

Untitled

2024-11-12

Series de Tiempo

```
library(ggplot2)
library(forecast)

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

library(tseries)
library(lmtest)

## Loading required package: zoo

##
## Attaching package: 'zoo'

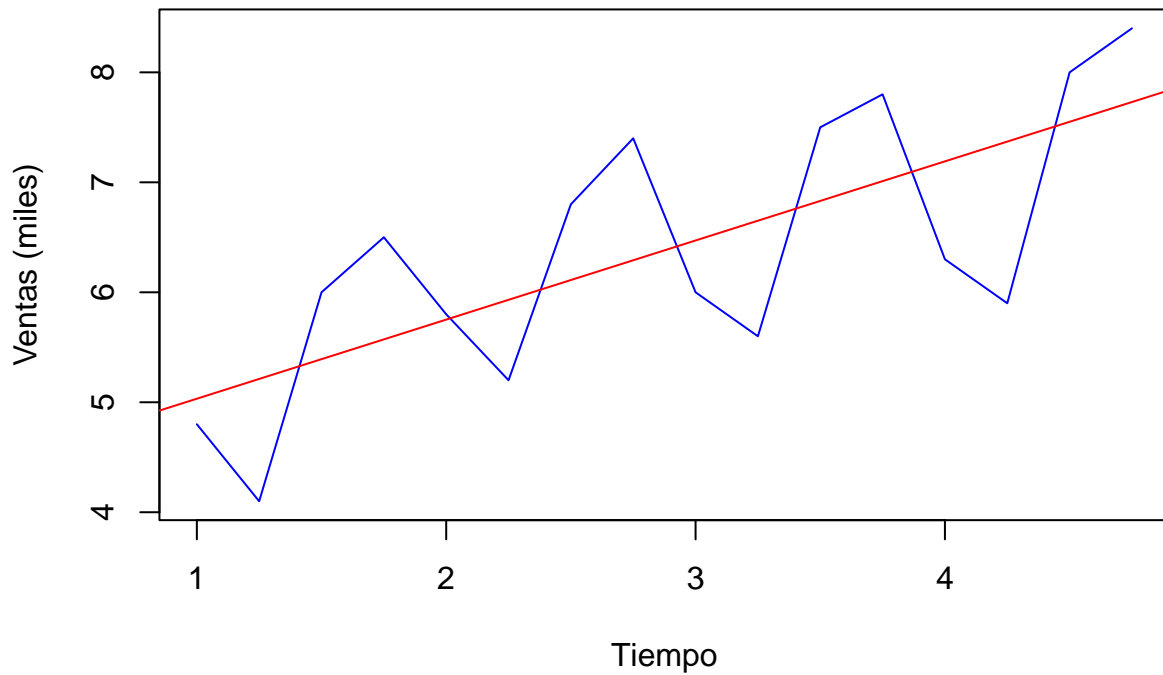
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric

ventas_data <- data.frame(
  Año = rep(1:4, each = 4),
  Trimestre = rep(1:4, times = 4),
  Ventas = c(4.8, 4.1, 6.0, 6.5, 5.8, 5.2, 6.8, 7.4, 6.0, 5.6, 7.5, 7.8, 6.3, 5.9, 8.0, 8.4)
)

ventas_ts <- ts(ventas_data$Ventas, start = c(1, 1), frequency = 4)

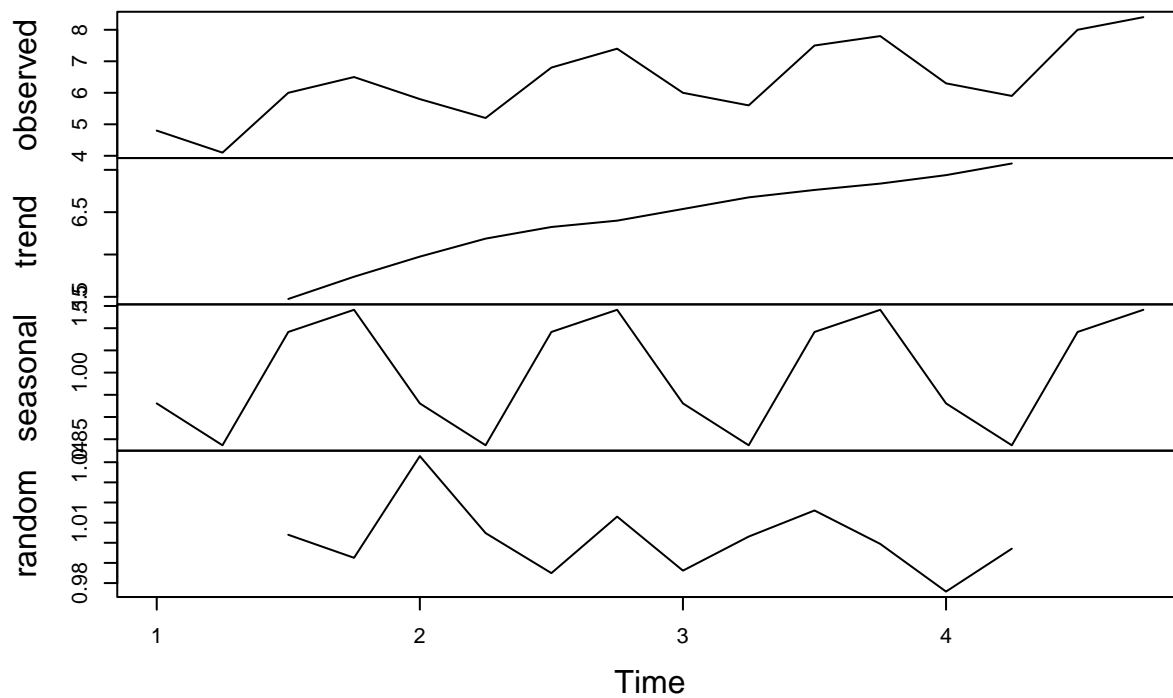
plot(ventas_ts, main = "Ventas Trimestrales", ylab = "Ventas (miles)", xlab = "Tiempo", col = "blue")
abline(reg = lm(ventas_ts ~ time(ventas_ts)), col = "red")
```

