

## Asignatura Series de Tiempo

Brayan Hernandez Cardona - Adriana Janneth Avila - WILLIAM ROJAS CARABALI

2024-12-09



# Contents

<b>1</b>	<b>Acerca de este libro</b>	<b>5</b>
1.1	Introducción . . . . .	5
<b>2</b>	<b>Pronóstico de Ventas en empresa Retail</b>	<b>7</b>
2.1	Optimizando el Futuro del Retail . . . . .	7
2.2	Estructura del dataset . . . . .	7
<b>3</b>	<b>Análisis de Series de Tiempo de Ventas</b>	<b>9</b>
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
<b>4</b>	<b>Preprocesamiento y visualización</b>	<b>15</b>
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: . . . . .	28
<b>5</b>	<b>Holt-Winters para analisis de tiempo</b>	<b>29</b>
5.1	Introducción . . . . .	29
5.2	Metodología: . . . . .	29
5.3	Resultados . . . . .	30

5.4	Conclusiones y Recomendaciones: . . . . .	34
5.5	Reflexión final . . . . .	35
<b>6</b>	<b>Modelos estacionarios en series de tiempo</b>	<b>37</b>
6.1	Análisis de las Ventas Históricas . . . . .	37
6.2	Metodología de Pronóstico . . . . .	40
6.3	Resultados del Pronóstico: . . . . .	41
6.4	Análisis de Residuos . . . . .	42
6.5	Conclusiones y Recomendaciones: . . . . .	42
<b>7</b>	<b>Group 6: Documento Bookdown - Propuesta avance 6</b>	<b>45</b>
7.1	Introducción . . . . .	45
7.2	Preprocesamiento de Datos . . . . .	46
7.3	Prueba de Estacionariedad . . . . .	47
7.4	Análisis de ACF y PACF . . . . .	47
7.5	Ajuste del Modelo ARIMA . . . . .	49
7.6	Predicción . . . . .	49
7.7	Detección de Puntos de Cambio . . . . .	50
7.8	Conclusión . . . . .	52
<b>8</b>	<b>Redes Neuronales de Elman y Jordan</b>	<b>53</b>
8.1	Introducción . . . . .	53
8.2	Preparación de Datos: . . . . .	53
8.3	Modelo de Elman: . . . . .	53
8.4	Modelo de Jordan: . . . . .	54
8.5	Resultados: . . . . .	54
8.6	Análisis de Resultados: . . . . .	54
8.7	Conclusión: . . . . .	54

# 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','numeri  
)
```

```

# 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.

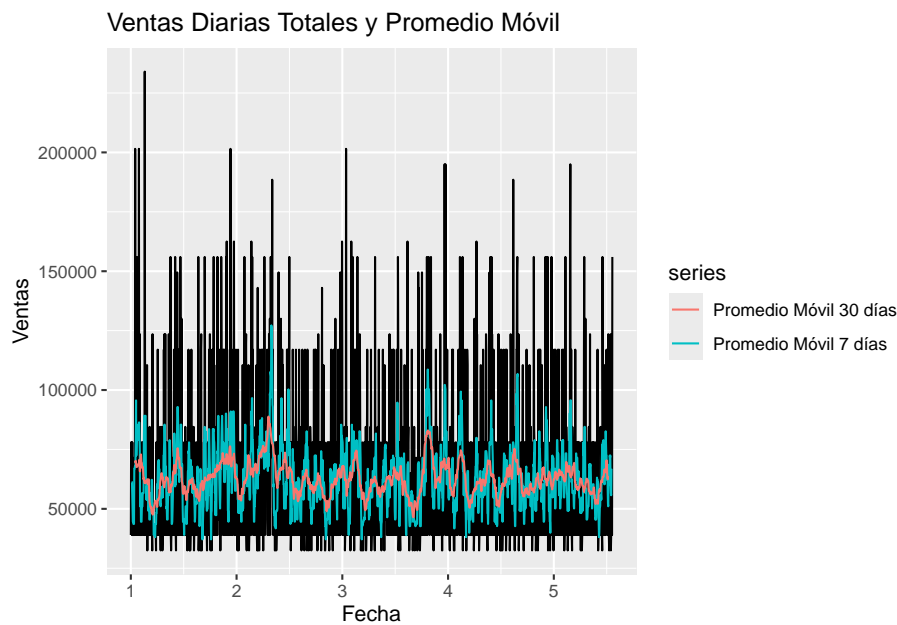
```

# 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.

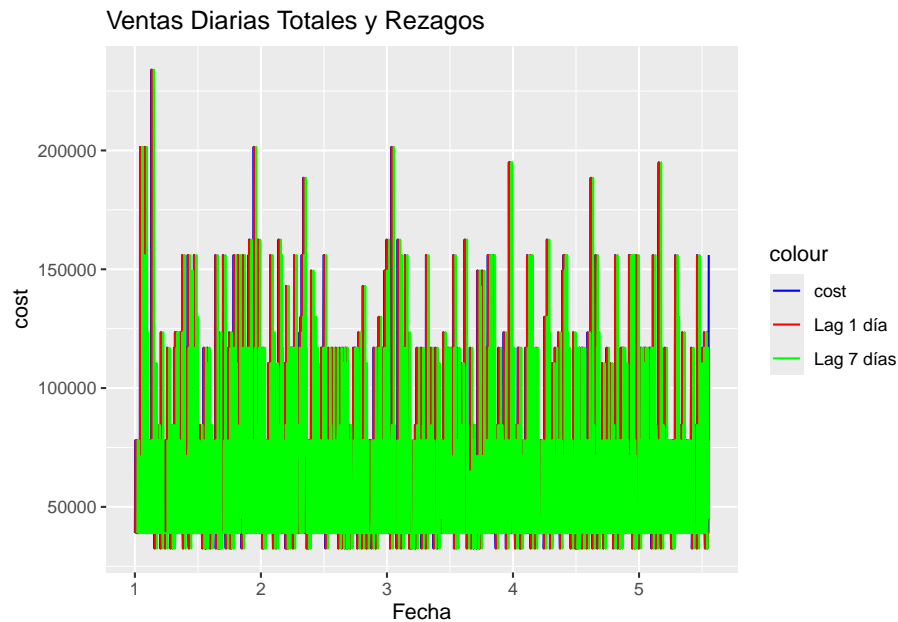
```
# 2. Rezagos (lags)
# Calcular el rezago de las ventas (ejemplo con 1 y 7 días de rezago)
ventas_lag1 <- lag(as.numeric(ventas_ts), 1)

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"))
```



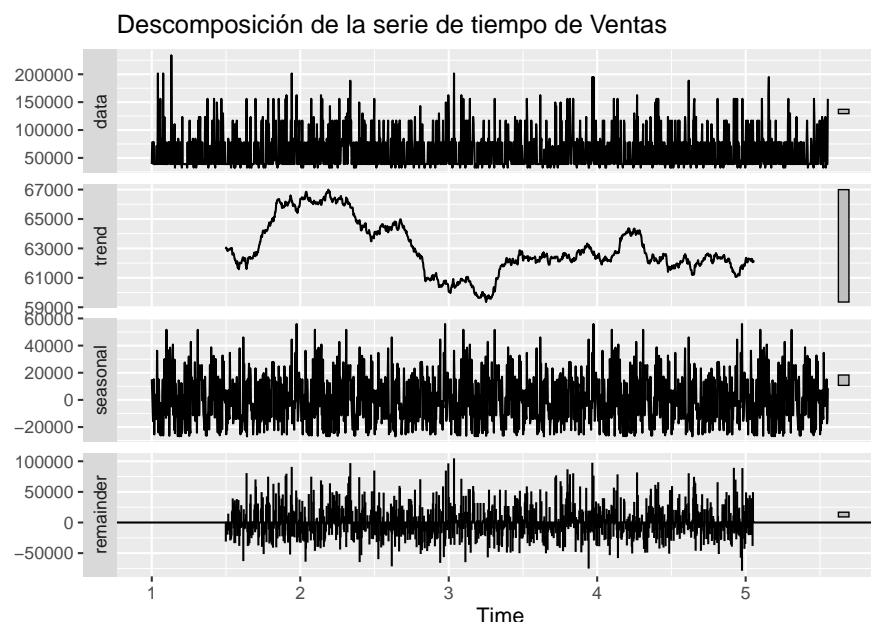
### 3.4 Análisis de Estacionalidad (Descomposición):

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")
```

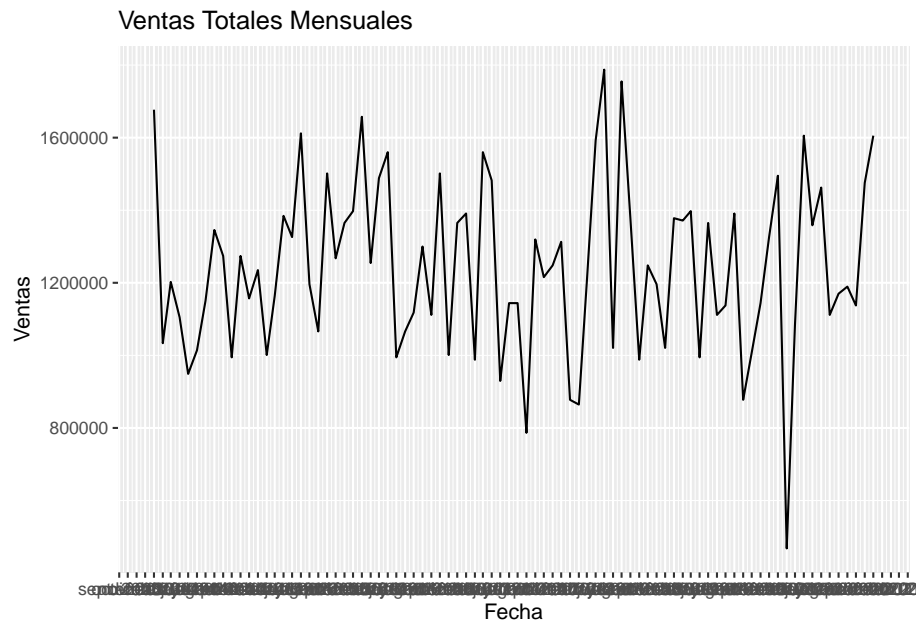


### 3.5 Análisis de Estacionalidad (Promedio Mensual):

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"), ]

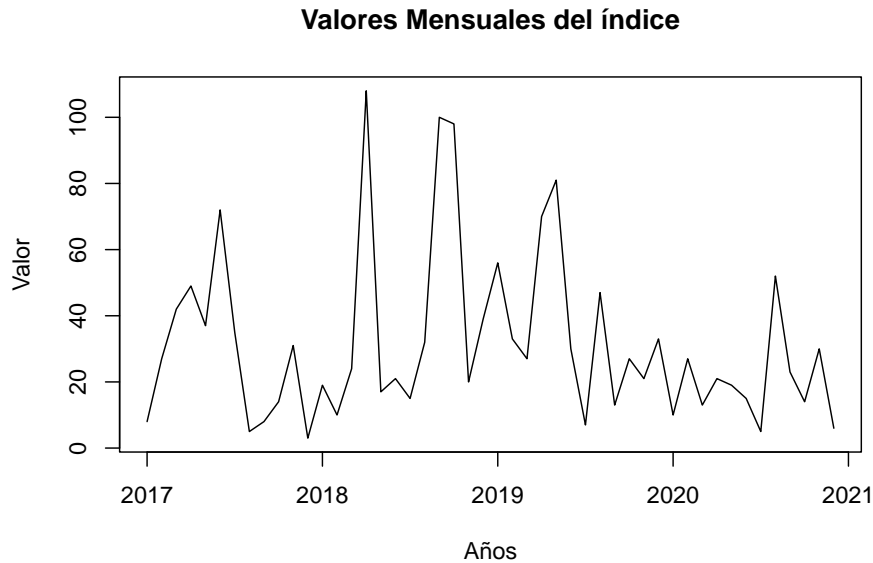
ventas_diarias <- aggregate(quantity ~ transaction_date, data = datos, FUN = sum)

