Assignment 3

Ivan Hung [CID - 06039211]

2025-11-20

Table of contents

Load R Packages 3

Cleaning dataset 3

ACF and PACF Plot Analysis 5
- ACF and PACF - Different Time Horizons . . . 7

Moving Average Crossover Strategy 11
- Performance Summary of Moving Average Strategy . . . 13

Bollinger Bands Trading Strategy (Created with GenAI) 17
- Bollinger Band Trading Strategy Evaluation . . . 20

Analysis of Findings 22

List of Figures

1 Equity Curve using MA Crossover Strategy . . . 13
2 Optiaml MA Crossover Performance Summary . . . 14
3 Pfizer Price with 50 and 120 Day MAs . . . 16
4 Moving Average Daily Returns Plot . . . 17
5 Pfizer Price with Bollinger Bands . . . 20
6 Bollinger Bands Daily Returns Plot . . . 21
7 Bollinger Bands Performance Summary . . . 22

List of Tables

1 Cleaned Dataset . . . 4
2 Simple Returns Table . . . 5
3 Optimal fast and slow MA lags . . . 13

Load R Packages

```r
install.packages(c("dplyr", "dtplyr", "data.table", "lubridate",
"ggplot2", "PerformanceAnalytics", "xts"))
```

```
The downloaded binary packages are in
    /var/folders/by/g895l71128j9y19qqn66xcxh0000gn/T//RtmpW7c0PG/downloaded_packages
```

```r
library(dplyr)
library(dtplyr)
library(data.table)
library(lubridate)
library(ggplot2)
library(PerformanceAnalytics)
library(xts)

options(repr.plot.width = 10, repr.plot.height = 4)  # For Jupyter notebooks
```

```r
df <- fread("/Users/ivanhung/Documents/GitHub/final-r-assignment/dataset.csv")
```

Cleaning dataset

```r
# Setting datadate to a date object
df %>% mutate(datadate = as.Date(datadate, format = "%m/%d/%Y")) -> df

# Remove irrelevant columns and other stocks except for Pfizer (PFE)
pfe <- df[tic == "PFE",
          .(datadate, cshtrd, prccd, prchd, prcld, prcod)]

head(pfe)

# Plot time series of PFE's closing prices
ggplot(data = pfe, aes(x = datadate, y = prccd)) +
  geom_line() +
  labs(title = "Pfizer (PFE) Closing Stock Prices Over Time",
       x = "Date",
```

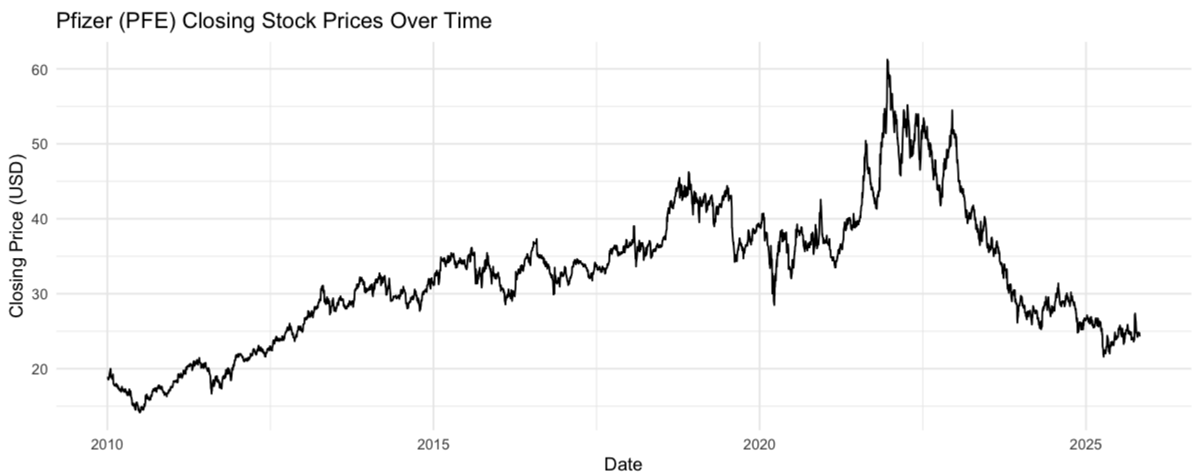
```
      y = "Closing Price (USD)") +
  theme_minimal()
```

A data.table: 6 × 6

Table 1: Cleaned Dataset

datadate <date>	cshtrd <dbl>	prccd <dbl>	prchd <dbl>	prcld <dbl>	prcod <dbl>
2010-01-04	52074710	18.93	18.94	18.235	18.27
2010-01-05	43368460	18.66	18.93	18.550	18.92
2010-01-06	41405070	18.60	18.81	18.510	18.66
2010-01-07	39427720	18.53	18.67	18.460	18.64
2010-01-08	30403370	18.68	18.71	18.520	18.62
2010-01-11	32442710	18.83	18.95	18.670	18.83

Pfizer Closing Stock Prices Over Time



Next, calculate simple returns so that we can make our prices stationary and allow us to have a better understanding our data as we can proceed to plot Pfizer's ACF and PACF plots and confirm for certain statistical characteristics.

```
# Calculating simple returns (simple returns in %),
# which we will denote as s_ret (s_ret_per)

pfe <- pfe %>%
    mutate(ret = round(((prccd/lag(prccd))-1), 4)) %>%
```

```
    mutate(s_ret = round(((prccd/lag(prccd))-1)*100, 4))

head(pfe)
```

