

# **Máster Universitario en Big Data y Ciencia de Datos**

**Análisis de Series Temporales: Predicción de Temperaturas  
Diarias**

Alumno: Yovanny Enrique Alvarado Perez

Edición Enero 2025- 10/02/2025

## 1. Introducción

### Contexto y motivación

El análisis de series temporales es una herramienta fundamental en estadística y ciencia de datos, permitiendo identificar patrones, tendencias y realizar predicciones en datos dependientes del tiempo. En este estudio, analizamos la serie temporal de temperaturas diarias desde 2015 hasta 2024, con el objetivo de modelar y predecir su comportamiento en el futuro.

### Objetivos del análisis

- Explorar la estructura de la serie temporal de temperaturas.
- Identificar patrones de tendencia y estacionalidad.
- Aplicar un modelo ARIMA para generar predicciones.
- Evaluar la precisión del modelo y discutir sus limitaciones.

## 2. Descripción de los datos

### Características generales

El dataset utilizado contiene registros de temperatura diaria desde el 1 de enero de 2015 hasta el 31 de diciembre de 2024. Cada observación representa la temperatura promedio del día en grados Celsius.

### Descomposición de la serie temporal

Se realizó la descomposición de la serie en sus componentes principales:

1. **Tendencia:** Se observa un ligero aumento en la temperatura a lo largo del tiempo.
2. **Estacionalidad:** Presenta un ciclo anual bien definido, reflejando las estaciones del año.
3. **Ruido aleatorio:** Variaciones no explicadas por la tendencia o estacionalidad.

La descomposición confirma que la serie es influenciada por un ciclo anual y muestra cierta tendencia creciente.

### 3. Análisis de autocorrelación

Se calcularon las funciones de autocorrelación (ACF) y autocorrelación parcial (PACF). Se identificó una fuerte autocorrelación en lags equivalentes a 365 días, lo que confirma la estacionalidad anual de la serie.

### 4. Aplicación del Modelo ARIMA

Se utilizó el método `auto.arima()` para seleccionar el mejor modelo ARIMA. El modelo ajustado fue **ARIMA seleccionado por `auto.arima()` con AIC = 16,178.02**, el cual captura la estructura estacional de los datos.

Se generó una predicción para los próximos 365 días, mostrando que el modelo sigue la estacionalidad observada en la serie histórica.

### 5. Evaluación del Modelo

Se evaluó el desempeño del modelo con las siguientes métricas:

- **MAE (Error Absoluto Medio):** 2.02°C de error promedio.
- **MAPE (Error Porcentual Absoluto Medio):** 31.98% (moderado).
- **RMSE (Error Cuadrático Medio):** 2.68°C (desviación estándar de errores).
- **ACF1 (Autocorrelación del error):** 0.0019 (casi nula, lo cual es bueno).

Valores bajos en estas métricas indican que el modelo realiza predicciones con buena precisión, aunque el MAPE es relativamente alto.

