EDA: Telecom Churn

Sebastián Castaño Suárez, Graciela Díaz Taveras, Sergio García Martín, Aurora Pérez García

2025-12-07

Introducción

En este trabajo realizaremos un Análisis Exploratorio de Datos (EDA) completo sobre un conjunto de datos de una compañía de telecomunicaciones. El objetivo es simular un proyecto de **Customer Analytics** para entender el comportamiento de los clientes, tal como se plantea en la descripción del problema de negocio.

El dataset proviene del archivo `telecom_churn_semi_structured.json` .

Objetivo

El objetivo principal es identificar patrones que nos ayuden a predecir y reducir el abandono de clientes (**Churn**).

- **¿Qué características tienen los clientes que abandonan la compañía?**
- **¿Existen factores económicos o de servicio determinantes?**

Paso 1: Cargar los datos y explorar su estructura

El objetivo es comprender la estructura del dataset y las variables clave.

1. Carga y Librerías

```
# 1. Instalar paquetes necesarios (comentado para evitar reinstalación al generar el reporte)
# install.packages("jsonlite")
# install.packages("tidyverse")
```

```
# 2. Cargar paquetes necesarios
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.5.2
```

```
## Warning: package 'ggplot2' was built under R version 4.5.1
```

```
## ── 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   4.0.0     ✓ tibble    3.3.0
## ✓ lubridate 1.9.4     ✓ tidyr     1.3.1
## ✓ purrr     1.0.4
## ── 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 be
come errors
```

```
library(jsonlite)
```

```
## Warning: package 'jsonlite' was built under R version 4.5.2
```

```
##
## Adjuntando el paquete: 'jsonlite'
##
## The following object is masked from 'package:purrr':
##
##     flatten
```

```
# 3. Cargar los datos
# Usamos flatten = TRUE para 'aplanar' estructuras anidadas automáticamente
churn_data <- fromJSON("telecom_churn_semi_structured.json", flatten = TRUE)

# 4. Convertir a tibble (formato de tabla mejorado de R)
churn_data <- as_tibble(churn_data)

# 5. Ver las primeras filas
head(churn_data)
```

```
## # A tibble: 6 × 24
##   customer_id personal_info.age personal_info.gender personal_info.location
##         <int>             <int> <chr>                <chr>
## 1           1                63 Male                 Urban
## 2           2                68 Female               Suburban
## 3           3                37 Female               Suburban
## 4           4                56 Female               Rural
## 5           5                36 Male                 Urban
## 6           6                61 Female               Rural
## # ℹ 20 more variables: subscription.plan_type <chr>,
## #   subscription.tenure_months <int>, subscription.monthly_charges <dbl>,
## #   subscription.billing_cycle <chr>, usage.total_call_minutes <int>,
## #   usage.data_usage_gb <dbl>, usage.num_sms_sent <int>,
## #   usage.streaming_services <chr>, usage.roaming_usage <dbl>,
## #   customer_support.customer_service_calls <int>,
## #   customer_support.customer_feedback_score <int>, …
```

