Assignment-Part 1 & 2

Predicting and Understanding Stock Performance Trends Using Historical Financial Data

By: Group 11

Naaz Fatima (NF1020595)

Sai Bhaskar Chundru (NF1010427)

Yuvraj Singh (NF1016125)

Azizakhon Erkinbaeva (NF1015818)

Master of Data Analytics, University of Niagara Falls

Data Analytics Case Study 3 (DAMO-611-6)

Summer 2025

Professor: Omid Isfahanialamdari

Aug 31, 2025

Table of Contents

1 INTRODUCTION

The stock market produces an enormous volume of financial data on a daily basis. Price trend prediction and market behavior understanding are valuable in making an informed investment. Due to the many factors that influence financial market volatility such as macroeconomic variables, investor mood, and geopolitical concerns, it is difficult to predict market trends. With the emergence of an abundance of historical stock price data and improved analytical techniques for developing predictive models, investors and analysts can predict stock price movements more accurately.

2 STATEMENT OF PURPOSE

The goal of this project is to better understand how stock prices behave and what patterns might help guide smarter investment decisions. Instead of looking at the market in a broad sense, this study focuses on a few specific questions: Do different companies move in similar ways, or do some show more extreme ups and downs? Can overnight price gaps give us clues about what will happen the next day? Do sudden bursts in trading activity usually mean bigger price swings? And finally, do stock returns follow seasonal patterns, such as certain months or weekdays performing better than others?

These questions matter because investors, traders, and even policymakers are always trying to make sense of what drives markets. By analyzing historical data with both statistical tools and clear visualizations, this project aims to turn raw market movements into practical insights. The hope is that by uncovering these relationships to find whether it's through return profiles, daily gaps, trading volume spikes, or seasonal cycles, with this we can shed light on how markets work and how investors can make more informed choices while managing risk.

3 SCOPE OF THE PROJECT

This project focuses on analyzing stock market data to explore how prices and trading activity behave under different conditions. The study is limited to four key areas:

1. **Return Profiles Across Companies** – Comparing whether companies' returns follow similar patterns or whether some show more extreme gains and losses than others.

2. **Overnight Price Gaps** – Investigating whether the difference between one day's closing price and the next day's opening price can predict the market trend for that day.

3. **Impact of Trading Volume Spikes** – Examining whether unusually high trading activity is linked to larger price changes and greater volatility.

4. **Seasonality in Returns** – Looking for repeating patterns in stock returns across different days of the week or months of the year.

The project relies on historical stock price and volume data, applying both statistical tests and visualization techniques to identify meaningful relationships. While the analysis provides insights into these specific questions, it does not attempt to build a full forecasting model or cover every possible factor that drives stock prices (such as macroeconomic indicators, company earnings, or geopolitical events).

By keeping the scope focused, the project aims to highlight a few practical aspects of stock behavior that can help investors and analysts think more critically about risk, predictability and decision-making.

3.1 DELIVERABLES OF THE PROJECT

The project will generate the following deliverables:

- **Data Preparation and Cleaning** – Stock market dataset prepared and cleaned, including creation of an adjusted close column. *(Outputs.:pynb file, .csv file)*

- **Computation of Financial Metrics** – Calculation of Daily Returns, Body size, Body Start, Company Average Return, Company Voltality, Intraday Move, Monthly Average Return, Overnight Gap, Return Category, Return Vs Company Average, Size and Voltality. *(Outputs: .twbx file)*

- **Dashboards and Visualizations** – Interactive Tableau dashboards and Python-based visualizations for clear interpretation of findings. *(Outputs: .twbx file, .ipynb file)*

- **Data Modeling and Evaluation** –

    - Statistical Models: Logistic regression, OLS regression, Chi-square, Mann–Whitney U, Kruskal–Wallis tests to test hypotheses.

    - Machine Learning Models: Logistic Regression classifier and Random Forest, evaluated with TimeSeriesSplit cross-validation and an 80/20 chronological holdout.

    - Performance Metrics: AUC, accuracy, Brier score, log-loss, plus backtest metrics (CAGR, Sharpe ratio, max drawdown).

    - Outputs: .ipynb file (model training & evaluation).

- **Research Findings Report** – A written summary of hypotheses, results, and insights derived from the analysis. *(Outputs: .pdf file)*.

4 PROBLEM DEFINITION

This project seeks to analyze historical stock market data to better understand the patterns and behaviors that influence price movements and investor decisions. Using data sourced from Yahoo Finance, the study applies statistical analysis and visualization techniques to uncover meaningful insights into stock returns, trading volume, and seasonal trends. The goal is not only to describe how stocks have behaved in the past but also to highlight patterns that may inform future decision-making and risk management.

Through this analysis, the project aims to:

- Compare the return profiles of different companies to identify whether some exhibit more extreme movements than others.
- Examine whether overnight price gaps can signal the direction of the following day's trend.
- Investigate how unusual spikes in trading volume are connected to larger-than-usual price changes.
- Explore whether seasonal effects, such as certain days or months performing differently, play a role in stock returns.

.

