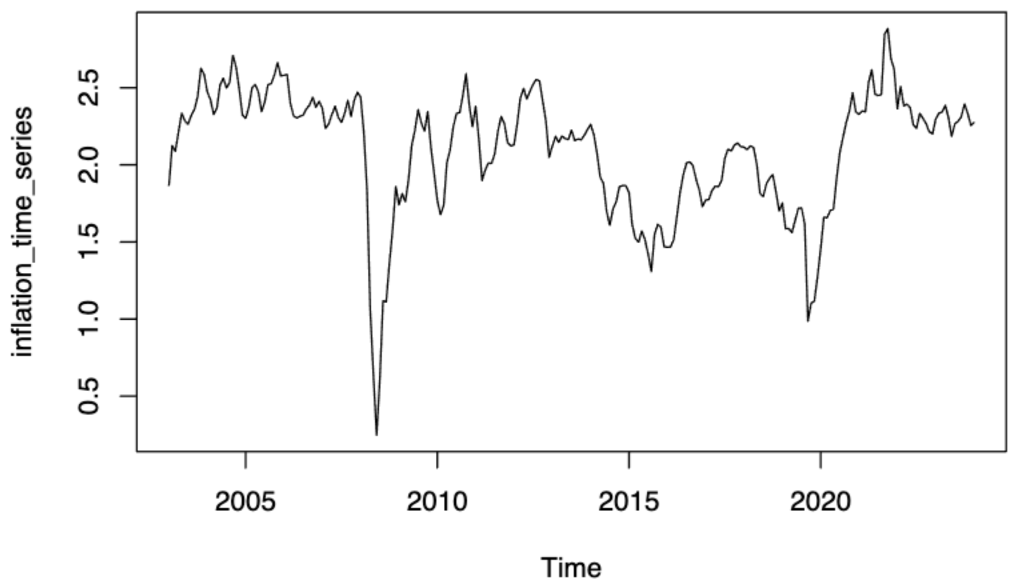
Inflation Rate Time Series Analysis

Hunter Grigsby

Time Series Plot

- **resources:** https://fred.stlouisfed.org/series/T10YIE#0.
- **data:** inflation_time_series

```
library(tseries)
library(vars)
library(readr)
mydata = read.csv("T10YIE.csv")
inflation_time_series <- ts(mydata$T10YIE, frequency = 12, start = c(2003, 1))
plot(inflation_time_series)
```

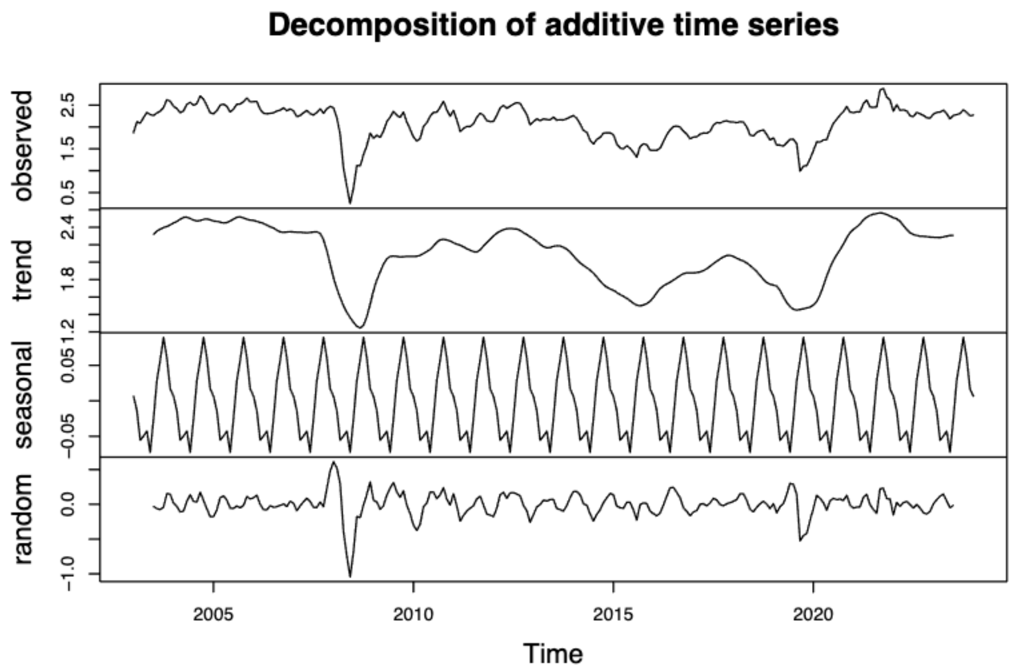


-**Q1:** Graphs shows a long-term trend with periods of rising and falling inflation rates over time. Significant dips around 2008 and 2020 due to financial crisis and COVID-19 pandemic.

