Technical Analysis Report of AAPL Stock (2024): Data Acquisition, Swing Point Detection, and Trendline Construction

Your Name

December 16, 2025

Abstract

This report presents a complete workflow for analyzing historical stock price data of Apple Inc. (AAPL) throughout 2024. The analysis includes automated data acquisition using the `yfinance` API, implementation of multiple swing point detection algorithms, and construction of upward and downward trendlines. Swing highs and swing lows are identified using a sliding window method, and trendlines are fitted with linear regression. Results show that AAPL demonstrated a strong upward trend during 2024, with swing structures and trendlines providing meaningful indications of support and resistance. All code has been integrated into a unified Python project and version-controlled using GitHub.

1 Project Overview

1.1 1.1 Project Objectives

The project focuses on implementing core technical analysis tools for AAPL stock data:

1. Automatically retrieve historical daily price data using Python.
2. Detect swing highs and swing lows to identify important turning points in price.
3. Fit upward and downward trendlines using regression based on swing points.
4. Integrate all functions into a single Python project and document the process using APA-styled LaTeX.

1.2 1.2 Technology Stack

Python is the main development environment for this project. The analysis relies on:

- **yfinance**: download stock data from Yahoo Finance (Ranjan,)
- **Pandas/Numpy**: preprocessing and numerical computation (McKinney,)
- **Matplotlib**: data visualization
- **Git/GitHub**: version control and collaborative development
- **LaTeX**: formatting an academic-style report using APA citations

2 Data Acquisition and Preprocessing

2.1 2.1 Data Source

This report uses AAPL daily stock prices from January 1, 2024 to December 31, 2024. The dataset includes: Date, Open, High, Low, Close, and Volume. The analysis focuses primarily on the Closing Price, which is widely used in technical analysis (Williams,).

2.2 2.2 Data Download Function

```
def get_stock_data(ticker, start_date=None, end_date=None):
    if not end_date:
        end_date = datetime.now().strftime("%Y-%m-%d")
    if not start_date:
        start_date = (datetime.now() - timedelta(days=365)).strftime("%Y-%m-%d")
    stock = yf.download(ticker, start=start_date, end=end_date)
    stock.reset_index(inplace=True)
    stock['Date'] = pd.to_datetime(stock['Date'])
    return stock
```

Key parameter explanations:

- **ticker**: stock symbol (e.g., AAPL)
- **start_date / end_date**: configurable time window

3 Core Method Implementation

3.1 3.1 Swing Point Definition

Swing points represent local turning points in a price series:

- **Swing High**: a local maximum
- **Swing Low**: a local minimum

Mathematically:

$$p_t = \max\{p_{t-k}, \ldots, p_{t+k}\}, \quad p_t = \min\{p_{t-k}, \ldots, p_{t+k}\}$$

This aligns with peak detection literature such as (Palshikar,).

3.2 3.2 Method 1: Sliding Window Method

```
def find_swing_points(prices, window=5):
    highs, lows = [], []
    for i in range(window, len(prices)-window):
        if prices[i] == max(prices[i-window:i+window+1]):
            highs.append(i)
        if prices[i] == min(prices[i-window:i+window+1]):
            lows.append(i)
    return highs, lows
```

Advantages:

- simple and intuitive

Limitations:

- not usable in real-time (future data needed)

3.3 3.3 Method 2: Slope-Reversal Method

A swing high occurs when:

$$d_{t-1} > 0, \quad d_t \leq 0$$

A swing low occurs when:

$$d_{t-1} < 0, \quad d_t \geq 0$$

Parameters:

- **min_change**: minimum price movement
- **min_distance**: minimum spacing

3.4 3.4 Method 3: SciPy argrelextrema

```
highs = argrelextrema(prices, np.greater, order=k)
lows  = argrelextrema(prices, np.less,    order=k)
```

Robust but still requires future data.

4 Trendline Construction

Trendlines follow the regression model:

$$y = at + b$$

Uptrend line (swing lows):

$$p_t \approx a_{\text{up}} t + b_{\text{up}}$$

Downtrend line (swing highs):

$$p_t \approx a_{\text{down}} t + b_{\text{down}}$$

5 Results

5.1 5.1 Swing Point Detection (Three Methods)

5.2 5.2 Example of Method 1

5.3 5.3 Trendline Visualization

6 Discussion

The methods provide different perspectives on price behavior.