Ventas Trimestrales



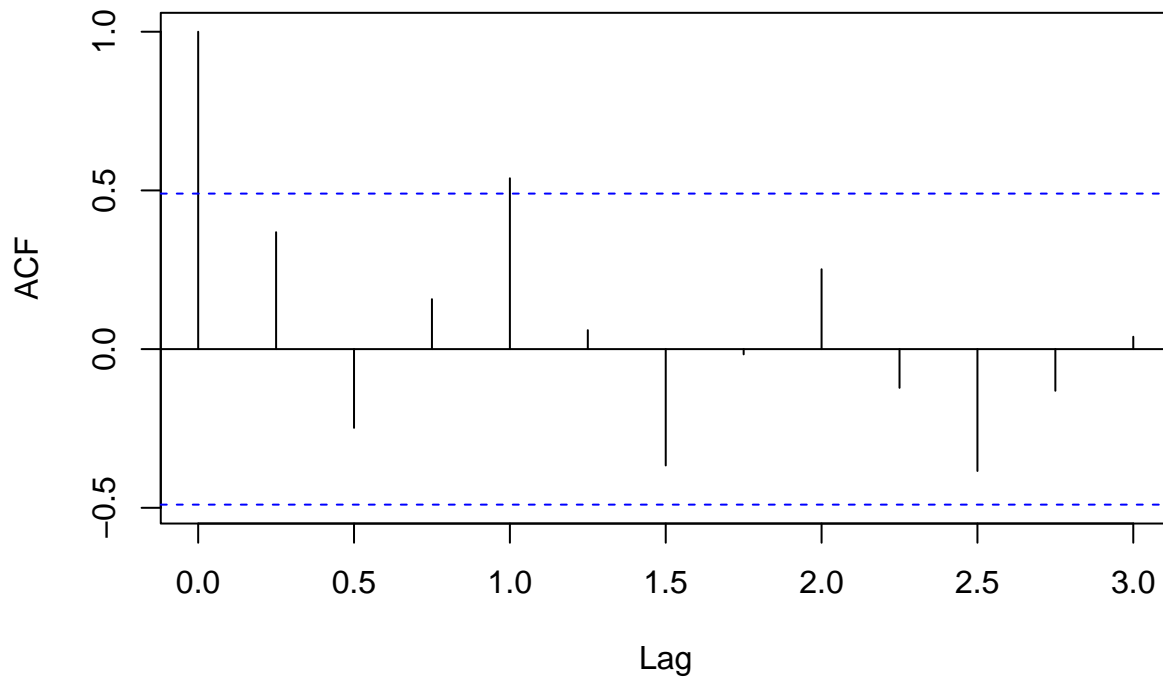
```
descomposicion <- decompose(ventas_ts, type = "multiplicative")
plot(descomposicion)
```

Decomposition of multiplicative time series



```
acf(ventas_ts, main = "Autocorrelación de Ventas Trimestrales")
```

