

A7

A00829752

2023-11-03

Problema 1

Introducción de datos

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

# Definir la función para calcular el Error Cuadrático Medio (CME)
calculate_CME <- function(observado, pronostico) {
  return(mean((observado - pronostico)^2))
}

# Método de promedios móviles
p_ma <- numeric(length(ventas))
e_ma <- numeric(length(ventas))

for (i in 4:length(ventas)) {
  p_ma[i] = mean(ventas[(i-3):(i-1)])
  e_ma[i] = p_ma[i] - ventas[i]
}

# Método de promedios móviles ponderados
p_wma <- numeric(length(ventas))
e_wma <- numeric(length(ventas))

for (i in 4:length(ventas)) {
  p_wma[i] = (1/6) * ventas[i-3] + (2/6) * ventas[i-2] + (3/6) * ventas[i-1]
  e_wma[i] = p_wma[i] - ventas[i]
}

# Método de suavizamiento exponencial
p_ee <- numeric(length(ventas))
e_ee <- numeric(length(ventas))
p_ee[1] <- ventas[1]
p_ee[2] <- ventas[1]
a <- 0.20

for (i in 3:length(ventas)) {
  p_ee[i] = a * ventas[i-1] + (1-a) * p_ee[i-1]
  e_ee[i] = ventas[i] - p_ee[i]
}
```

```

# Encontrar el valor de alfa que minimiza el CME en suavizamiento exponencial
best_alpha <- NULL
min_CME <- Inf
for (alpha in seq(0.01, 0.99, by = 0.01)) {
  p_ee_test <- numeric(length(ventas))
  e_ee_test <- numeric(length(ventas))
  p_ee_test[1] <- ventas[1]
  p_ee_test[2] <- ventas[1]
  for (i in 3:length(ventas)) {
    p_ee_test[i] = alpha * ventas[i-1] + (1-alpha) * p_ee_test[i-1]
    e_ee_test[i] = ventas[i] - p_ee_test[i]
  }
  current_CME <- calculate_CME(ventas[3:length(ventas)], p_ee_test[3:length(ventas)])
  if (current_CME < min_CME) {
    min_CME <- current_CME
    best_alpha <- alpha
  }
}

# Conclusión sobre cuál es el mejor método
best_method <- NULL
if (min(e_ma^2) < min(e_wma^2) & min(e_ma^2) < min(e_ee^2)) {
  best_method <- "Promedios Móviles"
} else if (min(e_wma^2) < min(e_ma^2) & min(e_wma^2) < min(e_ee^2)) {
  best_method <- "Promedios Móviles Ponderados"
} else {
  best_method <- "Suavizamiento Exponencial con alfa"
}

# Pronóstico para la semana 13 con el mejor método
if (best_method == "Promedios Móviles") {
  cat("El mejor método de pronóstico es:", best_method, "\n")
  p_best <- mean(ventas[10:12])
} else if (best_method == "Promedios Móviles Ponderados") {
  cat("El mejor método de pronóstico es:", best_method, "\n")
  p_best <- (1/6) * ventas[10] + (2/6) * ventas[11] + (3/6) * ventas[12]
} else {
  cat("El mejor método de pronóstico es: Suavizamiento Exponencial con alfa=", best_alpha, "\n")
  p_best <- best_alpha * ventas[12] + (1-best_alpha) * p_ee[12]
}

## El mejor método de pronóstico es: Suavizamiento Exponencial con alfa= 0.17
cat("Pronóstico de ventas de gasolina para la semana 13:", p_best, "miles de galones\n")

## Pronóstico de ventas de gasolina para la semana 13: 19.07939 miles de galones

library(ggplot2)
df <- data.frame(Semana = semana[0:length(semana)],
                 Ventas = ventas[0:length(ventas)],
                 Pronostico_PM = c(NA, NA, NA, p_ma[4:length(ventas)]),
                 Pronostico_PMP = c(NA, NA, NA, p_wma[4:length(ventas)]),
                 Pronostico_SE = c(NA, NA, NA, p_ee[4:length(ventas)]))
print(df)