5 RESEARCH QUESTIONS, HYPOTHESIS & JUSTIFICATION

This project will investigate the following research questions linking to the testable hypothesis along with the justification of each research question:

RQ1: Do companies share a similar return profile, or do some experience more extreme gains and losses compared to the rest?

- **H0 (Null):** The return distributions of all companies are statistically similar.
- **H1 (Alternative):** At least one company's return distribution differs, showing heavier tails or more extreme moves.
- **Justification:** Different companies operate in different industries, face different risks, and attract different types of investors. For example, technology firms often show higher volatility compared to stable utilities or consumer staples. By comparing distributions, we can identify whether some companies experience disproportionately large gains or losses, which is vital for portfolio diversification and risk management.

RQ2: Can the price difference between day T's close and day T+1's open (the overnight gap) predict the direction of the daily trend on day T+1?

- **H0 (Null):** The direction of the opening price gap has no statistically significant relationship with the final daily return on day T+1.
- **H1 (Alternative):** A significant positive opening gap predicts a positive daily trend (and vice versa for a negative gap).
- **Justification:** Overnight gaps often reflect after-hours news, earnings announcements, or macroeconomic updates. If these gaps carry predictive power, it suggests investor sentiment and momentum spill over into the trading session. This insight is valuable for

traders who act early in the market day and want to capitalize on intraday trends.

RQ3: Do sudden spikes in trading volume lead to larger-than-usual price changes in a company's stock?

- **H0 (Null):** Trading volume has no significant effect on the size of daily price changes.
- **H1 (Alternative):** Higher trading volumes are linked to larger daily price changes (higher volatility).
- **Justification:** The spikes in trading volume usually occur when significant information reaches the market, such as product launches, earnings reports, or market rumors. These events often lead to rapid price adjustments as investors rush to buy or sell. Testing this relationship helps us understand whether volume can be used as a proxy for market-moving events and volatility forecasting.

RQ4: Do stock returns show seasonal patterns, such as certain months or weekdays consistently performing better or worse?

- **H0 (Null):** Returns are random with no seasonal effect.
- **H1 (Alternative):** Certain months or days have significantly different average returns.
- **Justification:** Financial markets often show anomalies like the "January effect" or "Monday effect," where specific periods tend to have consistent return patterns. If these seasonal effects exist, they can inform timing strategies for investors and provide evidence against the notion that markets are fully efficient. Identifying seasonality also helps portfolio managers adjust strategies to exploit or mitigate these calendar-based risks.

6 LITERATURE

1. Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. The Journal of Finance, 25(2), 383–417.
2. Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. Expert Systems with Applications, 42(1), 259–268.

7 DATA COLLECTION AND UNDERSTANDING

This section explains about the data source and data understanding steps:

7.1 DATA SOURCE

The dataset has been extracted from Kaggle with the following details:

- **Dataset:** Massive Yahoo Finance Dataset (Kaggle); https://www.kaggle.com/datasets/iveeaten3223times/massive-yahoo-finance-dataset
- **File Name:** stock_details_5_years.csv
- **Content:** Daily stock prices, trading volume, and additional features for thousands of tickers over multiple years.
- **Variables:** Date, Open, High, Low, Close, Volume, Dividends, Stock, Splits & Company.

7.2 DATA UNDERSTANDING & CLEANING

This section describes the various steps involved in data cleaning which are done using Python. The Python code is executed in Jupyter notebook under visual studio code.

The below mentioned code is used to load the dataset, understand the structure, analysing the summary statistics, handling the missing values, duplicates, converting the data types and outlier detection. Once these steps are executed and the cleaned file is saved for further processing.

```
# Load Dataset
file_path = "stock_details_5_years.csv"
df = pd.read_csv(file_path)

# Understand Structure
print("Shape of dataset:", df.shape)
print("\nColumn names:", df.columns.tolist())
print("\nFirst 5 rows:\n", df.head())
print("\nData Types:\n", df.dtypes)

# Summary Statistics
print("\nSummary Statistics:\n", df.describe(include='all'))

# Missing Value Analysis
print("\nMissing values per column:\n", df.isnull().sum())
missing_percentage = df.isnull().mean() * 100
print("\nMissing value percentage:\n", missing_percentage)

# Handle Duplicates
duplicates = df.duplicated().sum()
print(f"\nNumber of duplicate rows: {duplicates}")
if duplicates > 0:
    df = df.drop_duplicates()

# Convert Data Types Safely
# Convert Date column to datetime, keep invalid as NaT
if 'Date' in df.columns:
    df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
    print("Number of invalid dates:", df['Date'].isnull().sum())
```

```
# Only convert numeric columns that exist
numeric_cols = ['Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume']
existing_numeric_cols = [col for col in numeric_cols if col in df.columns]

for col in existing_numeric_cols:
    df[col] = pd.to_numeric(df[col], errors='coerce')  # invalid entries become NaN
    df[col] = df[col].fillna(df[col].median())  # fill missing numeric values

# Handle Missing Values in Other Columns
non_numeric_cols = df.select_dtypes(exclude=[np.number, 'datetime']).columns
for col in non_numeric_cols:
    df[col] = df[col].fillna("Unknown")

# Outlier Detection (IQR method)
for col in existing_numeric_cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = ((df[col] < lower_bound) | (df[col] > upper_bound)).sum()
    print(f"Outliers in {col}: {outliers}")

# Save Cleaned Dataset
df.to_csv("cleaned_stock_details_5_years.csv", index=False)
print("\n Data cleaning complete. All rows preserved and saved as
'cleaned_stock_details_5_years.csv'")
```

The output of the code is as below, and after execution an cleaned file is saved as "cleaned_stock_details_5_years.csv".

