



Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey

Actividad 7. Introducción a series de tiempo. Series estacionarias

TC3007C.501 Inteligencia artificial avanzada para la ciencia de datos II

Profesores:

Iván Mauricio Amaya Contreras

Blanca Rosa Ruiz Hernández

Félix Ricardo Botello Urrutia

Edgar Covantes Osuna

Felipe Castillo Rendón

Hugo Terashima Marín

Alumno:

Alberto H Orozco Ramos – A00831719

2 de Noviembre de 2023

Actividad 7: Introducción a series de tiempo.

Series estacionarias

Instrucciones

Problema 1

Trabaja con el set de datos Weekly, que forma parte de la librería ISLR. Este set de datos contiene información sobre el rendimiento porcentual semanal del índice bursátil S&P 500 entre los años 1990 y 2010. Se busca predecir el tendimiento (positivo o negativo) dependiendo del comportamiento previo de diversas variables de la bolsa bursátil S&P 500.

```
library(tidyverse)
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ dplyr      1.1.3      ✓ readr      2.1.4
## ✓ forcats    1.0.0      ✓ stringr    1.5.0
## ✓ ggplot2    3.4.3      ✓ tibble     3.2.1
## ✓ lubridate  1.9.3      ✓ tidyr      1.3.0
## ✓ purrr      1.0.2
## — Conflicts — tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(forecast)
```

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

Usa los datos de las ventas de gasolina en una estación de servicio para analizar modelos de pronósticos de la serie de tiempo:

Semana	1	2	3	4	5	6	7	8	9	10	11	12
Galones de gasolina (miles)	17	21	19	23	18	16	20	18	22	20	15	22

```
# Datos proporcionados
semana <- 1:12
galones <- c(17, 21, 19, 23, 18, 16, 20, 18, 22, 20, 15, 22)

# Creamos un data frame
gasolina_data <- data.frame(semana, galones)

# Imprimimos el resultado
print(gasolina_data)
```

```
##      semana galones
## 1         1      17
## 2         2      21
## 3         3      19
## 4         4      23
## 5         5      18
## 6         6      16
## 7         7      20
## 8         8      18
## 9         9      22
## 10        10      20
## 11        11      15
## 12        12      22
```

- Utiliza los métodos de suavizamiento:
 - Promedios móviles

```
# 1. Promedios Móviles
gasolina_data$promedio_movil <- zoo::rollmean(gasolina_data$galones, k = 3, fill = NA)
```

- Promedios móviles ponderados

```
# 2. Promedios Móviles Ponderados
weights <- c(0.2, 0.3, 0.5) # Adjust weights as needed
gasolina_data$promedio_movil_ponderado <- stats::filter(gasolina_data$galones, weights, method =
"convolution", sides = 2)
```

- Método de suavizamiento exponencial

```
# 3. Método de Suavizamiento Exponencial
alpha_values <- seq(0.1, 0.9, by = 0.1)
```

- Crea un programa que te permita evaluar varios valores de α en el método de suavizamiento exponencial hasta encontrar el valor de α que minimice el CME.

```
best_alpha <- NULL

cme_values <- vector("numeric", length(alpha_values))

for (i in seq_along(alpha_values)) {
  gasolina_data$suavizado_exp <- stats::filter(gasolina_data$galones, alpha_values[i], method =
"recursive", sides = 2)
  cme_values[i] <- mean((gasolina_data$galones - gasolina_data$suavizado_exp)^2, na.rm = TRUE)

  # Verificamos si es el mejor alpha
  if (is.null(best_alpha) || cme_values[i] < min(cme_values)) {
    best_alpha <- alpha_values[i]
  }
}

# Desplegamos el mejor alpha
cat("Best alpha:", best_alpha, "\n")
```

```
## Best alpha: 0.1
```

```
# Imprimimos el resultado
print(gasolina_data)
```

```
##      semana galones promedio_movil promedio_movil_ponderado suavizado_exp
## 1         1      17              NA              NA          17.00000
## 2         2      21              19              18.6          36.30000
## 3         3      19              21              20.8          51.67000
## 4         4      23              20              20.0          69.50300
## 5         5      18              19              20.1          80.55270
## 6         6      16              18              17.8          88.49743
## 7         7      20              18              17.6          99.64769
## 8         8      18              20              19.8         107.68292
## 9         9      22              20              19.6         118.91463
## 10        10      20              19              20.0         127.02316
## 11        11      15              19              18.9         129.32085
## 12        12      22              NA              NA         138.38876
```

