UADE - Maestria TIC - Ciencia de Datos - Modelo de predicción para aprobación de Tarjetas de Crédito

Marcelo Capozzi (LU: 1119183) - Nicolas Gladkoff (LU: 1085075)

Abril 2020

GitHub: https://github.com/ngladkoff/ds-credit-card-approval

DataSource: https://www.kaggle.com/rikdifos/credit-card-approval-prediction

Introducción

La salud de la industria de tarjetas de crédito se mide no por el número de personas con tarjeta, sino por el número de personas que pagan sus consumos.

La empresa que nos contrata actualmente toma la decisión respecto a si aprueba o no un crédito basándose en informes comerciales sobre el historial crediticio de la persona solicitante.

En los casos en que la persona no tenga historial crediticio, o que el historial crediticio sea bueno, el crédito se otorga.

Se busca brindar a la empresa una herramienta adicional que mejore la selección en los casos que no se cuente con información crediticia histórica.

Objetivo

El presente trabajo busca evaluar si es posible predecir, en base a los datos de una solicitud de tarjeta de crédito, si el crédito tiene una alta probabilidad de quedar impago (créditos malos).

Para entrenar este modelo de predicción se analizarán los datos de la información histórica y del comportamiento de pago de los clientes pasados y actuales de la entidad.

Considerando que este modelo va a utilizarse solamente para aquellas solicitudes que no cuenten con información crediticia previa, el negocio establece como suficiente que detecte 2 de cada 3 créditos malos. Además, se busca que sea capaz de detectar al menos 3 de cada 4 créditos buenos.

Datos

La entidad crediticia nos proporcionó los datos históricos que tienen disponibles, datos que deberán ser analizados para evaluar su calidad y si son suficientes para ayudarnos a resolver el problema.

Se nos proporcionó 2 datasets:

- application_record.csv: que contiene los datos de las solicitudes de tarjetas de crédito
- credit_record.csv: que contiene la información histórica del comportamiento de pagos

En pos de cumplir la meta propuesta, al set de datos proporcionado vamos a agregarle una variable Objetivo *Approve*. Esta variable se completará analizando el historial crediticio que tuvieron las solicitudes, y tendrá dos valores posibles:

Valor	Descripción
1	Representa un crédito bueno, que debería ser aprobado
0	Representa un crédito malo, que no debería ser aprobado

Diccionario de datos

Descripción de los datasets

application_record

Nombre	Descripción	Observaciones
ID	Número de Cliente	
CODE_GENDER	Género	
FLAG_OWN_CAR	Posee automóvil propio?	
FLAG_OWN_REALTY	Es propietario?	
CNT_CHILDREN	Cantidad de Hijos	
AMT_INCOME_TOTAL	Ingreso Anual (u$d)	
NAME_INCOME_TYPE	Tipo de Ingreso	
NAME_EDUCATION_TYPE	Nivel Educativo	
NAME_FAMILY_STATUS	Estado Civil	
NAME_HOUSING_TYPE	Tipo de Vivienda	
DAYS_BIRTH	Días desde la fecha de nacimiento	
DAYS_EMPLOYED	Días desde la fecha de inicio laboral	
FLAG_MOBIL	Tiene Celular?	
FLAG_WORK_PHONE	Dejó Teléfono Laboral?	
FLAG_PHONE	Dejó Teléfono Particular?	
FLAG_EMAIL	Dejó su Email?	
OCCUPATION_TYPE	Actividad Laboral	
CNT_FAM_MEMBERS	Cantidad Integrantes de la familia	

credit_record

Nombre	Descripción	Observaciones
ID	Número de Cliente	
MONTHS_BALANCE	Mes del registro	El mes al que pertenece el registro es contado hacia atrás, 0 es el mes actual, -1 el anterior y así sucesivamente.
STATUS	Estado	*Ver tabla STATUS*

STATUS

STATUS	Descripción
0	1-29 días deudor
1	30-59 días deudor
2	60-89 días deudor

STATUS	Descripción
3	90-119 días deudor
4	120-149 días deudor
5	más de 150 días deudor
C	canceló las deudas ese mes
X	sin deudas ese mes

Variables generadas al procesar los datos

Nombre	Descripción	Observaciones
AGE	Edad	
YEARS_EMPLOYED	Años en el empleo	
Approved	(Objetivo) El crédito debe o no ser aprobado	

Configuración de ambiente y carga de los datos

Configuramos las librerías a utilizar y cargamos los datasets en el ambiente de trabajo.

```
##################
# Load libraries #
##################
library(ggplot2)
library(dplyr)
library(sqldf)
library(corrplot)
library(rms)
library(Information)
library(grid)
library(ROCR)
library(DMwR)
```

Si los datasets no están en el directorio de trabajo los descargamos:

```
######################
# Download datafiles #
######################

# Set the data directory
# It will be a subdirectory 'data/' of the current directory
maindir <- getwd()
datadir <- paste(maindir, "data", sep="/")

applRecordURL <- "https://raw.githubusercontent.com/ngladkoff/ds-credit-card-approval/master/data/appli
credRecordURL <- "https://raw.githubusercontent.com/ngladkoff/ds-credit-card-approval/master/data/credi

applRecordFile <- paste(datadir, "applRecord.csv", sep="/")
credRecordFile <- paste(datadir, "credRecord.csv", sep="/")

if (!dir.exists(datadir)) {
    print(paste("Creating data directory ", datadir))
    dir.create(datadir)
}
```

```
if (!file.exists(applRecordFile)) {
    print("Downloading Application Records")
    download.file(applRecordURL, applRecordFile, method="auto")
} else {
    print(paste("Data file", applRecordFile, "already exists"))
}
```

```
## [1] "Data file C:/dev/ds-credit-card-approval/src/CCA/data/applRecord.csv already exists"
```

```
if (!file.exists(credRecordFile)) {
    print("Downloading Application Records")
    download.file(credRecordURL, credRecordFile, method="auto")
} else {
    print(paste("Data file", credRecordFile, "already exists"))
}
```

```
## [1] "Data file C:/dev/ds-credit-card-approval/src/CCA/data/credRecord.csv already exists"
```

```
########################
# Load the dataframes #
########################

dfApplications <- read.csv(applRecordFile)
dfCredits <- read.csv(credRecordFile)
```

Definimos una función que vamos a necesitar mas adelante

```
nplot <- function(plist) {
  n <- length(plist)
  grid.newpage()
  pushViewport(viewport(layout=grid.layout(n,1)))
  vplayout=function(x,y) { viewport(layout.pos.row=x, layout.pos.col=y) }
  for (i in 1:n) {
    print(plist[[i]], vp=vplayout(i,1))
  }
}
```

Definimos una semilla por si usamos generadores de datos aleatoreos, para garantizar reproducibilidad.

```
set.seed(1234)
```

Análisis Exploratorio de Datos

1 - Exámen inicial de los datos

1.1 - Dataset: Applications

```
str(dfApplications)
```

1.1.1 - Estructura

```
## 'data.frame':    438557 obs. of  18 variables:
##  $ ID                 : int  5008804 5008805 5008806 5008808 5008809 5008810 5008811 5008812 5008813
##  $ CODE_GENDER        : Factor w/ 2 levels "F","M": 2 2 2 1 1 1 1 1 1 1 ...
##  $ FLAG_OWN_CAR       : Factor w/ 2 levels "N","Y": 2 2 2 1 1 1 1 1 1 1 ...
##  $ FLAG_OWN_REALTY    : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ CNT_CHILDREN       : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ AMT_INCOME_TOTAL   : num  427500 427500 112500 270000 270000 ...
##  $ NAME_INCOME_TYPE   : Factor w/ 5 levels "Commercial associate",..: 5 5 5 1 1 1 1 2 2 2 ...
##  $ NAME_EDUCATION_TYPE: Factor w/ 5 levels "Academic degree",..: 2 2 5 5 5 5 5 2 2 2 ...
##  $ NAME_FAMILY_STATUS : Factor w/ 5 levels "Civil marriage",..: 1 1 2 4 4 4 4 3 3 3 ...
##  $ NAME_HOUSING_TYPE  : Factor w/ 6 levels "Co-op apartment",..: 5 5 2 2 2 2 2 2 2 2 ...
##  $ DAYS_BIRTH         : int  -12005 -12005 -21474 -19110 -19110 -19110 -19110 -22464 -22464 -22464 .
##  $ DAYS_EMPLOYED      : int  -4542 -4542 -1134 -3051 -3051 -3051 -3051 365243 365243 365243 ...
##  $ FLAG_MOBIL         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_WORK_PHONE    : int  1 1 0 0 0 0 0 0 0 0 ...
##  $ FLAG_PHONE         : int  0 0 0 1 1 1 1 0 0 0 ...
##  $ FLAG_EMAIL         : int  0 0 0 1 1 1 1 0 0 0 ...
##  $ OCCUPATION_TYPE    : Factor w/ 19 levels "","Accountants",..: 1 1 18 16 16 16 16 1 1 1 ...
##  $ CNT_FAM_MEMBERS    : num  2 2 2 1 1 1 1 1 1 1 ...
```

En la estructura observamos el tipo de cada variable, observando algunas que necesitan ser convertidas en categóricas:

- FLAG_MOBIL
- FLAG_WORK_PHONE
- FLAG_PHONE
- FLAG_EMAIL

Se observa también que la variable OCCUPATION_TYPE es categórica y tiene muchas categorías. Esto es algo que debemos tener en cuenta al momento de entrenar el modelo, según la bibliografía las regresiones logísticas podrían verse afectadas en su predicción ante este tipo de variables.

Por último, nos llama la atención la variable DAYS_EMPLOYED que tiene valores negativos y positivos, algunos muy grandes (en el rango de los 1000 años)

```
dfApplications$FLAG_MOBIL <- factor(dfApplications$FLAG_MOBIL, ordered=FALSE)
dfApplications$FLAG_WORK_PHONE <- factor(dfApplications$FLAG_WORK_PHONE, ordered=FALSE)
dfApplications$FLAG_PHONE <- factor(dfApplications$FLAG_PHONE, ordered=FALSE)
dfApplications$FLAG_EMAIL <- factor(dfApplications$FLAG_EMAIL, ordered=FALSE)
```

1.1.2 - Conversión variables categóricas

```
summary(dfApplications)
```

1.1.3 - Summary

```
##        ID          CODE_GENDER FLAG_OWN_CAR FLAG_OWN_REALTY  CNT_CHILDREN
##  Min.   :5008804   F:294440    N:275459     N:134483        Min.   : 0.0000
##  1st Qu.:5609375   M:144117    Y:163098     Y:304074        1st Qu.: 0.0000
##  Median :6047745                                          Median : 0.0000
##  Mean   :6022176                                          Mean   : 0.4274
##  3rd Qu.:6456971                                          3rd Qu.: 1.0000
##  Max.   :7999952                                          Max.   :19.0000
##
##  AMT_INCOME_TOTAL             NAME_INCOME_TYPE
##  Min.   :  26100   Commercial associate:100757
##  1st Qu.: 121500   Pensioner           : 75493
##  Median : 160780   State servant       : 36186
##  Mean   : 187524   Student             :    17
##  3rd Qu.: 225000   Working             :226104
##  Max.   :6750000
```

```
##
##                     NAME_EDUCATION_TYPE            NAME_FAMILY_STATUS
##  Academic degree              :   312   Civil marriage      : 36532
##  Higher education             :117522   Married             :299828
##  Incomplete higher            : 14851   Separated           : 27251
##  Lower secondary              :  4051   Single / not married: 55271
##  Secondary / secondary special:301821   Widow               : 19675
##
##
##            NAME_HOUSING_TYPE    DAYS_BIRTH     DAYS_EMPLOYED    FLAG_MOBIL
##  Co-op apartment    :  1539   Min.   :-25201   Min.   :-17531   1:438557
##  House / apartment  :393831   1st Qu.:-19483   1st Qu.: -3103
##  Municipal apartment: 14214   Median :-15630   Median : -1467
##  Office apartment   :  3922   Mean   :-15998   Mean   : 60564
##  Rented apartment   :  5974   3rd Qu.:-12514   3rd Qu.:  -371
##  With parents       : 19077   Max.   : -7489   Max.   :365243
##
##  FLAG_WORK_PHONE FLAG_PHONE FLAG_EMAIL    OCCUPATION_TYPE   CNT_FAM_MEMBERS
##  0:348156        0:312353   0:391102              :134203   Min.   : 1.000
##  1: 90401        1:126204   1: 47455   Laborers   : 78240   1st Qu.: 2.000
##                                        Core staff : 43007   Median : 2.000
##                                        Sales staff: 41098   Mean   : 2.194
##                                        Managers   : 35487   3rd Qu.: 3.000
##                                        Drivers    : 26090   Max.   :20.000
##                                        (Other)    : 80432
```

Se vuelve a observar alguna anormalidad en la variable DAYS_EMPLOYED que deberá ser analizada.

Se observa que la variable FLAG_MOBIL solo tiene una categoría, por lo que ya se puede descartar por no aportar valor al modelo.

La variable OCCUPATION_TYPE además de tener muchas categorías, como ya se había notado anteriormente, parece tener datos incompletos. Deberá ser analizado.

Respecto a la cantidad de hijos y el ingreso total vamos a tener que evaluar la distribución, los máximos y mínimos están muy lejos del promedio y la mediana.

1.1.4 - Búsqueda de IDs duplicados Para asegurarnos que ambos datasets se puedan juntar una vez procesados debemos verificar no tener IDs duplicados en el dataset de Applications.

```
dfAppIds <- data.frame(table(dfApplications$ID))
cUniAppIds <- dim(dfApplications[unique(dfApplications$ID),])[1]
cDupAppIds <- dim(dfApplications[duplicated(dfApplications$ID),])[1]
cDupAppRecord <- dim(dfApplications[duplicated(dfApplications),])[1]
dfDupAppIds <- dfApplications[duplicated(dfApplications$ID),]
cAppIds <- dim(dfApplications)[1]
print(paste("IDs únicos:", cUniAppIds))
```

```
## [1] "IDs únicos: 438510"
```

```
print(paste("IDs duplicados:", cDupAppIds))
```

```
## [1] "IDs duplicados: 47"
```

```
print(paste("Records duplicados: ", cDupAppRecord))
```

```
## [1] "Records duplicados:  0"
```

```
print(paste("Total:", cAppIds))
```

```
## [1] "Total: 438557"
```

```
print(paste("Duplicados: %", (round((cDupAppIds*100/cAppIds), digits=2))))
```

```
## [1] "Duplicados: % 0.01"
```

Observamos que no tenemos registros duplicados, es decir, que todos sus valores coincidan. Sin embargo, detectamos 47 IDs duplicados. Concluimos que no corresponden a una misma solicitud cargada más de una vez sino solicitudes distintas con la misma numeración.

Por el bajo porcentaje que representan se decide eliminar del análisis estos registros duplicados.

```
dfDupAppIds2 <- data.frame(dfDupAppIds$ID)
dfDupIds <- sqldf("SELECT distinct ID FROM dfApplications WHERE ID in dfDupAppIds2")
dfApplicationsCleaned <- sqldf("SELECT * FROM dfApplications WHERE ID NOT in dfDupAppIds2")
cNewTotal <- dim(dfApplicationsCleaned)[1]
print(paste("Cantidad Anterior - Cantidad Nueva: ",cAppIds - cNewTotal))
```

```
## [1] "Cantidad Anterior - Cantidad Nueva:  94"
```

Validamos que la cantidad nueva tiene 94 registros menos (47 duplicados x 2).

1.2 - Dataset: Credits

```
str(dfCredits)
```

1.2.1 - Estructura

```
## 'data.frame':    1048575 obs. of  3 variables:
##  $ ID            : int  5001711 5001711 5001711 5001711 5001712 5001712 5001712 5001712 5001712 5001
##  $ MONTHS_BALANCE: int  0 -1 -2 -3 0 -1 -2 -3 -4 -5 ...
##  $ STATUS        : Factor w/ 8 levels "0","1","2","3",..: 8 1 1 1 7 7 7 7 7 7 ...
```

Observamos que por cada ID de solicitud se crean varios registros, uno por cada mes, con el estado de la deuda en dicho mes.

Se imprimen los primeros y últimos registros para confirmar:

```
head(dfCredits)
```

```
##        ID MONTHS_BALANCE STATUS
## 1 5001711              0      X
## 2 5001711             -1      0
## 3 5001711             -2      0
## 4 5001711             -3      0
## 5 5001712              0      C
## 6 5001712             -1      C
```

```
tail(dfCredits)
```

```
##              ID MONTHS_BALANCE STATUS
## 1048570 5150487            -24      C
## 1048571 5150487            -25      C
## 1048572 5150487            -26      C
## 1048573 5150487            -27      C
## 1048574 5150487            -28      C
## 1048575 5150487            -29      C
```

Para poder armar nuestra variable objetivo *Approve* vamos a necesitar reducir estos datos a un solo registro por ID.

```
summary(dfCredits)
```

1.2.2 - Summary

```
##        ID          MONTHS_BALANCE       STATUS
##  Min.   :5001711   Min.   :-60.00   C      :442031
##  1st Qu.:5023644   1st Qu.:-29.00   0      :383120
##  Median :5062104   Median :-17.00   X      :209230
##  Mean   :5068286   Mean   :-19.14   1      : 11090
##  3rd Qu.:5113856   3rd Qu.: -7.00   5      :  1693
##  Max.   :5150487   Max.   :  0.00   2      :   868
##                                     (Other):   543
```

No se observan números fuera de rango dentro de la variable MONTHS_BALANCE.

2 - Reducción dataset Credits

2.1 - Identificar IDs únicos

```
dfCredIds <- data.frame(table(dfCredits$ID))
```

```
str(dfCredIds)
```

2.1.1 - Estructura

```
## 'data.frame':    45985 obs. of  2 variables:
##  $ Var1: Factor w/ 45985 levels "5001711","5001712",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Freq: int  4 19 22 15 60 22 39 43 36 31 ...
```

Se observa que este dataset solo cuenta con 45985 IDs únicos. Esto nos indica que de los poco más de 438000 registros del dataset de Applications solo tenemos información de algunos.

```
summary(dfCredIds)
```

2.1.2 - Summary

```
##       Var1            Freq
##  5001711:    1   Min.   : 1.0
##  5001712:    1   1st Qu.:10.0
##  5001713:    1   Median :19.0
##  5001714:    1   Mean   :22.8
##  5001715:    1   3rd Qu.:34.0
##  5001717:    1   Max.   :61.0
##  (Other):45979
```

2.2 - Reducir registros

Se decidió que vamos a tomar todos los datos en la misma fecha, el mes 0 del que tenemos información.

Para esto recuperamos primero cual es ese último mes:

```
dfCreditsReduced <- sqldf("SELECT ID, MONTHS_BALANCE, STATUS FROM dfCredits WHERE MONTHS_BALANCE=0")
str(dfCreditsReduced)
```

```
## 'data.frame':    33856 obs. of  3 variables:
##  $ ID            : int  5001711 5001712 5001713 5001714 5001715 5001717 5001718 5001719 5001720 5001
##  $ MONTHS_BALANCE: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ STATUS        : Factor w/ 8 levels "0","1","2","3",..: 8 7 8 8 8 7 7 7 2 8 ...
```

Definimos nuestra variable objetivo: *Approve*. Si el estado del crédito en el último mes que se tiene registro no quedó con deuda (estados C, X y 0) se va a establecer un uno, y si quedó en otro de los estados, un cero.

```
dfCreditsReduced$Approve <- factor(ifelse(dfCreditsReduced$STATUS %in% as.character(1:5), 0, 1))
summary(dfCreditsReduced)
```

```
##        ID          MONTHS_BALANCE     STATUS      Approve
##  Min.   :5001711   Min.   :0      C      :17613   0:  404
##  1st Qu.:5023387   1st Qu.:0      0      : 8914   1:33452
##  Median :5062036   Median :0      X      : 6925
##  Mean   :5068049   Mean   :0      1      :  309
##  3rd Qu.:5113808   3rd Qu.:0      5      :   65
##  Max.   :5150487   Max.   :0      2      :   19
##                                   (Other):   11
```

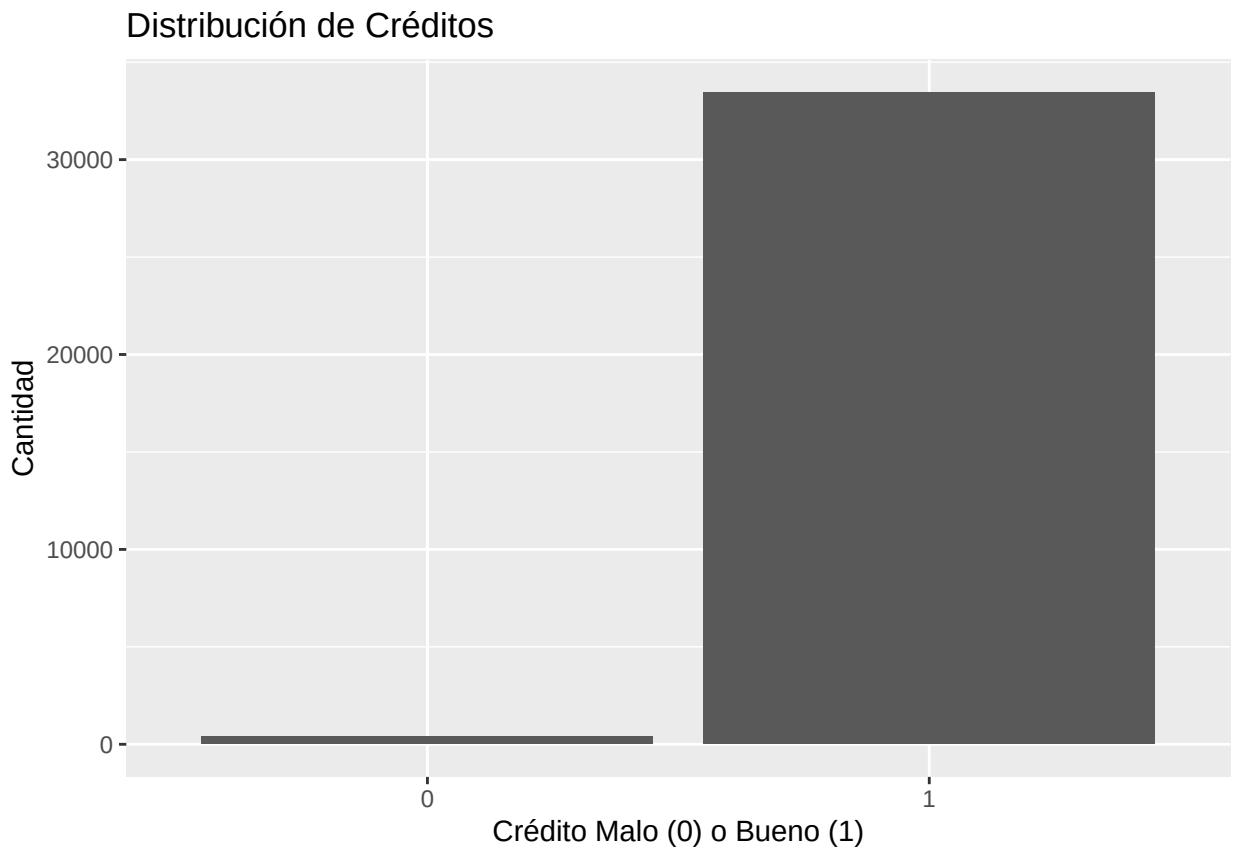
```
print(paste("% Créditos malos: ", round(100 * length(which(dfCreditsReduced$Approve == 0)) / length(whi
```

```
## [1] "% Créditos malos:  1.21"
```

Se observa que la cantidad de créditos "malos" es de aproximadamente el 1.2%, que representa un porcentaje bajo de incobrabilidad. Según lo que hemos investigado suele ser lijeramente superior al 3%.

A continuación graficamos la distribución de créditos.

```
ggplot(dfCreditsReduced) +
  geom_bar(aes(x=Approve)) +
  labs(title="Distribución de Créditos", x="Crédito Malo (0) o Bueno (1)", y="Cantidad")
```



3 - Unificar los datasets

3.1 - Unificar

```
dfCreditCard <- merge(dfApplicationsCleaned, dfCreditsReduced, by="ID")
```

3.2 - Estructura

```
str(dfCreditCard)
```

```
## 'data.frame':    24672 obs. of  21 variables:
##  $ ID                 : int  5008804 5008805 5008806 5008808 5008810 5008811 5008813 5008815 5008821
##  $ CODE_GENDER        : Factor w/ 2 levels "F","M": 2 2 2 1 1 1 1 2 2 2 ...
##  $ FLAG_OWN_CAR       : Factor w/ 2 levels "N","Y": 2 2 2 1 1 1 1 2 2 2 ...
##  $ FLAG_OWN_REALTY    : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ CNT_CHILDREN       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AMT_INCOME_TOTAL   : num  427500 427500 112500 270000 270000 ...
##  $ NAME_INCOME_TYPE   : Factor w/ 5 levels "Commercial associate",..: 5 5 5 1 1 1 2 5 1 1 ...
##  $ NAME_EDUCATION_TYPE: Factor w/ 5 levels "Academic degree",..: 2 2 5 5 5 5 2 2 5 5 ...
##  $ NAME_FAMILY_STATUS : Factor w/ 5 levels "Civil marriage",..: 1 1 2 4 4 4 3 2 2 2 ...
##  $ NAME_HOUSING_TYPE  : Factor w/ 6 levels "Co-op apartment",..: 5 5 2 2 2 2 2 2 2 2 ...
##  $ DAYS_BIRTH         : int  -12005 -12005 -21474 -19110 -19110 -19110 -22464 -16872 -17778 -17778 .
##  $ DAYS_EMPLOYED      : int  -4542 -4542 -1134 -3051 -3051 -3051 365243 -769 -1194 -1194 ...
##  $ FLAG_MOBIL         : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_WORK_PHONE    : Factor w/ 2 levels "0","1": 2 2 1 1 1 1 1 2 1 1 ...
##  $ FLAG_PHONE         : Factor w/ 2 levels "0","1": 1 1 1 2 2 2 1 2 1 1 ...
```

```
##  $ FLAG_EMAIL         : Factor w/ 2 levels "0","1": 1 1 1 2 2 2 1 2 1 1 ...
##  $ OCCUPATION_TYPE    : Factor w/ 19 levels "","Accountants",..: 1 1 18 16 16 16 1 2 10 10 ...
##  $ CNT_FAM_MEMBERS    : num  2 2 2 1 1 1 1 2 2 2 ...
##  $ MONTHS_BALANCE     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ STATUS             : Factor w/ 8 levels "0","1","2","3",..: 7 7 7 1 7 7 1 1 8 1 ...
##  $ Approve            : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

3.3 - Summary

```
summary(dfCreditCard)
```

```
##        ID          CODE_GENDER FLAG_OWN_CAR FLAG_OWN_REALTY  CNT_CHILDREN
##  Min.   :5008804   F:16406     N:15301      N: 8379         Min.   : 0.0000
##  1st Qu.:5045511   M: 8266     Y: 9371      Y:16293         1st Qu.: 0.0000
##  Median :5069454                                          Median : 0.0000
##  Mean   :5078863                                          Mean   : 0.4175
##  3rd Qu.:5115437                                          3rd Qu.: 1.0000
##  Max.   :5150487                                          Max.   :19.0000
##
##  AMT_INCOME_TOTAL              NAME_INCOME_TYPE
##  Min.   :  27000   Commercial associate: 5849
##  1st Qu.: 121500   Pensioner           : 4150
##  Median : 157500   State servant       : 1953
##  Mean   : 187075   Student             :    9
##  3rd Qu.: 225000   Working             :12711
##  Max.   :1575000
##
##                      NAME_EDUCATION_TYPE            NAME_FAMILY_STATUS
##  Academic degree              :   23    Civil marriage      : 1965
##  Higher education             : 6651    Married             :16942
##  Incomplete higher            :  966    Separated           : 1434
##  Lower secondary              :  250    Single / not married: 3319
##  Secondary / secondary special:16782    Widow               : 1012
##
##
##            NAME_HOUSING_TYPE   DAYS_BIRTH     DAYS_EMPLOYED    FLAG_MOBIL
##  Co-op apartment    :  108   Min.   :-25152   Min.   :-15713   1:24672
##  House / apartment  :22020   1st Qu.:-19453   1st Qu.: -3170
##  Municipal apartment:  785   Median :-15653   Median : -1546
##  Office apartment   :  179   Mean   :-16023   Mean   : 58970
##  Rented apartment   :  374   3rd Qu.:-12542   3rd Qu.:  -401
##  With parents       : 1206   Max.   : -7489   Max.   :365243
##
##  FLAG_WORK_PHONE FLAG_PHONE FLAG_EMAIL    OCCUPATION_TYPE CNT_FAM_MEMBERS
##  0:19049         0:17395    0:22410               :7629   Min.   : 1.000
##  1: 5623         1: 7277    1: 2262    Laborers   :4293   1st Qu.: 2.000
##                                        Core staff :2381   Median : 2.000
##                                        Sales staff:2328   Mean   : 2.184
##                                        Managers   :2008   3rd Qu.: 3.000
##                                        Drivers    :1504   Max.   :20.000
##                                        (Other)    :4529
##  MONTHS_BALANCE     STATUS      Approve
##  Min.   :0      C      :12974   0:  325
##  1st Qu.:0      0      : 6886   1:24347
```

```
##  Median :0      X      : 4487
##  Mean   :0      1      :  236
##  3rd Qu.:0      5      :   59
##  Max.   :0      2      :   19
##                 (Other):   11
```

3.4 - Conclusiones de la unificación

Luego de unir ambos datasets observamos que la cantidad de registros de clientes que poseen información histórica ha disminuido.

Esto se debe a que no todas las solicitudes de tarjeta de crédito tienen asociado un historial crediticio registrado.

Para los fines de este análisis, aceptaremos cómo válida esta reducción en el set de datos.