Shape of dataset: (602962, 9)

Column names: ['Date', 'Open', 'High', 'Low', 'Close', 'Volume', 'Dividends', 'Stock Splits', 'Company']

```
First 5 rows:
                       Date        Open        High         Low       Close  \
0  2018-11-29 00:00:00-05:00   43.829761   43.863354   42.639594   43.083508
1  2018-11-29 00:00:00-05:00  104.769074  105.519257  103.534595  104.636131
2  2018-11-29 00:00:00-05:00   54.176498   55.007500   54.099998   54.729000
3  2018-11-29 00:00:00-05:00   83.749496   84.499496   82.616501   83.678497
4  2018-11-29 00:00:00-05:00   39.692784   40.064904   38.735195   39.037853

      Volume  Dividends  Stock Splits Company
0  167080000       0.00           0.0    AAPL
1   28123200       0.00           0.0    MSFT
2   31004000       0.00           0.0   GOOGL
3  132264000       0.00           0.0    AMZN
4   54917200       0.04           0.0    NVDA

Data Types:
 Date            object
Open            float64
High            float64
Low             float64
Close           float64
Volume            int64
Dividends       float64
Stock Splits    float64
Company          object
dtype: object
```

```
Summary Statistics:
                           Date           Open           High  \
count                    602962  602962.000000  602962.000000
unique                     1258            NaN            NaN
top     2023-10-31 00:00:00-04:00            NaN            NaN
freq                        491            NaN            NaN
mean                        NaN     140.074711     141.853492
std                         NaN     275.401725     279.003191
min                         NaN       1.052425       1.061195
25%                         NaN      39.566159      40.056222
50%                         NaN      79.177964      80.125563
75%                         NaN     157.837190     159.746317
max                         NaN    6490.259766    6525.000000

                 Low          Close        Volume     Dividends  \
count   602962.000000  602962.000000  6.029620e+05  602962.00000
unique            NaN            NaN           NaN           NaN
top               NaN            NaN           NaN           NaN
freq              NaN            NaN           NaN           NaN
mean       138.276316     140.095204  5.895601e+06       0.00731
std        271.895276     275.477969  1.381596e+07       0.12057
min          1.026114       1.034884  0.000000e+00       0.00000
25%         39.058151      39.563746  1.031500e+06       0.00000
50%         78.193820      79.177906  2.228700e+06       0.00000
75%        155.841609     157.847153  5.277400e+06       0.00000
max       6405.000000    6509.350098  1.123003e+09      35.00000
```

```
        Stock Splits Company
count   602962.000000  602962
unique            NaN     491
top               NaN     LYV
freq              NaN    1258
mean         0.000344     NaN
std          0.050607     NaN
min          0.000000     NaN
25%          0.000000     NaN
50%          0.000000     NaN
75%          0.000000     NaN
max         20.000000     NaN

Missing values per column:
 Date            0
Open             0
High             0
Low              0
Close            0
Volume           0
Dividends        0
Stock Splits     0
Company          0
dtype: int64
```

```
Missing value percentage:
 Date            0.0
Open             0.0
High             0.0
Low              0.0
Close            0.0
Volume           0.0
Dividends        0.0
Stock Splits     0.0
Company          0.0
dtype: float64

Number of duplicate rows: 0
Number of invalid dates: 0
Outliers in Open: 42159
Outliers in High: 42238
Outliers in Low: 42134
Outliers in Close: 42173
Outliers in Volume: 66419
```

After the data cleaning is done an additional column for Adjusted Close is created by using the below code and the new file is save as “adjusted_dataset.csv”:

```
# Load the dataset
file_path = "cleaned_stock_details_5_years.csv"
df = pd.read_csv(file_path)

# View the first 5 rows and columns
print(df.head())
print(df.columns)

# Create Adjusted Close
if 'Close' in df.columns:
    # Assuming no splits for simplicity, only adjust for dividends
    if 'Dividends' in df.columns:
        df['Adj Close'] = (df['Close'] + df['Dividends']).round(3)
    else:
        df['Adj Close'] = df['Close'].round(3)
else:
    print("No 'Close' column found in the dataset.")

# Save the updated dataset
df.to_csv('adjusted_dataset.csv', index=False)
print("Adjusted Close column created and dataset saved as 'adjusted_dataset.csv'")
```

8 FEATURE ENGINEERING AND CALCULATION

As part of the report and dashboard creation using various plots & matrices, implemented various filters & sorting and created different calculated fields, parameters & groups.

