Data Assignment 1 - `t54zheng` (20939203)

Task 2 - Basic Statistics

In [6]:
```
# imports

import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats
from datetime import timedelta
from math import sqrt

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

In [2]:
```
# Import raw data
data_file = "djreturns.xlsx"

dj27 = pd.read_excel(data_file, sheet_name="dj27")
individual_dj27_returns = pd.read_excel(data_file, sheet_name="returns")
sp500_returns = pd.read_excel(data_file, sheet_name="sp500")
```

Task 2 - *Basic Statistics*

For each of the 27 stocks in `dj27`, and the market return `sp500_returns`, we want to find these statistics on their returns:

- The arithmetic mean
- standard deviation
- skewness
- kurtosis

In [3]:
```
# We need to group the data in individual_dj27_returns based on stock.
# Note that using PERMNO is a better idea since COMNAM can change.

# Let's show that our data actually has this issue:
duplicate_comnam_df = individual_dj27_returns[["PERMNO", "COMNAM"]].drop_duplicates().groupby("PERMNO").agg({'COMNAM': lambda x: list(x)})
duplicate_comnam_df
```

Out[3]:

PERMNO	COMNAM
10107	[MICROSOFT CORP]
10145	[HONEYWELL INTERNATIONAL INC]
11308	[COCA COLA CO]
12490	[INTERNATIONAL BUSINESS MACHS COR]
14008	[AMGEN INC]
14541	[CHEVRON CORP, CHEVRONTEXACO CORP, CHEVRON COR...
14593	[APPLE COMPUTER INC, APPLE INC]
18163	[PROCTER & GAMBLE CO]
18542	[CATERPILLAR INC]
19502	[WALGREEN CO, WALGREENS BOOTS ALLIANCE INC]
19561	[BOEING CO]
22111	[JOHNSON & JOHNSON]
22592	[MINNESOTA MINING & MFG CO, 3M CO]
22752	[MERCK & CO INC, MERCK & CO INC NEW]
26403	[DISNEY WALT CO]
43449	[MCDONALDS CORP]
47896	[CHASE MANHATTAN CORP NEW, J P MORGAN CHASE & ...
55976	[WAL MART STORES INC, WALMART INC]
57665	[NIKE INC]
59176	[AMERICAN EXPRESS CO]
59328	[INTEL CORP]
59459	[ST PAUL COS INC, ST PAUL TRAVELERS COS INC, T...
65875	[BELL ATLANTIC CORP, VERIZON COMMUNICATIONS INC]
66181	[HOME DEPOT INC]
76076	[CISCO SYSTEMS INC]
86868	[GOLDMAN SACHS GROUP INC]
92655	[UNITED HEALTHCARE CORP, UNITEDHEALTH GROUP INC]

We see many securities have multiple comnames as they have changed their company name over the period of the data, but PERMNO remains the same.

In [4]:
```
# So let's make a new dataframe for each PERMNO we have in dj27, and store them in a dict by PERMNO.
returns_dict = {} # permno -> dataframe(permno_returns)
```

```
permnos = dj27["PERMNO"]
for permno in permnos:
    returns_df = individual_dj27_returns[individual_dj27_returns["PERMNO"] == permno]
    returns_dict[permno] = returns_df
```

In [17]:

```
# Now that we have our data nicely organized, let's make a new dataframe to present our statistics
# We'll have every row describes the statistics for each return

stats_df = pd.DataFrame(columns=["permno", "Common Name(s)", "Mean (%)", "Standard Deviation (%)", "Skewness", "Kurtosis"])

# add using .loc[-1]
# First add the stats for the market portfolio
market_stats = {
    "permno": "market",
    "Common Name(s)": ["Market"],
    "Mean (%)": sp500_returns["SPRTRN"].mean(),
    "Standard Deviation (%)": sp500_returns["SPRTRN"].std(),
    "Skewness": stats.skew(sp500_returns["SPRTRN"], bias=False), # bias=False means we calculate biased estimator (since sample)
    "Kurtosis": stats.kurtosis(sp500_returns["SPRTRN"], bias=False, fisher=False)
    }

# stats_df = stats_df.append(market_stats, ignore_index=True)
stats_df.loc[0] = [v for v in market_stats.values()]

# Now we'll add the rest of the securities from dj27
duplicate_comnam_dict = duplicate_comnam_df.to_dict()['COMNAM']

i = 1
for permno, df in returns_dict.items():
    permno_stats = {
        "permno": permno,
        "Common Name(s)": duplicate_comnam_dict[permno],
        "Mean (%)": df["RET"].mean(),
        "Standard Deviation (%)": df["RET"].std(),
        "Skewness": stats.skew(df["RET"], bias=False),
        "Kurtosis": stats.kurtosis(df["RET"], bias=False, fisher=False)
    }
    # stats_df = stats_df.append(permno_stats, ignore_index=True)
    stats_df.loc[i] = [v for v in permno_stats.values()]
    i += 1

