Trabajo 2 ML1

Jose Castro

2024-08-27

Contents

Parte 1: Aprendizaje supervisado: Regresión

```
# importacion de librerias
library(ggplot2)
library(dplyr)
library(readr)
library(VIM)
library(mice)
library(rsample)
library(caret)
library(glmnet)
```

Item 1

1. Importe la base de datos, transfórmela en un data frame y elimine la variable "Emp.ID".

```
# Importo la base de datos y la cargo como dataframe
datos <- as.data.frame(read.csv("Employee Attrition.csv"))
```

```
# Elimino la variable "Emp.ID"
datos <- datos[, -which(names(datos) == "Emp.ID")]
```

```
#verifico si se eliminó la variable
head(datos)
```

```
##   satisfaction_level last_evaluation number_project average_montly_hours
## 1               0.38            0.53              2                   157
## 2               0.80            0.86              5                   262
## 3               0.11            0.88              7                   272
## 4               0.72            0.87              5                   223
## 5               0.37            0.52              2                   159
## 6               0.41            0.50              2                   153
##   time_spend_company Work_accident promotion_last_5years  dept salary
## 1                  3             0                     0 sales    low
## 2                  6             0                     0 sales medium
## 3                  4             0                     0 sales medium
## 4                  5             0                     0 sales    low
## 5                  3             0                     0 sales    low
## 6                  3             0                     0 sales    low
```

Item 2

2. Describa cada una de las variables e indique si corresponden a variables numéricas o categóricas. Si considera que hay un número excesivo de variables (o variables irrelevantes) describir solamente las de mayor interés.

Variables Numéricas:

- Emp_Id: Identificador único del empleado.

- satisfaction_level: Nivel de satisfacción del empleado, expresado como una proporción (0 a 1).
- last_evaluation: Evaluación de desempeño (0 a 1).
- number_project: Número de proyectos en los que el empleado está trabajando actualmente.
- average_montly_hours: Promedio de horas mensuales trabajadas por el empleado.
- time_spend_company: Tiempo de permanencia en la empresa por parte del empleado.

Variables Categóricas:

- Work_accident: Indicador de si el empleado ha estado involucrado en un accidente laboral (sí/no).
- promotion_last_5years: Indicador de si el empleado ha recibido una promoción en los últimos 5 años (sí/no).
- dept: Departamento en el que el empleado trabaja.
- salary: Categoría salarial del empleado (baja, media, alta).

Item 3

3. Realice estadística descriptiva para cada una de las variables (énfasis principal en la variable dependiente). Incorpore análisis gráfico

Considerando el contexto del conjunto de datos, que se enfoca en la satisfacción de los empleados y la retención de talentos en la empresa, creo que la variable que más sentido tiene como variable dependiente es **satisfaction_level**.

La satisfacción del empleado es un resultado o un efecto que puede ser influenciado por las otras variables, como la evaluación de desempeño, el número de proyectos, las horas trabajadas, la permanencia en la empresa, los accidentes laborales, las promociones, el departamento y la categoría salarial.

En otras palabras, la satisfacción del empleado es una variable que puede ser explicada o predicha por las otras variables, lo que la convierte en una buena candidata para ser la variable dependiente en un modelo de regresión o análisis de correlación.

```
# selecciono las columnas categoricas
# luego veo la cantidad de valores por cada categoria o clase
categorical_columns <- c("Work_accident",
                         "promotion_last_5years",
                         "dept", "salary")
valores_por_categoria <- lapply(datos[categorical_columns], table)
valores_por_categoria
```

```
## $Work_accident
##
##     0     1
## 12830  2169
##
## $promotion_last_5years
##
##     0     1
## 14680   319
##
## $dept
##
##  accounting          hr          IT  management   marketing product_mng
##         767         739        1227         630         858         902
```

```
##       RandD       sales     support   technical
##         787        4140        2229        2720
##
## $salary
##
##   high    low medium
##   1237   7316   6446
```

```r
# ahora veo los porcentajes de valores por cada categoria
# en cada columna categorica
col_interes_cat <- c("Work_accident", "promotion_last_5years", "dept", "salary")
porcentaje_por_clase <- lapply(datos[, col_interes_cat],
                               function(x) prop.table(table(x)) * 100)
porcentaje_por_clase
```

```
## $Work_accident
## x
##        0        1
## 85.53904 14.46096
##
## $promotion_last_5years
## x
##         0         1
## 97.873192  2.126808
##
## $dept
## x
##  accounting          hr          IT  management   marketing product_mng
##    5.113674    4.926995    8.180545    4.200280    5.720381    6.013734
##       RandD       sales     support   technical
##    5.247016   27.601840   14.860991   18.134542
##
## $salary
## x
##      high       low    medium
##  8.247216 48.776585 42.976198
```

Analisis Univariado Variables Numericas

La siguiente celda se utiliza para generar un resumen estadístico de las variables numéricas seleccionadas en el conjunto de datos Employee Attrition.

El resumen incluye medidas como la media, mediana, mínimo, máximo y los cuartiles (Q1 y Q3), que son útiles para entender la distribución de los datos.

```r
# veo un resumen de las variables numericas
# como la media, mediana (Q2), minimo, maximo, Q1, Q3
summary(datos[c("satisfaction_level",
                "last_evaluation", "number_project",
                "average_montly_hours", "time_spend_company")])
```

```
##  satisfaction_level last_evaluation  number_project  average_montly_hours
##  Min.   :0.0900     Min.   :0.3600   Min.   :2.000   Min.   : 96.0
##  1st Qu.:0.4400     1st Qu.:0.5600   1st Qu.:3.000   1st Qu.:156.0
```

```
##  Median :0.6400     Median :0.7200   Median :4.000   Median :200.0
##  Mean   :0.6128     Mean   :0.7161   Mean   :3.803   Mean   :201.1
##  3rd Qu.:0.8200     3rd Qu.:0.8700   3rd Qu.:5.000   3rd Qu.:245.0
##  Max.   :1.0000     Max.   :1.0000   Max.   :7.000   Max.   :310.0
##  time_spend_company
##  Min.   : 2.000
##  1st Qu.: 3.000
##  Median : 3.000
##  Mean   : 3.498
##  3rd Qu.: 4.000
##  Max.   :10.000
```

Medidas resumen incluidas:

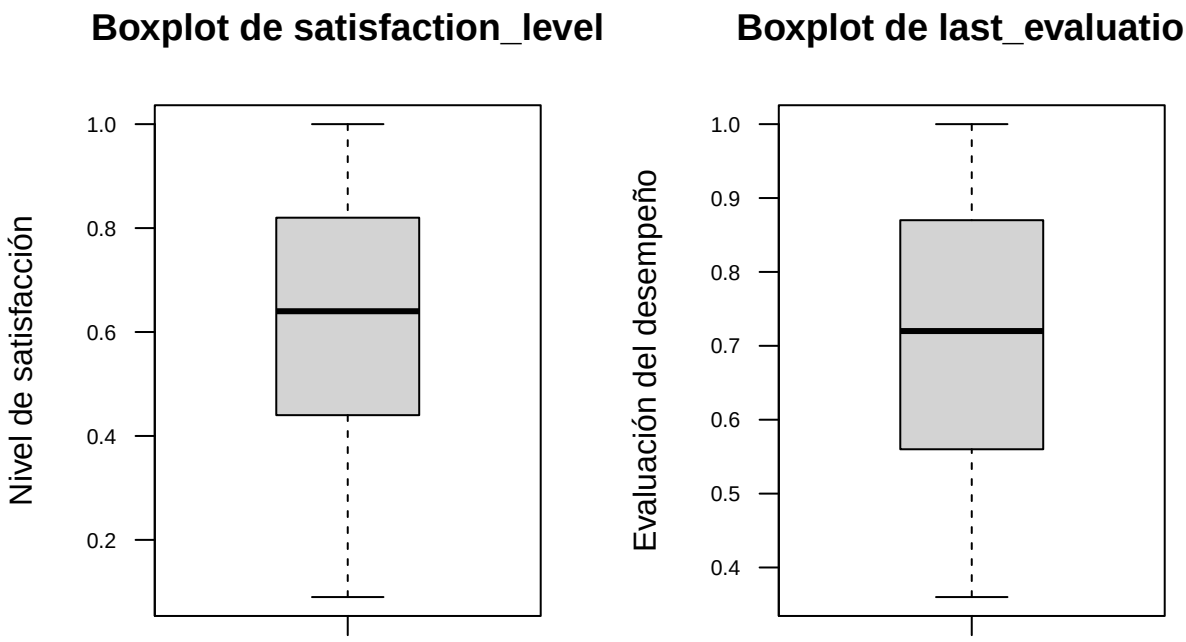
- Media: El promedio de los valores en cada columna.
- Mediana (Q2): El valor central cuando los datos están ordenados.
- Mínimo: El valor más pequeño en la columna.
- Máximo: El valor más grande en la columna.
- Primer cuartil (Q1): El valor por debajo del cual se encuentra el 25% de los datos.
- Tercer cuartil (Q3): El valor por debajo del cual se encuentra el 75% de los datos.

```r
# selecciono las columnas numericas que me interesan
# luego calculo el rango intercuartilico de cada una
col_interes_num <- c("satisfaction_level", "last_evaluation", "number_project",
                     "average_montly_hours", "time_spend_company")
valores_iqr <- sapply(datos[col_interes_num], IQR)
valores_iqr
```

```
##   satisfaction_level      last_evaluation       number_project
##                 0.38                 0.31                 2.00
## average_montly_hours   time_spend_company
##                89.00                 1.00
```

```r
# Ajusto el tamaño de la ventana gráfica
options(repr.plot.width = 10, repr.plot.height = 5)

# Divido el área de gráficos en una disposición de 1 fila por 2 columnas
par(mfrow = c(1, 2))

# Creo el primer boxplot para satisfaction_level
boxplot(datos$satisfaction_level,
        main = "Boxplot de satisfaction_level",
        ylab = "Nivel de satisfacción",
        xlab = "",
        outline = TRUE,
        axes = FALSE)  # Oculto ejes para personalizarlos después

# Añado de nuevo los ejes
axis(2, las = 1, cex.axis = 0.7,
     at = pretty(datos$satisfaction_level),
     labels = format(pretty(datos$satisfaction_level), scientific = FALSE))
axis(1, at = 1, labels = "")
box()  # Añado el cuadro alrededor del gráfico
```

```r
# Creo el segundo boxplot para last_evaluation
boxplot(datos$last_evaluation,
        main = "Boxplot de last_evaluation",
        ylab = "Evaluación del desempeño",
        xlab = "",
        outline = TRUE,
        axes = FALSE)  # Oculto ejes para personalizarlos después

# Añado de nuevo los ejes
axis(2, las = 1, cex.axis = 0.7,
     at = pretty(datos$last_evaluation),
     labels = format(pretty(datos$last_evaluation), scientific = FALSE))
axis(1, at = 1, labels = "")
box()  # Añado el cuadro alrededor del gráfico
```



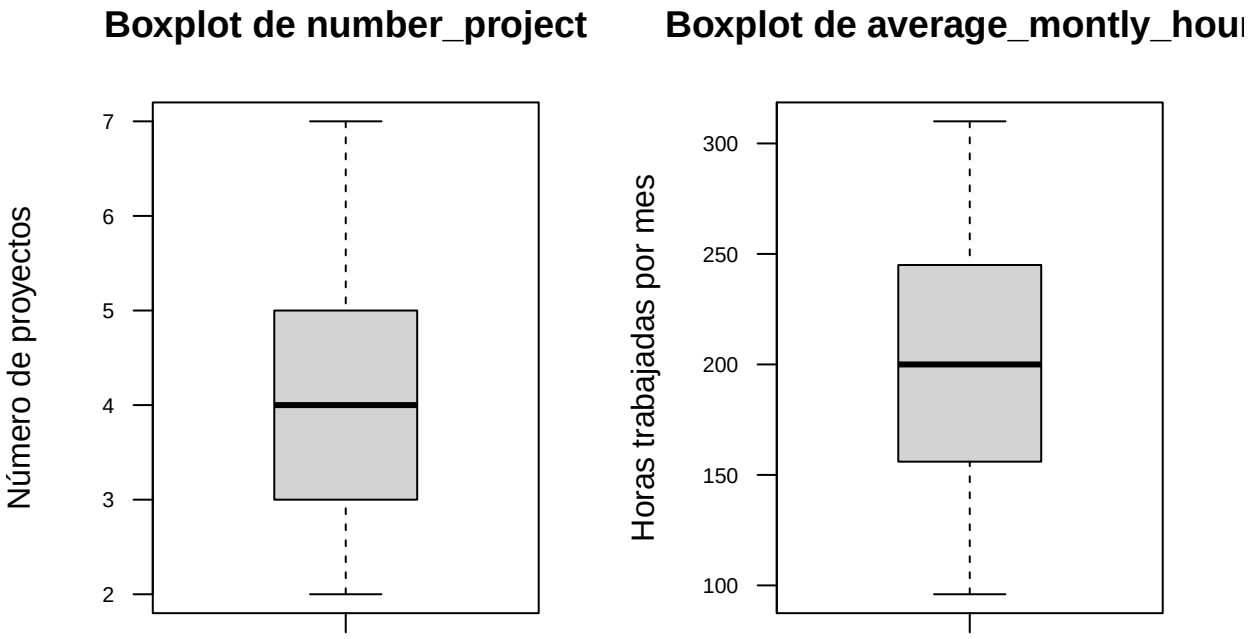
`satisfaction_level`:

- El boxplot muestra que el nivel de satisfacción de los empleados varía entre aproximadamente 0.1 y 1.0, con una mediana en torno a 0.6. La mayoría de los datos están concentrados entre 0.44 (primer cuartil) y 0.82 (tercer cuartil), indicando una distribución relativamente simétrica. No hay valores atípicos (outliers), lo que sugiere que los niveles de satisfacción están bastante concentrados dentro de este rango.

`last_evaluation`:

- El boxplot de la evaluación de desempeño muestra una distribución que abarca desde aproximadamente 0.36 hasta 1.0, con una mediana alrededor de 0.72. La mayoría de los empleados tienen una evaluación de desempeño entre 0.56 y 0.87. Tampoco se observan outliers en esta variable, indicando una dispersión similar a la de satisfaction_level.

```r
# Ajusto el tamaño de la ventana gráfica
options(repr.plot.width = 10, repr.plot.height = 5)

# Divido el área de gráficos en una disposición de 1 fila por 2 columnas
par(mfrow = c(1, 2))

# Creo el primer boxplot para number_project
boxplot(datos$number_project,
        main = "Boxplot de number_project",
        ylab = "Número de proyectos",
        xlab = "",
        outline = TRUE,
        axes = FALSE)  # Oculto ejes para personalizarlos después

# Añado de nuevo los ejes
axis(2, las = 1, cex.axis = 0.7,
     at = pretty(datos$number_project),
     labels = format(pretty(datos$number_project), scientific = FALSE))
axis(1, at = 1, labels = "")
box()  # Añado el cuadro alrededor del gráfico

# Creo el segundo boxplot para average_montly_hours
boxplot(datos$average_montly_hours,
        main = "Boxplot de average_montly_hours",
        ylab = "Horas trabajadas por mes",
        xlab = "",
        outline = TRUE,
        axes = FALSE)  # Oculto ejes para personalizarlos después

# Añado de nuevo los ejes
axis(2, las = 1, cex.axis = 0.7,
     at = pretty(datos$average_montly_hours),
     labels = format(pretty(datos$average_montly_hours), scientific = FALSE))
axis(1, at = 1, labels = "")
box()  # Añado el cuadro alrededor del gráfico
```



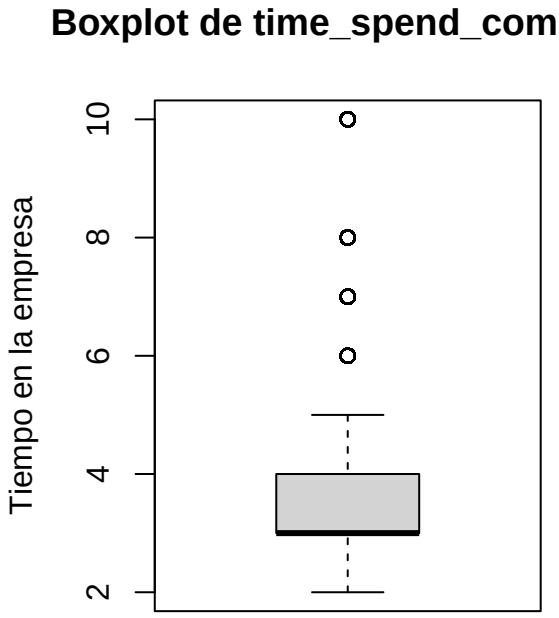
`number_project`:

El número de proyectos en los que los empleados están involucrados varía de 2 a 7, con una mediana en 4. La mayoría de los empleados tienen entre 3 y 5 proyectos. No hay outliers, lo que sugiere una distribución razonable de la carga de trabajo entre los empleados.

`average_montly_hours`:

Las horas trabajadas por mes varían de 96 a 310, con una mediana de aproximadamente 200. El rango intercuartílico (IQR) sugiere que la mayoría de los empleados trabajan entre 156 y 245 horas mensuales. No hay outliers significativos, lo que indica que la mayoría de los empleados tienen un horario laboral dentro de un rango esperado.

```r
#grafico time_spend_company
# Creo un boxplot
par(mfrow = c(1, 2))
boxplot(datos$time_spend_company,
        main = "Boxplot de time_spend_company",
        ylab = "Tiempo en la empresa",
        xlab = "",
        outline = TRUE)
```



`time_spend_company`:

El tiempo de permanencia en la empresa varía de 2 a 10 años, con una mediana de aproximadamente 3 años. El grueso de los empleados tiene una permanencia de entre 3 a 4 años aprox.

La siguiente celda calcula el rango intercuartílico (IQR) de la variable satisfaction_level para identificar los outliers. El IQR es una medida de la dispersión de los datos y se utiliza para detectar valores atípicos que están significativamente alejados del rango intercuartílico. Y luego identifico los outliers y los cuento.

```
# Cálculo del rango intercuartílico de satisfaction_level
q1 <- quantile(datos$satisfaction_level, 0.25)
q3 <- quantile(datos$satisfaction_level, 0.75)
iqr <- q3 - q1
lim_inf <- q1 - 1.5 * iqr
lim_sup <- q3 + 1.5 * iqr
# Identificación de los outliers
outliers <- datos %>%
  filter(satisfaction_level < lim_inf | satisfaction_level > lim_sup)
#numero de outliers
nrow(outliers)
```

```
## [1] 0
```

```
# Cálculo del rango intercuartílico de last_evaluation
q1 <- quantile(datos$last_evaluation, 0.25)
q3 <- quantile(datos$last_evaluation, 0.75)
iqr <- q3 - q1
lim_inf <- q1 - 1.5 * iqr
lim_sup <- q3 + 1.5 * iqr
# Identificación de los outliers
outliers <- datos %>%
  filter(last_evaluation < lim_inf | last_evaluation > lim_sup)
#numero de outliers
nrow(outliers)
```

```
## [1] 0
```

```
# Cálculo del rango intercuartílico de number_project
q1 <- quantile(datos$number_project, 0.25)
q3 <- quantile(datos$number_project, 0.75)
iqr <- q3 - q1
lim_inf <- q1 - 1.5 * iqr
lim_sup <- q3 + 1.5 * iqr
# Identificación de los outliers
outliers <- datos %>%
  filter(number_project < lim_inf | number_project > lim_sup)
#numero de outliers
nrow(outliers)
```

```
## [1] 0
```

```
# Cálculo del rango intercuartílico de average_montly_hours
q1 <- quantile(datos$average_montly_hours, 0.25)
q3 <- quantile(datos$average_montly_hours, 0.75)
iqr <- q3 - q1
lim_inf <- q1 - 1.5 * iqr
lim_sup <- q3 + 1.5 * iqr
# Identificación de los outliers
outliers <- datos %>%
  filter(average_montly_hours < lim_inf | average_montly_hours > lim_sup)
#numero de outliers
nrow(outliers)
```

```
## [1] 0
```

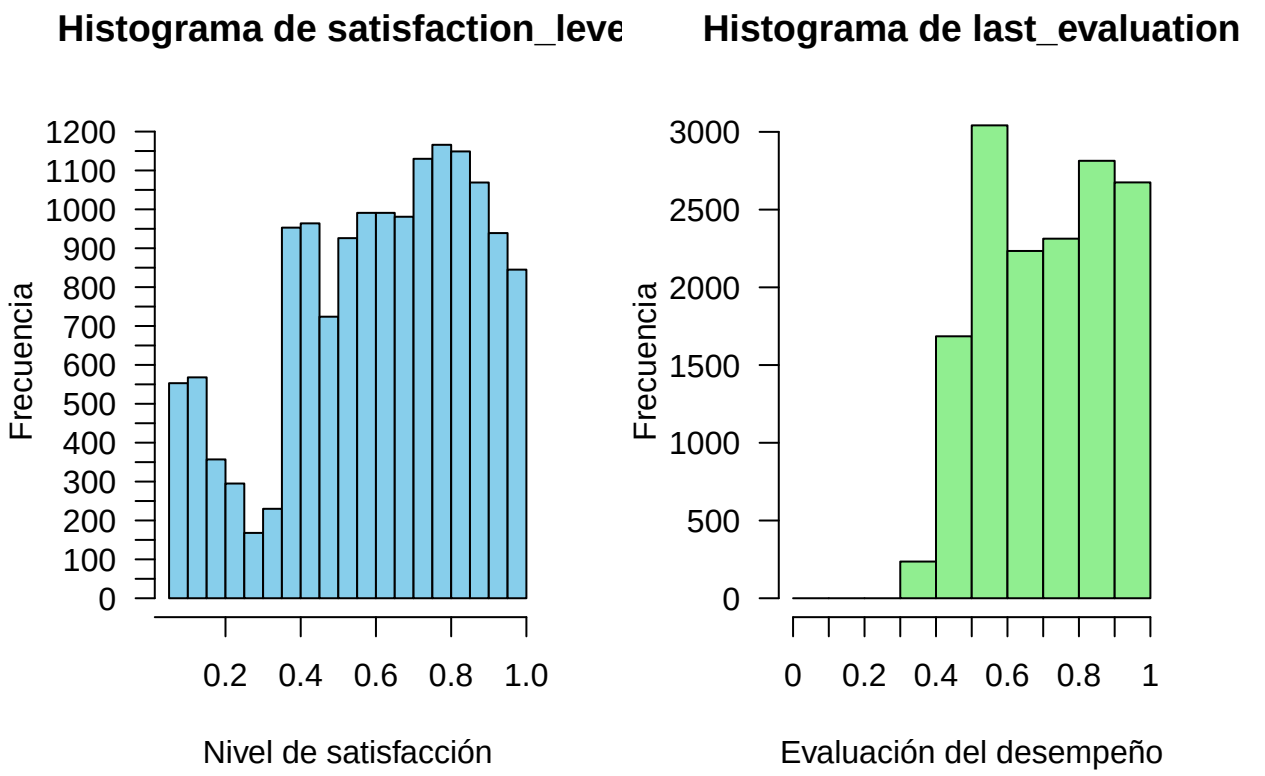
```
# Cálculo del rango intercuartílico de time_spend_company
q1 <- quantile(datos$time_spend_company, 0.25)
q3 <- quantile(datos$time_spend_company, 0.75)
iqr <- q3 - q1
lim_inf <- q1 - 1.5 * iqr
lim_sup <- q3 + 1.5 * iqr
# Identificación de los outliers
outliers <- datos %>%
  filter(time_spend_company < lim_inf | time_spend_company > lim_sup)
#numero de outliers
nrow(outliers)
```

```
## [1] 1282
```

La próxima celda tiene como objetivo crear histogramas de variables para visualizar la distribución de los datos. La visualización gráfica es una parte fundamental del análisis descriptivo y ayuda a identificar patrones y tendencias en los datos.

```
# Ajusto el tamaño de la ventana gráfica
options(repr.plot.width = 10, repr.plot.height = 5)

# Divido el área de gráficos en una disposición de 1 fila por 2 columnas
```

```r
par(mfrow = c(1, 2))

# Obtengo las frecuencias del histograma de satisfaction_level sin dibujarlo
hist_info <- hist(datos$satisfaction_level, breaks = 30, plot = FALSE)

# Calculo la frecuencia máxima
max_freq <- max(hist_info$counts)

# Creo el histograma para satisfaction_level con el ajuste del eje y
hist(datos$satisfaction_level,
     main = "Histograma de satisfaction_level",
     xlab = "Nivel de satisfacción",
     ylab = "Frecuencia",
     col = "skyblue",
     breaks = 30,
     ylim = c(0, max_freq + 50),  # Ajusto el rango del eje y
     axes = FALSE)  # Oculto ejes para personalizarlos después

# Añado los ejes personalizados
axis(1, at = pretty(datos$satisfaction_level),
     labels = format(pretty(datos$satisfaction_level), scientific = FALSE))
axis(2, las = 1, at = seq(0, max_freq + 50, by = 50))

# Creo el segundo histograma para last_evaluation con intervalos ajustados
max_eval <- max(datos$last_evaluation)
breaks_last_eval <- seq(floor(min(datos$last_evaluation)),
                        max_eval + (0.1 - max_eval %% 0.1), by = 0.1)
hist(datos$last_evaluation,
     main = "Histograma de last_evaluation",
     xlab = "Evaluación del desempeño",
     ylab = "Frecuencia",
     col = "lightgreen",
     # Intervalos ajustados para cubrir todo el rango
     breaks = breaks_last_eval,
     # Oculto ejes para personalizarlos después
     axes = FALSE)

# Añado los ejes personalizados
axis(1, at = seq(floor(min(datos$last_evaluation)),
                 max_eval + (0.1 - max_eval %% 0.1), by = 0.1),
     labels = seq(floor(min(datos$last_evaluation)),
                  max_eval + (0.1 - max_eval %% 0.1), by = 0.1))
axis(2, las = 1)
```



`satisfaction_level`:

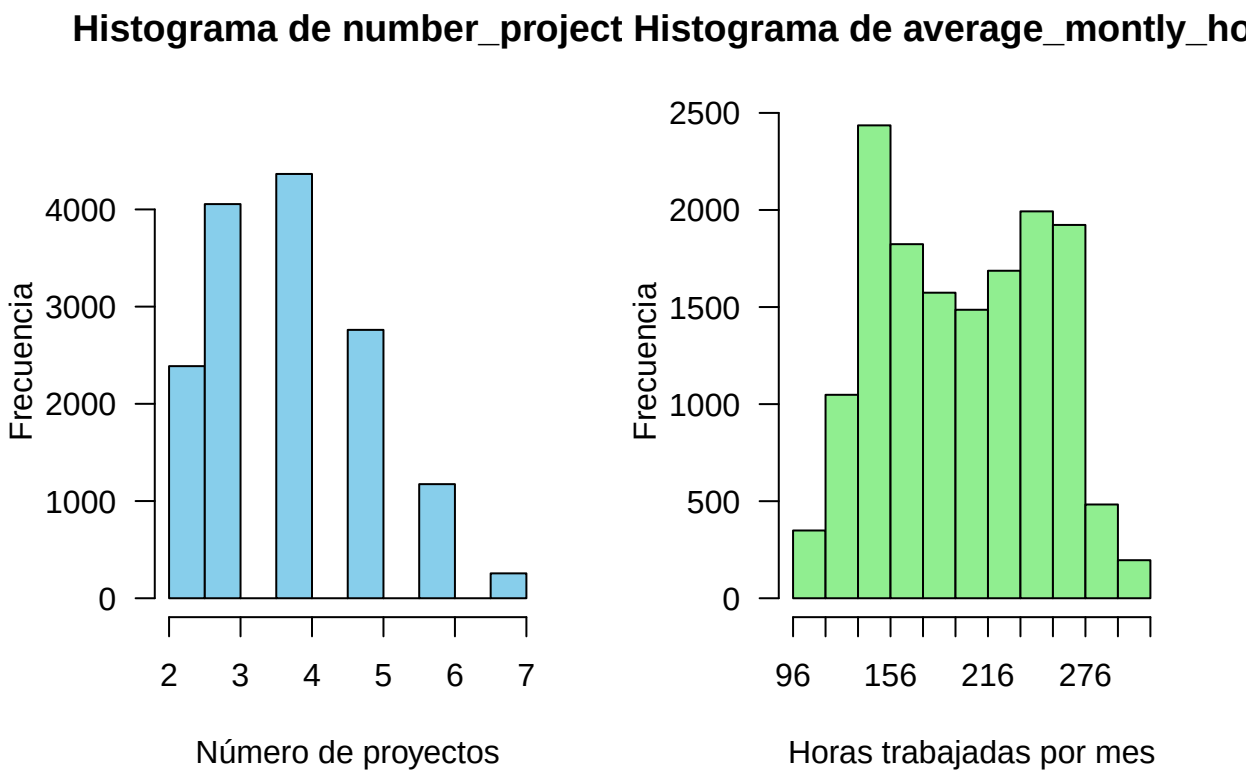
El histograma muestra una distribución multimodal, con maximos alrededor de los niveles de satisfacción de 0.4 y 0.8. Esto sugiere que hay dos grupos principales de empleados: uno con niveles de satisfacción moderados y otro con niveles de satisfacción altos. También se observa una menor frecuencia en niveles muy bajos o muy altos de satisfacción.

