Advanced Stock Market Dashboard

using R Shiny

Vijay Jadhav

Suyash Jindal

Rachit Jindal

Venkatesh Tejavath

Guidance of Dr. Aakash Anand

MTH208 - Data Science Lab I

Indian Institute of Technology, Kanpur

Submission: November 2025

Contents

Executive Summary 2

1 Approach to the Project 2

2 Introduction 2

3 Data for the Project 2
- 3.1 Primary Data Sources . . . 2
- 3.2 Data Fields and Period . . . 3
- 3.3 Ethics and Licensing . . . 3

4 Data Extraction and Cleaning 3
- 4.1 Retrieval Pipeline . . . 3
- 4.2 Preprocessing Steps . . . 3
- 4.3 Data Validation . . . 3

5 Technical Analysis Visualizations 3
- 5.1 Candlestick and Volume Charts . . . 4
- 5.2 Moving Averages (SMA and EMA) . . . 4
- 5.3 Relative Strength Index (RSI) . . . 4
- 5.4 Bollinger Bands . . . 5
- 5.5 MACD . . . 5

6 Portfolio Optimization 5
- 6.1 Mathematical Formulation . . . 5
- 6.2 Numerical Implementation . . . 6

7 Options Trading Analysis 7
- 7.1 Option Chain Visualization . . . 7
- 7.2 Black–Scholes–Merton (BSM) Model . . . 7

8 Limitations and Ethical Considerations 8
- 8.1 Model and Data Limitations . . . 8
- 8.2 Ethical considerations . . . 8

9 Reproducibility 8
- 9.1 Code and Environment . . . 8
- 9.2 How to run . . . 8
- 9.3 Files and structure (recommended) . . . 8

10 Conclusion and Future Work 9
- 10.1 Conclusion . . . 9
- 10.2 Future extensions . . . 9

11 Acknowledgements 9

12 References 9

Executive Summary

This project presents an **Advanced Stock Market Dashboard** built using `R Shiny` and The dashboard provides interactive technical analysis of stocks using OHLCV data from public sources such as Yahoo Finance and NSE India.
It features candlestick charts with key indicators including SMA, EMA, RSI, Bollinger Bands, and MACD, allowing users to visualize trends, momentum, and volatility dynamically. All computations and visualizations are implemented using R packages like `quantmod`, `tidyquant`, `TTR`, and `plotly`. Developed as part of the **Data Science Lab–I (MTH208)** course, the project emphasizes reproducibility, interactivity, and ethical use of open financial data, showcasing how `R Shiny` can integrate statistical computing and visualization for real-time market insights.

1 Approach to the Project

Problem Statement

Modern retail traders and students often switch between multiple tools to gather technical indicators, fundamentals, and portfolio analytics. This fragmentation slows analysis and reduces reproducibility. The aim is to produce a single interactive R Shiny application that integrates visualization, technical indicators, statistical trend signals, and portfolio optimization for educational and exploratory use.

Objectives

- Build a responsive R Shiny dashboard with interactive candlestick visualizations and technical overlays.
- Implement standard indicators: SMA, EMA, RSI, MACD, Bollinger Bands.
- Integrate mean-variance portfolio optimization and produce an efficient frontier plot.
- Provide benchmark comparisons and backtesting visuals for selected portfolios.
- Ensure reproducibility: provide instructions, code organization and package list.

2 Introduction

Financial markets produce vast amounts of time-series data. Traders use technical indicators to summarize short-term momentum and mean-variance theory to allocate across assets to balance return and risk. R and the Shiny framework are well-suited to create reproducible, interactive dashboards combining statistical computation and web-based presentation. This project is designed for students and novice investors to interactively explore price behavior and the risk-return trade-off.

Key motivations:

- Consolidate multiple analysis steps (data retrieval, indicator computation, visualization, optimization) into one reproducible workflow.
- Provide clear, interpretable visual signals for educational purposes.
- Create a deliverable that can be extended for research or classroom demonstration.