8.1 CALCULATED FIELDS

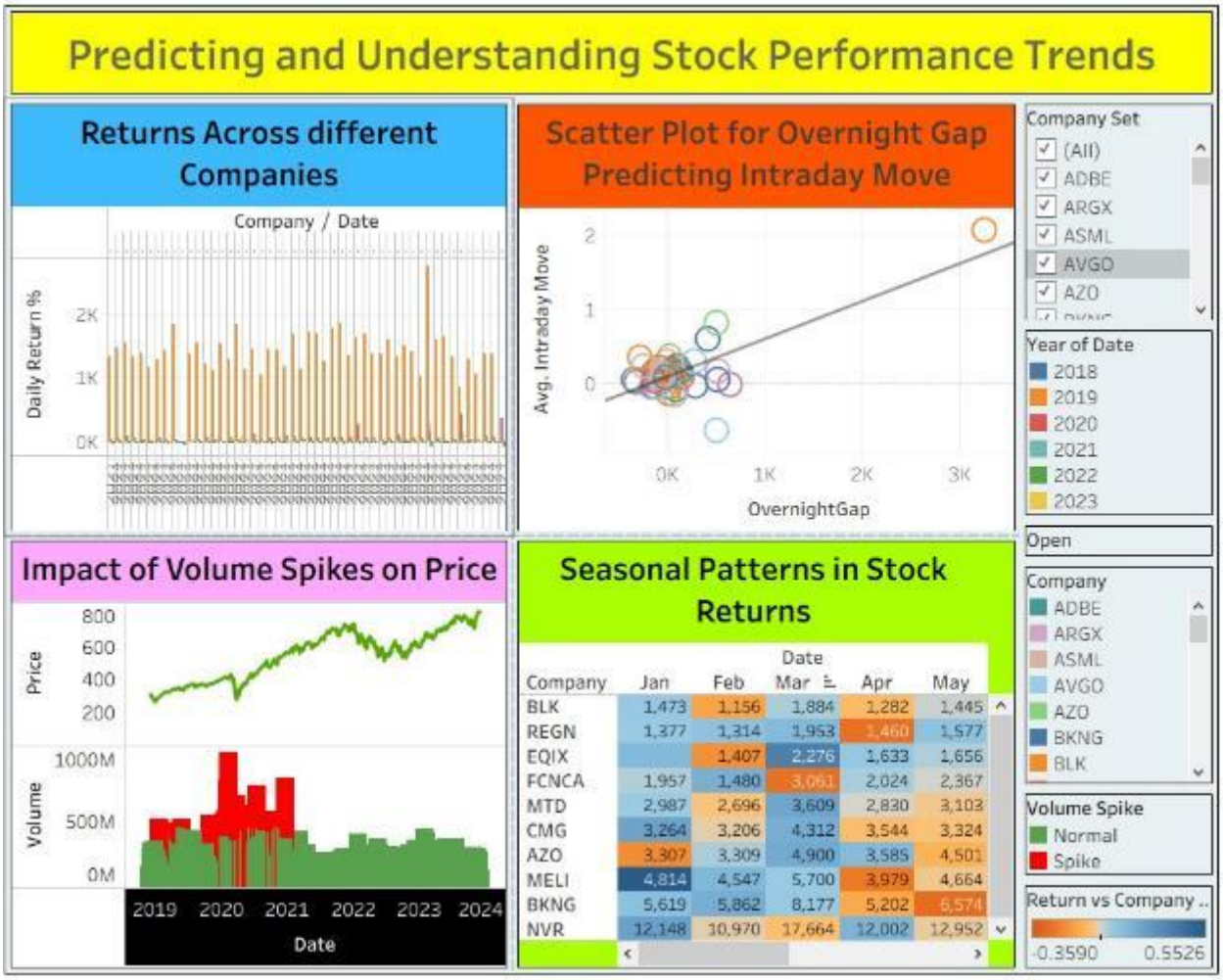
To generate the calculated fields in Tableau, click on **"Create Calculated Field"** and use the appropriate formulas, the various calculated fields created are Daily Returns, Body size, Body Start, Company Average Return, Company Volatility, Intraday Move, Monthly Average Return, Overnight Gap, Return Category, Return Vs Company Average, Size and Volatility.

8.2 PARAMETERS

To create a parameter, click on create parameter which will opens up the parameter pane. Now one can set the parameters as required. For this project a parameter (Volatility Threshold) is created.

9 ADVANCE VISUALIZATION AND DASHBOARD

The various visuals are generated under different sheets and are combined in dashboard.



9.1 RETURNS ACROSS DIFFERENT COMPANIES

The bar chart shows that while many companies have daily returns clustered around lower ranges, certain firms experience extreme spikes.

- For example, one company shows a return exceeding 2000% on an outlier day, highlighting how a few rare events can dominate return distributions.
- In contrast, most firms remain under 500% daily returns, which demonstrates the uneven risk exposure between companies.
- Key Takeaway: Investors should not assume all stocks share the same return behavior — some firms present much higher volatility, requiring different risk management strategies.

9.2 SCATTER PLOT FOR OVERNIGHT GAP PREDICTING INTRADAY MOVE

The scatter plot demonstrates a positive relationship between overnight gaps and intraday moves.

- For example, when AVGO had an overnight gap of around +2,800 points, the following day's average intraday move was above +2.0, showing strong continuation momentum.
- Smaller gaps (close to zero) cluster near flat intraday moves, confirming that large overnight changes carry more predictive power.
- Key Takeaway: Traders can use overnight gaps as an early signal — large positive gaps often precede positive trading days, while negative gaps are linked to weaker daily performance.