A data.table: 6 × 8

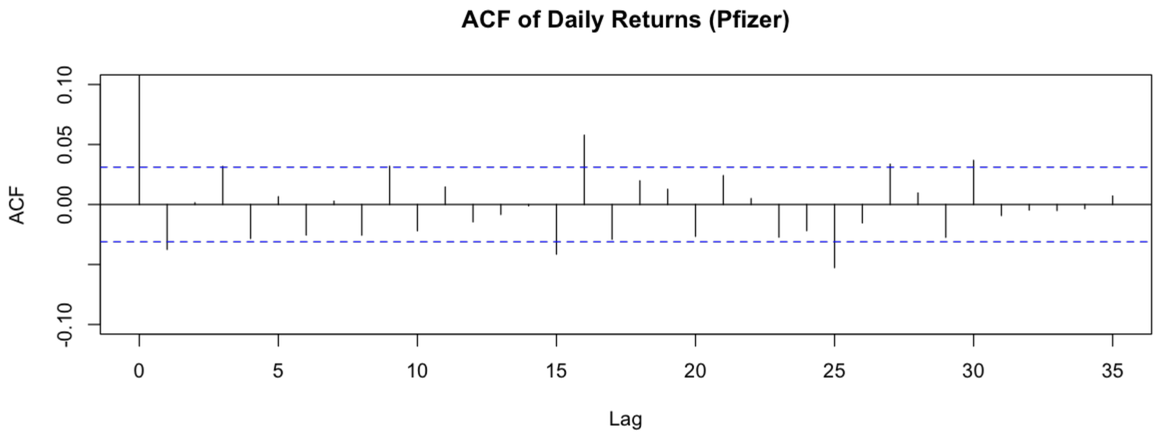
Table 2: Simple Returns Table

datadate <date>	cshtrd <dbl>	prccd <dbl>	prchd <dbl>	prcld <dbl>	prcod <dbl>	ret <dbl>	s_ret <dbl>
2010-01-04	52074710	18.93	18.94	18.235	18.27	NA	NA
2010-01-05	43368460	18.66	18.93	18.550	18.92	-0.0143	-1.4263
2010-01-06	41405070	18.60	18.81	18.510	18.66	-0.0032	-0.3215
2010-01-07	39427720	18.53	18.67	18.460	18.64	-0.0038	-0.3763
2010-01-08	30403370	18.68	18.71	18.520	18.62	0.0081	0.8095
2010-01-11	32442710	18.83	18.95	18.670	18.83	0.0080	0.8030

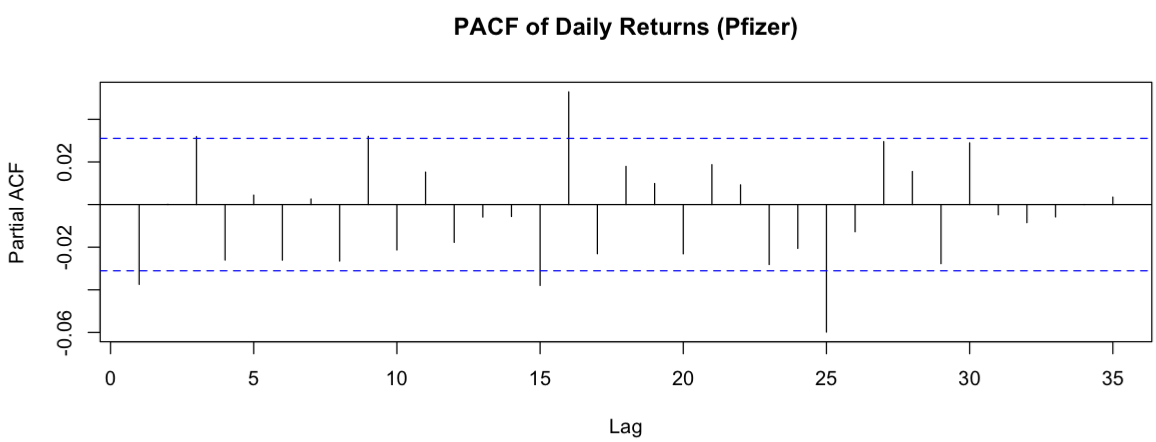
Next, we should plot the autocorrelation and partial-autocorrelation functions of our closing prices to identify if there are any seasonal structures or autocorrelation that we might need to deal with.

ACF and PACF Plot Analysis

```
# ACF plot to identify any autocorrelation or seasonality patterns in our data
#| fig-cap: "ACF of Daily Returns Pfizer"
#| warning: false
#| results: hide

acf(pfe$ret, lags = 20,
    na.action = na.omit,
    main = "ACF of Daily Returns (Pfizer)",
    ylim = c(-0.1,0.1))
```




```
# PACF plot
#| fig-cap: "PACF of Daily Returns Pfizer"
#| warning: false
#| results: hide

pacf(pfe$ret, lags = 20,
    na.action = na.omit,
    main = "PACF of Daily Returns (Pfizer)")
```



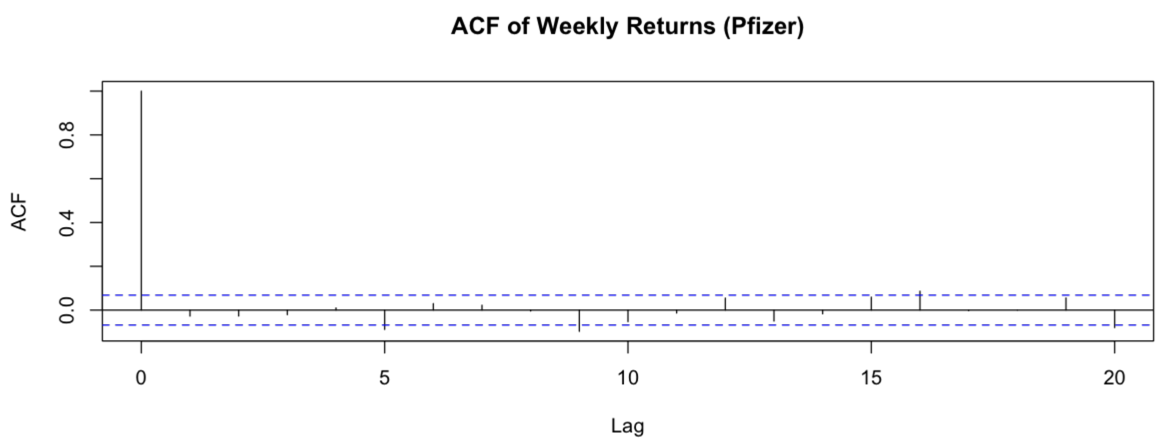
We can focus in on a specific time horizon and see if autocorrelation exists within a certain timeframe. This evidence can help us to determine whether there is statistical arbitrage in which our trading strategy can detect a pattern can profit from (potentially short term momentum), as opposed to having white noise (returns which follow a strong form of the EMH where all information about the stock is reflected in its prices). We decide to focus on a relatively long horizon as it can provide us more information of how the stock's price changed before, during, and after COVID; providing us with a holistic story and finding opportunities for statistical arbitrage in the '3' phases of COVID.

