

Act8-Series de Tiempo

Andrés Villarreal González

2024-11-12

Act 8 Series de Tiempo

Realiza el Análisis de tendencia y estacionalidad

```
ventas = c(4.8, 4.1, 6, 6.5, 5.8, 5.2, 6.8, 7.4, 6, 5.6, 7.5, 7.8, 6.3, 5.9, 8, 8.4)
x= ts(ventas, frequency = 4, start=c(2016,1))
tiempo = 1:16
```

Verifica si es una serie estacionaria

```
library(tseries)

## Warning: package 'tseries' was built under R version 4.3.3

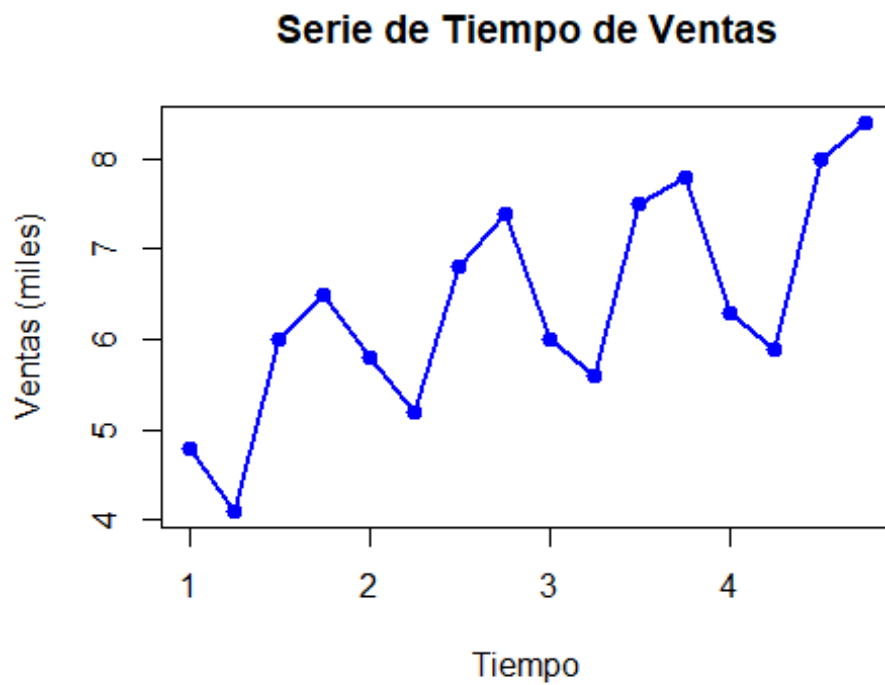
## Registered S3 method overwritten by 'quantmod':
##   method      from
## as.zoo.data.frame zoo

# Prueba de Dickey-Fuller Aumentada
adf_test <- adf.test(x)
print(adf_test)

##
## Augmented Dickey-Fuller Test
##
## data: x
## Dickey-Fuller = -2.7111, Lag order = 2, p-value = 0.3015
## alternative hypothesis: stationary
```

Grafica la serie para verificar estacionalidad

```
# Graficar la serie de tiempo
plot(x, main="Serie de Tiempo de Ventas", xlab="Tiempo", ylab="Ventas (miles)", col="blue", type="o", lwd=2, pch=19)
```