2. Tipos de Variables

Comprobamos qué tipo de datos tenemos y ajustamos los formatos necesarios.

```
# 1. Explorar la estructura de los tipos de datos
glimpse(churn_data)
```

```
## Rows: 55,000
## Columns: 24
## $ customer_id                              <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10…
## $ personal_info.age                        <int> 63, 68, 37, 56, 36, 61, 35, 2…
## $ personal_info.gender                     <chr> "Male", "Female", "Female", "…
## $ personal_info.location                   <chr> "Urban", "Suburban", "Suburba…
## $ subscription.plan_type                   <chr> "Premium", "Standard", "Premi…
## $ subscription.tenure_months               <int> 44, 61, 43, 25, 71, 58, 33, 4…
## $ subscription.monthly_charges             <dbl> 118.22, 78.54, 65.59, 61.91, …
## $ subscription.billing_cycle               <chr> "Quarterly", "Quarterly", "An…
## $ usage.total_call_minutes                 <int> 1267, 714, 254, 6267, 3222, 2…
## $ usage.data_usage_gb                      <dbl> 61.91, 70.87, 51.84, 35.29, 9…
## $ usage.num_sms_sent                       <int> 852, 84, 575, 29, 953, 351, 5…
## $ usage.streaming_services                 <chr> "No", "Yes", "Yes", "No", "No…
## $ usage.roaming_usage                      <dbl> 2.94, 4.96, 3.69, 8.61, 5.99,…
## $ customer_support.customer_service_calls  <int> 10, 9, 12, 5, 13, 9, 2, 13, 1…
## $ customer_support.customer_feedback_score <int> 7, 4, 9, 1, 3, 3, 4, 3, 2, 9,…
## $ customer_support.network_quality_score   <dbl> 3.6, 2.7, 4.6, 1.3, 3.5, 4.0,…
## $ payments.payment_method                  <chr> "UPI", "Debit Card", "Credit …
## $ payments.payment_failures                <int> 2, 2, 1, 0, 2, 4, 1, 3, 3, 0,…
## $ payments.discount_applied                <int> 0, 0, 1, 1, 0, 0, 0, 0, 1, 1,…
## $ additional_features.device_type          <chr> "Android", "iPhone", "iPhone"…
## $ additional_features.multiple_lines       <chr> "No", "Yes", "No", "No", "No"…
## $ additional_features.family_plan          <chr> "Yes", "No", "No", "Yes", "Ye…
## $ additional_features.premium_support      <chr> "No", "Yes", "No", "No", "Yes…
## $ additional_features.churn                <int> 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,…
```

```
# 2. Transformar las variables categóricas (de tipo 'character') a factor
churn_data <- churn_data %>%
  mutate(across(where(is.character), as.factor))

# Comprobamos que ha cambiado a Factor
glimpse(churn_data)
```

```
## Rows: 55,000
## Columns: 24
## $ customer_id                              <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10…
## $ personal_info.age                        <int> 63, 68, 37, 56, 36, 61, 35, 2…
## $ personal_info.gender                     <fct> Male, Female, Female, Female,…
## $ personal_info.location                   <fct> Urban, Suburban, Suburban, Ru…
## $ subscription.plan_type                   <fct> Premium, Standard, Premium, B…
## $ subscription.tenure_months               <int> 44, 61, 43, 25, 71, 58, 33, 4…
## $ subscription.monthly_charges             <dbl> 118.22, 78.54, 65.59, 61.91, …
## $ subscription.billing_cycle               <fct> Quarterly, Quarterly, Annuall…
## $ usage.total_call_minutes                 <int> 1267, 714, 254, 6267, 3222, 2…
## $ usage.data_usage_gb                      <dbl> 61.91, 70.87, 51.84, 35.29, 9…
## $ usage.num_sms_sent                       <int> 852, 84, 575, 29, 953, 351, 5…
## $ usage.streaming_services                 <fct> No, Yes, Yes, No, No, Yes, No…
## $ usage.roaming_usage                      <dbl> 2.94, 4.96, 3.69, 8.61, 5.99,…
## $ customer_support.customer_service_calls  <int> 10, 9, 12, 5, 13, 9, 2, 13, 1…
## $ customer_support.customer_feedback_score <int> 7, 4, 9, 1, 3, 3, 4, 3, 2, 9,…
## $ customer_support.network_quality_score   <dbl> 3.6, 2.7, 4.6, 1.3, 3.5, 4.0,…
## $ payments.payment_method                  <fct> UPI, Debit Card, Credit Card,…
## $ payments.payment_failures                <int> 2, 2, 1, 0, 2, 4, 1, 3, 3, 0,…
## $ payments.discount_applied                <int> 0, 0, 1, 1, 0, 0, 0, 0, 1, 1,…
## $ additional_features.device_type          <fct> Android, iPhone, iPhone, Andr…
## $ additional_features.multiple_lines       <fct> No, Yes, No, No, No, No, Yes,…
## $ additional_features.family_plan          <fct> Yes, No, No, Yes, Yes, No, No…
## $ additional_features.premium_support      <fct> No, Yes, No, No, Yes, No, Yes…
## $ additional_features.churn                <int> 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,…
```

