German Card Credit

David de Vega Martin

Minería de datos: PEC3 - Clasificación con árboles de decisión

DAVID DE VEGA MARTIN - ddevega

Noviembre 2021

Contents

Introducción

Carga de librerias:

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

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

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

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

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

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

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

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

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

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

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

if(!require(reshape2)){
    install.packages('reshape2', repos='http://cran.us.r-project.org')
    library(reshape2)}
```

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

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

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

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

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

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

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

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

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

Carga de los datos

```
data<-read.csv("./DATOS/credit.csv",header=TRUE, sep=",",stringsAsFactors=TRUE)
attach(data)
```

Funciones

Funcion para el calculo de las graficas

```
crear_grafico <- function(data, variable_comp, titulo_grafico_A,
                          titulo_grafico_B) {

  # Renombrar las tablas
  Tabla_A <- suppressMessages(

    data %>%
    group_by({{ variable_comp }}) %>%
    summarise(Total = n()) %>%
    mutate(Porcentaje = round(Total / sum(Total) * 100, 0))
  )

  Tabla_B <- suppressMessages(

    data %>%
    group_by({{ variable_comp }}, default) %>%
    summarise(Total = n()) %>%
    mutate(Porcentaje = round(Total / sum(Total) * 100, 0))
  )

  # Crear el primer gráfico
  grafico_A <- ggplot(Tabla_A, aes(x = {{ variable_comp }}, y = Total, fill =
                                     {{ variable_comp }})) +
    geom_bar(width = 0.9, stat = "identity", position = position_dodge()) +
    ylim(c(0, 1000)) +
    labs(x = NULL, y = "Frecuencia \n (Porcentajes)", title = titulo_grafico_A) +
    labs(fill = NULL) +
    geom_text(aes(label = paste0(Total, " ", "(", Porcentaje, "%", ")")),
              vjust = 0.3,
              color = "black",
              hjust = -0.05,
              position = position_dodge(0.9),
              angle = 0,
              size = 4.0) +
    theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
          legend.position = "None") +
    coord_flip()

  # Crear el segundo gráfico
  grafico_B <- ggplot(Tabla_B, aes(x = {{ variable_comp }}, y = Total,
                                   fill = default)) +
    geom_bar(width = 0.9, stat = "identity", position = position_dodge()) +
     ylim(c(0, 1000)) +
    labs(x = NULL, y = "Frecuencias \n (Porcentajes)", title = titulo_grafico_B) +
    labs(fill = "Default") +
    geom_text(
      aes(label = paste0(Total, " ", "(", Porcentaje, "%", ")")),
      hjust = -0.2,
      color = "black",
      size = 4.0,
```

```
      position = position_dodge(0.9),
      angle = 0
    ) +
    scale_fill_discrete(name = "Default", labels = c("Bueno ", "Malo")) +
    theme(axis.text.y = element_blank(),
          axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
          legend.position = "right") +
    coord_flip()

  invisible(
    grid.arrange(grafico_A + theme(plot.title =
                                     element_text(hjust = 0.5)),
                 grafico_B + theme(plot.title =
                                     element_text(hjust = 0.5)),
                 ncol = 2, nrow = 1)
  )
  }
```

Calcular performance:

```
calculate_performance <- function(model_description, confusion_matrix) {

  TP <- confusion_matrix[1,1]
  FP <- confusion_matrix[1,2]
  FN <- confusion_matrix[2,1]
  TN <- confusion_matrix[2,2]

  accuracy    <- round ((TP + TN) /(TP + TN + FN + FP),3)
  error <- round( (FN + FP)/(TP + TN + FN + FP),3)
  sensitivitie <- round (TP/(TP + FP),3)
  specificitie<- round (TN/(TN + FN),3)
  precision   <- round (TP/(TP + FP),3)

  #output_row <- data.frame(model_description,TP,FP,FN,TN,accuracy,error,
  #sensitivitie,specificitie,precision)
  output_row <- data.frame(model_description,TP,FP,FN,TN,accuracy,error)
  return(output_row)
}
```

Análisis inicial: Exploración de la base de datos

En primer lugar se calculan las dimensiones de la base de datos y se analizan que tipos de atributos presenta

El dataframe presenta 1000 registros o clientes (filas) y 21 variables (columnas).

```
dim(data)
```

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

Se comrprueba la ausencia de valores faltantes.

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

```
## [1]  0 21
```

No hay que hacer ajustes en el dataframe de preparacion. No hay ni datos ausentes ni nulos, en las 21 variables.

Descripcion de variables:

Encontramos información importante en el fichero german que nos descargamos de la página de la UCI:

- checking_balance: Saldo de la cuenta. Basado en el sueldo de al menos un año.
- months_loan_duration: Duración del préstamo en meses.
- credit_history: Historia crediticia. Préstamos pagados o impagados en este y otros bancos.
- purpose: Propósito del préstamo, es decir, a qué está destinado.
- amount: Importe del préstamo.
- savings_balance: Saldo de ahorros.
- employment_length: Duración del empleo en años.
- installment_rate: Tasa de apalancamiento en porcentaje sobre el sueldo disponible.
- personal_status: Estado civil y género.
- other_debtors: Otros deudores o garantes.
- residence_history: Años en su actual residencia.
- property: Tipo de propiedad
- age: Edad
- installment_plan: Otros planes de apalancamiento.
- housing: Régimen de vivienda.
- existing_credits: Número de créditos existentes en este banco.
- default: Deuda. Es la clasificación del riesgo. Toma dos valores: 1 -> Bueno 2 -> Malo
- dependents: Número de personas a su cargo.

- telephone: Teléfono a su nombre (Sí o No).
- foreign_worker: Trabajador extranjero(Sí o No).
- job: Tipo de empleo.

```
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    : int  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   : int  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    : 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           : 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 ...
```

```
summary(data)
```

```
##    checking_balance months_loan_duration                credit_history
##  < 0 DM    :274     Min.   : 4.0         critical              :293
##  > 200 DM  : 63     1st Qu.:12.0         delayed               : 88
##  1 - 200 DM:269     Median :18.0         fully repaid          : 40
##  unknown   :394     Mean   :20.9         fully repaid this bank: 49
##                     3rd Qu.:24.0         repaid                :530
##                     Max.   :72.0
##
##        purpose        amount           savings_balance  employment_length
##  radio/tv  :280   Min.   :  250   < 100 DM     :603    > 7 yrs   :253
##  car (new) :234   1st Qu.: 1366   > 1000 DM    : 48    0 - 1 yrs :172
##  furniture :181   Median : 2320   101 - 500 DM :103    1 - 4 yrs :339
##  car (used):103   Mean   : 3271   501 - 1000 DM: 63    4 - 7 yrs :174
##  business  : 97   3rd Qu.: 3972   unknown      :183    unemployed: 62
##  education : 50   Max.   :18424
##  (Other)   : 55
##  installment_rate      personal_status      other_debtors residence_history
##  Min.   :1.000    divorced male: 50    co-applicant: 41   Min.   :1.000
##  1st Qu.:2.000    female       :310    guarantor   : 52   1st Qu.:2.000
##  Median :3.000    married male : 92    none        :907   Median :3.000
```

```
##  Mean   :2.973    single male  :548                      Mean   :2.845
##  3rd Qu.:4.000                                          3rd Qu.:4.000
##  Max.   :4.000                                          Max.   :4.000
##
##                      property        age        installment_plan     housing
##  building society savings:232   Min.   :19.00   bank  :139       for free:108
##  other                    :332   1st Qu.:27.00   none  :814       own     :713
##  real estate              :282   Median :33.00   stores: 47       rent    :179
##  unknown/none             :154   Mean   :35.55
##                                  3rd Qu.:42.00
##                                  Max.   :75.00
##
##  existing_credits    default      dependents    telephone  foreign_worker
##  Min.   :1.000    Min.   :1.0   Min.   :1.000   none:596   no : 37
##  1st Qu.:1.000    1st Qu.:1.0   1st Qu.:1.000   yes :404   yes:963
##  Median :1.000    Median :1.0   Median :1.000
##  Mean   :1.407    Mean   :1.3   Mean   :1.155
##  3rd Qu.:2.000    3rd Qu.:2.0   3rd Qu.:1.000
##  Max.   :4.000    Max.   :2.0   Max.   :2.000
##
##                       job
##  mangement self-employed:148
##  skilled employee       :630
##  unemployed non-resident: 22
##  unskilled resident     :200
##
##
##
```

Analisis exploratorio de datos (EDA) y tratamiento de variables

El objetivo de este PEC, es un problema de evaluacion de la calidad crediticia. Recogida en la variable default, variable cualitativa dicotomica donde: 1 es calidad crediticia buena. 2 calidad crediticia mala.

Comparando esta variable defaults se procedera a evaluar cada una de las variables. Estas se agruparan en varias categorias para mejorar la comprension global de las mismas.

El primer objetivo es obtener un perfil del tipo de credito que otorga este banco, empleando las siguientes variables:

- Default, calidad creditia.
- Duration in month, duracion en meses.
- Credit Amount, importe del credito.
- Purpose, objeto por el que se pide la financiacion.
- Number of existing credit at this bank, numero de otros creditos en el banco.

El paso siguiente es elaborar un perfil del cliente de este banco. Para ello abordaremos el problema desde un triple prisma: situacion personal, patrimonial y financiera.

Para evaluar su situacion personal, se emplearan las siguientes variables:

- Personal status and sex, sexo y estado civil.
- Age , edad.
- Job, puesto de trabajo.
- Present employment since, tiempo que lleva en el trabajo actual.
- Foreign worker, si el cliente es nacional o extranjero.
- Dependents, personas a su cargo.
- Telephone si tiene telefono o no.

En la elaboracion del perfil patrimonial se abordara el analisis de las siguientes variables:

- Housing, regimen de tenencia de la vivienda.
- Property, tipo de propiedad
- Present residence since, tiempo que lleva el cliente en su residencia actual.

Finalmente se obtendra un perfil crediticio del cliente, empleando las variables:

- Status of existing checking account, saldo de la cuenta bancaria del cliente.
- Credit History, historial creditio.
- Saving account, cuenta corriente saldo.
- Installment rate in percentage of disposible income, apalancamiento de la operacion en funcion de los ingresos del cliente.
- Other debstor, otras deudas.
- Other installment plans, otros planes de compra con pago aplazado.

Perfil del tipo de credito concedido

Calidad crediticia: default

Es una variable factor, tiene dos niveles.

- Nivel 1 que podriamos denominar creditos de buena calidad.
- Nivel 2 que serian creditos de calidad mala o dudosa.

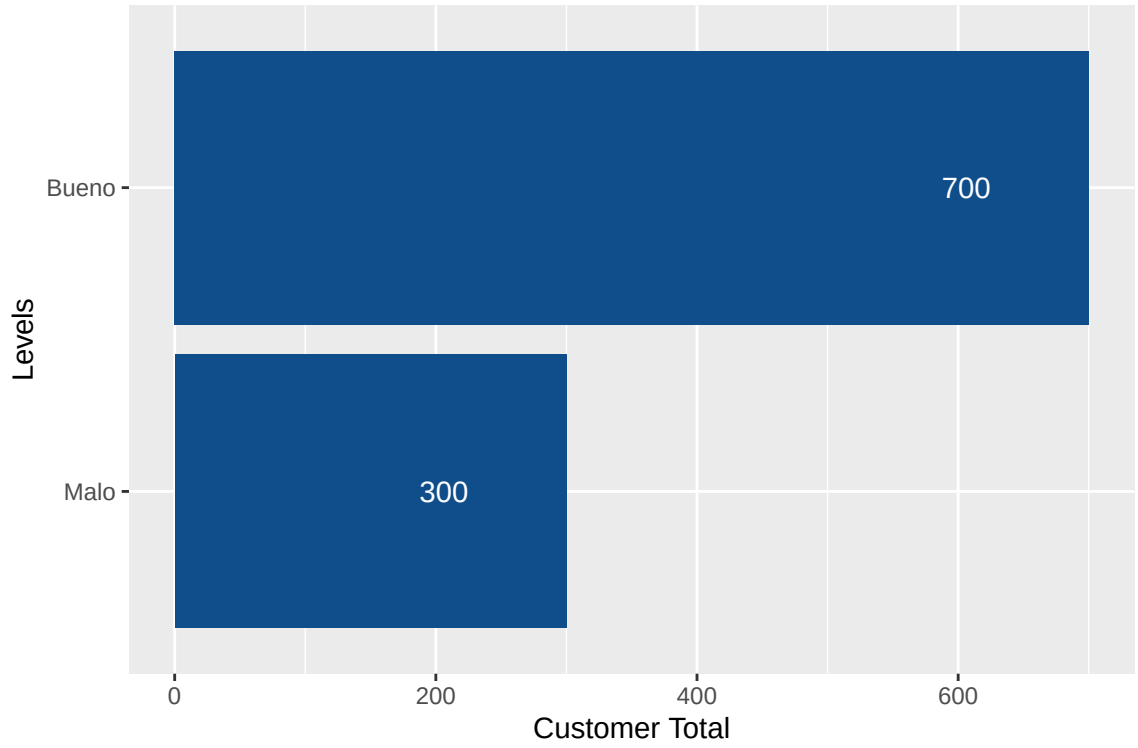
Puede observarse como el 70% de los creditos son de buena calidad. Para facilitar la interpretacion de la variable a lo largo del estudio, se cambian los valores de los niveles a Bueno/Malo.

Esta variable sera la que emplee para comparar con las demas. Asimismo sera la variable a predecir en los modelos que se desarrollaran en la ultima parte.

```
data$default <- ifelse(data$default == "1","Bueno","Malo")

data$default <- as.factor(data$default)
```

```
ggplot(data,
       aes(y = reorder(factor(default), factor(default), function(y) length(y))
)) +
  geom_bar(fill = "dodgerblue4") +
  theme(legend.position = "None") +
  geom_text(stat = 'count', aes(label = after_stat(count)), vjust = 0.5,
            hjust = 3, color = "white") +
  labs(x = "Customer Total", y = "Levels")
```



Pasemos ahora a determinar que tipo de creditos concede nuestro banco y que perfil de cliente tiene.

Historial crediticio: credit_history.

Es una varible factor con 5 niveles:

- Repaid: El cliente atiende a los pagos con normalidad.
- Full/repaid this bank : El cliente atiende a los pagos con normalidad para el banco que se esta estudiando.
- Fully repaid: El cliente ha pagado anteriores creditos completamente
- Delayed : El cliente ha tenido retrasos en los pagos anteriormente.
- Critical : Cuenta en estado critico por retrasos o por otros creditos existentes, no en este banco.

El factor se ordena, de este modo, de mayor a menor nivel de solvencia del cliente con repaid y critical como extremos de los mismos.

Obtenemos un barplot con los niveles del factor y la proporcion de creditos buenos y malos de cada nivel. En lo sucesivo, este sera el metodo de analisis empleado para todas las variables.

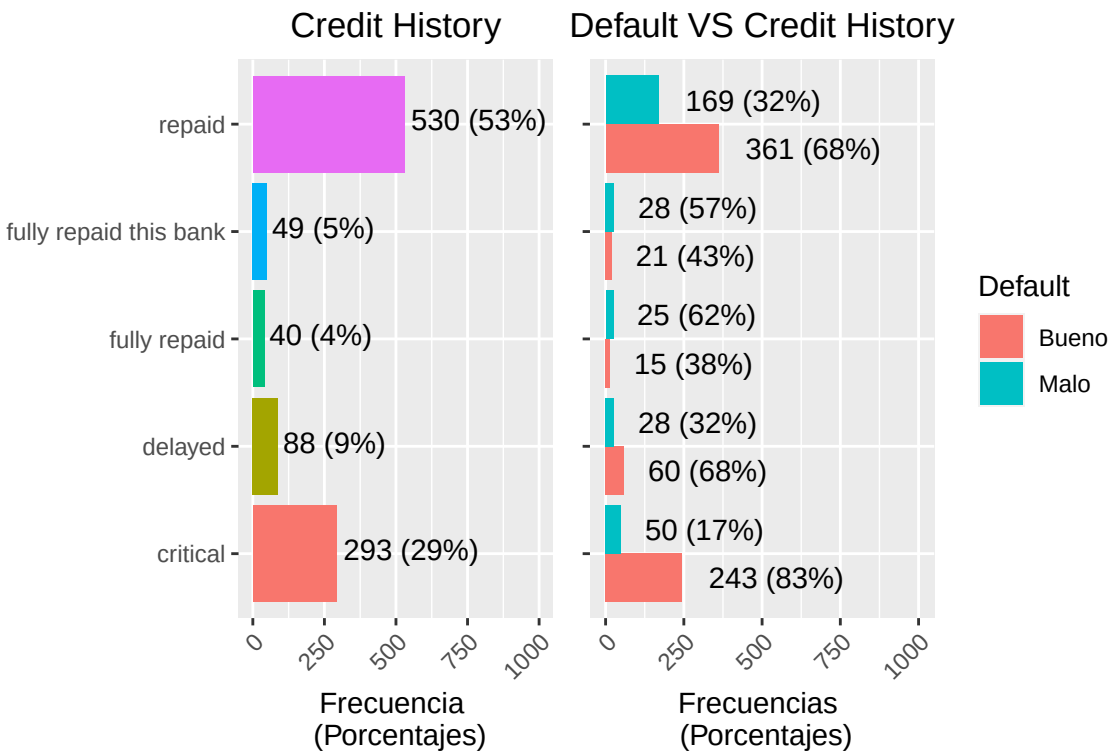
Son creditos solventes en el momento del analisis los pertenecientes a las categorias repaid, fully-repaid this bank, fully repaid, estos 3 niveles suponen un 60% del total de creditos. En la variable Levels teniamos un total de creditos buenos del 70%, esto se debe al alto impacto que tienen los creditos malos enlas categorias fully repaid this bank y fully repaid ambas con un 60%, esto es mas del doble de la proporcion de creditos malos 30% observado en la variableLevels.

Son creditos de dudosa solvencia los pertenecientes a las categorias: delayed y critical. Suponen un 38% del total, un 8% mas de lo obtenido en la variable Levels.

El analisis de las proporciones de tipo de credito en ambas categorias muestra que se encuentran dentro de los niveles de credito general 70-80 % de creditos buenos.

Seria necesario calcular el importe total de creditos para cada categoria del factor, asi como el importe en cada categoria de estado de credito.

