**Table of Contents**

[Section 1: Introduction 3](#_Toc57544)

[Section 2: Data Source Description 3](#_Toc57545)

[Section 3: Business Requirements and Scope 6](#_Toc57546)

[3.1 Business Context 6](#_Toc57547)

[3.2 Project Objectives 6](#_Toc57548)

[3.3 Business Questions to Address 7](#_Toc57549)

[Section 4: BI Solution Design and Architecture 7](#_Toc57550)

[4.1 Overview of BI Architecture 7](#_Toc57551)

[4.2 BI Architecture Diagram 8](#_Toc57552)

[4.3 Components Used in Each Layer 8](#_Toc57553)

[4.4 System Design Considerations 9](#_Toc57554)

[Section 5: ETL Process 9](#_Toc57555)

[5.1 Extract 9](#_Toc57556)

[5.2 Transform 10](#_Toc57557)

[5.3 Load 11](#_Toc57558)

[5.4 Python Libraries and Files Used 12](#_Toc57559)

[Section 6: Data Modelling and Star Schema Design 12](#_Toc57560)

[6.1 Overview of the Star Schema 12](#_Toc57561)

[6.2 Entity Relationship Diagram 13](#_Toc57562)

[6.3 Table Descriptions 14](#_Toc57563)

[6.4 Benefits of the Star Schema 15](#_Toc57564)

[Section 7: Power BI Reporting and Insights 16](#_Toc57565)

[7.1 Monthly Return by Sector (2013–2018) 16](#_Toc57566)

[7.2 Interest Rates vs Avg Monthly Return (2013–2018) 17](#_Toc57567)

[7.3 Top 10 & Bottom 10 Performing Stocks (2013–2018) 18](#_Toc57568)

[7.4 Top Industries by Avg Monthly Return 18](#_Toc57569)

[7.5 Sector Trading Volume by Year (2013–2018) 19](#_Toc57570)

[8. Key Findings and Business Implications 20](#_Toc57571)

[9. Evaluation 22](#_Toc57572)

[10. Conclusion 24](#_Toc57573)

[11. References 25](#_Toc57574)

[12. Appendix 26](#_Toc57575)

# Section 1: Introduction

The financial services industry underwent a major transformation because of expanding data availability combined with growing technological reliance for insights. Fidelity Investments along with other investment firms must implement advanced Business Intelligence tools to enable analysts and fund managers to extract timely actionable market insights from historical and real-time data.

Fidelity operates in a fast-paced environment where investment decisions are influenced by a complex interplay of factors — including stock price movements, trading volume, and macroeconomic indicators such as interest rates and inflation. The requirement for integrated dynamic Business Intelligence solutions has become essential because analysts previously depended on spreadsheets and static reports.

This project develops a contemporary Business Intelligence system to track stock asset performance along with sector trends during the five-year period from 2013 through 2018. The system was designed to:

☑ Track stock performance at monthly granularity

☑ Compare sector and industry returns

☑ Highlight top and bottom performing stocks

☑ Analyse trading volume and liquidity

☑ Overlay stock performance with macroeconomic indicators

The solution development process is fully explained through this report which covers data acquisition and ETL processing and relational MySQL modelling and Power BI dashboard presentation. The main objectives of this project consist of two parts.

* The project aims to show how contemporary data tools enable the creation of a dependable business intelligence system which supports data-driven decision-making.
* The system provides various user-friendly interactive visualizations which enable investment analysts to track asset performance while evaluating macroeconomic effects efficiently through actionable insights.

# Section 2: Data Source Description

The project draws its information from three primary datasets which together offer complete historical stock behavior data and company metadata and macroeconomic context. The business intelligence pipeline uses these datasets to calculate financial performance metrics while also supporting sector-based and economic overlays.

### 2.1 Overview of Datasets Used

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset Name** | **Description** | **Format** | **Source** |
| all\_stocks\_5yr.csv | Daily stock price data for over  500 companies from 2013–2018 | CSV | Kaggle: US Price Volume  Dataset |
| cleaned\_nasdaq\_screener. xlsx | Company metadata including sector, industry, and IPO year | Excel | NASDAQ Screener (cleaned and filtered) |
| macroeconomic\_data.csv | Monthly interest rate, inflation, and GDP growth data | CSV | Compiled manually from  FRED and IMF sources |

These datasets were pre-processed and structured to support the design of a star schema within MySQL, enabling visualisation and analysis in Power BI.

### 2.2 Stock Price Dataset – all\_stocks\_5yr.csv

The dataset includes daily stock information for more than 500 companies listed on US exchanges during the period from January 2013 to December 2017. Fields include:

* Ticker: company symbol (used for joining) ● Date: trading date
* Open, High, Low, Close: price information
* Volume: total shares traded on that day

This dataset served as the main data source to generate monthly financial metrics which included average return calculations together with trading volume measurement and volatility assessment. The FactStockMonthly table emerged from this dataset after performing monthly data aggregation.

### 2.3 Company Metadata – cleaned\_nasdaq\_screener.xlsx

The main dataset contains stock information from companies that can be found in this Excel file. It includes:

* Ticker symbol (for joins)
* Company name
* Sector and industry classification
* Country
* IPO year

The metadata was cleaned by filling missing IPO years with zeros and replacing unknown countries with a placeholder. The DimCompany dimension table was created from this data to support filtering and sector and industry and IPO era comparisons.

### 2.4 Macroeconomic Indicators – macroeconomic\_data.csv

The file contains essential economic data that supports market performance analysis. It includes:

* Interest rate (monthly average)
* Inflation rate (monthly)
* GDP growth rate (estimated monthly)

The DimDate table received its date range values after the date column was added to the original dataset. This allowed for macroeconomic overlays in Power BI and made it possible to visualise correlations between stock performance and wider economic conditions.

### 2.5 Data Preparation Considerations

* The Python-based Pandas library performed data cleaning on all files before converting them into regular monthly-grained datasets.
* DateKey and CompanyKey surrogate keys were generated for warehouse usage.
* The analysis excluded tickers which exhibited naming problems or had missing data points.
* The analysis only included companies that traded continuously for five years.

The final cleaned tables were exported in.csv format before loading them into MySQL Workbench for modeling and Power BI reporting.

# Section 3: Business Requirements and Scope

## 3.1 Business Context

As one of the leading investment management companies worldwide, Fidelity Investments oversees a broad range of financial assets in global markets. The internal teams of analysts, portfolio managers and strategists within the organization depend heavily on data to make decisions, evaluate risks, and detect opportunities for growth.

The data was often separated from the other sources and the reporting was usually done using Excel or manual research. Due to the increased complexity in the markets and the need for timely responses to the volatility, Fidelity's analytics team needed a centralized Business Intelligence solution to centralize performance analysis and include macroeconomic factors.