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 (DICKEY-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 = -2.9377, Lag order = 3, p-value = 0.1996
```

```
## 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

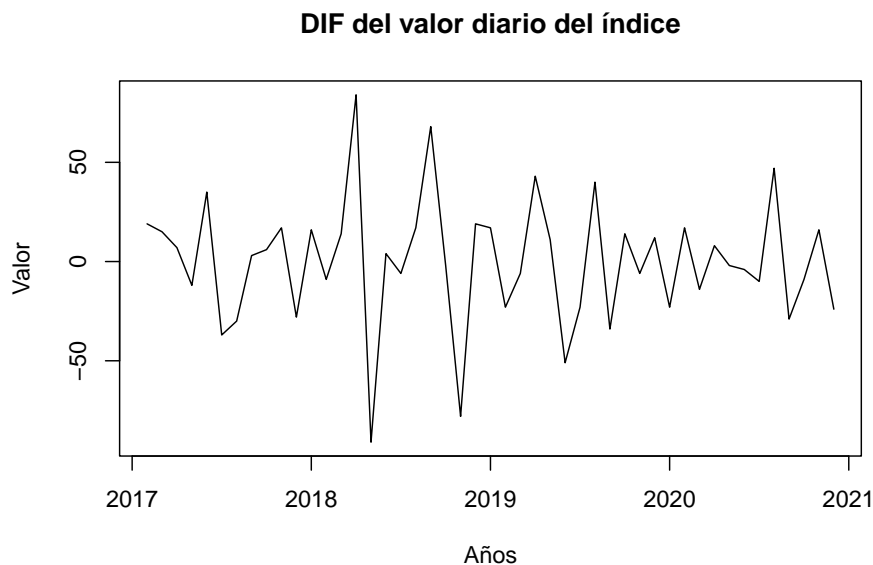
#Volver a realizar la prueba
adf.test(dif.indice.ts)

##
## Augmented Dickey-Fuller Test
##
## data: dif.indice.ts
## Dickey-Fuller = -5.6807, 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.

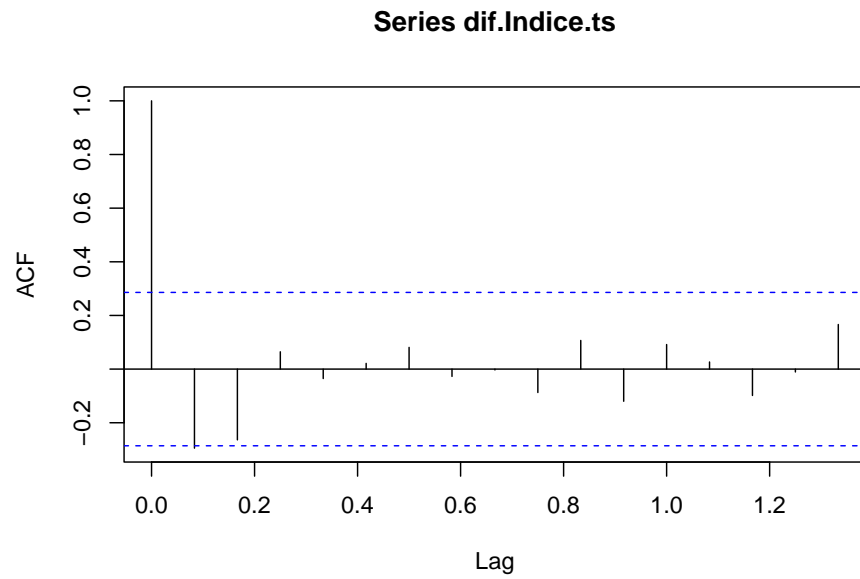
```
# Graficar la serie diferenciada
plot(dif.indice.ts, main = "DIF del valor diario del índice", xlab = "Años", ylab = "DIF del valor diario del índice")
```



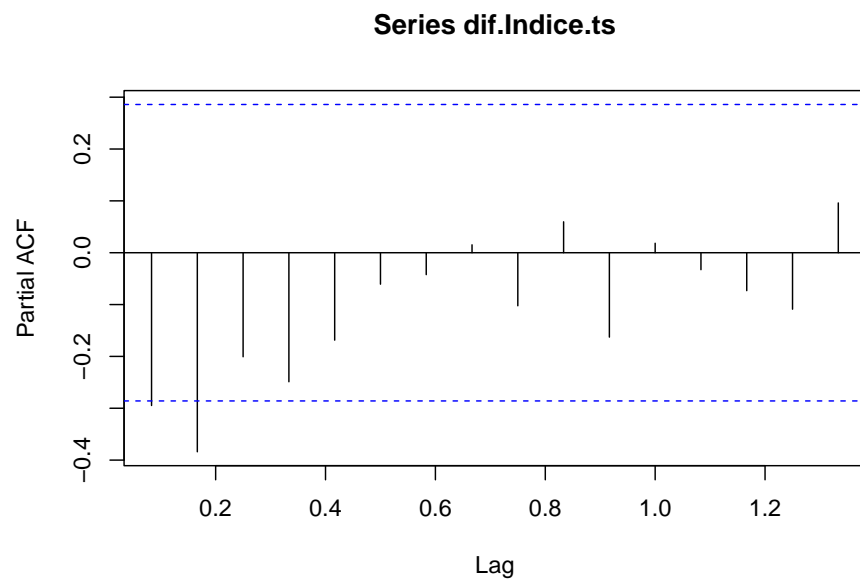
## 4.5 Funciones de Autocorrelación (ACF) y Autocorrelación Parcial (PACF):

- 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")
```

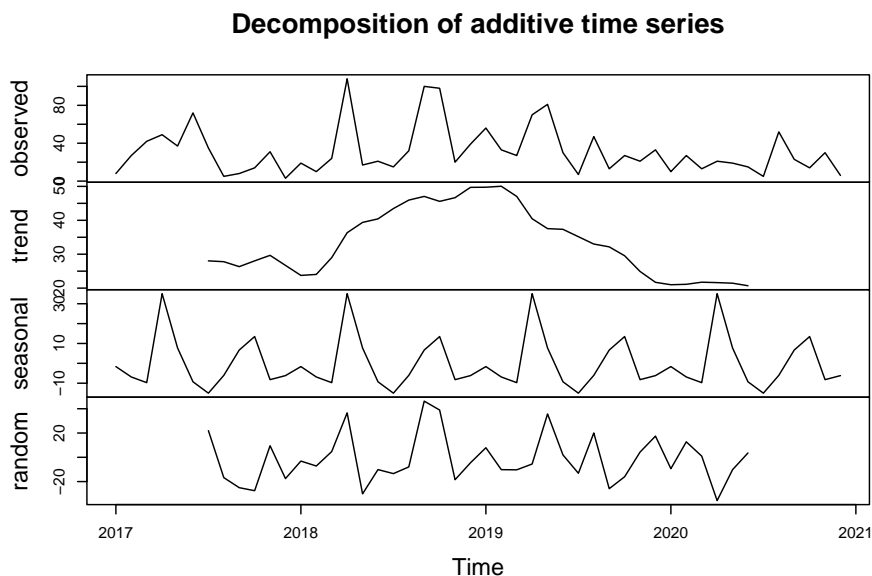


## 4.6 Descomposición de la Serie (Componente Estacional):

- 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(0,0,1) with zero mean
##
## Coefficients:
##          ma1
##        -0.8544
## s.e.      0.1099
##
## sigma^2 = 692.4: log likelihood = -220.53
## AIC=445.07   AICc=445.34   BIC=448.77
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set -0.4037116 26.03277 19.42164 -6.498803 189.911 0.5360862 0.1522609
```

## 4.8 Detección de Puntos de Cambio:

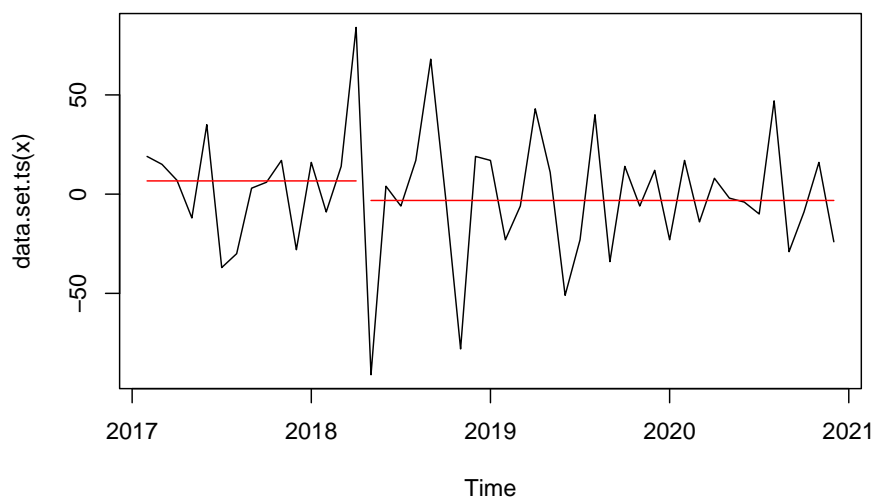
- La función `cpt.mean()` se utiliza para identificar puntos de cambio en la media de la serie. En este caso, se detectó un punto de cambio en el periodo 15 de la serie diferenciada. Esto indica un cambio abrupto en el comportamiento de las ventas en ese momento. Es crucial investigar las posibles causas 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:

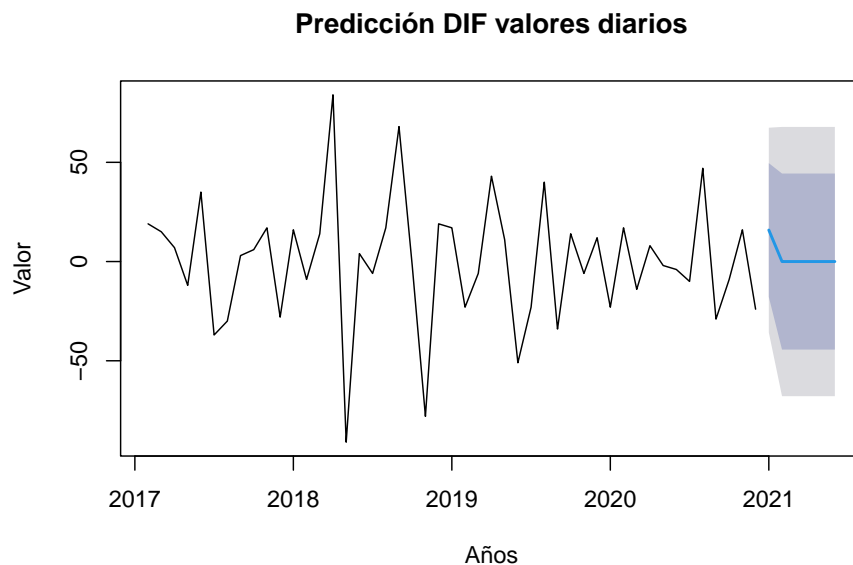
El `print(prediccion)` para 3 periodos ofrece la misma información, pero solo para los 3 meses siguientes. Es útil comparar las predicciones a 3 y 6 periodos para observar cómo evoluciona la incertidumbre a medida que se proyecta más hacia el futuro. Generalmente, la incertidumbre (y por lo tanto, la 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      15.86203 -17.86100  49.58505 -35.71289  67.43694
## Feb 2021       0.00000 -44.35647  44.35647 -67.83736  67.83736
## Mar 2021       0.00000 -44.35647  44.35647 -67.83736  67.83736
## Apr 2021       0.00000 -44.35647  44.35647 -67.83736  67.83736
## May 2021       0.00000 -44.35647  44.35647 -67.83736  67.83736
## Jun 2021       0.00000 -44.35647  44.35647 -67.83736  67.83736
```

