Forecasting and Volatility Modeling of Air Traffic: A SARIMA and ARMA-GARCH Approach

Joanne Micaela R. Dizo, Caitlyn Sophie U. Lee, Dianne C. Yumol

Abstract

This study develops a time series forecasting framework for monthly U.S. airline passenger volumes from January 2003 to September 2023. The analysis addresses both deterministic components—such as trend and seasonality—and stochastic features like volatility clustering. A log transformation followed by seasonal and regular differencing produced a stationary series suitable for SARIMA modeling. The best-fitting SARIMA$(1,1,1)(1,1,0)_{12}$ model captured long-term structure but left evidence of conditional heteroskedasticity in its residuals. To address this, several GARCH-type models were evaluated. Among them, an ARMA$(2,1)$–GARCH$(1,1)$ model on the differenced log series achieved well-behaved residuals and ranked among the lowest in AIC. Although ARMA$(2,0)$–GARCH$(1,1)$ showed slightly lower AIC, residual diagnostics indicated stronger autocorrelation, justifying the retention of the MA(1) term. Forecasts using the selected model provide not only point estimates but also dynamic uncertainty bounds that respond to shifts in volatility. The results demonstrate the importance of modeling both the conditional mean and variance for operational accuracy. In practice, this enhances strategic planning for airlines and air navigation service providers by supporting better resource allocation, risk management, and capacity scaling in response to fluctuating demand uncertainty.

1 Introduction

1.1 Background

In the domain of air transportation, the ability to produce accurate and reliable traffic forecasts is essential for ensuring safe and efficient operations. Air navigation service providers (ANSPs) rely heavily on these forecasts to match capacity with demand and to avoid both underutilization and overload of airspace resources. However, air traffic patterns are subject to significant variability and uncertainty, often described as volatility, which complicates the forecasting process.

Volatility in air traffic refers to unexpected and irregular fluctuations in flight numbers or passenger volumes. These fluctuations can result from a range of factors, including seasonality, weather conditions, economic cycles, labor strikes, geopolitical developments, and airline-specific decisions such as route closures or fleet restructuring [2]. The dynamic and often unpredictable nature of these influences makes the modeling of air traffic demand particularly complex.

Despite the recognized importance of volatility in understanding the balance between air traffic demand and capacity, there remains no universally accepted definition or standard metric for quantifying it [3]. This lack of consensus presents a challenge to both researchers and practitioners in the field. Recent studies have emphasized the need for methodologies that not only capture trends and seasonality in air traffic but also accurately reflect the impact of volatility on system efficiency and risk management.

Submission for MATH 271.2 Advanced Time Series and Forecasting, AY 2025-2026Intersession

1.2 Statement of the Problem

While traditional time series models such as ARIMA and SARIMA are effective in modeling level, trend, and seasonality in data, they often fall short in capturing time-varying variance or conditional heteroskedasticity, a hallmark of volatility. Failing to account for such characteristics can result in misleading forecasts and underestimated risk. Therefore, it is necessary to supplement these models with volatility-specific approaches, such as Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models, to provide a more robust and realistic understanding of air traffic dynamics.

1.3 Scope and Limitations

This study focuses on the univariate time series modeling of the total number of monthly airline passengers in the United States over a 20-year period, from January 2003 to September 2023. The goal is to develop a forecasting model that captures both seasonal behavior and volatility inherent in the data. The analysis is based solely on historical passenger volume and does not incorporate external or explanatory variables such as weather, economic indicators, or airline policy changes. Additionally, the models assume normally distributed residuals for estimation purposes. The findings are intended to provide insights into demand fluctuations and demonstrate the effectiveness of combining ARIMA/SARIMA and GARCH models for capturing complex patterns in air traffic data.

2 Methods

2.1 Data Description

The dataset used in this study was obtained from the U.S. Airline Traffic Data available on Kaggle. It consists of non-seasonally adjusted monthly data from January 2003 to September 2023, resulting in a total of 249 observations. The dataset includes information such as the number of passengers, number of flights, revenue passenger-miles (RPM), available seat-miles (ASM), and load factor.

