Final Project

STAT 184

David Goldberg, Jasmine Randhawa, Rishi Lal

2025-12-17

Table of contents

Introduction

The focus of our group project is to look into how Apple's supplier companies stock performances affect Apple's stock. We decided to look at 4 of Apple's main suppliers: Foxconn, Pegatron, Compal, and TSMC. All of these companies are based in Taiwan, and they also supply companies other than Apple.

Since Apple is one the worlds most valuable companies, its supply chain directly impacts its production and in turn its market performance. By examining how these key suppliers' stock movements relates to Apple's stock price, we hope to gain predictive insights into Apple's financial performance. Our analysis is quite relevant given how recent global supply chain disruptions and the increasing importance of semiconductor manufacturing impacts the tech industry.

The Dataset

We acquired our data for our project from the tidyquant R package. The tidyquant package is used to retrieve data about the stock market from the web (more specifically, Yahoo Finance).

Our dataset consists of daily adjusted closing prices for Apple (AAPL) and four major suppliers from January 1, 2022 to December 16, 2025. We chose these suppliers since they play critical roles in Apple's supply chain: Foxconn (Hon Hai Precision) handles final assembly, TSMC manufactures advanced chips, while Pegatron and Compal provide additional manufacturing capacity. Companies like Samsung and LG also supply Apple with parts, but we didn't choose they also manufacture their own products and are competitors with Apple.

The time period covers major events in the global economy, including: post-pandemic recovery, supply chain normalization, and the recent AI investment boom. This gives us a broad view of different market conditions, so we can analyze performance based on different scenarios.

FAIR and CARE Principles

The data that we pulled from the tidyquant package follows FAIR principles, but not CARE principles. The FAIR principles state that data must be findable, accessible, interoperable, and reusable; while the CARE principles emphasis that data should provide collective benefit, authority to control, responsibility, and ethics.

Since our data comes from the tidyquant package, it's both findable and accessible since anyone with RStudio can quickly find, load, and then use the data. The data is also interoperable as it the vocabulary and formats that it uses works with other tools, such as tidyverse and ggplot2.

Tidyquant can be used by others in context outside of our project, thus making it reusable as well. Thus, our data follows the FAIR principles.

The CARE principles are people-focused, as opposed to data-focused. Information about the stock market from tidyquant provides collective benefit as it can inform people about how stocks work. It is also responsibly collected and it is ethical to collect information about stock performance. However, it does not have authority to control since only the government (at least in the US) has jurisdiction over the stock market.

Methods

To compare stocks that have different price levels, we normalized each stock's performance by calculating its percent change in relation to its starting value on January 1st, 2022. By doing this, we can compare relative performance more easily. Apple's stock trades around $275, while Foxconn's stock trades around $13. This makes direct price comparison between Apple and its suppliers meaningless.

We calculated normalized percentage change using the formula: Percent Change = ((Adjusted Price / First Adjusted Price) - 1) * 100

We also computed annual returns by comparing each stock's adjusted closing price at the beginning and the end of each year. By splitting this up by year, we could see how the companies performed based on different market conditions and business cycles.

Our analysis includes both visual comparisons through line charts and bar plots, as well as tables that gave summaries of recent trading activity and stock performance. By doing this, we can look to see if there's any correlation between Apple's stock performance and its suppliers.

Results

Tables

Apple's Recent Trading Data

This table provides information about Apple's most recent trading activity, including the daily open, high, low, close, and trading volume for the last five trading days. This lets us see short-term fluctuations in Apple's stock price.

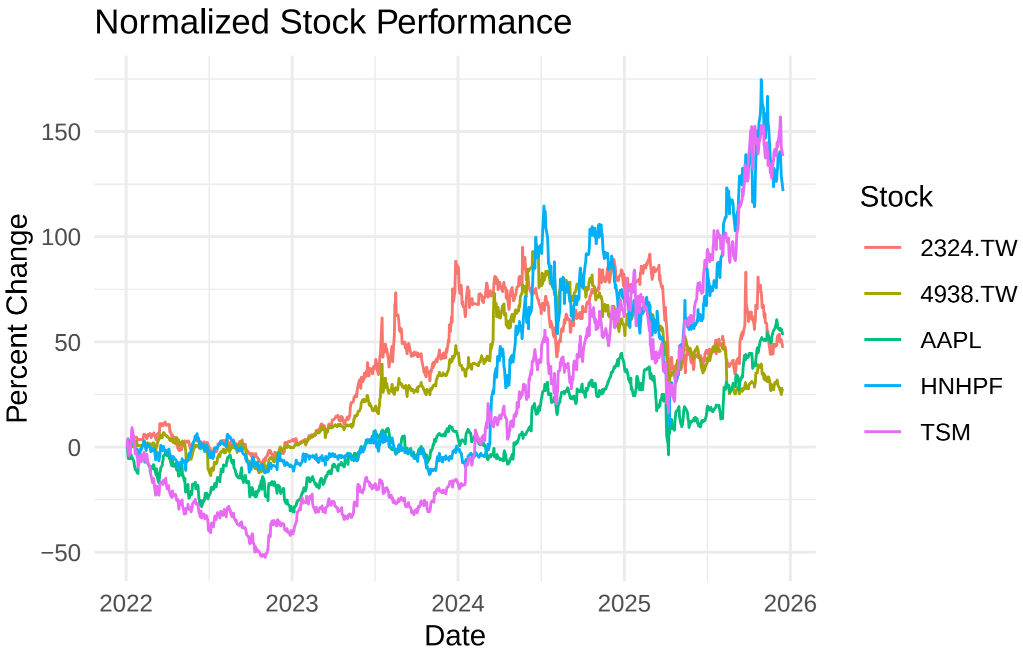
During this time, Apple's stock remained stable and fluctuated from $271 to $280. The most notable event was when Apple opened at $280.15 on December 15th, and then dipped to $274.11 at the end of the day. December 15th was also when the highest volume occurred, with over 50 million shares. Still, Apple's stock remains quite stable and only shows minor ups and downs.

