

Asignatura Series de Tiempo

Brayan Hernandez Cardona - Adriana Avila - WILLIAM ROJAS CARABALI

2024-11-04

Contents

1	Acerca de este libro	5
1.1	Introducción	5
2	Pronóstico de Ventas en empresa Retail	7
2.1	Optimizando el Futuro del Retail	7
2.2	Estructura del dataset	7
3	Análisis de Series de Tiempo de Ventas	9
3.1	Limpieza y procesamiento de los datos	9
3.2	Análisis con Promedio Móvil:	10
3.3	Análisis de Rezagos:	11
3.4	Análisis de Estacionalidad (Descomposición):	12
3.5	Análisis de Estacionalidad (Promedio Mensual):	13
3.6	Conclusión	14
4	Preprocesamiento y visualización	15
4.1	Metodología	15
4.2	Análisis Detallado de los Resultados:	16
4.3	Prueba de Estacionariedad (Dickey-Fuller Aumentada - ADF):	17
4.4	Visualización de la Serie Diferenciada:	18
4.5	Funciones de Autocorrelación (ACF) y Autocorrelación Parcial (PACF):	19
4.6	Descomposición de la Serie (Componente Estacional):	21
4.7	Modelo ARIMA (2, 0, 1):	21
4.8	Detección de Puntos de Cambio:	22
4.9	Predicción:	23
4.10	Validación del Modelo (Análisis de Residuos):	25
4.11	Conclusiones:	27

Chapter 1

Acerca de este libro

Este libro recopila los resultados más relevantes obtenidos durante el curso de series de tiempo de la Maestría en Ciencia de Datos, incluyendo análisis, modelos y predicciones.

Los datos para la realización de este book son suministrados por una empresa del sector retail en el ambito de electrónica de la cual se tiene el permiso para su uso durante esta asignatura.

1.1 Introducción

En el actual panorama empresarial, caracterizado por una dinámica competitiva intensa y fluctuaciones del mercado, la capacidad de previsión se erige como un factor determinante para la sostenibilidad y el crecimiento de las organizaciones. La predicción precisa de las ventas reviste una importancia capital para la optimización de recursos, la planificación estratégica y la toma de decisiones en áreas clave como la gestión de inventarios, la producción y el marketing. En este contexto, el análisis de series temporales se posiciona como una metodología cuantitativa robusta con un amplio potencial para identificar patrones, tendencias y comportamientos recurrentes en los datos de ventas, permitiendo la generación de pronósticos confiables y la anticipación a las fluctuaciones de la demanda.

Esta obra se propone profundizar en el marco teórico-práctico del análisis de series temporales aplicado a la predicción de ventas. A lo largo de sus capítulos, se abordarán con rigor los fundamentos matemáticos y estadísticos de esta disciplina, se examinarán las principales técnicas de modelado y se presentarán casos de estudio.

```
#install.packages("bookdown")  
bookdown::render_book()
```

```
bookdown::serve_book()
```

Chapter 2

Pronóstico de Ventas en empresa Retail

2.1 Optimizando el Futuro del Retail

Este proyecto se centra en el análisis y pronóstico de ventas de una empresa en el sector retail, una pequeña cadena de tiendas de electrónica con 3 sucursales ubicadas en diferentes ciudades de Colombia. El conjunto de datos utilizado contiene información detallada de las transacciones realizadas en cada tienda durante los años 2017 y 2020.

2.2 Estructura del dataset

En nuestro dataset se encuentra información de las transacciones de ventas de 4 años comprendidos entre el inicio de 2017 y finales de 2020. En el cual disponemos de la siguiente información:

- **Fecha y hora de la transacción:** Permite analizar patrones de compra a lo largo del tiempo.
- **Nombre de la sucursal:** Permite comparar el rendimiento de las diferentes sucursales.
- **Categoría del producto:** (repuestos, accesorios, electrodomesticos, servicios) Facilita el análisis de la demanda por categorías.
- **ID del producto:** Permite comparar el rendimiento de los diferentes productos.
- **Precio del producto:** Útil para analizar la relación entre precio y volumen de ventas.
- **Cantidad vendida:** Permite calcular las ventas totales y el rendimiento de cada producto.

- **Método de pago:** (e.g., efectivo, tarjeta de crédito, tarjeta débito) Permite identificar tendencias en los métodos de pago.
- **Descuentos y promociones aplicadas:** Permite evaluar la efectividad de las estrategias promocionales.

2.2.1 ¿Por qué es crucial pronosticar las ventas?

En un entorno minorista altamente competitivo, la capacidad de anticipar la demanda futura es fundamental para el éxito de las pequeñas empresas. Un pronóstico con buena asertividad permite:

- **Optimizar la gestión de inventario:** Evitar el exceso de stock y las roturas de stock, minimizando costos de almacenamiento y maximizando la disponibilidad de productos.
- **Planificar las necesidades de personal:** Asignar el número adecuado de empleados en cada tienda y en cada turno, optimizando la atención al cliente y la eficiencia operativa.
- **Maximizar la rentabilidad:** Ajustar las estrategias de precios y promociones en función de la demanda prevista, aumentando las ventas y la rentabilidad.
- **Tomar decisiones estratégicas:** Identificar tendencias de consumo, evaluar la efectividad de las estrategias de marketing, y planificar la expansión a nuevas ubicaciones.

2.2.2 ¿Qué podemos aportar desde la ciencia de datos?

Imaginemos que el análisis de datos nos permite pronosticar las ventas de productos con una asertividad del 80% en la sucursal de Tulúa durante los meses de julio y agosto. Con esta información la empresa puede:

- Definir con cierto grado de seguridad el stock de productos en la sucursal de Tulúa durante esos meses.
- La empresa podrá definir mejores acuerdo de compras de productos con sus proveedores si ya sabe de antemano lo que ocupará para esos meses.
- Se pueden reducir riesgo de desabastecimientos y gastos innecesarios en la recuperación del stock.

En resumen, el pronóstico de ventas proporciona a las empresas una herramienta poderosa para optimizar sus operaciones, mejorar su rentabilidad y fortalecer su posición en el mercado.