3. Resumen de las variables claves:

```
# 1. Generar un resumen estadístico de TODAS las variables
summary(churn_data)
```

```
##   customer_id    personal_info.age personal_info.gender personal_info.location
##  Min.   :    1   Min.   :18.00     Female:27757         Rural   :18379
##  1st Qu.:13751   1st Qu.:32.00     Male  :27243         Suburban:18253
##  Median :27501   Median :46.00                          Urban   :18368
##  Mean   :27501   Mean   :45.91
##  3rd Qu.:41250   3rd Qu.:60.00
##  Max.   :55000   Max.   :74.00
##  subscription.plan_type subscription.tenure_months subscription.monthly_charges
##  Basic   :18422         Min.   : 1.00              Min.   : 10.01
##  Premium :18128         1st Qu.:18.00              1st Qu.: 45.17
##  Standard:18450         Median :36.00              Median : 79.78
##                         Mean   :36.09              Mean   : 79.92
##                         3rd Qu.:54.00              3rd Qu.:114.89
##                         Max.   :71.00              Max.   :149.99
##  subscription.billing_cycle usage.total_call_minutes usage.data_usage_gb
##  Annually :18216            Min.   : 100             Min.   :  0.50
##  Monthly  :18330            1st Qu.:2587             1st Qu.: 25.36
##  Quarterly:18454            Median :5074             Median : 49.82
##                             Mean   :5062             Mean   : 50.13
##                             3rd Qu.:7536             3rd Qu.: 75.04
##                             Max.   :9999             Max.   :100.00
##  usage.num_sms_sent usage.streaming_services usage.roaming_usage
##  Min.   :  0.0      No :27501                Min.   : 0.000
##  1st Qu.:250.0      Yes:27499                1st Qu.: 2.500
##  Median :498.0                               Median : 5.030
##  Mean   :498.7                               Mean   : 5.014
##  3rd Qu.:748.0                               3rd Qu.: 7.520
##  Max.   :999.0                               Max.   :10.000
##  customer_support.customer_service_calls
##  Min.   : 0.000
##  1st Qu.: 5.000
##  Median :10.000
##  Mean   : 9.522
##  3rd Qu.:15.000
##  Max.   :19.000
##  customer_support.customer_feedback_score
##  Min.   :1.00
##  1st Qu.:3.00
##  Median :5.00
##  Mean   :4.99
##  3rd Qu.:7.00
##  Max.   :9.00
##  customer_support.network_quality_score payments.payment_method
##  Min.   :1                              Credit Card:13668
##  1st Qu.:2                              Debit Card :13699
##  Median :3                              UPI        :13727
##  Mean   :3                              Wallet     :13906
##  3rd Qu.:4
##  Max.   :5
##  payments.payment_failures payments.discount_applied
##  Min.   :0.000             Min.   :0.0000
##  1st Qu.:1.000             1st Qu.:0.0000
##  Median :2.000             Median :0.0000
##  Mean   :1.994             Mean   :0.4985
##  3rd Qu.:3.000             3rd Qu.:1.0000
```

```
##  Max.   :4.000             Max.   :1.0000
##  additional_features.device_type additional_features.multiple_lines
##  Android      :18557             No :27425
##  Feature Phone:18167             Yes:27575
##  iPhone       :18276
##
##
##
##  additional_features.family_plan additional_features.premium_support
##  No :27552                       No :27529
##  Yes:27448                       Yes:27471
##
##
##
##
##  additional_features.churn
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.3524
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

Interpretación inicial: Observamos que la media de la variable objetivo es **0.3524**, lo que indica que aproximadamente el **35%** de los clientes de la muestra han abandonado la compañía. No se observan valores nulos (`NA's`) en el resumen estadístico, por lo que no es necesaria la imputación de datos.

Paso 2: Limpieza y Calidad del Dato

Aunque no hay nulos, preparamos la variable objetivo y verificamos duplicados.

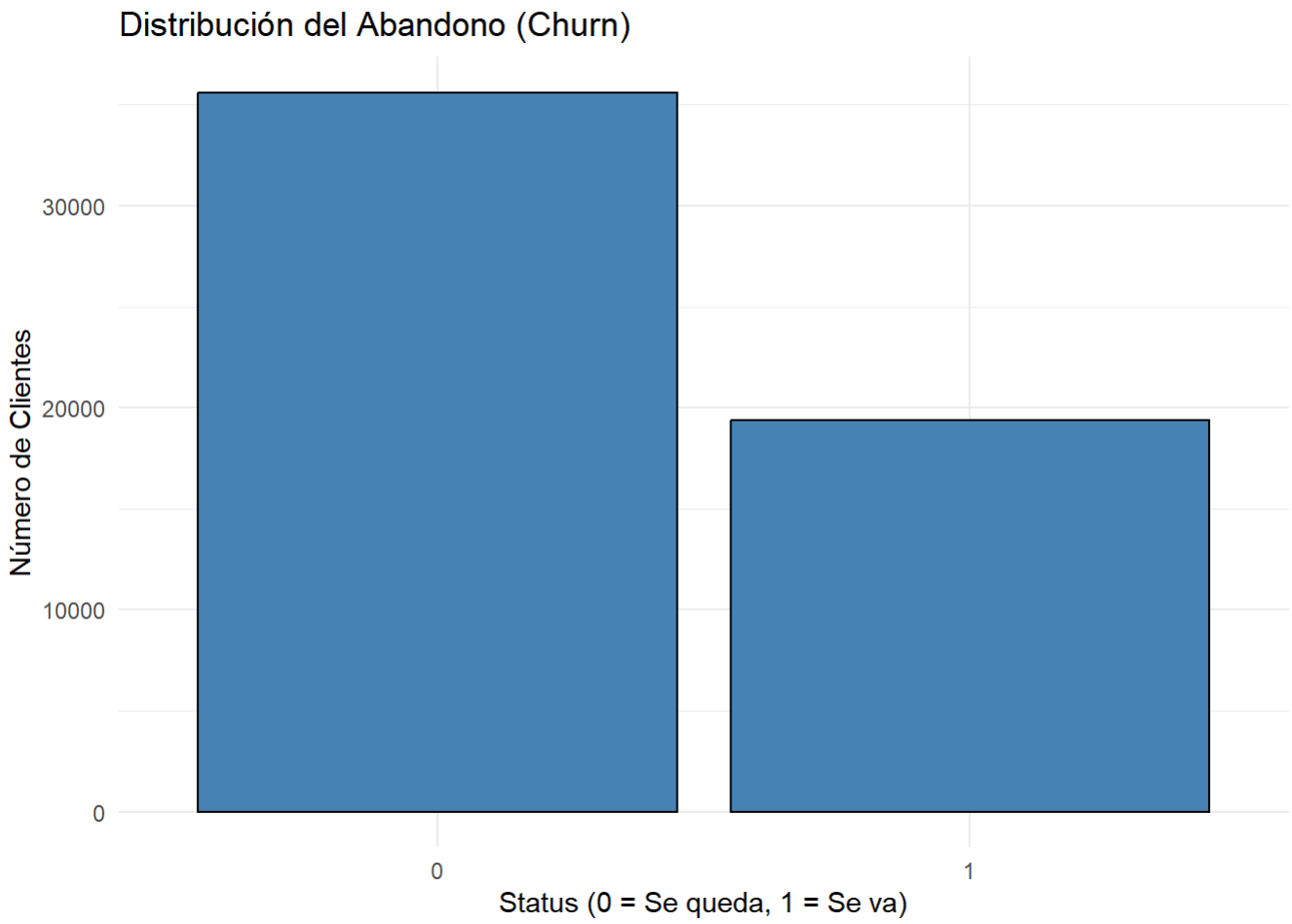
```
# 1. Convertir 'churn' de numérico (0/1) a factor (0/1) para el análisis gráfico
churn_data <- churn_data %>%
  mutate(additional_features.churn = as.factor(additional_features.churn))

