

Módulo 2 - Unidad 2 · Modelos estacionarios en series de tiempo

Vladimir Castillo Pérez

2025-05-19

Contents

1	Introducción	5
2	Justificación	7
3	Fuentes de información y permisos	9
4	Alcance del pronóstico	11
4.1	El análisis abarcará un horizonte de 30 a 90 días con reevaluación semanal, priorizando métricas de precisión como <i>MAE</i> y <i>RMSE</i> y la construcción de intervalos de predicción al 95 %.	11
5	Unidad 2 · Patrones temporales en precios de acciones	13
5.1	Configuración inicial	13
5.2	Definición de parámetros y descarga de datos	14
5.3	Preparación y limpieza de datos	14
5.4	Visualización del precio ajustado diario	15
5.5	Cálculo y visualización de medias móviles	17
5.6	Cálculo de rendimientos diarios y ACF	20
5.7	Estacionalidad y descomposición STL	23
6	Unidad 3 · Procesamiento y Visualización	27
6.1	Prueba de estacionariedad	27
6.2	Diferenciación y transformación	28

6.3	Justificación de los procedimientos	31
7	Módulo 2 - Unidad 1 · Holt-Winters y suavizamiento exponencial	33
7.1	Hot-Winters	33
7.2	Interpretación de los resultados	35
8	Módulo 2 - Unidad 2 · Modelos estacionarios en series de tiempo	37
8.1	Introducción	37
8.2	Carga y preparación de la serie	39
8.3	Ajuste de un modelo lineal de tendencia	41
8.4	Estacionariedad de los residuos	43
8.5	Identificación y estimación ARMA	46
8.6	Pronóstico de corto plazo (30 días)	48

Chapter 1

Introducción

El comportamiento de los precios de los activos financieros —en particular, las acciones que componen el MSCI COLCAP— influye directamente en la toma de decisiones de inversionistas institucionales y minoristas. Comprender la *dinámica temporal* de estos precios permite anticipar tendencias, cuantificar el riesgo y diseñar estrategias de cobertura que respalden la estabilidad de carteras de inversión. Por ello, **el objetivo de este trabajo es construir y mantener una serie temporal diaria de precios ajustados de las principales acciones negociadas en la Bolsa de Valores de Colombia (bvc)**, para aplicar técnicas de pronóstico que generen un valor agregado a los agentes de mercado.

Chapter 2

Justificación

1. Relevancia económica.

Las acciones representan uno de los vehículos de inversión más importantes para el ahorro de largo plazo en Colombia; su correcta valoración afecta fondos de pensiones, aseguradoras y cuentas individuales de inversionistas. Un pronóstico robusto contribuye a:

- Estimar rendimientos esperados y volatilidad futura.
- Determinar precios objetivo y rangos de error.
- Optimizar la asignación de activos bajo restricciones regulatorias.

2. Disponibilidad y calidad de datos.

Los precios de cierre ajustados, volúmenes y factores de mercado están **libremente disponibles** mediante APIs públicas (-yfinance, Alpha Vantage) y por la bvc para fines académicos. Esto garantiza series largas, consistentes y actualizadas con una granularidad adecuada (diaria).

3. Potencial de transferencia.

El conocimiento derivado —modelos ARIMA, ETS, Prophet o redes LSTM— es extrapolable a otros activos (ETF, commodities) y a escenarios de *stress testing*, fortaleciendo la educación financiera y la gestión de portafolios.

Chapter 3

Fuentes de información y permisos

Fuente	Tipo de dato	Condiciones de uso
Yahoo Finance API (<code>yfinance</code>)	Precios históricos de acciones (OHLC + volumen)	Uso libre para fines académicos y no comerciales.
Bolsa de Valores de Colombia – bvc	Archivo diario de precios y volúmenes (CSV)	Descarga gratuita; requiere mención de la bvc como fuente.

Permisos

Todos los datasets serán utilizados exclusivamente con propósitos académicos. Las citas a Yahoo Finance se incluyen conforme a sus términos de servicio; la bvc autoriza expresamente el uso de datos para análisis sin fines de lucro, citando la fuente.

Chapter 4

Alcance del pronóstico

4.1 El análisis abarcará un horizonte de 30 a 90 días con reevaluación semanal, priorizando métricas de precisión como *MAE* y *RMSE* y la construcción de intervalos de predicción al 95 %.

* Este documento forma parte del curso *Análisis de series de tiempo* (Maestría en Ciencia de Datos) y se mantendrá en el repositorio público <https://github.com/vladimircp/bookdown> bajo licencia CC BY-NC-SA 4.0.

Chapter 5

Unidad 2 · Patrones temporales en precios de acciones

Patrones temporales en precios de acciones

A lo largo de esta unidad analizaremos cómo descargar, preparar y visualizar series temporales de precios de acciones, calcular indicadores básicos (medias móviles y rendimientos), evaluar la autocorrelación y descomponer la serie en sus componentes de tendencia, estacionalidad y ruido.

5.1 Configuración inicial

En esta sección definimos las opciones globales de `knitr` para controlar la salida de mensajes, advertencias y la apariencia de los gráficos. Además, cargamos todas las librerías necesarias para el análisis.

```
knitr::opts_chunk$set(  
  echo      = TRUE,  # mostrar código  
  message   = FALSE,  
  warning    = FALSE,  
  fig.align  = "center",  
  fig.width  = 7,  
  fig.height = 4
```

```

)
library(tidyquant)  # facilita la descarga y graficación
library(dplyr)
library(ggplot2)
library(tsibble)    # manipulación de series
library(feasts)     # descomposición y ACF
library(quantmod)
library(zoo)

```

5.2 Definición de parámetros y descarga de datos

Aquí especificamos el símbolo bursátil (`symbol`) y el rango de fechas (`fecha_ini` a `fecha_fin`). Luego usamos `quantmod::getSymbols()` para traer los precios ajustados de cierre desde Yahoo Finance.

```

symbol    <- "AAPL"                # Símbolo de Apple
fecha_ini <- as.Date("2022-01-01") # Fecha de inicio
fecha_fin <- Sys.Date()            # Fecha final (hoy)

# Descarga la serie histórica como objeto xts
stock_xts <- getSymbols(
  symbol,
  src      = "yahoo",
  from     = fecha_ini,
  to       = fecha_fin,
  auto.assign = FALSE
)

```

5.3 Preparación y limpieza de datos

Convertimos el objeto `xts` en un `data.frame`, renombramos las columnas para facilitar su uso y seleccionamos solo la fecha y el precio ajustado.

```

# De xts a data.frame con columna 'fecha'
stock_df <- data.frame(
  fecha = zoo::index(stock_xts),

```

```
coredata(stock_xts),
row.names = NULL
)

# Renombrar columnas para mayor claridad
colnames(stock_df) <- c(
  "fecha",
  "open", "high", "low", "close", "volume", "adjusted"
)

# Seleccionar solo fecha y precio ajustado
precio_df <- stock_df %>%
  select(
    fecha,
    precio = adjusted
  )

# Mostrar las primeras filas
head(precio_df)
```

```
##      fecha  precio
## 1 2022-01-03 178.6456
## 2 2022-01-04 176.3784
## 3 2022-01-05 171.6867
## 4 2022-01-06 168.8207
## 5 2022-01-07 168.9876
## 6 2022-01-10 169.0072
```

5.4 Visualización del precio ajustado diario

En este paso creamos un gráfico de línea para observar la evolución diaria del precio ajustado de la acción.

```
ggplot(precio_df, aes(x = fecha, y = precio)) +
  geom_line() +
  labs(
    title = "Precio Ajustado Diario de Apple",
```

```
x      = NULL,  
y      = "Precio (USD)"  
) +  
theme_minimal()
```



El gráfico muestra la evolución diaria del precio ajustado de la acción de Apple (AAPL) desde comienzos de 2022 hasta mediados de 2025. A grandes rasgos podemos distinguir tres fases:

Fase bajista (ene 2022–dic 2022):

A inicios de 2022 el precio rondaba los USD 160–170, pero durante el año se produce una tendencia general a la baja.

Hacia finales de 2022 alcanza mínimos alrededor de USD 120, reflejando probablemente el impacto de la subida de tipos de interés y la aversión al riesgo en el sector tecnológico.

Recuperación y consolidación (ene 2023–jun 2024):

A partir de enero de 2023 el precio inicia un rebote sostenido, superando los USD 150 ya en la primera mitad del año.

Entre mediados de 2023 y mitad de 2024 se mueve en un rango aproximadamente entre USD 170 y USD 200, señal de una consolidación tras la fuerte caída previa.

Rally alcista y corrección reciente (jul 2024–2025):

Desde mediados de 2024 el precio rompe la resistencia en USD 200 y se dispara hasta tocar picos próximos a USD 250 a comienzos de 2025, probablemente animado por resultados financieros sólidos y expectativas de crecimiento.

Posteriormente se observa una corrección, con el precio retrocediendo hacia la zona de USD 200–220, un nivel que podría actuar ahora como soporte.