Table 1: Table 1: Recent Apple (AAPL) Stock Performance

Date	Open	High	Low	Close	Volume
December 10, 2025	277.75	279.75	276.44	278.78	33038300
December 11, 2025	279.10	279.59	273.81	278.03	33248000
December 12, 2025	277.90	279.22	276.82	278.28	39532900
December 15, 2025	280.15	280.15	272.84	274.11	50409100
December 16, 2025	272.82	275.50	271.79	274.61	37648600

Normalized Percent Change Comparison Table

This table shows how Apple and it's key suppliers' stock prices changed over time, by using normalized percent change. By setting each stock's starting value to 0%, we can compare their movements, despite the fact that the stock prices are quite varied.



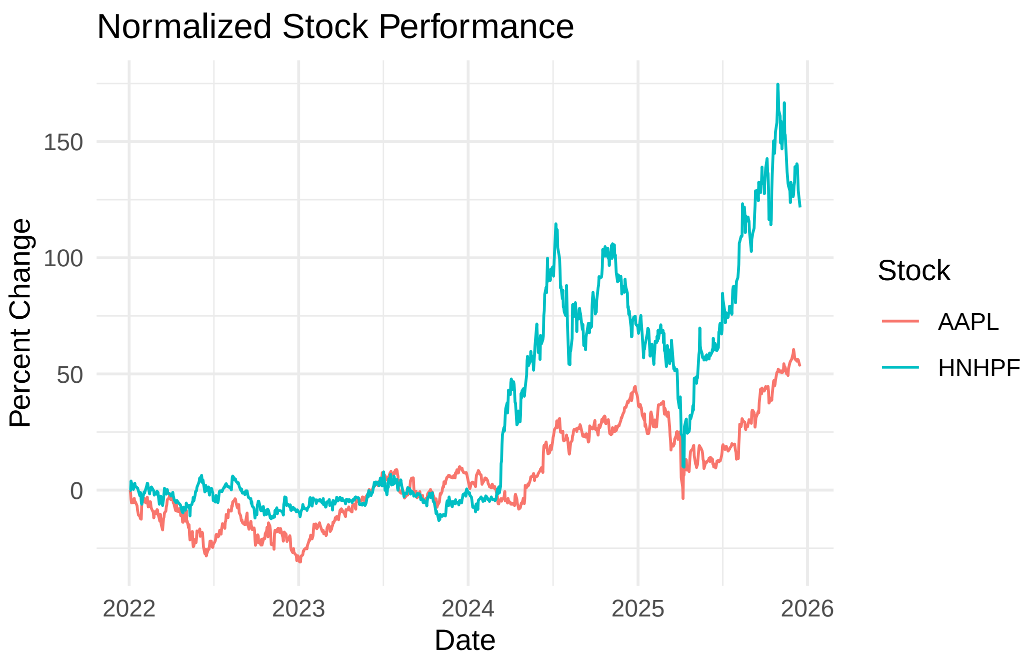


Table 2: Supplier Performance affects on Apple stock

symbol	date	adjusted	percent_change
AAPL	2022-01-03	178.270294	0
HNHPF	2022-01-03	6.279001	0
TSM	2022-01-03	120.327004	0
4938.TW	2022-01-03	55.112526	0
2324.TW	2022-01-03	19.877842	0

Annual Total Returns

This table summarizes the total yearly return from 2022 to 2025 for Apple, Foxconn, TSMC, Compal, and Pegatron. The returns are showed as percentages, and are calculated from the difference between the first and last adjusted closing price of the year. Doing this lets us see the difference in performance between the companies, regardless of their stock price.

2022 was the worst year, with Apple and most of its suppliers having negative returns. Only Compal had a small positive return of +3.0%. In 2023, companies started to improve with Apple gaining 54.8% and Compal rising by 80.2%. During 2024, stock performance was mixed but still mainly positive. TSMC and Foxconn did very well, with total returns of 97.3% and 73.3%. However, Compal started to slip with a return of -1.7%. In 2025, returns were more moderate with only Apple, Foxconn, and TSMC having decent positive returns (32.4%, 32.4%, 44.2%). However, both Pegtron and Compal declined.

Apple's performance moved in the same direction as its suppliers, but to varying extents. 2023 and 2024 were strong years that likely had widespread industry growth, while external events in 2022

Table 3: Annual Total Returns for Apple and Suppliers

Symbol	Year	Total Return (%)
2324.TW	2022	3.0%
4938.TW	2022	-0.4%
AAPL	2022	-28.2%
HNHPF	2022	-8.8%
TSM	2022	-40.9%
2324.TW	2023	80.2%
4938.TW	2023	44.7%
AAPL	2023	54.8%
HNHPF	2023	11.7%
TSM	2023	43.2%
2324.TW	2024	-1.7%
4938.TW	2024	9.0%
AAPL	2024	35.6%
HNHPF	2024	73.3%
TSM	2024	97.3%
2324.TW	2025	-13.9%
4938.TW	2025	-16.4%
AAPL	2025	13.1%
HNHPF	2025	32.4%
TSM	2025	44.2%

and 2025 (COVID, AI boom) led to some suppliers struggling, even when Apple had a positive performance.

Visualizations

Normalized Performance Line Chart

The normalized line chart shows how Apple and its suppliers' stocks have performed since their starting values in 2022.

