Assignment 2024/25

**Predictive Analytics II: Business Forecasting**

Contents

[1. Introduction 3](#_Toc198302555)

[2. Data exploration 3](#_Toc198302556)

[2.1 Data extracting and preparation 3](#_Toc198302557)

[2.2 Visual inspection and summary statistics 3](#_Toc198302558)

[2.3 Seasonal and trend decomposition 4](#_Toc198302559)

[2.4 Stationarity testing and differencing 5](#_Toc198302560)

[2.5 ACF/PACF analysis 6](#_Toc198302561)

[2.7 Initial model form suggestions 7](#_Toc198302562)

[4. Forecasting model development 7](#_Toc198302563)

[4.1 Train-test split strategy 7](#_Toc198302564)

[4.2 Accuracy metrics applied 8](#_Toc198302565)

[4.3 Benchmarking against naive models 8](#_Toc198302566)

[4.4 SMA models 10](#_Toc198302567)

[4.5 Exponential smoothing models 10](#_Toc198302568)

[4.6 ARIMA models 12](#_Toc198302569)

[4.6.1 Auto ARIMA 12](#_Toc198302570)

[4.6.2 Manual seasonal ARIMA 12](#_Toc198302571)

[4.5.3 Residual diagnostics 13](#_Toc198302572)

[4.7 Time series regression models 13](#_Toc198302573)

[5. Model comparison 14](#_Toc198302574)

[5.1 Comparison across models 14](#_Toc198302575)

[6. Forecast 16](#_Toc198302576)

[7. Discussion and insights 21](#_Toc198302577)

[7.1 Key findings 21](#_Toc198302578)

[7.2 Model performance discussion 21](#_Toc198302579)

[8. Appendix 22](#_Toc198302580)

# 1. Introduction

This report presents a forecasting for two monthly time series datasets, with the objective of identifying the most accurate statistical models for 14-period horizon.

The modelling process includes the following activities:

* Explore and understand the structure of each dataset through statistical and graphical visual analysis
* Apply and compare three major forecasting approaches:
  + Exponential Smoothing (ETS)
  + ARIMA
  + Time Series Regression
* Evaluate model performance using appropriate error metrics and benchmarks
* Select the best-performing model for each series and produce final forecasts

Data exploration includes decomposition and stationarity testing. Each time series is split into training and test datasets for out-of-sample performance evaluation. Both manual and automated variants of ETS and ARIMA models are developed, as well as a manual Time Series regression model. Forecast accuracy is estimated using metrics of RMSE and MAPE, and compared against naive benchmarks.

# 2. Data exploration

## 2.1 Data extracting and preparation

The dataset comprises two monthly time series, each beginning in October 1984. They were extracted from the cleaned Excel file “coursework data (cleaned).xlsx”, which consists of only two time series from the original “coursework data.xlsx” file, with missing values omitted. Both series were transformed into “ts” objects in R with a frequency of 12 to reflect monthly periods.

## 2.2 Visual inspection and summary statistics

Firstly, we built plots of raw series. They revealed potential seasonal fluctuations in both series. The time series are also different in:

ts1:

* There is no clear long-term upward or downward trend
* Visually, ts1 appears to be weakly stationary (constant mean and variance over time), but this would need confirmation with formal tests

ts2:

* Shows a potential downward trend.
* This trend suggests non-stationarity in mean, meaning the average value decreases over time.

|  |  |
| --- | --- |
| P89C1T1#yIS1 | P90C2T1#yIS1 |

Figure 1. Plots of raw data series

Visual inspection was followed by basic descriptive statistics.

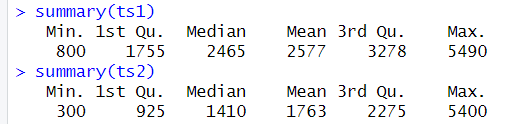


Figure 2. Descriptive statistics

In both ts1and ts2, the mean > median suggesting a right-skewed distribution.

## 2.3 Seasonal and trend decomposition

Both series were decomposed using multiplicative decomposition. This method isolated the trend, seasonal, and irregular components.

The decomposition confirmed clear and regular seasonal patterns, particularly strong in ts1. This helps to build models that incorporate seasonal structure such as Holt-Winters and ARIMA.

|  |  |  |
| --- | --- | --- |
| **Feature** | **ts1** | **ts2** |
| Trend | Nonlinear, rising then falling then rising again | A downward trend from 1985 to around 1992, then slightly flat |
| Seasonality | Regular, strong, 12-month cycle | Regular, strong, 12-month cycle |
| Randomness | Mostly stable, a few spikes | Mostly stable, a few spikes |

|  |  |
| --- | --- |
| P117C1T3#yIS1 | P118C2T3#yIS1 |

Figure 3. Decomposition plots

## 2.4 Stationarity testing and differencing

To assess stationarity, ADF tests were conducted (null hypothesis assumes non-stationarity):

* For ts1, the ADF test returned a p-value < 0.05, allowing us to reject the null hypothesis and implying that the series is stationary and no regular differencing is necessary.
* For ts2, a similar result was derived.

|  |  |
| --- | --- |
| P125C1T4#yIS1 | P126C2T4#yIS1 |

Figure 4. Results of ADF tests

We also run KPSS tests that have opposite null hypothis (assume stationary)

|  |  |
| --- | --- |
| P130C1T5#yIS1 | P131C2T5#yIS1 |

Conclusion: ts2 is not stationary (since the p-value < 0.05, we reject the null hypothesis), ts1 is likely stationary.

We also differenced datasets to explore further stability. Ts2 seems to be have weak stationarity.

|  |  |
| --- | --- |
| P136C1T6#yIS1 | P137C2T6#yIS1 |

Figure 5. Differencing plots

This indicates that ts1 is stationary and for ARIMA modelling **d = 0** is appropriate and no differencing needed.

|  |  |
| --- | --- |
| P141C1T7#yIS1 | P142C2T7#yIS1 |

Figure 6. Differencing ADF tests

We also tested for seasonal stationary

|  |  |
| --- | --- |
| P146C1T8#yIS1 | P147C2T8#yIS1 |
| P149C3T8#yIS1 | P150C4T8#yIS1 |

Figure 7. Seasonality tests

Ts1 seasonal passes both ADF and KPSS and clearly stationary

Ts2 seasonal has conflicting results:

* ADF implies stationary (p = 0.01418)
* KPSS implies not stationary (p = 0.02407)

## 2.5 ACF/PACF analysis

Visual inspection of the autocorrelation function (ACF)

|  |  |
| --- | --- |
| Ts1 | Ts2 |
| P163C3T9#yIS1 | P164C4T9#yIS1 |
| P166C5T9#yIS1 | P167C6T9#yIS1 |

Figure 8. ACF/PACF plots

|  |  |  |
| --- | --- | --- |
|  | Ts1 | Ts2 |
| ACF Plot | Significant spikes at multiple lags (lag 1, 2, 12). ACF tails off gradually and non-seasonal MA(1) component possible. | Decay across many lags (1,2,3,4) as a sign of non-stationarity, non-seasonal MA(0) is possible |
| ACF Plot differencing | Seasonal spikes are now gone | Most spikes are gone |
| PACF Plot | Significant at lag 1, and then declines quickly. PACF cuts off after lag 1 and non-seasonal AR(1) component possible. | Similar to ts1. AR(2) may be acceptable (marginal support), but not strongly confirmed visually |
| ACF Plot differencing | Could justify AR(1) (the spike at lag 1 is small and within the bands) | Similar to ts1. |

## 2.7 Initial model form suggestions

Based on our diagnostics, the following preliminary modelling directions for manual ARIMA might be suggested:

* **ts1:** Likely candidates include a seasonal ARIMA (as (1,0,1)(1,1,1) and multiplicative Holt-Winters (ETS).
* **ts2:** A seasonal ARIMA with stronger short-term dynamics (as (1,0,0)(0,1,1) or (2,0,0)(0,1,1) and potentially seasonal regression with lag variables.

These hypotheses were tested in the model-building phase, with iterative refinement based on accuracy and residual diagnostics.

# 4. Forecasting model development

## 4.1 Train-test split strategy

Each time series was split into a training and testing set to evaluate forecast performance. The final 12 observations were held out for testing (h=12), with the remaining data used for training. This split enabled an assessment of out-of-sample accuracy.

|  |  |
| --- | --- |
| P201C1T11#yIS1 | P202C2T11#yIS1 |

|  |  |
| --- | --- |
| P205C1T12#yIS1 | P206C2T12#yIS1 |

Figure 7. Train/test split

## 4.2 Accuracy metrics applied

To evaluate forecast performance, the following error metrics were used:

* Root Mean Squared Error (RMSE): Sensitive to large errors and penalizes them heavily.
* Mean Absolute Error (MAE): Measures average magnitude of forecast errors, providing a more interpretable measure.
* Mean Absolute Percentage Error (MAPE): Offers a relative error scale, useful when comparing across different series.