# 2. Comprobación de duplicados
num_duplicados <- sum(duplicated(churn_data))
print(paste("Número de filas duplicadas:", num_duplicados))
```

```
## [1] "Número de filas duplicadas: 0"
```

Paso 3: Análisis Descriptivo (Univariado)

1. Variable Objetivo: Churn

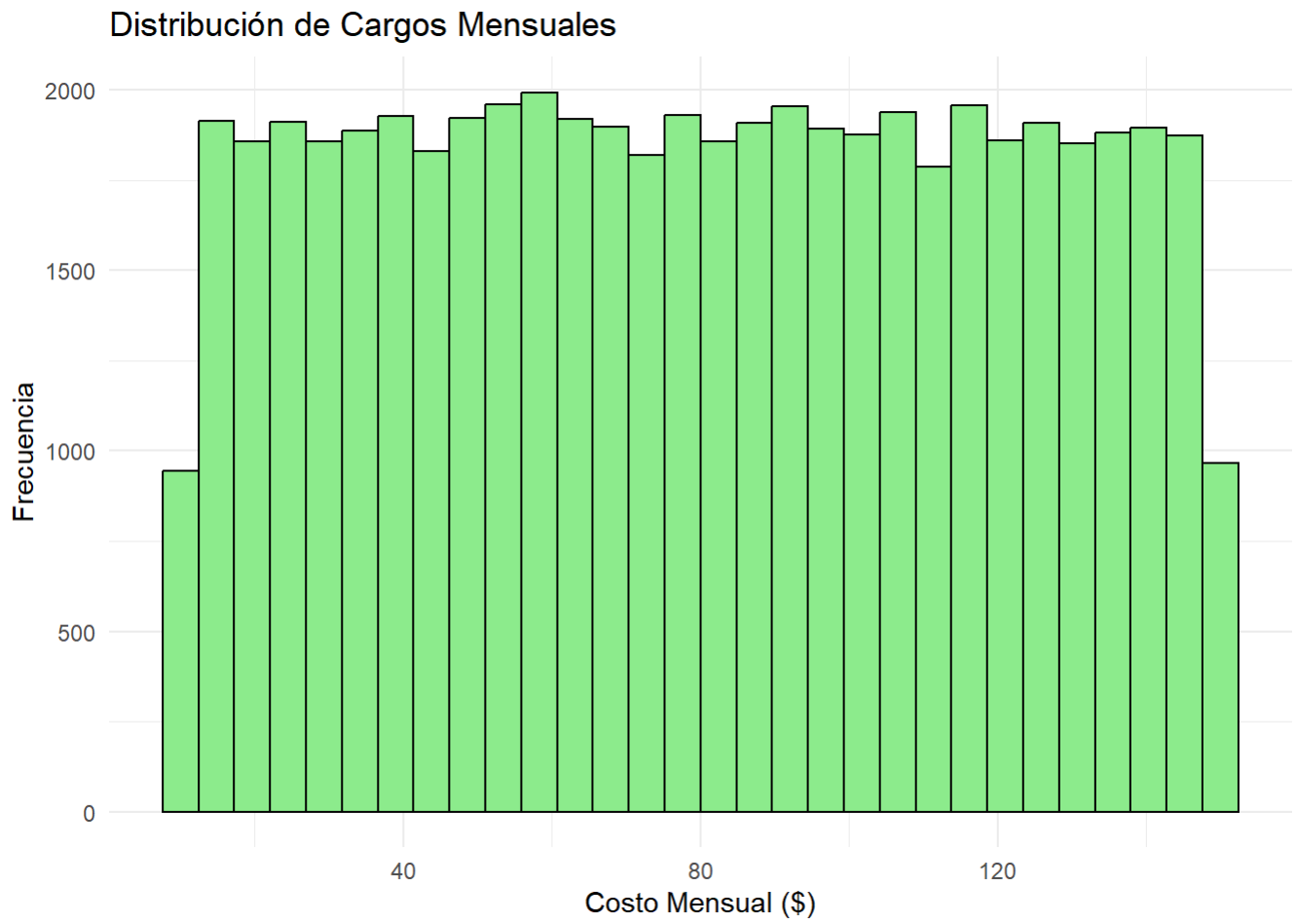
```
# Gráfico de barras para ver cuántos clientes se van vs los que se quedan
ggplot(churn_data, aes(x = additional_features.churn)) +
  geom_bar(fill = "steelblue", color = "black") +
  labs(title = "Distribución del Abandono (Churn)",
       x = "Status (0 = Se queda, 1 = Se va)",
       y = "Número de Clientes") +
  theme_minimal()
```



2. Variable Numérica: `subscription.monthly_charges`

Queremos ver cuánto pagan mensualmente los clientes.

