# Informe de Análisis de Ventas en US.

# Programación en R

# Definición del problema/objetivo de investigación

## Problemática:

La empresa "IquiqueMiami LTDA." ha observado fluctuaciones significativas en sus ingresos durante el último año fiscal. A pesar de tener una amplia variedad de productos y presencia en varias estados del país de Estados Unidos, la empresa ha enfrentado desafíos debido a la alta competitividad en el mercado y cambios en las preferencias de los consumidores. La dirección ha identificado que una estrategia de precios ineficaz podría ser una de las causas principales de estos resultados inconsistentes.

## Objetivos:

# Introducción

En los últimos años, el movimiento económico de empresas pequeñas enfrenta desafíos constantes debido a la competencia, las cambiantes preferencias de los consumidores o clientes, y la rápida evaluación de la tecnología. “Iquique Miami LTDA.”, una empresa en el sector minorista, se ha comprometido a mejorar sus ventas y la satisfacción del cliente mediante la implementación de un enfoque basado en datos.

## Descripción del conjunto de datos

El conjunto de datos fue obtenido a través de la página web [*https://excelbianalytics.com/wp/downloads-18-sample-csv-files-data-sets-for-testing-sales/*](https://excelbianalytics.com/wp/downloads-18-sample-csv-files-data-sets-for-testing-sales/)donde provee diversas fuentes de datos de tipo CSV que son de prueba en el contexto de ventas en estados unidos. Las características técnicas del archivo CSV utilizado son:

Tabla X: descripción técnica del archivo

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del Archivo | Tamaño del archivo | Total columnas | Tipo Archivo |
| 5m\_sales | 624 MB | 14 | csv |

Al descargar el archivo permite visualizar de forma previa la estructura y algunos datos debido a la cantidad de registros existentes (5M de registros), sin embargo es suficiente para leer, analizar y comprender el encabezado del conjunto de datos para el desarrollo del diccionario de datos.

### Diccionario de datos

Visualizar el encabezado del conjunto de datos es un buen comienzo para comprender el modelo de negocios del cual fue extraído anteriormente, es por esto que se detalla cada columna o variable en la siguiente Tabla X:

Tabla X: diccionario de datos

|  |  |  |
| --- | --- | --- |
| Nombre de la Columna | Descripción | Tipo de Datos |
| Region | Región del mundo donde se realizó la venta | Categórico |
| Country | País donde se realizó la venta | Categórico |
| Item\_Type | Tipo de artículo vendido | Categórico |
| Sales\_Channel | Canal de ventas (Online u Offline) | Categórico |
| Order\_Priority | Prioridad de la orden (H: Alta, M: Media, L: Baja, C: Crítica) | Categórico |
| Order\_Date | Fecha en que se realizó la orden | Fecha |
| Order\_ID | Identificador único de la orden | Numérico |
| Ship\_Date | Fecha en que se envió la orden | Fecha |
| Units\_Sold | Número de unidades vendidas | Numérico |
| Unit\_Price | Precio por unidad | Numérico |
| Unit\_Cost | Costo por unidad | Numérico |
| Total\_Revenue | Ingresos totales de la venta | Numérico |
| Total\_Cost | Costos totales de la venta | Numérico |
| Total\_Profit | Beneficio total de la venta | Numérico |

# Preprocesamiento de Datos

En un dataset es importante explorar y conocer las variables o tuplas que la componen, ya que esto va de la mano con la problemática de la empresa. En primer lugar, el desarrollo de este trabajo, se desglosará de la siguiente manera, tal como se ve en la Imagen X:



Imagen X: flujo de trabajo

En segundo lugar, se debe capturar o leer los datos del dataset de tipo CSV y guardarlos en un objeto, de tal manera que permita acceder a los atributos del mismo, se debe usar el siguiente código:

|  |
| --- |
| datos <- read\_csv("./data/5m\_sales.csv") |

En segundo lugar, al almacenar los datos del conjunto de datos en el objeto “datos”, podemos usar funciones preestablecidas para conocer los datos del mismo, tal como se ve en la imagen X:

