Imports & Settings

In [1]:

```
import numpy as np
import pandas as pd
import yfinance as yf

# Visualization tools
import missingno as msno
import mplfinance as mpf
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

# Ignoring warnings
import warnings
warnings.filterwarnings('ignore')
```

Data Gathering

S&P 500 Constituents

In [2]:

```
snp = pd.read_csv('../data/sp500.csv')
snp.head()
```

Out[2]:

	Symbol	Security	SEC filings	GICS Sector	GICS Sub-Industry	Headquarters Location	Date first added	CIK	Founded
0	MMM	3M	reports	Industrials	Industrial Conglomerates	Saint Paul, Minnesota	1976-08-09	66740	1902
1	ABT	Abbott Laboratories	reports	Health Care	Health Care Equipment	North Chicago, Illinois	1964-03-31	1800	1888
2	ABBV	AbbVie	reports	Health Care	Pharmaceuticals	North Chicago, Illinois	2012-12-31	1551152	2013 (1888)
3	ABMD	Abiomed	reports	Health Care	Health Care Equipment	Danvers, Massachusetts	2018-05-31	815094	1981
4	ACN	Accenture	reports	Information Technology	IT Consulting & Other Services	Dublin, Ireland	2011-07-06	1467373	1989

In [3]:

```
snp_tickers = snp.Symbol.values.tolist()
```

Price Histories

In [7]:

```
prices = yf.download(snp_tickers, group_by='tickers', period='max')
```

```
[*********************100%***********************]  505 of 505 completed

1 Failed download:
- BF.B: 1d data not available for startTime=-2208988800 and endTime=1629577039. Only 100 years worth of day granularity data are allo
wed to be fetched per request.
```

It appears that the ticker `BF.B` failed. The '.B' portion of the ticker refers to the specific share class of the security. Referencing the share class in a ticker is not fully standardized across different financial sources and can sometimes be represented as a hyphen instead of a period. I will try to download the data for `BF.B` by manually changing it to `BF-B` instead.

In [8]:

```
bfb = yf.download(tickers='BF-B', period='max')
```

```
[*********************100%***********************]  1 of 1 completed
```

In [10]:

```
prices.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 15014 entries, 1962-01-02 to 2021-08-20
Columns: 3030 entries, ('COST', 'Open') to ('ZION', 'Volume')
dtypes: float64(3030)
memory usage: 347.2 MB
```

The data returned by the `yfinance.download()` function placed everything into a single dataframe with columns grouped by ticker. While useful for quickly analyzing / plotting any of the companies' histories, there are two primary issues with having everything in one dataframe:

- The file is much larger than the 100MB limit imposed by GitHub. While solutions for handling large file uploads exist, it's preferable to avoid such workarounds unless absolutely needed.
- Thinking forward to the interactive dashboard portion of this project, loading in all of the price history data for every company in the S&P 500 whenever we want to analyze just one is a waste of resources and will likely lead to unwanted load times.

By splitting the data up into individual files, both of these issues are resolved. To start, I'll export the `BF.B / BF-B` dataframe and then move on to looping over all of the others.

In [11]:

```
bfb.to_csv('../data/price_histories/BF-B_history.csv')
print('Successfully saved BF-B\'s history.')
```

```
Successfully saved BF-B's history.
```

In [16]:

```
for i, ticker in enumerate(snp_tickers):
    if ticker == 'BF.B':
        continue
    df = prices[ticker].dropna()
    df.to_csv(f'../data/price_histories/{ticker}_history.csv')
    print(f'{ticker.ljust(5)} -- {i+1} / {len(snp_tickers)} completed', end='\r')
```

```
ZTS   -- 505 / 505 completed
```

Fundamental Data

Income Statement

In [12]:

```
df_income = pd.read_csv('../data/simfin/us-income-quarterly.csv',
                        delimiter=';',
                        parse_dates=['Report Date', 'Publish Date', 'Restated Date'])
df_income.head()
```

Out[12]:

	Ticker	SimFinId	Currency	Fiscal Year	Fiscal Period	Report Date	Publish Date	Restated Date	Shares (Basic)	Shares (Diluted)	...	Non-Operating Income (Loss)	Interest Expense, Net	Pretax Income (Loss), Adj.	Abnormal Gains (Losses)	Pretax Income (Loss
0	A	45846	USD	2010	Q3	2010-07-31	2010-10-06	2011-09-07	347000000.0	352000000.0	...	-15000000.0	-21000000.0	100000000	127000000.0	22700000(
1	A	45846	USD	2010	Q4	2010-10-31	2010-12-20	2012-12-20	344000000.0	356000000.0	...	35000000.0	-16000000.0	238000000	5000000.0	24300000(
2	A	45846	USD	2011	Q1	2011-01-31	2011-03-09	2012-03-05	347000000.0	355000000.0	...	-13000000.0	-19000000.0	198000000	NaN	19800000(
3	A	45846	USD	2011	Q2	2011-04-30	2011-06-07	2012-06-04	347000000.0	355000000.0	...	-6000000.0	-17000000.0	260000000	NaN	26000000(
4	A	45846	USD	2011	Q3	2011-07-31	2011-09-07	2012-09-05	348000000.0	357000000.0	...	0.0	-17000000.0	281000000	NaN	28100000(

