# Minería de datos: PEC3 - Clasificación con árboles de decisión

Autor: Juan Luis Acebal Rico

Diciembre 2024

## Contents

**Recursos básicos** **2**

**Ejemplo ilustrativo** **2**
Análisis inicial . . . 3
Preparación de los datos para el modelo . . . 10
Creación del modelo, calidad del modelo y extracción de reglas . . . 11
Validación del modelo con los datos reservados . . . 13
Prueba con una variación u otro enfoque algorítmico . . . 14
Interpretación de las variables en las predicciones. . . . 16

**Enunciado del ejercicio** **18**
Realizar un primer análisis descriptivo y de correlaciones. Es importante en este apartado entender bien los datos antes de seguir con los análisis posteriores. Lista todo lo que te haya sorprendido de los datos . . . 18

**Ahora realizo un análisis de correlaciones entre las variables.** **22**

**¿Y las variables categóricas?** **44**
Realizar un primer árbol de decisión. Puedes decidir utilizar todas las variables o, de forma justificada, quitar alguna para el ajuste del modelo . . . 50
Con el árbol obtenido, realiza una breve explicación de las reglas obtenidas así como de todos los puntos que te parezcan interesantes. Un elemento a considerar es, por ejemplo, cuantas observaciones caen dentro de cada regla . . . 53

**¿Y la importancia de las variables?** **53**
Una vez tengas un modelo válido, procede a realizar un análisis de la bondad de ajuste sobre el conjunto de test y matriz de confusión. ¿Te parece un modelo suficientemente bueno como para utilizarlo? Justifica tu respuesta considerando todos los posibles tipos de error . . . 55
Con un enfoque parecido a los puntos anteriores y considerando las mismas variables, enriquece el ejercicio mediante el ajuste de modelos de árbol de decisión complementarios. ¿Es el nuevo enfoque mejor que el original? Justifica la respuesta . . . 55

**F1 SCORE** **59**
Haz un resumen de las principales conclusiones de todos los análisis y modelos realizados . . . 59

**F2 SCORE** **61**

**F0.5 SCORE** **61**

---

## Recursos básicos

---

Esta Prueba de Evaluación Continuada (PEC) cubre principalmente el material didáctico de modelos supervisados y evaluación de modelos.

Complementarios:

- Material docente "Creación y evaluación de modelos no supervisados" proporcionado por la UOC.
- Fichero titanic.csv.
- R package C5.0 (Decision Trees and Rule-Based Models): https://cran.r-project.org/web/packages/C50/index.html
- Fichero de "German Credit": credit.csv (se obtuvo de https://www.kaggle.com/shravan3273/credit-approval)

La descripción de las variables se puede ver en https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

**La variable "default" es el target siendo 1 = "No default" y 2 = "Default". Se deben utilizar estos datos para la realización de los ejercicios.**

---

## Ejemplo ilustrativo

---

En este ejercicio vamos a seguir los pasos del ciclo de vida de un proyecto de minería de datos, para el caso de un algoritmo de clasificación usaremos un árbol de decisión, **que es el algoritmo supervisado que vamos a tratar en esta asignatura**. Primero y a modo de ejemplo sencillo lo haremos con el archivo titanic.csv, que se encuentra adjunto en el aula. Este archivo contiene un registro por cada pasajero que viajaba en el Titanic. En las variables se caracteriza si era hombre o mujer, adulto o menor (niño), en qué categoría viajaba o si era miembro de la tripulación. Se mostrará un ejemplo sencillo de solución con estos datos pero los alumnos deberéis responder a las preguntas de la rúbrica para otro conjunto: German Credit. Para este conjunto, tomaréis como referencia la variable "default" que indica el impago de créditos.

**Objetivos:**

- Estudiar los datos, por ejemplo: ¿Número de registros del fichero? ¿Distribuciones de valores por variables? ¿Hay campos mal informados o vacíos?
- Preparar los datos. En este caso ya están en el formato correcto y no es necesario discretizar ni generar atributos nuevos. Hay que elegir cuáles son las variables que se utilizarán para construir el modelo y cuál es la variable que clasifica. En este caso la variable por la que clasificaremos es el campo de si el pasajero sobrevivia o no.
- Instalar, si es necesario, el paquete C5.0 Se trata de una implementación más moderna del algoritmo ID3 de Quinlan. Tiene los principios teóricos del ID3 más la poda automática. Con este paquete generar un modelo de minería.
- ¿Cuál es la calidad del modelo?
- Generar el árbol gráfico.
- Generar y extraer las reglas del modelo.
- En función del modelo, el árbol y las reglas: ¿Cuál es el conocimiento que obtenemos?
- Probar el modelo generado presentándole nuevos registros. ¿Clasifica suficientemente bien?

A continuación, se plantean los puntos a realizar en la PEC 3 y, tomando como ejemplo el conjunto de datos de Titanic, se obtendrán, a modo de ejemplo, algunos resultados que pretender servir a modo de inspiración para los estudiantes. Los estudiantes deberán utilizar el conjunto de datos de "German Credit Data" que se pueden conseguir en este enlace: https://www.kaggle.com/shravan3273/credit-approval

Este recurso puede ser útil para profundizar sobre el paquete IML: https://uc-r.github.io/iml-pkg

Revisión de los datos, extracción visual de información y preparación de los datos

Carga de los datos:

```
data<-read.csv("./titanic.csv",header=T,sep=",")
attach(data)
```

## Análisis inicial

Empezaremos haciendo un breve análisis de los datos ya que nos interesa tener una idea general de los datos que disponemos.

### Exploración de la base de datos

Primero calcularemos las dimensiones de nuestra base de datos y analizaremos qué tipos de atributos tenemos.

Para empezar, calculamos las dimensiones de la base de datos mediante la función dim(). Obtenemos que disponemos de 2201 registros o pasajeros (filas) y 4 variables (columnas).

```
dim(data)
```

```
## [1] 2201    4
```

¿Cuáles son esas variables? Gracias a la función str() sabemos que las cuatro variables son categóricas o discretas, es decir, toman valores en un conjunto finito. La variable CLASS hace referencia a la clase en la que viajaban los pasajeros (1ª, 2ª, 3ª o crew), AGE determina si era adulto o niño (Adulto o Menor), la variable SEX si era hombre o mujer (Hombre o Mujer) y la última variable (SURVIVED) informa si el pasajero murió o sobrevivió en el accidente (Muere o Sobrevive).

```
str(data)
```

```
## 'data.frame':    2201 obs. of  4 variables:
##  $ CLASS   : chr  "1a" "1a" "1a" "1a" ...
##  $ AGE     : chr  "Adulto" "Adulto" "Adulto" "Adulto" ...
##  $ SEX     : chr  "Hombre" "Hombre" "Hombre" "Hombre" ...
##  $ SURVIVED: chr  "Sobrevive" "Sobrevive" "Sobrevive" "Sobrevive" ...
```

Vemos que las variables están definidas como carácter, así que las transformamos a tipo factor.

```
data[] <- lapply(data, factor)
str(data)
```

```
## 'data.frame':    2201 obs. of  4 variables:
##  $ CLASS   : Factor w/ 4 levels "1a","2a","3a",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ AGE     : Factor w/ 2 levels "Adulto","Menor": 1 1 1 1 1 1 1 1 1 1 ...
##  $ SEX     : Factor w/ 2 levels "Hombre","Mujer": 1 1 1 1 1 1 1 1 1 1 ...
##  $ SURVIVED: Factor w/ 2 levels "Muere","Sobrevive": 2 2 2 2 2 2 2 2 2 2 ...
```

Es de gran interés saber si tenemos muchos valores nulos (campos vacíos) y la distribución de valores por variables. Es por ello recomendable empezar el análisis con una visión general de las variables. Mostraremos para cada atributo la cantidad de valores perdidos mediante la función summary.

```
summary(data)
```

```
##   CLASS         AGE           SEX            SURVIVED
##  1a  :325   Adulto:2092   Hombre:1731   Muere    :1490
##  2a  :285   Menor : 109   Mujer : 470   Sobrevive: 711
##  3a  :706
##  crew:885
```

Como parte de la preparación de los datos, miraremos si hay valores missing.

```
missing <- data[is.na(data),]
dim(missing)
```

```
## [1] 0 4
```

Observamos fácilmente que no hay valores missing y, por tanto, no deberemos preparar los datos en este sentido. En caso de haberlos, habría que tomar decisiones para tratar los datos adecuadamente.

Disponemos por tanto de un data frame formado por cuatro variables categóricas sin valores nulos.

### Visualización

Para un conocimiento mayor sobre los datos, tenemos a nuestro alcance unas herramientas muy valiosas: las herramientas de visualización. Para dichas visualizaciones, haremos uso de los paquetes ggplot2, gridExtra y grid de R.

```
if(!require(ggplot2)){
    install.packages('ggplot2', repos='http://cran.us.r-project.org')
    library(ggplot2)
}
```

```
## Loading required package: ggplot2
```

```
if(!require(ggpubr)){
    install.packages('ggpubr', repos='http://cran.us.r-project.org')
    library(ggpubr)
}
```

```
## Loading required package: ggpubr
```

```
if(!require(grid)){
    install.packages('grid', repos='http://cran.us.r-project.org')
    library(grid)
}
```

```
## Loading required package: grid
```

```
if(!require(gridExtra)){
    install.packages('gridExtra', repos='http://cran.us.r-project.org')
    library(gridExtra)
}
```

```
## Loading required package: gridExtra
```

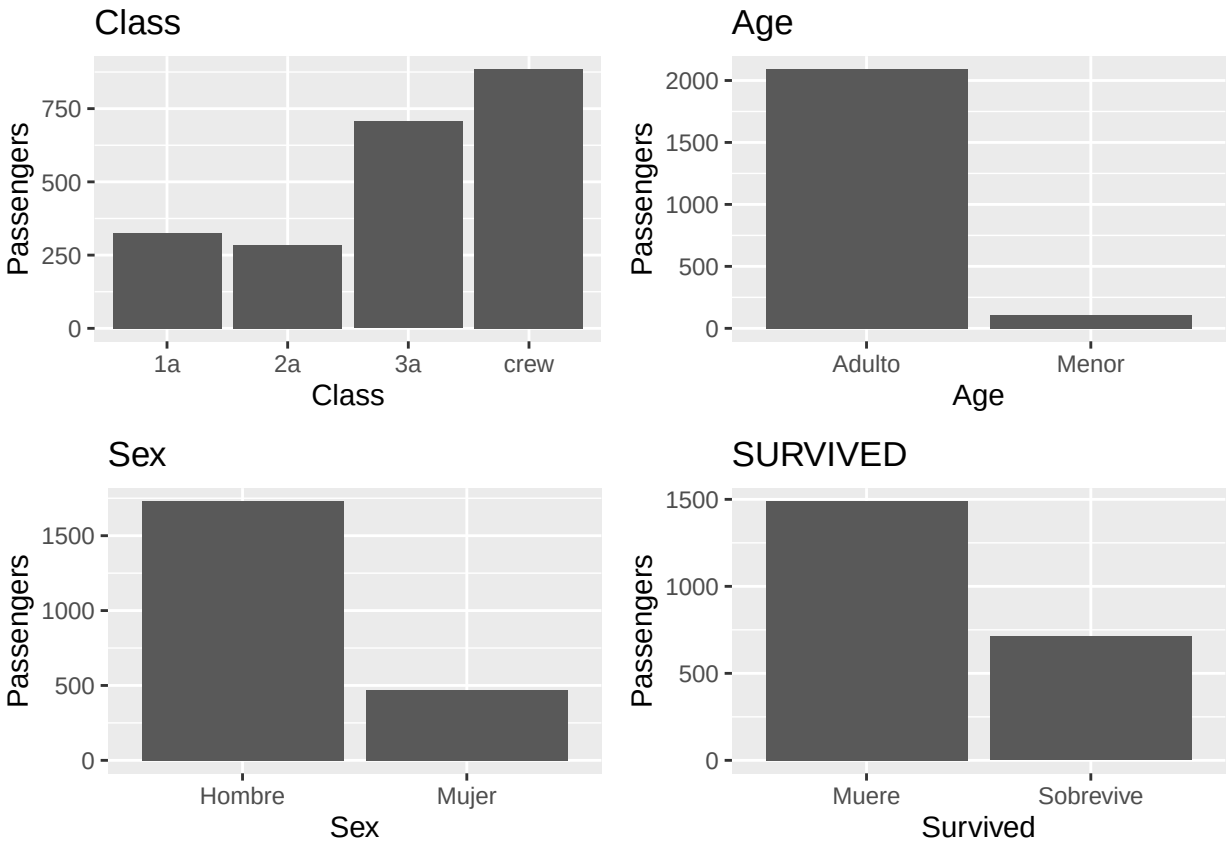
```
if(!require(C50)){
    install.packages('C50', repos='http://cran.us.r-project.org')
    library(C50)
}
```

```
## Loading required package: C50
```

Siempre es importante analizar los datos que tenemos ya que las conclusiones dependerán de las características de la muestra.

```
grid.newpage()
plotbyClass<-ggplot(data,aes(CLASS))+geom_bar() +labs(x="Class", y="Passengers")+ guides(fill=guide_leg
plotbyAge<-ggplot(data,aes(AGE))+geom_bar() +labs(x="Age", y="Passengers")+ guides(fill=guide_legend(ti
plotbySex<-ggplot(data,aes(SEX))+geom_bar() +labs(x="Sex", y="Passengers")+ guides(fill=guide_legend(ti
```

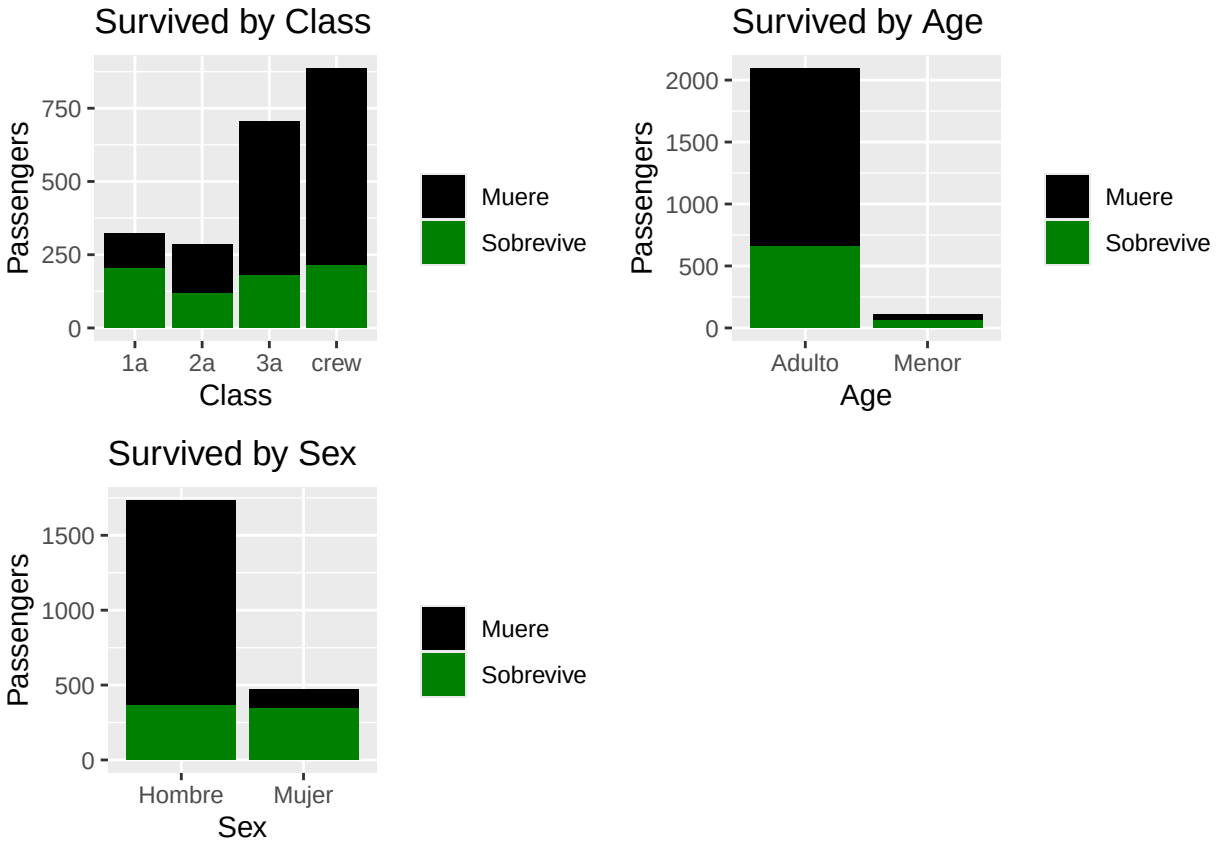
```
plotbySurvived<-ggplot(data,aes(SURVIVED))+geom_bar() +labs(x="Survived", y="Passengers")+ guides(fill=g
grid.arrange(plotbyClass,plotbyAge,plotbySex,plotbySurvived,ncol=2)
```




Claramente vemos cómo es la muestra analizando la distribución de las variables disponibles. De cara a los informes, es mucho más interesante esta información que la obtenida en summary, que se puede usar para complementar.

Nos interesa describir la relación entre la supervivencia y cada uno de las variables mencionadas anteriormente. Para ello, por un lado graficaremos mediante diagramas de barras la cantidad de muertos y supervivientes según la clase en la que viajaban, la edad o el sexo. Por otro lado, para obtener los datos que estamos graficando utilizaremos el comando table para dos variables que nos proporciona una tabla de contingencia.

```
grid.newpage()
plotbyClass<-ggplot(data,aes(CLASS,fill=SURVIVED))+geom_bar() +labs(x="Class", y="Passengers")+ guides(
plotbyAge<-ggplot(data,aes(AGE,fill=SURVIVED))+geom_bar() +labs(x="Age", y="Passengers")+ guides(fill=gu
plotbySex<-ggplot(data,aes(SEX,fill=SURVIVED))+geom_bar() +labs(x="Sex", y="Passengers")+ guides(fill=gu
grid.arrange(plotbyClass,plotbyAge,plotbySex,ncol=2)
```




De estos gráficos obtenemos información muy valiosa que complementamos con las tablas de contingencia (listadas abajo). Por un lado, la cantidad de pasajeros que sobrevivieron es similar en hombres y mujeres (hombres: 367 y mujeres 344). No, en cambio, si tenemos en cuenta el porcentaje respecto a su sexo. Es decir, pese a que la cantidad de mujeres y hombres que sobrevivieron es pareja, viajaban más hombres que mujeres (470 mujeres y 1731 hombres), por lo tanto, la tasa de muerte en hombres es muchísimo mayor (el 78,79% de los hombres murieron mientras que en mujeres ese porcentaje baja a 26,8%).

En cuanto a la clase en la que viajaban, los pasajeros que viajaban en primera clase fueron los únicos que el porcentaje de supervivencia era mayor que el de mortalidad. El 62,46% de los viajeros de primera clase sobrevivió, el 41,4% de los que viajaban en segunda clase mientras que de los viajeros de tercera y de la tripulación solo sobrevivieron un 25,21% y 23,95% respectivamente. Para finalizar, destacamos que la presencia de pasajeros adultos era mucho mayor que la de los niños (2092 frente a 109) y que la tasa de supervivencia en niños fue mucho mayor (52,29% frente a 31,26%), no podemos obviar, en cambio, que los únicos niños que murieron fueron todos pasajeros de tercera clase (52 niños).

```
tabla_SST <- table(SEX, SURVIVED)
tabla_SST
```

```
##         SURVIVED
## SEX      Muere Sobrevive
##   Hombre  1364       367
##   Mujer    126       344
```

```
prop.table(tabla_SST, margin = 1)
```

```
##         SURVIVED
## SEX          Muere Sobrevive
##   Hombre 0.7879838 0.2120162
```

```
##   Mujer  0.2680851 0.7319149
```

```
tabla_SCT <- table(CLASS,SURVIVED)
tabla_SCT
```

```
##       SURVIVED
## CLASS  Muere Sobrevive
##   1a     122       203
##   2a     167       118
##   3a     528       178
##   crew   673       212
```

```
prop.table(tabla_SCT, margin = 1)
```

```
##       SURVIVED
## CLASS      Muere Sobrevive
##   1a   0.3753846 0.6246154
##   2a   0.5859649 0.4140351
##   3a   0.7478754 0.2521246
##   crew 0.7604520 0.2395480
```

```
tabla_SAT <- table(AGE,SURVIVED)
tabla_SAT
```

```
##         SURVIVED
## AGE      Muere Sobrevive
##   Adulto  1438       654
##   Menor     52        57
```

```
prop.table(tabla_SAT, margin = 1)
```

```
##         SURVIVED
## AGE          Muere Sobrevive
##   Adulto 0.6873805 0.3126195
##   Menor  0.4770642 0.5229358
```

```
tabla_SAT.byClass <- table(AGE,SURVIVED,CLASS)
tabla_SAT.byClass
```

```
## , , CLASS = 1a
##
##         SURVIVED
## AGE      Muere Sobrevive
##   Adulto   122       197
##   Menor      0         6
##
## , , CLASS = 2a
##
##         SURVIVED
## AGE      Muere Sobrevive
##   Adulto   167        94
##   Menor      0        24
##
## , , CLASS = 3a
##
##         SURVIVED
## AGE      Muere Sobrevive
##   Adulto   476       151
```

```
##   Menor     52        27
##
## , , CLASS = crew
##
##          SURVIVED
## AGE       Muere Sobrevive
##   Adulto   673       212
##   Menor      0         0
```

### Test estadísticos de significancia

Los resultados anteriores muestran los datos de forma descriptiva, podemos añadir algún test estadístico para validar el grado de significancia de la relación. La librería "DescTools" nos permite instalarlo fácilmente.

```
if(!require(DescTools)){
    install.packages('DescTools', repos='http://cran.us.r-project.org')
    library(DescTools)
}
```

```
## Loading required package: DescTools
```

```
Phi(tabla_SST)
```

```
## [1] 0.4556048
```

```
CramerV(tabla_SST)
```

```
## [1] 0.4556048
```

```
Phi(tabla_SAT)
```

```
## [1] 0.09757511
```

```
CramerV(tabla_SAT)
```

```
## [1] 0.09757511
```

```
Phi(tabla_SCT)
```

```
## [1] 0.2941201
```

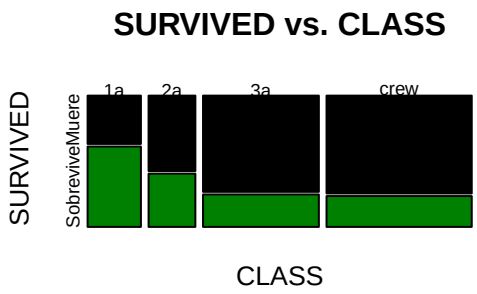
```
CramerV(tabla_SCT)
```

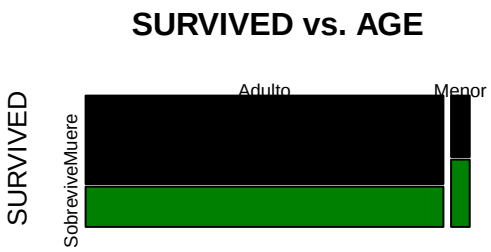
```
## [1] 0.2941201
```

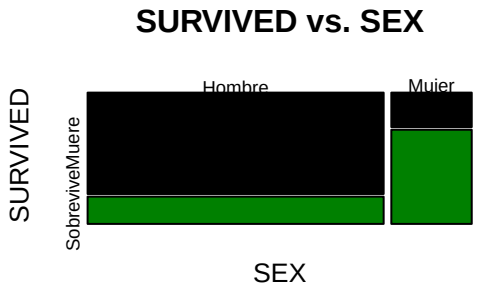
Valores de la V de Cramér (https://en.wikipedia.org/wiki/CramÃlr%27s_V) y Phi (https://en.wikipedia.org/wiki/Phi_coefficient) entre 0.1 y 0.3 nos indican que la asociación estadística es baja, y entre 0.3 y 0.5 se puede considerar una asociación media. Finalmente, si los valores fueran superiores a 0.5 (no es el caso), la asociación estadística entre las variables sería alta. Como se puede apreciar, los valores de Phi y V coinciden. Esto ocurre en el contexto de analizar tablas de contingencia 2x2.

