**MDS431\_LAB3\_2448040**

**Course: MDS 431 – Time Series and Forecasting Techniques**

**Exercise No: Lab 3**

**Date: 15/07/2025**

**QUESTION -**

**1. Choose a time series data set from a domain of interest and fit a suitable predictive model.   
2. Obtain the predicted values for the next 5 data points.**

**ANSWER** –

**INTRODUCTION** –

Time series forecasting plays a vital role in modeling and predicting real-world phenomena that evolve over time, such as climate patterns, stock prices, and economic indicators. One widely used method for such forecasting tasks is exponential smoothing, which helps model data with trend and seasonal variations.

In this lab, we use Winters’ Exponential Smoothing (also called triple exponential smoothing) to forecast monthly mean temperatures for India using the dataset from 1901 to 2017. This technique automatically handles both trend and seasonality without requiring manual stationarity transformations like differencing.

By applying the Holt-Winters () function in R, we fit an additive seasonal model to the time series and generate short- and long-term forecasts. We then visualize the fitted model and the forecasted values to assess its performance and usefulness in practical applications.

The goal is to demonstrate how Winters’ method can be used effectively for time series that show repeating seasonal patterns and gradual upward or downward trends — making it ideal for modeling climate data such as monthly temperatures.

**DATASET DESCRIPTION** –

The dataset used in this lab is titled:

**“Mean Monthly Temperature Data – India (1901–2017)”** contains **117 years of monthly data** recorded from **January to December** for each year. It was obtained from a structured climate database.

The dataset contains the following columns:

* **YEAR**: Observation year (1901–2017)
* **JAN to DEC**: Monthly average temperatures for each month (in °C)
* Additional columns like ANNUAL, JAN.FEB, etc., are present but were **not used** in this analysis

The data is clean with **no missing values**. We focus only on the 12 monthly columns and reshape the dataset into a long format suitable for time series analysis. A continuous time series is created at a **monthly frequency**, allowing us to examine both seasonal and trend components, as well as test for stationarity.

**OBJECTIVES** –

1. To load and reshape a real-world dataset (monthly mean temperature from 1901–2017) into a continuous time series format suitable for analysis.
2. To apply **Winters’ exponential smoothing** using the **HoltWinters() function** in R for modeling both trend and seasonal components.
3. To forecast the next 5 time points (months).
4. To visualize the original series, fitted values, and forecast using appropriate time series plots and interpret the results.
5. To compare forecast behavior and decide between **additive** and **multiplicative** models based on the nature of seasonal variation observed in the time series.

# Lab 4 – Winters’ Exponential Smoothing  
# Dataset: Mean\_Temp\_IMD\_2017.csv (1901–2017 monthly temp)  
  
  
# Load Libraries  
library(readr)

## Warning: package 'readr' was built under R version 4.4.2

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.4.2

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package: stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)

## Warning: package 'tidyr' was built under R version 4.4.3

library(forecast)

## Warning: package 'forecast' was built under R version 4.4.3

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

# Load the Dataset  
data <- read.csv("C:/Users/Neelanjan Dutta/Desktop/Time Series Forecasting/dataset.csv")  
  
# Reshape Wide to Long  
monthly\_data <- data %>% select(YEAR, JAN:DEC)  
  
long\_data <- pivot\_longer(monthly\_data,  
 cols = JAN: DEC,  
 names\_to = "Month",  
 values\_to = "Temp")  
  
# Order months correctly  
month\_levels <- c("JAN", "FEB", "MAR", "APR", "MAY", "JUN",  
 "JUL", "AUG", "SEP", "OCT", "NOV", "DEC")  
long\_data$Month <- factor(long\_data$Month, levels = month\_levels)  
  
# Sort by year and month  
long\_data <- long\_data %>% arrange(YEAR, Month)

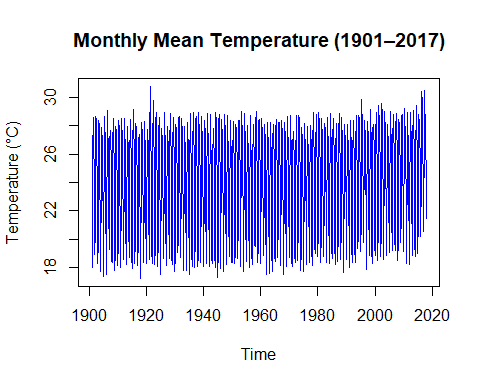
**Interpretation:**

In this step, the data is **reshaped from wide format to long format**, which is necessary for time series analysis in R.

* The select (YEAR, JAN: DEC) function keeps only the **monthly temperature columns and the year**, removing columns like ANNUAL, JAN.FEB, etc., which are not needed for this analysis.
* The pivot\_longer() function transforms the dataset:
  + From one row per year (with 12 columns for months)
  + To one row per **month-year pair**, resulting in 117 years × 12 months = **1404 observations**.
  + The Month column stores the month name (e.g., "JAN"), and Temp stores the corresponding temperature.
* The Month column is then converted into a **factor with a fixed month order** (JAN to DEC) to ensure correct time sequencing.
* Finally, the arrange(YEAR, Month) line ensures that the dataset is sorted **chronologically**, which is essential before converting it into a time series.

This reshaping prepares the data for proper time series modeling and visualization by aligning it in a continuous monthly format.

