# MÓDULO 1. INTRODUCCIÓN AL ANÁLISIS MULTIVARIANTE

# INSTRUCCIONES IMPORTANTES

---

## PASO 1: INSTALACIÓN Y PREPARACIÓN DEL ENTORNO

### 1.1 Instalar las librerías necesarias (solo la primera vez)

r

```r
# Instalar librerías (ejecutar solo una vez)
install.packages("ggplot2")
install.packages("dplyr")
install.packages("gridExtra")
install.packages("lmtest")
install.packages("car")
install.packages("readr")
```

**NOTA:** Solo necesitas ejecutar este código una vez. Si ya tienes las librerías instaladas, puedes omitir este paso.

### 1.2 Cargar las librerías necesarias

r

```r
# Cargar librerías necesarias (ejecutar en cada sesión)
library(ggplot2)    # Para gráficos avanzados
library(dplyr)      # Para manipulación de datos
library(gridExtra)  # Para combinar gráficos
```

---

## PASO 2: CARGA Y VERIFICACIÓN DE DATOS

Base de Datos: ChildCarSeats_clean.csv.

Las variables del conjunto de datos son:

• **Sales**: ventas unitarias, en miles, en cada ubicación.
• **CompPrice**: precio que cobra la competencia en cada ubicación.
• **Income**: nivel de ingresos comunitarios, en miles de dólares.
• **Advertising**: presupuesto de publicidad local de la empresa en cada ubicación, en miles de dólares.
• **Population**: tamaño de la población en la región, en miles.
• **Price**: precio de las sillitas de coche en cada ubicación.
• **ShelveLoc**: variable categórica con los niveles Bad, Good y Medium, que indica la calidad de la ubicación de las sillas de coche en la tienda.
• **Age**: edad media de la población local.
• **Education**: valor numérico que indica la media del nivel de educación (años de educación) de la población.
• **Urban**: variable binaria con los niveles Yes y No para indicar si la tienda se encuentra en una ubicación urbana o rural.
• **US**: variable binaria con los niveles Yes y No para indicar si la tienda se encuentra en EUA o no.

### 2.1 Cargar el dataset

#### *Opción A: Si el archivo es CSV (.csv)*

r

```r
# Usando read.csv() (Base R)
datos <- read.csv("mi_archivo.csv", header=TRUE, stringsAsFactors = TRUE)

# Alternativa con tidyverse
# library(readr)
# datos <- read_csv("mi_archivo.csv")
```

#### *Opción B: Si el archivo es RData (.RData o .rda)*

r

```r
# Cargar archivo RData
load("mi_archivo.RData")
```

```
# Verificar qué objetos se han cargado
ls()

# Si el objeto tiene un nombre diferente, asignarlo a 'datos'
# datos <- nombre_del_objeto_cargado
```

### 2.2 Verificar la estructura de los datos

r

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

---

## PASO 3: CLASIFICACIÓN DE VARIABLES

### 3.1 Separar variables por tipo

r

```
# Variables cuantitativas (numéricas)
variables_numericas <- data.frame(datos$variable_1, datos$variable_2, datos$variable_3,  datos$variable_4)

# Variables cualitativas
variables_cualitativas <- data.frame(datos$var_cualitativa_1, datos$var_cualitativa_2)

# Verificar la clasificación
print(variables_numericas)
print(cualitativasvariables_cualitativas)
```

---

## PASO 4: ANÁLISIS DESCRIPTIVO - VARIABLES CUANTITATIVAS

### 4.1 Medidas de tendencia central

r

```
# Método rápido para una variable
mean(datos$variable_interes)     # Media de la variable de interés
median(datos$variable_interes)   # Mediana de la variable de interés

# Para múltiples variables
summary(variables_numericas)     # Resumen completo de todas las variables numéricas
```

## 4.2 Medidas de dispersión

r

```r
# Desviación estándar y rango intercuartílico
sd(datos$variable_interes)      # Desviación estándar
IQR(datos$variable_interes)     # Rango intercuartílico

# Para todas las variables numéricas
sapply(variables_numericas, sd, na.rm = TRUE)   # Desviaciones estándar
sapply(variables_numericas, IQR, na.rm = TRUE)  # Rangos intercuartílicos
```