These metrics were calculated for both in-sample and out-of-sample forecasts, with task of final model selection.

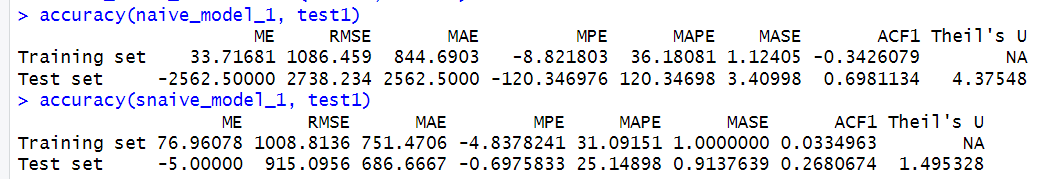
## 4.3 Benchmarking against naive models

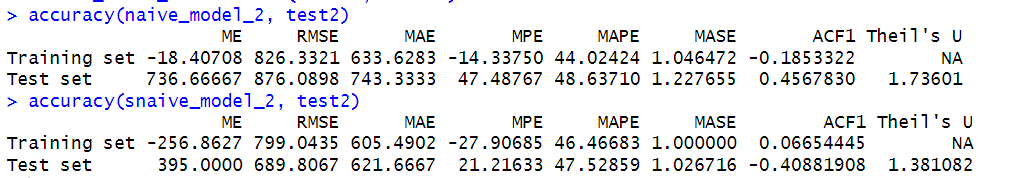
To evaluate model improvement, all forecasts were benchmarked against two models:

* Naive Model: Forecasts assume no change from the last observed value.
* Seasonal Naive Model: Forecasts replicate the value from the same month in the previous year.

These models provided baseline levels of performance. Our candidate models with seasonal components might significantly outperformed both benchmarks in terms of RMSE and MAPE.

Accuracy





The best RMSE was 915 on test dataset for simple naïve for ts1 and 690 for ts2

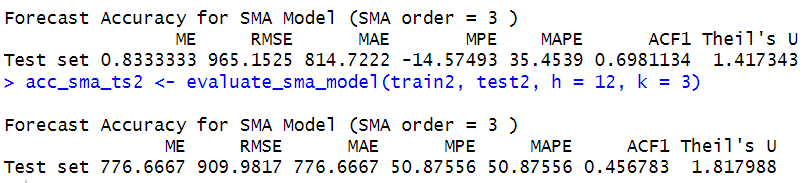
Below we produced 14-period forecasts

|  |  |
| --- | --- |
| P226C1T13#yIS1 | P227C2T13#yIS1 |
| P229C3T13#yIS1 | P230C4T13#yIS1 |

Figure 8. Plots of 14-period forecasts

## 4.4 SMA models

We applied a Simple Moving Average (SMA) with to the training data, generated a flat forecast, and compared it against the actual test values.



## 4.5 Exponential smoothing models

Multiple exponential smoothing models were fitted and compared:

* ETS("MMM"): multiplicative error, trend, and seasonality
* ETS("AAdN"): damped additive trend with no seasonality
* ETS("ZZZ"): fully automatic
* ETS("ZZN"): automatic non-seasonal
* Manual Holt-Winters: multiplicative seasonality
* Auto ETS: from ets() with automatic optimization

Each model was fit on the training data and used to generate 12-period forecasts to compare with the actual test values.

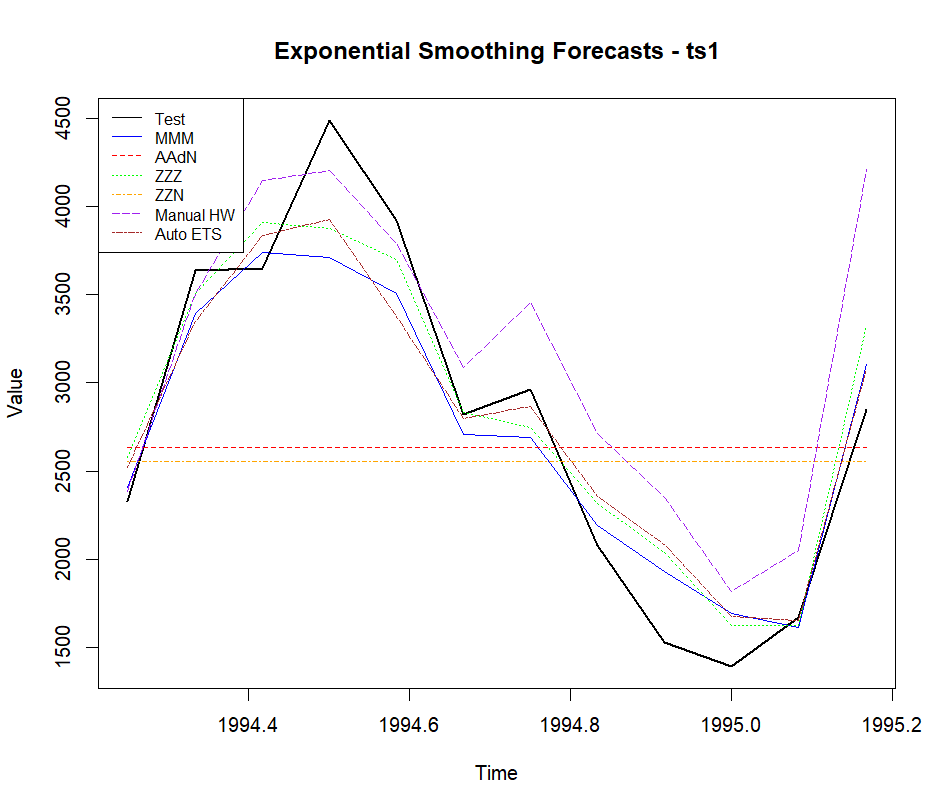


Figure 9. Actual test values vs ETS forecasts, ts1

For ts1, models with multiplicative seasonality such as ETS(MMM) and Holt-Winters performed notably better than non-seasonal or additive models. The automatic ETS selection also produced good results but was slightly less accurate than the manually specified seasonal alternatives. The regression and SMA models performed worse in comparison, highlighting the importance of seasonal structure in this series.

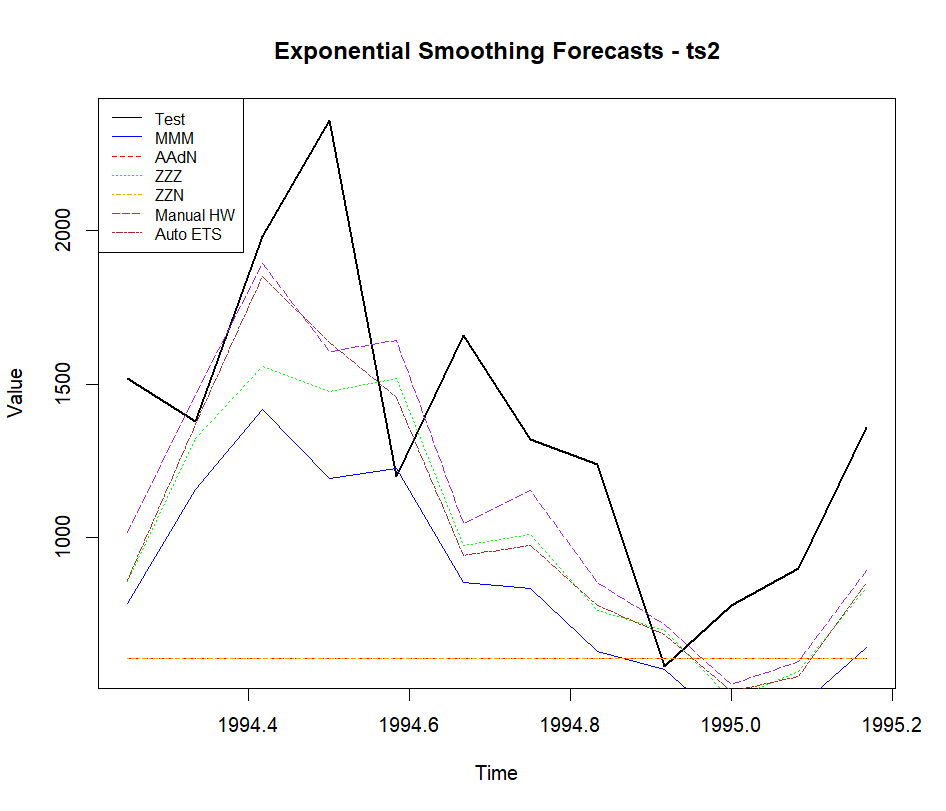
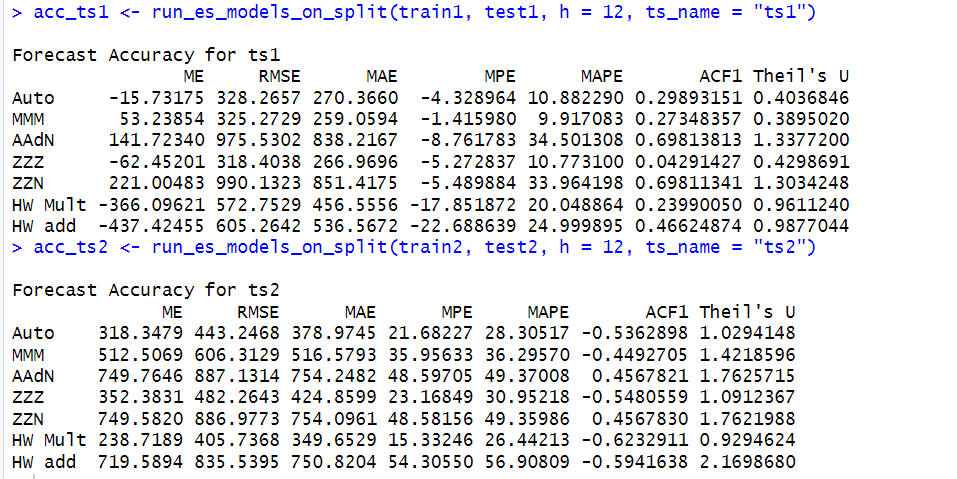


