Análisis Exploratorio de Datos (EDA), Funciones Estadísticas y Visualización en Estadística R

Realizada por Juan Pablo Ramirez Correa

UNICISO
WWW.PORTALUNICISO.COM

UNICISO
WWW.PORTALUNICISO.COM

Tabla de Contenido

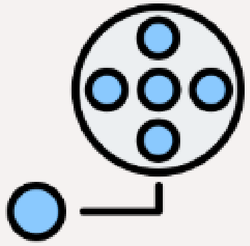
UNICISO
WWW.PORTALUNICISO.COM

¿Por qué es importante el Análisis Exploratorio de Datos (EDA)?

El Análisis Exploratorio de Datos (EDA) **es un paso esencial en cualquier estudio estadístico o de ciencia de datos**. Antes de aplicar modelos o realizar inferencias, es crucial entender la estructura de los datos, detectar patrones y evaluar posibles problemas como valores atípicos o datos faltantes. La meta principal del EDA es explorar, investigar y aprender de los datos.

Identifica patrones y relaciones entre variables.

Facilita la elección de métodos estadísticos adecuados.



Detecta valores atípicos y datos inconsistentes.

Mejora la comprensión de la distribución de los datos.

"Los beneficios del EDA se extienden más allá del análisis técnico, abarcando implicaciones estratégicas y empresariales, como una mejor toma de decisiones y la mitigación de riesgos" (IBM, 2021).

Paquetes esenciales en R para EDA

Para realizar un Análisis Exploratorio de Datos (EDA) en R, es fundamental contar con herramientas adecuadas**. Existen paquetes diseñados específicamente para manipular, resumir y visualizar datos de manera eficiente**.

Paquete	Descripción	Funciones clave para EDA	Ejemplo de uso en EDA
ggplot2	Visualización de datos basada en la gramática de gráficos.	ggplot(), aes(), geom_*(), facet_*(), labs(), scale_*(), theme()	Creación de diversos tipos de gráficos para explorar la distribución y las relaciones de los datos.
dplyr	Manipulación de datos.	filter(), select(), mutate(), arrange(), summarize(), group_by(), glimpse()	Limpieza, transformación y resumen de datos antes de la visualización.
tidyr	Transformación de datos ordenados.	gather(), spread(), separate(), unite(), drop_na(), fill()	Reestructuración de datos en un formato ordenado, adecuado para el análisis y la visualización.
readr	Lectura de datos rectangulares.	read_csv(), read_tsv()	Importación de archivos de datos a R.
tibble	Data frames modernos.	as_tibble()	Trabajando con una estructura de data frame más intuitiva.
skimr	Enfoque fluido para las estadísticas de resumen..	skim(), skim_with(), sfl()	Obtención de estadísticas de resumen completas, incluyendo valores faltantes e histogramas, para diferentes tipos y grupos de datos.

El ecosistema **tidyverse** en R agrupa una serie de paquetes diseñados para facilitar el Análisis Exploratorio de Datos (EDA).

dplyr **ggplot2** **tidyr**

readr **tibble** **skimr**

```
install.packages("tidyverse")
library(tidyverse)
```

UNICISO
WWW.PORTALUNICISO.COM

Exploración de Datos: Importación en R

El primer paso en un Análisis Exploratorio de Datos (EDA) **es importar los datos de manera eficiente y comenzar su inspección**. R ofrece múltiples funciones para cargar distintos formatos de datos, permitiendo una integración fluida con otras herramientas.

CSV y archivos delimitados: read_csv(), read_delim() (paquete readr).

Excel: read_excel() (paquete readxl).

Bases de datos: dbConnect() y dbReadTable() (paquete DBI).

Otros formatos: jsonlite::fromJSON(), readRDS(), foreign::read.spss().

```
install.packages("readxl")
library(readxl)  # Cargar la librería para leer archivos Excel
datos_excel <- read_excel(ruta_excel, sheet = 1)  # Leer la primera hoja
```

Nota: Para mayor información mirar la guía **“Transformando Datos en R: Fundamentos e para el Análisis”**

Inspeccionar la estructura de los datos (head(), str(), glimpse())

Antes de realizar un análisis exploratorio, **es fundamental comprender la estructura de los datos.** R proporciona varias funciones para inspeccionar rápidamente un conjunto de datos y evaluar sus variables.