5 rows × 28 columns

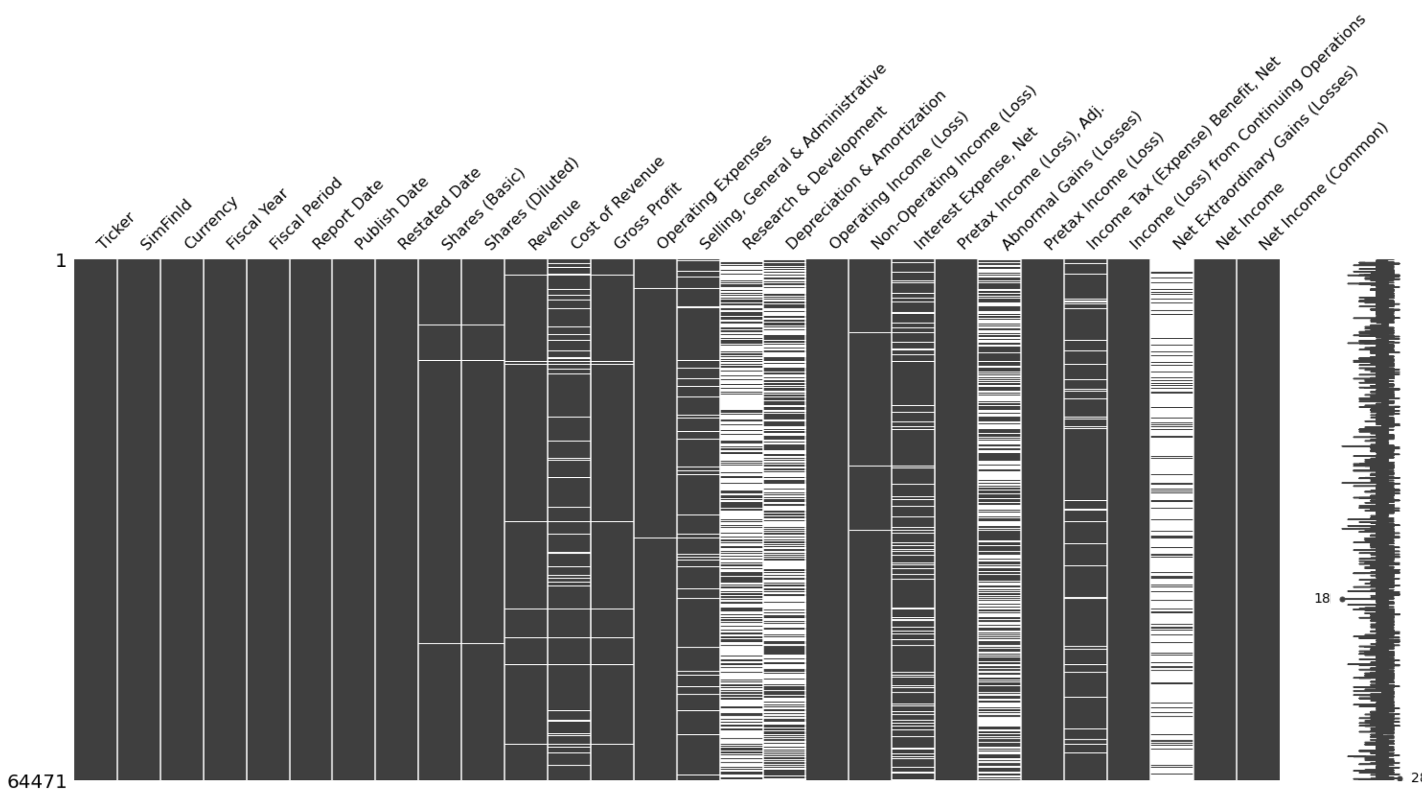
In [13]:

```
df_income.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64471 entries, 0 to 64470
Data columns (total 28 columns):
 #   Column                               Non-Null Count  Dtype
---  ------                               --------------  -----
 0   Ticker                               64471 non-null  object
 1   SimFinId                             64471 non-null  int64
 2   Currency                             64471 non-null  object
 3   Fiscal Year                          64471 non-null  int64
 4   Fiscal Period                        64471 non-null  object
 5   Report Date                          64471 non-null  datetime64[ns]
 6   Publish Date                         64471 non-null  datetime64[ns]
 7   Restated Date                        64471 non-null  datetime64[ns]
 8   Shares (Basic)                       63932 non-null  float64
 9   Shares (Diluted)                     63932 non-null  float64
 10  Revenue                              63435 non-null  float64
 11  Cost of Revenue                      58921 non-null  float64
 12  Gross Profit                         63466 non-null  float64
 13  Operating Expenses                   64313 non-null  float64
 14  Selling, General & Administrative    60840 non-null  float64
 15  Research & Development               25254 non-null  float64
 16  Depreciation & Amortization          29210 non-null  float64
 17  Operating Income (Loss)              64470 non-null  float64
 18  Non-Operating Income (Loss)          63844 non-null  float64
 19  Interest Expense, Net                56785 non-null  float64
 20  Pretax Income (Loss), Adj.           64471 non-null  int64
 21  Abnormal Gains (Losses)              34590 non-null  float64
 22  Pretax Income (Loss)                 64471 non-null  int64
 23  Income Tax (Expense) Benefit, Net    59898 non-null  float64
 24  Income (Loss) from Continuing Operations  64471 non-null  int64
 25  Net Extraordinary Gains (Losses)     10761 non-null  float64
 26  Net Income                           64471 non-null  int64
 27  Net Income (Common)                  64471 non-null  int64
dtypes: datetime64[ns](3), float64(15), int64(7), object(3)
memory usage: 13.8+ MB
```

