

NOMBRE COMPLETO: Ing. Hilaría Adima Vásquez Durán

NAO ID: 3033

FECHA: 2 de agosto de 2024

**NOMBRE DE LA TRAYECTORIA
EN LA QUE ESTÁS ENROLADO:** DATA ANALYST CORE

Título del Reto:
Programación R al elaborar reportes estadísticos

SPRINT 1 DESARROLLO R1

Crea un conjunto de programas en R para explorar los datos de Rossmann Pharma a través de tablas con resúmenes estadísticos y visualizaciones que cumplan con las siguientes directrices:

EJERCICIO 4

Realiza un programa("top_10_performance_stores_by_country.R") que determine cuales son las tiendas que se encuentran en top 10 en valor monetario de ventas y top 19 en cantidad de clientes de cada país, considerando el desempeño de tres meses hacia atrás a partir de la última fecha disponible donde existe data. El resultado se deberá proporcionar mediante una visualización donde a la vez se indique la cantidad promedio de ventas de cada tienda del top y promedio de clientes de cada tienda. Dicha visualización será denominada "top_10_performance_stores_by_country.png".

Desarrollo del ejercicio

Carga las bibliotecas necesarias para la manipulación de datos (tidyverse), la lectura de archivos Excel (readxl) y la manipulación de fechas (lubridate).

```
```{r}
library(tidyverse)
library(readxl)
```

```
library(lubridate)
```
```

```
> library(tidyverse)
— Attaching core tidyverse packages — tidyverse 2.0.0 —
✓ dplyr      1.1.4    ✓ readr      2.1.5
✓ forcats    1.0.0    ✓ stringr    1.5.1
✓ ggplot2    3.5.1    ✓ tibble     3.2.1
✓ lubridate  1.9.3    ✓ tidyr      1.3.1
✓ purrr      1.0.2
— Conflicts — tidyverse_conflicts() —
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
> library(readxl)
> library(lubridate)
```

#Establecer el directorio de trabajo a la ruta especificada en

```
DATA_PATH
```

```
DATA_PATH <- "/Users/Usuario/Desktop/Help ME"
setwd(DATA_PATH)
cat("Directorio de trabajo establecido a:", DATA_PATH, "\n")
```

Cargar los datos

```
# Establecer el directorio de trabajo
DATA_PATH <- "/Users/Usuario/Desktop/Help ME"
setwd(DATA_PATH)

# Leer los datos de ventas
SALES_DATA <- 'sales.csv'
sales <- read.csv2(SALES_DATA, sep=";", header=T)

# Leer los datos de tiendas
STORES_INFO_DATA := 'stores_info.xlsx'
stores <- read_excel(STORES_INFO_DATA, sheet = "data")
```

```
> # Establecer el directorio de trabajo
> DATA_PATH <- "/Users/Usuario/Desktop/Help ME"
> setwd(DATA_PATH)
> # Leer los datos de ventas
> SALES_DATA <- 'sales.csv'
> sales <- read.csv2(SALES_DATA, sep=";", header=T)
> # Leer los datos de tiendas
> STORES_INFO_DATA <- 'stores_info.xlsx'
> stores <- read_excel(STORES_INFO_DATA, sheet = "data")
> 
```

```
# Unir las tablas de ventas y sales_all <- sales
%>% left_join(stores, sales_all <- sales %>%
by=c("store"))
```

```
> # Unir las tablas de ventas y tiendas
> sales_all <- sales %>% left_join(stores, by=c("store"))
> 
```

Convertir la columna **date** a un objeto de fecha.

```
# Convertir la columna de fechas a tipo Date
sales_all$date <- ymd(sales_all$date)
```

Determinar la última fecha disponible en los datos y se calcula la fecha de tres meses antes.