The suggested BI system allows Fidelity to:

* Recognize past trends in stock performance.
* To compare the returns in different sectors and industries.
* To determine which stocks performed poorly and which ones performed well.
* Measure liquidity through trading volume trends.
* Discover the relationship between economic policies (such as interest rates) and equity market performance.

This system is intended to be expandable, updatable, and expandable to other markets, periods or classes of assets in the future.

## 3.2 Project Objectives

The main goals of the project were:

1. To **create a data warehouse** that combines stock, company and economic data using a star schema model.
2. **Create an ETL pipeline** to clean and transform the source files using Python and load the structured data into MySQL.
3. Create **Power BI dashboards** that allow for filtering and drilling down by ticker, sector, date and macroeconomic indicators.
4. To enable **pattern discovery** through the combination of return metrics with economic conditions to produce actionable investment insights.

## 3.3 Business Questions to Address

This project was framed around five key questions relevant to Fidelity’s equity analysts and investment strategists:

➹ What were the best and worst performing sectors over the five year period based on average monthly return?

➹ Which of the stocks were the best and worst performing and to which sector or industry did they belong?

➹ In what manner did changes in macroeconomic indicators like interest rates impact the average market returns for the same period?

➹ In which industries did strong returns occur consistently and how do they compare to their broader sector categories?

➹ Which sectors were the most liquid and how did the trading volume change from year to year?

Each question was answered through a specific Power BI visual that connected to the warehouse through live queries, thus allowing for dynamic exploration and future refresh capabilities.

# Section 4: BI Solution Design and Architecture

To deliver a flexible and refreshable Business Intelligence solution for Fidelity Investments, the project was implemented using a standard three-tier architecture: data integration, data storage, and data presentation. This approach ensures scalability, modularity, and the ability to independently update or extend each layer in the future.

## 4.1 Overview of BI Architecture

The BI solution was built using the following layered approach:

**Data Integration (ETL Layer)**

The raw source files underwent ingestion, cleaning and transformation through Python. The output from this layer consisted of clean, pre-aggregated.csv files structured at the monthly level.

**Data Storage (Warehouse Layer)**

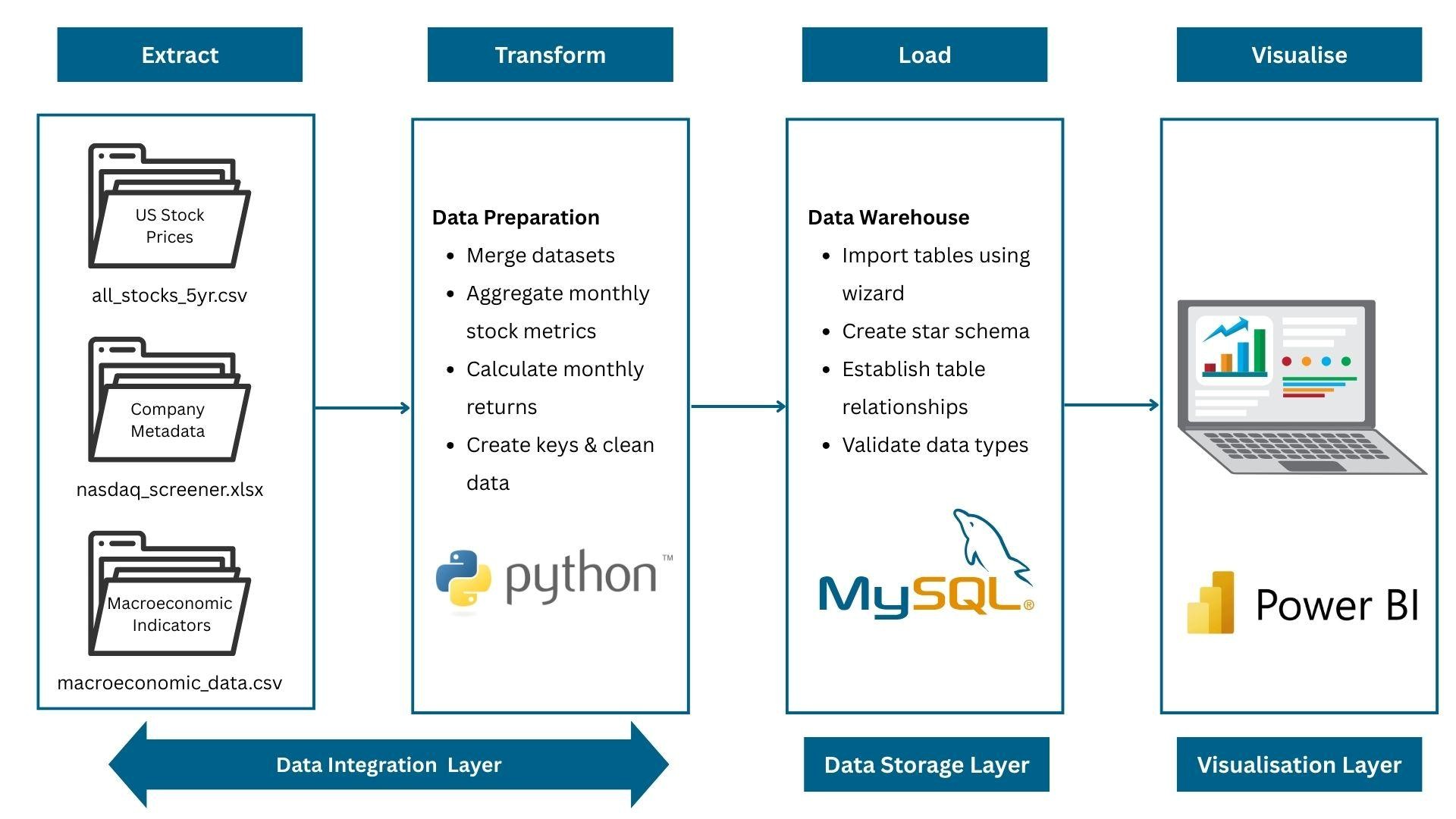
The relational star schema was implemented through MySQL Workbench. The model contained one fact table alongside two supporting dimension tables. All cleaned datasets received storage within this warehouse which allowed for straightforward reporting queries.

**Data Presentation (Visualisation Layer)**

Power BI Desktop linked to MySQL through the MySQL Connector. The reports used live queries for their design and the dashboard included filters for year, sector and ticker symbol.

