Time Series Analysis of S&P 500 Daily Prices (2014–2020)

Lionel NGAWA

ln57@students.uwf.edu

Chapter 1: Introduction

This project presents a comprehensive time series analysis of the S&P 500 indexna leading benchmark of the U.S. equity market representing 500 of the largest publicly traded companies. The dataset, obtained from Yahoo Finance, comprises daily records from January 2014 through November 2020, capturing the following variables for each trading day: date, open, high, low, close, adjusted close (accounting for dividends and splits), and trading volume. Our study focuses primarily on the **adjusted close** prices, from which we compute daily log returns to facilitate robust financial modeling.

The main objectives of this analysis are two fold:
1. **Trend and Forecasting:** To model the underlying trend and produce reliable forecasts of the S&P 500 index using classical and modern time series techniques.
2. **Volatility Modeling:** To analyze and predict the dynamic volatility of S&P 500 returns, accounting for the well known phenomenon of volatility clustering observed in financial markets.

Secondary goals include **model comparison** of different models and the **interpretation of results** in the context of economic and financial theory, with attention to limitations and opportunities for model improvement.

Literature Review

Time series analysis of financial data such as stock indices has a rich tradition. According to the Efficient Market Hypothesis (EMH) articulated by Fama, financial markets rapidly incorporate available information, so price changes are largely unpredictable and follow a random walk or unit root process (Fama 1970); (Cont 2001). This hypothesis is supported by extensive empirical studies, which find little to no mean reversion in major indices such as the S&P 500 (Challa, Malepati, and Kolusu 2020); (Kontopoulou et al. 2023).

A crucial characteristic of financial time series is **volatility clustering** periods of high volatility are often followed by further high volatility, and calm periods follow calm ones (Cont 2001). To address this, Engle introduced the **ARCH** model, enabling the conditional variance of returns to depend on past shocks (Engle 1982). Bollerslev subsequently developed the **GARCH** model, which further captures the persistent memory in volatility by including lagged variances (Bollerslev 1986). These models have become standard for modeling and forecasting financial volatility, offering important applications in risk management and Value at Risk estimation (Cont 2001).

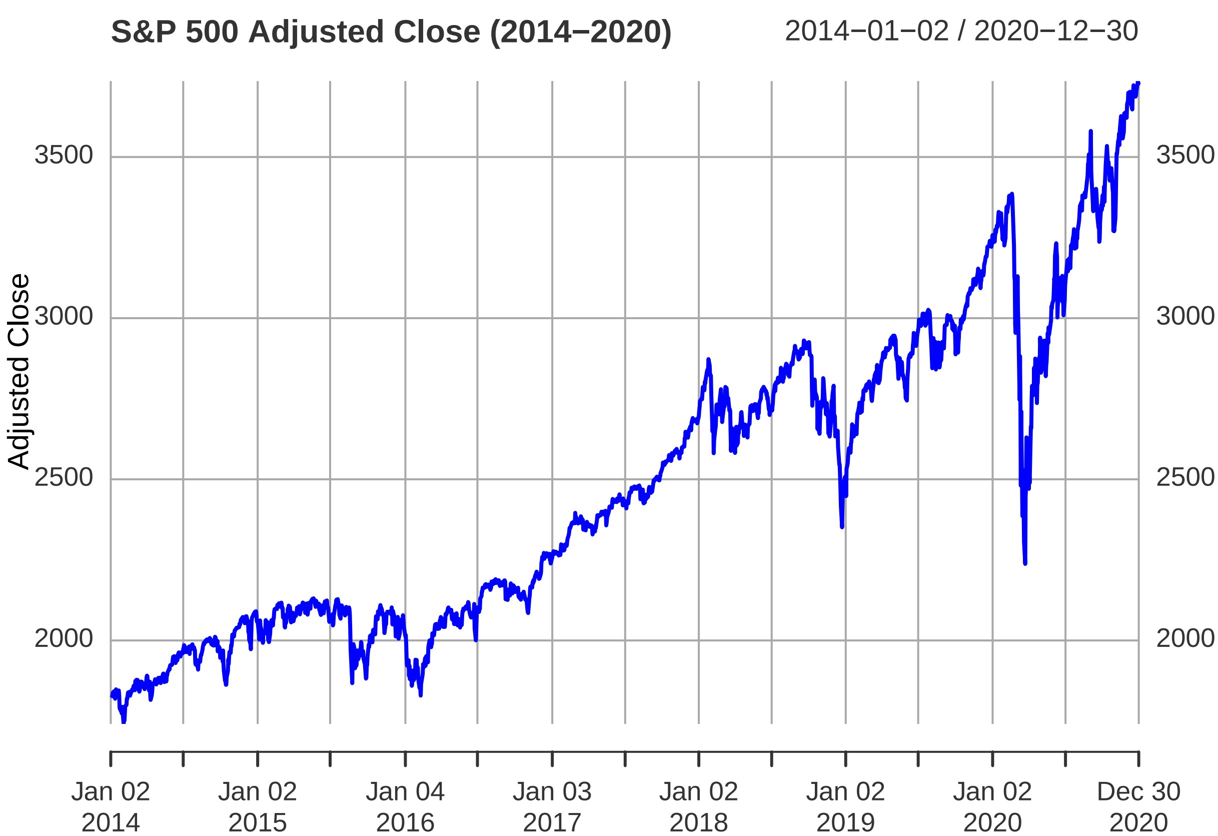
Empirical research demonstrates that **combining mean and volatility models** (such as ARIMA-GARCH hybrids) can improve modeling of both expected returns and risk, especially under volatile market conditions (Jin 2025); (Challa, Malepati, and Kolusu 2020). However, even these advanced models may require adaptations such as heavy tailed error assumptions during extreme events like the 2008 financial crisis or the COVID-19 pandemic (Jin 2025). Machine learning approaches and nonlinear time series models are increasingly explored as alternatives to classical methods, with some evidence of superior performance in forecasting complex financial data (Kontopoulou et al. 2023).

In summary, our analysis applies and evaluates these time series methodologies: ARIMA, ARCH/GARCH, and hybrid models to the S&P 500 index, aiming to generate insight into both the trend and the dynamic volatility of the market. All modeling choices and interpretations are informed by the established literature in financial econometrics and recent comparative studies (Box and Jenkins 1970); (Fama 1970); (Engle 1982); (Bollerslev 1986); (Cont 2001); (Jin 2025); (Kontopoulou et al. 2023); (Challa, Malepati, and Kolusu 2020).

Chapter 2: Preliminary Analysis

We begin with a visual inspection of the S&P 500 adjusted closing price over time. The raw time series plot (Figure 1) shows that from 2014 to 2020 the index exhibited an overall upward trend with notable fluctuations. There are periods of gradual growth punctuated by episodes of volatility. In particular, early 2020 stands out as a period of a sharp drop (the COVID-19 market crash in March 2020) followed by a rapid rebound. This upward drift and the non-constant mean indicate that the price series is nonstationary in level.