-**Q2:** The times series graph appears non-stationary because there is a clear changing trend over time rather than reverting to a constant mean or variance. Non-stationary is also supported by the sharp changes during economic events, indicating that external influences affect the mean and variance.

Decomposition Plot

```
mydata = diff(inflation_time_series, 1)
decomposed_data = decompose(inflation_time_series)
plot(decomposed_data)
```

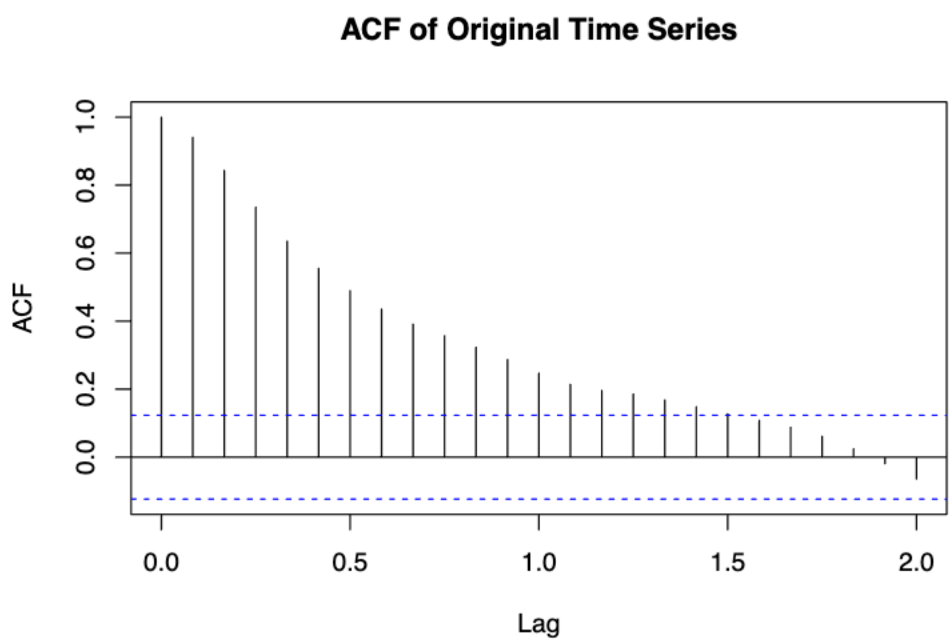


-**Q1:** The trend component shows long-term changes in inflation expectations. It highlights periods of increasing inflation (e.g., after 2015) and declining inflation (e.g., around 2008 and 2020). These trends align with major economic events like the 2008 financial crisis and the COVID-19 pandemic.

-**Q2:** The seasonal component is moderate but consistent, showing regular, repeating fluctuations. This indicates that inflation expectations have some recurring annual patterns, although the magnitude of these seasonal effects is relatively small compared to the trend and random components.

ACF Plot

```
acf(inflation_time_series, main = "ACF of Original Time Series")
```



-**Q1:** The autocorrelation function (ACF) shows significant correlations at multiple lags, which gradually decrease rather than dropping off quickly. This behavior suggests that the time series is non-stationary, as the correlations persist over time instead of diminishing rapidly as they would in a stationary series. This indicates that the series may need further transformation, such as difference or trending, to make it stationary. Non-stationary could affect the accuracy and applicability of time series models like ARIMA, which assume stationary.

-**Q2:** There is no clear evidence of a strong seasonal pattern in the ACF. A seasonal pattern would typically appear as distinct spikes at regular intervals (e.g., yearly or monthly lags), but these are not present in the plot. Instead, the ACF shows a gradual decay, indicating that the series is dominated by long-term trend components rather than consistent seasonal cycles. This further emphasizes the need to focus on the trend when analyzing and forecasting this time series.