## 4.2 BI Architecture Diagram

The following diagram outlines the data flow from raw files to dashboard:



*Figure 1 — ETL and BI System Architecture*

## 4.3 Components Used in Each Layer

|  |  |  |
| --- | --- | --- |
| **Layer** | **Tool(s) Used** | **Role & Purpose** |
| Data Integration | Python (Pandas) | Cleaned daily stock prices, merged macro data, created monthly aggregates |
| Data Storage | MySQL Workbench | Star schema design and implementation; enabled querying by date, sector, ticker |
| Data Presentation | Power BI Desktop | Dashboard creation, report interactivity, visual storytelling for stakeholders |

## 4.4 System Design Considerations

Multiple design choices were implemented to achieve performance and usability:

* The choice of monthly grain for fact table data aligned with available macroeconomic indicators and simplified report complexity.
* The warehouse joins and Power BI performance improved through the creation of surrogate keys (DateKey and CompanyKey).
* The MySQL live connection enables automatic dashboard updates when new data becomes available.

The current architecture enables future development of API-based data ingestion and cloud-based hosting as well as financial data source integration.

# Section 5: ETL Process

The ETL (Extract, Transform, Load) process functions as the fundamental component of this BI solution. Python executed the ETL process through a structured workflow which transformed three different datasets into a star schema structure.

This part describes the detailed process of transforming unorganized data into organized joinable tables for querying and visualization purposes.

## 5.1 Extract

Data was extracted from three different formats:

|  |  |  |
| --- | --- | --- |
| **Dataset** | **Format** | **Tool Used** |
| all\_stocks\_5yr.csv | CSV | Python (Pandas) |
| cleaned\_nasdaq\_screener.xlsx | Excel | Python (Pandas) |
| macroeconomic\_data.csv | CSV | Python (Pandas) |

All datasets were loaded using pandas.read\_csv() or pandas.read\_excel() and initially inspected for missing values, data types, and structural consistency.

## 5.2 Transform

The transformation phase was the most complex and was broken down by table.

#### 5.2.1 Transforming Stock Price Data (Fact Table)

**Goal:** Convert daily stock data into monthly aggregates and calculate derived financial metrics.

**Steps:**

1. Convert date column to datetime format.
2. Extract year and month into new columns.
3. Group by Ticker, Year, Month and compute:

○ month\_end\_close: last closing price in the month

○ monthly\_high: max of high

○ monthly\_low: min of low

○ total\_volume: sum of volume

○ monthly\_return was calculated in Python (Appendix 12.3)

1. This was done using .groupby() and .shift() in Pandas.
2. Generate DateKey in YYYYMM format as integer (e.g., 201401).
3. Filter out records with nulls or zeros after calculations.

The result was a clean, monthly-grain table of stock performance, with approximately 29,000 records and 6 key metrics.

**5.2.2 Transforming Company Metadata (DimCompany Table) Goal:** Standardise company context data and assign a surrogate key.

**Steps:**

1. Filter company metadata to include only tickers present in stock price dataset.
2. Standardise fields (Sector, Industry, Country) using .str.strip() and .fillna().
3. Replace missing IPO years with 0 and convert column to int.
4. Generate CompanyKey using the index position + 1.

**Final Columns:**

* CompanyKey (PK)
* Ticker
* CompanyName
* Sector
* Industry
* Country
* IPO\_Year

This dimension was used to drive filtering and grouping in Power BI visuals.

#### 5.2.3 Transforming Macroeconomic Data (DimDate Table)

**Goal:** Create a unified date dimension with monthly granularity and economic overlays.

**Steps:**

1. Generate a monthly date range from Jan 2013 to Dec 2017 using pandas.date\_range().
2. Assign each row a DateKey in YYYYMM format.
3. Merge the interest\_rate, inflation\_rate, and gdp\_growth columns into this base table.
4. Extract:

○ Year

○ Month

○ MonthName

○ Quarter

○ YearMonthLabel for Power BI visuals

**Final Columns:**

* DateKey (PK)
* Year, Month, MonthName, Quarter, YearMonthLabel
* interest\_rate, inflation\_rate, gdp\_growth

This dimension supported trend analysis and macro overlays in reporting.

## 5.3 Load

After the transformation was complete in Python, the cleaned datasets were exported as CSV files using DataFrame.to\_csv(). These CSV files were then loaded into MySQL using the MySQL Workbench Import Wizard (not via SQL scripts).

**Steps Followed:**

↓ Opened MySQL Workbench and selected the target database.

↓ Used Table Data Import Wizard to import:

↓ fact\_stockmonthly.csv

↓ dim\_company.csv

↓ dim\_date.csv

↓ Mapped CSV columns to MySQL fields and executed the import through the Workbench GUI.

This approach ensured clean schema definition and field types without writing manual SQL import queries.

**Tables created:**

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Type** | **Description** |
| FactStockMonthly | Fact | Stock performance metrics by month |
| DimCompany | Dim | Company-level metadata |
| DimDate | Dim | Calendar and macroeconomic information |

## 5.4 Python Libraries and Files Used

* pandas: data wrangling and aggregation
* numpy: numeric calculations
* datetime: date parsing and manipulation
* os: file path control for export

The ETL process was carried out entirely within a single Python script that was organized into modular blocks to allow for reusability and future scalability.

# Section 6: Data Modelling and Star Schema Design

The star schema emerged as the best data warehouse design because it provides simple structure and fast query performance and excellent compatibility with Power BI tabular models. The structure separates facts (numerical measurements) from dimensions (contextual descriptors), which enables efficient filtering, grouping, and slicing.

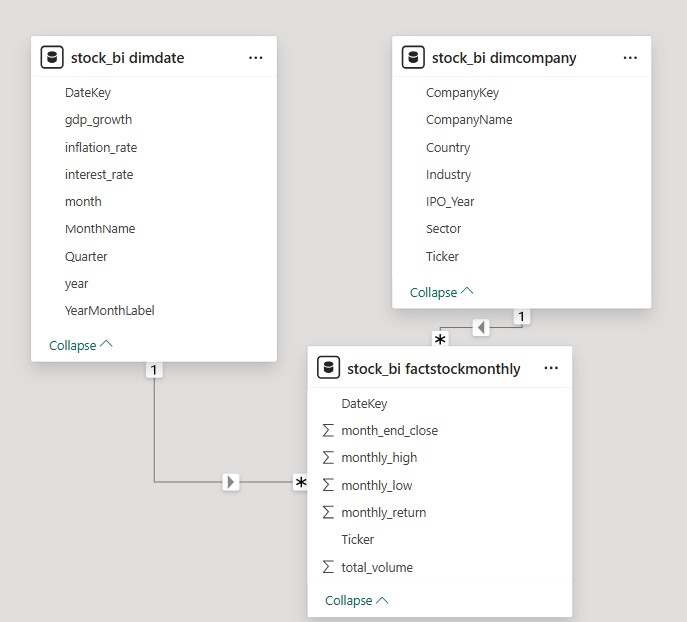
## 6.1 Overview of the Star Schema

The final schema consisted of:

* One fact table: FactStockMonthly
* Two dimension tables: DimCompany and DimDate

The structure supports flexible business analysis, such as evaluating performance by sector, year, or economic condition.