ACF and PACF - Different Time Horizons

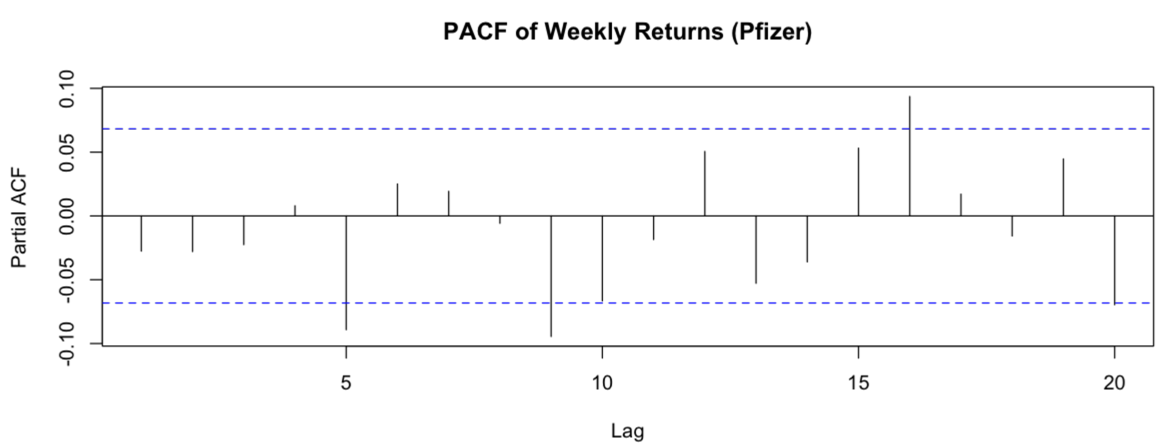
```
# Create a list of periods we want to look at
periods <- list(
    weekly = "week",
    monthly = "month",
    semi_annually = "6 months",
    annually = "year"
)

# Loop through each time period
for (period_name in names(periods)) {
    cat("Processing:", toupper(period_name), "\n")

    # Create proper period returns (one observation per period)
    pfe_period <- pfe %>%
        mutate(period = floor_date(datadate, periods[[period_name]])) %>%
        group_by(period) %>%
        arrange(datadate) %>%
        slice_tail(n = 1) %>%
        ungroup() %>%
        arrange(datadate) %>%
        mutate(period_ret = (prccd / lag(prccd)) - 1) %>%
        filter(!is.na(period_ret))

    # View tibble (dataframe) using the following:
    # print(head(pfe_period))
    # cat("Number of periods:", nrow(pfe_period), "\n\n")

    # ACF of actual period returns (suppress warnings)
    suppressWarnings(
        acf(pfe_period$period_ret, lag.max = 20,
            na.action = na.omit,
            main = paste("ACF of",
            tools::toTitleCase(gsub("_", " ", period_name)),
            "Returns (Pfizer)"))
    )
```

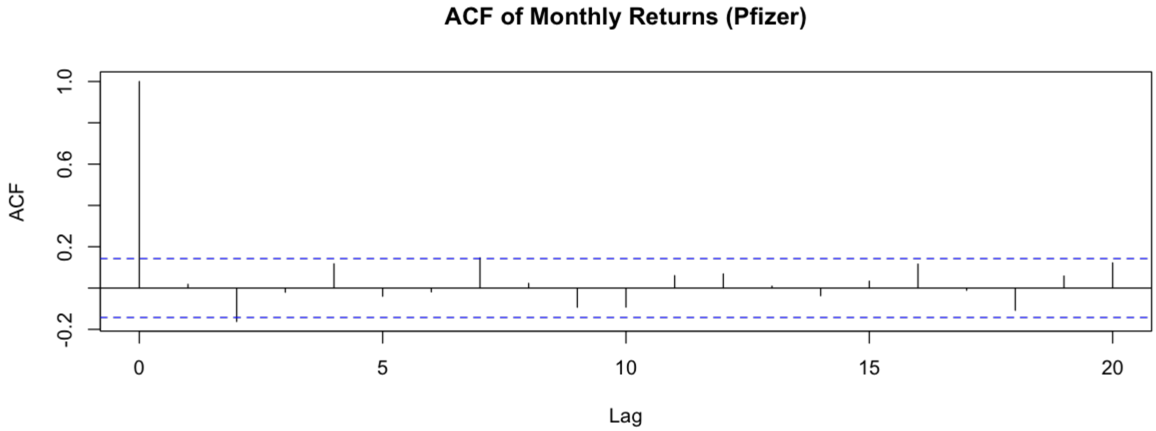
```
    # PACF of actual period returns (suppress warnings)
    suppressWarnings(
        pacf(pfe_period$period_ret, lag.max = 20,
             na.action = na.omit,
             main = paste("PACF of",
             tools::toTitleCase(gsub("_", " ", period_name)),
             "Returns (Pfizer)"))
    )
}
```

```
Processing: WEEKLY
```