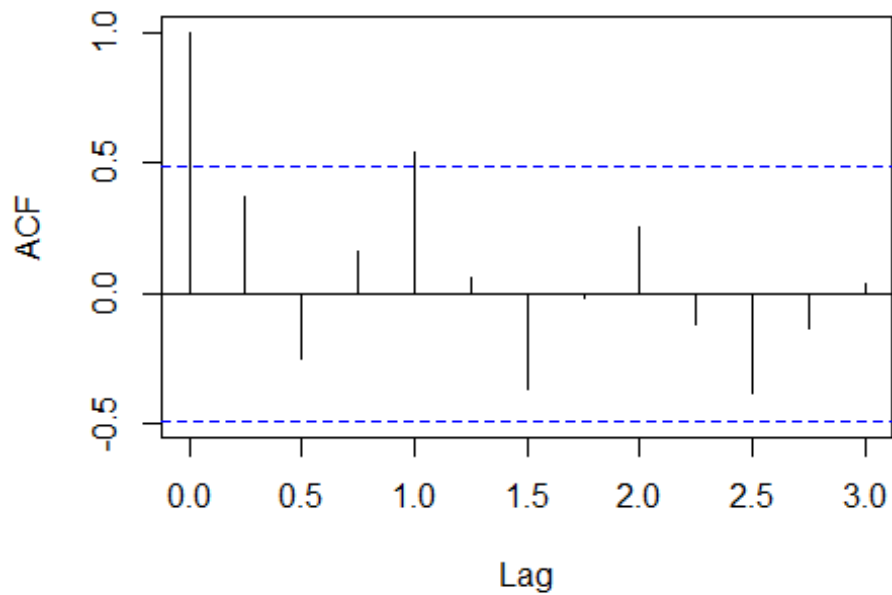


La serie no es estacionaria debido a la presencia de una tendencia creciente en la serie de tiempo original

Grafico de autocorrelacion

```
# Graficar la función de autocorrelación  
acf(x, main="Función de Autocorrelación (ACF)")
```

Función de Autocorrelación (ACF)



```
qnorm(1-0.05/2)/sqrt(length(ventas))
```

```
## [1] 0.489991
```

La serie muestra autocorrelación significativa en los primeros lags, lo cual podría indicar que existe una dependencia temporal a corto plazo.

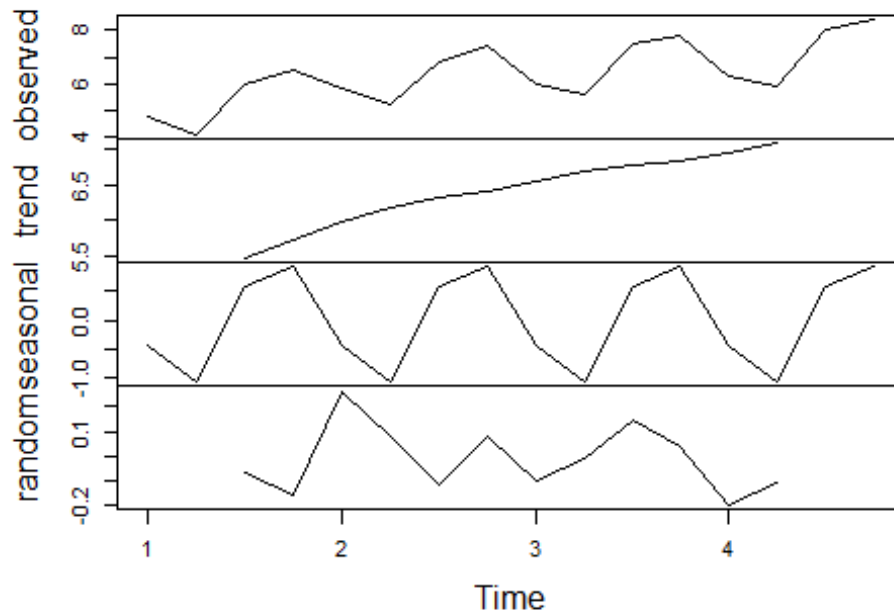
Identifica si el modelo puede ser aditivo o multiplicativo

```
# Descomposición aditiva
```

```
decompose_add <- decompose(x, type = "additive")
```