ADF Test

```
adf_test <- adf.test(inflation_time_series)
print(adf_test)
```

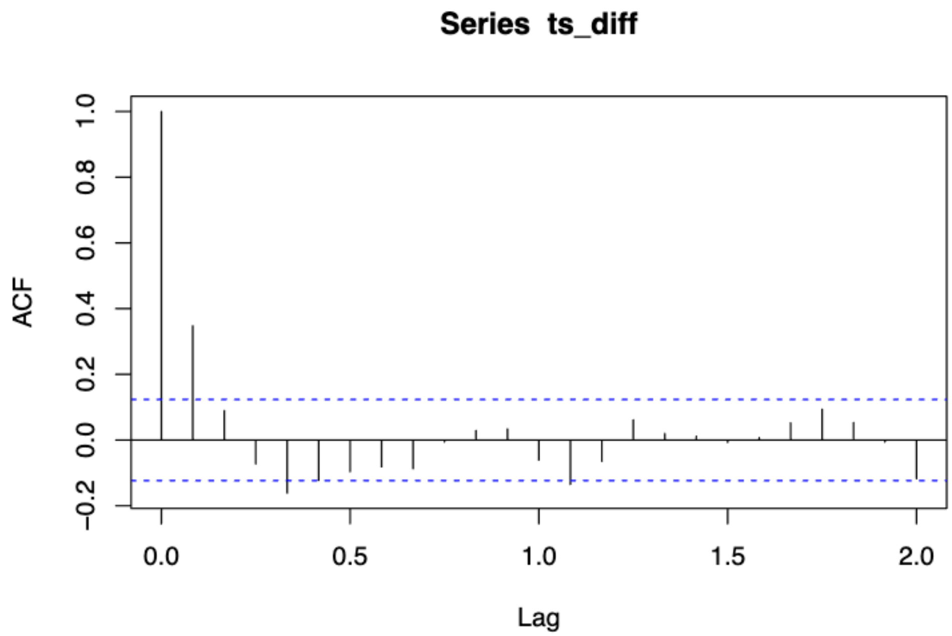
```
##
##  Augmented Dickey-Fuller Test
##
## data:  inflation_time_series
## Dickey-Fuller = -3.1828, Lag order = 6, p-value = 0.0912
## alternative hypothesis: stationary
```

-**Q1:** The p-value from the Augmented Dickey-Fuller (ADF) test is 0.0912. Since this p-value is greater than the common significance level of 0.05, we cannot reject the null hypothesis. The null hypothesis assumes that the time series is non-stationary.

-**Q2:** This result indicates that the time series is likely non-stationary. The presence of a trend or other time-dependent structures is supported by this outcome, and the series may require transformations, such as differencing, to achieve stationarity for accurate modeling and forecasting.

Differencing -> ADF Plot

```
ts_diff <- diff(inflation_time_series, 1)
plot(acf(ts_diff))
```




```r
ts1 = diff(inflation_time_series, 1)
adf.test(ts1)
```

```
## Warning in adf.test(ts1): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  ts1
## Dickey-Fuller = -6.9734, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
ts2 = diff(inflation_time_series, 2)
adf.test(ts2)
```

```
## Warning in adf.test(ts2): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  ts2
## Dickey-Fuller = -6.418, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

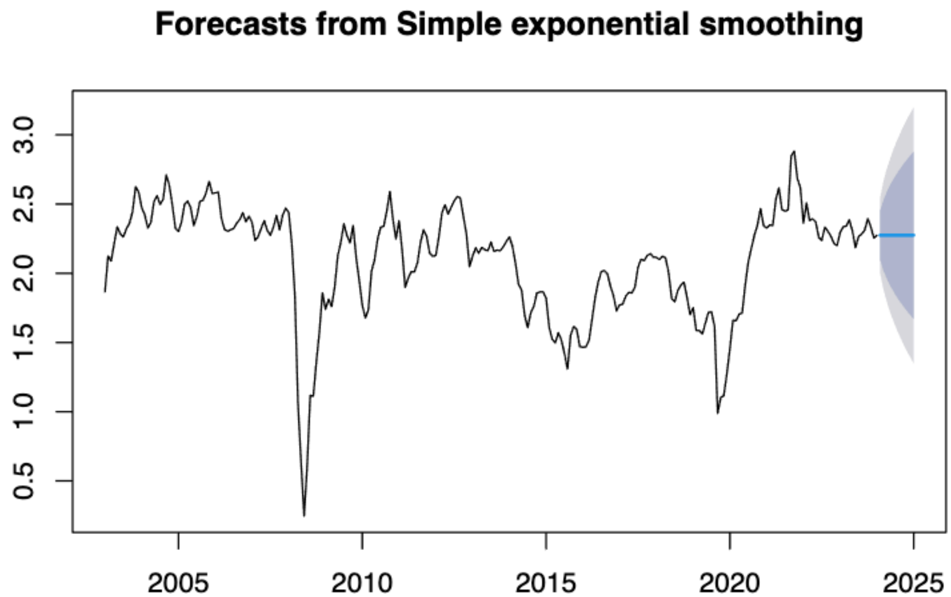
-**Q1:** The differenced series removes the trend seen in the original series, resulting in more stable fluctuations around a constant mean. This transformation helps eliminate non-stationary behavior caused by long-term trends, making the series appear more stationary.

-**Q2:** The ACF plot of the differenced series shows a rapid decline in auto correlations, with most lags falling within the confidence bounds after the first few lags. This behavior indicates that the series is now stationary. The results of the Augmented Dickey-Fuller (ADF) test confirm this, with a p-value of 0.01, allowing us to reject the null hypothesis of non-stationary. This demonstrates that differencing successfully addressed the non-stationary in the original series.

Simple Exponential Smoothing