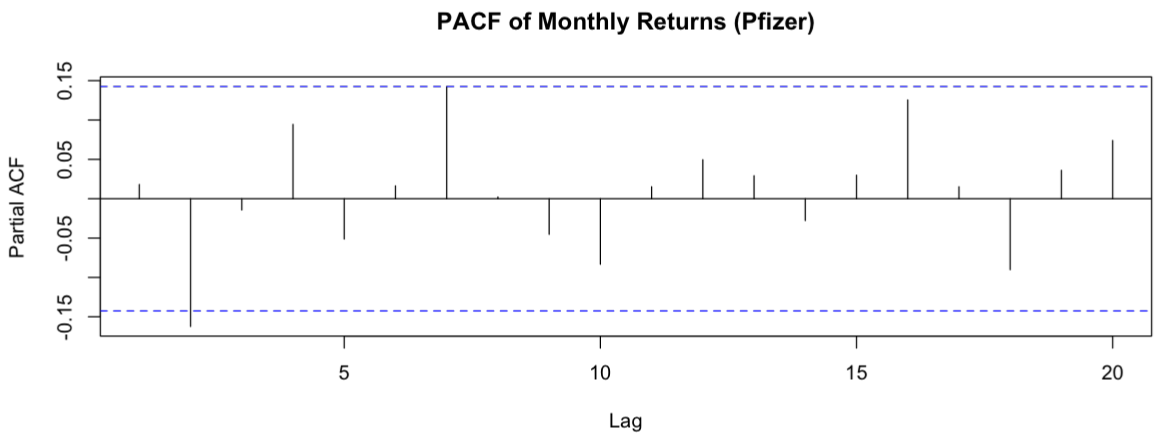


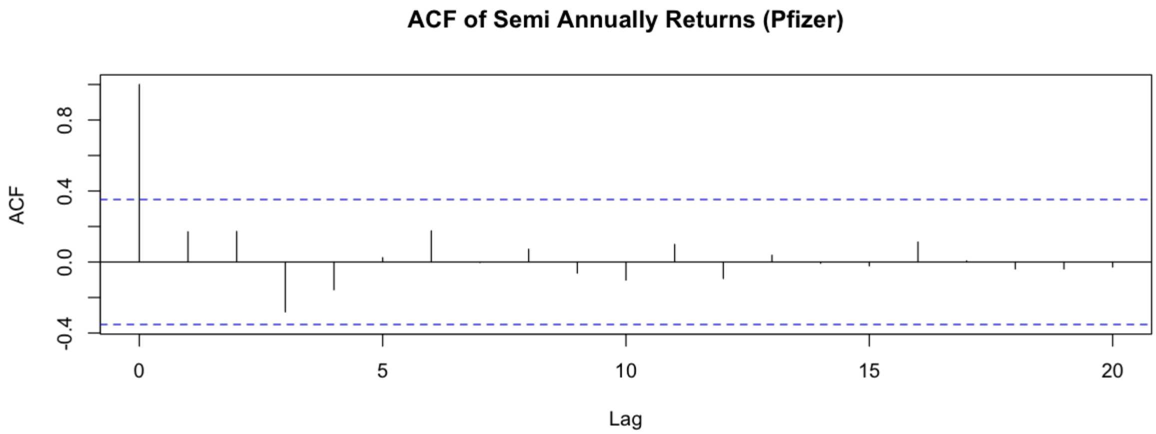

```
Processing: MONTHLY
```



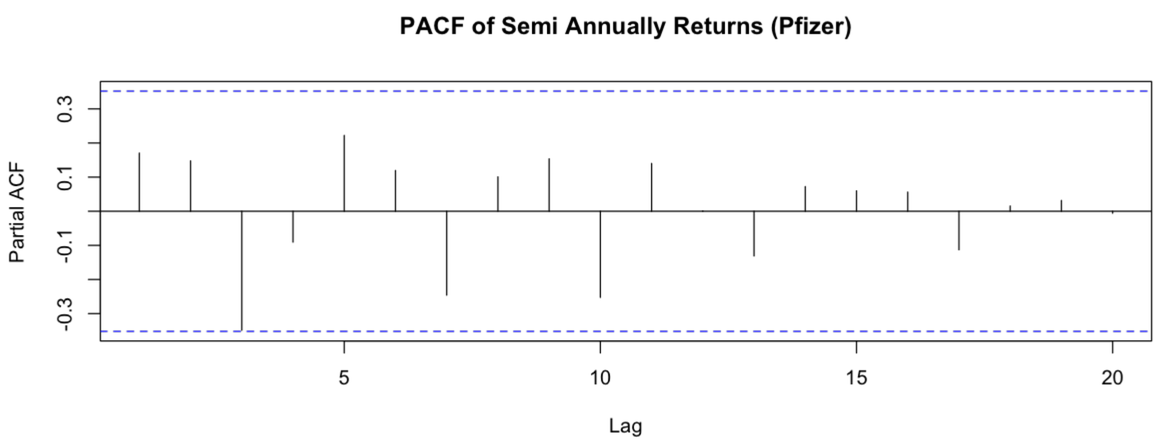


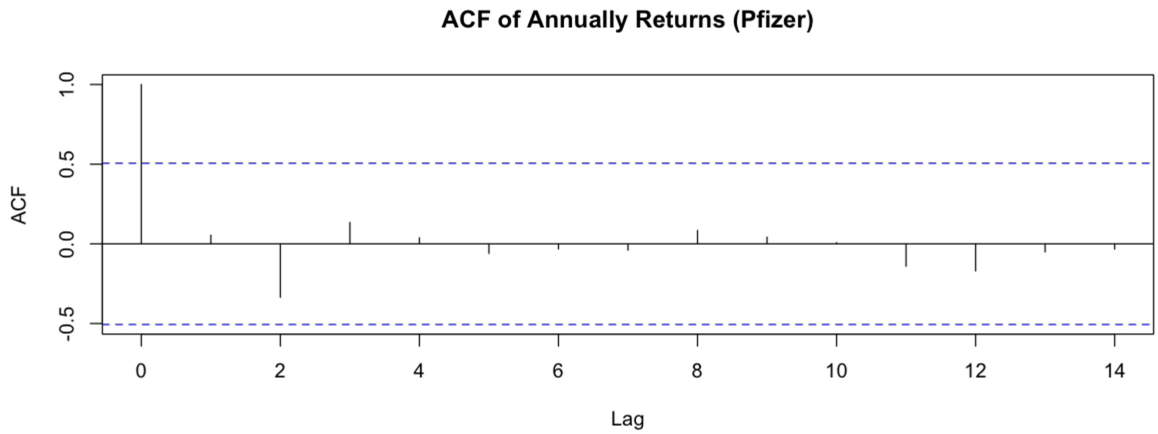

```
Processing: SEMI_ANNUALLY
```

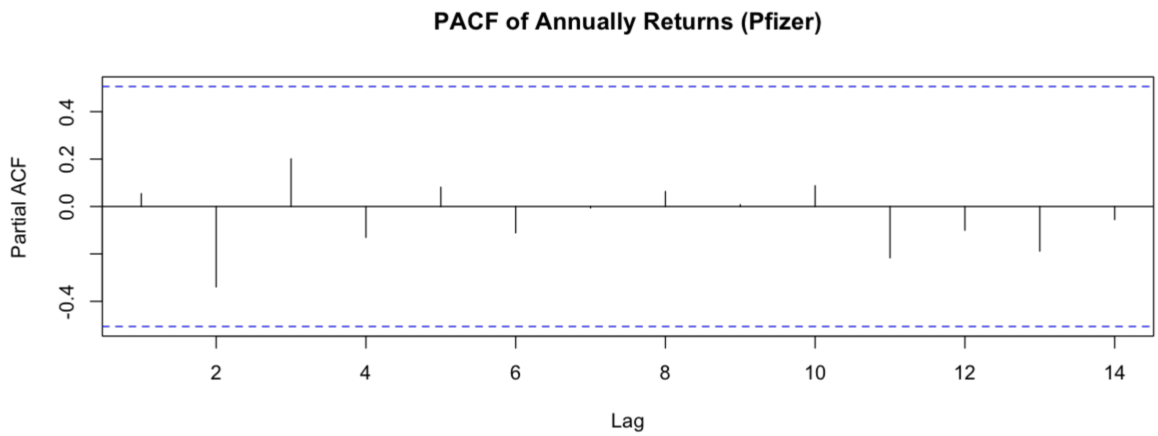




Processing: ANNUALLY







Moving Average Crossover Strategy

```
pfe <- pfe %>%
  filter(datadate >= as.Date("2020-01-01")) %>%
  arrange(datadate)
```

```
source('bs-signal.r')
source('backtest.r')

# Grids to search
fast_grid <- c(20, 30, 40, 50)
```

```r
slow_grid <- c(60, 90, 120, 150)

# Store data for each combination
results <- data.frame(
  fast = integer(),
  slow = integer(),
  final_equity = numeric(),
  stringsAsFactors = FALSE
)

for (f in fast_grid) {
  for (s in slow_grid) {
    if (f >= s) next

    # 1) Generate signals
    sig <- ma_signal(pfe$prccd, fast = f, slow = s)

    # 2) Backtest
    bt  <- backtest_ma(pfe$prccd, signal = sig)

    # 3) Store final equity (performance metric)
    final_eq <- tail(bt$equity, 1)

    results <- rbind(
      results,
      data.frame(fast = f, slow = s, final_equity = final_eq)
    )
  }
}

# Pick best-performing combination
best_idx   <- which.max(results$final_equity)
best_fast  <- results$fast[best_idx]
best_slow  <- results$slow[best_idx]
results[best_idx, ]
```

A data.frame: 1×3

Table 3: Optimal fast and slow MA lags