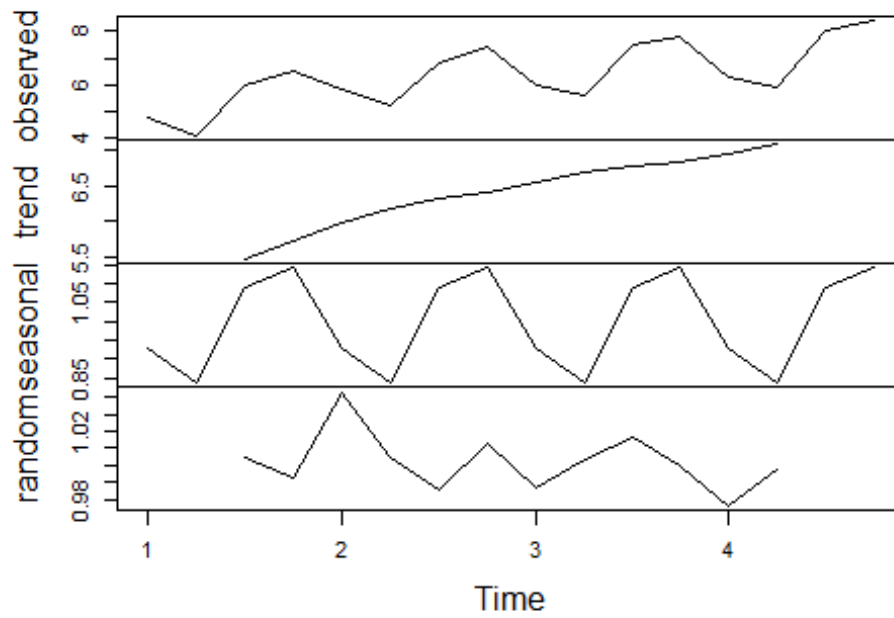
```
plot(decompose_add)
```

Decomposition of additive time series



```
# Descomposición multiplicativa  
decompose_mult <- decompose(x, type = "multiplicative")  
plot(decompose_mult)
```

Decomposition of multiplicative time series



Multiplicativa

Observada: La serie original muestra fluctuaciones con un patrón creciente y cierta estacionalidad.

Tendencia: La componente de tendencia muestra un crecimiento suave y continuo.

Estacionalidad: Se observa un patrón estacional que se repite de forma periódica, mostrando valores altos y bajos en intervalos regulares.

Componente Aleatorio: La variabilidad de este componente es menor, lo que indica que las irregularidades de la serie tienen un impacto más limitado en el comportamiento general de la serie.

Aditiva

Observada: Similar a la serie multiplicativa, muestra una tendencia creciente y un patrón estacional claro.

Tendencia: El crecimiento de la tendencia es idéntico al del modelo multiplicativo, lo que sugiere un aumento constante en los valores de la serie.

Estacionalidad: También se observa un patrón estacional repetitivo, pero aquí la amplitud es constante a lo largo del tiempo, característica del modelo aditivo.

Componente Aleatorio: El componente aleatorio presenta variabilidad con valores tanto positivos como negativos, mostrando fluctuaciones menores alrededor de cero.

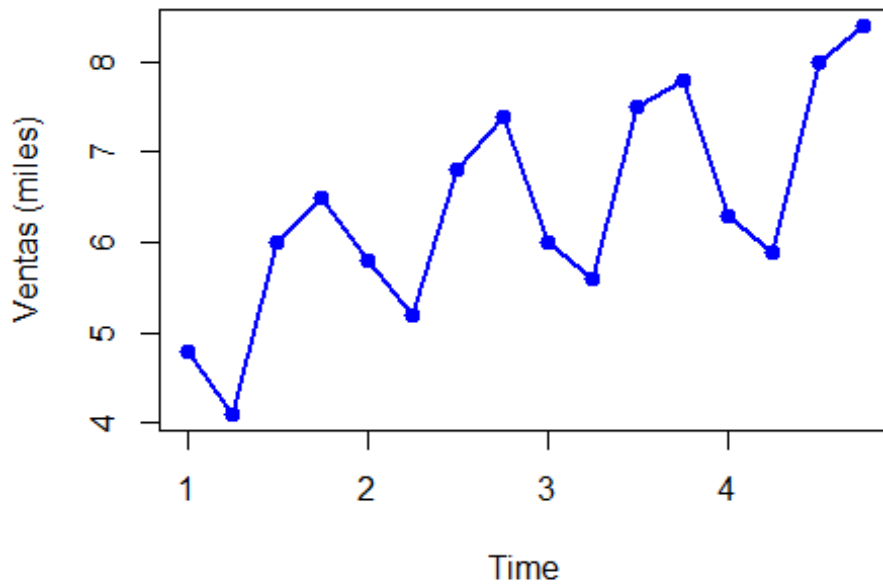
Calcula los índices estacionales y grafica la serie desestacionalizada

```
# Descomponer la serie en componentes estacionales, de tendencia y aleatorias
decomposed_series <- decompose(x, type = "multiplicative") # O puedes probar "additive"

# Extraer la componente estacional y desestacionalizar la serie
seasonal_index <- decomposed_series$seasonal
ventas_desest <- x / seasonal_index

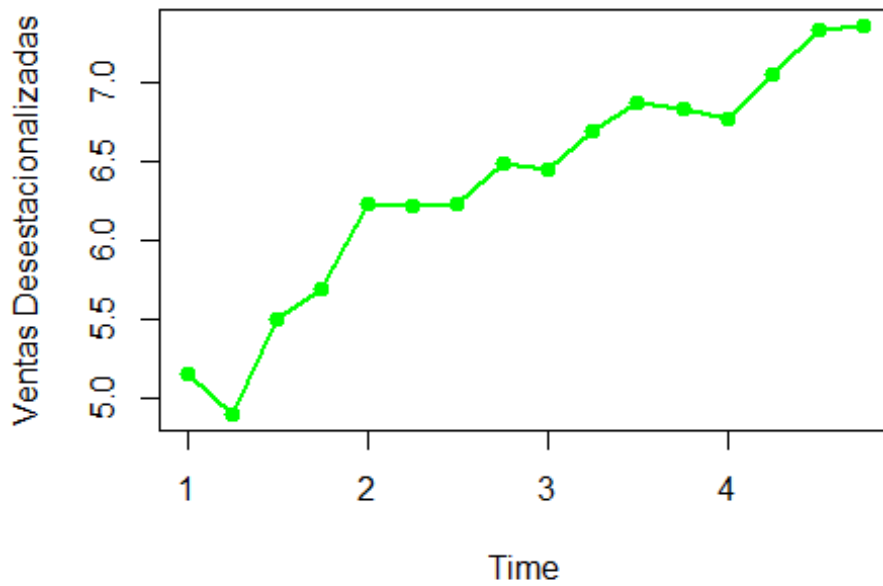
# Graficar la serie original y la desestacionalizada
plot(x, main="Serie de Tiempo Original", ylab="Ventas (miles)",
col="blue", type="o", lwd=2, pch=19)
```