Determinar La Última Fecha Disponible

```
last_date <- max(sales_all$date, na.rm = TRUE)
# Calcular la fecha de tres meses atrás
start_date <- last_date %m-% months(3)
```

```
> # Unir las tablas de ventas y tiendas
> sales_all <- sales %>% left_join(stores, by=c("store"))
> # Convertir la columna de fechas a tipo Date
> sales_all$date <- ymd(sales_all$date)
> # Determinar la última fecha disponible
> last_date <- max(sales_all$date, na.rm = TRUE)
> # Calcular la fecha de tres meses atrás
> start_date <- last_date %m-% months(3)
```

Filtrar los datos para incluir solo los registros de los últimos tres meses.

Filtrar datos de los últimos tres meses

```
recent_sales <- sales_all %>% filter(date >= start_date & date
```

Agrupar los datos por país y tienda, se calculan las ventas totales y el número total de clientes por tienda, se ordenan las tiendas por ventas totales en orden descendente, y se seleccionan las 10 tiendas principales por país.

Calcular el top 10 en valor monetario de ventas y en cantidad

```
top_stores_sales <- recent_sales %>%
  group_by(country, store) %>%
  summarise(total_sales = sum(sales), total_customers = sum(c
  arrange(desc(total_sales)) %>%
  group_by(country) %>%
  slice_head(n = 10) %>%
  ungroup()
```

```

> # Filtrar datos de los últimos tres meses
> recent_sales <- sales_all %>% filter(date >= start_date & date <= last_date)
> # Calcular el top 10 en valor monetario de ventas y en cantidad de clientes $
> top_stores_sales <- recent_sales %>%
+   group_by(country, store) %>%
+   summarise(total_sales = sum(sales), total_customers = sum(customers)) %>%
+   arrange(desc(total_sales)) %>%
+   group_by(country) %>%
+   slice_head(n = 10) %>%
+   ungroup()
`summarise()` has grouped output by 'country'. You can override using the `.groups` argument.
> 

```

Calcular Los Promedios De Ventas Y Clientes Para Las 10 Tiendas Principales De Cada País.

```

# Calcular promedios de ventas y clientes para el
top 10 top_stores_avg
<- top_stores_sales %>%   group_by(country)
%>%summarise(avg_sales = mean(total_sales),
avg_customers = mea