Time serie plot

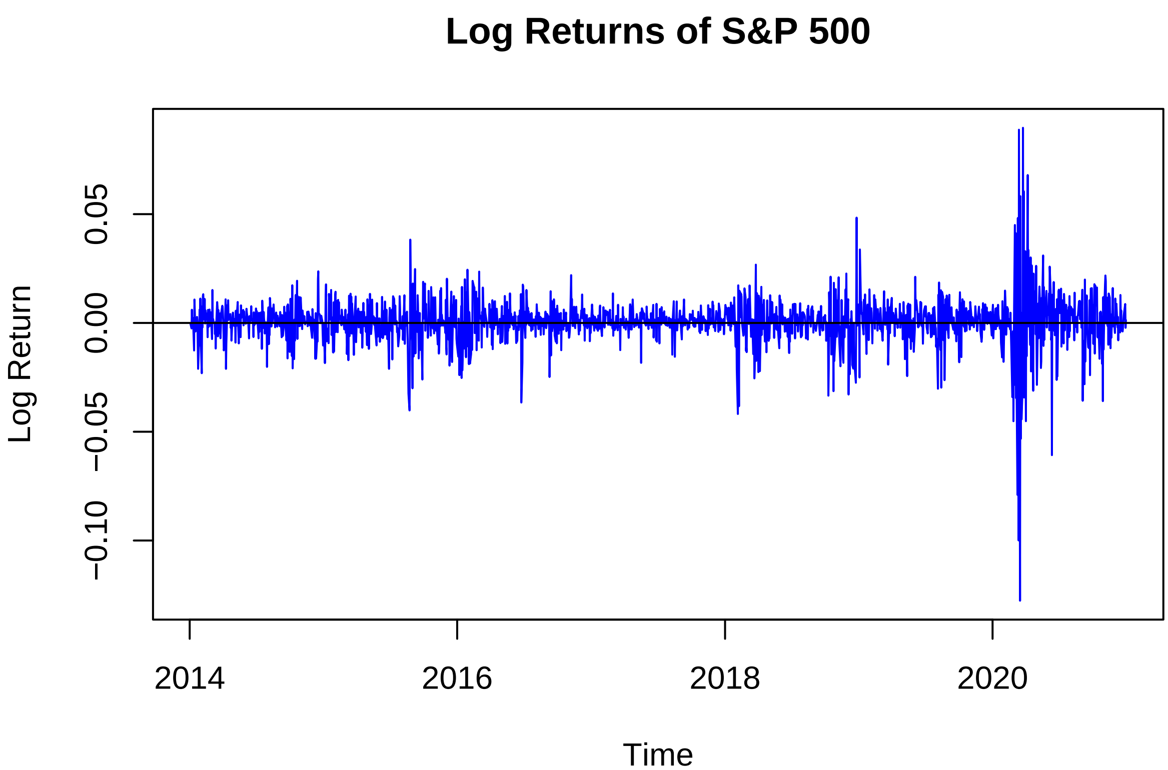


The index shows a long-run upward trend. Notable deviations include corrections in 2015–2016, 2018, and the sharp crash in early 2020 (COVID-19) followed by a rapid recovery. The persistent upward trend suggests the raw price is non-stationary, requiring differencing for many time-series models. So, to better analyse the series statistically, we transform price to log returns. Log returns are approximately the percentage change and are a common stationary transformation for financial data.

We examine basic statistics of the daily log return. In R, we can see that using the summary of the logReturn, the skewness and the kurtosis function, we can see that the mean daily log return is around 4×10^{-4} so approximately 0.04% per day (which annualizes to about 10ù per year consistent with the index's long-run growth). Moreover, we can see how the distribution ois non normal because the log return exibt a negative skewness (approximately -1.05) indicating the left tail is longer and excess kurtosis on the order of 21 indicating a heavy tails .The minimum one day log return in thsi period was about -12.8% in march 2020 and the maximum was about +9% in april 2020 reflecting the extreme volatility during the COVID. crash.

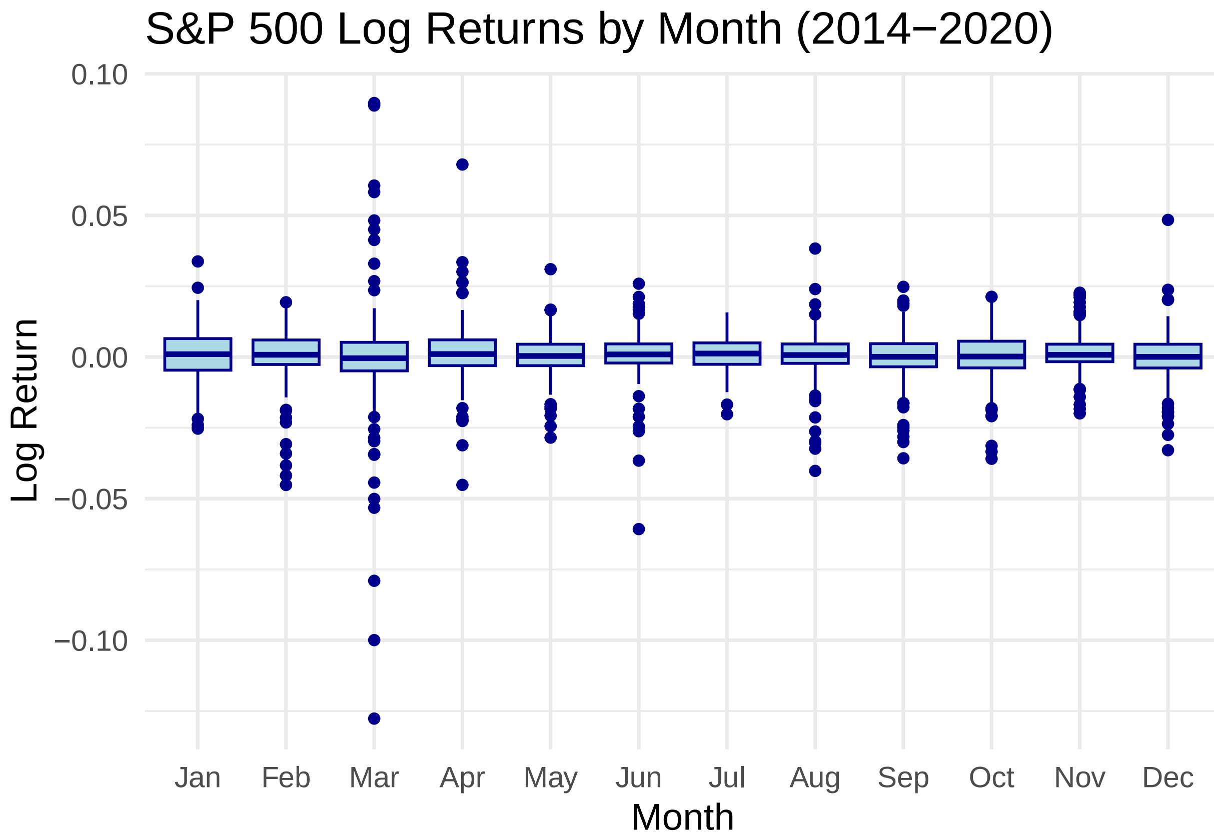
Log Returns Time Series

We can see that after transforming the raw data into a log return, the serie fluctuate around zero mean showing no long term trend but displays volatility clustering. Period of relative calm small change are followed by volatile periods. Notably, 2027-2019 were relatively steady whereas early 2020 shows an explosion of volatility during the covid-19.Such patterns suggest conditional heteroskedasticity, the variance of returns changes over time – which motivates ARCH/GARCH models.