`last_evaluation`:

La distribución de la evaluación del desempeño tiene una forma sesgada a la derecha, con un maximo notable alrededor de 0.6 a 0.8. Esto indica que la mayoría de los empleados tiene evaluaciones de desempeño por encima del promedio, con relativamente pocos empleados obteniendo evaluaciones bajas. Esta distribución podría reflejar un sistema de evaluación que favorece calificaciones positivas o que la mayoría de los empleados están cumpliendo o superando las expectativas.

```r
# Ajusto el tamaño de la ventana gráfica
options(repr.plot.width = 10, repr.plot.height = 5)

# Divido el área de gráficos en una disposición de 1 fila por 2 columnas
par(mfrow = c(1, 2))

# Obtengo las frecuencias del histograma de number_project sin dibujarlo
hist_info <- hist(datos$number_project, breaks = 10, plot = FALSE)

# Calculo la frecuencia máxima
max_freq <- max(hist_info$counts)

# Creo el histograma para number_project con el ajuste del eje y
hist(datos$number_project,
     main = "Histograma de number_project",
     xlab = "Número de proyectos",
```

```r
     ylab = "Frecuencia",
     col = "skyblue",
     breaks = 15,  # Aumento el número de barras
     ylim = c(0, max_freq + 500),  # Ajusto el rango del eje y
     axes = FALSE)  # Oculto ejes para personalizarlos después

# Añado los ejes personalizados
axis(1, at = seq(min(datos$number_project),
                 max(datos$number_project),
                 by = 1),
     labels = seq(min(datos$number_project),
                  max(datos$number_project),
                  by = 1))
axis(2, las = 1, at = seq(0, max_freq + 500, by = 1000))

# Creo el segundo histograma para average_montly_hours con intervalos ajustados
max_hours <- max(datos$average_montly_hours)
breaks_hours <- seq(floor(min(datos$average_montly_hours)),
                    max_hours + (20 - max_hours %% 20), by = 20)
hist(datos$average_montly_hours,
     main = "Histograma de average_montly_hours",
     xlab = "Horas trabajadas por mes",
     ylab = "Frecuencia",
     col = "lightgreen",
     # Intervalos ajustados para cubrir todo el rango
     breaks = breaks_hours,
     # Oculto ejes para personalizarlos después
     axes = FALSE)

# Añado los ejes personalizados
axis(1, at = seq(floor(min(datos$average_montly_hours)),
                 max_hours + (20 - max_hours %% 20), by = 20),
     labels = seq(floor(min(datos$average_montly_hours)),
                  max_hours + (20 - max_hours %% 20), by = 20))
axis(2, las = 1)
```



`number_project`:

El histograma muestra que la mayoría de los empleados están involucrados en 3 o 4 proyectos, con una caída significativa en la frecuencia para los empleados con 5 o más proyectos. Esta distribución podría sugerir que la carga de trabajo se concentra en un número específico de proyectos, y pocos empleados están llevando más de 5 proyectos simultáneamente.

`average_montly_hours`:

La distribución de las horas mensuales trabajadas es bimodal, con maximos alrededor de 150 y 250 horas. Esto podría indicar que hay dos patrones de trabajo: uno con una carga horaria moderada y otro con una carga horaria más alta. La existencia de dos grupos distintos podría estar relacionada con diferentes roles o niveles de responsabilidad dentro de la empresa.

```
# Ajusto el tamaño de la ventana gráfica
options(repr.plot.width = 5, repr.plot.height = 5)

# Obtengo las frecuencias del histograma de time_spend_company sin dibujarlo
hist_info <- hist(datos$time_spend_company, breaks = 5, plot = FALSE)

# Calculo la frecuencia máxima
max_freq <- max(hist_info$counts)

# Creo el histograma para time_spend_company con el ajuste del eje y
hist(datos$time_spend_company,
     main = "Histograma de time_spend_company",
     xlab = "Tiempo en la empresa",
     ylab = "Frecuencia",
     col = "skyblue",
     breaks = 5,  # Ajusto los breaks para mejor visualización
     ylim = c(0, max_freq + 1000),  # Ajusto el rango del eje y
```

```
     axes = FALSE)  # Oculto ejes para personalizarlos después

# Añado los ejes personalizados
axis(1, at = seq(min(datos$time_spend_company),
                 max(datos$time_spend_company),
                 by = 1),
     labels = seq(min(datos$time_spend_company),
                  max(datos$time_spend_company),
                  by = 1))
axis(2, las = 1, at = seq(0, max_freq + 1000, by = 2000))
```

`time_spend_company`:

Este histograma muestra una distribución altamente sesgada a la izquierda, con la mayoría de los empleados habiendo trabajado entre 2 y 4 años en la empresa. Un número significativamente menor de empleados ha permanecido más de 5 años, y muy pocos han trabajado 7 años o más. Esta distribución sugiere una alta tasa de rotación en los primeros años de empleo, lo que podría ser un punto de interés para investigar más a fondo en términos de retención de empleados.

```
variables <- c("satisfaction_level", "last_evaluation",
               "number_project", "average_montly_hours",
               "time_spend_company")
for (variable in variables) {
  varianza <- var(datos[, variable])
  desviacion_estandar <- sd(datos[, variable])
  print(paste("Varianza de", variable, ":", varianza),
        quote = FALSE)
  print(paste("Desviación estándar de", variable, ":", desviacion_estandar),
        quote = FALSE)
}
```

```
## [1] Varianza de satisfaction_level : 0.0618172006470876
## [1] Desviación estándar de satisfaction_level : 0.248630651061143
## [1] Varianza de last_evaluation : 0.0292988644315631
## [1] Desviación estándar de last_evaluation : 0.171169110623275
## [1] Varianza de number_project : 1.51928391438924
## [1] Desviación estándar de number_project : 1.23259235531835
## [1] Varianza de average_montly_hours : 2494.31317480996
## [1] Desviación estándar de average_montly_hours : 49.9430993712841
## [1] Varianza de time_spend_company : 2.13199781172236
## [1] Desviación estándar de time_spend_company : 1.46013623053548
```

Item 4

4. Cree la variable "number_project2" donde omita 100 valores para la variable "number_project" de forma aleatoria, suponga que algunos colaboradores no conocían el número de proyectos en el que participaron durante el último año y omitieron su respuesta al momento de ser encuestados. Mantenga la copia de la variable original e inserte la semilla 12345.

```r
# Fijo la semilla
set.seed(12345)

# Copio la variable original
datos$number_project2 <- datos$number_project

# Selecciono aleatoriamente 100 filas para asignar NA a "number_project2"
filas_con_na <- sample(1:nrow(datos), 100)

# Asigno NA a las filas seleccionadas
datos$number_project2[filas_con_na] <- NA

# Verifico cuántos valores faltantes hay en la nueva variable
sum(is.na(datos$number_project2))
```

```
## [1] 100
```

Item 5

5. Suponga que la variable de interés principal es "satisfaction_level". Por lo tanto, realice un breve análisis descriptivo de dicha variable considerando solo aquellas observaciones con valores perdidos en "number_project2" y repita lo mismo para aquellos casos sin valores perdidos en "number_project2". ¿Son similares los 2 conjuntos de resultados? Realice un test t para la diferencia de media de "satisfaction_level" considerando la comparación entre aquel grupo con valores perdidos en "number_project2" y aquel grupo sin valores perdidos. Concluya sobre cómo se distribuyen los valores perdidos en relación a su variable principal de interés, "satisfaction_level".

```r
# Separo los datos en dos grupos: con y sin valores perdidos
grupo_na <- datos[is.na(datos$number_project2), ]
grupo_no_na <- datos[!is.na(datos$number_project2), ]

# Análisis para observaciones con valores perdidos en number_project2
summary_perdidos <- summary(grupo_na$satisfaction_level)
```

```r
sd_perdidos <- sd(grupo_na$satisfaction_level)

# Análisis para observaciones sin valores perdidos en number_project2
summary_no_perdidos <- summary(grupo_no_na$satisfaction_level)
sd_no_perdidos <- sd(grupo_no_na$satisfaction_level)

# Imprimo resultados
print("Resumen para observaciones con valores perdidos:")
```

```
## [1] "Resumen para observaciones con valores perdidos:"
```

```r
print(summary_perdidos)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0900  0.4575  0.6800  0.6246  0.8125  0.9900
```

```r
print(paste("Desviación estándar:", sd_perdidos))
```

```
## [1] "Desviación estándar: 0.249433378081223"
```

```r
print("Resumen para observaciones sin valores perdidos:")
```

```
## [1] "Resumen para observaciones sin valores perdidos:"
```

```r
print(summary_no_perdidos)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0900  0.4400  0.6400  0.6128  0.8200  1.0000
```

```r
print(paste("Desviación estándar:", sd_no_perdidos))
```

```
## [1] "Desviación estándar: 0.248631771187715"
```

```r
# Test t para la diferencia de medias
t_test_result <- t.test(grupo_na$satisfaction_level,
                        grupo_no_na$satisfaction_level)

print("Resultados del test t:")
```

```
## [1] "Resultados del test t:"
```

```r
print(t_test_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  grupo_na$satisfaction_level and grupo_no_na$satisfaction_level
```

```
## t = 0.47332, df = 100.32, p-value = 0.637
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.03780419  0.06149509
## sample estimates:
## mean of x mean of y
## 0.6246000 0.6127545
```

Análisis descriptivo

El análisis descriptivo de la variable "satisfaction_level" para las observaciones con valores perdidos en "number_project2" (grupo_na) y sin valores perdidos (grupo_no_na) muestra que ambos conjuntos tienen una distribución similar.

La media de la satisfacción laboral para el grupo con valores perdidos es de 0,6246, mientras que para el grupo sin valores perdidos es de 0,6128. Estas medias son muy similares.

La desviación estándar para el grupo con valores perdidos es de 0,2494, mientras que para el grupo sin valores perdidos es de 0,2486. Estas desviaciones estándar también son muy similares.

Análisis Resultado del Test t

El test t muestra un p-valor de 0.637, que es mucho mayor que el nivel de significancia típico de 0.05. Esto indica que no hay evidencia estadística suficiente para rechazar la hipótesis nula de que las medias de "satisfaction_level" son iguales en ambos grupos.

El valor de t es de 0,47332, lo que indica que la diferencia entre las medias no es estadísticamente significativa.

Además, el intervalo de confianza del 95% para la diferencia de medias incluye el cero (-0.0378, 0.0615), lo que refuerza la conclusión de que no hay una diferencia estadísticamente significativa entre los grupos.

Basándonos en los resultados del análisis descriptivo y el test t, podemos concluir que los valores perdidos en la variable 'number_project2' se distribuyen aleatoriamente en relación a la variable 'satisfaction_level'. En otras palabras, la falta de información sobre el número de proyectos no parece estar relacionada con el nivel de satisfacción de los empleados.

Esta conclusión se alinea con el concepto de **Missing Completely at Random (MCAR)** que es cuando los datos son MCAR, la probabilidad de que un dato falte no depende de ninguna variable, ni observada ni no observada. En este caso, la omisión de la respuesta sobre el número de proyectos parece ser aleatoria y no está influenciada por el nivel de satisfacción del empleado.

Item 6

6. Realice el método de imputación Hot Deck para obtener una estimación de los valores perdidos a ser imputados. Para ello use las variables "satisfaction_level" y "salary" para apoyar el proceso. Compare sus resultados con el vector original y reporte su RMSE.

```r
# Realizo la imputación Hot Deck
datos_imputados <- hotdeck(datos, variable = "number_project2",
                           ord_var = c("satisfaction_level", "salary"))

# Extraigo los valores imputados
valores_imputados <- datos_imputados$number_project2

# Filtro solo los valores que originalmente eran NA para compararlos
originales_na <- datos$number_project[is.na(datos$number_project2)]
imputados_na <- valores_imputados[is.na(datos$number_project2)]
```

```r
# Calculo el RMSE solo para los valores imputados
rmse <- sqrt(mean((originales_na - imputados_na)^2))

# Imprimo el RMSE
print(paste("RMSE:", rmse))
```

```
## [1] "RMSE: 1.17046999107196"
```

```r
# Comparo los primeros valores originales e imputados
# (solo donde hubo imputación)
head(data.frame(Original = originales_na,
                Imputado = imputados_na,
                Diferencia = originales_na - imputados_na))
```

```
##   Original Imputado Diferencia
## 1        7        6          1
## 2        6        6          0
## 3        2        2          0
## 4        2        2          0
## 5        4        4          0
## 6        4        5         -1
```

- El método Hot Deck se basa en la idea de encontrar "donantes" (observaciones con valores completos) que sean similares a las observaciones con valores faltantes, basándose en las variables de apoyo. En este caso, las variables satisfaction_level y salary se utilizaron para identificar donantes con características similares a las observaciones con valores faltantes en number_project2.
- La precisión del método Hot Deck depende de la calidad de las variables de apoyo y de la similitud entre los donantes y las observaciones con valores faltantes. En este caso, los resultados sugieren que las variables satisfaction_level y salary fueron adecuadas para encontrar donantes similares y generar imputaciones precisas.
- Se utilizaron las variables "satisfaction_level" y "salary" como variables de ordenamiento (ord_var) en el proceso de imputación. Esto significa que el algoritmo buscará donantes que sean similares en términos de nivel de satisfacción y salario, lo cual es coherente con la solicitud de la pregunta.
- El método Hot Deck es apropiado en este tipo de situaciones donde los datos faltantes pueden estar asociados a grupos similares en la data, definidos por variables auxiliares. En este caso, el uso de satisfaction_level y salary es lógico, ya que estas variables podrían estar relacionadas con el número de proyectos en los que un empleado ha trabajado.

Interpretación del RMSE

- El RMSE de 1.1704 indica que, en promedio, la diferencia entre los valores imputados y los valores originales es de aproximadamente 1.17 unidades. Esto es un resultado razonable para un método de imputación que no pretende ser exacto, sino que busca mantener la coherencia en el contexto de las variables (satisfaction_level y salary).
- El RMSE está en la misma escala que la variable original (number_project).
- Un RMSE moderado como el que se obtuvo es esperable en un método como Hot Deck, donde la imputación se basa en la similitud y no en la exactitud. Según la teoría de valores perdidos, es más importante mantener la coherencia en la distribución general de los datos que obtener valores exactos, lo que se ha logrado este caso.

Item 7

7. Realice un método de imputación MICE para obtener una estimación de los valores perdidos. Utilice al menos 5 imputaciones, compare sus resultados con el vector original y reporte sus RMSEs (uno por cada conjunto de imputaciones, es decir, 5 RMSE).

```
# Preparo los datos para la imputación
# Selecciono las variables relevantes
datos_imp <- datos[, c("number_project2", "satisfaction_level", "salary")]

# Realizo la imputación MICE
imp <- mice(datos_imp, m = 5, maxit = 50, method = 'pmm', seed = 500)
```

```
# Extraigo los valores imputados
valores_imputados <- complete(imp, "all")

# Calculo el RMSE para cada conjunto de imputaciones
calcular_rmse <- function(originales, imputados) {
  sqrt(mean((originales - imputados)^2, na.rm = TRUE))
}

rmse_resultados <- sapply(1:5, function(i) {
  originales <- datos$number_project[is.na(datos$number_project2)]
  imputados <- valores_imputados[[i]]$number_project2[is.na(datos$number_project2)]
  calcular_rmse(originales, imputados)
})

# Muestro los resultados
print("RMSE para cada conjunto de imputaciones:")
```

```
## [1] "RMSE para cada conjunto de imputaciones:"
```

```
print(rmse_resultados)
```

```
## [1] 1.212436 1.303840 1.637071 1.244990 1.256981
```

```
# Calculo y muestro el RMSE promedio
rmse_promedio <- mean(rmse_resultados)
print(paste("RMSE promedio:", rmse_promedio))
```

```
## [1] "RMSE promedio: 1.33106341390195"
```

```
# Comparo los primeros valores originales e imputados del primer conjunto
head(data.frame(
  Original = datos$number_project[is.na(datos$number_project2)],
  Imputado = valores_imputados[[1]]$number_project2[is.na(datos$number_project2)],
  Diferencia = datos$number_project[is.na(datos$number_project2)] -
               valores_imputados[[1]]$number_project2[is.na(datos$number_project2)]
))
```

```
##   Original Imputado Diferencia
## 1        7        7          0
## 2        6        6          0
## 3        2        2          0
## 4        2        3         -1
## 5        4        4          0
## 6        4        5         -1
```

El método MICE es un método de imputación múltiple que genera varios conjuntos de datos imputados, teniendo en cuenta la incertidumbre asociada a la imputación. El método 'pmm' se utilizó para imputar variables numéricas y busca donantes con valores predichos similares para la variable a imputar.

El RMSE promedio es 1.331063. Este valor indica el error promedio entre los valores imputados y los valores originales de number_project en los casos donde se realizaron imputaciones.

Los RMSE obtenidos son:

- 1.212436
- 1.303840
- 1.637071
- 1.244990
- 1.256981

Los RMSE varían entre aproximadamente 1.21 y 1.63, lo que indica cierta variabilidad en la precisión de las imputaciones entre los diferentes conjuntos.

Los RMSE obtenidos son relativamente cercanos entre sí, lo que sugiere que el método de imputación está siendo consistente en cada uno de los cinco conjuntos de imputaciones.

Basado en el análisis, podemos concluir que la imputación realizada es efectiva y los valores imputados se aproximan adecuadamente a los valores originales, proporcionando una estimación confiable para los datos faltantes.

Para los valores originales 7, 6, 2, 2, 4, y 4, los valores imputados fueron 7, 6, 2, 3, 4, y 5, respectivamente.

La diferencia es mínima en la mayoría de los casos (0 en la mayoría y -1 en algunos), lo que es un buen indicador de que la imputación está funcionando correctamente.

Item 8

8. Investigue y explique cómo comparar una metodología de imputación (no múltiple) como Hot Deck, donde se obtiene solamente un conjunto de datos imputados, respecto de una de imputación múltiple. Además, señale cuáles son las ventajas y desventajas de usar cada algoritmo.

Comparación de Metodologías de Imputación no múltiple y múltiple

Sesgo y Varianza: Los métodos de imputación no múltiple como Hot Deck pueden introducir sesgo y subestimar la varianza, ya que solo proporcionan una única estimación de los valores perdidos. Los métodos de imputación múltiple, por otro lado, pueden reducir el sesgo y proporcionar una estimación más precisa de la varianza al tener en cuenta la incertidumbre asociada con la imputación de valores perdidos.

Suposiciones del Modelo: Los métodos de imputación no múltiple a menudo dependen de suposiciones fuertes del modelo, como la suposición de datos perdidos al azar (MAR, por sus siglas en inglés). Los métodos de imputación múltiple pueden relajar estas suposiciones y proporcionar resultados más robustos.

Complejidad Computacional: Los métodos de imputación no múltiple son generalmente más simples y rápidos desde el punto de vista computacional en comparación con los métodos de imputación múltiple, que requieren la creación de múltiples conjuntos de datos imputados.

Complejidad de los Datos: Los métodos de imputación no múltiple pueden no funcionar bien con estructuras de datos complejas, como datos longitudinales o jerárquicos. Los métodos de imputación múltiple pueden manejar estas estructuras de datos complejas de manera más efectiva.

Ventajas metodos imputación no múltiple (por ej: Hot Deck) - Simple y eficiente desde el punto de vista computacional. - Fácil de implementar e interpretar. - Puede preservar distribuciones multinomiales e imputar valores cero.

Desventajas metodos imputación no múltiple (por ej: Hot Deck) - Puede introducir sesgo y subestimar la varianza. - Depende de suposiciones fuertes del modelo. - Puede no funcionar bien con estructuras de datos complejas.

Ventajas metodos imputación múltiple - Puede reducir el sesgo y proporcionar una estimación más precisa de la varianza. - Puede relajar las suposiciones del modelo y proporcionar resultados más robustos. - Puede manejar estructuras de datos complejas de manera efectiva. - Proporciona una imagen más completa de los datos.

Desventajas metodos imputación múltiple - Computacionalmente más complejo y requiere más tiempo. - Requiere una consideración cuidadosa del modelo de imputación y los parámetros. - Puede ser difícil de interpretar y comunicar los resultados.

Item 9

Retome nuevamente el conjunto de datos original y considere como variable objetivo "satisfaction_level".

9. Realice un problema de aprendizaje supervisado de regresión mediante el método de regularización, prediga la variable objetivo utilizando regresión de lasso, ridge y elastic net. Considere los siguientes aspectos:

- Evalúe si es necesario escalar los datos.
- Separe en un conjunto de entrenamiento y otro de prueba.
- Ajuste el/los hiperparámetro/s mediante algún procedimiento de remuestreo.
- Prediga los resultados en el conjunto de prueba para cada modelo.
- Registre RMSE de cada modelo.

```
# Importo la base de datos y la cargo como dataframe
datos <- as.data.frame(read.csv("Employee Attrition.csv"))
```

```
# Elimino la variable "Emp.ID"
datos <- datos[, -which(names(datos) == "Emp.ID")]
```

Evalúo si es necesario escalar los datos

```
# Verifico la escala de las variables numéricas
summary(datos[, sapply(datos, is.numeric)])
```

```
##  satisfaction_level last_evaluation  number_project  average_montly_hours
##  Min.   :0.0900     Min.   :0.3600   Min.   :2.000   Min.   : 96.0
##  1st Qu.:0.4400     1st Qu.:0.5600   1st Qu.:3.000   1st Qu.:156.0
##  Median :0.6400     Median :0.7200   Median :4.000   Median :200.0
##  Mean   :0.6128     Mean   :0.7161   Mean   :3.803   Mean   :201.1
##  3rd Qu.:0.8200     3rd Qu.:0.8700   3rd Qu.:5.000   3rd Qu.:245.0
##  Max.   :1.0000     Max.   :1.0000   Max.   :7.000   Max.   :310.0
```

```
##  time_spend_company Work_accident    promotion_last_5years
##  Min.   : 2.000     Min.   :0.0000   Min.   :0.00000
##  1st Qu.: 3.000     1st Qu.:0.0000   1st Qu.:0.00000
##  Median : 3.000     Median :0.0000   Median :0.00000
##  Mean   : 3.498     Mean   :0.1446   Mean   :0.02127
##  3rd Qu.: 4.000     3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :10.000     Max.   :1.0000   Max.   :1.00000
```

Analizando los resultados del summary, podemos observar que las variables numéricas presentan diferentes escalas:

- satisfaction_level y last_evaluation: Ambas variables se encuentran en un rango de 0 a 1.
- number_project: El número de proyectos varía de 2 a 7.
- average_montly_hours: Las horas mensuales trabajadas van desde 96 hasta 310.
- time_spend_company: El tiempo en la empresa está entre 2 y 10 años.

Estas diferencias en la escala pueden influir negativamente en los modelos de regresión de regularización, como Lasso, Ridge, y Elastic Net. Modelos como estos, que utilizan regularización, son sensibles a la escala de las variables porque los coeficientes de regresión se ven afectados directamente por la magnitud de las variables independientes.