```r
library(tseries)
library(forecast)
library(ggplot2)

ses_model <- ses(inflation_time_series, h = 12)
plot(forecast(ses_model))
```




```r
forecast(ses_model)
```

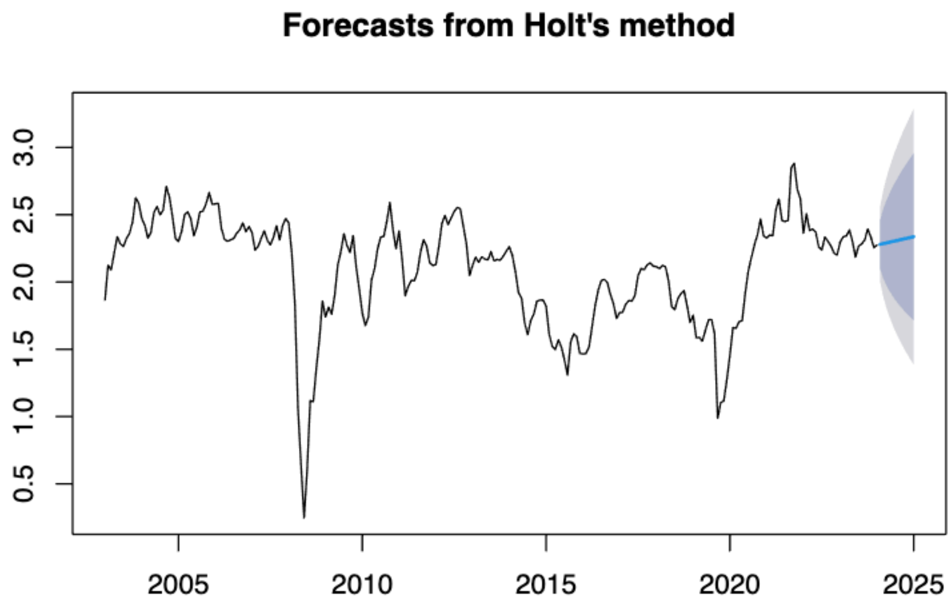
```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Feb 2024       2.274544 2.099701 2.449386 2.007145 2.541942
## Mar 2024       2.274544 2.027291 2.521796 1.896404 2.652683
## Apr 2024       2.274544 1.971728 2.577359 1.811426 2.737661
## May 2024       2.274544 1.924885 2.624202 1.739787 2.809301
## Jun 2024       2.274544 1.883615 2.665472 1.676670 2.872417
## Jul 2024       2.274544 1.846304 2.702783 1.619608 2.929479
## Aug 2024       2.274544 1.811993 2.737094 1.567134 2.981953
## Sep 2024       2.274544 1.780057 2.769030 1.518292 3.030795
## Oct 2024       2.274544 1.750063 2.799024 1.472419 3.076668
## Nov 2024       2.274544 1.721693 2.827394 1.429031 3.120056
## Dec 2024       2.274544 1.694709 2.854378 1.387763 3.161324
## Jan 2025       2.274544 1.668927 2.880160 1.348333 3.200754
```

-**Q1:** The differenced series removes the trend seen in the original series, resulting in more stable fluctuations around a constant mean. This transformation helps eliminate non-stationary behavior caused by long-term trends, making the series appear more stationary.

-**Q2:** The ACF plot of the differenced series shows a rapid decline in autocorrelations, with most lags falling within the confidence bounds after the first few lags. This behavior indicates that the series is now stationary. The results of the Augmented Dickey-Fuller (ADF) test confirm this, with a p-value of 0.01, allowing us to reject the null hypothesis of non-stationarity. This demonstrates that differencing successfully addressed the non-stationarity in the original series.

Holt's Linear Trend Method