In [25]:

```
msno.matrix(df_income);
```



Balance Sheet

In [10]:

```python
df_balance = pd.read_csv('../data/simfin/us-balance-quarterly.csv',
                         delimiter=';',
                         parse_dates=['Report Date', 'Publish Date', 'Restated Date'])
df_balance.head()
```

Out[10]:

	Ticker	SimFinId	Currency	Fiscal Year	Fiscal Period	Report Date	Publish Date	Restated Date	Shares (Basic)	Shares (Diluted)	...	Short Term Debt	Total Current Liabilities	Long Term Debt	Total Noncurrent Liabilities	
0	A	45846	USD	2010	Q3	2010-07-31	2010-10-06	2010-10-06	347000000.0	352000000.0	...	1.501000e+09	2.917000e+09	2.177000e+09	3.373000e+09	62
1	A	45846	USD	2010	Q4	2010-10-31	2010-12-20	2011-12-16	344000000.0	356000000.0	...	1.501000e+09	3.083000e+09	2.190000e+09	3.377000e+09	64
2	A	45846	USD	2011	Q1	2011-01-31	2011-03-09	2011-03-09	347000000.0	355000000.0	...	1.000000e+06	1.406000e+09	2.138000e+09	3.299000e+09	47
3	A	45846	USD	2011	Q2	2011-04-30	2011-06-07	2011-06-07	347000000.0	355000000.0	...	0.000000e+00	1.592000e+09	2.144000e+09	3.096000e+09	46
4	A	45846	USD	2011	Q3	2011-07-31	2011-09-07	2011-09-07	348000000.0	357000000.0	...	0.000000e+00	1.505000e+09	2.168000e+09	3.048000e+09	45