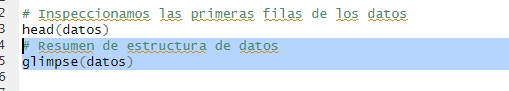


Imagen X: Inspeccionando el conjunto de datos

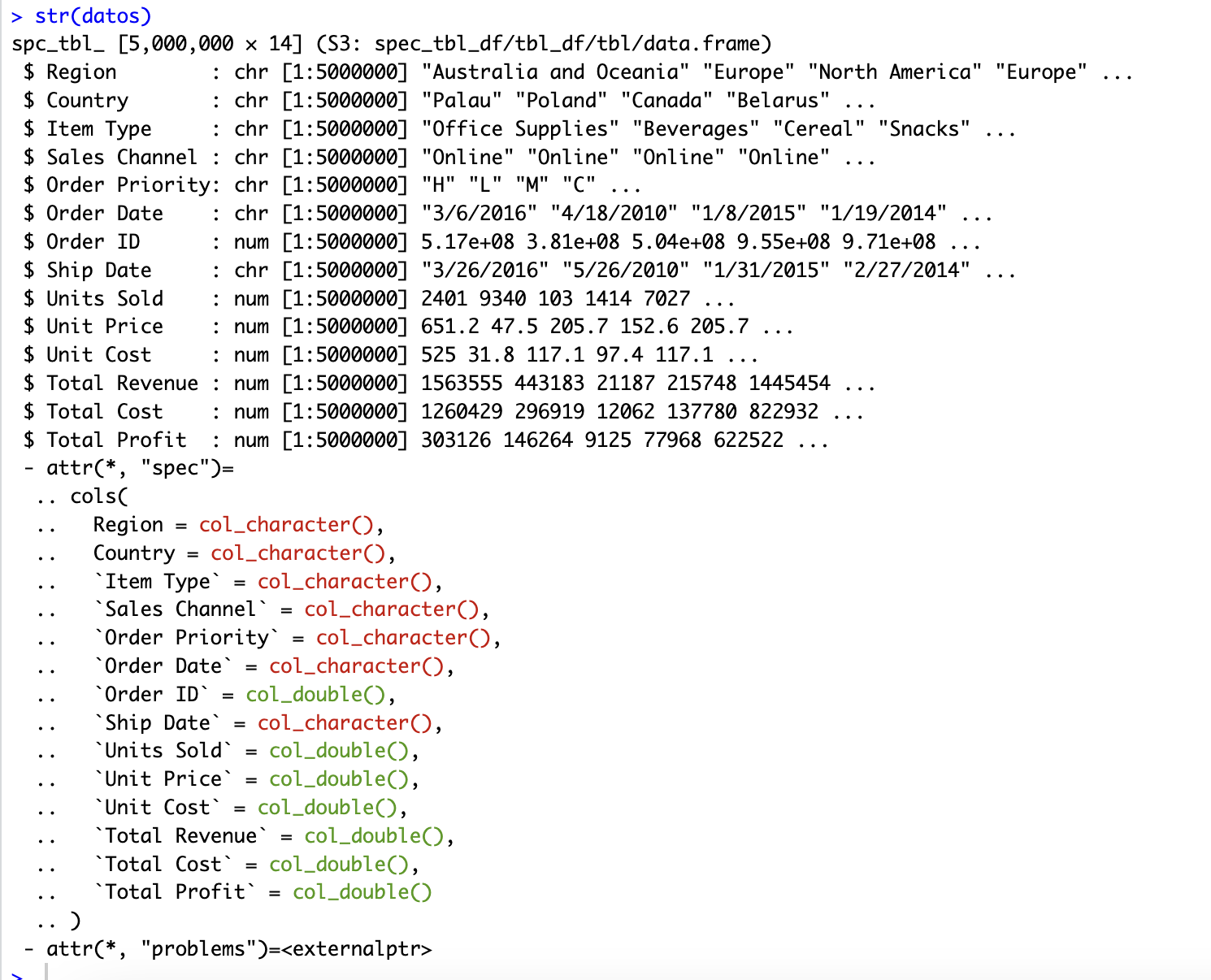
En la imagen X, se visualiza la implementación de dos códigos en R, por lo que permite conocer en breve algunos datos y los tipos de datos del dataset (por cada tupla), tal como se ve en la siguiente imagen X:  
  


Imagen X: visualizando estructura del conjunto de datos

# Análisis Exploratorio de Datos

En esta sección se da a conocer en detalle la estadística de los registros existentes en el dataset, por lo que, a continuación se visualiza dos variables cuantitativas y cualitativas, lo que se pudo obtener la información a través del siguiente código R:

|  |
| --- |
| *summary(datos)* |

El código anterior, me muestra por cada variable su análisis estadístico, pero también se implementó una función objetivo que permite manejar y mostrar (*custom)* el resumen estadístico de todas las variables cuantitativas del conjunto, a través del siguiente código R permite lo anterior:

|  |
| --- |
| # Función para calcular resúmenes numéricos de variables numéricas automáticamente  resumen\_numerico <- function(data) {  # Seleccionar solo las columnas numéricas  data\_selected <- data %>% select\_if(is.numeric)    # Calcular el resumen numérico para cada variable  resumen <- data\_selected %>%  summarise(across(everything(), list(  Min = ~ min(.),  Max = ~ max(.),  Media = ~ mean(.),  Mediana = ~ median(.),  Desviacion\_Estandar = ~ sd(.)  ), .names = "{col}\_{fn}"))    return(resumen)  } |

Al ejecutar los dos códigos de forma independiente, se visualiza resumen estadístico descriptivo de todas las variables cuantitativas, para esta oportunidad se seleccionaron dos variables cuantitativas, tal como se ve en la tabla X:

Tabla X: descripción estadística para variables cuantitativas

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Descripción Estadística | | |
| Total\_Profit | Min | 2.4 | |
| 1st. Quartil | 95145.7 | |
| Median | 281655.1 | |
| Mean | 392679.9 | |
| 3rd Quartil | 565962.3 | |
| Max | 1738700.0 | |
| Total\_Cost | Min | | 7 |
| 1st. Quartil | | 95145.7 |
| Median | | 281655.1 |
| Mean | | 392679.9 |
| 3rd Quartil | | 565962.3 |
| Max | | 1738700.0 |

En la tabla X, muestra información relevante sobre la cantidad de registros asociadas a estas variables:

* 1. Total\_Profit:
* Min: El valor mínimo de Total\_Profit es 2.4. Esto indica que la venta con la menor ganancia tuvo un beneficio de 2.4 dólares.
* 1st Quartil (Q1): El primer cuartil es 95145.7. Esto significa que el 25% de las ventas tienen un beneficio igual o menor a 95145.7 dólares.
* Median: La mediana es 281655.1. Esto significa que el 50% de las ventas tienen un beneficio igual o menor a 281655.1 dólares.
* Mean: La media es 392679.9. Esto indica que el beneficio promedio por venta es de 392679.9 unidades monetarias. La media puede verse afectada por valores atípicos (muy altos o muy bajos).
* 3rd Quartil (Q3): El tercer cuartil es 565962.3. Esto significa que el 75% de las ventas tienen un beneficio igual o menor a 565962.3 dólares.
* Max: El valor máximo de Total\_Profit es 1738700.0. Esto indica que la venta con la mayor ganancia tuvo un beneficio de 1738700.0 dólares.
  1. Sales:
* Min: El valor mínimo de Total\_Cost es 7. Esto indica que la venta con el menor costo tuvo un costo de 7 dólares.
* 1st Quartil (Q1): El primer cuartil es 95145.7. Esto significa que el 25% de las ventas tienen un costo igual o menor a 95145.7 dólares.
* Median: La mediana es 281655.1. Esto significa que el 50% de las ventas tienen un costo igual o menor a 281655.1 dólares.
* Mean: La media es 392679.9. Esto indica que el costo promedio por venta es de 392679.9 dólares.
* 3rd Quartil (Q3): El tercer cuartil es 565962.3. Esto significa que el 75% de las ventas tienen un costo igual o menor a 565962.3 dólares.
* Max: El valor máximo de Total\_Cost es 1738700.0. Esto indica que la venta con el mayor costo tuvo un costo de 1738700.0 dólares.
  1. Interpretación general
* Distribución de Datos: Las medidas de los cuartiles (Q1 y Q3) junto con la mediana nos ayudan a entender la distribución de los datos. Si la media y la mediana están muy cerca, esto sugiere que los datos están distribuidos de manera relativamente simétrica. Si están muy alejadas, puede indicar la presencia de valores atípicos.
* Comparación de Total\_Profit y Total\_Cost: Dado que las estadísticas descriptivas para Total\_Profit y Total\_Cost son idénticas en esta imagen, es probable que los costos y beneficios totales tengan una distribución similar en el conjunto de datos, lo cual puede ser poco común y debe ser revisado para posibles errores o confirmación de los datos.

Al explorar los datos del archivo CSV, podemos identificar datos vacíos o nulos y conocer los tipos de datos presentes. Sin embargo, ¿qué ocurre cuando los datos no están unidos en un archivo CSV? En la siguiente sección, cargaremos los datos en una base de datos para profundizar en el uso de R y su potencial como lenguaje de programación estadística. Para ello, aplicaremos ciertos pasos que permitan la carga de los datos en la base de datos, recordando que aún no han sido limpiados. Esto nos permitirá explorar los datos de una manera diferente, sin depender del archivo CSV, y aplicar conceptos como left join, inner join y outer join en R.

# Usando uniones en R con dplyr

En esta sección, los datos del archivo CSV no están limpios y pueden contener datos vacíos o nulos. Para efectos de la actividad, para demostrar las capacidades de R en la creación de gráficos y funciones, los datos se han preparado en dataframes para ser leídos y transformados, permitiendo así la transferencia de información a una base de datos MySQL. Sin embargo, antes de proceder con la carga, es necesario estructurar la base de datos a partir del archivo CSV, como se muestra en la Imagen X:

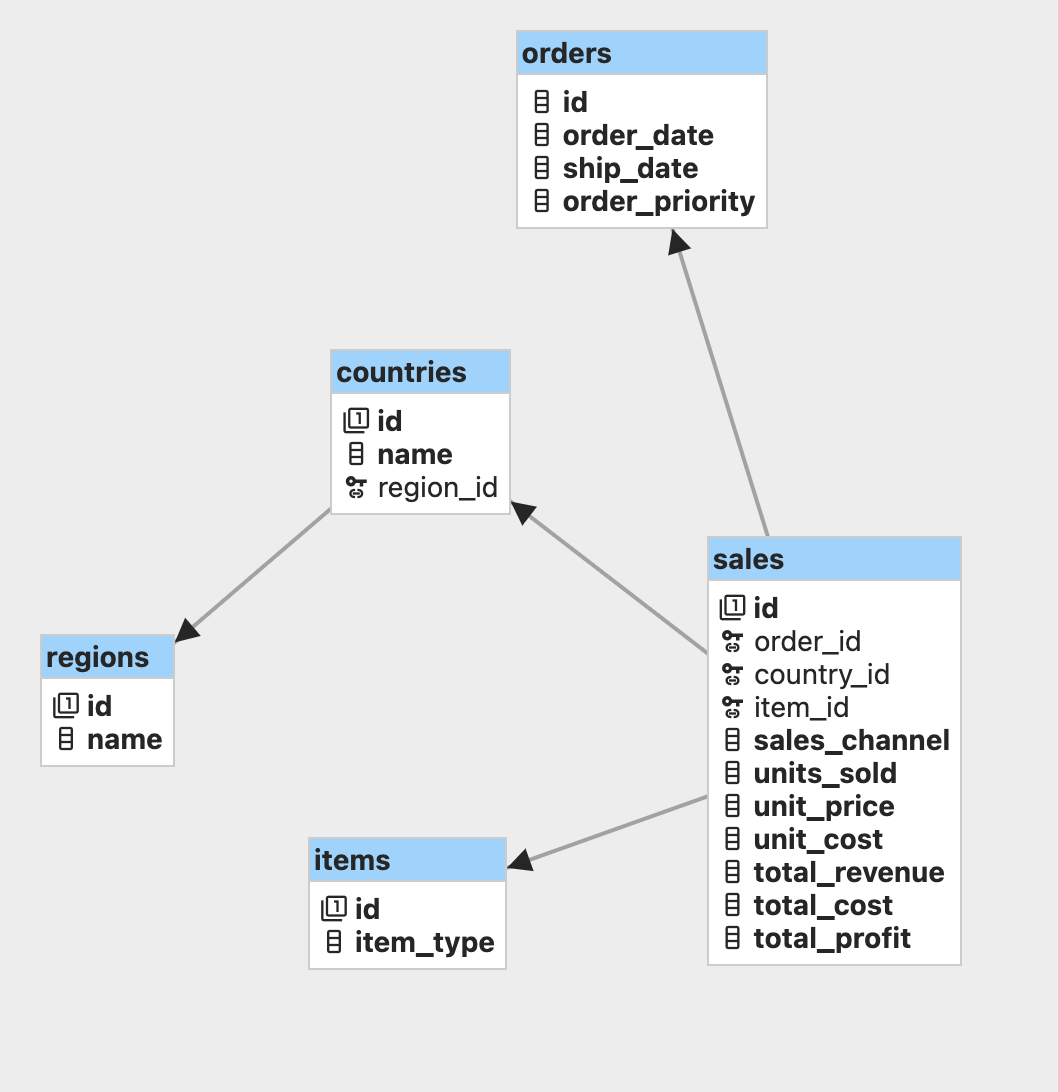


Imagen X: modelo físico del conjunto de datos

En la imagen anterior, se visualiza cinco tablas, del cual permitirá distribuir los datos de mejor forma para su posterior uso que se le quiera dar, por ejemplo: crear una API que permita realizar consultas a la base de datos, enviar correos a través de esta API para informar al cliente o enviar informes mensuales al gerente. Bajo esta misma línea, para proceder la carga de datos (usando el archivo csv) hacia la base de datos, se ha uso las siguientes librerías y códigos en R:

|  |
| --- |
| # Instalar las librerías si no están ya instaladas  install.packages("DBI")  install.packages("RMySQL")  install.packages("readr")  # Cargar las librerías  library(DBI)  library(RMySQL)  library(readr) |

Además, para proceder con la carga de datos, se usa los siguientes códigos en R:

|  |
| --- |
| # Conectar a la base de datos MySQL  con <- dbConnect(RMySQL::MySQL(), dbname = "sales", host = "127.0.0.1", port = 3306, user = "root", password = "")  # Leer el archivo CSV  data <- read\_csv("./data/5m\_sales.csv")  data <- data.frame(  region = data$Region,  country = data$Country,  item\_type = data$`Item Type`,  sales\_channel = data$`Sales Channel`,  order\_priority = data$`Order Priority`,  order\_date = as.Date(data$`Order Date`, format="%m/%d/%Y"),  order\_id = data$`Order ID`,  ship\_date = as.Date(data$`Ship Date`, format="%m/%d/%Y"),  units\_sold = data$`Units Sold`,  unit\_price = data$`Unit Price`,  unit\_cost = data$`Unit Cost`,  total\_revenue = data$`Total Revenue`,  total\_cost = data$`Total Cost`,  total\_profit = data$`Total Profit`  )  # Insertar datos en la tabla `regions`  regions <- unique(data.frame(name = data$region))  dbWriteTable(con, "regions", regions, append = TRUE, row.names = FALSE)  # Obtener el id de las regiones  regions\_ids <- dbReadTable(con, "regions")  # Insertar datos en la tabla `countries`  data$region\_id <- sapply(data$region, function(x) regions\_ids$id[regions\_ids$name == x])  countries <- unique(data.frame(name = data$country, region\_id = data$region\_id))  dbWriteTable(con, "countries", countries, append = TRUE, row.names = FALSE)  # Obtener el id de los países  countries\_ids <- dbReadTable(con, "countries")  # Insertar datos en la tabla `items`  items <- unique(data.frame(item\_type = data$item\_type))  dbWriteTable(con, "items", items, append = TRUE, row.names = FALSE)  # Obtener el id de los tipos de artículos  items\_ids <- dbReadTable(con, "items")  # Insertar datos en la tabla `orders`  orders <- unique(data.frame(id = data$order\_id, order\_date = data$order\_date, ship\_date = data$ship\_date, order\_priority = data$order\_priority))  dbWriteTable(con, "orders", orders, append = TRUE, row.names = FALSE)  # Insertar datos en la tabla `sales`  data$country\_id <- sapply(data$country, function(x) countries\_ids$id[countries\_ids$name == x])  data$item\_id <- sapply(data$item\_type, function(x) items\_ids$id[items\_ids$item\_type == x])  sales <- data.frame(  order\_id = data$order\_id,  country\_id = data$country\_id,  item\_id = data$item\_id,  sales\_channel = data$sales\_channel,  units\_sold = data$units\_sold,  unit\_price = data$unit\_price,  unit\_cost = data$unit\_cost,  total\_revenue = data$total\_revenue,  total\_cost = data$total\_cost,  total\_profit = data$total\_profit  )  dbWriteTable(con, "sales", sales, append = TRUE, row.names = FALSE) |

Algunas de las funciones y/ funcionalidades que tiene R para trabajar con base de datos Mysql, es el uso del dataframe, importante para tratar los datos y para usarlo en la función dbWriteTable() para realizar registro hacia la tabla que uno desee guardar los datos.

