# 7831 HW 2

Charlie Yi-cy2578

April 2025

**Abstract**

This report outlines an analysis conducted on the adjusted closing prices of Nike Inc. (ticker: NKE) over a four-year period. The primary objective is to forecast returns using ARMA$(p, q)$ models combined with a GARCH(1,1) specification and to compare these forecasts with a naive random walk model.

# 1 Introduction

In this analysis, four years of adjusted closing prices for NKE were downloaded from Yahoo Finance. Log returns were computed from these prices and were confirmed to be stationary via the Augmented Dickey-Fuller (ADF) test. The forecasting framework utilizes ARMA$(p, q)$ models with GARCH(1,1) errors, estimated over a training set (first $N - 5$ observations) with the last 5 observations reserved for out-of-sample evaluation.

# 2 Methodology

## 2.1 Data Acquisition and Preprocessing

- **Data Collection:** Four years of adjusted closing prices for NKE were obtained.
- **Return Calculation:** Logarithmic returns were calculated using the daily returns function.
- **Stationarity Test:** An ADF test was performed on the returns, rejecting the null hypothesis of a unit root. This confirms that the log returns are stationary.

## 2.2 Model Estimation

- **Sample Split:** The data was split into a training set (first $N-5$ observations) and a test set (last 5 observations).
- **ARMA+GARCH Models:** Models were fitted with ARMA$(p,q)$ specifications (where $0 \leq p,q \leq 4$) combined with a GARCH(1,1) error structure.
- **Model Selection:** For each model, the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) were recorded. The best models were selected based on the lowest AIC and BIC values. In this analysis, the best AIC model was ARMA(4,4) and the best BIC model was ARMA(0,0).

## 2.3 Forecasting and Evaluation

- **Forecast Generation:** The best AIC and BIC models were used to forecast the returns for the last 5 days.
- **Random Walk Benchmark:** A random walk forecast was also constructed by using the last observed return from the training set as the forecast for all 5 days.
- **Accuracy Metrics:** Forecast performance was evaluated using bias, Mean Absolute Deviation (MAD), and Mean Squared Error (MSE).

# 3 Results

## 3.1 ADF Test

The ADF test yielded a test statistic of approximately $-9.2663$ with a printed p-value of 0.01, strongly rejecting the null hypothesis of a unit root. This confirms the stationarity of the log returns.

## 3.2 Model Selection

- **Best AIC Model:** ARMA(4,4) (AIC = -5.0150)
- **Best BIC Model:** ARMA(0,0) (BIC = -4.9881)

### 3.3 Forecast Accuracy

Forecasts for the last 5 days were compared across three models. The accuracy metrics were as follows:

- **Best AIC Model (ARMA(4,4)):**
  - Bias: 0.00018
  - MAD: 0.01501
  - MSE: 0.0003212
- **Best BIC Model (ARMA(0,0)):**
  - Bias: -0.00249
  - MAD: 0.01413
  - MSE: 0.0003008
- **Random Walk Model:**
  - Bias: -0.00532
  - MAD: 0.01463
  - MSE: 0.0003230

The ARMA(0,0) model (selected via BIC) exhibits slightly better performance in terms of MSE and MAD compared to the ARMA(4,4) model and the random walk approach.

## 4 Discussion

While both ARMA+GARCH models and the naive random walk model provided comparable forecasting accuracy over a five-day horizon, the model selected using BIC (ARMA(0,0)) showed marginally improved performance. This suggests that for short-term forecasting of returns, simpler models may sometimes be as effective as more complex ones. The minimal differences in accuracy metrics indicate that the additional parameters in the ARMA(4,4) model did not significantly enhance the forecast quality over this short period.

# 5 Conclusion

This assignment demonstrates the application of ARMA+GARCH models to forecast financial returns. After confirming the stationarity of NKE returns, multiple models were estimated and the best ones were selected based on AIC and BIC criteria. Forecast evaluations indicate that the simpler model (ARMA(0,0)) selected by BIC performs slightly better than its more complex counterpart and the random walk model. Detailed R code is attached in the Appendix.