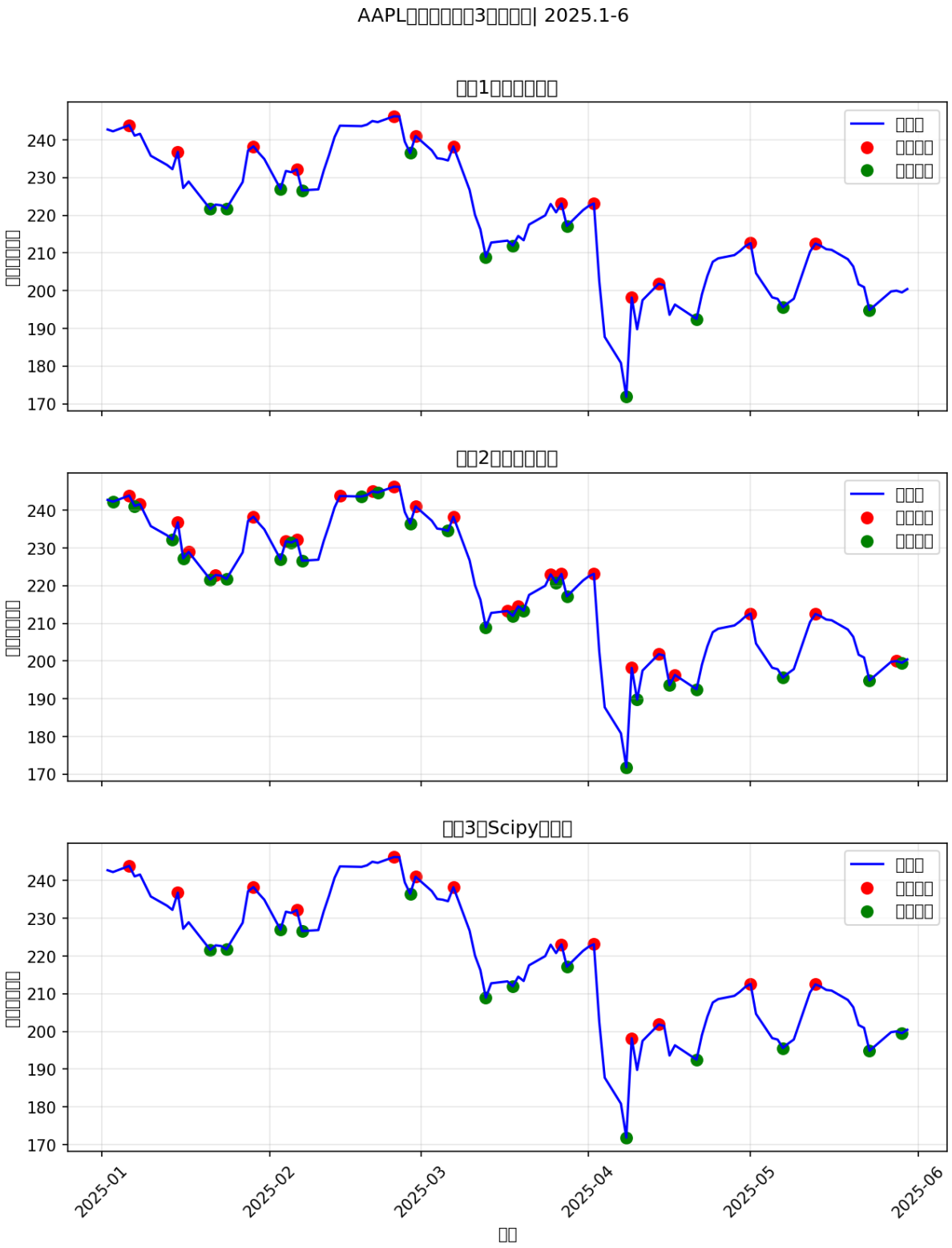

Figure 1: Swing point detection results using three methods.

