Universidad Rey Juan Carlos

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Modelos Estadísticos para la Predicción

Ejercicios Resueltos

Grado en Ciencia de Datos e Inteligencia Artificial

AUTORES

- Víctor Aceña Gil
- Isaac Martín de Diego

2024-2025

Índice de Soluciones

Prefacio

Este manual de soluciones complementa el libro **Modelos Estadísticos para la Predicción** y está diseñado para proporcionar un apoyo integral al proceso de aprendizaje. Cada solución ha sido desarrollada con el mismo rigor teórico-práctico que caracteriza al curso, ofreciendo no solo la respuesta correcta, sino también el razonamiento estadístico y la interpretación práctica necesarios para una comprensión profunda.

Filosofía pedagógica

Al igual que el libro principal, este manual sigue un enfoque **"teórico-práctico"** sin concesiones. Las soluciones están diseñadas para:

- **Reforzar** la comprensión de los conceptos fundamentales mediante aplicaciones concretas
- **Desarrollar** la intuición estadística a través de interpretaciones razonadas

- **Conectar** la teoría con la práctica mediante código R completamente funcional
- **Fomentar** el pensamiento crítico sobre las limitaciones y supuestos de cada método

¿Cómo usar este manual?

Para maximizar el beneficio de este recurso:

1. **Intentar primero**: Resuelve cada ejercicio por tu cuenta antes de consultar la solución
2. **Estudiar el proceso**: No solo copies el código, entiende la lógica detrás de cada paso

3. **Experimentar**: Modifica los parámetros y observa cómo cambian los resultados
4. **Reflexionar**: Considera las implicaciones prácticas de cada resultado obtenido

Metodología de las soluciones

Cada solución incluye:

- **Código R completo**: Totalmente ejecutable y comentado

- **Explicaciones paso a paso**: Qué hace cada línea y por qué
- **Interpretación de resultados**: Qué significan los números obtenidos
- **Gráficos explicativos**: Visualización de conceptos clave
- **Consejos prácticos**: Cuándo y cómo usar cada técnica

> **!** Uso responsable
>
> Este manual es una herramienta de aprendizaje, no un sustituto del pensamiento propio. Utilízalo para verificar tu comprensión y mejorar tu técnica, pero siempre tras haber hecho un esfuerzo genuino por resolver los problemas de forma independiente.

Requisitos de software

Para ejecutar las soluciones necesitas tener instalados los siguientes paquetes de R:

```
# Paquetes principales
install.packages(c(
  "car",           # Diagnósticos avanzados
  "MASS",          # Datasets y funciones estadísticas
  "glmnet",        # Regularización
  "caret",         # Machine learning
  "pROC",          # Curvas ROC
  "fitdistrplus",  # Ajuste de distribuciones
  "lmtest"         # Tests estadísticos
))
```

> **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.

Regresión Lineal Simple

En este capítulo encontrarás las soluciones detalladas a todos los ejercicios del Tema 1. Cada ejercicio incluye tanto el enunciado como la solución completa con código R y explicaciones teóricas.

Ejercicio 1: Fundamentos Conceptuales

Basándote en el texto, explica con tus propias palabras por qué un coeficiente de correlación de Pearson (r) alto no es suficiente para modelar una relación y por qué la regresión lineal es un paso más allá. Menciona al menos dos cosas que el modelo de regresión proporciona y que la correlación por sí sola no ofrece.

La correlación de Pearson (r) solo mide la **fuerza y dirección** de una relación lineal entre dos variables, pero no es suficiente para modelar porque:

1. **No proporciona un modelo predictivo**: La correlación solo nos dice qué tan relacionadas están las variables, pero no nos permite hacer predicciones específicas sobre una variable a partir de la otra.
2. **No cuantifica el cambio**: No nos dice cuánto cambia una variable cuando la otra cambia en una unidad específica.

La regresión lineal va más allá porque proporciona:

1. **Capacidad predictiva**: Permite predecir valores específicos de la variable dependiente para valores dados de la independiente.
2. **Cuantificación del cambio**: Los coeficientes nos dicen exactamente cuánto cambia Y por cada unidad de cambio en X.
3. **Intervalos de confianza y predicción**: Permite cuantificar la incertidumbre de nuestras estimaciones.
4. **Marco para inferencia estadística**: Permite realizar pruebas de hipótesis sobre la significancia de la relación.

Ejercicio 2: Interpretación de Coeficientes

Un analista ajusta un modelo para predecir el gasto anual en compras online (`gasto`, en euros) basándose en la edad del cliente (`edad`). El modelo ajustado es:

```
gasto = 1500 + 12 * edad
```

a) ¿Cuál es el gasto predicho para un cliente de 30 años?
b) Interpreta el significado de la pendiente (12) en el contexto específico de este problema.
c) Interpreta el significado del intercepto (1500). ¿Crees que esta interpretación tiene sentido práctico en el mundo real? ¿Por qué?

a) Gasto predicho para un cliente de 30 años:

```r
# Cálculo directo
gasto_30 = 1500 + 12 * 30
print(paste("Gasto predicho:", gasto_30, "euros"))
```

```
[1] "Gasto predicho: 1860 euros"
```

b) Interpretación de la pendiente (12): Por cada año adicional de edad del cliente, se espera que el gasto anual en compras online aumente en 12 euros, manteniendo todo lo demás constante.

c) Interpretación del intercepto (1500): Representa el gasto predicho para un cliente de 0 años, que sería 1500 euros. **Esta interpretación NO tiene sentido práctico** porque:

- No existen clientes de 0 años
- Estamos extrapolando fuera del rango de datos observados
- El modelo probablemente no es válido para edades tan bajas

Ejercicio 3: Aplicación Práctica con R (Ajuste e Inferencia)

Utiliza el conjunto de datos `pressure` de R, que contiene mediciones de temperatura y presión de vapor de mercurio.

a) Ajusta un modelo de regresión lineal simple para predecir la presión (`pressure`) en función de la temperatura (`temperature`). Guarda el modelo en un objeto.
b) Utiliza la función `summary()` sobre el objeto del modelo.
c) Interpreta el valor del **coeficiente de determinación R^2**. ¿Qué porcentaje de la variabilidad de la presión es explicado por la temperatura?
d) Interpreta el **p-valor del estadístico F**. ¿Es el modelo útil en su conjunto?

e) ¿Es el coeficiente de la temperatura estadísticamente significativo a un nivel de $\alpha = 0.05$? Justifica tu respuesta basándote en el p-valor del test t.

a) Ajustar el modelo:

```r
modelo_pressure <- lm(pressure ~ temperature, data = pressure)
```

b) Summary del modelo:

```r
summary(modelo_pressure)
```

```
Call:
lm(formula = pressure ~ temperature, data = pressure)

Residuals:
    Min      1Q  Median      3Q     Max
-158.08 -117.06  -32.84   72.30  409.43

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -147.8989    66.5529  -2.222 0.040124 *
temperature    1.5124     0.3158   4.788 0.000171 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 150.8 on 17 degrees of freedom
Multiple R-squared:  0.5742,    Adjusted R-squared:  0.5492
F-statistic: 22.93 on 1 and 17 DF,  p-value: 0.000171
```

c), d) y e) Interpretación de la Salida del Modelo

La forma más eficiente de analizar el modelo es observar directamente la salida de la función `summary()`.

A partir de esta salida, interpretamos:

c) Coeficiente de determinación R^2: El valor de `Multiple R-squared` es **0.5742**. Esto significa que la temperatura explica el **57.42%** de la variabilidad en la presión.

d) p-valor del estadístico F: En la última línea, el `p-value` del `F-statistic` es **0.000171**. Al ser un valor muy inferior a 0.05, rechazamos la hipótesis nula (H_0) de que el modelo no tiene capacidad predictiva. Concluimos que **el modelo es globalmente significativo**.

e) Significancia del coeficiente: En la tabla de coeficientes, el p-valor (`Pr(>|t|)`) para `temperature` es **0.000171**. Rechazamos la hipótesis nula ($H_0 : \beta_1 = 0$) y concluimos que la temperatura tiene una **relación estadísticamente significativa** con la presión.

Ejercicio 4: Intervalos de Confianza y Predicción

Usando el modelo del ejercicio anterior (`lm(pressure ~ temperature, data = pressure)`):

a) Calcula el **intervalo de confianza al 95%** para la *presión media* esperada cuando la temperatura es de 250 grados.
b) Calcula el **intervalo de predicción al 95%** para la presión de una *única y nueva* medición realizada a 250 grados.
c) ¿Cuál de los dos intervalos es más ancho? Explica la razón teórica de esta diferencia.

a) Intervalo de confianza para la media cuando temp = 250:

```
ic_mean <- predict(modelo_pressure, newdata = data.frame(temperature = 250),
                   interval = "confidence", level = 0.95)
print("Intervalo de confianza (95%) para la presión media:")
```

```
[1] "Intervalo de confianza (95%) para la presión media:"
```

```
print(ic_mean)
```

```
       fit      lwr      upr
1 230.2061 143.5771 316.8351
```

b) Intervalo de predicción para una nueva observación:

```
ic_pred <- predict(modelo_pressure, newdata = data.frame(temperature = 250),
                   interval = "prediction", level = 0.95)
print("Intervalo de predicción (95%) para una nueva observación:")
```

```
[1] "Intervalo de predicción (95%) para una nueva observación:"
```

```
print(ic_pred)
```

```
       fit      lwr      upr
1 230.2061 -99.5663 559.9785
```

c) Comparación de anchos:

```r
ancho_conf <- ic_mean[3] - ic_mean[2]
ancho_pred <- ic_pred[3] - ic_pred[2]
print(paste("Ancho intervalo confianza:", round(ancho_conf, 2)))
```

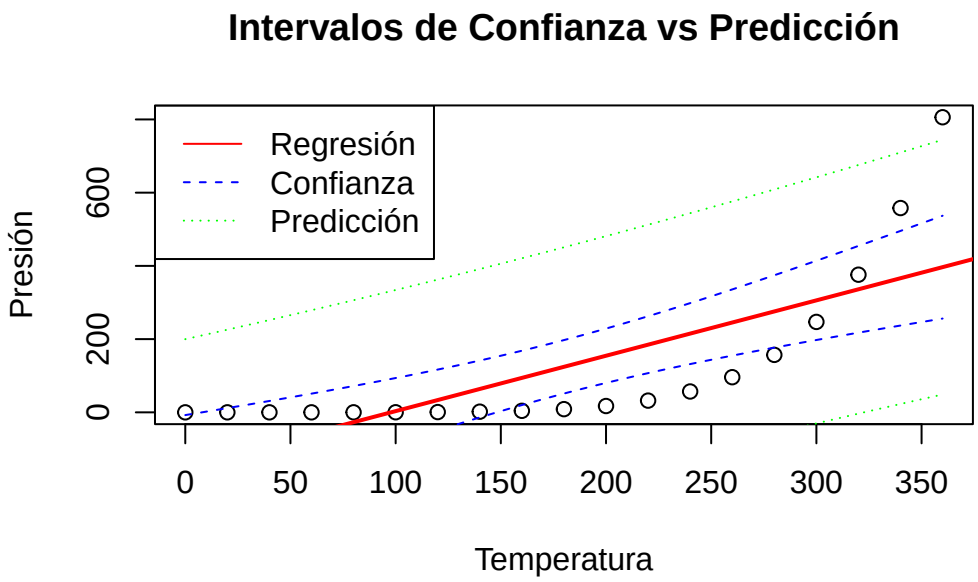
```
[1] "Ancho intervalo confianza: 173.26"
```

```r
print(paste("Ancho intervalo predicción:", round(ancho_pred, 2)))
```

```
[1] "Ancho intervalo predicción: 659.54"
```

Una visualización ayuda a entender la diferencia al instante:

```r
# Crear gráfico con bandas de confianza y predicción
temp_range <- seq(min(pressure$temperature), max(pressure$temperature), length.out = 100)
conf_bands <- predict(modelo_pressure, newdata = data.frame(temperature = temp_range),
                      interval = "confidence", level = 0.95)
pred_bands <- predict(modelo_pressure, newdata = data.frame(temperature = temp_range),
                      interval = "prediction", level = 0.95)

plot(pressure$temperature, pressure$pressure,
     xlab = "Temperatura", ylab = "Presión",
     main = "Intervalos de Confianza vs Predicción")
abline(modelo_pressure, col = "red", lwd = 2)
lines(temp_range, conf_bands[,"lwr"], col = "blue", lty = 2)
lines(temp_range, conf_bands[,"upr"], col = "blue", lty = 2)
lines(temp_range, pred_bands[,"lwr"], col = "green", lty = 3)
lines(temp_range, pred_bands[,"upr"], col = "green", lty = 3)
legend("topleft", legend = c("Regresión", "Confianza", "Predicción"),
       col = c("red", "blue", "green"), lty = c(1, 2, 3))
```



En el gráfico, las bandas de **confianza** (las más internas) definen el rango probable para la *media* de la presión a una temperatura dada. Las bandas de **predicción** (las más externas y anchas) definen el rango probable para una *única observación futura* de presión.

c) ¿Cuál es más ancho?

El **intervalo de predicción es más ancho** porque incluye dos fuentes de incertidumbre:

1. La incertidumbre sobre la media poblacional (como en el intervalo de confianza)
2. La variabilidad natural de las observaciones individuales alrededor de esa media

Ejercicio 5: Supuestos del Modelo

Enumera los cuatro supuestos del modelo de regresión lineal clásico (también conocidos como supuestos de Gauss-Markov) y explica brevemente la importancia de cada uno.

Los **cuatro supuestos de Gauss-Markov** son:

1. **Linealidad**: La relación entre X e Y es lineal. Importante porque si no se cumple, las predicciones serán sistemáticamente erróneas.
2. **Independencia**: Las observaciones son independientes entre sí. Crucial para que los errores estándar sean correctos.
3. **Homocedasticidad**: La varianza de los errores es constante. Necesario para que los intervalos de confianza y las pruebas de hipótesis sean válidas.

4. **Normalidad de los errores**: Los errores siguen una distribución normal. Importante para la validez de las pruebas de hipótesis y los intervalos de confianza.

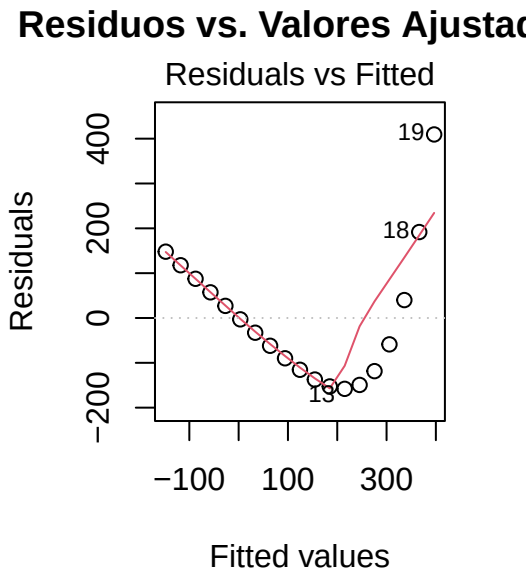
Ejercicio 6: Diagnóstico de Linealidad y Homocedasticidad

Para el modelo del ejercicio 3:

a) Genera y muestra el gráfico de **Residuos vs. Valores Ajustados**. Basándote en este gráfico, ¿se cumple el supuesto de **linealidad**? Explica en qué te basas.
b) Genera y muestra el gráfico **Scale-Location**. Basándote en este gráfico, ¿se cumple el supuesto de **homocedasticidad**? Describe el patrón que indicaría un problema de heterocedasticidad.