Este análisis se desarrollará utilizando el lenguaje de programación R y se documentará en un libro ‘bookdown’ alojado en GitHub. El libro incluirá el código utilizado, las visualizaciones generadas y las conclusiones del análisis.

Chapter 3

Análisis de Series de Tiempo de Ventas

Este capítulo presenta un análisis de una serie de tiempo de ventas, utilizando R y diversas librerías como readxl para leer los datos desde un libro de excel, forecast, dplyr, lubridate y ggplot2. El objetivo es explorar los datos, identificar patrones y tendencias, y sentar las bases para futuros análisis.

```
#install.packages("readxl")
#install.packages("forecast")
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
```

```
library(readxl)
library(dplyr)
library(lubridate)
library(ggplot2)
library(forecast)
```

3.1 Limpieza y procesamiento de los datos

El proceso comienza con la importación y preparación de los datos desde un archivo Excel. Se realizan tareas de limpieza, como la eliminación de caracteres especiales en la columna de costos y la conversión de la fecha de transacción al formato adecuado. Finalmente, se eliminan las filas con valores faltantes en la columna de costos.

Para facilitar el análisis de la estacionalidad, se crea una columna “Mes” a partir de las fechas de transacción.

```
datos <- read_excel("MUESTRA SERIE TIEMPO.xlsx", col_types = c('numeric','numeric','date','numeric'))
```

```

# Agregar una columna de mes
datos <- datos %>% mutate(Mes = month(transaction_date, label = TRUE, abbr = TRUE))

# Limpiar y transformar los datos
datos <- datos %>%
  mutate(cost = as.numeric(gsub("[\\$, ]", "", cost)), # Elimina $, espacios y comas d
    transaction_date = ymd(transaction_date)) %>%
  filter(!is.na(cost)) # Elimina filas con costos NA (Producto 23)

```

3.2 Análisis con Promedio Móvil:

Se crea una serie de tiempo (`ventas_ts`) a partir de las ventas diarias totales. Para suavizar la serie y visualizar tendencias, se calcula el promedio móvil con ventanas de 7 y 30 días. La gráfica resultante “Ventas Diarias Totales y Promedio Móvil” muestra la serie original junto con los promedios móviles. Se observa que el promedio móvil de 30 días suaviza aún más la serie, mostrando con mayor claridad las tendencias a largo plazo y minimizando las fluctuaciones diarias. El promedio móvil de 7 días, por otro lado, permite observar patrones semanales o de corto plazo.

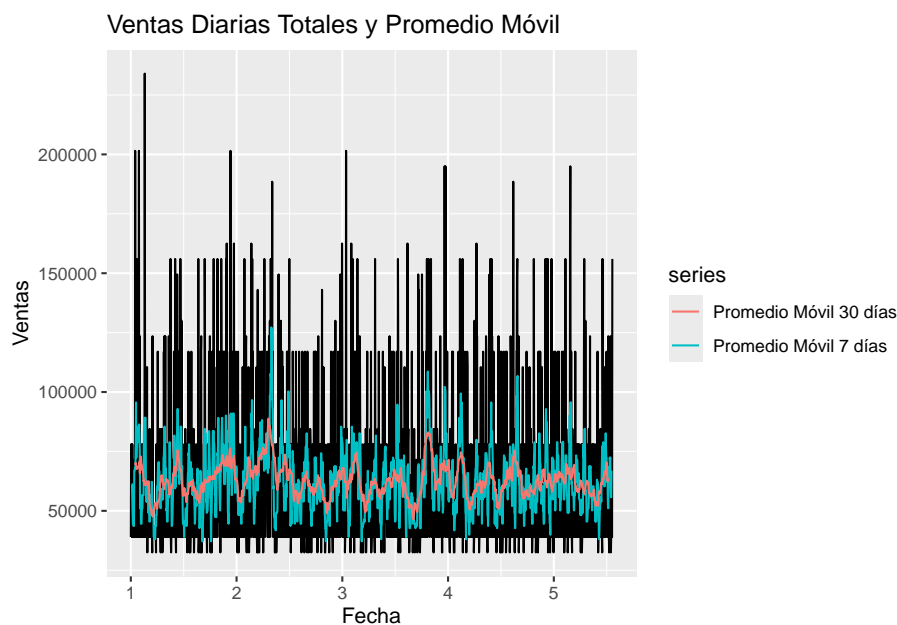
```

# Promedio móvil
# Crear una serie de tiempo para las ventas diarias totales
ventas_ts <- ts(datos %>% group_by(transaction_date) %>% summarise(VentasTotales = sum

# Calcular el promedio móvil (ejemplo con ventana de 7 días y 30 días)
ventas_ma7 <- ma(ventas_ts, order = 7)
ventas_ma30 <- ma(ventas_ts, order = 30)

# Graficar la serie de tiempo y el promedio móvil
autoplot(ventas_ts) +
  autolayer(ventas_ma7, series="Promedio Móvil 7 días") +
  autolayer(ventas_ma30, series = "Promedio Móvil 30 días")+
  ggtitle("Ventas Diarias Totales y Promedio Móvil") +
  xlab("Fecha") + ylab("Ventas")

```



3.3 Análisis de Rezagos:

Se calcula el rezago de las ventas con 1 y 7 días para explorar la autocorrelación en la serie. Estos rezagos se visualizan en la gráfica “Ventas Diarias Totales y Rezagos”, que permite observar la relación entre las ventas de un día y las ventas de uno o siete días anteriores. Esta información es relevante para entender la dependencia temporal de las ventas y puede ser útil en modelos predictivos.

