HW1 Associative Model Forecast

Group 2

2025-09-13

Table of contents

1 Explain Data

已對資料做整理，取每個月的最後一筆收盤價當成月的基礎。

再利用轉對數的形式讓小樣本的序列能更平穩

報酬率公式:

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right)$$

利用報酬率來預測輝達收盤價跟SOXX(美國半導體產業股票指數)的收盤價的關聯

本次使用ARIMAX 模型來探討，但因資料筆數為11筆，模型只能作為參考。

```
library(readr)
library(forecast)
```

```
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

```r
library(skimr)
library(knitr)
library(kableExtra)
data <- read_csv("nvdatsmc.csv")
```

```
New names:
* `` -> `...1`
```

```
Rows: 11 Columns: 6
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): MonthEnd
dbl (5): ...1, TSMC, TSMC_ret, NVDA, NVDA_ret

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
skim(data)
```

Table 1: Data summary

Name	data
Number of rows	11
Number of columns	6
_____	
Column type frequency:	
character	1
numeric	5
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
MonthEnd	0	1	9	10	0	11	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
...1	0	1	5.00	3.32	0.00	2.50	5.00	7.50	10.00	
TSMC	0	1	9.72	1.28	7.65	9.04	9.54	10.25	12.10	
TSMC_ret	0	1	-0.03	0.10	-0.18	-0.07	-0.07	-0.01	0.22	
NVDA	0	1	137.52	23.58	108.38	122.50	134.29	148.12	177.87	
NVDA_ret	0	1	0.03	0.11	-0.14	-0.03	0.04	0.10	0.22	

```
library(tseries)
library(knitr)
library(broom)
nvda_adf <- adf.test(data$NVDA_ret)
soxx_adf <- adf.test(data$TSMC_ret)
kable(tidy(nvda_adf), caption = "ADF test on NVDA returns", digits = 4)
```

Table 4: ADF test on NVDA returns

statistic	p.value	parameter	method	alternative
-1.8792	0.6184	2	Augmented Dickey-Fuller Test	stationary

```
kable(tidy(soxx_adf), caption = "ADF test on SOXX returns", digits = 4)
```

Table 5: ADF test on SOXX returns

statistic	p.value	parameter	method	alternative
-2.0361	0.5586	2	Augmented Dickey-Fuller Test	stationary

```
#| fig-align: "center"
#| fig-width: 10
#| fig-height: 5
y <- data$NVDA_ret
X <- stats::lag(data$TSMC_ret, -1)   #    1   t   NVDA    t-1   SOXX
df <- data.frame(y=y, X=X)
df <- na.omit(df)

y_ts <- ts(df$y, frequency = 12, start = c(2024,8))  #   2024   8
X_mat <- as.matrix(df$X)
# ----    ARIMAX ----
fit <- auto.arima(y_ts, xreg = X_mat, seasonal = FALSE,
```

Table 6: ARIMAX Model Coefficients

term	estimate	std.error
xreg	-0.0091721	0.3160346

Table 7: ARIMAX Model Fit Summary

sigma	logLik	AIC	BIC	nobs
0.1140704	8.796218	-13.59244	-12.79665	11

```
                  stepwise = FALSE, approximation = FALSE)
library(broom)
library(knitr)
library(kableExtra)

#
coef_tab <- tidy(fit)
kable(coef_tab, format="latex", booktabs=TRUE,
      caption="ARIMAX Model Coefficients") %>%
  kable_styling(latex_options="scale_down", position="center")
```

```
#    (AIC, BIC, logLik)
model_info <- glance(fit)
kable(model_info, format="latex", booktabs=TRUE,
      caption="ARIMAX Model Fit Summary") %>%
  kable_styling(latex_options="scale_down", position="center")
```

```
bp <- tidy(Box.test(fit$residuals, type="Box-Pierce", lag=1))
kable(bp, format="latex", booktabs=TRUE,
      caption="Box-Pierce Test on Residuals") %>%
  kable_styling(latex_options="scale_down", position="center")
```

Table 8: Box-Pierce Test on Residuals

statistic	p.value	parameter	method
0.9596366	0.3272785	1	Box-Pierce test

Table 9: NVDA Forecast (h = 4)

Step	Mean	Lo80	Hi80	Lo95	Hi95
1	0.000405	-0.145782	0.146592	-0.223169	0.223978
2	0.000405	-0.145782	0.146592	-0.223169	0.223978
3	0.000405	-0.145782	0.146592	-0.223169	0.223978
4	0.000405	-0.145782	0.146592	-0.223169	0.223978

```
#| fig-align: "center"
#| fig-width: 10
#| fig-height: 5

future_X <- matrix(rep(tail(df$X,1), 4), ncol = 1)
colnames(future_X) <- "X"
fc <- forecast(fit, xreg = future_X, h=4)
```