```
# Histograma para ver la distribución de precios
ggplot(churn_data, aes(x = subscription.monthly_charges)) +
  geom_histogram(bins = 30, fill = "lightgreen", color = "black") +
  labs(title = "Distribución de Cargos Mensuales",
       x = "Costo Mensual ($)",
       y = "Frecuencia") +
  theme_minimal()
```



Interpretación:

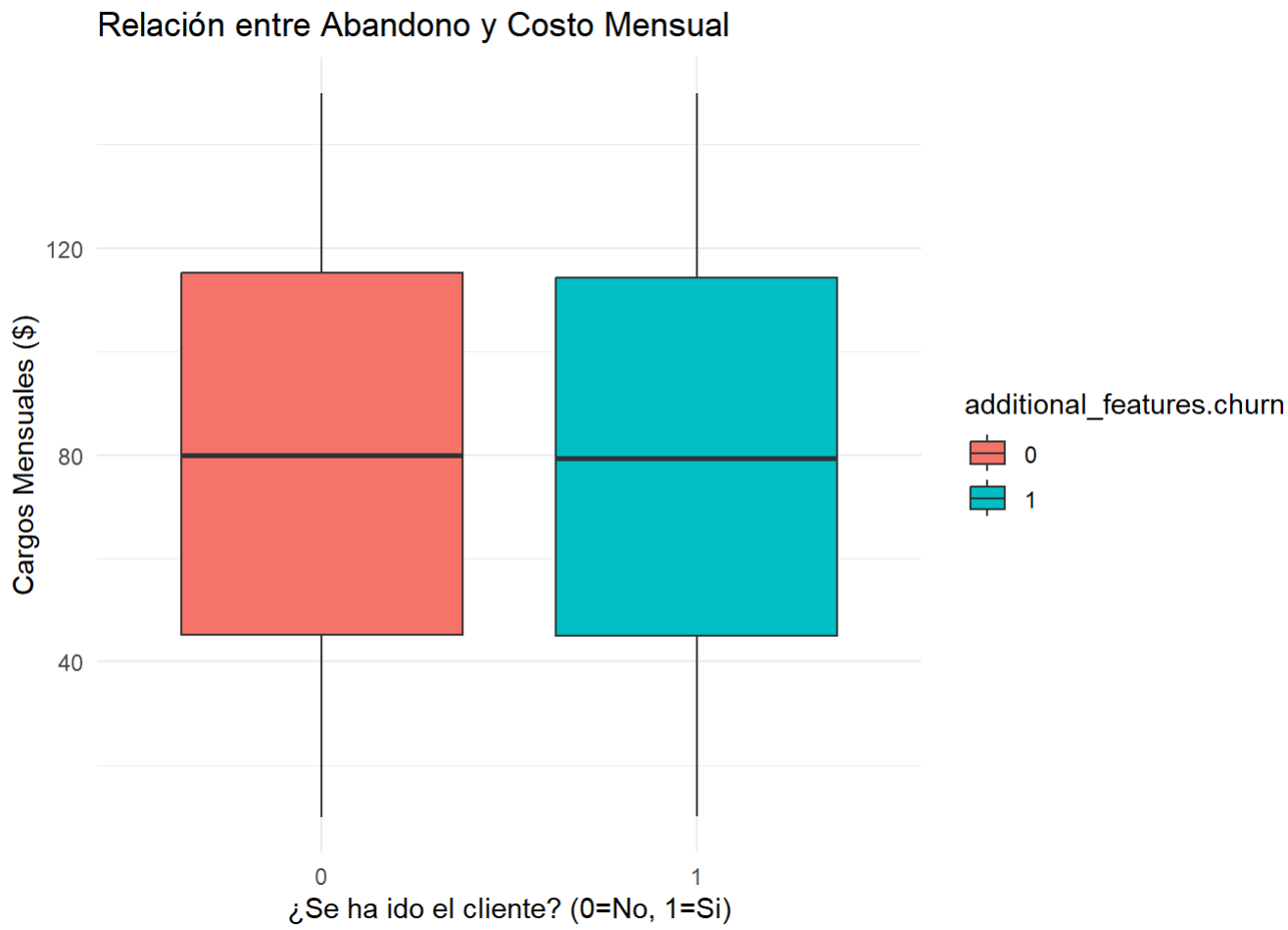
- **Churn (Gráfico Azul):** Vemos que la clase mayoritaria es la 0 (Clientes que se quedan). Como vimos en el resumen, el abandono es del 35%.
- **Cargos Mensuales (Gráfico Verde):** Observamos una distribución **Uniforme (plana)**. No hay un precio "típico" ni una concentración clara de tarifas.

Paso 4: Análisis Bivariado (Relación Económica)

El objetivo es examinar si el coste del servicio influye en la decisión de abandonar. Se analiza si pagan más, en promedio, los clientes que se van.

Gráfico de Cajas (churn vs. monthly_charges):

