Informe de Análisis de Ventas en US.

Programación en R

Nombre: Víctor Godoy Roa

Fecha de entrega: XX/XX/2024

Docente: Jaime Lincovil Curivil

Índice

Introducción

Este informe presenta un análisis de las transacciones de venta de “IquiqueMiami LTDA.”, una firma que ha marcado presencia en varios estados de Estados Unidos, enfrentando fluctuaciones en sus ingresos durante el último año fiscal. A través de un detallado estudio del conjunto de datos de ventas, el informe se enfoca en discernir patrones y tendencias que podrían esclarecer la raíz de los desafíos enfrentados, atribuidos a la alta competitividad del mercado y a cambios en las preferencias de los consumidores.

Utilizando la herramienta de programación en R, el informe procede a desglosar los datos de ventas, aplicando técnicas de preprocesamiento y análisis exploratorio para luego, mediante modelado estadístico, generar proyecciones y recomendaciones estratégicas. Este enfoque no solo permite una comprensión más profunda de las dinámicas de mercado actuales sino que también orienta en la toma de decisiones estratégicas para fortalecer la posición de la empresa en el mercado.

El análisis se complementa con visualizaciones gráficas y modelos predictivos que buscan potenciar la capacidad de la empresa para anticipar cambios en el mercado y ajustar su estrategia de precios y promociones de manera proactiva. Al final del informe, se discuten los resultados obtenidos y se proponen pasos concretos para la mejora continua en las operaciones de “IquiqueMiami LTDA.”, asegurando su crecimiento sostenible y eficiencia operativa.

1. Definición del problema/objetivo de investigación

1.1. Problemática:

La empresa IquiqueMiami LTDA, ha experimentado fluctuaciones en sus ingresos durante los últimos años, a pesar de su diversa variedad de productos y su amplia presencia en varios estados de Estados Unidos, lo que su competitividad en el mercado han resaltado la necesidad de revisar la estrategia de precios, identificada por la dirección como un factor clave detrás de los resultados inconsistentes.

1.2. Objetivos:

Los objetivos de este análisis y estudio están diseñados para responder de manera efectiva al modelo de negocio implementando por la empresa, evaluando tanto impactos positivos como negativos. Es relevante analizar detalladamente las ventas para evitar repetir errores anteriores. El equipo de analistas de datos realiza un estudio del conjunto de datos para identificar patrones y tendencias que opotimicen las estrategias comerciales, proporcionando insights para una toma de decisiones informada y mejorando la eficiencia operativa y el crecimiento sostenible de IquiqueMiami LTDA.

2. Ventas en US.

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 del pasado.

2.1. 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/* 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 1: 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), es suficiente para leer y analizar el encabezado del conjunto de datos para el desarrollo del diccionario de datos.

2.1.1. 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 2:

Tabla 2: 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

3. 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 Figura 1:

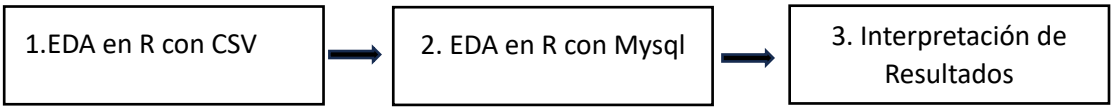

Figura 1: flujo de trabajo

Dada la figura anterior, en el primer paso se trabajará directamente con el archivo CSV, realizando operaciones básicas estadísticas descriptivas. En el segundo paso, se llevará a cabo un análisis descriptivo cargando y utilizando los datos en una base de datos, con el objetivo de demostrar y probar las funcionalidades de otra manera que ofrece R para los analistas. Este proceso complementará los conocimientos impartidos en clases, aplicando librerías y la teoría básica de probabilidad y estadística.

Para comenzar con el proceso EDA, 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 el siguiente código:

```
# Quitar espacios de los nombres de las columnas
datos <- quitar_espacios_nombres(datos)
head(datos)
str(datos)
# Validar si existen valores nulos o vacíos
validar_datos(datos)
# Inspección de datos
glimpse(datos)
```

En los códigos anteriores permite conocer en breve algunos datos y los tipos de datos del conjunto de datos (por cada tupla), tal como se ve en la siguiente imagen 1:

```
> str(datos)
spc_tbl_ [5,000,000 × 14] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ Region        : chr [1:5000000] "Australia and Oceania" "Europe" "North America" "Europe" ...
 $ Country       : chr [1:5000000] "Palau" "Poland" "Canada" "Belarus" ...
 $ Item Type     : chr [1:5000000] "Office Supplies" "Beverages" "Cereal" "Snacks" ...
 $ Sales Channel : chr [1:5000000] "Online" "Online" "Online" "Online" ...
 $ Order Priority: chr [1:5000000] "H" "L" "M" "C" ...
 $ Order Date    : chr [1:5000000] "3/6/2016" "4/18/2010" "1/8/2015" "1/19/2014" ...
 $ Order ID      : num [1:5000000] 5.17e+08 3.81e+08 5.04e+08 9.55e+08 9.71e+08 ...
 $ Ship Date     : chr [1:5000000] "3/26/2016" "5/26/2010" "1/31/2015" "2/27/2014" ...
 $ Units Sold    : num [1:5000000] 2401 9340 103 1414 7027 ...
 $ Unit Price    : num [1:5000000] 651.2 47.5 205.7 152.6 205.7 ...
 $ Unit Cost     : num [1:5000000] 525 31.8 117.1 97.4 117.1 ...
 $ Total Revenue : num [1:5000000] 1563555 443183 21187 215748 1445454 ...
 $ Total Cost    : num [1:5000000] 1260429 296919 12062 137780 822932 ...
 $ Total Profit  : num [1:5000000] 303126 146264 9125 77968 622522 ...
 - attr(*, "spec")=
  .. cols(
  ..   Region = col_character(),
  ..   Country = col_character(),
  ..   `Item Type` = col_character(),
  ..   `Sales Channel` = col_character(),
  ..   `Order Priority` = col_character(),
  ..   `Order Date` = col_character(),
  ..   `Order ID` = col_double(),
  ..   `Ship Date` = col_character(),
  ..   `Units Sold` = col_double(),
  ..   `Unit Price` = col_double(),
  ..   `Unit Cost` = col_double(),
  ..   `Total Revenue` = col_double(),
  ..   `Total Cost` = col_double(),
  ..   `Total Profit` = col_double()
  .. )
 - attr(*, "problems")=<externalptr>
>
```