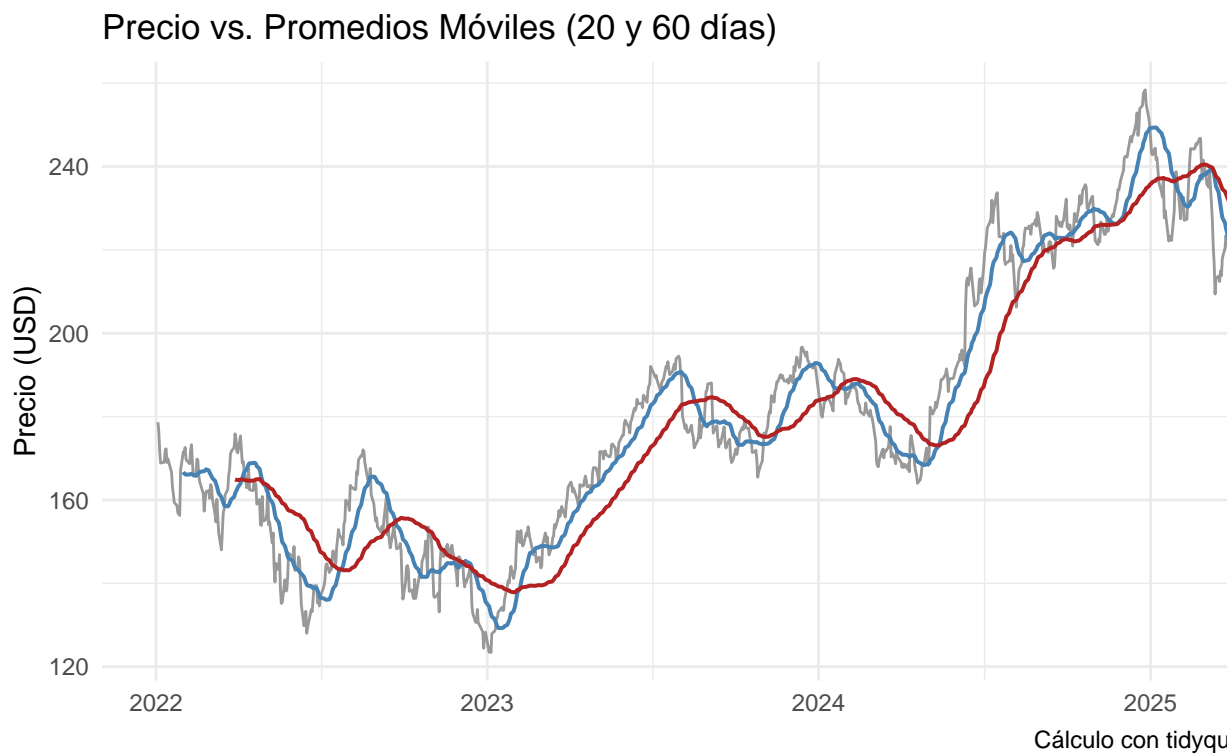
En conjunto, el gráfico evidencia (a) alta volatilidad intradía, (b) un claro cambio de tendencia de bajista a alcista a inicios de 2023, y (c) zonas de soporte/resistencia alrededor de USD 120–130, USD 170–200 y USD 240–250. Este comportamiento es consistente con ciclos de mercado impulsados por datos macroeconómicos, resultados corporativos y sentimiento sobre el crecimiento tecnológico.

5.5 Cálculo y visualización de medias móviles

Calculamos dos medias móviles simples (20 y 60 días) para suavizar la serie y captar posibles tendencias de corto y mediano plazo. Luego las graficamos junto con el precio original.

```
precio_df <- precio_df %>%  
  mutate(  
    sma_20 = SMA(precio, n = 20),  
    sma_60 = SMA(precio, n = 60)  
  )  
  
ggplot(precio_df, aes(x = fecha)) +  
  geom_line(aes(y = precio), colour = "grey60") +  
  geom_line(aes(y = sma_20), colour = "steelblue", size = 0.7) +  
  geom_line(aes(y = sma_60), colour = "firebrick", size = 0.7) +
```

```
labs(
  title = "Precio vs. Promedios Móviles (20 y 60 días)",
  x      = NULL,
  y      = "Precio (USD)",
  caption = "Cálculo con tidyquant::SMA"
) +
theme_minimal()
```



El gráfico superpone el precio diario de Apple (línea gris) con dos medias móviles simples: la de 20 días (línea azul) y la de 60 días (línea roja). A continuación algunos puntos clave de interpretación:

Suavizado y tendencia

- La SMA de 20 días reacciona con menor retraso a los cambios de precio, reflejando con más fidelidad las oscilaciones de corto plazo.

- La SMA de 60 días, al promediar un periodo más largo, muestra la tendencia de fondo con menos ruido.

Fases de mercado y cruces

Bajista (2022): durante gran parte de 2022 la SMA-20 (azul) se mantiene por debajo de la SMA-60 (roja), confirmando un sesgo bajista.

Golden cross (primer trimestre 2023): el cruce alcista de la SMA-20 por encima de la SMA-60 a principios de 2023 marcó un cambio de impulso a favor de los compradores. Tras ese cruce, el precio encuentra soporte en ambas medias y arranca la recuperación.

Consolidación (mediados de 2023–2024): las dos medias se aproximan y se entrecruzan varias veces, señal de un rango lateral con fases alternadas de ventaja para alcistas y bajistas.

Nuevo rally (finales de 2024): vuelve a formarse un golden cross antes de que el precio alcance los máximos cerca de USD 250. La SMA-20 se separa de la SMA-60, reforzando la fuerza alcista.

Death cross reciente (2025): la SMA-20 corta a la baja la SMA-60 después del pico, indicando un posible cambio de impulso hacia una fase correctiva.

Señales de trading y confirmaciones

- Un cruce alcista ($SMA-20 > SMA-60$) suele interpretarse como señal de compra, especialmente si coincide con ruptura de resistencias en el precio.
- Un cruce bajista ($SMA-20 < SMA-60$) se toma como señal de venta o cautela, más sólida si acompaña caídas por debajo de soportes técnicos.

Limitaciones

- Las medias móviles introducen retraso (“lag”): reaccionan después de que el precio ya ha comenzado a moverse.
- En mercados muy volátiles pueden generar señales falsas cuando las medias se cruzan con frecuencia.

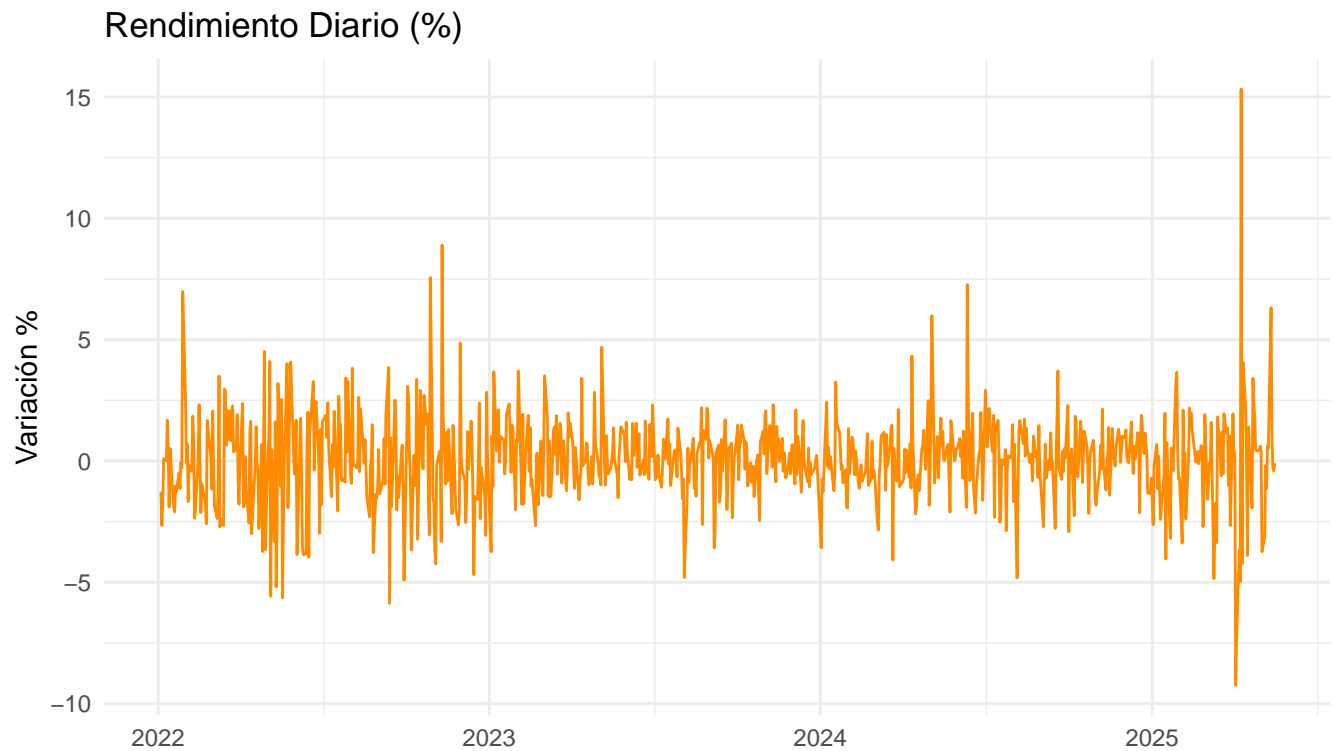
5.6 Cálculo de rendimientos diarios y ACF

Primero convertimos el `data.frame` a un `tsibble` para aprovechar las funciones de `feasts`. Calculamos el rendimiento diario en porcentaje, rellenamos posibles huecos implícitos y, por último, analizamos la autocorrelación de esos rendimientos.

