Compras de productos en Promoción

Taller de Análisis en “R”

NOMBRE: Rodríguez Karen
CARRERA: Ingeniería Informática
ASIGNATURA: Minería de Datos TI2061
PROFESOR: Marambio Fredy
FECHA: 22 octubre 2025

Índice

1 Introducción

En un mercado cada vez más competitivo, la capacidad de una empresa para comprender profundamente a sus clientes es un diferenciador clave para el éxito. El presente informe detalla un análisis exhaustivo de los datos de clientes con el objetivo de descubrir los patrones de comportamiento, las características demográficas y los perfiles financieros que influyen en la decisión de compra de una promoción específica.

Este estudio se divide en dos fases principales: un análisis descriptivo y un análisis predictivo. En la primera fase, se exploran los datos para construir un perfil detallado del comprador, evaluando la influencia de variables como el género, el nivel educacional y la salud financiera. El objetivo es responder a preguntas fundamentales sobre quiénes son nuestros clientes y qué características definen a quienes aceptan nuestras ofertas.

En la segunda fase, el análisis avanza hacia la construcción de modelos de machine learning para anticipar el comportamiento futuro. Se evalúa la capacidad de predecir la aceptación de una promoción, estimar montos de compra, segmentar a los clientes en grupos de consumo homogéneos e identificar las variables más influyentes en sus decisiones. El propósito final de este informe es transformar los datos en bruto en insights accionables que permitan optimizar las estrategias de marketing, personalizar las ofertas y mejorar la toma de decisiones estratégicas de la compañía.

2 Objetivos

2.1 Objetivo General

Elaborar un informe de análisis de datos utilizando el lenguaje de programación R para evaluar los resultados de una campaña promocional de una multitienda, identificando patrones de comportamiento del consumidor y generando un modelo predictivo que apoye la toma de decisiones empresariales futuras.

2.2 Objetivos Específicos

1. Preparar el conjunto de datos "Compras de productos en Promoción" para su análisis en RStudio, realizando la limpieza y transformación de variables necesarias.
2. Ejecutar un análisis estadístico descriptivo para caracterizar y perfilar a los clientes en función de sus datos demográficos, uso de la tarjeta de crédito y patrones de consumo.
3. Construir un modelo estadístico predictivo que permita identificar la probabilidad de que un cliente adquiera un producto ofertado en una promoción.
4. Crear visualizaciones gráficas, como histogramas y diagramas de dispersión, para interpretar y comunicar de manera efectiva los hallazgos del análisis.

3 Requerimientos

3.1 Requerimientos Descriptivos

1. **Perfil del Comprador:** ¿Cuál es el perfil demográfico principal (rango etario, nivel educacional, sexo) de los clientes que sí compraron el producto en promoción?
2. **Género y Compra:** Comparar el porcentaje de hombres versus mujeres que compraron el producto en promoción. ¿Las mujeres compraron más que los hombres?
3. **Educación y Compra:** ¿Influye el nivel educacional en la decisión de comprar el producto en promoción?
4. **Estado de Deuda:** ¿Es verdad que los clientes "Sin deuda" actual prefirieron la promoción en mayor medida que aquellos con deudas de 1 o 2 meses?
5. **Monto de Compra:** ¿Cuál es el monto de compra promedio de los clientes que aceptaron la promoción en comparación con los que no lo hicieron?
6. **Uso del Cupo:** ¿Qué grupo de clientes utiliza un mayor porcentaje de su cupo en la tarjeta: los que compran la promoción o los que no?
7. **Historial de Atrasos:** ¿Existe una diferencia en la cantidad histórica de atrasos en pagos entre quienes compraron y quienes no?
8. **Antigüedad del Cliente:** ¿Hay alguna relación entre la antigüedad del cliente y la probabilidad de que compre en una promoción?

3.2 Requerimientos Predictivos

1. **Predicción de Compra en Promoción:** ¿Podemos predecir si un cliente aceptará la oferta de la promoción basándonos en su perfil demográfico (edad, sexo, nivel educacional)?
2. **Identificación de Clientes Clave:** ¿Qué características de un cliente son las más importantes para determinar si comprará el producto en promoción?
3. **Estimación del Monto de Compra:** ¿Es posible estimar cuánto gastará un cliente en una compra futura, considerando su información financiera como el cupo de su tarjeta y sus hábitos de compra?
4. **Predicción de Comportamiento de Pago:** Basándonos en el historial de atrasos y el uso de la tarjeta, ¿podemos anticipar si un cliente estará al día con sus pagos en el futuro?
5. **Segmentación de Clientes por Consumo:** ¿Podemos agrupar a los clientes en diferentes perfiles según los montos que gastan y la cantidad de productos que adquieren?
6. **Predicción por Perfil Financiero:** Si solo usamos la información de la tarjeta de crédito (deuda actual, cupo, atrasos), ¿podemos predecir si un cliente comprará la promoción?
7. **Estimación de Compras Futuras del Producto A**: ¿Se podría estimar la demanda del producto A en promoción para los próximos meses, analizando las tendencias en las fechas de compra?
8. **Predicción de Pagos de Consumos:** ¿Podemos predecir el monto de los pagos que realizarán los clientes por sus consumos con la tarjeta de crédito?

4 Tipos de Datos

4.1 Estructura del Excel entregado

1. **Datos demográficos:**
    - Rango etario: con valores Menor que 30, Entre 30 y 40, Entre 40 y 50, Mayor que 50
    - Nivel educacional: Educ. Media Educ. Técnica, Estudiante Universitario, Educ. Universitaria
    - Sexo: Femenino, Masculino
    - Estado civil: Soltero, Casado, Separado, Viudo (permite valores nulos)
    - Actividad: Dependiente, Empresario, Independiente (permite valores nulos)
2. **Uso de Tarjeta de Crédito:**
    - Estado actual: Sin deuda, Deuda de 1 mes, Deuda de 2 meses
    - Año apertura tarjeta: 2005, 2006, …, 2013, 2014
    - Cupo máximo (de la tarjeta): valor entero, desde $301.500.-hasta $9.605.308.-
    - Porcentaje de uso del cupo: de 0% a 100%
    - Cantidad histórica de atrasos en pagos: valores entre 0 y 25
3. **Consumo:**
    - Veces que compra en promedio al año: valores entre 5 y 30
    - Unidades compradas del producto A: valores entre 0 y 10
    - Unidades compradas del producto B: valores entre 0 y 10
    - Compra producto en promoción: valores sí y no
    - Monto de compras
    - Fecha de compra

4.2 Datos en R

```
> #leer Excel
> productos_promocion <- read_excel("D:/inacap/S6/mineriaDatos/productos_promocion_ev2.xlsx")
> View(productos_promocion)
> datos <- productos_promocion
> #descripcion de los datos
> str(datos)
tibble [4,000 × 16] (S3: tbl_df/tbl/data.frame)
 $ RANGO ETARIO                      : chr [1:4000] "MENOR QUE 30" "MENOR QUE 30" "ENTRE 30 Y 40" "ENTRE 40 Y 50" ...
 $ NIVEL EDUCACIONAL                 : chr [1:4000] "ESTUDIANTE UNIVERSITARIO" "EDUC. UNIVERSITARIA" "EDUC. UNIVERSITARIA" "EDUC. UNIVERSITARIA" ...
 $ SEXO                              : chr [1:4000] "MASCULINO" "FEMENINO" "FEMENINO" "FEMENINO" ...
 $ ESTADO CIVIL                      : chr [1:4000] "SOLTERO" "SOLTERO" "VIUDO" "SOLTERO" ...
 $ ACTIVIDAD                         : chr [1:4000] "DEPENDIENTE" "ESTUDIANTE" "DEPENDIENTE" "DEPENDIENTE" ...
 $ ESTADO ACTUAL                     : chr [1:4000] "DEUDA DE 2 MESES" "SIN DEUDA" "SIN DEUDA" "SIN DEUDA" ...
 $ AÑO APERTURA TARJETA              : num [1:4000] 2014 2006 2011 2006 2012 ...
 $ CUPO MÁXIMO                       : num [1:4000] 301500 301500 4711251 301500 301500 ...
 $ PORCENTAJE DE USO DEL CUPO        : num [1:4000] 1 0 1 0 0.348 ...
 $ VECES QUE COMPRA EN PROMEDIO AL AÑO : num [1:4000] 6 6 9 6 6 7 5 8 7 5 ...
 $ UNIDADES COMPRADAS DEL PRODUCTO A : num [1:4000] 0 2 5 3 9 9 8 0 4 5 ...
 $ UNIDADES COMPRADAS DEL PRODUCTO B : num [1:4000] 5 3 3 0 3 8 8 6 9 4 ...
 $ CANTIDAD HISTORICA DE ATRASOS EN PAGOS: num [1:4000] 16 23 15 1 21 3 10 22 13 15 ...
 $ COMPRA PRODUCTO EN PROMOCIÓN      : chr [1:4000] "SI" "SI" "SI" "NO" ...
 $ MONTO DE COMPRAS                  : num [1:4000] 1250000 2000000 1750000 2000000 250000 750000 1750000 750000 750000 750000 ...
 $ FECHA COMPRA                      : POSIXct[1:4000], format: "2018-01-01" "2018-01-02" "2018-01-02" "2018-01-03" ...
```

Luego de cargar los datos de Excel visualizarlos y asignarlos a una nueva variable ‘datos’, se utilizó la función str(datos) para inspeccionar su estructura, se obtiene un diagnóstico inicial del data frame importado. El conjunto de datos consta de 4,000 observaciones (filas) y 16 variables (columnas).

El análisis de la estructura revela los siguientes puntos clave:

1. **Variables Categóricas Leídas como Texto:** Las variables que representan categorías, como Rango etario, Nivel educacional, Sexo, Estado civil, Actividad, Estado actual y Compra producto en promoción, fueron importadas como tipo carácter (chr).
2. **Variables Numéricas Leídas como Numéricas:** Las variables Año apertura tarjeta, Cupo máximo, Porcentaje de uso del cupo, Veces que compra en promedio al año, Unidades compradas del producto A, Unidades compradas del producto B, Cantidad histórica de

atrasos en pagos y Monto de compras, fueron reconocidas correctamente como numéricas (num).

3. **Variable Fecha y Hora**: La variable Fecha compra es leída con formato POSIXct, que es la forma más común que tiene R para trabajar con fechas.

5 Limpieza y transformación de los datos

5.1 Modificación títulos de los datos

```
> #cambiar nombre de los titulos de las columnas
> colnames(datos) <- c("rango_etario", "educacion", "sexo", "est_civil", "actividad", "est_actual", "anio_apertura", "cupo_max", "porcentaje_uso_cupo", "compras_promedio_anio", "unidades_pro
d_A", "unidades_prod_B", "cant_atrasos", "compra_promo", "compras", "Fecha")
> str(datos)
tibble [4,000 × 16] (S3: tbl_df/tbl/data.frame)
 $ rango_etario         : chr [1:4000] "MENOR QUE 30" "MENOR QUE 30" "ENTRE 30 Y 40" "ENTRE 40 Y 50" ...
 $ educacion            : chr [1:4000] "ESTUDIANTE UNIVERSITARIO" "EDUC. UNIVERSITARIA" "EDUC. UNIVERSITARIA" "EDUC. UNIVERSITARIA" ...
 $ sexo                 : chr [1:4000] "MASCULINO" "FEMENINO" "FEMENINO" "FEMENINO" ...
 $ est_civil            : chr [1:4000] "SOLTERO" "SOLTERO" "VIUDO" "SOLTERO" ...
 $ actividad            : chr [1:4000] "DEPENDIENTE" "ESTUDIANTE" "DEPENDIENTE" "DEPENDIENTE" ...
 $ est_actual           : chr [1:4000] "DEUDA DE 2 MESES" "SIN DEUDA" "SIN DEUDA" "SIN DEUDA" ...
 $ anio_apertura        : num [1:4000] 2014 2006 2011 2006 2012 ...
 $ cupo_max             : num [1:4000] 301500 301500 4711251 301500 301500 ...
 $ porcentaje_uso_cupo  : num [1:4000] 1 0 1 0 0.348 ...
 $ compras_promedio_anio: num [1:4000] 6 6 9 6 6 7 5 8 7 5 ...
 $ unidades_prod_A      : num [1:4000] 0 2 5 3 9 9 8 0 4 5 ...
 $ unidades_prod_B      : num [1:4000] 5 3 3 0 3 8 8 6 9 4 ...
 $ cant_atrasos         : num [1:4000] 16 23 15 1 21 3 10 22 13 15 ...
 $ compra_promo         : chr [1:4000] "SI" "SI" "SI" "NO" ...
 $ compras              : num [1:4000] 1250000 2000000 1750000 2000000 250000 750000 1750000 750000 750000 750000 ...
 $ fecha                : POSIXct[1:4000], format: "2018-01-01" "2018-01-02" "2018-01-02" "2018-01-03" ...
```

Se realizó el cambio de titulo de los nombres de los datos por unos más fáciles para trabajar en R, y se verificó el cambio con la función str(). Los cambios son:

Nombre original	Nombre modificado
Rango etario	rango_etario
Nivel educacional	educacion
Sexo	sexo
Estado civil	est_civil
Actividad	actividad
Estado actual	est_actual
Estado actual Año apertura tarjeta	anio_apertura
Cupo máximo	cupo_max
Porcentaje de uso del cupo	porcentaje_uso_cupo
Veces que compra en promedio al año	compras_promedio_anio
Unidades compradas del producto A	unidades_prod_A
Unidades compradas del producto B	unidades_prod_B
Cantidad histórica de atrasos en pagos	cant_atrasos
Compra producto en promoción	compra_promo
Monto de compras	compras
Fecha compra	fecha

5.2 Eliminar o modificar datos nulos

Una etapa fundamental en cualquier análisis de datos es la verificación de la calidad y la integridad de la información. Antes de realizar cualquier cálculo o visualización, es crucial identificar y manejar los valores ausentes o faltantes (conocidos como NA en R) para garantizar que los resultados del estudio sean precisos y fiables.

