Introduction

This report details the process of analysing historical market data for stocks in the S&P 500 index using Principal Component Analysis (PCA). The primary objective is to identify the main underlying factors that explain the variance in stock returns. The analysis involves several stages: data loading and cleaning, filtering to a consistent set of stocks and dates, transforming prices into returns, and finally, applying PCA to both raw and normalised returns data to interpret the results.

Task 1: Loading the Data

The first step was to load the provided `stockdata.csv` file into a pandas DataFrame, a standard structure for data manipulation in Python. This allows for efficient handling of the tabular stock data.

```
In [1]: import pandas as pd

        df = pd.read_csv('stockdata.csv')
```

Task 2: Initial Data Exploration

To understand the scope of the dataset, the unique stock tickers (names) were identified and counted. The list of names was sorted alphabetically to provide a structured overview.

```
In [3]: # Task 2a: Identify and sort unique stock names
        # We select the 'Name' column, get the unique values, and convert it to
        all_names = sorted(df['Name'].unique())

        # Task 2b: Count how many unique names there are
        num_names = len(all_names)
        print(f"Task 2b: There are {num_names} unique stock names.")
```

Task 3: Filtering Stocks by Date Range

The dataset was filtered to only include stocks with a complete trading history for the specified 3-year period: 1st November 2019 to 31st October 2022. This ensures that all stocks in the subsequent analysis are comparable over the same timeframe, removing stocks that started trading late or ceased trading early.

```
Task 3b: The following names were removed:
['ABNB', 'CARR', 'CEG', 'GEHC', 'GEV', 'KVUE', 'OTIS', 'PLTR', 'SOLV',
'VLTO']

Task 3c: There are 490 names left after filtering.
```

Task 4: Filtering for Common Dates

To ensure data integrity for multivariate analysis, the date range was further refined to include only those dates for which all 481 remaining stocks had trading data. This removes days that were holidays for some exchanges or days where specific stocks experienced trading halts.

```
Task 4c: There are 755 common trading dates left.

Task 4d: The first 5 dates are:
['2019-11-01', '2019-11-04', '2019-11-05', '2019-11-06', '2019-11-07']

Task 4d: The last 5 dates are:
['2022-10-25', '2022-10-26', '2022-10-27', '2022-10-28', '2022-10-31']
```

Task 5: Pivoting the Data

The data was transformed from a "long" format (one row per stock per day) to a "wide" format. The resulting DataFrame has dates as its index, stock tickers as columns, and the corresponding closing prices as its values. This matrix structure is a prerequisite for PCA.

```python
# Now, pivot this final dataframe to get the desired structure.
# index='date': makes the 'date' column the new row labels.
# columns='Name': makes the unique values in the 'Name' column the new column headers.
# values='close': fills the table with the corresponding 'close' prices.
close_prices_df = final_df.pivot(index='date', columns='Name', values='close')
```

Task 6: Calculating Returns

Stock prices were converted to daily percentage returns using the formula `(today - yesterday) / yesterday`. Returns are a scale-free measure of performance and are more stationary than prices, making them better suited for statistical analysis.

Task 7: Performing PCA on Raw Returns

PCA was applied to the returns DataFrame. This technique identifies the orthogonal components (uncorrelated factors) that capture the maximum amount of variance in the data. The top components represent the most dominant patterns in the market's movements.

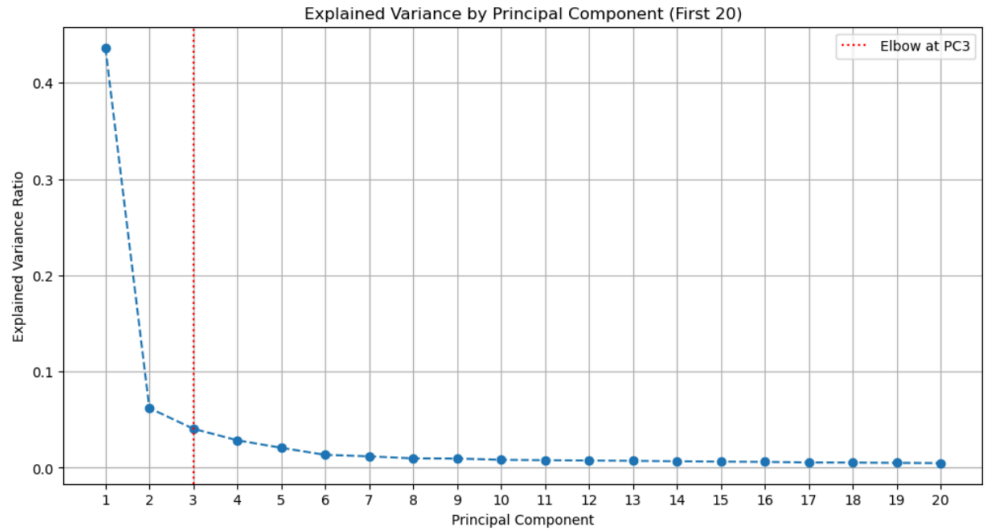
```
pca.fit(returns_df)

# (7b) Print the top five PCs.
# The PCs are stored in the '.components_' attribute of the fitted pca object.
# They are already sorted by importance (eigenvalue), so the first row is PC1, the second is PC2, etc
# We'll create a new DataFrame to display them nicely with labels.
top_five_pcs = pd.DataFrame(
    pca.components_[:5],
    columns=returns_df.columns,
    index=[f'PC{i+1}' for i in range(5)]
)
```

