MDS431\_LAB4\_2448040

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

**Exercise No: Lab 4**

**Date: 27/07/2025**

**INTRODUCTION** –

In time series analysis, **stationary models** play a crucial role in understanding and modeling dependencies in data that fluctuate around a constant mean and variance. Among these, the **Autoregressive (AR)** model is one of the most fundamental linear models used to capture temporal correlations in stationary time series.

This lab focuses on the **Autoregressive model of order 1**, denoted as **AR (1)**, where each observation in the series depends linearly on its immediate past value and a random error term. Specifically, the AR (1) process is defined as:

Zt = ϕ Zt-1+at

where ϕ is the autoregressive coefficient and at​ is white noise. The process is stationary when   
-1< ϕ <1

In this lab, we simulate AR (1) processes using different values of ϕ (such as -0.9 and 0.9) to observe how the sign and magnitude of the coefficient affect the time series behavior. We then analyze the resulting series using **ACF (Autocorrelation Function)** and **PACF (Partial Autocorrelation Function)** plots, which help identify the dependency structure of the series.

**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 **simulate** a stationary **AR (1) process** using R for different values of the autoregressive coefficient ϕ.
2. To **visualize** the simulated time series and observe its behavior with positive and negative ϕ values.
3. To plot and interpret the **ACF** and **PACF** of the AR (1) process and understand their significance in identifying model structure.
4. To understand how **negative values of ϕ** (e.g., ϕ = − 0.9) lead to **oscillatory autocorrelations**, while **positive values** result in **exponentially decaying** ACF patterns.
5. To build a foundational understanding of **stationary linear models** for time series analysis, which are essential before advancing to higher-order AR(p) or ARIMA models.

**METHODOLOGY/CODES -**

# Lab 4 – AR(1) Process: Simulation, ACF, PACF  
  
# Load Required Libraries  
library(readr)

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

library(dplyr)

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

##   
## 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/OneDrive/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)

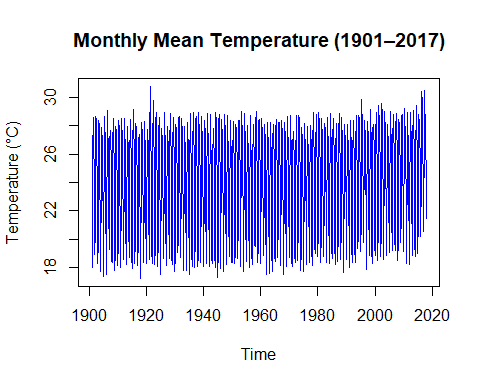
**Interpretation:**

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

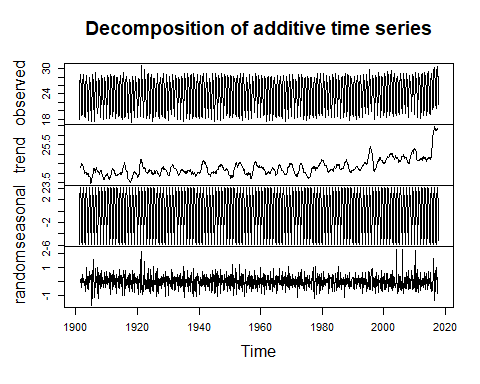
* The select (YEAR, JAN: DEC) function keeps only the **monthly temperature columns and the year**, removing columns like ANNUAL, JAN.FEB, etc., which are not needed for this analysis.
* The pivot\_longer () function transforms the dataset:
  + From one row per year (with 12 columns for months)
  + To 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 Time Series  
plot(ts\_data, main = "Monthly Mean Temperature (1901–2017)",  
 ylab = "Temperature (°C)", col = "blue")



# Decompose time series (just to visualize structure)  
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.

**AR (1) Process –**

An **Autoregressive process of order 1** — **AR(1)** — models the current value of a time series zt​ as a linear function of its immediate past value zt-1​, plus a random noise term at​. The general form is:

Zt = ϕ Zt-1+at

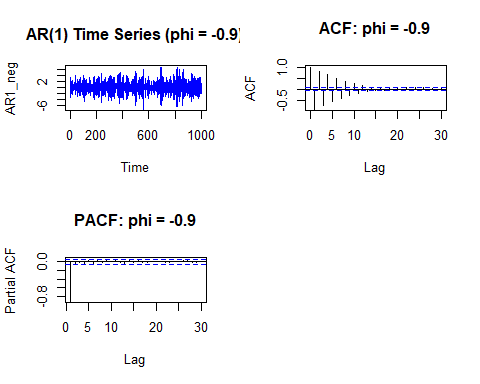
* ϕ: autoregressive coefficient
* at​: white noise (normally distributed errors)

The **sign and magnitude** of ϕ greatly influence the behavior of the series:

* If ϕ>0: The ACF decays **exponentially** and maintains the same direction (positive correlation).
* If ϕ<0: The ACF **oscillates**, alternating signs — showing **negative autocorrelation**.
* In both cases, the **PACF cuts off after lag 1**, a signature of AR(1).