Una alternativa interesante a las barras de diagramas, es el plot de las tablas de contingencia. Obtenemos la misma información pero para algunos receptores puede resultar más visual.

```
par(mfrow=c(2,2))
plot(tabla_SCT, col = c("black","#008000"), main = "SURVIVED vs. CLASS")
plot(tabla_SAT, col = c("black","#008000"), main = "SURVIVED vs. AGE")
plot(tabla_SST, col = c("black","#008000"), main = "SURVIVED vs. SEX")
```




Nuestro objetivo es crear un árbol de decisión que permita analizar qué tipo de pasajero del Titanic tenía probabilidades de sobrevivir o no. Por lo tanto, la variable por la que clasificaremos es el campo de si el pasajero sobrevivió o no. De todas maneras, al imprimir las primeras (con head) y últimas 10 (con tail) filas nos damos cuenta de que los datos están ordenados.

```
head(data,10)
```

```
##    CLASS    AGE    SEX  SURVIVED
## 1     1a Adulto Hombre Sobrevive
## 2     1a Adulto Hombre Sobrevive
## 3     1a Adulto Hombre Sobrevive
## 4     1a Adulto Hombre Sobrevive
## 5     1a Adulto Hombre Sobrevive
## 6     1a Adulto Hombre Sobrevive
## 7     1a Adulto Hombre Sobrevive
## 8     1a Adulto Hombre Sobrevive
## 9     1a Adulto Hombre Sobrevive
## 10    1a Adulto Hombre Sobrevive
```

```
tail(data,10)
```

```
##      CLASS    AGE   SEX  SURVIVED
## 2192  crew Adulto Mujer Sobrevive
## 2193  crew Adulto Mujer Sobrevive
## 2194  crew Adulto Mujer Sobrevive
## 2195  crew Adulto Mujer Sobrevive
## 2196  crew Adulto Mujer Sobrevive
## 2197  crew Adulto Mujer Sobrevive
```

```
## 2198  crew Adulto Mujer Sobrevive
## 2199  crew Adulto Mujer     Muere
## 2200  crew Adulto Mujer     Muere
## 2201  crew Adulto Mujer     Muere
```

## Preparación de los datos para el modelo

Para la futura evaluación del árbol de decisión, es necesario dividir el conjunto de datos en un conjunto de entrenamiento y un conjunto de prueba. El conjunto de entrenamiento es el subconjunto del conjunto original de datos utilizado para construir un primer modelo; y el conjunto de prueba, el subconjunto del conjunto original de datos utilizado para evaluar la calidad del modelo.

Lo más correcto será utilizar un conjunto de datos diferente del que utilizamos para construir el árbol, es decir, un conjunto diferente del de entrenamiento. No hay ninguna proporción fijada con respecto al número relativo de componentes de cada subconjunto, pero la más utilizada acostumbra a ser 2/3 para el conjunto de entrenamiento y 1/3, para el conjunto de prueba.

La variable por la que clasificaremos es el campo de si el pasajero sobrevivió o no, que está en la cuarta columna. De esta forma, tendremos un conjunto de datos para el entrenamiento y uno para la validación

```
set.seed(666)
y <- data[,4]
X <- data[,1:3]
```

De forma dinámica podemos definir una forma de separar los datos en función de un parámetro. Así, definimos un parámetro que controla el split de forma dinámica en el test.

```
split_prop <- 3
indexes = sample(1:nrow(data), size=floor(((split_prop-1)/split_prop)*nrow(data)))
trainX<-X[indexes,]
trainy<-y[indexes]
testX<-X[-indexes,]
testy<-y[-indexes]
```

Después de una extracción aleatoria de casos es altamente recomendable efectuar un análisis de datos mínimo para asegurarnos de no obtener clasificadores sesgados por los valores que contiene cada muestra. En este caso, verificaremos que la proporción del supervivientes es más o menos constante en los dos conjuntos:

```
summary(trainX);
```

```
##   CLASS         AGE           SEX
##  1a  :208   Adulto:1395   Hombre:1153
##  2a  :185   Menor :  72   Mujer : 314
##  3a  :477
##  crew:597
```

```
summary(trainy)
```

```
##     Muere Sobrevive
##       997       470
```

```
summary(testX)
```

```
##   CLASS         AGE          SEX
##  1a  :117   Adulto:697   Hombre:578
##  2a  :100   Menor : 37   Mujer :156
##  3a  :229
##  crew:288
```

```r
summary(testy)
```

```
##     Muere Sobrevive
##       493       241
```

Verificamos fácilmente que no hay diferencias graves que puedan sesgar las conclusiones.

## Creación del modelo, calidad del modelo y extracción de reglas

Se crea el árbol de decisión usando los datos de entrenamiento (no hay que olvidar que la variable outcome es de tipo factor):

```r
trainy <-  as.factor(trainy)
model <- C50::C5.0(trainX, trainy,rules=TRUE )
summary(model)
```

```
##
## Call:
## C5.0.default(x = trainX, y = trainy, rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Sun Dec 29 18:25:14 2024
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 1467 cases (4 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (1153/243, lift 1.2)
##  SEX = Hombre
##  ->  class Muere  [0.789]
##
## Rule 2: (477/123, lift 1.1)
##  CLASS = 3a
##  ->  class Muere  [0.741]
##
## Rule 3: (178/15, lift 2.8)
##  CLASS in {1a, 2a, crew}
##  SEX = Mujer
##  ->  class Sobrevive  [0.911]
##
## Default class: Muere
##
##
## Evaluation on training data (1467 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##       3  322(21.9%)   <<
##
##
```

```
##     (a)   (b)    <-classified as
##    ----  ----
##     982    15    (a): class Muere
##     307   163    (b): class Sobrevive
##
##
##  Attribute usage:
##
##   90.73% SEX
##   44.65% CLASS
##
##
## Time: 0.0 secs
```

Errors muestra el número y porcentaje de casos mal clasificados en el subconjunto de entrenamiento. El árbol obtenido clasifica erróneamente 322 de los 1467 casos dados, una tasa de error del 21.9%.

A partir del árbol de decisión de dos hojas que hemos modelado, se pueden extraer las siguientes reglas de decisión (gracias a rules=TRUE podemos imprimir las reglas directamente):

SEX = "Hombre" → Muere. Validez: 78,9%

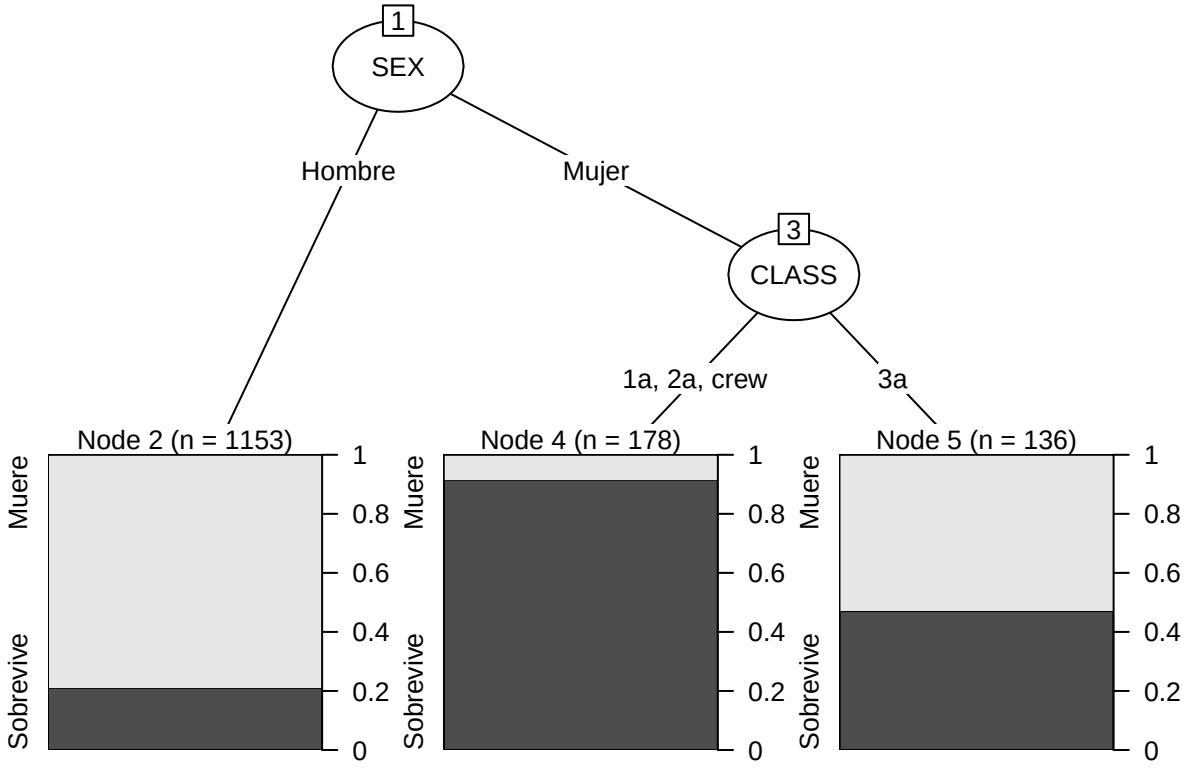
CLASS "3ª" → Muere. Validez: 74,1%

CLASS "1ª", "2ª", "crew" y SEX = "Mujer" → Sobrevive. Validez: 91,1%

Por tanto, podemos concluir que el conocimiento extraído y cruzado con el análisis visual se resume en "las mujeres y los niños primero a excepción de que fueras de 3ª clase".

A continuación, mostramos el árbol obtenido.

```
model <- C50::C5.0(trainX, trainy)
plot(model,gp = gpar(fontsize = 9.5))
```




## Validación del modelo con los datos reservados

Una vez tenemos el modelo, podemos comprobar su calidad prediciendo la clase para los datos de prueba que nos hemos reservado al principio.

```
predicted_model <- predict( model, testX, type="class" )
print(sprintf("La precisión del árbol es: %.4f %%",100*sum(predicted_model == testy) / length(predicted
```

```
## [1] "La precisión del árbol es: 78.8828 %"
```

Cuando hay pocas clases, la calidad de la predicción se puede analizar mediante una matriz de confusión que identifica los tipos de errores cometidos.

```
mat_conf<-table(testy,Predicted=predicted_model)
mat_conf
```

```
##            Predicted
## testy       Muere Sobrevive
##   Muere       488         5
##   Sobrevive   150        91
```

Otra manera de calcular el porcentaje de registros correctamente clasificados usando la matriz de confusión:

```
porcentaje_correct<-100 * sum(diag(mat_conf)) / sum(mat_conf)
print(sprintf("El %% de registros correctamente clasificados es: %.4f %%",porcentaje_correct))
```

```
## [1] "El % de registros correctamente clasificados es: 78.8828 %"
```

Además, tenemos a nuestra disposición el paquete gmodels para obtener información más completa:

```
if(!require(gmodels)){
    install.packages('gmodels', repos='http://cran.us.r-project.org')
    library(gmodels)
}
```

```
## Loading required package: gmodels
```

```
## Registered S3 method overwritten by 'gdata':
##   method         from
##   reorder.factor DescTools
```

```
CrossTable(testy, predicted_model,prop.chisq  = FALSE, prop.c = FALSE, prop.r =FALSE,dnn = c('Reality',
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  734
##
##
##              | Prediction
##      Reality |     Muere | Sobrevive | Row Total |
## -------------|-----------|-----------|-----------|
##        Muere |       488 |         5 |       493 |
##              |     0.665 |     0.007 |           |
## -------------|-----------|-----------|-----------|
##    Sobrevive |       150 |        91 |       241 |
##              |     0.204 |     0.124 |           |
## -------------|-----------|-----------|-----------|
## Column Total |       638 |        96 |       734 |
## -------------|-----------|-----------|-----------|
##
##
```

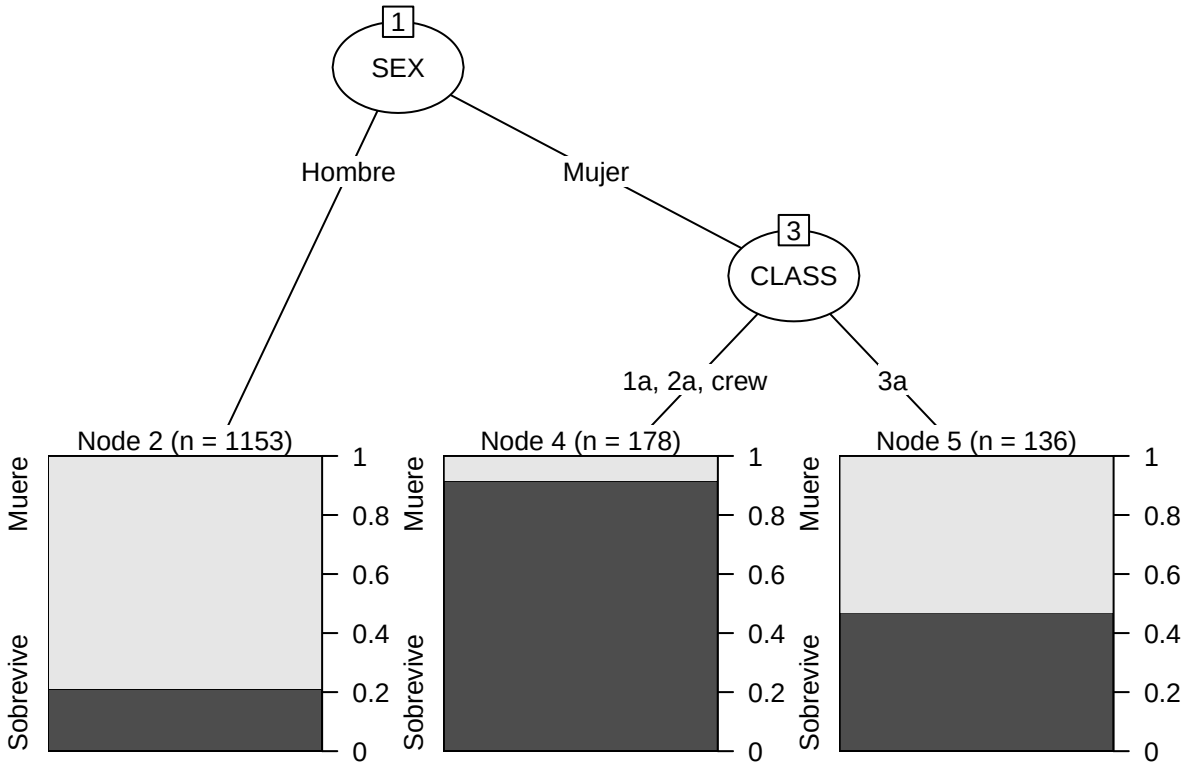
## Prueba con una variación u otro enfoque algorítmico

### Variaciones del paquete C5.0

En este apartado buscaremos probar con las variaciones que nos ofrece el paquete C5.0 para analizar cómo afectan a la creación de los árboles generados. Existen muchas posibles variaciones con otras funciones que podéis investigar. La idea es seguir con el enfoque de árboles de decisión explorando posibles opciones. Una vez tengamos un método alternativo, debemos analizar cómo se modifica el árbol y cómo afecta a la capacidad predictiva en el conjunto de test.

A continuación, utilizamos otro enfoque para comparar los resultados: incorpora como novedad "adaptative boosting", basado en el trabajo Rob Schapire and Yoav Freund (1999). La idea de esta técnica es generar varios clasificadores, con sus correspondientes arboles de decisión y su ser de reglas. Cuando un nuevo caso va a ser clasificado, cada clasificador vota cual es la clase predicha. Los votos son sumados y determina la clase final.

```
modelo2 <- C50::C5.0(trainX, trainy, trials = 10)
plot(modelo2,gp = gpar(fontsize = 9.5))
```




En este caso, dada la simplicidad del conjunto de ejemplo, no se aprecian diferencias, pero aparecerán en datos de mayor complejidad y modificando el parámetro "trials" se puede intentar mejorar los resultados.

Vemos a continuación cómo son las predicciones del nuevo árbol:

```
predicted_model2 <- predict( modelo2, testX, type="class" )
print(sprintf("La precisión del árbol es: %.4f %%",100*sum(predicted_model2 == testy) / length(predicte
```

```
## [1] "La precisión del árbol es: 75.0681 %"
```

Observamos como se modifica levemente la precisión del modelo a mejor.

```
mat_conf<-table(testy,Predicted=predicted_model2)
mat_conf
```

```
##            Predicted
## testy       Muere Sobrevive
##   Muere       438        55
##   Sobrevive   128       113
```

Otra manera de calcular el porcentaje de registros correctamente clasificados usando la matriz de confusión:

```
porcentaje_correct<-100 * sum(diag(mat_conf)) / sum(mat_conf)
print(sprintf("El %% de registros correctamente clasificados es: %.4f %%",porcentaje_correct))
```

```
## [1] "El % de registros correctamente clasificados es: 75.0681 %"
```

El algoritmo C5.0 incorpora algunas opciones para ver la importancia de las variables (ver documentación para los detalles entre los dos métodos):

```
importancia_usage <- C50::C5imp(modelo2, metric = "usage")
importancia_splits <- C50::C5imp(modelo2, metric = "splits")
importancia_usage
```

```
##       Overall
## CLASS  100.00
## SEX    100.00
## AGE     93.73
```

```
importancia_splits
```

```
##       Overall
## CLASS      40
## SEX        40
## AGE        20
```

Curiosamente y aunque el conjunto de datos es muy sencillo, se aprecian diferencias en los métodos de importancia de las variables. Se recomienda en vuestro ejercicio mejorar la visualización de los resultados con la función ggplo2 o similar.

### Interpretación de las variables en las predicciones.

Nos interesa saber para las predicciones que variable son las que tienen más influencia. Así, probaremos con un enfoque algorítmico de Random Forest y obtendremos métricas de interpretabilidad con la librería IML (https://cran.r-project.org/web/packages/iml/iml.pdf). As:

```
if(!require(randomForest)){
  install.packages('randomForest',repos='http://cran.us.r-project.org')
  library(randomForest)
}
```

```
## Loading required package: randomForest
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
if(!require(iml)){
  install.packages('iml', repos='http://cran.us.r-project.org')
  library(iml)
}
```

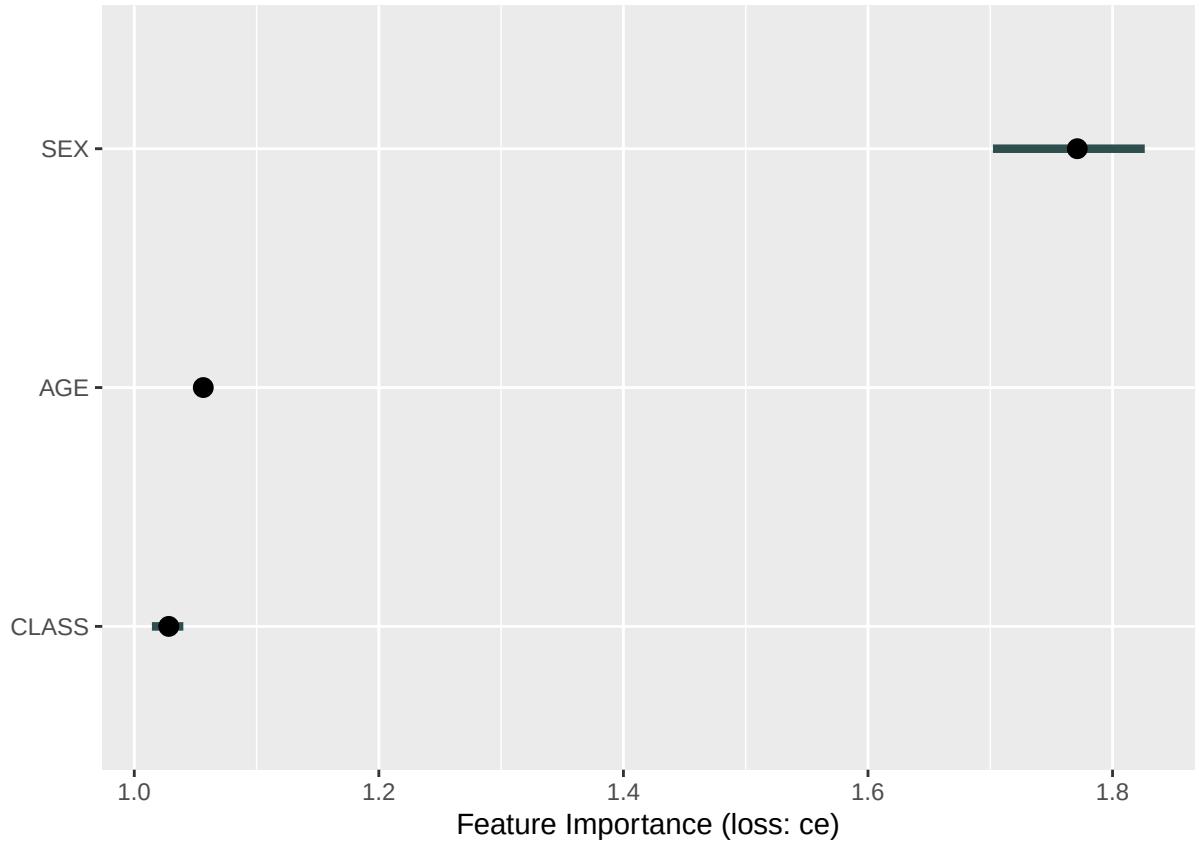
```
## Loading required package: iml
```

Empezamos ejecutado un Random Forest:

```
train.data <- as.data.frame(cbind(trainX,trainy))
colnames(train.data)[4] <- "SURVIVED"
rf <-  randomForest(SURVIVED ~ ., data = train.data, ntree = 50)
```

Podemos medir y graficar la importancia de cada variable para las predicciones del random forest con *FeatureImp*. La medida se basa funciones de pérdida de rendimiento que en nuestro caso será con el objetivo de clasificación ("ce").

```
X <- train.data[which(names(train.data) != "SURVIVED")]
predictor <- Predictor$new(rf, data = X, y = train.data$SURVIVED)
imp <- FeatureImp$new(predictor, loss = "ce")
plot(imp)
```




```
imp$results
```

```
##   feature importance.05 importance importance.95 permutation.error
## 1     SEX      1.702194   1.771160      1.826332         0.3851397
## 2     AGE      1.053292   1.056426      1.062069         0.2297205
## 3   CLASS      1.014420   1.028213      1.040125         0.2235855
```

Adicionalmente, podemos también dibujar los efectos locales acumulados (ALE) de la variable usando la libreria *patchwork*:

```
if(!require(patchwork)){
    install.packages('patchwork',repos='http://cran.us.r-project.org')
    library(patchwork)
}
```

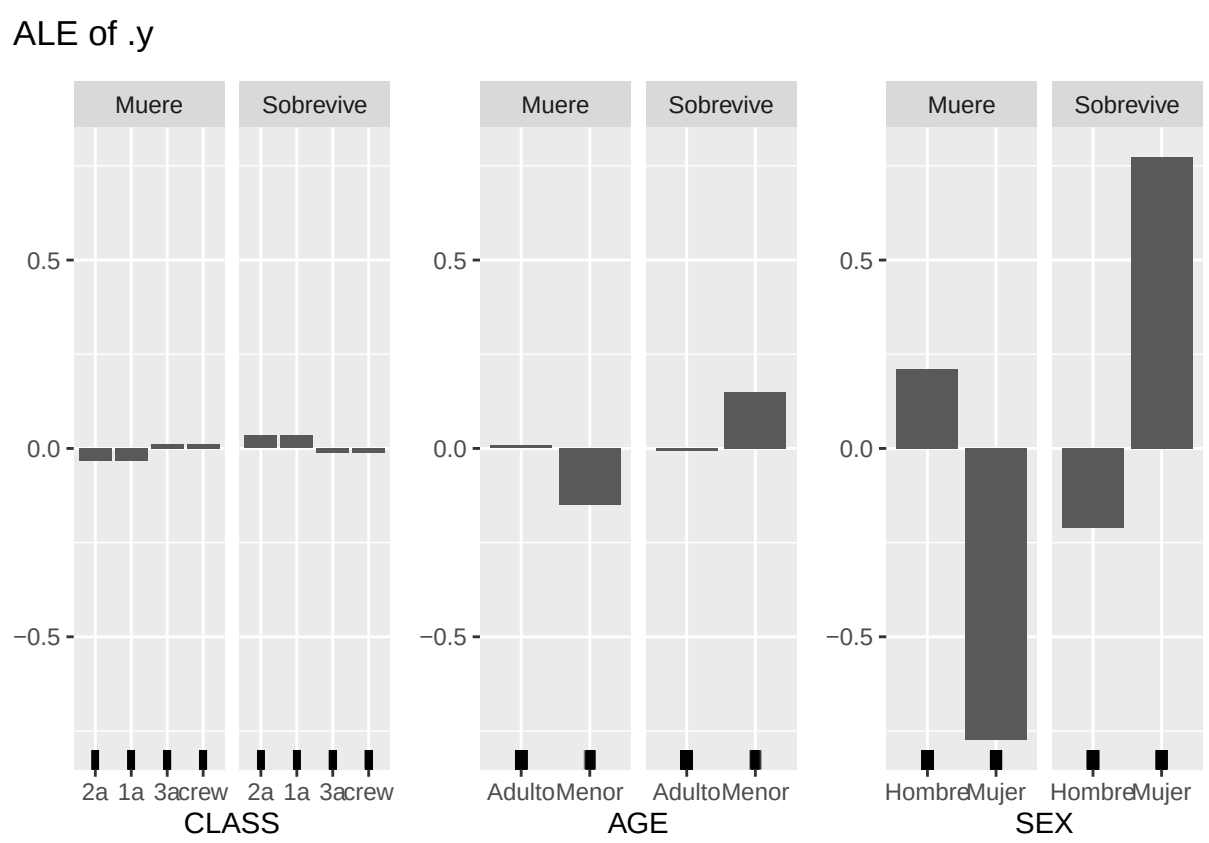
```
## Loading required package: patchwork
```

```
##
## Attaching package: 'patchwork'
```

```
## The following object is masked from 'package:MASS':
```

```
##
##     area
```

```
effs <- FeatureEffects$new(predictor)
plot(effs)
```




Como podemos ver, el género es la variable con más importancia para las predicciones, siendo las mujeres mucho más propensas a sobrevivir. Nota: Se espera que los alumnos profundicen en la función de cara a la resolución de los ejercicios.

