     ...
2013-02-13            0.150
2013-02-15           -1.460
2013-04-01            1.570
2013-04-10            1.160
2013-10-07            0.610
```

```
[127652 rows x 25 columns]
```

In [ ]:
```
for permno, ipo_date in ipo_dates.items():
    end_date = ipo_dates_3yrs[permno]
    subset_data = data_reset[(data_reset['PERMNO'] == permno) &
                             (data_reset['date'] >= ipo_date) &
                             (data_reset['date'] <= end_date)]
    ipo_3yrs_data = pd.concat([ipo_3yrs_data, subset_data])

# If you wish, set 'date' back as the index
ipo_3yrs_data.set_index('date', inplace=True)

print(ipo_3yrs_data)
```

In [ ]:
```
# Convert IPO dates to year-month format for grouping
ipo_dates_df = ipo_dates.reset_index()
ipo_dates_df['year_month'] = ipo_dates_df['date'].dt.to_period('M')

# Count the number of IPOs for each month
ipo_counts_per_month = ipo_dates_df.groupby('year_month').size()

# For a complete list of months from 1963 to 2022, even if some months have zero IP
full_range = pd.period_range(start='1963-01', end='2022-12', freq='M')
ipo_counts_per_month = ipo_counts_per_month.reindex(full_range, fill_value=0)

print(ipo_counts_per_month)
```

```
1963-01     0
1963-02     0
1963-03     0
1963-04     0
1963-05     0
           ..
2022-08    84
2022-09    83
2022-10    72
2022-11    72
2022-12    70
Freq: M, Length: 720, dtype: int64
```

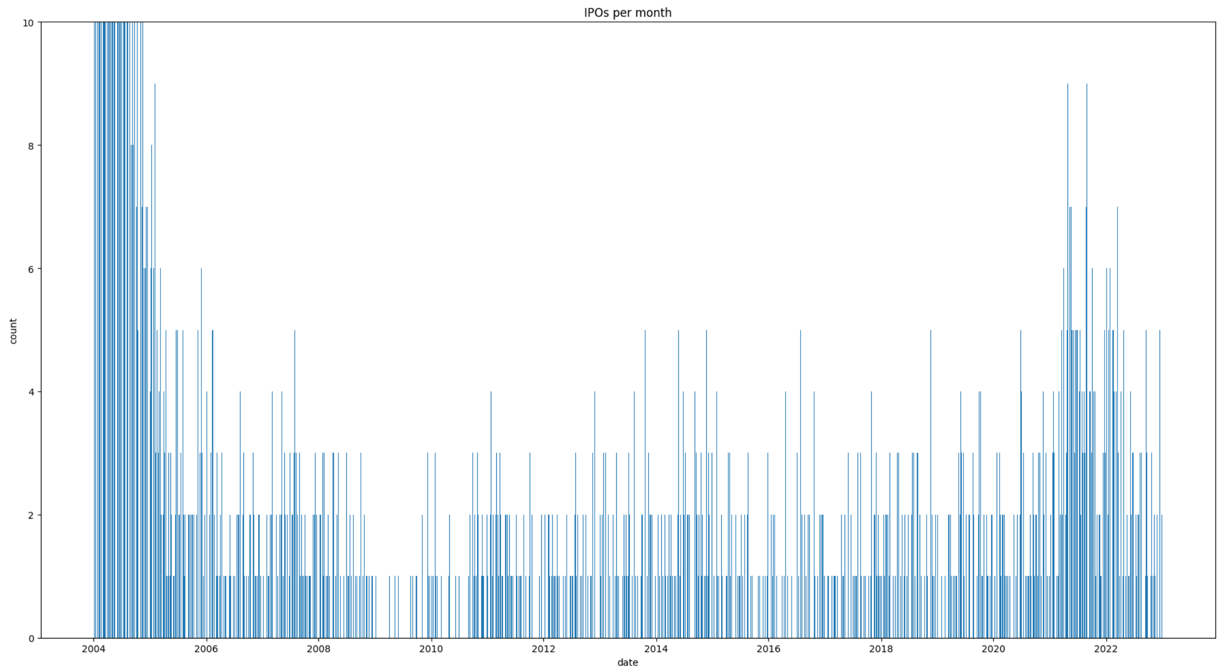
In [ ]:
```
ipo_dates_df = ipo_dates.reset_index()
ipo_dates_df['year_month'] = ipo_dates_df['date'].dt.to_period('M')