Es necesario escalar los datos antes de aplicar los modelos de regularización. Esto es especialmente importante para asegurarse de que todas las variables contribuyan equitativamente al modelo y que los coeficientes de regresión no se vean dominados por las variables con mayores magnitudes

```r
# Identifico variables categóricas y las convierto en factores
categorical_vars <- sapply(datos, is.character)
datos[categorical_vars] <- lapply(datos[categorical_vars], as.factor)
```

```r
# identifico missing en variable dependiente
sum(is.na(datos$satisfaction_level))
```

```
## [1] 0
```

Separo en un conjunto de entrenamiento y otro de prueba

```r
# Semilla
set.seed(2106)

# Partición de datos
training.samples <- createDataPartition(datos$satisfaction_level,
                                        p = 0.8, list = FALSE)
train.data <- datos[training.samples, ]
test.data <- datos[-training.samples, ]

# x e y en conjunto de entrenamiento
x <- model.matrix(satisfaction_level ~ ., train.data)[, -1]
y <- train.data$satisfaction_level

# x e y en conjunto de testeo
x_test <- model.matrix(satisfaction_level ~ ., test.data)[, -1]
y_test <- test.data$satisfaction_level
```

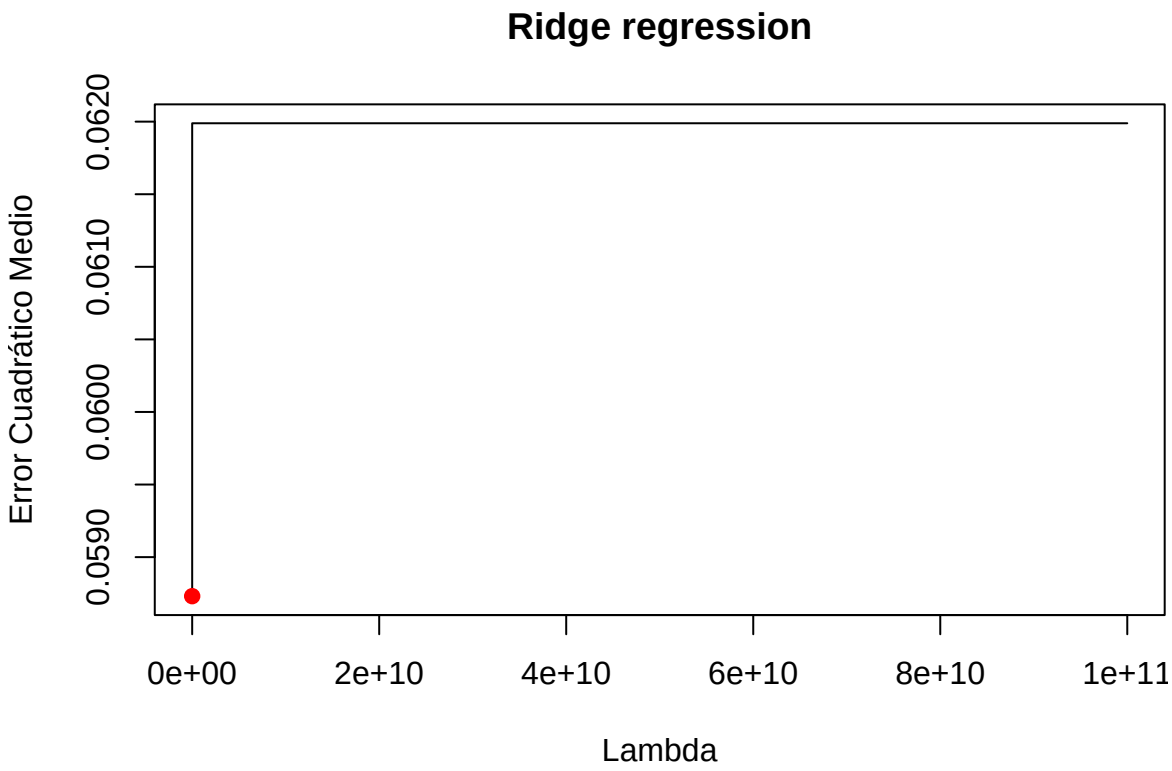
```r
# Grilla lambda
lambda <- 10^seq(11, -1, length = 200)
```

Ajusto los hiperparámetros mediante validación cruzada

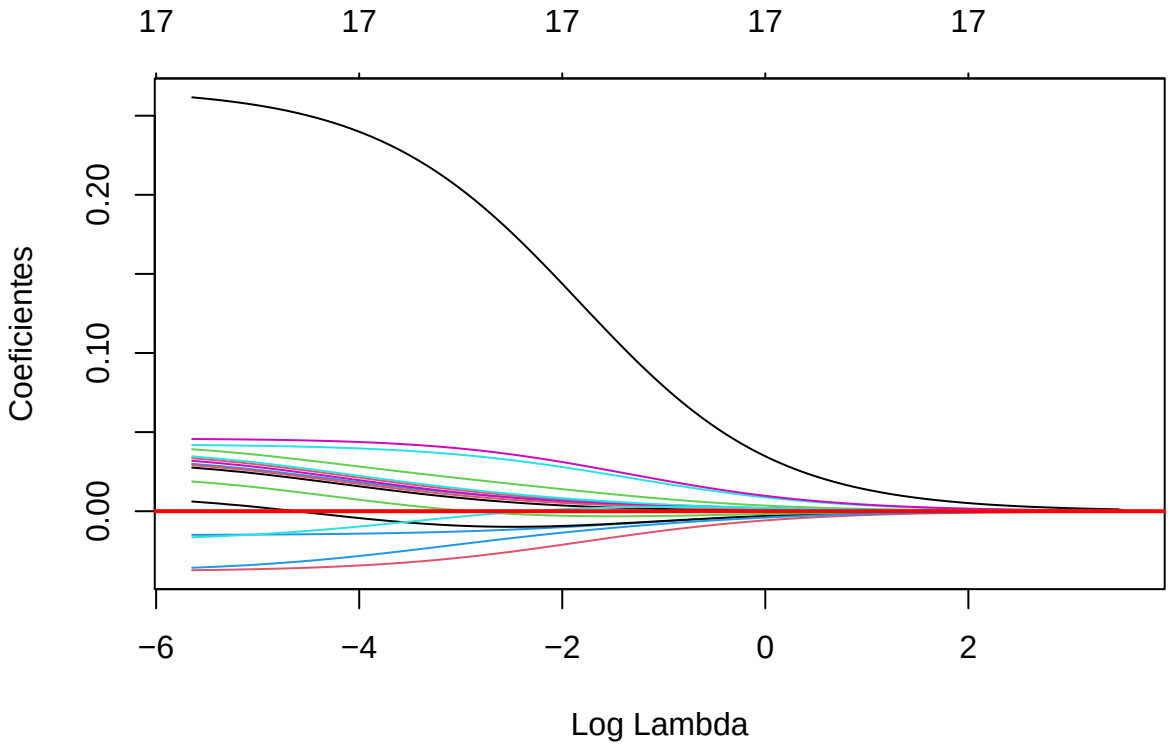
```r
# Ridge
ridge_cv <- cv.glmnet(x, y, alpha = 0, lambda = lambda,
                      standardize = TRUE, nfolds = 10)
bestlam_ridge <- ridge_cv$lambda.min
cat("Mejor lambda para Ridge:", bestlam_ridge, "\n")
```

```
## Mejor lambda para Ridge: 0.1
```

```r
# Gráfico ECM y lambda para Ridge
plot(ridge_cv$lambda, ridge_cv$cvm, type = "l",
     xlab = "Lambda", ylab = "Error Cuadrático Medio",
     main = "Ridge regression")
points(bestlam_ridge, min(ridge_cv$cvm), col = "red", pch = 19)
```




```r
# Gráfico coeficientes y log(Lambda) para Ridge
ridge.mod <- glmnet(x, y, alpha = 0)
plot(ridge.mod, xvar = "lambda", ylab = "Coeficientes")
abline(h = 0, lwd = 2, col = "red")
```



Al usar standardize = TRUE, me aseguré de que en el código todas las variables predictoras están en la misma escala.

Ajuste del hiperparámetro lambda:

- La validación cruzada (CV) se utiliza para ajustar el hiperparámetro lambda, que controla la cantidad de regularización aplicada en el modelo Ridge. El gráfico resultante muestra cómo varía el Error Cuadrático Medio (ECM) en función de lambda. El punto rojo en el gráfico marca el valor de lambda que minimiza el ECM, es decir, el mejor valor de lambda.
- El mejor valor de lambda encontrado mediante validación cruzada para la regresión Ridge es 0.1. Este valor relativamente bajo sugiere que el modelo no requiere una penalización muy fuerte, lo que indica que probablemente no hay un problema grave de multicolinealidad o sobreajuste en los datos.

Gráfico de Error Cuadrático Medio (ECM) vs Lambda:

- El gráfico demuestra que al aumentar lambda, el ECM se estabiliza, lo que indica que el modelo no mejora significativamente con una mayor regularización.

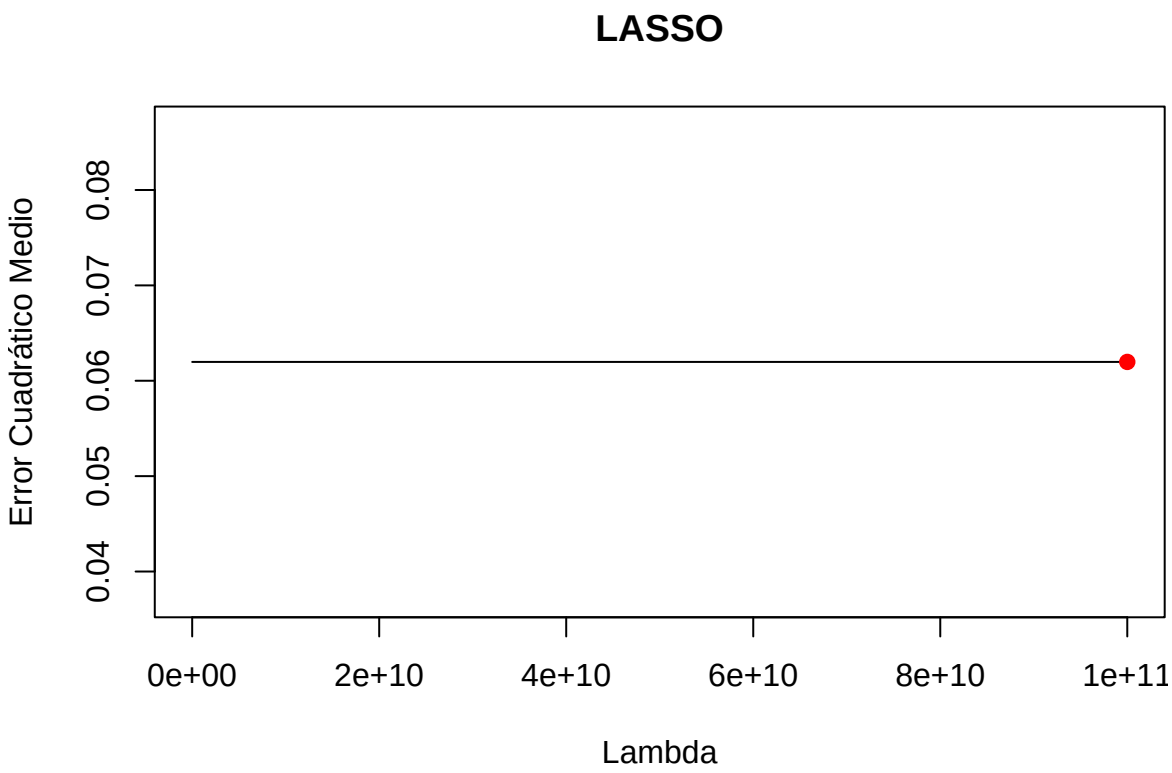
Gráfico de Coeficientes vs Log(Lambda):

- El segundo gráfico muestra cómo varían los coeficientes de las variables predictoras en función del logaritmo de lambda. A medida que lambda aumenta, los coeficientes tienden a acercarse a cero, lo cual es esperado en la regresión Ridge ya que el objetivo es reducir la complejidad del modelo sin eliminar completamente las variables predictoras (a diferencia del Lasso, que puede llevar coeficientes a cero).
- La línea roja horizontal en y=0 ayuda a visualizar qué coeficientes cambian de signo con la regularización.

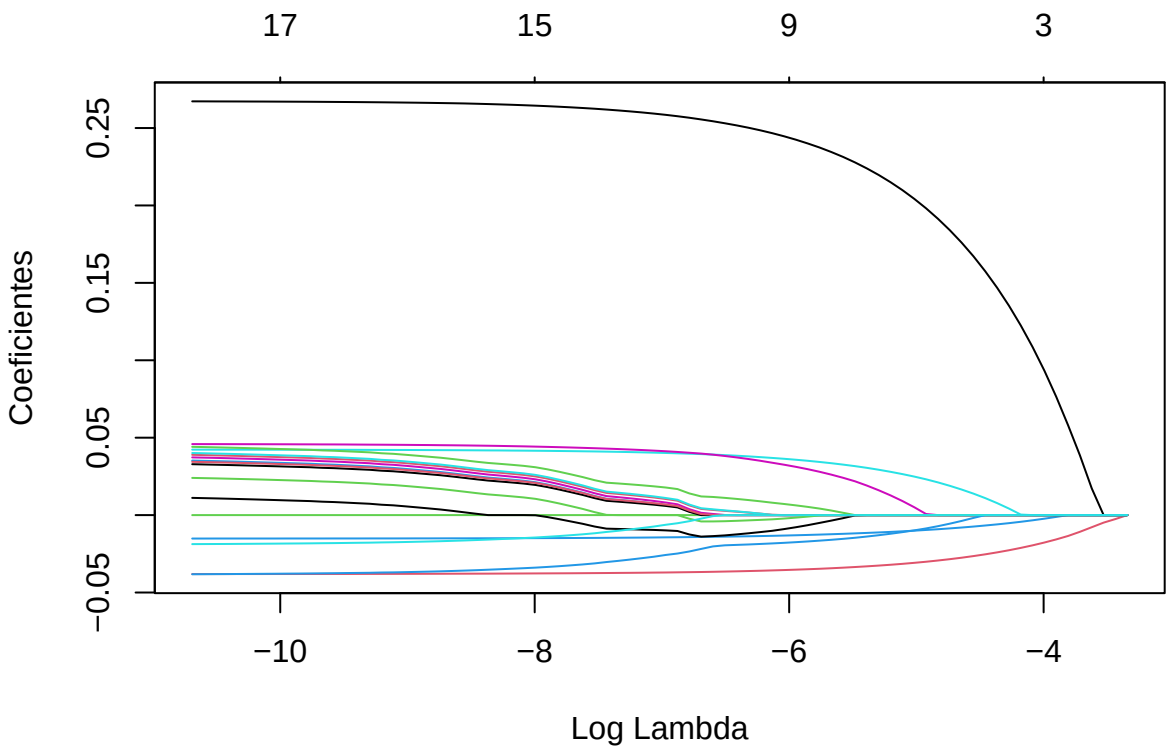
```r
# Lasso
lasso_cv <- cv.glmnet(x, y, alpha = 1, lambda = lambda,
                      standardize = TRUE, nfolds = 10)
bestlam_lasso <- lasso_cv$lambda.min
cat("Mejor lambda para Lasso:", bestlam_lasso, "\n")
```

```
## Mejor lambda para Lasso: 1e+11
```

```r
# Gráfico ECM y lambda para Lasso
plot(lasso_cv$lambda, lasso_cv$cvm, type = "l",
     xlab = "Lambda", ylab = "Error Cuadrático Medio", main = "LASSO")
points(bestlam_lasso, min(lasso_cv$cvm), col = "red", pch = 19)
```




```r
# Gráfico coeficientes y log(Lambda) para Lasso
lasso.mod <- glmnet(x, y, alpha = 1)
plot(lasso.mod, xvar = "lambda", ylab = "Coeficientes")
```



Escalado: Se realiza nuevamente el escalado de las variables a través del parámetro standardize = TRUE.

Selección del mejor Lambda para Lasso:

Al igual que en el caso de Ridge, se realizó una validación cruzada para determinar el mejor valor de lambda para el modelo Lasso. El gráfico que relaciona lambda con el Error Cuadrático Medio (ECM) muestra cómo el error varía a medida que se ajusta el valor de lambda.

Mejor Lambda para Lasso: 1e+11

Este valor de lambda es significativamente grande, lo que indica una fuerte penalización en los coeficientes. En Lasso, esto puede resultar en que algunos coeficientes se reduzcan exactamente a cero, eliminando variables del modelo. Este es un comportamiento distintivo de Lasso, que puede ayudar a simplificar el modelo eliminando variables irrelevantes.

Gráfico de Error Cuadrático Medio (ECM) vs Lambda:

El gráfico muestra una línea casi plana, con el mínimo ECM identificado por el punto rojo. La falta de una caída pronunciada en el ECM podría sugerir que los datos no están muy afectados por diferentes niveles de regularización, o que las variables relevantes son pocas y ya están bien ajustadas con menos regularización.

Gráfico de Coeficientes vs Log(Lambda):

En el gráfico de coeficientes contra log(lambda), se observa cómo los coeficientes de las variables predictoras se ajustan con la penalización de Lasso:

A medida que lambda aumenta, varios coeficientes comienzan a reducirse a cero. Este es el efecto de selección de variables propio de Lasso.

Puntos Importantes:

- Lasso tiende a hacer un modelo más parsimonioso (más simple), eliminando variables que no son relevantes para predecir la variable objetivo (satisfaction_level).
- A diferencia de Ridge, donde los coeficientes se reducen pero no llegan a cero, Lasso establece explícitamente algunos coeficientes en cero, lo que facilita la interpretación del modelo al indicar cuáles variables son realmente importantes.

```r
# Elastic Net
alpha_for <- seq(0, 1, length = 200)
lambda_for <- 0 * seq(0, 1, length = 200)
ecm_for <- 0 * seq(0, 1, length = 200)

for(i in 1:200){
  cv_for <- cv.glmnet(x, y, alpha = alpha_for[i], lambda = lambda)
  lambda_for[i] <- cv_for$lambda.min
  ecm_for[i] <- min(cv_for$cvm)
}

min(ecm_for)
```

```
## [1] 0.05869821
```

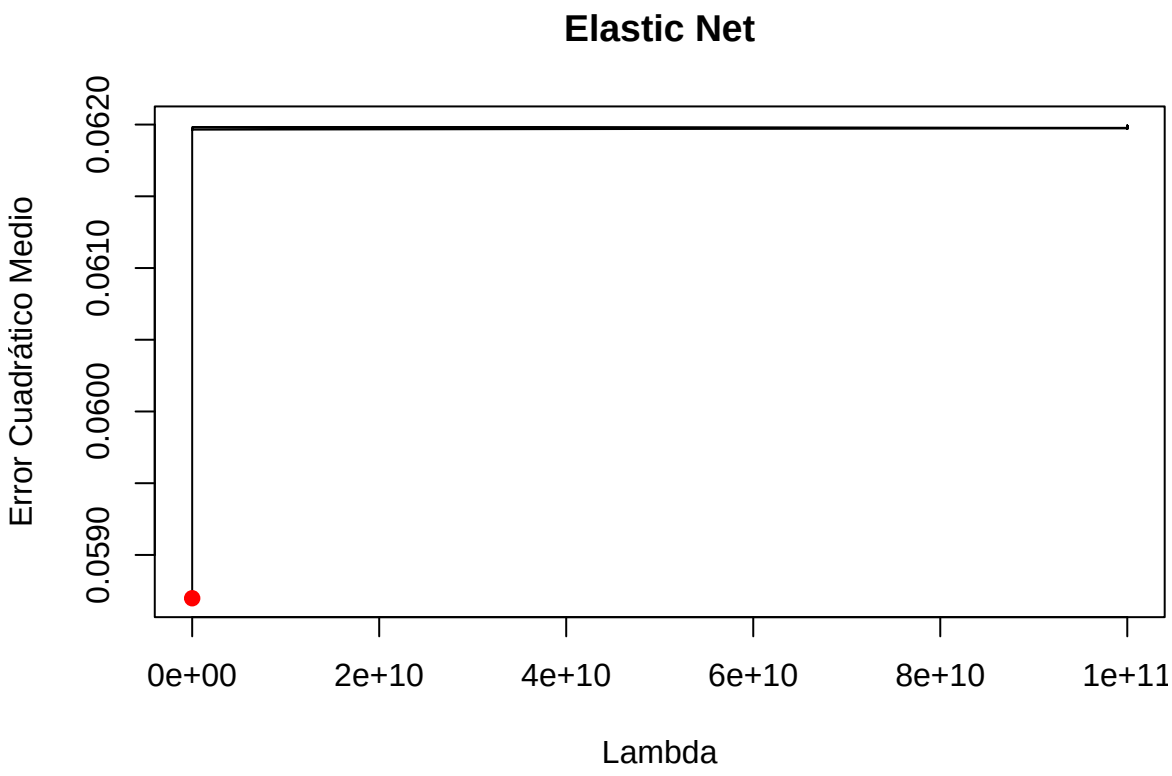
```r
diMin <- which(ecm_for == min(ecm_for))
alpha_en <- alpha_for[diMin]
lambda_en <- lambda_for[diMin]

cat("Lambda óptimo para Elastic Net:", lambda_en, "\n")
```

```
## Lambda óptimo para Elastic Net: 0.1
```

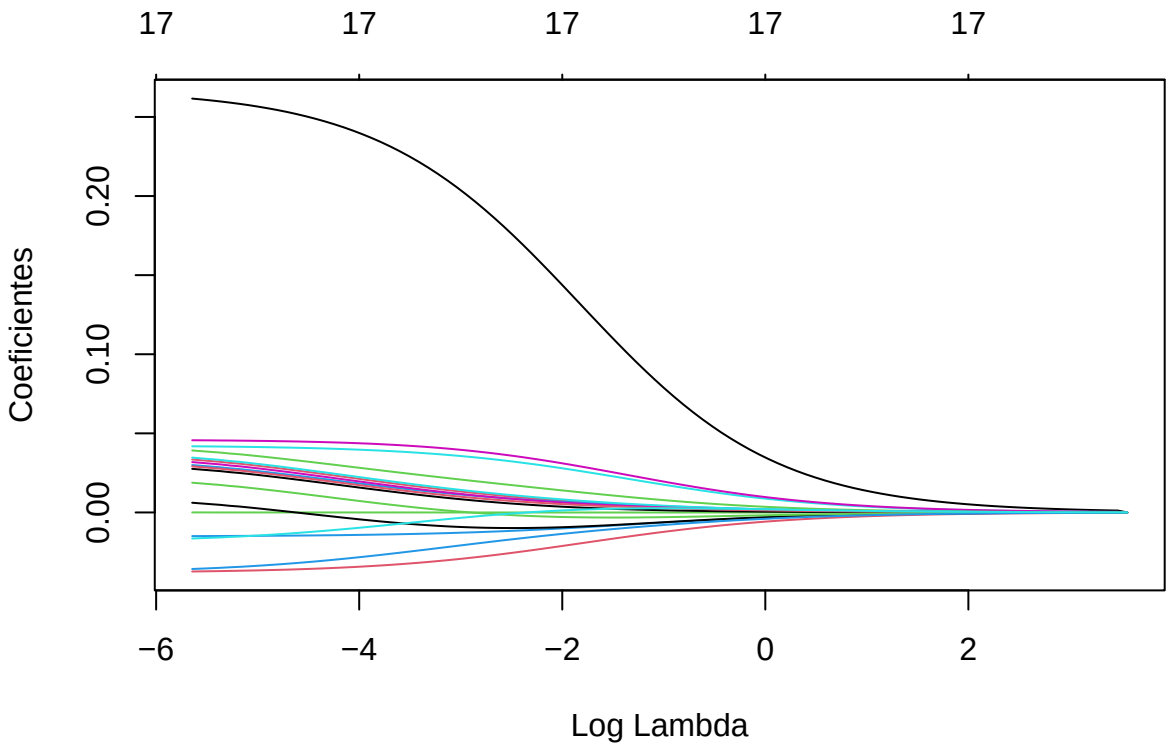
```r
cat("Alpha óptimo para Elastic Net:", alpha_en, "\n")
```

```
## Alpha óptimo para Elastic Net: 0
```

```r
# Gráfico ECM y lambda para Elastic Net
plot(lambda_for, ecm_for, type = "l", xlab = "Lambda",
     ylab = "Error Cuadrático Medio", main = "Elastic Net")
points(lambda_en, min(ecm_for), col = "red", pch = 19)
```




```r
# Gráfico coeficientes y log(Lambda) para Elastic Net
elast.mod <- glmnet(x, y, alpha = alpha_en)
plot(elast.mod, xvar = "lambda", ylab = "Coeficientes")
```



El código de Elastic Net no incluye explícitamente la opción standardize = TRUE en cv.glmnet. Sin embargo,

por defecto, la función glmnet tiene standardize = TRUE, lo que significa que las variables predictoras se estandarizan automáticamente antes de ajustar el modelo.

Selección de los Hiperparámetros Lambda y Alpha para Elastic Net:

El modelo Elastic Net combina las propiedades de Lasso y Ridge al utilizar dos hiperparámetros: lambda, que controla la penalización general, y alpha, que ajusta la mezcla entre la penalización L1 (Lasso) y L2 (Ridge). En el código, se realiza un ciclo para encontrar la mejor combinación de alpha y lambda utilizando validación cruzada.

- **Lambda óptimo: 0.1**
- **Alpha óptimo: 0**

El valor de alpha es 0, lo que significa que el modelo Elastic Net en este caso se comporta como un modelo de Ridge (ya que cuando alpha = 0, Elastic Net se reduce a Ridge). Esto es un resultado interesante porque sugiere que, para estos datos, la regularización L2 de Ridge es más efectiva que la regularización L1 de Lasso o una combinación de ambas.

Error Cuadrático Medio (ECM):

El ECM mínimo encontrado es aproximadamente 0.0587, lo cual es relativamente bajo considerando que la variable objetivo (satisfaction_level) está en una escala de 0 a 1.

Gráfico de Error Cuadrático Medio (ECM) vs Lambda:

El gráfico muestra el Error Cuadrático Medio (ECM) en función de lambda. El punto rojo marca el valor óptimo de lambda que minimiza el ECM. Como en los casos anteriores, el ECM se estabiliza para valores grandes de lambda, indicando que una mayor regularización no mejora significativamente el modelo.

Gráfico de Coeficientes vs Log(Lambda):

- El gráfico de coeficientes contra log(lambda) muestra cómo los coeficientes de las variables predictoras cambian a medida que se ajusta lambda. Dado que alpha = 0, este gráfico es muy similar al del modelo Ridge:
- Los coeficientes se reducen gradualmente a medida que lambda aumenta, pero ninguno de ellos llega a ser exactamente cero, lo que es consistente con el comportamiento de Ridge.

Predigo los resultados en el conjunto de prueba para cada modelo

```
# Ridge
ridge.mod <- glmnet(x, y, alpha = 0, lambda = bestlam_ridge, thresh = 1e-12)
ridge.pred <- predict(ridge.mod, s = bestlam_ridge, newx = x_test)

# Lasso
lasso.mod <- glmnet(x, y, alpha = 1, lambda = bestlam_lasso, thresh = 1e-12)
lasso.pred <- predict(lasso.mod, s = bestlam_lasso, newx = x_test)

# Elastic Net
elast.mod <- glmnet(x, y, alpha = alpha_en, lambda = lambda_en, thresh = 1e-12)
elast.pred <- predict(elast.mod, s = lambda_en, newx = x_test)
```

Registro el RMSE de cada modelo

```r
# Ridge
RMSE_ridge <- sqrt(mean((ridge.pred - y_test)^2))

# Lasso
RMSE_lasso <- sqrt(mean((lasso.pred - y_test)^2))

# Elastic Net
RMSE_elast <- sqrt(mean((elast.pred - y_test)^2))

# Resultados
cat("RMSE Ridge:", RMSE_ridge, "\n")
```

```
## RMSE Ridge: 0.240417
```

```r
cat("RMSE Lasso:", RMSE_lasso, "\n")
```

```
## RMSE Lasso: 0.247371
```

```r
cat("RMSE Elastic Net:", RMSE_elast, "\n")
```

```
## RMSE Elastic Net: 0.240417
```

```r
# Comparación de coeficientes
coef_comparador <- data.frame(
  "Ridge" = predict(ridge.mod, type = "coefficients", s = bestlam_ridge)[,1],
  "LASSO" = predict(lasso.mod, type = "coefficients", s = bestlam_lasso)[,1],
  "ElasticNet" = predict(elast.mod, type = "coefficients", s = lambda_en)[,1]
)
print(coef_comparador)
```

```
##                              Ridge     LASSO    ElasticNet
## (Intercept)           6.286628e-01 0.6125792  6.286628e-01
## last_evaluation       1.637984e-01 0.0000000  1.637984e-01
## number_project       -2.390886e-02 0.0000000 -2.390886e-02
## average_montly_hours -2.773025e-05 0.0000000 -2.773025e-05
## time_spend_company   -1.107308e-02 0.0000000 -1.107308e-02
## Work_accident         3.066552e-02 0.0000000  3.066552e-02
## promotion_last_5years 3.411133e-02 0.0000000  3.411133e-02
## depthr               -9.761655e-03 0.0000000 -9.761655e-03
## deptIT                8.957965e-03 0.0000000  8.957965e-03
## deptmanagement        1.597826e-02 0.0000000  1.597826e-02
## deptmarketing         7.906217e-03 0.0000000  7.906217e-03
## deptproduct_mng       9.837969e-03 0.0000000  9.837969e-03
## deptRandD             7.656843e-03 0.0000000  7.656843e-03
## deptsales             4.769456e-03 0.0000000  4.769456e-03
## deptsupport           6.411058e-03 0.0000000  6.411058e-03
## depttechnical        -2.252988e-03 0.0000000 -2.252988e-03
## salarylow            -1.561683e-02 0.0000000 -1.561683e-02
## salarymedium          1.734382e-04 0.0000000  1.734382e-04
```

Item 10

10. Comente y compare los resultados de las estimaciones e indique cuál presenta un mejor desempeño. Explique.

El RMSE se calculó para cada uno de los tres modelos (Ridge, Lasso y Elastic Net) en el conjunto de prueba. Los resultados son los siguientes:

- RMSE Ridge: 0.240417
- RMSE Lasso: 0.247371
- RMSE Elastic Net: 0.240417

Análisis del RMSE

Ridge y Elastic Net tienen exactamente el mismo RMSE, lo que sugiere que ambos modelos ofrecen un desempeño muy similar en términos de ajuste a los datos de prueba. Lasso tiene un RMSE ligeramente mayor, lo que indica que su desempeño predictivo es un poco inferior al de Ridge y Elastic Net en este conjunto de datos.