Para llevar a cabo esta verificación, se utilizó el siguiente comando en R:

```
> #Ver datos faltantes
> colSums(is.na(datos))
        rango_etario            educacion                sexo           est_civil           actividad          est_actual        anio_apertura            cupo_max 
                   0                    0                    0                  14                 190                   0                   0                   0 
  porcentaje_uso_cupo compras_promedio_anio      unidades_prod_A      unidades_prod_B       cant_atrasos        compra_promo              compras               fecha 
                   0                    0                    0                   0                   0                   0                   0                   0 
```

El resultado indica que la mayoría de las columnas están completas. Sin embargo, se detectaron dos columnas con datos faltantes:

- **est_civil:** Presenta 14 valores ausentes.
- **actividad:** Presenta 190 valores ausentes.

Una vez identificados los datos faltantes, se procedió a aplicar una estrategia de limpieza diferenciada para cada caso, con el objetivo de preservar la mayor cantidad de información valiosa posible.

1. **Para la columna est_civil:** Se tomó la decisión de eliminar las 14 filas completas donde faltaba este dato. Dado que el número de registros afectados es muy bajo en comparación con el total del conjunto de datos, su eliminación no genera un impacto significativo en los resultados globales del análisis.

```
> #Eliminar 14 datos faltantes de est_civil
> datos <- datos[!is.na(datos$est_civil), ]
> colSums(is.na(datos))
        rango_etario            educacion                sexo           est_civil           actividad          est_actual 
                   0                    0                    0                   0                 190                   0 
       anio_apertura             cupo_max   porcentaje_uso_cupo compras_promedio_anio       unidades_prod_A       unidades_prod_B 
                   0                    0                    0                   0                   0                   0 
        cant_atrasos         compra_promo              compras               fecha 
                   0                    0                    0                   0 
```

2. **Para la columna actividad (190 faltantes):** Eliminar 190 registros podría causar una pérdida importante de información. Por lo tanto, se optó por una estrategia donde se reemplazaría los valores faltantes por una nueva categoría: "NO INFORMADO". Esto permite conservar el resto de los datos del cliente en esas filas y, a la vez, tratar a este grupo como una categoría distinta en análisis futuros.

```
#Reemplazar 190 datos faltantes de actividad por "No informado"
datos$actividad[is.na(datos$actividad)] <- "NO INFORMADO"
colSums(is.na(datos))
       rango_etario            educacion                sexo           est_civil           actividad          est_actual 
                  0                    0                    0                   0                   0                   0 
      anio_apertura             cupo_max   porcentaje_uso_cupo compras_promedio_anio       unidades_prod_A       unidades_prod_B 
                  0                    0                    0                   0                   0                   0 
       cant_atrasos         compra_promo              compras               fecha 
                  0                    0                    0                   0 
```

5.3 Transformar datos de char a factor

Una vez que el conjunto de datos está limpio y sin valores ausentes, el siguiente paso es asegurar que cada variable sea interpretada correctamente por R. Las variables que representan categorías (como "sexo" o "nivel educacional") deben ser transformadas de un formato de texto simple (character) a un formato categórico especial llamado factor.

Este paso es crucial porque los modelos estadísticos y las herramientas de visualización en R requieren que las categorías estén definidas como factor para poder realizar agrupaciones, comparaciones y cálculos de manera correcta.

Para realizar esta conversión, se aplicó la función as.factor() a todas las columnas que, por su naturaleza, son categóricas, y para confirmar que la transformación se realizó con éxito, se ejecutó el comando str(datos), que muestra la estructura interna del conjunto de datos.

```
> #Transformar datos char a factor
> datos$rango_etario <- as.factor(datos$rango_etario)
> datos$educacion <- as.factor(datos$educacion)
> datos$sexo <- as.factor(datos$sexo)
> datos$est_civil <- as.factor(datos$est_civil)
> datos$actividad <- as.factor(datos$actividad)
> datos$est_actual <- as.factor(datos$est_actual)
> datos$compra_promo <- as.factor(datos$compra_promo)
> str(datos)
tibble [3,986 × 16] (S3: tbl_df/tbl/data.frame)
 $ rango_etario         : Factor w/ 4 levels "ENTRE 30 Y 40",..: 4 4 1 2 3 1 1 1 2 1 ...
 $ educacion            : Factor w/ 4 levels "EDUC. MEDIA",..: 4 3 3 3 3 2 2 3 2 4 ...
 $ sexo                 : Factor w/ 2 levels "FEMENINO","MASCULINO": 2 1 1 1 2 1 1 1 1 2 ...
 $ est_civil            : Factor w/ 4 levels "CASADO","SEPARADO",..: 3 3 4 3 3 1 3 3 2 1 ...
 $ actividad            : Factor w/ 4 levels "DEPENDIENTE",..: 1 3 1 1 1 1 1 1 1 1 ...
 $ est_actual           : Factor w/ 3 levels "DEUDA DE 1 MES",..: 2 3 3 3 3 3 3 3 3 3 ...
 $ anio_apertura        : num [1:3986] 2014 2006 2011 2006 2012 ...
 $ cupo_max             : num [1:3986] 301500 301500 4711251 301500 301500 ...
 $ porcentaje_uso_cupo  : num [1:3986] 1 0 1 0 0.348 ...
 $ compras_promedio_anio: num [1:3986] 6 6 9 6 6 7 5 8 7 5 ...
 $ unidades_prod_A      : num [1:3986] 0 2 5 3 9 9 8 0 4 5 ...
 $ unidades_prod_B      : num [1:3986] 5 3 3 0 3 8 8 6 9 4 ...
 $ cant_atrasos         : num [1:3986] 16 23 15 1 21 3 10 22 13 15 ...
 $ compra_promo         : Factor w/ 2 levels "NO","SI": 2 2 2 1 2 2 1 1 2 2 ...
 $ compras              : num [1:3986] 1250000 2000000 1750000 2000000 250000 750000 1750000 750000 750000 750000 ...
 $ fecha                : POSIXct[1:3986], format: "2018-01-01" "2018-01-02" "2018-01-02" "2018-01-03" ...
```

La salida de la consola confirma varios puntos importantes:

- **Dimensiones del Dataset:** La primera línea indica que, tras la limpieza, nuestro conjunto de datos final para el análisis consta de 3,986 observaciones (filas) y 16 variables (columnas).
- **Correcta Transformación:** Las variables de interés ahora aparecen claramente marcadas con el tipo Factor. Por ejemplo, la salida sexo: Factor w/ 2 levels "FEMENINO","MASCULINO" confirma que R ahora entiende la variable sexo no como simple texto, sino como una categoría con dos niveles definidos.
- **Tipos de Datos Mixtos:** Se puede observar que las variables no transformadas conservan su tipo de dato original, como num para las variables numéricas (ej. cupo_max, compras) o POSIXct para la variable de fecha.

Con esta transformación, el conjunto de datos queda estructuralmente preparado y es completamente apto para la fase de análisis descriptivo y la construcción de modelos predictivos.

6 Análisis y Gráficos

6.1 Requerimientos descriptivos

6.1.1 Requerimiento 1: Perfil del Comprador

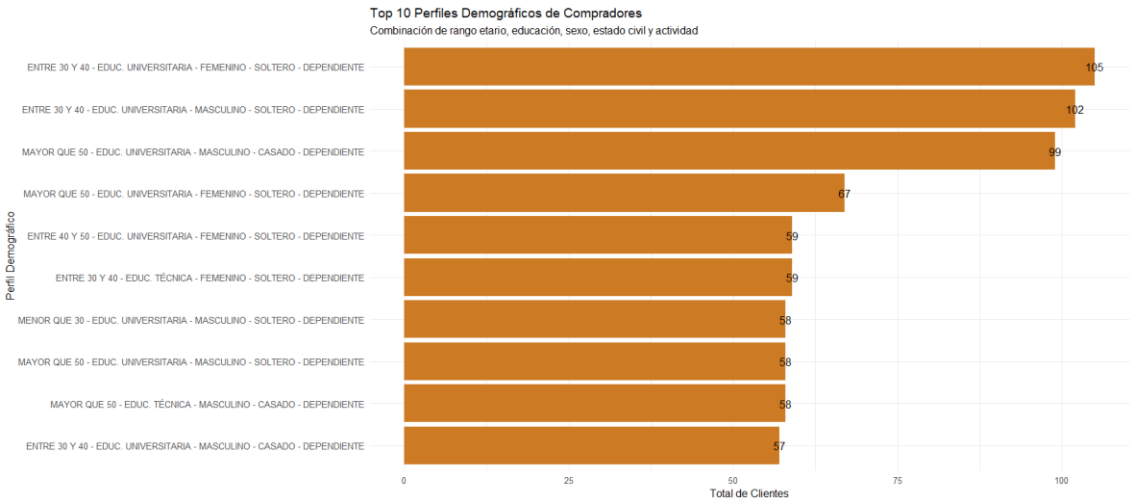
Pregunta: ¿Cuál es el perfil demográfico principal (rango etario, nivel educacional, sexo) de los clientes que sí compraron el producto en promoción?

Respuesta: La respuesta es que el perfil principal corresponde a clientes de entre 30 y 40 años, solteros, dependientes y con educación universitaria, tanto mujeres como hombres. Curiosamente, no hay una dominancia muy marcada de un sexo sobre el otro en los perfiles más altos, ya que los dos primeros lugares están ocupados por mujeres y hombres con características muy similares y en cantidades casi iguales. El análisis no solo identifica un único perfil principal, sino que presenta un ranking de los 10 perfiles demográficos más importantes.

```
> #1. Perfil del comprador
> perfil_demografico_combinado <- datos %>%
+   filter(compra_promo == "SI") %>%
+   count(rango_etario, educacion, sexo, est_civil, actividad, sort = TRUE, name = "total_clientes")
>
> top_10_perfiles <- head(perfil_demografico_combinado, 10)
> print(top_10_perfiles)
# A tibble: 10 × 6
   rango_etario  educacion           sexo      est_civil actividad   total_clientes
   <fct>         <fct>               <fct>     <fct>     <fct>                <int>
 1 ENTRE 30 Y 40 EDUC. UNIVERSITARIA FEMENINO  SOLTERO   DEPENDIENTE            105
 2 ENTRE 30 Y 40 EDUC. UNIVERSITARIA MASCULINO SOLTERO   DEPENDIENTE            102
 3 MAYOR QUE 50  EDUC. UNIVERSITARIA MASCULINO CASADO    DEPENDIENTE             99
 4 MAYOR QUE 50  EDUC. UNIVERSITARIA FEMENINO  SOLTERO   DEPENDIENTE             67
 5 ENTRE 30 Y 40 EDUC. TÉCNICA       FEMENINO  SOLTERO   DEPENDIENTE             59
 6 ENTRE 40 Y 50 EDUC. UNIVERSITARIA FEMENINO  SOLTERO   DEPENDIENTE             59
 7 MAYOR QUE 50  EDUC. TÉCNICA       MASCULINO CASADO    DEPENDIENTE             58
 8 MAYOR QUE 50  EDUC. UNIVERSITARIA MASCULINO SOLTERO   DEPENDIENTE             58
 9 MENOR QUE 30  EDUC. UNIVERSITARIA MASCULINO SOLTERO   DEPENDIENTE             58
10 ENTRE 30 Y 40 EDUC. UNIVERSITARIA MASCULINO CASADO    DEPENDIENTE             57
```

Análisis del Código: El código busca encontrar las combinaciones de características demográficas más comunes entre los clientes que adquirieron la promoción. Para lograrlo, sigue los siguientes pasos:

- **Filtrado de Clientes:** Primero, utiliza la función filter() para seleccionar exclusivamente al subgrupo de clientes que respondieron "SI" a la variable compra_promo. Esto asegura que el análisis se centre únicamente en el público objetivo.
- **Conteo y Agrupación:** Luego, con la función count(), agrupa a estos clientes según cinco variables demográficas (rango_etario, educacion, sexo, est_civil y actividad). El código cuenta cuántos clientes pertenecen a cada combinación única y las ordena de la más a la menos frecuente.
- **Selección del Top 10**: Finalmente, se utiliza head() para extraer los 10 perfiles más comunes, que son los que se muestran en la tabla de la consola y se utilizan para generar el gráfico.



Análisis del Gráfico: El gráfico visualiza de manera clara y ordenada la información obtenida por el código, permitiendo una interpretación rápida de los resultados. El gráfico es un diagrama de barras horizontales que presenta los 10 perfiles demográficos más frecuentes entre los compradores de la promoción. Cada barra representa un perfil único, y su longitud es proporcional al número de clientes que lo componen.

6.1.2 Requerimiento 2: Género y Compra

Pregunta: Comparar el porcentaje de hombres versus mujeres que compraron el producto en promoción. ¿Las mujeres compraron más que los hombres?

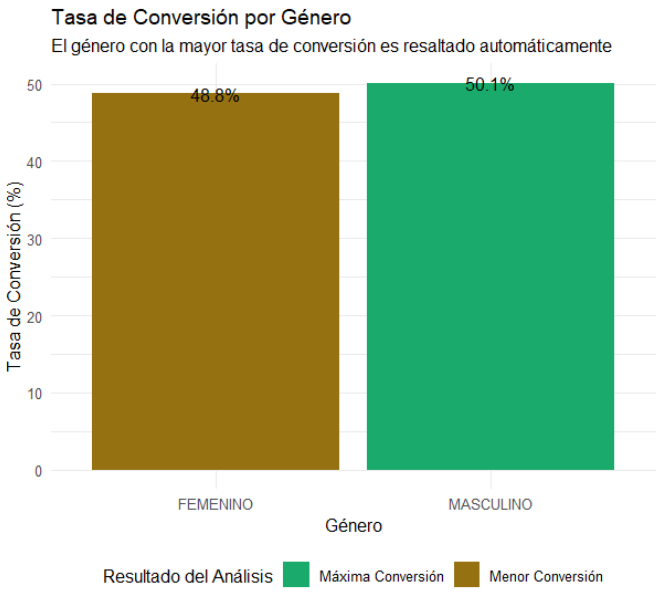
Respuesta: En números absolutos, sí, compraron más mujeres (1001) que hombres (968). Aunque en cifras totales el número de compradoras fue mayor que el de compradores, esto se debe a que la base total de clientes femeninos (2053) es también mayor. El requerimiento pide comparar el porcentaje, y en este aspecto, los hombres muestran una propensión a la compra ligeramente mayor. En conclusión, si bien se vendió la promoción a más mujeres en total, un hombre seleccionado al azar tenía una probabilidad más alta (50.1%) de comprar el producto que una mujer seleccionada al azar (48.8%).

```
> tasa_por_genero <- datos %>%
+   group_by(sexo) %>%
+   summarise(
+     total_clientes = n(),
+     total_compradores = sum(compra_promo == "SI"),
+     tasa_conversion = (total_compradores / total_clientes) * 100
+   ) %>%
+   ungroup() %>%
+   mutate(
+     destacado = if_else(tasa_conversion == max(tasa_conversion), "Máxima Conversión", "Menor Conversión")
+   )
> print(tasa_por_genero)
# A tibble: 2 x 5
  sexo      total_clientes total_compradores tasa_conversion destacado
  <fct>              <int>             <int>           <dbl> <chr>
1 FEMENINO            2053              1001            48.8 Menor Conversión
2 MASCULINO           1933               968            50.1 Máxima Conversión
```

Análisis del Código: El objetivo de este bloque de código es calcular y comparar la tasa de conversión para cada género.