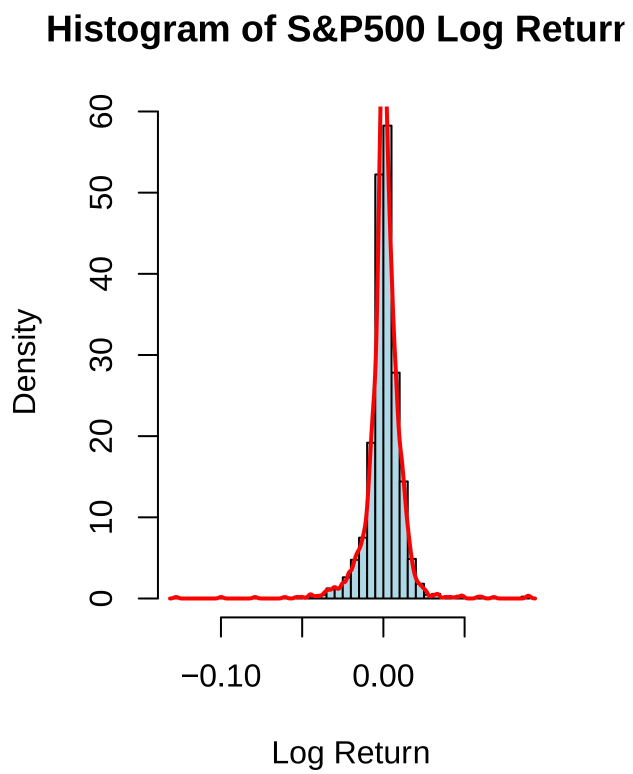
From the plot above, we can see that returns have heteroskedastic variance: for example, 2014–2015 had moderate volatility, 2016 saw spikes around market corrections, 2017 was unusually calm (low volatility), late 2018 volatility rose (market sell off), and 2020 had extreme volatility. This visual evidence of volatility clustering (quiet and turbulent periods) will be confirmed by statistical tests and is a strong indication that ARCH/GARCH models will be appropriate.

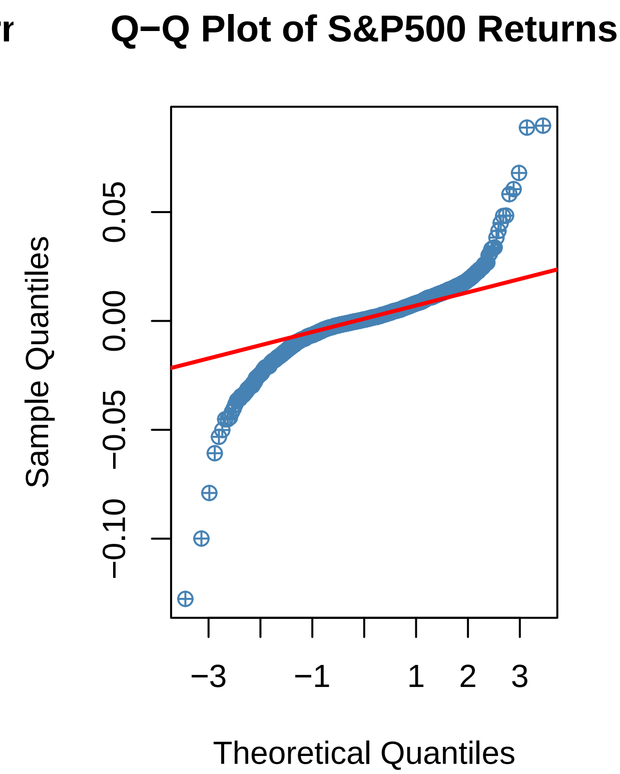
Boxplot



The boxplot of S&P 500 log returns above highlights a significant number of outliers, particularly on the negative side, consistent with a heavy-tailed and negatively skewed distribution. This suggests that extreme losses occur more frequently than extreme gains, underscoring the need for volatility models and heavy-tailed error assumptions in financial time series analysis (Cont 2001). These characteristics have important implications for both risk management and return forecasting.

Histogram and QQ plot





The histogram and Q-Q plot of S&P 500 daily log returns both highlight pronounced deviations from normality. The distribution is sharply peaked at zero and exhibits substantial heavy tails, as seen in both the density curve and the spread of Q-Q plot points away from the line. These features indicate that extreme returns especially large lossesn occur more frequently than predicted by a Gaussian model. Such heavy-tailed, negatively skewed behavior is a well-documented stylized fact in financial econometrics (Cont 2001). This finding motivates the use of volatility models (such as GARCH) and fat-tailed error distributions in risk management and return forecasting, as standard models assuming normality tend to underestimate the probability and magnitude of extreme market events.

Stationarity test

To test the statonarity of the GSPX.Adjusted of the S&P 500, we apply the augmented Dicker Fuller (ADF) test to check the stationarity of the time series. the table below precise only the type 1 model:
The null hypothesis H_0 is that the series is non-stationary (has a unit root). The alternative H_1 hypothesis is stationarity.

Summary Table

Type 1: No drift, no trend

lag	ADF	p.value
0	1.94	0.987
1	2.35	0.990

lag	ADF	p.value
2	2.09	0.990
3	1.97	0.989
4	2.16	0.990
5	2.10	0.990
6	2.33	0.990
7	1.98	0.989