## 6.2 Entity Relationship Diagram



*Figure 2 — Star Schema ERD for Fidelity BI Solution*

The ERD shows:

* One-to-many relationships:

○ DimCompany.CompanyKey → FactStockMonthly.CompanyKey

○ DimDate.DateKey → FactStockMonthly.DateKey

* All primary keys are surrogate integer keys, optimised for joins in SQL and Power BI.

## 6.3 Table Descriptions

#### FactStockMonthly

This table holds monthly stock performance metrics, aggregated from daily data.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Type** | **Description** |
| CompanyKey | INT | Foreign key to DimCompany |
| DateKey | INT | Foreign key to DimDate |
| month\_end\_close | FLOAT | Closing price on last trading day of month |
| monthly\_high | FLOAT | Highest price in the month |
| monthly\_low | FLOAT | Lowest price in the month |
| total\_volume | BIGINT | Total volume traded during the month |
| monthly\_return | FLOAT | % change from previous month close |

#### DimCompany

Contains company-level metadata used for contextual filtering.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Type** | **Description** |
| CompanyKey | INT | Surrogate primary key |
| Ticker | VARCHAR | Unique symbol for the stock |
| CompanyName | VARCHAR | Full name of the company |
| Sector | VARCHAR | Sector classification (e.g., Tech) |
| Industry | VARCHAR | Industry classification |
| Country | VARCHAR | Country of incorporation |
| IPO\_Year | INT | Year of IPO |

#### DimDate

Holds temporal data and macroeconomic indicators.

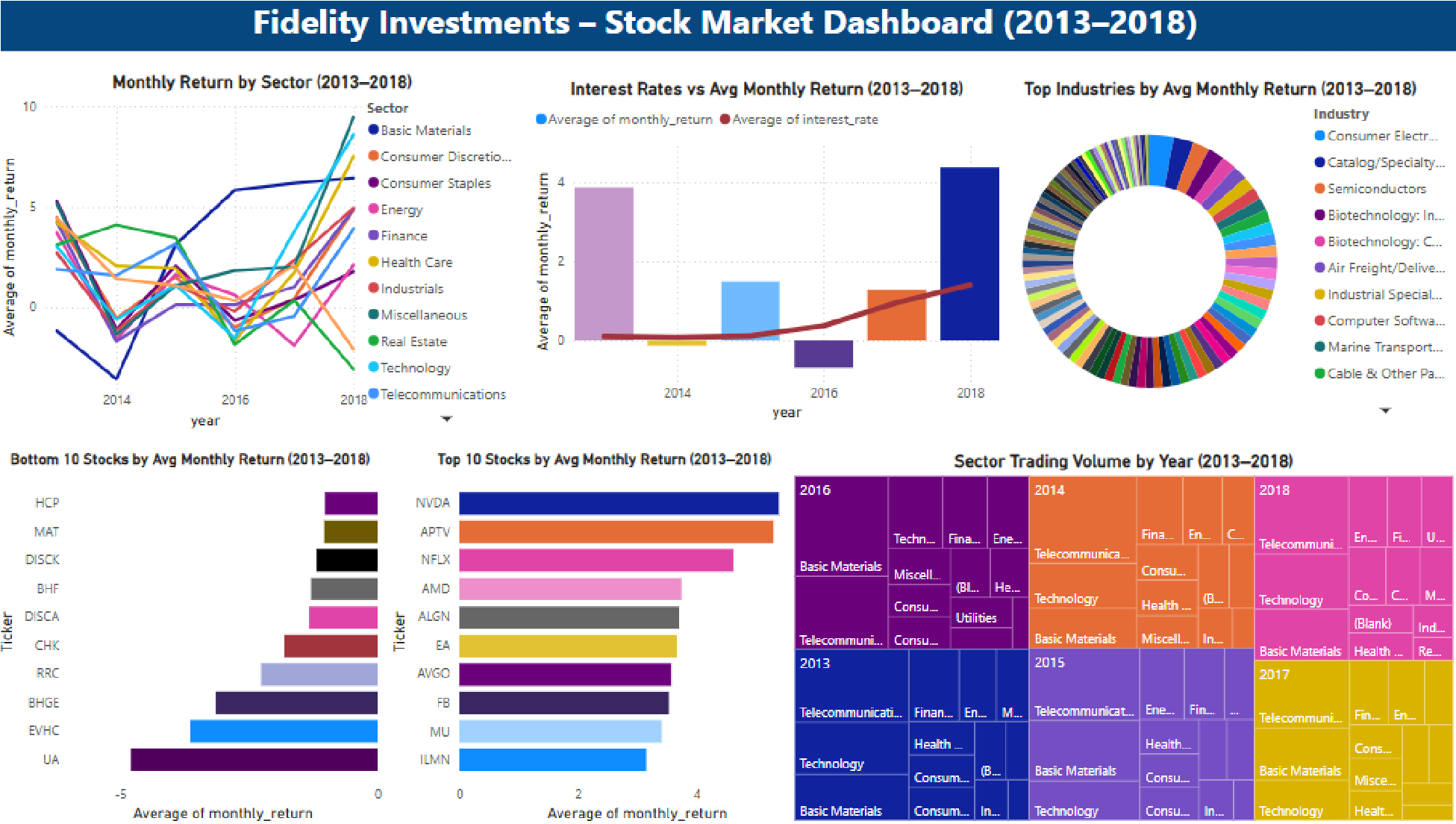
|  |  |  |
| --- | --- | --- |
| **Column Name** | **Type** | **Description** |
| DateKey | INT | Surrogate key (e.g., 201701) |
| Year | INT | Calendar year |
| Month | INT | Calendar month |
| MonthName | VARCHAR | Name of the month |
| Quarter | INT | Fiscal quarter |
| YearMonthLabel | VARCHAR | e.g., “Jan 2017” |
| interest\_rate | FLOAT | Central bank rate for that month |
| inflation\_rate | FLOAT | Monthly inflation |
| gdp\_growth | FLOAT | Monthly GDP growth  (estimated) |

## 6.4 Benefits of the Star Schema

* Efficient dimension lookups are possible because of one-to-many joins.
* Non-technical personnel find the structure straightforward to comprehend.
* The structure is designed to integrate perfectly into Power BI’s relationship view.
* New fact or dimension tables (e.g., ESG scores, earnings reports) can be added without changing the existing queries.

# Section 7: Power BI Reporting and Insights

The final Power BI dashboard presents sector returns together with macroeconomic trends and trading volume and individual stock performance across the 2013–2018 period. The visuals within this dashboard assist Fidelity Investments’ research and portfolio strategy teams in their quick and interactive assessment of market behaviour across different dimensions..