# A R Script

Listing 1: R Script for the Reprot

```r
library(quantmod)
library(tseries)
library(rugarch)

# ------------------------------------------------------------

# 1.1 Get four years of adjusted closing prices
# Define your start and end dates (you can adjust as needed)
start_date <- as.Date("2019-04-01")
end_date   <- as.Date("2023-04-01")

# Download data from Yahoo Finance
getSymbols("NKE",
           src  = "yahoo",
           from = start_date,
           to   = end_date)

# Extract the Adjusted Close column
NKE_adj <- Ad(NKE)  # Ad() extracts the adjusted close prices

# 1.2 Calculate log returns
# We use dailyReturn() with type = "log" or manually compute diff(log(.))
NKE_returns <- dailyReturn(NKE_adj, type = "log")

# You might remove any NAs that appear (usually at the first data point)
NKE_returns <- na.omit(NKE_returns)

# 1.3 Perform the Augmented Dickey-Fuller test
adf_result <- adf.test(NKE_returns)

# Print ADF test results
print(adf_result)

# ------------------------------------------------------------
```

```r
# 2.1. Split data into training (N-5) and test (5) sets

N <- length(NKE_returns)
train_returns <- NKE_returns[1:(N-5)]  # first N-5 for model estimation
test_returns  <- NKE_returns[(N-4):N]  # last 5 for out-of-sample checks (

# 2.2 Run ARMA(p, q) + GARCH(1,1) over p,q in [0..4]
# We'll store the AIC and BIC in a data frame
results <- data.frame(
  p   = integer(),
  q   = integer(),
  AIC = numeric(),
  BIC = numeric(),
  stringsAsFactors = FALSE
)

# Loop over all p, q combinations
for (p in 0:4) {
  for (q in 0:4) {
    # Specify an sGARCH(1,1) model with ARMA(p, q) in the mean
    spec <- ugarchspec(
      variance.model = list(model = "sGARCH", garchOrder = c(1,1)),
      mean.model     = list(armaOrder = c(p,q), include.mean = TRUE),
      distribution.model = "norm"  # Normal distribution assumption
    )

    # Fit the model to the training set
    fit <- ugarchfit(spec = spec, data = train_returns, solver = "hybrid")

    # Extract info criteria
    ic <- infocriteria(fit)
    # By default, infocriteria returns: c(Akaike, Bayes, Shibata, Hannan-Q
    AIC_val <- ic[1]  # AIC
    BIC_val <- ic[2]  # BIC

    # Append to results data frame
    results <- rbind(results, data.frame(
      p   = p,
```

```
        q    = q,
        AIC = AIC_val,
        BIC = BIC_val
      ))
    }
}
# 2.3 Review the table of AIC/BIC and pick best models
print(results)

# Find the best model by AIC
best_AIC <- results[which.min(results$AIC), ]
cat("Best model by AIC:\n")
print(best_AIC)

# Find the best model by BIC
best_BIC <- results[which.min(results$BIC), ]
cat("Best model by BIC:\n")
print(best_BIC)

# ---------------------------------------------------------------

# 3.1 Use the actual p,q you found in your previous loops.
# For demonstration, let's assume:
pAIC <- best_AIC$p
qAIC <- best_AIC$q
pBIC <- best_BIC$p
qBIC <- best_BIC$q

# 3.2 Define and fit each model on the training data
# Best AIC model
specAIC <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1,1)),
  mean.model     = list(armaOrder = c(pAIC, qAIC), include.mean = TRUE),
  distribution.model = "norm"
)
fitAIC <- ugarchfit(spec = specAIC, data = train_returns, solver = "hybrid

# Best BIC model
```

```r
specBIC <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1,1)),
  mean.model     = list(armaOrder = c(pBIC, qBIC), include.mean = TRUE),
  distribution.model = "norm"
)
fitBIC <- ugarchfit(spec = specBIC, data = train_returns, solver = "hybrid

# 3.3 Forecast for the last 5 days using ugarchforecast()
fcastAIC <- ugarchforecast(fitAIC, n.ahead = 5)
fcastBIC <- ugarchforecast(fitBIC, n.ahead = 5)

# Extract the forecasted returns (mean forecasts)
# seriesFor is a matrix with columns for each step
aic_pred <- as.numeric(fcastAIC@forecast$seriesFor)
bic_pred <- as.numeric(fcastBIC@forecast$seriesFor)

# Actual returns for the test period
test_actual <- as.numeric(test_returns)  # length should be 5

# 3.3 Define a "random walk" forecast
last_train_return <- tail(train_returns, 1)
rw_pred <- rep(last_train_return, 5)

# 3.4 Compute accuracy metrics: bias, MAD, MSE
calc_errors <- function(actual, forecast) {
  e <- forecast - actual
  bias <- mean(e)
  mad  <- mean(abs(e))
  mse  <- mean(e^2)
  return(c(bias = bias, MAD = mad, MSE = mse))
}

aic_err <- calc_errors(test_actual, aic_pred)
bic_err <- calc_errors(test_actual, bic_pred)
rw_err  <- calc_errors(test_actual, rw_pred)

cat("Accuracy metrics (Bias, MAD, MSE):\n\n")
```

```
cat("Best AIC Model:\n")
print(aic_err)
cat("\n")

cat("Best BIC Model:\n")
print(bic_err)
cat("\n")

cat("Random Walk Model:\n")
print(rw_err)
cat("\n")
```