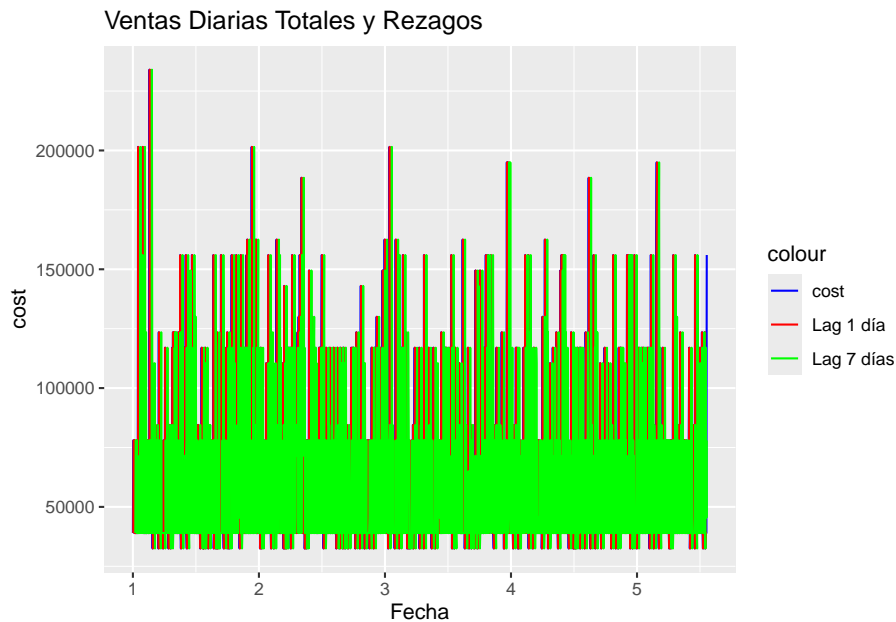
```
# 2. Rezagos (lags)
# Calcular el rezago de las ventas (ejemplo con 1 y 7 días de rezago)
ventas_lag1 <- lag(as.numeric(ventas_ts), 1)

ventas_lag7 <- lag(as.numeric(ventas_ts), 7)

# Crear el data.frame con los vectores numéricos
df_lags <- data.frame(Fecha = time(ventas_ts),
                      cost = ventas_ts,
                      Lag1 = ventas_lag1,
                      Lag7 = ventas_lag7)

# Graficar la serie y los rezagos
ggplot(df_lags, aes(x = Fecha)) +
```

```
geom_line(aes(y = cost, color = "cost")) +
geom_line(aes(y = Lag1, color = "Lag 1 día")) +
geom_line(aes(y = Lag7, color = "Lag 7 días")) +
ggtitle("Ventas Diarias Totales y Rezagos") +
xlab("Fecha") + ylab("cost") +
scale_color_manual(values = c("cost" = "blue", "Lag 1 día" = "red", "Lag 7 días" = "green"))
```



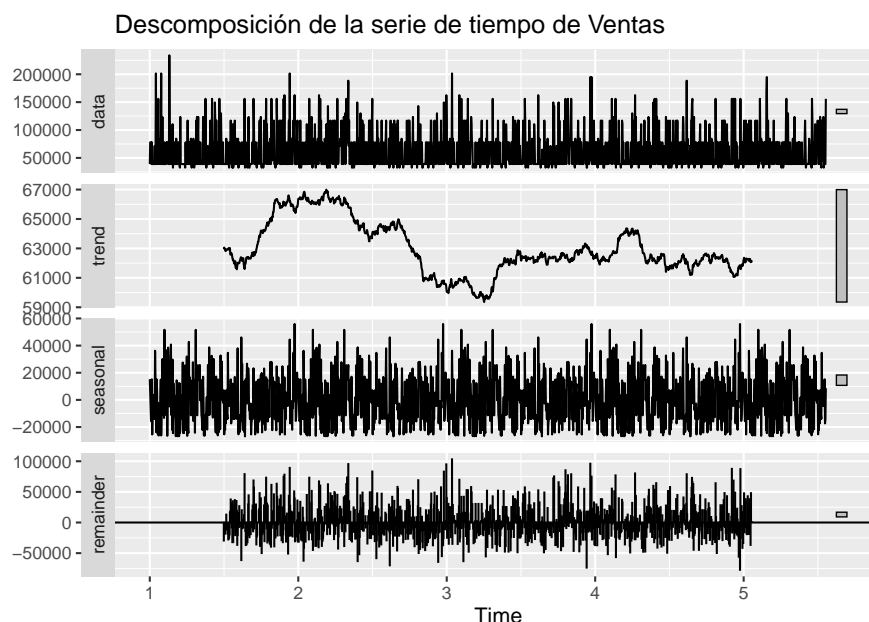
3.4 Análisis de Estacionalidad (Descomposición):

Para analizar la estacionalidad, se utiliza la función `decompose` que descompone la serie de tiempo en sus componentes: tendencia, estacionalidad y residuos. La gráfica resultante “Descomposición de la serie de tiempo de Ventas” muestra cada componente por separado. Se observa la tendencia general de las ventas a lo largo del tiempo, así como un patrón estacional recurrente. El componente de residuos representa la variabilidad que no se explica por la tendencia ni la estacionalidad.

```
# 3. Estacionalidad (usando descomposición)

ventas_decomp <- decompose(ventas_ts)

autoplot(ventas_decomp) + ggtitle("Descomposición de la serie de tiempo de Ventas")
```

