

graph

July 16, 2025

```
[ ]: import pandas as pd
import numpy as np
import os
```

1 Return Correlation

```
[ ]: log_returns = pd.read_csv("log_returns.csv")
log_returns.drop(columns = "Unnamed: 0", inplace = True)
log_returns.date = pd.to_datetime(log_returns.date)
log_returns.set_index("date", inplace = True)
rolling_corr = log_returns.rolling(window=5).corr()
rolling_corr
```

```
[ ]:
```

		NVDA	VLO	AAPL	JNJ	BA	AMZN	\
date								
2021-01-04	NVDA	NaN	NaN	NaN	NaN	NaN	NaN	
	VLO	NaN	NaN	NaN	NaN	NaN	NaN	
	AAPL	NaN	NaN	NaN	NaN	NaN	NaN	
	JNJ	NaN	NaN	NaN	NaN	NaN	NaN	
	BA	NaN	NaN	NaN	NaN	NaN	NaN	
...								
2025-05-30	BA	0.297966	0.020235	0.884133	0.588819	1.000000	0.797797	
	AMZN	0.487336	0.276038	0.894156	0.516270	0.797797	1.000000	
	TMO	0.183718	0.624478	0.896739	0.621013	0.652047	0.827441	
	TSLA	0.611365	0.117493	0.581519	0.228056	0.485031	0.871617	
	COST	-0.231704	0.731084	0.557233	0.502712	0.269147	0.303736	
		TMO	TSLA	COST				
date								
2021-01-04	NVDA	NaN	NaN	NaN				
	VLO	NaN	NaN	NaN				
	AAPL	NaN	NaN	NaN				
	JNJ	NaN	NaN	NaN				
	BA	NaN	NaN	NaN				
...								
2025-05-30	BA	0.652047	0.485031	0.269147				
	AMZN	0.827441	0.871617	0.303736				

TMO	1.000000	0.532582	0.785010
TSLA	0.532582	1.000000	-0.078498
COST	0.785010	-0.078498	1.000000

[9963 rows x 9 columns]

```
[ ]: flat_corr_data = []

for date, matrix in rolling_corr.groupby(level=0):
    matrix_df = matrix.droplevel(0)

    row = {'date': date}
    for i in sorted(matrix_df.columns):
        for j in sorted(matrix_df.index):
            if i < j:
                pair_name = f"{i}&{j}"
                row[pair_name] = matrix_df.at[j, i]

    flat_corr_data.append(row)

pairwise_corr_df = pd.DataFrame(flat_corr_data)

pairwise_corr_df = pairwise_corr_df.set_index('date')
pairwise_corr_df = pairwise_corr_df.reindex(sorted(pairwise_corr_df.columns),
axis=1)
pairwise_corr_df = pairwise_corr_df.dropna(axis = 0, how = "all")
```

```
[ ]: binary_corr_df = (pairwise_corr_df > 0.5).astype(int)
binary_corr_df
```

```
[ ]:
      AAPL&AMZN  AAPL&BA  AAPL&COST  AAPL&JNJ  AAPL&NVDA  AAPL&TMO  \
date
2021-01-11      1      0      1      0      1      1
2021-01-12      1      0      1      0      1      0
2021-01-13      1      0      1      1      0      0
2021-01-14      1      0      1      0      0      0
2021-01-15      1      0      1      0      0      0
...
2025-05-23      1      1      0      0      1      0
2025-05-27      1      1      1      1      1      1
2025-05-28      1      1      1      1      1      1
2025-05-29      1      1      1      0      0      1
2025-05-30      1      1      1      1      0      1

      AAPL&TSLA  AAPL&VLO  AMZN&BA  AMZN&COST  ...  JNJ&NVDA  JNJ&TMO  \
date
...
```

2021-01-11	1	0	1	1	...	0	0
2021-01-12	1	0	0	1	...	0	1
2021-01-13	1	0	1	1	...	1	1
2021-01-14	1	0	0	1	...	0	0
2021-01-15	1	0	0	1	...	0	0
...
2025-05-23	1	0	1	0	...	0	1
2025-05-27	1	1	1	1	...	0	1
2025-05-28	1	0	1	1	...	0	1
2025-05-29	1	0	1	1	...	0	1
2025-05-30	1	0	1	0	...	0	1