```
holt_model <- holt(inflation_time_series, h = 12)
plot(forecast(holt_model))
```




```
forecast(holt_model)
```

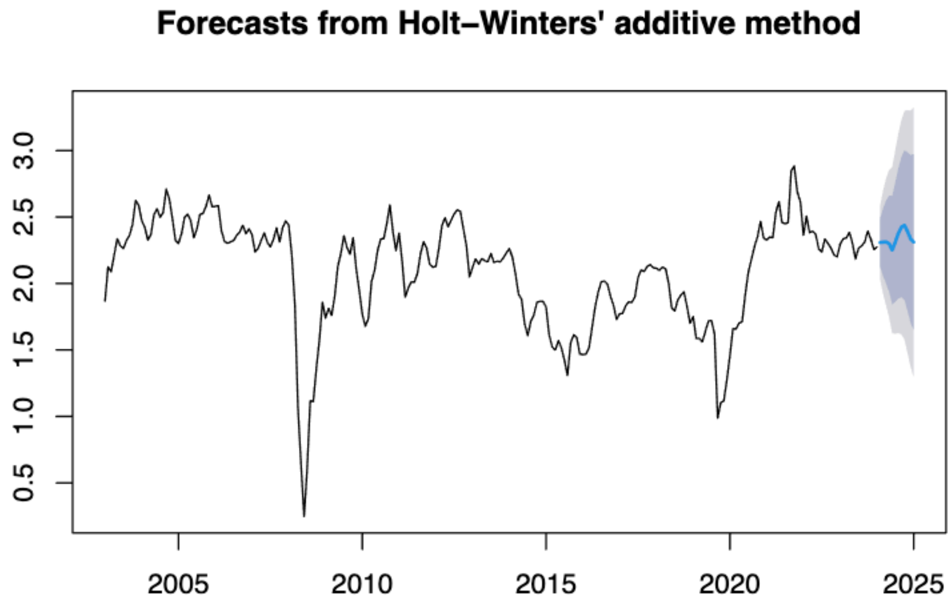
```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Feb 2024       2.279683 2.100590 2.458775 2.005784 2.553581
## Mar 2024       2.284821 2.031546 2.538096 1.897470 2.672172
## Apr 2024       2.289960 1.979752 2.600167 1.815538 2.764381
## May 2024       2.295098 1.936887 2.653310 1.747261 2.842936
## Jun 2024       2.300237 1.899726 2.700748 1.687708 2.912765
## Jul 2024       2.305376 1.866617 2.744134 1.634353 2.976398
## Aug 2024       2.310514 1.836579 2.784450 1.585692 3.035336
## Sep 2024       2.315653 1.808970 2.822335 1.540748 3.090557
## Oct 2024       2.320791 1.783347 2.858236 1.498841 3.142741
## Nov 2024       2.325930 1.759386 2.892474 1.459476 3.192384
## Dec 2024       2.331068 1.736843 2.925294 1.422279 3.239858
## Jan 2025       2.336207 1.715529 2.956886 1.386961 3.285453
```

-**Q1:** The Holt method performs well in capturing the overall trend in the data. It accounts for upward and downward movements over time by modeling both the level and trend components of the series. However, it does not account for seasonality, which may limit its accuracy for data with seasonal patterns.

-**Q2:** Yes, the model's forecast follows the general trend in the data, providing a smooth continuation of the observed trend. While the forecast aligns with the overall direction of the data, it may oversimplify some of the finer fluctuations and seasonal variations due to its lack of a seasonal component.

Holt-Winters Method