Serie de Tiempo Original



```
plot(ventas_desest, main="Serie de Tiempo Desestacionalizada",  
ylab="Ventas Desestacionalizadas", col="green", type="o", lwd=2, pch=19)
```

Serie de Tiempo Desestacionalizada



La desestacionalización es útil en este caso para identificar una tendencia positiva en las ventas.

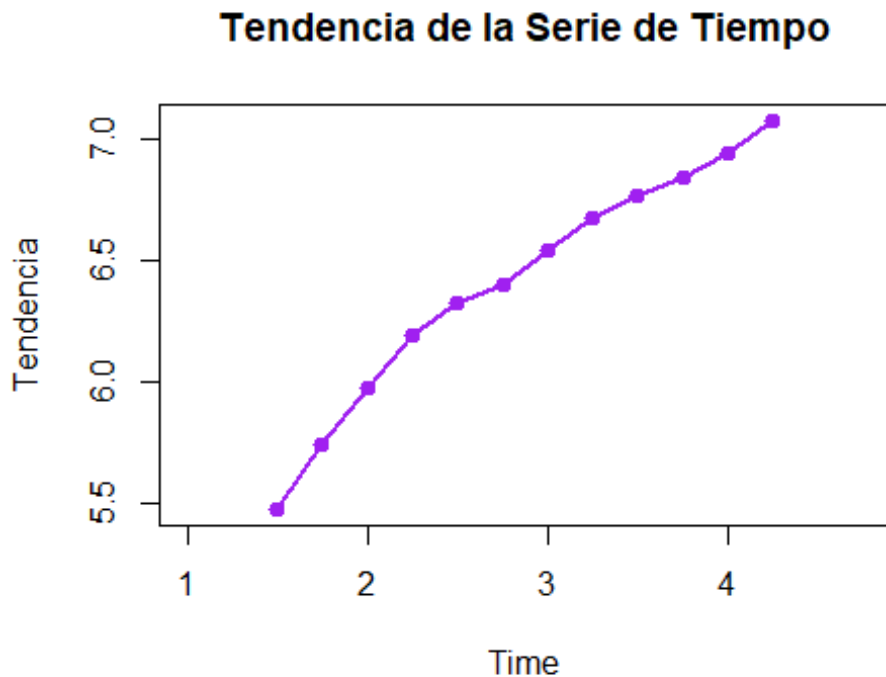
Análisis del modelo lineal de la tendencia

```
# Extraer la tendencia de la descomposición
```

```
trend <- decomposed_series$trend
```

```
# Graficar la tendencia
```

```
plot(trend, main="Tendencia de la Serie de Tiempo", ylab="Tendencia",  
col="purple", type="o", lwd=2, pch=19)
```