	JNJ&TSLA	JNJ&VLO	NVDA&TMO	NVDA&TSLA	NVDA&VLO	TMO&TSLA	\
date							
2021-01-11	0	1	0	0	0	1	
2021-01-12	0	0	0	0	0	0	
2021-01-13	0	0	1	0	0	0	
2021-01-14	0	1	1	0	0	0	
2021-01-15	0	1	0	0	0	0	
...	
2025-05-23	0	0	0	1	1	0	
2025-05-27	1	1	1	1	1	1	
2025-05-28	1	1	1	1	0	1	
2025-05-29	0	1	0	1	0	1	
2025-05-30	0	1	0	1	0	1	

	TMO&VLO	TSLA&VLO
date		
2021-01-11	0	0
2021-01-12	0	0
2021-01-13	0	0
2021-01-14	0	0
2021-01-15	0	0
...
2025-05-23	1	1
2025-05-27	1	1
2025-05-28	1	1
2025-05-29	0	1
2025-05-30	1	0

[1102 rows x 36 columns]

2 Sentiment Score Correlation

```
[ ]: tickers = ['AAPL', 'AMZN', 'BA', 'COST', 'JNJ', 'NVDA', 'TMO', 'TSLA', 'VLO']
data_dir = "/Data"
all_data = {}
min_length = float('inf')

for stock in tickers:
    sentiment_path = os.path.join(data_dir, f"{stock}_daily_sentiment_summary.
    ↪csv")
    sentiment_df = pd.read_csv(sentiment_path)
    sentiment_df['adjusted_date'] = pd.
    ↪to_datetime(sentiment_df['adjusted_date'])
    df = sentiment_df.copy()
    df.rename(columns = {"adjusted_date" : "date"}, inplace = True)
    all_data[stock] = df
```

```
[ ]: senti = all_data["AAPL"][["date", "mean_sentiment"]]
senti.rename(columns = {"mean_sentiment": "AAPL"}, inplace = True)
for ticker in ['AMZN', 'BA', 'COST', 'JNJ', 'NVDA', 'TMO', 'TSLA', 'VLO']:
    stock_senti = all_data[ticker][["date", "mean_sentiment"]]
    stock_senti.rename(columns = {"mean_sentiment" : f"{ticker}"}, inplace =
    ↪True)
    senti = pd.merge(senti, stock_senti, on = "date", how = "left")
```

/var/folders/w_/wzxdnvq13mxbxkhjgg1mtty_80000gn/T/ipykernel_63466/1560417984.py:2

: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
senti.rename(columns = {"mean_sentiment": "AAPL"}, inplace = True)
```

/var/folders/w_/wzxdnvq13mxbxkhjgg1mtty_80000gn/T/ipykernel_63466/1560417984.py:5

: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
stock_senti.rename(columns = {"mean_sentiment" : f"{ticker}"}, inplace = True)
```

/var/folders/w_/wzxdnvq13mxbxkhjgg1mtty_80000gn/T/ipykernel_63466/1560417984.py:5

: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
stock_senti.rename(columns = {"mean_sentiment" : f"{ticker}"}, inplace = True)
```

```
/var/folders/w_/wzxdnvq13mxbxkhjgg1mty_80000gn/T/ipykernel_63466/1560417984.py:5
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
stock_senti.rename(columns = {"mean_sentiment" : f"{ticker}"}, inplace = True)
/var/folders/w_/wzxdnvq13mxbxkhjgg1mty_80000gn/T/ipykernel_63466/1560417984.py:5
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
stock_senti.rename(columns = {"mean_sentiment" : f"{ticker}"}, inplace = True)
/var/folders/w_/wzxdnvq13mxbxkhjgg1mty_80000gn/T/ipykernel_63466/1560417984.py:5
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
stock_senti.rename(columns = {"mean_sentiment" : f"{ticker}"}, inplace = True)
/var/folders/w_/wzxdnvq13mxbxkhjgg1mty_80000gn/T/ipykernel_63466/1560417984.py:5
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
stock_senti.rename(columns = {"mean_sentiment" : f"{ticker}"}, inplace = True)
/var/folders/w_/wzxdnvq13mxbxkhjgg1mty_80000gn/T/ipykernel_63466/1560417984.py:5
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
stock_senti.rename(columns = {"mean_sentiment" : f"{ticker}"}, inplace = True)
/var/folders/w_/wzxdnvq13mxbxkhjgg1mty_80000gn/T/ipykernel_63466/1560417984.py:5
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
stock_senti.rename(columns = {"mean_sentiment" : f"{ticker}"}, inplace = True)
```