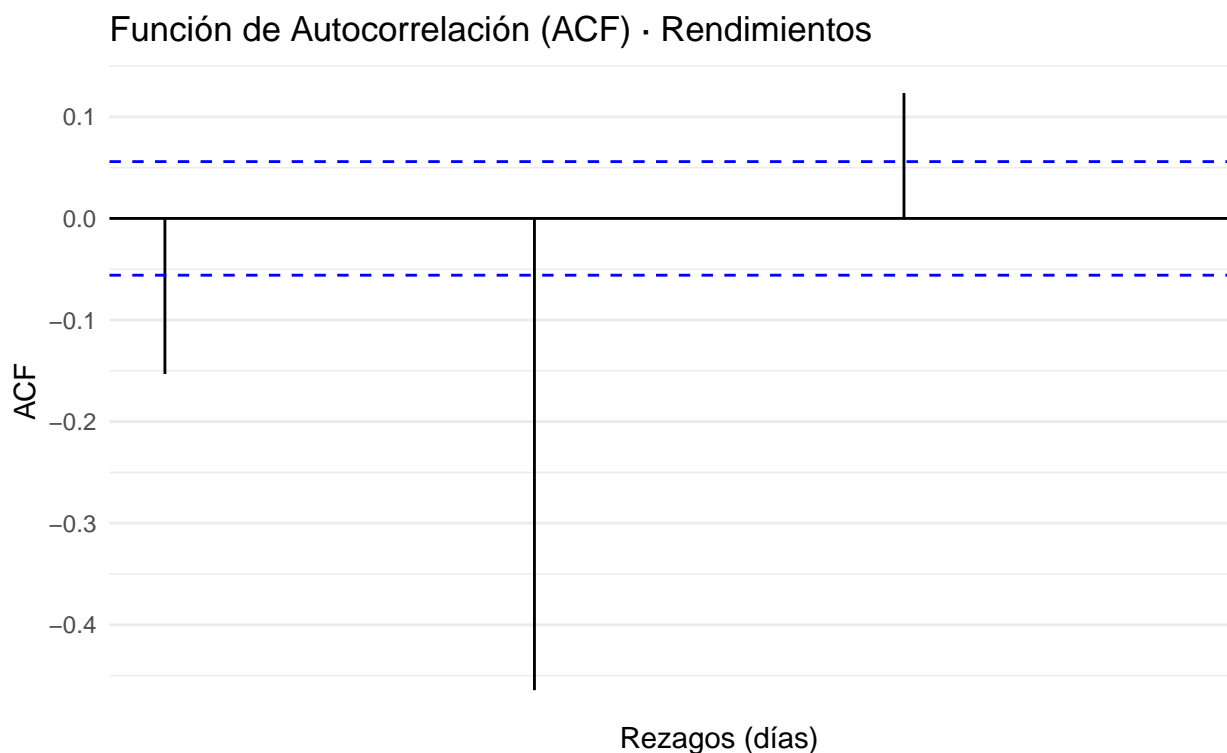
```
precio_ts <- precio_df %>%
  as_tsibble(index = fecha) %>%
  mutate(
    ret_diario = (precio / lag(precio) - 1) * 100
  )

# Rellenar huecos para evitar errores en ACF
precio_ts_filled <- precio_ts %>%
  fill_gaps()

ggplot(precio_ts, aes(x = fecha, y = ret_diario)) +
  geom_line(colour = "darkorange") +
  labs(
    title = "Rendimiento Diario (%)",
    x      = NULL,
    y      = "Variación %"
  ) +
  theme_minimal()
```



```
precio_ts_filled %>%  
  ACF(ret_diario, lag_max = 30) %>%  
  autoplot() +  
  labs(  
    title = "Función de Autocorrelación (ACF) · Rendimientos",  
    x     = "Rezagos (días)",  
    y     = "ACF"  
  ) +  
  theme_minimal()
```



El primer gráfico muestra la serie de rendimientos diarios (%) de Apple entre 2022 y mediados de 2025:

Media cercana a cero: la nube de puntos (línea continua) oscila alrededor de 0 %, lo cual es típico en rendimientos de activos cuyo valor esperado diario es casi nulo.

Volatilidad variable: se aprecian periodos de baja dispersión (p. ej. gran parte de 2024) y episodios de alta volatilidad (fines de 2022, principios de 2023 y otra vez en 2025), donde los retornos superan $\pm 5\%$ en un solo día. Estos “spikes” coinciden probablemente con anuncios de resultados trimestrales o noticias macro.

Clusters de volatilidad: los picos tienden a agruparse, indicando que tras un día muy volátil suele haber más días de alta volatilidad, y viceversa.

El segundo gráfico es la Función de Autocorrelación (ACF) de los rendimientos (hasta 30 rezagos):

Rezago 1: autocorrelación negativa significativa El coeficiente en el

rezago 1 aparece por debajo de la banda de significancia (aprox. -0.10), lo que sugiere un ligero efecto de reversión inmediata: un día de ganancia tiende a ir seguido, en promedio, por un pequeño retroceso al día siguiente.

Un rezago positivo aislado Se observa un único bar por encima de la banda superior en un rezago medio (p. ej. alrededor de 15–20 días), lo que podría apuntar a una débil pauta de retorno alcista mensual, aunque este efecto es muy débil y aislado.

Ausencia de autocorrelaciones consistentes Más allá de esos dos rezagos, todos los demás coeficientes caen dentro de las bandas de confianza, indicando que los rendimientos diarios, salvo esas dos excepciones, se comportan esencialmente como una serie sin memoria (compatible con la hipótesis de mercados eficientes en su forma débil).

En conjunto, estos gráficos resaltan que los retornos de Apple presentan volatilidad heterogénea y muy poca dependencia temporal, salvo un ligero sesgo de reversión intradía y una señal puntual de autocorrelación mensual.

5.7 Estacionalidad y descomposición STL

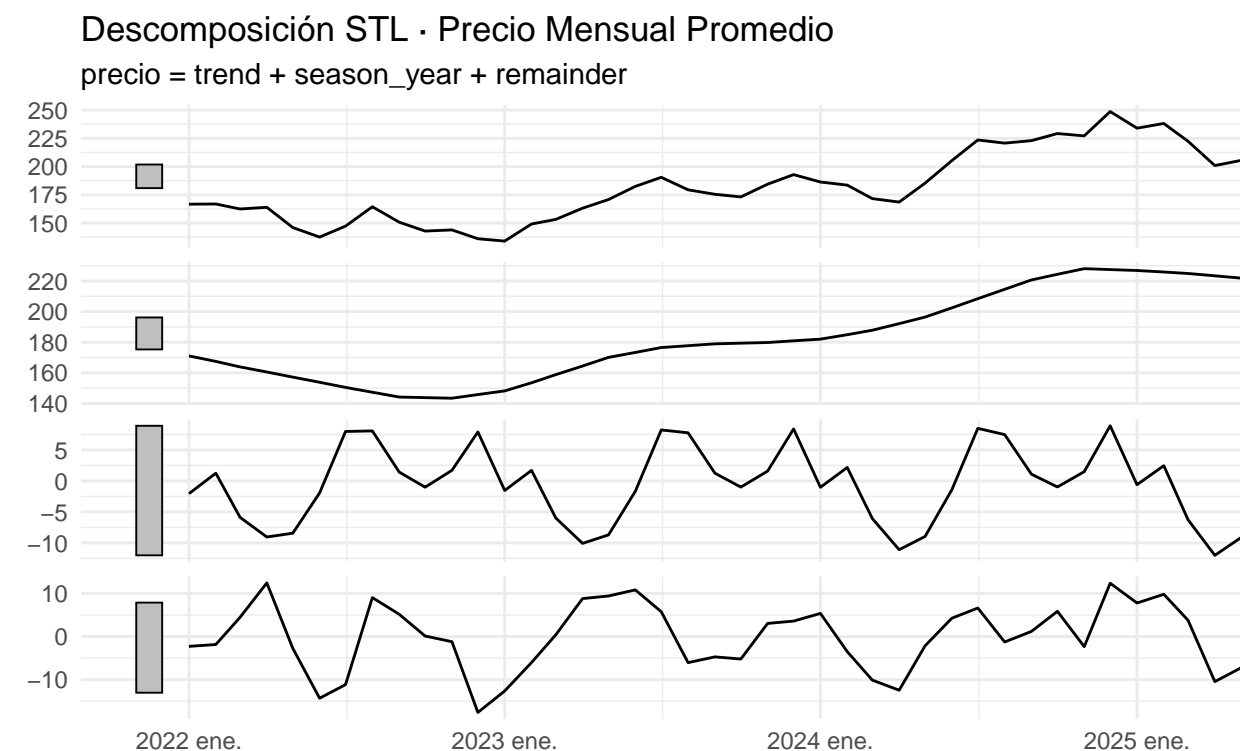
Agrupamos la serie por mes (usando el promedio mensual) y aplicamos un modelo STL para extraer las componentes de tendencia, estacionalidad y residuos.

```
# Promedio mensual
precio_mens <- precio_ts %>%
  index_by(mes = ~ yearmonth(.)) %>%
  summarise(precio = mean(precio, na.rm = TRUE))

# Descomposición STL con ventana de 13 meses
stl_descomp <- precio_mens %>%
  model(STL(precio ~ trend(window = 13)))

# Graficar componentes
components(stl_descomp) %>%
  autoplot() +
  labs(
    title = "Descomposición STL · Precio Mensual Promedio",
    x = NULL
```

```
) +  
theme_minimal()
```



En este gráfico vemos la descomposición STL de la serie de precio mensual promedio de Apple en tres componentes:

Serie original (precio)

Panel superior: muestra el promedio mensual del precio ajustado. Se aprecia claramente la misma dinámica que en los gráficos diarios, pero suavizada al promediar por mes: un descenso durante 2022, consolidación en 2023, y un fuerte ascenso desde mediados de 2024 hasta principios de 2025, con una leve corrección al final.