```
# Prediccion con 3 periodos
```

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

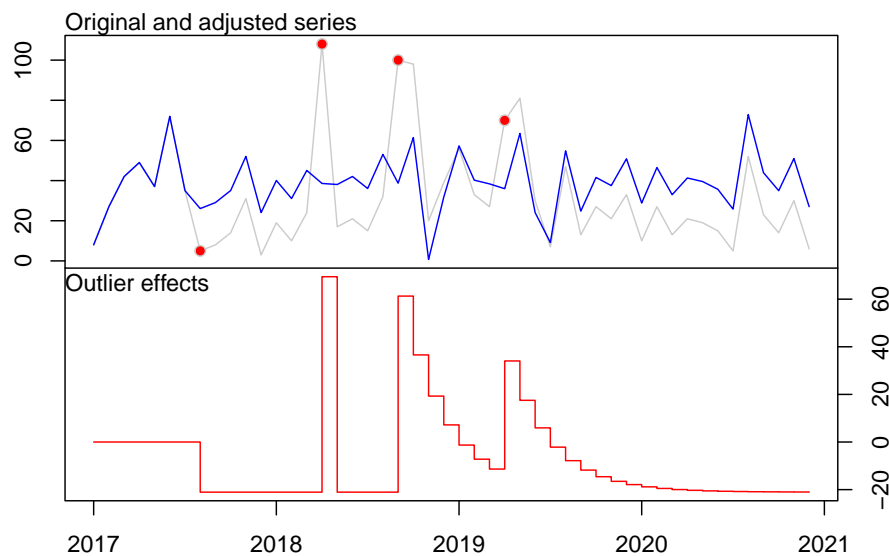
```
##          Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
## Jan 2021      15.86203 -17.86100  49.58505 -35.71289  67.43694
## Feb 2021       0.00000 -44.35647  44.35647 -67.83736  67.83736
## Mar 2021       0.00000 -44.35647  44.35647 -67.83736  67.83736
```

## 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)
```



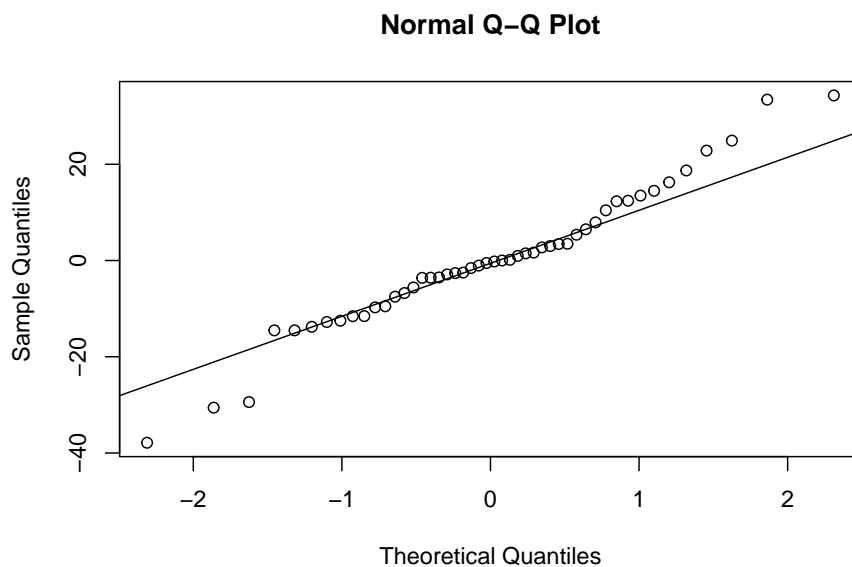
```
# Supuestos del modelo ARIMA (residuos)
mr <- outliers$fit$residuals
# Media cero
t.test(mr)
```

```
##
## One Sample t-test
##
## data: mr
## t = -2.5502e-15, df = 47, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -4.203924  4.203924
## sample estimates:
## mean of x
## -5.329071e-15

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

##
## Box-Ljung test
##
## data: mr
## X-squared = 8.3789, df = 10, p-value = 0.5919

# Distribución normal
qqnorm(mr)
qqline(mr)
```



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

```
##  
## Shapiro-Wilk normality test  
##  
## data: mr  
## W = 0.96886, p-value = 0.2287
```

### 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, promociones, 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.

## Chapter 5

# Holt-Winters para analisis de tiempo

### 5.1 Introducción

La gestión eficiente del inventario es un pilar fundamental para el éxito en el competitivo mercado minorista actual. Un inventario excesivo genera costos de almacenamiento, obsolescencia y depreciación, mientras que un inventario insuficiente resulta en pérdidas de ventas, insatisfacción del cliente y daño a la imagen de la marca. Encontrar el equilibrio óptimo entre la oferta y la demanda es crucial para maximizar la rentabilidad y la eficiencia operativa.

Este informe se centra en el análisis de las series temporales de ventas de una empresa retail, con el propósito de pronosticar la demanda futura y, en consecuencia, optimizar la gestión de inventario. A través de la aplicación de técnicas estadísticas de series temporales, podemos identificar patrones, tendencias y estacionalidades en los datos históricos de ventas, lo que nos permite anticipar las fluctuaciones de la demanda y tomar decisiones informadas sobre la cantidad de productos a mantener en stock.

### 5.2 Metodología:

Se utilizó el método de Holt-Winters para el análisis y pronóstico de la demanda. Este método es particularmente adecuado para series temporales que presentan tanto tendencia como estacionalidad, características observadas en nuestros datos de ventas. El proceso incluyó los siguientes pasos:

- **Preparación de los datos:** Se agregaron las cantidades vendidas por fecha, convirtiendo la información en una serie temporal.

- **Suavizamiento exponencial simple:** Se aplicó inicialmente un suavizamiento exponencial simple para una primera aproximación al comportamiento de la serie.
- **Modelo Holt-Winters:** Se implementó el modelo Holt-Winters, incorporando componentes de tendencia y estacionalidad para un ajuste más preciso.
- **Evaluación del modelo:** Se calculó el error de suma de cuadrados (SSE) para evaluar la precisión del modelo.
- **Pronóstico:** Se generaron pronósticos para los próximos 12 periodos.

```
# Cargar la librería forecast para predicciones
library(forecast)
library(Metrics)

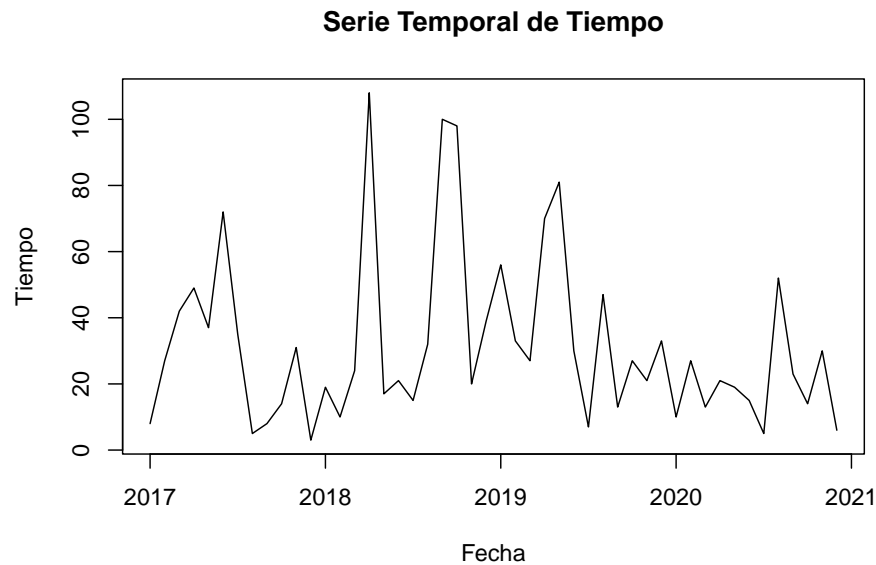
# Leer los datos y transformar los datos
datos <- readxl::read_excel("MUESTRA SERIE TIEMPO.xlsx")
datos$transaction_date <- as.Date(datos$transaction_date, origin="1899-12-30")
datos <- aggregate(quantity ~ transaction_date, data = datos, FUN = sum)
```

### 5.3 Resultados

- **Visualización de la serie:** El gráfico inicial de la serie temporal de ventas ilustra la evolución de la cantidad de productos vendidos a lo largo del tiempo. Se observa una clara tendencia general al alza, lo que indica un crecimiento en la demanda a lo largo de los años. Además, se aprecia un patrón estacional repetitivo, con picos y valles que ocurren en periodos similares cada año. Esta estacionalidad sugiere la influencia de factores recurrentes, como temporadas de compras o eventos específicos, en el comportamiento de las ventas.

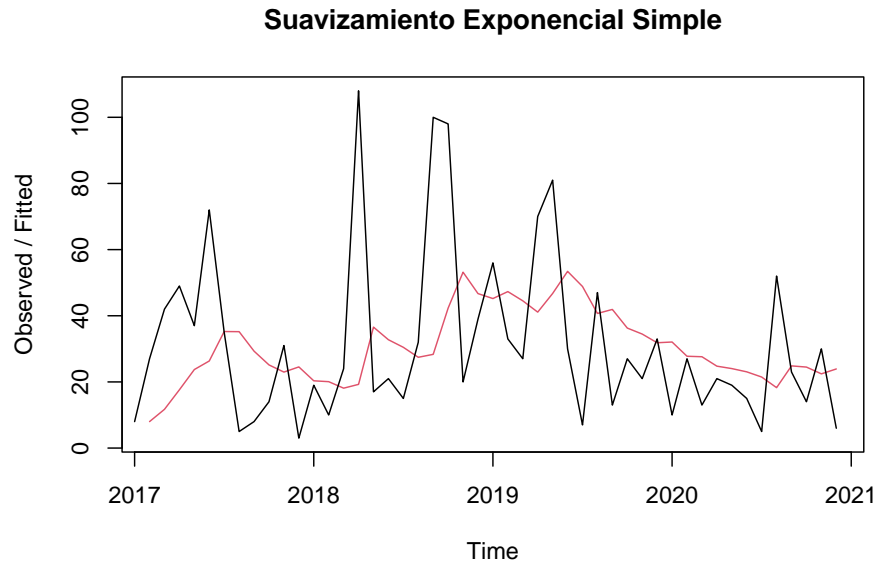
```
# Convertir los datos a un objeto de serie temporal
serie_tiempo <- ts(datos$quantity, start = c(2017, 1), end = c(2020, 12), frequency = 12)

# Visualizar la serie temporal
plot(serie_tiempo, main = "Serie Temporal de Tiempo", ylab = "Tiempo", xlab = "Fecha")
```



\* **Suavizamiento exponencial simple:** La aplicación del suavizamiento exponencial simple genera una línea suavizada que captura la tendencia general de la serie. Sin embargo, este método no considera la estacionalidad presente en los datos. Por lo tanto, la línea suavizada no refleja las fluctuaciones periódicas observadas en la serie original, subestimando los picos y sobrestimando los valles. Esto evidencia la necesidad de un modelo más sofisticado que incorpore la componente estacional.