- **Agrupación por Género:** Primero, el código utiliza group_by(sexo) para separar el conjunto de datos en dos grupos: uno para "FEMENINO" y otro para "MASCULINO".
- **Cálculo de Métricas:** Para cada uno de estos grupos, summarise() calcula tres valores clave: total_clientes (El número total de personas en ese género), total_compradores (El número de personas de ese género que sí compraron) y la tasa_conversion (El porcentaje que resulta de dividir los compradores entre el total de clientes).

- **Etiquetado Automático:** Finalmente, el código añade una columna llamada destacado que etiqueta automáticamente al género con la tasa más alta como "Máxima Conversión", a través de la estructura de control if_else.



Análisis del Gráfico: El gráfico de barras es una herramienta visual efectiva que compara directamente la tasa de conversión entre hombres y mujeres. El gráfico presenta dos barras, una para cada género. La altura de cada barra representa su tasa de conversión. Se observa que la barra "MASCULINO" es ligeramente más alta que la barra "FEMENINO", indicando una mayor tasa de conversión.

6.1.3 Requerimiento 3: Educación y Compra

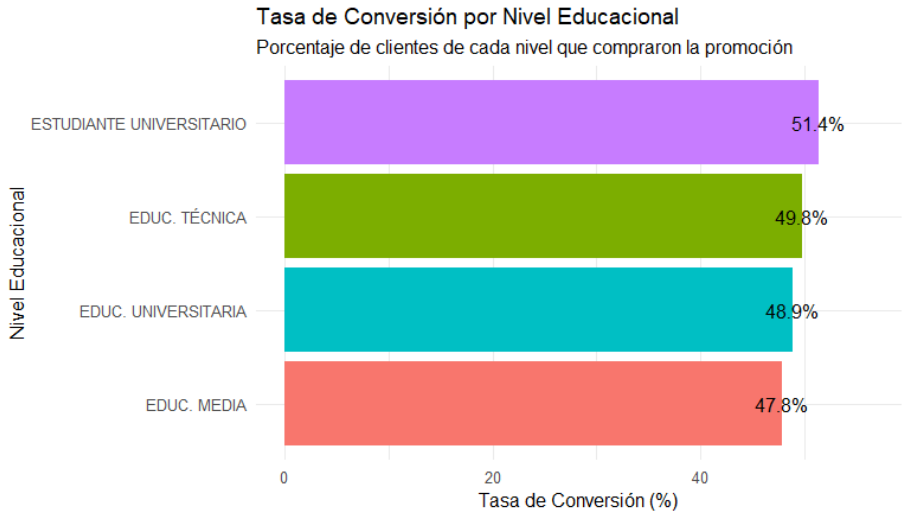
Pregunta: ¿Influye el nivel educacional en la decisión de comprar el producto en promoción?

Respuesta: Sí, el nivel educacional influye en la decisión de compra. El análisis muestra que los estudiantes universitarios presentan la tasa de compra más alta, seguidos por los profesionales técnicos y universitarios, mientras que el nivel de educación media tiene la tasa más baja. Aunque existe una influencia, es importante notar que las diferencias entre los grupos no son drásticas. Sin embargo, la tendencia es consistente y demuestra que, a mayor nivel de formación educativa, hay una ligera mayor probabilidad de compra.

```
> tasa_por_educacion <- datos %>%
+   group_by(educacion) %>%
+   summarise(
+     total_clientes = n(),
+     total_compradores = sum(compra_promo == "SI"),
+     tasa_conversion = (total_compradores / total_clientes) * 100
+   ) %>%
+   arrange(desc(tasa_conversion))
> print(tasa_por_educacion)
# A tibble: 4 × 4
  educacion               total_clientes total_compradores tasa_conversion
  <fct>                            <int>             <int>           <dbl>
1 ESTUDIANTE UNIVERSITARIO           329               169            51.4
2 EDUC. TÉCNICA                     1236               616            49.8
3 EDUC. UNIVERSITARIA               2352              1151            48.9
4 EDUC. MEDIA                         69                33            47.8
```

Análisis del Código: El código busca medir y comparar la efectividad de la promoción a través de los diferentes niveles educacionales de los clientes. La lógica es muy similar a la del análisis de género.

- **Agrupación por Nivel Educacional:** Se utiliza group_by(educacion) para segmentar a todos los clientes en cuatro grupos distintos según su nivel de estudios (Media, Técnica, Universitaria y Estudiante Universitario).
- **Cálculo de Tasa de Conversión**: Para cada uno de estos grupos, el comando summarise() calcula el total de clientes, el total de compradores y, lo más importante, la tasa de conversión (el porcentaje de compradores dentro de ese grupo).
- **Ordenamiento de Resultados:** Finalmente, arrange(desc(tasa_conversion)) ordena la tabla de resultados de mayor a menor según la tasa de conversión, lo que permite identificar rápidamente qué grupo educativo respondió mejor a la campaña.



Análisis del Gráfico: El gráfico presenta cuatro barras, una por cada nivel educacional. La longitud de cada barra representa la tasa de conversión de ese grupo. Las barras están ordenadas de arriba hacia abajo, desde la más alta a la más baja tasa de conversión. Las etiquetas sobre cada barra muestran el porcentaje exacto, lo que permite cuantificar la diferencia entre los grupos. Por ejemplo, se ve que la tasa de los estudiantes universitarios (51.4%) es casi 4 puntos porcentuales mayor que la de los clientes con educación media (47.8%).

6.1.4 Requerimiento 4: Estado de Deuda

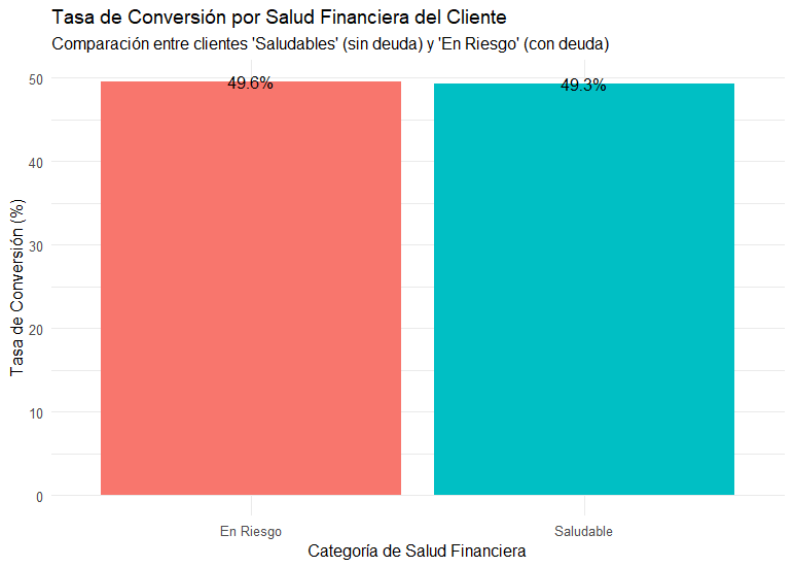
Pregunta: ¿Es verdad que los clientes "Sin deuda" actual prefirieron la promoción en mayor medida que aquellos con deudas de 1 o 2 meses?

Respuesta: No, no es verdad. Contrario a la hipótesis, el análisis demuestra que no hay una diferencia significativa en el comportamiento de compra entre ambos grupos. Los datos demuestran que los clientes "Sin deuda" no prefirieron la promoción en mayor medida. La tasa de conversión del grupo "Saludable" (49.3%) y la del grupo "En Riesgo" (49.6%) son prácticamente iguales. La conclusión es que el estado de deuda actual de un cliente (siempre que sea una deuda menor a 3 meses) no es un factor determinante en su decisión de aceptar o no la oferta de la promoción. Ambos segmentos de clientes son igualmente propensos a comprar.

```
> #4. Estados de deuda
> datos_con_salud <- datos %>%
+   mutate(
+     salud_financiera = case_when(
+       est_actual == "SIN DEUDA"      ~ "Saludable",
+       est_actual == "DEUDA DE 1 MES" ~ "En Riesgo",
+       est_actual == "DEUDA DE 2 MESES" ~ "En Riesgo",
+       TRUE                           ~ "Otro"
+     )
+   )
>
> tasa_por_salud <- datos_con_salud %>%
+   group_by(salud_financiera) %>%
+   summarise(
+     total_clientes = n(),
+     total_compradores = sum(compra_promo == "SI"),
+     tasa_conversion = (total_compradores / total_clientes) * 100
+   ) %>%
+   arrange(desc(tasa_conversion))
> print(tasa_por_salud)
# A tibble: 2 × 4
  salud_financiera total_clientes total_compradores tasa_conversion
  <chr>                    <int>             <int>           <dbl>
1 En Riesgo                  768               381            49.6
2 Saludable                 3218              1588            49.3
```

Análisis del Código: Para responder al requerimiento, el código primero realiza una transformación de los datos y luego calcula la tasa de conversión.

- **Creación de una Variable Simplificada:** El primer paso utiliza mutate() para crear una nueva columna llamada salud_financiera. Esta columna agrupa a los clientes en dos categorías más simples: Saludable (clientes que figuran como "SIN DEUDA") y En Riesgo (clientes que tienen "DEUDA DE 1 MES" o "DEUDA DE 2 MESES). A través de la estructura de control case_when.
- **Cálculo de la Tasa de Conversión**: Una vez creadas las nuevas categorías, el código agrupa a los clientes por salud_financiera y calcula la tasa de conversión para cada grupo (el porcentaje de clientes que compraron la promoción).



Análisis del Gráfico: El gráfico presenta dos barras, una para la categoría "En Riesgo" y otra para "Saludable". La altura de cada barra indica la tasa de conversión para ese segmento de clientes. La observación más importante es que ambas barras tienen una altura casi idéntica. Esto demuestra visualmente que no hay una diferencia relevante en su comportamiento de compra. Las etiquetas de porcentaje sobre cada barra confirman la similitud: 49.6% para el grupo "En Riesgo" y 49.3% para el "Saludable". Esta mínima diferencia de 0.3 puntos porcentuales es insignificante en la práctica.

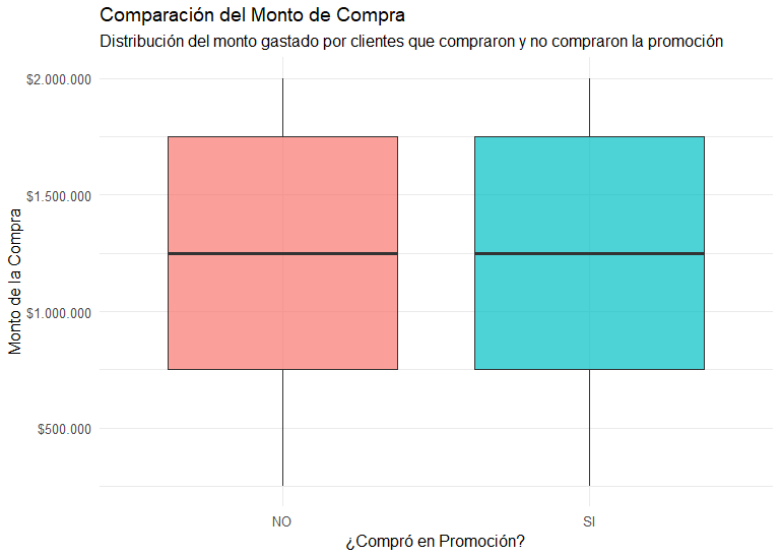
6.1.5 Requerimiento 5: Monto de Compra

Pregunta: ¿Cuál es el monto de compra promedio de los clientes que aceptaron la promoción en comparación con los que no lo hicieron?

Respuesta: El monto de compra promedio es prácticamente idéntico para ambos grupos. El promedio de compra para quienes NO aceptaron la promoción fue de $1,206,867. El promedio de compra para quienes SÍ aceptaron fue de $1,202,006. Más importante aún, la mediana es exactamente la misma para ambos grupos: $1,250,000. Esto lleva a la conclusión de que el hábito de consumo de un cliente, ya sea que gaste mucho o poco, no parece ser un factor que determine si aprovechará o no la promoción.

```
> resumen_monto_compras <- datos %>%
+   group_by(compra_promo) %>%
+   summarise(
+     promedio = mean(compras, na.rm = TRUE),
+     mediana = median(compras, na.rm = TRUE),
+     desviacion_estandar = sd(compras, na.rm = TRUE),
+     total_clientes = n()
+   )
> print(resumen_monto_compras)
# A tibble: 2 x 5
  compra_promo promedio mediana desviacion_estandar total_clientes
  <fct>           <dbl>   <dbl>               <dbl>          <int>
1 NO           1206867. 1250000             558849.           2017
2 SI           1202006. 1250000             583354.           1969
```

Análisis del Código. Primero, el código agrupa a los clientes en dos categorías: los que compraron la promoción ("SI") y los que no ("NO"). Luego, para cada grupo, calcula un resumen estadístico de sus montos de compra (compras). Este resumen no solo incluye el promedio, sino también la mediana (el valor central, menos sensible a valores extremos), la desviación estándar y el total de clientes, ofreciendo una comparación numérica detallada.