## Enunciado del ejercicio

Para el conjunto de datos German Credit, los alumnos deben completar aquí la solución a la PEC3 que consiste de los siguientes apartados. Notad que se detalla el contenido necesario para cada apartado en la Sección 4 (Rúbrica).

Se debe entregar la PEC en el buzón de entregas del aula, como en las anteriores PECs.

### Realizar un primer análisis descriptivo y de correlaciones. Es importante en este apartado entender bien los datos antes de seguir con los análisis posteriores. Lista todo lo que te haya sorprendido de los datos

```
data<-read.csv("./credit.csv",header=T,sep=",")
summary(data)
```

```
##  checking_balance   months_loan_duration credit_history       purpose
```

```
##  Length:1000        Min.   : 4.0         Length:1000        Length:1000
##  Class :character   1st Qu.:12.0         Class :character   Class :character
##  Mode  :character   Median :18.0         Mode  :character   Mode  :character
##                     Mean   :20.9
##                     3rd Qu.:24.0
##                     Max.   :72.0
##      amount      savings_balance    employment_length  installment_rate
##  Min.   :  250   Length:1000        Length:1000        Min.   :1.000
##  1st Qu.: 1366   Class :character   Class :character   1st Qu.:2.000
##  Median : 2320   Mode  :character   Mode  :character   Median :3.000
##  Mean   : 3271                                         Mean   :2.973
##  3rd Qu.: 3972                                         3rd Qu.:4.000
##  Max.   :18424                                         Max.   :4.000
##  personal_status    other_debtors      residence_history   property
##  Length:1000        Length:1000        Min.   :1.000     Length:1000
##  Class :character   Class :character   1st Qu.:2.000     Class :character
##  Mode  :character   Mode  :character   Median :3.000     Mode  :character
##                                        Mean   :2.845
##                                        3rd Qu.:4.000
##                                        Max.   :4.000
##       age        installment_plan     housing          existing_credits
##  Min.   :19.00   Length:1000        Length:1000        Min.   :1.000
##  1st Qu.:27.00   Class :character   Class :character   1st Qu.:1.000
##  Median :33.00   Mode  :character   Mode  :character   Median :1.000
##  Mean   :35.55                                         Mean   :1.407
##  3rd Qu.:42.00                                         3rd Qu.:2.000
##  Max.   :75.00                                         Max.   :4.000
##     default      dependents     telephone         foreign_worker
##  Min.   :1.0   Min.   :1.000   Length:1000        Length:1000
##  1st Qu.:1.0   1st Qu.:1.000   Class :character   Class :character
##  Median :1.0   Median :1.000   Mode  :character   Mode  :character
##  Mean   :1.3   Mean   :1.155
##  3rd Qu.:2.0   3rd Qu.:1.000
##  Max.   :2.0   Max.   :2.000
##      job
##  Length:1000
##  Class :character
##  Mode  :character
##
##
##
```

```
head(data)
```

```
##   checking_balance months_loan_duration credit_history   purpose amount
## 1           < 0 DM                    6       critical  radio/tv   1169
## 2       1 - 200 DM                   48         repaid  radio/tv   5951
## 3          unknown                   12       critical education   2096
## 4           < 0 DM                   42         repaid furniture   7882
## 5           < 0 DM                   24        delayed car (new)   4870
## 6          unknown                   36         repaid education   9055
##   savings_balance employment_length installment_rate personal_status
## 1         unknown           > 7 yrs                4     single male
## 2        < 100 DM         1 - 4 yrs                2          female
## 3        < 100 DM         4 - 7 yrs                2     single male
```

```
## 4        < 100 DM         4 - 7 yrs               2     single male
## 5        < 100 DM         1 - 4 yrs               3     single male
## 6         unknown         1 - 4 yrs               2     single male
##   other_debtors residence_history               property age installment_plan
## 1          none                 4            real estate  67             none
## 2          none                 2            real estate  22             none
## 3          none                 3            real estate  49             none
## 4     guarantor                 4 building society savings  45             none
## 5          none                 4            unknown/none  53             none
## 6          none                 4            unknown/none  35             none
##    housing existing_credits default dependents telephone foreign_worker
## 1      own                2       1          1       yes            yes
## 2      own                1       2          1      none            yes
## 3      own                1       1          2      none            yes
## 4 for free                1       1          2      none            yes
## 5 for free                2       2          2      none            yes
## 6 for free                1       1          2       yes            yes
##                  job
## 1   skilled employee
## 2   skilled employee
## 3 unskilled resident
## 4   skilled employee
## 5   skilled employee
## 6 unskilled resident
```

#La descricion de las columnas es: **checking_balance:** Cuenta corriente, con valores "unknown", "1 - 200 DM", "no checking", "> 200 DM", entre otros, entiendo yo que DM es Deutch Mark, son caracteres y tendríamos que convertirlo a factor. **months_loan_duration:** Duración del préstamo en meses, numérico. **credit_history:** Historial crediticio, con valores "delayed", "critical","repaid", entre otros, entiendo que es factor. **purpose:** Propósito préstamo, con valores "radio/tv", "education", "furniture", "car", ... entre otros, entiendo que es factor también. **amount:** Monto del préstamo, numérico. **savings_balance:** Saldo de ahorros, con valores "unknown", "no known savings", "500 - 1000 DM", "100 - 500 DM", entre otros, entiendo que es factor. **employment_length:** Duración del empleo, con valores "1 - 4 years", "4 - 7 years", "unemployed", "less than 1 year", entre otros, entiendo que es factor también. Siempre podemos sacar el promedio para valores numéricos factorizados. **installment_rate:** Porcentaje de ingresos totales en pago a plazos, numérico. **personal_status:** Estado civil, con valores como "single male", "female", etc **other_debtors:** Otros deudores, con valores como "none", "guarantor", "co-applicant", entiendo que es factor. **residence_history:** Historial de residencia, numérico. **property:** Propiedad, con valores como "real estate", "unknown/none", "building society savings", "other", entiendo que es factor. **age:** Edad, numérico. **installment_plan:** Plan de pago a plazos, con valores como "none", "bank", "stores", entiendo que es factor. **housing:** Tipo de vivienda, con valores como "own", "for free", "rent", entiendo que es factor. **existing_credits:** Créditos existentes, numérico. **default:** Incumplimiento, con valores "yes" y "no", entiendo que es la variable a predecir. **dependents:** Dependientes, numérico. **telephone:** Teléfono, con valores "yes" y "none", entiendo que es factor. **foreign_worker:** Trabajador extranjero, con valores "yes" y "no", entiendo que es factor. **job:** Trabajo, con valores "skilled employee", "unskilled resident", "mangement self-employed", etc... es factor

```
dim(data)
```

```
## [1] 1000   21
```

```
str(data)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ checking_balance    : chr  "< 0 DM" "1 - 200 DM" "unknown" "< 0 DM" ...
##  $ months_loan_duration: int  6 48 12 42 24 36 24 36 12 30 ...
```

```
##  $ credit_history      : chr  "critical" "repaid" "critical" "repaid" ...
##  $ purpose             : chr  "radio/tv" "radio/tv" "education" "furniture" ...
##  $ amount              : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ savings_balance     : chr  "unknown" "< 100 DM" "< 100 DM" "< 100 DM" ...
##  $ employment_length   : chr  "> 7 yrs" "1 - 4 yrs" "4 - 7 yrs" "4 - 7 yrs" ...
##  $ installment_rate    : int  4 2 2 2 3 2 3 2 2 4 ...
##  $ personal_status     : chr  "single male" "female" "single male" "single male" ...
##  $ other_debtors       : chr  "none" "none" "none" "guarantor" ...
##  $ residence_history   : int  4 2 3 4 4 4 4 2 4 2 ...
##  $ property            : chr  "real estate" "real estate" "real estate" "building society savings" .
##  $ age                 : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ installment_plan    : chr  "none" "none" "none" "none" ...
##  $ housing             : chr  "own" "own" "own" "for free" ...
##  $ existing_credits    : int  2 1 1 1 2 1 1 1 1 2 ...
##  $ default             : int  1 2 1 1 2 1 1 1 1 2 ...
##  $ dependents          : int  1 1 2 2 2 2 1 1 1 1 ...
##  $ telephone           : chr  "yes" "none" "none" "none" ...
##  $ foreign_worker      : chr  "yes" "yes" "yes" "yes" ...
##  $ job                 : chr  "skilled employee" "skilled employee" "unskilled resident" "skilled em
```

```
data[is.na(data),]
```

```
##  [1] checking_balance     months_loan_duration credit_history
##  [4] purpose              amount               savings_balance
##  [7] employment_length    installment_rate     personal_status
## [10] other_debtors        residence_history    property
## [13] age                  installment_plan     housing
## [16] existing_credits     default              dependents
## [19] telephone            foreign_worker       job
## <0 rows> (or 0-length row.names)
```

No hay nulos, entonces no hago nada al respecto

```
sapply(data[, sapply(data, is.integer)], function(x) length(unique(x)))
```

```
## months_loan_duration               amount     installment_rate
##                   33                  921                    4
##    residence_history                  age     existing_credits
##                    4                   53                    4
##              default           dependents
##                    2                    2
```

Veo que las variables numericas amount, months_loan_duration y age deberian quedar como numericas y el resto podemos hacerlas factor.

```
data[] <- lapply(data, function(x) if (is.character(x)) as.factor(x) else x)
data[["installment_rate"]] <- as.factor(data[["installment_rate"]])
data[["residence_history"]] <- as.factor(data[["residence_history"]])
data[["existing_credits"]] <- as.factor(data[["existing_credits"]])
data[["dependents"]] <- as.factor(data[["dependents"]])
data[["default"]] <- as.factor(data[["default"]])
str(data)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ checking_balance    : Factor w/ 4 levels "< 0 DM","> 200 DM",..: 1 3 4 1 1 4 4 3 4 3 ...
##  $ months_loan_duration: int  6 48 12 42 24 36 24 36 12 30 ...
##  $ credit_history      : Factor w/ 5 levels "critical","delayed",..: 1 5 1 5 2 5 5 5 5 1 ...
```

```
##  $ purpose             : Factor w/ 10 levels "business","car (new)",..: 8 8 5 6 2 5 6 3 8 2 ...
##  $ amount              : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ savings_balance     : Factor w/ 5 levels "< 100 DM","> 1000 DM",..: 5 1 1 1 1 5 4 1 2 1 ...
##  $ employment_length   : Factor w/ 5 levels "> 7 yrs","0 - 1 yrs",..: 1 3 4 4 3 3 1 3 4 5 ...
##  $ installment_rate    : Factor w/ 4 levels "1","2","3","4": 4 2 2 2 3 2 3 2 2 4 ...
##  $ personal_status     : Factor w/ 4 levels "divorced male",..: 4 2 4 4 4 4 4 4 1 3 ...
##  $ other_debtors       : Factor w/ 3 levels "co-applicant",..: 3 3 3 2 3 3 3 3 3 3 ...
##  $ residence_history   : Factor w/ 4 levels "1","2","3","4": 4 2 3 4 4 4 4 2 4 2 ...
##  $ property            : Factor w/ 4 levels "building society savings",..: 3 3 3 1 4 4 1 2 3 2 ...
##  $ age                 : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ installment_plan    : Factor w/ 3 levels "bank","none",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ housing             : Factor w/ 3 levels "for free","own",..: 2 2 2 1 1 1 2 3 2 2 ...
##  $ existing_credits    : Factor w/ 4 levels "1","2","3","4": 2 1 1 1 2 1 1 1 1 2 ...
##  $ default             : Factor w/ 2 levels "1","2": 1 2 1 1 2 1 1 1 1 2 ...
##  $ dependents          : Factor w/ 2 levels "1","2": 1 1 2 2 2 2 1 1 1 1 ...
##  $ telephone           : Factor w/ 2 levels "none","yes": 2 1 1 1 1 2 1 2 1 1 ...
##  $ foreign_worker      : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ job                 : Factor w/ 4 levels "mangement self-employed",..: 2 2 4 2 2 4 2 1 4 1 ...
```

## Ahora realizo un análisis de correlaciones entre las variables.

#¿Hay alguna variable que se pueda eliminar por estar muy correlacionada con otra?

```
correlacion_matrix <- cor(data[sapply(data, is.numeric)])
print("Matriz de correlación:")
```

```
## [1] "Matriz de correlación:"
```

```
print(correlacion_matrix)
```

```
##                       months_loan_duration     amount         age
## months_loan_duration           1.00000000 0.62498420 -0.03613637
## amount                         0.62498420 1.00000000  0.03271642
## age                           -0.03613637 0.03271642  1.00000000
```

```
top_corr <- correlacion_matrix
top_corr[abs(top_corr) <= 0.2] <- NA
print("Correlaciones top:")
```

```
## [1] "Correlaciones top:"
```

```
print(top_corr)
```

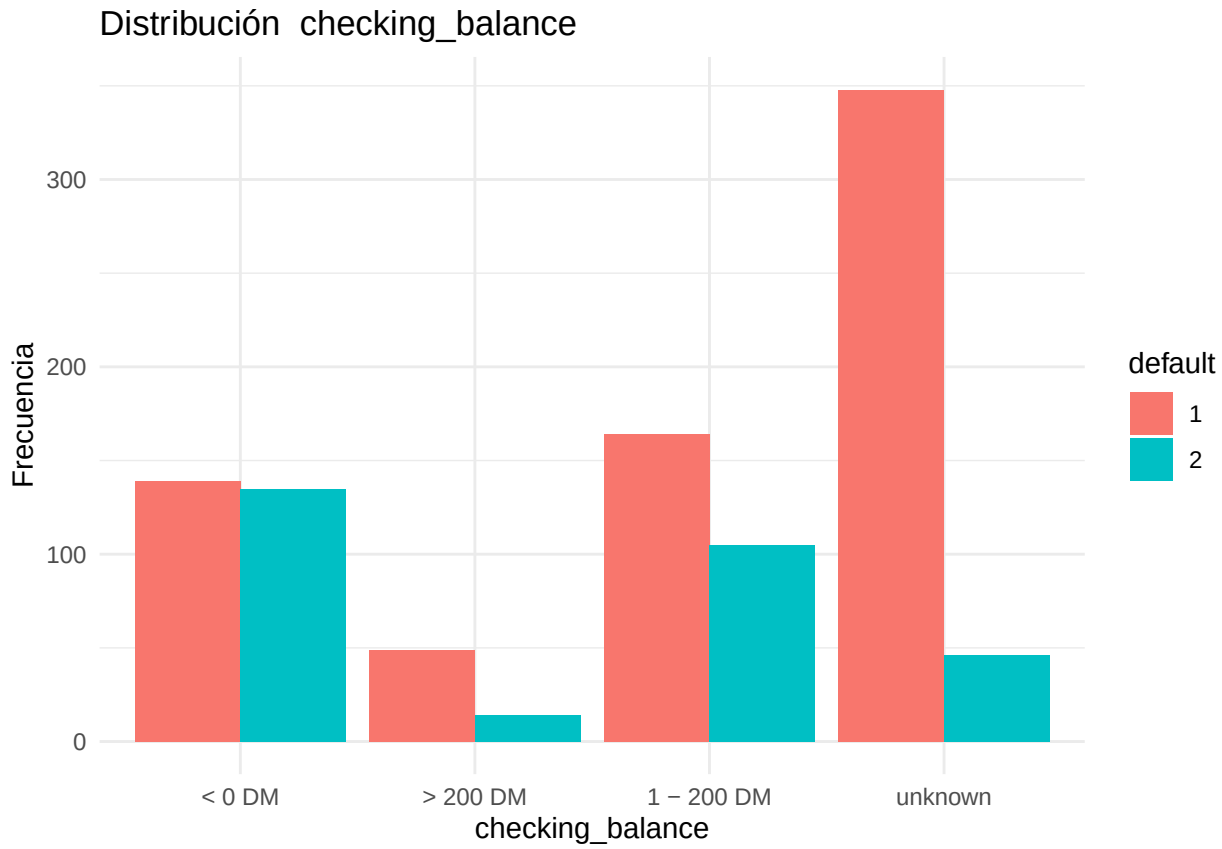
```
##                       months_loan_duration    amount age
## months_loan_duration            1.0000000 0.6249842  NA
## amount                          0.6249842 1.0000000  NA
## age                                    NA        NA   1
```

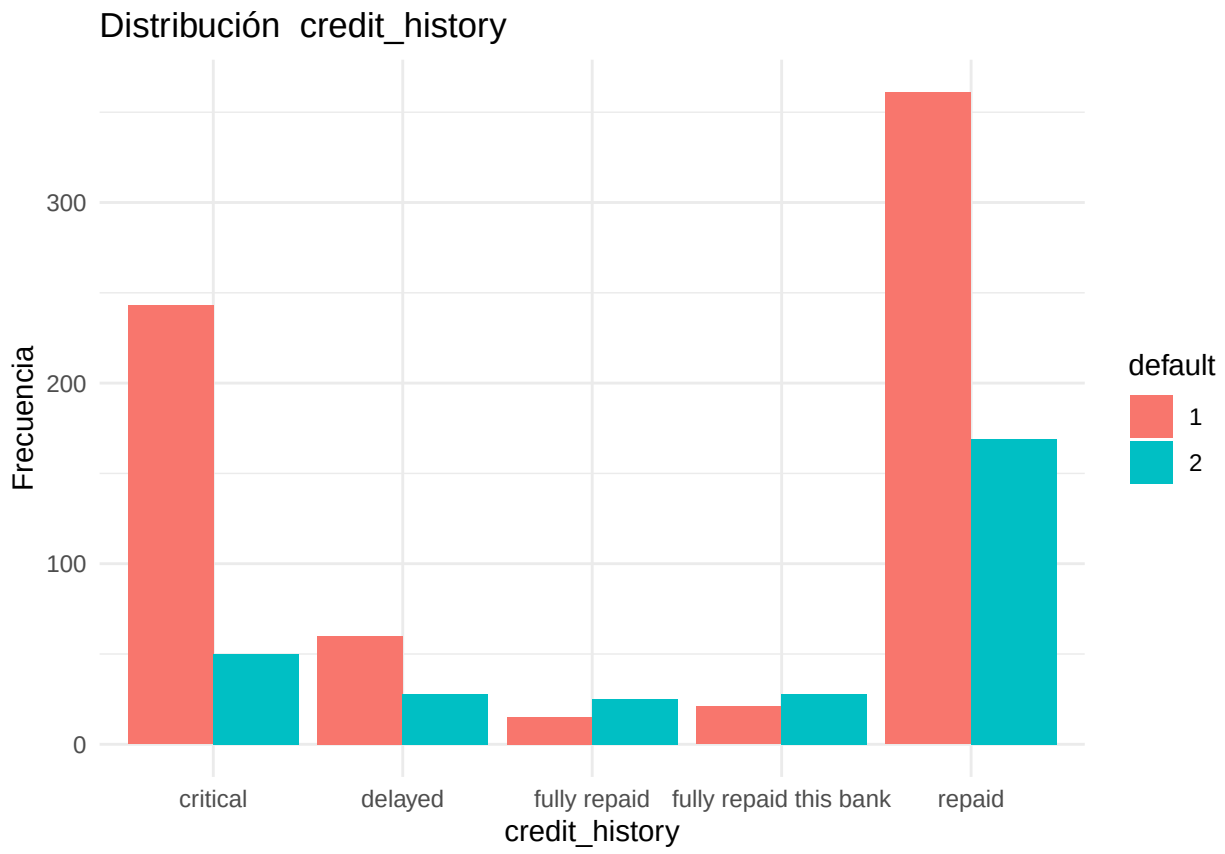
Quizás se podría eliminar la variable months_loan_duration, o la variable amount, aunque lo dejare asi. Si bien no están totalmente correlacionadas, si tienen una correlación importante del 0.6249 entre ellas.