	fast <dbl>	slow <dbl>	final_equity <dbl>
15	50	120	1.178767

```r
price <- pfe$prccd

# Generate signals and backtest using the optimal windows
best_signal <- ma_signal(price, fast = best_fast, slow = best_slow)
best_bt     <- backtest_ma(price, signal = best_signal)

# Equity curve plot
plot(
  pfe$datadate, best_bt$equity, type = "l",
  xlab = "Date", ylab = "Equity (start = 1)",
  main = paste("Equity Curve - MA Crossover (fast =", best_fast,
               ", slow =", best_slow, ")")
)
abline(h = 1, lty = 5)
```

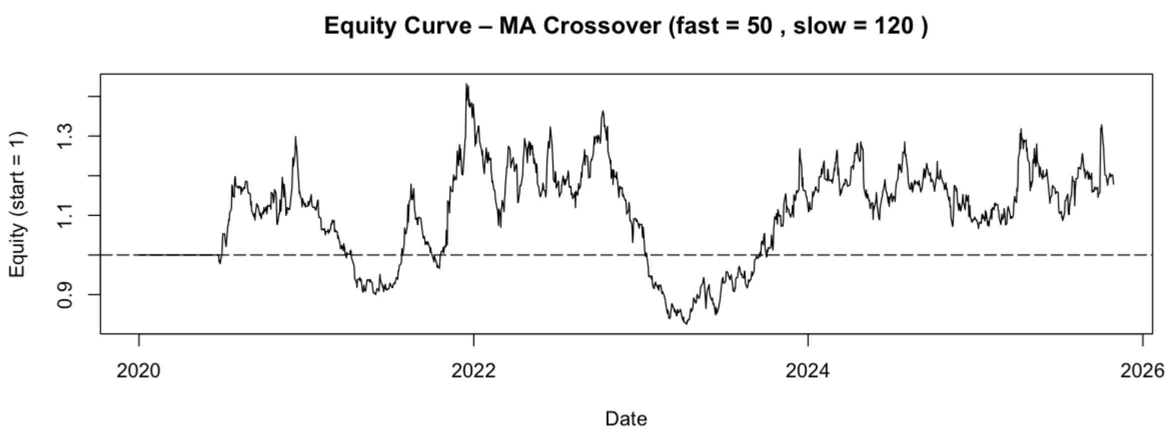


Figure 1: Equity Curve using MA Crossover Strategy

Performance Summary of Moving Average Strategy

```
# Store data in an xts readable dataframe
strategy_returns <- xts(best_bt$returns, order.by = pfe$datadate)
benchmark_returns <- xts(pfe$ret, order.by = pfe$datadate)

comparison <- merge(strategy_returns, benchmark_returns)
colnames(comparison) <- c("MA Crossover Strategy (50,120)",
                          "Buy & Hold Benchmark (PFE Returns)")

# Performance Summary Chart
options(repr.plot.width = 10, repr.plot.height = 5.5)
charts.PerformanceSummary(
    comparison,
    geometric = FALSE,
    main = paste("Strategy Performance (Fast MA =",
        best_fast, ", Slow MA =", best_slow, ")")
)

# Reset to default height
options(repr.plot.width = 10, repr.plot.height = 4)
```