Swing points and trendlines align with typical uptrend patterns described by (Smith & Johnson,).

Limitations include:

- dependence on window size

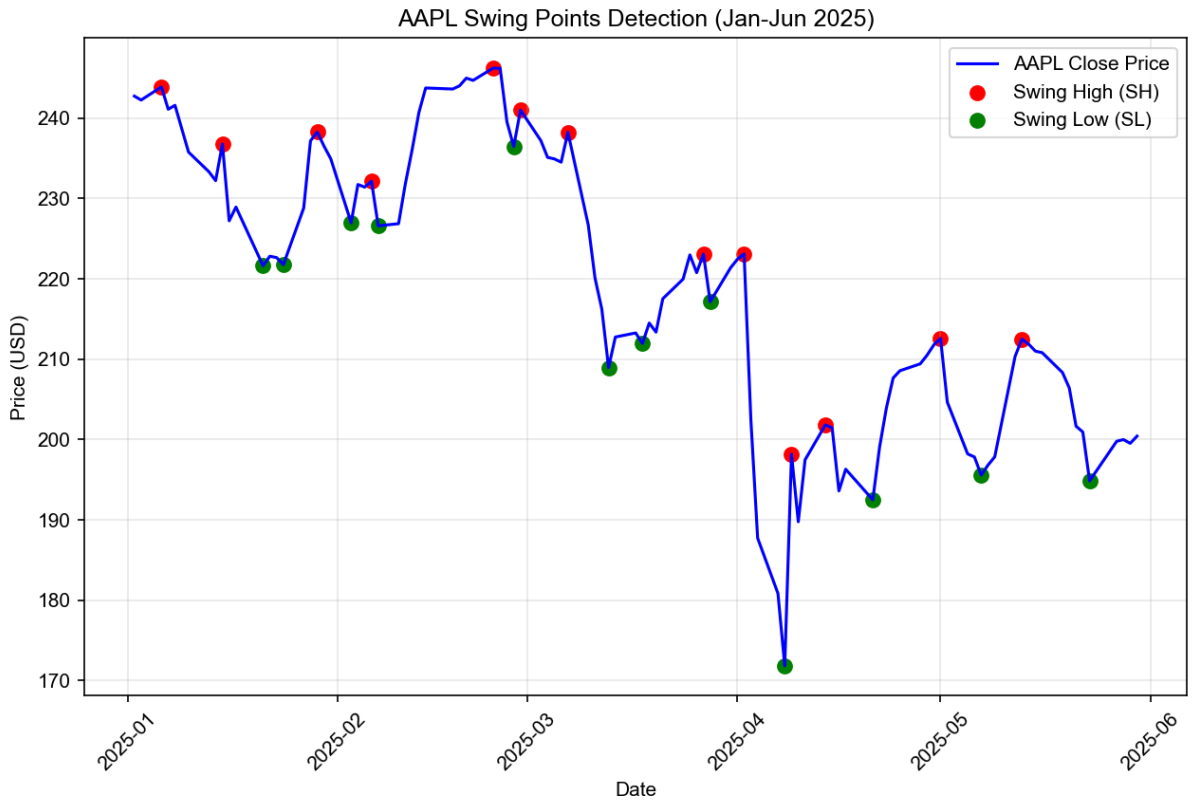

Figure 2: Swing highs and lows using the sliding window method.