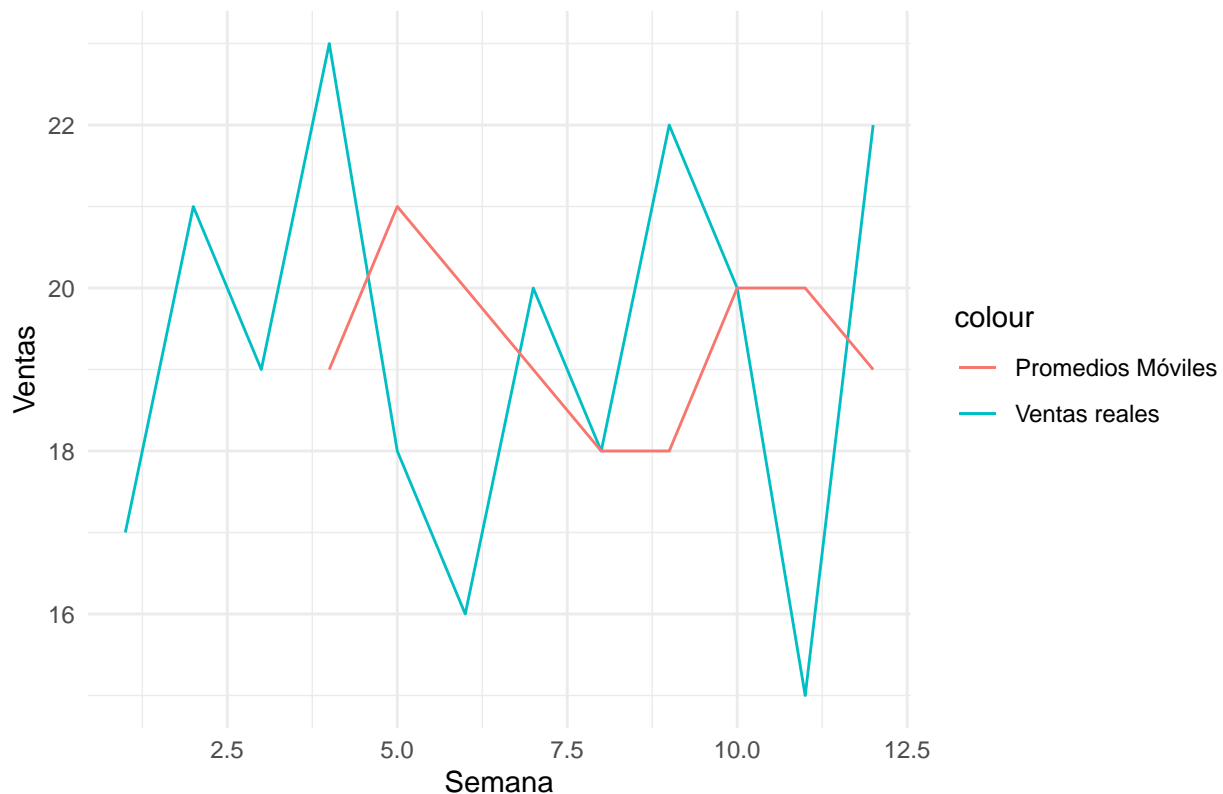
```

```
##   Semana Ventas Pronostico_PM Pronostico_PMP Pronostico_SE
## 1      1     17           NA           NA           NA
## 2      2     21           NA           NA           NA
## 3      3     19           NA           NA           NA
## 4      4     23          19      19.33333      18.04000
## 5      5     18          21      21.33333      19.03200
## 6      6     16          20      19.83333      18.82560
## 7      7     20          19      17.83333      18.26048
## 8      8     18          18      18.33333      18.60838
## 9      9     22          18      18.33333      18.48671
## 10     10     20          20      20.33333      19.18937
## 11     11     15          20      20.33333      19.35149
## 12     12     22          19      17.83333      18.48119
```

```
ggplot(df, aes(x = Semana)) +
  geom_line(aes(y = Ventas, color = "Ventas reales")) +
  geom_line(aes(y = Pronostico_PM, color = "Promedios Móviles")) +
  labs(title = "Pronóstico con Promedios Móviles") +
  theme_minimal()
```