Comparación de Coeficientes

Ahora respecto a los coeficientes estimados por cada modelo. Los coeficientes indican la magnitud y dirección de la relación entre cada predictor y la variable objetivo.

Ridge:

No elimina ninguno de los predictores; todos los coeficientes tienen valores distintos de cero, aunque algunos son muy pequeños. Proporciona un ajuste equilibrado, sin reducir a cero los coeficientes, lo que puede ser útil cuando se espera que todas las variables tengan algún impacto en la predicción.

Lasso:

Ha eliminado efectivamente todas las variables del modelo excepto el intercepto. Este resultado sugiere un sobreajuste extremo y no proporciona información útil sobre los factores que influyen en la satisfacción laboral.

Elastic Net:

Se comporta de manera idéntica a Ridge en este caso, con todos los coeficientes conservados pero con regularización aplicada para evitar que los coeficientes sean demasiado grandes. Dado que alpha = 0 en este caso, Elastic Net actúa exactamente como Ridge, lo que explica por qué los coeficientes son iguales a los de Ridge.

Mejor desempeño:

Los modelos Ridge y Elastic Net muestran el mejor desempeño en términos de RMSE y ofrecen una interpretación más detallada de los factores que influyen en la satisfacción laboral.

Razones:

- Menor RMSE: Indican una mejor capacidad predictiva en el conjunto de prueba.
- Retención de variables: Mantienen todas las variables en el modelo, lo que permite una interpretación más completa de los factores que afectan la satisfacción laboral.
- Penalización equilibrada: La regularización L2 (Ridge) parece ser más apropiada para este conjunto de datos, reduciendo la magnitud de los coeficientes sin eliminarlos por completo.

El modelo Lasso, a pesar de su capacidad teórica para selección de variables, ha resultado en una sobre-regularización que elimina toda la información útil del modelo.

Conclusión

Para este problema de predicción de la satisfacción laboral, los modelos Ridge y Elastic Net (que convergió a Ridge) presentan el mejor desempeño. Ofrecen un buen equilibrio entre capacidad predictiva y retención de información sobre la importancia relativa de las variables.

La evaluación del desempeño (last_evaluation) es el factor más influyente en la satisfacción laboral, seguido por las promociones recientes. Factores como el número de proyectos y el tiempo en la empresa tienen un impacto negativo, lo que podría indicar posibles áreas de mejora en la gestión de recursos humanos.

PARTE 2: Aprendizaje Supervisado - Clasificación

Modelos de clasificación, determinación de clases estimadas y evaluación de resultados.

Conjunto de datos: "METROPOLITANA_2016.csv".

La Encuesta Mundial sobre Tabaco en Jóvenes (GYTS, por sus siglas en inglés) es un estándar global para el monitoreo sistemático del consumo de tabaco (fumado y sin humo) en los jóvenes y de los indicadores clave de control del tabaco. La GYTS forma parte del Sistema Mundial de Vigilancia del Tabaquismo (GTSS, por su sigla en inglés), el mayor sistema mundial de vigilancia de la salud pública jamás desarrollado y mantenido. El archivo "METROPOLITANA_2016.csv" contiene información sobre el hábito de fumar de casi 2.800 jóvenes de la Región Metropolitana.

```
# Elimino todas las variables en el entorno de trabajo
rm(list = ls())
```

```
# cargo librerias
library(readr)
library(MASS)
library(ggplot2)
library(dplyr)
library(VIM)
library(mice)
library(caret)
library(naivebayes)
library(rsample)
library(pROC)
library(glmnet)
```

Item 1

1. Importe la base de datos y guárdela en un data frame llamado "gytsAux".

```
gytsAux <- as.data.frame(read.csv("METROPOLITANA_2016.csv"))
```

Item 2

2. Recuerde que el tipo de datos de las variables al realizar la importación podría no coincidir con la definición real de los tipos de datos. Para determinar los tipos de datos reales debe analizar la información complementaria sobre la encuesta. Una vez que haya corregido los tipos de datos (si es que es necesario), describa estadísticamente y de forma concisa cada una de las variables que componen el data frame, indicando también si corresponden a variables numéricas o categóricas. Mantenga solamente las 8 variables de mayor interés para usted justificadamente. En su justificación mencione aspectos teóricos y técnicos. Mantenga estas 8 variables en un data frame nuevo llamado "gyts".

De acuerdo a la información contenida en el pdf `"GYTSPAHO2016 Chile All Schools Region 4 (Metropolitana) Web Codebook.pdf"` procedo a convertir las columnas a sus tipos adecuados.

```
# Identifico todas las columnas excepto FinalWgt y PSU
cols_a_factor <- setdiff(names(gytsAux), c("FinalWgt", "PSU"))

# Convierto todas las columnas identificadas a factor
gytsAux[cols_a_factor] <- lapply(gytsAux[cols_a_factor], as.factor)
```

```r
# Convierto FinalWgt y PSU a numérico
gytsAux$FinalWgt <- as.numeric(gytsAux$FinalWgt)
gytsAux$PSU <- as.numeric(gytsAux$PSU)
```

Compruebo que todas las columnas estan consideradas en el dataframe con el tipo correcto de dato:

```r
str(gytsAux)
```

```
## 'data.frame':    2778 obs. of  75 variables:
##  $ FinalWgt: num  177 177 177 177 177 ...
##  $ CR1     : Factor w/ 7 levels "1","2","3","4",..: 5 5 5 5 5 5 5 6 6 6 ...
##  $ CR2     : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR3    : Factor w/ 6 levels "1","2","3","4",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ CLR4    : Factor w/ 7 levels "1","2","3","4",..: 1 1 1 2 4 4 4 3 3 3 ...
##  $ CR5     : Factor w/ 2 levels "1","2": 1 2 2 2 1 1 1 1 1 2 ...
##  $ CR6     : Factor w/ 7 levels "1","2","3","4",..: 6 1 1 1 4 4 6 3 5 1 ...
##  $ CR7     : Factor w/ 7 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR8     : Factor w/ 7 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR9     : Factor w/ 2 levels "1","2": 2 2 2 2 1 2 2 2 1 2 ...
##  $ CR10    : Factor w/ 2 levels "1","2": 2 2 2 2 1 2 2 2 2 2 ...
##  $ CR11    : Factor w/ 4 levels "1","2","3","4": 2 1 1 1 1 2 2 2 2 1 ...
##  $ CR12    : Factor w/ 8 levels "1","2","3","4",..: 2 1 1 1 1 2 1 2 1 1 ...
##  $ CR13    : Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 NA 2 2 ...
##  $ CR14    : Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 NA 2 2 ...
##  $ OR9     : Factor w/ 8 levels "1","2","3","4",..: 1 1 1 1 1 6 1 5 1 1 ...
##  $ CLR16   : Factor w/ 5 levels "1","2","3","4",..: 5 5 5 5 5 1 5 2 5 5 ...
##  $ CLR17   : Factor w/ 4 levels "1","2","3","4": 2 1 1 1 1 3 1 2 3 1 ...
##  $ ELR2    : Factor w/ 7 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR15    : Factor w/ 4 levels "1","2","3","4": 2 1 1 1 1 2 1 2 2 1 ...
##  $ CR16    : Factor w/ 4 levels "1","2","3","4": 2 1 1 1 1 3 1 2 2 1 ...
##  $ OR12    : Factor w/ 7 levels "1","2","3","4",..: 5 1 1 1 1 2 1 7 6 1 ...
##  $ OR13    : Factor w/ 7 levels "1","2","3","4",..: 3 1 1 1 1 3 1 3 3 1 ...
##  $ CR17    : Factor w/ 4 levels "1","2","3","4": 2 1 1 1 1 2 1 2 2 1 ...
##  $ CR18    : Factor w/ 6 levels "1","2","3","4",..: 6 1 1 1 1 1 1 6 6 1 ...
##  $ CR19    : Factor w/ 5 levels "1","2","3","4",..: 3 1 4 1 1 2 5 3 5 1 ...
##  $ CR20    : Factor w/ 5 levels "1","2","3","4",..: 1 1 1 4 1 1 1 1 4 1 ...
##  $ CLR27   : Factor w/ 5 levels "1","2","3","4",..: 2 4 2 2 3 3 3 2 3 4 ...
##  $ CLR28   : Factor w/ 2 levels "1","2": 1 2 2 2 2 1 2 2 1 2 ...
##  $ CR21    : Factor w/ 5 levels "1","2","3","4",..: 3 2 4 4 1 1 1 2 5 2 ...
##  $ CR22    : Factor w/ 2 levels "1","2": 1 1 2 1 1 2 1 1 2 2 ...
##  $ CR23    : Factor w/ 4 levels "1","2","3","4": 3 3 3 4 3 3 4 4 3 3 ...
##  $ CR24    : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR33   : Factor w/ 3 levels "1","2","3": 3 2 1 3 2 2 2 2 1 2 ...
##  $ CLR34   : Factor w/ 2 levels "1","2": 2 1 2 1 1 2 2 2 2 2 ...
##  $ CR25    : Factor w/ 2 levels "1","2": 2 1 2 1 2 2 1 2 2 1 ...
##  $ CLR36   : Factor w/ 6 levels "1","2","3","4",..: 1 1 1 1 1 4 1 1 1 1 ...
##  $ CR27    : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR38   : Factor w/ 5 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR39   : Factor w/ 6 levels "1","2","3","4",..: 6 1 4 1 2 5 6 1 3 6 ...
##  $ CLR40   : Factor w/ 3 levels "1","2","3": 3 1 3 3 1 3 2 1 3 3 ...
##  $ CLR41   : Factor w/ 3 levels "1","2","3": 1 3 3 1 1 NA 3 1 3 3 ...
##  $ CLR42   : Factor w/ 8 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR30    : Factor w/ 2 levels "1","2": 1 2 1 2 1 2 2 2 2 1 ...
```

```
##  $ CR31    : Factor w/ 3 levels "1","2","3": 2 1 1 3 1 3 1 1 2 2 ...
##  $ CR32    : Factor w/ 3 levels "1","2","3": 2 2 1 3 1 3 2 3 2 3 ...
##  $ CLR46   : Factor w/ 3 levels "1","2","3": 3 2 3 3 3 2 2 2 1 1 ...
##  $ CLR47   : Factor w/ 3 levels "1","2","3": 3 3 3 3 3 2 2 2 3 3 ...
##  $ CLR48   : Factor w/ 3 levels "1","2","3": 3 2 3 3 3 1 1 2 1 3 ...
##  $ CLR49   : Factor w/ 6 levels "1","2","3","4",..: 3 3 2 2 2 1 3 3 2 1 ...
##  $ CR34    : Factor w/ 3 levels "1","2","3": 3 3 2 3 2 3 3 3 3 2 ...
##  $ CR35    : Factor w/ 3 levels "1","2","3": 2 3 2 3 3 3 3 3 3 3 ...
##  $ CR36    : Factor w/ 3 levels "1","2","3": 3 3 2 3 2 2 3 3 2 3 ...
##  $ CR37    : Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 2 2 2 ...
##  $ CR38    : Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 2 2 2 ...
##  $ OR45    : Factor w/ 5 levels "1","2","3","4",..: 4 1 2 2 1 1 3 2 1 4 ...
##  $ CR39    : Factor w/ 4 levels "1","2","3","4": 1 1 2 1 2 2 1 1 2 2 ...
##  $ CR40    : Factor w/ 4 levels "1","2","3","4": 1 1 2 1 2 1 1 1 1 2 ...
##  $ CR41    : Factor w/ 4 levels "1","2","3","4": 3 3 3 1 4 4 2 1 3 3 ...
##  $ CR42    : Factor w/ 3 levels "1","2","3": 3 3 1 3 1 3 3 1 2 3 ...
##  $ CR43    : Factor w/ 5 levels "1","2","3","4",..: 4 4 3 3 4 5 5 4 1 4 ...
##  $ OR55    : Factor w/ 2 levels "1","2": 2 1 1 1 1 1 2 1 1 1 ...
##  $ CLR62   : Factor w/ 4 levels "1","2","3","4": 4 1 1 1 2 4 2 3 1 1 ...
##  $ CLR63   : Factor w/ 5 levels "1","2","3","4",..: 4 1 1 1 4 5 1 4 1 1 ...
##  $ CLR64   : Factor w/ 4 levels "1","2","3","4": 1 1 2 1 1 2 1 3 1 2 ...
##  $ CLR65   : Factor w/ 3 levels "1","2","3": 3 2 3 3 2 1 2 1 2 3 ...
##  $ OR46    : Factor w/ 4 levels "1","2","3","4": 2 1 2 3 2 2 2 1 2 3 ...
##  $ OR54    : Factor w/ 4 levels "1","2","3","4": 1 1 2 1 1 1 1 1 1 1 ...
##  $ CLR68   : Factor w/ 7 levels "1","2","3","4",..: 7 1 7 7 7 7 2 7 2 7 ...
##  $ CLR69   : Factor w/ 7 levels "1","2","3","4",..: 7 7 7 7 7 7 2 7 2 7 ...
##  $ CLR70   : Factor w/ 3 levels "1","2","3": 2 2 3 2 2 3 1 2 2 2 ...
##  $ CLR71   : Factor w/ 3 levels "1","2","3": 3 3 3 2 3 3 3 2 3 2 ...
##  $ CLR72   : Factor w/ 3 levels "1","2","3": 3 3 3 2 3 3 3 3 3 2 ...
##  $ Stratum : Factor w/ 12 levels "201604001","201604002",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PSU     : num  1 1 1 1 1 1 1 1 1 1 ...
```

Justificacion Técnica de la elección de las 8 variables

```r
# Elimino filas con valores faltantes
gytsAux_sin_na <- na.omit(gytsAux)

# Ajusto un modelo inicial
modelo_inicial <- lm(FinalWgt ~ ., data = gytsAux_sin_na)
```

```r
# Capturo el output del proceso de stepwise en una variable, pero sin imprimirlo
# demora alrededor de 2 minutos
step_output <- capture.output({
  modelo_step_AIC <- step(modelo_inicial, direction = "both", k = 2)
})

# Imprimo solo la últimas 81 lineas del proceso
# que corresponden a la última iteración
# que es la que nos interesa
cat(tail(step_output, 81), sep = "\n")
```

```
## Step:  AIC=12970.79
## FinalWgt ~ CR1 + CR2 + CLR3 + CR10 + OR9 + CLR27 + CR22 + CR23 +
```

```
##     CR25 + CLR38 + CLR40 + CLR42 + CLR49 + CR37 + OR55 + OR46 +
##     Stratum + PSU
##
##           Df Sum of Sq     RSS   AIC
## <none>                  1597239 12971
## + CR34     2      3339 1593900 12971
## + CR17     3      4972 1592267 12971
## + CR42     2      2903 1594336 12971
## - CLR49    5      8863 1606102 12971
## + CR38     1       748 1596491 12972
## + CLR34    1       722 1596517 12972
## + CR30     1       704 1596535 12972
## + CR41     3      3947 1593292 12972
## + CR5      1       572 1596667 12972
## + CLR28    1       507 1596732 12972
## + CLR47    2      2166 1595073 12972
## + CR32     2      2029 1595210 12972
## + CR13     1       183 1597056 12973
## + CR24     1       126 1597113 12973
## + CR9      1        92 1597147 12973
## + CR14     1        15 1597224 12973
## + OR45     4      5022 1592217 12973
## + CLR16    4      4931 1592308 12973
## + CLR39    5      6598 1590641 12973
## + CLR72    2      1416 1595823 12973
## + CR12     7      9676 1587563 12973
## + CR16     3      2999 1594240 12973
## + CLR36    5      6305 1590934 12973
## - CR37     1      3751 1600990 12973
## + CR31     2      1164 1596075 12973
## - CR25     1      3891 1601130 12973
## + CR11     3      2791 1594448 12974
## - OR55     1      4004 1601243 12974
## + CLR70    2      1021 1596218 12974
## + CLR33    2       823 1596416 12974
## + CLR65    2       500 1596739 12974
## + CLR64    3      2064 1595175 12974
## + CLR46    2       378 1596861 12974
## + CLR41    2       365 1596874 12974
## + CR35     2       303 1596937 12974
## + CLR71    2       185 1597055 12975
## + CR39     3      1846 1595393 12975
## + CLR48    2       122 1597117 12975
## + CR36     2       109 1597130 12975
## + CR27     2        50 1597189 12975
## + CR18     5      4971 1592268 12975
## - CLR40    2      6830 1604069 12975
## + CLR17    3      1517 1595722 12975
## - CR23     3      8637 1605877 12975
## - CR10     1      5580 1602819 12975
## + CR40     3       987 1596252 12976
## + CR43     4      2571 1594668 12976
## + CR15     3       638 1596601 12976
## + CLR63    4      2244 1594995 12976
```

```
## + OR54     3       387 1596852 12976
## - CLR27    4     11389 1608629 12976
## + OR12     6      5220 1592019 12976
## + CLR62    3        64 1597175 12977
## - CR1      6     15190 1612429 12977
## + CR20     4      1536 1595703 12977
## + OR13     6      4875 1592364 12977
## + CLR4     6      4839 1592400 12977
## - OR9      7     17207 1614446 12977
## + CR7      6      4557 1592682 12977
## - CLR38    4     12287 1609526 12977
## + CR19     4       911 1596328 12978
## + CLR68    6      3887 1593352 12978
## + CR21     4       367 1596872 12978
## + CLR69    6      3243 1593996 12979
## + ELR2     6      3112 1594127 12979
## - OR46     3     12092 1609331 12979
## + CR8      6      2896 1594343 12979
## + CR6      6      1922 1595317 12980
## - CR22     1     17666 1614905 12990
## - CLR42    7     34292 1631531 12997
## - PSU      1     74041 1671280 13055
## - CLR3     5    217702 1814941 13205
## - CR2      1    229448 1826687 13225
## - Stratum 11    713364 2310603 13653
```

```r
# Capturo el output del proceso de stepwise con BIC en una variable,
# pero solo guardandolo
# demora alrededor de 2 minutos
step_output_BIC <- capture.output({
  modelo_step_BIC <- step(modelo_inicial, direction = "both",
                          k = log(nrow(gytsAux)))
})

# Imprimo solo la últimas 79 lineas del proceso
# que corresponden a la última iteración
# que es la que nos interesa
cat(tail(step_output_BIC, 79), sep = "\n")
```

```
## Step:  AIC=13148.82
## FinalWgt ~ CR2 + CLR3 + CR22 + Stratum + PSU
##
##           Df Sum of Sq     RSS   AIC
## <none>                 1727363 13149
## + CR25     1      3189 1724174 13153
## + CR10     1      3179 1724184 13153
## + OR55     1      2857 1724506 13154
## + CR37     1      2019 1725344 13154
## + CR38     1      1516 1725848 13155
## + CLR40    2      8362 1719001 13155
## + CR9      1       980 1726383 13156
## + CR5      1       588 1726775 13156
## + CLR28    1       217 1727146 13156
## + CR30     1       164 1727199 13157
```

```
## + CR24     1       109 1727254 13157
## + CLR34    1       108 1727256 13157
## + CR13     1         2 1727361 13157
## + CR14     1         0 1727363 13157
## + OR46     3     12436 1714927 13159
## + CR34     2      4524 1722839 13160
## + CLR47    2      3496 1723867 13161
## + CR42     2      2602 1724761 13162
## + CLR70    2      2371 1724992 13162
## + CR23     3      9044 1718319 13163
## + CR31     2      1642 1725721 13163
## + CR32     2      1318 1726045 13163
## + CLR48    2      1314 1726049 13163
## + CLR33    2       882 1726482 13164
## + CR36     2       753 1726610 13164
## + CLR41    2       608 1726755 13164
## + CLR65    2       514 1726849 13164
## + CR35     2       311 1727052 13164
## + CLR72    2       221 1727142 13164
## + CR27     2       145 1727218 13164
## + CLR46    2       101 1727262 13165
## + CLR71    2        18 1727345 13165
## - CR22     1     22880 1750243 13166
## + CR41     3      5128 1722235 13167
## + CLR27    4     11951 1715412 13167
## + CR17     3      4365 1722998 13168
## + CR15     3      3354 1724009 13169
## + CR39     3      2749 1724614 13170
## + CLR64    3      2349 1725014 13170
## + CR11     3      1753 1725610 13171
## + CR16     3      1455 1725908 13171
## + CR40     3      1434 1725929 13171
## + CLR17    3      1016 1726347 13172
## + CLR62    3       864 1726499 13172
## + OR54     3       135 1727228 13172
## + OR45     4      6927 1720436 13173
## + CLR49    5     13061 1714302 13174
## + CLR63    4      4169 1723194 13176
## + CLR16    4      3529 1723834 13177
## + CR43     4      2190 1725173 13178
## + CLR38    4      1516 1725847 13179
## + CR1      6     15403 1711960 13179
## + CR21     4       648 1726715 13180
## + CLR39    5      7734 1719629 13180
## + CR20     4       430 1726933 13180
## + CR19     4       268 1727095 13180
## + CLR36    5      5045 1722318 13183
## + CR18     5      3637 1723726 13184
## + OR12     6      9049 1718314 13186
## + CR12     7     16038 1711325 13186
## + CLR42    7     15463 1711900 13187
## + CR7      6      8166 1719197 13187
## + CLR68    6      5739 1721624 13190
## + OR13     6      4890 1722474 13191
```

```
## + CR8      6      4712 1722651 13191
## + CLR4     6      4207 1723157 13192
## + CLR69    6      2569 1724794 13194
## + OR9      7      9248 1718116 13194
## + ELR2     6      2091 1725272 13194
## + CR6      6      1315 1726048 13195
## - PSU      1     82976 1810339 13230
## - CR2      1    240560 1967923 13390
## - CLR3     5    626589 2353952 13700
## - Stratum 11    823917 2551280 13806
```

Nota importante: aunque se muestra como "AIC", el valor reportado es realmente el BIC, ya que estoy utilizando `k = log(nrow(gytsAux))` para penalizar los modelos. Y esto es porque la función `step()` en R está diseñada para mostrar el AIC en su salida, incluso cuando estoy utilizando BIC como criterio de selección.

Análisis del Stepwise Regression con AIC

En el último paso de la selección con AIC, las siguientes variables fueron retenidas: - CR1, CR2, CLR3, CR10, OR9, CLR27, CR22, CR23, CR25, CLR38, CLR40, CLR42, CLR49, CR37, OR55, OR46, Stratum y PSU.

Análisis del Stepwise Regression con BIC

En el último paso de la selección con BIC, las siguientes variables fueron retenidas: - CR2, CLR3, CR22 y PSU.

Justificación de la Selección de Variables mediante criterio tecnico

- AIC vs. BIC: El criterio AIC tiende a seleccionar un mayor número de variables porque penaliza menos la complejidad del modelo. BIC, en cambio, penaliza más la complejidad y selecciona un modelo más parsimonioso (con menos variables).
- Variables Comunes: Al observar las variables seleccionadas por ambos criterios, noto que CR2, CLR3, CR22 y PSU son comunes en ambos. Por ende, estas variables son altamente relevantes para el modelo y por eso son parte en la selección final, con la excepcion de PSU que no está definida que significa en el PDF `"GYTSPAHO2016 Chile All Schools Region 4 (Metropolitana) Web Codebook.pdf"` ni en la pagina web https://extranet.who.int/ncdsmicrodata/index.php/catalog/387/data-dictionary/F3?file_name=METROPOLITANA_2016

Hasta ahora, he seleccionado las variables explicativas **CR2, CLR3 y CR22** mediante análisis técnico. Sin embargo, me faltan elegir 5 variables adicionales. Dado que en la siguiente etapa de Machine Learning se indica que CR7 será la variable objetivo ("Durante los últimos 30 días, ¿en cuántos días fumaste cigarrillos?"), analizaré cuidadosamente cuáles serán las otras cuatro variables que elegiré, considerando su potencial para predecir CR7 de manera efectiva.

Justificación analisis teorico

CLR41 - Is it possible for you to buy individual sticks where you live?:

- La disponibilidad de cigarrillos en pequeñas cantidades, como la compra de cigarrillos individuales, reduce la barrera económica de acceso al tabaco y facilita el inicio o la continuación del hábito de fumar. Esto es especialmente relevante para jóvenes o personas con menor poder adquisitivo, quienes pueden encontrar más fácil experimentar con el tabaco o mantener el hábito al poder comprar cigarrillos en menor cantidad. Por lo tanto, la accesibilidad a cigarrillos individuales es un factor clave que puede influir en la prevalencia del tabaquismo en la comunidad, haciendo que esta variable sea un predictor significativo en el modelo.

CR19 - Smoked inside home in your presence:

- Considero que la exposición al humo de segunda mano en casa es un predictor significativo de la aceptación y normalización del hábito de fumar. Esto indica que la exposición al humo en casa puede influir en la percepción y la aceptación del comportamiento de fumar, lo que hace que esta variable sea un componente crucial en el modelo.

CR17 - Stop smoking if wanted to:

- La percepción de la capacidad para dejar de fumar refleja la motivación y la autoeficacia del individuo para abandonar el hábito.
- La capacidad percibida para dejar de fumar (capturada por CR17) está estrechamente relacionada con estas motivaciones internas y, por lo tanto, es un predictor importante del comportamiento de fumar.

CR20 - Smoked inside public place in your presence past 7 days:

- Similar a la exposición en el hogar, la exposición al humo en lugares públicos refleja la normalización social del fumar, lo que puede influir en el comportamiento del encuestado. Osea que la exposición al humo en lugares públicos puede influir en la percepción y aceptación social del fumar, haciendo de esta variable un importante predictor.

OR46 - Closest friends smoke:

- Se justifica por el impacto significativo que las influencias sociales, especialmente las de los amigos cercanos, tienen en el comportamiento de fumar. Esta variable captura la dinámica del grupo social inmediato del encuestado, que es un factor crucial en la adopción y persistencia del hábito de fumar.