# Count the number of IPOs for each month
ipo_counts_per_month = ipo_dates_df.groupby('year_month').size()
```

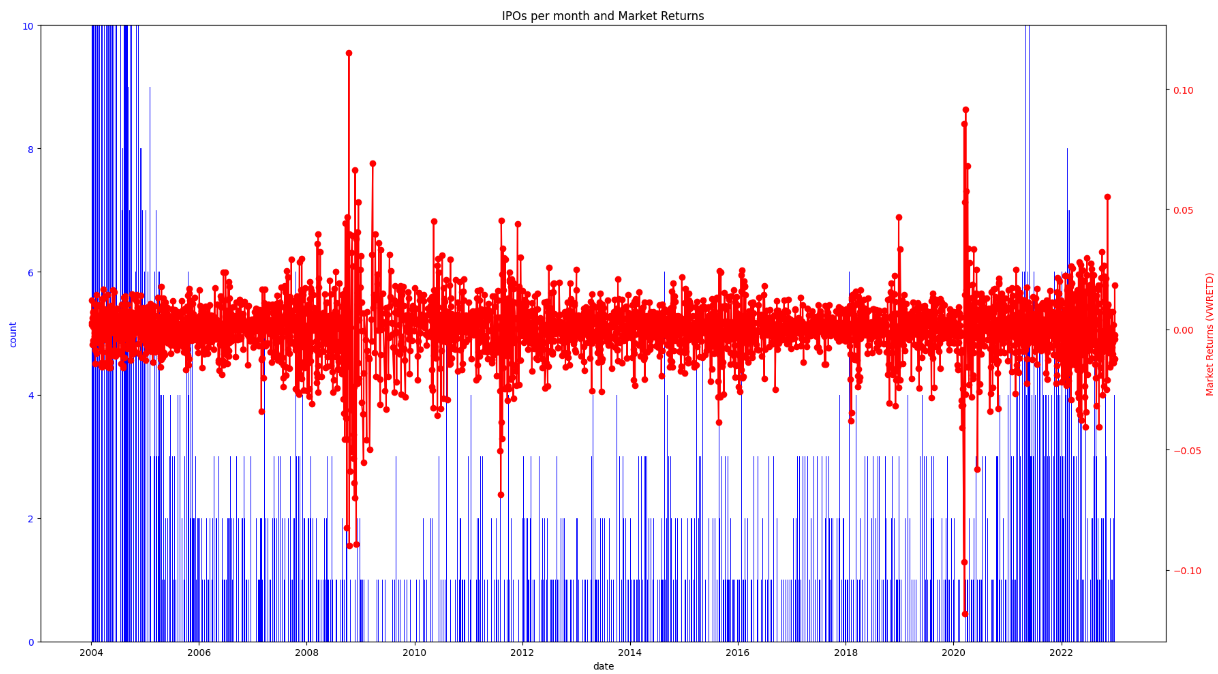
In [ ]:
```
# For a complete list of months from 1963 to 2022, even if some months have zero IP
full_range = pd.period_range(start='1963-01', end='2022-12', freq='M')
ipo_counts_per_month = ipo_counts_per_month.reindex(full_range, fill_value=0)

print(ipo_counts_per_month)
```

2.2 Plot number of IPO shares vs VWRETD per month

In [ ]:
```
#data_sample = data_sample.reset_index()

#Dropping duplicate company names rows, keeping the first occurance
data_sample = data_sample.drop_duplicates(subset='PERMCO', keep='first')

print(data_sample)

#count number of rows for each unique date
count = data_sample['date'].value_counts().reset_index()
count.columns = ['date', 'count']
count['date'] = pd.to_datetime(count['date'])

count = count.sort_values(by='date')
count = count.iloc[1:]


# Calculate average VWRETD for each unique date
avg_returns = data_sample.groupby('date')['vwretd'].mean().reset_index()
avg_returns['date'] = pd.to_datetime(avg_returns['date'])

merged_data = pd.merge(count, avg_returns, on='date', how='left')

fig, ax1 = plt.subplots(figsize=(18, 10))