Autocorrelación de Ventas Trimestrales

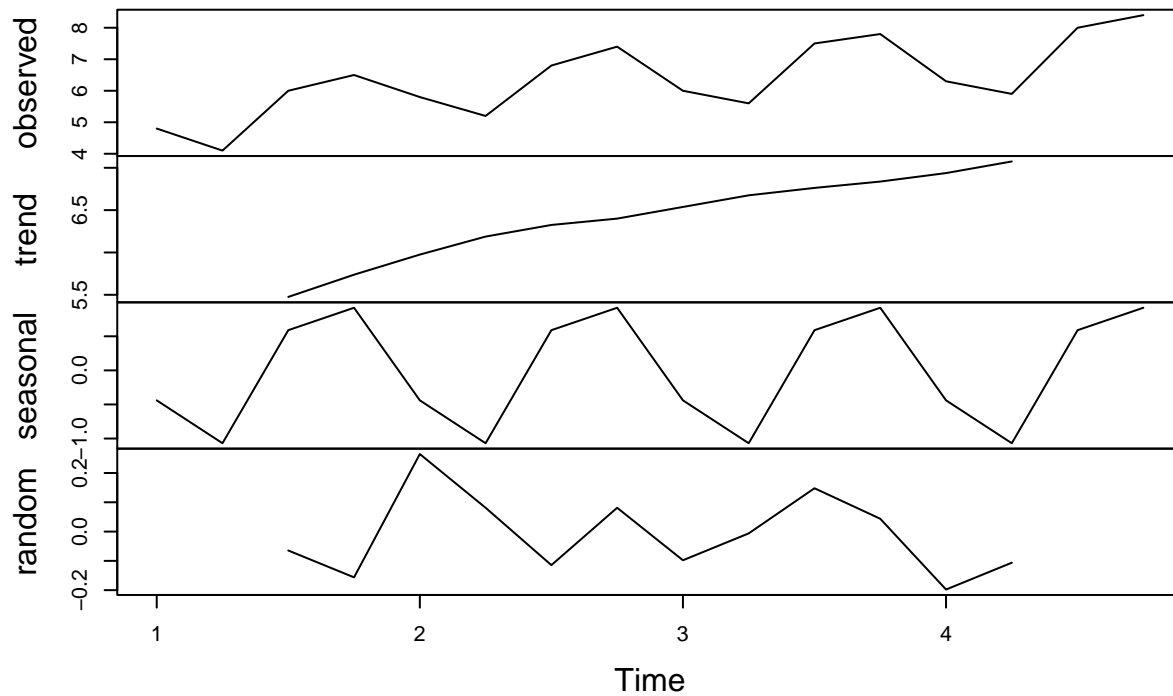


```
adf_test <- adf.test(ventas_ts, alternative = "stationary")  
adf_test$p.value
```

```
## [1] 0.3014787
```

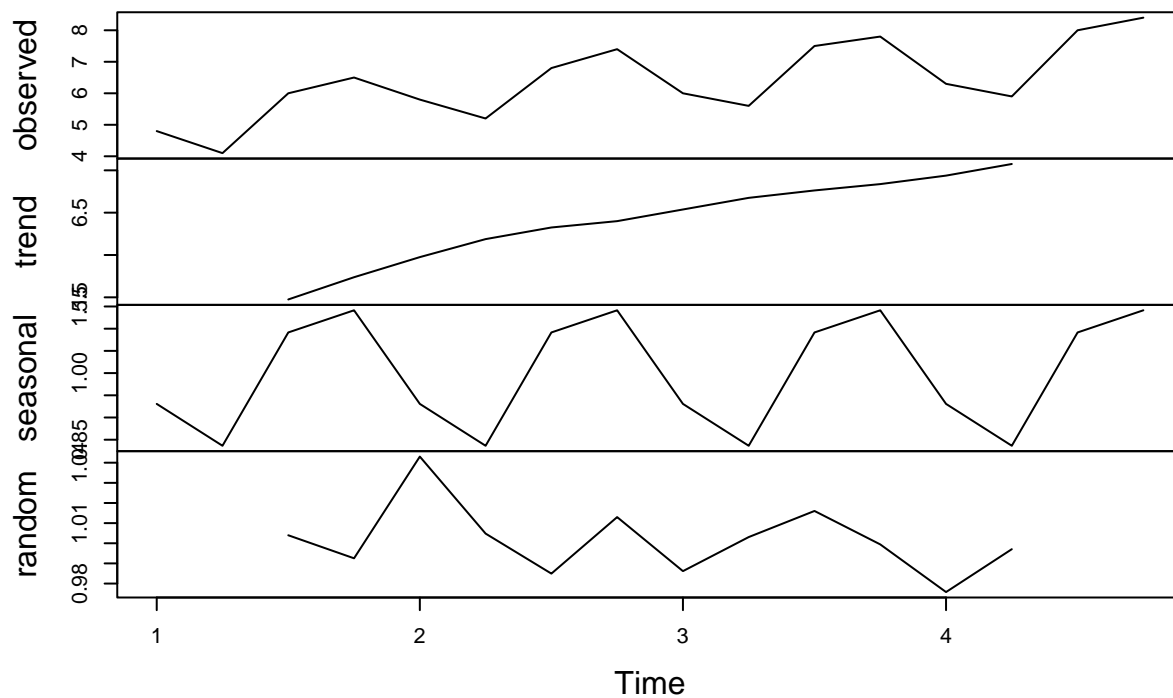
```
descomposicion_aditiva <- decompose(ventas_ts, type = "additive")  
descomposicion_multiplicativa <- decompose(ventas_ts, type = "multiplicative")  
  
par(mfrow = c(2, 1))  
plot(descomposicion_aditiva)
```