Para esto, unos de los códigos para aplicar en R, que permita unir datos, son los siguientes:

* 1. Inner Join en R

Al usar inner join en R, devuelve solo registros que tienen coincidencia en ambas tablas.

|  |
| --- |
| # Realizar INNER JOIN  data\_inner\_join <- sales\_data %>%  inner\_join(orders\_data, by = c("order\_id" = "id")) %>%  inner\_join(countries\_data, by = c("country\_id" = "id")) %>%  inner\_join(items\_data, by = c("item\_id" = "id")) %>%  inner\_join(regions\_data, by = c("region\_id" = "id"))  # Convertir order\_date a clase Date  data\_inner\_join$order\_date <- as.Date(data\_inner\_join$order\_date)  # Procesar datos para INNER JOIN  monthly\_data\_inner <- data\_inner\_join %>%  mutate(month = floor\_date(order\_date, "month")) %>%  group\_by(month) %>%  summarise(total\_units\_sold = sum(units\_sold, na.rm = TRUE),  total\_cost = sum(total\_cost, na.rm = TRUE))  # Crear gráfico de líneas de unidades vendidas totales por mes usando INNER JOIN  crear\_grafico(monthly\_data\_inner, "lineas", "month", "total\_units\_sold")  # Crear gráfico de líneas de costo total por mes usando INNER JOIN  crear\_grafico(monthly\_data\_inner, "barras", "month", "total\_cost") |

Esto es útil cuando se quiere obtener registros que tienen datos completos en ambas tablas relacionadas.

* 1. Left Join en R

Al usar left join, devuelve registros del conjunto o tabla izquierda y las filas coincidentes de la tabla de la derecha.

|  |
| --- |
| # Realizar LEFT JOIN  data\_left\_join <- sales\_data %>%  left\_join(orders\_data, by = c("order\_id" = "id")) %>%  left\_join(countries\_data, by = c("country\_id" = "id")) %>%  left\_join(items\_data, by = c("item\_id" = "id")) %>%  left\_join(regions\_data, by = c("region\_id" = "id"))  # Convertir order\_date a clase Date  data\_left\_join$order\_date <- as.Date(data\_left\_join$order\_date)  # Procesar datos para LEFT JOIN  annual\_data\_left <- data\_left\_join %>%  mutate(year = year(order\_date)) %>%  group\_by(year) %>%  summarise(total\_units\_sold = sum(units\_sold, na.rm = TRUE),  total\_cost = sum(total\_cost, na.rm = TRUE))  # Crear gráfico de líneas de unidades vendidas totales por año usando LEFT JOIN  crear\_grafico(annual\_data\_left, "lineas", "year", "total\_units\_sold")  # Crear gráfico de líneas de costo total por año usando LEFT JOIN  crear\_grafico(annual\_data\_left, "lineas", "year", "total\_cost") |

Si no hay coincidencia, los valores de la tabla de la derecha serán “*Null”* o Vacías. Por otro lado, esto es útil cuando se quiere obtener registros de la tabla de la izquierda, independientemente de si hay coincidencias en la tabla de la derecha.

* 1. Right join en R

Al usar right join, devuelve registros del conjunto o tabla de la derecha y las filas coincidentes del conjunto izquierdo o tabla izquierda.

|  |
| --- |
| # Realizar RIGHT JOIN  data\_right\_join <- sales\_data %>%  right\_join(orders\_data, by = c("order\_id" = "id")) %>%  right\_join(countries\_data, by = c("country\_id" = "id")) %>%  right\_join(items\_data, by = c("item\_id" = "id")) %>%  right\_join(regions\_data, by = c("region\_id" = "id"))  # Convertir order\_date a clase Date  data\_right\_join$order\_date <- as.Date(data\_right\_join$order\_date)  # Procesar datos para RIGHT JOIN  annual\_data\_right <- data\_right\_join %>%  mutate(year = year(order\_date)) %>%  group\_by(year) %>%  summarise(total\_units\_sold = sum(units\_sold, na.rm = TRUE),  total\_cost = sum(total\_cost, na.rm = TRUE))  # Crear gráfico de líneas de unidades vendidas totales por año usando RIGHT JOIN  crear\_grafico(annual\_data\_right, "lineas", "year", "total\_units\_sold")  # Crear gráfico de líneas de costo total por año usando RIGHT JOIN  crear\_grafico(annual\_data\_right, "lineas", "year", "total\_cost") |

Si no hay coincidencia, los valores del conjunto izquierdo o tabla izquierda serán “*null”*, sin embargo esto es útil cuando se quiere obtener registros del conjunto derecho o tabla derecha, independiente si hay coincidencia del conjunto izquierdo.