# Primary y-axis: Number of IPOs
ax1.bar(merged_data["date"], merged_data["count"], color='blue')
ax1.set_xlabel('date')
ax1.set_ylabel('count', color='blue')
ax1.tick_params(axis='y', labelcolor='blue')
ax1.set_title('IPOs per month and Market Returns')
ax1.set_ylim(0, 10)

# Secondary y-axis: Market Returns
ax2 = ax1.twinx()
ax2.plot(merged_data["date"], merged_data["vwretd"], color='red', marker='o')
ax2.set_ylabel('Market Returns (VWRETD)', color='red')
ax2.tick_params(axis='y', labelcolor='red')

fig.tight_layout()
plt.show()
```

```
        level_0   index       date  PERMNO  SHRCD  EXCHCD  PERMCO    ISSUNO  
0             0       0 2004-03-05   10001   11.0     3.0    7953     10398  \
36           36      36 2004-01-30   10002   11.0     3.0    7954     10399   
67           67      67 2004-10-12   10025   11.0     3.0    7975     10432   
95           95      95 2004-02-06   10026   11.0     3.0    7976     10433   
138         138     138 2004-02-27   10028   11.0     3.0    7978     22226   
...         ...     ...        ...     ...    ...     ...     ...       ...   
348464   348464  348464 2011-01-25   93432   11.0     3.0   53450  70065498   
348466   348466  348466 2010-09-21   93433   11.0     3.0   53451     66040   
348483   348483  348483 2011-06-10   93434   11.0     3.0   53427     66342   
348510   348510  348510 2011-10-11   93435   11.0     3.0   53452     29092   
348513   348513  348513 2011-04-29   93436   11.0     3.0   53453     66252   

        HEXCD  HSICCD  ...    ASK   SHROUT     CFACPR  CFACSHR  OPENPRC  
0           2  4925.0  ...   6.35   2596.0   1.500000      1.5     5.81  \
36          3  6020.0  ...  16.11   8745.0   1.000000      1.0    16.92   
67          3  3081.0  ...  11.08   8405.0   1.000000      1.0    11.00   
95          3  2052.0  ...  44.30   8801.0   2.000000      2.0    43.19   
138         2  5094.0  ...   2.88   4913.0   1.000000      1.0     2.74   
...       ...     ...  ...    ...      ...        ...      ...      ...   
348464      3  7389.0  ...   5.57  13659.0   1.000000      1.0     5.75   
348466      3  9999.0  ...   9.09  40046.0   0.110204      0.1     8.95   
348483      3  9999.0  ...   3.78   5800.0   1.000000      1.0     3.75   
348510      3  6163.0  ...   1.16  23864.0   1.000000      1.0     1.16   
348513      3  9999.0  ...  27.60  95633.0  15.000000     15.0    27.69   

        NUMTRD      RETX    vwretd  Daily Range  Close-Open Diff  
0         12.0 -0.079265  0.002974       1.0890            0.451  
36       119.0 -0.048940 -0.000614       0.9100           -0.810  
67        11.0  0.004533 -0.002378       0.0800            0.080  
95       229.0  0.054511  0.014344       1.8800            1.110  
138       15.0 -0.000347  0.001570       0.1400            0.139  
...        ...       ...       ...          ...              ...  
348464   286.0 -0.031579 -0.000429       0.3700           -0.230  
348466   603.0  0.030752 -0.003099       0.3300            0.100  
348483    26.0  0.010638 -0.014085       0.1000            0.050  
348510   156.0 -0.008547  0.001058       0.0599            0.000  
348513  4274.0 -0.002169  0.003120       0.4500           -0.090  

[13812 rows x 27 columns]
```





In [ ]:
```
#count number of rows for each unique date
count = data_sample['date'].value_counts().reset_index()
count.columns = ['date', 'count']
count['date'] = pd.to_datetime(count['date'])

count = count.sort_values(by='date')
count = count.iloc[1:]