- Concluye sobre cuál de los modelos usados es el mejor.

```
# Print the CME values
cat("CME for Promedio Móvil:", mean((gasolina_data$galones - gasolina_data$promedio_movil)^2, n
a.rm = TRUE), "\n")
```

```
## CME for Promedio Móvil: 5.1
```

```
cat("CME for Promedio Móvil Ponderado:", mean((gasolina_data$galones - gasolina_data$promedio_mo
vil_ponderado)^2, na.rm = TRUE), "\n")
```

```
## CME for Promedio Móvil Ponderado: 5.562
```

```
cat("CME for Suavizamiento Exponencial:", cme_values, "\n")
```

```
## CME for Suavizamiento Exponencial: 4.07778 20.20966 57.78426 134.6981 287.1181 593.3874 1236.85 2682.66 6206.787
```

El CME más bajo corresponde a el valor de alfa de 0.1, por ende, el Suavizamiento Exponencial con $\alpha = 0.1$ es tentativamente el mejor ajuste de todos los métodos aplicados

- Predice cuáles son las ventas de gasolina esperadas para la semana 13 con el mejor método que hayas obtenido.

```
# Predecir ventas para la semana 13 utilizando suavizado exponencial
gasolina_data$suavizado_exp <- stats::filter(gasolina_data$galones, best_alpha, method = "recursive", sides = 2)
pred_week_13 <- gasolina_data$suavizado_exp[length(gasolina_data$suavizado_exp)]

cat("Predicción de ventas para la semana 13:", round(pred_week_13, 2), "mil galones\n")
```

```
## Predicción de ventas para la semana 13: 23.72 mil galones
```

Problema 2

Se registró el precio de las acciones de una compañía al cierre de cada día hábil del 24 de agosto al 16 de septiembre. Los data recopilados son:

Día	Precio(\$)
24 de Agosto	81.32
25 de Agosto	81.10
26 de Agosto	80.38
29 de Agosto	81.34
30 de Agosto	80.54
31 de Agosto	80.62
1 de Septiembre	79.54
2 de Septiembre	79.46
6 de Septiembre	81.02
Día	Precio(\$)
7 de Septiembre	80.98
8 de Septiembre	80.80

Día	Precio(\$)
9 de Septiembre	81.44
12 de Septiembre	81.48
13 de Septiembre	80.75
14 de Septiembre	80.48
15 de Septiembre	80.01
16 de Septiembre	80.33

```
# Datos proporcionados
data <- data.frame(
  Día = c("24 de Agosto", "25 de Agosto", "26 de Agosto", "29 de Agosto", "30 de Agosto", "31 de Agosto",
    "1 de Septiembre", "2 de Septiembre", "6 de Septiembre", "7 de Septiembre", "8 de Septiembre",
    "9 de Septiembre", "12 de Septiembre", "13 de Septiembre", "14 de Septiembre", "15 de Septiembre",
    "16 de Septiembre"),
  Precio = c(81.32, 81.10, 80.38, 81.34, 80.54, 80.62, 79.54, 79.46, 81.02, 80.98, 80.80, 81.44,
    81.48, 80.75, 80.48, 80.01, 80.33)
)
```

- a. Use un promedio móvil de tre días para suavizar la serie de tiempo. Pronostique el precio de cierre del 19 de Septiembre del 2005 (que es el día siguiente de operaciones).

```
# Calculamos el promedio móvil de tres días
data$PromedioMovil3 <- zoo::rollmean(data$Precio, k = 3, fill = "extend", align = "right")

# Filtramos el dato correspondiente al 19 de Septiembre
pronostico_a <- tail(data$PromedioMovil3, n = 1)

cat(pronostico_a)
```

```
## 80.27333
```

- b. Emplee el suavizamiento exponencial con $\alpha = 0.6$ como constante de suavizamiento para suavizar la serie de tiempo. Pronostique el precio de cierre del 19 de Septiembre del 2005 (que es el día siguiente de operaciones).

```
# Suavizamiento exponencial con  $\alpha = 0.6$ 
alpha <- 0.6
data$SuavizadoExp <- stats::filter(data$Precio, alpha, method = "recursive", sides = 2)

# Filtramos el dato correspondiente al 19 de Septiembre
pronostico_b <- tail(data$SuavizadoExp, n = 1)

cat(pronostico_b)
```

```
## 201.0208
```

c. ¿Cuál de los 2 métodos prefiere? ¿Por qué? Se prefiere el método suavizado exponencial debido a que el pronóstico es más alto con 201.0208 que el promedio móvil con 80.27333.