9.3 IMPACT OF VOLUME SPIKES ON PRICE

The dual-axis chart highlights how volume surges coincide with sharp price change.

- For example, in 2020, trading volume spiked above 1,000M shares, coinciding with price fluctuations between $400 and $600.
- After volume normalized post-2021, price growth stabilized and continued its upward trend toward $800+ by 2024.
- Key Takeaway: Abnormal trading activity (volume spikes) is a strong indicator of heightened volatility and often accompanies major market events like earnings or macroeconomic news.

9.4 SEASONAL PATTERNS IN STOCK RETURNS

The heatmap reveals clear seasonal patterns in returns across companies.

- For example:

  NVR achieved exceptionally high returns in March (17,664), far above other months, showing a seasonal spike.

  MELI consistently performed strongly in January (4,814) and February (4,547) compared to weaker months.

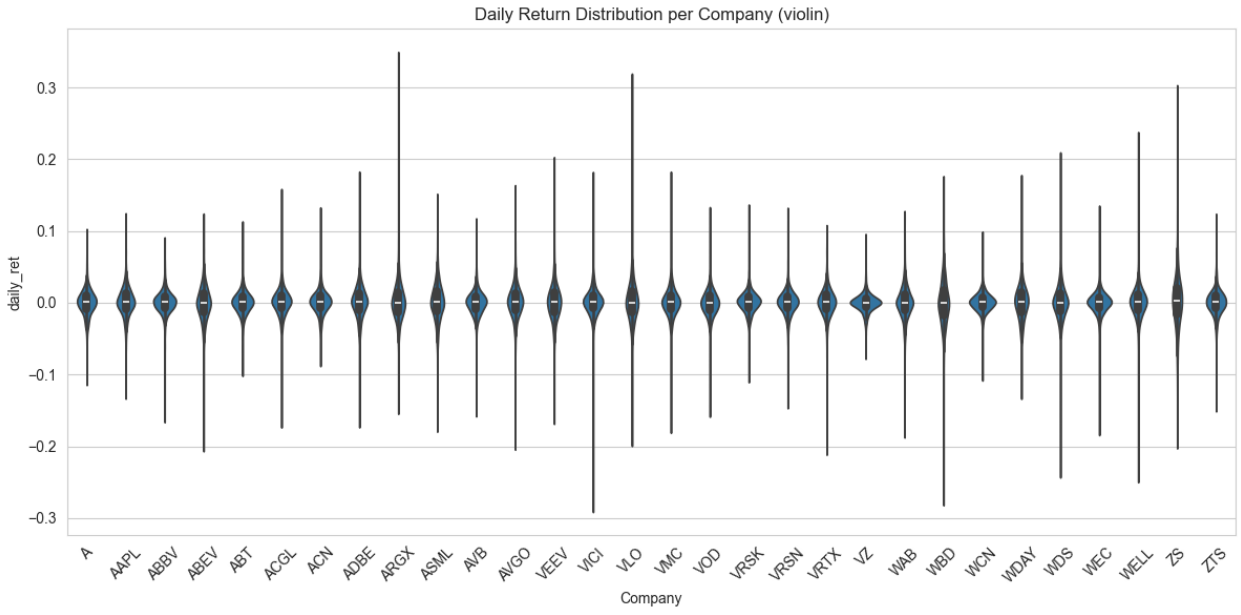
  EQIX displayed stronger performance in March (4,207) but lower returns in May (1,656).

- Key Takeaway: Certain companies display predictable seasonal effects, suggesting that timing strategies (e.g., holding NVR in March or MELI in early months) can enhance portfolio performance.

Apart from these visuals, few more visuals developed in python to further support the research questions and hypothesis, below sections provide the brief about these visuals.

9.5 DAILY RETURN DISTRIBUTION PER COMPANY

The violin plot illustrates the spread and shape of daily return distributions across multiple companies.

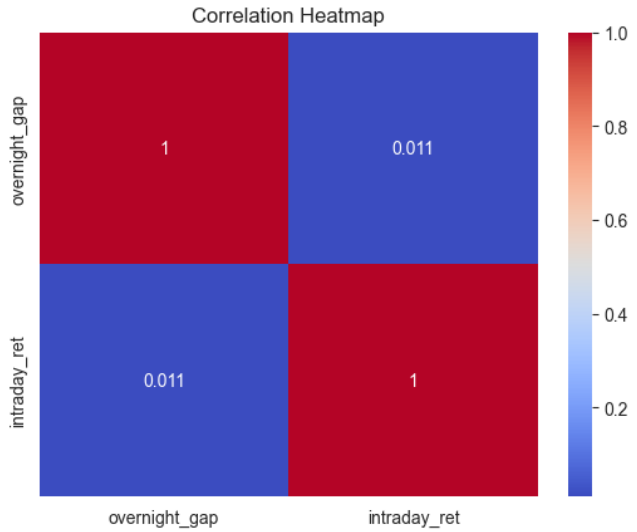


- While most companies show a concentrated return distribution around 0%, several firms (e.g., ZS, AAPL) exhibit long tails, indicating more frequent extreme gains and losses.
- The wider spread of some violins signals higher volatility, while narrow ones suggest more stable return profiles.