```
# Boxplot comparando lo que pagan los que se quedan (0) vs los que se van (1)
ggplot(churn_data, aes(x = additional_features.churn, y = subscription.monthly_charges, fill
= additional_features.churn)) +
  geom_boxplot() +
  labs(title = "Relación entre Abandono y Costo Mensual",
       x = "¿Se ha ido el cliente? (0=No, 1=Si)",
       y = "Cargos Mensuales ($)") +
  theme_minimal()
```




```
# Calculamos la media de precio para cada grupo
churn_data %>%
  group_by(additional_features.churn) %>%
  summarise(
    Media_Precio = mean(subscription.monthly_charges),
    Mediana_Precio = median(subscription.monthly_charges)
  )
```

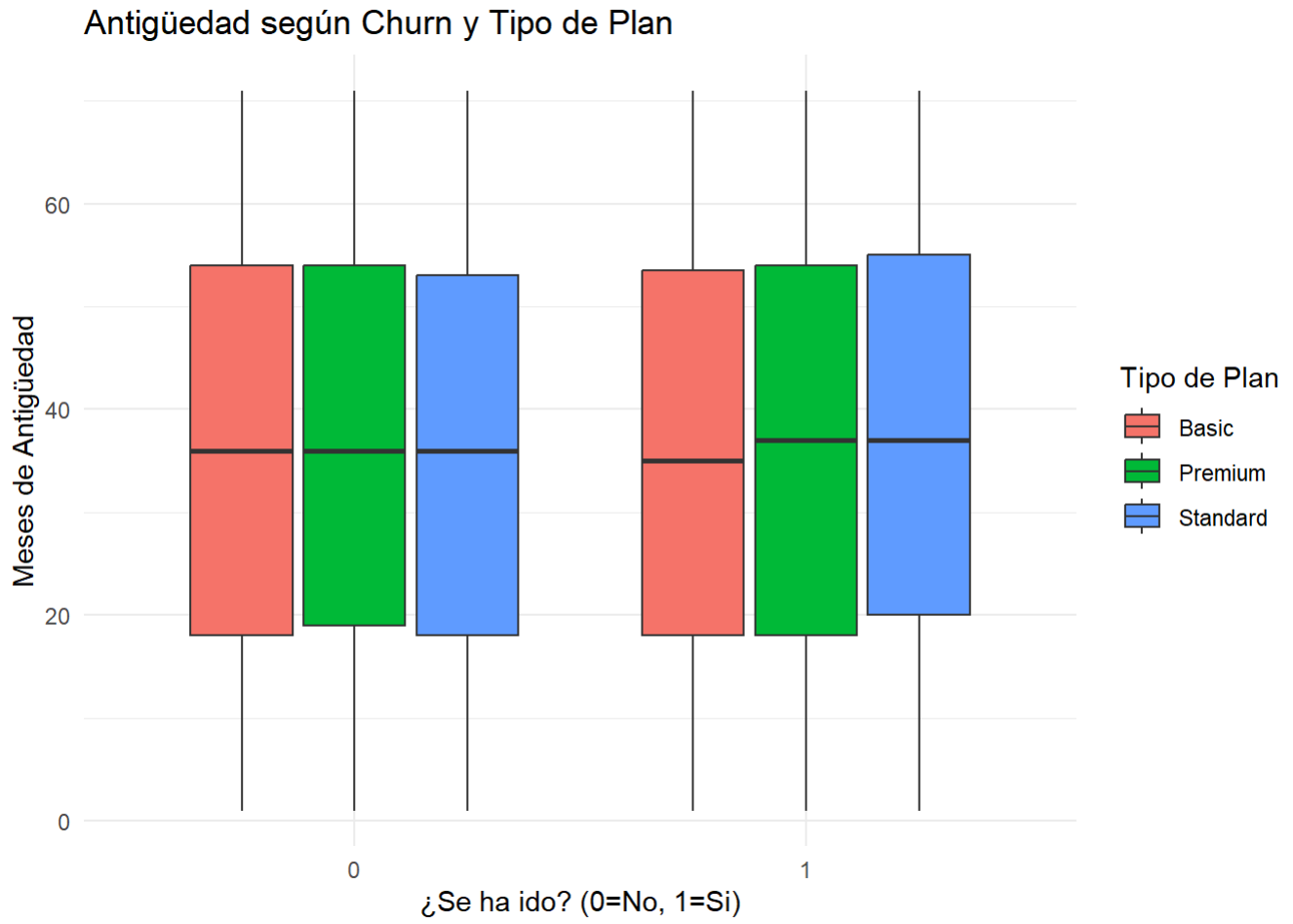
```
## # A tibble: 2 × 3
##   additional_features.churn Media_Precio Mediana_Precio
##   <fct>                           <dbl>          <dbl>
## 1 0                                80.0           80.0
## 2 1                                79.7           79.3
```

Interpretación: No existe diferencia significativa en el costo mensual entre los clientes que abandonan y los que permanecen. Las medias y medianas son prácticamente idénticas.

Paso 5: Análisis Multivariado

Buscamos si hay diferencia de Churn entre los clientes nuevos y los veteranos, desglosado por plan.