```
# Uso de la función
crear_grafico(data , variable_comp = credit_history,
              titulo_grafico_A = "Credit History",
              titulo_grafico_B = "Default VS Credit History")
```



El 62% de los credito estan al corriente de pagos y no suponen problema. EL 38% restante lo forman los delayed y critical, que ya tienen retraso o son insolvente.

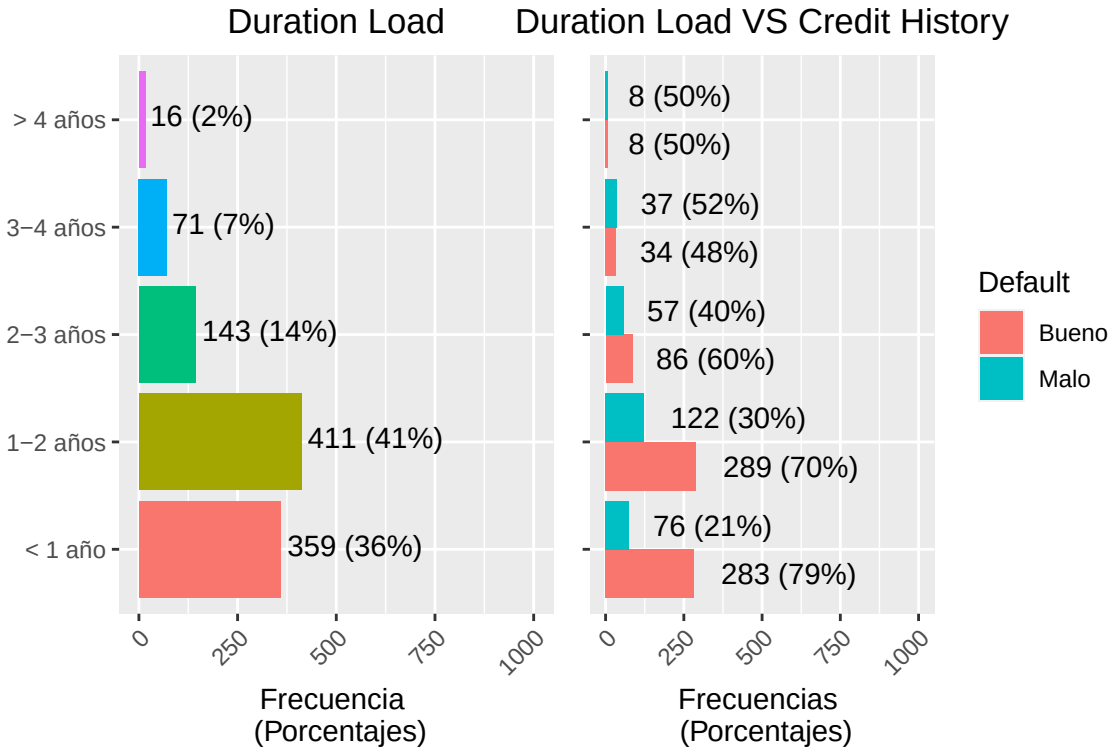
Duracion en meses del prestamo : months_loan_duration .

Tenemos demasiados niveles, tendriamos que discretizar la variable: Para evitar subjetividades de que se considera corto, medio y largo plazo, discretizaremos teniendo en cuenta el numero de meses, en multiplos de año.

Podemos observar que un 75% de los creditos tienen duracion de dos años o menos.

```
data$year_duration_loan <- cut(data$months_loan_duration,
                               breaks = c(0,12,24,36,48,100),
                               labels = c('< 1 año','1-2 años','2-3 años',
                                          '3-4 años','> 4 años' )
                               )
```

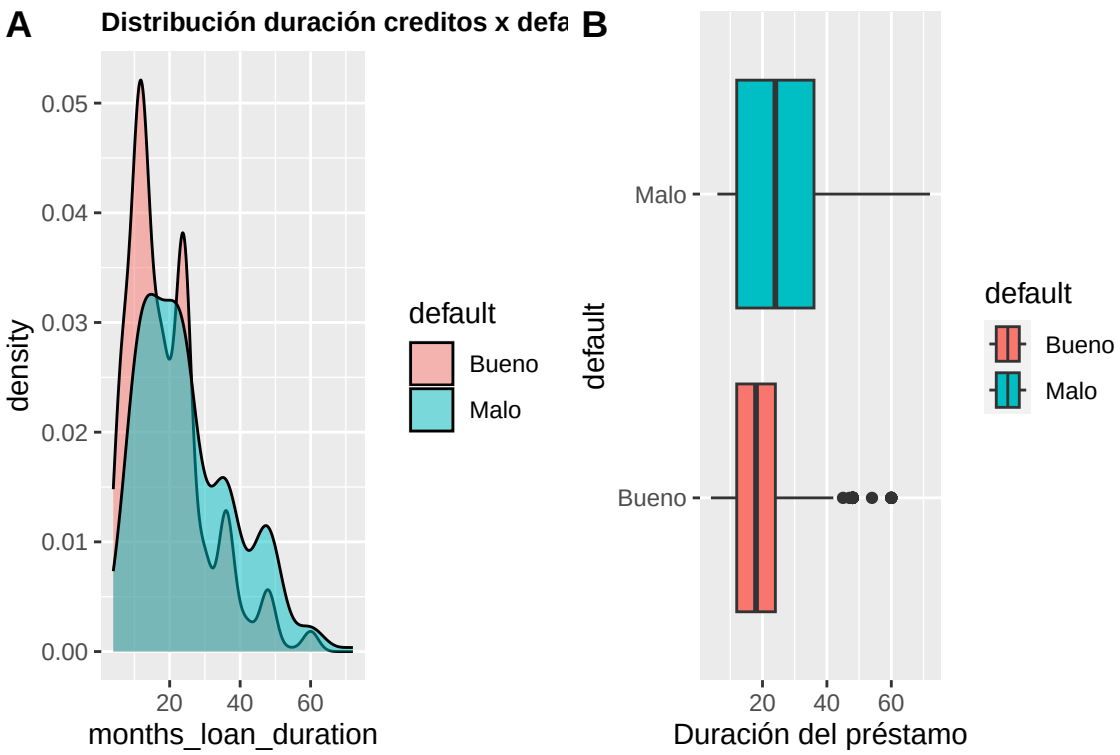
```
# Uso de la función
crear_grafico(data , variable_comp = year_duration_loan,
              titulo_grafico_A = "Duration Load",
              titulo_grafico_B = "Duration Load VS Credit History")
```




```
duration_1 <-
ggplot(data, aes(months_loan_duration, fill=default)) +
        geom_density(alpha=.5)+
        labs(title="Distribución duración creditos x default")+
        theme(plot.title = element_text(size=10,face = "bold"))

duration_2 <-
ggplot(data, aes(x = months_loan_duration,
                 y = default, fill = default)) +
geom_boxplot() + xlab("Duración del préstamo")
```

```
ggarrange(duration_1,duration_2,labels = c("A", "B" ),ncol = 2, nrow = 1)
```



Puede observarse como los creditos malos tienen una duracion mayor que los buenos, en todas las medidas su recorrido intercuartilico es el doble, su mediana e incluso los bigote superior tambien se duplica.

Tendria que examinarse la causa, esto es, si es por la propia naturaleza del mismo, esto es, su periodo de duracion es el establecido inicialmente o si viene motivado por retrasos en las cuotas.

En cualquier caso puede verse como en el caso de los creditos buenos el 77% de observaciones tienen duracion de dos años o menos.

Hay que reseñar que a medida que aumenta la duracion los prestamos aumenta la proporcion de creditos malos.

Para los creditos de duracion menor o igual a dos años se mantiene de media la proporcion 70% buenos que se observo para la variable Level. Sin embargo a medida que aumenta la duracion la proporcion de creditos buenos baja hasta el 50%.

Amount: Cantidad del prestamo.

No tenemos informacion sobre si es el montante original del credito o la parte viva del mismo en el momento de toma de datos.

En cualquier caso es una variable cuantitativa que vamos a representar mediante box plot y barplot.

```
amount_1 <-
ggplot(data, aes( x = factor(0),y= amount, fill = amount)) +
geom_boxplot() + xlab("Valor del préstamo")

amount_2 <-
ggplot(data, aes(y = amount, x = default, fill = default)) +
geom_boxplot() + xlab("Duración del préstamo")
```

Podemos observar como los creditos malos son de mayor importe.

Si discretizamos la variable y calculamos por tramos de 5000 DM obtenemos lo siguiente:

```
data$amount_levels <- cut(data$amount,
                          breaks = c(0,5000,10000,
                                  15000,20000),

                          labels = c('< 5k','5-10 k',
                                     '10-15k','>15' ))
```

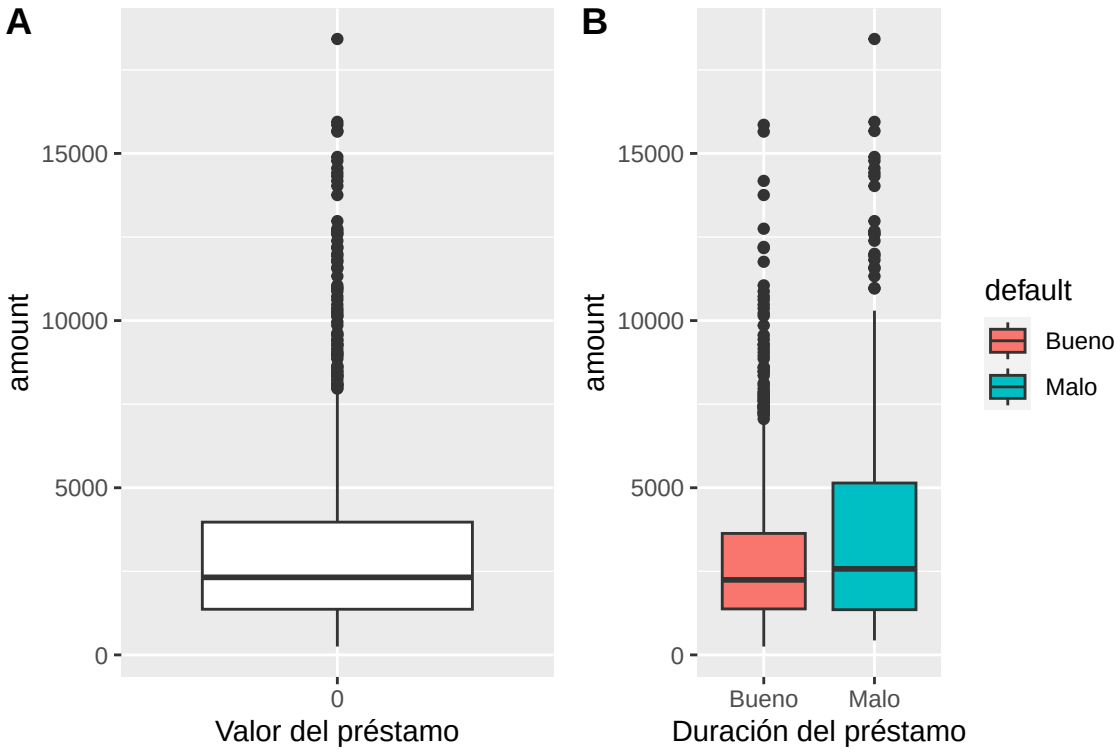
El 80% de los prestamos que otorga este banco son de 5000 marcos o menos. El importe del credito medio se situa alrededor de los 2500 marcos.

Se observa una mayor cuantia en la prestamos malos

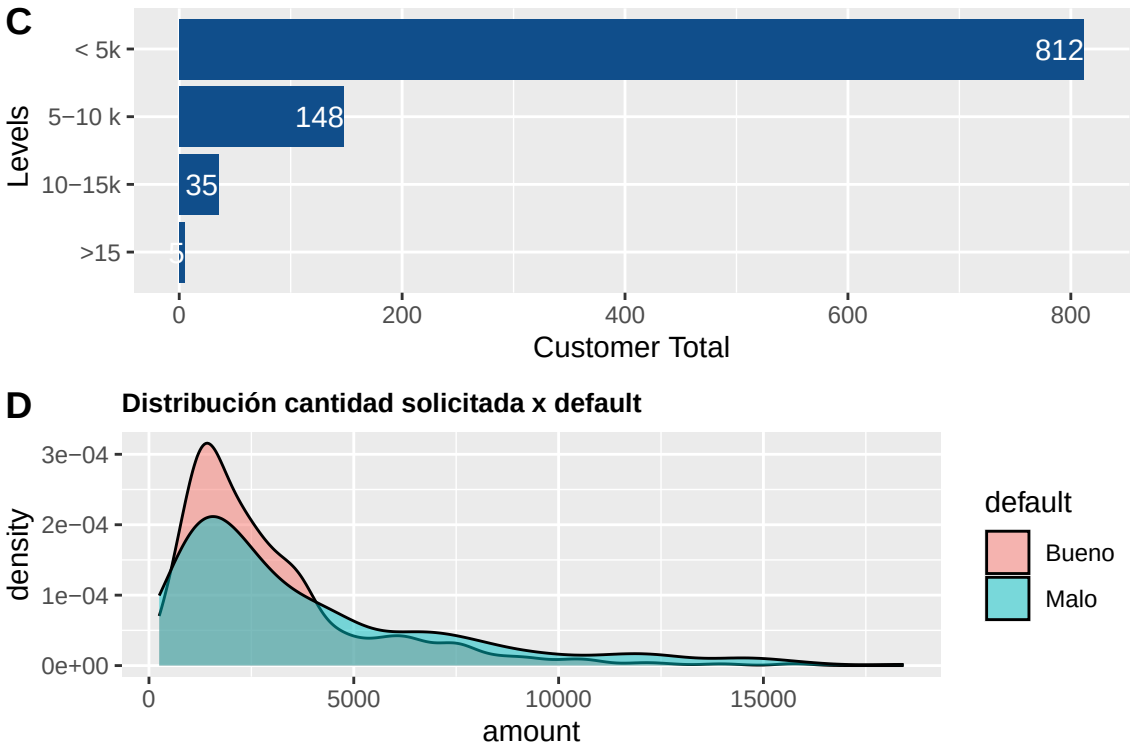
```
amount_3 <-
ggplot(data,
aes(y =reorder( factor(amount_levels),factor(amount_levels),
function(y)length(y))
)) +
geom_bar(fill = " dodgerblue4")+
theme(legend.position = "None")+
geom_text(stat='count', aes(label=..count..), vjust=0.5,
hjust = 1,color ="white")+
labs(x = "Customer Total", y = " Levels")

amount_4 <-
ggplot(data, aes(amount, fill=default)) +geom_density(alpha=.5) +
  labs(title="Distribución cantidad solicitada x default")+
  theme(plot.title = element_text(size=10,face = "bold"))

ggarrange(amount_1 , amount_2,labels = c("A", "B"),ncol = 2, nrow = 1)
```




```
ggarrange(amount_3,amount_4,labels = c("C","D"),ncol = 1, nrow = 2 )
```

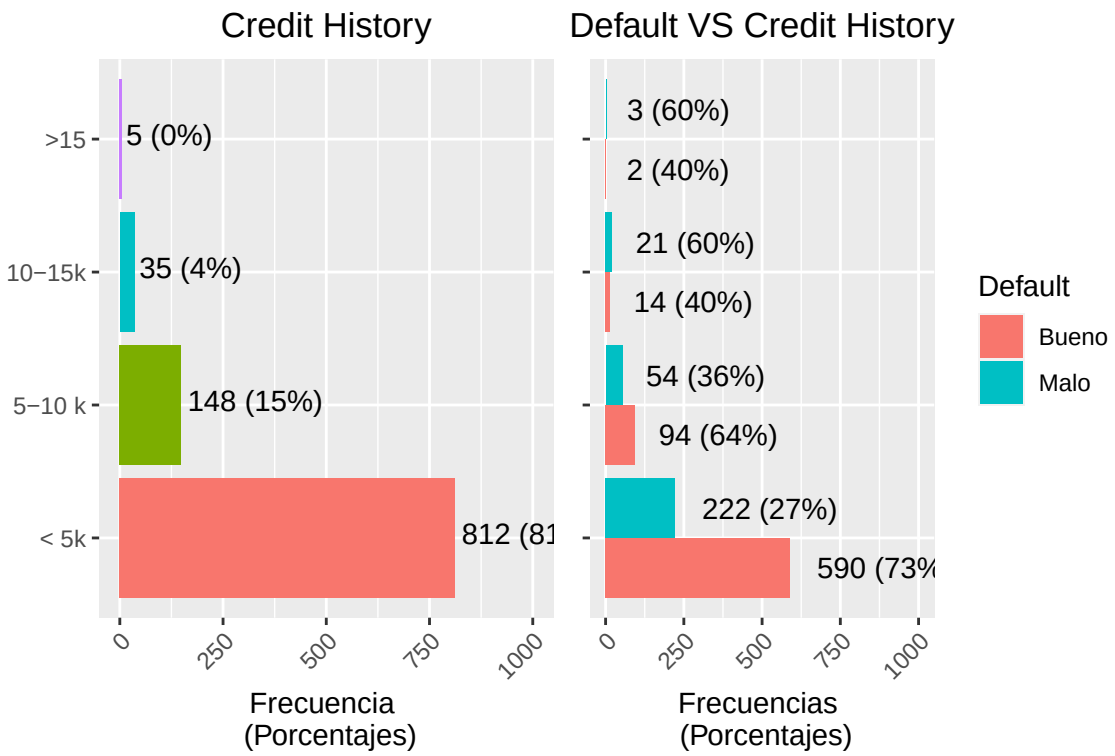


La distribucion es muy similar para cada ambos tipos de credito por default. En el tramo de hasta 5000 marcos se observa una mayor concentracion de creditos malos. A partir de esa zona las distribuciones son

equivalentes.

Los creditos malos, por tanto se concentran en el tramos hasta 5000 marcos.