```
## Warning: Removed 3 rows containing missing values (`geom_line()`).
```

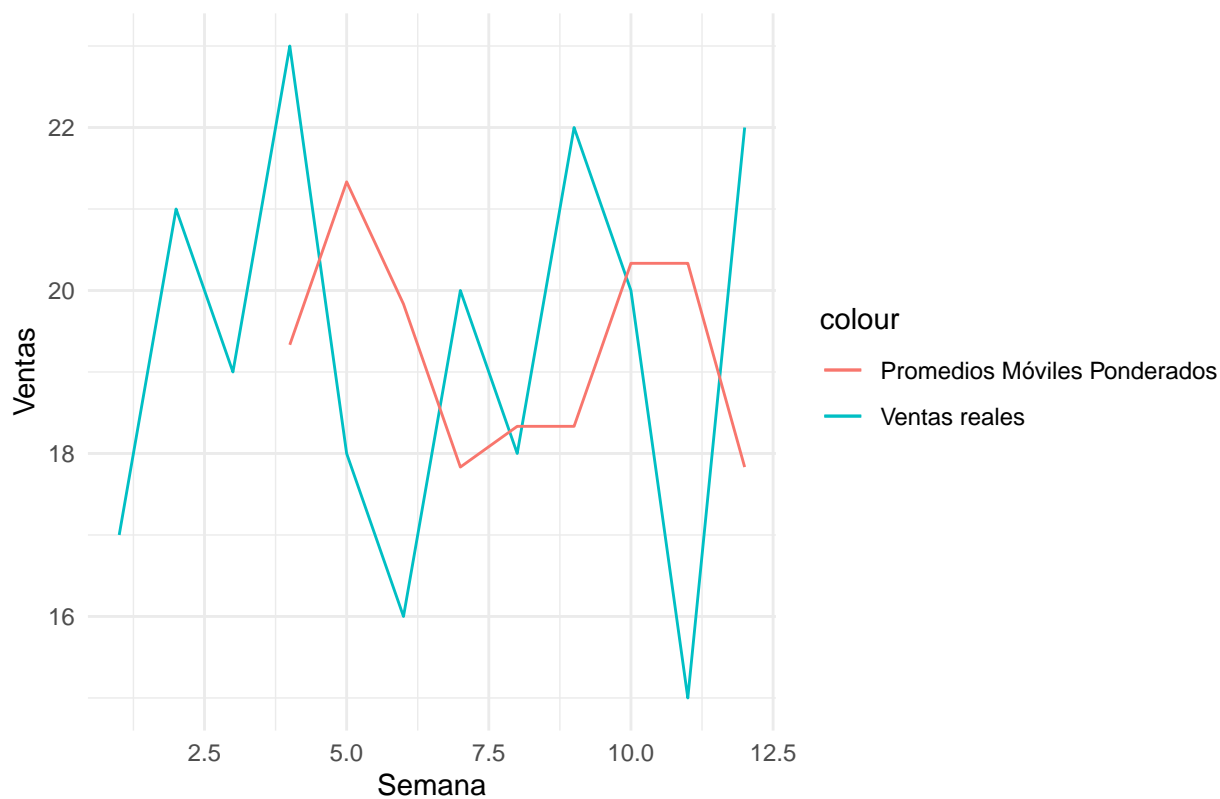
Pronóstico con Promedios Móviles



```
ggplot(df, aes(x = Semana)) +
  geom_line(aes(y = Ventas, color = "Ventas reales")) +
  geom_line(aes(y = Pronostico_PMP, color = "Promedios Móviles Ponderados")) +
  labs(title = "Pronóstico con Promedios Móviles Ponderados") +
  theme_minimal()
```

