FIN3018 Financial Econometrics and Data Science

Time Series Modelling: Mean vs Volatility Dynamics

Individual Project

AY 2025/2026

Group: S

Assigned data set: S&P 500 Industrials, Period 1

Ziyi Wu

Word count:997 (inc. titles)

FIN3018 Financial Econometrics and Data Science

Coursework submission form

Name: Ziyi Wu

Group: S

☒ I declare that the work submitted represents my own work. I declare that the work has not been taken from the works of others save where appropriately referenced.

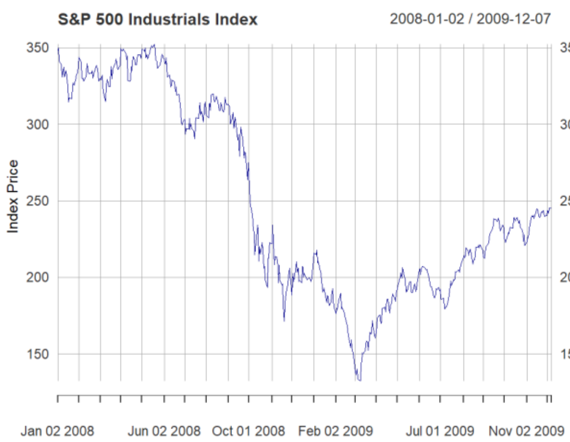
☒ I declare that all use of generative AI has been detailed in this document.

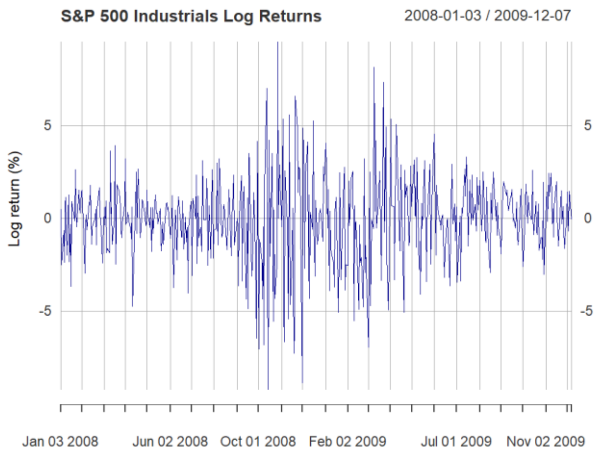
☒ By submitting this form I declare that the items listed below have been added to Canvas. I acknowledge that the report will be considered incomplete without all items listed, and the relevant penalty for late submission will be applied in respect of any omissions:

- Cover sheet
- Report, subject to a maximum of 1,000 words
- R Scripts

Introduction

This report begins by examining the time-series characteristics of the S&P 500 Industrials Index (Period 1). Plots of the full index and its corresponding log returns were generated to visually assess their behaviour.