```
crear_grafico(data , variable_comp = amount_levels,
              "Credit History", "Default VS Credit History")
```



El 96% de los creditos es de 10k o inferior.

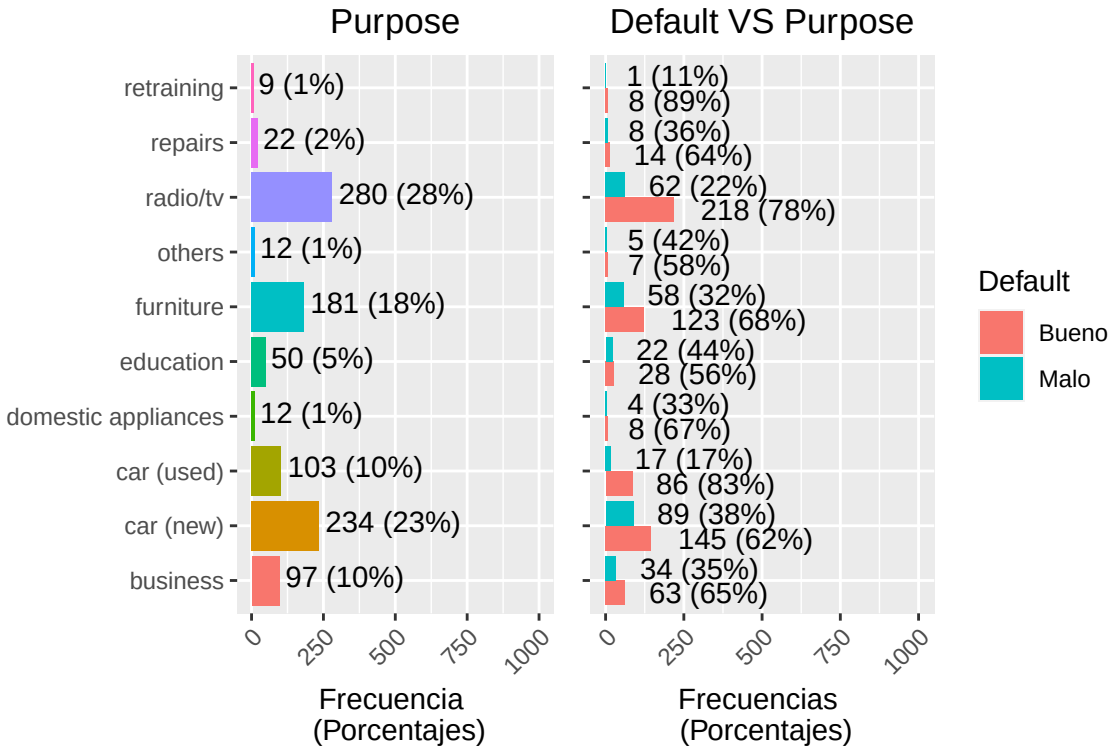
Purpose. Destino del prestamos obtenido.

Esta variable es un factor, con 10 niveles, que recoge en que bienes se materializa el dinero prestado.

Tenemos:

- business: para financiar algun negocio.
- car: adquisicion de un vehiculo ya sea nuevo o de segunda mano.
- domestic appliances : electrodomesticos.
- education : creditos para cursar estudios.
- furniture: adquisicion de mobiliario domestico.
- radio/tv : adquisicion de radio/tv
- repairs : reparaciones domesticas, reformas etc.
- retraining : mejorar-recuperar el historial crediticio.
- others : otros.

```
crear_grafico(data , variable_comp = purpose,
titulo_grafico_A = "Purpose",
titulo_grafico_B = "Default VS Purpose")
```



Como puede apreciarse en la grafica, esta variable cuenta con demasiados niveles para visualizar los datos correctamente, ademas la frecuencia absoluta de cada nivel es muy dispar, lo que complica ademas visualizar las proporciones de cada tipo de credito.

Si puede apreciarse que un tercio del total de creditos son para adquisicion de vehiculos tanto nuevos como usados y las proporciones de tipo de creditos no se aleja demasiado de los niveles de la variable levels.

Casi la mitad de los creditos se conceden para la adquisicion de elementos cotidianos del hogar.

Estas dos categorias suponen el 80% de creditos concedidos.

El resto podria clasificarse como prestamos para financiar negocios o educacion y finalmente una clase que recogiese el resto de tipos.

Puesto que el objeticvo de esta pec es operar arboles de decision y dado que esta variable puede incrementar mucho el tamaño del mismo se propone discretizar la variable en los 4 niveles que se acaban de referir y si llegado el caso fuera variable decisiva se emplearia la variable discretizada.

Conforme a este criterio se obtiene:

Se ha discretizado la variable en 4 categorias:

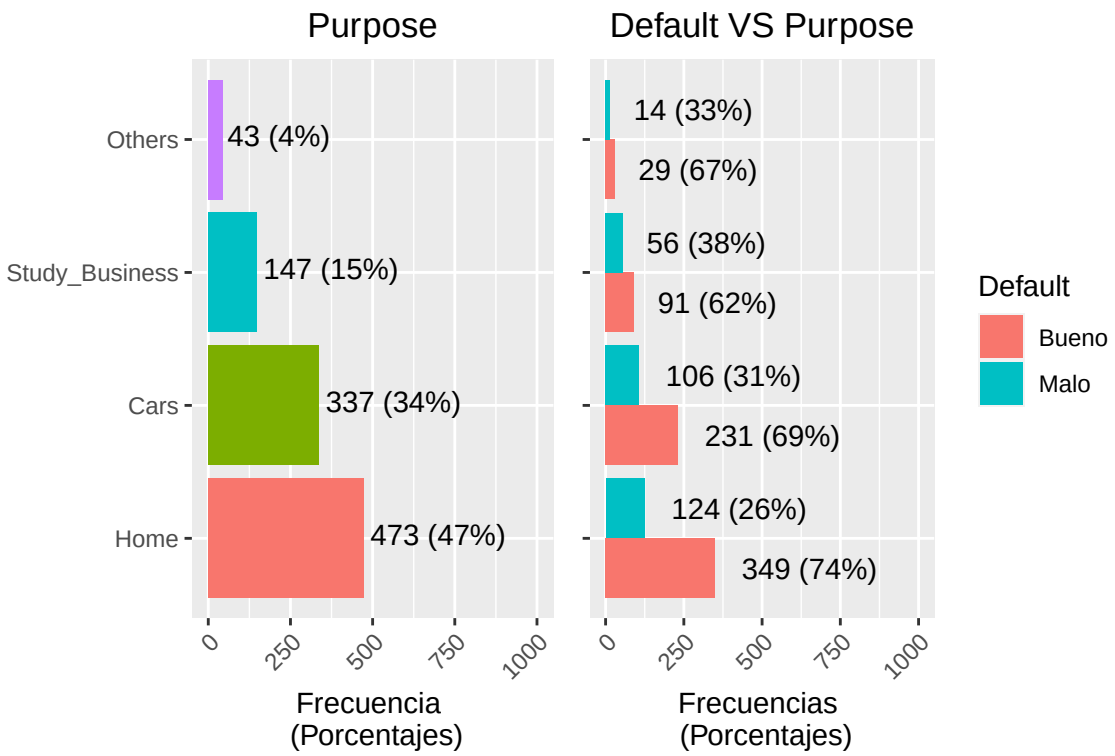
- Home: Se recoge todos los creditos para elementos muebles de uso cotidiano para el hogar, muebles, electrodomesticos, reformas etc Agrupa fruniture, radio/tv, domestic appliances.
- Cars :Adqusicion de vehiculos nuevo o de segunda mano.
- Study/Business : Recoge creditos estudiantiles y aquellos efectuados en negocios.
- Others: Recoge el resto de categorias: retraining, repairs y otras.

```
data$purpose <- as.factor(data$purpose)

data['purpose_levels']<- data['purpose']

levels(data$purpose_levels) <- list(
  Home = c('furniture','radio/tv','domestic appliances'),
  Cars = c('car (used)','car (new)'),
  Study_Business = c('business','education'),
  Others = c('retraining','repairs','others')
)
```

```
crear_grafico(data , purpose_levels,
              "Purpose", "Default VS Purpose")
```



El 81% de los prestamos se concentran en Hogar y Cars.

Alrededor del 70% de los prestamos son de buena calidad independientemente de la categoria, salvo study-business que se reduce a casi el 60%, esto es, se mantienen las proporciones observadas en la variable Levels.

Apalancamiento, en proporcion a los ingresos disponibles: installment_rate in percentage of disposable income.

Es una variable tipo factor con un rango entre 1 y 4. Mide el apalancamiento financiero de la operacion, mediante una escala interna que no se suministra en la documentacion.

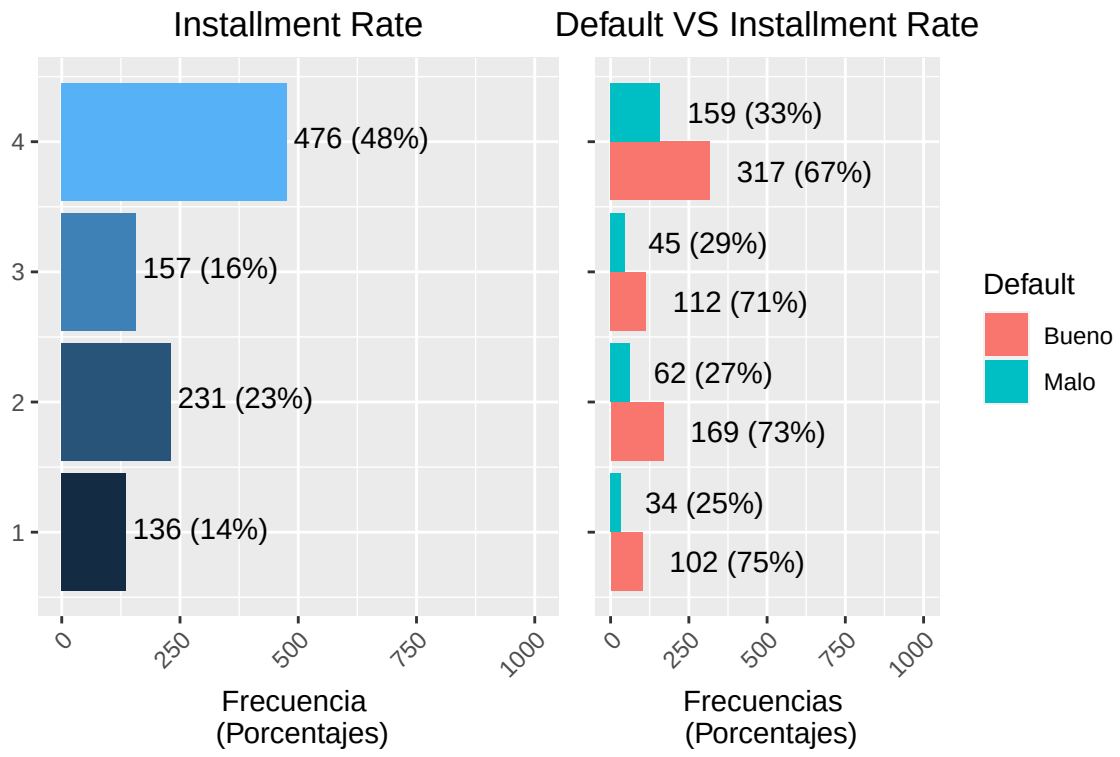
- Nivel 1 .- Poco o ningun apalancamiento.
- Nivel 4 .- Maximo apalancamiento.

Asi el 64% del total de creditos deben considerarse con bastante o maximo apalancamiento.

Esta variable siguen para todos los niveles de factor las proporciones de la variables Levels con un 70% aproximadamente de creditos buenos.

Se desconoce si existe relacion entre esta variable y la que se procedera a evaluar a continuacion, existing_credits.

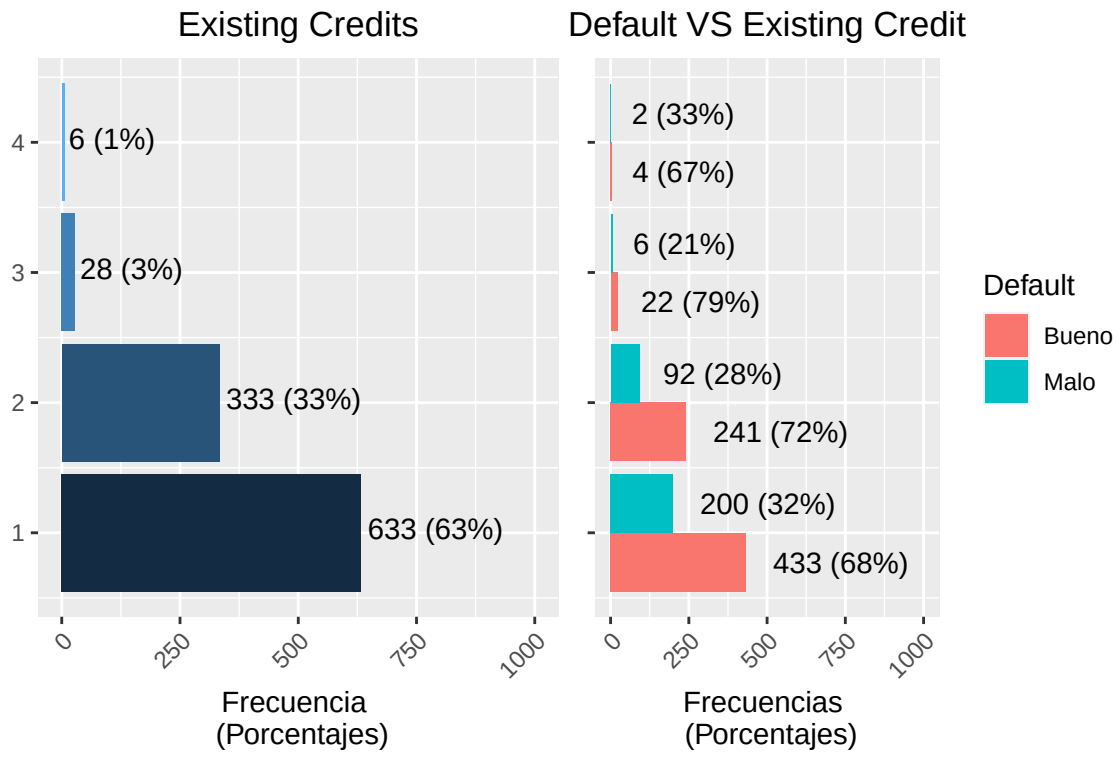
```
crear_grafico(data , variable_comp = installment_rate,
              titulo_grafico_A = "Installment Rate",
              titulo_grafico_B = "Default VS Installment Rate")
```



Creditos existentes en este banco: existing_credits.

Refleja el numero de creditos que ya tiene el cliente con este banco.

```
crear_grafico(data , existing_credits,
              "Existing Credits",
              "Default VS Existing Credit")
```



El 60 % de los clientes tienen un credito con este banco, el 99% un maximo de dos. La proporcion de creditos malos es mas elevada entre los que tienen unicamente un prestamo.

La proporcion entre tipos de credito sigue las proporciones para la variable levels.

Perfil personal del cliente.

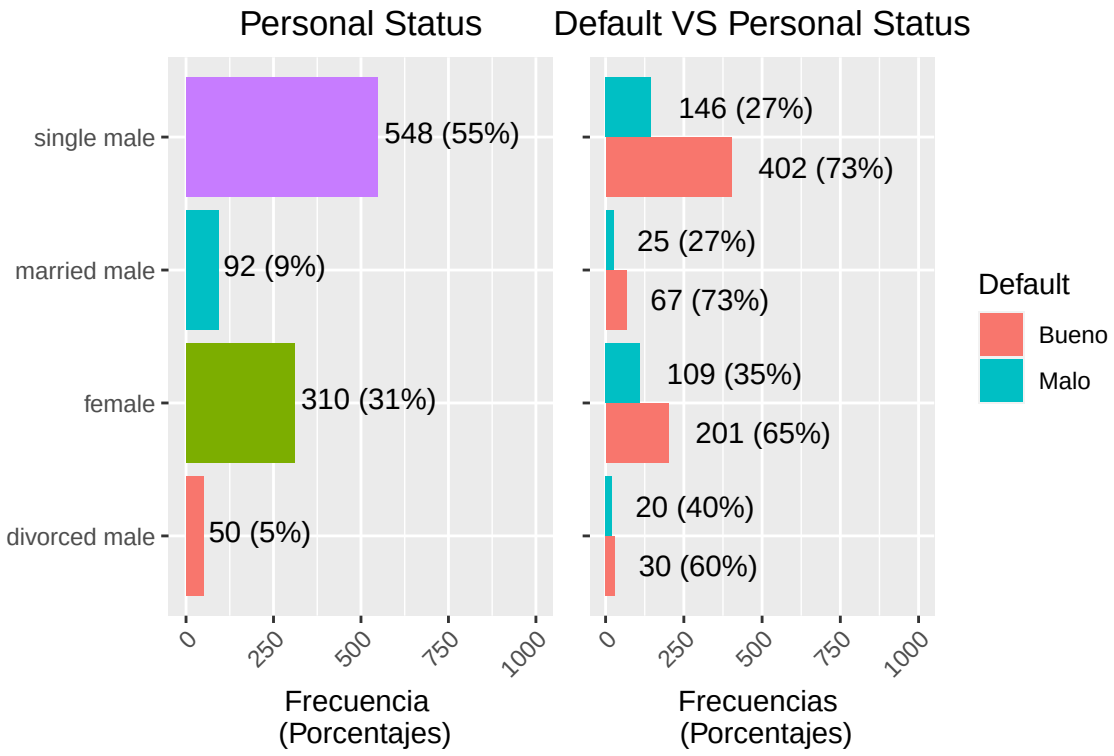
Sexo y Estado civil : Personal_status

La variable consta de 4 niveles:

- Mujer
- Hombres soltero
- Hombres casado
- Hombres divorciado

El 70% de las observaciones son hombres, de los cuales son solteros el 78%. Son el grupo mas solvente. En general, la distribucion respeta las proporciones para la variable levels, a excepcion del grupo de hombre divorciado.

