Winter Quarter 2026 ORCA

Web-Based Pipeline for Standardized Event Window Stock Return Analysis and Research-Ready Visualization

Yung-Sian Fang

University of California, Riverside, A. Gary Anderson Graduate School of Management
yungsian.fang@email.ucr.edu

Motivation

"The event study is an important research tool in economics and finance." — MacKinlay (1997)

"Event studies are widely used in finance research…" — El Ghoul et al. (2022)

"Daily data generally present few difficulties for event studies." — Brown & Warner (1985)

- **Manual spreadsheet + ad hoc scripts → inconsistent cleaning/window definition**
- **Hard to reproduce charts & summaries → low transparency**
- **Collaboration friction → outputs not comparable across teams**
- **Barrier for non-technical users → method becomes inaccessible**

Methodology: Event-Window Return Computation

Return definition

Let denote the closing price of stock on trading day . The simple daily return is computed as:

$$R_{i,t} = \frac{P_{i,t} - P_{i,t-1}}{P_{i,t-1}}$$

For reporting convenience, the platform converts returns to percentages:

$$\text{PctChange}_{i,t} = 100 \times R_{i,t}$$

Window definition（pre/post）

Given an event date , the current platform version extracts a pre-event and post-event window and computes average daily returns within each window. For a symmetric five-day specification:

$$\text{Pre-event window: } [t_0 - 5, t_0 - 1]$$
$$\text{Post-event window: } [t_0 + 1, t_0 + 5]$$

The average return in each window is:

$$\bar{R}_{\text{pre}} = \frac{1}{5}\sum_{k=1}^{5} R_{i,t_0-k}$$

System Overview

1

Input

Upload CSV（one/multi stocks） + pick event date

2

Pipeline

clean → parse dates → sort → compute returns → extract windows

3

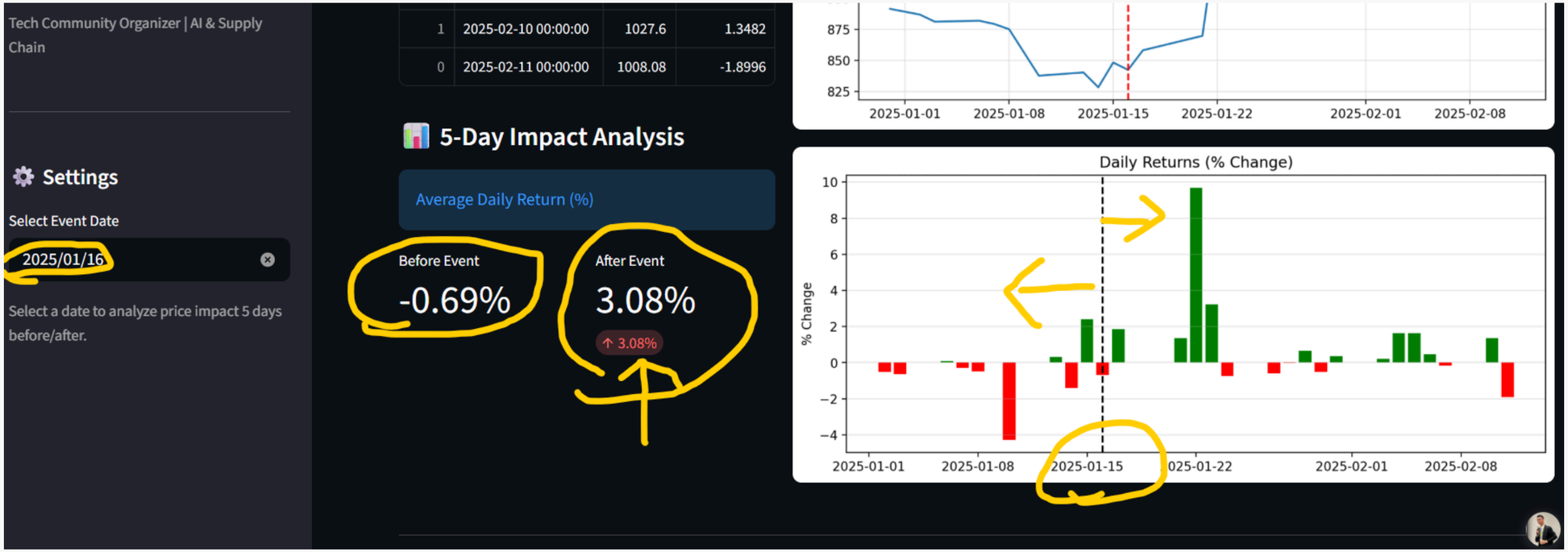
Outputs

pre/post summary + event-marked charts + export-ready visuals

```python
import pandas as pd
import matplotlib.pyplot as plt

def analyze_stock(csv_path, company_name, event_date_str=None):
    # 1. Read CSV and preprocess data
    df = pd.read_csv(csv_path)
    for col in ["Open", "High", "Low", "Close", "Volume"]:
        df[col] = df[col].str.replace(",", "").astype(float)
    df["Date"] = pd.to_datetime(df["Date"], format="%m/%d/%Y")
    df = df.sort_values("Date")

    # 2. Calculate the daily rate of return (% change)
    df["Pct_Change"] = df["Close"].pct_change() * 100

    # If there is a specified event date, then capture the 5 trading days before and after.
    if event_date_str is not None:
        event_date = pd.to_datetime(event_date_str)
        window = df.loc[(df["Date"] >= event_date - pd.Timedelta(days=10)) &
                        (df["Date"] <= event_date + pd.Timedelta(days=10))]
        before = window[window["Date"] < event_date].tail(5)
        after  = window[window["Date"] > event_date].head(5)

        print(f"{company_name} - around {event_date.date()}")
        print("Avg % change (5 days before):", before["Pct_Change"].mean())
        print("Avg % change (5 days after): ", after["Pct_Change"].mean())
        print()
```

Demo: Netflix case setup

- **Stock**: Netflix (daily OHLCV)
- **Period**: Jan–Feb 2025
- **Event date**: Jan16, 2025
- **Window spec**: ±5 trading days (excluding event day)
- **Output**: summary + charts with event marker



Key Results & Interpretation

5-Day Event Window Summary
(Netflix, Event Date: 2025-01-16)

- **Pre-event average daily return: -0.69%**
- **Post-event average daily return: +3.08%**
- **Spread (Post − Pre): +3.77 percentage points**

Event Evidence Timeline (real-world)

- 2024-12-13: Netflix announced it would release Q4 2024 financial results on Jan 21, 2025 (1:00 PM PT).
- 2025-01-21: Netflix posted Q4 results / outlook and held the earnings interview.
- 2025-01-21 to 01-22: Media reported stock surge following record subscriber additions and upgraded guidance.

What likely drove repricing (from reported disclosures)

- Record Q4 subscriber additions (about 19M) and stronger-than-expected results were highlighted in coverage.
- Netflix also raised 2025 revenue outlook and expanded buyback authorization, which support positive sentiment.

Interpretation for chart

- The user-selected event date (1/16) captures a post-event reaction window that includes the major repricing around the actual earnings release (1/21).
- This demonstrates why a platform should support event-date validation / alternative event anchors (e.g., announcement date vs. market reaction date).

Conclusion & RoadMap

Takeaways

- Web-based, standardized pipeline reduces friction & inconsistency
- Produces reproducible summaries + research-ready visuals
- Works as a front-end screening layer before formal event-study inference

Future work

- Abnormal returns (market-adjusted / market model)
- CAR / AAR / CAAR + inference (parametric & nonparametric tests)
- Multi-stock panel events + benchmark integration + auto-report export

- Abnormal return :

$$AR_{i,t} = R_{i,t} - E(R_{i,t} \mid X_t)$$

- Market model :

$$E(R_{i,t}) = \alpha_i + \beta_i R_{m,t}$$

- Cumulative abnormal return :

$$CAR_i(t_1, t_2) = \sum_{t=t_1}^{t_2} AR_{i,t}$$

Winter Quarter 2026 ORCA

Thank You

Web-Based Pipeline for Standardized Event Window Stock Return Analysis and Research-Ready Visualization

Yung-Sian Fang

University of California, Riverside, A. Gary Anderson Graduate School of Management
yungsian.fang@email.ucr.edu