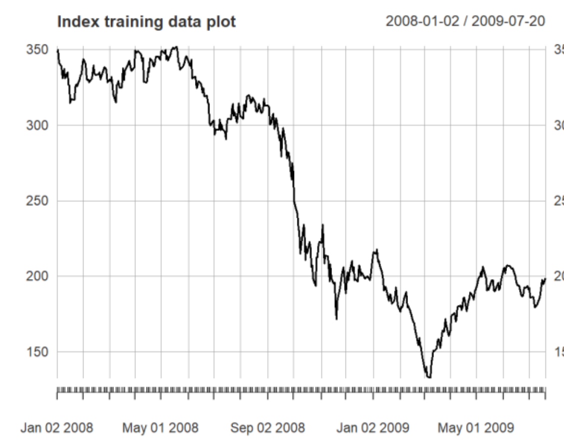


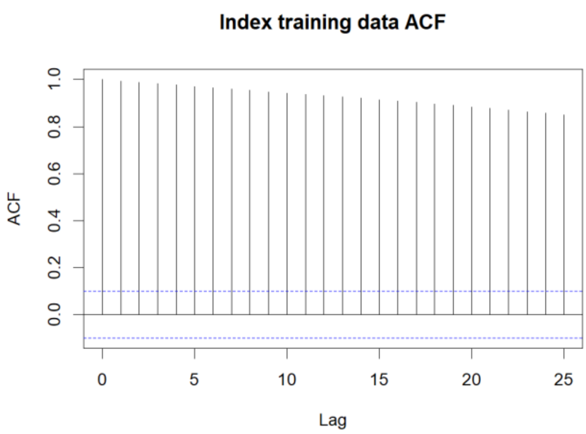


Data Comparison

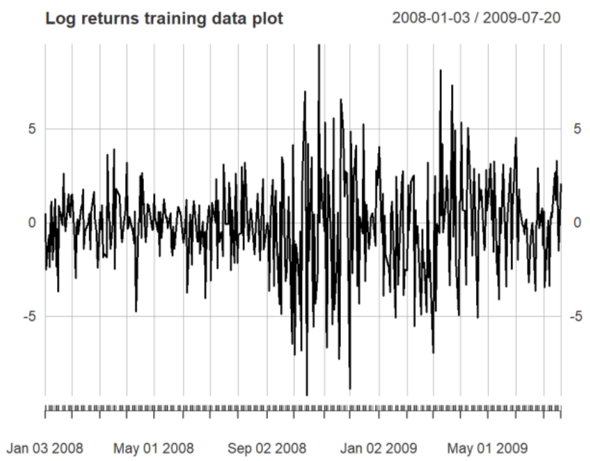
The index price level shows a persistent upward trend and noticeable volatility, suggesting non-stationarity. In contrast, the log returns fluctuate around a mean close to zero, indicating stationarity, though volatility clustering is evident. These features are consistent with the MSCI World Index from the group project.

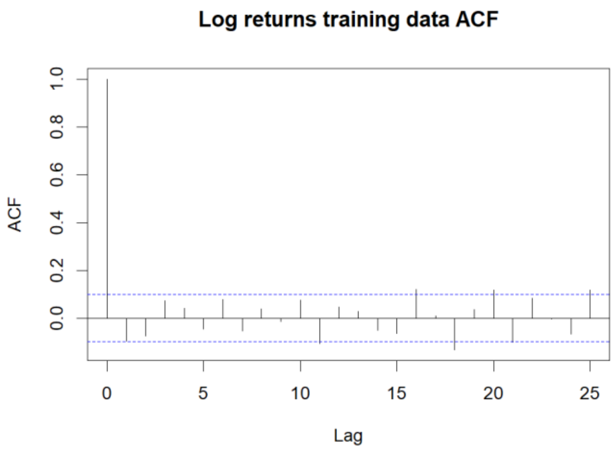
To confirm these visual impressions, the data were divided into an 80% training set (ending 2009-07-20) and a 20% test set. For the training data, the index plot retains a clear trend, while the Autocorrelation Function (ACF) decays slowly—both characteristic of a non-stationary, integrated I(1) process. The Augmented Dickey-Fuller (ADF) test supports this: the test statistic (-1.2879) exceeds the 5% critical value (-3.42), so the null of a unit root cannot be rejected. The index series is therefore non-stationary.





Next, a similar investigation conducted for the log returns led to the opposite conclusion. Visually, the plot shows the series reverting towards a constant mean. This is supported by the ACF plot, which cuts off sharply after lag 0 – behaviour characteristic of a stationary I(0) process. The ADF test (type = "none") further solidifies this conclusion: the test statistic (-15.7029) is substantially smaller (more negative) than the 5% critical value (-1.95), allowing for a strong rejection of the null hypothesis. Consequently, the log returns training data can be considered stationary.





In summary, this analysis shows two key findings:

- The index price series is **I(1) non-stationary**, confirming it requires differencing for modelling.
- The log return series is **I(0) stationary**, confirming it is suitable for volatility modelling.