- Key Takeaway: Return risk is not uniform across firms — certain companies consistently face heavier tail risks, making them more prone to sudden price shocks.

9.6 CORRELATION HEATMAP (OVERNIGHT GAP VS INTRADAY RETURN)

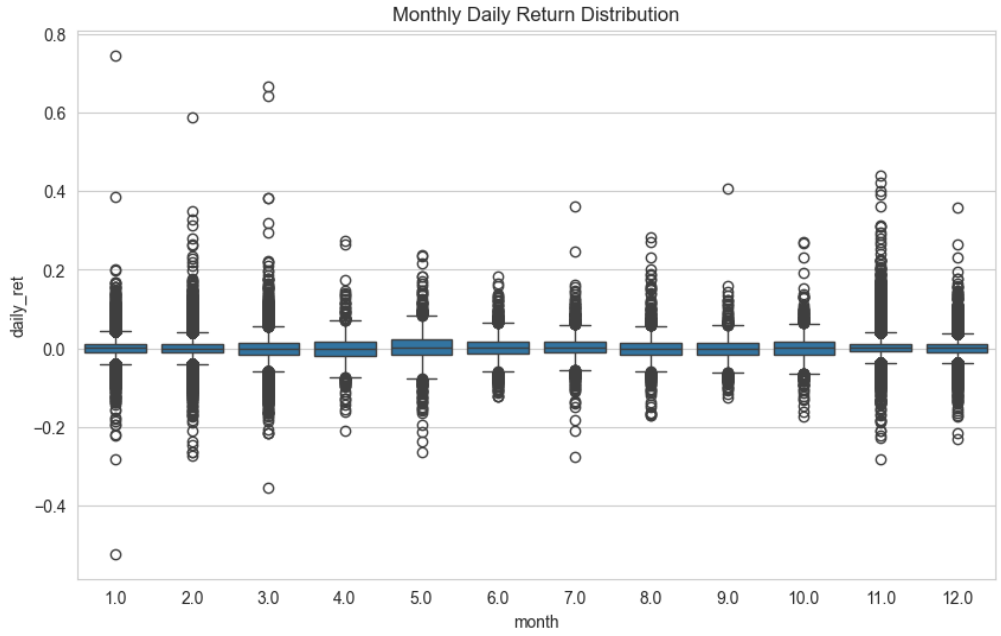
The correlation heatmap compares overnight gaps with intraday returns.



- The near-zero correlation coefficient (0.011) suggests there is no strong linear relationship between gap size and subsequent intraday movement.
- This aligns with statistical test results (RQ2), where overnight gaps had weak predictive power for daily trends.
- Key Takeaway: Overnight price gaps may reflect sentiment or after-hours news, but they do not reliably predict the next day's direction in a linear sense.

9.7 MONTHLY DAILY RETURN DISTRIBUTION

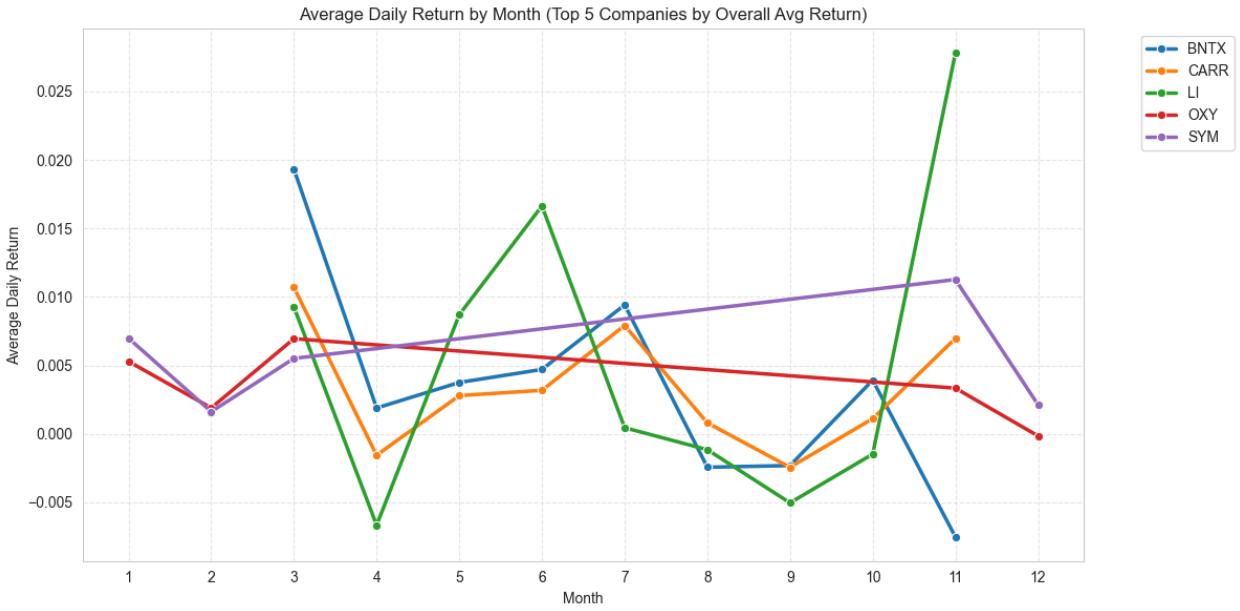
The box plot shows how daily returns are distributed across months of the year.