Figure 10. Actual test values vs ETS forecasts, ts2

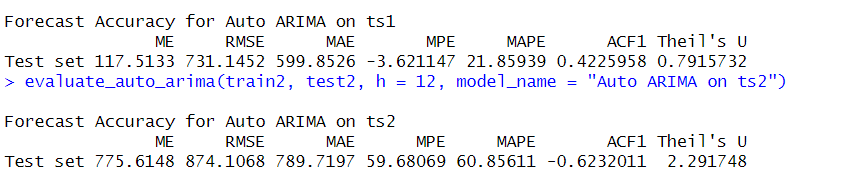


For ts1, the best accuracy was reached with ZZZ model (RMSE of 318) and for ts2 with HW multiplicative (RMSE of 406)

## 4.6 ARIMA models

### 4.6.1 Auto ARIMA

Auto ARIMA models were built using the auto.arima() function from the forecast package, which automatically selected the best-fitting model based on AICc. For both ts1 and ts2, seasonal ARIMA models were allowed by setting seasonal = TRUE.



For ts1, the auto-selected model achieved a fit with a test RMSE of 731 and a MAPE of 21.86%.

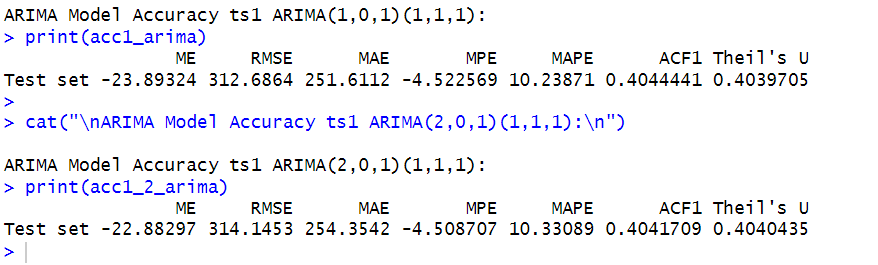
The auto-selected model for ts2 performed less effectively with RMSE of 874 and a MAPE over 60%.

### 4.6.2 Manual seasonal ARIMA

Manual ARIMA models were developed by following the Box-Jenkins methodology. This involved inspecting ACF and PACF plots, performing stationarity checks using the Augmented Dickey-Fuller test, and examining residuals at each step.

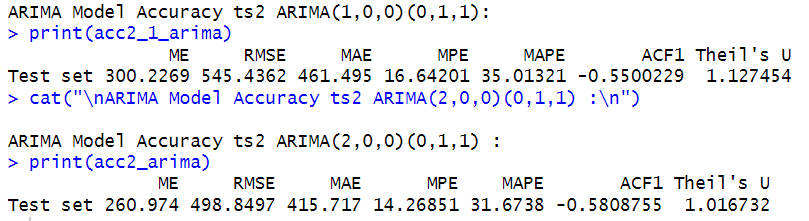
**ts1:**

After iterative refinement, ARIMA(1,0,1)(1,1,1)[12] was chosen as the final model. It provided significantly improved accuracy, with a test RMSE of 312 and MAPE of just 10.24%.



**ts2:**

ARIMA(2,0,0)(0,1,1)[12] was selected as the best model with RMSE of 498.



### 4.5.3 Residual diagnostics

Residual plots showed white noise behaviour with no significant autocorrelations, supporting the validity of the chosen ARIMA specifications.

|  |  |
| --- | --- |
| Ts1  P272C1T14#yIS1 | Ts2  P274C2T14#yIS1 |

Figure 11. Residual diagnostics

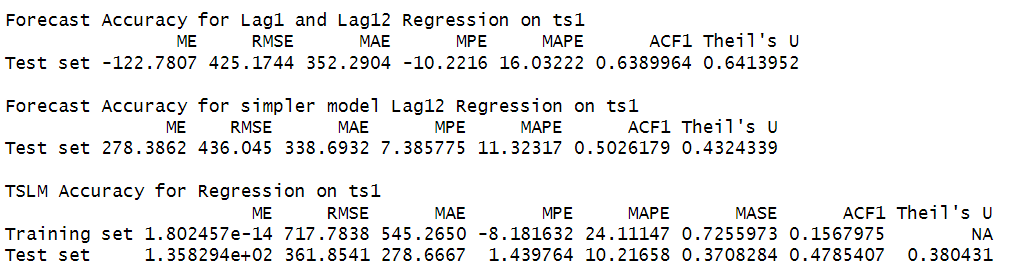
## 4.7 Time series regression models

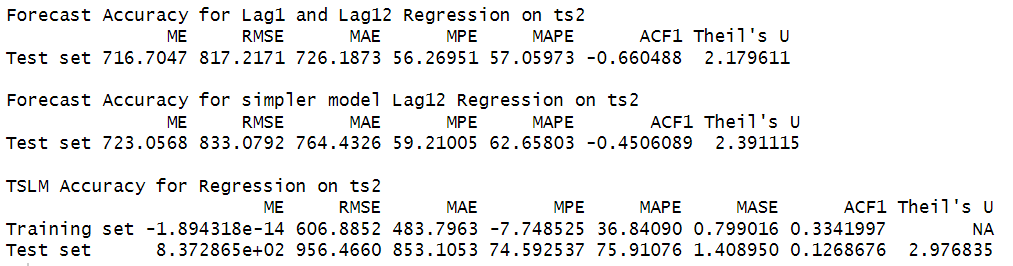
Each series was modelled using:

* Lag 1 (L1)
* Lag 12 (L12)
* Linear time trend
* Monthly seasonal dummies (by season )

Regression was fit using lm() and train datasets while forecasts were generated for 12 periods. This approach aimed to model autocorrelation and seasonality linearly.

We also modelled by using tslm and simpler model with only Lag 12





The best model for ts1 was based on tslm (RMSE of 361) and for ts2 based on Lag1 and Lag12 (RMSE of 817). Both models were below results of ARIMA models