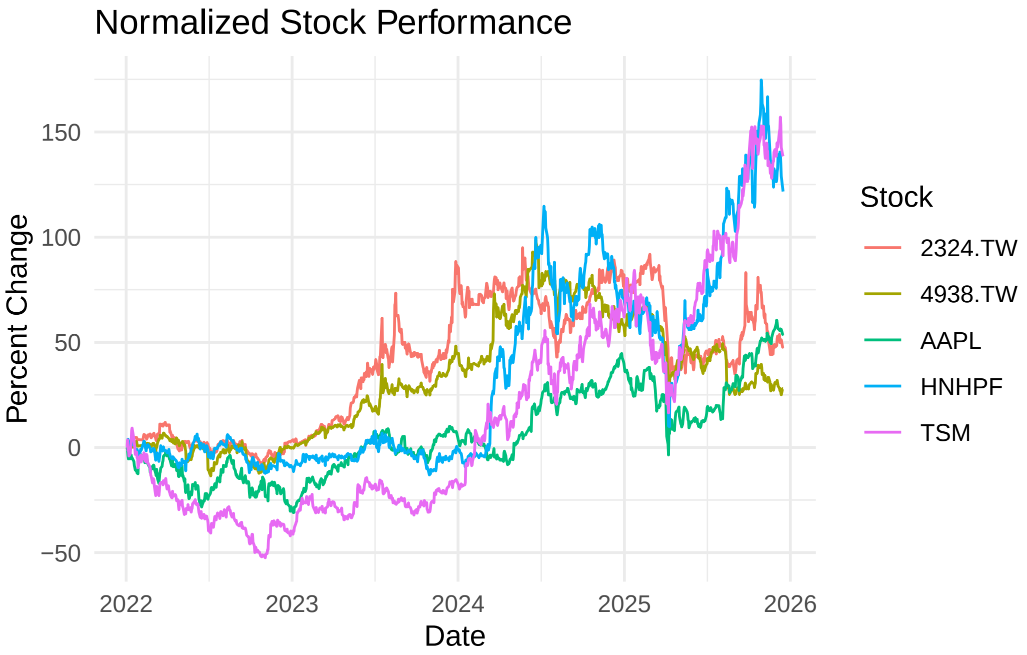
TSMC (TSM) had the greatest overall growth, and had a steep upward trend that began in 2024. However, TSMC also had the worst normalized stock performance from 2022 until 2024. This suggest that TSMC was likely hit the hardest of the suppliers by COVID-19, and it's strong performance came as a result from increased demand for semiconductors and its role in the current AI boom.

Foxconn (HNHPF) also demonstrates significant growth from 2024 and onward. It's percent change is very close to TSMC and it's stock didn't drop as much as TSMC from 2022 to 2024. This is likely due to Foxconn being Apple's main supplier for assembly.

Pegatron (4938.TW) and Compal (2324.TW) had moderate growth, which then dipped in 2025. This suggests that they have a slower recovery compared to TSMC and Foxconn and are likely not involved with AI.

Apple (AAPL) remains relatively stable and has had a decent upward trend. It's much less volatile than its suppliers, which is likely due to its presence in multiple markets and that Apple is an established company.

The difference in stock performance shows that supplier stocks can outperform Apple's stock in relative terms, especially in growth phases. Currently, AI is being heavily invested in and supplier companies that produce things for AI can experience a lot of growth. TSMC and Foxconn are the most dynamic of the supplier stocks, and can offer predictive insight for Apple's performance.

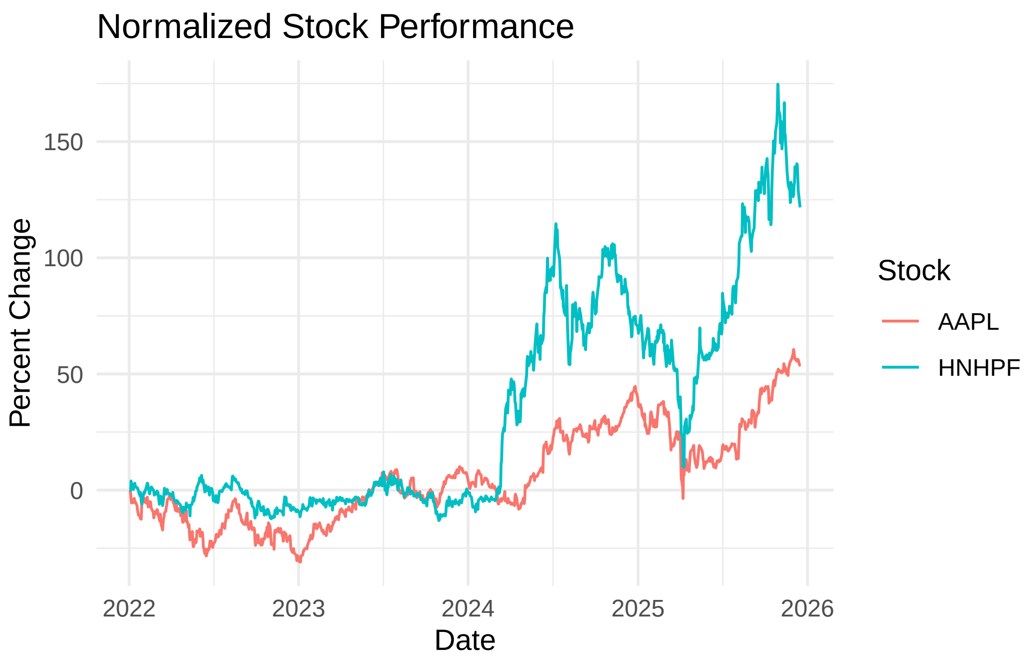


Normalized Apple and Foxconn Stock Performance

Looking at the previous graph, we concluded that both Foxconn and TSMC could have predictive value for Apple. We decided to look into the relationship between Foxconn and Apple since Foxconn is a larger supplier to Apple than TSMC.

Foxconn is more volatile than Apple, and starting in early 2024 has much sharper swings in percent change. At this time, its percent change also surpasses Apple's. Despite having a lower stock price, Foxconn has stronger relative growth.

The difference in volatility between Apple and Foxconn indicates that Foxconn is more sensitive to demand and supply chain pressures. Foxconn's performance can potentially change before Apple's. Foxconn trended upwards before Apple followed suit in both early 2024 and 2025, which supports the idea that supplier stocks can be leading indicators in performance. However, Apple's steadiness and modest fluctuations also implies that supplier volatility doesn't always directly translate into Apple's performance.



Annual Growth Line Plot

This visualization shows how Apple's stock performance compares with it's suppliers every year. We plotted cumulative returns for each company, and reset their performance to 0% at the start of each year. This way, we can look at how each stock changes across different market conditions.