Funciones clave para explorar un dataset en R:

```
# Cargar paquetes
library(dplyr)

# Inspeccionar la estructura del dataset
head(datos)      # Muestra las primeras 6 filas
tail(datos)      # Muestra las últimas 6 filas
str(datos)       # Información estructural del dataset
glimpse(datos)   # Vista compacta del dataset (dplyr)

# Otras funciones útiles
dim(datos)       # Número de filas y columnas
colnames(datos)  # Nombres de las columnas
```

Resumen estadístico de los datos en R

Después de inspeccionar la estructura de los datos, es importante obtener un resumen estadístico. Esto permite identificar valores atípicos, distribución de las variables y posibles inconsistencias. R ofrece varias funciones para calcular estadísticas descriptivas, como **summary**() para obtener medidas básicas y **skim**() del paquete skimr, que proporciona un análisis más detallado con visualización de distribuciones.

```
summary(datos)
```

Muestra estadísticas básicas (mínimo, máximo, media, mediana).

```
library(skimr)
skim(datos)
```

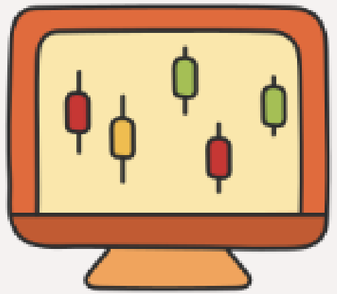
Proporciona **un resumen detallado, incluyendo valores faltantes y distribuciones.**

```
library(psych)
describe(datos)
```

Muestra estadísticas adicionales como desviación estándar y curtosis.

Detección de valores atípicos y datos faltantes

Identificar **valores atípicos y datos faltantes puede mejorar la interpretación de los resultados**. Los valores atípicos pueden distorsionar las medidas estadísticas y las visualizaciones, mientras que los datos faltantes pueden generar sesgos o reducir la precisión del modelo.



Valores atípicos:

- Utilizar diagramas de caja **(boxplot())** para identificar valores extremos.
- Calcular los cuantiles y el rango intercuartil (IQR) para detectar valores fuera de los límites típicos.
- Visualizar distribuciones con histogramas o gráficos de dispersión.

Datos faltantes:

- **Usar is.na()** para identificar celdas con valores faltantes.
- **sum(is.na(datos))** permite contar la cantidad de datos faltantes.
- **vis_miss()** del paquete naniar visualiza patrones de valores perdidos.

Código de ejemplo en R

```
# Cargar paquetes
library(ggplot2)
```

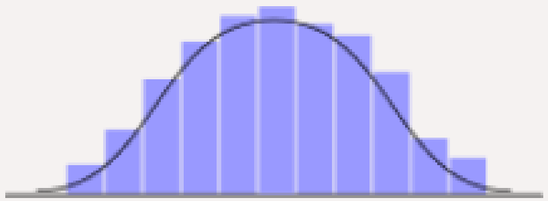
```
# Detección de valores atípicos con boxplot
ggplot(datos, aes(x = "", y = variable_numerica)) +
  geom_boxplot() +
  labs(title = "Detección de valores atípicos")
```

```
# Identificación de valores atípicos con IQR
Q1 <- quantile(datos$variable_numerica, 0.25,
na.rm = TRUE)
Q3 <- quantile(datos$variable_numerica, 0.75,
na.rm = TRUE)
IQR <- Q3 - Q1
outliers <- datos$variable_numerica < (Q1 - 1.5 *
IQR) |
            datos$variable_numerica > (Q3 + 1.5 *
IQR)
sum(outliers)  # Número de valores atípicos
```

```
# Identificación de datos faltantes
sum(is.na(datos))  # Cantidad de valores faltantes
```

Existen paquetes como **naniar que ofrecen herramientas para visualizar los patrones de datos faltantes**, lo que puede proporcionar información valiosa sobre su distribución.

```
library(naniar)
# Visualización de valores faltantes
vis_miss(datos)
```



Nota: "variable_numerica" se refiere a la variable a la cual se le realiza el proceso. **(datos$variable_numerica)**

Funciones Estadísticas Claves

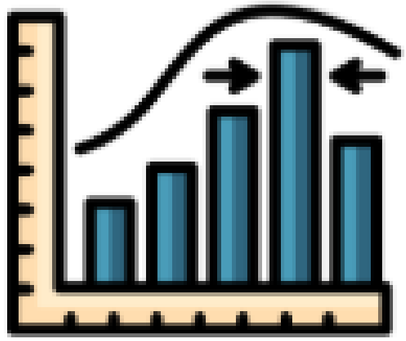
UNICISO
WWW.PORTALUNICISO.COM

Medidas de tendencia central:

Las medidas de tendencia central permiten describir el valor típico de un conjunto de datos. Estas estadísticas resumen la distribución de los valores y ayudan a comprender la estructura de los datos.