Análisis del Gráfico: El gráfico presenta dos "cajas" una al lado de la otra, representando a los clientes que compraron ("SI") y no compraron ("NO") la promoción. La característica más notoria es que ambas cajas son virtualmente idénticas. La línea gruesa central (la mediana) está a la misma altura en ambos casos, indicando que el monto de compra central es el mismo. La altura de las cajas (el rango intercuartílico, que contiene al 50% central de los clientes) y la extensión de los "bigotes" también son muy similares. Esta similitud visual demuestra de forma contundente que no hay una diferencia real en los patrones de gasto entre quienes aceptan la oferta y quiénes no.

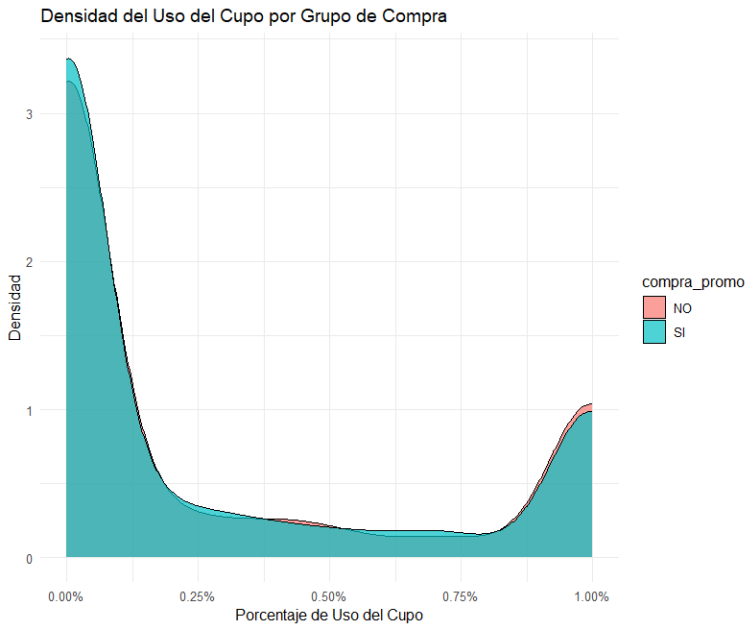
6.1.6 Requerimiento 6: Uso del Cupo

Pregunta: ¿Qué grupo de clientes utiliza un mayor porcentaje de su cupo en la tarjeta: los que compran la promoción o los que no?

Respuesta: El análisis demuestra que el uso del cupo no es un factor diferenciador para la compra de la promoción. El promedio de uso para quienes NO compraron fue del 27.4%. El promedio de uso para quienes SÍ compraron fue del 26.4%. Más revelador aún es el dato de la mediana, que es del 0% para ambos grupos. Esto significa que al menos la mitad de los clientes de cada categoría no utiliza su cupo en absoluto. Como conclusión es que el patrón de uso del cupo de la tarjeta es el mismo tanto para los clientes que aceptan la promoción como para los que la rechazan. Por lo tanto, esta variable no sirve para distinguir a los dos grupos.

```
> resumen_uso_cupo <- datos %>%
+   group_by(compra_promo) %>%
+   summarise(
+     promedio_uso = mean(porcentaje_uso_cupo, na.rm = TRUE),
+     mediana_uso = median(porcentaje_uso_cupo, na.rm = TRUE),
+     desviacion_estandar = sd(porcentaje_uso_cupo, na.rm = TRUE)
+   )
> print(resumen_uso_cupo)
# A tibble: 2 x 4
  compra_promo promedio_uso mediana_uso desviacion_estandar
  <fct>              <dbl>       <dbl>               <dbl>
1 NO                 0.274           0               0.404
2 SI                 0.264           0               0.394
```

Análisis del Código: Primero, el código agrupa a los clientes según si compraron la promoción ("SI" o "NO"). A continuación, calcula las estadísticas centrales para la variable porcentaje_uso_cupo en cada grupo: el promedio de uso, la mediana y la desviación estándar. Esto permite una comparación numérica directa.



Análisis del Gráfico: El gráfico presenta dos curvas de densidad superpuestas. Una representa a los clientes que compraron la promoción ("SI") y la otra a los que no ("NO"). La altura de la curva en cualquier punto indica la concentración de clientes con ese porcentaje de uso. La conclusión principal es inmediata: las dos curvas son casi idénticas y están perfectamente superpuestas. Ambas tienen un pico muy pronunciado en el 0%, lo que indica que una gran cantidad de clientes en ambos grupos no utiliza su cupo o lo utiliza muy poco. A medida que el porcentaje de uso aumenta, la cantidad de clientes disminuye drásticamente en ambos casos.

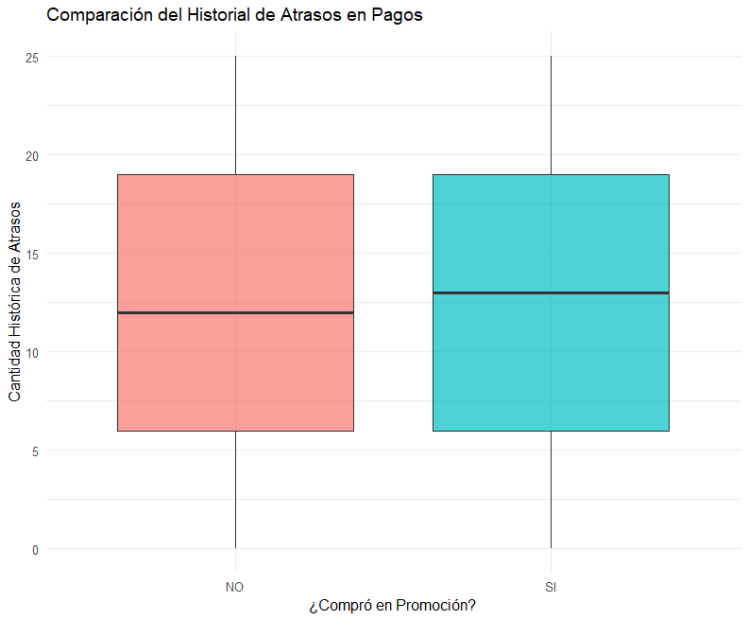
6.1.7 Requerimiento 7: Historial de Atrasos

Pregunta: ¿Existe una diferencia en la cantidad histórica de atrasos en pagos entre quienes compraron y quienes no?

Respuesta: No, no existe una diferencia significativa. El historial de atrasos en los pagos es prácticamente idéntico para ambos grupos de clientes. El promedio de atrasos para quienes NO compraron fue de 12.4. El promedio de atrasos para quienes SÍ compraron fue de 12.3. La mediana de atrasos también fue muy similar: 12 para el grupo "NO" y 13 para el grupo "SI". Se concluye que el historial de pagos de un cliente no es un factor que determine si va a aceptar la promoción. Tanto los clientes que históricamente son más puntuales como los que tienen más atrasos en sus pagos reaccionan de manera similar ante la oferta.

```
> resumen_atrasos <- datos %>%
+   group_by(compra_promo) %>%
+   summarise(
+     promedio_atrasos = mean(cant_atrasos, na.rm = TRUE),
+     mediana_atrasos = median(cant_atrasos, na.rm = TRUE),
+     desviacion_estandar = sd(cant_atrasos, na.rm = TRUE)
+   )
> print(resumen_atrasos)
# A tibble: 2 x 4
  compra_promo promedio_atrasos mediana_atrasos desviacion_estandar
  <fct>                   <dbl>           <dbl>               <dbl>
1 NO                       12.4              12                7.39
2 SI                       12.3              13                7.48
```

Análisis del Código: Primero, se agrupan los datos en dos segmentos: clientes que compraron la promoción ("SI") y los que no ("NO"). A continuación, para cada segmento, se calculan las métricas estadísticas centrales de la variable cant_atrasos: el promedio de atrasos, la mediana (el valor que se encuentra en la mitad de los datos) y la desviación estándar.



Análisis del Gráfico: El gráfico contiene dos "cajas" que representan a los dos grupos. Cada caja resume la distribución de la cantidad de atrasos históricos de sus miembros. La conclusión visual es inmediata: las dos cajas son casi idénticas. La línea central (mediana) se encuentra a una altura muy similar en ambos casos. El tamaño de las cajas (que representa al 50% central de los clientes) y la longitud de los "bigotes" son prácticamente iguales. Esta gran similitud entre las cajas indica que la distribución de la cantidad de atrasos es la misma para ambos grupos.

6.1.8 Requerimiento 8: Antigüedad del Cliente

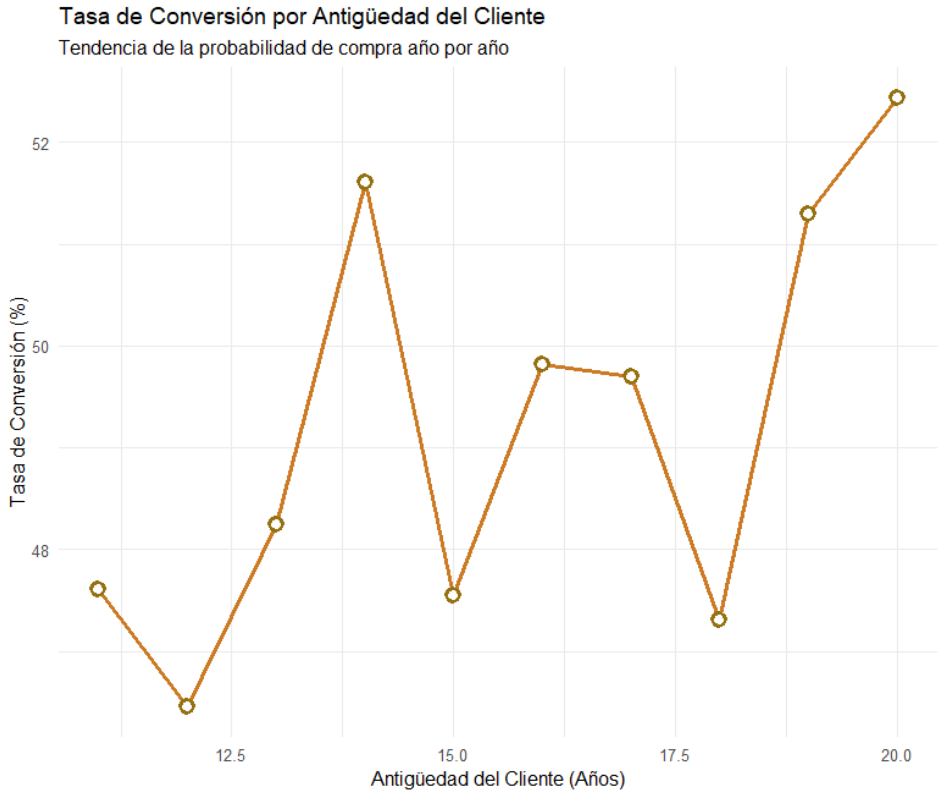
Pregunta: ¿Hay alguna relación entre la antigüedad del cliente y la probabilidad de que compre en una promoción?

Respuesta: Sí, existe una relación. El análisis confirma que existe una correlación positiva entre la antigüedad del cliente y su propensión a comprar en promoción. Aunque la relación no es perfectamente lineal y muestra variaciones año a año, la tendencia general es clara: los clientes más leales y con más años en la compañía son, en promedio, más receptivos a las ofertas. La tasa de conversión pasa de un 46.4% en los clientes de 12 años a un 52.4% en los de 20 años.

```
> datos_analisis_final <- datos %>%
+   mutate(
+     antiguedad = (2025 - anio_apertura),
+     compra_promo_limpia = str_to_lower(compra_promo)
+   )
> tasa_por_antiguedad_anual <- datos_analisis_final %>%
+   group_by(antiguedad) %>%
+   summarise(
+     tasa_conversion = mean(compra_promo_limpia == "si") * 100
+   )
> print(tasa_por_antiguedad_anual)
# A tibble: 10 x 2
   antiguedad tasa_conversion
        <dbl>           <dbl>
 1         11            47.6
 2         12            46.4
 3         13            48.2
 4         14            51.6
 5         15            47.5
 6         16            49.8
 7         17            49.7
 8         18            47.3
 9         19            51.3
10         20            52.4
```

Análisis del Código: Para analizar la relación entre la antigüedad y la compra, el código primero necesita crear la variable "antigüedad" y luego calcular la tasa de conversión para cada año.

- C**álculo de la Antigüedad:** El primer paso es crear una nueva columna llamada antiguedad. El código la calcula restando el año de apertura de la cuenta (anio_apertura) de un año de referencia (2025). Este proceso transforma un dato existente en una variable nueva y más útil para el análisis.
- **Cálculo de la Tasa de Conversión Anual:** Una vez que cada cliente tiene asignada su antigüedad en años, el código los agrupa por este valor. Luego, para cada año de antigüedad (todos los clientes de 11 años, todos los de 12, etc.), calcula la tasa de conversión promedio, es decir, el porcentaje de clientes de esa antigüedad específica que compraron la promoción.



Análisis del Gráfico: El gráfico traza la tasa de conversión (eje Y) para cada año de antigüedad del cliente (eje X). Cada punto representa la tasa de compra para los clientes que tienen exactamente esa cantidad de años con la empresa.

La característica más importante es la tendencia general ascendente de la línea. Aunque presenta fluctuaciones (subidas y bajadas anuales), la dirección general de la línea va de abajo a la izquierda hacia arriba a la derecha.

Se puede observar que los clientes con menor antigüedad (11-13 años) tienen tasas de conversión que rondan el 47-48%, mientras que los clientes más antiguos (19-20 años) muestran las tasas más altas, superando el 51%. El punto más alto se da en los clientes con 20 años de antigüedad (52.4%).