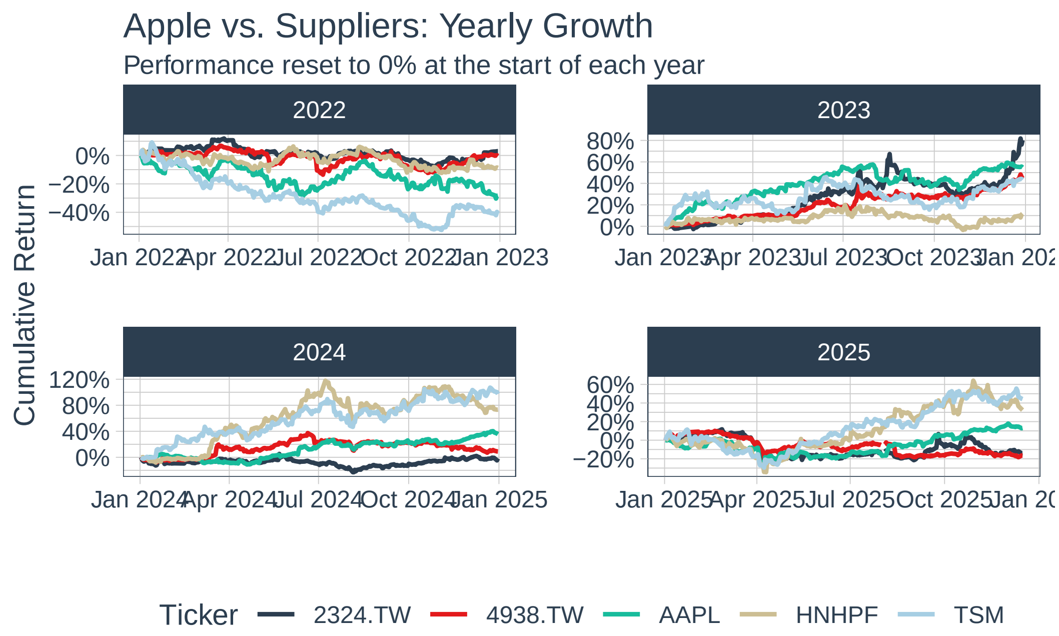
In 2022, none of the stocks performed particularly well. Cumulative returns ranged from around 0% to -40% at the end of the year.

2023 is when most of the companies started recovering and experiencing strong growth. Compal had the most growth, with cumulative returns at the end of the year reaching 80%. Foxconn, however, remained flat this year despite Apple's surge. This suggests that not all of the suppliers immediately respond to Apple's growth.

Unlike in 2023, Foxconn had a dramatic rise in 2024, which peaked around 120% in the middle of the year before it stabilized. This likely reflects either an increase in production or a shift in its operations. In 2024, TSMC also continued its upward growth from the previous year while Apple, Compal, and Pegatron's growth remained more flat and stable.

In 2025, both Compal and Pegatron's performance dipped and they are both ending the year with a cumulative return of around -20%. Apple maintained a mild and steady growth, while TSMC and Foxconn experienced stronger growth (despite a dip earlier in the year).

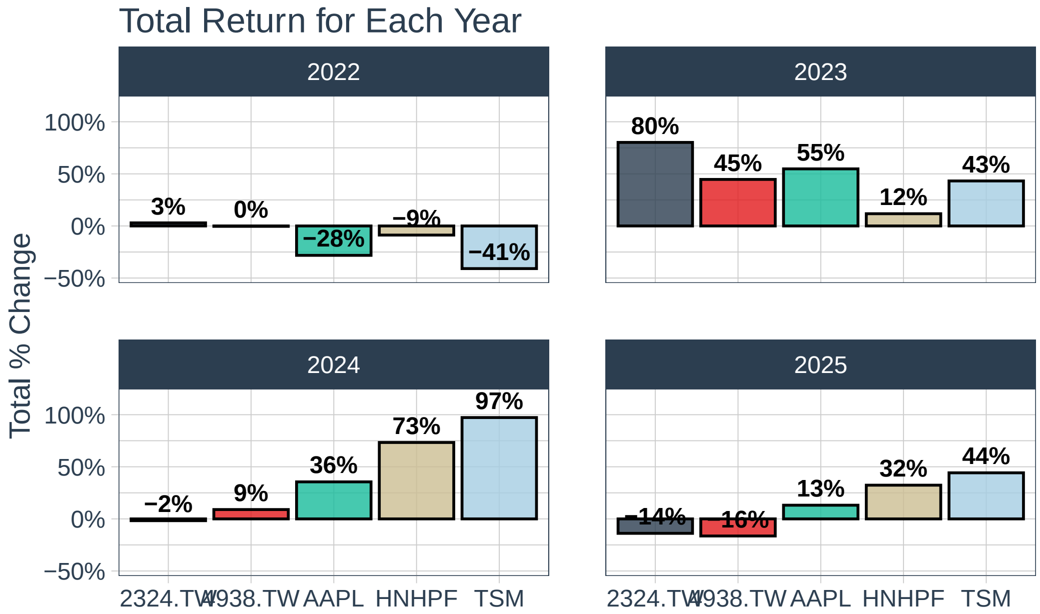
By looking at these plots, we can see that TSMC consistently tracks Apple's performance while also often having stronger returns. Foxconn's growth was more delayed than TSMC's but still managed to have strong growth in 2024 and 2025, which suggests that it responds to Apple's demand cycles more slowly (which could offer predictive value). Pegatron and Compal's performances are less aligned with Apple, which indicates that they have less influence on Apple's stock or are less responsive.



Annual Returns Bar Chart

The bar chart below summarizes the total annual return for Apple and its four main suppliers from 2022 to the present. Each bar represents the percent change in returns from the beginning to the end of the year.