# Calculate average VWRETD for each unique date
avg_returns = data_sample.groupby('date')['vwretd'].mean().reset_index()
avg_returns['date'] = pd.to_datetime(avg_returns['date'])

merged_data = pd.merge(count, avg_returns, on='date', how='left')

fig, ax1 = plt.subplots(figsize=(18, 10))
```

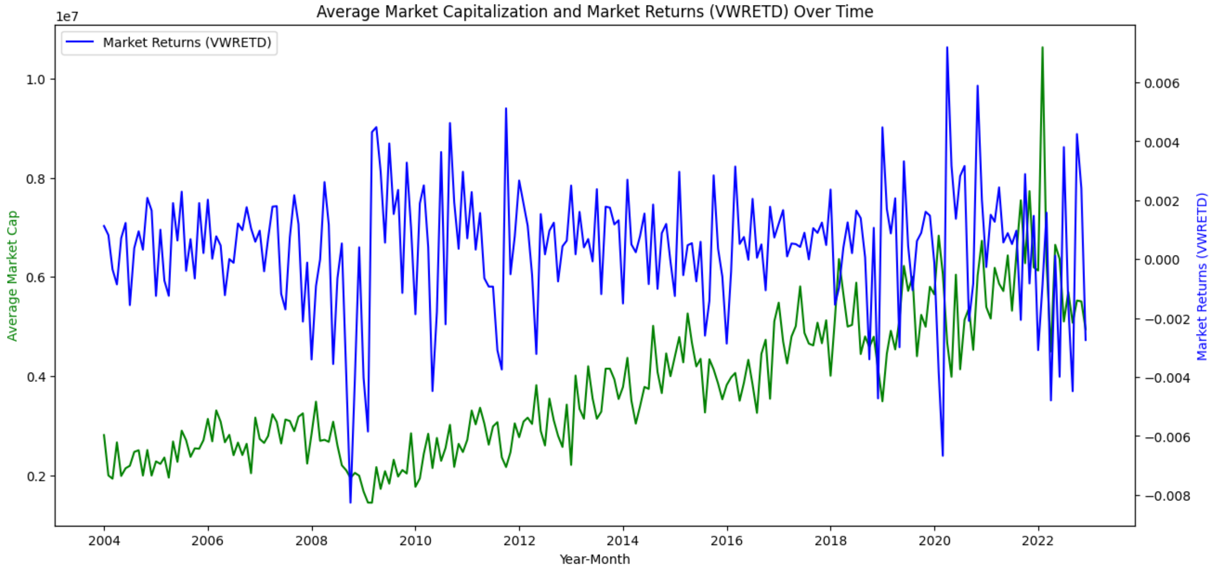
```
# Primary y-axis: Number of IPOs
ax1.bar(merged_data["date"], merged_data["count"], color='blue')
ax1.set_xlabel('date')
ax1.set_ylabel('count', color='blue')
ax1.tick_params(axis='y', labelcolor='blue')
ax1.set_title('IPOs per month and Market Returns')
ax1.set_ylim(0, 10)

# Secondary y-axis: Market Returns
ax2 = ax1.twinx()
ax2.plot(merged_data["date"], merged_data["vwretd"], color='red', marker='o')
ax2.set_ylabel('Market Returns (VWRETD)', color='red')
ax2.tick_params(axis='y', labelcolor='red')

fig.tight_layout()
plt.show()
```

2.3 Plot average size of IPO vs VWRETD per month

In [ ]:

```
data_sample['market_cap'] = data_sample['PRC'].abs() * data_sample['SHROUT']
avg_market_cap_per_month = data_sample.groupby(data_sample.index.to_period('M'))['m
print(avg_market_cap_per_month)
monthly_returns = data_sample.groupby(data_sample.index.to_period('M'))['vwretd'].m

avg_market_cap_per_month.index = avg_market_cap_per_month.index.to_timestamp()
monthly_returns.index = monthly_returns.index.to_timestamp()

# Plot
fig, ax1 = plt.subplots(figsize=(15, 7))

# Twin the axes
ax2 = ax1.twinx()

# Plot data
ax1.plot(avg_market_cap_per_month.index, avg_market_cap_per_month.values, 'g-', lab
ax2.plot(monthly_returns.index, monthly_returns.values, 'b-', label='Market Returns

# Set axis labels and title
ax1.set_xlabel('Year-Month')
ax1.set_ylabel('Average Market Cap', color='g')
ax2.set_ylabel('Market Returns (VWRETD)', color='b')
ax1.set_title('Average Market Capitalization and Market Returns (VWRETD) Over Time'

