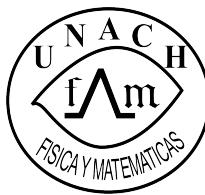




UNIVERSIDAD AUTÓNOMA DE
CHIAPAS



FACULTAD DE CIENCIAS EN FÍSICA Y
MATEMÁTICAS

Análisis comparativo del desempeño en
métodos para el pronóstico de series
temporales

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
LICENCIADA EN MATEMÁTICAS APLICADAS

PRESENTA:

JENNIFER SHERLYN LÓPEZ GARCÍA

DIRECTOR:

DR. YOFRE HERNÁN GARCÍA GÓMEZ

Tuxtla Gutiérrez, Chiapas a - de - del 2024.

Dedicatoria

Poner la dedicatoria.

Agradecimientos

Aquí ponemos agradecimientos

Tabla de contenidos

Resumen	5
Introducción	6
Objetivos	7
Objetivo General	7
Objetivos Específicos	7
I. Preliminares	8
1. Teoría de conjuntos	9
2. Probabilidad	14
2.1. Espacio muestral y eventos	15
2.2. Definición de probabilidad	17
2.3. Probabilidad condicional	20
2.4. Independencia de eventos	23
2.5. Variables aleatorias	24
2.5.1. Función de distribución	25
2.5.2. Función de densidad	28
2.5.3. Variables aleatorias discretas	28
2.5.4. Variables aleatorias continuas	29
2.6. Vectores aleatorios	30
3. Estadística	32
3.1. Datos univariados	32
3.1.1. Valor Esperado y momentos	33
3.1.2. Muestreo	36
3.2. Datos multivariados	37
4. Procesos estocásticos	41

II. Series de tiempo	44
5. Series de Tiempo	45
5.1. Conceptos básicos y manipulación de series de tiempo	46
5.1.1. Series de tiempo estacionarias	47
5.1.2. Conjuntos de datos de series temporales.	52
5.1.3. Suavizado de datos de series temporales.	59
5.2. Análisis y técnicas de descomposición.	61
5.2.1. Descomposición de datos estacionales	61
5.2.2. Ajuste estacional	71
5.3. Pronóstico y métodos predictivos.	73
5.3.1. Suavizador de media móvil predictivo	74
5.3.2. Suavizado exponencial	75
5.3.3. Pronóstico Holt-Winters	77
5.3.4. Modelo Autoregresivo (AR)	80
5.4. Evaluación de la precisión de los pronósticos	86
5.4.1. MAE	86
5.4.2. MSE	87
5.4.3. RMSE	87
5.4.4. MAPE	88
III. Redes neuronales	89
6. Redes Neuronales	90
6.1. Elementos fundamentales de las Redes Neuronales Artificiales	91
6.2. Arquitectura	96
6.2.1. Perceptrón simple	96
6.2.2. Perceptrón Multicapa (MLP)	99
6.3. Perceptrón	101
6.3.1. Teorema de convergencia del perceptrón	103
6.4. Funciones de activación	103
6.4.1. Lineal	104
6.4.2. Sigmoidal	104
6.4.3. Unidad lineal rectificadora (ReLU)	105
6.4.4. ReLU con fugas	105
6.4.5. Tangente hiperbólica	106
6.4.6. Softmax	106
6.5. Funciones de coste	106
6.6. Gradiente descendente	108
6.6.1. Algoritmo gradiente descendente	108
6.7. Perceptrón Multicapa	109
6.7.1. Entrenamiento y aprendizaje del Perceptrón Multicapa	111

6.8.	Evaluación de modelos de aprendizaje automático	116
6.8.1.	División de datos en entrenamiento y prueba	116
6.8.2.	Validación cruzada de K pliegues	117
IV. Estudio de caso		118
	Modelación y pronóstico del número de casos confirmados y fallecidos por COVID-19 en IRÁN	119
7. Pronóstico de infectados diarios		121
7.1.	Obtención de datos	121
7.2.	Ánalisis de la serie de tiempo de casos confirmados de COVID-19 en Irán	121
7.2.1.	Estadística descriptiva	121
7.2.2.	Componentes de la serie de tiempo	123
7.2.3.	Estacionariedad	124
7.3.	Entrenamiento, modelado, pronóstico y métricas de rendimiento	128
7.3.1.	Holt-Winters	128
7.3.2.	MLP	130
7.3.3.	Comparación de pronósticos con el conjunto de datos de prueba	132
7.3.4.	Conclusión	134
7.4.	Pronóstico de los próximos 30 días	135
7.4.1.	Implementación en R	135
7.4.2.	Implementación en Python	135
8. Pronóstico de decesos diarios		141
8.1.	Pronóstico, comparación y métricas de rendimiento	142
8.2.	Pronóstico de los próximos 30 días	145
8.2.1.	Implementación en R	145
8.2.2.	Implementación en Python	147
9. Conclusiones		149
References		151

Resumen

Esta tesis presenta un análisis comparativo de métodos de pronóstico para datos de series temporales, centrándose en la aplicación del método Holt-Winters y de las redes neuronales Perceptrón Multicapa (MLP). El estudio abarca una revisión exhaustiva de la teoría del análisis de series temporales y los fundamentos de las redes neuronales, proporcionando una sólida base teórica para comprender las metodologías empleadas.

La investigación examina el desempeño del método de Holt-Winters, una técnica clásica de suavizado exponencial, y MLP, un enfoque potente de modelado no lineal, en el pronóstico del número de casos confirmados y fallecidos debido al COVID-19 en Irán. Al aplicar estos métodos a datos del mundo real, la tesis evalúa su precisión, robustez y eficiencia computacional en la captura de la compleja dinámica de la progresión de la pandemia.

A través del análisis empírico y las evaluaciones comparativas, este estudio tiene como objetivo proporcionar información sobre las fortalezas y limitaciones de cada enfoque de pronóstico, contribuyendo así al avance de las metodologías de pronóstico para datos de series temporales, especialmente en el contexto de crisis de salud pública.

Introducción

La creciente disponibilidad de datos temporales en una amplia gama de campos ha impulsado la necesidad de desarrollar métodos eficaces para pronosticar su comportamiento futuro. En particular, el análisis y pronóstico de series temporales desempeña un papel crucial en la toma de decisiones en áreas como la economía, la meteorología, la ingeniería y la salud pública. Con la aparición de la pandemia de COVID-19, la capacidad de prever la evolución de la enfermedad se ha convertido en una prioridad urgente para los responsables de la salud y los planificadores de políticas.

Este trabajo se centra en el análisis comparativo de métodos de pronóstico para series temporales, con un enfoque específico en el método de Holt-Winters y las redes neuronales Perceptrón Multicapa (MLP). Para comprender en profundidad estos enfoques, se presenta una revisión exhaustiva de la teoría del análisis de series temporales y los fundamentos de las redes neuronales, estableciendo así las bases teóricas necesarias para su aplicación.

El método de Holt-Winters, basado en el suavizado exponencial, y MLP, una técnica de modelado no lineal, son seleccionados como enfoques principales debido a su amplia aplicación y capacidad para capturar patrones complejos en los datos temporales. Se busca evaluar el desempeño de estos métodos en la predicción del número de casos confirmados y fallecidos por COVID-19 en Irán, utilizando datos reales recopilados durante el curso de la pandemia.

El estudio se estructura en torno al desarrollo de modelos de pronóstico basados en Holt-Winters y MLP, seguido de una comparación exhaustiva de su precisión, robustez y eficiencia computacional. Además, se explora el potencial de estas técnicas para proporcionar información valiosa en la gestión de crisis de salud pública y la planificación de recursos.

En última instancia, se espera que este trabajo contribuya al avance de las metodologías de pronóstico para datos de series temporales, ofreciendo perspectivas significativas para la mejora de la capacidad predictiva en situaciones críticas como la pandemia de COVID-19.

Objetivos

Objetivo General

El objetivo principal de este estudio es comparar el método de pronóstico Holt-Winters con el pronóstico realizado mediante una red MLP (Perceptrón Multicapa), utilizando métricas de error relevantes para evaluar su desempeño.

Objetivos Específicos

1. Identificar y comprender las técnicas de pronóstico, incluyendo el suavizado exponencial y el aprendizaje profundo mediante una red MLP.
2. Evaluar y contrastar la eficacia de las implementaciones de Holt-Winters y la red MLP en los lenguajes de programación Python y R. Este análisis incluirá la identificación de ventajas y desventajas asociadas con cada implementación.
3. Reconocer y abordar las dificultades inherentes al pronóstico de datos epidemiológicos iniciales, particularmente en el contexto de una pandemia causada por un virus de propagación vectorial, como es el caso del COVID-19, que inicialmente es prácticamente desconocido.

Parte I.

Preliminares

1. Teoría de conjuntos

En esta sección, se abordan algunas de las ideas y conceptos elementales de la teoría de conjuntos que son necesarios para una introducción moderna a la teoría de la probabilidad.

Considere una colección de objetos en la que cada objeto se denomina punto o elemento. Se asume que dicha colección de objetos es lo suficientemente amplia como para incluir todos los puntos considerados en una discusión específica. La totalidad de estos puntos se conoce como espacio, universo o conjunto universal.

$\Omega = \mathbb{R}^2$, donde \mathbb{R}^2 es la colección de puntos ω en el plano y $\omega = (x, y)$ es cualquier par de números reales x e y .

Por lo general, se utilizarán letras latinas mayúsculas al comienzo del alfabeto, con o sin subíndices, para denotar conjuntos. Si ω es un punto o elemento que pertenece al conjunto A , se escribirá $\omega \in A$; si ω no es un elemento de A , se escribirá $\omega \notin A$.

Definición 1.1 (Subconjunto). Si cada elemento de un conjunto A también es un elemento de un conjunto B , entonces se define que A es un subconjunto de B , y se escribirá $A \subset B$ o $B \supset A$; se lee como “ A está contenido en B ” o “ B contiene a A ”.

Definición 1.2 (Conjuntos equivalentes). Dos conjuntos A y B se definen como equivalentes, o iguales, si $A \subset B$ y $B \subset A$. Esto se indicará escribiendo $A = B$.

Definición 1.3 (Conjunto vacío). Si un conjunto A no contiene puntos, se le llamará conjunto nulo o conjunto vacío, y se denotará por \emptyset .

Definición 1.4 (Complemento). El complemento de un conjunto A con respecto al espacio Ω , denotado por \bar{A} , A^c o $\Omega - A$, es el conjunto de todos los puntos que están en Ω pero no en A .

Definición 1.5 (Unión). Sea A y B dos subconjuntos cualesquiera de Ω ; entonces el conjunto que consiste en todos los puntos que están en A , en B o en ambos se define como la unión de A y B , y se escribe $A \cup B$.

Definición 1.6 (Intersección). Sean A y B dos subconjuntos cualesquiera de Ω ; entonces el conjunto formado por todos los puntos que están tanto en A como en B se define como la intersección de A y B , y se escribe $A \cap B$.

Definición 1.7 (Diferencia de conjuntos). Sean A y B dos subconjuntos cualesquiera de Ω . El conjunto de todos los puntos en A que no están en B se denotará por $A - B$ y se define como la diferencia de conjuntos.

Las operaciones de complemento, unión e intersección de conjuntos se han introducido en las definiciones Definición 1.4 a Definición 1.6, respectivamente. Estas operaciones de conjuntos satisfacen varias leyes, que a continuación se sintetizan. (Ash y Doleans-Dade 2000)

Teorema 1.1 (Leyes del álgebra de conjuntos).

i. Leyes de idempotencia

$$\begin{aligned} A \cup A &= A \\ A \cap A &= A \end{aligned}$$

ii. Leyes asociativas

$$\begin{aligned} (A \cup B) \cup C &= A \cup (B \cup C) \\ (A \cap B) \cap C &= A \cap (B \cap C) \end{aligned}$$

iii. Leyes commutativas

$$\begin{aligned} A \cup B &= B \cup A \\ A \cap B &= B \cap A \end{aligned}$$

iv. Leyes distributivas

$$\begin{aligned} A \cup (B \cap C) &= (A \cup B) \cap (A \cup C) \\ A \cap (B \cup C) &= (A \cap B) \cup (A \cap C) \end{aligned}$$

v. Leyes de identidad

$$\begin{aligned} A \cup \emptyset &= A \\ A \cap \Omega &= A \\ A \cup \Omega &= \Omega \\ A \cap \emptyset &= \emptyset \end{aligned}$$

vi. Leyes de complemento

$$\begin{aligned} A \cup A^c &= \Omega \\ A \cap A^c &= \emptyset \\ (A^c)^c &= A \\ \Omega^c &= \emptyset \\ \emptyset^c &= \Omega \end{aligned}$$

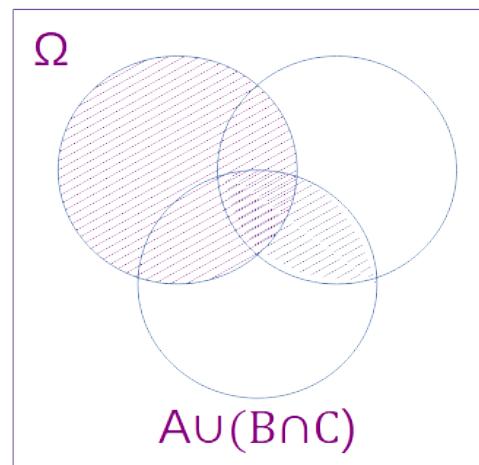
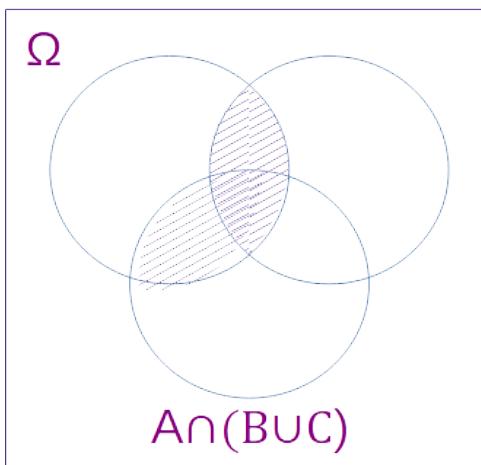
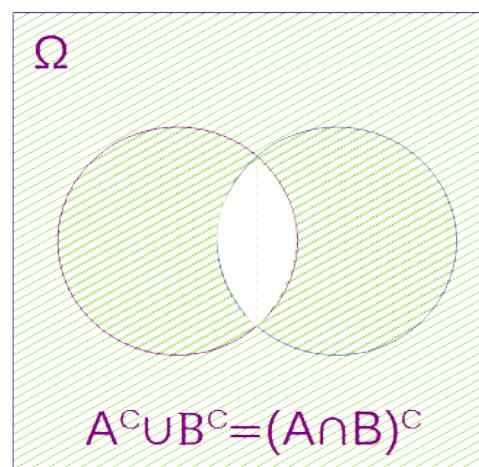
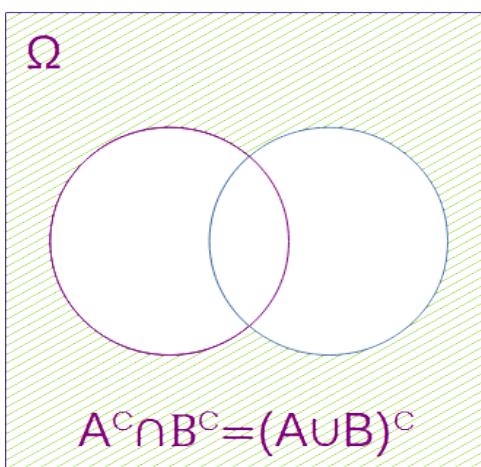
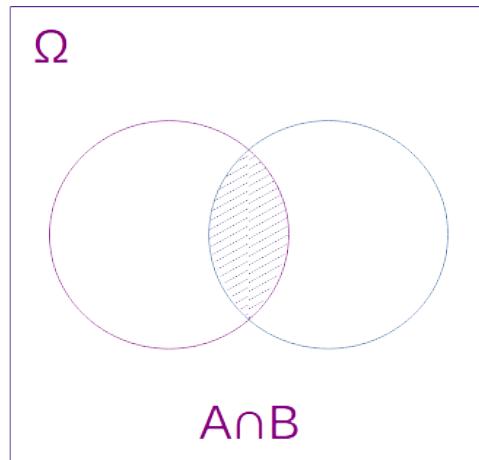
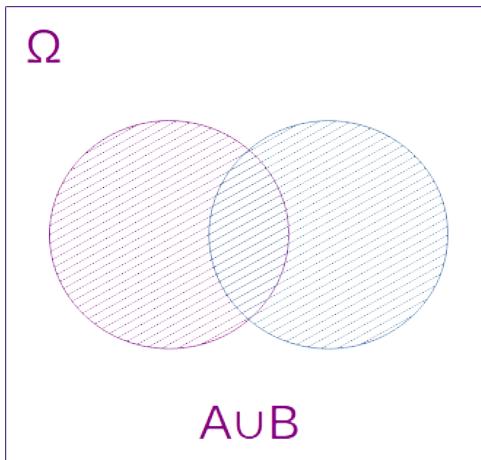


Figura 1.1.: Diagramas de Venn

vii. Leyes de De Morgan

$$(A \cup B)^c = A^c \cap B^c$$

$$(A \cap B)^c = A^c \cup B^c$$

Algunas de las leyes mencionadas anteriormente se ilustran en los diagramas de Venn en la Figura 1.1. Aunque se utilizará libremente cualquiera de las leyes mencionadas, podría resultar instructivo proporcionar una prueba de una de ellas para ilustrar la técnica. Se considera el siguiente ejemplo:

Ejemplo

Demostrar que $(A \cup B)^c = A^c \cap B^c$.

Prueba. Según la definición, dos conjuntos son iguales si cada uno está contenido en el otro. Primero se demuestra que $(A \cup B)^c \subset A^c \cap B^c$ al probar que si $\omega \in (A \cup B)^c$, entonces $\omega \in A^c \cap B^c$. Ahora bien, $\omega \in (A \cup B)^c$ implica que $\omega \notin A \cup B$, lo cual implica que $\omega \notin A$ y $\omega \notin B$, lo que a su vez implica que $\omega \in A^c$ y $\omega \in B^c$; es decir, $\omega \in A^c \cap B^c$. A continuación se demuestra que $A^c \cap B^c \subset (A \cup B)^c$. Sea $\omega \in A^c \cap B^c$, lo que significa que ω pertenece tanto a A^c como a B^c . Entonces, $\omega \notin A \cup B$, ya que de lo contrario ω debería pertenecer al menos a uno de los conjuntos A o B , lo cual contradice que ω pertenezca tanto a A^c como a B^c ; sin embargo, $\omega \notin A \cup B$ implica que $\omega \in (A \cup B)^c$, lo que completa la prueba. \square

Se han introducido la unión y la intersección de dos conjuntos; estas definiciones se extienden inmediatamente a más de dos conjuntos, de hecho, a un número arbitrario de conjuntos. Es costumbre distinguir entre los conjuntos en una colección de subconjuntos de Ω asignándoles nombres en forma de subíndices.

Se considera el conjunto de índices Λ como el catálogo de nombres o índices. A Λ también se le denomina conjunto de índices. Por ejemplo, si se tiene interés únicamente en dos conjuntos, entonces el conjunto de índices Λ incluye solo dos índices, por ejemplo, 1 y 2; así, $\Lambda = \{1, 2\}$.

Definición 1.8 (Unión e intersección de conjuntos). Sea Λ un conjunto de índices y $\{A_\lambda : \lambda \in \Lambda\} = \{A_\lambda\}$, una colección de subconjuntos de Ω indexados por Λ . El conjunto de puntos que consiste en todos los puntos que pertenecen a A_λ para al menos un λ se denomina unión de los conjuntos $\{A_\lambda\}$ y se denota como $\bigcup_{\lambda \in \Lambda} A_\lambda$. El conjunto de puntos que consiste en todos los puntos que pertenecen a A_λ para cada λ se denomina intersección de los conjuntos $\{A_\lambda\}$ y se denota como $\bigcap_{\lambda \in \Lambda} A_\lambda$. Si Λ está vacío, entonces se define $\bigcup_{\lambda \in \Lambda} A_\lambda = \emptyset$ y $\bigcap_{\lambda \in \Lambda} A_\lambda = \Omega$.

Uno de los teoremas más fundamentales que relaciona las uniones, intersecciones y complementos para una colección arbitraria de conjuntos se debe a De Morgan.

Teorema 1.2 (Teorema de De Morgan). *Sea Λ un conjunto de índices y $\{A_\lambda\}$ una colección de subconjuntos de Ω indexados por Λ . Entonces,*

$$i. \left(\bigcup_{\lambda \in \Lambda} A_\lambda \right)^c = \bigcap_{\lambda \in \Lambda} A_\lambda^c$$

$$ii. \left(\bigcap_{\lambda \in \Lambda} A_\lambda \right)^c = \bigcup_{\lambda \in \Lambda} A_\lambda^c$$

Definición 1.9 (Disjuntos o mutuamente excluyentes). Los subconjuntos A y B de Ω se definen como mutuamente excluyentes o disjuntos si $A \cap B = \emptyset$. Los subconjuntos A_1, A_2, \dots se definen como mutuamente excluyentes si $A_i \cap A_j = \emptyset$ para cada $i \neq j$.

2. Probabilidad

Una de las herramientas fundamentales de la estadística es la probabilidad, la cual tuvo sus inicios formales con los juegos de azar en el siglo XVII. Los juegos de azar, como su nombre indica, involucran acciones como girar una rueda de ruleta, lanzar dados, lanzar una moneda, sacar una carta, entre otros, en los que el resultado de un evento es incierto. No obstante, se reconoce que aunque el resultado de cada evento en particular pueda ser incierto, existe un patrón predecible a largo plazo. Por ejemplo, se sabe que en múltiples lanzamientos de una moneda ideal (equilibrada y simétrica), aproximadamente la mitad de los resultados serán caras. Es esta regularidad predecible a largo plazo la que permite a las casas de juego mantener sus negocios.

Un tipo similar de incertidumbre y regularidad a largo plazo se observa con frecuencia en la ciencia experimental. Por ejemplo, en la ciencia de la genética no se puede determinar con certeza si una descendencia será masculina o femenina, pero a largo plazo se sabe aproximadamente qué porcentaje de descendencia será de cada sexo. De manera similar, una compañía de seguros de vida no puede predecir qué personas en los Estados Unidos morirán a los 50 años, pero puede hacer predicciones precisas sobre cuántas personas morirán a esa edad en promedio.

Para brindar una idea de lo que es la probabilidad, (Mood, Graybill, y Boes 1986) proporciona las siguientes definiciones:

Definición 2.1 (Probabilidad clásica). Si un experimento aleatorio puede resultar en n resultados mutuamente excluyentes e igualmente probables y si s de estos resultados tienen un atributo A , entonces la probabilidad de A es la fracción s/n .

Definición 2.2 (Probabilidad frecuentista). Suponiendo que después de n repeticiones, para valores muy grandes de n , un evento A puede ocurrir s veces. Entonces $p = s/n$.

Estas definiciones, a pesar de su intuición, presentan limitaciones significativas. Por ejemplo, la primera definición es circular, ya que la frase “igualmente probables” es justamente lo que se intenta definir. Además, la segunda definición no especifica los valores de n , lo cual puede generar ambigüedad. Estas definiciones son consideradas antiguas, pero aún pueden brindar una comprensión general del concepto de probabilidad.

2.1. Espacio muestral y eventos

A continuación, se presentarán algunas definiciones que resultarán de gran utilidad para adquirir un mayor conocimiento sobre el concepto de probabilidad. Se usará como referencia (Lipschutz 1996).

Definición 2.3 (Espacio muestral). El espacio muestral, denotado por Ω , es la colección o totalidad de todos los posibles resultados de un experimento conceptual.

Un resultado particular, es decir, un elemento del espacio muestral Ω , se denomina un *punto muestral* o una *muestra*.

Definición 2.4 (Evento). Un evento A es un subconjunto del espacio muestral Ω , es decir, es un conjunto de resultados.

Definición 2.5 (Espacio de eventos). La clase de todos los eventos asociados a un experimento dado se define como el espacio de eventos y se denominará por \mathfrak{F} .

Definición 2.6 (Evento particular). El evento $\{\omega\}$, que está constituido por un solo punto $\omega \in \Omega$, se denomina *evento muestral* o *punto muestral*.

Las definiciones anteriores no definen con precisión lo que es un evento. Un evento siempre será un subconjunto del espacio muestral, pero para espacios muestrales suficientemente grandes, no todos los subconjuntos serán eventos. Por lo tanto, la clase de todos los subconjuntos del espacio muestral no necesariamente corresponderá al espacio de eventos. Sin embargo, se observará que la clase de todos los eventos siempre se puede seleccionar lo suficientemente grande como para incluir todos aquellos subconjuntos (eventos) cuya probabilidad se desee analizar. Si el espacio muestral consta solo de un número finito de puntos, entonces el espacio de eventos correspondiente será la clase de todos los subconjuntos del espacio muestral.

Los conceptos presentados se ilustran con unos ejemplos muy simples;

Ejemplo (*Lanzamiento de una moneda*)

Considerando el experimento de lanzar una moneda, este experimento es uno de los más sencillos, pero permite representar claramente los conceptos. El espacio muestral estaría conformado por $\Omega = \{A, S\}$, donde A representa el resultado de que caiga águila y S representa el resultado de que caiga sol. El conjunto \mathfrak{F} podría estar representado por $\{\{A\}, \{S\}\}$, que también es un subconjunto de Ω .

Es importante destacar que el conjunto vacío \emptyset y el conjunto completo Ω también son subconjuntos de Ω , pero generalmente no se consideran eventos de interés en este contexto.

Ejemplo (*Lanzamiento de un dado*)

Considerando el experimento de lanzar un dado de 6 caras. El espacio muestral Ω se define como

$$\Omega = \{1, 2, 3, 4, 5, 6\}.$$

Un punto muestral podría ser un resultado específico, por ejemplo, $\{1\}$. Todos los subconjuntos posibles de resultados constituirían el conjunto \mathfrak{F} . Se definen los eventos A , que representa el evento de obtener un resultado par; B , que representa el evento de obtener un resultado impar; y C , que representa el evento de obtener un resultado mayor a 3. Por lo tanto, se tiene:

$$A = \{2, 4, 6\}, \quad B = \{1, 3, 5\}, \quad C = \{4, 5, 6\}.$$

Se observa que el evento de la unión de los eventos B y C , denotado como $B \cup C$, es igual a $\{1, 3, 4, 5, 6\}$, el cual también es un evento en el espacio muestral Ω . Finalmente, se destaca que los eventos A y B no tienen elementos en común, es decir, $A \cap B = \emptyset$. Por lo tanto, se dice que estos eventos son ajenos, mutuamente excluyentes o disjuntos.

La definición de espacio muestral es precisa y satisfactoria, mientras que las definiciones de evento y espacio de eventos no son completamente satisfactorias. Se mencionó que si el espacio muestral era “suficientemente grande”, no todos los subconjuntos del espacio muestral serían eventos; sin embargo, no se especificó exactamente qué subconjuntos serían eventos y cuáles no lo serían. En lugar de desarrollar las matemáticas necesarias para definir con precisión qué subconjuntos de Ω constituyen nuestro espacio de eventos \mathfrak{F} , se pueden enunciar algunas propiedades de \mathfrak{F} que parecen razonables requerir.

- i. $\Omega \in \mathfrak{F}$.
- ii. Si $A \in \mathfrak{F}$, entonces $A^c \in \mathfrak{F}$.
- iii. Si A_1 y $A_2 \in \mathfrak{F}$, entonces $A_1 \cup A_2 \in \mathfrak{F}$.

Se mencionó anteriormente que el interés principal radica en los eventos debido a la probabilidad de que ocurran. Por lo tanto, es deseable que \mathfrak{F} incluya Ω , el evento seguro. Además, si A es un evento, lo que significa que se puede hablar sobre la probabilidad de que ocurra A , entonces A^c también debería ser un evento para poder hablar sobre la probabilidad de que A no ocurra. De manera similar, si A_1 y A_2 son eventos, entonces $A_1 \cup A_2$ también debería ser un evento.

Cualquier colección de eventos con propiedades (i.) y (iii.) se denomina álgebra booleana, o simplemente álgebra, de eventos. Cabe señalar que la colección de todos los subconjuntos de Ω satisface necesariamente las propiedades mencionadas anteriormente. Varios resultados se derivan de las propiedades asumidas anteriormente de \mathfrak{F} .

2.2. Definición de probabilidad

En esta sección se presenta la definición axiomática de probabilidad. Aunque esta definición formal de probabilidad por sí sola no permite asignar probabilidades reales a eventos que consisten en ciertos resultados de experimentos aleatorios, es otra de una serie de definiciones que conducen a ese objetivo. Dado que la probabilidad, al igual que los conceptos que se presentarán, se define como una función particular, se inicia esta subsección con una revisión del concepto de función

Definición 2.7 (Función). Una función, llamada $f(\cdot)$, con dominio A y contradominio B , es una colección de pares ordenados, llamados (a, b) , que cumplen las siguientes condiciones:

- i. $a \in A$ y $b \in B$
- ii. Cada $a \in A$ aparece como el primer elemento de algún par ordenado en la colección (cada $b \in B$ no necesariamente es el segundo elemento de algún par ordenado)
- iii. Ningún par ordenado en la colección tiene el mismo primer elemento que otro par ordenado distinto.

Si $(a, b) \in f(\cdot)$, se escribe $b = f(a)$ (se lee “ b es igual a f de a ”) y se denomina $f(a)$ como el valor de $f(\cdot)$ en a . Para cualquier $a \in A$, $f(a)$ es un elemento de B ; mientras que $f(\cdot)$ es un conjunto de pares ordenados. El conjunto de todos los valores de $f(\cdot)$ se denomina rango de $f(\cdot)$; es decir, el rango de $f(\cdot) = \{b \in B : b = f(a) \text{ para algún } a \in A\}$ y siempre es un subconjunto del contradominio B , pero no necesariamente igual a él. $f(a)$ también se denomina imagen de a bajo $f(\cdot)$, y a se denomina preimagen de $f(a)$.

Ejemplo

Sean $f_1(\cdot)$ y $f_2(\cdot)$ dos funciones con la recta real como su dominio y contradominio, definidas por

$$f_1(\cdot) = \{(x, y) : y = x^3 + 1, -\infty < x < \infty\}$$

y

$$f_2(\cdot) = \{(x, y) : y = x^2, -\infty < x < \infty\}$$

El rango de $f_1(\cdot)$ es el contradominio, que es toda la recta real, pero el rango de $f_2(\cdot)$ son todos los números reales no negativos, que no es igual al contradominio.

De particular interés será una clase de funciones conocidas como funciones indicadoras.

Definición 2.8 (Función indicadora). Sea Ω cualquier espacio con puntos ω y A cualquier subconjunto de Ω . La función indicadora de A , denominada $I_A(\cdot)$, es la función con dominio Ω y contradominio formado por dos números reales, 0 y 1, definida por

$$I_A(\omega) = \begin{cases} 1 & \text{si } \omega \in A \\ 0 & \text{si } \omega \notin A \end{cases}$$

$I_A(\cdot)$ claramente “indica” el conjunto A .

Propiedades de la función indicadora.

Sea Ω cualquier espacio y \mathfrak{F} cualquier colección de subconjuntos de Ω :

- i. $I_A(\omega) = 1 - I_{A^c}(\omega)$ para cada $A \in \mathfrak{F}$.
- ii. $I_{A_1 A_2 \dots A_n}(\omega) = I_{A_1}(\omega) \cdot I_{A_2}(\omega) \dots I_{A_n}(\omega)$ para $A_1, \dots, A_n \in \mathfrak{F}$.
- iii. $I_{A_1 \cup A_2 \cup \dots \cup A_n}(\omega) = \max [I_{A_1}(\omega), I_{A_2}(\omega), \dots, I_{A_n}(\omega)]$ para $A_1, \dots, A_n \in \mathfrak{F}$.
- iv. $I_A^2(\omega) = I_A(\omega)$ para cada $A \in \mathfrak{F}$.

La función indicadora será utilizada para “indicar” subconjuntos de la recta real; por ejemplo;

$$I_{\{[0,1]\}}(x) = I_{[0,1]}(x) = \begin{cases} 1 & \text{si } 0 \leq x < 1 \\ 0 & \text{otro caso} \end{cases}$$

y si I^+ denota el conjunto de números enteros positivos,

$$I_{I^+}(X) = \begin{cases} 1 & \text{si } x \text{ es algún entero positivo} \\ 0 & \text{otro caso} \end{cases}$$

Otro tipo de función del cual se tendrá ocasión de discutir es la función de conjunto definida como cualquier función que tiene como dominio una colección de conjuntos y como contradominio la recta real, incluyendo posiblemente el infinito. A continuación se muestra un ejemplos de función de conjunto.

Ejemplo

Sea Ω el espacio muestral correspondiente al experimento de lanzar dos dados, y sea \mathfrak{F} la colección de todos los subconjuntos de Ω . Para cualquier $A \in \mathfrak{F}$, se define $N(A)$ como el número de resultados, o puntos en Ω , que están en A . Entonces, $N(\emptyset) = 0$,

$N(\Omega) = 36$ y $N(A) = 6$ si A es el evento que contiene aquellos resultados que tienen un total de siete puntos arriba.

La función de tamaño del conjunto aludida en el ejemplo anterior puede ser definida, en general, para cualquier conjunto A como el número de puntos en A , donde A es un miembro de una colección arbitraria de conjuntos \mathfrak{F} .

La función de probabilidad que se definirá será una función de conjunto particular.

Definición 2.9 (Función de probabilidad). Sea A un evento del espacio muestral Ω . Una función $P : \mathfrak{F} \rightarrow [0, 1]$ es llamada función de probabilidad y $P(A)$ se denomina la *probabilidad* del evento A si se cumplen los siguientes axiomas:

- i. **No negatividad:** Para todo evento A en \mathfrak{F} , la probabilidad $P(A)$ es un número no negativo, es decir, $P(A) \geq 0$.
- ii. **Probabilidad unitaria:** La probabilidad del espacio muestral completo Ω es igual a 1, es decir, $P(\Omega) = 1$.
- iii. **Aditividad:** Para cualquier colección de eventos mutuamente excluyentes A_1, A_2, A_3, \dots , la probabilidad de la unión de estos eventos es igual a la suma de las probabilidades individuales, es decir,

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)$$

Estos axiomas establecen las propiedades esenciales que debe cumplir una función de probabilidad para ser considerada válida. Cumplir con estos axiomas garantiza que la función de probabilidad asigna valores coherentes y consistentes a los eventos en el espacio muestral.

A partir de los axiomas, se derivan otras propiedades que ayudan a calcular las probabilidades de varios eventos.

- i. $P(\emptyset) = 0$.
- ii. Si A_1, \dots, A_n son eventos mutuamente excluyentes en \mathfrak{F} , entonces

$$P\left(\bigcup_{i=1}^n A_i\right) = \sum_{i=1}^n P(A_i).$$

- iii. Si A es un evento en \mathfrak{F} , entonces

$$P(A^c) = 1 - P(A).$$

iv. Si $A, B \in \mathfrak{F}$, entonces

$$P(A) = P(A \cap B) + P(A \cap B^c)$$

y

$$P(A - B) = P(A \cap B^c) = P(A) - P(A \cap B).$$

v. Si $A, B \in \mathfrak{F}$ y $A \subset B$, entonces $P(A) \leq P(B)$.

vi. Para cualesquiera dos eventos $A, B \in \mathfrak{F}$;

$$P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

Más generalmente, para eventos $A_1, A_2, \dots, A_n \in \mathfrak{F}$

$$\begin{aligned} P\left(\bigcup_{i=1}^n A_i\right) &= \sum_{j=1}^n P(A_j) - \sum \sum_{i < j} P(A_i \cap A_j) \\ &\quad + \sum \sum \sum_{i < j < k} P(A_i \cap A_j \cap A_k) - \dots + (-1)^{n+1} P(A_1 \cap \dots \cap A_n) \end{aligned}$$

Teorema 2.1 (Desigualdad de Boole). *Si $A_1, A_2, \dots, A_n \in \mathfrak{F}$, entonces*

$$P\left(\bigcup_{i=1}^n A_i\right) \leq \sum_{i=1}^n P(A_i)$$

Finalmente se concluye esta subsección con la siguiente definición;

Definición 2.10 (Espacio de probabilidad). Un espacio de probabilidad es la terna $(\Omega, \mathfrak{F}, P)$, donde Ω es un espacio muestral, \mathfrak{F} es una colección (asumida como un álgebra) de eventos (cada uno un subconjunto de Ω), y P es una función de probabilidad con dominio \mathfrak{F} .

2.3. Probabilidad condicional

En ocasiones, es de interés conocer la probabilidad de un evento, dado que haya ocurrido otro. En este sentido, se define la probabilidad condicional.

Definición 2.11 (Probabilidad condicional). Sean A y B dos eventos en \mathfrak{F} del espacio de probabilidad dado $(\Omega, \mathfrak{F}, P)$. La probabilidad condicional del evento A dado el evento B , denotada por $P(A|B)$, se define como sigue;

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad \text{si } P(B) > 0,$$

y se deja sin definir si $P(B) = 0$.

Observación. Una fórmula que es evidente a partir de la definición es

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$$

si tanto $P(A)$ como $P(B)$ son diferentes de cero. Esta fórmula relaciona $P(A|B)$ con $P(B|A)$ en términos de las probabilidades incondicionales $P(A)$ y $P(B)$.

De la definición anterior, se desprenden las siguientes propiedades de la función de probabilidad condicional. Se asume que el espacio de probabilidad $(\Omega, \mathfrak{F}, P)$ está dado, y se considera que $B \in \mathfrak{F}$ cumple con $P(B) > 0$.

- i. $P(\emptyset|B) = 0$.
- ii. Si A_1, A_2, \dots, A_n son eventos mutuamente excluyentes en \mathfrak{F} , entonces

$$P\left(\bigcup_{i=1}^n A_i|B\right) = \sum_{i=1}^n P(A_i|B).$$

- iii. Si A es un evento en \mathfrak{F} , entonces $P(A^c|B) = 1 - P(A|B)$.
- iv. Si $A_1, A_2 \in \mathfrak{F}$, entonces $P(A_1|B) = P(A_1 \cap A_2|B) + P(A_1 \cap A_2^c|B)$.
- v. Para cualesquiera dos eventos $A_1, A_2 \in \mathfrak{F}$

$$P(A_1 \cup A_2|B) = P(A_1|B) + P(A_2|B) - P(A_1 \cap A_2|B).$$

- vi. Si $A_1, A_2 \in \mathfrak{F}$ y $A_1 \subset A_2$, entonces $P(A_1|B) \leq P(A_2|B)$.
- vii. Si $A_1, A_2, \dots, A_n \in \mathfrak{F}$, entonces

$$P\left(\bigcup_{i=1}^n A_i|B\right) \leq \sum_{i=1}^n P(A_i|B).$$

A continuación se mencionan unos teoremas de gran importancia. La aplicación de dichos teoremas se ilustran con unos ejemplos.

Teorema 2.2 (Teorema de probabilidades totales). *Para un espacio de probabilidad dado $(\Omega, \mathfrak{F}, P)$, si B_1, B_2, \dots, B_n es una colección de eventos mutuamente disjuntos en \mathfrak{F} que satisfacen $\Omega = \bigcup_{j=1}^n B_j$ y $P(B_j) > 0$ para $j = 1, \dots, n$, entonces para cada $A \in \mathfrak{F}$,*

$$P(A) = \sum_{j=1}^n P(A|B_j)P(B_j).$$

Prueba. Se observa que $A = \bigcup_{j=1}^n A \cap B_j$ y los conjuntos $A \cap B_j$ son mutuamente disjuntos; por lo tanto,

$$P(A) = P\left(\bigcup_{j=1}^n A \cap B_j\right) = \sum_{j=1}^n P(A \cap B_j) = \sum_{j=1}^n P(A|B_j)P(B_j)$$

□

Ejemplo (*Seleccionar una pelota de varias urnas*)

Hay dos urnas con pelotas de diferentes colores, todas del mismo tamaño. En la primera urna, hay tres pelotas rojas, tres blancas y cuatro negras; en la segunda urna, hay cuatro pelotas rojas, tres blancas y una negra. Se elige una urna al azar y se saca una pelota de ella. ¿Cuál es la probabilidad de que la pelota extraída sea blanca?

Solución. Se debe observar que la elección de las urnas constituye dos eventos mutuamente excluyentes, ya que la unión de ambos eventos constituye el espacio muestral (todas las pelotas están en la primera o segunda urna). Se denomina B_1 al evento de seleccionar la primera urna, y B_2 al evento de seleccionar la segunda urna.

El evento de extraer una pelota blanca puede tener lugar tanto al elegir la primera urna como al elegir la segunda urna, lo que permite la aplicación de la fórmula del teorema de probabilidades totales. Se designa como A al evento de seleccionar una pelota blanca.

De esta manera,

$$P(A) = \sum_{j=1}^2 P(A|B_j)P(B_j) = P(A|B_1)P(B_1) + P(A|B_2)P(B_2)$$

Bajo la premisa de probabilidades equivalentes, se tiene que $P(B_1) = P(B_2) = \frac{1}{2}$, $P(A|B_1) = \frac{3}{10}$ y $P(A|B_2) = \frac{3}{8}$, por lo que la probabilidad de elegir una pelota blanca es 0.3375.