Se observa claramente una tendencia creciente

Regresión lineal de la tendencia

```
# Crear un dataframe con el tiempo y la serie desestacionalizada para la  
regresión
```

```
tiempo <- 1:length(ventas_desest)
```

```
df <- data.frame(tiempo = tiempo, ventas_desest =
```

```
as.numeric(ventas_desest))
```

```
# Ajustar el modelo de regresión lineal
```

```
modelo_lineal <- lm(ventas_desest ~ tiempo, data = df)
```

```
# Resumen del modelo para verificar la significancia
```

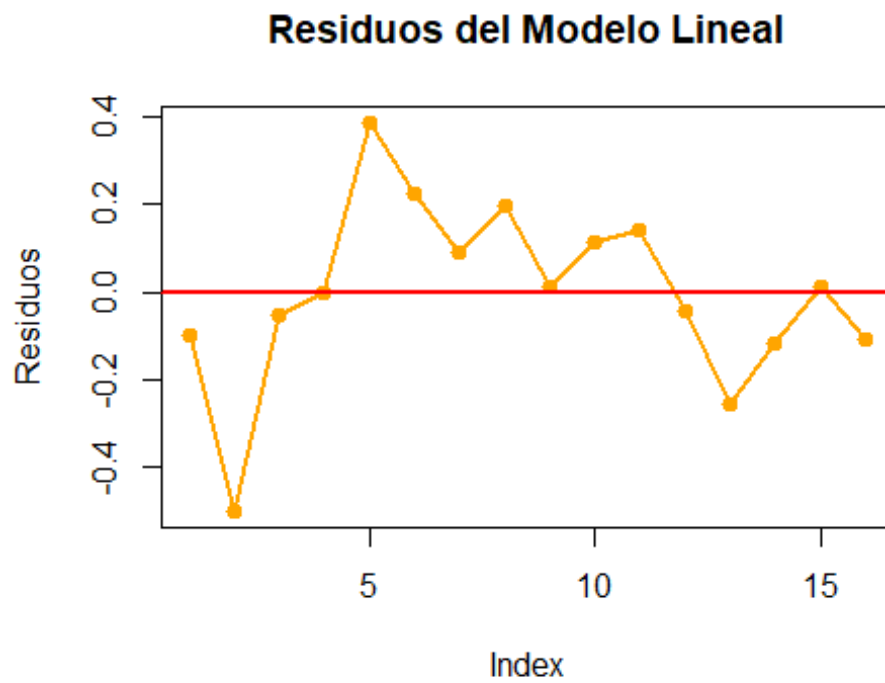
```
summary(modelo_lineal)
```

```
##
## Call:
## lm(formula = ventas_destest ~ tiempo, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.5007 -0.1001  0.0037  0.1207  0.3872
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.10804    0.11171   45.73  < 2e-16 ***
## tiempo      0.14738    0.01155   12.76 4.25e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.213 on 14 degrees of freedom
## Multiple R-squared:  0.9208, Adjusted R-squared:  0.9151
## F-statistic: 162.7 on 1 and 14 DF,  p-value: 4.248e-09
```