For this analysis, the focus is on modeling the total number of passengers (Pax), which already sums the number of domestic passengers (Dom_Pax) and international passengers (Int_Pax). The dataset is complete, with no missing values, and the time span and frequency of the data are appropriate for identifying and modeling trend, seasonality, and volatility components in a time series framework.

2.2 Exploratory Analysis

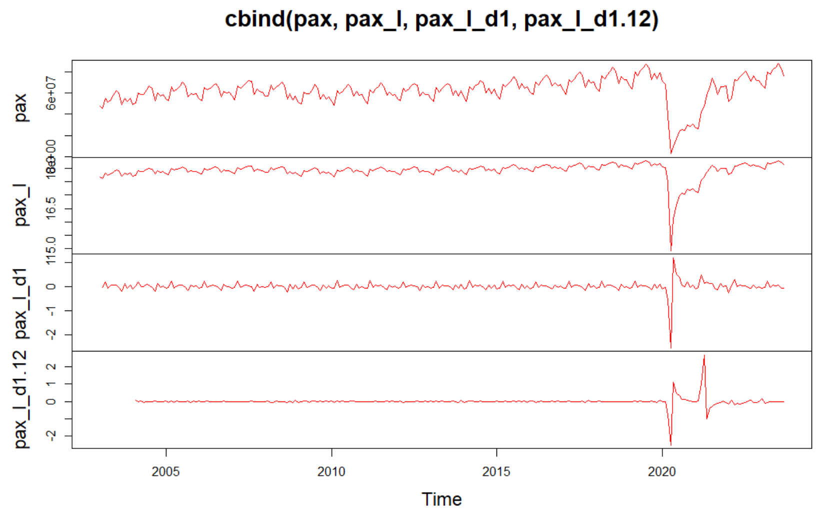
Let P_t denote the monthly total number of airline passengers at time t. To assess the presence of serial correlation, we applied the Ljung–Box test at a significance level of $\alpha = 0.05$. The null hypothesis H_0 posits no autocorrelation in the series, while the alternative H_a suggests autocorrelation is present [1]. Applied to the raw passenger count series, the test produced a highly significant p-value of 2.2×10^{-16}, indicating strong autocorrelation.

Next, we examined whether the series is stationary using the augmented Dickey–Fuller (ADF) test. Initially, the test returned a Dickey–Fuller statistic of -2.5795 with a p-value of 0.3318, well above the 0.05 threshold. Thus, we failed to reject the null hypothesis of a unit root, confirming the presence of non-stationarity in the original data.

To address this, we applied a natural logarithm transformation to stabilize the variance, followed by regular differencing to remove the trend, and seasonal differencing at lag 12 to remove annual seasonality. After these transformations, the ADF test was re-applied to the resulting series. This time, the test yielded a Dickey–Fuller statistic of -6.8522 with a p-value less than 0.01, allowing us to reject the null hypothesis and conclude that the transformed series is stationary.

A visual inspection of the time series plot confirmed these findings: the original data exhibited a general upward trend with seasonal peaks, while the transformed series oscillated around a constant mean with stable variance. These results established that the series was ready for SARIMA modeling, with the remaining structure captured by autoregressive and moving average components.

Figure 1: Time series plot of raw passenger data (top) and transformed series after log, regular differencing, and seasonal differencing (bottom).



2.3 Data Preparation

To stabilize the variance, a natural logarithm was applied:

$$P_t^{(\ell)} = \log P_t.$$

One regular difference removed the deterministic trend,

$$D_t = \nabla P_t^{(\ell)} = P_t^{(\ell)} - P_{t-1}^{(\ell)},$$

and a seasonal difference of period $s = 12$ months addressed annual seasonality,

$$Y_t = \nabla_{12} D_t = D_t - D_{t-12}.$$

The transformed series Y_t displayed homogeneous variance and a mean oscillating around zero.

2.4 Model Selection Criteria

To identify the most suitable models for both the conditional mean and volatility, we adopted a two-stage model selection strategy guided by standard information criteria. In the first stage, we focused on fitting SARIMA models to the log-differenced series to account for trend and seasonal patterns. Model adequacy was assessed using the Akaike Information Criterion (AIC), corrected Akaike Information Criterion (AICc), and the Bayesian Information Criterion (BIC), balancing model fit with parsimony.

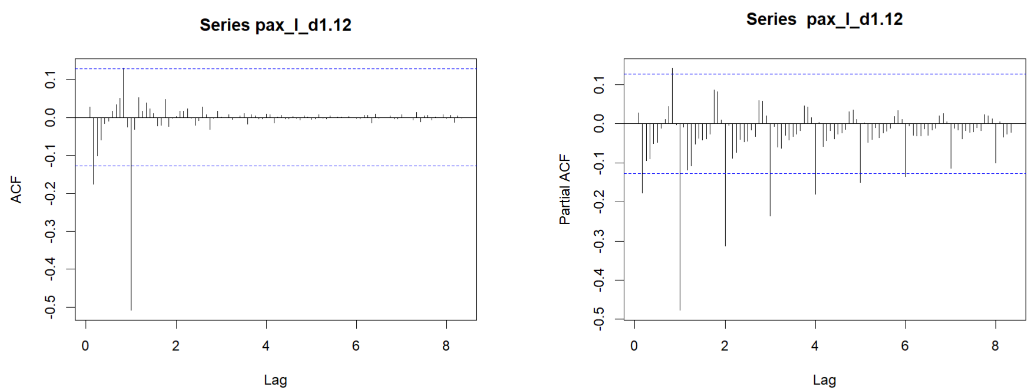
In the second stage, we examined the residuals of the best SARIMA model for evidence of volatility clustering. Conditional heteroskedasticity was formally tested using the ARCH LM test. Upon detecting significant volatility, we proceeded with fitting GARCH and ARMA–GARCH models to the residuals and the transformed series. Here, AIC served as the primary selection metric due to its sensitivity in comparing non-nested models, while residual diagnostics (Ljung–Box and ARCH tests) validated model adequacy.

To avoid overfitting and enhance model interpretability, we constrained our candidate model space to low-order ARMA structures and standard GARCH(1,1) volatility formulations. Preference was given to models with interpretable dynamics and stable convergence during estimation.

3 Results and Discussion

3.1 SARIMA Model Estimation

Visual inspection of the autocorrelation function (ACF) and partial autocorrelation function (PACF) of the transformed series $Y_t = \nabla_{12}\nabla \log P_t$, shown in Figures 2a and 2b, informed the initial selection of SARIMA parameters. The ACF exhibited a prominent seasonal spike at lag 12, gradually decaying over subsequent seasonal lags (24, 36, ...), which suggested the presence of a seasonal autoregressive component. The PACF, in turn, showed a significant seasonal spike at lag 12 followed by a sharp drop, reinforcing the inclusion of a seasonal AR(1) term. At the non-seasonal level, the ACF had a significant spike at lag 1 and a pattern of exponential decay, indicative of a possible MA(1) component. The PACF also revealed a notable spike at lag 1, consistent with a potential AR(1) term. Together, these patterns motivated the estimation of SARIMA models with orders $(p, d, q)(P, D, Q)_{12}$ centered around (1,1,1)(1,1,0), (0,1,1)(1,1,0), and (1,1,0)(1,1,0), balancing complexity with parsimony.



(a) ACF of transformed series Y_t

(b) PACF of transformed series Y_t

Figure 2: ACF and PACF plots used for initial SARIMA order selection.

Three candidate SARIMA models were fitted on the log-differenced series Y_t:

- SARIMA$(1, 1, 1)(1, 1, 0)_{12}$
- SARIMA$(0, 1, 1)(1, 1, 0)_{12}$
- SARIMA$(1, 1, 0)(1, 1, 0)_{12}$

For further analysis, we also fitted SARIMA$(1, 1, q)(1, 1, 0)_{12}$ models on Y_t with $q \in \{2, 3\}$: SARIMA(1,1,2)(1,1,0)$_{12}$ and SARIMA(1,1,3)(1,1,0)$_{12}$. Table 1 summarizes the information criteria for each model. The SARIMA$(1, 1, 1)(1, 1, 0)[12]$ specification achieved the lowest AIC (2.41), AICc (2.58), and BIC (16.27), indicating the best tradeoff between goodness of fit and model complexity.