Decomposition of additive time series



```
plot(descomposicion_multiplicativa)
```

Decomposition of multiplicative time series



```
indices_estacionales <- descomposicion_multiplicativa$seasonal
print("Índices estacionales:")
```

```
## [1] "Índices estacionales:"
```

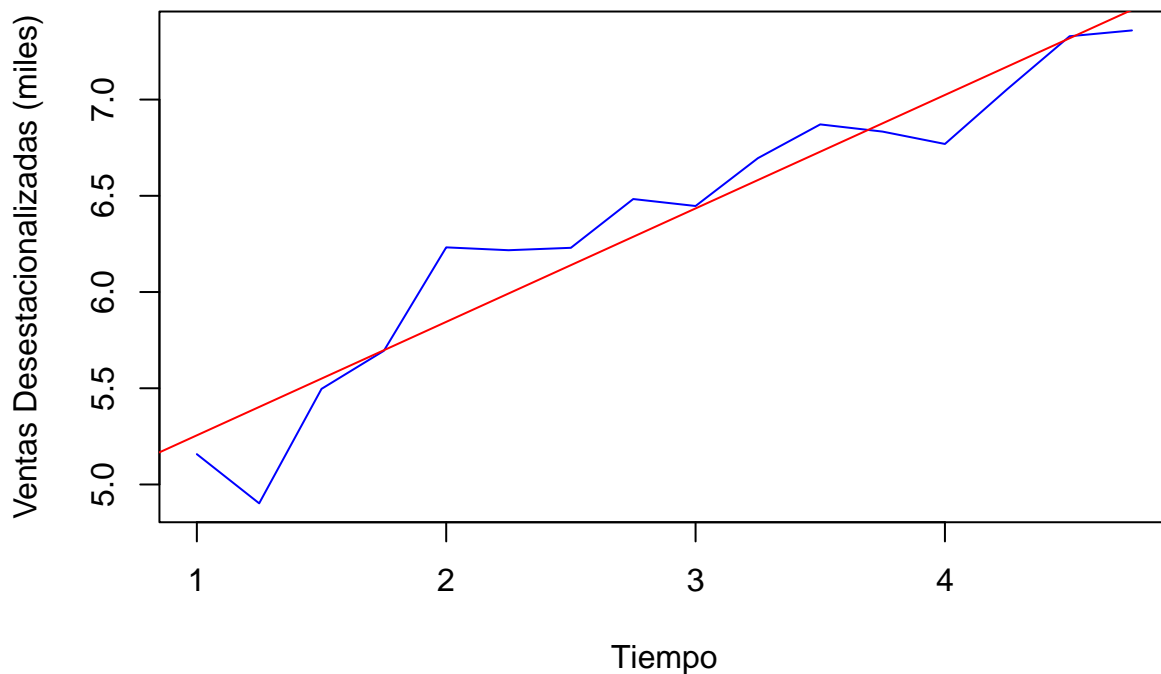
```
print(indices_estacionales)
```

```
##           Qtr1      Qtr2      Qtr3      Qtr4
## 1 0.9306617 0.8363763 1.0915441 1.1414179
## 2 0.9306617 0.8363763 1.0915441 1.1414179
## 3 0.9306617 0.8363763 1.0915441 1.1414179
## 4 0.9306617 0.8363763 1.0915441 1.1414179
```

```
ventas_desestacionalizadas <- ventas_ts / indices_estacionales
```

```
plot(ventas_desestacionalizadas, main = "Serie Desestacionalizada de Ventas Trimestrales",
     ylab = "Ventas Desestacionalizadas (miles)", xlab = "Tiempo", col = "blue")
abline(reg = lm(ventas_desestacionalizadas ~ time(ventas_desestacionalizadas)), col = "red")
```