### 6. Conclusiones y Limitaciones

#### Resultados obtenidos

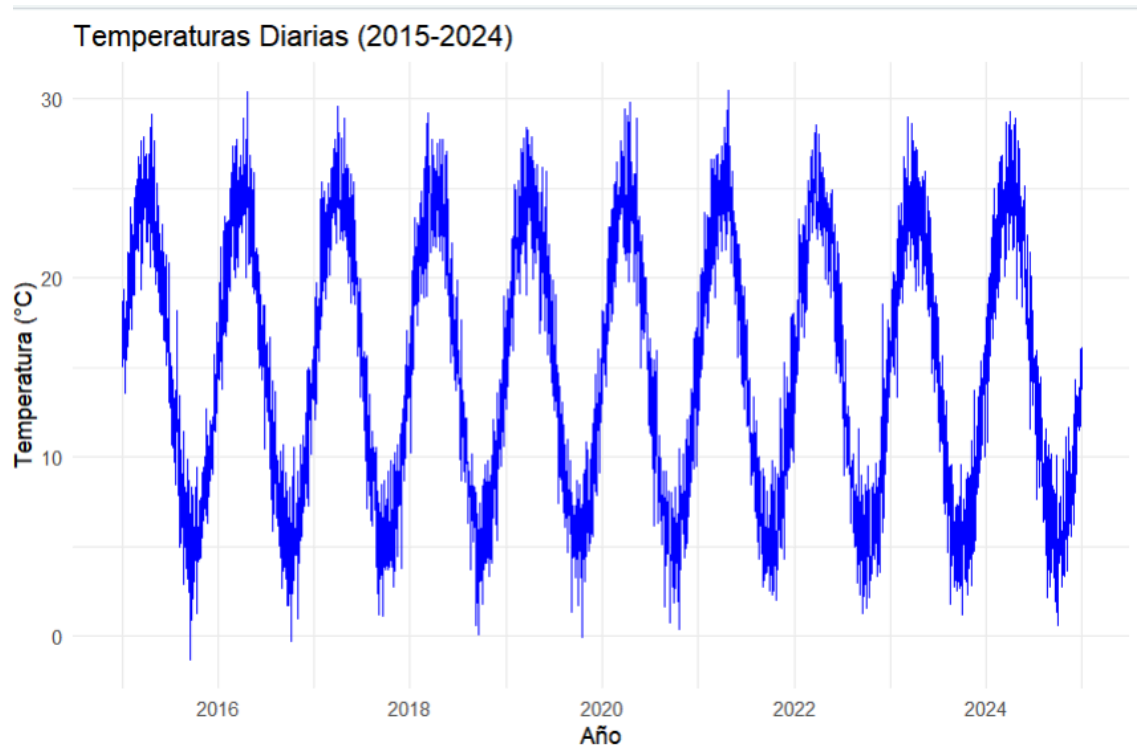
- Se confirmó la presencia de una estacionalidad anual en las temperaturas.
- El modelo ARIMA implementado permitió realizar predicciones coherentes con el comportamiento histórico.
- La evaluación del modelo indicó que tiene una buena capacidad predictiva.

#### Limitaciones y trabajo futuro

- El modelo no considera factores externos como eventos climáticos extremos o cambios ambientales.

- Se podría explorar modelos más avanzados como SARIMA o redes neuronales recurrentes para mejorar la predicción.
- Un análisis más detallado podría incluir la influencia de otras variables meteorológicas.