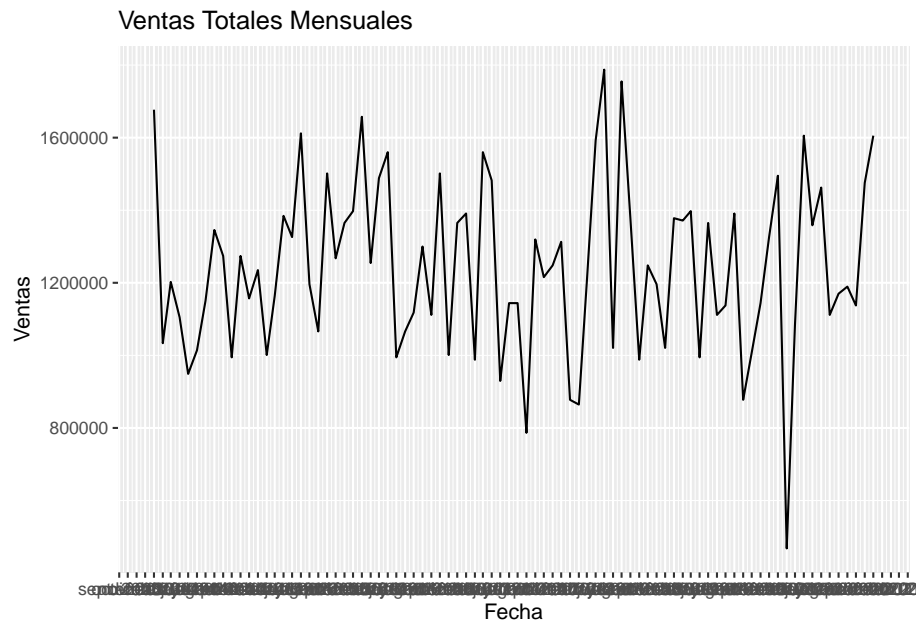


3.5 Análisis de Estacionalidad (Promedio Mensual):

Para una visión más clara de la estacionalidad, se agrega un análisis de las ventas totales mensuales. La gráfica “Ventas Totales Mensuales” muestra las ventas agregadas por mes. Esta visualización permite identificar patrones estacionales a lo largo del año, como picos o valles de ventas en determinados meses. La configuración del eje x con intervalos de un mes y etiquetas que indican el mes y el año facilita la interpretación de la estacionalidad en la serie.

```
# Análisis con promedio mensual para ver la estacionalidad
ventas_mensual <- datos %>%
  group_by(Mes, transaction_date = floor_date(transaction_date, "month")) %>% # Agrupa por mes y
  summarise(VentasTotales = sum(cost))

ggplot(ventas_mensual, aes(x = transaction_date, y = VentasTotales)) +
  geom_line() +
  ggtitle("Ventas Totales Mensuales") +
  xlab("Fecha") + ylab("Ventas")+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %Y") # Ajusta etiquetas del eje x
```



3.6 Conclusión

En resumen, este capítulo realiza un análisis exploratorio de las ventas utilizando diferentes técnicas de series de tiempo. El uso de promedios móviles, rezagos y descomposición permite identificar tendencias, patrones estacionales y autocorrelación en los datos, sentando las bases para un análisis más profundo y la construcción de modelos predictivos en capítulos posteriores.

Chapter 4

Preprocesamiento y visualización

Este informe profundiza en el análisis de las series de tiempo de las ventas de un producto. La comprensión de los patrones históricos de ventas es fundamental para la toma de decisiones estratégicas en las compañías. Un análisis preciso permite optimizar la gestión de inventario, prever la demanda futura, ajustar las estrategias de marketing y, en última instancia, maximizar la rentabilidad.

```
# Instalar paquetes necesarios
if(!require("forecast")) install.packages("forecast")
if(!require("tseries")) install.packages("tseries")
if(!require("ggplot2")) install.packages("ggplot2")
if(!require("zoo")) install.packages("zoo")
if(!require("changepoint")) install.packages("changepoint")
if(!require("tsoutliers")) install.packages("tsoutliers")

# Cargar paquetes
library(forecast)
library(tseries)
library(ggplot2)
library(zoo)
library(changepoint)
library(tsoutliers)
```

4.1 Metodología

Se empleó el lenguaje de programación R, aprovechando sus potentes librerías para el análisis de series de tiempo. Las librerías utilizadas incluyen:

- **forecast:** Para la creación y evaluación de modelos de predicción, incluyendo ARIMA.
- **tseries:** Para realizar pruebas de estacionariedad, como la prueba de Dickey-Fuller Aumentada (ADF).
- **ggplot2:** Para la creación de gráficos de alta calidad y visualmente informativos.
- **zoo:** Para trabajar con series de tiempo irregulares.
- **changepoint:** Para la detección de puntos de cambio en la serie.
- **tsoutliers:** Para la identificación y tratamiento de valores atípicos en series de tiempo.

4.2 Análisis Detallado de los Resultados:

4.2.1 Visualización Inicial y Exploración de Datos:

- La gráfica de la serie de tiempo original revela la evolución de las ventas diarias del producto a lo largo del tiempo (2017-2021). Esta visualización inicial nos permite apreciar la variabilidad de las ventas y la posible presencia de patrones, tendencias o estacionalidad. Se observan periodos con ventas elevadas y periodos con ventas bajas, lo cual justifica un análisis más profundo para comprender las causas de estas fluctuaciones.

```
# Leer los datos
datos <- readxl::read_excel("MUESTRA SERIE TIEMPO.xlsx")

# Crear serie de tiempo

datos["transaction_date"] <- as.Date(datos$transaction_date, origin = "1900-01-01")

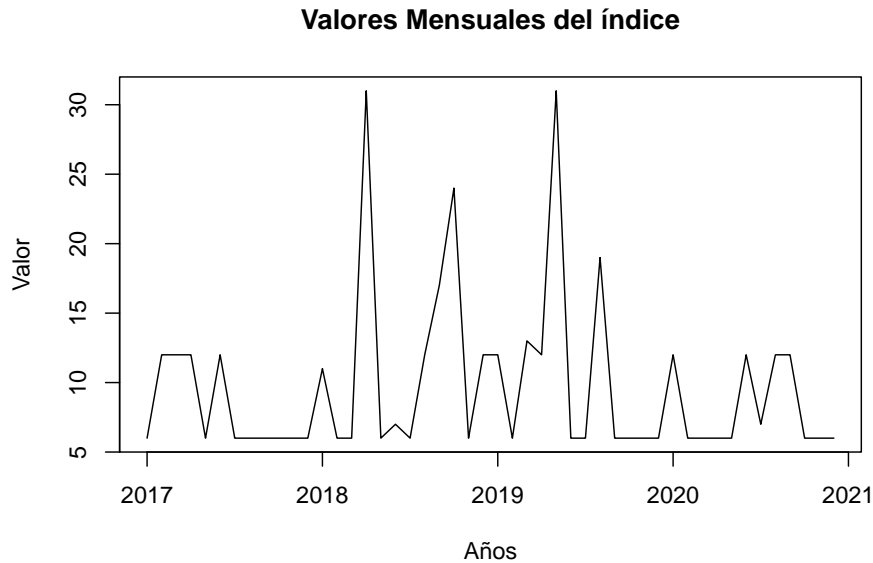
# Filtramos las fechas anteriores o iguales a 2020.
datos <- datos[datos$transaction_date <= as.Date("2020-12-31"), ]

ventas_diarias <- aggregate(quantity ~ transaction_date, data = datos, FUN = sum)

indice.ts <- ts(ventas_diarias[["quantity"]], frequency = 12, start = c(2017, 1), end = 2021)

# Graficar la serie original
plot(indice.ts, main = "Valores Mensuales del índice", xlab = "Años", ylab = "Valor")
```