Serie Desestacionalizada de Ventas Trimestrales



```
tiempo <- 1:length(ventas_desestacionalizadas)
ventas_des_df <- data.frame(Tiempo = tiempo, Ventas_Desestacionalizadas = ventas_desestacionalizadas)

modelo_lineal <- lm(Ventas_Desestacionalizadas ~ Tiempo, data = ventas_des_df)

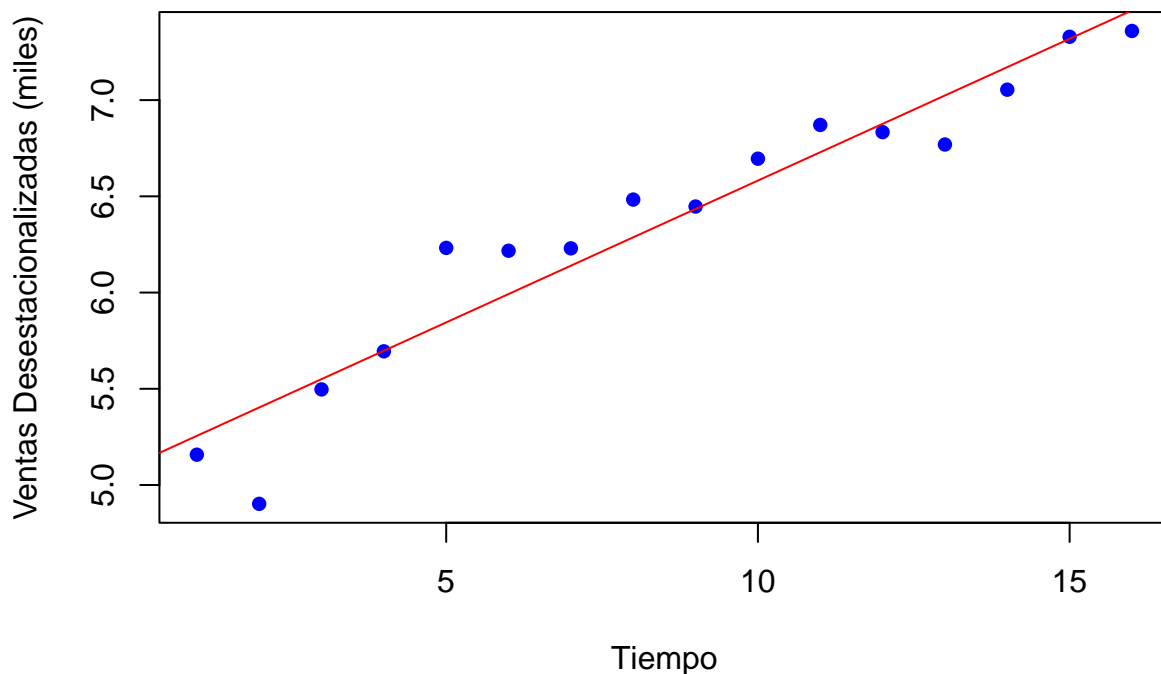
summary(modelo_lineal)
```

```
##
## Call:
## lm(formula = Ventas_Desestacionalizadas ~ Tiempo, data = ventas_des_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

plot(ventas_des_df$Tiempo, ventas_des_df$Ventas_Desestacionalizadas, main = "Ventas Desestacionalizadas",
     xlab = "Tiempo", ylab = "Ventas Desestacionalizadas (miles)", pch = 16, col = "blue")
abline(modelo_lineal, col = "red")
```

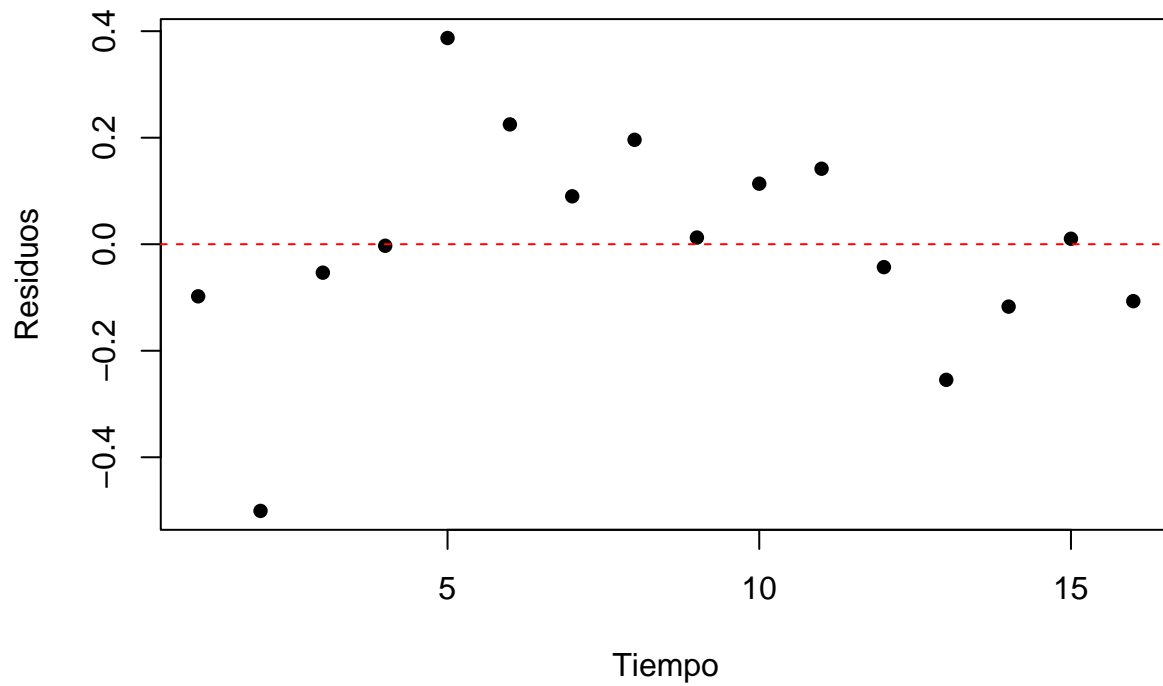
Ventas Desestacionalizadas con Tendencia Lineal