```

Crear una visualización usando **ggplot2**, mostrando un gráfico de barras del top 10 de tiendas por valor monetario de ventas, con etiquetas que indican las ventas totales. La visualización se faceta por país, y las etiquetas de las tiendas se rotan para mejorar la legibilidad. Luego, el gráfico se muestra en la consola.

```

# Crear visualización top_stores_plot

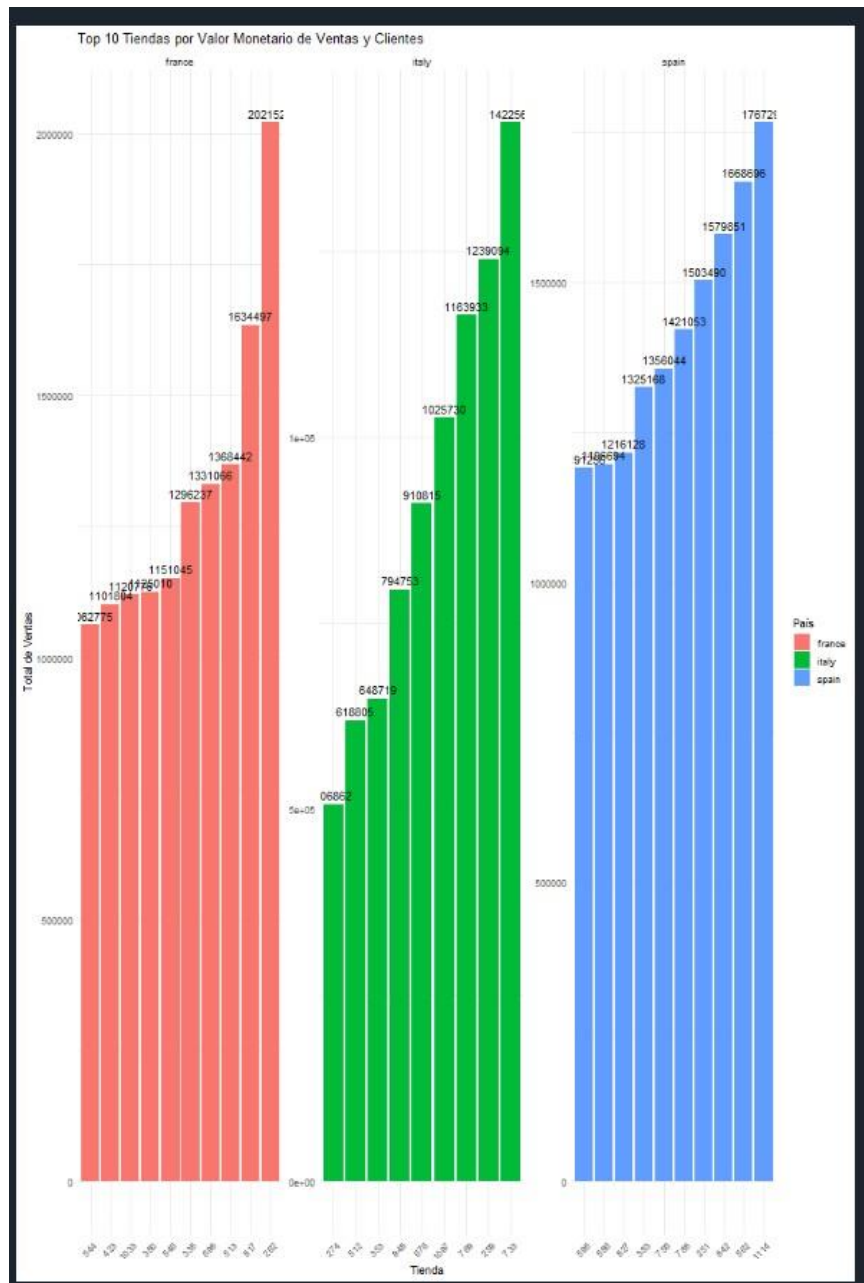
ggplot(top_stores_sales, aes(x = reorder(s
geom_bar(stat = "identity") +   geom_text(aes(label =
total_sales), vjust = -0.5) +   labs(title = "Top 10
Tiendas por Valor Monetario de Ventas y           x =
"Tienda", y = "Total de Ventas",           fill = "País") +
theme_minimal() +
  facet_wrap(~country, scales = "free") +
  theme(axis.text.x = element_text(angle = 45, hjust =
1))

```

Mostrar la visualización en la consola

print(top_stores_plot)

```
>
> # Calcular promedios de ventas y clientes para el top 10
> top_stores_avg <- top_stores_sales %>%
+   group_by(country) %>%
+   summarise(avg_sales = mean(total_sales), avg_customers = mean(total_customers))
> # Crear visualización
> top_stores_plot <- ggplot(top_stores_sales, aes(x = reorder(store, total_sales),
+   geom_bar(stat = "identity") +
+   geom_text(aes(label = total_sales), vjust = -0.5) +
+   labs(title = "Top 10 Tiendas por Valor Monetario de Ventas y Clientes",
+     x = "Tienda", y = "Total de Ventas",
+     fill = "País") +
+   theme_minimal() +
+   facet_wrap(~country, scales = "free") +
+   theme(axis.text.x = element_text(angle = 45, hjust = 1))
>
> # Mostrar la visualización en la consola
> print(top_stores_plot)
>
```



Guardar el gráfico como un archivo PNG con un tamaño especificado de 10.7 pulgadas.

Guardar la figura

```
ggsave("top_10_performance_stores_by_country.png", plot = top_
```

```
>  
> # Imprimir mensaje de éxito  
> cat("La figura 'top_10_performance_stores_by_country.png' ha sido creada exitosamente.  
La figura 'top_10_performance_stores_by_country.png' ha sido creada exitosamente.  
> Guarda este código en un archivo con el nombre 'top_10_performance_stores_by$  
Error: unexpected symbol in \"Guarda este\"  
> █
```