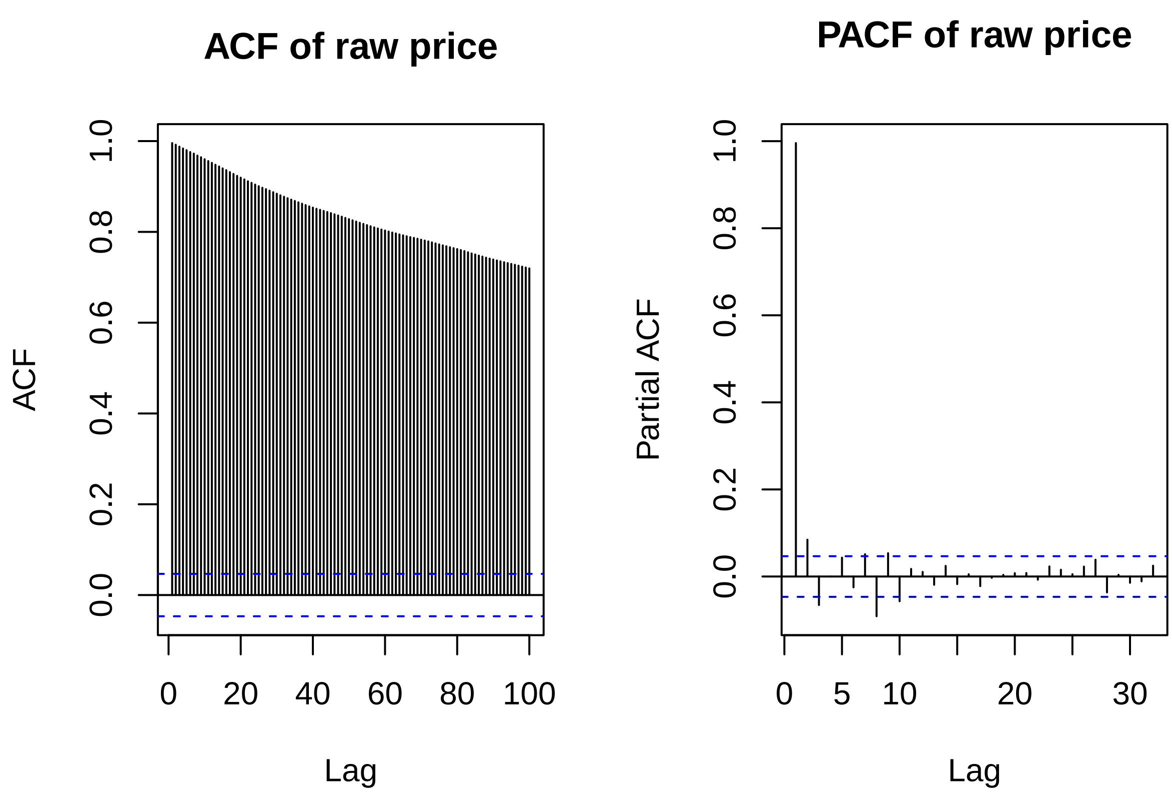
The ADF test on the price series fails to reject the null hypothesis (test statistic around –0.987 with p»0.05, confirming that the index level is a unit-root process (the series is non-stationary) and needs differencing.

In contrast the ADF test on log return strongly rejects the null of unit root (test statistic ≈-12.94, p-value«0.05), indicating the return serie is stationary.

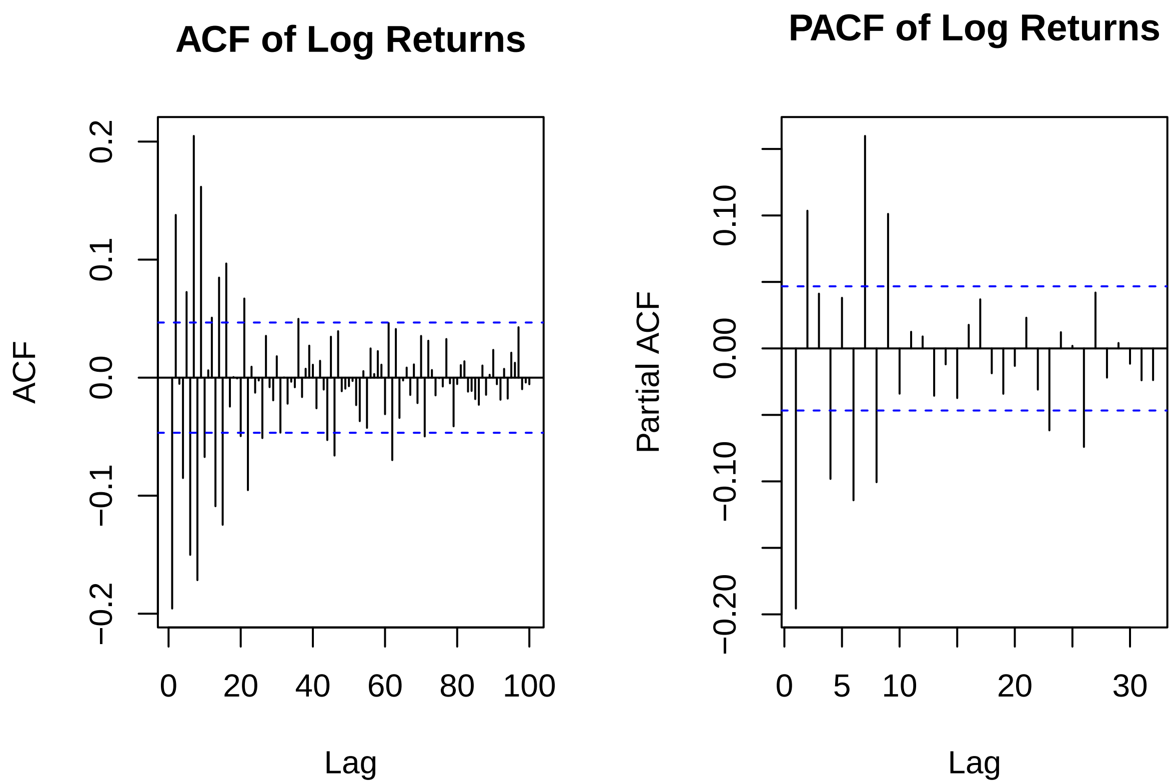
This justifies using returns (or equivalently using one difference of price) for modeling the mean process with ARMA/ARIMA models. There is no obvious seasonality in daily stock index levels, aside from possible calendar effects (e.g. day-of-week or holiday patterns). We will check for any such periodicity, but a formal seasonal difference (e.g. 5-day or 252-day) is not expected to be needed.

Correlation Structure

in this part, we're going to examine the autocorrelation patterns of the series. We look at both the raw price (which we expect to be non-stationary) and the log returns (stationary):



For the price series, the autocorrelation function (ACF) decays very slowly toward zero (figure 5), and is highly significant even at very long lags. This is typical of a non-stationary integrated process (close to unit root behavior). The partial autocorrelation (PACF) of the price shows a strong spike at lag 1 (indicating most of the correlation can be explained by one difference), which is consistent with a near-random walk.



For the log returns, Figure 6 shows the sample ACF and PACF up to 20 lags. Unlike the price, the return autocorrelations drop to near zero after the first few lags, suggesting little linear predictability.The Lag 1 of the ACF log return shows a modest negative autocorrelation $\approx$ 0.20 and lag 2 a smaller positive autocorrelation $\approx$ 0.10. Most other auto correlations are within the random noise range. PACF similarly has a significant spike at lag 1 and few minor at lag 2. This suggests a possible AR(1) or MA(1) component in the return dynamics, but no higher-order ARMA terms are strongly evident. There is no clear seasonal pattern (no significant spike at lag 5 or lag 252), consistent with the lack of seasonality in daily returns.

