

proyecto_rmd

Alexis Rangel

2023-04-05

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## vforcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyrr    1.3.0
## v purrr    1.0.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(explore)
library(tidyr)
library(gridExtra)
```

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

```
library(glue)
library(PerformanceAnalytics)
```

```
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
##
## ##### WARNING #####
## # We noticed you have dplyr installed. The dplyr lag() function breaks how #
## # base R's lag() function is supposed to work, which breaks lag(my_xts).      #
```

```

## #
## # Calls to lag(my_xts) that you enter or source() into this session won't #
## # work correctly. #
## #
## # All package code is unaffected because it is protected by the R namespace #
## # mechanism. #
## #
## # Set 'options(xts.warn_dplyr_breaks_lag = FALSE)' to suppress this warning. #
## #
## # You can use stats::lag() to make sure you're not using dplyr::lag(), or you #
## # can add conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop #
## # dplyr from breaking base R's lag() function. #
## ##### WARNING #####
## #
## Attaching package: 'xts'
##
## The following objects are masked from 'package:dplyr':
##
##     first, last
##
## Attaching package: 'PerformanceAnalytics'
##
## The following object is masked from 'package:graphics':
##
##     legend

library(dataxray)
library(correlationfunnel)

## == correlationfunnel Tip #2 =====
## Clean your NA's prior to using 'binarize()' .
## Missing values and cleaning data are critical to getting great correlations. :)

library(caret)

## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift

library(ROCR)
library(olsrr)

##
## Attaching package: 'olsrr'
##
## The following object is masked from 'package:datasets':
##
##     rivers

```

```

library(leaps)
library(boot)

## 
## Attaching package: 'boot'
##
## The following object is masked from 'package:lattice':
## 
##     melanoma

library(Amelia)

## Loading required package: Rcpp
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.8.1, built: 2022-11-18)
## ## Copyright (C) 2005-2023 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##

load("~/Maestria/1er semestre/Modelos lineales/proyecto/DatosParaProyecto.RData")

```

Objetivo y alcances del proyecto

Descripción de la base