5 rows × 30 columns

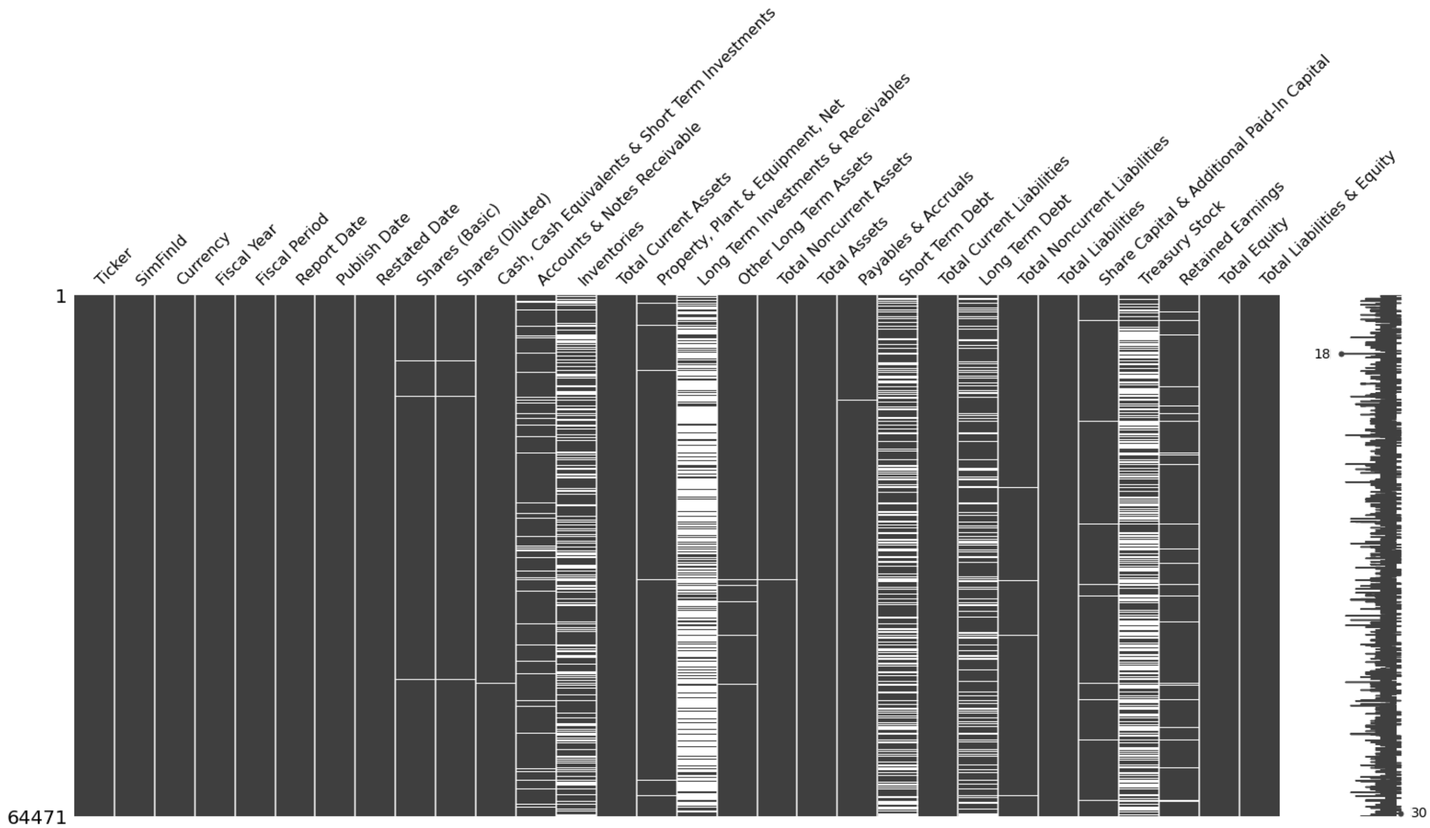
In [11]:

```python
df_balance.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64471 entries, 0 to 64470
Data columns (total 30 columns):
 #   Column                                         Non-Null Count  Dtype
---  ------                                         --------------  -----
 0   Ticker                                         64471 non-null  object
 1   SimFinId                                       64471 non-null  int64
 2   Currency                                       64471 non-null  object
 3   Fiscal Year                                    64471 non-null  int64
 4   Fiscal Period                                  64471 non-null  object
 5   Report Date                                    64471 non-null  datetime64[ns]
 6   Publish Date                                   64471 non-null  datetime64[ns]
 7   Restated Date                                  64471 non-null  datetime64[ns]
 8   Shares (Basic)                                 63932 non-null  float64
 9   Shares (Diluted)                               63932 non-null  float64
 10  Cash, Cash Equivalents & Short Term Investments  64311 non-null  float64
 11  Accounts & Notes Receivable                    59102 non-null  float64
 12  Inventories                                    44891 non-null  float64
 13  Total Current Assets                           64467 non-null  float64
 14  Property, Plant & Equipment, Net               63509 non-null  float64
 15  Long Term Investments & Receivables            17592 non-null  float64
 16  Other Long Term Assets                         63754 non-null  float64
 17  Total Noncurrent Assets                        64262 non-null  float64
 18  Total Assets                                   64471 non-null  int64
 19  Payables & Accruals                            64237 non-null  float64
 20  Short Term Debt                                42547 non-null  float64
 21  Total Current Liabilities                      64469 non-null  float64
 22  Long Term Debt                                 51006 non-null  float64
 23  Total Noncurrent Liabilities                   63706 non-null  float64
 24  Total Liabilities                              64471 non-null  int64
 25  Share Capital & Additional Paid-In Capital     63379 non-null  float64
 26  Treasury Stock                                 30427 non-null  float64
 27  Retained Earnings                              61579 non-null  float64
 28  Total Equity                                   64470 non-null  float64
 29  Total Liabilities & Equity                     64471 non-null  int64
dtypes: datetime64[ns](3), float64(19), int64(5), object(3)
memory usage: 14.8+ MB
```

In [28]:

```
msno.matrix(df_balance);
```



Cash Flow Statement

In [8]:

```
df_cashflow = pd.read_csv('../data/simfin/us-cashflow-quarterly.csv',
                          delimiter=';',
                          parse_dates=['Report Date', 'Publish Date', 'Restated Date'])
df_cashflow.head()
```

Out[8]:

	Ticker	SimFinId	Currency	Fiscal Year	Fiscal Period	Report Date	Publish Date	Restated Date	Shares (Basic)	Shares (Diluted)	...	Net Cash from Operating Activities	Change in Fixed Assets & Intangibles	Net Change in Long Term Investment	Net Cash from Acquisitions & Divestitures	Net Ca: In Ac
0	A	45846	USD	2010	Q3	2010-07-31	2010-10-06	2011-09-07	347000000.0	352000000.0	...	90000000.0	-27000000.0	30000000.0	-1.102000e+09	-1.112C
1	A	45846	USD	2010	Q4	2010-10-31	2010-12-20	2012-12-20	344000000.0	356000000.0	...	373000000.0	-34000000.0	0.0	-1.400000e+07	-1.400C
2	A	45846	USD	2011	Q1	2011-01-31	2011-03-09	2012-03-05	347000000.0	355000000.0	...	120000000.0	-38000000.0	5000000.0	0.000000e+00	1.500C
3	A	45846	USD	2011	Q2	2011-04-30	2011-06-07	2012-06-04	347000000.0	355000000.0	...	378000000.0	-51000000.0	9000000.0	-9.600000e+07	-1.260C
4	A	45846	USD	2011	Q3	2011-07-31	2011-09-07	2012-09-05	348000000.0	357000000.0	...	252000000.0	-32000000.0	0.0	0.000000e+00	-3.200C