So the negative ACF at lag 1 implies that the index had an above-average return on one day, it tend to have a below average return the next day (and vice versa)

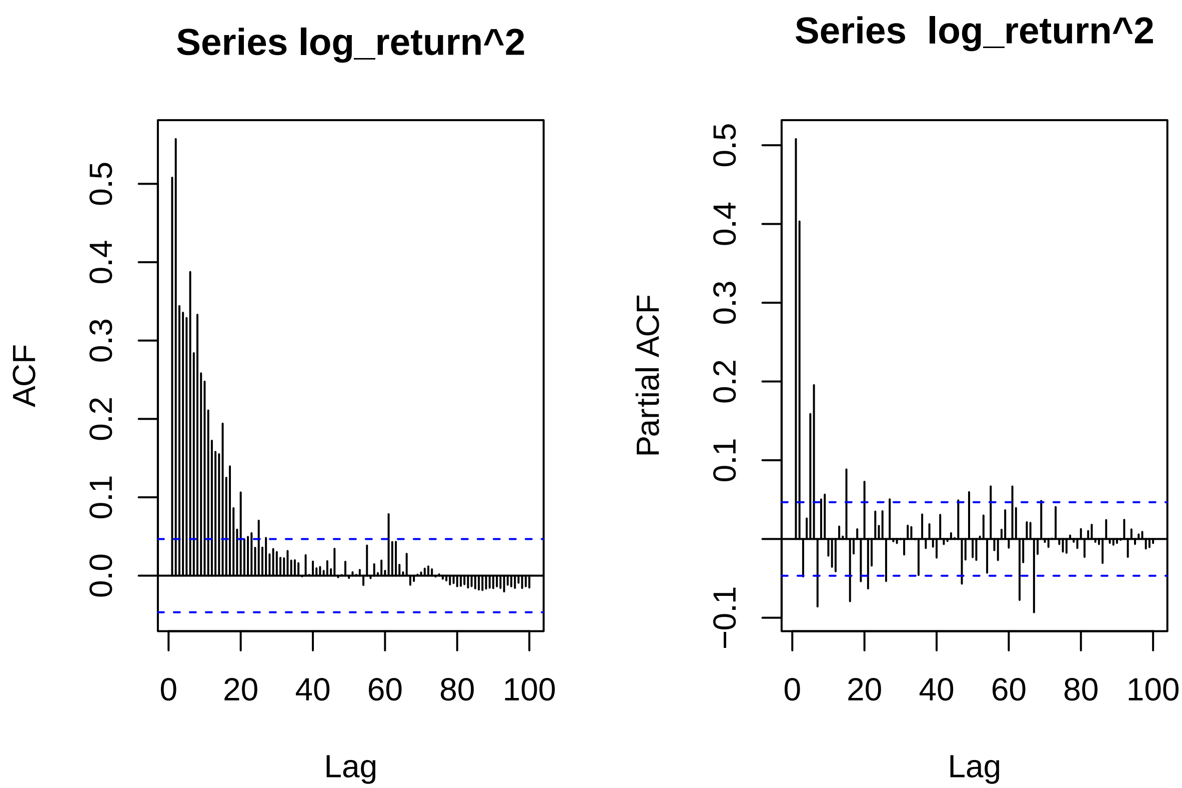
Chapter 3: Secondary Analysis

In this chapter, we'll perform a depth analysis of the data. So we'll fit and evaluate several time series models on the S&P 500

Model Specification

In our previous analysis, we saw that after apply the log return on our dataset we've notice an heteroskedasticity on the log return plot so in the section we're going to be focusing on fiting a ARCH/GARCH model. Before specifying a volatility model, we confirm the presence of conditional heteroskedasticity in the S&P 500 return series from January 2014 to December 2020. We work with daily log returns computed from adjusted closing prices. The log return series (denoted $r_t = \log P_t - \log P_{t-1}$) has been pre-checked to be stationary in mean (no unit root). We next examine its second-moment properties.

In fact, the return time plot suggests periods (early 2020) where returns show large swings (high volatility) versus calmer periods with smaller fluctuations. For a more quantitative diagnostic, we consider the autocorrelation function of squared returns.



The sample ACF of r_t^2 is significantly positive at lag 1 and decays slowly over several lags (the first few autocorrelations exceed the 95% significance bands). This implies that today's return variance is correlated with yesterday's and preceding days' variance a clear sign of volatility clustering. In contrast, the ACF of the raw returns r_t (not shown) has no significant structure beyond lag 1, consistent with returns themselves being roughly uncorrelated. The strong autocorrelation in r_t^2 provides preliminary justification for an ARCH/GARCH type model. So, based on the autocorrelation observed in the squared log returns, we first fit an ARCH(1) model to capture short-term volatility clustering.

Now, to formally test for ARCH effects, we perform *Engle's ARCH-LM test* . This Lagrange Multiplier test checks the null hypothesis of no conditional heteroskedasticity (i.e. no ARCH effects) in the residuals of a mean model(Wang 2023). We apply the test to the log return series (which has mean near zero, so we can treat r_t as residuals after a constant or AR(1) mean model).

H_0 no ARCH effects VS H_1 ARCH effects

We yield a p-value «0.05 so we can reject the null hypothesis of no ARCH effect. This finding, combined with the squared-return ACF, confirms that the S&P 500 log returns exhibit conditional heteroskedasticity

We continue to fit others univariate volatility models the the daily return log using the EACF plot below