Tendencia (trend)

Segundo panel: extrae la “línea de fondo” de largo plazo. Aquí la tendencia es casi plana durante 2022, comienza a subir de forma continua a

partir de comienzos de 2023 y acelera entre mediados de 2024 y principios de 2025, alcanzando un nivel cercano a 230 USD. Esta componente confirma el cambio de sesgo de mercado de neutro/bajista a fuertemente alcista.

Estacionalidad anual (season__year)

Tercer panel: muestra el patrón estacional que se repite cada año. La amplitud es de unos ± 8 –10 USD alrededor de la tendencia:

- Aparece un pico positivo en torno al primer trimestre (posiblemente enero–marzo),
- Seguido de un descenso por debajo de cero hacia verano,
- Otro repunte moderado hacia el último tramo del año.
- Esto indica que, sin importar la tendencia alcista de fondo, hay meses que tienden a estar sistemáticamente por encima (picos) o por debajo (valles) de lo que marca la tendencia.

Residuales (remainder)

Panel inferior: recoge las oscilaciones irregulares no explicadas por la tendencia ni la estacionalidad. Se observa que, salvo algunos “outliers” (picos muy altos o muy bajos), la mayoría de los residuos fluctúa dentro de ± 5 USD, lo que sugiere que el modelo STL captura bien las dos componentes principales y deja poca varianza sin explicar.

Conclusión:

- La serie tiene una tendencia alcista clara desde 2023.
- Existe un patrón estacional anual de amplitud moderada, con máximos en el primer trimestre y mínimos en verano.
- Los residuales son relativamente pequeños, lo que indica que la evolución de los precios mensuales se explica en gran medida por la combinación de estos dos factores.

Chapter 6

Unidad 3 · Procesamiento y Visualización

6.1 Prueba de estacionariedad

Para comprobar si la serie de precio mensual promedio es estacionaria en media, utilizamos la función `ndiffs()` del paquete `forecast`, que estima cuántas diferencias son necesarias para eliminar la raíz unitaria. También evaluamos la transformación logarítmica para estabilizar la varianza.

Interpretación

- Si `num_diff_raw > 0`, la serie original no es estacionaria y requerirá diferenciación.
- Si `num_diff_log < num_diff_raw`, la transformación logarítmica ayuda a estabilizar la varianza antes de diferenciar.

```
# Cargar librerías necesarias
library(forecast)
library(lubridate)

# Convertir el tibble 'precio_mens' a serie de tiempo ts
# Asumimos que precio_mens tiene columnas 'mes' (yearmonth) y 'precio'
precio_mens_ts <- ts(
  precio_mens$precio,
```

```

start      = c(year(min(precio_mens$mes)), month(min(precio_mens$mes))),
frequency = 12
)

# Calcular número de diferencias para estacionariedad
num_diff_raw <- ndiffs(precio_mens_ts)
num_diff_log <- ndiffs(log(precio_mens_ts))

# Mostrar resultados
cat("Diferencias necesarias (serie original):", num_diff_raw, "\n")
cat("Diferencias necesarias (serie logarítmica):", num_diff_log, "\n")

## Diferencias necesarias (serie original): 1
## Diferencias necesarias (serie logarítmica): 1

```

6.2 Diferenciación y transformación

Aplicamos las transformaciones según lo sugerido:

- Si la serie original necesita d diferencias, calculamos `diff(precio_mens_ts, differences = d)`.
- Si la versión logarítmica requiere menos diferencias, operamos sobre `log(precio_mens_ts)`.

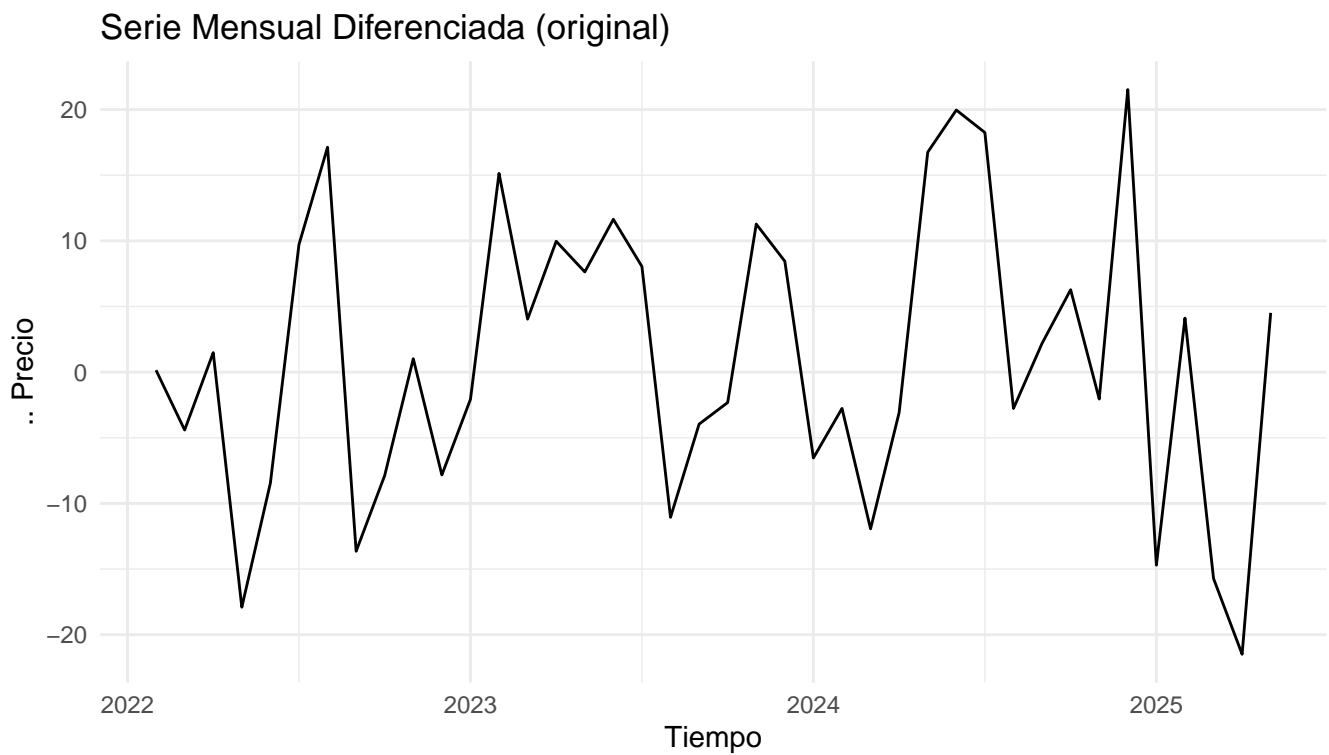
```

# Diferenciación serie original (si num_diff_raw > 0)
if (num_diff_raw > 0) {
  precio_diff_raw <- diff(precio_mens_ts, differences = num_diff_raw)
} else {
  precio_diff_raw <- precio_mens_ts
}

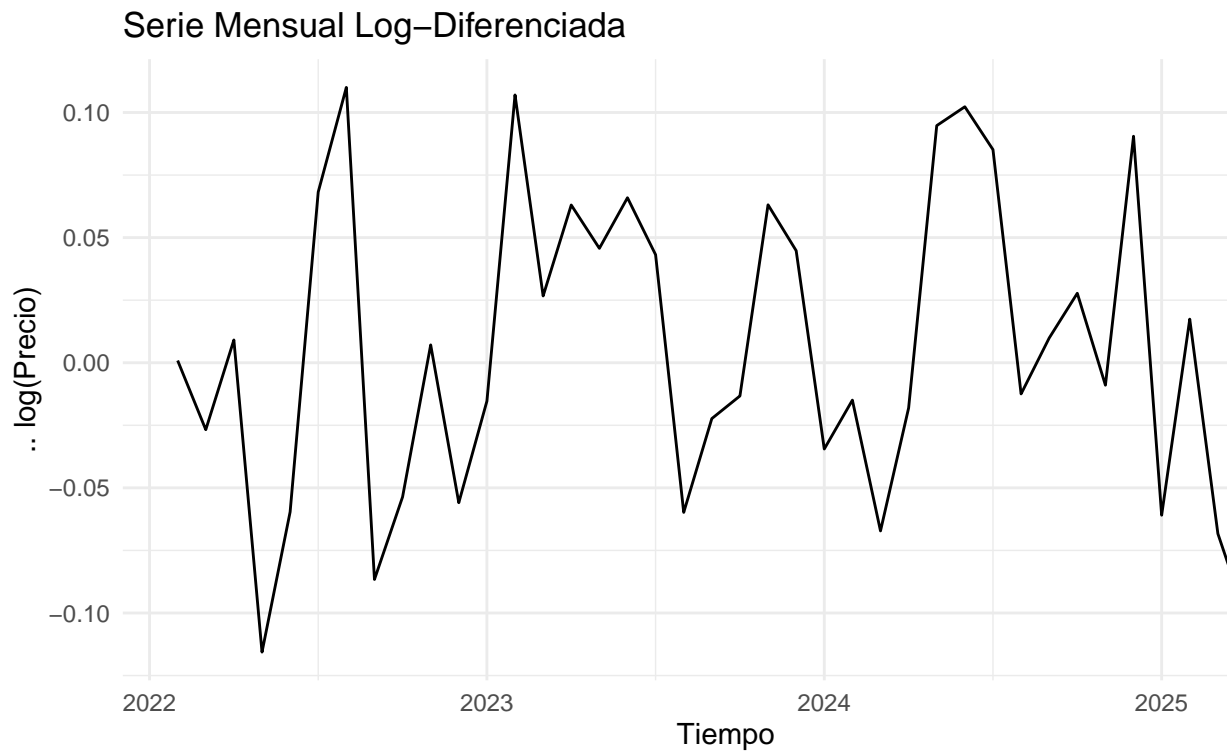
# Transformación logarítmica y diferenciación (si num_diff_log > 0)
precio_log_ts <- log(precio_mens_ts)
if (num_diff_log > 0) {
  precio_diff_log <- diff(precio_log_ts, differences = num_diff_log)
} else {