```

# View(Train)
# names(Train)

Train_copy = Train
Test_copy = Test
# nrow(Train_copy)

```

La base de datos consta de 18 variable, una variable a predecir y 17 candidatas a predictoras. 34209 registros de cada variable.

Sobre la base:

1. Loan.Status (VARIABLE DICOTOMA A PREDECIR, VALORES POSIBLES -> [Fully Paid, Charged Off])

Nombre variable | tipo de variable | num valores posibles | valores posibles

2. Term | nominal | 2 | ["Long Term", "Short Term"]
3. Years.in.current.job | ordinal | 12 | ["n/a", "< 1 year", "1 year": "9 years", "10+ years"]
4. Home.Ownership | nominal | 4 | ["Rent", "Home Mortgage", "Own Home", "HaveMortgage"]
5. Purpose | nominal | 16 | [...] >15 valores posibles...] | « Requiere homologar como factor "other" and "Other" »
6. Current.Loan.Amount | continua | [min -> 21449.74 , mean -> 14044734.38, max -> 100000002.5]

7. Credit.Score | continua | [min -> 584.00, mean -> 1048.48, max -> 7509.0]
8. Annual.Income | continua | [min -> 164596.55, mean -> 1441110.95, max -> 30838993.9]
9. Monthly.Debt | continua | [min -> 0.00, mean -> 19025.56, max -> 229056.4]
10. Years.of.Credit.History | continua | [min -> 2.30, mean -> 19.21, max -> 59.6]
11. Months.since.last.delinquent | continua | [min -> 0.00, mean -> 34.96, max -> 178.1] (Meses desde el último moroso)
12. Number.of.Open.Accounts | continua | [min -> 0.00, mean -> 11.43, max -> 49.0]
13. Number.of.Credit.Problems | continua | [min -> 0.00, mean -> 0.53, max -> 15.0]
14. Current.Credit.Balance | continua | [min -> 0.00, mean -> 259532.60, max -> 7140732.6]
15. Maximum.Open.Credit | continua | [min -> 0.00, mean -> 666270.83, max -> 798255369.7]
16. Bankruptcies | continua | [min -> 0.00, mean -> 0.46, max -> 7.0]
17. Tax.Liens | continua | [min -> 0.00, mean -> 0.42, max -> 14.0] (gravámenes fiscales)
18. ID — excluir variable de cualquier análisis —

— agrupando variables para el análisis grafico univariado —

Conteos: nominal - 3 ordinal - 1 continua - 8

Sobre la variable dependiente y de acuerdo con la siguiente pagina web:

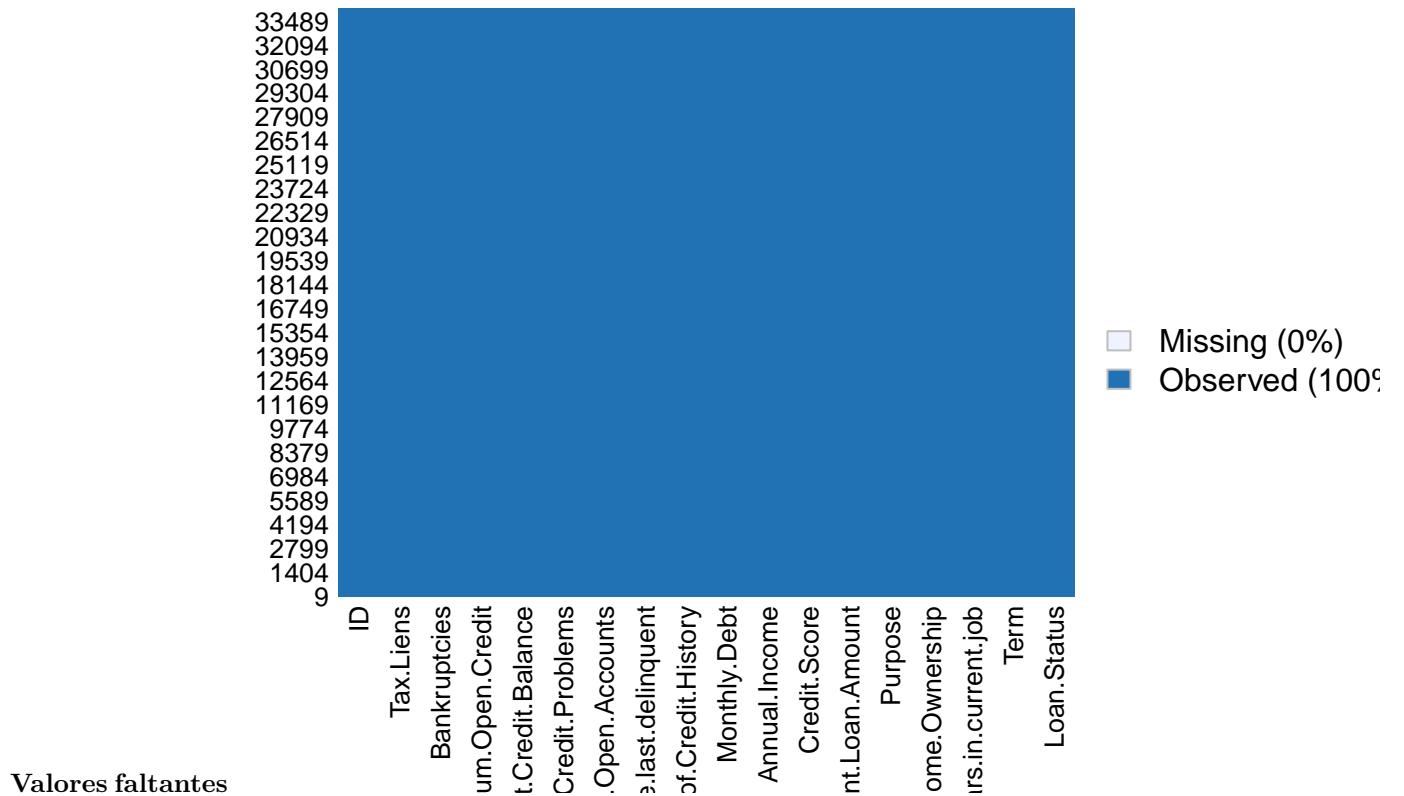
A charge-off is the opposite of paid in full. It means the lender hasn't received payment for at least 180 days, and the account is in default. The lender, or a third-party collection agency, can still come after this kind of debt. Charge-offs have an extremely negative effect on your credit score.

Fuente: <https://budgeting.thenest.com/account-paid-full-vs-chargeoff-23884.html>

Análisis exploratorio

