

Ejercicio Básico — Análisis de Ventas en Excel con R

Enunciado del ejercicio

Este ejercicio tiene como objetivo practicar el trabajo con archivos Excel en R utilizando las funciones más sencillas posibles y sin aplicar formatos ni gráficos.

Se parte de un archivo existente llamado `ventas_ejercicio.xlsx`, que contiene tres hojas: Enero, Febrero y Marzo. Cada hoja tiene las columnas:

- fecha
- producto
- categoria
- unidades
- precio

Tu tarea es escribir un script en R que realice los siguientes pasos:

1. Verificar que el archivo '`ventas_ejercicio.xlsx`' existe en el directorio de trabajo.
2. Leer todas las hojas (Enero, Febrero y Marzo) y combinarlas en una sola tabla.
3. Limpiar los nombres de las columnas para facilitar el manejo.
4. Calcular una nueva columna llamada '`ventas`' como `unidades * precio`.
5. Crear un resumen por categoría con la suma total de ventas y la media de unidades vendidas.
6. Obtener el Top 5 de productos con mayores ventas totales.
7. Exportar un nuevo archivo '`reporte_basico.xlsx`' con dos hojas: '`Resumen`' y '`TopProductos`'.

Explicación paso a paso

1) Verificar existencia del archivo

Antes de comenzar, se comprueba que el archivo 'ventas_ejercicio.xlsx' está en el directorio de trabajo.

Esto evita errores si el archivo no se encuentra.

```
if (!file.exists('ventas_ejercicio.xlsx')) stop('No se encontró el archivo.')
```

2) Leer y unir todas las hojas

Se usa `readxl::excel_sheets()` para obtener los nombres de las hojas, y `lapply()` junto con `read_excel()` para leer cada una. Luego se combinan con `bind_rows()`.

```
library(readxl)
hojas <- excel_sheets('ventas_ejercicio.xlsx')
listas <- lapply(hojas, function(sh) read_excel('ventas_ejercicio.xlsx', sheet = sh))
todas <- dplyr::bind_rows(listas, .id = '.indice_hoja')
```

3) Limpiar nombres y calcular ventas

Con `janitor::clean_names()` se limpian los nombres de las columnas (sin espacios ni acentos). Luego, con `mutate()` de `dplyr` se crean nuevas columnas y se convierten los tipos de datos si es necesario.

```
library(dplyr); library(janitor)
todas <- todas |> clean_names() |>
  mutate(
    fecha = as.Date(fecha),
    unidades = as.numeric(unidades),
    precio = as.numeric(precio),
    ventas = unidades * precio
  )
```

4) Crear el resumen por categoría

Usando `group_by()` y `summarise()` de `dplyr`, se agrupan los datos por categoría y se calculan dos medidas: `ventas_total` (suma) y `unidades_media` (promedio).

```
resumen <- todas |>
  group_by(categoría) |>
  summarise(
    ventas_total = sum(ventas, na.rm = TRUE),
    unidades_media = mean(unidades, na.rm = TRUE),
    .groups = 'drop'
  ) |>
  arrange(desc(ventas_total))
```

5) Calcular el Top 5 de productos

Se agrupan los datos por producto, se suman las ventas y se seleccionan los cinco primeros con slice_head().

```
top_productos <- todas |>
  group_by(producto) |>
  summarise(ventas_total = sum(ventas, na.rm = TRUE), .groups = 'drop') |>
  arrange(desc(ventas_total)) |>
  slice_head(n = 5)
```

6) Exportar el reporte final

Con writexl::write_xlsx() se exportan las dos tablas (Resumen y TopProductos) en un solo archivo Excel. Cada elemento de la lista corresponde a una hoja distinta.

```
library(writexl)
writexl::write_xlsx(
  list(Resumen = resumen, TopProductos = top_productos),
  path = 'reporte_basico.xlsx'
)
```

Resultado esperado

Al ejecutar el script, se genera el archivo 'reporte_basico.xlsx' con dos hojas:

- Hoja 'Resumen': muestra cada categoría con el total de ventas y el promedio de unidades.
- Hoja 'TopProductos': lista los cinco productos con mayores ventas totales.

No se aplica formato ni estilos, ideal para centrarse en la lógica del flujo de datos.