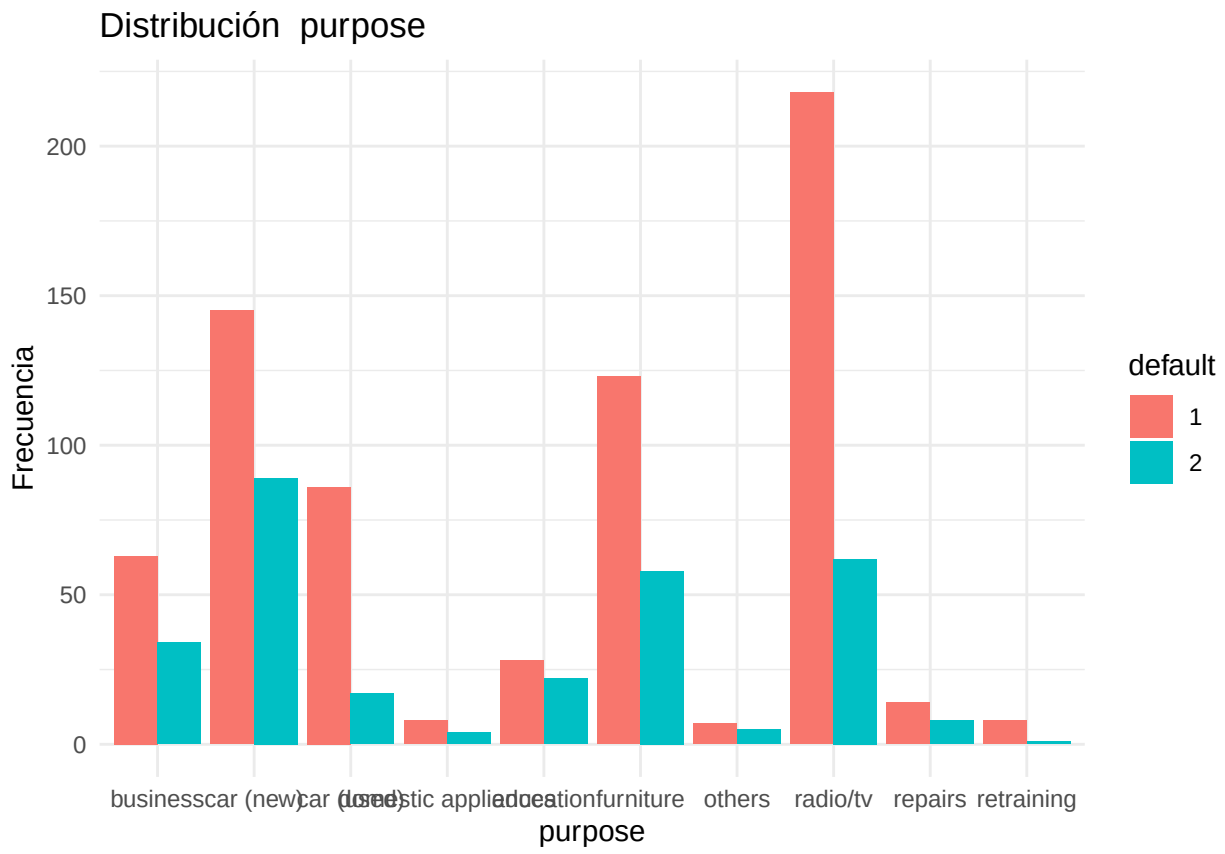
```
columna_objetivo <- "default"

categoricas <- names(data)[sapply(data, is.factor)]
numericas <- names(data)[sapply(data, is.numeric)]

for (var_cat in categoricas) {
```

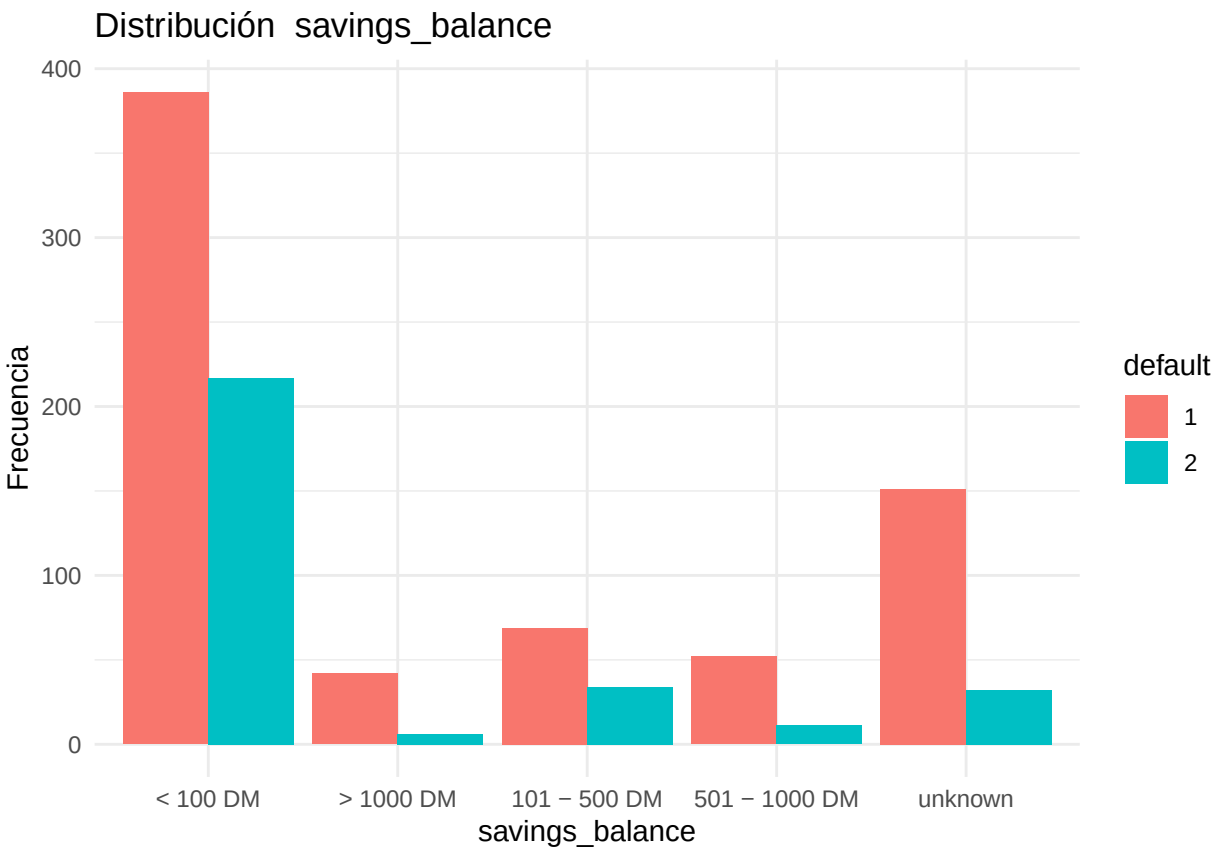
```
  print(
    ggplot(data, aes_string(x = var_cat, fill = columna_objetivo)) +
      geom_bar(position = "dodge") +
      labs(title = paste("Distribución ", var_cat), x = var_cat, y = "Frecuencia") +
      theme_minimal()
  )
}
```

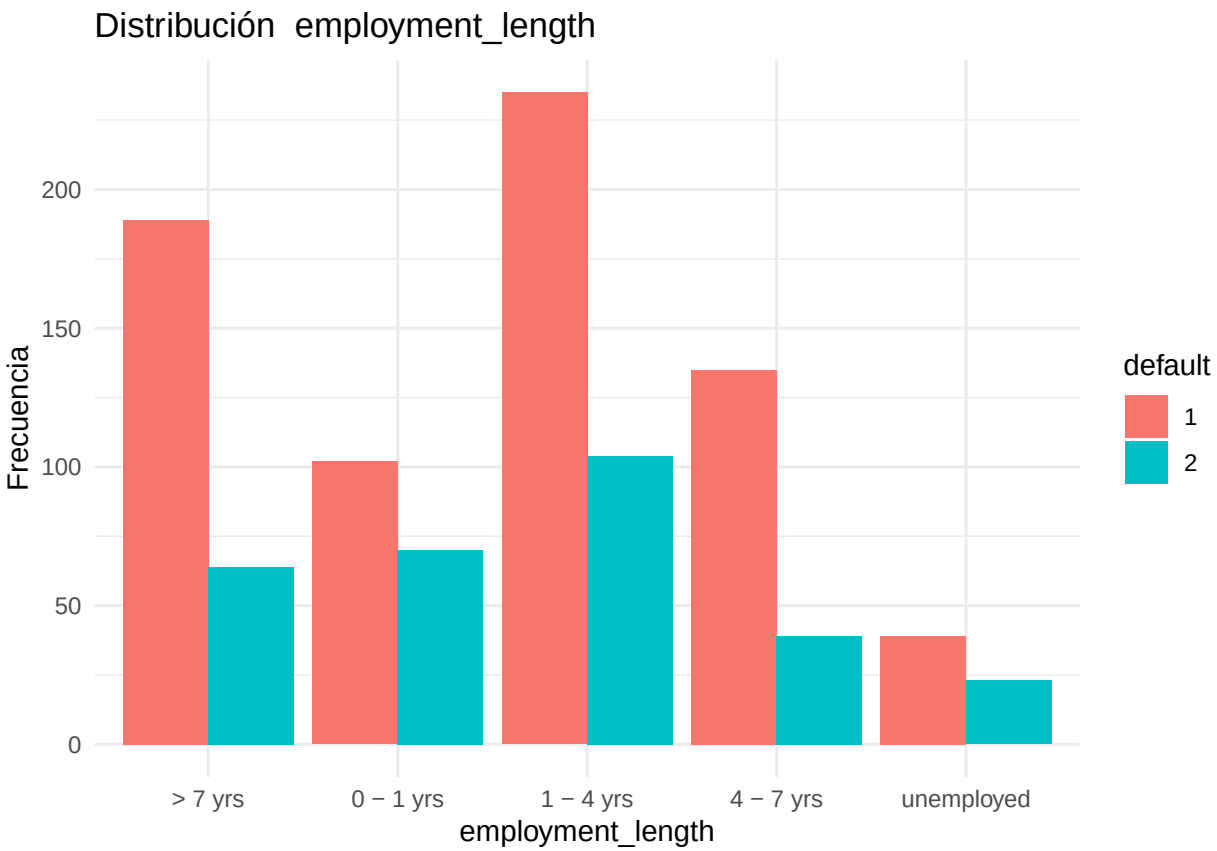
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

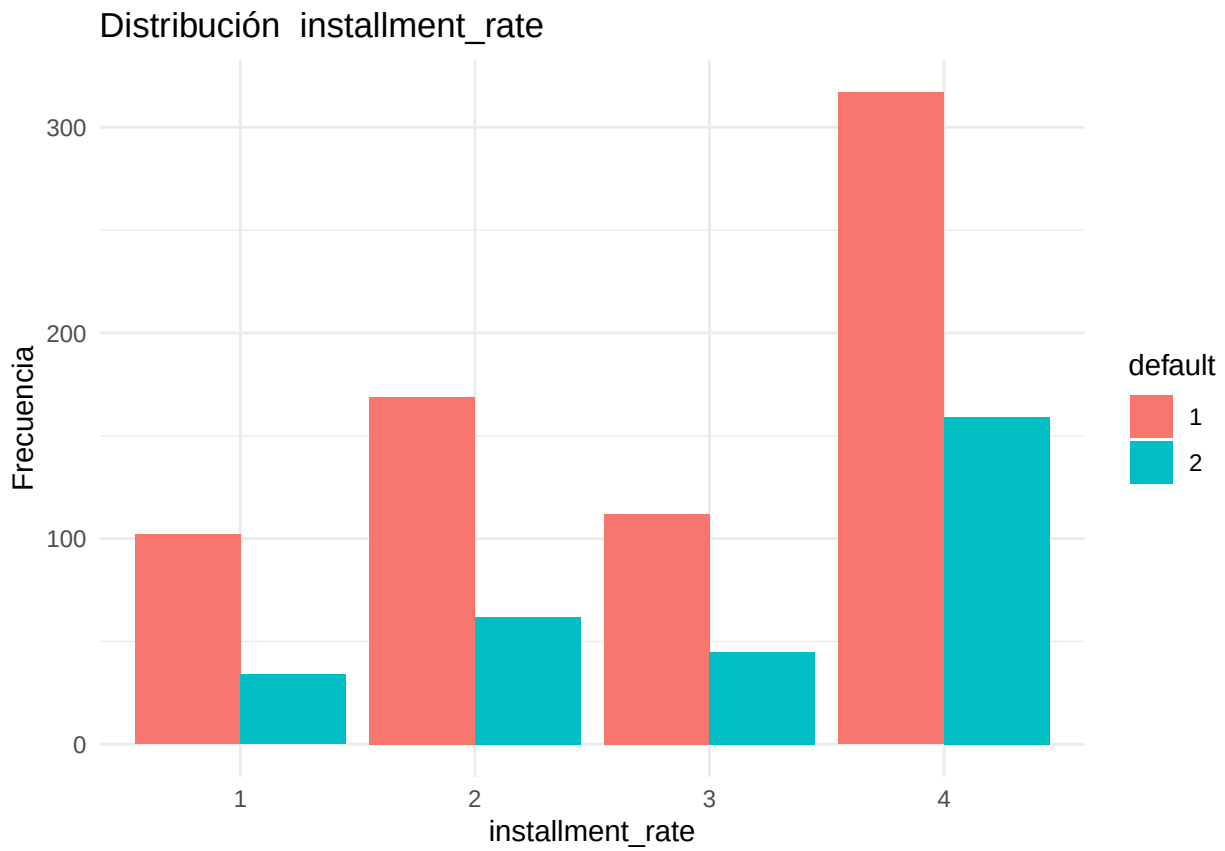


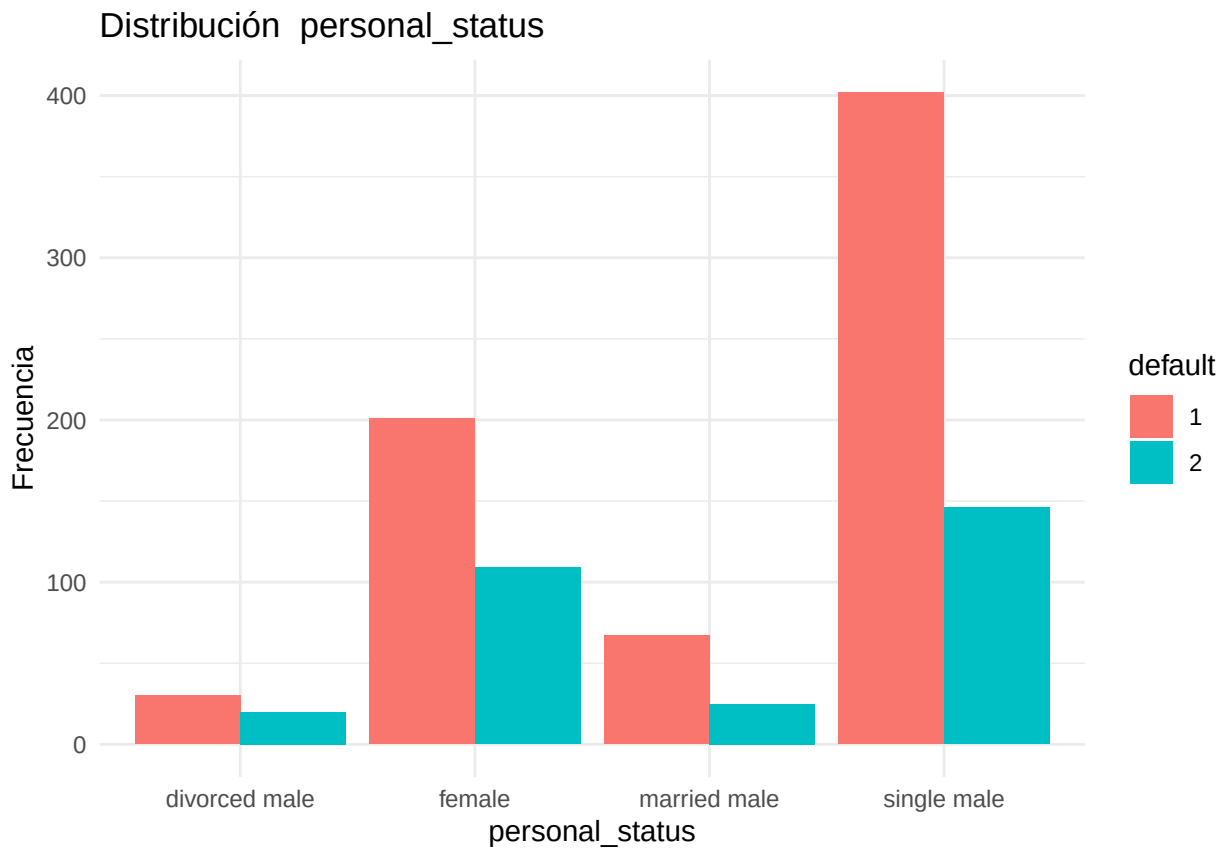

Distribución credit_history
Frecuencia
300
200
100
0
critical
delayed
fully repaid
fully repaid this bank
repaid
credit_history
default
1
2

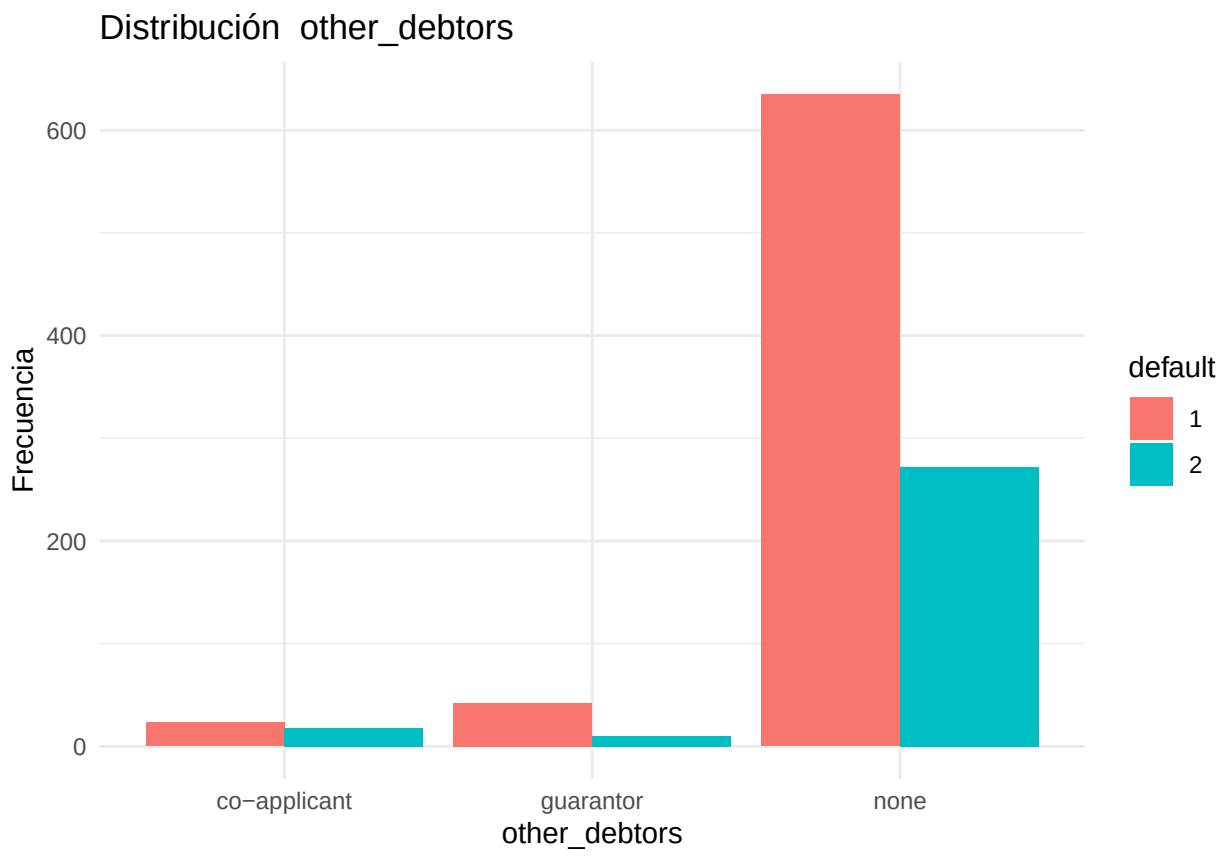

Distribución purpose
Frecuencia
200
150
100
50
0
business
car (new)
car (used)
domestic appliance
education
furniture
others
radio/tv
repairs
retraining
purpose
default
1
2

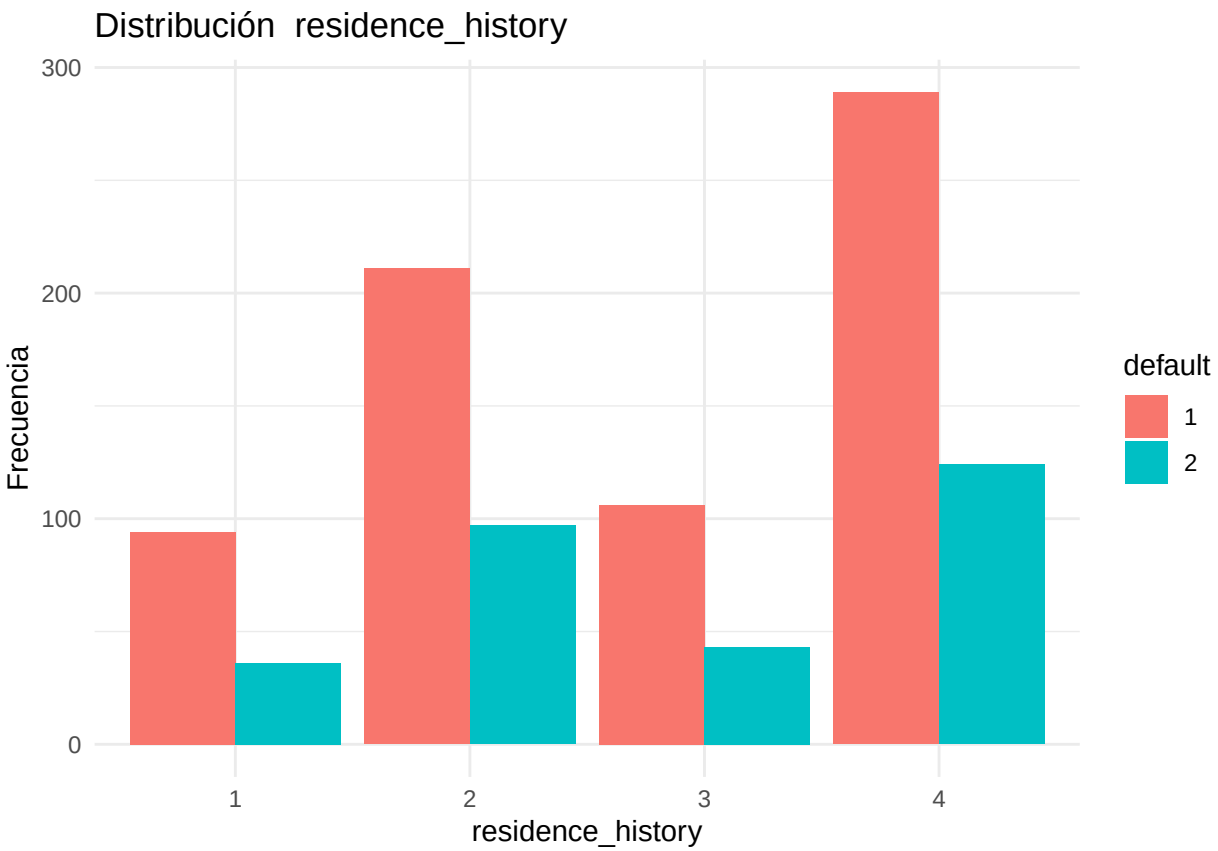

Distribución savings_balance
400
300
200
100
0
Frecuencia
< 100 DM
> 1000 DM
101 − 500 DM
501 − 1000 DM
unknown
savings_balance
default
1
2