Imagen 1: visualizando estructura del conjunto de datos

En la imagen anterior, entrega información de manera estructurada, pero no tan detallada y limpia, sin embargo permite al analista entender con qué tipo de datos va a trabajar durante el desarrollo de la actividad.

4. Análisis Exploratorio de Datos

En esta sección se da a conocer en detalle la estadística de los registros existentes en el conjunto de datos, 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 (de manera personalizable*)* el resumen estadístico de todas las variables cuantitativas del conjunto (Anexo 7.1.1).

Al ejecutar los 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 3:

Tabla 3: 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 3, muestra información relevante sobre la cantidad de registros asociadas a estas variables:

4.1. 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.

4.2. 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.

4.3. 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.

En R, existen diversas formas de calcular y visualizar los datos durante el proceso de EDA. Una alternativa amigable que cumple con estos objetivos es el siguiente código:

```
skim(datos)
```

Al ejecutar el código R, muestra el siguiente resultado amigable, tal como se ve en la siguiente Tabla 4 y Tabla 5:

Tabla 4: Variables de tipo categórica

Variable	Var. Missing	Rate	Mínimo	Máximo	Vacíos	Valores únicos	Espacios en blanco
Region	0	1	4	33	0	7	0
Country	0	1	4	32	0	185	0
Item_type	0	1	1	15	0	12	0
Sales_Channel	0	1	1	7	0	2	0
Order_Priority	0	1	8	1	0	4	0
Order_Date	0	1	8	10	0	3906	0
Ship_Date	0	1	8	10	0	3956	0

En base a la Tabla 4, se deduce lo siguiente:

- **Region**: la variable tiene 7 valores unicis diferentes. Esto implica que las ventas se realizaron en 7 regiones distintas. No hay valores perdidos ni espacios en blanco.
- **Country**: la variable tiene 185 valores únicos, lo que indica una alta diversidad en los lugares donde se realizaron las ventas. Tampoco hay valores perdidos ni espacios en blanco.
- **Item Type**: la variable tiene 12 tipos únicos de artículos. No se presentan valores perdidos ni espacios en blanco.
- **Sales Channel**: la variable tiene 2 canales de ventas únicos (probablemente “Online” y “Offline”), sin valores perdidos ni espacios en blanco.
- **Order Priority**: la variable tiene 4 niveles diferentes de prioridad para los pedidos. No se observan valores perdidos ni espacios en blanco.
- **Order Date**: la variable tiene 3906 fechas únicas de pedidos, lo que sugiere un largo período de recolección de datos. No hay valores perdidos ni espacios en blanco.
- **Ship Date**: Similar a las fechas de pedidos, la variable tiene 3956 fechas únicas de envío. No hay valores perdidos ni espacios en blanco.

Tabla 5: Variables de tipo numérico

Var.	Var.Perdidos	Rate	Media	Std	Min.	P25	P75	P100
Units_Sold	0	1	5000.5	2887.0	1	2500	7500	10000
Unit_Price	0	1	266.27	217.6	0.29	109	437	668
Unit_Cost	0	1	188.2	176.0	6.92	56.7	365	525
Total_Revenue	0	1	1331958	1469902	0.93	277964	1822444	6682700
Total_Cost	0	1	938378	1150104	0.62	161925	11974434	5249600
Total_Profit	0	1	392680	379117	2.41	91564	281655	1738700

En base a la Tabla 5, se deduce lo siguiente:

- **Units Sold**: la variable varía de 1 a 10000, con una media de 5000.5 y una desviación estándar de 2887. Esto indica una alta variabilidad en las unidades vendidas por transacción.
- **Unit Price**: la variable varía desde $0.29 a $668, con una media de $266.27 y una desviación estándar de $217.6. Esto muestra que hay una amplia gama de precios entre los artículos vendidos.
- **Unit Cost**: la variable varía de $6.92 a $525, con una media de $188.2 y una desviación estándar de $176. Esto también indica una considerable variabilidad en los costos unitarios de los artículos vendidos.
- **Total Revenue**: la variable varían de $0.93 a $6682700, con una media de $13319508 y una desviación estándar de $1469902. Los ingresos muestran una gran dispersión, sugiriendo que algunas ventas son significativamente más altas que otras.
- **Total Cost**: la variable varía desde $0.62 a $5249600, con una media de $938378 y una desviación estándar de $1150104. Esto muestra que los costos también tienen una amplia dispersión, similar a los ingresos.
- **Total Profit**: El beneficio total varía de $2.41 a $1738700, con una media de $392680 y una desviación estándar de $379117. Esto indica que las ganancias pueden ser altamente variables, con algunas transacciones mucho más lucrativas que otras.

El análisis de las variables categóricas y numéricas revela una diversidad geográfica y de productos en las ventas, así como una considerable variabilidad en precios y costos. Los ingresos y beneficios (*profit)* muestran una gran dispersión, indicando transacciones con altos y bajos valores.