Por ende de acuerdo al analisis tecnico y teorico las 8 variables seleccionadas son:

CR2, CLR3, CR22, CLR41, CR19, CR17, CR20, y OR46

Y de acuerdo al correo respondido el día 17/08/2024 con copia a todo el curso por parte del profesor, procedo a describir estadisticamente y de forma concisa sólo las 8 variables que justificadamente elegí.

```
# genero variable con las columnas seleccionadas
# que son todas categoricas
gyts <- as.data.frame(gytsAux[, c("CR2", "CLR3", "CR22", "CLR41", "CR19", "CR17", "CR20", "OR46")])

summary(gyts)
```

```
##    CR2          CLR3        CR22        CLR41         CR19         CR17
##  1   :1278   1   :517   1   :1509   1   :1136   1   :1653   1   :1589
##  2   :1480   2   :547   2   :1201   2   : 591   2   : 348   2   : 464
##  NA's:  20   3   :543   NA's:  68   3   : 996   3   : 201   3   : 650
##              4   :462               NA's:  55   4   : 108   4   :  42
##              5   :336                           5   : 442   NA's:  33
##              6   :358                           NA's:  26
##              NA's: 15
##    CR20         OR46
##  1   :1397   1   : 746
##  2   : 594   2   :1279
##  3   : 273   3   : 579
##  4   : 135   4   :  86
##  5   : 352   NA's:  88
##  NA's:  27
##
```

```r
# Calculo los porcentajes por clase para cada columma categórica
porcentajes_por_clase <- lapply(gyts, function(col) {
  prop.table(table(col))
})

# Muestra los porcentajes por clase
porcentajes_por_clase
```

```
## $CR2
## col
##         1         2
## 0.4633793 0.5366207
##
## $CLR3
## col
##         1         2         3         4         5         6
## 0.1871155 0.1979732 0.1965255 0.1672096 0.1216069 0.1295693
##
## $CR22
## col
##         1         2
## 0.5568266 0.4431734
##
## $CLR41
## col
##         1         2         3
## 0.4171869 0.2170400 0.3657730
##
## $CR19
## col
##          1          2          3          4          5
## 0.60065407 0.12645349 0.07303779 0.03924419 0.16061047
##
## $CR17
## col
##          1          2          3          4
## 0.57887067 0.16903461 0.23679417 0.01530055
##
## $CR20
## col
##          1          2          3          4          5
## 0.50781534 0.21592148 0.09923664 0.04907306 0.12795347
##
## $OR46
## col
##          1          2          3          4
## 0.27732342 0.47546468 0.21524164 0.03197026
```

```r
#todos los posibles valores para cada variable categórica
#en español
valores_categoria <- list(
  CR2 = c("1" = "Hombre", "2" = "Mujer"),
  CLR3 = c("1" = "7° básico",
```

```
            "2" = "8° básico",
            "3" = "1° medio",
            "4" = "2° medio",
            "5" = "3° medio",
            "6" = "4° medio"),
  CR22 = c("1" = "Sí", "2" = "No"),
  CLR41 = c("1" = "Sí", "2" = "No", "3" = "No lo sé"),
  CR19 = c("1" = "0 días", "2" = "1 a 2 días", "3" = "3 a 4 días",
           "4" = "5 a 6 días", "5" = "7 días"),
  CR17 = c("1" = "Nunca he fumado",
           "2" = "No fumo ahora",
           "3" = "Sí",
           "4" = "No"),
  CR20 = c("1" = "0 días", "2" = "1 a 2 días", "3" = "3 a 4 días",
           "4" = "5 a 6 días", "5" = "7 días"),
  OR46 = c("1" = "Ninguno de ellos",
           "2" = "Algunos de ellos",
           "3" = "La mayoría de ellos",
           "4" = "Todos ellos")
)
```

```
# Defino los significados de las variables en español
significados_variables <- list(
  CR2 = "Sexo",
  CLR3 = "Nivel de educación",
  CR22 = "Visto fumar en la escuela",
  CLR41 = "Posibilidad de comprar cigarrillos sueltos",
  CR19 = "Fumaron en casa en su presencia",
  CR17 = "Dejar de fumar si lo desea",
  CR20 = "Fumaron en lugar público en su presencia",
  OR46 = "Amigos cercanos que fuman"
)
```

```
# Función para describir estadísticamente una variable
# con su significado
describir_variable <- function(var_name) {
  var <- gytsAux[[var_name]]
  n_unique <- length(unique(var))
  moda <- names(sort(table(var), decreasing = TRUE))[1]
  moda_significado <- valores_categoria[[var_name]][moda]
  significado_var <- significados_variables[[var_name]]

  # Determino el tipo de variable
  tipo_var <- ifelse(class(var) == "factor", "categórica",
                     ifelse(class(var) == "numeric", "númerica", class(var)))

  cat(sprintf("Descripción de la variable: %s (%s)\n",
              var_name, significado_var))
  cat(sprintf("Tipo de variable: %s\n", tipo_var))
  cat(sprintf("Número de categorías únicas: %d\n", n_unique))
  # Solo muestro la moda y su significado si es categórica
  if (tipo_var == "categórica") {
    cat(sprintf("Moda (valor más frecuente): %s (%s)\n",
```

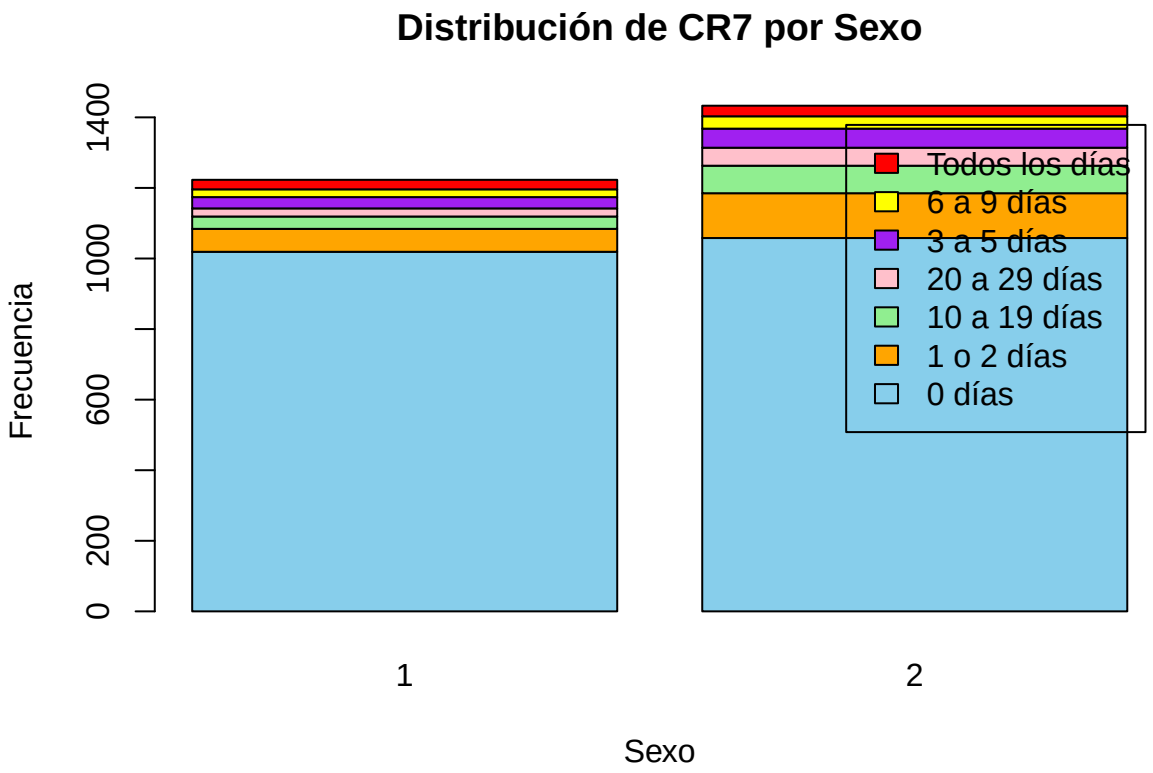
```
                moda, moda_significado))
  }
  cat("\n")  # Agrego línea en blanco para separar las descripciones
}

# Aplico la función a cada variable de interés
gyts <- c("CR2", "CLR3", "CR22", "CLR41", "CR19", "CR17", "CR20", "OR46")

for (col in gyts) {
  describir_variable(col)
}
```

```
## Descripción de la variable: CR2 (Sexo)
## Tipo de variable: categórica
## Número de categorías únicas: 3
## Moda (valor más frecuente): 2 (Mujer)
##
## Descripción de la variable: CLR3 (Nivel de educación)
## Tipo de variable: categórica
## Número de categorías únicas: 7
## Moda (valor más frecuente): 2 (8° básico)
##
## Descripción de la variable: CR22 (Visto fumar en la escuela)
## Tipo de variable: categórica
## Número de categorías únicas: 3
## Moda (valor más frecuente): 1 (Sí)
##
## Descripción de la variable: CLR41 (Posibilidad de comprar cigarrillos sueltos)
## Tipo de variable: categórica
## Número de categorías únicas: 4
## Moda (valor más frecuente): 1 (Sí)
##
## Descripción de la variable: CR19 (Fumaron en casa en su presencia)
## Tipo de variable: categórica
## Número de categorías únicas: 6
## Moda (valor más frecuente): 1 (0 días)
##
## Descripción de la variable: CR17 (Dejar de fumar si lo desea)
## Tipo de variable: categórica
## Número de categorías únicas: 5
## Moda (valor más frecuente): 1 (Nunca he fumado)
##
## Descripción de la variable: CR20 (Fumaron en lugar público en su presencia)
## Tipo de variable: categórica
## Número de categorías únicas: 6
## Moda (valor más frecuente): 1 (0 días)
##
## Descripción de la variable: OR46 (Amigos cercanos que fuman)
## Tipo de variable: categórica
## Número de categorías únicas: 5
## Moda (valor más frecuente): 2 (Algunos de ellos)
```

Item 3

3. Realice estadística descriptiva con mayor detalle principalmente sobre la variable "Q7 (CR7)…". Se espera que cruce dicha variable con otras 4-5 de interés. Incorpore análisis gráfico.

```r
# Gráfico de barras apiladas para CR7 y CR2 (Sexo)
barplot(table(gytsAux$CR7, gytsAux$CR2),
        legend.text = c("0 días", "1 o 2 días", "10 a 19 días", "20 a 29 días",
                        "3 a 5 días", "6 a 9 días", "Todos los días"),
        col = c("skyblue", "orange", "lightgreen", "pink",
                "purple", "yellow", "red"),
        main = "Distribución de CR7 por Sexo",
        xlab = "Sexo",
        ylab = "Frecuencia",
        beside = FALSE)
```



CR2 (Sexo):

- 1 = Hombre
- 2 = Mujer

CR7 (Días fumados en los últimos 30 días):

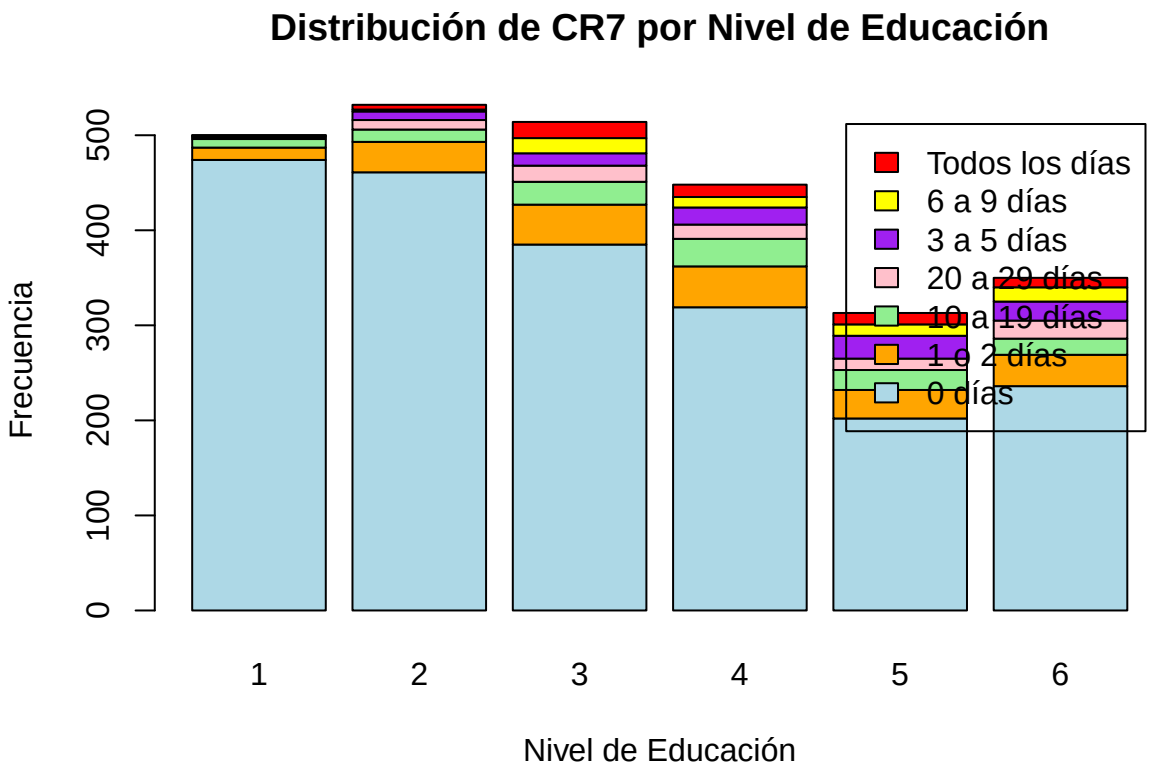
- 1 = 0 días
- 2 = 1 o 2 días
- 3 = 10 a 19 días
- 4 = 20 a 29 días
- 5 = 3 a 5 días
- 6 = 6 a 9 días

- 7 = Todos los días

La mayoría de los encuestados, tanto hombres como mujeres, no han fumado en los últimos 30 días (0 días).

Tambien se observa que tanto en hombres como en mujeres, el mayor porcentaje de individuos no ha fumado en los últimos 30 días. Sin embargo, hay una ligera tendencia a que las mujeres presenten una menor frecuencia en las categorías de consumo más alto (3 a 5 días, 6 a 9 días, todos los días) en comparación con los hombres.

```
# Gráfico de barras apiladas para CR7 y CLR3 (Nivel de educación)
barplot(table(gytsAux$CR7, gytsAux$CLR3),
        legend.text = c("0 días", "1 o 2 días", "10 a 19 días", "20 a 29 días",
                        "3 a 5 días", "6 a 9 días", "Todos los días"),
        col = c("lightblue", "orange", "lightgreen", "pink",
                "purple", "yellow", "red"),
        main = "Distribución de CR7 por Nivel de Educación",
        xlab = "Nivel de Educación",
        ylab = "Frecuencia",
        beside = FALSE)
```



CLR3 (Nivel de educación):

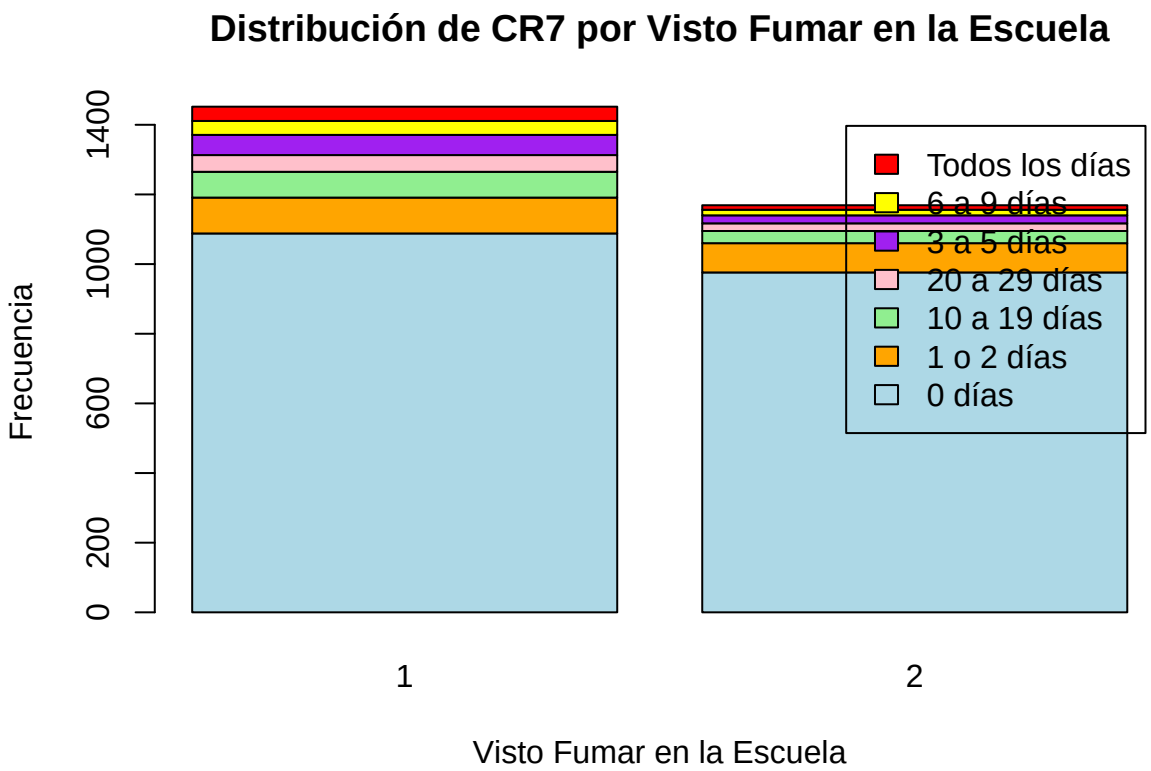
- 1 = 7° básico
- 2 = 8° básico
- 3 = 1° medio
- 4 = 2° medio
- 5 = 3° medio
- 6 = 4° medio

CR7 (Días fumados en los últimos 30 días):

- 1 = 0 días
- 2 = 1 o 2 días
- 3 = 10 a 19 días
- 4 = 20 a 29 días
- 5 = 3 a 5 días
- 6 = 6 a 9 días
- 7 = Todos los días

Se observa en el gráfico que los estudiantes de niveles más bajos de educación (7° y 8° básico) muestran un mayor porcentaje de no fumadores (0 días). Sin embargo, conforme aumenta el nivel educativo, hay una ligera disminución en la proporción de no fumadores y un aumento en aquellos que han fumado entre 1 a 9 días o más en los últimos 30 días.

```
# Gráfico de barras apiladas para CR7 y CR22 (Visto fumar en la escuela)
barplot(table(gytsAux$CR7, gytsAux$CR22),
        legend.text = c("0 días", "1 o 2 días", "10 a 19 días",
                        "20 a 29 días", "3 a 5 días", "6 a 9 días",
                        "Todos los días"),
        col = c("lightblue", "orange", "lightgreen", "pink",
                "purple", "yellow", "red"),
        main = "Distribución de CR7 por Visto Fumar en la Escuela",
        xlab = "Visto Fumar en la Escuela",
        ylab = "Frecuencia",
        beside = FALSE)
```



CR22 (Visto fumar en la escuela):

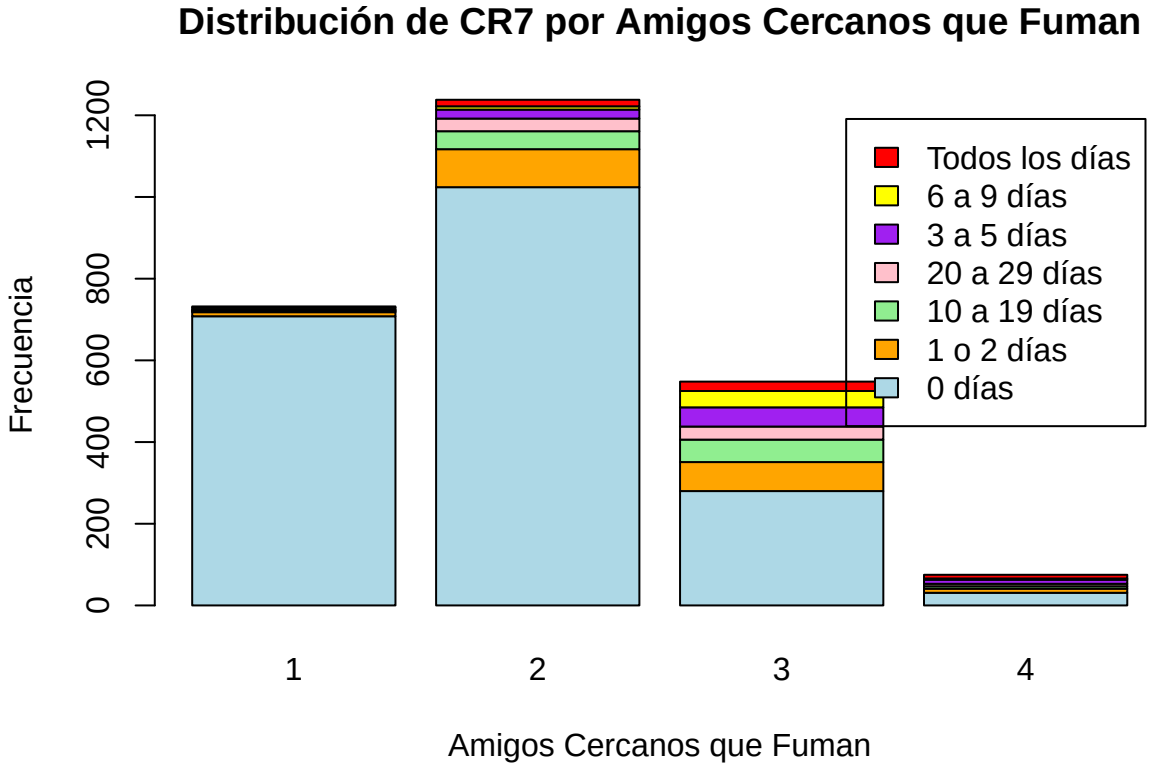
- 1 = Sí
- 2 = No

CR7 (Días fumados en los últimos 30 días):

- 1 = 0 días
- 2 = 1 o 2 días
- 3 = 10 a 19 días
- 4 = 20 a 29 días
- 5 = 3 a 5 días
- 6 = 6 a 9 días
- 7 = Todos los días

Se ve graficamente que aquellos que han visto fumar en la escuela tienen una mayor tendencia a haber fumado en los últimos 30 días, comparado con aquellos que no lo han visto. Esta relación sugiere que la exposición al comportamiento de fumar en el entorno escolar podría influir en la decisión de fumar de los estudiantes.

```r
# Gráfico de barras apiladas para CR7 y OR46 (Amigos cercanos que fuman)
barplot(table(gytsAux$CR7, gytsAux$OR46),
        legend.text = c("0 días", "1 o 2 días", "10 a 19 días",
                        "20 a 29 días", "3 a 5 días", "6 a 9 días",
                        "Todos los días"),
        col = c("lightblue", "orange", "lightgreen", "pink",
                "purple", "yellow", "red"),
        main = "Distribución de CR7 por Amigos Cercanos que Fuman",
        xlab = "Amigos Cercanos que Fuman",
        ylab = "Frecuencia",
        beside = FALSE)
```



OR46 (Amigos cercanos que fuman):

- 1 = Ninguno de ellos
- 2 = Algunos de ellos
- 3 = La mayoría de ellos
- 4 = Todos ellos

CR7 (Días fumados en los últimos 30 días):

- 1 = 0 días
- 2 = 1 o 2 días
- 3 = 10 a 19 días
- 4 = 20 a 29 días
- 5 = 3 a 5 días
- 6 = 6 a 9 días
- 7 = Todos los días

Veo en el gráfico que los estudiantes cuyos amigos cercanos no fuman (Ninguno de ellos) son en su mayoría no fumadores. Sin embargo, a medida que aumenta el número de amigos que fuman (especialmente "Todos ellos"), también aumenta la probabilidad de que el estudiante haya fumado en los últimos 30 días, destacando la fuerte influencia social en el comportamiento de fumar.

Analisis estadístico descriptivo sobre la variable "Q7 (CR7)…". Y cruces de dicha variable con otras 4-5 de interés.

```
# Defino los posibles valores para las variables CR7 y CR2
valores_CR7 <- c("1" = "0 días", "2" = "1 o 2 días", "3" = "10 a 19 días",
                 "4" = "20 a 29 días", "5" = "3 a 5 días", "6" = "6 a 9 días",
                 "7" = "Todos los días")

valores_CR2 <- c("1" = "Hombre", "2" = "Mujer")

# Creo tabla de contingencia con nombres
tabla_CR7_CR2 <- table(factor(gytsAux$CR7, labels = valores_CR7),
                       factor(gytsAux$CR2, labels = valores_CR2))
# Imprimo tabla de contingencia
print(tabla_CR7_CR2)
```

```
##
##                   Hombre Mujer
##   0 días            1019  1058
##   1 o 2 días          65   127
##   10 a 19 días        35    78
##   20 a 29 días        23    51
##   3 a 5 días          32    54
##   6 a 9 días          22    35
##   Todos los días      27    30
```

Asociación entre CR7 (Días fumados en los últimos 30 días) y CR2 (Sexo):

- La mayoría de los encuestados, tanto hombres como mujeres, no han fumado en los últimos 30 días ("0 días").
- Sin embargo, se observa una tendencia en la que hay más mujeres que hombres que reportan haber fumado "1 o 2 días" y "10 a 19 días".

- A medida que aumenta la frecuencia de fumar (más días fumados), las diferencias entre sexos se reducen.

Esto sugiere que mientras la mayoría de ambos sexos no fuman, hay una proporción ligeramente mayor de mujeres que hombres en las categorías de fumadores ocasionales. Esto podría reflejar diferencias en los patrones de iniciación o en la exposición al tabaco entre sexos.

```
valores_CLR3 <- c("1" = "7° básico", "2" = "8° básico", "3" = "1° medio",
                  "4" = "2° medio", "5" = "3° medio", "6" = "4° medio")

# Tabla de contingencia entre CR7 y CLR3 (Nivel de educación)
tabla_CR7_CLR3 <- table(factor(gytsAux$CR7, labels = valores_CR7),
                        factor(gytsAux$CLR3, labels = valores_CLR3))
print(tabla_CR7_CLR3)
```

```
##
##                  7° básico 8° básico 1° medio 2° medio 3° medio 4° medio
##   0 días               474       461       385       319       202       236
##   1 o 2 días            13        32        42        43        30        33
##   10 a 19 días           9        13        24        29        21        17
##   20 a 29 días           2        10        17        15        12        19
##   3 a 5 días             1         9        13        18        24        20
##   6 a 9 días             1         2        16        11        12        15
##   Todos los días         0         5        17        13        12        10
```

Asociación entre CR7 (Días fumados en los últimos 30 días) y CLR3 (Nivel de educación):

- Los estudiantes de 7° y 8° básico tienen una mayor proporción de no fumadores en comparación con los de niveles superiores.
- A medida que el nivel de educación aumenta, parece haber un incremento en el número de días fumados, especialmente en las categorías “3 a 5 días” y “Todos los días”.

La relación entre el nivel educativo y el comportamiento de fumar puede estar influenciada por varios factores, como la exposición al tabaco y la presión de pares en niveles educativos superiores. El hecho de que los estudiantes en niveles más avanzados tengan una mayor prevalencia de días fumados podría indicar que el riesgo de fumar aumenta con la edad o el acceso a cigarrillos.

```
valores_CR22 <- c("1" = "Sí", "2" = "No")

# Tabla de contingencia entre CR7 y CR22 (Visto fumar en la escuela)
tabla_CR7_CR22 <- table(factor(gytsAux$CR7, labels = valores_CR7),
                        factor(gytsAux$CR22, labels = valores_CR22))
print(tabla_CR7_CR22)
```

```
##
##                    Sí   No
##   0 días         1088  976
##   1 o 2 días      103   84
##   10 a 19 días     74   35
##   20 a 29 días     48   22
##   3 a 5 días       58   23
##   6 a 9 días       40   16
##   Todos los días   41   13
```

Asociación entre CR7 (Días fumados en los últimos 30 días) y CR22 (Visto fumar en la escuela):

- Aquellos que han visto a otros fumar en la escuela tienden a reportar más días fumados en los últimos 30 días.
- La proporción de encuestados que no han fumado ("0 días") es menor entre aquellos que han visto fumar en la escuela en comparación con aquellos que no lo han visto.