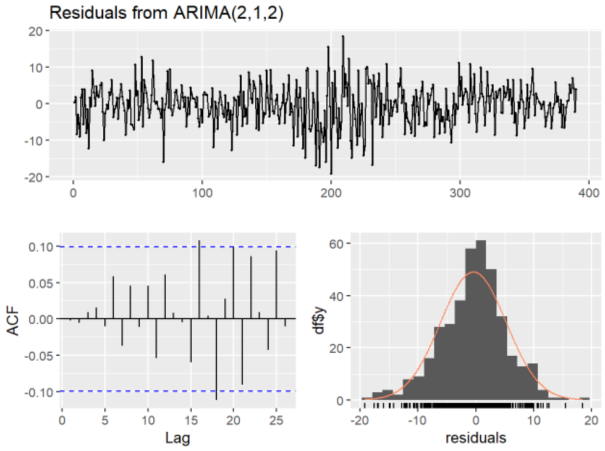
This matches the group project and confirms typical financial index behaviour.

ARIMA Estimation and Forecasting

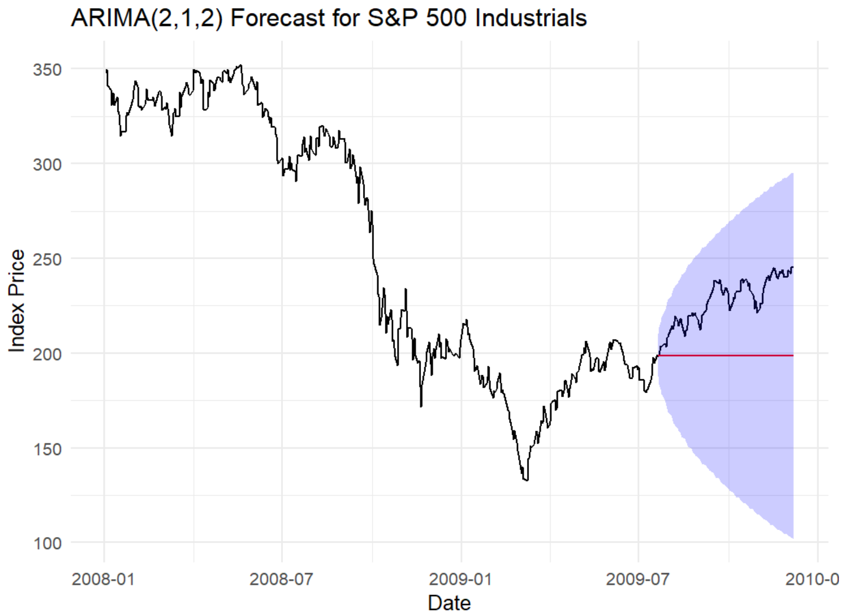
Following the data analysis result, the ARIMA(2,1,2) model (the order from the group project) was estimated on the training data.

The resulting equation is: $(1 - 0.0124B + 0.4251B^2)(1 - B)\, y_t = (1 - 0.1336B + 0.3596B^2)\, \varepsilon_t$

A residual analysis was performed to check the fit. The Ljung-Box test p-value of 0.7028 ($P > 0.05$) confirms the residuals are white noise. The model is therefore a good statistical fit.



The forecast plot illustrates the model's predicted values and 95% intervals alongside the full series. The evaluation metrics (RMSE 31.19, MAPE 12.35%) suggest the model fits well in-sample but fails to capture the test-period dynamics.