4.4. Análisis de Correlación

Antes de proceder con el modelado de datos, es útil entender como las diferentes variables numéricas están relacionadas entre sí, para esto es necesario aplicar el siguiente código R:

```
# Calcular la matriz de correlación
correlaciones <- cor(datos %>% select_if(is.numeric))
# Visualizar la matriz de correlación usando ggplot2
library(corrplot)
corrplot(correlaciones, method = "circle")
```

Este código realiza dos tareas principales:

- Calcula la matriz de correlación para todas las variables numéricas en el conjunto de datos. La función cor() de R proporciona un coeficiente de correlación para cada par de variables, lo que nos ayuda a identificar la fuerza y dirección de la relación entre ellas.
- Visualiza la matriz de correlación utilizando la función corrplot() del paquete corrplot. Este método gráfico nos permite observar rápidamente las relaciones entre todas las variables numéricas, con círculos cuyo tamaño y color varían según la fuerza de la correlación. Los círculos más grandes y más oscuros indican una correlación más fuerte, ya sea positiva (azul) o negativa (rojo).

Por otro lado, también se puede analizar correlaciones por regiones a través de un mapa de calor, tal como se ve en el siguiente código R:

```
# Convertir la matriz de correlación en un data frame
cor_data <- as.data.frame(correlaciones)
cor_data$variable <- rownames(cor_data)
# Transformar los datos en formato largo
```

```
cor_data_long <- pivot_longer(cor_data, cols = -variable, names_to = "Var2", values_to = "value")
```

5. Modelado de datos

El modelado de datos es un proceso crítico en el análisis de datos que implica la creación de modelos matemáticos para interpretar y predecir comportamientos de variables en base a datos históricos. Este capítulo describe las técnicas y metodologías empleadas para desarrollar modelos estadísticos con el fin de proporcionar una comprensión más profunda de los patrones subyacentes en los datos.

5.1. Modelos Estadísticos

Los modelos estadísticos son fundamentales para analizar relaciones entre variables y para la inferencia. En este proyecto, utilizamos:

- Regresión Lineal: Para predecir variables continuas. Por ejemplo, empleamos la regresión lineal para prever el Total_Revenue a partir de Units_Sold y Unit_Price. Este modelo nos ayuda a entender el impacto directo de las unidades vendidas y el precio unitario sobre los ingresos totales.
- Regresión Logística: Utilizada para clasificar datos en categorías. En nuestro caso, se utiliza para determinar si los ingresos de una venta son altos o bajos, basados en la mediana del conjunto de datos. Este modelo es particularmente útil para decisiones binarias y para evaluar la probabilidad de eventos categorizados.

5.2. Regresión lineal

En este capítulo, exploraremos cómo el número de unidades vendidas (Units_Sold) y el precio unitario (Unit_Price) afectan a los ingresos totales (Total_Revenue) a través de un modelo de regresión lineal.

La fórmula general de un modelo de regresión lineal múltiple es:

$$Y = \beta_0 + \beta_1 \mathrm{X}_1 + \beta_2 \mathrm{X}_2 + \beta_3 \mathrm{X}_3 + \ldots + \beta_\rho X_\rho + \epsilon$$

Donde:

- Y es la variable dependiente a predecir.
- β_0 es la intersección (valor de Y cuando todas las variables independientes son 0).

- $\beta_1\beta_2\beta_3$ son los coeficientes de las variables independientes, además $X_1X_2X_3$ (variables numéricas a usar)
- ϵ es el término de error que representa la variabilidad en Y que no puede ser explicada por los X.

Dicho lo anterior, se aplica en el siguiente código R:

```
linear_model <- lm(Total_Revenue ~ Units_Sold + Unit_Price, data = data)
```

Este modelo fue implementado para predecir simultáneamente tanto los ingresos como los beneficios a partir de múltiples predictores.

5.2.1. Evaluación del modelo

En esta sección, se evalúa el modelo de regresión lineal desarrollado para predecir el Total_Revenue a partir de variables como Units_Sold y Unit_Price. La evaluación se centra tanto en el análisis estadístico como en los diagnósticos visuales para asegurar la validez del modelo.

```
par(mfrow = c(2, 2))
# Gráfico de Residuos vs Valores Ajustados
plot(linear_model$fitted.values, residuals(linear_model),
     xlab = "Valores Ajustados", ylab = "Residuos",
     main = "Residuos vs Valores Ajustados")

# Gráfico Q-Q de los residuos para verificar la normalidad
qqnorm(residuals(linear_model))
qqline(residuals(linear_model), col = "red")

# Gráfico de Escala-Localización (Scale-Location Plot)
plot(linear_model$fitted.values, sqrt(abs(residuals(linear_model))),
     xlab = "Valores Ajustados", ylab = "Raíz Cuadrada de los Residuos Absolutos",
     main = "Escala-Localización")

# Gráfico de Distancias de Cook para identificar influencias atípicas
plot(linear_model, which = 4, main = "Distancias de Cook")
```

En el proceso de evaluación de modelos de regresión lineal, la implementación de diagnósticos visuales es fundamental para mantener la solidez del modelo. El gráfico de Residuos vs Valores Ajustados es para evaluar si la varianza de los residuos es constante, un concepto conocido como homoscedasticidad. Un patrón aleatorio de residuos dispersos a lo largo del eje horizontal indica que el modelo satisface este supuesto. En cambio, la aparición de patrones en forma de embudo o curvas señala heteroscedasticidad, lo que puede llevar a considerar la transformación de datos o la aplicación de modelos alternativos.

El gráfico Q-Q de los residuos examina la normalidad de los residuos, un supuesto para asegurar la validez de las pruebas estadísticas aplicadas a los coeficientes del modelo. Este gráfico compara los cuartiles de los residuos con los cuartiles de una distribución normal esperada. Si los puntos se alinean aproximadamente en una línea recta, esto indica una distribución normal de los residuos.

Además, el gráfico de Escala-Localización se emplea para confirmar, una vez más, la homogeneidad en la varianza de los residuos. Graficar la raíz cuadrada de los residuos absolutos contra los valores ajustados ayuda a verificar si la varianza de los residuos se mantiene constante a lo largo de los valores ajustados.