Media (promedio): Representa el valor promedio de un conjunto de datos y se calcula sumando todos los valores y dividiéndolos por el número total de observaciones.

```
# Cálculo de medidas de tendencia central (Media)
media_valor <- mean(datos$variable_numerica, na.rm = TRUE)  # Media
media_valor
```



Mediana: Es el valor central cuando los datos están ordenados.

```
# Cálculo de medidas de tendencia central (Media)
mediana_valor <- median(datos$variable_numerica, na.rm = TRUE)  # Mediana
mediana_valor
```

Moda: Es el valor que más se repite en el conjunto de datos. Puede haber más de una moda o incluso ninguna si todos los valores son únicos.

```
# Cálculo de la moda
moda_valor <- datos %>%
  count(variable_numerica) %>%
  filter(n == max(n)) %>%
  pull(variable_numerica)
moda_valor
```

Medidas de dispersión:

Las medidas de dispersión indican cuán dispersos o agrupados están los datos alrededor de la tendencia central.

```
# Cálculo de medidas de dispersión
rango_valor <- range(datos$variable_numerica, na.rm = TRUE)  # Rango
varianza_valor <- var(datos$variable_numerica, na.rm = TRUE)  # Varianza
desviacion_valor <- sd(datos$variable_numerica, na.rm = TRUE)  # Desviación estándar
iqr_valor <- IQR(datos$variable_numerica, na.rm = TRUE)  # Rango intercuartil
```

1. Rango:

Diferencia entre el valor máximo y mínimo en el conjunto de datos.

2. Varianza:

Mide la dispersión de los datos respecto a la media.

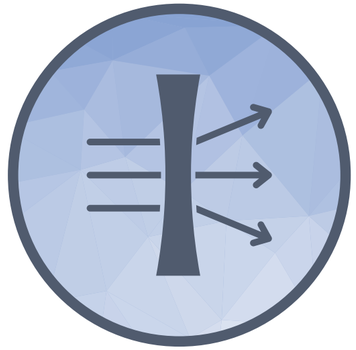
3. Desviación estándar:

Es la raíz cuadrada de la varianza y representa la dispersión en las mismas unidades que los datos originales.

4. Rango intercuartílico (IQR):

Mide la dispersión entre el primer y tercer cuartil (Q3 - Q1), lo que permite evaluar la variabilidad excluyendo valores extremos.

```
# Mostrar resultados
rango_valor
varianza_valor
desviacion_valor
iqr_valor
```



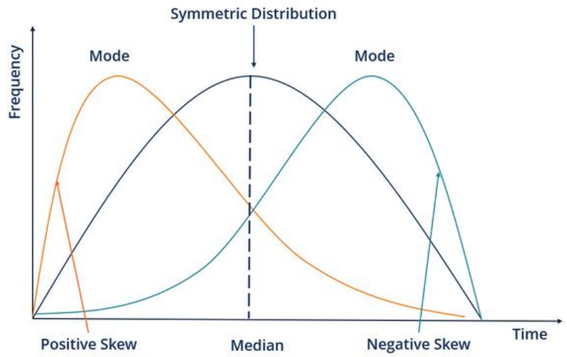
Distribución de datos (asimetría y curtosis)

Asimetría:

Indica si los datos están sesgados hacia la derecha o la izquierda.

- **Asimetría positiva:** La cola derecha es más larga (valores extremos más grandes).
- **Asimetría negativa:** La cola izquierda es más larga (valores extremos más pequeños).
- **Asimetría cercana a 0:** La distribución es aproximadamente simétrica.

```
# Calcular asimetría.
asimetria <- skewness(datos$variable_numerica, na.rm = TRUE)
asimetria
```

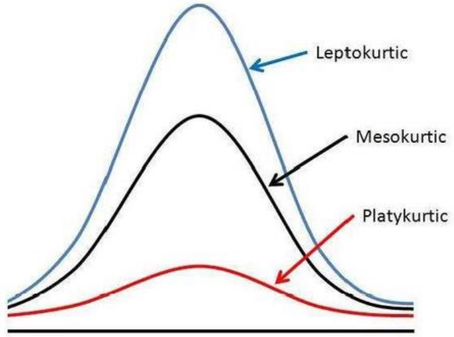


Curtosis:

Mide si la distribución tiene colas más largas o más cortas que una distribución normal.