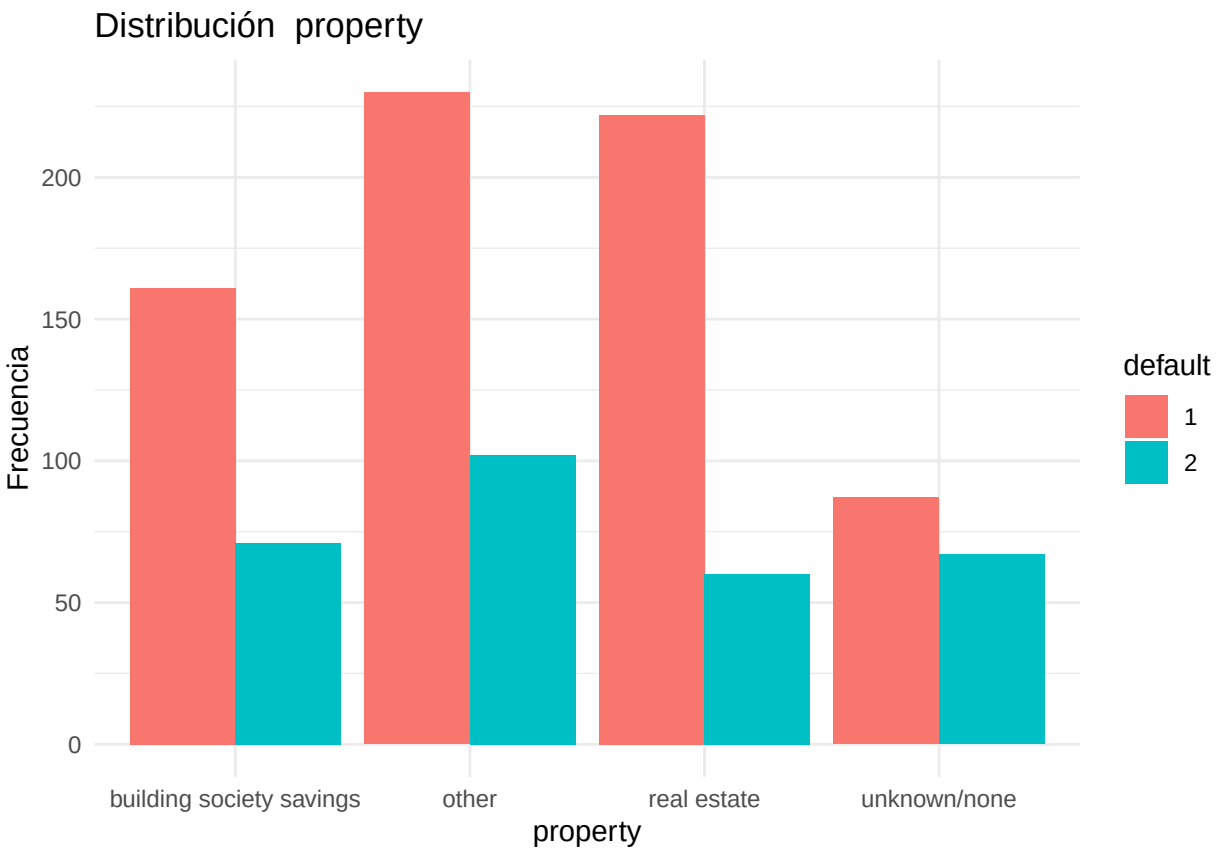

Distribución employment_length
Frecuencia
200
150
100
50
0
> 7 yrs
0 − 1 yrs
1 − 4 yrs
4 − 7 yrs
unemployed
employment_length
default
1
2

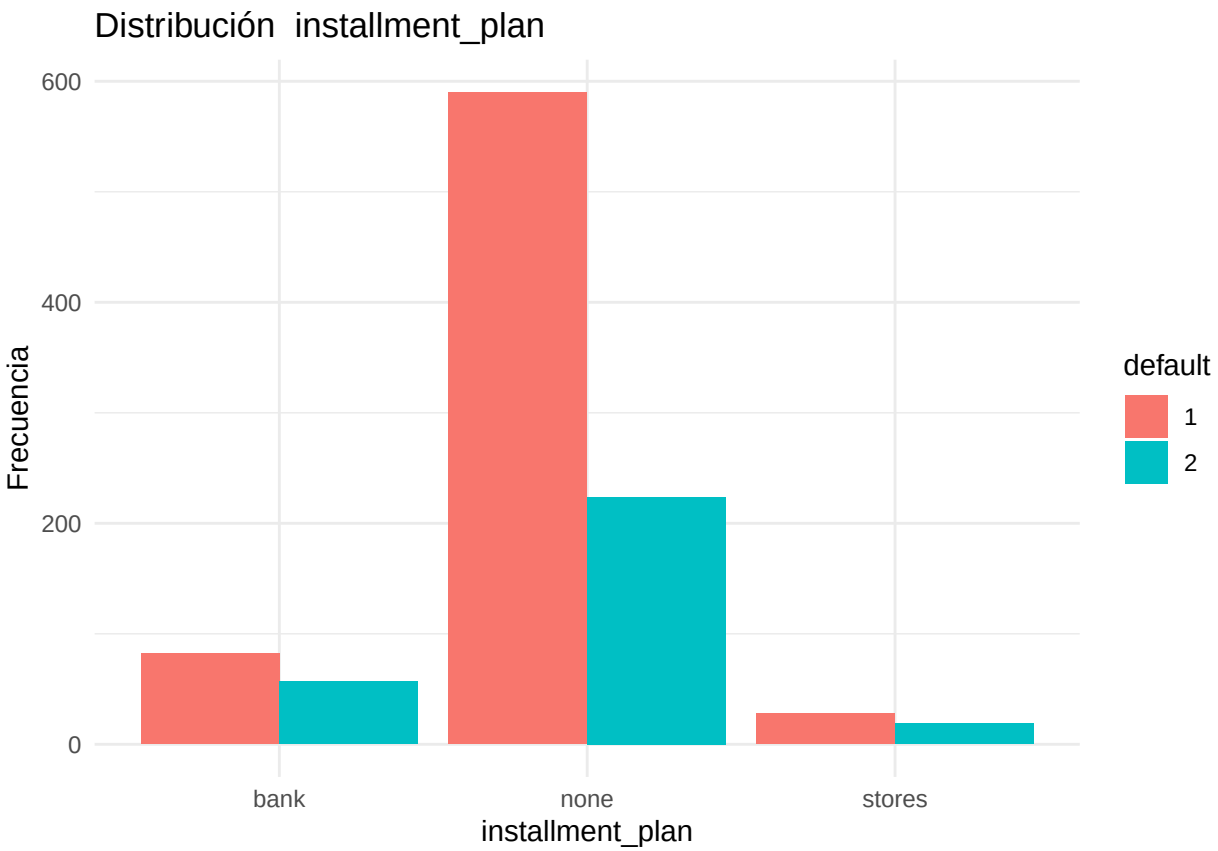

Distribución installment_rate
300
200
100
0
Frecuencia
1
2
3
4
installment_rate
default
1
2

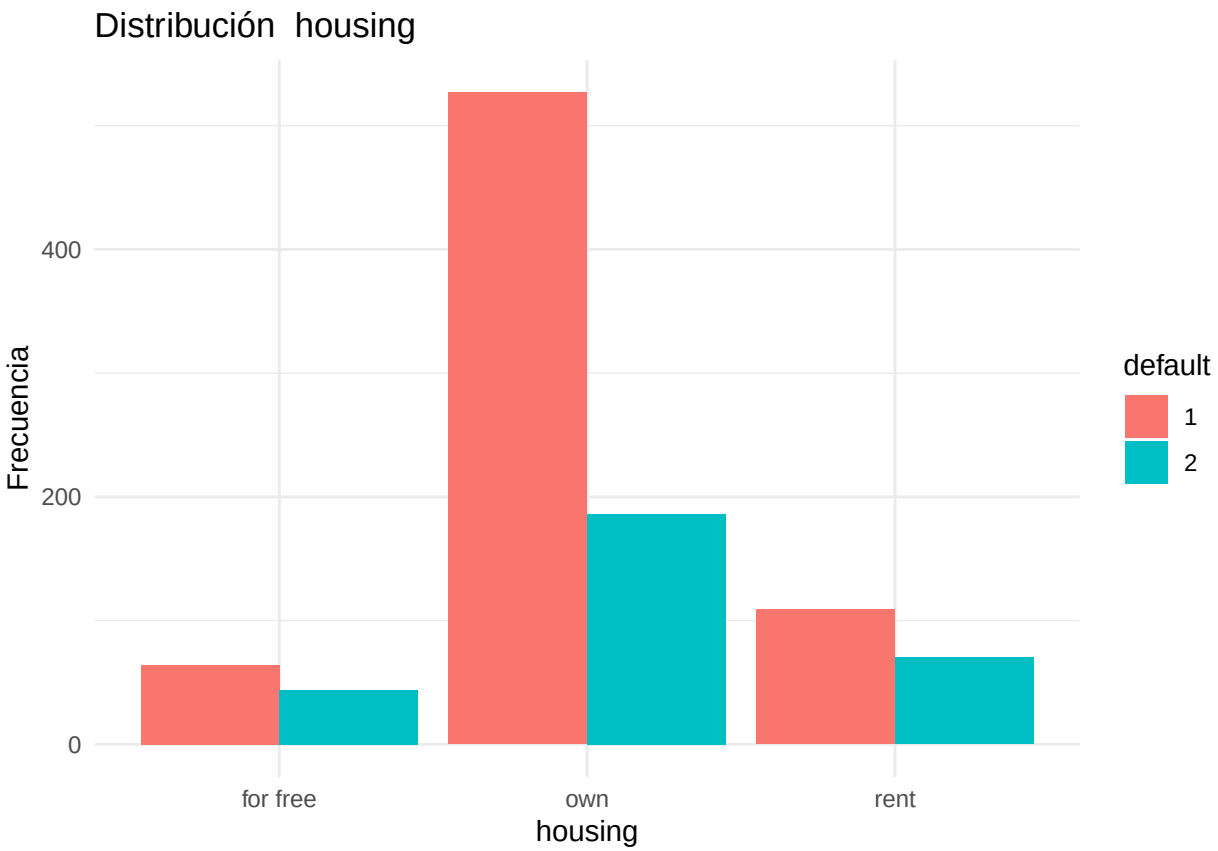

Distribución personal_status
Frecuencia
400
300
200
100
0
divorced male
female
married male
single male
personal_status
default
1
2

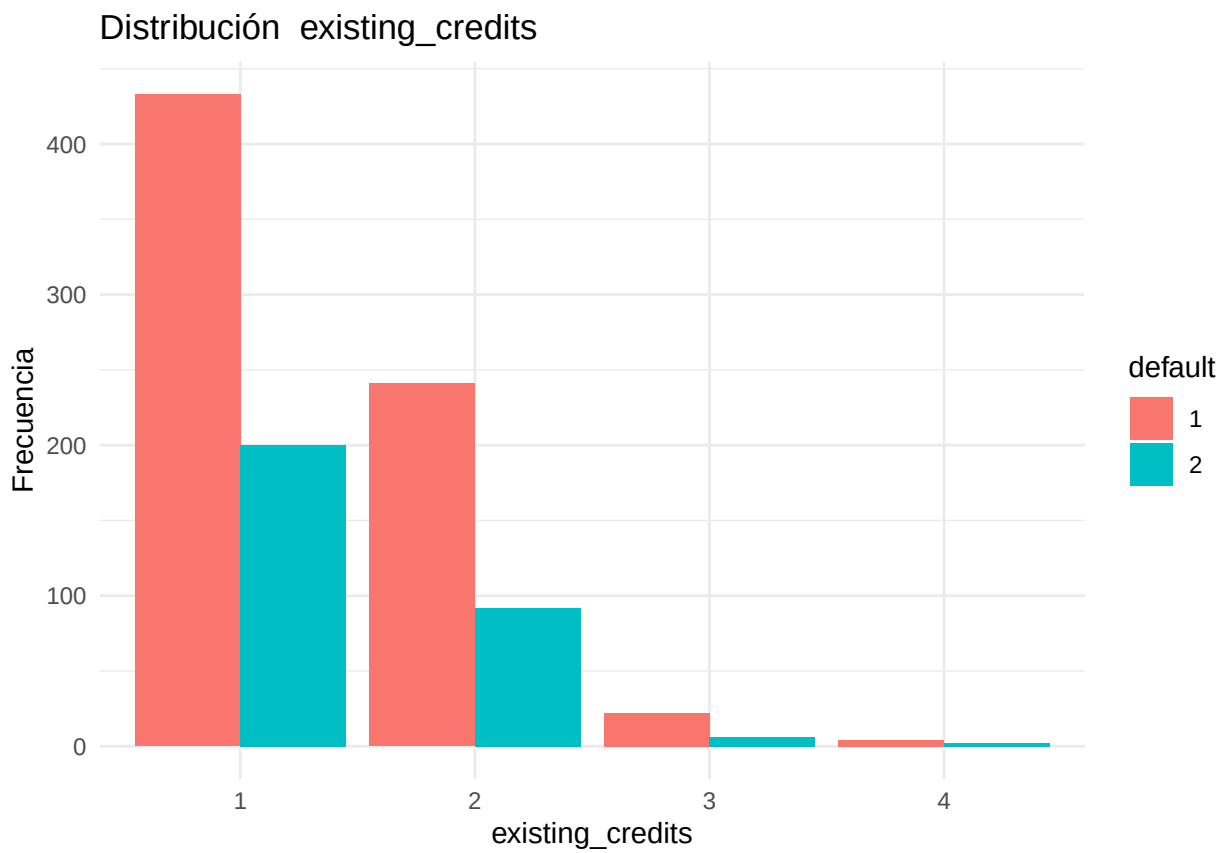

Distribución other_debtors
600
400
200
0
Frecuencia
co−applicant
guarantor
none
other_debtors
default
1
2

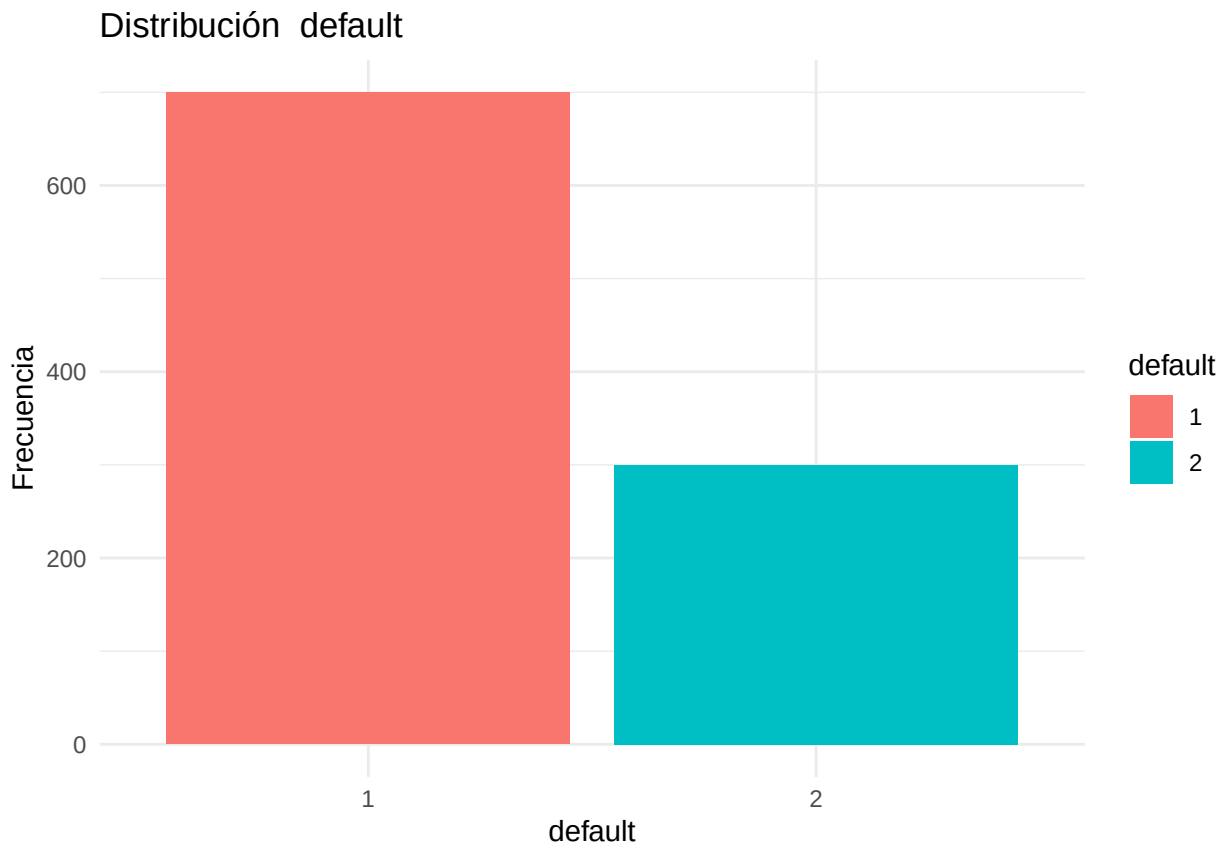

Distribución residence_history
300
200
100
0
Frecuencia
1
2
3
4
residence_history
default
1
2

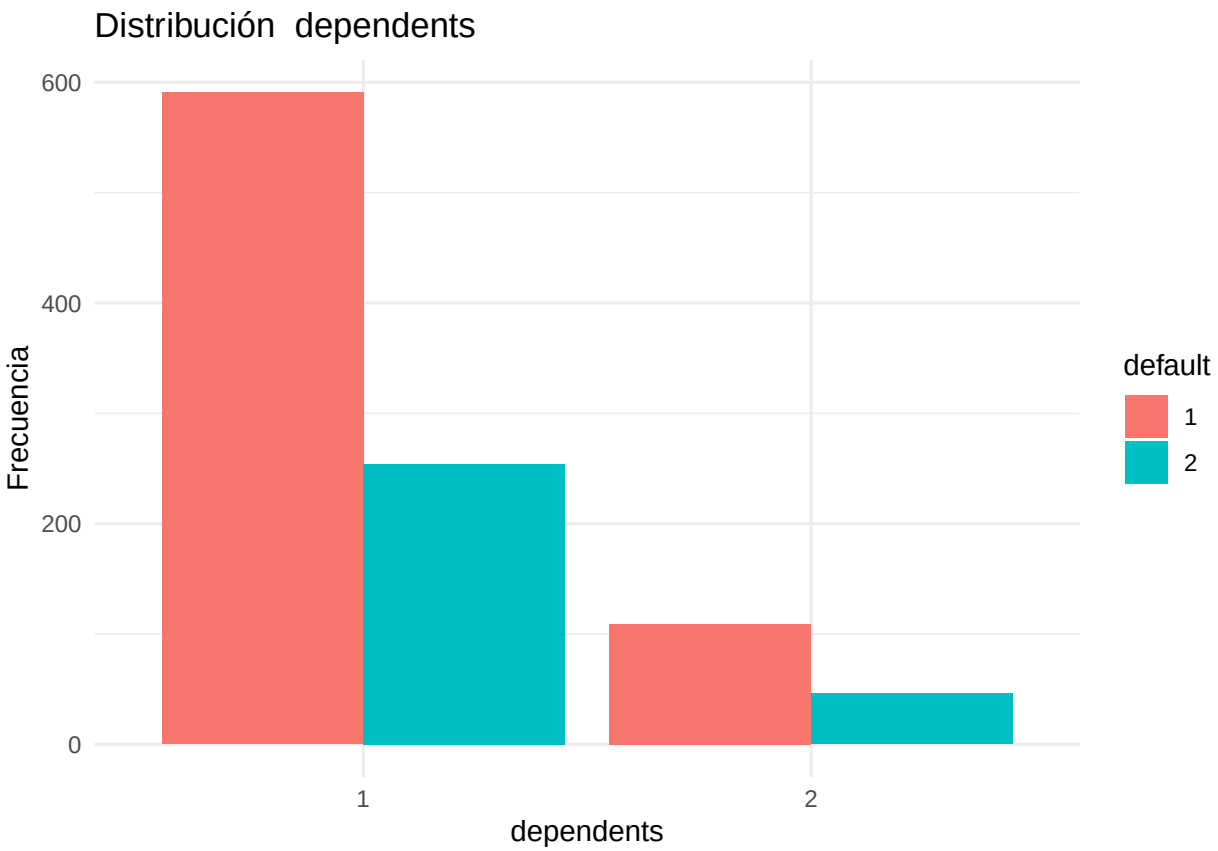

Distribución property
Frecuencia
200
150
100
50
0
building society savings
other
real estate
unknown/none
property
default
1
2

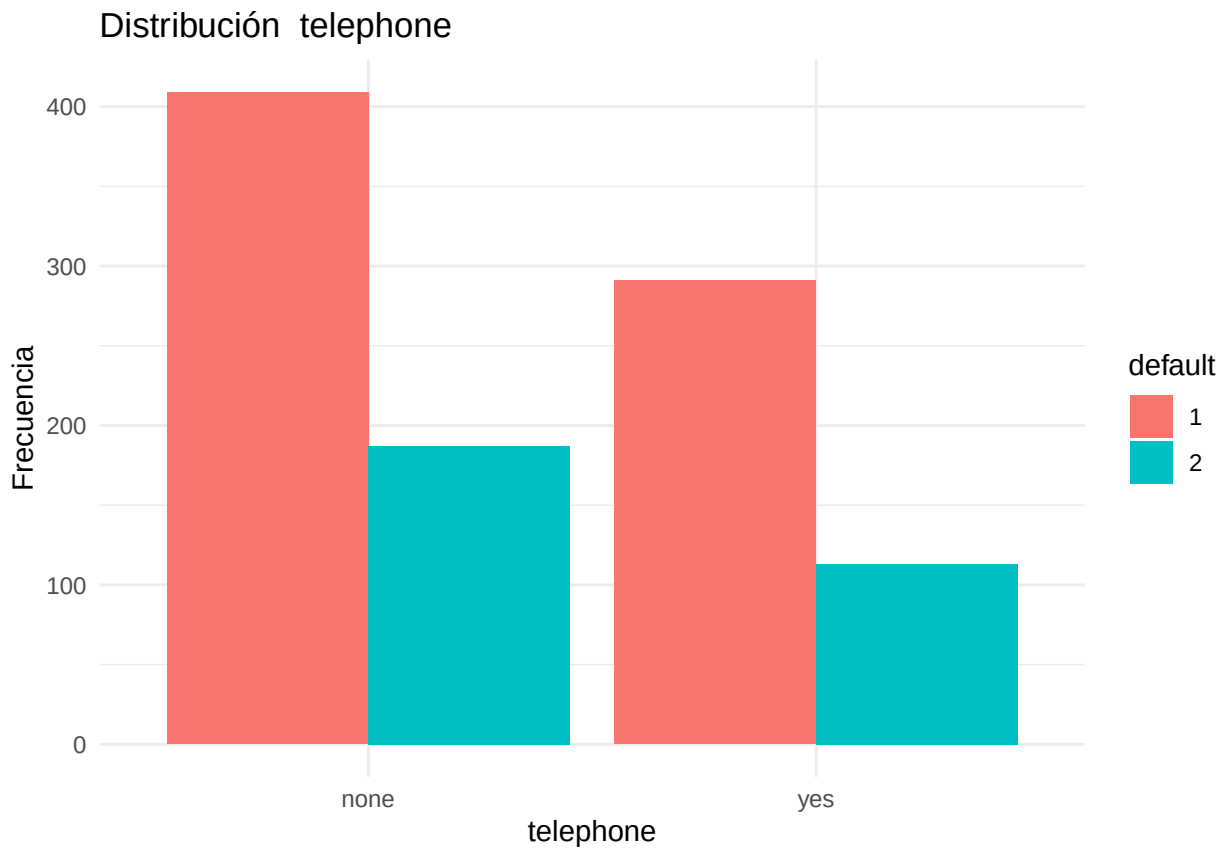

Distribución installment_plan
600
400
200
0
Frecuencia
bank
none
stores
installment_plan
default
1
2