```
missmap(Train_copy, main = "Missing values vs observed")
```

Missing values vs observed

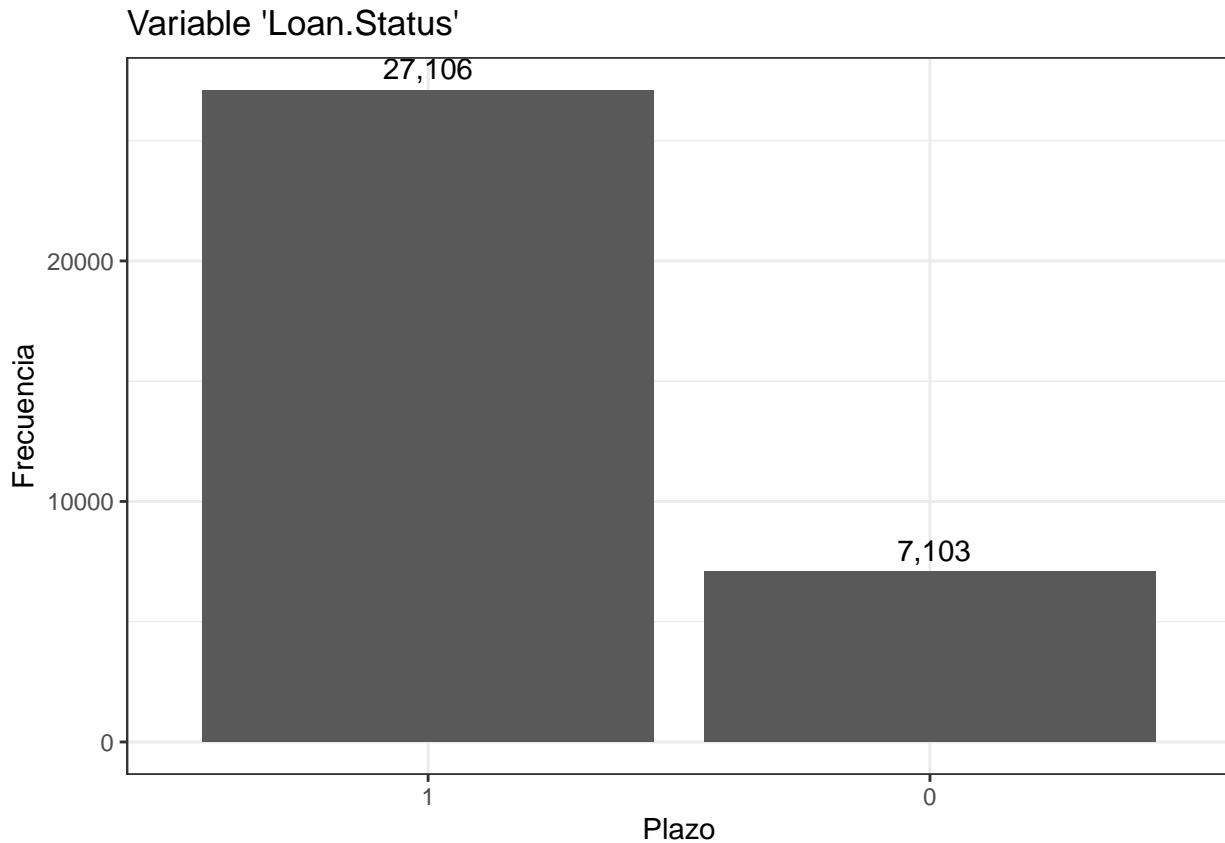


Análisis univariado Inicialmente pasamos a binario el pago (Fully Paid) o el default (Charged Off) con un 1 y un 0 respectivamente.