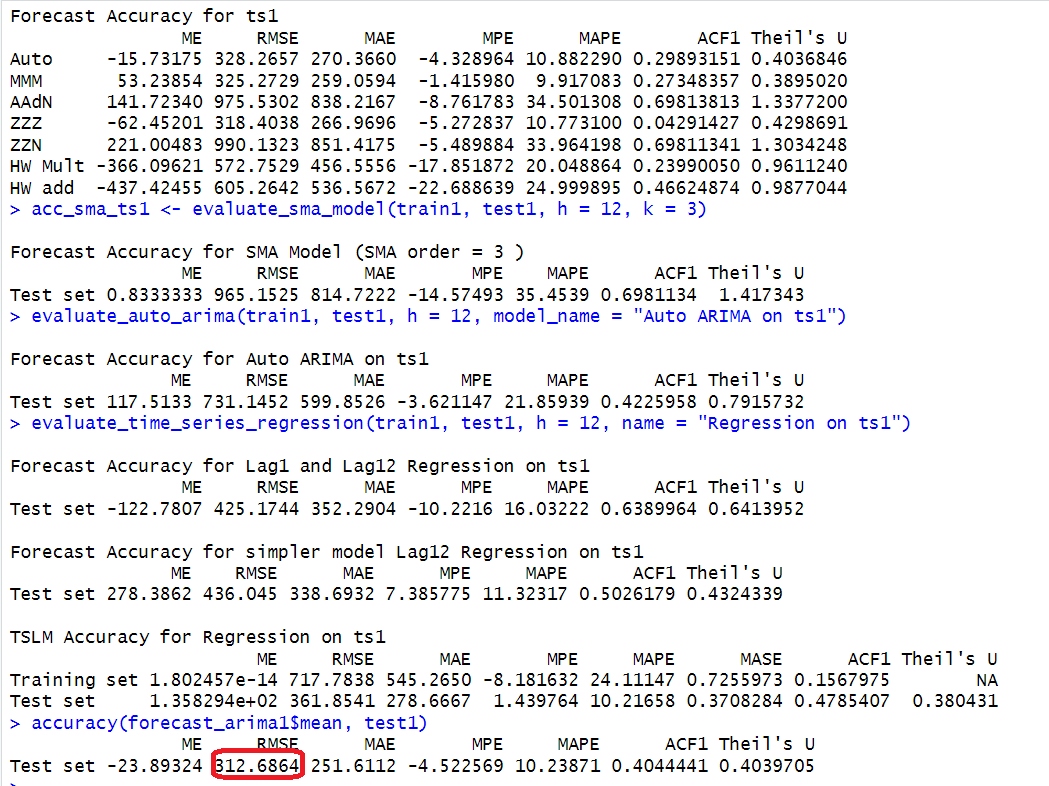
# 5. Model comparison

## 5.1 Comparison across models

Multiple forecasting models were developed for each time series, including exponential smoothing methods, simple moving averages, Auto ARIMA, manually specified ARIMA models, and time series regression. Model performance was assessed using accuracy metrics such as RMSE and MAPE on a test set (12 months).

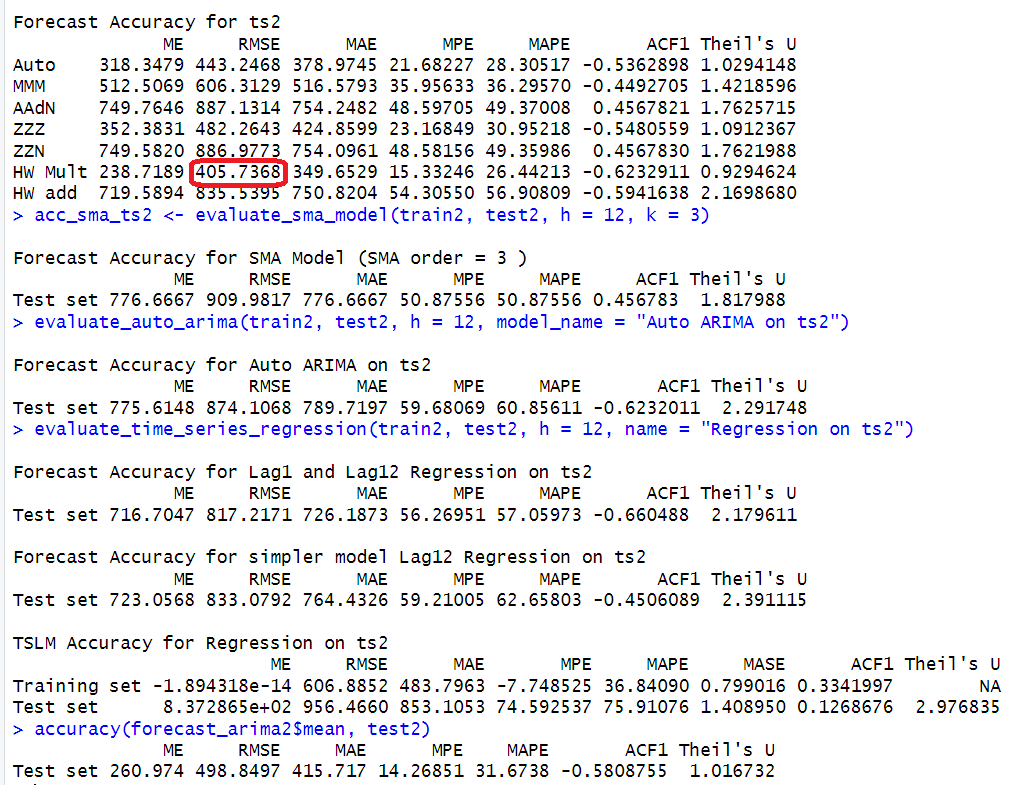
**Best Model for Time Series 1: ARIMA(1,0,1)(1,1,1)[12]**

This model provided the best overall forecast accuracy, achieving the lowest RMSE (312.7) and MAPE (10.2%).



**Best Model for Time Series 2: HW Multiplicative**

It achieved an RMSE of 405 and a MAPE of 26%, substantially better than other models. The second top performing was manual ARIMA

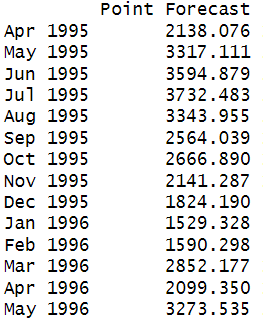


Overall, models that incorporated seasonality and appropriate differencing (particularly seasonal ARIMA) consistently produced good forecasts across both series. The evaluation results confirm the importance of iterative model specification, residual diagnostics, and formal accuracy comparisons, all of which helped to identify the best-performing models.

# 6. Forecast

This analysis supports the use of data-driven time series modelling for enhancing forecast reliability and operational decision-making, especially in environments where seasonality and autocorrelation are prominent features of the underlying data.

**Ts1.** We produced 14 period forecast below



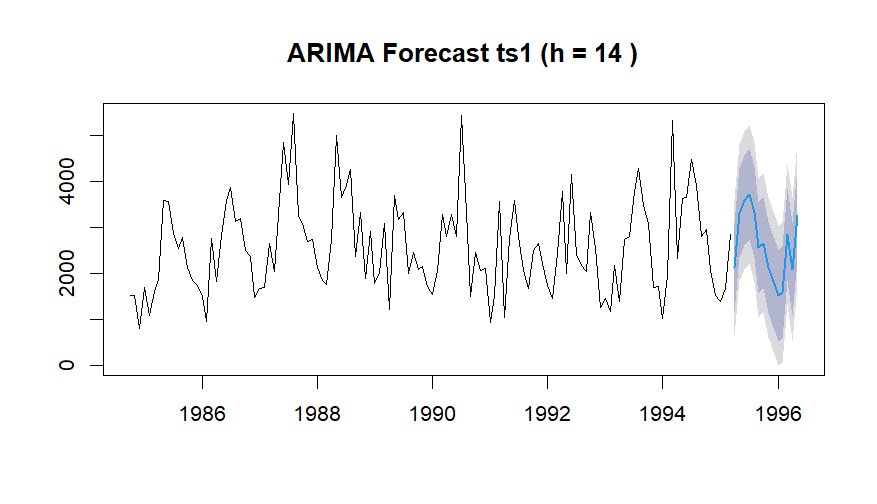
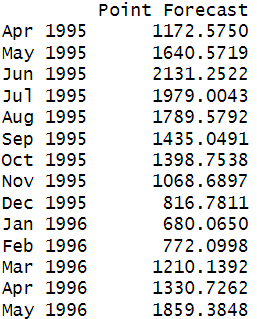
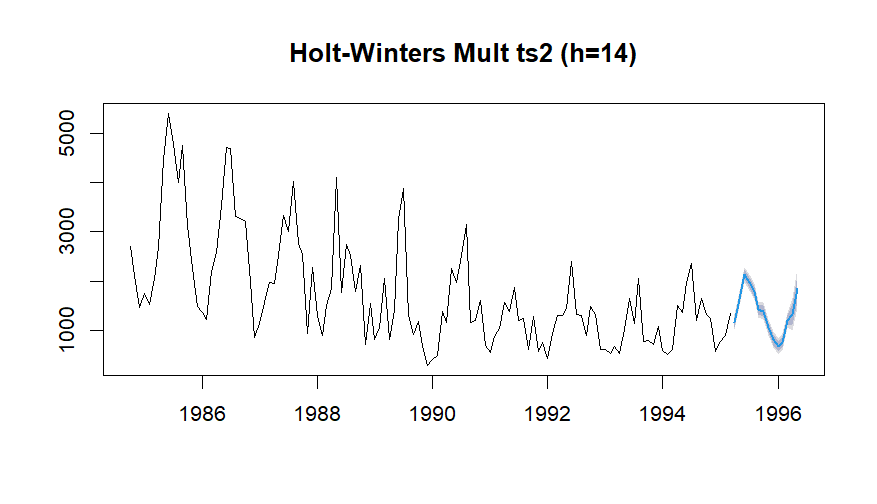