Por último, el gráfico de Distancias de Cook es para detectar observaciones influyentes que podrían influir de manera desproporcionada en el ajuste del modelo. Observaciones con distancias de Cook elevadas podrían requerir una revisión o eliminación del análisis para prevenir distorsiones significativas en los resultados del modelo.

5.3. Regresión Logística

La regresión logística es una técnica de modelado estadístico utilizada para predecir la probabilidad de un evento binario, es decir, un resultado que puede tener dos posibles valores (por ejemplo, sí/no, éxito/fracaso, alto/bajo). A diferencia del modelo anterior, que predice valores continuos, la regresión logística predice probabilidades que están acotadas entre 0 y 1.

La fórmula general para la regresión logística es:

$$log(\frac{P(Y=1)}{1-P(Y=1)}) = \beta_0 + \beta_1 \mathrm{X}_1 + \beta_2 \mathrm{X}_2 + \beta_3 \mathrm{X}_3 + \ldots + \beta_\rho X_\rho$$

Donde:

- P(Y=1) es la probabilidad de que el evento ocurra.
- β_0 es la intersección.
- $\beta_1\beta_2\beta_3$ son los coeficientes de las variables independientes $X_1X_2X_3$
- $log(\frac{P(Y=1)}{1-P(Y=1)})$ es el logaritmo de probabilidades.

Dicho lo anterior, implementar la regresión logística para clasificar transacciones basándose en si superan un umbral de rentabilidad, utilizando variables como Order_Priority, Sales_Channel y Unit_Price.

Para crear una variable binaria que representa si los beneficios superan un umbral establecido y para ajustar un modelo que predice esta variable, se utilizó el siguiente código R:

```
logit_model <- glm(High_Revenue ~ Units_Sold + Unit_Price + Order_Priority + Sales_Channel, family = binomial(), data = train_data)
# Ver el resumen del modelo
summary(logit_model)
```

La precisión del modelo se evaluó, proporcionando resultados importantes sobre su capacidad para identificar transacciones de alto rendimiento.

6. 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

6.1. 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 3:

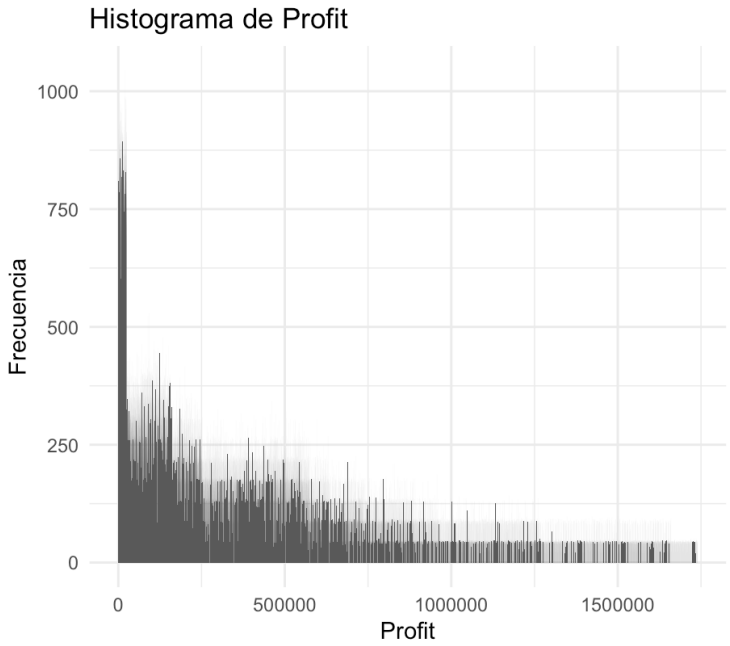

Imagen 3: histograma de la variable total_profit

En la grafico anterior, la mayoría de los registros en el conjunto de datos tienen un sesgo de “total profit” muy bajo, lo que podría indicar que la mayoría de las transacciones generar ganancias relativamente pequeñas, por lo tanto la distribución está sesgada hacia la derecha, lo cual es común en datos financieros, donde unas pocas transacciones pueden generar ganancias excepcionalmente altas.

6.2. 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 (Total_Cost), esto se visualiza en la siguiente Imagen 4:

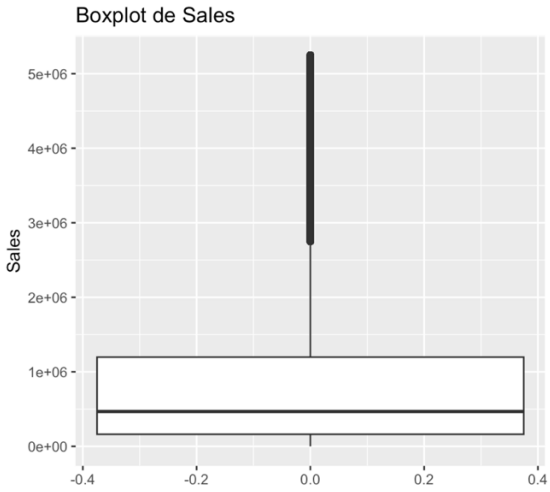

Imagen 4: diagrama de cajas de la variable Total_Cost.

En la imagen anterior se visualiza en el eje Y los valores de la variable “Total Cost” y los valores van desde 0 hasta más de 5000000. En la caja central muestra el rango intercuartil (IQR), que es el rango donde se encuentra el 50% de los datos, el borde inferior representa el primer cuartil (Q1 = 161,925), la línea dentro de la caja representa la mediana (Q2 = 467,712) y el borde superior de la caja representa el tercer cuartil (Q3 = 1,197,434)) y finalmente los puntos afuera de los bigotes se consideran valores atípicos.