```
hw_model <- hw(inflation_time_series, h = 12)
plot(forecast(hw_model))
```




```r
forecast(hw_model)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Feb 2024       2.308466 2.129907 2.487025 2.035383 2.581549
## Mar 2024       2.311241 2.057289 2.565194 1.922854 2.699629
## Apr 2024       2.311590 1.998596 2.624584 1.832907 2.790273
## May 2024       2.298752 1.935001 2.662502 1.742444 2.855060
## Jun 2024       2.250614 1.841285 2.659943 1.624599 2.876629
## Jul 2024       2.312752 1.861437 2.764067 1.622525 3.002978
## Aug 2024       2.377886 1.887244 2.868528 1.627514 3.128258
## Sep 2024       2.424182 1.896266 2.952097 1.616804 3.231559
## Oct 2024       2.438485 1.874931 3.002039 1.576604 3.300366
## Nov 2024       2.389244 1.791386 2.987102 1.474899 3.303588
## Dec 2024       2.335754 1.704698 2.966809 1.370638 3.300869
## Jan 2025       2.311212 1.647891 2.974532 1.296751 3.325672
```

-**Q1:** The Holt-Winters additive method effectively captures both the trend and seasonality in the data by incorporating three smoothing components: level, trend, and seasonal effects. This allows it to account for long-term changes in inflation rates while also modeling consistent seasonal fluctuations observed in the time series.

-**Q2:** Yes, the forecasted values are consistent with the observed patterns in the data. The model follows the general trend of the series and accurately reflects the seasonal variations, providing a realistic projection of future values. The inclusion of confidence intervals further highlights the expected range of variation, aligning with the historical behavior of the series.

Evaluate the Accuracy of the Three Exponential Smoothing Models

```r
accuracy(ses_model)
```

```
##                       ME    RMSE        MAE        MPE   MAPE      MASE
## Training set 0.001615363 0.13589 0.09409124 -0.6964063 5.8492 0.2754746
##                   ACF1
## Training set 0.348462
```

```r
accuracy(holt_model)
```

```
##                        ME      RMSE        MAE        MPE     MAPE      MASE
## Training set -0.001837653 0.1386376 0.09541306 -0.8689543 5.934346 0.2793445
##                   ACF1
## Training set 0.3580422
```

```r
accuracy(hw_model)
```

```
##                        ME      RMSE        MAE       MPE    MAPE      MASE
## Training set 0.0005219474 0.1348529 0.09350993 -0.616952 5.75179 0.2737727
##                   ACF1
## Training set 0.3445558
```

-**Q1:** The Holt-Winters model (hw_model) provides the best forecast accuracy overall. It has the lowest RMSE (0.13485) and MAE (0.09351) compared to the Simple Exponential Smoothing (SES) and Holt models, indicating it performs better in minimizing errors. Additionally, its MAPE (5.75%) is slightly lower than the SES model (5.84%), which also supports its superior accuracy. This is because the Holt-Winters method accounts for both the trend and seasonality in the data, which aligns better with the characteristics of the time series.

-**Q2:** SES Model: Has a slightly higher RMSE (0.13589) and MAE (0.09409) compared to the Holt-Winters model, as it does not account for trends or seasonality. The MAPE (5.84%) is close to Holt-Winters but slightly higher.

Holt Model: Performs similarly to the SES model, with RMSE (0.13864) and MAE (0.09541), but does slightly worse in accounting for trends. Its MAPE (5.93%) is the highest among the three models.

Holt-Winters Model: Has the lowest overall error metrics (RMSE, MAE, and MAPE), indicating it captures both the trend and seasonality in the data better than the other two models.

ARIMA Estimation

```r
arima <- auto.arima(inflation_time_series)
```

-**Q1:** p (autoagressive term): 1 d (differencing order): 1 q (moving average term): 0

This indicates that the model includes one autoregressive term, the series was differenced once to make it stationary, and no moving average terms were included.

ARIMA Results

```r
summary(arima)
```

```
## Series: inflation_time_series
## ARIMA(1,1,0)
##
## Coefficients:
##          ar1
##       0.3521
## s.e.  0.0592
##
## sigma^2 = 0.01632:  log likelihood = 161.41
## AIC=-318.81   AICc=-318.76   BIC=-311.75
##
## Training set error measures:
##                       ME      RMSE        MAE        MPE     MAPE      MASE
```