- **Curtosis alta (>3):** Distribución con colas pesadas (leptocúrtica).
- **Curtosis baja (<3):** Distribución con colas ligeras (platicúrtica).
- **Curtosis cercana a 3:** Similar a la distribución normal (mesocúrtica).

```
# Calcular asimetría.
curtosis_valor <- kurtosis(datos$variable_numerica, na.rm = TRUE)
curtosis_valor
```



UNICISO
WWW.PORTALUNICISO.COM

Tablas de frecuencias y cruces de variables

Las tablas de frecuencia permiten analizar la distribución de valores en variables categóricas y numéricas discretas. Además, los cruces de variables ayudan a identificar relaciones entre diferentes categorías en un conjunto de datos.

Tablas de frecuencia absoluta:

Indican cuántas veces aparece cada valor en una variable.

```
# Tabla de frecuencia absoluta
table(datos$variable_categorica)
```

Tablas de frecuencia relativa:

Muestran la proporción de cada valor en relación con el total.

```
# Tabla de frecuencia relativa
prop.table(table(datos$variable_categorica))
```

Cruces de variables:

Permiten analizar la distribución conjunta de dos variables categóricas.

```
# Cruzar dos variables categóricas
table(datos$variable1, datos$variable2)
```

UNICISO
WWW.PORTALUNICISO.COM

Visualización de Datos con ggplot2

Se basa en la "**gramática de los gráficos**". Los gráficos se construyen paso a paso mediante capas, permitiendo un alto grado de personalización y claridad en la representación de datos.

Paquete	Función	Ejemplo de uso
ggplot()	Inicia el gráfico y define el dataset. No genera un gráfico por sí mismo, pero establece qué datos se usarán.	ggplot(data = df)
aes()	Mapea las variables del dataset a propiedades visuales como ejes, colores, tamaños y formas. Especifica cómo se relacionan los datos con la apariencia del gráfico.	ggplot(df, aes(x = variable_x, y = variable_y))
geom_*()	Añade capas geométricas para representar los datos en distintas formas. Por ejemplo, geom_point() para puntos, geom_bar() para barras y geom_line() para líneas.	ggplot(df, aes(x, y)) + geom_point()
labs()	Personaliza los títulos del gráfico, etiquetas de los ejes y la leyenda. Es útil para mejorar la interpretación del gráfico.	ggplot(df, aes(x, y)) + geom_point() + labs(title = "Gráfico de dispersión", x = "Eje X", y = "Eje Y")
facet_wrap()	Divide el gráfico en múltiples subgráficos según una variable categórica, permitiendo comparaciones entre grupos.	ggplot(df, aes(x, y)) + geom_point() + facet_wrap(~categoria)
theme_*()	Ajusta la apariencia del gráfico, incluyendo colores, fuentes y fondos. Hay varios estilos predefinidos como theme_minimal(), theme_classic(), y theme_dark().	ggplot(df, aes(x, y)) + geom_point() + theme_minimal()

ggplot() **aes()**

geom_*() **labs()**

facet_wrap()

theme_*()

```
install.packages("ggplot2")
library(ggplot2)
```

Gráficos de Distribución: Histogramas y Boxplots

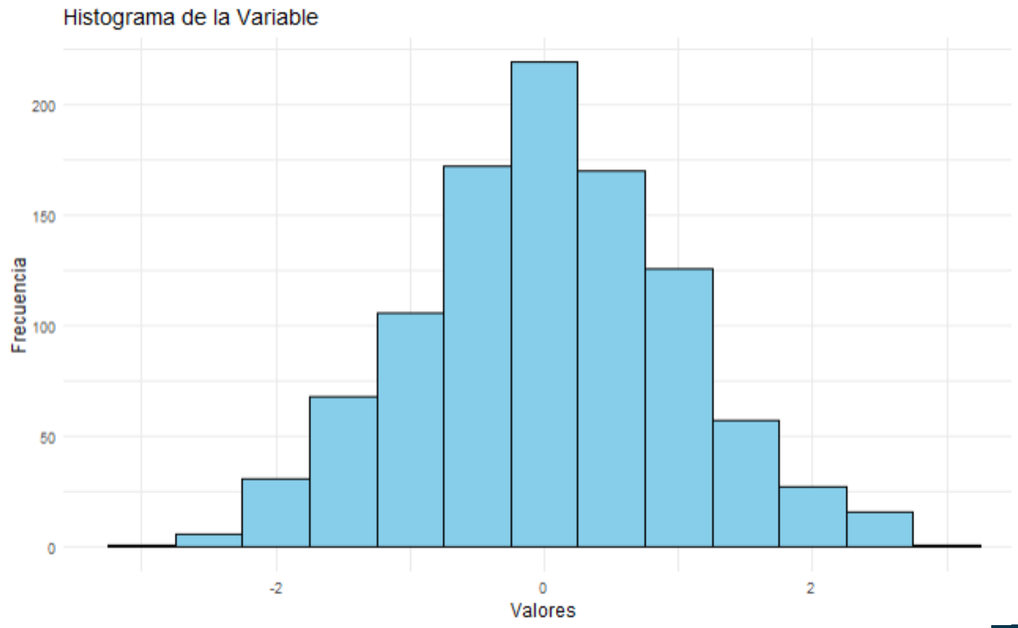
Histograma:

Se utilizan para ver la distribución de una variable numérica dividiéndola en intervalos (bins).

```
# Generar datos aleatorios
set.seed(123)
datos <- data.frame(valor = rnorm(1000))  # 200 valores distribuidos normalmente

# Histograma básico
ggplot(datos, aes(x = valor)) +
  geom_histogram(binwidth = 0.5, fill = "skyblue", color = "black") +
  labs(title = "Histograma de la Variable", x = "Valores", y = "Frecuencia") +
  theme_minimal()
```

Corresponde a una distribución normal



ggplot + aes + **geom_histogram** + labs + theme_minimal

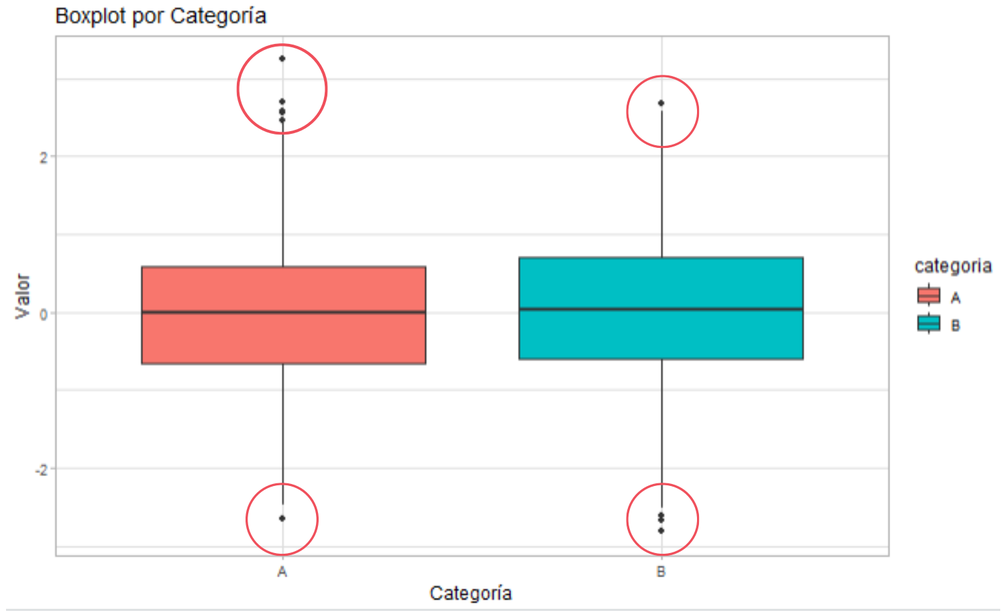
Boxplot:

Muestran la distribución de los datos, identificando valores atípicos y dispersión.

```
# Generar datos con categorías
datos$categoria <- sample(c("A", "B"), 200, replace = TRUE)

# Boxplot básico
ggplot(datos, aes(x = categoria, y = variable_x, fill = categoria)) +
  geom_boxplot() +
  labs(title = "Boxplot por Categoría", x = "Categoría", y = "Valor") +
  theme_light()
```