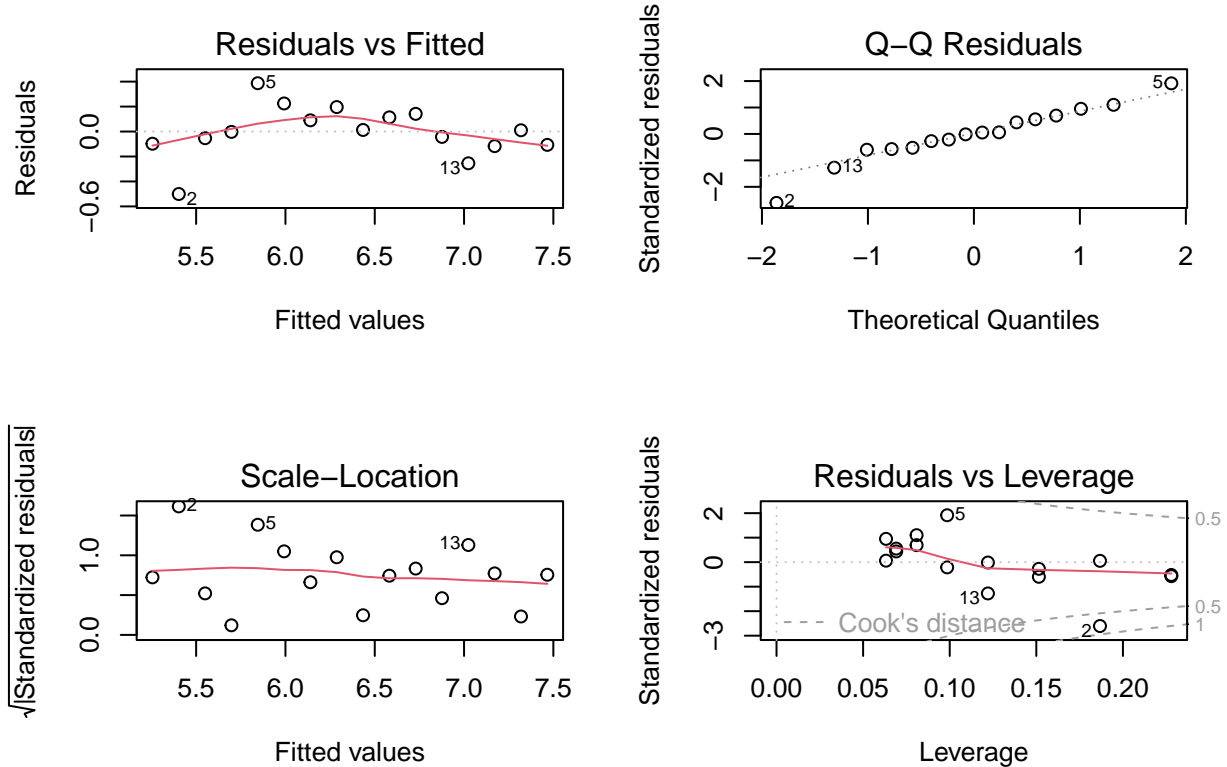
```
residuos <- residuals(modelo_lineal)

plot(tiempo, residuos, main = "Análisis de Residuos", xlab = "Tiempo", ylab = "Residuos", pch = 16)
abline(h = 0, col = "red", lty = 2)
```