- Outliers are present in most months, with certain months (e.g., March, November) showing larger spikes in daily returns.
- The interquartile ranges are fairly consistent, but some months demonstrate wider spreads, pointing to seasonal volatility differences.
- Key Takeaway: Return behavior varies across months, supporting evidence of seasonality in stock performance. Investors may face systematically different risks depending on the calendar month.

9.8 AVERAGE DAILY RETURN BY MONTH

The line chart plots average monthly returns for the top 5 companies ranked by overall performance.



- Some companies (e.g., LI in November) show sharp spikes, while others (e.g., SYM) exhibit relatively consistent positive averages.
- The divergence across companies highlights idiosyncratic seasonal effects — not all firms follow the same return cycle.
- Key Takeaway: Certain companies experience stronger seasonal return patterns than others, which may present selective opportunities for timing strategies.

10 DATA MODELING & DATA EVALUATION

Two main categories of modeling approaches were employed, statistical inference models and machine learning classification models. Together, they allowed both hypothesis-driven testing and predictive performance evaluation.

Statistical Methods: The various methods are used to validate the different research questions with respect to their hypothesis.

- RQ1: Levene's and Fligner's tests for variance equality; kurtosis and % extreme-move calculations.
- RQ2: Chi-square test (gap vs. outcome) and logistic regression (odds ratios).
- RQ3: OLS regression (abs intraday return ~ lagged volume z-score) and Mann–Whitney U test (spike vs. non-spike days).
- RQ4: Kruskal–Wallis tests across weekdays and months (returns normalized per company).

Machine Learning Models

- Logistic Regression: interpretable baseline.
- Random Forest Classifier: nonlinear predictive model.
- Evaluation metrics: AUC, Accuracy, Brier score, Log-loss.

Validation Approach: Two types of validation approaches adopted, first the model is trained and tuned with rolling cross-validation and later final performance was assessed on the last 20% of the sample, simulating a real out-of-sample forecast.

- TimeSeriesSplit (per company): Rolling splits for walk-forward CV, simulating real-world forecasting.
- 80/20 Chronological Holdout: Final out-of-sample evaluation using the last 20% of the time series.
- This dual approach ensures both robust model development and unbiased evaluation.

Backtesting

A naive "gap rule" strategy is tested to check if overnight gap (long) > 0, if gap (short) < 0. Performance is evaluated with CAGR, Sharpe ratio, and maximum drawdown, including simple turnover costs.

The data modeling and data evaluation is carried out using python code.

10.1 DATA PREPARATION & FEATURE ENGINEERING

As part of data preparation, imported required modules and libraries into the code. The cleaned dataset "adjusted_dataset.csv", which was the output of the part 1 is loaded into the python.

The below features created:

- Daily return: close-to-close percentage change.
- Intraday return: open-to-close percentage change.
- Overnight gap: prior close-to-next open percentage change.
- Volume z-scores and spike indicators (rolling 20-day baseline).
- Lagged volume features (to avoid look-ahead bias).
- Calendar features (weekday, month).
- Target: target_up = 1 if intraday return > 0, else 0.

10.2 METHODOLOGY

The methodology below is adopted for model building and model evaluation:

- Statistical inference models (Logit, OLS, Chi-square, KW tests) addresses the hypothesis-testing aspect of the research questions.

- Machine learning classification models (Random Forests, Logistic Regression classifiers) used to validate the predictive aspect, by checking if signals could forecast stock movements.
- Together, they provided a comprehensive framework:
    - Statistical inference for explanation.
    - Machine learning for prediction.

Statistical inference models

These models are used to test hypotheses (H_0 compared with H_1) and measure the statistical significance of explanatory factors.

- Logistic Regression (Logit Model):
    - Used to test the overnight gap could significantly predict the probability of positive intraday returns.
    - The outputs coefficients, p-values, and odds ratios are used to understand the gap variable greater than odds ratio >1, which indicates positive gaps are likely to mean a higher probability of increasing intraday prices.
- Ordinary Least Squares (OLS) Regression:
    - Used to test lagged z-scores in the volume which can explain the magnitudes of intraday returns.
    - The outputs of this model are regression coefficients and p-values.
- Non-parametric Tests (Chi-square, Mann–Whitney U, Kruskal–Wallis):
    - Chi-square test used to test independence between categorical gap direction and intraday trend.

- Mann–Whitney U test used to test distributional differences in the magnitudes of returns following volume spikes.
- Kruskal–Wallis test used to test seasonal differences by days of the week/months.

Modeling - Machine Learning Models

These models are used for predictive modeling, which is a progression beyond significance testing to see if the historical data could predict outcomes in the correct direction.