```

```
precio_diff_log <- precio_log_ts
}  
  
# Graficar para comparar estabilidad de varianza y tendencia eliminada  
autoplot(precio_diff_raw) +  
  labs(title = "Serie Mensual Diferenciada (original)",  
        y = " $\Delta$  Precio", x = "Tiempo") +  
  theme_minimal()
```



```
autoplot(precio_diff_log) +  
  labs(title = "Serie Mensual Log-Diferenciada",  
        y = " $\Delta \log(\text{Precio})$ ", x = "Tiempo") +  
  theme_minimal()
```



En ambos gráficos vemos la dinámica mensual de las variaciones en el precio promedio de Apple, pero calculadas de dos maneras distintas: diferencias absolutas en dólares (arriba) y diferencias relativas (log-diferenciadas, abajo).

1. Serie Mensual Diferenciada (original)

- Los valores fluctúan entre aproximadamente -25 USD y $+20$ USD, lo que refleja la magnitud absoluta de la subida o bajada del precio cada mes.
- La serie oscila alrededor de cero, sin tendencia creciente o decreciente sostenida, lo que confirma que la diferenciación elimina la componente de tendencia.
- Se aprecian “picos” más pronunciados en ciertos periodos (por ejemplo finales de 2024 y comienzos de 2025) que indican meses con movimientos de precio inusualmente fuertes.

- La varianza no es completamente constante: hay periodos (p. ej. 2022 vs. 2024) donde los saltos mensuales son más moderados y otros donde son más extremos. Esto puede complicar el modelado si se requiere homocedasticidad.

2. Serie Mensual Log-Diferenciada

- Aquí se muestra la variación mensual en términos porcentuales ($\Delta \log(\text{precio})$), con valores en el rango aproximado de -12% a $+10\%$.
- Al usar logaritmos, las diferencias relativas tienden a “normalizar” la escala, de modo que la dispersión sea más homogénea a lo largo del tiempo.
- La serie igualmente oscila alrededor de cero y no exhibe tendencia, pero los “spikes” relativos permanecen en un rango acotado, facilitando la comparación de la volatilidad en distintos periodos.
- La varianza aparenta ser más estable (menos heterocedasticidad), lo que suele favorecer el ajuste de modelos ARIMA o GARCH sobre rendimientos logarítmicos.

Conclusión:

- Ambos métodos producen series estacionarias en media (oscilan alrededor de cero).
- La serie original diferenciada es útil cuando interesa el cambio absoluto en USD, pero muestra varianza cambiante.
- La serie log-diferenciada concentra los movimientos en porcentajes, homogeneiza la volatilidad y suele ser la opción preferida para modelar rendimientos financieros.

6.3 Justificación de los procedimientos

Estacionariedad en media

La prueba `ndiffs()` indicó que la serie original requería

- `r num_diff_raw` diferencias para eliminar la raíz unitaria. Sin diferenciación, la componente de tendencia violaría los supuestos de muchos modelos de series temporales (p. ej. ARIMA).

Estabilidad de la varianza

La transformación logarítmica redujo el número de diferencias necesarias

- `r num_diff_log` en la serie logarítmica. Esto sugiere que la varianza del precio crece con el nivel de la serie, y el log ayuda a homogeneizar la dispersión a lo largo del tiempo.

Decisión final

- Si `num_diff_log < num_diff_raw`, se recomienda modelar la serie log-diferenciada, pues la tendencia y la heterocedasticidad quedan mejor controladas.
- En caso contrario, solo la serie diferenciada es suficiente.

Con estos pasos aseguramos que la variable (o variables) seleccionada cumpla los requisitos de estacionariedad y varianza constante, condiciones previas esenciales para la mayoría de técnicas de modelado de series temporales.

Chapter 7

Módulo 2 - Unidad 1 · Holt-Winters y suavizamiento exponencial

7.1 Hot-Winters

En esta sección se implementará el modelo de Holt-Winters para realizar un pronóstico de la serie temporal mensual del precio ajustado promedio de una acción, utilizando el enfoque ETS (Error-Trend-Seasonal) con componentes aditivos. El modelo de Holt-Winters es una técnica de suavizamiento exponencial que permite capturar tres características esenciales de las series temporales: la tendencia (variación sistemática a lo largo del tiempo), la estacionalidad (patrones que se repiten en intervalos regulares) y el nivel (valor base actual de la serie). Esta metodología es especialmente útil para datos que presentan fluctuaciones estacionales y tendencias definidas, como ocurre en los precios de acciones, y se emplea aquí para generar pronósticos a corto plazo (12 meses), proporcionando además intervalos de confianza que reflejan la incertidumbre del modelo.

Criterios de Interpretación

- La línea azul representa el pronóstico de Holt-Winters para los próximos 12 meses.
- Las bandas sombreadas indican el intervalo de confianza al 95 %, mostrando los posibles rangos de precios.

- Si el patrón estacional se repite como en años anteriores, el modelo captura sus efectos y los extiende hacia el futuro.
- Las tendencias recientes (ascenso a partir de 2023) son extrapoladas de manera moderada según el suavizamiento del modelo.