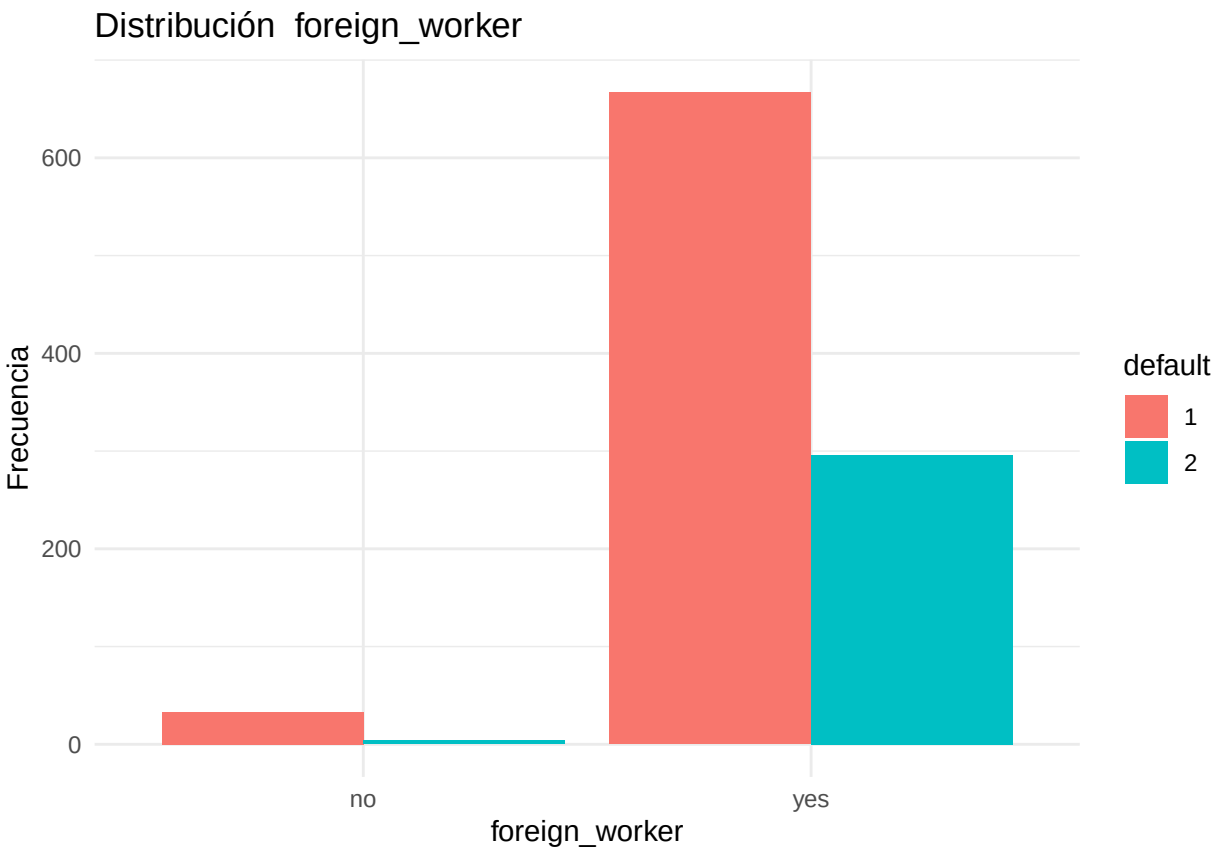

Distribución housing
Frecuencia
400
200
0
for free
own
rent
housing
default
1
2

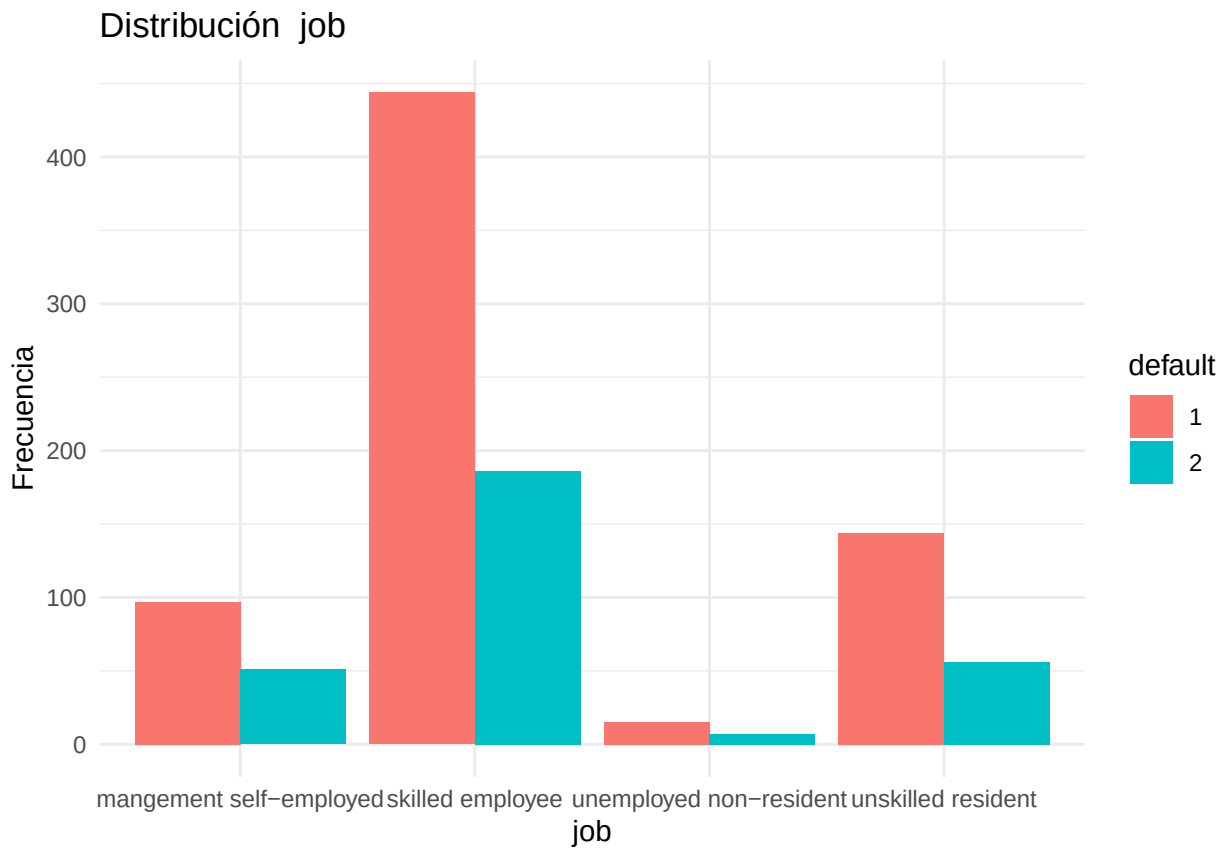

Distribución existing_credits
Frecuencia
400
300
200
100
0
1
2
3
4
existing_credits
default
1
2


Distribución default
600
400
200
0
Frecuencia
1
2
default
default
1
2

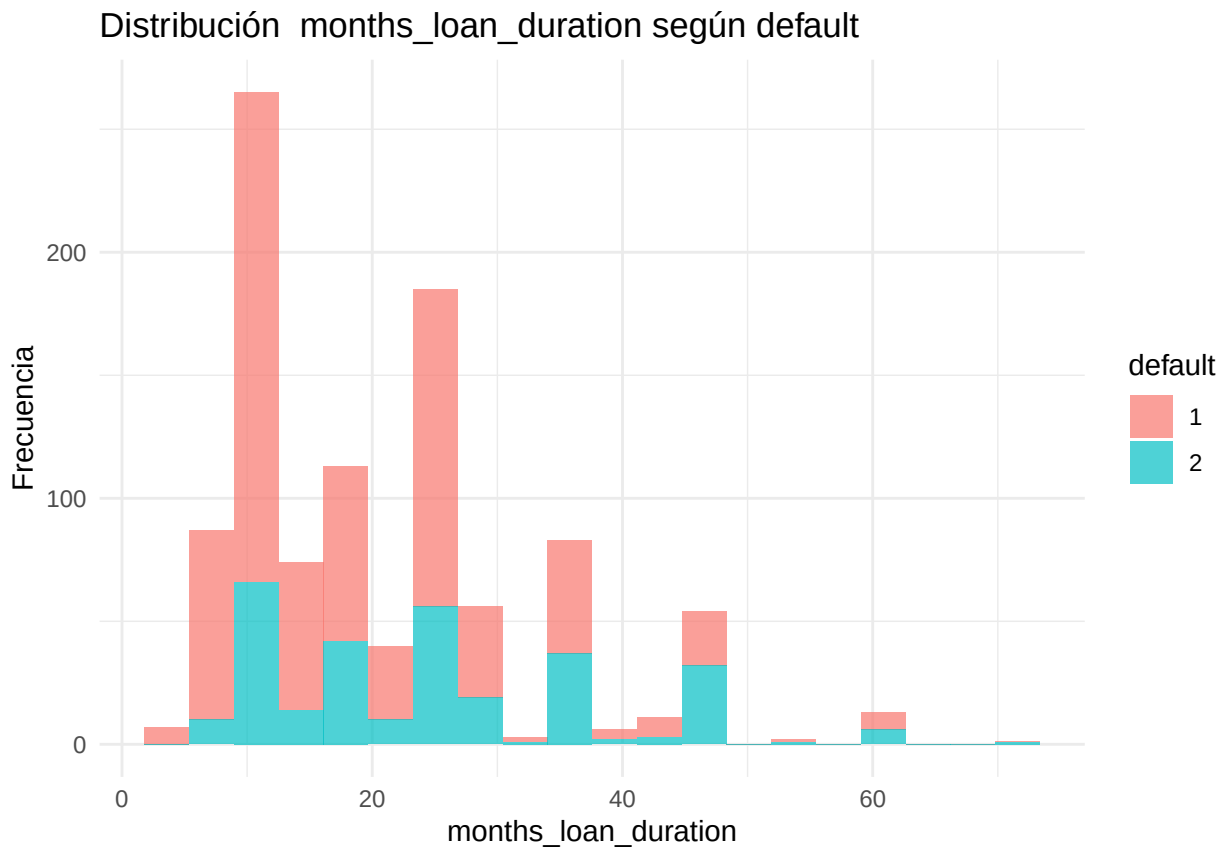

Distribución dependents
600
400
200
0
Frecuencia
1
2
dependents
default
1
2

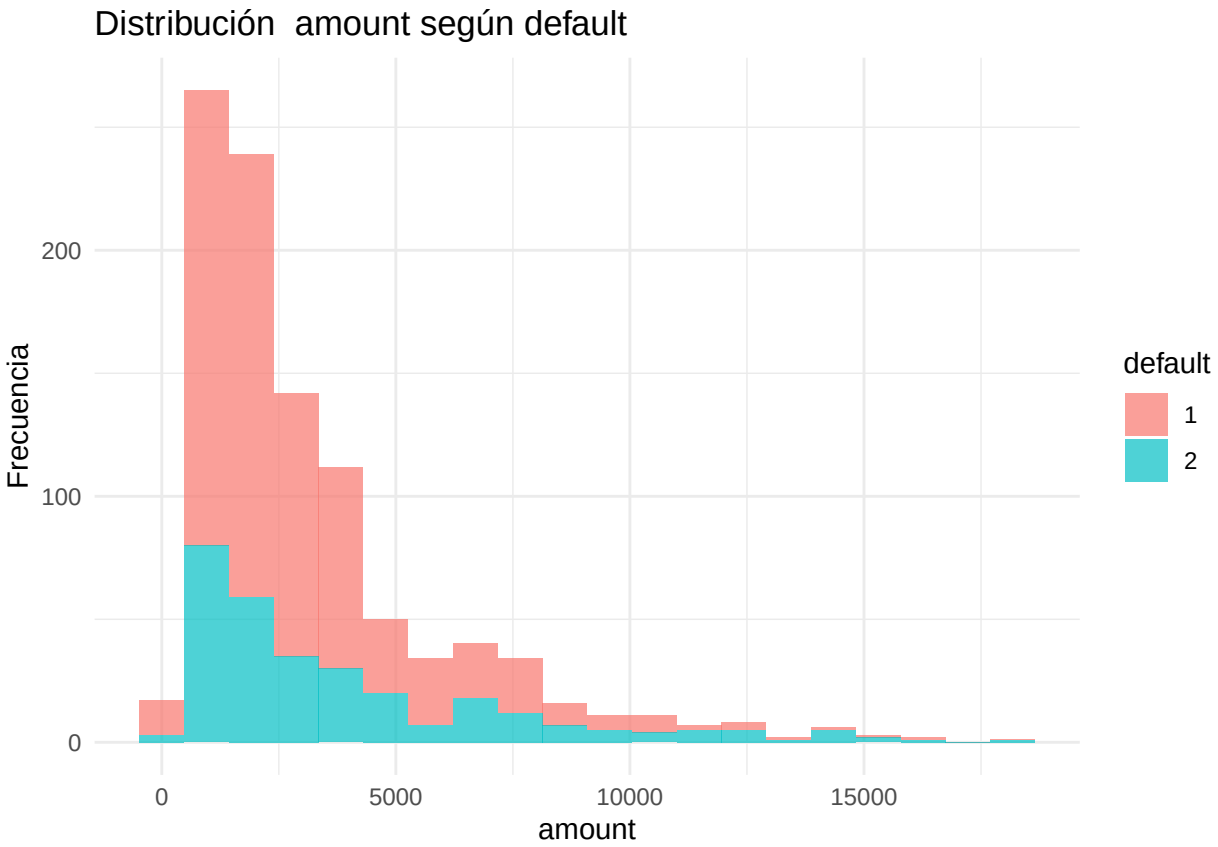

Distribución telephone
400
300
200
100
0
Frecuencia
none
yes
telephone
default
1
2

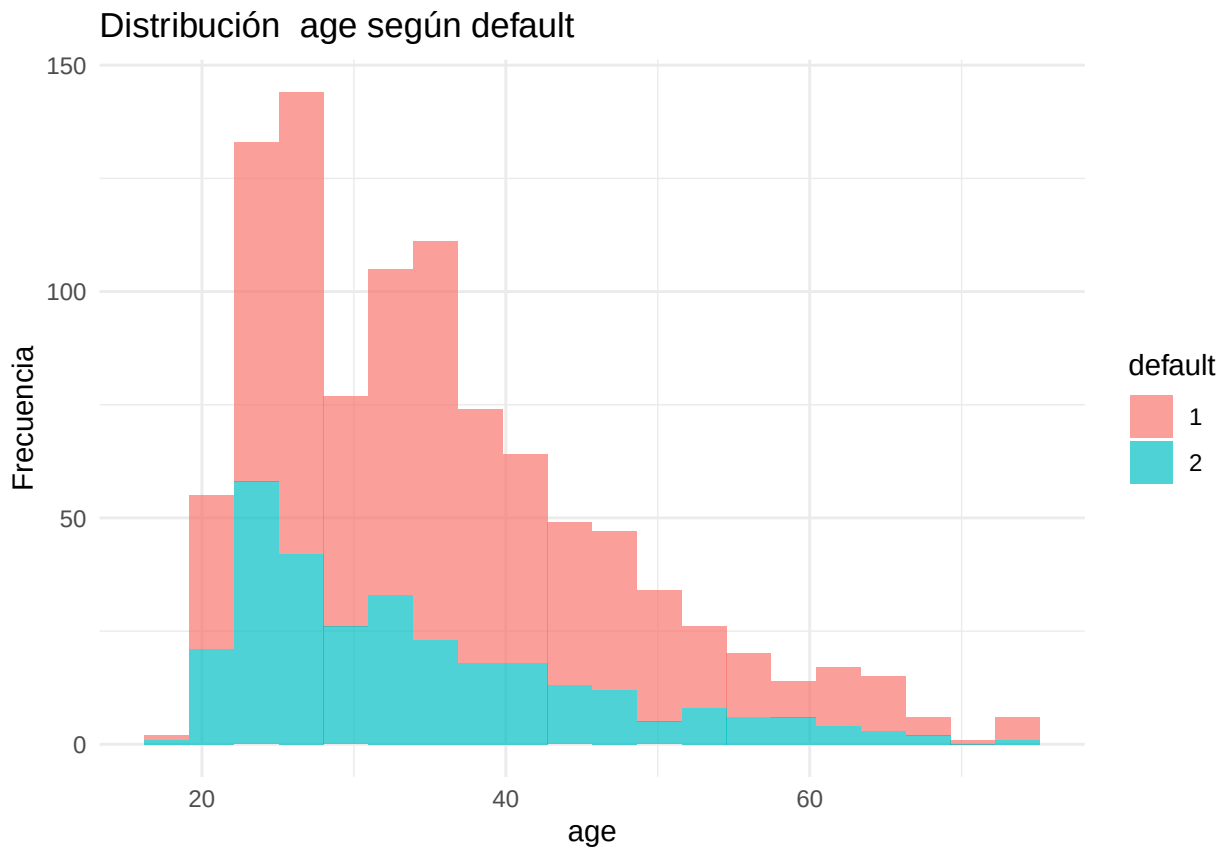

Distribución foreign_worker
600
400
200
0
Frecuencia
no
yes
foreign_worker
default
1
2



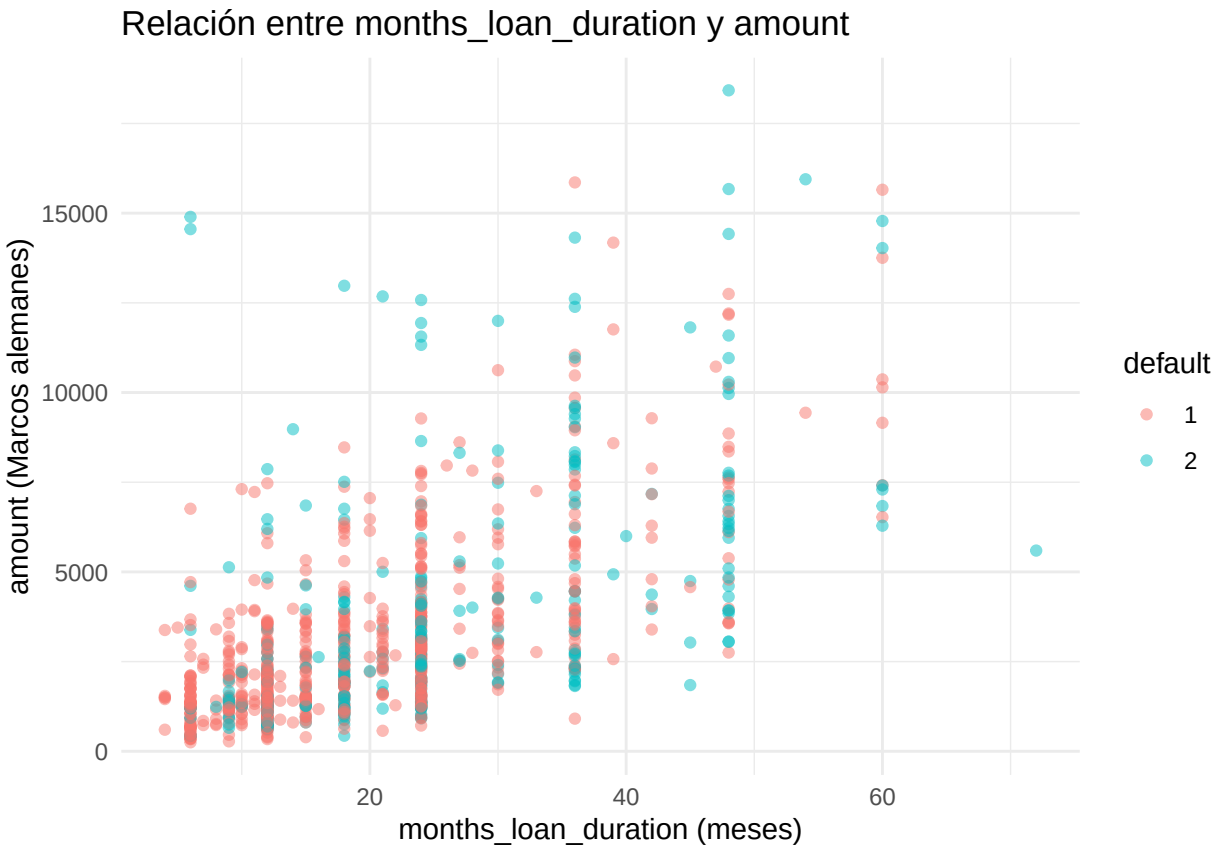

```
for (var_num in numericas) {
  print(
    ggplot(data, aes_string(x = var_num, fill = columna_objetivo)) +
      geom_histogram(bins = 20, alpha = 0.7) +
      labs(title = paste("Distribución ", var_num, "según", columna_objetivo),
           x = var_num, y = "Frecuencia", fill = columna_objetivo) +
      theme_minimal()
  )
}
```


Distribución months_loan_duration según default
Frecuencia
200
100
0
0
20
40
60
months_loan_duration
default
1
2


Distribución amount según default
Frecuencia
200
100
0
0
5000
10000
15000
amount
default
1
2




```
ggplot(data, aes(x = months_loan_duration, y = amount, color = default)) +
  geom_point(alpha = 0.5) +
  labs(title = "Relación entre months_loan_duration y amount",
       x = "months_loan_duration (meses)",
       y = "amount (Marcos alemanes)") +
  theme_minimal()
```