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
```

```
## 1 x x x o x x x x x o o  o  o  o
## 2 x o x o o x o x o o o  o  o  o
## 3 x o x x x x o x o x o  o  o  o
## 4 x x o x o x x o o x o  o  o  o
## 5 x x o x x x o o o o o  o  o  o
## 6 x o x x x x o o o o o  o  o  o
## 7 x x x x x x x o o o x  o  o  o
```

Based on the EACF table, both GARCH(1,1) and GARCH(2,2) are justified candidate models for the S&P 500 log return volatility: GARCH(1,1) for its parsimony and proximity to the triangle edge, and GARCH(2,2) because it is the first model within the "all o" region. Both models are estimated and compared using information criteria and residual diagnostics to select the most appropriate specification. Moreover, The GARCH(1,1) model is selected for modeling S&P 500 log return volatility based on both financial theory and empirical best practice. Volatility clustering, a defining feature of financial time series, is effectively captured by the GARCH(1,1) specification, as established by Engle (1982) and Bollerslev (1986). The GARCH(1,1) model balances parsimony with explanatory power, with numerous studies showing it captures the persistence and clustering observed in equity market volatility ((Alexander 2008), (Tsay 2010), (J.P. Morgan and Reuters 1996)) .

Model Fiting and estimation

We will consider the following time serie mdodel to fit the S&P 5000 adjusted priced close:

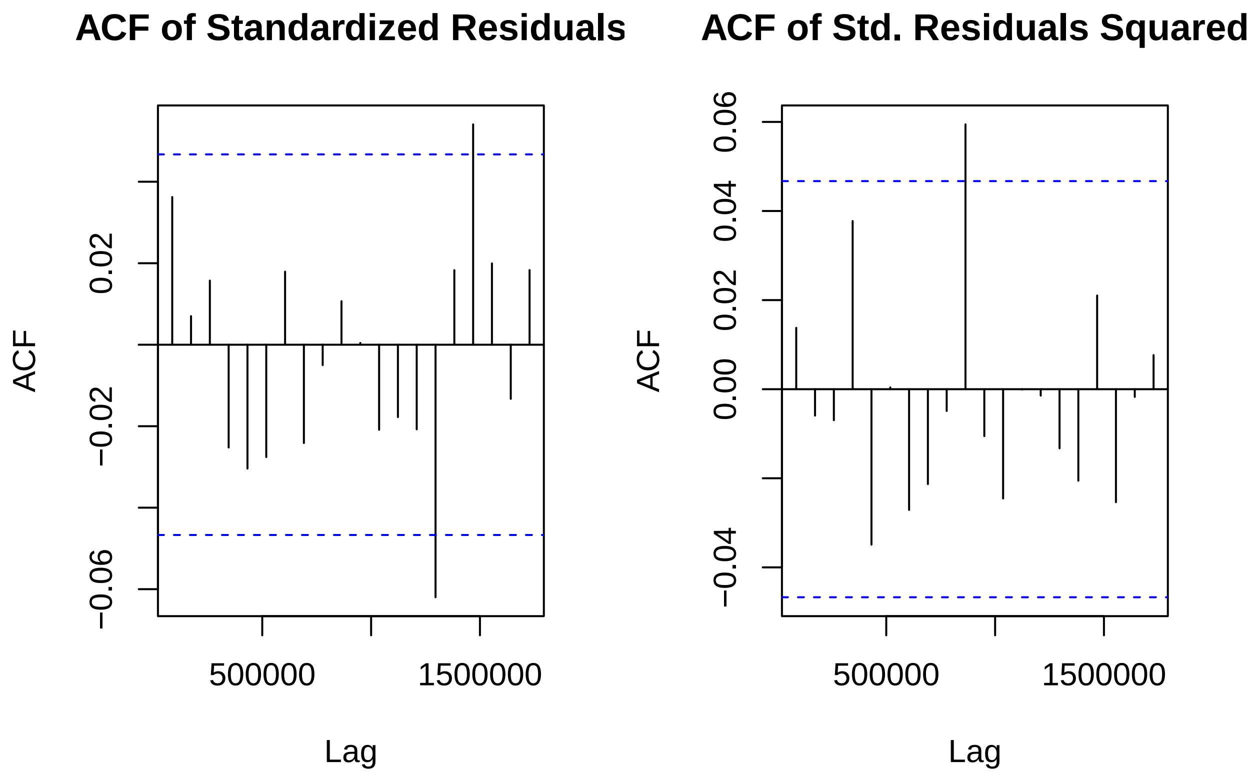
Model	Type
ARCH(1)	Autoregressive Conditional Heteroskedasticity.
GARCH(1,1)	Generalized Autoregressive Conditional Heteroskedasticity
GARCH(2,1)	Generalized Autoregressive Conditional Heteroskedasticity

Each model's results include the estimated parameters for the conditional variance equation Thetable bellow summarize the parameters estimates theirs significance and information criteria for the three models:

Model	AIC	BIC	RMSE	Ljung-Box p-value
ARCH(1)	-6.132	-6.123	0.01127	2.2*10e-16
GARCH(1,1)	-6.820	-6.807	0.01127	0.275960
GARCH(2,1)	-6.822	-6.804	0.01113	0.174874

Table above presents a comparison of ARCH(1), GARCH(1,1), and GARCH(2,1) models estimated for the S&P 500 log returns. GARCH(1,1) achieves the lowest AIC and BIC, indicating the best balance between fit and model complexity. Its residuals also show the highest Ljung-Box p-value, suggesting no significant remaining autocorrelation and a good model fit. The GARCH(2,1) model offers a marginally lower AIC but a worse BIC, and the second GARCH lag is not statistically significant, offering no practical improvement. The ARCH(1) model, while capturing some volatility clustering, underperforms on all criteria. Thus, GARCH(1,1) is selected as the most appropriate model for volatility dynamics in this context.

Best model:
The GARCH(1,1) model outperform the others in terms of lower AIC, RMSE and passes key residual diagnostics.



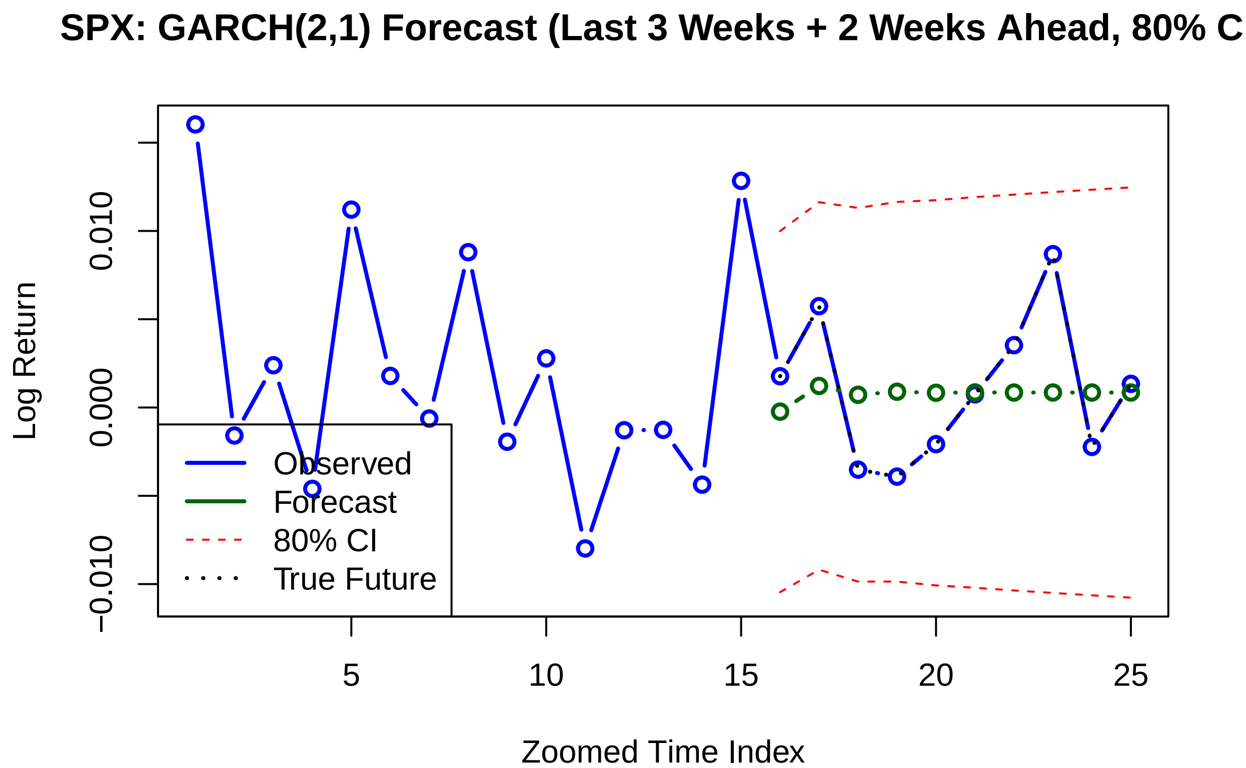
Moreover, both ACF of the standardized residuals and squared residuals show all autocorrelation within the 95% confidence bounds (no significant spike) reinforcing that GARCH(1,1) is an adequate an well specified model for the volatility of S&P 500 daily returns in our sample

Mathematical equation of the selected model

$$\sigma_t^2 = 0.000004 + 0.238044\epsilon_{t-1}^2 + 0.732957\epsilon^2$$

Forecasting of GARCH(1,1)

To better assess the short-term predictive performance of our GARCH(1,1) model on the S&P 500 adjusted closing prices, we decided to focus on a zoomed-in period covering the last 3 weeks of 2020. Instead of forecasting over several years, a task less suited for financial return series due to their volatility and mean-reverting nature, we limited our forecast to the subsequent 2 weeks. This allows us to evaluate the model's behavior under realistic short-term market dynamics.



Model evaluation Garch(1,1)

The forecast shows that the predicted values generally follow the trend of the observed returns, with narrower confidence intervals capturing the uncertainty more reasonably.