```
crear_grafico(data , personal_status,
              "Personal Status",
              "Default VS Personal Status")
```

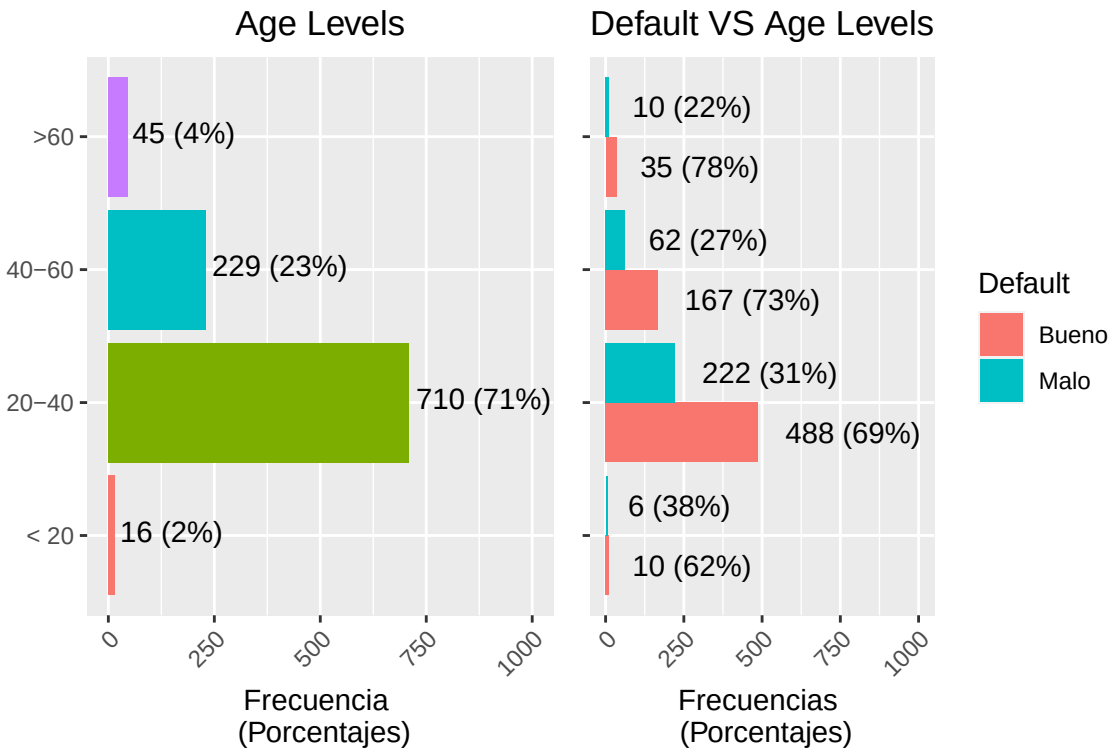


Edad: Age

Se procede a discretizar la variable en grupos de edad de 20 años.

```
data$age <- as.numeric(data$age)
data$age_levels <- cut(data$age, breaks = c(0,20,40,60,100),
labels = c('< 20','20-40','40-60','>60'))

crear_grafico(data , age_levels,
              "Age Levels",
              "Default VS Age Levels")
```

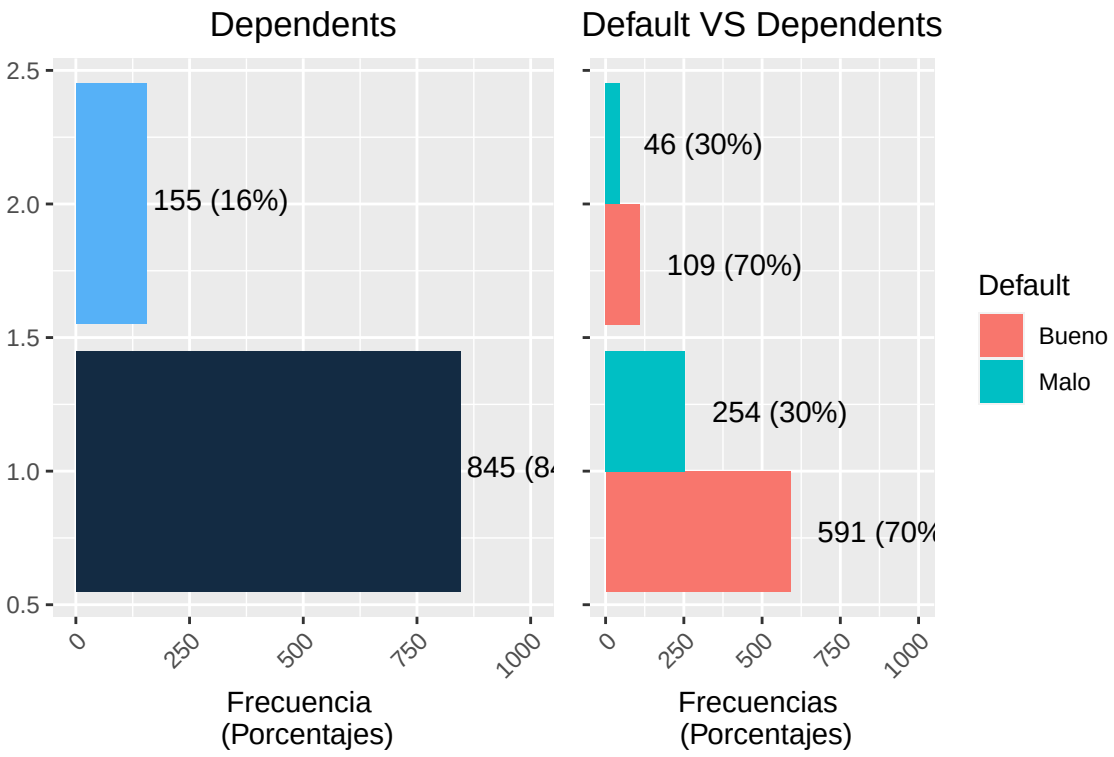


El 94% de los clientes esta comprendido entre los 20 y los 60 años. Nuevamente, los grupos de edad siguen la proporcion de la variable Levels.

Numero de dependientes : Number of dependents

Recoge el numero de personas a cargo del cliente.

```
crear_grafico(data, dependents,
              "Dependents",
              "Default VS Dependents")
```



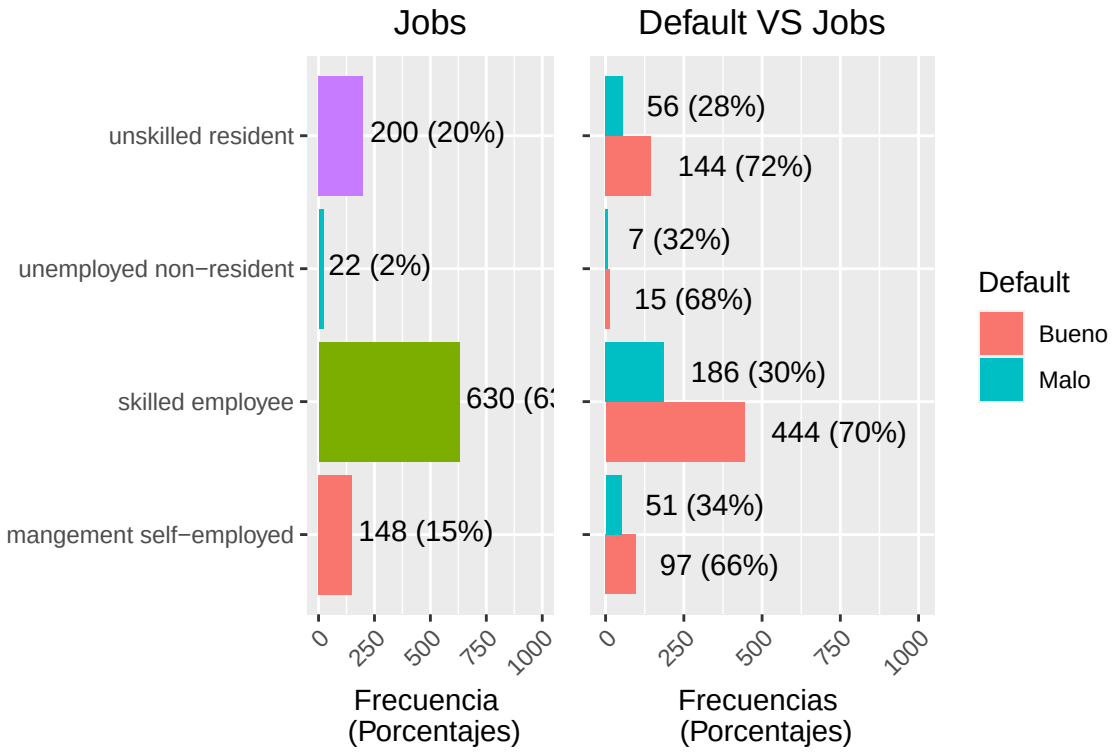
El 85% de los clientes tiene al menos una persona dependiente a su cargo. La proporcion de creditos buenos/manos sigue es la ya vista en Levels.

Trabajo: job.

Factor. Consta de 4 niveles.

- Management self - employed: Trabajadores autonomos
- Skilled employed: Trabajadores especializados.
- Unemployed non-resident:Desempleados no residentes , se entiendes que sin permiso de trabajo.
- Unskilled resident: Trabajadores no especializados.

```
crear_grafico(data, job,
              "Jobs",
              "Default VS Jobs")
```



El 83% de los prestamos estan suscritos por empleados por cuenta ajena cualificados o no.

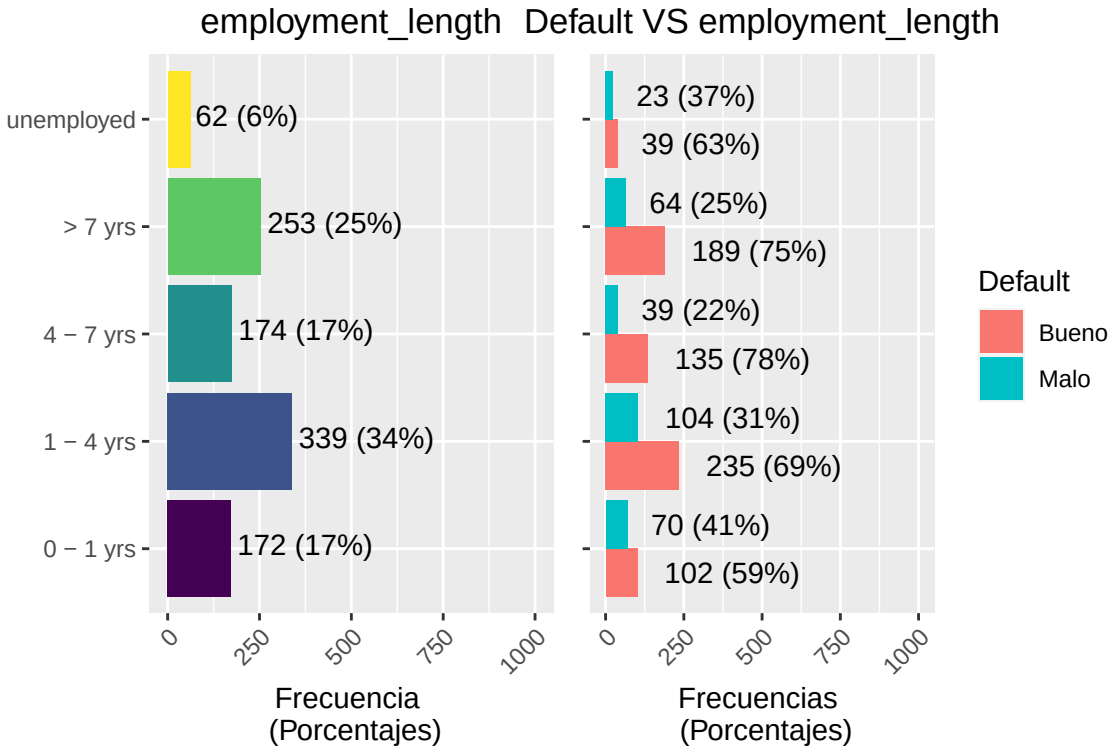
Nuevamente se cumplen las proporciones de prestamos buenos malos de la variable Levels.

Employmente_lenght : Tiempo que lleva el cliente en su empleo.

Se transforma la variable en factor y se procede a ordenarla:

```
data$employment_length <- as.factor(data$employment_length)
data$employment_length <- ordered(data$employment_length,
                                  levels = c("0 - 1 yrs" ,"1 - 4 yrs",
                                             "4 - 7 yrs","> 7 yrs","unemployed"
                                             ))

crear_grafico(data, employment_length, "employment_length",
              "Default VS employment_length")
```



La mitad de los clientes lleva un minimo de 4 años en su puesto de trabajo.

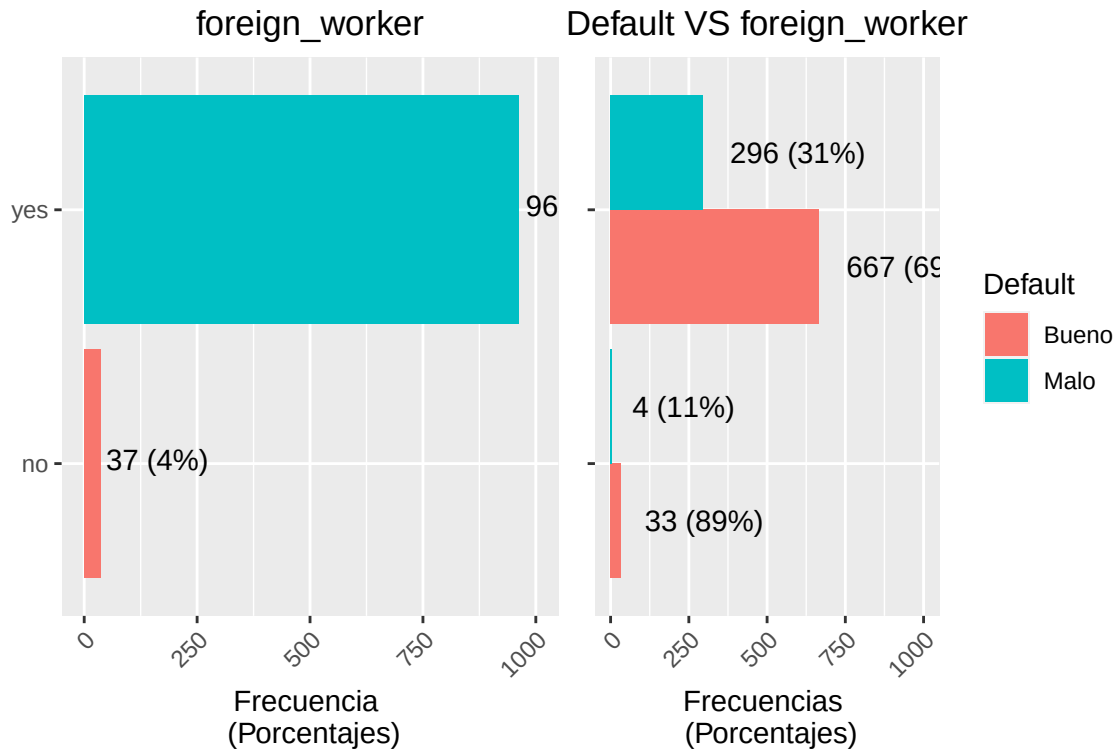
Precisamente, el grupo de hasta un año es el que mayor porbabilidad de creditos malos junto con los desempleados con una proporcion de creditos malos del 40%.

El resto de grupos siguen la proporcion usual de la variable Levels.

Trabajador extranjero : foreign_worker

Factor, dos niveles dicotomica Si/No.

```
crear_grafico(data, foreign_worker, "foreign_worker",
              "Default VS foreign_worker")
```

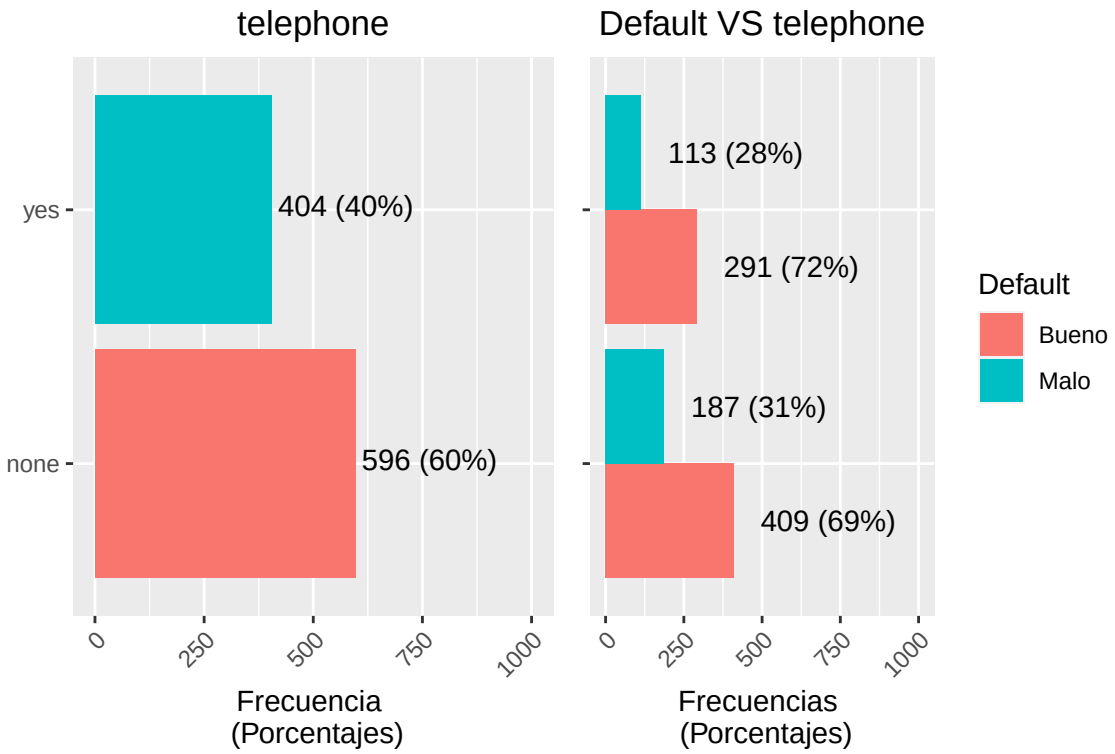


El 96% de los prestatarios es un extranjero. Esta variable puede eliminarse ya que no aporta mas informacion que esa ya que la proporcion de creditos buenos/malos es 70/30% como en la variable Levels.

Telefono : telephone.

Factor dicotomico recoge si el cliente tiene telefono en su domicilio.