*Figure 3 — PowerBI Dashboard*

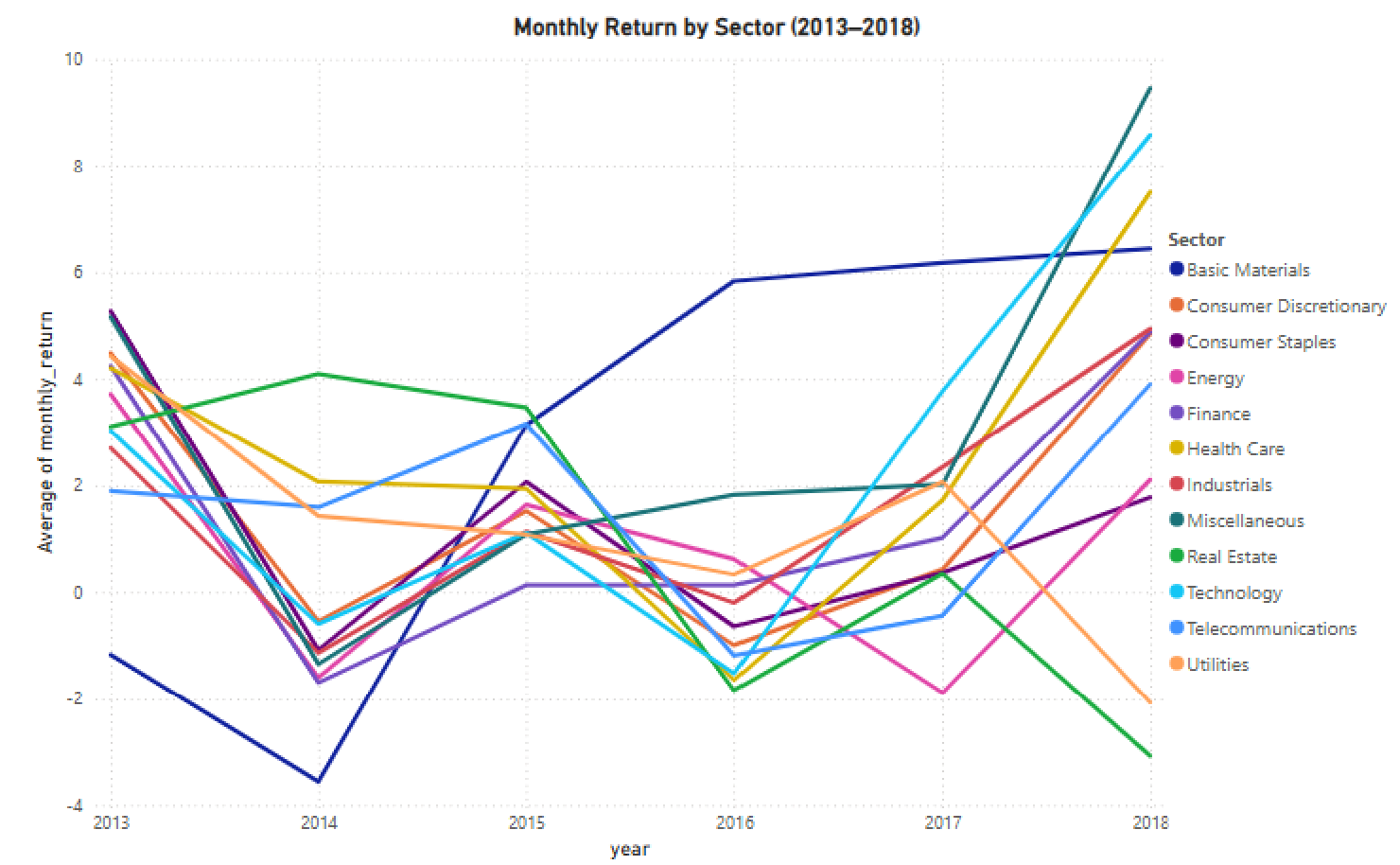
## 7.1 Monthly Return by Sector (2013–2018)

**Visual Type:** Line Chart  **Key Fields:**

* X-axis: YearMonthLabel
* Y-axis: AVG(monthly\_return)
* Legend: Sector

**Insight:** Technology, Consumer Discretionary, and Finance sectors showed strong but volatile performance while Utilities and Health Care sectors provided stable yet lower returns.

**Use Case:** The Fidelity sector strategists can use this to rebalance their portfolios based on historical sector patterns and select sectors that perform well during low-rate periods or market recovery times.



*Figure 4 - Line Chart*

## 7.2 Interest Rates vs Avg Monthly Return (2013–2018)

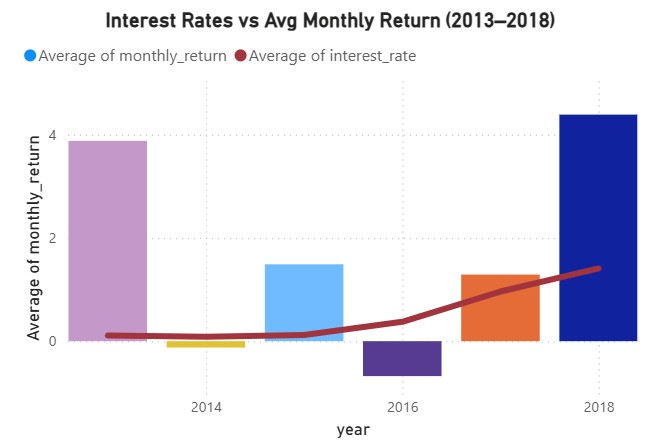
**Visual Type:** Combo Chart (Bar + Line)  **Key Fields:**

* Bar: AVG(monthly\_return)
* Line: AVG(interest\_rate)
* X-axis: Year

**Insight:** Interest rates and market returns showed an apparent negative relationship because consumer sector returns decreased when interest rates increased during 2017–2018.

**Use Case:** Economists at Fidelity will employ this visual to predict interest rate scenarios and how they affect portfolio returns specifically in macro-sensitive sectors.