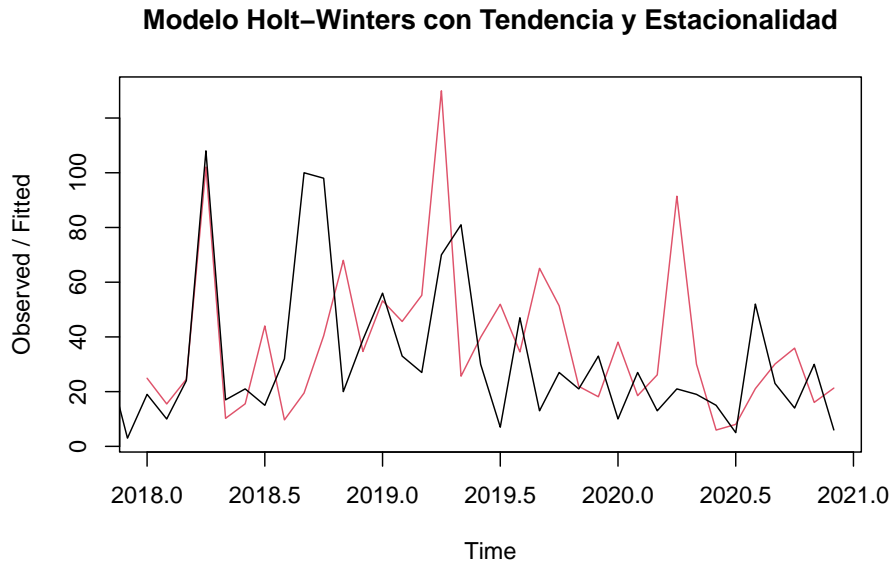
```
# Suavizamiento exponencial simple  
suavizado_simple <- HoltWinters(serie_tiempo, beta = FALSE, gamma = FALSE)  
plot(suavizado_simple, main = "Suavizamiento Exponencial Simple")
```



- **Modelo Holt-Winters:** El modelo Holt-Winters, al incorporar componentes de tendencia y estacionalidad, se ajusta de manera significativamente más precisa a los datos observados. La línea ajustada del modelo sigue de cerca las fluctuaciones de la serie, capturando tanto la tendencia ascendente como los patrones estacionales. Esto indica que el modelo es capaz de representar el comportamiento histórico de las ventas con mayor fidelidad, lo que a su vez aumenta la confiabilidad de las predicciones generadas. Asimismo esto no es indicio que presentar mejores resultados, si no que más bien, visualmente el metodo de winters trata de adaptarse al comportamiento de la demanda.

```
# Aplicar el modelo Holt-Winters con componentes de tendencia y estacionalidad
modelo_holt_winters <- HoltWinters(serie_tiempo)
plot(modelo_holt_winters, main = "Modelo Holt-Winters con Tendencia y Estacionalidad")
```





- **Evaluación del modelo:** El modelo de winters se ve que se tiende a tomar una forma de picos y valles tratando de organizarse según la estacionalidad de los datos, en este sentido se puede ver que para ese tipo de datos puede ser mucho más eficaz que la suavización simple.

Para esto caso cabe resaltar que los datos de ventas tienen un comportamiento estable o perpetuo, es decir, no representa una estacionalidad marcada a lo largo del tiempo y la distribución de su forma es más homogénea, esto afecta al modelo de winters dado que no es un dataset para el que este modelo se halla pensado.

- **Pronóstico a 12 periodos:** El gráfico del pronóstico muestra la proyección de la demanda para los próximos 12 periodos, considerando tanto la tendencia ascendente como el patrón estacional identificado en los datos históricos. Las líneas azules representan el intervalo de confianza del pronóstico, indicando el rango dentro del cual se espera que se encuentren las ventas futuras con un cierto nivel de probabilidad. Este pronóstico proporciona una herramienta valiosa para la planificación de la gestión de inventario, permitiendo anticipar las fluctuaciones de la demanda y tomar decisiones informadas sobre la adquisición y el almacenamiento de productos.

```
# Calcular las métricas de precisión
rmse <- rmse(serie_tiempo, modelo_holt_winters$fitted)
mae <- mae(serie_tiempo, modelo_holt_winters$fitted)
mape <- mape(serie_tiempo, modelo_holt_winters$fitted) * 100
```

```

# Imprimir las métricas
cat("RMSE:", rmse, "\n")

## RMSE: 36.73575

cat("MAE:", mae, "\n")

## MAE: 27.51567

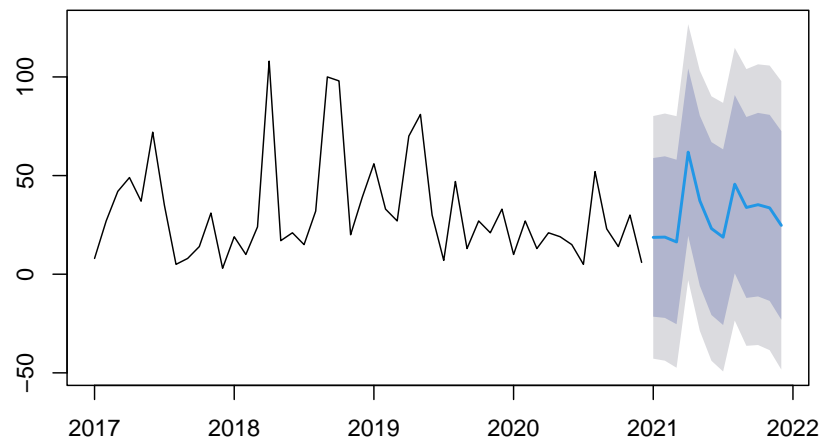
cat("MAPE:", mape, "%", "\n")

## MAPE: 107.8621 %

# Generar pronósticos para los próximos 12 períodos
pronostico <- forecast(modelo_holt_winters, h = 12)
plot(pronostico, main = "Pronóstico de la Serie Temporal con Holt-Winters")

```

**Pronóstico de la Serie Temporal con Holt-Winters**



## 5.4 Conclusiones y Recomendaciones:

El análisis de series temporales realizado mediante el modelo Holt-Winters ha demostrado ser una herramienta eficaz para pronosticar las ventas de un conjunto estacional. Pero así mismo, cuando se prueba sobre un dataset con datos más perpetuos, puede verse que le cuesta mucho ya que sobreestima el movimiento de la demanda.

A pesar de esto, podemos ver como las series de tiempo ha estudiado diversidad

de comportamientos y esta totalmente comprobado que cada modelo es muy bueno en su campo y este punto debe respetarse.

El pronóstico generado para los próximos 12 periodos ofrece una visión anticipada de la demanda esperada, lo que permite una planificación más precisa y eficiente. En base a estos resultados.

## 5.5 Reflexión final

- Este análisis demuestra el valor estratégico que aporta la ciencia de datos a la gestión de inventario en el sector retail. La capacidad de analizar grandes volúmenes de datos, identificar patrones complejos y generar pronósticos precisos, como se ha demostrado con la aplicación del modelo Holt-Winters, transforma la toma de decisiones de un enfoque reactivo a uno proactivo.
- En un entorno cada vez más competitivo y dinámico, la ciencia de datos se convierte en una herramienta indispensable para anticipar las fluctuaciones de la demanda, optimizar los niveles de inventario y maximizar la eficiencia de la cadena de suministro. La inversión en herramientas y capacidades analíticas no solo mejora la gestión de inventario, sino que también impulsa la rentabilidad, la satisfacción del cliente y la ventaja competitiva en el mercado. A medida que la cantidad de datos disponibles continúa creciendo, el potencial de la ciencia de datos para optimizar las operaciones y la toma de decisiones estratégicas en el sector retail se amplía aún más. La adopción de estas técnicas no es simplemente una ventaja, sino una necesidad para las empresas que buscan prosperar en el futuro.



## Chapter 6

# Modelos estacionarios en series de tiempo

Este informe presenta un análisis de las ventas históricas y un pronóstico para los próximos 12 meses. El objetivo es proporcionar información clara y accionable para la toma de decisiones estratégicas. Utilizamos métodos estadísticos robustos para identificar patrones, tendencias y estacionalidades en los datos de ventas, lo que nos permite anticipar la demanda futura.

### 6.1 Análisis de las Ventas Históricas

Inicialmente, se realizó una revisión detallada del historial de ventas. Se observaron fluctuaciones a lo largo del tiempo, lo que sugiere la presencia de patrones recurrentes y posibles factores externos que influyen en las ventas. Para comprender mejor estas fluctuaciones, aplicamos diversas técnicas de análisis de series de tiempo. La visualización de los datos a través de gráficos nos permitió identificar tendencias generales y estacionalidades. Se emplearon herramientas estadísticas como la función de autocorrelación para determinar la relación entre las ventas en diferentes momentos del tiempo.

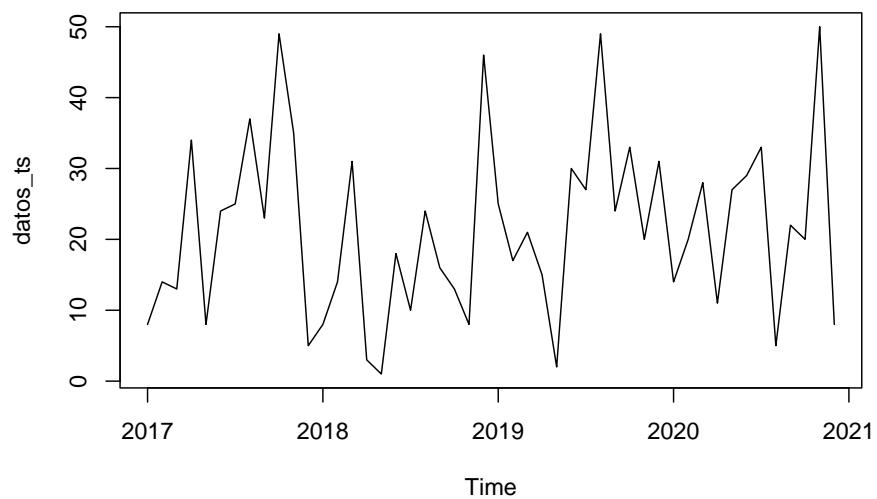
```
library(forecast)
library(tseries)

datos <- readxl::read_excel("MUESTRA SERIE TIEMPO.xlsx")
sum(datos$quantity)

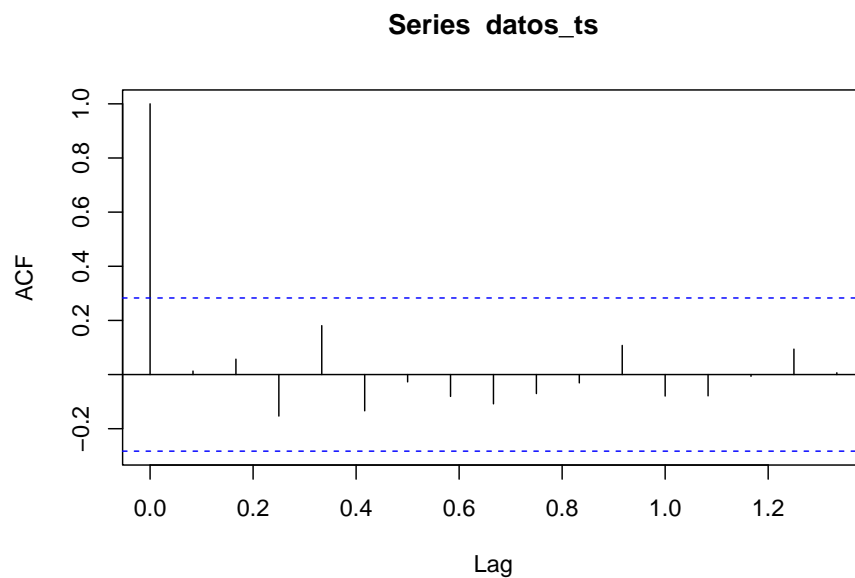
## [1] 56617

datos$transaction_date <- as.Date(datos$transaction_date, origin="1899-12-30")
```

