AMProject_Clean

Laura Gullicksen, Erich Gozebina, Daria Palitzsch

23/05/2025

1. Introduction

#TODO: describe data choice

2. Data & Descriptive Analysis

#TODO: Explain the daily aggregation -> 7-day-trading-strategy -> daily makes more sense

Using log returns instead of simple (arithmetic) returns is a standard practice in financial econometrics and modeling.

- Log returns are more symmetrically distributed and are better approximated by a normal distribution, especially for small time intervals (e.g., hourly/daily). This makes them more suitable for:
  - Linear models
  - Hypothesis testing
  - Machine learning regressors
- Because log returns are additive, they allow you to aggregate returns over multiple periods simply by summing simple return becomes undefined. Log return avoids this issue as long as prices are strictly positive, which is true for most financial assets (especially crypto).

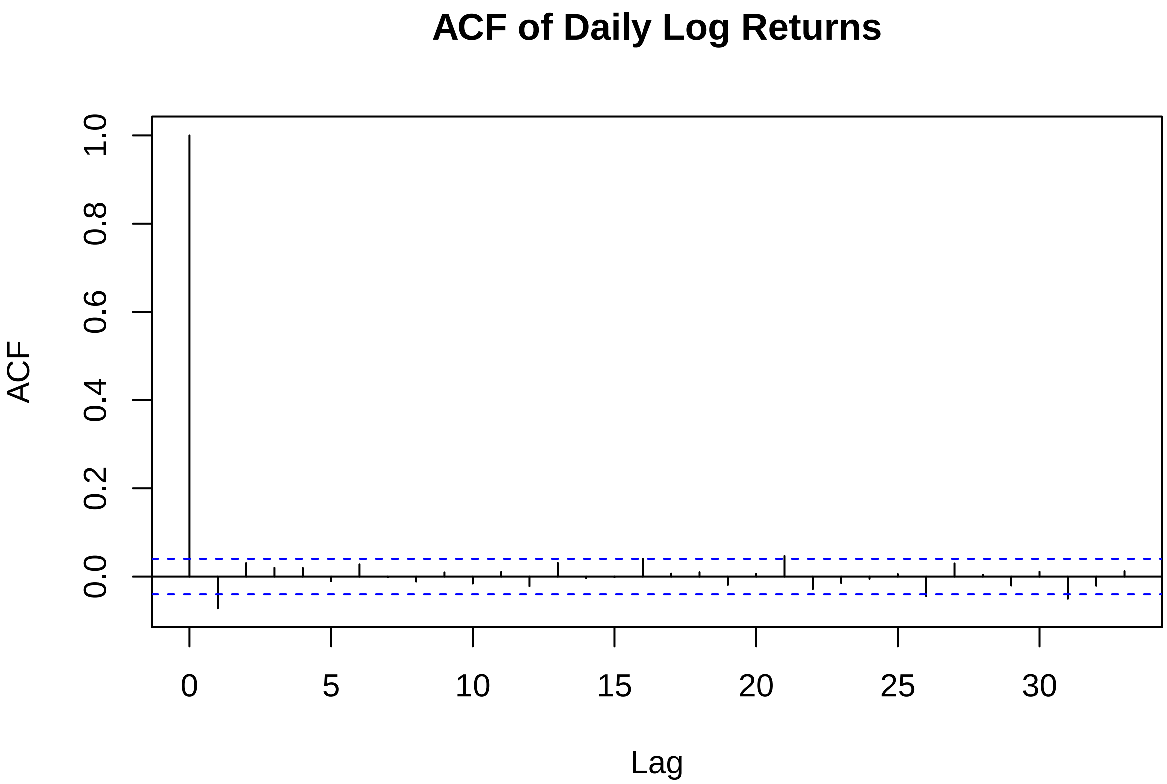
```r
# Descriptive stats for prices and returns
summary_stats <- df_daily %>%
  summarise(
    n_obs = n(),
    mean_close = mean(close_price, na.rm = TRUE),
    sd_close = sd(close_price, na.rm = TRUE),
    min_close = min(close_price, na.rm = TRUE),
    max_close = max(close_price, na.rm = TRUE),
    mean_return = mean(log_return, na.rm = TRUE),
    sd_return = sd(log_return, na.rm = TRUE),
    min_return = min(log_return, na.rm = TRUE),
    max_return = max(log_return, na.rm = TRUE)
  )
```

```r
summary_stats_long <- as.data.frame(t(summary_stats))
colnames(summary_stats_long) <- "Value"

# Add a column for metric names
summary_stats_long <- tibble::rownames_to_column(summary_stats_long, var = "Statistic")

# Show result
summary_stats_long
```

```
##     Statistic         Value
## 1       n_obs  2.383000e+03
## 2  mean_close  9.148397e+00
## 3    sd_close  9.540702e+00
## 4   min_close  1.452550e-01
## 5   max_close  5.210000e+01
## 6 mean_return  1.594292e-03
## 7   sd_return  6.764975e-02
## 8  min_return -6.776430e-01
## 9  max_return  4.761717e-01
```

```r
#TODO: Show them side-by-side

# Plot closing price
# ggplot(prices_link, aes(x = date, y = close_price)) +
#   geom_line(color = "steelblue") +
#   labs(title = "Daily Close Price", x = "Date", y = "Price")
#
# # Plot returns
# ggplot(prices_link, aes(x = date, y = log_return)) +
#   geom_line(color = "darkred") +
#   labs(title = "Daily Log Returns", x = "Date", y = "Log Return")
```

```r
# ACF plot of returns
acf(na.omit(df_daily$log_return), main = "ACF of Daily Log Returns")
```