```
[ ]: senti = senti.fillna(0)
senti.set_index("date", inplace = True)
senti_corr = senti.rolling(window=5).corr()
```

```
[ ]: flat_corr_senti = []

for date, matrix in senti_corr.groupby(level=0):
    matrix_df = matrix.droplevel(0)

    row = {'date': date}
    for i in sorted(matrix_df.columns):
        for j in sorted(matrix_df.index):
            if i < j:
                pair_name = f"{i}&{j}"
                row[pair_name] = matrix_df.at[j, i]

    flat_corr_senti.append(row)

pairwise_corr_senti = pd.DataFrame(flat_corr_senti)
pairwise_corr_senti = pairwise_corr_senti.set_index('date')
pairwise_corr_senti = pairwise_corr_senti.reindex(sorted(pairwise_corr_senti.
↳ columns), axis=1)
```

```
[ ]: pairwise_corr_senti = pairwise_corr_senti.fillna(0)
pairwise_corr_senti = pairwise_corr_senti[pairwise_corr_senti.index.
↳ isin(binary_corr_df.index)]
binary_corr_senti = (pairwise_corr_senti > 0.5).astype(int)
binary_corr_senti
```

```
[ ]:
```

	AAPL&AMZN	AAPL&BA	AAPL&COST	AAPL&JNJ	AAPL&NVDA	AAPL&TMO	\
date							
2021-01-11	0	1	0	0	1	0	
2021-01-12	0	1	0	0	1	0	
2021-01-13	0	0	1	1	0	0	
2021-01-14	0	0	1	0	0	0	
2021-01-15	0	0	1	0	0	0	
...	
2025-05-23	0	0	0	0	0	0	
2025-05-27	0	0	0	0	0	0	
2025-05-28	0	0	0	0	0	0	
2025-05-29	0	0	0	0	0	0	
2025-05-30	0	0	0	0	0	0	

	AAPL&TSLA	AAPL&VLO	AMZN&BA	AMZN&COST	...	JNJ&NVDA	JNJ&TMO	\
date					...			
2021-01-11	1	1	1	0	...	0	0	
2021-01-12	0	1	1	0	...	0	0	
2021-01-13	0	0	0	0	...	0	0	
2021-01-14	0	0	1	0	...	0	0	
2021-01-15	0	0	1	0	...	1	0	

...
2025-05-23	0	0	0	0	...	0	0
2025-05-27	0	0	0	0	...	0	0
2025-05-28	0	0	0	0	...	0	0
2025-05-29	0	0	0	0	...	0	0
2025-05-30	1	0	0	0	...	0	0

	JNJ&TSLA	JNJ&VLO	NVDA&TMO	NVDA&TSLA	NVDA&VLO	TMO&TSLA	\
date							
2021-01-11	0	0	0	1	0	0	
2021-01-12	0	0	0	1	0	0	
2021-01-13	0	0	0	1	0	0	
2021-01-14	0	0	0	1	0	0	
2021-01-15	0	0	0	0	0	0	
...	
2025-05-23	1	0	0	0	0	0	
2025-05-27	1	0	0	0	0	0	
2025-05-28	1	0	0	0	0	0	
2025-05-29	1	0	0	0	0	0	
2025-05-30	1	0	0	0	0	0	

	TMO&VLO	TSLA&VLO
date		
2021-01-11	0	1
2021-01-12	0	0
2021-01-13	0	0
2021-01-14	0	0
2021-01-15	0	0
...
2025-05-23	0	0
2025-05-27	0	0
2025-05-28	0	0
2025-05-29	0	0
2025-05-30	0	0

[1102 rows x 36 columns]

3 Sector