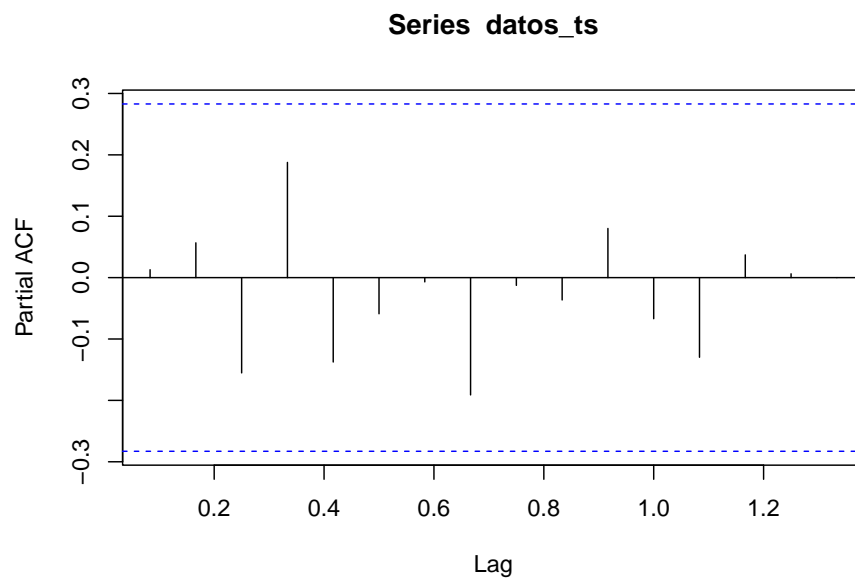
```
datos_ts <- ts(datos$quantity, frequency = 12, start = c(2017, 1), end = c(2020,12))  
plot(datos_ts)
```



```
acf(datos_ts)
```



```
pacf(datos_ts)
```



Se comprobó la estacionariedad de la serie de tiempo, es decir, se verificó si las propiedades estadísticas de la serie, como la media y la varianza, permanecen constantes a lo largo del tiempo. En caso de no ser estacionaria, se aplicaron

transformaciones para estabilizar la serie y obtener resultados más fiables en el pronóstico.

```
adf.test(datos_ts)

##
## Augmented Dickey-Fuller Test
##
## data:  datos_ts
## Dickey-Fuller = -2.7329, Lag order = 3, p-value = 0.2814
## alternative hypothesis: stationary
#Si la prueba indica que la serie no es estacionaria, deberás diferenciarla:
datos_ts_diff <- diff(datos_ts)

adf.test(datos_ts_diff)

##
## Augmented Dickey-Fuller Test
##
## data:  datos_ts_diff
## Dickey-Fuller = -4.6247, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

## 6.2 Metodología de Pronóstico

Para generar el pronóstico, utilizamos un modelo ARIMA. Este modelo es especialmente adecuado para series de tiempo que presentan autocorrelación, es decir, cuando los valores pasados influyen en los valores futuros. El modelo ARIMA se seleccionó automáticamente con base en criterios de información que buscan el mejor ajuste a los datos históricos. Además del modelo ARIMA, se exploraron otras metodologías de pronóstico, como el modelo de Holt-Winters, el cual es particularmente útil para series de tiempo que presentan tendencias y estacionalidades. Esto nos permitió comparar diferentes enfoques y seleccionar el modelo que mejor se adapta a las características de los datos.

```
modelo_auto <- auto.arima(datos_ts_diff, ic = "bic")
summary(modelo_auto)

## Series: datos_ts_diff
## ARIMA(1,0,0) with zero mean
##
## Coefficients:
##          ar1
##        -0.5468
## s.e.      0.1299
##
```

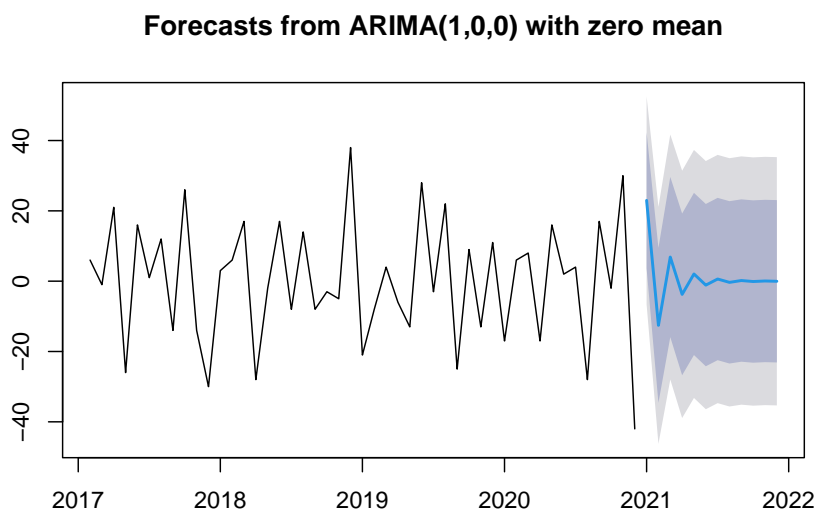


```
## sigma^2 = 227.5: log likelihood = -193.9
## AIC=391.79 AICc=392.07 BIC=395.49
##
## Training set error measures:
##           ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 0.4678724 14.92075 11.85491 80.97321 148.3785 0.6092831 -0.1055607
```

### 6.3 Resultados del Pronóstico:

El pronóstico generado por el modelo se presenta en un gráfico que muestra las ventas esperadas para los próximos 12 meses, junto con intervalos de confianza. Estos intervalos indican el rango dentro del cual se espera que se encuentren las ventas futuras con un cierto nivel de probabilidad. Es importante destacar que los pronósticos son estimaciones basadas en datos históricos y en las propiedades estadísticas de la serie de tiempo. Por lo tanto, existe un grado de incertidumbre asociado a los valores pronosticados. Es fundamental monitorear el desempeño real de las ventas y compararlas con el pronóstico para realizar ajustes en el modelo.

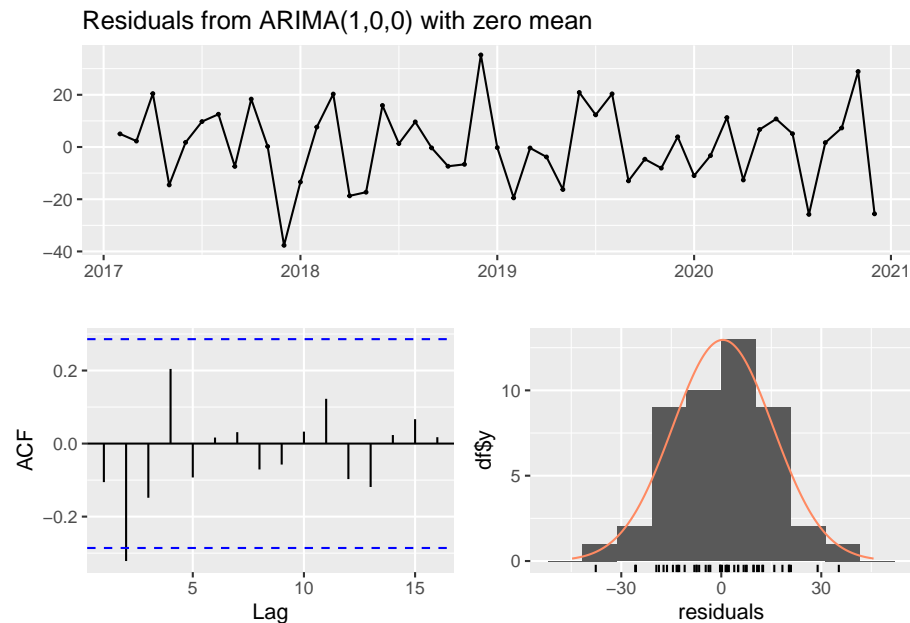
```
pronóstico <- forecast(modelo_auto, h = 12)
plot(pronóstico)
```



## 6.4 Análisis de Residuos

Para validar la confiabilidad del modelo, se realizó un análisis de los residuos, que son las diferencias entre los valores reales y los valores pronosticados por el modelo. Se verificó si los residuos cumplen con las propiedades estadísticas deseadas, como la media cero y la varianza constante. Este análisis nos permite evaluar la calidad del ajuste del modelo y detectar posibles desviaciones o patrones no explicados.

```
checkresiduals(modelo_auto)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with zero mean
## Q* = 10.279, df = 8, p-value = 0.246
##
## Model df: 1.   Total lags used: 9
```

## 6.5 Conclusiones y Recomendaciones:

El análisis de series de tiempo proporciona información valiosa. El pronóstico de ventas permite anticipar la demanda futura y optimizar la gestión de inventarios, la asignación de recursos y las estrategias de marketing.

Se recomienda utilizar el pronóstico generado como una herramienta de apoyo

para la toma de decisiones, pero siempre teniendo en cuenta la incertidumbre inherente a los pronósticos. Es importante monitorear el desempeño real de las ventas y ajustar las estrategias de la empresa en función de los resultados observados.



## Chapter 7

# Group 6: Documento Bookdown - Propuesta avance 6

### 7.1 Introducción

Este informe presenta un análisis de los datos de series temporales de la cantidad de ventas. Los datos fueron agregados por fecha de transacción, y se aplicaron varios tests y modelos estadísticos para entender los patrones subyacentes y predecir tendencias futuras. El análisis se realizó utilizando R, centrándose en la estacionariedad, el ajuste del modelo y la predicción utilizando ARIMA, así como en la detección de valores atípicos y puntos de cambio.

```
if(!require("forecast")) install.packages("forecast")
if(!require("tseries")) install.packages("tseries")
if(!require("timsac")) install.packages("timsac")
if(!require("ggplot2")) install.packages("ggplot2")
if(!require("changepoint")) install.packages("changepoint")
if(!require("tsoutliers")) install.packages("tsoutliers")

library(forecast)
library(tseries)
library(timsac)
library(ggplot2)
library(changepoint)
library(tsoutliers)
```

## 7.2 Preprocesamiento de Datos

Los datos de cantidad de ventas fueron procesados inicialmente agregando las cantidades por fecha de transacción. La serie temporal se creó comenzando desde la primera fecha disponible, utilizando una frecuencia diaria para asegurar una cobertura integral del comportamiento de las ventas a lo largo del tiempo.