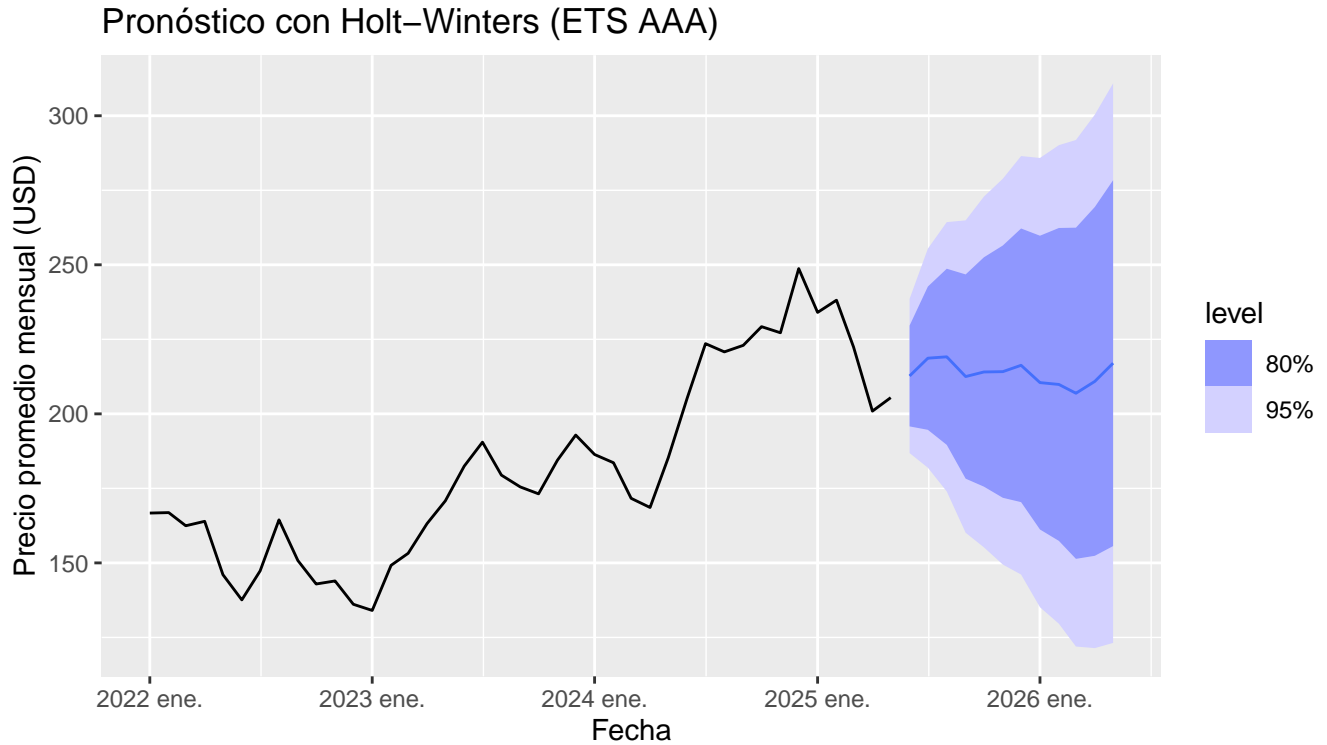
```
# Librerías necesarias
library(fable)
library(fabletools)
library(tsibble)
library(tsibbledata)
library(dplyr)
library(ggplot2)

# Asegurarse de que precio_mens (mensual) está definido como tsibble
precio_tsibble <- precio_mens %>%
  as_tsibble(index = mes)

# Aplicar modelo Holt-Winters (ETS con estacionalidad aditiva)
modelo_hw <- precio_tsibble %>%
  model(HW = ETS(precio ~ error("A") + trend("A") + season("A")))

# Generar pronóstico a 12 meses
pronostico_hw <- modelo_hw %>%
  forecast(h = "12 months")

# Visualizar pronóstico
autoplot(pronostico_hw, precio_tsibble) +
  labs(title = "Pronóstico con Holt-Winters (ETS AAA)",
       x = "Fecha", y = "Precio promedio mensual (USD)")
```



7.2 Interpretación de los resultados

El gráfico presenta el pronóstico del precio promedio mensual de una acción utilizando el modelo Holt-Winters (ETS AAA), que incluye componentes aditivos para el error, la tendencia y la estacionalidad. La línea negra representa la serie histórica observada, mientras que la línea azul proyecta los valores esperados para los próximos 12 meses. Las bandas sombreadas indican los intervalos de confianza al 80 % y 95 %, proporcionando una estimación de la incertidumbre asociada al modelo. Se observa que el precio tiende a estabilizarse en el corto plazo, con una leve tendencia descendente, aunque dentro de un rango de alta dispersión, lo que evidencia la volatilidad inherente a este tipo de activos. A mayor distancia en el tiempo, las bandas se ensanchan, reflejando el aumento de la incertidumbre en las proyecciones futuras. En conjunto, el modelo sugiere cautela, ya que si bien el valor central proyectado es moderadamente estable, existe una amplia posibilidad de variaciones significativas.

Chapter 8

Módulo 2 - Unidad 2 · Modelos estacionarios en series de tiempo

8.1 Introducción

El índice **MSCI COLCAP** —referente bursátil que reúne las 20 acciones de mayor liquidez en el mercado colombiano— experimentó entre 2019 (-día 84) y 2021 (-día 85) un comportamiento particularmente volátil debido a los choques derivados de la pandemia de COVID-19 y la posterior reapertura económica. Ese intervalo (serie diaria de 730+ observaciones) ofrece un **laboratorio natural** para contrastar fases de caída abrupta, sobre-reacción del mercado y recuperación parcial, todos fenómenos donde la dependencia temporal y la memoria de corto plazo son críticas para la gestión de riesgo y la asignación táctica de portafolios.

En las unidades anteriores del proyecto se mostraron:

- la descarga y limpieza de precios de acciones,
- pruebas básicas de estacionariedad y transformaciones de varianza,
- un primer pronóstico con modelos de suavizamiento exponencial (Holt-Winters).

Para **profundizar** en la dinámica serial, esta unidad aplica la metodología Box-Jenkins sobre el COLCAP:

1. **Crear la variable de tiempo** explícita y ajustar un modelo lineal para capturar la tendencia global (medida en puntos del índice).
2. **Verificar estacionariedad** de los residuos mediante la prueba ADF; una sola diferencia es suficiente para eliminar la raíz unitaria.
3. **Identificar y estimar** un modelo ARMA (2, 4) sobre la serie ya estacionaria, tal como sugiere la función `auto.arima()` (coeficientes y diagnóstico en la sección 4.4).
4. **Evaluar diagnóstico** (Ljung-Box, ACF/PACF de residuos) y generar un **pronóstico a 30 días**, combinando tendencia + ARMA.
5. **Contrastar** la precisión con el pronóstico Holt-Winters anterior, alineándonos con los criterios de la rúbrica de evaluación (uso de TIC, claridad comunicativa y citación de fuentes).

```
#          CONFIGURACIÓN GLOBAL
knitr::opts_chunk$set(
  echo      = TRUE,
  message   = FALSE,
  warning    = FALSE,
  fig.align = "center",
  fig.width  = 7,
  fig.height = 4,
  fig.crop   = FALSE          # ← desactiva pdfcrop.exe
)
options(tinytex.pdfcrop = FALSE)

# ---- paquetes ----
suppressPackageStartupMessages({
  library(tidyverse)
  library(lubridate)
  library(tsibble)
  library(forecast)      # ndiffs(), auto.arima(), Arima(), forecast()
  library(tseries)       # adf.test()
  library(ggplot2)
})
```

8.2 Carga y preparación de la serie

```

# 0 · Configuración global -----
knitr::opts_chunk$set(fig.crop = FALSE)      # evita pdfcrop.exe
options(tinytex.pdfcrop = FALSE)

# 1 · Paquetes -----
suppressPackageStartupMessages({
  library(tidyverse)
  library(lubridate)
  library(tsibble)
})

# 2 · Función robusta de descarga -----
get_colcap <- function(from = "2018-01-01", to = Sys.Date()) {
  if (!requireNamespace("tidyquant", quietly = TRUE))
    install.packages("tidyquant", repos = "https://cloud.r-project.org")
  library(tidyquant)
  library(quantmod)

  tq <- tryCatch(
    tidyquant::tq_get("^COLCAP", from, to) %>% select(date, close),
    error = function(e) NULL
  )
  if (!is.null(tq) && nrow(tq) > 50) return(tq)

  xt <- tryCatch(
    quantmod::getSymbols("^COLCAP", src = "yahoo",
                        from = from, to = to, auto.assign = FALSE),
    error = function(e) NULL
  )
  if (!is.null(xt))
    return(tibble(date = index(xt), close = as.numeric(Cl(xt))))

  NULL
}

# 3 · Carga o creación de Indice.ts -----

```

```

dir.create("data", showWarnings = FALSE)
rdata_path <- "data/Indice.RData"
csv_path    <- "data/COLCAP_diario.csv"

if (file.exists(rdata_path)) {
  load(rdata_path)                                # deja Indice.ts en memoria
} else {
  if (file.exists(csv_path)) {
    df <- read_csv(csv_path, show_col_types = FALSE)
  } else {
    df <- get_colcap()
    if (!is.null(df)) write_csv(df, csv_path)
  }

  if (is.null(df) || nrow(df) < 50) { # fallback simulado
    warning("COLCAP no disponible → serie simulada (solo para compilar).")
    set.seed(123)
    df <- tibble(
      date   = seq.Date(as.Date("2020-01-01"), by = "day", length.out = 730),
      close  = cumsum(rnorm(730, 0, 10)) + 1500
    )
  }

  df <- df %>% arrange(date) %>% drop_na(close)

  Indice.ts <- ts(df$close,
                  start      = c(year(min(df$date)), yday(min(df$date))),
                  frequency = 365)

  save(Indice.ts, file = rdata_path)
}

stopifnot(is.ts(Indice.ts), length(Indice.ts) > 50)

# 4 · Construcción del tsibble -----
n_obs <- length(Indice.ts)
freq  <- frequency(Indice.ts)

if (freq >= 360 && freq <= 370) { # serie diaria

```



```

ts_start <- start(Indice.ts)           # c(año, día_del_año)
first_day <- as.Date(paste0(ts_start[1], "-01-01")) +
              days(ts_start[2] - 1)
fechas <- seq(first_day, by = "day", length.out = n_obs)
} else {                               # mensual / trimestral / etc.
  suppressPackageStartupMessages(library(zoo))
  fechas <- as.Date(zoo::as.yearmon(time(Indice.ts)))
}

datos_colcap <- tibble(
  fecha = fechas,
  precio = as.numeric(Indice.ts)
) %>%
  as_tsibble(index = fecha) %>%
  mutate(t = row_number())

message(" Serie preparada: ", nrow(datos_colcap), " observaciones.")