# Now we need to annualize the mean and standard deviation of the returns (currently monthly)
stats_df["Mean (%)"] = stats_df["Mean (%)"] * 12 # Annualize by multiplying by 12 (no compounding)
stats_df["Standard Deviation (%)"] = stats_df["Standard Deviation (%)"] * sqrt(12) # Annualizing stdev

# Format the results
stats_df["Mean (%)"] *= 100
stats_df["Standard Deviation (%)"] *= 100

# Round to 4 decimal places
stats_df = stats_df.round(4)
stats_df
```

Out[17]:

	permno	Common Name(s)	Mean (%)	Standard Deviation (%)	Skewness	Kurtosis
0	market	[Market]	6.5021	15.0357	-0.5343	4.1287
1	10107	[MICROSOFT CORP]	14.1157	28.4694	0.2172	6.4101
2	10145	[HONEYWELL INTERNATIONAL INC]	12.5522	28.6748	-0.1329	10.9648
3	11308	[COCA COLA CO]	7.5283	17.6022	-0.5042	4.2233
4	12490	[INTERNATIONAL BUSINESS MACHS COR]	6.5405	25.0215	0.4145	6.6475
5	14008	[AMGEN INC]	10.3270	25.4240	0.5245	4.8601
6	14541	[CHEVRON CORP, CHEVRONTEXACO CORP, CHEVRON COR...	10.7444	22.6451	0.4094	5.3366
7	14593	[APPLE COMPUTER INC, APPLE INC]	33.2421	39.5911	-0.6364	6.5268
8	18163	[PROCTER & GAMBLE CO]	9.3172	17.7001	-1.5145	12.3682
9	18542	[CATERPILLAR INC]	17.3336	30.6904	-0.0676	5.0133
10	19502	[WALGREEN CO, WALGREENS BOOTS ALLIANCE INC]	7.7165	26.1702	0.3545	3.7247
11	19561	[BOEING CO]	14.4704	31.8393	-0.3117	7.5503
12	22111	[JOHNSON & JOHNSON]	9.8556	16.5092	-0.1837	4.3062
13	22592	[MINNESOTA MINING & MFG CO, 3M CO]	10.4157	20.0074	-0.0242	3.6496
14	22752	[MERCK & CO INC, MERCK & CO INC NEW]	7.4162	23.8799	-0.2528	4.2859
15	26403	[DISNEY WALT CO]	12.0839	25.8382	0.0231	4.5581
16	43449	[MCDONALDS CORP]	13.0400	19.4611	-0.4955	5.4100
17	47896	[CHASE MANHATTAN CORP NEW, J P MORGAN CHASE & ...	12.4934	29.8458	-0.2406	4.2515
18	55976	[WAL MART STORES INC, WALMART INC]	6.9605	18.9113	-0.2630	4.1381
19	57665	[NIKE INC]	19.8069	26.3183	-0.1524	8.5396
20	59176	[AMERICAN EXPRESS CO]	11.4081	31.7457	2.7701	31.7846
21	59328	[INTEL CORP]	9.0218	33.1277	-0.5344	5.5981
22	59459	[ST PAUL COS INC, ST PAUL TRAVELERS COS INC, T...	12.4100	24.5435	1.4497	15.5440
23	65875	[BELL ATLANTIC CORP, VERIZON COMMUNICATIONS INC]	6.5369	21.9020	0.8740	8.6004
24	66181	[HOME DEPOT INC]	13.3259	25.0232	-0.1948	3.4422
25	76076	[CISCO SYSTEMS INC]	7.8689	33.3501	-0.1939	5.2589
26	86868	[GOLDMAN SACHS GROUP INC]	12.6296	31.9316	0.1213	3.7759
27	92655	[UNITED HEALTHCARE CORP, UNITEDHEALTH GROUP INC]	23.7253	24.4657	-0.6260	5.7142

In [18]:
```
# The market portfolio
stats_df[stats_df["permno"] == "market"]
```

Out[18]:

	permno	Common Name(s)	Mean (%)	Standard Deviation (%)	Skewness	Kurtosis
0	market	[Market]	6.5021	15.0357	-0.5343	4.1287

Statistics of the market portfolio

Here's a table comparing the sample skewness and kurtosis of the market returns to a normal distribution

	Normal Distribution	Market	Interpretation
Skewness	0	-0.5343	`Negative Skew`
Kurtosis	3	4.1287	`High Kurtosis`

Note that we calculated the kurtosis using the Pearson definition, which defaults the normal distribution to 3

Skewness: Since the market's returns are negatively skewed, the returns are more heavily distributed towards the right, above zero. We can expect the market to return positive returns more often than if the market returns were perfectly normally distributed

Kurtosis: Since the market's returns have a high kurtosis (> 3), meaning that the distribution of market returns have fatter tails and a sharper peak about the mean, implying that market returns are more risky than if it were normally distributed.

In [ ]: