Time_series

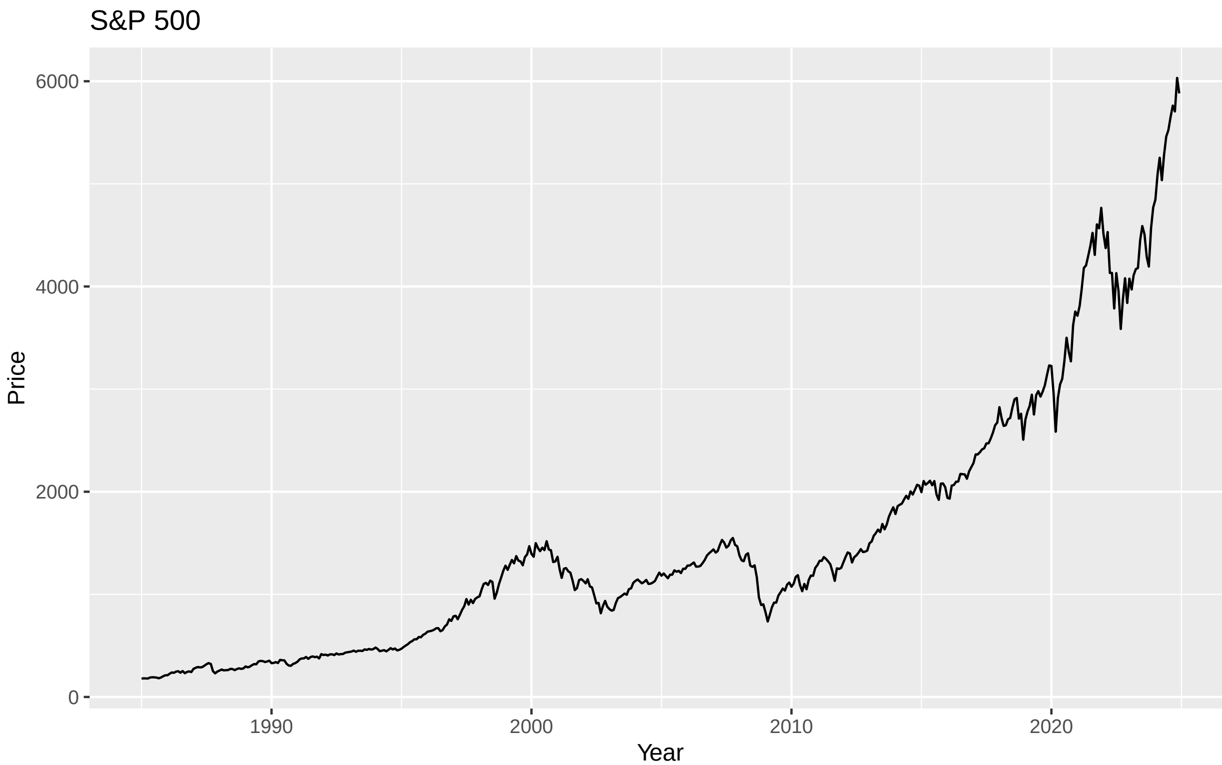
Shubham Rawat

2025-05-01

1. INTRODUCTION

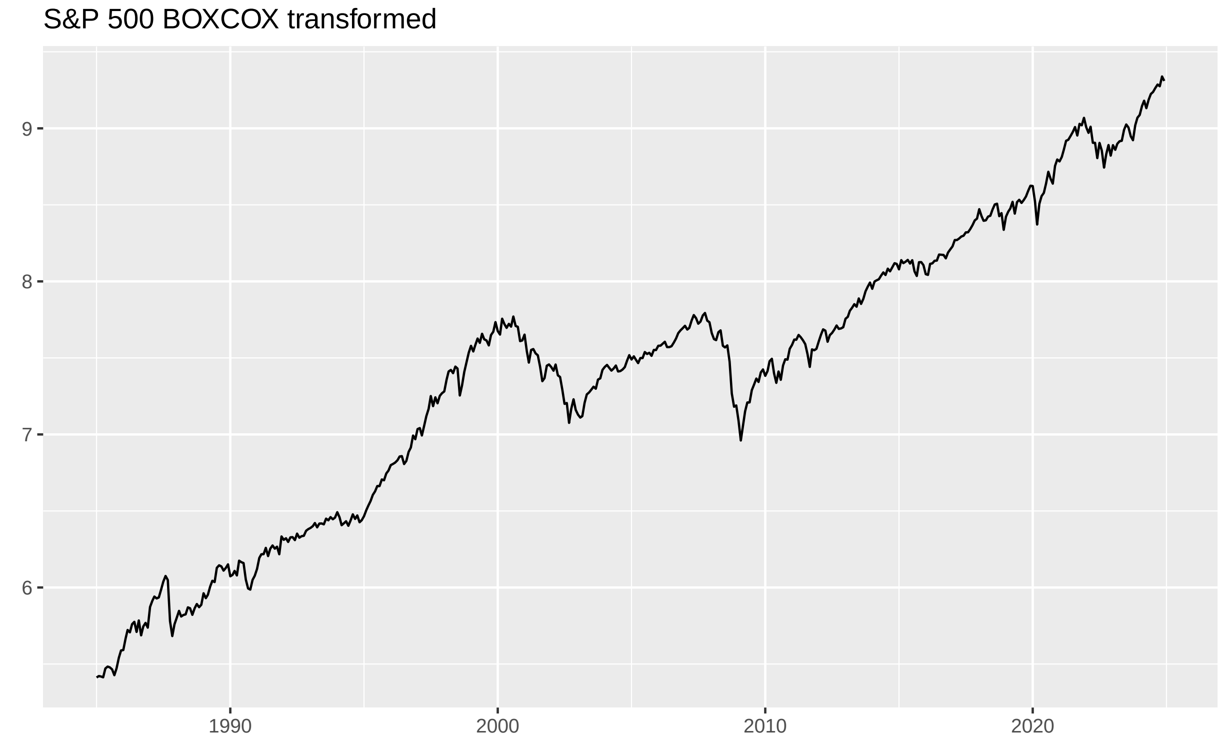
This project aims to analyze and forecast the S&P 500 index from January 1985 to December 2024 using various time series models. The objective is to evaluate model performance, understand the trends and seasonality, and determine the best forecasting method to predict future values. Techniques like Exponential Smoothing (ETS), ARIMA, SARIMA, Holt-Winters, and baseline models were applied.

2. DATA PREPRATION



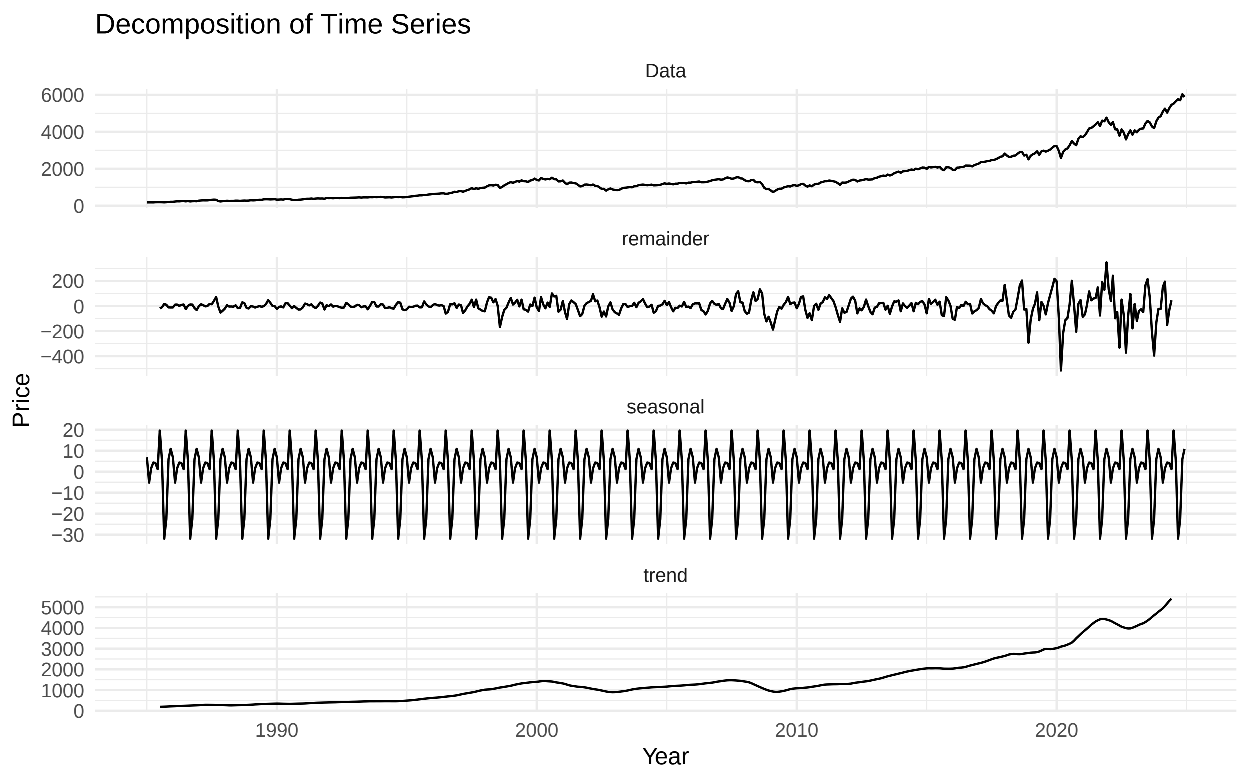
This plot shows a strong upward trend, significant periods of growth and decline, and apparent changes in volatility over time, consistent with what one might expect from a stock price index like the S&P 500.

3. DATA TRANSFORMATION



To stabilize the variance, a Box-Cox transformation was applied. The optimal lambda was found using `BoxCox.lambda()`, which returned a value of approximately 0.016. This transformation makes the data more suitable for modeling by reducing heteroscedasticity.

4. Decomposition of Time Series



The time series decomposition shown above breaks down the original S&P 500 index data into three key components:

Trend The trend component reflects the long-term progression of the index. It captures the sustained growth in the S&P 500 over time, with noticeable upward movement especially after 2010. The trend line smooths out short-term fluctuations to reveal the underlying direction of the market.

Seasonality The seasonal component reveals repetitive, periodic patterns within the data—often occurring annually. In this decomposition, we observe consistent seasonal fluctuations of about ±20 points each year. This suggests that certain months tend to exhibit predictable behavior in market activity, possibly due to investor cycles or macroeconomic patterns.

Remainder (Residuals) The remainder or irregular component captures the noise or random fluctuations not explained by trend or seasonality. It appears relatively stable until around 2010, after which variability increases—possibly due to market volatility, economic shocks, or geopolitical events. A good model should leave behind residuals that resemble white noise, indicating that most structure in the data has been explained.

Original Data At the top, the original time series data shows the raw monthly S&P 500 index values from 1985 to 2024. The index exhibits exponential growth over time, along with short-term variability.

5. MODELS

5.a. Base Models

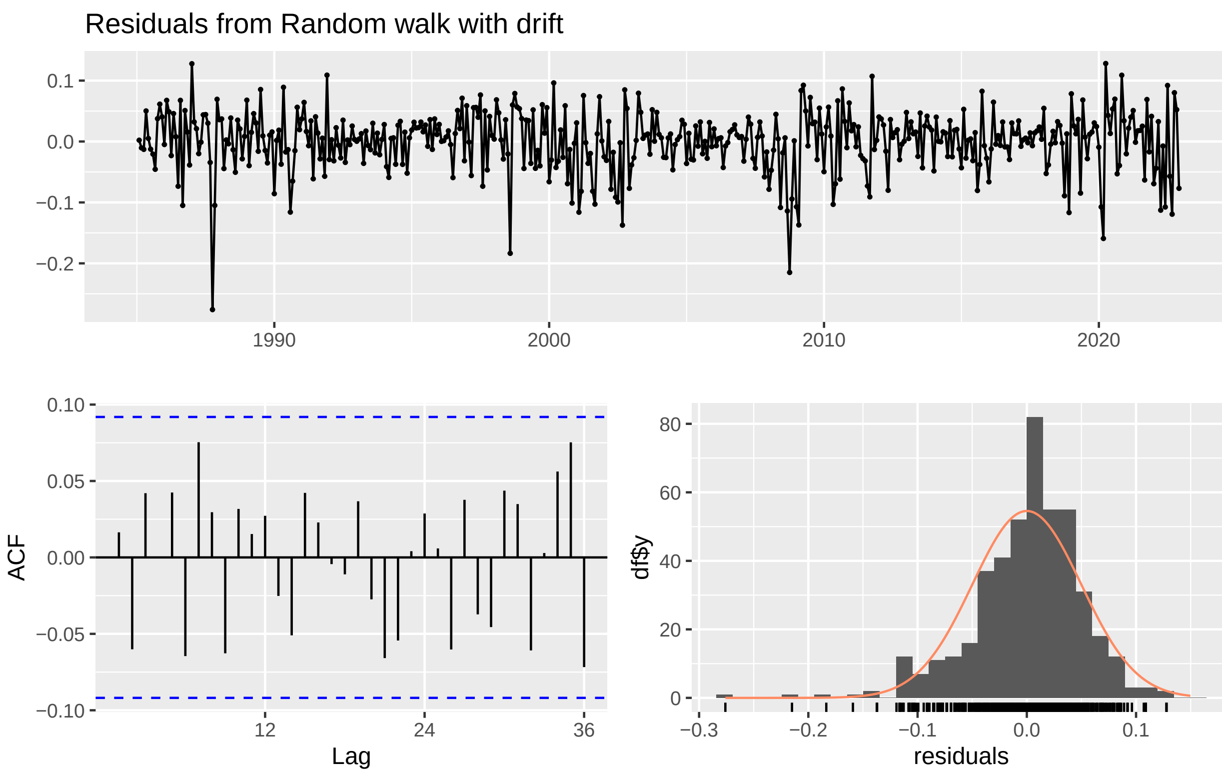
```
##   Model      RMSE       MAE       MAPE
## 1  Mean 1.7915117 1.7852306 0.19620495
## 2 Naive 0.3062943 0.2671147 0.02912608
```

```
## 3 Drift 0.2003695 0.1734423 0.01890806
```

```
## Best model based on RMSE is: Drift
```

Based on the Root Mean Squared Error (RMSE), the Drift model is selected as the best-performing base model. Its ability to capture the overall upward trend in the S&P 500 makes it a stronger foundation than the mean or naive models.

Checking residuals to see if there's normality and white noise




```
##
##  Ljung-Box test
##
## data:  Residuals from Random walk with drift
## Q* = 18.752, df = 24, p-value = 0.7651
##
## Model df: 0.   Total lags used: 24
```

Since the p-value is well above 0.05, we fail to reject the null hypothesis that the residuals are independently distributed. This means there is no significant autocorrelation, which is desirable for a well-fitted time series model.

The residual analysis confirms that the Drift model produces residuals that behave like white noise — they are approximately normally distributed, have constant variance, and show no autocorrelation. These findings support the model's validity for forecasting short-term trends in the S&P 500 index.

Ljung-Box Test (for autocorrelation)

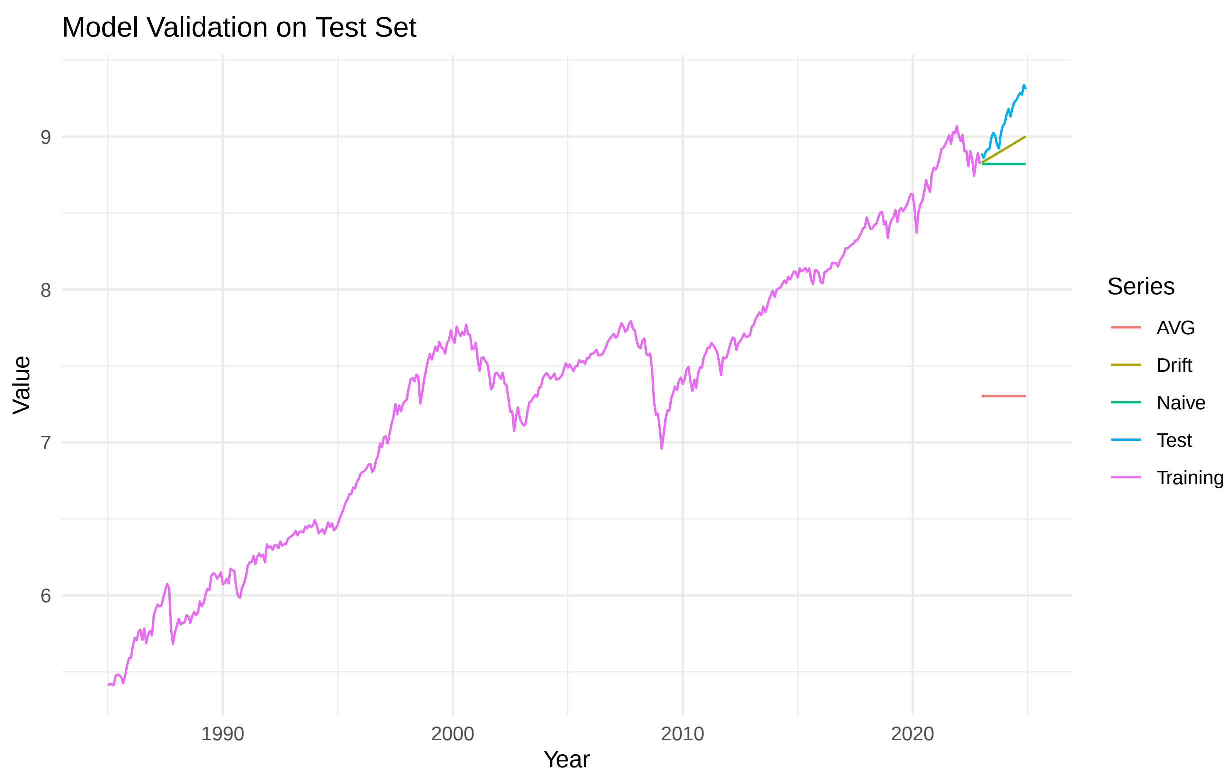
```
##
##  Box-Ljung test
##
## data:  drift_model$residuals
## X-squared = 0.12342, df = 1, p-value = 0.7254
```

The null hypothesis of the Box-Ljung test is that the residuals are independently distributed (i.e., no autocorrelation). A high p-value (above 0.05) means we fail to reject the null hypothesis.

In this case, the p-value of 0.7254 strongly suggests that there is no significant autocorrelation in the residuals of the Drift model. This confirms that the model has captured all the predictable structure in the data, and the remaining residuals behave like random noise.

This test result reinforces that the Drift model is statistically sound, with residuals that are well-behaved and uncorrelated. It strengthens the justification for choosing the Drift model as the best baseline forecasting method.

Plotted Base Models



This visual comparison confirms the findings from error metrics: the Drift model best captures the upward trend in the S&P 500 index and produces forecasts closest to actual values in the test set

5.b. ETS models

ETS Multiplicative(Without BoxCox)

```
##      Model     RMSE       MAE      MAPE      AIC     AICc
## 1 Auto_ETS 1168.969  995.1137 0.1904849 6243.161 6243.294
## 2     ETS2 1288.956 1084.2267 0.2068714 6278.933 6280.498
## 3     ETS3 1230.565 1050.6149 0.2013221 6266.095 6267.661
```

```
## Best model based on RMSE is: Auto_ETS
```

Among the multiplicative ETS models tested on the original scale, Auto_ETS is the best performer in terms of forecast accuracy and model simplicity. It will serve as a benchmark before evaluating ETS models with Box-Cox transformation.

ETS Additive(With BoxCox)

```
##         Model      RMSE       MAE       MAPE       AIC      AICc
## 1 Auto_ETS_BC 0.1997519 0.1728912 0.01884793  65.80563  65.93896
## 2     ETS_BC2 0.2083541 0.1803218 0.01965949 120.45200 121.84926
## 3     ETS_BC3 0.2963702 0.2570306 0.02802177 177.26673 178.35763
```

```
## Best model based on RMSE is: Auto_ETS_BC
```

Auto_ETS_BC clearly outperforms the manually specified models across all evaluation metrics. The lowest RMSE and MAE indicate that this model captures trends and patterns more accurately than the others. The low MAPE of 1.88% implies that, on average, forecasted values deviate less than 2% from the actual data—a strong level of predictive precision. The Auto_ETS_BC model (ETS with Box-Cox transformation) is the best-performing model among the ETS family.

Shapiro-Wilk Test for normality

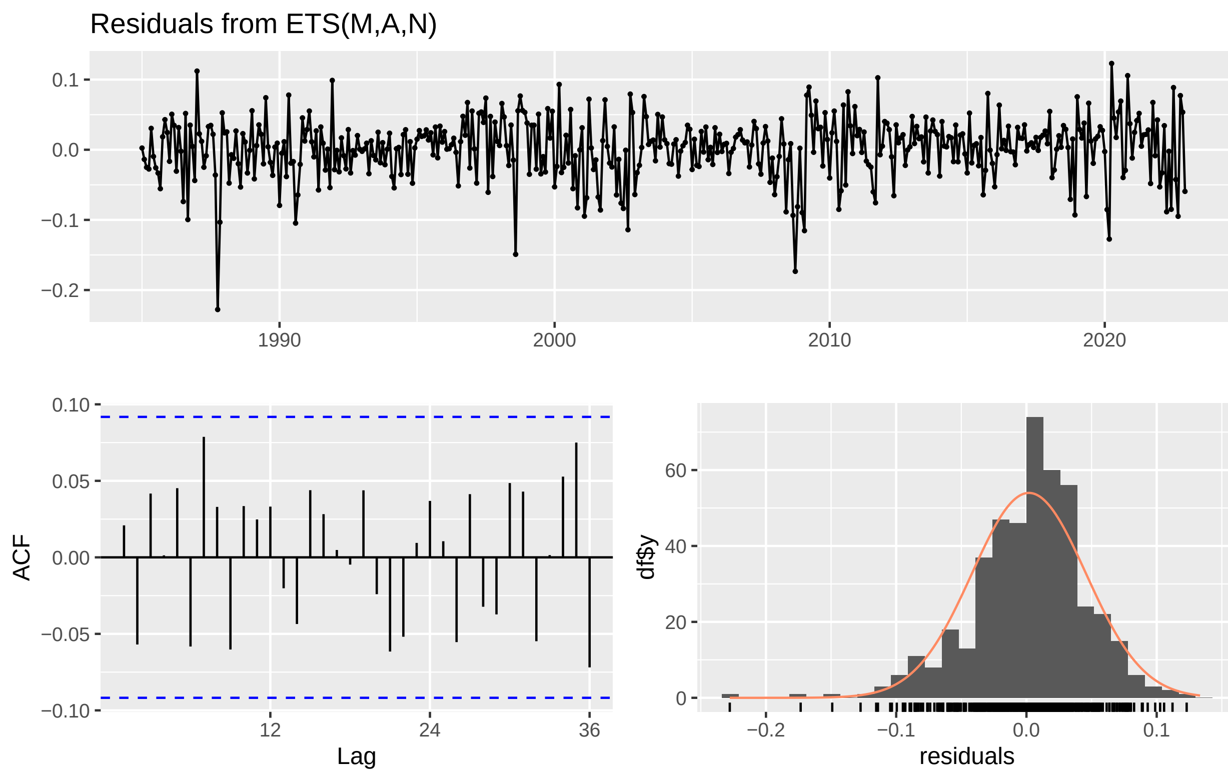
```
## 
##  Shapiro-Wilk normality test
## 
## data:  residuals(ets_fc_1)
## W = 0.9693, p-value = 3.517e-08
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  residuals(ets_fc_box_1)
## W = 0.95502, p-value = 1.439e-10
```

In both cases, the p-values are significantly less than 0.05, leading us to reject the null hypothesis of normality. This suggests that the residuals from both ETS models (with and without Box-Cox transformation) do not follow a normal distribution. However, in time series forecasting, normality of residuals is not a strict requirement—especially if the residuals are uncorrelated and centered around zero. Despite non-normality, if the residuals are homoscedastic (constant variance) and exhibit no autocorrelation, the model may still be valid for forecasting purposes.

The residuals from both ETS models are not perfectly normal, as indicated by the Shapiro-Wilk test. However, this does not disqualify the models. Residuals should be assessed in combination with autocorrelation checks (e.g., Ljung-Box test)

Checking residuals to see if there's normality and white noise




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,A,N)
## Q* = 18.84, df = 24, p-value = 0.7605
##
## Model df: 0.   Total lags used: 24
```

Since the p-value is well above 0.05, we fail to reject the null hypothesis that residuals are independently distributed. This confirms that the residuals show no significant autocorrelation, which is a desirable property.

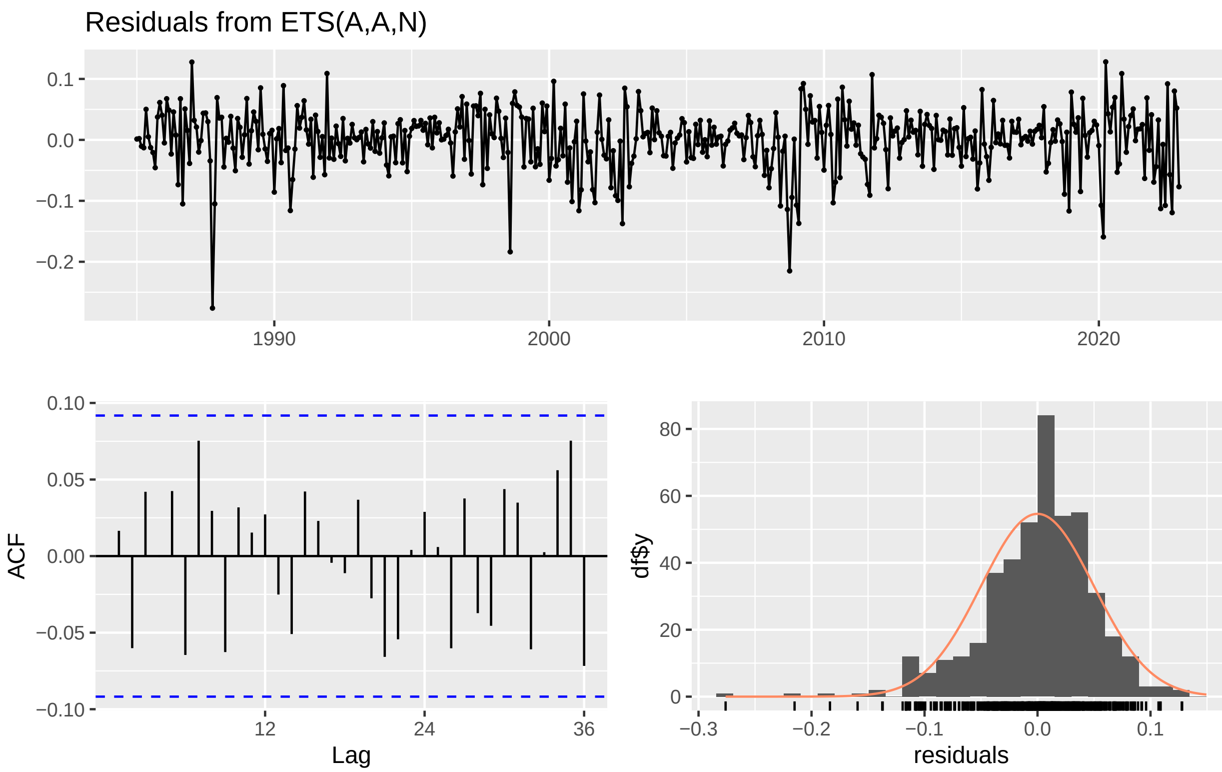
Residual Plots Top Plot (Time Series of Residuals): Residuals are randomly scattered around zero without visible structure, suggesting that the model captured the underlying trend effectively.

ACF Plot (Autocorrelation Function): The autocorrelations of residuals fall within the 95% confidence limits, reinforcing that there is no systematic pattern left unexplained.

Histogram with Normal Curve: While the Shapiro-Wilk test previously indicated non-normality, the residual histogram shows a roughly bell-shaped distribution. The slight asymmetry is not uncommon in financial data but does not invalidate the model.

The ETS(M,A,N) model exhibits well-behaved residuals — no autocorrelation, constant variance, and approximate normality. Despite a non-normal result in the Shapiro-Wilk test, the residual behavior aligns with expectations for a valid forecasting model.

Checking residuals to see if there's normality and white noise




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,A,N)
## Q* = 18.791, df = 24, p-value = 0.763
##
## Model df: 0.   Total lags used: 24
```

A high p-value (> 0.05) indicates that the residuals are not significantly autocorrelated. This result confirms that the model has effectively captured the time series dynamics and that the residuals resemble white noise—i.e., they are random and uncorrelated.

The residuals from the ETS(A,A,N) model meet the key assumptions for valid forecasting: no autocorrelation, zero mean, and homoscedasticity (constant variance). These characteristics confirm that the model has extracted the meaningful structure from the data and can be trusted for generating reliable forecasts.

Ljung-Box Test (for autocorrelation)

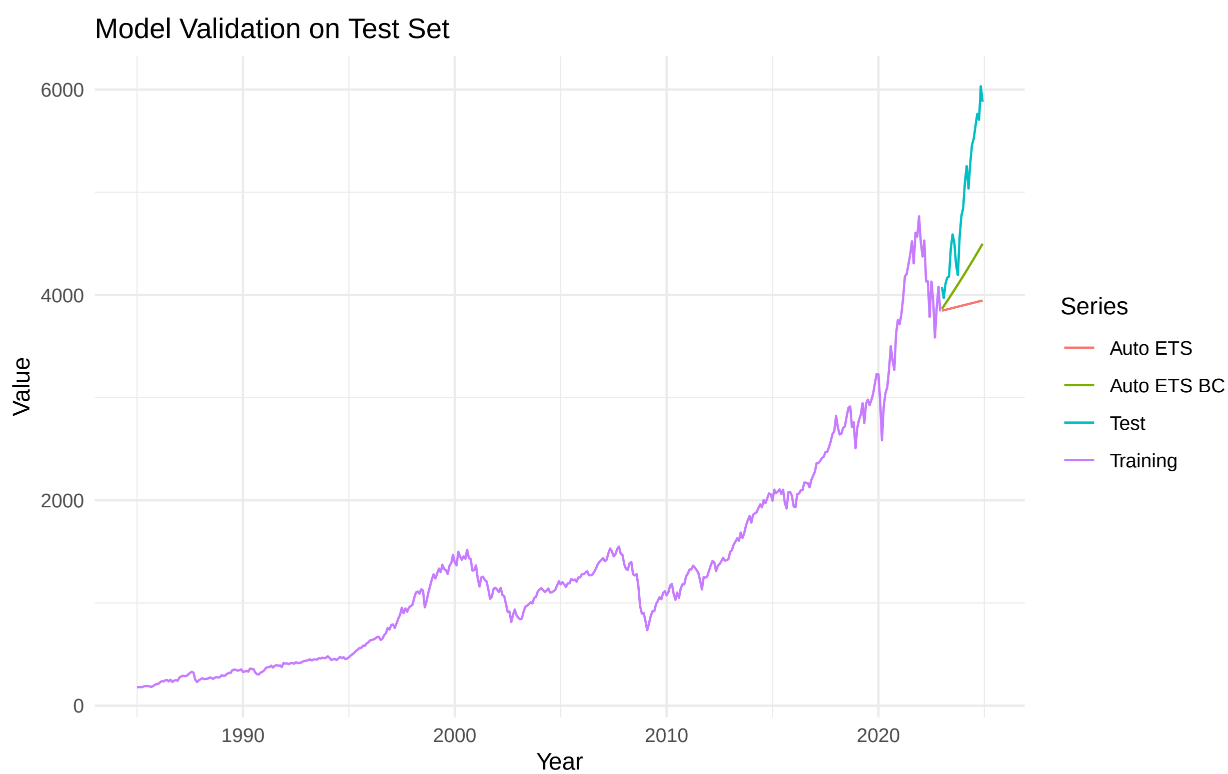
```
##
##  Box-Ljung test
##
## data:  ets_fc_1$residuals
## X-squared = 0.20071, df = 1, p-value = 0.6541

##
##  Box-Ljung test
##
## data:  ets_fc_box_1$residuals
## X-squared = 0.12514, df = 1, p-value = 0.7235
```

Both p-values are well above the 0.05 threshold, meaning we fail to reject the null hypothesis in both cases. This indicates that no significant autocorrelation is present in the residuals of either model. The results validate that the ETS models (both with and without transformation) have effectively captured the patterns in the data, and the residuals behave like random noise.

The Box-Ljung test results confirm that the residuals from both ETS models (transformed and untransformed) are uncorrelated and independent. This strengthens the reliability of these models for time series.

Plotted Best multiplicative and Additive ETS Models



The Auto ETS BC model more accurately follows the upward path of the test data, showing better alignment with actual values. In contrast, the Auto ETS (without transformation) tends to underpredict the sharp growth seen in the test period. The visual difference aligns with the previously shown error metrics, where Auto ETS BC had lower RMSE, MAE, and MAPE values.

The visual validation strongly supports the conclusion that the Auto ETS with Box-Cox transformation is superior to its untransformed counterpart.

5.c. SES,HOLT & HOLT-WINTERS

```
##         Model      RMSE       MAE       MAPE       AIC      AICc
## 1         SES 0.3062882 0.2671077 0.02912532  71.95673  72.00983
## 2        Holt 0.1997665 0.1729042 0.01884935  65.80563  65.93896
## 3      Holt D 0.2846714 0.2476512 0.02700128  71.34136  71.52844
## 4 Holt-Winter 0.2083541 0.1803218 0.01965949 120.45200 121.84926
```

```
## Best model based on RMSE is: Holt
```

Holt performed best across all major metrics (RMSE, MAE, MAPE, AIC), indicating it accurately captures the upward trend without unnecessary complexity. Holt-Winters had slightly higher errors and much higher AIC, suggesting overfitting due to unneeded seasonal components. SES underperformed due to its inability to model the strong trend present in the S&P 500 data. Holt Damped improved on SES but still lagged behind the standard Holt method.

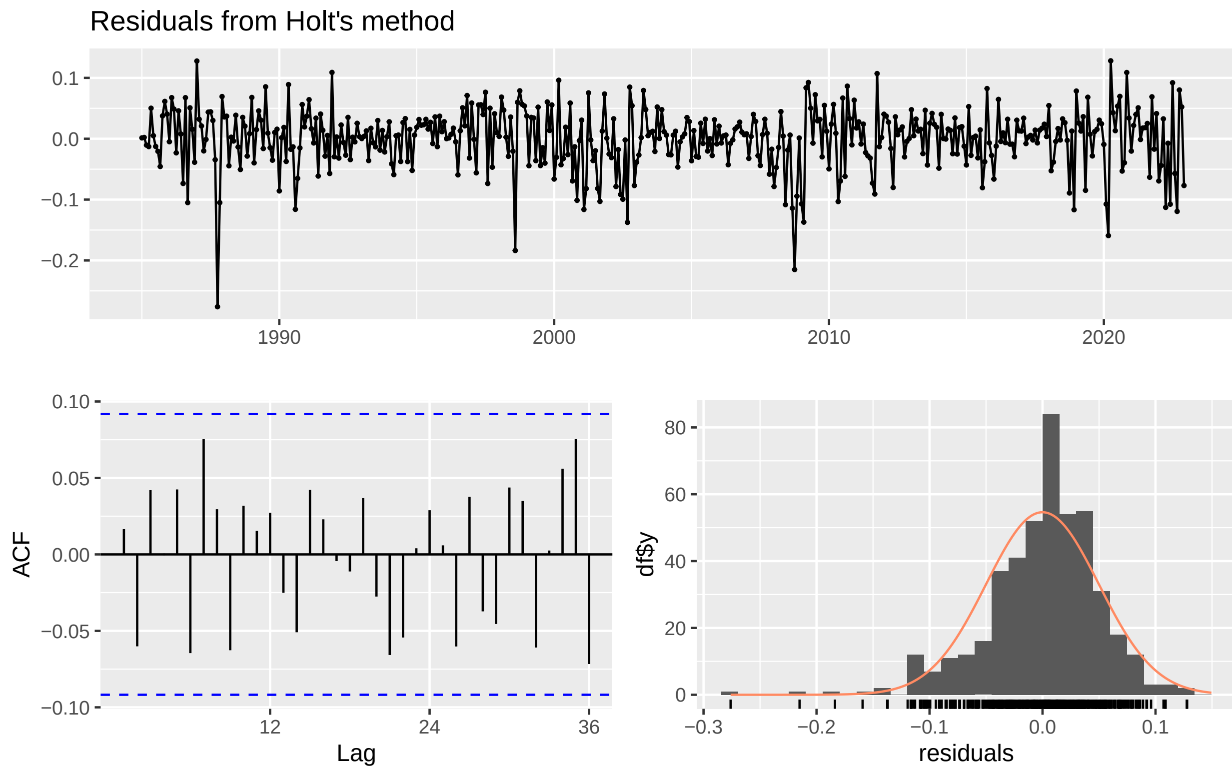
The Holt model is the best among the smoothing techniques tested.

Ljung-Box Test (for autocorrelation)

```
##
##  Box-Ljung test
##
## data:  holt_fc$residuals
## X-squared = 0.12514, df = 1, p-value = 0.7235
```

The p-value is significantly higher than 0.05, so we fail to reject the null hypothesis.This indicates that there is no significant autocorrelation in the residuals of the Holt model.

Checking residuals to see if there's normality and white noise

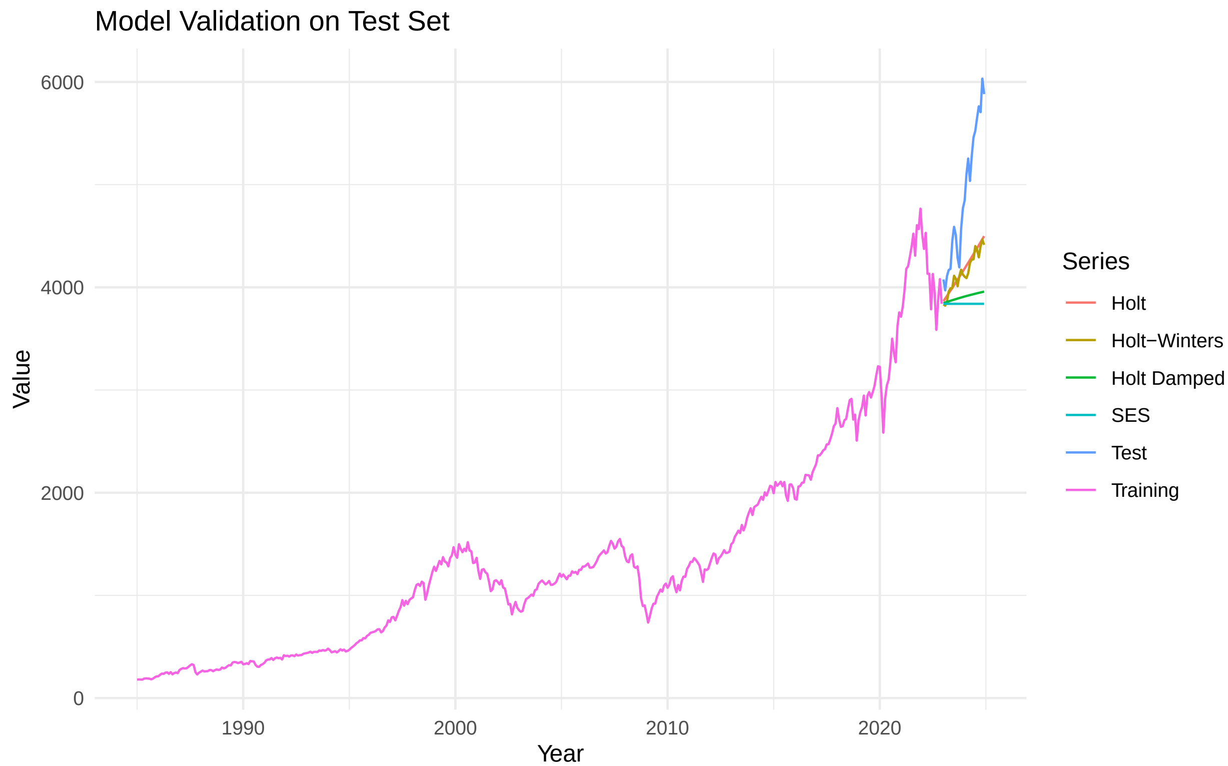



```
##
##  Ljung-Box test
##
## data:  Residuals from Holt's method
## Q* = 18.791, df = 24, p-value = 0.763
##
## Model df: 0.   Total lags used: 24
```

The high p-value indicates that we fail to reject the null hypothesis of no autocorrelation. This confirms that Holt's model residuals behave like white noise — meaning they are random, uncorrelated, and the model has captured the underlying structure in the data.

The diagnostic plots and statistical tests indicate that Holt's method is a well-specified and valid forecasting model. Its residuals meet key assumptions, including independence and zero autocorrelation, making it reliable for short-term forecasting.

Plotted SES,HOLT & HOLT-WINTERS



Holt is the most effective of the four methods, generating forecasts that align well with actual test data. This supports its selection based on the lowest RMSE and AIC. Holt-Winters attempts to model seasonal variation, which appears unnecessary in this case, adding complexity without improved accuracy. SES and Holt Damped are too conservative for the strong trend in the S&P 500, resulting in underperformance.

The visual comparison supports the quantitative evaluation — Holt's method is the most suitable smoothing model for this dataset. Its ability to adapt to trend without overfitting seasonal noise makes it ideal for forecasting the upward momentum of the S&P 500 index.

5.d. ARIMA & SARIMA

Splitting the data for ARIMA and SARIMA models

Check for stationarity using Augmented Dickey-Fuller test

```
##
##  Augmented Dickey-Fuller Test
##
## data:  train_ts_ar
## Dickey-Fuller = -1.6814, Lag order = 7, p-value = 0.7126
## alternative hypothesis: stationary
```

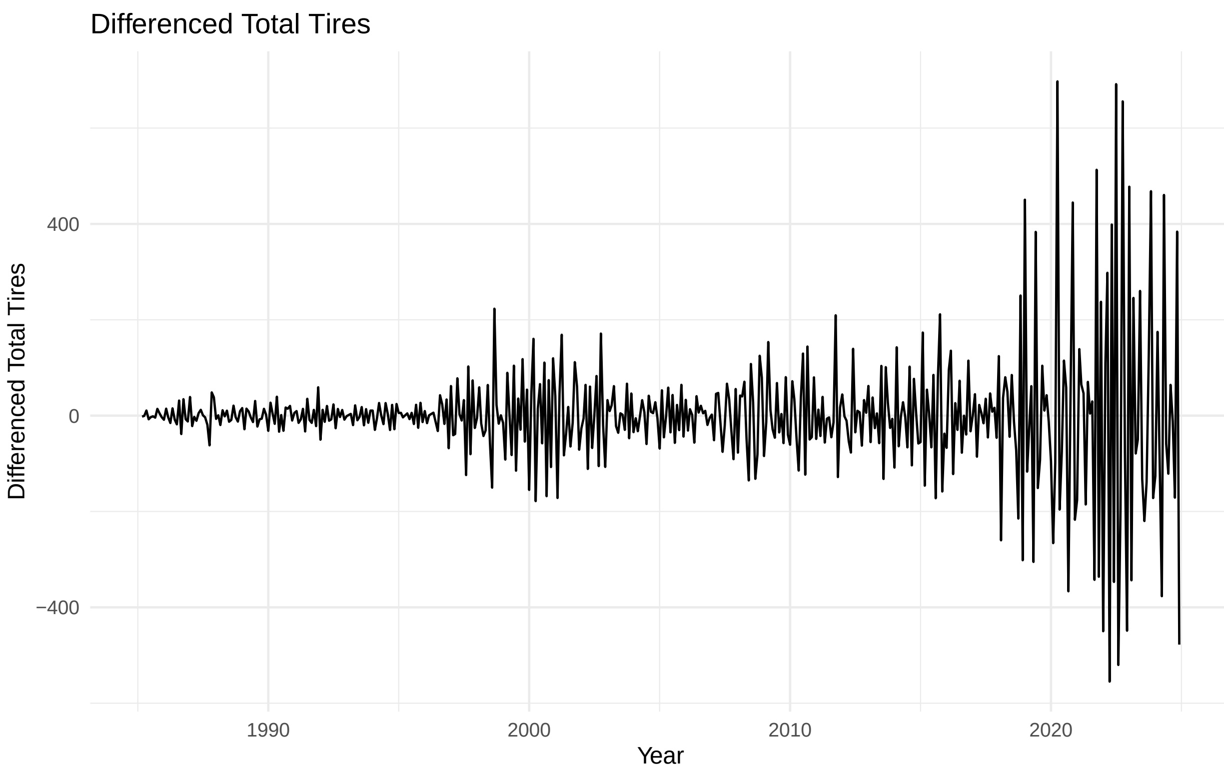
The high p-value (0.7126) means we fail to reject the null hypothesis, which states that the series is non-stationary.

Check for stationarity using Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test

```
##
##  KPSS Test for Level Stationarity
##
## data:  train_ts_ar
## KPSS Level = 5.9325, Truncation lag parameter = 5, p-value = 0.01
```

A low p-value (0.05) leads us to reject the null hypothesis, which means the series is not level stationary. This supports the result from the ADF test, which also indicated non-stationarity.

Plotting the differenced data



The differenced series appears to be stationary with mean reversion around zero, making it a valid input for ARIMA model estimation. A second-order differencing is applied to round of differencing still results in a non-stationary series, which can be confirmed with follow-up stationarity tests.

Check for stationarity using Augmented Dickey-Fuller test for total tires

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff
```

```
## Dickey-Fuller = -13.508, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

The p-value of 0.01 is below the conventional threshold of 0.05, so we reject the null hypothesis. This means the differenced series is now stationary

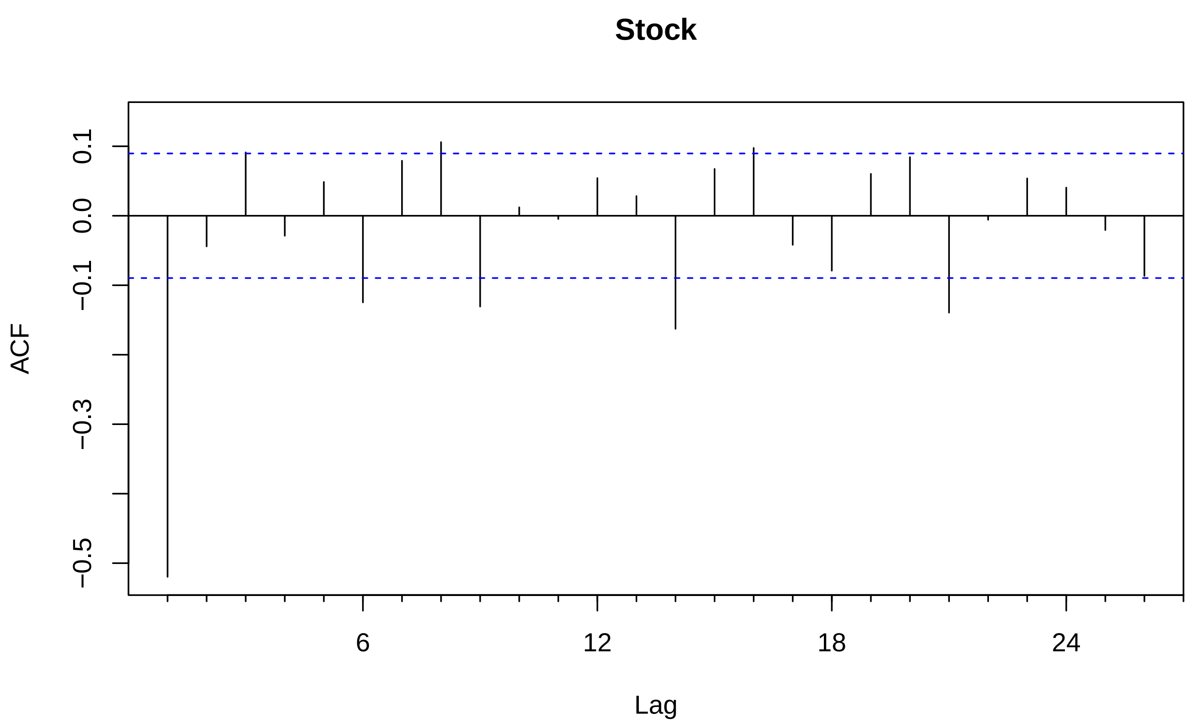
Check for stationarity using Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff
## KPSS Level = 0.013896, Truncation lag parameter = 5, p-value = 0.1
```

The null hypothesis of the KPSS test is that the series is level stationary. A p-value greater than 0.05 means we fail to reject the null hypothesis, indicating that the differenced series is stationary.

The KPSS test validates that the differenced series is level stationary, supporting the findings of the ADF test. With both tests aligned, the time series is now statistically ready for fitting ARIMA and SARIMA models.

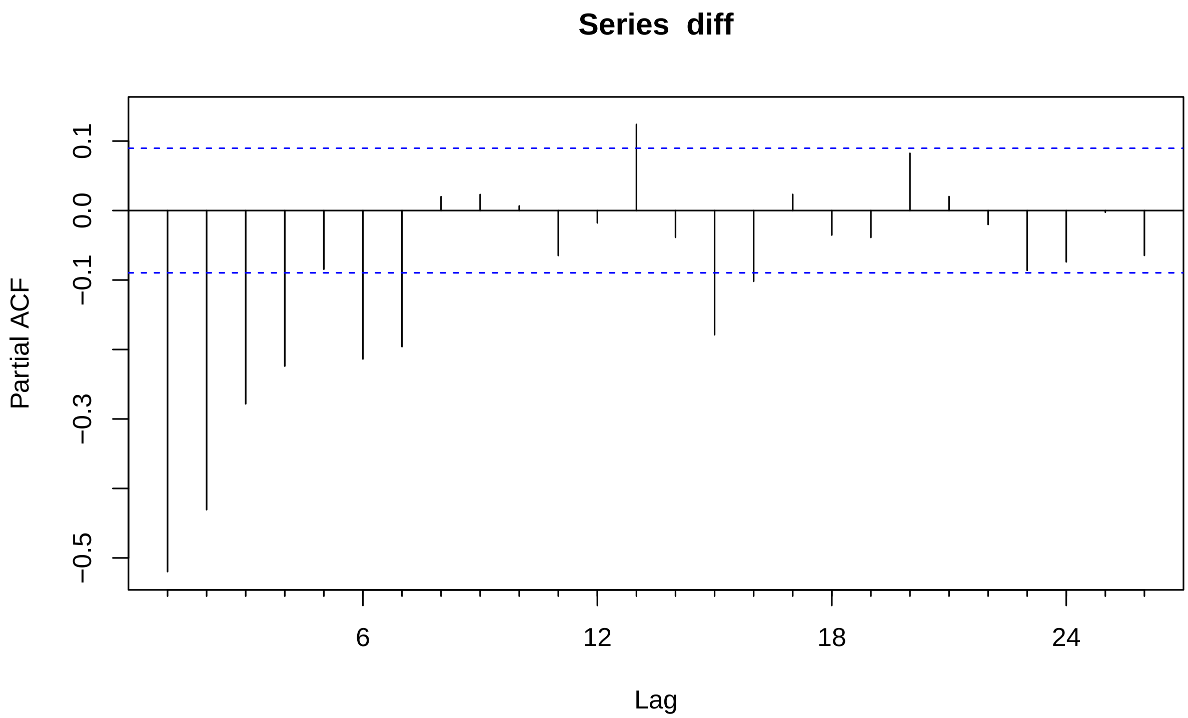
Auto-correlation plot



The strong spike at lag 1 and relative insignificance beyond it supports starting with a simple MA(1) model for the ARIMA process.

This ACF plot supports the inclusion of a Moving Average component (q 1) in the ARIMA model. It also visually reinforces the result of the ADF/KPSS tests, showing the differenced series has no strong autocorrelation structure beyond the first few lags.

Partial auto-correlation plot



The prominent spike at lag 1 and diminishing values afterward is characteristic of an AR(1) or potentially AR(2) process. This pattern supports the inclusion of autoregressive terms in the ARIMA model.

ARIMA models

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,1,2)            with drift         : Inf
##  ARIMA(0,1,0)            with drift         : 5256.992
##  ARIMA(1,1,0)            with drift         : 5252.912
##  ARIMA(0,1,1)            with drift         : 5250.764
##  ARIMA(0,1,0)                               : 5259.726
##  ARIMA(1,1,1)            with drift         : 5253.465
##  ARIMA(0,1,2)            with drift         : 5251.943
##  ARIMA(1,1,2)            with drift         : Inf
##  ARIMA(0,1,1)                               : 5255.279
##
##  Now re-fitting the best model(s) without approximations...
##
```

```
##  ARIMA(0,1,1)            with drift         : 5259.521
##
##  Best model: ARIMA(0,1,1)            with drift
```

```
##    Model      RMSE      MAE      MAPE      AIC     AICc
## 1 Arima1 1091.5960 919.7239 0.1754955 5259.468 5259.521
## 2 Arima2 1019.2807 856.0353 0.1631927 5255.811 5255.864
## 3 Arima3 1006.0892 843.9628 0.1608370 5256.158 5256.247
## 4 Arima4  996.2697 834.7678 0.1590321 5253.658 5253.747
## 5 Arima5  994.3231 831.5694 0.1583290 5255.113 5255.247
```

```
## Best model based on RMSE is: Arima5
```

Arima5 has the lowest RMSE, MAE, and MAPE, making it the most accurate model overall. Although Arima4 has the lowest AIC, its slightly higher RMSE and MAPE suggest a trade-off between simplicity and predictive performance. The improvement across models from Arima1 to Arima5 shows the benefit of iteratively tuning parameters.

Based on the accuracy metrics (RMSE, MAE, MAPE), Arima5 is the best-performing ARIMA model.

SARIMA models

```
##
##  ARIMA(0,1,0)                                : 5268.463
##  ARIMA(0,1,0)            with drift         : 5265.729
##  ARIMA(0,1,0)(0,0,1)[12]                    : 5269.69
##  ARIMA(0,1,0)(0,0,1)[12] with drift         : 5267.398
##  ARIMA(0,1,0)(0,0,2)[12]                    : 5268.975
##  ARIMA(0,1,0)(0,0,2)[12] with drift         : 5267.652
##  ARIMA(0,1,0)(0,0,3)[12]                    : 5261.578
##  ARIMA(0,1,0)(0,0,3)[12] with drift         : 5258.217
##  ARIMA(0,1,0)(1,0,0)[12]                    : 5269.573
##  ARIMA(0,1,0)(1,0,0)[12] with drift         : 5267.355
##  ARIMA(0,1,0)(1,0,1)[12]                    : 5271.384
##  ARIMA(0,1,0)(1,0,1)[12] with drift         : 5269.299
##  ARIMA(0,1,0)(1,0,2)[12]                    : 5261.734
##  ARIMA(0,1,0)(1,0,2)[12] with drift         : 5260.443
##  ARIMA(0,1,0)(1,0,3)[12]                    : 5259.298
##  ARIMA(0,1,0)(1,0,3)[12] with drift         : 5256.379
##  ARIMA(0,1,0)(2,0,0)[12]                    : 5269.991
##  ARIMA(0,1,0)(2,0,0)[12] with drift         : 5268.33
##  ARIMA(0,1,0)(2,0,1)[12]                    : 5264.504
##  ARIMA(0,1,0)(2,0,1)[12] with drift         : 5263.004
##  ARIMA(0,1,0)(2,0,2)[12]                    : 5259.412
##  ARIMA(0,1,0)(2,0,2)[12] with drift         : 5257.374
##  ARIMA(0,1,0)(2,0,3)[12]                    : 5261.125
##  ARIMA(0,1,0)(2,0,3)[12] with drift         : 5257.862
##  ARIMA(0,1,0)(3,0,0)[12]                    : 5258.426
##  ARIMA(0,1,0)(3,0,0)[12] with drift         : 5254.752
##  ARIMA(0,1,0)(3,0,1)[12]                    : 5259.624
##  ARIMA(0,1,0)(3,0,1)[12] with drift         : 5256.462
##  ARIMA(0,1,0)(3,0,2)[12]                    : 5261.279
```

```
##  ARIMA(0,1,0)(3,0,2)[12] with drift         : 5257.689
##  ARIMA(0,1,1)                               : 5264.032
##  ARIMA(0,1,1)            with drift         : 5259.521
##  ARIMA(0,1,1)(0,0,1)[12]                    : 5265.534
##  ARIMA(0,1,1)(0,0,1)[12] with drift         : 5261.472
##  ARIMA(0,1,1)(0,0,2)[12]                    : 5264.571
##  ARIMA(0,1,1)(0,0,2)[12] with drift         : 5261.735
##  ARIMA(0,1,1)(0,0,3)[12]                    : 5260.091
##  ARIMA(0,1,1)(0,0,3)[12] with drift         : 5255.537
##  ARIMA(0,1,1)(1,0,0)[12]                    : 5265.447
##  ARIMA(0,1,1)(1,0,0)[12] with drift         : 5261.461
##  ARIMA(0,1,1)(1,0,1)[12]                    : 5267.151
##  ARIMA(0,1,1)(1,0,1)[12] with drift         : 5263.416
##  ARIMA(0,1,1)(1,0,2)[12]                    : 5258.596
##  ARIMA(0,1,1)(1,0,2)[12] with drift         : 5255.892
##  ARIMA(0,1,1)(1,0,3)[12]                    : 5257.566
##  ARIMA(0,1,1)(1,0,3)[12] with drift         : 5253.416
##  ARIMA(0,1,1)(2,0,0)[12]                    : 5265.179
##  ARIMA(0,1,1)(2,0,0)[12] with drift         : 5262.019
##  ARIMA(0,1,1)(2,0,1)[12]                    : 5260.483
##  ARIMA(0,1,1)(2,0,1)[12] with drift         : 5257.487
##  ARIMA(0,1,1)(2,0,2)[12]                    : 5257.089
##  ARIMA(0,1,1)(2,0,2)[12] with drift         : 5253.602
##  ARIMA(0,1,1)(3,0,0)[12]                    : 5257.598
##  ARIMA(0,1,1)(3,0,0)[12] with drift         : 5252.795
##  ARIMA(0,1,1)(3,0,1)[12]                    : 5258.313
##  ARIMA(0,1,1)(3,0,1)[12] with drift         : 5254.074
##  ARIMA(0,1,2)                               : 5265.549
##  ARIMA(0,1,2)            with drift         : 5260.701
##  ARIMA(0,1,2)(0,0,1)[12]                    : 5267.101
##  ARIMA(0,1,2)(0,0,1)[12] with drift         : 5262.693
##  ARIMA(0,1,2)(0,0,2)[12]                    : 5266.43
##  ARIMA(0,1,2)(0,0,2)[12] with drift         : 5263.347
##  ARIMA(0,1,2)(0,0,3)[12]                    : 5261.312
##  ARIMA(0,1,2)(0,0,3)[12] with drift         : 5255.916
##  ARIMA(0,1,2)(1,0,0)[12]                    : 5267.025
##  ARIMA(0,1,2)(1,0,0)[12] with drift         : 5262.687
##  ARIMA(0,1,2)(1,0,1)[12]                    : 5268.804
##  ARIMA(0,1,2)(1,0,1)[12] with drift         : 5264.693
##  ARIMA(0,1,2)(1,0,2)[12]                    : 5260.18
##  ARIMA(0,1,2)(1,0,2)[12] with drift         : 5257.231
##  ARIMA(0,1,2)(2,0,0)[12]                    : 5267.018
##  ARIMA(0,1,2)(2,0,0)[12] with drift         : 5263.579
##  ARIMA(0,1,2)(2,0,1)[12]                    : 5262.164
##  ARIMA(0,1,2)(2,0,1)[12] with drift         : 5258.888
##  ARIMA(0,1,2)(3,0,0)[12]                    : 5258.698
##  ARIMA(0,1,2)(3,0,0)[12] with drift         : 5253.032
##  ARIMA(0,1,3)                               : 5260.15
##  ARIMA(0,1,3)            with drift         : 5256.406
##  ARIMA(0,1,3)(0,0,1)[12]                    : 5262.133
##  ARIMA(0,1,3)(0,0,1)[12] with drift         : 5258.439
##  ARIMA(0,1,3)(0,0,2)[12]                    : 5261.592
##  ARIMA(0,1,3)(0,0,2)[12] with drift         : 5259.042
##  ARIMA(0,1,3)(1,0,0)[12]                    : 5262.122
```

```
##  ARIMA(0,1,3)(1,0,0)[12] with drift         : 5258.437
##  ARIMA(0,1,3)(1,0,1)[12]                    : 5264.031
##  ARIMA(0,1,3)(1,0,1)[12] with drift         : 5255.501
##  ARIMA(0,1,3)(2,0,0)[12]                    : 5261.829
##  ARIMA(0,1,3)(2,0,0)[12] with drift         : 5259.002
##  ARIMA(1,1,0)                               : 5264.767
##  ARIMA(1,1,0)            with drift         : 5260.67
##  ARIMA(1,1,0)(0,0,1)[12]                    : 5266.21
##  ARIMA(1,1,0)(0,0,1)[12] with drift         : 5262.565
##  ARIMA(1,1,0)(0,0,2)[12]                    : 5265.122
##  ARIMA(1,1,0)(0,0,2)[12] with drift         : 5262.64
##  ARIMA(1,1,0)(0,0,3)[12]                    : 5260.542
##  ARIMA(1,1,0)(0,0,3)[12] with drift         : 5256.334
##  ARIMA(1,1,0)(1,0,0)[12]                    : 5266.111
##  ARIMA(1,1,0)(1,0,0)[12] with drift         : 5262.545
##  ARIMA(1,1,0)(1,0,1)[12]                    : 5267.801
##  ARIMA(1,1,0)(1,0,1)[12] with drift         : 5264.477
##  ARIMA(1,1,0)(1,0,2)[12]                    : 5259.158
##  ARIMA(1,1,0)(1,0,2)[12] with drift         : 5256.763
##  ARIMA(1,1,0)(1,0,3)[12]                    : 5258.105
##  ARIMA(1,1,0)(1,0,3)[12] with drift         : 5254.317
##  ARIMA(1,1,0)(2,0,0)[12]                    : 5265.796
##  ARIMA(1,1,0)(2,0,0)[12] with drift         : 5263.013
##  ARIMA(1,1,0)(2,0,1)[12]                    : 5261.106
##  ARIMA(1,1,0)(2,0,1)[12] with drift         : 5258.461
##  ARIMA(1,1,0)(2,0,2)[12]                    : Inf
##  ARIMA(1,1,0)(2,0,2)[12] with drift         : Inf
##  ARIMA(1,1,0)(3,0,0)[12]                    : Inf
##  ARIMA(1,1,0)(3,0,0)[12] with drift         : Inf
##  ARIMA(1,1,0)(3,0,1)[12]                    : Inf
##  ARIMA(1,1,0)(3,0,1)[12] with drift         : Inf
##  ARIMA(1,1,1)                               : 5265.885
##  ARIMA(1,1,1)            with drift         : 5261.234
##  ARIMA(1,1,1)(0,0,1)[12]                    : 5267.409
##  ARIMA(1,1,1)(0,0,1)[12] with drift         : 5263.206
##  ARIMA(1,1,1)(0,0,2)[12]                    : 5266.549
##  ARIMA(1,1,1)(0,0,2)[12] with drift         : 5263.622
##  ARIMA(1,1,1)(0,0,3)[12]                    : 5261.908
##  ARIMA(1,1,1)(0,0,3)[12] with drift         : 5257.081
##  ARIMA(1,1,1)(1,0,0)[12]                    : 5267.325
##  ARIMA(1,1,1)(1,0,0)[12] with drift         : 5263.197
##  ARIMA(1,1,1)(1,0,1)[12]                    : 5269.06
##  ARIMA(1,1,1)(1,0,1)[12] with drift         : 5265.177
##  ARIMA(1,1,1)(1,0,2)[12]                    : 5260.491
##  ARIMA(1,1,1)(1,0,2)[12] with drift         : 5257.691
##  ARIMA(1,1,1)(2,0,0)[12]                    : 5267.151
##  ARIMA(1,1,1)(2,0,0)[12] with drift         : 5263.886
##  ARIMA(1,1,1)(2,0,1)[12]                    : 5262.407
##  ARIMA(1,1,1)(2,0,1)[12] with drift         : 5259.305
##  ARIMA(1,1,1)(3,0,0)[12]                    : Inf
##  ARIMA(1,1,1)(3,0,0)[12] with drift         : 5254.326
##  ARIMA(1,1,2)                               : Inf
##  ARIMA(1,1,2)            with drift         : Inf
##  ARIMA(1,1,2)(0,0,1)[12]                    : Inf
```

```
## ARIMA(1,1,2)(0,0,1)[12] with drift         : Inf
## ARIMA(1,1,2)(0,0,2)[12]                    : Inf
## ARIMA(1,1,2)(0,0,2)[12] with drift         : Inf
## ARIMA(1,1,2)(1,0,0)[12]                    : Inf
## ARIMA(1,1,2)(1,0,0)[12] with drift         : Inf
## ARIMA(1,1,2)(1,0,1)[12]                    : Inf
## ARIMA(1,1,2)(1,0,1)[12] with drift         : Inf
## ARIMA(1,1,2)(2,0,0)[12]                    : Inf
## ARIMA(1,1,2)(2,0,0)[12] with drift         : Inf
## ARIMA(1,1,3)                               : 5256.302
## ARIMA(1,1,3)            with drift         : 5254.974
## ARIMA(1,1,3)(0,0,1)[12]                    : 5258.353
## ARIMA(1,1,3)(0,0,1)[12] with drift         : 5256.972
## ARIMA(1,1,3)(1,0,0)[12]                    : 5258.352
## ARIMA(1,1,3)(1,0,0)[12] with drift         : 5256.963
## ARIMA(2,1,0)                               : 5264.144
## ARIMA(2,1,0)            with drift         : 5258.929
## ARIMA(2,1,0)(0,0,1)[12]                    : 5265.817
## ARIMA(2,1,0)(0,0,1)[12] with drift         : 5260.966
## ARIMA(2,1,0)(0,0,2)[12]                    : 5265.444
## ARIMA(2,1,0)(0,0,2)[12] with drift         : 5261.908
## ARIMA(2,1,0)(0,0,3)[12]                    : 5260.081
## ARIMA(2,1,0)(0,0,3)[12] with drift         : 5254.026
## ARIMA(2,1,0)(1,0,0)[12]                    : 5265.763
## ARIMA(2,1,0)(1,0,0)[12] with drift         : 5260.965
## ARIMA(2,1,0)(1,0,1)[12]                    : 5267.593
## ARIMA(2,1,0)(1,0,1)[12] with drift         : 5263.011
## ARIMA(2,1,0)(1,0,2)[12]                    : 5259.025
## ARIMA(2,1,0)(1,0,2)[12] with drift         : 5255.695
## ARIMA(2,1,0)(2,0,0)[12]                    : 5265.932
## ARIMA(2,1,0)(2,0,0)[12] with drift         : 5262.026
## ARIMA(2,1,0)(2,0,1)[12]                    : 5260.951
## ARIMA(2,1,0)(2,0,1)[12] with drift         : 5257.237
## ARIMA(2,1,0)(3,0,0)[12]                    : 5257.447
## ARIMA(2,1,0)(3,0,0)[12] with drift         : 5251.105
## ARIMA(2,1,1)                               : Inf
## ARIMA(2,1,1)            with drift         : 5260.36
## ARIMA(2,1,1)(0,0,1)[12]                    : Inf
## ARIMA(2,1,1)(0,0,1)[12] with drift         : 5262.408
## ARIMA(2,1,1)(0,0,2)[12]                    : Inf
## ARIMA(2,1,1)(0,0,2)[12] with drift         : Inf
## ARIMA(2,1,1)(1,0,0)[12]                    : Inf
## ARIMA(2,1,1)(1,0,0)[12] with drift         : 5262.407
## ARIMA(2,1,1)(1,0,1)[12]                    : Inf
## ARIMA(2,1,1)(1,0,1)[12] with drift         : Inf
## ARIMA(2,1,1)(2,0,0)[12]                    : Inf
## ARIMA(2,1,1)(2,0,0)[12] with drift         : 5263.5
## ARIMA(2,1,2)                               : 5232.902
## ARIMA(2,1,2)            with drift         : 5228.588
## ARIMA(2,1,2)(0,0,1)[12]                    : 5234.863
## ARIMA(2,1,2)(0,0,1)[12] with drift         : 5230.088
## ARIMA(2,1,2)(1,0,0)[12]                    : 5234.848
## ARIMA(2,1,2)(1,0,0)[12] with drift         : 5230.025
## ARIMA(2,1,3)                               : Inf
```

```
##  ARIMA(2,1,3)            with drift         : 5257.037
##  ARIMA(3,1,0)                               : 5263.455
##  ARIMA(3,1,0)            with drift         : 5259.157
##  ARIMA(3,1,0)(0,0,1)[12]                    : 5265.251
##  ARIMA(3,1,0)(0,0,1)[12] with drift         : 5261.207
##  ARIMA(3,1,0)(0,0,2)[12]                    : 5265.018
##  ARIMA(3,1,0)(0,0,2)[12] with drift         : 5262.123
##  ARIMA(3,1,0)(1,0,0)[12]                    : 5265.215
##  ARIMA(3,1,0)(1,0,0)[12] with drift         : 5261.207
##  ARIMA(3,1,0)(1,0,1)[12]                    : 5267.095
##  ARIMA(3,1,0)(1,0,1)[12] with drift         : 5263.253
##  ARIMA(3,1,0)(2,0,0)[12]                    : 5265.448
##  ARIMA(3,1,0)(2,0,0)[12] with drift         : 5262.231
##  ARIMA(3,1,1)                               : 5256.908
##  ARIMA(3,1,1)            with drift         : 5255.58
##  ARIMA(3,1,1)(0,0,1)[12]                    : 5258.948
##  ARIMA(3,1,1)(0,0,1)[12] with drift         : 5257.573
##  ARIMA(3,1,1)(1,0,0)[12]                    : 5258.946
##  ARIMA(3,1,1)(1,0,0)[12] with drift         : 5257.564
##  ARIMA(3,1,2)                               : 5258.225
##  ARIMA(3,1,2) with drift         : Inf
##
##
##
##  Best model: ARIMA(2,1,2)            with drift
```

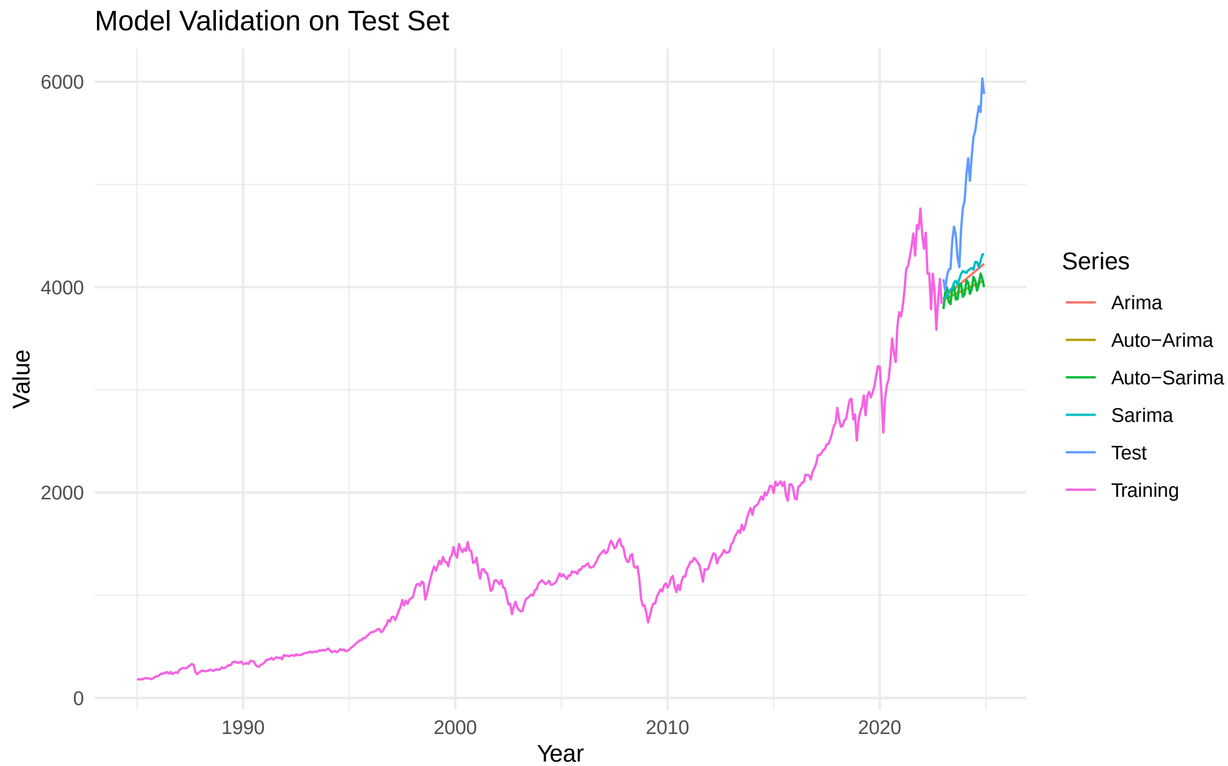
```
##     Model      RMSE       MAE      MAPE      AIC     AICc
## 1 Sarima1 1087.3810  913.3714 0.1742182 5228.400 5228.588
## 2 Sarima2 1045.7900  858.7841 0.1626644 5154.676 5155.008
## 3 Sarima3  941.0985  779.6849 0.1480952 5172.234 5172.651
## 4 Sarima4 1288.6044 1060.0199 0.2008390 5251.757 5252.080
## 5 Sarima5 1260.3740 1038.4539 0.1968478 5253.196 5253.519
## 6 Sarima6 1132.8677  931.1653 0.1764152 5154.026 5154.442
```

```
## Best model based on RMSE is: Sarima3
```

Sarima3 is the best-performing model across all key metrics: lowest RMSE (941.10), MAE (779.68), and MAPE (14.81%). This indicates that it delivers the most accurate forecasts. While Sarima2 and Sarima6 have lower AIC values, their forecast errors are notably higher, suggesting a trade-off between model fit and predictive accuracy.

Sarima3 is the optimal SARIMA model, outperforming all other seasonal models in terms of both forecast accuracy and information criteria.

Plotted Arima and Sarima



Sarima3 (Green) and Auto-Sarima (Teal) align more closely with the actual test values than the non-seasonal ARIMA models. Auto-Arima performs better than the manually chosen ARIMA in this case, but still falls short compared to SARIMA. SARIMA-based models clearly capture the trend and seasonal behavior better, which is consistent with the error metrics presented earlier.

The visual validation confirms that SARIMA models—particularly Sarima3 and Auto-Sarima—provide more accurate forecasts

6. MODEL COMPARISION

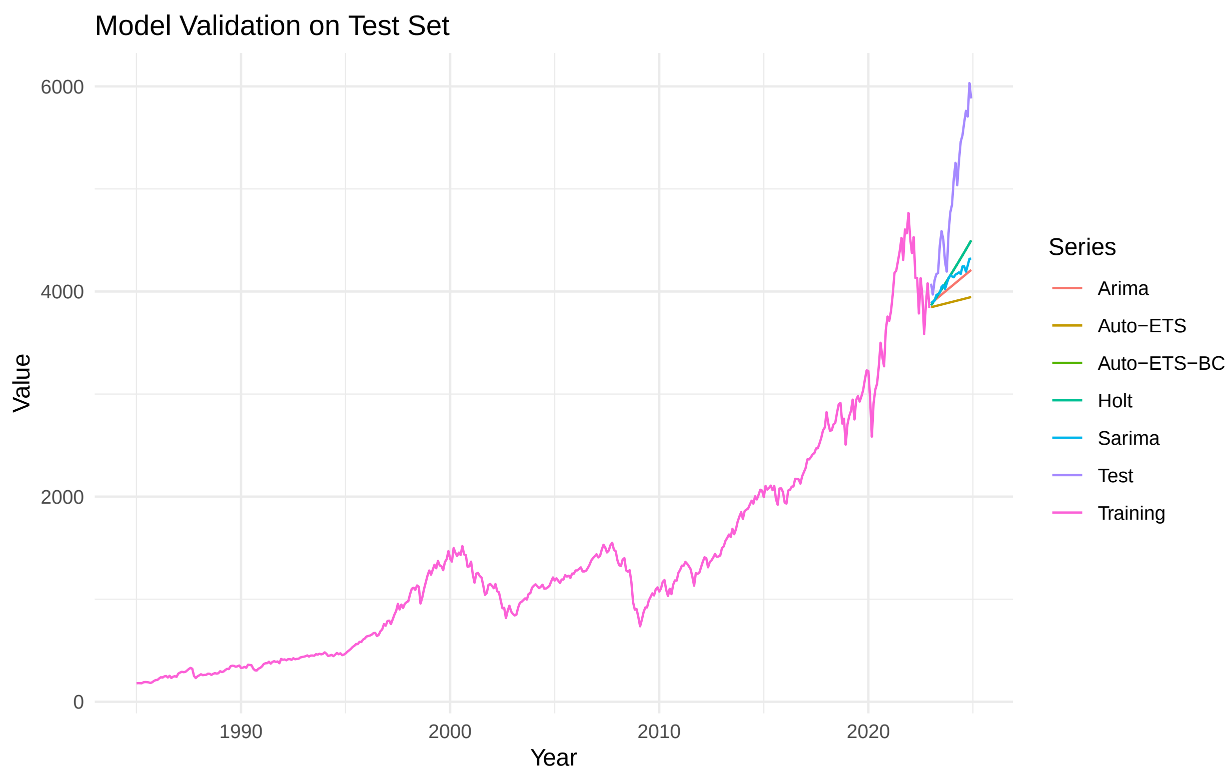
Compare forecast accuracy of all top models

```
##    Model      RMSE      MAPE
## 1    ETS 1168.9690 0.1904849
## 2 ETS_BC  851.7577 0.1369174
## 3   Holt  851.8130 0.1369268
## 4  ARIMA  994.3231 0.1583290
## 5 SARIMA  941.0985 0.1480952

## Best model based on RMSE is: ETS_BC
## Model = AAN
```

ETS_BC (ETS with Box-Cox Transformation) and Holt produced the lowest RMSE and MAPE, indicating that they delivered the most accurate forecasts overall. ETS_BC offers the best overall accuracy, slightly outperforming Holt, with the lowest RMSE and MAPE. Holt remains a close and simple alternative, also delivering robust results. SARIMA is a strong seasonal model but falls slightly behind ETS_BC and Holt in this dataset. Thus, the ETS model with Box-Cox transformation is recommended as the final model.

Plotted all top models



Auto-ETS-BC (Green Line) follows the test data more closely than any other model, staying closest to the steep upward trajectory of the index.

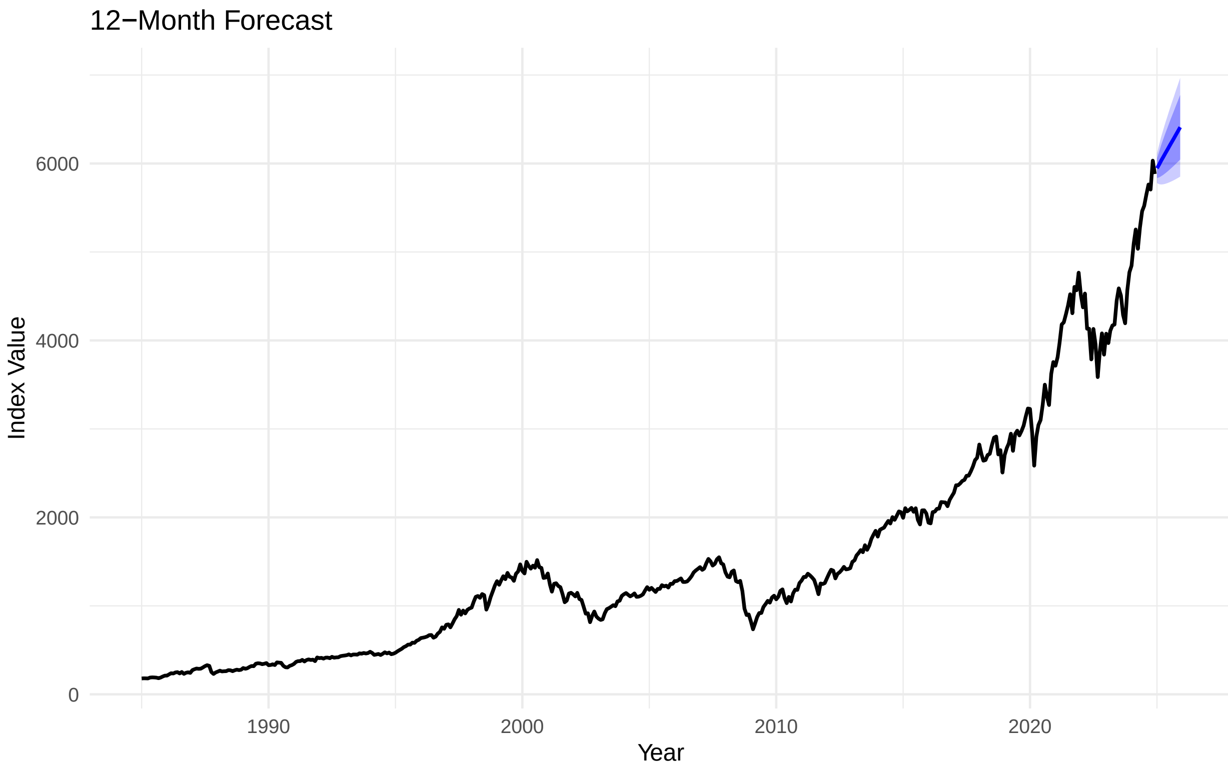
Holt (Yellow Line) is nearly tied in performance, slightly diverging toward the end.

Sarima (Teal Line) captures some trend but remains underpredictive overall.

ARIMA and Auto-ETS (Brown and Orange Lines) perform the worst, showing a lag behind actual market growth.

7. FORECASTING

Forecasting for next 1 year



The forecast shows a continued upward trajectory, consistent with the long-term trend observed in the S&P 500. The tight prediction intervals indicate high model confidence in the near-term outlook, a hallmark of the ETS_BC model's superior fit and well-behaved residuals.

This forecast can serve as a valuable decision-support tool for investors, analysts, or policy-makers needing a reliable projection of market momentum over the next 12 months.

8. CONCLUSION AND RECOMMENDATION

Conclusion

This time series analysis of the S&P 500 index (1985–2024) applied multiple forecasting models, including ETS, Holt's method, ARIMA, and SARIMA. Each model was rigorously tested using statistical diagnostics (ADF, KPSS, Ljung-Box, Shapiro-Wilk) and evaluated with RMSE, MAPE, and AIC criteria. The models were also validated on a holdout test set.

The best-performing model was the Auto ETS with Box-Cox transformation (ETS_BC), which achieved the lowest RMSE (851.76) and MAPE (13.69%), outperforming all other methods. The model's residuals showed no autocorrelation and behaved like white noise, confirming its reliability. Holt's linear model was a close second, while SARIMA and ARIMA underperformed comparatively.

A 12-month forecast using the ETS(AAN) model projected a continued upward trend in the index, supported by narrow prediction intervals—demonstrating high model confidence.

Recommendation

Adopt the ETS_BC model for short-term S&P 500 forecasting, as it balances accuracy, simplicity, and interpretability.

Avoid overly complex seasonal models like SARIMA unless strong seasonal patterns are present.

Update forecasts regularly to reflect new market data and ensure continued model validity.

Use prediction intervals alongside point estimates to inform risk-aware financial decisions.

9. REFERENCES

- Hyndman, R. J., & Athanasopoulos, G. (2018). "Forecasting: Principles and Practice".
- Time Series Coursework Material.