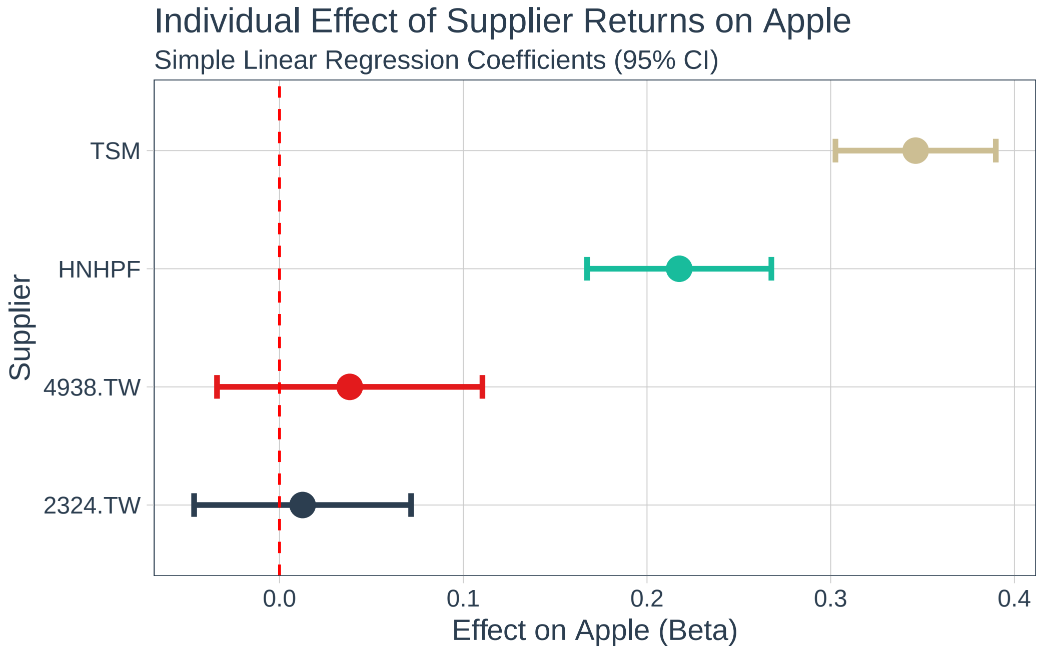
2022 was a difficult year for every company, which correlates to how the tech market was still recovering from the effects of COVID-19, and the fears of a recession. 2023 was when the Apple and its supplier companies started to rebound. Compal and Apple had the strongest performance. In 2024, growth slowed for Pegatron and Compal while Apple maintained solid growth.This also coincided with the popularity of AI, of which microchips is an essential component,so, companies like TSMC and Foxconn surged, likely due to increased demand or changes in operations. During 2025, Apple and its two main suppliers still grew, but at a more modest rate. Pegatron and Compal declined, likely due to a lack of demand and a lull in sales for non-AI technology.



Linear Regression Analysis

To quantify how much supplier stock performance influences Apple's stock price, we performed a linear regression. We modeled Apple's daily log returns as a function of the daily log returns of Foxconn, TSMC, Pegatron, and Compal.

The coefficient estimate (Beta) on the horizontal axis, represents the sensitivity of Apple's stock to each supplier. A higher estimate indicates a stronger positive relationship. The plot below visualizes these estimates with 95% confidence intervals. If the confidence interval (the bar) does not cross the red zero line, the relationship is statistically significant.



Based on the visualization, the linear regression analysis reveals a distinct hierarchy of influence among Apple's suppliers. The dots represent the estimated "Beta" coefficient, indicating the sensitivity of Apple's stock to each supplier's daily returns, while the horizontal bars show the 95% confidence interval, or the range of uncertainty around that estimate. TSMC (TSM) exhibits the strongest and most reliable relationship, indicated by a coefficient estimate positioned furthest to the right and a confidence interval that stays clearly above the red dashed horizontal line. Let's assume that this is 0.35 on the horizontal axis,this indicates On a day when TSMC's stock goes up by 1%, Apple's stock typically goes up by 0.35%. This statistically confirms that positive movements in TSMC's stock are significantly associated with gains in Apple's stock. In contrast, other suppliers like Foxconn (HNHPF) show a positive but potentially weaker or more variable effect (a wider bar). Smaller assemblers like Pegatron or Compal may have intervals that cross the zero line, suggesting their stock movements do not statistically predict Apple's performance.

Discussion

Relationship Between Apple and Suppliers

Our analysis shows that Apple's stock is less volatile compared to its suppliers. Apple's stability is likely due to brand loyalty, that it generates revenue from multiple sources, and that it has money reserved to act as a buffer against supply chain fluctuations. In contrasts, suppliers have more dramatic swings and are more sensitive to market changes. While Foxconn, TSMC, Pegatron, and Compal have some correlation with Apple's performance, the strength of their relationship is affected by market conditions.

Predictive Value

Foxconn and TSMC have the strongest predictive signals for Apple's stock. In early 2024, both companies had an upward trajectory weeks before Apple's similar positive trend. This suggests that supply chain improvements or an increase in production orders will first affect supplier stocks before affecting Apple's. However, this mainly applies to normal periods and doesn't account for when Apple launches new products. During those periods, consumer demand and innovation have more of an impact on Apple's stock performance than supply chain factors.

Limitations

There are several important limitations that have affected our analysis. First, we analyzed only four years of stock patterns in order to avoid dealing with outlying data that came as a result of the COVID-19 pandemic. However, by doing this we may have not been able to catch longer-term patterns in the market. Second, we aren't able to isolate Apple-specific demand from outside market forces that affect suppliers. TSMC had dramatic growth over the last few years due to a high increase in demand for AI chips from multiple companies, not just Apple. Third, Compal and Pegatron are listed on the Taiwanese stock market and are recorded in TWD (Taiwanese Dollars) instead of USD. Changes in the currency exchange rate could have impacted returns for their stocks. Finally, geopolitical tensions between Taiwan, China, and the US has led to Taiwanese companies being negatively affected. This could lead to contracts being renogotiated, especially with Pegatron and Compal which are smaller companies. Thus their recent decrease in performance could be more linked to external forces than Apple.

Conclusion

Our analysis suggests that while Apple's stock performance has correlation with key suppliers, the relationship is complex and influenced by external factors. Foxconn and TSMC provide positive indicators during periods of supply chain growth or changes within the tech industry. However, Apple's dominance and diversified revenue allows its stock to also be affected by consumer demand during product launches or other brand movements, not only its suppliers' performance.

Monitoring supplier stocks is still beneficial to investors. Following Foxconn and TSMC can give early insight or warning signals about Apple's production capacity and manufacturing expenses. However, these signals should be considered along with other factors like product innovation, consumer trends, public sentiment, and Apple's services revenue (e.g., AppleTV, Apple App Store). To build a more comprehensive predictive model could benefit from more research into earnings reports and estimates about production volumes.