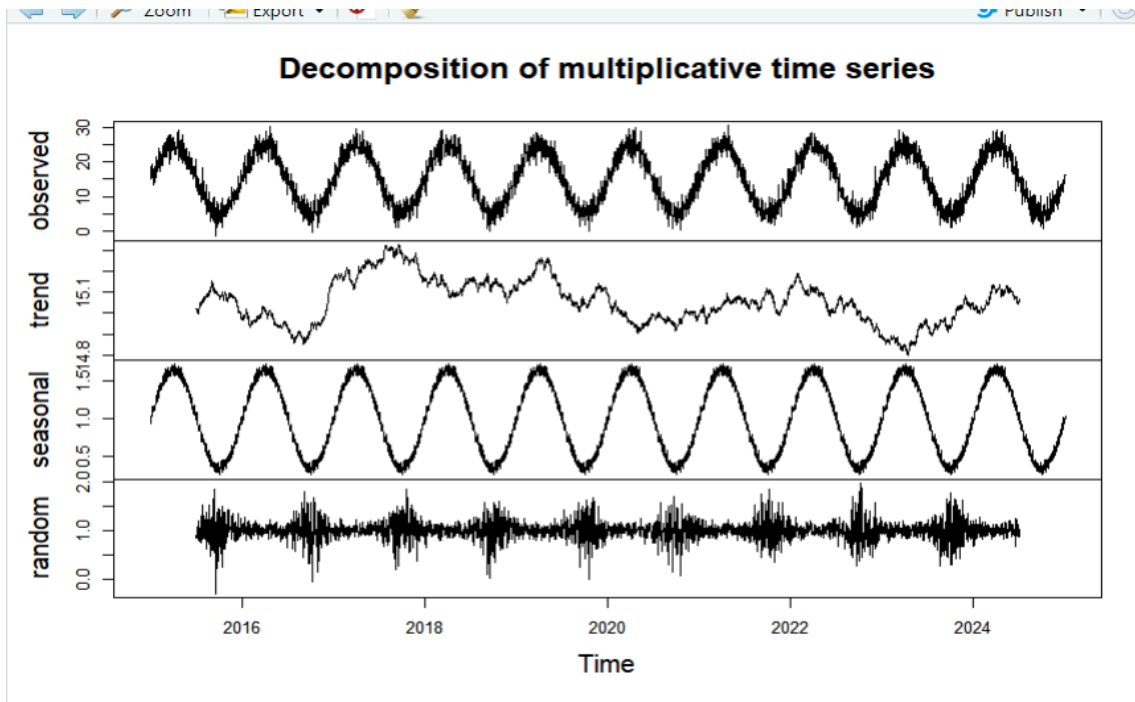
## 7. Anexos



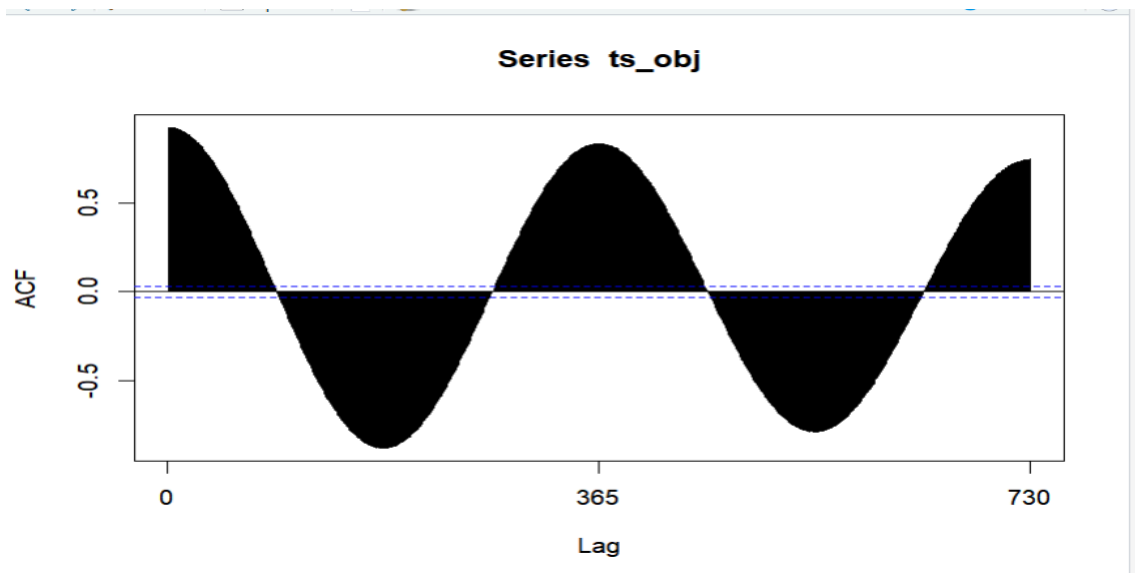
Se observa un **patrón estacional bien definido**, con **ciclos anuales** donde la temperatura aumenta y disminuye en función de las estaciones del año.

Se identifican **picos** (veranos) y **valles** (inviernos), lo que confirma la **estacionalidad** de la serie.