6.2 Requerimientos predictivos

6.2.1 Preparación de los datos

Antes de construir cualquier modelo predictivo, es una práctica estándar y fundamental dividir el conjunto de datos principal en dos subconjuntos independientes. Esta separación es crucial para desarrollar un modelo robusto y evaluar su verdadero poder predictivo. Los dos conjuntos son:

- **Conjunto de Entrenamiento (Training Set):** Corresponde a la mayor parte de los datos (generalmente 70-80%). Se utiliza para "entrenar" al modelo de machine learning, es decir, para que aprenda los patrones y relaciones presentes en los datos.
- **Conjunto de Prueba (Test Set):** Es una porción más pequeña de los datos (20-30%) que el modelo no ve durante su entrenamiento. Se reserva para el final, para simular cómo se comportaría el modelo con datos nuevos y así evaluar su rendimiento de manera objetiva.

```
> View(datos_analisis_final)
> datos_modelo <- datos %>%
+   mutate(compra_promo = str_to_lower(compra_promo))
> set.seed(123)
> indices_entrenamiento <- createDataPartition(datos_modelo$compra_promo, p = 0.7, list = FALSE)
> datos_entrenamiento <- datos_modelo[indices_entrenamiento, ]
> datos_prueba <- datos_modelo[-indices_entrenamiento, ]
>
> cat("Datos listos. Entrenamiento:", nrow(datos_entrenamiento), "filas. Prueba:", nrow(datos_prueba),
"filas.\n")
Datos listos. Entrenamiento: 2791 filas. Prueba: 1195 filas.
```

Análisis del Código: El anterior código realiza la división con los siguientes pasos:

- **Preparación:** Primero, se asegura de que la variable objetivo (compra_promo) esté en minúsculas ("si"/"no") para mantener la consistencia.
- **Reproducibilidad:** Se utiliza set.seed(123) para fijar el punto de partida del generador de números aleatorios. Esto garantiza que, si el código se ejecuta de nuevo, la división de los datos será exactamente la misma, lo que es vital para que los resultados sean reproducibles.
- **División Estratificada:** Se usa la función createDataPartition para dividir los datos en una proporción de 70% para entrenamiento y 30% para prueba.
- **Asignación:** Finalmente, el código crea los dos dataframes (datos_entrenamiento y datos_prueba) basándose en los índices generados en el paso anterior.

Resultado y Verificación: Para confirmar que la división se realizó correctamente, el código imprime el número de filas de cada nuevo conjunto de datos.

El resultado confirma que el dataset original fue exitosamente dividido en un conjunto de entrenamiento de 2,791 clientes y un conjunto de prueba de 1,195 clientes. Con esta preparación finalizada, el proyecto está listo para comenzar la fase de entrenamiento y evaluación de los modelos predictivos.

6.2.2 Requerimiento 1: Predicción de Compra en Promoción

Pregunta: ¿Podemos predecir si un cliente aceptará la oferta de la promoción basándonos en su perfil demográfico (edad, sexo, nivel educacional)?

Respuesta: Sí, es posible construir un modelo predictivo, pero su capacidad de predicción es muy baja. La precisión del modelo del 51.88% es extremadamente baja y apenas supera el 50.63% que se lograría por azar (NIR). Esto demuestra que las variables demográficas por sí solas (edad, sexo, educación, etc.) no contienen suficiente información para distinguir de manera fiable entre los clientes que comprarán la promoción y los que no.

```
> predicciones_demograficas <- predict(modelo_arbol_demografico, datos_prueba, type = "class")
> niveles_completos <- c("no", "si")
> predicciones_factor_dem <- factor(predicciones_demograficas, levels = niveles_completos)
> reales_factor_dem <- factor(datos_prueba$compra_promo, levels = niveles_completos)
> matriz_confusion_demografica <- confusionMatrix(predicciones_factor_dem, reales_factor_dem)
> print(matriz_confusion_demografica)
Confusion Matrix and Statistics

          Reference
Prediction  no  si
        no 438 408
        si 167 182

               Accuracy : 0.5188
                 95% CI : (0.4901, 0.5475)
    No Information Rate : 0.5063
    P-Value [Acc > NIR] : 0.2008

                  Kappa : 0.0326

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.7240
            Specificity : 0.3085
         Pos Pred Value : 0.5177
         Neg Pred Value : 0.5215
             Prevalence : 0.5063
         Detection Rate : 0.3665
   Detection Prevalence : 0.7079
      Balanced Accuracy : 0.5162

       'Positive' Class : no
```

Análisis del Código: El código ejecuta el proceso completo de modelamiento: entrenar un modelo, visualizarlo, usarlo para predecir y evaluar su rendimiento.

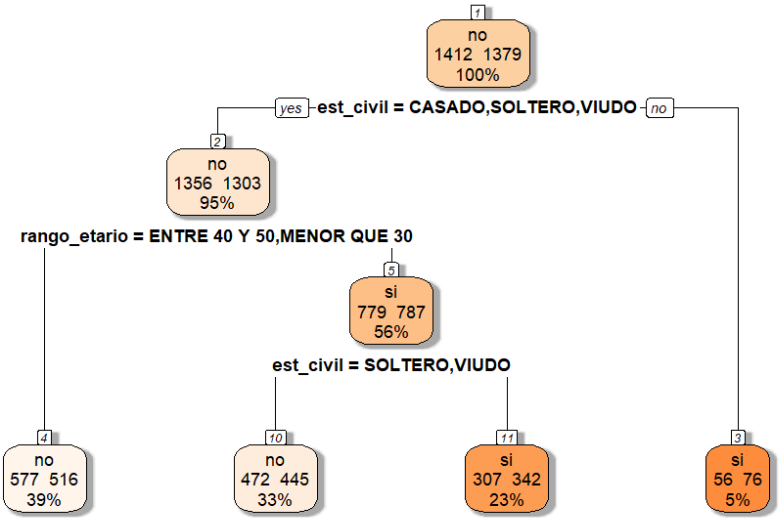
- **Entrenamiento del Modelo:** Se utiliza la función rpart para construir un Árbol de Decisión. Este tipo de modelo aprende una serie de reglas (similares a un diagrama de flujo) para clasificar a los clientes. La fórmula compra_promo ~ ... le indica al modelo que debe predecir la variable compra_promo utilizando cinco variables demográficas. Es crucial destacar que el modelo se entrena exclusivamente con datos_entrenamiento.
- **Predicción:** Una vez entrenado, el modelo se utiliza para predecir el comportamiento de los clientes en datos_prueba, un conjunto de datos que nunca antes ha visto.
- **Evaluación:** Finalmente, el código genera una Matriz de Confusión. Esta es una tabla que compara las predicciones del modelo con los resultados reales del conjunto de prueba, permitiendo medir de forma objetiva qué tan preciso fue.

Análisis de la Matriz de Confusión: Esta tabla es el resultado final que mide el rendimiento del modelo.

- **Lectura de la Matriz:** Aciertos: El modelo predijo correctamente a 438 clientes que "no" comprarían y a 182 que "si" comprarían. Errores: Se equivocó con 408 clientes (predijo "no" pero en realidad compraron "si") y con 167 clientes (predijo "si" pero compraron "no").
- **Métricas Clave:** Accuracy: 0.5188 (51.88%). Esta es la métrica principal. Indica que el modelo acertó en casi el 52% de sus predicciones. No Information Rate: 0.5063 (50.63%). Este valor representa la precisión que se obtendría si simplemente se predijera siempre la

clase más común ("no"). P-Value [Acc > NIR]: 0.2008. Un valor alto (mayor a 0.05) indica que la precisión del modelo no es estadísticamente superior a la de simplemente adivinar.

Gráfico 1: Predicción de Compra Usando Solo Perfil Demográfico



Análisis del Gráfico: El gráfico muestra el "cerebro" del modelo: el conjunto de reglas que aprendió para tomar decisiones. Es un diagrama de flujo. Se comienza en el nodo superior (la raíz) y se desciende por las ramas según las características de cada cliente.

El árbol hace una serie de preguntas. La primera y más importante (la de más arriba) es sobre la actividad del cliente. Si es "DEPENDIENTE", se sigue el camino de la izquierda; si no, el de la derecha. El proceso se repite con otras variables como educacion y rango_etario hasta llegar a un nodo final. Cada nodo final en la parte inferior del árbol representa una predicción final ("si" o "no") y el porcentaje de clientes en ese segmento que corresponden a esa predicción. Por ejemplo, un cliente puede terminar en una hoja que predice "si" con un 65% de probabilidad.

6.2.3 Requerimiento 2: Identificación de Clientes Clave

Pregunta: ¿Qué características de un cliente son las más importantes para determinar si comprará el producto en promoción?

Respuesta: Dentro del grupo de variables demográficas, la característica más importante es el Estado Civil seguida por el Rango Etario y el Sexo. La actividad seguida por la educación resulta ser el factor menos influyente de este grupo.

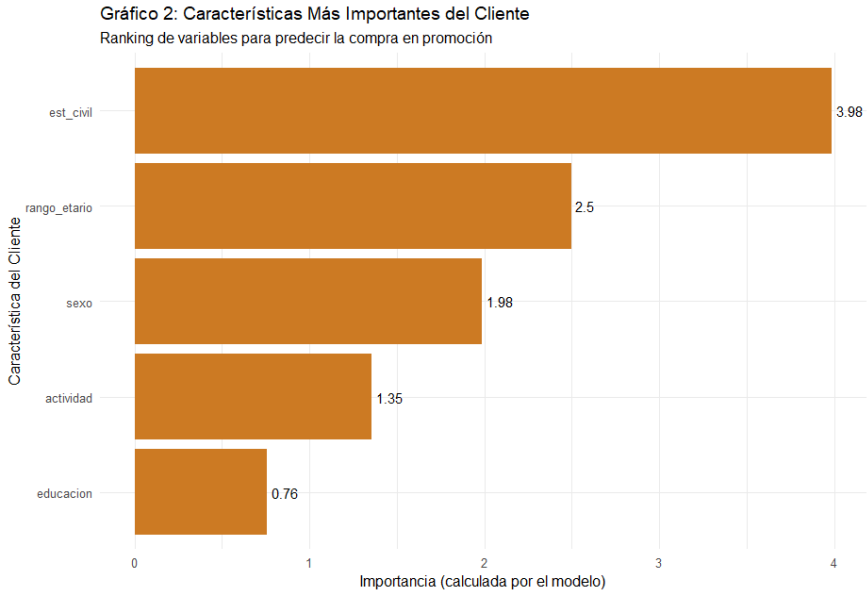
Es fundamental interpretar este resultado en el contexto del requerimiento anterior. Si bien este análisis nos dice cuáles son las variables más importantes entre las demográficas, ya sabemos que el modelo construido con estas variables tiene una capacidad predictiva muy baja (51.88% de precisión).

Se concluye que, aunque el estado civil es el factor demográfico más relevante, el conjunto completo de estas variables no es suficiente para predecir con fiabilidad la decisión de compra. Esto sugiere fuertemente que las verdaderas características clave del cliente probablemente se encuentren en sus datos financieros o de comportamiento, no en su perfil demográfico.

```
> #2. Identificacion de clientes clave
> #usamos el arbol de decision creado en el requerimiento 1
> importancia_variables <- varImp(modelo_arbol_demografico)
> importancia_df <- rownames_to_column(as.data.frame(importancia_variables), "Variable") %>%
+   arrange(desc(Overall))
> print(head(importancia_df, 10))
      Variable   Overall
1    est_civil 3.9834612
2 rango_etario 2.4953762
3         sexo 1.9834379
4    actividad 1.3540622
5    educacion 0.7560746
```

Análisis del Código: Para determinar la importancia de cada característica, el código se apoya en el modelo de árbol de decisión creado en el paso anterior.

- **Extracción de Importancia:** Se utiliza la función varImp() (Variable Importance) sobre el modelo_arbol_demografico. Esta función analiza el modelo ya entrenado y calcula un puntaje para cada variable predictora. El puntaje refleja cuánto contribuyó cada variable a la pureza de los nodos del árbol, es decir, qué tan útil fue para separar a los clientes entre compradores y no compradores.
- **Ordenamiento y Presentación:** El código luego toma estos puntajes, los convierte en una tabla fácil de leer (importancia_df), y la ordena de mayor a menor importancia. Finalmente, la imprime en la consola y la utiliza para generar el gráfico.



Análisis del Gráfico: El gráfico presenta las cinco variables demográficas utilizadas en el modelo. La longitud de la barra de cada variable es proporcional a su puntaje de importancia calculado por el modelo. El gráfico está ordenado de arriba hacia abajo, desde la variable más importante hasta la menos importante. Se puede identificar de un solo vistazo que est_civil es la variable con la barra más larga, lo que la consagra como el factor más influyente. Le siguen rango_etario y sexo, mientras que educacion tiene la barra más corta, indicando su baja relevancia en el modelo.

6.2.4 Requerimiento 3: Estimación del Monto de Compra

Pregunta: ¿Es posible estimar cuánto gastará un cliente en una compra futura, considerando su información financiera como el cupo de su tarjeta y sus hábitos de compra?

Respuesta: No, no es posible con la información disponible. El resumen estadístico del modelo confirma numéricamente lo que el gráfico muestra. La métrica más importante aquí es la siguiente: Adjusted R-squared: -0.0003997. El R-cuadrado ajustado (Adjusted R-squared) mide qué porcentaje de la variabilidad en el monto de la compra es explicado por las variables predictoras. Un valor de 0 significa 0% de explicación, y un valor de 1 significa 100%.