```
[ ]: membership_df = pd.read_csv("Stock_Membership.csv")
membership_df.set_index("date", inplace = True)
membership_df.index = pd.to_datetime(membership_df.index)
membership_df = pd.merge(binary_corr_senti["AAPL&AMZN"],membership_df, left_on=
    ↳ binary_corr_senti.index, right_on = membership_df.index, how = "left" )
membership_df.drop(columns = "AAPL&AMZN_x", inplace = True)
membership_df.bfill(inplace = True)
```

```
membership_df.rename(columns={"key_0": "date", "AAPL&AMZN_y": "AAPL&AMZN"}, inplace=True)
membership_df.set_index("date", inplace = True)
membership_df.index = pd.to_datetime(membership_df.index)
membership_df
```

```
[ ]:
      AAPL&AMZN  AAPL&BA  AAPL&COST  AAPL&JNJ  AAPL&NVDA  AAPL&TMO  \
date
2021-01-11      0.0      0.0      0.0      0.0      1.0      0.0
2021-01-12      0.0      0.0      0.0      0.0      1.0      0.0
2021-01-13      0.0      0.0      0.0      0.0      1.0      0.0
2021-01-14      0.0      0.0      0.0      0.0      1.0      0.0
2021-01-15      0.0      0.0      0.0      0.0      1.0      0.0
...
2025-05-23      0.0      0.0      0.0      0.0      1.0      0.0
2025-05-27      0.0      0.0      0.0      0.0      1.0      0.0
2025-05-28      0.0      0.0      0.0      0.0      1.0      0.0
2025-05-29      0.0      0.0      0.0      0.0      1.0      0.0
2025-05-30      0.0      0.0      0.0      0.0      1.0      0.0
```

```
      AAPL&TSLA  AAPL&VLO  AMZN&BA  AMZN&COST  ...  JNJ&NVDA  JNJ&TMO  \
date
2021-01-11      0.0      0.0      0.0      0.0  ...      0.0      1.0
2021-01-12      0.0      0.0      0.0      0.0  ...      0.0      1.0
2021-01-13      0.0      0.0      0.0      0.0  ...      0.0      1.0
2021-01-14      0.0      0.0      0.0      0.0  ...      0.0      1.0
2021-01-15      0.0      0.0      0.0      0.0  ...      0.0      1.0
...
2025-05-23      0.0      0.0      0.0      0.0  ...      0.0      1.0
2025-05-27      0.0      0.0      0.0      0.0  ...      0.0      1.0
2025-05-28      0.0      0.0      0.0      0.0  ...      0.0      1.0
2025-05-29      0.0      0.0      0.0      0.0  ...      0.0      1.0
2025-05-30      0.0      0.0      0.0      0.0  ...      0.0      1.0
```

```
      JNJ&TSLA  JNJ&VLO  NVDA&TMO  NVDA&TSLA  NVDA&VLO  TMO&TSLA  \
date
2021-01-11      0.0      0.0      0.0      0.0      0.0      0.0
2021-01-12      0.0      0.0      0.0      0.0      0.0      0.0
2021-01-13      0.0      0.0      0.0      0.0      0.0      0.0
2021-01-14      0.0      0.0      0.0      0.0      0.0      0.0
2021-01-15      0.0      0.0      0.0      0.0      0.0      0.0
...
2025-05-23      0.0      0.0      0.0      0.0      0.0      0.0
2025-05-27      0.0      0.0      0.0      0.0      0.0      0.0
2025-05-28      0.0      0.0      0.0      0.0      0.0      0.0
2025-05-29      0.0      0.0      0.0      0.0      0.0      0.0
2025-05-30      0.0      0.0      0.0      0.0      0.0      0.0
```


	TMO&VLO	TSLA&VLO
date		
2021-01-11	0.0	0.0
2021-01-12	0.0	0.0
2021-01-13	0.0	0.0
2021-01-14	0.0	0.0
2021-01-15	0.0	0.0
...
2025-05-23	0.0	0.0
2025-05-27	0.0	0.0
2025-05-28	0.0	0.0
2025-05-29	0.0	0.0
2025-05-30	0.0	0.0

[1102 rows x 36 columns]

4 Take Union to Get the Final Dynamic Graph