# Interpretación de Resultados

En esta sección se visualiza los gráficos que son parte del EDA, en particular se procedió a realizar los gráficos usando R, pero

* 1. Histograma de la variable *Total\_Profit*

En el siguiente gráfico consta del eje X (Total\_Profit) que representa los valores de las ganancias en dólares ($) y el eje Y (Count) representa la frecuencia o el conteo de las ventas que tienen un valor especifico de Profit, por ende la altura de cada barra indica cuantas veces ocurre un rango particular (sesgo) de ganancias en el conjunto de datos, tal como se puede ver en la imagen X:

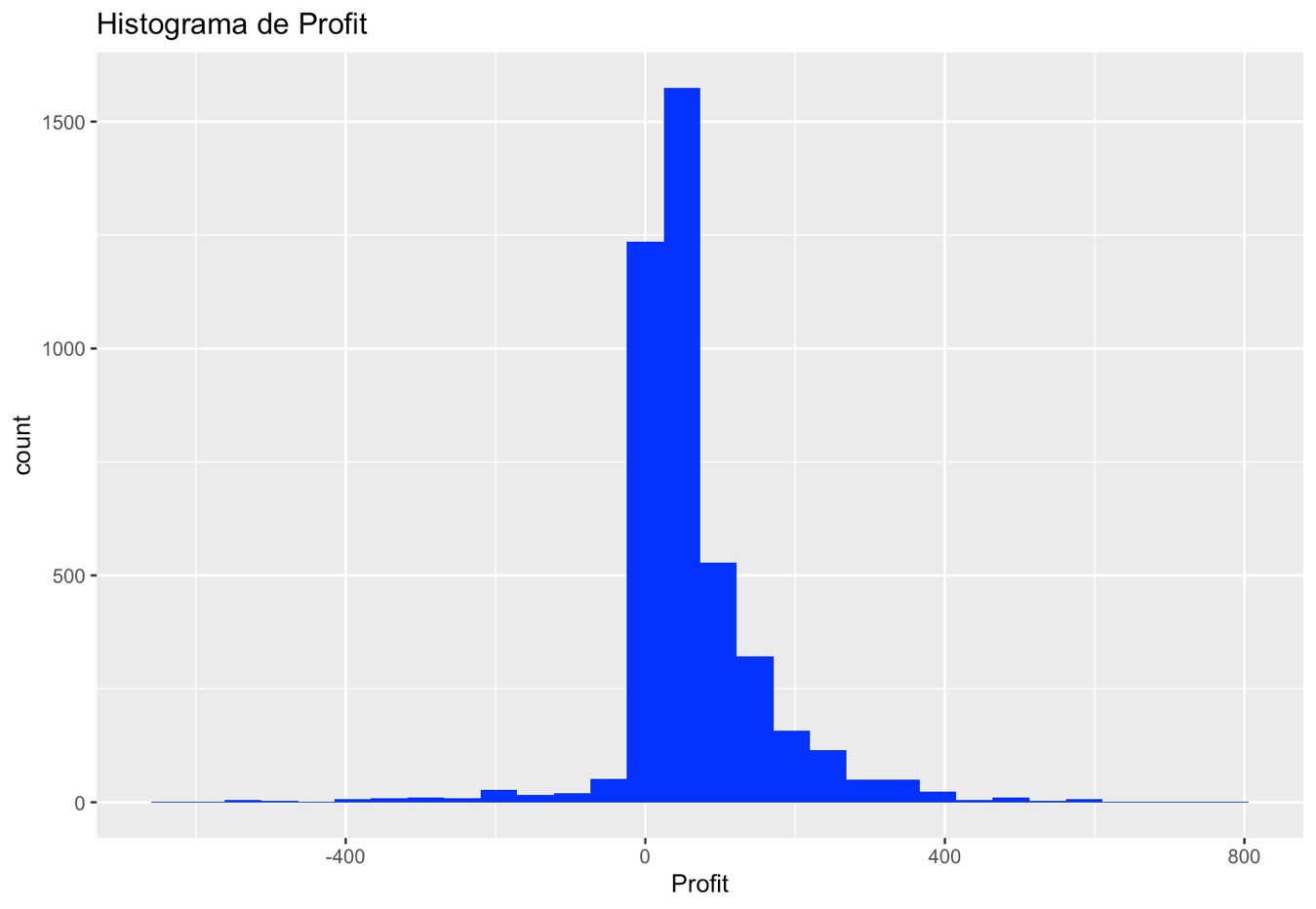


Imagen X: histograma de la variable profit

En el gráfico anterior, la mayoría de las ventas tienen ganancias bajas o cercano a cero, este sesgo de las ventas puede indicar que la empresa experimenta tanto perdidas como ganancias, sin embargo, esta distribución sesgada muestra transacciones rentables para el tipo de producto vendido.

* 1. Diagrama de cajas de *Sales*

El gráfico de cajas permite ver la dispersión de datos y validar la información descriptiva estadística de una variable numérica (Sales), esto se visualiza en la siguiente Imagen X:

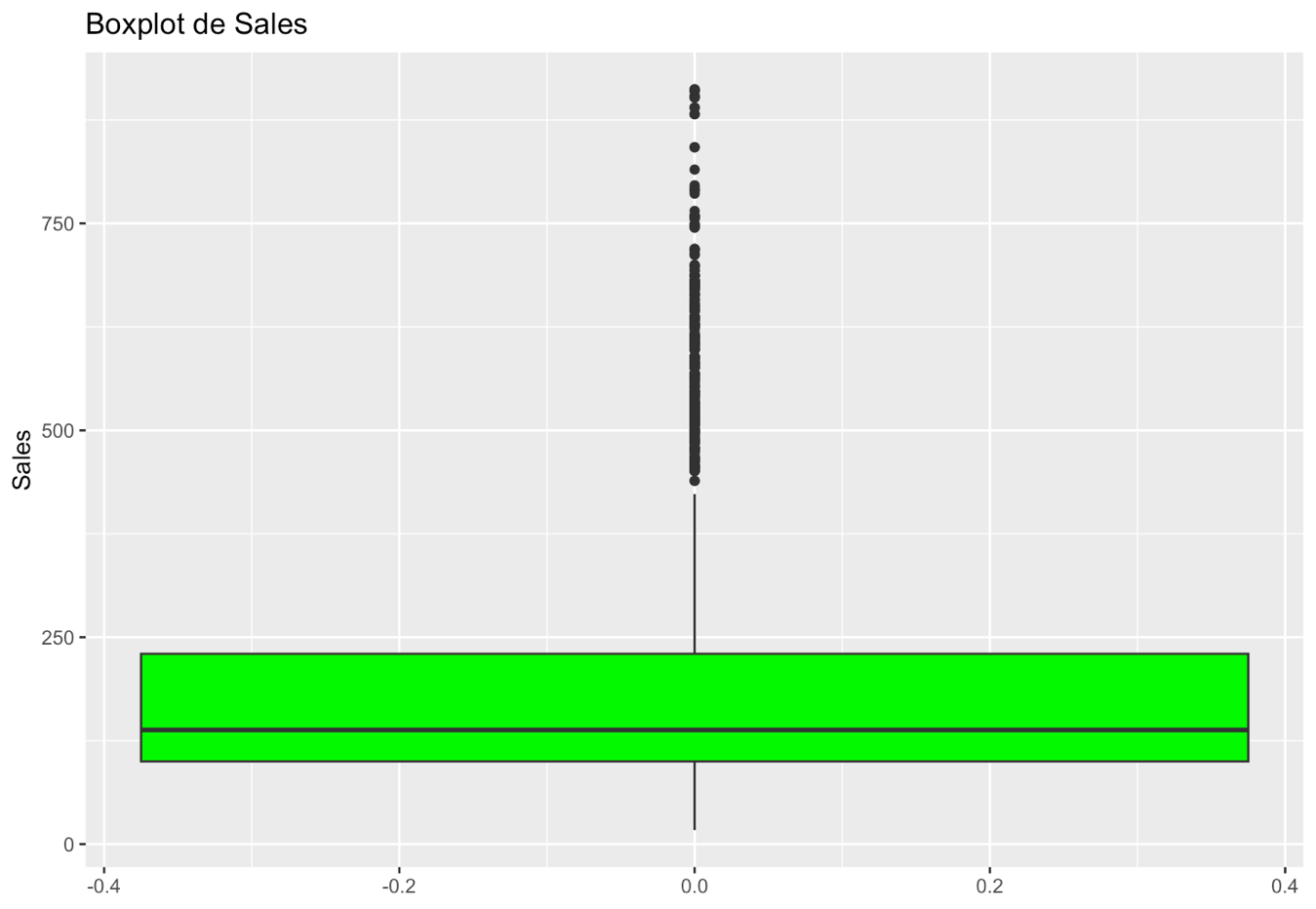
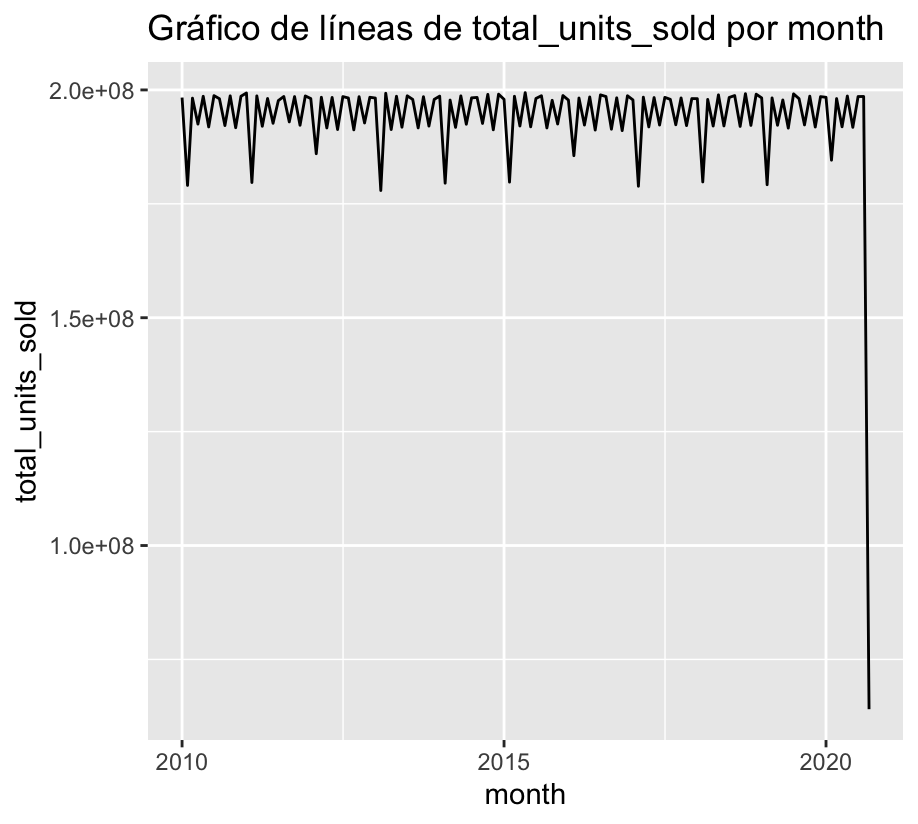


Imagen X: diagrama de cajas de la variable Sales.

En la imagen anterior se visualiza la distribución donde la mayoría de las ventas están concentradas en un rango de 100 a 230 dólares, con una mediana alrededor de 140 dólares, además existe una pronunciada variabilidad en las ventas como lo indica el amplio IQR (rango intercualictico).