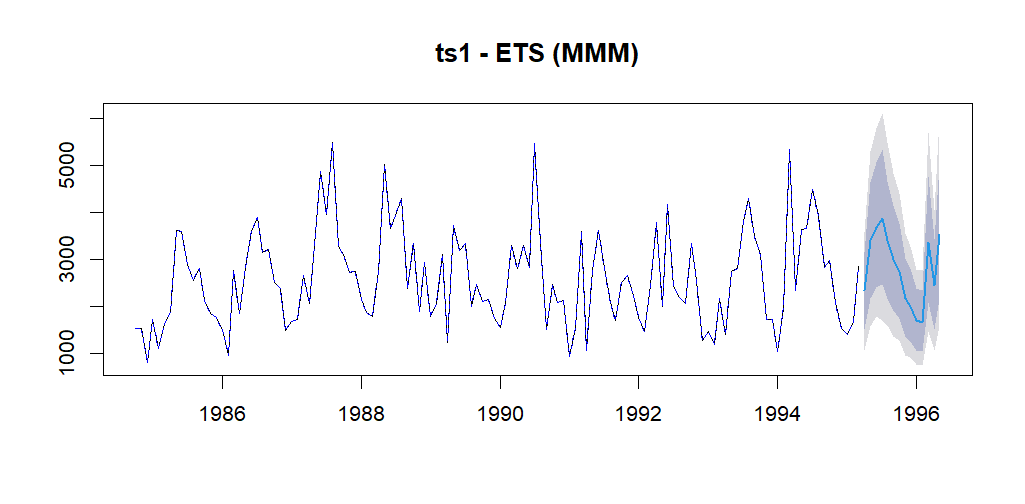
Figure 12. Forecast for ts1

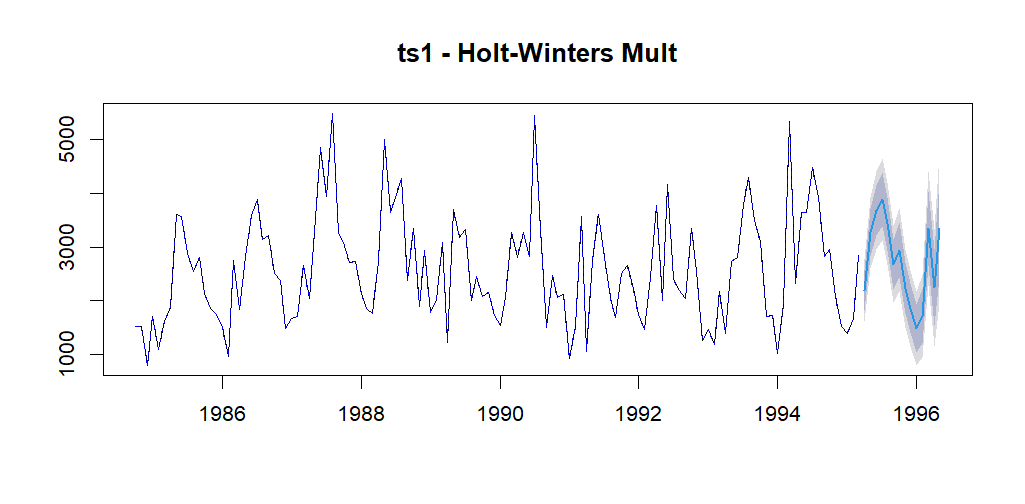
**Ts2.** We produced 14 period forecast below

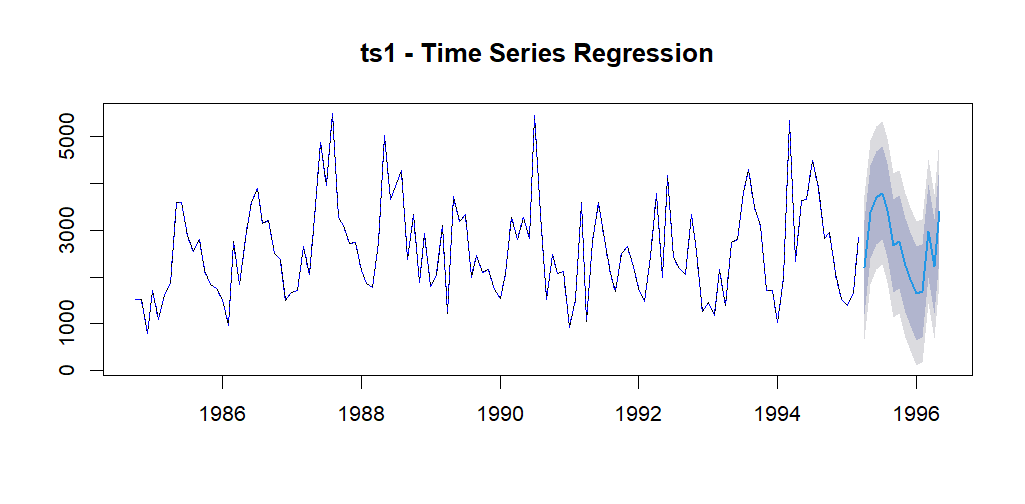


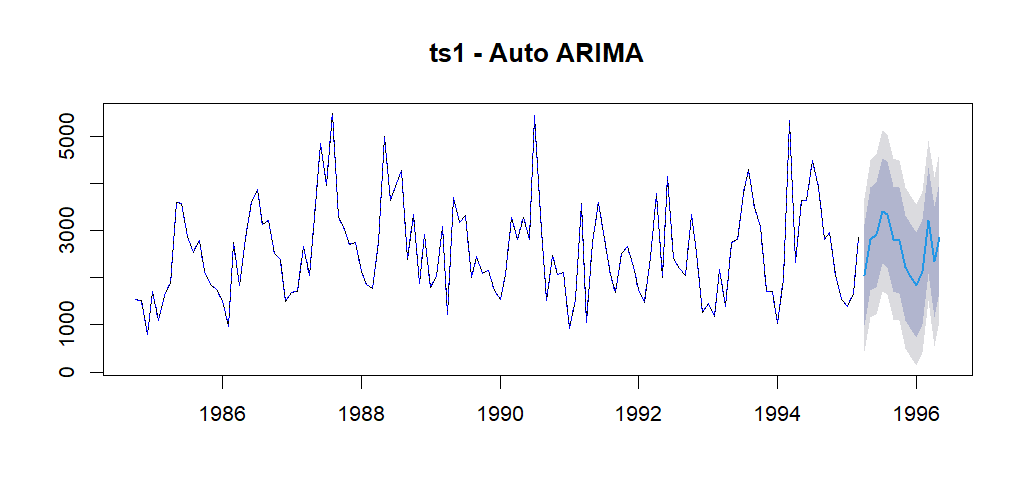
Figure 12. Forecast for ts2

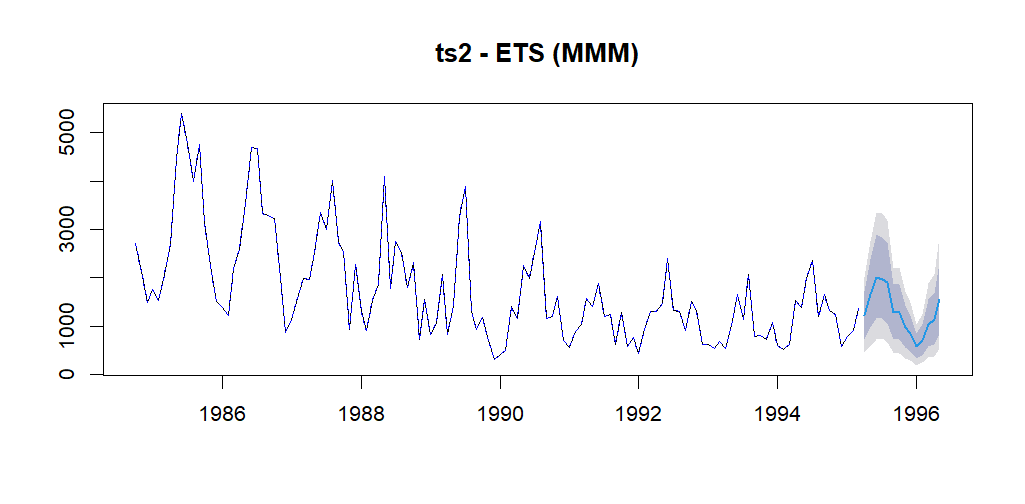
We also provided several forecasts for other models