5 rows × 28 columns

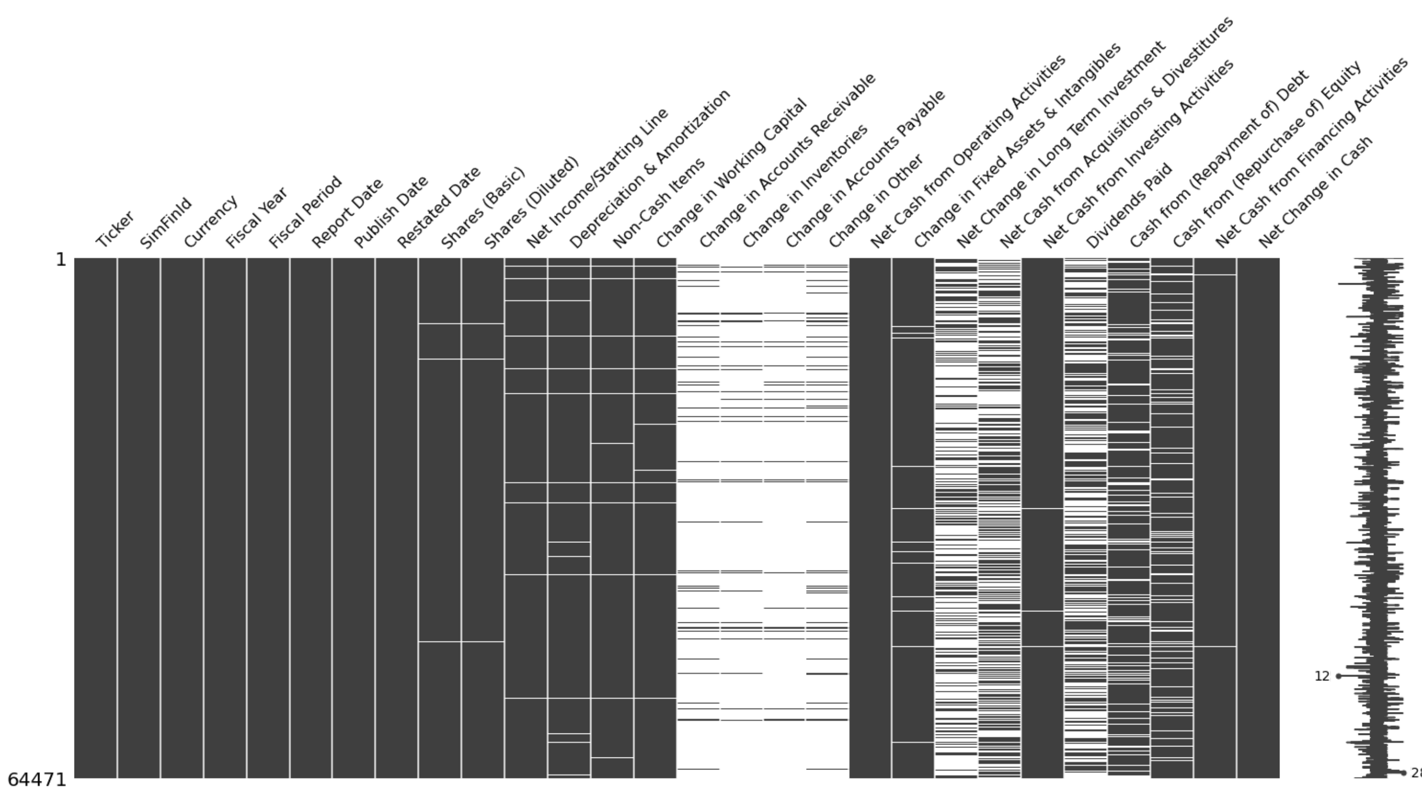
In [9]:

```
df_cashflow.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64471 entries, 0 to 64470
Data columns (total 28 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   Ticker                                 64471 non-null  object
 1   SimFinId                               64471 non-null  int64
 2   Currency                               64471 non-null  object
 3   Fiscal Year                            64471 non-null  int64
 4   Fiscal Period                          64471 non-null  object
 5   Report Date                            64471 non-null  datetime64[ns]
 6   Publish Date                           64471 non-null  datetime64[ns]
 7   Restated Date                          64471 non-null  datetime64[ns]
 8   Shares (Basic)                         63932 non-null  float64
 9   Shares (Diluted)                       63932 non-null  float64
 10  Net Income/Starting Line               63449 non-null  float64
 11  Depreciation & Amortization            62356 non-null  float64
 12  Non-Cash Items                         63349 non-null  float64
 13  Change in Working Capital              63086 non-null  float64
 14  Change in Accounts Receivable          5567 non-null   float64
 15  Change in Inventories                  4126 non-null   float64
 16  Change in Accounts Payable             4408 non-null   float64
 17  Change in Other                        6530 non-null   float64
 18  Net Cash from Operating Activities     64467 non-null  float64
 19  Change in Fixed Assets & Intangibles   62995 non-null  float64
 20  Net Change in Long Term Investment     22432 non-null  float64
 21  Net Cash from Acquisitions & Divestitures  31335 non-null  float64
 22  Net Cash from Investing Activities     64174 non-null  float64
 23  Dividends Paid                         32441 non-null  float64
 24  Cash from (Repayment of) Debt          53911 non-null  float64
 25  Cash from (Repurchase of) Equity       54287 non-null  float64
 26  Net Cash from Financing Activities     64078 non-null  float64
 27  Net Change in Cash                     64471 non-null  int64
dtypes: datetime64[ns](3), float64(19), int64(3), object(3)
memory usage: 13.8+ MB
```