```
Warning in forecast.forecast_ARIMA(fit, xreg = future_X, h = 4): xreg contains
different column names from the xreg used in training. Please check that the
regressors are in the same order.
```

```
library(knitr)
library(kableExtra)

fc_table <- data.frame(
  Step  = 1:length(fc$mean),
  Mean  = round(as.numeric(fc$mean), 6),
  Lo80  = round(as.numeric(fc$lower[, "80%"]), 6),
  Hi80  = round(as.numeric(fc$upper[, "80%"]), 6),
  Lo95  = round(as.numeric(fc$lower[, "95%"]), 6),
  Hi95  = round(as.numeric(fc$upper[, "95%"]), 6)
)

kable(
  fc_table,
  format = "latex", booktabs = TRUE,
  caption = "NVDA Forecast (h = 4)",
  align = c("c", rep("r", 5))
) %>%
  kable_styling(latex_options = "scale_down", position = "center")
```

```r
h <- 4
ret4 <- as.numeric(tail(fc$mean, h))

#   →
last_price <- as.numeric(tail(data$NVDA, 1))
price4 <- last_price * cumprod(exp(ret4))

#    =4
pred_table <- data.frame(
  Month = c("2025-09", "2025-10", "2025-11", "2025-12"),
  NVDA_ret_forecast   = ret4,
  NVDA_price_forecast = price4,
  stringsAsFactors = FALSE
)



library(knitr)
library(kableExtra)
library(dplyr)
```

```
Attaching package: 'dplyr'

The following object is masked from 'package:kableExtra':

    group_rows

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
kable(
  pred_table %>%
    mutate(
      NVDA_ret_forecast   = round(NVDA_ret_forecast, 8),
      NVDA_price_forecast = round(NVDA_price_forecast, 4)
    ),
```

Month	NVDA_ret_forecast	NVDA_price_forecast
2025-09	0.0004046	174.2505
2025-10	0.0004046	174.3210
2025-11	0.0004046	174.3916
2025-12	0.0004046	174.4622

```
  format = "latex", booktabs = TRUE,
  col.names = c("Month", "NVDA_ret_forecast", "NVDA_price_forecast"),
  align = c("c","r","r")
) %>%
  kable_styling(latex_options = "scale_down", position = "center")
```

```
#| fig-align: "center"
#| fig-width: 10
#| fig-height: 5


set.seed(42)   #
h <- 4         #    NVDA 4
#   NVDA_t   SOXX_{t-1}     3    10~12


y <- data$NVDA_ret
X <- stats::lag(data$TSMC_ret, -1)        #   t   NVDA   t-1   SOXX
df <- na.omit(data.frame(y = y, X = X))

y_ts  <- ts(df$y,  frequency = 12, start = c(2024, 8))
X_mat <- as.matrix(df$X)

fit_nvda <- auto.arima(
  y_ts, xreg = X_mat, seasonal = FALSE,
  stepwise = FALSE, approximation = FALSE
)


# (1)   SOXX_ret   ARIMA   SOXX

soxx_ts <- ts(na.omit(data$TSMC_ret), frequency = 12, start = c(2024, 8))
fit_soxx <- auto.arima(
  soxx_ts, seasonal = FALSE,
  stepwise = FALSE, approximation = FALSE
)
```

```r
# (2)      SOXX_ret   3
# simulate 2000 times

n_sim <- 2000
lastX <- as.numeric(tail(soxx_ts, 1))

#    3    SOXX ARIMA    3
#      SOXX   3
sim_one_soxx <- function() {
  as.numeric(simulate(fit_soxx, nsim = 3, future = TRUE))
}

soxx_future_mat <- t(replicate(n_sim, sim_one_soxx()))  # n_sim x 3


# (3)       NVDA   ARIMAX
#
#         N()
#
## 95% = mean ± z * sd  →  sd   (upper-mean)/z
draw_one_from_fc <- function(fc_obj) {
  m  <- as.numeric(fc_obj$mean)
  lo <- as.numeric(fc_obj$lower[, "95%"])
  hi <- as.numeric(fc_obj$upper[, "95%"])
  z  <- qnorm(0.95)
  sd <- (hi - m) / z
  rnorm(length(m), mean = m, sd = sd)
}

nvda_ret_sims <- matrix(NA_real_, nrow = n_sim, ncol = h)

for (i in 1:n_sim) {
  # lag   1  NVDA   lastX      3
  future_X <- c(lastX, soxx_future_mat[i, ])
  fc_i <- forecast(fit_nvda, xreg = matrix(future_X, ncol = 1), h = h)
  #            ARIMAX
  nvda_ret_sims[i, ] <- draw_one_from_fc(fc_i)
}


# (4)   →

last_price <- as.numeric(tail(data$NVDA, 1))  #
```

```r
#
nvda_price_sims <- t(apply(nvda_ret_sims, 1, function(r) last_price * exp(cumsum(r))))