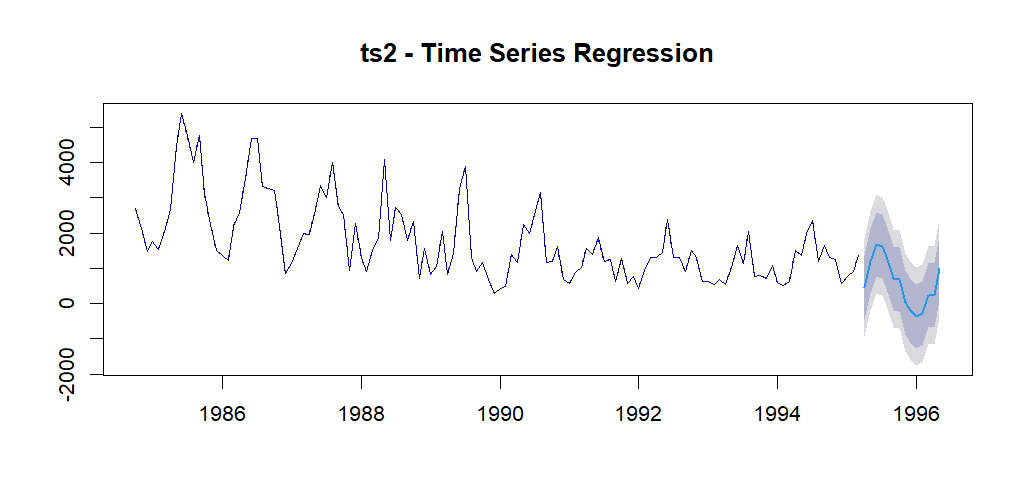


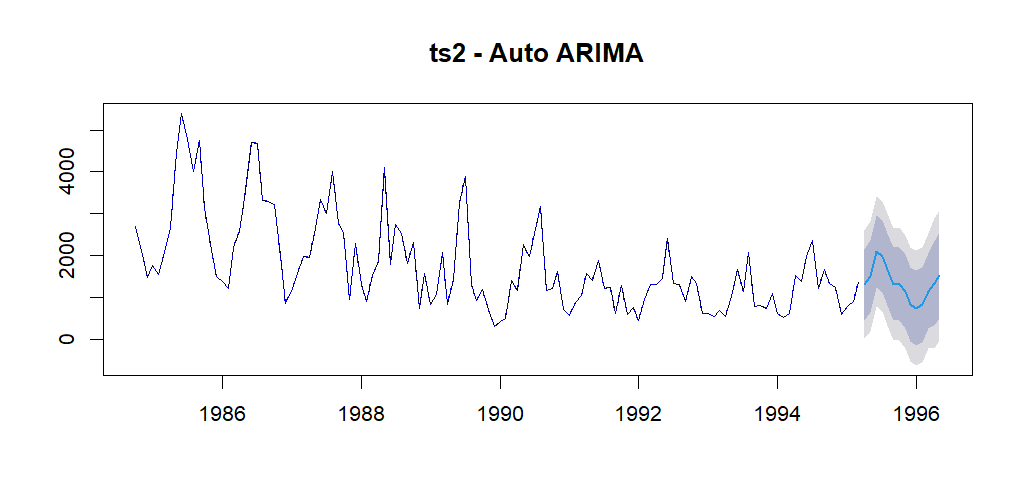












# 7. Discussion and insights

## 7.1 Key findings

ts1 exhibited strong, regular seasonality and short-term autocorrelation. After testing multiple methods, ARIMA(1,0,1)(1,1,1)[12] emerged as the most effective model with the lowest RMSE and MAPE, supported by white-noise residuals and passing the Ljung-Box test.

ts2 showed a declining trend with weaker but consistent seasonality. The ARIMA(2,0,0)(0,1,1)[12] model showed good top performance close to the best HW multiplicative model

Simple moving average and auto ARIMA models produced higher errors and less reliable diagnostics.

## 7.2 Model performance discussion

| **Model** | **ts1** | **ts2** |
| --- | --- | --- |
| Manual ARIMA | Best overall; strong seasonal structure captured | Good fit. Modelled declining level and seasonal MA |
| ETS ZZZ | Competitive but slightly underperformed | Adequate trend fit but weaker on ts2's irregularity |
| ETS HW multiplicative | Likely distorts seasonal effects | HW reacts quickly to recent patterns, thanks to exponential smoothing. Was better than ARIMA due to reliance on fixed lag structure (AR and MA terms) and slower adaptation to recent regime shifts |
| Regression | Worked on ts1 but missed nonlinear seasonal effects | High error; did not reflect long-term trend decay |
| Auto ARIMA | Quick and automated baseline | Overfitted seasonal noise; lacked stable parameters |
| SMA | Useful as a benchmark. Too simplistic for both series | Same as ts1 |

Manual SARIMA models offered the best balance of accuracy, interpretability, and residual behaviour.

7.3 Limitations and alternative considerations

The short forecast horizon (12–14 periods) may not reflect long-term dynamics or structural changes.

No external regressors were included. Introducing additional variables (e.g., economic indicators, marketing events) could improve model.

# 8. Appendix

**R script:**

library(readxl)

library(forecast)

library(tseries)

library(smooth)

library(tsutils)

data <- read\_excel(

"coursework data (cleaned).xlsx",

sheet = "Monthly Data"

)

ts1 <- ts(na.omit(as.numeric(data[1, 5:ncol(data)])), start = c(1984,10), frequency = 12)

ts2 <- ts(na.omit(as.numeric(data[2, 5:ncol(data)])), start = c(1984,10), frequency = 12)

#Data Exploration

plot(ts1, main = "First Time Series ts1")

print(ts1)

plot(ts2, main = "Second Time Series ts2")

print(ts2)

summary(ts1)

summary(ts2)

#Plot the Decomposition ts1

decomposition1 <- decompose(ts1, type = "multiplicative")

plot(decomposition1)

title(main = "ts1") # Adds title to current plot

str(decomposition1)

decomposition1$irregular

#Extract the Trend

decomposition1$trend

#Extract the Seasonality

decomposition1$season

#Plot the Decomposition ts2

decomposition2 <- decompose(ts2, type = "multiplicative")

plot(decomposition2)

title(main = "ts2")

str(decomposition2)

decomposition2$irregular

#Extract the Trend

decomposition2$trend

#Extract the Seasonality

decomposition2$season

print(adf.test(ts1, alternative = "stationary"))

print(adf.test(ts2, alternative = "stationary"))

kpss.test(ts1)

kpss.test(ts2)

#Differencing

diff\_data1 <- diff(ts1)

diff\_data2 <- diff(ts2)

plot(diff\_data1)

plot(diff\_data2)

adf.test(diff(ts1))

adf.test(diff(ts2))

# Seasonal stationary

seasonal\_diff1 <- diff(ts1, lag = 12) # Lag 12

seasonal\_diff2 <- diff(ts2, lag = 12) # Lag 12

print(adf.test(seasonal\_diff1, alternative = "stationary"))

print(adf.test(seasonal\_diff2, alternative = "stationary"))

print(kpss.test(seasonal\_diff1))

print(kpss.test(seasonal\_diff2))

#ACF and PACF analysis

tsdisplay(ts1, main = "ACF and PACF of Original Series (ts1)")

tsdisplay(ts2, main = "ACF and PACF of Original Series (ts2)")

#Seasonal Patterns

seasplot(ts1)

title(main = "ts1", col.main = "blue", font.main = 3, cex.main = 0.8)

seasplot(ts2)

title(main = "ts2", col.main = "blue", font.main = 3, cex.main = 0.8)

#Train/test split

h <- 12

split\_ts\_monthly <- function(ts\_data, h) {

n <- length(ts\_data)

train\_end <- time(ts\_data)[n - h]

test\_start <- time(ts\_data)[n - h + 1]

train <- window(ts\_data, end = train\_end)

test <- window(ts\_data, start = test\_start)

return(list(train = train, test = test))

}

split\_data1 <- split\_ts\_monthly(ts1, h)

train1 <- split\_data1$train

test1 <- split\_data1$test

plot(train1, main = "Train 1")

plot(test1, main = "Test 2")

split\_data2 <- split\_ts\_monthly(ts2, h)

train2 <- split\_data2$train

test2 <- split\_data2$test

plot(train2, main = "Train 2")

plot(test2, main = "Test 2")

#Naive and seasonal naive

naive\_model\_1 <- naive(train1, h = 12)

snaive\_model\_1 <- snaive(train1, h = 12)

accuracy(naive\_model\_1, test1)

accuracy(snaive\_model\_1, test1)

naive\_model\_2 <- naive(train2, h = 12)

snaive\_model\_2 <- snaive(train2, h = 12)

accuracy(naive\_model\_2, test2)

accuracy(snaive\_model\_2, test2)

#Exponential smoothing models

run\_es\_models\_on\_split <- function(train, test, h = 12, ts\_name = "Series") {

acc\_list <- list()

#Automatic ETS

es\_auto <- ets(train)

es\_auto\_forecast <- forecast(es\_auto, h = h)

acc\_list[[1]] <- accuracy(es\_auto\_forecast$mean, test)

names(acc\_list)[1] <- "Auto"

#ETS MMM

es\_MMM <- ets(train, model = "MMM")

es\_MMM\_forecast <- forecast(es\_MMM, h = h)

acc\_list[[2]] <- accuracy(es\_MMM\_forecast$mean, test)

names(acc\_list)[length(acc\_list)] <- "MMM"