```
[ ]: cols = membership_df.columns
combined = pd.DataFrame(np.maximum(binary_corr_senti[cols],
    ↪binary_corr_df[cols], membership_df[cols]),
                        index=binary_corr_senti.index, columns=cols)
combined
```

```
[ ]: AAPL&AMZN AAPL&BA AAPL&COST AAPL&JNJ AAPL&NVDA AAPL&TMO \
date
2021-01-11 1.0 1.0 1.0 0.0 1.0 1.0
2021-01-12 1.0 1.0 1.0 0.0 1.0 0.0
2021-01-13 1.0 0.0 1.0 1.0 0.0 0.0
2021-01-14 1.0 0.0 1.0 0.0 0.0 0.0
2021-01-15 1.0 0.0 1.0 0.0 0.0 0.0
...
2025-05-23 1.0 1.0 0.0 0.0 1.0 0.0
2025-05-27 1.0 1.0 1.0 1.0 1.0 1.0
2025-05-28 1.0 1.0 1.0 1.0 1.0 1.0
2025-05-29 1.0 1.0 1.0 0.0 0.0 1.0
2025-05-30 1.0 1.0 1.0 1.0 0.0 1.0

AAPL&TSLA AAPL&VLO AMZN&BA AMZN&COST ... JNJ&NVDA JNJ&TMO \
date
2021-01-11 1.0 1.0 1.0 1.0 ... 0.0 0.0
2021-01-12 1.0 1.0 1.0 1.0 ... 0.0 1.0
2021-01-13 1.0 0.0 1.0 1.0 ... 1.0 1.0
2021-01-14 1.0 0.0 1.0 1.0 ... 0.0 0.0
2021-01-15 1.0 0.0 1.0 1.0 ... 1.0 0.0
```

```

...
2025-05-23      1.0      0.0      1.0      0.0 ...      0.0      1.0
2025-05-27      1.0      1.0      1.0      1.0 ...      0.0      1.0
2025-05-28      1.0      0.0      1.0      1.0 ...      0.0      1.0
2025-05-29      1.0      0.0      1.0      1.0 ...      0.0      1.0
2025-05-30      1.0      0.0      1.0      0.0 ...      0.0      1.0

```

```

      JNJ&TSLA  JNJ&VLO  NVDA&TMO  NVDA&TSLA  NVDA&VLO  TMO&TSLA  \
date
2021-01-11      0.0      1.0      0.0      1.0      0.0      1.0
2021-01-12      0.0      0.0      0.0      1.0      0.0      0.0
2021-01-13      0.0      0.0      1.0      1.0      0.0      0.0
2021-01-14      0.0      1.0      1.0      1.0      0.0      0.0
2021-01-15      0.0      1.0      0.0      0.0      0.0      0.0
...
2025-05-23      1.0      0.0      0.0      1.0      1.0      0.0
2025-05-27      1.0      1.0      1.0      1.0      1.0      1.0
2025-05-28      1.0      1.0      1.0      1.0      0.0      1.0
2025-05-29      1.0      1.0      0.0      1.0      0.0      1.0
2025-05-30      1.0      1.0      0.0      1.0      0.0      1.0

```

```

      TMO&VLO  TSLA&VLO
date
2021-01-11      0.0      1.0
2021-01-12      0.0      0.0
2021-01-13      0.0      0.0
2021-01-14      0.0      0.0
2021-01-15      0.0      0.0
...
2025-05-23      1.0      1.0
2025-05-27      1.0      1.0
2025-05-28      1.0      1.0
2025-05-29      0.0      1.0
2025-05-30      1.0      0.0

```

```
[1102 rows x 36 columns]
```

```
[ ]: combined.to_csv("graph_final.csv")
```