4.3. PRUEBA DE ESTACIONARIEDAD (DICKY-FULLER AUMENTADA - ADF):17



4.3 Prueba de Estacionariedad (Dickey-Fuller Aumentada - ADF):

- La estacionariedad es un requisito fundamental para muchos modelos de series de tiempo. Una serie estacionaria tiene una media y varianza constantes a lo largo del tiempo. La prueba ADF se utiliza para evaluar la estacionariedad. En este caso, la prueba inicial en la serie original rechazó la hipótesis nula de estacionariedad ($p\text{-valor} > 0.05$).
- Dado que no podemos rechazar la hipótesis de que nuestros datos no son estacionarios, podemos emplear la diferenciación, esta una técnica común para transformar una serie no estacionaria en estacionaria. En este análisis, se aplicó una diferenciación de primer orden, lo que significa que se calculó la diferencia entre las ventas de un mes y las del mes anterior. La prueba ADF aplicada a la serie diferenciada indicó estacionariedad ($p\text{-valor} < 0.05$), cumpliendo así con el requisito para la modelización ARIMA.

```
# Prueba de estacionariedad (Dickey-Fuller)
adf.test(indice.ts)
```

```
##
## Augmented Dickey-Fuller Test
##
## data:  indice.ts
## Dickey-Fuller = -3.046, Lag order = 3, p-value = 0.1564
```

```
## alternative hypothesis: stationary
# Diferenciación

ndiffs(indice.ts) # Para determinar el orden de diferenciación necesario

## [1] 0
dif.indice.ts <- diff(indice.ts, differences = 1) # Diferenciar la serie

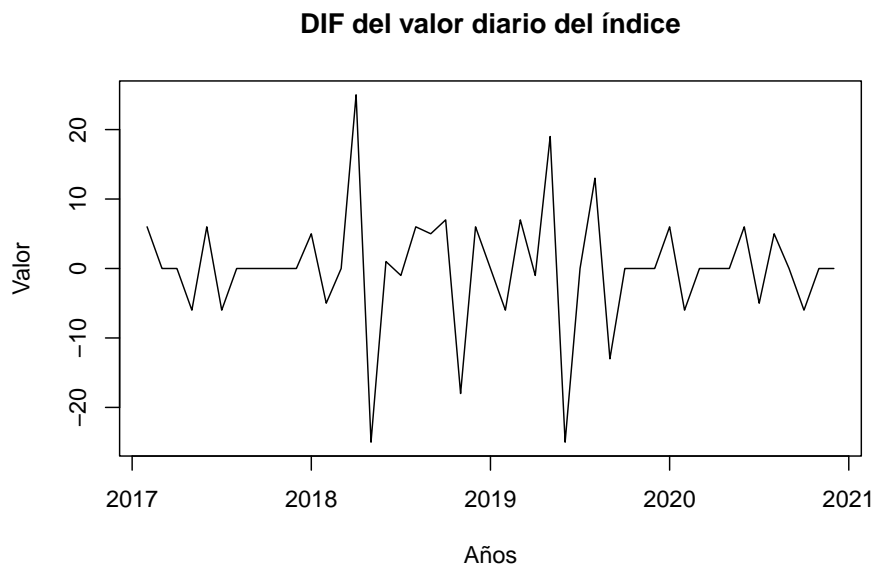
#Volver a realizar la prueba
adf.test(dif.indice.ts)

##
## Augmented Dickey-Fuller Test
##
## data: dif.indice.ts
## Dickey-Fuller = -5.7382, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

4.4 Visualización de la Serie Diferenciada:

La gráfica de la serie diferenciada ilustra las variaciones mensuales en las ventas. Muestra la magnitud y la dirección de los cambios en las ventas de un mes a otro. Esta visualización ayuda a identificar periodos de crecimiento, decrecimiento o volatilidad en las ventas.

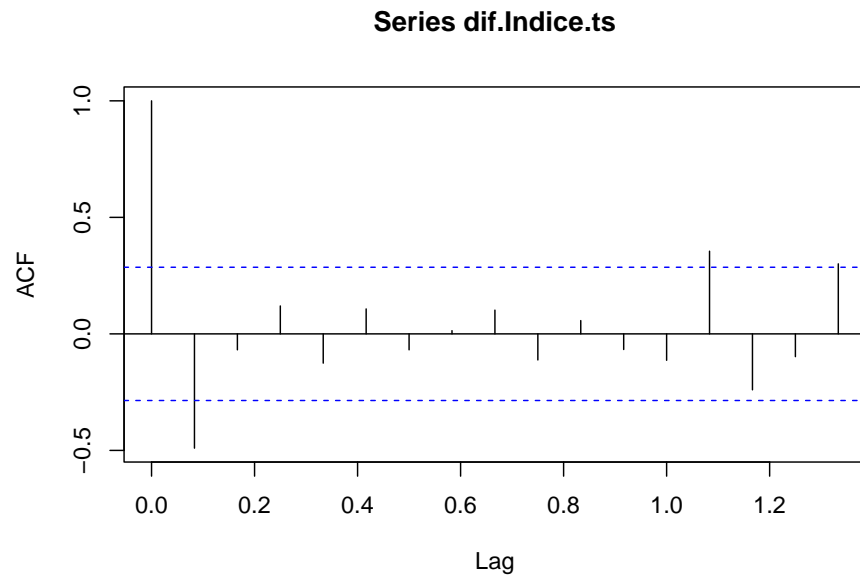
```
# Graficar la serie diferenciada
plot(dif.indice.ts, main = "DIF del valor diario del índice", xlab = "Años", ylab = "DIF del valor diario del índice")
```



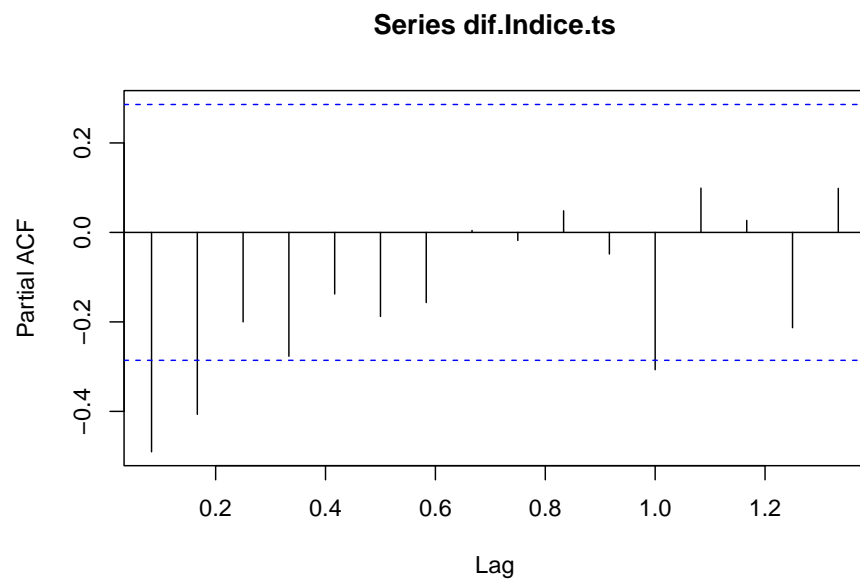
4.5 Funciones de Autocorrelación (ACF) y Autocorrelación Parcial (PACF):

- La ACF mide la correlación entre la serie y sus propios rezagos. La PACF, por otro lado, mide la correlación entre la serie y un rezago específico, eliminando la influencia de los rezagos intermedios.
- El análisis de la ACF y PACF de la serie diferenciada es crucial para determinar el orden del modelo ARIMA (p, d, q). Los rezagos significativos en la ACF y PACF sugieren la inclusión de términos autoregresivos (AR - p) y de media móvil (MA - q) en el modelo. En este caso, la ACF y PACF muestran correlaciones significativas en los primeros rezagos, lo que justifica la elección de un modelo ARIMA con componentes AR y MA.