6.3. Gráfico de Cajas de Total_Revenue

Este gráfico visualiza la distribución de los ingresos totales y destaca los valores atípicos o ‘outliers’ que se alejan de la tendencia general del conjunto de datos.

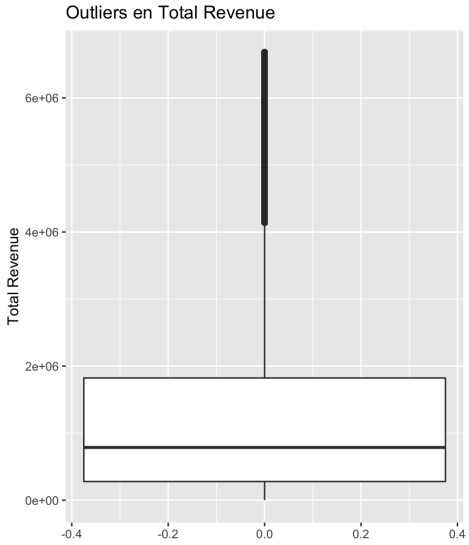

Imagen 5: outliers de Total Revenue

Los ‘bigotes’ que se extienden desde la caja indican la variabilidad fuera del rango intercuartílico, y se extienden hasta 1.5 veces el IQR desde la caja, llegando hasta el valor máximo y mínimo dentro de este rango.

6.4. Gráfico de Líneas de Ventas por Mes

En el siguiente grafico de líneas muestra la cantidad total de unidades vendidas por mes a lo largo del tiempo, en el eje X (mes): representa el tiempo en meses desde el año 2010 hasta aproximadamente el año 2020 y en el eje Y, representa el número total de unidades vendidas por mes.

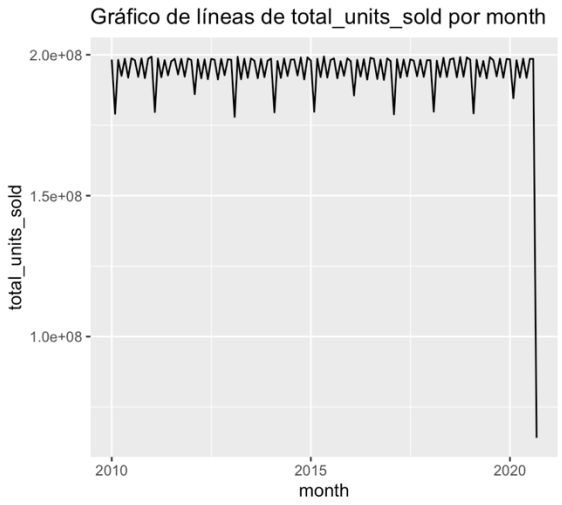

Imagen 5: gráfico de líneas ‘total_unitssold’ por mes

En la Imagen 5, la línea en el gráfico muestra una tendencia de unidades vendidas por mes, se observa que la línea tiene picos y valles regulares, esto quiere decir que las ventas mensuales son consistentemente altas (estable) durante todo el periodo, además puede haber meses con promociones o eventos especiales que aumentan las ventas, seguidos de meses con ventas más bajas.

Finalmente, hacia el final del periodo (aproximadamente 2020), se visualiza una caída importante en el número de unidades vendidas. Esto puede deberse a varios factores como: pandemia, interrupciones en la cadena de suministro, cambios en la demanda o incluso errores en los datos.

6.5. Análisis de correlación

La matriz de correlación analizada muestra las relaciones entre diversas variables numéricas dentro del conjunto de datos analizado. Esta visualización es fundamental para identificar la intensidad y la dirección de las asociaciones entre pared de las variables. A continuación, se detallan las observaciones clave extraídas de esta matriz:

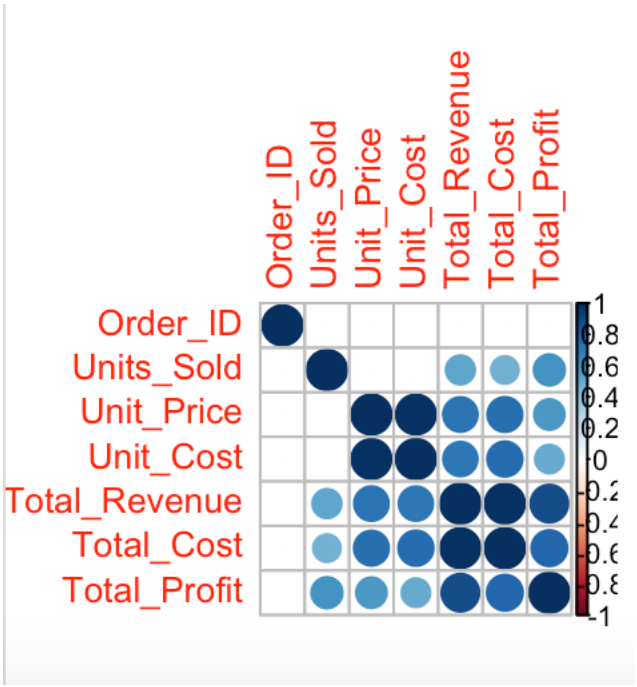

Imagen 6: análisis de correlación

En la Imagen 6, se observa lo siguiente:

- Correlación entre Unidades Vendidas y Total de Ingresos: Se observa una correlación positiva sustancial entre las unidades vendidas (Units_Sold) y el total de ingresos (Total_Revenue). Este vínculo fuerte sugiere que a medida que aumenta el número de unidades vendidas, los ingresos totales tienden a aumentar, lo cual es coherente con las expectativas de rendimiento de ventas.
- Relación entre Precio Unitario y Total de Ingresos: La correlación entre el precio unitario (Unit_Price) y el total de ingresos también es positivamente fuerte, indicando que los precios más altos por unidad están asociados con mayores ingresos totales. Esto podría reflejar una estrategia de precios premium que efectivamente contribuye al aumento de los ingresos.
- Costo Unitario y Total de Costos: Existe una correlación notable entre el costo unitario (Unit_Cost) y el total de costos (Total_Cost), lo que indica que los

incrementos en el costo por unidad se reflejan proporcionalmente en el aumento de los costos totales. Este patrón resalta la importancia del control de costos unitarios dentro de la gestión financiera.