#Damped Holt AAdN

es\_AAdN <- es(train, model = "AAdN", h = h)

es\_AAdN\_forecast <- forecast(es\_AAdN, h = h)

acc\_list[[3]] <- accuracy(es\_AAdN\_forecast$mean, test)

names(acc\_list)[length(acc\_list)] <- "AAdN"

#Optimized ETS ZZZ

es\_ZZZ <- es(train, model = "ZZZ", h = h)

es\_ZZZ\_forecast <- forecast(es\_ZZZ, h = h)

acc\_list[[4]] <- accuracy(es\_ZZZ\_forecast$mean, test)

names(acc\_list)[length(acc\_list)] <- "ZZZ"

#Non-seasonal ETS ZZN

es\_ZZN <- es(train, model = "ZZN", h = h, silent = TRUE)

es\_ZZN\_forecast <- forecast(es\_ZZN, h = h)

acc\_list[[5]] <- accuracy(es\_ZZN\_forecast$mean, test)

names(acc\_list)[length(acc\_list)] <- "ZZN"

# Manual Holt-Winters

es\_manual <- HoltWinters(train, seasonal = "multiplicative")

es\_manual\_forecast <- forecast(es\_manual, h = h)

acc\_list[[6]] <- accuracy(es\_manual\_forecast$mean, test)

names(acc\_list)[length(acc\_list)] <- "HW Mult"

#Model 6: Manual Holt-Winters

es\_manual <- HoltWinters(train, seasonal = "additive")

es\_manual\_forecast <- forecast(es\_manual, h = h)

acc\_list[[7]] <- accuracy(es\_manual\_forecast$mean, test)

names(acc\_list)[length(acc\_list)] <- "HW add"

#Accuracy

accuracy\_table <- do.call(rbind, acc\_list)

rownames(accuracy\_table) <- names(acc\_list)

cat("\nForecast Accuracy for", ts\_name, "\n")

print(accuracy\_table)

return(accuracy\_table)

}

acc\_ts1 <- run\_es\_models\_on\_split(train1, test1, h = 12, ts\_name = "ts1")

acc\_ts2 <- run\_es\_models\_on\_split(train2, test2, h = 12, ts\_name = "ts2")

#FORECAST EXPONENTIAL SMOOTHING MODELS

#TS1

es\_MMM <- ets(train1, model = "MMM")

es\_AAdN <- es(train1, model = "AAdN", h = 12)

es\_ZZZ <- es(train1, model = "ZZZ", h = 12)

es\_ZZN <- es(train1, model = "ZZN", h = 12)

es\_manual <- HoltWinters(train1, seasonal = "multiplicative")

es\_auto <- ets(train1)

es\_MMM\_forecast <- forecast(es\_MMM, h = 12)

es\_AAdN\_forecast <- forecast(es\_AAdN, h = 12)

es\_ZZZ\_forecast <- forecast(es\_ZZZ, h = 12)

es\_ZZN\_forecast <- forecast(es\_ZZN, h = 12)

es\_manual\_forecast <- forecast(es\_manual, h = 12)

es\_auto\_forecast <- forecast(es\_auto, h = 12)

plot(test1, type = "l", col = "black", lwd = 2,

main = "Exponential Smoothing Forecasts - ts1",

ylab = "Value", xlab = "Time")

lines(es\_MMM\_forecast$mean, col = "blue", lty = 1)

lines(es\_AAdN\_forecast$mean, col = "red", lty = 2)

lines(es\_ZZZ\_forecast$mean, col = "green", lty = 3)

lines(es\_ZZN\_forecast$mean, col = "orange", lty = 4)

lines(es\_manual\_forecast$mean, col = "purple", lty = 5)

lines(es\_auto\_forecast$mean, col = "brown", lty = 6)

legend("topleft",

legend = c("Test", "MMM", "AAdN", "ZZZ", "ZZN", "Manual HW", "Auto ETS"),

col = c("black", "blue", "red", "green", "orange", "purple", "brown"),

lty = c(1,1,2,3,4,5,6), cex = 0.8)

#TS2

es\_MMM\_2 <- ets(train2, model = "MMM")

es\_AAdN\_2 <- es(train2, model = "AAdN", h = 12)

es\_ZZZ\_2 <- es(train2, model = "ZZZ", h = 12)

es\_ZZN\_2 <- es(train2, model = "ZZN", h = 12)

es\_manual\_2 <- HoltWinters(train2, seasonal = "multiplicative")

es\_auto\_2 <- ets(train2)

es\_MMM\_forecast\_2 <- forecast(es\_MMM\_2, h = 12)

es\_AAdN\_forecast\_2 <- forecast(es\_AAdN\_2, h = 12)

es\_ZZZ\_forecast\_2 <- forecast(es\_ZZZ\_2, h = 12)

es\_ZZN\_forecast\_2 <- forecast(es\_ZZN\_2, h = 12)

es\_manual\_forecast\_2 <- forecast(es\_manual\_2, h = 12)

es\_auto\_forecast\_2 <- forecast(es\_auto\_2, h = 12)

plot(test2, type = "l", col = "black", lwd = 2,

main = "Exponential Smoothing Forecasts - ts2",

ylab = "Value", xlab = "Time")

lines(es\_MMM\_forecast\_2$mean, col = "blue", lty = 1)

lines(es\_AAdN\_forecast\_2$mean, col = "red", lty = 2)

lines(es\_ZZZ\_forecast\_2$mean, col = "green", lty = 3)

lines(es\_ZZN\_forecast\_2$mean, col = "orange", lty = 4)

lines(es\_manual\_forecast\_2$mean, col = "purple", lty = 5)

lines(es\_auto\_forecast\_2$mean, col = "brown", lty = 6)

legend("topleft",

legend = c("Test", "MMM", "AAdN", "ZZZ", "ZZN", "Manual HW", "Auto ETS"),

col = c("black", "blue", "red", "green", "orange", "purple", "brown"),

lty = c(1,1,2,3,4,5,6), cex = 0.8)

evaluate\_sma\_model <- function(train, test, h = 12, k = 3, model\_name = "SMA Model") {

library(forecast)

sma\_series <- ma(train, order = k, centre = FALSE)

sma\_no\_na <- sma\_series[!is.na(sma\_series)]

last\_sma <- tail(sma\_no\_na, 1)

sma\_forecast <- ts(

rep(last\_sma, h),

start = start(test),

frequency = frequency(test)

)

cat("\nForecast Accuracy for", model\_name, "(SMA order =", k, ")\n")

print(accuracy(sma\_forecast, test))

}

acc\_sma\_ts1 <- evaluate\_sma\_model(train1, test1, h = 12, k = 3)

acc\_sma\_ts2 <- evaluate\_sma\_model(train2, test2, h = 12, k = 3)

#ARIMA AUTO

evaluate\_auto\_arima <- function(train, test, h = 12, model\_name = "Auto ARIMA") {

library(forecast)

# Fit auto ARIMA model

model <- auto.arima(train, seasonal = TRUE)

# Forecast h steps ahead

forecast\_result <- forecast(model, h = h)

# Print accuracy

cat("\nForecast Accuracy for", model\_name, "\n")

print(accuracy(forecast\_result$mean, test))

}

evaluate\_auto\_arima(train1, test1, h = 12, model\_name = "Auto ARIMA on ts1")

evaluate\_auto\_arima(train2, test2, h = 12, model\_name = "Auto ARIMA on ts2")

#Manual Seasonal ARIMA

#ts1

adf.test(ts1)

tsdisplay(ts1, main = "Raw Series ACF/PACF")

tsdisplay(diff(ts1, lag = 12), main = "After Seasonal Differencing ts1")

arima1\_manual <- Arima(train1, order = c(1, 0, 1),

seasonal = list(order = c(1, 1, 1), period = 12))

summary(arima1\_manual)

tsdisplay(residuals(arima1\_manual), main = "Residuals: ARIMA(1,0,1)(1,1,1)[12]")

Box.test(residuals(arima1\_manual), lag = 24, type = "Ljung-Box")

forecast\_arima1 <- forecast(arima1\_manual, h = 12)

acc1\_arima <- accuracy(forecast\_arima1$mean, test1)

cat("\nARIMA Model Accuracy ts1 ARIMA(1,0,1)(1,1,1):\n")

print(acc1\_arima)

arima1\_2\_manual <- Arima(train1, order = c(2, 0, 1),

seasonal = list(order = c(1, 1, 1), period = 12))

summary(arima1\_2\_manual)

tsdisplay(residuals(arima1\_2\_manual), main = "Residuals: ARIMA(2,0,1)(1,1,1)[12]")