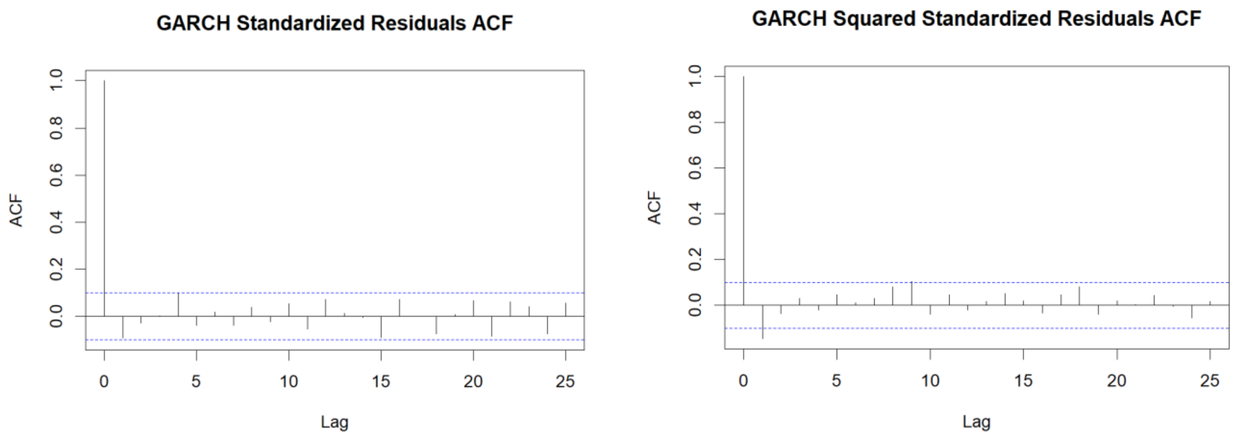
GARCH Estimation and Forecasting

To model the volatility clustering, a GARCH (1,1) model was estimated on the stationary log returns. The estimated equation is:

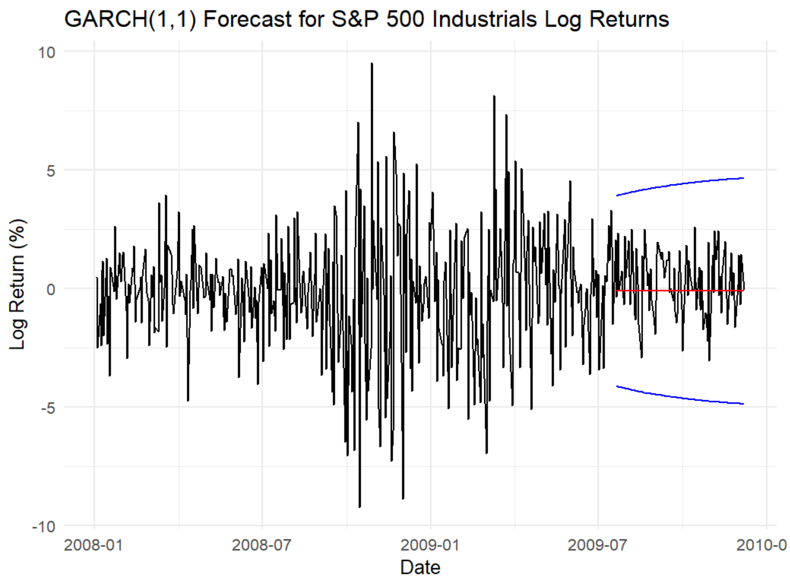
$$h_t = 0.0822 + 0.0876\varepsilon_{t-1}^2 + 0.9000h_{t-1}$$

The ($\alpha1$=0.0876) and ($\beta1$=0.9000) coefficients were highly significant, confirming ARCH/GARCH effects. The volatility persistence $\alpha1+\beta1$=0.9876, indicating shocks are highly persistent.

A residual analysis was performed on the squared standardised residuals. The Ljung-Box test yielded a p-value of 0.0513 (at lag 10). $P > 0.05$, shows that the model is a good fit and successfully captured the volatility clustering.



The visualisation displays the full log returns, the forecast, and the 95% prediction intervals. The forecast accuracy for the mean was RMSE: 1.3205.



Conclusions and Comparison with Group Work

This section compares my results (2008–2011) with the group's (2023–2024) to assess whether the same conclusions were reached.

The answer is mixed. Regarding the underlying data properties and the suitability of the chosen models, the conclusions were indeed identical. As detailed below, both datasets confirmed the standard behaviour of financial time series and the appropriateness of the ARIMA and GARCH frameworks. However, regarding forecasting performance, the conclusions were notably different, highlighting the impact of distinct market regimes.

Similar Findings:

- **Data Properties:** Both analyses confirmed that the index prices were I(1) non-stationary, while the log returns were I(0) stationary. This indicates that the basic statistical patterns of the indices remain steady even with varying market conditions and time frames.
- **Model Fit:** Both the ARIMA(2,1,2) and GARCH(1,1) models were found to be good statistical fits for their respective datasets. Both models successfully produced white noise residuals (ARIMA p-values: 0.4273 vs 0.7028; GARCH Ljung-Box p-values: 0.0777 vs 0.051), validating their structural appropriateness. Both GARCH models also confirmed extremely high volatility persistence (0.935 vs 0.988).