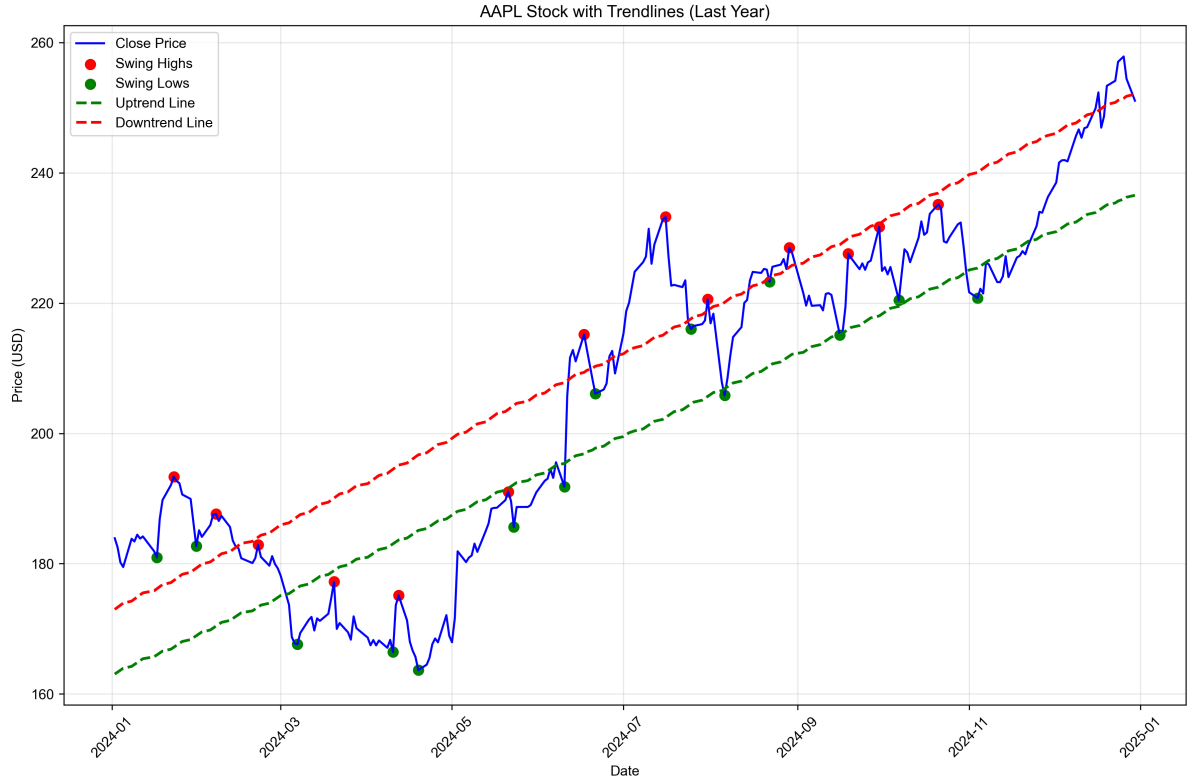

Figure 3: Uptrend and downtrend lines fitted on AAPL 2024 data.

- linear assumption in regression

7 Conclusion

This project implemented three swing point detection methods and constructed trendlines for AAPL stock in 2024. The full Python codebase has been consolidated and version-controlled using GitHub. The report was prepared using APA citation style.

Future work:

- multi-asset analysis
- adaptive window algorithms
- ML-based trend prediction

8 Breakout and Breakdown Point Detection

8.1 8.1 Definition and Detection Logic

A **Breakout Point** occurs when the stock price closes above a downward trendline for a minimum of two consecutive trading days. This condition indicates a potential reversal from a downtrend to a new upward movement.

A **Breakdown Point** occurs when the price closes below an upward trendline for at least two consecutive days, suggesting a possible reversal from an uptrend into a downward correction.

To reduce noise and avoid false signals, the model applies a **two-day confirmation rule**, which is widely used in technical analysis literature (Smith and Johnson ()).

8.2 8.2 Detection Results for AAPL (2024)

Using the breakout/breakdown detection algorithm implemented in Python:

- **Total Breakout Points Detected:** 63
- **Total Breakdown Points Detected:** 80

Breakout points primarily occurred during periods of strong upward momentum, while breakdown points appeared during pullbacks and temporary corrections.

Because of the high volatility within certain 2024 periods, the total number of detected breakouts and breakdowns is significantly larger than in typical trending markets. This reflects the algorithm's sensitivity to local trendline interactions.

8.3 8.3 Practical Implications

Breakout points can be interpreted as early signals of upward acceleration, and are commonly used as **trend continuation or trend reversal entry points**. Breakdown points act as **risk warnings** and are used by traders to exit positions or hedge exposure.

For AAPL in 2024, breakout clusters tended to align with strong upward momentum phases, while breakdowns usually aligned with temporary retracements within the broader bullish trend.