#### *Figure 5 - Combo Chart (Bar and Line Chart)*



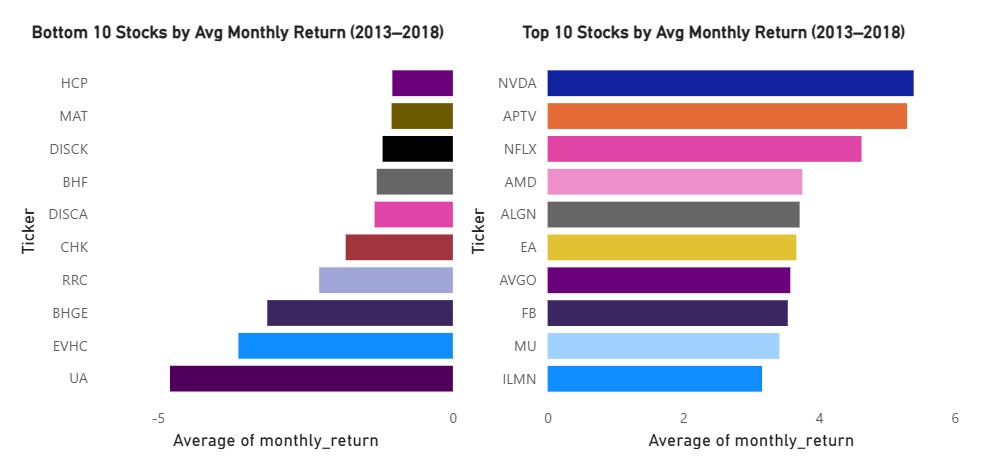
## 7.3 Top 10 & Bottom 10 Performing Stocks (2013–2018)

**Visual Type:** Dual Bar Charts  **Key Fields:**

* X-axis: AVG(monthly\_return)
* Y-axis: Ticker
* Filter: Top N & Bottom N = 10

**Insight:** The top performers included technology-driven NVDA and NFLX while underperformers focused on energy and legacy manufacturing stocks.

**Use Case:** Equity analysts who work at Fidelity can use this dashboard to measure their fund performance against top-performing stocks while monitoring underperforming stocks for possible divestment or strategic evaluation.



*Figure 6 - Dual Bar Chart*

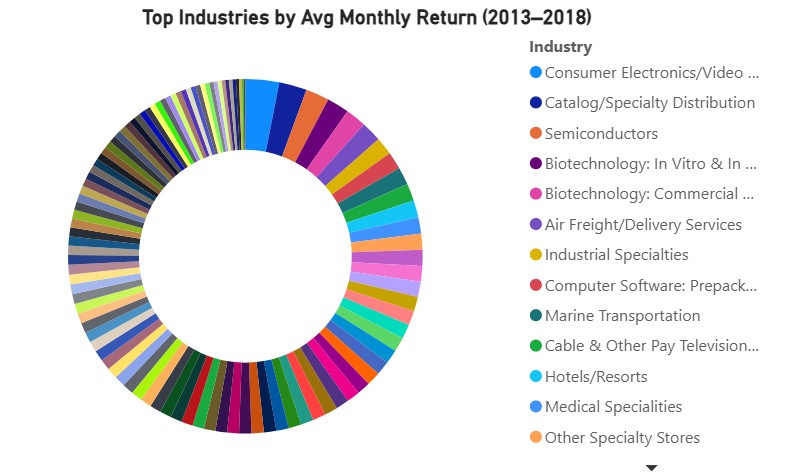
## 7.4 Top Industries by Avg Monthly Return

**Visual Type:** Donut Chart  **Key Fields:**

* Legend: Industry
* Values: AVG(monthly\_return)

**Insight:** The Semiconductors, Biotech, and Software industries provided continuous high returns to investors. The visual provides detailed analysis of specific industries which goes beyond sector-level performance.

**Use Case:** Thematically focused fund managers and innovation fund leaders at Fidelity use this dashboard to direct investments toward high-growth business sectors.



*Figure 7 - Donut Chart*

## 7.5 Sector Trading Volume by Year (2013–2018)

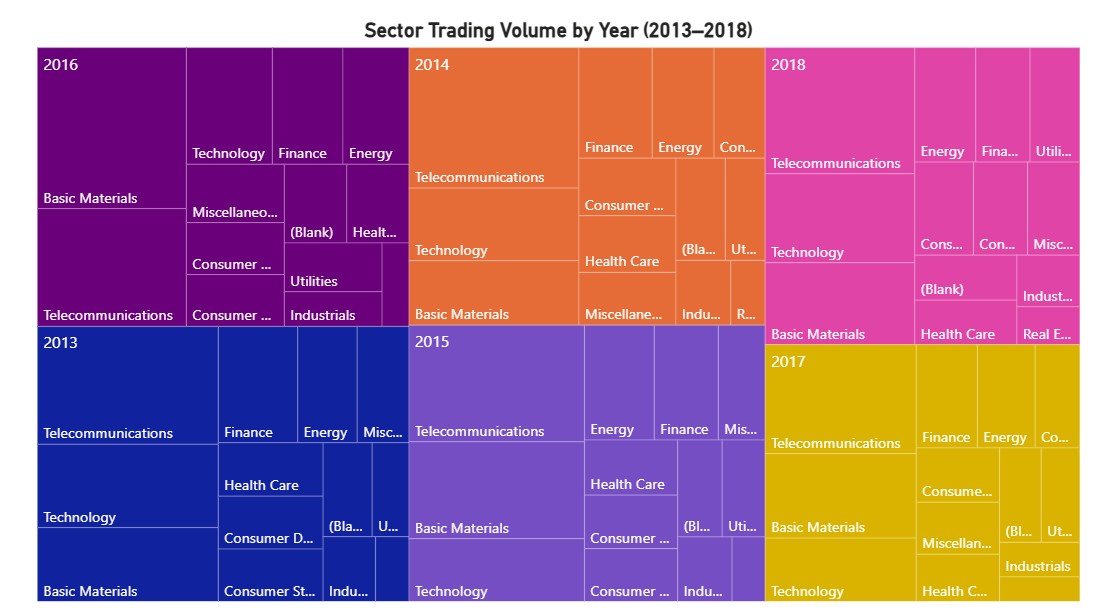
**Visual Type:** Treemap  **Key Fields:**

* Group: Sector
* Values: AVG(total\_volume)
* Tooltips: Year

**Insight:** The trade volume analysis indicates Technology and Telecommunications sectors maintained high liquidity levels but Real Estate and Miscellaneous sectors experienced lower trade activity.

**Use Case:** The liquidity risk teams along with high-frequency traders can identify active market sectors which helps them develop their execution approaches.

#### *Figure 8 - Donut Chart*



# 8. Key Findings and Business Implications

An analysis of five years of U.S. stock market data offers valuable insights into sector performance, stocks, macroeconomic factors, and trading activity. The analysis results in this paper are useful for Fidelity Investments’ institutional investors and portfolio analysts because they affect portfolio diversification decisions and when to enter or exit investments.

### 8.1 Sector-Level Observations

* Technology and Telecommunications sectors provided higher average monthly returns from 2013 to 2018 and their performance was especially strong from 2017 to 2018.
* Utilities and Real Estate sectors provided flat or negative return patterns, which indicates these sectors may not be the best choice for aggressive growth strategies.
* Energy and Consumer Discretionary sectors displayed high volatility, which suggests that there may be speculative investment opportunities with higher risk.

Implication: Managers can enhance portfolio returns by allocating more weight to performing sectors in bull markets and adjust the weights in bear markets to reduce portfolio volatility.