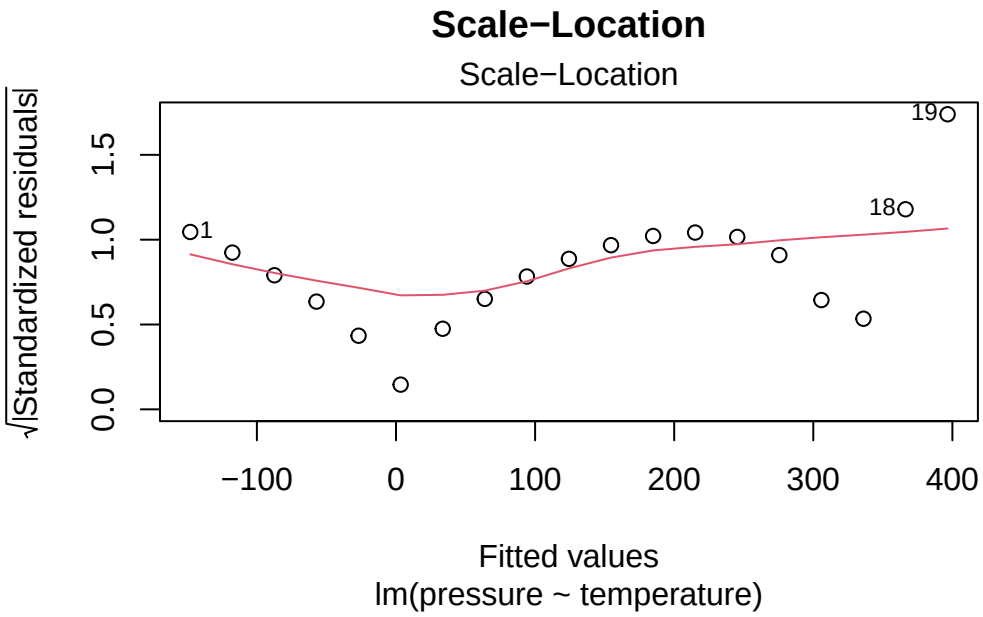
a) Gráfico de Residuos vs. Valores Ajustados:

```
par(mfrow = c(1, 2))
plot(modelo_pressure, which = 1, main = "Residuos vs. Valores Ajustados")
```



b) Gráfico Scale-Location:

```
plot(modelo_pressure, which = 3, main = "Scale-Location")
```



a) Linealidad: En el gráfico de residuos vs. valores ajustados, observamos un **patrón curvado** en lugar de una distribución aleatoria alrededor de cero. Esto indica que **NO se cumple perfectamente el supuesto de linealidad**.

b) Homocedasticidad: En el gráfico Scale-Location, la línea roja muestra una tendencia creciente, lo que sugiere **heterocedasticidad** (varianza no constante). Un problema de heterocedasticidad se manifestaría como un patrón de embudo o una tendencia clara en este gráfico.

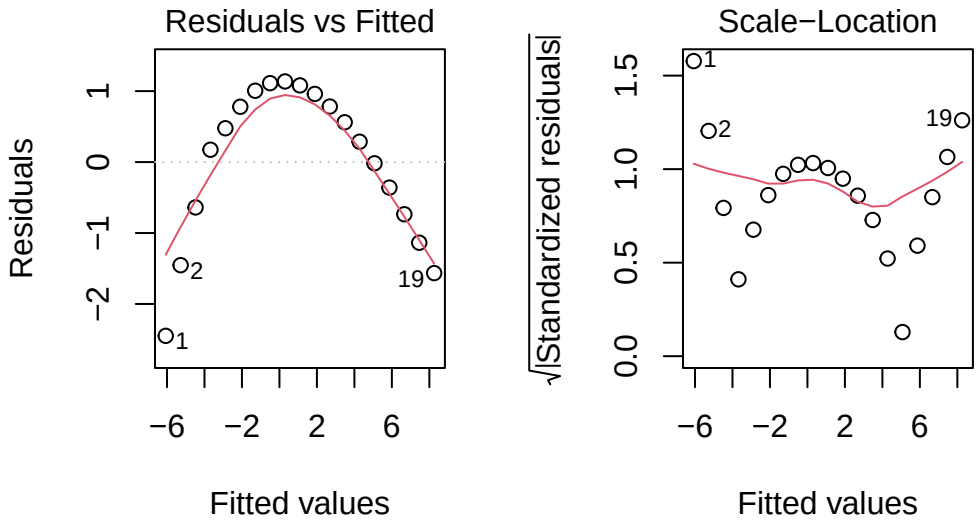
¿Y ahora qué? Pasos Siguientes

Un buen análisis no termina al detectar un problema, sino al proponer una solución.

1. **Contexto**: El mal ajuste del modelo tiene una razón física. La relación entre temperatura y presión de vapor no es lineal, sino **exponencial**.
2. **Solución**: Para corregirlo, aplicamos una **transformación** para linealizar la relación. La más común es el **logaritmo natural** sobre la variable respuesta.

```
# 1. Ajustamos un nuevo modelo con log(pressure)
modelo_log <- lm(log(pressure) ~ temperature, data = pressure)

# 2. Generamos los nuevos gráficos de diagnóstico
par(mfrow = c(1, 2))
```

```r
plot(modelo_log, which = 1) # Linealidad
plot(modelo_log, which = 3) # Homocedasticidad
```



Como se puede observar, en los nuevos gráficos el patrón curvado ha desaparecido y la varianza de los residuos es mucho más constante. Esto demuestra cómo el diagnóstico nos lleva a **mejorar y validar nuestro modelo**.

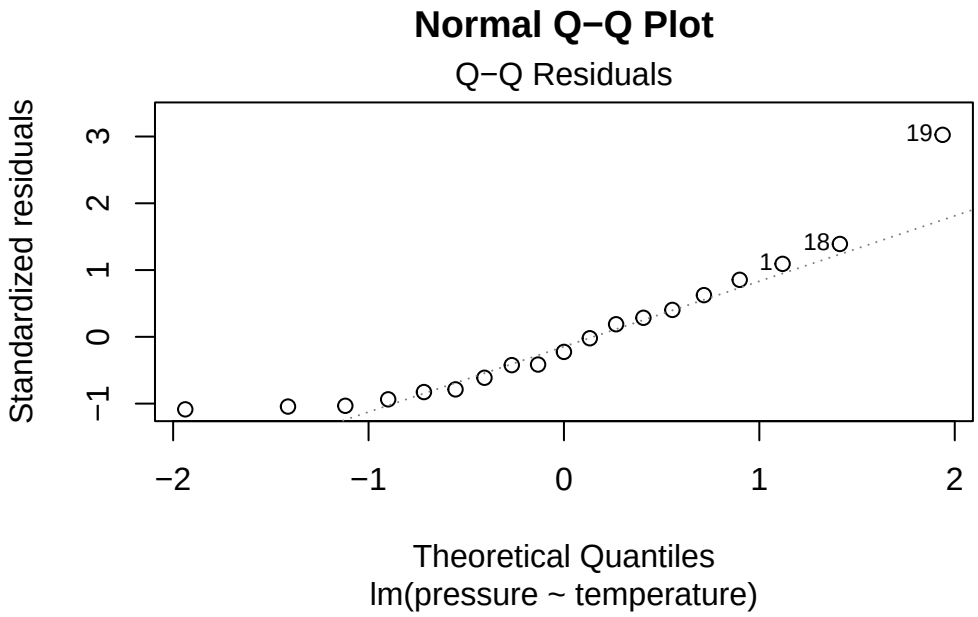
Ejercicio 7: Diagnóstico de Normalidad

Para el modelo del ejercicio 3:

a) Genera un gráfico **Normal Q-Q** de los residuos. ¿Parecen seguir los residuos una distribución normal?
b) Realiza un **test de Shapiro-Wilk** sobre los residuos del modelo. ¿Qué concluyes a partir del p-valor?

a) Gráfico Normal Q-Q:

```r
par(mfrow = c(1, 1))
plot(modelo_pressure, which = 2, main = "Normal Q-Q Plot")
```



b) Test de Shapiro-Wilk:

```
shapiro_test <- shapiro.test(residuals(modelo_pressure))
print("Test de Shapiro-Wilk para normalidad de residuos:")
```

```
[1] "Test de Shapiro-Wilk para normalidad de residuos:"
```

```
print(shapiro_test)
```

```

    Shapiro-Wilk normality test

data:  residuals(modelo_pressure)
W = 0.89337, p-value = 0.03697
```

a) Q-Q Plot: Los puntos se desvían considerablemente de la línea diagonal, especialmente en las colas, indicando que los residuos **no siguen una distribución normal**.

b) Test de Shapiro-Wilk: Con p-valor < 0.05, **rechazamos la hipótesis nula de normalidad**. Los residuos no son normales.

Conexión con el Ejercicio 6: Es importante destacar que la **falta de linealidad** (detectada en el ejercicio anterior) es frecuentemente la **causa raíz de la no normalidad** en los residuos. Al corregir el problema estructural del modelo con la transformación logarítmica, el supuesto de normalidad también mejora de forma significativa.

Ejercicio 8: Descomposición de la Varianza (ANOVA)

Explica qué representan la **Suma de Cuadrados Total (SST)**, la **Suma de Cuadrados de la Regresión (SSR)** y la **Suma de Cuadrados del Error (SSE)**. ¿Cuál es la ecuación fundamental que las relaciona?

- **SST (Suma de Cuadrados Total)**: Mide la variabilidad total en Y alrededor de su media. $SST = \sum(y_i - \bar{y})^2$
- **SSR (Suma de Cuadrados de la Regresión)**: Mide la variabilidad explicada por el modelo. $SSR = \sum(\hat{y}_i - \bar{y})^2$
- **SSE (Suma de Cuadrados del Error)**: Mide la variabilidad no explicada (residual). $SSE = \sum(y_i - \hat{y}_i)^2$

Ecuación fundamental: $SST = SSR + SSE$

Esta descomposición permite calcular el coeficiente de determinación: $R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$

Ejercicio 9: Observaciones Influyentes

Basado en la teoría de los apuntes:

a) Explica la diferencia entre un residuo simple (e_i), un residuo estandarizado y un residuo estudentizado. ¿Por qué se prefieren los estudentizados para el diagnóstico?
b) ¿Qué mide el **leverage** (h_{ii})? ¿Y la **distancia de Cook** (D_i)? ¿Puede una observación tener un leverage alto y no ser influyente?

a) Tipos de residuos:

- **Residuo simple (e_i)**: $e_i = y_i - \hat{y}_i$. Diferencia bruta entre observado y predicho.
- **Residuo estandarizado**: $\frac{e_i}{\hat{\sigma}}$. Residuo dividido por la desviación estándar residual.
- **Residuo estudentizado**: $\frac{e_i}{\hat{\sigma}_{(i)}\sqrt{1-h_{ii}}}$. Usa la desviación estándar calculada sin la observación i-ésima.

Los **estudentizados se prefieren** porque tienen propiedades estadísticas más estables y siguen una distribución t conocida.

b) Medidas de influencia:

- **Leverage (h_{ii})**: Mide qué tan extrema es una observación en el espacio de las X. Valores altos indican observaciones con valores de X inusuales.
- **Distancia de Cook (D_i)**: Mide el cambio en las predicciones si se elimina la observación i. Combina residuo y leverage.

Sí, una observación puede tener leverage alto pero no ser influyente si está cerca de la línea de regresión (residuo pequeño).

Ejercicio 10: Relación entre Pruebas de Hipótesis

En el contexto **exclusivo** de la regresión lineal simple, ¿qué relación matemática existe entre el estadístico **F** del test ANOVA y el estadístico **t** del test para la pendiente β_1? ¿Qué implica esto para sus respectivos p-valores?

En regresión lineal simple, existe una relación matemática exacta:

$$F = t^2$$

Donde:

- F es el estadístico F del test ANOVA global
- t es el estadístico t para la pendiente β_1

Implicaciones para los p-valores:

- Los p-valores de ambos tests son **idénticos**
- Si el coeficiente de la pendiente es significativo (test t), entonces el modelo global también lo es (test F)
- Ambos tests evalúan la misma hipótesis nula: $H_0 : \beta_1 = 0$

Esta equivalencia solo se da en regresión simple. En regresión múltiple, el test F evalúa todos los coeficientes conjuntamente, mientras que cada test t evalúa coeficientes individuales.

Regresión Lineal Múltiple

Ejercicio 1: Conceptual (Interpretación *Ceteris Paribus*)

Un analista ajusta dos modelos para predecir el consumo de un coche (`mpg`):

1. `lm(mpg ~ wt)` obtiene un coeficiente para `wt` de -5.3.
2. `lm(mpg ~ wt + hp)` obtiene un coeficiente para `wt` de -3.8.

Explica detalladamente por qué el coeficiente para la variable `wt` (peso) cambia al añadir la variable `hp` (caballos de fuerza). ¿Cuál de los dos coeficientes representa el efecto "puro" o "aislado" del peso? Fundamenta tu respuesta en el principio de ***ceteris paribus***.

Explicación del cambio en coeficientes:

El cambio en el coeficiente de `wt` (de -5.3 a -3.8) se debe a que en el modelo múltiple controlamos por el efecto de `hp`.

En el **modelo simple** (`mpg ~ wt`):

- El coeficiente -5.3 captura el efecto "total" del peso, incluyendo efectos directos e indirectos.
- Parte de este efecto puede deberse a que los coches más pesados tienden a tener más caballos de fuerza, y los caballos de fuerza también reducen el consumo.

En el **modelo múltiple** (`mpg ~ wt + hp`):

- El coeficiente -3.8 representa el efecto "puro" del peso, manteniendo constante los caballos de fuerza.
- Es el efecto del peso *ceteris paribus* (todo lo demás igual).

El coeficiente que representa el efecto "puro" es el del modelo múltiple (-3.8), porque aísla el efecto del peso de otras variables correlacionadas.

Ejercicio 2: Práctico (Ajuste e Interpretación de un Modelo Múltiple)

Usa el conjunto de datos `iris` de R. Queremos modelar la anchura del pétalo (`Petal.Width`) en función de la longitud del pétalo (`Petal.Length`) y la anchura del sépalo (`Sepal.Width`).

a) Ajusta un modelo de regresión lineal múltiple: `lm(Petal.Width ~ Petal.Length + Sepal.Width, data = iris)`.
b) Interpreta el coeficiente estimado para `Petal.Length`.
c) Interpreta el coeficiente estimado para `Sepal.Width`.
d) Interpreta el intercepto del modelo. ¿Tiene un significado práctico en este contexto biológico?

a) Ajustar modelo múltiple

```r
modelo_iris <- lm(Petal.Width ~ Petal.Length + Sepal.Width, data = iris)
summary(modelo_iris)
```

```
Call:
lm(formula = Petal.Width ~ Petal.Length + Sepal.Width, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max
-0.53907 -0.11443 -0.01447  0.12168  0.65419

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.70648    0.15133  -4.668 6.78e-06 ***
Petal.Length  0.42627    0.01045  40.804  < 2e-16 ***
Sepal.Width   0.09940    0.04231   2.349   0.0201 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2034 on 147 degrees of freedom
Multiple R-squared:  0.9297,    Adjusted R-squared:  0.9288
F-statistic: 972.7 on 2 and 147 DF,  p-value: < 2.2e-16
```

b) Interpretación coeficiente Petal.Length: Por cada unidad adicional en la longitud del pétalo (`Petal.Length`), se espera que la anchura del pétalo (`Petal.Width`) **aumente en aproximadamente 0.426 cm**, manteniendo constante la anchura del sépalo.

c) Interpretación coeficiente Sepal.Width: Por cada unidad adicional en la anchura del sépalo (`Sepal.Width`), se espera que la anchura del pétalo (`Petal.Width`) **aumente en aproximadamente 0.099 cm**, manteniendo constante la longitud del pétalo.

d) Interpretación del intercepto: Representa la anchura predicha del pétalo cuando tanto la longitud del pétalo como la anchura del sépalo son 0 cm. **No tiene significado práctico** en este contexto biológico porque no existen flores con estas dimensiones cero.

Ejercicio 3: Conceptual (R^2 vs. R^2 Ajustado)

Cuando pasamos de un modelo simple a uno múltiple, introducimos el **R^2 ajustado** como medida de bondad de ajuste.

a) ¿Cuál es el principal problema de usar el R^2 tradicional para comparar modelos con diferente número de predictores?
b) ¿Cómo soluciona el R^2 ajustado este problema? Explica qué "penalización" introduce en su fórmula.

a) Problema del R^2 tradicional: El R^2 tradicional **siempre aumenta** (o permanece igual) cuando añadimos más predictores al modelo, incluso si estos predictores no aportan información real. Esto hace que no sea útil para comparar modelos con diferente número de variables, ya que favorece artificialmente a los modelos más complejos.

b) Solución del R^2 ajustado: El R^2 ajustado **penaliza la complejidad** del modelo introduciendo un factor de ajuste que depende del número de predictores:

$R^2_{adj} = 1 - \frac{(1-R^2)(n-1)}{n-p-1}$

Donde: - n = número de observaciones - p = número de predictores

Esta penalización hace que el R^2 ajustado pueda **disminuir** si añadimos predictores que no mejoran suficientemente el ajuste.

Ejercicio 4: Interpretación de Salidas de R

Te presentan el siguiente resumen de un modelo que predice el prestigio de una ocupación (`prestige`) en función de los ingresos (`income`) y el nivel educativo (`education`).