La exposición a compañeros fumando en la escuela parece correlacionarse con una mayor propensión a fumar. Esto resalta la importancia del entorno escolar en la influencia sobre el comportamiento de los adolescentes respecto al tabaquismo.

```
valores_OR46 <- c("1" = "Ninguno de ellos", "2" = "Algunos de ellos",
                  "3" = "La mayoría de ellos", "4" = "Todos ellos")

# Tabla de contingencia entre CR7 y OR46 (Amigos cercanos que fuman)
tabla_CR7_OR46 <- table(factor(gytsAux$CR7, labels = valores_CR7),
                        factor(gytsAux$OR46, labels = valores_OR46))
print(tabla_CR7_OR46)
```

```
##
##                  Ninguno de ellos Algunos de ellos La mayoría de ellos
##   0 días                      708             1024                 280
##   1 o 2 días                   10               93                  71
##   10 a 19 días                  2               44                  55
##   20 a 29 días                  2               31                  32
##   3 a 5 días                    2               22                  47
##   6 a 9 días                    5                8                  40
##   Todos los días                3               16                  23
##
##                  Todos ellos
##   0 días                  31
##   1 o 2 días              10
##   10 a 19 días             6
##   20 a 29 días             6
##   3 a 5 días              10
##   6 a 9 días               3
##   Todos los días           9
```

Asociación entre CR7 (Días fumados en los últimos 30 días) y OR46 (Amigos cercanos que fuman):

- Una gran proporción de no fumadores ("0 días") tiene pocos o ningún amigo que fuma.
- A medida que el número de amigos fumadores aumenta, también lo hace la cantidad de días que el encuestado reporta haber fumado, especialmente en las categorías de fumadores más frecuentes ("10 a 19 días", "20 a 29 días", "Todos los días").

La influencia de los amigos cercanos es claramente un factor importante en el comportamiento de fumar. Los datos muestran que tener más amigos que fuman se asocia con un mayor número de días fumados, lo que subraya la importancia de los vinculos sociales en la adopción y mantenimiento del hábito de fumar.

Analisis de asociacion con Chi-cuadrado

```r
# Test Chi-cuadrado entre CR7 y CR2 (Sexo)
chisq_test_CR7_CR2 <- chisq.test(gytsAux$CR7, gytsAux$CR2)
cat("Resultado del Test Chi-cuadrado entre CR7 y CR2 (Sexo):\n")
```

```
## Resultado del Test Chi-cuadrado entre CR7 y CR2 (Sexo):
```

```r
print(chisq_test_CR7_CR2)
```

```
##
##  Pearson's Chi-squared test
##
## data:  gytsAux$CR7 and gytsAux$CR2
## X-squared = 40.108, df = 6, p-value = 4.338e-07
```

```r
cat("\n")
```

```r
# Test Chi-cuadrado entre CR7 y CLR3 (Nivel de educación)
chisq_test_CR7_CLR3 <- chisq.test(gytsAux$CR7, gytsAux$CLR3)
cat("Resultado del Test Chi-cuadrado entre CR7 y CLR3 (Nivel de educación):\n")
```

```
## Resultado del Test Chi-cuadrado entre CR7 y CLR3 (Nivel de educación):
```

```r
print(chisq_test_CR7_CLR3)
```

```
##
##  Pearson's Chi-squared test
##
## data:  gytsAux$CR7 and gytsAux$CLR3
## X-squared = 207.48, df = 30, p-value < 2.2e-16
```

```r
cat("\n")
```

```r
# Test Chi-cuadrado entre CR7 y CR22 (Visto fumar en la escuela)
chisq_test_CR7_CR22 <- chisq.test(gytsAux$CR7, gytsAux$CR22)
cat("Resultado del Test Chi-cuadrado entre CR7 y CR22 (Visto fumar en la escuela):\n")
```

```
## Resultado del Test Chi-cuadrado entre CR7 y CR22 (Visto fumar en la escuela):
```

```r
print(chisq_test_CR7_CR22)
```

```
##
##  Pearson's Chi-squared test
##
## data:  gytsAux$CR7 and gytsAux$CR22
## X-squared = 41.474, df = 6, p-value = 2.335e-07
```

```r
cat("\n")
```

```r
# Test Chi-cuadrado entre CR7 y OR46 (Amigos cercanos que fuman)
chisq_test_CR7_OR46 <- chisq.test(gytsAux$CR7, gytsAux$OR46)
```

```
## Warning in chisq.test(gytsAux$CR7, gytsAux$OR46): Chi-squared approximation may
## be incorrect
```

```r
cat("Resultado del Test Chi-cuadrado entre CR7 y OR46 (Amigos cercanos que fuman):\n")
```

```
## Resultado del Test Chi-cuadrado entre CR7 y OR46 (Amigos cercanos que fuman):
```

```r
print(chisq_test_CR7_OR46)
```

```
##
##  Pearson's Chi-squared test
##
## data:  gytsAux$CR7 and gytsAux$OR46
## X-squared = 539.82, df = 18, p-value < 2.2e-16
```

```r
cat("\n")
```

1. Asociación entre CR7 (Días fumados en los últimos 30 días) y CR2 (Sexo):

- Estadístico Chi-cuadrado (X-squared): 40.108
- Grados de libertad (df): 6
- p-valor: 4.338e-07

El p-valor es extremadamente pequeño (mucho menor que 0.05), lo que indica que hay una relación significativa entre la cantidad de días que los encuestados han fumado en los últimos 30 días y su sexo. Esto sugiere que el comportamiento de fumar podría diferir entre hombres y mujeres, lo que coincide con la literatura que a menudo señala diferencias de género en el hábito de fumar.

2. Asociación entre CR7 (Días fumados en los últimos 30 días) y CLR3 (Nivel de educación):

- Estadístico Chi-cuadrado (X-squared): 207.48
- Grados de libertad (df): 30
- p-valor: $<$ 2.2e-16

El p-valor es nuevamente extremadamente pequeño, indicando una fuerte relación entre la cantidad de días que los encuestados han fumado y su nivel de educación. Este resultado sugiere que el nivel de educación puede influir en la frecuencia de fumar, probablemente debido a la influencia de la educación en la percepción de los riesgos para la salud y el acceso a información sobre los peligros del tabaco.

3. Asociación entre CR7 (Días fumados en los últimos 30 días) y CR22 (Visto fumar en la escuela):

- Estadístico Chi-cuadrado (X-squared): 41.474
- Grados de libertad (df): 6
- p-valor: 2.335e-07

El p-valor indica una relación significativa entre la cantidad de días que los encuestados han fumado en los últimos 30 días y si han visto a otros fumar en la escuela. Esto sugiere que la exposición al comportamiento de fumar en un entorno escolar puede estar asociada con una mayor probabilidad de que los estudiantes fumen, lo que es coherente con teorías sobre el modelado de comportamiento y la influencia de pares.

4. Asociación entre CR7 (Días fumados en los últimos 30 días) y OR46 (Amigos cercanos que fuman):

- Estadístico Chi-cuadrado (X-squared): 539.82
- Grados de libertad (df): 18
- p-valor: $< 2.2e\text{-}16$

El p-valor extremadamente pequeño indica una relación muy fuerte entre la cantidad de días que los encuestados han fumado en los últimos 30 días y si sus amigos cercanos fuman. Esto respalda la idea de que la presión de grupo y las normas sociales dentro de los círculos de amigos son factores clave en la adopción del hábito de fumar.

La advertencia sobre la aproximación incorrecta del Chi-cuadrado sugiere que podría haber celdas en la tabla de contingencia con frecuencias esperadas muy bajas. Esto podría afectar la validez del test de Chi-cuadrado, sugiriendo que los resultados deben interpretarse con cautela.

`En los próximos apartados debe analizar un problema de aprendizaje supervisado de clasificación mediante una regresión logística y el algoritmo Naive Bayes. Considere como variable objetivo una transformación binaria de la variable "Q7 (CR7) During the past 30 days, on how many days did you smoke cigarettes?" (vea el archivo "GYTSPAHO2016 Chile All Schools Region 4 (Metropolitana) Web Codebook.pdf"). En R realice una copia independiente de la data para cada modelo, llámelas "gytsRL" y "gytsNB" para la regresión logística y Naive Bayes correspondientemente. Para el desarrollo del informe realizar el procedimiento completo para una técnica y luego continuar con la siguiente:`

Item 4

4. Elimine las 110 observaciones con valores perdidos en la variable "Q7 (CR7)..." Transforme la variable categórica "Q7 (CR7) During the past 30 days, on how many days did you smoke cigarettes?" en una variable binaria de valor 1 cuando el/la joven muestre signos de ser fumador/a y 0 en otro caso. Comente.

```
# Vuelvo a cargar los datos originales y sin modificaciones
gytsAux <- as.data.frame(read.csv("METROPOLITANA_2016.csv"))
```

```
# Creo copias independientes de la base de datos
gytsRL <- gytsAux
gytsNB <- gytsAux
```

```
# Verifico la cantidad de valores faltantes por columna
cat("Número de filas originales:", nrow(gytsRL), "\n")
```

```
## Número de filas originales: 2778
```

```
cat("Número de valores faltantes por columna:\n")
```

```
## Número de valores faltantes por columna:
```

```r
print(colSums(is.na(gytsRL)))
```

```
## FinalWgt      CR1      CR2     CLR3     CLR4      CR5      CR6      CR7
##        0        2       20       15       22       55       53      110
##      CR8      CR9     CR10     CR11     CR12     CR13     CR14      OR9
##      107       78       59       25       26       58       69       38
##    CLR16    CLR17     ELR2     CR15     CR16     OR12     OR13     CR17
##       30       34       23       22       19       48       18       33
##     CR18     CR19     CR20    CLR27    CLR28     CR21     CR22     CR23
##       32       26       27       40       64       34       68       33
##     CR24    CLR33    CLR34     CR25    CLR36     CR27    CLR38    CLR39
##       78       44       57       55       51       54       40       36
##    CLR40    CLR41    CLR42     CR30     CR31     CR32    CLR46    CLR47
##       42       55       35       79       48       81       45       53
##    CLR48    CLR49     CR34     CR35     CR36     CR37     CR38     OR45
##       68       45       65       72       64      212      169       56
##     CR39     CR40     CR41     CR42     CR43     OR55    CLR62    CLR63
##       53       70       50       84       83      127       81       75
##    CLR64    CLR65     OR46     OR54    CLR68    CLR69    CLR70    CLR71
##       82       92       88      104       90      104      107      109
##    CLR72  Stratum      PSU
##      144        0        0
```

```r
# Imprimo número de filas con al menos un valor faltante
num_filas_faltantes <- sum(rowSums(is.na(gytsRL)) > 0)
cat("Número de filas con al menos un valor faltante:", num_filas_faltantes, "\n")
```

```
## Número de filas con al menos un valor faltante: 870
```

```r
# Elimino las observaciones con valores faltantes en la variable CR7
gytsRL <- gytsRL[!is.na(gytsRL$CR7), ]
gytsNB <- gytsNB[!is.na(gytsNB$CR7), ]
```

Considerando lo siguiente:

CR7 (Días fumados en los últimos 30 días)

- 1 = 0 días
- 2 = 1 o 2 días
- 3 = 10 a 19 días
- 4 = 20 a 29 días
- 5 = 3 a 5 días
- 6 = 6 a 9 días
- 7 = Todos los días

creo una nueva variable llamada `CR7_bin` en ambos dataframes (`gytsRL` y `gytsNB`). La variable `CR7_bin` tomará el valor de 1 si el individuo mostró signos de ser fumador (es decir, si `CR7` tiene alguno de los valores del 2 al 7) y 0 en caso contrario (cuando `CR7` es igual a 1, lo que representa 0 días fumados).

```r
# Transformo CR7 en una variable binaria
gytsRL$CR7_bin <- ifelse(gytsRL$CR7 %in% 2:7, 1, 0)
```

```r
gytsNB$CR7_bin <- ifelse(gytsNB$CR7 %in% 2:7, 1, 0)

# Elimino la variable CR7
# dado que CR7_bin es una transformación de CR7
# y no se necesita tener ambas variables
gytsRL$CR7 <- NULL
gytsNB$CR7 <- NULL
```

Item 5: Modelo Naive Bayes

5. **(Naive Bayes)** Realice el tratamiento necesario a las variables que utilizará finalmente en función del algoritmo. Puede crear otras variables si lo desea. Explique.

Como vimos anteriormente de un total de 2778 filas, hay 870 filas con almenos 1 valor faltante por lo que simplemente borrar esas filas no es algo factible dado que son demasiadas las filas que habrian que borrar, por lo que se procederá a realizar un proceso de imputacion MICE, para así tener 0 valores missing.

```r
# Selecciono variables para la imputación
# (excluyendo FinalWgt, Stratum, PSU y CR7_bin)
vars_imput <- names(gytsNB)[!names(gytsNB) %in%
                              c("FinalWgt", "Stratum", "PSU", "CR7_bin")]
```

```r
# Especifico el método de imputación
methods <- make.method(gytsNB)
# Uso 'pmm' para todas las variables a imputar
methods[vars_imput] <- "pmm"
```

```r
# Realizo la imputación MICE
# toma alrededor de 12 minutos
set.seed(12345)
imp <- mice(gytsNB[, vars_imput], m = 5, maxit = 50,
            method = methods[vars_imput], seed = 12345)
```

```r
# Extraigo el conjunto de datos imputado completo
gytsNB_imput <- complete(imp, 1)

# Añado las columnas que no se imputaron
gytsNB_imput$FinalWgt <- gytsNB$FinalWgt
gytsNB_imput$Stratum <- gytsNB$Stratum
gytsNB_imput$PSU <- gytsNB$PSU
gytsNB_imput$CR7_bin <- gytsNB$CR7_bin

# Verifico que no haya valores NA en el conjunto imputado
cat("Número de valores faltantes después de la imputación:\n")
```

```
## Número de valores faltantes después de la imputación:
```

```r
print(colSums(is.na(gytsNB_imput)))
```

```
##      CR1      CR2     CLR3     CLR4      CR5      CR6      CR8      CR9
##        0        0        0        0        0        0        0        0
##     CR10     CR11     CR12     CR13     CR14      OR9    CLR16    CLR17
##        0        0        0        0        0        0        0        0
##     ELR2     CR15     CR16     OR12     OR13     CR17     CR18     CR19
##        0        0        0        0        0        0        0        0
##     CR20    CLR27    CLR28     CR21     CR22     CR23     CR24    CLR33
##        0        0        0        0        0        0        0        0
##    CLR34     CR25    CLR36     CR27    CLR38    CLR39    CLR40    CLR41
##        0        0        0        0        0        0        0        0
##    CLR42     CR30     CR31     CR32    CLR46    CLR47    CLR48    CLR49
##        0        0        0        0        0        0        0        0
##     CR34     CR35     CR36     CR37     CR38      OR45     CR39     CR40
##        0        0        0        0        0        0        0        0
##     CR41     CR42     CR43     OR55    CLR62    CLR63    CLR64    CLR65
##        0        0        0        0        0        0        0        0
##     OR46     OR54    CLR68    CLR69    CLR70    CLR71    CLR72 FinalWgt
##        0        0        0        0        0        0        0        0
##  Stratum      PSU  CR7_bin
##        0        0        0
```

```r
# Función para crear dummies n-1 para cada columna categórica
crear_dummies <- function(df, cols) {
  for (col in cols) {
    # Identifico los niveles de la variable
    niveles <- sort(unique(df[[col]]))

    # Itero sobre los niveles menos uno
    for (nivel in niveles[-length(niveles)]) {
      nombre_columna <- paste0(col, "_", nivel)
      df[[nombre_columna]] <- as.integer(df[[col]] == nivel)
    }

    # Elimino la columna original categórica
    df[[col]] <- NULL
  }

  return(df)
}
```

```r
# Lista de columnas categóricas que convertiré en variables dummy
cols_categoricas <- c("Stratum", "CR1", "CR2", "CLR3", "CLR4", "CR5", "CR6",
                      "CR8", "CR9", "CR10", "CR11", "CR12", "CR13", "CR14",
                      "OR9", "CLR16", "CLR17", "ELR2", "CR15", "CR16",
                      "OR12", "OR13", "CR17", "CR18", "CR19", "CR20",
                      "CLR27", "CLR28", "CR21", "CR22", "CR23", "CR24",
                      "CLR33", "CLR34", "CR25", "CLR36", "CR27", "CLR38",
                      "CLR39", "CLR40", "CLR41", "CLR42", "CR30", "CR31",
                      "CR32", "CLR46", "CLR47", "CLR48", "CLR49", "CR34",
                      "CR35", "CR36", "CR37", "CR38", "OR45", "CR39",
                      "CR40", "CR41", "CR42", "CR43", "OR55", "CLR62",
                      "CLR63", "CLR64", "CLR65", "OR46", "OR54", "CLR68",
                      "CLR69", "CLR70", "CLR71", "CLR72")
```

```r
# Aplico la función a la base de datos
gytsNB_imput <- crear_dummies(gytsNB_imput, cols_categoricas)
```

```r
# Verifico que no haya valores NA después de crear dummies
print(colSums(is.na(gytsNB_imput)))
```

```
##          FinalWgt               PSU          CR7_bin Stratum_201604001
##                 0                 0                0                 0
## Stratum_201604002 Stratum_201604003 Stratum_201604004 Stratum_201604005
##                 0                 0                0                 0
## Stratum_201604006 Stratum_201604007 Stratum_201604008 Stratum_201604009
##                 0                 0                0                 0
## Stratum_201604010 Stratum_201604011            CR1_1             CR1_2
##                 0                 0                0                 0
##             CR1_3             CR1_4            CR1_5             CR1_6
##                 0                 0                0                 0
##             CR2_1            CLR3_1           CLR3_2            CLR3_3
##                 0                 0                0                 0
##            CLR3_4            CLR3_5           CLR4_1            CLR4_2
##                 0                 0                0                 0
##            CLR4_3            CLR4_4           CLR4_5            CLR4_6
##                 0                 0                0                 0
##             CR5_1             CR6_1            CR6_2             CR6_3
##                 0                 0                0                 0
##             CR6_4             CR6_5            CR6_6             CR8_1
##                 0                 0                0                 0
##             CR8_2             CR8_3            CR8_4             CR8_5
##                 0                 0                0                 0
##             CR8_6             CR9_1           CR10_1            CR11_1
##                 0                 0                0                 0
##            CR11_2            CR11_3           CR12_1            CR12_2
##                 0                 0                0                 0
##            CR12_3            CR12_4           CR12_5            CR12_6
##                 0                 0                0                 0
##            CR12_7            CR13_1           CR14_1             OR9_1
##                 0                 0                0                 0
##             OR9_2             OR9_3            OR9_4             OR9_5
##                 0                 0                0                 0
##             OR9_6             OR9_7           CLR16_1           CLR16_2
##                 0                 0                0                 0
##           CLR16_3           CLR16_4           CLR17_1           CLR17_2
##                 0                 0                0                 0
##           CLR17_3            ELR2_1           ELR2_2            ELR2_3
##                 0                 0                0                 0
##            ELR2_4            ELR2_5           ELR2_6            CR15_1
##                 0                 0                0                 0
##            CR15_2            CR15_3           CR16_1            CR16_2
##                 0                 0                0                 0
##            CR16_3            OR12_1           OR12_2            OR12_3
##                 0                 0                0                 0
##            OR12_4            OR12_5           OR12_6            OR13_1
##                 0                 0                0                 0
##            OR13_2            OR13_3           OR13_4            OR13_5
```

```
##          0          0          0          0
##     OR13_6     CR17_1     CR17_2     CR17_3
##          0          0          0          0
##     CR18_1     CR18_2     CR18_3     CR18_4
##          0          0          0          0
##     CR18_5     CR19_1     CR19_2     CR19_3
##          0          0          0          0
##     CR19_4     CR20_1     CR20_2     CR20_3
##          0          0          0          0
##     CR20_4    CLR27_1    CLR27_2    CLR27_3
##          0          0          0          0
##    CLR27_4    CLR28_1     CR21_1     CR21_2
##          0          0          0          0
##     CR21_3     CR21_4     CR22_1     CR23_1
##          0          0          0          0
##     CR23_2     CR23_3     CR24_1    CLR33_1
##          0          0          0          0
##    CLR33_2    CLR34_1     CR25_1    CLR36_1
##          0          0          0          0
##    CLR36_2    CLR36_3    CLR36_4    CLR36_5
##          0          0          0          0
##     CR27_1     CR27_2    CLR38_1    CLR38_2
##          0          0          0          0
##    CLR38_3    CLR38_4    CLR39_1    CLR39_2
##          0          0          0          0
##    CLR39_3    CLR39_4    CLR39_5    CLR40_1
##          0          0          0          0
##    CLR40_2    CLR41_1    CLR41_2    CLR42_1
##          0          0          0          0
##    CLR42_2    CLR42_3    CLR42_4    CLR42_5
##          0          0          0          0
##    CLR42_6    CLR42_7     CR30_1     CR31_1
##          0          0          0          0
##     CR31_2     CR32_1     CR32_2    CLR46_1
##          0          0          0          0
##    CLR46_2    CLR47_1    CLR47_2    CLR48_1
##          0          0          0          0
##    CLR48_2    CLR49_1    CLR49_2    CLR49_3
##          0          0          0          0
##    CLR49_4    CLR49_5     CR34_1     CR34_2
##          0          0          0          0
##     CR35_1     CR35_2     CR36_1     CR36_2
##          0          0          0          0
##     CR37_1     CR38_1     OR45_1     OR45_2
##          0          0          0          0
##     OR45_3     OR45_4     CR39_1     CR39_2
##          0          0          0          0
##     CR39_3     CR40_1     CR40_2     CR40_3
##          0          0          0          0
##     CR41_1     CR41_2     CR41_3     CR42_1
##          0          0          0          0
##     CR42_2     CR43_1     CR43_2     CR43_3
##          0          0          0          0
##     CR43_4     OR55_1    CLR62_1    CLR62_2
```

```
##                0                0                0                0
##          CLR62_3          CLR63_1          CLR63_2          CLR63_3
##                0                0                0                0
##          CLR63_4          CLR64_1          CLR64_2          CLR64_3
##                0                0                0                0
##          CLR65_1          CLR65_2           OR46_1           OR46_2
##                0                0                0                0
##           OR46_3           OR54_1           OR54_2           OR54_3
##                0                0                0                0
##          CLR68_1          CLR68_2          CLR68_3          CLR68_4
##                0                0                0                0
##          CLR68_5          CLR68_6          CLR69_1          CLR69_2
##                0                0                0                0
##          CLR69_3          CLR69_4          CLR69_5          CLR69_6
##                0                0                0                0
##          CLR70_1          CLR70_2          CLR71_1          CLR71_2
##                0                0                0                0
##          CLR72_1          CLR72_2
##                0                0
```

```
# Muestro las dimensiones del nuevo dataframe
print(dim(gytsNB_imput))
```

```
## [1] 2668  238
```

```
# Muestro los nombres de las columnas
print(names(gytsNB_imput))
```

```
##   [1] "FinalWgt"          "PSU"               "CR7_bin"
##   [4] "Stratum_201604001" "Stratum_201604002" "Stratum_201604003"
##   [7] "Stratum_201604004" "Stratum_201604005" "Stratum_201604006"
##  [10] "Stratum_201604007" "Stratum_201604008" "Stratum_201604009"
##  [13] "Stratum_201604010" "Stratum_201604011" "CR1_1"
##  [16] "CR1_2"             "CR1_3"             "CR1_4"
##  [19] "CR1_5"             "CR1_6"             "CR2_1"
##  [22] "CLR3_1"            "CLR3_2"            "CLR3_3"
##  [25] "CLR3_4"            "CLR3_5"            "CLR4_1"
##  [28] "CLR4_2"            "CLR4_3"            "CLR4_4"
##  [31] "CLR4_5"            "CLR4_6"            "CR5_1"
##  [34] "CR6_1"             "CR6_2"             "CR6_3"
##  [37] "CR6_4"             "CR6_5"             "CR6_6"
##  [40] "CR8_1"             "CR8_2"             "CR8_3"
##  [43] "CR8_4"             "CR8_5"             "CR8_6"
##  [46] "CR9_1"             "CR10_1"            "CR11_1"
##  [49] "CR11_2"            "CR11_3"            "CR12_1"
##  [52] "CR12_2"            "CR12_3"            "CR12_4"
##  [55] "CR12_5"            "CR12_6"            "CR12_7"
##  [58] "CR13_1"            "CR14_1"            "OR9_1"
##  [61] "OR9_2"             "OR9_3"             "OR9_4"
##  [64] "OR9_5"             "OR9_6"             "OR9_7"
##  [67] "CLR16_1"           "CLR16_2"           "CLR16_3"
##  [70] "CLR16_4"           "CLR17_1"           "CLR17_2"
##  [73] "CLR17_3"           "ELR2_1"            "ELR2_2"
```

```
##  [76] "ELR2_3"           "ELR2_4"           "ELR2_5"
##  [79] "ELR2_6"           "CR15_1"           "CR15_2"
##  [82] "CR15_3"           "CR16_1"           "CR16_2"
##  [85] "CR16_3"           "OR12_1"           "OR12_2"
##  [88] "OR12_3"           "OR12_4"           "OR12_5"
##  [91] "OR12_6"           "OR13_1"           "OR13_2"
##  [94] "OR13_3"           "OR13_4"           "OR13_5"
##  [97] "OR13_6"           "CR17_1"           "CR17_2"
## [100] "CR17_3"           "CR18_1"           "CR18_2"
## [103] "CR18_3"           "CR18_4"           "CR18_5"
## [106] "CR19_1"           "CR19_2"           "CR19_3"
## [109] "CR19_4"           "CR20_1"           "CR20_2"
## [112] "CR20_3"           "CR20_4"           "CLR27_1"
## [115] "CLR27_2"          "CLR27_3"          "CLR27_4"
## [118] "CLR28_1"          "CR21_1"           "CR21_2"
## [121] "CR21_3"           "CR21_4"           "CR22_1"
## [124] "CR23_1"           "CR23_2"           "CR23_3"
## [127] "CR24_1"           "CLR33_1"          "CLR33_2"
## [130] "CLR34_1"          "CR25_1"           "CLR36_1"
## [133] "CLR36_2"          "CLR36_3"          "CLR36_4"
## [136] "CLR36_5"          "CR27_1"           "CR27_2"
## [139] "CLR38_1"          "CLR38_2"          "CLR38_3"
## [142] "CLR38_4"          "CLR39_1"          "CLR39_2"
## [145] "CLR39_3"          "CLR39_4"          "CLR39_5"
## [148] "CLR40_1"          "CLR40_2"          "CLR41_1"
## [151] "CLR41_2"          "CLR42_1"          "CLR42_2"
## [154] "CLR42_3"          "CLR42_4"          "CLR42_5"
## [157] "CLR42_6"          "CLR42_7"          "CR30_1"
## [160] "CR31_1"           "CR31_2"           "CR32_1"
## [163] "CR32_2"           "CLR46_1"          "CLR46_2"
## [166] "CLR47_1"          "CLR47_2"          "CLR48_1"
## [169] "CLR48_2"          "CLR49_1"          "CLR49_2"
## [172] "CLR49_3"          "CLR49_4"          "CLR49_5"
## [175] "CR34_1"           "CR34_2"           "CR35_1"
## [178] "CR35_2"           "CR36_1"           "CR36_2"
## [181] "CR37_1"           "CR38_1"           "OR45_1"
## [184] "OR45_2"           "OR45_3"           "OR45_4"
## [187] "CR39_1"           "CR39_2"           "CR39_3"
## [190] "CR40_1"           "CR40_2"           "CR40_3"
## [193] "CR41_1"           "CR41_2"           "CR41_3"
## [196] "CR42_1"           "CR42_2"           "CR43_1"
## [199] "CR43_2"           "CR43_3"           "CR43_4"
## [202] "OR55_1"           "CLR62_1"          "CLR62_2"
## [205] "CLR62_3"          "CLR63_1"          "CLR63_2"
## [208] "CLR63_3"          "CLR63_4"          "CLR64_1"
## [211] "CLR64_2"          "CLR64_3"          "CLR65_1"
## [214] "CLR65_2"          "OR46_1"           "OR46_2"
## [217] "OR46_3"           "OR54_1"           "OR54_2"
## [220] "OR54_3"           "CLR68_1"          "CLR68_2"
## [223] "CLR68_3"          "CLR68_4"          "CLR68_5"
## [226] "CLR68_6"          "CLR69_1"          "CLR69_2"
## [229] "CLR69_3"          "CLR69_4"          "CLR69_5"
## [232] "CLR69_6"          "CLR70_1"          "CLR70_2"
## [235] "CLR71_1"          "CLR71_2"          "CLR72_1"
```

```
## [238] "CLR72_2"
```

¿Por qué consideré agregar a Stratum dentro del modelo de Machine learning pero no a PSU ni FinalWgt?