```
# 1. Gráfico de cajas: Antigüedad vs Churn, coloreado por Tipo de Plan
ggplot(churn_data, aes(x = additional_features.churn, y = subscription.tenure_months, fill =
subscription.plan_type)) +
  geom_boxplot() +
  labs(title = "Antigüedad según Churn y Tipo de Plan",
       x = "¿Se ha ido? (0=No, 1=Si)",
       y = "Meses de Antigüedad",
       fill = "Tipo de Plan") +
  theme_minimal()
```



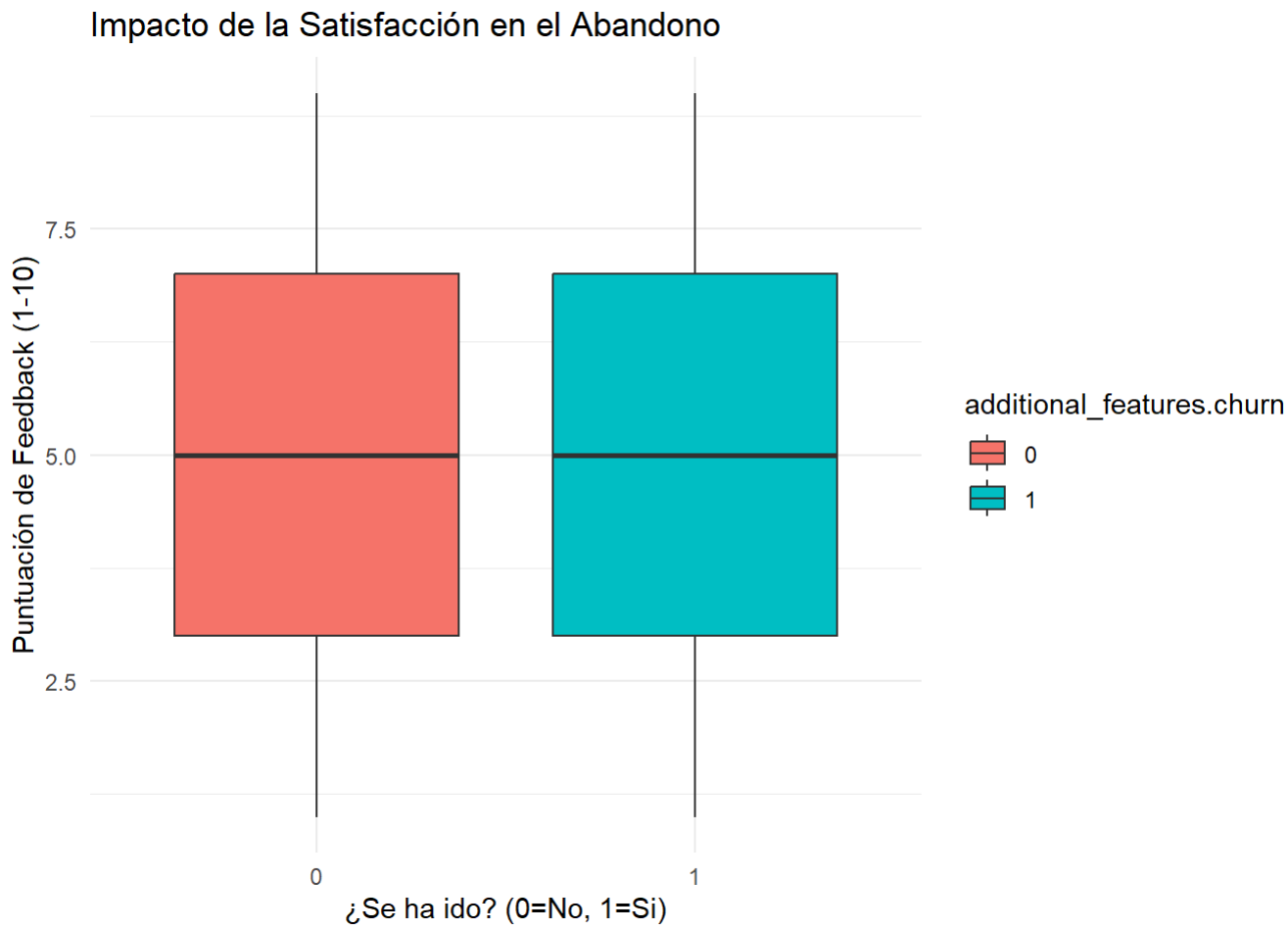
Interpretación: No existe diferencia significativa en la antigüedad ni el tipo de plan entre los clientes que abandonan y los que permanecen.

Paso 6: Análisis de Satisfacción y Ubicación

1. Satisfacción vs Churn

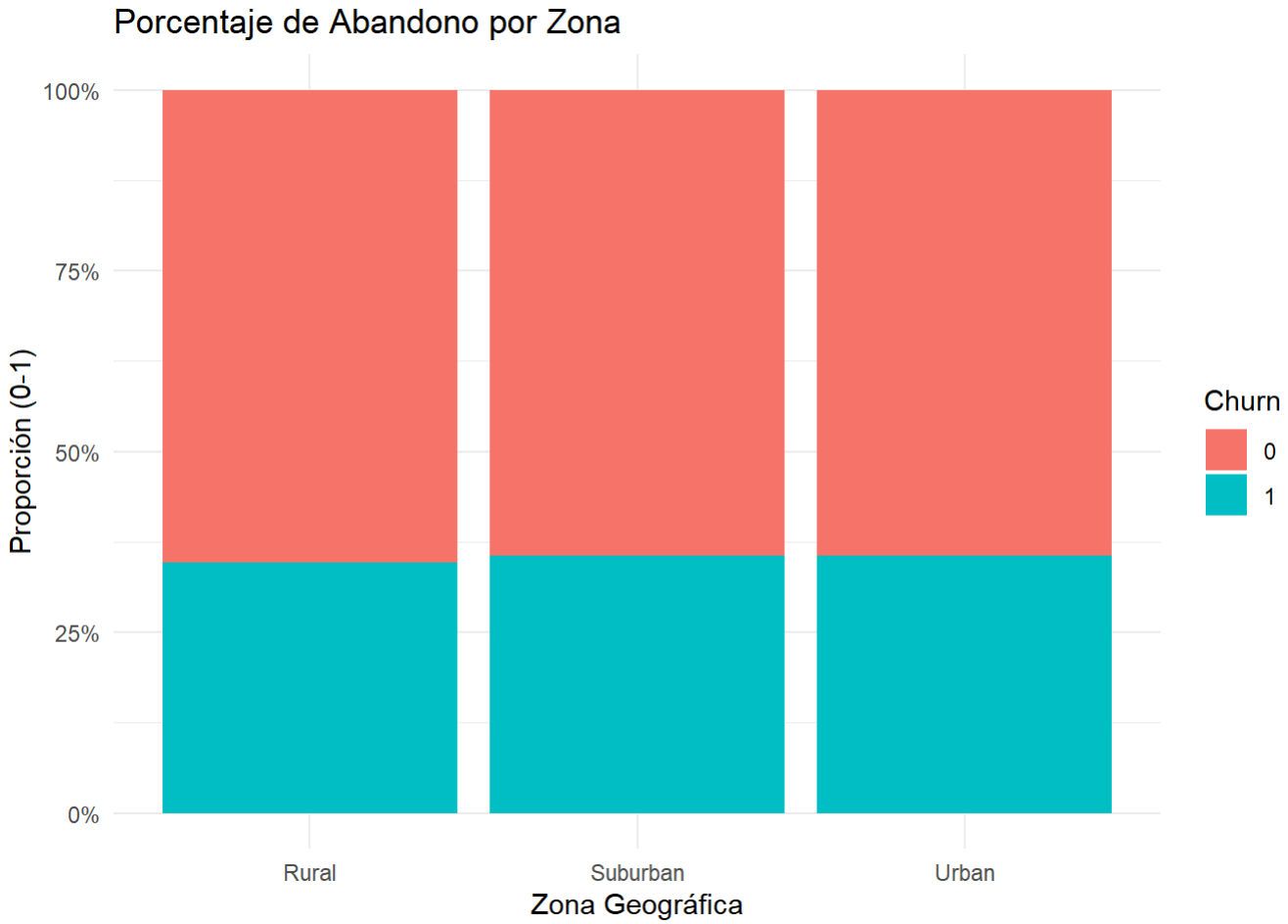
¿Los clientes que se van tienen puntuaciones más bajas?