```
crear_grafico(data, telephone, "telephone",
              "Default VS telephone")
```



El 60% de los clientes no tiene telefono. Independientemente de ellos ambos grupos respetan la proporcion 70% de creditos buenos de la variable Levels.

Esta variable apenas aporta informacion y seria factible prescindir de la misma.

Perfil situacion patrimonial de los clientes.

Se divide en dos partes:

Activos inmobiliarios:Basicamente la vivienda habitual y su regimen de tenencia.

Activos financieros como son saldos de cuenta etc

Patrimonio inmobiliario.

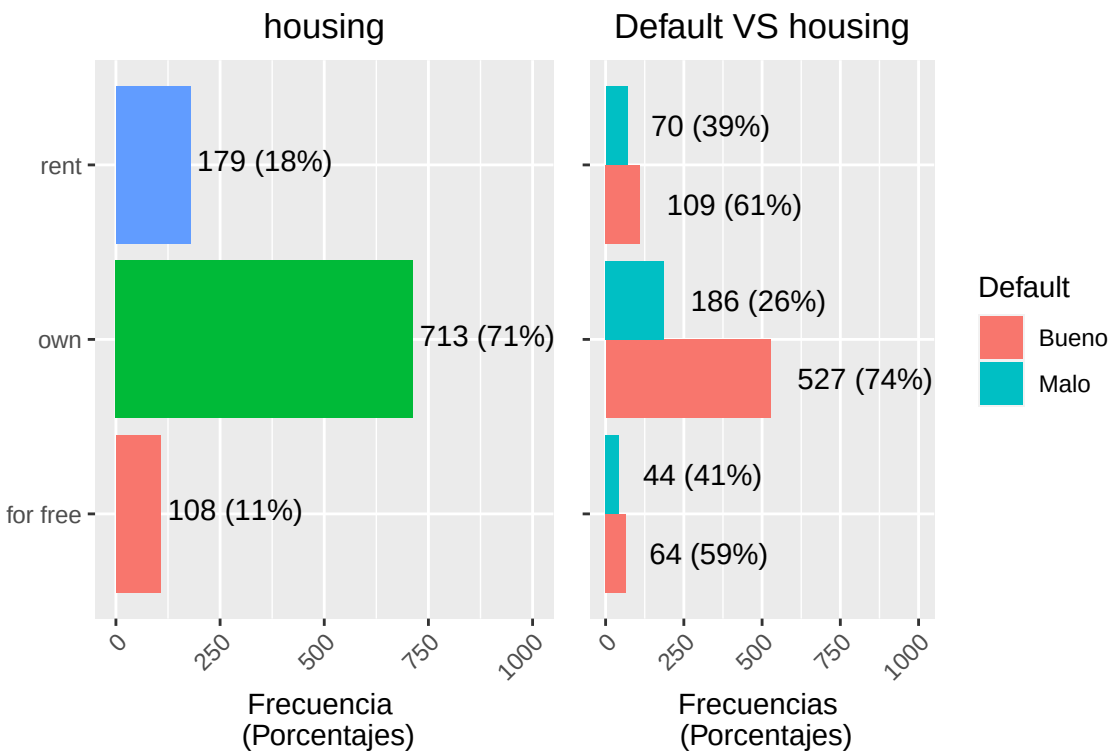
Vivienda: housing

Recoge el de tenencia de la vivienda habitual del cliente.

Factor de 3 niveles:

- Rent: En alquiler.
- Own: En propiedad.
- For Free: Gratuito.

```
crear_grafico(data, housing, "housing",
              "Default VS housing")
```



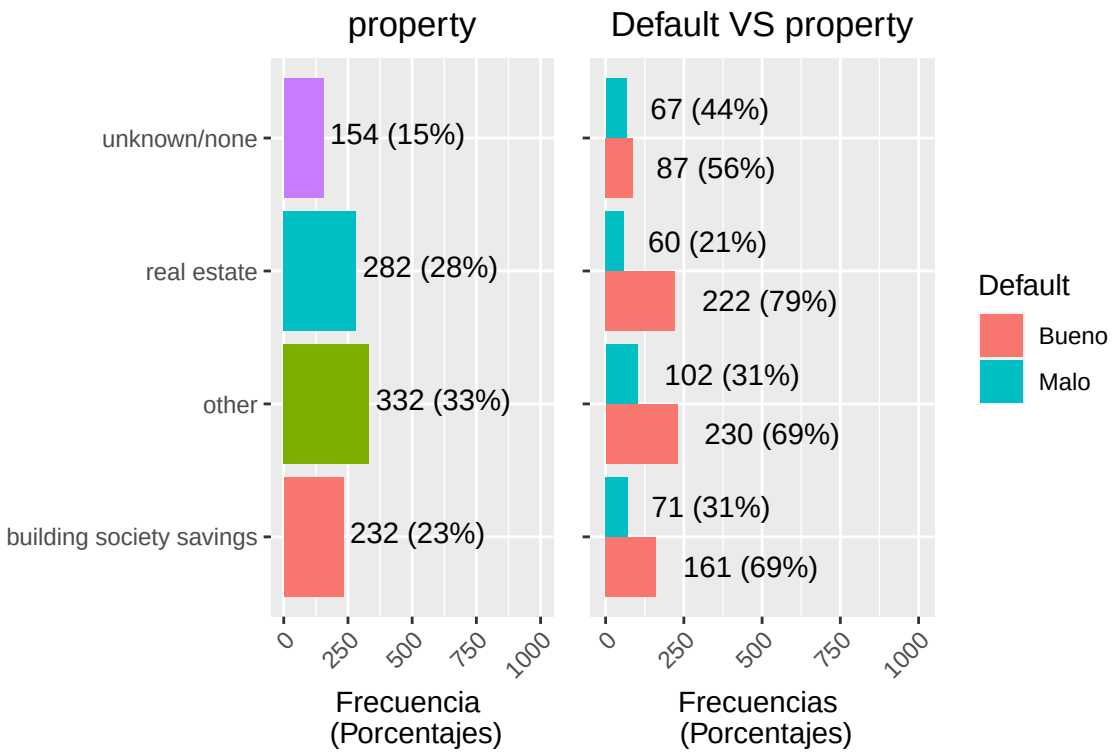
El 71% de los clientes tienen su vivienda en propiedad siendo el grupo mas solvente de todos, superando la proporcion de la variable levels con un 74% de creditos buenos.

En el caso de vivienda free y rent la proporcion de buenos prestamos se ubica en un 60%. Esta variable es relativamente simple ya que contiene pocos factores y delimita claramente niveles de solvencia distintos de los mostrados en la variable Levels.

Propiedad vivienda: Property.

Factor 4 niveles: • building society savings: adquirido con recursos propios de su negocio. 26 • real estate : Comprado con hipoteca. • unknow / none : Desconocido o sin vivienda en propiedad. • other : Resto de situaciones.

```
crear_grafico(data, property , "property",
              "Default VS property")
```

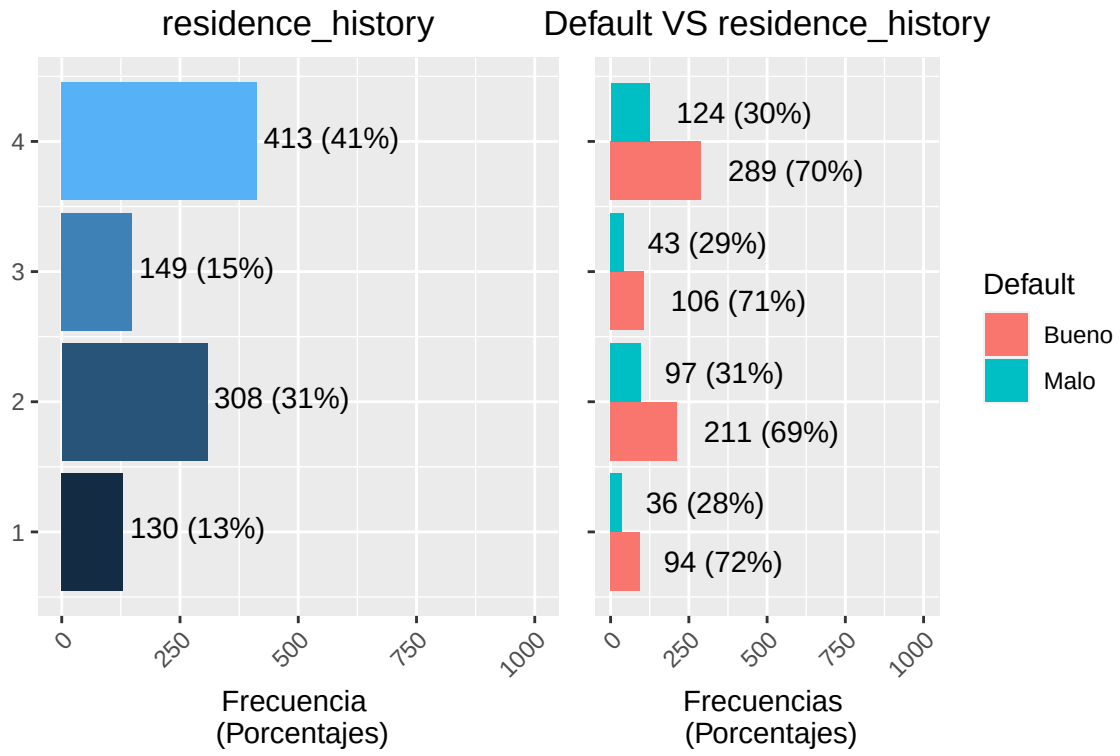


La mitad de las viviendas se han adquirida via ahorros o mediante hipoteca. La tipologia de mas solvencia es la de vivienda adquirida con hipoteca con un 80% de creditos buenos. La opcion desconocido hay que prestarle especial atencion ya que unicamente tiene un 56% de creditos buenos. El resto de opciones se encuentra en las proporcion de Levels con un 70% de creditos buenos.

Historico residencial: residence_history

Años en su residencia actual.

```
crear_grafico(data, residence_history , "residence_history",
              "Default VS residence_history")
```



El 56% de los creditos estan destinados a personas que llevan 3 o mas años viviendo en su actual residencia habitual.

La proporcion de creditos buenos en todos los segmentos respetan la proporcion de la variables Levels.

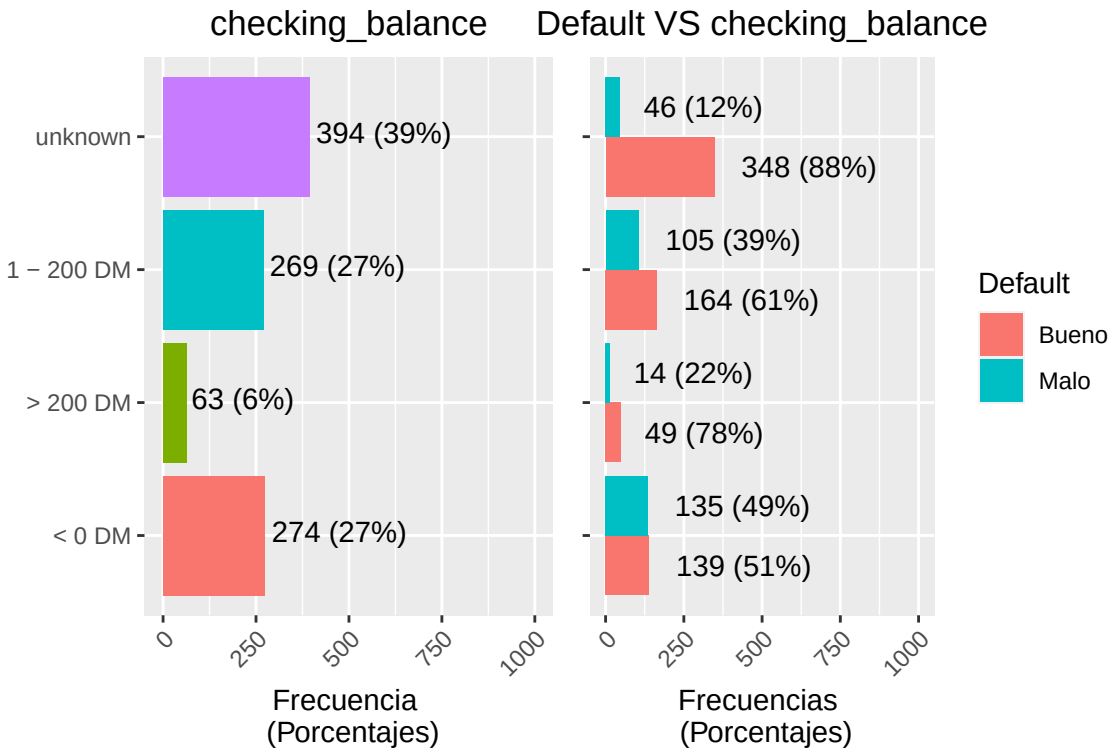
Perfil situacion financiera de los clientes.

Saldo en cuenta corriente: checking_balance.

Esta expresado en DM Deutsche Mark Character a transformar en factor. Tiene 4 niveles:

- < 0 DM - no tiene saldo positivo. Esta en descubierto.
- 0 y < 200 DM - Saldo positivo de hasta 200 DM.
- 200DM - Saldo positivo superior a 200 DM

```
crear_grafico(data, checking_balance , "checking_balance",
              "Default VS checking_balance")
```



Se desconocen los saldos de cuentas de clientes en un 39% de los prestamos. Si hay que mencionar, que ese categoria es la mas solvente con mucha diferencia con un 88% de creditos buenos.

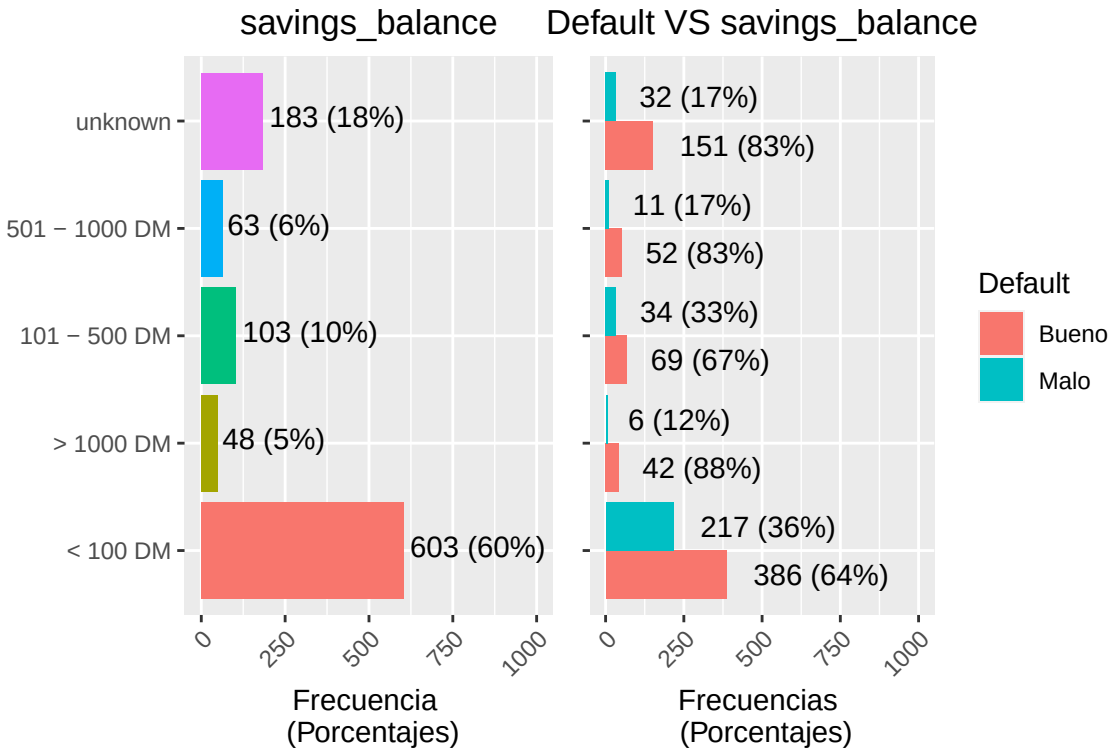
El 54% de los creditos tienen una cantidad en cuenta de hasta 200 DM y es importante destacar que en este caso la proporcion de creditos buenos baja del 60 %.

Nivel de ahorro: .

Esta expresado en DM Deutsche Mark Character a transformar en factor. Tiene 5 niveles:

- < 100 DM - El saldo es inferior a 100 DM.
- 101 - 500 DM - Saldo positivo de 101 hasta 500 DM.
- 501- 1000 - Saldo positivo de 500Dm a 1000 DM
- 1000 - Saldo positivo superior a 1000 DM.