## ¿Y las variables categóricas?

```
variables_tipo_factor <- data[, sapply(data, is.factor)]
nombres_vars <- names(variables_tipo_factor)
combinaciones <- combn(nombres_vars, 2, simplify = FALSE)
resultados <- data.frame(
  Variable1 = character(),
  Variable2 = character(),
  CramersV = numeric(),
  Asociacion = character(),
  p_value = numeric(),
  stringsAsFactors = FALSE
)

for(pair in combinaciones) {
  var1 <- variables_tipo_factor[[pair[1]]]
  var2 <- variables_tipo_factor[[pair[2]]]

  tabla <- table(var1, var2)
  V <- CramerV(tabla, unbiased = TRUE)

  test <- suppressWarnings(chisq.test(tabla, correct = FALSE))
  p_val <- test$p.value
```

```
  asociacion <- ifelse(V > 0.5, "Alta",
                       ifelse(V > 0.3, "Media",
                              ifelse(V > 0.1, "Baja", "Muy Baja")))

  if(p_val < 0.05) {
    resultados <- rbind(resultados, data.frame(
      Variable1 = pair[1],
      Variable2 = pair[2],
      CramersV = round(V, 3),
      Asociacion = asociacion,
      p_value = round(p_val, 5),
      stringsAsFactors = FALSE
    ))
  }
}


print(resultados[order(-resultados$CramersV), ])
```

```
##            Variable1         Variable2 CramersV Asociacion p_value
## 66          property           housing    0.553       Alta 0.00000
## 85         telephone               job    0.426      Media 0.00000
## 17    credit_history  existing_credits    0.378      Media 0.00000
## 10  checking_balance           default    0.352      Media 0.00000
## 46 employment_length               job    0.311      Media 0.00000
## 54   personal_status        dependents    0.284       Baja 0.00000
## 39 employment_length residence_history    0.261       Baja 0.00000
## 18    credit_history           default    0.248       Baja 0.00000
## 62 residence_history           housing    0.237       Baja 0.00000
## 31           purpose         telephone    0.221       Baja 0.00000
## 15    credit_history  installment_plan    0.215       Baja 0.00000
## 28           purpose           housing    0.210       Baja 0.00000
## 26           purpose          property    0.206       Baja 0.00000
## 33           purpose               job    0.203       Baja 0.00000
## 51   personal_status           housing    0.202       Baja 0.00000
## 69          property         telephone    0.198       Baja 0.00000
## 71          property               job    0.194       Baja 0.00000
## 36   savings_balance           default    0.190       Baja 0.00000
## 29           purpose           default    0.183       Baja 0.00012
## 3   checking_balance   savings_balance    0.176       Baja 0.00000
## 41 employment_length           housing    0.173       Baja 0.00000
## 12    credit_history           purpose    0.167       Baja 0.00000
## 32           purpose    foreign_worker    0.166       Baja 0.00113
## 38 employment_length   personal_status    0.166       Baja 0.00000
## 24           purpose     other_debtors    0.165       Baja 0.00001
## 30           purpose        dependents    0.164       Baja 0.00150
## 67          property           default    0.154       Baja 0.00003
## 23           purpose   personal_status    0.151       Baja 0.00002
## 45 employment_length         telephone    0.151       Baja 0.00015
## 2   checking_balance           purpose    0.149       Baja 0.00003
## 40 employment_length          property    0.147       Baja 0.00000
## 83        dependents               job    0.146       Baja 0.00009
## 27           purpose  installment_plan    0.143       Baja 0.00157
## 1   checking_balance    credit_history    0.142       Baja 0.00000
```

```
## 57     other_debtors          property    0.142       Baja 0.00000
## 70          property    foreign_worker    0.142       Baja 0.00015
## 22           purpose  installment_rate    0.140       Baja 0.00042
## 43 employment_length           default    0.136       Baja 0.00105
## 61 residence_history          property    0.136       Baja 0.00000
## 75           housing           default    0.135       Baja 0.00011
## 25           purpose residence_history    0.130       Baja 0.00384
## 48  installment_rate    foreign_worker    0.127       Baja 0.00111
## 78           housing               job    0.127       Baja 0.00001
## 76           housing        dependents    0.126       Baja 0.00035
## 21           purpose employment_length    0.122       Baja 0.00859
## 79  existing_credits        dependents    0.121       Baja 0.00223
## 50   personal_status          property    0.120       Baja 0.00000
## 59     other_debtors    foreign_worker    0.120       Baja 0.00070
## 77           housing         telephone    0.118       Baja 0.00092
## 42 employment_length  existing_credits    0.116       Baja 0.00007
## 20           purpose   savings_balance    0.114       Baja 0.04260
## 73  installment_plan           default    0.113       Baja 0.00163
## 80  existing_credits               job    0.112       Baja 0.00002
## 5   checking_balance     other_debtors    0.109       Baja 0.00063
## 37 employment_length  installment_rate    0.108       Baja 0.00043
## 84         telephone    foreign_worker    0.107       Baja 0.00068
## 49   personal_status residence_history    0.106       Baja 0.00011
## 86    foreign_worker               job    0.105       Baja 0.01159
## 47  installment_rate   personal_status    0.103       Baja 0.00024
## 13    credit_history employment_length    0.101       Baja 0.00064
## 74           housing  existing_credits    0.100   Muy Baja 0.00295
## 8   checking_balance           housing    0.099   Muy Baja 0.00344
## 6   checking_balance residence_history    0.098   Muy Baja 0.00068
## 19    credit_history        dependents    0.098   Muy Baja 0.04888
## 44 employment_length        dependents    0.098   Muy Baja 0.04769
## 53   personal_status           default    0.098   Muy Baja 0.02224
## 64 residence_history         telephone    0.098   Muy Baja 0.02213
## 16    credit_history           housing    0.097   Muy Baja 0.01528
## 35   savings_balance     other_debtors    0.097   Muy Baja 0.01698
## 4   checking_balance employment_length    0.095   Muy Baja 0.00710
## 68          property        dependents    0.095   Muy Baja 0.02954
## 14    credit_history   personal_status    0.092   Muy Baja 0.01213
## 72  installment_plan           housing    0.090   Muy Baja 0.00292
## 55   personal_status         telephone    0.089   Muy Baja 0.04708
## 63 residence_history  existing_credits    0.086   Muy Baja 0.00765
## 34   savings_balance employment_length    0.085   Muy Baja 0.02631
## 58     other_debtors           default    0.082   Muy Baja 0.03606
## 65          property  installment_plan    0.082   Muy Baja 0.03778
## 81           default    foreign_worker    0.082   Muy Baja 0.00944
## 60     other_debtors               job    0.080   Muy Baja 0.04394
## 9   checking_balance  existing_credits    0.077   Muy Baja 0.03804
## 56   personal_status               job    0.077   Muy Baja 0.03695
## 82        dependents    foreign_worker    0.077   Muy Baja 0.01480
## 7   checking_balance          property    0.076   Muy Baja 0.04538
## 52   personal_status  existing_credits    0.076   Muy Baja 0.04293
## 11  checking_balance               job    0.075   Muy Baja 0.04940
```

Aquí nos pasa igual, tenemos una correlación significativa, pero muy lejos de 1, entre housing y property, con

un valor de 0.553118.

```
eta <- function(num_var, cat_var) {
  anova_result <- summary(aov(num_var ~ cat_var))[[1]]
  ss_between <- anova_result["cat_var", "Sum Sq"]
  ss_total <- sum(anova_result[, "Sum Sq"])
  return(sqrt(ss_between / ss_total))
}

categorical_vars <- data[, sapply(data, is.factor)]
numerical_vars <- data[, sapply(data, is.numeric)]

eta_matrix <- matrix(NA, nrow = ncol(categorical_vars), ncol = ncol(numerical_vars),
                     dimnames = list(colnames(categorical_vars), colnames(numerical_vars)))

for (cat in colnames(categorical_vars)) {
  for (num in colnames(numerical_vars)) {
    eta_matrix[cat, num] <- eta(numerical_vars[[num]], categorical_vars[[cat]])
  }
}

print("Matriz de Coeficiente Eta:")
```

```
## [1] "Matriz de Coeficiente Eta:"
```

```
print(eta_matrix)
```

```
##                    months_loan_duration     amount         age
## checking_balance            0.11885510 0.14555608 0.090730499
## credit_history              0.19465363 0.19328338 0.176836040
## purpose                     0.27369172 0.37095411 0.171764736
## savings_balance             0.10558638 0.12950708 0.112602911
## employment_length           0.09399589 0.11190536 0.409607063
## installment_rate            0.11345797 0.27776257 0.059112056
## personal_status             0.13341922 0.18701358 0.245808733
## other_debtors               0.04838687 0.10016410 0.030888080
## residence_history           0.08196595 0.08142659 0.274095854
## property                    0.30427442 0.31833910 0.224743306
## installment_plan            0.07790207 0.04833623 0.047068880
## housing                     0.19217435 0.20181165 0.307001631
## existing_credits            0.07842615 0.03019835 0.172233703
## default                     0.21492667 0.15473864 0.091127409
## dependents                  0.02383448 0.01714215 0.118200833
## telephone                   0.16471821 0.27699511 0.145258701
## foreign_worker              0.13819629 0.05005001 0.006151396
## job                         0.21868814 0.33460674 0.164476470
```

```
top_corr <- eta_matrix
top_corr[abs(top_corr) <= 0.2] <- NA
print("Correlaciones top:")
```

```
## [1] "Correlaciones top:"
```

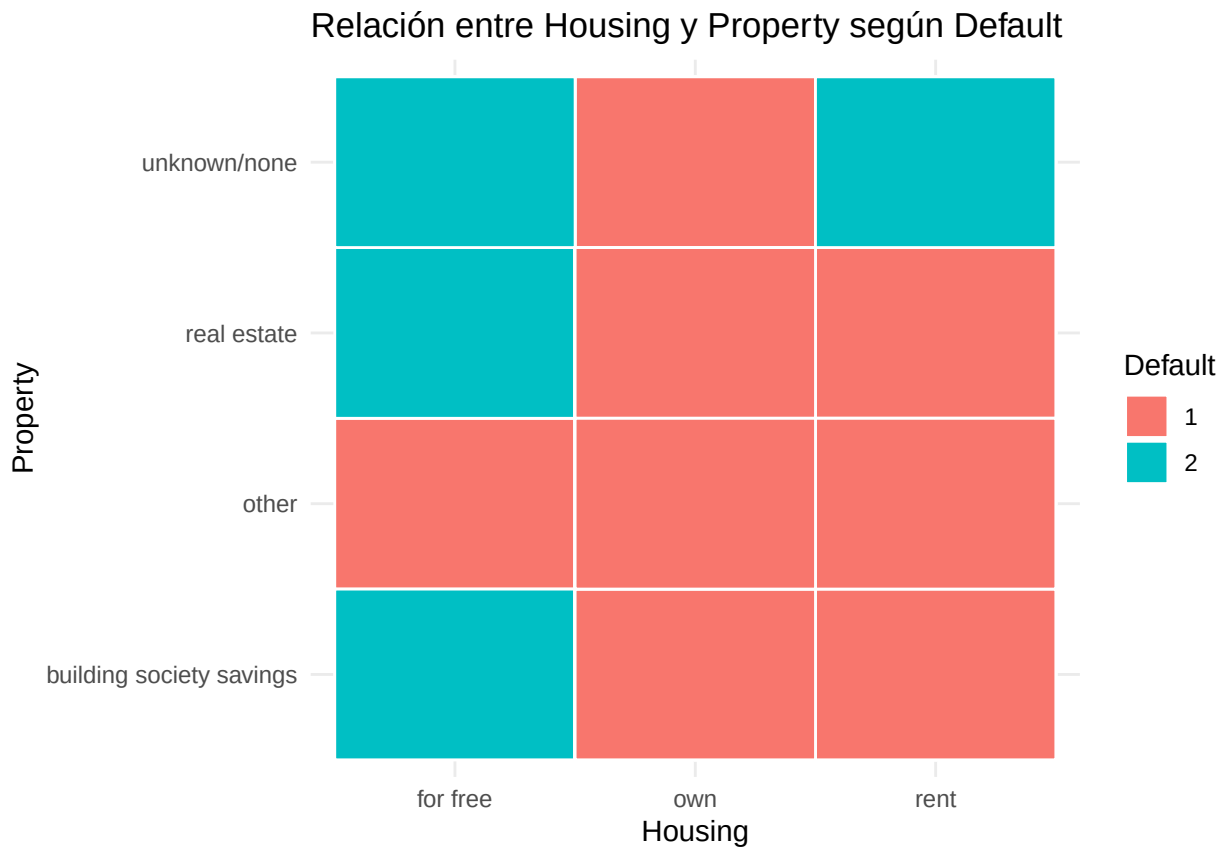
```
print(top_corr)
```

```
##                    months_loan_duration    amount       age
## checking_balance                     NA        NA        NA
```

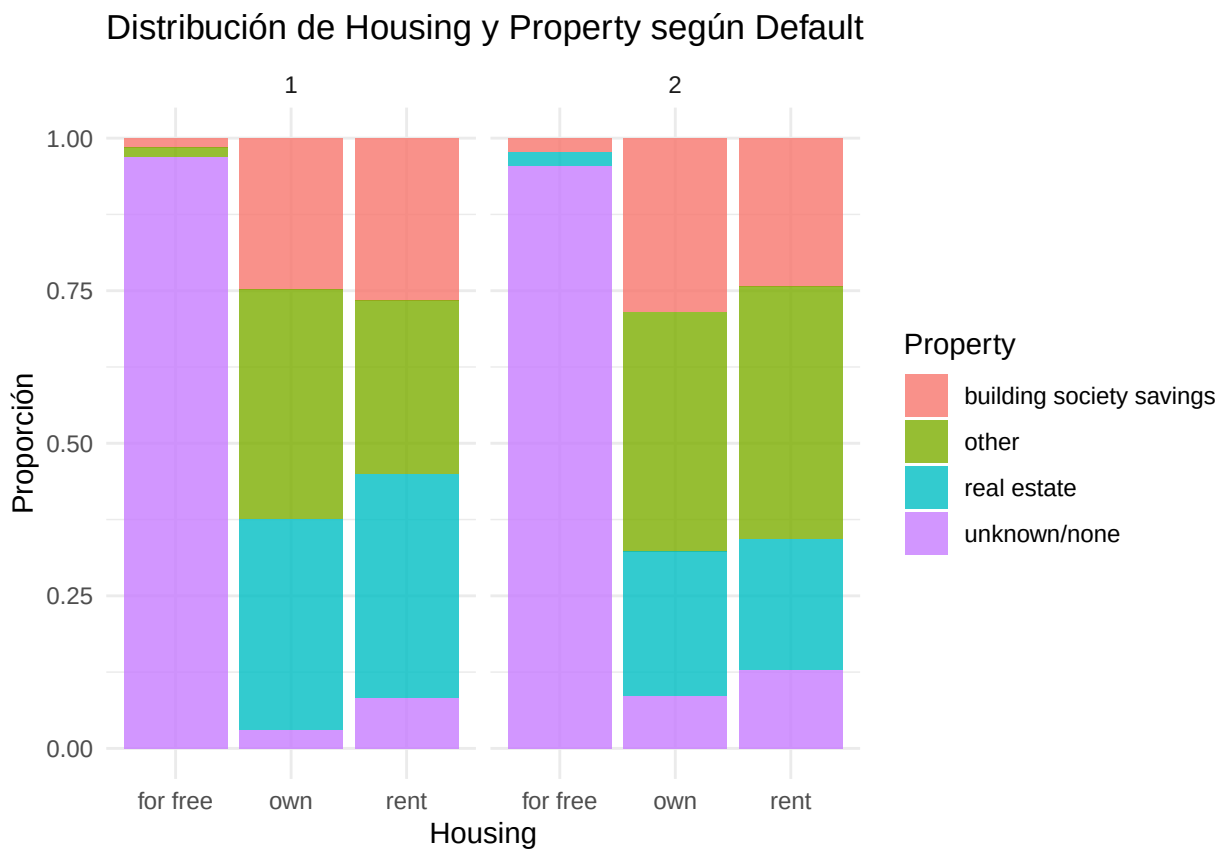
```
## credit_history                 NA        NA        NA
## purpose                 0.2736917 0.3709541        NA
## savings_balance                NA        NA        NA
## employment_length              NA        NA 0.4096071
## installment_rate               NA 0.2777626        NA
## personal_status                NA        NA 0.2458087
## other_debtors                  NA        NA        NA
## residence_history              NA        NA 0.2740959
## property                0.3042744 0.3183391 0.2247433
## installment_plan               NA        NA        NA
## housing                        NA 0.2018116 0.3070016
## existing_credits               NA        NA        NA
## default                 0.2149267        NA        NA
## dependents                     NA        NA        NA
## telephone                      NA 0.2769951        NA
## foreign_worker                 NA        NA        NA
## job                     0.2186881 0.3346067        NA
```

No hay nada significativo aqui, a excepcion de existing_credits con credit_history con un 0.595094 de correlación.

```
ggplot(data, aes(x = housing, y = property, fill = default)) +
  geom_tile(color = "white", lwd = 0.5, linetype = 1) +
  labs(title = "Relación entre Housing y Property según Default",
       x = "Housing",
       y = "Property",
       fill = "Default") +
  theme_minimal()
```




```
ggplot(data, aes(x = housing, fill = property)) +
  geom_bar(position = "fill", alpha = 0.8) +
  facet_wrap(~default) +
  labs(title = "Distribución de Housing y Property según Default",
       x = "Housing",
       y = "Proporción",
       fill = "Property") +
  theme_minimal()
```




Si hago un grafico estilo mosaico, se puede ver como es mas importante en las variables housing y property la variable default. Se puede ver que las distintas propiedades, en el grafico de barras apilado, tienen una distribucion realmente no tan diferente aunque de insights diría que, si es real estate, tiene menos probabilidad de default que si es el resto de tipos de propiedades, pero realmente no tan significativo.

De momento, dejo las variables como están, teniendo en cuenta, que para el modelo de árbol de decisión, no es necesario inicialmente eliminar variables correlacionadas, y en el caso que quisiera hacer diferentes pruebas después, siempre está bien conocer de antemano las correlaciones entre las variables.

### Realizar un primer árbol de decisión. Puedes decidir utilizar todas las variables o, de forma justificada, quitar alguna para el ajuste del modelo

```
data[[columna_objetivo]] <- as.factor(data[[columna_objetivo]])
set.seed(123)
indices <- sample(1:nrow(data), size = 0.7 * nrow(data))
train <- data[indices, ]
test <- data[-indices, ]
model <- C50::C5.0(train[, colnames(train) != columna_objetivo], train[[columna_objetivo]])
summary(model)
```

```
##
## Call:
## C5.0.default(x = train[, colnames(train) != columna_objetivo], y
##  = train[[columna_objetivo]])
##
##
## C5.0 [Release 2.07 GPL Edition]      Sun Dec 29 18:25:17 2024
```

```
## ------------------------------
##
## Class specified by attribute `outcome'
##
## Read 700 cases (21 attributes) from undefined.data
##
## Decision tree:
##
## checking_balance = unknown: 1 (272/31)
## checking_balance in {< 0 DM,> 200 DM,1 - 200 DM}:
## :...months_loan_duration > 42:
##     :...savings_balance in {< 100 DM,> 1000 DM,101 - 500 DM,
##     :   :                   501 - 1000 DM}: 2 (32/4)
##     :   savings_balance = unknown: 1 (5)
##     months_loan_duration <= 42:
##     :...credit_history in {fully repaid,fully repaid this bank}:
##         :...other_debtors in {co-applicant,guarantor}: 1 (4/1)
##         :   other_debtors = none: 2 (42/13)
##         credit_history in {critical,delayed,repaid}:
##         :...other_debtors = co-applicant:
##             :...employment_length in {> 7 yrs,0 - 1 yrs,4 - 7 yrs,
##             :   :                      unemployed}: 1 (7/1)
##             :   employment_length = 1 - 4 yrs: 2 (7/1)
##             other_debtors = guarantor:
##             :...credit_history in {delayed,repaid}: 1 (20)
##             :   credit_history = critical:
##             :   :...purpose = car (new): 2 (2)
##             :       purpose in {business,car (used),domestic appliances,
##             :                   education,furniture,others,radio/tv,repairs,
##             :                   retraining}: 1 (4/1)
##             other_debtors = none:
##             :...months_loan_duration <= 11:
##                 :...amount <= 10722: 1 (68/10)
##                 :   amount > 10722: 2 (2)
##                 months_loan_duration > 11:
##                 :...purpose in {domestic appliances,repairs}: 2 (8/2)
##                     purpose in {others,retraining}: 1 (3/1)
##                     purpose = business:
##                     :...personal_status = divorced male: 2 (2)
##                     :   personal_status in {female,married male,
##                     :                       single male}: 1 (20/1)
##                     purpose = car (used):
##                     :...amount <= 8648: 1 (20/2)
##                     :   amount > 8648: 2 (5/1)
##                     purpose = car (new):
##                     :...savings_balance in {> 1000 DM,
##                     :   :                   501 - 1000 DM}: 1 (4)
##                     :   savings_balance = unknown: 2 (8/2)
##                     :   savings_balance = 101 - 500 DM:
##                     :   :...personal_status = female: 2 (2)
##                     :   :   personal_status in {divorced male,married male,
##                     :   :                       single male}: 1 (6)
##                     :   savings_balance = < 100 DM:
##                     :   :...installment_rate = 2: 1 (6/1)
```

```
##                     :       installment_rate in {3,4}: 2 (25/6)
##                     :       installment_rate = 1:
##                     :       :...amount <= 1778: 2 (2)
##                     :           amount > 1778: 1 (2)
##                     purpose = education:
##                     :...dependents = 2: 2 (2)
##                     :   dependents = 1:
##                     :   :...housing in {for free,rent}: 1 (5)
##                     :       housing = own:
##                     :       :...checking_balance in {< 0 DM,> 200 DM}: 2 (3)
##                     :           checking_balance = 1 - 200 DM: 1 (1)
##                     purpose = furniture:
##                     :...residence_history = 1:
##                     :   :...job in {mangement self-employed,
##                     :   :   :       skilled employee}: 1 (7)
##                     :   :   job in {unemployed non-resident,
##                     :   :           unskilled resident}: 2 (2)
##                     :   residence_history = 2:
##                     :   :...employment_length in {> 7 yrs,4 - 7 yrs,
##                     :   :   :                     unemployed}: 2 (6)
##                     :   :   employment_length in {0 - 1 yrs,
##                     :   :                         1 - 4 yrs}: 1 (7/1)
##                     :   residence_history = 4:
##                     :   :...property in {building society savings,other,
##                     :   :   :            unknown/none}: 1 (21/3)
##                     :   :   property = real estate: 2 (3/1)
##                     :   residence_history = 3:
##                     :   :...installment_rate in {1,2}: 1 (2)
##                     :       installment_rate = 3: 2 (2)
##                     :       installment_rate = 4:
##                     :       :...checking_balance in {< 0 DM,> 200 DM}: 1 (2)
##                     :           checking_balance = 1 - 200 DM: 2 (2)
##                     purpose = radio/tv:
##                     :...dependents = 2: 2 (6/1)
##                         dependents = 1:
##                         :...employment_length = > 7 yrs: 1 (10)
##                             employment_length in {0 - 1 yrs,1 - 4 yrs,
##                             :                     4 - 7 yrs,unemployed}:
##                             :...savings_balance in {> 1000 DM,
##                                 :                   501 - 1000 DM}: 1 (5/1)
##                                 savings_balance = 101 - 500 DM:
##                                 :...credit_history = delayed: 1 (1)
##                                 :   credit_history in {critical,repaid}: 2 (4)
##                                 savings_balance = unknown:
##                                 :...job = mangement self-employed: 2 (2)
##                                 :   job in {skilled employee,
##                                 :           unemployed non-resident,
##                                 :           unskilled resident}: 1 (3)
##                                 savings_balance = < 100 DM:
##                                 :...employment_length = 0 - 1 yrs: 2 (7/1)
##                                     employment_length in {4 - 7 yrs,
##                                     :                     unemployed}: 1 (4)
##                                     employment_length = 1 - 4 yrs:
##                                     :...existing_credits in {1,3,
```

```
##                                         :                      4}: 1 (13/4)
##                                        existing_credits = 2: 2 (2)
##
##
## Evaluation on training data (700 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##      51   90(12.9%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     464    32    (a): class 1
##      58   146    (b): class 2
##
##
##  Attribute usage:
##
##  100.00% checking_balance
##   61.14% months_loan_duration
##   55.86% credit_history
##   55.86% other_debtors
##   34.43% purpose
##   19.00% savings_balance
##   14.14% amount
##   11.14% employment_length
##    9.71% dependents
##    7.71% residence_history
##    6.14% installment_rate
##    4.29% personal_status
##    3.43% property
##    2.14% existing_credits
##    2.00% job
##    1.29% housing
##
##
## Time: 0.0 secs
```

**Con el árbol obtenido, realiza una breve explicación de las reglas obtenidas así como de todos los puntos que te parezcan interesantes. Un elemento a considerar es, por ejemplo, cuantas observaciones caen dentro de cada regla**

## ¿Y la importancia de las variables?

```
C5imp(model)
```

```
##                      Overall
## checking_balance      100.00
## months_loan_duration   61.14
## credit_history         55.86
```