```
print(paste("Registros en el historial:", dim(dfCreditsReduced)[1]))
```

```
## [1] "Registros en el historial: 33856"
```

```
print(paste("Solicitudes de tarjetas de crédito con registros en el historial:", dim(dfCreditCard)[1]))
```

```
## [1] "Solicitudes de tarjetas de crédito con registros en el historial: 24672"
```

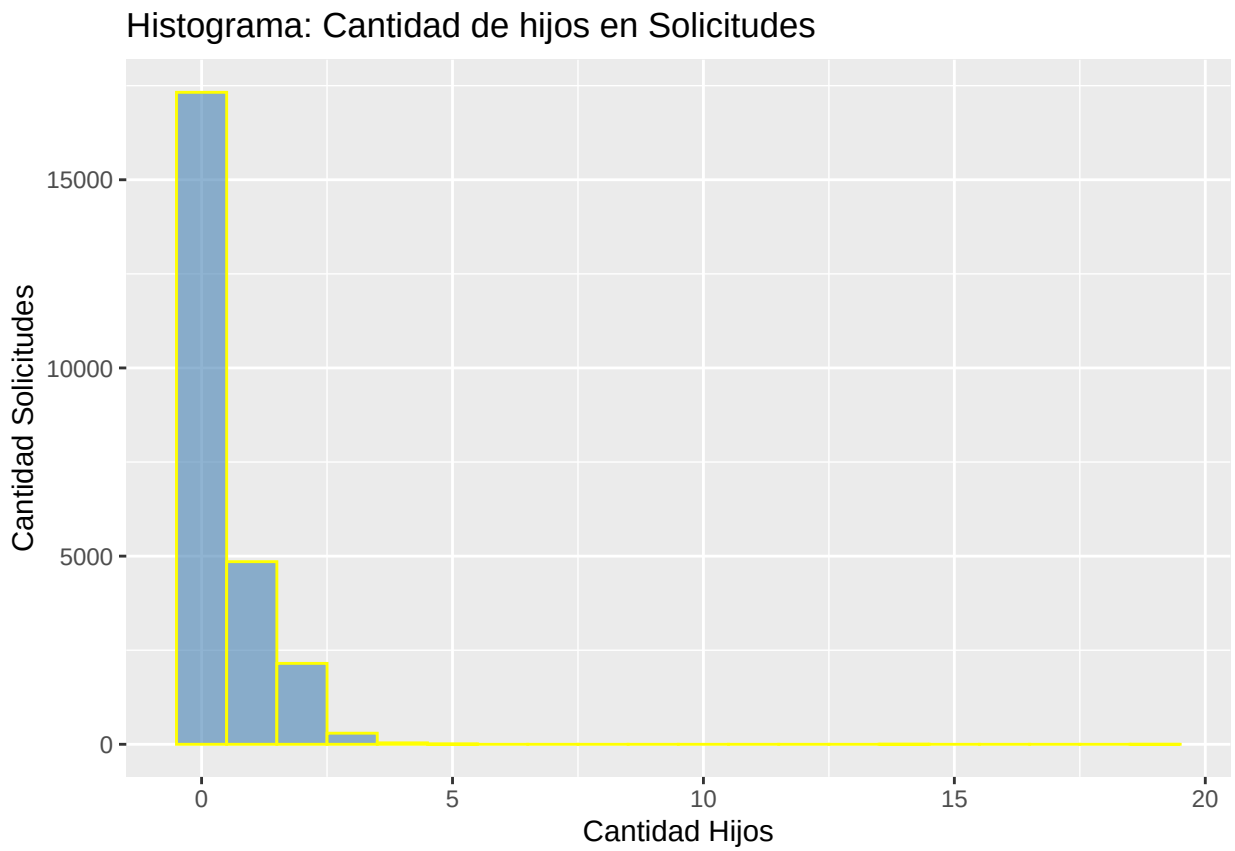
4 - Análisis de variables sobre dataset unificado

4.1 - Análisis de variables númericas

4.1.1. - Variable: CNT_CHILDREN

CNT_CHILDREN En relación a la variable *CNT_CHILDREN* observaremos la distribución de la cantidad total de hijos en cada barra del siguiente gráfico:

```
ggplot(dfCreditCard) +
  geom_histogram(
    aes(x=CNT_CHILDREN),
    fill="steel blue",
    col= "yellow",
    alpha= .6,
    binwidth = 1) +
  labs(title="Histograma: Cantidad de hijos en Solicitudes", x="Cantidad Hijos", y="Cantidad Solicitudes
```



Se observa una marcada concentración respeto de la cantidad de hijos cuando dicho valor es menor o igual a 2.

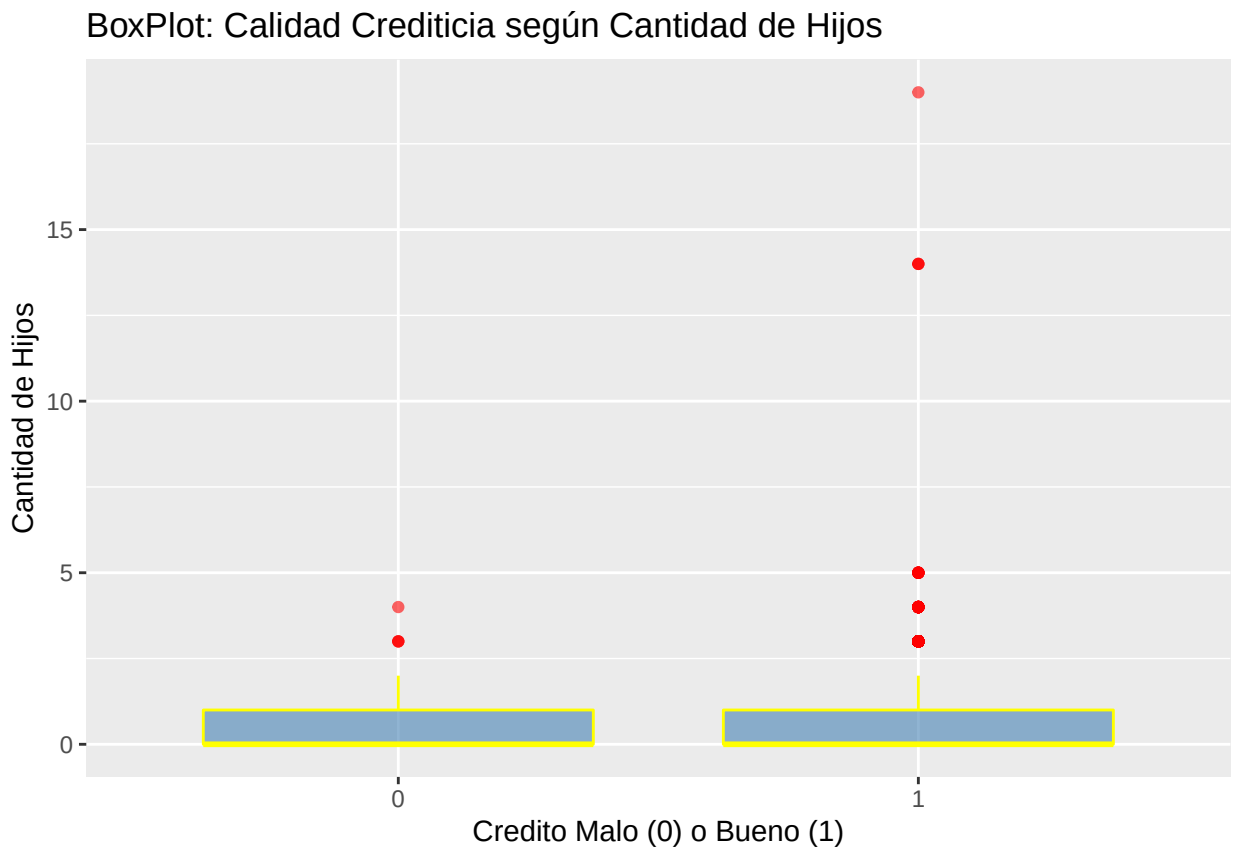
```
summary(dfCreditCard$CNT_CHILDREN)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.4175  1.0000 19.0000
```

Se observa en el summary la presencia de valores cercanos a 20, pero como puede observarse en el gráfico anterior son outliers, no hay concentración en esas cantidades.

CNT_CHILDREN vs APPROVE Vamos a utilizar el gráfico BoxPlot para tratar de entender como se relacionan la variable CNT_CHILDREN y nuestra variable Objetivo.

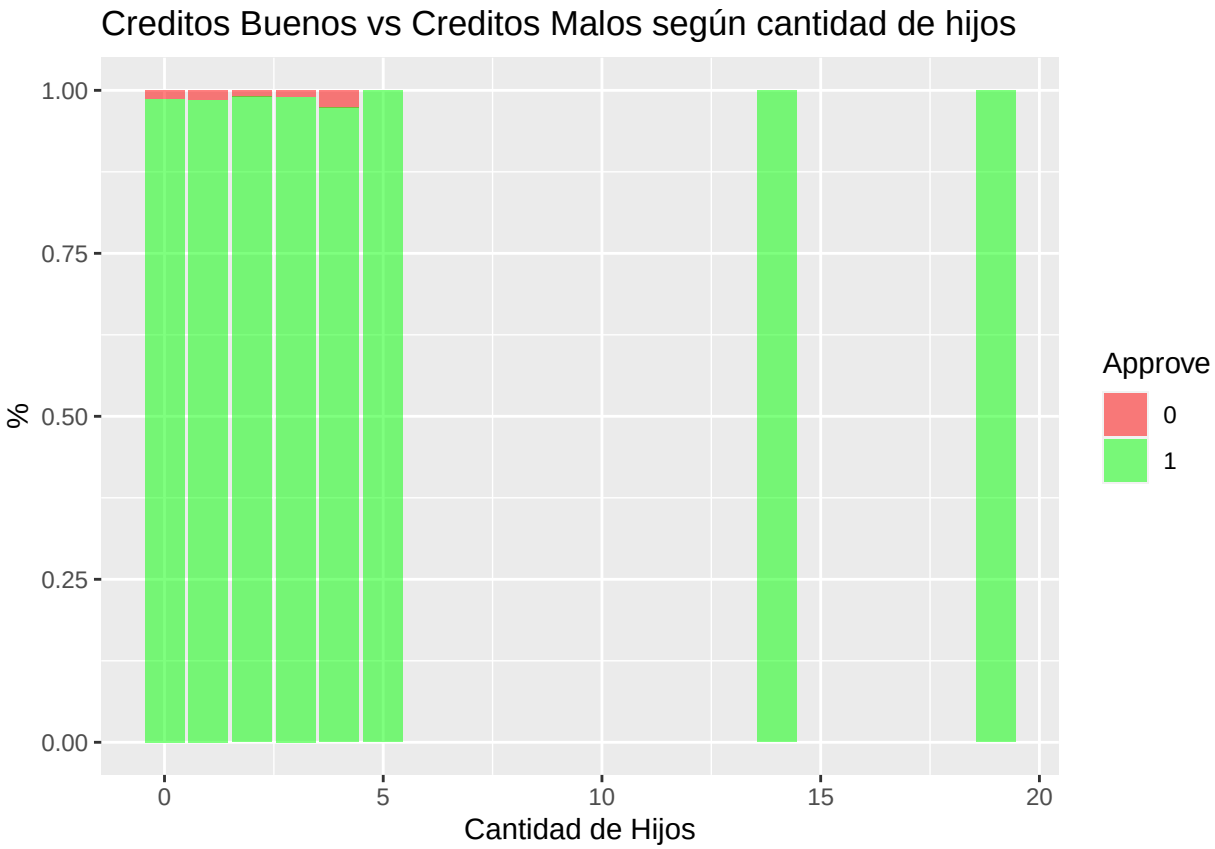
```
ggplot(dfCreditCard) +
  geom_boxplot(aes(x=Approve, y=CNT_CHILDREN), outlier.color = "red", col="yellow", fill="steel blue", a
  labs(title="BoxPlot: Calidad Crediticia según Cantidad de Hijos", x="Credito Malo (0) o Bueno (1)", y=
```



Observamos que la cantidad de casos aprobados en comparación de los que no, se mantienen en valores similares.

Vamos a intentar representarlo de manera diferente.

```
ggplot(dfCreditCard) +
  geom_bar(aes(x=CNT_CHILDREN, fill=Approve), position="fill", alpha=.5) +
  labs(title="Creditos Buenos vs Creditos Malos según cantidad de hijos", x="Cantidad de Hijos", y="%")
  scale_fill_manual(values= c("0"= "red", "1"="green"))
```

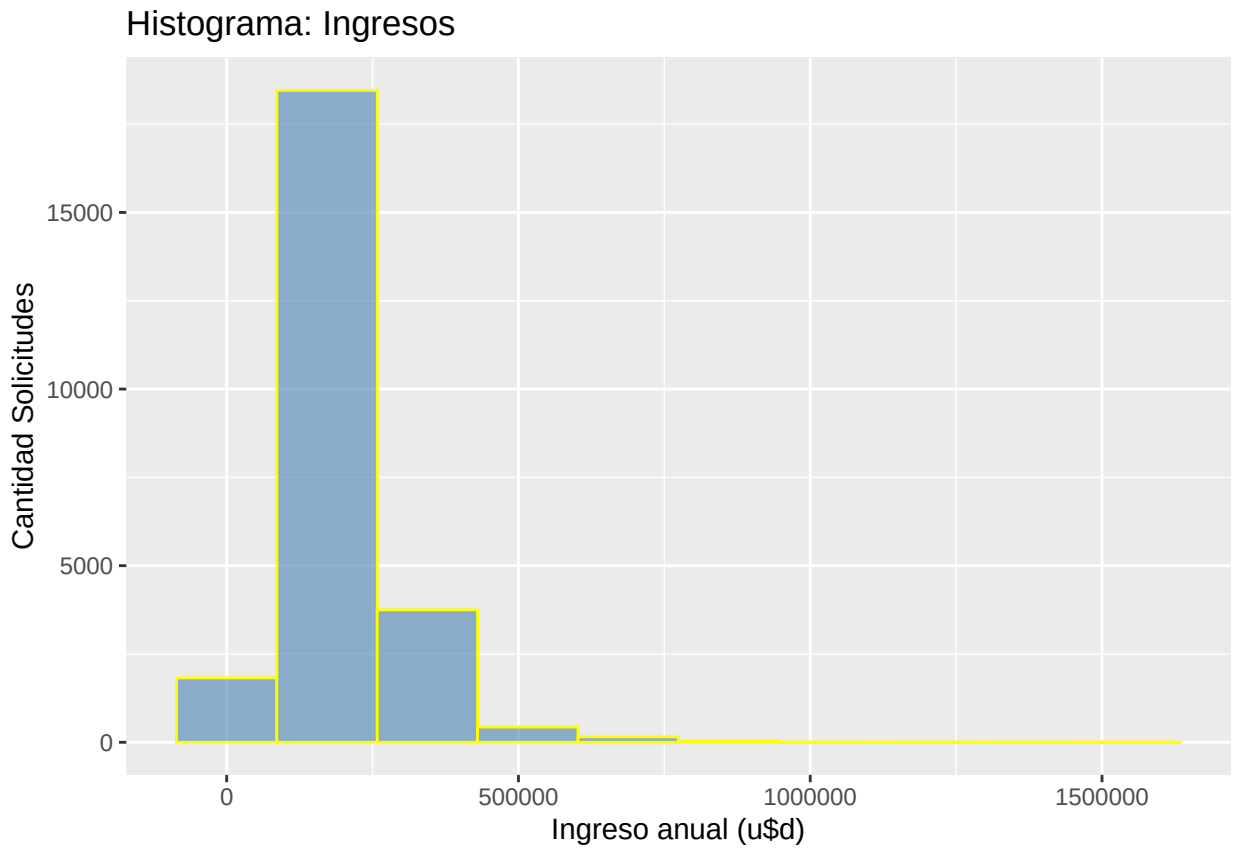


No se observa en el gráfico una relación aparente entre la variable CNT_CHILDREN y la varible objetivo.

4.1.2. - Variable: AMT_INCOME_TOTAL

AMT_INCOME_TOTAL En relación a la variable *AMT_INCOME_TOTAL* observaremos su distribuciónen el siguiente gráfico:

```
ggplot(dfCreditCard) +
  geom_histogram(
    aes(x=AMT_INCOME_TOTAL),
    fill="steel blue",
    col= "yellow",
    alpha= .6, bins=10) +
  labs(title="Histograma: Ingresos", x="Ingreso anual (u$d)", y="Cantidad Solicitudes")
```



Se observa una marcada concentración de solicitudes en un rango medio de ingresos anuales. si bien se observa la presencia de algunas solicitudes con ingresos muy superiores.

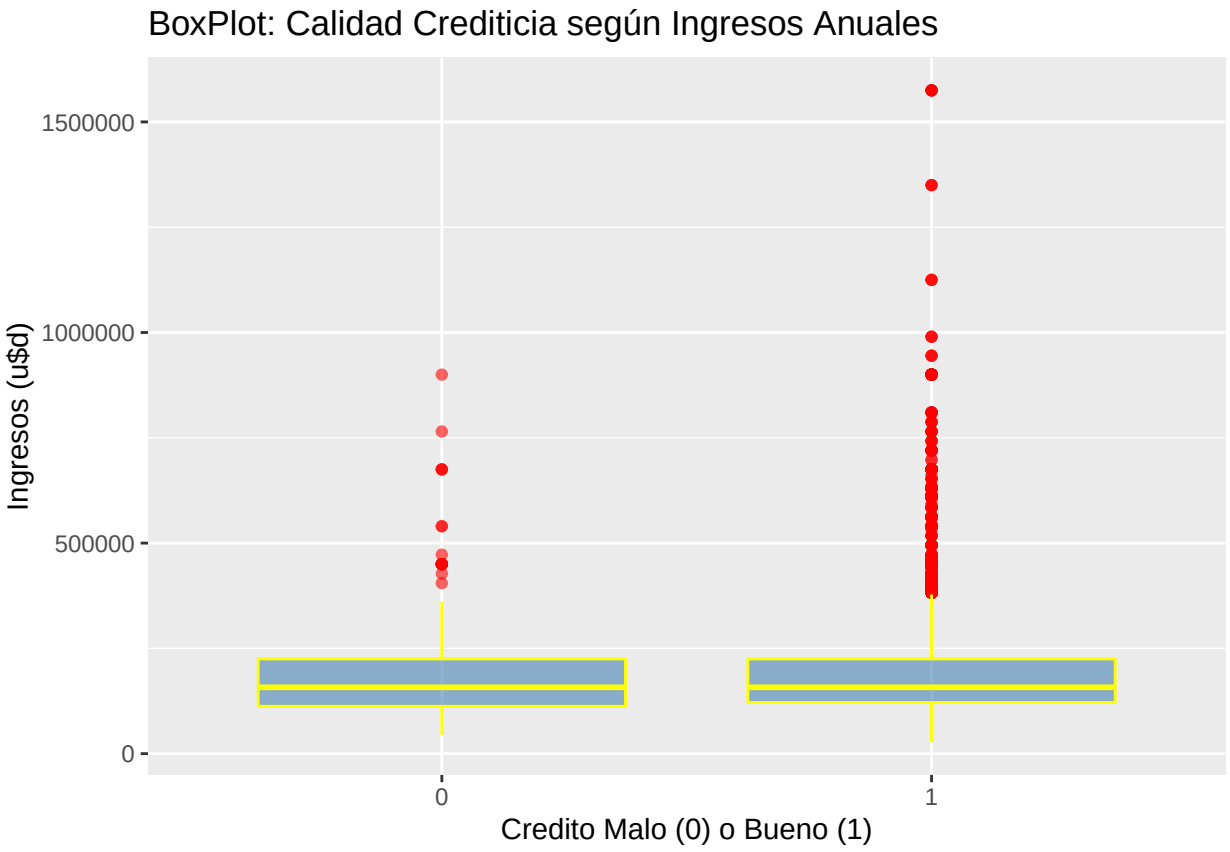
```
summary(dfCreditCard$AMT_INCOME_TOTAL)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   27000  121500  157500  187075  225000 1575000
```

Dentro del rango de los 120000 y los 225000 dolares anuales de ingreso se concentra el 50% de las solicitudes de tarjeta de crédito. Vemos la presencia de valores muy inferiores y muy superiores.

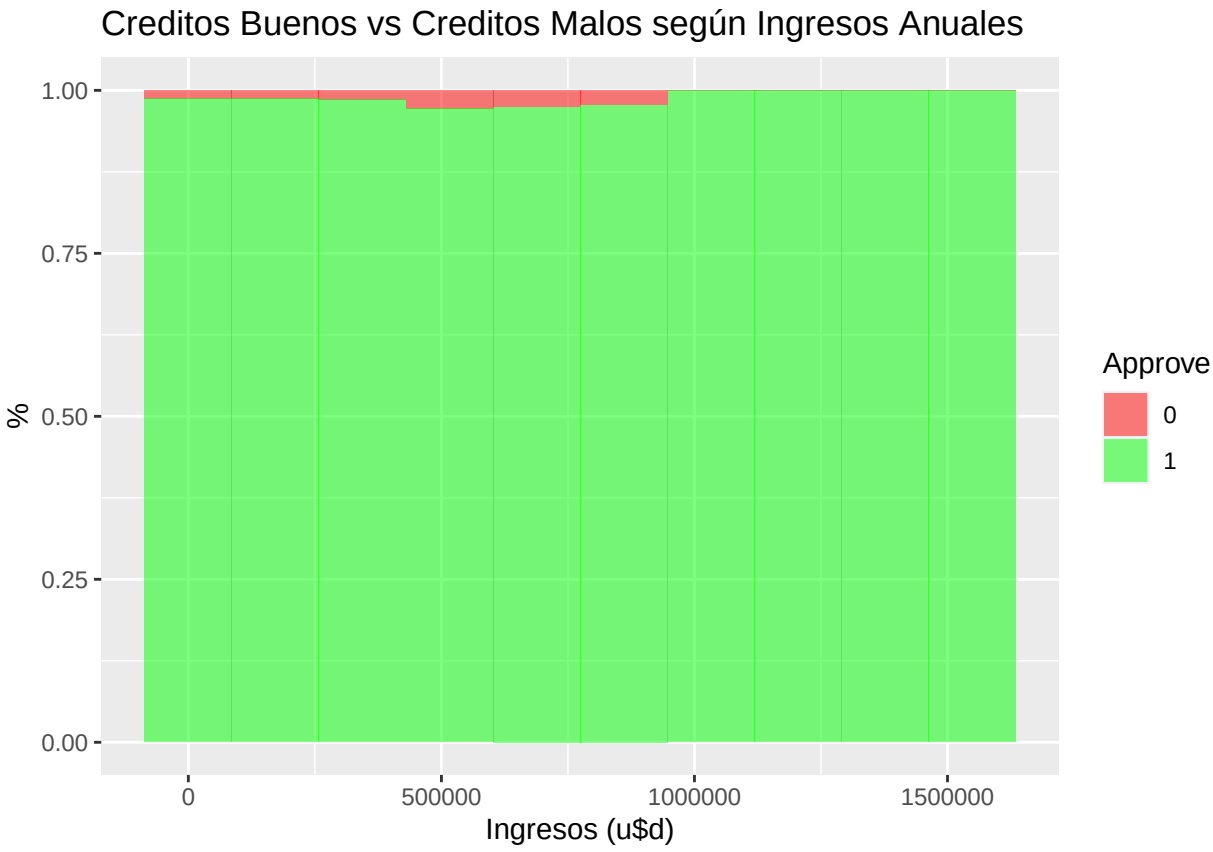
AMT_INCOME_TOTAL vs APPROVE Vamos a utilizar el gráfico BoxPlot para tratar de entender como se relacionan la variable AMT_INCOME_TOTAL y nuestra variable Objetivo.

```
ggplot(dfCreditCard) +
  geom_boxplot(aes(x=Approve, y=AMT_INCOME_TOTAL), outlier.color = "red", col="yellow", fill="steel blu
  labs(title="BoxPlot: Calidad Crediticia según Ingresos Anuales", x="Credito Malo (0) o Bueno (1)", y=
```



No observamos diferencias significativas en este gráfico. Vamos a intentar representarlo de manera diferente.

```
ggplot(dfCreditCard) +
  geom_histogram(aes(x=AMT_INCOME_TOTAL, fill=Approve), position="fill", alpha=.5, bins=10) +
  labs(title="Creditos Buenos vs Creditos Malos según Ingresos Anuales", x="Ingresos (u$d)", y="%") +
  scale_fill_manual(values= c("0"= "red", "1"="green"))
```



Se observa una ligera relación entre las variables. Podría ser un posible predictor.

4.1.3. - Variable: DAYS_BIRTH

```
summary(dfCreditCard$DAYS_BIRTH)
```

DAYS_BIRTH

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -25152  -19453  -15653  -16023  -12542   -7489
```

Como se observa, está expresado en días y en negativo, vamos a convertirlo a una escala a la que estamos más acostumbrados a ver, que es expresar la edad en años.

```
dfCreditCard$AGE <- (dfCreditCard$DAYS_BIRTH * -1) %/% 365
summary(dfCreditCard$AGE)
```

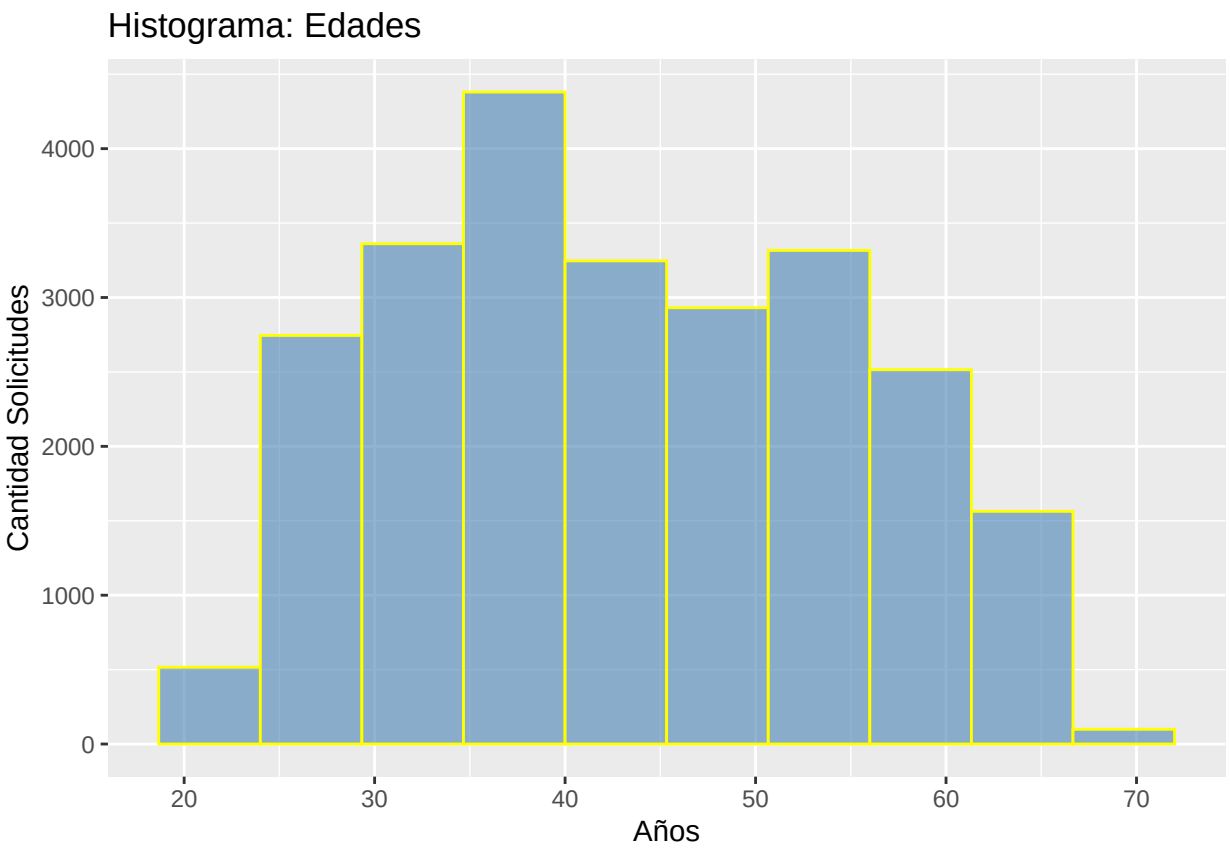
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   20.00   34.00   42.00   43.39   53.00   68.00
```

Expresado de esta manera podemos observar que los valores son razonables.

Observaremos la distribución en el siguiente gráfico:

```
ggplot(dfCreditCard) +
  geom_histogram(
    aes(x=AGE),
    fill="steel blue",
```

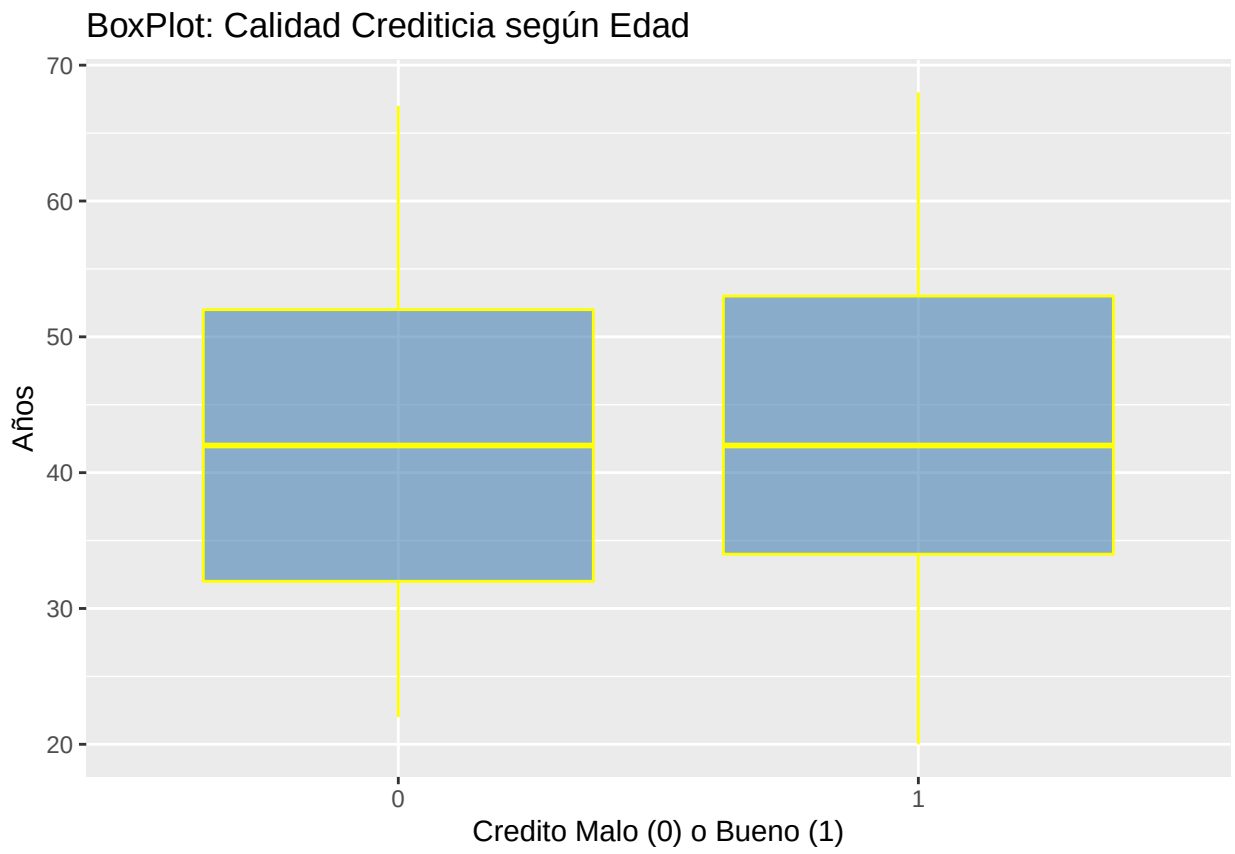
```
    col= "yellow",
    alpha= .6, bins=10) +
  labs(title="Histograma: Edades", x="Años", y="Cantidad Solicitudes")
```



Observamos una distribución con cierta semejanza a una campana de Gauss (Normal).

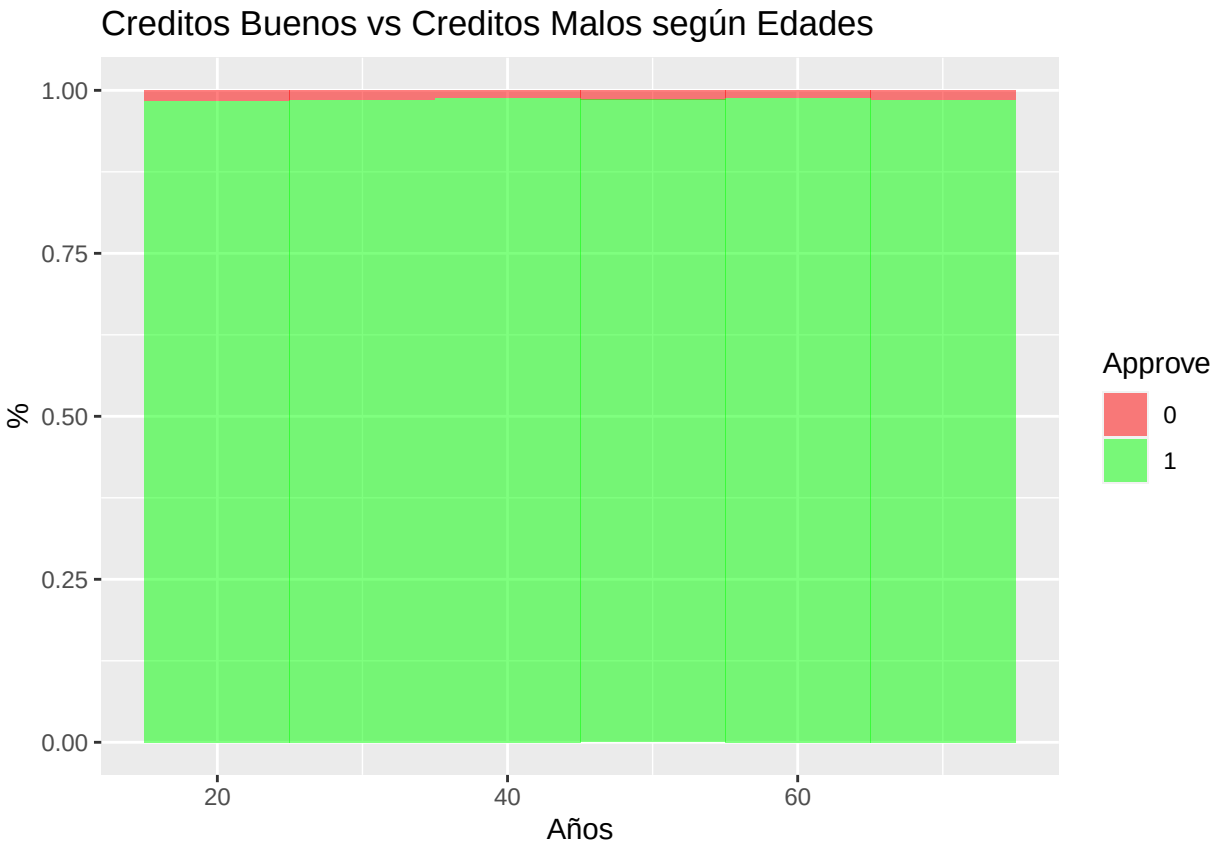
AGE vs APPROVE Vamos a utilizar el gráfico BoxPlot para tratar de entender como se relacionan la variable AGE y nuestra variable Objetivo.

```
ggplot(dfCreditCard) +
  geom_boxplot(aes(x=Approve, y=AGE), outlier.color = "red", col="yellow", fill="steel blue", alpha=0.6)
  labs(title="BoxPlot: Calidad Crediticia según Edad", x="Credito Malo (0) o Bueno (1)", y="Años")
```



Se observa alguna leve tendencia. Vamos a intentar representarlo de manera diferente.

```
ggplot(dfCreditCard) +
  geom_histogram(aes(x=AGE, fill=Approve), position="fill", alpha=.5, binwidth = 10) +
  labs(title="Creditos Buenos vs Creditos Malos según Edades", x="Años", y="%") +
  scale_fill_manual(values= c("0"= "red", "1"="green"))
```



Parece existir una ligera tendencia a que los créditos mejoren a medida que aumenta la edad. Podría ser un posible predictor.

4.1.4. - Variable: DAYS_EMPLOYED

```
summary(dfCreditCard$DAYS_EMPLOYED)
```

DAYS_EMPLOYED

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -15713   -3170   -1546   58970    -401  365243
```

Como se observa, está expresado en días y en negativo, vamos a convertirlo a una escala a la que estamos más acostumbrados a ver, que es expresarlo en años. Se observan también valores positivos, que se les va a tener que dar un tratamiento.

```
dfCreditCard$YEARS_EMPLOYED <- (dfCreditCard$DAYS_EMPLOYED * -1) %/% 365
summary(dfCreditCard$YEARS_EMPLOYED)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -1001       1       4    -162       8      43
```

Expresado de esta manera podemos observar edades negativas.