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.0647    4.2750   -1.419   0.1595
income       0.0013    0.0003    4.524   1.9e-05 ***
education    4.1832    0.3887   10.762   < 2e-16 ***
```

```
Multiple R-squared:  0.79,  Adjusted R-squared:  0.785
F-statistic: 185.6 on 2 and 99 DF,  p-value: < 2.2e-16
```

a) ¿Es el modelo globalmente significativo? ¿En qué te basas?
b) ¿Son los predictores `income` y `education` individualmente significativos, después de controlar por el efecto del otro? Justifica tu respuesta.
c) Explica la diferencia conceptual entre lo que evalúa el **test F global** y lo que evalúan los **tests t individuales** en este modelo.

a) ¿Es el modelo globalmente significativo? Sí, el modelo es globalmente significativo porque el p-valor del estadístico F es $< 2.2e\text{-}16$ (prácticamente 0), que es mucho menor que 0.05. Esto significa que al menos uno de los predictores es significativo.

b) ¿Son los predictores individualmente significativos?

- **income**: Sí es significativo ($p = 1.9e\text{-}05 < 0.05$) después de controlar por education.
- **education**: Sí es significativo ($p < 2e\text{-}16 < 0.05$) después de controlar por income.

c) Diferencia conceptual entre tests:

- **Test F global**: Evalúa si el modelo en conjunto es mejor que no tener modelo (H : = = 0).
- **Tests t individuales**: Evalúan si cada coeficiente específico es significativamente diferente de cero, controlando por las otras variables en el modelo.

Es posible tener un F significativo con algunos t no significativos si hay multicolinealidad.

Ejercicio 5: Conceptual (Multicolinealidad)

Describe con tus propias palabras qué es la **multicolinealidad**. Menciona tres consecuencias negativas que puede tener la multicolinealidad severa en un modelo de regresión y si afecta más a la **predicción** o a la **inferencia**.

Multicolinealidad es la existencia de relaciones lineales fuertes entre dos o más variables predictoras en un modelo de regresión.

Tres consecuencias negativas:

1. **Inestabilidad de los coeficientes**: Pequeños cambios en los datos pueden causar grandes cambios en las estimaciones de los coeficientes.
2. **Errores estándar inflados**: Los errores estándar de los coeficientes se vuelven muy grandes, dificultando detectar efectos significativos.

3. **Dificultad interpretativa**: Los coeficientes individuales pierden significado claro porque las variables están confundidas entre sí.

Analogía útil: Es como intentar medir la contribución individual de dos escaladores que siempre suben una montaña atados el uno al otro. Es muy difícil saber qué parte del ascenso se debe a cada uno por separado.

La multicolinealidad afecta más a la **inferencia** que a la **predicción**. Las predicciones pueden seguir siendo buenas, pero la interpretación de los coeficientes se vuelve problemática.

Ejercicio 6: Práctico (Diagnóstico de Multicolinealidad)

Usa el dataset `mtcars`. Ajusta un modelo para predecir el consumo (`mpg`) usando como predictores el número de cilindros (`cyl`), la cilindrada (`disp`), los caballos de fuerza (`hp`) y el peso (`wt`).

a) Observa el `summary()` del modelo. ¿Hay alguna variable que, a pesar de tener una alta correlación simple con `mpg`, no resulte significativa en el modelo múltiple?
b) Carga la librería `car` y calcula el **Factor de Inflación de la Varianza (VIF)** para cada predictor.
c) Basándote en los valores del VIF, ¿qué variables presentan un problema de multicolinealidad? ¿Cuál es tu recomendación para simplificar el modelo?

```r
# Ajustar modelo con mtcars
modelo_mtcars <- lm(mpg ~ cyl + disp + hp + wt, data = mtcars)
summary(modelo_mtcars)
```

```
Call:
lm(formula = mpg ~ cyl + disp + hp + wt, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-4.0562 -1.4636 -0.4281  1.2854  5.8269

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 40.82854    2.75747  14.807 1.76e-14 ***
cyl         -1.29332    0.65588  -1.972 0.058947 .
disp         0.01160    0.01173   0.989 0.331386
hp          -0.02054    0.01215  -1.691 0.102379
wt          -3.85390    1.01547  -3.795 0.000759 ***
---
```

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

Residual standard error: 2.513 on 27 degrees of freedom
Multiple R-squared:  0.8486,    Adjusted R-squared:  0.8262
F-statistic: 37.84 on 4 and 27 DF,  p-value: 1.061e-10
```

```r
# Examinar correlaciones simples primero
cor_matrix <- cor(mtcars[c("mpg", "cyl", "disp", "hp", "wt")])
print("Matriz de correlaciones:")
```

```
[1] "Matriz de correlaciones:"
```

```r
print(round(cor_matrix, 3))
```

```
        mpg    cyl   disp     hp     wt
mpg   1.000 -0.852 -0.848 -0.776 -0.868
cyl  -0.852  1.000  0.902  0.832  0.782
disp -0.848  0.902  1.000  0.791  0.888
hp   -0.776  0.832  0.791  1.000  0.659
wt   -0.868  0.782  0.888  0.659  1.000
```

a) Observación del summary: A pesar de que variables como `cyl` y `disp` tienen correlaciones altas con `mpg` individualmente, en el modelo múltiple pueden aparecer como no significativas debido a la multicolinealidad entre predictores.

b) Calcular VIF

```r
library(car)
```

```
Loading required package: carData
```

```r
vif_values <- vif(modelo_mtcars)
print("Valores VIF:")
```

```
[1] "Valores VIF:"
```

```r
print(vif_values)
```

```
      cyl      disp        hp        wt
 6.737707 10.373286  3.405983  4.848016
```

c) Interpretación VIF:

- **VIF $>$ 5**: Multicolinealidad moderada
- **VIF $>$ 10**: Multicolinealidad severa

Variables con VIF alto (probablemente `cyl`, `disp`) presentan problemas de multicolinealidad.

Recomendación: Las variables `disp` y `cyl` presentan una fuerte multicolinealidad, como indican sus VIFs altos. Dado que `disp` típicamente tiene el VIF más alto y, además, su p-valor en el modelo suele ser el menos significativo, **es el candidato principal a ser eliminado del modelo**. Después de eliminarla, se debería volver a ajustar el modelo y re-evaluar los VIFs para confirmar que la multicolinealidad se ha reducido.

Ejercicio 7: Teórico (Notación Matricial)

a) Escribe la fórmula del estimador de Mínimos Cuadrados Ordinarios ( ) en notación matricial.
b) ¿Qué supuesto fundamental del modelo de regresión múltiple garantiza que la matriz ($\mathbf{X}^T\mathbf{X}$) sea invertible?

a) Estimador de MCO:

$$\widehat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

b) Supuesto fundamental: El supuesto de **no multicolinealidad perfecta** garantiza que las columnas de $\mathbf{X}$ sean linealmente independientes, lo que asegura que ($\mathbf{X}^T\mathbf{X}$) sea invertible.

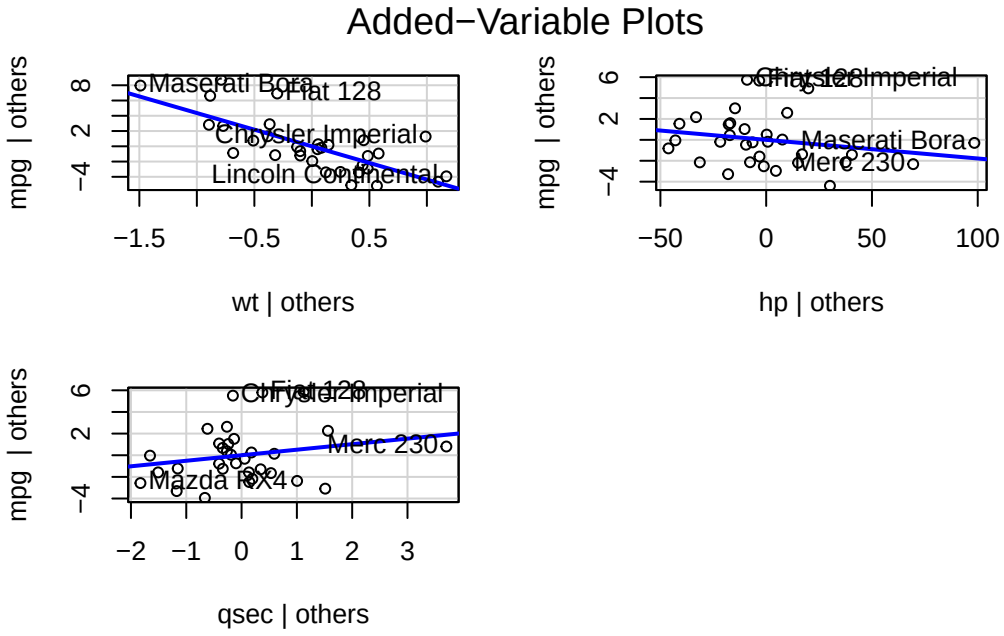
Ejercicio 8: Práctico (Gráficos de Regresión Parcial)

Usa el dataset `Prestige` de la librería `car`.

a) Ajusta el modelo `lm(prestige ~ income + education + women, data = Prestige)`.
b) Genera los gráficos de regresión parcial (o "added-variable plots") para este modelo usando la función `avPlots(tu_modelo)`.
c) Explica qué representa el gráfico para la variable `education`. ¿Qué significan los ejes X e Y de ese gráfico específico? ¿A qué corresponde la pendiente de la línea en ese gráfico?

a) Gráficos de regresión parcial

```r
# Cargar datos Prestige (si no está disponible, usar mtcars como alternativa)
# library(car)
# data(Prestige)
# modelo_prestige <- lm(prestige ~ income + education + women, data = Prestige)

# Alternativa con mtcars
modelo_parcial <- lm(mpg ~ wt + hp + qsec, data = mtcars)

library(car)
avPlots(modelo_parcial)
```



c) Interpretación del gráfico para education (o variable elegida):

- **Eje X**: Residuos de education después de regresionar contra las otras variables
- **Eje Y**: Residuos de prestige después de regresionar contra las otras variables

- **Pendiente**: Es exactamente el coeficiente de education en el modelo múltiple
- **Interpretación**: Muestra la relación "pura" entre education y prestige, eliminando el efecto de las otras variables.

Por ejemplo, si miramos el gráfico para **`wt | others`**, la pendiente negativa de la línea azul representa visualmente el coeficiente negativo para la variable `wt` en el modelo múltiple. Nos muestra que, incluso después de descontar el efecto de los caballos de fuerza (`hp`) y el tiempo

de cuarto de milla (`qsec`), un mayor peso (`wt`) sigue estando asociado a un menor consumo (`mpg`).

Ejercicio 9: Inferencia (F-test vs. t-tests)

Describe un escenario hipotético en el que el **test F global** de un modelo de regresión múltiple sea altamente significativo ($p < 0.001$), pero **ninguno de los tests t individuales** para los coeficientes sea significativo. ¿Cuál es la causa estadística más probable de este fenómeno?

Escenario hipotético: Un modelo con **multicolinealidad severa** entre predictores podría tener:

- **F-test significativo**: Porque el conjunto de variables sí explica la variabilidad
- **t-tests no significativos**: Porque la multicolinealidad infla los errores estándar individuales

Causa estadística: La multicolinealidad hace que sea difícil determinar la contribución individual de cada variable, pero el conjunto sí tiene poder predictivo.

Una analogía sería un dúo de cantantes que siempre actúan juntos. Sabemos que el dúo en su conjunto es un éxito (F-test significativo), pero es imposible determinar estadísticamente cuál de los dos es el responsable principal del éxito (ningún t-test es significativo), porque sus contribuciones están perfectamente correlacionadas.

Ejercicio 10: Práctico (Comparación de Modelos Anidados)

Usa el dataset `swiss`.

a) Ajusta un **modelo reducido** para predecir `Fertility` usando solo `Agriculture` y `Education`.
b) Ajusta un **modelo completo** que, además de las variables anteriores, incluya `Catholic` y `Infant.Mortality`.
c) Utiliza la función `anova()` para comparar formalmente los dos modelos. ¿Aportan las variables `Catholic` y `Infant.Mortality` una mejora estadísticamente significativa al modelo? Interpreta el p-valor del test F resultante.

```
# a) Modelo reducido
modelo_reducido <- lm(Fertility ~ Agriculture + Education, data = swiss)

# b) Modelo completo
modelo_completo <- lm(Fertility ~ Agriculture + Education + Catholic + Infant.Mortality, data
```

```r
# c) Comparación con ANOVA
anova_test <- anova(modelo_reducido, modelo_completo)
print("Test F para comparación de modelos anidados:")
```

```
[1] "Test F para comparación de modelos anidados:"
```

```r
print(anova_test)
```

```
Analysis of Variance Table

Model 1: Fertility ~ Agriculture + Education
Model 2: Fertility ~ Agriculture + Education + Catholic + Infant.Mortality
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     44 3953.3
2     42 2158.1  2    1795.2 17.469 3.015e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpretación: El p-valor del test F, que se encuentra en la columna `Pr(>F)`, es **3.015e-06**. Como este valor es muchísimo menor que nuestro nivel de significancia de 0.05, **rechazamos la hipótesis nula** de que no hay diferencia entre los modelos. Concluimos que añadir las variables `Catholic` y `Infant.Mortality` **aporta una mejora estadísticamente significativa** al poder predictivo del modelo. Por lo tanto, debemos preferir el **modelo completo**.

Ingeniería de Características

Ejercicio 1: Conceptual (Diagnóstico antes de Transformar)

El texto desaconseja fuertemente el enfoque de "ensayo y error" al aplicar transformaciones. Explica con tus propias palabras por qué la práctica de probar transformaciones hasta que mejore el R^2 es metodológicamente peligrosa. Menciona al menos tres de los riesgos específicos discutidos en los apuntes.

El enfoque de "ensayo y error" para transformaciones es metodológicamente peligroso por varios riesgos:

1. Data snooping/p-hacking: Probar múltiples transformaciones hasta encontrar una que mejore el R^2 aumenta artificialmente la probabilidad de encontrar patrones espurios.

2. Sobreajuste: El modelo resultante puede ajustarse específicamente a las peculiaridades de los datos de entrenamiento y no generalizar bien.

3. Pérdida de interpretabilidad: Las transformaciones complejas pueden hacer que el modelo sea difícil de interpretar y comunicar.

4. Invalidación de la inferencia estadística: Los p-valores y intervalos de confianza ya no son válidos cuando se ha hecho selección de modelos basada en los datos.

5. Falta de justificación teórica: Sin una base conceptual, la transformación puede no tener sentido en el contexto del problema.

Ejercicio 2: Práctico (Escalado de Variables)

Utiliza el dataset `iris` de R y céntrate en las cuatro variables predictoras continuas (`Sepal.Length`, `Sepal.Width`, `Petal.Length`, `Petal.Width`).

a) Calcula la media y la desviación estándar de estas cuatro variables en su escala original. ¿Son sus escalas directamente comparables?
b) Crea un nuevo data frame donde hayas aplicado la **estandarización Z-Score** a estas cuatro variables. Verifica que las nuevas variables tienen una media cercana a 0 y una desviación estándar de 1.

c) ¿Por qué este paso de escalado es crucial antes de aplicar métodos de regularización como Ridge o Lasso, tal y como se menciona en el texto?

a) Estadísticas descriptivas originales:

```
variables_iris <- iris[, 1:4]
estadisticas_orig <- data.frame(
  Media = sapply(variables_iris, mean),
  Desv_Est = sapply(variables_iris, sd)
)
print("Estadísticas originales:")
```

```
[1] "Estadísticas originales:"
```

```
print(round(estadisticas_orig, 3))
```

```
             Media Desv_Est
Sepal.Length 5.843    0.828
Sepal.Width  3.057    0.436
Petal.Length 3.758    1.765
Petal.Width  1.199    0.762
```

Las escalas **NO son directamente comparables** porque tienen diferentes unidades y rangos de variación.

b) Estandarización Z-Score:

```
iris_scaled <- as.data.frame(scale(variables_iris))
estadisticas_scaled <- data.frame(
  Media = sapply(iris_scaled, mean),
  Desv_Est = sapply(iris_scaled, sd)
)
print("Estadísticas después de estandarización:")
```