# Display the plot
plt.legend(loc="upper left")
plt.show()
```

```
date
2004-01    2.809849e+06
2004-02    1.996135e+06
2004-03    1.929766e+06
2004-04    2.662969e+06
2004-05    1.984726e+06
               ...
2022-08    5.695858e+06
2022-09    5.074928e+06
2022-10    5.523071e+06
2022-11    5.505504e+06
2022-12    4.944545e+06
Freq: M, Name: market_cap, Length: 228, dtype: float64
```



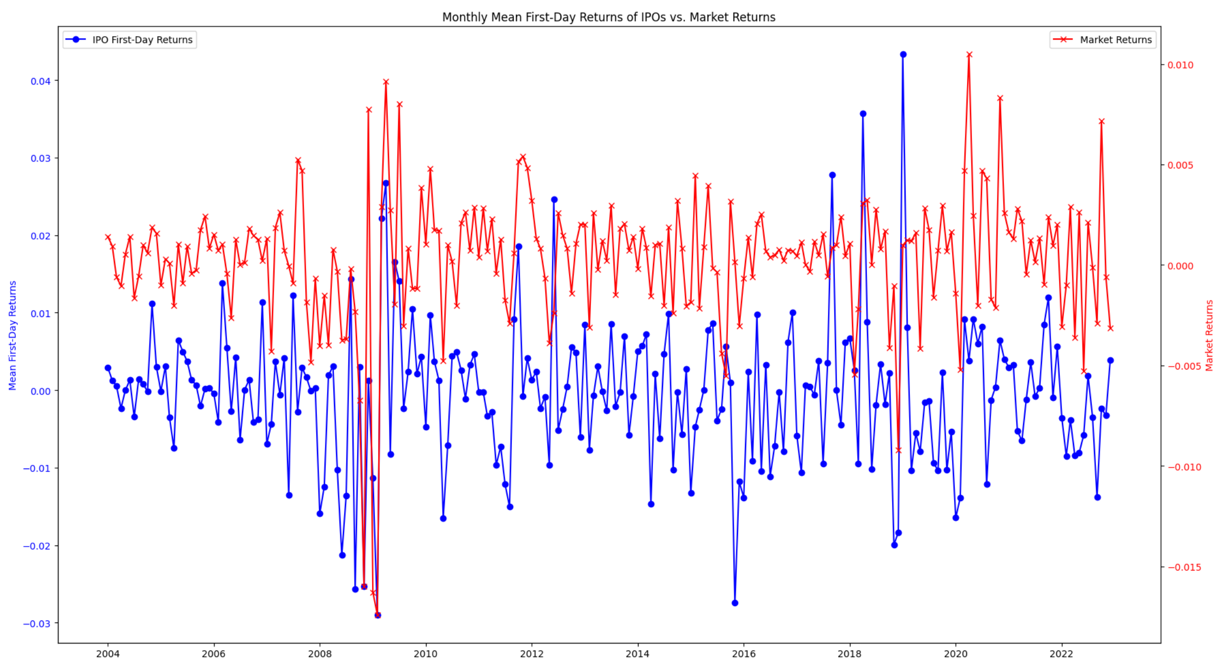
2.2, 2.3, 2.4 Discussion

The number of IPO's fluctuated over these 20 years. In the early 2000's there was a tech boom. A significant rise in technology and internet based companies. The emergence of these companies, such as facebook, uber, airbnb, there was a notable spike in tech IPOs. In addition to the tech sector, biotechnology also saw numerous companies going public to fund drug development and research. By the time 2008 rolled around, the US experienced a historical financial crisis. Market returns dropped, as well as IPO's. The economic downturn, combined with tirghtened liquidity and investor skpticism made it challenging for companies to go public and get desirable valuations. Following the crisis, central banks implemented monetary policy, increasing liquidity, leading to a bull run in stock markets, thus an dincrease in market returns and IPOs over time. Post 2020 pandemic we see a surge in IPO activity. This is due to the pandemic halting all IPO activity, but since then, the market rebounded with record low interest rates and ample liquidity. All of these historical events are seen in the graphs above.

2.4 Plot first day return of IPO vs VWRETD per month

In [ ]:
```
data_sample['date'] = pd.to_datetime(data_sample['date'])
```

```
monthly_means = data_sample.groupby([data_sample['date'].dt.year.rename('year'), da