Buscamos estos valores:

```
dfNegatives <- filter(dfCreditCard, YEARS_EMPLOYED < 0)
str(dfNegatives)
```

```
## 'data.frame':    4133 obs. of  23 variables:
##  $ ID                 : int  5008813 5008827 5008884 5008976 5008977 5008979 5009033 5009035 5009036
##  $ CODE_GENDER        : Factor w/ 2 levels "F","M": 1 2 1 1 1 1 1 1 1 ...
##  $ FLAG_OWN_CAR       : Factor w/ 2 levels "N","Y": 1 2 1 1 1 1 1 1 1 ...
##  $ FLAG_OWN_REALTY    : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 1 1 1 ...
##  $ CNT_CHILDREN       : int  0 0 0 0 0 0 0 0 0 ...
##  $ AMT_INCOME_TOTAL   : num  283500 180000 315000 112500 112500 ...
##  $ NAME_INCOME_TYPE   : Factor w/ 5 levels "Commercial associate",..: 2 2 2 2 2 2 2 2 2 ...
##  $ NAME_EDUCATION_TYPE: Factor w/ 5 levels "Academic degree",..: 2 2 5 5 5 5 3 3 3 3 ...
##  $ NAME_FAMILY_STATUS : Factor w/ 5 levels "Civil marriage",..: 3 2 5 2 2 2 1 1 1 1 ...
##  $ NAME_HOUSING_TYPE  : Factor w/ 6 levels "Co-op apartment",..: 2 2 2 2 2 2 5 5 5 5 ...
##  $ DAYS_BIRTH         : int  -22464 -18772 -20186 -22319 -22319 -22319 -18682 -18682 -18682 -18682 .
##  $ DAYS_EMPLOYED      : int  365243 365243 365243 365243 365243 365243 365243 365243 365243 365243 .
##  $ FLAG_MOBIL         : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_WORK_PHONE    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_PHONE         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_EMAIL         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 ...
##  $ OCCUPATION_TYPE    : Factor w/ 19 levels "","Accountants",..: 1 1 1 1 1 1 1 1 1 ...
##  $ CNT_FAM_MEMBERS    : num  1 2 1 2 2 2 2 2 2 ...
##  $ MONTHS_BALANCE     : int  0 0 0 0 0 0 0 0 0 ...
##  $ STATUS             : Factor w/ 8 levels "0","1","2","3",..: 1 7 7 1 1 7 8 8 7 8 ...
##  $ Approve            : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 ...
##  $ AGE                : num  61 51 55 61 61 61 51 51 51 51 ...
##  $ YEARS_EMPLOYED     : num  -1001 -1001 -1001 -1001 -1001 ...
```

Observamos un número importante de registros con valores negativos (inválidos). Cómo no podemos consultar a negocio que significan estos valores decidimos interpretarlos cómo "antigüedad laboral desconocida", los vamos a reemplazar por la mediana, que es más representativa de la antigüedad que el promedio, justamente por la presencia de estos valores.

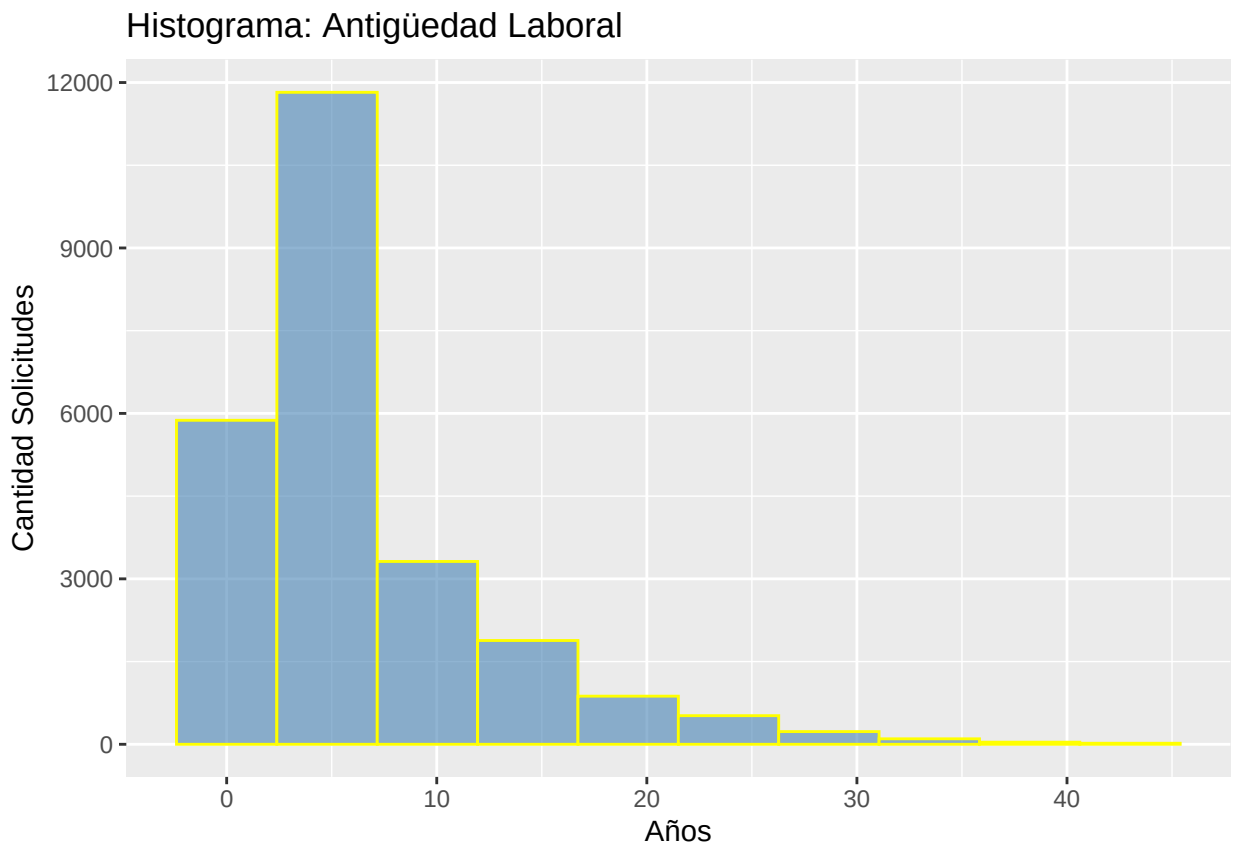
```
dfCreditCard$YEARS_EMPLOYED[dfCreditCard$YEARS_EMPLOYED < 0] <- median(dfCreditCard$YEARS_EMPLOYED)

summary(dfCreditCard$YEARS_EMPLOYED)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.000   4.000   6.321   8.000  43.000
```

Ahora los valores parecen razonables.

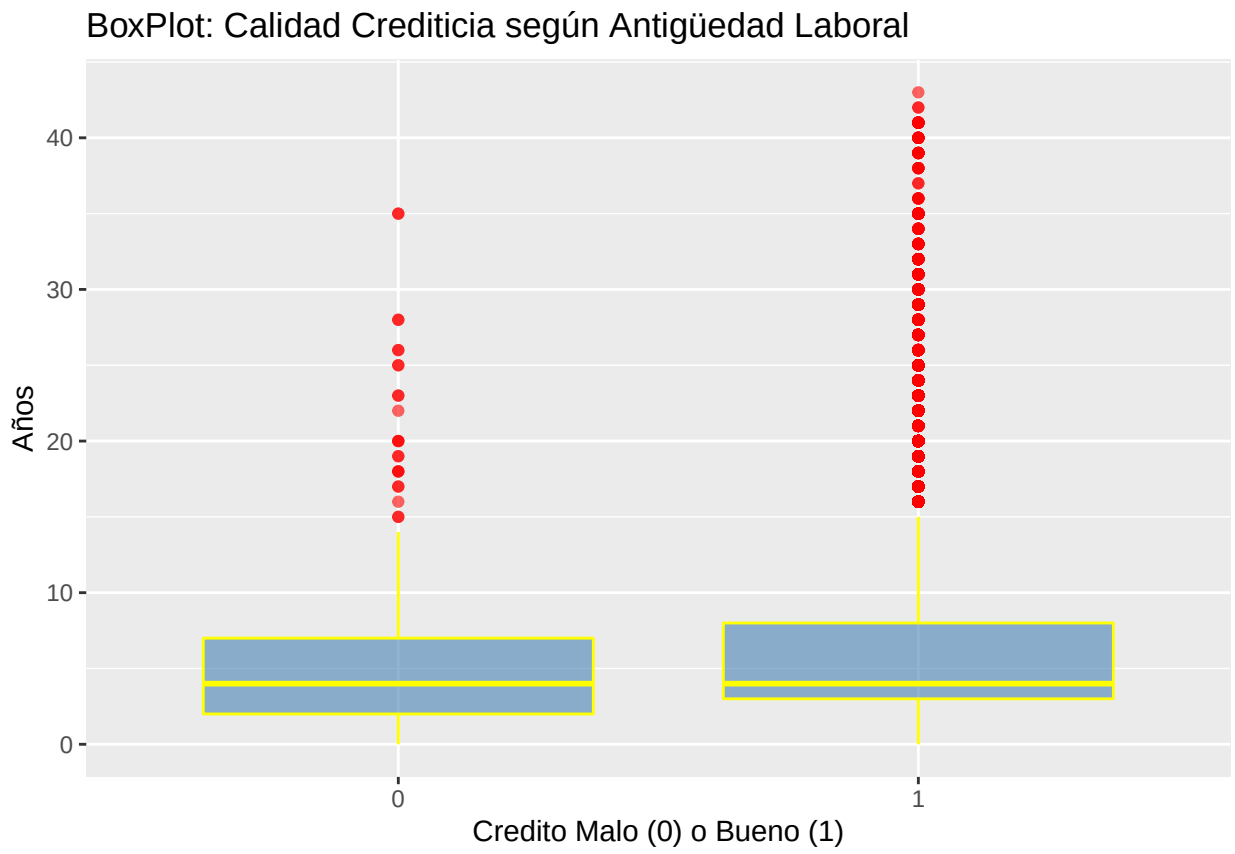
Observaremos la distribución en el siguiente gráfico:

```
ggplot(dfCreditCard) +
  geom_histogram(
    aes(x=YEARS_EMPLOYED),
    fill="steel blue",
    col= "yellow",
    alpha= .6, bins=10) +
  labs(title="Histograma: Antigüedad Laboral", x="Años", y="Cantidad Solicitudes")
```



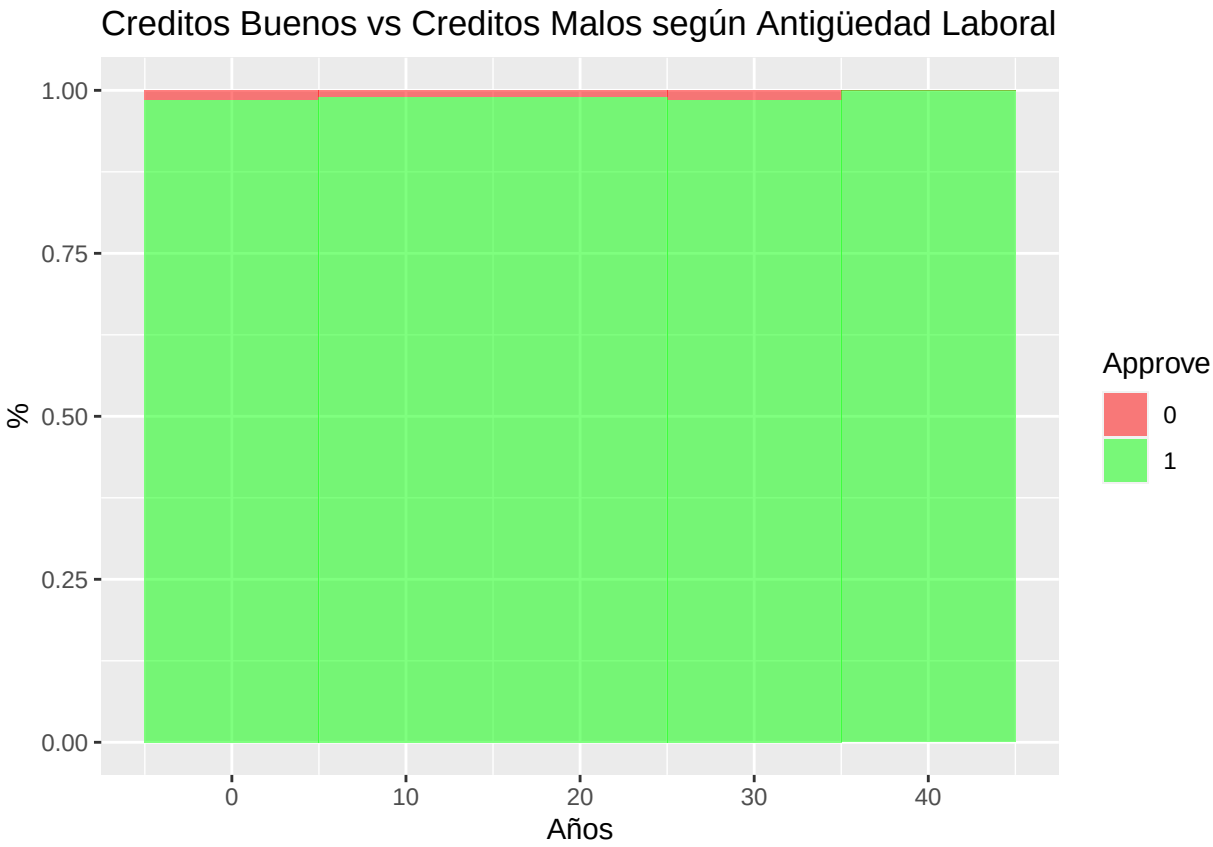
YEARS_EMPLOYED vs APPROVE Vamos a utilizar el gráfico BoxPlot para tratar de entender como se relacionan la variable YEARS_EMPLOYED y nuestra variable Objetivo.

```
ggplot(dfCreditCard) +
  geom_boxplot(aes(x=Approve, y=YEARS_EMPLOYED), outlier.color = "red", col="yellow", fill="steel blue"
  labs(title="BoxPlot: Calidad Crediticia según Antigüedad Laboral", x="Credito Malo (0) o Bueno (1)", y
```



Se observan algunas leves diferencias en este gráfico. Vamos a intentar representarlo de manera diferente.

```
ggplot(dfCreditCard) +
  geom_histogram(aes(x=YEARS_EMPLOYED, fill=Approve), position="fill", alpha=.5, binwidth = 10) +
  labs(title="Creditos Buenos vs Creditos Malos según Antigüedad Laboral", x="Años", y="%") +
  scale_fill_manual(values= c("0"= "red", "1"="green"))
```

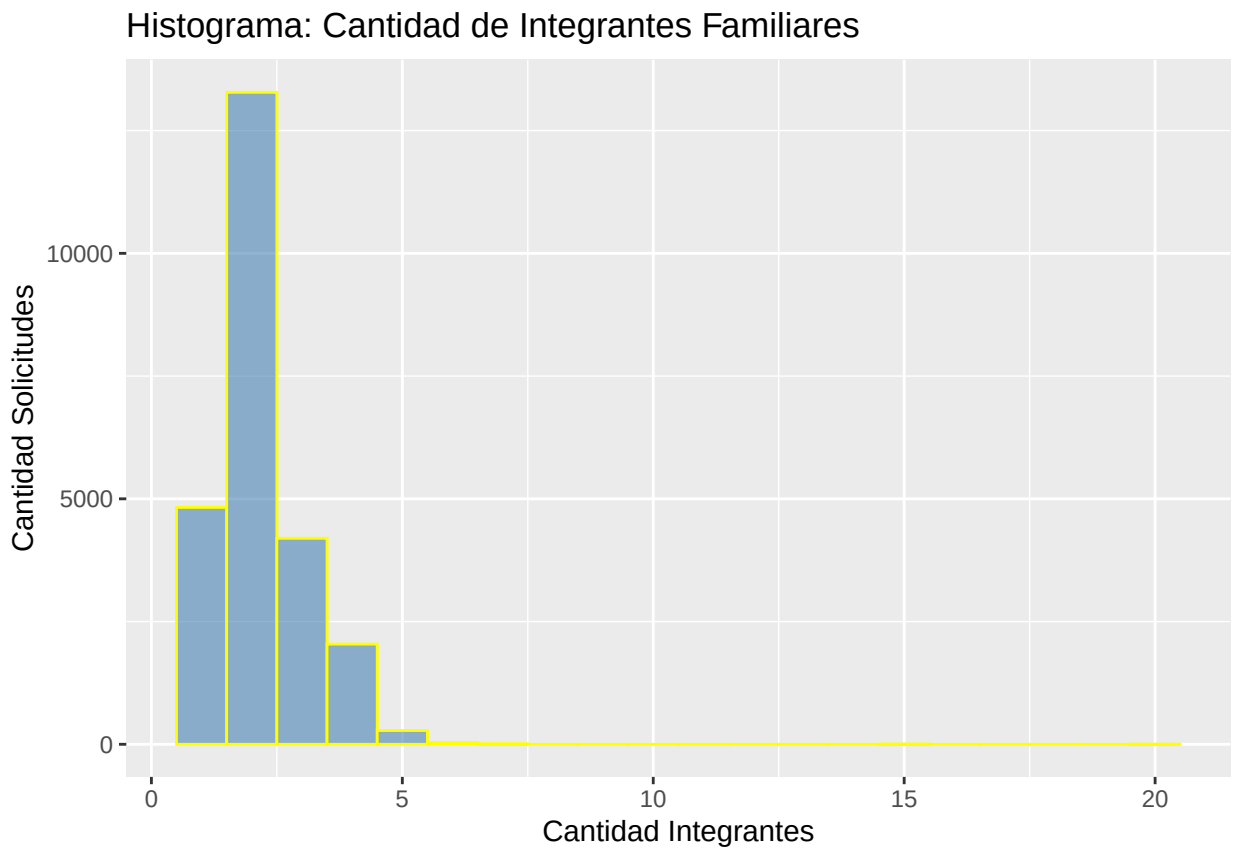


La leve diferencia notada en el gráfico anterior se mantiene, podría ser un posible predictor.

4.1.5. - Variable: CNT_FAM_MEMBERS

CNT_FAM_MEMBERS En relación a la variable *CNT_FAM_MEMBERS* observaremos la distribución de la cantidad total de integrantes de la familia, en cada barra del siguiente gráfico:

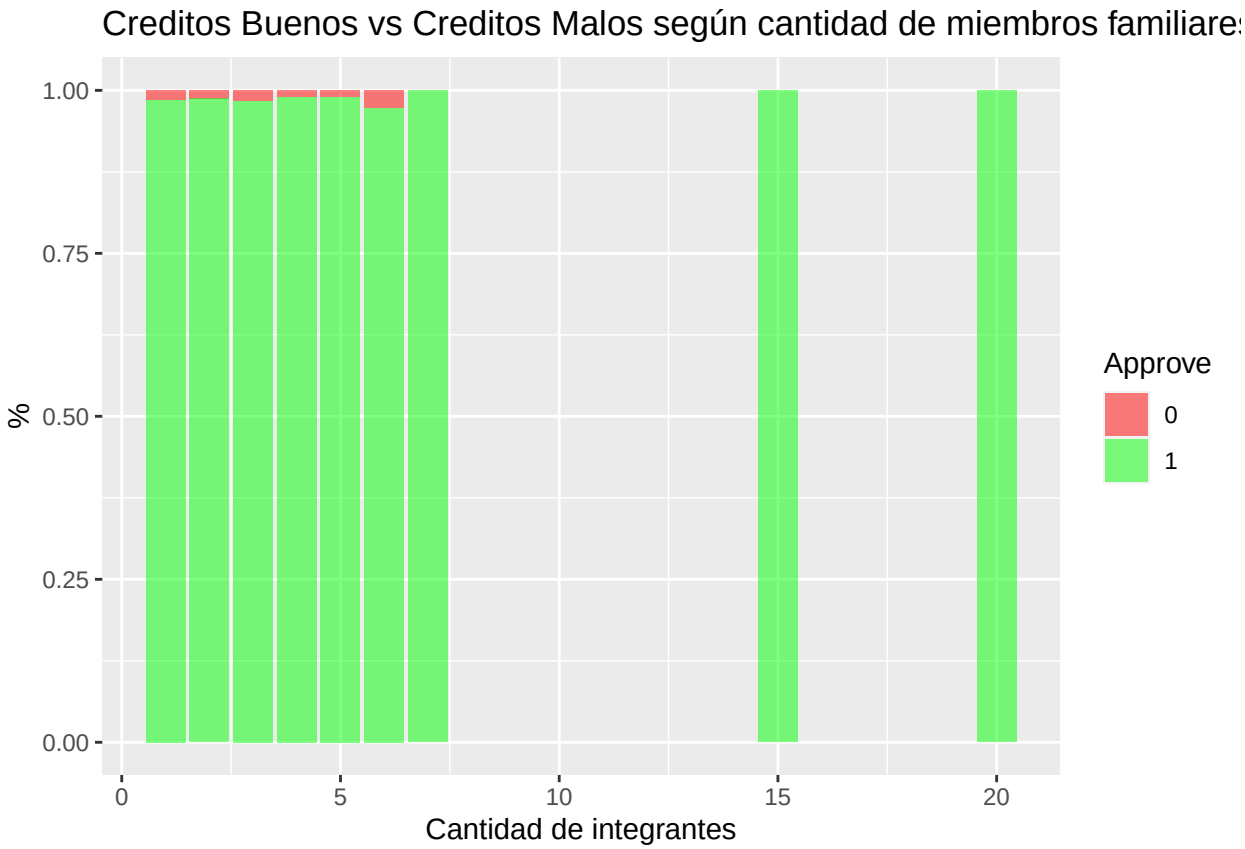
```
ggplot(dfCreditCard) +
  geom_histogram(
    aes(x=CNT_FAM_MEMBERS),
    fill="steel blue",
    col= "yellow",
    alpha= .6,
    binwidth = 1) +
  labs(title="Histograma: Cantidad de Integrantes Familiares", x="Cantidad Integrantes", y="Cantidad Sol
```



Como era de esperarse se observa una distribución similar a la cantidad de hijos, son variables claramente relacionadas.

CNT_FAM_MEMBERS vs APPROVE Es de esperarse que la relación entre estas dos variables sea similar a lo ya evaluado con cantidad de hijos, se utiliza el gráfico de barras para convalidarlo.

```
ggplot(dfCreditCard) +
  geom_bar(aes(x=CNT_FAM_MEMBERS, fill=Approve), position="fill", alpha=.5) +
  labs(title="Creditos Buenos vs Creditos Malos según cantidad de miembros familiares", x="Cantidad de
  scale_fill_manual(values= c("0"= "red", "1"="green"))
```

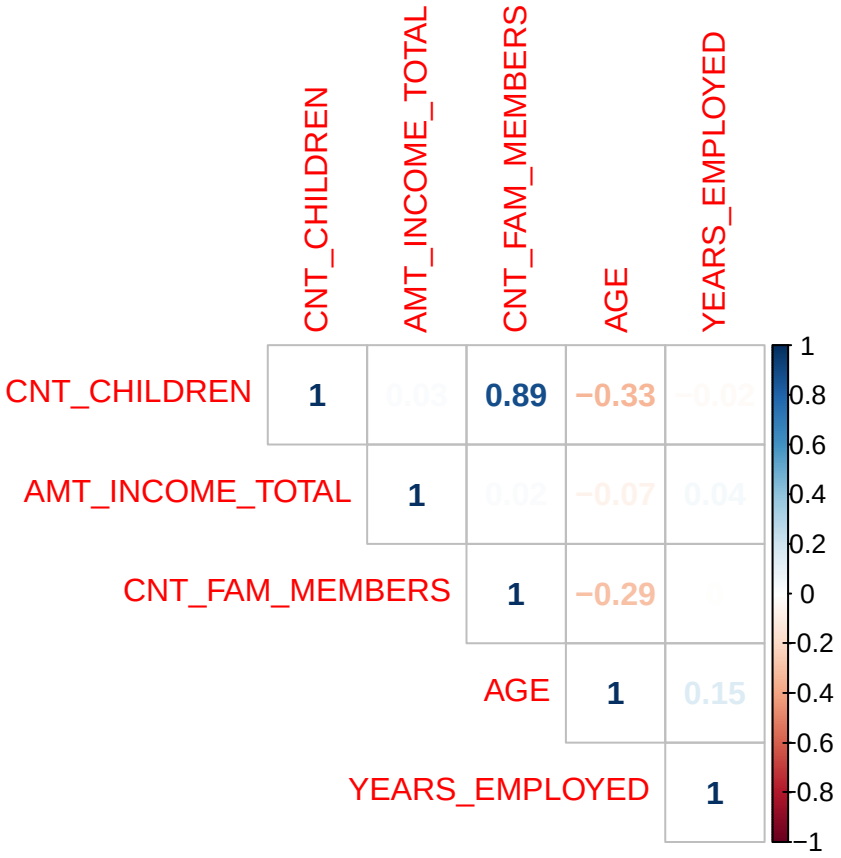


No se verifica una tendencia que justifique incluirlo como predictor.

4.2 - Correlación de variables numéricas

Mediante un gráfico de correlación de las variables numéricas vamos a analizar como se relacionan entre ellas.

```
dfcor <- select_if(dfCreditCard, is.numeric)
dfcor <- select(dfcor, -ID, -MONTHS_BALANCE, -DAYS_EMPLOYED, -DAYS_BIRTH)
cm <- cor(dfcor)
corrplot(cm, method = "number", type = "upper")
```



Observamos en el gráfico que Cantidad de Hijos y Cantidad de Miembros Familiares están muy relaionados, nos parece más amplio el concepto de Miembros Familiares si hubiera que seleccionarlo como predictor.

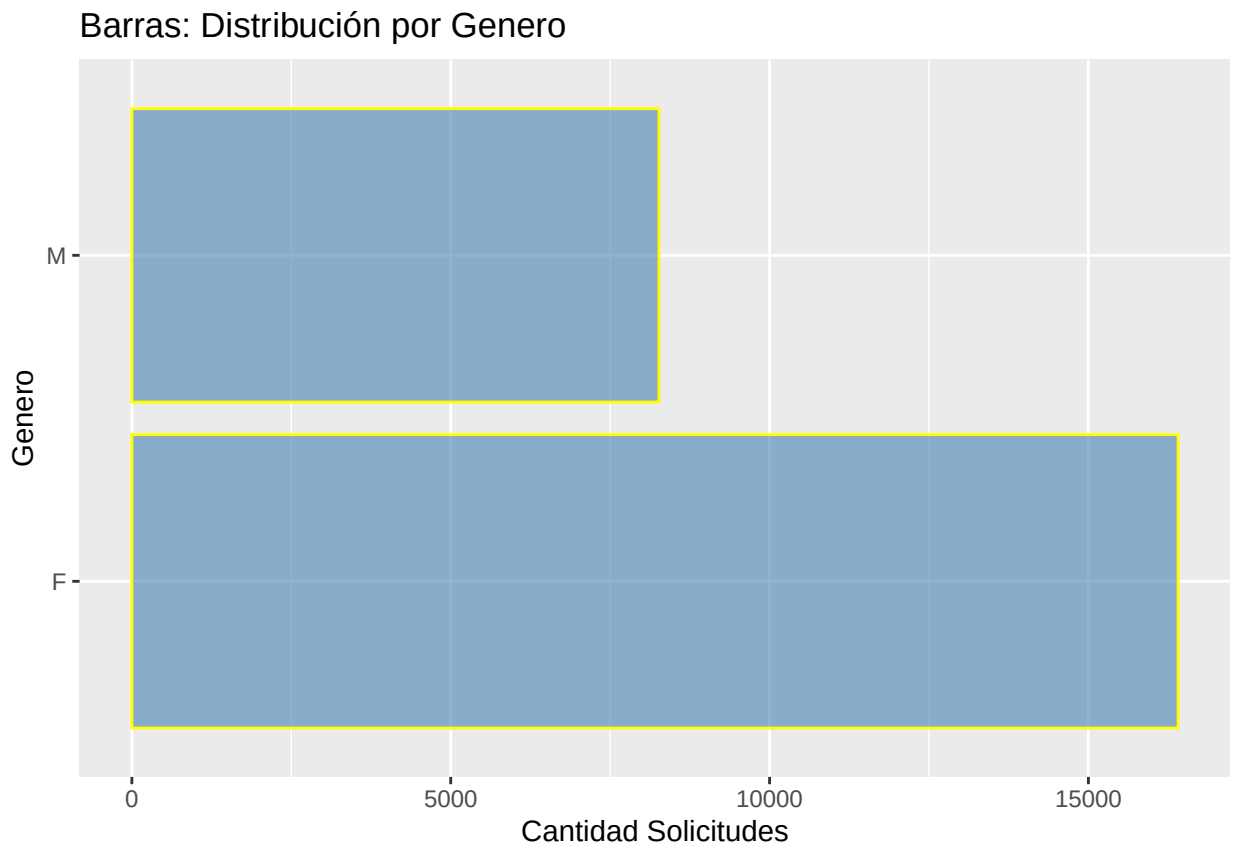
No observamos correlaciones significativas entre el resto de las variables.

4.3 - Análisis de variables categóricas

4.3.1 - Variable: CODE_GENDER

CODE_GENDER Para conocere la distribución de la muestra graficamos con barras las 2 categorías:

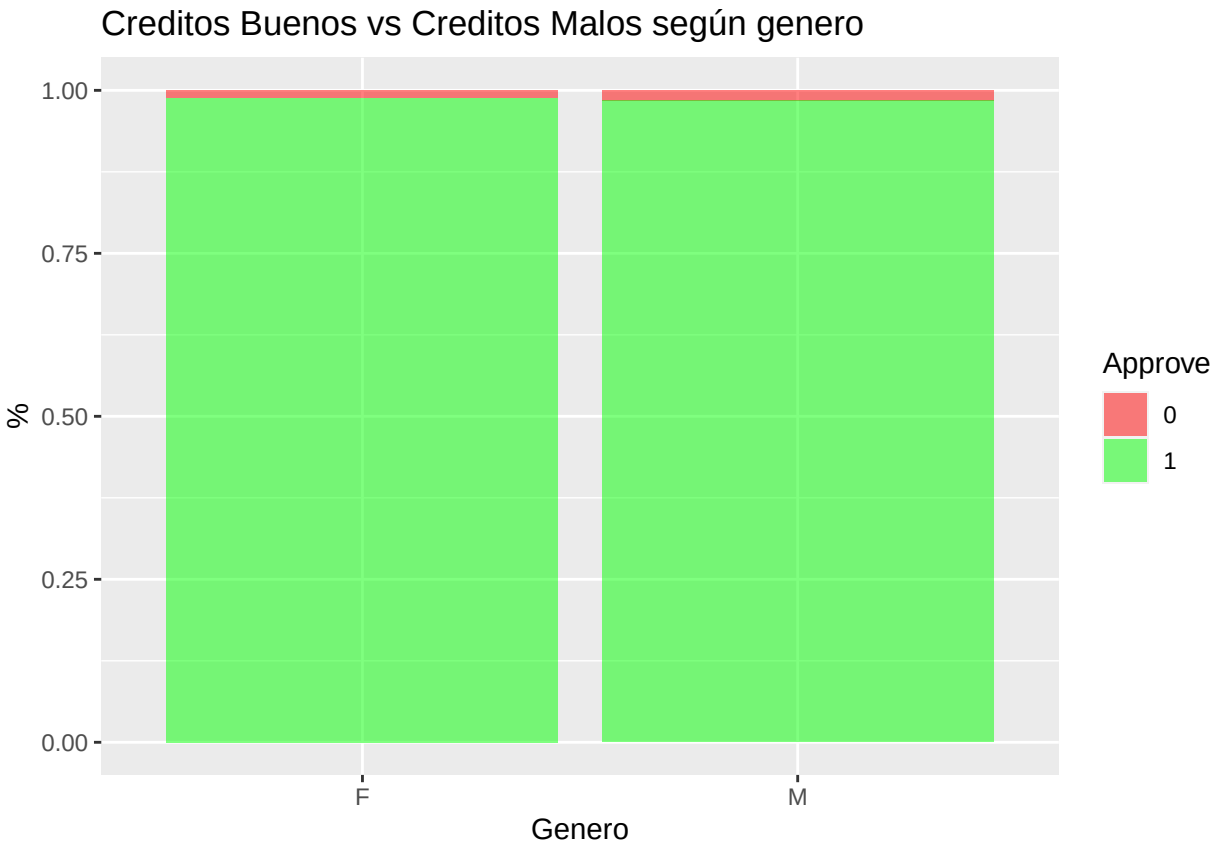
```
ggplot(dfCreditCard) +
  geom_bar(
    aes(x=CODE_GENDER),
    stat="count",
    fill="steel blue",
    col= "yellow",
    alpha= .6) +
    labs(title="Barras: Distribución por Genero", x="Genero", y="Cantidad Solicitudes") +
    coord_flip()
```



Se observa que la muestra contiene aproximadamente el doble de mujeres. Evaluar si es algo que podría sesgar el modelo.