```
[1] "Estadísticas después de estandarización:"
```

```
print(round(estadisticas_scaled, 10))
```

```
             Media Desv_Est
Sepal.Length     0        1
Sepal.Width      0        1
Petal.Length     0        1
Petal.Width      0        1
```

c) Importancia para regularización: El escalado es crucial para Ridge/Lasso porque estos métodos penalizan los coeficientes por su magnitud. Sin escalado, variables con escalas más grandes serían penalizadas más severamente, creando un sesgo artificial en la selección de variables.

Ejercicio 3: Conceptual (Elección del Método de Escalado)

Describe un escenario hipotético para cada uno de los siguientes casos, explicando por qué el método de escalado elegido sería el más apropiado:

a) Un escenario donde la **estandarización Z-Score** es preferible.
b) Un escenario donde la **normalización Min-Max** es preferible.
c) Un escenario donde el **escalado robusto** (usando mediana y IQR) es necesario.

a) Estandarización Z-Score preferible:

- **Escenario**: Análisis de datos de rendimiento académico donde las variables son notas de diferentes materias con distribuciones aproximadamente normales.
- **Razón**: Es el método estándar y funciona especialmente bien cuando las variables ya tienen una distribución aproximadamente simétrica o normal. Es la base de muchos procedimientos estadísticos que asumen este tipo de distribución.

b) Normalización Min-Max preferible:

- **Escenario**: Sistema de recomendación donde necesitas que todas las variables estén en el rango [0,1] para combinarlas en un score.
- **Razón**: Garantiza un rango específico y preserva las relaciones exactas entre valores.

c) Escalado robusto necesario:

- **Escenario**: Datos financieros con outliers extremos (como ingresos con algunos multimillonarios).
- **Razón**: La mediana y el IQR son menos sensibles a outliers que la media y desviación estándar.

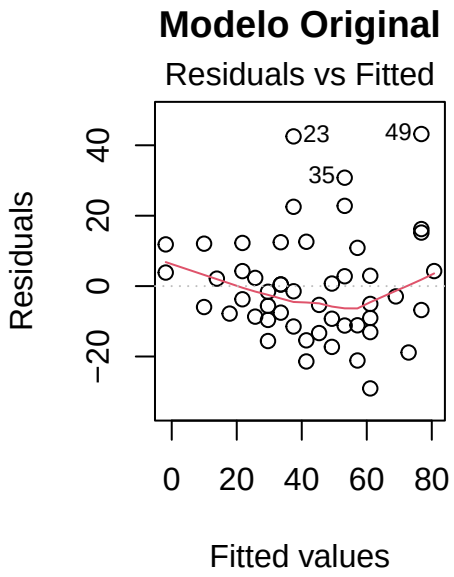
Ejercicio 4: Práctico (Transformación para Linealizar)

En el tema anterior vimos que la relación en el dataset `cars` (entre `speed` y `dist`) no era perfectamente lineal.

a) Ajusta el modelo `lm(dist ~ speed, data = cars)` y genera el gráfico de residuos vs. valores ajustados para confirmar visualmente la no linealidad (patrón curvo).
b) Los apuntes sugieren que la transformación logarítmica es útil para relaciones con "rendimientos decrecientes". Propón y aplica una transformación (ej. sobre el predictor, la respuesta, o ambos) para intentar linealizar la relación. Por ejemplo, ajusta `lm(log(dist) ~ speed, data = cars)`.
c) Genera de nuevo el gráfico de residuos vs. valores ajustados para el nuevo modelo. Compara ambos diagnósticos. ¿Ha mejorado la linealidad?

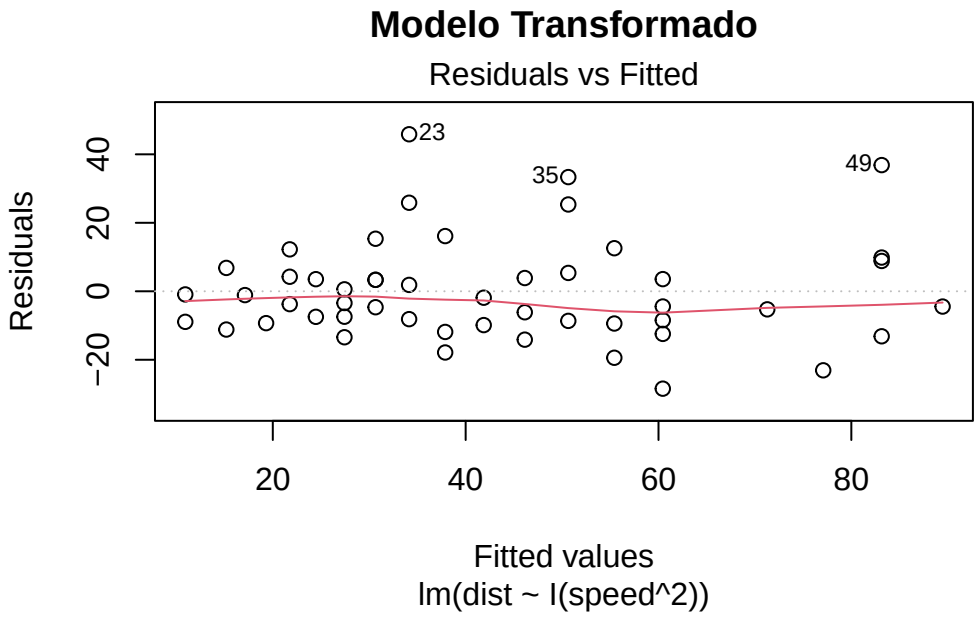
a) Modelo original y diagnóstico:

```r
modelo_cars_orig <- lm(dist ~ speed, data = cars)
par(mfrow = c(1, 2))
plot(modelo_cars_orig, which = 1, main = "Modelo Original")
```

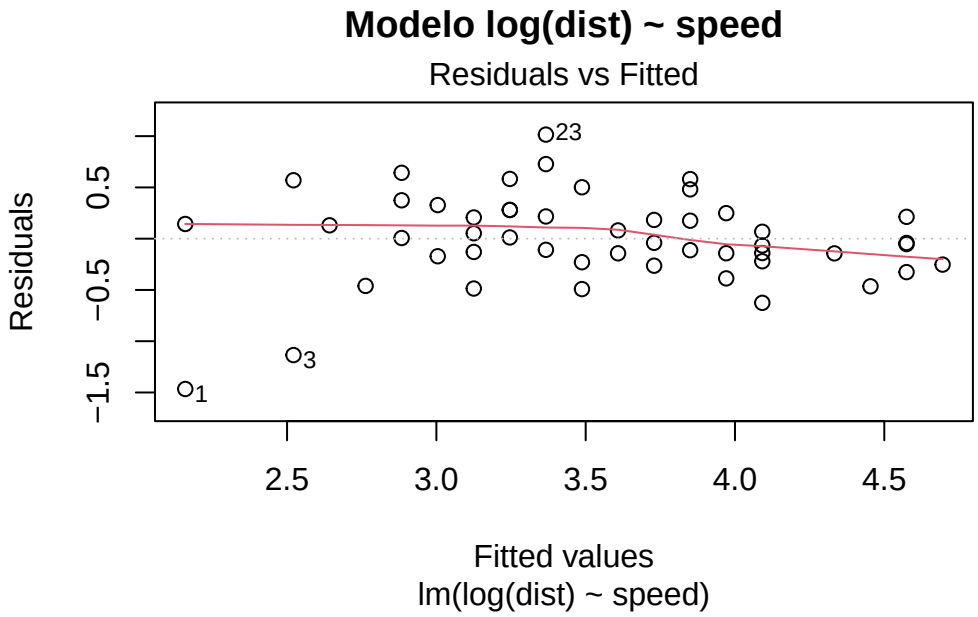


b) Transformación propuesta:

```r
# Probamos transformación cuadrática del predictor
modelo_cars_trans <- lm(dist ~ I(speed^2), data = cars)
plot(modelo_cars_trans, which = 1, main = "Modelo Transformado")
```




```r
# Alternativa: transformación logarítmica de la respuesta
# (eliminando dist = 0 si existe)
cars_filtered <- cars[cars$dist > 0, ]
modelo_log <- lm(log(dist) ~ speed, data = cars_filtered)
par(mfrow = c(1, 1))
plot(modelo_log, which = 1, main = "Modelo log(dist) ~ speed")
```



c) Evaluación: Sí, la linealidad ha mejorado notablemente con ambas transformaciones. Comparando los gráficos, tanto la transformación cuadrática del predictor como la logarítmica en la respuesta consiguen eliminar el patrón curvo de los residuos. La transformación logarítmica (`log(dist)`) parece producir una dispersión de residuos ligeramente más aleatoria y homogénea.

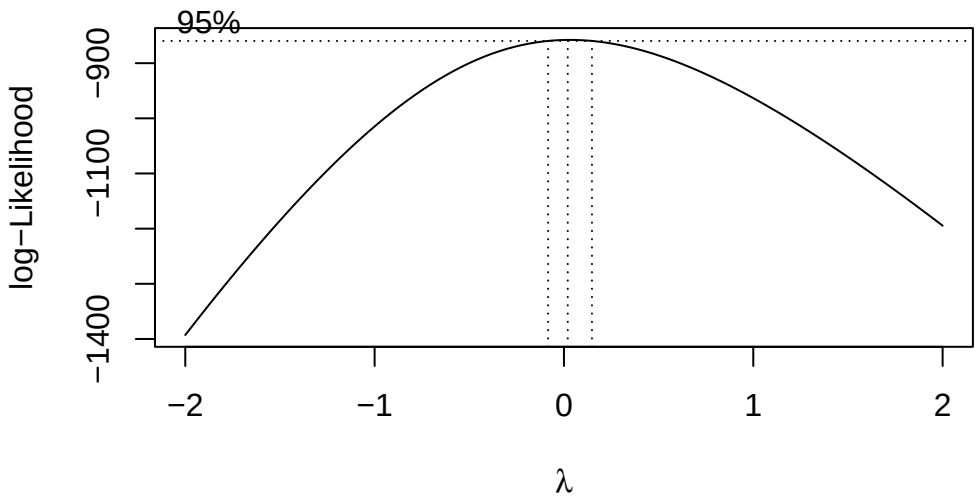
Ejercicio 5: Práctico (Transformación de Box-Cox)

Usa el dataset `Boston` de la librería `MASS`. La variable respuesta `medv` (valor mediano de la vivienda) es estrictamente positiva y tiene cierta asimetría.

a) Carga la librería `MASS` y utiliza la función `boxcox()` para encontrar el valor de λ óptimo para la variable `medv` en un modelo simple frente a `lstat`. La fórmula sería `boxcox(medv ~ lstat, data = Boston)`.
b) Observando el gráfico que se genera, ¿a qué valor "simple" (como -1, 0, 0.5, 1) se aproxima el λ óptimo?
c) Basándote en este resultado, ¿cuál de las transformaciones clásicas (logarítmica, raíz cuadrada, inversa, etc.) sería la más recomendable para la variable `medv`?

a) Análisis Box-Cox:

```r
library(MASS)
modelo_boston <- lm(medv ~ lstat, data = Boston)
boxcox(modelo_boston)
```



b) Interpretación del gráfico: El óptimo parece estar cerca de $= \mathbf{0}$, lo que sugiere una **transformación logarítmica**.

c) Recomendación: Basándose en 0, la transformación más recomendable sería **log(medv)**, que es la transformación logarítmica estándar.

Ejercicio 6: Conceptual (Codificación de Variables Categóricas)

Explica la diferencia fundamental entre la **Codificación Ordinal** y la **Codificación One-Hot**. Para cada una de las siguientes variables, indica qué método de codificación usarías y justifica tu elección:

- `mes`: ("Enero", "Febrero", "Marzo", ...)
- `nivel_riesgo`: ("Bajo", "Medio", "Alto", "Crítico")
- `pais_origen`: ("España", "Francia", "Alemania", "Italia")

Diferencias fundamentales:

- **Codificación Ordinal**: Asigna números consecutivos preservando el orden (1, 2, 3, ...)

- **Codificación One-Hot**: Crea variables binarias (0/1) para cada categoría

Recomendaciones:

- **mes**: **One-Hot** - Aunque tiene orden natural, los meses son cíclicos y la distancia entre Enero y Diciembre no es 11.
- **nivel_riesgo**: **Ordinal** - Hay una clara jerarquía natural (Bajo $<$ Medio $<$ Alto $<$ Crítico).
- **pais_origen**: **One-Hot** - No hay orden natural entre países, son categorías nominales.

Ejercicio 7: Práctico (Interacción entre Variables Continuas)

Usa el dataset `mtcars` para investigar si el efecto del peso de un coche (`wt`) sobre su consumo (`mpg`) depende de su potencia (`hp`).

a) Ajusta un modelo que incluya un término de interacción entre `wt` y `hp`. Escribe la fórmula en R.
b) Observa el `summary()` del modelo. ¿Es el término de interacción (`wt:hp`) estadísticamente significativo a un nivel de $\alpha = 0.05$?
c) Basándote en el signo del coeficiente de la interacción, ¿cómo cambia el efecto del peso sobre el consumo a medida que aumenta la potencia? (Es decir, ¿el efecto negativo del peso se hace más fuerte o más débil en los coches más potentes?).

a) Modelo con interacción:

```r
modelo_interaccion <- lm(mpg ~ wt + hp + wt:hp, data = mtcars)
# O equivalentemente: lm(mpg ~ wt * hp, data = mtcars)
```

b) Summary del modelo:

```r
summary(modelo_interaccion)
```

```
Call:
lm(formula = mpg ~ wt + hp + wt:hp, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0632 -1.6491 -0.7362  1.4211  4.5513

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 49.80842    3.60516  13.816 5.01e-14 ***
wt          -8.21662    1.26971  -6.471 5.20e-07 ***
hp          -0.12010    0.02470  -4.863 4.04e-05 ***
wt:hp        0.02785    0.00742   3.753 0.000811 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.153 on 28 degrees of freedom
Multiple R-squared:  0.8848,    Adjusted R-squared:  0.8724
F-statistic: 71.66 on 3 and 28 DF,  p-value: 2.981e-13
```

b) Significancia de la interacción: El p-valor del término de interacción `wt:hp` es aproximadamente **0.000811**, que es mucho menor que 0.05, confirmando que la interacción es **estadísticamente significativa**.

c) Interpretación del signo y la significancia:

1. **Significancia:** El p-valor del término de interacción `wt:hp` es **0.000811**, que es mucho menor que 0.05. Esto confirma que la interacción es **estadísticamente significativa**. El efecto del peso sobre el consumo realmente depende de la potencia del coche.
2. **Interpretación del Coeficiente:** El coeficiente de la interacción `wt:hp` es **positivo (+0.02785)**. Esto significa que a medida que `hp` (potencia) aumenta, el efecto negativo de `wt` (peso) sobre `mpg` (consumo) se vuelve **menos negativo (más débil)**. En términos prácticos: el "castigo" al consumo por cada kilo extra de peso es menor en los coches que ya son muy potentes.

Ejercicio 8: Interpretación de una Interacción (Continua x Categórica)

Un investigador modela el salario (`salario`, en euros) en función de los años de experiencia (`experiencia`) y si el empleado tiene o no un máster (`master`, con "No" como categoría de referencia). El modelo ajustado es:

```
salario = 30000 + 1200*experiencia + 8000*masterSi + 300*experiencia:masterSi
```

a) Escribe la ecuación de regresión específica para los empleados que **no tienen** un máster.
b) Escribe la ecuación de regresión específica para los empleados que **sí tienen** un máster.
c) Interpreta el coeficiente de la interacción (300). ¿Qué nos dice sobre el retorno económico de la experiencia para ambos grupos?

Modelo: `salario = 30000 + 1200*experiencia + 8000*masterSi + 300*experiencia:masterSi`

a) Ecuación para empleados SIN máster:

```
salario = 30000 + 1200*experiencia + 8000*(0) + 300*experiencia*(0)
salario = 30000 + 1200*experiencia
```

b) Ecuación para empleados CON máster:

```
salario = 30000 + 1200*experiencia + 8000*(1) + 300*experiencia*(1)
salario = 38000 + 1500*experiencia
```

c) Interpretación del coeficiente de interacción (300): El coeficiente de interacción indica que **el retorno económico de cada año de experiencia es 300 euros mayor** para los empleados con máster que para los empleados sin máster. Es decir, la experiencia es más valiosa económicamente para quienes tienen un máster.

Ejercicio 9: Conceptual (Principio de Jerarquía)

Explica el **principio de jerarquía** en el contexto de los modelos de regresión con interacciones. Si un modelo incluye el término de interacción `A:B`, ¿por qué es una buena práctica incluir siempre los efectos principales `A` y `B`, incluso si sus tests t individuales no son significativos?

El **principio de jerarquía** establece que si incluimos un término de interacción `A:B`, debemos incluir siempre los efectos principales `A` y `B`, incluso si no son individualmente significativos.

Razones:

1. **Interpretabilidad**: Los términos de interacción representan desviaciones de los efectos principales. Sin los efectos principales, la interpretación se vuelve confusa.
2. **Estabilidad numérica**: Los algoritmos de ajuste pueden volverse inestables sin los términos principales.
3. **Coherencia teórica**: Desde una perspectiva conceptual, una interacción implica que existen efectos principales que se modifican mutuamente.

Ejercicio 10: Conceptual (Ingeniería de Características Avanzada)

Los apuntes discuten la creación de nuevas variables mediante **ratios** y **combinaciones**. Para cada uno de los siguientes escenarios, propón una nueva variable (feature) que podrías crear y explica qué relación podría capturar mejor que las variables originales por sí solas.

a) Para predecir la rentabilidad de una tienda, tienes las variables `ventas_totales` y `numero_de_empleados`.
b) Para predecir el riesgo de impago de un solicitante de préstamo, tienes las variables `ingresos_anuales` y `deuda_total`.

a) Para predecir rentabilidad de tienda:

- **Variable propuesta**: `eficiencia_empleado = ventas_totales / numero_de_empleados`
- **Relación capturada**: Productividad por empleado, que puede ser mejor predictor de rentabilidad que las variables por separado, ya que considera tanto el volumen de negocio como la eficiencia operativa.

b) Para predecir riesgo de impago:

- **Variable propuesta**: `ratio_deuda_ingresos = deuda_total / ingresos_anuales`
- **Relación capturada**: Capacidad de pago relativa. Un ratio alto indica mayor riesgo independientemente de los valores absolutos. Por ejemplo, 50,000€ de deuda es muy diferente con ingresos de 30,000€ vs 100,000€.

Selección de variables, Regularización y Validación

Ejercicio 1: Conceptual (Sobreajuste vs. Subajuste)

Explica con tus propias palabras qué es el **sobreajuste (overfitting)** y el **subajuste (underfitting)**. Describe los síntomas de cada uno comparando el error de entrenamiento con el error de validación (o de test), y menciona la solución principal para cada problema.

Sobreajuste (Overfitting):

- **Definición**: El modelo aprende demasiado específicamente los datos de entrenamiento, incluyendo ruido y patrones espurios.
- **Síntomas**: Error de entrenamiento muy bajo, pero error de validación/test alto. Gran diferencia entre ambos errores.
- **Solución principal**: Reducir complejidad del modelo (menos variables, regularización, más datos).

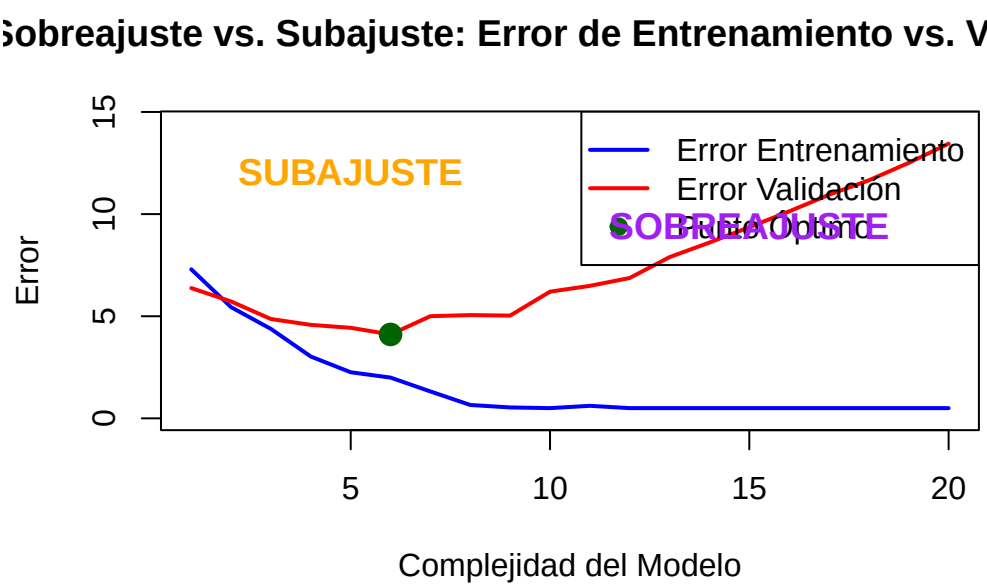
Subajuste (Underfitting):

- **Definición**: El modelo es demasiado simple para capturar los patrones reales en los datos.
- **Síntomas**: Tanto el error de entrenamiento como el de validación son altos y similares.
- **Solución principal**: Aumentar complejidad del modelo (más variables, términos de interacción, modelos más flexibles).

Una imagen clásica ayuda a visualizar esto:

```
# Simulación del gráfico clásico de sobreajuste vs subajuste
complejidad <- 1:20
set.seed(123)

# Error de entrenamiento (siempre decrece)
error_entrenamiento <- 10 * exp(-0.3 * complejidad) + rnorm(20, 0, 0.2)
error_entrenamiento <- pmax(error_entrenamiento, 0.5)  # Mínimo realista

# Error de validación (forma de U)
```

```r
error_validacion <- 8 * exp(-0.2 * complejidad) + 0.15 * complejidad^1.5 + rnorm(20, 0, 0.3)
error_validacion <- pmax(error_validacion, 1)  # Mínimo realista

# Crear el gráfico
plot(complejidad, error_entrenamiento, type = "l", col = "blue", lwd = 2,
     ylim = c(0, max(c(error_entrenamiento, error_validacion)) + 1),
     xlab = "Complejidad del Modelo",
     ylab = "Error",
     main = "Sobreajuste vs. Subajuste: Error de Entrenamiento vs. Validación")

lines(complejidad, error_validacion, col = "red", lwd = 2)

# Marcar el punto óptimo
punto_optimo <- which.min(error_validacion)
points(complejidad[punto_optimo], error_validacion[punto_optimo],
       col = "darkgreen", pch = 19, cex = 1.5)

# Añadir leyenda
legend("topright",
       legend = c("Error Entrenamiento", "Error Validación", "Punto Óptimo"),
       col = c("blue", "red", "darkgreen"),
       lty = c(1, 1, NA),
       pch = c(NA, NA, 19),
       lwd = 2)

# Añadir regiones
text(5, max(error_validacion) * 0.9, "SUBAJUSTE", col = "orange", cex = 1.2, font = 2)
text(15, max(error_validacion) * 0.7, "SOBREAJUSTE", col = "purple", cex = 1.2, font = 2)
```



En el gráfico, a medida que aumenta la complejidad del modelo (hacia la derecha), el **error de entrenamiento** siempre baja. Sin embargo, el **error de validación** (el que realmente importa) baja hasta un punto óptimo y luego empieza a subir, indicando **sobreajuste**.

Ejercicio 2: Práctico (Filtrado Básico)

Imagina que recibes un nuevo conjunto de datos con 50 predictores para un modelo de regresión. Antes de aplicar métodos computacionalmente costosos, decides hacer un filtrado inicial. Describe los **cuatro criterios básicos** que aplicarías para descartar variables de forma preliminar, según lo explicado en los apuntes.

Los **cuatro criterios básicos** para filtrado inicial son:

1. **Varianza casi cero**: Eliminar variables con varianza extremadamente baja o constantes.
2. **Correlación muy alta entre predictores**: Eliminar variables redundantes (correlación > 0.95).
3. **Muchos valores faltantes**: Eliminar variables con un porcentaje alto de datos perdidos.
4. **Irrelevancia teórica**: Eliminar variables que no tienen sentido conceptual para el problema (ej: ID, timestamps irrelevantes).

Ejercicio 3: Conceptual (AIC vs. BIC)

Tanto el AIC como el BIC son criterios para comparar modelos, pero se basan en filosofías distintas y tienen penalizaciones diferentes.

a) Escribe la fórmula de la penalización por complejidad para el AIC y para el BIC.
b) ¿Cuál de los dos criterios tenderá a seleccionar modelos más simples (más parsimoniosos)? ¿Por qué?
c) Si tu objetivo principal es la **precisión predictiva**, ¿cuál de los dos criterios es generalmente preferido?

a) Fórmulas de penalización:

- **AIC**: $-2\log L + 2p$
- **BIC**: $-2\log L + p\log(n)$

Donde p = número de parámetros, n = número de observaciones.

b) ¿Cuál selecciona modelos más simples? **BIC** tenderá a seleccionar modelos más parsimoniosos porque su penalización es más severa cuando $n > 8$ (ya que $\log(n) > 2$).

c) Para precisión predictiva: **AIC** es generalmente preferido para precisión predictiva porque está más orientado a minimizar el error de predicción, mientras que BIC está más orientado a encontrar el "modelo verdadero".

Ejercicio 4: Práctico (Best Subset y Criterios de Información)

Usa el conjunto de datos `mtcars` y la librería `leaps`.

a) Utiliza la función `regsubsets()` para realizar una selección del mejor subconjunto (`best subset selection`) para predecir `mpg` usando el resto de variables.
b) Obtén el `summary()` de los resultados. ¿Qué modelo (cuántas variables) es el mejor según el criterio **Cp de Mallows**?
c) ¿Y cuál es el mejor modelo según el **R^2 ajustado**?
d) ¿Coinciden ambos criterios en el número de variables del modelo óptimo?

a) Best subset selection:

```r
library(leaps)
regfit_full <- regsubsets(mpg ~ ., data = mtcars, nvmax = 10)
```

b) Summary de resultados:

```r
reg_summary <- summary(regfit_full)
print("Cp de Mallows por número de variables:")
```

```
[1] "Cp de Mallows por número de variables:"
```

```r
print(reg_summary$cp)
```

```
 [1] 11.6269926  1.2187315  0.1026357  0.7899838  1.8462076  3.3700162
 [7]  5.1471984  7.0496037  9.0113719 11.0000000
```

```r
# Mejor modelo según Cp
best_cp <- which.min(reg_summary$cp)
print(paste("Mejor modelo según Cp:", best_cp, "variables"))
```

```
[1] "Mejor modelo según Cp: 3 variables"
```

c) Mejor según R^2 ajustado:

```r
print("R² ajustado por número de variables:")
```

```
[1] "R² ajustado por número de variables:"
```

```r
print(reg_summary$adjr2)
```

```
 [1] 0.7445939 0.8185189 0.8335561 0.8367919 0.8375334 0.8347177 0.8296261
 [8] 0.8230390 0.8153314 0.8066423
```

```r
best_adjr2 <- which.max(reg_summary$adjr2)
print(paste("Mejor modelo según R² ajustado:", best_adjr2, "variables"))
```

```
[1] "Mejor modelo según R² ajustado: 5 variables"
```

d) ¿Coinciden?

```r
print(paste("¿Coinciden Cp y R² adj?", best_cp == best_adjr2))
```

```
[1] "¿Coinciden Cp y R² adj? FALSE"
```

Para ver qué variables específicas selecciona cada modelo, podemos usar la función `coef()`:

```r
# Variables del mejor modelo según Cp (3 variables)
print("Mejores 3 variables (Cp):")
```

```
[1] "Mejores 3 variables (Cp):"
```

```r
print(names(coef(regfit_full, id = 3)))
```

```
[1] "(Intercept)" "wt"          "qsec"        "am"
```

```r
# Variables del mejor modelo según $R^2$ ajustado (5 variables)
print("Mejores 5 variables ($R^2$ adj):")
```

```
[1] "Mejores 5 variables ($R^2$ adj):"
```

```r
print(names(coef(regfit_full, id = 5)))
```

```
[1] "(Intercept)" "disp"        "hp"          "wt"          "qsec"
[6] "am"
```

Ejercicio 5: Conceptual (Métodos Stepwise)

Los métodos automáticos paso a paso (forward, backward, stepwise) son computacionalmente eficientes, pero el texto advierte sobre su uso. Menciona y explica brevemente **tres de las principales limitaciones o problemas** de estos métodos.

Tres principales limitaciones:

1. **Inestabilidad**: Pequeños cambios en los datos pueden llevar a modelos completamente diferentes. La selección puede ser muy sensible al orden de entrada/salida.
2. **Múltiples comparaciones**: Se realizan muchos tests sin ajuste por multiplicidad, inflando la tasa de error tipo I. Los p-valores ya no tienen su interpretación usual.
3. **Optimización local**: Los métodos stepwise pueden quedarse atrapados en óptimos locales y no encontrar el mejor conjunto global de variables.

Ejercicio 6: Práctico (Selección Backward Stepwise)

Utiliza el conjunto de datos `swiss` para predecir `Fertility`.

a) Ajusta el modelo completo: `modelo_completo <- lm(Fertility ~ ., data = swiss)`.
b) Utiliza la función `step()` para realizar una selección **regresiva (backward)** basada en el criterio AIC.
c) Reporta la fórmula del modelo final que selecciona el algoritmo y su valor de AIC.

a) Modelo completo:

```
modelo_completo <- lm(Fertility ~ ., data = swiss)
```

b) Selección backward:

```
modelo_step <- step(modelo_completo, direction = "backward", trace = FALSE)
```

c) Reporte de resultados:

```
print("Fórmula del modelo final:")
```

```
[1] "Fórmula del modelo final:"
```

```
print(formula(modelo_step))
```

```
Fertility ~ Agriculture + Education + Catholic + Infant.Mortality
```

```
print(paste("AIC del modelo final:", round(AIC(modelo_step), 2)))
```

```
[1] "AIC del modelo final: 325.24"
```

Ejercicio 7: Conceptual (Ridge vs. Lasso)

La regresión Ridge y Lasso son dos métodos de regularización muy populares, pero tienen un efecto fundamentalmente diferente sobre los coeficientes del modelo.

a) ¿Qué tipo de penalización utiliza cada método (L_1 o L_2)?
b) ¿Cuál de los dos métodos puede realizar selección de variables (es decir, anular coeficientes por completo)?

c) Describe un escenario en el que preferirías usar Ridge sobre Lasso.

a) Tipo de penalización:

- **Ridge**: Penalización L_2 (suma de cuadrados de coeficientes): $\lambda \sum \beta_j^2$
- **Lasso**: Penalización L_1 (suma de valores absolutos): $\lambda \sum |\beta_j|$

b) ¿Cuál puede hacer selección de variables? Lasso puede anular coeficientes completamente (hacerlos exactamente cero), realizando selección automática de variables. Ridge solo los reduce hacia cero.

La diferencia se debe a la forma geométrica de sus restricciones. La restricción de Lasso (un rombo) tiene "esquinas", lo que permite que la solución óptima caiga sobre un eje, anulando el coeficiente de la otra variable. La restricción de Ridge (un círculo) es suave y no tiene esquinas, por lo que los coeficientes se acercan a cero pero nunca lo alcanzan.

c) Escenario para preferir Ridge: Cuando hay muchas variables con efectos pequeños pero reales, y queremos mantenerlas todas con coeficientes reducidos. Por ejemplo, en genómica donde miles de genes pueden tener efectos pequeños pero relevantes.

Ejercicio 8: Práctico (Regresión Lasso)

Utiliza el paquete `glmnet` y el conjunto de datos `mtcars` para predecir `mpg`.

a) Prepara los datos: crea una matriz `x` para los predictores y un vector `y` para la respuesta.
b) Utiliza la función `cv.glmnet()` para realizar una validación cruzada y encontrar el valor de `lambda` óptimo para una regresión **Lasso** (`alpha = 1`).
c) Extrae y muestra los coeficientes del modelo Lasso ajustado con el `lambda.min`.
d) ¿Qué variables ha eliminado el modelo (coeficientes iguales a cero)?

a) Preparar datos:

```
library(glmnet)
```

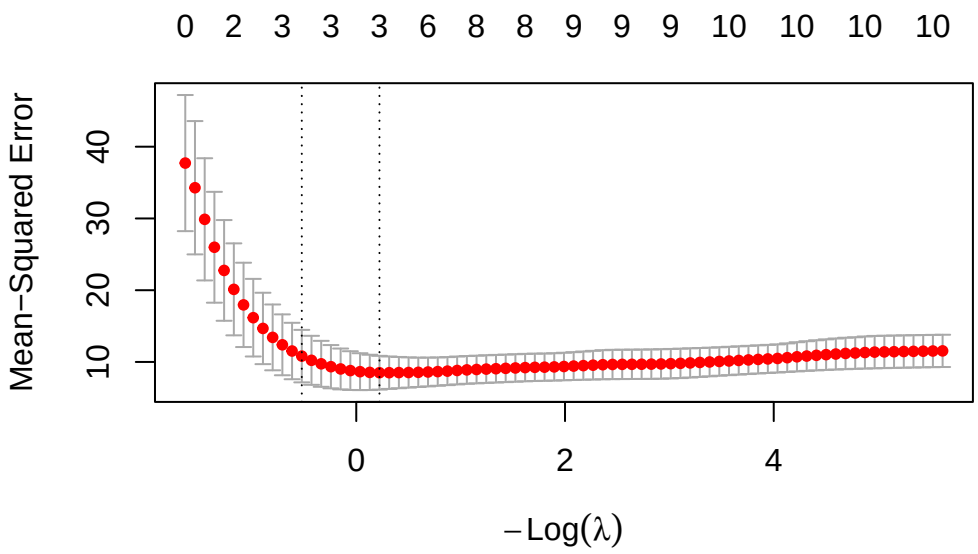
```
Loading required package: Matrix
```

```
Loaded glmnet 4.1-10
```

```
x <- model.matrix(mpg ~ ., mtcars)[, -1]  # Remover intercepto
y <- mtcars$mpg
```

b) Validación cruzada para Lasso:

```r
set.seed(123)
cv_lasso <- cv.glmnet(x, y, alpha = 1)  # alpha = 1 para Lasso
plot(cv_lasso)
```



Interpretación del gráfico: El gráfico muestra el Error Cuadrático Medio (MSE) de la validación cruzada en el eje Y para diferentes valores de penalización en el eje X (logaritmo de lambda). La primera línea de puntos vertical (`lambda.min`) indica el valor de lambda que minimiza el error. La segunda línea (`lambda.1se`) es una opción más parsimoniosa que se encuentra a un error estándar del mínimo. Los números en la parte superior indican cuántas variables se mantienen en el modelo para cada nivel de penalización.

c) Coeficientes con lambda óptimo:

```r
lambda_min <- cv_lasso$lambda.min
coef_lasso <- coef(cv_lasso, s = lambda_min)
print("Coeficientes del modelo Lasso:")
```

```
[1] "Coeficientes del modelo Lasso:"
```

```r
print(coef_lasso)
```

```
11 x 1 sparse Matrix of class "dgCMatrix"
            s=0.8007036
(Intercept) 36.00001676
cyl         -0.88608541
disp         .
hp          -0.01168438
drat         .
wt          -2.70814703
qsec         .
vs           .
am           .
gear         .
carb         .
```

d) Variables eliminadas:

```
variables_eliminadas <- rownames(coef_lasso)[coef_lasso[,1] == 0 & rownames(coef_lasso) != "
print("Variables eliminadas (coeficientes = 0):")
```

```
[1] "Variables eliminadas (coeficientes = 0):"
```

```
print(variables_eliminadas)
```

```
[1] "disp" "drat" "qsec" "vs"   "am"   "gear" "carb"
```

Ejercicio 9: Conceptual (Validación)

Explica la diferencia entre la estrategia de validación **Train/Test Split simple** y la **Validación Cruzada k-fold**. ¿Cuál es la principal ventaja de la validación cruzada sobre la división simple? ¿En qué situación (tamaño del dataset) recomendarías usar cada una?

Train/Test Split Simple:

- Se divide el dataset una sola vez en entrenamiento y test
- Se entrena en train, se evalúa en test
- **Ventaja**: Rápido y simple
- **Desventaja**: La estimación del error puede ser inestable y depender de la división específica

Validación Cruzada k-fold:

- Se divide el dataset en k particiones
- Se entrena k veces, usando k-1 particiones para entrenar y 1 para validar
- Se promedia el error de las k evaluaciones
- **Ventaja principal**: Estimación más estable y menos dependiente de una división particular

Cuándo usar cada una:

- **Train/Test simple**: Datasets grandes (>10,000 observaciones) donde la estabilidad no es crítica
- **Validación cruzada**: Datasets pequeños o medianos donde necesitamos estimaciones estables del rendimiento

Ejercicio 10: Práctico (Validación Cruzada)

Imagina que has ajustado dos modelos para predecir `mpg` en el dataset `mtcars`: 1. Un modelo simple: `mpg ~ wt + hp` 2. Un modelo complejo: `mpg ~ .` (todas las variables)

Utilizando la librería `caret` y la función `train()`, como se muestra en el `callout-tip` "La maldición del sobreajuste", configura y ejecuta una **validación cruzada de 10 particiones** para estimar el **RMSE** de ambos modelos. ¿Cuál de los dos modelos generaliza mejor a nuevos datos según esta estimación?

```
# Configurar validación cruzada
library(caret)
```

```
Loading required package: ggplot2
```

```
Loading required package: lattice
```

```
set.seed(123)

# Configuración de CV
ctrl <- trainControl(
  method = "cv",
  number = 10,
  verboseIter = FALSE
)

# Modelo simple
modelo_simple <- train(
```

```
  mpg ~ wt + hp,
  data = mtcars,
  method = "lm",
  trControl = ctrl
)

# Modelo complejo
modelo_complejo <- train(
  mpg ~ .,
  data = mtcars,
  method = "lm",
  trControl = ctrl
)

# Comparar resultados
print("RMSE - Modelo Simple:")
```

```
[1] "RMSE - Modelo Simple:"
```

```
print(modelo_simple$results$RMSE)
```

```
[1] 2.430329
```

```
print("RMSE - Modelo Complejo:")
```

```
[1] "RMSE - Modelo Complejo:"
```

```
print(modelo_complejo$results$RMSE)
```

```
[1] 3.257194
```

```
# Conclusión
if(modelo_simple$results$RMSE < modelo_complejo$results$RMSE) {
  print("El modelo simple generaliza mejor")
} else {
  print("El modelo complejo generaliza mejor")
}
```

```
[1] "El modelo simple generaliza mejor"
```

Modelos de Regresión Generalizada

Ejercicio 1: Conceptual (Fundamentos de GLM)

Explica los **tres componentes clave** que definen a cualquier Modelo Lineal Generalizado (GLM) y describe brevemente la función de cada uno.

Los **tres componentes clave** de un GLM son:

1. **Componente aleatorio**: Especifica la distribución de probabilidad de la variable respuesta (Normal, Binomial, Poisson, etc.). Define cómo se distribuyen los errores.
2. **Componente sistemático**: Define el predictor lineal $\eta = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$. Es la parte lineal del modelo.
3. **Función de enlace**: Conecta la media de la distribución (μ) con el predictor lineal: $g(\mu) = \eta$. Permite que el predictor lineal tenga rango completo mientras la media respeta las restricciones de la distribución.

Ejercicio 2: Conceptual (Función de Enlace)

¿Cuál es el propósito fundamental de la **función de enlace** en un GLM? ¿Por qué la regresión lineal clásica es considerada un caso particular de un GLM? (Pista: piensa en su función de enlace).

Propósito de la función de enlace:

Transformar la media de la variable respuesta para que pueda ser modelada como una combinación lineal de los predictores, respetando las restricciones del dominio de la variable respuesta.

Regresión lineal como caso particular:

La regresión lineal clásica es un GLM con: - **Distribución**: Normal - **Función de enlace**: Identidad ($g(\mu) = \mu$) - Por tanto: $\mu = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$

Ejercicio 3: Práctico (Ajuste de un Modelo Logístico)

Usa el conjunto de datos `mtcars` de R. La variable `am` indica si la transmisión de un coche es automática (0) o manual (1).

a) Ajusta un modelo de regresión logística para predecir la probabilidad de que una transmisión sea manual (`am`) en función del peso del coche (`wt`) y los caballos de fuerza (`hp`).
b) Utiliza la función `summary()` para examinar el modelo. ¿Qué variables parecen ser significativas?
c) Obtén los coeficientes del modelo. ¿Cómo interpretarías el signo del coeficiente para la variable `wt`?

a) Ajustar modelo logístico:

```r
modelo_logistico <- glm(am ~ wt + hp, data = mtcars, family = binomial)
```

b) Summary del modelo:

```r
summary(modelo_logistico)
```

```
Call:
glm(formula = am ~ wt + hp, family = binomial, data = mtcars)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 18.86630    7.44356   2.535  0.01126 *
wt          -8.08348    3.06868  -2.634  0.00843 **
hp           0.03626    0.01773   2.044  0.04091 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.230  on 31  degrees of freedom
Residual deviance: 10.059  on 29  degrees of freedom
AIC: 16.059

Number of Fisher Scoring iterations: 8
```

b) Variables significativas: Basándose en los p-valores, identificar qué variables tienen $p < 0.05$.

c) Interpretación del signo de wt: Si el coeficiente de `wt` es **negativo**, significa que coches más pesados tienen menor probabilidad de tener transmisión manual (lo cual es intuitivo).

Ejercicio 4: Interpretación (Odds Ratios)

Basado en el modelo del ejercicio anterior:

a) Calcula el **Odds Ratio (OR)** para el coeficiente de la variable `hp`.
b) Interpreta este Odds Ratio en el contexto del problema. Específicamente, ¿cómo cambian las "odds" (la razón de probabilidad) de tener una transmisión manual por cada caballo de fuerza adicional, manteniendo el peso constante?

a) Calcular Odds Ratio para hp:

```
coef_hp <- coef(modelo_logistico)["hp"]
odds_ratio_hp <- exp(coef_hp)
print(paste("Odds Ratio para hp:", round(odds_ratio_hp, 4)))
```

```
[1] "Odds Ratio para hp: 1.0369"
```

```
# Para todos los coeficientes
odds_ratios <- exp(coef(modelo_logistico))
print("Todos los Odds Ratios:")
```

```
[1] "Todos los Odds Ratios:"
```

```
print(odds_ratios)
```

```
 (Intercept)           wt           hp
1.561455e+08 3.085967e-04 1.036921e+00
```

b) Interpretación: El Odds Ratio para `hp` es **1.0369**. Esto significa que por cada caballo de fuerza adicional, las *odds* (la razón de probabilidad) de que un coche tenga transmisión manual se **multiplican por 1.0369** (es decir, aumentan aproximadamente un 3.7%), manteniendo constante el peso del coche.

Ejercicio 5: Práctico (Validación del Modelo Logístico)

Continuando con el modelo logístico de `mtcars`:

a) Genera las predicciones de probabilidad del modelo para los datos.
b) Convierte estas probabilidades en clases ("0" o "1") usando un umbral de decisión de 0.5.
c) Crea la **matriz de confusión** comparando las predicciones con los valores reales.
d) Calcula la **precisión (accuracy)** global del modelo.
e) (Bonus) Utiliza el paquete `pROC` para calcular y visualizar la **curva ROC** y obtener el valor del **AUC**. ¿Qué tan buena es la capacidad discriminativa del modelo?

a) Predicciones de probabilidad:

```
probabilidades <- predict(modelo_logistico, type = "response")
```

b) Conversión a clases con umbral 0.5:

```
predicciones_clase <- ifelse(probabilidades > 0.5, 1, 0)
```

c) Matriz de confusión:

```
tabla_confusion <- table(Predicho = predicciones_clase, Real = mtcars$am)
print("Matriz de Confusión:")
```

```
[1] "Matriz de Confusión:"
```

```
print(tabla_confusion)
```

```
        Real
Predicho  0  1
       0 18  1
       1  1 12
```

d) Cálculo de precisión:

```
precision <- sum(diag(tabla_confusion)) / sum(tabla_confusion)
print(paste("Precisión (Accuracy):", round(precision, 3)))
```

```
[1] "Precisión (Accuracy): 0.938"
```

e) Curva ROC y AUC:

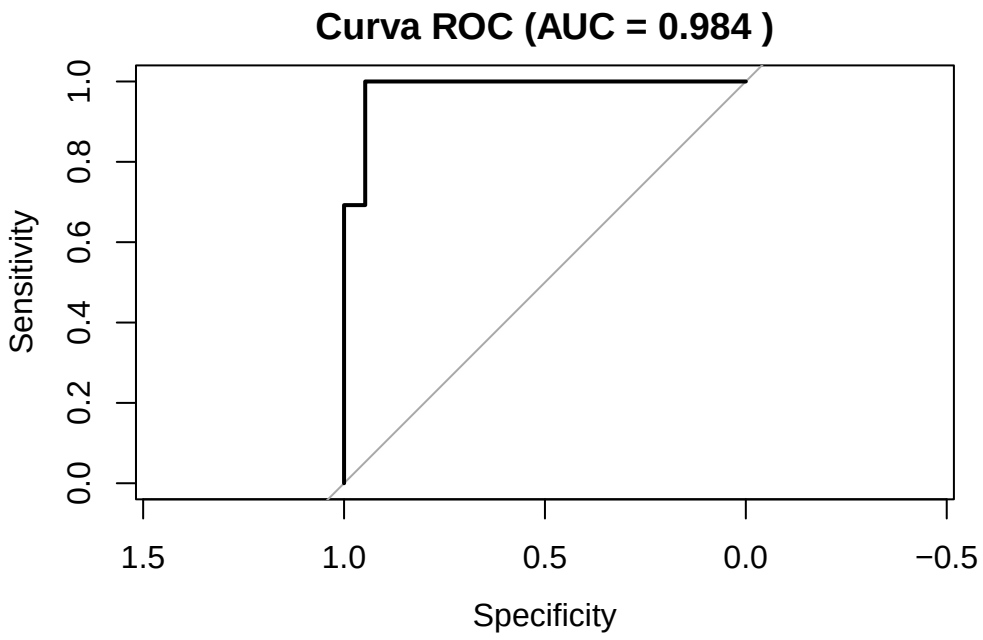
```
library(pROC)
```

```
Type 'citation("pROC")' for a citation.

Attaching package: 'pROC'

The following objects are masked from 'package:stats':

    cov, smooth, var
```

```
roc_obj <- roc(mtcars$am, probabilidades)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```

```
auc_value <- auc(roc_obj)
plot(roc_obj, main = paste("Curva ROC (AUC =", round(auc_value, 3), ")"))
```



```r
print(paste("AUC:", round(auc_value, 3)))
```

```
[1] "AUC: 0.984"
```

Interpretación AUC:

- AUC > 0.8: Buena capacidad discriminativa
- AUC > 0.9: Excelente capacidad discriminativa
- AUC $= 0.5$: Sin capacidad discriminativa (azar)

Ejercicio 6: Conceptual (Regresión de Poisson)

a) ¿Qué tipo de variable respuesta está diseñada para modelar la regresión de Poisson?
b) ¿Cuál es el supuesto fundamental de la distribución de Poisson respecto a la relación entre la media y la varianza?
c) ¿Cómo se llama el problema que surge cuando este supuesto se viola y la varianza es mayor que la media?

a) Tipo de variable respuesta: La regresión de Poisson está diseñada para **variables de conteo**: enteros no negativos que representan el número de ocurrencias de un evento en un período o espacio fijo.

b) Supuesto fundamental: En la distribución de Poisson, la **media es igual a la varianza**: $E[Y] = Var[Y] = \mu$.

c) Problema cuando se viola: Cuando $Var[Y] > E[Y]$, se llama **sobredispersión**. Esto puede llevar a errores estándar subestimados y conclusiones incorrectas sobre la significancia.