# Combine the year and month back into a single datetime column
monthly_means['date'] = pd.to_datetime(monthly_means[['year', 'month']].assign(DAY=
monthly_means = monthly_means.drop(columns=['year', 'month'])

fig, ax1 = plt.subplots(figsize=(18, 10))

# Primary y-axis: Mean First-Day Returns of IPOs
ax1.plot(monthly_means["date"], monthly_means["RET"], color='blue', marker='o', lab
ax1.set_xlabel('Date')
ax1.set_ylabel('Mean First-Day Returns', color='blue')
ax1.tick_params(axis='y', labelcolor='blue')
ax1.legend(loc='upper left')

# Secondary y-axis: Market Returns
ax2 = ax1.twinx()
ax2.plot(monthly_means["date"], monthly_means["vwretd"], color='red', marker='x', l
ax2.set_ylabel('Market Returns', color='red')
ax2.tick_params(axis='y', labelcolor='red')
ax2.legend(loc='upper right')

plt.title('Monthly Mean First-Day Returns of IPOs vs. Market Returns')
fig.tight_layout()
plt.show()
```



2.5 IPO Return Analysis

1 day returns

In [ ]:
```
first_day_returns = data_sample1.groupby('PERMCO').first()['RET']
print(first_day_returns)

data_sample1['date'] = pd.to_datetime(data_sample1['date'])
```

```
data_sample1['RET'] = pd.to_numeric(data_sample1['RET'], errors='coerce')

# Drop rows containing NaN values in the 'RET' column
data_sample1 = data_sample1.dropna(subset=['RET'])
from scipy.stats import skew, kurtosis

stats = {}
stats['N'] = len(first_day_returns)
stats['mean'] = first_day_returns.mean()
stats['std_dev'] = first_day_returns.std()
stats['skewness'] = skew(first_day_returns)
stats['kurtosis'] = kurtosis(first_day_returns)
stats['min'] = first_day_returns.min()
stats['max'] = first_day_returns.max()
stats['1%'] = first_day_returns.quantile(0.01)
stats['5%'] = first_day_returns.quantile(0.05)
stats['25%'] = first_day_returns.quantile(0.25)
stats['50%'] = first_day_returns.quantile(0.50)
stats['75%'] = first_day_returns.quantile(0.75)
stats['95%'] = first_day_returns.quantile(0.95)
stats['99%'] = first_day_returns.quantile(0.99)
print(stats)
```

```
PERMCO
5         0.000000
7        -0.021870
25       -0.005636
29       -0.007975
33        0.002206
            ...
59432     0.963540
59435     0.000000
59436     0.012000
59438    -0.113757
59441    -0.046489
Name: RET, Length: 13770, dtype: float64
{'N': 13770, 'mean': -0.0005394550472040669, 'std_dev': 0.042029823473302155, 'skewn
ess': 5.277595452613153, 'kurtosis': 179.09557554320907, 'min': -0.715649, 'max': 1.
481675, '1%': -0.10673866, '5%': -0.052632, '25%': -0.0125895, '50%': 0.0, '75%': 0.
010256, '95%': 0.05, '99%': 0.11210101999999997}
```

1 month return

In [ ]:
```
from scipy.stats import skew, kurtosis



def compute_first_month_return(group):
    first_month = group['date'].iloc[0].month
    first_year = group['date'].iloc[0].year
    first_month_data = group[(group['date'].dt.month == first_month) & (group['date

    # Compute cumulative return for the first month
    cumulative_return = (first_month_data['RET'] + 1).prod() - 1
    return cumulative_return

```

```python
# Sort and group by PERMCO
df_data_sample1 = data_sample1.sort_values(by=['PERMCO', 'date'])
first_month_returns = df_data_sample1.groupby('PERMCO').apply(compute_first_month_r
print(first_month_returns)

stats = {}
stats['N'] = len(first_month_returns)
stats['mean'] = first_month_returns.mean()
stats['std_dev'] = first_month_returns.std()
stats['skewness'] = skew(first_month_returns)
stats['kurtosis'] = kurtosis(first_month_returns)
stats['min'] = first_month_returns.min()
stats['max'] = first_month_returns.max()
stats['1%'] = first_month_returns.quantile(0.01)
stats['5%'] = first_month_returns.quantile(0.05)
stats['25%'] = first_month_returns.quantile(0.25)
stats['50%'] = first_month_returns.quantile(0.50)
stats['75%'] = first_month_returns.quantile(0.75)
stats['95%'] = first_month_returns.quantile(0.95)
stats['99%'] = first_month_returns.quantile(0.99)