```
datos<-readxl::read_excel("MUESTRA SERIE TIEMPO.xlsx")
datos$transaction_date <- as.Date(datos$transaction_date, origin = "1900-01-01")

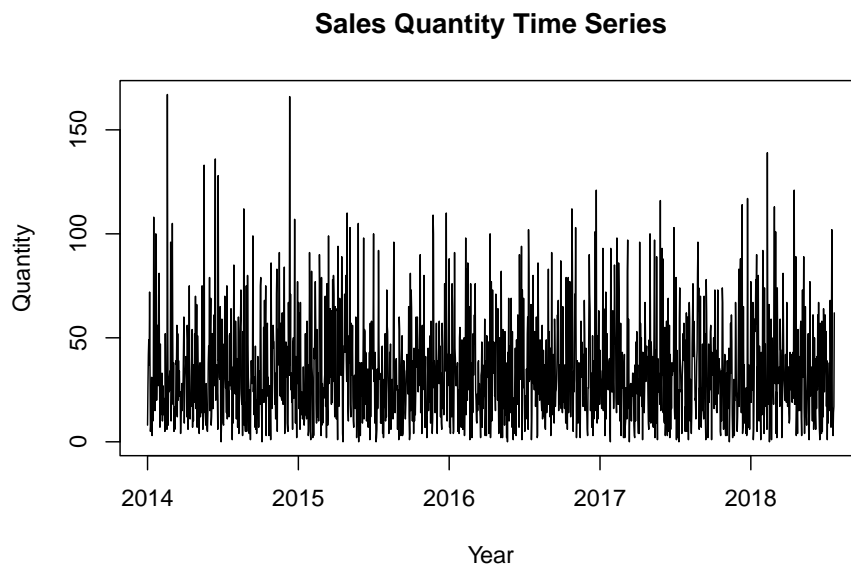
datos <- datos[order(datos$transaction_date), ]

# Aggregate the quantity sold per day (or other metric as needed)
agg_data <- aggregate(quantity ~ transaction_date, data = datos, sum)

# Convert the data to a time series starting from the earliest date
start_year <- as.numeric(format(min(agg_data$transaction_date), "%Y"))
start_day <- as.numeric(format(min(agg_data$transaction_date), "%j"))

# Create a time series object from the aggregated data
quantity.ts <- ts(agg_data$quantity, start = c(start_year, start_day), frequency = 365)

# Plot the original time series
plot(quantity.ts, main="Sales Quantity Time Series", ylab="Quantity", xlab="Year")
```



## 7.3 Prueba de Estacionariedad

Para determinar si la serie temporal era estacionaria, se aplicó un test de Dickey-Fuller Aumentado (ADF). Los resultados del test fueron:

- Estadístico Dickey-Fuller: -12.835
- Orden de Rezago: 11
- p-valor: 0.01

El p-valor fue menor a 0.05, lo cual indica que la serie temporal es estacionaria. Esto es crucial para garantizar la aplicabilidad de muchas técnicas de predicción estadística, ya que las series estacionarias generalmente producen modelos más fiables e interpretables.

```
# Test for stationarity using Augmented Dickey-Fuller test
adf_result <- adf.test(quantity.ts, alternative = "stationary")
print(adf_result)
```

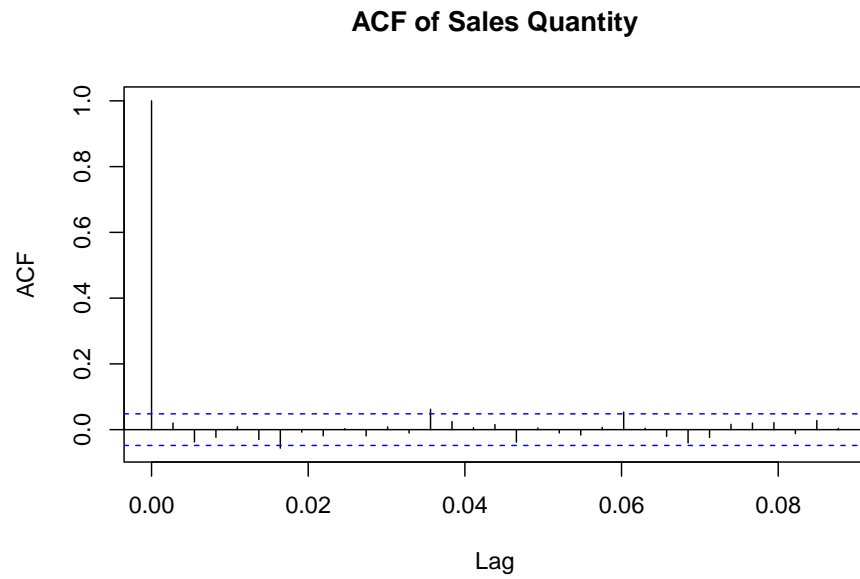
```
##
##   Augmented Dickey-Fuller Test
##
## data:   quantity.ts
## Dickey-Fuller = -12.835, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

## 7.4 Análisis de ACF y PACF

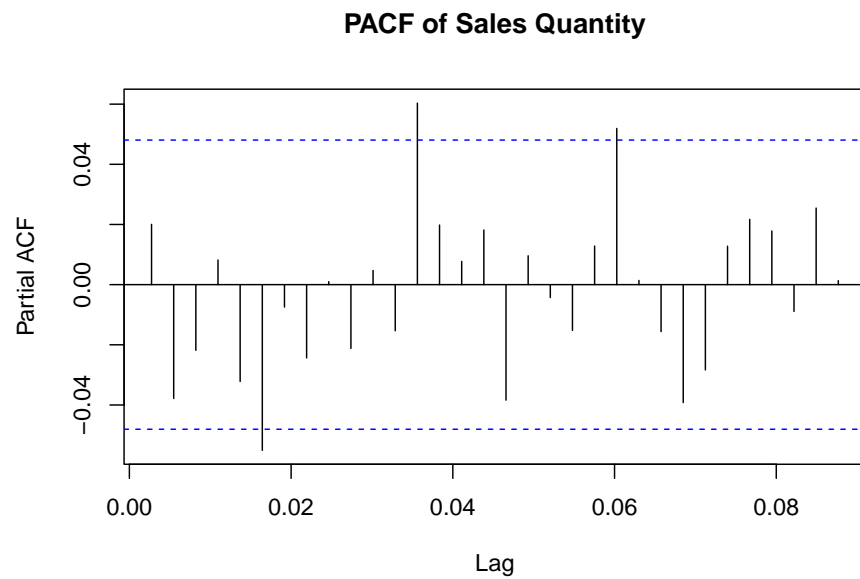
Las gráficas de la Función de Autocorrelación (ACF) y la Función de Autocorrelación Parcial (PACF) se utilizaron para determinar los parámetros apropiados para los términos AR y MA en el modelo ARIMA.

- Gráfica ACF: Mostró autocorrelaciones significativas, lo cual ayudó a identificar el componente de media móvil (MA).
- Gráfica PACF: Mostró autocorrelaciones parciales significativas, lo cual ayudó a identificar el componente autorregresivo (AR).

```
# ACF and PACF plots to determine AR and MA terms
acf(quantity.ts, main="ACF of Sales Quantity")
```



```
pacf(quantity.ts, main="PACF of Sales Quantity")
```



Estas gráficas proporcionan información sobre las dependencias temporales en los datos, que son fundamentales para seleccionar un modelo ARIMA adecuado.



## 7.5 Ajuste del Modelo ARIMA

Utilizando la función `auto.arima`, se seleccionó automáticamente el mejor modelo ARIMA. El modelo elegido fue:

- ARIMA(0,0,0) con media distinta de cero
- Coeficiente de la Media: 9.6422 (Error Estándar: 0.1255)
- $\sigma^2$  (Estimación de Varianza): 26.2

También se calcularon los valores de verosimilitud, AIC y BIC para evaluar la bondad del ajuste del modelo:

- Verosimilitud Logarítmica: -7699.62
- AIC: 15403.23
- BIC: 15414.06

```
# Fit ARIMA model using auto.arima
model <- auto.arima(quantity.ts)
print(model)

## Series: quantity.ts
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##          mean
##       34.0451
## s.e.    0.6082
##
## sigma^2 = 615.6: log likelihood = -7699.62
## AIC=15403.23   AICc=15403.24   BIC=15414.06
```

## 7.6 Predicción

Se generó una predicción para los próximos 12 períodos de tiempo utilizando el modelo ARIMA ajustado. Las predicciones puntuales junto con los intervalos de confianza del 80% y 95% fueron los siguientes:

Las predicciones sugieren una cantidad de ventas estable, centrada alrededor de 34.0451, con una variabilidad moderada representada por los intervalos de confianza.

```
# Forecast the next 12 time periods
forecast_result <- forecast(model, h = 12)
print(forecast_result)
```

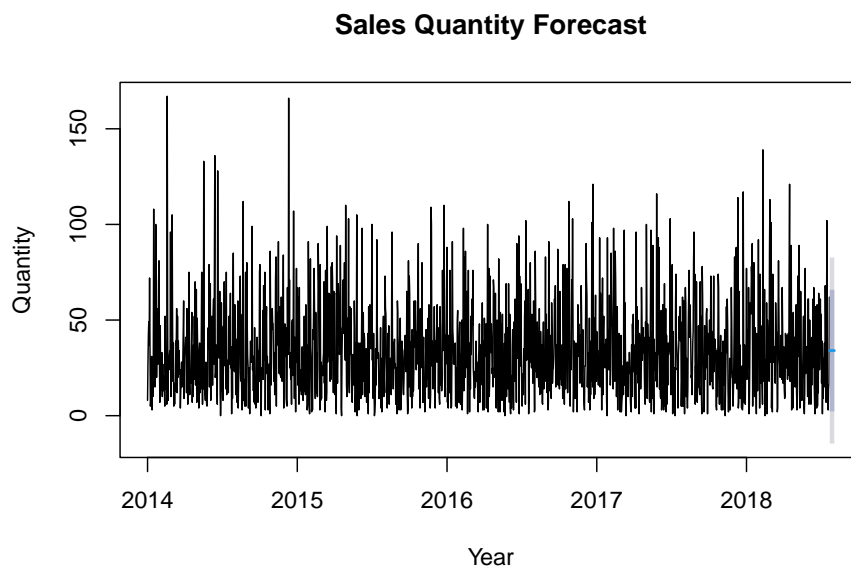
	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
2018.5562	34.0451	2.247558	65.84264	-14.58504	82.67524
2018.5589	34.0451	2.247558	65.84264	-14.58504	82.67524
2018.5616	34.0451	2.247558	65.84264	-14.58504	82.67524

```
## 2018.5644      34.0451  2.247558  65.84264 -14.58504  82.67524
## 2018.5671      34.0451  2.247558  65.84264 -14.58504  82.67524
## 2018.5699      34.0451  2.247558  65.84264 -14.58504  82.67524
## 2018.5726      34.0451  2.247558  65.84264 -14.58504  82.67524
## 2018.5753      34.0451  2.247558  65.84264 -14.58504  82.67524
## 2018.5781      34.0451  2.247558  65.84264 -14.58504  82.67524
## 2018.5808      34.0451  2.247558  65.84264 -14.58504  82.67524
## 2018.5836      34.0451  2.247558  65.84264 -14.58504  82.67524
## 2018.5863      34.0451  2.247558  65.84264 -14.58504  82.67524
```