Por otro lado, la visualización de muchos outliers indica que, por encima del tercer cuartil existen algunas ventas muy altas que no son comunes y pueden estar asociado a la media de las ventas.

* 1. Gráfico de Líneas de Ventas por Mes



* 1. Análisis complejo por tipo variable cuantitativa

# Anexos

* 1. Código R

# importación de librerias

library(tidyverse)

library(ggplot2)

#0 Definición del problema/objetivo de investigación (ficticia).

# 0.1 La empresa "IquiqueMiami LTDA." ha observado fluctuaciones significativas en sus ingresos durante

# el último año fiscal. A pesar de tener una amplia variedad de productos y presencia en varias estados

# del país de Estados Unidos, la empresa ha enfrentado desafíos debido a la alta competitividad en el

# mercado y cambios en las preferencias de los consumidores. La dirección ha identificado que una

# estrategia de precios ineficaz podría ser una de las causas principales de estos resultados

# inconsistentes.

# 1. Introducción: Descripción del conjunto de datos seleccionado y los objetivos del análisis (ficticia).

# En los últimos años, el movimiento económico de empresas pequeñas enfrenta desafíos constantes

# debido a la competencia, las cambiantes preferencias de los consumidores o clientes, y la rápida

# evaluación de la tecnología. “Iquique Miami LTDA.”, una empresa en el sector minorista, se ha

# comprometido a mejorar sus ventas y la satisfacción del cliente mediante la implementación de un

# enfoque basado en datos.

# 1.1. Diccionario de datos

# Area Code: Store's Code;

# State: Store's State;

# Market: Store's Region;

# Market Size: Store's Size;

# Profit: Profits in Dollars ($);

# Margin: Profit + Total Expenses ($) OR Sales - COGS ($);

# Sales: Values Acquired in Sales ($);

# COGS: Cost of Goods Sold ($);

# Total Expenses: Total Expenses to get the Product to Sell ($);

# Marketing: Expenses in Marketing ($);

# Inventory: Inventory Value of the Product in the Sale Moment ($);

# Budget Profit: Expected Profit ($);

# Budget COGS: Expected COGS ($);

# Budget Margin: Expected Profit + Expected Total Expenses ($) OR Expected Sales - Expected COGS ($);

# Budget Sales: Expected Value Acquired in Sales ($);

# ProductID: Product ID;

# Date: Sale Date;

# Product Type: Product Category;

# Product: Product Description;

# Type: Type;

#####1.2. DATOS CUANTITATIVOS

####

# Area code: Entero

# Profit: Entero

# Margin: Entero

# Sales: Entero

# COGS: Entero

# Total Expenses: Entero

# Marketing: Entero

# Inventory: Entero

# Budget Profit: Entero

# Budget COGS: Entero

# Budget Margin: Entero

# Budget Sales: Entero

# ProductId: Entero

###

#####1.3. DATOS CUALITATIVOS

####

#State: Cadena

#Market: Cadena

#Market Size: Cadena

#Product Type: Cadena

#Product: Cadena

####

# 2. Preprocesamiento de Datos: Descripción detallada de los pasos tomados para limpiar y preparar los datos para el análisis utilizando Tidyverse.

datos <- read\_delim("sales.csv", delim = ",") # Usar read\_delim con el delimitador especificado

# Inspeccionamos las primeras filas de los datos

head(datos)

# Resumen de estructura de datos (imprime algunos datos y tipos de datos)

glimpse(datos)

# Inspeccionar y limpiar los datos si es necesario (imprime la estructura y los tipos de datos)

str(datos)

# 3. Análisis Exploratorio de Datos: Implementación de códigos en R para explorar los datos. Esto puede incluir visualizaciones de datos, medidas de tendencia central, etc. importación de datos

# 3.1. resumen general/total estadísticos de todos los datos presentes en el dataset

summary(datos)

# 3.2. Comprobar si hay valores faltantes en las columnas importantes

na\_count <- sapply(datos, function(x) sum(is.na(x)))

na\_count

# 3.3. Comprobar si hay valores faltantes en las columnas importantes seleccionadas que exige el informe.

sum(is.na(datos))

sum(is.na(datos$Profit))

sum(is.na(datos$Sales))

sum(is.na(datos$COGS))

# 3.4. Histograma de Profit

ggplot(datos, aes(x = Profit)) +

geom\_histogram(bins = 30, fill = "blue") +

ggtitle("Histograma de Profit")

# 3.5. Boxplot de Sales

ggplot(datos, aes(y = Sales)) +

geom\_boxplot(fill = "green") +

ggtitle("Boxplot de Sales")

# 3.6. Boxplot de Total Expenses por Market

ggplot(datos, aes(x = Market, y = `Total Expenses`)) +

geom\_boxplot(fill = "cyan", color = "black") +

ggtitle("Boxplot de Total Expenses por Market") +

xlab("Market") +

ylab("Total Expenses")

# 4 Preprocesamiento de Datos (formateo de solamente fecha, ya que la fecha incluye horas, minutos y segundos)

datos <- datos %>%

mutate(fecha = as.Date(fecha, format = "%d-%m-%Y")) %>% # Convertir la columna 'fecha' al formato de fecha (no incluye hh:mm:ss)

filter(!is.na(precio)) %>% # Eliminar filas donde el precio es NA

arrange(desc(precio)) # Ordenar los datos por precio de forma descendente