CODE_GENDER vs APPROVE Verificamos si podría existir alguna relación entre ambas variables mediante un gráfico de barras apiladas.

```
ggplot(dfCreditCard) +
  geom_bar(aes(x=CODE_GENDER, fill=Approve), position="fill", alpha=.5) +
  labs(title="Creditos Buenos vs Creditos Malos según genero", x="Genero", y="%") +
  scale_fill_manual(values= c("0"= "red", "1"="green"))
```

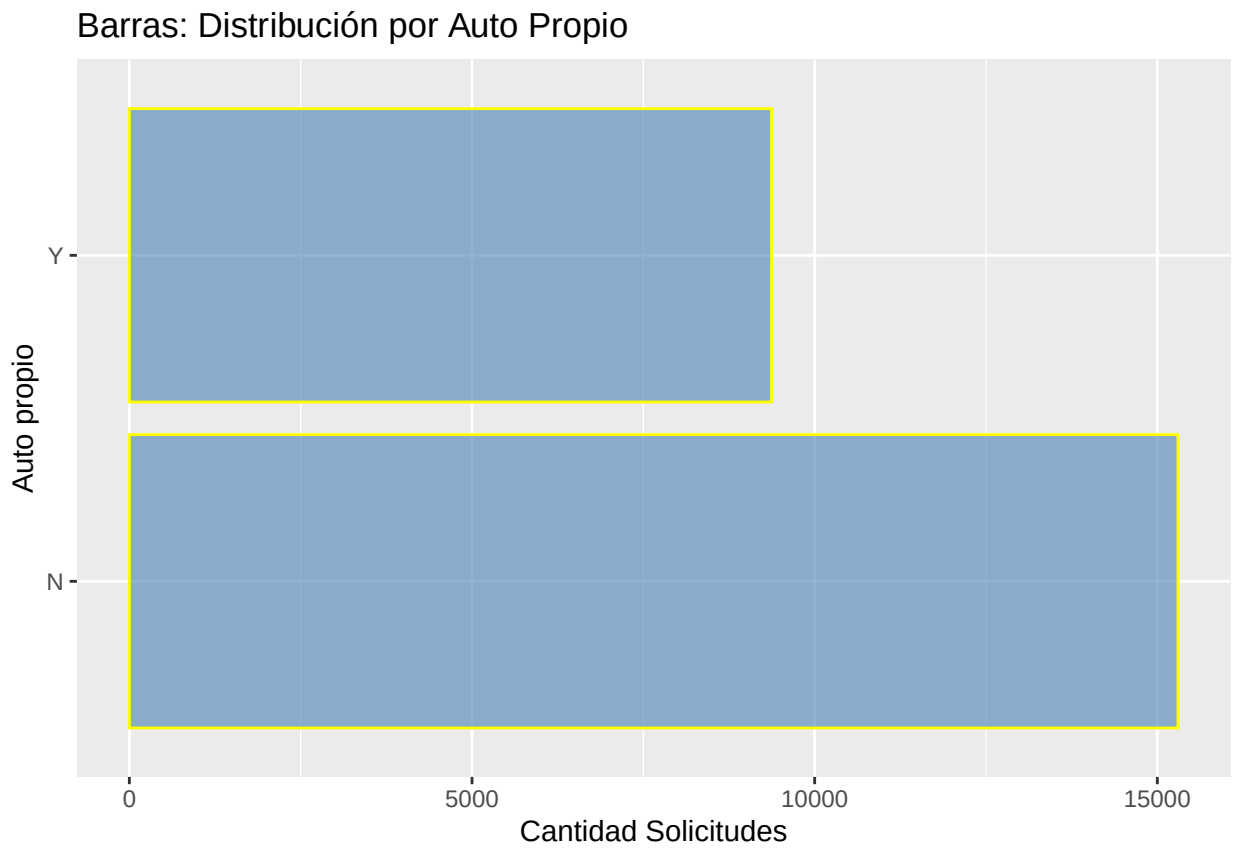


A pesar que la muestra contenía mayor cantidad de solicitudes del sexo femenino, la distribución respecto a la calidad crediticia es similar a las solicitudes presentadas por el sexo masculino. No parece existir una relación entre el genero y la calidad crediticia.

4.3.2 - Variable: FLAG_OWN_CAR

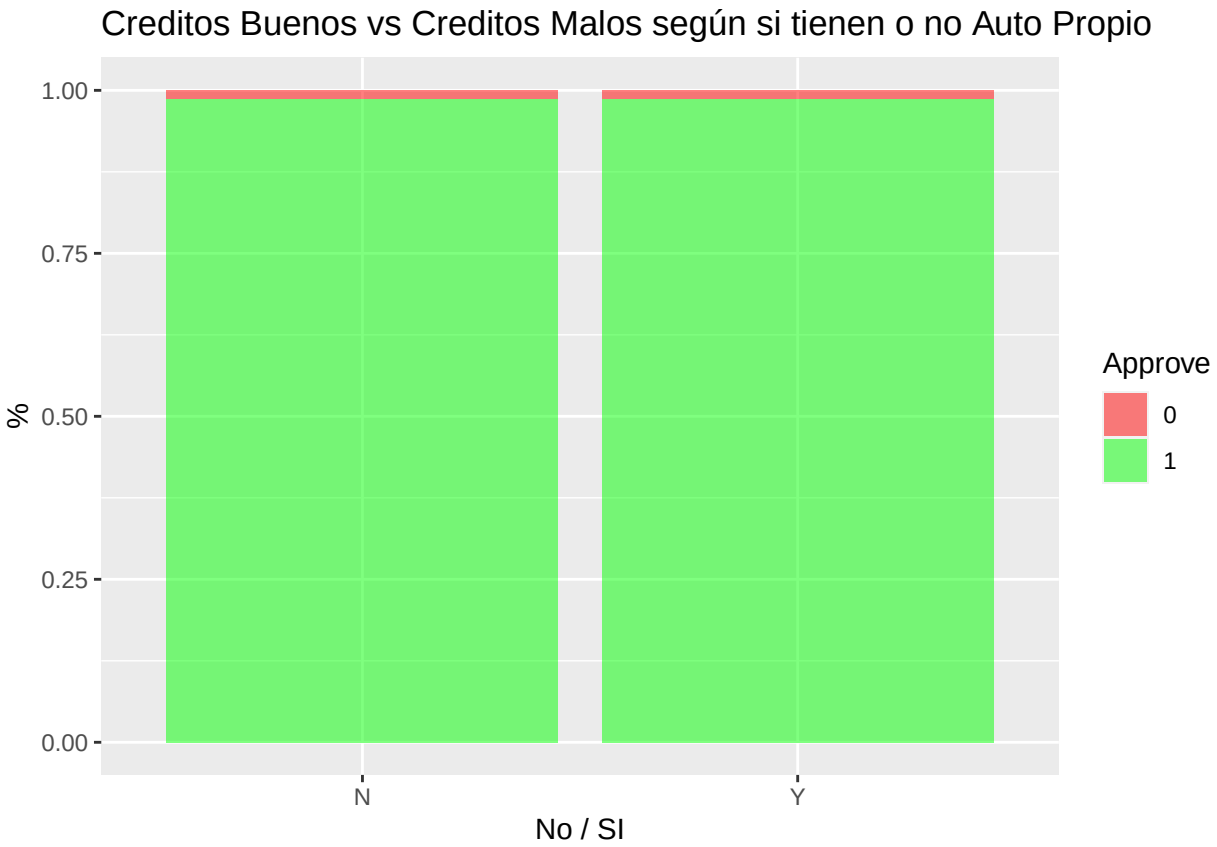
FLAG_OWN_CAR Para conocere la distribución de la muestra graficamos con barras las 2 categorías:

```
ggplot(dfCreditCard) +
  geom_bar(
    aes(x=FLAG_OWN_CAR),
    stat="count",
    fill="steel blue",
    col= "yellow",
    alpha= .6) +
    labs(title="Barras: Distribución por Auto Propio", x="Auto propio", y="Cantidad Solicitudes") +
    coord_flip()
```



FLAG_OWN_CAR vs APPROVE Verificamos si podría existir alguna relación entre ambas variables mediante un gráfico de barras apiladas.

```
ggplot(dfCreditCard) +
  geom_bar(aes(x=FLAG_OWN_CAR, fill=Approve), position="fill", alpha=.5) +
  labs(title="Creditos Buenos vs Creditos Malos según si tienen o no Auto Propio", x="No / SI", y="%") +
  scale_fill_manual(values= c("0"= "red", "1"="green"))
```

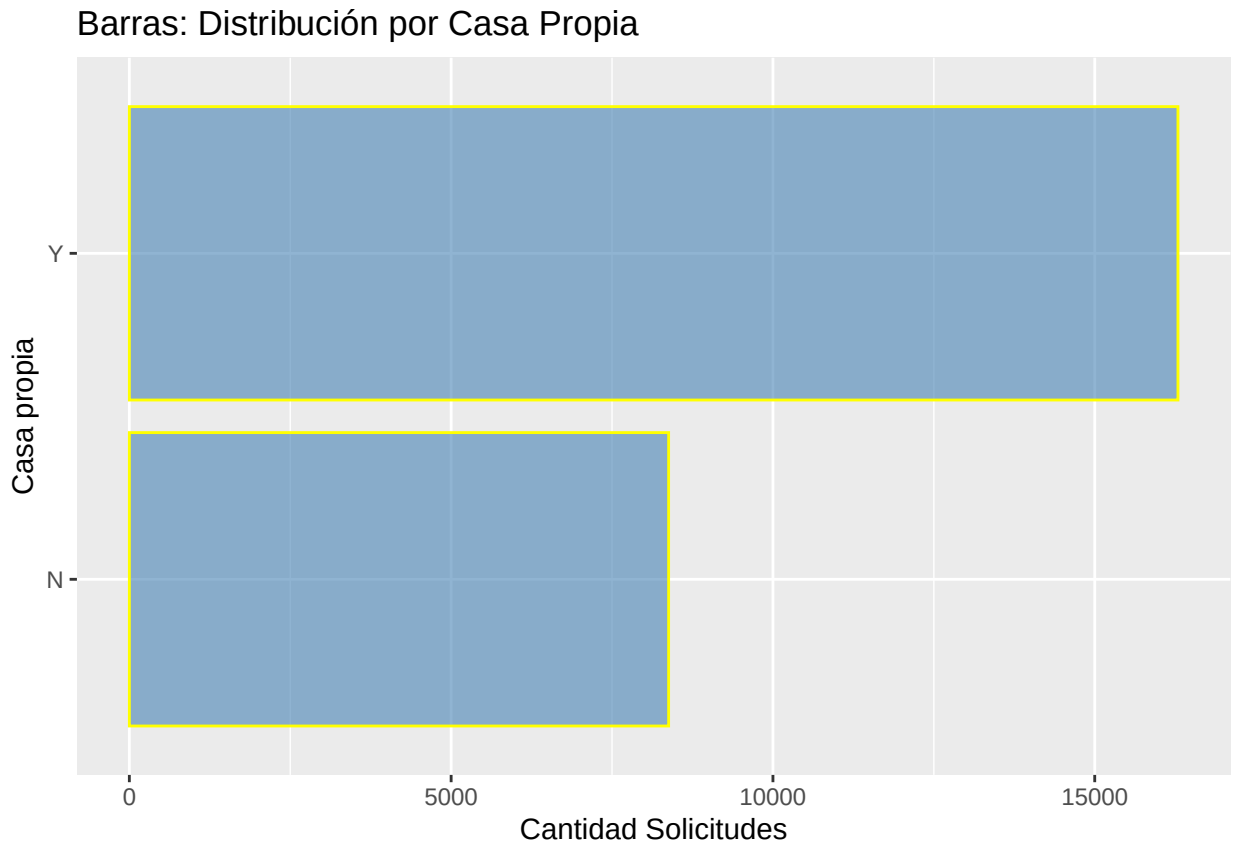


No parece existir una relación entre el Auto Propio y la calidad crediticia.

4.3.3 - Variable: FLAG_OWN_REALTY

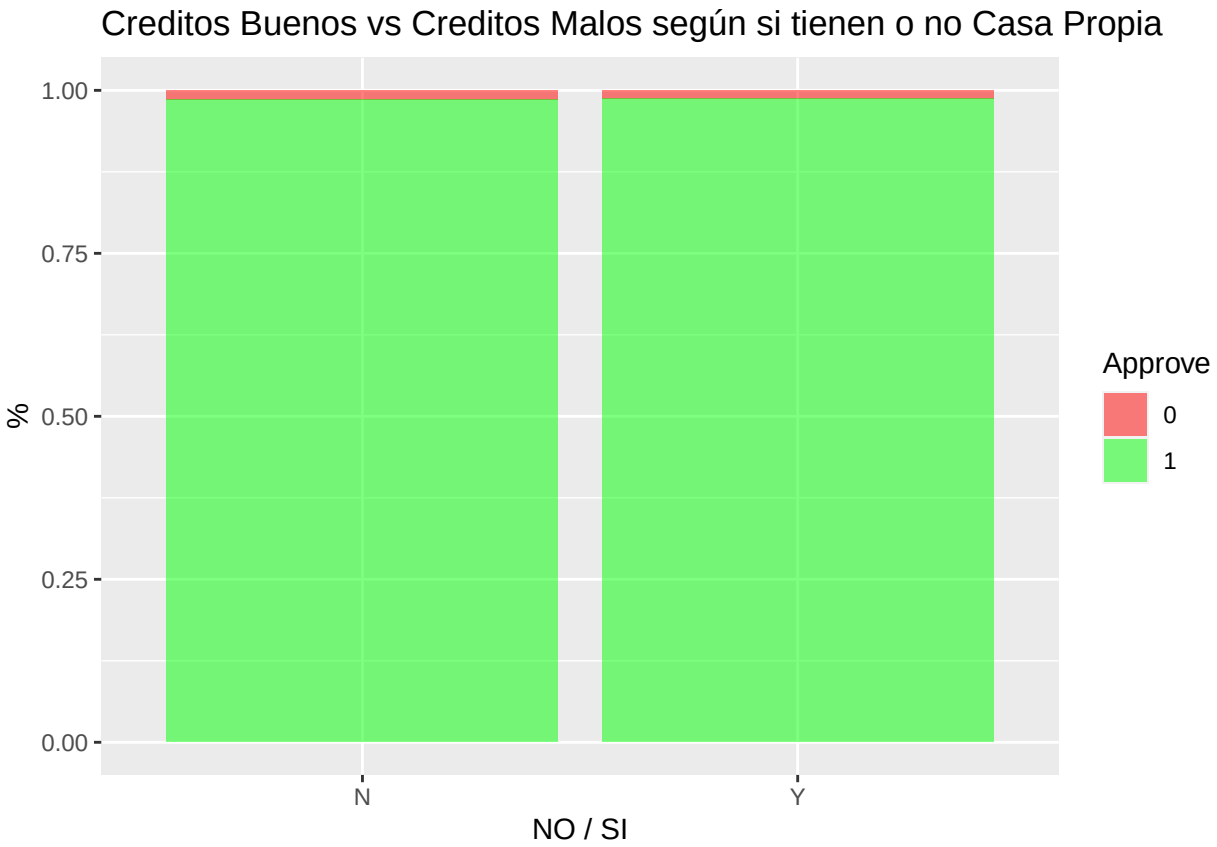
FLAG_OWN_REALTY Para conocere la distribución de la muestra graficamos con barras las 2 categorías:

```
ggplot(dfCreditCard) +
  geom_bar(
    aes(x=FLAG_OWN_REALTY),
    stat="count",
    fill="steel blue",
    col= "yellow",
    alpha= .6) +
    labs(title="Barras: Distribución por Casa Propia", x="Casa propia", y="Cantidad Solicitudes") +
    coord_flip()
```



FLAG_OWN_REALTY vs APPROVE Verificamos si podría existir alguna relación entre ambas variables mediante un gráfico de barras apiladas.

```
ggplot(dfCreditCard) +
  geom_bar(aes(x=FLAG_OWN_REALTY, fill=Approve), position="fill", alpha=.5) +
  labs(title="Creditos Buenos vs Creditos Malos según si tienen o no Casa Propia", x="NO / SI", y="%") +
  scale_fill_manual(values= c("0"= "red", "1"="green"))
```

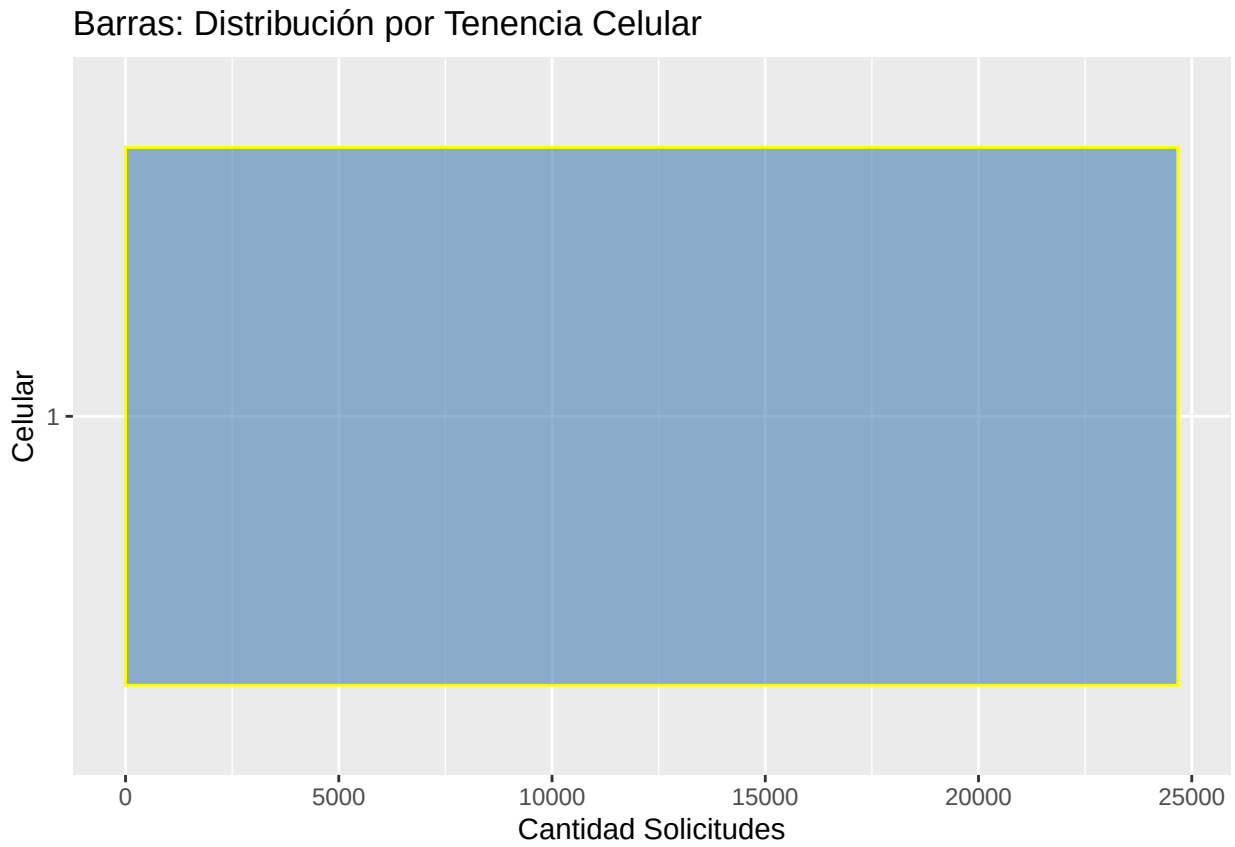


Parecería existir alguna tendencia a favor de los que poseen casa propia, podría ser un posible predictor.

4.3.4 - Variable: FLAG_MOBIL

FLAG_MOBIL Para conocere la distribución de la muestra graficamos con barras las 2 categorías:

```
ggplot(dfCreditCard) +
  geom_bar(
    aes(x=FLAG_MOBIL),
    stat="count",
    fill="steel blue",
    col= "yellow",
    alpha= .6) +
    labs(title="Barras: Distribución por Tenencia Celular", x="Celular", y="Cantidad Solicitudes") +
    coord_flip()
```

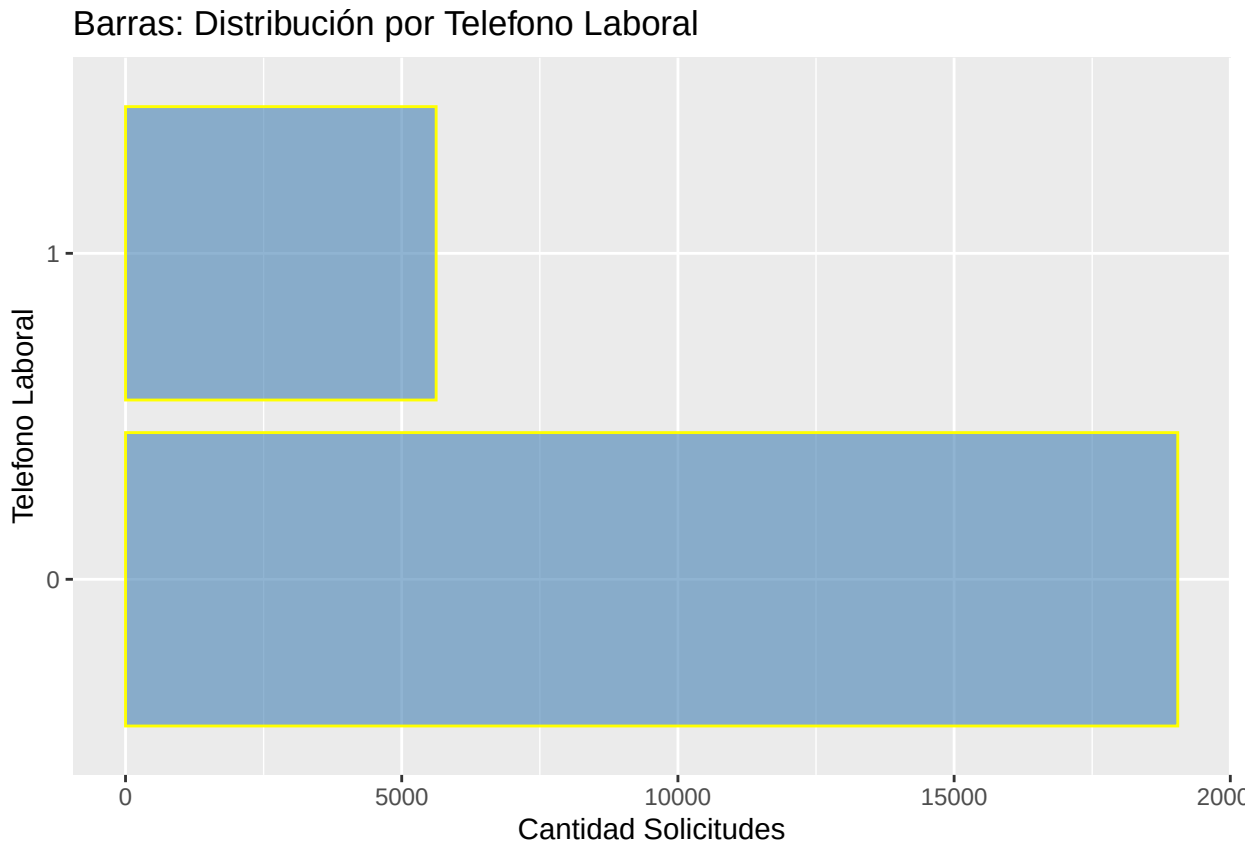


Como ya habíamos mencionado anteriormente, se observa que todos las solicitudes tienen Teléfono Celular, no nos sirve para la predicción.

4.3.5 - Variable: FLAG_WORK_PHONE

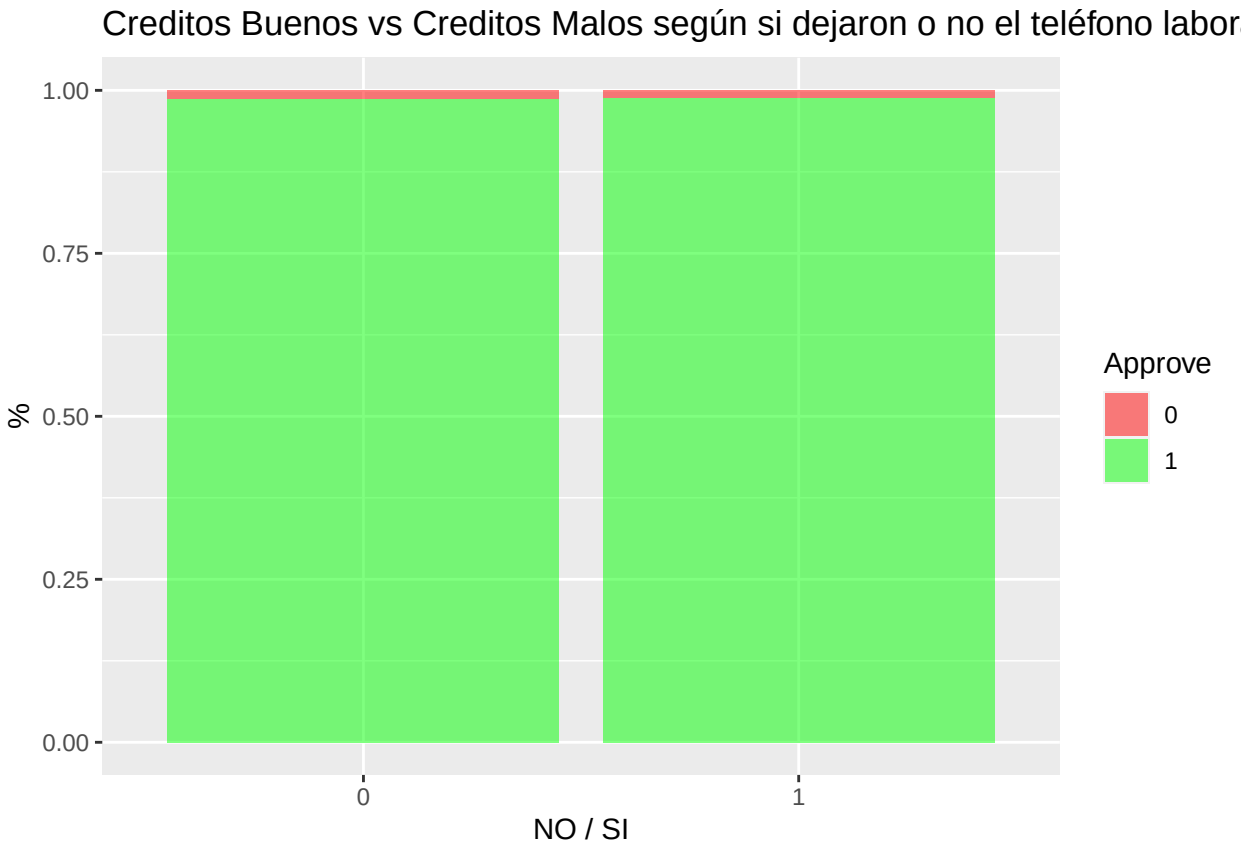
FLAG_WORK_PHONE Para conocere la distribución de la muestra graficamos con barras las 2 categorías:

```
ggplot(dfCreditCard) +
  geom_bar(
    aes(x=FLAG_WORK_PHONE),
    stat="count",
    fill="steel blue",
    col= "yellow",
    alpha= .6) +
    labs(title="Barras: Distribución por Telefono Laboral", x="Telefono Laboral", y="Cantidad Solicitud
    coord_flip()
```



FLAG_WORK_PHONE vs APPROVE Verificamos si podría existir alguna relación entre ambas variables mediante un gráfico de barras apiladas.

```
ggplot(dfCreditCard) +
  geom_bar(aes(x=FLAG_WORK_PHONE, fill=Approve), position="fill", alpha=.5) +
  labs(title="Creditos Buenos vs Creditos Malos según si dejaron o no el teléfono laboral", x="NO / SI"
  scale_fill_manual(values= c("0"= "red", "1"="green"))
```

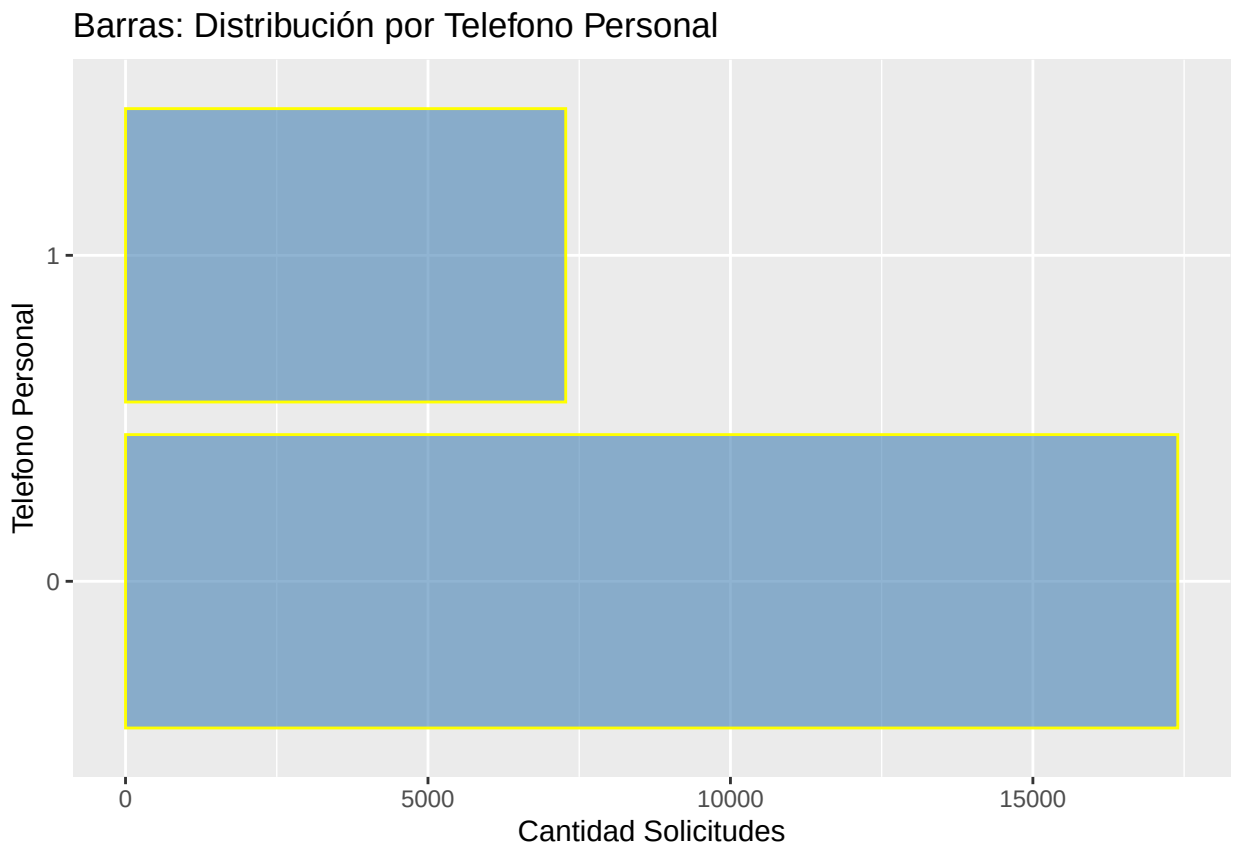


Parece existir una ligera diferencia, podría ser un posible predictor.