3 Data for the Project

3.1 Primary Data Sources

The dashboard uses only publicly available data:

- **Yahoo Finance** via `quantmod::getSymbols` and `tidyquant::tq_get`: OHLCV daily series per ticker.
- **NSE India** website (where applicable) for local tickers / validation.

3.2 Data Fields and Period

For each selected symbol we collect:
- Date, Open, High, Low, Close, Adjusted Close, Volume
- Derived fields: daily log returns, rolling volatility

Typical data horizon used in examples: last 5 years (adjustable via date picker).

3.3 Ethics and Licensing

Only public market quotes are used; no proprietary or private data is accessed. Users of the dashboard should be aware that outputs are educational and not financial advice.

4 Data Extraction and Cleaning

4.1 Retrieval Pipeline

Data retrieval is implemented using reactive calls in Shiny:
1. User inputs ticker and date range.
2. The server fetches OHLCV via `quantmod::getSymbols` or `tidyquant::tq_get`.
3. Results are cached for repeated queries to reduce API calls.

4.2 Preprocessing Steps

- **Missing values:** linear interpolation for isolated NA's, forward-fill for short gaps. For extended gaps the dashboard warns the user.
- **Adjusted prices:** where available, adjusted close is used for return computation to account for corporate actions.
- **Return series:** log-returns computed as $r_t = \log(P_t/P_{t-1})$.
- **Outlier treatment:** extreme returns beyond a threshold can be flagged but not automatically removed — decisions left to analyst.

4.3 Data Validation

Spot checks compare fetched time-series with Yahoo Finance web snapshots for a handful of tickers to ensure correctness. Timestamps are localized, and missing trading days (holidays) are handled using business-day alignment.

5 Technical Analysis Visualizations

This section outlines the single-stock analysis module and the suite of technical indicators integrated into the dashboard. The visualization components are designed to provide traders and learners with an intuitive, interactive understanding of market behavior through dynamic charts and overlays.

5.1 Candlestick and Volume Charts

The foundation of the visualization module is an interactive candlestick chart complemented by volume bars. This plot serves as the primary interface for examining price dynamics and market activity. Key features include:

- Interactive zooming, panning, and tooltips for precise data exploration (implemented via `plotly` or `highcharter`).
- Overlay of multiple moving averages to highlight trend directions.
- Hover-based display of detailed OHLCV values for granular inspection.

These features collectively enable users to examine price trends and trading volumes in a seamless, data-rich environment.

Figure 1: Interactive candlestick and indicator visualization from the dashboard.

5.2 Moving Averages (SMA and EMA)

Simple and exponential moving averages (SMA, EMA) are calculated for standard window lengths of 20, 50, 100, and 200 days. These averages are superimposed on the candlestick chart to help users identify short- and long-term trend patterns.

Implementation Details

- **SMA:** Computed as `TTR::SMA(close, n)`.
- **EMA:** Computed as `TTR::EMA(close, n)`.
- **Crossover Detection:** The dashboard highlights *golden* (bullish) and *death* (bearish) crosses, such as the 50/200-day crossover, as visual trading signals.

5.3 Relative Strength Index (RSI)

The Relative Strength Index (RSI) is calculated using a 14-day look-back period to quantify momentum and potential reversal zones. The RSI is visualized in a subpanel beneath the price chart, with reference thresholds at 30 and 70 marking oversold and overbought regions respectively. This allows users to identify periods of market exhaustion and potential entry or exit points.

Figure 2: RSI subpanel plot.

Figure 3: RSI subpanel plot.

Figure 4: RSI subpanel plot.

Figure 5: RSI subpanel plot.

5.4 Bollinger Bands

Bollinger Bands (20-day SMA $\pm$ 2 standard deviations) are included to visualize volatility expansion or squeeze.