```
## Warning: Removed 3 rows containing missing values (`geom_line()`).
```

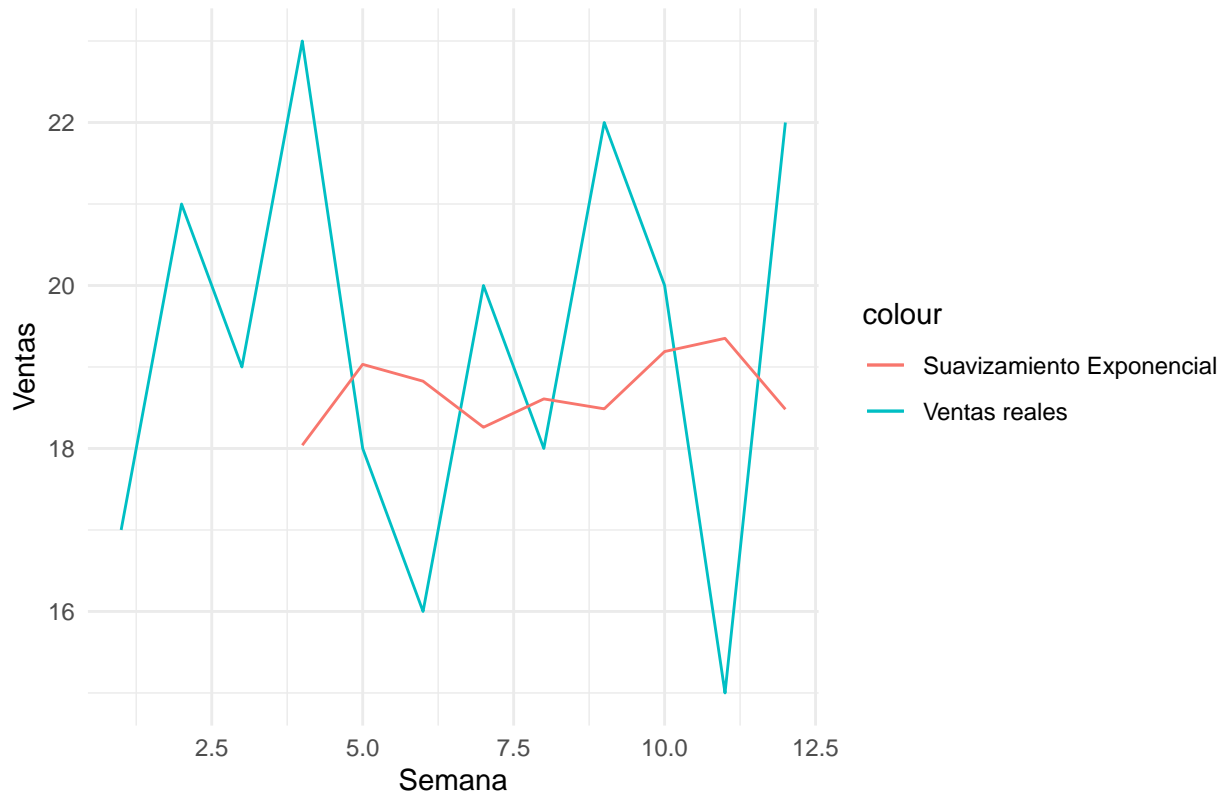
Pronóstico con Promedios Móviles Ponderados



```
ggplot(df, aes(x = Semana)) +  
  geom_line(aes(y = Ventas, color = "Ventas reales")) +  
  geom_line(aes(y = Pronostico_SE, color = "Suavizamiento Exponencial")) +  
  labs(title = "Pronóstico con Suavizamiento Exponencial (alfa=0.20)") +  
  theme_minimal()
```

Warning: Removed 3 rows containing missing values (`geom_line()`).

Pronóstico con Suavizamiento Exponencial (alfa=0.20)



Problema 2

```
precio_acciones <- c(81.32, 81.1, 80.38, 81.34, 80.54, 80.62, 79.54, 79.46, 81.02, 80.98, 80.8, 81.44, 81.32)
```

Promedio móvil

```
promedio_movil_3dias <- numeric(length(precio_acciones))
for (i in 4:length(precio_acciones)) {
  promedio_movil_3dias[i] <- mean(precio_acciones[(i-2):i])
}
print(promedio_movil_3dias)
```

```
## [1] 0.00000 0.00000 0.00000 80.94000 80.75333 80.83333 80.23333 79.87333
## [9] 80.00667 80.48667 80.93333 81.07333 81.24000 81.22333 80.90333 80.41333
## [17] 80.27333
```

Predicción

```
prediccion_promedio_movil <- mean(precio_acciones[(length(precio_acciones)-2):length(precio_acciones)])
cat("Predicción promedio móvil de 3 días:", prediccion_promedio_movil, "\n")
```

```
## Predicción promedio móvil de 3 días: 80.27333
```

Suavizamiento exponencial

```
suavizamiento_exponencial <- numeric(length(precio_acciones))
suavizamiento_exponencial[1] <- precio_acciones[1]
for (i in 2:length(precio_acciones)) {
  suavizamiento_exponencial[i] <- 0.6 * precio_acciones[i-1] + 0.4 * suavizamiento_exponencial[i-1]
}
print(suavizamiento_exponencial)
```

```
## [1] 81.32000 81.32000 81.18800 80.70320 81.08528 80.75811 80.67524 79.99410
## [9] 79.67364 80.48146 80.78058 80.79223 81.18089 81.36036 80.99414 80.68566
## [17] 80.28026
```

Predicción

```
prediccion_suavizamiento_exponencial <- 0.6 * precio_acciones[length(precio_acciones)] + 0.4 * suavizamiento_exponencial[length(precio_acciones)]
cat("Predicción con suavizamiento exponencial (alfa=0.6):", prediccion_suavizamiento_exponencial, "\n")

## Predicción con suavizamiento exponencial (alfa=0.6): 80.31011
```