print(stats)
```

```
PERMCO
5         0.000000
7        -0.021870
25       -0.005636
29       -0.007975
33        0.002206
            ...
59432     0.963540
59435     0.000000
59436     0.012000
59438    -0.113757
59441    -0.046489
Name: RET, Length: 13770, dtype: float64
PERMCO
5         0.000000
7        -0.021870
25       -0.005636
29       -0.007403
33        0.002206
            ...
59432     0.963540
59435     0.000000
59436     0.012000
59438    -0.113757
59441    -0.049143
Length: 13770, dtype: float64
{'N': 13770, 'mean': -0.0003907737755268307, 'std_dev': 0.044198026486069956, 'skewn
ess': 4.89227590816543, 'kurtosis': 151.47559990573083, 'min': -0.715649, 'max': 1.4
81675, '1%': -0.11158608999999999, '5%': -0.05555600000000005, '25%': -0.01315800000
0000003, '50%': 0.0, '75%': 0.01082099999999997, '95%': 0.05273773316279997, '99%':
0.12423724683774993}
```

12 month return

In [ ]:
```
def compute_12_month_return(group):
    start_date = group['date'].iloc[0]
    end_date = start_date + pd.DateOffset(years=1)
    first_12_months_data = group[(group['date'] >= start_date) & (group['date'] < e

    # Compute cumulative return for the first 12 months
    cumulative_return = (first_12_months_data['RET'] + 1).prod() - 1
    return cumulative_return

# Sort and group by PERMCO
data_sample1 = data_sample1.sort_values(by=['PERMCO', 'date'])
twelve_month_returns = data_sample1.groupby('PERMCO').apply(compute_12_month_return
print(twelve_month_returns)
stats = {}
stats['N'] = len(twelve_month_returns)
stats['mean'] = twelve_month_returns.mean()
stats['std_dev'] = twelve_month_returns.std()
stats['skewness'] = skew(twelve_month_returns)
stats['kurtosis'] = kurtosis(twelve_month_returns)
stats['min'] = twelve_month_returns.min()
stats['max'] = twelve_month_returns.max()
stats['1%'] = twelve_month_returns.quantile(0.01)
stats['5%'] = twelve_month_returns.quantile(0.05)
stats['25%'] = twelve_month_returns.quantile(0.25)
stats['50%'] = twelve_month_returns.quantile(0.50)
stats['75%'] = twelve_month_returns.quantile(0.75)
stats['95%'] = twelve_month_returns.quantile(0.95)
stats['99%'] = twelve_month_returns.quantile(0.99)

print(stats)
```

```
PERMCO
5        0.000000
7        0.005508
25       0.000179
29      -0.017410
33      -0.032493
           ...
59432    0.963540
59435    0.000000
59436    0.012000
59438   -0.113757
59441   -0.049143
Length: 13770, dtype: float64
{'N': 13770, 'mean': -1.1862972045443802e-05, 'std_dev': 0.07293851502145096, 'skewn
ess': 2.844362097210636, 'kurtosis': 64.85133643906374, 'min': -0.715649, 'max': 1.9
441133274453843, '1%': -0.1942481897819914, '5%': -0.10118120460055711, '25%': -0.02
4345036314500024, '50%': 0.0, '75%': 0.022225905426983106, '95%': 0.0999368371918000
5, '99%': 0.21413557688030377}
```

24 month return

In [ ]:
```
def compute_24_month_return(group):
    start_date = group['date'].iloc[0]
    end_date = start_date + pd.DateOffset(years=2)
```

```
    first_24_months_data = group[(group['date'] >= start_date) & (group['date'] < e

    # Compute cumulative return for the first 24 months
    cumulative_return = (first_24_months_data['RET'] + 1).prod() - 1
    return cumulative_return