5.5 MACD

MACD(12,26,9) is computed and displayed as a histogram + signal line to highlight momentum shifts.

6 Portfolio Optimization

The dashboard supports constructing a portfolio from user-selected tickers (3–5 assets by default) and computes mean-variance optimal allocations.

6.1 Mathematical Formulation

Let μ be the vector of expected returns (sample mean of log returns) and Σ the sample covariance matrix. The classical Markowitz mean-variance optimization has two equivalent formulations:

Formulation 1 (Minimize Risk for Target Return):

$$\min_{w} \ \frac{1}{2} w^\top \Sigma w \quad \text{s.t. } w^\top \mu = \mu_p, \quad w^\top \mathbf{1} = 1, \quad w \geq 0$$

Formulation 2 (Maximize Return for Target Risk):

$$\max_{w} \ w^\top \mu - \frac{\lambda}{2} w^\top \Sigma w \quad \text{s.t. } w^\top \mathbf{1} = 1, \quad w \geq 0$$

where μ_p is the target portfolio return and λ is the risk aversion parameter. The efficient frontier is traced by solving either formulation for a range of μ_p values (first formulation) or λ values (second formulation). The point with maximum Sharpe ratio $(\mu_p - r_f)/\sigma_p$ represents the tangency portfolio, where r_f is the risk-free rate.

6.2 Numerical Implementation

Optimization uses `quadprog` or `PortfolioAnalytics` in R. Shorting is disabled by default (weights ≥ 0). The app returns:

- Optimal weights for selected points on the frontier
- Expected portfolio return and volatility
- Sharpe ratio relative to a user-specified risk-free rate