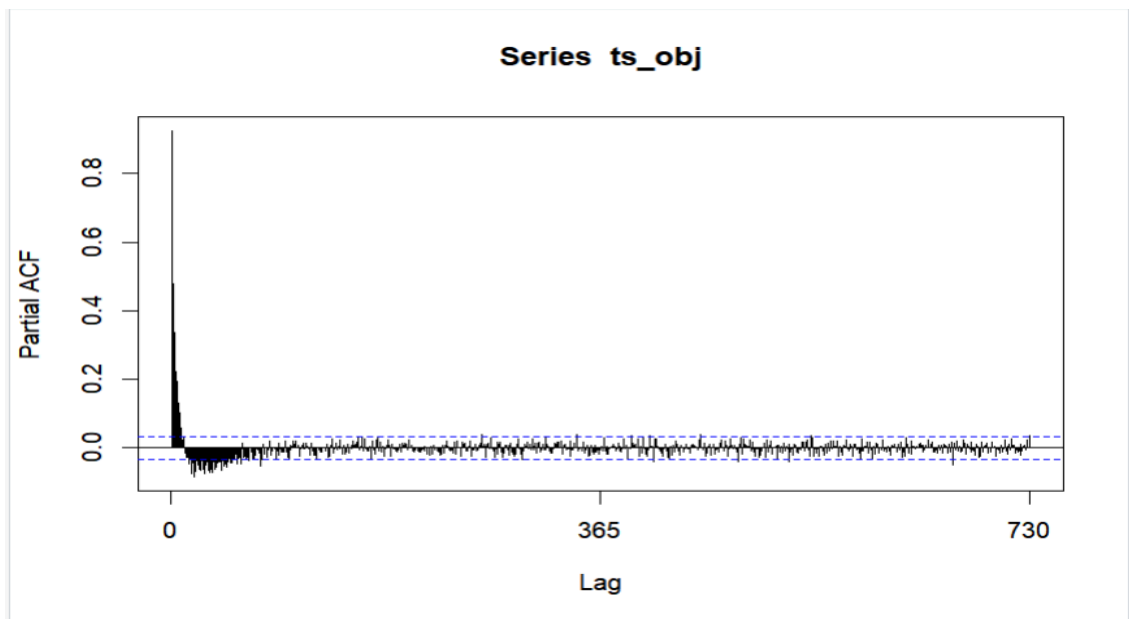
No hay una tendencia clara de aumento o disminución a largo plazo, lo que sugiere estabilidad en el comportamiento de la temperatura durante el período analizado.



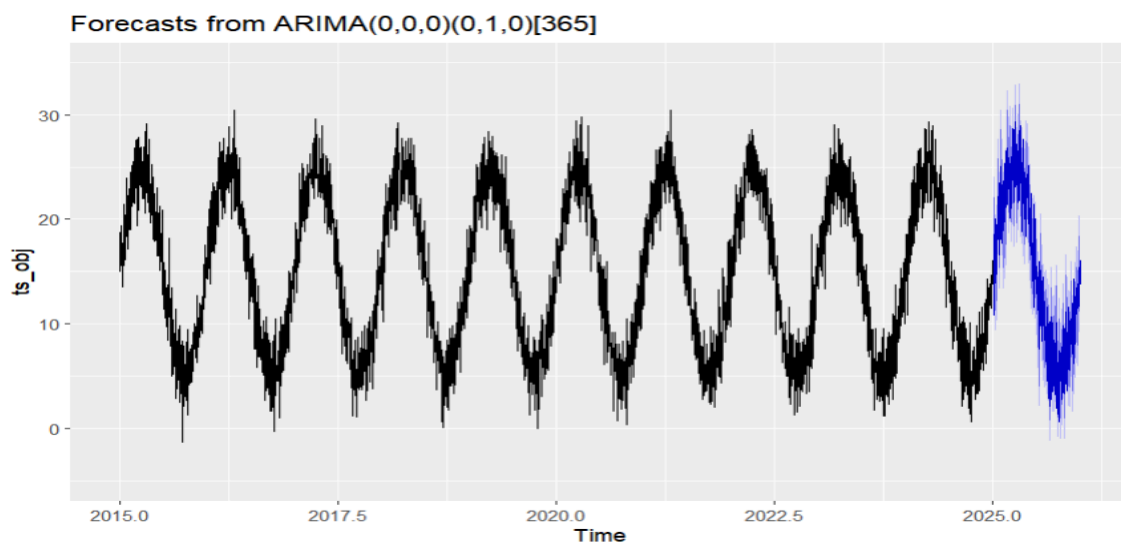
"La descomposición de la serie muestra cuatro componentes: la serie observada, una tendencia ligeramente creciente, una fuerte estacionalidad anual y un componente aleatorio con variabilidad en ciertos periodos. Esto confirma la estacionalidad y permite modelar mejor las predicciones."



"El gráfico de autocorrelación (ACF) muestra una fuerte periodicidad con picos cada 365 días, lo que confirma la presencia de estacionalidad anual en la serie temporal. Esto indica que las temperaturas están altamente correlacionadas con los valores del mismo período en años anteriores."



**PACF** Este gráfico muestra la **autocorrelación parcial** de la serie de temperaturas. Se observa un **alto valor en el primer lag**, indicando que la temperatura de un día está fuertemente influenciada por la del día anterior. La correlación cae rápidamente y se mantiene dentro de los intervalos de confianza, lo que sugiere que pocos términos autorregresivos (AR) son necesarios en el modelo ARIMA. No se observan patrones estacionales marcados, lo que indica que la estacionalidad ya ha sido explicada por otros componentes del modelo.