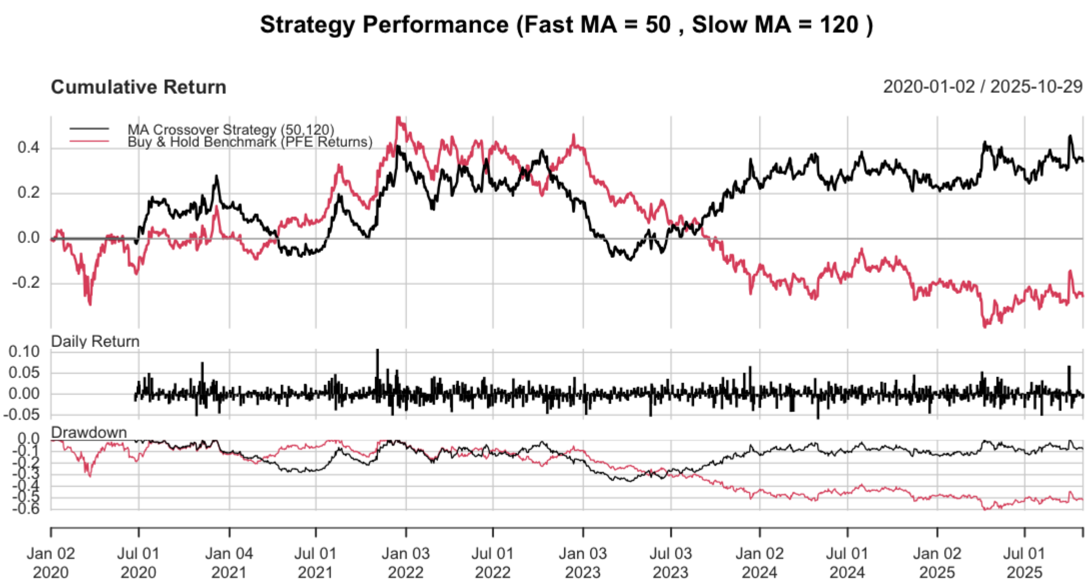


Figure 2: Optiaml MA Crossover Performance Summary

```r
# Compute moving averages for the best windows
fast_ma_best <- moving_avg(price, win_size = best_fast)
slow_ma_best <- moving_avg(price, win_size = best_slow)

# Base plot: price
plot(
  pfe$datadate, price, type = "l",
  xlab = "Date", ylab = "Price",
  main = paste("Pfizer Price with", best_fast,
  "and", best_slow, "Day MAs")
)

# Add moving averages
lines(pfe$datadate, fast_ma_best, col = "blue")
lines(pfe$datadate, slow_ma_best, col = "red")

# Identify crossover points
spread    <- fast_ma_best - slow_ma_best
cross_idx <- which(diff(sign(spread)) != 0) + 1

# Add crossover markers to the plot
points(
  pfe$datadate[cross_idx],
  price[cross_idx],
  pch = 16, col = "darkgreen", cex = 1.2
)

legend(
  "topleft",
  legend = c(
    "Price",
    paste0("Fast MA (", best_fast, ")"),
    paste0("Slow MA (", best_slow, ")"),
    "Crossover Point"
  ),
  col = c("black", "blue", "red", "darkgreen"),
  lty = c(1, 1, 1, NA),
  pch = c(NA, NA, NA, 16),
  bty = "n"
)
```

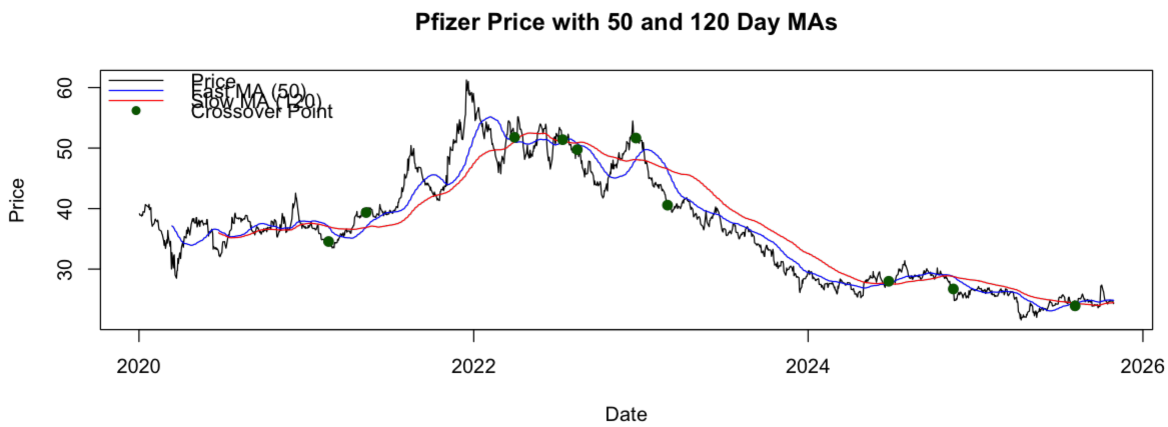


Figure 3: Pfizer Price with 50 and 120 Day MAs

```
# Correct level for ±0.05% daily return
threshold <- 0.0005

plot(
  pfe$datadate, strategy_returns,
  type = "h",
  xlab = "Date",
  ylab = "Daily Return",
  main = "Daily Returns - Moving Average Strategy"
)

# Zero line
abline(h = 0, lty = 2, col = "gray")

# ±0.05% marker lines
abline(h = threshold,  lty = 2, col = "blue")
abline(h = -threshold, lty = 2, col = "blue")

legend(
  "topright",
  legend = c("Daily Returns", "0.05% Threshold"),
  col = c("black", "blue"),
  lty = c(1, 2),
  bty = "n"
)
```

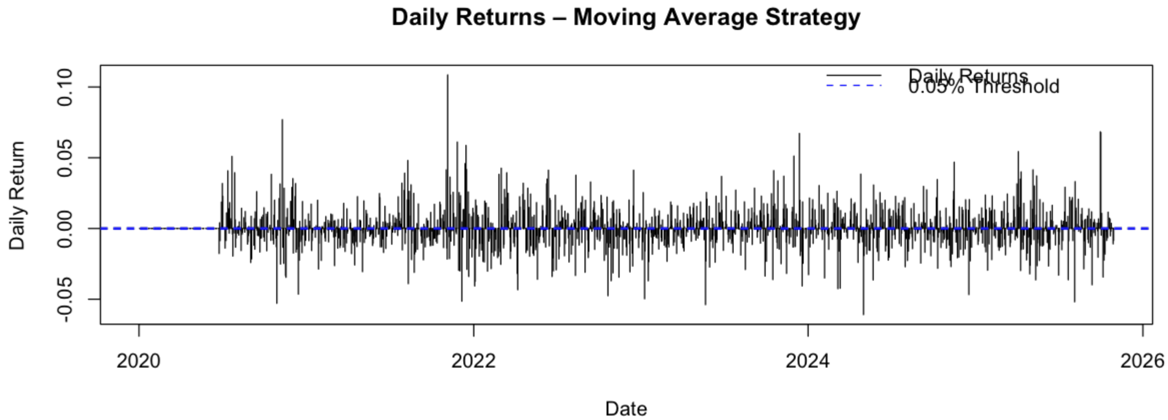


Figure 4: Moving Average Daily Returns Plot

Bollinger Bands Trading Strategy (Created with GenAI)

We will also document the performance of another strategy, the Bollinger Bands Strategy. We utilise generative-AI to develop a function which replicates the Bollinger Bands Strategy. The fundamental approach of this strategy is to plot the SMA, an upper and lower band (+/- 2 standard deviatinons away from SMA) and observe whether there is volatility and potentialy overbuying/overselling conditions that we can arbitrage from. We see that the function the LLM develops, is to indicate whether we should hold a long or short position, depending on the movement of prices. The function tells us that we should take a long position if prices go above the upper band and a short position if prices go below the lower band.