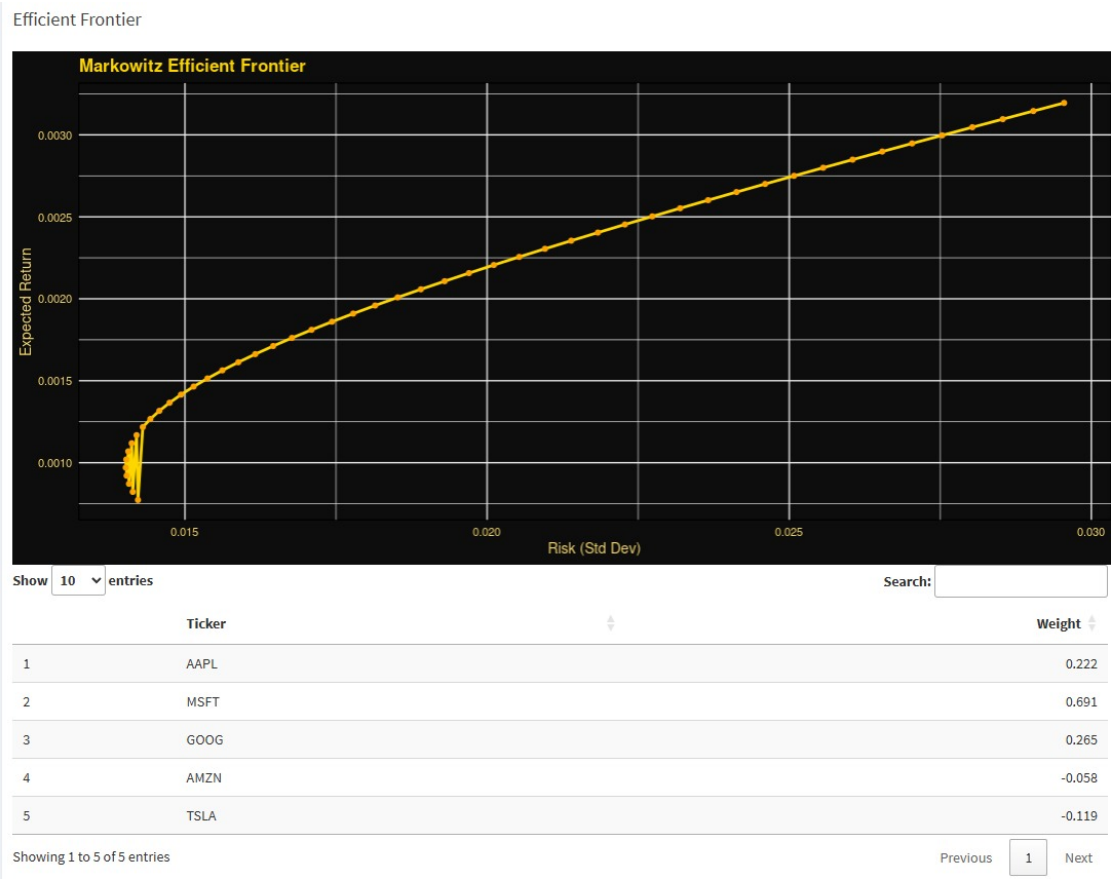


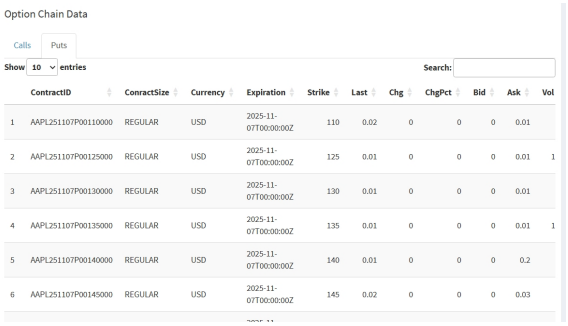
Figure 6: Efficient frontier and sample portfolios. Replace with the generated plot from your optimization module.

7 Options Trading Analysis

This section outlines the design and implementation of the Options Trading Analysis module, encompassing the Black–Scholes–Merton pricing framework and interactive Option Chain visualization for comprehensive derivative evaluation

7.1 Option Chain Visualization

- Displays a structured Option Chain showing strike prices, bid–ask premiums, implied volatilities, and expiry dates for selected underlying assets
- Highlights in-the-money (ITM), at-the-money (ATM), and out-of-the-money (OTM) options, helping users quickly interpret moneyness and relative pricing users to compare options across strikes and maturities interactively, supporting strategy evaluation and market sentiment analysis in real time.


Option Chain Data

Calls Puts

Show 10 entries Search:

	ContractID	ConractSize	Currency	Expiration	Strike	Last	Chg	ChgPct	Bid	Ask	Vol
1	AAPL251107P00110000	REGULAR	USD	2025-11-07T00:00:00Z	110	0.02	0	0	0	0.01	
2	AAPL251107P00125000	REGULAR	USD	2025-11-07T00:00:00Z	125	0.01	0	0	0	0.01	1
3	AAPL251107P00130000	REGULAR	USD	2025-11-07T00:00:00Z	130	0.01	0	0	0	0.01	
4	AAPL251107P00135000	REGULAR	USD	2025-11-07T00:00:00Z	135	0.01	0	0	0	0.01	1
5	AAPL251107P00140000	REGULAR	USD	2025-11-07T00:00:00Z	140	0.01	0	0	0	0.2	
6	AAPL251107P00145000	REGULAR	USD	2025-11-07T00:00:00Z	145	0.02	0	0	0	0.03	

Figure 7: Option chain data.

7.2 Black–Scholes–Merton (BSM) Model

- Implements the Black–Scholes–Merton (BSM) model to calculate European call and put option prices using key inputs such as spot price, strike price, volatility, risk-free rate, and time to expiry.
- Computes the five principal Greeks—Delta, Gamma, Vega, Theta, and Rho—to evaluate how option prices respond to changes in market conditions like price movement, volatility, and time decay.
- Enables users to understand option behavior and risk exposure through quantitative outputs and interactive visualization, enhancing practical understanding of derivative pricing.