4.3.6 - Variable: FLAG_PHONE

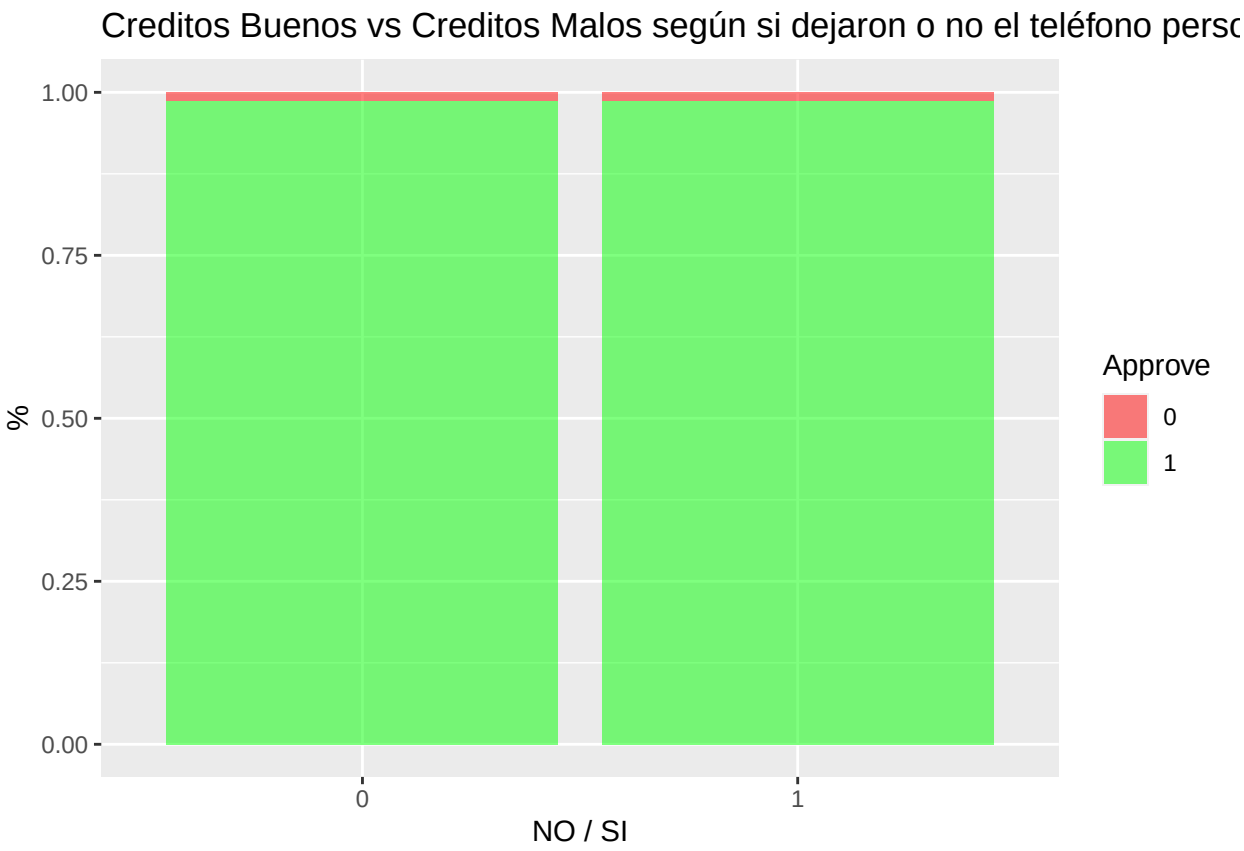
FLAG_PHONE Para conocere la distribución de la muestra graficamos con barras las 2 categorías:

```
ggplot(dfCreditCard) +
  geom_bar(
    aes(x=FLAG_PHONE),
    stat="count",
    fill="steel blue",
    col= "yellow",
    alpha= .6) +
    labs(title="Barras: Distribución por Telefono Personal", x="Telefono Personal", y="Cantidad Solicitu
    coord_flip()
```



FLAG_PHONE vs APPROVE Verificamos si podría existir alguna relación entre ambas variables mediante un gráfico de barras apiladas.

```
ggplot(dfCreditCard) +
  geom_bar(aes(x=FLAG_PHONE, fill=Approve), position="fill", alpha=.5) +
  labs(title="Creditos Buenos vs Creditos Malos según si dejaron o no el teléfono personal", x="NO / SI
  scale_fill_manual(values= c("0"= "red", "1"="green"))
```

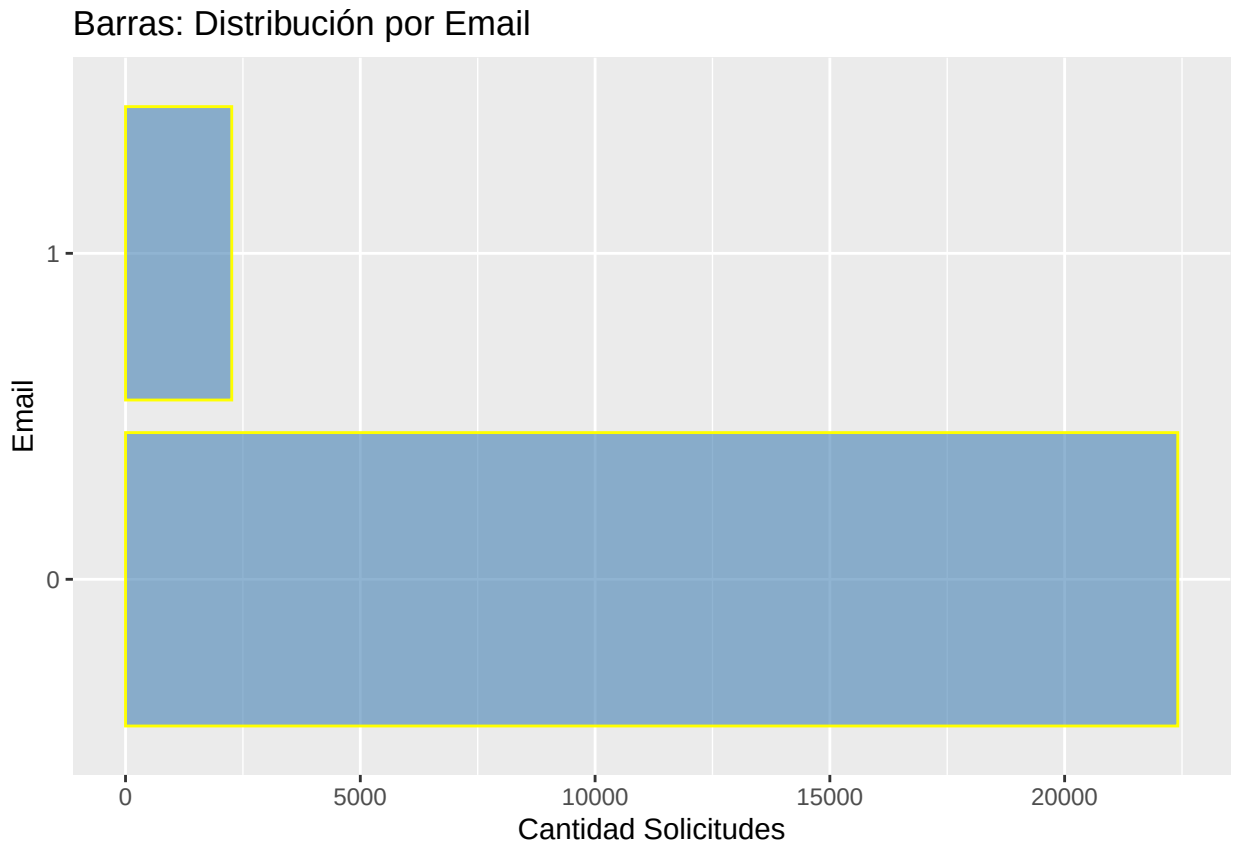


No parece existir una relación entre el teléfono personal y la calidad crediticia.

4.3.7 - Variable: FLAG_EMAIL

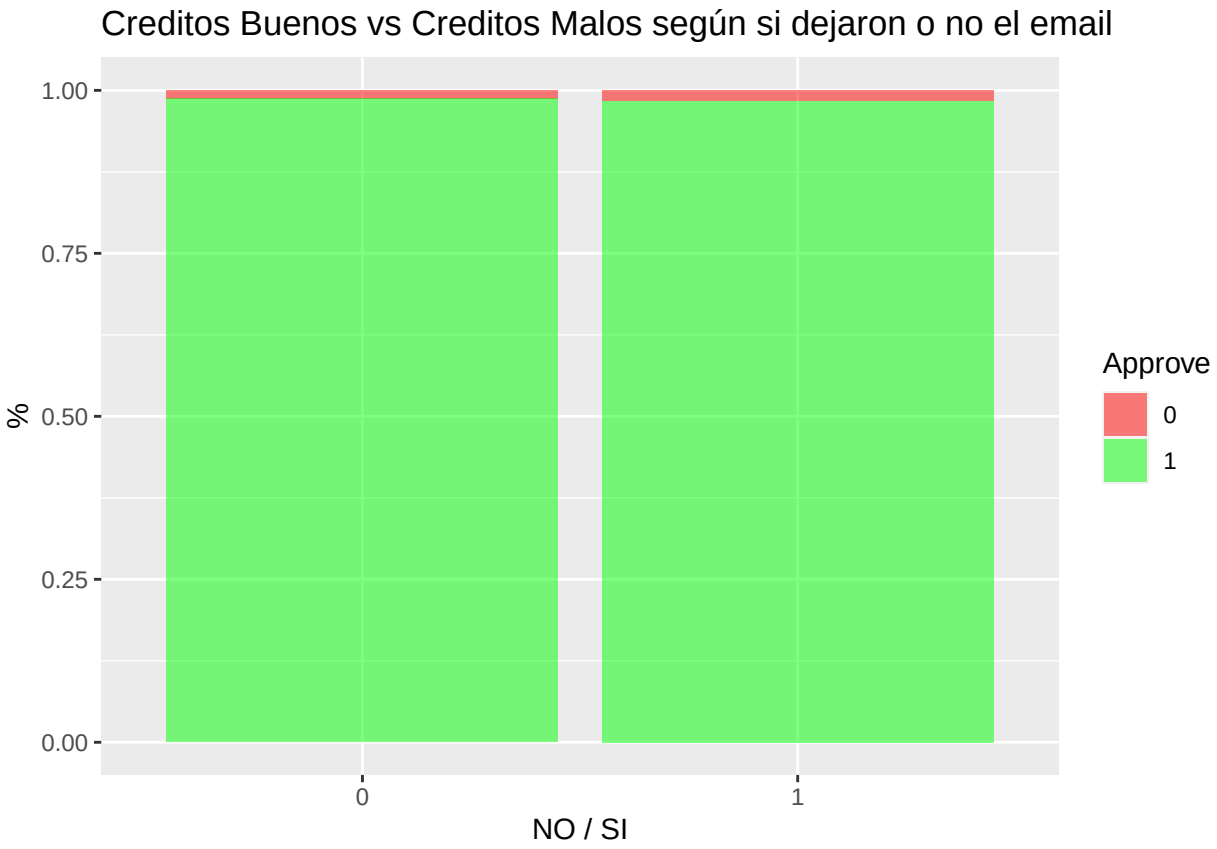
FLAG_EMAIL Para conocere la distribución de la muestra graficamos con barras las 2 categorías:

```
ggplot(dfCreditCard) +
  geom_bar(
    aes(x=FLAG_EMAIL),
    stat="count",
    fill="steel blue",
    col= "yellow",
    alpha= .6) +
    labs(title="Barras: Distribución por Email", x="Email", y="Cantidad Solicitudes") +
    coord_flip()
```



FLAG_EMAIL vs APPROVE Verificamos si podría existir alguna relación entre ambas variables mediante un gráfico de barras apiladas.

```
ggplot(dfCreditCard) +
  geom_bar(aes(x=FLAG_EMAIL, fill=Approve), position="fill", alpha=.5) +
  labs(title="Creditos Buenos vs Creditos Malos según si dejaron o no el email", x="NO / SI", y="%") +
  scale_fill_manual(values= c("0"= "red", "1"="green"))
```



Parece existir una muy ligera diferencia, podría ser un posible predictor.

4.3.8 - Variable: NAME_INCOME_TYPE

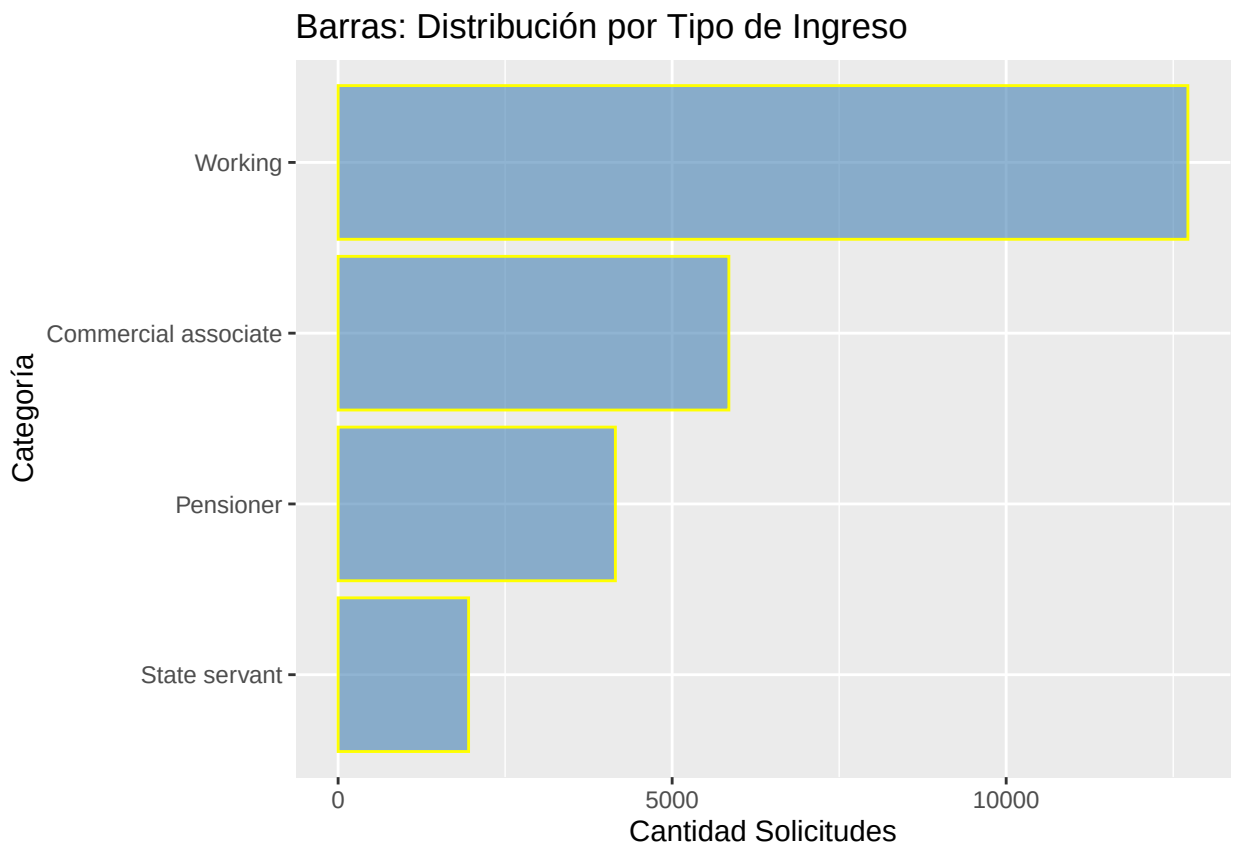
NAME_INCOME_TYPE En esta variable se observó una categoría *student* muy poco representada, sin casos negativos, que va a generar ruido en el modelo predictivo. A los fines del análisis se decidió reemplazarlos por la categoría mas representativa *working*

```
dfCreditCard$NAME_INCOME_TYPE[dfCreditCard$NAME_INCOME_TYPE == "Student"] <- "Working"
dfCreditCard$NAME_INCOME_TYPE <- factor(dfCreditCard$NAME_INCOME_TYPE)
```

Para conocere la distribución de la muestra graficamos con barras diferentes categorías:

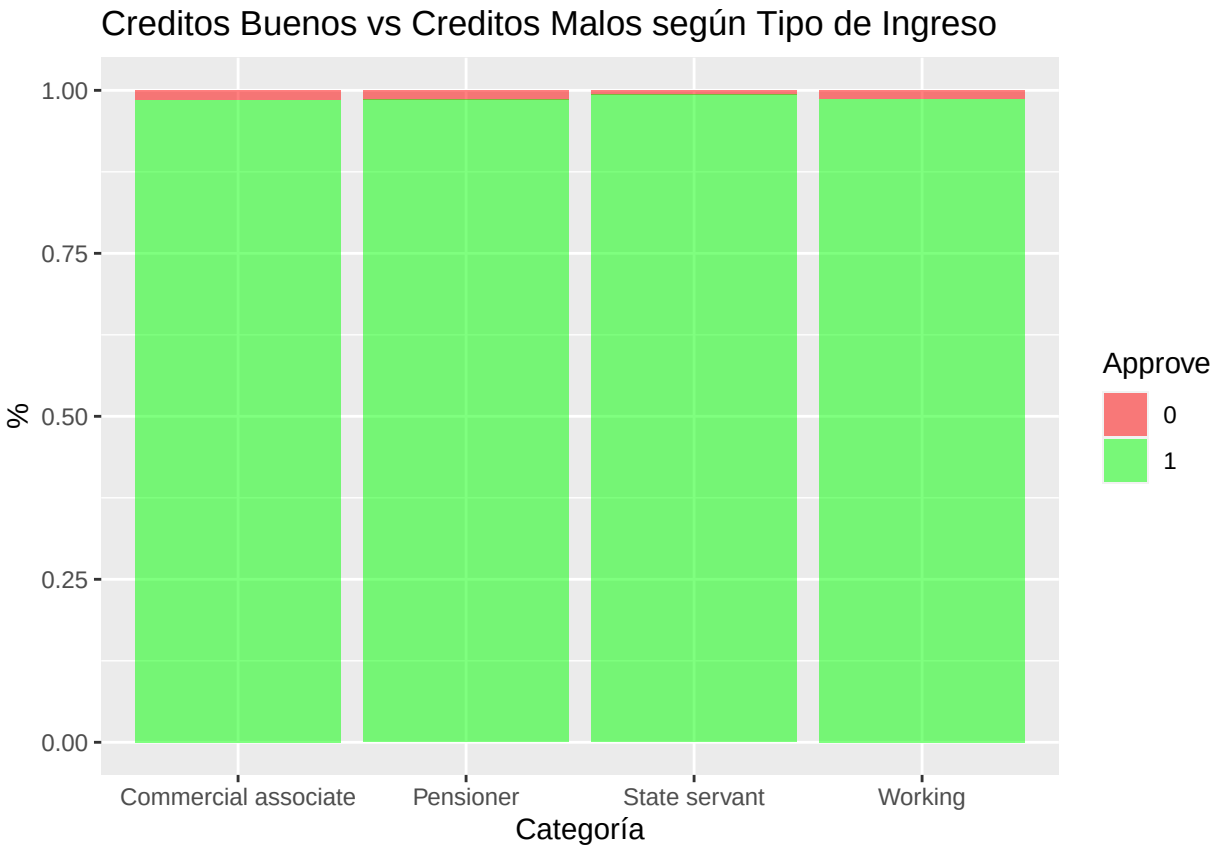
```
catIncomeType <- table(dfCreditCard$NAME_INCOME_TYPE)
dfCatIncomeType <- as.data.frame(catIncomeType)
names(dfCatIncomeType) <- c("Categoria","Cantidad")

# Ordenamos
dfCatIncomeType <- transform(dfCatIncomeType, Categoria=reorder(Categoria, Cantidad))

ggplot(dfCatIncomeType) +
  geom_bar(
    aes(x=Categoria, y=Cantidad),
    stat="identity",
    fill="steel blue",
    col= "yellow",
    alpha= .6) +
```

```
    labs(title="Barras: Distribución por Tipo de Ingreso", x="Categoría", y="Cantidad Solicitudes") +
    coord_flip()
```



NAME_INCOME_TYPE vs APPROVE Verificamos si podría existir alguna relación entre ambas variables mediante un gráfico de barras apiladas.

```
ggplot(dfCreditCard) +
  geom_bar(aes(x=NAME_INCOME_TYPE, fill=Approve), position="fill", alpha=.5) +
  labs(title="Creditos Buenos vs Creditos Malos según Tipo de Ingreso", x="Categoría", y="%") +
  scale_fill_manual(values= c("0"= "red", "1"="green"))
```

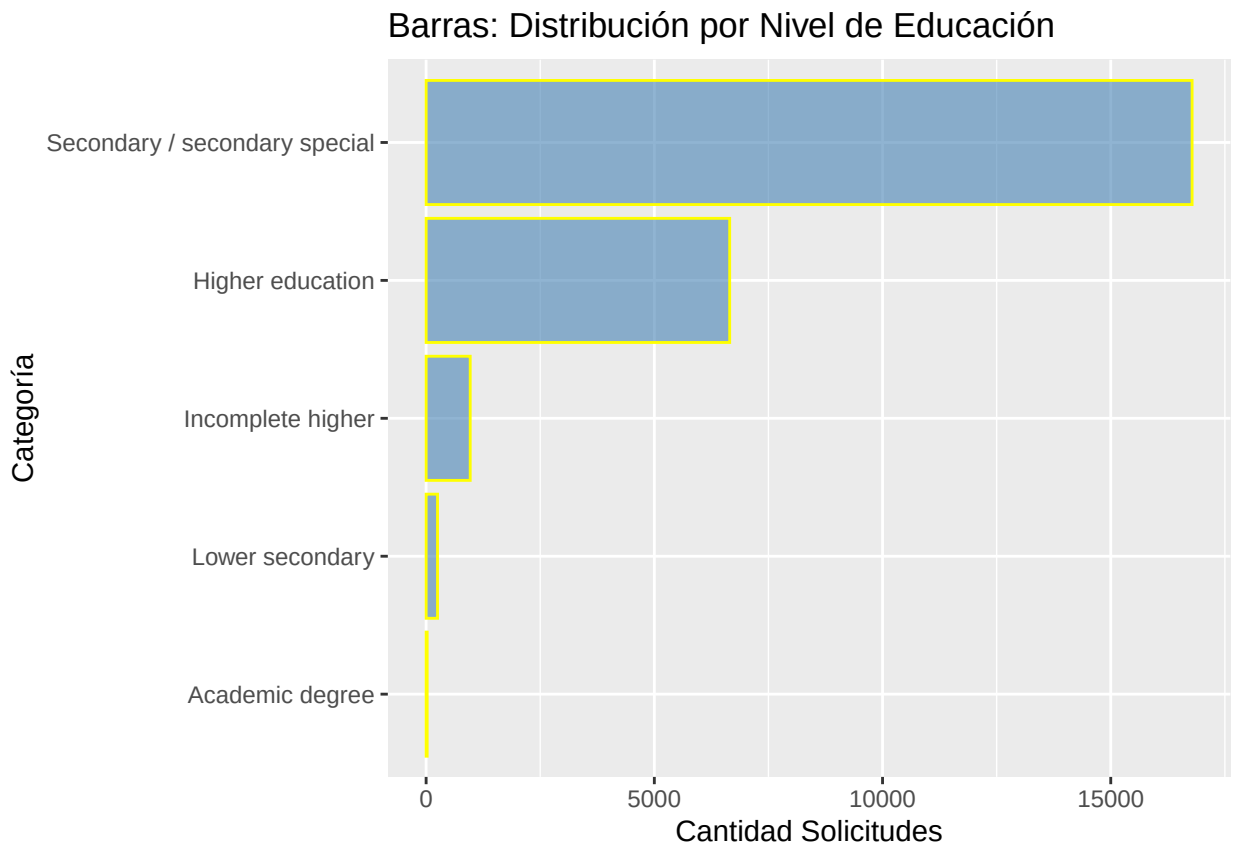


Podría existir alguna relación entre las variables, aunque la diferencia observada en la categoría estudiantes debe ser tenida en cuenta en el contexto observado en el gráfico anterior, donde vemos que las solicitudes de estudiantes son insignificantes en la muestra. Podría ser un posible predictor.

4.3.9 - Variable: NAME_EDUCATION_TYPE

NAME_EDUCATION_TYPE Para conocere la distribución de la muestra graficamos con barras diferentes categorías:

```
catIncomeType <- table(dfCreditCard$NAME_EDUCATION_TYPE)
dfCatIncomeType <- as.data.frame(catIncomeType)
names(dfCatIncomeType) <- c("Categoria","Cantidad")
# Ordenamos
dfCatIncomeType <- transform(dfCatIncomeType, Categoria=reorder(Categoria, Cantidad))
ggplot(dfCatIncomeType) +
  geom_bar(
    aes(x=Categoria, y=Cantidad),
    stat="identity",
    fill="steel blue",
    col= "yellow",
    alpha= .6) +
    labs(title="Barras: Distribución por Nivel de Educación", x="Categoría", y="Cantidad Solicitudes") +
    coord_flip()
```

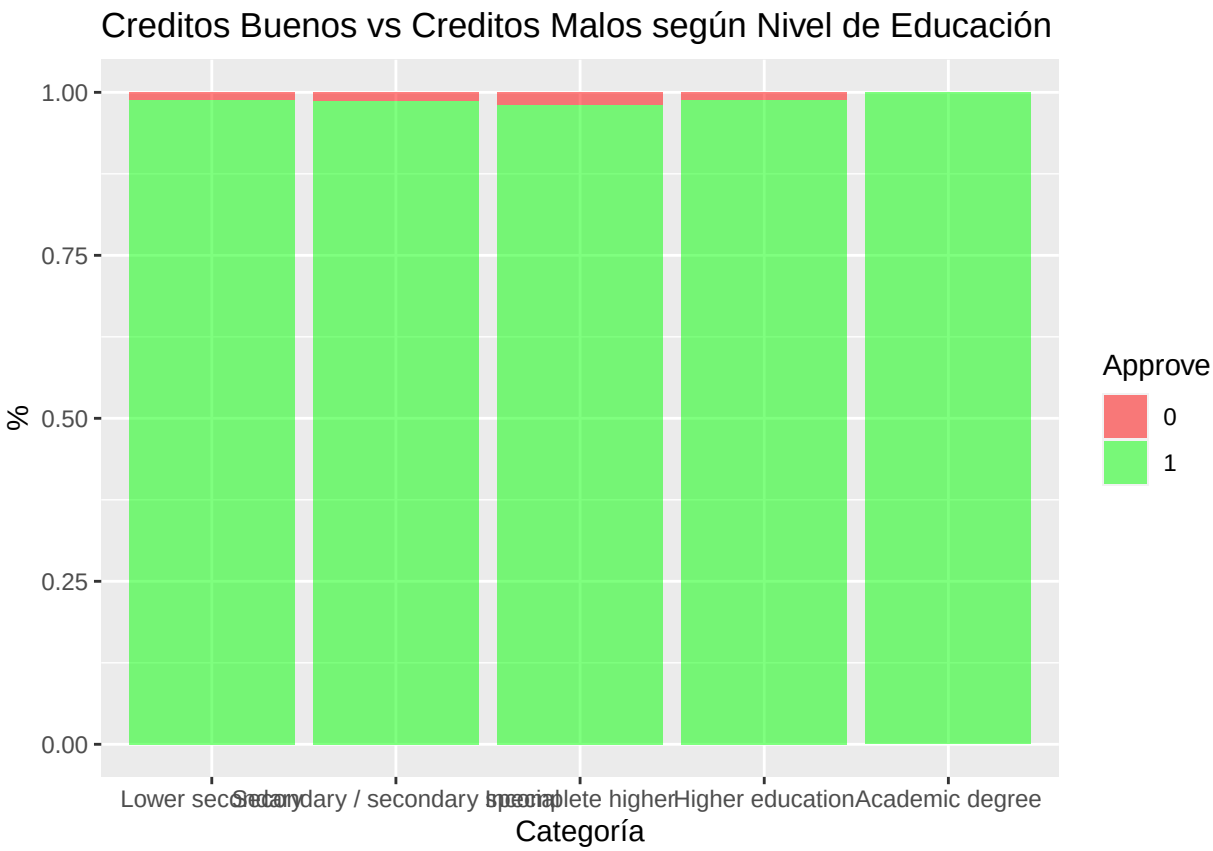


NAME_EDUCATION_TYPE vs APPROVE Verificamos si podría existir alguna relación entre ambas variables mediante un gráfico de barras apiladas.

```
print(levels(dfCreditCard$NAME_EDUCATION_TYPE))
```

```
## [1] "Academic degree"               "Higher education"
## [3] "Incomplete higher"             "Lower secondary"
## [5] "Secondary / secondary special"
```

```
dfCreditCard$NAME_EDUCATION_TYPE <- factor(dfCreditCard$NAME_EDUCATION_TYPE, levels = c("Lower secondary
ggplot(dfCreditCard) +
  geom_bar(aes(x=NAME_EDUCATION_TYPE, fill=Approve), position="fill", alpha=.5) +
  labs(title="Creditos Buenos vs Creditos Malos según Nivel de Educación", x="Categoría", y="%") +
  scale_fill_manual(values= c("0"= "red", "1"="green"))
```

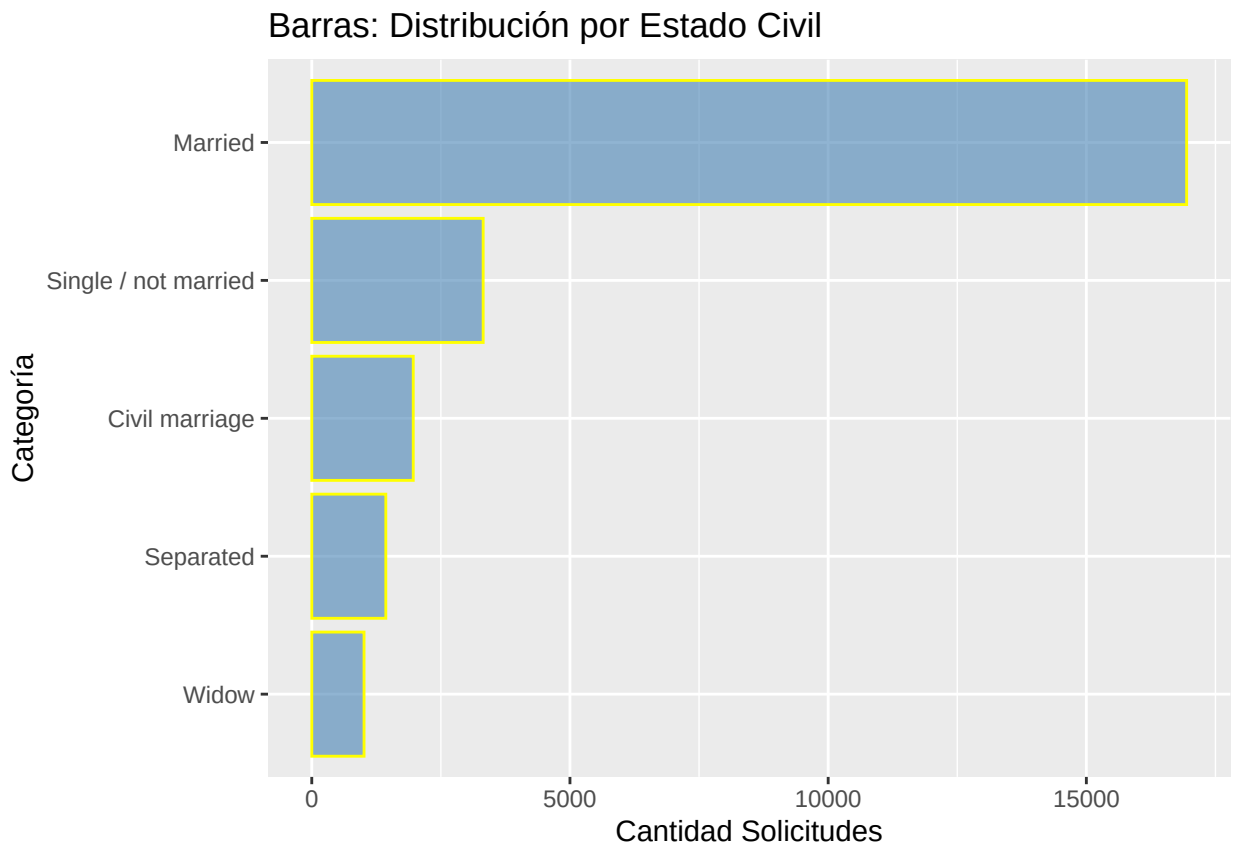


Visualizamos que podría existir una relación entre el nivel de educación y la calidad crediticia. Podría ser un posible predictor.

4.3.10 - Variable: NAME_FAMILY_STATUS

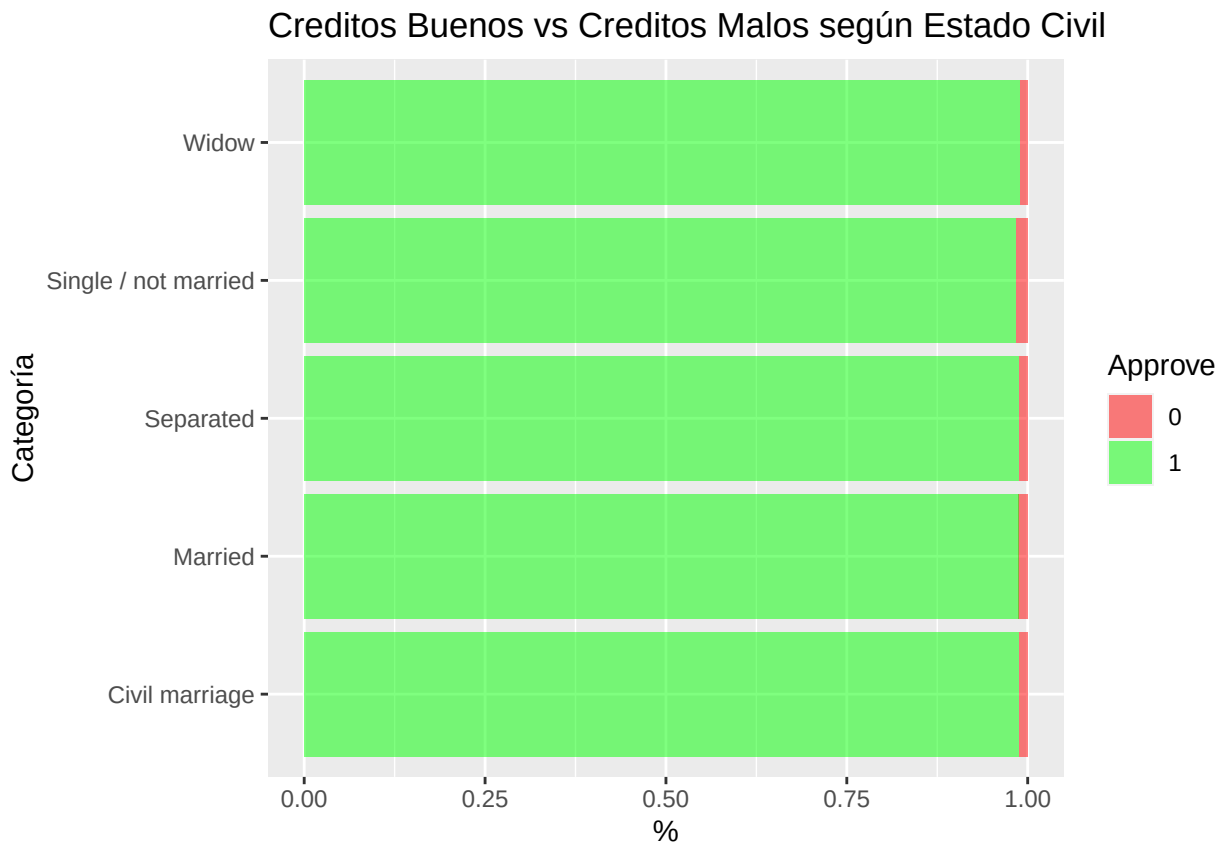
NAME_FAMILY_STATUS Para conocere la distribución de la muestra graficamos con barras diferentes categorías:

```
catIncomeType <- table(dfCreditCard$NAME_FAMILY_STATUS)
dfCatIncomeType <- as.data.frame(catIncomeType)
names(dfCatIncomeType) <- c("Categoria","Cantidad")
# Ordenamos
dfCatIncomeType <- transform(dfCatIncomeType, Categoria=reorder(Categoria, Cantidad))
ggplot(dfCatIncomeType) +
  geom_bar(
    aes(x=Categoria, y=Cantidad),
    stat="identity",
    fill="steel blue",
    col= "yellow",
    alpha= .6) +
    labs(title="Barras: Distribución por Estado Civil", x="Categoría", y="Cantidad Solicitudes") +
    coord_flip()
```



NAME_FAMILY_STATUS vs APPROVE Verificamos si podría existir alguna relación entre ambas variables mediante un gráfico de barras apiladas.

```
ggplot(dfCreditCard) +
  geom_bar(aes(x=NAME_FAMILY_STATUS, fill=Approve), position="fill", alpha=.5) +
  labs(title="Creditos Buenos vs Creditos Malos según Estado Civil", x="Categoría", y="%") +
  scale_fill_manual(values= c("0"= "red", "1"="green")) +
    coord_flip()
```

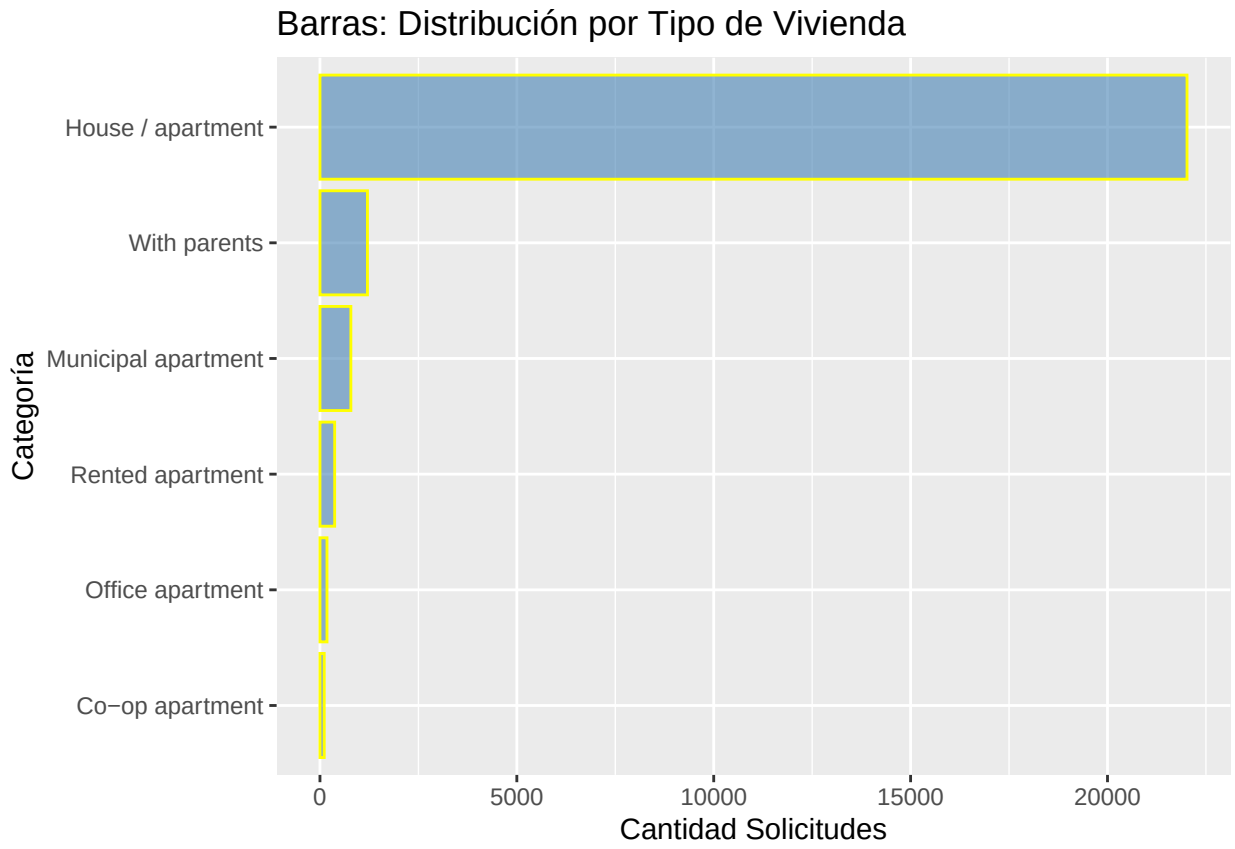


Se visualiza una ligera relación entre las variables. Podría ser un posible predictor.