```
## Training set 0.001015069 0.1272397 0.08994648 -0.2830496 5.379802 0.2633398
##                     ACF1
## Training set 0.009860845
```

-**Q1:**

$$y_t = 0.35y_{t-1} + \varepsilon_t$$

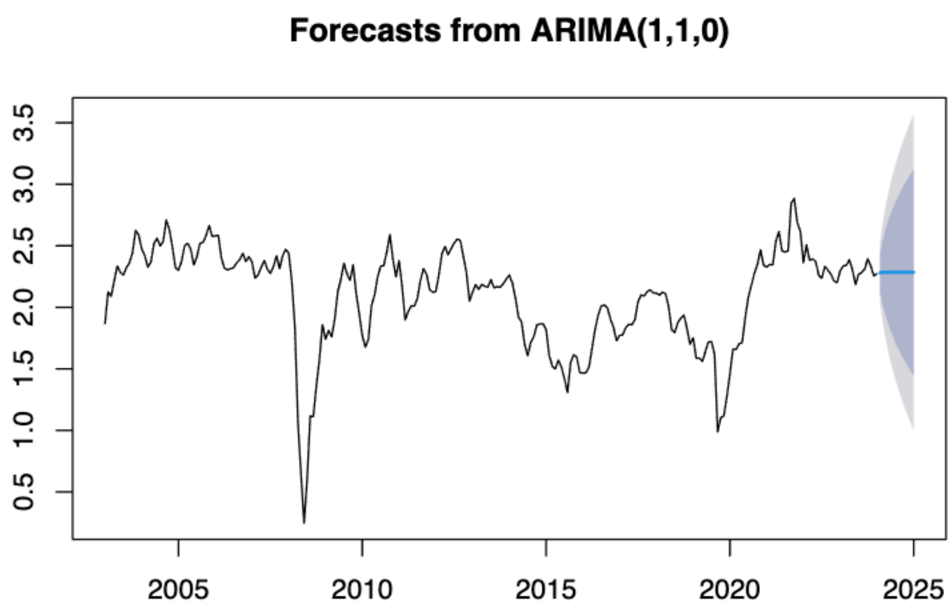
-**Q2:** The ARIMA(1,1,0) model adequately explains the time series data as it captures the key patterns after differencing the series to achieve stationarity. The low error metrics (RMSE = 0.1272, MAE = 0.0899, and MAPE = 5.38%) indicate that the model provides accurate forecasts. Additionally, the low AIC (-318.81) and BIC (-311.75) values demonstrate a good balance between model complexity and fit. The minimal residual autocorrelation (ACF1 = 0.0099) suggests that the model successfully captures the main dynamics of the data. However, it may overlook any potential seasonal effects or more complex relationships, which could require further exploration.

ARIMA Forecast

```r
forecast(arima, h= 12)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Feb 2024       2.281336 2.117623 2.445048 2.030959 2.531712
## Mar 2024       2.283727 2.008403 2.559051 1.862655 2.704799
## Apr 2024       2.284569 1.918229 2.650909 1.724300 2.844837
## May 2024       2.284865 1.842019 2.727712 1.607590 2.962140
## Jun 2024       2.284970 1.775774 2.794165 1.506223 3.063717
## Jul 2024       2.285006 1.716769 2.853243 1.415963 3.154050
## Aug 2024       2.285019 1.663196 2.906843 1.334023 3.236016
## Sep 2024       2.285024 1.613838 2.956210 1.258533 3.311515
## Oct 2024       2.285026 1.567853 3.002198 1.188204 3.381847
## Nov 2024       2.285026 1.524638 3.045414 1.122113 3.447939
## Dec 2024       2.285026 1.483749 3.086304 1.059578 3.510474
## Jan 2025       2.285026 1.444847 3.125206 1.000083 3.569970
```