In [31]:

```
msno.matrix(df_cashflow);
```



Merging Data

In [109]:

```
for i, ticker in enumerate(snp_tickers):
    # Manually catching 'BF.B'
    if ticker == 'BF.B':
        ticker = 'BF-B'

    # Setting dataframes
    prices = pd.read_csv(f'../data/price_histories/{ticker}_history.csv', parse_dates=['Date'])
    prices.rename(columns={'Date': 'Price Date'}, inplace=True)
    income = df_income[df_income.Ticker == ticker].set_index('Report Date')
    balance = df_balance[df_balance.Ticker == ticker].set_index('Report Date')
    cashflow = df_cashflow[df_cashflow.Ticker == ticker].set_index('Report Date')

    # Concatenating financials
    financials = pd.concat([income, balance, cashflow], axis=1).reset_index()

    # Dropping unnecessary columns
    to_drop = ['Ticker', 'SimFinId', 'Currency', 'Fiscal Year', 'Fiscal Period',
               'Publish Date', 'Restated Date', 'Shares (Basic)', 'Shares (Diluted)']
    financials = financials.drop(columns=to_drop)

    # Merging in prices
    final_df = pd.merge_asof(left=financials,
                             right=prices,
                             left_on='Report Date',
                             right_on='Price Date',
                             direction='backward')

    # Exporting dataframe
    final_df.to_csv(f'../data/merged_data/{ticker}_merged.csv', index=False)

    # Printing progress
    print(f'{i+1}/{len(snp_tickers)} -- merged and saved {ticker.ljust(5)}', end='\r')
```

```
505/505 -- merged and saved ZTS
```

Loading in an example to ensure everything worked properly:

In [112]:

```
msft_merged = pd.read_csv('../data/merged_data/MSFT_merged.csv', parse_dates=['Report Date', 'Price Date'])
msft_merged
```

Out[112]:

	Report Date	Revenue	Cost of Revenue	Gross Profit	Operating Expenses	Selling, General & Administrative	Research & Development	Depreciation & Amortization	Operating Income (Loss)	Non-Operating Income (Loss)	...	Cash from (Repurchase of) Equity
0	2004-03-31	9.175000e+09	-1.411000e+09	7.764000e+09	-6.486000e+09	-4.948000e+09	-1.538000e+09	NaN	1.278000e+09	1.001000e+09	...	-1.040000e+09
1	2004-06-30	9.292000e+09	-1.361000e+09	7.931000e+09	-4.798000e+09	-3.183000e+09	-1.615000e+09	NaN	3.133000e+09	5.710000e+08	...	1.296000e+09
2	2004-09-30	9.189000e+09	-1.405000e+09	7.784000e+09	-4.290000e+09	-2.760000e+09	-1.530000e+09	NaN	3.494000e+09	2.790000e+08	...	1.320000e+08
3	2004-12-31	1.081800e+10	-1.875000e+09	8.943000e+09	-4.194000e+09	-2.773000e+09	-1.421000e+09	NaN	4.749000e+09	4.200000e+08	...	-1.740000e+08
4	2005-03-31	9.620000e+09	-1.363000e+09	8.257000e+09	-4.928000e+09	-3.446000e+09	-1.482000e+09	NaN	3.329000e+09	4.960000e+08	...	-2.073000e+09
...	...	...	...	...	...	...	...	...	...	...	...	...
61	2019-06-30	3.371700e+10	-1.041200e+10	2.330500e+10	-1.090000e+10	-6.387000e+09	-4.513000e+09	NaN	1.240500e+10	1.910000e+08	...	-4.325000e+09
62	2019-09-30	3.305500e+10	-1.040600e+10	2.264900e+10	-9.963000e+09	-5.398000e+09	-4.565000e+09	NaN	1.268600e+10	0.000000e+00	...	-4.485000e+09
63	2019-12-31	3.690600e+10	-1.235800e+10	2.454800e+10	-1.065700e+10	-6.054000e+09	-4.603000e+09	NaN	1.389100e+10	1.940000e+08	...	-4.972000e+09
64	2020-03-31	3.502100e+10	-1.097500e+10	2.404600e+10	-1.107100e+10	-6.184000e+09	-4.887000e+09	NaN	1.297500e+10	-1.320000e+08	...	-6.717000e+09
65	2020-06-30	3.803300e+10	-1.233900e+10	2.569400e+10	-1.228700e+10	-7.073000e+09	-5.214000e+09	NaN	1.340700e+10	1.500000e+07	...	-5.451000e+09