4.3.11 - Variable: NAME_HOUSING_TYPE

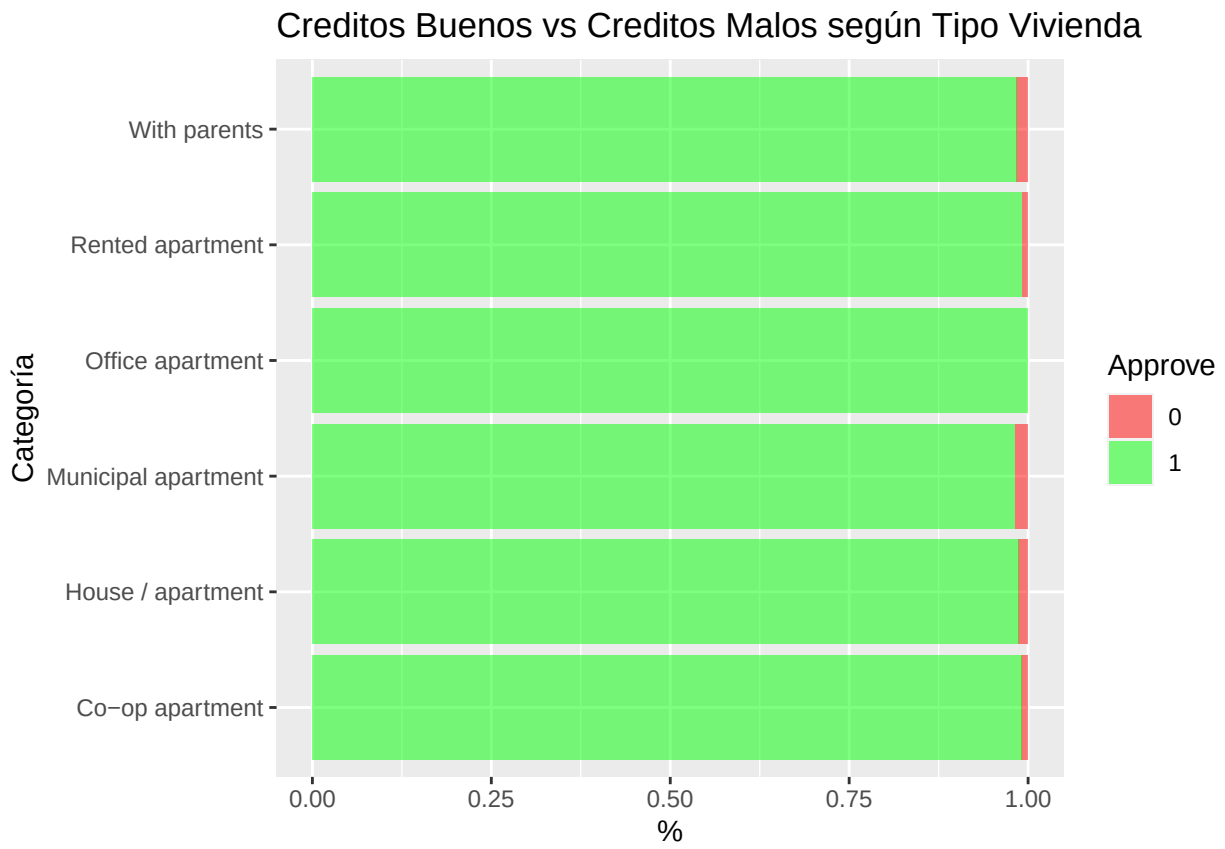
NAME_HOUSING_TYPE Para conocere la distribución de la muestra graficamos con barras diferentes categorías:

```
catIncomeType <- table(dfCreditCard$NAME_HOUSING_TYPE)
dfCatIncomeType <- as.data.frame(catIncomeType)
names(dfCatIncomeType) <- c("Categoria","Cantidad")
# Ordenamos
dfCatIncomeType <- transform(dfCatIncomeType, Categoria=reorder(Categoria, Cantidad))
ggplot(dfCatIncomeType) +
  geom_bar(
    aes(x=Categoria, y=Cantidad),
    stat="identity",
    fill="steel blue",
    col= "yellow",
    alpha= .6) +
    labs(title="Barras: Distribución por Tipo de Vivienda", x="Categoría", y="Cantidad Solicitudes") +
    coord_flip()
```



NAME_HOUSING_TYPE vs APPROVE Verificamos si podría existir alguna relación entre ambas variables mediante un gráfico de barras apiladas.

```
ggplot(dfCreditCard) +
  geom_bar(aes(x=NAME_HOUSING_TYPE, fill=Approve), position="fill", alpha=.5) +
  labs(title="Creditos Buenos vs Creditos Malos según Tipo Vivienda", x="Categoría", y="%") +
  scale_fill_manual(values= c("0"= "red", "1"="green")) +
    coord_flip()
```



Se visualiza una relación entre las variables. Podría ser un posible predictor.

4.3.12 - Variable: OCCUPATION_TYPE

OCCUPATION_TYPE Como se observó en el summary, la categoría tiene datos incompletos.

```
summary(dfCreditCard$OCCUPATION_TYPE)
```

```
##                              Accountants        Cleaning staff
##                 7629                 853                   385
##         Cooking staff            Core staff               Drivers
##                  449                2381                  1504
## High skill tech staff              HR staff            IT staff
##                  950                  53                    46
##              Laborers    Low-skill Laborers             Managers
##                 4293                 123                  2008
##        Medicine staff Private service staff         Realty agents
##                  796                 235                    47
##           Sales staff           Secretaries        Security staff
##                 2328                  98                   406
##  Waiters/barmen staff
##                   88
```

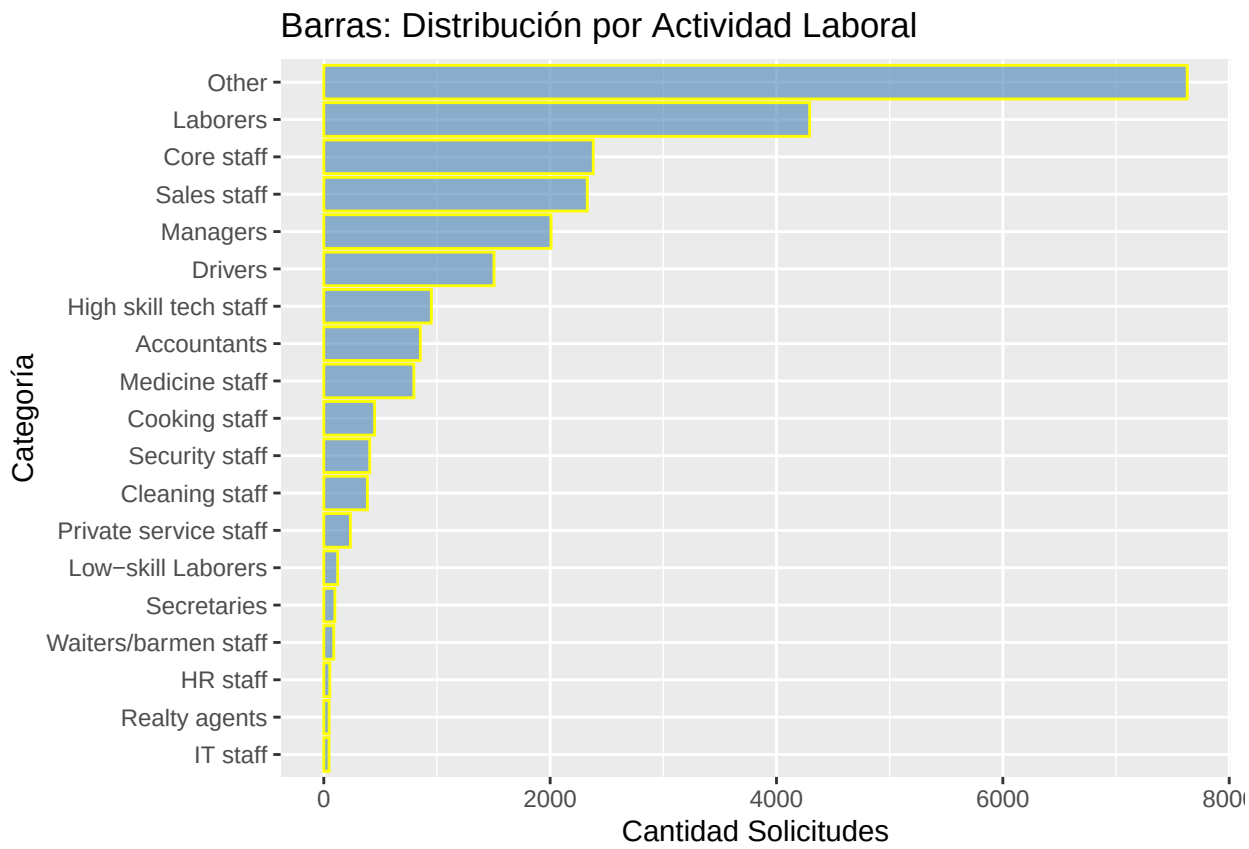
No podemos consultar a negocio si son solicitudes incompletas o representa la categoría “Otros”. A los fines del ejercicio se decide etiquetarlos como “Otros”.

```
levels(dfCreditCard$OCCUPATION_TYPE)[levels(dfCreditCard$OCCUPATION_TYPE) == ""] <- "Other"
summary(dfCreditCard$OCCUPATION_TYPE)
```

```
##                 Other           Accountants        Cleaning staff
##                  7629                   853                   385
##         Cooking staff            Core staff               Drivers
##                   449                  2381                  1504
## High skill tech staff              HR staff            IT staff
##                   950                    53                    46
##              Laborers    Low-skill Laborers              Managers
##                  4293                   123                  2008
##        Medicine staff Private service staff         Realty agents
##                   796                   235                    47
##           Sales staff           Secretaries        Security staff
##                  2328                    98                   406
##  Waiters/barmen staff
##                    88
```

Para conocere la distribución de la muestra graficamos con barras diferentes categorías:

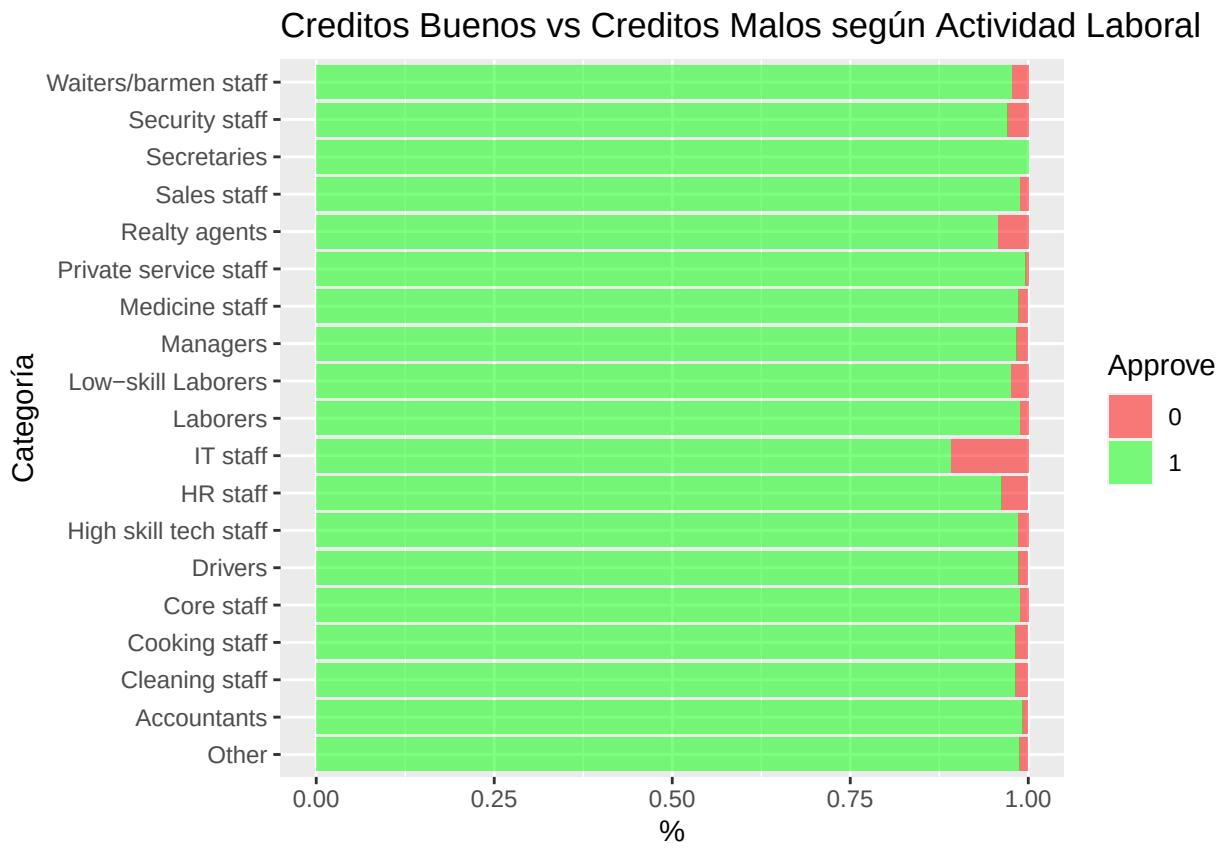
```
catIncomeType <- table(dfCreditCard$OCCUPATION_TYPE)
dfCatIncomeType <- as.data.frame(catIncomeType)
names(dfCatIncomeType) <- c("Categoria","Cantidad")
# Ordenamos
dfCatIncomeType <- transform(dfCatIncomeType, Categoria=reorder(Categoria, Cantidad))
ggplot(dfCatIncomeType) +
  geom_bar(
    aes(x=Categoria, y=Cantidad),
    stat="identity",
    fill="steel blue",
    col= "yellow",
    alpha= .6) +
    labs(title="Barras: Distribución por Actividad Laboral", x="Categoría", y="Cantidad Solicitudes") +
    coord_flip()
```



En esta variable existen muchas categorías. Se deberá evaluar si afecta o no al modelo.

OCCUPATION_TYPE vs APPROVE Verificamos si podría existir alguna relación entre ambas variables mediante un gráfico de barras apiladas.

```
ggplot(dfCreditCard) +
  geom_bar(aes(x=OCCUPATION_TYPE, fill=Approve), position="fill", alpha=.5) +
  labs(title="Creditos Buenos vs Creditos Malos según Actividad Laboral", x="Categoría", y="%") +
  scale_fill_manual(values= c("0"= "red", "1"="green")) +
    coord_flip()
```



Se visualiza una relación entre las variables. Podría ser un posible predictor.

Construcción del Modelo Predictivo

1 - Preparar los datasets de Training y de Testing

Dividimos el set de datos en un 80% para entrenamiento y un 20% para prueba.

```
muestra <- floor(nrow(dfCreditCard) * 0.8)
trainingIndex <- sample(nrow(dfCreditCard), muestra, replace=FALSE)
testIndex <- seq_len(nrow(dfCreditCard))[!(seq_len(nrow(dfCreditCard)) %in% trainingIndex)]
dfTrainLR <- dfCreditCard[trainingIndex,]
dfTestLR <- dfCreditCard[testIndex,]
```

```
summary(dfTrainLR)
```

```
##        ID          CODE_GENDER FLAG_OWN_CAR FLAG_OWN_REALTY  CNT_CHILDREN
##  Min.   :5008804   F:13186     N:12263      N: 6742         Min.   : 0.0000
##  1st Qu.:5045521   M: 6551     Y: 7474      Y:12995         1st Qu.: 0.0000
##  Median :5069509                                           Median : 0.0000
##  Mean   :5078994                                           Mean   : 0.4159
##  3rd Qu.:5115548                                           3rd Qu.: 1.0000
##  Max.   :5150487                                           Max.   :19.0000
##
##  AMT_INCOME_TOTAL              NAME_INCOME_TYPE
##  Min.   :  27000   Commercial associate: 4688
##  1st Qu.: 121500   Pensioner           : 3305
```

```
##  Median : 157500   State servant       : 1562
##  Mean   : 186881   Working             :10182
##  3rd Qu.: 225000
##  Max.   :1575000
##
##                     NAME_EDUCATION_TYPE            NAME_FAMILY_STATUS
##  Lower secondary              :  195    Civil marriage      : 1592
##  Secondary / secondary special:13435    Married             :13566
##  Incomplete higher            :  770    Separated           : 1157
##  Higher education             : 5320    Single / not married: 2607
##  Academic degree              :   17    Widow               :  815
##
##
##           NAME_HOUSING_TYPE   DAYS_BIRTH     DAYS_EMPLOYED    FLAG_MOBIL
##  Co-op apartment    :   87   Min.   :-25152   Min.   :-15713   1:19737
##  House / apartment  :17628   1st Qu.:-19453   1st Qu.: -3170
##  Municipal apartment:  607   Median :-15653   Median : -1567
##  Office apartment   :  148   Mean   :-16022   Mean   : 58677
##  Rented apartment   :  290   3rd Qu.:-12544   3rd Qu.:  -407
##  With parents       :  977   Max.   : -7489   Max.   :365243
##
##  FLAG_WORK_PHONE FLAG_PHONE FLAG_EMAIL    OCCUPATION_TYPE CNT_FAM_MEMBERS
##  0:15229         0:13933    0:17939    Other      :6087   Min.   : 1.000
##  1: 4508         1: 5804    1: 1798    Laborers   :3423   1st Qu.: 2.000
##                                        Core staff :1894   Median : 2.000
##                                        Sales staff:1851   Mean   : 2.185
##                                        Managers   :1600   3rd Qu.: 3.000
##                                        Drivers    :1180   Max.   :20.000
##                                        (Other)    :3702
##  MONTHS_BALANCE     STATUS      Approve        AGE        YEARS_EMPLOYED
##  Min.   :0      C      :10375   0:  266   Min.   :20.00   Min.   : 0.000
##  1st Qu.:0      0      : 5497   1:19471   1st Qu.:34.00   1st Qu.: 3.000
##  Median :0      X      : 3599             Median :42.00   Median : 4.000
##  Mean   :0      1      :  193             Mean   :43.39   Mean   : 6.346
##  3rd Qu.:0      5      :   52             3rd Qu.:53.00   3rd Qu.: 8.000
##  Max.   :0      2      :   13             Max.   :68.00   Max.   :43.000
##                 (Other):    8
```

Como la variable objetivo está muy desbalanceada se decidió aplicar la técnica SMOTE (Synthetic Minority Over-Sampling Technique) para mejorar el balance.

Aplicamos la técnica SMOTE, con un over de 500% para el grupo minoritario y un under de 150% para el grupo mayoritario. Con un un k=5 que representa el número de vecinos cercanos que va a utilzar el algoritmo. La cantidad de vecinos cercanos tomamos la que utiliza la técnica por defecto, mientras que los % de grupos minoritarios y mayoritarios los fuimos ajustamos un par de veces hasta que nos dió balanceada la cantidad de Creditos Buenos y Creditos Malos.

```
dfTrainLR <- SMOTE(Approve ~ ., dfTrainLR, perc.over = 500, k=5, perc.under=150)
```

```
## Warning in if (class(data[, col]) %in% c("factor", "character")) {: the
## condition has length > 1 and only the first element will be used
```

```
## Warning in `[<-.factor`(`*tmp*`, ri, value = c(3.27886032499373, 2,
## 3.4557075118646, : invalid factor level, NA generated
```

```
str(dfTrainLR)
```

```
## 'data.frame':    3591 obs. of  23 variables:
##  $ ID                 : num  5140046 5143234 5022917 5125819 5117423 ...
##  $ CODE_GENDER        : Factor w/ 2 levels "F","M": 1 1 1 1 2 1 1 1 1 2 ...
##  $ FLAG_OWN_CAR       : Factor w/ 2 levels "N","Y": 1 2 1 1 2 1 1 1 1 1 ...
##  $ FLAG_OWN_REALTY    : Factor w/ 2 levels "N","Y": 1 2 2 2 1 2 2 2 2 2 ...
##  $ CNT_CHILDREN       : num  0 1 0 2 3 0 0 0 2 0 ...
##  $ AMT_INCOME_TOTAL   : num  90000 1575000 225000 157500 180000 ...
##  $ NAME_INCOME_TYPE   : Factor w/ 4 levels "Commercial associate",..: 2 1 4 1 4 2 3 2 4 4 ...
##  $ NAME_EDUCATION_TYPE: Ord.factor w/ 5 levels "Lower secondary"<..: 2 4 4 4 2 2 2 2 2 4 ...
##  $ NAME_FAMILY_STATUS : Factor w/ 5 levels "Civil marriage",..: 5 4 4 2 2 2 2 2 2 4 ...
##  $ NAME_HOUSING_TYPE  : Factor w/ 6 levels "Co-op apartment",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ DAYS_BIRTH         : num  -23240 -10142 -12195 -11221 -10048 ...
##  $ DAYS_EMPLOYED      : num  365243 -2479 -2592 -4244 -3475 ...
##  $ FLAG_MOBIL         : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_WORK_PHONE    : Factor w/ 2 levels "0","1": 1 1 1 2 2 1 1 1 1 1 ...
##  $ FLAG_PHONE         : Factor w/ 2 levels "0","1": 1 1 2 2 1 2 1 1 1 1 ...
##  $ FLAG_EMAIL         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ...
##  $ OCCUPATION_TYPE    : Factor w/ 19 levels "Other","Accountants",..: 1 12 12 10 1 1 1 1 16 10 ...
##  $ CNT_FAM_MEMBERS    : num  1 2 1 4 5 2 2 2 4 1 ...
##  $ MONTHS_BALANCE     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ STATUS             : Factor w/ 8 levels "0","1","2","3",..: 8 1 7 7 7 7 1 7 7 1 ...
##  $ Approve            : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ AGE                : num  63 27 33 30 27 62 48 55 38 25 ...
##  $ YEARS_EMPLOYED     : num  4 6 7 11 9 4 6 4 3 2 ...
```

```
summary(dfTrainLR)
```

```
##        ID          CODE_GENDER FLAG_OWN_CAR FLAG_OWN_REALTY  CNT_CHILDREN
##  Min.   :5008804   F:2431      N:2233       N:1421          Min.   :0.0000
##  1st Qu.:5046179   M:1160      Y:1358       Y:2170          1st Qu.:0.0000
##  Median :5077617                                           Median :0.0000
##  Mean   :5079155                                           Mean   :0.3771
##  3rd Qu.:5114713                                           3rd Qu.:0.8224
##  Max.   :5150483                                           Max.   :5.0000
##
##  AMT_INCOME_TOTAL              NAME_INCOME_TYPE
##  Min.   :  33300   Commercial associate: 701
##  1st Qu.: 112620   Pensioner           :1075
##  Median : 157500   State servant       : 179
##  Mean   : 181845   Working             :1636
##  3rd Qu.: 225000
##  Max.   :1575000
##
##                      NAME_EDUCATION_TYPE            NAME_FAMILY_STATUS
##  Lower secondary              :  12     Civil marriage      : 213
##  Secondary / secondary special:1546     Married             :2669
##  Incomplete higher            : 112     Separated           : 176
##  Higher education             : 588     Single / not married: 416
##  Academic degree              :   3     Widow               : 117
##  NA's                         :1330
##
##            NAME_HOUSING_TYPE   DAYS_BIRTH     DAYS_EMPLOYED    FLAG_MOBIL
```

```
##  Co-op apartment    :  14    Min.   :-24963   Min.   :-15227   1:3591
##  House / apartment  :3137    1st Qu.:-20469   1st Qu.: -2175
##  Municipal apartment: 231    Median :-16959   Median :  -528
##  Office apartment   :  16    Mean   :-16784   Mean   :101994
##  Rented apartment   :  47    3rd Qu.:-13248   3rd Qu.:225645
##  With parents       : 146    Max.   : -7757   Max.   :365243
##
##  FLAG_WORK_PHONE FLAG_PHONE FLAG_EMAIL      OCCUPATION_TYPE CNT_FAM_MEMBERS
##  0:2790          0:2564     0:3329     Other        :1452   Min.   :1.000
##  1: 801          1:1027     1: 262     Laborers     : 505   1st Qu.:2.000
##                                        Managers     : 272   Median :2.000
##                                        Sales staff  : 270   Mean   :2.187
##                                        Core staff   : 251   3rd Qu.:2.610
##                                        Cooking staff: 173   Max.   :7.000
##                                        (Other)      : 668
##  MONTHS_BALANCE     STATUS     Approve       AGE        YEARS_EMPLOYED
##  Min.   :0      1      :1229   0:1596   Min.   :21.00   Min.   : 0.000
##  1st Qu.:0      C      :1076   1:1995   1st Qu.:36.00   1st Qu.: 2.745
##  Median :0      0      : 566            Median :46.00   Median : 4.000
##  Mean   :0      X      : 353            Mean   :45.50   Mean   : 5.591
##  3rd Qu.:0      5      : 176            3rd Qu.:55.68   3rd Qu.: 7.000
##  Max.   :0      4      : 142            Max.   :68.00   Max.   :41.000
##                 (Other):  49
```

2 - Fittiamos el modelo de Regresión Logística

Como el problema planteado es de clasificación binaria, es decir intentamos predecir una variable dicotómica, el modelo a probar primero es la Regresión Logística.

2.1 - Seleccionamos Predictores en base al análisis

Como primera aproximación vamos a utilizar como predictor la variable *OCCUPATION_TYPE* que es la que mayor relación con la variable objetivo mostraba en los gráficos.

```
objetivo <- "Approve"
predictores <- c("OCCUPATION_TYPE")
```

2.2 - Creamos la formula del modelo

```
formula <- paste(objetivo, paste(predictores, collapse = " + "), sep=" ~ ")
print(formula)
```

```
## [1] "Approve ~ OCCUPATION_TYPE"
```

2.3 - Entrenamos el modelo

```
modeloLR <- glm(formula, data=dfTrainLR, family=binomial(link="logit"))
```

```
summary(modeloLR)
```

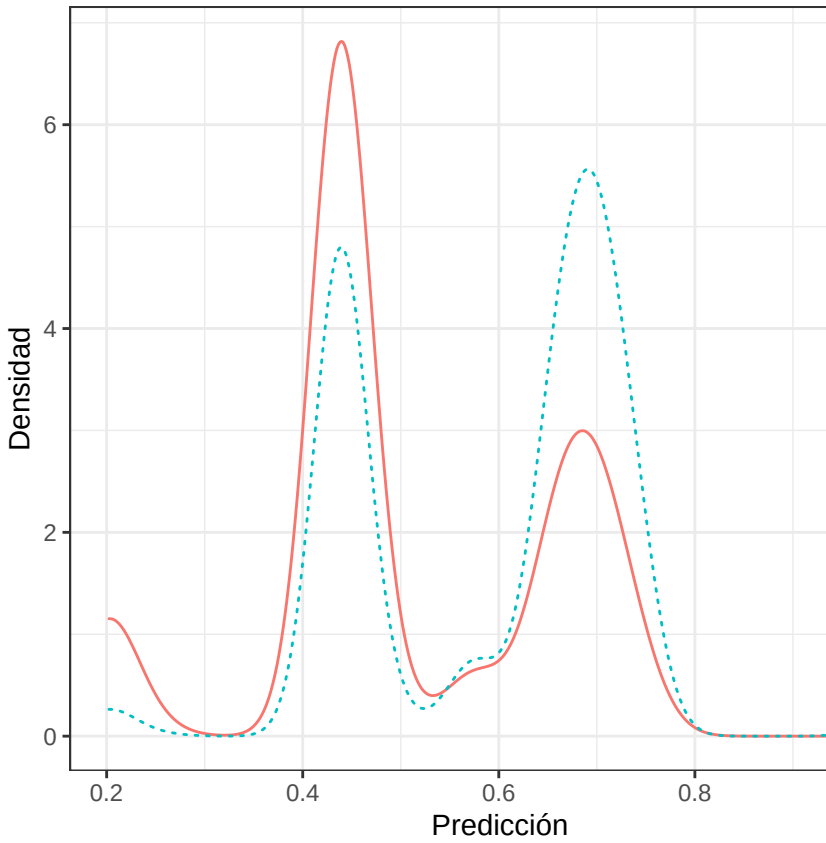
2.3.1 - Visualizamos el summary del modelo

```
##
## Call:
```

```
## glm(formula = formula, family = binomial(link = "logit"), data = dfTrainLR)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6345  -1.0747   0.7812   0.9209   1.7877
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        -0.24642    0.05289  -4.659 3.17e-06 ***
## OCCUPATION_TYPEAccountants          1.19550    0.24616   4.857 1.19e-06 ***
## OCCUPATION_TYPECleaning staff       0.52267    0.28764   1.817   0.0692 .
## OCCUPATION_TYPECooking staff       -1.12549    0.19651  -5.727 1.02e-08 ***
## OCCUPATION_TYPECore staff           0.96960    0.14460   6.705 2.01e-11 ***
## OCCUPATION_TYPEDrivers              1.01961    0.17281   5.900 3.63e-09 ***
## OCCUPATION_TYPEHigh skill tech staff  0.83948  0.19941   4.210 2.56e-05 ***
## OCCUPATION_TYPEHR staff             1.16271    0.83833   1.387   0.1655
## OCCUPATION_TYPEIT staff            -0.85219    0.66876  -1.274   0.2026
## OCCUPATION_TYPELaborers             1.09844    0.11065   9.927  < 2e-16 ***
## OCCUPATION_TYPELow-skill Laborers   0.55657    0.40047   1.390   0.1646
## OCCUPATION_TYPEManagers             0.88491    0.13803   6.411 1.45e-10 ***
## OCCUPATION_TYPEMedicine staff       0.55315    0.21750   2.543   0.0110 *
## OCCUPATION_TYPEPrivate service staff  14.81249  214.09672  0.069   0.9448
## OCCUPATION_TYPERealty agents        0.09227    0.55886   0.165   0.8689
## OCCUPATION_TYPESales staff          1.27704    0.14801   8.628  < 2e-16 ***
## OCCUPATION_TYPESecretaries         14.81249  333.64564   0.044   0.9646
## OCCUPATION_TYPESecurity staff       0.21133    0.27017   0.782   0.4341
## OCCUPATION_TYPEWaiters/barmen staff  0.06410   0.60784   0.105   0.9160
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4933.8  on 3590  degrees of freedom
## Residual deviance: 4593.7  on 3572  degrees of freedom
## AIC: 4631.7
##
## Number of Fisher Scoring iterations: 13
```

2.3.2 - Evaluamos el modelo

```
dfTrainLR$pred <- predict(modeloLR, newdata= dfTrainLR, type="response")
dfTestLR$pred <- predict(modeloLR, newdata= dfTestLR, type="response")
ggplot(dfTrainLR) +
  geom_density(aes(x=pred, colour=as.factor(Approve), linetype=as.factor(Approve))) +
  xlab("Predicción") +
  ylab("Densidad") +
  theme_bw()
```



2.3.2.1 - Plotear distribución de predicciones

Se observan los los picos bien definidos y mucho ruido dentro de cada categoría.