Task 8: Analyzing Explained Variance

To evaluate the significance of each principal component, the explained variance ratios were extracted from the fitted PCA model. These ratios indicate the proportion of the dataset's total variance that each component captures. The first principal component was found to account for approximately **43.65%** of the total variance, highlighting it as the single most dominant factor in market returns. A scree plot was generated to visualise the explained variance of the first 20 components. Using the 'elbow method' to identify the point of diminishing returns, an elbow was marked at the third principal component. This suggests that the first three components collectively capture the most substantial and structurally important information, while subsequent components explain progressively smaller, less significant portions of the variance.

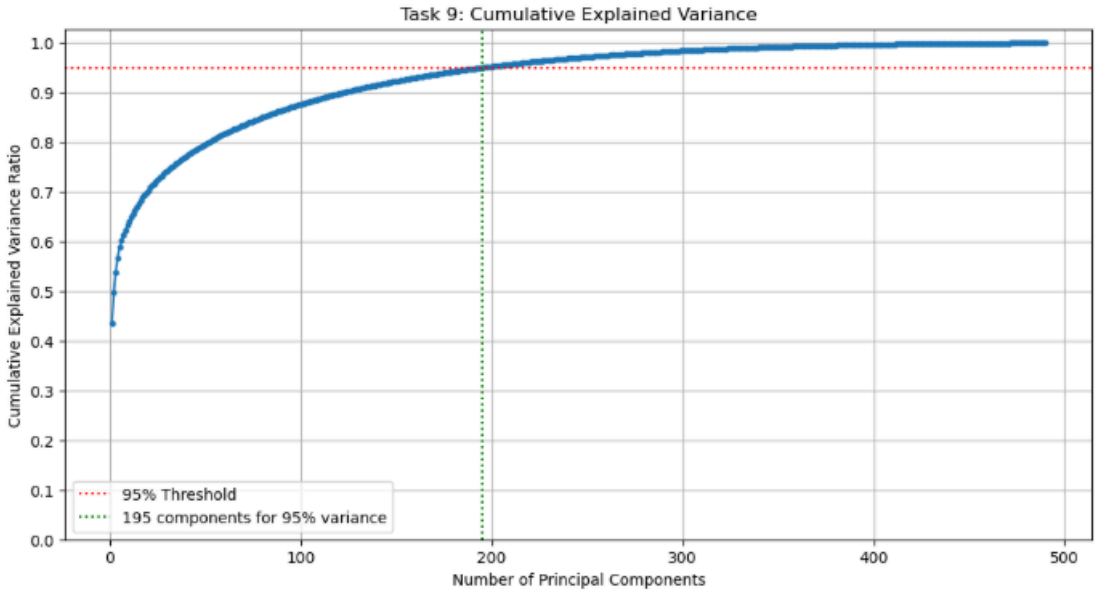
```
Task 8b: The first principal component explains 43.65% of the variance.
```



Task 9: Cumulative Variance Analysis

While Task 8 identified the importance of individual components, this task assesses their collective power. The cumulative sum of the explained variance ratios was calculated to determine how many components are required to capture a specified percentage of the total market variance. This is a key step in dimensionality reduction, as it quantifies the trade-off between model complexity (number of components) and information retention.

The goal was to find the minimum number of components needed to explain at least 95% of the total variance in the stock returns.