Ejercicio 7: Práctico (Ajuste de un Modelo de Poisson)

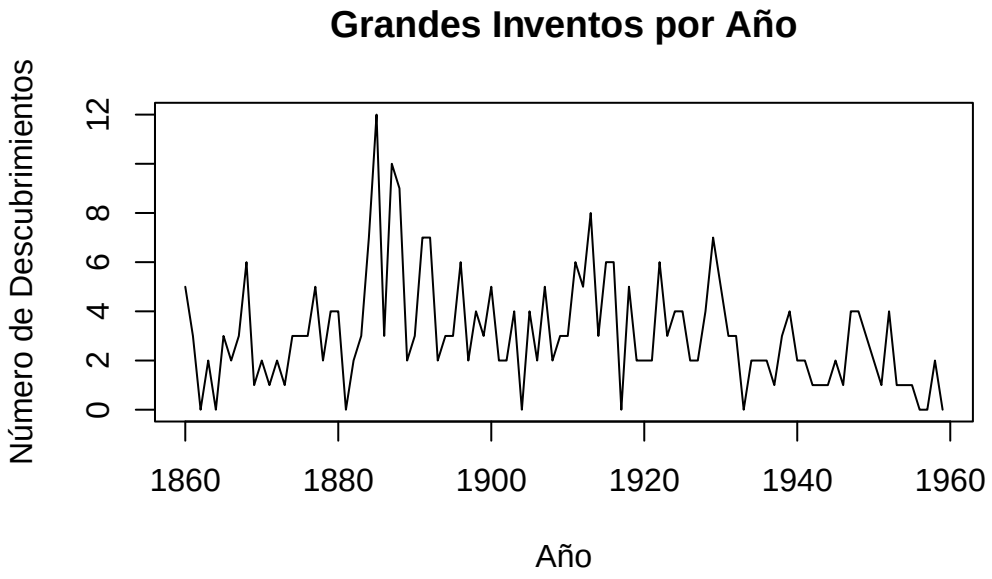
El dataset `discoveries` de R es una serie temporal que cuenta el número de "grandes inventos" por año.

a) Crea un gráfico de la serie temporal. ¿Parece la media del conteo constante a lo largo del tiempo?
b) Ajusta un modelo de regresión de Poisson simple donde `discoveries` es la respuesta y el tiempo (`time(discoveries)`) es el predictor.
c) Interpreta el coeficiente del tiempo. (Pista: recuerda exponenciarlo para obtener el Incidence Rate Ratio - IRR).

```r
# Usar dataset discoveries
data(discoveries)
```

a) Gráfico de la serie temporal:

```r
plot(discoveries, main = "Grandes Inventos por Año",
     ylab = "Número de Descubrimientos", xlab = "Año")
```



b) Ajustar modelo de Poisson:

```r
# Crear data frame con tiempo
df_discoveries <- data.frame(
  count = as.numeric(discoveries),
  time = as.numeric(time(discoveries))
)

modelo_poisson <- glm(count ~ time, data = df_discoveries, family = poisson)
summary(modelo_poisson)
```

```
Call:
glm(formula = count ~ time, family = poisson, data = df_discoveries)
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 11.354807   3.775677   3.007  0.00264 **
time        -0.005360   0.001982  -2.705  0.00683 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 164.68  on 99  degrees of freedom
Residual deviance: 157.32  on 98  degrees of freedom
AIC: 430.32

Number of Fisher Scoring iterations: 5
```

c) Interpretación del coeficiente de tiempo:

```r
# IRR (Incidence Rate Ratio)
coef_time <- coef(modelo_poisson)["time"]
irr <- exp(coef_time)
print(paste("IRR para tiempo:", round(irr, 6)))
```

```
[1] "IRR para tiempo: 0.994654"
```

El IRR para el tiempo es **0.994654**. Esto significa que por cada año que pasa, se espera que la tasa de grandes inventos se **multiplique por 0.994654**, lo que representa una **disminución anual de aproximadamente 0.53%** (calculado como 1 - 0.994654).

Ejercicio 8: Diagnóstico (Sobredispersión)

a) Para el modelo de Poisson del ejercicio anterior, calcula el **estadístico de dispersión ($\hat{\phi}$)**. (Pista: $\hat{\phi} = \frac{\sum r_i^2}{n-p}$, donde los r_i son los residuos Pearson).
b) Basándote en el valor de $\hat{\phi}$, ¿hay evidencia de sobredispersión?
c) Si encuentras sobredispersión, ¿cuál es el modelo alternativo que proponen los apuntes? ¿Qué ventaja teórica ofrece este modelo alternativo?

a) Calcular estadístico de dispersión:

```r
residuos_pearson <- residuals(modelo_poisson, type = "pearson")
n <- nrow(df_discoveries)
p <- length(coef(modelo_poisson))
phi_hat <- sum(residuos_pearson^2) / (n - p)

print(paste("Estadístico de dispersión (^):", round(phi_hat, 3)))
```

```
[1] "Estadístico de dispersión (^): 1.541"
```

b) Evidencia de sobredispersión: Si $\hat{} > 1$, hay evidencia de sobredispersión. Valores > 1.5 indican sobredispersión considerable.

c) Modelo alternativo: Si hay sobredispersión, se puede usar **regresión Binomial Negativa**, que incluye un parámetro adicional que permite que la varianza sea mayor que la media: $Var[Y] = \mu + \alpha\mu^2$.

```r
# Ajuste con binomial negativa si hay sobredispersión
if(phi_hat > 1.5) {
  library(MASS)
  modelo_nb <- glm.nb(count ~ time, data = df_discoveries)
  print("Modelo Binomial Negativo ajustado:")
  print(summary(modelo_nb))
}
```

```
[1] "Modelo Binomial Negativo ajustado:"

Call:
glm.nb(formula = count ~ time, data = df_discoveries, init.theta = 6.214857583,
    link = log)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 12.254546   4.628240   2.648   0.0081 **
time        -0.005832   0.002428  -2.402   0.0163 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(6.2149) family taken to be 1)

    Null deviance: 114.04  on 99  degrees of freedom
Residual deviance: 108.69  on 98  degrees of freedom
```

```
AIC: 422.34

Number of Fisher Scoring iterations: 1

              Theta:  6.21
          Std. Err.:  2.67

 2 x log-likelihood:  -416.34
```

Ejercicio 9: Conceptual (Deviance)

La **deviance** es la medida principal de bondad de ajuste en los GLM. Explica conceptualmente qué mide. ¿Cómo se utiliza la diferencia en deviance entre dos modelos anidados para decidir cuál es mejor?

Deviance mide qué tan bien el modelo ajustado se compara con el modelo saturado (perfecto). Es análogo a la suma de cuadrados residuales en regresión lineal.

Fórmula: $D = 2[L(\text{modelo saturado}) - L(\text{modelo ajustado})]$

Uso para comparar modelos anidados: La diferencia en deviance entre dos modelos anidados sigue una distribución ^{2} con grados de libertad igual a la diferencia en número de parámetros. Si esta diferencia es significativa, el modelo más complejo es preferible.

Ejercicio 10: Elección del Modelo Adecuado

Para cada uno de los siguientes escenarios, indica qué tipo de GLM (Logístico, Poisson, Binomial Negativo, Gamma…) sería el más apropiado y por qué.

a) Quieres modelar el **tiempo (en minutos)** que tarda un cliente en resolver una consulta en un centro de atención telefónica. El tiempo es siempre positivo y muchos valores se agrupan en tiempos cortos, con una cola larga de tiempos muy largos.
b) Quieres predecir la **presencia o ausencia** de una especie de planta en diferentes parcelas de un bosque.
c) Quieres modelar el **número de visitas** que cada usuario hace a una página web en un mes. Observas que la varianza del número de visitas es mucho mayor que la media.

a) Tiempo de resolución de consultas: Modelo Gamma sería más apropiado porque:

- La variable es continua y positiva
- Típicamente tiene distribución asimétrica con cola derecha larga
- La distribución Gamma es flexible para este tipo de datos

b) Presencia/ausencia de especies: Regresión Logística es la elección obvia porque:

- Variable respuesta binaria (presencia = 1, ausencia = 0)
- Queremos modelar probabilidades que están restringidas al intervalo [0,1]

c) Número de visitas con varianza mayor que la media: Regresión Binomial Negativa sería más apropiada porque:

- Variable de conteo (número de visitas)
- La sobredispersión (varianza > media) viola el supuesto de Poisson
- La binomial negativa maneja naturalmente la sobredispersión

Ejercicios Avanzados

Ejercicio 1: Derivación de Estimadores

Considera el modelo de regresión lineal simple $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$. Partiendo de la función objetivo de Mínimos Cuadrados Ordinarios (MCO), $S(\beta_0, \beta_1) = \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$, realiza la derivación matemática completa para obtener las expresiones de los estimadores $\hat{\beta}_0$ y $\hat{\beta}_1$. Muestra todos los pasos, desde el cálculo de las derivadas parciales hasta la resolución de las ecuaciones normales.

El objetivo es encontrar los valores de β_0 y β_1 que minimizan la Suma de Cuadrados del Error (SSE). Para ello, calculamos las derivadas parciales de la función $S(\beta_0, \beta_1)$ con respecto a cada parámetro y las igualamos a cero.

1. **Derivada parcial con respecto a β_0:**

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

   Igualando a cero y dividiendo por -2:

$$\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) = 0 \implies \sum y_i - n\hat{\beta}_0 - \hat{\beta}_1 \sum x_i = 0$$

   Reordenando, obtenemos la primera **ecuación normal**:

$$n\hat{\beta}_0 + \hat{\beta}_1 \sum x_i = \sum y_i \quad (1)$$

2. **Derivada parcial con respecto a β_1:**

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

   Igualando a cero y dividiendo por -2:

$$\sum_{i=1}^{n} x_i(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) = 0 \implies \sum x_i y_i - \hat{\beta}_0 \sum x_i - \hat{\beta}_1 \sum x_i^2 = 0$$

   Reordenando, obtenemos la segunda **ecuación normal**:

$$\hat{\beta}_0 \sum x_i + \hat{\beta}_1 \sum x_i^2 = \sum x_i y_i \quad (2)$$

3. **Resolución del sistema:** De la ecuación (1), dividiendo por n, podemos despejar $\hat{\beta}_0$:

$$\hat{\beta}_0 + \hat{\beta}_1 \bar{x} = \bar{y} \implies \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Esta es la fórmula para el intercepto, que depende de la pendiente.

Sustituimos esta expresión de $\hat{\beta}_0$ en la ecuación (2):

$$(\bar{y} - \hat{\beta}_1 \bar{x}) \sum x_i + \hat{\beta}_1 \sum x_i^2 = \sum x_i y_i$$

$$\bar{y} \sum x_i - \hat{\beta}_1 \bar{x} \sum x_i + \hat{\beta}_1 \sum x_i^2 = \sum x_i y_i$$

Agrupamos los términos con $\hat{\beta}_1$:

$$\hat{\beta}_1 (\sum x_i^2 - \bar{x} \sum x_i) = \sum x_i y_i - \bar{y} \sum x_i$$

Sabiendo que $\sum x_i = n\bar{x}$ y $\sum y_i = n\bar{y}$:

$$\hat{\beta}_1 (\sum x_i^2 - n\bar{x}^2) = \sum x_i y_i - n\bar{x}\bar{y}$$

Las expresiones entre paréntesis son las fórmulas de la suma de cuadrados de X (S_{xx}) y la suma de productos cruzados de X e Y (S_{xy}):

$$\hat{\beta}_1 S_{xx} = S_{xy}$$

Finalmente, despejamos el estimador de la pendiente:

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

Estas son las expresiones para los estimadores de MCO.

Ejercicio 2: El Impacto de la Multicolinealidad

En un modelo de regresión múltiple con dos predictores estandarizados (X_1, X_2), la varianza del estimador $\hat{\beta}_1$ viene dada por $\text{Var}(\hat{\beta}_1) = \frac{\sigma^2}{n(1-r_{12}^2)}$, donde r_{12} es la correlación entre X_1 y X_2.

a) Explica matemáticamente qué le ocurre a la varianza de $\hat{\beta}_1$ cuando la correlación entre los predictores (r_{12}) se aproxima a 1 (multicolinealidad perfecta).
b) Relaciona esta fórmula con la del Factor de Inflación de la Varianza (VIF). ¿Cómo demuestra esta expresión que la multicolinealidad "infla" la varianza de los estimadores de los coeficientes?

a) Efecto de la correlación en la varianza: La varianza del estimador, $\text{Var}(\hat{\beta}_1)$, es una medida de su imprecisión. La fórmula $\frac{\sigma^2}{n(1-r_{12}^2)}$ muestra que la varianza depende inversamente del término $(1-r_{12}^2)$. - Si $r_{12} = 0$ (no hay correlación), la varianza es mínima: $\text{Var}(\hat{\beta}_1) = \sigma^2/n$. - A medida que la correlación $|r_{12}|$ aumenta y se acerca a 1 (multicolinealidad perfecta), el término r_{12}^2 también se acerca a 1. - Consecuentemente, el denominador $(1 - r_{12}^2)$ se aproxima a 0. - Matemáticamente, cuando el denominador de una fracción tiende a cero, el valor de la fracción tiende a infinito. Por lo tanto:

$$\lim_{|r_{12}|\to 1} \text{Var}(\hat{\beta}_1) = \lim_{|r_{12}|\to 1} \frac{\sigma^2}{n(1-r_{12}^2)} = \infty$$

Esto significa que con multicolinealidad severa, la varianza de los estimadores de los coeficientes "explota", volviéndolos extremadamente inestables y poco fiables.

b) Relación con el Factor de Inflación de la Varianza (VIF): El VIF para un predictor X_j se define como $VIF_j = \frac{1}{1-R_j^2}$, donde R_j^2 es el R-cuadrado de la regresión de X_j sobre todos los demás predictores. En el caso de solo dos predictores (X_1, X_2), el R^2 de la regresión de X_1 sobre X_2 es simplemente el cuadrado de su coeficiente de correlación, es decir, $R_1^2 = r_{12}^2$. Sustituyendo esto en la fórmula del VIF, tenemos:

$$VIF_1 = \frac{1}{1-r_{12}^2}$$

Ahora podemos reescribir la fórmula de la varianza de $\hat{\beta}_1$ usando el VIF:

$$\text{Var}(\hat{\beta}_1) = \frac{\sigma^2}{n} \cdot \frac{1}{1-r_{12}^2} = \frac{\sigma^2}{n} \cdot VIF_1$$

Esta expresión demuestra que el VIF es, literalmente, el **factor multiplicativo** por el cual la varianza del estimador del coeficiente se "infla" en comparación con el caso base en el que no habría correlación (donde VIF = 1).

Ejercicio 3: Interpretación de Coeficientes en Modelos Transformados

Considera un modelo de regresión **log-log**: $\log(Y_i) = \beta_0 + \beta_1 \log(X_i) + \varepsilon_i$. Demuestra matemáticamente que el coeficiente β_1 puede interpretarse como una **elasticidad**, es decir, el cambio porcentual en Y ante un cambio del 1% en X. (Pista: utiliza la derivada de $\log(Y)$ con respecto a $\log(X)$).

La elasticidad de Y con respecto a X se define como el cambio porcentual en Y para un cambio del 1% en X. Para cambios infinitesimales, esta se expresa como:

$$\eta = \frac{\%\Delta Y}{\%\Delta X} = \frac{dY/Y}{dX/X}$$

Una propiedad matemática de los logaritmos es que para cambios pequeños, $d(\log(z)) \approx \frac{dz}{z}$, que representa un cambio relativo o porcentual. Por lo tanto, la elasticidad puede expresarse como la derivada del logaritmo de Y con respecto al logaritmo de X:

$$\eta = \frac{d(\log Y)}{d(\log X)}$$

Partiendo de nuestro modelo poblacional (ignorando el término de error para analizar la relación sistemática):

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

Ahora, simplemente calculamos la derivada de la ecuación con respecto a $\log(X)$:

$$\frac{d(\log Y)}{d(\log X)} = \frac{d}{d(\log X)}(\beta_0 + \beta_1 \log(X))$$

El término β_0 es una constante, por lo que su derivada es 0. El término $\beta_1 \log(X)$ tiene una derivada de β_1 con respecto a $\log(X)$. Por lo tanto:

$$\frac{d(\log Y)}{d(\log X)} = \beta_1$$

Hemos demostrado que el coeficiente β_1 es igual a la elasticidad de Y con respecto a X. Así, β_1 representa el cambio porcentual promedio en Y que se asocia con un aumento del 1% en X.

Ejercicio 4: Fundamentos de la Regularización

Explica desde una perspectiva geométrica por qué la regularización **Lasso (penalización L1)** es capaz de reducir los coeficientes exactamente a cero, realizando así selección de variables, mientras que la regularización **Ridge (penalización L2)** solo puede encoger los coeficientes hacia cero sin anularlos por completo. Apoya tu explicación con un dibujo o descripción de las "regiones de restricción" de ambos métodos en un espacio de dos coeficientes (β_1, β_2).

La estimación en regresión regularizada puede entenderse como un problema de optimización restringida. El objetivo es encontrar el conjunto de coeficientes $(\beta_1, \beta_2, ...)$ que minimice la Suma de Cuadrados del Error (SSE), sujeto a una restricción en el tamaño de dichos coeficientes.

