



## CÓDIGO R STUDIO - T1 INTRODUCCIÓN AL ANÁLISIS MULTIVARIANTE

### VÍDEO 2 – ESTADÍSTICA DESCRIPTIVA

# Instalar librerías (ejecutar solo una vez)

```
install.packages("ggplot2")  
install.packages("dplyr")  
install.packages("gridExtra")  
install.packages("lmtest")  
install.packages("car")  
install.packages("readr")
```

# Cargar librerías necesarias (ejecutar en cada sesión)

```
library(ggplot2) # Para gráficos avanzados  
library(dplyr) # Para manipulación de datos  
library(gridExtra) # Para combinar gráficos  
library(readr)
```

# Verificar qué objetos se han cargado

```
ls()
```

# Ver la estructura del dataset

```
str(datos)
```

# Variables cuantitativas (numéricas)

```
variables_numericas <- data.frame(datos$Sales, datos$CompPrice, datos$Income,  
datos$Advertising, datos$Population, datos$Price, datos$Age, datos$Education)  
print(variables_numericas)
```

# Variables cualitativas

```
variables_cualitativas <- data.frame(datos$ShelveLoc, datos$Urban, datos$US)  
print(variables_cualitativas)
```

# Método rápido para una variable

```
mean(datos$Sales) # Media de la variable de interés  
median(datos$Sales) # Mediana de la variable de interés
```

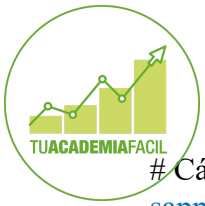
# Para múltiples variables

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

# Para una variable

```
summary(datos$Sales)
```





```
# Cálculo de una medida para todas las variables numéricas
```

```
sapply(variables_numericas, sd, na.rm = TRUE) # Desviaciones estándar
```

```
sapply(variables_numericas, IQR, na.rm = TRUE) # Rangos intercuartílicos
```

```
# Cuartiles de la variable de interés
```

```
quantile(datos$Price)
```

```
# Resumen completo de la variable de interés
```

```
summary(datos$Price)
```

```
# Boxplot para visualizar distribución y outliers
```

```
ggplot(datos, aes(y=Sales)) + geom_boxplot() + ggtitle("Distribución de Sales")
```

```
# Histograma para ver la forma de la distribución
```

```
ggplot(datos, aes(x=Price)) + geom_histogram(bins=10, fill="lightblue", color="black") +  
ggtitle("Histograma de Price")
```

```
# Scatter plot para relación entre dos variables
```

```
ggplot(datos, aes(x=Sales, y=Price)) + geom_point() + ggtitle("Relación Sales vs Price")
```

```
# Combinar en una sola visualización
```

```
g1<-ggplot(datos, aes(y=Sales)) + geom_boxplot() + ggtitle("Distribución de Sales")
```

```
g2<-ggplot(datos, aes(x=Sales, y=Price)) + geom_point() + ggtitle("Relación Sales vs Price")
```

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

```
# Frecuencias absolutas
```

```
table(datos$ShelveLoc)
```

```
# Frecuencias relativas (porcentajes)
```

```
prop.table(table(datos$ShelveLoc)) * 100
```

```
# Preparar datos para gráficos de variables cualitativas
```

```
resumen_US <- summarize(group_by(datos, US), n=length(US))
```

```
# Gráfico de barras
```

```
ggplot(resumen_US, aes(x=US, y=n, fill=US)) + geom_bar(stat="identity") +  
ggtitle("Frecuencia por Categoría")
```

```
resumen_ShelveLoc <- summarize(group_by(datos, ShelveLoc), n=length(ShelveLoc))
```

```
ggplot(resumen_ShelveLoc, aes(x=ShelveLoc, y=n, fill=ShelveLoc)) + geom_bar(stat="identity")  
+ ggtitle("Frecuencia por Categoría")
```

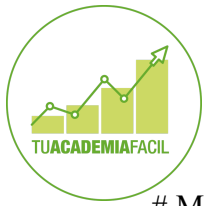
```
# Gráfico circular
```

```
ggplot(resumen_ShelveLoc, aes(x="", y=n, fill=ShelveLoc)) + geom_bar(width = 1, stat =  
"identity") + coord_polar("y", start=0) + ggtitle("Distribución por Categoría")
```

```
# Correlación entre dos variables específicas
```

```
cor(datos$Sales, datos$Price)
```





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

# Crear tabla de contingencia
tabla_contingencia <- table(datos$US, datos$ShelveLoc)
print("Tabla de contingencia:")
print(tabla_contingencia)

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

## VÍDEO 4 – REGRESIÓN LINEAL SIMPLE

```
# Crear modelo: predecir variable dependiente usando variable numérica
modelo_1 <- lm(Sales ~ Price, data = datos)
# Ver resultados del modelo
summary(modelo_1)

# Calcular residuos y valores ajustados
residuos <- rstandard(modelo_1)
valores_ajustados <- fitted(modelo_1)
# Configurar ventana de gráficos
par(mfrow = c(1,2))
# 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)

# Crear nuevos datos para predecir
nuevos_datos <- data.frame(Price = 100) # Cambiar por valor deseado
# Realizar predicción
predict(modelo_1, nuevos_datos)

# Crear modelo: predecir variable dependiente usando variable categórica
modelo_2 <- lm(Sales ~ Urban, data = datos)
# Ver resultados
summary(modelo_2)
```





## VÍDEO 6 – REGRESIÓN LINEAL MÚLTIPLE

# Modelo con todas las variables

```
modelo_completo <- lm(Sales ~ CompPrice + Income + Advertising + Population + Price +  
ShelveLoc + Age + Education + Urban + US, data = datos)
```

# Ver resultados

```
summary(modelo_completo)
```

# Modelo solo con variables significativas ( $p < 0.05$ )

# NOTA: Cambiar variables según resultados del modelo completo

```
modelo_reducido <- lm(Sales ~ Population + Price + Urban, data = datos)
```

# Ver resultados

```
summary(modelo_reducido)
```

# Generar los 4 gráficos de diagnóstico automáticamente

```
par(mfrow = c(2,2))
```

```
plot(modelo_reducido)
```

# Test de independencia de residuos (Durbin-Watson)

```
library(lmtest)
```

```
dwttest(modelo_reducido)
```

#Análisis de multicolinealidad (VIF)

```
library(car)
```

```
vif(modelo_reducido)
```

# Crear nuevos datos para predecir

```
datos_prediccion <- data.frame(Population = 120, # Cambiar por valores deseados  
Price = 70,  
Urban = "Yes")
```

# Realizar predicción con intervalo de confianza

```
predict(modelo_reducido, datos_prediccion, type = "response",  
se.fit = TRUE, interval = "confidence")$fit
```