- Total de Costos y Total de Ingresos: La fuerte correlación entre el total de costos y el total de ingresos sugiere que los negocios con mayores ingresos también incurren en costos más elevados, lo que puede ser indicativo de una estructura de costos escalable o de inversiones significativas en la capacidad de producción o expansión.
- Rentabilidad (Total de Ingresos vs. Total de Ganancias): El vínculo entre el total de ingresos (Total_Revenue) y el total de ganancias (Total_Profit) es altamente positivo, destacando que los incrementos en los ingresos están directamente relacionados con un aumento en las ganancias. Este es un indicativo claro de una gestión eficiente que maximiza la rentabilidad a partir del crecimiento de las ventas.

A continuación, para generar el mapa de calor de análisis de correlación de ingresos y unidades vendidas por región, aplicando un gradiente de color que va desde el azul (correlación negativa) al rojo (correlación positiva), con blanco indicando una correlación neutra (cero), tal como se ve en la siguiente gráfico:

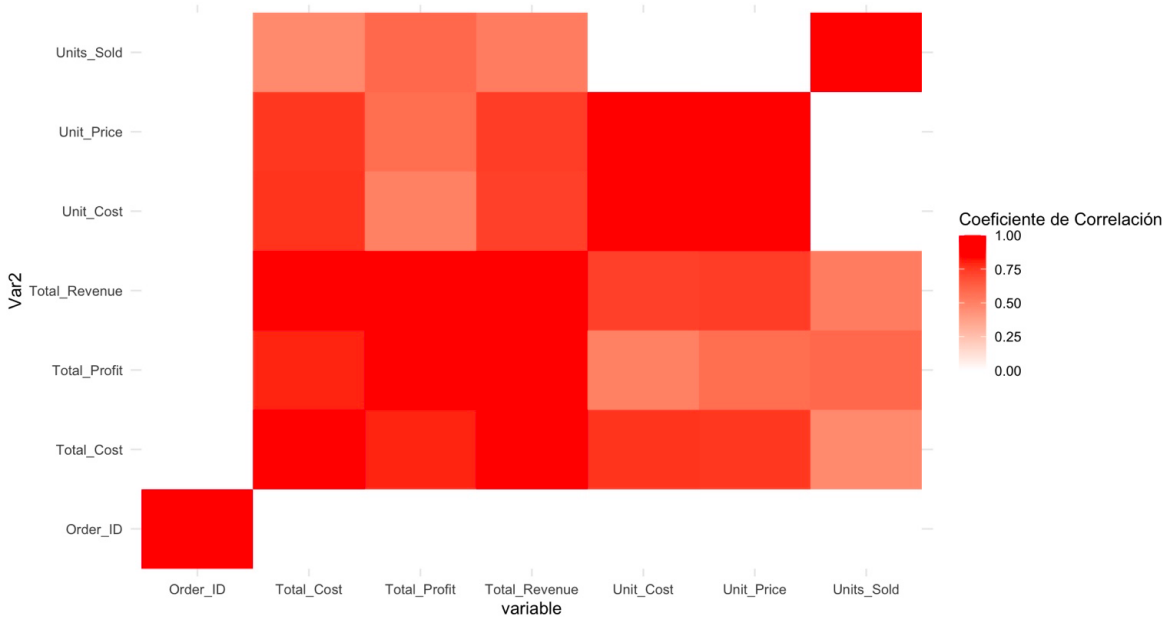

Imagen X: correlación por regiones

Algunas observaciones son:

- Alta Correlación Positiva: Variables como Units_Sold y Total_Revenue muestran una fuerte correlación positiva, lo que indica que un aumento en una está asociado con un aumento en la otra.
- Baja o Ninguna Correlación: La relación entre Order_ID y otras variables numéricas es esencialmente no correlacionada, lo cual es esperado ya que los ID de orden generalmente son identificadores únicos sin relación numérica con otras métricas.

6.6. Regresión lineal

Este modelo ha sido utilizado para explorar la relación entre los ingresos totales y variables predictoras como las unidades vendidas y el precio unitario. Los resultados indican cómo cada unidad adicional vendida o cada incremento en el precio unitario afecta los ingresos totales, proporcionando una base cuantitativa para decisiones de gestión de inventario y estrategias de precios, tal como se ve en la Imagen 7:

```
Residuals:
     Min       1Q   Median       3Q      Max
-2010489  -361307        4   361385  2010327

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.331e+06  6.572e+02   -2025   <2e-16 ***
Units_Sold   2.661e+02  9.705e-02    2742   <2e-16 ***
Unit_Price   5.000e+03  1.291e+00    3874   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 626400 on 4999997 degrees of freedom
Multiple R-squared:  0.8184,    Adjusted R-squared:  0.8184
F-statistic: 1.126e+07 on 2 and 4999997 DF,  p-value: < 2.2e-16
```