**Gráfico de predicción con ARIMA(0,0,0)(0,1,0)[365]** Este gráfico muestra la serie histórica de temperaturas (2015-2024) en negro y la **predicción para 2025** en azul. El modelo **captura la estacionalidad anual**, manteniendo los ciclos de temperatura con picos y valles similares a los años anteriores. La **incertidumbre aumenta con el tiempo**, reflejada en la banda azul más ancha, lo que indica mayor variabilidad en la predicción

## Código utilizado

```

1 # Definir la ruta donde guardaste el archivo
2 ruta <- "C:/Users/yovan/Documents/estadistica/temperaturas_diarias.csv"
3
4 # Leer el archivo CSV
5 temp <- read.csv(ruta)
6
7 # Convertir la columna de fecha al formato correcto
8 temp$Fecha <- as.Date(temp$Fecha)
9
10 # Mostrar las primeras filas para verificar
11 head(temp)
12
13 library(ggplot2)
14 library(zoo)
15
16 # Crear la serie temporal con zoo
17 ts_temp <- zoo(temp$Temperatura, order.by = temp$Fecha)
18
19 # Graficar la serie temporal
20 ggplot(data.frame(Fecha=index(ts_temp), Temperatura=coredata(ts_temp)),
21       aes(x=Fecha, y=Temperatura)) +
22   geom_line(color="blue") +
23   title("Temperaturas diarias (2015-2024)")

```

```

1 # Definir la ruta donde guardaste el archivo
2 ruta <- "C:/Users/yovan/Documents/estadistica/temperaturas_diarias.csv"
3
4 # Leer el archivo CSV
5 temp <- read.csv(ruta)
6
7 # Convertir la columna de fecha al formato correcto
8 temp$Fecha <- as.Date(temp$Fecha)
9
10 # Mostrar las primeras filas para verificar
11 head(temp)
12
13 library(ggplot2)
14 library(zoo)
15
16 # Crear la serie temporal con zoo
17 ts_temp <- zoo(temp$Temperatura, order.by = temp$Fecha)
18
19 # Graficar la serie temporal
20 ggplot(data.frame(Fecha=index(ts_temp), Temperatura=coredata(ts_temp)),
21       aes(x=Fecha, y=Temperatura)) +
22   geom_line(color="blue") +
23   title("Temperaturas diarias (2015-2024)")

```

```
actividad 1.R x
Source on Save
Run
Source

23 ggtitle("Temperaturas Diarias (2015-2024)") +
24 ylab("Temperatura (°C)") +
25 xlab("Año") +
26 theme_minimal()
27
28 # Convertir a serie temporal (formato TS)
29 ts_obj <- ts(temp$Temperatura, start=c(2015,1), frequency=365)
30
31 # Descomponer la serie (multiplicativa)
32 descomp <- decompose(ts_obj, type="multiplicative")
33
34 # Graficar la descomposición
35 plot(descomp)
36
37 library(forecast)
38 Acf(ts_obj) # Autocorrelación simple
39 Pacf(ts_obj) # Autocorrelación parcial
40
41 modelo <- auto.arima(ts_obj) # Encuentra automáticamente el mejor modelo ARIMA
42 forecasted <- forecast(modelo, h=365) # Predicción para 1 año (365 días)
43 autoplot(forecasted)
44
18:1 (Top Level) R Script
```

```
actividad 1.R x
Source on Save
Run
Source

38 Acf(ts_obj) # Autocorrelación simple
39 Pacf(ts_obj) # Autocorrelación parcial
40
41 modelo <- auto.arima(ts_obj) # Encuentra automáticamente el mejor modelo ARIMA
42 forecasted <- forecast(modelo, h=365) # Predicción para 1 año (365 días)
43 autoplot(forecasted)
44
45 accuracy(modelo)
46
47 plot(ts_obj, main="Serie Temporal Original", col="blue")
48
49
50 length(ts_obj)
51
52
53 modelo <- auto.arima(ts_obj)
54
55 AIC(modelo)
56
57
58
59
```