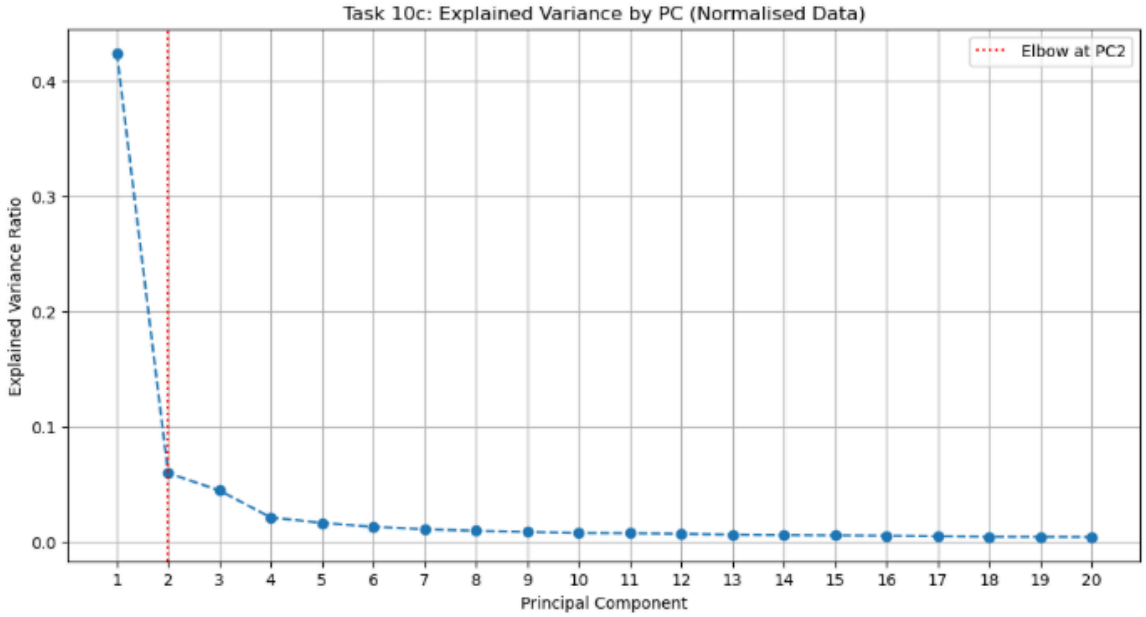


Task 10: PCA on Normalised Data

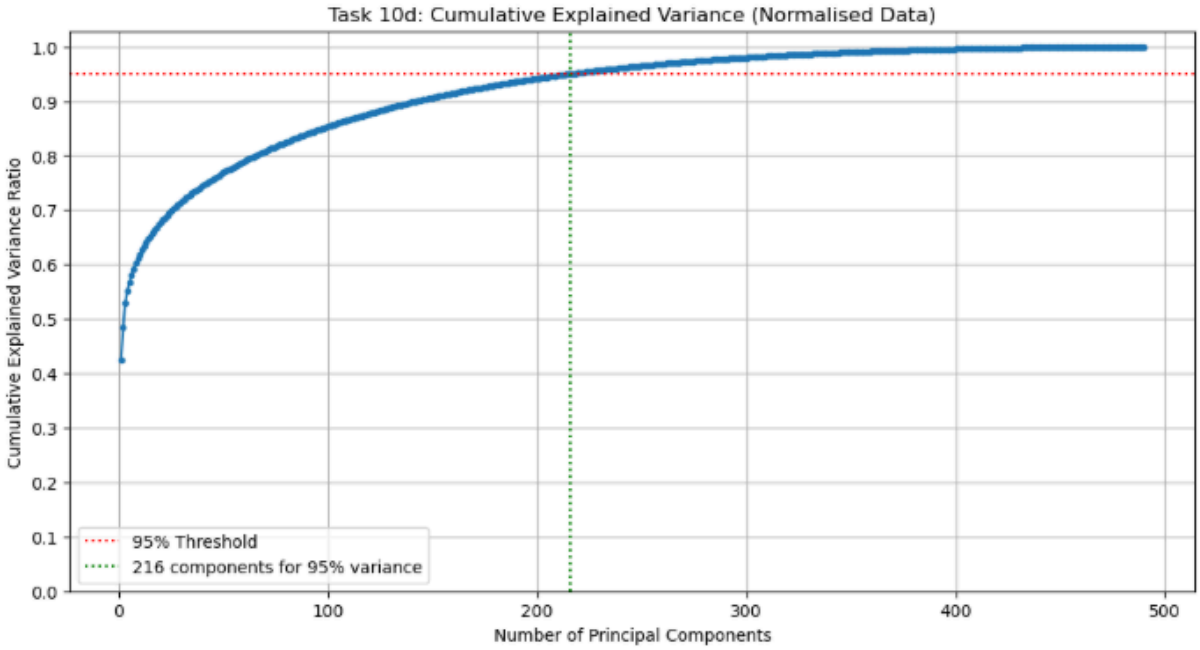
The final task was to repeat the entire PCA process on normalised returns data. PCA is sensitive to the variance of the initial variables; without normalisation, stocks with higher intrinsic volatility would disproportionately influence the principal components. By using `StandardScaler` to transform each stock's returns to have a mean of zero and unit variance, we ensure that each stock contributes equally to the analysis based on its correlation with other stocks, not its volatility.

This allows for the identification of underlying factors that are not biased by the scale of individual stock returns, typically leading to a more robust and interpretable model.

Task 10c: The first PC (normalised) explains 42.42% of the variance.



Task 10d: 216 components are needed for 95% variance on normalised dat a.



Conclusion:

This assignment successfully demonstrated the use of Principal Component Analysis to simplify complex S&P 500 stock data into its core underlying factors. Through a rigorous process of data cleaning and transformation, the analysis revealed that the daily movements of hundreds of stocks are largely driven by a much smaller set of shared components, with the first component clearly representing the overall market trend.

Crucially, the project highlighted the critical importance of data normalisation. Standardizing the stock returns before the analysis resulted in a more robust and efficient model, allowing the underlying market structure to be identified more clearly. This exercise provided a practical demonstration of how dimensionality reduction can extract meaningful signals from noisy financial data, a core principle in modern risk management and portfolio theory.