2.3.2.2 - Matriz de Confusión

Probamos con un umbral del 55%.

```
thresh <- 0.55
predObj <-prediction(dfTrainLR$pred, dfTrainLR$Approve)
precObj <- performance(predObj, measure = "prec")
recObj <- performance(predObj, measure = "rec")
precision <- (precObj@y.values)[[1]]
prec.x <- (precObj@x.values)[[1]]
recall <- (recObj@y.values)[[1]]
pnull <- (length(which(dfTrainLR$Approve == 1)) / length(which(dfTrainLR$Approve != 2)))
pnull
```

```
## [1] 0.5555556
```

```
ctab.test <- table(prediccion=as.numeric(dfTestLR$pred > thresh), datos=dfTestLR$Approve)
ctab.test
```

```
##           datos
## prediccion    0    1
##          0   25 1706
##          1   34 3170
```

```
precision <- ctab.test[2,2]/sum(ctab.test[2,])
paste("Precision:", precision)
```

```
## [1] "Precision: 0.989388264669164"
```

```
recall <- ctab.test[2,2]/sum(ctab.test[,2])
paste("Recall:",recall)
```

```
## [1] "Recall: 0.650123051681706"
```

```
enrich <- precision / pnull
paste("Enrich:",enrich)
```

```
## [1] "Enrich: 1.78089887640449"
```

```
PorcFalsosDetect <- ctab.test[1,1]/sum(ctab.test[,1])
PorcPositivosRech <- ctab.test[1,2]/sum(ctab.test[,2])
paste("% Falsos detectados: ", round(PorcFalsosDetect * 100, digits=2))
```

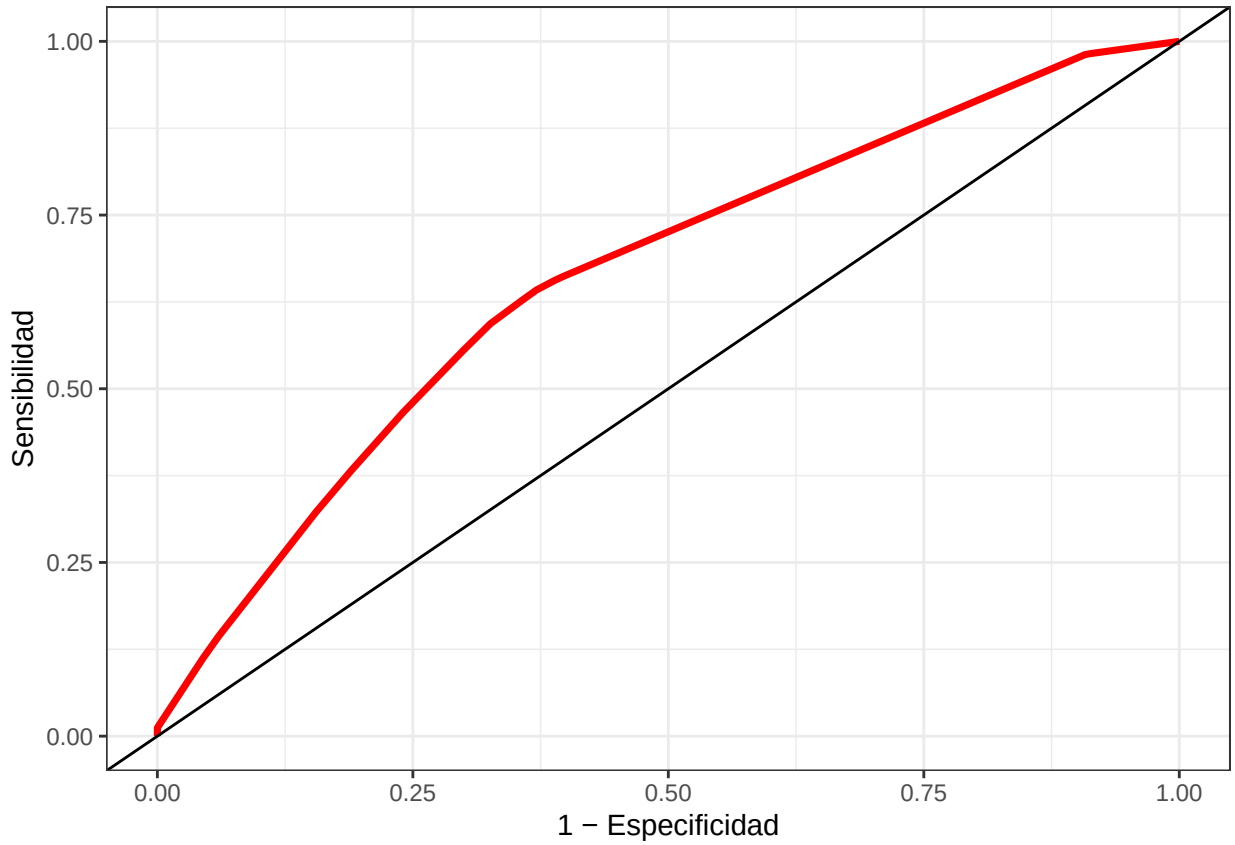
```
## [1] "% Falsos detectados:  42.37"
```

```
paste("% Creditos buenos rechazados: ", round(PorcPositivosRech * 100, digits=2))
```

```
## [1] "% Creditos buenos rechazados:  34.99"
```

2.3.2.3 - Curva ROC Antes de decidir ajustar el umbral graficamos la curva ROC para entender las capacidades de predicción del modelo.

```
sensObj <- performance(predObj, measure="sens")
specObj <- performance(predObj, measure="spec")
sensitivity <- (sensObj@y.values)[[1]]
specificity <- (specObj@y.values)[[1]]
dfplot <- data.frame(specm1=1-specificity, sens=sensitivity)
fig <- ggplot(dfplot, aes(x=specm1, y=sens))
fig <- fig + geom_line(color="red", size=1.25)
fig <- fig + geom_abline(aes(slope=1, intercept=0))
fig <- fig + theme_bw()
fig <- fig + xlab("1 - Especificidad")
fig <- fig + ylab("Sensibilidad")
fig
```



La capacidad de predicción de este modelo es muy baja. Vamos a Reentrenar el modelo con mayor cantidad de variables.

2.4 - Probamos re-entrenar el modelo

Probaremos con el resto de los predictores.

```
predictores <- c("CODE_GENDER", "FLAG_OWN_CAR", "FLAG_OWN_REALTY", "AMT_INCOME_TOTAL", "NAME_INCOME_TYPE
```

```
print("Predictores:")
```

```
## [1] "Predictores:"
```

```
print(predictores)
```

```
##  [1] "CODE_GENDER"          "FLAG_OWN_CAR"         "FLAG_OWN_REALTY"
##  [4] "AMT_INCOME_TOTAL"     "NAME_INCOME_TYPE"     "NAME_EDUCATION_TYPE"
##  [7] "NAME_FAMILY_STATUS"   "NAME_HOUSING_TYPE"    "FLAG_WORK_PHONE"
## [10] "FLAG_PHONE"           "FLAG_EMAIL"           "OCCUPATION_TYPE"
## [13] "CNT_FAM_MEMBERS"      "AGE"                  "YEARS_EMPLOYED"
```

```
formula <- paste(objetivo, paste(predictores, collapse = " + "), sep=" ~ ")
modeloLR <- glm(formula, data=dfTrainLR, family=binomial(link="logit"))
```

Visualizamos el summary del modelo.

```
summary(modeloLR)
```

```
##
```

```
## Call:
## glm(formula = formula, family = binomial(link = "logit"), data = dfTrainLR)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4003   0.4047   0.4710   0.5266   1.0948
##
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                          5.536e+00  2.771e+02   0.020   0.9841
## CODE_GENDERM                        -1.701e-01  1.707e-01  -0.996   0.3190
## FLAG_OWN_CARY                        6.223e-02  1.543e-01   0.403   0.6867
## FLAG_OWN_REALTYY                     1.040e-01  1.487e-01   0.700   0.4842
## AMT_INCOME_TOTAL                    -2.725e-07  6.788e-07  -0.401   0.6881
## NAME_INCOME_TYPEPensioner           -2.453e-01  3.209e-01  -0.764   0.4446
## NAME_INCOME_TYPEState servant        5.109e-01  3.372e-01   1.515   0.1298
## NAME_INCOME_TYPEWorking              8.632e-02  1.677e-01   0.515   0.6068
## NAME_EDUCATION_TYPE.L                9.346e+00  8.762e+02   0.011   0.9915
## NAME_EDUCATION_TYPE.Q                7.320e+00  7.405e+02   0.010   0.9921
## NAME_EDUCATION_TYPE.C                4.448e+00  4.381e+02   0.010   0.9919
## NAME_EDUCATION_TYPE^4                1.564e+00  1.656e+02   0.009   0.9925
## NAME_FAMILY_STATUSMarried           -1.237e-01  2.733e-01  -0.453   0.6509
## NAME_FAMILY_STATUSSeparated         -1.485e-01  3.980e-01  -0.373   0.7091
## NAME_FAMILY_STATUSSingle / not married -4.425e-01  3.334e-01  -1.327   0.1844
## NAME_FAMILY_STATUSWidow              7.574e-02  4.626e-01   0.164   0.8700
## NAME_HOUSING_TYPEHouse / apartment  -5.690e-01  1.089e+00  -0.523   0.6013
## NAME_HOUSING_TYPEMunicipal apartment -1.027e+00  1.134e+00  -0.905   0.3652
## NAME_HOUSING_TYPEOffice apartment    1.405e+01  5.924e+02   0.024   0.9811
## NAME_HOUSING_TYPERented apartment   -1.020e-01  1.243e+00  -0.082   0.9346
## NAME_HOUSING_TYPEWith parents       -6.461e-01  1.130e+00  -0.572   0.5675
## FLAG_WORK_PHONE1                     7.136e-02  1.825e-01   0.391   0.6958
## FLAG_PHONE1                         -6.913e-02  1.523e-01  -0.454   0.6500
## FLAG_EMAIL1                         -1.815e-02  2.334e-01  -0.078   0.9380
## OCCUPATION_TYPEAccountants          -1.882e-01  4.512e-01  -0.417   0.6766
## OCCUPATION_TYPECleaning staff       -8.623e-01  4.781e-01  -1.804   0.0713 .
## OCCUPATION_TYPECooking staff        -4.676e-01  4.897e-01  -0.955   0.3396
## OCCUPATION_TYPECore staff           -4.630e-01  2.870e-01  -1.613   0.1067
## OCCUPATION_TYPEDrivers              -1.884e-01  3.399e-01  -0.554   0.5795
## OCCUPATION_TYPEHigh skill tech staff -4.791e-01  3.708e-01  -1.292   0.1962
## OCCUPATION_TYPEHR staff             -6.406e-01  1.122e+00  -0.571   0.5681
## OCCUPATION_TYPEIT staff             -1.639e+00  9.434e-01  -1.737   0.0823 .
## OCCUPATION_TYPELaborers              1.600e-03  2.576e-01   0.006   0.9950
## OCCUPATION_TYPELow-skill Laborers   -4.304e-01  6.691e-01  -0.643   0.5201
## OCCUPATION_TYPEManagers             -4.303e-01  2.969e-01  -1.450   0.1472
## OCCUPATION_TYPEMedicine staff       -8.814e-01  3.935e-01  -2.240   0.0251 *
## OCCUPATION_TYPEPrivate service staff  1.440e+01  5.795e+02   0.025   0.9802
## OCCUPATION_TYPERealty agents        -1.206e+00  8.623e-01  -1.399   0.1619
## OCCUPATION_TYPESales staff           2.110e-01  3.168e-01   0.666   0.5055
## OCCUPATION_TYPESecretaries           1.426e+01  9.023e+02   0.016   0.9874
## OCCUPATION_TYPESecurity staff       -8.212e-01  4.616e-01  -1.779   0.0753 .
## OCCUPATION_TYPEWaiters/barmen staff  -1.424e+00  8.666e-01  -1.644   0.1003
## CNT_FAM_MEMBERS                     -6.364e-02  9.901e-02  -0.643   0.5204
## AGE                                  4.851e-03  8.679e-03   0.559   0.5762
## YEARS_EMPLOYED                       1.642e-02  1.328e-02   1.236   0.2163
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1637.9  on 2260  degrees of freedom
## Residual deviance: 1590.3  on 2216  degrees of freedom
##   (1330 observations deleted due to missingness)
## AIC: 1680.3
##
## Number of Fisher Scoring iterations: 15
```

Visualizamos el Factor de Inflación de Varianza:

```
vif(modeloLR)
```

```
##                            CODE_GENDERM                            FLAG_OWN_CARY
##                            1.575918e+00                            1.290879e+00
##                          FLAG_OWN_REALTYY                          AMT_INCOME_TOTAL
##                            1.165717e+00                            1.285749e+00
##                NAME_INCOME_TYPEPensioner             NAME_INCOME_TYPEState servant
##                            3.482000e+00                            1.258608e+00
##                  NAME_INCOME_TYPEWorking                   NAME_EDUCATION_TYPE.L
##                            1.613993e+00                            1.278386e+07
##                    NAME_EDUCATION_TYPE.Q                   NAME_EDUCATION_TYPE.C
##                            1.045966e+06                            1.261618e+07
##                    NAME_EDUCATION_TYPE^4                NAME_FAMILY_STATUSMarried
##                            4.598766e+05                            3.720227e+00
##              NAME_FAMILY_STATUSSeparated NAME_FAMILY_STATUSSingle / not married
##                            1.957709e+00                            3.430952e+00
##                 NAME_FAMILY_STATUSWidow     NAME_HOUSING_TYPEHouse / apartment
##                            1.661640e+00                            2.728031e+01
##    NAME_HOUSING_TYPEMunicipal apartment      NAME_HOUSING_TYPEOffice apartment
##                            1.186062e+01                            1.000003e+00
##       NAME_HOUSING_TYPERented apartment          NAME_HOUSING_TYPEWith parents
##                            4.168221e+00                            1.531070e+01
##                          FLAG_WORK_PHONE1                             FLAG_PHONE1
##                            1.323516e+00                            1.152248e+00
##                             FLAG_EMAIL1             OCCUPATION_TYPEAccountants
##                            1.052631e+00                            1.235267e+00
##           OCCUPATION_TYPECleaning staff           OCCUPATION_TYPECooking staff
##                            1.234367e+00                            1.182197e+00
##                OCCUPATION_TYPECore staff                 OCCUPATION_TYPEDrivers
##                            1.651415e+00                            1.587764e+00
##    OCCUPATION_TYPEHigh skill tech staff                OCCUPATION_TYPEHR staff
##                            1.341197e+00                            1.041833e+00
##                 OCCUPATION_TYPEIT staff                OCCUPATION_TYPELaborers
##                            1.057598e+00                            2.036303e+00
##       OCCUPATION_TYPELow-skill Laborers                OCCUPATION_TYPEManagers
##                            1.099941e+00                            1.780446e+00
##           OCCUPATION_TYPEMedicine staff   OCCUPATION_TYPEPrivate service staff
##                            1.333957e+00                            1.000000e+00
##            OCCUPATION_TYPERealty agents             OCCUPATION_TYPESales staff
##                            1.098981e+00                            1.529917e+00
##              OCCUPATION_TYPESecretaries          OCCUPATION_TYPESecurity staff
```

```
##                              1.000000e+00                              1.282749e+00
##    OCCUPATION_TYPEWaiters/barmen staff                     CNT_FAM_MEMBERS
##                              1.059843e+00                              1.900574e+00
##                                       AGE                       YEARS_EMPLOYED
##                              2.397502e+00                              1.231306e+00
```

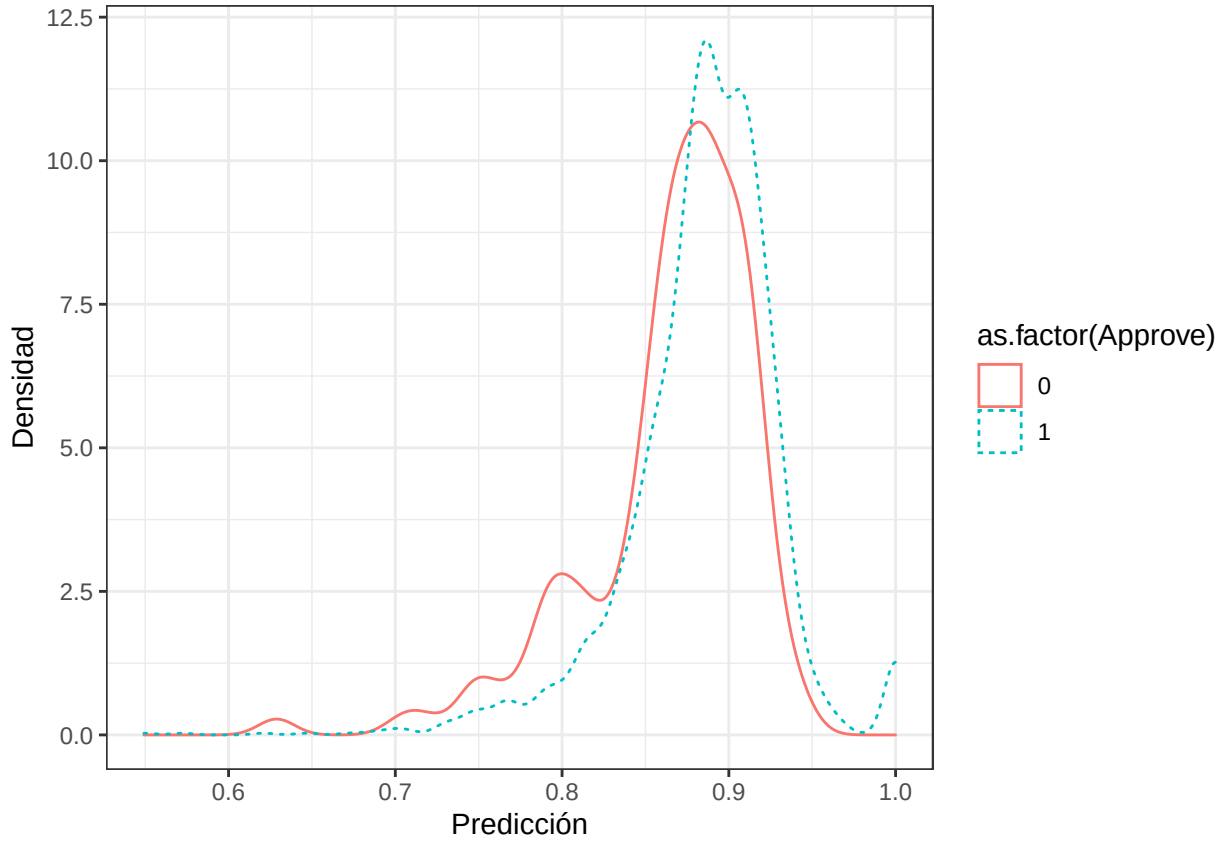
Observamos que NAME_EDUCATION_TYPE está muy correlacionado con otras variables.

2.4.1 - Evaluamos el modelo

```
dfTrainLR$pred <- predict(modeloLR, newdata= dfTrainLR, type="response")
dfTestLR$pred <- predict(modeloLR, newdata= dfTestLR, type="response")
ggplot(dfTrainLR) +
  geom_density(aes(x=pred, colour=as.factor(Approve), linetype=as.factor(Approve))) +
  xlab("Predicción") +
  ylab("Densidad") +
  theme_bw()
```

2.4.1.1 - Plotear distribución de predicciones

```
## Warning: Removed 1330 rows containing non-finite values (stat_density).
```



La distribución de predicciones nos marca que este modelo no es el adecuado, no tenemos 2 picos claramente definidos.

2.5 - Probamos re-entrenar el modelo

Vamos a quitar NAME_EDUCATION_TYPE y re-entrenar el modelo.

```
predictores <- c("CODE_GENDER", "FLAG_OWN_CAR", "FLAG_OWN_REALTY", "AMT_INCOME_TOTAL", "NAME_INCOME_TYPE
print("Predictores:")
```

```
## [1] "Predictores:"
```

```
print(predictores)
```

```
##  [1] "CODE_GENDER"        "FLAG_OWN_CAR"       "FLAG_OWN_REALTY"
##  [4] "AMT_INCOME_TOTAL"   "NAME_INCOME_TYPE"   "NAME_FAMILY_STATUS"
##  [7] "NAME_HOUSING_TYPE"  "FLAG_WORK_PHONE"    "FLAG_PHONE"
## [10] "FLAG_EMAIL"         "OCCUPATION_TYPE"    "CNT_FAM_MEMBERS"
## [13] "AGE"                "YEARS_EMPLOYED"
```

```
formula <- paste(objetivo, paste(predictores, collapse = " + "), sep=" ~ ")
modeloLR <- glm(formula, data=dfTrainLR, family=binomial(link="logit"))
```

Visualizamos el summary del modelo

```
summary(modeloLR)
```

```
##
## Call:
## glm(formula = formula, family = binomial(link = "logit"), data = dfTrainLR)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4006  -0.9577   0.5285   0.8724   2.3462
##
## Coefficients:
##                                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)                           1.565e+00  7.581e-01   2.064 0.038993
## CODE_GENDERM                          2.761e-01  9.452e-02   2.921 0.003486
## FLAG_OWN_CARY                        -2.232e-01  8.429e-02  -2.648 0.008102
## FLAG_OWN_REALTYY                      6.690e-01  8.299e-02   8.061 7.57e-16
## AMT_INCOME_TOTAL                      9.006e-08  4.262e-07   0.211 0.832648
## NAME_INCOME_TYPEPensioner            -1.147e+00  1.307e-01  -8.776  < 2e-16
## NAME_INCOME_TYPEState servant         6.201e-01  2.134e-01   2.906 0.003658
## NAME_INCOME_TYPEWorking              -7.568e-02  1.056e-01  -0.717 0.473402
## NAME_FAMILY_STATUSMarried            -6.770e-01  1.770e-01  -3.825 0.000131
## NAME_FAMILY_STATUSSeparated           3.843e-02  2.591e-01   0.148 0.882091
## NAME_FAMILY_STATUSSingle / not married -4.059e-01  2.157e-01  -1.882 0.059808
## NAME_FAMILY_STATUSWidow               9.614e-01  2.964e-01   3.244 0.001179
## NAME_HOUSING_TYPEHouse / apartment    -4.293e-01  6.472e-01  -0.663 0.507136
## NAME_HOUSING_TYPEMunicipal apartment  -1.675e+00  6.679e-01  -2.508 0.012141
## NAME_HOUSING_TYPEOffice apartment      1.449e+01  3.503e+02   0.041 0.967014
## NAME_HOUSING_TYPERented apartment      1.267e-01  7.363e-01   0.172 0.863416
## NAME_HOUSING_TYPEWith parents         -1.131e-01  6.777e-01  -0.167 0.867433
## FLAG_WORK_PHONE1                      -2.052e-01  1.024e-01  -2.003 0.045166
## FLAG_PHONE1                           2.445e-01  8.837e-02   2.767 0.005657
## FLAG_EMAIL1                           2.892e-01  1.562e-01   1.852 0.064057
## OCCUPATION_TYPEAccountants            6.442e-01  2.685e-01   2.399 0.016418
## OCCUPATION_TYPECleaning staff         -1.728e-01  3.236e-01  -0.534 0.593317
## OCCUPATION_TYPECooking staff          -1.711e+00  2.178e-01  -7.858 3.91e-15
```

```
## OCCUPATION_TYPECore staff               2.690e-01  1.666e-01   1.615 0.106349
## OCCUPATION_TYPEDrivers                  4.805e-01  1.988e-01   2.417 0.015649
## OCCUPATION_TYPEHigh skill tech staff    1.040e-01  2.190e-01   0.475 0.634961
## OCCUPATION_TYPEHR staff                 7.066e-01  8.971e-01   0.788 0.430886
## OCCUPATION_TYPEIT staff                -1.429e+00  7.034e-01  -2.032 0.042187
## OCCUPATION_TYPELaborers                 3.978e-01  1.308e-01   3.042 0.002353
## OCCUPATION_TYPELow-skill Laborers      -2.116e-01  4.271e-01  -0.495 0.620345
## OCCUPATION_TYPEManagers                 1.764e-01  1.613e-01   1.093 0.274178
## OCCUPATION_TYPEMedicine staff          -2.779e-01  2.497e-01  -1.113 0.265741
## OCCUPATION_TYPEPrivate service staff    1.470e+01  3.316e+02   0.044 0.964651
## OCCUPATION_TYPERealty agents           -7.367e-01  6.010e-01  -1.226 0.220225
## OCCUPATION_TYPESales staff              8.123e-01  1.686e-01   4.818 1.45e-06
## OCCUPATION_TYPESecretaries              1.498e+01  5.339e+02   0.028 0.977612
## OCCUPATION_TYPESecurity staff          -2.468e-01  2.915e-01  -0.846 0.397334
## OCCUPATION_TYPEWaiters/barmen staff    -4.384e-01  6.279e-01  -0.698 0.485104
## CNT_FAM_MEMBERS                        -7.605e-02  6.197e-02  -1.227 0.219765
## AGE                                    -1.405e-02  4.611e-03  -3.047 0.002314
## YEARS_EMPLOYED                          4.912e-02  8.460e-03   5.806 6.40e-09
##
## (Intercept)                           *
## CODE_GENDERM                          **
## FLAG_OWN_CARY                         **
## FLAG_OWN_REALTYY                      ***
## AMT_INCOME_TOTAL
## NAME_INCOME_TYPEPensioner             ***
## NAME_INCOME_TYPEState servant         **
## NAME_INCOME_TYPEWorking
## NAME_FAMILY_STATUSMarried             ***
## NAME_FAMILY_STATUSSeparated
## NAME_FAMILY_STATUSSingle / not married .
## NAME_FAMILY_STATUSWidow               **
## NAME_HOUSING_TYPEHouse / apartment
## NAME_HOUSING_TYPEMunicipal apartment  *
## NAME_HOUSING_TYPEOffice apartment
## NAME_HOUSING_TYPERented apartment
## NAME_HOUSING_TYPEWith parents
## FLAG_WORK_PHONE1                      *
## FLAG_PHONE1                           **
## FLAG_EMAIL1                           .
## OCCUPATION_TYPEAccountants            *
## OCCUPATION_TYPECleaning staff
## OCCUPATION_TYPECooking staff          ***
## OCCUPATION_TYPECore staff
## OCCUPATION_TYPEDrivers                *
## OCCUPATION_TYPEHigh skill tech staff
## OCCUPATION_TYPEHR staff
## OCCUPATION_TYPEIT staff               *
## OCCUPATION_TYPELaborers               **
## OCCUPATION_TYPELow-skill Laborers
## OCCUPATION_TYPEManagers
## OCCUPATION_TYPEMedicine staff
## OCCUPATION_TYPEPrivate service staff
## OCCUPATION_TYPERealty agents
## OCCUPATION_TYPESales staff            ***
```

```
## OCCUPATION_TYPESecretaries
## OCCUPATION_TYPESecurity staff
## OCCUPATION_TYPEWaiters/barmen staff
## CNT_FAM_MEMBERS
## AGE                                    **
## YEARS_EMPLOYED                         ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4933.8  on 3590  degrees of freedom
## Residual deviance: 4067.5  on 3550  degrees of freedom
## AIC: 4149.5
##
## Number of Fisher Scoring iterations: 14
```

Las variables AMT_INCOME_TOTAL y CNT_FAM_MEMBERS no parecen tener gran capacidad de predicción.

Visualizamos el Factor de Inflación de Varianza:

```
vif(modeloLR)
```

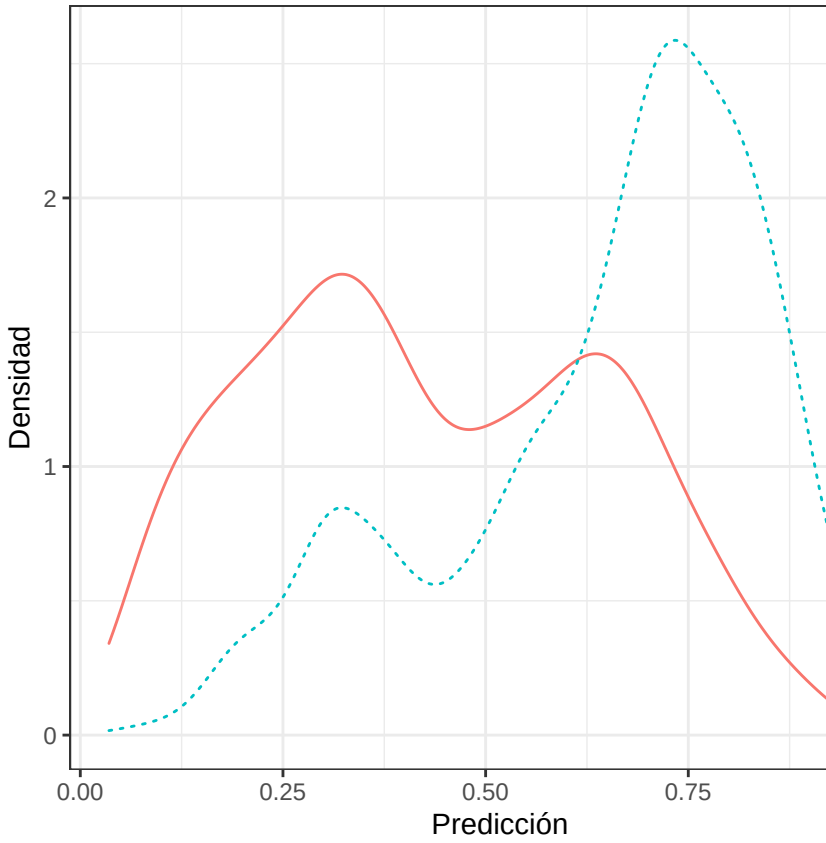
```
##                            CODE_GENDERM                            FLAG_OWN_CARY
##                                1.346555                                 1.171047
##                          FLAG_OWN_REALTYY                          AMT_INCOME_TOTAL
##                                1.143210                                 1.212120
##              NAME_INCOME_TYPEPensioner           NAME_INCOME_TYPEState servant
##                                2.512364                                 1.244086
##                NAME_INCOME_TYPEWorking               NAME_FAMILY_STATUSMarried
##                                1.909110                                 3.849730
##            NAME_FAMILY_STATUSSeparated NAME_FAMILY_STATUSSingle / not married
##                                1.884774                                 3.172434
##                NAME_FAMILY_STATUSWidow     NAME_HOUSING_TYPEHouse / apartment
##                                1.642431                                27.395890
##   NAME_HOUSING_TYPEMunicipal apartment      NAME_HOUSING_TYPEOffice apartment
##                               15.050556                                 1.000003
##      NAME_HOUSING_TYPERented apartment          NAME_HOUSING_TYPEWith parents
##                                4.331351                                11.934803
##                         FLAG_WORK_PHONE1                             FLAG_PHONE1
##                                1.255330                                 1.102759
##                              FLAG_EMAIL1             OCCUPATION_TYPEAccountants
##                                1.047826                                 1.088740
##          OCCUPATION_TYPECleaning staff           OCCUPATION_TYPECooking staff
##                                1.063230                                 1.177455
##              OCCUPATION_TYPECore staff                 OCCUPATION_TYPEDrivers
##                                1.216715                                 1.229416
##   OCCUPATION_TYPEHigh skill tech staff                OCCUPATION_TYPEHR staff
##                                1.108726                                 1.010786
##                OCCUPATION_TYPEIT staff                OCCUPATION_TYPELaborers
##                                1.034078                                 1.383452
##      OCCUPATION_TYPELow-skill Laborers                OCCUPATION_TYPEManagers
##                                1.040421                                 1.284795
##          OCCUPATION_TYPEMedicine staff   OCCUPATION_TYPEPrivate service staff
##                                1.127408                                 1.000000
```

```
##           OCCUPATION_TYPERealty agents             OCCUPATION_TYPESales staff
##                                1.039742                                1.235211
##             OCCUPATION_TYPESecretaries          OCCUPATION_TYPESecurity staff
##                                1.000000                                1.087756
##    OCCUPATION_TYPEWaiters/barmen staff                      CNT_FAM_MEMBERS
##                                1.017612                                1.647904
##                                     AGE                       YEARS_EMPLOYED
##                                1.982217                                1.147302
```

No se observan valores mayores a cinco que puedan estar indicando correlación entre variables.