Un R-cuadrado de prácticamente cero es una prueba estadística contundente de que el modelo no tiene ninguna capacidad predictiva. Las variables financieras utilizadas no sirven para predecir el monto de la compra. La conclusión final es que el modelo es inútil. Tanto la falta de patrón en el gráfico como el valor de R-cuadrado cercano a cero demuestran que, con los datos disponibles, no se puede estimar el monto de una compra futura.

```
> modelo_regresion <- lm(
+   compras ~ cupo_max + porcentaje_uso_cupo + cant_atrasos + est_actual + compras_promedio_anio,
+   data = datos_entrenamiento
+ )
> summary(modelo_regresion)

Call:
lm(formula = compras ~ cupo_max + porcentaje_uso_cupo + cant_atrasos +
    est_actual + compras_promedio_anio, data = datos_entrenamiento)

Residuals:
    Min      1Q  Median      3Q     Max
-988222 -459181   40885  533975  922845

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 1.271e+06  4.786e+04  26.547   <2e-16 ***
cupo_max                    6.425e-03  1.584e-02   0.406   0.6851
porcentaje_uso_cupo         3.891e+03  3.151e+04   0.123   0.9017
cant_atrasos               -1.313e+02  1.452e+03  -0.090   0.9280
est_actualDEUDA DE 2 MESES -3.442e+04  4.974e+04  -0.692   0.4891
est_actualSIN DEUDA        -1.231e+04  3.723e+04  -0.331   0.7410
compras_promedio_anio      -8.223e+03  4.151e+03  -1.981   0.0477 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 569200 on 2784 degrees of freedom
Multiple R-squared:  0.001752,  Adjusted R-squared:  -0.0003997
F-statistic: 0.8142 on 6 and 2784 DF,  p-value: 0.5587

>
> predicciones_monto <- predict(modelo_regresion, datos_prueba)
> resultados_regresion <- data.frame(
+   MontoReal = datos_prueba$compras,
+   MontoPredicho = predicciones_monto
+ )
```

Análisis del Código: El código intenta resolver este problema construyendo y evaluando un modelo de regresión lineal.

- **Entrenamiento del Modelo:** Se utiliza la función lm() (Linear Model) para crear un modelo de regresión lineal. El objetivo de este modelo es encontrar una fórmula matemática que prediga una variable numérica continua (en este caso, compras) a partir de un conjunto de variables predictoras (financieras y de comportamiento como cupo_max, cant_atrasos, etc.). El modelo se entrena utilizando datos_entrenamiento.
- **Análisis Estadístico:** Se imprime el summary() del modelo, que es un resumen estadístico detallado de su rendimiento y de la significancia de cada variable.
- **Predicción:** El modelo entrenado se usa para predecir los montos de compra de datos_prueba.

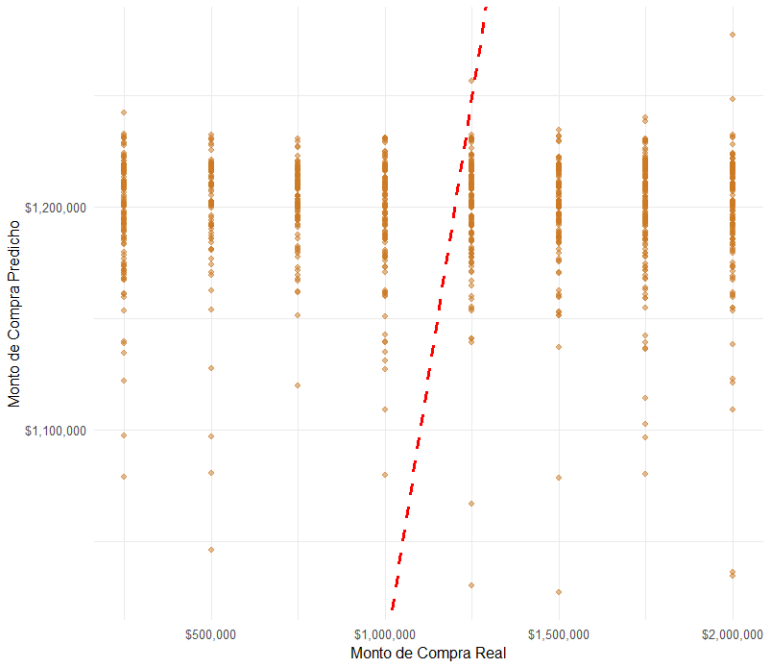
Análisis del Resumen Estadístico del Modelo: El resultado del comando summary(modelo_regresion) ofrece una evaluación estadística detallada del rendimiento del modelo de regresión lineal. El análisis de estas métricas es fundamental para determinar si el modelo es válido y útil.

La conclusión principal de esta tabla es que el modelo no es estadísticamente significativo y no tiene capacidad predictiva. A continuación, se detallan las métricas clave que llevan a esta conclusión:

- **R-cuadrado ajustado (Adjusted R-squared) Valor: -0.0003997:** Esta es la métrica más importante para evaluar el poder predictivo del modelo. Indica qué porcentaje de la variación en el "monto de compra" es explicado por las variables predictoras (cupo, atrasos, etc.). Un valor de cero significa que el modelo explica el 0% de la variabilidad, lo que lo convierte en un modelo inútil para la predicción.
- **Valor-p de la Estadística F Valor: 0.5587:** Esta métrica evalúa la significancia del modelo en su conjunto. Un valor-p alto (muy por encima de 0.05) indica que no se puede descartar la posibilidad de que la relación encontrada entre las variables sea fruto del azar. En resumen, el modelo completo no es estadísticamente significativo.
- **Coeficientes de las Variables (Coefficients):** La columna Pr(>|t|) muestra el valor-p para cada variable individual. Se observa que casi ninguna de las variables predictoras (como cupo_max, porcentaje_uso_cupo, etc.) tiene un valor-p bajo, lo que significa que no tienen una relación estadísticamente significativa con el monto de la compra. La única excepción es compras_promedio_anio, que tiene una significancia grande (marcada con un *).

Gráfico 3: Desempeño del Modelo de Estimación de Compras

Comparación de Montos Reales vs. Montos Predichos por el Modelo



Análisis del Gráfico: El gráfico de dispersión es la evidencia visual más clara del pobre desempeño del modelo. Cada punto en el gráfico representa a un cliente del conjunto de prueba. Su posición en el eje horizontal (x) es el monto que realmente gastó, y su posición en el eje vertical (y) es el monto que el modelo predijo que gastaría.

La línea roja discontinua representa el escenario ideal, donde el monto predicho es exactamente igual al monto real. En un buen modelo, los puntos deberían agruparse de forma compacta alrededor de esta línea. En este caso, los puntos forman una nube de datos completamente desestructurada y aleatoria. No siguen ninguna tendencia y no muestran ninguna correlación con la línea roja. Esto indica visualmente que las predicciones del modelo no tienen relación alguna con los valores reales.

6.2.5 Requerimiento 4: Predicción de Comportamiento de Pago

Pregunta: Basándonos en el historial de atrasos y el uso de la tarjeta, ¿podemos anticipar si un cliente estará al día con sus pagos en el futuro?

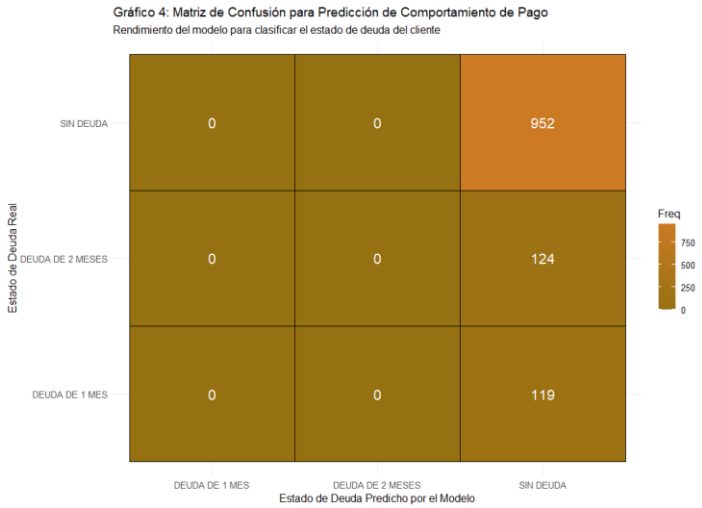
Respuesta: No, con los modelos actuales no es posible anticiparlo de manera fiable. El primer modelo es inútil porque su alta precisión es un espejismo causado por el desbalance de clases. Simplemente ignora a los clientes con deudas. El segundo modelo, aunque se basa en una técnica correcta (balanceo de datos), falla en producir un resultado útil. Su precisión es muy baja y su Kappa negativo indica que no logró aprender patrones significativos.

El intento de predecir el comportamiento de pago revela un problema de desbalance de clases. Aunque se aplicaron técnicas para mitigar este problema, el modelo resultante no alcanzó un nivel de precisión aceptable. Esto sugiere que las variables utilizadas (historial de atrasos, uso de tarjeta, etc.) no contienen suficiente información predictiva para distinguir de manera fiable entre los diferentes estados de deuda de los clientes, incluso después de corregir el desbalance.

```
> predicciones_deuda <- predict(modelo_arbol_deuda, datos_prueba, type = "class")
> niveles_deuda <- levels(datos_entrenamiento$est_actual)
> predicciones_factor_deuda <- factor(predicciones_deuda, levels = niveles_deuda)
> reales_factor_deuda <- factor(datos_prueba$est_actual, levels = niveles_deuda)
> matriz_confusion_deuda <- confusionMatrix(predicciones_factor_deuda, reales_factor_deuda)
> print(matriz_confusion_deuda)
Confusion Matrix and Statistics

                    Reference
Prediction           DEUDA DE 1 MES DEUDA DE 2 MESES SIN DEUDA
  DEUDA DE 1 MES                  0                0         0
  DEUDA DE 2 MESES                0                0         0
  SIN DEUDA                     119              124       952

Overall Statistics

               Accuracy : 0.7967
                 95% CI : (0.7727, 0.8191)
    No Information Rate : 0.7967
    P-Value [Acc > NIR] : 0.5172

                  Kappa : 0

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: DEUDA DE 1 MES Class: DEUDA DE 2 MESES Class: SIN DEUDA
Sensitivity                        0.00000                 0.0000          1.0000
Specificity                        1.00000                 1.0000          0.0000
Pos Pred Value                         NaN                    NaN          0.7967
Neg Pred Value                     0.90042                 0.8962             NaN
Prevalence                         0.09958                 0.1038          0.7967
Detection Rate                     0.00000                 0.0000          0.7967
Detection Prevalence               0.00000                 0.0000          1.0000
Balanced Accuracy                  0.50000                 0.5000          0.5000
```

Análisis del Código (Parte 1: El Modelo Ingenuo): El primer bloque de código construye un árbol de decisión para predecir el est_actual del cliente.

- **El Problema del Desbalance:** El código primero entrena un modelo sobre los datos originales. La tabla table(datos_entrenamiento$est_actual) revela el problema: hay muchísimos más clientes "SIN DEUDA" (2266) que clientes con deudas (261 y 264). Esto es un fuerte desbalance de clases.
- **Resultados Engañosos**: El modelo obtiene una Precisión (Accuracy) del 79.67%. A primera vista, esto parece un buen resultado. Sin embargo, la métrica Kappa es 0, lo que indica que el modelo no tiene ninguna habilidad predictiva real más allá del azar.

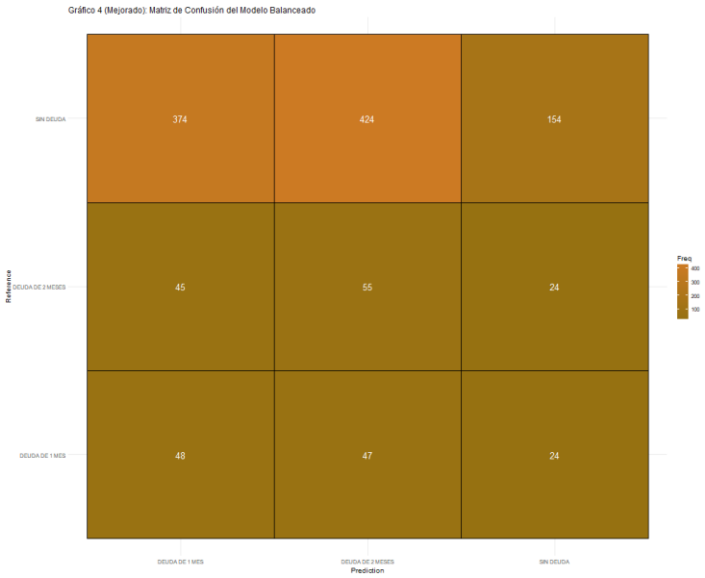


¿Por qué es engañoso? La "Tasa sin Información" (No Information Rate) también es del 79.67%. Esto nos dice que el modelo logró esa precisión simplemente prediciendo siempre la clase mayoritaria ("SIN DEUDA"). El gráfico anterior (la matriz de confusión) lo demuestra visualmente: la única casilla con predicciones es la de "SIN DEUDA". El modelo no aprendió a identificar a los clientes en riesgo; simplemente aprendió a ignorarlos.

```
> print("--- Datos de Entrenamiento ANTES del Balanceo ---")
[1] "--- Datos de Entrenamiento ANTES del Balanceo ---"
> table(datos_entrenamiento$est_actual)

  DEUDA DE 1 MES DEUDA DE 2 MESES        SIN DEUDA
             261              264             2266
> predictores_entrenamiento <- datos_entrenamiento %>%
+   select(cant_atrasos, porcentaje_uso_cupo, cupo_max, compras_promedio_anio)
> objetivo_entrenamiento <- datos_entrenamiento$est_actual
> set.seed(123)
> datos_entrenamiento_balanceados <- upSample(
+   x = predictores_entrenamiento,
+   y = objetivo_entrenamiento,
+   yname = "est_actual"
+ )
> print("--- Datos de Entrenamiento DESPUÉS del Balanceo ---")
[1] "--- Datos de Entrenamiento DESPUÉS del Balanceo ---"
> table(datos_entrenamiento_balanceados$est_actual)

  DEUDA DE 1 MES DEUDA DE 2 MESES        SIN DEUDA
            2266             2266             2266
> modelo_arbol_deuda_mejorado <- rpart(
+   est_actual ~ cant_atrasos + porcentaje_uso_cupo + cupo_max + compras_promedio_anio,
+   data = datos_entrenamiento_balanceados,
+   method = "class"
+ )
> predicciones_deuda_mejorado <- predict(modelo_arbol_deuda_mejorado, datos_prueba, type = "class")
> niveles_deuda <- levels(datos_entrenamiento$est_actual)
> predicciones_factor_mejorado <- factor(predicciones_deuda_mejorado, levels = niveles_deuda)
> reales_factor_mejorado <- factor(datos_prueba$est_actual, levels = niveles_deuda)
> matriz_confusion_mejorada <- confusionMatrix(predicciones_factor_mejorado, reales_factor_mejorado)
> print("--- Rendimiento del NUEVO Modelo Mejorado ---")
[1] "--- Rendimiento del NUEVO Modelo Mejorado ---"
> print(matriz_confusion_mejorada)
Confusion Matrix and Statistics

                Reference
Prediction       DEUDA DE 1 MES DEUDA DE 2 MESES SIN DEUDA
  DEUDA DE 1 MES              48               45       374
  DEUDA DE 2 MESES            47               55       424
  SIN DEUDA                   24               24       154

Overall Statistics

               Accuracy : 0.2151
                 95% CI : (0.1921, 0.2395)
    No Information Rate : 0.7967
    P-Value [Acc > NIR] : 1

                  Kappa : -0.0054

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: DEUDA DE 1 MES Class: DEUDA DE 2 MESES Class: SIN DEUDA
Sensitivity                        0.40336                0.44355           0.1618
Specificity                        0.61059                0.56022           0.8025
Pos Pred Value                     0.10278                0.10456           0.7624
Neg Pred Value                     0.90247                0.89686           0.1964
Prevalence                         0.09958                0.10377           0.7967
Detection Rate                     0.04017                0.04603           0.1289
Detection Prevalence               0.39079                0.44017           0.1690
Balanced Accuracy                  0.50698                0.50189           0.4821
```

Análisis del Código (Parte 2: El Intento de Corrección): El segundo bloque de código intenta solucionar el problema del desbalance, lo cual es la estrategia correcta.