```
# Funciones ACF y PACF (Se aplica a la diferenciada ya que la original no se puede considerar estacionaria)
acf(dif.indice.ts, main = "Series dif.Indice.ts")
```



```
pacf(dif.indice.ts, main = "Series dif.Indice.ts")
```

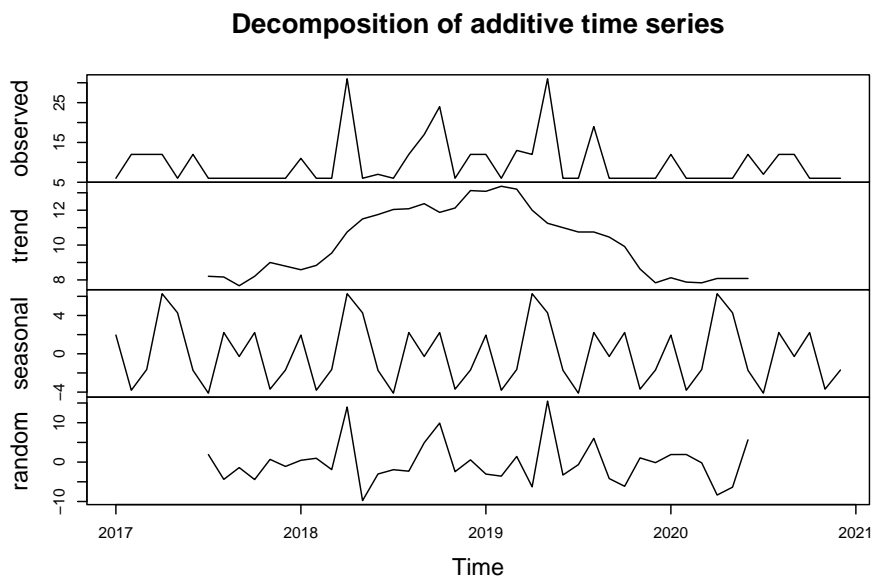


4.6 Descomposición de la Serie (Componente Estacional):

- se utiliza la función `decompose()` para separar la serie en sus componentes: tendencia, estacionalidad y componente aleatoria.

Esto permitiría un análisis más preciso de cada componente y una mejor comprensión de los factores que influyen en las ventas. Se podría, por ejemplo, modelar la componente estacional para realizar predicciones más ajustadas a la realidad.

```
# Descomposición de la serie
decomp <- decompose(indice.ts)
plot(decomp)
```



4.7 Modelo ARIMA (2, 0, 1):

Se empleó la función `auto.arima()` para determinar y ajustar automáticamente el mejor modelo ARIMA a la serie diferenciada. El modelo seleccionado fue un ARIMA(2, 0, 1) con media cero. Desglosemos la notación y los resultados del modelo:

- **(2, 0, 1):** Indica 2 términos autoregresivos (AR), 0 diferencias (la diferenciación ya se realizó previamente) y 1 término de media móvil (MA).
- **Coefficients:**

- **ar1**: -0.0897 (coeficiente del primer término AR). Un valor negativo sugiere una relación inversa entre las ventas del día actual y las ventas del día anterior.
- **ar2**: -0.0965 (coeficiente del segundo término AR). Influencia de las ventas de hace dos días.
- **ma1**: -0.8751 (coeficiente del primer término MA). Un valor negativo indica que los shocks aleatorios del pasado tienen un efecto inverso en las ventas actuales.
- **s.e.**: Error estándar asociado a cada coeficiente. Estos valores nos dan una idea de la precisión de las estimaciones de los coeficientes.

En resumen, el modelo ARIMA(2, 0, 1) captura la dinámica de la serie diferenciada, pero es importante tener en cuenta las medidas de error y la no normalidad de los residuos para una evaluación completa del modelo y la consideración de posibles mejoras.