Se presentan datos atípicos en el grafico de boxplot para la categoría A y B



ggplot + aes + **geom_boxplot** + labs + theme_minimal

UNICISO
WWW.PORTALUNICISO.COM

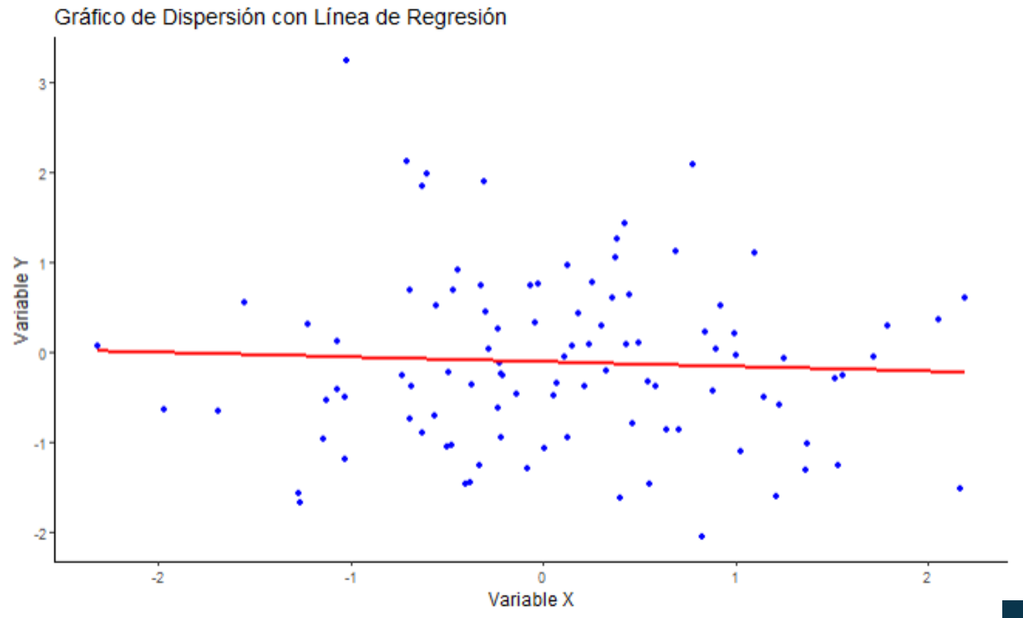
Gráficos de Relación: Dispersión y Líneas

Gráficos de Dispersión: Ideales para ver correlaciones entre variables.

```
# Datos para el gráfico de dispersión
set.seed(123)
relacion_data <- data.frame(x = rnorm(100), y = rnorm(100))

# Gráfico de dispersión con línea de regresión
ggplot(relacion_data, aes(x = x, y = y)) +
  geom_point(color = "blue") +  # Puntos del gráfico
  geom_smooth(method = "lm", color = "red", se = FALSE) +  # Línea de
regresión
  labs(title = "Gráfico de Dispersión con Línea de Regresión",
       x = "Variable X",
       y = "Variable Y") +
  theme_classic()
```

Datos no correlacionados



ggplot + aes + **geom_point** + **geom_smooth +**
labs + theme_classic

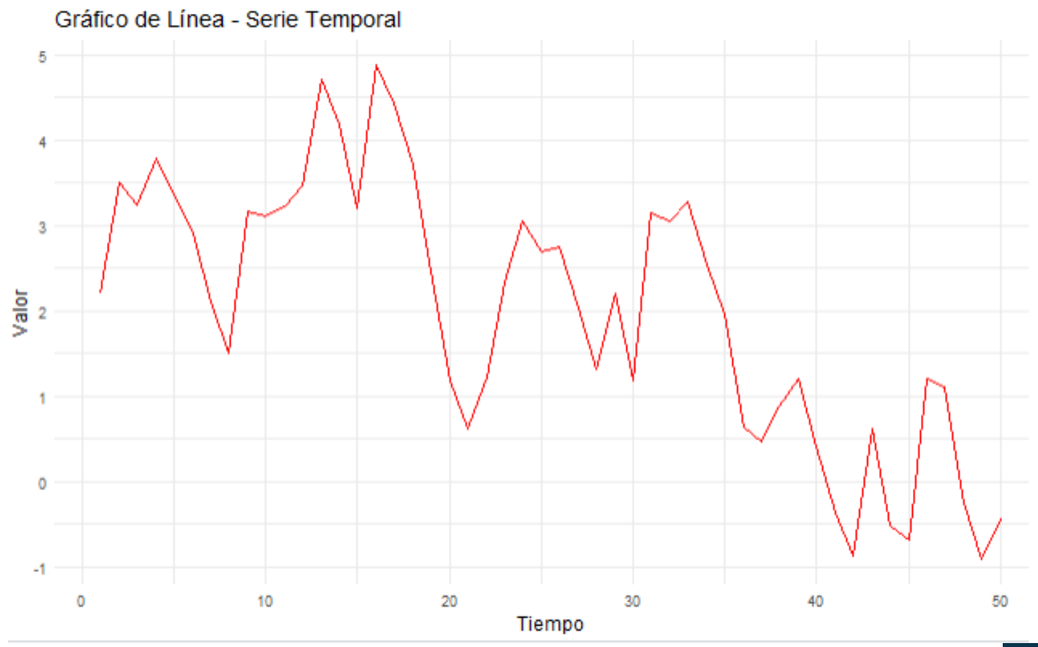
Gráficos de Líneas:

Útiles para analizar tendencias a lo largo del tiempo.

```
# Generar datos temporales
time_data <- data.frame(
  tiempo = 1:50,
  valor = cumsum(rnorm(50)))

# Gráfico de líneas
ggplot(time_data, aes(x = tiempo, y = valor)) +
  geom_line(color = "red") +
  labs(title = "Gráfico de Línea - Serie Temporal", x = "Tiempo", y =
"Valor") +
  theme_minimal()
```

Gráfico de tendencia - Serie de tiempo



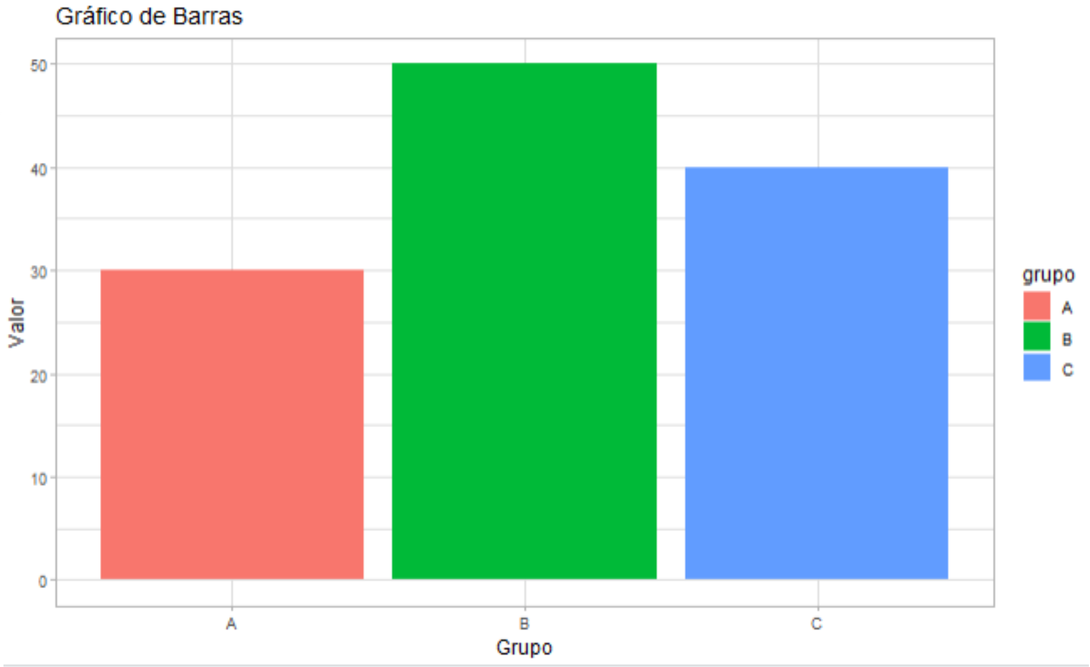
ggplot + aes + **geom_line** + labs + theme_minimal

Gráficos de Comparación: Barras y Sectores

Gráficos de Barras: Usados para comparar frecuencias entre grupos.

```
# Datos de categorías
categorias <- data.frame(
  grupo = c("A", "B", "C"),
  Valor = c(30, 50, 40)
)

# Gráfico de barras
ggplot(categorias, aes(x = grupo, y = valor, fill = grupo)) +
  geom_bar(stat = "identity") +
  labs(title = "Gráfico de Barras", x = "Grupo", y = "Valor") +
  theme_light()
```



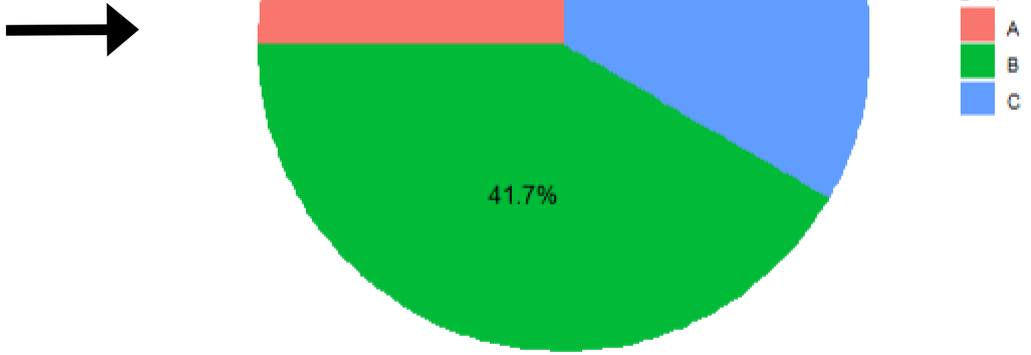
ggplot + aes + **geom_bar +** labs + theme_light