```
crear_grafico(data, savings_balance , "savings_balance",
              "Default VS savings_balance")
```



El 60% de los clientes tiene un saldo de cuentas de hasta 100 marcos.

Existen dos niveles de solvencia:

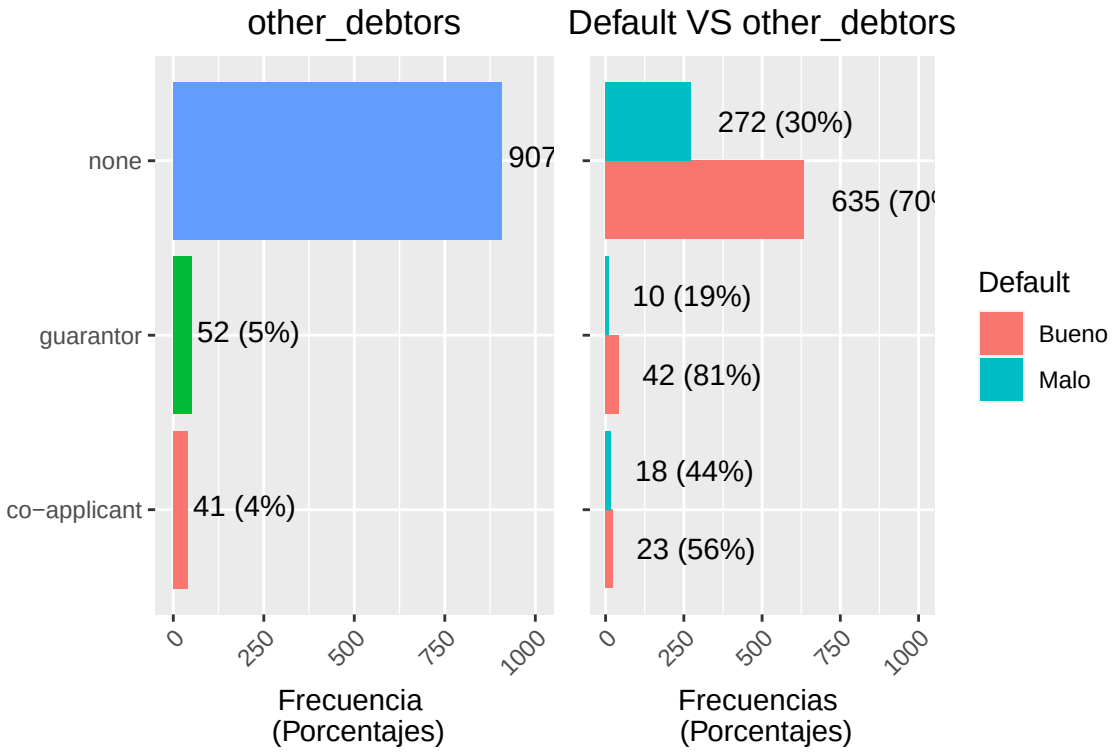
Nivel hasta 500 DM con un ratio de credito buenos alrededor del 60 % inferior al de la variable Levels.

Nivel superior a 500 DM donde el incremento de solvencia es muy notable alrededor del 80%

Otras deudas: other_debtors

Factor 3 niveles: • co-applicant: co-solicitante. • Guarantor: avalista. • None: ninguno.

```
crear_grafico(data, other_debtors , "other_debtors",
              "Default VS other_debtors")
```



El 90% de clientes no tienen otros deudas, lo que resulta en una proporcion de creditos buenos conforme a la variable Levels del 70%.

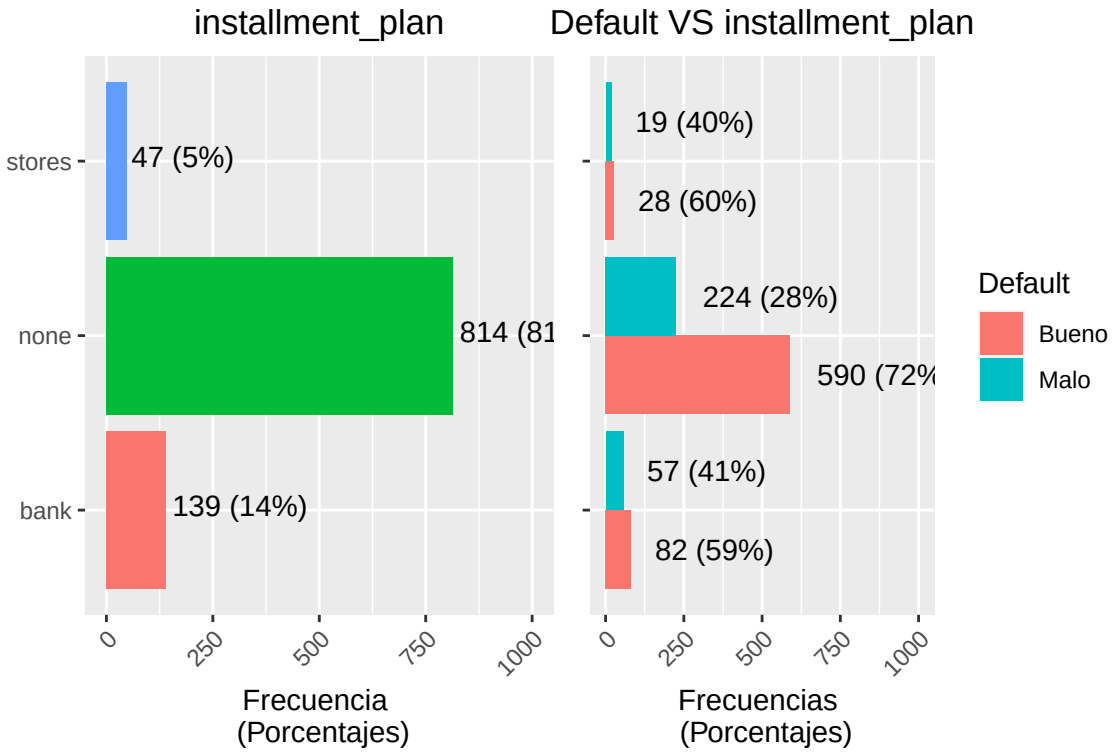
Entre el 10% restante la proporcion es muy dispar muy solventes en el caso guarantor un 81%, y en el co-applicant la solvencia baja hasta el 56%.

Bienes comprados a plazos: installmment_plan

Factor que consta de 3 niveles:

- None.
- Stores.
- Bank.

```
crear_grafico(data, installment_plan , "installment_plan",
              "Default VS installment_plan")
```



El 81% de los clientes no tienen planes de compra aplazados y estos siguen un ratio de creditos buenos del 70% en la linea de la variable Levels, un 70%.

En los otros dos grupos caen en un 60%.

```
# Return default values to original scale.
data$default <- ifelse(data$default == "Bueno","1","2")

# Move default column to last place
data <- data %>% relocate(default,.after=last_col())
credit <-data

# Custom columns created for EDA are eliminated.
credit <- subset(credit, select = -c(year_duration_loan, amount_levels,
                                     age_levels, purpose_levels))
str(credit)
```

```
## '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   : Ord.factor w/ 5 levels "0 - 1 yrs"<"1 - 4 yrs"<..: 4 2 3 3 2 2 4 2 3 5 ...
##  $ installment_rate    : int  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   : int  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                 : num  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    : int  2 1 1 1 2 1 1 1 1 2 ...
##  $ dependents          : int  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 ...
##  $ default             : chr  "1" "2" "1" "1" ...
```

Conclusiones del EDA

*** Tipologia de credito concedido.**

El banco oferta creditos de consumo se concentran un 81% en financiacion de coches(tanto nuevos como de segunda mano), como bienes para el hogar (muebles, Tv/radio).

Existe una proporcion de 70% de creditos buenos (son solventes) y un 30% malos (no solvente).

El importe en un 95% de los casos tienen un importe de hasta 5K DM.

El 77% tiene una duracion maxima de 2 años.

El apalancamiento es alto o muy alto en un 64% de los casos y el cliente presenta hasta dos creditos mas.

*** Perfil del cliente.**

Es un varon extranjero soltero, de entre 20-40 años, que tiene al menos una persona dependiente a su cargo. Es un trabajador por cuenta ajena en un 83% de los casos y que lleva en su actual puesto de trabajo un minimo de 4 años.

*** Perfil patrimonial del cliente.**

Un 71% de los clientes tiene vivienda en propiedad adquirida mediante ahorros y/o hipoteca.

El credito se solicita cuando el cliente lleva al menos 3 años residiendo en su actual vivienda.

*** Perfil situacion financiera del cliente.**

El 60% de los clientes tiene un saldo en cuenta de hasta 100DM.

Se desconoce el saldo de de tarjeta de credito para un 40% de los clientes, si bien hay que destacar que esa clase es la mas solvente con diferencia con un 88% de creditos buenos.

Para el resto de clientes, la mitad tiene un saldo en tarjeta de credito de hasta 200DM,y la proporcion de creditos buenos baja al 60%.

El 80% de los clientes no tienen otros planes de compra aplazados.

El 90% de los clientes no tiene deudas con otras entidades de las que no son clientes.

En general son creditos de baja cuantia, pero alto apalancamiento sobre ingresos, para un cliente que cuenta con un patrimonio consistente en su vivienda habitual, la cual continuan pagando, y una baja cantidad de efectivo en cuentas.

Modelos arboles de decision

Actuaciones previas.

Este es un problema de clasificacion de riesgo crediticio. Por tanto, la tarea consiste en hallar que variables minimizan el riesgo de impago o maximizar la probabilidad de cobrar. La variable default es la que medira que clientes eran buenos pagadores y cuales malos pagadores o posibles morosos. Niveles:

- 1 .- Credito bueno.
- 2 .- Credito malo.

En primer lugar se efectua una extraccion aleatoria, un sample, del dataset

```
#RNGversion("3.5.2")
set.seed(101)
train_sample <- sample(1000, 660)
str(train_sample)
```

```
##  int [1:660] 841 825 430 95 209 442 351 829 956 315 ...
```

```
credit_train <- credit[ train_sample, ]
credit_test  <- credit[-train_sample, ]
```

Comprobacion de que las proporciones se mantienen y la extraccion es correcta. Train:

```
round(prop.table(table(credit_train$default)),3)
```

```
##
##     1     2
## 0.708 0.292
```

Test:

```
round(prop.table(table(credit_test$default)),3)
```

```
##
##     1     2
## 0.685 0.315
```

Se mantienen las proporciones.

Modelo C 5.0

Dataset Completo parametros Default

```
# Modelo C5.0 completo default
credit_train$default <- factor(credit_train$default)

modelo_1 <- C5.0(credit_train[, names(credit_train) != "default"],
                 credit_train[, names(credit_train) == "default"], rules=TRUE)
summary(modelo_1 )
```

```
##
## Call:
## C5.0.default(x = credit_train[, names(credit_train) != "default"], y
##  = credit_train[, names(credit_train) == "default"], rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Fri May 17 01:44:00 2024
## -------------------------------
##
## Class specified by attribute 'outcome'
##
## Read 660 cases (21 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (16, lift 1.3)
##  checking_balance in {< 0 DM, 1 - 200 DM}
##  months_loan_duration <= 24
##  credit_history in {critical, delayed, repaid}
##  purpose = business
##  age <= 52
##  ->  class 1  [0.944]
##
## Rule 2: (108/9, lift 1.3)
##  months_loan_duration <= 11
##  credit_history in {critical, delayed, repaid}
##  amount <= 7882
##  ->  class 1  [0.909]
##
## Rule 3: (143/17, lift 1.2)
##  months_loan_duration <= 36
##  credit_history in {critical, delayed, repaid}
##  purpose = radio/tv
##  property in {building society savings, other, real estate}
##  ->  class 1  [0.876]
##
## Rule 4: (308/42, lift 1.2)
##  checking_balance in {> 200 DM, unknown}
##  ->  class 1  [0.861]
##
## Rule 5: (74/11, lift 1.2)
##  purpose in {car (used), retraining}
```

```
##  ->  class 1  [0.842]
##
## Rule 6: (300/47, lift 1.2)
##  months_loan_duration <= 22
##  credit_history in {critical, delayed, repaid}
##  amount <= 7882
##  installment_plan = none
##  ->  class 1  [0.841]
##
## Rule 7: (164/33, lift 1.1)
##  savings_balance in {501 - 1000 DM, unknown}
##  ->  class 1  [0.795]
##
## Rule 8: (8, lift 3.1)
##  checking_balance in {< 0 DM, 1 - 200 DM}
##  credit_history in {fully repaid, fully repaid this bank}
##  dependents > 1
##  ->  class 2  [0.900]
##
## Rule 9: (4, lift 2.8)
##  months_loan_duration > 11
##  purpose = furniture
##  employment_length = unemployed
##  ->  class 2  [0.833]
##
## Rule 10: (7/1, lift 2.7)
##  checking_balance = < 0 DM
##  purpose = education
##  amount <= 7882
##  ->  class 2  [0.778]
##
## Rule 11: (23/5, lift 2.6)
##  checking_balance in {< 0 DM, 1 - 200 DM}
##  credit_history in {critical, delayed, repaid}
##  amount > 7882
##  ->  class 2  [0.760]
##
## Rule 12: (352/201, lift 1.5)
##  checking_balance in {< 0 DM, 1 - 200 DM}
##  ->  class 2  [0.429]
##
## Default class: 1
##
##
## Evaluation on training data (660 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##      12  118(17.9%)   <<
##
##
##     (a)   (b)    <-classified as
```

```
##    ----  ----
##     436    31    (a): class 1
##      87   106    (b): class 2
##
##
##  Attribute usage:
##
##  100.00% checking_balance
##   62.27% credit_history
##   57.88% months_loan_duration
##   50.91% amount
##   45.45% installment_plan
##   36.97% purpose
##   24.85% savings_balance
##   21.67% property
##    2.42% age
##    1.21% dependents
##    0.61% employment_length
##
##
## Time: 0.0 secs
```

El algoritmo genera 12 reglas. Las primeras 7 reglas son de clase 1 creditos buenos, las 5 siguientes son la clase 2 credito malo y la probabilidad asociada.

Regla 1.- Probabilidad Credito Bueno 0.944 si :

- Balance en cuenta corriente entre descubierto a 200 DM
- Duracion menor o igual a 24 meses
- El historial de credito puede tener 3 estados : critical, delayed, repaid.
- El proposito del credito es business.
- La edad del cliente es de 52 años o menos.

Regla 2.- Probabilidad Credito Bueno 0.909 si:

- Duracion del prestamo menor o igual a 11 meses.
- El historial de credito puede tener 3 estados : critical, delayed, repaid.
- Cantidad del prestamo hasta 7882 euros
- La edad del cliente es de 52 años o menos.

Regla 3.- Probabilidad Credito Bueno 0.876 si:

- Duracion del prestamo menor o igual a 36 meses.
- El historial de credito puede tener 3 estados : critical, delayed, repaid.
- El proposito del credito es radio/tv.
- Su propiedad procede de ahorros de negocios, otros, inmuebles.

Regla 4.- Probabilidad Credito Bueno 0.861 si:

- El saldo en cuenta va de 200 a desconocido.

Regla 5 .- Probabilidad Credito Bueno 0.842 si:

- El proposito es comprar un coche usado y retraining.

Regla 6 .- Probabilidad Credito Bueno 0.84 si:

- Duracion del prestamo menor o igual a 22 meses.
- El historial de credito puede tener 3 estados : critical, delayed, repaid.
- antidad del prestamo hasta 7882 euros
- El cliente no tienen installment_plan

Regla 7 .- Probabilidad Credito Bueno 0.795 si:

- Saldo en cuenta de mas de 500 DM a desconocido.

Regla 8 .- Probabilidad Credito Malo 0.90 si:

- Saldo en cuenta de descubierto hasta 500 DM.
- El historial de credito puede tener 3 estados : fully repaid, fully repaid bank.
- Tiene mas de un dependiente a su cargo.

Regla 9 .- Probabilidad Credito Malo 0.833 si:

- Duracion del prestamo mayor a 11 meses.
- Proposito del prestamo compra de muebles.
- El cliente esta desempleado.

Regla 10.- Probabilidad Credito Malo 0.7882 si:

- El cliente tiene descubierto en cuenta.
- El credito es para educacion.
- Cuantia igual o inferior a 7882

Regla 11.- Probabilidad Credito Malo 0.76 si:

- Saldo en cuenta de descubierto hasta 500 DM.
- El historial de credito puede tener 3 estados : critical, delayed, repaid.
- Cantidad del prestamo mayor de 7882 euros.

Regla 12.- Probabilidad Credito Malo 0.429 si:

- Saldo en cuenta de descubierto hasta 500 DM.
- El arbol que genera es complejo y dificil de graficar. Es posible que empleando menos niveles en algunas variables permitiria obtener un arbol mas simple.

```
# Representacion grafica del modelo

# modelo_1 <- C5.0(credit_train[, names(credit_train) != "default"],
# credit_train[, names(credit_train) == "default"])
# plot(modelo_1)
```

El arbol que se obtiene empleando todas las variables del dataset es muy grande, con muchas condiciones, lo que dificulta su lectura, compresion y obtencion de conocimiento.

Se obtienen 118 observaciones mal clasificadas lo que es un 18%.

Dataset Simplificado parametros Default

Se selecciona las variables que superan el 50% de attribute use del modelo anterior a fin de reducir la complejidad del arbol.