Imagen 7: resultados modelo lineal

- Intercepto: El valor del intercepto es aproximadamente -1,331,000. Este valor, aunque teóricamente representa los ingresos esperados cuando no se venden unidades y el precio unitario es cero, es principalmente de interés conceptual ya que tal escenario es impracticable en un contexto empresarial real.
- Units_Sold: El coeficiente asociado con Units_Sold es de aproximadamente 266.1, indicando que por cada unidad adicional vendida, los ingresos totales aumentan en promedio por 266.1 unidades monetarias, manteniendo constante el precio por unidad.
- Unit_Price: Similarmente, el coeficiente para Unit_Price es de 5000, lo que indica un aumento de una unidad monetaria en el precio unitario se asocia con un incremento

de 5000 unidades monetarias en los ingresos totales, asumiendo que el número de unidades vendidas se mantiene constante. El resultado muestra que aproximadamente el 81.84% de la variabilidad en los ingresos totales (R-cuadrado = 0.8184), lo que indica un buen ajuste del modelo a los datos. Este alto R-cuadrado ajustado asegura que tanto el volumen de ventas como el precio unitario son predictores relevantes y efectivos de los ingresos totales. Por otro lado, se realiza el siguiente gráfico para visualizar el resultado del modelo, tal como se ve en la siguiente Imagen 8:

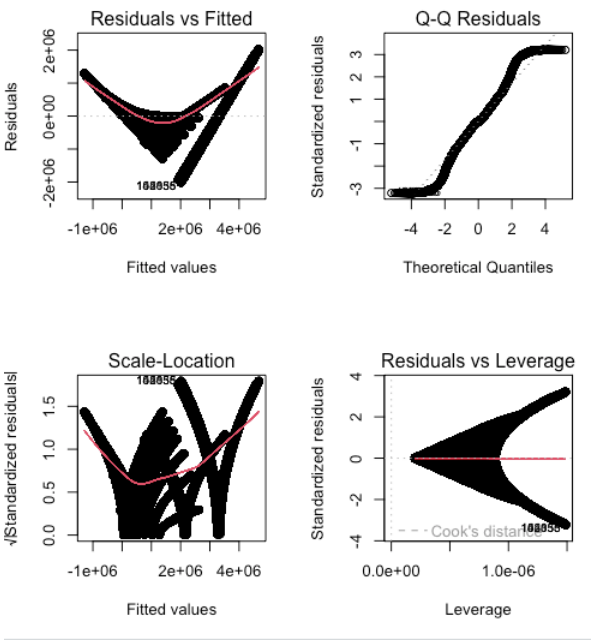

Imagen 8: gráfico modelo lineal

A partir de la Imagen 7, se deduce lo siguiente:

- Residuos vs Valores Ajustados: Se observa un patrón en forma de “V” o embudo en el gráfico. Este patrón indica que la varianza de los errores no es constante a lo largo de los valores predichos.
- Gráfico Q-Q de Residuos: El gráfico muestra desviaciones de los residuos estandarizados respecto a una distribución normal teórica. La normalidad de los residuos es relevante para la validez de muchas pruebas estadística asociada a la regresión lineal.

- Gráfico de Ubicación-Escala: Este gráfico refuerza la observación anterior del primer gráfico, mostrando variabilidad en la dispersión de los residuos.

6.7. Regresión Logística

Este modelo ha sido utilizado para predecir la probabilidad de que una transacción genere altos ingresos, basada en características similares. Este modelo ayuda a identificar las características significativas que determinan el éxito en términos de altos ingresos.

```
Call:
glm(formula = High_Revenue ~ Units_Sold + Unit_Price + Order_Priority +
    Sales_Channel, family = binomial(), data = train_data)

Coefficients:
                    Estimate Std. Error  z value Pr(>|z|)
(Intercept)        -8.947e+00  1.046e-02 -855.455  < 2e-16 ***
Units_Sold          9.868e-04  1.130e-06  873.223  < 2e-16 ***
Unit_Price          1.606e-02  1.829e-05  878.539  < 2e-16 ***
Order_PriorityH     3.323e-04  5.015e-03    0.066  0.94716
Order_PriorityL     2.582e-04  5.016e-03    0.051  0.95895
Order_PriorityM    -1.970e-03  5.016e-03   -0.393  0.69447
Sales_ChannelOnline 9.872e-03  3.546e-03    2.784  0.00537 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5545177  on 3999999  degrees of freedom
Residual deviance: 2043243  on 3999993  degrees of freedom
AIC: 2043257

Number of Fisher Scoring iterations: 7
```

Imagen 9: resultados modelo logístico

- Intercepto (Intercept):
  - ✓ Estimación: -8.547e+00
  - ✓ El intercepto es el logaritmo de la razón de probabilidades (log-odds) de que el evento ocurra (en este caso, que los ingresos sean altos) cuando todas las variables independientes son cero.
- Units_Sold:
  - ✓ Estimación: 3.866e-04
  - ✓ p-valor: < 2e-16 (muy significativo)
  - ✓ Por cada unidad adicional vendida, la log-odds de alcanzar altos ingresos aumenta en 0.0003866. Las mayores ventas están moderadamente asociadas con mayores ingresos, aunque el impacto por unidad vendida es pequeño.
- Unit_Price:
  - ✓ Estimación: 9.888e-04
  - ✓ p-valor: < 2e-16 (muy significativo)

- ✓ Por cada unidad adicional en el precio, la log-odds de alcanzar altos ingresos aumenta en 0.0009888. Similar a Units_Sold, aunque el efecto por unidad de cambio en el precio es pequeño, es estadísticamente significativo.

- Order_PriorityM:
  - ✓ Estimación: -2.428e-01
  - ✓ p-valor: 0.00357 (significativo)
  - ✓ Las órdenes con una prioridad media (M) tienen menos probabilidades de alcanzar altos ingresos en comparación con la categoría de referencia (probablemente Low).
- Order_PriorityH (Order_PriorityH):
  - ✓ Estimación: -8.272e-02
  - ✓ p-valor: 0.5037 (no significativo)
  - ✓ Las órdenes con alta prioridad (H) no muestran una diferencia significativa en la probabilidad de altos ingresos en comparación con la categoría de referencia, dado el p-valor alto.
- Sales_ChannelOnline:
  - ✓ Estimación: -5.066e-01
  - ✓ p-valor: 0.00393 (significativo)
  - ✓ Las órdenes realizadas a través del canal online tienen menos probabilidades de generar altos ingresos comparado con el canal de referencia (probablemente el canal físico).

