Assignment 3B LLM Transcript

Muhammad Suffyan Khan

2026-02-15

LLM Transcript and Citation

Tool Used: ChatGPT (OpenAI, GPT-5.2)
Purpose: Designing a reproducible workflow for retrieving stock time-series data and implementing window-function calculations (YTD average and 6-day moving average) using `dplyr` and `slider` in R
Course: DATA 607 – Assignment 3B (Window Functions)
Date Accessed: February 15, 2026

Citation:
OpenAI. (2026). *ChatGPT (GPT-5.2)* [Large language model]. https://chat.openai.com.

I used ChatGPT to help me through this assignment. Specifically, I used it to (1) decide whether to use live data or a saved dataset for reproducibility, (2) retrieve stock price time-series data and save it to a CSV file, (3) implement window-function calculations in R using cummean() for the year-to-date average and slider::slide_dbl() for the six-day moving average, (4) add extended visualizations to interpret the results, and (5) handle missing values created by rolling-window calculations during plotting.

---

[1] Conversation with ChatGPT (Dataset choice + reproducibility strategy)

Is it good to fetch live stock data or use a saved dataset and then work?

Assistant (ChatGPT):
For reproducibility, it's best to fetch the data once and then save it (e.g., a CSV in the repo) so results don't change over time and the instructor can run the analysis without relying on live retrieval. A recommended workflow is:

fetch → save CSV → load CSV for analysis.

---

[2] Conversation with ChatGPT (Retrieving stock data and saving CSV)

How do I fetch live stock data from Yahoo Finance and save it as a CSV?

Assistant (ChatGPT):
Use `tidyquant::tq_get()` to retrieve daily stock prices for multiple tickers since 2022-01-01, keep the required columns (`symbol`, `date`, `close`), and then save the data to a local CSV using `readr::write_csv()`.

Suggested code:

```
library(tidyquant)
library(tidyverse)

tickers <- c("AAPL", "MSFT")
start_date <- "2022-01-01"

stock_data <- tq_get(
  tickers,
  from = start_date,
  get  = "stock.prices"
) %>%
  select(symbol, date, close)

write_csv(stock_data, "stock_data.csv")
head(stock_data)
```

[3] Conversation with ChatGPT (Window calculations: YTD and moving average)

How should I calculate the six-day moving average since the stock market is closed on weekends? Should I count calendar days or trading days?

Assistant (ChatGPT):
For stock price data, moving averages should be calculated using **trading days**, not calendar days. Since the dataset only contains trading days, you can compute the rolling average directly over the previous observations in the dataset.

For the year-to-date (YTD) calculation, you can compute a cumulative average within each year using `cummean()` grouped by `symbol` and `year`.

For the moving average, you can use `slider::slide_dbl()` with `.before = 5`.
This means the calculation includes the **current day plus the previous five trading days**, creating a **6-day moving average window**.

Example:

```
stock_features <- stock_data %>%
  mutate(year = lubridate::year(date)) %>%
  arrange(symbol, date)

stock_features <- stock_features %>%
  group_by(symbol, year) %>%
  mutate(ytd_avg_close = cummean(close)) %>%
  ungroup()

stock_features <- stock_features %>%
  group_by(symbol) %>%
  arrange(date, .by_group = TRUE) %>%
  mutate(
    ma6_close = slider::slide_dbl(
      close,
      mean,
      .before = 5,
      .complete = TRUE
    )
  ) %>%
  ungroup()
```

[4] Conversation with ChatGPT (Extended visualization ideas)

How can I add an extension to visualize results?

Assistant (ChatGPT):
You can visualize close price vs. moving averages for (1) a single stock and (2) multiple stocks using facets.

Suggested code:

```
library(ggplot2)
library(scales)

symbol_to_plot <- "AAPL"
```

```r
stock_features %>%
  filter(symbol == symbol_to_plot) %>%
  ggplot(aes(x = date)) +
  geom_line(aes(y = close)) +
  geom_line(aes(y = ytd_avg_close)) +
  geom_line(aes(y = ma6_close), na.rm = TRUE) +
  scale_y_continuous(labels = dollar_format()) +
  labs(
    title = paste("Close vs YTD Avg vs 6-Day Moving Avg:", symbol_to_plot),
    x = "Date",
    y = "Price (USD)"
  )

stock_features %>%
  ggplot(aes(x = date)) +
  geom_line(aes(y = close)) +
  geom_line(aes(y = ma6_close), na.rm = TRUE) +
  facet_wrap(~ symbol, scales = "free_y") +
  labs(
    title = "Close vs 6-Day Moving Average (Multiple Stocks)",
    x = "Date",
    y = "Price"
  )
```

[5] Conversation with ChatGPT (Handling missing values in moving averages for plots)

I got a warning removed rows containing missing values for the moving average.

Assistant (ChatGPT):
This happens because rolling windows create NA values at the start of each series (for example, the first 5 rows when computing a 6-day moving average with .complete = TRUE). The calculations are correct.

Suggested code:

```r
geom_line(aes(y = ma6_close), na.rm = TRUE)
```

ChatGPT is AI and can make mistakes.