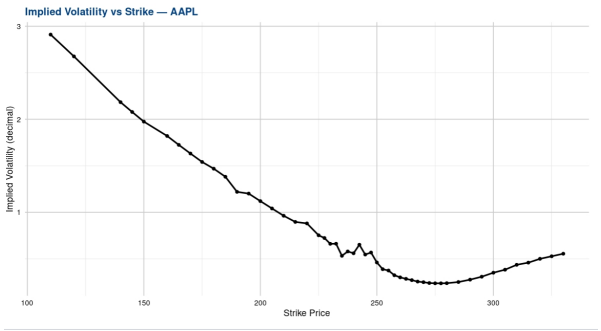


Figure 8: Implied volatility vs strike.

8 Limitations and Ethical Considerations

8.1 Model and Data Limitations

- **Historical bias:** All forecasts and optimizations are based on historical returns and covariances which may not hold in future regimes.
- **No transaction costs:** Backtests ignore trading costs and slippage; including them would reduce net returns and affect rebalancing choices.
- **Survivorship bias:** Care must be taken when selecting historical universes to avoid excluding delisted or merged securities.
- **Data quality:** Free public APIs can have outages or small discrepancies; caching and validation steps mitigate but do not eliminate these problems.

8.2 Ethical considerations

- The dashboard is strictly for educational and research use; it is not investment advice.
- Users should verify results and exercise caution before making financial decisions.
- All data sources are cited and publicly accessible; no private user data is collected or stored.

9 Reproducibility

9.1 Code and Environment

- Language: **R** (version 4.x recommended).
- Core packages: `shiny`, `quantmod`, `tidyquant`, `TTR`, `plotly`, `PerformanceAnalytics`, `PortfolioAnalytics`, `quadprog`.
- Recommended: use `renv` or containerization (Docker rocker/shiny) to pin package versions and ensure reproducibility.

9.2 How to run

1. Clone repository from GitHub (insert your repo link).
2. Open RStudio and restore environment: `renv::restore()` (if renv used).
3. From project root, launch the app: `shiny::runApp("app.R")`.
4. For batch generation of figures (for report), run the scripts in scripts/ (e.g., Rscript scripts/generate_figures.R).

9.3 Files and structure (recommended)

```
project-root/
  app.R
  R/
    modules/
    utils.R
  data/
    raw/
    cached/
  scripts/
  README.md
  requirements.txt
```

10 Conclusion and Future Work

10.1 Conclusion

This project demonstrates that an integrated R Shiny dashboard can consolidate technical analysis and portfolio optimization into a single interactive tool suitable for teaching and exploratory research. The interface enables rapid, reproducible experiments with indicators, allocations and backtests.

10.2 Future extensions

Possible improvements include:

- Incorporating machine learning models (LSTM, XGBoost) for price or regime prediction.
- Adding options analytics (implied volatility surfaces, Greeks).
- Modeling transaction costs and slippage for more realistic backtests.
- Implementing user authentication and persistent portfolios for multi-session studies.
- Streaming real-time quotes and alerts (via websockets) for near real-time monitoring.

11 Acknowledgements

We sincerely thank **Dr. Aakash Anand** for guidance throughout the project and the course instructors and TAs for feedback during development. The R community and CRAN package authors provided critical building blocks that made this project feasible.

12 References

- CRAN: `quantmod`, `tidyquant`, `TTR`, `PerformanceAnalytics`, `PortfolioAnalytics`.
- Yahoo Finance API: https://finance.yahoo.com/
- Hull, J. *Options, Futures and Other Derivatives*.
- Bodie, Z., Kane, A., Marcus, A. *Investments*.
- Relevant course notes and MTH208 materials.

Prepared by: Vijay Jadhav, Suyash Jindal, Rachit Jindal, Venkatesh Tejavath.
Course: Data Science Lab I (MTH208A)
Submission: November 2025