# (5)     P10 P90

quant <- function(x, p) as.numeric(quantile(x, probs = p, na.rm = TRUE))
months <- c("2025-09","2025-10","2025-11","2025-12")
library(knitr)
library(kableExtra)

summary_table <- data.frame(
  Month = months,
  Ret_Mean = apply(nvda_ret_sims, 2, mean),
  Ret_Median = apply(nvda_ret_sims, 2, median),
  Ret_P10 = apply(nvda_ret_sims, 2, quant, p = 0.10),
  Ret_P90 = apply(nvda_ret_sims, 2, quant, p = 0.90),
  Price_Mean = apply(nvda_price_sims, 2, mean),
  Price_Median = apply(nvda_price_sims, 2, median),
  Price_P10 = apply(nvda_price_sims, 2, quant, p = 0.10),
  Price_P90 = apply(nvda_price_sims, 2, quant, p = 0.90)
)

kable(
  summary_table %>%
    mutate(
      Ret_Mean     = round(Ret_Mean, 6),
      Ret_Median   = round(Ret_Median, 6),
      Ret_P10      = round(Ret_P10, 6),
      Ret_P90      = round(Ret_P90, 6),
      Price_Mean   = round(Price_Mean, 2),
      Price_Median = round(Price_Median, 2),
      Price_P10    = round(Price_P10, 2),
      Price_P90    = round(Price_P90, 2)
    ),
  format = "latex", booktabs = TRUE,
  caption = "NVDA Forecast (Monte Carlo Simulation, 2000 runs)",
  align = c("c", rep("r", 8))
) %>%
  kable_styling(latex_options = "scale_down", position = "center")
```

```r
# ------------------------------
# 6)     bear/base/bull
```

Table 10: NVDA Forecast (Monte Carlo Simulation, 2000 runs)

Month	Ret_Mean	Ret_Median	Ret_P10	Ret_P90	Price_Mean	Price_Median	Price_P10	Price_P90
2025-09	0.001100	0.005126	-0.178079	0.178011	176.04	175.08	145.77	208.12
2025-10	-0.000620	-0.001203	-0.181543	0.168849	177.47	174.10	136.75	222.49
2025-11	-0.004456	-0.008315	-0.176493	0.169850	178.27	173.79	128.53	233.43
2025-12	-0.001409	0.000377	-0.174285	0.169114	179.61	172.63	123.31	243.93

```
#      SOXX_ret   ARIMA    ±
# ------------------------------
#   SOXX   3     95%    →    sd
fc_soxx <- forecast(fit_soxx, h = 3)
m_sx <- as.numeric(fc_soxx$mean)
sd_sx <- (as.numeric(fc_soxx$upper[, "95%"]) - m_sx) / qnorm(0.95)

scenarios <- list(
  bear = m_sx - sd_sx,
  base = m_sx,
  bull = m_sx + sd_sx
)

run_scn <- function(path3) {
  future_X <- c(lastX, path3)
  fc <- forecast(fit_nvda, xreg = matrix(future_X, ncol = 1), h = h)
  #
  r_mean <- as.numeric(fc$mean)
  price_path <- last_price * exp(cumsum(r_mean))
  data.frame(
    Month = months,
    NVDA_ret = r_mean,
    NVDA_price = price_path
  )
}

scn_bear <- run_scn(scenarios$bear)
scn_base <- run_scn(scenarios$base)
scn_bull <- run_scn(scenarios$bull)

print_scn <- function(df, title){
  kable(
    df %>%
      mutate(
        NVDA_ret   = round(NVDA_ret, 6),
        NVDA_price = round(NVDA_price, 2)
```

Table 11: Scenario: Bear

Month	NVDA_ret_forecast	NVDA_price_forecast
2025-09	0.000405	174.25
2025-10	0.001134	174.45
2025-11	0.001134	174.65
2025-12	0.001134	174.84

Table 12: Scenario: Base

Month	NVDA_ret_forecast	NVDA_price_forecast
2025-09	0.000405	174.25
2025-10	0.000000	174.25
2025-11	0.000000	174.25
2025-12	0.000000	174.25

```
      ),
    format = "latex", booktabs = TRUE,
    caption = paste("Scenario:", title),
    col.names = c("Month", "NVDA_ret_forecast", "NVDA_price_forecast"),
    align = c("c","r","r")
  ) %>%
    kable_styling(latex_options = "scale_down", position = "center")
}

print_scn(scn_bear, "Bear")
```

```
print_scn(scn_base, "Base")
```

```
print_scn(scn_bull, "Bull")
```

Table 13: Scenario: Bull

Month	NVDA_ret_forecast	NVDA_price_forecast
2025-09	0.000405	174.25
2025-10	-0.001134	174.05
2025-11	-0.001134	173.86
2025-12	-0.001134	173.66