66 rows × 64 columns

In [113]:

```
msft_merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 66 entries, 0 to 65
Data columns (total 64 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   Report Date                                 66 non-null     datetime64[ns]
 1   Revenue                                     66 non-null     float64
 2   Cost of Revenue                             66 non-null     float64
 3   Gross Profit                                66 non-null     float64
 4   Operating Expenses                          66 non-null     float64
 5   Selling, General & Administrative           66 non-null     float64
 6   Research & Development                      66 non-null     float64
 7   Depreciation & Amortization                 0 non-null      float64
 8   Operating Income (Loss)                     66 non-null     float64
 9   Non-Operating Income (Loss)                 66 non-null     float64
 10  Interest Expense, Net                       0 non-null      float64
 11  Pretax Income (Loss), Adj.                  66 non-null     int64
 12  Abnormal Gains (Losses)                     17 non-null     float64
 13  Pretax Income (Loss)                        66 non-null     int64
 14  Income Tax (Expense) Benefit, Net           66 non-null     float64
 15  Income (Loss) from Continuing Operations    66 non-null     int64
 16  Net Extraordinary Gains (Losses)            0 non-null      float64
 17  Net Income                                  66 non-null     int64
 18  Net Income (Common)                         66 non-null     int64
 19  Cash, Cash Equivalents & Short Term Investments  66 non-null     float64
 20  Accounts & Notes Receivable                 66 non-null     float64
 21  Inventories                                 66 non-null     float64
 22  Total Current Assets                        66 non-null     float64
 23  Property, Plant & Equipment, Net            66 non-null     float64
 24  Long Term Investments & Receivables         66 non-null     float64
 25  Other Long Term Assets                      66 non-null     float64
 26  Total Noncurrent Assets                     66 non-null     float64
 27  Total Assets                                66 non-null     int64
 28  Payables & Accruals                         66 non-null     float64
 29  Short Term Debt                             45 non-null     float64
 30  Total Current Liabilities                   66 non-null     float64
 31  Long Term Debt                              45 non-null     float64
 32  Total Noncurrent Liabilities                66 non-null     float64
 33  Total Liabilities                           66 non-null     int64
 34  Share Capital & Additional Paid-In Capital  66 non-null     float64
 35  Treasury Stock                              0 non-null      float64
 36  Retained Earnings                           66 non-null     float64
 37  Total Equity                                66 non-null     float64
 38  Total Liabilities & Equity                  66 non-null     int64
 39  Net Income/Starting Line                    66 non-null     float64
 40  Depreciation & Amortization.1               66 non-null     float64
 41  Non-Cash Items                              66 non-null     float64
 42  Change in Working Capital                   66 non-null     float64
 43  Change in Accounts Receivable               66 non-null     float64
 44  Change in Inventories                       45 non-null     float64
 45  Change in Accounts Payable                  45 non-null     float64
 46  Change in Other                             66 non-null     float64
 47  Net Cash from Operating Activities          66 non-null     float64
 48  Change in Fixed Assets & Intangibles        66 non-null     float64
 49  Net Change in Long Term Investment          66 non-null     float64
 50  Net Cash from Acquisitions & Divestitures   64 non-null     float64
 51  Net Cash from Investing Activities          66 non-null     float64
 52  Dividends Paid                              65 non-null     float64
 53  Cash from (Repayment of) Debt               48 non-null     float64
 54  Cash from (Repurchase of) Equity            66 non-null     float64
 55  Net Cash from Financing Activities          66 non-null     float64
 56  Net Change in Cash                          66 non-null     int64
 57  Price Date                                  66 non-null     datetime64[ns]
 58  Open                                        66 non-null     float64
 59  High                                        66 non-null     float64
 60  Low                                         66 non-null     float64
 61  Close                                       66 non-null     float64
 62  Adj Close                                   66 non-null     float64
 63  Volume                                      66 non-null     float64
dtypes: datetime64[ns](2), float64(53), int64(9)
memory usage: 33.1 KB
```

Exploratory Data Analysis

Plotting Price Histories

In [184]:

```
def plot_price_history(ticker, volume=False):
    prices = pd.read_csv(f'../data/price_histories/{ticker}_history.csv',
                         index_col='Date',
                         parse_dates=True)
    fig, ax = mpf.plot(prices,
                       type='line',
                       volume=volume,
                       returnfig=True,
                       figscale=0.75)
    ax[0].set_title(f'{ticker} Price History')
```