```
# Modelo C5.0 simplificado default
modelo_2 <- C5.0(credit_train[, names(credit_train) %in% c(
  "checking_balance","credit_history", "months_loan_duration", "amount",
  "installment_plan")],
  credit_train[, names(credit_train) == "default"],rules = TRUE)

summary(modelo_2)
```

```
##
## Call:
## C5.0.default(x = credit_train[, names(credit_train) %in%
##  "amount", "installment_plan")], y = credit_train[, names(credit_train)
##  == "default"], rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Fri May 17 01:44:00 2024
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 660 cases (6 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (308/42, lift 1.2)
##  checking_balance in {> 200 DM, unknown}
##  ->  class 1  [0.861]
##
## Rule 2: (590/155, lift 1.0)
##  amount <= 7882
##  installment_plan in {bank, none}
##  ->  class 1  [0.736]
##
## Rule 3: (40/9, lift 2.6)
##  checking_balance in {< 0 DM, 1 - 200 DM}
##  credit_history in {fully repaid, fully repaid this bank}
##  amount <= 7882
##  ->  class 2  [0.762]
##
## Rule 4: (6/1, lift 2.6)
##  checking_balance = 1 - 200 DM
##  installment_plan = stores
##  ->  class 2  [0.750]
##
## Rule 5: (31/9, lift 2.4)
##  checking_balance in {< 0 DM, 1 - 200 DM}
##  amount > 7882
##  ->  class 2  [0.697]
##
```

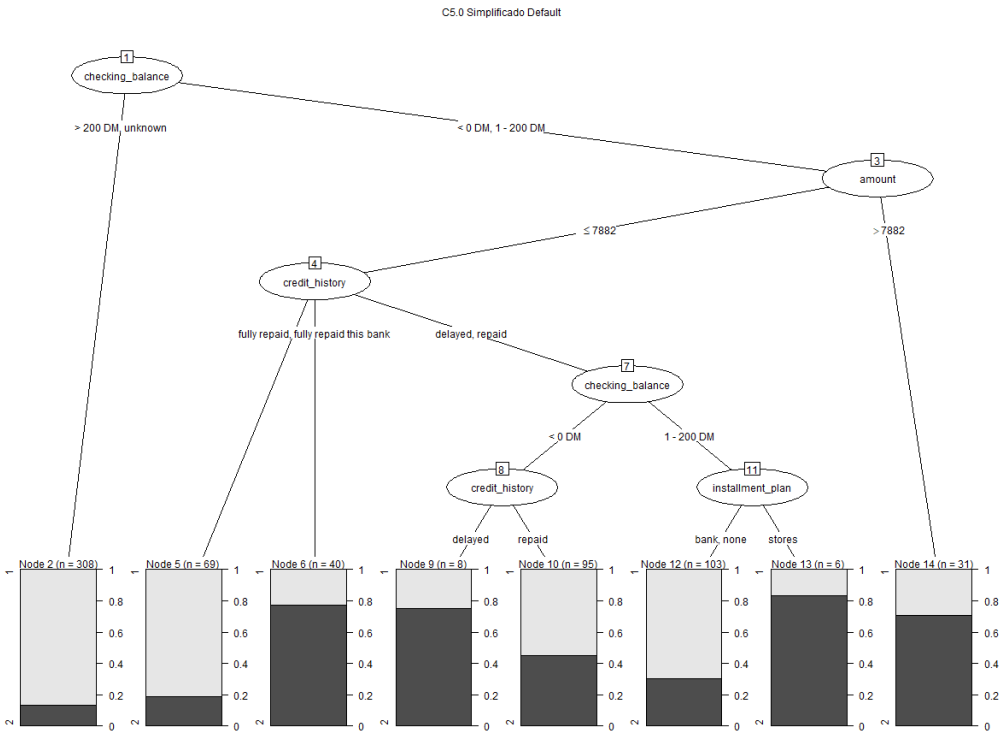
```
## Default class: 1
##
##
## Evaluation on training data (660 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##       5  154(23.3%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     448    19    (a): class 1
##     135    58    (b): class 2
##
##
##  Attribute usage:
##
##   94.70% amount
##   90.30% installment_plan
##   58.33% checking_balance
##    6.06% credit_history
##
##
## Time: 0.0 secs
```

```
modelo_2 <- C5.0(credit_train[, names(credit_train) %in% c(
  "checking_balance","credit_history", "months_loan_duration", "amount",
  "installment_plan")],
  credit_train[, names(credit_train) == "default"])

# Generar el gráfico y guardarlo como un archivo PNG
png("C50_Simplificado_Default.png", width = 1200, height = 900)
# Ajusta el tamaño según necesites
plot(modelo_2, main = "C5.0 Simplificado Default")
dev.off()  # Terminar la captura del gráfico
```

```
## pdf
##   2
```

```
img <- readPNG("C50_Simplificado_Default.png")
grid::grid.raster(img)
```



El algoritmo genera 5 reglas.

Las primeras 2 reglas son de clase: 1 creditos buenos, las 3 siguientes son la clase 2 credito malo y la probabilidad asociada.

Regla 1.- Probabilidad Credito Bueno 0.861 si :

- Balance en cuenta corriente mayor o igual 200 DM a desconocido.

Regla 2.- Probabilidad Credito Bueno 0.736 si :

- Cantidad del prestamo menor o igual a 7882.
- El installment plan puede ser en bank o inexistente.

Regla 3.- Probabilidad Credito Malo 0.762 si :

- El saldo bancario oscila entre descubierto y 200DM.
- Cantidad del prestamo menor o igual a 7882.
- Historico de credito con valor: fully repaid, fully repaid this bank.

Regla 4.- Probabilidad Credito Malo 0.75 si :

- El saldo bancario oscila entre 0 y 200DM.
- El installment plan es stores.

Regla 5.- Probabilidad Credito Malo 0.697 si :

- El saldo bancario oscila entre descubierto y 200DM.
- Cantidad del prestamo mayor a 7882.
- Puede obtenerse conocimiento de estas reglas de modo visual muy facilmente con la representacion del grafico:

Nivel 1 .- Variable Checking Balance

- = 200 DM a unknow: Nodo terminal con 308 casos

```
--> Nodo 2

--> Regla 1 (Prob 0.86)
```

- < 0 - 200 DM - Variable Amount

```
--> Nivel 2  == Nodo 3
```

Nivel 2 .- Checking Balance [< 0 - 200 DM] - Checking Balance [< 0 - 200 DM]

- Amount > 7882 Nodo terminal con 14 casos

```
--> Nodo 14

--> Regla 5 (Prob 0.697)
```

- Amount <= 7882

Nivel 3 .- Checking Balance [< 0 - 200 DM] - Checking Balance [< 0 - 200 DM] - Variable Amount [<=7882] – Credit History

- fully

```
--> Nodo Terminal 5

--> Regla 3 (Prob 0.762)
```

- fully repaid this bank.

```
--> Nodo Terminal 6

--> Regla 3 (Prob 0.762)
```

Nivel 4 .- Checking Balance [< 0 - 200 DM]- Checking Balance [< 0 - 200 DM] - Variable Amount [<=7882] – Credit History [delayed repaid]

- Checking Balance < 0 DM
- Checking Balance 1 - 200 DM

Nivel 5.1 .- Checking Balance [< 0 - 200 DM]- Checking Balance [< 0 - 200 DM] - Variable Amount [<=7882] – Credit History [delayed repaid] - - Checking Balance [< 0 DM]

- Credit History delayed repaid

```
--> Nodo Terminal 6

--> Regla 3 (Prob 0.762)
```

Nivel 5.2 .- Checking Balance [< 0 - 200 DM] - Checking Balance [< 0 - 200 DM] - Variable Amount [<=7882] - Credit History [delayed repaid] - - Checking Balance [1- 200 DM]

- Installment_plan

```
--> bank, none
--> Nodo Terminal 12
--> Regla 2 (Prob 0.736)

--> stores
--> Nodo Terminal 12
--> Regla 2 (Prob 0.736)
```

Conclusiones: Con esta solucion se obtiene 154 instancias mal clasificadas lo que supone 36 instancias mas que con el dataset completo y un error del 23.3% frente al 17.9%.

El numero de reglas se reduce de 12 a 5.

En este caso: 2 reglas clase 1 credito bueno: 3 reglas clase 2 credito malo.

Las reglas tienen un maximo de 3 condiciones, en el modelo simplificado frente a las 5 del modelo completo.

Esto es, se obtiene menos de la mitad de las reglas con el dataset simplificado y no solo eso,la complejidad de la mismas se reduce tambien a la mitad.

El arbol por tanto puede graficarse, es mucho mas simple de obtener conocimiento a partir de las reglas y unicamente hay que evaluar la perdida de informacion para decidir.

Analisis de la Bondad de Ajuste

```
# Modelo 1: C5.0 Completo Default

modelo_1_predict <- predict(modelo_1,
                            credit_test[, names( credit_test) != "default"],
                            type="class")

confusion_matrix_modelo_1 <- table(credit_test$default, modelo_1_predict)
modelo <- "C5.0 Completo Default"
performance_modelo1 <- calculate_performance(modelo, confusion_matrix_modelo_1)

# Modelo 2: C5.0 Simplificado Default

modelo_2_predict <- predict(modelo_2,
                            credit_test[, names( credit_test) != "default"],
                            type="class")

confusion_matrix_modelo2 <- table(credit_test$default, modelo_2_predict)
modelo <- "C5.0 Simplificado Default"
performance_modelo2 <- calculate_performance(modelo, confusion_matrix_modelo2 )
```

```r
# Unimos resultados:
Results_C50 <-rbind(performance_modelo1, performance_modelo2)
print(Results_C50)
```

```
##           model_description  TP FP FN TN accuracy error
## 1     C5.0 Completo Default 209 24 61 46    0.750 0.250
## 2 C5.0 Simplificado Default 219 14 84 23    0.712 0.288
```

En el modelo simplificado implica una reduccion de instancias correctas de un 0.75 a un 0.71 y el error se incrementa de un 0.25 a un 0.28. Pese a ello es mas interesante el modelo simplificado ya que facilita la comprension del arbol y sus reglas.

Modelo C5.0 Boosting

Debido a que se va a fijar un parametro para el boosting trials = 10, no se analizaran las reglas de los arboles, ya que alargarian innecesariamente el documento, por esta razon, se procedera a calcular los modelos y analizar la bondad de ajuste de los modelos generados.

Modelo C5.0 Dataset Completo Adaptative Boosting

```
# C5.0 Boosting <= Dataset Completo
modelo_3 <- C5.0(credit_train[, names(credit_train) != "default"],
                 credit_train[, names(credit_train) == "default"], trials=10)

# # Generar el gráfico y guardarlo como un archivo PNG
# png("C5.0 Completo Boosting.png", width = 1200, height = 900)
# Ajusta el tamaño según necesites
# plot(modelo_3, main = "C5.0 Completo Boosting")
# dev.off()
# # Terminar la captura del gráfico
# img <- readPNG("C5.0 Completo Boosting.png")
# grid::grid.raster(img)
```

```
# C5.0 Boosting <= Dataset Completo
modelo_3_predict <- predict(modelo_3, credit_test[
  , names(credit_test) != "default"], type="class")

table(credit_test[, names(credit_test) == "default", ],
      Predicted=modelo_3_predict)
```

```
##    Predicted
##        1   2
##   1 209  24
##   2  58  49
```

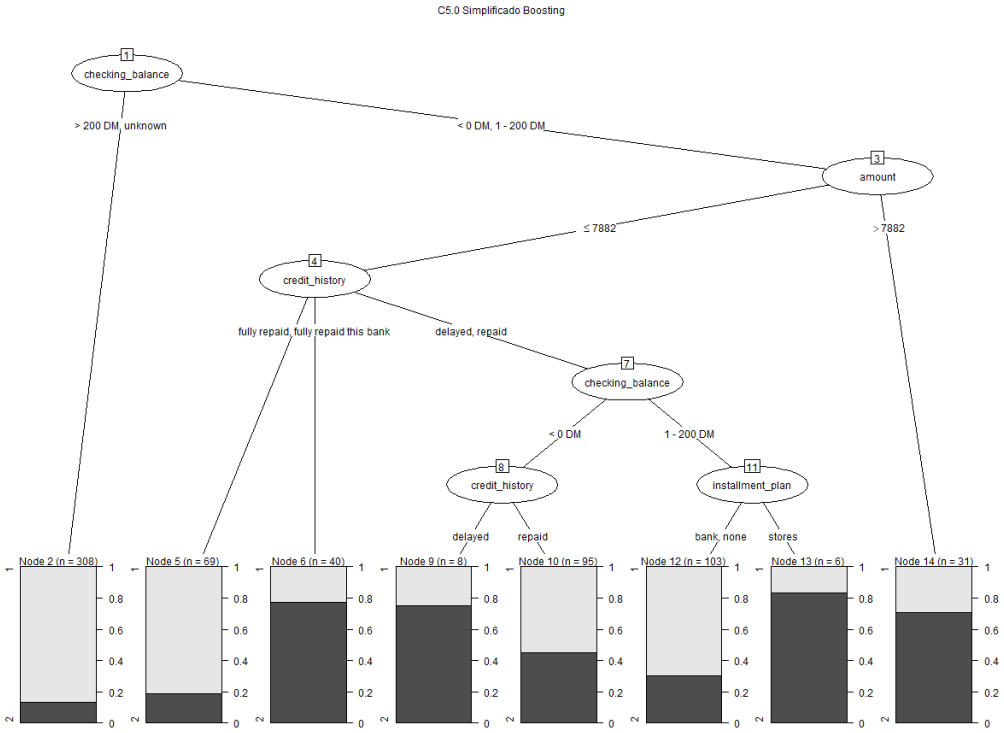
Modelo C5.0 Dataset Simplificado Adaptative Boosting

```
# C5.0 - Boosting <= Dataset Simplificado [arbol]
modelo_4 <- C5.0(credit_train[, names(credit_train) %in% c(
  "checking_balance","credit_history", "months_loan_duration", "amount",
  "installment_plan")], credit_train[, names(credit_train) == "default"],
  trials=10)
#plot(modelo_4)

# Generar el gráfico y guardarlo como un archivo PNG
png("C50_Simplificado_Boosting.png", width = 1200, height = 900)
# Ajusta el tamaño según necesites
plot(modelo_4, main = "C5.0 Simplificado Boosting")
dev.off()  # Terminar la captura del gráfico
```

```
## pdf
##   2
```

```
img <- readPNG("C50_Simplificado_Boosting.png")
grid::grid.raster(img)
```




```
# C5.0 - Boosting <= Dataset Simplificado  ## PREDICCIONES ##
modelo_4_predict <- predict(modelo_4,
                            credit_test[, names(credit_test) != "default"],
                            type="class")
table(credit_test[, names(credit_test) == "default",],Predicted=modelo_4_predict)
```

```
##     Predicted
##       1   2
##   1 211  22
##   2  65  42
```

```
# C5.0 Boosting Completo
confusion_matrix_modelo_3 <- table(credit_test$default, modelo_3_predict)
modelo <- "C5.0 Completo Boosting"
performance_modelo3 <- calculate_performance(modelo, confusion_matrix_modelo_3)

# C5.0 Boosting Simplificado
confusion_matrix_modelo_4 <- table(credit_test$default, modelo_4_predict)
modelo <- "C5.0 Simplificado Boosting"
performance_modelo4 <- calculate_performance(modelo, confusion_matrix_modelo_4)

# Resultados Boosting completo
Results_C50Bost <-rbind(performance_modelo3, performance_modelo4)
print(Results_C50Bost)
```

```
##            model_description  TP FP FN TN accuracy error
## 1     C5.0 Completo Boosting 209 24 58 49    0.759 0.241
## 2 C5.0 Simplificado Boosting 211 22 65 42    0.744 0.256
```

La mejora obtenida con el boosting es inapreciable en el caso del dataset completo.

Sin embargo en el caso del dataset simplificado con boosting la mejora es muy notable ya que nos acerca a los resultados del modelo completo sin boosting.

Por esta razon de todos los modelos presentados se escogeria el Simplificado con Boosting, es el que mejor equilibrio ofrece en cuanto a nivel de acierto y error , unido a una menor dificultad y capacidad explicativa del arbol obtenido.

```
# Resultados Boosting completo
Results_C50_Total <-rbind(Results_C50, Results_C50Bost)
print(Results_C50_Total)
```

```
##            model_description  TP FP FN TN accuracy error
## 1      C5.0 Completo Default 209 24 61 46    0.750 0.250
## 2  C5.0 Simplificado Default 219 14 84 23    0.712 0.288
## 3     C5.0 Completo Boosting 209 24 58 49    0.759 0.241
## 4 C5.0 Simplificado Boosting 211 22 65 42    0.744 0.256
```

Modelo C5.0 con Penalizacion

Se establecera una penalizacion en los FN, se pretende evitar al maximo este caso, aquellos creditos asignados buenos que en realidad no lo son.

Se le aplica una penalizacion diferente a los FP con valor 1 minima y de valor 4 maxima a los FN, con este criterio se procede a recalcular todos los modelos.

```
# Generamos matriz dimensiones
matrix_dimensions <- list(c("1", "2"), c("1", "2"))
names(matrix_dimensions) <- c("predicted", "actual")
matrix_dimensions
```

```
## $predicted
## [1] "1" "2"
##
## $actual
## [1] "1" "2"
```

```
# Generamos matriz de errores

error_cost <- matrix(c(0, 1, 4, 0), nrow = 2,
dimnames = matrix_dimensions)
error_cost
```

```
##          actual
## predicted 1 2
##         1 0 4
##         2 1 0
```

```r
# Modelo 5: C5.0 Completo Default Penalizacion

modelo_5 <- C5.0(credit_train[, names(credit_train) != "default"],
                 credit_train[, names(credit_train) == "default"], rules=TRUE,
                 costs = error_cost)

modelo_5_predict <- predict(modelo_5,
                            credit_test[, names( credit_test) != "default"],
                            type="class")

confusion_matrix_modelo_5 <- table(credit_test$default, modelo_5_predict)
modelo <- "C5.0 Completo Default Penalizacion"
performance_modelo5 <- calculate_performance(modelo, confusion_matrix_modelo_5)

# Modelo 6: C5.0 Simplificado Default Penalizacion

modelo_6 <- C5.0(credit_train[, names(credit_train) %in% c(
  "checking_balance","credit_history", "months_loan_duration", "amount",
  "installment_plan")],
  credit_train[, names(credit_train) == "default"],rules = TRUE,
  costs = error_cost)

modelo_6_predict <- predict(modelo_6,
                            credit_test[, names( credit_test) != "default"],
                            type="class")

confusion_matrix_modelo6 <- table(credit_test$default, modelo_6_predict)
modelo <- "C5.0 Simplificado Default Penalizacion"
performance_modelo6 <- calculate_performance(modelo, confusion_matrix_modelo6 )

# Unimos resultados:
Results_C50_Pen <-rbind(performance_modelo5, performance_modelo6)

# C5.0 Boosting Completo Penalizacion
modelo_7 <- C5.0(credit_train[, names(credit_train) != "default"],
                 credit_train[, names(credit_train) == "default"], trials=10,
                 costs = error_cost)

modelo_7_predict <- predict(modelo_7, credit_test[
  , names(credit_test) != "default"], type="class")

confusion_matrix_modelo_7 <- table(credit_test$default, modelo_7_predict)
modelo <- "C5.0 Completo Boosting Penalizacion"
performance_modelo7 <- calculate_performance(modelo, confusion_matrix_modelo_7)

# C5.0 - Boosting Simplificado Penalizacion
modelo_8 <- C5.0(credit_train[, names(credit_train) %in% c(
  "checking_balance","credit_history", "months_loan_duration", "amount",
  "installment_plan")], credit_train[, names(credit_train) == "default"],
  trials=10,costs = error_cost)

modelo_8_predict <- predict(modelo_8,
                            credit_test[, names(credit_test) != "default"],
```

```
                          type="class")

confusion_matrix_modelo_8 <- table(credit_test$default, modelo_8_predict)
modelo <- "C5.0 Simplificado Boosting Penalizacion"
performance_modelo8 <- calculate_performance(modelo, confusion_matrix_modelo_8)

# Resultados Boosting completo
Results_C50Bost_Pen <-rbind(performance_modelo7, performance_modelo8)

Result_CS50_Total_Pen <- rbind(Results_C50_Pen,Results_C50Bost_Pen)

print(Result_CS50_Total_Pen)
```

```
##                         model_description  TP  FP FN TN accuracy error
## 1      C5.0 Completo Default Penalizacion  98 135 14 93    0.562 0.438
## 2  C5.0 Simplificado Default Penalizacion 111 122 18 89    0.588 0.412
## 3     C5.0 Completo Boosting Penalizacion 222  11 74 33    0.750 0.250
## 4 C5.0 Simplificado Boosting Penalizacion 206  27 75 32    0.700 0.300
```

La aplicacion de la penalizacion al modelo completo por default resulta en una bajada muy considerable de los FN a costa de disminuir la accuracy y aumentar el error como era de esperar. El problema es que los TP disminuyen a menos de la mitad con lo que la concesion de creditos se veria muy mermada y por otra parte los FP se multiplican por 5. La reduccion de FN implicaria un cambio muy grande en el modelo de negocio.

La aplicacion de la penalizacion en los casos de boosting, produce unos resultados mas cercanos a los modelos previos a la aplicacion de la penalizacion por el propio efecto promedio que genera el boosting. Aumentan los TP sensiblemente pero a costa de emperorar los TN que era justo lo que se pretendia evitar.

