

# Time Series Analysis and Forecasting of S&P 500 Weekly Returns: An ARMA Modeling Approach

Matthew Dikman

October 30, 2025

## Abstract

This project identifies and validates an appropriate ARMA model for forecasting S&P 500 weekly log returns and price levels using Yahoo Finance data from January 1990 to October 2025 ( $n = 1,868$  weeks). Adjusted closing prices were converted to log returns to achieve a stationary dataset suitable for ARIMA modeling. Candidate models were compared using AICc, BIC, and residual diagnostics. The top models, ARMA(1,1) and AR(2), had nearly identical criteria values and were evaluated through 52-week walk-forward cross-validation. Diebold–Mariano testing found no significant difference in predictive accuracy. For parsimony and interpretability, ARMA(1,1) was selected as the final model. Forecasts for 26 weeks ahead showed rapid mean reversion toward zero and widening confidence intervals at the price level, consistent with efficient market expectations. The analysis confirms ARIMA models capture weak short-term dependence in returns but provide limited predictive power beyond a few weeks.

## 1. Introduction

### 1.1 Motivation and Dataset Selection

The S&P 500 is a key benchmark representing about 80% of U.S. market capitalization across 500 leading companies. Understanding short-term return dynamics helps portfolio managers and analysts assess predictability and market efficiency.

ARMA models provide a framework to examine serial dependence in financial returns. While the efficient market hypothesis suggests prices follow a random walk, small mean-reverting or momentum effects may exist. Modeling these allows assessment of market predictability.

Weekly returns are used instead of daily data to capture meaningful short-term behavior while reducing high-frequency noise. The dataset spans 1990–2025, covering multiple economic cycles (Dot-Com bubble, 2008 crisis, and COVID-19), offering diverse market conditions to improve generalizability.

## 1.2 Data Source and Preparation

Historical prices for the S&P 500 index (^GSPC) were obtained via the quantmod package in R. Daily adjusted closes were aggregated to weekly data ( $n = 1,868$ ). A Box–Cox test indicated a log transformation was appropriate ( $\lambda \approx 0$ ). Log-differenced weekly prices produced stationary returns as follows:

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

where  $(P_t)$  denotes the weekly adjusted close price and  $r_t$  denotes the log return.

## 1.3 Research Objectives

This analysis aims to: (1) identify an appropriate ARMA model specification for S&P 500 weekly log returns through information criteria and residual diagnostic evaluation, (2) validate the selected model's forecasting performance through out-of-sample testing with walk-forward cross-validation, and (3) generate a 26-week future forecast of both returns and price levels with 95% confidence intervals.

## 2. Methodology

### 2.1 Exploratory Data Analysis and Stationarity Assessment

The analysis begins by plotting the price levels and log returns to assess stationarity properties and identify appropriate modeling strategies.

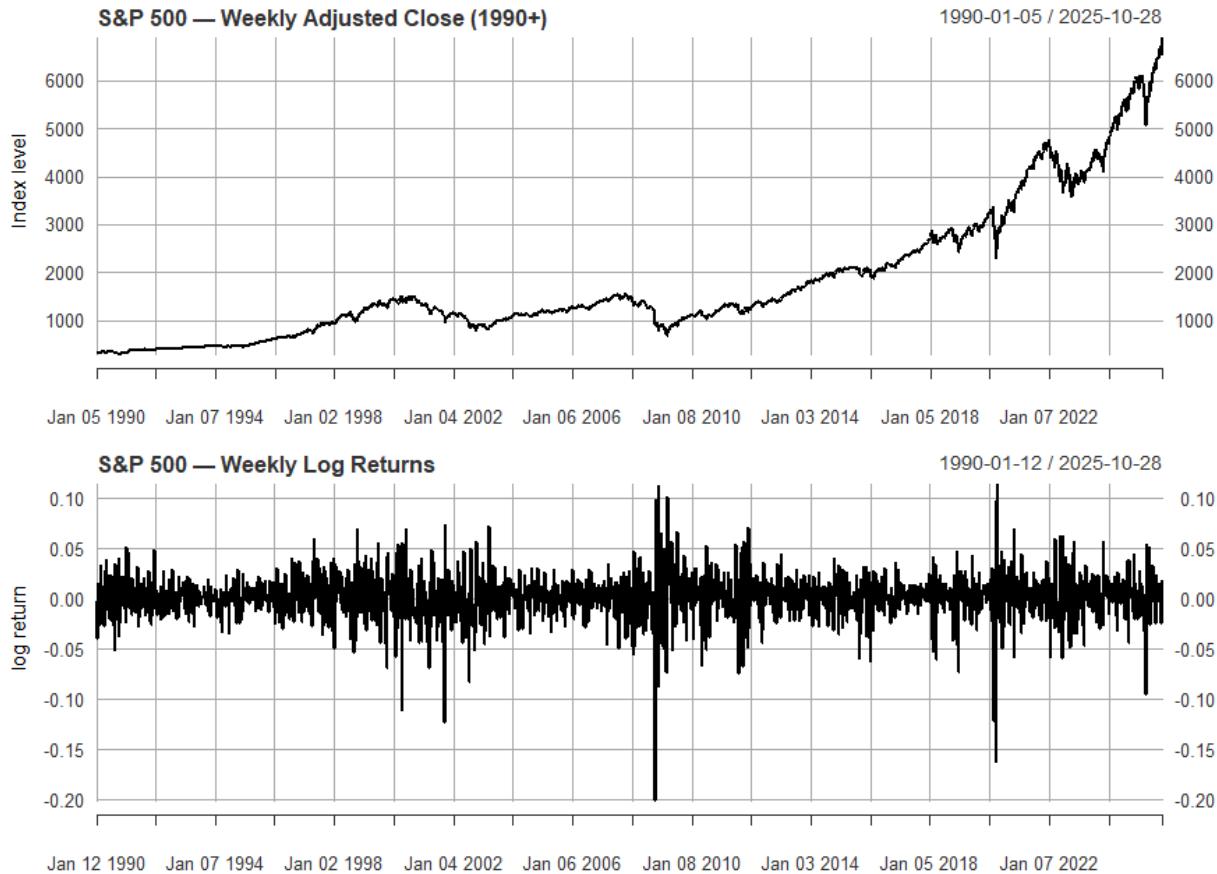


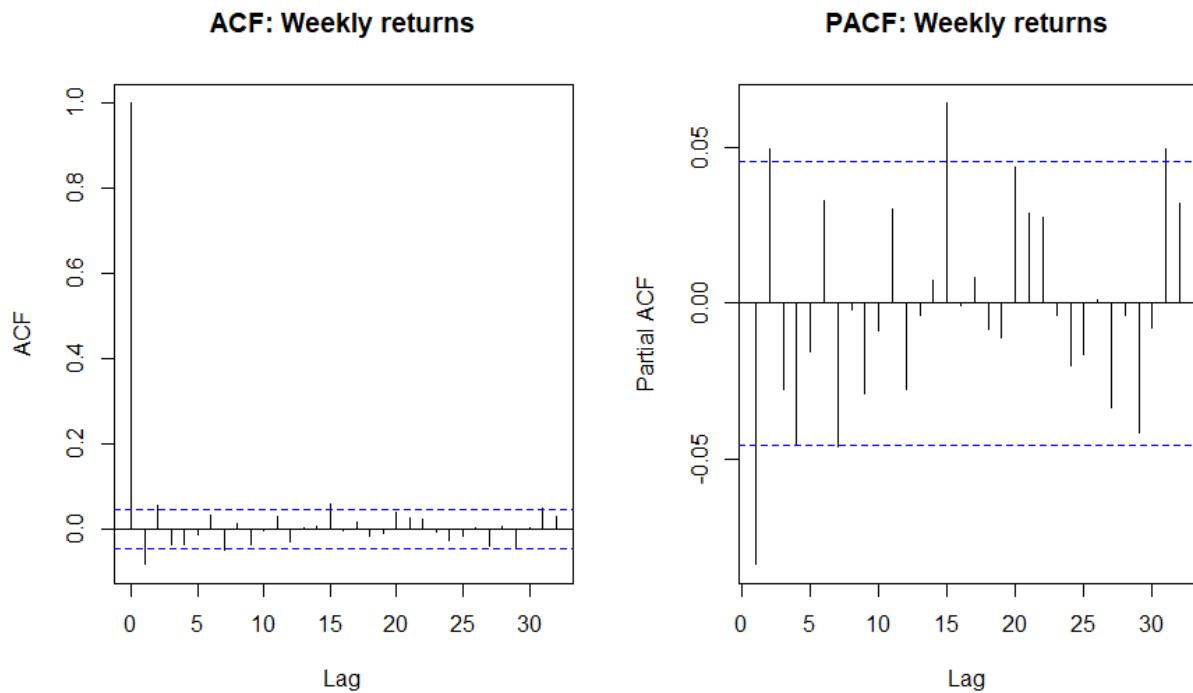
Figure 1: S&P 500 weekly adjusted close and log returns (January 1990 – October 2025). The top panels shows the non-stationary price levels with a clear upward trend and variance evolving with time. The bottom panel shows log returns oscillating around zero with no persistent trend, with volatility clustering during corresponding market crises.

As expected, price levels exhibit clear non-stationarity with an upward trend and heteroskedastic variance that magnifies during extreme market conditions. Comparatively, log returns fluctuate around a stable mean near zero and do not present any clear trend. However, there is pronounced volatility clustering, a common trait of financial time series data, indicating varying conditional variance depends upon time. ARIMA models do not capture this feature, however GARCH family extensions could address this in future work. To formally test for stationarity, the Augmented Dickey-Fuller (ADF) test was utilized. The test strongly rejects the null hypothesis

of a unit root, confirming that the series is stationary and that proceeding with ARIMA modeling is appropriate.

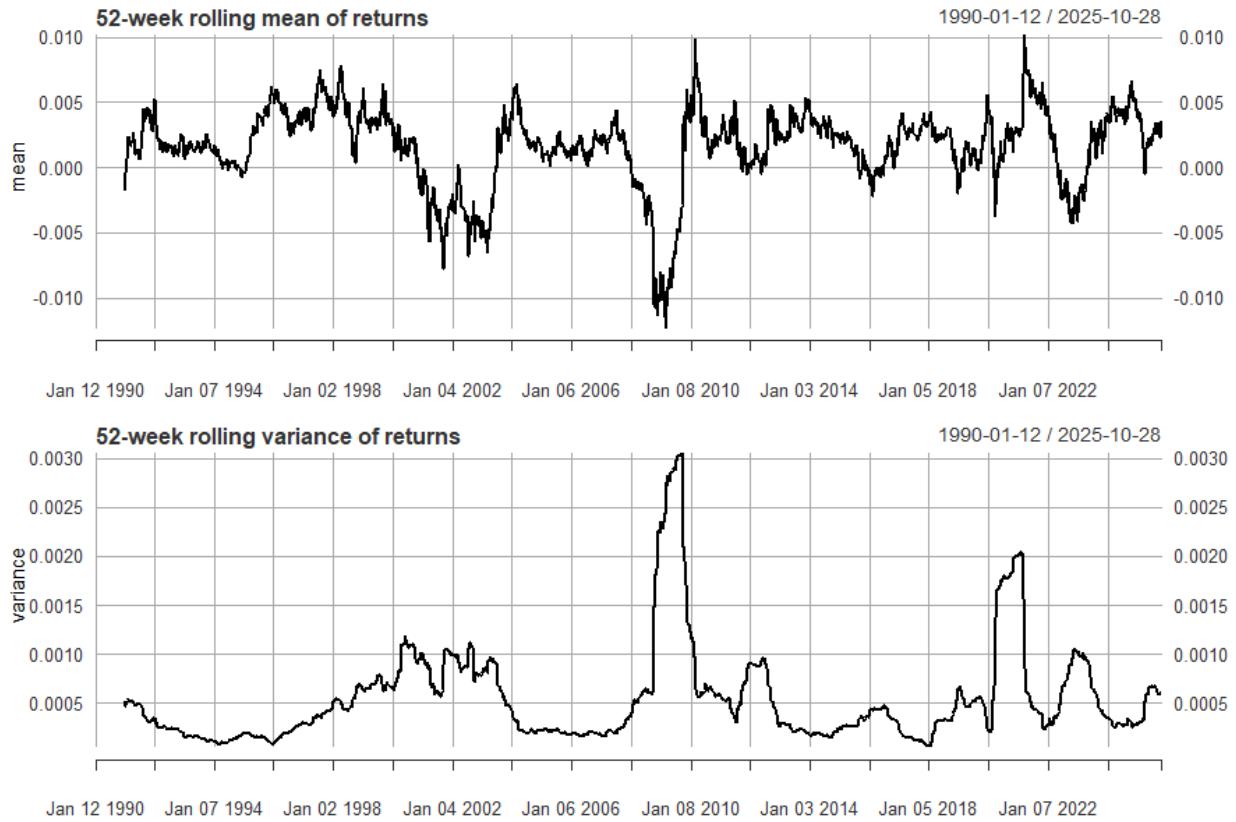
## 2.2 Autocorrelation Structure

The autocorrelation function (ACF) and partial autocorrelation function (PACF) provide initial guidance for candidate models by revealing the magnitude and pattern of serial dependence in the data.



*Figure 2: ACF and PACF of S&P weekly log returns. Both plots show minimal autocorrelation beyond lag 1, with only a weak negative spike at lag 1. Nearly all lags fall within the 95% confidence bands suggesting the series exhibits behavior similar to white noise.*

The ACF and PACF indicate almost no significant autocorrelation beyond lag 1, which would be expected for equity returns. A small but statistically significant negative spike at lag 1 suggests weak short-term mean reversion. The absence of a strong autocorrelation pattern confirms that the returns are approximately stationary and that only low-order ARMA models need to be considered. To further investigate stationarity over time, 52-week rolling means and variances were calculated and plotted.



*Figure 3: 52-week rolling mean and variance of weekly log returns. The rolling mean (top) remains persistent near zero throughout the entire sample period, confirming mean stationarity. The rolling variance (bottom) exhibits extreme spikes during major crises demonstrating volatility clustering.*

The rolling mean fluctuates around zero without a trend, supporting the mean stationarity assumption necessary for ARMA modeling. However, the rolling variance clearly varies overtime with extreme increases during volatile market conditions. While ARMA models are sufficient for the conditional mean, the conditional variance is not constant. This feature will be addressed using GARCH models in future analysis beyond the scope of this project.

## 2.3 Model Selection Framework

Based on the observed ACF and PACF patterns showing significant spikes at one or two lags, only low-order models were considered: AR(1), AR(2), MA(1), MA(2), ARMA(1,1), ARMA(2,1), and ARMA(1,2). This initial selection represents models capable of capturing the observed autocorrelation structure with the least complexity. All models were estimated using

maximum likelihood with the intercept included, as a t-test confirmed the unconditional mean of returns was significantly different from zero. Each model was fit using R's arima() function.

Model comparison utilized three information criteria: AIC, AICc, and BIC. Models were ranked primarily by AICc as the more conservative criterion for comparing models with different number of parameters. The following results were obtained.

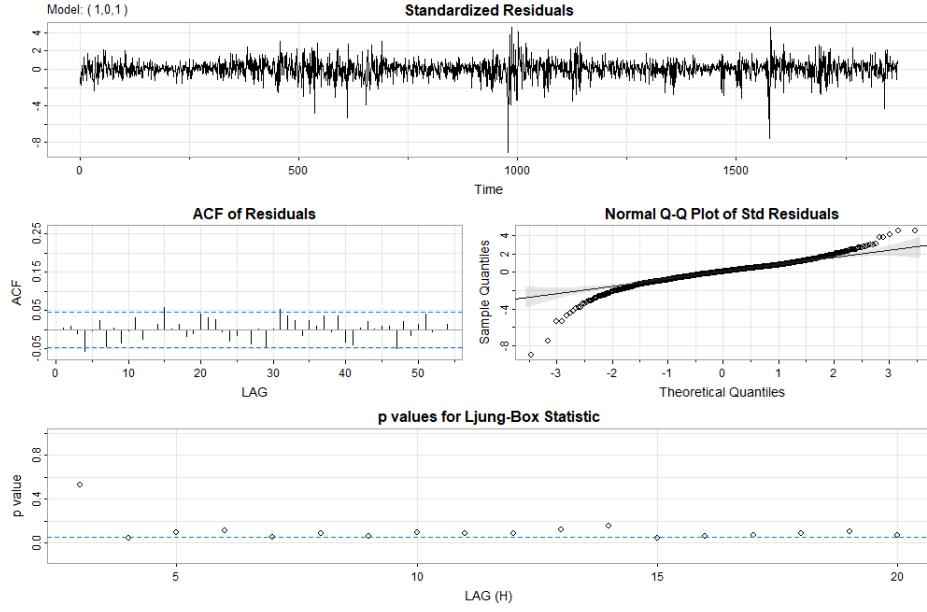
<b>Model</b>	<b>AIC</b>	<b>AICc</b>	<b>BIC</b>
<b>ARMA(1,1)</b>	-8753.035	-8753.014	-8730.905
<b>AR(2)</b>	-8752.980	-8752.958	-8730.849
<b>MA(2)</b>	-8752.367	-8752.346	-8730.237
<b>ARMA(1,2)</b>	-8751.678	-8751.646	-8724.015
<b>ARMA(2,1)</b>	-8751.513	-8751.481	-8723.85
<b>AR(1)</b>	-8750.359	-8750.346	-8733.761
<b>MA(1)</b>	-8749.141	-8749.128	-8732.543

*Table 1: Information criteria for candidate ARMA models. All models estimated with maximum likelihood on the full sample (n=1868). ARMA(1,1) and AR(2) show nearly identical AICc results indicating no clear winner.*

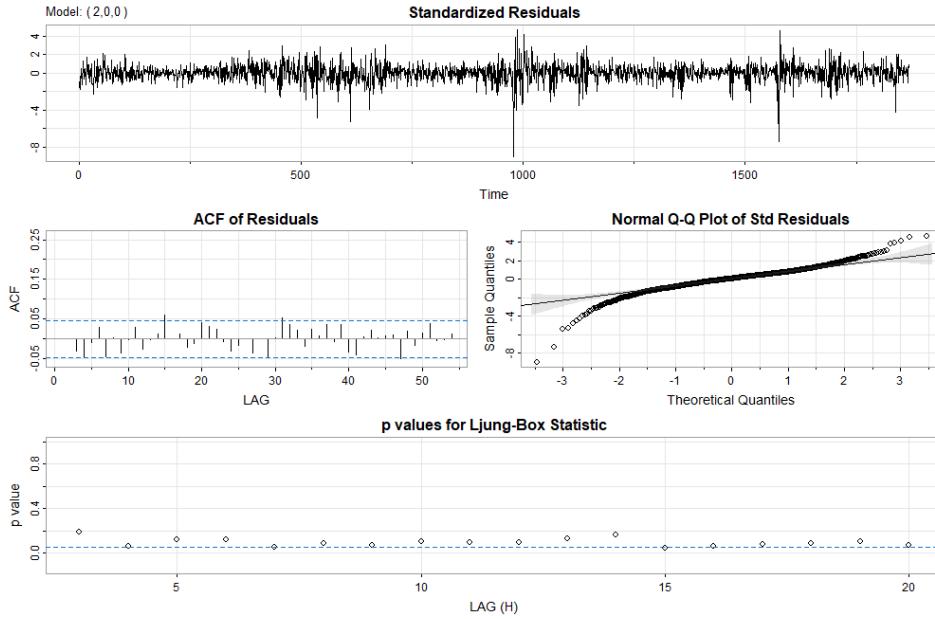
The three top models: ARMA(1,1), AR(2), and MA(2) all have AICc values within one unit of each other, suggesting a statistically equivalent fit. Given this similarity, analysis proceeded with ARMA(1,1) and AR(2) to represent two parsimonious but distinct processes, one combining autoregressive and moving-average components, and one modeling a pure autoregressive process.

## 2.4 Residual Diagnostics

ARMA model specification requires that residuals behave as white noise, uncorrelated across time with a constant variance and approximately normal distribution. Residual diagnostic checks were performed for both ARMA(1,1) and AR(2) models.



*Figure 4: Diagnostic plots for ARMA(1,1). Standardized residuals fluctuate randomly around zero with no clear pattern. ACF shows no significant spikes or patterns. The Q-Q plot indicates approximately normal residuals with deviance in the tails, expected for equity return data. Ljung-Box p-values remain mostly above the .05 level with no systematic pattern of consecutive violations.*



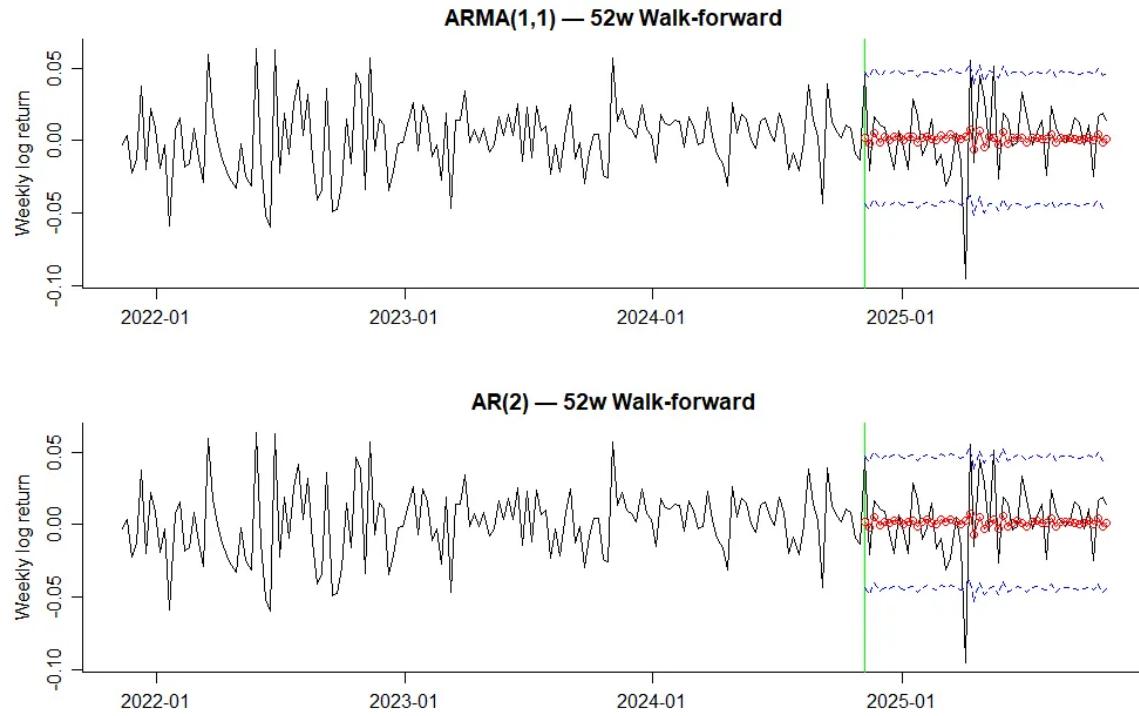
*Figure 5: Diagnostic plots for AR(2). Results nearly identical to ARMA(1,1) plots in Figure 4. Similarly, well behaved residuals that mimic white noise. No significant autocorrelation.*

Both models pass all diagnostic tests. The standardized residuals show no obvious patterns or trends and ACF plots show no significant autocorrelations. The Q-Q plots show departures from normality in the tails, which is a common feature of equity returns due to occasional periods of market volatility. This volatility will be captured in future work using GARCH-type models.

Overall, Ljung-Box test p-values remain above the .05 level, exhibiting no systematic pattern of violations.

## 2.4 Out-of-Sample Validation

Since information criteria and residual diagnostics did not distinguish between ARMA (1,1) and AR(2), out-of-sample forecasting was performed as a tiebreaker. The most recent 52 weeks of returns were withheld for testing, and both models were evaluated on their ability to forecast these unseen returns. This testing was performed using a walk-forward procedure where each model predicts the next week's return based upon all the data available at that point in time. The process is repeated over each week in the holdout period (52 weeks). Forecasting accuracy was measured using root mean squared error (RMSE) and mean absolute error (MAE).



*Figure 6: Walk-forward 1-step-ahead forecasts on 52-week holdout period. Both ARMA(1,1) (top) and AR(2) (bottom) track actual returns (black) similarly, with forecasts (red) remaining close to zero and 95% prediction intervals (blue dashed) capturing most observations. The green vertical line marks the beginning of the holdout period. Visual comparison confirms the similar performance indicated by RMSE/MAE metrics.*

The forecast plots reveal that both models produce similar predictions that hover near zero, the unconditional mean of returns. This behavior is expected for equity returns, where past values provide minimal information about future movements (supporting the efficient market theory). The 95% confidence bands capture most actual observations, indicating well-calibrated uncertainty estimates. Occasional large deviations during volatile periods reflect the inherent unpredictability of equity markets. Future work will utilize GARCH-type models to try to capture some of this conditional volatility.

Model	RMSE	MAE
<b>ARMA(1,1)</b>	0.0241	0.0177
<b>AR(2)</b>	0.0239	0.0176

*Table 2: Out-of-sample forecast accuracy on 52-week holdout using walk-forward 1-step-ahead forecasts. Both models perform nearly identically.*

Forecasting results show virtually no difference in forecast accuracy with RMSE and MAE values differing by an insignificant amount. The Diebold-Mariano test confirms this, failing to reject the null hypothesis of equal predictive accuracy ( $p = 0.217$ ). Since both models forecast equally well, the ARMA(1,1) model is selected based on parsimony as it uses only two one-lag terms compared to AR(2)'s lag 2 term. Additionally, ARMA(1,1) provides a more flexible specification by incorporating both autoregressive and moving average dynamics, whereas AR(2) captures only autoregressive structure.

### 3. Results

#### 3.1 Selected Model Specification

The estimated model for weekly log returns is as follows:

$$r_t = \mu + \varphi_1 r_{t-1} + \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

where  $\varepsilon_t \sim WN(0, 0.0005378)$

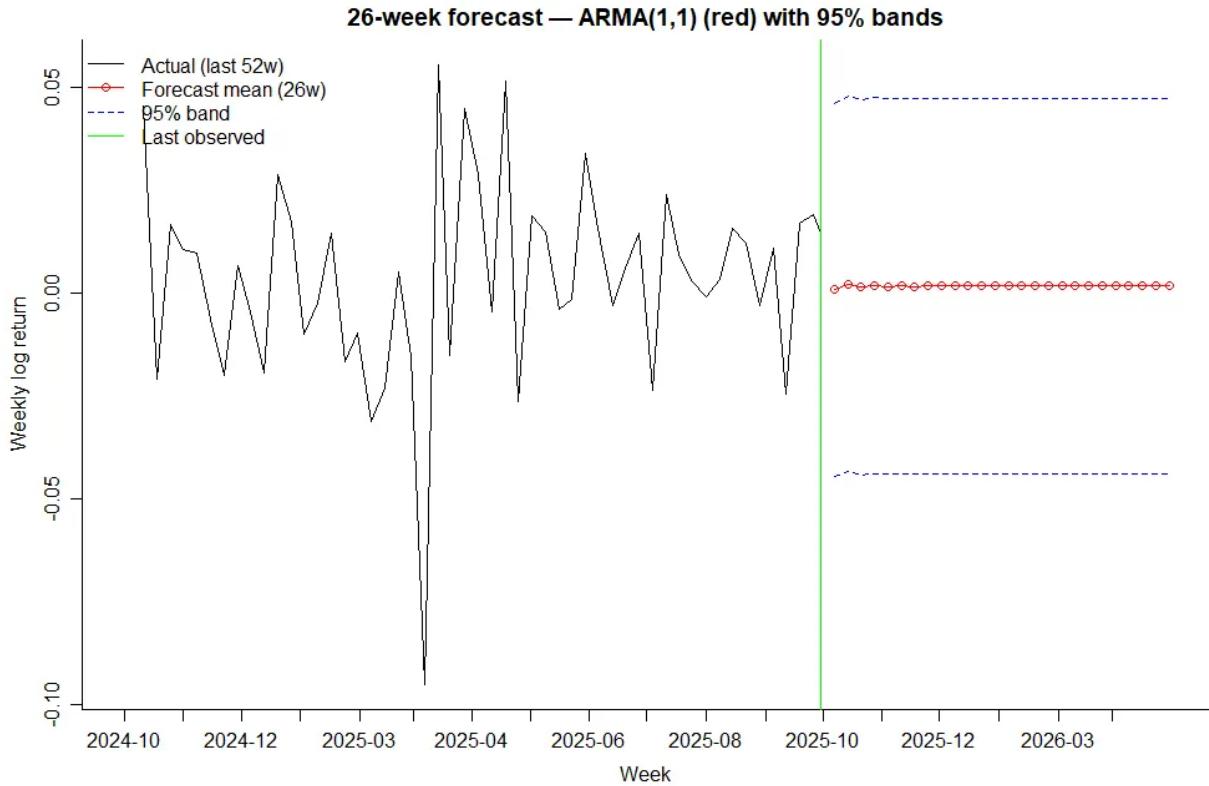
Parameter	Estimate	Std. Error	t-value	p-value
$\varphi_1$	-0.5425	0.1859	-2.92	0.0036
$\theta_1$	0.4601	0.1963	2.34	0.019
$\mu$	0.0016	0.0005	3.2	0.0014

The parameter estimates reveal a negative AR(1) coefficient and a positive MA(1) coefficient.

The negative autoregressive term suggests mild mean reversion, where positive returns tend to be followed by slightly negative returns on average. Additionally, the MA(1) component captures short-term shock adjustments. All coefficients are statistically significant at conventional levels.

#### 3.2 26-Week Return Forecast

The ARMA(1,1) model was refitted on the complete dataset ( $n=1868$  weeks) to generate forecasts for the next 26 weeks. The forecast procedure uses the model's estimated parameters to recursively predict each future period, with uncertainty quantified through 95% prediction intervals based on forecast standard errors.



*Figure 7: 26-week return forecast using ARMA(1,1). The plot shows the last 52 weeks of actual returns (black) for context, followed by 26 weeks of forecasts (red) with 95% confidence bands (blue). The green vertical line marks the last observed data point. Forecasts quickly revert to near-zero, reflecting the weak serial dependence in returns.*

The forecast path reveals a key characteristic of equity return forecasting where predictions rapidly converge to the unconditional mean (near zero). After just 3-4 weeks, the forecasts become essentially flat, indicating the model expects future returns to revert to their long-run average. This behavior confirms the weak autocorrelation structure observed in the ACF/PACF plots. The 95% confidence bands remain approximately constant width throughout the forecast horizon. This occurs because each weekly forecast has similar one-step-ahead uncertainty. While the model successfully captures the subtle short-term dynamics present in the data, it cannot produce directional forecasts beyond a few weeks. This is a common challenge when modeling efficient financial markets.

### 3.3. 26-Week Price Level Forecast

While return forecasts are statistically appropriate, price-level forecasts are often more interpretable for analysts and investors. Return forecasts were transformed to price forecasts by compounding from the last observed price:

$$\widehat{P_{t+h}} = P_t \cdot \exp\left(\sum_{i=1}^h \widehat{r_{t+1}}\right)$$

where  $P_t$  is the last observed adjusted close and  $\widehat{r_{t+1}}$  are the forecasted log returns.

Unlike return forecast uncertainty, which remains constant, price forecast uncertainty accumulates over time as small return uncertainties compound. To properly capture this compounding effect, 95% prediction intervals were constructed via Monte Carlo simulation with 5,000 paths, each simulating 26 weeks of returns from the model's predictive distributions, then converting to prices.

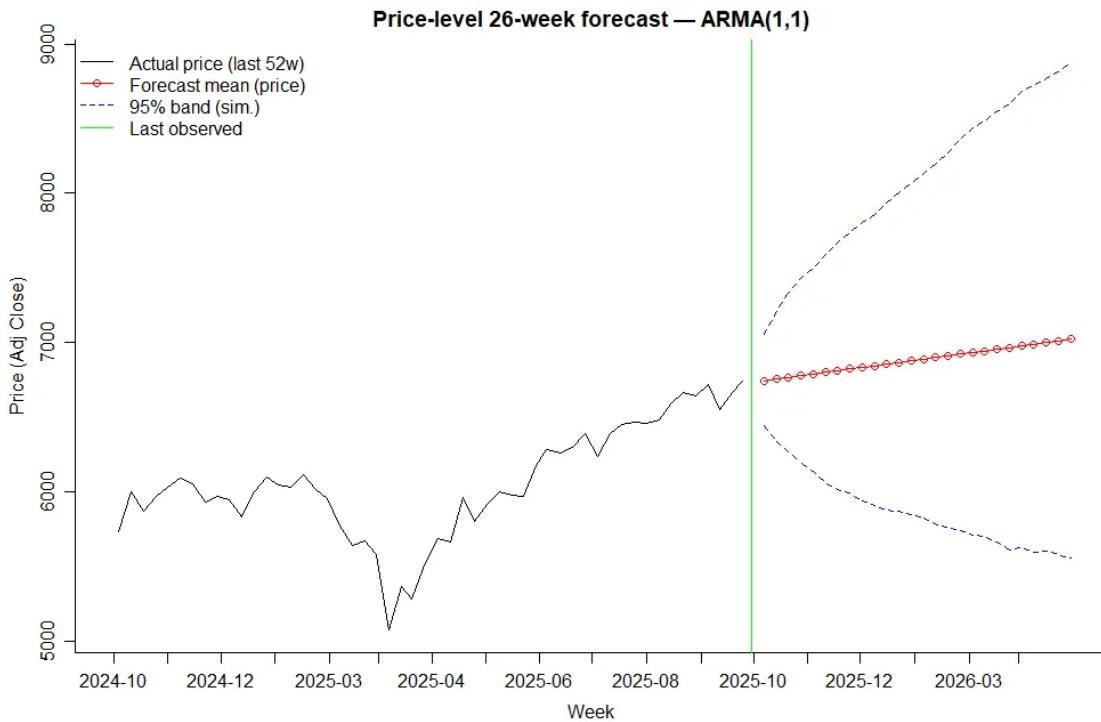


Figure 8: Price-level forecast derived from ARMA(1,1) return forecasts. The plot shows the last 52 weeks of actual adjusted close prices (black) for context, followed by 26 weeks of mean price forecasts (red) with 95% prediction bands (blue) from Monte Carlo simulation.

The price forecast shows the mean path trending slightly upward, driven by the small positive unconditional mean in returns. However, the defining feature is the dramatic widening of the prediction bands. By week 26, the 95% interval spans a substantial range, illustrating how even modest weekly return uncertainty compounds into large price-level uncertainty over six months. This widening confidence interval implies that while we can model short-term return dynamics, translating these into reliable medium-term price predictions is not reliable. The uncertainty grows so large that the forecasts offer limited practical guidance for timing market entry or exit points

## 4. Discussion

### 4.1 Limitations and Future Research Direction

Several limitations should be noted. First, the model assumes constant variance, which the rolling variance plots contradict. Volatility clustering suggests forecast uncertainty changes over time. GARCH models could better capture this time-varying volatility.

Second, the model is backward-looking, using only past returns and ignoring macroeconomic, policy, and fundamental factors. Including such variables in ARIMAX or VAR models could improve accuracy.

Third, weekly aggregation may mask short-lived dynamics visible in higher-frequency data. While it reduces noise, it can smooth out brief but meaningful patterns.

### 4.2 Conclusion

This project presents a complete time series modeling workflow for S&P 500 weekly log returns, from exploratory analysis and stationarity testing to ARMA model selection, validation, and forecasting.

The ARMA(1,1) model offered the best balance of simplicity, flexibility, and fit. Diagnostics confirmed well-behaved, white-noise residuals, and cross-validation with Diebold–Mariano testing showed no significant difference from the AR(2) model, leading to ARMA(1,1)'s selection.

The 26-week forecasts highlight a core feature of equity markets, returns quickly revert to the mean and reflect weak serial dependence and market efficiency. Although the model captures short-term behavior, predictive power fades beyond a few weeks, and price-level forecasts show rapidly widening uncertainty.

Overall, the findings reinforce that while time-series models effectively describe return dynamics, the efficiency and randomness of financial markets inherently limit forecast accuracy.

---

## References

Shumway, R. H., & Stoffer, D. S. (2017). *Time Series Analysis and Its Applications: With R Examples* (4th ed.). Springer.

Yahoo Finance. (2025). S&P 500 (^GSPC) Historical Data. <https://finance.yahoo.com/>

---

## Appendix A: R Code

```
# Libraries
library(quantmod)
library(tseries)
library(forecast)

#Download data from Yahoo
getSymbols("^GSPC",
           src = "yahoo",
           from = "1990-01-01",
           to   = "2025-10-24",
           auto.assign = TRUE)

# Use Adjusted Close for return calculations
px_d <- Ad(GSPC)  #

# Aggregate to weekly
wk   <- to.weekly(px_d, indexAt = "endof", drop.time = TRUE)
px_w <- Cl(wk)      # weekly adjusted close

# Box-Cox Transformation Check
bc <- BoxCox.lambda(px_w)
bc

# Weekly log returns (stationary)
r_w  <- na.omit(diff(log(px_w)))

# Plots: price & returns
par(mfrow = c(2,1))
plot(px_w, main = "S&P 500 - Weekly Adjusted Close", ylab = "Index level")
```

```

plot(r_w, main = "S&P 500 - Weekly Log Returns", ylab = "log return")
par(mfrow = c(1,1))

# ACF/PACF on log returns
par(mfrow = c(1,2))
acf(r_w, main = "ACF: Weekly returns")
pacf(r_w, main = "PACF: Weekly returns")
par(mfrow = c(1,1))

# Stationarity (returns) ADF test
adf_ret <- adf.test(as.numeric(r_w)); adf_ret

# Rolling mean/variance (descriptive; variance clustering expected)
roll_mean <- rollapply(r_w, width = 52, FUN = mean, by = 1, align = "right", fill = NA)
roll_var <- rollapply(r_w, width = 52, FUN = var, by = 1, align = "right", fill = NA)
par(mfrow=c(2,1))
plot(roll_mean, main = "52-week rolling mean of returns", ylab = "mean"); abline(h = 0, lty = 2)
plot(roll_var, main = "52-week rolling variance of returns", ylab = "variance")
par(mfrow=c(1,1))

# test if mean is statistically non-zero
mean_ret <- mean(r_w)
t_mean <- t.test(as.numeric(r_w), mu = 0)
mean_ret; t_mean$p.value
inc_mean <- TRUE

# Candidate grid + fits
cands <- list(
  c(1,0,0), # AR(1)
  c(2,0,0), # AR(2)
  c(0,0,1), # MA(1)
  c(0,0,2), # MA(2)
  c(1,0,1), # ARMA(1,1)
  c(2,0,1), # ARMA(2,1)
  c(1,0,2) # ARMA(1,2)
)
cand_names <- c("AR(1)", "AR(2)", "MA(1)", "MA(2)", "ARMA(1,1)", "ARMA(2,1)", "ARMA(1,2)")

fit_one_base <- function(ord, x, inc_mean=TRUE){
  suppressWarnings(arima(x, order = ord, include.mean = inc_mean, method = "ML"))
}

# AICc
AICc_manual <- function(model) {
  n <- length(model$residuals)
  k <- length(coef(model)) + 1 # +1 for sigma^2
  AIC(model) + (2*k*(k+1)) / (n - k - 1)
}

fits_base <- lapply(cands, \o) fit_one_base(o, r_w, inc_mean))
names(fits_base) <- cand_names

IC <- data.frame(
  Model = names(fits_base),
  p = sapply(cands, \z z[1]),
  d = sapply(cands, \z z[2]),
  q = sapply(cands, \z z[3]),
  AIC = sapply(fits_base, AIC),
  AICc = sapply(fits_base, AICc_manual),
  BIC = sapply(fits_base, BIC),

```

```

    row.names = NULL
  }
IC <- IC[order(IC$AICc), ]
IC

#Top Models (ARMA(1,1) & AR(2))
top <- IC[1:2, c("Model","p","d","q")]
top

fit_top1 <- fits_base[[ top$Model[1] ]]
fit_top2 <- fits_base[[ top$Model[2] ]]

#Summary
print(summary(fit_top1))
print(summary(fit_top2))

#Residual Dianostics

library(astsa)
astsa::sarima(r_w, top$p[1], top$d[1], top$q[1],
              no.constant = !inc_mean, details = TRUE)
astsa::sarima(r_w, top$p[2], top$d[2], top$q[2],
              no.constant = !inc_mean, details = TRUE)

# 52w holdout

# Split data
y_xts <- r_w
y      <- as.numeric(y_xts)
ti     <- index(y_xts)

h <- 52
n <- length(y)
train_idx <- 1:(n - h)
test_idx  <- (n - h + 1):n

y_train  <- y[train_idx]
y_test   <- y[test_idx]
ti_train <- ti[train_idx]
ti_test  <- ti[test_idx]

# Top-2 orders
ord1  <- c(top$p[1], top$d[1], top$q[1]); name1 <- top$Model[1]
ord2  <- c(top$p[2], top$d[2], top$q[2]); name2 <- top$Model[2]

# Walk-forward helper
wf_forecast <- function(y_all, train_end_index, horizon_idx, order_vec,
                           include.mean = TRUE, band = 0.95) {
  k <- length(horizon_idx)
  fmean <- fse <- err <- rep(NA_real_, k)
  L <- U <- rep(NA_real_, k)
  z <- qnorm((1 + band)/2)
  for (i in seq_len(k)) {
    fit_end <- train_end_index + i - 1
    fit_y   <- y_all[1:fit_end]
    fit     <- arima(fit_y, order = order_vec, include.mean = include.mean, method = "ML")
    pr     <- predict(fit, n.ahead = 1)
    fmean[i] <- pr$pred[1]
    fse[i]   <- pr$se[1]
  }
}

```

```

L[i]      <- fmean[i] - z * fse[i]
U[i]      <- fmean[i] + z * fse[i]
y_act    <- y_all[horizon_idx[i]]
err[i]   <- y_act - fmean[i]
}
list(mean = fmean, se = fse, L = L, U = U, error = err)
}

# 52-week walk forward evaluation
res1 <- wf_forecast(y, max(train_idx), test_idx, ord1, include.mean = inc_mean, band = 0.95)
res2 <- wf_forecast(y, max(train_idx), test_idx, ord2, include.mean = inc_mean, band = 0.95)

RMSE1 <- sqrt(mean(res1$error^2)); MAE1 <- mean(abs(res1$error))
RMSE2 <- sqrt(mean(res2$error^2)); MAE2 <- mean(abs(res2$error))

cat("\n--- Holdout (52w) 1-step-ahead errors ---\n")
cat(sprintf("%-12s RMSE = %.6f | MAE = %.6f | order=c(%d,%d,%d)\n",
            name1, RMSE1, MAE1, ord1[1], ord1[2], ord1[3]))
cat(sprintf("%-12s RMSE = %.6f | MAE = %.6f | order=c(%d,%d,%d)\n",
            name2, RMSE2, MAE2, ord2[1], ord2[2], ord2[3]))

# 52w plots
years_back <- 156 # ~3 years; change to 104 for ~2 years
start_zoom <- max(1, (n - h) - years_back + 1)

plot_zoom_panel <- function(tag, res_obj) {
  ylim <- range(c(y[start_zoom:n], res_obj$L, res_obj$U), na.rm = TRUE)
  plot(ti[start_zoom:n], y[start_zoom:n],
       type = "l", col = "black", lwd = 1.2,
       xlab = "", ylab = "Weekly log return",
       main = sprintf("%s - 52w Walk-forward", tag),
       ylim = ylim, xaxt = "n", yaxt = "s", bty = "l")
  axis.Date(1, at = pretty(ti[start_zoom:n]), format = "%Y-%m")
  abline(v = ti_test[1], col = "green") # holdout start
  lines(ti_test, y_test, col = "black", lwd = 1.5) # actual in holdout
  lines(ti_test, res_obj$mean, col = "red", type = "o") # forecast mean
  lines(ti_test, res_obj$U, col = "blue", lty = "dashed")
  lines(ti_test, res_obj$L, col = "blue", lty = "dashed")
}
op <- par(no.readonly = TRUE)

par(mfrow = c(2,1), mar = c(4,4,2,1), mgp = c(2.2, .8, 0))

plot_zoom_panel(name1, res1) # ARMA(1,1)
plot_zoom_panel(name2, res2) # AR(2)

par(op)

# Diebold-Mariano test
suppressWarnings(suppressMessages(require(forecast)))
dm_out <- dm.test(e1 = res1$error, e2 = res2$error, alternative = "two.sided", h = 1, power = 2)
print(dm_out)

# RMSE / MAE comparison table
results <- data.frame(
  Model = c(name1, name2),

```

```

p = c(ord1[1], ord2[1]),
d = c(ord1[2], ord2[2]),
q = c(ord1[3], ord2[3]),
RMSE = c(RMSE1, RMSE2),
MAE = c(MAE1, MAE2)
)

print(results, row.names = FALSE)

winner <- list(name = name1, order = ord1)

# Compute t-values and p-values
est <- fit_top1$coef
se <- sqrt(diag(fit_top1$var.coef))
tval <- est / se
pval <- 2 * (1 - pnorm(abs(tval)))

data.frame(
  Parameter = names(est),
  Estimate = est,
  Std.Error = se,
  t.value = tval,
  p.value = pval
)

# 26-week RETURNS plot
fit_full <- arima(y, order = ord1, include.mean = inc_mean, method = "ML")
fc26 <- predict(fit_full, n.ahead = 26)

z95 <- qnorm(0.975)
L95ret <- fc26$pred - z95 * fc26$se
U95ret <- fc26$pred + z95 * fc26$se

ti_future <- seq(from = ti[n], by = 7, length.out = 27)[-1]

pre_weeks <- 52
post_weeks <- 26
x_left <- as.Date(ti[n]) - pre_weeks*7
x_right <- as.Date(ti[n]) + post_weeks*7

start_idx <- max(1, n - pre_weeks + 1)
ti_obs_win <- ti[start_idx:n]
y_obs_win <- y[start_idx:n]
ylim_zoom <- range(c(y_obs_win, L95ret, U95ret), na.rm = TRUE)

op <- par(no.readonly = TRUE)
par(mfrow = c(1,1), mar = c(4,4,2,1), mgp = c(2.2, .8, 0))

plot(ti_obs_win, y_obs_win, type = "l", col = "black", lwd = 1.2,
      xlab = "Week", ylab = "Weekly log return",
      main = sprintf("26-week forecast - %s (red) with 95% bands", winner$name),
      xlim = c(x_left, x_right), ylim = ylim_zoom,
      xaxt = "n", yaxt = "s", bty = "l")
axis.Date(1, at = seq(x_left, x_right, by = "1 month"), format = "%Y-%m")
abline(v = ti[n], col = "green")
lines(ti_future, fc26$pred, col = "red", type = "o")
lines(ti_future, U95ret, col = "blue", lty = "dashed")
lines(ti_future, L95ret, col = "blue", lty = "dashed")

legend("topleft",
       c("Actual (last 52w)", "Forecast mean (26w)", "95% band", "Last observed"),

```

```

col = c("black","red","blue","green"), lty = c(1,1,2,1), pch = c(NA,1,NA,NA), bty =
"n")
par(op)

# 26-week PRICE plot
P_last <- as.numeric(last(px_w))
mu_ret <- as.numeric(fc26$pred)
se_ret <- as.numeric(fc26$se)

cum_mu <- cumsum(mu_ret)
P_mean <- P_last * exp(cum_mu)

set.seed(123)
B <- 5000
R_sim <- matrix(rnorm(B * length(mu_ret), mean = rep(mu_ret, each=B), sd = rep(se_ret,
each=B)),
nrow = B)
Cum_sim <- t(apply(R_sim, 1, cumsum))
P_sim <- P_last * exp(Cum_sim)
P_L95 <- apply(P_sim, 2, quantile, probs = 0.025, na.rm = TRUE)
P_U95 <- apply(P_sim, 2, quantile, probs = 0.975, na.rm = TRUE)

pre_weeks_price <- 52
x_leftP <- as.Date(ti[n]) - pre_weeks_price*7
x_rightP <- as.Date(ti[n]) + 26*7
px_obs <- px_w[paste0(as.Date(x_leftP), "/", as.Date(ti[n]))]
ylimP <- range(c(as.numeric(px_obs), P_L95, P_U95), na.rm = TRUE)

op <- par(no.readonly = TRUE)
par(mfrow = c(1,1), mar = c(4,4,2,1), mgp = c(2.2, .8, 0))

plot(index(px_obs), as.numeric(px_obs),
      type = "l", col = "black", lwd = 1.2,
      xlab = "Week", ylab = "Price (Adj Close)",
      main = sprintf("Price-level 26-week forecast - %s", winner$name),
      xlim = c(x_leftP, x_rightP), ylim = ylimP, xaxt = "n", bty = "l")
axis.Date(1, at = seq(x_leftP, x_rightP, by = "1 month"), format = "%Y-%m")
abline(v = ti[n], col = "green")
lines(ti_future, P_mean, col = "red", type = "o")
lines(ti_future, P_U95, col = "blue", lty = "dashed")
lines(ti_future, P_L95, col = "blue", lty = "dashed")

legend("topleft",
      c("Actual price (last 52w)", "Forecast mean (price)", "95% band (sim.)", "Last
observed"),
      col = c("black","red","blue","green"), lty = c(1,1,2,1), pch = c(NA,1,NA,NA), bty =
"n")
par(op)

```