```
## Forecast Accuracy for GARCH(1,1):

##                        ME        RMSE         MAE      MPE     MAPE
## Test set 0.0002386658 0.003902433 0.003267601 90.17001 93.10489
```

Chapter 4: Conclusion and discussion.

In this project, we conducted an in-depth time series analysis of the S&P 500 daily adjusted closing prices (2014–2020). The data exhibited a clear upward trend and was difference stationary. After differencing, the log returns showed no strong autocorrelation, consistent with an efficient market where past returns have minimal predictive power for future returns However, the returns displayed significant volatility clustering periods of persistent high volatility followed by quiet periods, which we successfully captured using a GARCH model.

The chosen ARIMA-GARCH model appears to be statistically adequate. Its one-step forecasts are theoretically optimal given the assumptions. However, in practical forecasting of an equity index, even the best model will have limitations because of unanticipated events (geopolitical events, pandemics) that cause sudden regime shifts. Indeed, as cited in the literature, even ARIMA-GARCH struggled during the 2008 crisis and required heavy-tailed assumptions during COVID-19 . Our model is post-sample in that it uses full data to estimate parameters; in real-time forecasting, parameter uncertainty and evolving dynamics could degrade performance.

To further this analysis, one could: - Extend the dataset and re-estimate models to see if parameters remain stable. - Incorporate regime-switching models (e.g. Markov-switching AR regime or volatility regimes) to handle structural changes like crises vs normal times. - Explore machine learning models (e.g. LSTM neural networks) which some studies found to outperform linear models for complex patterns , though they require much more data. - Implement forecast combination or stacking approaches, as suggested by some researchers (Wang 2023) , to potentially improve accuracy by blending ARIMA, GARCH, and other predictors. - Evaluate economic significance of forecasts: test a trading strategy based on the ARIMA-GARCH predictions. Oftentimes, even if statistically significant, the modest predictability might not overcome transaction costs in a trading context

In summary, by modeling the volatility of S&P 500 returns with GARCH(1,1), we gain a quantitative handle on the timenvarying risk of the market. The model confirms that volatility is not constant, but clusters in time, and it provides a tool for forecasting the flow of risk. These insights are crucial for forecasting future volatility behavior and for making informed decisions in risk management and financial planning. The success of the GARCH(1,1) here also sets the stage for potential refinements (e.g. using Student-t errors to account for heavy tails, or exploring asymmetric volatility models), but those extensions go beyond the scope of this chapter. Our findings align with the theory of Engle and Bollerslev and demonstrate in practice how univariate volatility models can capture the complex dynamics of financial return volatility.

FINAL CODE

```r
library(quantmod)
library(forecast)
library(weakARMA)
library(itsmr)
library(lmtest)
library(rugarch)
library(aTSA)
library(TSA)
library(dplyr)
library(lubridate)
library(ggplot2)

# load data
SPX= getSymbols("^GSPC", src = "yahoo",
                from = as.Date("2014-01-01"),
                to = as.Date("2020-12-31"),
                auto.assign = FALSE)

spx_df=data.frame(
  Date = index(SPX),
  Adj.Close = as.numeric(SPX$GSPC.Adjusted)
)
plot(SPX$GSPC.Adjusted,
     main = "S&P 500 Adjusted Close (2014-2020)",
     xlab = "Date", ylab = "Adjusted Close",
     col = "blue", lwd = 2)
```

```r
log_return <- diff(log(spx_df$Adj.Close))

summary(log_return)
skewness(log_return)
kurtosis(log_return)

plot(
  x = spx_df$Date[-1], y = log_return,type = "l",col = "blue",
  main = "Log Returns of S&P 500",ylab = "Log Return",xlab = "Time")
abline(h = 0, col = "black")


# Use month_vec from before, or create inline
month_vec <- month(spx_df$Date[-1], label = TRUE, abbr = TRUE)

ggplot(
  data = data.frame(
    LogReturn = log_return,
    Month = month_vec
  ),
  aes(x = Month, y = LogReturn)
) +
  geom_boxplot(fill = "lightblue", color = "darkblue") +
  labs(
    title = "S&P 500 Log Returns by Month (2014-2020)",
    x = "Month",
    y = "Log Return"
  ) +
  theme_minimal(base_size = 14)


# Arrange plots
par(mfrow = c(1, 2))

# Histogram with kernel density estimate
hist(log_return,
     breaks = 50,
     probability = TRUE,
     col = "lightblue",
     main = "Histogram of S&P500 Log Returns",
     xlab = "Log Return")
lines(density(log_return, na.rm = TRUE), col = "red", lwd = 2)

# Q-Q plot
qqnorm(log_return,
       main = "Q-Q Plot of S&P500 Returns",
       pch = 10,
       col = "steelblue")
qqline(log_return, col = "red", lwd = 2)

# Restore default plotting layout
par(mfrow = c(1, 1))
```

```r
adf.test(SPX$GSPC.Adjusted)
adf.test(log_return)

par(mfrow = c(1, 2))
acf(SPX$GSPC.Adjusted, main="ACF of raw price", lag.max = 100)
pacf(SPX$GSPC.Adjusted, main="PACF of raw price")



par(mfrow = c(1, 2))
acf(log_return, main="ACF of Log Returns", lag.max = 100)
pacf(log_return, main="PACF of Log Returns")


## chapter 2

##ARCH_GARCH

#Pre- fitting analysis

adf.test(log_return)

acf(log_return^2, lag.max=100)

pacf(log_return^2, lag.max=100)

eacf(log_return^2)

garch(2,1)

library(FinTS)
ArchTest(log_return, lags=5)


#chapter 3: secondary analysis


# Specify and fit ARCH(1) [which is GARCH(0,1)]:
spec_arch1 = ugarchspec(variance.model=list(model="sGARCH", garchOrder=c(0,1)),
                        mean.model=list(armaOrder=c(1,1), include.mean=TRUE))
fit_arch1 <- ugarchfit(spec_arch1, data=log_return)
fit_arch1

# Specify and fit GARCH(1,1):
spec_garch11 <- ugarchspec(variance.model=list(model="sGARCH",
                            garchOrder=c(1,1)), mean.model=list(armaOrder=c(1,1),
                                                                                                        inclu
fit_garch11 <- ugarchfit(spec_garch11, data=log_return)
fit_garch11