- **PSU (Primary Sampling Unit)**: PSU es una variable utilizada para controlar la estructura de la muestra y ajustar los errores estándar en análisis estadísticos. Por ende en contexto de machine learning no será útil porque no aporta directamente a la predicción, sino que es más relevante en el contexto del diseño de la encuesta.
- **FinalWgt (Final Weight)**: Es un peso de encuesta diseñado para ajustar las estimaciones para que sean representativas de la población general. No se utiliza como predictor en modelos de machine learning, sino más bien para ajustar las estimaciones agregadas a nivel de población. Si incluyera esta variable en el modelo podría sesgar los resultados en mi modelo, ya que los modelos de machine learning ya buscan patrones en los datos sin requerir este tipo de ponderación.
- **Stratum**: La variable Stratum puede ser relevante si los estratos capturan variaciones significativas en la población que pueden influir en el comportamiento que estás modelando (por ejemplo, si diferentes estratos demográficos o geográficos tienen comportamientos muy distintos en relación a la variable objetivo, lo que podría ser razonable considerando la desigualdad socioeconomica que existe en santiago).

```r
# Por ende procedo a eliminar las columnas que no aportan información
# PSU y FinalWgt
gytsNB_imput$PSU <- NULL
gytsNB_imput$FinalWgt <- NULL
```

```r
# Transformo las variables dummy y la variable objetivo a factores
gytsNB_imput <- gytsNB_imput %>%
  mutate(across(where(is.integer), as.factor))

# me aseguro de que la variable objetivo es un factor
gytsNB_imput$CR7_bin <- as.factor(gytsNB_imput$CR7_bin)

# Verificación final de la estructura del dataset
str(gytsNB_imput)
```

```
## 'data.frame':    2668 obs. of  236 variables:
##  $ CR7_bin          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604001: Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Stratum_201604002: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604003: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604004: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604005: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604006: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604007: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604008: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604009: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604010: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604011: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR1_1            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR1_2            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR1_3            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR1_4            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ CR1_5           : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 1 1 ...
##  $ CR1_6           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 2 2 ...
##  $ CR2_1           : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ CLR3_1          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR3_2          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR3_3          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR3_4          : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ CLR3_5          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR4_1          : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 1 1 1 1 ...
##  $ CLR4_2          : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
##  $ CLR4_3          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 2 2 ...
##  $ CLR4_4          : Factor w/ 2 levels "0","1": 1 1 1 1 2 2 2 1 1 1 ...
##  $ CLR4_5          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR4_6          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR5_1           : Factor w/ 2 levels "0","1": 2 1 1 1 2 2 2 2 2 1 ...
##  $ CR6_1           : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 1 2 ...
##  $ CR6_2           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR6_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ CR6_4           : Factor w/ 2 levels "0","1": 1 1 1 1 2 2 1 1 1 1 ...
##  $ CR6_5           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
##  $ CR6_6           : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 1 1 1 ...
##  $ CR8_1           : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ CR8_2           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR8_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR8_4           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR8_5           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR8_6           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR9_1           : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 2 1 ...
##  $ CR10_1          : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 1 ...
##  $ CR11_1          : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 1 1 1 2 ...
##  $ CR11_2          : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 2 2 2 1 ...
##  $ CR11_3          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR12_1          : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 2 2 ...
##  $ CR12_2          : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 1 2 1 1 ...
##  $ CR12_3          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR12_4          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR12_5          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR12_6          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR12_7          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR13_1          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR14_1          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR9_1           : Factor w/ 2 levels "0","1": 2 2 2 2 2 1 2 1 2 2 ...
##  $ OR9_2           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR9_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR9_4           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR9_5           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ OR9_6           : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
##  $ OR9_7           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR16_1         : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
##  $ CLR16_2         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ CLR16_3         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR16_4         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR17_1         : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 1 2 ...
##  $ CLR17_2         : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 2 1 1 ...
```

```
##  $ CLR17_3          : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 2 1 ...
##  $ ELR2_1           : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ ELR2_2           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ELR2_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ELR2_4           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ELR2_5           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ELR2_6           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR15_1           : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 1 2 ...
##  $ CR15_2           : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 1 2 2 1 ...
##  $ CR15_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR16_1           : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 1 2 ...
##  $ CR16_2           : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 2 2 1 ...
##  $ CR16_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
##  $ OR12_1           : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 1 2 ...
##  $ OR12_2           : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
##  $ OR12_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR12_4           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR12_5           : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
##  $ OR12_6           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
##  $ OR13_1           : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 1 2 ...
##  $ OR13_2           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR13_3           : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 1 2 2 1 ...
##  $ OR13_4           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR13_5           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR13_6           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR17_1           : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 1 2 ...
##  $ CR17_2           : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 1 2 2 1 ...
##  $ CR17_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR18_1           : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 1 1 2 ...
##   [list output truncated]
```

Item 6: Modelo Naive Bayes

Evaluacion del modelo Naive Bayes mediante ROC, teniendo como output un vector de probabilidades

6. **(Naive Bayes)** Inserte una semilla. Divida la base de datos en los conjuntos de entrenamiento y prueba. Verifique que la variable objetivo cumpla el supuesto de proporción en cada conjunto.

```r
# Inserto semilla para reproducibilidad
set.seed(12345)

# Divido el conjunto de datos en entrenamiento (70%) y prueba (30%)
split <- initial_split(gytsNB_imput, prop = 0.7, strata = "CR7_bin")
gytsNB_train <- training(split)
gytsNB_test <- testing(split)
```

```r
# Verificación de la proporción de la variable objetivo en los conjuntos
cat("Proporción de la variable objetivo en el conjunto de entrenamiento:\n")
```

```
## Proporción de la variable objetivo en el conjunto de entrenamiento:
```

```r
print(round(prop.table(table(gytsNB_train$CR7_bin)), 2))
```

```
##
##    0    1
## 0.78 0.22
```

```r
cat("Proporción de la variable objetivo en el conjunto de prueba:\n")
```

```
## Proporción de la variable objetivo en el conjunto de prueba:
```

```r
print(round(prop.table(table(gytsNB_test$CR7_bin)), 2))
```

```
##
##    0    1
## 0.78 0.22
```

Se realizó la verificación de la proporción de la variable objetivo CR7_bin, que indica si un individuo mostró signos de ser fumador en los últimos 30 días (1 para fumador, 0 para no fumador). Este paso es crucial para asegurarse de que las proporciones en los conjuntos de entrenamiento y prueba reflejen adecuadamente la distribución de la población original, lo que garantiza que los modelos entrenados y evaluados sean representativos.

Resultados:

Conjunto de Entrenamiento: - No fumador (X0): 78% - Fumador (X1): 22%

Conjunto de Prueba: - No fumador (X0): 78% - Fumador (X1): 22%

Estos resultados muestran que el 78% de las observaciones en ambos conjuntos representan individuos que no fumaron en los últimos 30 días (X0), mientras que el 22% representan a aquellos que sí lo hicieron (X1). La coincidencia exacta en la proporción entre los conjuntos de entrenamiento y prueba (0.78 para X0 y 0.22 para X1) indica que ambos conjuntos son representativos de la distribución original de la variable objetivo en la base de datos completa.

Esta representatividad es fundamental para garantizar que el modelo Naive Bayes que se entrene en este conjunto de datos tenga una generalización adecuada cuando se evalúe en el conjunto de prueba, y que los resultados obtenidos puedan aplicarse a nuevos datos con una distribución similar.

Item 7: Modelo Naive Bayes

7. **(Naive Bayes)** Prediga la variable objetivo del conjunto de prueba y muestre los resultados de la curva ROC. Reporte cuál es el punto de corte que seleccionó para transformar las probabilidades estimadas en clases estimadas y fundamente. Muestre la matriz de confusión final e interprete.

```r
# Me aseguro de que los niveles de CR7_bin sean nombres válidos en R
gytsNB_train$CR7_bin <- make.names(as.factor(gytsNB_train$CR7_bin))
gytsNB_test$CR7_bin <- make.names(as.factor(gytsNB_test$CR7_bin))
```

```r
# Estrategia de remuestreo con cross-validation k-fold (5) enfocada en ROC
cv <- trainControl(method = "cv", number = 5, classProbs = TRUE,
                   summaryFunction = twoClassSummary)
```

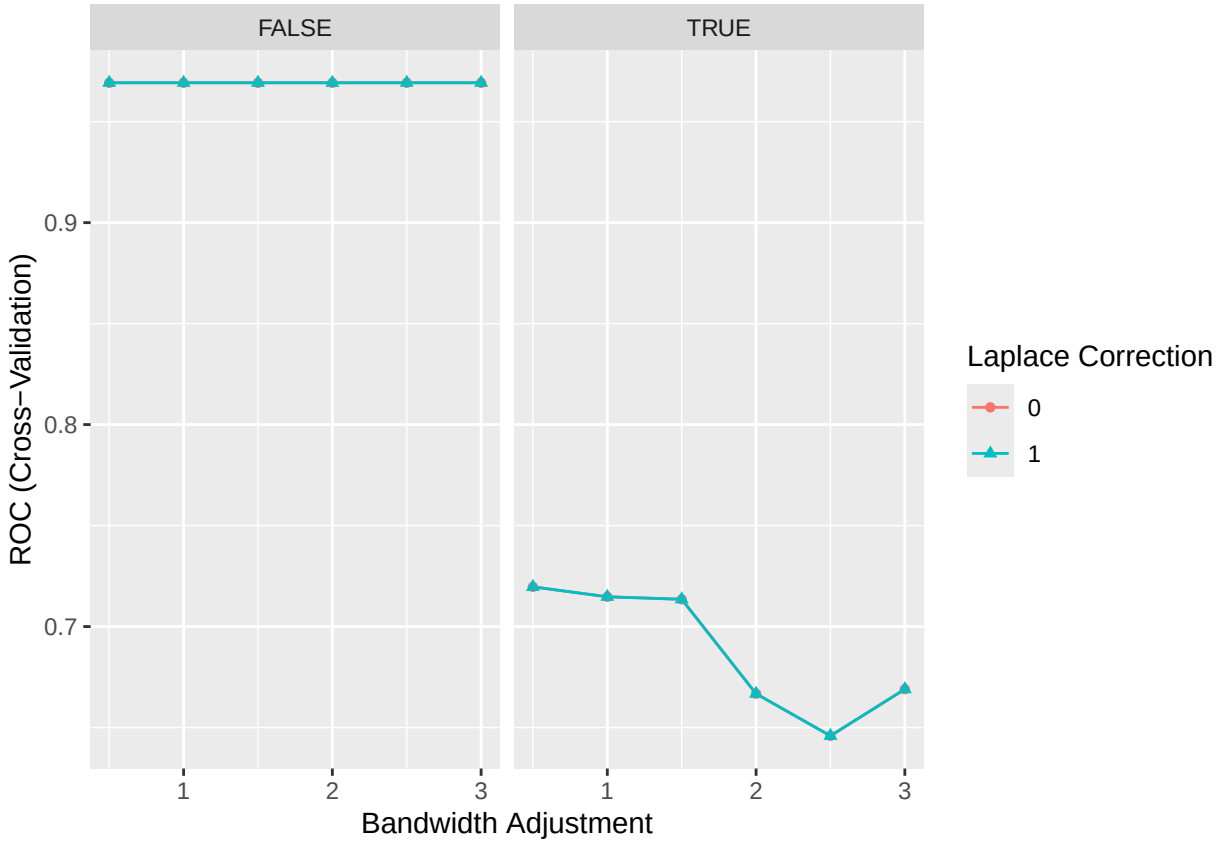
```r
# grilla de hiperparámetros
hyper_grid <- expand.grid(
  laplace = c(0, 1),
  usekernel = c(TRUE, FALSE),
  adjust = seq(0.5, 3, by = 0.5)
)
```

```r
# Entrenamiento del modelo Naive Bayes optimizado para ROC
set.seed(12345)
naive_bayes_fit_roc <- train(
  CR7_bin ~ .,
  data = gytsNB_train,
  method = "naive_bayes",
  metric = "ROC",
  trControl = cv,
  tuneGrid = hyper_grid
)

# Muestro los resultados del modelo
print(naive_bayes_fit_roc)
```

```
## Naive Bayes
##
## 1867 samples
##  235 predictor
##    2 classes: 'X0', 'X1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 1494, 1493, 1494, 1494, 1493
## Resampling results across tuning parameters:
##
##   laplace  usekernel  adjust  ROC        Sens       Spec
##   0        FALSE      0.5     0.9692618  0.9253425  0.9582656
##   0        FALSE      1.0     0.9692618  0.9253425  0.9582656
##   0        FALSE      1.5     0.9692618  0.9253425  0.9582656
##   0        FALSE      2.0     0.9692618  0.9253425  0.9582656
##   0        FALSE      2.5     0.9692618  0.9253425  0.9582656
##   0        FALSE      3.0     0.9692618  0.9253425  0.9582656
##   0         TRUE      0.5     0.7196589  1.0000000  0.0000000
##   0         TRUE      1.0     0.7147550  1.0000000  0.0000000
##   0         TRUE      1.5     0.7135101  1.0000000  0.0000000
##   0         TRUE      2.0     0.6667439  1.0000000  0.0000000
##   0         TRUE      2.5     0.6458857  1.0000000  0.0000000
##   0         TRUE      3.0     0.6689937  1.0000000  0.0000000
##   1        FALSE      0.5     0.9692618  0.9253425  0.9582656
##   1        FALSE      1.0     0.9692618  0.9253425  0.9582656
##   1        FALSE      1.5     0.9692618  0.9253425  0.9582656
##   1        FALSE      2.0     0.9692618  0.9253425  0.9582656
##   1        FALSE      2.5     0.9692618  0.9253425  0.9582656
##   1        FALSE      3.0     0.9692618  0.9253425  0.9582656
##   1         TRUE      0.5     0.7196589  1.0000000  0.0000000
##   1         TRUE      1.0     0.7147550  1.0000000  0.0000000
```

```
##   1         TRUE      1.5     0.7135101  1.0000000  0.0000000
##   1         TRUE      2.0     0.6667439  1.0000000  0.0000000
##   1         TRUE      2.5     0.6458857  1.0000000  0.0000000
##   1         TRUE      3.0     0.6689937  1.0000000  0.0000000
##
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were laplace = 0, usekernel = FALSE
##  and adjust = 0.5.
```

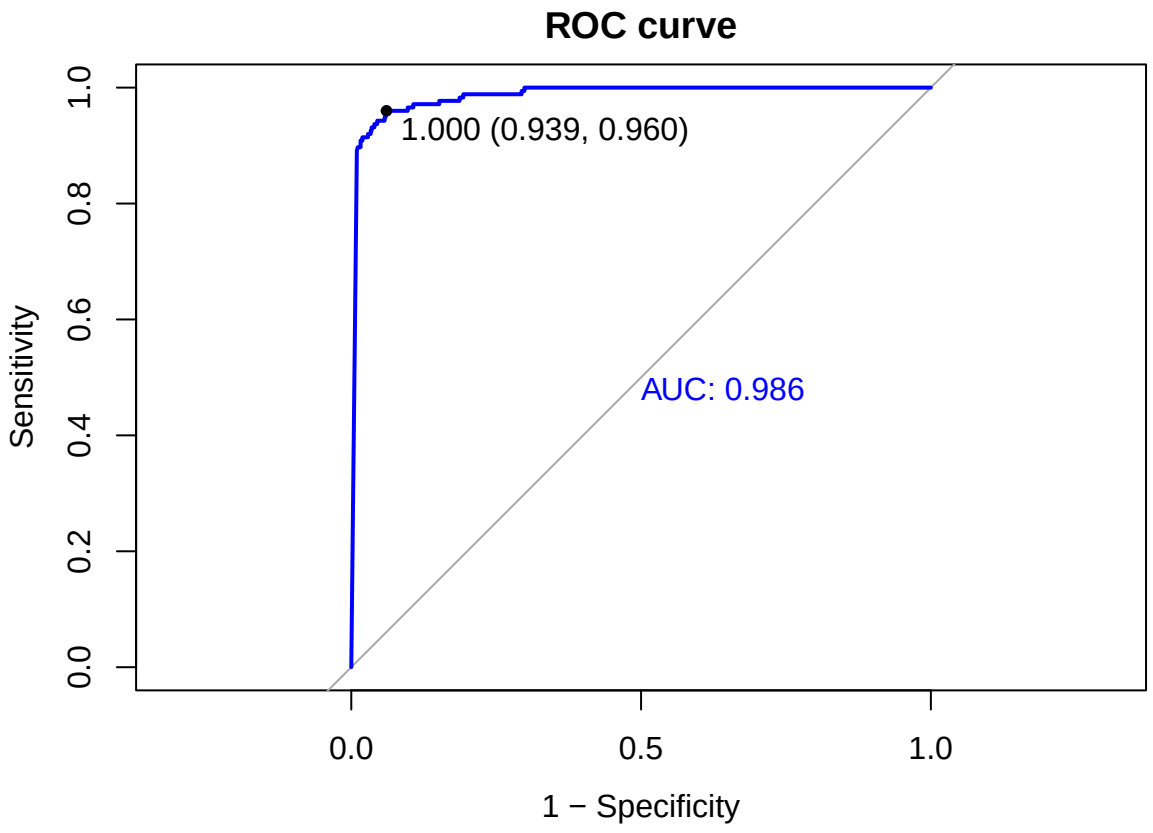
```r
ggplot(naive_bayes_fit_roc)
```




```r
# Entrenamiento del Mejor Modelo basado en ROC
set.seed(12345)
nb_roc <- naive_bayes(
  CR7_bin ~ .,
  data = gytsNB_train,
  laplace = 1, #el suavizado de laplace resuelve el problema de probabilidad 0
  usekernel = FALSE,
  adjust = 0.5
)
```

```r
# Predicciones de probabilidad en el conjunto de prueba
nb_roc_prob <- predict(
  nb_roc,
  gytsNB_test[, colnames(gytsNB_test) != "CR7_bin"],
  type = "prob"
```

```
)
nb_roc_prob <- as.numeric(nb_roc_prob[, "X1"])

# Generación de la curva ROC
nb_roc_curve <- roc(gytsNB_test$CR7_bin, nb_roc_prob,
                    levels = c("X0", "X1"), direction = "<")
plot.roc(nb_roc_curve, main = "ROC curve", col = "blue", lwd = 2,
         legacy.axes = TRUE, print.thres = "best", print.auc = TRUE)
```




```
# Cálculo del punto de corte óptimo basado en la suma
# de sensibilidades y especificidades
aux <- nb_roc_curve$sensitivities + nb_roc_curve$specificities
corte <- nb_roc_curve$thresholds[which(aux == max(aux))]

# Convierto las probabilidades a clases binarias usando el punto de corte óptimo
nb_roc_class <- ifelse(nb_roc_prob < corte, "X0", "X1")
nb_roc_class <- factor(nb_roc_class, levels = c("X0", "X1"))

# Me aseguro de que `gytsNB_test$CR7_bin` tenga los mismos niveles
gytsNB_test$CR7_bin <- factor(gytsNB_test$CR7_bin, levels = c("X0", "X1"))

# Matriz de confusión y sensibilidad + especificidad
results <- confusionMatrix(nb_roc_class, gytsNB_test$CR7_bin)
print(results$table)
```

```
##           Reference
```

```
## Prediction  X0  X1
##         X0 588   7
##         X1  38 168
```

- La variable objetivo en este análisis es CR7_bin, que indica si un individuo mostró signos de ser fumador, es decir, si fumó al menos un día en los últimos 30 días (CR7_bin = 1), o si no fumó en absoluto (CR7_bin = 0). Todas las demás variables en el dataset se utilizaron como predictores para construir el modelo Naive Bayes.
- El modelo Naive Bayes fue entrenado y evaluado utilizando la métrica de ROC, lo que permitió medir la capacidad del modelo para discriminar entre los individuos que son fumadores y los que no lo son. La curva ROC obtenida mostró un AUC (Área Bajo la Curva) de 0.986, lo que indica un excelente rendimiento del modelo. Un AUC cercano a 1 implica que el modelo es capaz de distinguir casi perfectamente entre fumadores y no fumadores.
- El punto de corte óptimo fue seleccionado basándose en la maximización de la suma de sensibilidades y especificidades. Esto se hizo al calcular para cada posible umbral la suma de la sensibilidad (tasa de verdaderos positivos) y la especificidad (tasa de verdaderos negativos), y eligiendo el umbral que maximiza esta suma.

Punto de corte seleccionado: 0.939

- Este valor indica que cualquier probabilidad de que un individuo sea fumador superior a 0.939 se clasifica como 1 (fumador), y cualquier probabilidad igual o inferior se clasifica como 0 (no fumador). Este punto de corte fue elegido porque ofrece el mejor equilibrio entre la sensibilidad y la especificidad, lo que es crucial en contextos donde ambas medidas son importantes para una correcta clasificación.

La matriz de confusión generada a partir del modelo usando el punto de corte óptimo es la siguiente:

Predicción / Realidad	No Fumador (0)	Fumador (1)
No Fumador (0)	588	7
Fumador (1)	38	168

Interpretación de la Matriz de Confusión:

Verdaderos Negativos (588): El modelo clasificó correctamente a 588 individuos como no fumadores cuando realmente no lo eran.

Falsos Negativos (7): El modelo clasificó incorrectamente a 7 individuos como no fumadores cuando en realidad eran fumadores. Esto representa una pequeña cantidad de individuos fumadores que el modelo no pudo identificar.

Falsos Positivos (38): El modelo clasificó incorrectamente a 38 individuos como fumadores cuando en realidad no lo eran. Este número de falsos positivos es relevante para analizar, ya que en ciertos contextos, un alto número de falsos positivos podría ser problemático (por ejemplo, en intervenciones donde los recursos se destinan a aquellos clasificados como fumadores).

Verdaderos Positivos (168): El modelo identificó correctamente a 168 individuos como fumadores cuando realmente lo eran.

El modelo, con el punto de corte óptimo, ha logrado un buen equilibrio, reduciendo tanto los falsos negativos como los falsos positivos. Sin embargo, la decisión de qué umbral utilizar podría ajustarse dependiendo de la prioridad dada a la minimización de falsos negativos o falsos positivos en escenarios específicos.

Modelo de Regresión logistica

Item 5: Modelo Regresión Logistica

5. **(Regresión lógistica)** Realice el tratamiento necesario a las variables que utilizará finalmente en función del algoritmo. Puede crear otras variables si lo desea. Explique.