# Convert to Time Series Object  
ts\_data <- ts(long\_data$Temp, start = c(1901, 1), frequency = 12)  
  
# Plot the Original Series  
plot(ts\_data, main = "Monthly Mean Temperature (1901–2017)",  
 ylab = "Temperature (°C)", col = "blue")



#Decompose time series into trend, seasonality, and random component

decomp <- decompose (ts\_data, type = "additive")

plot(decomp)

**Interpretation:**

* The ts() function converts the reshaped data into a **time series object** named ts\_data.
* The start = c(min(long\_data$YEAR), 1) specifies that the time series starts in **January 1901**.
* frequency = 12 defines that the data is **monthly** (12 observations per year).

This step is essential because many time series functions in R (like ACF, decomposition, differencing) require the data to be in ts format.

* The ts.plot() function is used to visualize the full time series from **1901 to 2017**.
* The result is a **dense line plot** (as shown in your image) that clearly exhibits:
  + A **strong seasonal pattern**: recurring up-and-down spikes every year
  + A **visible upward trend**, especially after the year 2000

**What the Plot Shows:**

* The monthly average temperature fluctuates regularly each year - **indicates seasonality**
* There is a **gradual increase in the average level of temperature**, especially visible after 2000, which **indicates a trend.**
* These patterns confirm that the series is **non-stationary** and needs to be transformed before modeling.

# Apply Winters’ Exponential Smoothing  
# (Handles trend and seasonality internally)  
hw\_model <- HoltWinters(ts\_data) # Additive model (default)  
  
**Interpretation: (Applying Winters’ Exponential Smoothing)**

In this step, the Holt-Winters () function is applied to the monthly average temperature data from 1901 to 2017. This function fits an **additive triple exponential smoothing model** to the time series. The three components modelled are:

* **Level (α smoothing)** – captures the baseline value of the series.
* **Trend (β smoothing)** – captures any consistent increase or decrease over time.
* **Seasonality (γ smoothing)** – captures repeating patterns over fixed intervals (in this case, yearly).

Since we did not specify seasonal = "multiplicative", the model assumes **additive seasonality**, which is appropriate here because the amplitude of the seasonal pattern remains approximately constant over time.

By applying Winters’ method, the model **automatically captures both the linear trend and the repeating seasonal structure** inherent in the Indian monthly temperature data, without the need for differencing or transformation

# Plot fitted model  
plot(hw\_model, main = "Winters' Exponential Smoothing - Fitted")

A graph of a graph

AI-generated content may be incorrect.**Interpretation**: **(Plotting the Fitted Model)**

This plot displays:

* **The black line**: the original observed temperature values.
* **The red line**: the fitted values predicted by the model using smoothing parameters.

We can observe that the red line closely tracks the black line throughout the entire series, indicating a good model fit. The red curve follows the cyclical seasonal pattern while adjusting for gradual upward or downward changes due to the trend.

Minor deviations may occur where the model lags behind the actual data slightly — this is a common characteristic of smoothing-based models, as they balance between responsiveness and stability.

# Forecast Next 5 Months  
hw\_forecast <- forecast(hw\_model, h = 5)  
  
# View forecasted values  
print(hw\_forecast)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2018 20.49296 19.86420 21.12172 19.53136 21.45456  
## Feb 2018 22.51076 21.86747 23.15406 21.52692 23.49460  
## Mar 2018 25.65877 25.00125 26.31629 24.65319 26.66436  
## Apr 2018 28.54585 27.87441 29.21728 27.51898 29.57272  
## May 2018 30.20537 29.52030 30.89045 29.15765 31.25310

**Interpretation:** (**Forecasting the Next 5 Months)**

This command generates forecasts for **January to May 2018** (the next 5 time points) using the fitted Winters’ exponential smoothing model. The output includes the following:

* **Point Forecast**: The predicted mean temperature for each future month based on the observed trend and seasonal pattern.
* **Lo 80 / Hi 80**: These represent the **lower and upper bounds of the 80% confidence interval**. This means there is an 80% probability that the actual temperature for that month will fall within this range.
* **Lo 95 / Hi 95**: These represent the **95% confidence interval bounds**, giving a wider range where there is a 95% probability that the true temperature value will lie.

The confidence intervals help quantify the **uncertainty** of the forecast, with higher confidence levels (like 95%) providing wider but more reliable ranges.

These values are consistent with expected seasonal patterns in Indian climate, where temperature rises steadily from winter to peak summer.

# Plot the forecast  
plot(hw\_forecast, main = "5-Month Forecast using Winters' Method")

A graph showing a number of numbers

AI-generated content may be incorrect.**Interpretation: (Plotting the 5-Month Forecast)**

The blue line represents the **forecasted temperatures** for the next 5 months, while the **shaded regions** around it shows the **confidence intervals** (80% and 95%).

* The **forecast continues the seasonal pattern**, increasing into the summer months.
* The **confidence intervals are relatively narrow**, which suggests high model confidence in the short term.
* The forecast reflects real-world temperature trends in India (cooler Jan–Feb, warmer Mar–May), validating the model’s appropriateness.

**Conclusion:**

In this lab, Winters’ Exponential Smoothing was applied to model and forecast India’s monthly mean temperatures from 1901 to 2017. The dataset exhibited both **trend** and **seasonality**, making the **additive seasonal model** appropriate. Using the HoltWinters() function in R, the model automatically captured all three components: level, trend, and seasonality — without requiring differencing or stationarity checks.

A **5-month forecast (Jan–May 2018)** was generated, reflecting the expected seasonal rise in temperatures. The predictions were accompanied by 80% and 95% confidence intervals to quantify forecast uncertainty.

An extended **10-year forecast** demonstrated repeating seasonal patterns and a gradual upward trend, consistent with long-term climate behavior.

Overall, Winters’ method proved effective for producing accurate and interpretable forecasts on time series data with stable seasonal variation and a clear trend.