2.5.1 - Evaluamos el modelo

```
dfTrainLR$pred <- predict(modeloLR, newdata= dfTrainLR, type="response")
dfTestLR$pred <- predict(modeloLR, newdata= dfTestLR, type="response")
ggplot(dfTrainLR) +
  geom_density(aes(x=pred, colour=as.factor(Approve), linetype=as.factor(Approve))) +
  xlab("Predicción") +
  ylab("Densidad") +
  theme_bw()
```



2.5.1.1 - Plotear distribución de predicciones

2.5.1.2 - Matriz de Confusión

Probamos con un umbral del 50%.

```
thresh <- 0.50
predObj <-prediction(dfTrainLR$pred, dfTrainLR$Approve)
precObj <- performance(predObj, measure = "prec")
```

```
recObj <- performance(predObj, measure = "rec")
precision <- (precObj@y.values)[[1]]
prec.x <- (precObj@x.values)[[1]]
recall <- (recObj@y.values)[[1]]
pnull <- (length(which(dfTrainLR$Approve == 1)) / length(which(dfTrainLR$Approve != 2)))
pnull
```

```
## [1] 0.5555556
```

```
ctab.test <- table(prediccion=as.numeric(dfTestLR$pred > thresh), datos=dfTestLR$Approve)
ctab.test
```

```
##           datos
## prediccion    0    1
##          0   17 1050
##          1   42 3826
```

```
precision <- ctab.test[2,2]/sum(ctab.test[2,])
paste("Precision:", precision)
```

```
## [1] "Precision: 0.989141675284385"
```

```
recall <- ctab.test[2,2]/sum(ctab.test[,2])
paste("Recall:",recall)
```

```
## [1] "Recall: 0.784659557013946"
```

```
enrich <- precision / pnull
paste("Enrich:",enrich)
```

```
## [1] "Enrich: 1.78045501551189"
```

```
PorcFalsosDetect <- ctab.test[1,1]/sum(ctab.test[,1])
PorcPositivosRech <- ctab.test[1,2]/sum(ctab.test[,2])
paste("% Falsos detectados: ", round(PorcFalsosDetect * 100, digits=2))
```

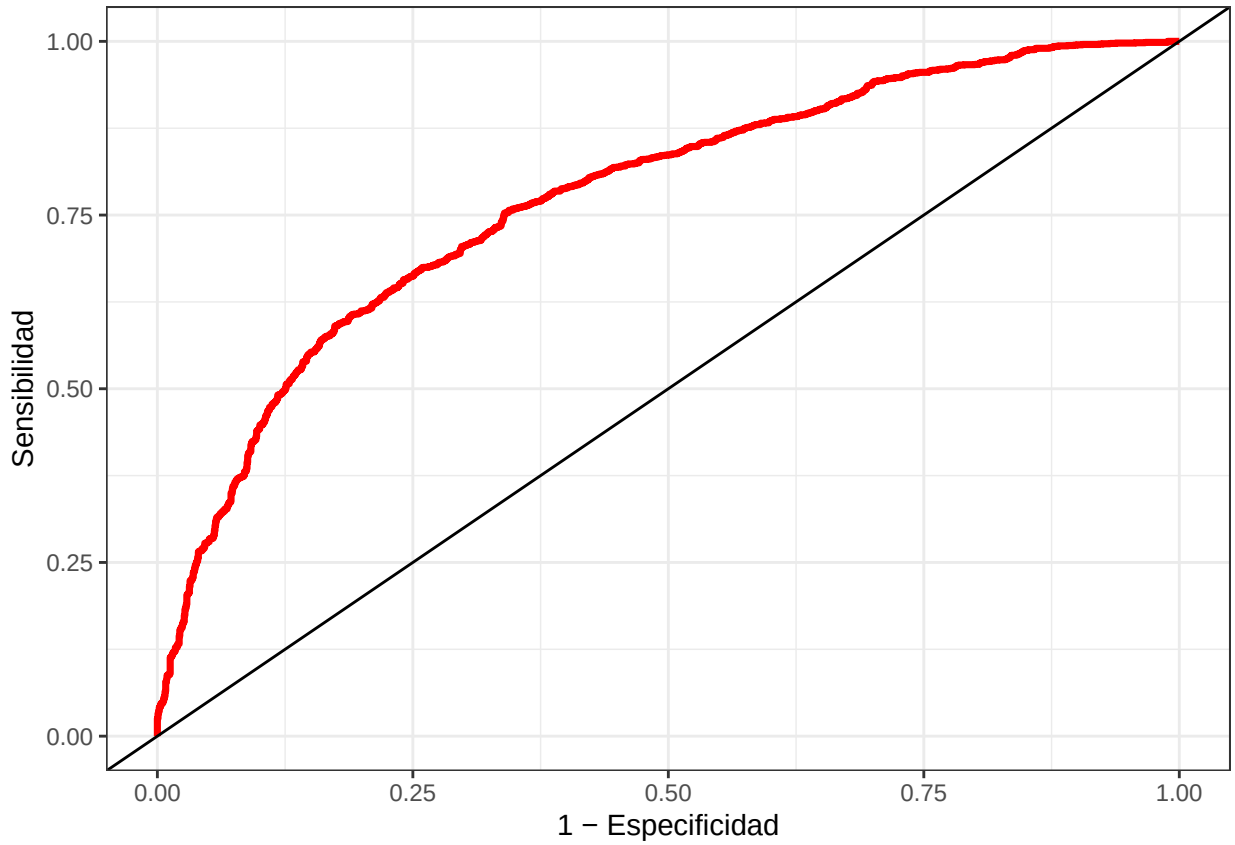
```
## [1] "% Falsos detectados:  28.81"
```

```
paste("% Creditos buenos rechazados: ", round(PorcPositivosRech * 100, digits=2))
```

```
## [1] "% Creditos buenos rechazados:  21.53"
```

2.5.1.3 - Curva ROC Antes de decidir ajustar el umbral graficamos la curva ROC para entender las capacidades de predicción del modelo.

```
sensObj <- performance(predObj, measure="sens")
specObj <- performance(predObj, measure="spec")
sensitivity <- (sensObj@y.values)[[1]]
specificity <- (specObj@y.values)[[1]]
dfplot <- data.frame(specm1=1-specificity, sens=sensitivity)
fig <- ggplot(dfplot, aes(x=specm1, y=sens))
fig <- fig + geom_line(color="red", size=1.25)
fig <- fig + geom_abline(aes(slope=1, intercept=0))
fig <- fig + theme_bw()
fig <- fig + xlab("1 - Especificidad")
fig <- fig + ylab("Sensibilidad")
fig
```



2.6 - Probamos re-entrenar el modelo

Vamos a quitar AMT_INCOME_TOTAL y CNT_FAM_MEMBERS para re-entrenar el modelo.

```
predictores <- c("CODE_GENDER", "FLAG_OWN_CAR", "FLAG_OWN_REALTY", "NAME_INCOME_TYPE", "NAME_FAMILY_STA
```

```
print("Predictores:")
```

```
## [1] "Predictores:"
```

```
print(predictores)
```

```
##  [1] "CODE_GENDER"        "FLAG_OWN_CAR"       "FLAG_OWN_REALTY"
##  [4] "NAME_INCOME_TYPE"   "NAME_FAMILY_STATUS" "NAME_HOUSING_TYPE"
##  [7] "FLAG_WORK_PHONE"    "FLAG_PHONE"         "FLAG_EMAIL"
## [10] "OCCUPATION_TYPE"    "AGE"                "YEARS_EMPLOYED"
```

```
formula <- paste(objetivo, paste(predictores, collapse = " + "), sep=" ~ ")
modeloLR <- glm(formula, data=dfTrainLR, family=binomial(link="logit"))
```

Visualizamos el summary del modelo

```
summary(modeloLR)
```

```
##
## Call:
## glm(formula = formula, family = binomial(link = "logit"), data = dfTrainLR)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -2.3804  -0.9568   0.5289   0.8761   2.3468
##
## Coefficients:
##                                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)                              1.286427   0.703550   1.828 0.067477
## CODE_GENDERM                             0.278096   0.093537   2.973 0.002948
## FLAG_OWN_CARY                           -0.225276   0.083467  -2.699 0.006955
## FLAG_OWN_REALTYY                         0.666858   0.082773   8.056 7.85e-16
## NAME_INCOME_TYPEPensioner               -1.154225   0.129735  -8.897  < 2e-16
## NAME_INCOME_TYPEState servant            0.615945   0.213143   2.890 0.003855
## NAME_INCOME_TYPEWorking                 -0.081450   0.104595  -0.779 0.436148
## NAME_FAMILY_STATUSMarried               -0.682473   0.176975  -3.856 0.000115
## NAME_FAMILY_STATUSSeparated              0.103433   0.253103   0.409 0.682788
## NAME_FAMILY_STATUSSingle / not married  -0.328410   0.206167  -1.593 0.111175
## NAME_FAMILY_STATUSWidow                  1.028951   0.290940   3.537 0.000405
## NAME_HOUSING_TYPEHouse / apartment      -0.390213   0.642929  -0.607 0.543898
## NAME_HOUSING_TYPEMunicipal apartment    -1.638753   0.663758  -2.469 0.013553
## NAME_HOUSING_TYPEOffice apartment       14.483540 351.143744   0.041 0.967099
## NAME_HOUSING_TYPERented apartment        0.176744   0.732305   0.241 0.809281
## NAME_HOUSING_TYPEWith parents           -0.057301   0.672398  -0.085 0.932087
## FLAG_WORK_PHONE1                        -0.207036   0.101906  -2.032 0.042191
## FLAG_PHONE1                              0.248873   0.087739   2.837 0.004561
## FLAG_EMAIL1                              0.298203   0.155759   1.915 0.055553
## OCCUPATION_TYPEAccountants               0.652304   0.268458   2.430 0.015106
## OCCUPATION_TYPECleaning staff           -0.184161   0.323039  -0.570 0.568618
## OCCUPATION_TYPECooking staff            -1.714350   0.217791  -7.872 3.50e-15
## OCCUPATION_TYPECore staff                0.259691   0.165922   1.565 0.117551
## OCCUPATION_TYPEDrivers                   0.486079   0.198805   2.445 0.014485
## OCCUPATION_TYPEHigh skill tech staff     0.108569   0.218927   0.496 0.619956
## OCCUPATION_TYPEHR staff                  0.714950   0.892653   0.801 0.423174
## OCCUPATION_TYPEIT staff                 -1.414645   0.701948  -2.015 0.043872
## OCCUPATION_TYPELaborers                  0.397654   0.130730   3.042 0.002352
## OCCUPATION_TYPELow-skill Laborers       -0.222830   0.426561  -0.522 0.601401
## OCCUPATION_TYPEManagers                  0.180543   0.158233   1.141 0.253873
## OCCUPATION_TYPEMedicine staff           -0.277883   0.249683  -1.113 0.265733
## OCCUPATION_TYPEPrivate service staff    14.700511 332.050640   0.044 0.964688
## OCCUPATION_TYPERealty agents            -0.712673   0.599785  -1.188 0.234749
## OCCUPATION_TYPESales staff               0.803391   0.168154   4.778 1.77e-06
## OCCUPATION_TYPESecretaries              15.012978 534.307699   0.028 0.977584
## OCCUPATION_TYPESecurity staff           -0.262152   0.291298  -0.900 0.368150
## OCCUPATION_TYPEWaiters/barmen staff     -0.431937   0.626920  -0.689 0.490834
## AGE                                     -0.012189   0.004331  -2.814 0.004888
## YEARS_EMPLOYED                           0.048944   0.008454   5.789 7.07e-09
##
## (Intercept)                             .
## CODE_GENDERM                            **
## FLAG_OWN_CARY                           **
## FLAG_OWN_REALTYY                        ***
## NAME_INCOME_TYPEPensioner               ***
## NAME_INCOME_TYPEState servant           **
## NAME_INCOME_TYPEWorking
## NAME_FAMILY_STATUSMarried               ***
## NAME_FAMILY_STATUSSeparated
```

```
## NAME_FAMILY_STATUSSingle / not married
## NAME_FAMILY_STATUSWidow                ***
## NAME_HOUSING_TYPEHouse / apartment
## NAME_HOUSING_TYPEMunicipal apartment   *
## NAME_HOUSING_TYPEOffice apartment
## NAME_HOUSING_TYPERented apartment
## NAME_HOUSING_TYPEWith parents
## FLAG_WORK_PHONE1                       *
## FLAG_PHONE1                            **
## FLAG_EMAIL1                            .
## OCCUPATION_TYPEAccountants             *
## OCCUPATION_TYPECleaning staff
## OCCUPATION_TYPECooking staff           ***
## OCCUPATION_TYPECore staff
## OCCUPATION_TYPEDrivers                 *
## OCCUPATION_TYPEHigh skill tech staff
## OCCUPATION_TYPEHR staff
## OCCUPATION_TYPEIT staff                *
## OCCUPATION_TYPELaborers                **
## OCCUPATION_TYPELow-skill Laborers
## OCCUPATION_TYPEManagers
## OCCUPATION_TYPEMedicine staff
## OCCUPATION_TYPEPrivate service staff
## OCCUPATION_TYPERealty agents
## OCCUPATION_TYPESales staff             ***
## OCCUPATION_TYPESecretaries
## OCCUPATION_TYPESecurity staff
## OCCUPATION_TYPEWaiters/barmen staff
## AGE                                    **
## YEARS_EMPLOYED                         ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4933.8  on 3590  degrees of freedom
## Residual deviance: 4069.1  on 3552  degrees of freedom
## AIC: 4147.1
##
## Number of Fisher Scoring iterations: 14
```

Visualizamos el Factor de Inflación de Varianza:

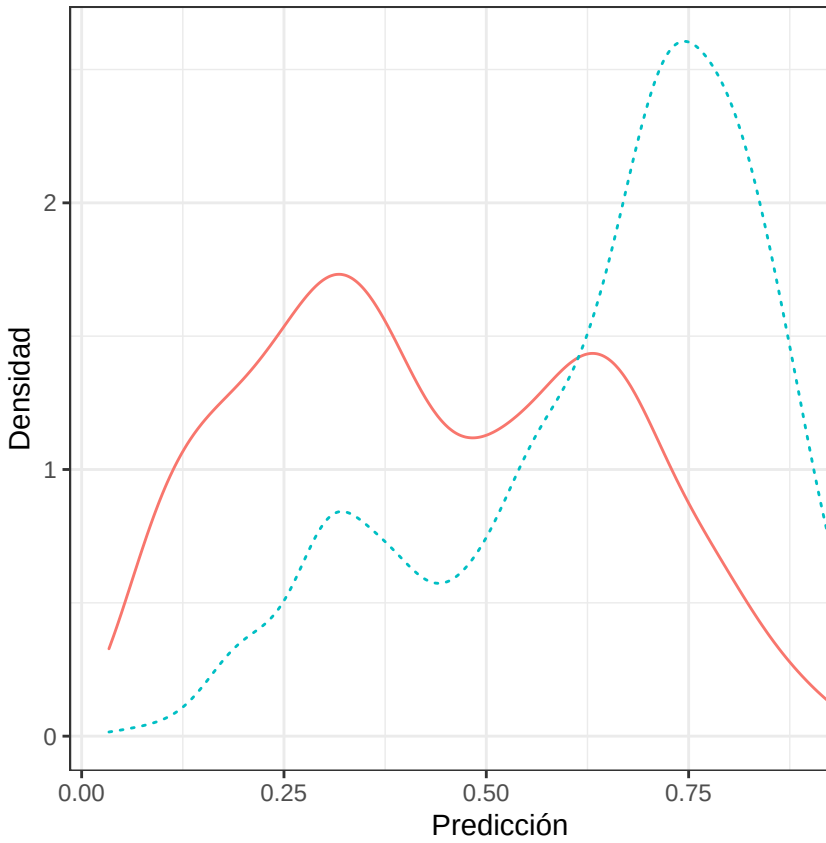
```
vif(modeloLR)
```

```
##                            CODE_GENDERM                            FLAG_OWN_CARY
##                                1.319198                                 1.148706
##                          FLAG_OWN_REALTYY                NAME_INCOME_TYPEPensioner
##                                1.137852                                 2.474546
##          NAME_INCOME_TYPEState servant                  NAME_INCOME_TYPEWorking
##                                1.241454                                 1.875468
##              NAME_FAMILY_STATUSMarried             NAME_FAMILY_STATUSSeparated
##                                3.862158                                 1.808004
## NAME_FAMILY_STATUSSingle / not married                 NAME_FAMILY_STATUSWidow
##                                2.918244                                 1.586544
```

```
##     NAME_HOUSING_TYPEHouse / apartment   NAME_HOUSING_TYPEMunicipal apartment
##                              27.083849                              14.882085
##      NAME_HOUSING_TYPEOffice apartment     NAME_HOUSING_TYPERented apartment
##                              1.000003                               4.290258
##          NAME_HOUSING_TYPEWith parents                    FLAG_WORK_PHONE1
##                             11.780371                               1.244196
##                           FLAG_PHONE1                          FLAG_EMAIL1
##                             1.087410                               1.043592
##            OCCUPATION_TYPEAccountants         OCCUPATION_TYPECleaning staff
##                             1.087151                               1.061821
##           OCCUPATION_TYPECooking staff             OCCUPATION_TYPECore staff
##                             1.177201                               1.210890
##                OCCUPATION_TYPEDrivers   OCCUPATION_TYPEHigh skill tech staff
##                             1.228323                               1.106428
##               OCCUPATION_TYPEHR staff              OCCUPATION_TYPEIT staff
##                             1.010839                               1.032053
##               OCCUPATION_TYPELaborers     OCCUPATION_TYPELow-skill Laborers
##                             1.382561                               1.037726
##               OCCUPATION_TYPEManagers         OCCUPATION_TYPEMedicine staff
##                             1.236785                               1.126244
##   OCCUPATION_TYPEPrivate service staff          OCCUPATION_TYPERealty agents
##                             1.000000                               1.036249
##            OCCUPATION_TYPESales staff            OCCUPATION_TYPESecretaries
##                             1.232990                               1.000000
##          OCCUPATION_TYPESecurity staff    OCCUPATION_TYPEWaiters/barmen staff
##                             1.085415                               1.016412
##                                  AGE                        YEARS_EMPLOYED
##                             1.752175                               1.146453
```

2.6.1 - Evaluamos el modelo

```
dfTrainLR$pred <- predict(modeloLR, newdata= dfTrainLR, type="response")
dfTestLR$pred <- predict(modeloLR, newdata= dfTestLR, type="response")
ggplot(dfTrainLR) +
  geom_density(aes(x=pred, colour=as.factor(Approve), linetype=as.factor(Approve))) +
  xlab("Predicción") +
  ylab("Densidad") +
  theme_bw()
```



2.6.1.1 - Plotear distribución de predicciones

2.6.1.2 - Matriz de Confusión

Probamos con un umbral del 50%.

```
thresh <- 0.50
predObj <-prediction(dfTrainLR$pred, dfTrainLR$Approve)
precObj <- performance(predObj, measure = "prec")
recObj <- performance(predObj, measure = "rec")
precision <- (precObj@y.values)[[1]]
prec.x <- (precObj@x.values)[[1]]
recall <- (recObj@y.values)[[1]]
pnull <- (length(which(dfTrainLR$Approve == 1)) / length(which(dfTrainLR$Approve != 2)))
pnull
```

```
## [1] 0.5555556
```

```
ctab.test <- table(prediccion=as.numeric(dfTestLR$pred > thresh), datos=dfTestLR$Approve)
ctab.test
```

```
##           datos
## prediccion    0    1
##          0   18 1049
##          1   41 3827
```

```
precision <- ctab.test[2,2]/sum(ctab.test[2,])
paste("Precision:", precision)
```

```
## [1] "Precision: 0.989400206825233"
```

```r
recall <- ctab.test[2,2]/sum(ctab.test[,2])
paste("Recall:",recall)
```

```
## [1] "Recall: 0.784864643150123"
```

```r
enrich <- precision / pnull
paste("Enrich:",enrich)
```

```
## [1] "Enrich: 1.78092037228542"
```

```r
PorcFalsosDetect <- ctab.test[1,1]/sum(ctab.test[,1])
PorcPositivosRech <- ctab.test[1,2]/sum(ctab.test[,2])
paste("% Falsos detectados: ", round(PorcFalsosDetect * 100, digits=2))
```

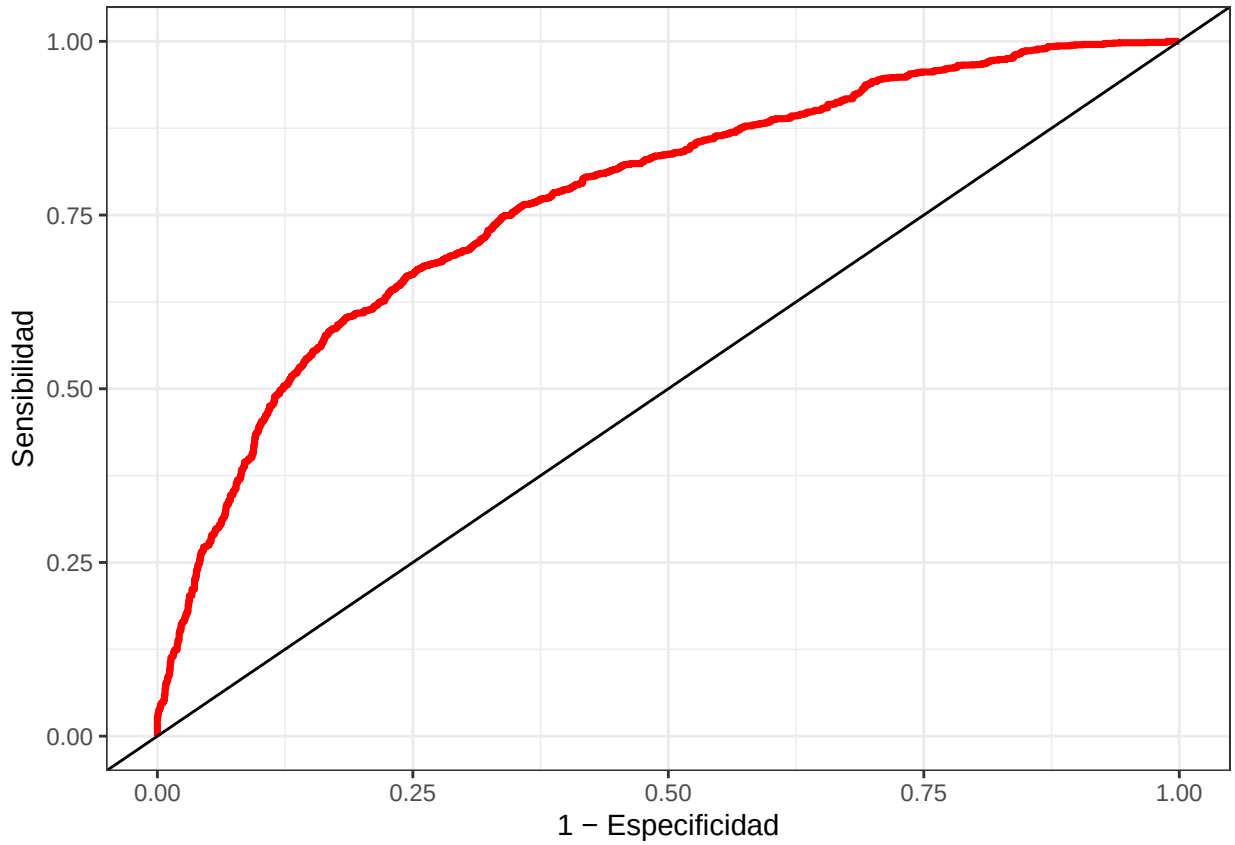
```
## [1] "% Falsos detectados:  30.51"
```

```r
paste("% Creditos buenos rechazados: ", round(PorcPositivosRech * 100, digits=2))
```

```
## [1] "% Creditos buenos rechazados:  21.51"
```

2.6.1.3 - Curva ROC Antes de decidir ajustar el umbral graficamos la curva ROC para entender las capacidades de predicción del modelo.

```r
sensObj <- performance(predObj, measure="sens")
specObj <- performance(predObj, measure="spec")
sensitivity <- (sensObj@y.values)[[1]]
specificity <- (specObj@y.values)[[1]]
dfplot <- data.frame(specm1=1-specificity, sens=sensitivity)
fig <- ggplot(dfplot, aes(x=specm1, y=sens))
fig <- fig + geom_line(color="red", size=1.25)
fig <- fig + geom_abline(aes(slope=1, intercept=0))
fig <- fig + theme_bw()
fig <- fig + xlab("1 - Especificidad")
fig <- fig + ylab("Sensibilidad")
fig
```



3 - Ajustamos el umbral del modelo seleccionado (2.6)

Seleccionamos el modelo 2.6 porque de los que mejor distribución de probabilidades y curva ROC tienen es el mas simple, o sea, que tiene menor cantidad de predictores.

3.1 - Rearmamos el modelo

```
predictores <- c("CODE_GENDER", "FLAG_OWN_CAR", "FLAG_OWN_REALTY", "NAME_INCOME_TYPE", "NAME_FAMILY_STA
formula <- paste(objetivo, paste(predictores, collapse = " + "), sep=" ~ ")
modeloLR <- glm(formula, data=dfTrainLR, family=binomial(link="logit"))
dfTrainLR$pred <- predict(modeloLR, newdata= dfTrainLR, type="response")
dfTestLR$pred <- predict(modeloLR, newdata= dfTestLR, type="response")
```

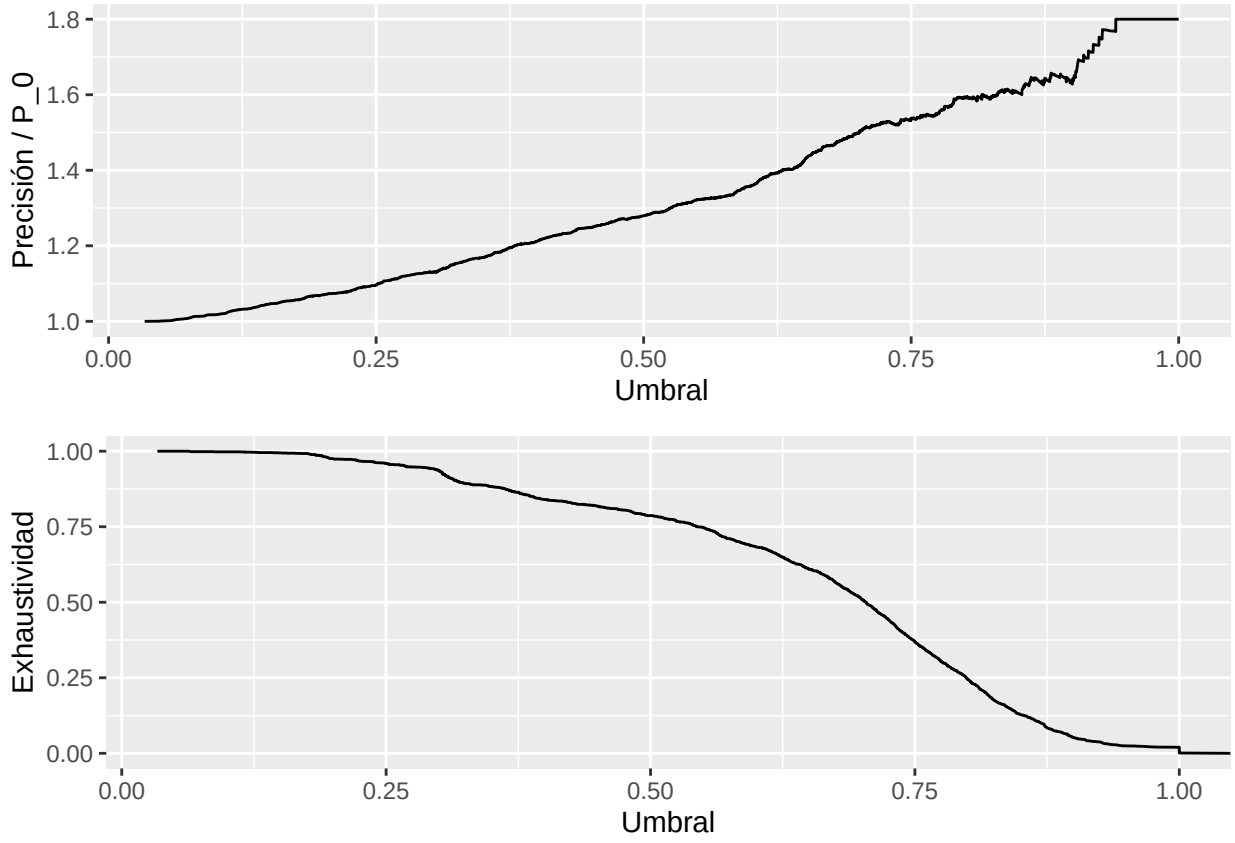
3.2 - Graficamos la precisión y la exhaustividad

```
predObj <-prediction(dfTrainLR$pred, dfTrainLR$Approve)
precObj <- performance(predObj, measure = "prec")
recObj <- performance(predObj, measure = "rec")
precision <- (precObj@y.values)[[1]]
prec.x <- (precObj@x.values)[[1]]
recall <- (recObj@y.values)[[1]]
pnull <- (length(which(dfTrainLR$Approve == 1)) / length(which(dfTrainLR$Approve != 2)))
pnull
```

```
## [1] 0.5555556
```

```r
# Data for the plot
dfplot <- data.frame(threshold=prec.x, precision=precision, recall=recall)
p1 <- ggplot(dfplot, aes(x=threshold)) +
  geom_line(aes(y=precision/pnull)) +
  xlab("Umbral") +
  ylab("Precisión / P_0")
p2 <- ggplot(dfplot, aes(x=threshold)) +
  geom_line(aes(y=recall)) +
  xlab("Umbral") +
  ylab("Exhaustividad")
nplot(list(p1, p2))
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



3.3 - Creamos la matriz de confusión

Tomamos un umbral del 50% para ir ajustando.

```r
thresh <- 0.50
ctab.test <- table(prediccion=as.numeric(dfTestLR$pred > thresh), datos=dfTestLR$Approve)
ctab.test
```

```
##           datos
## prediccion    0    1
##          0   18 1049
##          1   41 3827
```

```r
precision <- ctab.test[2,2]/sum(ctab.test[2,])
paste("Precision:", precision)
```

```
## [1] "Precision: 0.989400206825233"
```

```r
recall <- ctab.test[2,2]/sum(ctab.test[,2])
paste("Recall:",recall)
```

```
## [1] "Recall: 0.784864643150123"
```

```r
enrich <- precision / pnull
paste("Enrich:",enrich)
```

```
## [1] "Enrich: 1.78092037228542"
```

```r
PorcFalsosDetect <- ctab.test[1,1]/sum(ctab.test[,1])
PorcPositivosRech <- ctab.test[1,2]/sum(ctab.test[,2])
paste("% Falsos detectados: ", round(PorcFalsosDetect * 100, digits=2))
```

```
## [1] "% Falsos detectados:  30.51"
```

```r
paste("% Creditos buenos rechazados: ", round(PorcPositivosRech * 100, digits=2))
```

```
## [1] "% Creditos buenos rechazados:  21.51"
```

3.4 - Ajustamos el Umbral del modelo

Probamos con 45%, 55%, 60% y 65%. El umbral que más cerca está de las necesidades de negocio es del 55%.

```r
thresh <- 0.55
ctab.test <- table(prediccion=as.numeric(dfTestLR$pred > thresh), datos=dfTestLR$Approve)
ctab.test
```

```
##           datos
## prediccion    0    1
##          0   21 1258
##          1   38 3618
```

```r
precision <- ctab.test[2,2]/sum(ctab.test[2,])
paste("Precision:", precision)
```

```
## [1] "Precision: 0.989606126914661"
```

```r
recall <- ctab.test[2,2]/sum(ctab.test[,2])
paste("Recall:",recall)
```

```
## [1] "Recall: 0.742001640689089"
```

```r
enrich <- precision / pnull
paste("Enrich:",enrich)
```

```
## [1] "Enrich: 1.78129102844639"
```

```r
PorcFalsosDetect <- ctab.test[1,1]/sum(ctab.test[,1])
PorcPositivosRech <- ctab.test[1,2]/sum(ctab.test[,2])
paste("% Falsos detectados: ", round(PorcFalsosDetect * 100, digits=2))
```

```
## [1] "% Falsos detectados:  35.59"
```

```r
paste("% Creditos buenos rechazados: ", round(PorcPositivosRech * 100, digits=2))
```

```
## [1] "% Creditos buenos rechazados:  25.8"
```

Análisis de Resultados y Conclusiones

Los objetivos que nos planteó el negocio fueron 2:

- detectar 2/3 de los créditos malos
- detectar 3/4 de los créditos buenos

De acuerdo con el análisis realizado no logramos ajustar un modelo que cumpla con las necesidades de negocio propuestas.

Se observa un modelo que, si bien es un 78% mejor que el modelo nulo, a los fines de negocio tiene una muy baja detección de créditos malos y un alto rechazo de créditos buenos.

Por todo esto, entendemos que el proceso de Ciencia de Datos aplicado hasta el momento no logró un modelo que cumpla con los objetivos planteados.

Para una mejor evaluación de las capacidades predictivas del modelo, cuando se utilice para predicciones de datos no utilizados durante la etapa de entrenamiento, se podría aplicar cross-validation.

Existen otros algoritmos de clasificación que podrían probarse con el set de datos actual como Árboles de Decisión, Random Forest o Support Vector Machines, y verificar si mejoran la capacidad de predicción.

Otra opción sería tratar de conseguir un set de datos con mayor cantidad de deudores. En este set solo se tiene información de 382 casos. Como mencionamos anteriormente, la media de deudores del mercado de tarjetas de Estados Unidos es de aproximadamente 3%, lo que implicaría conseguir un set de datos mucho más grande.