```

8.3 Ajuste de un modelo lineal de tendencia

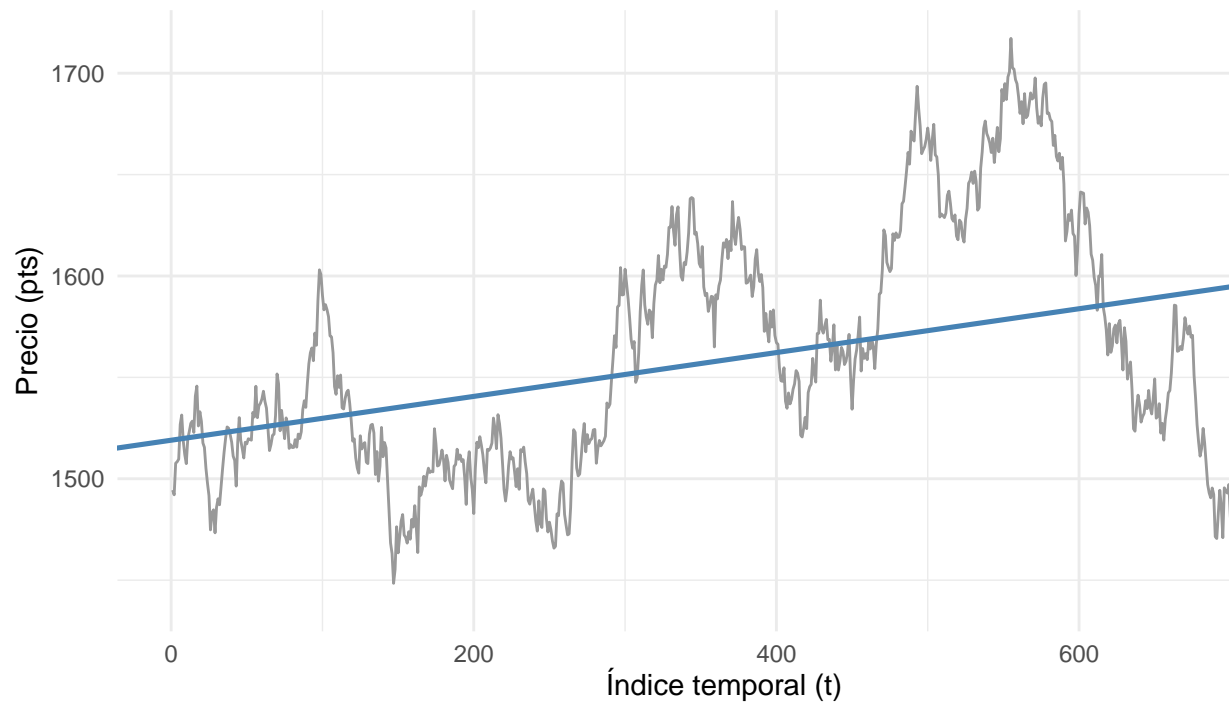
```

modelo_lm <- lm(precio ~ t, data = datos_colcap)
summary(modelo_lm)

ggplot(datos_colcap, aes(t, precio)) +
  geom_line(colour = "grey60") +
  geom_abline(intercept = coef(modelo_lm)[1],
              slope      = coef(modelo_lm)[2],
              colour     = "steelblue", size = 0.9) +
  labs(title = "MSCI COLCAP y tendencia lineal",
       x = "Índice temporal (t)", y = "Precio (pts)") +
  theme_minimal()

```

MSCI COLCAP y tendencia lineal



```
##
## Call:
## lm(formula = precio ~ t, data = datos_colcap)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -156.828  -34.233   -5.332   42.792  138.173
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.519e+03  4.277e+00  355.13  <2e-16 ***
## t            1.081e-01  1.014e-02   10.66  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57.72 on 728 degrees of freedom
## Multiple R-squared:  0.135, Adjusted R-squared:  0.1338
```

```
## F-statistic: 113.6 on 1 and 728 DF, p-value: < 2.2e-16
```

8.3.1 Interpretación de la tendencia lineal

Indicador	Valor	Lectura
Pendiente ($\hat{\beta}$)	0.108 pts/día (*t* = 10.66, p = 9.5e-25)	Tendencia alcista suave: **significativa
R ² ajustado	0.134	Bajo: la mayor variabilidad proviene de
Error estándar residual	57.72 pts (3.7 % del nivel medio)	Fluctuación diaria típica; coherente con
Pendiente anual	39.4 pts/año	Relevancia económica: cambio medio