MLA Citations

R

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, 2023. Web. 15 Dec. 2025.

dplyr

Hadley Wickham, Romain François, Lionel Henry, and Kirill Müller. *dplyr: A Grammar of Data Manipulation.* R package version 1.1.4, 2023. Web. 15 Dec. 2025.

tidyquant

Matt Dancho and Davis Vaughan. *tidyquant: Tidy Quantitative Financial Analysis.* R package version 1.0.8, 2023. Web. 15 Dec. 2025.

ggplot2

Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York, 2016. Web. 15 Dec. 2025.

scales

Hadley Wickham and Dana Seidel. *scales: Scale Functions for Visualization.* R package version 1.3.0, 2023. Web. 15 Dec. 2025.

lubridate

Garrett Grolemund and Hadley Wickham. *lubridate: Make Dealing with Dates a Little Easier.* R package version 1.9.3, 2023. Web. 15 Dec. 2025.

kableExtra

Hao Zhu. *kableExtra: Construct Complex Table with 'kable' and Pipe Syntax.* R package version 1.4.0, 2024. Web. 15 Dec. 2025.

knitr

Yihui Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R.* R package version 1.45, 2023. Web. 15 Dec. 2025.

lintr

Jim Hester, Michael Quinn, and Kun Ren. *lintr: A 'Linter' for R Code.* R package version 3.1.2, 2024. Web. 15 Dec. 2025.

Author Contribution

David: Data Manipulation, Data Visualization; Rishi: Narrative Text, Data Visualization; Jasmine: Narrative text, Data Visualization