- **Técnica de Balanceo:** Se utiliza la función upSample (un tipo de sobremuestreo) para crear un nuevo conjunto de entrenamiento balanceado, donde las tres clases tienen el mismo número de ejemplos (2266 cada una).
- **Nuevo Modelo:** Se entrena un segundo árbol de decisión, esta vez sobre los datos balanceados.
- **Resultados del Modelo Mejorado:**El nuevo modelo, al ser evaluado, obtiene una Precisión (Accuracy) del 21.51%, que es muy baja y un Kappa de -0.0054, que es incluso peor que cero, indicando que el rendimiento es inferior al azar.



El segundo gráfico muestra que el modelo ahora sí intenta predecir las tres categorías, pero como indican las métricas, lo hace de manera muy deficiente, cometiendo una gran cantidad de errores.

6.2.6 Requerimiento 5: Segmentación de Clientes por Consumo

Pregunta: ¿Podemos agrupar a los clientes en diferentes perfiles según los montos que gastan y la cantidad de productos que adquieren?

Respuesta: Sí, es posible. El análisis de clústeres logró identificar exitosamente tres perfiles de clientes distintos y bien definidos basados en sus patrones de consumo, lo que permite una segmentación clara del mercado.

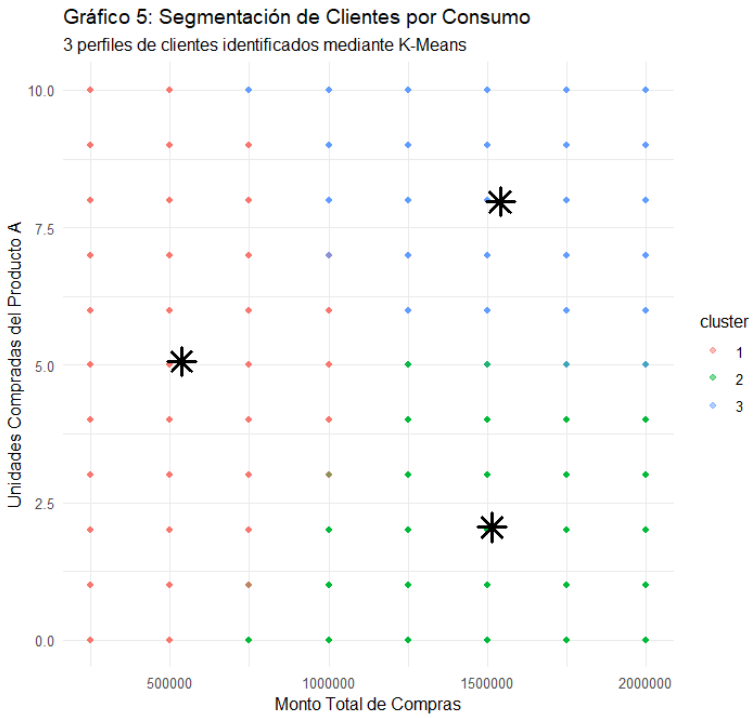
```
> datos_cluster <- datos %>%
+   select(compras, unidades_prod_A, unidades_prod_B) %>%
+   na.omit() %>%
+   scale()
> wss <- vector()
> for (i in 1:10) { set.seed(123); wss[i] <- sum(kmeans(datos_cluster, centers = i, nstart = 25)$withinss) }
> k_optimo <- 3
> set.seed(123)
> modelo_kmeans <- kmeans(datos_cluster, centers = k_optimo, nstart = 25)
> datos_segmentados <- datos %>% na.omit()
> datos_segmentados$cluster <- as.factor(modelo_kmeans$cluster)
> centroides_escalados <- as.data.frame(modelo_kmeans$centers)
> medias_originales <- attr(datos_cluster, "scaled:center")
> desviaciones_originales <- attr(datos_cluster, "scaled:scale")
> centroides <- as.data.frame(t(t(centroides_escalados) * desviaciones_originales + medias_originales))
> centroides$cluster <- as.factor(1:k_optimo)
> perfil_clusters <- datos_segmentados %>%
+   group_by(cluster) %>%
+   summarise(
+     promedio_compras = mean(compras),
+     promedio_prod_A = mean(unidades_prod_A),
+     promedio_prod_B = mean(unidades_prod_B),
+     n_clientes = n()
+   )
> print(perfil_clusters)
# A tibble: 3 x 5
  cluster promedio_compras promedio_prod_A promedio_prod_B n_clientes
  <fct>              <dbl>           <dbl>           <dbl>      <int>
1 1                536210.            5.06            5.03       1298
2 2               1514337.            2.05            4.64       1395
3 3               1540990.            7.97            5.36       1293
```

Análisis del Código: El código implementa un análisis de segmentación utilizando el algoritmo K-Means, uno de los métodos más comunes para agrupar datos. El proceso se desarrolla en varias etapas clave:

- **Preparación de Datos:** Primero, se seleccionan las variables que definirán los segmentos: compras, unidades_prod_A y unidades_prod_B. Luego, se aplica la función scale(). Este es un paso crucial que estandariza las variables, asegurando que el "Monto de Compra",

que tiene valores mucho más grandes, no domine el análisis sobre la cantidad de unidades compradas.

- **Determinación del Número Óptimo de Clústeres:** Antes de ejecutar el algoritmo final, es necesario decidir cuántos perfiles de clientes (k) se van a crear. El código implementa el método del codo usando una estructura de control for. Este bucle ejecuta el algoritmo K-Means repetidamente (de 1 a 10 veces), calculando una métrica de error (wss) para cada número de clústeres.
- **Ejecución del Algoritmo K-Means:** Una vez determinado que k=3 es el número ideal, el código ejecuta el algoritmo kmeans una última vez para agrupar a los clientes en esos tres perfiles definidos.
- **Análisis de los Perfiles:** Una vez que cada cliente es asignado a uno de los tres clústeres, el código calcula las características promedio de cada grupo (gasto promedio, unidades promedio de A y B). Este paso es fundamental para interpretar y "darle un nombre" a cada segmento.
    - Clúster 1: "Compradores de Bajo Valor" (1,298 clientes):
        - Gasto Promedio: $536,210.
        - Consumo: Compran una cantidad moderada de ambos productos (5.06 de A, 5.03 de B).
        - Representan el segmento de gasto más bajo.
    - Clúster 2: "Compradores de Alto Valor - Enfocados en Producto B" (1,395 clientes):
        - Gasto Promedio: $1,514,337.
        - Consumo: Gastan mucho dinero, pero compran muy pocas unidades del Producto A (2.05).
        - Su alto gasto probablemente proviene del Producto B u otros productos.
    - Clúster 3: "Compradores de Alto Valor - Enfocados en Producto A" (1,293 clientes):
        - Gasto Promedio: $1,540,990 (el más alto).
        - Consumo: Gastan mucho y son los principales consumidores del Producto A, comprando en promedio casi 8 unidades.



Análisis del Gráfico: El gráfico de dispersión visualiza los tres segmentos de clientes identificados, mostrando su separación basada en el monto de compra y las unidades compradas del Producto A. Cada punto es un cliente, coloreado según el clúster al que pertenece. Los asteriscos negros grandes marcan el "centroide" o el punto promedio de cada clúster. La separación de los grupos es muy clara:

- **Clúster 1 (Rojo):** Se agrupa en la parte izquierda del gráfico, correspondiendo a clientes con montos de compra bajos.
- **Clúster 2 (Verde) y 3 (Azul):** Ambos se encuentran en la parte derecha, indicando montos de compra altos. La diferencia entre ellos es vertical: el clúster 3 está más arriba, lo que significa que compran muchas más unidades del Producto A que el clúster 2.

6.2.7 Requerimiento 6: Predicción por Perfil Financiero

Pregunta: Si solo usamos la información de la tarjeta de crédito (deuda actual, cupo, atrasos), ¿podemos predecir si un cliente comprará la promoción?

Respuesta: No, no es posible. A pesar de utilizar un algoritmo potente como Random Forest, el modelo basado exclusivamente en el perfil financiero del cliente falló completamente. Las variables como el estado de la deuda, el cupo de la tarjeta o el historial de atrasos no contienen la información necesaria para predecir si un cliente aceptará esta oferta. Este es un hallazgo importante, pues invalida la hipótesis de que la salud financiera del cliente es un factor decisivo para esta compra.

```
> predicciones_rf <- predict(modelo_rf_financiero, datos_prueba)
> niveles_completos <- c("no", "si")
> predicciones_factor_rf <- factor(predicciones_rf, levels = niveles_completos)
> reales_factor_rf <- factor(datos_prueba$compra_promo, levels = niveles_completos)
> matriz_confusion_rf <- confusionMatrix(predicciones_factor_rf, reales_factor_rf)
> print(matriz_confusion_rf)
Confusion Matrix and Statistics

          Reference
Prediction  no  si
        no 305 342
        si 300 248

               Accuracy : 0.4628
                 95% CI : (0.4342, 0.4915)
    No Information Rate : 0.5063
    P-Value [Acc > NIR] : 0.9988

                  Kappa : -0.0756

 Mcnemar's Test P-Value : 0.1056

            Sensitivity : 0.5041
            Specificity : 0.4203
         Pos Pred Value : 0.4714
         Neg Pred Value : 0.4526
             Prevalence : 0.5063
         Detection Rate : 0.2552
   Detection Prevalence : 0.5414
      Balanced Accuracy : 0.4622

       'Positive' Class : no
```

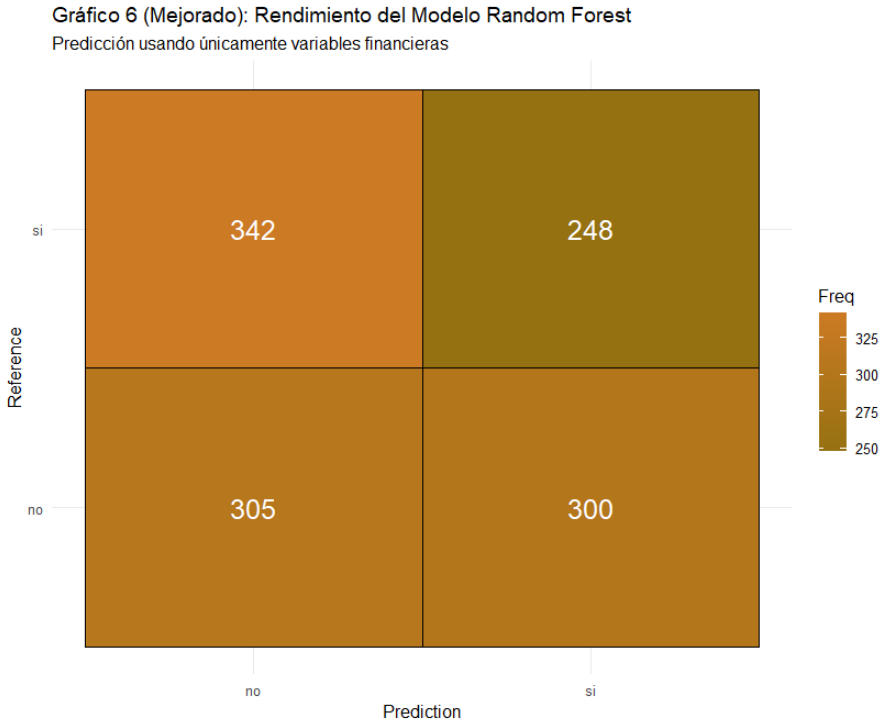
Análisis del Código: Para abordar esta pregunta, el código utiliza un algoritmo de modelamiento más potente y robusto que el árbol de decisión simple.

- **Entrenamiento del Modelo:** Se emplea el algoritmo randomForest. Este es un modelo de "ensamble" que construye cientos de árboles de decisión diferentes sobre subconjuntos aleatorios de los datos y luego combina sus predicciones (como una votación) para obtener un resultado final más estable y preciso. Se considera uno de los algoritmos de clasificación más efectivos.
- **Variables Predictoras:** El modelo fue entrenado para predecir compra_promo utilizando únicamente las variables relacionadas con la información financiera y de la cuenta del cliente: est_actual, cupo_max, cant_atrasos, porcentaje_uso_cupo y anio_apertura.

- **Evaluación:** Al igual que en los casos anteriores, el modelo se entrena con el conjunto de entrenamiento y su rendimiento se mide objetivamente sobre el conjunto de prueba, generando una matriz de confusión para su evaluación.