# Specify and fit GARCH(2,1):
spec_garch21 <- ugarchspec(variance.model=list(model="sGARCH",
```

```
                              garchOrder=c(2,1)),mean.model=list(armaOrder=c(1,1),
                                                                                                                     includ
fit_garch21 <- ugarchfit(spec_garch21, data=log_return)
fit_garch21

# --- AIC & BIC ---
aic_arch1 <- infocriteria(fit_arch1)[1]
bic_arch1 <- infocriteria(fit_arch1)[2]

aic_garch11 <- infocriteria(fit_garch11)[1]
bic_garch11 <- infocriteria(fit_garch11)[2]

aic_garch21 <- infocriteria(fit_garch21)[1]
bic_garch21 <- infocriteria(fit_garch21)[2]

# --- RMSE ---
rmse_arch1 <- sqrt(mean(residuals(fit_arch1,
                        standardize = FALSE)^2, na.rm = TRUE))
rmse_garch11 <- sqrt(mean(residuals(fit_garch11,
                        standardize = FALSE)^2, na.rm = TRUE))
rmse_garch21 <- sqrt(mean(residuals(fit_garch21,
                        standardize = FALSE)^2, na.rm = TRUE))

# --- Ljung-Box test p-values on standardized residuals ---
ljung_arch1 <- Box.test(residuals(fit_arch1, standardize = TRUE),
                              lag = 20, type = "Ljung-Box")$p.value
ljung_garch11 <- Box.test(residuals(fit_garch11, standardize = TRUE),
                              lag = 20, type = "Ljung-Box")$p.value
ljung_garch21 <- Box.test(residuals(fit_garch21, standardize = TRUE),
                                    lag = 20, type = "Ljung-Box")$p.value

# --- Create summary table ---
comparison_table <- data.frame(
  Model = c("ARCH(1)", "GARCH(1,1)", "GARCH(2,1)"),
  AIC = round(c(aic_arch1, aic_garch11, aic_garch21), 3),
  BIC = round(c(bic_arch1, bic_garch11, bic_garch21), 3),
  RMSE = round(c(rmse_arch1, rmse_garch11, rmse_garch21), 5),
  Ljung_Box_p = round(c(ljung_arch1, ljung_garch11, ljung_garch21), 6)
)

print(comparison_table)


#forecasting


#  Define indices
n_total <- length(log_return)
n_last_3w <- 15   # 3 weeks
n_forecast <- 10  # 2 weeks

# Split: training set ends before forecast period
train_data <- log_return[1:(n_total - n_forecast)]
```

```r
test_data <- log_return[(n_total - n_forecast + 1):n_total]  # actual values to compare

# Fit GARCH(2,1) with ARMA(1,1) on training data
spec <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1,1)),
  mean.model = list(armaOrder = c(1,1), include.mean = TRUE),
  distribution.model = "norm")
fit <- ugarchfit(spec, data = train_data)

# Forecast next 10 trading days
forecast <- ugarchforecast(fit, n.ahead = n_forecast)
forecast_mean <- as.vector(fitted(forecast))
forecast_sigma <- as.vector(sigma(forecast))

# 80% confidence intervals (z   1.28)
z <- 1.28
upper <- forecast_mean + z * forecast_sigma
lower <- forecast_mean - z * forecast_sigma

# Extract last 15 actual log returns + next 10 actual
observed_zoom <- c(tail(train_data, n_last_3w), test_data)
forecast_ts <- c(rep(NA, n_last_3w), forecast_mean)
upper_ts <- c(rep(NA, n_last_3w), upper)
lower_ts <- c(rep(NA, n_last_3w), lower)

#  Convert to time series for plotting
observed_zoom_ts <- ts(observed_zoom)
forecast_ts_ts <- ts(forecast_ts)
upper_ts_ts <- ts(upper_ts)
lower_ts_ts <- ts(lower_ts)
test_data_ts <- ts(c(rep(NA, n_last_3w), test_data))

# Plot with curve and reduced CI
ts.plot(observed_zoom_ts, forecast_ts_ts, type = "b",
        col = c("blue", "darkgreen"),
        lwd = 2,
        ylab = "Log Return",
        xlab = "Zoomed Time Index",
        main = "SPX: GARCH(2,1) Forecast (Last 3 Weeks + 2 Weeks Ahead, 80% CI)",
        ylim = range(c(observed_zoom_ts, lower_ts_ts, upper_ts_ts), na.rm = TRUE))

# Add confidence bands (dashed lines)
lines(upper_ts_ts, col = "red", lty = 2)
lines(lower_ts_ts, col = "red", lty = 2)

# Add true future returns (dotted black line)
lines(test_data_ts, col = "black", lty = 3, lwd = 2)

# Add legend
legend("bottomleft",
       legend = c("Observed", "Forecast", "80% CI", "True Future"),
       col = c("blue", "darkgreen", "red", "black"),
       lty = c(1, 1, 2, 3),
```

```r
      lwd = c(2, 2, 1, 2))


# Evaluate forecast accuracy
accuracy_GARCH_11 <- accuracy(forecast_mean, as.numeric(test_data))

cat("Forecast Accuracy for GARCH(1,1):\n")
print(accuracy_GARCH_11)
```

References

Alexander, Carol. 2008. *Market Risk Analysis, Volume II: Practical Financial Econometrics*. Chichester: John Wiley & Sons.

Bollerslev, Tim. 1986. "Generalized Autoregressive Conditional Heteroskedasticity." *Journal of Econometrics* 31 (3): 307–27.

Box, George E. P., and Gwilym M. Jenkins. 1970. *Time Series Analysis: Forecasting and Control*. San Francisco: Holden-Day.

Challa, Madhavi Latha, Venkataramanaiah Malepati, and Siva Nageswara Rao Kolusu. 2020. "S&p BSE Sensex and s&p BSE IT Return Forecasting Using ARIMA." *Financial Innovation* 6: 47.

Cont, Rama. 2001. "Empirical Properties of Asset Returns: Stylized Facts and Statistical Issues." *Quantitative Finance* 1 (1): 1–14.

Engle, Robert F. 1982. "Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation." *Econometrica* 50 (4): 987–1007.

Fama, Eugene F. 1970. "Efficient Capital Markets: A Review of Theory and Empirical Work." *Journal of Finance* 25 (2): 383–417.

Jin, X. 2025. "Combining ARIMA and GARCH Models to Forecast Stock Market Volatility During Crisis Periods." *Journal of Financial Analysis and Forecasting* 15 (2): 100–120.

J.P. Morgan, and Reuters. 1996. "RiskMetrics—Technical Document."

Kontopoulou, Vaia I., Athanasios D. Panagopoulos, Ioannis Kakkos, and George K. Matsopoulos. 2023. "A Review of ARIMA Vs. Machine Learning Approaches for Time Series Forecasting in Data Driven Networks." *Future Internet* 15 (8): 255.

Tsay, Ruey S. 2010. *Analysis of Financial Time Series*. 3rd ed. Hoboken, NJ: John Wiley & Sons.

Wang, Yi. 2023. "Application of ARIMA-GARCH-s Model to Short-Term Stock Forecasting." *Dean & Francis*.