```
ggplot(churn_data, aes(x = additional_features.churn, y = customer_support.customer_feedback_
score, fill = additional_features.churn)) +
  geom_boxplot() +
  labs(title = "Impacto de la Satisfacción en el Abandono",
       x = "¿Se ha ido? (0=No, 1=Si)",
       y = "Puntuación de Feedback (1-10)") +
  theme_minimal()
```



2. Localización vs Churn

```
ggplot(churn_data, aes(x = personal_info.location, fill = additional_features.churn)) +
  geom_bar(position = "fill") +
  labs(title = "Porcentaje de Abandono por Zona",
       x = "Zona Geográfica",
       y = "Proporción (0-1)",
       fill = "Churn") +
  theme_minimal() +
  scale_y_continuous(labels = scales::percent)
```



Interpretación: No existe diferencia significativa en la localización o la satisfacción entre los clientes que abandonan y los que permanecen.

Conclusiones y Decisión

El análisis exploratorio evidencia la ausencia total de correlaciones significativas, sugiriendo un origen sintético y aleatorio de los datos. Al no existir patrones predictivos válidos, se optará por **descartar este dataset** para la fase de modelado. Se procederá a generar datos modificados que incorporen relaciones de negocio lógicas y anomalías controladas para permitir un modelado efectivo en la siguiente fase del proyecto.