Análisis de Residuos



```
par(mfrow = c(2, 2))
plot(modelo_lineal)
```



```
shapiro_test <- shapiro.test(residuos)
shapiro_test$p.value
```

```
## [1] 0.7306976
dw_test <- dwtest(modelo_lineal)
dw_test$p.value

## [1] 0.01752991
predicciones <- predict(modelo_lineal)

cme <- mean((ventas_des_df$Ventas_Desestacionalizadas - predicciones)^2)
print(paste("CME:", cme))

## [1] "CME: 0.0397064045453044"
epam <- mean(abs((ventas_des_df$Ventas_Desestacionalizadas - predicciones) / ventas_des_df$Ventas_Desestacionalizadas))
print(paste("EPAM:", epam))

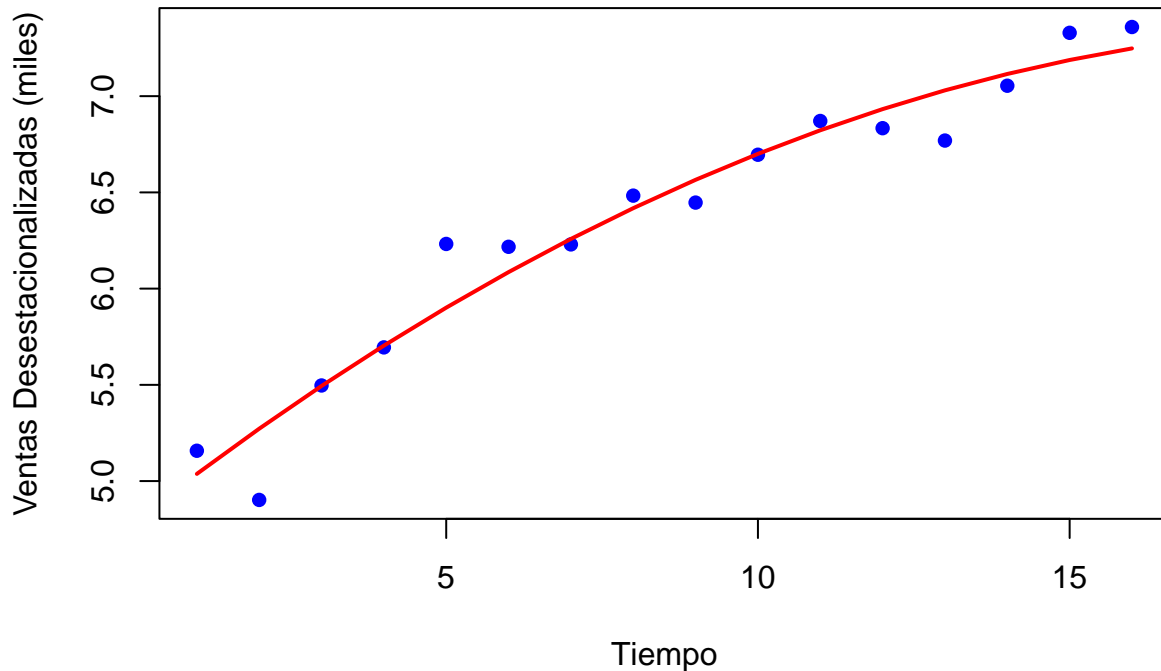
## [1] "EPAM: 0.024395328407222"
ventas_des_df$Tiempo2 <- ventas_des_df$Tiempo^2

modelo_cuadratico <- lm(Ventas_Desestacionalizadas ~ Tiempo + Tiempo2, data = ventas_des_df)

summary(modelo_cuadratico)

##
## Call:
## lm(formula = Ventas_Desestacionalizadas ~ Tiempo + Tiempo2, data = ventas_des_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36986 -0.07058 -0.00100  0.11345  0.33110
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.790283   0.152429  31.426 1.20e-13 ***
## Tiempo       0.253302   0.041269   6.138 3.56e-05 ***
## Tiempo2     -0.006231   0.002360  -2.640  0.0204 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1784 on 13 degrees of freedom
## Multiple R-squared:  0.9484, Adjusted R-squared:  0.9405
## F-statistic: 119.6 on 2 and 13 DF,  p-value: 4.268e-09
plot(ventas_des_df$Tiempo, ventas_des_df$Ventas_Desestacionalizadas, main = "Ventas Desestacionalizadas",
     xlab = "Tiempo", ylab = "Ventas Desestacionalizadas (miles)", pch = 16, col = "blue")
lines(ventas_des_df$Tiempo, predict(modelo_cuadratico), col = "red", lwd = 2)
```