Los residuos de desviación muestra una idea de cómo el modelo se ajusta a los datos en cada observación. Los valores mínimos y máximos indican la presencia de *outliers* que podrían necesitar más investigación.

6.7.1. Evaluación del Modelo

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 441577  59738
         1  58725 439960

               Accuracy : 0.8815
                 95% CI : (0.8809, 0.8822)
    No Information Rate : 0.5003
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7631

 Mcnemar's Test P-Value : 0.003279

            Sensitivity : 0.8826
            Specificity : 0.8805
         Pos Pred Value : 0.8808
         Neg Pred Value : 0.8822
             Prevalence : 0.5003
         Detection Rate : 0.4416
   Detection Prevalence : 0.5013
      Balanced Accuracy : 0.8815

       'Positive' Class : 0
```

Imagen 9: matriz de confusión

- Referencia (Verdaderos):
  - ✓ 0: 441837 casos fueron correctamente predichos como clase 0 (no altos ingresos).
  - ✓ 1: 557325 casos fueron incorrectamente clasificados como clase 0 cuando en realidad eran clase 1 (altos ingresos).
- Predicción (Predichos):
  - ✓ 0: 58725 casos fueron incorrectamente clasificados como clase 1 cuando en realidad eran clase 0.
  - ✓ 1: 439998 casos fueron correctamente predichos como clase 1.

7. Anexos

7.1. Código R (análisis con archivo CSV)

7.1.1. Funcion Resumen Numerico

```
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)
}
```

7.1.2. Función dinámica para gráficar

```
# Función para crear gráficos dinámicos
crear_grafico_dinamico <- function(data, tipo_grafico, x_var = NULL, y_var = NULL,
fill_var = NULL, date_var = NULL, units_var = NULL, title = "", x_label = "", y_label = "",
bins = 30) {
  # Validar que las variables existen en el data frame
  if (!is.null(x_var) && !x_var %in% colnames(data)) {
    stop(paste("La variable", x_var, "no existe en el data frame."))
  }
  if (!is.null(y_var) && !y_var %in% colnames(data)) {
    stop(paste("La variable", y_var, "no existe en el data frame."))
  }
  if (!is.null(fill_var) && !fill_var %in% colnames(data)) {
    stop(paste("La variable", fill_var, "no existe en el data frame."))
  }
  if (tipo_grafico == "line" && (!date_var %in% colnames(data) || !units_var %in%
colnames(data))) {
    stop("Las variables de fecha y unidades especificadas no existen en el data frame.")
  }

  if (tipo_grafico == "histograma") {
    p <- ggplot(data, aes_string(x = x_var)) +
      geom_histogram(binwidth = bins) +
```

```
    labs(title = title, x = x_label, y = y_label)
  } else if (tipo_grafico == "boxplot") {
    p <- ggplot(data, aes_string(x = x_var, y = y_var, fill = fill_var)) +
      geom_boxplot() +
      labs(title = title, x = x_label, y = y_label)
  } else if (tipo_grafico == "scatter") {
    p <- ggplot(data, aes_string(x = x_var, y = y_var, color = fill_var)) +
      geom_point() +
      labs(title = title, x = x_label, y = y_label)
  } else if (tipo_grafico == "line") {
    data <- data %>%
      mutate(Month = floor_date(as.Date(get(date_var), format = "%m/%d/%Y"), "month"))
%>%
      group_by(Month) %>%
      summarise(Total_Units_Sold = sum(get(units_var), na.rm = TRUE)) %>%
      ungroup()

    p <- ggplot(data, aes(x = Month, y = Total_Units_Sold)) +
      geom_line() +
      labs(title = title, x = x_label, y = y_label) +
      theme_minimal()
  } else {
    stop("Tipo de gráfico no soportado. Usa 'histograma', 'boxplot', 'scatter' o 'line'.")
  }

  p <- p + theme_minimal()
  print(p)
}
```

7.1.3. Función base de datos reducida

```
# Función para crear una base de datos reducida
base_datos_reducida <- function(data, date_var = "Order_Date", group_vars = NULL) {
  # Registrar el tiempo de inicio
```

```
  start_time <- Sys.time()

  # Filtrar filas con fechas válidas
  data <- data %>% filter(!is.na(!!sym(date_var)))

  # Filtrar solo las columnas que existen en el data frame
  valid_group_vars <- group_vars[group_vars %in% colnames(data)]
  if (length(valid_group_vars) == 0) {
    valid_group_vars <- NULL
  }

  # Agrupar por Date y variables cualitativas válidas, y sumar las variables cuantitativas
  data_reducida <- data %>%
    group_by(across(all_of(c(date_var, valid_group_vars)))) %>%
    summarise(across(where(is.numeric), sum, na.rm = TRUE), .groups = 'drop') %>%
    arrange(across(all_of(date_var)))

  # Registrar el tiempo de finalización
  end_time <- Sys.time()

  # Calcular la duración
  duration <- end_time - start_time

  # Mostrar el avance y cuánto tiempo tomó
  cat("Proceso completado. Tiempo tomado:", duration, "\n")

  return(data_reducida)
}
```

7.2. Código R (Análisis con Mysql)

7.2.1. Procesamiento y Carga de datos a Base de Datos

```
# 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)
```

7.2.2. Función para la Limpieza de datos

```
limpiar_datos_nulos <- function(data) {
  # Reemplazar valores nulos en variables numéricas por la media de la columna
  data <- data %>%
    mutate(across(where(is.numeric), ~ ifelse(is.na(.), mean(., na.rm = TRUE), .)))

  # Reemplazar valores nulos en variables cualitativas por "Desconocido"
  data <- data %>%
    mutate(across(where(is.character), ~ ifelse(is.na(.), "Desconocido", .)))

  return(data)
}
```