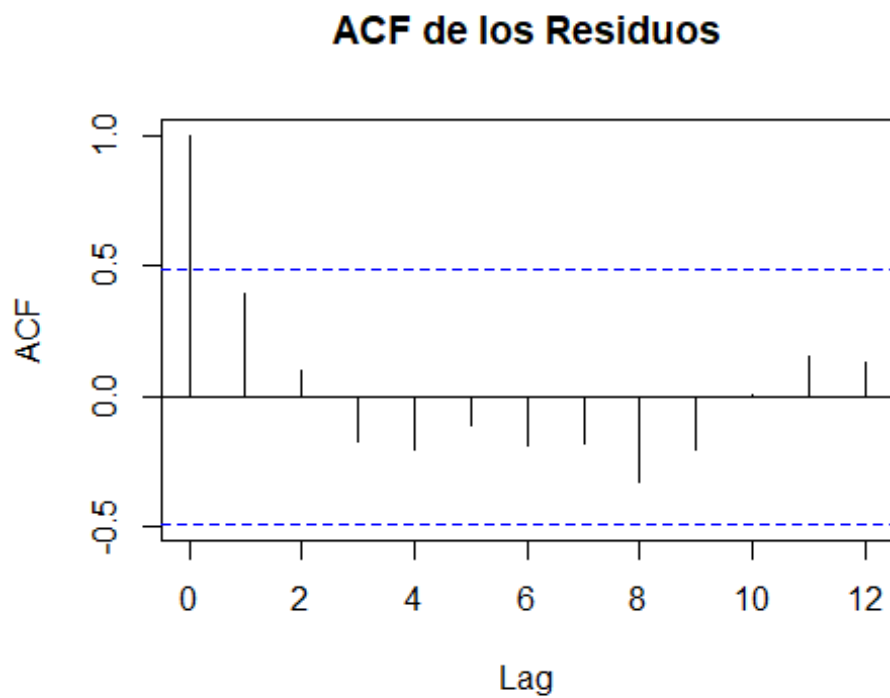
El modelo de regresión lineal entre el tiempo y las ventas desestacionalizadas es significativo y tiene un buen ajuste (con un R^2 alto). Existe una relación positiva significativa entre el tiempo y las ventas desestacionalizadas, lo que sugiere que, a lo largo del tiempo, las ventas desestacionalizadas han tendido a aumentar consistentemente.

Analisis de resisduos

```
# Graficar Los residuos
plot(residuals(modelo_lineal), main="Residuos del Modelo Lineal",
     ylab="Residuos", col="orange", type="o", lwd=2, pch=19)
abline(h = 0, col="red", lwd=2)
```

```
# Graficar ACF de los residuos  
acf(residuals(modelo_lineal), main="ACF de los Residuos")
```



El análisis de los residuos muestra que el modelo lineal ajusta razonablemente bien la serie de tiempo. Aunque hay una leve autocorrelación en el primer lag, no parece haber un patrón de autocorrelación significativo en los residuos en general. Esto sugiere que el modelo captura bien la estructura de la serie de tiempo

Calcular el CME (Cuadrado Medio del Error) y EPAM (Error Promedio Absoluto de la Predicción)

```
# Predicciones del modelo
predicciones <- predict(modelo_lineal, newdata = data.frame(tiempo =
tiempo))

# Calcular el CME
CME <- mean((df$ventas_desest - predicciones)^2)
print(paste("Cuadrado Medio del Error (CME):", CME))

## [1] "Cuadrado Medio del Error (CME): 0.0397064045453044"

# Calcular el EPAM
EPAM <- mean(abs(df$ventas_desest - predicciones))
print(paste("Error Promedio Absoluto de la Predicción (EPAM):", EPAM))

## [1] "Error Promedio Absoluto de la Predicción (EPAM):
0.147060152076123"
```

Ambos errores, el CME y el EPAM, son relativamente bajos, lo que confirma que el modelo lineal está haciendo un buen trabajo al predecir las ventas desestacionalizadas.

Hacer predicciones

```
# Realizar el pronóstico para los siguientes 4 trimestres en la serie
desestacionalizada
nuevo_tiempo <- data.frame(tiempo = (length(ventas_desest) +
1):(length(ventas_desest) + 4))
pronostico_desest <- predict(modelo_lineal, newdata = nuevo_tiempo)

# Recuperar los índices estacionales para el próximo año
indices_estacionales <- decomposed_series$seasonal[1:4] # ya que es
frecuencia trimestral

# "Re-estacionalizar" el pronóstico multiplicando por los índices
estacionales
pronostico_estacionalizado <- pronostico_desest * indices_estacionales

# Crear una serie de tiempo con el pronóstico
pronostico_ts <- ts(c(ventas, pronostico_estacionalizado), frequency = 4,
start = c(2016, 1))

# Graficar los datos originales y el pronóstico
plot(pronostico_ts, main = "Pronóstico de Ventas para el Próximo Año",
```

```

xlab = "Tiempo", ylab = "Ventas (miles)", col = "blue", type = "o", lwd =
2, pch = 19)
lines(ts(ventas, frequency = 4, start = c(2016, 1)), col = "black", lwd =
2, type = "o", pch = 19) # Añadir los datos originales en negro
legend("topleft", legend = c("Datos Originales", "Pronóstico"), col =
c("black", "blue"), lty = 1, pch = 19)

```

Pronóstico de Ventas para el Próximo Año