Differences in prediction results:

- **ARIMA:** Forecasting performance diverged significantly. The group's model had a very high RMSE (172.24) but low MAPE (4.41%), whereas the individual model showed the opposite pattern (RMSE 31.19, MAPE 12.35%).
- **GARCH:** The GARCH mean forecast accuracy also differed, with the group's RMSE (0.670) being superior to the individual RMSE (1.321).

The divergence reflects different market regimes. The group data (2023-2024) represents a recent period, while the individual data (2008-2011) covers the extreme downturn and subsequent non-linear recovery following the Global Financial Crisis. The individual ARIMA model, effectively a random walk, failed to predict this structural break, resulting in its high MAPE. This comparison confirms that while ARIMA/GARCH models are structurally sound, their forecasting power is unreliable during periods of extreme volatility and non-linear market movements.

Reflection

The key learning outcome for me was understanding the difference between a statistically adequate model and one that delivers meaningful forecasts. Our team divided tasks efficiently: others led the Data Analysis and ARIMA modelling, while another member and I jointly handled the GARCH(1,1) forecasting and estimation. Together, we interpreted the alpha and beta coefficients, calculated volatility persistence, and performed the Ljung–Box diagnostics. We also contributed to drafting the “ARIMA & GARCH Evaluation” section.

The area I contributed least to was the ARIMA model selection, which was handled by other team members. My individual ARIMA model was well-fitted (Ljung–Box p = 0.7028) but produced a weak forecast (MAPE = 12.35%), reinforcing that predictive accuracy depends on both model design and market conditions.

Our team worked efficiently through a clear division of tasks and regular communication. Defining roles early helped us stay organised, while continuous updates kept everyone aligned. This collaboration made me appreciate how planning and communication directly improve the quality and consistency of analytical work.

Appendix – Generative AI Declarations

Generative AI Statement – please briefly discuss the use of GenAI in your group work
For my individual report, I only used Generative AI for its acceptable diagnostic use in R Studio. When I had problems with running my R code such as colnames issues, object not found errors where I couldn't determine fixes for them, I used the AI to assist me in changing the code that wouldn't execute. The AI was not used to create code from scratch or to finalize the analysis.

Please paste prompts and responses from GenAI tools below

Error in colnames<-(*tmp*, value = "upper") : attempt to set 'colnames' on an object with less than two dimensions. Why?

Show thinking

This is an excellent and very common R error. It's a subtle but important one.

The error message `attempt to set 'colnames' on an object with less than two dimensions` means exactly what it says: you are trying to give a "column name" to an object that R considers to be a **one-dimensional (1D) vector** (just a list of numbers).

The `colnames()` function only works on **two-dimensional (2D) objects**, like a matrix or a data frame.

Why Is This Happening?

1. When you run the `predict()` function, the `forecast_return` object it creates is often a 2D object (a matrix with one column). This is why your first line, `colnames(forecast_xts) <- "forecast"`, **works**.
2. However, when you use R to do math with these objects (`upper_pi <- forecast_return + 1.96 * forecast_std_dev`), R "simplifies" the result into a simple 1D vector.
3. When you then run `xts(as.matrix(upper_pi), ...)` you are creating a 1D `xts` object.
4. The very next line, `colnames(upper_xts) <- "upper"`, fails because this 1D object has no "columns" to name.

The Solution

The fix is to **force** your 1D vectors (`upper_pi` and `lower_pi`) back into a 2D matrix *before* you create the `xts` object. The function for this is `as.matrix()`.

Error: object 'h_grach' not found. Why?

Show thinking

This is a common and frustrating **typo**.

The error message `Error: object 'h_grach' not found` means R cannot find an object with that specific spelling.

The Reason Why

In your R script, you correctly defined the variable for the forecast length with the spelling `h_garch` (g-a-r-c-h):

R

```r
# You defined the variable here:
h_garch <- length(log_returns_test_decimals)
```

However, in the *next* line (the one that caused the error), you accidentally misspelled it as `h_grach` (g-r-a-c-h), swapping the 'c' and 'h':

R

```r
# Your failing code (with the typo):
fcast_garch_individual <- predict(garch_individual, n.ahead = h_grach, plot = FALSE
```