```
# Modelo ARIMA (automático)
arima_model <- auto.arima(dif.indice.ts) # Usado en la serie diferenciada
summary(arima_model)

## Series: dif.indice.ts
## ARIMA(2,0,1) with zero mean
##
## Coefficients:
##          ar1          ar2          ma1
##      -0.0897   -0.0965   -0.8751
## s.e.    0.1982    0.1858    0.1677
##
## sigma^2 = 40.61:  log likelihood = -153.07
## AIC=314.13   AICc=315.08   BIC=321.53
##
## Training set error measures:
##              ME      RMSE      MAE MPE MAPE      MASE      ACF1
## Training set -0.1026029 6.165966 4.421767 NaN  Inf 0.4194088 -0.01012808
```

4.8 Detección de Puntos de Cambio:

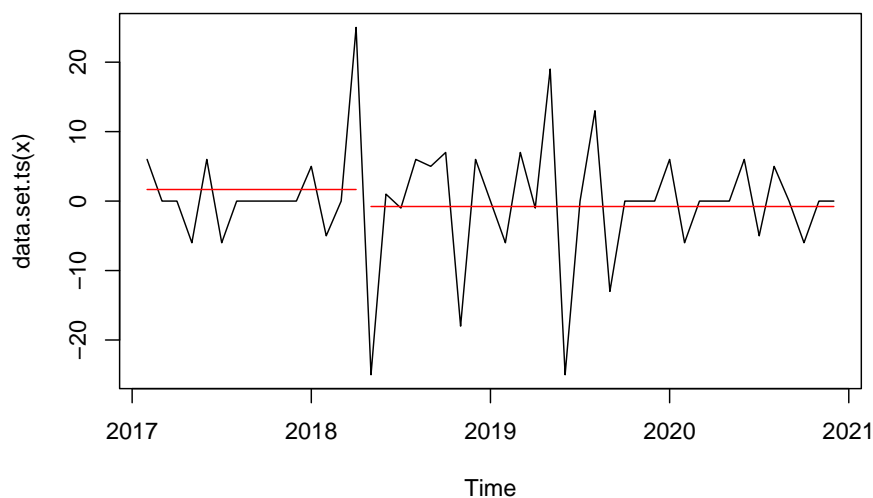
- La función `cpt.mean()` se utiliza para identificar puntos de cambio en la media de la serie. En este caso, se detectó un punto de cambio en el periodo 15 de la serie diferenciada. Esto indica un cambio abrupto en el comportamiento de las ventas en ese momento. Es crucial investigar las posibles causas de este cambio, como cambios en el mercado, promociones especiales, o factores externos, para comprender su impacto en las ventas y ajustar las estrategias comerciales en consecuencia.

```
# Detección de puntos de cambio
cp <- cpt.mean(dif.indice.ts) # Usando la serie diferenciada
```

```
cp
```

```
## Class 'cpt' : Changepoint Object
##      ~~      : S4 class containing 12 slots with names
##      cpttype date version data.set method test.stat pen.type pen.value minseglen cpts
##
## Created on   : Sat Nov  2 03:21:20 2024
##
## summary(.)  :
## -----
## Created Using changepoint version 2.2.4
## Changepoint type      : Change in mean
## Method of analysis    : AMOC
## Test Statistic       : Normal
## Type of penalty       : MBIC with value, 11.55044
## Minimum Segment Length : 1
## Maximum no. of cpts   : 1
## Changepoint Locations : 15
```

```
plot(cp)
```



4.9 Predicción:

Utilizando el modelo ARIMA(2, 0, 1) ajustado, se generaron predicciones para 3 y 6 periodos futuros. La función `forecast()` en R proporciona no solo el valor

predicho (“Point Forecast”), sino también intervalos de confianza que nos ayudan a entender la incertidumbre asociada a estas predicciones.

4.9.1 Predicción a 6 periodos:

La predicción para 6 periodos muestra lo siguiente:

- **Point Forecast:** El valor predicho para las ventas diferenciadas en cada uno de los 6 meses siguientes. Por ejemplo, para enero de 2021, se espera un cambio en las ventas de aproximadamente 2.85 unidades respecto al mes anterior. Es crucial recordar que estas son predicciones de la serie diferenciada, no de las ventas reales.
- **Lo 80 y Hi 80:** Los límites inferior y superior del intervalo de confianza del 80%. Por ejemplo, para enero de 2021, hay un 80% de probabilidad de que el cambio en las ventas esté entre -5.32 y 11.01 unidades.
- **Lo 95 y Hi 95:** Los límites inferior y superior del intervalo de confianza del 95%. Para enero de 2021, hay un 95% de probabilidad de que el cambio en las ventas esté entre -9.65 y 15.34 unidades. Observamos que, como es de esperar, el intervalo de confianza del 95% es más amplio que el del 80%, reflejando una mayor incertidumbre.

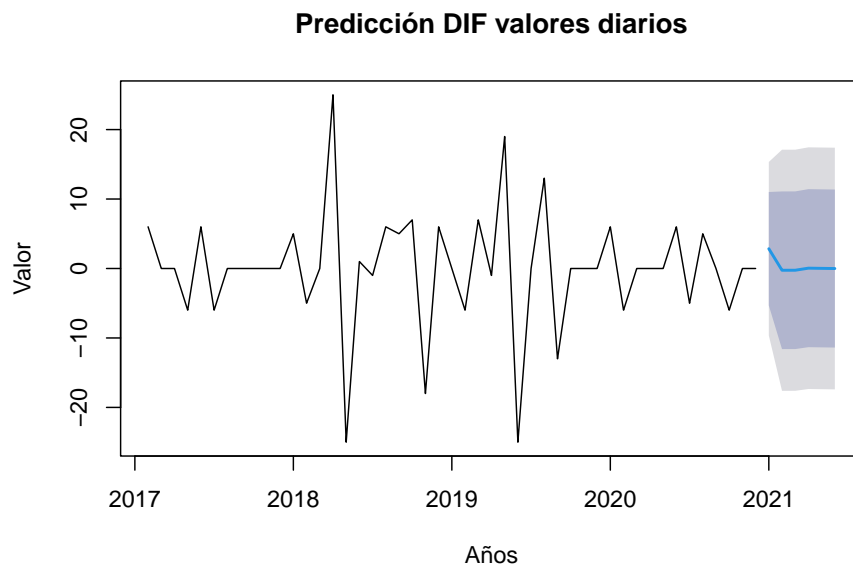
4.9.2 Predicción a 3 periodos:

El `print(prediccion)` para 3 periodos ofrece la misma información, pero solo para los 3 meses siguientes. Es útil comparar las predicciones a 3 y 6 periodos para observar cómo evoluciona la incertidumbre a medida que se proyecta más hacia el futuro. Generalmente, la incertidumbre (y por lo tanto, la amplitud de los intervalos de confianza) aumenta con el horizonte de predicción.