### 8.2 Stock Performance Trends

NVDA (NVIDIA), APTV, and NFLX (Netflix) were among the top 10 stocks in terms of average monthly return, which means that investors were very confident in these stocks and they performed well over the years.

On the other hand, stocks like UA, EVHC, and BHGE performed poorly, which could be due to company-specific problems or sector-related challenges.

**Implication:** These high performers can be picked up by Fidelity’s equity research teams for further analysis and can be included in active management strategies or thematic investment portfolios.

### 8.3 Macroeconomic Influences

➤ There is a weak negative relationship between interest rates and stock returns as seen in the visual correlation, stock returns were higher during lower interest rates especially before 2017.

➤ This implies that easy monetary policy has a positive impact on equity market growth.

**Implication:** Interest rates are other macroeconomic indicators that can be used as leading indicators to help sector rotation and risk management in the market.

### 8.4 Trading Volume Insights

⤒ The Technology and Telecommunications sectors had the highest trading volume year on year which shows that these sectors are the most active in terms of investor participation.

⤒ The heatmap analysis also showed a surge in sector-wide trading activity in 2016 and 2018, which could be due to macroeconomic announcements or earnings cycles.

**Implication:** High trading volumes usually mean high liquidity and positive investor sentiment. These sectors can be used in tactical strategies that focus on high turnover and news sensitivity.

### 8.5 Industry Diversification Potential

⟶ The donut chart showed more than 70 industries, and most of the top performing stocks fell under multiple industries.

⟶ The most prominent industries were Semiconductors, Software and Biotechnology, all of which were major contributors to the high return clusters.

**Implication:** Fidelity can create multi-industry ETF baskets or thematic funds on fast growing micro industries to target millennial and Gen Z investors who want to invest in niche areas.

### 8.6 Use Case for Fidelity Investments

⧫ Research analysts to observe how different sectors react to trends in the macroeconomy.

⧫ Fund managers to identify top and bottom performers over multi-year periods.

⧫ Traders to observe liquidity hotspots over years.

⧫ Executives to view market dynamics at a glance.

It is a useful tool for investment decisions, performance measurement and looking at the future allocation of resources.

# 9. Evaluation

The last part evaluates how the BI solution developed for Fidelity Investments works and how well it meets business requirements. The goal was to design a complete dashboard for stock market analytics to help investment analysts and business managers. This evaluation looks at what did well, what issues occurred, and how the solution meets real business needs.

### 9.1 Successes

A number of aspects of this BI solution operated correctly:

➹ **Robust Data Integration:** The ETL process cleaned and joined and imported stock performance and macroeconomic data from multiple CSV and SQL sources into a unified MySQL database. This enabled seamless integration into Power BI.

➹ **Insightful Visualisations:** The dashboard shows the key trends in sector-wise performance, year-on-year trading volume, and the relationship between interest rates and stock returns. Each chart answers a distinct business question and is very simple to read.

➹ **Effective Use of Data Modelling:** The star schema design made it easy to slice by dimensions such as sector, year, and industry. The relationships between the fact and dimension tables were well defined and optimised within Power BI.

➹ **Practical Analytical Relevance:** The dashboard shows the top and bottom performing stocks, plots the returns against macroeconomic indicators and presents the trading behavior which can be used to make strategic investment decisions.

### 9.2 Challenges Encountered

⤲ **Power BI SQL Connector Issues:** At the start, Power BI had problems connecting to the MySQL database because of missing components. The problem was solved by downloading and installing the MySQL ODBC connector and using an alternate authentication.

⤲ **Data Cleansing and Normalisation:** The fields gdp\_growth and interest\_rate were stored as text and had to be converted to numeric types. Null or placeholder values (e.g., '0' for IPO year) also needed to be cleaned.

⤲ **Dashboard Layout Constraints:** Controlling visual clutter especially in dense visuals like the donut chart of industries was challenging and needed careful formatting. Achieving readability without losing depth of analysis was an ongoing balance.

⤲ **Rounded Visual Formatting:** While trying to make large volume numbers and returns easy to read, Power BI’s formatting options were at times limited and only possible without the use of DAX formulas.

### 9.3 Limitations of the Analysis

It is crucial to note the following limitations of the solution despite the fact that it offers considerable insights:

⥀ **Static Dataset:** The analysis uses historical data (2013–2018), limiting real-time forecasting or live market responsiveness.

⥀ **Lack of Sentiment or News Signals:** External factors like market sentiment, geopolitical events, or breaking news were not included in the scope.

⥀ **Limited Financial Metrics:** The dashboard focuses solely on stock returns and volumes. The P/E ratio, earnings per share or debt-to-equity ratios were outside the dataset but could offer further insight.

### 9.4 Future Enhancements

With more time and resources the following improvements could be made to significantly enhance the dashboard’s strategic utility:

**Live Market Integration:** To enable more responsive decision-making, connect APIs like Alpha Vantage or Yahoo Finance to get near real-time data refreshes.

**Forecasting Models:** Using Python or DAX, incorporating time-series forecasting models (ARIMA or Prophet) to predict stock movements based on historical trends.

**Interactive Sector Drill-Down:** To allow users to drill into individual companies or industries for comparative analysis within or across sectors.

**Enhanced Financial Analysis:** Adding valuation metrics and comparing fundamentals (e.g., ROI, margins) across companies to support a more comprehensive investment analysis.

**User-Specific Dashboards:** Creating role based dashboards for different personas, for example, equity analyst, risk manager, or senior portfolio strategist.

### 9.5 Alignment to Business Needs

The dashboard delivers value by turning raw financial data into actionable insights. For a firm like Fidelity Investments, this dashboard would:

⟾ **Support Portfolio Management:** Fast identification of top-performing sectors and stocks helps in dynamic portfolio allocation and risk mitigation.

⟾ **Assist in Economic Contextualisation:** The inclusion of interest and inflation rates helps strategists to understand the performance in a macroeconomic environment.

The BI solution transforms historical stock and economic data into a strategic decision-making asset for a leading investment firm.

# 10. Conclusion

This project showed how to build a Business Intelligence solution for analyzing stock performance and macroeconomic trends for Fidelity Investments through end-to-end development. The solution merged stock price data with company metadata and macroeconomic indicators in a MySQL-hosted star schema to generate interactive Power BI dashboards.

The Python transformation and extraction process resulted in uniform and clean data tables that shared a common monthly time step. The resulting warehouse allowed users to slice data through sector and ticker and industry and economic conditions. The visualizations answered essential investment questions by displaying which sectors performed better than others and which stocks performed well and which macro factors affected returns and which markets had high liquidity.

The Fidelity Investments dashboard delivers essential value to different departments within the organization.

* Portfolio managers track sector rotation patterns as they evaluate performance fluctuations.
* Analysts utilize the system to analyze return drivers while identifying strong performing assets and underperforming assets.
* Risk teams can utilize market behavior data in combination with economic indicators to develop exposure plans.