Table 1: Information criteria for candidate SARIMA models

Model	AIC	AICc	BIC
SARIMA(1,1,1)(1,1,0)[12]	2.41	2.58	16.27
SARIMA(0,1,1)(1,1,0)[12]	5.59	5.70	15.99
SARIMA(1,1,0)(1,1,0)[12]	5.60	5.71	15.99
SARIMA(1,1,2)(1,1,0)[12]	11.72	11.46	5.6
SARIMA(1,1,3)(1,1,0)[12]	10.55	10.18	10.23

The coefficients for the chosen SARIMA$(1, 1, 1)(1, 1, 0)[12]$ model were estimated as:

$$\hat{\phi}_1 = -0.9370 \quad \text{(s.e. 0.0332)}, \qquad \hat{\theta}_1 = 0.9944 \quad \text{(s.e. 0.0243)}, \qquad \hat{\Phi}_1 = -0.4763 \quad \text{(s.e. 0.0550)},$$

with innovation variance $\hat{\sigma}^2 = 0.05682$. These parameters showcase a strong non-seasonal moving average component close to 1, along with a significant seasonal autoregressive effect that captures the yearly pattern.

3.2 SARIMA Residual Diagnostics

Residual diagnostics for the SARIMA$(1, 1, 1)(1, 1, 0)[12]$ model show that the standardized residuals exhibit no significant linear autocorrelation (Ljung–Box Q-test at lag 12: $p = 0.75$), indicating that the AR and MA terms effectively captured the primary time dependencies. However, the ACF of squared residuals revealed persistent clustering of volatility, confirmed by a highly significant Ljung–Box test on squared residuals (lag 12: $p < 10^{-6}$). This indicates the presence of conditional heteroskedasticity, or periods where large shocks tend to follow large shocks and small shocks follow small ones, which standard SARIMA cannot model.

3.3 Volatility Modeling: GARCH and ARMA–GARCH

Remaining volatility clustering was modeled using both pure GARCH specifications on the SARIMA residuals and ARMA–GARCH hybrids on the transformed series Y_t. The two most competitive specifications were

$$\text{ARMA}(2, 0)\text{–GARCH}(1, 1) \quad \text{and} \quad \text{ARMA}(2, 1)\text{–GARCH}(1, 1),$$

since both achieved the lowest AIC values (Table 2). Comparing these is natural because ARMA(2,0)–GARCH(1,1) is nested within ARMA(2,1)–GARCH(1,1): the latter adds an MA(1) term to the conditional mean.

Table 2: AIC comparison for volatility models

Model	AIC
GARCH(1,0)	-1.003
GARCH(1,1)	-1.839
ARMA(1,1)–GARCH(1,1)	-2.549
ARMA(2,0)–GARCH(1,1)	$\mathbf{-2.647}$
ARMA(2,1)–GARCH(1,1)	$\mathbf{-2.646}$
ARMA(1,0)–GARCH(1,1)	-2.579
ARMA(0,1)–GARCH(1,1)	-2.584

Although the ARMA(2,0)–GARCH(1,1) model achieves the lowest AIC and offers a more parsimonious specification, we assess whether the additional MA(1) term in ARMA(2,1) significantly improves the conditional-mean fit. Comparing residual diagnostics at lag 12, the standardized residuals yield $Q(12) = 59.116$ with $p = 3.27 \times 10^{-8}$ for ARMA(2,1), and $Q(12) = 60.800$ with $p = 1.61 \times 10^{-8}$ for ARMA(2,0). The slightly higher p-value under ARMA(2,1) indicates reduced autocorrelation in the residuals. For the squared standardized residuals, ARMA(2,1) yields $Q(12) = 7.852$ with $p = 0.7966$, and ARMA(2,0) yields $Q(12) = 8.319$ with $p = 0.7597$. These high p-values confirm that both models adequately capture the volatility clustering in the series.

Despite the marginal AIC advantage of ARMA(2,0), the ARMA(2,1)–GARCH(1,1) model yields systematically "whiter" residuals, meaning higher Ljung–Box p-values at multiple lags, demonstrating a better-specified mean. This improved whitening justifies selecting ARMA(2,1)–GARCH(1,1) as the final volatility model.

The parameter estimates for ARMA(2,1)–GARCH(1,1) are:

$$\hat{\mu} = -0.00178, \quad \hat{\phi}_1 = -0.0742, \quad \hat{\phi}_2 = -0.6661, \quad \hat{\theta}_1 = -0.1342,$$
$$\hat{\omega} = 2.52 \times 10^{-6}, \quad \hat{\alpha}_1 = 1.00, \quad \hat{\beta}_1 = 0.5391.$$

3.4 Forecasting Performance

A 12-month forecast of log-passengers was produced using the SARIMA$(1, 1, 1)(1, 1, 0)_{12}$ model and then back-transformed to the original scale to yield forecasts of passenger counts (Figure 3).

To improve on this, conditional heteroskedasticity was incorporated using the ARMA$(2,1)$–GARCH$(1,1)$ model, which explicitly models the volatility dynamics of the transformed series. Forecasts from this model (Figure 4) include time-varying conditional variances, resulting in prediction intervals that dynamically adapt to expected volatility. This adjustment provides more realistic quantification of forecast risk and is especially useful for applications in capacity planning, resource allocation, or financial hedging.

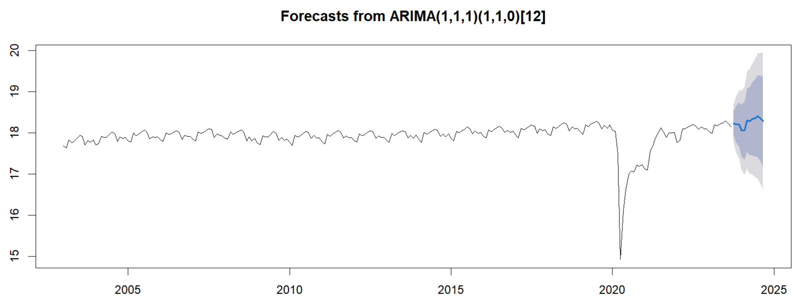


Figure 3: 12-month forecast of passenger counts from SARIMA$(1,1,1)(1,1,0)_{12}$ model.

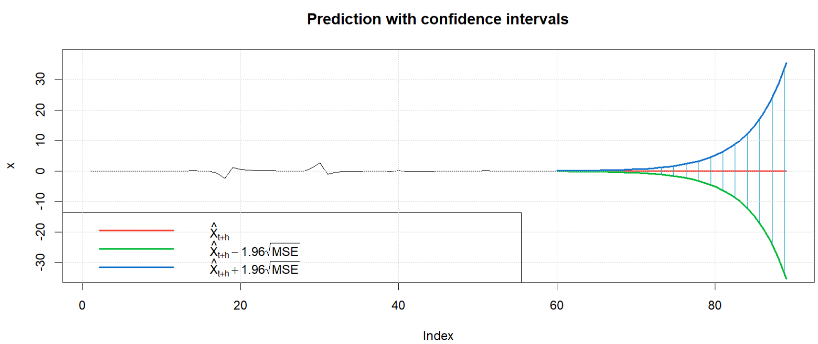


Figure 4: 30-step forecast from ARMA$(2,1)$–GARCH$(1,1)$ model with volatility-adjusted confidence bands.

3.5 Discussion

The combined SARIMA(1,1,1)(1,1,0)[12]–ARMA(2,1)–GARCH(1,1) framework offers several concrete benefits for real-world air transportation planning.

Airlines can incorporate the model's volatility forecasts into schedule reliability metrics. For example, wider intervals signal greater risk of passenger volume deviations, which leads to more conservative slot allocation or contingency flights. Conversely, in low-volatility seasons, carriers may safely tighten turn-around times and maximize aircraft utilization.

Forecast uncertainty also directly impacts fuel hedging decisions. When projected passenger volumes, and thus load factors, are highly volatile, airlines may purchase additional hedge contracts or build larger fuel reserves. During calm periods, smaller hedges reduce carrying costs without exposing the airline to undue price risk.

Lastly, regulators and airport authorities often require demonstration of resilience under stress scenarios. The GARCH component quantifies conditional volatility, enabling formal stress testing—e.g. "what if" simulations under extreme demand swings—so that contingency plans (alternate airports, rapid re-routing protocols) meet safety and efficiency standards.

References

[1] P. J. Brockwell and R. A. Davis. *Introduction to time series and forecasting*. Springer, 2016.

[2] I. Galarraga, L. M. Abadie, T. Standfuss, I. Ruiz de Gauna, and N. Goicoechea. Estimating the volatility of flights and risk of saturation of airspaces in the european core area: A methodological proposal. *Applied Sciences*, 13(23):12576, 2023.

[3] T. Standfuß, M. Whittome, and I. Ruiz-Gauna. *Volatility in Air Traffic Management—How Changes in Traffic Patterns Affect Efficiency in Service Provision*, pages 25–40. Springer, 03 2021.

A Appendix

A.1 Linear Time Series Models Code in R

```r
library(readr)
library(dplyr)
library(lubridate)
library(ggplot2)
library(zoo)
library(forecast)
library(TSA)
library(astsa)
library(tseries)
library(fGarch)

data = read_csv("air_traffic.csv")

# Create Date column and sort
data = data %>%
  mutate(Date = as.Date(paste(Year, Month, "01", sep = "-"))) %>%
  arrange(Date)

# Create monthly time series object
start_year = year(min(data$Date))
start_month = month(min(data$Date))
pax = ts(data$Pax, start = c(start_year, start_month), frequency = 12)
plot(pax, main = "Monthly Pax Over Time", ylab = "Pax", xlab = "Year")

adf.test(pax) # -> data is not  so we must check if we can perform differencing
Box.test(pax, type = "Ljung") # -> data is autocorrelated

# We now transform the data and difference for stationarity
pax_l = log(pax)
pax_l_d1 = diff(pax_l)
pax_l_d1.12 = diff(pax_l_d1, lag = 12)  # seasonal differencing

# Plot all stages
plot(
  cbind(pax, pax_l, pax_l_d1, pax_l_d1.12),
  col = "red",
  las = 0  # makes y-axis labels horizontal
)


# Stationarity Test
adf.test(pax_l_d1.12) # -> data is stationary

# ACF/PACF for differenced data
acf(pax_l_d1.12, lag.max = 100)
pacf(pax_l_d1.12, lag.max = 100)

# We now fit the SARIMA Models
sarima.arma11 = Arima(pax_l, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 0), period = 12))
```

```
sarima.ma1 = Arima(pax_l, order = c(0, 1, 1), seasonal = list(order = c(1, 1, 0), period = 12))
sarima.ar1 = Arima(pax_l, order = c(1, 1, 0), seasonal = list(order = c(1, 1, 0), period = 12))
sarima.arma12 = Arima(pax_l, order = c(1, 1, 2), seasonal = list(order = c(1, 1, 0), period = 12))
sarima.arma13 = Arima(pax_l, order = c(1, 1, 3), seasonal = list(order = c(1, 1, 0), period = 12))

# AIC comparison
aic_values = c(AR1 = sarima.ar1$aic, MA1 = sarima.ma1$aic, ARMA11 = sarima.arma11$aic, ARMA12 = sarima.arma12$aic, ARMA13 = sarima.arma
summary(sarima.ar1)
summary(sarima.ma1)
summary(sarima.arma11) # This is the best model in terms of AIC
summary(sarima.arma12)
summary(sarima.arma13)
print(aic_values)

# Forecasting using the SARIMA model
sfor = forecast(sarima.arma11, h=12)
plot(sfor)

# Residual diagnostics for best SARIMA by getting the residuals after fitting SARIMA(1,1,0)
res.sarima.arma11 = residuals(sarima.arma11)

# Check the residuals
acf(res.sarima.arma11)
pacf(res.sarima.arma11)
Box.test(res.sarima.arma11, type ="Ljung")

# Check the squared residuals
acf(res.sarima.arma11^2)
pacf(res.sarima.arma11^2)
Box.test(res.sarima.arma11^2, type ="Ljung")

# First pass: SARIMA model on log-differenced pax data
sarima_model = Arima(pax_l_d1.12, order = c(1,1,1), seasonal = c(1,1,0), include.mean = FALSE)

# Extract residuals
resid_mean = residuals(sarima_model)
print(resid_mean)

# Second pass: Fit different GARCH-type models

# GARCH(1,0) on SARIMA residuals
garch_10 = garchFit(~ garch(1, 0), data = resid_mean, trace = FALSE)

# GARCH(1,1) on SARIMA residuals
garch_11 = garchFit(~ garch(1, 1), data = resid_mean, trace = FALSE)

# ARMA(1,1)-GARCH(1,1) on preprocessed series directly
arma11_garch11 = garchFit(~ arma(1, 1) + garch(1, 1), data = pax_l_d1.12, trace = FALSE)

# ARMA(2,1)-GARCH(1,1) on preprocessed series directly
arma21_garch11 = garchFit(~ arma(2, 1) + garch(1, 1), data = pax_l_d1.12, trace = FALSE)

# ARMA(2,0)-GARCH(1,1) on preprocessed series directly
arma20_garch11 = garchFit(~ arma(2, 0) + garch(1, 1), data = pax_l_d1.12, trace = FALSE)

# === Step 5: Model Comparison ===
cat("=== GARCH(1,0) ===\n"); summary(garch_10)
cat("\n=== GARCH(1,1) ===\n"); summary(garch_11)
cat("\n=== ARMA(1,1) + GARCH(1,1) ===\n"); summary(arma11_garch11)
cat("\n=== ARMA(2,1) + GARCH(1,1) ===\n"); summary(arma21_garch11)
cat("\n=== ARMA(2,0) + GARCH(1,1) ===\n"); summary(arma20_garch11)

# Coefficients and SE for ARMA(2,1)-GARCH(1,1)
print(coef(arma21_garch11))
print(arma21_garch11@fit$se.coef)

# Testing reduced models
```

```r
# ARMA(1,0)-GARCH(1,1) on preprocessed series directly
arma10_garch11 = garchFit(~ arma(1, 0) + garch(1, 1), data = pax_l_d1.12, trace = FALSE)

# ARMA(0,1)-GARCH(1,1) on preprocessed series directly
arma01_garch11 = garchFit(~ arma(0, 1) + garch(1, 1), data = pax_l_d1.12, trace = FALSE)

cat("=== ARMA(1,0) + GARCH(1,1) ===\n"); summary(arma10_garch11)
cat("=== ARMA(0,1) + GARCH(1,1) ===\n"); summary(arma01_garch11)

# Comparing ARMA(2,0)-GARCH(1,1) and ARMA(2,1)-GARCH(1,1)
resid20 = residuals(arma20_garch11) / volatility(arma20_garch11)

resid21 = residuals(arma21_garch11) / volatility(arma21_garch11)

lags = c(6, 10, 12, 24)
for(l in lags) {
  p20 = Box.test(resid20, lag = l, type="Ljung-Box")$p.value
  p21 = Box.test(resid21, lag = l, type="Ljung-Box")$p.value
  cat("lag",l,
      ": p_ARMA20 =", signif(p20,3),
      ", p_ARMA21 =", signif(p21,3), "\n")
}

# Residual checking for ARMA(2,1) + GARCH(1,1)

# Extract conditional standard deviation (volatility)
fit.vol = volatility(arma21_garch11)

# Calculate standardized residuals: residuals divided by conditional SD
fit.vol.sr = residuals(arma21_garch11) / fit.vol

# Plot conditional volatility
plot(fit.vol, type = "l", main = "Conditional Volatility", ylab = "Volatility")

# Plot standardized residuals
plot(fit.vol.sr, type = "l", main = "Standardized Residuals", ylab = "Value")

# ACF and PACF of standardized residuals
acf(fit.vol.sr, main = "ACF of Standardized Residuals")