Teorema 2.3 (Teorema de Bayes). *Para un espacio de probabilidad dado $(\Omega, \mathfrak{F}, P)$, si B_1, B_2, \dots, B_n es una colección de eventos mutuamente disjuntos en \mathfrak{F} que satisfacen $\Omega = \bigcup_{j=1}^n B_j$ y $P(B_j) > 0$ para $j = 1, \dots, n$, entonces para cada $A \in \mathfrak{F}$ para el cual $P(A) > 0$*

$$P(B_k|A) = \frac{P(A|B_k)P(B_k)}{\sum_{j=1}^n P(A|B_j)P(B_j)}.$$

Prueba. Se observa que

$$P(B_k|A) = \frac{P(B_k \cap A)}{P(A)} = \frac{P(A|B_k)P(B_k)}{\sum_{j=1}^n P(A|B_j)P(B_j)}$$

utilizando tanto la definición de probabilidad condicional como el teorema de probabilidad total. \square

Ejemplo (*Seleccionar una pelota de varias urnas*)

Considérese el problema de las urnas. Teniendo en cuenta que la pelota extraída fue blanca, ¿cuál es la probabilidad de que haya provenido de la primera urna?

Solución. Debe calcularse la probabilidad $P(B_1|A)$. Al utilizar el teorema de Bayes, solo se realiza la sustitución en la fórmula y se obtiene que la probabilidad resultante es 0.4444.

Teorema 2.4 (Regla de multiplicación). *Para un espacio de probabilidad dado $(\Omega, \mathfrak{F}, P)$, sean A_1, A_2, \dots, A_n eventos pertenecientes a \mathfrak{F} para los cuales $P(A_1 \cdots A_{n-1}) > 0$; entonces*

$$P(A_1 A_2 \cdots A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1 A_2) \cdots P(A_n|A_1 \cdots A_{n-1})$$

2.4. Independencia de eventos

Si $P(A|B)$ no depende del evento B , es decir, $P(A|B) = P(A)$, entonces parecería natural decir que el evento A es independiente del evento B . Esto se establece en la siguiente definición.

Definición 2.12 (Eventos independientes). Para un espacio de probabilidad dado $(\Omega, \mathfrak{F}, P)$, sean A y B dos eventos en \mathfrak{F} . Los eventos A y B se definen como *independientes* si y solo si se cumple alguna de las siguientes condiciones:

- i. $P(A \cap B) = P(A)P(B)$.
- ii. $P(A|B) = P(A)$ si $P(B) > 0$.
- iii. $P(B|A) = P(B)$ si $P(A) > 0$.

De la definición anterior, se desprende lo siguiente

Si A y B son dos eventos independientes definidos en un espacio de probabilidad dado $(\Omega, \mathfrak{F}, P)$, entonces los siguientes eventos también son independientes

- i. A y B^c ,
- ii. A^c y B ,
- iii. A^c y B^c .

Observación. No debe confundirse los términos **eventos independientes** y **eventos disjuntos**. De hecho, los eventos disjuntos suelen ser muy dependientes por que la ocurrencia de uno implica la no ocurrencia del otro. El único evento que es independiente y ajeno es el vacío \emptyset .

La noción de eventos independientes puede ser extendido a más de dos eventos como se sigue;

Definición 2.13 (Independencia de varios eventos). Para un espacio de probabilidad dado $(\Omega, \mathfrak{F}, P)$, sean A_1, A_2, \dots, A_n eventos en \mathfrak{F} . Los eventos A_1, A_2, \dots, A_n se definen como independientes si y solo si $P\left(\bigcap_{i=1}^n A_i\right) = \prod_{i=1}^n P(A_i)$

2.5. Variables aleatorias

Hasta el momento se conoce cómo asignar probabilidades a eventos del espacio muestral, sin embargo en la práctica esto no siempre es posible ya que sería complicado mencionar o enumerar todos los elementos del espacio muestral.

Por esta razón es necesario “traducir” dichos eventos a números reales. Esto es posible mediante el uso de *variables aleatorias*.

Definición 2.14 (Variable aleatoria). Para un espacio de probabilidad dado $(\Omega, \mathfrak{F}, P)$, una variable aleatoria, denotada por X o $X(\cdot)$, es una función con dominio Ω y contradominio la recta real. La función $X(\cdot)$ debe ser tal si $\omega \in \Omega$ entonces $X(\omega) \in \mathbb{R}$. Si $B \subset \mathbb{R}$ entonces $X^{-1}(B) \in \mathfrak{F}$, donde

$$X^{-1}(B) = \{\omega \in \Omega | X(\omega) \in B\}.$$

Existen dos tipos de variables aleatorias: discretas y continuas. Las variables aleatorias discretas toman sus valores en un conjunto finito o numerable, por ejemplo, el conjunto de los números naturales \mathbb{N} . A este conjunto de valores se le conoce como conjunto de valores posibles o D_X . Las variables aleatorias continuas, por el contrario, toman sus valores en el conjunto de los números reales \mathbb{R} .

2.5.1. Función de distribución

Para describir el comportamiento de una variable aleatoria, se debe conocer cómo se comportan sus probabilidades, esto puede realizarse mediante la *función de distribución*.

Definición 2.15 (Función de distribución acumulada). La función de distribución acumulada de una variable aleatoria X , denotada por $F_X(\cdot)$, se define como aquella función con dominio la recta real y contradominio el intervalo $[0, 1]$, que satisface $F_X(x) = P(X \leq x) = P(\omega : X(\omega) \leq x)$ para cada número real x .

Una función de distribución definida es única para cada variable y siempre existirá, es importante conocerla por que con ella se pueden calcular probabilidades de la variable aleatoria.

A continuación, se presentan ejemplos y propiedades de la función de distribución acumulada.

Ejemplo

Se considera el experimento de lanzar una moneda. Supongamos que la moneda es justa. Sea X el número de caras obtenidas. Entonces,

$$F_X(x) = \begin{cases} 0 & \text{si } x < 0 \\ \frac{1}{2} & \text{si } 0 \leq x < 1 \\ 1 & \text{si } 1 \leq x \end{cases}$$

O $F_X(x) = \frac{1}{2}I_{[0,1)}(x) + I_{[1,\infty)}(x)$ en nuestra notación de función indicadora.

2.5.1.1. Propiedades de la función de distribución acumulada.

- i. $F_X(-\infty) \equiv \lim_{x \rightarrow -\infty} F_X(x) = 0$, y $F_X(+\infty) \equiv \lim_{x \rightarrow +\infty} F_X(x) = 1$.
- ii. $F_X(\cdot)$ es una función monótona creciente; es decir, para toda $a < b$ entonces $F_X(a) \leq F_X(b)$.
- iii. $F_X(\cdot)$ es continua por la derecha, esto es $\lim_{0 < h \rightarrow 0} F_X(x + h) = F_X(x)$.

Definición 2.16 (Función de distribución acumulada). Cualquier función $F(\cdot)$ con dominio la recta real y contradominio el intervalo $[0, 1]$, que satisface las tres propiedades mencionadas anteriormente, se define como una función de distribución acumulada.

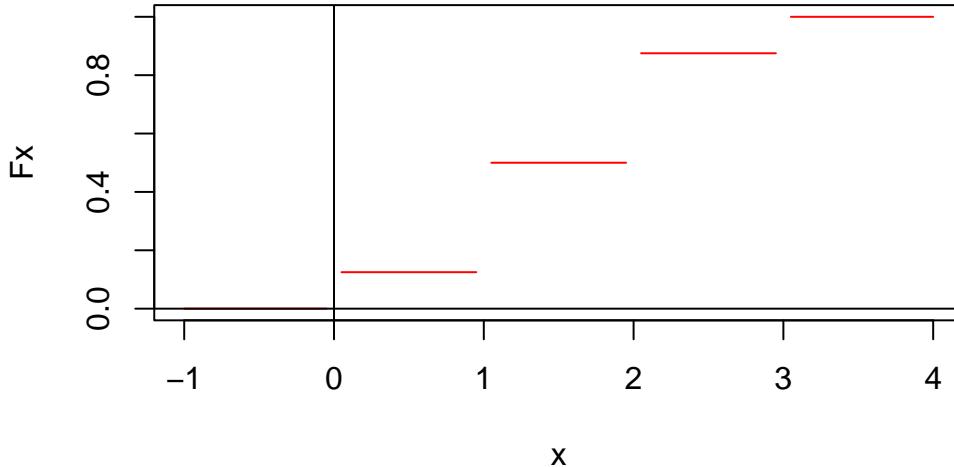
Ejemplo (*Lanzar 3 monedas*)

Considérese el ejemplo del lanzamiento de tres monedas. Sea X el número de águilas en tres lanzamientos.

La función de distribución es la siguiente:

$$F_X(x) = \begin{cases} 0 & \text{si } x < 0 \\ \frac{1}{8} & \text{si } 0 \leq x < 1 \\ \frac{1}{2} & \text{si } 1 \leq x < 2 \\ \frac{7}{8} & \text{si } 2 \leq x < 3 \\ 1 & \text{si } 3 \leq x \end{cases}$$

A continuación se muestra una gráfica de la función



Por ejemplo, la probabilidad de que se obtenga 0 a 1 agujas es $\frac{1}{8}$.

Ejemplo (*Duración de una llamada telefónica*)

Para las variables aleatorias continuas, la forma de la función de distribución es un poco distinta, pero sigue cumpliendo las mismas propiedades.

A continuación se representa una función de distribución acumulada de la variable aleatoria X que podría ser usada para modelar la duración de las llamadas telefónicas.

$$F_X(x) = \begin{cases} 0 & \text{si } x \leq 0 \\ 1 - e^{-x} & \text{si } 0 < x \end{cases}$$

En este caso, se puede ver que la probabilidad de que la variable aleatoria X sea menor a cero es 0, ya que el soporte de la distribución son los números reales positivos.

Como puede apreciarse, conociendo la función de distribución es posible obtener las probabilidades de cualquier evento, simplemente evaluando los valores en la función, por ejemplo:

- $P(X < 2) = 1 - e^{-2} = 0.8646$
- $P(X > 5) = 1 - (1 - e^{-5}) = 0.0067$
- $P(1 < X < 3) = P(X < 3) - P(X < 1) = 0.3180$

Observación. Se debe tener cuidado cuando se calcula probabilidades de variables aleatorias discretas ya que en general no es lo mismo $P(X < x)$ que $P(X \leq x)$.

2.5.2. Función de densidad

Otra función relacionada con las variables aleatorias es la función de densidad.

A diferencia de la función de distribución, esta función es distinta según si la variable aleatoria es discreta o continua. Primero se definirá para el caso discreto y posteriormente para el caso continuo.

2.5.3. Variables aleatorias discretas

Definición 2.17 (Variable aleatoria discreta). Se definirá una variable aleatoria X como discreta si el rango de X es numerable. Si una variable aleatoria X es discreta, entonces su función de distribución acumulada correspondiente $F_X(\cdot)$ se definirá como discreta.

Definición 2.18 (Función de densidad discreta de una variable aleatoria discreta). Si X es una variable aleatoria discreta con $D_x = x_1, x_2, \dots$ entonces la función, denominada por $f_X(\cdot)$ y definida por

$$f_X(x) = \begin{cases} P(X = x_j) & \text{si } x \in D_x \\ 0 & \text{cualquier otro caso.} \end{cases}$$

es la función de densidad discreta de X .

Observación. En ocasiones se usa la función indicadora $I_{D_x}(x) = 1$ si $x \in D_x$ y $I_{D_x}(x) = 0$ si $x \notin D_x$ para expresar la función de densidad en una sola línea.

Teorema 2.5. Sea X una variable aleatoria discreta. $F_X(\cdot)$ puede ser obtenido a partir de $f_X(\cdot)$, y viceversa.

Definición 2.19 (Función de densidad discreta). Cualquier función $f(\cdot)$ con dominio \mathbb{R} y contradominio $[0, 1]$ se define como una *función de densidad discreta* si para algún conjunto contable $D = \{x_1, x_2, \dots\}$ se cumple lo siguiente;

- i. $f(x_j) > 0$ para $j = 1, 2, \dots$
- ii. $f(x) = 0$ para $x \neq x_j$ con $j = 1, 2, \dots$
- iii. $\sum_D f(x_j) = 1$

2.5.4. Variables aleatorias continuas

Definición 2.20 (Variable aleatoria continua). Se llama variable aleatoria continua a X si existe una función $f_X(\cdot)$ tal que $F_X(x) = \int_{-\infty}^x f_X(u)du$ para cada número real x . La función de distribución acumulada $F_X(\cdot)$ de una variable aleatoria continua X se llama *absolutamente continua*.

Definición 2.21 (Función de densidad de una variable aleatoria continua). Si X es una variable aleatoria continua, la función $f_X(\cdot)$ en $F_X(x) = \int_{-\infty}^x f_X(u)du$ se llama *función de densidad* de X .

Observación. En ocasiones se usa la función indicadora $I_{(a,b)}(x) = 1$ si $x \in (a, b)$ y $I_{(a,b)}(x) = 0$ si $x \notin (a, b)$ para expresar la función de densidad en una sola línea.

Teorema 2.6. *Si X es una variable aleatoria continua, entonces $F_X(\cdot)$ se puede obtener a partir de una función $f_X(\cdot)$, y viceversa.*

Las demostraciones del Teorema 2.5 y Teorema 2.6 pueden ser consultados en (Mood, Graybill, y Boes 1986).

Ejemplo (*Duración de una llamada telefónica*)

Usando el teorema fundamental del cálculo, se puede probar la propiedad anterior. Se ilustrará para el ejemplo de la duración de llamadas telefónicas.

Supóngase que la función de distribución para modelar la duración de las llamadas telefónicas es

$$F_X(x) = \begin{cases} 0 & \text{si } x \leq 0 \\ 1 - e^{-x} & \text{si } 0 < x \end{cases}$$

Se observa que $F_X(x)$ está definida en dos partes, por lo que la función no es *absolutamente continua* en cero, por lo que solo será diferenciable en el intervalo de los reales positivos.

$$f_X(x) = \frac{dF_X(x)}{dx} = \frac{d(1 - e^{-x})}{dx} = -\frac{de^{-x}}{dx} = e^{-x}$$

es decir;

$$f_X(x) = e^{-x} I_{(0,\infty)}(x).$$

Por otro lado, se observa que

$$F_X(x) = \int_{-\infty}^x e^{-u} du = \int_0^x e^{-u} du = 1 - e^{-x},$$

es decir

$$F_X(x) = 1 - e^{-x} I_{(0, \infty)}(x)$$

De esta manera, se comprueba la propiedad.

Observación. Se utilizará el término “función de densidad” sin el modificador de “discreta” o “de probabilidad” para representar cualquier tipo de densidad.

2.6. Vectores aleatorios

Definición 2.22 (Vector aleatorio n -dimensional). Sean X_1, X_2, \dots, X_n variables aleatorias reales definidas sobre el mismo espacio de probabilidad $(\Omega, \mathfrak{F}, P)$. La función $\mathbf{X} : \Omega \rightarrow \mathbb{R}^n$ definida como

$$\mathbf{X}(\omega) = (X_1(\omega), \dots, X_n(\omega))$$

se denomina vector aleatorio n -dimensional.

Definición 2.23 (Distribución de un vector aleatorio). Sea \mathbf{X} un vector aleatorio n -dimensional. La medida de probabilidad definida por

$$P_{\mathbf{X}}(B) := P(\mathbf{X} \in B); B \in \mathcal{B}_n$$

donde \mathcal{B}_n representa la sigma álgebra del Borel sobre \mathbb{R}^n , es denominada la *distribución* del vector aleatorio \mathbf{X} .

Definición 2.24 (Función de masa de probabilidad conjunta). Sea $\mathbf{X} = (X_1, X_2, \dots, X_n)$ un vector aleatorio de n dimensiones. Si las variables aleatorias X_i , donde $i = 1, \dots, n$, son todas discretas, se afirma que el vector aleatorio \mathbf{X} es discreto. En esta situación, la función de densidad de \mathbf{X} , también conocida como la *función de densidad conjunta* de las variables aleatorias X_1, X_2, \dots, X_n , queda definida por

$$p_{\mathbf{X}}(x) := \begin{cases} P(\mathbf{X} = x) & \text{si } x \text{ pertenece a la imagen de } \mathbf{X} \\ 0 & \text{en caso contrario.} \end{cases}$$

Definición 2.25 (Función de distribución acumulada conjunta). Sea $\mathbf{X} = (X_1, X_2, \dots, X_n)$ un vector aleatorio de n dimensiones. La función definida por

$$F(x_1, x_2, \dots, x_n) := P(X_1 \leq x_1, X_2 \leq x_2, \dots, X_n \leq x_n),$$

para todo $(x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ recibe el nombre de función de distribución acumulativa conjunta de las variables aleatorias X_1, X_2, \dots, X_n , o simplemente la función de distribución del vector aleatorio n -dimensional \mathbf{X} .

3. Estadística

La estadística tiene un origen que se remonta a tiempos antiguos, cuando las civilizaciones antiguas recopilaban y analizaban datos para tomar decisiones informadas en diversas áreas. Sin embargo, fue en el siglo XVII cuando los trabajos de pensadores como John Graunt y William Petty sentaron las bases para los métodos estadísticos modernos. Graunt realizó estudios sobre la mortalidad y estableció principios de recopilación de datos, mientras que Petty aplicó el análisis estadístico a la economía y la demografía.

En la era moderna, con el advenimiento de la computación y la disponibilidad de grandes volúmenes de datos, la estadística ha cobrado una importancia aún mayor. Técnicas avanzadas como el análisis de series de tiempo, la regresión múltiple, el análisis de componentes principales y el aprendizaje automático han transformado la forma en que se aborda la predicción de datos. Estas herramientas permiten modelar relaciones complejas y patrones ocultos en los datos, lo que es crucial para la toma de decisiones en áreas como el marketing, la medicina, la economía y la planificación empresarial.

(Mann 2010) presenta dos significados para la palabra “estadística”. En el sentido más común, la estadística hace referencia a hechos numéricos. El segundo significado de estadística se relaciona con el campo o disciplina de estudio. Bajo esta perspectiva, la estadística se define de la siguiente manera

Definición 3.1 (Estadística). Una estadística es una función de variables aleatorias observables, la cual es en sí misma una variable aleatoria observable y no contiene ningún parámetro desconocido.

3.1. Datos univariados

En esta sección se definirán conceptos básicos de la estadística univariada. Se comienza con los siguientes conceptos;

Definición 3.2 (Población). Una *población* consiste en todos los elementos (individuos, elementos u objetos) cuyas características se están estudiando. La población que se está estudiando también se denomina *población objetivo*.

Definición 3.3 (Parámetro). Un *parámetro* es una característica numérica o descriptiva de una población o probabilidad distribución.

3.1.1. Valor Esperado y momentos

Un concepto sumamente útil en problemas que implican variables aleatorias o distribuciones es el de la esperanza (valor esperado). En esta subsección, se presentan definiciones y resultados relacionados con la esperanza.

Definición 3.4 (Media). Sea X una variable aleatoria. La *media* de X , denotada como μ_X o $E[X]$, se define de la siguiente manera:

$$E[X] = \sum x_j f_X(x_j)$$

Si X es discreta con puntos de densidad $x_1, x_2, \dots, x_j, \dots$

$$E[X] = \int_{-\infty}^{\infty} x f_X(x) dx$$

Si X es continua con una función de densidad de probabilidad $f_X(x)$

$$E[X] = \int_0^{\infty} [1 - F_X(x)] dx - \int_{-\infty}^0 F_X(x) dx$$

para cualquier variable aleatoria X .

Observación. $E[X]$ representa el centro de gravedad (o centroide) de la región unitaria determinada por la función de densidad de X . De esta manera, la media de X proporciona una medida de la ubicación central de los valores de la variable aleatoria X .

La media de una variable aleatoria X es una medida de ubicación central de la densidad de X . La varianza de una variable aleatoria X es una medida de la dispersión o propagación de la densidad de X .

Definición 3.5 (Varianza). Sea X una variable aleatoria, y se define μ_X como $E[X]$, la *varianza* de X , denotada como σ_X^2 o $Var[X]$, se define de la siguiente manera

$$Var[X] = \sum_j (x_j - \mu_X)^2 f_X(x_j)$$

si X es discreta con puntos $x_1, x_2, \dots, x_j, \dots$

$$Var[X] = \int_{-\infty}^{\infty} (x - \mu_X)^2 f_X(x) dx$$

Si X es continua con una función de densidad de probabilidad $f_X(x)$

$$\text{Var}[X] = \int_0^\infty 2x[1 - F_X(x) + F_X(-x)]dx - \mu_X^2$$

para una variable aleatoria X arbitraria.

Se vio que una media era el centro de gravedad de una densidad; de manera similar, la varianza representa el momento de inercia de la misma densidad con respecto a un eje perpendicular que pasa por el centro de gravedad.

Definición 3.6 (Desviación estándar). Si X es una variable aleatoria, la *desviación estándar* de X , denotada por σ_X , se define como $+\sqrt{\text{Var}[X]}$.

La desviación estándar de una variable aleatoria, al igual que la varianza, es una medida de la dispersión o propagación de los valores de la variable aleatoria. En muchas aplicaciones, es preferible a la varianza como medida, ya que tendrá las mismas unidades de medida que la propia variable aleatoria.

3.1.1.1. Valor esperado de una función de una variable aleatoria

Se definió la esperanza de una variable aleatoria arbitraria X , llamada la media de X . En esta subsección, se definirá la esperanza de una función de una variable aleatoria para variables aleatorias discretas o continuas

Definición 3.7 (Valor esperado). Sea X una variable aleatoria y $g(\cdot)$ una función con dominio y codominio en la recta real. La esperanza o valor esperado de la función $g(\cdot)$ de la variable aleatoria X , denotada por $E[g(X)]$, se define de la siguiente manera:

$$E[g(X)] = \sum_j g(x_j)f_X(x_j) \quad (3.1)$$

Si X es discreta con puntos $x_1, x_2, \dots, x_j, \dots$ (siempre que esta serie sea absolutamente convergente).

$$E[g(X)] = \int_{-\infty}^{\infty} g(x)f_X(x)dx \quad (3.2)$$

Si X es continua con función de densidad de probabilidad $f_X(x)$ (siempre que $\int_{-\infty}^{\infty} |g(x)|f_X(x)dx < \infty$).

Observación. Si $g(x) = x$, entonces $E[g(X)] = E[X]$ es la media de X . Si $g(x) = (x - \mu_X)^2$, entonces $E[g(X)] = E[(X - \mu_X)^2] = \text{Var}[X]$.

Teorema 3.1. A continuación se presentan las propiedades del valor esperado,

- i. $E[c] = c$ para una constante c .
- ii. $E[cg(X)] = cE[g(X)]$ para una constante c .
- iii. $E[c_1g_1(X) + c_2g_2(X)] = c_1E[g_1(X)] + c_2E[g_2(X)]$.
- iv. $E[g_1(X)] \leq E[g_2(X)]$ si $g_1(x) \leq g_2(x)$ para todo x .

Teorema 3.2. Si X es una variable aleatoria, entonces

$$\text{Var}[X] = E[(X - E[X])^2] = E[X^2] - (E[X])^2, \quad (3.3)$$

siempre que $E[X^2]$ existe.

Las pruebas de los teoremas anteriores se pueden consultar en (Mood, Graybill, y Boes 1986).

3.1.1.2. Momentos

Los momentos de una variable aleatoria o de una distribución son los valores esperados de las potencias de la variable aleatoria que tiene la distribución dada.

Definición 3.8 (Momento). Si X es una variable aleatoria, el r -ésimo momento de X , generalmente denotado por μ'_r , se define como

$$\mu'_r = E[X^r]$$

si el valor esperado existe.

Note que $\mu'_1 = E[X] = \mu_X$, que es la media de X .

Definición 3.9 (Cuantil). El q -ésimo cuantil de una variable aleatoria X o de su distribución correspondiente se denota como ξ_q y se define como el número más pequeño ξ que cumple con la condición $F_X(\xi) \geq q$.

Si X es una variable aleatoria continua, entonces el q -ésimo cuantil de X se calcula como el número más pequeño ξ que cumple con la condición $F_X(\xi) = q$.

Definición 3.10 (Mediana). La mediana de una variable aleatoria X , denotada como med_X , $\text{med}(x)$ o $\xi_{0.5}$, es el cuantil 0.5.

3.1.2. Muestreo

Definición 3.11 (Muestra). Una porción de la población seleccionada para el estudio es conocida como una *muestra*.

Una muestra puede ser aleatoria o no aleatoria. En una muestra aleatoria, cada elemento de la población tiene la posibilidad de ser incluido en la muestra. Sin embargo, en una muestra no aleatoria este puede no ser el caso.

Definición 3.12 (Muestra aleatoria). Sean X_1, X_2, \dots, X_n variables aleatorias con una densidad conjunta $f_{(X_1, \dots, X_n)}(\cdot, \dots, \cdot)$ que se descompone de la siguiente manera:

$$f_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) = f(x_1)f(x_2) \cdots f(x_n),$$

donde $f(\cdot)$ es la densidad (común) de cada X_i . Entonces, se define que X_1, X_2, \dots, X_n es una *muestra aleatoria* de tamaño n de una población con densidad $f(\cdot)$.

Definición 3.13 (Media Muestral). El primer momento de la muestra es la *media muestral*, definida como

$$\bar{X} = \bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i,$$

donde X_1, X_2, \dots, X_n es una muestra aleatoria de una densidad $f(\cdot)$. \bar{X} es una función de las variables aleatorias X_1, \dots, X_n , y por lo tanto, en teoría se puede determinar la distribución de \bar{X} .

Definición 3.14 (Varianza muestral). Sea X_1, X_2, \dots, X_n una muestra aleatoria con densidad $f(\cdot)$, entonces

$$S_n^2 = S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2 \quad \text{para } n > 1$$

se define como la *varianza muestral*.

Teorema 3.3. Sea X_1, X_2, \dots, X_n una muestra aleatoria con densidad $f(\cdot)$, la cual tiene una media μ y una varianza finita σ^2 , y sea $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$. Luego,

$$\mathbb{E}[\bar{X}] = \mu_{\bar{X}} = \mu \quad \text{y} \quad \text{Var}[\bar{X}] = \sigma_{\bar{X}}^2 = \frac{1}{n} \sigma^2.$$

Prueba. Vea Mood, Graybill, y Boes (1986). □

Teorema 3.4 (Teorema del límite central). *Supongamos que $f(\cdot)$ es una densidad con media μ y varianza finita σ^2 . Si \bar{X}_n es el promedio de una muestra aleatoria de tamaño n extraída de $f(\cdot)$ y definimos la variable aleatoria Z_n como*

$$Z_n = \frac{\bar{X}_n - \mathbb{E}[\bar{X}_n]}{\sqrt{\text{Var}[\bar{X}_n]}} = \frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}}$$

Entonces, la distribución de Z_n se acerca a la distribución normal estándar a medida que n tiende a infinito.

Prueba. Vea Wackerly et al. (2009). □

3.2. Datos multivariados

Cuando dos o más variables aleatorias son observadas en miembros de una muestra aleatoria, los datos resultantes se denominan datos multivariados. El caso especial de dos variables se refiere como datos bivariados.

Ejemplo (Calificaciones finales)

Considere los datos en la Tabla 3.1 que representan una muestra de 30 estudiantes en una universidad grande que fueron asignados al azar a un curso de Introducción a la Informática. En la Tabla se muestran las puntuaciones del cuestionario, además de la calificación del examen final de cada estudiante.

Tabla 3.1.: Puntuaciones y calificación final.

Est.	Puntuación Final	Est.	Puntuación Final	Est.	Puntuación Final
1	7.4	79.8	11	7.6	80.7
2	8.4	82.0	12	8.8	94.5
3	8.8	76.1	13	6.1	50.1
4	6.4	62.7	14	7.2	68.3
5	10.0	98.2	15	6.6	64.4
6	5.5	43.0	16	7.0	67.2
7	7.3	76.5	17	5.3	53.9
8	5.9	61.4	18	7.9	78.8
9	7.1	78.5	19	8.1	85.7
10	7.9	88.7	20	7.6	81.7
				Suma	222.6
					2253.2

Los valores medios y las desviaciones estándar se proporcionan a continuación. Las puntuaciones del cuestionario están en el vector x y las puntuaciones del examen final están en y . Se tienen los siguientes resultados:

$$\text{Media}(x) = 7.42$$

$$\text{Desviación est\'andar}(x) = 1.15$$

$$\text{Media}(y) = 75.11$$

$$\text{Desviación est\'andar}(y) = 13.15$$

Los histogramas de estas dos variables se muestran en la Figura 3.1. Allí se puede observar que los histogramas de ambas variables tienen una forma aproximadamente en campana, con las puntuaciones del examen final ligeramente sesgadas hacia la izquierda. Al examinar el histograma en la Figura 3.1(b), se observa que la media de las puntuaciones del examen final parece ser de alrededor de 75, lo cual es coherente con los resultados anteriores. La mayoría de las puntuaciones están entre 60 y 90, con algunas por encima de 90 y algunas por debajo de 60.

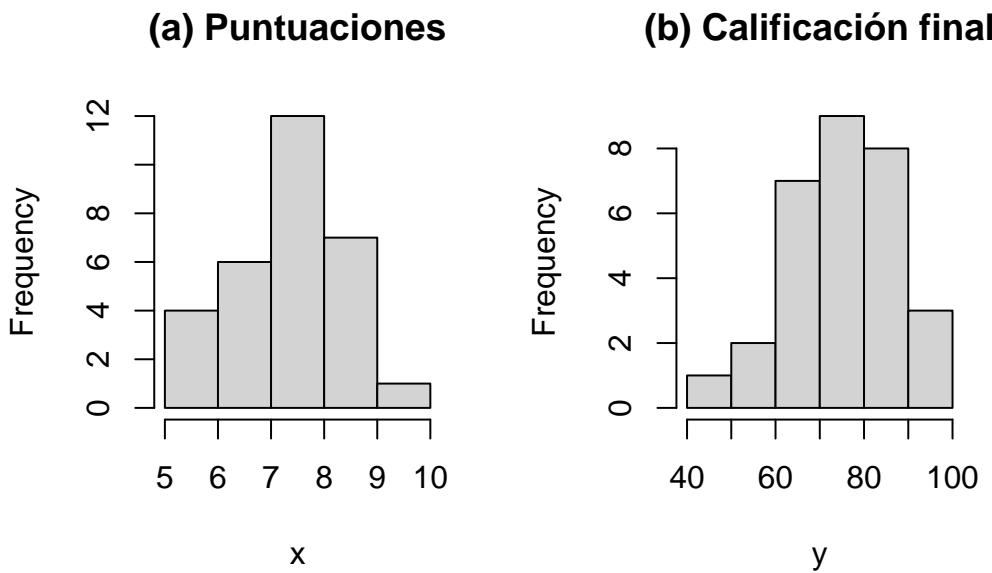


Figura 3.1.: Puntajes y calificaciones finales para una muestra aleatoria de 30 estudiantes inscritos al curso.

Los histogramas, las medias y las desviaciones estándar no proporcionan información sobre cómo dos variables están relacionadas entre sí. Como ejemplo, un instructor

probablemente quisiera saber si los estudiantes que les fue bien en el cuestionario también tendieron a desempeñarse bien en el examen final, y viceversa. También podría querer saber si algunos estudiantes que les fue mal en el cuestionario mejoraron drásticamente su calificación en el examen final. Los histogramas y las estadísticas de muestra mostrados anteriormente no responden a estas preguntas.

Lo que se necesita son medidas de la relación entre las dos variables. Los parámetros poblacionales más comunes utilizados para medir tales relaciones son la covarianza, γ_{XY} , y la correlación, ρ_{XY} .

Definición 3.15 (Covarianza). Si X e Y son dos variables aleatorias definidas en el mismo espacio de probabilidad, la *covarianza* de X e Y , denotada como $\text{Cov}[X, Y]$ o γ_{XY} , se define como

$$\gamma_{XY} = \text{Cov}[X, Y] = E[(X - \mu_X)(Y - \mu_Y)]$$

siempre que la esperanza indicada exista.

Definición 3.16 (Correlación). La *correlación*, denotada como $\rho[X, Y]$ o ρ_{XY} , de las variables aleatorias X e Y , se define como

$$\rho_{XY} = \frac{\gamma_{XY}}{\sigma_X \sigma_Y}$$

siempre que γ_{XY}, σ_X y σ_Y existan, con $\sigma_X, \sigma_Y > 0$.

Técnicamente, la covarianza es el valor esperado (o promedio teórico) del producto cruzado $(X - \mu_X)(Y - \mu_Y)$. Es una medida de cómo dos variables “se mueven juntas”. Para facilitar la interpretación, generalmente se usa la correlación, que es una versión “estandarizada” de la covarianza que tiene la propiedad $-1 \leq \rho_{XY} \leq 1$ para cualquier par de variables aleatorias X e Y .

Observación.

- a. $\gamma_{XX} = E[(X - \mu_X)(X - \mu_X)] = E[(X - \mu_X)^2] = \sigma_X^2$.
- b. $\rho_{XX} = \frac{\sigma_X^2}{\sigma_X \sigma_X} = 1$.

Corolario 3.1. Si X e Y son independientes, entonces $\gamma_{XY} = 0$.

Prueba. Observe que

$$\text{Cov}[X, Y] = E[(X - \mu_X)(Y - \mu_Y)] = E[X - \mu_X]E[Y - \mu_Y] = 0$$

Ya que $E[(X - \mu_X)] = 0$. □

Teorema 3.5. Sean X e Y variables aleatorias definidas sobre el mismo espacio de probabilidad tal que $E(X^2) < \infty$ y $E(Y^2) < \infty$. Entonces:

- i. $\text{Cov}[X, Y] = E[XY] - E[X]E[Y]$.
- ii. $\text{Cov}[X, Y] = \text{Cov}[Y, X]$.
- iii. $\text{Var}[X] = \text{Cov}[X, X]$.
- iv. $\text{Cov}[aX + b, Y] = a\text{Cov}[X, Y]$ para cualquier $a, b \in \mathbb{R}$.

Prueba. Vea Castañeda, Arunachalam, y Dharmaraja (2014). □

Definición 3.17 (Variables aleatorias no correlacionadas). Las variables aleatorias X e Y se definen como no correlacionadas si y solo si $\text{Cov}[X, Y] = 0$.

Observación. La afirmación contraria al corolario anterior no siempre es cierta; es decir, $\text{Cov}[X, Y] = 0$ no siempre implica que X e Y sean independientes.

4. Procesos estocásticos

Los procesos estocásticos desempeñan un papel fundamental en la modelización y análisis de una amplia gama de fenómenos en diversas disciplinas, desde la física hasta la economía. Estos procesos son esenciales para comprender y predecir el comportamiento de sistemas que involucran aleatoriedad y variabilidad en el tiempo.

Un proceso estocástico (o probabilístico) puede considerarse una generalización de una muestra aleatoria, en el sentido de que las variables aleatorias **no son necesariamente independientes** y su distribución podría cambiar. (Castañeda, Arunachalam, y Dharmaraja 2014) proporciona la siguiente definición para proceso estocástico.

Definición 4.1 (Proceso estocástico). Un *proceso estocástico* real es una colección de variables aleatorias $\{X_t; t \in T\}$ definida en un espacio de probabilidad común $(\Omega, \mathfrak{F}, P)$ con valores en \mathbb{R} . T se le llama al conjunto índice del proceso o espacio paramétrico, que generalmente es un subconjunto de \mathbb{R} . El conjunto de valores que la variable aleatoria X_t puede tomar se denomina *espacio de estado del proceso* y es denotado por S .

De la definición anterior se entiende que las variables *dependerán* del parámetro t (usualmente el tiempo) y están ordenadas. Además el conjunto T puede ser discreto o continuo. Si el conjunto de índices T es discreto, se le conoce como *proceso estocástico de tiempo discreto* mientras que si T es continuo, se le conoce como *proceso estocástico de tiempo continuo*. Las variables aleatorias X_t pueden tomar tanto valores continuos o valores discretos. Note que si se fija un punto t , se tiene X_t una variable aleatoria (Ross 1995).

Definición 4.2 (Trayectoria). La asignación definida para cada $\omega \in \Omega$ fijo,

$$\begin{aligned} X(\omega) : T &\rightarrow S \\ t &\mapsto X_t(\omega) \end{aligned}$$

se denomina una trayectoria de muestra del proceso a lo largo del tiempo o una realización del proceso estocástico.

Definición 4.3 (Proceso completamente especificado). Se dice que un proceso estocástico $\{X(t) : t \in T\}$ está completamente especificado si para cualquier valor del tiempo $t_1 < t_2 < \dots < t_n$ con $n \in \mathbb{N}$, la distribución conjunta de $(X_{t_1}, X_{t_2}, \dots, X_{t_n})$ es conocida.

Definición 4.4 (Proceso estocástico con incrementos independientes). Si, para todo $t_0, t_1, t_2, \dots, t_n$ tal que $t_0 < t_1 < t_2 < \dots < t_n$, las variables aleatorias $X_{t_0}, X_{t_1} - X_{t_0}, X_{t_2} - X_{t_1}, \dots, X_{t_n} - X_{t_{n-1}}$ son independientes (o de manera equivalente, $X_{t+\tau} - X_\tau$ es independiente de X_s para $s < \tau$), entonces se dice que el proceso $\{X_t; t \in T\}$ es un *proceso con incrementos independientes*.

Observación. No resulta pertinente afirmar que X_{t_1} sea inferior a X_{t_2} dado que las variables aleatorias no se encuentran ordenadas.

Definición 4.5 (Proceso estocástico con incrementos estacionarios). Se dice que un proceso estocástico $X_t; t \in T$ tiene *incrementos estacionarios* si $X_{t_2+\tau} - X_{t_1+\tau}$ tiene la misma distribución que $X_{t_2} - X_{t_1}$ para todas las elecciones de t_1, t_2 y $\tau > 0$.

Definición 4.6 (Proceso estacionario). Si para cualquier conjunto de t_1, t_2, \dots, t_n arbitrarios, tal que $t_1 < t_2 < \dots < t_n$, las distribuciones conjuntas de las variables aleatorias vectoriales $(X_{t_1}, X_{t_2}, \dots, X_{t_n})$ y $(X_{t_1+h}, X_{t_2+h}, \dots, X_{t_n+h})$ son iguales para todo $h > 0$, entonces se dice que el proceso estocástico $\{X_t, t \in T\}$ es un proceso estacionario estocástico de orden n (o simplemente un *proceso estacionario*). El proceso estocástico $\{X_t, t \in T\}$ se dice que es un *proceso estocástico fuertemente estacionario* o *estrictamente estacionario* si la propiedad anterior se cumple para todo n .

Ejemplo

Suponga que $\{X_n; n \geq 1\}$ es una secuencia de variables aleatorias independientes e idénticamente distribuidas. Se define la secuencia $\{Y_n, n \geq 1\}$ como

$$Y_n = X_n + aX_{n-1}$$

donde a es una constante real. Entonces, es fácil observar que $\{Y_n; n \geq 1\}$ es estrictamente estacionaria.

Definición 4.7 (Proceso de segundo orden). Un proceso estocástico $\{X_t; t \in T\}$ se dice un *proceso de segundo orden* si

$$\mathbb{E}((X_t)^2) < \infty$$

para todo $t \in T$.

Definición 4.8 (Proceso estacionario por covarianza). Un proceso estocástico de segundo orden $\{X_t, t \in T\}$ se denomina *estacionario por covarianza* o *estacionario débil* si su función de media $m(t) = \mathbb{E}[X_t]$ es independiente de t y su función de covarianza $\text{Cov}(X_s, X_t)$ depende únicamente de la diferencia $|t - s|$ para todos $s, t \in T$. Es decir:

$$\text{Cov}(X_s, X_t) = f(|t - s|).$$

Ejemplo

Sea $\{X_n; n \geq 1\}$ un conjunto de variables aleatorias no correlacionadas con media cero y varianza uno. Entonces, la covarianza $\text{Cov}(X_m, X_n) = E(X_m X_n)$ es igual a 0 si $m \neq n$ y 1 si $m = n$. Esto demuestra que $\{X_n, n \geq 1\}$ es un proceso estacionario por covarianza.

Definición 4.9 (Proceso evolutivo). Un proceso estocástico que no es estacionario (en ningún sentido), se dice ser un proceso estocástico evolutivo.

Parte II.

Series de tiempo

5. Series de Tiempo

Las series temporales tienen su origen en la necesidad de comprender y predecir patrones presentes en datos secuenciales a lo largo del tiempo. A lo largo de la historia, diversas civilizaciones han registrado observaciones cronológicas con el propósito de anticipar fenómenos naturales, eventos económicos y otros procesos cambiantes.

No obstante, el enfoque más sistemático en el análisis de series temporales se inició en el ámbito de la estadística y la econometría durante el siglo XX. Figuras pioneras como Norbert Wiener y Andrey Kolmogorov establecieron los cimientos teóricos en torno a los procesos estocásticos.

La relevancia de las series temporales en la predicción de datos radica en su habilidad para capturar patrones temporales y tendencias presentes en conjuntos de datos. A medida que se acumulan datos a lo largo del tiempo, es posible identificar relaciones y ciclos que facilitan la realización de pronósticos futuros. Esto resulta especialmente valioso en campos como la economía, la meteorología, la epidemiología y las finanzas, donde comprender los patrones temporales es crucial para tomar decisiones informadas.

En la actualidad, con el advenimiento de la computación y las técnicas de análisis más avanzadas, las series temporales han adquirido una importancia aún mayor. Modelos matemáticos y estadísticos avanzados, como los modelos ARIMA (Media Móvil Integrada Autoregresiva) y las redes neuronales recurrentes, permiten analizar y predecir series temporales con mayor precisión y complejidad. Estos modelos son esenciales para la toma de decisiones estratégicas en diversas industrias, ya que ayudan a anticipar tendencias, identificar patrones estacionales y enfrentar la incertidumbre en el futuro.

En términos generales, una serie temporal puede considerarse como una recopilación de observaciones realizadas secuencialmente en el tiempo. El interés de este análisis no recae en las series que son deterministas, sino en aquellas cuyos valores se comportan siguiendo las leyes de la probabilidad. Se discutirán los principios fundamentales involucrados en el análisis estadístico de series temporales. Para comenzar, se debe prestar una atención más meticulosa a la definición de serie temporal, dado que en realidad es un tipo particular de proceso estocástico.

5.1. Conceptos básicos y manipulación de series de tiempo

Definición 5.1 (Serie de tiempo). Un proceso estocástico $X(t); t \in T$ se define como una colección de variables aleatorias, donde T es un conjunto de índices para el cual todas las variables aleatorias, $X(t)$, donde t pertenece a T , están definidas en el mismo espacio muestral. Cuando T representa el tiempo, se hace referencia al proceso estocástico como una *serie de tiempo*.

Si T toma un rango continuo de valores (por ejemplo, $T = (-\infty, \infty)$ o $T = (0, \infty)$) , el proceso se dice que es un *proceso de parámetro continuo*. Si, por otro lado, T toma un conjunto discreto de valores (por ejemplo, $T = \{0, 1, 2, \dots\}$ o $T = \{0, \pm 1, \pm 2, \dots\}$), el proceso se dice que es un *proceso de parámetro discreto*. De hecho, es común referirse a estos como procesos continuos y discretos, respectivamente.

Se utilizará la notación de subíndice, X_t , cuando se esté tratando específicamente con un proceso de parámetro discreto. Sin embargo, cuando el proceso involucrado sea de parámetro continuo o de tipo no especificado, se utilizará la notación de función, $X(t)$. Además, cuando no haya confusión, a menudo se utiliza la notación $\{X(t)\}$ o simplemente $X(t)$ para denotar una serie de tiempo. De manera similar, a menudo se acortará $\{X_t; t = 0, \pm 1, \dots\}$ a $X_t, t = \{0, \pm 1, \dots\}$ o simplemente se usará X_t .

Nótese que una variable aleatoria, γ , es una función definida en un espacio muestral Ω cuyo rango son los números reales. Un valor observado de la variable aleatoria γ es un número real $y = \gamma(\omega)$ para algún $\omega \in \Omega$. Para una serie de tiempo $\{X(t)\}$, su “valor” $\{X(t, \omega); t \in T\}$ para algún $\omega \in \Omega$ fijo es una colección de números reales. Esto lleva a la siguiente definición.

Definición 5.2 (Realización). Una *realización* de la serie de tiempo $\{X(t); t \in T\}$ es el conjunto de resultados de valores reales, $\{X(t, \omega); t \in T\}$ para un valor fijo de $\omega \in \Omega$.

La colección de todas las posibles realizaciones se denomina *conjunto*, y, para un valor dado de t , la expectativa de la variable aleatoria $X(t)$ se denomina *media del conjunto* y se denotará como $E[X(t)] = \mu(t)$. La *varianza* de $X(t)$ se expresa como

$$\text{Var}[X(t)] := E[(X(t) - \mu(t))^2]$$

y a menudo se denota como $\sigma^2(t)$ ya que también puede depender de t .

De especial interés en el análisis de una serie temporal es la covarianza entre $X(t_1)$ y $X(t_2)$, donde $t_1, t_2 \in T$. Dado que esta es la covarianza (Definición 3.15) dentro de la misma serie temporal, se refiere a ella como *autocovarianza*. De igual manera, se refiere a correlación (Definición 3.16) como *autocorrelación* y se denotan por

$$\gamma(t_1, t_2) := E[(X(t_1) - \mu(t_1))(X(t_2) - \mu(t_2))] \tag{5.1}$$

y

$$\rho(t_1, t_2) := \frac{\gamma(t_1, t_2)}{\sigma(t_1)\sigma(t_2)} \quad (5.2)$$

respectivamente.

5.1.1. Series de tiempo estacionarias

En el estudio de una serie de tiempo, es común que solo se tenga disponible una única realización de la serie. El análisis de una serie temporal basado únicamente en una realización es análogo a analizar las propiedades de una variable aleatoria en función de una sola observación. Los conceptos de estacionariedad y ergodicidad jugarán un papel importante en mejorar la capacidad de análisis de una serie temporal basada en una única realización de manera efectiva. Un proceso se considera estacionario si está en un estado de “equilibrio estadístico”. El comportamiento básico de dicha serie de tiempo no cambia con el tiempo. Como ejemplo, para dicho proceso, $\mu(t)$ no dependería del tiempo y, por lo tanto, podría ser denotado como μ para todo t . Parecería que, dado que $x(t)$ para cada $t \in T$ proporciona información sobre la media del conjunto, μ , podría ser posible estimar μ en función de una única realización. Un proceso **ergódico** es aquel para el cual promedios de conjunto como μ pueden estimarse consistentemente a partir de una sola realización. En esta sección, se presentarán definiciones más formales de estacionariedad. La noción más restrictiva de estacionariedad es la de estacionariedad estricta, que se define de la siguiente manera.

Definición 5.3 (Proceso estrictamente estacionario). Se dice que un proceso $\{X(t); t \in T\}$ es estrictamente estacionario si para cualquier $t_1, t_2, \dots, t_k \in T$ y cualquier $h \in T$, la distribución conjunta de $\{X(t_1), X(t_2), \dots, X(t_k)\}$ es idéntica a la de $\{X(t_1 + h), X(t_2 + h), \dots, X(t_k + h)\}$.

La estacionariedad estricta requiere, entre otras cosas, que para cualquier $t_1, t_2 \in T$, las distribuciones de $X(t_1)$ y $X(t_2)$ deben ser las mismas, y además que todas las distribuciones bivariadas de pares $\{X(t), X(t+h)\}$ sean iguales para todos los h , etc. El requisito de estacionariedad estricta es riguroso y suele ser difícil de establecer matemáticamente. De hecho, para la mayoría de las aplicaciones, las distribuciones involucradas no se conocen. Por esta razón, se han desarrollado nociones menos restrictivas de estacionariedad. La más común de ellas es la estacionariedad por covarianza.

Definición 5.4 (Estacionariedad por covarianza). La serie de tiempo $\{X(t); t \in T\}$ se considera estacionaria por covarianza si

- i. $\mu_{x_t} = E[X(t)] = \mu$ (media constante para todo t)

- ii. $\sigma_{x_t}^2 = \text{Var}[X(t)] = \sigma^2 < \infty$ (es decir, una constante finita para todo t)
- iii. $\gamma_{x_{t_1}, x_{t_2}}$ y $\rho_{x_{t_1}, x_{t_2}}$ depende solo de $t_2 - t_1$.

Si se cumple la condición iii., no habrá confusión al reemplazar la notación, $\gamma_{x_{t_1}, x_{t_2}}$, con $\gamma_{t_1 - t_2}$, y de manera similar, al denotar $\rho_{x_{t_1}, x_{t_2}}$ como $\rho_{t_1 - t_2}$. Cuando se establece $t_2 - t_1 = k$, se hace referencia a γ_k y ρ_k como la autocovarianza y la autocorrelación con un rezago de k , respectivamente.

La función de autocovarianza de una serie de tiempo estacionaria satisface las siguientes propiedades:

- i. $\gamma_0 = E[(X_t - \mu)(X_t - \mu)] = E[(X_t - \mu)^2] = \sigma^2$.
- ii. $|\gamma_k| \leq \gamma_0$ para todo k .
- iii. $\gamma_k = E[(X_{t-k} - \mu)(X_t - \mu)] = E[(X_t - \mu)(X_{t-k} - \mu)] = \gamma_{-k}$.
- iv. La función γ_k es semidefinida positiva. Esto es, para cualquier conjunto de puntos de tiempo $t_1, t_2, \dots, t_k \in T$ y para los reales b_1, b_2, \dots, b_k , se tiene

$$\sum_{i=1}^k \sum_{j=1}^k \gamma(t_i - t_j) b_i b_j \geq 0.$$

La función de autocorrelación satisface las siguientes propiedades análogas:

- i. $\rho_0 = 1$.
- ii. $|\rho_k| \leq 1$ para todo k .
- iii. $\rho_k = \rho_{-k}$.
- iv. La función ρ_k es semidefinida positiva, y para series de tiempo discretas definidas en $t = 0, 1, 2, \dots$, la matriz

$$\rho_k = \begin{pmatrix} 1 & \rho_1 & \dots & \rho_k \\ \rho_1 & 1 & \dots & \rho_{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_k & \rho_{k+1} & \dots & 1 \end{pmatrix}$$

es semidefinida positiva para cada k .

Observación. La estacionariedad por covarianza también se conoce como estacionariedad débil, estacionariedad en el sentido amplio y estacionariedad de segundo orden. En el resto de esta tesis, a menos que se especifique lo contrario, el término estacionariedad se referirá a la estacionariedad por covarianza.

En las series de tiempo, al igual que en la mayoría de las otras áreas de la estadística, los datos no correlacionados desempeñan un papel importante. No hay dificultad en definir dicho proceso en el caso de una serie temporal de parámetro discreto. Es decir, la serie temporal $\{X_t; t = 0, \pm 1, \pm 2, \dots\}$ se llama “proceso puramente aleatorio” si los X_t son variables aleatorias no correlacionadas. Al considerar procesos puramente aleatorios, solo nos interesarán el caso en el que los X_t también estén distribuidos de manera idéntica. En esta situación, es más común referirse a la serie de tiempo como ruido blanco. La siguiente definición resume estas observaciones.

Definición 5.5 (Proceso de ruido blanco). Se dice que un proceso X_t es ruido blanco si se cumplen las siguientes condiciones.

1. Los X_t están distribuidos de manera idéntica.
2. $\gamma_{t_2-t_1} = 0$ cuando $t_2 \neq t_1$.
3. $\gamma_{t-t} = \sigma^2$, donde $0 < \sigma^2 < \infty$.

En un proceso de ruido blanco, cada observación está no correlacionada con todas las demás observaciones. Un hecho importante es que los procesos de ruido blanco son estacionarios.

5.1.1.1. Estimación de los parámetros de un proceso estacionario.

5.1.1.1.1. Estimación de μ .

Dada la realización $\{x_t, t = 1, 2, \dots, n\}$ de una serie temporal estacionaria, la estimación natural de la media común μ es la media muestral.

$$\bar{x} = \frac{1}{n} \sum_{t=1}^n x_t \quad (5.3)$$

Es evidente que el estimador \bar{X} es imparcial para μ .

Para una serie temporal estacionaria, se puede emplear los datos a lo largo del tiempo para estimar la media, dado que se asume que la media es constante para cada instante de tiempo, t .

5.1.1.1.1. Ergodicidad de X .

Se dice que X es ergódico para μ si X converge en el sentido de la media cuadrática hacia μ a medida que n aumenta, es decir, si $\lim_{n \rightarrow \infty} E[(\bar{X} - \mu)^2] = 0$.

Teorema 5.1. *Sea $\{X_t; t = 0, \pm 1, \pm 2, \dots\}$ una serie de tiempo estacionaria. Entonces,*

$\bar{X} = \frac{1}{n} \sum_{t=1}^n X_t$ *es ergódica para μ si y solo si*

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{j=0}^{k-1} \gamma_j = 0. \quad (5.4)$$

Prueba. Vea Yaglom (1962). □

Corolario 5.1. *Sea X_t es una serie de tiempo estacionaria con parámetros discretos, como se establece en el Teorema 5.1 . Entonces, \bar{X} es ergódico para μ si*

$$\lim_{k \rightarrow \infty} \gamma_k = 0, \quad (5.5)$$

o equivalentemente si

$$\lim_{k \rightarrow \infty} \rho_k = 0. \quad (5.6)$$

La condición suficiente para la ergodicidad de \bar{X} , dada en el Corolario 5.1 , resulta muy útil y es una condición que se cumple para la amplia clase de series temporales autorregresivas de media móvil, ARMA(p, q) , estacionarias, que se discutirán más adelante. A pesar de que X_t 's “cercanos” en el tiempo pueden tener una correlación sustancial, la condición en el Corolario 5.1 asegura que para una “gran” separación, están casi no correlacionados.

5.1.1.1.2. Varianza de \bar{X} .

Antes de abandonar el tema de estimar μ a partir de una realización de un proceso estacionario, en el Teorema 5.2 se proporciona una fórmula útil para la Varianza de \bar{X} .

Teorema 5.2. *Si X_t es una serie de tiempo estacionaria, entonces la varianza de \bar{X} basada en una realización de longitud n está dada por*

$$\text{Var}(\bar{X}) = \frac{\sigma_X^2}{n} \left(1 + 2 \sum_{k=1}^{n-1} \left(1 - \frac{|k|}{n} \right) \rho_k \right) \quad (5.7)$$

El resultado en la Ecuación 5.7 muestra el efecto de la autocorrelación en la varianza de \bar{X} , y si X_t es ruido blanco, es decir, $\gamma_k = 0$ si $k \neq 0$, entonces la Ecuación 5.7 se convierte en el conocido resultado $Var(\bar{X}) = \sigma^2/n$.

Utilizando la notación $\hat{\rho}_k$ para denotar las autocorrelaciones estimadas (muestra) y $\hat{\sigma}^2 = \hat{\gamma}_0$ para denotar la varianza muestral, es práctica común obtener intervalos de confianza aproximados del 95% para μ usando

$$\left(\bar{X} - 1.96 \sqrt{\frac{\hat{\sigma}^2}{n} \sum_{k=-(n-1)}^{n-1} \left(1 - \frac{|k|}{n} \right) \hat{\rho}_k}, \bar{X} + 1.96 \sqrt{\frac{\hat{\sigma}^2}{n} \sum_{k=-(n-1)}^{n-1} \left(1 - \frac{|k|}{n} \right) \hat{\rho}_k} \right). \quad (5.8)$$

5.1.1.1.2. Estimación de γ_k .

Debido a la estacionariedad, $E[(X_t - \mu)(X_{t+k} - \mu)] = \gamma_k$ no depende de t . Como consecuencia, parece razonable estimar γ_k a partir de una sola realización, por

$$\begin{aligned} \hat{\gamma}_k &= \frac{1}{n} \sum_{t=1}^{n-k} (X_t - \bar{X})(X_{t+k} - \bar{X}), \quad 0 \leq k \leq n \\ &= 0, \quad k \geq n \\ &= \hat{\gamma}_{-k}, \quad k < 0 \end{aligned} \quad (5.9)$$

Observación. Usando Ecuación 5.9 se deduce que

$$\hat{\gamma}_0 = \frac{1}{n} \sum_{t=1}^n (X_t - \bar{X})^2 \quad (5.10)$$

5.1.1.1.3. Estimación de ρ_k .

Definición 5.6 (Autocorrelación muestral). El estimador de la autocorrelación, ρ_k se obtiene mediante

$$\hat{\rho}_k = \hat{\gamma}_k / \hat{\gamma}_0 \quad (5.11)$$

A este estimador se le conoce como la autocorrelación muestral.

A partir del examen de la Ecuación 5.9, es evidente que los valores de $\hat{\gamma}_k$ (y $\hat{\rho}_k$) tenderán a ser “pequeños” cuando k sea grande en relación con n .

5.1.2. Conjuntos de datos de series temporales.

Los comportamientos exhibidos por los datos de series temporales son diversos y se analizarán en las secciones subsiguientes. Dichos tipos de comportamiento se ilustrarán mediante ejemplos provenientes de la realidad, tales como los datos intrigantes de manchas solares, registros de temperatura, el índice Dow Jones, precios de acciones individuales, datos de ventas (tanto mensuales como diarios), entre otros. El análisis comenzará con una discusión sobre datos que presenten algún tipo de patrón cíclico (Woodward, Sadler, y Robertson (2022)).

5.1.2.1. Datos Cílicos

Muchos conjuntos de datos de series temporales muestran un patrón cíclico, lo que significa que los datos presentan aumentos y disminuciones de manera algo repetitiva. Estos datos a veces se denominan “*pseudo-periódicos*”, un término que usaremos de manera sinónima con “*cíclico*”. Los datos de manchas solares en la Figura 5.1 es un ejemplo de datos cílicos.

Observación. Los datos verdaderamente periódicos exhiben un comportamiento que se replica de manera precisa a lo largo de un período de tiempo establecido. Un caso ilustrativo de datos puramente periódicos se encuentra en la forma de la curva sinusoidal. De este modo, los datos pseudoperiódicos (o cílicos) se refieren a aquellos conjuntos de datos que tienden a mostrar repeticiones en sus comportamientos.

Ejemplo 5.1.

Datos de manchas solares

La Figura 5.1 muestra datos anuales de manchas solares para los años 1700-2020. Las manchas solares son áreas de explosiones solares o disturbios atmosféricos extremos en el sol. En 1848, el astrónomo suizo Rudolf Wolf introdujo un método para contar la actividad de las manchas solares, y los datos mensuales utilizando su método están disponibles desde 1749. (Waldmeier (1961)).

```
library(tswge)
plot(sunspot2.0, xlab='Año', ylab='Manchas solares')
```

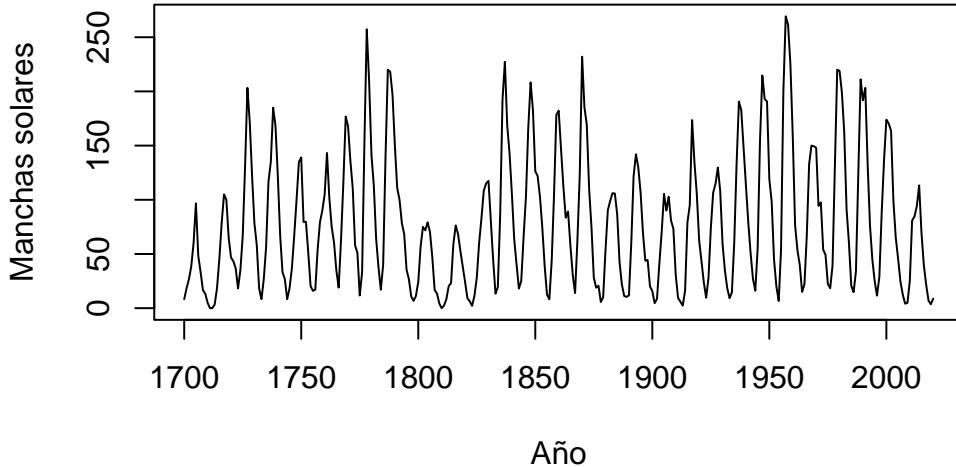


Figura 5.1.: Número anual de manchas solares desde 1700 hasta 2020.

Las manchas solares han generado un considerable interés en la comunidad científica por dos razones principales:

- La actividad de las manchas solares tiende a afectarnos aquí en la Tierra. Por ejemplo, una alta actividad de manchas solares provoca interferencias en la comunicación por radio y se asocia con una mayor intensidad de luz ultravioleta y actividad de auroras boreales.
- La actividad de las manchas solares tiene un comportamiento cíclico que tiene una duración de ciclo de aproximadamente 11 años. Al examinar la Figura 5.1 se observa que hay 29 ciclos en los 321 años, con una duración media del ciclo de aproximadamente 11 años. De hecho, las duraciones de los ciclos tienden a variar aleatoriamente entre 9 y 13 años.

Si bien el comportamiento cíclico en la Figura 5.1 es claro, a menudo es útil examinar fragmentos cortos de los datos para visualizar mejor el comportamiento específico. La Figura 5.2 muestra el número de manchas solares desde 1867 hasta 1950. Las líneas verticales identifican los años en los que hubo un pico en los números de manchas solares y las flechas horizontales representan el tiempo entre los picos.

Para los años representados en la Figura 5.2, las duraciones de los ciclos fueron de 13, 10, 12, 12, 11, 9 y 10 años, respectivamente. Las duraciones de los ciclos parecen variar aleatoriamente y no parece haber un “ajuste a una duración de ciclo fija”. De hecho, según la comprensión de estos autores, los científicos no tienen una explicación física para el ciclo de aproximadamente 11 años. Los datos de las manchas solares son un ejemplo clásico de datos cíclicos con duraciones de ciclo variables. De hecho, Yule (1971) desarrolló el proceso autorregresivo como un medio para describir el comportamiento periódico “perturbado” de los datos de las manchas solares.

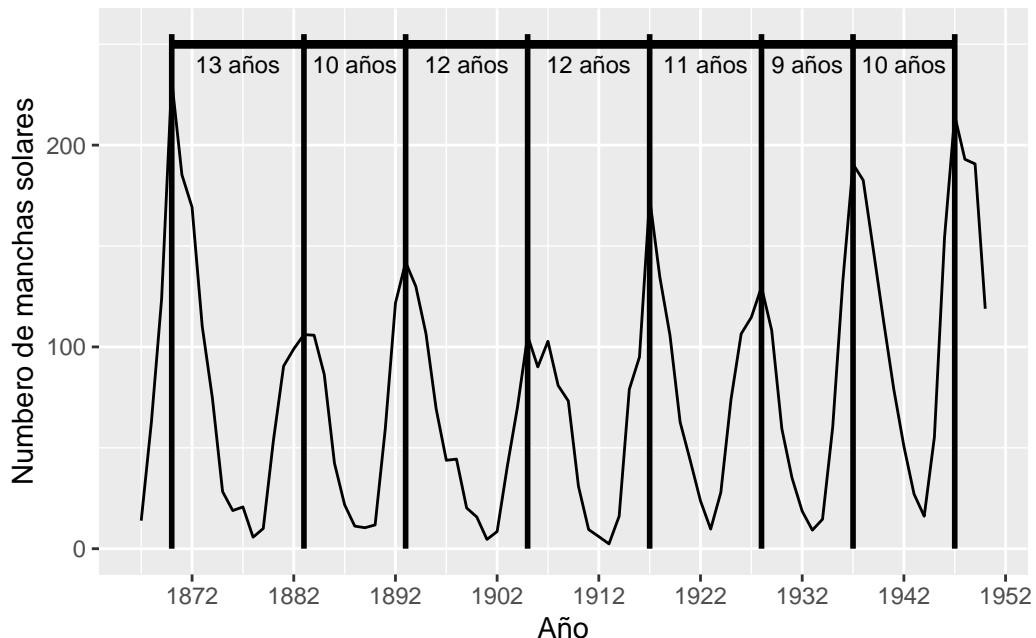


Figura 5.2.: Fragmento de la Figura 5.1 que muestra los años 1867-1950

Ejemplo 5.2.

Datos de pasajeros aéreos

La Figura 5.3 es un conjunto de datos que contiene el número total (en miles) de pasajeros de líneas aéreas internacionales por mes durante los 12 años, desde 1949 hasta 1960. Estos datos han sido analizados exhaustivamente y son un conjunto de datos clásico en la literatura de series temporales. Los datos siguen un patrón cíclico de 12 meses que es similar de un año a otro y está basado en el año calendario. Por lo tanto, los datos de Pasajeros Aéreos son otro ejemplo de datos estacionales. Además, los datos tienden a mostrar una tendencia al alza con el tiempo. Es decir, el número

de pasajeros de líneas aéreas está aumentando con el tiempo. El comportamiento de tendencia en series temporales se discutirá en la Sección 5.1.2.2. También hay una variabilidad creciente dentro del año. Este tipo de comportamiento, conocido como *estacionalidad multiplicativa*.

```
library(tswge)
plot(AirPassengers, xlab='Año', ylab='Número de pasajeros')
```

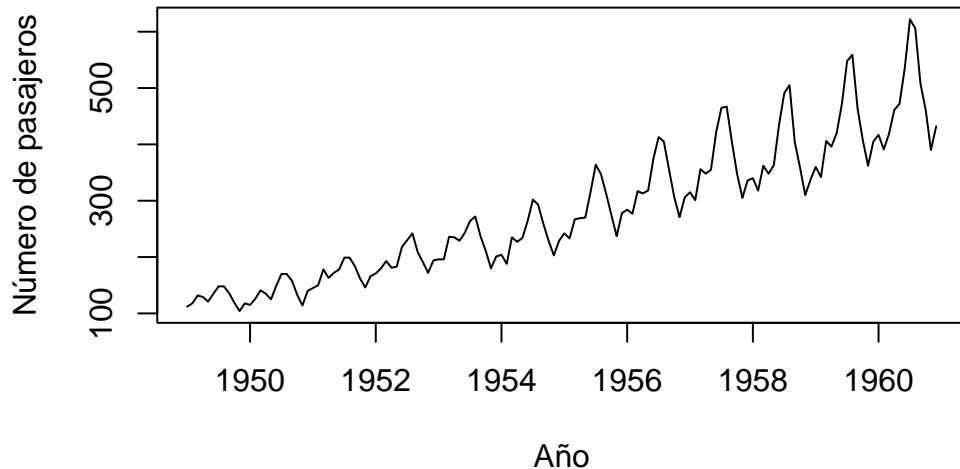


Figura 5.3.: Número de Pasajeros Internacionales en Aerolíneas de 1949 a 1960

La Figura 5.4 muestra un fragmento de los datos de pasajeros aéreos desde 1957 hasta 1960. Se observa que el viaje aéreo es ligero desde Enero hasta Abril, aumenta durante los meses de verano y comienza a disminuir en septiembre hasta noviembre con un ligero aumento en diciembre. Este patrón, aunque no es sinusoidal, se repite de un año a otro. El comportamiento cíclico de los datos de pasajeros aéreos se repite anualmente y es un ejemplo de datos estacionales que no son seudosenoidales.

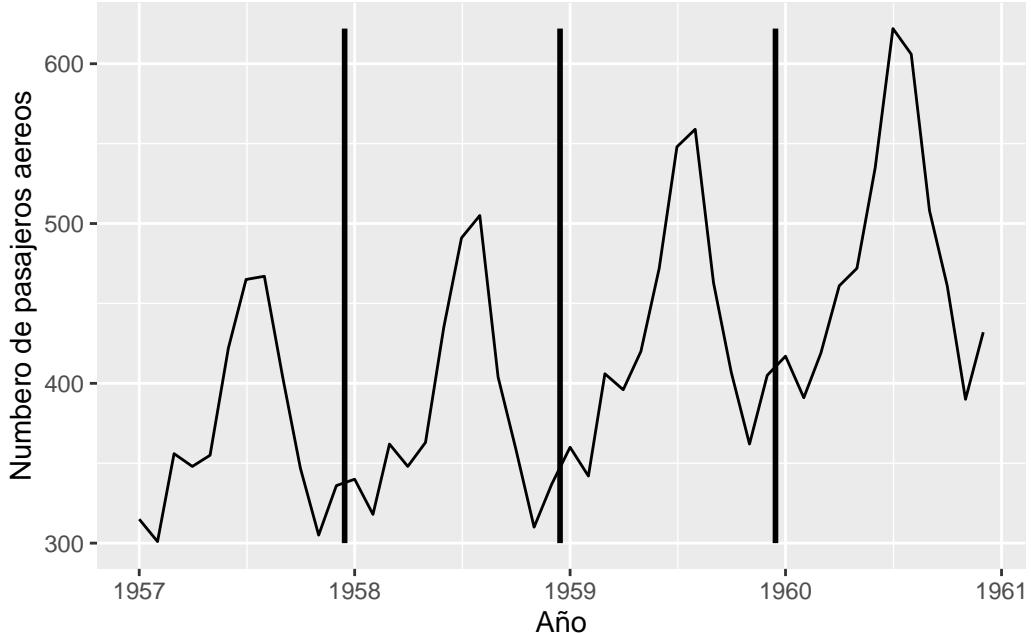


Figura 5.4.: Fragmento de la Figura 5.3 que muestra los años 1957-1960

5.1.2.2. Tendencias

Una tendencia en un contexto de análisis de datos se refiere a la inclinación de una serie de datos a experimentar un incremento o disminución constante a lo largo del tiempo. En el caso específico de los datos de Pasajeros Aéreos mostrados en la Figura 5.3, se identifica un patrón de crecimiento además del patrón estacional previamente observado. Una tendencia lineal se caracteriza por el aumento o la disminución de los datos de manera constante y progresiva, tal como se ilustra en la Figura 5.5a. Las tendencias pueden seguir una curva, como lo ejemplifica la tendencia exponencial en la Figura 5.5b. Por otro lado, la Figura 5.5c exhibe una serie temporal con una tendencia descendente, pero su naturaleza es más irregular en comparación con las tendencias representadas en las Figuras (a) y (b). Un patrón común en conjuntos de datos es un comportamiento de tendencia aleatoria, como se muestra en la Figura 5.5d, la cual sugiere una trayectoria sin un rumbo definido. Esto implica que pueden existir tendencias de corta o larga duración, en ocasiones en direcciones opuestas.

Definición 5.7. La función $g(t)$ es *periódica* con periodo (o longitud del ciclo) $p > 0$ si p es el valor más pequeño tal que $g(t) = g(t + kp)$ para todo t y enteros k . Se dice que una función $g(t)$ es *aperiódica* si no existe tal p .

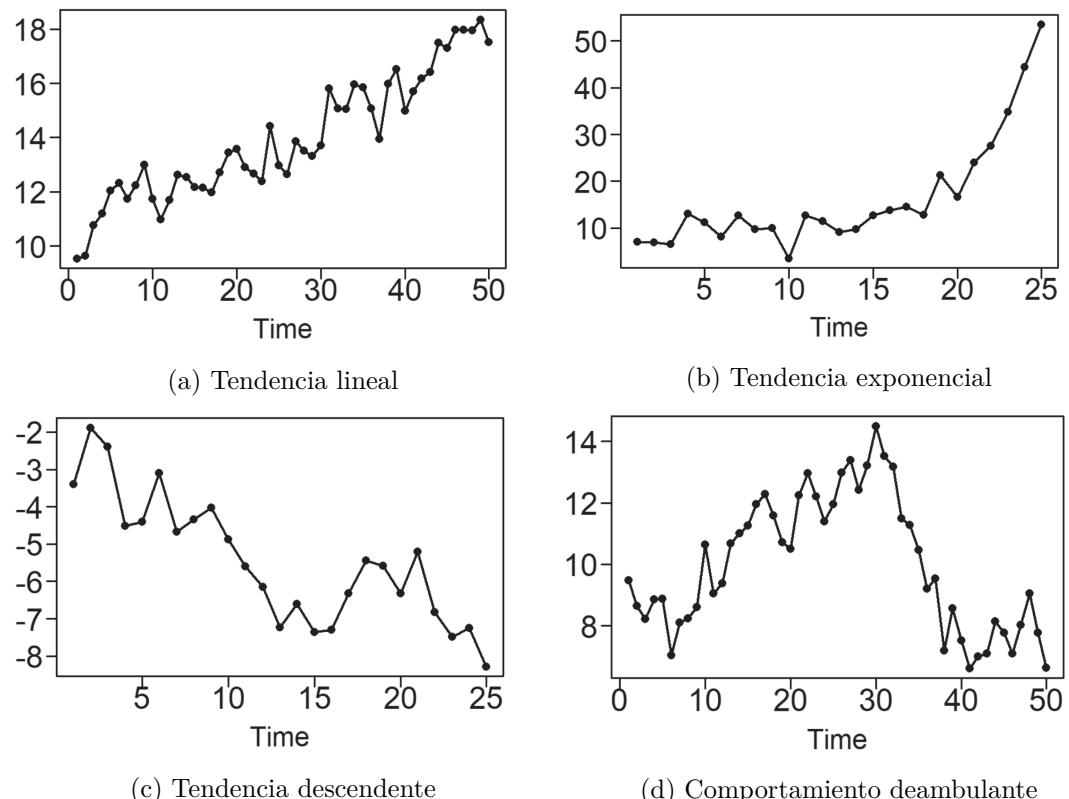


Figura 5.5.: Gráficos que muestran (a) una tendencia lineal, (b) una tendencia exponencial, (c) una tendencia descendente irregular y (d) un patrón de deambulación.

Definición 5.8 (Frecuencia). La frecuencia, denotada por f , puede ser descrita de las siguientes dos maneras;

1. $f = 1/\text{periodo}$ (tamaño del ciclo)
2. El número de ciclos en la función a través de una unidad de tiempo.

Observación. Los datos con comportamiento de tendencia y deambulación aleatoria no son cílicos por naturaleza. A veces se les llama aperiódicos debido a la ausencia de un comportamiento regular de ascenso y descenso.

5.1.2.3. Definición y propiedades del espectro y densidad espectral

Definición 5.9. Sea X_t una serie de tiempo estacionaria con autocovarianza γ_k y autocorrelación ρ_k . Entonces para $|f| \leq 0.5$:

- i. El *espectro* de X_t se define como

$$P_x(f) = \sum_{k=-\infty}^{\infty} e^{-2\pi ifk} \gamma_k. \quad (5.12)$$

- ii. La *densidad espectral* de X_t se define como

$$S_x(f) = \sum_{k=-\infty}^{\infty} e^{-2\pi ifk} \rho_k. \quad (5.13)$$

Usando la fórmula de Euler, se obtienen las fórmulas “más agradables”.

$$P_x(f) = \sigma_x^2 + 2 \sum_{k=1}^{\infty} \gamma_k \cos(2\pi fk)$$

Y

$$S_x(f) = 1 + 2 \sum_{k=1}^{\infty} \rho_k \cos(2\pi fk)$$

Estas fórmulas enfatizan que el espectro y la densidad espectral son funciones de valor real, lo que no es evidente en Ecuación 5.12 y Ecuación 5.13.

Propiedades importantes de densidades spectrales

- i. $S_x(f) \geq 0$.

$$\text{ii. } S_x(f) = S_x(-f).$$

$$\text{iii. } S_x(f) = 1 + 2 \sum_{k=1}^{\infty} \rho_k \cos(2\pi f k), \text{ donde } |f| \leq 0.5.$$

$$\text{iv. } \sum_{-0.5}^{0.5} S_x(f) e^{2\pi i f k} df = \rho_k$$

Las propiedades i y ii muestran que $S_x(f)$ es una función par no negativa.

5.1.3. Suavizado de datos de series temporales.

Existen varios métodos para “suavizar” el comportamiento ruidoso (posiblemente poco importante) de una serie temporal para que se pueda entender mejor una señal importante subyacente. Se comienza discutiendo el método de suavizado de promedio móvil centrado, que es el más básico.

5.1.3.1. Suavizado de datos utilizando un suavizador de promedio móvil centrado

El suavizador de promedio móvil centrado es un método para reemplazar los valores de datos en una serie temporal con un promedio de los valores de datos que rodean (e incluyen) ese punto de datos. Por ejemplo, un suavizador de promedio móvil centrado de orden tres reemplaza un valor de datos x_t en el tiempo t con $s_t = (x_{t-1} + x_t + x_{t+1})/3$. Es decir, se asigna el valor promedio al punto de tiempo medio. Por lo tanto, un suavizador de promedio móvil centrado de orden tres no puede asignarse al primer o último punto de tiempo de una serie temporal. Se sigue que a mayor orden, más valores faltarán al principio y al final del conjunto de datos suavizado. Para un promedio móvil centrado de tercer orden, la fórmula de promediado se desplaza a lo largo del conjunto de datos de la serie temporal, considerando tres valores de datos consecutivos juntos hasta llegar a los últimos tres puntos temporales.

Definición 5.10 (Suavizador de Promedio Móvil Centrado). Sea $x_t, t = 1, \dots, n$ un conjunto de datos de series temporales. El suavizador de promedio móvil centrado se define de la siguiente manera:

Caso 1: m es un número impar

Sea $k = (m - 1)/2$. Para $k < t < n - k$, el valor de los datos suavizados, s_t , en el tiempo t se da por

$$s_t = \frac{1}{m} \sum_{i=t-k}^{t+k} x_i \quad (5.14)$$

Caso 2: m es un número par

Sea $k = m/2$. Para $k < t < n - k$, el valor de los datos suavizados, s_t , en el tiempo t se da por

$$s_t = \frac{x_{t-k}}{2m} + \frac{1}{m} \sum_{i=t-k+1}^{t+k-1} x_i + \frac{x_{t+k}}{2m} \quad (5.15)$$

Ejemplos

- Para un promedio móvil centrado de quinto orden en tiempos $2 < t < n - 2$, se tiene

$$s_t = (x_{t-2} + x_{t-1} + x_t + x_{t+1} + x_{t+2})/5.$$

- El suavizador de promedio móvil de cuarto orden en tiempos $2 < t < n - 2$, está dado por

$$s_t = \frac{x_{t-2}}{8} + \frac{x_{t-1} + x_t + x_{t+1}}{4} + \frac{x_{t+2}}{8}$$

El suavizador de promedio móvil centrado tiene dos usos básicos:

- Suavizado diseñado para eliminar fluctuaciones (potencialmente sin sentido) de los datos.
- Eliminar el comportamiento cíclico de datos estacionales u otros datos cíclicos con longitudes de ciclo fijas.

El Ejemplo 5.3 muestra el uso del suavizado de promedio móvil centrado con el propósito de detectar o comprender mejor señales subyacentes y fundamentales en los datos.

Ejemplo 5.3.

Suavizando los datos de temperatura de Tesla y DFW.

Los precios de las acciones de Tesla desde el 1 de enero de 2020 hasta el 30 de abril de 2021 se muestran en la Figura 5.6a. Se observa el hecho de que hubo un aumento constante hasta principios de 2021, momento en el cual el precio se estabilizó y disminuyó. Sin embargo, hay una considerable fluctuación de un día para otro, especialmente en 2021. La Figura 5.6d muestra las temperaturas medias anuales de DFW (Dallas Ft. Worth) desde 1900 hasta 2020. Allí se observa una considerable fluctuación de un año a otro, pero algo de aumento a partir de aproximadamente 2000. La Figura 5.6b, (c), (e) y (f) muestran versiones suavizadas de las Figuras Figura 5.6a y (d). Al utilizar el suavizador de promedio móvil

centrado, las fluctuaciones de un día para otro se suavizan; se nota que el orden 8 produce más suavizado que el orden 3. El comportamiento fundamental, incluida la estabilización y disminución a principios de 2021, se ve con más claridad al minimizar los cambios ruidosos de un día para otro. El efecto del suavizado es más evidente en los datos de temperatura de DFW. La fluctuación de un año a otro es bastante dramática en el conjunto de datos original en la Figura 5.6d . Un suavizado de orden 3 proporciona cierta claridad, pero el suavizado de orden 8 mostrado en la Figura 5.6e muestra claramente un comportamiento casi estable desde 1900 hasta aproximadamente 1985. Desde entonces, ha habido un aumento que puede haberse estabilizado o no en los últimos años. Es particularmente notable en la Figura 5.6e que el suavizado ha eliminado los extremos. Específicamente, las temperaturas extremadamente altas en 2012, 2016 y 2017 se ven moderadas por las temperaturas más bajas en los años circundantes. (Tomado de Woodward, Sadler, y Robertson (2022))

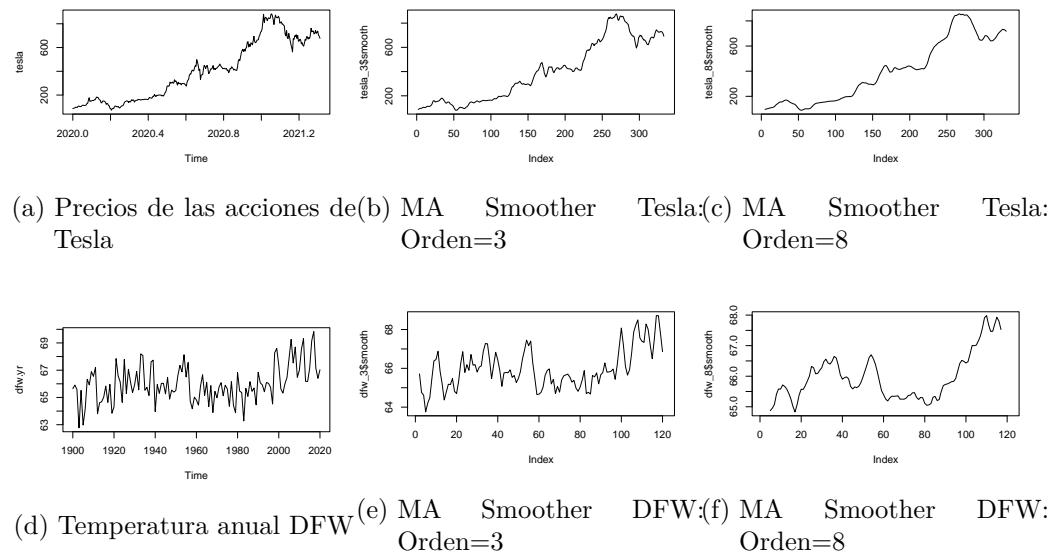


Figura 5.6.: Precios de las acciones de Tesla y datos de temperatura anual de DFW antes y después de aplicar suavizadores de promedio móvil de orden 3 y 8

5.2. Análisis y técnicas de descomposición.

5.2.1. Descomposición de datos estacionales

En el Ejemplo 5.2 se aborda la naturaleza de los datos estacionales, entendidos como una serie de datos cíclicos con períodos consistentes y un patrón que guarda relación con el calendario. El conjunto de datos de AirPassengers presentado en la Figura 5.3 se clasifica como un ejemplo paradigmático de datos estacionales. Este conjunto de datos exhibe un comportamiento estacional anual, además de una tendencia de crecimiento, que se aproxima a ser lineal. Es convencional considerar que los datos estacionales, denotados como x_t , comprenden:

- a. Un componente estacional intrínseco anual, identificado como s_t ,
- b. Un componente de tendencia a largo plazo, referido como tr_t , y
- c. Un componente de variabilidad aleatoria, conocido como z_t .

Se ha observado esta estructura en el conjuntos de datos ya mencionado. Los expertos en análisis de series temporales se enfocan en dos categorías de modelos estacionales:

Datos estacionales aditivos

Los datos x_t , en el tiempo t pueden ser considerados como una suma dada en la Ecuación 5.16

$$x_t = s_t + tr_t + z_t. \quad (5.16)$$

Datos estacionales multiplicativos

Los datos, x_t , en el tiempo t pueden ser expresados como el producto dado en la Ecuación 5.17

$$x_t = s_t \times tr_t \times z_t. \quad (5.17)$$

Ejemplo 5.4.

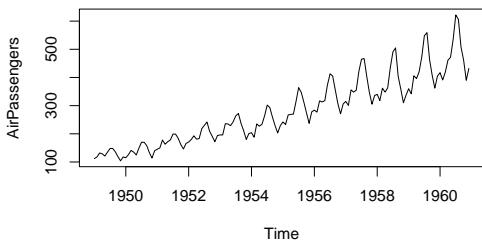
Datos de pasajeros aéreos

Para ilustrar la diferencia entre los tipos de datos que se ajustan mejor a un modelo aditivo y a uno multiplicativo, se utiliza el conjunto de datos de AirPassengers. Como se ha señalado anteriormente, los datos de AirPassengers, representados en la Figura 5.7a, tienen un componente estacional y de tendencia, pero también la variabilidad dentro del año aumenta con el tiempo.

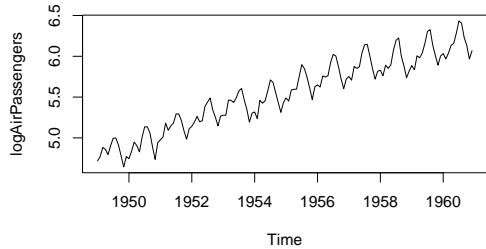
```

library(tswge)
data(AirPassengers)
logAirPassengers=log(AirPassengers)
plot(AirPassengers)
plot(logAirPassengers)

```



(a) Datos de pasajeros aéreos: 1949-1960



(b) Datos de pasajeros aéreos en escala logarítmica

Figura 5.7.: Datos de pasajeros aereos y logaritmo de los datos de pasajeros aereos

Los conjuntos de datos con este tipo de comportamiento suelen modelarse utilizando modelos multiplicativos. Para eliminar el aumento en la variabilidad, los analistas suelen tomar el logaritmo de los datos y utilizan los “datos logarítmicos” para el análisis.

Los datos logarítmicos de logAirPassengers en la Figura 5.7b no muestran un aumento en la variabilidad dentro del año y son un ejemplo clásico de datos que se modelan utilizando el modelo aditivo en la Ecuación 5.16.

Se comenzó discutiendo el modelo aditivo, considerado el más intuitivo de los dos.

A continuación, se discutirán las diferencias en las estrategias de modelado para estos dos conjuntos de datos.

Las descomposiciones aditivas y multiplicativas siguen pasos de implementación similares:

1. Estimar el componente de tendencia.
2. Eliminar el componente de tendencia, lo que resulta en un conjunto de datos compuesto principalmente por las fluctuaciones estacionales en los datos.
3. Calcular un componente estacional “promedio” dentro del año.
4. Encontrar el ruido restante.

Se comenzará discutiendo el modelo aditivo, que es el más intuitivo de los dos.

5.2.1.1. Descomposición aditiva

Cuando se analizan datos utilizando el modelo aditivo en la Ecuación 5.16, se parte del supuesto de que los datos son la suma de componentes estacionales, de tendencia y de ruido aleatorio. Se discuten los pasos de análisis involucrados en la descomposición de los datos logarítmicos de logAirPassengers. En la práctica, los componentes en Ecuación 5.16 se estiman y un modelo estimado puede describirse como

$$x_t = \hat{s}_t + \hat{tr}_t + \hat{z}_t \quad (5.18)$$

Ejemplo 5.5.

Descomposición aditiva de LogAirPassengers

La descomposición de los datos logAirPassengers se logra mediante los siguientes pasos.

- a. **Estimar el Componente de Tendencia:** La Figura 5.8 es una representación gráfica del conjunto de datos logAirPassengers superpuesto con el resultado de aplicar un suavizador de media móvil centrada de orden 12 a los datos.

```
library(tswge)
data(AirPassengers)
logAirPassengers=log(AirPassengers)
logair.12=ma.smooth.wge(logAirPassengers,order=12)
logair.12$smooth
```

```
[1]      NA      NA      NA      NA      NA      NA 4.837280 4.841114
[9] 4.846596 4.851238 4.854488 4.859954 4.869840 4.881389 4.893411 4.904293
[17] 4.912752 4.923701 4.940483 4.957406 4.974380 4.991942 5.013095 5.033804
[25] 5.047776 5.060902 5.073812 5.088378 5.106906 5.124312 5.138282 5.152751
[33] 5.163718 5.171454 5.178401 5.189431 5.203909 5.218093 5.231553 5.243722
[41] 5.257413 5.270736 5.282916 5.292150 5.304079 5.323338 5.343560 5.357427
[49] 5.367695 5.378309 5.388417 5.397805 5.403849 5.407220 5.410364 5.410294
[57] 5.408381 5.406761 5.406218 5.410571 5.419628 5.428330 5.435128 5.442237
[65] 5.450659 5.461103 5.473655 5.489713 5.503974 5.516367 5.529403 5.542725
[73] 5.557864 5.572693 5.587498 5.602730 5.616658 5.631189 5.645937 5.659812
[81] 5.674172 5.687636 5.700766 5.714738 5.727153 5.738856 5.750676 5.760658
[89] 5.770846 5.780430 5.788745 5.796524 5.804821 5.814072 5.823075 5.832692
[97] 5.842665 5.853541 5.864863 5.875490 5.885654 5.894475 5.901555 5.907026
[105] 5.910012 5.910708 5.911637 5.913829 5.917360 5.922887 5.926146 5.927563
[113] 5.929657 5.930458 5.932964 5.938377 5.946188 5.956352 5.967813 5.977291
```

```
[121] 5.985269 5.994078 6.003991 6.014899 6.026589 6.040709 6.054492 6.066195
[129] 6.073088 6.080733 6.091930 6.102013 6.112511 6.121153 6.128381 6.137437
[137] 6.145733 6.151526      NA      NA      NA      NA      NA      NA      NA
```

```
logair.sm12=ts(logair.12$smooth,start=c(1949,1),frequency=12)
```

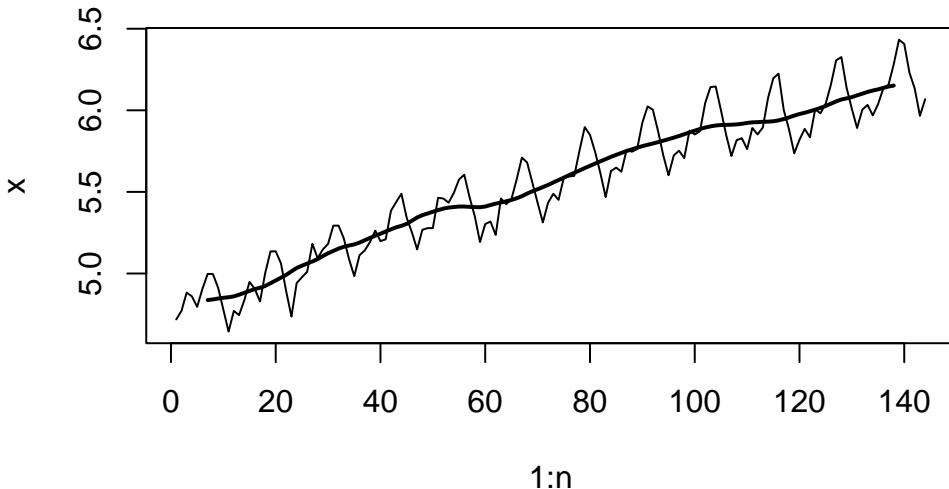


Figura 5.8.: LogAirPassengers con un suavizador de promedio móvil centrado de orden 12.

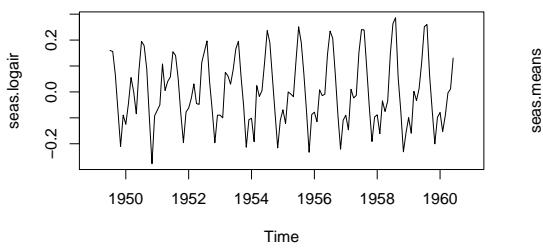
En relación con el modelo estimado en Ecuación 5.18, $\hat{tr}_t = \text{logair.sm12}$ representa la curva casi lineal en la Figura 5.8.

- b. **Eliminar el Componente de Tendencia de los Datos:** El paso subsiguiente implica la sustracción del componente de tendencia estimado de los datos (logAirPassengers).

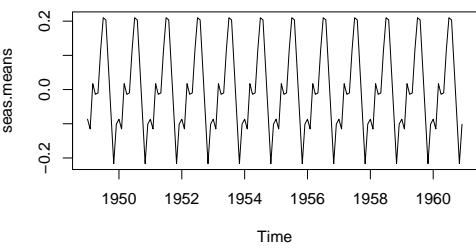
```
seas.logair=logAirPassengers-logair.sm12
round(seas.logair,4)
```

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
1949	NA	NA	NA	NA	NA	NA	0.1599	0.1561	0.0661
1950	-0.1249	-0.0451	0.0553	0.0010	-0.0844	0.0802	0.1953	0.1784	0.0882

1951	-0.0710	-0.0503	0.1080	0.0054	0.0406	0.0575	0.1550	0.1406	0.0512
1952	-0.0622	-0.0251	0.0311	-0.0452	-0.0479	0.1138	0.1552	0.1968	0.0383
1953	-0.0896	-0.1002	0.0754	0.0618	0.0299	0.0858	0.1656	0.1955	0.0597
1954	-0.1015	-0.1919	0.0245	-0.0173	0.0047	0.1148	0.2368	0.1905	0.0529
1955	-0.0689	-0.1217	-0.0002	-0.0080	-0.0182	0.1214	0.2512	0.1895	0.0688
1956	-0.0782	-0.1148	0.0082	-0.0145	-0.0088	0.1438	0.2347	0.2074	0.0673
1957	-0.0901	-0.1464	0.0101	-0.0233	-0.0135	0.1505	0.2405	0.2393	0.0914
1958	-0.0884	-0.1608	-0.0345	-0.0754	-0.0353	0.1449	0.2635	0.2862	0.0552
1959	-0.0992	-0.1593	0.0024	-0.0335	0.0137	0.1163	0.2518	0.2600	0.0646
1960	-0.0794	-0.1524	-0.0905	-0.0040	0.0112	0.1307	NA	NA	NA
	Oct	Nov	Dec						
1949	-0.0721	-0.2101	-0.0893						
1950	-0.1016	-0.2769	-0.0922						
1951	-0.0839	-0.1948	-0.0774						
1952	-0.0711	-0.1961	-0.0896						
1953	-0.0549	-0.2133	-0.1073						
1954	-0.0826	-0.2162	-0.1090						
1955	-0.0745	-0.2327	-0.0871						
1956	-0.0905	-0.2210	-0.1091						
1957	-0.0614	-0.1913	-0.0967						
1958	-0.0730	-0.2312	-0.1572						
1959	-0.0719	-0.2003	-0.0981						
1960	NA	NA	NA						



(a) LogAirPassengers sin tendencia



(b) Componente estacional estimado

Figura 5.9.: Datos de LogAirPassengers sin el componente de tendencia y componente estacional estimado

Se aprecia con mayor claridad el comportamiento estacional año tras año en la Figura 5.9a sin la “interferencia” de la tendencia. Específicamente, se observa un patrón similar en cada año (mayor cantidad de viajes en verano, disminución en noviembre, aún bajos pero con una ligera alza en diciembre, continuamente bajos

en enero y febrero, alza en marzo, y así sucesivamente). No obstante, se presentan variaciones de un año a otro: los viajes aéreos en noviembre fueron notablemente bajos en 1950 y luego inusualmente altos en julio y agosto de 1958.

- c. ***Calcular un “Promedio” del Componente Estacional Dentro del Año:*** Es importante notar que el componente estacional en el modelo (2.4) es un patrón general que se mantiene igual de un año a otro. Es decir, $\{s_t, t = 1, \dots, 12\} = \{s_{t+12}, t = 1, \dots, 12\} = \{s_{t+2(12)}, t = 1, \dots, 12\} = \dots$. El componente de ruido, z_t , ajusta las variaciones de un año a otro del patrón estacional general. El patrón estacional, $s_t, t = 1, \dots, 12$, se estima calculando el promedio a lo largo de los años, y el componente estacional estimado, \hat{s}_t (el cual es el mismo para cada año), se muestra en la Figura 5.9b.

```
seas.logair.numeric=as.numeric(seas.logair)
seas.logair.matrix=matrix(seas.logair.numeric,ncol=12)
seas.logair.matrix.t=t(seas.logair.matrix)
months=colMeans(seas.logair.matrix.t, na.rm=TRUE)
round(months,4)
```

```
[1] -0.0867 -0.1153  0.0172 -0.0139 -0.0098  0.1145  0.2100  0.2036  0.0640
[10] -0.0761 -0.2167 -0.1012
```

```
seas.means=rep(months,12)
seas.means=ts(seas.means,start=c(1949,1),frequency=12)
```

- d. ***Encontrar el componente de ruido restante:*** El ruido estimado en Ecuación 5.18, \hat{z}_t , es calculado de la siguiente manera

```
logair.noise = logAirPassengers - logair.sm12 - seas.means
plot(decompose(logAirPassengers))
```

Decomposition of additive time series

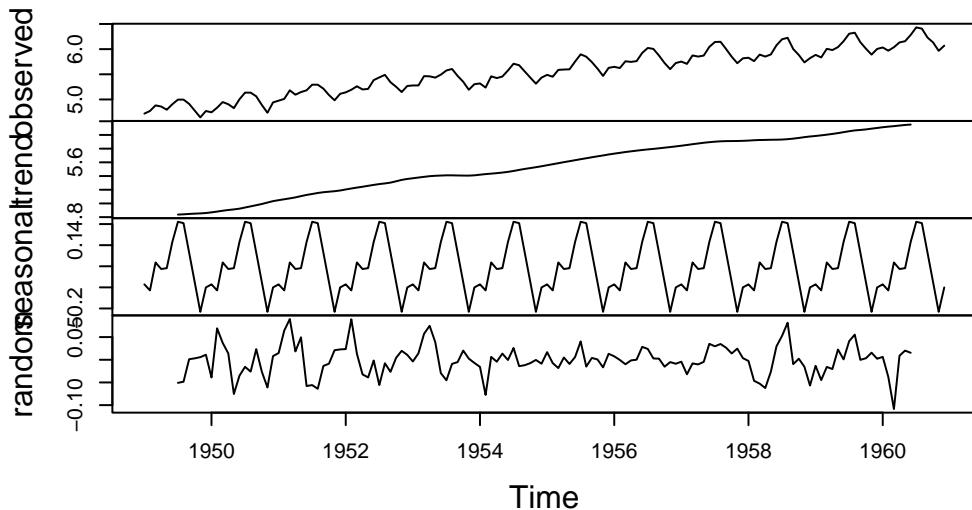


Figura 5.10.: Descomposición aditiva de LogAirPassengers

La Figura 5.10 muestra un gráfico de los datos de LogAirPassengers junto con las partes del procedimiento de descomposición.

5.2.1.2. Descomposición multiplicativa

Se llevará a cabo en el Ejemplo 5.6 una descomposición multiplicativa de los datos de AirPassengers. Se destaca que esta serie temporal exhibe un patrón estacional y una variabilidad intra-anual crecientes con el tiempo. A pesar de la posibilidad de modelar estos datos mediante el uso del logaritmo seguido de un modelo aditivo, en esta sección se opta por un enfoque multiplicativo para analizar los datos originales de AirPassengers. Al emplear la descomposición multiplicativa en el análisis de datos, se hace la suposición de que la serie temporal es el resultado de componentes estacionales, de tendencia y de ruido. El modelo estimado se expresa como;

$$x_t = \hat{s}_t \times \hat{tr}_t \times \hat{z}_t. \quad (5.19)$$

Ejemplo 5.6.

Descomposición multiplicativa de AirPassengers

- a. **Estimar el Componente de Tendencia:** Al igual que con el modelo aditivo, el primer paso consiste en utilizar un suavizador de media móvil centrada, nuevamente en este caso de orden 12. Anteriormente se calculó y representó gráficamente el suavizador de media móvil en la Figura 5.11.

```
library(tswge)
data(AirPassengers)
AirPass.sm12=ma.smooth.wge(AirPassengers,order=12)
AirPass.sm12=ts(AirPass.sm12$smooth,start=c(1949,1),frequency=12)
```

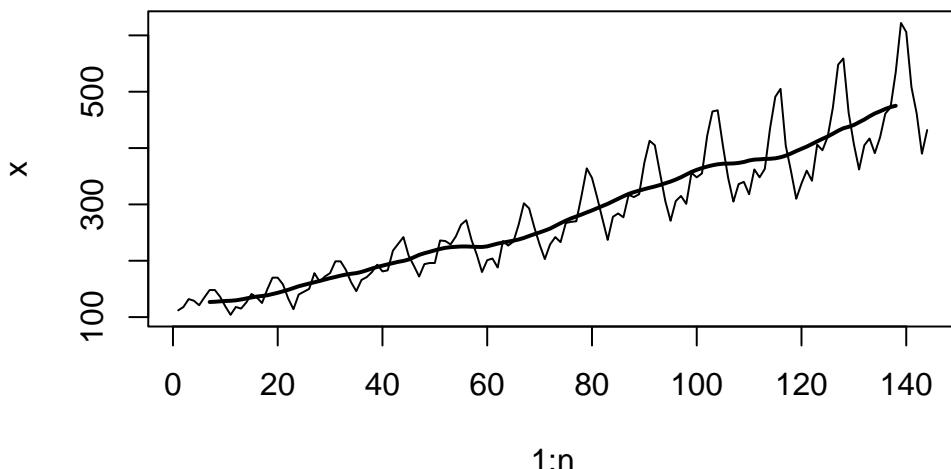


Figura 5.11.: Datos de pasajeros aéreos con suavizado de orden 12.

Es importante recordar que, en relación con el modelo estimado en Ecuación 5.19, $\hat{t}_t = \text{AirPass.sm12}$. Esta curva casi lineal se muestra como parte de la descomposición completa en la Figura 5.13.

- b. **Eliminar el Componente de Tendencia de los Datos:** El siguiente paso consiste en eliminar el componente de tendencia estimado del conjunto de datos de AirPassengers. Esto se puede lograr mediante la división (en lugar de la resta).

```
seas.AirPass=AirPassengers/AirPass.sm12
```

La conducta estacional de un año a otro resulta mucho más clara en la Figura 5.12a después de eliminar la “interferencia” de la tendencia y el aumento de la variabilidad dentro del año. También se observa que la variabilidad dentro del año no está aumentando tanto como en la Figura 5.3. El incremento en la variabilidad en el modelo final Ecuación 5.19 se debe a la tendencia creciente que se multiplica por los datos estacionales en la Figura 5.12a. Los patrones estacionales en la Figura 5.12a son similares a los de los datos aditivos mostrados en la Figura 5.9a.

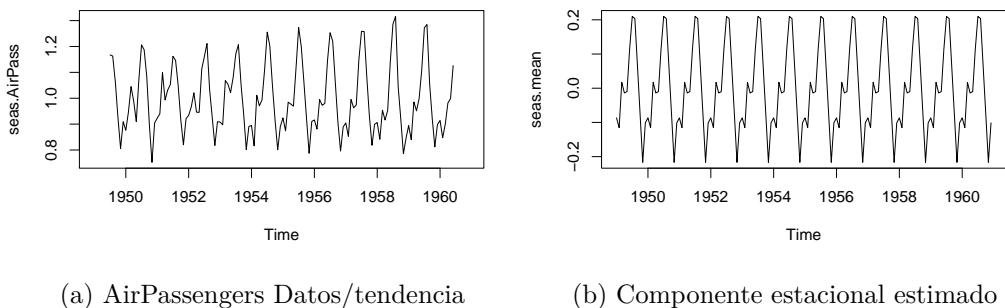


Figura 5.12.: Datos de AirPassengers sin el componente de tendencia y componente estacional estimado

- c. *Calcular un “Promedio” del Componente Estacional Dentro del Año:* Al igual que en el modelo aditivo, en el modelo Ecuación 5.17 , el componente estacional es un patrón general que se supone igual de un año a otro, y el componente de ruido, z_t , se ajusta a las variaciones de un año a otro con respecto al patrón estacional general. El componente estacional estimado, \hat{s}_t (que es idéntico para cada año), se representa en la Figura 5.12b. Se observa la similitud entre la Figura 5.12b y la Figura 5.9b, que fue el componente estacional para la descomposición aditiva de los datos logAirPassengers.

```
seas.AirPass.numeric=as.numeric(seas.AirPass)
seas.AirPass.matrix=matrix(seas.AirPass.numeric,ncol=12)
seas.AirPass.matrix.t=t(seas.AirPass.matrix)
months=colMeans(seas.AirPass.matrix.t,na.rm=TRUE)
seas.means=rep(months,12)
seas.means=ts(seas.means,start=c(1949,1),frequency=12)
```

- d. *Encontrar el componente de ruido restante:* El ruido estimado en Ecuación 5.19, \hat{z}_t , es calculado de la siguiente manera

```
Air.Pass.noise = AirPassengers / (AirPass.sm12 * seas.mean)
plot(decompose(AirPassengers))
```

Decomposition of additive time series

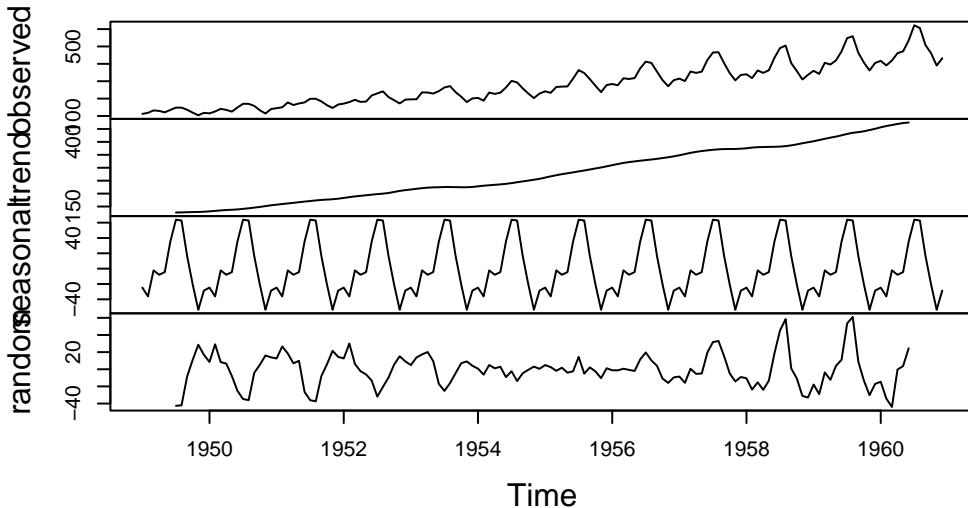


Figura 5.13.: Descomposición multiplicativa de AirPassengers

La Figura 5.13 muestra un gráfico de los datos de AirPassengers junto con las partes del procedimiento de descomposición.

5.2.2. Ajuste estacional

5.2.2.1. Ajuste estacional aditivo

Los ajustes estacionales están relacionados con las descomposiciones discutidas previamente. Si una descomposición aditiva es apropiada, entonces se utiliza un ajuste estacional aditivo. Un emparejamiento similar se aplica en el caso multiplicativo. El método de ajuste estacional aditivo más directo, consiste en obtener los datos ajustados estacionalmente, \hat{s}_t , utilizando la fórmula

$$s\hat{a}_t = x_t - \hat{s}_t, \quad (5.20)$$

que resta el componente estacional (mostrado en la Figura 5.9b) de los datos.

5.2.2.2. Ajuste estacional multiplicativo

Dado que la descomposición multiplicativa fue apropiada para los datos de Pasajeros Aéreos, se empleará un ajuste estacional multiplicativo para este conjunto de datos. Similar al método utilizado para el ajuste estacional aditivo, los datos ajustados estacionalmente utilizan la fórmula

$$s\hat{a}_t = x_t / \hat{s}_t, \quad (5.21)$$

para dividir los datos por el componente estacional (mostrado en la Figura 5.12b).

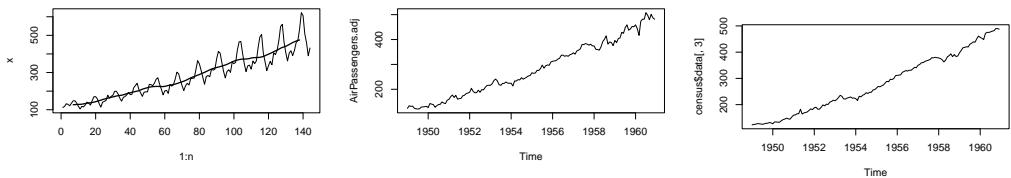
Ejemplo 5.7.

AirPassengers

La Figura Figura 5.14a muestra los datos de Pasajeros Aéreos superpuestos con la estimación de tendencia obtenida utilizando un suavizador de media móvil centrada de orden 12. La Figura 5.14b es una representación gráfica de los datos ajustados estacionalmente calculados utilizando Ecuación 5.21. Es decir, es una representación de **AirPassengers.adj**. Una vez más, los datos ajustados estacionalmente son similares a la tendencia estimada pero con más detalle respecto a los cambios mensuales.

El análisis de la Figura 5.14c muestra que el ajuste estacional utilizando **seas** es más suave y se ve menos afectado por el aumento de la variabilidad dentro del año en años posteriores. En general, los resultados son similares a los vistos en la Figura 5.14b con los efectos estacionales eliminados.

```
AirPassengers.adj = AirPassengers/seas.means
library(seasonal)
library(tswge)
data(AirPassengers)
AirPass.sm12=ma.smooth.wge(AirPassengers,order=12)
AirPass.sm12=ts(AirPass.sm12$smooth,start=c(1949,1),frequency=12)
census=seas(AirPassengers)
plot(AirPassengers.adj)
plot(census$data[,3])
```



(a) AirPassengers suavizamiento con(b) Ajuste estacional usando la fórmula (c) Ajuste estacional: Census

Figura 5.14.: Datos de AirPassengers y ajuste estacional multiplicativo

5.3. Pronóstico y métodos predictivos.

Una aplicación principal del análisis de series temporales es la de realizar pronósticos. Anteriormente se mencionó que uno de los propósitos del suavizado es prever valores futuros. De hecho, si existe evidencia de que los patrones previos en un conjunto de datos continuarán en el futuro, se pueden utilizar diversas técnicas para realizar pronósticos.

Por ejemplo, un propietario de negocio puede desear prever la demanda futura de cierto producto para asegurarse de tener la cantidad apropiada de inventario en stock. Una ciudad que toma decisiones sobre infraestructura podría necesitar prever su población en diez años. Un problema difícil pero de interés para muchos en el sector financiero (y para la mayoría de las personas, en realidad) es predecir las fluctuaciones del mercado de valores y de las acciones individuales para que se puedan tomar decisiones de inversión sólidas. Cada uno de estos ejemplos ilustra la aplicabilidad y necesidad de técnicas de pronóstico. Sin una opción mejor, dichos pronósticos a menudo se hacen de manera algo subjetiva, basados en la memoria pasada de eventos similares, rumores o tal vez mediante cálculos intuitivos pero presumidos que proporcionan conjecturas educadas y estimaciones.

Afortunadamente, las técnicas de análisis de series temporales proporcionan una alternativa matemática basada en si las suposiciones matemáticas subyacentes son apropiadas. Esto resulta en algoritmos que calculan pronósticos junto con límites de predicción correspondientes a un determinado nivel de confianza, análogos al cálculo de la media muestral más o menos un margen de error en el entorno de muestra aleatoria no temporal. El escenario típico es que se desarrolle un algoritmo de pronóstico que luego se utilice para predecir un resultado de interés. El pronóstico comprende estimaciones de parámetros que pueden encontrarse utilizando software en un esfuerzo por lograr una capacidad predictiva óptima.

Observación. Se utilizarán los términos predicción y pronóstico de manera sinónima.

5.3.1. Suavizador de media móvil predictivo

En la Sección 5.1.3.1 se explicó cómo se utiliza el suavizador de media móvil centrada para visualizar una versión suavizada de los datos con el propósito de recuperar señales subyacentes o eliminar ruido o efectos estacionales. También se pueden emplear medias móviles para la predicción. En lugar de la media móvil centrada discutida en la sección mencionada, se utilizará el promedio móvil predictivo para prever valores futuros. Si los datos no exhiben estacionalidad o tendencia, entonces para predecir el valor de la serie temporal en el instante $t + 1$, tiene sentido ‘predecir’ x_{t+1} como el promedio de los k valores de datos anteriores para algún k . Es decir, dejando que \tilde{x}_{t+1} denote la predicción de un paso hacia adelante de x_{t+1} dada la información hasta el tiempo t , entonces un predictor razonable y muy simple sería

$$\tilde{x}_{t+1} = \left(\sum_{i=0}^{k+1} x_{t-i} \right) / k$$

Esto es, el predictor de x_{t+1} es el promedio de los últimos k valores de datos.

Ejemplos

- Si se quiere usar un predictor de promedio móvil de 3 puntos para un conjunto de datos de longitud $t = 20$. Se observa que la predicción un paso hacia adelante de x_4 usando este predictor sería

$$\tilde{x}_4 = (x_3 + x_2 + x_1) / 3$$

Dado que el conjunto de datos tiene 20 valores, hay un valor conocido para x_4 , por lo que podemos comparar el predictor \tilde{x}_4 con el valor real, x_4 , para evaluar la precisión del predictor.

- Para predecir los valores x_4 hasta x_n , la predicción es

$$\tilde{x}_n = (x_{n-1} + x_{n-2} + x_{n-3}) / 3$$

Nuevamente, todos estos predictores de un paso hacia adelante pueden compararse con los valores reales para determinar la calidad de cada predicción.

- La predicción de x_{n+1} está dado por

$$\tilde{x}_{n+1} = (x_n + x_{n-1} + x_{n-2}) / 3$$

En este caso, \tilde{x}_{n+1} es un predictor de un valor futuro que presumiblemente no se conoce. Se puede hacer una evaluación sobre la calidad de esta predicción

basándose en las predicciones de x_4, x_5, \dots , todas las cuales pueden ser verificadas. El procedimiento se ilustra en el Ejemplo 5.8.

5.3.2. Suavizado exponencial

Si bien la predicción mediante el promedio móvil es fácil de conceptualizar y calcular, resulta poco realista asumir que todos los valores de datos precedentes tendrán una influencia igual en los valores futuros. Resulta más intuitivo pensar que, en muchos casos, los datos más recientes deberían tener mayor influencia en los valores futuros que los datos en un pasado más distante, ya que son más representativos del estado actual de la realidad. Un método de suavizado que tiene en cuenta esto se conoce como *suavizado exponencial*. El suavizado exponencial fue introducido por primera vez por R.G. Brown. Nuevamente se considera que cada valor de datos x_t en una serie temporal está compuesto por un valor medio en cada punto de tiempo t y un término de error independiente con media cero y varianza constante. Es decir, $x_t = \mu_t + e_t$. Para $0 \leq \alpha \leq 1$, la recursión de suavizado exponencial para $t = 1, 2, \dots, n$ es

$$u_{t+1} = \alpha x_t + (1 - \alpha)u_t \quad (5.22)$$

La Ecuación 5.22 es una combinación lineal del valor actual x_t junto con u_t , que es la estimación de μ_t basada en datos hasta, pero no incluyendo, el tiempo t . En el momento t , el promedio ponderado dado en la expresión anterior es el predictor de μ_{t+1} . Nótese que la estimación μ_t depende en gran medida del parámetro de suavizado α , que varía de 0 a 1. Cuanto más cercano esté α a 1, más peso se le otorga a los datos más recientes, mientras que cuanto más cercano esté α a 0, menos peso se le otorga a los datos más recientes. Naturalmente, el valor óptimo de α dependerá de la aplicación particular del conjunto de datos y se puede ajustar en consecuencia. Debido a que la fórmula anterior es recursiva, cada nueva estimación μ_t depende de la estimación anterior μ_{t-1} , que a su vez depende de la estimación μ_{t-2} , y así sucesivamente. Esta fórmula no parece ser “exponencial” de inmediato, ¿entonces cómo recibe el método su nombre? Dejando $t = 1, 2, 3, 4$, vemos que la fórmula recursiva produce lo siguiente:

$$\begin{aligned}
u_1 &= x_1 \\
u_2 &= \alpha x_1 + (1 - \alpha)u_1 \\
&= \alpha x_1 + (1 - \alpha)x_1 \\
&= x_1 \\
u_3 &= \alpha x_2 + (1 - \alpha)u_2 \\
&= \alpha x_2 + (1 - \alpha)x_1 \\
u_4 &= \alpha x_3 + (1 - \alpha)u_3 \\
&= \alpha x_3 + (1 - \alpha)[\alpha x_2 + (1 - \alpha)x_1] \\
&= \alpha x_3 + \alpha(1 - \alpha)x_2 + (1 - \alpha)^2 x_1 \\
u_5 &= \alpha x_4 + (1 - \alpha)u_4 \\
&= \alpha x_4 + (1 - \alpha)[\alpha x_3 + \alpha(1 - \alpha)x_2 + (1 - \alpha)^2 x_1] \\
&= \alpha x_4 + \alpha(1 - \alpha)x_3 + \alpha(1 - \alpha)^2 x_2 + (1 - \alpha)^3 x_1
\end{aligned}$$

Y en general,

$$u_k = \sum_{j=1}^{k-2} \alpha(1 - \alpha)^{j-1} x_{k-j} + (1 - \alpha)^{k-2} x_1. \quad (5.23)$$

Es claro a partir de la Ecuación 5.23 que dado que α está entre cero y uno, la fórmula recursiva otorga menos peso a las observaciones anteriores y más peso a las observaciones más recientes, y que la magnitud del ponderado es exponencial.

Ejemplo 5.8.

Predicción por promedio móvil y suavizado exponencial de AirPassengers

La Figura 5.15a muestra la gráfica de los datos de AirPassengers. La función `ma.pred.wge` implementada en el software R, que hace parte de la librería `tswge`, calcula predicciones a un paso utilizando un suavizador de media móvil de quinto orden. Utilizando los datos, la función extiende las predicciones hasta x_{n+20} . La Figura 5.15b representa tanto los datos como las predicciones.

La Figura 5.15c muestra los valores u_t , exhibe pronósticos a un paso utilizando suavizado exponencial con $\alpha = 0.4$. Los predictores resultantes son similares a los de la Figura 5.15b.

```

library(tswge)
plot(AirPassengers)
# Predicción por promedio móvil
ma.pred.wge(AirPassengers,order=5,n.ahead=20)
# Predicción por suavizamiento exponencial
expsmooth.wge(AirPassengers,alpha=0.4,n.ahead=20)

```

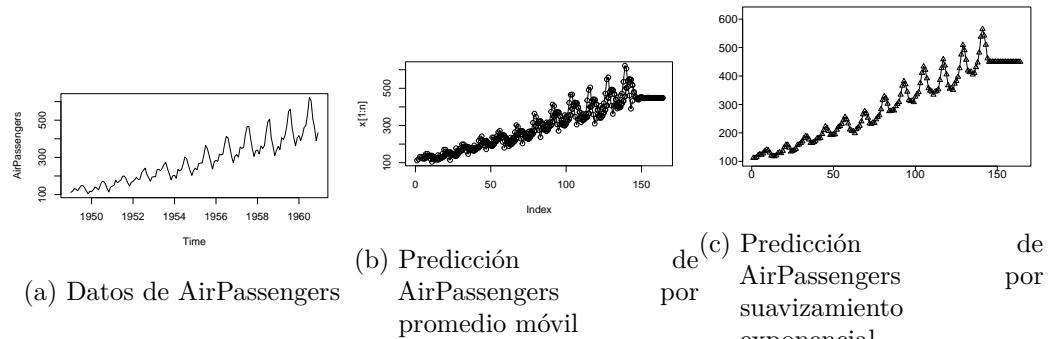


Figura 5.15.: Datos de AirPassengers, predicción por promedio móvil y suavizado exponencial

5.3.3. Pronóstico Holt-Winters

El enfoque Holt-Winters es una técnica desarrollada por los economistas Holt (Holt 1957) y Winters (Winters 1960).

Este método de predicción es una extensión del suavizado exponencial y se aplica a series temporales univariadas. El método no necesita un gran almacenamiento de datos y es simple. Es adecuado para la predicción a corto plazo y utiliza la función de máxima verosimilitud para estimar los parámetros. Existen dos modelos de Holt-Winter que utilizan modelos aditivos o multiplicativos basados en el componente estacional. Los modelos aditivos se aplican para un modelo con una tendencia lineal y con una tendencia exponencial.

5.3.3.1. Ecuaciones aditivas de Holt-Winters

El modelo aditivo de Holt-Winters para datos con tendencia y estacionalidad que no aumentan con el tiempo es adecuado (consulte la Ecuación 5.16). Las fórmulas para la predicción de Holt-Winters son generalizaciones de las ecuaciones de suavizado exponencial, y las ecuaciones de Holt-Winters son las siguientes:

$$\begin{aligned} u_t &= \alpha(x_t - s_{t-m}) + (1-\alpha)(u_{t-1} + v_{t-1}) \\ v_t &= \beta(u_t - u_{t-1}) + (1-\beta)v_{t-1} \\ s_t &= \gamma(x_t - u_{t-1}) + (1-\gamma)s_{t-m} \end{aligned} \quad (5.24)$$

Donde $0 \leq \alpha, \beta, \gamma \leq 1$, y donde m es la longitud del período. Para datos mensuales, $m = 12$, y para datos trimestrales, $m = 4$. Los u_t están relacionados con el suavizado exponencial simple y proporcionan una línea de base. Los v_t y s_t se relacionan con los efectos de tendencia y estacionales, respectivamente. Para los tiempos $t = m+1, \dots, n$, las predicciones de un paso adelante, \hat{x}_t , para la media en el tiempo t , se expresan como:

$$\hat{x}_t = u_{t-1} + v_{t-1} + s_{t-m}$$

Las predicciones para $x_{n+l}, l = 1, \dots, K$ (es decir, hasta K pasos más allá del final de los datos observados), se proporcionan de manera recursiva mediante:

$$\hat{x}_{n+l|n} = u_n + mv_n + s_{n+l-ml'}$$

Donde $l' = \left[\frac{l-1}{m}\right] + 1$, con $\left[\frac{l-1}{m}\right]$ denotando el entero mayor o igual a $\frac{l-1}{m}$. Aquí, α, β y γ son parámetros de suavizado, y se pueden obtener utilizando la función **HoltWinters** la cual parte de las funciones base que conforman al software *R*.

5.3.3.2. Ecuaciones multiplicativas de Holt-Winters

Como sugiere el término, las ecuaciones multiplicativas de Holt-Winters son aplicables a datos para los cuales el modelo multiplicativo la Ecuación 5.17 es apropiado. En este caso, las ecuaciones de Holt-Winters son:

$$\begin{aligned} u_t &= \alpha(x_t / s_{t-m}) + (1-\alpha)(u_{t-1} + v_{t-1}) \\ v_t &= \beta(u_t - u_{t-1}) + (1-\beta)v_{t-1} \\ s_t &= \gamma(x_t / u_{t-1}) + (1-\gamma)s_{t-m} \end{aligned}$$

Donde $0 \leq \alpha, \beta, \gamma \leq 1$, y donde nuevamente, m es la frecuencia. Las predicciones para los valores de x_t se expresan mediante:

$$\hat{x}_t = (u_{t-1} + v_{t-1})s_{t-m}$$

Las predicciones para $x_{n+l}, l = 1, \dots, K$ (es decir, hasta K pasos más allá del final de los datos observados), se proporcionan de manera recursiva mediante:

$$\hat{x}_{n+l|n} = (u_n + lv_n)s_{n+l-ml'}.$$

Donde $l' = \left[\frac{l-1}{m} \right] + 1$.

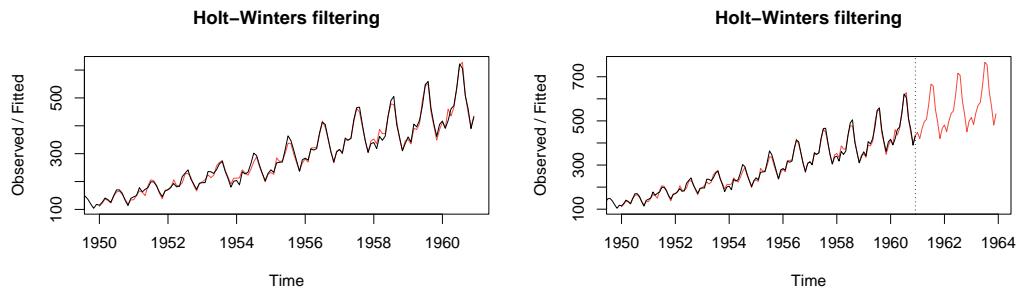
Ejemplo 5.9.

AirPassengers

Reconsidere los datos de AirPassengers. Dado que los puntos temporales posteriores revelan magnitudes crecientes de los picos y valles cíclicos, la serie temporal es multiplicativa en lugar de aditiva. La Figura 5.16a muestra las predicciones de un paso adelante de Holt-Winters para los años 1950-1960 (superpuestas a los datos reales) dadas por la fórmula utilizando los parámetros de suavizado y coeficientes estimados anteriormente. Las predicciones de un paso adelante (línea sólida) son bastante precisas y apenas se distinguen de los datos (puntos en la línea). La Figura 5.16b muestra los datos de AirPassengers junto con las predicciones de Holt-Winters para los próximos tres años (línea punteada). Estas predicciones parecen extender con precisión el patrón precedente.

```
# Figure 2.20(a)
ap.hw=HoltWinters(AirPassengers,seasonal="mult")
plot(ap.hw)

# Figure 2.20(b)
ap.pred=predict(ap.hw,n.ahead=36)
plot(ap.hw,ap.pred,lty=1:2)
```



- (a) Predicción de AirPassengers un paso adelante (b) Predicción por Holt-Winters de AirPassengers

Figura 5.16.: Predicción mediante Holt-Winters de los datos de AirPassengers

5.3.4. Modelo Autoregresivo (AR)

Definición 5.11. Se afirma que el proceso X_t satisface un modelo AR(p) (Autoregresivo de orden p) si

$$X_t = a_t + \beta + \sum_{k=1}^p \phi_k X_{t-k} \quad (5.25)$$

donde $\phi_k, k = 1, \dots, p$ son constantes reales, $\beta = (1 - \phi_1 - \phi_2 - \dots - \phi_p) \mu$, $\phi_p \neq 0$, y a_t es un proceso de ruido blanco con media cero y varianza finita σ_a^2 .

La fórmula en la Ecuación 5.25 indica que el valor del proceso en el tiempo t es una combinación lineal de los p valores anteriores más un componente de ruido aleatorio en a_t . Iniciamos nuestra discusión sobre los modelos AR al abordar sus propiedades, incluidas las condiciones de estacionariedad y el comportamiento de las autocorrelaciones y densidades espectrales para modelos específicos.

El modelo AR(p) general definido en la Definición 5.11, se asemeja a una ecuación de regresión múltiple, donde, en este caso, las “variables independientes” son los p valores previos de la “variable dependiente” X_t . Otra forma de escribir la Ecuación 5.25, después de reorganizar los términos, es:

$$X_t - \mu - \phi_1(X_{t-1} - \mu) - \phi_2(X_{t-2} - \mu) - \dots - \phi_p(X_{t-p} - \mu) = a_t \quad (5.26)$$

Al igual que en el caso de los modelos AR(1) y AR(2), se expresará con frecuencia el AR(p) en la forma de media cero:

$$X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} - \dots - \phi_p X_{t-p} = a_t \quad (5.27)$$

Las ecuaciones Ecuación 5.25 a Ecuación 5.27 dan la impresión de que un modelo AR(p) será mucho más complicado de manejar que un modelo AR(1) o AR(2). La comprensión de las características de los modelos AR(1) y AR(2) conduce directamente a comprender el comportamiento de un modelo AR(p).

5.3.4.0.1. Hechos sobre el modelo AR(p)

- i. $E[X_t] = \mu$, para la forma “no nula de la media” del modelo AR(p) en la Ecuación 5.26 y Ecuación 5.27.

ii. El proceso de varianza es

$$\sigma_x^2 = \gamma_0 = \frac{\sigma_a^2}{1 - \phi_1 \rho_1 - \phi_2 \rho_2 - \dots - \phi_p \rho_p}$$

la cual es constante y finita cuando X_t es estacionaria.

iii. La autocorrelación de un proceso AR(p) satisface

$$\rho_k = \phi_p + \sum_{n=1}^{p-1} \phi_n \rho_{k-n}$$

La ecuación (5.28) es una generalización de (5.17) para el caso AR(2), y conduce a las ecuaciones de Yule-Walker de orden p :

$$\begin{aligned}\rho_1 &= \phi_1 + \phi_2 \rho_1 + \dots + \phi_p \rho_{p-1} \\ \rho_2 &= \phi_1 \rho_1 + \phi_2 + \dots + \phi_p \rho_{p-2} \\ &\vdots \\ \rho_p &= \phi_1 \rho_{p-1} + \phi_2 \rho_{p-2} + \dots + \phi_p.\end{aligned}$$

Análogo al caso AR(2), conocer los valores de $\phi_1, \phi_2, \dots, \phi_p$ nos permite resolver este sistema de ecuaciones de dimensión $p \times p$ para ρ_k , donde $k = 1, 2, \dots, p$. Las autocorrelaciones basadas en el modelo, ρ_k , para $k > p$, se pueden calcular utilizando la recursión $\phi_1 \rho_{k-1} - \phi_2 \rho_{k-2} + \dots + \phi_p \rho_{k-p}$. No sorprendentemente, se utilizan funciones computacionales para realizar estos cálculos.

iv. La densidad espectral de un modelo AR(p) es dada por

$$S_x(f) = \frac{\sigma_a^2}{\gamma_0 |1 - \phi_1 e^{-2\pi i f} - \phi_2 e^{-4\pi i f} - \dots - \phi_p e^{-2p\pi i f}|^2}$$

Definición 5.12 (Autocorrelaciones parciales). Sea X_t un proceso estacionario con autocorrelaciones $\rho_j = j = 0, 1, \dots$

- a. La autocorrelación parcial en rezago k , denotada como ϕ_{kk} , es la correlación entre X_t y X_{t+k} condicional al “conocimiento” de las variables interviniéntes $X_{t+1}, X_{t+2}, \dots, X_{t+k-1}$.
- b. Considere las siguientes ecuaciones de Yule-Walker donde ϕ_{kj} denota el coeficiente j -ésimo asociado con las ecuaciones de Yule-Walker de orden k .

$$\begin{aligned} k &= 1 \\ \rho_1 &= \phi_{11} \end{aligned}$$

$$\begin{aligned} k &= 2 \\ \rho_1 &= \phi_{21} + \phi_{22}\rho_1 \\ \rho_2 &= \phi_{21}\rho_1 + \phi_{22} \end{aligned}$$

En general...

$$\begin{aligned} \rho_1 &= \phi_{k1} + \phi_{k2}\rho_1 + \cdots + \phi_{kk}\rho_{k-1} \\ \rho_2 &= \phi_{k1}\rho_1 + \phi_{k2} + \cdots + \phi_{kk}\rho_{k-2} \\ &\vdots \\ \rho_k &= \phi_{k1}\rho_{k-1} + \phi_{k2}\rho_{k-2} + \cdots + \phi_{kk} \end{aligned}$$

La función de autocorrelación parcial se define como $\phi_{kk}, k = 1, 2, \dots$

5.3.4.0.2. Notación de operador y ecuación característica para un AR(p)

El modelo AR(p) en la Ecuación 5.26 puede ser escrito en notación de operador como

$$(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p)(X_t - \mu) = a_t$$

O utilizando una notación abreviada, $\phi(B)(X_t - \mu) = a_t$, donde $\phi(B)$ es el operador de orden p

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p.$$

Convirtiendo el operador $\phi(B)$ en la cantidad algebraica $\phi(z)$ resulta en el polinomio característico general AR(p)

$$\phi(z) = 1 - \phi_1 z - \phi_2 z^2 - \cdots - \phi_p z^p$$

La correspondiente ecuación característica AR(p) es

$$\phi(z) = 1 - \phi_1 z - \phi_2 z^2 - \cdots - \phi_p z^p = 0$$

La ecuación característica tiene p raíces r_1, r_2, \dots, r_p que son reales y/o complejas, donde las raíces complejas aparecen como pares conjugados y algunas raíces pueden ser repetidas.

Teorema 5.3. *Un proceso AR(p) es estacionario si y solo si todas las raíces de la ecuación característica son mayores que uno en valor absoluto.*

Prueba. Vea Harvey (1981). □

Ejemplo 5.10.

Un modelo AR(4)

Considere el modelo AR(4)

$$X_t - 0.13X_{t-1} - 1.4414X_{t-2} + .0326X_{t-3} + .8865X_{t-4} = a_t \quad (5.28)$$

donde $\sigma_a^2 = 1$. La notación del operador para este modelo es

$$(1 - 0.13B + 1.4414B^2 - .0326B^3 + 0.8865B^4)X_t = a_t,$$

y la correspondiente ecuación característica es

$$1 - 0.13z + 1.4414z^2 - .0326z^3 + 0.8865z^4 = 0.$$

La Figura 5.17 representa una realización de longitud $n = 200$ del proceso descrito en la Ecuación 5.28, junto con las autocorrelaciones muestrales asociadas y la estimación de la densidad espectral Parzen, respectivamente.

```
library(tswge)
x=gen.arma.wge(n=200,phi=c(0.1300,1.4414,-.0326,-.8865),sn=9310,plot=FALSE)
plotts.sample.wge(x)
```

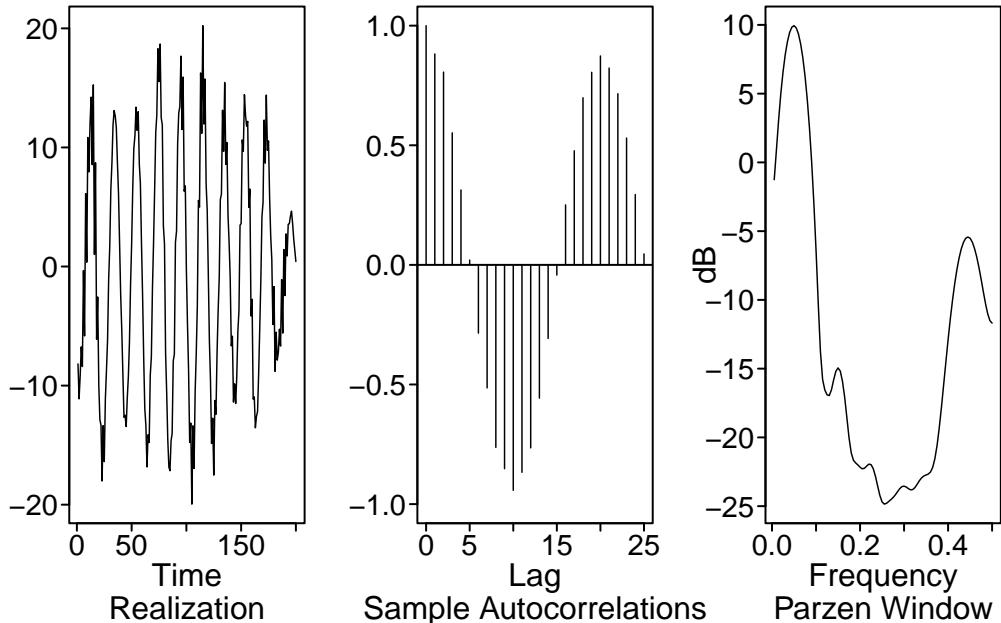


Figura 5.17.: Realización del modelo AR(4)

5.3.4.0.3. El Test Aumentado de Dickey-Fuller

Este test ha estado en uso durante muchos años para probar las hipótesis:

- H_0 : el modelo contiene una raíz unitaria
- H_a : el modelo no contiene una raíz unitaria (y por lo tanto es estacionario)

Estadístico de prueba: τ

Región de rechazo: Rechazar H_0 si $\tau < d_\alpha$ donde d_α es el valor crítico de nivel α .

David Alan Dickey (1976) obtiene la distribución límite (complicada) del estadístico de prueba. Si $\tau \geq d_{.05}$, entonces no se rechaza H_0 y la prueba de Dickey-Fuller detecta una raíz unitaria. Nótese que el rechazo de la hipótesis nula lleva a la conclusión de que el proceso es estacionario. Por lo tanto, la conclusión de una raíz unitaria se basa en no rechazar la hipótesis nula. Es importante recalcar que no rechazar la hipótesis nula no implica creer que la hipótesis nula sea verdadera, sino simplemente que no hubo suficiente evidencia para rechazarla.

Para llevar a cabo pruebas de raíz unitaria, se empleará una implementación del test del software estadístico R. Existen diversas opciones, y se utilizará el siguiente comando que incorpora una constante pero no una tendencia en el modelo, y utiliza el criterio de información de Akaike (AIC) para seleccionar el número de rezagos. Se recomienda

consultar las obras de David A. Dickey y Fuller (1979) o Fuller (1995) para obtener más detalles al respecto.

5.3.4.0.4. Factorización del polinomio característico de un AR(p)

Las raíces de una ecuación cuadrática se pueden encontrar mediante el uso de la fórmula cuadrática. Sin embargo, las cosas se vuelven más complicadas para órdenes polinómicos mayores a dos. La ecuación cúbica

$$1 - 2.1z + 1.6z^2 - .3z^3 = 0$$

puede factorizarse en la forma $(1 - .5z)(1 - 1.6z + .8z^2)$. Basándose en esta factorización, las raíces que se obtienen son $r_1 = 1/.5 = 2$, $r_2 = 1 + .5i$ y $r_3 = 1 - .5i$. Es decir, este AR(3) tendrá un comportamiento de primer orden asociado con $1 - .5B$ (es decir, una frecuencia de cero), un comportamiento cíclico de segundo orden con una frecuencia de sistema $f_0 = 0.07$, y el proceso es estacionario porque todas las raíces están fuera del círculo unitario.

Teorema 5.4. *El polinomio de orden p , $1 - \phi_1 z - \phi_2 z^2 - \dots - \phi_p z^p$, siempre puede descomponerse como un producto de*

- i. factores de primer orden (lineales) asociados con raíces reales
- ii. factores de segundo orden (cuadráticos) cuyas raíces son pares conjugados complejos.

5.3.4.0.5. Tablas de factores para modelos AR(p)

El Teorema 5.4 establece que cualquier polinomio de orden p puede expresarse como un producto de factores de primer orden y/o factores irreducibles de segundo orden. Comprender los factores de primer y segundo orden es la clave para comprender el modelo AR(p).

Ejemplo 5.10 (Continuación...)

Se considera de nuevo el modelo AR(4) en Ecuación 5.28. La ecuación característica asociada es

$$1 - .13z - 1.4414z^2 + .0326z^3 + .8865z^4 = 0$$

La forma factorizada (obtenida numéricamente) es

$$(1 - 1.89B + .985B^2)(1 + 1.76B + .9B^2) = 0$$

La tabla de factores es una herramienta muy útil para resumir rápidamente la

escencia de un modelo AR(p) con respecto a los factores de primer y segundo orden.

```
library(tswge)
factor.wge(phi=c(.13, 1.4414, -.0326, -.8865))
```

Coefficients of AR polynomial:
0.1300 1.4414 -0.0326 -0.8865

AR Factor Table				
Factor	Roots	Abs Recip	System	Freq
1-1.8900B+0.9850B^2	0.9594+-0.3079i	0.9925	0.0494	
1+1.7600B+0.9000B^2	-0.9778+-0.3938i	0.9487	0.4391	

En este caso, hay dos factores de segundo orden, $1 - 1.89B + .985B^2$ y $1 + 1.76B + .9B^2$.

5.4. Evaluación de la precisión de los pronósticos

Para obtener una cuantificación “general” de la calidad de las predicciones, se evalúa qué tan bien coinciden las predicciones (f_t) con los valores reales (y_t) en el tiempo t . Afortunadamente, se han ideado métricas de error para evaluar la calidad del modelo y permitir la comparación con otras regresiones que poseen diferentes parámetros. Estas métricas son resúmenes breves pero informativos de la calidad de los datos. A continuación se presentan algunas métricas de rendimiento más comunes (Segall y Niu (2022)).

5.4.1. MAE

El error absoluto medio (MAE) se calcula tomando el residuo para cada punto de datos, considerando únicamente el valor absoluto para minimizar el impacto de los valores atípicos en comparación con Ecuación 5.31. Luego, se obtiene el promedio de todos estos residuos. La ecuación formal se presenta a continuación:

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - f_t| \quad (5.29)$$

Dado que se emplea el valor absoluto del residuo, no se indica el rendimiento inferior o superior del modelo. Cada residuo contribuye de manera equitativa al error total, y los errores más grandes tienen una mayor contribución al error general. Un MAE pequeño indica un buen rendimiento de predicción, mientras que un MAE grande sugiere que el modelo puede tener dificultades en ciertas áreas. Obtener MAE perfecta de 0 es rara, indica que el modelo es un predictor impecable.

Sin embargo, el uso del valor absoluto del residuo puede no ser el mejor enfoque para interpretar los datos, ya que los *valores atípicos* (es decir, los puntos de datos que se alejan significativamente de la tendencia general de los datos) pueden afectar significativamente el rendimiento del modelo. Dependiendo del tratamiento de los valores atípicos y extremos en los datos, es posible que se desee resaltar o minimizar su impacto. Como resultado, la elección de la métrica de error adecuada puede verse influida por el problema de los valores atípicos.

5.4.2. MSE

El error cuadrático medio (MSE) es similar al MAE, pero eleva al cuadrado la diferencia antes de sumarlos todos en lugar de utilizar el valor absoluto. Esta diferencia se puede observar en la siguiente ecuación:

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - f_t)^2 \quad (5.30)$$

El error medio absoluto (MAE) y el error cuadrático medio (MSE) son métricas de error comúnmente utilizadas en la evaluación de modelos. Sin embargo, el MSE suele ser mayor que el MAE debido al cuadrado de la diferencia. Comparar los dos directamente no siempre es posible, y en su lugar, debemos comparar las métricas de error de nuestro modelo con las de un modelo ya conocido o que se ajuste a la serie de datos. El efecto de los valores atípicos en nuestros datos es más evidente con la presencia del término cuadrado en la ecuación MSE. Mientras que cada residuo en MAE contribuye proporcionalmente al error total, el error crece cuadráticamente en MSE. En última instancia, esto significa que los valores atípicos en nuestros datos contribuirán a un error total mucho mayor en el MSE que en el MAE. Del mismo modo, nuestro modelo se verá más penalizado por hacer predicciones que difieran mucho del valor real correspondiente.

5.4.3. RMSE

RMSE, o error cuadrático medio, es una medida frecuentemente utilizada para evaluar la diferencia entre los valores predichos f_t y los valores observados y_t . Su función se expresa a continuación, donde n representa el número de observaciones. En comparación con el error cuadrático medio (MSE), RMSE toma la raíz cuadrada de MSE y restituye

la unidad al mismo nivel que la variable dependiente. Por lo tanto, tiene la ventaja de ser interpretado directamente. En general, un valor de RMSE más bajo es preferible, y RMSE= 0 indica un ajuste perfecto de los datos. La desventaja de RMSE es su sensibilidad a valores atípicos, ya que unos pocos errores grandes en la suma pueden generar un aumento significativo, y la prueba no distingue entre subestimación y sobreestimación. Como se discutió anteriormente en la descripción de los datos, el conjunto de datos que se utiliza tiene varios valores extremos de gastos elevados, por lo que utilizar solo RMSE como medida podría no ser muy adecuado.

La fórmula para el cálculo de RMSE es la siguiente:

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (y_t - f_t)^2}{n}} \quad (5.31)$$

5.4.4. MAPE

El error porcentual absoluto medio (MAPE) mide la precisión de la predicción como un porcentaje y se define generalmente de la siguiente manera. La ventaja es que es muy intuitivo interpretar el error relativo, y un MAPE más bajo significa un error menor. MAPE es similar a MAE, pero normaliza MAE mediante observaciones reales, resolviendo así el problema de que MAE proporciona poca información sobre el error al comparar datos de diferentes escalas. Sin embargo, también presenta la desventaja de que puede producir valores infinitos o indefinidos para valores reales cercanos o iguales a cero. Otra limitación de MAPE es que penaliza más los errores negativos que los errores positivos. Por ejemplo, para un valor real de 100 y un valor estimado de 90, el MAPE es 0.10. Para el mismo valor estimado y un valor real de 80, el MAPE es 0.125. Como resultado, si se utiliza MAPE como función objetivo, el estimador preferirá valores más pequeños y puede sesgarse hacia errores negativos.

$$MAPE = \frac{100}{n} \sum_{t=1}^n \left| \frac{y_t - f_t}{y_t} \right| \quad (5.32)$$

Parte III.

Redes neuronales

6. Redes Neuronales

El funcionamiento de los cerebros de humanos y otros animales es intrigante porque son capaces de realizar tareas muy complejas en un tiempo muy corto y con alta eficiencia. Por ejemplo, las señales de los sensores en el cuerpo transmiten información relacionada con la vista, el oído, el gusto, el olfato, el tacto, el equilibrio, la temperatura, el dolor, etc. Luego, las neuronas del cerebro, que son unidades autónomas, transmiten, procesan y almacenan esta información para que se pueda responder con éxito a estímulos externos e internos. Las neuronas de muchos animales transmiten picos de actividad eléctrica a través de una hebra larga y delgada llamada axón. Un axón se divide en miles de terminales o ramas, donde, según el tamaño de la señal, se conectan a dendritas de otras neuronas (Figura 6.1). Se estima que el cerebro está compuesto por alrededor de 10^{11} neuronas que trabajan en paralelo, ya que el procesamiento realizado por las neuronas y la memoria capturada por las sinapsis se distribuyen juntas sobre la red. La cantidad de información procesada y almacenada depende de los niveles umbral de disparo y también del peso dado por cada neurona a cada una de sus entradas. Una de las características de las neuronas biológicas, a las que deben su gran capacidad para procesar y realizar tareas altamente complejas, es que están altamente conectadas con otras neuronas de las cuales reciben estímulos de un evento a medida que ocurre, o cientos de señales eléctricas con la información aprendida.

Las redes neuronales tienen sus raíces en la búsqueda de emular el funcionamiento del cerebro humano en la década de 1940. McCulloch y Pitts (1943) propusieron el concepto inicial de una “neurona artificial” que podría realizar operaciones lógicas básicas. Sin embargo, fue en la década de 1950 cuando el psicólogo Frank Rosenblatt desarrolló el “perceptrón”, Rosenblatt (1960), un tipo de red neuronal que podía aprender a reconocer patrones a través de entrenamiento.

A pesar de su potencial, las limitaciones del perceptrón y la falta de avances en la capacidad computacional llevaron a un declive en la investigación en redes neuronales en los años siguientes. Sin embargo, en la década de 1980 y 1990, hubo un resurgimiento del interés debido a nuevos algoritmos de aprendizaje y avances en el hardware, permitiendo el entrenamiento de redes más complejas.

La importancia de las redes neuronales en la predicción de datos radica en su capacidad para aprender patrones y relaciones en conjuntos de datos vastos y complejos. A través del proceso de entrenamiento, una red neuronal ajusta sus pesos y conexiones internas para capturar características relevantes en los datos. Esto les permite realizar tareas como clasificación y regresión, lo que a su vez permite la predicción de resultados futuros.

Con el tiempo, las redes neuronales se han vuelto más sofisticadas, dando lugar a arquitecturas como las redes neuronales convolucionales (CNN) para el procesamiento de imágenes y las redes neuronales recurrentes (RNN) para el procesamiento de secuencias. Además, el surgimiento del aprendizaje profundo (deep learning) ha permitido el entrenamiento de redes neuronales con muchas capas, lo que ha mejorado significativamente su capacidad para abordar problemas complejos de predicción.

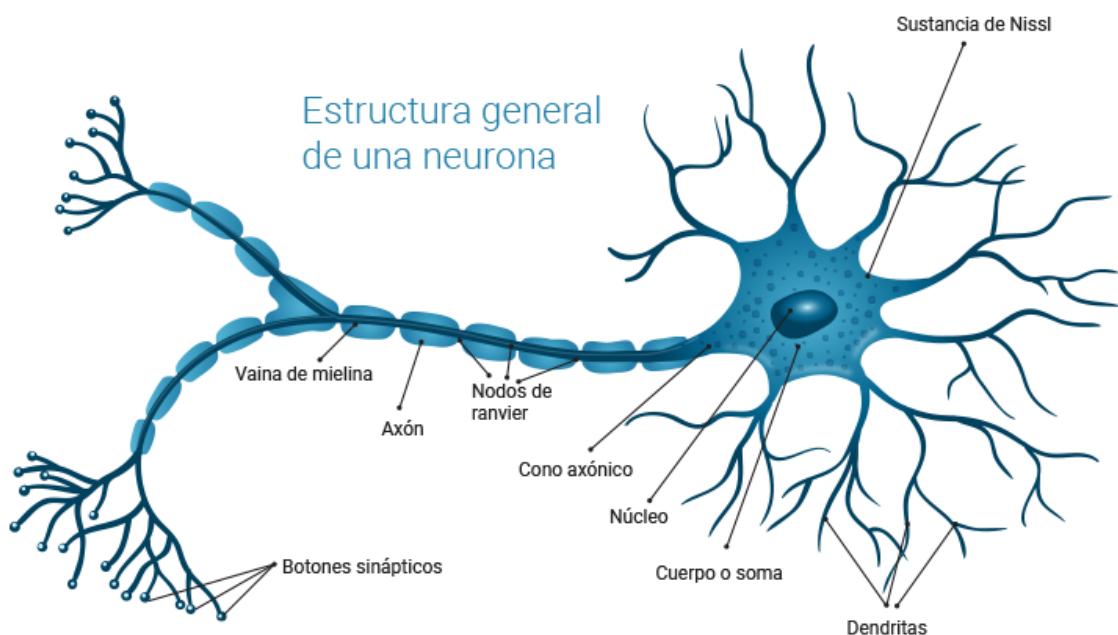


Figura 6.1.: Representación gráfica de una neurona biológica

6.1. Elementos fundamentales de las Redes Neuronales Artificiales

Para obtener una comprensión clara de los principales elementos utilizados para construir modelos de redes neuronales artificiales (RNA), en la Figura 6.2 se presenta un modelo general de red neuronal artificial que incorpora los componentes fundamentales para este tipo de modelos.

La información de entrada, x_1, \dots, x_p , es recibida por la neurona del sistema sensorial externo u otras neuronas con las que tiene conexión. El vector de pesos sinápticos $\mathbf{w} = (w_1, \dots, w_p)$ modifica la información recibida emulando la sinapsis entre las neuronas biológicas. Estos pueden interpretarse como ganancias que pueden atenuar o amplificar los valores que desean propagar hacia la neurona. El parámetro b_j se conoce como el sesgo

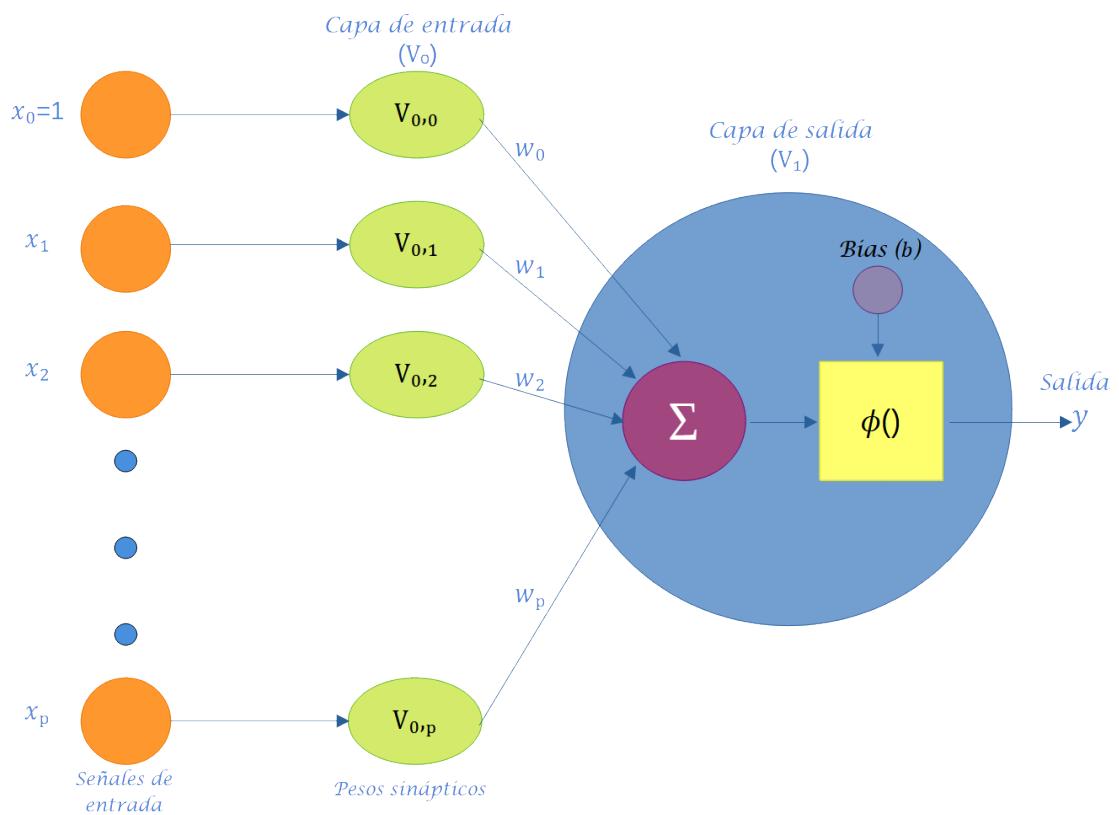


Figura 6.2.: Modelo general de redes neuronales artificiales.

(intercepto o umbral) de una neurona. En redes neuronales artificiales, el aprendizaje se refiere al método de modificar los pesos de las conexiones entre los nodos (neuronas) de una red especificada.

Los valores recibidos por la neurona son ajustados por los pesos sinápticos y que sumados para generar la entrada neta, expresada matemáticamente como:

$$v_j = \sum_{j=1}^p \omega_{ij} x_j$$

La entrada neta (v_j) determina si la neurona se activa o no. La activación de la neurona depende de la función de activación, evaluándose la entrada neta en dicha función para obtener la salida de la red, como se ilustra a continuación:

$$y_j = g(v_j)$$

donde g representa la función de activación. Por ejemplo, si se define esta función como un escalón unitario (también llamado umbral), la salida será 1 si la entrada neta es mayor que cero; de lo contrario, la salida será 0.

Aunque no existe un comportamiento biológico análogo a las neuronas cerebrales, el uso de la función de activación es un artificio que permite aplicar RNA a una variedad de problemas reales. La salida y_j de la neurona se genera al evaluar la entrada neta (v_j) en la función de activación, pudiendo propagarse a otras neuronas o ser la salida final de la red, con una interpretación específica según la aplicación.

En términos generales, el funcionamiento de un modelo de red neuronal artificial se lleva a cabo mediante elementos simples denominados neuronas. Las señales se transmiten entre neuronas a través de enlaces de conexión, cada uno con un peso asociado que multiplica la señal transmitida. Cada neurona aplica una función de activación (generalmente no lineal) a las entradas de la red (suma ponderada de las señales de entrada) para determinar su signo correspondiente.

Un modelo de RNA de una sola capa, como el presentado en la Figura 6.2, posee una capacidad de procesamiento limitada por sí mismo y una aplicabilidad reducida; su verdadero poder radica en la interconexión de múltiples redes neuronales artificiales, similar al funcionamiento del cerebro humano. Este enfoque ha motivado a diversos investigadores a proponer diversas arquitecturas para la interconexión de neuronas en el contexto de RNA. A continuación, se presentan las definiciones de RNA y aprendizaje profundo (Montesinos López, Montesinos López, y Crossa (2022)).

Definición 6.1 (Red Neuronal Artificial). Una red neuronal artificial es un sistema compuesto por numerosos elementos de procesamiento simples que operan en paralelo, y cuya función está determinada por la estructura de la red y el peso de las conexiones.

En cada uno de los nodos o elementos de cómputo, que posee una capacidad de procesamiento baja, se lleva a cabo el procesamiento.

Definición 6.2 (Aprendizaje profundo). Se define el aprendizaje profundo como una generalización de RNA donde se utilizan más de una capa oculta, lo que implica que se utilizan más neuronas para implementar el modelo. Por esta razón, a una red neuronal artificial con múltiples capas ocultas se le llama Red Neuronal Profunda (RNP) y la práctica de entrenar este tipo de redes se llama aprendizaje profundo (AP).

Para una comprensión más completa de los elementos que componen una red neuronal artificial, resulta crucial diferenciar entre las diversas categorías de capas y tipos de neuronas. Por consiguiente, se procede a detallar los tipos de capas seguido por una exposición más detallada de los tipos de neuronas.

- a. **Capa de entrada:** Es el conjunto de neuronas que recibe directamente la información proveniente de las fuentes externas de la red. En el contexto de la Figura 6.3, esta información es x_1, \dots, x_8 . Por lo tanto, el número de neuronas en una capa de entrada es la mayoría de las veces igual al número de variables explicativas de entrada proporcionadas a la red. Por lo general, las capas de entrada están seguidas por al menos una capa oculta. Solo en las redes neuronales feedforward, las capas de entrada están completamente conectadas a la siguiente capa oculta.
- b. **Capas ocultas:** Consisten en un conjunto de neuronas internas de la red que no tienen contacto directo con el exterior. El número de capas ocultas puede ser 0, 1 o más. En general, las neuronas de cada capa oculta comparten el mismo tipo de información; por esta razón, se llaman capas ocultas. Las neuronas de las capas ocultas pueden estar interconectadas de diferentes maneras; esto determina, junto con su número, las diferentes arquitecturas de RNA y RNP. La información aprendida extraída de los datos de entrenamiento se almacena y captura mediante los valores de peso de las conexiones entre las capas de la red neuronal artificial. Además, es importante señalar que las capas ocultas son componentes clave para capturar de manera más eficiente comportamientos no lineales complejos de los datos.
- c. **Capa de salida:** Es un conjunto de neuronas que transfiere la información procesada por la red hacia el exterior. En la Figura 6.3, las neuronas de salida corresponden a las variables de salida y_1, y_2, y_3 e y_4 . Esto implica que la capa de salida proporciona la respuesta o predicción del modelo de red neuronal artificial basada en la entrada proveniente de la capa de entrada. La salida final puede ser continua, binaria, ordinal o de conteo, dependiendo de la configuración de la RNA, la cual está controlada por la función de activación especificada en las neuronas de la capa de salida.

A continuación, se definen los tipos de neuronas:

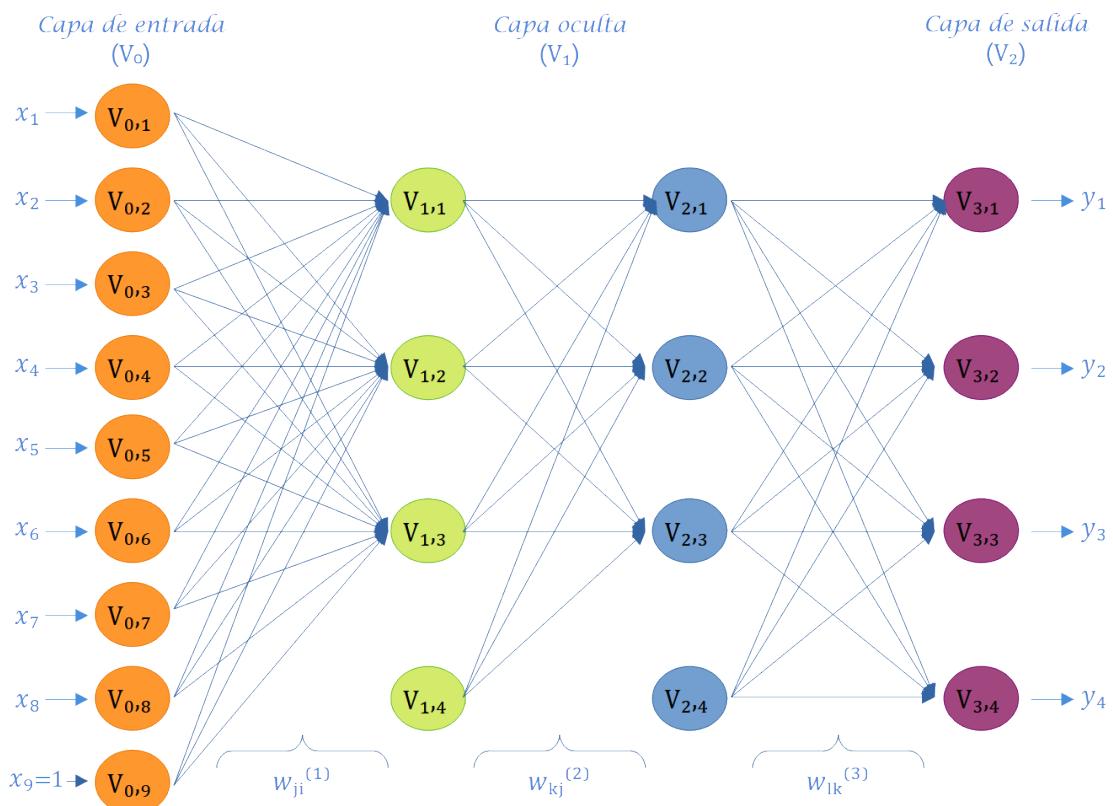


Figura 6.3.: Red neuronal artificial profunda feedforward con ocho variables de entrada, cuatro variables de salida y dos capas ocultas con tres neuronas cada una

1. **Neurona de entrada:** Una neurona que recibe entradas externas desde fuera de la red.
2. **Neurona de salida:** Una neurona que produce algunas de las salidas de la red.
3. **Neurona oculta:** Una neurona que no tiene interacción directa con el “mundo exterior” sino solo con otras neuronas dentro de la red. Una terminología similar se utiliza a nivel de capa para redes neuronales multicapa.

Como se aprecia en la Figura 6.3, la disposición de las neuronas en una red neuronal artificial se lleva a cabo mediante la formación de niveles que contienen un número específico de neuronas. Cuando un conjunto de neuronas artificiales recibe simultáneamente el mismo tipo de información, se le denomina capa. Además, se hace referencia a una red compuesta por tres tipos de niveles como capas. La Figura 6.4 exhibe otras seis redes con diversos números de capas, y la mitad de ellas (Figura 6.4a, Figura 6.4c, Figura 6.4e) son univariadas, ya que la variable de respuesta a predecir es única, mientras que la otra mitad (Figura 6.4b, Figura 6.4d, Figura 6.4f) son multivariadas, dado que la red tiene el propósito de predecir dos salidas. Es relevante destacar que los paneles a y b en la Figura 6.4 representan redes con solo una capa y sin capas ocultas; por consiguiente, este tipo de redes corresponde a modelos convencionales de regresión o clasificación por regresión.

En consecuencia, la arquitectura de una red neuronal artificial se refiere a la manera en que las neuronas están organizadas en la red, y está estrechamente vinculada al algoritmo de aprendizaje empleado para entrenar la red. Según el número de capas, clasificamos las redes como monocapa o multicapa; y si consideramos la dirección del flujo de información como criterio clasificadorio, las redes se denominan de avance o recurrentes. Cada tipo de arquitectura se aborda la siguiente sección.

6.2. Arquitectura

6.2.1. Perceptrón simple

El *perceptrón simple* consta de cuatro componentes fundamentales en su estructura. Estos son: las entradas (input) con conexiones y pesos (nodos ponderados), el nodo de procesamiento o suma, la función de activación y las salidas (output). El nodo de procesamiento realiza una regresión lineal, involucrando la suma ponderada de los pesos en cada nodo de las entradas y un término de sesgo o término independiente. En esencia, el perceptrón simple funciona como un discriminador lineal que, a partir de un umbral establecido, produce una salida binaria.

Desde una perspectiva matemática, el perceptrón simple se representa mediante la siguiente ecuación:



Figura 6.4.: Diferentes estructuras de redes neuronales univariadas y multivariadas.

$$\hat{\mathbf{y}}(\mathbf{x}) = f(\mathbf{w}^T \mathbf{x} + b). \quad (6.1)$$

La arquitectura que modela esta ecuación se describe a través de la Figura 6.5.

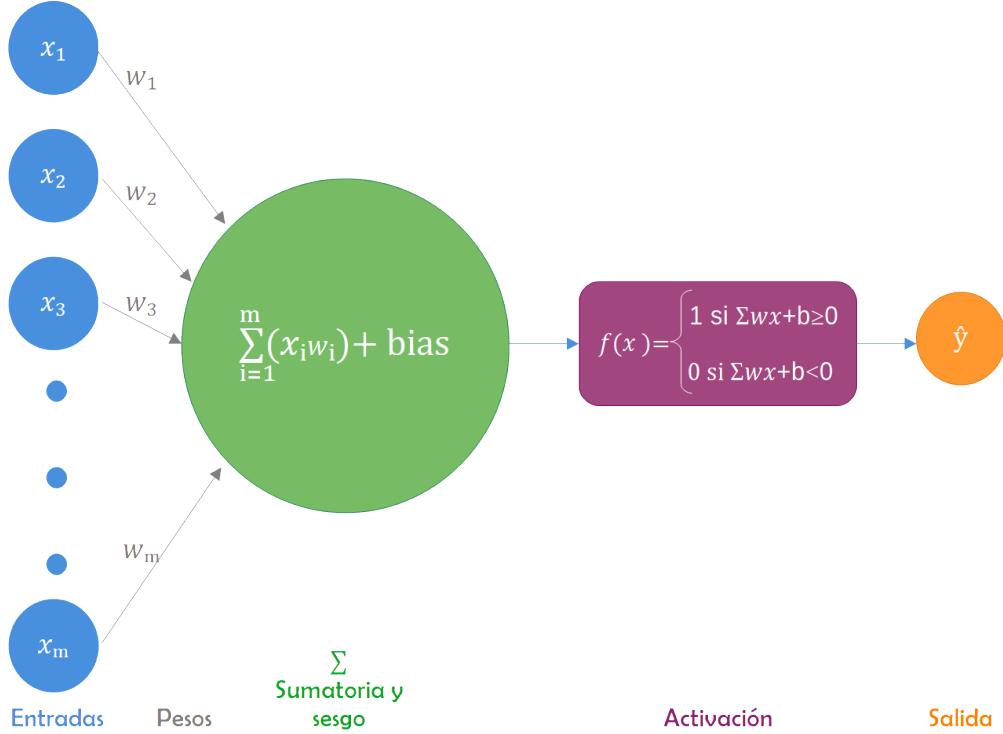


Figura 6.5.: Arquitectura de un perceptrón simple

donde \mathbf{x} denota el vector de entradas, \mathbf{w} denota el vector de pesos asociados a cada nodo, b denota el sesgo o intercepto de la regresión, \sum denota el nodo de procesamiento o combinador lineal, y f denota la función de activación o función limitadora, siendo esta última una transformación no lineal de la regresión obtenida en el nodo de procesamiento.

Aunque el perceptrón simple demuestra eficacia en el aprendizaje y la resolución de problemas linealmente separables, como las compuertas lógicas *AND* (Figura 6.6c) y *OR* (Figura 6.6a), presenta limitaciones en la resolución de problemas que no son de este tipo. Un ejemplo paradigmático de ello es su incapacidad para clasificar las salidas de una compuerta lógica del tipo *XOR* (Figura 6.6e), ya que el nodo de procesamiento solo permite la separación de la información mediante una única recta de regresión.

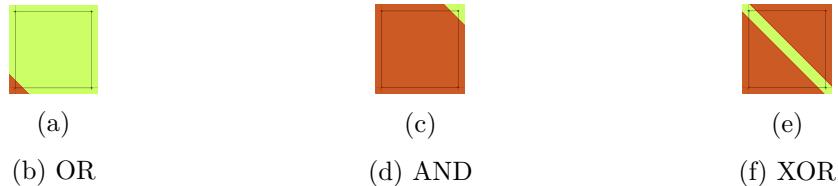


Figura 6.6.: Compuertas lógicas

6.2.2. Perceptrón Multicapa (MLP)

La solución al problema de la puerta lógica *XOR* consiste en la adición de una neurona adicional, permitiendo así la definición de una nueva recta de regresión, como se ilustra en la Figura 6.6e. Esto conduce a la creación de lo que se conoce como *Perceptrón Multicapa* o *MLP* (por sus siglas en inglés), también reconocido como *Red Neuronal Profunda*. Esta estructura representa una generalización del perceptrón simple, incorporando más de un nivel de neuronas y/o una o varias capas de neuronas “entre” la capa de entradas y la capa de salidas, las cuales son denominadas capas ocultas. En estas capas ocultas, las funciones de activación entre las neuronas no son necesariamente lineales. Las MLP son consideradas las redes neuronales artificiales por defecto y se representan mediante un diagrama simple, que transmite las entradas de capa en capa hasta alcanzar la capa final.

La red neuronal de la Figura 6.7 ejemplifica un MLP de dos capas ocultas. En esta representación, los superíndices indican la posición en las capas, mientras que los subíndices indican la posición relativa de cada nodo en su respectiva capa. La red consta de un vector de entradas $\mathbf{x} \in \mathbb{R}^{d_0}$, donde $\mathbf{x} = (x_1, \dots, x_{d_0})^T$, capas ocultas denotadas por a^l , y un vector de salidas $\hat{\mathbf{y}} \in \mathbb{R}^{d_L}$ con $\hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_{d_L})^T$. Las capas ocultas contienen nodos de procesamiento o neuronas representadas por $a = f(z)$, donde f es la función de activación de cada capa y z es un combinador lineal (matricial). Las conexiones entre las capas están ponderadas por \mathbf{w} , que representa las matrices de pesos asociadas en cada capa. Por ejemplo, w_{ij}^1 representa el peso asociado a la conexión entre la entrada j -ésima y el i -ésimo nodo de procesamiento en la primera capa oculta. Las matrices \mathbf{w}^l tienen dimensiones $(d_l \times d_{l-1})$, donde la capa de entrada se considera como capa cero ($l = 0$).

Es importante destacar que el término b , que indica el sesgo en el perceptrón simple, también se incluye en la red MLP en cada nodo de procesamiento, específicamente en el combinador lineal z . A partir de este momento, el nodo de procesamiento incorporará el término de sesgo b , y $b^1 \in a^1$ denominará el vector de sesgo en la primera capa oculta.

La ecuación matemática que describe la red de la Figura 6.7 es la siguiente:

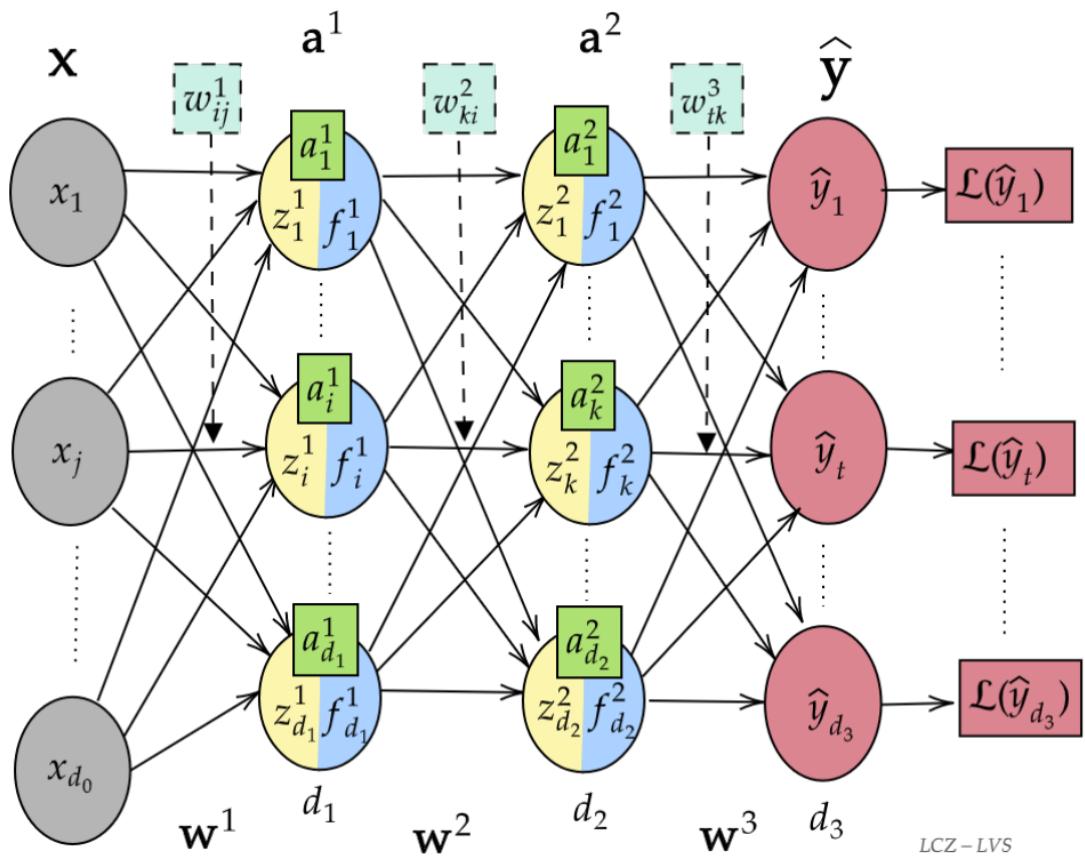


Figura 6.7.: Red neuronal MLP con dos capas ocultas (Sosa Jerez, Zamora Alvarado, et al. (s. f.)).

$$\begin{aligned}
\hat{\mathbf{y}} &= f^3(z^3) \\
&= f^3(\mathbf{w}^3 a^2 + b^3) \\
&= f^3(\mathbf{w}^3(f^2(z^2)) + b^3) \\
&= f^3(\mathbf{w}^3(f^2(\mathbf{w}^2 a^1 + b^2)) + b^3) \\
&= f^3(\mathbf{w}^3(f^2(\mathbf{w}^2(f^1(z^1)) + b^2)) + b^3) \\
&= f^3(\mathbf{w}^3(f^2(\mathbf{w}^2(f^1(\mathbf{w}^1 \mathbf{x} + b^1)) + b^2)) + b^3)
\end{aligned}$$

Se observa que f^3 representa la función de activación en la capa de salida, la cual comúnmente se elige como la identidad. Sin embargo, en algunos modelos de clasificación, la predicción puede ser más precisa si esta función es no lineal y limitadora.

Por otro lado, la última columna en la Figura 6.7 constituye una capa adicional en la cual, a través de una función de pérdida, se evalúa el rendimiento de la red. Esta evaluación relaciona la información obtenida en la capa de salida con los datos esperados en un modelo de aprendizaje supervisado.

6.3. Perceptrón

En un principio, se establece un conjunto de datos a estudiar denominado $\mathbf{X} \subseteq \mathbb{R}^{m+1}$. Este conjunto se partitiona en dos clases linealmente separables, \mathcal{C}_1 y \mathcal{C}_2 . A los vectores $\mathbf{x} = (x_1, x_2, \dots, x_m, 1)^T$ que pertenecen a \mathbf{X} , se les denomina *entradas*.

A continuación, se introduce un conjunto $\mathbf{W} \subseteq \mathbb{R}^{m+1}$, que contiene etiquetas para los nodos del perceptrón. Los elementos de este conjunto, denotados como $\mathbf{w} = (w_1, w_2, \dots, w_m, b)^T$, se llaman *pesos sinápticos*. Aquí, b es un número real fijo conocido como *sesgo*. Con el propósito de describir el algoritmo del Perceptrón, se presentan cuatro definiciones fundamentales:

Definición 6.3 (Clases linealmente separables). Sean \mathcal{C}_1 y \mathcal{C}_2 dos clases en un espacio n -dimensional. \mathcal{C}_1 y \mathcal{C}_2 se consideran clases linealmente separables si existe un vector $\mathbf{w} \in \mathbb{R}^{m+1}$ de pesos sinápticos que cumple con las siguientes condiciones:

$$\begin{aligned}
\mathbf{w}^T \mathbf{x}_1 &> 0 \text{ para cada vector de entrada } \mathbf{x}_1 \in \mathcal{C}_1. \\
\mathbf{w}^T \mathbf{x}_2 &\leq 0 \text{ para cada vector de entrada } \mathbf{x}_2 \in \mathcal{C}_2.
\end{aligned}$$

Definición 6.4 (Combinador lineal). Dados $\mathbf{x} = (x_1, x_2, \dots, x_m, 1)^T$ y $\mathbf{w} = (w_1, w_2, \dots, w_m, b)^T$, se define la función $\mathcal{V} : \mathbb{R}^{m+1} \times \mathbb{R}^{m+1} \rightarrow \mathbb{R}$ como $\mathcal{V}(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x}$, donde $\mathcal{V}(\mathbf{x}, \mathbf{w}) = 0$ representa el hiperplano de separación entre dos regiones de decisión.

Definición 6.5 (Función limitadora). Sea \mathcal{A} el conjunto de todas las combinaciones lineales $\mathcal{V}(\mathbf{x}, \mathbf{w})$. Considerando $t \in \mathcal{A}$, se define la función limitadora g como sigue:

$$g : \mathcal{A} \rightarrow \{1, -1\}$$

$$t \rightarrow g(t) = \begin{cases} 1 & \text{si } t > 0 \\ -1 & \text{si } t \leq 0 \end{cases}$$

Definición 6.6 (Función perceptrón). Dadas $\mathcal{V}(\mathbf{x}, \mathbf{w})$ y $g(t)$, se define la aplicación clasificadora $\mathcal{P} : \mathbb{R}^{m+1} \times \mathbb{R}^{m+1} \rightarrow \{1, -1\}$ como $\mathcal{P}(\mathbf{x}, \mathbf{w}) = g(\mathcal{V}(\mathbf{x}, \mathbf{w})) = \hat{y}$, donde $\hat{y} \in \{-1, 1\}$ es la salida de la función perceptrón. Además, la aplicación perceptrón posee una representación gráfica mediante un dígrafo simple, como se muestra en la Figura 6.5.

A través de la función establecida en la Definición 6.6, se desarrolla un modelo de aprendizaje supervisado de clasificación binaria denominado *Perceptrón*. Este modelo involucra las funciones previamente definidas con el objetivo de clasificar correctamente un conjunto de entradas \mathbf{X} , linealmente separables en dos clases. Se aplica una regla de aprendizaje adaptativa sobre cada uno de los pesos sinápticos (\mathbf{w}) en una cantidad finita de pasos (n), proceso conocido como *algoritmo de aprendizaje del Perceptrón*.

Definición 6.7 (Combinador lineal del perceptrón). Considerando las entradas y los pesos sinápticos en el perceptrón, $\mathbf{x}(n) = (x_1(n), x_2(n), \dots, x_m(n), 1)^T$ y $\mathbf{w}(n) = (w_1(n), w_2(n), \dots, w_m(n), b)^T$, se define el combinador lineal del perceptrón como

$$\mathcal{V} = \mathbf{w}^T(n)\mathbf{x}(n),$$

donde n denota el número de iteraciones en la aplicación del algoritmo.

Se considera $\mathcal{H} \subset \mathbf{X}$ como el subespacio vectorial de entrenamiento. \mathcal{H}_1 es el subespacio de vectores de entrenamiento $\mathbf{x}_1(1), \mathbf{x}_1(2), \dots$ que pertenecen a la clase \mathcal{C}_1 , y \mathcal{H}_2 es el espacio de vectores de entrenamiento $\mathbf{x}_2(1), \mathbf{x}_2(2), \dots$ que pertenecen a la clase \mathcal{C}_2 . Se define $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$. Con el fin de evitar un sobreentrenamiento en alguna de las dos clases, se garantiza que \mathcal{H}_1 y \mathcal{H}_2 tengan la misma cardinalidad.

Dado que el perceptrón es un modelo de aprendizaje supervisado, se establece $y(k) \in \{-1, 1\}$ como la clase a la que realmente pertenece cada entrada $x(k)$ de \mathcal{H} . Se observa que el valor $y(k) - \hat{y}(k)$ representa el error cometido por el Perceptrón en su clasificación, y de este error se deriva la siguiente definición:

Definición 6.8 (Función actualización por corrección del error). Se define la regla de actualización de los pesos sinápticos como sigue:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta(n)[y(n) - \hat{y}(n)]x(n).$$

De esta manera,

$$\mathbf{w}(n+1) = \begin{cases} \mathbf{w}(n) + 2\eta(n)x(n) & \text{si } y(n) = 1 \text{ y } \hat{y} = -1, \\ \mathbf{w}(n) & \text{si } y(n) = \hat{y}(n), \\ \mathbf{w}(n) - 2\eta(n)x(n) & \text{si } y(n) = -1 \text{ y } \hat{y} = 1, \end{cases}$$

donde $\eta(n) = \eta > 0$ es una regla de adaptación de incremento fijo llamada *tasa de aprendizaje*.

6.3.1. Teorema de convergencia del perceptrón

Teorema 6.1. Sean \mathcal{H}_1 y \mathcal{H}_2 , subconjuntos de vectores de entrenamiento linealmente separables. Consideré las m entradas presentadas al perceptrón, como elementos de estos dos subconjuntos. El perceptrón converge después de n_0 iteraciones, en el sentido que:

$$w(n_0) = w(n_0 + 1) = w(n_0 + 2) = \dots,$$

es un vector solución para $n_0 \leq n_{\max}$.

Prueba. Vea Sosa Jerez, Zamora Alvarado, et al. (s. f.). □

6.4. Funciones de activación

La asignación entre las entradas y una capa oculta en una Red Neuronal Artificial (RNA) y una Red Neuronal Profunda (RNP) es determinada por funciones de activación. Dichas funciones propagan la información generada mediante la combinación lineal de los pesos y las entradas hacia la siguiente capa, incluyendo la capa de salida. Como se ha mencionado anteriormente, existe una analogía entre las neuronas biológicas y las redes neuronales artificiales; en este contexto, las funciones de activación son análogas a la tasa del potencial de acción disparado en el cerebro.

Las funciones de activación son transformaciones de funciones escalares a escalares que proporcionan una salida específica para cada neurona. Estas funciones introducen no linealidades en las capacidades de modelado de la red. La función de activación de una

neurona (nodo) define la forma funcional de su activación. Por ejemplo, si se define una función de activación lineal como $g(z) = z$, en este caso, el valor de la neurona sería la entrada cruda x multiplicada por el peso aprendido, representando así un modelo lineal. A continuación, se describen las funciones de activación más populares.

6.4.1. Lineal

La Figura 6.8a exhibe una función de activación lineal que es esencialmente la función identidad. Esta se define como

$$F(x) = Wx + b,$$

donde la variable dependiente mantiene una relación directa y proporcional con la variable independiente. En términos prácticos, esto implica que la función transmite la señal sin cambios. Sin embargo, el inconveniente al utilizar funciones de activación lineales radica en que esto no permite aprender formas funcionales no lineales.

6.4.2. Sigmoide

La función de activación sigmoide desempeña el papel de un mecanismo que transforma variables independientes, abarcando un rango prácticamente infinito, en probabilidades situadas dentro del intervalo de 0 a 1. La mayor concentración de su producción tiende a agruparse estrechamente alrededor de los valores 0 o 1. Funcionando como una transformación logística, los sigmoides exhiben la capacidad de mitigar valores extremos o atípicos en los datos sin eliminarlos. Las ecuaciones que describen la función sigmoidal y su derivada son las siguientes:

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad \sigma'(x) = \frac{e^{-x}}{(1 + e^{-x})^2}.$$

Ampliamente utilizada en la construcción de Redes Neuronales Artificiales (RNA) y Redes Neuronales Profundas (DNN), especialmente en escenarios donde el resultado deseado es una probabilidad o un resultado binario, la función de activación sigmoide representa uno de los tipos más frecuentemente empleados.

La función de activación $\sigma(x) : \mathbb{R} \rightarrow [0, 1]$ se caracteriza por ser una función suave y diferenciable en todo punto. Compacta cualquier valor entre 0 y 1 y destaca por su naturaleza estrictamente creciente, logrando un delicado equilibrio entre comportamiento lineal y no lineal. Sin embargo, es susceptible de experimentar “atascos”, un fenómeno en el cual los valores de salida convergen muy cerca de 1 o 0, especialmente cuando los valores de entrada son muy positivos o negativos (consulte la Figura 6.8b). Al referirnos a que la función de activación se “atasca”, implicamos que el proceso de aprendizaje deja

de mejorar debido al dominio de valores de salida grandes o pequeños dentro de esta función de activación.

6.4.3. Unidad lineal rectificadora (ReLU)

La función de activación de la unidad lineal rectificadora (ReLU) destaca como una de las más adoptadas. Exhibe una respuesta plana por debajo de un umbral especificado, normalmente establecido en cero, y luego se vuelve lineal. La activación en una ReLU se produce solo cuando la entrada supera un determinado umbral. Cuando la entrada está por debajo de cero, la salida sigue siendo cero, pero al exceder el umbral, como se ilustra en la Figura 6.8c., establece una relación lineal con la variable dependiente, de la siguiente manera

$$F(x) = \max(0, x)$$

A pesar de su aparente simplicidad, la función de activación de ReLU facilita las transformaciones no lineales, lo que permite la aproximación de funciones no lineales arbitrarias mediante el uso de rectificadores lineales suficientes. Esto contrasta con los escenarios en los que se emplean exclusivamente funciones de activación lineal.

En la actualidad, las ReLU representan el estado de la técnica, demostrando su eficacia en diversas situaciones. Sin embargo, debido a que el gradiente de la ReLU es cero o una constante, plantea desafíos en el control de problemas como la desaparición y la explosión de gradientes, comúnmente conocido como el problema de la “ReLU moribunda”. En particular, las funciones de activación de ReLU han mostrado un rendimiento de entrenamiento superior en la práctica en comparación con las funciones de activación sigmoide. Esta función de activación se emplea más comúnmente en capas ocultas y capas de salida cuando la variable de respuesta es continua y supera cero.

6.4.4. ReLu con fugas

Las ReLU con fugas sirven como medida correctiva para abordar el fenómeno de la “ReLU moribunda”. A diferencia de la ReLU convencional, que asigna un valor cero a la función cuando $x < 0$, la ReLU con fugas introduce una pequeña pendiente negativa, denotada como α , donde α es un valor escalar dentro del rango de 0 a 1 (consulte la Figura 6.8d). Si bien esta variación de ReLU ha demostrado cierto éxito en aplicaciones prácticas, los resultados no son uniformemente consistentes. La expresión matemática de esta función de activación se proporciona a continuación:

$$F(x) = \begin{cases} x & \text{si } x > 0 \\ \alpha x & \text{otro caso} \end{cases}.$$

6.4.5. Tangente hiperbólica

La función de activación tangente hiperbólica (\tanh) es una modificación de la función sigmoide y se define como

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Cuyas gráficas se observan en la Figura 6.8e. La función de activación tangete hiperbólica $\tanh(x) : \mathbb{R} \rightarrow [-1, 1]$ es una función suave y diferenciable en todo punto. Similar a la función de activación sigmoide, produce una salida sigmoidal (en forma de "S"). Sin embargo, la función \tanh tiene la ventaja de ser menos propensa al problema de "atascarse" en comparación con la función de activación sigmoide. Esto se atribuye a que los valores de salida de la función \tanh se encuentran dentro del rango de -1 a 1 . En consecuencia, a menudo se prefiere la función de activación \tanh para capas ocultas. Una ventaja adicional de \tanh es su capacidad para manejar los números negativos de manera más efectiva. Sin embargo, el gradiente de la función evaluado en valores muy alejados al origen será un valor muy pequeño, por lo que sigue generando un estancamiento en el proceso de retropropagación.

6.4.6. Softmax

La función Softmax se emplea predominantemente en redes neuronales dedicadas a abordar problemas de clasificación. Su resultado proporciona un porcentaje que indica la probabilidad de que los datos ingresados pertenezcan a cada una de las clases. Es habitual utilizar esta función de activación en las capas finales de la red neuronal. La expresión que la define es:

$$S = \frac{e^{a_i^l}}{\sum_{k=1}^K (e^{a_k^l})}, \text{ para } i = 1, \dots, K$$

donde a es la salida de las capas ocultas y K es el número de clases en el modelo. La Figura 6.8f ejemplifica esta función de activación.

6.5. Funciones de coste

Las funciones de costo, pérdida u objetivo desempeñan un papel fundamental al medir la disparidad entre los resultados obtenidos y los valores deseados. En el contexto del descenso del gradiente, estas funciones son cruciales, ya que buscan minimizar la salida de la función de costo, lo que lleva a que los valores generados por la red neuronal sean cercanos a los valores deseados.

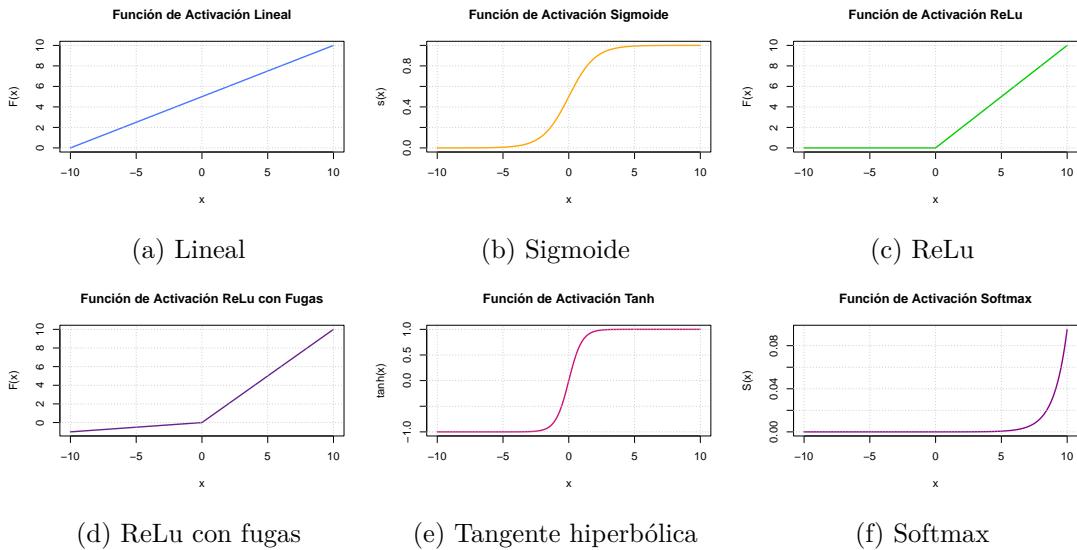


Figura 6.8.: Funciones de activación

Para ser empleada en el proceso de retropropagación, la función de costo debe cumplir con dos propiedades fundamentales:

1. La función de costo C debe expresarse como un promedio:

$$C = \frac{1}{n} \sum_x \mathcal{L}_x,$$

donde \mathcal{L}_x representa las funciones de pérdida para ejemplos individuales x en el conjunto de entrenamiento.

2. La función de costo C no debe depender de ningún valor de activación, excepto los valores de salida a^L . Si la función de costo depende de otras capas de activación además de la capa de salida, la retropropagación no será válida, ya que la idea de propagación hacia atrás dejará de funcionar.

Observación. Es importante destacar que la función de costo y la función de pérdida son conceptos distintos. La función de costo representa el promedio de las pérdidas de todas las muestras o datos de entrenamiento, mientras que la función de pérdida se refiere a las pérdidas individuales para cada ejemplo. A pesar de esta diferencia, es común observar el uso de ambos términos de manera intercambiable o con propósitos similares en la literatura.

6.6. Gradiente descendente

El gradiente o vector gradiente se presenta como una generalización de la derivada en varias variables, su definición formal se muestra a continuación.

Definición 6.9 (Vector gradiente). Sea $f : U \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ una función diferenciable definida en el conjunto abierto $U \in \mathbb{R}^n$. Se define el vector gradiente de la función f en el punto x_0 de U , denotado por $\nabla f(x_0)$, como el vector en \mathbb{R}^n dado por

$$\nabla f(x_0) = \left(\frac{\partial f}{\partial x_1}(x_0), \frac{\partial f}{\partial x_2}(x_0), \dots, \frac{\partial f}{\partial x_n}(x_0) \right).$$

Adicionalmente, el vector gradiente señala la dirección en la cual la función f experimenta el crecimiento más rápido. Este resultado se formaliza mediante el siguiente teorema

Teorema 6.2. *Sea $f : X \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ diferenciable en $x_0 \in X$, el gradiente apunta hacia la dirección de mayor crecimiento de f .*

Prueba. Vea Stewart (2017). □

El método de descenso del gradiente desempeña un papel fundamental en el entrenamiento de las redes neuronales. A través de este método, se logra considerar los valores más óptimos y eficaces, específicamente los pesos w de la red neuronal. Este enfoque permite estimar cada nuevo parámetro basándose en el anterior, teniendo en cuenta la derivada de la función de coste. Además, el proceso presenta ventajas como la simplicidad y la rapidez de convergencia.

6.6.1. Algoritmo gradiente descendente

Considere una función de costo \mathcal{C} definida como $\mathcal{C} : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$. El algoritmo de gradiente descendente es utilizado para encontrar un valor w en Ω tal que $\mathcal{C}(w)$ alcance un mínimo (extremo local).

Las actualizaciones de w se realizan de la siguiente manera:

$$w_{k+1} = w_k - \alpha \nabla \mathcal{C}(w_k),$$

donde α es la tasa de aprendizaje y k es el número de iteraciones. Se elige inicialmente un valor inicial w_0 (puede ser seleccionado de forma aleatoria o elegido manualmente). El algoritmo comienza en este punto con el propósito de ajustar el valor del peso inicial hasta situarlo en el mínimo de la función.

6.7. Perceptrón Multicapa

Como se expuso previamente en la Sección 6.2.2, se pueden representar mediante un diagrama simple, que incluye nodos ponderados y un conjunto de atributos que las caracterizan. En esta sección, la estructura de la red será formalizada junto con sus definiciones correspondientes (Sosa Jerez, Zamora Alvarado, et al. (s. f.)).

Definición 6.10 (Perceptrón Multicapa (MLP)). Una red neuronal artificial MLP se define formalmente como una tripla $\langle \mathcal{D}, \{f\}, \mathcal{A} \rangle$, donde;

- \mathcal{D} es un dígrafo contable, localmente finito, con nodos etiquetados. Sus vértices corresponden a los nodos de procesamiento (neuronas), mientras que las etiquetas de los nodos, denominadas *pesos*, representan las intensidades de las conexiones sinápticas. Dichas intensidades se denotan por w_{ij} , indicando el peso de la conexión entre la neurona j -ésima y la i -ésima.
- \mathcal{A} es el conjunto que contiene los elementos de “entrada” de las unidades o nodos de procesamiento, generalmente representado por $A = \mathbb{R}$.
- $\{f : \mathcal{A} \rightarrow \mathcal{A}\}$, es una colección de funciones de activación.

En el dígrafo \mathcal{D} , se definen las capas como las columnas de vértices en \mathcal{D} . Cada una de estas columnas puede ser representada matemáticamente a través de un vector, de la siguiente manera:

1. **Capa de entrada:** corresponde a la primera columna de vértices de \mathcal{D} , cuya representación matemática se expresa como:

$$\mathbf{x} = (x_1, \dots, x_{d_0})^T \text{ donde } \mathbf{x} \in \mathbb{R}^{(d_0 \times 1)}$$

2. **Capa de salida:** corresponde a la última columna de vértices de \mathcal{D} , cuya representación matemática se describe como:

$$\hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_{d_L})^T \text{ donde } \hat{y} \in \mathbb{R}^{(d_L \times 1)}$$

3. **Capas ocultas:** corresponden a las columnas intermedias entre la capa de entrada y la de salida. Su representación matemática está dada por:

$$\begin{aligned} \mathbf{a}^l &= (a_1^l, \dots, a_{d_l}^l)^T \text{ donde } a^l \in \mathbb{R}^{(d_l \times 1)} \\ &= f^l(z^l) \text{ con } l = 1, \dots, L - 1 \end{aligned}$$

Cabe destacar que cada \mathbf{a}^l corresponde a una columna de vértices en \mathcal{D} , donde L denotará la totalidad de capas en la red. f^l será una función de activación vectorial y z^l será el combinador lineal matricial, ambos en la capa l . De esta manera, la capa de salida también puede representarse como el vector \mathbf{a}^L , y la capa de entrada como el vector \mathbf{a}^0 .

Definición 6.11 (Neuronas o nodos de procesamiento). Las neuronas de la red MLP son los vértices de las capas ocultas en \mathcal{D} , es decir, las componentes de \mathbf{a}^l se denotarán como a_i^l , donde:

$$a_i^l = f^l(z_i^l).$$

Definición 6.12 (Función de activación). Se define f^l como una función de activación vectorial, de modo que:

$$f^l : \mathbb{R}^{(d_1 \times 1)} \rightarrow \mathbb{R}^{(d_1 \times 1)}.$$

Definición 6.13 (Matriz de pesos). Para cada capa l en \mathcal{D} , se define \mathbf{w}^l como una matriz de dimensiones $d_l \times d_{l-1}$, donde d_l representa la cantidad de neuronas en la capa l , de la siguiente manera:

$$\mathbf{w}^l = \begin{bmatrix} w_{11}^l & \cdots & w_{1j}^l & \cdots & w_{1d_{l-1}}^l \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ w_{i1}^l & \cdots & w_{ij}^l & \cdots & w_{id_{l-1}}^l \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ w_{d_l 1}^l & \cdots & w_{d_l j}^l & \cdots & w_{d_l d_{l-1}}^l \end{bmatrix}$$

Definición 6.14 (Sesgo). Se define el sesgo como el vector

$$\mathbf{b}^l = (b_1^l, \dots, b_{d_l}^l)^T \quad \text{con } \mathbf{b}^l \in \mathbb{R}^{(d_1 \times 1)}$$

correspondiente a la capa l , cuyas entradas son el parámetro de sesgo de cada neurona.

Definición 6.15 (Combinador lineal matricial). Dados $\mathbf{a}^{l-1}, \mathbf{w}^l$ y \mathbf{b}^l se define el combinador lineal como

$$\begin{aligned} \mathbf{z}^l &= (z_1^l, \dots, z_{d_l}^l)^T \\ &= \mathbf{w}\mathbf{a}^{l-1} + \mathbf{b}^l, \end{aligned}$$

donde

$$z_i^l = \sum_{j=1}^{d_{l-1}} w_{ij}^l a_j^{l-1} + b_i. \tag{6.2}$$

Se observa que la ecuación Ecuación 6.2 guarda una fuerte relación con la Definición 6.4. No obstante, en el caso de \mathbf{z}^l , se ha incorporado el vector de parámetros de sesgo \mathbf{b}^l .

Definición 6.16 (Función de pérdida). Se define $\mathcal{L} : \mathbb{R}^n \rightarrow \mathbb{R}$ de manera que

$$\begin{aligned}\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) &= \frac{1}{2} \|\mathbf{y} - \hat{\mathbf{y}}\|^2 \\ &= \frac{1}{2} \|\mathbf{y} - \mathbf{a}^L\|^2 \\ &= \frac{1}{2} \sum_{r=1}^{d_L} (y_r - a_r^L)^2,\end{aligned}$$

como la función de pérdida de la red MLP.

Definición 6.17 (Conjunto de datos de entrenamiento). Sea $\mathbf{X} = (\mathbf{x}(1), \dots, \mathbf{x}(n))$, donde $\mathbf{x}(k)$, con $k = 1, \dots, n$, representa el k -ésimo dato en el conjunto \mathbf{X} , siendo este el vector de entradas de la red neuronal en la k -ésima etapa.

Definición 6.18 (Conjunto de salidas de la red). Se define $\hat{\mathbf{Y}} = (\hat{\mathbf{y}}(1), \dots, \hat{\mathbf{y}}(n))$, donde $\hat{\mathbf{y}}(k)$, con $k = 1, \dots, n$, representa el k -ésimo dato en el conjunto $\hat{\mathbf{Y}}$, siendo este el vector de salidas de la red neuronal en la k -ésima etapa.

Definición 6.19 (Resultados esperados). Se define $\mathbf{Y} = (\mathbf{y}(1), \dots, \mathbf{y}(n))$, donde $\mathbf{y}(k)$, con $k = 1, \dots, n$, representa el k -ésimo dato en el conjunto \mathbf{Y} , siendo este el vector de resultados esperados correspondiente al dato $\mathbf{x}(k)$.

6.7.1. Entrenamiento y aprendizaje del Perceptrón Multicapa

El proceso de aprendizaje de una red neuronal se configura como un modelo de aprendizaje supervisado. En este proceso, se establece un algoritmo que, a partir de un conjunto de datos de entrenamiento que incluye entradas y resultados esperados, permite el entrenamiento gradual de la red. El objetivo principal es que la red pueda calcular de manera autónoma los valores óptimos de pesos y sesgos para clasificar las entradas en salidas, minimizando la discrepancia con respecto a los resultados esperados.

Al concluir este proceso de entrenamiento, se espera que la red neuronal desarrolle la capacidad de clasificar cualquier dato, incluso aquellos no presentes en el conjunto de entrenamiento inicial (datos de prueba), generando salidas con un error de clasificación mínimo. Este proceso de entrenamiento se compone de dos etapas esenciales: la propagación hacia adelante o *feedforward*, y la retropropagación, también conocida como *back-propagation*.

6.7.1.1. Propagación hacia adelante

El proceso de prealimentación constituye la base del entrenamiento y aprendizaje de la red, considerando los siguientes pasos:

1. Se elige un vector de datos $\mathbf{x} \in \mathbf{X}$ como entrada de la red neuronal MLP.
2. Se establecen matrices \mathbf{w}^l de pesos y vectores \mathbf{b}^l de sesgo, cuyas componentes tienen entradas aleatorias que pertenecen a un umbral prefijado.
3. Se “alimenta” la red neuronal en una única dirección. Para ello, se inicia estableciendo lo que se tendrá en la primera capa de procesamiento y luego se generaliza el proceso:
 - **Alimentación primera capa:** Se establecen los productos matriciales en cada nodo de procesamiento, dados por:

$$\begin{aligned}\mathbf{z}^1 &= (\mathbf{w}^1 \mathbf{x}) + \mathbf{b}^1 \\ \mathbf{a}^1 &= f^1(\mathbf{z}^1).\end{aligned}$$

- **Generalización:** Considerando cómo se “alimenta” la red en la primera capa, se repite el mismo proceso para cada capa siguiente:

$$\begin{aligned}\mathbf{z}^l &= (\mathbf{w}^l \mathbf{a}^{l-1}) + \mathbf{b}^l \\ \mathbf{a}^l &= f^l(\mathbf{z}^l).\end{aligned}$$

Se observa que en este paso, lo que en la primera capa era \mathbf{x} , en cualquier capa diferente será \mathbf{a}^{l-1} . Esto se debe a que la red es un dígrafo \mathcal{D} , donde la salida de la capa anterior ($l - 1$) se convierte en el vector de entrada para la capa siguiente (l).

6.7.1.2. Retropropagación

La retropropagación se emplea en las redes neuronales como algoritmo de aprendizaje, y su objetivo es ajustar de manera eficiente los pesos de la red. Este proceso consiste en establecer inicialmente de manera aleatoria los pesos requeridos en la red para obtener una salida, la cual se compara mediante la función de pérdida \mathcal{L} con el resultado esperado. De esta manera, se calcula el error de aproximación de la red con el objetivo de minimizar dicho error a través de la optimización de la función \mathcal{L} . La optimización se realiza mediante una generalización del algoritmo de descenso del gradiente, utilizando la regla de la cadena y recorriendo la red de atrás hacia adelante.

De forma iterativa, la red aprende a establecer los pesos y sesgos adecuados para cada neurona, con el fin de obtener una salida que se aproxime al resultado esperado. Para comprender el funcionamiento del algoritmo de retropropagación, es necesario comenzar calculando las derivadas respecto a los parámetros de pesos y sesgos de la función de coste en la red prealimentada.

Se inicia calculando la derivada de \mathcal{L} respecto a uno de los pesos que afectan a la última capa:

$$\begin{aligned}
 \frac{\partial \mathcal{L}}{\partial w_{ij}^L} &= \frac{1}{2} \sum_{r=1}^{d_L} \frac{\partial}{\partial w_{ij}^L} (y_r - a_r^L)^2 \\
 &= \sum_{r=1}^{d_L} (a_r^L - y_r) \left(\frac{\partial a_r^L}{\partial w_{ij}^L} \right) \\
 &= \sum_{r=1}^{d_L} (a_r^L - y_r) \frac{\partial}{\partial w_{ij}^L} f^L(z_r^L) \\
 &= \sum_{r=1}^{d_L} (a_r^L - y_r) \frac{\partial}{\partial w_{ij}^L} f^L \left(\sum_{t=1}^{d_L} w_{rt}^L a_t^{L-1} + b_r^L \right),
 \end{aligned} \tag{6.3}$$

Esta expresión se anula en todos los valores en los que $r \neq i$ o $t \neq j$. Si $r = i$ y $t = j$, se tiene:

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^L} = (a_i^L - y_i) f^{(1)L}(z_i^L) a_j^{L-1} \tag{6.4}$$

Esta ecuación proporciona la derivada particular de la función de pérdida respecto a un único peso. Para generalizar esta situación y calcular $\frac{\partial \mathcal{L}}{\partial \mathbf{w}^L}$, se deben tener en cuenta las dimensiones y definir una nueva operación matricial.

Definición 6.20 (Producto Hadamard). Dadas A, B dos matrices de dimensión $(m \times n)$, el producto de Hadamard ($A \odot B$) es una matriz de dimensión $(m \times n)$ tal que:

$$(A \odot B)_{ij} = [a_{ij} b_{ij}].$$

Generalizando la Ecuación 6.4 y haciendo uso de la definición previa, se puede expresar la derivada parcial de la función de pérdida respecto a los pesos en la última capa como:

$$\begin{aligned}
 \frac{\partial \mathcal{L}}{\partial \mathbf{w}^L} &= \frac{\partial \mathcal{L}}{\partial \mathbf{a}^L} \frac{\partial \mathbf{a}^L}{\partial \mathbf{z}^L} \frac{\partial \mathbf{z}^L}{\partial \mathbf{w}^L} \\
 &= [(a^L - y) \odot f^{(1)L}(\mathbf{z}^L)] (\mathbf{a}^{L-1})^T,
 \end{aligned} \tag{6.5}$$

Donde el error en la última capa se denota como $(\mathbf{a}^L - y)$ y se representa como \mathbf{e}^L . También se introduce la notación δ^L para referirse al producto de Hadamard $[(\mathbf{a}^L - y) \odot f^{(1)L}(\mathbf{z}^L)]$. La expresión se simplifica como:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}^L} = \delta^L (\mathbf{a}^{L-1})^T. \quad (6.6)$$

Al extender este proceso desde la Ecuación 6.3 hasta la Ecuación 6.6 para calcular $\frac{\partial \mathcal{L}}{\partial \mathbf{b}^L}$, se obtiene:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}^L} = \delta. \quad (6.7)$$

Se reconoce que la función de pérdida \mathcal{L} en el conjunto de datos \mathcal{D} depende de las matrices de pesos y los vectores de sesgo de cada capa. Por lo tanto, se busca minimizar la función de pérdida en cada capa l . Al considerar las derivadas en la capa $L-1$, se obtiene:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}^{L-1}} &= \frac{\partial \mathcal{L}}{\partial \mathbf{a}^L} \frac{\partial \mathbf{a}^L}{\partial \mathbf{z}^L} \frac{\partial \mathbf{z}^L}{\partial \mathbf{a}^{L-1}} \frac{\partial \mathbf{a}^{L-1}}{\partial \mathbf{z}^{L-1}} \frac{\partial \mathbf{z}^{L-1}}{\partial \mathbf{w}^{L-1}} \\ &= [((\mathbf{w}^L)^T \delta^L) \odot f^{L-1(1)}(\mathbf{z}^{L-1})] (\mathbf{a}^{L-2})^T \\ &= \delta^{L-1} (\mathbf{a}^{L-2})^T, \end{aligned} \quad (6.8)$$

Donde $((\mathbf{w}^L)^T \delta^L) = \mathbf{e}^{L-1}$ y $[((\mathbf{w}^L)^T \delta^L) \odot f^{L-1}(\mathbf{z}^{L-1})] = \delta^{L-1}$.

De manera análoga, la derivada parcial de la función de pérdida con respecto al sesgo en la capa $l-1$ se expresa como:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}^{L-1}} = \delta^{L-1}. \quad (6.9)$$

Al generalizar este proceso, se obtiene la relación recurrente para δ^l :

$$\delta^l = [((\mathbf{w}^{l+1})^T \delta^{l+1}) \odot f^l(\mathbf{z}^l)], \quad \text{para } l = L-1, \dots, 1.$$

De esta forma, las derivadas parciales de la función de pérdida respecto a los pesos y sesgos en cada capa se expresan como:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}^l} = \delta^l (\mathbf{a}^{l-1})^T, \quad \frac{\partial \mathcal{L}}{\partial \mathbf{b}^l} = \delta^l. \quad (6.10)$$

Con estas expresiones, se define el proceso iterativo de retropropagación en los siguientes pasos:

1. Calcular el error \mathbf{e}^L y δ^L en la última capa, como se muestra en la Ecuación 6.5 y Ecuación 6.6.
2. Calcular \mathbf{e}^l y δ^l en cada capa l mediante las relaciones:

$$\begin{cases} \mathbf{e}^l &= (\mathbf{w}^{l+1})^T \delta^{l+1} \\ \delta^l &= (f^{l(1)}(\mathbf{z}^l)) \odot \mathbf{e}^l. \end{cases}$$

3. Proceder a la actualización de pesos y sesgos. Para ello, se introduce la función de coste \mathcal{C} aplicada a la pérdida \mathcal{L} y se calculan las derivadas con respecto a los pesos y sesgos:

$$\mathcal{C} = \frac{1}{n} \sum_{k=1}^n \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})(k)$$

Estas derivadas son equivalentes a las obtenidas en la Ecuación 6.6, Ecuación 6.7 y Ecuación 6.10. Luego, se aplica el algoritmo de descenso del gradiente para actualizar las matrices de pesos y sesgos:

$$\begin{aligned} \frac{\partial \mathcal{C}}{\partial \mathbf{w}^l} &= \frac{1}{n} \sum_{k=1}^n \frac{\partial}{\partial \mathbf{w}^l} \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})(k) \\ &= \nabla_{\mathbf{w}^l} \mathcal{C} \\ \frac{\partial \mathcal{C}}{\partial \mathbf{b}^l} &= \frac{1}{n} \sum_{k=1}^n \frac{\partial}{\partial \mathbf{b}^l} \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})(k) = \nabla_{\mathbf{b}^l} \mathcal{C}, \end{aligned} \quad (6.11)$$

Note que en la Ecuación 6.11, las derivadas con respecto a \mathbf{w}^L y \mathbf{b}^L , son las mismas que se dedujeron en la Ecuación 6.10, con estos últimos parámetros obtenidos, se puede aplicar el algoritmo del descenso del gradiente, para actualizar las matrices de pesos y sesgos, así:

$$\begin{cases} \mathbf{w}^l(t) &:= \mathbf{w}^{l-1}(t-1) - \eta \nabla_{\mathbf{w}^l} \mathcal{C}, \\ \mathbf{b}^l(t) &:= \mathbf{b}^{l-1}(t-1) - \eta \nabla_{\mathbf{b}^l} \mathcal{C}, \end{cases}$$

donde t representa el iterador de épocas asignadas a la red.

Al aplicar estos algoritmos (Feed-Forward y Backpropagation), se puede construir una red neuronal que, a partir de un conjunto de datos (ya sea de prueba o entrenamiento), se entrena para lograr una clasificación precisa de los datos. Es importante destacar que antes de aplicar el conjunto de prueba a la red, se debe realizar un proceso de preprocessamiento de datos, que incluye la eliminación de datos atípicos y la normalización del conjunto de entrenamiento, con el fin de evitar confusiones en la clasificación.

6.8. Evaluación de modelos de aprendizaje automático

Una vez entrenado un modelo de Machine Learning con datos etiquetados, se espera que funcione correctamente con nuevos datos. Sin embargo, es crucial asegurar la precisión de las predicciones del modelo en condiciones de producción.

Para lograr este objetivo, es imprescindible validar el modelo. Este procedimiento implica determinar si los resultados digitales que cuantifican las relaciones hipotéticas entre las variables son adecuados como descripciones de los datos.

Con el propósito de evaluar el rendimiento de un modelo de Machine Learning, es necesario ponerlo a prueba con datos nuevos. A partir del desempeño del modelo con datos desconocidos, se puede determinar si requiere ajustes adicionales, si ha sido sobreajustado o si está generalizado de manera adecuada.

Una de las técnicas más utilizadas para evaluar la eficacia de un modelo de Machine Learning es la validación cruzada. Este método, que también se considera un procedimiento de re-muestreo, permite evaluar un modelo incluso cuando se cuenta con datos limitados.

Para llevar a cabo la validación cruzada, es necesario reservar previamente una parte de los datos de entrenamiento. Estos datos no se emplearán durante el proceso de entrenamiento del modelo, sino que se utilizarán posteriormente para probarlo y validar sus resultados.

Frecuentemente en el ámbito del Machine Learning, la validación cruzada se emplea para comparar diferentes modelos y seleccionar aquel que sea más apropiado para un problema específico. Esta técnica, además de ser fácil de comprender e implementar, presenta menos sesgos que otros métodos. A continuación se exploran las principales técnicas de validación cruzada.

6.8.1. División de datos en entrenamiento y prueba

El enfoque denominado división datos en entrenamiento y prueba se basa en la aleatoria división de una serie de datos en dos conjuntos. Uno de estos conjuntos se destina al entrenamiento del modelo de Machine Learning, mientras que el otro se reserva para la validación del mismo.

Generalmente, se asigna entre un 70% y un 80% de los datos totales para el entrenamiento, dejando el 20 – 30% restante para llevar a cabo la validación cruzada.

Aunque esta técnica suele ser efectiva, su utilidad puede verse comprometida en casos de disponibilidad limitada de datos. En tales situaciones, existe la posibilidad de que se pierda información relevante durante el entrenamiento, lo que podría resultar en sesgos significativos en los resultados obtenidos.

No obstante, en escenarios donde la cantidad de datos es suficientemente amplia y la distribución entre los conjuntos es equilibrada, este enfoque se muestra completamente apropiado.

6.8.2. Validación cruzada de K pliegues

La técnica validación cruzada de K pliegues se caracteriza por su accesibilidad y su reconocimiento generalizado en el ámbito de la validación cruzada. En comparación con otros métodos de validación cruzada, tiende a proporcionar modelos con un menor sesgo (Zhang (1993)).

Esta técnica asegura que todas las observaciones originales de la serie de datos tengan la oportunidad de formar parte tanto del conjunto de entrenamiento como del conjunto de prueba. Es especialmente valiosa en situaciones donde los datos de entrada son limitados.

El proceso comienza al dividir de manera aleatoria la serie de datos en K pliegues. El parámetro K determina el número de grupos en los que se dividirá la muestra.

Es fundamental elegir un valor adecuado para K , evitando que sea demasiado bajo o demasiado alto. Usualmente, se selecciona un valor entre 5 y 10 dependiendo del tamaño de la serie de datos. Por ejemplo, si $K = 10$, la serie de datos se divide en 10 partes iguales.

Un valor de K más alto reduce el sesgo del modelo, pero una varianza excesiva puede conducir al sobreajuste. Un valor más bajo equivale prácticamente al enfoque de división de datos en entrenamiento y prueba.

Parte IV.

Estudio de caso

Modelación y pronóstico del número de casos confirmados y fallecidos por COVID-19 en IRÁN

A finales de diciembre de 2019, se identificó la aparición de un nuevo virus en Wuhan, China, el cual manifestó un impacto agudo en el sistema respiratorio y presentó una rápida propagación. La Organización Mundial de la Salud (OMS) catalogó este virus como el SARS-CoV-2, perteneciente a la familia de los coronavirus. Aunque algunas investigaciones y evidencias sugieren que los murciélagos podrían ser el origen principal del COVID-19, esta afirmación aún no está definitivamente confirmada y requiere una mayor investigación. Esta enfermedad infecciosa aguda se caracteriza por su alta tasa de contagio, lo que llevó a declararla una pandemia global debido a su rápida expansión y diseminación a nivel mundial.

Los síntomas comunes de esta enfermedad incluyen complicaciones respiratorias, tos seca, fiebre, escalofríos, dificultad para respirar, dolor torácico, neumonía, entre otros. No obstante, a medida que progresó la enfermedad, los síntomas en los pacientes evolucionaron y varían.

Una de las principales problemáticas asociadas a este virus es su periodo de incubación de hasta 14 días, durante el cual puede transmitirse la infección sin presentar síntomas. Además, algunas personas infectadas con el COVID-19 manifiestan síntomas leves, similares a un resfriado común o a la gripe. Esta pandemia ha ejercido una presión significativa sobre los gobiernos y los sistemas de salud pública. La escasez de equipamiento médico en hospitales, incluyendo camas, unidades de cuidados intensivos, personal médico, ventiladores, entre otros, constituye uno de los principales desafíos. Asimismo, han surgido repercusiones económicas y sociales a raíz de la propagación de la enfermedad y la implementación de cuarentenas estrictas, lo que ha afectado la salud mental de las comunidades, entre otros aspectos.



El surgimiento de las problemáticas mencionadas, sumado a la ausencia de tratamientos definitivos para esta enfermedad, la naturaleza dinámica del virus y su propagación global, subraya la necesidad de investigar exhaustivamente este virus y su comportamiento. Se han explorado diversos campos y metodologías de pronóstico y modelización. Uno de estos enfoques de pronóstico radica en la creación de modelos para anticipar el número de casos futuros, basados en registros de casos confirmados. Aunque las proyecciones sobre el número de pacientes futuros no son totalmente precisas, sirven de apoyo a los gobiernos y a los responsables de políticas de salud para adoptar decisiones cruciales y aplicar restricciones que reduzcan la prevalencia. Asimismo, resulta crucial anticipar futuros brotes, posibles mutaciones del virus y su propagación, especialmente identificar el pico para mitigar sus efectos graves. Estas proyecciones asisten a los tomadores de decisiones para prevenir e incluso controlar la propagación de la enfermedad mediante políticas efectivas y rigurosas. Cabe destacar que la falta de información suficiente con anticipación constituye uno de los desafíos principales en el pronóstico, aunque sigue siendo una herramienta de orientación efectiva para los gobiernos en la contención de la enfermedad.

Por consiguiente, dado el papel potencialmente efectivo de los modelos estadísticos y matemáticos en la predicción de la tendencia futura de la enfermedad, en este estudio se emplearon dos modelos, Holt-Winter y MLP (Multilayer Perceptron), con el propósito de determinar el mejor modelo para pronosticar, de manera independiente, el número de casos confirmados y muertes en Irán para los próximos 30 días.

En el presente estudio, se utilizó el conjunto de datos disponible en el sitio web <https://www.who.int/>, el cual contempla el número absoluto de casos confirmados y muertes por día, excluyendo otros factores debido a su falta de disponibilidad.

El análisis a realizar tiene como propósito verificar los resultados obtenidos en el estudio de Talkhi et al. (2021) .

7. Pronóstico de infectados diarios

7.1. Obtención de datos

Conforme se ha referido previamente, se emplea el conjunto de datos global informado diariamente, disponible para su descarga en <https://covid19.who.int/WHO-COVID-19-global-data.csv>. Resulta relevante destacar que la base de datos consultada corresponde al 16 de Enero del 2023, restringiéndose a los datos concernientes exclusivamente a los casos confirmados en Irán entre el 20 de febrero y el 15 de agosto de 2020.

El análisis posterior se ha llevado a cabo empleando el software R versión 4.3.2. Con el propósito de realizar el análisis de los datos y la generación de gráficos, se procedió a convertir los datos al formato *ts*, lo que permitió su representación como una serie temporal.

```
# Se crea un objeto 'Date' diario
inds <- seq(as.Date("2020-02-20"), as.Date("2020-08-15"), by = "day")
# Se crea un objeto 'serie de tiempo' de frecuencia diaria
Confirmed_ts <- ts(Confirmed_df[2],
                     start = c(2020, as.numeric(format(inds[1], "%j"))),
                     frequency = 365)
```

La gráfica de la Figura 7.1 exhibe la serie temporal derivada de la base de datos, en la cual se evidencia la ausencia de información para los días 27 y 29 de Febrero, así como para el 02 de Marzo y el 05 de Abril de 2020. Para subsanar esta carencia de datos, se llevó a cabo una interpolación promedio a fin de sustituir los valores faltantes. La Figura 7.2 muestra la serie de tiempo resultante de estas correcciones.

7.2. Análisis de la serie de tiempo de casos confirmados de COVID-19 en Irán

7.2.1. Estadística descriptiva

Con el propósito de llevar a cabo una auditoría de los datos y al mismo tiempo una descripción preliminar, se ejecuta un estudio de estadística descriptiva que arroja los

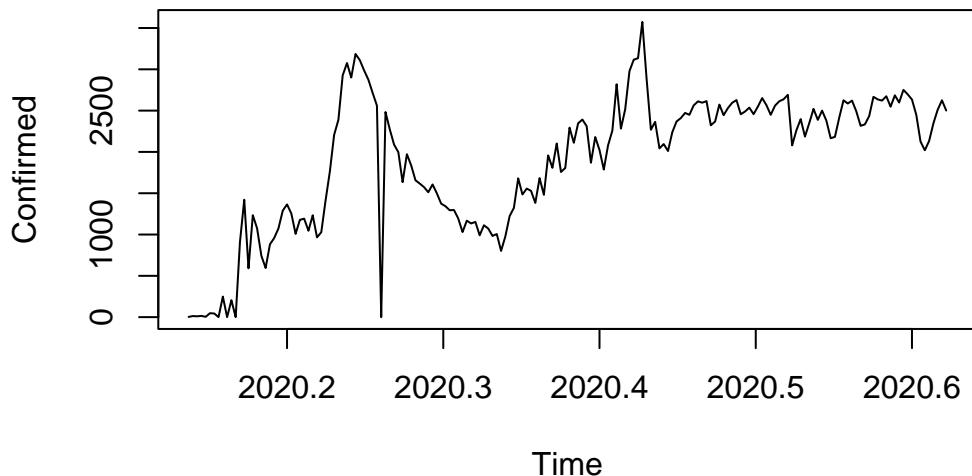


Figura 7.1.: Serie de tiempo de los casos de COVID-19 confirmados en Irán del 20-02-2020 al 15-08-2020

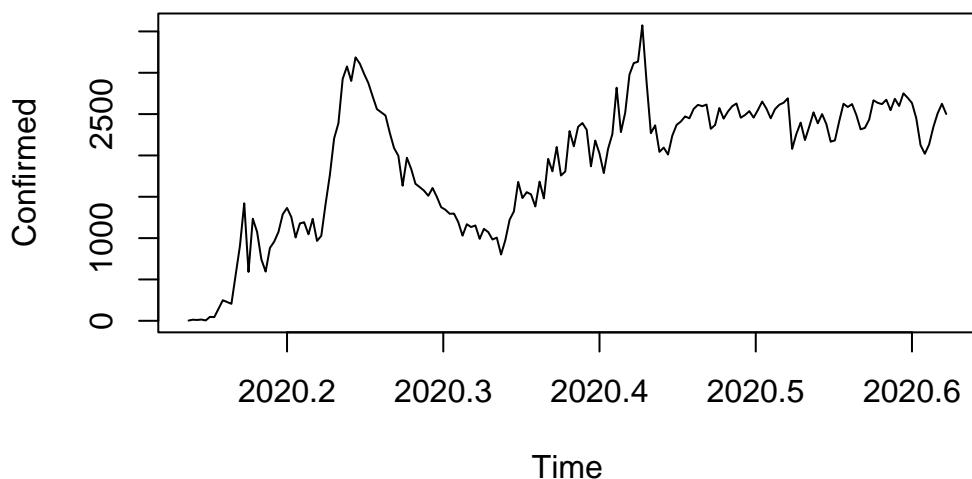


Figura 7.2.: Serie de tiempo de los casos de COVID-19 confirmados en Irán del 20-02-2020 al 15-08-2020

resultados correspondientes, incluyendo un gráfico Boxplot (Figura 7.3) para representar la información o

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
3	1304	2181	1923	2529	3574

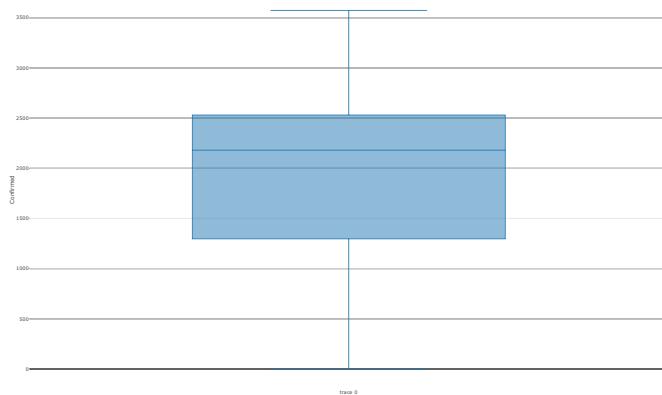


Figura 7.3.: Boxplot de casos confirmados de COVID-19 en Irán del 20-02-2020 al 15-08-2020

7.2.2. Componentes de la serie de tiempo

Los componentes identificados en la serie de tiempo de casos confirmados de COVID-19 en Irán, revelan distintos patrones y características.

En primer lugar, se observa una tendencia discernible en el gráfico de la serie temporal (@fig-ts). Por ejemplo, entre el 30 de marzo y el 03 de mayo de 2020, se evidencia una tendencia negativa o decreciente, seguida por una tendencia creciente a partir del 03 de mayo en adelante. Estos cambios en la tendencia podrían indicar fluctuaciones significativas en la evolución de los casos confirmados durante esos períodos específicos.

En cuanto a la estacionalidad, aunque no se identifica claramente a simple vista en el periodo observado, la extensión del análisis a un periodo más amplio podría revelar

patrones recurrentes o ciclos temporales característicos. Es posible que ciertos patrones estacionales se manifiesten en intervalos más extensos de la serie temporal, lo que implicaría variaciones sistemáticas y repetitivas en los datos en períodos específicos.

Por último, se destacan pequeñas subidas y bajadas en el gráfico que sugieren la presencia de **ruido** en la serie temporal. Estas fluctuaciones irregulares podrían atribuirse a diversas causas, como posibles errores en la recolección de datos o fluctuaciones aleatorias inherentes al comportamiento de la enfermedad. Es importante considerar estas variaciones no sistemáticas al analizar la serie temporal, ya que podrían influir en la interpretación de los patrones y tendencias observadas.

7.2.3. Estacionariedad

A continuación, se emplea el test de Dickey-Fuller para examinar la presencia de estacionariedad en la serie temporal. Este test fue utilizado con la finalidad de identificar la existencia de raíces unitarias en la serie, lo cual permite inferir la presencia o ausencia de estacionariedad en los datos analizados.

```
adf.test(Confirmed_ts, alternative = "stationary")
```

```
Augmented Dickey-Fuller Test

data: Confirmed_ts
Dickey-Fuller = -2.9529, Lag order = 5, p-value = 0.1781
alternative hypothesis: stationary
```

La hipótesis nula (H_0) asume la presencia de raíces unitarias, lo que indica no estacionariedad en la serie. Al obtener un p -valor superior al nivel de significancia establecido el cuál es del 95%, no se rechaza la hipótesis nula, sugiriendo la ausencia de estacionariedad en la serie de tiempo de casos confirmados.

Además, se complementa la evaluación de la estacionalidad mediante la inspección de los gráficos de la función de autocorrelación (ACF) y la función de autocorrelación parcial (PACF). Estos gráficos se utilizan para identificar patrones de autocorrelación en la serie temporal, lo que permite visualizar la presencia de estacionalidad, tendencias o ciclos.

La serie de tiempo representada en la `?@fig-ts` exhibe un comportamiento característico de deambulación aleatoria. Dado que el valor de la variable X_{t+1} generalmente se encuentra en proximidad al valor X_t , se evidencia una autocorrelación positiva notablemente marcada entre las variables X_t y X_{t+1} .

En la Figura 7.4 se observa que la autocorrelación (vea Ecuación 5.2) entre X_t y X_{t+k} decrece con el incremento del retraso k . Este declive conduce a la constatación de que,

```
autoplot(acf(Confirmed_ts, plot = FALSE),  
        main="Autocorrelograma de casos confirmados")
```

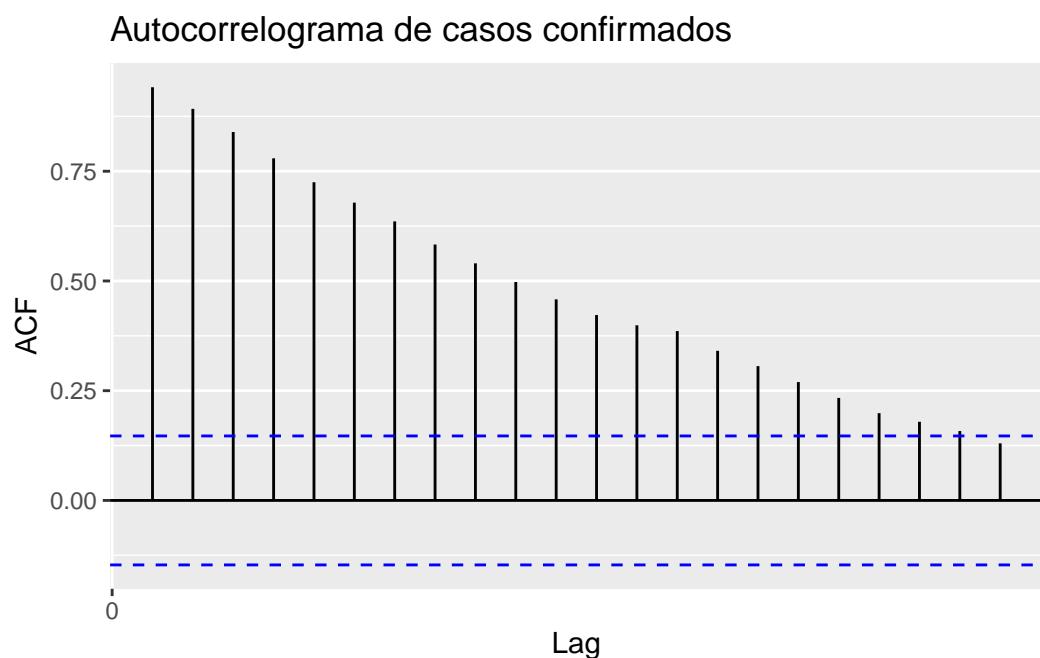


Figura 7.4.: Autocorrelograma de los casos confirmados de COVID-19 en Irán

a un desfase de 20, existe una correlación bastante débil entre X_t y X_{t+20} . Al analizar el gráfico de la función de autocorrelación (ACF), se aprecia que $\rho_{20} \approx 0.19$.

La gráfica de la Función de Autocorrelación Parcial (PACF) proporciona información valiosa sobre la estructura de autocorrelación de una serie temporal una vez han sido eliminadas las correlaciones debidas a los intervalos de tiempo intermedios.

```
ggPacf((Confirmed_ts), main = 'Autocorrelograma parcial de casos confirmados')
```

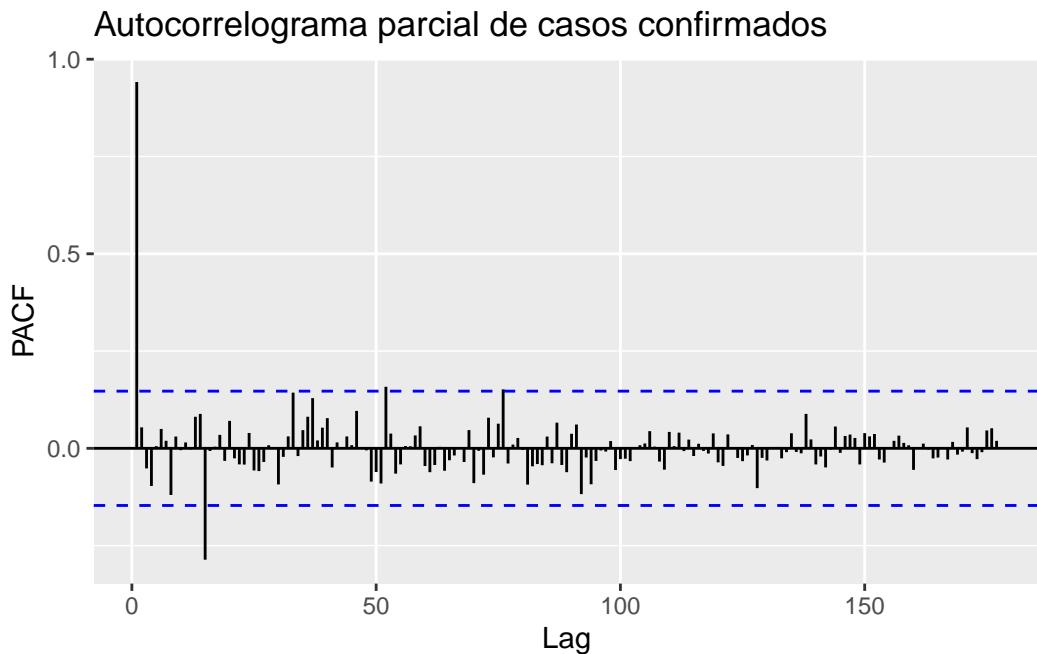


Figura 7.5.: Autocorrelograma Parcial de los casos confirmados de COVID-19 en Irán

Considerando que los datos se ajustan a un modelo de series de tiempo, la Figura 7.5 indica que el valor de correlación ϕ_{15} es ligeramente superior a 0.25, aproximadamente $\phi_{52} \approx 0.16$, y $\phi_{76} \approx 0.15$, mientras que para los restantes valores, la correlación parcial no es nula.

Observación. De acuerdo con la gráfica de la Función de Autocorrelación Parcial Figura 7.5, se observa un corte abrupto después del rezago 4, lo cual sugiere que las autocorrelaciones parciales más allá de ese punto no poseen significancia estadística. Por consiguiente, se infiere la posibilidad de ajustar un modelo autoregresivo AR(4) a la base de datos.

```
library(tswge)
coeff <- est.ar.wge(Confirmed_ts, p=4)
```

```
Coefficients of AR polynomial:
0.8656 0.1724 0.0592 -0.1318
```

AR Factor Table				
Factor	Roots	Abs	Recip	System Freq
1-0.9605B	1.0411	0.9605	0.0000	
1-0.5355B	1.8675	0.5355	0.0000	
1+0.6303B+0.2563B^2	-1.2298+-1.5459i	0.5062	0.3570	

```
coeff$phi #coeficientes
```

```
[1] 0.86563811 0.17237111 0.05917558 -0.13180534
```

```
coeff$xbar #media
```

```
[1] 1922.868
```

```
coeff$avar #varianza finita
```

```
[1] 47818.04
```

El modelo autoregresivo AR(4) se expresa mediante la siguiente ecuación:

$$(1 - 0.865B - 0.172B^2 - 0.059B^3 + 0.131B^4)(X_t - 1922.868) + a_t \quad (7.1)$$

donde $\hat{\sigma}_a^2 = 47818.04$.

El análisis del ACF y PACF proporcionó información sobre la relación de los puntos de datos con sus rezagos, permitiendo observar posibles patrones estacionales. La presencia de picos significativos en estos gráficos podría indicar la existencia de estacionalidad en la serie de tiempo.

7.3. Entrenamiento, modelado, pronóstico y métricas de rendimiento

Se procede a la evaluación del rendimiento de métodos destinados al ajuste y consecuente pronóstico. Específicamente, se contempla el método de suavizamiento exponencial de Holt-Winters y el ajuste mediante un modelo de red neuronal del tipo perceptrón multicapa. Ambos procedimientos requieren la subdivisión de los datos en conjuntos destinados a entrenamiento y prueba. El set inicial, compuesto por el 70% de los datos, se emplea para el entrenamiento de los modelos, mientras que el 30% restante se reservará para llevar a cabo las pruebas pertinentes.

```
Confirmed_ts <- ts(Confirmed_ts,frequency=1)
tsize <- round(0.7 * nrow(Confirmed_df))
train_confirmed <- window(Confirmed_ts,end=tsize)
test_confirmed <- window(Confirmed_ts,start=tsize+1)
```

7.3.1. Holt-Winters

Con el fin de determinar la descomposición más adecuada para los datos en cuestión, se empleó un criterio elaborado basado en el coeficiente de variación, el cual proporciona una recomendación entre las dos versiones disponibles.

```
DescRec <- function(x){
  n = length(x)
  di = rep(0, n-1)
  ci = rep(0, n-1)
  for (i in 1:n-1) {
    di[i] = x[i+1] - x[i]
    ci[i] = x[i+1] / x[i]
  }
  d <- cv(di)
  c <- cv(ci) / mean(di)
  if(d < c)
    print("Se recomienda la descomposición aditiva")
  else
    print("Se recomienda la descomposición multiplicativa")
}
DescRec(train_confirmed)
```

```
[1] "Se recomienda la descomposición multiplicativa"
```

De acuerdo con la recomendación observada, se sugiere la utilización de la versión multiplicativa (vea Ecuación 5.19). En consecuencia, se procede a mostrar la representación gráfica de la descomposición multiplicativa de la serie temporal.

```
ts_train <- ts(train_confirmed, frequency = 2)
components_ts <- decompose(ts_train, type = 'mult')
plot(components_ts)
```

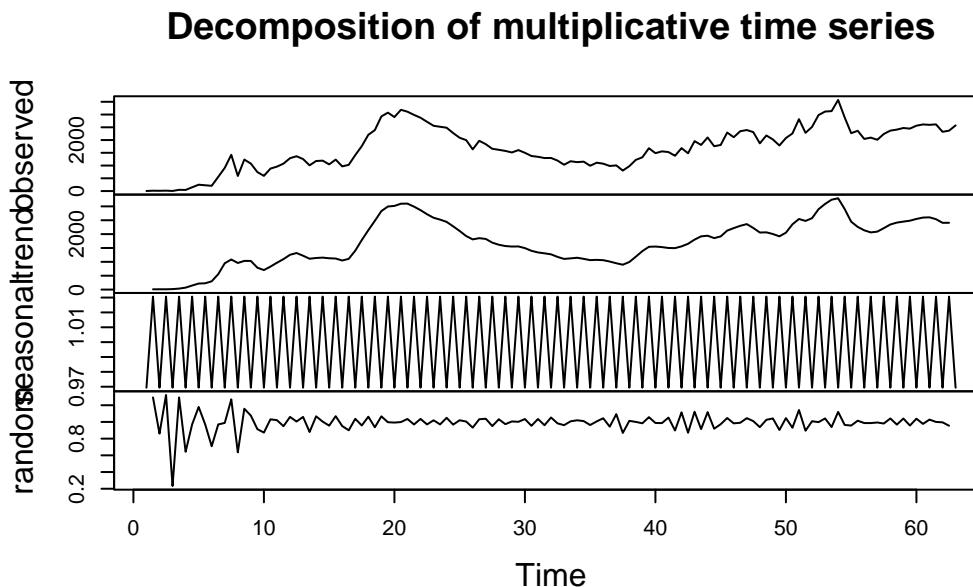


Figura 7.6.: Descomposición multiplicativa de la serie de tiempo

Se procede ahora a la aplicación del modelo multiplicativo de Holt-Winters a la serie temporal de los datos de entrenamiento utilizando una frecuencia de dos, con el fin de permitir la aplicabilidad del modelo.

```
HWc <- HoltWinters(ts_train, seasonal = 'mult')
HWc
```

Holt-Winters exponential smoothing with trend and multiplicative seasonal component.

Call:
HoltWinters(x = ts_train, seasonal = "mult")

Smoothing parameters:
alpha: 0.729859

```
beta : 0  
gamma: 0.6494094
```

Coefficients:

```
[,1]  
a 2289.283568  
b 2.250000  
s1 1.119185  
s2 1.113547
```

Finalmente, utilizando el modelo de entrenamiento desarrollado en la fase previa, se lleva a cabo la proyección con un horizonte de predicción igual en extensión a los datos de prueba, acompañado de un intervalo de confianza que oscila entre el 80% y el 95%.

```
HWc_for <- forecast(HWc, h=length(test_confirmed))
```

i Nota

Las funciones aplicadas en esta sección son parte de la librería **stats** (2023) de R.

7.3.2. MLP

Posteriormente, se procede al entrenamiento del modelo MLP (Perceptrón Multicapa). La cantidad de capas ocultas y la configuración de nodos en cada capa se determinaron de manera automatizada mediante el método de [validación cruzada de 5 pliegues](#). Asimismo, se eligió la función de activación como sigmoide, y el proceso de entrenamiento del modelo se ejecutó a lo largo de 20 iteraciones.

```
fitc <- mlp(train_confirmed, hd.auto.type="cv", reps=20, comb='median')  
fitc
```

```
MLP fit with 1 hidden node and 20 repetitions.  
Univariate lags: (1,2,4)  
Forecast combined using the median operator.  
MSE: 57287.47.
```

Para llevar a cabo el pronóstico, se emplea el modelo de entrenamiento creado en la etapa anterior, manteniendo un horizonte de predicción que coincide en duración con los datos de prueba, tal como se hizo con la técnica anterior.

```
plot(fitc)
```

MLP

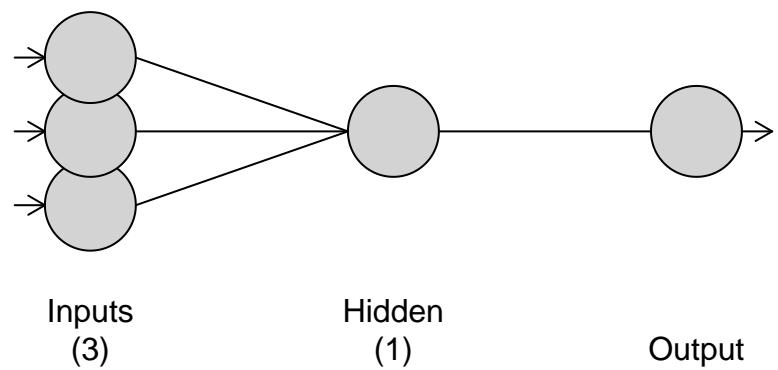


Figura 7.7.: Estructura de la red neuronal resultante

```
frcc <- forecast(fitc,h=length(test_confirmed))
```

i Nota

Las funciones aplicadas en esta sección son parte de la librería **nnfor** Kourentzes (2022) de R.

7.3.3. Comparación de pronósticos con el conjunto de datos de prueba

Con el propósito de llevar a cabo un análisis cuantitativo exhaustivo, se presenta a continuación una tabla comparativa de los resultados derivados de las dos técnicas implementadas y la base de datos de prueba. Posteriormente, se exhiben gráficas representativas de estos resultados. En la Figura 7.8 se muestra el pronóstico mediante Holt-Winters acompañado de su respectivo intervalo de confianza. En contraste, en la Figura 7.9, la gráfica punteada en color rojo representa el comportamiento real de los datos, mientras que en azul se representa el pronóstico obtenido a través de la red MLP.

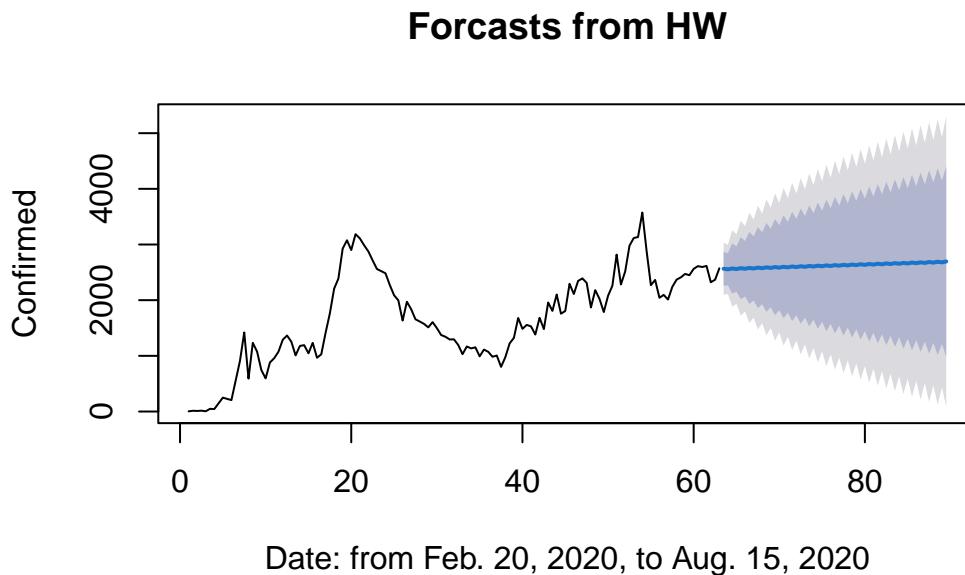


Figura 7.8.: Pronóstico obtenido mediante la técnica de Holt-Winters.