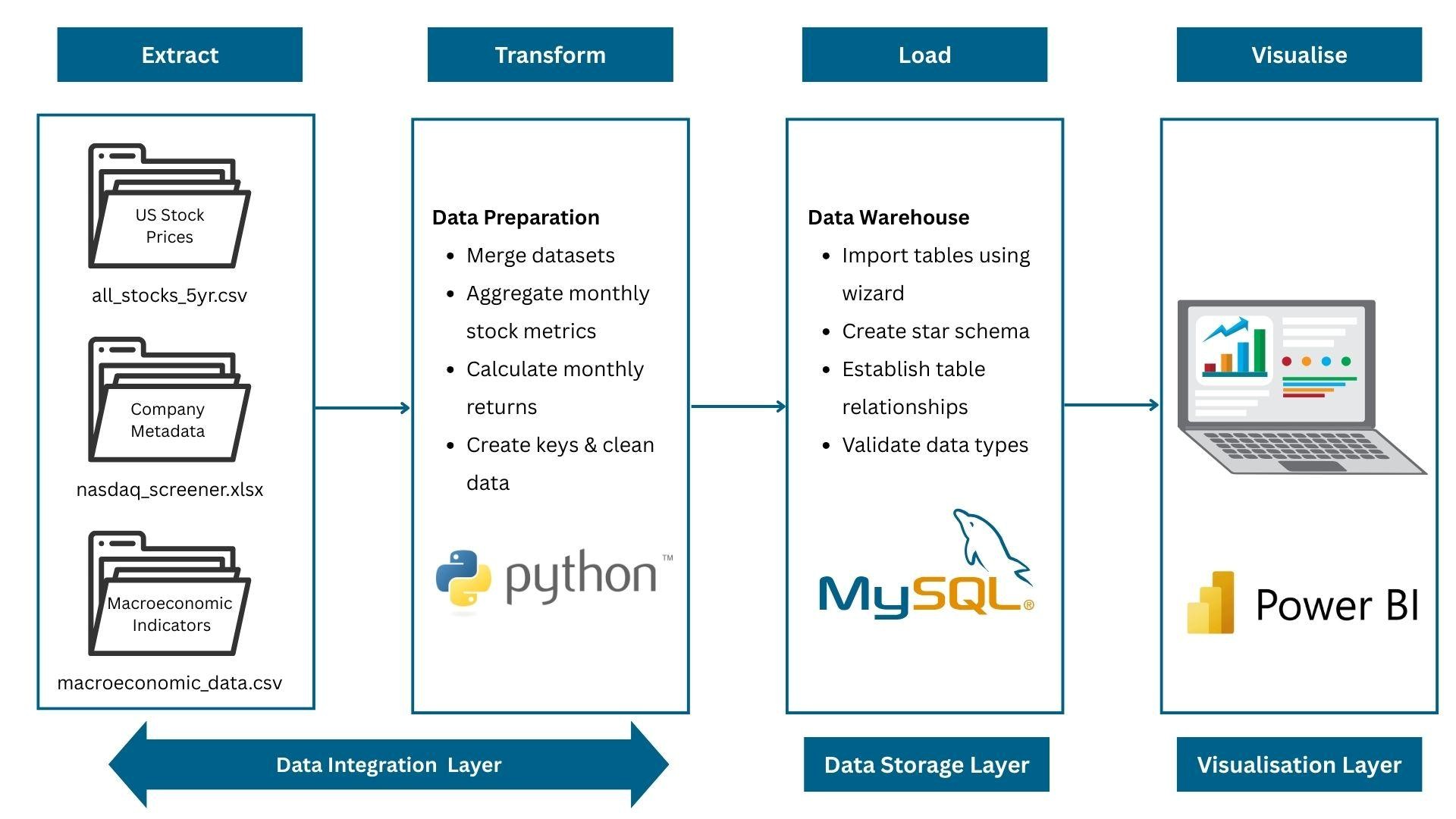
The dashboard uses historical data to generate strategic insights while its architecture allows future expansion. Additional development would allow the architecture to receive real-time data feeds and implement predictive analytics and ESG metrics integration.

The integrated BI pipeline demonstrates substantial analytical depth and operational value and visual clarity for high-stakes decision-making within institutional finance by using open-source and enterprise-friendly tools.

# 12. Appendix

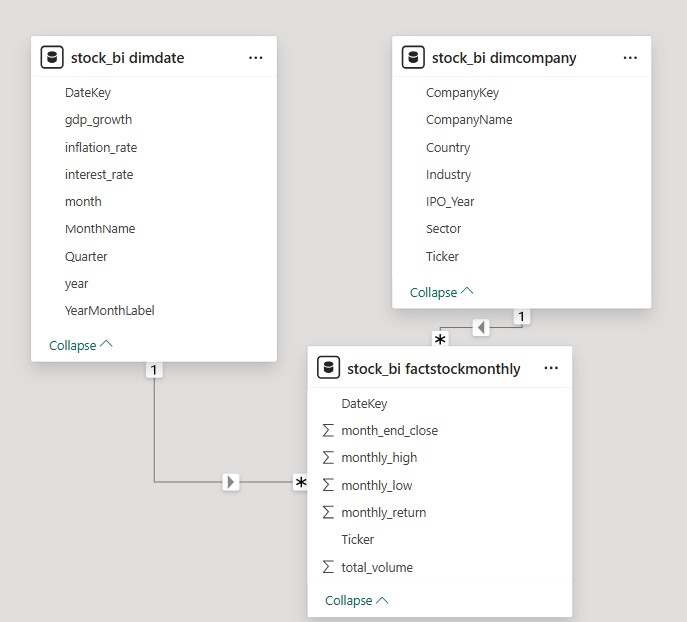
This appendix includes supporting artefacts that provide evidence of the technical implementation and back-end development of the BI solution. The materials were not included in the main body of the report but were essential to the construction of the data pipeline and model.

### 12.1 ETL Process Flowchart



This diagram illustrates the full ETL process used in the project — from extracting raw stock and macroeconomic data to delivering interactive dashboards in Power BI. The process was executed using Python for transformation, MySQL for data warehousing, and Power BI for visualisation.

### 12.2 Entity Relationship Diagram (ERD)



The data warehouse was modelled using a star schema consisting of one fact table and two dimension tables. The ERD below shows the one-to-many relationships between FactStockMonthly, DimCompany, and DimDate.

### 12.3 Sample Python Code (ETL Script)

The following Python snippet shows the logic used to transform raw daily stock price data into monthly-level aggregates during the ETL process.



### 12.4 Sample SQL Code (Fact Table Creation)

The fact table below stores stock performance metrics by company and month. It was created using the following SQL statement in MySQL Workbench.