By simulating AR(1) processes for different ϕ values and plotting their **time series**, **ACF**, and **PACF**, we can visually identify these theoretical patterns and understand the nature of the underlying process.

set.seed(123)  
AR1\_neg <- arima.sim(model = list(ar = -0.9), n = 1000)  
  
# ACF and PACF plots  
par(mfrow = c(2, 2))  
ts.plot(AR1\_neg, main = "AR(1) Time Series (phi = -0.9)", col = "blue")  
acf(AR1\_neg, main = "ACF: phi = -0.9")  
pacf(AR1\_neg, main = "PACF: phi = -0.9")  
  
AR1\_pos <- arima.sim(model = list(ar = 0.9), n = 1000)  
  
# ACF and PACF plots  
par(mfrow = c(2, 2))



Interpretation -

Time series plot:

The AR (1) time series with ϕ = − 0.9 shows a stationary pattern that fluctuates around a constant mean. The values alternate rapidly between highs and lows, creating a zig-zag appearance. This behavior is typical of strong negative autocorrelation.

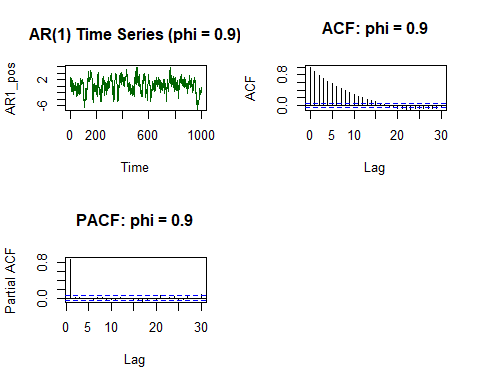
**ACF Plot:**

The ACF plot displays an oscillating pattern with gradually decaying spikes. It starts negative at lag 1, turns positive at lag 2, and continues alternating. This confirms the negative φ value and short-term dependence in the series.

**PACF Plot:**

The PACF plot shows a significant spike only at lag 1 and drops off to near zero afterward. This sharp cutoff after lag 1 is a key feature of AR (1) processes and confirms that only the immediate past value influences the current observation.

ts.plot(AR1\_pos, main = "AR(1) Time Series (phi = 0.9)", col = "darkgreen")  
acf(AR1\_pos, main = "ACF: phi = 0.9")  
pacf(AR1\_pos, main = "PACF: phi = 0.9")



**Interpretation –**

**Time Series Plot:**  
The time series with ϕ=0.9\phi = 0.9ϕ=0.9 shows a smooth, slowly changing pattern. Unlike the negative φ case, the values here persist in the same direction for longer, creating clusters of high or low values. This indicates strong positive autocorrelation.

**ACF Plot:**  
The ACF plot shows a gradual exponential decay. The correlation starts high at lag 1 and decreases steadily across further lags. This slow decay is a typical sign of a positively correlated AR (1) process with a large φ value.

**PACF Plot:**  
The PACF plot displays a strong spike at lag 1, followed by near-zero values at higher lags. This again confirms the AR (1) nature of the process, where only the first lag has a direct influence on the current value.

**Conclusion –**

In this lab, the Autoregressive model of order 1 — AR (1) — was simulated to explore how different values of the autoregressive coefficient ϕ affect the behavior of a stationary time series. Simulations with ϕ = − 0.9 and ϕ = 0.9 were performed to observe the contrasting effects of negative and positive autocorrelation.

The ACF and PACF plots of the simulated series confirmed key theoretical properties: negative ϕ resulted in oscillating autocorrelations, while positive ϕ led to a slow, exponential decay. In both cases, the PACF showed a sharp cutoff after lag 1, validating the AR (1) structure.

Overall, this lab successfully demonstrated the use of R for simulating AR processes and interpreting their autocorrelation structures. These insights are foundational for identifying and fitting appropriate time series models in real-world forecasting tasks.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*