Code Appendix

```r
# Rename columns to be capitalized for a professional look.
library(tidyquant)
library(tidyverse)
library(knitr)
library(kableExtra)
library(ggplot2)
library(scales) # Useful in formatting
library(lubridate) # Allows us to work with dates easily
library(dplyr)
library(broom)

apple_data <- tq_get("AAPL", get = "stock.prices", from = "2022-01-01")

table_sample <- apple_data %>%
  tail(5) %>%                                      # Select only the last 5 rows
  select(date, open, high, low, close, volume) %>% # Keep only essential columns
  mutate(date = format(date, "%B %d, %Y")) %>%     # Format date nicely (e.g., "Jan 01, 2024")
  rename(                                          # Rename columns for display
    "Date"   = date,
    "Open"   = open,
    "High"   = high,
    "Low"    = low,
    "Close"  = close,
    "Volume" = volume
  )

# 'scale_down' automatically shrinks the table if it's too wide for the PDF margin.
# 'hold_position' forces the table to appear exactly where you put the code, not floating away
kbl(table_sample, caption = "Table 1: Recent Apple (AAPL) Stock Performance", booktabs = T) %>
  kable_styling(latex_options = c("striped", "scale_down", "hold_position")) %>%
  column_spec(1, bold = TRUE) # Make the Date column bold
tickers <- c("AAPL","HNHPF", "TSM", "4938.TW", "2324.TW")

tickers_small <- c("AAPL", "HNHPF")

start_date <- "2022-01-01"
end_date   <- Sys.Date() # Today
```

```
stock_data <- tq_get(tickers,
                     get  = "stock.prices",
                     from = start_date,
                     to   = end_date)

stock_data_small <- tq_get(tickers_small,

                     get  = "stock.prices",
                     from = start_date,
                     to   = end_date)

#I normalized the data b/c APPL stock price is over 100 and HNHPF is less than 10,
#so we wont be able to see the couple cent changes in foxconn and comparison to the dollar char
#In other words HNHPF will look like a flat line even when APPL in visually moving up and down


normalized_data <- stock_data %>%
  select(symbol, date, adjusted) %>%
  group_by(symbol) %>%
  arrange(date) %>%
  mutate(
    percent_change = ((adjusted / first(adjusted) - 1)*100)
         )

normalized_data_small <- stock_data_small %>%
  select(symbol, date, adjusted) %>%
  group_by(symbol) %>%
  arrange(date) %>%
  mutate(
    percent_change = ((adjusted / first(adjusted) - 1)*100)
  )

# Line chart
ggplot(normalized_data, aes(x = date, y = percent_change, color = symbol)) +
  geom_line(linewidth = .5) +
  labs(title = "Normalized Stock Performance",
       x = "Date",
       y = "Percent Change",
       color = "Stock") +
  theme_minimal()


ggplot(normalized_data_small, aes(x = date, y = percent_change, color = symbol)) +
  geom_line(linewidth = .5) +
  labs(title = "Normalized Stock Performance",
       x = "Date",
       y = "Percent Change",
       color = "Stock") +
  theme_minimal()

normalized_data %>%
head(5) %>%
kable(
  caption = "Supplier Performance affects on Apple stock",
  align = c("l", rep("c",7))
)%>%
  kable_classic( # Pre-built styling
    font_size = 16, # Control Font Size
    lightable_options = "striped" # Adds striping in a pre-built table style
  )

# Annual Return Table
annual_summary <- stock_data %>%
  mutate(year = year(date)) %>%
  group_by(symbol, year) %>%
  summarize(
    total_return = (last(adjusted) - first(adjusted)) / first(adjusted),
    .groups = "drop"
  )

annual_return_table <- annual_summary %>%
  mutate(
    total_return_pct = scales::percent(total_return, accuracy = 0.1)
  ) %>%
  select(symbol, year, total_return_pct) %>%  # Keep only the formatted column
  arrange(year, symbol)

# Create a styled table
kbl(
  annual_return_table,
  caption = "Annual Total Returns for Apple and Suppliers",
  col.names = c("Symbol", "Year", "Total Return (%)"),
  align = c("c","c","c"),
) %>%
  kable_styling() %>%
  column_spec(1, bold = TRUE)   # Bolds the ticker column
ggplot(normalized_data, aes(x = date, y = percent_change, color = symbol)) +
  geom_line(linewidth = .5) +
  labs(title = "Normalized Stock Performance",
       x = "Date",
       y = "Percent Change",
       color = "Stock") +
  theme_minimal()
ggplot(normalized_data_small, aes(x = date, y = percent_change, color = symbol)) +
  geom_line(linewidth = .5) +
  labs(title = "Normalized Stock Performance",
       x = "Date",
       y = "Percent Change",
       color = "Stock") +
  theme_minimal()
# Prepare Cumulative Growth chart
yearly_growth <- stock_data %>% # Using the stock data from above.
  mutate(year = year(date)) %>% # mUtates the data, new column called year
  group_by(symbol, year) %>%
  mutate(
    # Reset growth to 0% at start of every year
    pct_change = (adjusted / first(adjusted)) - 1
  ) %>%
  ungroup()

line_plot <- ggplot(yearly_growth, aes(x = date, y = pct_change, color = symbol)) +
  geom_line(linewidth = 0.8) +
  facet_wrap(~ year, scales = "free", ncol = 2) +
  scale_y_continuous(labels = percent_format()) +
  theme_tq() +
  scale_color_tq() +
  labs(
    title = "Apple vs. Suppliers: Yearly Growth",
    subtitle = "Performance reset to 0% at the start of each year",
    y = "Cumulative Return",
    x = "",
    color = "Ticker"
  ) +
  theme(legend.position = "bottom")

print(line_plot) # This will result in the line plot for every year.
bar_plot <- ggplot(annual_summary, aes(x = symbol, y = total_return, fill = symbol)) +
  geom_col(color = "black", alpha = 0.8) +
  facet_wrap(~ year, ncol = 2) +
  geom_text(aes(label = percent(total_return, accuracy = 1)),
            vjust = -0.5, size = 3, fontface = "bold") +
  scale_y_continuous(labels = percent_format(), expand = expansion(mult = c(0.1, 0.2))) +
  scale_fill_tq() +
  theme_tq() +
  labs(
    title = "Total Return for Each Year",
    y = "Total % Change",
    x = ""
  ) +
  theme(legend.position = "none")
```

```r
print(bar_plot) # This will result in the bar plot for every year.
# Calculate Log Returns for Regression

returns_wide <- stock_data %>%
  group_by(symbol) %>%
  tq_transmute(select     = adjusted,
               mutate_fun = periodReturn,
               period     = "daily",
               type       = "log",
               col_rename = "returns") %>%
  ungroup() %>%
  pivot_wider(names_from = symbol, values_from = returns) %>%
  na.omit()

# List of suppliers to regress against AAPL
suppliers <- c("HNHPF", "TSM", "4938.TW", "2324.TW")

# Function to run individual simple linear regressions
run_model <- function(supp_ticker) {
  # Create formula: AAPL ~ Supplier
  # Use backticks for tickers that might have numbers/dots
  f <- as.formula(paste("AAPL ~ `", supp_ticker, "`", sep = ""))

  lm(f, data = returns_wide) %>%
    tidy(conf.int = TRUE) %>%
    filter(term != "(Intercept)") %>%
    mutate(term = supp_ticker) # Rename the term to the clean ticker name
}

# Run the model for each supplier and combine results into one table
model_results <- map_dfr(suppliers, run_model)

# Create the regression Plot
reg_plot <- ggplot(model_results, aes(x = term, y = estimate, color = term)) +
  geom_point(size = 4) +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high), width = 0.2, linewidth = 1) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  coord_flip() +
  theme_tq() +
  scale_color_tq() +
  labs(
    title = "Individual Effect of Supplier Returns on Apple",
    subtitle = "Simple Linear Regression Coefficients (95% CI)",
    y = "Effect on Apple (Beta)",
    x = "Supplier"
  ) +
  theme(legend.position = "none")
```

```
print(reg_plot)
# Style Guide: Tidyverse

# Load necessary libraries
library(dplyr)
library(tidyquant)
library(ggplot2)
library(scales) # Useful in formatting
library(lubridate) # Allows us to work with dates easily
library(kableExtra)
library(knitr)