1ero hay que recordar que ya se le han hecho en el código 3 cambios a la copia de la base de datos para el modelo de regresión logistica que son los siguientes:

Se Eliminaron las observaciones con valores faltantes en la variable CR7: - `gytsRL <- gytsRL[!is.na(gytsRL$CR7), ]`

Se transformó CR7 en una variable binaria - `gytsRL$CR7_bin <- ifelse(gytsRL$CR7 %in% 2:7, 1, 0)`

Se eliminó la variable CR7 dado que CR7_bin es una transformación de CR7 y no se necesita tener ambas variables: - `gytsRL$CR7 <- NULL`

```
# Selecciono variables para la imputación
# (excluyendo FinalWgt, Stratum, PSU y CR7_bin)
vars_imput_RL <- names(gytsRL)[!names(gytsRL) %in%
                                    c("FinalWgt", "Stratum", "PSU", "CR7_bin")]
```

```
# Especifico el método de imputación
methods <- make.method(gytsRL)
# Uso 'pmm' para todas las variables a imputar
methods[vars_imput_RL] <- "pmm"
```

```
# Realizo la imputación MICE
# toma alrededor de 12 minutos
set.seed(12345)
imp <- mice(gytsRL[, vars_imput_RL], m = 5, maxit = 50,
            method = methods[vars_imput_RL], seed = 12345)
```

```
# Extraigo el conjunto de datos imputado completo
gytsRL_imput <- complete(imp, 1)

# Añado las columnas que no se imputaron
gytsRL_imput$FinalWgt <- gytsRL$FinalWgt
gytsRL_imput$Stratum <- gytsRL$Stratum
gytsRL_imput$PSU <- gytsRL$PSU
gytsRL_imput$CR7_bin <- gytsRL$CR7_bin

# Verifico que no haya valores NA en el conjunto imputado
cat("Número de valores faltantes después de la imputación:\n")
```

```
## Número de valores faltantes después de la imputación:
```

```
print(colSums(is.na(gytsRL_imput)))
```

```
##      CR1      CR2     CLR3     CLR4      CR5      CR6      CR8      CR9
##        0        0        0        0        0        0        0        0
##     CR10     CR11     CR12     CR13     CR14      OR9    CLR16    CLR17
##        0        0        0        0        0        0        0        0
##     ELR2     CR15     CR16     OR12     OR13     CR17     CR18     CR19
```

```
##        0        0        0        0        0        0        0        0
##     CR20    CLR27    CLR28     CR21     CR22     CR23     CR24    CLR33
##        0        0        0        0        0        0        0        0
##    CLR34     CR25    CLR36     CR27    CLR38    CLR39    CLR40    CLR41
##        0        0        0        0        0        0        0        0
##    CLR42     CR30     CR31     CR32    CLR46    CLR47    CLR48    CLR49
##        0        0        0        0        0        0        0        0
##     CR34     CR35     CR36     CR37     CR38     OR45     CR39     CR40
##        0        0        0        0        0        0        0        0
##     CR41     CR42     CR43     OR55    CLR62    CLR63    CLR64    CLR65
##        0        0        0        0        0        0        0        0
##     OR46     OR54    CLR68    CLR69    CLR70    CLR71    CLR72 FinalWgt
##        0        0        0        0        0        0        0        0
##  Stratum      PSU  CR7_bin
##        0        0        0
```

```
# Aplico la función de creación de dummies ya creada en Naive Bayes
# Considerando la variable "cols_categoricas" ya creada en NB
gytsRL_imput <- crear_dummies(gytsRL_imput, cols_categoricas)
```

```
# Verifico que no haya valores NA después de crear dummies
print(colSums(is.na(gytsRL_imput)))
```

```
##          FinalWgt               PSU           CR7_bin Stratum_201604001
##                 0                 0                 0                 0
## Stratum_201604002 Stratum_201604003 Stratum_201604004 Stratum_201604005
##                 0                 0                 0                 0
## Stratum_201604006 Stratum_201604007 Stratum_201604008 Stratum_201604009
##                 0                 0                 0                 0
## Stratum_201604010 Stratum_201604011            CR1_1            CR1_2
##                 0                 0                 0                 0
##            CR1_3            CR1_4            CR1_5            CR1_6
##                 0                 0                 0                 0
##            CR2_1           CLR3_1           CLR3_2           CLR3_3
##                 0                 0                 0                 0
##           CLR3_4           CLR3_5           CLR4_1           CLR4_2
##                 0                 0                 0                 0
##           CLR4_3           CLR4_4           CLR4_5           CLR4_6
##                 0                 0                 0                 0
##            CR5_1            CR6_1            CR6_2            CR6_3
##                 0                 0                 0                 0
##            CR6_4            CR6_5            CR6_6            CR8_1
##                 0                 0                 0                 0
##            CR8_2            CR8_3            CR8_4            CR8_5
##                 0                 0                 0                 0
##            CR8_6            CR9_1           CR10_1           CR11_1
##                 0                 0                 0                 0
##           CR11_2           CR11_3           CR12_1           CR12_2
##                 0                 0                 0                 0
##           CR12_3           CR12_4           CR12_5           CR12_6
##                 0                 0                 0                 0
##           CR12_7           CR13_1           CR14_1            OR9_1
##                 0                 0                 0                 0
```

```
##          OR9_2          OR9_3          OR9_4          OR9_5
##              0              0              0              0
##          OR9_6          OR9_7        CLR16_1        CLR16_2
##              0              0              0              0
##        CLR16_3        CLR16_4        CLR17_1        CLR17_2
##              0              0              0              0
##        CLR17_3         ELR2_1         ELR2_2         ELR2_3
##              0              0              0              0
##         ELR2_4         ELR2_5         ELR2_6         CR15_1
##              0              0              0              0
##         CR15_2         CR15_3         CR16_1         CR16_2
##              0              0              0              0
##         CR16_3         OR12_1         OR12_2         OR12_3
##              0              0              0              0
##         OR12_4         OR12_5         OR12_6         OR13_1
##              0              0              0              0
##         OR13_2         OR13_3         OR13_4         OR13_5
##              0              0              0              0
##         OR13_6         CR17_1         CR17_2         CR17_3
##              0              0              0              0
##         CR18_1         CR18_2         CR18_3         CR18_4
##              0              0              0              0
##         CR18_5         CR19_1         CR19_2         CR19_3
##              0              0              0              0
##         CR19_4         CR20_1         CR20_2         CR20_3
##              0              0              0              0
##         CR20_4        CLR27_1        CLR27_2        CLR27_3
##              0              0              0              0
##        CLR27_4        CLR28_1         CR21_1         CR21_2
##              0              0              0              0
##         CR21_3         CR21_4         CR22_1         CR23_1
##              0              0              0              0
##         CR23_2         CR23_3         CR24_1        CLR33_1
##              0              0              0              0
##        CLR33_2        CLR34_1         CR25_1        CLR36_1
##              0              0              0              0
##        CLR36_2        CLR36_3        CLR36_4        CLR36_5
##              0              0              0              0
##         CR27_1         CR27_2        CLR38_1        CLR38_2
##              0              0              0              0
##        CLR38_3        CLR38_4        CLR39_1        CLR39_2
##              0              0              0              0
##        CLR39_3        CLR39_4        CLR39_5        CLR40_1
##              0              0              0              0
##        CLR40_2        CLR41_1        CLR41_2        CLR42_1
##              0              0              0              0
##        CLR42_2        CLR42_3        CLR42_4        CLR42_5
##              0              0              0              0
##        CLR42_6        CLR42_7         CR30_1         CR31_1
##              0              0              0              0
##         CR31_2         CR32_1         CR32_2        CLR46_1
##              0              0              0              0
##        CLR46_2        CLR47_1        CLR47_2        CLR48_1
##              0              0              0              0
```

```
##          CLR48_2          CLR49_1          CLR49_2          CLR49_3
##                0                0                0                0
##          CLR49_4          CLR49_5           CR34_1           CR34_2
##                0                0                0                0
##           CR35_1           CR35_2           CR36_1           CR36_2
##                0                0                0                0
##           CR37_1           CR38_1           OR45_1           OR45_2
##                0                0                0                0
##           OR45_3           OR45_4           CR39_1           CR39_2
##                0                0                0                0
##           CR39_3           CR40_1           CR40_2           CR40_3
##                0                0                0                0
##           CR41_1           CR41_2           CR41_3           CR42_1
##                0                0                0                0
##           CR42_2           CR43_1           CR43_2           CR43_3
##                0                0                0                0
##           CR43_4           OR55_1          CLR62_1          CLR62_2
##                0                0                0                0
##          CLR62_3          CLR63_1          CLR63_2          CLR63_3
##                0                0                0                0
##          CLR63_4          CLR64_1          CLR64_2          CLR64_3
##                0                0                0                0
##          CLR65_1          CLR65_2           OR46_1           OR46_2
##                0                0                0                0
##           OR46_3           OR54_1           OR54_2           OR54_3
##                0                0                0                0
##          CLR68_1          CLR68_2          CLR68_3          CLR68_4
##                0                0                0                0
##          CLR68_5          CLR68_6          CLR69_1          CLR69_2
##                0                0                0                0
##          CLR69_3          CLR69_4          CLR69_5          CLR69_6
##                0                0                0                0
##          CLR70_1          CLR70_2          CLR71_1          CLR71_2
##                0                0                0                0
##          CLR72_1          CLR72_2
##                0                0
```

```
# Muestro las dimensiones del nuevo dataframe
print(dim(gytsRL_imput))
```

```
## [1] 2668  238
```

```
# Muestro los nombres de las columnas
print(names(gytsRL_imput))
```

```
##   [1] "FinalWgt"          "PSU"               "CR7_bin"
##   [4] "Stratum_201604001" "Stratum_201604002" "Stratum_201604003"
##   [7] "Stratum_201604004" "Stratum_201604005" "Stratum_201604006"
##  [10] "Stratum_201604007" "Stratum_201604008" "Stratum_201604009"
##  [13] "Stratum_201604010" "Stratum_201604011" "CR1_1"
##  [16] "CR1_2"             "CR1_3"             "CR1_4"
##  [19] "CR1_5"             "CR1_6"             "CR2_1"
##  [22] "CLR3_1"            "CLR3_2"            "CLR3_3"
```

```
##  [25] "CLR3_4"           "CLR3_5"           "CLR4_1"
##  [28] "CLR4_2"           "CLR4_3"           "CLR4_4"
##  [31] "CLR4_5"           "CLR4_6"           "CR5_1"
##  [34] "CR6_1"            "CR6_2"            "CR6_3"
##  [37] "CR6_4"            "CR6_5"            "CR6_6"
##  [40] "CR8_1"            "CR8_2"            "CR8_3"
##  [43] "CR8_4"            "CR8_5"            "CR8_6"
##  [46] "CR9_1"            "CR10_1"           "CR11_1"
##  [49] "CR11_2"           "CR11_3"           "CR12_1"
##  [52] "CR12_2"           "CR12_3"           "CR12_4"
##  [55] "CR12_5"           "CR12_6"           "CR12_7"
##  [58] "CR13_1"           "CR14_1"           "OR9_1"
##  [61] "OR9_2"            "OR9_3"            "OR9_4"
##  [64] "OR9_5"            "OR9_6"            "OR9_7"
##  [67] "CLR16_1"          "CLR16_2"          "CLR16_3"
##  [70] "CLR16_4"          "CLR17_1"          "CLR17_2"
##  [73] "CLR17_3"          "ELR2_1"           "ELR2_2"
##  [76] "ELR2_3"           "ELR2_4"           "ELR2_5"
##  [79] "ELR2_6"           "CR15_1"           "CR15_2"
##  [82] "CR15_3"           "CR16_1"           "CR16_2"
##  [85] "CR16_3"           "OR12_1"           "OR12_2"
##  [88] "OR12_3"           "OR12_4"           "OR12_5"
##  [91] "OR12_6"           "OR13_1"           "OR13_2"
##  [94] "OR13_3"           "OR13_4"           "OR13_5"
##  [97] "OR13_6"           "CR17_1"           "CR17_2"
## [100] "CR17_3"           "CR18_1"           "CR18_2"
## [103] "CR18_3"           "CR18_4"           "CR18_5"
## [106] "CR19_1"           "CR19_2"           "CR19_3"
## [109] "CR19_4"           "CR20_1"           "CR20_2"
## [112] "CR20_3"           "CR20_4"           "CLR27_1"
## [115] "CLR27_2"          "CLR27_3"          "CLR27_4"
## [118] "CLR28_1"          "CR21_1"           "CR21_2"
## [121] "CR21_3"           "CR21_4"           "CR22_1"
## [124] "CR23_1"           "CR23_2"           "CR23_3"
## [127] "CR24_1"           "CLR33_1"          "CLR33_2"
## [130] "CLR34_1"          "CR25_1"           "CLR36_1"
## [133] "CLR36_2"          "CLR36_3"          "CLR36_4"
## [136] "CLR36_5"          "CR27_1"           "CR27_2"
## [139] "CLR38_1"          "CLR38_2"          "CLR38_3"
## [142] "CLR38_4"          "CLR39_1"          "CLR39_2"
## [145] "CLR39_3"          "CLR39_4"          "CLR39_5"
## [148] "CLR40_1"          "CLR40_2"          "CLR41_1"
## [151] "CLR41_2"          "CLR42_1"          "CLR42_2"
## [154] "CLR42_3"          "CLR42_4"          "CLR42_5"
## [157] "CLR42_6"          "CLR42_7"          "CR30_1"
## [160] "CR31_1"           "CR31_2"           "CR32_1"
## [163] "CR32_2"           "CLR46_1"          "CLR46_2"
## [166] "CLR47_1"          "CLR47_2"          "CLR48_1"
## [169] "CLR48_2"          "CLR49_1"          "CLR49_2"
## [172] "CLR49_3"          "CLR49_4"          "CLR49_5"
## [175] "CR34_1"           "CR34_2"           "CR35_1"
## [178] "CR35_2"           "CR36_1"           "CR36_2"
## [181] "CR37_1"           "CR38_1"           "OR45_1"
## [184] "OR45_2"           "OR45_3"           "OR45_4"
```

```
## [187] "CR39_1"            "CR39_2"            "CR39_3"
## [190] "CR40_1"            "CR40_2"            "CR40_3"
## [193] "CR41_1"            "CR41_2"            "CR41_3"
## [196] "CR42_1"            "CR42_2"            "CR43_1"
## [199] "CR43_2"            "CR43_3"            "CR43_4"
## [202] "OR55_1"            "CLR62_1"           "CLR62_2"
## [205] "CLR62_3"           "CLR63_1"           "CLR63_2"
## [208] "CLR63_3"           "CLR63_4"           "CLR64_1"
## [211] "CLR64_2"           "CLR64_3"           "CLR65_1"
## [214] "CLR65_2"           "OR46_1"            "OR46_2"
## [217] "OR46_3"            "OR54_1"            "OR54_2"
## [220] "OR54_3"            "CLR68_1"           "CLR68_2"
## [223] "CLR68_3"           "CLR68_4"           "CLR68_5"
## [226] "CLR68_6"           "CLR69_1"           "CLR69_2"
## [229] "CLR69_3"           "CLR69_4"           "CLR69_5"
## [232] "CLR69_6"           "CLR70_1"           "CLR70_2"
## [235] "CLR71_1"           "CLR71_2"           "CLR72_1"
## [238] "CLR72_2"
```

```
#procedo a eliminar las columnas que no aportan información
# PSU y FinalWgt
gytsRL_imput$PSU <- NULL
gytsRL_imput$FinalWgt <- NULL
```

```
# Convierto la variable objetivo en factor
gytsRL_imput$CR7_bin <- as.factor(gytsRL_imput$CR7_bin)

# Convierto todas las variables dummy a factores
gytsRL_imput <- gytsRL_imput %>%
  mutate(across(where(is.integer), as.factor))

# Verificación final de la estructura del dataset
str(gytsRL_imput)
```

```
## 'data.frame':    2668 obs. of  236 variables:
##  $ CR7_bin          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604001: Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Stratum_201604002: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604003: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604004: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604005: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604006: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604007: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604008: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604009: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604010: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Stratum_201604011: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR1_1            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR1_2            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR1_3            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR1_4            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR1_5            : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 1 1 ...
##  $ CR1_6            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 2 2 ...
```

```
##  $ CR2_1           : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ CLR3_1          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR3_2          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR3_3          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR3_4          : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ CLR3_5          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR4_1          : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 1 1 1 1 ...
##  $ CLR4_2          : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
##  $ CLR4_3          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 2 2 ...
##  $ CLR4_4          : Factor w/ 2 levels "0","1": 1 1 1 1 2 2 2 1 1 1 ...
##  $ CLR4_5          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR4_6          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR5_1           : Factor w/ 2 levels "0","1": 2 1 1 1 2 2 2 2 2 1 ...
##  $ CR6_1           : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 1 2 ...
##  $ CR6_2           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR6_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ CR6_4           : Factor w/ 2 levels "0","1": 1 1 1 1 2 2 1 1 1 1 ...
##  $ CR6_5           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
##  $ CR6_6           : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 1 1 1 ...
##  $ CR8_1           : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ CR8_2           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR8_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR8_4           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR8_5           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR8_6           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR9_1           : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 2 1 ...
##  $ CR10_1          : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 1 ...
##  $ CR11_1          : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 1 1 1 2 ...
##  $ CR11_2          : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 2 2 2 1 ...
##  $ CR11_3          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR12_1          : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 2 2 ...
##  $ CR12_2          : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 1 2 1 1 ...
##  $ CR12_3          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR12_4          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR12_5          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR12_6          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR12_7          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR13_1          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR14_1          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR9_1           : Factor w/ 2 levels "0","1": 2 2 2 2 2 1 2 1 2 2 ...
##  $ OR9_2           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR9_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR9_4           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR9_5           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ OR9_6           : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
##  $ OR9_7           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR16_1         : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
##  $ CLR16_2         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ CLR16_3         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR16_4         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CLR17_1         : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 1 2 ...
##  $ CLR17_2         : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 2 1 1 ...
##  $ CLR17_3         : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 2 1 ...
##  $ ELR2_1          : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

```
##  $ ELR2_2           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ELR2_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ELR2_4           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ELR2_5           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ELR2_6           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR15_1           : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 1 2 ...
##  $ CR15_2           : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 1 2 2 1 ...
##  $ CR15_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR16_1           : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 1 2 ...
##  $ CR16_2           : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 2 2 1 ...
##  $ CR16_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
##  $ OR12_1           : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 1 2 ...
##  $ OR12_2           : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
##  $ OR12_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR12_4           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR12_5           : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
##  $ OR12_6           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
##  $ OR13_1           : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 1 2 ...
##  $ OR13_2           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR13_3           : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 1 2 2 1 ...
##  $ OR13_4           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR13_5           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OR13_6           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR17_1           : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 1 1 2 ...
##  $ CR17_2           : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 1 2 2 1 ...
##  $ CR17_3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CR18_1           : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 1 1 2 ...
##   [list output truncated]
```

Item 6: Modelo Regresión Logistica

6. **(Regresion logistica)** Inserte una semilla. Divida la base de datos en los conjuntos de entrenamiento y prueba. Verifique que la variable objetivo cumpla el supuesto de proporción en cada conjunto.

```r
# Inserto semilla para reproducibilidad
set.seed(12345)

# División del conjunto de datos en entrenamiento (70%) y prueba (30%)
split <- initial_split(gytsRL_imput, prop = 0.7, strata = "CR7_bin")
gytsRL_train <- training(split)
gytsRL_test <- testing(split)
```

```r
# Verificación de la proporción de la variable objetivo en los conjuntos
cat("Proporción de la variable objetivo en el conjunto de entrenamiento:\n")
```

```
## Proporción de la variable objetivo en el conjunto de entrenamiento:
```

```r
print(round(prop.table(table(gytsRL_train$CR7_bin)), 2))
```

```
##
##    0    1
## 0.78 0.22
```

```r
cat("Proporción de la variable objetivo en el conjunto de prueba:\n")
```

```
## Proporción de la variable objetivo en el conjunto de prueba:
```

```r
print(round(prop.table(table(gytsRL_test$CR7_bin)), 2))
```

```
##
##    0    1
## 0.78 0.22
```

Item 7: Modelo Regresión Logistica

7. **(Regresion logistica)** Prediga la variable objetivo del conjunto de prueba y muestre los resultados de la curva ROC. Reporte cuál es el punto de corte que seleccionó para transformar las probabilidades estimadas en clases estimadas y fundamente. Muestre la matriz de confusión final e interprete.

```r
# Ajusto el modelo
set.seed(12345)
model_rl <- glm(CR7_bin ~ ., data = gytsRL_train,
                family = binomial(), control = list(maxit = 50))
```

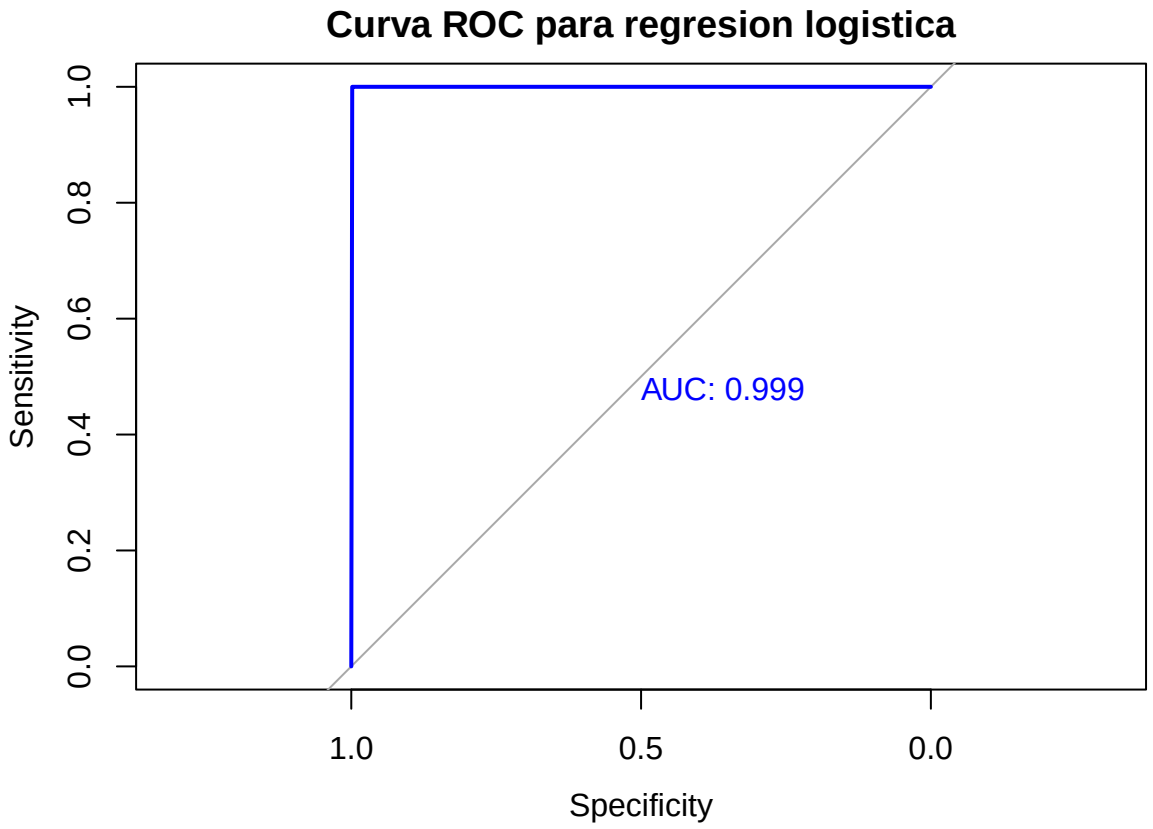
```r
# Realizo la predicción de probabilidades en el conjunto de prueba
prob_pred_rl <- predict(model_rl, gytsRL_test, type = "response")
```

```r
# Generación de la curva ROC
roc_rl <- roc(gytsRL_test$CR7_bin, prob_pred_rl)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
plot.roc(roc_rl, main = "Curva ROC para regresion logistica",
         col = "blue", lwd = 2, print.auc = TRUE)
```




```r
# Determino el punto de corte óptimo
corte_opt <- coords(roc_rl, "best", ret = "threshold", transpose = TRUE)

# Me aseguro de que sea un valor numérico
corte_opt <- as.numeric(corte_opt)

# Imprimo el punto de corte óptimo
cat("Punto de corte óptimo:", corte_opt, "\n")
```

```
## Punto de corte óptimo: 0.5
```

```r
# Conversión de probabilidades a clases utilizando el punto de corte óptimo
pred_class_rl <- ifelse(prob_pred_rl > corte_opt, 1, 0)
pred_class_rl <- factor(pred_class_rl, levels = c(0, 1))
```

```r
# genero matriz de confusión
conf_matrix_rl <- confusionMatrix(pred_class_rl, gytsRL_test$CR7_bin)
print(conf_matrix_rl$table)
```

```
##          Reference
## Prediction   0   1
##          0 625   0
##          1   1 175
```

Interpretación de la Matriz de Confusión:

- Verdaderos Negativos (TN): 625 (casos que son 0 y fueron predichos como 0).
- Falsos Negativos (FN): 0 (casos que son 1 pero fueron predichos como 0).
- Verdaderos Positivos (TP): 175 (casos que son 1 y fueron predichos como 1).
- Falsos Positivos (FP): 1 (casos que son 0 pero fueron predichos como 1).
- Perfecta Sensibilidad (FN = 0): Tener 0 falsos negativos significa que el modelo no cometió errores al identificar los casos positivos (los 1). Esto podría sugerir que el modelo está sobreajustado, especialmente si la variable objetivo está desbalanceada (muchos más 0 que 1).
- Casi Perfecta Especificidad (FP = 1): Tener solo 1 falso positivo es bastante inusual y sugiere que el modelo está funcionando excepcionalmente bien o que el conjunto de datos tiene un fuerte sesgo que facilita la predicción (por ejemplo, si los predictores tienen una relación muy fuerte con la variable objetivo).

Umbral de Corte (Threshold) en 0.5:

El hecho de que el umbral óptimo haya sido 0.5 es común en muchos problemas de clasificación, pero en algunos casos, podría ser un indicador de que el modelo no está considerando un balance adecuado entre sensibilidad y especificidad.

Posible Sobreajuste

Perfecta Separación: Los resultados que muestran un casi perfecto ajuste (solo un error) pueden indicar que el modelo está sobreajustado a los datos de entrenamiento, especialmente si el conjunto de datos es pequeño o si hay demasiadas variables predictoras. Esto puede resultar en un modelo que no generaliza bien a nuevos datos.

- AUC Alto: Un AUC de 0.999 significa que el modelo tiene una capacidad casi perfecta para distinguir entre las clases positivas y negativas. En teoría, esto es excelente, pero en la práctica, un valor tan alto puede indicar que el modelo está sobreajustado (overfitting)
- Sesgo de Datos: Otro posible problema podría ser que los datos están sesgados o que hay una fuerte correlación entre las variables predictoras y la variable objetivo, lo que hace que el modelo parezca casi perfecto en el conjunto de datos actual, pero posiblemente no se desempeñe tan bien en un conjunto de datos diferente.

Indique el algoritmo que tenga el mejor desempeño de acuerdo a la curva ROC. Explique su decisión.

Tras analizar el problema de clasificación del hábito de fumar en jóvenes, utilizando Regresión Logística y Naive Bayes, se concluye que ambos algoritmos son altamente efectivos para predecir si un individuo es fumador o no.

Comparación de desempeño:

Modelo	AUC	Precisión	Sensibilidad	Especificidad	Punto de Corte
Regresión Logística	0.999	0.998	1.0	0.998	0.5
Naive Bayes	0.986	0.917	0.823	0.960	0.939

Análisis Regresión Logística:

Ventajas:

Alcanzó una precisión casi perfecta (0.998) en el conjunto de prueba, clasificando correctamente casi todas las instancias.

Obtuvo un AUC de 0.999, lo que indica una capacidad de discriminación excepcional.

La matriz de confusión muestra un solo falso positivo y ningún falso negativo, lo que indica una alta capacidad para identificar correctamente tanto a fumadores como a no fumadores.

Desventajas:

La precisión casi perfecta, con tan pocos errores, genera preocupación por un posible sobreajuste a los datos de entrenamiento.

Análisis Naive Bayes:

Ventajas:

Mostró una excelente capacidad de discriminación con un AUC de 0.986 en la curva ROC.

Logró un buen equilibrio entre sensibilidad (0.823) y especificidad (0.960) con un punto de corte optimizado de 0.939.

Desventajas:

Presentó un mayor número de errores de clasificación en comparación con la regresión logística, con 7 falsos negativos y 38 falsos positivos.

Conclusión Final y Comparación de Modelos

Comparación y Elección del Mejor Modelo:

ROC y AUC: Aunque ambos modelos tienen un AUC alto, el modelo de regresión logística muestra un AUC ligeramente superior. Sin embargo, el modelo de Naive Bayes podría ser más robusto, ya que no mostró indicios significativos de sobreajuste en el conjunto de prueba.

Punto de Corte y Matriz de Confusión: El modelo de Naive Bayes parece manejar mejor el equilibrio entre la sensibilidad y especificidad con un punto de corte más alto. La matriz de confusión sugiere que Naive Bayes tiene un mejor desempeño práctico en la clasificación correcta de fumadores y no fumadores, aunque con un pequeño sacrificio en sensibilidad.

Decisión Basada en la curva ROC:

Dado que el AUC de la regresión logística es más alto, este algoritmo tiene un mejor desempeño en la clasificación de las observaciones en comparación con Naive Bayes.

La decisión se basa en el hecho de que un mayor AUC implica una mejor capacidad para predecir correctamente la clase positiva (fumador) y la clase negativa (no fumador) en un rango más amplio de puntos de corte. Esto hace que la regresión logística sea superior en términos de precisión y robustez para este problema específico.

De acuerdo a la curva ROC y al valor de AUC obtenido, la regresión logística tiene el mejor desempeño. La elección se justifica por su capacidad superior para distinguir entre las clases en comparación con Naive Bayes, lo que se refleja en un AUC más alto. Sin embargo, es importante tener en cuenta la posibilidad de sobreajuste debido al rendimiento excepcionalmente alto en el conjunto de prueba, lo cual podría indicar que el modelo ha capturado demasiado bien las características específicas del conjunto de entrenamiento, reduciendo su capacidad de generalización a nuevos datos.