- Random Forest Classifier (RF):
  - A non-linear, ensemble based supervised learning approach.
  - Multiple decision trees are created by adding more randomization, in terms of bootstrapping samples and features, to mitigate overfitting.
  - The random forest classifier features are Overnight gap & Lagged volume z-score
  - It is used to predict whether the intraday return (target_up) is in the positive or negative direction.
- Evaluation Metrics:
  - AUC (Area Under the ROC Curve) ability is to discriminate between an up, versus down days.
  - Fraction of days can be predicted accurately.
  - Brier Score provides the calibration for probabilistic predictions.
  - Log-loss penalizes wrong predictions which are too confident.

Validation Techniques

- TimeSeriesSplit Cross-Validation (walk-forward CV):

- Performed at the company level to simulate rolling, real-time forecasting.
- Explicitly separates past and future data to avoid look-ahead bias.

- 80/20 Chronological Holdout:
  - The last 20% of the dataset is reserved for final testing.
  - Provides an unbiased measure of predictive power, mimicking out-of-sample forecasting.

10.3 RESULTS & HYPOTHESIS TESTING

With the execution of the code the summary of the results is listed below:

RQ1 – Return Profile Differences

- Levene's Results: p = 7.65e-57, Fligner's p = 5.77e-161 → Reject H_0.
- Remarks: Companies exhibit significant heterogeneity in variance and tail risk.
- Example extremes: ZS (variance = 0.0055, kurtosis = 252.8), AAPL (variance=0.0014, kurtosis=196.3).
- Conclusion: RQ1 supports H1, i.e the return profiles differ, with some companies showing much heavier tails.

RQ2 – Overnight Gaps Predicting Intraday Direction

- Chi-square p = 0.152 → Fail to reject H_0 (no strong independence signal).
- Logit: overnight gap coefficient = -0.86, OR = 0.42, p = 0.125 → not significant.
- Random Forest CV: median AUC ≈ 0.49, accuracy ≈ 0.50 (near-random).

80/20 Holdout:

- Logit: AUC = 0.478, ACC = 0.527
- RF: AUC = 0.501, ACC = 0.514
- Conclusion: RQ2 supports H0, i.e the overnight gaps show limited/mixed predictive power. The models perform close to random guessing.

RQ3 – Volume Spikes and Price Volatility

- OLS: coefficient(vol_zscore_lag1) = +0.0013, p = 3.39e-32.
- Mann–Whitney: p = 4.59e-10 → significant distributional shift.
- Conclusion: RQ3 supports H1, i.e the higher prior volume is strongly linked to larger intraday price moves.

RQ4 – Seasonality Effects

Kruskal–Wallis:

- Weekday p = 0.0152
- Month p = 1.03e-07
- Conclusion: RQ4 supports H1, i.e the returns differ significantly across weekdays and months, confirming seasonal patterns.

Backtest – Naive Gap Rule

- CAGR = –10.21%
- Sharpe ratio = –0.72
- Max drawdown = –26.35%

- Conclusion: Although gaps are statistically observable, the naive gap-trading rule is not profitable and underperforms once costs are included.

10.4SUMMARY OF RESULTS

The below points outline the summary of the test results:

- RQ1 confirms company-level heterogeneity: risk and extreme events are not uniform across tickers.
- RQ2 shows that although overnight gaps are intuitively linked to next-day sentiment, their predictive value is weak in practice.
- RQ3 provides strong evidence that volume contains information about subsequent volatility — spikes in trading activity foreshadow larger moves.
- RQ4 reveals seasonal patterns, consistent with behavioral finance findings about day-of-week and month effects.
- The backtest demonstrates that statistical signals do not automatically translate into profitable trading strategies without further refinement.

11 CONCLUSION

This project set out to explore whether historical stock market data can be used to understand and predict short-term stock performance trends. Four research questions (RQ1–RQ4) guided the analysis, focusing on return profiles, overnight gaps, trading volume spikes, and seasonal effects. Both statistical hypothesis testing and machine learning models were applied, complemented by visual exploration in Tableau and Python.

The findings can be summarized as follows:

- Return Profiles Differ Across Companies (RQ1): Statistical tests strongly rejected the null hypothesis of uniform return distributions.
- Overnight Gaps Show Limited Predictive Power (RQ2): while overnight gaps reflect after-hours sentiment, they do not provide robust predictive signals on their own.
- Volume Spikes Drive Higher Volatility (RQ3): volume is a reliable indicator of short-term volatility, aligning with market microstructure theories of information arrival.
- Seasonality Effects Are Statistically Significant (RQ4): calendar effects exist, though they vary by company.

12 REFERENCES

- Tableau. (n.d.). *Tutorial: Get started with Tableau Desktop,* Retrieved from https://help.tableau.com/current/guides/get-started-tutorial/en-us/get-started-tutorial-home.htm
- Tableau. (n.d.). *Data visualization*. Tableau. https://www.tableau.com/resource/data-visualization
- Tableau. (n.d.). *Create dashboards*. Tableau. https://help.tableau.com/current/pro/desktop/en-gb/dashboards_create.htm
- The Python Tutorial, https://docs.python.org/3/tutorial/index.html
- VanderPlas, J. (n.d.). *Python Data Science Handbook: Data Manipulation with Pandas*. https://jakevdp.github.io/PythonDataScienceHandbook/