The prompt we provided to the LLM was:

Write clean, well-commented R code to implement a Bollinger Bands breakout trading strategy. Assume the dataset `pfe` contains `datadate` (Date) and `prccd` (closing price).

The script must:

1. Define a function to compute Bollinger Bands (middle, upper, lower) using parameters `n` and `k`.
2. Create a function to generate trading signals (+1 long, -1 short, 0 neutral) based on Bollinger breakout logic.
3. Include a backtesting function that computes daily strategy returns and an equity curve, using lagged signals to prevent lookahead bias.
4. Compare the strategy to a buy-and-hold benchmark and produce an interpretable performance summary including an equity curve plot.
5. Ensure code is readable, well-structured, and appropriate for use in a technical report.

```r
# Setting up Bollinger Bands
bollinger_bands <- function(prices, n = 20, k = 2) {
  N   <- length(prices)
  mid <- rep(NA_real_, N)
  sdv <- rep(NA_real_, N)

  if (n > N) {
    warning("Window size n is larger than length of price series.")
    return(list(
      middle = mid,
      upper  = rep(NA_real_, N),
      lower  = rep(NA_real_, N)
    ))
  }

  for (i in n:N) {
    window_vals <- prices[(i - n + 1):i]
    mid[i] <- mean(window_vals, na.rm = TRUE)
    sdv[i] <- stats::sd(window_vals, na.rm = TRUE)
  }

  upper <- mid + k * sdv
  lower <- mid - k * sdv

  return(list(
    middle = mid,
    upper  = upper,
    lower  = lower
  ))
}

# Bollinger Bands Indicator
bb_signal <- function(price, n = 20, k = 2) {
  bb <- bollinger_bands(price, n = n, k = k)

  upper <- bb$upper
  lower <- bb$lower

  signal <- rep(0, length(price))

  # breakout logic
  signal[price > upper] <-  1   # long when breaking above upper band
```

```r
  signal[price < lower] <- -1   # short when breaking below lower band

  return(signal)
}
```

```r
# Compute Bollinger Bands
bb    <- bollinger_bands(price, n = 20, k = 2)
mid   <- bb$middle
upper <- bb$upper
lower <- bb$lower

# Generate signals (breakout)
sig_bb <- bb_signal(price, n = 20, k = 2)

# Store Bollinger outputs in data frame
pfe$bb_mid    <- mid
pfe$bb_upper  <- upper
pfe$bb_lower  <- lower
pfe$bb_signal <- sig_bb
```

```r
plot(
  pfe$datadate, price, type = "l",
  xlab = "Date", ylab = "Price",
  main = "Pfizer Price with Bollinger Bands (n = 20, k = 2)"
)

# Add bands
lines(pfe$datadate, pfe$bb_mid,   col = "blue")
lines(pfe$datadate, pfe$bb_upper, col = "purple")
lines(pfe$datadate, pfe$bb_lower, col = "purple")

# Mark buy (break above upper) and sell (break below lower)
buy_idx  <- which(pfe$bb_signal ==  1)
sell_idx <- which(pfe$bb_signal == -1)

points(pfe$datadate[buy_idx],  price[buy_idx],  pch = 24, bg = "green")
points(pfe$datadate[sell_idx], price[sell_idx], pch = 25, bg = "red")

legend(
  "topleft",
  legend = c("Price", "Middle Band", "Upper/Lower Bands",
              "Buy Signal", "Sell Signal"),
```

```r
  col    = c("black", "blue", "purple", "green", "purple"),
  lty    = c(1, 1, 1, NA, NA),
  pch    = c(NA, NA, NA, 24, 25),
  pt.bg  = c(NA, NA, NA, "green", "red"),
  bty    = "n"
)
```

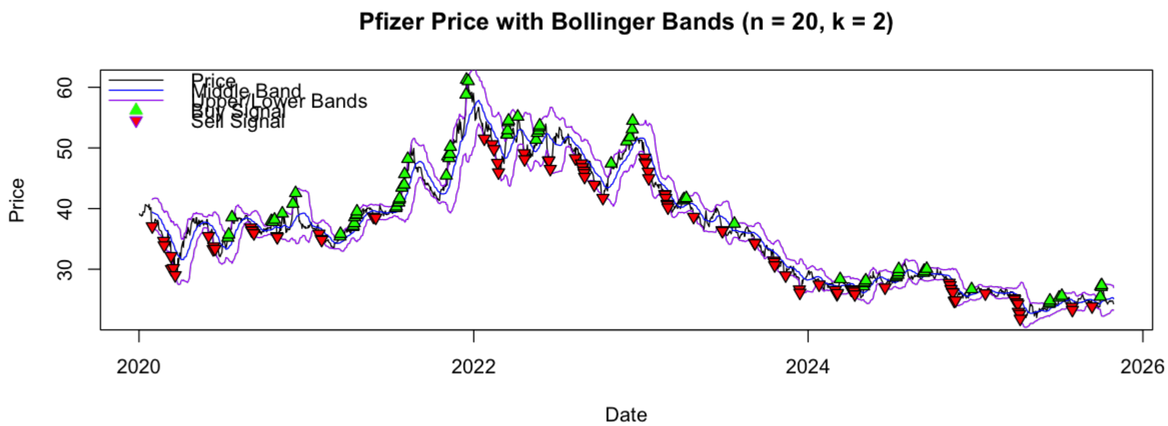


Figure 5: Pfizer Price with Bollinger Bands

Bollinger Band Trading Strategy Evaluation

```r
# Daily returns from the Bollinger Bands strategy
strategy_returns_bb <- bt_bb$returns

plot(
  pfe$datadate, strategy_returns_bb,
  type = "h",
  xlab = "Date",
  ylab = "Daily Return",
  main = "Daily Returns - Bollinger Bands Trading Strategy"
)

# Zero line
abline(h = 0, lty = 2, col = "gray")

# ±0.05% marker lines
```

```r
abline(h = threshold,  lty = 2, col = "blue")
abline(h = -threshold, lty = 2, col = "blue")

legend(
  "topright",
  legend = c("Daily Returns", "0.05% Threshold"),
  col = c("black", "blue"),
  lty = c(1, 2),
  bty = "n"
)
```