Ventas Desestacionalizadas con Tendencia Cuadrática



```

predicciones_cuadratico <- predict(modelo_cuadratico)

cme_cuadratico <- mean((ventas_des_df$Ventas_Desestacionalizadas - predicciones_cuadratico)^2)
print(paste("CME (Modelo Cuadrático):", cme_cuadratico))

## [1] "CME (Modelo Cuadrático): 0.0258476659392304"

epam_cuadratico <- mean(abs((ventas_des_df$Ventas_Desestacionalizadas - predicciones_cuadratico) / vent
print(paste("EPAM (Modelo Cuadrático):", epam_cuadratico))

## [1] "EPAM (Modelo Cuadrático): 0.0194934923823864"

n_pronostico <- 4

tiempo_futuro <- max(ventas_des_df$Tiempo) + 1:n_pronostico
tiempo_futuro2 <- tiempo_futuro^2
pronostico_df <- data.frame(Tiempo = tiempo_futuro, Tiempo2 = tiempo_futuro2)

pronostico_desestacionalizado <- predict(modelo_cuadratico, newdata = pronostico_df)

indices_estacionales_futuro <- tail(indices_estacionales, n_pronostico)
pronostico_estacionalizado <- pronostico_desestacionalizado * indices_estacionales_futuro

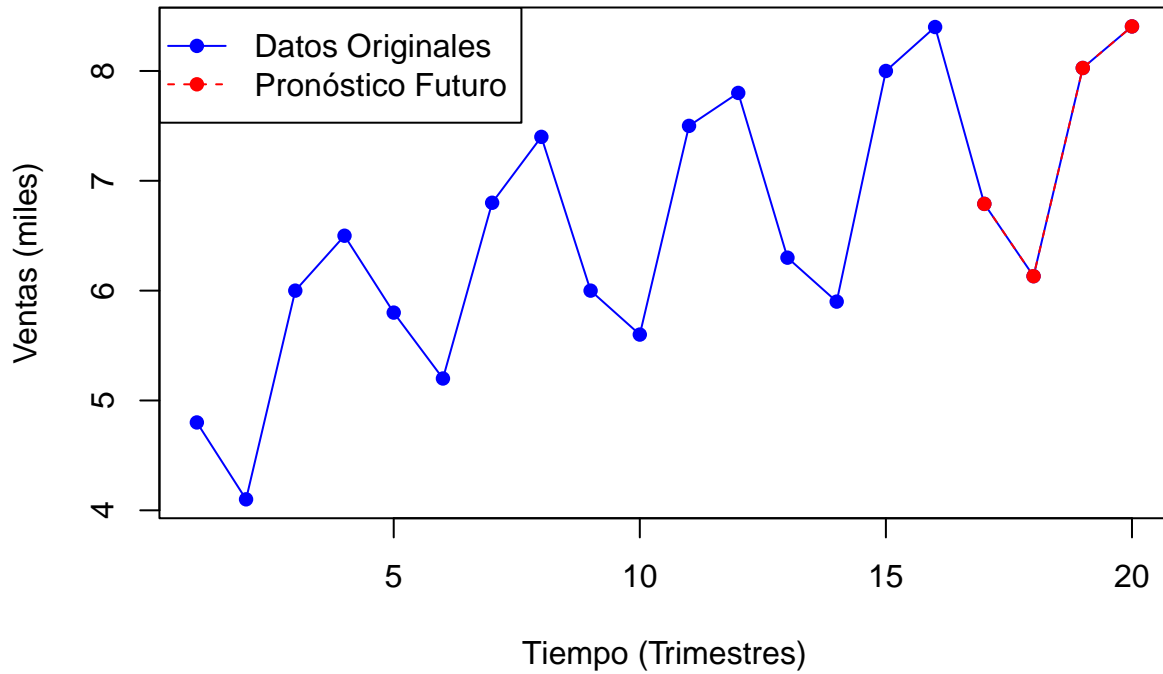
ventas_completo <- c(ventas_ts, pronostico_estacionalizado)
tiempo_completo <- 1:length(ventas_completo)

plot(tiempo_completo, ventas_completo, type = "o", col = "blue", pch = 16,
     main = "Pronóstico de Ventas para el Próximo Año",
     xlab = "Tiempo (Trimestres)", ylab = "Ventas (miles)")
points(tiempo_futuro, pronostico_estacionalizado, col = "red", pch = 16)
lines(tiempo_futuro, pronostico_estacionalizado, col = "red", lty = 2)

```

```
legend("topleft", legend = c("Datos Originales", "Pronóstico Futuro"),
      col = c("blue", "red"), pch = 16, lty = c(1, 2))
```

Pronóstico de Ventas para el Próximo Año



El modelo cuadrático captura mejor la tendencia de la serie desestacionalizada, posiblemente debido a una ligera curvatura en la tendencia de los datos. Esto lo hace el mejor modelo para describir la tendencia de esta serie de tiempo en comparación con el modelo lineal, que solo puede representar una relación lineal simple.