## 7.7 Detección de Puntos de Cambio

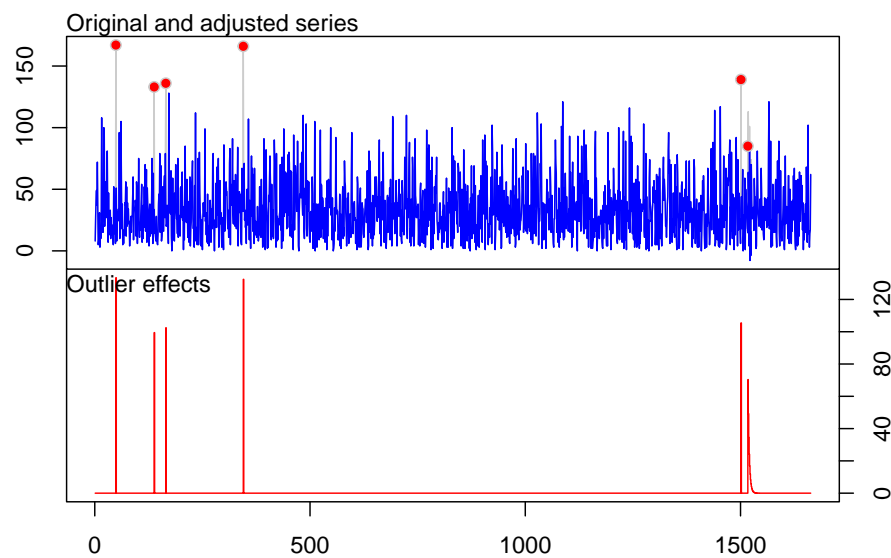
Se realizó un análisis de puntos de cambio utilizando la función `cpt.mean` del paquete `changept`. El análisis tuvo como objetivo identificar cambios significativos en el nivel medio de la cantidad de ventas a lo largo del tiempo. El resultado mostró un punto de cambio detectado, lo cual sugiere un cambio notable en la cantidad promedio de ventas en un momento particular del tiempo. Este tipo de análisis es útil para entender cambios abruptos, como los causados por factores externos como promociones o cambios en el mercado.

```
# Plot the forecast
plot(forecast_result, main="Sales Quantity Forecast", ylab="Quantity", xlab="Year")
```

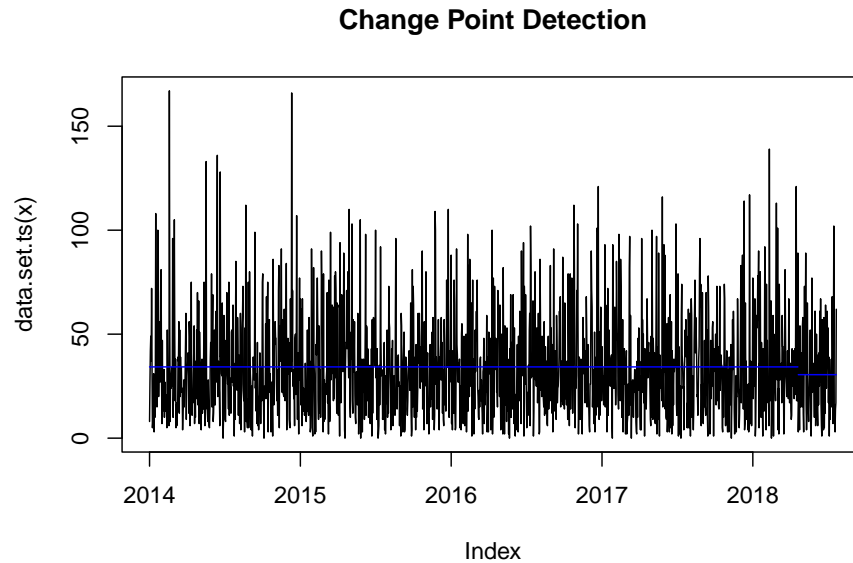


```
# Detect outliers in the time series
dat.ts <- ts(quantity.ts, frequency=1)
```

```
data.ts.outliers <- tso(dat.ts)
plot(data.ts.outliers)
```



```
# Detect change points in the time series
mval <- cpt.mean(quantity.ts, method = "AMOC")
plot(mval, type = "l", cpt.col = "blue", xlab = "Index", main = "Change Point Detection")
```



## 7.8 Conclusión

El análisis de la serie temporal de los datos de cantidad de ventas reveló las siguientes conclusiones clave:

- 1. Estacionariedad: La serie se encontró estacionaria, lo que la hace adecuada para el modelado ARIMA.
- 2. Modelo ARIMA: Se ajustó un modelo ARIMA(0,0,0) con media distinta de cero a los datos, lo cual sugiere un enfoque simple de predicción basado en el nivel medio.
- 3. Predicción: La predicción indicó una cantidad de ventas esperada consistente, aunque los intervalos de confianza indicaron cierta variabilidad.
- 4. Valores Atípicos y Puntos de Cambio: Se detectaron valores atípicos y puntos de cambio, lo cual destaca períodos con desviaciones significativas o cambios en los patrones de ventas. Estos hallazgos proporcionan una base para un análisis adicional, como la identificación de las causas de los valores atípicos o los puntos de cambio detectados y el uso de la predicción para la toma de decisiones en la gestión de inventarios y la cadena de suministro. También se puede considerar una mayor refinación del modelo para mejorar la precisión de las predicciones.

## Chapter 8

# Redes Neuronales de Elman y Jordan

### 8.1 Introducción

Este informe presenta los resultados de la aplicación de redes neuronales recurrentes de Elman y Jordan para la predicción de una serie temporal que representa la cantidad de artículos vendidos diariamente. Los datos fueron procesados y preparados, incluyendo la normalización y división en conjuntos de entrenamiento y prueba. Ambos modelos fueron entrenados y evaluados utilizando métricas como el RMSE, MAE y  $R^2$ .

### 8.2 Preparación de Datos:

Se cargó un conjunto de datos desde un archivo Excel, se realizó la conversión de la variable “cantidad” a una serie temporal y la fecha de transacción se formateó a tipo Date. Para trabajar con una serie temporal regular, se completaron los datos faltantes con el valor cero para los días sin registros. Se dividieron los datos en entrenamiento (80%) y prueba (20%). Finalmente, los datos se normalizaron utilizando una función min-max para mejorar el rendimiento de las redes neuronales.

### 8.3 Modelo de Elman:

Se implementó una red neuronal de Elman con una capa oculta de 5 neuronas, una tasa de aprendizaje de 0.1 y se entrenó durante 1000 iteraciones. Después del entrenamiento, se realizaron predicciones sobre el conjunto de prueba y se desnormalizaron los resultados para compararlos con los valores reales.

## 8.4 Modelo de Jordan:

Similar al modelo de Elman, se implementó una red neuronal de Jordan con la misma configuración de parámetros (5 neuronas en la capa oculta, tasa de aprendizaje de 0.1 y 1000 iteraciones). Las predicciones fueron generadas, desnormalizadas y comparadas con los valores reales.

## 8.5 Resultados:

Se presentan los resultados de las métricas de evaluación para ambos modelos:

Elman:

RMSE: 26.19

MAE: 21.38

$R^2$ : -0.007

Jordan:

RMSE: 23.80

MAE: 19.44

$R^2$ : 0.168

## 8.6 Análisis de Resultados:

Observando las métricas, se aprecia que el modelo de Jordan presenta un mejor rendimiento en comparación con el modelo de Elman, evidenciado por un RMSE y MAE más bajos. El  $R^2$  del modelo de Jordan, aunque bajo, es positivo e indica un ajuste ligeramente mejor a los datos que el modelo de Elman, cuyo  $R^2$  negativo sugiere que el modelo tiene un rendimiento peor que una línea horizontal simple. Los gráficos generados muestran visualmente las predicciones de ambos modelos en comparación con los valores reales. La gráfica de residuos muestra la diferencia entre las predicciones y los valores reales a lo largo del tiempo.

## 8.7 Conclusión:

El modelo de Jordan demostró un rendimiento superior al de Elman para este conjunto de datos específico y la configuración utilizada. Sin embargo, el bajo valor de  $R^2$  para ambos modelos indica que ninguna de las redes logra capturar completamente la complejidad de la serie temporal. Esto se debe a que es uno de los datasets más complicados y uno de los productos con venta más errática de la compañía.

```
# Cargar el paquete
```

```
library(readxl)
```

```
# Leer el archivo Excel
```

```
data <- read_excel("MUESTRA SERIE TIEMPO.xlsx")
```

```
# Verificar los datos cargados
```

```
head(data)
```

```
## # A tibble: 6 x 18
```

```
##   transaction_item_id Transaction_id transaction_date      Time Sucursal_id
```

```
##           <dbl>           <dbl> <dtm>           <dbl>       <dbl>
```

```
## 1             51             34 2014-01-01 00:00:00 0.648         0
```

```
## 2             80             55 2014-01-02 00:00:00 0.434         0
```

```
## 3            110             78 2014-01-02 00:00:00 0.489         2
```

```
## 4            142            103 2014-01-03 00:00:00 0.682         0
```

```
## 5            146            110 2014-01-03 00:00:00 0.655         0
```

```
## 6            168            124 2014-01-04 00:00:00 0.578         2
```

```
## # i 13 more variables: sucursal_name <chr>, Product_id <dbl>,
```

```
## #   product_name <chr>, category <chr>, brand <chr>, quantity <dbl>,
```

```
## #   cost <dbl>, discount <dbl>, Promotion <chr>, id_cliente <dbl>,
```

```
## #   client_name <chr>, Type <chr>, location <chr>
```

```
str(data)
```

```
## tibble [2,679 x 18] (S3: tbl_df/tbl/data.frame)
```

```
## $ transaction_item_id: num [1:2679] 51 80 110 142 146 168 206 237 278 314 ...
```

```
## $ Transaction_id      : num [1:2679] 34 55 78 103 110 124 151 165 189 213 ...
```

```
## $ transaction_date    : POSIXct[1:2679], format: "2014-01-01" "2014-01-02" ...
```

```
## $ Time               : num [1:2679] 0.648 0.434 0.489 0.682 0.655 0.578 0.401 0.465 0.369 0.6...
```

```
## $ Sucursal_id        : num [1:2679] 0 0 2 0 0 2 0 1 0 0 ...
```

```
## $ sucursal_name      : chr [1:2679] "Guacari" "Guacari" "Tulúa" "Guacari" ...
```

```
## $ Product_id         : num [1:2679] 14 14 14 14 14 14 14 14 14 14 ...
```

```
## $ product_name       : chr [1:2679] "Producto 14" "Producto 14" "Producto 14" "Producto 14" ...
```

```
## $ category           : chr [1:2679] "Accesorios" "Accesorios" "Accesorios" "Accesorios" ...
```

```
## $ brand              : chr [1:2679] "Panasonic" "Panasonic" "Panasonic" "Panasonic" ...
```

```
## $ quantity           : num [1:2679] 8 14 13 34 8 24 25 37 23 49 ...
```

```
## $ cost               : num [1:2679] 39000 39000 39000 39000 39000 39000 39000 39000 39000 390...
```

```
## $ discount           : num [1:2679] 0.02 0.02 0.03 0.03 0.02 0.06 0.02 0 0.06 0.06 ...
```

```
## $ Promotion          : chr [1:2679] "N" "N" "N" "N" ...
```

```
## $ id_cliente         : num [1:2679] 1 54 81 1 8 2 1 40 1 58 ...
```

```
## $ client_name        : chr [1:2679] "Clientes Varios" "Cliente_53" "Cliente_80" "Clientes Var...
```

```
## $ Type               : chr [1:2679] "Natural" "Natural" "Natural" "Natural" ...
```

```
## $ location           : chr [1:2679] "Guacari" "Tulua" "Guacari" "Guacari" ...
```

```

#convertir la variable "quantity" en una serie de tiempo diaria

data$transaction_date <- as.Date(data$transaction_date, format = "%Y-%m-%d") # Ajusta

#agrupar los datos por dia

library(dplyr)
library(tidyr)

# Crear una serie diaria completa

daily_data <- data %>%
  group_by(transaction_date) %>%
  summarise(quantity = sum(quantity, na.rm = TRUE)) %>%
  complete(transaction_date = seq(min(transaction_date), max(transaction_date), by = "d",
    fill = list(quantity = 0))

# Crear la serie temporal

quantity_ts <- ts(daily_data$quantity, start = c(as.numeric(format(min(daily_data$transaction_date), "%Y-%m-%d")),
  as.numeric(format(max(daily_data$transaction_date), "%Y-%m-%d")),
  frequency = 365)

#dividir los datos en entrenamiento y prueba

train_size <- floor(0.8 * length(quantity_ts))
train <- quantity_ts[1:train_size]
test <- quantity_ts[(train_size + 1):length(quantity_ts)]

#Normalizar los datos

normalize <- function(x) (x - min(x)) / (max(x) - min(x))
denormalize <- function(x, min_val, max_val) x * (max_val - min_val) + min_val

train_norm <- normalize(train)
test_norm <- normalize(test)

#ELMAN MODEL

library(RSNNS)

# Preparar datos para el modelo Elman