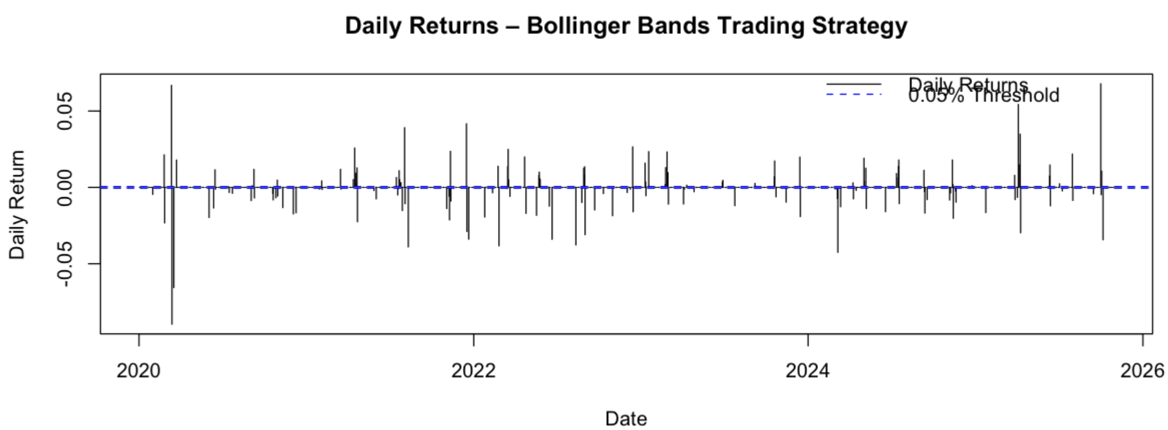


Figure 6: Bollinger Bands Daily Returns Plot

```r
bollinger_returns <- xts(
  bt_bb$returns,
  order.by = pfe$datadate   # or pfe_2020$datadate if you're using the subset
)

comparison2 <- merge(bollinger_returns, benchmark_returns, all = FALSE)
colnames(comparison2) <- c("Bollinger Bands Strategy",
                           "Buy & Hold Benchmark (PFE Returns)")

# Performance Summary Chart
options(repr.plot.width = 10, repr.plot.height = 5.5)
charts.PerformanceSummary(
    comparison2,
    geometric = FALSE,
```

```
    main = paste("Strategy Performance (Fast MA =",
        best_fast, ", Slow MA =", best_slow, ")")
)
```

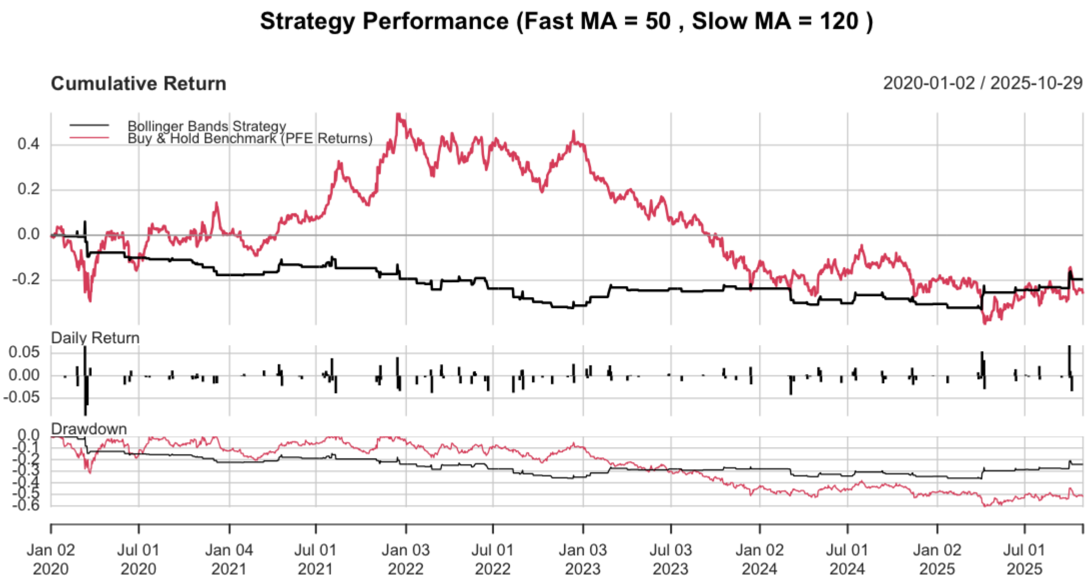


Figure 7: Bollinger Bands Performance Summary

Analysis of Findings

The autocorrelation analysis across multiple return frequencies (daily to annual) shows that nearly all ACF and PACF coefficients fall within statistical confidence bands. No lag exceeds the 0.2 threshold while remaining statistically significant, indicating there is no meaningful autocorrelation structure in Pfizer's returns over the examined period.

This behaviour is consistent with white-noise return dynamics, where past values fail to predict future movements. While this reduces overfitting risk in forecasting models, it also means there is little evidence of momentum, mean reversion, or seasonality that could support autocorrelation-driven trading signals. As such, autocorrelation alone provides limited insight for systematic timing or arbitrage in this asset.

Despite this, the moving average crossover strategy demonstrates stronger performance than a buy-and-hold benchmark. As shown in Figure 2, outperformance is most notable during downward market phases where the crossover rules actively reduce or reverse exposure. The drawdown panel further highlights risk control benefits, with shallower and faster-recovering equity declines. This aligns with the logic of moving average rules: when short-term price momentum deteriorates, positions reduce or flip, limiting downside participation.

Figure 3 shows relatively few crossover execution points, suggesting the strategy trades infrequently and targets structural trends rather than short-term noise. However, periods of flat or highly erratic price movement reduce effectiveness, as false signals can emerge before trends establish.

The Bollinger Bands strategy presents a more conservative trading profile. From Figure 7, the daily returns appear intermittent rather than continuous, reflecting conditional triggers rather than constant positioning. Though cumulative returns are lower than the MA crossover system, the approach offers smoother drawdowns and lower volatility than buy-and-hold.

Overall, both systematic strategies outperform passive holding in terms of drawdown management, but the moving average crossover achieves stronger return enhancement. The weaker performance of Bollinger Bands likely stems from Pfizer's trending rather than mean-reverting price behaviour; Bollinger systems typically perform best in more stationary environments where prices oscillate around a stable equilibrium.