```
Modelo_CS50_Completo <-rbind(Results_C50_Total,Result_CS50_Total_Pen)
Modelo_CS50_Completo
```

```
##                         model_description  TP  FP FN TN accuracy error
## 1                    C5.0 Completo Default 209  24 61 46    0.750 0.250
## 2                C5.0 Simplificado Default 219  14 84 23    0.712 0.288
## 3                   C5.0 Completo Boosting 209  24 58 49    0.759 0.241
## 4               C5.0 Simplificado Boosting 211  22 65 42    0.744 0.256
## 5      C5.0 Completo Default Penalizacion  98 135 14 93    0.562 0.438
## 6  C5.0 Simplificado Default Penalizacion 111 122 18 89    0.588 0.412
## 7     C5.0 Completo Boosting Penalizacion 222  11 74 33    0.750 0.250
## 8 C5.0 Simplificado Boosting Penalizacion 206  27 75 32    0.700 0.300
```

En esta ultima tabla se muestra todos los modelos calculados para este algoritmo.

Se seleccionaria el modelo C5.0 Simplificado Bosting ya que minimiza el error con un 0.256, minimiza los FN, maximiza los TP y ademas proporciona el arbol mas simple.

Modelo CART

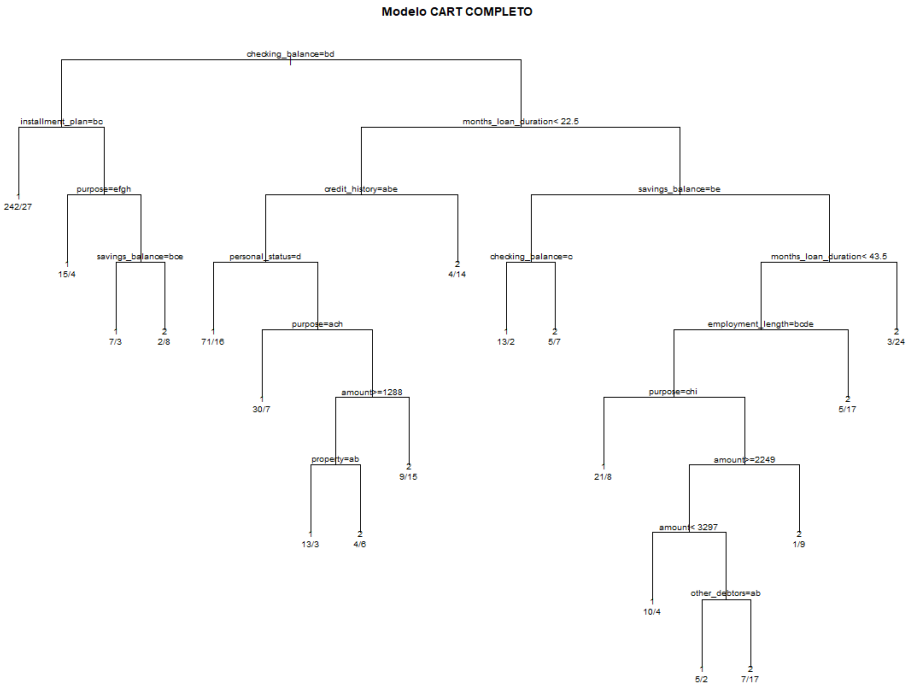
Cart Completo

```
modelo_cart  <- rpart(default~., method="class", data=credit_train)

# Generar el gráfico y guardarlo como un archivo PNG
png("Modelo_CART_Completo.png", width = 1200, height = 900)
# Ajusta el tamaño según necesites
plot(modelo_cart, uniform=TRUE,main = "Modelo CART COMPLETO")
text(modelo_cart , use.n=TRUE, cex=.8)
dev.off()  # Terminar la captura del gráfico
```

```
## pdf
##   2
```

```
img <- readPNG("Modelo_CART_Completo.png")
grid::grid.raster(img)
```




```
modelo_cart_predict <- modelo_cart %>% predict(credit_test, type = "class")

confusion_matrix_modelo_cart <- table(credit_test$default, modelo_cart_predict)
modelo <-"CART model"

performance_modelo_cart <- calculate_performance(modelo,
                                                 confusion_matrix_modelo_cart)
```

```
performance_modelo_cart
```

```
##   model_description  TP FP FN TN accuracy error
## 1        CART model 199 34 58 49    0.729 0.271
```

El modelo obtenido esta en la linea de los obtenidos con el modelo C50 accurancy entorno al 0.73 y un error cercano al 0.27.

Este modelo genera un arbol binario y esta empleando todas las variables del modelo.

Cart Pruned

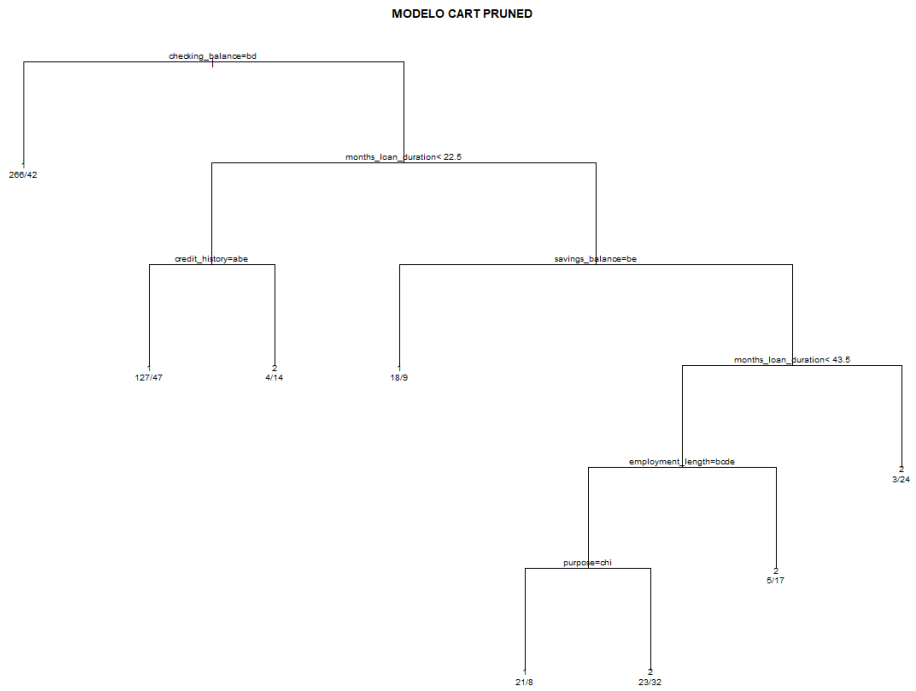
```
# Modelo CART Pruned->
modelo_cart_pruned <- prune(modelo_cart,
                            cp=modelo_cart$cptable
                            [which.min(modelo_cart$cptable[,"xerror"]),"CP"])

# Generar el gráfico y guardarlo como un archivo PNG
png("Modelo_CART_Pruned.png", width = 1200, height = 900)

# Ajusta el tamaño según necesites
plot(modelo_cart_pruned, uniform=TRUE,main = "MODELO CART PRUNED")
text(modelo_cart_pruned , use.n=TRUE, cex=.8)
dev.off()  # Terminar la captura del gráfico
```

```
## pdf
##   2
```

```
img <- readPNG("Modelo_CART_Pruned.png")
grid::grid.raster(img)
```




```
modelo_cart_pruned_predict <- modelo_cart_pruned %>% predict(credit_test,
                                                             type = "class")

confusion_matrix_modelo_cart_pruned <- table(credit_test$default,
                                             modelo_cart_pruned_predict)
modelo <-"CART model pruned"

performance_modelo_cart_pruned <- calculate_performance(modelo,
                                  confusion_matrix_modelo_cart_pruned)

performance_modelo_cart_pruned
```

```
##   model_description  TP FP FN TN accuracy error
## 1 CART model pruned 214 19 65 42    0.753 0.247
```

Con el modelo podado el numero de FN aumenta a pesar de que accuracy y error se mantienen a la par que el modelo sin poder, el incremento es minimo, en cambio los nodos del arbol pruned contienen condiciones mas sencillas, binarias.

Por ultimo se procede obtener el arbol con el modelo simplificado.

Cart Completo Simplificado

```
# Seleccionar solo las columnas necesarias y la variable objetivo
credit_train_subset <- credit_train[, c(
  "checking_balance", "credit_history", "months_loan_duration", "amount",
  "installment_plan", "default")]
```

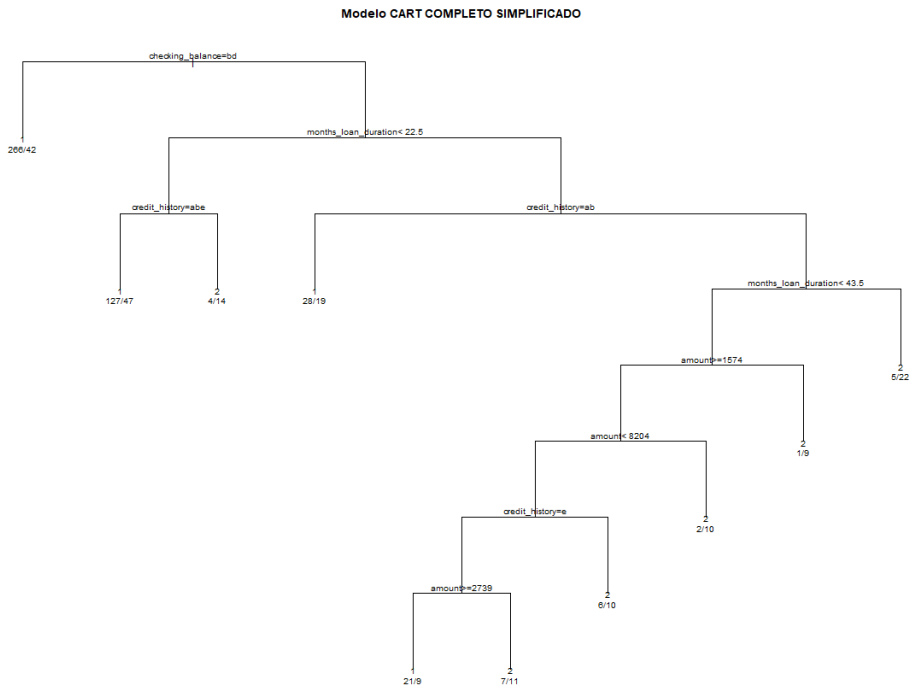
```r
# Filtrar filas con valores completos (sin NA) en las columnas seleccionadas
credit_train_clean <- credit_train_subset[complete.cases(credit_train_subset), ]

# Ajustar el modelo CART con las variables seleccionadas
modelo_cart_simp <- rpart(default ~ checking_balance + credit_history +
                          months_loan_duration + amount + installment_plan,
                          method = "class", data = credit_train_clean)

# Generar el gráfico y guardarlo como un archivo PNG
png("Modelo_CART_Completo_Simplificado.png", width = 1200, height = 900)
plot(modelo_cart_simp, uniform = TRUE, main = "Modelo CART COMPLETO SIMPLIFICADO")
text(modelo_cart_simp, use.n = TRUE, cex = .8)
dev.off()  # Terminar la captura del gráfico
```

```
## pdf
##   2
```

```r
# Leer y mostrar la imagen
library(png)
library(grid)
img <- readPNG("Modelo_CART_Completo_Simplificado.png")
grid::grid.raster(img)
```




```r
modelo_cart_predict_simp <- modelo_cart_simp %>% predict(credit_test, type = "class")

confusion_matrix_modelo_cart_simp <- table(credit_test$default, modelo_cart_predict_simp)
modelo <-"CART model simplificado"
```

```
performance_modelo_cart_simp <- calculate_performance(modelo,
                                                      confusion_matrix_modelo_cart_simp)

performance_modelo_cart_simp
```

```
##         model_description  TP FP FN TN accuracy error
## 1 CART model simplificado 219 14 79 28    0.726 0.274
```

Cart Completo Simplificado Pruned

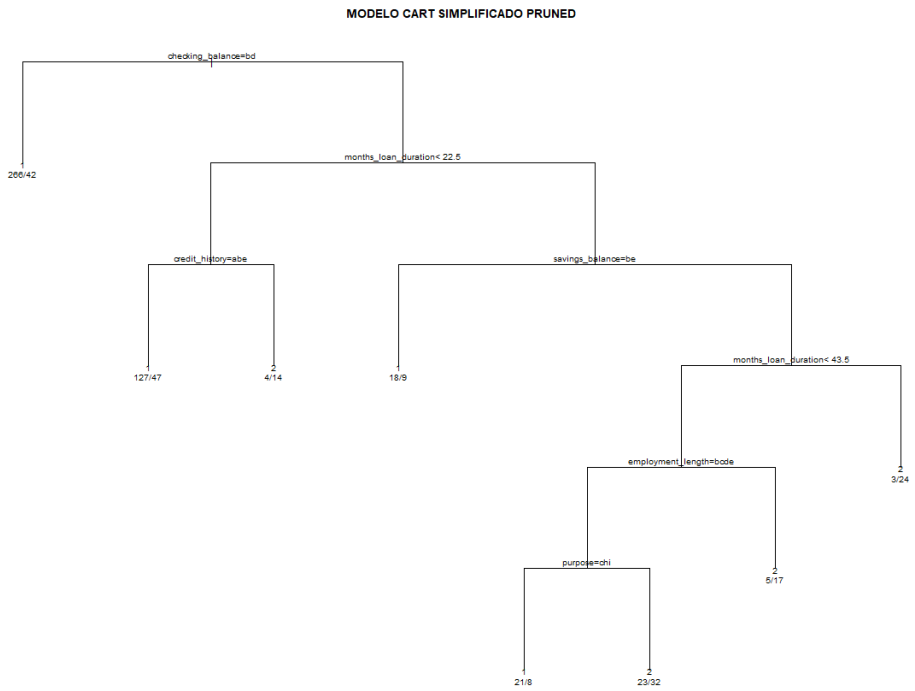
```
# Modelo CART Pruned->
modelo_cart_simp_pruned <- prune(modelo_cart_simp,
                           cp=modelo_cart$cptable
                           [which.min(modelo_cart$cptable[,"xerror"]),"CP"])

# Generar el gráfico y guardarlo como un archivo PNG
png("Modelo_CART_Simp_Pruned.png", width = 1200, height = 900)

# Ajusta el tamaño según necesites
plot(modelo_cart_pruned, uniform=TRUE,main = "MODELO CART SIMPLIFICADO PRUNED")
text(modelo_cart_pruned , use.n=TRUE, cex=.8)
dev.off()  # Terminar la captura del gráfico
```

```
## pdf
##   2
```

```
img <- readPNG("Modelo_CART_Simp_Pruned.png")
grid::grid.raster(img)
```




```
modelo_cart_predict_simp_pruned <- modelo_cart_simp_pruned %>% predict(credit_test, type = "class")

confusion_matrix_modelo_cart_simp_pruned <- table(credit_test$default, modelo_cart_predict_simp_pruned)
modelo <-"CART model simplificado pruned"

performance_modelo_cart_simp_pruned <- calculate_performance(modelo,
                                                             confusion_matrix_modelo_cart_simp_pruned)

performance_modelo_cart_simp_pruned
```

```
##                model_description  TP FP FN TN accuracy error
## 1 CART model simplificado pruned 207 26 71 36    0.715 0.285
```

Resumen de todos los modelos CART planteados.

```
Modelo_Cart_Total <- rbind(performance_modelo_cart,
                           performance_modelo_cart_pruned,
                           performance_modelo_cart_simp,
                           performance_modelo_cart_simp_pruned
                           )

Modelo_Cart_Total
```

```
##                model_description  TP FP FN TN accuracy error
## 1                      CART model 199 34 58 49    0.729 0.271
## 2               CART model pruned 214 19 65 42    0.753 0.247
## 3        CART model simplificado 219 14 79 28    0.726 0.274
## 4 CART model simplificado pruned 207 26 71 36    0.715 0.285
```

Todos los resultados estan en un rango proximo, mucho mas de lo que se obtuvo para el algoritmo C5.0.

Hay que recordar que el algoritmo a superar de C50 es:

```
performance_modelo4
```

```
##            model_description  TP FP FN TN accuracy error
## 1 C5.0 Simplificado Boosting 211 22 65 42    0.744 0.256
```

Los mejores modelos son CART model pruned, y C5.0 Simplificado Boosting.

Los resultados son muy similares, por lo que un diferente subset de variables o alguna discretizacion en alguna variable podria marcar mas diferencias.

Con los datos seleccionados CART model pruned proporciona el arbol mas simple posible con unas diferencias en el poder de prediccion inapreciables, razon por la cual, la opcion a elegir seria CART model pruned.