Problema 3

Realiza el problema sobre el registro del precio de las acciones (Otros ejemplos de la diapositiva “Series de tiempo estacionarias”).

En el 2005, los ingresos que obtuvo Xerox Corporation por sus productos y servicios de color fueron de \$4.6 mil millones, 30% del total de sus ingresos. En la tabla siguiente se presentan las variaciones porcentuales trimestrales a lo largo de 12 trimestres (*Democrat and Chronicle*, 5 de marzo de 2006).

Año	Trimestre	Crecimiento
2003	1	15
	2	19
	3	15
	4	20
2004	1	26
	2	17
	3	18
	4	21
2005	1	15
	2	17
	3	22
	4	17

```
# Datos proporcionados
data <- data.frame(
  Año = c(2003, 2003, 2003, 2003, 2004, 2004, 2004, 2004, 2005, 2005, 2005, 2005),
  Trimestre = rep(1:4, times = 3),
  Crecimiento = c(15, 19, 15, 20, 26, 17, 18, 21, 15, 17, 22, 17)
)

print(data)
```

```
##      Año Trimestre Crecimiento
## 1  2003          1          15
## 2  2003          2          19
## 3  2003          3          15
## 4  2003          4          20
## 5  2004          1          26
## 6  2004          2          17
## 7  2004          3          18
## 8  2004          4          21
## 9  2005          1          15
## 10 2005          2          17
## 11 2005          3          22
## 12 2005          4          17
```

- Use el suavizamiento exponencial para pronosticar la serie de tiempo. Emplee las constantes de suavizamiento $\alpha = 0.1$, $\alpha = 0.2$, $\alpha = 0.3$.
- ¿Con cuál de estos valores de suavizamiento se obtiene un mejor pronóstico?


```

# Función de suavizamiento exponencial
suavizamiento_exponencial <- function(data, alpha) {
  n <- nrow(data)
  pronosticos <- numeric(n)
  pronosticos[1] <- data$Crecimiento[1] # Inicialización

  # Cálculo de pronósticos
  for (i in 2:n) {
    pronosticos[i] <- alpha * data$Crecimiento[i - 1] + (1 - alpha) * pronosticos[i - 1]
  }

  return(pronosticos)
}

# Pronósticos con diferentes valores de alpha
alpha1 <- 0.1
alpha2 <- 0.2
alpha3 <- 0.3

pronosticos_alpha1 <- suavizamiento_exponencial(data, alpha1)
pronosticos_alpha2 <- suavizamiento_exponencial(data, alpha2)
pronosticos_alpha3 <- suavizamiento_exponencial(data, alpha3)

# Error cuadrático medio (MSE) para evaluar la precisión
mse_alpha1 <- mean((data$Crecimiento - pronosticos_alpha1)^2)
mse_alpha2 <- mean((data$Crecimiento - pronosticos_alpha2)^2)
mse_alpha3 <- mean((data$Crecimiento - pronosticos_alpha3)^2)

# Mostrar Los resultados
cat("MSE para alpha =", alpha1, ":", mse_alpha1, "\n")

```

```
## MSE para alpha = 0.1 : 15.71754
```

```
cat("MSE para alpha =", alpha2, ":", mse_alpha2, "\n")
```

```
## MSE para alpha = 0.2 : 14.25887
```

```
cat("MSE para alpha =", alpha3, ":", mse_alpha3, "\n")
```

```
## MSE para alpha = 0.3 : 14.20919
```

```

# Identificar el mejor pronóstico
mejor_alpha <- which.min(c(mse_alpha1, mse_alpha2, mse_alpha3))
cat("El mejor pronóstico se obtiene con alpha =", c(alpha1, alpha2, alpha3)[mejor_alpha], "\n")

```

```
## El mejor pronóstico se obtiene con alpha = 0.3
```