# Add tickers for Apple, Foxconn, TSMC, Pegatron, and Compal
# AAPL    = Apple Inc.
# HNHPF   = Hon Hai Precision (Foxconn) - OTC US listing
# TSM     = Taiwan Semiconductor - NYSE ADR
# 4938.TW = Pegatron Corp - Taiwan Stock Exchange
# 2324.TW = Compal Electronics - Taiwan Stock Exchange

#| fig-alt: "A line chart showing the normalized percentage change in stock prices from 2022 t

tickers <- c("AAPL","HNHPF", "TSM", "4938.TW", "2324.TW")
tickers_small <- c("AAPL", "HNHPF")

# Start from Jan 1, 2022
start_date <- "2022-01-01"
end_date   <- Sys.Date() # Today

#Get the daily stock price data from Yahoo Finance.
stock_data <- tq_get(tickers,
                     get  = "stock.prices",
                     from = start_date,
                     to   = end_date)

stock_data_small <- tq_get(tickers_small,

                     get  = "stock.prices",
                     from = start_date,
                     to   = end_date)


"I normalized the data b/c APPL stock price is over 100 and HNHPF is less than 10,
so we wont be able to see the couple cent changes in foxconn and comparison to the dollar chang
In other words HNHPF will look like a flat line even when APPL in visually moving up and down."
```

```
normalized_data <- stock_data %>%
  select(symbol, date, adjusted) %>%
  group_by(symbol) %>%
  arrange(date) %>%
  mutate(
    percent_change = ((adjusted / first(adjusted) - 1)*100)
         )

normalized_data_small <- stock_data_small %>%
  select(symbol, date, adjusted) %>%
  group_by(symbol) %>%
  arrange(date) %>%
  mutate(
    percent_change = ((adjusted / first(adjusted) - 1)*100)
  )


# Fetch Data (if you haven't already)
apple_data <- tq_get("AAPL", get = "stock.prices", from = "2023-01-01")

# Rename columns to be capitalized for a professional look.
table_sample <- apple_data %>%
  tail(5) %>%                                      # Select only the last 5 rows
  select(date, open, high, low, close, volume) %>% # Keep only essential columns
  mutate(date = format(date, "%B %d, %Y")) %>%     # Format date nicely (e.g., "Jan 01, 2024")
  rename(                                          # Rename columns for display
    "Date"   = date,
    "Open"   = open,
    "High"   = high,
    "Low"    = low,
    "Close"  = close,
    "Volume" = volume
  )

# 'scale_down' automatically shrinks the table if it's too wide for the PDF margin.
# 'hold_position' forces the table to appear exactly where you put the code, not floating away
kbl(table_sample, caption = "Table 1: Recent Apple (AAPL) Stock Performance", booktabs = T) %>
  kable_styling(latex_options = c("striped", "scale_down", "hold_position")) %>%
  column_spec(1, bold = TRUE) # Make the Date column bold

normalized_data %>%
kable(
  caption = "<center>Supplier Performance affects on Apple stock<center/>",
  align = c("l", rep("c",7))
)%>%
  kable_classic( # Pre-built styling
    font_size = 16, # Control Font Size
    lightable_options = "striped" # Adds striping in a pre-built table style
  )

# Annual Return Table
annual_return_table <- annual_summary %>%
  mutate(
    total_return_pct = scales::percent(total_return, accuracy = 0.1)
  ) %>%
  select(symbol, year, total_return_pct) %>%  # Keep only the formatted column
  arrange(year, symbol)

# Create a styled table
kbl(
  annual_return_table,
  caption = "Annual Total Returns for Apple and Suppliers",
  col.names = c("Symbol", "Year", "Total Return (%)"),
  align = c("c","c","c"),
) %>%
  kable_styling() %>%
  column_spec(1, bold = TRUE)   # Bolds the ticker column


#| fig-alt: "A line chart displaying the normalized stock performance of Apple, Foxconn, TSMC,

# Line chart
ggplot(normalized_data, aes(x = date, y = percent_change, color = symbol)) +
  geom_line(linewidth = .5) +
  labs(title = "Normalized Stock Performance",
       x = "Date",
       y = "Percent Change",
       color = "Stock") +
  theme_minimal()

#| fig-alt: "A line chart comparing only Apple and Foxconn stock performance from 2022 to 2025

ggplot(normalized_data_small, aes(x = date, y = percent_change, color = symbol)) +
  geom_line(linewidth = .5) +
  labs(title = "Normalized Stock Performance",
       x = "Date",
       y = "Percent Change",
       color = "Stock") +
  theme_minimal()


# Prepare Cumulative Growth chart
yearly_growth <- stock_data %>% # Using the stock data from above.
  mutate(year = year(date)) %>% # mUtates the data, new column called year
```

```r
  group_by(symbol, year) %>%
  mutate(
    # Reset growth to 0% at start of every year
    pct_change = (adjusted / first(adjusted)) - 1
  ) %>%
  ungroup()

# B. Total Annual Return for the Bar Chart
annual_summary <- stock_data %>%
  mutate(year = year(date)) %>%
  group_by(symbol, year) %>%
  summarize(
    total_return = (last(adjusted) - first(adjusted)) / first(adjusted),
    .groups = "drop"
  )


#| fig-alt: "Four separate line plots, one for each year from 2022 to 2025, showing the cumula

# Plot-1: Line Chart
line_plot <- ggplot(yearly_growth, aes(x = date, y = pct_change, color = symbol)) +
  geom_line(linewidth = 0.8) +
  facet_wrap(~ year, scales = "free", ncol = 2) +
  scale_y_continuous(labels = percent_format()) +
  theme_tq() +
  scale_color_tq() +
  labs(
    title = "Apple vs. Suppliers: Yearly Growth",
    subtitle = "Performance reset to 0% at the start of each year",
    y = "Cumulative Return",
    x = "",
    color = "Ticker"
  ) +
  theme(legend.position = "bottom")

print(line_plot) # This will result in the line plot for every year.

#| fig-alt: "A set of bar charts representing total annual stock return percentages for each c

# Plot-2: Bar Chart
bar_plot <- ggplot(annual_summary, aes(x = symbol, y = total_return, fill = symbol)) +
  geom_col(color = "black", alpha = 0.8) +
  facet_wrap(~ year, ncol = 2) +
  geom_text(aes(label = percent(total_return, accuracy = 1)),
            vjust = -0.5, size = 3, fontface = "bold") +
  scale_y_continuous(labels = percent_format(), expand = expansion(mult = c(0.1, 0.2))) +
  scale_fill_tq() +
  theme_tq() +
  labs(
    title = "Total Return for Each Year",
    y = "Total % Change",
    x = ""
  ) +
  theme(legend.position = "none")

print(bar_plot) # This will result in the bar plot for every year.

#Linear Regression Analysis

#| fig-alt: "A forest plot showing linear regression coefficients (Betas) with 95% confidence

# Calculate Log Returns for Regression

returns_wide <- stock_data %>%
  group_by(symbol) %>%
  tq_transmute(select     = adjusted,
               mutate_fun = periodReturn,
               period     = "daily",
               type       = "log",
               col_rename = "returns") %>%
  ungroup() %>%
  pivot_wider(names_from = symbol, values_from = returns) %>%
  na.omit()

# List of suppliers to regress against AAPL
suppliers <- c("HNHPF", "TSM", "4938.TW", "2324.TW")

# Function to run individual simple linear regressions
run_model <- function(supp_ticker) {
  # Create formula: AAPL ~ Supplier
  # Use backticks for tickers that might have numbers/dots
  f <- as.formula(paste("AAPL ~ `", supp_ticker, "`", sep = ""))

  lm(f, data = returns_wide) %>%
    tidy(conf.int = TRUE) %>%
    filter(term != "(Intercept)") %>%
    mutate(term = supp_ticker) # Rename the term to the clean ticker name
}

# Run the model for each supplier and combine results into one table
model_results <- map_dfr(suppliers, run_model)

# Create the regression Plot
reg_plot <- ggplot(model_results, aes(x = term, y = estimate, color = term)) +
  geom_point(size = 4) +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high), width = 0.2, linewidth = 1) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  coord_flip() +
  theme_tq() +
  scale_color_tq() +
  labs(
    title = "Individual Effect of Supplier Returns on Apple",
    subtitle = "Simple Linear Regression Coefficients (95% CI)",
    y = "Effect on Apple (Beta)",
    x = "Supplier"
  ) +
  theme(legend.position = "none")

print(reg_plot)
```