Interpretation: The autocorrelation function of daily log returns shows no statistically significant linear dependence, indicating that past returns do not linearly predict future returns. This supports the weak-form Efficient Market Hypothesis. However, this does not rule out the presence of exploitable patterns through non-linear or directional indicators. Therefore, we adopt a momentum-based strategy, using the sign of past multi-day returns to generate long or short trading signals.

3. Standard Model

#Momentum Signal Strategy

We define the 7-day momentum as the log return over the past 7 days:

$$\text{Momentum}_t = \log\left(\frac{P_t}{P_{t-7}}\right)$$

The trading signal is then determined as:

$$\text{Signal}_t = \begin{cases} +1 & \text{if Momentum}_t > 0 \quad \text{(go long)} \\ -1 & \text{if Momentum}_t < 0 \quad \text{(go short)} \\ 0 & \text{otherwise (no position)} \end{cases}$$

The strategy return is computed as:

$$r_{t+1}^{\text{strategy}} = \text{Signal}_t \cdot r_{t+1}$$

where $r_{t+1} = \log\left(\frac{P_{t+1}}{P_t}\right)$ is the daily log return.

```r
# Assume df_daily already has `date` and `daily_close` and is sorted by date

# 1. Compute 7-day momentum
df_daily <- df_daily %>%
  mutate(
    momentum_7d = log(close_price / lag(close_price, 7)),
    signal = case_when(
      momentum_7d > 0 ~ 1,    # Long
      momentum_7d < 0 ~ -1,   # Short
      TRUE ~ 0                # No signal
    )
  )

# 2. Shift signal forward by one day to avoid look-ahead bias
df_daily <- df_daily %>%
  mutate(
    signal_lagged = lag(signal),
    strategy_return = signal_lagged * log_return
  )
```

#Performance Comparison

```r
# Cumulative returns
df_daily <- df_daily %>%
  mutate(
    cum_ret_strategy = cumsum(coalesce(strategy_return, 0)),
    cum_ret_bh = cumsum(coalesce(log_return, 0))
  )

# Plot
library(ggplot2)
ggplot(df_daily, aes(x = date)) +
  geom_line(aes(y = cum_ret_strategy, color = "Strategy")) +
  geom_line(aes(y = cum_ret_bh, color = "Buy & Hold")) +
  labs(title = "7-Day Momentum Strategy vs Buy-and-Hold",
       x = "Date", y = "Cumulative Log Return") +
  scale_color_manual(values = c("Strategy" = "blue", "Buy & Hold" = "black"))
```

4. Extension

5. Forecasting & Backtesting

6. Conclusion