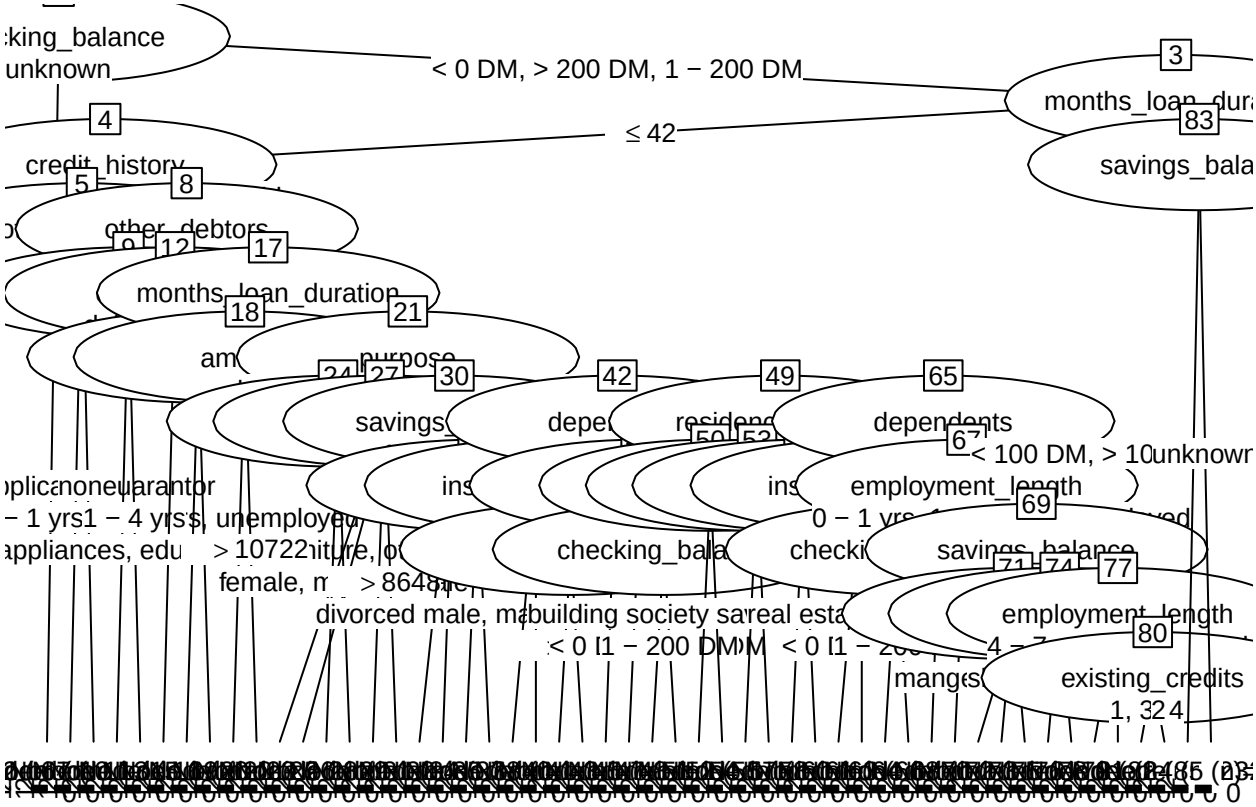
```
## other_debtors          55.86
## purpose                34.43
## savings_balance        19.00
## amount                 14.14
## employment_length      11.14
## dependents              9.71
## residence_history       7.71
## installment_rate        6.14
## personal_status         4.29
## property                3.43
## existing_credits        2.14
## job                     2.00
## housing                 1.29
## age                     0.00
## installment_plan        0.00
## telephone               0.00
## foreign_worker          0.00
```

Vemos que, checking_balance es la variable más importante, seguida de months_loan_duration , credit_history other_debtors, purpose y savings_balance.

```
model <- C50::C5.0(train[, colnames(train) != columna_objetivo], train[[columna_objetivo]],
                   control = C5.0Control(minCases = 2))
```

Si pongo minCases 2 la verdad es que tiene buen resultado, aunque no es totalmente visible en el grafico, pero mas adelante lo veremos bien.

```
plot(model, gp = gpar(fontsize = 9.5))
```



**Una vez tengas un modelo válido, procede a realizar un análisis de la bondad de ajuste sobre el conjunto de test y matriz de confusión. ¿Te parece un modelo suficientemente bueno como para utilizarlo? Justifica tu respuesta considerando todos los posibles tipos de error**

```
predicciones_test <- predict(model, newdata = test)

matriz_confusion <- table(Predicted = predicciones_test, Actual = test$default)

print(matriz_confusion)
```

```
##          Actual
## Predicted   1   2
##         1 172  52
##         2  32  44
```

Es suficientemente bueno pero es mejorable, los falsos positivos son altos y los falsos negativos no tan altos pero tambien

**Con un enfoque parecido a los puntos anteriores y considerando las mismas variables, enriquece el ejercicio mediante el ajuste de modelos de árbol de decisión complementarios. ¿Es el nuevo enfoque mejor que el original? Justifica la respuesta**

```
# cuadricula con todas las combinaciones a probar
param_grid <- expand.grid(
  minCases = seq(0, 200, by = 25), # minimo de casos por rama
  trials = c(1, 5, 10, 15, 20, 25, 30, 35, 40), # intentos
  CF = c(0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45), #ajuste
  winnow = c(TRUE, FALSE), # filtrar variables no importantes
  bands = c(5, 10, 15), # division en variables numericas
  rules = c(TRUE, FALSE), # basado en reglas o arbol tradicional
  noGlobalPruning = c(TRUE, FALSE) # desactiva la poda global
)

# si rules es false, guardamos NA en bands y no lo usamos
param_grid$bands <- ifelse(param_grid$rules == FALSE, NA, param_grid$bands)

# duplicados en rules false y bands NA
param_grid <- param_grid[!duplicated(param_grid), ]

#combinaciones totales que voy a probar, he reducido mucho ya que me tardaba horas sino, ahora es unos
nrow(param_grid)
```

```
## [1] 11664
```

```
resultados <- data.frame(
  minCases = integer(0),
  trials = integer(0),
  CF = numeric(0),
  winnow = logical(0),
  bands = numeric(0),
  noGlobalPruning = logical(0),
  rules = logical(0),
```

```
  Exactitud = numeric(0),
  Sensibilidad = numeric(0),
  Especificidad = numeric(0),
  Precisión = numeric(0),
  F1_Score = numeric(0),
  F2_Score = numeric(0),
  F0.5_Score = numeric(0)
)

set.seed(123)
indices <- sample(1:nrow(data), size = 0.7 * nrow(data))
train <- data[indices, ]
test <- data[-indices, ]
```

```
for (i in 1:nrow(param_grid)) {
  mc <- param_grid$minCases[i]
  trials <- param_grid$trials[i]
  cf <- param_grid$CF[i]
  winnow <- param_grid$winnow[i]
  bands <- param_grid$bands[i]
  rules <- param_grid$rules[i]
  noGlobalPruning <- param_grid$noGlobalPruning[i]
  earlyStopping <- param_grid$earlyStopping[i]

modelo <- C5.0(
  x = train[, -which(names(train) == columna_objetivo)],
  y = train[[columna_objetivo]],
  trials = trials,
  control = if (rules) {
    C5.0Control(
      minCases = mc,
      CF = cf,
      winnow = winnow,
      earlyStopping = TRUE,
      bands = bands,
      noGlobalPruning = noGlobalPruning
    )
  } else {
    C5.0Control(
      minCases = mc,
      CF = cf,
      winnow = winnow,
      earlyStopping = TRUE,
      noGlobalPruning = noGlobalPruning
    )
  },
  rules = rules
)


  predicciones <- predict(modelo, newdata = test)
  matriz_confusion <- table(Predicted = predicciones, Actual = test[[columna_objetivo]])

  clases <- sort(unique(test[[columna_objetivo]]))
```

```
  matriz_confusion <- matriz_confusion[clases, clases]


  TP <- matriz_confusion["2", "2"]  # VP
  TN <- matriz_confusion["1", "1"]  # VN
  FP <- matriz_confusion["2", "1"]  # FP
  FN <- matriz_confusion["1", "2"]  # FN

  exactitud <- (TP + TN) / sum(matriz_confusion)
  sensibilidad <- TP / (TP + FN)
  especificidad <- TN / (TN + FP)
  precision <- TP / (TP + FP)
  f1_score <- 2 * (precision * sensibilidad) / (precision + sensibilidad)
  f2_score <- 5 * (precision * sensibilidad) / (4 * precision + sensibilidad)
  f0_5_score <- 1.25 * (precision * sensibilidad) / (0.25 * precision + sensibilidad)

  resultados <- rbind(resultados, data.frame(
    minCases = mc,
    trials = trials,
    CF = cf,
    winnow = winnow,
    bands = bands,
    noGlobalPruning = noGlobalPruning,
    rules = rules,
    Exactitud = round(exactitud * 100, 2),
    Sensibilidad = round(sensibilidad * 100, 2),
    Especificidad = round(especificidad * 100, 2),
    Precisión = round(precision * 100, 2),
    F1_Score = round(f1_score * 100, 2),
    F2_Score = round(f2_score * 100, 2),
    F0.5_Score = round(f0_5_score * 100, 2)
  ))
}
```

```
print(head(resultados[order(-resultados$F1_Score), ], n=20))
```

```
##       minCases trials   CF winnow bands noGlobalPruning rules Exactitud
## 6756       125     15 0.15  FALSE     5           FALSE  TRUE     77.33
## 8214       125     15 0.15  FALSE    10           FALSE  TRUE     77.33
## 9672       125     15 0.15  FALSE    15           FALSE  TRUE     77.33
## 11246      100     35 0.20  FALSE    NA           FALSE FALSE     77.33
## 11255      100     40 0.20  FALSE    NA           FALSE FALSE     77.33
## 22          75     10 0.05   TRUE     5            TRUE  TRUE     75.33
## 31          75     15 0.05   TRUE     5            TRUE  TRUE     75.33
## 40          75     20 0.05   TRUE     5            TRUE  TRUE     75.33
## 49          75     25 0.05   TRUE     5            TRUE  TRUE     75.33
## 58          75     30 0.05   TRUE     5            TRUE  TRUE     75.33
## 67          75     35 0.05   TRUE     5            TRUE  TRUE     75.33
## 76          75     40 0.05   TRUE     5            TRUE  TRUE     75.33
## 1480        75     10 0.05   TRUE    10            TRUE  TRUE     75.33
## 1489        75     15 0.05   TRUE    10            TRUE  TRUE     75.33
## 1498        75     20 0.05   TRUE    10            TRUE  TRUE     75.33
## 1507        75     25 0.05   TRUE    10            TRUE  TRUE     75.33
## 1516        75     30 0.05   TRUE    10            TRUE  TRUE     75.33
```

```
## 1525        75     35 0.05   TRUE    10            TRUE  TRUE     75.33
## 1534        75     40 0.05   TRUE    10            TRUE  TRUE     75.33
## 2938        75     10 0.05   TRUE    15            TRUE  TRUE     75.33
##       Sensibilidad Especificidad Precisión F1_Score F2_Score F0.5_Score
## 6756         55.21         87.75     67.95    60.92    57.36      64.95
## 8214         55.21         87.75     67.95    60.92    57.36      64.95
## 9672         55.21         87.75     67.95    60.92    57.36      64.95
## 11246        51.04         89.71     70.00    59.04    53.96      65.16
## 11255        51.04         89.71     70.00    59.04    53.96      65.16
## 22           55.21         84.80     63.10    58.89    56.62      61.34
## 31           55.21         84.80     63.10    58.89    56.62      61.34
## 40           55.21         84.80     63.10    58.89    56.62      61.34
## 49           55.21         84.80     63.10    58.89    56.62      61.34
## 58           55.21         84.80     63.10    58.89    56.62      61.34
## 67           55.21         84.80     63.10    58.89    56.62      61.34
## 76           55.21         84.80     63.10    58.89    56.62      61.34
## 1480         55.21         84.80     63.10    58.89    56.62      61.34
## 1489         55.21         84.80     63.10    58.89    56.62      61.34
## 1498         55.21         84.80     63.10    58.89    56.62      61.34
## 1507         55.21         84.80     63.10    58.89    56.62      61.34
## 1516         55.21         84.80     63.10    58.89    56.62      61.34
## 1525         55.21         84.80     63.10    58.89    56.62      61.34
## 1534         55.21         84.80     63.10    58.89    56.62      61.34
## 2938         55.21         84.80     63.10    58.89    56.62      61.34
```

```
summary(resultados)
```

```
##     minCases       trials           CF         winnow           bands
##  Min.   :  0   Min.   : 1.00   Min.   :0.05   Mode :logical   Min.   : 5
##  1st Qu.: 50   1st Qu.:10.00   1st Qu.:0.15   FALSE:5832      1st Qu.: 5
##  Median :100   Median :20.00   Median :0.25   TRUE :5832      Median :10
##  Mean   :100   Mean   :20.11   Mean   :0.25                   Mean   :10
##  3rd Qu.:150   3rd Qu.:30.00   3rd Qu.:0.35                   3rd Qu.:15
##  Max.   :200   Max.   :40.00   Max.   :0.45                   Max.   :15
##                                                               NA's   :2916
##  noGlobalPruning   rules           Exactitud      Sensibilidad
##  Mode :logical   Mode :logical   Min.   :66.00   Min.   : 0.00
##  FALSE:5832      FALSE:2916      1st Qu.:70.33   1st Qu.:23.96
##  TRUE :5832      TRUE :8748      Median :73.33   Median :39.58
##                                  Mean   :72.52   Mean   :33.89
##                                  3rd Qu.:74.33   3rd Qu.:44.79
##                                  Max.   :78.00   Max.   :56.25
##
##  Especificidad      Precisión        F1_Score        F2_Score
##  Min.   : 78.92   Min.   :45.45   Min.   :14.55   Min.   :10.05
##  1st Qu.: 86.76   1st Qu.:61.43   1st Qu.:41.48   1st Qu.:33.88
##  Median : 90.69   Median :63.16   Median :49.70   Median :44.02
##  Mean   : 90.69   Mean   :63.79   Mean   :44.25   Mean   :38.86
##  3rd Qu.: 95.10   3rd Qu.:67.31   3rd Qu.:52.36   3rd Qu.:47.40
##  Max.   :100.00   Max.   :90.00   Max.   :60.92   Max.   :57.36
##                   NA's   :720     NA's   :720     NA's   :720
##    F0.5_Score
##  Min.   :26.32
##  1st Qu.:52.38
##  Median :57.18
```

```
##  Mean   :52.77
##  3rd Qu.:58.82
##  Max.   :67.07
##  NA's   :720
```

# F1 SCORE

```
filtrado <- resultados[resultados$F1_Score > 50, ]
filtrado$bands <- ifelse(is.na(filtrado$bands), 0, filtrado$bands)
filtrado <- na.omit(filtrado)
filtrado <- filtrado[, !(names(filtrado) %in% c("Exactitud", "Sensibilidad", "Especificidad", "Precisió
filtrado <- filtrado[order(-filtrado$F1_Score), ]
```

### Haz un resumen de las principales conclusiones de todos los análisis y modelos realizados

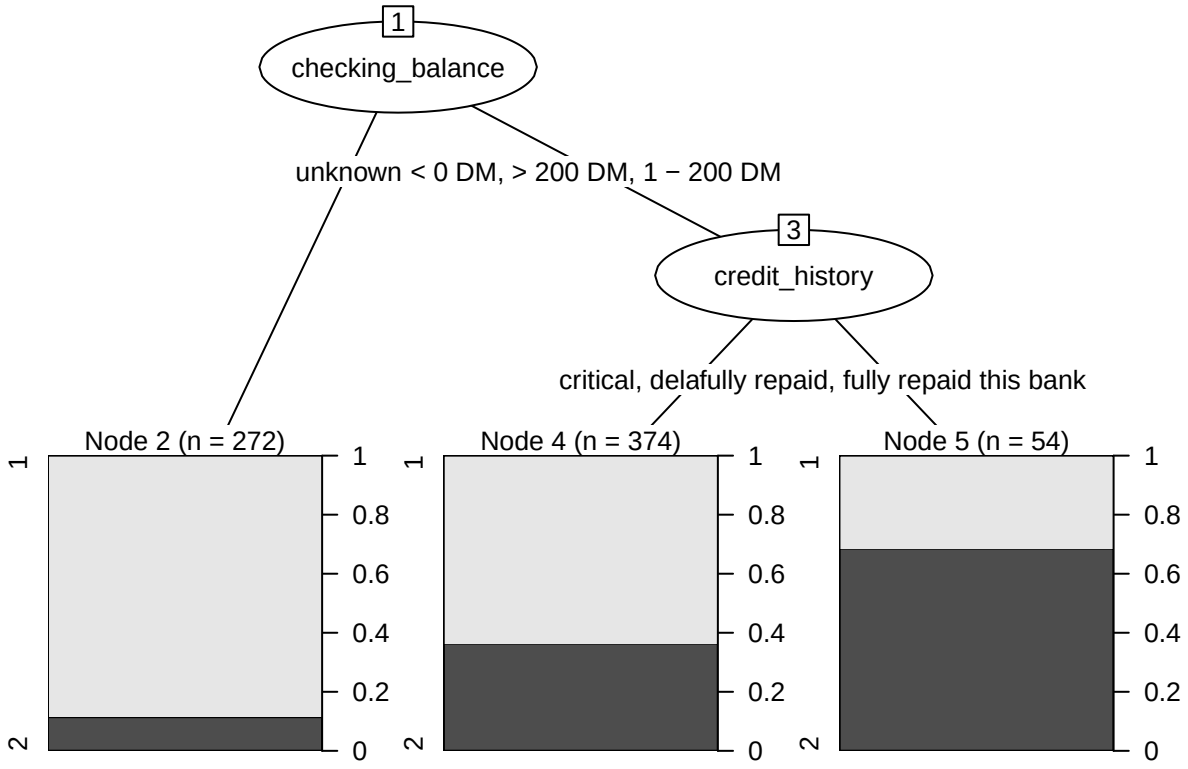
En un modelo de predicion de impago nos conviene seguramente un equilibrio entre errores tipo I y II, con un ajuste personalizado dependiendo de la necesidad natural de una entidad bancaria a dar prestamos, es decir, si no hay mucha liquidez, hay una restriccion bancaria y podriamos pensar que el modelo deberia ser mas restrictivo para ello. Tambien puede ser importante en otro momento, de bonanza economica, ser mas permisivo, para no perder oportunidades de negocio. Entonces teniendo en cuenta ello, estando en 2024 y no en 2012, donde en general hay una bonanza economica comparable y que el banco puede permitirse mas impagos, un punto de vista interesante es darle mas importancia a la *precision*, minimizando los falsos negativos, si estamos en un contexto de situacion economica desfavorable quizas tiene mas sentido tener menos impagos, es decir un *recall* mas alto. Ya no depende solamente de la situacion economica, sino por ejemplo la tasa de impago en la entidad que hace el modelo, si es muy alta va a necesitar arriesgar menos, si es baja, podra arriesgar mas. Por tanto es interesante comparar *precision vs recall*, y un ejemplo en este caso podria ser un modelo *f0.5 score* en caso de querer riesgo o *f2 score* en caso de buscar mas prudencia. Ademas es muy costoso en terminos de tiempo y dinero, hacer un modelo que no sea interpretable, por lo que es importante tener en cuenta la interpretabilidad del modelo, y en este caso, un arbol de decision es una buena opcion, ya que es facil de interpretar y de explicar a la parte de negocio. Tambien es muy costoso en tiempo de computacion hacer un modelo en base a pruebas como he hecho (11664 pruebas de modelos, y tuve que reducir ya que era mucho tiempo), que si bien es correcto en terminos academicos, en entornos de produccion quizas no es la mejor opcion, ya que no se trabaja con 700 registros, sino, al menos con miles de registros. Por tanto, en resumen, un modelo de arbol de decision, con un f1 score de 60,92, si bien hay algunos F0.5 Score de 65.16 y F2 Score de 57.36, con un minimos casos (minCases) de 125 pasadas (trials) de 15 ajuste (CF) de 0.15 filtro de variables no importantes (winnow) en FALSE division de variables no numericas (bands) en 5,10 o 15 poda global (noGlobalPruning) en FALSE reglas (rules) en TRUE Sin usar reglas, minCases 100 , trials 35, ajuste 0.20, con F1 de 59.04 F2 de 53.96 y F0.5 score de 65.16 Son los mejores modelos para este caso, tanto si se usan reglas como si no, ya que son interpretables, tienen un buen f1 score, y son un modelo que se puede ajustar a las necesidades de la entidad bancaria que lo utilice. A continuacion visualizo el modelo final sin usar reglas, donde podemos ver como se divide el arbol, siendo el primer nodo checking_balance:

```
head(filtrado, n=5)
```

```
##       minCases trials   CF winnow bands rules F1_Score F2_Score F0.5_Score
## 6756       125     15 0.15  FALSE     5  TRUE    60.92    57.36      64.95
## 8214       125     15 0.15  FALSE    10  TRUE    60.92    57.36      64.95
## 9672       125     15 0.15  FALSE    15  TRUE    60.92    57.36      64.95
## 11246      100     35 0.20  FALSE     0 FALSE    59.04    53.96      65.16
## 11255      100     40 0.20  FALSE     0 FALSE    59.04    53.96      65.16
```

```
modelo <- C5.0(
  x = train[, -which(names(train) == columna_objetivo)],
```

```
  y = train[[columna_objetivo]],
  trials = 15,
  control = {
    C5.0Control(
      minCases = 125,
      CF = 0.15,
      winnow = FALSE,
      earlyStopping = TRUE,
      noGlobalPruning = FALSE
    )
  },
  rules = FALSE
)

plot(modelo, gp = gpar(fontsize = 9.5))
```




```
C5imp(modelo)
```

```
##                     Overall
## checking_balance      100.00
## credit_history        100.00
## savings_balance       100.00
## employment_length     100.00
## other_debtors          82.86
## months_loan_duration   79.29
## existing_credits       71.57
```

```
## telephone              67.29
## housing                55.29
## purpose                 0.00
## amount                  0.00
## installment_rate        0.00
## personal_status         0.00
## residence_history       0.00
## property                0.00
## age                     0.00
## installment_plan        0.00
## dependents              0.00
## foreign_worker          0.00
## job                     0.00
```

```
predicciones_test <- predict(modelo, newdata = test)

matriz_confusion <- table(Predicted = predicciones_test, Actual = test$default)

print(matriz_confusion)
```

```
##           Actual
## Predicted   1   2
##         1 186  62
##         2  18  34
```

En la matriz de confusión observamos que hay 62 errores de tipo I y 18 errores de tipo II. Esto es aceptable en un contexto donde el enfoque está en aumentar el riesgo en la concesion de creditos, pero no seria aceptable en un entorno de maxima prudencia. Sin embargo, si quisiéramos priorizar la reducción de los errores de tipo II, podríamos ajustar el modelo para que favorezca lo contrario, ya que he hecho suficientes pruebas para elegir un modelo con un F2 score (los mas altos tienen 57.36).

# F2 SCORE

```
filtrado <- resultados[resultados$F2_Score > 50, ]
filtrado$bands <- ifelse(is.na(filtrado$bands), 0, filtrado$bands)
filtrado <- na.omit(filtrado)
filtrado <- filtrado[, !(names(filtrado) %in% c("Exactitud", "Sensibilidad", "Especificidad", "Precisió
filtrado <- filtrado[order(-filtrado$F2_Score), ]
head(filtrado, n=5)
```

```
##      minCases trials   CF winnow bands rules F1_Score F2_Score F0.5_Score
## 6756      125     15 0.15  FALSE     5  TRUE    60.92    57.36      64.95
## 8214      125     15 0.15  FALSE    10  TRUE    60.92    57.36      64.95
## 9672      125     15 0.15  FALSE    15  TRUE    60.92    57.36      64.95
## 418        75      5 0.30   TRUE     5  TRUE    58.70    57.20      60.27
## 499        75      5 0.35   TRUE     5  TRUE    58.70    57.20      60.27
```

# F0.5 SCORE

Si queremos usar como referencia F0.5 score (ya que el primer filtrado era con F1 Score), podemos usar estas configuraciones, donde tenemos incluso valores de 66 hasta 67.07:

```
filtrado <- resultados[resultados$F0.5_Score > 50, ]
filtrado$bands <- ifelse(is.na(filtrado$bands), 0, filtrado$bands)
```

```r
filtrado <- na.omit(filtrado)
filtrado <- filtrado[, !(names(filtrado) %in% c("Exactitud", "Sensibilidad", "Especificidad", "Precisió
filtrado <- filtrado[order(-filtrado$F0.5_Score), ]
head(filtrado, n=5)
```

```
##       minCases trials   CF winnow bands rules F1_Score F2_Score F0.5_Score
## 5657       100     35 0.35  FALSE     0 FALSE    57.14    49.77      67.07
## 11237      100     30 0.20  FALSE     0 FALSE    58.23    51.57      66.86
## 7187       100     30 0.40  FALSE     5  TRUE    55.63    47.84      66.46
## 7268       100     30 0.45  FALSE     5  TRUE    55.63    47.84      66.46
## 8645       100     30 0.40  FALSE    10  TRUE    55.63    47.84      66.46
```

‘‘‘