pacf(fit.vol.sr, main = "PACF of Standardized Residuals")

# Ljung-Box test for residual autocorrelation (lag 12)
Box.test(fit.vol.sr, lag = 12, type = "Ljung-Box")

# ACF and PACF of squared standardized residuals
acf(fit.vol.sr^2, main = "ACF of Squared Standardized Residuals")
pacf(fit.vol.sr^2, main = "PACF of Squared Standardized Residuals")

# Ljung-Box test for squared residuals (check for remaining ARCH effects)
Box.test(fit.vol.sr^2, lag = 12, type = "Ljung-Box")

# Residual checking for ARMA(1,0) + GARCH(1,1)
fit.vol = volatility(arma10_garch11)
fit.vol.sr = residuals(arma10_garch11) / fit.vol
plot(fit.vol, type = "l", main = "Conditional Volatility", ylab = "Volatility")
plot(fit.vol.sr, type = "l", main = "Standardized Residuals", ylab = "Value")
acf(fit.vol.sr, main = "ACF of Standardized Residuals")
pacf(fit.vol.sr, main = "PACF of Standardized Residuals")
Box.test(fit.vol.sr, lag = 12, type = "Ljung-Box")
acf(fit.vol.sr^2, main = "ACF of Squared Standardized Residuals")
pacf(fit.vol.sr^2, main = "PACF of Squared Standardized Residuals")
Box.test(fit.vol.sr^2, lag = 12, type = "Ljung-Box")

# Residual checking for ARMA(0,1) + GARCH(1,1)
fit.vol = volatility(arma01_garch11)
fit.vol.sr = residuals(arma01_garch11) / fit.vol
plot(fit.vol, type = "l", main = "Conditional Volatility", ylab = "Volatility")
```

```r
plot(fit.vol.sr, type = "l", main = "Standardized Residuals", ylab = "Value")
acf(fit.vol.sr, main = "ACF of Standardized Residuals")
pacf(fit.vol.sr, main = "PACF of Standardized Residuals")
Box.test(fit.vol.sr, lag = 12, type = "Ljung-Box")
acf(fit.vol.sr^2, main = "ACF of Squared Standardized Residuals")
pacf(fit.vol.sr^2, main = "PACF of Squared Standardized Residuals")
Box.test(fit.vol.sr^2, lag = 12, type = "Ljung-Box")

# Residual checking for ARMA(1,1) + GARCH(1,1)
fit.vol = volatility(arma11_garch11)
fit.vol.sr = residuals(arma11_garch11) / fit.vol
plot(fit.vol, type = "l", main = "Conditional Volatility", ylab = "Volatility")
plot(fit.vol.sr, type = "l", main = "Standardized Residuals", ylab = "Value")
acf(fit.vol.sr, main = "ACF of Standardized Residuals")
pacf(fit.vol.sr, main = "PACF of Standardized Residuals")
Box.test(fit.vol.sr, lag = 12, type = "Ljung-Box")
acf(fit.vol.sr^2, main = "ACF of Squared Standardized Residuals")
pacf(fit.vol.sr^2, main = "PACF of Squared Standardized Residuals")
Box.test(fit.vol.sr^2, lag = 12, type = "Ljung-Box")

# Residual checking for ARMA(2,0) + GARCH(1,1)
fit.vol = volatility(arma20_garch11)
fit.vol.sr = residuals(arma20_garch11) / fit.vol
plot(fit.vol, type = "l", main = "Conditional Volatility", ylab = "Volatility")
plot(fit.vol.sr, type = "l", main = "Standardized Residuals", ylab = "Value")
acf(fit.vol.sr, main = "ACF of Standardized Residuals")
pacf(fit.vol.sr, main = "PACF of Standardized Residuals")
Box.test(fit.vol.sr, lag = 12, type = "Ljung-Box")
acf(fit.vol.sr^2, main = "ACF of Squared Standardized Residuals")
pacf(fit.vol.sr^2, main = "PACF of Squared Standardized Residuals")
Box.test(fit.vol.sr^2, lag = 12, type = "Ljung-Box")

arma_garch_forecast = predict(arma21_garch11, n.ahead=30,plot=T)
arma_garch_forecast
```