Tabla 7.1.: Comparación de Resultados entre las técnicas y los datos reales para evaluar precisión

	Date	Confirmed	Forecast_HW	Forecast_MLP
t+1	2020-06-24	2445	2564.651	2505.038
t+2	2020-06-25	2531	2554.235	2546.790
t+3	2020-06-26	2595	2569.687	2553.964
t+4	2020-06-27	2628	2559.246	2541.389
t+5	2020-06-28	2456	2574.724	2543.295
t+6	2020-06-29	2489	2564.257	2535.901
t+7	2020-06-30	2536	2579.760	2530.003
t+8	2020-07-01	2457	2569.268	2525.599
t+9	2020-07-02	2549	2584.796	2520.655
t+10	2020-07-03	2652	2574.279	2516.960
t+11	2020-07-04	2566	2589.833	2513.834
t+12	2020-07-05	2449	2579.290	2511.227
t+13	2020-07-06	2560	2594.869	2509.216
t+14	2020-07-07	2613	2584.301	2507.597
t+15	2020-07-08	2637	2599.905	2506.337
t+16	2020-07-09	2691	2589.311	2505.366
t+17	2020-07-10	2079	2604.942	2504.615
t+18	2020-07-11	2262	2594.322	2504.044
t+19	2020-07-12	2397	2609.978	2503.610
t+20	2020-07-13	2186	2599.333	2503.281
t+21	2020-07-14	2349	2615.014	2503.035
t+22	2020-07-15	2521	2604.344	2502.851
t+23	2020-07-16	2388	2620.051	2502.714
t+24	2020-07-17	2500	2609.355	2502.612
t+25	2020-07-18	2379	2625.087	2502.537
t+26	2020-07-19	2166	2614.366	2502.482
t+27	2020-07-20	2182	2630.123	2502.441
t+28	2020-07-21	2414	2619.377	2502.411
t+29	2020-07-22	2625	2635.160	2502.390
t+30	2020-07-23	2586	2624.388	2502.374
t+31	2020-07-24	2621	2640.196	2502.363
t+32	2020-07-25	2489	2629.399	2502.355
t+33	2020-07-26	2316	2645.232	2502.349
t+34	2020-07-27	2333	2634.410	2502.345
t+35	2020-07-28	2434	2650.269	2502.342
t+36	2020-07-29	2667	2639.421	2502.340
t+37	2020-07-30	2636	2655.305	2502.339
t+38	2020-07-31	2621	2644.432	2502.338
t+39	2020-08-01	2674	2660.341	2502.337
t+40	2020-08-02	2548	2649.443	2502.337
t+41	2020-08-03	2685	2665.378	2502.336
t+42	2020-08-04	2598	2654.454	2502.336
t+43	2020-08-05	2751	2670.4145	2502.336
t+44	2020-08-06	2697	2659.465	2502.336
t+45	2020-08-07	2634	2675.450	2502.336
t+46	2020-08-08	2450	2664.476	2502.336
t+47	2020-08-09	2125	2680.487	2502.336
t+48	2020-08-10	2020	2669.487	2502.336
t+49	2020-08-11	2132	2685.523	2502.336
t+50	2020-08-12	2345	2674.498	2502.336
t+51	2020-08-13	2510	2690.559	2502.336
t+52	2020-08-14	2225	2670.502	2502.336

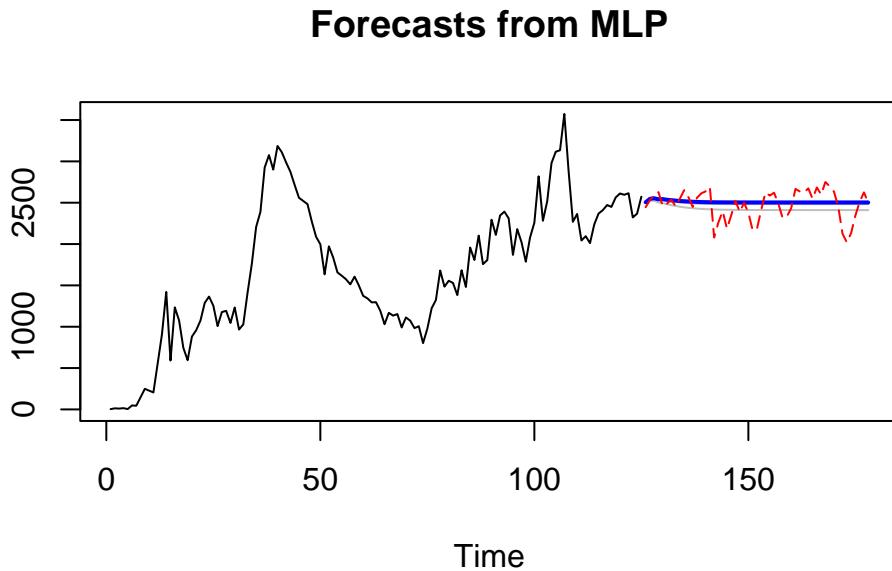


Figura 7.9.: Pronóstico obtenido mediante la red neuronal MLP.

7.3.3.1. Métricas de rendimiento

Para evaluar la calidad o bondad de ajuste de los métodos utilizados en este estudio y seleccionar el modelo más apropiado, se aplican tres métricas de rendimiento, Error Cuadrático Medio (Ecuación 5.31), Error Absoluto Medio (Ecuación 5.29) y Error Porcentual Absoluto Medio (Ecuación 5.32) tanto en las fases de entrenamiento como en las de prueba. Los resultados correspondientes a éstas métricas se presentan en la Tabla 7.2 .

Tabla 7.2.: Errores de los modelos para casos confirmados.

	<i>Training</i>			<i>Testing</i>		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Holt-Winters	262.9925	190.0482	20.8415	234.0094	165.8208	6.2967
MLP	239.3483	180.9937	14.6079	177.0605	136.4799	5.4441

7.3.4. Conclusión

Basándose en los resultados extraídos tanto de la tabla de pronósticos (Tabla 7.1) como de la tabla de errores (Tabla 7.2), se llega a la conclusión de que, para esta base de

datos en particular, la técnica de redes neuronales MLP demuestra ser más efectiva en la predicción realizada. Esto se fundamenta en la evidencia de un menor error registrado en las tres métricas calculadas, tanto durante la fase de entrenamiento como en la fase de prueba.

7.4. Pronóstico de los próximos 30 días

Tras la identificación del modelo óptimo, se procedió a prever el comportamiento futuro de la serie temporal de casos confirmados para los próximos 30 días utilizando dicho modelo. Se elaboraron representaciones gráficas de la predicción de casos confirmados de COVID-19 a 30 días, realizando una comparación de la efectividad entre las implementaciones de redes neuronales en la paquetería de R y la paquetería nativa de Python, las cuales se encuentran en las figuras Figura 7.10 y Figura 7.11, respectivamente.

7.4.1. Implementación en R

En la implementación de R, siguiendo el mismo procedimiento que en las fases de entrenamiento y prueba, se empleó un número específico de capas y nodos ocultos determinados automáticamente a través del método de validación cruzada de 5 pliegues. Esta configuración se llevó a cabo con una función de activación sigmoide, ejecutando 20 iteraciones para el entrenamiento de la red neuronal.

```
fit.mlp = mlp(ts(Confirmed_df$Confirmed), reps = 20, hd.auto.type = 'cv',
               comb="median")
fore.mlp = forecast(fit.mlp, h = 30)
```

Los resultados del pronóstico indican que el 14 de septiembre de 2020 se proyectan aproximadamente 2494 nuevos casos confirmados de COVID-19. Estos valores correspondientes al período de 30 días se detallan a continuación en la Tabla 7.3 .

7.4.2. Implementación en Python

En esta sección, se realizaron ajustes en el método para su implementación. Cada día proyectado se forma utilizando el dato del día anterior. En cada paso, se actualiza la secuencia de entrada eliminando el valor más antiguo e incorporando la predicción más reciente como el dato más reciente. Esta dinámica se representa esquemáticamente a continuación, donde n representa la extensión de la secuencia de entrada y T es la longitud de la serie temporal.

Tabla 7.3.: Pronóstico de casos confirmados de COVID-19 en Irán en los próximos 30 días

	Fecha	Infectados
t+1	2020-08-16	2538.350
t+2	2020-08-17	2522.142
t+3	2020-08-18	2496.552
t+4	2020-08-19	2493.339
t+5	2020-08-20	2484.594
t+6	2020-08-21	2477.845
t+7	2020-08-22	2475.811
t+8	2020-08-23	2473.210
t+9	2020-08-24	2472.426
t+10	2020-08-25	2472.609
t+11	2020-08-26	2474.396
t+12	2020-08-27	2475.853
t+13	2020-08-28	2477.092
t+14	2020-08-29	2478.059
t+15	2020-08-30	2478.764
t+16	2020-08-31	2479.198
t+17	2020-09-01	2479.416
t+18	2020-09-02	2479.474
t+19	2020-09-03	2479.425
t+20	2020-09-04	2479.317
t+21	2020-09-05	2479.186
t+22	2020-09-06	2478.541
t+23	2020-09-07	2476.515
t+24	2020-09-08	2477.146
t+25	2020-09-09	2478.806
t+26	2020-09-10	2478.772
t+27	2020-09-11	2478.756
t+28	2020-09-12	2478.755
t+29	2020-09-13	2478.762
t+30	2020-09-14	2478.774

Forecasts from MLP

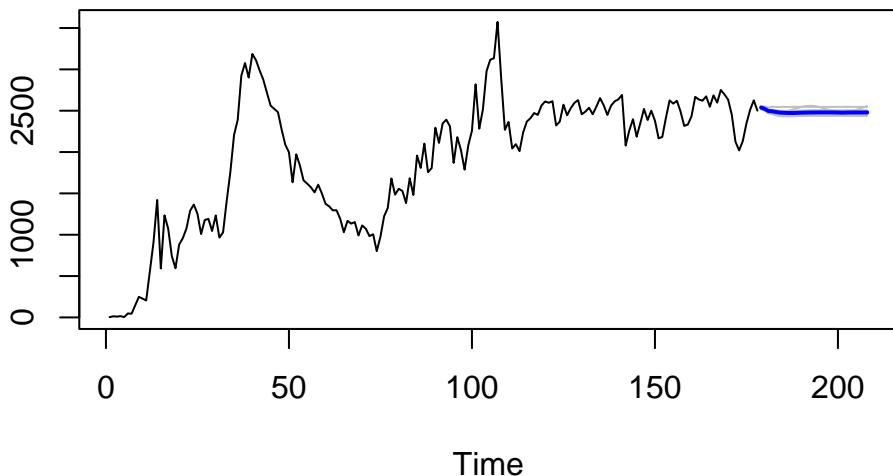


Figura 7.10.: Predicción futura de la serie tiempo para infectados diariamente mediante el modelo MLP

$$\begin{array}{lll}
 y : \text{Observado} & \hat{y} : \text{Pronosticado} \\
 \begin{matrix} y_{T-n+1} & y_{T-n+2} & y_{T-n+3} & \cdots & y_{T-2} & y_{T-1} & y_T & \rightarrow & \hat{y}_{T+1} \\ y_{T-n+2} & y_{T-n+3} & y_{T-n+4} & \cdots & y_{T-1} & y_T & \hat{y}_{T+1} & \rightarrow & \hat{y}_{T+2} \\ y_{T-n+3} & y_{T-n+4} & y_{T-n+5} & \cdots & y_T & \hat{y}_{T+1} & \hat{y}_{T+2} & \rightarrow & \hat{y}_{T+3} \\ \ddots & & & & & & & & \end{matrix}
 \end{array}$$

Se exhibe a continuación el código utilizado y el gráfico correspondiente al pronóstico generado por la red neuronal.

```

import numpy as np
import pandas as pd
import yfinance as yf
import tensorflow as tf
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
from tensorflow.keras.layers import Dense, LSTM
from tensorflow.keras.models import Sequential

```

```

from sklearn.preprocessing import MinMaxScaler

pd.options.mode.chained_assignment = None
tf.random.set_seed(0)
df = pd.read_excel('Data.xlsx')

# ----- Entrenamiento y prueba del modelo -----
y = df['Confirmed'].fillna(method='ffill')
y = y.values.reshape(-1, 1)
# scale the data
scaler = MinMaxScaler(feature_range=(0, 1))
scaler = scaler.fit(y)
y = scaler.transform(y)

# generate the input and output sequences
n_lookback = 53 # length of input sequences (lookback period)
n_forecast = 30 # length of output sequences (forecast period)

X = []
Y = []

for i in range(n_lookback, len(y) - n_forecast + 1):
    X.append(y[i - n_lookback: i])
    Y.append(y[i: i + n_forecast])

X = np.array(X)
Y = np.array(Y)

# fit the model
model = Sequential()
model.add(Dense(20, activation='sigmoid', input_dim=n_lookback))
model.add(Dense(n_forecast))

model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(X, Y, epochs=20, batch_size=4, verbose=0)

# generate the forecasts
X_ = y[- n_lookback:] # last available input sequence
X_ = X_.reshape(1, n_lookback, 1)

Y_ = model.predict(X_).reshape(-1, 1)
Y_ = scaler.inverse_transform(Y_)

```

```

# organize the results in a data frame
df_past = df
df_past.rename(columns={'Date': 'Date', 'Confirmed': 'Actual'}, inplace=True)
df_past['Date'] = pd.to_datetime(df_past['Date'])
df_past['Forecast'] = np.nan
df_past['Forecast'].iloc[-1] = df_past['Actual'].iloc[-1]

df_future = pd.DataFrame(columns=['Date', 'Actual', 'Forecast'])
df_future['Date'] = pd.date_range(
    start=df_past['Date'].iloc[-1] + pd.Timedelta(days=1),
    periods=n_forecast)
df_future['Forecast'] = Y_.flatten()
df_future['Actual'] = np.nan

results = df_past.append(df_future).set_index('Date')
# Calculate minimum, median, and maximum for each forecasted date
results['Min'] = results['Forecast'].rolling(window=2).min()
results['Max'] = results['Forecast'].rolling(window=2).max()
results['Median'] = results['Forecast'].rolling(window=2).median()

# Creamos la gráfica con las predicciones
#fig = px.line(results, x=results.index, y=['Actual', 'Forecast', 'Median'],
fig = px.line(results, x=results.index, y=['Actual', 'Median'],
              labels={'index': 'Date', 'value': 'Confirmed Cases'},
              title='Casos Confirmados',
              line_shape='linear')

fig.update_traces(line=dict(color='cornflowerblue'),
                   selector=dict(name='Actual'))
fig.update_traces(line=dict(color='orange'),
                   selector=dict(name='Forecast'))
fig.update_traces(line=dict(color='mediumvioletred'),
                   selector=dict(name='Median'))

# Agregar gráficos de área para el mínimo y el máximo
fig.add_trace(
    go.Scatter(x=results.index,
               y=results['Min'],
               fill=None, mode='lines',
               line=dict(color='hotpink'),
               name='Min'))
fig.add_trace(
    go.Scatter(x=results.index,

```

```

y=results['Max'],
fill='tonexty',
mode='lines',
line=dict(color='deeppink'),
name='Max'))
fig.show('')

```

Casos Confirmados

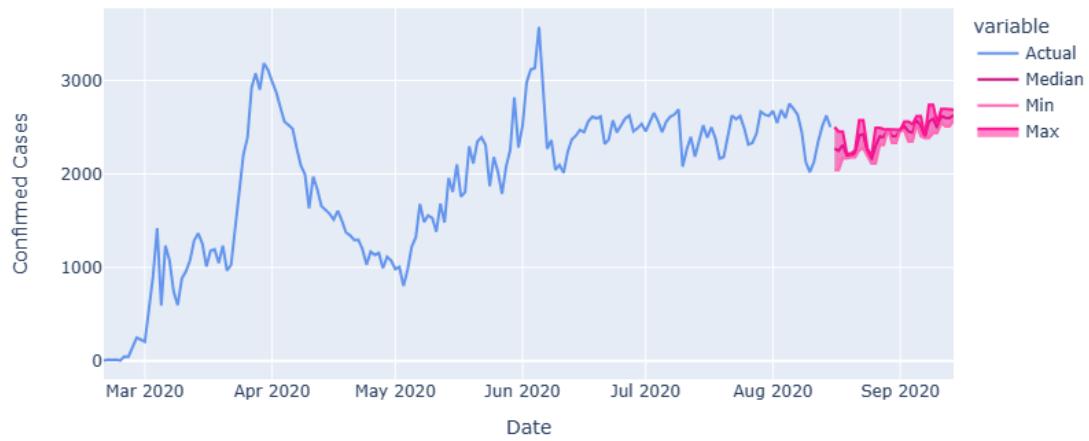


Figura 7.11.: Pronostico de casos confirmados de COVID-19 en Irán (implementación en Python)

Los resultados del pronóstico en Figura 7.11 indican que el 14 de Septiembre de 2020 se proyectan aproximadamente 2476 nuevos casos confirmados de COVID-19.

8. Pronóstico de decesos diarios

De manera similar a la evaluación llevada a cabo en la base de datos de los casos confirmados de COVID-19 en Irán, se ejecutó un análisis para los datos de muertes por COVID-19 en Irán, abarcando el mismo período temporal, desde el 20 de febrero hasta el 15 de agosto de 2020. El propósito fue determinar si, de forma análoga, la red neuronal MLP demuestra un mejor ajuste a los datos en comparación con la técnica de series temporales Holt-Winters, o si, en este caso particular, la técnica Holt-Winters ofrece una mayor precisión.

Es relevante señalar la presencia de datos faltantes para las fechas 21 de febrero, 1 de marzo y 5 de mayo de 2020 en el conjunto de datos. Por consiguiente, al igual que en los casos confirmados, se procedió con una interpolación promedio para sustituir dichos datos faltantes. La Figura 8.1 exhibe la serie temporal resultante luego de estas correcciones.

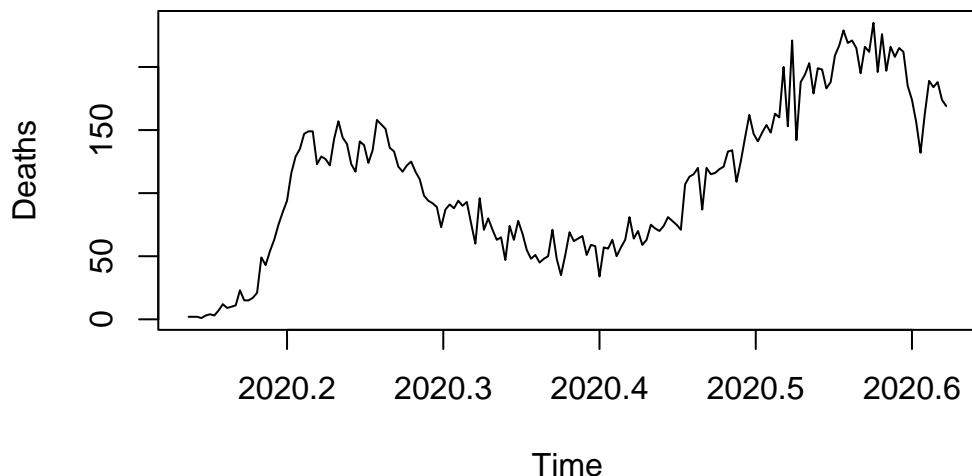


Figura 8.1.: Serie de tiempo de los casos de muerte por COVID-19 en Irán del 20-02-2020 al 15-08-2020

8.1. Pronóstico, comparación y métricas de rendimiento

La evaluación del desempeño de los métodos se realiza a través de la separación del conjunto de datos en entrenamiento y prueba. El set inicial, compuesto por el 70% de los datos, se emplea para el entrenamiento de los modelos, mientras que el 30% restante se reservará para llevar a cabo las pruebas pertinentes.

```
Deaths_ts <- ts(Deaths_ts,frequency=1)
tsize <- round(0.7 * nrow(Deaths_df))
train_deaths <- window(Deaths_ts,end=tsize)
test_deaths <- window(Deaths_ts,start=tsize+1)
```

Con el propósito de llevar a cabo un análisis cuantitativo exhaustivo, se presenta a continuación una tabla comparativa de los resultados derivados de las dos técnicas implementadas (después de realizar el entrenamiento, modelado y pronóstico), y la base de datos de prueba. Posteriormente, se exhiben gráficas representativas de estos resultados. En la Figura 8.2 se muestra el pronóstico mediante Holt-Winters acompañado de su respectivo intervalo de confianza. En contraste, en la Figura 8.3, la gráfica punteada en color rojo representa el comportamiento real de los datos, mientras que en azul se representa el pronóstico obtenido a través de la red MLP.

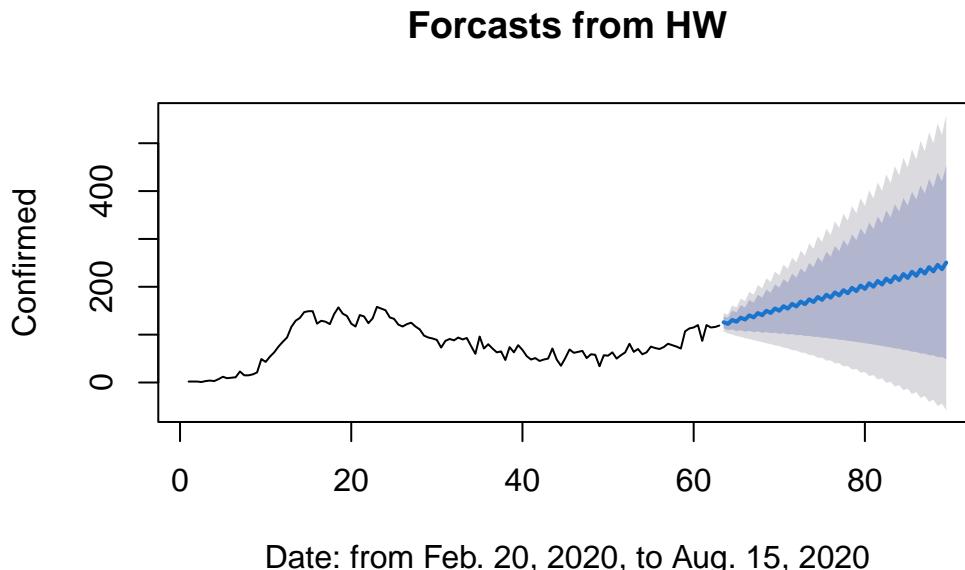


Figura 8.2.: Pronóstico obtenido mediante la técnica de Holt-Winters.

Tabla 8.1.: Comparación de Resultados entre las técnicas y los datos reales para evaluar precisión

	Date	Deaths	Forecast_HW	Forecast_MLP
t+1	2020-06-24	121	125.8962	118.8661
t+2	2020-06-25	133	122.3430	119.3998
t+3	2020-06-26	134	130.6941	119.7316
t+4	2020-06-27	109	126.9184	120.0653
t+5	2020-06-28	125	135.4921	120.3577
t+6	2020-06-29	144	131.4938	120.6225
t+7	2020-06-30	162	140.2900	120.8546
t+8	2020-07-01	147	136.0691	121.0584
t+9	2020-07-02	141	145.0880	121.2392
t+10	2020-07-03	148	140.6445	121.3995
t+11	2020-07-04	154	149.8859	121.5415
t+12	2020-07-05	148	145.2198	121.6672
t+13	2020-07-06	163	154.6839	121.7784
t+14	2020-07-07	160	149.7952	121.8767
t+15	2020-07-08	200	159.4818	121.9636
t+16	2020-07-09	153	154.3705	122.0403
t+17	2020-07-10	221	164.2798	122.1080
t+18	2020-07-11	142	158.9459	122.1678
t+19	2020-07-12	188	169.0777	122.2206
t+20	2020-07-13	194	163.5212	122.2671
t+21	2020-07-14	203	173.8757	122.3081
t+22	2020-07-15	179	168.0966	122.3442
t+23	2020-07-16	199	178.6736	122.3761
t+24	2020-07-17	198	172.6719	122.4042
t+25	2020-07-18	183	183.4716	122.4290
t+26	2020-07-19	188	177.2473	122.4508
t+27	2020-07-20	209	188.2695	122.4700
t+28	2020-07-21	217	181.8226	122.4869
t+29	2020-07-22	229	193.0675	122.5018
t+30	2020-07-23	219	186.3980	122.5149
t+31	2020-07-24	221	197.8654	122.5265
t+32	2020-07-25	215	190.9733	122.5367
t+33	2020-07-26	195	202.6634	122.5457
t+34	2020-07-27	216	195.5487	122.5536
t+35	2020-07-28	212	207.4613	122.5605
t+36	2020-07-29	235	200.1240	122.5666
t+37	2020-07-30	196	212.2593	122.5720
t+38	2020-07-31	226	204.6994	122.5768
t+39	2020-08-01	197	217.0572	122.5809
t+40	2020-08-02	216	209.2747	122.5846
t+41	2020-08-03	208	221.8552	122.5879
t+42	2020-08-04	215	213.8501	122.5907
t+43	2020-08-05	212	226.6531	145 122.5932
t+44	2020-08-06	185	218.4254	122.5954
t+45	2020-08-07	174	231.4511	122.5974
t+46	2020-08-08	156	223.0008	122.5991
t+47	2020-08-09	132	236.2490	122.6006
t+48	2020-08-10	163	227.5761	122.6019
t+49	2020-08-11	189	241.0469	122.6031
t+50	2020-08-12	184	232.1515	122.6041
t+51	2020-08-13	188	245.8449	122.6050
t+52	2020-08-14	174	223.7922	122.6056

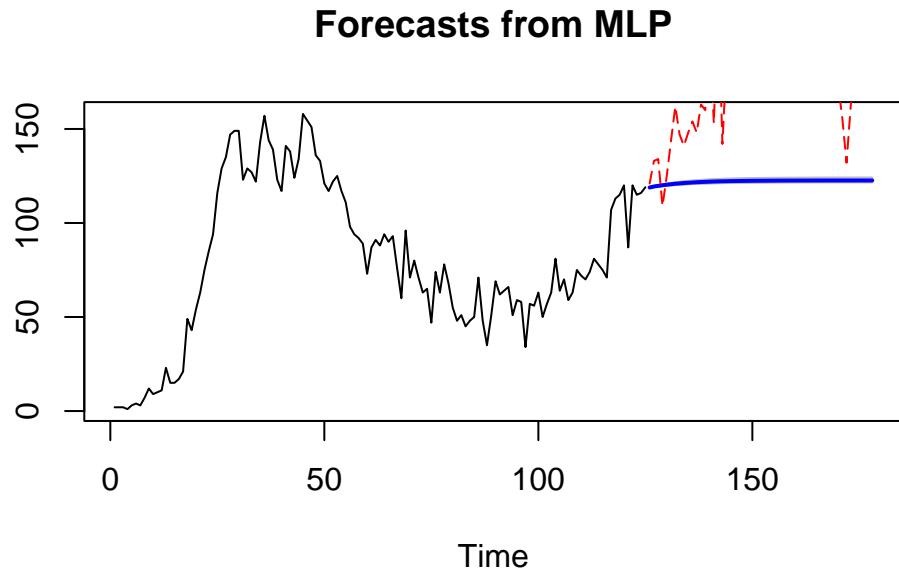


Figura 8.3.: Pronóstico obtenido mediante la red neuronal MLP.

A través de las Figura 8.2 y Figura 8.3, se observa que el pronóstico derivado de la técnica Holt-Winters muestra una mayor proximidad al comportamiento de la gráfica real.

Utilizando las métricas RMSE, MAE y MAPE, se lleva a cabo la evaluación de la calidad o bondad de ajuste de los métodos empleados en este estudio con el fin de seleccionar el modelo más adecuado.

Tabla 8.2.: Errores de los modelos para casos confirmados.

	<i>Training</i>			<i>Testing</i>		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Holt-Winters	12.6365	9.6103	19.4096	34.0665	25.5356	12.9770
MLP	11.7584	8.8446	15.6144	67.2031	59.7811	49.0686

Los resultados derivados de la tabla de errores (Tabla 8.2) llevan a la conclusión de que, en esta base de datos específica, a pesar de que la red MLP exhibe errores más reducidos durante la fase de entrenamiento, la técnica de Holt-Winters presenta un error considerablemente menor en la etapa de prueba en comparación con MLP. Esto evidencia su mayor eficacia en la predicción realizada. Por consiguiente, se recomienda el uso del método Holt-Winters para llevar a cabo el pronóstico de los próximos 30 días.

8.2. Pronóstico de los próximos 30 días

Tras la identificación del modelo óptimo, se procedió a prever el comportamiento futuro de la serie temporal de casos de muerte para los próximos 30 días utilizando dicho modelo. Se elaboraron representaciones gráficas de la predicción de estos casos, realizando una comparación de la efectividad entre las implementaciones de redes neuronales en la paquetería de R y la paquetería nativa de Python, las cuales se encuentran en las figuras Figura 8.4 y Figura 8.5, respectivamente.

8.2.1. Implementación en R

```
HW_deaths <- HoltWinters(ts(Deaths_df$Deaths,frequency = 2))
HW_for_d <- forecast(HW_deaths, h=30, level=c(80,95))
```

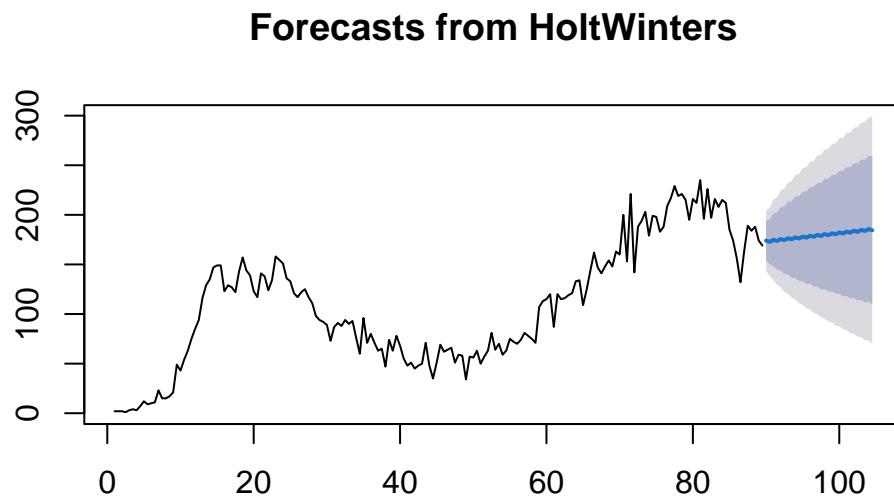


Figura 8.4.: Predicción futura de la serie tiempo para decesos diarios mediante el modelo Holt-Winters.

Los resultados del pronóstico indican que el 14 de septiembre de 2020 se proyectan 184 nuevos casos de muerte por COVID-19. Estos valores correspondientes al período de 30 días se detallan a continuación en la Tabla 8.3 .

Tabla 8.3.: Pronóstico de decesos por COVID-19 en Irán en los próximos 30 días

	Fecha	Decesos
1	2020-08-16	174.0023
2	2020-08-17	172.7223
3	2020-08-18	174.8375
4	2020-08-19	173.5575
5	2020-08-20	175.6728
6	2020-08-21	174.3927
7	2020-08-22	176.5080
8	2020-08-23	175.2279
9	2020-08-24	177.3432
10	2020-08-25	176.0631
11	2020-08-26	178.1784
12	2020-08-27	176.8983
13	2020-08-28	179.0136
14	2020-08-29	177.7335
15	2020-08-30	179.8488
16	2020-08-31	178.5688
17	2020-09-01	180.6840
18	2020-09-02	179.4040
19	2020-09-03	181.5192
20	2020-09-04	180.2392
21	2020-09-05	182.3545
22	2020-09-06	181.0744
23	2020-09-07	183.1897
24	2020-09-08	181.9096
25	2020-09-09	184.0249
26	2020-09-10	182.7448
27	2020-09-11	184.8601
28	2020-09-12	183.5800
29	2020-09-13	185.6953
30	2020-09-14	184.4152

8.2.2. Implementación en Python

Código

```
import numpy as np
import pandas as pd
import plotly.express as px
from statsmodels.tsa.holtwinters import ExponentialSmoothing

# Importación de los datos de CASOS DE MUERTE del 20-02 al 15-08-2020
data = pd.read_excel('muertes.xlsx')
df = pd.DataFrame(data)

# Configurar el modelo Holt-Winters
model = ExponentialSmoothing(df['Deaths'], trend='add',
                               seasonal='add', seasonal_periods=14)
model_fit = model.fit(optimized=True)

# Generar predicciones
forecast_days = 30 # Número de días a predecir
forecast = model_fit.forecast(steps=forecast_days)

# Crear un DataFrame con las fechas de predicción
forecast_dates = pd.date_range(start=df['Date'].max() +
                                pd.Timedelta(days=1), periods=forecast_days, freq='D')

# Agregar las fechas y valores predichos al DataFrame original
forecast_df = pd.DataFrame({'Date': forecast_dates,
                            'Forecast': forecast})
df_t = pd.concat([df, forecast_df])

# Crear una gráfica interactiva con Plotly
fig = px.line(df, x='Date', y='Deaths', title='Holt-Winters Forecast',
               labels={'Deaths': 'Deaths'})
fig.add_scatter(x=forecast_dates, y=forecast, mode='lines',
                 name='Forecast', line=dict(color='red'))
fig.show()
```

Los resultados del pronóstico en la Figura 8.5 indican que el 14 de septiembre de 2020 se proyectan aproximadamente 132 nuevos casos de muerte por COVID-19.

Holt-Winters Forecast

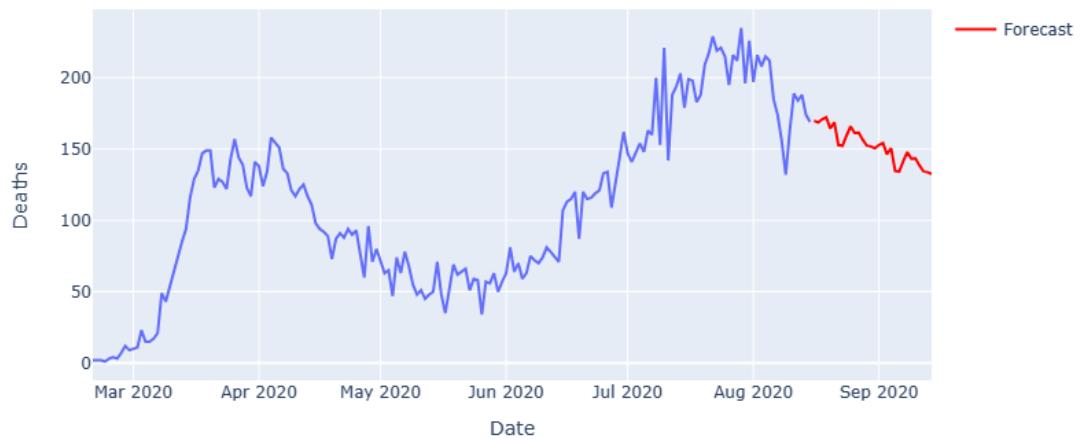


Figura 8.5.: Pronóstico de casos de muerte por COVID-19 en Irán en los próximos 30 días (implementación en Python)

9. Conclusiones

El presente estudio ha abordado la complejidad inherente a la modelación y pronóstico del número de casos confirmados y fallecidos por COVID-19 en Irán. A través de un análisis exhaustivo de datos recopilados hasta el 16 de enero de 2023, focalizado en el período comprendido entre el 20 de febrero y el 15 de agosto de 2020, se ha explorado la eficacia de distintos métodos de predicción. Los resultados obtenidos han arrojado luces sobre la idoneidad de las técnicas utilizadas y han resaltado la importancia de considerar diversos factores al seleccionar el enfoque predictivo más adecuado.

Durante el periodo analizado, se ha constatado que la técnica de redes neuronales Perceptrón Multicapa (MLP) muestra una notable eficacia en la predicción de la evolución de los casos confirmados de COVID-19. En contraste, para los fallecimientos asociados a esta enfermedad, la técnica de suavizamiento exponencial de Holt-Winters ha demostrado ser más precisa. Estos hallazgos subrayan la relevancia de adaptar el enfoque predictivo según las características específicas de los datos.

Además, aunque el análisis de estacionariedad no se ha incluido en esta etapa de la investigación, es importante reconocer que tanto la base de datos de casos confirmados como la de fallecimientos por COVID-19 han sido identificadas como no estacionarias. Este aspecto añade un nivel adicional de complejidad al proceso de modelación y predicción de estos eventos epidemiológicos.

En cuanto a las métricas de error utilizadas, se ha observado una variabilidad significativa entre las bases de datos de casos confirmados y de muerte. Mientras que en la primera, los errores RMSE y MAE han mostrado valores elevados, la métrica MAPE ha emergido como la más adecuada tanto en la fase de entrenamiento como en la de prueba. Por otro lado, en la base de datos de fallecimientos, las tres métricas de error han mostrado una mayor coherencia, reflejando la naturaleza menos oscilante de estos datos.

En relación a la efectividad entre las implementaciones de Holt-Winters y Perceptrón Multicapa en R y Python, se ha observado un rendimiento superior de Python en términos de capacidad para visualizar detalladamente el comportamiento de las predicciones. Si bien las implementaciones en Python han sido enriquecidas con modificaciones que pueden influir en los resultados, especialmente en el caso de la técnica de redes neuronales, este enfoque ha permitido un análisis más profundo de los datos.

Es crucial recalcar la importancia de limitar el horizonte de predicción para evitar estimaciones poco fiables. Aunque se haya respetado la cantidad de días predichos con

el objetivo de corroborar los hallazgos del estudio, se reconoce que esta elección puede influir en la precisión de las predicciones. Por lo tanto, se recomienda ejercer prudencia al establecer el horizonte de predicción en futuros estudios epidemiológicos, considerando cuidadosamente las limitaciones de los datos y los métodos de análisis utilizados.

Finalmente, una posible extensión de este trabajo para abordar la no estacionariedad de la serie temporal sería considerar la serie como la solución de un modelo epidemiológico SIR con perturbaciones aleatorias. Estos supuestos conducen al planteamiento del SIR como una ecuación diferencial estocástica con parámetros de transmisión y de recuperación desconocidos. El problema a resolver consistiría, en general, en desarrollar un algoritmo para muestrear la serie de tiempo con el fin de estimar dichos parámetros desconocidos de forma recursiva para el pronóstico de una cantidad limitada de días.

References

- Ash, R. B., y C. A. Doleans-Dade. 2000. *Probability and Measure Theory*. Elsevier Science. <https://books.google.com.mx/books?id=GkqQoRpCO2QC>.
- Castañeda, L. B., V. Arunachalam, y S. Dharmaraja. 2014. *Introduction to Probability and Stochastic Processes with Applications*. Wiley. <https://books.google.com.mx/books?id=M0hYBAAQBAJ>.
- Dickey, David A, y Wayne A Fuller. 1979. «Distribution of the estimators for autoregressive time series with a unit root». *Journal of the American statistical association* 74 (366a): 427-31.
- Dickey, David Alan. 1976. *Estimation and Hypothesis Testing in Nonstationary Time Series*. Iowa State University.
- Fuller, W. A. 1995. *Introduction to Statistical Time Series*. Wiley Series en Probability y Statistics. Wiley. <https://books.google.com.mx/books?id=wyRhjmAPQIYC>.
- Harvey, AC. 1981. «The econometric analysis of time series. Philip Allan». Oxford.
- Holt, Charles C. 1957. «Forecasting trends and seasonals by exponentially weighted moving averages». *ONR Memorandum* 52 (52): 5-10.
- Kourentzes, Nikolaos. 2022. «nnfor: Time Series Forecasting with Neural Networks». <https://CRAN.R-project.org/package=nnfor>.
- Lipschutz, S. 1996. *Probabilidad*. McGraw-Hill. <https://books.google.com.mx/books?id=vndlwgEACAAJ>.
- Mann, P. S. 2010. *Introductory Statistics*. John Wiley & Sons. https://books.google.com.mx/books?id=N_mEBiCYaqkC.
- McCulloch, Warren S, y Walter Pitts. 1943. «A logical calculus of the ideas immanent in nervous activity». *The bulletin of mathematical biophysics* 5: 115-33.
- Montesinos López, Osval Antonio, Abelardo Montesinos López, y Jose Crossa. 2022. «Fundamentals of Artificial Neural Networks and Deep Learning». En *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, 379-425. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-89010-0_10.
- Mood, A. M. F., F. A. Graybill, y D. C. Boes. 1986. *Introduction to the Theory of Statistics*. McGraw-Hill series en Probability y Statistics. <https://books.google.com.mx/books?id=bKHBjgEACAAJ>.
- R Core Team. 2023. «R: A Language and Environment for Statistical Computing». <https://www.R-project.org/>.
- Rosenblatt, Frank. 1960. «Perceptron simulation experiments». *Proceedings of the IRE* 48 (3): 301-9.
- Ross, S. M. 1995. *Stochastic Processes*. Wiley series en probability y mathematical statistics. Wiley. <https://books.google.com.mx/books?id=qILdCQAAQBAJ>.

- Segall, R. S., y G. Niu. 2022. *Biomedical and Business Applications Using Artificial Neural Networks and Machine Learning*. Advances en Computational Intelligence y Robotics. IGI Global. <https://books.google.com.mx/books?id=9G9bEAAAQBAJ>.
- Sosa Jerez, Lexly Vanessa, Laura Camila Zamora Alvarado, et al. s. f. «Estructura de redes neuronales (MLP) y su aplicación como aproximador universal». {B.S.} thesis.
- Stewart, J. 2017. *Cálculo de varias variables: trascendentes tempranas*. Cengage Learning. <https://books.google.com.mx/books?id=bKSvtAEACAAJ>.
- Talkhi, Nasrin, Narges Akhavan Fatemi, Zahra Ataei, y Mehdi Jabbari Nooghabi. 2021. «Modeling and forecasting number of confirmed and death caused COVID-19 in IRAN: A comparison of time series forecasting methods». *Biomedical signal processing and control* 66: 102494.
- Wackerly, D. D., D. D. Wackerly, W. Mendenhall, y R. L. Scheaffer. 2009. *Estadística Matemática Con Aplicaciones*. CENGAGE Learning. <https://books.google.com.mx/books?id=8bTfwAEACAAJ>.
- Waldmeier, Max. 1961. «The sunspot-activity in the years 1610-1960». *Zurich: Schulthess*.
- Winters, Peter R. 1960. «Forecasting sales by exponentially weighted moving averages». *Management science* 6 (3): 324-42.
- Woodward, W. A., B. P. Sadler, y S. Robertson. 2022. *Time Series for Data Science: Analysis and Forecasting*. A Chapman & Hall Book. CRC Press, Taylor & Francis Group. <https://books.google.com.mx/books?id=gM3gzgEACAAJ>.
- Yaglom, A. M. 1962. *An Introduction to the Theory of Stationary Random Functions*. Selected Russian publications en the mathematical sciences. Prentice-Hall. https://books.google.com.mx/books?id=l_JvAAAAIAAJ.
- Yule, George Udny. 1971. «On a method of investigating periodicities in disturbed series with special reference to Wolfer's sunspot numbers». *Statistical Papers of George Udny Yule*, 389-420.
- Zhang, Ping. 1993. «Model Selection Via Multifold Cross Validation». *The Annals of Statistics* 21 (1): 299-313. <http://www.jstor.org/stable/3035592>.