- **Geometría del problema:** El conjunto de todos los posibles valores de los coeficientes para un mismo valor de SSE forma una elipse (en un espacio de dos coeficientes, β_1, β_2) centrada en la solución de Mínimos Cuadrados Ordinarios (MCO). El objetivo es encontrar la elipse más pequeña posible que toque la "región de restricción".
- **Regresión Ridge (Penalización L2):** La restricción es $\sum \beta_j^2 \leq s$. En dos dimensiones, $\beta_1^2 + \beta_2^2 \leq s$ es la ecuación de un **círculo**. Esta región es convexa y no tiene "esquinas". Cuando las elipses del SSE se expanden desde el punto MCO, el primer punto de contacto

con el círculo será un punto de tangencia. Debido a la forma suave y redondeada del círculo, es extremadamente improbable que este punto de tangencia ocurra exactamente sobre un eje (donde uno de los coeficientes sería cero). Por lo tanto, Ridge reduce la magnitud de ambos coeficientes, pero no los anula.

- **Regresión Lasso (Penalización L1):** La restricción es $\sum |\beta_j| \leq s$. En dos dimensiones, $|\beta_1| + |\beta_2| \leq s$ es la ecuación de un **rombo (o diamante)**, rotado 45 grados. La característica clave de esta región son sus **vértices afilados, que se encuentran sobre los ejes**. Cuando las elipses del SSE se expanden, es mucho más probable que toquen la región de restricción en uno de estos vértices que en una de las aristas. Si el punto de contacto es un vértice sobre un eje (por ejemplo, el punto $(0, \beta_2)$), significa que el otro coeficiente (β_1) es **exactamente cero**. Es esta propiedad geométrica, las "esquinas" de la región de penalización L1, lo que induce la escasez (*sparsity*) y permite a Lasso realizar selección de variables.

Ejercicio 5: La Familia Exponencial y los GLM

La teoría de los Modelos Lineales Generalizados (GLM) se basa en que distribuciones como la Normal, Binomial o Poisson pertenecen a la **familia exponencial**. La forma canónica de esta familia establece una relación directa entre la media y la varianza a través de la **función de varianza** $V(\mu)$. Explica cuál es la función de varianza para un modelo de **Poisson** y para un modelo **Binomial**. ¿Qué implicaciones tiene la forma de $V(\mu)$ en cada caso sobre el comportamiento de los datos y los supuestos del modelo?

La **función de varianza** $V(\mu)$ es la "firma" de cada distribución dentro de la familia exponencial, ya que define la relación teórica entre la media μ y la varianza de la variable respuesta.

- **Modelo de Poisson:**
  - **Función de Varianza:** $V(\mu) = \mu$.
  - **Implicación:** Esto implica que la varianza de la variable respuesta es teóricamente igual a su media: $\text{Var}(Y) = \mu$. Este supuesto se conoce como **equidispersión**. La implicación más importante para el modelado es que, si los datos reales muestran una varianza significativamente mayor que la media (un fenómeno muy común llamado **sobredispersión**), el modelo de Poisson será inadecuado. Los errores estándar de los coeficientes estarán subestimados, llevando a p-valores incorrectamente bajos y a una inferencia errónea.
- **Modelo Binomial:**
  - **Función de Varianza:** $V(\mu) = \mu(1 - \mu)$.

– **Implicación:** En este caso, la varianza no es constante, sino que es una función cuadrática de la media (la probabilidad de éxito). La varianza es mínima cuando μ se acerca a 0 o 1, y es máxima cuando $\mu = 0.5$. Esta es la heterocedasticidad inherente a los datos de proporciones. El modelo GLM maneja esto de forma natural a través del algoritmo de estimación (IRLS), que da más peso a las observaciones con menor varianza (aquellas con probabilidades predichas cercanas a 0 o 1) y menos peso a las más inciertas (aquellas con probabilidades cercanas a 0.5).

Ejercicio 6: El Problema de la Inferencia en Métodos Stepwise

Los apuntes advierten que los p-valores de un modelo final obtenido mediante selección por pasos (stepwise) están **sesgados y son excesivamente optimistas**. Explica el razonamiento estadístico detrás de esta advertencia. ¿Por qué el proceso iterativo de "buscar y seleccionar" la variable más significativa en cada paso invalida los supuestos teóricos del test t estándar?

La advertencia se debe a que los métodos stepwise violan un principio fundamental de la prueba de hipótesis: el modelo y las hipótesis deben ser especificados **a priori**, antes de examinar las relaciones en los datos. Los métodos stepwise hacen exactamente lo contrario.

1. **Problema de Múltiples Comparaciones:** En cada paso, un algoritmo como la selección *forward* realiza múltiples tests (un test t para cada variable candidata a entrar) y selecciona la variable "ganadora", que es la que tiene el p-valor más pequeño. Al elegir el valor mínimo de un conjunto de pruebas, estamos seleccionando un valor extremo de la distribución de p-valores bajo la hipótesis nula. El p-valor reportado para esa variable (p. ej., 0.03) no refleja la probabilidad de observar un resultado tan extremo en un solo intento, sino la probabilidad de que *el mejor de varios intentos* sea tan extremo, lo cual es una probabilidad mucho mayor.
2. **Invalidez de la Distribución Teórica:** El p-valor de un test t se calcula asumiendo que el coeficiente sigue una distribución t de Student. Sin embargo, el coeficiente de una variable seleccionada por un algoritmo stepwise no sigue esta distribución. Sigue una distribución más compleja (una "distribución de un estadístico de orden"), porque ha sido seleccionado condicionalmente por ser el mejor.
3. **Sesgo de Selección:** El proceso está diseñado para encontrar relaciones, incluso en datos puramente aleatorios. Si tenemos muchas variables de ruido, la probabilidad de que una de ellas parezca significativa por puro azar es alta. El método stepwise seleccionará esa variable y reportará un p-valor bajo y engañoso.

En resumen, los p-valores de un modelo stepwise están **sesgados a la baja (son demasiado pequeños)** porque no tienen en cuenta el proceso de búsqueda y selección que los ha producido. Esto lleva a una **inflación de la tasa de error de Tipo I**, haciendo que concluyamos que ciertas variables son significativas cuando en realidad no lo son.

Ejercicio 7: Propiedades de los Estimadores MCO

El **Teorema de Gauss-Markov** establece que, bajo ciertos supuestos, los estimadores de Mínimos Cuadrados Ordinarios (MCO) son **MELI (Mejores Estimadores Lineales Insesgados)**. Demuestra la propiedad de **insesgadez** para el estimador $\widehat{\beta}$ en notación matricial. Es decir, demuestra que $E[\widehat{\beta}] = \beta$. Muestra todos los pasos y menciona qué supuestos del modelo estás utilizando en cada paso.

1. Comenzamos con la fórmula del estimador MCO en notación matricial:

$$\widehat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

2. Sustituimos el modelo poblacional verdadero para el vector $\mathbf{y}$, que es $\mathbf{y} = \mathbf{X}\beta + \varepsilon$:

$$\widehat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\beta + \varepsilon)$$

3. Aplicamos el operador de valor esperado $E[\cdot]$ a ambos lados. Tratamos la matriz de diseño $\mathbf{X}$ como fija (no aleatoria):

$$E[\widehat{\beta}] = E[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\beta + \varepsilon)]$$

4. Distribuimos el término $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ dentro del paréntesis:

$$E[\widehat{\beta}] = E[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\beta + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\varepsilon]$$

5. Usamos la propiedad de linealidad del valor esperado ($E[A + B] = E[A] + E[B]$):

$$E[\widehat{\beta}] = E[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\beta] + E[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\varepsilon]$$

6. Analizamos cada término por separado:

   - En el primer término, todo es constante excepto el operador de valor esperado, y $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}$ es la matriz identidad $\mathbf{I}$. Por lo tanto, $E[\mathbf{I}\beta] = \beta$.
   - En el segundo término, podemos sacar las constantes del valor esperado: $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T E[\varepsilon]$.

7. Aplicamos el supuesto de **exogeneidad** (o media del error nula), que establece que el valor esperado del término de error es cero: $E[\varepsilon] = \mathbf{0}$.

$$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T E[\varepsilon] = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{0} = \mathbf{0}$$

8. Uniendo los resultados, concluimos:

$$E[\widehat{\beta}] = \beta + \mathbf{0} = \beta$$

   Esto demuestra que el estimador MCO $\widehat{\beta}$ es insesgado, ya que su valor esperado es el verdadero parámetro poblacional β.

Ejercicio 8: Intervalos de Confianza vs. Predicción

La fórmula para el intervalo de predicción para una nueva observación en regresión lineal simple es:

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

Explica el origen y el significado de cada uno de los **tres términos** que se encuentran dentro del paréntesis bajo la raíz cuadrada. ¿Qué fuente de incertidumbre representa cada término y por qué la suma de los tres es necesaria para un intervalo de predicción?

La fórmula cuantifica la incertidumbre total de predecir una *única* nueva observación. Esta incertidumbre proviene de dos fuentes: la incertidumbre sobre la posición de la verdadera línea de regresión y la variabilidad inherente de un punto individual alrededor de esa línea. Los tres términos dentro del paréntesis representan estas fuentes de varianza (escaladas por MSE, que estima σ^2):

1. **Término `1`**: Esta es la componente más importante y la que distingue al intervalo de predicción. Representa la **varianza del error aleatorio de la nueva observación**, $\text{Var}(\varepsilon_0) = \sigma^2$. Es la incertidumbre irreducible o inherente de un solo punto, que siempre se desviará de la media. Esta es la razón principal por la que un intervalo de predicción es siempre más ancho que uno de confianza.
2. **Término `1/n`**: Esta componente está relacionada con la **incertidumbre en la estimación del intercepto** $\widehat{\beta}_0$. Representa la incertidumbre sobre la "altura" general de la línea de regresión. A medida que el tamaño de la muestra (n) aumenta, nuestra confianza en la posición de la línea mejora, y este término de incertidumbre se hace más pequeño.
3. **Término `(x_0 - \bar{x})^2 / S_{xx}`**: Esta componente representa la **incertidumbre debida a la estimación de la pendiente** $\widehat{\beta}_1$. La incertidumbre en la pendiente tiene un mayor impacto cuanto más nos alejamos del centro de los datos ($\bar{x}$). Si predecimos en el punto medio de nuestros datos ($x_0 = \bar{x}$), este término se anula. A medida que x_0 se aleja de $\bar{x}$, el efecto de un pequeño error en la estimación de la pendiente se magnifica, ensanchando el intervalo.

En resumen, los términos `1/n` y `(x_0 - \bar{x})^2 / S_{xx}` juntos cuantifican la incertidumbre sobre dónde está la **línea de regresión verdadera** (lo que cubre el intervalo de confianza). El término `1` añade la incertidumbre de **un nuevo punto individual** alrededor de esa línea.

Ejercicio 9: Estimación por Máxima Verosimilitud

Para un modelo de regresión logística, la función de log-verosimilitud es:

$$\ell(\beta) = \sum_{i=1}^{n} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

donde $p_i = \frac{1}{1+e^{-\mathbf{x}_i^T\beta}}$. Deriva la **ecuación de puntuación (score equation)** para un coeficiente β_j (es decir, calcula $\frac{\partial \ell}{\partial \beta_j}$) y demuestra que se iguala a cero cuando $\sum_{i=1}^{n} x_{ij}(y_i - p_i) = 0$. Interpreta el significado de esta condición final.

1. La función de log-verosimilitud para la regresión logística es:

$$\ell(\beta) = \sum_{i=1}^{n} \left[y_i \mathbf{x}_i^T \beta - \log(1 + e^{\mathbf{x}_i^T\beta}) \right]$$

2. La ecuación de puntuación (*score equation*) se obtiene al calcular la primera derivada de la log-verosimilitud con respecto a un parámetro, en este caso β_j. Debemos calcular $\frac{\partial \ell}{\partial \beta_j}$ y igualarla a cero.

3. La derivada de una suma es la suma de las derivadas, por lo que podemos analizar el término dentro del sumatorio para una observación i:

$$\frac{\partial}{\partial \beta_j} \left[y_i \mathbf{x}_i^T \beta - \log(1 + e^{\mathbf{x}_i^T\beta}) \right]$$

4. La derivada del primer término, $y_i \mathbf{x}_i^T \beta = y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_j x_{ij} + \dots)$, con respecto a β_j es simplemente $y_i x_{ij}$.

5. La derivada del segundo término, $\log(1 + e^{\mathbf{x}_i^T\beta})$, requiere la regla de la cadena. Sea $u = 1 + e^{\mathbf{x}_i^T\beta}$.

$$\frac{\partial}{\partial \beta_j} \log(u) = \frac{1}{u} \cdot \frac{\partial u}{\partial \beta_j} = \frac{1}{1 + e^{\mathbf{x}_i^T\beta}} \cdot \frac{\partial}{\partial \beta_j}(1 + e^{\mathbf{x}_i^T\beta})$$

$$= \frac{1}{1 + e^{\mathbf{x}_i^T\beta}} \cdot (e^{\mathbf{x}_i^T\beta} \cdot x_{ij}) = \left(\frac{e^{\mathbf{x}_i^T\beta}}{1 + e^{\mathbf{x}_i^T\beta}} \right) x_{ij}$$

6. Reconocemos que el término entre paréntesis es la definición de la probabilidad p_i en el modelo logístico ($p_i = \frac{1}{1+e^{-\mathbf{x}_i^T\beta}}$). Por lo tanto, la derivada del segundo término es $p_i x_{ij}$.

7. Uniendo ambos resultados, la derivada para la observación i es $y_i x_{ij} - p_i x_{ij}$.

8. La ecuación de puntuación completa es la suma sobre todas las observaciones:

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{n} (y_i x_{ij} - p_i x_{ij}) = \sum_{i=1}^{n} x_{ij}(y_i - p_i)$$

9. Los estimadores de máxima verosimilitud se encuentran al igualar esta ecuación a cero:

$$\sum_{i=1}^{n} x_{ij}(y_i - p_i) = 0$$

Interpretación: Esta condición final significa que los estimadores de máxima verosimilitud se encuentran cuando los residuos del modelo ($y_i - p_i$, la diferencia entre lo observado y la probabilidad predicha) son **ortogonales** (no están correlacionados) a los predictores x_{ij}. Esto es análogo a las ecuaciones normales de MCO y significa que el modelo ha extraído toda la información linealmente asociable a los predictores, no quedando ningún patrón relacionado con ellos en los errores.

Ejercicio 10: El Coeficiente de Regresión Parcial

El texto afirma que el coeficiente $\hat{\beta}_j$ de una regresión múltiple puede entenderse como el coeficiente de una regresión simple entre dos conjuntos de residuos. Explica con detalle este concepto de **regresión parcial**. ¿Qué se está "parcializando" o "eliminando" de la variable respuesta Y y del predictor X_j antes de calcular su relación? ¿Por qué este concepto es fundamental para entender la interpretación *ceteris paribus*?

El concepto de regresión parcial es fundamental para entender la interpretación *ceteris paribus* de un coeficiente en regresión múltiple. Afirma que el coeficiente $\hat{\beta}_j$ del predictor X_j en un modelo múltiple es matemáticamente idéntico a la pendiente de una regresión simple entre dos conjuntos de residuos.

El proceso de "parcialización" consiste en eliminar la influencia de todos los demás predictores (denotados como X_{-j}) tanto de la variable respuesta Y como del predictor de interés X_j.

1. **Parcialización de Y:** Se ajusta un modelo de regresión de Y en función de todos los demás predictores: $Y \sim X_{-j}$. Los residuos de este modelo, $e_{Y|X_{-j}}$, representan la parte de la variabilidad de Y que **no puede ser explicada** por el resto de variables del modelo. Es la "información única" de Y.
2. **Parcialización de X_j:** Se ajusta un modelo de regresión de X_j en función de todos los demás predictores: $X_j \sim X_{-j}$. Los residuos de este modelo, $e_{X_j|X_{-j}}$, representan la parte de la variabilidad de X_j que es **única** y no está correlacionada con el resto de variables. Es la "información única" que X_j aporta.
3. **Relación entre los residuos:** Si ahora ajustamos una regresión lineal simple entre estos dos conjuntos de residuos:

   $$e_{Y|X_{-j}} \sim e_{X_j|X_{-j}}$$

   La pendiente de esta regresión simple es **exactamente igual** al coeficiente de regresión múltiple $\hat{\beta}_j$ del modelo original completo.

Conclusión: Esto demuestra que $\hat{\beta}_j$ no mide la relación "bruta" entre Y y X_j, sino la relación entre la parte de Y que no es explicada por los otros predictores y la parte de X_j que es única. Es la asociación "limpia" entre Y y X_j después de haber controlado estadísticamente por la influencia de todas las demás variables en el modelo. Esto es, precisamente, la formalización matemática del principio *ceteris paribus*.