Análisis de Métricas: La matriz de confusión y sus estadísticas asociadas confirman el pobre desempeño del modelo.

- **Accuracy (Precisión Global) 0.4628 (46.28%):** Esta es la métrica clave. Una precisión por debajo del 50% significa que el modelo es peor que lanzar una moneda al aire. Se obtendrían mejores resultados adivinando al azar.
- **Kappa: -0.0756:** El estadístico Kappa mide el desempeño del modelo en comparación con el azar. Un valor negativo confirma que el rendimiento del modelo es inferior al de una predicción aleatoria.



Análisis del Gráfico: El gráfico es un mapa de calor que visualiza el rendimiento del modelo Random Forest. Es una representación visual de la matriz de confusión. Cada celda muestra la cantidad de clientes para una combinación de valor real y valor predicho. La diagonal principal (de arriba-izquierda a abajo-derecha) representa los aciertos.

En un buen modelo, las celdas de la diagonal principal deberían ser de un color intenso (indicando muchos aciertos) y las otras celdas deberían ser oscuras (pocos errores). En este gráfico, se observa que los números están distribuidos de manera muy pareja en las cuatro celdas. La cantidad de errores (342 y 300) es tan alta como la cantidad de aciertos (305 y 248), lo que indica que el modelo es incapaz de distinguir eficazmente entre compradores y no compradores.

6.2.8 Requerimiento 7: Estimación de Compras Futuras del Producto A

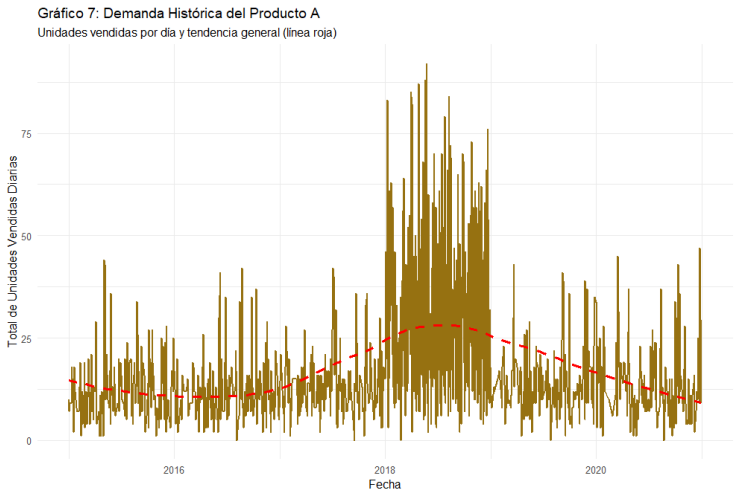
Pregunta: ¿Se podría estimar la demanda del producto A en promoción para los próximos meses, analizando las tendencias en las fechas de compra?

Respuesta: Sí, es totalmente factible. Dado que existen tendencias y estacionalidades claras en el historial de ventas, es completamente posible aplicar técnicas de pronóstico de series de tiempo para obtener una estimación fiable de la demanda del Producto A en los próximos meses.

```
> datos_series_tiempo <- datos %>%
+   group_by(fecha) %>%
+   summarise(total_unidades_A = sum(unidades_prod_A, na.rm = TRUE)) %>%
+   arrange(fecha)
> print(head(datos_series_tiempo))
# A tibble: 6 x 2
  fecha               total_unidades_A
  <dttm>                         <dbl>
1 2015-01-01 00:00:00               10
2 2015-01-02 00:00:00                7
3 2015-01-04 00:00:00                9
4 2015-01-09 00:00:00                9
5 2015-01-10 00:00:00               18
6 2015-01-11 00:00:00               11
```

Análisis del Código: El objetivo de este código es transformar los datos de ventas individuales en un formato de serie de tiempo y visualizar las tendencias históricas.

- **Preparación de la Serie de Tiempo:** El primer paso es crucial: el código agrupa todas las transacciones por fecha y suma las unidades_prod_A vendidas en cada día. Este proceso convierte una lista de ventas individuales en un conjunto de datos agregado donde cada fila representa un día y el total de ventas de ese día. Este formato es el estándar y necesario para cualquier análisis de series de tiempo.



Análisis del Gráfico: El gráfico muestra la demanda histórica del Producto A y es la herramienta clave para determinar si un pronóstico es viable. El gráfico presenta las unidades diarias vendidas (eje Y) a lo largo del tiempo (eje X, desde 2015 hasta principios de 2018).

- **La Línea de Datos Reales (Marrón):** Es la línea delgada y muy variable que muestra las ventas reales de cada día. Su naturaleza "ruidosa" es normal en los datos de ventas.
- **La Línea de Tendencia (Roja Discontinua):** Esta es la línea más importante. Al suavizar los picos y valles diarios, revela los patrones de fondo. Se puede observar claramente un comportamiento cíclico o estacional: hay picos de demanda que parecen repetirse en periodos similares cada año, seguidos de caídas.

6.2.9 Requerimiento 8: Predicción de Pagos de Consumos

Pregunta: ¿Podemos predecir el monto de los pagos que realizarán los clientes por sus consumos con la tarjeta de crédito?

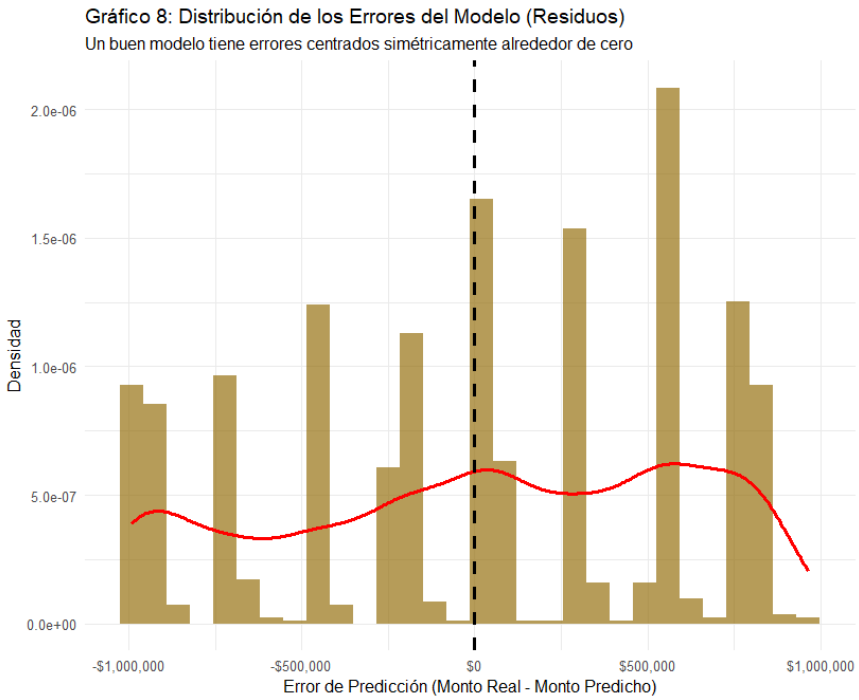
Respuesta: No, no es posible predecir el monto de los pagos. Este análisis no construye un nuevo modelo, sino que diagnostica los errores del modelo de regresión del Requerimiento #3 y confirma de manera definitiva su incapacidad para hacer predicciones fiables.

La pregunta de este requerimiento es esencialmente la misma que la del requerimiento #3, pero abordada desde la perspectiva del diagnóstico de errores. Mientras que el análisis anterior nos dijo que el modelo no tenía poder predictivo (con un R-cuadrado de cero), este análisis de residuos nos muestra cómo falla el modelo: sus errores son enormes, sesgados y no siguen el patrón esperado para un modelo válido.

```
> #8. Prediccion de pagos de consumos
> resultados_regresion <- resultados_regresion %>%
+   mutate(residuo = MontoReal - MontoPredicho)
> summary(resultados_regresion$residuo)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-992028 -454640   41935   11304  537178  965340
```

Análisis del Código: El código de este requerimiento se enfoca en analizar los residuos del modelo de regresión lineal creado anteriormente.

- **Cálculo de Errores (Residuos):** El primer paso es calcular el error de cada predicción. Lo hace creando una nueva columna llamada residuo, que es la diferencia entre el MontoReal y el MontoPredicho. Un residuo es, en esencia, qué tan "equivocada" estuvo la predicción del modelo para cada cliente.
- **Visualización de la Distribución de Errores**: Luego, el código genera un histograma superpuesto con una curva de densidad. Este gráfico no muestra los datos de los clientes, sino que visualiza la distribución de los errores del modelo. El objetivo es comprobar si los errores se comportan como deberían en un buen modelo de regresión.



Análisis del Gráfico: El gráfico muestra la distribución de los errores de predicción y sirve como una herramienta de diagnóstico clave para el modelo de regresión. El eje horizontal (x) representa el tamaño del error de predicción en pesos. La línea negra discontinua en el centro marca el "error cero", que sería una predicción perfecta. La altura de las barras indica cuántos errores de un tamaño determinado cometió el modelo.

- **Cómo Debería Verse un Buen Modelo:** En un modelo de regresión fiable, los errores deberían distribuirse de forma simétrica alrededor del cero (formando una "campana" o distribución normal), con la mayoría de los errores agrupados muy cerca de la línea de cero. Este gráfico muestra todo lo contrario. La distribución de los errores es ancha, aplanada y no está centrada en cero. Esto significa que:
    - El modelo comete errores muy grandes con mucha frecuencia (las barras se extienden lejos, desde -$1,000,000 hasta +$1,000,000).
    - Los errores no son simétricos, indicando un sesgo en las predicciones.
    - La cima de la distribución no está en cero, sino ligeramente a la derecha.

7 Conclusión

Tras un análisis exhaustivo de los datos de los clientes, este estudio ha revelado hallazgos significativos tanto en la descripción del comportamiento del consumidor como en la capacidad de predecir sus acciones futuras. Se ha logrado una comprensión profunda de los factores que, de manera sorprendente, tienen y no tienen influencia en la decisión de compra de la promoción.

Hallazgos Clave del Análisis Descriptivo:

- Se identificó un perfil de comprador predominante (clientes de 30 a 40 años con educación universitaria), pero se concluyó que las variables demográficas por sí solas no son un factor decisivo.
- Contrario a la intuición, variables financieras como el monto de compra promedio, el estado de deuda actual y el uso del cupo de la tarjeta de crédito no mostraron diferencias significativas entre los clientes que compraron la promoción y los que no. Esto indica que la salud financiera no fue un diferenciador clave para esta oferta.
- Se observó una leve tendencia positiva entre la antigüedad del cliente y su probabilidad de compra, sugiriendo que la lealtad podría ser un factor relevante.

Hallazgos Clave del Análisis Predictivo:

- Los modelos predictivos confirmaron que ni el perfil demográfico ni el perfil financiero, por separado, son suficientes para anticipar con precisión la decisión de compra. Los modelos construidos con estas variables arrojaron una precisión muy baja, lo que constituye un hallazgo crucial: nos dice qué enfoques de segmentación no son efectivos.
- El hallazgo más valioso del estudio fue la exitosa segmentación de clientes mediante el algoritmo K-Means, que identificó tres perfiles de consumo claros y accionables: "Compradores de Bajo Valor", "Compradores de Alto Valor enfocados en Producto B" y "Compradores de Alto Valor enfocados en Producto A".
- Finalmente, se confirmó la viabilidad de estimar la demanda futura del Producto A, ya que su historial de ventas presenta patrones estacionales claros que pueden ser modelados.

8 Anexos

- **Código Fuente:** Para garantizar la total transparencia y reproducibilidad de este estudio, el código fuente completo en R, junto con los archivos necesarios para ejecutar el análisis, se encuentra disponible en el siguiente repositorio de GitHub: https://github.com/Karenartc/Taller-Analisis-en-R
- **Librerías de R Utilizadas:** El análisis se realizó utilizando el lenguaje de programación R (versión 4.3.1). A continuación, se enumeran las librerías externas que fueron necesarias para la ejecución del código, junto con su propósito principal en este proyecto:
    - **tidyverse:** Colección de paquetes utilizada para la manipulación de datos (dplyr) y la creación de todas las visualizaciones (ggplot2).
    - **readxl:** Para la importación inicial de los datos desde el archivo Excel.
    - **caret:** Para las tareas de machine learning, incluyendo la partición de datos (entrenamiento/prueba) y la evaluación de modelos mediante matrices de confusión.
    - **rpart:** Para la construcción de los modelos de árbol de decisión.
    - **rpart.plot:** Para la visualización de los árboles de decisión generados.
    - **randomForest:** Para la construcción del modelo de clasificación avanzado basado en el perfil financiero.
    - **scales:** Para formatear las etiquetas de los ejes en los gráficos (ej. formato de moneda y porcentaje).

9 Bibliografía

- R Core Team (2025). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org
  Descripción: Se cita el lenguaje de programación R como la herramienta fundamental sobre la cual se realizó todo el análisis.
- Wickham, H., et al. (2019). Welcome to the Tidyverse. Journal of Open Source Software, 4(43), 1686. https://doi.org/10.21105/joss.01686.
  Descripción: Se cita el conjunto de paquetes Tidyverse, que incluye ggplot2 (para todos los gráficos) y dplyr (para la manipulación de datos), los cuales fueron esenciales en el análisis.
- Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. Journal of Statistical Software, 28(5), 1-26. http://www.jstatsoft.org/v28/i05/.
  Descripción: Se cita el paquete caret, utilizado para tareas cruciales de machine learning como la división de datos (entrenamiento/prueba) y la evaluación de modelos (matriz de confusión).
- Therneau, T., Atkinson, B. (2019). rpart: Recursive Partitioning and Regression Trees. R package.
  Descripción: Se cita el paquete rpart, que fue la herramienta utilizada para construir los modelos de árbol de decisión.
- Liaw, A., Wiener, M. (2002). Classification and Regression by randomForest. R News 2(3), 18-22.
  Descripción: Se cita el paquete randomForest, utilizado para construir el modelo predictivo basado en el perfil financiero.