## 4.3 Medidas de posición

r

```r
# Cuartiles de la variable de interés
quantile(datos$variable_interes)

# Resumen completo de la variable de interés
summary(datos$variable_interes)
```

---

# PASO 5: VISUALIZACIÓN - VARIABLES CUANTITATIVAS

## 5.1 Gráficos individuales

r

```r
# Boxplot para visualizar distribución y outliers
ggplot(datos, aes(y=variable_interes)) + geom_boxplot() + ggtitle("Distribución de Variable de Interés")

# Histograma para ver la forma de la distribución
ggplot(datos, aes(x=variable_interes)) + geom_histogram(bins=10, fill="lightblue", color="black") +
 ggtitle("Histograma de Variable de Interés")

# Scatter plot para relación entre dos variables
ggplot(datos, aes(x=variable_1, y=variable_2)) +
 geom_point() +
 ggtitle("Relación Variable_1 vs Variable_2")
```

## 5.2 Combinar gráficos

r

```r
# Crear gráficos por separado
g1 <- ggplot(datos, aes(y=variable_interes)) +
    geom_boxplot() +
    ggtitle("Boxplot Variable de Interés")

g2 <- ggplot(datos, aes(x=variable_2, y=variable_interes)) +
    geom_point() +
    ggtitle("Variable_2 vs Variable de Interés")

# Combinar en una sola visualización
grid.arrange(g1, g2, nrow=1)  # En una fila
```

---

# PASO 6: ANÁLISIS DESCRIPTIVO - VARIABLES CUALITATIVAS

## 6.1 Tablas de frecuencia

r

```r
# Frecuencias absolutas
table(datos$var_cualitativa)

# Frecuencias relativas (porcentajes)
prop.table(table(datos$var_cualitativa)) * 100

# Resumen
summary(datos$var_cualitativa)
```

## 6.2 Gráficos para variables cualitativas

r

```r
# Preparar datos para gráficos
resumen_cualitativa <- summarize(group_by(datos, var_cualitativa),
                                 n=length(var_cualitativa))



# Gráfico de barras
ggplot(resumen_cualitativa, aes(x=var_cualitativa, y=n, fill=var_cualitativa)) +
  geom_bar(stat="identity") +
  ggtitle("Frecuencia por Categoría")

# Gráfico circular
ggplot(resumen_cualitativa, aes(x="", y=n, fill=var_cualitativa)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start=0) +
  ggtitle("Distribución por Categoría")
```

---

# PASO 7: ANÁLISIS BIVARIANTE - CORRELACIONES

## 7.1 Correlación entre variables numéricas

r

```r
# Correlación entre dos variables específicas
cor(datos$variable_1, datos$variable_2)

# Matriz de correlaciones
matriz_correlacion <- cor(variables_numericas, use = "complete.obs")
print(round(matriz_correlacion, 3))
```

---

# PASO 8: ANÁLISIS BIVARIANTE - VARIABLES CATEGÓRICAS

## 8.1 Tablas de contingencia y test de independencia

r

```r
# Crear tabla de contingencia
tabla_contingencia <- table(datos$var_categorica_1, datos$var_categorica_2)
print("Tabla de contingencia:")
print(tabla_contingencia)

# Test chi-cuadrado de independencia
test_chi <- chisq.test(tabla_contingencia)
print(test_chi)
```

---

# PASO 9: REGRESIÓN LINEAL SIMPLE

## 9.1 Modelo de regresión (Variable numérica como predictora)

r

```r
# Crear modelo: predecir variable dependiente usando variable numérica
modelo_1 <- lm(variable_dependiente ~ var_numerica_1, data = datos)

# Ver resultados del modelo
summary(modelo_1)
```

## 9.2 Diagnóstico del modelo

r

```r
# Calcular residuos y valores ajustados
residuos <- rstandard(modelo_1)
valores_ajustados <- fitted(modelo_1)

# Configurar ventana de gráficos
par(mfrow = c(1,2))
```

```r
# Gráfico 1: Residuos vs Valores Ajustados
plot(valores_ajustados, residuos,
     main = "Residuos vs Valores Ajustados",
     xlab = "Valores Ajustados",
     ylab = "Residuos Estandarizados")

# Gráfico 2: Q-Q Plot
qqnorm(residuos, main = "Q-Q Plot")
qqline(residuos)
```