In [185]:

```
plot_price_history('GOOGL')
```

In [186]:

```
plot_price_history('MSFT')
```

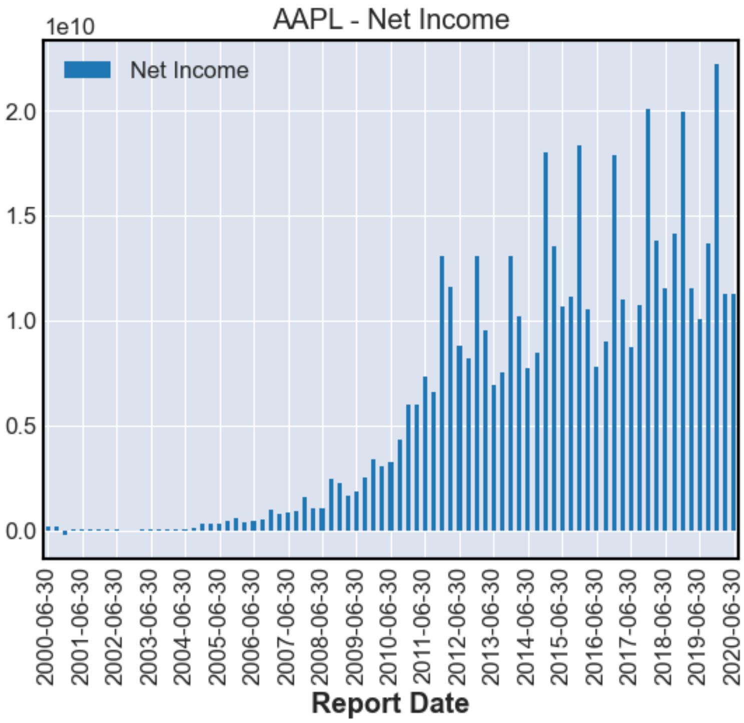
Plotting Fundamental Histories

In [236]:

```
def plot_fundamental_history(ticker, fundamental):
    df = pd.read_csv(f'../data/merged_data/{ticker}_merged.csv', index_col='Report Date')
    df = df[[fundamental]]
    df.plot(kind='bar')
    plt.xticks(ticks=np.arange(0, len(df), 4), labels=df.index[::4])
    plt.title(f'{ticker} - {fundamental}')
```

In [237]:

```
plot_fundamental_history('AAPL', 'Net Income')
```



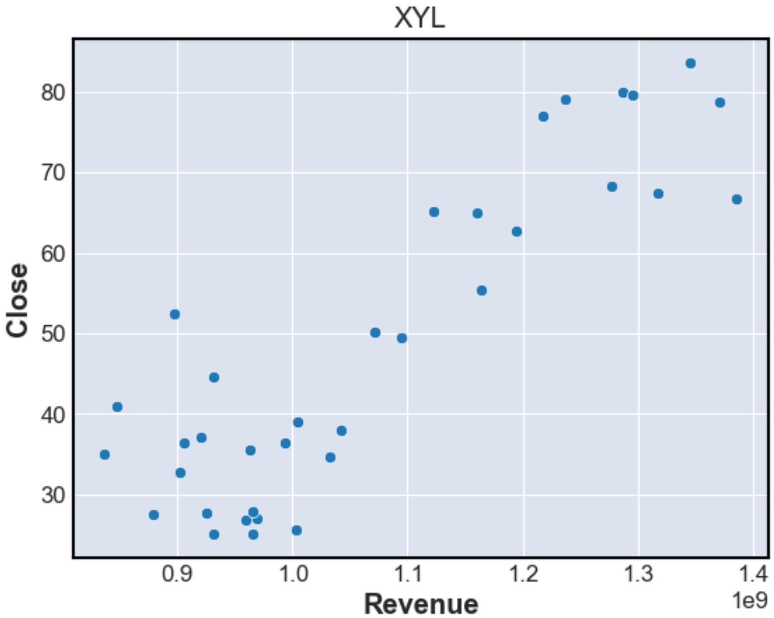
Scatter Plots of Price vs Fundamental

In [153]:

```
def plot_scatter(ticker, fundamental):
    df = pd.read_csv(f'../data/merged_data/{ticker}_merged.csv')
    fundamental = df[fundamental]
    price = df.Close
    sns.scatterplot(fundamental, price)
    plt.title(ticker)
```

In [187]:

```
plot_scatter('XYL', 'Revenue')
```



In [188]:

```
plot_scatter('MSFT', 'Research & Development')
```

Data Preprocessing

Handling Missing Values

The primary preprocessing need is handling missing values. Imputation is a somewhat risky methodology for this particular data, even with more advanced techniques such as iterative imputation. While not ideal, the safest route is to drop those columns which do not meet a certain minimum non-null threshold and then the remaining rows with null values thereafter. This threshold will be set at 30 in order to ensure there is enough data available for the model to work with. This will likely exclude some companies from being able to be modeled but this is acceptable for a proof of concept.

In [48]:

```
for i, ticker in enumerate(snp_tickers):
    # Manually catching 'BF.B'
    if ticker == 'BF.B':
        ticker = 'BF-B'

    # Loading in data
    df = pd.read_csv(f'../data/merged_data/{ticker}_merged.csv')

    # Dropping columns below the threshold
    threshold = 30
    to_drop = [col for col in df.columns if df[col].isna().sum() > len(df) - threshold]
    df = df.drop(columns=to_drop)

    # Dropping rows with remaining missing data
    df = df.dropna()

    # Exporting dataframe
    df.to_csv(f'../data/preprocessed_data/{ticker}_preprocessed.csv', index=False)

    # Printing progress
    print(f'{i+1}/{len(snp_tickers)} -- processed and saved {ticker.ljust(5)}', end='\r')
```

```
505/505 -- processed and saved ZTS
```

Number of Companies Excluded by Threshold Value

In [49]:

```
from pandas.errors import EmptyDataError
```

In [50]:

```
empty = []
for ticker in snp_tickers:
    if ticker == 'BF.B':
        ticker = 'BF-B'
    try:
        df = pd.read_csv(f'../data/preprocessed_data/{ticker}_preprocessed.csv')
    except EmptyDataError as e:
        empty.append(ticker)
```

In [51]:

```
len(empty)
```

Out[51]:

```
151
```