```
# Predicción
```

```
prediccion <- forecast(arima_model, h = 6) # 6 periodos adelante
```

```
plot(prediccion, main = "Predicción DIF valores diarios", xlab = "Años", ylab = "Valor
```

```
print(prediccion)
```

```
##          Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
## Jan 2021      2.84496582    -5.321986    11.01192    -9.645309    15.33524
## Feb 2021     -0.25516199   -11.603296    11.09297   -17.610635    17.10031
## Mar 2021     -0.25171815   -11.600145    11.09671   -17.607640    17.10420
## Apr 2021      0.04720522   -11.327168    11.42158   -17.348398    17.44281
## May 2021      0.02006269   -11.354474    11.39460   -17.375790    17.41592
## Jun 2021     -0.00635577   -11.381100    11.36839   -17.402526    17.38981
```

```
# Prediccion con 3 periodos
```

```
prediccion <- forecast(arima_model, h = 3) # 3 periodos adelante
```

```
print(prediccion)
```

```
##          Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
## Jan 2021      2.8449658    -5.321986    11.01192    -9.645309    15.33524
## Feb 2021     -0.2551620   -11.603296    11.09297   -17.610635    17.10031
## Mar 2021     -0.2517182   -11.600145    11.09671   -17.607640    17.10420
```

4.10 Validación del Modelo (Análisis de Residuos):

Un modelo ARIMA válido requiere que los residuos (la diferencia entre los valores observados y los valores predichos) cumplan con ciertos supuestos. Se realizaron las siguientes pruebas para validar el modelo:

- Media cero: Se aplicó una prueba t de una muestra para verificar si la media de los residuos es significativamente diferente de cero. Un p-valor alto como el obtenido en este caso indica que no hay evidencia suficiente para rechazar la hipótesis nula de media cero.
- Independencia: La prueba de Ljung-Box se utiliza para evaluar la independencia de los residuos. El p-valor alto indica que los residuos no están autocorrelacionados, lo cual es deseable.
- Normalidad: Se evaluó la normalidad de los residuos utilizando un qqplot y la prueba de Shapiro-Wilk. El qqplot muestra una desviación de la normalidad, especialmente en las colas de la distribución. La prueba de Shapiro-Wilk confirmó la no normalidad de los residuos con un p-valor muy bajo. Esta no normalidad puede afectar la validez de los intervalos de confianza y las pruebas de hipótesis relacionadas con el modelo. Se recomienda considerar transformaciones de la serie o modelos alternativos si la no normalidad es severa.

```
# Validación (detección de outliers, usando 'tsoutliers')
outliers <- tso(indice.ts, types = c("AO", "TC", "LS", "IO")) # Identificar outliers
plot(outliers)
```

```
## 'x' does not contain outliers to display
```

```
## NULL
```

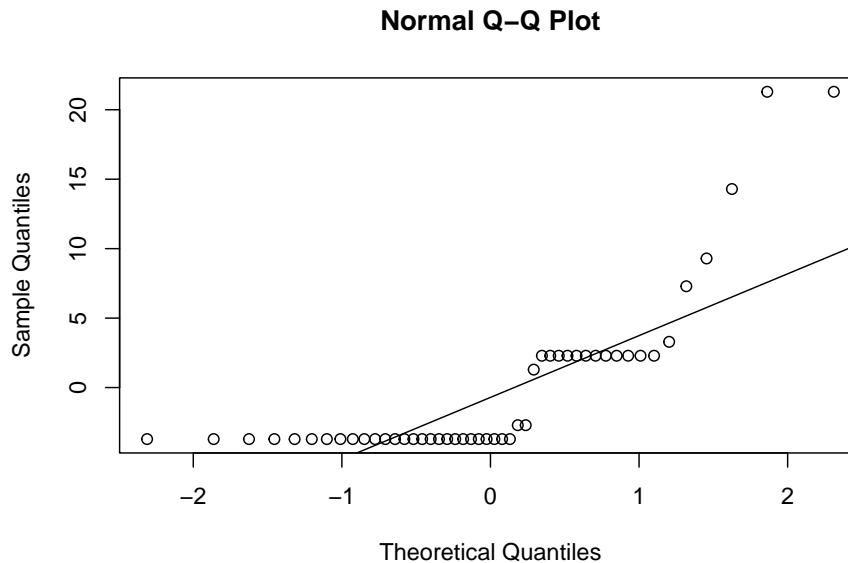
```
# Supuestos del modelo ARIMA (residuos)
mr <- outliers$fit$residuals
# Media cero
t.test(mr)
```

```
##
## One Sample t-test
##
## data: mr
## t = 1.1128e-12, df = 47, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -1.752442 1.752442
## sample estimates:
## mean of x
## 9.693727e-13
```

```
# Independencia
Box.test(mr, lag = 10, type = "Ljung-Box") # type = "Box-Pierce" también es válido
```

```
##
## Box-Ljung test
##
```

```
## data:  mr
## X-squared = 2.9461, df = 10, p-value = 0.9827
# Distribución normal
qqnorm(mr)
qqline(mr)
```



```
shapiro.test(mr)
```

```
##
## Shapiro-Wilk normality test
##
## data:  mr
## W = 0.64698, p-value = 1.721e-09
```

4.11 Conclusiones:

El análisis de series de tiempo proporciona información valiosa sobre el comportamiento pasado y futuro de las ventas del producto. El modelo ARIMA(2, 0, 1) ofrece una base para la predicción, pero la no normalidad de los residuos sugiere la necesidad de refinamiento.

Se recomienda:

Investigar a fondo las causas del punto de cambio detectado en el periodo 15.

Considerar la incorporación de variables externas al modelo, como precio, pro-

mociones, competencia, o factores económicos, para mejorar la precisión de las predicciones.

Explorar transformaciones de la serie, como la transformación logarítmica, para intentar estabilizar la varianza y mejorar la normalidad de los residuos.