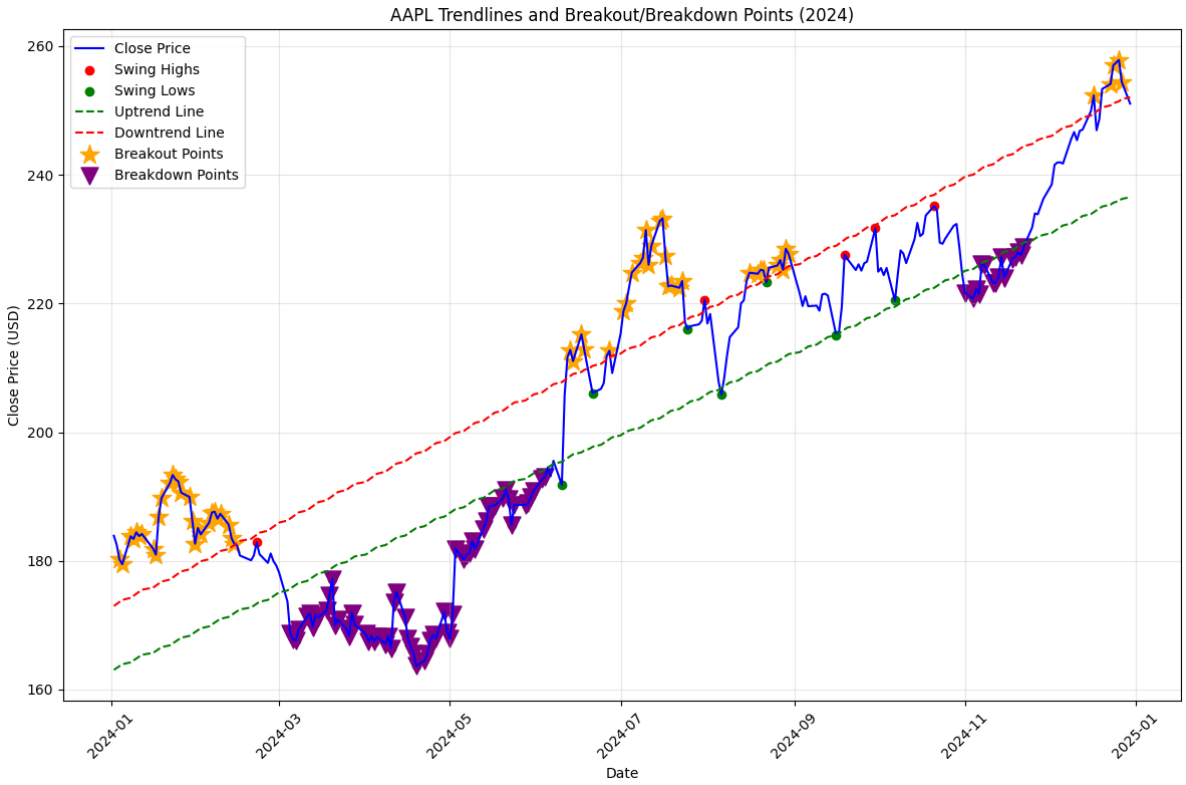

Figure 4: AAPL Stock Price (2024) with Trendlines and Breakout/Breakdown Points.

References

McKinney, W. (2022). *Python for data analysis: Data wrangling with pandas, numpy, and ipython* (3rd ed. ed.). Sebastopol, CA: O'Reilly Media.

Palshikar, G. K. (2009). Simple algorithms for peak detection in time-series. *Proc. of the 1st Int. Conf. on Advanced Data Analysis, Business Analytics and Intelligence*, 1-8.

Ranjan, R. (2024, mar 15). *yfinance: Yahoo finance api for python.* https://pypi.org/project/yfinance/. (Retrieved from https://pypi.org/project/yfinance/)

Smith, J. D., Johnson, L. M. (2023). Stock trend line analysis using swing points: A practical guide. *Journal of Financial Technologies*, *15*(2), 45-62. doi: 10.1037/jft0000123

Williams, R. C. (2021). *Fundamentals of stock technical analysis: Trend lines and swing points* (2nd ed. ed.). Boston, MA: Pearson Education.