Gráficos de Sectores (Pie Chart):

Representan proporciones de un conjunto.

```
# Calcular los porcentajes
categorias <- categorias %>%
  mutate(porcentaje = Valor / sum(Valor)*100) # Proporción de cada categoría

# Gráfico de sectores con etiquetas de porcentaje
ggplot(categorias, aes(x = "", y = Valor, fill = grupo)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  geom_text(aes(label = sprintf("%.1f%%", porcentaje)),
            position = position_stack(vjust = 0.5)) +  # Agregar etiquetas de porcentaje
  labs(title = "Gráfico de Sectores con Porcentajes") +
  theme_void()
```



ggplot + aes + **geom_bar** + coord_polar + **geom_text +** labs + theme_void

Errores Comunes en el Análisis Exploratorio de Datos (EDA) y Cómo Evitarlos

Tipo de Error	Descripción	Solución
No revisar la estructura de los datos	Trabajar sin inspeccionar la estructura, tipos de variables o valores faltantes.	Usar str(), summary(), skim() y glimpse() para entender el dataset antes del análisis.
No tratar valores faltantes	Ignorar valores NA o eliminarlos sin evaluar su impacto en el análisis.	Identificar valores faltantes (is.na()), imputar con mediana/media o eliminar según sea necesario.
No identificar valores atípicos	No detectar outliers que puedan distorsionar las métricas.	Usar boxplots y estadísticas descriptivas para identificarlos (geom boxplot()).
No visualizar los datos antes del modelado	Saltar directamente a modelos sin analizar la distribución de los datos.	Utilizar gráficos exploratorios como histogramas, boxplots y gráficos de dispersión.
No considerar la escala de las variables	Comparar variables con escalas muy diferentes sin normalizarlas.	Normalizar los datos con scale() para mejorar comparaciones.
Confiar ciegamente en métricas sin interpretar los datos	Depender solo de medidas estadísticas sin verificar patrones visuales.	Complementar análisis numérico con visualización de datos (geom density()).

UNICISO
WWW.PORTALUNICISO.COM

Referencias

- Datos abiertos del Gobierno de España. (2024). Guía práctica de introducción al análisis exploratorio de datos en R. https://datos.gob.es/es/documentacion/guia-practica-de-introduccion-al-analisis-exploratorio-de-datos
- GeeksforGeeks. (2023). Exploratory Graphs for EDA in R. https://www.geeksforgeeks.org/exploratory-graphs-for-eda-in-r/
- Grolemund, H. W. and G. (2023a). 7 Exploratory Data Analysis | R for Data Science. https://r4ds.had.co.nz/exploratory-data-analysis.html
- Grolemund, H. W. and G. (2023b). R for Data Science. https://r4ds.had.co.nz/
- IBM. (2021, October 4). What is Exploratory Data Analysis? |. https://www.ibm.com/think/topics/exploratory-data-analysis
- JMP Statistical Discovery. (2025). Análisis exploratorio de datos. https://www.jmp.com/es/statistics-knowledge-portal/exploratory-data-analysis
- Keilor Rojas-Jimenez. (2022). Ciencia de Datos para Ciencias Naturales. https://bookdown.org/keilor_rojas/CienciaDatos/
- r4ds. (2023a). R Para Ciencia de Datos—7 Análisis exploratorio de datos (EDA). https://es.r4ds.hadley.nz/07-eda.html
- r4ds. (2023b, June 6). R Para Ciencia de Datos. https://es.r4ds.hadley.nz/
- Rojas-Jimenez, K. (2022). Capítulo 3 Visualización de Datos | Ciencia de Datos para Ciencias Naturales. https://bookdown.org/keilor_rojas/CienciaDatos/visualizaci%C3%B3n-de-datos.html

CITA DE LA GUÍA

Ramirez, J.P. (2025). Análisis exploratorio de datos (EDA). Funciones, datos y visualizaciones en estadística R. UNICISO. Disponible en: www.portaluniciso.com

SÍGUENOS:

UNICISO
WWW.PORTALUNICISO.COM