## 9.3 Hacer predicciones

r

```r
# Crear nuevos datos para predecir
nuevos_datos <- data.frame(var_numerica_1 = 100)  # Cambiar por valor deseado

# Realizar predicción
predict(modelo_1, nuevos_datos)
```

## 9.4 Modelo con variable categórica

r

```r
# Crear modelo: predecir variable dependiente usando variable categórica
modelo_2 <- lm(variable_dependiente ~ var_categorica_1, data = datos)

# Ver resultados
summary(modelo_2)
```

---

# PASO 10: REGRESIÓN LINEAL MÚLTIPLE

## 10.1 Modelo completo

r

```r
# Modelo con todas las variables
modelo_completo <- lm(variable_dependiente ~ var_numerica_1 + var_numerica_2 +
                      var_numerica_3 + var_categorica_1 + var_categorica_2,
                      data = datos)

# Ver resultados
summary(modelo_completo)
```

## 10.2 Modelo reducido (solo variables significativas)

r

```r
# Modelo solo con variables significativas (p < 0.05)
# NOTA: Cambiar variables según resultados del modelo completo
modelo_reducido <- lm(variable_dependiente ~ var_numerica_1 + var_numerica_2 +
                      var_categorica_1, data = datos)

# Ver resultados
summary(modelo_reducido)
```

## 10.3 Diagnóstico del modelo múltiple

r

```r
# Generar los 4 gráficos de diagnóstico automáticamente
par(mfrow = c(2,2))
plot(modelo_reducido)
```

## 10.4 Tests adicionales

r

```r
# Test de independencia de residuos (Durbin-Watson)
library(lmtest)
dwtest(modelo_reducido)

# Análisis de multicolinealidad (VIF)
library(car)
vif(modelo_reducido)
```

---

# PASO 11: VALIDACIÓN Y CAPACIDAD PREDICTIVA

## 11.1 Validación cruzada

r

```r
# Establecer semilla para reproducibilidad
set.seed(1234)

# Dividir datos: 75% entrenamiento, 25% prueba
indices_entrenamiento <- sample(nrow(datos), nrow(datos)*0.75, replace = FALSE)
datos_entrenamiento <- datos[indices_entrenamiento, ]
datos_prueba <- datos[-indices_entrenamiento, ]

# Entrenar modelo en datos de entrenamiento
modelo_validacion <- lm(variable_dependiente ~ var_numerica_1 + var_numerica_2 +
                        var_categorica_1, data = datos_entrenamiento)

# Hacer predicciones en datos de prueba
predicciones <- modelo_validacion %>% predict(datos_prueba)
```

## 11.2 Calcular métricas de evaluación

r

```r
# Calcular métricas de evaluación
r_cuadrado <- 1 - (sum((datos_prueba$variable_dependiente - predicciones)^2) /
                  sum((datos_prueba$variable_dependiente - mean(datos_prueba$variable_dependiente))^2))
rmse <- sqrt(mean((datos_prueba$variable_dependiente - predicciones)^2))
mae <- mean(abs(datos_prueba$variable_dependiente - predicciones))

# Mostrar resultados
data.frame(r_cuadrado, rmse, mae)

# Calcular tasa de error
tasa_error <- rmse / mean(datos_prueba$variable_dependiente)
print(paste("Tasa de error:", round(tasa_error*100, 2), "%"))
```

---

# PASO 12: HACER PREDICCIONES FINALES

## 12.1 Predicción con nuevos datos

r

```r
# Crear nuevos datos para predecir
datos_prediccion <- data.frame(var_numerica_1 = 120,   # Cambiar por valores deseados
                               var_numerica_2 = 70,
                               var_categorica_1 = "categoria_ejemplo")

# Realizar predicción con intervalo de confianza
predict(modelo, datos_prediccion, type = "response",
        se.fit = TRUE, interval = "confidence")$fit
```

---