```



```

inputs <- embed(train_norm, 2)[, 2, drop = FALSE]
outputs <- embed(train_norm, 2)[, 1]

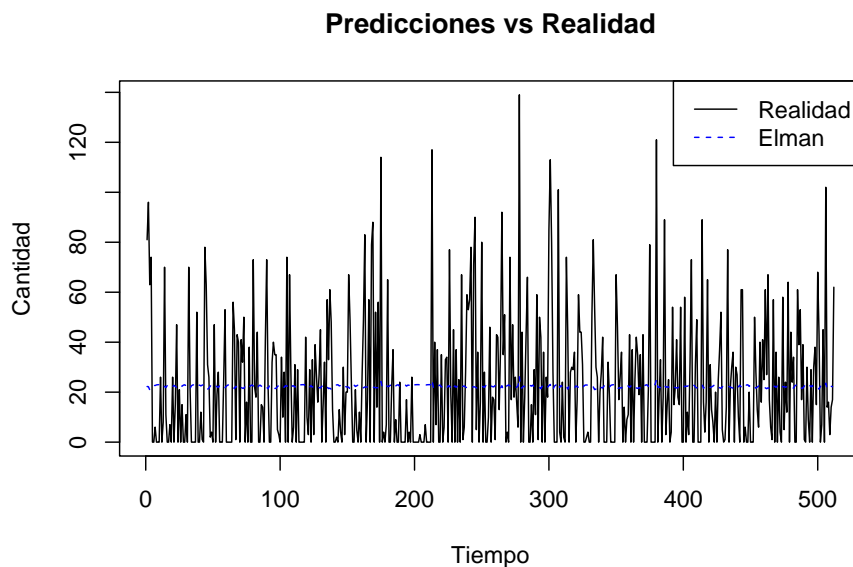
# Entrenar el modelo
elman_model <- elman(
  inputs,
  outputs,
  size = c(5),           # Número de neuronas en la capa oculta
  learnFuncParams = c(0.1), # Tasa de aprendizaje
  maxit = 1000           # Iteraciones máximas
)

# Predecir en el conjunto de prueba
inputs_test <- embed(test_norm, 2)[, 2, drop = FALSE]
predictions_elman <- predict(elman_model, inputs_test)

# Desnormalizar las predicciones
predictions_elman <- denormalize(predictions_elman, min(train), max(train))

# Grafica de predicción vrs realidad
# Graficar predicciones frente a los valores reales
plot(test, type = "l", col = "black", main = "Predicciones vs Realidad", xlab = "Tiempo", ylab = "Cantidad",
lines(predictions_elman, col = "blue", lty = 2) # Predicciones del modelo Elman
legend("topright", legend = c("Realidad", "Elman"), col = c("black", "blue"), lty = 1:2)

```



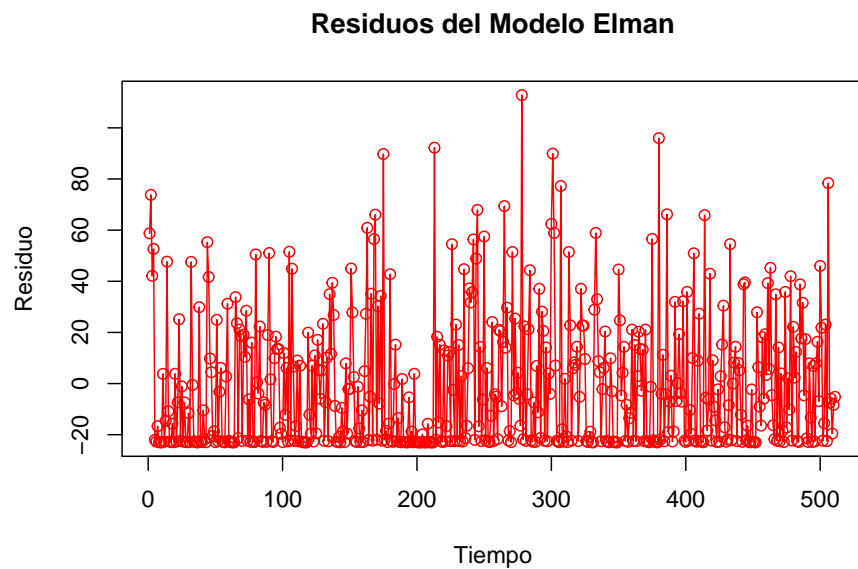
```

# Grafica de residuos
# Asegurarte de que 'predictions_elman' tenga el mismo tamaño que 'test'
predictions_elman <- predictions_elman[1:length(test)]

# Calcular los residuos
residuals <- test - predictions_elman

# Graficar los residuos
plot(residuals, type = "o", col = "red", main = "Residuos del Modelo Elman", xlab = "T")

```



```

# Metricas
# Calcular el RMSE
rmse <- function(real, predicted) sqrt(mean((real - predicted)^2))
rmse_value <- rmse(test, predictions_elman)
print(paste("RMSE:", rmse_value))

## [1] "RMSE: NA"

# Verificar valores NA en 'test' y 'predictions_elman'
sum(is.na(test)) # Ver cuántos valores faltan en el conjunto de prueba

## [1] 0

sum(is.na(predictions_elman)) # Ver cuántos valores faltan en las predicciones

## [1] 1

```

```

# Eliminar valores NA de 'test' y 'predictions_elman'
valid_indices <- complete.cases(test, predictions_elman)
test <- test[valid_indices]
predictions_elman <- predictions_elman[valid_indices]

# Recalcular el RMSE
rmse_value <- sqrt(mean((test - predictions_elman)^2))
print(paste("RMSE:", rmse_value))

## [1] "RMSE: 26.2415983785737"

# Calcular el MAE
mae <- function(real, predicted) mean(abs(real - predicted))
mae_value <- mae(test, predictions_elman)
print(paste("MAE:", mae_value))

## [1] "MAE: 21.3693898320927"

# Calcular el R²
r2_value <- 1 - sum((test - predictions_elman)^2) / sum((test - mean(test))^2)
print(paste("R²:", r2_value))

## [1] "R²: -0.0113139455683893"

##JORDAN

# Entrenar el modelo Jordan
jordan_model <- jordan(
  inputs,
  outputs,
  size = c(5),           # Número de neuronas en la capa oculta
  learnFuncParams = c(0.1), # Tasa de aprendizaje
  maxit = 1000           # Iteraciones máximas
)

# Predecir en el conjunto de prueba
predictions_jordan <- predict(jordan_model, inputs_test)

# Desnormalizar las predicciones
predictions_jordan <- denormalize(predictions_jordan, min(train), max(train))

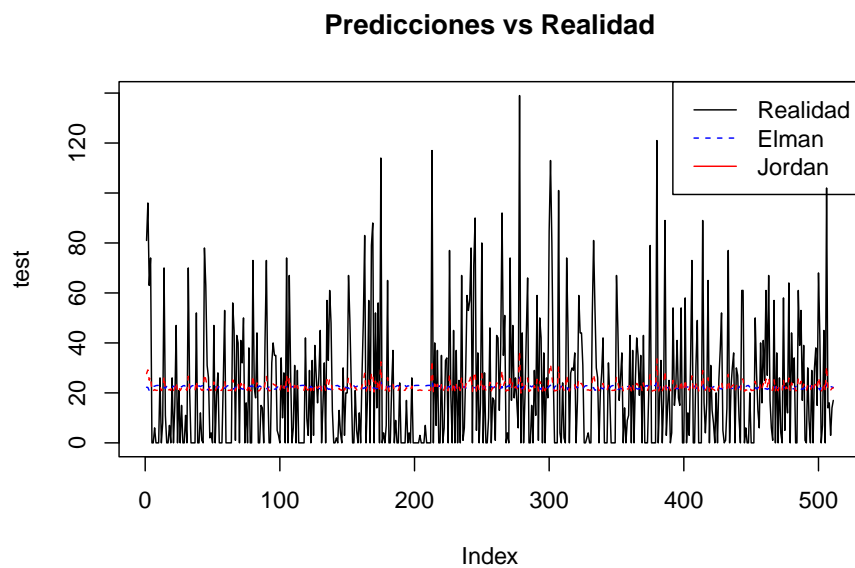
# Calcular RMSE
rmse <- function(real, predicted) sqrt(mean((real - predicted)^2))

rmse_elman <- rmse(test, predictions_elman)
rmse_jordan <- rmse(test, predictions_jordan)

```

```
# Graficar resultados
```

```
plot(test, type = "l", col = "black", main = "Predicciones vs Realidad")
lines(predictions_elman, col = "blue", lty = 2) # Elman
lines(predictions_jordan, col = "red", lty = 2) # Jordan
legend("topright", legend = c("Realidad", "Elman", "Jordan"),
      col = c("black", "blue", "red"), lty = 1:2)
```



```
#Comparación de métricas
```

```
# Cálculo de RMSE para Elman
```

```
rmse_elman <- sqrt(mean((test - predictions_elman)^2))
```

```
# Cálculo de RMSE para Jordan
```

```
rmse_jordan <- sqrt(mean((test - predictions_jordan)^2))
```

```
# Imprimir los resultados
```

```
print(paste("RMSE Elman:", rmse_elman))
```

```
## [1] "RMSE Elman: 26.2415983785737"
```

```
print(paste("RMSE Jordan:", rmse_jordan))
```

```
## [1] "RMSE Jordan: 23.9057614026529"
```

```

# Cálculo de MAE para Elman
mae_elman <- mean(abs(test - predictions_elman))

# Cálculo de MAE para Jordan
mae_jordan <- mean(abs(test - predictions_jordan))

# Imprimir los resultados
print(paste("MAE Elman:", mae_elman))

## [1] "MAE Elman: 21.3693898320927"
print(paste("MAE Jordan:", mae_jordan))

## [1] "MAE Jordan: 19.5052141987093"

# Cálculo de R² para Elman
r_squared_elman <- 1 - sum((test - predictions_elman)^2) / sum((test - mean(test))^2)

# Cálculo de R² para Jordan
r_squared_jordan <- 1 - sum((test - predictions_jordan)^2) / sum((test - mean(test))^2)

# Imprimir los resultados
print(paste("R² Elman:", r_squared_elman))

## [1] "R² Elman: -0.0113139455683893"
print(paste("R² Jordan:", r_squared_jordan))

## [1] "R² Jordan: 0.160712821963783"

# Graficar las predicciones de ambos modelos
plot(test, type = "l", col = "black", main = "Comparación de Predicciones: Elman vs. Jordan",
      xlab = "Tiempo", ylab = "Cantidad")
lines(predictions_elman, col = "blue", lty = 2)
lines(predictions_jordan, col = "red", lty = 3)
legend("topright", legend = c("Realidad", "Predicciones Elman", "Predicciones Jordan"),
      col = c("black", "blue", "red"), lty = 1:3)

```

**Comparación de Predicciones: Elman vs. Jordan**