Modelos Estadísticos para la Predicción

Víctor Aceña - Isaac Martín

2025-08-07

Table of contents

Prefacio

Los modelos estadísticos han emergido como herramientas fundamentales en la era de la información, donde la capacidad de analizar y predecir comportamientos a partir de datos se ha convertido en una habilidad esencial. En este contexto, los modelos para la predicción juegan un papel crucial al permitirnos describir y cuantificar las relaciones entre variables, así como anticipar resultados futuros. Este libro está diseñado para proporcionar una comprensión profunda y práctica de estas técnicas, basándose en el contenido de la asignatura impartida en el **Grado en Matemáticas**.

A lo largo de los capítulos, encontrarás una combinación de teoría rigurosa y aplicaciones prácticas. Se abordarán temas como la regresión lineal simple y múltiple, métodos de selección de variables y regularización, ingeniería de características y modelos generalizados, entre otros. Además, todos los conceptos se ilustrarán con ejemplos en **R**, permitiéndote aplicar lo aprendido a conjuntos de datos reales.

El objetivo de este libro es doble: por un lado, proporcionar herramientas avanzadas para analizar relaciones sujetas a incertidumbre y, por otro, capacitarte para elegir el método más apropiado para resolver problemas de predicción o explicación, analizando la naturaleza de las variables y sus posibles interacciones. Al finalizar, habrás desarrollado una comprensión sólida de los modelos estadísticos y estarás preparado para enfrentar desafíos en el análisis predictivo con confianza y creatividad.

Filosofía pedagógica del volumen

La filosofía que subyace a la obra es un enfoque **"teórico-práctico"** deliberado y sin concesiones. No nos conformamos con una mera aplicación de "recetas" o una guía de funciones de software. Buscamos fomentar una comprensión profunda del *modus operandi* de cada modelo y método. Perseguimos un equilibrio entre la técnica estadística y la estrategia de resolución de problemas, bajo la firme convicción de que la labor práctica se desarrolla con mayor fluidez, creatividad y éxito cuando se cimienta en una comprensión robusta de los principios matemáticos y estadísticos subyacentes, tal y como defiende (Harrell 2015) en su influyente obra.

¿Qué aprenderás con este libro?

Al completar este recorrido, habrás desarrollado habilidades clave para:

- **Modelar** la dependencia entre una variable respuesta y múltiples predictores en conjuntos de datos complejos.
- **Resolver** problemas con iniciativa y creatividad, eligiendo las técnicas estadísticas más adecuadas para cada caso.
- **Evaluar** de forma crítica las ventajas e inconvenientes de diferentes alternativas metodológicas.
- **Implementar** estos modelos utilizando software estadístico profesional como R.
- **Interpretar** correctamente los resultados, proponer mejoras y tomar decisiones basadas en datos.
- **Adquirir** las competencias y la autonomía necesarias para emprender con éxito estudios de posgrado o proyectos profesionales en ciencia de datos.

Agradecemos a los profesores y colegas que han contribuido al desarrollo de esta asignatura y a la elaboración de este libro. Su dedicación y conocimiento han sido fundamentales para la creación de este recurso.

Esperamos que esta guía te resulte útil y enriquecedora.

¡Comenzamos!

! Grado en Matemáticas

Este libro presenta el material de la asignatura de Modelos Estadísticos para la Predicción del Grado en Matemáticas de la Universidad Rey Juan Carlos. Su contenido está fuertemente relacionado con las asignaturas de Estadística Matemática y Minería de Datos.

! Conocimientos previos

Es altamente recomendable que los alumnos que cursen esta materia manejen con soltura los conocimientos adquiridos en las asignaturas de Probabilidad y Estadística Matemática, así como herramientas de cálculo univariante, multivariante y álgebra lineal.

i Sobre los autores

Víctor Aceña Gil es graduado en Matemáticas por la UNED, máster en Tratamiento Estadístico y Computacional de la Información por la UCM y la UPM, doctor en Tecnologías de la Información y las Comunicaciones por la URJC y profesor del departamento de Informática y Estadística de la URJC. Miembro del grupo de investigación de alto

rendimiento en Fundamentos y Aplicaciones de la Ciencia de Datos, DSLAB, de la URJC. Pertenece al grupo de innovación docente, DSLAB-TI.
Isaac Martín de Diego es diplomado en Estadística por la Universidad de Valladolid (UVA), licenciado en Ciencias y Técnicas Estadísticas por la Universidad Carlos III de Madrid (UC3M), doctor en Ingeniería Matemática por la UC3M, catedrático de Ciencias de la Computación e Inteligencia Artificial del departamento de Informática y Estadística de la URJC. Es fundador y coordinador del DSLAB y del DSLAB-TI.

1 Introducción a los modelos de regresión

Este tema inaugural tiene como misión construir el andamiaje conceptual y filosófico sobre el que se asienta el modelado estadístico moderno. A lo largo de estas páginas, contextualizaremos la regresión no solo como una técnica, sino como un marco de pensamiento indispensable en la ciencia de datos y en cualquier disciplina de investigación cuantitativa. Exploraremos en profundidad su propósito dual, desgranaremos sus componentes axiomáticos hasta el último detalle, y ofreceremos una visión panorámica, rica en matices, de la vasta familia de modelos de regresión. El objetivo es preparar al lector, con solidez y sin prisas, para las inmersiones técnicas que seguirán en los capítulos posteriores. Como lectura complementaria que comparte esta filosofía de aprendizaje profundo pero aplicado, recomendamos encarecidamente la obra de (James et al. 2021).

1.1 Predecir vs. explicar

El modelado de regresión constituye una de las herramientas más potentes y flexibles del arsenal estadístico. Ofrece un marco metodológico riguroso para investigar y cuantificar las relaciones entre un conjunto de variables, y su aplicabilidad abarca un espectro extraordinariamente amplio de disciplinas: desde la física de partículas y la ingeniería aeroespacial, donde se usa para modelar sistemas complejos, hasta la econometría, la psicometría, la epidemiología o las finanzas, donde es fundamental para entender mercados y comportamientos.

Aunque en la práctica ambos objetivos a menudo se entrelazan, conceptualmente, el modelado estadístico se orienta hacia uno de dos polos, una dicotomía fundamental articulada brillantemente por (Shmueli 2010): la **predicción** o la **inferencia (explicación)**. Comprender esta distinción es el primer paso para convertirse en un modelador eficaz.

1. **Predicción**: El objetivo principal es la **precisión**. Se busca construir un modelo que pueda estimar con el menor error posible el valor de una variable de interés (la *respuesta*) basándose en la información proporcionada por otras variables (las *predictoras*). En este paradigma, el modelo puede ser tratado como una "caja negra" (*black box*). Su funcionamiento interno o la interpretabilidad de sus componentes son secundarios, siempre y cuando sus predicciones sean consistentemente fiables y robustas en datos no observados previamente.

Ejemplo

Una entidad financiera quiere predecir la probabilidad de que un cliente incurra en impago de un crédito. Utilizan variables como la edad, ingresos, nivel de estudios y historial crediticio. El banco no necesita necesariamente entender la "causa" exacta del impago; su principal interés es tener un modelo que clasifique correctamente a los futuros solicitantes como de alto o bajo riesgo para minimizar pérdidas.

2. **Inferencia**: El foco se desplaza radicalmente hacia la **comprensión** y la **interpretación**. El objetivo no es solo predecir, sino dilucidar la naturaleza de las interdependencias entre las variables. Se busca cuantificar cómo un cambio en una variable predictora influye, ya sea de forma causal o asociativa, en la variable de respuesta. Aquí, la interpretabilidad del modelo es primordial. El interés reside en la magnitud, el signo y, crucialmente, la incertidumbre estadística (expresada mediante errores estándar, intervalos de confianza y p-valores) de los parámetros estimados.

Ejemplo

Una epidemióloga investiga los factores de riesgo de una enfermedad cardíaca. Modela la presión arterial en función de variables como el índice de masa corporal (IMC), el consumo diario de sal y las horas de ejercicio semanales. Su objetivo no es solo predecir la presión arterial de un paciente, sino entender y cuantificar la relación: "¿En cuántos mmHg aumenta la presión arterial, en promedio, por cada gramo adicional de sal consumido al día, manteniendo constantes el IMC y el ejercicio?". La respuesta a esta pregunta tiene implicaciones directas para la salud pública y las recomendaciones dietéticas.

Una relación simbiótica

Aunque conceptualmente distintos, ambos objetivos no son mutuamente excluyentes; a menudo se benefician el uno del otro. Un modelo con una base inferencial sólida, que captura relaciones causales o asociativas verdaderas, suele tener un buen rendimiento predictivo. A la inversa, un modelo que demuestra una alta precisión predictiva en datos nuevos nos da confianza en que las relaciones que ha aprendido no son meras casualidades del conjunto de datos de entrenamiento, sino que probablemente reflejen patrones reales y generalizables. La tensión entre interpretabilidad y precisión es uno de los debates más fascinantes en la ciencia de datos moderna.

1.2 Anatomía de un modelo de regresión: los componentes axiomáticos

Todo modelo de regresión, desde el más simple hasta el más sofisticado, se construye sobre tres pilares fundamentales. Estos componentes, definidos en textos clásicos como el de (Kutner et al. 2005), son los ladrillos con los que edificaremos todo nuestro conocimiento.

1.2.1 La variable de respuesta

También designada como **variable dependiente**, **variable de salida**, **target**, **variable objetivo** o **variable explicada**. Representa el fenómeno o la característica principal cuyo comportamiento se busca modelar, comprender o predecir. La naturaleza de esta variable es, quizás, el factor más determinante a la hora de elegir el tipo de modelo de regresión. Puede ser:

- **Continua**: Una variable que puede tomar cualquier valor dentro de un rango. Ej: temperatura, altura, precio de una acción, concentración de un compuesto químico.
- **Discreta de Conteo**: Una variable que representa un número de eventos. Ej: número de accidentes en una intersección, número de clientes que entran en una tienda, número de mutaciones en un gen.
- **Binaria o Dicotómica**: Una variable con solo dos resultados posibles. Ej: éxito/fracaso, enfermo/sano, compra/no compra, spam/no spam.
- **Categórica**: Una variable que representa grupos o categorías. Si no tiene orden, es **nominal** (ej: tipo de sangre, partido político); si tiene un orden intrínseco, es **ordinal** (ej: nivel de satisfacción "bajo/medio/alto", estadio de una enfermedad "I/II/III/IV").

1.2.2 Las variables predictoras

Conocidas indistintamente como **variables independientes**, **explicativas**, **regresoras**, **covariables** o **características** (*features*). Son las magnitudes, atributos o factores que se postula que influyen o están asociados con el comportamiento de la variable de respuesta. Al igual que la variable de respuesta, pueden ser de diversa naturaleza (continuas, categóricas, etc.). La selección de estas variables es una de las fases más críticas del modelado, requiriendo una combinación de conocimiento del dominio, análisis exploratorio de datos y técnicas estadísticas formales.

1.2.3 El término de error aleatorio

Este componente, a menudo subestimado, es conceptualmente crucial. Simboliza la variabilidad intrínseca de la variable de respuesta que **no es capturada o explicada** por las variables

predictoras incluidas explícitamente en el modelo. El término de error ϵ no es un simple "error" en el sentido de equivocación; es un componente estocástico que amalgama múltiples fuentes de variabilidad:

- **Variables Omitidas**: Ningún modelo es perfecto. Siempre habrá factores que influyen en Y pero que no han sido medidos o incluidos en el modelo (variables latentes).
- **Error de Medición**: Las mediciones de Y (y también de X) pueden no ser perfectamente precisas.
- **Aleatoriedad Intrínseca**: Muchos fenómenos naturales y sociales tienen un componente de variabilidad irreducible. Dos individuos con idénticos valores en todas las variables predictoras pueden, aun así, tener valores distintos en la variable de respuesta.

Formalmente, la relación fundamental de la regresión se expresa como la descomposición de la variable de respuesta en una parte sistemática y una parte aleatoria:

$$Y = \underbrace{f(X_1, \dots, X_k)}_{\text{Componente Sistemática}} + \underbrace{\epsilon}_{\text{Componente Aleatoria}}$$

donde $f(\cdot)$ denota la **componente sistemática** (o determinística) del modelo, que representa el valor esperado de Y para unos valores dados de las X. La función f es lo que intentamos estimar a partir de los datos. Por su parte, ϵ es la **componente aleatoria**, y gran parte del diagnóstico y la inferencia en regresión se basa en verificar los supuestos que hacemos sobre la distribución de este término (ej: que su media es cero, que su varianza es constante, etc.).

🔥 Linealidad en los parámetros, no en las variables

Una característica que define a los **modelos de regresión lineal** (y que se extiende a muchos otros tipos de modelos) es que la función $f(\cdot)$ mantiene una relación lineal con respecto a sus **parámetros desconocidos** (los coeficientes beta, β_j). Es crucial enfatizar que esta "linealidad en los parámetros" **no impone una restricción de linealidad en las variables predictoras mismas**.
Por el contrario, es común y metodológicamente válido incorporar transformaciones no lineales de los predictores o interacciones complejas entre ellos para capturar relaciones más sofisticadas. Por ejemplo, el siguiente modelo es un **modelo de regresión lineal**:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 \log(X_2) + \beta_4 (X_1 \cdot X_2) + \epsilon$$

Aunque la relación entre Y y las variables X_1 y X_2 es claramente no lineal (es cuadrática en X_1, logarítmica en X_2 e incluye una interacción), el modelo es **lineal en los parámetros** $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$. La función f es una combinación lineal de estos coeficientes. Esta flexibilidad es una de las razones de la enorme potencia de los modelos lineales.
El siguiente bloque de código en R genera un ejemplo visual. Simulamos datos que siguen una relación cuadrática y luego ajustamos un modelo lineal que incluye un tér-

mino cuadrático (X^2). Como se puede observar en la figura, la línea de regresión (azul) captura perfectamente la curvatura de los datos, demostrando que un modelo lineal en sus parámetros puede modelar relaciones no lineales en sus variables.

```
# Cargar la librería necesaria para la visualización
library(ggplot2)

# 1. Simulación de datos
set.seed(42) # Para reproducibilidad
n <- 100 # Número de observaciones
x <- runif(n, -5, 5)
# La relación verdadera es cuadrática: y = 1.5 + 0.5*x + 0.8*x^2 + error
y <- 1.5 + 0.5 * x + 0.8 * x^2 + rnorm(n, mean = 0, sd = 5)
datos <- data.frame(x, y)

# 2. Ajuste del modelo lineal
# Usamos I(x^2) para indicar que tratamos x^2 como una variable
modelo_cuadratico <- lm(y ~ x + I(x^2), data = datos)

# 3. Visualización con ggplot2
ggplot(datos, aes(x = x, y = y)) +
  geom_point(alpha = 0.6, color = "gray40") + # Puntos de los datos originales
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), se = FALSE, color = "#0072B2", size
  labs(
    title = "Modelo Lineal con Término Cuadrático",
    x = "Variable Predictora (X)",
    y = "Variable de Respuesta (Y)"
  ) +
  theme_classic(base_size = 14)
```

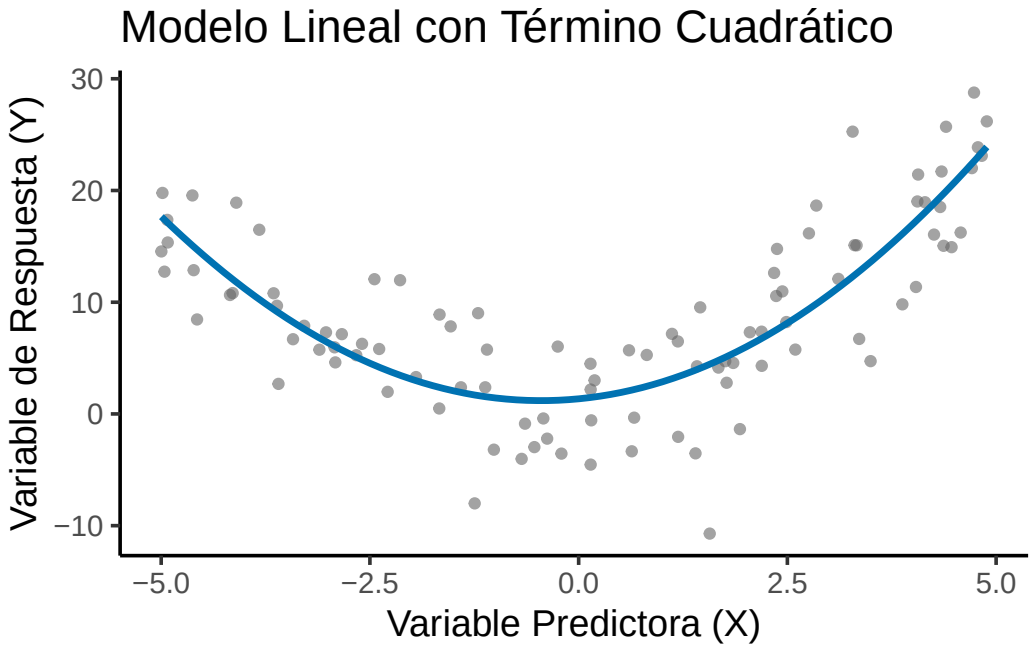


Figure 1.1: Ejemplo de un modelo lineal en los parámetros que captura una relación no lineal (cuadrática) en los datos.

1.3 Un viaje preliminar por el universo de los modelos de regresión

La regresión lineal clásica, que será el objeto de estudio de los primeros capítulos, es el punto de partida y la piedra angular sobre la cual se erige una prolífica y fascinante gama de metodologías estadísticas avanzadas. Este volumen se dedicará a desentrañar con rigor las siguientes extensiones y especializaciones, que permiten al analista abordar una variedad casi infinita de problemas.

1.3.1 Modelos lineales (LMs)

Constituyen el paradigma fundamental, el alfabeto sobre el que se escribe el lenguaje del modelado estadístico. Son mucho más que una simple técnica para ajustar una recta a una nube de puntos; son el laboratorio donde se forjan y se comprenden los conceptos esenciales que nos acompañarán durante todo nuestro viaje. Es aquí donde aprenderemos a:

- **Estimar parámetros** e interpretar su significado en el contexto del problema.

- **Cuantificar la incertidumbre** de nuestras estimaciones mediante errores estándar e intervalos de confianza.
- **Realizar contrastes de hipótesis** para evaluar si la relación entre nuestras variables es estadísticamente significativa o fruto del azar.
- **Diagnosticar la "salud" de un modelo**, examinando si los supuestos sobre los que se construye son razonables para nuestros datos.

En su forma más clásica, el modelo lineal asume que la variable de respuesta (y, por consecuencia, el término de error aleatorio) sigue una distribución Normal o Gaussiana. Esta asunción es la clave que desbloquea todo el elegante aparato de la inferencia estadística, permitiéndonos realizar pruebas exactas y derivar propiedades matemáticas bien conocidas. Técnicas tan ubicuas en la ciencia como el Análisis de la Varianza (ANOVA) o el Análisis de la Covarianza (ANCOVA) no son más que casos particulares de la gran familia de los modelos lineales, un hecho que unifica campos de la estadística que históricamente se estudiaban por separado. Dominar los LMs es, sencillamente, un requisito indispensable.

1.3.2 Modelos lineales generalizados (GLMs)

Si los LMs son el alfabeto, los GLMs son la gramática que nos permite construir frases complejas y con significado en una variedad de contextos mucho más amplia. Introducidos en el influyente y verdaderamente revolucionario trabajo de (Nelder and Wedderburn 1972), los GLMs representan un salto conceptual que expande de forma masiva el universo de problemas que podemos abordar. Suponen una generalización elegante que nos permite escapar de la "tiranía" de la distribución Normal y modelar respuestas con una variedad mucho más amplia de naturalezas y escalas.

Esta flexibilidad se logra mediante la combinación de dos ingeniosos mecanismos que son el corazón de la teoría:

1. **La familia exponencial de distribuciones**: Los GLMs no funcionan con cualquier distribución, sino con aquellas que pertenecen a una "familia" matemática con propiedades muy convenientes: la **familia exponencial**. Este "club" de distribuciones es muy selecto, pero incluye a miembros tan importantes como la Normal, la Poisson (para datos de conteo), la Binomial (para datos de proporciones o binarios), la Gamma (para datos continuos positivos y asimétricos) o la Binomial Negativa. Su estructura matemática común permite desarrollar una teoría unificada para la estimación de parámetros, lo que es un logro teórico de primer orden.
2. **La función de enlace (link function)**: Este es el verdadero golpe de genialidad. El predictor lineal de nuestro modelo, $X\beta$, puede tomar cualquier valor en la recta real, desde $-\infty$ hasta $+\infty$. Sin embargo, la media de nuestra variable de respuesta, $E[Y] = \mu$, a menudo está restringida. Por ejemplo, una probabilidad (μ en un modelo binomial) debe estar entre 0 y 1; un conteo (μ en un modelo de Poisson) debe ser

positivo. La función de enlace, $g(\cdot)$, actúa como un **"traductor"** o un **"puente"** que conecta estos dos mundos. Transforma la media restringida de la respuesta para que pueda ser modelada por el predictor lineal no restringido. La relación fundamental es, por tanto, $g(E[Y]) = g(\mu) = X\beta$.

- Para datos de **conteo** (Poisson), se usa un **enlace logarítmico** ($g(\mu) = \log(\mu)$). Esto garantiza que, al invertir la función para obtener la media ($\mu = \exp(X\beta)$), el resultado será siempre positivo, como debe ser un conteo.
- Para datos **binarios** (Binomial), se usa un **enlace logit** ($g(\mu) = \log(\frac{\mu}{1-\mu})$). Esta función toma una probabilidad μ en el rango (0, 1) y la proyecta sobre toda la recta real, permitiendo que sea modelada por $X\beta$.

Gracias a los GLMs, podemos usar el mismo marco conceptual de la regresión lineal para modelar una gama de fenómenos increíblemente diversa, desde predecir la cantidad de ciclistas en una ciudad (Poisson) hasta la probabilidad de que un paciente responda a un tratamiento (logística).

1.3.3 Modelos de efectos mixtos (Mixed Models)

Su desarrollo responde a la necesidad crítica de analizar datos que exhiben estructuras de dependencia o correlación, como agrupamientos, anidamientos o jerarquías. En datos estándar, asumimos que las observaciones son independientes, pero esta asunción se viola en casos como: * **Medidas repetidas** sobre los mismos sujetos (ej: medir la presión arterial de un paciente cada mes). * **Datos longitudinales** (un tipo de medida repetida a lo largo del tiempo). * **Datos agrupados** (ej: estudiantes anidados dentro de clases, que a su vez están anidadas dentro de colegios). Estos modelos, detallados en obras como la de (Pinheiro and Bates 2000), introducen explícitamente una estructura de correlación en el término de error mediante la incorporación de **efectos aleatorios**, que permiten capturar la variabilidad entre los diferentes grupos o individuos, además de los **efectos fijos** que representan a la población general.

1.3.4 Modelos aditivos generalizados (GAMs)

Representan una extensión natural y altamente flexible de los GLMs que relaja el supuesto de linealidad entre el predictor transformado y las covariables. Los GAMs, cuya implementación moderna se debe en gran parte al trabajo de (Wood 2017), permiten modelar estas relaciones mediante **funciones suaves** no paramétricas (como *splines*), manteniendo al mismo tiempo la estructura aditiva del modelo. La forma general es $g(\mu) = \alpha + f_1(x_1) + f_2(x_2) + ... + f_p(x_p)$, donde las $f_i(\cdot)$ son funciones suaves de los predictores estimadas a partir de los datos. Esto permite capturar patrones no lineales complejos sin necesidad de especificar una forma funcional paramétrica a priori, logrando un equilibrio excepcional entre flexibilidad e interpretabilidad.

R como lenguaje del modelado estadístico

Este compendio no es un texto puramente teórico. Fusiona intrínsecamente la exposición de los conceptos con su aplicación computacional directa a través del lenguaje y entorno estadístico **R**. R se ha consolidado como el estándar de facto en la investigación estadística y la ciencia de datos académica por su potencia, flexibilidad y el inmenso ecosistema de paquetes contribuidos por la comunidad científica. Se presupone en el lector una familiaridad operativa básica con R, y se fomenta activamente el desarrollo de una fluidez progresiva mediante la reproducción, modificación y experimentación con los numerosos ejemplos y fragmentos de código presentados.
La capacidad de ejecutar análisis en R es fundamental para todo el ciclo de vida del modelado:

- La **exploración de datos** y la visualización inicial.
- La **estimación de parámetros** y el ajuste de los modelos.
- El **diagnóstico riguroso** de la adecuación del modelo y la validación de sus supuestos.
- La **producción de gráficos** y tablas de alta calidad para comunicar los resultados.

En R, las herramientas fundamentales para la regresión lineal (`lm()`) y los modelos lineales generalizados (`glm()`) están incluidas en el paquete **`stats`**, que es uno de los **paquetes base** y se carga automáticamente con cada sesión. Por lo tanto, no necesitamos instalarlo ni cargarlo.
A lo largo del libro, extenderemos esta funcionalidad base con paquetes especializados que sí requieren instalación y carga. Entre los más importantes que usaremos se encuentran:

- `mgcv`: La implementación de referencia para GAMs, mantenida por su creador, Simon Wood, y citada en (Wood 2017).
- `lme4` y `nlme`: Los dos paquetes fundamentales para el ajuste de modelos de efectos mixtos, desarrollados por los pioneros en el campo (Pinheiro and Bates 2000; Bates et al. 2015).
- `rms`: Un paquete y una filosofía de trabajo para implementar estrategias de modelado de regresión robustas, como se detalla en la obra de (Harrell 2015).
- `gamair`: Contiene numerosos conjuntos de datos que acompañan al libro de (Wood 2017), ideales para practicar con GAMs.

1.4 Una breve crónica del desarrollo de la regresión

1.4.1 Los orígenes: Galton y la "regresión a la mediocridad"

La gestación de la metodología de regresión se traza hasta las investigaciones pioneras de Sir **Francis Galton**, un polímata de la era victoriana. A finales del siglo XIX, estudiando la herencia de la estatura, Galton recopiló datos de padres e hijos y notó un fenómeno curioso: los padres muy altos tendían a tener hijos altos, pero, en promedio, no tan altos como ellos. Análogamente, los padres muy bajos tenían hijos bajos, pero no tan bajos como ellos. Acuñó el término **"regresión a la mediocridad"** (hoy diríamos "regresión a la media") para describir esta tendencia de las características de la descendencia a "regresar" hacia la media de la población, en lugar de perpetuar los extremos de los progenitores (Galton 1886).

Estudios de Galton sobre estatura

Datos recopilados

- Galton recopiló datos sobre las estaturas de **928 hijos** y sus respectivos **padres**.
- Las medidas fueron expresadas en pulgadas (1 pulgada = 2.54 cm).

- En sus análisis, utilizó el promedio de las estaturas de ambos padres, conocido como **estatura media parental**, para compararlo con la estatura de los hijos.

Principales hallazgos

1. **Relación lineal entre padres e hijos:**
   Galton observó que existe una relación positiva entre la estatura de los padres y la de los hijos. Los padres altos tienden a tener hijos altos, y los padres bajos tienden a tener hijos bajos. Esta relación puede modelarse con una línea recta, lo que inspiró la formulación de la regresión lineal.
2. **Regresión a la media:**
   - Aunque los hijos de padres altos son, en promedio, más altos que el promedio general de la población, también tienden a ser **menos altos que sus padres**.
   - De manera similar, los hijos de padres bajos son más bajos que el promedio general, pero suelen ser **menos bajos que sus padres**.
   - Este fenómeno, que Galton llamó "regresión a la media", ocurre porque las características extremas tienden a suavizarse en la siguiente generación debido a la influencia de múltiples factores genéticos y ambientales.

3. **Ecuación de la recta de regresión:**
   Galton ajustó una recta para describir la relación entre la estatura media parental (X) y la estatura de los hijos (Y):

$$Y = \beta_0 + \beta_1 X$$

   Donde:

   - β_0: Intercepto, representa la estatura promedio de los hijos cuando la estatura parental es promedio.
   - β_1: Pendiente, indica cómo cambia la estatura de los hijos por cada unidad de cambio en la estatura media parental.

Importancia en la Estadística

1. **Regresión lineal:**
   Este estudio introdujo el concepto de **recta de regresión**, que describe cómo varía la media de una variable dependiente en función de una variable independiente.

2. **Correlación:**
   Galton también estudió el grado de relación entre variables, precursor del concepto de **coeficiente de correlación** desarrollado posteriormente por Karl Pearson, un discípulo suyo.

3. **Regresión a la media:**
   El término y la idea detrás de "regresión a la media" surgieron de estos estudios y son hoy fundamentales en estadística y genética.

Ejemplo Gráfico
Galton representó sus datos en gráficos de dispersión, mostrando cómo los puntos (pares de estatura media parental y estatura de los hijos) se agrupan alrededor de la recta de regresión, ilustrando la tendencia general de la relación.

```r
# Cargar los paquetes necesarios
library(ggplot2)
library(HistData)

# Cargar los datos de Galton
data("GaltonFamilies")

# Crear el modelo de regresión lineal para obtener los coeficientes
modelo <- lm(childHeight ~ midparentHeight, data = GaltonFamilies)

# Crear la etiqueta para la ecuación de la recta de forma más limpia
# Usamos sprintf() para un formato más controlado y legible
eq_label <- sprintf("y = %.2f + %.2f * x", coef(modelo)[1], coef(modelo)[2])

# --- Gráfico Mejorado ---
# Usamos un tema más limpio y colores más suaves para una apariencia profesional.
# geom_jitter() es mejor que geom_point() para estos datos, ya que evita la superposición c
ggplot(GaltonFamilies, aes(x = midparentHeight, y = childHeight)) +

  # 1. Puntos de datos: Usamos geom_jitter para visualizar mejor los puntos superpuestos
  #    y añadimos transparencia (alpha) para ver la densidad.
  geom_jitter(alpha = 0.3, color = "gray50", width = 0.1, height = 0.1) +

  # 2. Línea de regresión: En un color azul profesional y más gruesa para que destaque.
  geom_smooth(method = "lm", se = FALSE, color = "#0072B2", size = 1.2) +

  # 3. Anotación: Añadimos la ecuación de la recta de forma elegante,
  #    usando el mismo color que la línea para crear cohesión visual.
  annotate(
    "text",
    x = 66, y = 74, # Posición ajustada para mejor visibilidad
    label = eq_label,
    color = "#0072B2", # Mismo color que la línea
    size = 4.5, # Tamaño de la fuente
    fontface = "italic" # Cursiva para la ecuación
  ) +

  # 4. Títulos y etiquetas: Mejorados para mayor claridad y contexto.
  #    Añadimos un subtítulo y una fuente.
  labs(
    title = "Regresión de la Estatura de Hijos vs. Padres",
    x = "Promedio de Estatura de los Padres (pulgadas)",
    y = "Estatura del Hijo/a (pulgadas)",
    caption = "Fuente: Paquete HistData de R"
  ) +

  # 5. Tema: Usamos un tema limpio y profesional como base.
  theme_classic(base_size = 14)
```

```
`geom_smooth()` using formula = 'y ~ x'
```

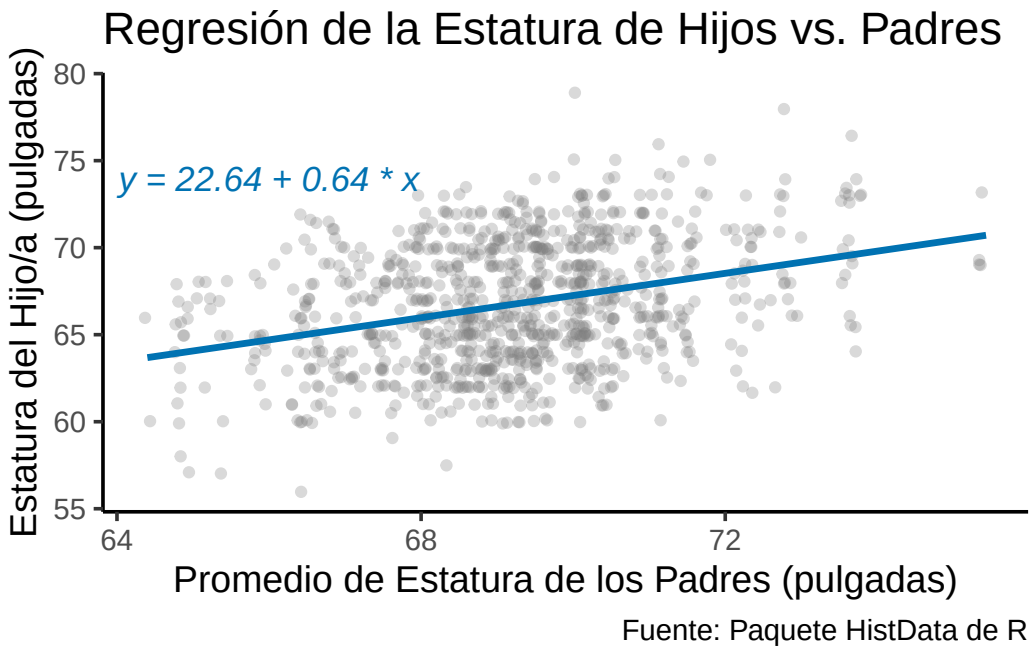


Figure 1.2: Datos históricos del estudio sobre la 'regresión a la media'

1.4.2 La formalización matemática: Legendre y Gauss

Aunque Galton sentó las bases conceptuales e introdujo el término, la formalización matemática de la estimación de parámetros en modelos lineales se atribuye a dos de los más grandes matemáticos de la historia. **Adrien-Marie Legendre** publicó en 1805 el "Método de los mínimos cuadrados" como un procedimiento numérico para ajustar observaciones astronómicas. Pocos años después, **Carl Friedrich Gauss** no solo publicó que había desarrollado el mismo método de forma independiente años antes, sino que lo dotó de una profundidad teórica mucho mayor, conectándolo con la teoría de la probabilidad y derivándolo bajo el supuesto de errores distribuidos normalmente, convirtiéndolo en la técnica fundamental para la estimación en modelos lineales que sigue siendo hoy.

1.4.3 El desarrollo moderno: la revolución de los GLMs

A lo largo del siglo XX, la regresión experimentó un desarrollo explosivo. Sin embargo, el hito que probablemente más ha influido en la práctica estadística moderna fue la publicación del

artículo sobre **Modelos Lineales Generalizados (GLMs)** por **John Nelder** y **Robert Wedderburn** en 1972 (Nelder and Wedderburn 1972). Esta obra seminal fue revolucionaria porque unificó bajo un mismo paraguas conceptual y computacional diversas clases de modelos que hasta entonces se trataban por separado: la regresión lineal para datos normales, la regresión logística para datos binarios y la regresión de Poisson para datos de conteo. Esto estimuló enormemente el desarrollo de software y la aplicación del modelado estadístico a una nueva y vasta gama de problemas.

1.4.4 La evolución contemporánea

Este legado continúa evolucionando a un ritmo vertiginoso, con la inclusión de modelos jerárquicos y bayesianos, métodos no paramétricos y de *machine learning* como los árboles de regresión, y la adaptación de la regresión al análisis de datos masivos (*big data*). La regresión ha evolucionado desde una observación sobre la herencia biológica hasta convertirse en una de las herramientas más versátiles y poderosas del arsenal analítico moderno.

2 El modelo de regresión lineal simple

La regresión lineal constituye uno de los pilares fundamentales de la modelización estadística. Es, a menudo, el primer y más importante modelo predictivo que se aprende, no solo por su simplicidad e interpretabilidad, sino porque los conceptos que exploraremos aquí son la base sobre la que se construyen técnicas mucho más avanzadas, como el **modelo de regresión lineal múltiple**, los **modelos lineales generalizados (GLM)** o incluso conceptos utilizados en algoritmos de *machine learning* (Draper 1998; Kutner et al. 2005; James et al. 2021).

En este capítulo, daremos el primer y más crucial paso en nuestro viaje por el modelado predictivo: el estudio del **modelo de regresión lineal simple**. Para ello, seguiremos el ciclo de vida completo de un proyecto de modelado: comenzaremos con la exploración visual y cuantitativa de los datos, formalizaremos después nuestras observaciones mediante el lenguaje matemático del modelo y sus supuestos, aprenderemos a estimar sus parámetros, realizaremos inferencias sobre ellos y, finalmente, diagnosticaremos la validez de nuestro modelo (Fox and Weisberg 2018; Harrell 2015).

La comprensión profunda que desarrollaremos aquí es esencial, ya que los principios de estimación, inferencia y diagnóstico que aprenderemos son directamente escalables al **modelo de regresión lineal múltiple**, que exploraremos en el siguiente capítulo.

! Objetivos de aprendizaje

Al finalizar este capítulo, serás capaz de:

1. **Comprender y aplicar** el proceso de modelización estadística para un problema con una única variable predictora.
2. **Identificar y medir la correlación lineal** entre dos variables como paso previo al modelado.
3. **Describir la formulación matemática** del modelo de regresión lineal simple e interpretar el significado práctico de sus parámetros.
4. **Estimar los coeficientes** del modelo mediante el método de mínimos cuadrados ordinarios (MCO) y entender su derivación matemática y propiedades.
5. **Realizar inferencias sobre los parámetros** del modelo y evaluar su bondad de ajuste mediante el análisis de la varianza y el coeficiente de determinación R^2.
6. **Diagnosticar la adecuación del modelo**, evaluando visual y analíticamente si se cumplen los supuestos del modelo lineal.

2.1 Exploración inicial: visualización y cuantificación de la relación

Antes de sumergirnos en la teoría de la regresión, debemos hacer lo que todo buen analista hace primero: **observar y cuantificar la relación en los datos**. Este paso exploratorio es fundamental para formular hipótesis y justificar la elección de un modelo lineal.

2.1.1 Visualización: el gráfico de dispersión

La herramienta más potente para examinar la relación entre dos variables continuas es el **gráfico de dispersión** (*scatterplot*). Nos permite intuir visualmente la **forma**, la **dirección** y la **fuerza** de la relación. Una inspección visual es siempre el punto de partida.

2.1.2 Cuantificación de la asociación: covarianza y correlación

Una vez que la visualización sugiere una tendencia, necesitamos métricas para cuantificarla.

2.1.2.1 Covarianza

La **covarianza** es una medida de la variabilidad conjunta de dos variables aleatorias, X e Y. Nos indica la dirección de la relación lineal. La covarianza muestral, calculada a partir de nuestras observaciones (x_i, y_i), es:

$$\text{Cov}(x, y) = s_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

El principal inconveniente de la covarianza es que su magnitud depende de las unidades de las variables, lo que la hace difícil de interpretar.

2.1.2.2 Coeficiente de correlación de Pearson

Para solucionar el problema de la escala, estandarizamos la covarianza, dividiéndola por el producto de las desviaciones típicas de cada variable. El resultado es el **coeficiente de correlación de Pearson** (r):

$$r = r_{xy} = \frac{s_{xy}}{s_x s_y}$$

Este coeficiente es **adimensional** y siempre varía entre **-1 y 1**, lo que permite una interpretación universal de la fuerza de la asociación *lineal*.

Ejemplo práctico: Horas de estudio vs. Calificaciones

Vamos a plantear un problema que nos acompañará durante todo el capítulo: queremos saber si el tiempo de estudio semanal influye en las calificaciones finales.

```
library(ggplot2)
set.seed(123) # Para reproducibilidad

# Simulación de datos
datos <- data.frame(
  Tiempo_Estudio = round(runif(100, min = 5, max = 40), 1)
)
datos$Calificaciones <- round(5 + 0.1 * datos$Tiempo_Estudio + rnorm(100, mean = 0, sd = 0.

# Visualización
ggplot(datos, aes(x = Tiempo_Estudio, y = Calificaciones)) +
  geom_point(color = "#0072B2", alpha = 0.7) +
  labs(
    title = "Relación entre Tiempo de Estudio y Calificaciones",
    x = "Tiempo de Estudio (horas/semana)",
    y = "Calificaciones (promedio)"
  ) +
  theme_classic(base_size = 14)

# Cuantificación (los objetos se guardan para usarlos en el texto)
covarianza <- cov(datos$Tiempo_Estudio, datos$Calificaciones)
correlacion <- cor(datos$Tiempo_Estudio, datos$Calificaciones)
```

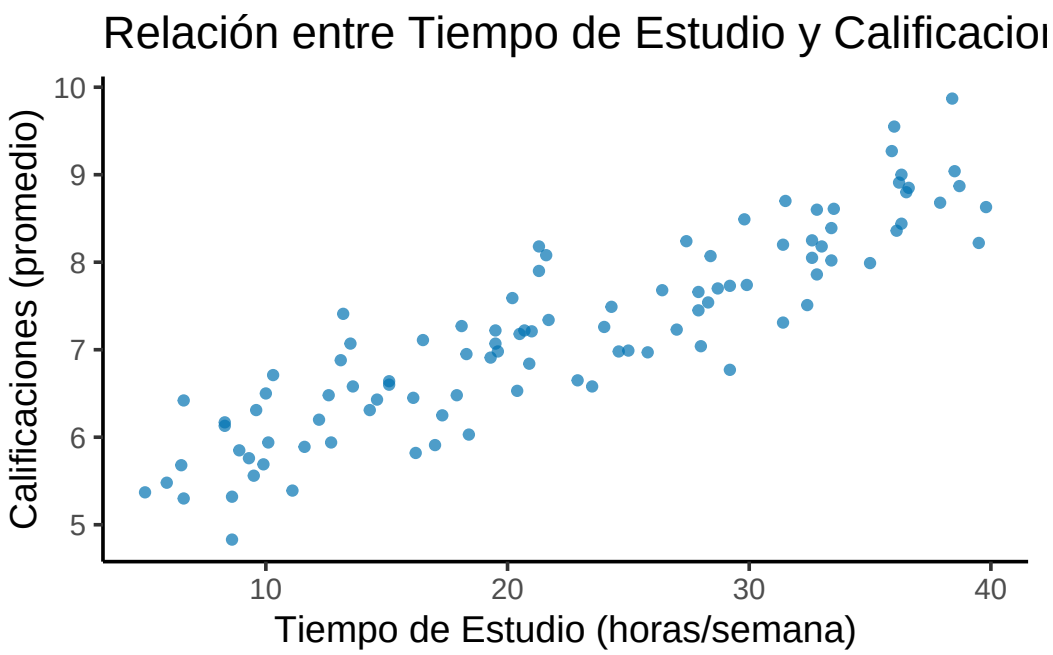


Figure 2.1: Relación entre tiempo de estudio y calificaciones.

El gráfico muestra una clara tendencia lineal positiva. La covarianza toma un valor de 9.82, y el coeficiente de correlación de Pearson es de 0.9. Ambos valores confirman que la asociación lineal es, además de positiva, muy fuerte. Esta evidencia visual y numérica nos da una base sólida para proponer un modelo de regresión lineal.

¡Correlación no implica causalidad!

El haber encontrado una fuerte correlación positiva entre el tiempo de estudio y las calificaciones (0.9) no nos autoriza a concluir que *una cosa causa la otra*. La regresión lineal puede demostrar que las variables se mueven juntas y nos permite predecir una a partir de la otra, pero no explica el porqué de la relación.
Podría existir una tercera variable oculta (p. ej., el interés del alumno en la materia) que influya tanto en las horas de estudio como en las calificaciones. Establecer causalidad requiere un diseño experimental riguroso (asignando aleatoriamente a los estudiantes a diferentes tiempos de estudio), no solo un análisis observacional.

2.2 Formulación teórica del modelo

Una vez que la exploración sugiere una relación lineal, el siguiente paso es formalizarla matemáticamente. Aquí es donde definimos la estructura teórica del modelo y los supuestos bajo los cuales operará.

2.2.1 El modelo poblacional y sus componentes

El **modelo poblacional** postula que la relación verdadera entre la variable respuesta Y y la predictora X sigue una línea recta, aunque contaminada por cierta aleatoriedad. Para cualquier individuo i de la población, esta relación se describe como:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

En esta ecuación, β_0 y β_1 son los **parámetros poblacionales** (el intercepto y la pendiente verdaderos pero desconocidos), y ε_i es el **error aleatorio**, un componente fundamental que captura todas las fuentes de variabilidad que el modelo no puede explicar por sí solo. Específicamente, este término incluye:

- **Variables omitidas:** Factores que también afectan a las calificaciones (como la calidad del sueño, la motivación del estudiante o su conocimiento previo) y que no están en el modelo.
- **Error de medida:** Pequeñas imprecisiones al medir las variables (p. ej., un estudiante podría reportar 20 horas de estudio cuando en realidad fueron 19.5).
- **Aleatoriedad inherente:** La variabilidad puramente estocástica o impredecible en el comportamiento humano.

Como nunca observamos la población entera, nuestro trabajo consiste en usar una muestra para estimar el **modelo muestral**:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

Aquí, los "gorros" ($\hat{\cdot}$) denotan **estimaciones** calculadas a partir de la muestra. La diferencia entre el valor real y el predicho, $e_i = y_i - \hat{y}_i$, se conoce como **residuo**.

2.2.2 Los supuestos del modelo lineal clásico (Gauss-Markov)

Para que el puente entre nuestro modelo muestral y la realidad poblacional sea sólido, debemos asumir que los errores teóricos ε_i se comportan de una manera predecible y ordenada. Estos supuestos, conocidos como condiciones de Gauss-Markov (Kutner et al. 2005; Weisberg 2005), son fundamentales para las propiedades óptimas de los estimadores de mínimos cuadrados.

1. **Linealidad**: La relación entre X y el valor esperado de Y es, en promedio, una línea recta: $E[Y_i|X_i] = \beta_0 + \beta_1 X_i$.
2. **Independencia de los errores**: El error de una observación no está correlacionado con el error de ninguna otra: $\text{Cov}(\varepsilon_i, \varepsilon_j) = 0$ para $i \neq j$.
3. **Homocedasticidad**: La varianza del error es constante (σ^2) para todos los valores de X: $Var(\varepsilon_i|X_i) = \sigma^2$. Esto significa que la dispersión de los datos alrededor de la línea de regresión es la misma a lo largo de todos los valores de la variable predictora. La violación de este supuesto se conoce como **heterocedasticidad**, donde la dispersión de los errores cambia (p. ej., aumenta a medida que X crece).

Cuando el objetivo no es sólo estimar la recta, sino inferir con ella, entonces se asume una hipótesis más: la normalidad de la variable respuesta, o lo que es lo mismo, del error aleatorio:

4. **Normalidad de los errores**: Para la inferencia, se asume que los errores siguen una distribución Normal con media cero y varianza σ^2: $\varepsilon_i \sim N(0, \sigma^2)$.

Estos supuestos son esenciales para garantizar la validez de las estimaciones y conclusiones derivadas del modelo.

2.3 Estimación de los parámetros

Necesitamos un método para encontrar la "mejor" recta de ajuste. El **Método de Mínimos Cuadrados Ordinarios (MCO/OLS)** nos proporciona este criterio.

2.3.1 El criterio de mínimos cuadrados

MCO busca la recta que minimice la **Suma de los Cuadrados del Error (SSE)**, es decir, la suma de las distancias verticales al cuadrado entre los puntos observados y la recta de regresión:

$$\text{SSE}(\beta_0, \beta_1) = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \widehat{y})^2 = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2$$

2.3.2 Derivación matemática de los estimadores

Para encontrar los valores de β_0 y β_1 que minimizan esta función, recurrimos al cálculo. Tratamos la SSE como una función de dos variables y calculamos sus derivadas parciales, igualándolas a cero para encontrar el mínimo.

$$\frac{\partial \text{SSE}}{\partial \beta_0} = -2 \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\frac{\partial \text{SSE}}{\partial \beta_1} = -2 \sum_{i=1}^{n} x_i (y_i - \beta_0 - \beta_1 x_i) = 0$$

La resolución de este sistema de dos ecuaciones (conocidas como las **ecuaciones normales**) nos proporciona las fórmulas para los estimadores de MCO:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} = \frac{s_{xy}}{s_{xx}}$$
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

2.3.2.1 Interpretación práctica de los coeficientes

Una vez estimados, los coeficientes tienen una interpretación muy concreta y útil:

- **Pendiente ($\hat{\beta}_1$)**: Representa el cambio promedio estimado en la variable respuesta Y por cada **aumento de una unidad** en la variable predictora X. En nuestro ejemplo, sería el número de puntos que se espera que aumente la calificación final por cada hora adicional de estudio semanal.
- **Intercepto ($\hat{\beta}_0$)**: Es el valor promedio estimado de la variable respuesta Y cuando la variable predictora X es igual a cero. La interpretación del intercepto solo tiene sentido práctico si $X = 0$ es un valor plausible y se encuentra dentro del rango de nuestros datos. De lo contrario (como en nuestro ejemplo, donde nadie estudia 0 horas), a menudo se considera simplemente un ancla matemática para la recta de regresión.

i Minimización de SSE

La obtención de los estimadores de mínimos cuadrados para la regresión lineal simple se basa en minimizar la suma de los cuadrados de los residuos (SSE). Este método, desarrollado por Legendre y Gauss a principios del siglo XIX (Galton 1886; Weisberg 2005), es fundamental en la estadística moderna. Aquí está el proceso paso a paso:
Para minimizar SSE, derivamos parcialmente con respecto a β_0 y β_1 y resolvemos el

sistema de ecuaciones.

1. **Primera derivada con respecto a** β_0:

$$\frac{\partial SSE}{\partial \beta_0} = -2 \sum_{i=1}^{n} \left(y_i - (\beta_0 + \beta_1 x_i) \right).$$

`Igualando a cero:`

$$\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) = 0.$$

`Reordenando:`

$$n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i. \tag{1}$$

2. **Primera derivada con respecto a** β_1:

$$\frac{\partial SSE}{\partial \beta_1} = -2 \sum_{i=1}^{n} x_i \left(y_i - (\beta_0 + \beta_1 x_i) \right).$$

`Igualando a cero:`

$$\sum_{i=1}^{n} x_i \left(y_i - \beta_0 - \beta_1 x_i \right) = 0.$$

`Reordenando:`

$$\beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i. \tag{2}$$

Resolución del Sistema de Ecuaciones
El sistema está dado por las ecuaciones (1) y (2):

1. $n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$.
2. $\beta_0 \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i$.

Resolviendo para β_0 y β_1:

1. De la primera ecuación, despejamos β_0:

$$\beta_0 = \frac{\sum_{i=1}^{n} y_i - \beta_1 \sum_{i=1}^{n} x_i}{n}. \tag{3}$$

2. Sustituimos β_0 en la segunda ecuación:

$$\frac{\sum_{i=1}^{n} y_i - \beta_1 \sum_{i=1}^{n} x_i}{n} \sum_{i=1}^{n} x_i + \beta_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i.$$

   Simplificando:

$$\beta_1 \left(\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n} \right) = \sum_{i=1}^{n} x_i y_i - \frac{\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n}.$$

3. Expresamos β_1:

$$\beta_1 = \frac{\sum_{i=1}^{n} x_i y_i - \frac{\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n}}{\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n}}.$$

   Esta es la fórmula para β_1, que puede reescribirse como:

$$\beta_1 = \frac{\mathrm{Cov}(x, y)}{\mathrm{Var}(x)},$$

   donde $\mathrm{Cov}(x, y)$ y $\mathrm{Var}(x)$ son la covarianza y la varianza muestral de x y y.

4. Finalmente, sustituimos β_1 en la ecuación (3) para obtener β_0:

$$\beta_0 = \bar{y} - \beta_1 \bar{x},$$

   donde $\bar{x}$ y $\bar{y}$ son las medias de x y y.

Bajo los supuestos del modelo, el **Teorema de Gauss-Markov** demuestra que estos estimadores son los **Mejores Estimadores Lineales Insesgados (MELI / BLUE)**.

2.4 Inferencia y bondad de ajuste

Una vez hemos estimado los parámetros del modelo, nuestro trabajo apenas ha comenzado. Ahora debemos pasar de la descripción a la inferencia. Necesitamos un conjunto de herramientas que nos permitan responder a preguntas cruciales: ¿Son nuestros coeficientes estimados,

$\hat{\beta}_0$ y $\hat{\beta}_1$, meras casualidades de nuestra muestra o reflejan una relación real en la población? ¿Qué tan bueno es nuestro modelo para explicar la variabilidad de la variable respuesta? Esta sección se dedica a responder estas preguntas.

2.4.1 Propiedades de los estimadores de MCO

Antes de realizar inferencias, es fundamental entender las propiedades teóricas de los estimadores que hemos calculado.

- **Insesgadez**: Los estimadores de MCO son insesgados. Esto significa que si pudiéramos repetir nuestro muestreo muchísimas veces y calcular los estimadores en cada muestra, el promedio de todas nuestras estimaciones de $\hat{\beta}_0$ y $\hat{\beta}_1$ convergería a los verdaderos valores poblacionales β_0 y β_1. Matemáticamente:

$$E[\hat{\beta}_0] = \beta_0 \quad \text{y} \quad E[\hat{\beta}_1] = \beta_1$$

- **Varianza de los estimadores**: Las fórmulas para la varianza de nuestros estimadores cuantifican su precisión. Una varianza pequeña implica que el estimador es más estable a través de diferentes muestras.

$$Var(\hat{\beta}_1) = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sigma^2}{S_{xx}}$$

$$Var(\hat{\beta}_0) = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}\right]$$

  Donde σ^2 es la varianza (desconocida) del término de error ε.

- **Teorema de Gauss-Markov**: Este es uno de los resultados más importantes de la teoría de la regresión. Establece que, bajo los supuestos de linealidad, independencia y homocedasticidad (no se requiere normalidad), los estimadores de MCO son los **Mejores Estimadores Lineales Insesgados** (MELI, o BLUE en inglés). Esto significa que, de entre toda la clase de estimadores que son lineales e insesgados, los de MCO son los que tienen la menor varianza posible.

> **i** Propiedades adicionales para las predicciones y para los residuos
>
> - La suma de los residuos es cero:
>
> $$\sum_{i=1}^{n} e_i = \sum_{i=1}^{n}(y_i - \hat{y}_i) = 0$$

- La suma de los valores observados es igual a la suma de los valores ajustados:

$$\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \hat{y}_i$$

- La suma de los residuos ponderados por los regresores es cero:

$$\sum_{i=1}^{n} x_i e_i = 0$$

- La suma de los residuos ponderados por las predicciones es cero:

$$\sum_{i=1}^{n} \hat{y}_i e_i = 0$$

- La recta de regresión contiene el punto $(\bar{x}, \bar{y})$:

Ejemplo

Para los datos de calificaciones y tiempo de estudio, estos son los estimadores de los parámetros del modelo de regresión:

```
# 1. Ajustamos el modelo lineal
modelo_estudio <- lm(Calificaciones ~ Tiempo_Estudio, data = datos)

# 2. Obtenemos el resumen completo del modelo
summary(modelo_estudio)
```

```
Call:
lm(formula = Calificaciones ~ Tiempo_Estudio, data = datos)

Residuals:
     Min       1Q   Median       3Q      Max
-1.11465 -0.30262 -0.00942  0.29509  1.10533

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     5.00118    0.11977   41.76   <2e-16 ***
Tiempo_Estudio  0.09875    0.00488   20.23   <2e-16 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4842 on 98 degrees of freedom
Multiple R-squared:  0.8069,    Adjusted R-squared:  0.8049
F-statistic: 409.5 on 1 and 98 DF,  p-value: < 2.2e-16
```

2.4.2 Estimación de la varianza del error

Las fórmulas de la varianza de los estimadores dependen de σ^2, la varianza del error poblacional, que es desconocida. Por lo tanto, necesitamos estimarla a partir de nuestros datos. Un estimador insesgado de σ^2 es la **Media Cuadrática del Error (MSE)**:

$$\hat{\sigma}^2 = \text{MSE} = \frac{\text{SSE}}{n-2} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}$$

Dividimos por $n-2$, los **grados de libertad del error**, porque hemos "gastado" dos grados de libertad de nuestros datos para estimar los dos parámetros, β_0 y β_1. La raíz cuadrada de la MSE, $\hat{\sigma}$, se conoce como el **error estándar de los residuos** y es una medida de la dispersión promedio de los puntos alrededor de la recta de regresión.

2.4.2.1 El error estándar de los residuos y el RMSE

La raíz cuadrada de la MSE, $\hat{\sigma}$, se conoce formalmente como el **error estándar de los residuos** (*Residual Standard Error*). Este valor es nuestra estimación de la desviación estándar del error poblacional, σ, y es una medida de la dispersión promedio de los puntos alrededor de la recta de regresión.

$$\hat{\sigma} = \sqrt{\text{MSE}}$$

En el campo del modelado predictivo y el *machine learning*, esta misma cantidad se conoce como la **Raíz del Error Cuadrático Medio** o **RMSE** (*Root Mean Squared Error*). Aunque la fórmula es idéntica, la interpretación del RMSE se centra en la **evaluación del rendimiento predictivo** del modelo. El RMSE nos dice, en promedio, cuál es la magnitud del error de predicción de nuestro modelo, y tiene la ventaja de estar en las **mismas unidades** que la variable respuesta Y. Por ejemplo, si estamos prediciendo precios de viviendas en euros, un RMSE de 5000 significa que nuestras predicciones se desvían, en promedio, unos 5000 € de los precios reales.

2.4.3 Análisis de la Varianza (ANOVA) para la significancia de la regresión

Una vez hemos estimado los coeficientes, necesitamos una prueba formal para determinar si el modelo en su conjunto es útil. Es decir, ¿la variable predictora X explica una porción de la variabilidad de la variable respuesta Y que sea estadísticamente significativa, o la relación que observamos podría deberse simplemente al azar? El **Análisis de la Varianza (ANOVA)** nos proporciona la herramienta para responder a esta pregunta a través del **contraste F de significancia global**.

Las hipótesis de este contraste son:

- $H_0 : \beta_1 = 0$: La hipótesis nula postula que no existe una relación lineal entre X e Y. El modelo no tiene poder explicativo y no es mejor que usar simplemente la media, $\bar{y}$, como predicción para cualquier valor de x.
- $H_1 : \beta_1 \neq 0$: La hipótesis alternativa sostiene que sí existe una relación lineal significativa.

> **Repaso**
>
> Es conveniente repasar el tema de *Análisis de la Varianza* estudiado en la asignatura de Inferencia, ya que los conceptos son directamente aplicables aquí.

La idea fundamental del ANOVA es comparar la variabilidad que nuestro modelo *explica* con la variabilidad que *no puede explicar* (el error residual). Para ello, se descompone la variabilidad total de nuestras observaciones (y_i) en dos partes ortogonales.

1. La **Suma Total de Cuadrados (SST)** mide la variabilidad total de los datos alrededor de su media. Es nuestra referencia base de la dispersión total que hay que explicar.

$$\text{SST} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

2. Esta variabilidad se descompone en:

   - **Suma de Cuadrados de la Regresión (SSR)**: Mide la parte de la variabilidad total que es explicada por nuestro modelo. Cuantifica cuánto se desvían las predicciones del modelo ($\hat{y}_i$) de la media general ($\bar{y}$).

$$\text{SSR} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

   - **Suma de Cuadrados del Error (SSE)**: Mide la variabilidad residual, es decir, la parte que el modelo no puede capturar. Cuantifica la dispersión de los puntos

reales (y_i) alrededor de la recta de regresión ($\hat{y}_i$).

$$\text{SSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

La descomposición fundamental de la varianza es, por tanto: $\text{SST} = \text{SSR} + \text{SSE}$.

Para poder comparar estas sumas de cuadrados de forma justa, las estandarizamos dividiéndolas por sus respectivos **grados de libertad**, obteniendo así las **Medias Cuadráticas** (MS):

$$\text{MSR} = \frac{\text{SSR}}{1} \qquad \text{MSE} = \frac{\text{SSE}}{n-2}$$

Finalmente, el **estadístico F** se construye como el cociente entre la variabilidad explicada por el modelo y la variabilidad no explicada:

$$F = \frac{\text{MSR}}{\text{MSE}}$$

Intuitivamente, el estadístico F actúa como una **ratio de señal a ruido**. La MSR (la "señal") representa la variabilidad que nuestro modelo captura sistemáticamente, mientras que la MSE (el "ruido") representa la variabilidad aleatoria o residual. Un valor de F grande nos dice que la señal es mucho más fuerte que el ruido, lo que apoya la hipótesis de que la relación que hemos modelado es real y no fruto del azar.

Toda esta información se organiza de forma estándar en la **tabla ANOVA**:

Fuente	df	SS	$MS = SS/df$	Estadístico F
Regresión	1	SSR	MSR	$F = MSR/MSE$
Error	$n-2$	SSE	MSE	
Total	$n-1$	SST		

Bajo la hipótesis nula ($H_0 : \beta_1 = 0$), el estadístico F sigue una distribución F con 1 y $n-2$ grados de libertad. Si el p-valor asociado a nuestro estadístico F es suficientemente pequeño ($p < \alpha$), rechazamos H_0 y concluimos que nuestro modelo tiene un poder explicativo estadísticamente significativo.

2.4.4 Bondad del ajuste: coeficiente de determinación

El coeficiente de determinación (R^2) es una medida clave que cuantifica qué proporción de la variabilidad total observada en la muestra (y_i) es explicada por la relación lineal con X a través del modelo. Su fórmula se deriva de la descomposición de la varianza:

$$R^2 = \frac{\text{SSR}}{\text{SST}} = 1 - \frac{\text{SSE}}{\text{SST}}$$

Donde las sumas de cuadrados se calculan a partir de los datos muestrales:

- $\text{SST} = \sum_{i=1}^{n}(y_i - \bar{y})^2$: Suma Total de Cuadrados, mide la variabilidad total de las observaciones.
- $\text{SSR} = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$: Suma de Cuadrados de la Regresión, mide la variabilidad explicada por el modelo.
- $\text{SSE} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$: Suma de Cuadrados del Error, mide la variabilidad no explicada (residual).

Un R^2 cercano a 1 indica que el modelo ajusta bien los datos, mientras que un R^2 cercano a 0 indica un ajuste pobre.

Relación entre R^2 y el coeficiente de correlación

En el caso específico del modelo de regresión lineal simple, existe una relación directa y simple: el coeficiente de determinación R^2 es literalmente el cuadrado del coeficiente de correlación de Pearson (r) entre X e Y.

$$R^2 = (r_{xy})^2$$

Esto refuerza la idea de que ambos miden la fuerza de la asociación *lineal*, aunque R^2 lo hace desde la perspectiva de la varianza explicada por el modelo.

Interpretación de R^2

El coeficiente de determinación, R^2, es una métrica muy popular, pero su interpretación requiere cautela. Un valor alto no garantiza un buen modelo, y un valor bajo no siempre implica un modelo inútil. Es fundamental tener en cuenta las siguientes observaciones:

- R^2 no mide la linealidad de la relación. Un modelo puede tener un R^2 muy alto incluso si la relación subyacente entre las variables X e Y no es lineal. Por ello, un R^2 elevado nunca debe sustituir a un análisis gráfico de los residuos para verificar el supuesto de linealidad.
- R^2 es sensible al rango de la variable predictora X. Si el modelo de regresión es

adecuado, la magnitud de R^2 aumentará si aumenta la dispersión de las observaciones x_i (es decir, si S_{xx} crece). Esto se debe a que un mayor rango en X tiende a aumentar la Suma Total de Cuadrados (SST), lo que puede inflar el valor de R^2 sin que la precisión del modelo (medida por la MSE) haya mejorado.

- **Un rango restringido en** X puede producir un R^2 artificialmente bajo. Como consecuencia del punto anterior, si los datos se han recogido en un rango muy estrecho de la variable X, el R^2 puede ser muy pequeño, aunque exista una relación fuerte y significativa entre las variables. Esto podría llevar a la conclusión errónea de que el predictor no es útil.

2.4.5 Inferencia sobre los coeficientes

Además de la prueba F global, podemos realizar inferencias sobre cada parámetro individualmente. Para ello, necesitamos el supuesto de normalidad de los errores.

2.4.5.1 Distribución de los estimadores

Bajo el supuesto de normalidad, se puede demostrar que los estimadores también siguen una distribución Normal:

$$\hat{\beta}_1 \sim N\left(\beta_1, \frac{\sigma^2}{S_{xx}}\right) \qquad \hat{\beta}_0 \sim N\left(\beta_0, \sigma^2\left[\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}\right]\right)$$

Al estandarizar y reemplazar la desconocida σ^2 por su estimador $\hat{\sigma}^2 = \text{MSE}$, obtenemos un estadístico que sigue una distribución t-Student con $n-2$ grados de libertad:

$$t = \frac{\hat{\beta}_1 - \beta_1}{\text{SE}(\hat{\beta}_1)} \sim t_{n-2}$$

donde $\text{SE}(\hat{\beta}_1) = \sqrt{\frac{\text{MSE}}{S_{xx}}}$ es el **error estándar** del estimador $\hat{\beta}_1$.

2.4.5.2 Contraste de hipótesis para la pendiente

El contraste más común es el de la significancia de la pendiente: * $H_0 : \beta_1 = 0$ * $H_1 : \beta_1 \neq 0$

Bajo H_0, el estadístico de contraste es:

$$t_0 = \frac{\hat{\beta}_1}{\text{SE}(\hat{\beta}_1)}$$

Rechazamos H_0 si $|t_0| > t_{\alpha/2, n-2}$ o, equivalentemente, si el p-valor asociado es menor que α.

Relación entre el contraste F y el contraste t

En el contexto de la **regresión lineal simple** (y solo en este caso), el contraste F para la significancia global del modelo es matemáticamente equivalente al contraste t para la significancia del coeficiente β_1. Se puede demostrar que $F = t^2$, y el p-valor de ambos contrastes será idéntico.

2.4.5.3 Intervalo de confianza para la pendiente

A partir de la distribución t, podemos construir un intervalo de confianza al $100(1-\alpha)\%$ para el verdadero valor de la pendiente β_1:

$$\hat{\beta}_1 \pm t_{\alpha/2,n-2} \cdot \text{SE}(\hat{\beta}_1)$$

Este intervalo nos da un rango de valores plausibles para el efecto de X sobre Y. Si el intervalo no contiene el cero, es equivalente a rechazar la hipótesis nula $H_0 : \beta_1 = 0$.

Para recordar

En los programas estadísticos se suele proporcionar el p-valor del contraste. Puedes repasar el significado de p-valor proporcionado en la asignatura de Inferencia.

Ejemplo: Interpretación del `summary`

La función `summary()` en R nos proporciona toda esta información.

```
summary(modelo_estudio)
```

```
Call:
lm(formula = Calificaciones ~ Tiempo_Estudio, data = datos)

Residuals:
     Min       1Q   Median       3Q      Max
-1.11465 -0.30262 -0.00942  0.29509  1.10533

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     5.00118    0.11977   41.76   <2e-16 ***
Tiempo_Estudio  0.09875    0.00488   20.23   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.4842 on 98 degrees of freedom
Multiple R-squared:  0.8069,    Adjusted R-squared:  0.8049
F-statistic: 409.5 on 1 and 98 DF,  p-value: < 2.2e-16
```

Interpretación:

- **Coefficients**: El p-valor para `Tiempo_Estudio` (<0.001) es muy pequeño, por lo que rechazamos H_0 y concluimos que la variable es un predictor significativo.
- **R-squared**: El valor de R^2 (0.81) nos indica que el 81% de la variabilidad en las calificaciones es explicada por el tiempo de estudio.
- **F-statistic**: El p-valor del estadístico F

  (98) confirma que el modelo en su conjunto es estadísticamente significativo.

2.5 Predicción de nuevas observaciones

Una vez que hemos ajustado y validado un modelo de regresión, uno de sus propósitos más importantes es utilizarlo para hacer predicciones. Sin embargo, es fundamental distinguir entre dos tipos de predicción:

1. **Estimar la respuesta media** para un valor dado de X. Por ejemplo: "¿Cuál es la calificación *promedio* que esperamos para todos los estudiantes que estudian 25 horas semanales?".
2. **Predecir una respuesta individual** para un valor dado de X. Por ejemplo: "Si un estudiante *concreto* estudia 25 horas semanales, ¿entre qué valores esperamos que se encuentre su calificación?".

Estos dos objetivos, aunque parecidos, responden a preguntas distintas y manejan diferentes fuentes de incertidumbre, lo que da lugar a dos tipos de intervalos.

2.5.1 Intervalo de confianza para la respuesta media

Este intervalo estima el valor esperado de Y para un valor concreto del regresor, x_0. Su objetivo es acotar dónde se encuentra la **línea de regresión poblacional verdadera** para ese punto x_0. La estimación puntual es $\hat{y}_0 = \hat{\beta}_0 + \hat{\beta}_1 x_0$.

El intervalo de confianza al $100(1-\alpha)\%$ para la respuesta media $E[Y|X = x_0]$ viene dado por:

$$\hat{y}_0 \pm t_{\alpha/2,n-2} \cdot \sqrt{\text{MSE}\left(\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}\right)}$$

La anchura de este intervalo depende de dos fuentes de error: la incertidumbre en la estimación de la recta y la distancia del punto x_0 a la media $\bar{x}$. El intervalo es más estrecho cerca del centro de los datos y más ancho en los extremos.

2.5.2 Intervalo de predicción para una respuesta individual

Este intervalo es el que debemos usar cuando queremos predecir el valor para **una única observación futura**, no para la media. Como indicas, este intervalo debe tener en cuenta dos fuentes de variabilidad:

1. La incertidumbre sobre la localización de la verdadera recta de regresión (la misma que en el intervalo de confianza).
2. La variabilidad inherente de una observación individual alrededor de la recta de regresión (el error aleatorio ε_i, cuya varianza estimamos con la MSE).

Por esta razón, el intervalo de predicción **siempre será más ancho** que el intervalo de confianza para la respuesta media. El intervalo de predicción al $100(1-\alpha)\%$ para una observación futura y_0 en el punto x_0 es:

$$\hat{y}_0 \pm t_{\alpha/2,n-2} \cdot \sqrt{\text{MSE}\left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}\right)}$$

La única diferencia matemática es el **"+1"** dentro de la raíz cuadrada, que representa la varianza σ^2 del error de una sola observación.

2.5.2.1 Predicción para la media de *m* observaciones futuras

Si se desea un intervalo de predicción para la media de m futuras observaciones en un valor x_0, la fórmula se modifica ligeramente. Este intervalo será más estrecho que el de una sola observación pero más ancho que el de la respuesta media:

$$\hat{y}_0 \pm t_{\alpha/2,n-2} \cdot \sqrt{\text{MSE}\left(\frac{1}{m} + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}\right)}$$

Ejemplo práctico: Predicción de calificaciones

Vamos a calcular y visualizar los intervalos para nuestro modelo de estudio. Usaremos la función `predict()` de R, que calcula estos intervalos de forma automática.

```r
# 1. Crear una secuencia de nuevos valores de X para predecir
nuevos_datos <- data.frame(
  Tiempo_Estudio = seq(min(datos$Tiempo_Estudio), max(datos$Tiempo_Estudio), length.out = 1
)

# 2. Calcular el intervalo de confianza para la RESPUESTA MEDIA
conf_interval <- predict(
  modelo_estudio,
  newdata = nuevos_datos,
  interval = "confidence",
  level = 0.95
)

# 3. Calcular el intervalo de predicción para una OBSERVACIÓN INDIVIDUAL
pred_interval <- predict(
  modelo_estudio,
  newdata = nuevos_datos,
  interval = "prediction",
  level = 0.95
)

# 4. Unir todo para graficar con ggplot2
plot_data <- cbind(nuevos_datos, as.data.frame(conf_interval), pred_pred = as.data.frame(pr
colnames(plot_data) <- c("Tiempo_Estudio", "fit_conf", "lwr_conf", "upr_conf", "fit_pred",

# 5. Visualización
ggplot() +
  # Capa 1: Puntos originales del dataframe 'datos'
  geom_point(data = datos, aes(x = Tiempo_Estudio, y = Calificaciones), color = "#0072B2",

  # Capa 2: Línea de regresión del dataframe 'plot_data'
  geom_line(data = plot_data, aes(x = Tiempo_Estudio, y = fit_conf), color = "black", linew

  # Capa 3: Banda de predicción (roja) del dataframe 'plot_data'
  geom_ribbon(data = plot_data, aes(x = Tiempo_Estudio, ymin = lwr_pred, ymax = upr_pred),

  # Capa 4: Banda de confianza (azul) del dataframe 'plot_data'
  geom_ribbon(data = plot_data, aes(x = Tiempo_Estudio, ymin = lwr_conf, ymax = upr_conf),

  # Etiquetas y tema
  labs(
    title = "Intervalos de Confianza y Predicción",
    x = "Tiempo de Estudio (horas/semana)",
    y = "Calificaciones (promedio)",
    caption = "La banda azul (más estrecha) es el IC del 95% para la media.\nLa banda roja
  ) +
  theme_classic(base_size = 14)
```

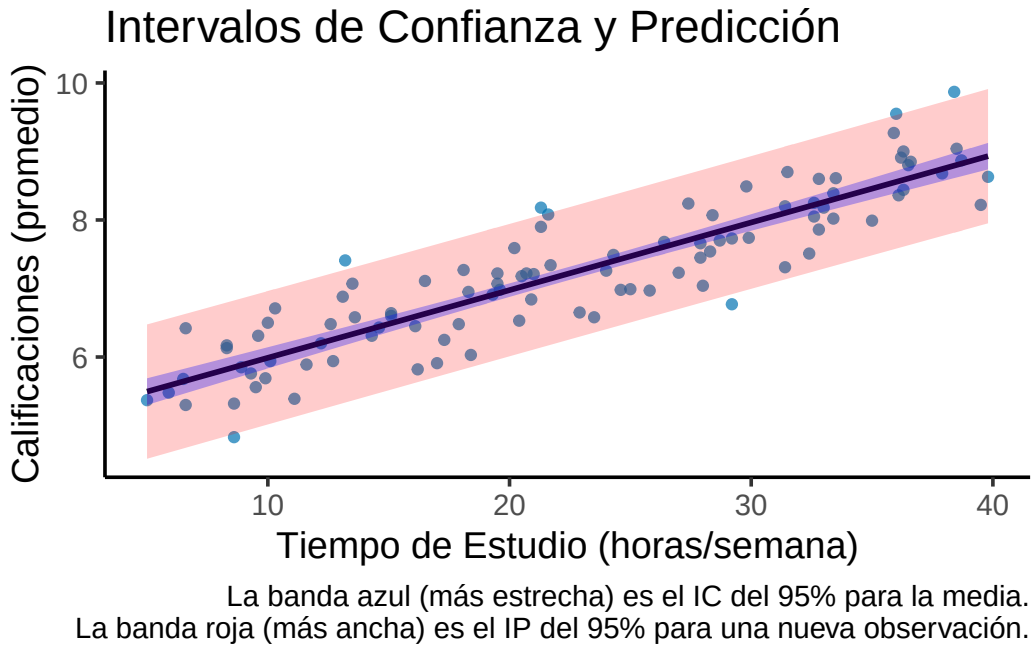


Figure 2.2: Comparación visual del intervalo de confianza (azul, más estrecho) y el intervalo de predicción (rojo, más ancho).

El gráfico muestra claramente que la incertidumbre al predecir una calificación individual es mucho mayor que la incertidumbre al estimar la calificación promedio. Ambas bandas se ensanchan al alejarse del centro de los datos.
Si quisiéramos una predicción para un estudiante que estudia 25 horas:

```
dato_nuevo <- data.frame(Tiempo_Estudio = 25)

# Guardamos la predicción para la media en un objeto
pred_media <- predict(modelo_estudio, newdata = dato_nuevo, interval = "confidence")

# Guardamos la predicción para un individuo en un objeto
pred_indiv <- predict(modelo_estudio, newdata = dato_nuevo, interval = "prediction")
```

Interpretación:

- Con un 95% de confianza, la calificación **promedio** de los estudiantes que estudian 25 horas está entre **7.37** y **7.57**.
- Con un 95% de confianza, la calificación de **un estudiante concreto** que estudia 25 horas estará entre **6.5** y **8.44**.

2.6 Diagnóstico del Modelo

Una vez que hemos ajustado un modelo y evaluado su significancia, el trabajo no ha terminado. Un paso crucial, a menudo subestimado, es el **diagnóstico del modelo** (Fox and Weisberg 2018; Harrell 2015). Este proceso consiste en verificar si se cumplen los supuestos del modelo de regresión lineal clásico. La fiabilidad de nuestras inferencias (los p-valores de los contrastes t y F, y los intervalos de confianza) depende directamente de la validez de estos supuestos.

El diagnóstico se realiza principalmente a través del **análisis de los residuos** del modelo ($e_i = y_i - \hat{y}_i$). Los residuos son nuestra mejor aproximación empírica de los errores teóricos no observables (ε_i). A continuación, se detalla cómo verificar cada uno de los supuestos clave.

2.6.1 Linealidad

Este supuesto establece que la relación entre la variable predictora X y el valor esperado de la variable respuesta Y es, en promedio, una línea recta: $E[Y|X] = \beta_0 + \beta_1 X$.

La herramienta fundamental para diagnosticar la linealidad es el gráfico de **residuos** (e_i) frente a los **valores ajustados** por el modelo ($\hat{y}_i$). La lógica de este gráfico es sencilla pero potente: si el modelo lineal es adecuado, los errores que comete (los residuos) deberían ser completamente aleatorios, sin guardar relación alguna con la magnitud de las predicciones. En esencia, buscamos confirmar que no queda ninguna información sistemática en los errores que el modelo no haya capturado.

En un escenario ideal, este gráfico debería parecer una nube de puntos distribuida horizontalmente y sin estructura aparente, centrada en la línea del cero. Esto nos indica que los errores son, en promedio, nulos para todos los niveles de predicción, cumpliendo así el supuesto de linealidad. La línea roja que R superpone en este gráfico, que suaviza la tendencia de los puntos, debería ser prácticamente plana y pegada al cero, confirmando la ausencia de patrones.

> Ejemplo de un modelo válido
>
> Para nuestro `modelo_estudio`, podemos generar específicamente el primer gráfico de diagnóstico, que es el de Residuos vs. Valores Ajustados.

```r
# Crear un dataframe con los datos para ggplot2
library(ggplot2)
library(broom)

# Extraer residuos y valores ajustados
datos_diagnostico <- data.frame(
  residuos = residuals(modelo_estudio),
  valores_ajustados = fitted(modelo_estudio)
)

# Gráfico de Residuos vs. Valores Ajustados con ggplot2
ggplot(datos_diagnostico, aes(x = valores_ajustados, y = residuos)) +
  geom_point(color = "#0072B2", alpha = 0.7) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  geom_smooth(method = "loess", se = FALSE, color = "red", linewidth = 0.8) +
  labs(
    title = "Residuos vs. Valores Ajustados",
    x = "Valores Ajustados",
    y = "Residuos"
  ) +
  theme_classic(base_size = 12)
```

```
`geom_smooth()` using formula = 'y ~ x'
```

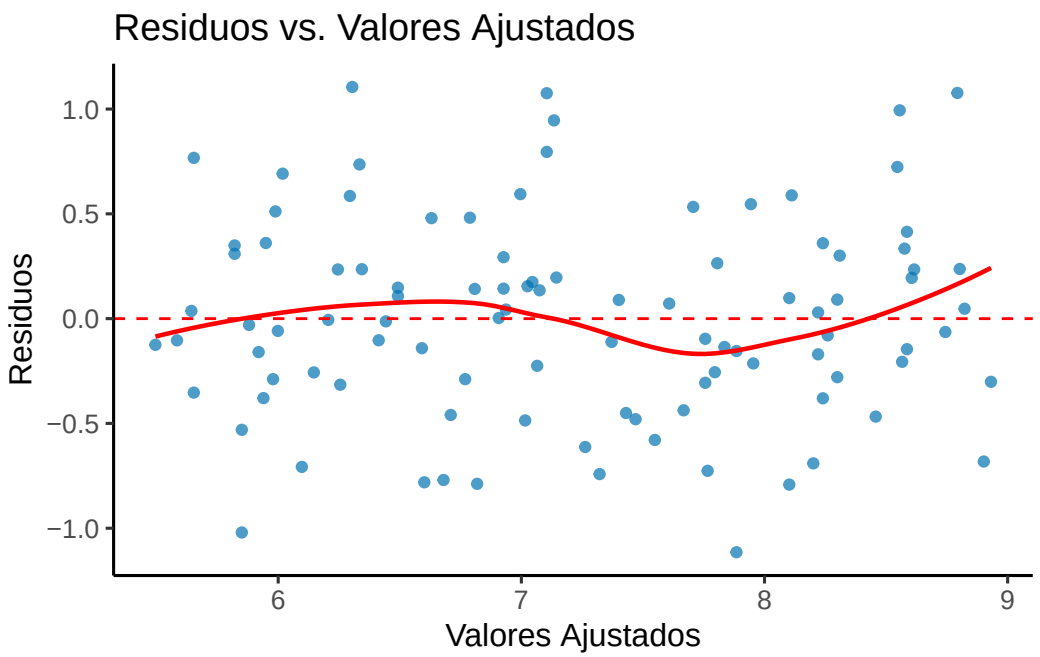


Figure 2.3: Gráfico de Residuos vs. Valores Ajustados para el modelo de estudio. No se observan patrones.

Como se puede observar, los puntos se distribuyen de forma aleatoria alrededor de la línea horizontal en cero. La línea roja, que suaviza la tendencia de los residuos, es prácticamente plana. Esto es un claro indicativo de que el supuesto de linealidad se cumple en nuestro modelo.

Por el contrario, la aparición de un **patrón sistemático** en los residuos es la señal de alarma de que algo anda mal. En lo que respecta al supuesto de **linealidad**, la evidencia más clara de una violación es una **tendencia curvilínea** (como una "U" o una parábola). Este patrón nos dice que el modelo es estructuralmente incapaz de capturar la forma de los datos y, por lo tanto, comete errores predecibles. Por ejemplo, puede subestimar la respuesta en los extremos (generando residuos positivos) y sobreestimarla en el centro (residuos negativos), lo que invalida el modelo lineal.

Contraejemplo: Violación del supuesto de linealidad

Ahora, vamos a simular a propósito unos datos que siguen una relación cuadrática (curva) y ajustaremos incorrectamente un modelo lineal para ver cómo se manifiesta el problema en el gráfico de diagnóstico.

```r
# 1. Simulación de datos no lineales
set.seed(42) # Nueva semilla para este ejemplo
x_no_lineal <- runif(100, 0, 10)
# La relación verdadera es cuadrática (y = 10 - (x-5)^2) más un error
y_no_lineal <- 10 - (x_no_lineal - 5)^2 + rnorm(100, 0, 4)
datos_no_lineal <- data.frame(x = x_no_lineal, y = y_no_lineal)

# 2. Ajuste de un modelo lineal (incorrecto)
modelo_no_lineal <- lm(y ~ x, data = datos_no_lineal)

# 3. Gráfico de Residuos vs. Valores Ajustados con ggplot2
datos_diag_no_lineal <- data.frame(
  residuos = residuals(modelo_no_lineal),
  valores_ajustados = fitted(modelo_no_lineal)
)

ggplot(datos_diag_no_lineal, aes(x = valores_ajustados, y = residuos)) +
  geom_point(color = "#D55E00", alpha = 0.7) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  geom_smooth(method = "loess", se = FALSE, color = "red", linewidth = 0.8) +
  labs(
    title = "Residuos vs. Valores Ajustados (Violación de Linealidad)",
    x = "Valores Ajustados",
    y = "Residuos"
  ) +
  theme_classic(base_size = 12)
```

```
`geom_smooth()` using formula = 'y ~ x'
```

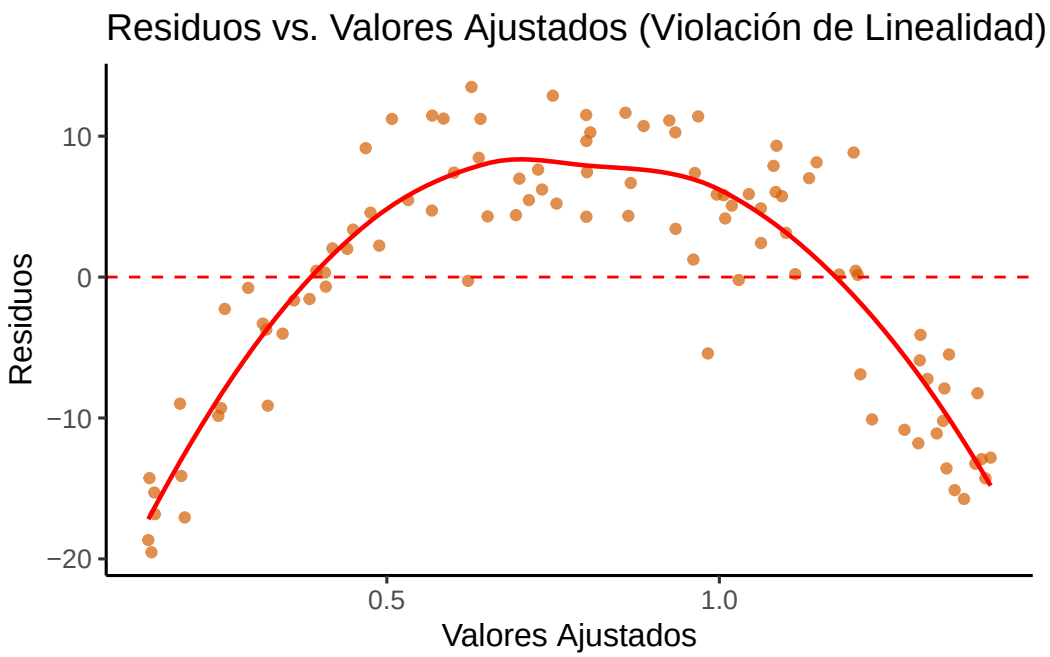


Figure 2.4: Patrón curvo evidente en los residuos, violando el supuesto de linealidad.

El gráfico de diagnóstico es inequívoco. A diferencia del ejemplo anterior, donde los puntos formaban una nube aleatoria, aquí los residuos dibujan un **patrón parabólico** perfecto (una "U" invertida). La línea roja de tendencia, en lugar de ser plana, sigue fielmente esta curva.

2.6.2 Homocedasticidad

El supuesto de homocedasticidad establece que la varianza de los errores del modelo debe ser constante para todos los niveles de la variable predictora. Es decir, la dispersión de los datos alrededor de la línea de regresión es la misma en todo su recorrido ($Var(\varepsilon_i|X_i) = \sigma^2$). La violación de este supuesto se conoce como **heteroscedasticidad**, y es un problema común en el modelado.

¿Por qué es tan importante? Si un modelo es heteroscedástico, los errores estándar de los coeficientes (β_0, β_1) estarán calculados de forma incorrecta. Como consecuencia, los intervalos de confianza y los contrastes de hipótesis (p-valores) no serán fiables, pudiendo llevarnos a conclusiones erróneas sobre la significancia de nuestras variables.

Sobre los residuos estandarizados

Los **residuos simples** ($e_i = y_i - \hat{y}_i$) no son directamente comparables entre sí porque tienen diferentes varianzas dependiendo de su apalancamiento (*leverage*). Por eso, en los gráficos de diagnóstico se utilizan **residuos estandarizados** o, mejor aún, **residuos estudentizados**, que ponen todos los residuos en una escala común. Esto facilita la identificación de patrones y valores atípicos. La explicación detallada de estos conceptos se encuentra en la sección de identificación de observaciones influyentes.

La heteroscedasticidad se detecta principalmente buscando patrones en la dispersión de los residuos.

- **Gráfico de Residuos vs. Valores Ajustados:** Como en la prueba de linealidad, este gráfico es nuestra primera herramienta. Aquí no buscamos patrones en la media de los residuos (que debe ser cero), sino en su **dispersión**. La señal de alarma inequívoca de heteroscedasticidad es una **forma de embudo o megáfono**, donde la dispersión de los residuos aumenta o disminuye a medida que cambian los valores ajustados.
- **Gráfico Scale-Location:** Este gráfico está diseñado específicamente para detectar heteroscedasticidad. Muestra la raíz cuadrada de los residuos estandarizados en el eje Y (`sqrt(|Standardized residuals|)`) frente a los valores ajustados en el eje X. Al usar la raíz cuadrada, se suaviza la distribución de los residuos, haciendo los patrones de varianza más fáciles de ver. Si la varianza es constante (homocedasticidad), deberíamos ver una nube de puntos aleatoria con una línea de tendencia roja aproximadamente plana. Una pendiente en esta línea roja indica que la varianza cambia con el nivel de la respuesta.
- **Prueba de Breusch-Pagan:** Es el contraste de hipótesis formal. Su lógica es ingeniosa: realiza una regresión auxiliar donde intenta predecir los residuos al cuadrado a partir de las variables predictoras originales. Si las variables predictoras ayudan a explicar la magnitud de los residuos al cuadrado, significa que la varianza del error depende de los predictores, y por tanto, hay heteroscedasticidad.
  - **Hipótesis Nula (H_0):** El modelo es homocedástico.
  - **Decisión:** Un p-valor pequeño (p. ej., < 0.05) es evidencia en contra de la homocedasticidad.

Ejemplo de un modelo válido

Analicemos nuestro `modelo_estudio`. Nos centraremos en el gráfico `Scale-Location` (`which = 3`) y en la prueba de Breusch-Pagan.

```r
# Crear datos para el gráfico Scale-Location
datos_scale_loc <- data.frame(
  valores_ajustados = fitted(modelo_estudio),
  residuos_std_sqrt = sqrt(abs(rstandard(modelo_estudio)))
)

# Gráfico Scale-Location con ggplot2
ggplot(datos_scale_loc, aes(x = valores_ajustados, y = residuos_std_sqrt)) +
  geom_point(color = "#0072B2", alpha = 0.7) +
  geom_smooth(method = "loess", se = FALSE, color = "red", linewidth = 0.8) +
  labs(
    title = "Scale-Location",
    x = "Valores Ajustados",
    y = expression(sqrt("|Residuos Estandarizados|"))
  ) +
  theme_classic(base_size = 12)
```

```
`geom_smooth()` using formula = 'y ~ x'
```

```r
# Prueba de Breusch-Pagan
library(lmtest)
```

```
Loading required package: zoo
```

```
Attaching package: 'zoo'
```

```
The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

```r
bptest(modelo_estudio)
```

```

    studentized Breusch-Pagan test

data:  modelo_estudio
BP = 0.019638, df = 1, p-value = 0.8886
```

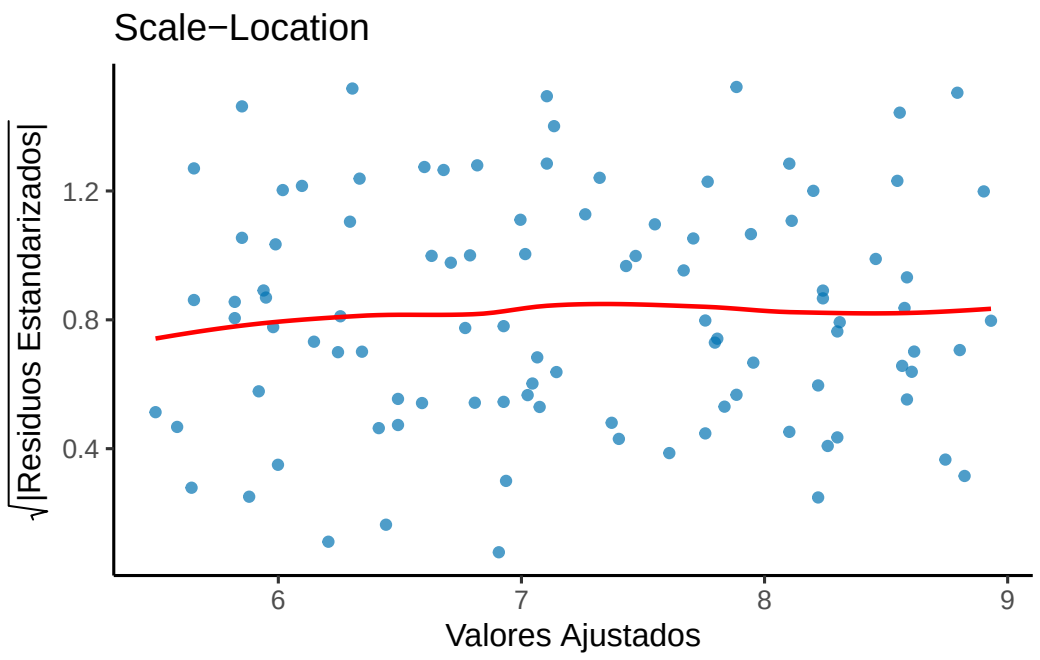


Figure 2.5: Gráfico Scale-Location para el modelo de estudio. La línea de tendencia es casi plana.

El diagnóstico es positivo. En el gráfico `Scale-Location`, la línea roja es casi horizontal, lo que indica que la varianza de los residuos es estable a lo largo de los valores ajustados. Esto se confirma con la prueba de Breusch-Pagan, que arroja un **p-valor alto**, por lo que no tenemos evidencia para rechazar la hipótesis nula de homocedasticidad. **Nuestro modelo cumple el supuesto**.

Contraejemplo: Violación del supuesto de homocedasticidad

Ahora, simularemos datos donde el error aumenta a medida que `x` crece, un caso clásico de heteroscedasticidad.

```r
# 1. Simulación de datos heteroscedásticos
set.seed(101)
x_hetero <- 1:100
y_hetero <- 10 + 2 * x_hetero + rnorm(100, mean = 0, sd = 0.4 * x_hetero)
datos_hetero <- data.frame(x = x_hetero, y = y_hetero)
modelo_hetero <- lm(y ~ x, data = datos_hetero)

# 2. Preparar datos para los gráficos
datos_diag_hetero <- data.frame(
  residuos = residuals(modelo_hetero),
  valores_ajustados = fitted(modelo_hetero),
  residuos_std_sqrt = sqrt(abs(rstandard(modelo_hetero)))
)

# 3. Gráfico de Residuos vs. Valores Ajustados
p1 <- ggplot(datos_diag_hetero, aes(x = valores_ajustados, y = residuos)) +
  geom_point(color = "#D55E00", alpha = 0.7) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  geom_smooth(method = "loess", se = FALSE, color = "red", linewidth = 0.8) +
  labs(
    title = "Residuos vs. Valores Ajustados",
    x = "Valores Ajustados",
    y = "Residuos"
  ) +
  theme_classic(base_size = 10)

# 4. Gráfico Scale-Location
p2 <- ggplot(datos_diag_hetero, aes(x = valores_ajustados, y = residuos_std_sqrt)) +
  geom_point(color = "#D55E00", alpha = 0.7) +
  geom_smooth(method = "loess", se = FALSE, color = "red", linewidth = 0.8) +
  labs(
    title = "Scale-Location",
    x = "Valores Ajustados",
    y = expression(sqrt("|Residuos Estandarizados|"))
  ) +
  theme_classic(base_size = 10)

# 5. Mostrar ambos gráficos lado a lado
library(gridExtra)
grid.arrange(p1, p2, ncol = 2)
```

```
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
```

```
# 6. Prueba de Breusch-Pagan
library(lmtest)
test_values <- bptest(modelo_hetero)
```

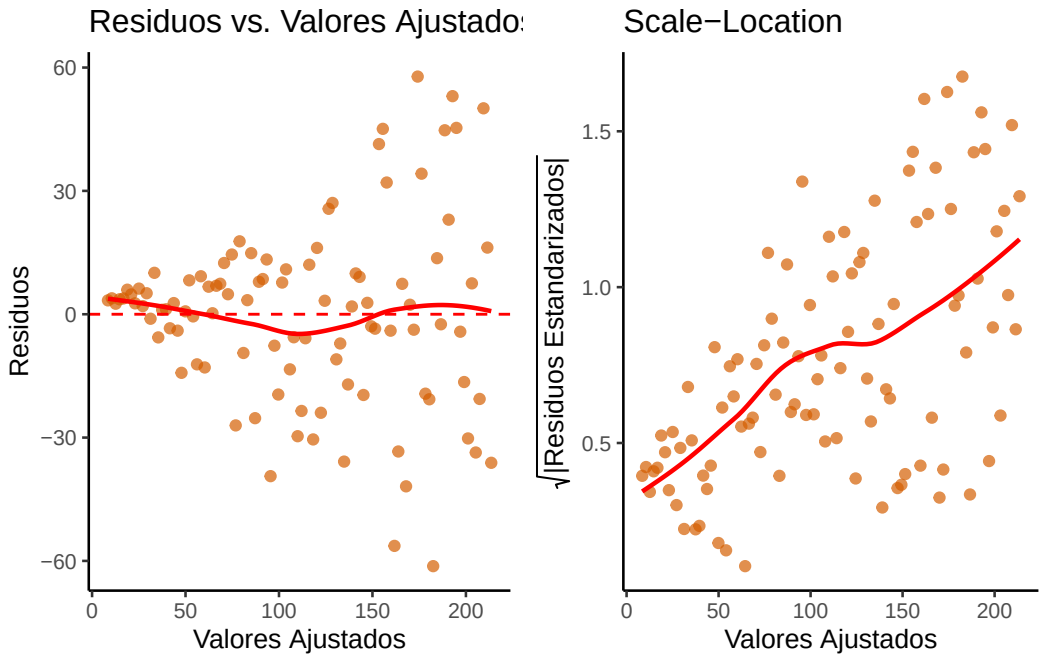


Figure 2.6: Diagnóstico de heteroscedasticidad. Izquierda: Gráfico de Residuos vs. Ajustados (patrón de embudo). Derecha: Gráfico Scale-Location (tendencia ascendente).

Los resultados son un libro de texto sobre la heteroscedasticidad.

- El gráfico de **Residuos vs. Valores Ajustados** (izquierda) tiene una **forma de embudo** inconfundible: la dispersión de los puntos aumenta drásticamente de izquierda a derecha.
- El gráfico **Scale-Location** (derecha) confirma el problema, mostrando una línea roja con una **clara pendiente ascendente**.
- La **prueba de Breusch-Pagan** arroja un **p-valor 7.43e-07**, dándonos una fuerte evidencia estadística para rechazar la hipótesis nula de homocedasticidad.

Este modelo viola claramente el supuesto, y las inferencias basadas en él (como el p-valor del coeficiente de `x`) no serían fiables.

2.6.3 Normalidad de los residuos

Este supuesto postula que los residuos del modelo (ε_i) siguen una distribución normal: $\varepsilon_i \sim N(0, \sigma^2)$. Es especialmente importante para la validez de los intervalos de confianza y los contrastes de hipótesis cuando el tamaño de la muestra es pequeño.

Para evaluar la normalidad disponemos de herramientas visuales y analíticas:

- **Gráfico Normal Q-Q (`Normal Q-Q Plot`):** Compara los cuantiles de los residuos estandarizados con los cuantiles de una distribución normal teórica. Los puntos deben caer muy cerca de la línea diagonal de 45 grados.
- **Histograma de los Residuos:** Un simple histograma de los residuos debe mostrar una forma aproximada de campana de Gauss.
- **Prueba de Shapiro-Wilk:** Es uno de los contrastes más potentes para la normalidad.
  - **Hipótesis Nula (H_0):** Los residuos provienen de una distribución normal.
  - **Decisión:** Un p-valor pequeño (< 0.05) sugiere rechazar H_0.

Ejemplo de normalidad válida

Para nuestro `modelo_estudio`, examinamos la normalidad mediante el gráfico Q-Q y la prueba de Shapiro-Wilk.

```r
# Crear datos para los gráficos
residuos <- residuals(modelo_estudio)

# 1. Gráfico Q-Q con ggplot2
datos_qq <- data.frame(residuos = residuos)

p1 <- ggplot(datos_qq, aes(sample = residuos)) +
  geom_qq(color = "#0072B2", alpha = 0.7) +
  geom_qq_line(color = "red", linetype = "dashed") +
  labs(
    title = "Normal Q-Q Plot",
    x = "Cuantiles Teóricos",
    y = "Cuantiles de la Muestra"
  ) +
  theme_classic(base_size = 10)

# 2. Histograma con ggplot2
datos_hist <- data.frame(residuos = residuos)

p2 <- ggplot(datos_hist, aes(x = residuos)) +
  geom_histogram(aes(y = after_stat(density)), bins = 15, fill = "lightblue",
                 color = "black", alpha = 0.7) +
  stat_function(fun = dnorm,
                args = list(mean = mean(residuos), sd = sd(residuos)),
                color = "red", linewidth = 1) +
  labs(
    title = "Histograma de Residuos",
    x = "Residuos",
    y = "Densidad"
  ) +
  theme_classic(base_size = 10)

# 3. Mostrar ambos gráficos lado a lado
library(gridExtra)
grid.arrange(p1, p2, ncol = 2)

# Prueba de Shapiro-Wilk
library(lmtest)
shapiro_resultado <- shapiro.test(residuals(modelo_estudio))
```

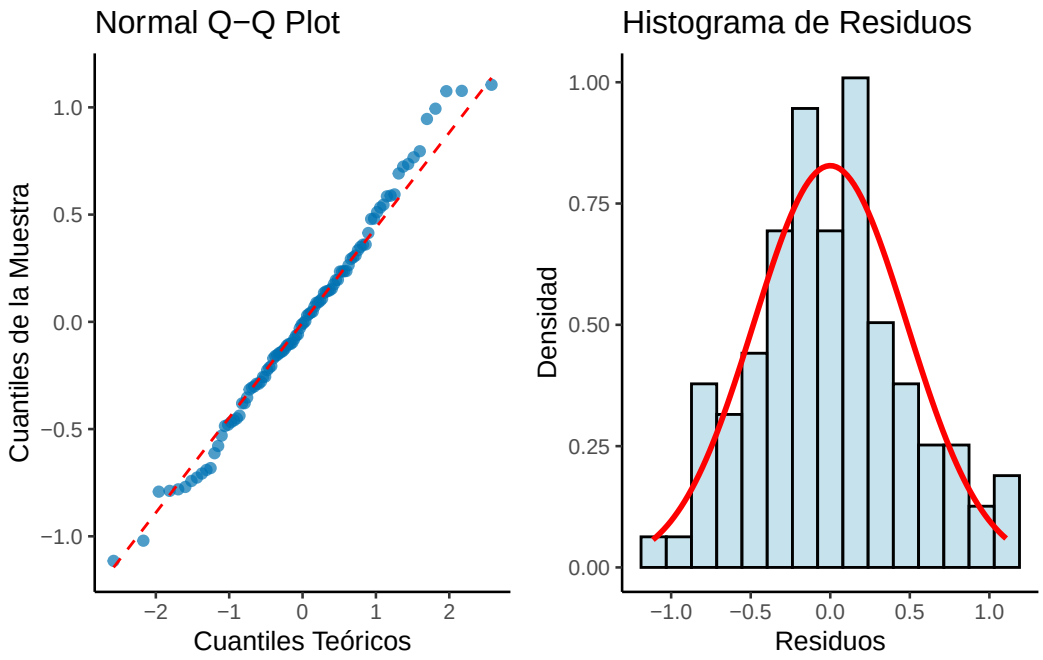


Figure 2.7: Diagnóstico de normalidad del modelo de estudio. Izquierda: Q-Q Plot. Derecha: Histograma de residuos.

El diagnóstico es excelente. En el gráfico Q-Q, los puntos se alinean muy bien con la línea diagonal, indicando normalidad. El histograma muestra una distribución aproximadamente simétrica que se ajusta bien a la curva normal teórica (línea roja). La prueba de Shapiro-Wilk confirma esto con un **p-valor alto** (6.71e-01), por lo que no rechazamos la hipótesis nula de normalidad.

Contraejemplo: Violación del supuesto de normalidad

Ahora simularemos datos donde los residuos siguen una distribución asimétrica (distribución exponencial) para mostrar una violación clara del supuesto de normalidad.

```r
# 1. Simulación de datos con errores no normales (exponenciales)
set.seed(456)
x_no_normal <- 1:100
# Errores exponenciales (muy asimétricos) centrados en 0
errores_exp <- rexp(100, rate = 1) - 1  # Restamos 1 para centrar en 0
y_no_normal <- 5 + 2 * x_no_normal + errores_exp * 10
datos_no_normal <- data.frame(x = x_no_normal, y = y_no_normal)
modelo_no_normal <- lm(y ~ x, data = datos_no_normal)

# 2. Crear datos para los gráficos
residuos_no_normal <- residuals(modelo_no_normal)

# 3. Gráfico Q-Q con ggplot2
datos_qq_mal <- data.frame(residuos = residuos_no_normal)

p1_mal <- ggplot(datos_qq_mal, aes(sample = residuos)) +
  geom_qq(color = "#D55E00", alpha = 0.7) +
  geom_qq_line(color = "red", linetype = "dashed") +
  labs(
    title = "Normal Q-Q Plot (Violación)",
    x = "Cuantiles Teóricos",
    y = "Cuantiles de la Muestra"
  ) +
  theme_classic(base_size = 10)

# 4. Histograma con ggplot2
datos_hist_mal <- data.frame(residuos = residuos_no_normal)

p2_mal <- ggplot(datos_hist_mal, aes(x = residuos)) +
  geom_histogram(aes(y = after_stat(density)), bins = 15, fill = "lightcoral",
                 color = "black", alpha = 0.7) +
  stat_function(fun = dnorm,
                args = list(mean = mean(residuos_no_normal), sd = sd(residuos_no_normal)),
                color = "blue", linewidth = 1) +
  labs(
    title = "Histograma de Residuos (Violación)",
    x = "Residuos",
    y = "Densidad"
  ) +
  theme_classic(base_size = 10)

# 5. Mostrar ambos gráficos lado a lado
grid.arrange(p1_mal, p2_mal, ncol = 2)

# Prueba de Shapiro-Wilk
shapiro_resultado <- shapiro.test(residuals(modelo_no_normal))
```

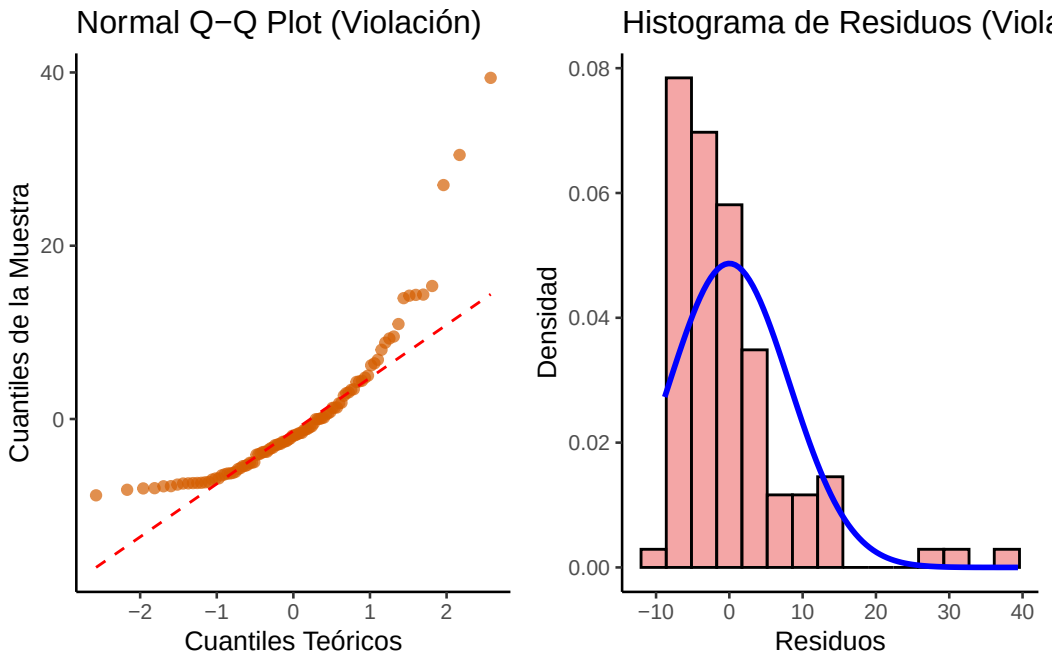


Figure 2.8: Violación del supuesto de normalidad. Izquierda: Q-Q Plot con clara desviación. Derecha: Histograma asimétricamente distribuido.

La violación es evidente. En el gráfico Q-Q, los puntos se desvían sistemáticamente de la línea diagonal, especialmente en los extremos, formando una curva característica de distribuciones asimétricas. El histograma muestra una clara asimetría hacia la derecha que no se ajusta a la curva normal teórica (línea azul). La prueba de Shapiro-Wilk arroja un **p-valor muy pequeño** (1.78e-10), rechazando fuertemente la hipótesis nula de normalidad.

2.6.4 Independencia de los residuos

Este supuesto afirma que el error de una observación no está correlacionado con el de ninguna otra: $Cov(\varepsilon_i, \varepsilon_j) = 0$ para $i \neq j$. La violación, conocida como **autocorrelación**, es común en datos de series temporales.

La **Prueba de Durbin-Watson** es el contraste clásico para la autocorrelación de primer orden. Su estadístico se calcula como:

$$DW = \frac{\sum_{i=2}^{n}(e_i - e_{i-1})^2}{\sum_{i=1}^{n} e_i^2}$$

El estadístico varía entre 0 y 4. Un valor cercano a 2 sugiere no autocorrelación. Valores cercanos a 0 indican autocorrelación positiva, y cercanos a 4, autocorrelación negativa.

Ejemplo de independencia válida

Para nuestro `modelo_estudio`, evaluamos la independencia mediante el gráfico de residuos vs orden y la prueba de Durbin-Watson.

```
# Gráfico de residuos vs orden de observación con ggplot2
datos_orden <- data.frame(
  orden = 1:length(residuals(modelo_estudio)),
  residuos = residuals(modelo_estudio)
)

ggplot(datos_orden, aes(x = orden, y = residuos)) +
  geom_point(color = "#0072B2", alpha = 0.7) +
  geom_line(color = "#0072B2", alpha = 0.5) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(
    title = "Residuos vs Orden de Observación",
    x = "Orden de observación",
    y = "Residuos"
  ) +
  theme_classic(base_size = 12)

# Prueba de Durbin-Watson
library(lmtest)
dw_resultado_valido <- dwtest(modelo_estudio)
```

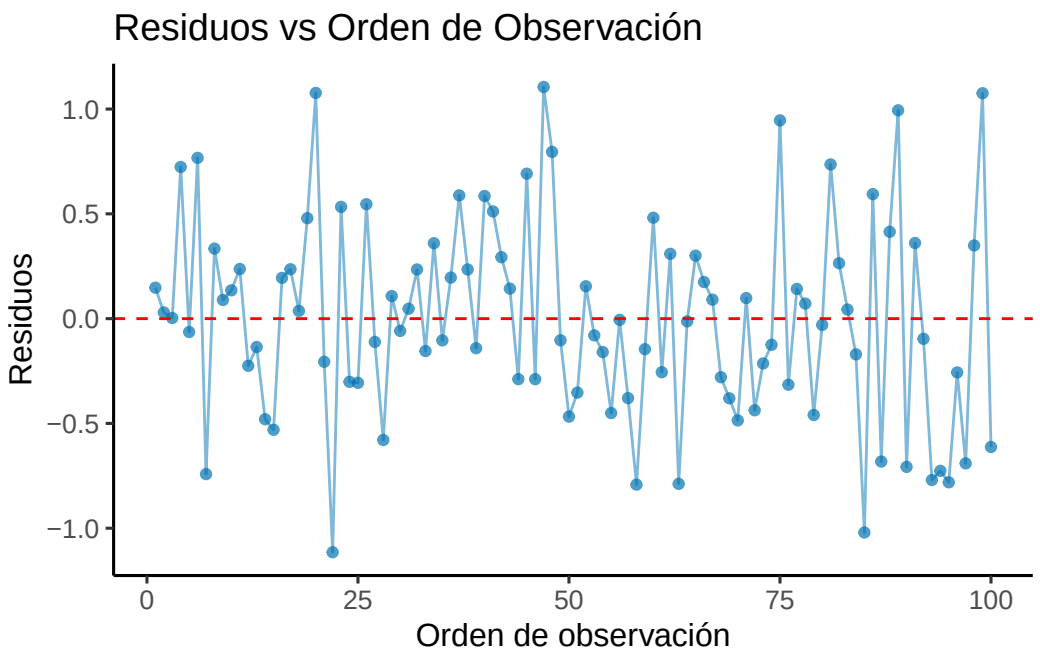


Figure 2.9: Diagnóstico de independencia del modelo de estudio. Los residuos no muestran patrones temporales.

El diagnóstico es satisfactorio. El gráfico de residuos vs orden no muestra ningún patrón sistemático o tendencia temporal, los puntos fluctúan aleatoriamente alrededor del cero. La prueba de Durbin-Watson arroja un estadístico de **2.056** (cercano a 2) y un **p-valor de 0.61**, confirmando que no hay evidencia de autocorrelación. **El supuesto de independencia se cumple**.

Contraejemplo: Violación del supuesto de independencia

Simularemos datos con autocorrelación positiva, donde cada residuo está correlacionado con el anterior, violando el supuesto de independencia.

```r
# 1. Simulación de datos con autocorrelación
set.seed(789)
n <- 100
x_autocorr <- 1:n
# Generamos errores autocorrelacionados (AR1 con phi = 0.7)
errores_autocorr <- numeric(n)
errores_autocorr[1] <- rnorm(1)
for(i in 2:n) {
  errores_autocorr[i] <- 0.7 * errores_autocorr[i-1] + rnorm(1, sd = 0.5)
}

y_autocorr <- 10 + 1.5 * x_autocorr + errores_autocorr * 3
datos_autocorr <- data.frame(x = x_autocorr, y = y_autocorr)
modelo_autocorr <- lm(y ~ x, data = datos_autocorr)

# 2. Gráfico de residuos vs orden con ggplot2
datos_orden_autocorr <- data.frame(
  orden = 1:length(residuals(modelo_autocorr)),
  residuos = residuals(modelo_autocorr)
)

ggplot(datos_orden_autocorr, aes(x = orden, y = residuos)) +
  geom_point(color = "#D55E00", alpha = 0.7) +
  geom_line(color = "#D55E00", alpha = 0.5) +
  geom_hline(yintercept = 0, color = "blue", linetype = "dashed") +
  labs(
    title = "Residuos vs Orden de Observación (Violación)",
    x = "Orden de observación",
    y = "Residuos"
  ) +
  theme_classic(base_size = 12)

# 3. Prueba de Durbin-Watson
dw_resultado <- dwtest(modelo_autocorr)
```

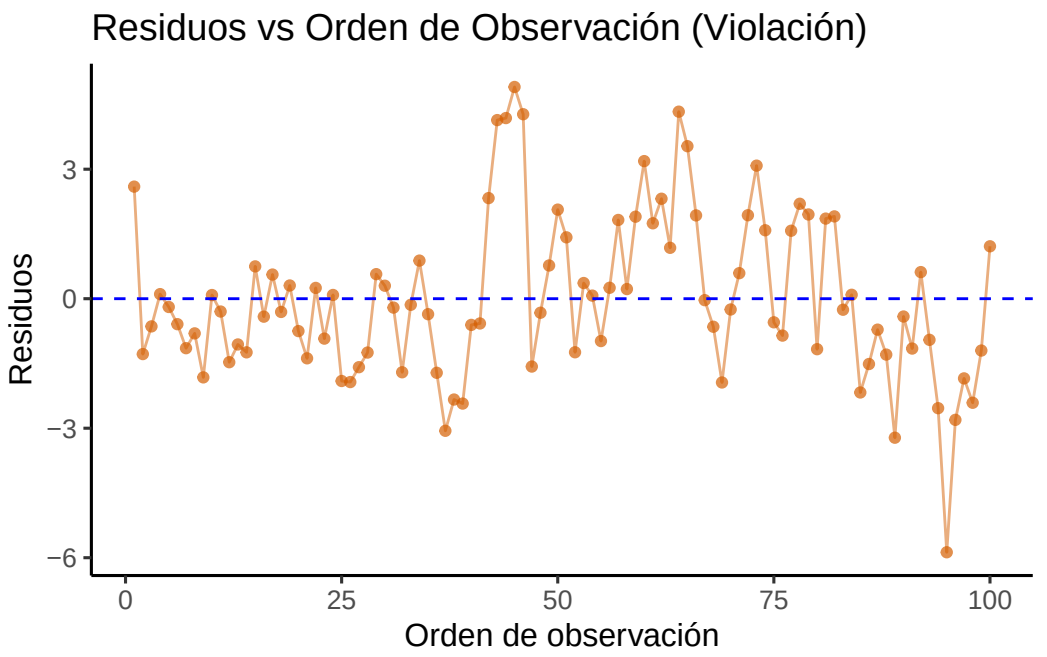


Figure 2.10: Violación del supuesto de independencia. Los residuos muestran un patrón de autocorrelación positiva.

La violación es clara. El gráfico de residuos vs orden muestra un patrón ondulante característico: los residuos tienden a mantenerse del mismo signo durante varias observaciones consecutivas (rachas de valores positivos seguidas de rachas de valores negativos). Esto indica **autocorrelación positiva**. La prueba de Durbin-Watson confirma esto con un estadístico muy por debajo de 2 (DW = **0.74**) y un **p-valor muy pequeño** (**5.01e-11**), rechazando fuertemente la hipótesis nula de independencia.

2.6.5 Media nula de los residuos

Un requisito fundamental del modelo es que la media de los residuos debe ser exactamente cero: $E[e_i] = 0$. Esta propiedad se deriva matemáticamente del método de mínimos cuadrados y su verificación sirve como una comprobación de que nuestros cálculos son correctos.

2.6.6 Identificación de observaciones influyentes y atípicas

Algunos puntos pueden tener una influencia desproporcionada en el modelo. Es crucial identificarlos usando diferentes métricas que evalúan aspectos complementarios de la influencia

(Kutner et al. 2005; Fox and Weisberg 2018). Las métricas desarrolladas por Cook, Belsley, Kuh y Welsch proporcionan herramientas robustas para este diagnóstico.

i Fundamento teórico: de los residuos simples a los estudentizados

Antes de analizar las métricas de influencia, debemos entender por qué no todos los **residuos simples** ($e_i = y_i - \hat{y}_i$) son comparables entre sí. El problema fundamental es que **no tienen la misma varianza**, incluso bajo homocedasticidad.
La varianza teórica del residuo e_i depende del **apalancamiento** (*leverage*) de la observación:

$$\text{Var}(e_i) = \sigma^2(1 - h_{ii})$$

donde el apalancamiento h_{ii} se define como:

$$h_{ii} = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{j=1}^{n}(x_j - \bar{x})^2}$$

Las observaciones con valores de X más alejados de la media tendrán mayor apalancamiento y, paradójicamente, residuos con **menor varianza**. Por esto, un residuo pequeño en una observación de alto leverage puede ser más preocupante que un residuo grande en el centro de los datos.
Los residuos estandarizados solucionan parcialmente este problema:

$$r_i^* = \frac{e_i}{\sqrt{\text{MSE}(1 - h_{ii})}}$$

Pero **los residuos estudentizados** van un paso más allá, eliminando el sesgo de autoinfluencia:

$$r_i = \frac{e_i}{\sqrt{\text{MSE}_{(-i)}(1 - h_{ii})}}$$

donde $\text{MSE}_{(-i)}$ excluye la observación i del ajuste. Esto evita que un outlier "contamine" su propia evaluación y proporciona una distribución teórica exacta (t de Student con $n - k - 2$ grados de libertad).
¿Por qué son superiores los residuos estudentizados? Por tres razones clave: (1) **eliminan el sesgo de autoinfluencia** al excluir cada observación de su propia evaluación, (2) **evitan la contaminación** que un outlier produce en la MSE global, y (3) **siguen una distribución conocida** (t de Student) que permite umbrales estadísticamente precisos. **En la práctica:** $|r_i| > 2$ indica posibles outliers (5% en normalidad) y $|r_i| > 3$ outliers muy probables (<1%).

Las métricas fundamentales de influencia para identificar observaciones problemáticas son:

- **Apalancamiento (Leverage,** h_{ii}): Mide cuán atípico es el valor de la variable predictora X_i de una observación. Un apalancamiento alto significa que el punto tiene el

potencial de ser muy influyente. En regresión simple, se calcula como:

$$h_{ii} = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{j=1}^{n}(x_j - \bar{x})^2}$$

Una regla común es considerar un apalancamiento alto si $h_{ii} > \frac{2(k+1)}{n}$, donde k es el número de predictores (1 en regresión simple).

- **Distancia de Cook (D_i)**: Mide la influencia global de una observación, combinando su apalancamiento y su residuo. Representa cuánto cambian los coeficientes del modelo si la i-ésima observación es eliminada.

$$D_i = \frac{r_i^2}{k+1} \cdot \frac{h_{ii}}{1 - h_{ii}}$$

Se considera que un punto es influyente si su distancia de Cook es grande, por ejemplo, si $D_i > 1$ o $D_i > 4/(n-k-1)$.

- **DFFITS:** Mide cuánto cambia la predicción $\hat{y}_i$ cuando se elimina la i-ésima observación. Es una medida estandarizada que combina el residuo estudentizado y el apalancamiento.

$$\text{DFFITS}_i = r_i \sqrt{\frac{h_{ii}}{1 - h_{ii}}}$$

Un punto se considera influyente si $|\text{DFFITS}_i| > 2\sqrt{(k+1)/n}$, donde k es el número de predictores.

Ejemplo: Cálculo y análisis de DFFITS

DFFITS es especialmente útil para evaluar cómo cada observación afecta a su propia predicción. Analicemos esta medida con nuestro `modelo_estudio`.

```r
# Calcular DFFITS y sus componentes
dffits_vals <- dffits(modelo_estudio)
residuos_stud <- rstudent(modelo_estudio)  # Residuos estudentizados
leverage_vals <- hatvalues(modelo_estudio)

# Crear dataframe para análisis
datos_dffits <- data.frame(
  observacion = 1:length(dffits_vals),
  dffits = dffits_vals,
  residuo_stud = residuos_stud,
  leverage = leverage_vals
)

# Umbral de DFFITS
n <- nrow(datos)
k <- 1  # número de predictores
dffits_threshold <- 2 * sqrt((k + 1) / n)

# Gráfico de DFFITS
ggplot(datos_dffits, aes(x = observacion, y = dffits)) +
  geom_point(color = "#0072B2", alpha = 0.7) +
  geom_hline(yintercept = 0, color = "gray", linetype = "dashed") +
  geom_hline(yintercept = c(-dffits_threshold, dffits_threshold),
             color = "red", linetype = "dashed", alpha = 0.7) +
  labs(
    title = "DFFITS por Observación",
    x = "Número de Observación",
    y = "DFFITS",
    caption = paste("Líneas rojas: umbrales ±", round(dffits_threshold, 3))
  ) +
  theme_classic(base_size = 12)

# Análisis cuantitativo
influential_dffits <- which(abs(dffits_vals) > dffits_threshold)
top_indices <- order(abs(dffits_vals), decreasing = TRUE)[1:5]
```

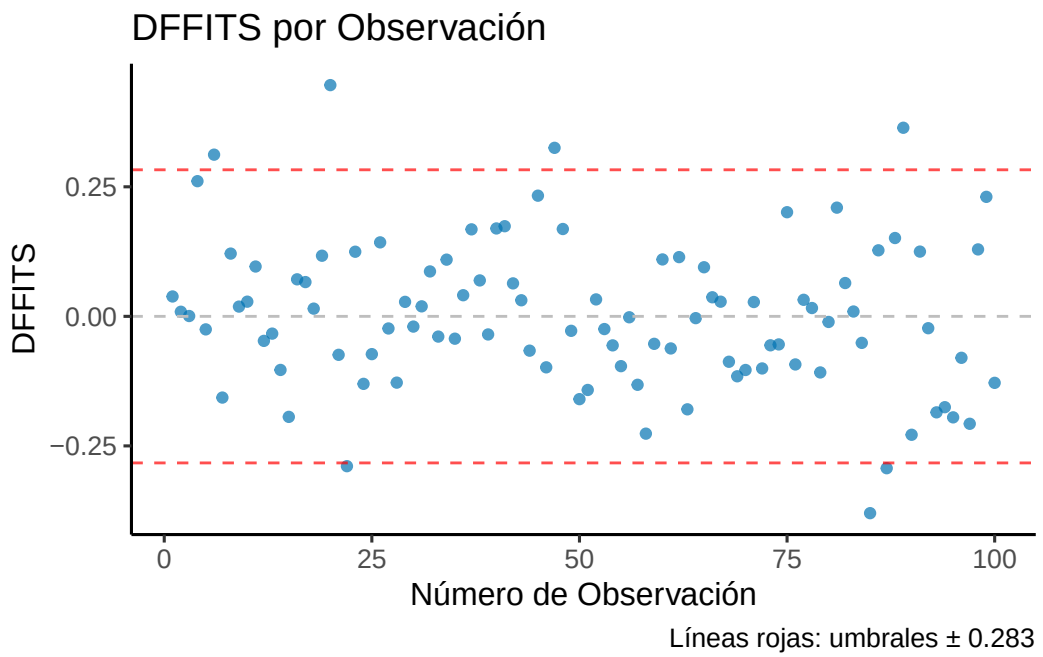


Figure 2.11: Análisis de DFFITS para identificar observaciones que afectan significativamente a sus propias predicciones.

Análisis de resultados:
El umbral de influencia para DFFITS es 0.283. En nuestro modelo, **7 observaciones superan este umbral**: las observaciones 6, 20, 22, 47, 85, 87, 89, lo que las clasifica como influyentes según este criterio.
Las **cinco observaciones con mayor |DFFITS|** son las observaciones 20, 85, 89, 47, 6, con valores de 0.446, -0.38, 0.364, 0.325, 0.312 respectivamente. Lo más notable es que **todas estas cinco observaciones** (20, 85, 89, 47, 6) **superan el umbral de DFFITS**, confirmando su carácter influyente.
Interpretación clave: La observación 20 es el caso más destacado: tiene un DFFITS de 0.446, superando el umbral de 0.283. Esta combinación de residuo y apalancamiento resulta en un DFFITS significativo que indica cambios sustanciales en su predicción.
Conclusión práctica: Tenemos **7 observaciones influyentes** según DFFITS (6, 20, 22, 47, 85, 87, 89) que merecen investigación adicional. Estas observaciones cambian significativamente sus propias predicciones cuando son eliminadas del modelo, sugiriendo que podrían representar casos especiales o errores de medición que deberían ser examinados más detalladamente.

El gráfico `Residuals vs. Leverage` es la herramienta visual más importante para el diagnóstico de influencia, ya que combina en un solo gráfico el **apalancamiento** (eje X) y los **residuos estudentizados** (eje Y), permitiendo identificar simultáneamente observaciones

con alto leverage y outliers. Además, incluye curvas que delimitan regiones de alta **Distancia de Cook**, facilitando la identificación visual de los puntos más problemáticos.

Ejemplo: Gráfico Residuals vs. Leverage

Vamos a analizar el gráfico más importante para el diagnóstico de influencia usando nuestro `modelo_estudio`.

```r
# Crear datos para el gráfico Residuals vs. Leverage
leverage_vals <- hatvalues(modelo_estudio)
residuos_stud <- rstudent(modelo_estudio)  # Residuos estudentizados
cook_dist <- cooks.distance(modelo_estudio)

datos_leverage <- data.frame(
  leverage = leverage_vals,
  residuos_stud = residuos_stud,
  cook = cook_dist,
  observacion = 1:length(leverage_vals)
)

# Calcular umbrales
n <- nrow(datos)
k <- 1
leverage_threshold <- 2 * (k + 1) / n
cook_threshold <- 4 / (n - k - 1)

# Función para crear curvas de Cook
cook_curve <- function(leverage, cook_value, k) {
  sqrt(cook_value * (k + 1) * (1 - leverage) / leverage)
}

# Crear curvas de Cook para diferentes valores
lev_seq <- seq(0.001, max(leverage_vals) * 1.1, length.out = 100)
cook_05 <- data.frame(
  leverage = lev_seq,
  pos = cook_curve(lev_seq, 0.5, k),
  neg = -cook_curve(lev_seq, 0.5, k)
)
cook_1 <- data.frame(
  leverage = lev_seq,
  pos = cook_curve(lev_seq, 1, k),
  neg = -cook_curve(lev_seq, 1, k)
)

# Gráfico Residuals vs. Leverage con ggplot2
ggplot(datos_leverage, aes(x = leverage, y = residuos_stud)) +
  # Curvas de Cook
  geom_line(data = cook_05, aes(x = leverage, y = pos),
            color = "red", linetype = "dashed", alpha = 0.6, inherit.aes = FALSE) +
  geom_line(data = cook_05, aes(x = leverage, y = neg),
            color = "red", linetype = "dashed", alpha = 0.6, inherit.aes = FALSE) +
  geom_line(data = cook_1, aes(x = leverage, y = pos),
            color = "red", linetype = "solid", alpha = 0.8, inherit.aes = FALSE) +
  geom_line(data = cook_1, aes(x = leverage, y = neg),
            color = "red", linetype = "solid", alpha = 0.8, inherit.aes = FALSE) +
  # Puntos de datos
  geom_point(color = "#0072B2", alpha = 0.7, size = 2) +
  # Líneas de referencia
  geom_hline(yintercept = 0, color = "gray", linetype = "dashed") +
  geom_hline(yintercept = c(-2, 2), color = "orange", linetype = "dotted", alpha = 0.7) +
```

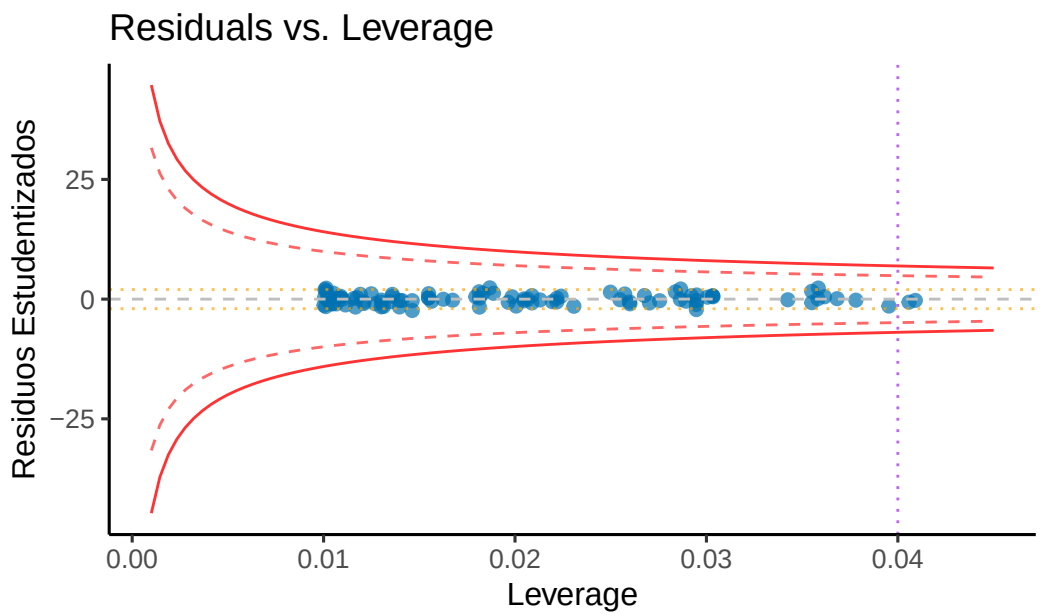


Figure 2.12: Gráfico Residuals vs. Leverage para identificar observaciones influyentes.

Análisis del gráfico:
El gráfico revela varios puntos importantes. Tenemos 6 outliers (residuos estudentizados > 2): las observaciones 20, 22, 47, 85, 89, 99. Además, 2 observaciones superan el umbral de leverage (> 0.04): las observaciones 24, 74.
Interpretación por regiones:

- **Zona derecha (alto leverage)**: Las observaciones 24, 74 superan el umbral de leverage, lo que significa que tienen valores de X atípicos y **alto potencial influyente**
- **Zona izquierda superior/inferior**: Los 6 outliers (20, 22, 47, 85, 89, 99) están distribuidos aquí, con leverage bajo-moderado pero residuos grandes
- **Esquinas críticas**: Afortunadamente vacías (alto leverage + outlier sería muy problemático)

Distancia de Cook: Las curvas rojas muestran que aunque ningún punto supera Cook = 1.0 (línea continua), **varios puntos se acercan a la curva de Cook = 0.5** (línea discontinua), indicando influencia moderada. Las observaciones con alto leverage están en esta zona de influencia moderada.
Conclusión práctica: El modelo presenta una **situación favorable**: aunque tenemos **outliers** (observaciones 20, 22, 47, 85, 89, 99) que son atípicos en Y, y **observaciones**

de alto leverage (observaciones 24, 74) que son atípicos en X, **crucialmente no hay solapamiento entre ambos grupos**. Esto significa que no tenemos la situación más problemática (alto leverage + outlier). Aun así, ambos grupos merecen investigación.

2.6.6.1 Interpretación práctica de las medidas de influencia

Cada medida nos proporciona información complementaria sobre diferentes aspectos de la influencia:

- **Leverage (Apalancamiento):** Identifica observaciones con valores "raros" en las variables predictoras. Alto leverage no es necesariamente problemático, pero indica potencial para ser influyente.
- **Distancia de Cook:** Es la medida más general de influencia. Valores altos indican que eliminar esa observación cambiaría substancialmente los coeficientes del modelo.
- **DFFITS:** Se enfoca específicamente en cómo cambia la predicción de cada punto cuando se elimina esa observación. Es especialmente útil para evaluar el impacto en las predicciones.

En la práctica, una observación es especialmente preocupante si es problemática según **múltiples criterios** a la vez.

Diagnóstico completo del modelo de estudio

A continuación, realizamos todas las verificaciones de diagnóstico para nuestro `modelo_estudio`:

```r
# Preparar todos los datos necesarios para los gráficos
residuos_completo <- residuals(modelo_estudio)
valores_ajustados_completo <- fitted(modelo_estudio)
residuos_std <- rstandard(modelo_estudio)
leverage_vals <- hatvalues(modelo_estudio)

# 1. Gráfico Residuos vs. Valores Ajustados
p1_completo <- ggplot(data.frame(x = valores_ajustados_completo, y = residuos_completo),
                      aes(x = x, y = y)) +
  geom_point(color = "#0072B2", alpha = 0.7) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  geom_smooth(method = "loess", se = FALSE, color = "red", linewidth = 0.8) +
  labs(title = "Residuos vs. Valores Ajustados", x = "Valores Ajustados", y = "Residuos") +
  theme_classic(base_size = 10)

# 2. Gráfico Q-Q Normal
datos_qq_completo <- data.frame(residuos = residuos_std)

p2_completo <- ggplot(datos_qq_completo, aes(sample = residuos)) +
  geom_qq(color = "#0072B2", alpha = 0.7) +
  geom_qq_line(color = "red", linetype = "dashed") +
  labs(title = "Normal Q-Q Plot", x = "Cuantiles Teóricos", y = "Cuantiles de la Muestra") +
  theme_classic(base_size = 10)

# 3. Gráfico Scale-Location
p3_completo <- ggplot(data.frame(x = valores_ajustados_completo,
                                 y = sqrt(abs(residuos_std))),
                      aes(x = x, y = y)) +
  geom_point(color = "#0072B2", alpha = 0.7) +
  geom_smooth(method = "loess", se = FALSE, color = "red", linewidth = 0.8) +
  labs(title = "Scale-Location", x = "Valores Ajustados",
       y = expression(sqrt("|Residuos Estandarizados|"))) +
  theme_classic(base_size = 10)

# 4. Gráfico Residuos vs. Leverage
p4_completo <- ggplot(data.frame(x = leverage_vals, y = residuos_std),
                      aes(x = x, y = y)) +
  geom_point(color = "#0072B2", alpha = 0.7) +
  geom_smooth(method = "loess", se = FALSE, color = "red", linewidth = 0.8) +
  geom_hline(yintercept = 0, color = "gray", linetype = "dashed") +
  labs(title = "Residuos vs. Leverage", x = "Leverage", y = "Residuos Estandarizados") +
  theme_classic(base_size = 10)

# 5. Histograma de residuos
p5_completo <- ggplot(data.frame(residuos = residuos_completo), aes(x = residuos)) +
  geom_histogram(aes(y = after_stat(density)), bins = 15, fill = "lightblue",
                 color = "black", alpha = 0.7) +
  stat_function(fun = dnorm,
                args = list(mean = mean(residuos_completo), sd = sd(residuos_completo)),
                color = "red", linewidth = 1) +
  labs(title = "Histograma de Residuos", x = "Residuos", y = "Densidad") +
  theme_classic(base_size = 10)
```

```
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
```

```
# Pruebas analíticas
library(lmtest)

cat("=== DIAGNÓSTICO ANALÍTICO ===\n\n")
```

```
=== DIAGNÓSTICO ANALÍTICO ===
```

```
# 1. Verificación de media nula de residuos
media_residuos <- mean(residuals(modelo_estudio))
cat("Media de los residuos:", round(media_residuos, 10), "\n")
```

```
Media de los residuos: 0
```

```
cat("(Debe ser prácticamente 0)\n\n")
```

```
(Debe ser prácticamente 0)
```

```
# 2. Homocedasticidad: Breusch-Pagan
cat("HOMOCEDASTICIDAD - Prueba de Breusch-Pagan:\n")
```

```
HOMOCEDASTICIDAD - Prueba de Breusch-Pagan:
```

```
bp_test <- bptest(modelo_estudio)
print(bp_test)
```

```

    studentized Breusch-Pagan test

data:  modelo_estudio
BP = 0.019638, df = 1, p-value = 0.8886
```

```
cat("Interpretación: p-valor >", 0.05, "→ No hay evidencia de heterocedasticidad\n\n")
```

```
Interpretación: p-valor > 0.05 → No hay evidencia de heterocedasticidad
```

```r
# 3. Normalidad: Shapiro-Wilk
cat("NORMALIDAD - Prueba de Shapiro-Wilk:\n")
```

```
NORMALIDAD - Prueba de Shapiro-Wilk:
```

```r
sw_test <- shapiro.test(residuals(modelo_estudio))
print(sw_test)
```

```

    Shapiro-Wilk normality test

data:  residuals(modelo_estudio)
W = 0.99008, p-value = 0.671
```

```r
cat("Interpretación: p-valor >", 0.05, "→ No hay evidencia contra la normalidad\n\n")
```

```
Interpretación: p-valor > 0.05 → No hay evidencia contra la normalidad
```

```r
# 4. Independencia: Durbin-Watson
cat("INDEPENDENCIA - Prueba de Durbin-Watson:\n")
```

```
INDEPENDENCIA - Prueba de Durbin-Watson:
```

```r
dw_test <- dwtest(modelo_estudio)
cat("Estadístico DW:", round(dw_test$statistic, 3), "\n")
```

```
Estadístico DW: 2.056
```

```r
cat("p-valor:", format.pval(dw_test$p.value, digits=3, eps=0.001), "\n")
```

```
p-valor: 0.61
```

```r
cat("Interpretación: Estadístico  2 y p-valor >", 0.05, "→ No hay autocorrelación\n\n")
```

```
Interpretación: Estadístico  2 y p-valor > 0.05 → No hay autocorrelación
```

```r
# 5. Identificación de observaciones influyentes
cat("OBSERVACIONES INFLUYENTES:\n")
```

OBSERVACIONES INFLUYENTES:

```r
# Calculamos las métricas principales
leverage <- hatvalues(modelo_estudio)
cook_dist <- cooks.distance(modelo_estudio)
std_residuals <- rstandard(modelo_estudio)
dffits_vals <- dffits(modelo_estudio)
covratio_vals <- covratio(modelo_estudio)

# Umbrales de corte
n <- nrow(datos)
k <- 1  # número de predictores
leverage_threshold <- 2 * (k + 1) / n
cook_threshold <- 4 / (n - k - 1)
dffits_threshold <- 2 * sqrt((k + 1) / n)
covratio_threshold <- 3 * (k + 1) / n

# Identificamos observaciones problemáticas
high_leverage <- which(leverage > leverage_threshold)
influential_cook <- which(cook_dist > cook_threshold)
outliers <- which(abs(std_residuals) > 2)
influential_dffits <- which(abs(dffits_vals) > dffits_threshold)
problematic_covratio <- which(abs(covratio_vals - 1) > covratio_threshold)

cat("Observaciones con alto apalancamiento (>", round(leverage_threshold, 3), "):",
    ifelse(length(high_leverage) > 0, paste(high_leverage, collapse = ", "), "Ninguna"), "\
```

```
Observaciones con alto apalancamiento (> 0.04 ): 24, 74
```

```r
cat("Observaciones influyentes por Cook (>", round(cook_threshold, 3), "):",
    ifelse(length(influential_cook) > 0, paste(influential_cook, collapse = ", "), "Ninguna
```

```
Observaciones influyentes por Cook (> 0.041 ): 6, 20, 47, 85, 87, 89
```

```r
cat("Observaciones influyentes por DFFITS (>", round(dffits_threshold, 3), "):",
    ifelse(length(influential_dffits) > 0, paste(influential_dffits, collapse = ", "), "Nin
```

```
Observaciones influyentes por DFFITS (> 0.283 ): 6, 20, 22, 47, 85, 87, 89
```

```r
cat("Posibles outliers (|residuo std| > 2):",
    ifelse(length(outliers) > 0, paste(outliers, collapse = ", "), "Ninguna"), "\n")
```

```
Posibles outliers (|residuo std| > 2): 20, 22, 47, 85, 89, 99
```

```r
# Resumen de observaciones más problemáticas
all_problematic <- unique(c(high_leverage, influential_cook, influential_dffits,
                            problematic_covratio, outliers))
if(length(all_problematic) > 0) {
  cat("\nRESUMEN - Observaciones que requieren atención:", paste(all_problematic, collapse
  cat("Estas observaciones deberían ser investigadas más detalladamente.\n")
} else {
  cat("\nRESUMEN - No hay observaciones problemáticas detectadas.\n")
}
```

```
RESUMEN - Observaciones que requieren atención: 24, 74, 6, 20, 47, 85, 87, 89, 22, 99
Estas observaciones deberían ser investigadas más detalladamente.
```

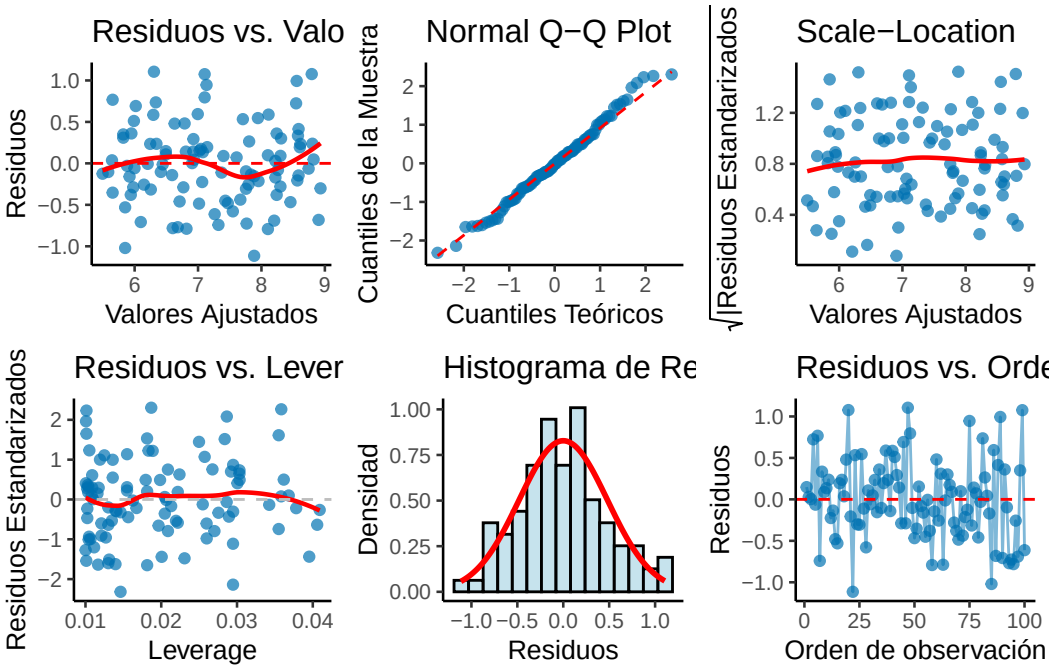


Figure 2.13: Gráficos de diagnóstico completo del modelo de regresión.

Conclusión del diagnóstico:
Nuestro modelo de estudio pasa exitosamente todas las verificaciones: - **Linealidad**: Sin patrones en residuos vs ajustados - **Homocedasticidad**: Varianza constante (Breusch-

Pagan $p > 0.05$) - **Normalidad**: Los residuos siguen distribución normal (Shapiro-Wilk $p > 0.05$) - **Independencia**: Sin autocorrelación (Durbin-Watson 2) - **Media nula**: La media de residuos es prácticamente 0 - **Sin valores influyentes**: No hay observaciones problemáticas según Cook, DFFITS o COVRATIO - **Sin outliers**: No hay residuos estandarizados extremos
Esto confirma que nuestras inferencias estadísticas (p-valores, intervalos de confianza) son válidas y confiables (James et al. 2021; Harrell 2015).

3 Métodos de selección de variables y problemas de regularización

En los modelos de regresión, especialmente cuando se trabaja con conjuntos de datos que incluyen un gran número de variables predictoras, es común enfrentarse al desafío de identificar qué variables son realmente relevantes para explicar la variable respuesta. La inclusión de demasiadas variables en un modelo puede llevar a problemas como el sobreajuste, pérdida de interpretabilidad y complejidad innecesaria, mientras que la exclusión de variables importantes puede resultar en modelos subóptimos.

3.1 Proceso de construcción del modelo de regresión

La construcción de un modelo de regresión múltiple es un proceso sistemático que busca explicar la relación entre una variable respuesta (Y) y múltiples variables predictoras ($X_1, X_2, \ldots, X_k$). Este proceso consta de varias etapas clave (Kutner et al. 2005):

1. **Definición del problema y variables de interés:**
    - Identificar claramente el objetivo del análisis, ya sea realizar predicciones, evaluar relaciones o controlar por efectos de variables confusoras.
    - Seleccionar las variables predictoras potenciales en función de su relevancia teórica, conocimiento previo o exploración inicial de los datos.
2. **Recogida de datos:**

- La calidad de los datos recogidos influye directamente en la validez de los resultados y conclusiones obtenidas. El proceso de recogida de datos consiste en recopilar información de manera organizada y sistemática para responder a las preguntas de investigación planteadas. Dependiendo del diseño del estudio y los objetivos del análisis, se pueden emplear diferentes tipos de experimentos o métodos de recogida de datos.
- Debemos asegurar las siguientes características sobre los datos.
    - **Fiabilidad:** Asegurar que los datos sean consistentes y puedan reproducirse bajo condiciones similares.
    - **Validez:** Garantizar que los datos recojan realmente la información necesaria para responder a las preguntas de investigación.
    - **Ética:** Asegurar la privacidad y el consentimiento informado de los participantes.

- **Control de Sesgos:** Diseñar el estudio de manera que se minimicen los sesgos que puedan distorsionar los resultados.

Tipos de experimentos

La elección del tipo de experimento o método de recogida de datos dependerá de la naturaleza del problema a investigar, los recursos disponibles y las limitaciones del estudio. Una correcta planificación y ejecución de esta etapa sienta las bases para un análisis robusto y confiable.

1. **Experimentos controlados:**
   - Los experimentos controlados son diseñados de manera que los investigadores manipulan deliberadamente una o más variables independientes (llamadas factores o variables controladas) para observar su efecto en la variable dependiente.
   - Incluyen la aleatorización de sujetos entre grupos (por ejemplo, grupos de control y tratamiento) para minimizar sesgos y asegurar comparabilidad.
   - En muchas ocasiones la información suplementaria no se puede incorporar en el diseño del experimento. A esas variables, no controladas, se les suel llamar covariables.
   - **Ejemplo:** Un estudio clínico donde se prueba un nuevo medicamento y se compara su efecto con un placebo.
2. **Estudios observacionales exploratorios:**
   - En este enfoque, los datos se recogen sin intervenir ni manipular las condiciones. Los investigadores observan y registran los fenómenos tal como ocurren en la naturaleza.
   - Pueden clasificarse en:
     - **Estudios transversales:** Los datos se recogen en un único punto temporal.
     - **Estudios longitudinales:** Los datos se recogen durante un periodo para analizar cambios a lo largo del tiempo.
   - **Ejemplo:** Investigar los hábitos alimenticios y su asociación con enfermedades cardiovasculares en una población.
3. **Estudios observacionales confirmatorios:**
   - En este enfoque, los datos se recogen para testear (confirmar o no) hipótesis derivadas de estudios previos o de ideas que pueden tener los investigadores.
   - En este contexto, las variables que aparecen involucradas en la hipótesis que se quiere confirmar se denominan variables primarias, y las variables explicativas que se sabe inluyen en la respuesta se llaman variables de control (en Epidemiología nos referimos a ellas como factores de riesgo)

- **Ejemplo:** Un equipo de investigadores, basándose en estudios previos, plantea la hipótesis de que existe una relación positiva entre el hábito de fumar (variable explicativa principal) y la incidencia de cáncer de pulmón (variable respuesta). Para confirmar esta hipótesis, realizan un estudio observacional en el que recopilan datos de una población durante un periodo determinado. Dado que no es ético inducir a las personas a fumar para realizar un experimento controlado, este estudio se realiza de forma observacional. Los datos se analizan para evaluar la asociación entre las variables, permitiendo confirmar (o refutar) la hipótesis planteada con un diseño adecuado y controlando los posibles factores de confusión.

4. **Encuestas y cuestionarios:**

   - Las encuestas son una técnica común para recoger datos de manera estructurada sobre actitudes, opiniones, comportamientos o características demográficas.
   - Pueden aplicarse en formato presencial, en línea, por teléfono o mediante correo.
   - **Ejemplo:** Una encuesta para medir el grado de satisfacción de los clientes con un servicio.

5. **Experimentos naturales:**

   - Se producen cuando un fenómeno natural o social actúa como una intervención en un entorno sin que los investigadores tengan control sobre el experimento.
   - Este tipo de estudio aprovecha eventos únicos para analizar sus impactos.
   - **Ejemplo:** Estudiar los efectos económicos de una nueva política fiscal aplicada en una región específica.

6. **Estudios de simulación:**

   - Los datos se generan a través de modelos matemáticos o computacionales que representan un sistema real o hipotético.
   - Este método se usa cuando es difícil o costoso realizar experimentos reales.
   - **Ejemplo:** Simular el comportamiento de un mercado financiero bajo diferentes escenarios económicos.

7. **Recogida de datos secundarios:**

   - En lugar de recoger datos nuevos, se utilizan datos ya existentes recopilados por terceros, como censos, registros administrativos o bases de datos públicas.
   - Aunque es eficiente en tiempo y costos, el investigador tiene menor control sobre la calidad y las características de los datos.
   - **Ejemplo:** Analizar datos de encuestas nacionales para estudiar tendencias sociales.

3. **Análisis Exploratorio de Datos (EDA):**

   - Inspeccionar los datos mediante análisis descriptivo y visual para identificar posibles problemas como valores atípicos, datos faltantes y multicolinealidad.
   - Escalar o transformar las variables si es necesario, especialmente si están en diferentes escalas o presentan distribuciones no lineales.

4. **Ajuste del modelo:**

   - Especificar el modelo de regresión múltiple en su forma general:

   $$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon,$$

   donde ε representa los errores aleatorios.
   - Estimar los coeficientes del modelo $(\beta_0, \beta_1, \ldots, \beta_p)$ utilizando el método de mínimos cuadrados, que minimiza la suma de los errores al cuadrado.

5. **Evaluación del modelo:**

   - Analizar el ajuste general del modelo utilizando métricas como R^2 y R^2 ajustado, que miden la proporción de la variabilidad explicada.
   - Examinar la tabla ANOVA para evaluar la significancia global del modelo.
   - Realizar pruebas de hipótesis para los coeficientes individuales, verificando si las variables predictoras tienen un efecto significativo en la variable respuesta.

6. **Diagnóstico del modelo:**

   - Examinar los residuos para evaluar supuestos como la linealidad, homocedasticidad, normalidad de los errores y ausencia de autocorrelación.
   - Identificar observaciones atípicas, leverage y puntos de influencia utilizando herramientas como la distancia de Cook, DFBETAS y DFFITS.

7. **Reducción de variables:**

   - En análisis de regresión, especialmente cuando se trabaja con conjuntos de datos de alta dimensionalidad, es común enfrentar situaciones en las que el número de variables explicativas es muy grande. Esto puede llevar a problemas como el sobreajuste, dificultades en la interpretación del modelo y una mayor complejidad computacional. Por ello, reducir el número de variables explicativas, sin perder información relevante, se convierte en un paso crucial para construir modelos más eficientes y robustos.

8. **Validación del modelo:**

   - Evaluar el desempeño del modelo con datos de validación o mediante técnicas como validación cruzada para garantizar su capacidad predictiva en nuevos conjuntos de datos.

El objetivo principal de este tema es presentar las técnicas más relevantes para la selección de variables y regularización, entender sus fundamentos teóricos, y aplicarlas a casos prácticos. Esto no solo permitirá construir modelos más robustos y eficientes, sino que también ayudará a obtener insights más claros y útiles a partir de los datos.

3.2 Reducción de variables

En análisis de regresión, especialmente cuando se trabaja con conjuntos de datos de alta dimensionalidad, es común enfrentar situaciones en las que el número de variables explicativas es muy grande. Esto puede llevar a problemas como el sobreajuste, dificultades en la interpretación del modelo y una mayor complejidad computacional. Por ello, reducir el número de variables explicativas, sin perder información relevante, se convierte en un paso crucial para construir modelos más eficientes y robustos.

Es especialmente importante reducir el número de variables explicativas en los estudios observacionales exploratorios, ya que en otros tipos de estudios, como los diseñados previamente, las variables incluidas suelen estar seleccionadas de antemano porque se conoce su relación con la variable respuesta o porque han sido identificadas como relevantes en investigaciones previas.

La **reducción de variables explicativas** busca simplificar el modelo al seleccionar un subconjunto de predictores que capturen la mayor parte de la información relevante de los datos. Este proceso puede realizarse a través de diferentes enfoques, dependiendo del contexto y de las características del conjunto de datos.

3.2.1 Motivaciones para reducir variables

Al limitar el número de predictores, no solo se simplifica el modelo, sino que también se optimizan diversos aspectos fundamentales en el análisis.

1. **Evitar el sobreajuste:**
    - Cuando hay demasiadas variables en relación al número de observaciones, el modelo puede ajustarse demasiado a los datos de entrenamiento y perder capacidad predictiva en nuevos conjuntos de datos.
2. **Mejorar la interpretabilidad:**
    - Un modelo con menos variables es más fácil de interpretar, lo que resulta fundamental en aplicaciones como ciencias sociales, biomedicina o economía.
3. **Reducción de complejidad computacional:**

- Al disminuir el número de variables, se reducen los costos de tiempo y memoria en el ajuste y evaluación del modelo.

4. **Manejo de multicolinealidad:**
   - La reducción puede eliminar variables redundantes que presentan una alta correlación entre sí, estabilizando las estimaciones del modelo.

3.2.2 Métodos de reducción de variables

Algunas de las ideas más comunes para tratar de reducir el número de variables de un modelo son:

1. **Selección de variables:**
   - Utiliza estrategias como selección directa, métodos automáticos (forward, backward o stepwise), o técnicas basadas en regularización (Lasso, Elastic Net) para seleccionar las variables más relevantes.
2. **Técnicas de transformación de datos:**
   - Se proyectan las variables explicativas en un nuevo espacio de menor dimensionalidad, manteniendo la mayor cantidad posible de información. Algunas de estas técnicas se estudian en la asignatura de Aprendizaje Automático:
     - **Análisis de Componentes Principales (PCA):** Reduce las variables explicativas a un conjunto de componentes ortogonales que explican la mayor parte de la varianza.
     - **Análisis de Factores:** Agrupa variables relacionadas en factores latentes que capturan la esencia de la información.
3. **Filtrado basado en información:**
   - Identifica y descarta variables con baja variabilidad o poca relación con la variable respuesta, utilizando métricas como la correlación o importancia estadística.
4. **Métodos de selección basados en modelos:**
   - Ajusta modelos iterativamente para evaluar la contribución de cada variable explicativa y descartar aquellas con menor relevancia según criterios como el p-valor, AIC o BIC.

> **!** Consideraciones importantes
>
> - La reducción de variables debe realizarse cuidadosamente para evitar la pérdida de información clave que pueda comprometer la calidad del modelo.
> - Es fundamental validar el modelo resultante, asegurándose de que mantenga su

capacidad predictiva mediante técnicas como validación cruzada.
- En algunos casos, la selección o transformación de variables puede implicar compromisos entre simplicidad e interpretabilidad.

3.3 Selección de variables

Simplificar un modelo eliminando variables irrelevantes es fundamental para mejorar su parsimonia y evitar el sobreajuste. Este objetivo puede lograrse mediante métodos de selección de variables, ya sean directos, automáticos o basados en regularización. Estas técnicas permiten identificar subconjuntos óptimos de predictores, optimizando tanto la simplicidad como la precisión del modelo. En particular, los métodos de regularización, como Ridge, Lasso y Elastic Net, introducen penalizaciones al modelo para controlar la complejidad y prevenir el sobreajuste, convirtiéndose en herramientas clave en el análisis de datos modernos.

Cuando se dispone de p variables explicativas, es posible construir hasta 2^p modelos diferentes considerando todas las combinaciones posibles de estas variables. Sin embargo, explorar de manera exhaustiva todos estos modelos puede ser inviable, especialmente si p es grande. Por ejemplo, con solo 10 variables regresoras, se generarían $2^{10} = 1024$ modelos posibles. Aunque la tecnología actual permite ajustar todos estos modelos, evaluar cada uno en términos de bondad de ajuste, gráficos de residuos, detección de observaciones influyentes y otros diagnósticos sería extremadamente complejo y costoso.

Para superar este desafío, se han desarrollado criterios específicos de selección de variables que ayudan a los analistas a identificar un pequeño subconjunto de modelos que cumplan con los estándares de calidad deseados. Este enfoque permite centrar el análisis en un grupo reducido de modelos "buenos", generalmente entre 4 y 6, y realizar un estudio más profundo y detallado de ellos. Esta estrategia facilita tanto la interpretación como la eficiencia del proceso analítico, optimizando el uso de recursos computacionales y asegurando que los resultados sean robustos y fiables.

3.4 Métodos de selección directa

Los métodos de selección directa son un enfoque fundamental en la búsqueda de un subconjunto óptimo de variables predictoras en modelos de regresión. Este enfoque evalúa de manera sistemática diferentes combinaciones de variables para identificar cuál de ellas proporciona el mejor ajuste al modelo en función de un criterio predefinido, como el coeficiente de determinación ajustado (R^2 ajustado), el error cuadrático medio (ECM) o criterios de información como AIC o BIC.

A diferencia de los métodos automáticos, los métodos de selección directa no dependen de un proceso iterativo de adición o eliminación de variables. En cambio, buscan exhaustivamente (o mediante aproximaciones computacionalmente más eficientes) entre todas las posibles combinaciones de variables, lo que garantiza un análisis completo de las interacciones y relevancias potenciales.

Estos métodos son especialmente útiles cuando el número de predictores no es demasiado grande, ya que el esfuerzo computacional crece exponencialmente con el número de variables. Aunque el costo computacional puede ser elevado en datasets amplios, los métodos de selección directa proporcionan una referencia sólida y transparente para evaluar qué variables son fundamentales en el modelo.

En esta sección, analizaremos los métodos de selección directa más comunes, su implementación práctica y las métricas utilizadas para comparar modelos, destacando sus ventajas y limitaciones.

3.5 Métodos automáticos

Los métodos automáticos de selección de variables son herramientas prácticas y eficientes diseñadas para identificar subconjuntos relevantes de predictores en un modelo de regresión. A diferencia de los métodos de selección directa, que exploran exhaustivamente todas las combinaciones posibles de variables, los métodos automáticos siguen un enfoque iterativo que simplifica el proceso de selección. Estos métodos son especialmente útiles en situaciones donde el número de predictores es elevado, ya que reducen significativamente el esfuerzo computacional.

El principio clave detrás de los métodos automáticos es el ajuste dinámico del conjunto de variables en función de criterios estadísticos, como p-valores, coeficientes de determinación ajustados (R^2) ajustado), o criterios de información como AIC y BIC. Entre las estrategias más comunes se encuentran:

- **Método Forward (selección progresiva):** Parte de un modelo vacío e incorpora variables de manera secuencial, añadiendo en cada paso la variable que mejora más el modelo.
- **Método Backward (eliminación regresiva):** Comienza con todas las variables en el modelo y elimina iterativamente aquellas que tienen menor impacto.
- **Método Stepwise:** Combina las estrategias forward y backward, permitiendo tanto la inclusión como la exclusión de variables en cada iteración.

Estos métodos ofrecen una manera estructurada y ágil de seleccionar variables, aunque no garantizan encontrar el mejor modelo global debido a su naturaleza secuencial. A lo largo de esta sección, examinaremos cada uno de estos métodos, sus ventajas, limitaciones y aplicaciones en diferentes contextos de análisis.

3.6 Métodos basados en regularización

En los modelos de regresión, especialmente cuando se trabaja con un gran número de variables predictoras o con datos multicolineales, los métodos tradicionales de selección de variables pueden resultar ineficaces o inestables. En estos casos, los métodos basados en regularización surgen como una alternativa poderosa que no solo selecciona variables, sino que también mejora la estabilidad y la precisión del modelo.

La regularización consiste en introducir una penalización en la función de ajuste del modelo, lo que tiene dos efectos principales: controlar el sobreajuste al reducir la complejidad del modelo y forzar la selección de un subconjunto más parsimonioso de predictores. Estas penalizaciones ajustan los coeficientes de las variables predictoras, favoreciendo soluciones más simples y robustas (James et al. 2013).

Entre los métodos de regularización más destacados se encuentran:

- **Ridge Regression:** Aplica una penalización proporcional al cuadrado de los coeficientes, lo que permite manejar problemas de multicolinealidad pero no conduce a la eliminación completa de variables.
- **Lasso (Least Absolute Shrinkage and Selection Operator):** Introduce una penalización basada en el valor absoluto de los coeficientes, lo que no solo reduce su magnitud, sino que también puede anularlos completamente, realizando una selección automática de variables.
- **Elastic Net:** Combina las penalizaciones de Ridge y Lasso, ofreciendo mayor flexibilidad en situaciones donde hay una gran correlación entre los predictores.

Estos métodos son especialmente útiles en problemas donde el número de variables predictoras excede el número de observaciones, o cuando se desea un modelo más interpretable. En esta sección, exploraremos en detalle los fundamentos teóricos, la implementación práctica y las aplicaciones de cada uno de estos métodos, destacando sus ventajas en escenarios complejos y desafiantes.

3.6.1 Ridge regression

La regresión Ridge introduce una penalización en la estimación de los coeficientes de regresión, lo que ayuda a reducir la varianza del modelo y mejora su capacidad predictiva en presencia de datos altamente correlacionados o con muchas variables (Marquardt and Snee 1975). El modelo de regresión Ridge es una extensión de la regresión lineal estándar. Dado un conjunto de datos con n observaciones y p predictores, expresamos el modelo de regresión lineal múltiple como:

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon$$

donde:

- $\mathbf{Y}$ es el vector de respuesta de dimensión $n \times 1$.
- $\mathbf{X}$ es la matriz de diseño de dimensión $n \times p$.
- β es el vector de coeficientes de regresión de dimensión $p \times 1$.
- ε es el vector de errores aleatorios.

En mínimos cuadrados ordinarios (OLS), los coeficientes se estiman minimizando la suma de los errores al cuadrado:

$$SSE = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 = \|\mathbf{Y} - \mathbf{X}\beta\|^2.$$

Sin embargo, cuando hay multicolinealidad, la matriz $X^T X$ puede ser casi singular, generando coeficientes inestables. Para evitar esto, la regresión Ridge añade un **término de penalización** λ, de la siguiente manera:

$$SSE_{ridge} = \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda \sum_{j=1}^{p} \beta_j^2.$$

Este término adicional, es un **término de penalización** ($L_2 = \sum \beta_j^2$) impone una restricción sobre los coeficientes, evitando que tomen valores excesivamente grandes. La estimación de β en Ridge se obtiene resolviendo:

$$\hat{\beta}_{ridge} = (\mathbf{X}^T \mathbf{X} + \lambda I)^{-1} \mathbf{X}^T \mathbf{Y}.$$

donde I es la matriz identidad y $\lambda \geq 0$ es un hiperparámetro que controla la cantidad de penalización aplicada.

Interpretación del parámetro λ

- Si $\lambda = 0$, el modelo Ridge es equivalente a la regresión lineal tradicional (OLS).
- A medida que λ aumenta, los coeficientes β_j se reducen en magnitud, lo que ayuda a controlar la varianza del modelo y a prevenir el sobreajuste.
- Si λ es demasiado grande, los coeficientes se acercan a cero y el modelo puede perder interpretabilidad.

La elección óptima de λ se determina generalmente mediante **validación cruzada**.

> **Aviso**
>
> Los detalles de la validación cruzada son tratados en la asignatura de Aprendizaje Automático.

Propiedades Clave

- **Manejo de la multicolinealidad:** La regularización reduce la sensibilidad del modelo cuando los predictores están altamente correlacionados.
- **Menor varianza en las predicciones:** El modelo Ridge tiende a ser más estable en comparación con OLS, lo que mejora la capacidad de generalización en conjuntos de datos nuevos.
- **No realiza selección de variables:** A diferencia de Lasso, Ridge **no anula coeficientes**, sino que reduce su magnitud. Esto es útil cuando se sospecha que todas las variables tienen algún grado de importancia en el modelo.

Ejemplo

```
# Cargar librerías
library(glmnet)
```

```
Loading required package: Matrix
```

```
Loaded glmnet 4.1-8
```

```
# Datos simulados
set.seed(123)
X <- matrix(rnorm(100 * 10), 100, 10)  # 100 observaciones, 10 predictores
Y <- X %*% rnorm(10) + rnorm(100)  # Variable de respuesta con ruido

# Ajustar modelo Ridge
modelo_ridge <- glmnet(X, Y, alpha = 0)  # alpha = 0 indica regresión Ridge

# Seleccionar lambda óptimo con validación cruzada
cv_ridge <- cv.glmnet(X, Y, alpha = 0)
lambda_optimo <- cv_ridge$lambda.min  # Mejor valor de lambda

print(lambda_optimo)
```

```
[1] 0.2583753
```

```
# Ajustar modelo final con lambda óptimo
modelo_ridge_final <- glmnet(X, Y, alpha = 0, lambda = lambda_optimo)

modelo_ridge_final
```

```
Call:  glmnet(x = X, y = Y, alpha = 0, lambda = lambda_optimo)

  Df  %Dev Lambda
1 10 93.55 0.2584
```

```r
# Comparación modelo clásico

modelo_lm <- lm(Y~X)

# Mostrar coeficientes
output=cbind(round(coef(modelo_ridge_final),3),
             round(coef(modelo_lm),3))

colnames(output)=c("RIDGE","OLS")

output
```

```
11 x 2 sparse Matrix of class "dgCMatrix"
              RIDGE    OLS
(Intercept)  0.118  0.132
V1          -0.874 -0.995
V2          -1.019 -1.131
V3           0.040  0.039
V4           0.002  0.001
V5          -2.500 -2.703
V6           1.001  1.104
V7           0.247  0.274
V8           2.125  2.244
V9           0.635  0.658
V10         -0.390 -0.427
```

La regresión Ridge es una técnica poderosa para mejorar la estabilidad de los modelos de regresión en presencia de multicolinealidad. A diferencia de OLS, que puede generar coeficientes inestables, Ridge introduce una penalización que reduce la magnitud de los coeficientes, evitando valores extremos. Aunque Ridge no realiza selección de variables, su capacidad para reducir la varianza y mejorar la capacidad predictiva lo convierte en una herramienta esencial en el análisis de datos modernos.

En la siguiente sección, exploraremos la **regresión Lasso**, que extiende este concepto permitiendo la eliminación de variables irrelevantes del modelo.

3.6.2 Regresión Lasso

Cuando se tiene un conjunto de predictores con posibles redundancias o ruido, Lasso permite identificar cuáles son las variables más relevantes para el modelo, lo que facilita la interpretación y reduce la complejidad del análisis.

Al igual que ocurría en Ridge Regression, el modelo de regresión Lasso se basa en la minimización de la siguiente función de error (Ranstam and Cook 2018):

$$SSE_{lasso} = \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

donde el **término de penalización**, ($L_1 = \sum |\beta_j|$) hace que algunos coeficientes se reduzcan exactamente a **cero**, lo que significa que esas variables son eliminadas del modelo.

La diferencia clave con **Ridge Regressión**, visto anteriormente, es que Ridge reduce la magnitud de los coeficientes pero no los anula, mientras que **Lasso puede eliminar variables por completo**.

Interpretación del parámetro λ

- Si $\lambda = 0$, el modelo es equivalente a la regresión lineal tradicional (OLS).
- A medida que λ aumenta, más coeficientes se reducen a cero, lo que equivale a realizar **selección de variables**.
- Si λ es demasiado grande, se eliminan demasiadas variables, lo que puede resultar en un modelo subóptimo.

Al igual que en el método *Risge*, la selección óptima de λ se realiza generalmente mediante **validación cruzada**.

Propiedades Clave

- **Selección de variables automática:** Lasso no solo regulariza, sino que también selecciona las variables más importantes eliminando aquellas menos relevantes.
- **Manejo de la multicolinealidad:** Puede mejorar la interpretación del modelo cuando hay muchas variables correlacionadas.
- **Simplicidad y interpretabilidad:** Un modelo con menos variables es más fácil de interpretar y aplicar en la práctica.

- **Reduce el sobreajuste:** La penalización L_1 evita que el modelo se ajuste demasiado a los datos de entrenamiento, mejorando su capacidad predictiva en datos nuevos.

Ejemplo

```
# Ajustar modelo Lasso
modelo_lasso <- glmnet(X, Y, alpha = 1)  # alpha = 1 indica regresión Lasso

# Seleccionar lambda óptimo con validación cruzada
cv_lasso <- cv.glmnet(X, Y, alpha = 1)
lambda_optimo <- cv_lasso$lambda.min  # Mejor valor de lambda

print(lambda_optimo)
```

```
[1] 0.03260326
```

```
# Ajustar modelo final con lambda óptimo
modelo_lasso_final <- glmnet(X, Y, alpha = 1, lambda = lambda_optimo)


# Mostrar coeficientes
output=cbind(round(coef(modelo_lasso_final),3),output)

colnames(output)=c("LASSO","RIDGE","OLS")

output
```

```
11 x 3 sparse Matrix of class "dgCMatrix"
             LASSO  RIDGE    OLS
(Intercept)  0.131  0.118  0.132
V1          -0.950 -0.874 -0.995
V2          -1.078 -1.019 -1.131
V3           0.006  0.040  0.039
V4           .      0.002  0.001
V5          -2.652 -2.500 -2.703
V6           1.058  1.001  1.104
V7           0.235  0.247  0.274
V8           2.213  2.125  2.244
V9           0.629  0.635  0.658
V10         -0.392 -0.390 -0.427
```

Consideraciones Importantes

La regresión Lasso es una poderosa técnica de regularización que no solo mejora la estabilidad del modelo en presencia de muchas variables predictoras, sino que también realiza una selección automática de las más relevantes. Su capacidad para reducir coeficientes a cero la convierte en una herramienta esencial en el análisis de datos de alta dimensión.

- **Lasso puede eliminar demasiadas variables si** λ es demasiado grande, lo que puede llevar a la pérdida de información importante.
- **No maneja bien grupos de predictores altamente correlacionados**, ya que selecciona solo uno de ellos y elimina los demás.
- **Elastic Net**, que combina Ridge y Lasso, puede ser una mejor opción cuando hay **multicolinealidad fuerte** en los datos.

En la siguiente sección, exploraremos **Elastic Net**, una técnica híbrida que combina las ventajas de Ridge y Lasso para mejorar la selección de variables en presencia de predictores altamente correlacionados.

3.6.3 Elastic Net

La regresión **Elastic Net** es una técnica de regularización que combina las propiedades de **Ridge** y **Lasso**, abordando algunas de sus limitaciones individuales (Zou and Hastie 2005). Mientras que Ridge es útil para manejar la multicolinealidad sin eliminar variables y Lasso selecciona un subconjunto de predictores, Elastic Net equilibra ambos enfoques permitiendo la selección de variables en presencia de alta correlación entre los predictores.

Este método es particularmente efectivo cuando el número de predictores es grande y existe **multicolinealidad**, ya que permite controlar simultáneamente la **reducción de la magnitud de los coeficientes** y la **eliminación de variables irrelevantes**.

Elastic Net introduce una penalización que combina los términos de Ridge (L_2) y Lasso (L_1):

$$SSE_{\text{Elastic Net}} = \|Y - X\beta\|^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2$$

donde:

- λ_1 (asociado a Lasso) controla la cantidad de coeficientes que se reducen a **cero**.
- λ_2 (asociado a Ridge) controla la **reducción de magnitud** de los coeficientes sin anularlos.
- α es un parámetro adicional que pondera la combinación entre Lasso y Ridge, con:
  - $\alpha = 1 \rightarrow$ Elastic Net se comporta como Lasso.
  - $\alpha = 0 \rightarrow$ Elastic Net se comporta como Ridge.

- $0 < \alpha < 1 \rightarrow$ Elastic Net combina ambos métodos.

La estimación de los coeficientes en Elastic Net se obtiene resolviendo:

$$\hat{\beta}_{\text{Elastic Net}} = \arg\min_{\beta} \left(\|Y - X\beta\|^2 + \lambda \left(\alpha \sum |\beta_j| + (1-\alpha) \sum \beta_j^2 \right) \right)$$

Propiedades Clave

- **Manejo de la Multicolinealidad:** A diferencia de Lasso, que selecciona solo una de las variables correlacionadas y elimina las demás, Elastic Net distribuye la penalización entre todas las variables correlacionadas, evitando una selección arbitraria.
- **Selección de variables más estable:** La combinación de Lasso y Ridge permite una selección más robusta, manteniendo información relevante del modelo sin eliminar predictores clave.
- **Mejora del rendimiento predictivo:** Al utilizar validación cruzada para seleccionar los hiperparámetros λ_1, λ_2 y α, se optimiza la capacidad del modelo para generalizar a nuevos datos.

Ejemplo

```r
# Ajustar modelo Elastic Net
modelo_elastic_net <- glmnet(X, Y, alpha = 0.5)  # Alpha = 0.5 (50% Ridge, 50% Lasso)

# Seleccionar lambda óptimo con validación cruzada
cv_elastic_net <- cv.glmnet(X, Y, alpha = 0.5)
lambda_optimo <- cv_elastic_net$lambda.min  # Mejor valor de lambda

print(lambda_optimo)
```

```
[1] 0.0213522
```

```r
# Ajustar modelo final con lambda óptimo
modelo_elastic_final <- glmnet(X, Y, alpha = 0.5, lambda = lambda_optimo)

# Mostrar coeficientes
output=cbind(round(coef(modelo_elastic_final),3),output)

colnames(output)=c("ELASTIC","LASSO","RIDGE","OLS")

output
```

```
11 x 4 sparse Matrix of class "dgCMatrix"
            ELASTIC  LASSO  RIDGE    OLS
(Intercept)   0.131  0.131  0.118  0.132
V1           -0.975 -0.950 -0.874 -0.995
V2           -1.108 -1.078 -1.019 -1.131
V3            0.028  0.006  0.040  0.039
V4            .      .      0.002  0.001
V5           -2.677 -2.652 -2.500 -2.703
V6            1.084  1.058  1.001  1.104
V7            0.260  0.235  0.247  0.274
V8            2.229  2.213  2.125  2.244
V9            0.647  0.629  0.635  0.658
V10          -0.414 -0.392 -0.390 -0.427
```

Para determinar el mejor valor de α, se usa **validación cruzada** probando distintos valores entre 0 y 1. Algunas estrategias comunes incluyen:

- **Si hay muchas variables irrelevantes**, se recomienda α cercano a 1 (Lasso).
- **Si hay fuerte multicolinealidad**, se recomienda α cercano a 0 (Ridge).
- **Si se desea un balance entre selección y estabilidad**, se suele usar $\alpha = 0.5$.

La regresión Elastic Net combina lo mejor de Ridge y Lasso, ofreciendo un método de regularización robusto para modelos con muchas variables predictoras y posible multicolinealidad. Su capacidad para seleccionar variables sin eliminar información clave lo convierte en una opción ideal para modelos complejos y de alta dimensionalidad.

3.6.4 Comparación de los métodos de Regularización

Método	Penalización	Efecto sobre los coeficientes
OLS	Ninguna	Sin restricción, puede haber multicolinealidad
Ridge	L_2	Reduce la magnitud de los coeficientes, pero no los anula
Lasso	L_1	Puede anular coeficientes, permitiendo selección de variables
Elastic Net	$L_1 + L_2$	Combinación de Ridge y Lasso

Lasso es especialmente útil cuando se sospecha que muchas variables son irrelevantes, mientras que Ridge es preferido cuando se espera que todas las variables aporten información al modelo.

Elastic Net es ideal cuando hay **muchas variables correlacionadas** y se desea un modelo **estable y parsimonioso**.

- Elastic Net mejora la estabilidad del modelo en comparación con Lasso, especialmente cuando hay variables predictoras altamente correlacionadas.
- Es más flexible que Ridge y Lasso individualmente, permitiendo un ajuste más fino a distintos tipos de problemas.
- Requiere la selección de hiperparámetros (λ y α), por lo que debe usarse validación cruzada para encontrar la combinación óptima.

---

4 Modelos no lineales. Transformación de variables. Ingeniería de características.

En el análisis de datos, muchas relaciones entre variables no pueden ser capturadas adecuadamente mediante modelos de regresión lineal. Aunque la regresión lineal es una herramienta poderosa por su simplicidad e interpretabilidad, existen numerosos escenarios donde las relaciones entre las variables son inherentemente **no lineales**.

Limitaciones de la regresión lineal:

- Relaciones no lineales: En muchos casos, la relación entre las variables no es proporcional ni constante. Por ejemplo, el crecimiento poblacional o la desintegración radiactiva siguen patrones exponenciales.
- Interacciones no consideradas: La regresión lineal simple no capta interacciones entre variables a menos que se incluyan explícitamente.
- Sensibilidad a valores atípicos: Los modelos lineales pueden verse influenciados por outliers, afectando la precisión del modelo.
- Supuestos estrictos: La regresión lineal asume homocedasticidad, normalidad de los errores e independencia, lo cual no siempre se cumple en la práctica.

Estas limitaciones abren la puerta a la necesidad de modelos más flexibles (**modelos no lineales**) que puedan capturar relaciones complejas entre las variables.

Además de utilizar modelos no lineales, otra estrategia fundamental es la **transformación de variables**. Mediante transformaciones matemáticas (como logaritmos, potencias o funciones exponenciales), es posible linearizar relaciones no lineales o mejorar la adecuación del modelo. Estas transformaciones pueden también ayudar a cumplir con los supuestos de normalidad y homocedasticidad en los modelos.

Finalmente, la **ingeniería de características** juega un papel crucial en la mejora del rendimiento de los modelos predictivos. Este proceso implica la creación, modificación o combinación de variables explicativas para extraer mayor información de los datos. La ingeniería de características es clave para mejorar la capacidad predictiva de los modelos, especialmente en entornos de alta dimensionalidad o con datos complejos.

A lo largo de este tema, exploraremos estos tres pilares: cómo identificar y ajustar modelos no lineales, cómo aplicar transformaciones efectivas a las variables y cómo desarrollar nuevas características que potencien el análisis y la predicción.

4.1 Modelos no lineales

Los modelos de regresión no lineal son herramientas esenciales cuando las relaciones entre las variables no pueden capturarse adecuadamente con modelos lineales. La regresión polinómica, exponencial, logarítmica y los modelos por tramos ofrecen diferentes enfoques para representar patrones complejos en los datos. Comprender cuándo y cómo aplicar estos modelos es fundamental para mejorar la precisión y la interpretabilidad en el análisis de datos.

4.1.1 Regresión Polinómica

La **regresión polinómica** es una extensión de la regresión lineal que permite capturar relaciones no lineales mediante la inclusión de términos polinómicos (cuadráticos, cúbicos, etc.). Aunque sigue siendo un modelo lineal en los parámetros, la inclusión de potencias de las variables independientes permite ajustar curvas en lugar de líneas rectas.

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \cdots + \beta_k X^k + \varepsilon$$

Donde k es el grado del polinomio. A medida que aumenta el grado, el modelo se vuelve más flexible y puede ajustarse a relaciones más complejas.

Este tipo de modelos son capaces de capturar curvaturas suaves en los datos. Se trata de modelos fáciles de implementar y comprender, aunque la interpretación de los coeficientes asociados a altos grados del polinomio puede ser compleja. De hecho, exite un claro riesgo de sobreajuste cuando se utilizan polinomios de alto grado.

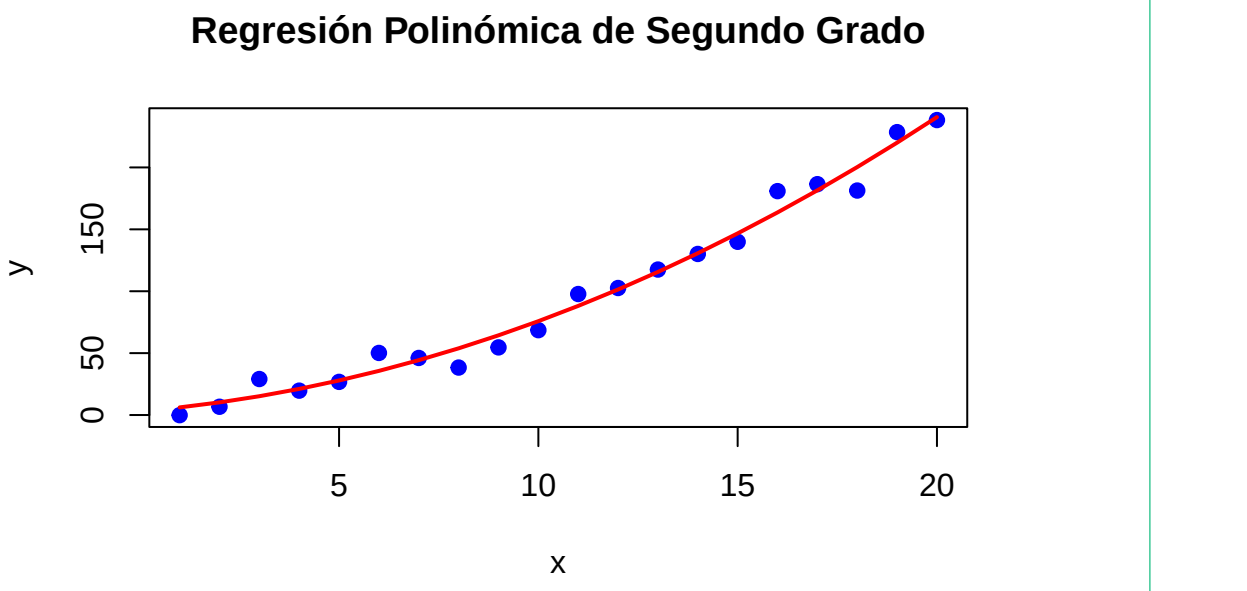
Ejemplo

```r
# Datos simulados
set.seed(123)
x <- 1:20
y <- 3 + 2 * x + 0.5 * x^2 + rnorm(20, mean = 0, sd = 10)

# Ajuste del modelo polinómico de grado 2
modelo_polinomico <- lm(y ~ poly(x, 2))

summary(modelo_polinomico)
```

```
Call:
lm(formula = y ~ poly(x, 2))
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-18.9643  -6.4011  -0.8541   5.8504  17.2160

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    97.17       2.27  42.796  < 2e-16 ***
poly(x, 2)1   318.21      10.15  31.339  < 2e-16 ***
poly(x, 2)2    60.98      10.15   6.006 1.42e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.15 on 17 degrees of freedom
Multiple R-squared:  0.9836,    Adjusted R-squared:  0.9816
F-statistic: 509.1 on 2 and 17 DF,  p-value: 6.778e-16
```

```r
# Visualización
plot(x, y, main = "Regresión Polinómica de Segundo Grado", pch = 19, col = "blue")
lines(x, predict(modelo_polinomico), col = "red", lwd = 2)
```



4.1.2 Modelos de Regresión Exponencial y Logarítmica

Cuando la relación entre la variable dependiente y la independiente sigue un **crecimiento o decaimiento exponencial**, o una relación logarítmica, los modelos lineales tradicionales no son suficientes. En estos casos, se pueden utilizar transformaciones exponenciales o logarítmicas.

Regresión Exponencial

Este modelo es útil cuando la variable dependiente crece (o decrece) a una tasa proporcional a su valor actual.

$$Y = \beta_0 e^{\beta_1 X} + \varepsilon$$

Este modelo puede **linearizarse** tomando el logaritmo de la variable dependiente:

$$\log(Y) = \log(\beta_0) + \beta_1 X + \varepsilon$$

Regresión Logarítmica

Útil cuando la tasa de cambio de la variable dependiente disminuye a medida que aumenta la variable independiente.

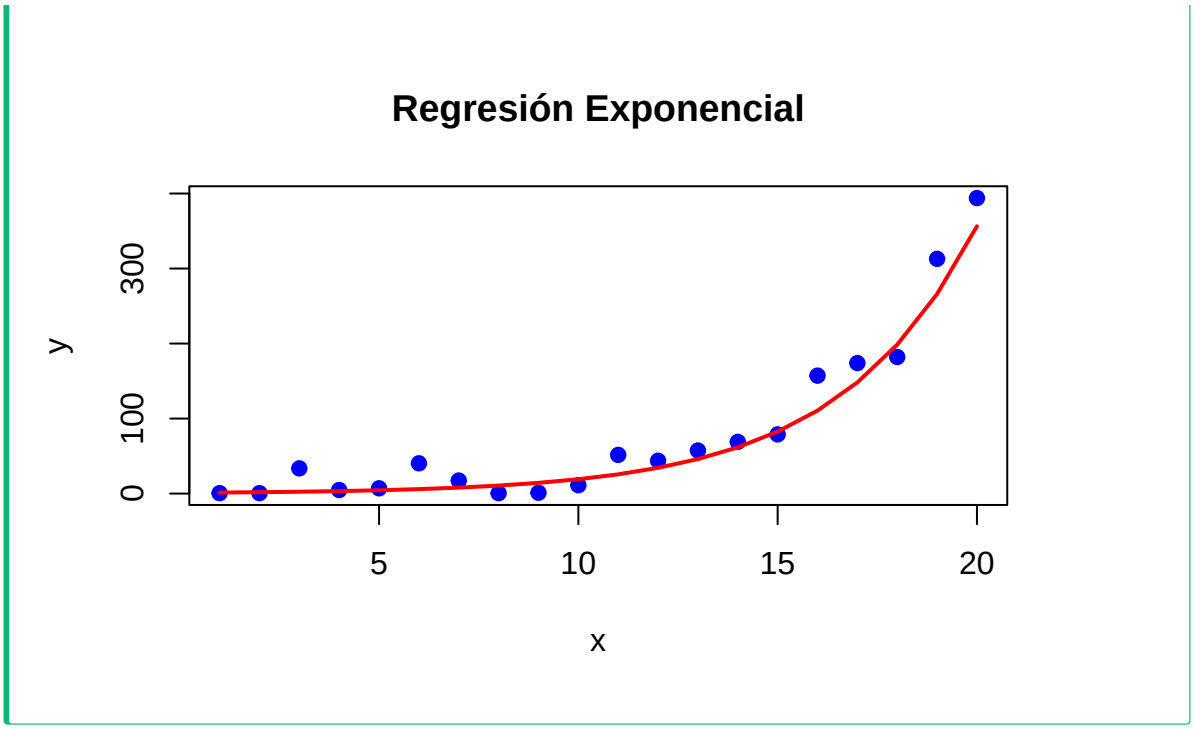
$$Y = \beta_0 + \beta_1 \log(X) + \varepsilon$$

Ejemplo

```r
# Datos simulados para un modelo exponencial
set.seed(123)
x <- 1:20
y <- exp(0.3 * x) + rnorm(20, mean = 0, sd = 20)

# Asegurarse de que todos los valores de 'y' sean positivos para aplicar logaritmo
y[y <= 0] <- min(y[y > 0]) * 0.5  # Reemplaza valores no positivos por un valor pequeño pos

# Ajuste del modelo exponencial (transformación logarítmica)
modelo_exponencial <- lm(log(y) ~ x)

# Resumen del modelo
summary(modelo_exponencial)
```

```
Call:
lm(formula = log(y) ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-2.9269 -0.1997  0.1612  0.3837  2.6104

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.02791    0.59598   0.047    0.963
x            0.29240    0.04975   5.877 1.45e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.283 on 18 degrees of freedom
Multiple R-squared:  0.6574,    Adjusted R-squared:  0.6384
F-statistic: 34.54 on 1 and 18 DF,  p-value: 1.45e-05
```

```r
# Visualización
plot(x, y, main = "Regresión Exponencial", pch = 19, col = "blue", ylab = "y", xlab = "x")

# Predicciones para los mismos valores de x
predicciones <- predict(modelo_exponencial, newdata = data.frame(x = x))

# Convertir predicciones a la escala original (exponencial inverso del log)
lines(x, exp(predicciones), col = "red", lwd = 2)
```



4.1.3 Regresión Spline y modelos basados en Segmentos

Los **splines** y los **modelos segmentados** son técnicas que permiten ajustar relaciones no lineales mediante la división de los datos en segmentos y el ajuste de funciones lineales o polinómicas en cada segmento. Estos métodos son especialmente útiles cuando la relación entre las variables cambia en diferentes rangos de los datos.

Modelos por tramos (Piecewise Regression)

En este enfoque, se ajustan diferentes regresiones lineales a distintos rangos de la variable independiente. A diferencia de los splines, las transiciones entre segmentos no necesariamente son suaves.

Estos modelos permiten capturar relaciones complejas con menor riesgo de sobreajuste en comparación con polinomios de alto grado.

Splines

Los **splines** son una poderosa herramienta en el análisis de regresión para modelar relaciones no lineales entre variables. A diferencia de los modelos polinómicos tradicionales, que ajustan un solo polinomio a todos los datos, los splines permiten dividir el rango de la variable independiente en diferentes **tramos** y ajustar polinomios separados en cada uno de ellos. Esto proporciona mayor flexibilidad para capturar patrones complejos en los datos, sin los problemas de inestabilidad y sobreajuste que pueden surgir al utilizar polinomios de alto grado.

Un **spline** es una función que está compuesta por múltiples **polinomios por tramos** que se ajustan en diferentes intervalos del dominio de la variable independiente. Estos polinomios están conectados en puntos específicos llamados **nudos** (*knots*), que marcan el final de un tramo y el inicio de otro. La principal característica de los splines es que estos polinomios están diseñados para unirse de manera **suave** en los nudos, asegurando que la función resultante sea continua y, en muchos casos, que sus derivadas también sean continuas.

Elementos clave

- **Tramos:** Intervalos del dominio de la variable independiente en los que se ajusta un polinomio distinto.
- **Nudos:** Puntos donde los tramos se conectan. Los nudos definen la estructura del spline y determinan dónde la función puede cambiar de forma.
- **Continuidad:** Los splines están construidos para que no haya saltos abruptos en la función o en sus derivadas en los nudos. Por ejemplo, un spline cúbico asegura continuidad en la función, la primera derivada (pendiente) y la segunda derivada (curvatura).

Tipos de Splines

Splines Lineales:

- Se ajustan líneas rectas entre los nudos.
- Garantizan la continuidad en los puntos de unión, pero no necesariamente en la pendiente.
- Son simples, pero pueden generar ángulos abruptos en los nudos.

Splines Cuadráticos:

- Se utilizan polinomios de segundo grado en cada tramo.
- Aseguran continuidad en la función y en la pendiente, pero no en la curvatura.

Splines Cúbicos:

- Los **splines cúbicos** son los más utilizados en análisis de regresión.
- Utilizan polinomios de tercer grado en cada tramo.
- Garantizan suavidad en la función y en sus primeras dos derivadas, lo que significa que la función es continua, su pendiente es continua y la curvatura es suave.
- Evitan el sobreajuste que puede ocurrir con polinomios de alto grado, proporcionando un ajuste flexible sin perder estabilidad.
- A diferencia de los polinomios globales de alto grado, los splines cúbicos pueden capturar patrones complejos sin oscilar de manera excesiva entre los puntos de datos.

- Los splines permiten que el modelo se adapte localmente a diferentes patrones en distintos tramos del dominio de la variable independiente.

Splines Naturales:

- Son una variante de los splines cúbicos que imponen condiciones adicionales en los extremos del rango de los datos, forzando la segunda derivada a ser cero en los extremos. Esto ayuda a evitar oscilaciones no deseadas fuera del rango de los datos.

El uso de splines en regresión permite modelar relaciones no lineales de manera flexible. La elección del número y la ubicación de los **nudos** es un aspecto fundamental del ajuste con splines:

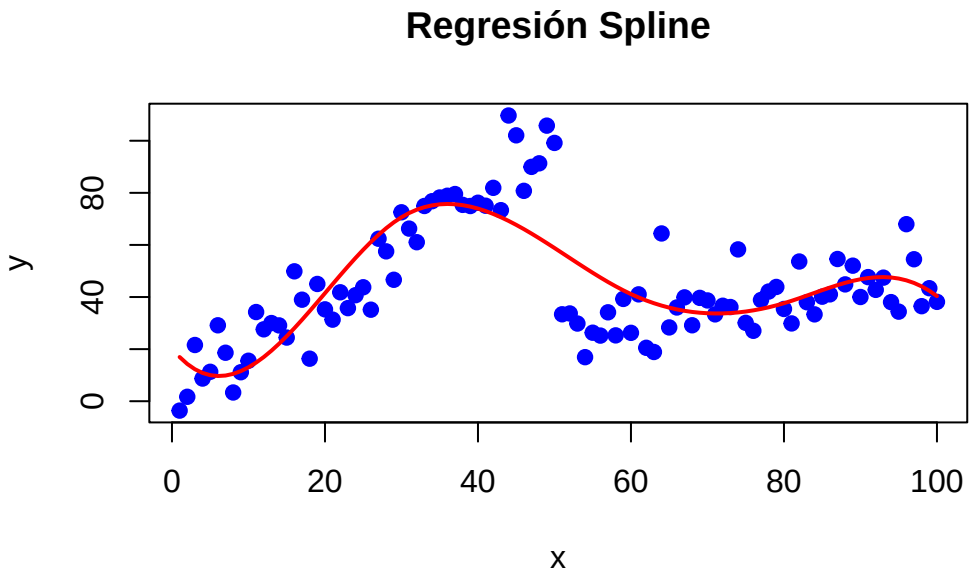
- Número de nudos: Demasiados nudos pueden llevar a un sobreajuste, mientras que muy pocos pueden no capturar adecuadamente la relación entre las variables. El uso de técnicas de validación cruzada puede ayudar a encontrar el equilibrio adecuado.
- Ubicación de los nudos: Los nudos pueden colocarse en puntos equidistantes, en cuantiles de la variable independiente, o en puntos donde se sospecha que la relación entre las variables cambia. La colocación de nudos es clave para obtener un buen ajuste. Los nudos pueden seleccionarse de manera automática (por ejemplo, en los cuantiles de la variable independiente) o manualmente según el conocimiento del problema.

Aviso

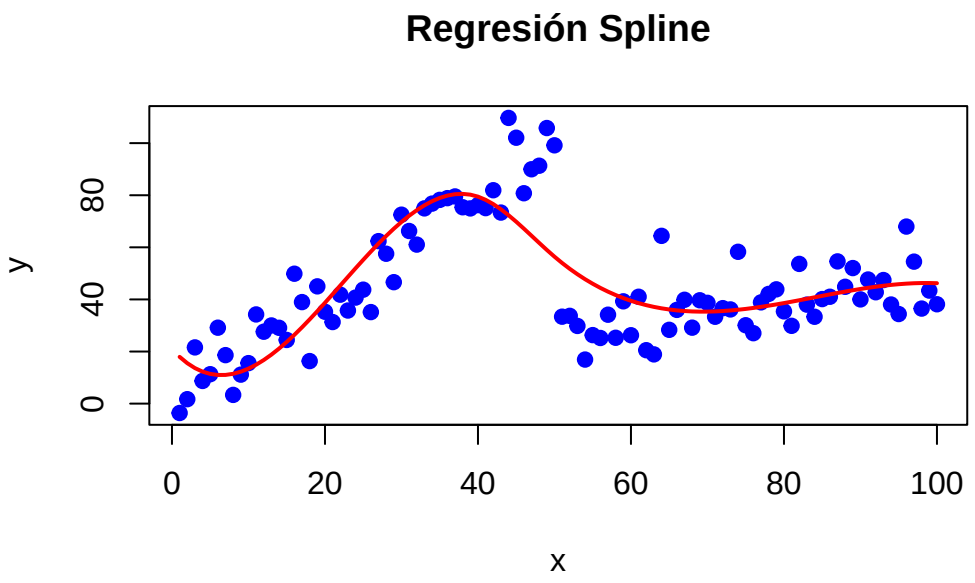
A diferencia de la regresión lineal simple, los coeficientes de los splines no tienen una interpretación directa. El enfoque se centra en la forma general del ajuste en lugar de en el valor de los coeficientes individuales.

Ejemplo

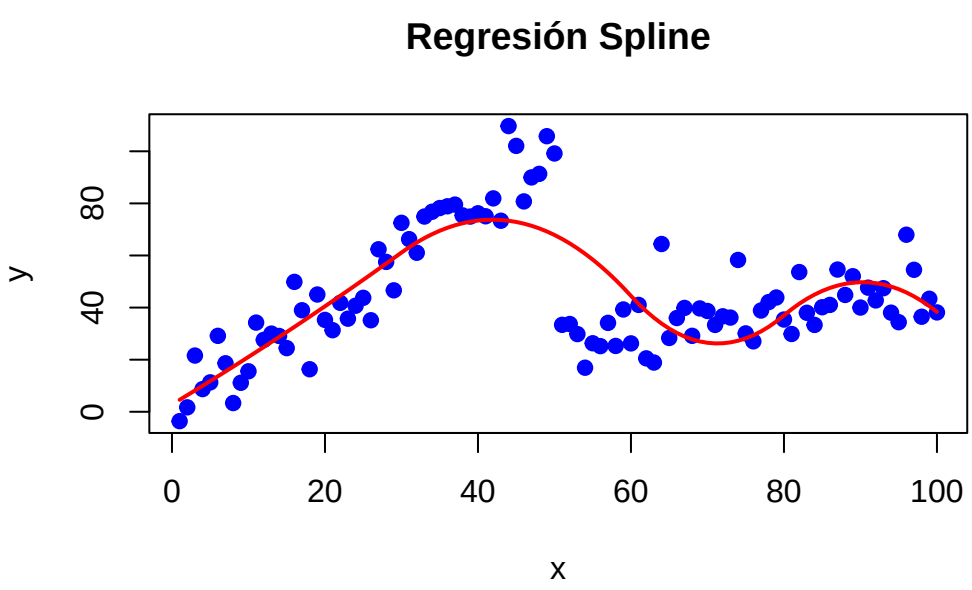
```r
# Cargar librería para splines
library(splines)

# Datos simulados
set.seed(123)
x <- seq(1, 100, by = 1)
y <- ifelse(x <= 50, 2 * x + rnorm(100, 0, 10), 0.5 * x + rnorm(100, 0, 10))

# Ajuste del modelo spline
modelo_spline <- lm(y ~ bs(x, knots = c(30, 60, 80)))

# Visualización
plot(x, y, main = "Regresión Spline", pch = 19, col = "blue")
lines(x, predict(modelo_spline), col = "red", lwd = 2)
```



```r
# Cambio en la posición de los nodos
# Ajuste del modelo spline
modelo_spline <- lm(y ~ bs(x, knots = c(40, 50)))

# Visualización
plot(x, y, main = "Regresión Spline", pch = 19, col = "blue")
lines(x, predict(modelo_spline), col = "red", lwd = 2)
```




```r
# Ajustamos un spline cuadrático

# Ajuste del modelo spline
modelo_spline <- lm(y ~ bs(x, degree=2, knots = c(30, 60, 80)))

# Visualización
plot(x, y, main = "Regresión Spline", pch = 19, col = "blue")
lines(x, predict(modelo_spline), col = "red", lwd = 2)
```



4.2 Transformación de variables

En el análisis de datos y la construcción de modelos estadísticos, no siempre es posible capturar adecuadamente la relación entre las variables independientes y la variable dependiente utilizando modelos lineales en su forma original. Aquí es donde entran en juego las **transformaciones de variables**, que permiten modificar la estructura de los datos para mejorar el ajuste del modelo, cumplir con los supuestos de la regresión y, en muchos casos, facilitar la interpretación.

Las transformaciones de variables son una herramienta fundamental para mejorar el rendimiento y la precisión de los modelos estadísticos. Estas transformaciones pueden aplicarse tanto a la variable dependiente como a las variables independientes.

i Objetivos

Linearizar relaciones no lineales:
Muchas relaciones entre variables no son lineales en su forma original. Aplicar una transformación adecuada puede **convertir una relación no lineal en lineal**, permitiendo el uso de técnicas de regresión lineal. Por ejemplo, una relación exponencial

$$Y = \beta_0 e^{\beta_1 X}$$

puede linearizarse tomando el logaritmo de Y:

$$\log(Y) = \log(\beta_0) + \beta_1 X$$

Corregir problemas de heterocedasticidad:
La regresión lineal asume que los errores tienen **varianza constante** (homocedasticidad). Sin embargo, en la práctica, es común encontrar datos con **heterocedasticidad** (la varianza de los errores cambia con el nivel de la variable independiente). Las transformaciones pueden ayudar a estabilizar la varianza. Por ejemplo, transformar la variable dependiente Y usando un logaritmo o una raíz cuadrada puede reducir la heterocedasticidad.
Normalizar la distribución de los errores:
La regresión lineal también asume que los errores están **normalmente distribuidos**. Las transformaciones pueden ayudar a que los residuos del modelo se ajusten mejor a una distribución normal, lo que mejora la validez de los intervalos de confianza y las pruebas de hipótesis.
Reducir la influencia de valores atípicos:
Algunas transformaciones pueden disminuir la influencia de los **valores atípicos** en el modelo, haciendo que el ajuste sea más robusto.
Mejorar la interpretabilidad del modelo:
Aunque algunas transformaciones pueden complicar la interpretación directa de los coeficientes, otras pueden facilitar el entendimiento de la relación entre variables (por ejemplo, tasas de crecimiento constantes).

4.2.1 Tipos de transformaciones comunes

Existen diversas transformaciones que pueden aplicarse a los datos según el problema que se desea abordar. A continuación, se describen las transformaciones más utilizadas en el análisis de regresión.

Transformación Logarítmica (log)

Se emplea para linearizar relaciones exponenciales, reducir la heterocedasticidad, y estabilizar la varianza: -

$$Y = \log(Y)$$

o

$$X = \log(X)$$

Es una transformación adecuada cuando la variable tiene una distribución sesgada a la derecha o cuando el efecto marginal disminuye con el valor de la variable (Ingresos, crecimiento poblacional, y tasas de interés, etc).

Ejemplo

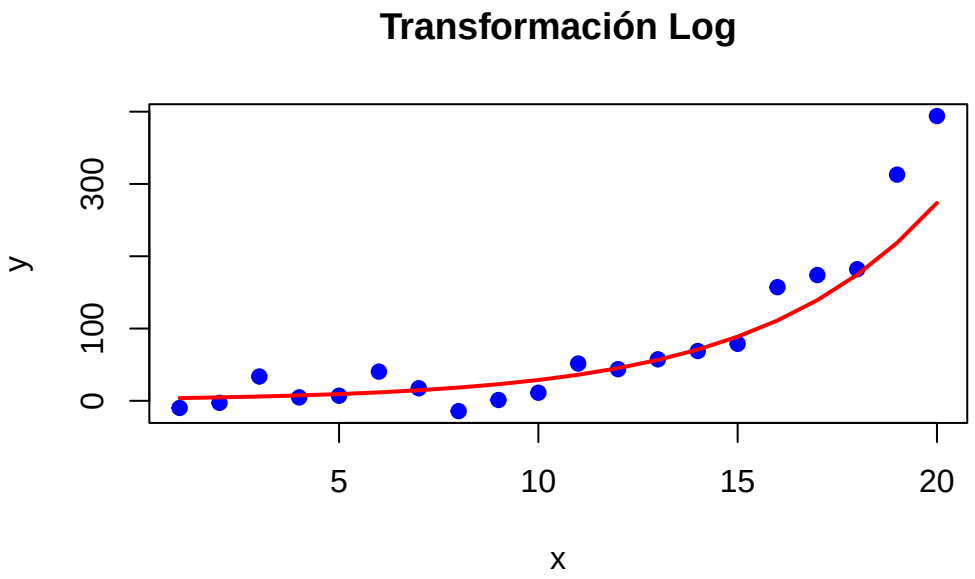
```r
# Datos simulados con crecimiento exponencial
set.seed(123)
x <- 1:20
y <- exp(0.3 * x) + rnorm(20, mean = 0, sd = 20)


# Transformación logarítmica para linearizar la relación
modelo_log <- lm(log(y) ~ x)
```

```
Warning in log(y): NaNs produced
```

```r
summary(modelo_log)
```

```

Call:
lm(formula = log(y) ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-3.00071 -0.11760  0.04277  0.35876  1.73316

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.10652    0.59757   1.852 0.083853 .
x            0.22528    0.04655   4.839 0.000217 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.011 on 15 degrees of freedom
  (3 observations deleted due to missingness)
Multiple R-squared:  0.6096,    Adjusted R-squared:  0.5835
F-statistic: 23.42 on 1 and 15 DF,  p-value: 0.0002166
```

```r
# Visualización
plot(x, y, main = "Transformación Log", pch = 19, col = "blue", ylab = "y", xlab = "x")

# Predicciones para los mismos valores de x
predicciones <- predict(modelo_log, newdata = data.frame(x = x))

# Convertir predicciones a la escala original (exponencial inverso del log)
lines(x, exp(predicciones), col = "red", lwd = 2)
```



Transformación de Raíz Cuadrada ($\sqrt{}$)

Se emplea para reducir la heterocedasticidad, especialmente cuando la varianza aumenta linealmente con la media:

$$Y = \sqrt{Y}$$

o

$$X = \sqrt{X}$$

Se emplea comúnmente en conteos de eventos o variables positivas (número de llamadas, defectos, etc.).

Ejemplo

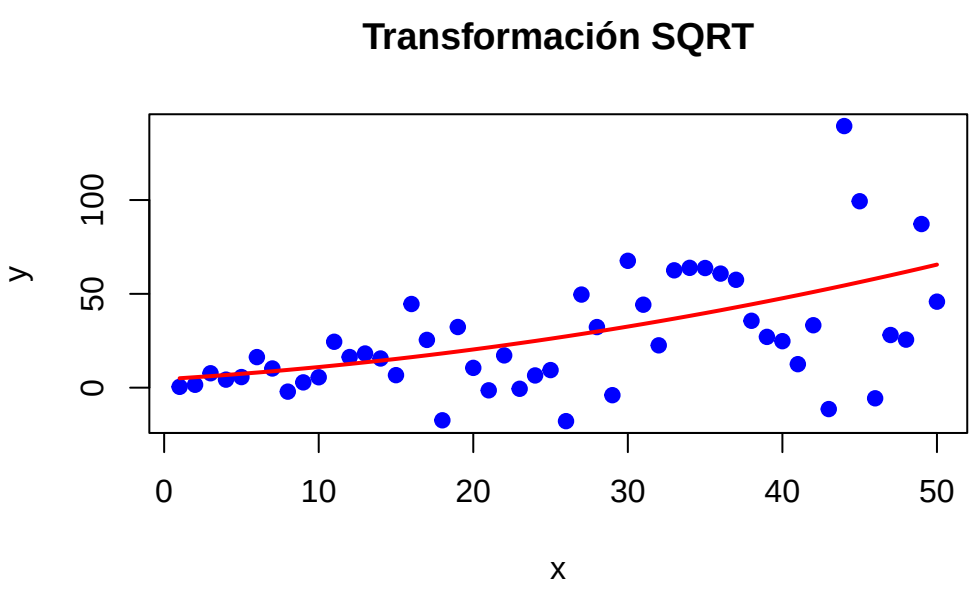
```r
# Datos simulados con variabilidad creciente
set.seed(123)
x <- 1:50
y <- x + rnorm(50, mean = 0, sd = x)

# Aplicando raíz cuadrada a la variable dependiente
modelo_sqrt <- lm(sqrt(y) ~ x)
```

```
Warning in sqrt(y): NaNs produced
```

```r
summary(modelo_sqrt)
```

```

Call:
lm(formula = sqrt(y) ~ x)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.4838 -1.3312  0.1822  1.3441  4.4278 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.12028    0.53305   3.978 0.000284 ***
x            0.11955    0.01819   6.574 7.39e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.752 on 40 degrees of freedom
  (8 observations deleted due to missingness)
Multiple R-squared:  0.5193,    Adjusted R-squared:  0.5073 
F-statistic: 43.22 on 1 and 40 DF,  p-value: 7.387e-08
```

```r
# Visualización
plot(x, y, main = "Transformación SQRT", pch = 19, col = "blue", ylab = "y", xlab = "x")

# Predicciones para los mismos valores de x
predicciones <- predict(modelo_sqrt, newdata = data.frame(x = x))

# Convertir predicciones a la escala original
lines(x, predicciones^2, col = "red", lwd = 2)
```



Transformación Inversa ($\frac{1}{X}$)

Se emplea la tranformación inversa para modelar relaciones donde el efecto de la variable independiente disminuye rápidamente.

$$Y = \frac{1}{X}$$

Es especialmente útil cuando se espera que un aumento en X tenga un efecto decreciente en Y (Relaciones físicas como la ley de la gravitación, velocidad vs. tiempo en fricción, etc).

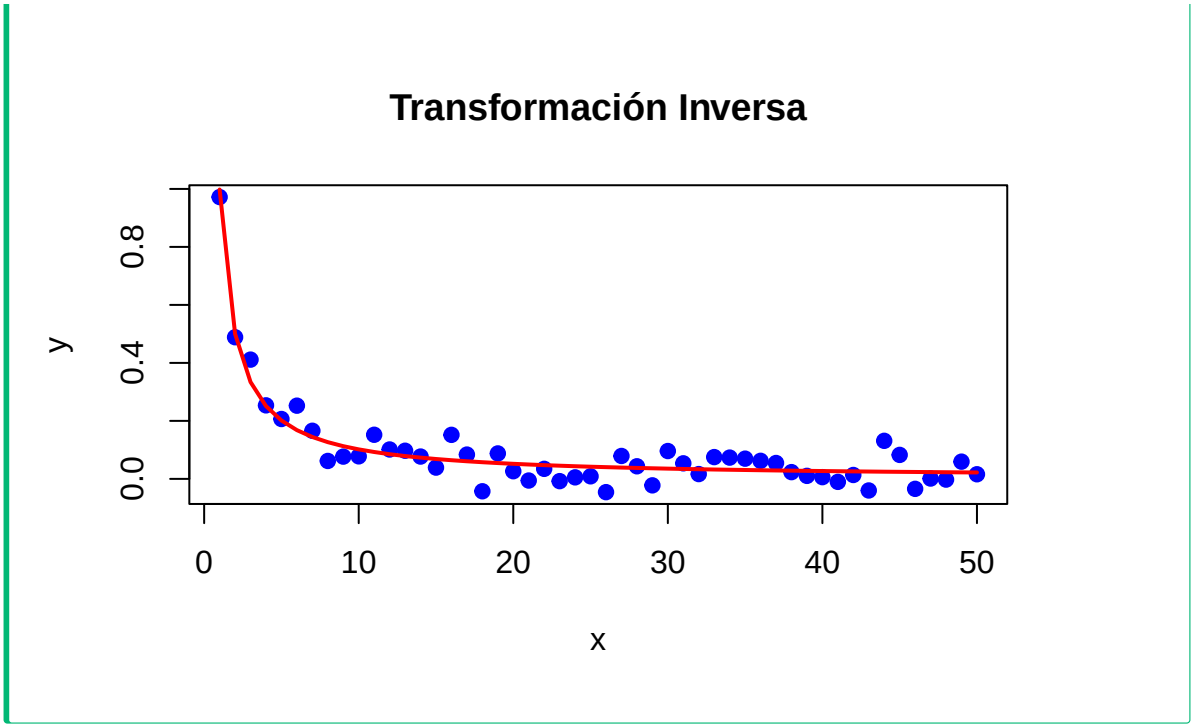
Ejemplo

```r
# Datos simulados con relación inversa
set.seed(123)
x <- 1:50
y <- 1 / x + rnorm(50, mean = 0, sd = 0.05)

# Ajuste del modelo con transformación inversa
modelo_inverso <- lm(y ~ I(1/x))
summary(modelo_inverso)
```

```
Call:
```

```
lm(formula = y ~ I(1/x))

Residuals:
      Min        1Q    Median        3Q       Max
-0.100193 -0.028703 -0.005628  0.033009  0.106450

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.002092   0.007633   0.274    0.785
I(1/x)      0.995869   0.042338  23.522   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04677 on 48 degrees of freedom
Multiple R-squared:  0.9202,    Adjusted R-squared:  0.9185
F-statistic: 553.3 on 1 and 48 DF,  p-value: < 2.2e-16
```

```r
# Visualización
plot(x, y, main = "Transformación Inversa", pch = 19, col = "blue", ylab = "y", xlab = "x")

# Predicciones para los mismos valores de x
predicciones <- predict(modelo_inverso, newdata = data.frame(x = x))

# Convertir predicciones a la escala original
lines(x, predicciones, col = "red", lwd = 2)
```



4.2.2 Transformación de Box-Cox

La **transformación de Box-Cox** es un método que busca automáticamente la mejor transformación para estabilizar la varianza y aproximar la normalidad de los errores. La transformación se define como:

$$Y(\lambda) = \begin{cases} \frac{Y^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \log(Y), & \lambda = 0 \end{cases}$$

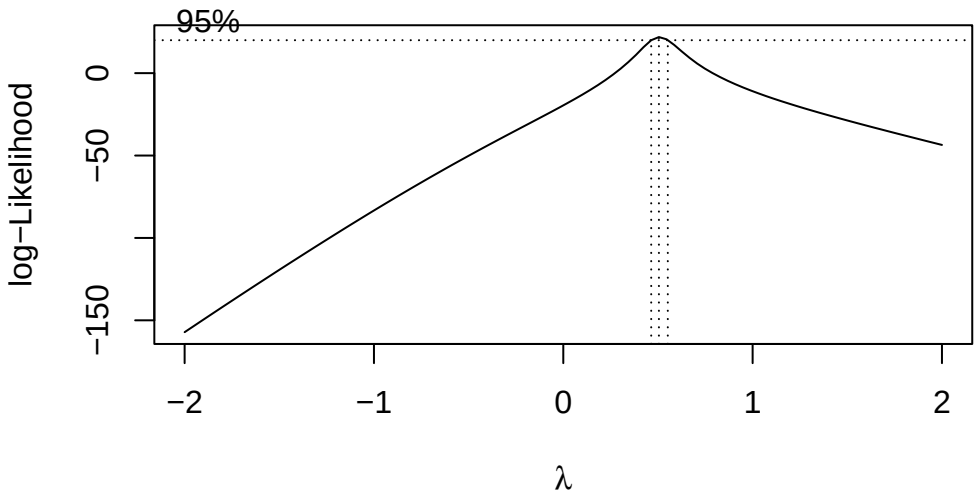
Se emplea para encontrar la transformación óptima para los datos. Se utiliza cuando no está claro qué transformación aplicar. Por ejemplo, en caso de variables continuas con varianza no constante o distribución no normal.

Ejemplo

```r
# Cargar librería para Box-Cox
library(MASS)

# Datos simulados
set.seed(123)
x <- 1:20
y <- x^2 + rnorm(20, mean = 0, sd = 5)

# Verificar si hay valores negativos
min(y)
```

```
[1] -1.802378
```

```r
# Si hay valores negativos, sumar una constante para que todos los valores sean positivos
if (min(y) <= 0) {
  y <- y + abs(min(y)) + 1  # Desplaza todos los valores para que sean positivos
}

# Ajuste de un modelo lineal simple
modelo_bc <- lm(y ~ x)

# Aplicación de la transformación de Box-Cox
boxcox(modelo_bc)
```



La gráfica de **Box-Cox** mostrará el valor óptimo de λ, que indica la mejor transformación para los datos.

4.2.3 Consideracione sobre las transformaciones

Antes de aplicar transformaciones, es importante diagnosticar si realmente son necesarias. Existen varias herramientas para identificar problemas en los datos que pueden solucionarse con transformaciones:

- **Gráficos de Dispersión:** Visualizar la relación entre la variable dependiente y las independientes puede revelar patrones no lineales o heterocedasticidad.
- **Análisis de Residuos:** Un gráfico de los residuos frente a los valores ajustados debe mostrar una distribución aleatoria. Patrones sistemáticos o "abanicos" indican la necesidad de transformación. El gráfico de **QQ-Plot** de los residuos ayuda a evaluar la normalidad.
- **Pruebas Estadísticas:** Pruebas de normalidad como **Shapiro-Wilk** para los residuos. Pruebas de heterocedasticidad como **Breusch-Pagan**.

Si bien las transformaciones pueden mejorar el ajuste del modelo, también pueden afectar la **interpretación de los coeficientes**. Es importante tener en cuenta cómo cambia el significado de los resultados:

- **Transformaciones en la variable dependiente:**
  - Si aplicas $\log(Y)$, los coeficientes representan **cambios proporcionales** en Y.
  - Si aplicas $\sqrt{Y}$, los coeficientes representan la tasa de cambio en la raíz cuadrada de Y.
- **Transformaciones en la variable independiente:**
  - Si transformas X con $\log(X)$, los coeficientes indican cómo cambia Y por cada **incremento porcentual** en X.
  - Si transformas X con $\frac{1}{X}$, los coeficientes representan el cambio en Y por cada unidad de **disminución** en X.
- **Revertir Transformaciones para Interpretación:** Después de ajustar un modelo, es posible transformar las predicciones de nuevo a la escala original para facilitar la interpretación.

4.3 Ingeniería de características

La **ingeniería de características** es el arte y la ciencia de transformar los datos brutos en representaciones que faciliten el aprendizaje y mejoren la capacidad predictiva de los modelos. Consiste en el proceso de **crear, transformar y seleccionar** las variables que se utilizan en un modelo para mejorar su rendimiento. Una característica bien diseñada puede hacer que un modelo simple supere a modelos más complejos, mientras que características irrelevantes o mal definidas pueden degradar significativamente la calidad del análisis. Este proceso incluye:

- **Creación de nuevas variables** a partir de las existentes.
- **Transformación de variables** para mejorar su distribución o relación con la variable objetivo.
- **Selección de las características más relevantes**, eliminando aquellas que no aportan valor o introducen ruido.
- **Preparación de datos para modelos específicos**, asegurando que las variables cumplan con los requisitos del algoritmo (por ejemplo, escalado, normalización o codificación).

4.3.1 Creación de nuevas variables

Una de las tareas más importantes en la ingeniería de características es la **creación de nuevas variables** que puedan capturar relaciones complejas entre las variables independientes y la variable objetivo.

Las **interacciones entre variables** permiten capturar relaciones no lineales entre las variables al considerar cómo el efecto de una variable puede depender del valor de otra. Si se dispone de dos variables X_i y X_j, es posible crear una nueva variable de interacción $X_{interacción} = X_i \times X_j$.

Ejemplo

```r
# Ejemplo en R de interacción de variables
set.seed(123)
X1 <- rnorm(100)
X2 <- rnorm(100)
Y <- 3 + 2 * X1 + 4 * X2 + 1.5 * X1 * X2 + rnorm(100)

# Crear variable de interacción
X_interaccion <- X1 * X2

# Ajustar modelo con interacción
modelo_interaccion <- lm(Y ~ X1 + X2 + X_interaccion)
summary(modelo_interaccion)
```

```
Call:
lm(formula = Y ~ X1 + X2 + X_interaccion)

Residuals:
    Min      1Q  Median      3Q     Max
-1.8719 -0.6777 -0.1086  0.5897  2.3166

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.14098    0.09578   32.80   <2e-16 ***
X1             1.90719    0.10834   17.60   <2e-16 ***
X2             4.03434    0.09881   40.83   <2e-16 ***
X_interaccion  1.65911    0.11449   14.49   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.9468 on 96 degrees of freedom
Multiple R-squared:  0.953, Adjusted R-squared:  0.9516
F-statistic: 649.2 on 3 and 96 DF,  p-value: < 2.2e-16
```

Tal y como estudiamos cuando tratamos el tema de regresión polinómica, agregar **términos polinómicos** permite capturar relaciones no lineales al incluir potencias de las variables independientes. Por ejemplo, para una variable X, se puede crear X^2, X^3, etc.

Ejemplo

```r
# Datos simulados
x <- 1:20
y <- 5 + 2 * x + 0.5 * x^2 + rnorm(20, mean = 0, sd = 5)

# Incluir término cuadrático en el modelo
modelo_polinomico <- lm(y ~ x + I(x^2))
summary(modelo_polinomico)
```

```
Call:
lm(formula = y ~ x + I(x^2))

Residuals:
    Min      1Q  Median      3Q     Max
-9.9235 -2.9142  0.6081  3.0085  9.6944

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.49742    4.18141  -0.119  0.90670
x            2.85574    0.91705   3.114  0.00631 **
I(x^2)       0.47433    0.04242  11.182 2.94e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.62 on 17 degrees of freedom
Multiple R-squared:  0.9953,    Adjusted R-squared:  0.9947
F-statistic:  1792 on 2 and 17 DF,  p-value: < 2.2e-16
```

Crear combinaciones simples de variables puede capturar relaciones ocultas en los datos. Por ejemplo:

- Sumas o diferencias: $X_{\text{nuevo}} = X_1 + X_2$

- Ratios: $X_{\text{ratio}} = \frac{X_1}{X_2}$
- Variables categóricas combinadas: Fusionar categorías relacionadas en una nueva variable. Deben ser categorías que, desde un punto de vista del dominio de aplicación, tenga sentido combinar.

4.3.2 Selección y Reducción de variables

Una vez que se han creado nuevas características, es importante **seleccionar** las que son más relevantes para el modelo y **eliminar** aquellas que no aportan valor o introducen ruido.

Se trató con detalle estos conceptos en el tema anterior. Planteamos un ejemplo de selección *Stepwise*.

Ejemplo

```r
# Datos simulados
set.seed(123)
X1 <- rnorm(100)
X2 <- rnorm(100)
X3 <- rnorm(100)
Y <- 3 + 2 * X1 + 4 * X2 + rnorm(100)

# Modelo completo con todas las variables
modelo_completo <- lm(Y ~ X1 + X2 + X3)

# Selección de variables usando stepwise
modelo_seleccionado <- step(modelo_completo, direction = "both")
```

```
Start:  AIC=13.96
Y ~ X1 + X2 + X3

       Df Sum of Sq     RSS     AIC
- X3    1      0.29  106.43  12.236
<none>               106.15  13.964
- X1    1    306.06  412.21 147.636
- X2    1   1510.95 1617.09 284.321

Step:  AIC=12.24
Y ~ X1 + X2
```

```
       Df Sum of Sq     RSS     AIC
<none>               106.43  12.236
+ X3    1      0.29  106.15  13.964
- X1    1    313.60  420.04 147.517
- X2    1   1510.83 1617.26 282.332
```

```
summary(modelo_seleccionado)
```

```

Call:
lm(formula = Y ~ X1 + X2)

Residuals:
     Min       1Q   Median       3Q      Max
-2.47672 -0.67285  0.09839  0.70676  2.62566

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.9729     0.1059   28.08   <2e-16 ***
X1            1.9522     0.1155   16.91   <2e-16 ***
X2            4.0449     0.1090   37.11   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.048 on 97 degrees of freedom
Multiple R-squared:  0.943, Adjusted R-squared:  0.9418
F-statistic: 802.3 on 2 and 97 DF,  p-value: < 2.2e-16
```

Recordemos que los métodos de regularización, estudiados en el tema 2, no solo ajustan el modelo, sino que también penalizan la complejidad, lo que ayuda a eliminar variables irrelevantes.

4.3.3 Escalado y Normalización de Variables

Muchos algoritmos de aprendizaje automático, como la regresión, las redes neuronales y los métodos basados en distancia (k-NN, SVM), son sensibles a la **escala de las variables**. Por lo tanto, es fundamental **escalar** o **normalizar** los datos para garantizar que todas las variables contribuyan de manera equitativa al modelo.

Estandarización (Z-Score Normalization)

La estandarización consiste en restar la media y dividir por la desviación estándar, lo que produce variables con **media cero** y **desviación estándar uno**.

$$X_{\text{estandarizado}} = \frac{X - \bar{X}}{\sigma_X}$$

Ejemplo

```
# Estandarización de una variable
X1_estandarizado <- scale(X1)
```

Normalización Min-Max

La **normalización Min-Max** escala las variables a un rango específico, típicamente entre 0 y 1.

$$X_{\text{normalizado}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

Ejemplo

```
# Normalización Min-Max
X1_min_max <- (X1 - min(X1)) / (max(X1) - min(X1))
```

4.3.4 Técnicas avanzadas de Ingeniería de Características

Cuando se trabaja con grandes conjuntos de datos o con variables altamente correlacionadas, puede ser necesario aplicar técnicas más avanzadas para reducir la dimensionalidad y extraer características relevantes.

4.3.4.1 Análisis de Componentes Principales (PCA)

El **Análisis de Componentes Principales (PCA)** es una técnica de reducción de dimensionalidad que transforma un conjunto de variables correlacionadas en un conjunto más pequeño de **componentes principales** no correlacionados que explican la mayor parte de la varianza en los datos.

Aviso

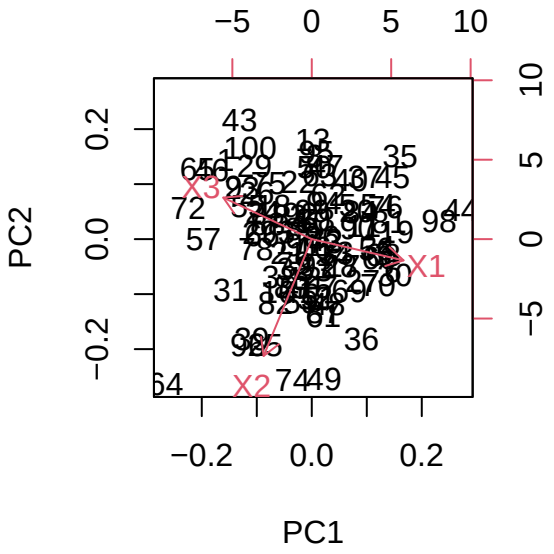
Los detalles del Análisis de Componentes Principales son tratados en la asignatura de Aprendizaje Automático.

Ejemplo

```r
# Datos simulados
set.seed(123)
datos <- data.frame(X1, X2, X3)

# Aplicar PCA
pca_resultado <- prcomp(datos, scale. = TRUE)

# Visualización de los resultados
summary(pca_resultado)
```

```
Importance of components:
                          PC1    PC2    PC3
Standard deviation     1.0726 0.9900 0.9324
Proportion of Variance 0.3835 0.3267 0.2898
Cumulative Proportion  0.3835 0.7102 1.0000
```

```r
biplot(pca_resultado)
```



4.3.4.2 Codificación de variables categóricas

Las variables categóricas deben convertirse en variables numéricas antes de ser utilizadas en muchos modelos. Esto puede hacerse mediante:

Codificación One-Hot

El **One-Hot Encoding** es una técnica utilizada en el **preprocesamiento de datos** para convertir variables categóricas en variables numéricas. Muchos algoritmos de aprendizaje automático y estadística (como la regresión lineal, redes neuronales y máquinas de soporte vectorial) requieren que las variables de entrada sean numéricas, ya que no pueden manejar directamente datos categóricos.

El **One-Hot Encoding** transforma cada categoría en una nueva columna binaria (0 o 1), donde el **1** indica la presencia de una categoría específica y el **0** su ausencia.

Supongamos que tienes una variable categórica llamada **Color** con tres categorías: **Rojo**, **Verde**, y **Azul**.

ID	Color
1	Rojo
2	Verde
3	Azul

ID	Color
4	Rojo
5	Verde

Con **One-Hot Encoding**, creamos una nueva columna para cada categoría única:

ID	Color_Rojo	Color_Verde	Color_Azul
1	1	0	0
2	0	1	0
3	0	0	1
4	1	0	0
5	0	1	0

Cada fila tiene un **único 1** que indica la categoría correspondiente y **ceros** en las otras columnas. Esto convierte la información categórica en un formato que los algoritmos numéricos pueden procesar.

Propiedades Clave

Ventajas del One-Hot Encoding

- Compatibilidad con algoritmos numéricos: La mayoría de los modelos de aprendizaje automático requieren variables numéricas. El One-Hot Encoding convierte las categorías en un formato adecuado.
- Evita suposiciones erróneas A diferencia de la **codificación ordinal**, que asigna valores numéricos secuenciales a categorías (por ejemplo, Rojo = 1, Verde = 2, Azul = 3), el One-Hot Encoding **no introduce un orden artificial** entre las categorías. Esto es importante cuando no hay una jerarquía natural.
- Mejora la Interpretabilidad en Modelos Lineales: En modelos como la regresión lineal, cada columna creada mediante One-Hot Encoding representa el efecto específico de esa categoría.

Desventajas del One-Hot Encoding

- Incremento de la Dimensionalidad: Si la variable categórica tiene muchas categorías únicas (por ejemplo, países o códigos postales), el número de columnas creadas puede ser muy grande. Esto puede conducir a problemas de **"curse of dimensionality"** (la maldición de la dimensionalidad), afectando el rendimiento del modelo y aumentando el tiempo de computación.

- Colinealidad Perfecta (Dummy Variable Trap): Al crear una columna para cada categoría, una de las columnas se puede representar como una combinación lineal de las demás. Esto puede causar problemas en modelos lineales. Para evitarlo, se elimina una categoría de referencia (típicamente la primera), lo que se conoce como **evitar la trampa de las variables ficticias** (*dummy variable trap*).

Ejemplo

```r
# Variable categórica simulada
categoria <- factor(c("bajo", "medio", "alto", "medio", "alto"))

# Codificación one-hot
library(caret)
```

```
Loading required package: ggplot2

Loading required package: lattice
```

```r
dummies <- dummyVars(~ categoria, data = data.frame(categoria))
print(dummies)
```

```
Dummy Variable Object

Formula: ~categoria
1 variables, 1 factors
Variables and levels will be separated by '.'
A less than full rank encoding is used
```

```r
datos_codificados <- predict(dummies, newdata = data.frame(categoria))
```

Ejemplo

```r
# Instalar y cargar la librería caret
library(caret)

# Crear un data frame
datos <- data.frame(ID = 1:5, Color = c("Rojo", "Verde", "Azul", "Rojo", "Verde"))

# Aplicar One-Hot Encoding usando dummyVars
dummy <- dummyVars(~ Color, data = datos)
one_hot <- predict(dummy, newdata = datos)

# Ver los resultados
print(one_hot)
```

```
  ColorAzul ColorRojo ColorVerde
1         0         1          0
2         0         0          1
3         1         0          0
4         0         1          0
5         0         0          1
```

Aviso

Cuando aplicamos One-Hot Encoding, el conjunto de variables resultantes puede generar colinealidad perfecta, lo que puede ser problemático en modelos lineales. Para evitarlo, es común eliminar una de las columnas creadas (que actuará como categoría de referencia).

```r
# One-Hot Encoding con eliminación de una categoría (categoría de referencia)
one_hot_ref <- model.matrix(~ Color, data = datos)[, -1]  # Eliminar la primera columna

print(one_hot_ref)
```

```
  ColorRojo ColorVerde
1         1          0
2         0          1
3         0          0
4         1          0
5         0          1
```

Esto elimina una columna (por ejemplo, **ColorAzul**) y permite que las otras columnas se interpreten en relación a la categoría de referencia.

4.3.4.3 Codificación Ordinal

La **codificación ordinal** es una técnica de preprocesamiento de datos utilizada para convertir variables categóricas en valores numéricos **manteniendo el orden natural** entre las categorías. A diferencia del **One-Hot Encoding**, que trata a todas las categorías como independientes y sin relación entre sí, la codificación ordinal es útil cuando las categorías tienen una **jerarquía** o un **orden lógico**.

En la **codificación ordinal**, a cada categoría se le asigna un número entero que refleja su **posición** o **nivel** en un orden determinado. Esto permite que los modelos estadísticos y de aprendizaje automático interpreten que **algunas categorías son mayores o menores** que otras, lo que es especialmente útil en variables que representan rangos, niveles o clasificaciones.

Supongamos que tenemos una variable llamada **"Nivel de Satisfacción"** con las siguientes categorías:

Nivel de Satisfacción
Muy Insatisfecho
Insatisfecho
Neutral
Satisfecho
Muy Satisfecho

Aquí, hay un orden lógico desde **"Muy Insatisfecho"** hasta **"Muy Satisfecho"**. Aplicando codificación ordinal, asignamos números que reflejen esta jerarquía:

Nivel de Satisfacción	**Codificación Ordinal**
Muy Insatisfecho	1
Insatisfecho	2
Neutral	3
Satisfecho	4
Muy Satisfecho	5

---

i Propiedades clave

La codificación ordinal es adecuada cuando:

- Las categorías tienen un orden natural: Ejemplos incluyen niveles educativos

(**Primaria**, **Secundaria**, **Universidad**), calificaciones (**Bajo**, **Medio**, **Alto**), o satisfacción del cliente (**Insatisfecho**, **Neutral**, **Satisfecho**).

- El modelo puede interpretar la relación de orden: Algunos algoritmos, como la **regresión lineal** o las **máquinas de vectores soporte (SVM)**, pueden beneficiarse de la codificación ordinal si el orden es relevante para la variable objetivo.
- Reducción de dimensionalidad: A diferencia del One-Hot Encoding, que puede crear muchas columnas para variables con múltiples categorías, la codificación ordinal **mantiene la variable en una sola columna**, lo que es más eficiente para conjuntos de datos grandes.

Ventajas y desventajas

Ventajas

- Preserva la jerarquía de las categorías: Permite que el modelo entienda que ciertas categorías son **mayores** o **menores** que otras.
- Reducción de la dimensionalidad: A diferencia del One-Hot Encoding, no aumenta el número de columnas, lo que reduce el riesgo de la **maldición de la dimensionalidad**.
- Simplicidad Computacional: Es más eficiente en términos de almacenamiento y tiempo de computación, especialmente para variables con muchas categorías.

Desventajas

- Riesgo de interpretación incorrecta del orden: Si la variable categórica **no tiene un orden lógico**, la codificación ordinal puede inducir al modelo a asumir relaciones que no existen. Por ejemplo, supongamos que tienes una variable **Color** con categorías **Rojo**, **Verde**, y **Azul**. Asignarles valores como Rojo = 1, Verde = 2, Azul = 3 podría inducir al modelo a pensar que Verde es "mayor" que Rojo y Azul es "mayor" que Verde, lo cual no tiene sentido en este contexto.
- No Captura la Magnitud de la Diferencia: Aunque las categorías están ordenadas, la codificación ordinal **no refleja la magnitud real de las diferencias** entre categorías. Por ejemplo, la diferencia entre "Insatisfecho" y "Neutral" puede no ser la misma que entre "Satisfecho" y "Muy Satisfecho".

Ejemplo

```r
# Crear un data frame con una variable categórica ordinal
datos <- data.frame(
  ID = 1:5,
  Satisfaccion = c("Muy Insatisfecho", "Insatisfecho", "Neutral", "Satisfecho", "Muy Satisf
)

# Convertir la variable en un factor ordenado
datos$Satisfaccion_ordinal <- factor(datos$Satisfaccion,
                                     levels = c("Muy Insatisfecho", "Insatisfecho", "Neutra
                                     ordered = TRUE)

# Ver la estructura del factor
str(datos)
```

```
'data.frame':   5 obs. of  3 variables:
 $ ID                  : int  1 2 3 4 5
 $ Satisfaccion        : chr  "Muy Insatisfecho" "Insatisfecho" "Neutral" "Satisfecho" ...
 $ Satisfaccion_ordinal: Ord.factor w/ 5 levels "Muy Insatisfecho"<..: 1 2 3 4 5
```

```r
# Asignar valores numéricos a las categorías ordenadas
datos$Satisfaccion_codificada <- as.numeric(datos$Satisfaccion_ordinal)

# Ver el resultado
print(datos)
```

```
  ID     Satisfaccion Satisfaccion_ordinal Satisfaccion_codificada
1  1 Muy Insatisfecho     Muy Insatisfecho                       1
2  2     Insatisfecho         Insatisfecho                       2
3  3          Neutral              Neutral                       3
4  4        Satisfecho            Satisfecho                      4
5  5   Muy Satisfecho       Muy Satisfecho                       5
```

Podemos usar la variable codificada ordinalmente en un modelo de regresión para evaluar su impacto en una variable dependiente, como una **puntuación de satisfacción general**.

Ejemplo

```r
# Simular una variable de respuesta (puntuación general)
set.seed(123)
datos$Puntuacion <- c(2, 4, 6, 8, 10) + rnorm(5, mean = 0, sd = 0.5)

# Ajustar un modelo de regresión lineal usando la variable codificada
modelo_ordinal <- lm(Puntuacion ~ Satisfaccion_codificada, data = datos)

# Resumen del modelo
summary(modelo_ordinal)
```

```
Call:
lm(formula = Puntuacion ~ Satisfaccion_codificada, data = datos)

Residuals:
      1       2       3       4       5
-0.2090 -0.1279  0.6826 -0.1455 -0.2002

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)              -0.1552     0.4640  -0.335 0.759968
Satisfaccion_codificada   2.0840     0.1399  14.896 0.000657 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4424 on 3 degrees of freedom
Multiple R-squared:  0.9867,    Adjusted R-squared:  0.9822
F-statistic: 221.9 on 1 and 3 DF,  p-value: 0.0006565
```

El modelo ajustará la relación entre la puntuación de satisfacción y el nivel de satisfacción codificado ordinalmente. El coeficiente de la variable **`Satisfaccion_codificada`** indicará cómo cambia la puntuación a medida que aumenta el nivel de satisfacción.

4.3.4.4 Diferencias entre Codificación Ordinal y One-Hot Encoding

Característica	**Codificación Ordinal**	**One-Hot Encoding**
Preserva el orden	Sí, refleja la jerarquía entre categorías.	No, trata cada categoría como independiente.

Característica	**Codificación Ordinal**	**One-Hot Encoding**
Aumenta la dimensionalidad	No, mantiene la variable en una sola columna.	Sí, crea una columna para cada categoría única.
Adecuado para	Variables con orden natural (ej. educación).	Variables sin orden (ej. color, género, ciudad).
Riesgo	Puede inducir al modelo a asumir relaciones falsas si no hay orden real.	Incremento de la complejidad del modelo.

5 Modelos de regresión generalizada

Hasta ahora hemos estuadiado la regresión lineal como una herramienta poderosa para modelar la relación entre una variable dependiente continua y un conjunto de variables independientes. Sin embargo, en muchos contextos del mundo real, las suposiciones de la regresión lineal tradicional no son adecuadas. ¿Qué sucede si la variable dependiente es binaria, como en un diagnóstico médico (enfermo/sano)? ¿O si estás modelando el número de accidentes en una intersección o la cantidad de compras realizadas por un cliente?

Para abordar estos desafíos, se utilizan los llamados **Modelos Lineales Generalizados (GLM)**. Esta clase de modelos amplía la regresión lineal al permitir que la variable dependiente tenga distribuciones diferentes a la normal, como la binomial o la de Poisson. Además, los GLM utilizan funciones de enlace que transforman la relación entre la variable dependiente y los predictores, permitiendo una mayor flexibilidad en el modelado.

Algunos de los modelos más comunes dentro de los GLM son:

- Regresión Logística: Ideal para variables dependientes binarias (sí/no, éxito/fracaso).
- Regresión de Poisson: Utilizada para modelar datos de conteo (número de eventos).
- Regresión Binomial Negativa: Una extensión de la regresión de Poisson para datos de conteo con sobredispersión.
- Modelos de Gamma y Inverso Gaussiano: Utilizados para modelar variables continuas positivas y sesgadas, como tiempos de espera o costos.

En este tema, exploraremos cómo utilizar estos modelos para resolver problemas del mundo real, interpretar sus resultados y evaluar su ajuste.

5.1 Introducción a los GLM

5.1.1 ¿Qué son los Modelos Lineales Generalizados?

Los **Modelos Lineales Generalizados (GLM)** son una extensión de los modelos de regresión lineal que permiten manejar una mayor variedad de tipos de datos y relaciones entre variables (Nelder and Wedderburn 1972). Mientras que la regresión lineal tradicional asume que la variable dependiente es continua y sigue una distribución normal, los GLM permiten trabajar con variables dependientes que:

- Son **binarias** (como éxito/fracaso o sí/no).
- Representan **conteos** de eventos (número de llamadas, accidentes, etc.).
- Son **continuas positivas** y no siguen una distribución normal (como tiempos o costos).

Los GLM proporcionan una estructura flexible para modelar la relación entre una o más variables independientes y una variable dependiente que sigue alguna distribución de la **familia exponencial** (binomial, Poisson, gamma, entre otras).

5.1.2 Componentes de un Modelo Lineal Generalizado

Un GLM se define por tres componentes clave:

1. **Componente Aleatorio:**

   Este componente describe la distribución de la variable dependiente. En la regresión lineal, la variable dependiente sigue una distribución normal. En los GLM, puede seguir otras distribuciones de la **familia exponencial**, como:

   - **Distribución Binomial:** Para variables categóricas binarias (0/1, éxito/fracaso).
   - **Distribución de Poisson:** Para datos de conteo (número de eventos).
   - **Distribución Gamma:** Para variables continuas y positivas (como costos o tiempos).

2. **Componente Sistemático:**
   Este componente describe cómo las variables independientes se combinan linealmente en el modelo. Se define como:

   $$\eta = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

   Donde η es el **predictor lineal** y β representa los coeficientes del modelo.

3. **Función de Enlace:**
   La función de enlace conecta el componente sistemático con la media de la variable dependiente. Mientras que en la regresión lineal la relación es directa (\$Y = \$), en los GLM se utiliza una función de enlace $g(\mu)$ para transformar la media μ y ajustar diferentes tipos de datos.

   $$g(\mu) = \eta$$

Ejemplos de funciones de enlace:

- **Logística (Logit):** Para la regresión logística, que modela la probabilidad de un evento.

$$g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$$

- **Logarítmica:** Para la regresión de Poisson, que modela tasas de eventos.

$$g(\mu) = \log(\mu)$$

- **Identidad:** Para la regresión lineal estándar.

$$g(\mu) = \mu$$

i Aplicaciones

Los GLM se utilizan en una amplia variedad de disciplinas para resolver problemas del mundo real:
Regresión Logística (para variables binarias):

- **Medicina:** Predicción de la presencia o ausencia de una enfermedad basada en factores de riesgo.
- **Marketing:** Determinación de la probabilidad de que un cliente compre un producto.
- **Finanzas:** Evaluación de la probabilidad de incumplimiento de pago de un préstamo.

Regresión de Poisson (para datos de conteo):

- **Transporte:** Modelado del número de accidentes en una carretera en un período de tiempo.
- **Ecología:** Conteo de especies en un área determinada.
- **Telecomunicaciones:** Número de llamadas recibidas por un centro de atención.

Regresión Binomial Negativa (para conteos con sobredispersión):

- **Salud Pública:** Modelado del número de visitas al médico o incidentes de una enfermedad en una población.

Modelos Gamma (para variables continuas positivas):

- **Seguros:** Estimación de los costos de reclamos de seguros.
- **Ingeniería:** Modelado de tiempos de falla en procesos industriales.

5.1.3 Diferencias clave entre la Regresión Lineal y los GLM

Característica	Regresión Lineal	Modelos Lineales Generalizados (GLM)
Distribución de la variable dependiente	Normal	Familia exponencial (binomial, Poisson, gamma, etc.)
Tipo de variable dependiente	Continua	Binaria, de conteo, continua positiva
Relación entre las variables	Lineal directa	Relación transformada mediante una función de enlace
Función de Enlace	Identidad ($g(\mu) = \mu$)	Logit, logarítmica, inversa, etc.

Las ventajas principales de los GLM son:

- **Flexibilidad:** Los GLM permiten modelar diferentes tipos de variables dependientes, lo que amplía significativamente el rango de problemas que se pueden abordar.
- **Interpretación Coherente:** Aunque se utilizan funciones de enlace, los coeficientes de los GLM pueden interpretarse de manera similar a los modelos lineales, proporcionando información sobre el impacto de cada variable independiente.
- **Evaluación Estadística Robusta:** Los GLM permiten la realización de pruebas de hipótesis, la construcción de intervalos de confianza y la evaluación de la bondad del ajuste mediante medidas como el **AIC** y el **BIC**.

Ejemplo

```r
# Cargar librería y datos
library(MASS)
data(Pima.tr)  # Datos sobre diabetes en mujeres de origen pima

# Ajustar un modelo de regresión logística
modelo_logistico <- glm(type ~ npreg + glu + bmi, data = Pima.tr, family = binomial)

# Resumen del modelo
summary(modelo_logistico)
```

```
Call:
```

```
glm(formula = type ~ npreg + glu + bmi, family = binomial, data = Pima.tr)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.718723   1.411080  -6.179 6.46e-10 ***
npreg        0.149213   0.051833   2.879  0.00399 **
glu          0.033879   0.006327   5.355 8.55e-08 ***
bmi          0.094817   0.032405   2.926  0.00343 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 256.41  on 199  degrees of freedom
Residual deviance: 189.89  on 196  degrees of freedom
AIC: 197.89

Number of Fisher Scoring iterations: 5
```

```r
# Predicciones de la probabilidad de tener diabetes
predicciones <- predict(modelo_logistico, type = "response")

# Ver primeras predicciones
head(predicciones)
```

```
         1          2          3          4          5          6
0.10014804 0.78786795 0.12244031 0.80425012 0.06975347 0.21233644
```

Los **Modelos Lineales Generalizados** amplían el alcance de la regresión lineal clásica, proporcionando herramientas para modelar una amplia variedad de tipos de datos, desde variables binarias hasta datos de conteo y variables continuas no normales. A través del uso de funciones de enlace y distribuciones flexibles, los GLM permiten resolver problemas complejos del mundo real en campos tan diversos como la medicina, el marketing, la ingeniería y las ciencias sociales.

En las próximas secciones, exploraremos en detalle cómo aplicar estos modelos específicos, como la **regresión logística** y la **regresión de Poisson**, y cómo interpretar sus resultados en diferentes contextos.

5.2 Regresión Logística

La **regresión logística** es una herramienta fundamental para modelar la probabilidad de eventos binarios en una variedad de contextos, desde la medicina hasta la economía y el marketing (Hosmer Jr, Lemeshow, and Sturdivant 2013). La correcta interpretación de los coeficientes mediante **odds ratios**, así como la evaluación del ajuste del modelo mediante curvas **ROC** y matrices de confusión, son esenciales para extraer conclusiones válidas de los datos.

5.2.1 Fundamentos de la Regresión Logística

La **regresión logística** es una técnica estadística utilizada para modelar la probabilidad de ocurrencia de un evento binario, es decir, cuando la variable dependiente toma solo dos posibles valores (por ejemplo, **éxito/fracaso**, **sí/no**, **enfermo/sano**). A diferencia de la regresión lineal, que modela una relación lineal entre variables, la regresión logística utiliza una **función logística** para asegurar que las predicciones estén en el rango [0,1], lo cual es necesario para interpretar los resultados como probabilidades.

La función Logística (Sigmoide)

La función logística transforma cualquier valor real en un valor comprendido entre 0 y 1. La forma matemática de la función logística es:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p)}}$$

Donde:

- $P(Y = 1|X)$ es la probabilidad de que el evento ocurra.
- β_0 es el intercepto y $\beta_1, \beta_2, \ldots, \beta_p$ son los coeficientes asociados a las variables independientes $X_1, X_2, \ldots, X_p$.

La **curva sigmoide** que representa esta función tiene forma de "S", lo que refleja que para valores muy pequeños o muy grandes del predictor, la probabilidad se aplana hacia 0 o 1, respectivamente.

Función de Enlace Logit

En la regresión logística, la relación entre el predictor lineal y la probabilidad se establece mediante la **función de enlace logit**. El logit de una probabilidad p se define como:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

Esta transformación convierte una probabilidad en una escala que va de $-\infty$ a $+\infty$, lo que permite ajustar un modelo lineal a los datos. El modelo logístico puede expresarse como:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

5.2.2 Interpretación de coeficientes y Odds Ratios

Uno de los aspectos más importantes de la regresión logística es la interpretación de los coeficientes. Dado que los coeficientes están en la escala del logit, su interpretación directa no es tan intuitiva como en la regresión lineal. Sin embargo, podemos interpretarlos utilizando **odds** y **odds ratios**.

El **odds** o razón de probabilidades de que ocurra un evento es el cociente entre la probabilidad de que ocurra el evento y la probabilidad de que no ocurra:

$$\text{odds} = \frac{p}{1-p}$$

Por ejemplo, si la probabilidad de éxito es 0.8, el odds sería:

$$\text{odds} = \frac{0.8}{1-0.8} = 4$$

Esto significa que el evento es **4 veces más probable** que no ocurra.

El **odds ratio (OR)** mide el cambio en los odds cuando una variable independiente aumenta en una unidad. Se calcula como el exponencial del coeficiente de la regresión logística:

$$\text{OR} = e^{\beta}$$

Interpretación de OR:

- Si $\mathbf{OR > 1}$, el evento es más probable a medida que aumenta la variable independiente.
- Si $\mathbf{OR < 1}$, el evento es menos probable a medida que aumenta la variable independiente.
- Si $\mathbf{OR = 1}$, no hay efecto.

Ejemplo

Supongamos que ajustamos un modelo de regresión logística para predecir la probabilidad de tener diabetes en función del índice de masa corporal (BMI). El coeficiente asociado a **BMI** es 0.08.

$$\text{OR} = e^{0.08} \approx 1.083$$

Esto significa que por cada incremento de 1 unidad en el BMI, la **odds** de tener diabetes aumentan en un **8.3%**.

5.2.3 Evaluación del modelo Logístico

A diferencia de la regresión lineal, donde se usa el coeficiente de determinación (R^2) para evaluar el ajuste, en la regresión logística se utilizan otros métodos para medir la calidad del modelo.

Matriz de Confusión

La **matriz de confusión** compara las predicciones del modelo con los valores reales, clasificando las observaciones en:

- **Verdaderos Positivos (VP):** Predijo positivo y es positivo.
- **Falsos Positivos (FP):** Predijo positivo pero es negativo.
- **Verdaderos Negativos (VN):** Predijo negativo y es negativo.
- **Falsos Negativos (FN):** Predijo negativo pero es positivo.

A partir de esta matriz, se pueden calcular métricas importantes como:

- **Precisión (Accuracy):** $\frac{VP+VN}{\text{Total}}$
- **Sensibilidad (Recall o Tasa de Verdaderos Positivos):** $\frac{VP}{VP+FN}$
- **Especificidad (Tasa de Verdaderos Negativos):** $\frac{VN}{VN+FP}$

> Aviso
>
> Los detalles de la evaluación de un modelo empleando la Matríz de Confusión son ampliamente tratados en la asignatura de Aprendizaje Automático.

Curva ROC y AUC

La **Curva ROC (Receiver Operating Characteristic)** muestra la relación entre la **tasa de verdaderos positivos** y la **tasa de falsos positivos** a diferentes umbrales de clasificación.

El **AUC (Área Bajo la Curva ROC)** mide la capacidad del modelo para discriminar entre las clases. Un AUC de 0.5 indica que el modelo no tiene capacidad predictiva, mientras que un AUC de 1.0 indica un modelo perfecto.

Pseudo R² (Nagelkerke, McFadden)

Aunque el R^2 tradicional no se aplica directamente a la regresión logística, existen medidas como el **pseudo** R^2 que proporcionan una idea de la bondad del ajuste del modelo.

- **McFadden's R^2:**

$$R^2_{\text{McFadden}} = 1 - \frac{\log L_{\text{modelo}}}{\log L_{\text{modelo nulo}}}$$

Donde $\log L_{\text{modelo}}$ es el log-likelihood del modelo ajustado y $\log L_{\text{modelo nulo}}$ es el log-likelihood de un modelo sin predictores.

Ejemplo

Vamos a aplicar la regresión logística en R utilizando el conjunto de datos `Pima.tr` del paquete `MASS`, que contiene información sobre mujeres pima y si tienen o no diabetes.

```r
# Cargar la librería y el conjunto de datos
library(MASS)
data(Pima.tr)

# Ajustar el modelo de regresión logística
modelo_logistico <- glm(type ~ npreg + glu + bmi, data = Pima.tr, family = binomial)

# Resumen del modelo
summary(modelo_logistico)
```

```
Call:
glm(formula = type ~ npreg + glu + bmi, family = binomial, data = Pima.tr)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.718723   1.411080  -6.179 6.46e-10 ***
npreg        0.149213   0.051833   2.879  0.00399 **
glu          0.033879   0.006327   5.355 8.55e-08 ***
bmi          0.094817   0.032405   2.926  0.00343 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 256.41  on 199  degrees of freedom
Residual deviance: 189.89  on 196  degrees of freedom
AIC: 197.89

Number of Fisher Scoring iterations: 5
```

```r
# Predicciones de probabilidad
predicciones_prob <- predict(modelo_logistico, type = "response")

# Clasificación con un umbral de 0.5
predicciones_clase <- ifelse(predicciones_prob > 0.5, "Yes", "No")

# Crear matriz de confusión
tabla_confusion <- table(Predicted = predicciones_clase, Actual = Pima.tr$type)
print(tabla_confusion)
```

```
         Actual
Predicted  No Yes
      No  114  29
      Yes  18  39
```

```r
# Calcular precisión
accuracy <- sum(diag(tabla_confusion)) / sum(tabla_confusion)
print(paste("Precisión:", round(accuracy, 3)))
```

```
[1] "Precisión: 0.765"
```

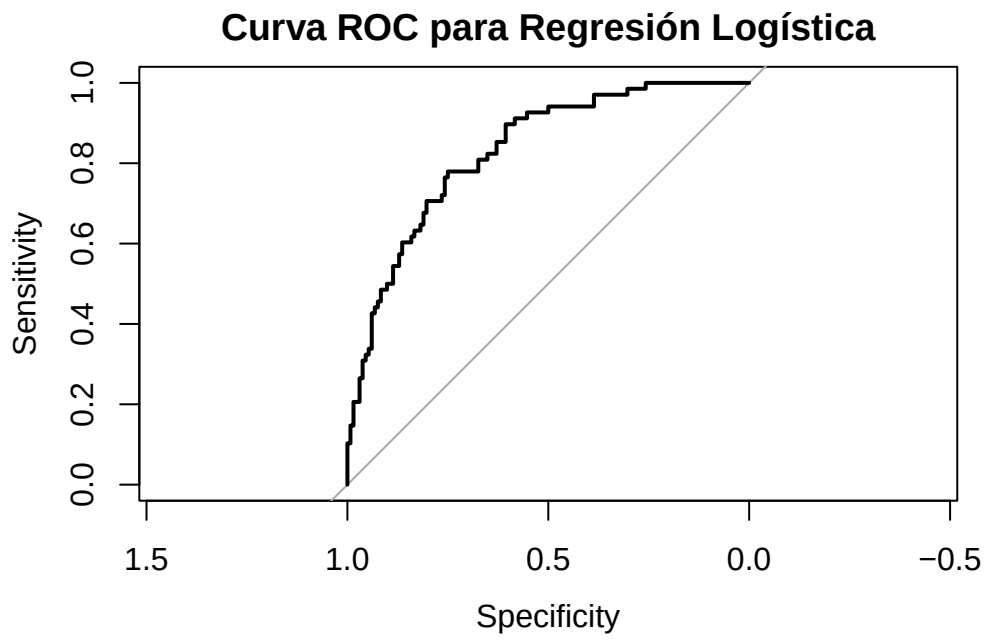
```r
# Cargar librería para curvas ROC
library(pROC)
```

```
Type 'citation("pROC")' for a citation.


Attaching package: 'pROC'

The following objects are masked from 'package:stats':

    cov, smooth, var
```

```r
# Curva ROC
roc_obj <- roc(Pima.tr$type, predicciones_prob)
```

```
Setting levels: control = No, case = Yes

Setting direction: controls < cases
```

```r
plot(roc_obj, main = "Curva ROC para Regresión Logística")
```




```r
# Calcular AUC
auc_valor <- auc(roc_obj)
print(paste("AUC:", round(auc_valor, 3)))
```

```
[1] "AUC: 0.831"
```

5.3 Regresión de Poisson

La **regresión de Poisson** es una técnica estadística utilizada para modelar **datos de conteo**, es decir, situaciones en las que la variable dependiente representa el número de veces que ocurre un evento en un período de tiempo o espacio específico (Coxe, West, and Aiken 2009). Este tipo de modelo es adecuado cuando la variable dependiente toma valores enteros no negativos $(0, 1, 2, ...)$ y sigue una distribución de **Poisson**.

La distribución de Poisson describe la probabilidad de que ocurra un número determinado de eventos en un intervalo fijo, dado que estos eventos ocurren de forma independiente y a una tasa constante.

La **función de probabilidad** de la distribución de Poisson es:

$$P(Y = y) = \frac{e^{-\lambda}\lambda^{y}}{y!}$$

Donde:

- Y es la variable aleatoria que representa el número de eventos.
- λ es la **tasa media de ocurrencia** de los eventos (esperanza de Y).
- y es el número de eventos observados ($y = 0, 1, 2, ...$).

5.3.1 Modelo de regresión de Poisson

En la **regresión de Poisson**, el objetivo es modelar la relación entre la **tasa de ocurrencia de los eventos** (λ) y un conjunto de variables predictoras $X_1, X_2, \dots, X_p$.

La **forma funcional** del modelo de Poisson es:

$$\log(\lambda) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

Donde:

- $\log(\lambda)$ es la **función de enlace logarítmica** que asegura que la tasa λ sea siempre positiva.
- $\beta_0, \beta_1, \dots, \beta_p$ son los coeficientes del modelo que describen la influencia de cada predictor sobre la tasa de eventos.

El modelo puede expresarse en términos de la **tasa esperada de eventos** como:

$$\lambda = e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p}$$

5.3.2 Supuestos y limitaciones de la regresión de Poisson

Tal y como ocurre en el modelo de regresión lineal, para que la regresión de Poisson sea adecuada, se deben cumplir ciertos **supuestos**:

- **Independencia de los eventos:** Los eventos deben ocurrir de manera independiente unos de otros.
- **Distribución de Poisson de la variable dependiente:** La variable de respuesta debe seguir una distribución de Poisson, donde la **media** y la **varianza** son iguales:

$$E(Y) = Var(Y) = \lambda$$

- **No sobredispersión:** Uno de los problemas comunes en los datos de conteo es la **sobredispersión**, que ocurre cuando la varianza de los datos es mayor que la media ($Var(Y) > E(Y)$). La presencia de sobredispersión indica que el modelo de Poisson puede no ser adecuado, y puede ser necesario considerar modelos alternativos como la **regresión binomial negativa**.
- **No exceso de ceros:** Si hay demasiados ceros en los datos (por ejemplo, en el número de accidentes en diferentes localidades donde muchas tienen cero accidentes), puede ser necesario utilizar modelos de **Poisson inflados en ceros (ZIP)** (Lambert 1992).

5.3.3 Interpretación de los resultados

La interpretación de los coeficientes en la regresión de Poisson difiere de la regresión lineal debido al uso de la función de enlace logarítmica.

Los coeficientes β representan el **logaritmo de la tasa** de eventos asociados con un cambio en la variable independiente. Para interpretar en términos de la tasa de ocurrencia, se utiliza el **exponencial de los coeficientes**:

$$e^{\beta_i}$$

Esto representa el **factor de cambio multiplicativo** en la tasa de eventos por cada unidad adicional en la variable X_i.

Ejemplo

Si $\beta_1 = 0.5$, entonces $e^{0.5} \approx 1.65$. Esto significa que por cada unidad adicional en X_1, la tasa de ocurrencia de eventos **aumenta en un 65%**.
Si $\beta_1 = -0.3$, entonces $e^{-0.3} \approx 0.74$. Esto indica que por cada unidad adicional en X_1, la tasa de eventos **disminuye en un 26%**.

Ejemplo

Vamos a utilizar R para ajustar un modelo de regresión de Poisson. Supongamos que tenemos datos sobre el **número de accidentes de tráfico** en diferentes intersecciones de una ciudad, junto con variables como el volumen de tráfico y la visibilidad.

```r
# Simulación de datos para el número de accidentes
set.seed(123)
n <- 100  # Número de observaciones

# Variables predictoras
trafico <- rnorm(n, mean = 1000, sd = 300)  # Volumen de tráfico en vehículos por día
visibilidad <- rnorm(n, mean = 5, sd = 2)   # Visibilidad en kilómetros

# Generar la tasa de accidentes (lambda) usando un modelo logarítmico
lambda <- exp(0.01 * trafico - 0.2 * visibilidad)

# Generar el número de accidentes como una variable de Poisson
accidentes <- rpois(n, lambda = lambda)

# Crear el data frame
datos_accidentes <- data.frame(accidentes, trafico, visibilidad)
head(datos_accidentes)
```

```
  accidentes   trafico visibilidad
1       2102  831.8573    3.579187
2       3744  930.9468    5.513767
3     959848 1467.6125    4.506616
4      11356 1021.1525    4.304915
5      17411 1038.7863    3.096763
6    1415794 1514.5195    4.909945
```

```r
# Ajustar el modelo de regresión de Poisson
modelo_poisson <- glm(accidentes ~ trafico + visibilidad, data = datos_accidentes, family =

# Resumen del modelo
summary(modelo_poisson)
```

```
Call:
glm(formula = accidentes ~ trafico + visibilidad, family = poisson,
    data = datos_accidentes)

Coefficients:
              Estimate Std. Error   z value Pr(>|z|)
(Intercept) -1.610e-03  2.116e-03    -0.761    0.447
trafico      1.000e-02  1.323e-06  7557.681   <2e-16 ***
```

```
visibilidad -2.001e-01  1.025e-04 -1951.765   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1.2621e+08  on 99  degrees of freedom
Residual deviance: 9.5511e+01  on 97  degrees of freedom
AIC: 1216.4

Number of Fisher Scoring iterations: 3
```

El coeficiente asociado a `trafico` indica cómo el volumen de tráfico afecta la tasa de accidentes.
El coeficiente asociado a `visibilidad` muestra cómo la visibilidad afecta la frecuencia de accidentes.

```
# Exponenciar los coeficientes para interpretar en términos de tasas
exp(coef(modelo_poisson))
```

```
(Intercept)     trafico visibilidad
  0.9983910   1.0100516   0.8186814
```

Un coeficiente positivo implica que un aumento en la variable está asociado con un **aumento en la tasa de accidentes**.
Un coeficiente negativo implica que un aumento en la variable está asociado con una **disminución en la tasa de accidentes**.

5.3.4 Evaluación del modelo de Poisson

La **sobredispersión** ocurre cuando la varianza de los datos es mayor que la media, lo que puede invalidar los supuestos de la regresión de Poisson.

Ejemplo

```
# Calcular la relación entre el deviance y los grados de libertad
deviance <- modelo_poisson$deviance
grados_libertad <- modelo_poisson$df.residual
sobredispersion <- deviance / grados_libertad

print(paste("Índice de Sobredispersión:", round(sobredispersion, 2)))
```

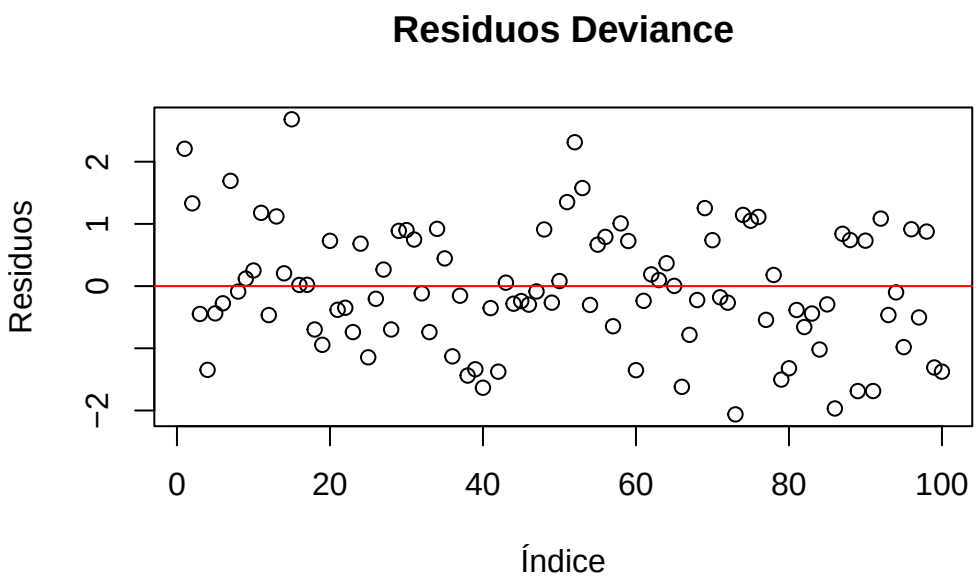
```
[1] "Índice de Sobredispersión: 0.98"
```

Un valor del índice de sobredispersión cercano a 1 sugiere que no hay sobredispersión. Por contra, un valor significativamente mayor que 1 sugiere la presencia de sobredispersión, y puede ser necesario considerar una **regresión binomial negativa**.

Diagnóstico de Residuos:

Ejemplo

```
# Gráfico de residuos deviance para evaluar el ajuste
plot(residuals(modelo_poisson, type = "deviance"), main = "Residuos Deviance", ylab = "Resi
abline(h = 0, col = "red")
```



Ejemplo sobre datos reales

Un conjunto de datos clásico en R es **warpbreaks**, que contiene el número de roturas de hilo en diferentes condiciones de tensión y longitud del hilo.

```r
# Datos de ejemplo: número de roturas de hilo
data(warpbreaks)

# Ajustar un modelo de Poisson para el número de roturas en función de la tensión
modelo_poisson_real <- glm(breaks ~ wool + tension, data = warpbreaks, family = poisson)

# Resumen del modelo
summary(modelo_poisson_real)
```

```
Call:
glm(formula = breaks ~ wool + tension, family = poisson, data = warpbreaks)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.69196    0.04541  81.302  < 2e-16 ***
woolB       -0.20599    0.05157  -3.994 6.49e-05 ***
tensionM    -0.32132    0.06027  -5.332 9.73e-08 ***
tensionH    -0.51849    0.06396  -8.107 5.21e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 297.37  on 53  degrees of freedom
Residual deviance: 210.39  on 50  degrees of freedom
AIC: 493.06

Number of Fisher Scoring iterations: 4
```

```r
# Interpretación de coeficientes
exp(coef(modelo_poisson_real))
```

```
(Intercept)       woolB    tensionM    tensionH
 40.1235380   0.8138425   0.7251908   0.5954198
```

5.3.5 Limitaciones y alternativas

Si la varianza de los datos es mayor que la media, el modelo de Poisson no será adecuado. En este caso, se recomienda utilizar la **regresión binomial negativa**, que introduce un parámetro adicional para manejar la sobredispersión.

Si hay más ceros de los esperados (por ejemplo, muchas intersecciones con cero accidentes), puede ser necesario utilizar modelos de **Poisson inflados en ceros (ZIP)** o **binomial negativa inflada en ceros (ZINB)**.

5.4 Otros GLMs

La **regresión binomial negativa** y los **modelos basados en distribuciones como Gamma e Inversa Gaussiana** amplían la capacidad de los **Modelos Lineales Generalizados (GLM)** para adaptarse a una amplia variedad de situaciones del mundo real. Estos modelos son especialmente útiles cuando los datos presentan características como sobredispersión, sesgo o restricciones en el dominio (por ejemplo, solo valores positivos). La elección adecuada del modelo y la función de enlace garantiza predicciones precisas y válidas, contribuyendo a la toma de decisiones informadas en campos como la salud, la ingeniería y la economía.

5.4.1 Regresión Binomial Negativa

Tal y como hemos visto en apartados anteriores, la **sobredispersión** ocurre cuando la varianza de los datos de conteo es **mayor que la media**, lo cual viola uno de los supuestos clave de la regresión de Poisson, que asume que la media y la varianza son iguales ($E(Y) = Var(Y)$). La sobredispersión puede surgir por varias razones:

- **Heterogeneidad no modelada:** Existen factores que afectan la variable dependiente pero no han sido incluidos en el modelo.
- **Dependencia entre eventos:** Los eventos no ocurren de forma independiente.
- **Exceso de ceros:** Hay más ceros en los datos de los que predice la distribución de Poisson.

Cuando la sobredispersión está presente, la regresión de Poisson subestima los errores estándar, lo que puede llevar a conclusiones incorrectas sobre la significancia de los predictores.

La **regresión binomial negativa** es una extensión de la regresión de Poisson que introduce un parámetro adicional para manejar la sobredispersión. Este modelo permite que la varianza sea mayor que la media:

$$Var(Y) = \lambda + \alpha\lambda^2$$

Donde α es el parámetro de dispersión. Si $\alpha = 0$, el modelo se reduce a la regresión de Poisson.

La forma funcional del modelo binomial negativa es similar al de Poisson:

$$\log(\lambda) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

Pero la varianza ahora incluye el término adicional α para capturar la sobredispersión.

Ejemplo

```r
# Instalar y cargar la librería MASS que contiene la función glm.nb
library(MASS)

# Ajuste de un modelo binomial negativo con los datos simulados de accidentes
modelo_binom_neg <- glm.nb(accidentes ~ trafico + visibilidad, data = datos_accidentes)
```

```
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
```

```
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
```

```
Warning in glm.nb(accidentes ~ trafico + visibilidad, data = datos_accidentes):
alternation limit reached
```

```r
# Resumen del modelo
summary(modelo_binom_neg)
```

```

Call:
glm.nb(formula = accidentes ~ trafico + visibilidad, data = datos_accidentes,
    init.theta = 245462679.3, link = log)

Coefficients:
              Estimate Std. Error  z value Pr(>|z|)
(Intercept) -1.620e-03  2.122e-03   -0.763    0.445
```

```
trafico      1.000e-02  1.329e-06  7523.491   <2e-16 ***
visibilidad -2.001e-01  1.036e-04 -1931.079   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(245406071) family taken to be 1)

    Null deviance: 1.2568e+08  on 99  degrees of freedom
Residual deviance: 9.5469e+01  on 97  degrees of freedom
AIC: 1218.5

Number of Fisher Scoring iterations: 1

              Theta:  245462679
          Std. Err.:  488358373
Warning while fitting theta: alternation limit reached

 2 x log-likelihood:  -1210.497
```

```
# Comparar la dispersión con el modelo de Poisson
cat("Dispersión en Poisson:", modelo_poisson$deviance / modelo_poisson$df.residual, "\n")
```

```
Dispersión en Poisson: 0.9846515
```

```
cat("Dispersión en Binomial Negativa:", modelo_binom_neg$theta, "\n")
```

```
Dispersión en Binomial Negativa: 245462679
```

- El parámetro θ (dispersión) ajustado en el modelo binomial negativa ayuda a corregir la varianza subestimada en el modelo de Poisson.
- Si θ es significativamente mayor que 1, se confirma la presencia de sobredispersión.

5.4.2 Modelos para variables continuas No Normales

Existen situaciones en las que la variable dependiente es **continua**, pero **no sigue una distribución normal**. En estos casos, los **Modelos Lineales Generalizados (GLM)** permiten utilizar distribuciones alternativas como **Gamma** o **Inversa Gaussiana**, junto con funciones de enlace específicas.

5.4.2.1 Regresión Gamma para datos positivos y sesgados

La **regresión Gamma** es adecuada para modelar variables continuas que son **positivas** y tienen una distribución **sesgada a la derecha**. Ejemplos típicos incluyen tiempos de espera, costos médicos o duración de procesos.

- La distribución Gamma asume que la variable dependiente es continua y positiva.
- La varianza de la variable dependiente aumenta proporcionalmente al cuadrado de la media.

Función de Enlace Común:

$$\log(\mu) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

Ejemplo

```r
# Simulación de costos médicos
set.seed(123)
n <- 100
ingresos <- rnorm(n, mean = 50000, sd = 10000)
edad <- rnorm(n, mean = 45, sd = 10)
costos <- rgamma(n, shape = 2, rate = 0.00005 * ingresos + 0.01 * edad)

# Ajuste del modelo Gamma
modelo_gamma <- glm(costos ~ ingresos + edad, family = Gamma(link = "log"))

# Resumen del modelo
summary(modelo_gamma)
```

```
Call:
glm(formula = costos ~ ingresos + edad, family = Gamma(link = "log"))

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.440e-01  5.294e-01   0.650   0.5173
ingresos    -1.807e-05  7.804e-06  -2.316   0.0227 *
edad         3.584e-03  7.366e-03   0.487   0.6277
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Gamma family taken to be 0.5010938)
```

```
    Null deviance: 60.771  on 99  degrees of freedom
Residual deviance: 58.345  on 97  degrees of freedom
AIC: 105.47

Number of Fisher Scoring iterations: 5
```

- Los coeficientes muestran cómo los ingresos y la edad afectan los costos médicos esperados.
- El enlace logarítmico asegura que las predicciones sean siempre positivas.

5.4.2.2 Regresión Inversa Gaussiana

La **regresión Inversa Gaussiana** es útil para modelar tiempos de respuesta o variables donde la varianza disminuye rápidamente a medida que la media aumenta. Este modelo se aplica en campos como la ingeniería, donde se analizan tiempos hasta fallas de sistemas.

Función de Enlace Común:

$$\frac{1}{\mu^2} = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

Ejemplo

```r
# Instalar y cargar la librería correcta
library(statmod)

# Simulación de datos
set.seed(123)
n <- 100
carga_trabajo <- rnorm(n, mean = 50, sd = 10)

# Generar tiempos hasta el fallo usando la distribución inversa gaussiana
# Aseguramos que los valores de carga_trabajo sean positivos para evitar problemas numéric
carga_trabajo[carga_trabajo <= 0] <- 1
tiempo_fallo <- rinvgauss(n, mean = 100 / carga_trabajo, dispersion = 1)

# Ajuste del modelo Inversa Gaussiana con enlace logarítmico
modelo_inversa_gauss <- glm(tiempo_fallo ~ carga_trabajo, family = inverse.gaussian(link =
```

```
Warning in sqrt(eta): NaNs produced

Warning: step size truncated due to divergence
```

```r
# Resumen del modelo
summary(modelo_inversa_gauss)
```

```
Call:
glm(formula = tiempo_fallo ~ carga_trabajo, family = inverse.gaussian(link = "1/mu^2"),
    start = c(0.01, 0.01))

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.106143   0.462230  -0.230    0.819
carga_trabajo  0.008261   0.009623   0.859    0.393

(Dispersion parameter for inverse.gaussian family taken to be 1.348442)

    Null deviance: 94.085  on 99  degrees of freedom
Residual deviance: 93.171  on 98  degrees of freedom
AIC: 294.2

Number of Fisher Scoring iterations: 5
```

5.5 Comparación de modelos y evaluación del ajuste

Una vez que se han ajustado varios modelos, es crucial evaluar su rendimiento y seleccionar el más adecuado para el problema en cuestión. La **evaluación del ajuste** y la **comparación de modelos** permiten identificar cuál modelo describe mejor los datos sin caer en el sobreajuste, es decir, ajustarse demasiado a los datos de entrenamiento a costa de un mal desempeño en datos nuevos.

Esta sección abordará los métodos más comunes para evaluar y comparar modelos, incluyendo criterios estadísticos, técnicas de validación y análisis de residuos.

5.5.1 La Deviance

La **deviance** (o **desviación**) es una medida estadística que evalúa qué tan bien se ajusta un modelo estadístico a los datos observados. Se utiliza principalmente en modelos de regresión que no se basan en supuestos de normalidad estricta, como los **modelos lineales generalizados** (GLM), incluyendo:

- Regresión logística

- Regresión de Poisson
- Modelos exponenciales, etc.

La deviance mide la **diferencia** entre el modelo propuesto y el **modelo saturado** (el modelo que predice perfectamente los datos observados).

- El **modelo saturado** tiene tantos parámetros como observaciones, por lo que ajusta cada punto de datos perfectamente.
- El **modelo propuesto** es el modelo que intentamos evaluar.

La deviance se interpreta como una generalización de la **suma de cuadrados de los residuos** utilizada en regresión lineal.

La deviance se define como:

$$D = 2 \sum_{i=1}^{n} [\ell(y_i; y_i) - \ell(y_i; \hat{y}_i)]$$

donde:

- D = Deviance
- $\ell(y_i; y_i)$ = Log-verosimilitud del modelo saturado (máxima verosimilitud posible)
- $\ell(y_i; \hat{y}_i)$ = Log-verosimilitud del modelo propuesto

En términos simples, mide cuánto peor es el modelo propuesto comparado con el modelo que se ajusta perfectamente.

5.5.1.1 Interpretación

- **Valores pequeños de deviance** → Indican que el modelo se ajusta bien a los datos.
- **Valores grandes de deviance** → Indican que el modelo no se ajusta bien.

Si el modelo es perfecto, la deviance es **cero**.

5.5.1.2 Deviance Residuals

En lugar de evaluar la deviance global, los **residuos de deviance** permiten identificar qué observaciones individuales no se ajustan bien.

$$d_i = \text{sign}(y_i - \hat{y}_i) \sqrt{2 \left[\ell(y_i; y_i) - \ell(y_i; \hat{y}_i)\right]}$$

Los residuos de deviance se comportan de forma similar a los residuos en regresión lineal, permitiendo identificar observaciones atípicas o mal ajustadas.

5.5.1.3 Relación con otros conceptos

- En **regresión lineal**, la deviance es equivalente a la **suma de los cuadrados de los residuos** (RSS).
- En **regresión logística**, la deviance se utiliza como alternativa a la suma de errores cuadráticos debido a que los residuos no se distribuyen normalmente.
- En **pruebas de bondad de ajuste**, se usa la **deviance nula** para comparar el modelo sin predictores (solo la media) con el modelo completo.

5.5.1.4 Ejemplo en Regresión Logística

Supongamos que estamos modelando la probabilidad de que una persona compre un producto en función de su edad. La deviance nos ayuda a evaluar qué tan bien el modelo logra predecir esas probabilidades comparado con un modelo perfecto.

- **Deviance baja:** El modelo ajusta bien las probabilidades.
- **Deviance alta:** El modelo falla en capturar los patrones en los datos.

5.5.2 Criterios de selección de modelos (AIC, BIC)

Los criterios de selección de modelos como el **AIC** y el **BIC** permiten comparar modelos que han sido ajustados a los mismos datos, penalizando la complejidad para evitar el sobreajuste.

Akaike Information Criterion (AIC)

El **Criterio de Información de Akaike (AIC)** es una medida que equilibra la calidad del ajuste y la complejidad del modelo. Se calcula como:

$$AIC = -2\log(L) + 2k$$

Donde:

- L es el **log-likelihood** del modelo (medida de la probabilidad de los datos dados los parámetros del modelo).
- k es el número de parámetros del modelo.

Respecto a su interpretación, el AIC **no tiene una interpretación absoluta**, solo es útil para comparar modelos ajustados a los mismos datos. Un **AIC menor** indica un mejor equilibrio entre ajuste y simplicidad.

Bayesian Information Criterion (BIC)

El **Criterio de Información Bayesiano (BIC)** es similar al AIC, pero penaliza más severamente la complejidad del modelo, especialmente cuando el número de observaciones es grande. Se calcula como:

$$BIC = -2\log(L) + k\log(n)$$

Donde: - n es el número de observaciones.

El BIC favorece modelos más simples en comparación con el AIC. Un **BIC menor** indica un mejor modelo.

Ejemplo: comparación de AIC y BIC

```r
# Comparación de modelos: Poisson vs Binomial Negativa

# Modelo de Poisson
modelo_poisson <- glm(accidentes ~ trafico + visibilidad, family = poisson, data = datos_ac

# Modelo Binomial Negativa
library(MASS)
modelo_binom_neg <- glm.nb(accidentes ~ trafico + visibilidad, data = datos_accidentes)
```

```
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
```

```
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached
Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
control$trace > : iteration limit reached

Warning in glm.nb(accidentes ~ trafico + visibilidad, data = datos_accidentes):
alternation limit reached
```

```
# Calcular AIC y BIC
AIC(modelo_poisson, modelo_binom_neg)
```

```
                 df      AIC
modelo_poisson    3 1216.421
modelo_binom_neg  4 1218.497
```

```r
BIC(modelo_poisson, modelo_binom_neg)
```

```
                df      BIC
modelo_poisson   3 1224.236
modelo_binom_neg 4 1228.918
```

El modelo con el menor **AIC** o **BIC** es preferido. Si ambos modelos tienen valores similares, se puede preferir el modelo más simple (con menos parámetros).

5.5.3 Validación cruzada y técnicas de evaluación predictiva

La **validación cruzada** y otras técnicas de evaluación predictiva permiten estimar cómo se desempeñará un modelo en datos no vistos, lo que es esencial para evitar el sobreajuste.

La **validación cruzada k-fold** divide el conjunto de datos en **k** subconjuntos (o "folds"). El modelo se ajusta **k** veces, cada vez utilizando $k-1$ folds para el entrenamiento y el fold restante para la prueba. El rendimiento se promedia sobre todas las iteraciones. La validación cruzada utiliza eficientemente los datos disponibles, proporcionando una estimación robusta del rendimiento del modelo.

Ejemplo: Validación cruzada

```r
# Instalar y cargar la librería caret
library(caret)
```

```
Loading required package: ggplot2
```

```
Loading required package: lattice
```

```r
# Definir la validación cruzada de 5-fold
control <- trainControl(method = "cv", number = 5)

# Ajustar un modelo de regresión logística con validación cruzada
modelo_cv <- train(type ~ npreg + glu + bmi, data = Pima.tr, method = "glm", family = "bino

# Resultados de la validación cruzada
print(modelo_cv)
```

```
Generalized Linear Model

200 samples
```

```
  3 predictor
  2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 161, 160, 161, 159, 159
Resampling results:

  Accuracy   Kappa
  0.7603815  0.4378568
```

Otra técnica común es dividir el conjunto de datos en un **conjunto de entrenamiento** (por ejemplo, el 70%) y un **conjunto de prueba** (30%). El modelo se ajusta en el conjunto de entrenamiento y se evalúa en el conjunto de prueba.

Ejemplo: Train/Test

```r
# Dividir el conjunto de datos en entrenamiento y prueba
set.seed(123)
library(caret)
indices <- createDataPartition(Pima.tr$type, p = 0.7, list = FALSE)
entrenamiento <- Pima.tr[indices, ]
prueba <- Pima.tr[-indices, ]

# Ajustar el modelo en el conjunto de entrenamiento
modelo_entrenamiento <- glm(type ~ npreg + glu + bmi, data = entrenamiento, family = binomi

# Realizar predicciones en el conjunto de prueba
predicciones <- predict(modelo_entrenamiento, newdata = prueba, type = "response")

# Evaluar precisión
predicciones_clase <- ifelse(predicciones > 0.5, "Yes", "No")
confusionMatrix(as.factor(predicciones_clase), as.factor(prueba$type))
```

```
Confusion Matrix and Statistics

          Reference
Prediction No Yes
       No  34  10
       Yes  5  10

               Accuracy : 0.7458
```

```
               95% CI : (0.6156, 0.8502)
    No Information Rate : 0.661
    P-Value [Acc > NIR] : 0.1062

                  Kappa : 0.3959

 Mcnemar's Test P-Value : 0.3017

            Sensitivity : 0.8718
            Specificity : 0.5000
         Pos Pred Value : 0.7727
         Neg Pred Value : 0.6667
             Prevalence : 0.6610
         Detection Rate : 0.5763
   Detection Prevalence : 0.7458
      Balanced Accuracy : 0.6859

       'Positive' Class : No
```

5.5.4 Diagnóstico de residuos y buenas prácticas

El análisis de **residuos** es fundamental para evaluar el ajuste del modelo y detectar problemas como la falta de ajuste, valores atípicos o violaciones de los supuestos del modelo.

Los **residuos deviance** son una medida común en los **Modelos Lineales Generalizados (GLM)**. Representan la diferencia entre el modelo ajustado y el modelo perfecto (donde la predicción es exactamente igual al valor observado).

Tipos de residuos:

- **Residuos Pearson:**

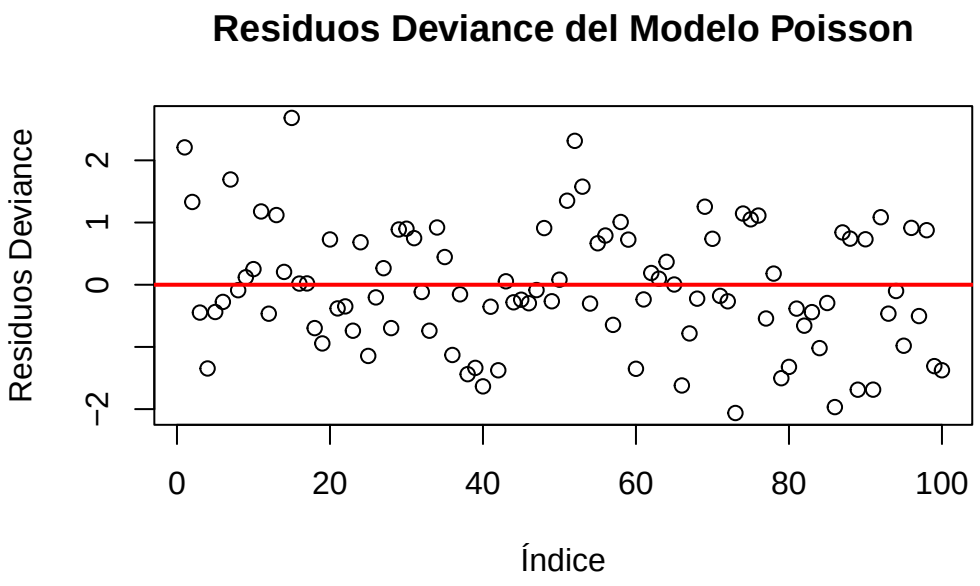
$$r_i = \frac{y_i - \hat{y}_i}{\sqrt{\hat{V}(y_i)}}$$

  Donde $\hat{V}(y_i)$ es la varianza estimada de y_i.

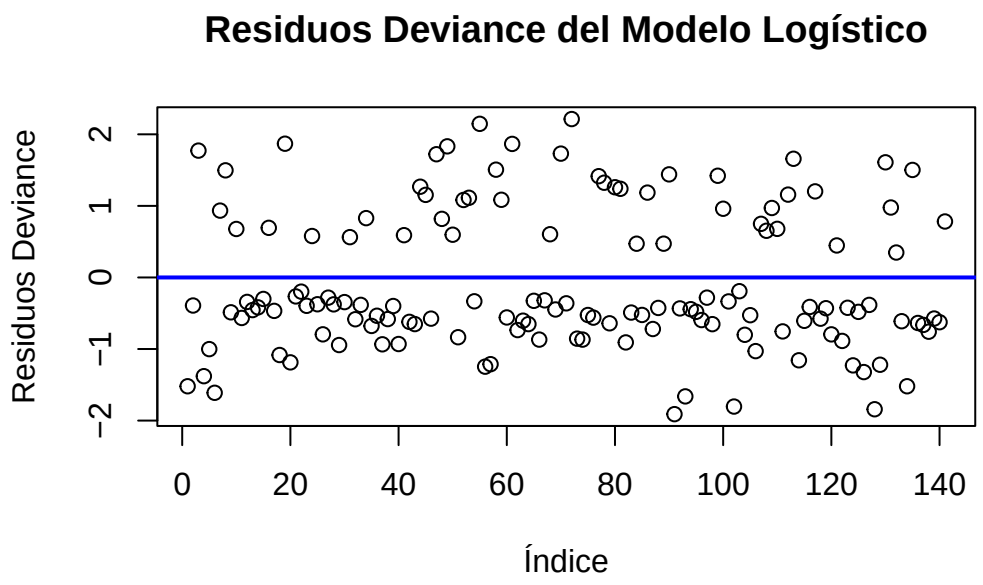
- **Residuos Deviance:**
  Representan la contribución de cada observación al deviance total del modelo.

Ejemplo: Análisis de los residuos

```r
# Gráfico de residuos deviance para un modelo de Poisson
plot(residuals(modelo_poisson, type = "deviance"),
     main = "Residuos Deviance del Modelo Poisson",
     ylab = "Residuos Deviance", xlab = "Índice")
abline(h = 0, col = "red", lwd = 2)
```




```r
# Gráfico de residuos para la regresión logística
plot(residuals(modelo_entrenamiento, type = "deviance"),
     main = "Residuos Deviance del Modelo Logístico",
     ylab = "Residuos Deviance", xlab = "Índice")
abline(h = 0, col = "blue", lwd = 2)
```

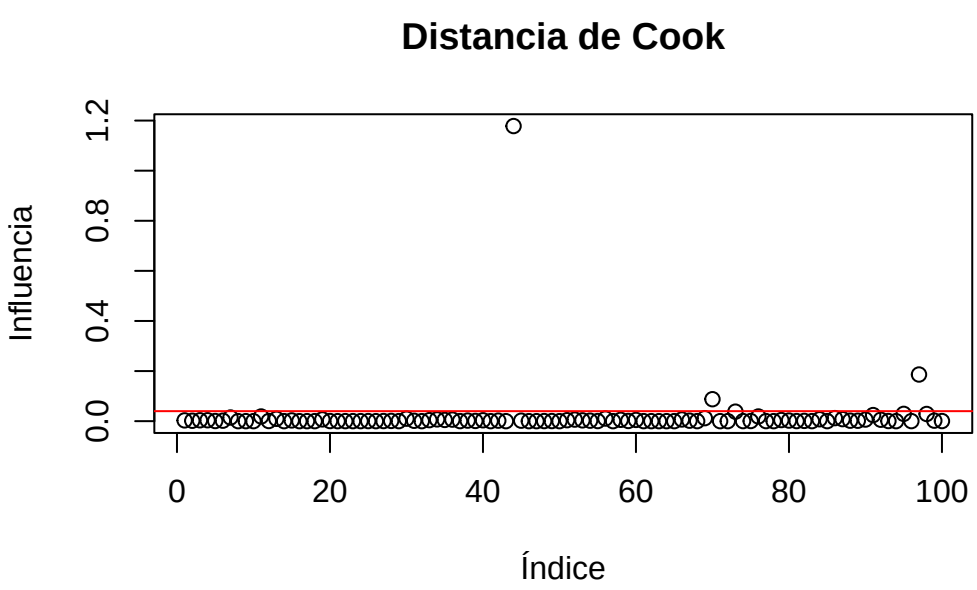


Para detectar valores atípicos y su influencia emplearemos:

- **Distancia de Cook:** Identifica observaciones influyentes que tienen un impacto significativo en los coeficientes del modelo.
- **Leverage:** Mide el impacto potencial de una observación en el ajuste del modelo.

Ejemplo: Detección de observaciones influyentes

```
# Distancia de Cook para identificar observaciones influyentes
cooksd <- cooks.distance(modelo_poisson)

# Gráfico de la distancia de Cook
plot(cooksd, main = "Distancia de Cook", ylab = "Influencia", xlab = "Índice")
abline(h = 4 / length(cooksd), col = "red")  # Línea de referencia
```


Distancia de Cook
Influencia
0.0
0.4
0.8
1.2
0
20
40
60
80
100
Índice

6 Otros modelos de regresión: Modelos Aditivos Generalizados (GAMs)

En el análisis de datos y la modelización estadística, a menudo asumimos que las relaciones entre las variables independientes y la variable dependiente son lineales o que pueden transformarse fácilmente para cumplir con esta suposición. Sin embargo, en muchos contextos del mundo real, las relaciones entre las variables son **no lineales** y **complejas**, lo que limita la efectividad de los modelos de regresión tradicionales como la regresión lineal o polinomial.

Los **Modelos de Regresión Aditiva Generalizada (GAMs)** ofrecen una solución poderosa y flexible para este problema. Los GAMs permiten modelar relaciones no lineales sin necesidad de especificar de antemano la forma exacta de la no linealidad (Hastie 2017). En lugar de ajustar una única función global para todos los predictores, los GAMs aplican funciones de suavizado a cada variable independiente por separado, lo que permite capturar patrones complejos y sutiles en los datos.

Un **Modelo Aditivo Generalizado (GAM)** es una extensión de los **Modelos Lineales Generalizados (GLM)** vistos en el tema anterior y que permite que la relación entre la variable dependiente y las variables independientes sea **no lineal** y **flexible**. En un GAM, la variable dependiente se modela como una suma de funciones suavizadas de las variables independientes:

$$g(\mu) = \beta_0 + f_1(X_1) + f_2(X_2) + \cdots + f_p(X_p)$$

Donde:

- $g(\mu)$ es la **función de enlace** que conecta la media de la variable dependiente (μ) con los predictores.
- β_0 es el **intercepto** del modelo.
- $f_i(X_i)$ son funciones suavizadas que capturan la relación entre cada predictor X_i y la variable dependiente.

A diferencia de la regresión lineal, donde los predictores tienen una relación lineal con la variable dependiente, en los GAMs cada predictor puede tener una forma funcional diferente, permitiendo capturar **curvas**, **patrones no lineales** y **efectos complejos** en los datos.

Ventajas de los GAMs

Flexibilidad para modelar No Linealidades:
Los GAMs permiten capturar relaciones no lineales complejas sin necesidad de especificar una forma funcional exacta.
Interpretabilidad:
A pesar de su flexibilidad, los GAMs siguen siendo interpretables, ya que el efecto de cada predictor puede visualizarse por separado.
Evita el sobreajuste:
A través de técnicas de suavizado controlado, los GAMs pueden evitar el sobreajuste al equilibrar la complejidad del modelo con la precisión de las predicciones.
Adaptabilidad a diferentes tipos de datos:
Los GAMs pueden aplicarse a variables continuas, categóricas y de conteo, lo que los hace útiles en una amplia variedad de contextos.

Los principales casos de uso de los GAMS son:

- **Relaciones no lineales complejas:** Cuando se sospecha que la relación entre las variables no es lineal y los modelos polinomiales no capturan adecuadamente la estructura de los datos.
- **Interacciones complejas entre variables:** Cuando los efectos de las variables pueden variar dependiendo del contexto o de otros predictores.
- **Datos con distribuciones no estándar:** En combinación con funciones de enlace, los GAMs pueden manejar diferentes tipos de distribuciones en la variable dependiente, como datos de conteo, binarios o continuos sesgados.

Los **Modelos Aditivos Generalizados (GAMs)** tienen aplicaciones en una amplia variedad de disciplinas debido a su capacidad para capturar relaciones no lineales complejas. En **medicina y epidemiología**, se utilizan para modelar el riesgo de enfermedades en función de múltiples factores de riesgo que interactúan de manera no lineal, permitiendo identificar patrones sutiles en la salud de las poblaciones. En **economía y finanzas**, los GAMs son útiles para analizar la relación entre variables económicas, como la inflación y el crecimiento del PIB, donde las interacciones y los efectos pueden variar a lo largo del tiempo. En el campo de las **ciencias ambientales**, permiten modelar la relación entre la temperatura y la concentración de contaminantes atmosféricos, lo cual es crucial para entender el impacto del cambio climático. Finalmente, en **marketing y negocios**, los GAMs ayudan a analizar el comportamiento del cliente, como la probabilidad de compra, en función de variables como el ingreso y la edad, proporcionando insights valiosos para la toma de decisiones estratégicas.

6.1 Fundamentos de los GAMs

Los **Modelos Aditivos Generalizados (GAMs)** son una extensión de los **Modelos Lineales Generalizados (GLM)** que permiten capturar relaciones **no lineales** entre la variable dependiente y las variables independientes. Mientras que los GLM asumen una relación lineal (o lineal después de una transformación mediante una función de enlace), los GAMs relajan esta suposición al permitir que cada predictor tenga su propia forma funcional no paramétrica.

Un **GLM** se expresa como:

$$g(\mu) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

Donde: - $g(\mu)$ es la función de enlace que relaciona la media de la variable dependiente (μ) con el predictor lineal. - $\beta_0, \beta_1, \ldots, \beta_p$ son los coeficientes del modelo que representan el efecto lineal de cada predictor.

En contraste, un **GAM** se define como:

$$g(\mu) = \beta_0 + f_1(X_1) + f_2(X_2) + \cdots + f_p(X_p)$$

Donde $f_i(X_i)$ son **funciones suavizadas** que capturan la relación (posiblemente no lineal) entre el predictor X_i y la variable dependiente.

En los GLM, los efectos de los predictores son estrictamente lineales o transformados de forma lineal. En los GAMs, la relación puede ser cualquier forma no paramétrica, determinada por los datos. Los GAMs ofrecen mayor flexibilidad al permitir que la forma de la relación entre cada predictor y la respuesta sea modelada directamente a partir de los datos. Aunque los GAMs son más flexibles, siguen siendo interpretables, ya que el efecto de cada variable puede visualizarse y analizarse individualmente.

6.1.1 Suavizado en los GAMs

El componente fundamental de los GAMs es el uso de **funciones de suavizado**, que permiten modelar relaciones no lineales de manera flexible y controlada. El suavizado evita el sobreajuste (overfitting) al no intentar seguir cada fluctuación en los datos, sino al capturar las **tendencias generales** subyacentes.

El **suavizado** consiste en ajustar una curva a los datos de tal manera que se capturen las tendencias generales sin que el modelo sea demasiado sensible al ruido o a las fluctuaciones

aleatorias. En el contexto de los GAMs, cada predictor tiene su propia función de suavizado que determina cómo se ajusta la relación entre esa variable y la variable dependiente.

$$g(\mu) = \beta_0 + s_1(X_1) + s_2(X_2) + \cdots + s_p(X_p)$$

Donde $s_i(X_i)$ representa una función suavizada para el predictor X_i.

El grado de suavizado controla cuánto sigue el modelo las fluctuaciones de los datos:

- **Suavizado bajo:** El modelo se ajusta demasiado a los datos, capturando incluso el ruido aleatorio. Esto puede llevar al **sobreajuste**.
- **Suavizado alto:** El modelo puede no capturar adecuadamente la estructura subyacente de los datos, llevando al **subajuste** (underfitting).

El **criterio de suavizado óptimo** se selecciona automáticamente mediante técnicas como la **minimización del criterio de información de Akaike (AIC)** o el uso de **validación cruzada**.

El **suavizado** en los **Modelos Aditivos Generalizados (GAMs)** ofrece múltiples ventajas que los convierten en una herramienta poderosa para el análisis de datos complejos. En primer lugar, permite **capturar no linealidades complejas**, detectando patrones que no pueden ser representados adecuadamente por términos lineales o polinomiales simples. Esta flexibilidad es crucial para modelar relaciones reales que rara vez son estrictamente lineales. Además, el suavizado ayuda a **evitar el sobreajuste**; a diferencia de los polinomios de alto grado, que pueden generar oscilaciones indeseadas y seguir de manera excesiva las fluctuaciones del ruido en los datos, el suavizado controlado proporciona una representación más estable y generalizable de la relación entre las variables. Finalmente, una de las características más valiosas del suavizado en GAMs es su **interpretación intuitiva**. Las funciones suavizadas pueden visualizarse de manera clara y directa, lo que facilita la comprensión del impacto de cada predictor sobre la variable de respuesta, haciendo que los GAMs sean no solo potentes, sino también accesibles desde el punto de vista interpretativo.

6.1.2 Splines

En los GAMs, las funciones de suavizado se implementan comúnmente mediante **splines**, que son funciones polinómicas definidas por tramos. Estas permiten una flexibilidad controlada al ajustar diferentes tramos de los datos mientras se mantiene la continuidad y la suavidad en los puntos de unión (nudos).

- **Splines Lineales:**
  Son polinomios de primer grado ajustados por tramos. Aunque permiten cierta flexibilidad, pueden generar ángulos agudos en los puntos de unión.

- **Splines Cúbicos:**
  Utilizan polinomios de tercer grado en cada tramo, asegurando continuidad en la primera y segunda derivada en los **nudos**. Los **splines cúbicos** son los más utilizados en la práctica debido a su capacidad para capturar curvaturas suaves sin introducir oscilaciones no deseadas.

Los **splines penalizados** añaden una penalización al modelo para controlar la suavidad de la curva. Esto se logra añadiendo un término de penalización al proceso de ajuste que limita la complejidad de la función suavizada.

$$\min \left(\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int [f''(x)]^2 dx \right)$$

Donde:

- λ es el **parámetro de suavizado** que controla el equilibrio entre el ajuste a los datos y la suavidad de la curva.
- Si λ es grande, el modelo será más suave; si es pequeño, el modelo se ajustará más a los datos.

El número y la ubicación de los **nudos** (puntos donde cambian los tramos polinómicos) es un aspecto crucial en el ajuste de splines:

- **Muchos nudos:** Mayor flexibilidad, pero riesgo de sobreajuste.
- **Pocos nudos:** Modelo más simple, pero riesgo de no capturar la estructura subyacente de los datos.

La elección óptima del número de nudos puede realizarse mediante criterios automáticos como el **AIC** o mediante validación cruzada.

Ejemplo: Ajuste de un GAM con Splines

Vamos a ajustar un **GAM** utilizando la librería `mgcv` en R, que es una de las herramientas más utilizadas para trabajar con GAMs.

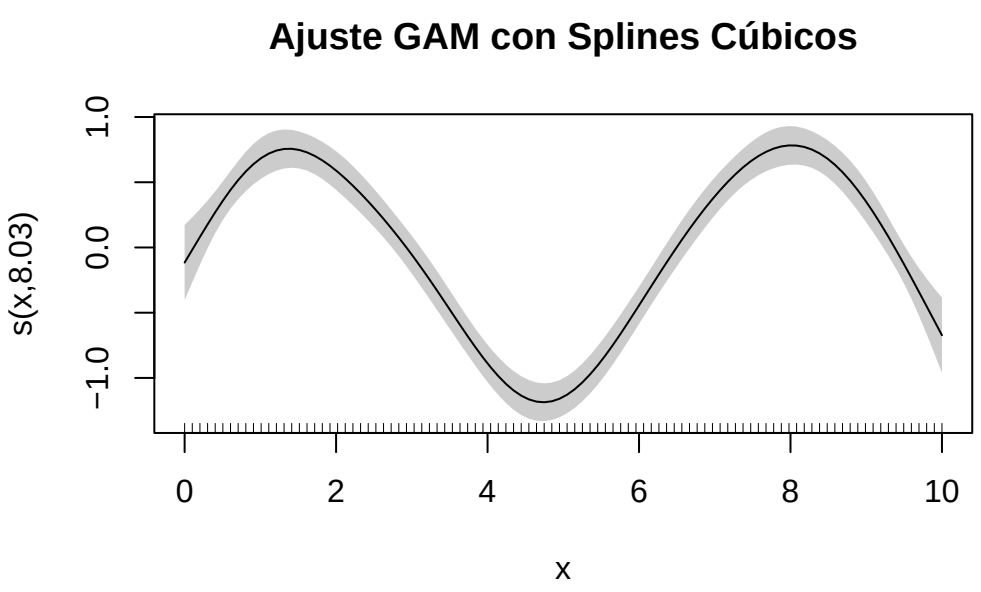
```
# Instalar y cargar la librería mgcv
library(mgcv)
```

```
Loading required package: nlme

This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```r
# Simulación de datos
set.seed(123)
n <- 100
x <- seq(0, 10, length.out = n)
y <- sin(x) + rnorm(n, sd = 0.3)

# Ajuste de un GAM con splines cúbicos
modelo_gam <- gam(y ~ s(x), method = "REML")

# Resumen del modelo
summary(modelo_gam)
```

```
Family: gaussian
Link function: identity

Formula:
y ~ s(x)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.20631    0.02749   7.505 4.01e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
      edf Ref.df     F p-value
s(x) 8.03   8.75 62.31  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.846   Deviance explained = 85.9%
-REML =  30.36  Scale est. = 0.075575  n = 100
```

```r
# Visualización de la función suavizada
plot(modelo_gam, main = "Ajuste GAM con Splines Cúbicos", shade = TRUE)
```



6.2 Interpretación de los resultados

Después de ajustar el modelo, el siguiente paso es interpretar los resultados proporcionados por la función `summary()` de R.

El comando `summary(modelo_gam)` proporciona la siguiente información clave:

1. **Resumen de la suavización:**
   - **`edf (effective degrees of freedom):`** Indica el grado de flexibilidad del suavizado.
     - Un **edf cercano a 1** sugiere una relación lineal.
     - Un **edf mayor que 1** indica una relación no lineal.
2. **Significancia de los predictores:**
   - **`p-values:`** Indican si la función suavizada para cada predictor es significativa. Un valor $p < 0.05$ sugiere que la relación entre el predictor y la variable dependiente es estadísticamente significativa.
3. **Medidas de ajuste:**

- **`Deviance explained`:** Similar al R^2 en regresión lineal, indica el porcentaje de la variabilidad de los datos explicada por el modelo.
- **`GCV score` y `AIC`:** Utilizados para evaluar la calidad del ajuste y comparar diferentes modelos.

Ejemplo

```
summary(modelo_gam)
```

```
Family: gaussian
Link function: identity

Formula:
y ~ s(x)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.20631    0.02749   7.505 4.01e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
      edf Ref.df     F p-value
s(x) 8.03   8.75 62.31  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.846   Deviance explained = 85.9%
-REML =  30.36  Scale est. = 0.075575  n = 100
```

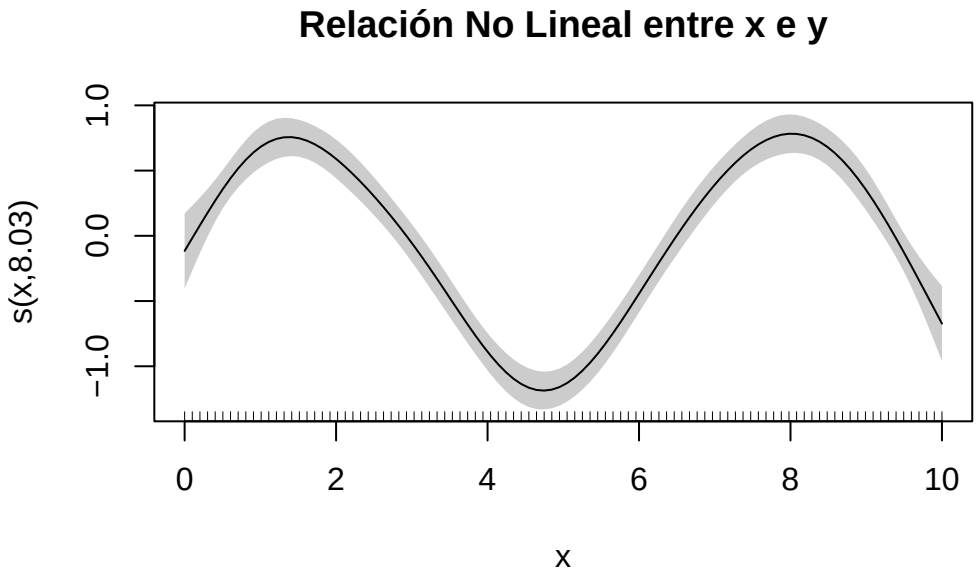
- El **intercepto** (0.01852) no es significativo, lo que sugiere que el valor medio de y cuando $x = 0$ no difiere significativamente de cero.
- La función suavizada $s(x)$ tiene un **edf de 7.54**, indicando que la relación entre x e y es altamente no lineal.
- El **valor p (<2e-16)** para $s(x)$ sugiere que la relación no lineal es estadísticamente significativa.
- El **82.1% de la devianza explicada** indica que el modelo captura bien la variabilidad de los datos.
- Un **GCV (Generalized Cross-Validation)** bajo indica un buen ajuste.

Una de las principales ventajas de los GAMs es la posibilidad de visualizar fácilmente la

relación entre cada predictor y la variable dependiente. La función `plot()` de `mgcv` permite crear gráficos claros e intuitivos de los efectos suavizados.

Ejemplo

```
# Visualización del efecto suavizado de x en y
plot(modelo_gam, shade = TRUE, main = "Relación No Lineal entre x e y")
```

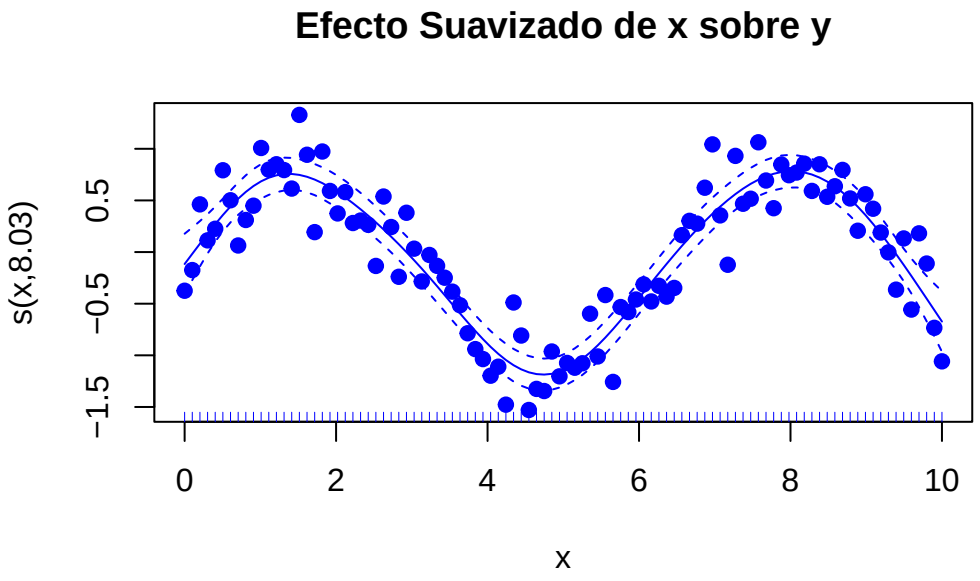


El **área sombreada** alrededor de la curva representa el **intervalo de confianza** al 95%. La **forma de la curva** muestra la relación entre x e $ y$; en este caso, debería reflejar una forma sinusoidal. Si la curva es recta, la relación es aproximadamente lineal.

Es posible personalizar el gráfico para mejorar la presentación:

Ejemplo

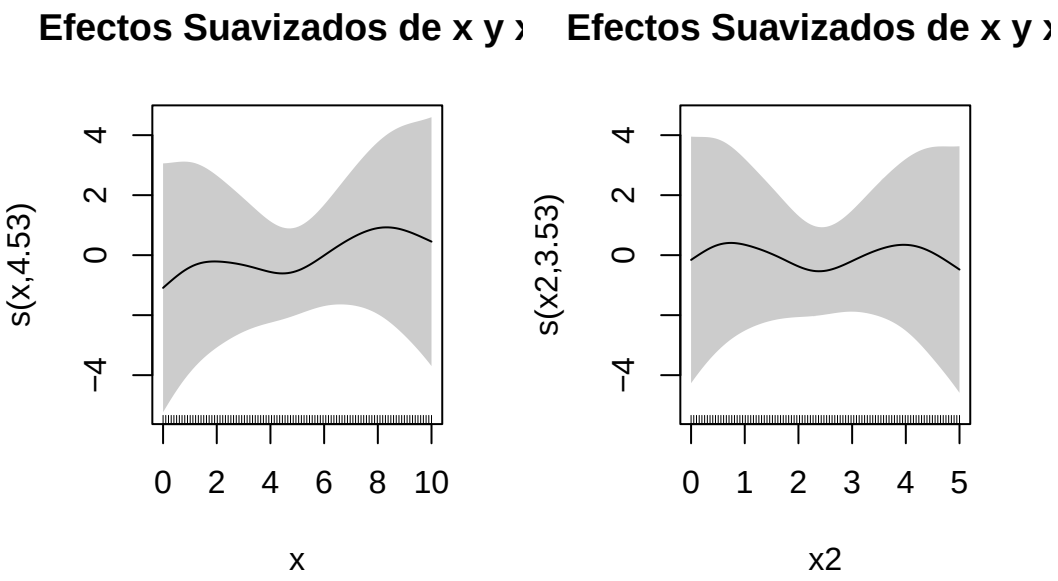
```
# Personalización avanzada del gráfico
plot(modelo_gam,
     residuals = TRUE,  # Muestra los residuos
     pch = 19,          # Estilo de los puntos de datos
     col = "blue",      # Color de la curva suavizada
     seWithMean = TRUE, # Muestra intervalos de confianza ajustados al promedio
     rug = TRUE,        # Añade marcas en el eje x para indicar la densidad de los datos
     main = "Efecto Suavizado de x sobre y")
```



Si el modelo incluye múltiples predictores suavizados, `plot()` creará un gráfico para cada uno.

Ejemplo

```r
# Simulación de un segundo predictor
set.seed(123)
x2 <- seq(0, 5, length.out = n)
y2 <- sin(x) + log(x2 + 1) + rnorm(n, sd = 0.3)

# Ajuste del GAM con dos predictores suavizados
modelo_gam_multi <- gam(y2 ~ s(x) + s(x2), method = "REML")

# Visualización de los efectos suavizados
plot(modelo_gam_multi, pages = 1, shade = TRUE, main = "Efectos Suavizados de x y x2")
```



El parámetro `pages = 1` muestra todos los efectos en una sola página. Cada gráfico muestra cómo cada predictor afecta la variable dependiente, permitiendo una interpretación clara de efectos individuales.

Un valor de **`edf` cercano a 1** indica un efecto lineal, mientras que valores mayores sugieren una relación no lineal más compleja. La **significancia estadística** de los efectos suavizados indica qué predictores tienen una relación significativa con la variable dependiente. La **devianza explicada** y el **AIC** proporcionan medidas para evaluar la calidad del ajuste y comparar diferentes modelos. Los gráficos permiten identificar patrones no lineales complejos y facilitan la comunicación de los resultados a audiencias no técnicas.

6.3 Evaluación del modelo y selección de parámetros en GAMs

Una vez ajustado un **Modelo Aditivo Generalizado (GAM)**, es fundamental evaluar su calidad y ajustar adecuadamente el **grado de suavizado** para garantizar que el modelo capture las relaciones relevantes sin caer en el **sobreajuste** o **subajuste**. Esta sección explora las técnicas para evaluar el rendimiento de los GAMs, identificar problemas en el ajuste y seleccionar los parámetros óptimos de suavizado.

6.3.1 Técnicas para evaluar la calidad del ajuste en GAMs

La evaluación de la calidad del ajuste en GAMs implica una combinación de **métricas estadísticas** y **diagnósticos gráficos**. Estas herramientas permiten determinar qué tan bien el modelo se ajusta a los datos y si los supuestos subyacentes son válidos.

La **deviance explicada** es una medida análoga al R^2 en la regresión lineal. Indica la proporción de la variabilidad de la variable dependiente que es explicada por el modelo:

$$\text{Deviance Explicada} = 1 - \frac{\text{Deviance del Modelo}}{\text{Deviance del Modelo Nulo}}$$

- **Valores cercanos a 1** indican que el modelo explica bien la variabilidad de los datos.
- **Valores cercanos a 0** sugieren que el modelo no captura adecuadamente la estructura de los datos.

El R^2 **ajustado** también puede interpretarse en modelos GAM cuando la familia es `gaussian`. Este valor se reporta en la salida de `summary(modelo_gam)`.

6.3.2 Criterios de Información (AIC, GCV)

Los **criterios de información** permiten comparar modelos y evaluar su capacidad para generalizar a nuevos datos. Dos de las métricas más comunes en GAMs son:

AIC (Akaike Information Criterion):

$$AIC = -2\log(L) + 2k$$

Visto en temas anteriores, donde L es el log-likelihood del modelo y k es el número de parámetros. **Un AIC más bajo** sugiere un modelo mejor ajustado.

GCV (Generalized Cross-Validation):

El **GCV** es una medida específica para modelos de suavizado y penalización. Estima el error de predicción esperado usando una forma eficiente de validación cruzada. Un **GCV bajo** indica un buen ajuste sin sobreajuste.

6.3.3 Análisis de residuos

Como en otros modelos de regresión, el análisis de residuos es una herramienta esencial para diagnosticar el ajuste del modelo y detectar patrones no explicados.

- **Residuos Pearson y Deviance:**
  Deben distribuirse aleatoriamente alrededor de cero si el modelo está bien especificado.
- **Gráficos de residuos:**
  Permiten identificar valores atípicos, heterocedasticidad y patrones no capturados por el modelo.

Ejemplo: Evaluación del ajuste de un GAM

```r
# Ajuste del modelo GAM
library(mgcv)
set.seed(123)
n <- 200
x <- seq(0, 10, length.out = n)
y <- sin(x) + rnorm(n, sd = 0.3)

modelo_gam <- gam(y ~ s(x), method = "REML")

# Resumen del modelo
summary(modelo_gam)
```

```
Family: gaussian
Link function: identity

Formula:
y ~ s(x)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.17902    0.02011   8.902  4.3e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
       edf Ref.df     F p-value
```

```
s(x) 8.414  8.904 123.7  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.847   Deviance explained = 85.3%
-REML =  52.84  Scale est. = 0.080887  n = 200
```

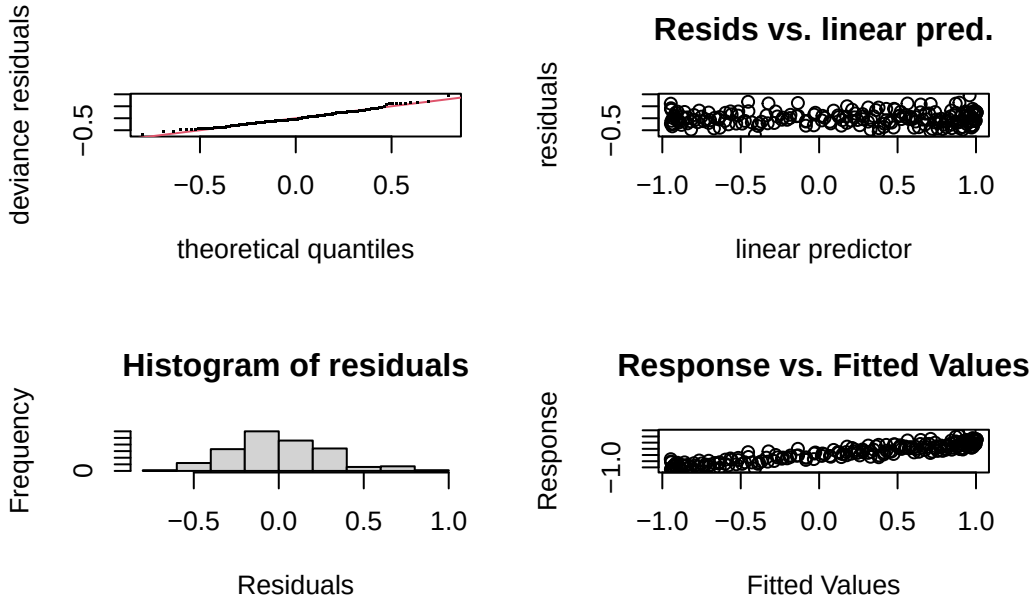
```
# Verificar la deviance explicada y el AIC
cat("Deviance Explicada:", summary(modelo_gam)$dev.expl, "\n")
```

```
Deviance Explicada: 0.8533086
```

```
cat("AIC del Modelo:", AIC(modelo_gam), "\n")
```

```
AIC del Modelo: 75.89817
```

```
# Diagnóstico de residuos
par(mfrow = c(2, 2))
gam.check(modelo_gam)
```



```
Method: REML   Optimizer: outer newton
full convergence after 8 iterations.
Gradient range [-3.529919e-09,3.476724e-09]
(score 52.83958 & scale 0.08088732).
Hessian positive definite, eigenvalue range [3.67346,99.14414].
Model rank =  10 / 10

Basis dimension (k) checking results. Low p-value (k-index<1) may
indicate that k is too low, especially if edf is close to k'.

       k'  edf k-index p-value
s(x) 9.00 8.41     1.1    0.94
```

El comando `gam.check()` proporciona múltiples gráficos de diagnóstico para evaluar la adecuación del modelo, incluyendo la distribución de los residuos y la comprobación del grado de suavizado.

6.3.4 Selección del grado de suavizado y control del sobreajuste

El **grado de suavizado** en los GAMs controla la flexibilidad del modelo. Un suavizado adecuado permite capturar la estructura subyacente de los datos sin ajustarse al ruido aleatorio.

El **grado de suavizado** determina la complejidad de la función que se ajusta a los datos. Esto se representa mediante los **grados de libertad efectivos (edf)**:

- **edf cercano a 1:** Indica una relación casi lineal.
- **edf mayor que 1:** Sugiere una relación no lineal más compleja.

Un **grado de suavizado demasiado bajo** puede llevar al **subajuste** (el modelo no captura adecuadamente los patrones de los datos), mientras que un **suavizado excesivo** puede llevar al **sobreajuste** (el modelo sigue el ruido en lugar de la tendencia general).

La librería `mgcv` ajusta automáticamente el grado de suavizado utilizando métodos de optimización como **REML** (Restricted Maximum Likelihood) o **GCV** (Generalized Cross-Validation).

- **`method = "REML"`:** Proporciona un ajuste más robusto y es menos propenso al sobreajuste que GCV.
- **`method = "GCV.Cp"`:** Utiliza la validación cruzada para seleccionar el suavizado, pero puede ser más sensible al ruido.

Ejemplo: Comparación de métodos de suavizado

```
# Ajuste usando REML
modelo_reml <- gam(y ~ s(x), method = "REML")

# Ajuste usando GCV
modelo_gcv <- gam(y ~ s(x), method = "GCV.Cp")

# Comparación de AIC y GCV
cat("AIC (REML):", AIC(modelo_reml), "\n")
```

```
AIC (REML): 75.89817
```

```
cat("GCV (GCV.Cp):", modelo_gcv$gcv.ubre, "\n")
```

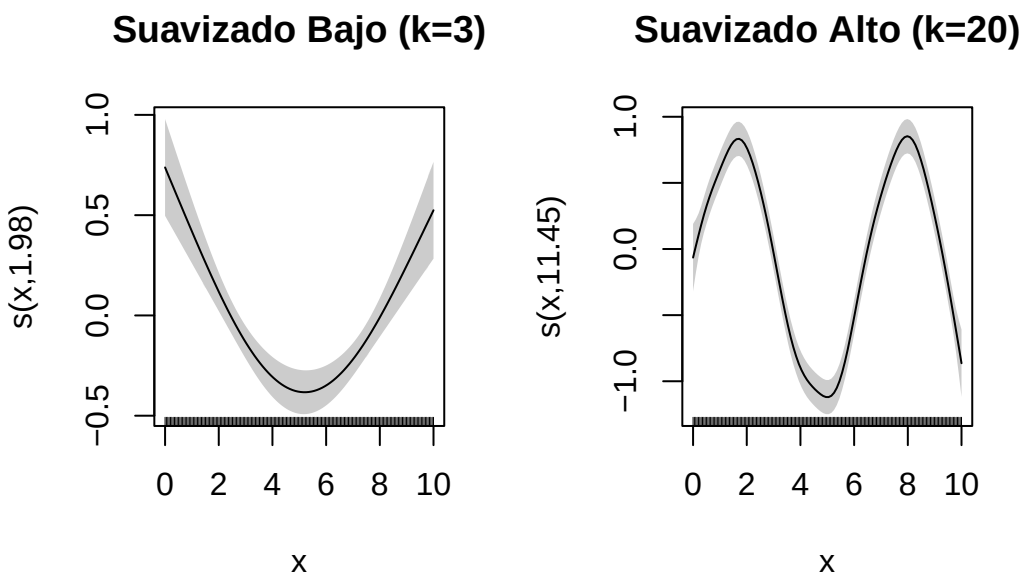
```
GCV (GCV.Cp): 0.08456186
```

Aunque `mgcv` selecciona automáticamente el grado de suavizado, es posible **controlar manualmente** la complejidad del modelo especificando el número de **bases de suavizado** mediante el argumento `k` en la función `s()`:

- **`k` pequeño:** Menor flexibilidad, puede llevar al subajuste.
- **`k` grande:** Mayor flexibilidad, riesgo de sobreajuste.

Ejemplo: Control manual del suavizado

```
# Menor suavizado (k = 3)
modelo_suave <- gam(y ~ s(x, k = 3), method = "REML")

# Mayor suavizado (k = 20)
modelo_mas_flexible <- gam(y ~ s(x, k = 20), method = "REML")

# Visualización de los diferentes ajustes
par(mfrow = c(1, 2))
plot(modelo_suave, main = "Suavizado Bajo (k=3)", shade = TRUE)
plot(modelo_mas_flexible, main = "Suavizado Alto (k=20)", shade = TRUE)
```



La **validación cruzada** es una técnica robusta para seleccionar el grado de suavizado. Tal y como hemos visto en temas anteriores, en este enfoque, el conjunto de datos se divide en varios subconjuntos (folds), y el modelo se entrena y evalúa en diferentes combinaciones de estos subconjuntos.

Ejemplo: Valización cruzada para GAMs

```
# Cargar librerías necesarias
library(cvTools)
```

```
Loading required package: lattice
```

```
Loading required package: robustbase
```

```r
library(mgcv)

# Configuración de la validación cruzada 5-fold
set.seed(123)
folds <- cvFolds(n = n, K = 5)

# Inicializar un vector para almacenar el error
errores <- numeric(5)

# Validación cruzada manual
for (i in 1:5) {
  # Dividir en conjunto de entrenamiento y prueba
  test_idx <- which(folds$which == i)
  train_idx <- setdiff(1:n, test_idx)

  # Ajustar el modelo en el conjunto de entrenamiento
  modelo_gam_cv <- gam(y ~ s(x), data = data.frame(x = x[train_idx], y = y[train_idx]), met

  # Predecir en el conjunto de prueba
  predicciones <- predict(modelo_gam_cv, newdata = data.frame(x = x[test_idx]))

  # Calcular el error cuadrático medio (RMSE)
  errores[i] <- sqrt(mean((y[test_idx] - predicciones)^2))
}

# Promedio del error de la validación cruzada
error_promedio <- mean(errores)
cat("Error Promedio (RMSE) de la Validación Cruzada:", error_promedio)
```

```
Error Promedio (RMSE) de la Validación Cruzada: 0.2946876
```

6.3.5 Diagnóstico de sobreajuste

Un modelo sobreajustado sigue de cerca las fluctuaciones del ruido en los datos, lo que puede detectarse mediante:

- **Residuos estructurados:** Los residuos no deberían mostrar patrones sistemáticos.
- **Curvas excesivamente flexibles:** Si la curva suavizada presenta oscilaciones innecesarias, es señal de sobreajuste.
- **Baja generalización:** Evaluar el rendimiento del modelo en datos de prueba puede revelar problemas de sobreajuste.

7 Conclusiones

A lo largo de cinco capítulos hemos presentado el material de la asignatura de **Modelos de Regresión** del grado en **Ciencia e Ingeniería de Datos** de la **Universidad Rey Juan Carlos**.

En este capítulo final, te invitamos a reflexionar sobre las principales lecciones aprendidas durante el curso **"Modelos de Regresión"** y a destacar la importancia de las técnicas abordadas en la formación de futuros profesionales en el campo de la Ciencia e Ingeniería de Datos. Además, animamos a los estudiantes a continuar explorando y ampliando sus conocimientos en cursos posteriores, consolidando así una base sólida para afrontar los desafíos de la ciencia de datos moderna.

Al finalizar este recorrido por los conceptos y técnicas de la regresión, queda claro que esta disciplina es fundamental en el análisis y la interpretación de datos en el mundo actual. Hemos explorado desde los **fundamentos teóricos** hasta las **aplicaciones prácticas**, proporcionando a los estudiantes las herramientas necesarias para construir sus habilidades en el análisis de datos y en la toma de decisiones basadas en evidencia.

7.1 Resumen de los aprendizajes

A lo largo del libro, hemos abordado diversos temas que forman la columna vertebral del análisis de regresión:

1. **Modelos de Regresión Lineal Simple y Múltiple:**
   Comprendimos cómo los modelos lineales permiten describir la relación entre una variable dependiente y una o más variables independientes. Exploramos técnicas para estimar parámetros, interpretar coeficientes y diagnosticar la validez del modelo.
2. **Métodos de selección de variables y Regularización:**
   Aprendimos a identificar las variables más relevantes mediante técnicas de selección como el **stepwise**, así como métodos de regularización como **Ridge**, **Lasso** y **Elastic Net**, que ayudan a mejorar la generalización del modelo.
3. **Modelos no lineales y transformación de variables:**
   Introdujimos métodos para capturar relaciones no lineales entre variables, incluyendo la **regresión polinomial**, los **splines** y la **ingeniería de características** para mejorar el rendimiento predictivo.

4. **Modelos de Regresión generalizada:**
   Ampliamos nuestro conocimiento hacia modelos que permiten trabajar con diferentes tipos de variables dependientes, como la **regresión logística** para variables binarias y la **regresión de Poisson** para datos de conteo.

5. **Otros modelos avanzados:**
   Finalmente, exploramos técnicas avanzadas como los **Modelos Aditivos Generalizados (GAMs)**, que proporcionan una manera flexible de capturar relaciones no lineales complejas sin perder interpretabilidad.

7.2 Reflexiones finales

El aprendizaje de los **Modelos de Regresión** va más allá de la simple aplicación de fórmulas o técnicas estadísticas. A través de este curso, hemos desarrollado un enfoque crítico para **analizar datos**, **identificar patrones** y **tomar decisiones basadas en evidencia**. Hemos comprendido la importancia de realizar un diagnóstico adecuado del modelo, asegurando que los supuestos estadísticos se cumplan y que los resultados sean fiables y reproducibles.

La **interpretabilidad** y la **validación del modelo** son pilares fundamentales en el análisis de regresión. Entender cómo y por qué un modelo llega a ciertas conclusiones es tan importante como la precisión de sus predicciones. En un mundo donde los datos juegan un papel cada vez más crucial, la habilidad para interpretar resultados y comunicar hallazgos de manera clara y concisa es esencial para cualquier profesional de la ciencia de datos.

7.3 Mirando hacia adelante

Con el conocimiento y las habilidades adquiridas en este curso, los estudiantes están mejor preparados para profundizar en el vasto campo de la ciencia de datos. Los modelos de regresión seguirán evolucionando con el avance de la tecnología y la creciente disponibilidad de datos. Por lo tanto, es crucial que los futuros profesionales mantengan una **mentalidad de aprendizaje continuo** y estén abiertos a adoptar nuevas metodologías y herramientas.

Los modelos de regresión son solo el comienzo de un camino más amplio en el análisis predictivo y el aprendizaje automático. En los próximos cursos, como **Aprendizaje Automático I** y **Aprendizaje Automático II**, aplicarás muchas de las técnicas y conceptos que hemos aprendido, avanzando hacia modelos más complejos como los **árboles de decisión**, las **redes neuronales** y los **modelos de ensamblaje**.

En conclusión, esperamos que este libro haya proporcionado una comprensión profunda y práctica de los **Modelos de Regresión**, y que inspire a los estudiantes a aplicar estos conocimientos con confianza y creatividad en sus proyectos futuros. La capacidad de **analizar datos de**

manera crítica y **tomar decisiones basadas en evidencias** es una habilidad poderosa y transformadora, que sin duda abrirá numerosas oportunidades en el ámbito profesional.

¡Buena suerte en tu camino en la ciencia de datos!

Bibliografía

Bates, Douglas, Martin Mächler, Ben Bolker, and Steve Walker. 2015. "Fitting Linear Mixed-Effects Models Using Lme4." *Journal of Statistical Software* 67 (1): 1–48.

Coxe, Stefany, Stephen G West, and Leona S Aiken. 2009. "The Analysis of Count Data: A Gentle Introduction to Poisson Regression and Its Alternatives." *Journal of Personality Assessment* 91 (2): 121–36.

Draper, NR. 1998. *Applied Regression Analysis*. McGraw-Hill. Inc.

Fox, John, and Sanford Weisberg. 2018. *An r Companion to Applied Regression*. Sage publications.

Galton, Francis. 1886. "Regression Towards Mediocrity in Hereditary Stature." *The Journal of the Anthropological Institute of Great Britain and Ireland* 15: 246–63.

Harrell, Frank E., Jr. 2015. *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. Second. Springer.

Hastie, Trevor J. 2017. "Generalized Additive Models." In *Statistical Models in s*, 249–307. Routledge.

Hosmer Jr, David W, Stanley Lemeshow, and Rodney X Sturdivant. 2013. *Applied Logistic Regression*. John Wiley & Sons.

James, Gareth, Daniela Witten, Trevor Hastie, Robert Tibshirani, et al. 2013. *An Introduction to Statistical Learning*. Vol. 112. Springer.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2021. *An Introduction to Statistical Learning with Applications in r*. Second. Springer.

Kutner, Michael H, Christopher J Nachtsheim, John Neter, and William Li. 2005. *Applied Linear Statistical Models*. McGraw-hill.

Lambert, Diane. 1992. "Zero-Inflated Poisson Regression, with an Application to Defects in Manufacturing." *Technometrics* 34 (1): 1–14.

Marquardt, Donald W, and Ronald D Snee. 1975. "Ridge Regression in Practice." *The American Statistician* 29 (1): 3–20.

Nelder, John Ashworth, and Robert WM Wedderburn. 1972. "Generalized Linear Models." *Journal of the Royal Statistical Society Series A: Statistics in Society* 135 (3): 370–84.

Pinheiro, José C., and Douglas M. Bates. 2000. *Mixed-Effects Models in s and s-PLUS*. New York: Springer.

Ranstam, Jonas, and Jonathan A Cook. 2018. "LASSO Regression." *Journal of British Surgery* 105 (10): 1348–48.

Shmueli, Galit. 2010. "To Explain or to Predict?" *Statistical Science* 25 (3): 289–310.

Weisberg, S. 2005. "Applied Linear Regression." Wiley.

Wood, Simon N. 2017. *Generalized Additive Models: An Introduction with r*. Second. Chapman; Hall/CRC.

Zou, Hui, and Trevor Hastie. 2005. "Regularization and Variable Selection via the Elastic Net." *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67 (2): 301–20.