```r
plot(forecast(arima, h= 12))
```



-**Q1:** The confidence intervals suggest that there is increasing uncertainty in the forecasts as the time horizon extends further into the future. The 80% and 95% confidence intervals widen with each forecasted period, indicating that the model is less certain about the accuracy of its predictions over longer time frames. This is typical in time series forecasting, as more variability and potential for deviation are introduced the further the forecast extends. The intervals provide a range of plausible values, helping to capture the inherent uncertainty in the data while still centering the predictions around the point forecast.

-**Q2:** The forecasted values from the ARIMA(1,1,0) model closely follow the historical trend of the data, maintaining a relatively stable trajectory that aligns with recent patterns. The model captures the general trend without abrupt deviations, indicating it is well-suited for projecting the short-term behavior of the series. However, it lacks the ability to model seasonality or finer variations seen in the historical data.

The forecast appears reliable for short-term predictions, as the ARIMA(1,1,0) model is effective in capturing the primary trend and maintaining stability. However, the widening confidence intervals suggest increasing uncertainty over time, making the forecasts less reliable for longer horizons. Additionally, the lack of seasonal modeling may limit accuracy if the data contains recurring patterns, further reducing reliability for extended predictions. Overall, the model is dependable for short-term trends but requires caution for long-term use.

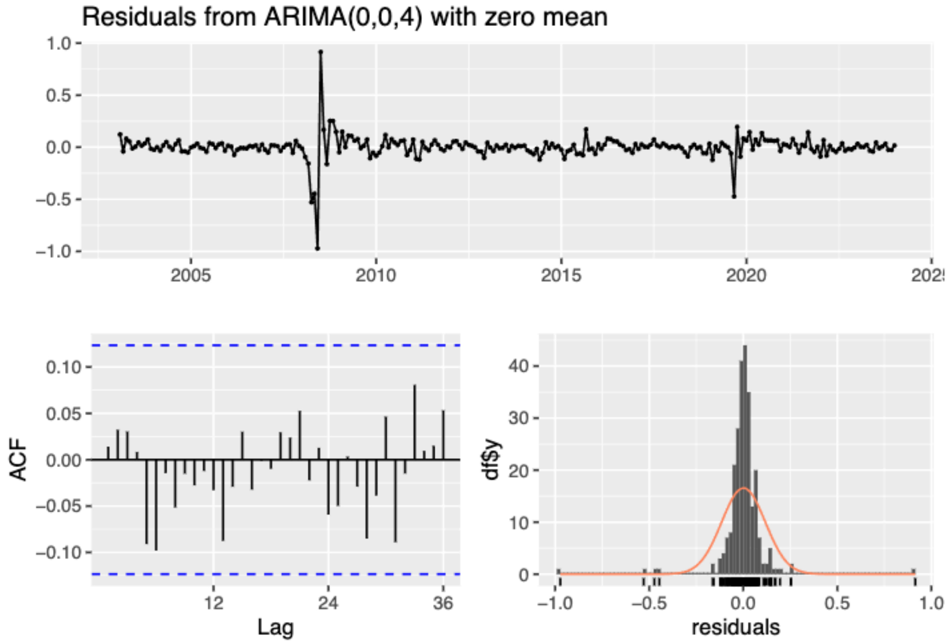
Model Dianogstics by Using Checkresiduals()

```r
log_inflation <- inflation_time_series %>% log() %>% diff()

fit.arima <- auto.arima(log_inflation)
auto.arima(inflation_time_series)
```

```
## Series: inflation_time_series
## ARIMA(1,1,0)
##
## Coefficients:
##          ar1
##       0.3521
## s.e.  0.0592
##
## sigma^2 = 0.01632:  log likelihood = 161.41
## AIC=-318.81   AICc=-318.76   BIC=-311.75
```

```r
checkresiduals(fit.arima)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,4) with zero mean
## Q* = 11.655, df = 20, p-value = 0.9274
##
## Model df: 4.   Total lags used: 24
```

-**Q1:**Since the p-value (0.9274) is much greater than 0.05, we fail to reject the null hypothesis of the Ljung-Box test. This suggests that there is no significant autocorrelation in the residuals at different lags, indicating that the ARIMA(1,1,0) model sufficiently captures the data's structure.

-**Q2:** Since the p-value (0.9274) is much greater than 0.05, we fail to reject the null hypothesis of the Ljung-Box test. This indicates that there is no significant autocorrelation in the residuals. The residuals appear to behave like white noise, suggesting that the ARIMA(1,1,0) model sufficiently captures the data's underlying structure.

-**Q3:** Histogram with Overlayed Curve: The histogram of residuals shows a general bell shape, indicating approximate normality. However, there are slight deviations in the tails, suggesting mild skewness or kurtosis, which deviates from a perfect normal distribution.

Centered Around Zero: The residuals are centered around zero, which is consistent with normality, but there is some evidence of heavier tails.

ACF of Residuals: The autocorrelations in the residuals are small and mostly within the confidence bounds, suggesting that the residuals behave like white noise, a key assumption for normality.