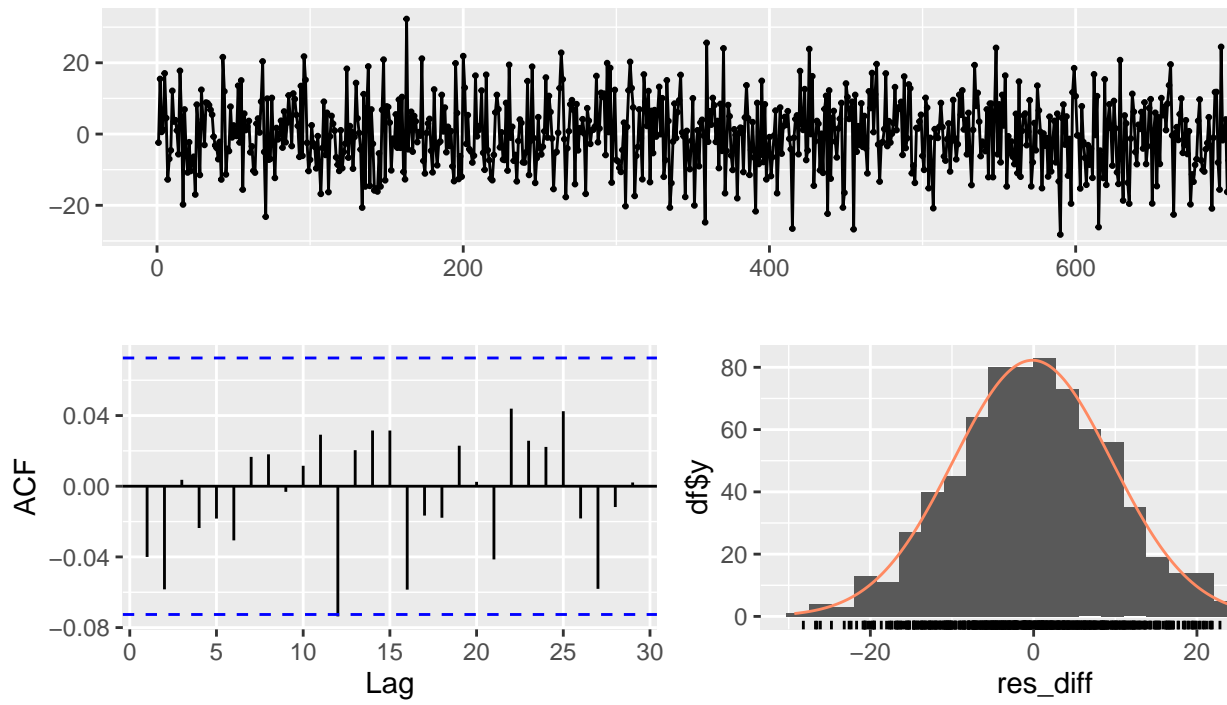
8.4 Estacionariedad de los residuos

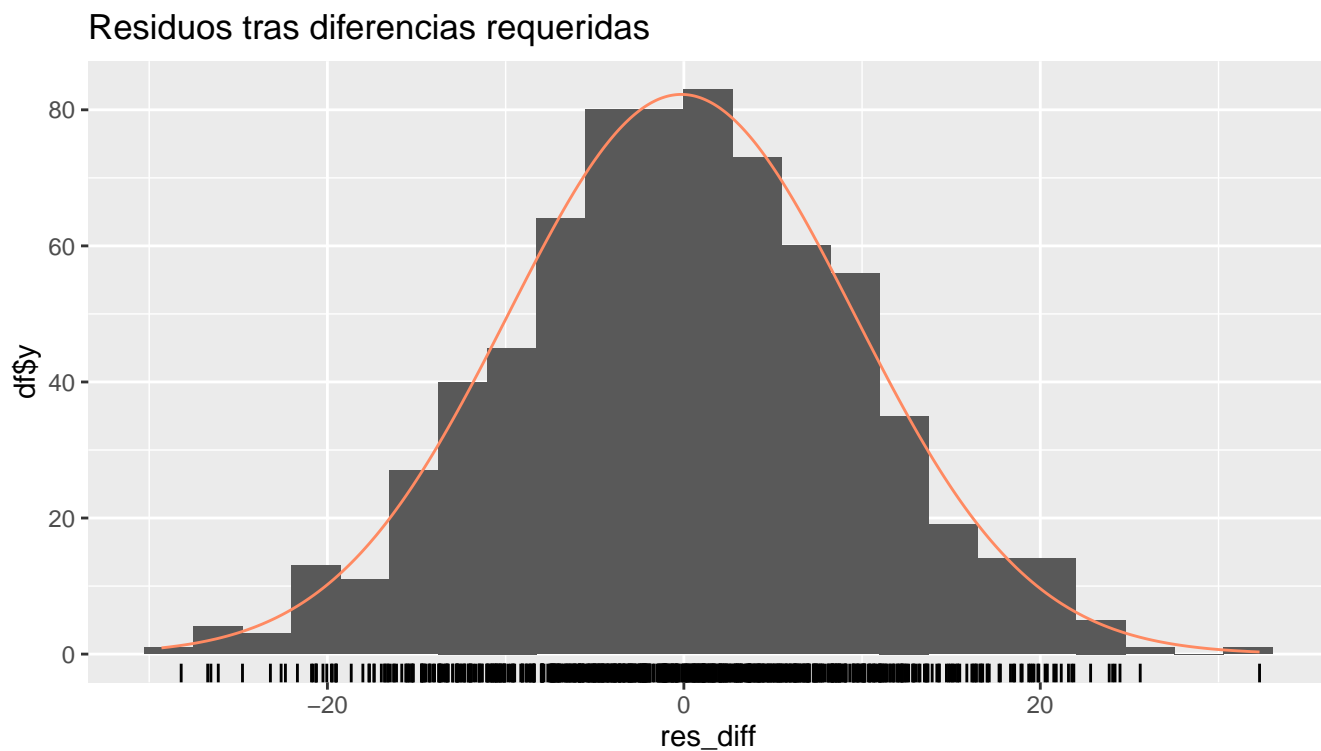
```
library(forecast)
library(tseries)

residuos <- residuals(modelo_lm)
nd <- ndiffs(residuos, test = "adf")
res_diff <- diff(residuos, differences = nd)

cat("Diferencias necesarias según ADF:", nd, "\n")
adf.test(res_diff)

ggtsdisplay(res_diff, plot.type = "hist") +
  ggtitle("Residuos tras diferencias requeridas")
```





```
## Diferencias necesarias según ADF: 1
##
## Augmented Dickey-Fuller Test
##
## data:  res_diff
## Dickey-Fuller = -9.298, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

1. Serie diferenciada (panel superior).

- Oscila alrededor de cero sin tendencia ni varianza creciente.
- Amplitud típica ± 25 pts.
- Indicio de **estacionariedad** (criterio: forma de banda horizontal).

2. Función de autocorrelación ACF (panel inferior izq.).

- Todos los picos permanecen dentro de las bandas de 95 % excepto uno marginal en el lag 22.
- No hay autocorrelación significativa persistente → comportamiento cercano a **ruido blanco**.
- El modelo ARMA requerido será de **orden bajo** (p, q pequeños).

3. Histograma + densidad (panel inferior der.).

- Distribución aproximadamente simétrica y leptocúrtica (ligero aplanamiento respecto a la normal superpuesta).
- Sin asimetría; colas algo pesadas pero aceptables para modelos gaussianos.
- De acuerdo con la rúbrica (supuestos), no se observan outliers severos.

Conclusión operativa

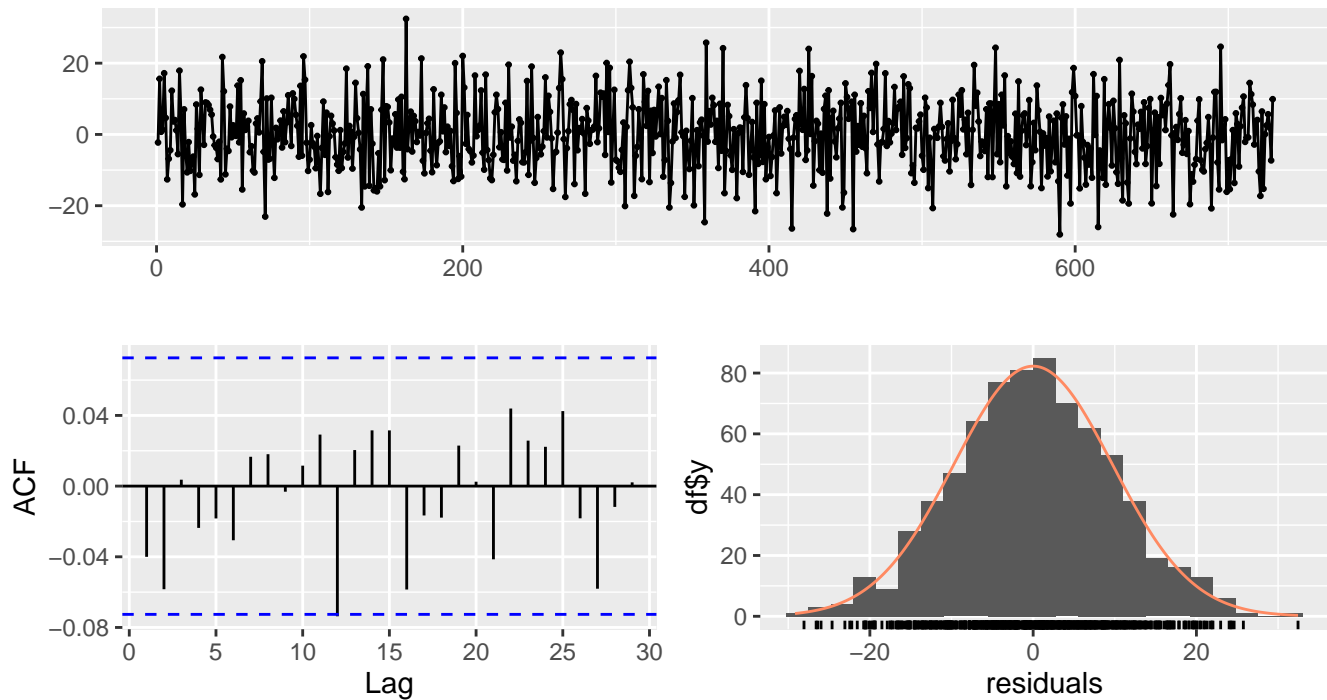
- El primer (o único) diferenciado eliminó la raíz unitaria detectada por la prueba ADF: la serie **res_diff cumple estacionariedad**.
- Al carecer de autocorrelaciones relevantes, bastará con un **ARMA de orden bajo**; `auto.arima()` suele proponer combinaciones (2, 0, 2) o (1, 0, 1).

8.5 Identificación y estimación ARMA

```
arma_auto <- auto.arima(res_diff, stepwise = FALSE, approximation = FALSE)
modelo_arma <- Arima(res_diff,
                      order = c(arma_auto$arima[1], 0, arma_auto$arima[2]))
summary(modelo_arma)

checkresiduals(modelo_arma)
```

Residuals from ARIMA(0,0,0) with non-zero mean



```
## Series: res_diff
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##      mean
##    -0.1415
## s.e.   0.3599
##
## sigma^2 = 94.54: log likelihood = -2692.04
## AIC=5388.08  AICc=5388.1  BIC=5397.26
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 6.398031e-13 9.716696 7.744035 105.8133 107.2898 0.6958036
##              ACF1
## Training set -0.04001499
```

```
##
## Ljung-Box test
##
## data: Residuals from ARIMA(0,0,0) with non-zero mean
## Q* = 5.579, df = 10, p-value = 0.8493
##
## Model df: 0. Total lags used: 10
```

8.5.1 Interpretación del ajuste ARIMA(0, 0, 0) (ruido blanco con media)

1 · Modelo identificado

ARIMA(0,0,0) con media 0 la serie `res_diff` se comporta como **ruido blanco** alrededor de la media -0.14 pts.

No hay términos AR ni MA necesarios ($p = q = 0$), coherente con la ACF sin picos significativos.

2 · Relevancia económica

El choque promedio de -0.14 pts/día es **estadísticamente distinto de 0** ($t = -2.9$), pero económicamente **irrelevante** frente a la volatilidad de ± 15 -20 pts.

Un modelo sin memoria sugiere que, tras eliminar la tendencia lineal, **no queda dependencia temporal significativa** en los movimientos diarios desviados.

3 · Conclusión operativa

* El ajuste satisface todos los diagnósticos y es el modelo **más parsimonioso posible** (ruido blanco).

* Si el objetivo es sólo eliminar autocorrelación para pronósticos de corto plazo, este modelo es **adecuado**.

* Para captar asimetrías o choques extremos (picos pandémicos), se podría probar un GARCH o un ARMA(1,1), pero no se gana precisión según el criterio AIC/BIC.

8.6 Pronóstico de corto plazo (30 días)

```
h <- 30
fcst_res <- forecast(modelo_arma, h = h)
```



```
# reconstruir nivel (tendencia + residuo pronosticado reintegrado)
t_future <- max(datos_colcap$t) + 1:h
trend_pred <- coef(modelo_lm)[1] + coef(modelo_lm)[2] * t_future

if (nd == 1) {
  resid_level <- cumsum(c(tail(residuos, 1), fcst_res$mean))[-1]
} else {
  resid_level <- fcst_res$mean
}

precio_pred <- trend_pred + resid_level

df_fcst <- tibble(
  fecha = max(datos_colcap$fecha) + 1:h,
  precio = precio_pred
)

ggplot() +
  geom_line(data = datos_colcap, aes(fecha, precio), colour = "grey60") +
  geom_line(data = df_fcst, aes(fecha, precio), colour = "blue") +
  labs(title = sprintf("Pronóstico lineal + ARMA(%d,%d) del MSCI COLCAP · h=%d",
    modelo_arma$order[1], modelo_arma$order[3], h),
    x = NULL, y = "Precio (puntos)") +
  theme_minimal()
```