# Sort and group by PERMCO
data_sample1 = data_sample1.sort_values(by=['PERMCO', 'date'])
twenty_four_month_returns = data_sample1.groupby('PERMCO').apply(compute_24_month_r
print(twenty_four_month_returns)
stats = {}
stats['N'] = len(twenty_four_month_returns)
stats['mean'] = twenty_four_month_returns.mean()
stats['std_dev'] = twenty_four_month_returns.std()
stats['skewness'] = skew(twenty_four_month_returns)
stats['kurtosis'] = kurtosis(twenty_four_month_returns)
stats['min'] = twenty_four_month_returns.min()
stats['max'] = twenty_four_month_returns.max()
stats['1%'] = twenty_four_month_returns.quantile(0.01)
stats['5%'] = twenty_four_month_returns.quantile(0.05)
stats['25%'] = twenty_four_month_returns.quantile(0.25)
stats['50%'] = twenty_four_month_returns.quantile(0.50)
stats['75%'] = twenty_four_month_returns.quantile(0.75)
stats['95%'] = twenty_four_month_returns.quantile(0.95)
stats['99%'] = twenty_four_month_returns.quantile(0.99)

print(stats)
```

```
PERMCO
5       -0.039484
7        0.027409
25      -0.000789
29      -0.024208
33       0.004906
           ...
59432    0.963540
59435    0.000000
59436    0.012000
59438   -0.113757
59441   -0.049143
Length: 13770, dtype: float64
{'N': 13770, 'mean': 0.0007489505869738743, 'std_dev': 0.09335833014101658, 'skewnes
s': 2.118243195833167, 'kurtosis': 32.465991961385214, 'min': -0.715649, 'max': 1.90
24511797487045, '1%': -0.24858262739072692, '5%': -0.13300727784729469, '25%': -0.03
330642381984392, '50%': -1.0133607263040911e-05, '75%': 0.030787084726744463, '95%':
0.13254474013467296, '99%': 0.2843795066194436}
```

36 month return

In [ ]:
```
from scipy.stats import skew, kurtosis

def compute_36_month_return(group):
    start_date = group['date'].iloc[0]
    end_date = start_date + pd.DateOffset(years=3)
    first_36_months_data = group[(group['date'] >= start_date) & (group['date'] < e
```

```
    # Compute cumulative return for the first 36 months
    cumulative_return = (first_36_months_data['RET'] + 1).prod() - 1
    return cumulative_return

# Sort and group by PERMCO
data_sample1 = data_sample1.sort_values(by=['PERMCO', 'date'])
thirty_six_month_returns = df_data_sample1.groupby('PERMCO').apply(compute_36_month
print(thirty_six_month_returns)

stats = {}
stats['N'] = len(thirty_six_month_returns)
stats['mean'] = thirty_six_month_returns.mean()
stats['std_dev'] = thirty_six_month_returns.std()
stats['skewness'] = skew(thirty_six_month_returns)
stats['kurtosis'] = kurtosis(thirty_six_month_returns)
stats['min'] = thirty_six_month_returns.min()
stats['max'] = thirty_six_month_returns.max()
stats['1%'] = thirty_six_month_returns.quantile(0.01)
stats['5%'] = thirty_six_month_returns.quantile(0.05)
stats['25%'] = thirty_six_month_returns.quantile(0.25)
stats['50%'] = thirty_six_month_returns.quantile(0.50)
stats['75%'] = thirty_six_month_returns.quantile(0.75)
stats['95%'] = thirty_six_month_returns.quantile(0.95)
stats['99%'] = thirty_six_month_returns.quantile(0.99)

print(stats)
```

```
PERMCO
5       -0.072415
7        0.044897
25      -0.000196
29      -0.007292
33       0.009322
           ...
59432    0.963540
59435    0.000000
59436    0.012000
59438   -0.113757
59441   -0.049143
Length: 13770, dtype: float64
{'N': 13770, 'mean': 0.001843646052418374, 'std_dev': 0.10815213441661818, 'skewnes
s': 2.28730809674088, 'kurtosis': 34.84572972821234, 'min': -0.8199333534433886, 'ma
x': 2.133005408683566, '1%': -0.28080693782194144, '5%': -0.15027842013278417, '2
5%': -0.03906111859676592, '50%': 0.0, '75%': 0.037621130778424994, '95%': 0.1563550
2366520102, '99%': 0.3299668063242318}
```

2.5 Discussion

As the IPO's age, the return overall increases. In the early stages of the maturity (i.e., 1 month, 12 month..) there is higher kurtosis, which means that theres more volatility in the returns and losses.