Box.test(residuals(arima1\_2\_manual), lag = 24, type = "Ljung-Box")

forecast\_arima1\_2 <- forecast(arima1\_2\_manual, h = 12)

acc1\_2\_arima <- accuracy(forecast\_arima1\_2$mean, test1)

cat("\nARIMA Model Accuracy ts1 ARIMA(2,0,1)(1,1,1):\n")

print(acc1\_2\_arima)

cat("\nARIMA Model Accuracy ts1 ARIMA(1,0,1)(1,1,1):\n")

print(acc1\_arima)

cat("\nARIMA Model Accuracy ts1 ARIMA(2,0,1)(1,1,1):\n")

print(acc1\_2\_arima)

#ts2

adf.test(ts2)

tsdisplay(ts2, main = "Raw Series ACF/PACF")

tsdisplay(ts2, main = "Raw Series ACF/PACF")

tsdisplay(diff(ts2, lag = 12), main = "After Seasonal Differencing ts2")

# two models for ts2

arima2\_1\_manual <- Arima(train2, order = c(1,0, 0),

seasonal = list(order = c(0, 1, 1), period = 12))

summary(arima2\_1\_manual)

tsdisplay(residuals(arima2\_1\_manual), main = "Residuals: ARIMA(1,0,0)(0,1,1)[12]")

Box.test(residuals(arima2\_1\_manual), lag = 24, type = "Ljung-Box")

forecast\_arima2\_1 <- forecast(arima2\_1\_manual, h = 12)

acc2\_1\_arima <- accuracy(forecast\_arima2\_1$mean, test2)

cat("\nARIMA Model Accuracy ts2 ARIMA(1,0,0)(0,1,1):\n")

print(acc2\_1\_arima)

arima2\_manual <- Arima(train2, order = c(2,0, 0),

seasonal = list(order = c(0, 1, 1), period = 12))

summary(arima2\_manual)

tsdisplay(residuals(arima2\_manual), main = "Residuals: ARIMA(2,0,0)(0,1,1)[12]")

Box.test(residuals(arima2\_manual), lag = 24, type = "Ljung-Box")

forecast\_arima2 <- forecast(arima2\_manual, h = 12)

acc2\_arima <- accuracy(forecast\_arima2$mean, test2)

cat("\nARIMA Model Accuracy ts2 ARIMA(2,0,0)(0,1,1) :\n")

print(acc2\_arima)

cat("\nARIMA Model Accuracy ts2 ARIMA(1,0,0)(0,1,1):\n")

print(acc2\_1\_arima)

cat("\nARIMA Model Accuracy ts2 ARIMA(2,0,0)(0,1,1) :\n")

print(acc2\_arima)

#REGRESSION

evaluate\_time\_series\_regression <- function(train, test, h = 12, name = "TS Regression") {

L1 <- lag(train, -1)

L12 <- lag(train, -12)

unified\_start <- c(start(train)[1] + 1, start(train)[2])

Y\_trimmed <- window(train, start = unified\_start)

L1\_trimmed <- window(L1, start = unified\_start)

L12\_trimmed <- window(L12, start = unified\_start)

min\_len <- min(length(Y\_trimmed), length(L1\_trimmed), length(L12\_trimmed))

Y\_trimmed <- window(Y\_trimmed, end = time(Y\_trimmed)[min\_len])

L1\_trimmed <- window(L1\_trimmed, end = time(L1\_trimmed)[min\_len])

L12\_trimmed <- window(L12\_trimmed, end = time(L12\_trimmed)[min\_len])

trend\_vals <- 1:min\_len

season\_vals <- cycle(Y\_trimmed)

data\_train <- data.frame(

Y = as.numeric(Y\_trimmed),

L1\_Y = as.numeric(L1\_trimmed),

L12\_Y = as.numeric(L12\_trimmed),

trend = trend\_vals,

season = factor(season\_vals)

)

fit <- lm(Y ~ trend + season + L1\_Y + L12\_Y, data = data\_train)

last\_L1 <- tail(as.numeric(train), 1)

last\_L12 <- as.numeric(train[(length(train) - 12 + 1):(length(train) - 12 + h)])

trend\_future <- (max(trend\_vals) + 1):(max(trend\_vals) + h)

season\_future <- cycle(ts(rep(NA, h), start = time(test)[1], frequency = frequency(train)))

new\_data <- data.frame(

trend = trend\_future,

season = factor(season\_future),

L1\_Y = rep(last\_L1, h),

L12\_Y = last\_L12

)

preds <- predict(fit, newdata = new\_data)

preds\_ts <- ts(preds, start = start(test), frequency = frequency(test))

cat("\nForecast Accuracy for Lag1 and Lag12", name, "\n")

print(accuracy(preds\_ts, test))

fit\_simpler <- lm(Y ~ trend + season + L12\_Y, data = data\_train)

forecast\_simpler\_vals <- predict(fit\_simpler, newdata = new\_data)

cat("\nForecast Accuracy for simpler model Lag12", name, "\n")

print(accuracy(forecast\_simpler\_vals, test))

fit\_tslm <- tslm(train ~ trend + season)

forecast\_tslm <- forecast(fit\_tslm, h = h)

cat("\nTSLM Accuracy for", name, "\n")

print(accuracy(forecast\_tslm, test))

}

evaluate\_time\_series\_regression(train1, test1, h = 12, name = "Regression on ts1")

evaluate\_time\_series\_regression(train2, test2, h = 12, name = "Regression on ts2")

# Final accuracy

acc\_ts1 <- run\_es\_models\_on\_split(train1, test1, h = 12, ts\_name = "ts1")

acc\_sma\_ts1 <- evaluate\_sma\_model(train1, test1, h = 12, k = 3)

evaluate\_auto\_arima(train1, test1, h = 12, model\_name = "Auto ARIMA on ts1")

evaluate\_time\_series\_regression(train1, test1, h = 12, name = "Regression on ts1")

accuracy(forecast\_arima1$mean, test1)

acc\_ts2 <- run\_es\_models\_on\_split(train2, test2, h = 12, ts\_name = "ts2")

acc\_sma\_ts2 <- evaluate\_sma\_model(train2, test2, h = 12, k = 3)

evaluate\_auto\_arima(train2, test2, h = 12, model\_name = "Auto ARIMA on ts2")

evaluate\_time\_series\_regression(train2, test2, h = 12, name = "Regression on ts2")

accuracy(forecast\_arima2$mean, test2)

#BEST FORECASTS

arima1\_manual\_final <- Arima(ts1, order = c(1, 0, 1),

seasonal = list(order = c(1, 1, 1), period = 12))

length(ts1)

start(ts1)

forecast\_arima1\_final <- forecast(arima1\_manual\_final, h = 14)

print(forecast\_arima1\_final)

length(forecast\_arima1\_final)

plot(forecast\_arima1\_final, main = paste("ARIMA Forecast ts1 (h = 14", ")"))

# 4. Holt-Winters Multiplicative

hw\_model <- HoltWinters(ts2, seasonal = "multiplicative")

hw\_forecast <- forecast(hw\_model, h = 14)

plot(hw\_forecast, main = paste("Holt-Winters Mult ts2 (h=14)"))

print(hw\_forecast)

#FORECASTS OTHERS

run\_and\_plot\_simple <- function(ts\_data, ts\_name = "Series", h = 12) {

# 1. Naive

naive\_model <- naive(ts\_data, h = h)

plot(naive\_model, main = paste(ts\_name, "- Naive"))

lines(ts\_data, col = "blue", lty = 2)

# 2. Seasonal Naive

snaive\_model <- snaive(ts\_data, h = h)

plot(snaive\_model, main = paste(ts\_name, "- Seasonal Naive"))

lines(ts\_data, col = "blue", lty = 2)

# 3. ETS (MMM)

ets\_model <- ets(ts\_data, model = "MMM")

ets\_forecast <- forecast(ets\_model, h = h)

plot(ets\_forecast, main = paste(ts\_name, "- ETS (MMM)"))

lines(ts\_data, col = "blue", lty = 2)

# 4. Holt-Winters Multiplicative

hw\_model <- HoltWinters(ts\_data, seasonal = "multiplicative")

hw\_forecast <- forecast(hw\_model, h = h)

plot(hw\_forecast, main = paste(ts\_name, "- Holt-Winters Mult"))

lines(ts\_data, col = "blue", lty = 2)

# 5. Time Series Regression (TSLM)

tslm\_model <- tslm(ts\_data ~ trend + season)

tslm\_forecast <- forecast(tslm\_model, h = h)

plot(tslm\_forecast, main = paste(ts\_name, "- Time Series Regression"))

lines(ts\_data, col = "blue", lty = 2)

# 6. Auto ARIMA

arima\_model <- auto.arima(ts\_data)

arima\_forecast <- forecast(arima\_model, h = h)

plot(arima\_forecast, main = paste(ts\_name, "- Auto ARIMA"))

lines(ts\_data, col = "blue", lty = 2)

}

# --- Run for ts1 ---

run\_and\_plot\_simple(ts1, ts\_name = "ts1", h = 14)

# --- Run for ts2 ---

run\_and\_plot\_simple(ts2, ts\_name = "ts2", h = 14)