```
Train_copy$Loan.Status = ifelse(Train_copy$Loan.Status == "Fully Paid", 1, 0)
Train_copy$Loan.Status = as.factor(Train_copy$Loan.Status)
```

Loan.Status

```
ggplot(Train_copy, aes(fct_infreq(Loan.Status)))+
  geom_bar(stat="count")+
  labs(title = "Variable 'Loan.Status'", 
       x = "Plazo",
       y = "Frecuencia") +
  theme_bw() +
  # Agregar el número de observaciones al filo de cada barra
  geom_text(stat = "count", aes(label = after_stat(format(count, big.mark = ","))), vjust = -0.5)
```

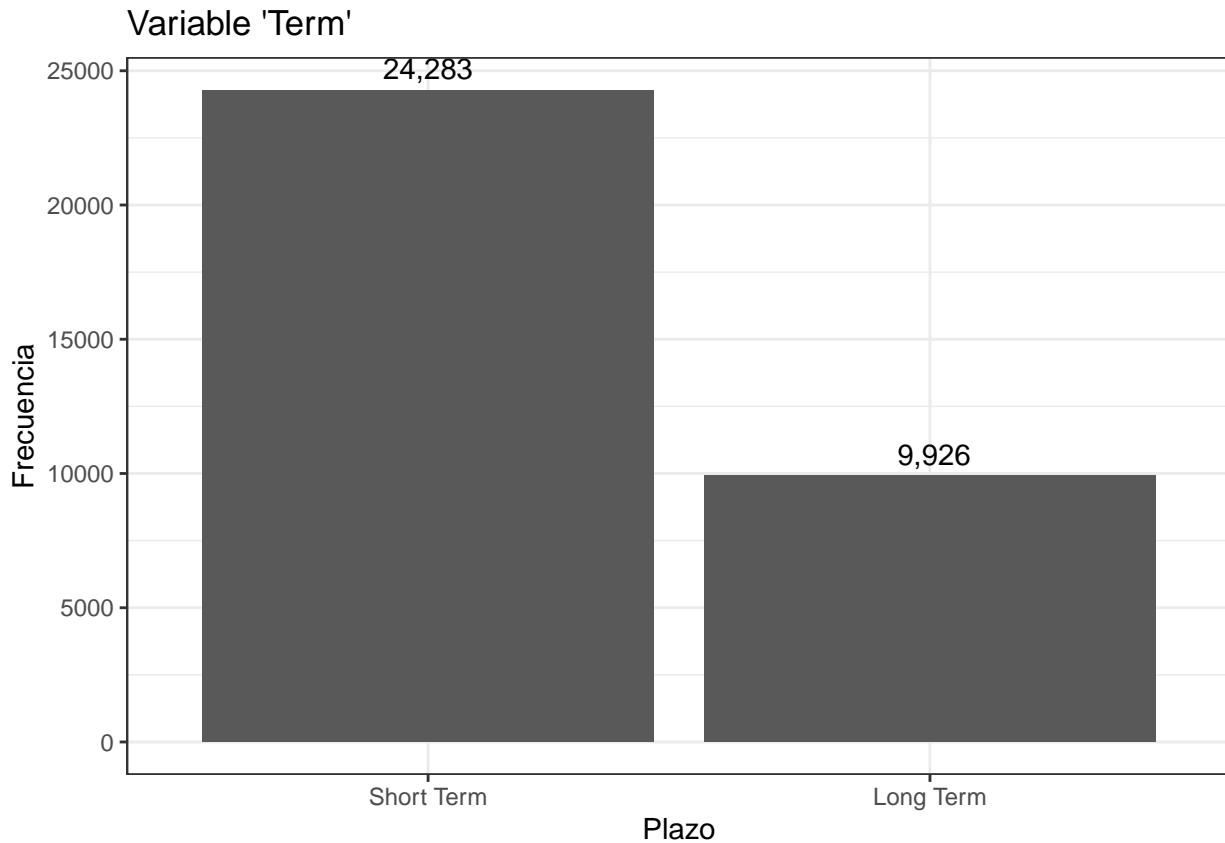


Análisis univariado variables nominales y ordinales Paso numero 1, pasar a factores todas estas variables nominales y ordinales

a. Term

```
#class(Train_copy$Term)
Train_copy$Term = as.factor(Train_copy$Term)

ggplot(Train_copy, aes(fct_infreq(Term))+
  geom_bar(stat="count")+
  labs(title = "Variable 'Term'", 
       x = "Plazo",
       y = "Frecuencia") +
  theme_bw() +
  # Agregar el número de observaciones al filo de cada barra
  geom_text(stat = "count", aes(label = after_stat(format(count, big.mark = ","))), vjust = -0.5)
```



b. Years.in.current.job

```
Train_copy$Years.in.current.job = as.factor(Train_copy$Years.in.current.job)

ggplot(Train_copy, aes(fct_infreq(Years.in.current.job)))+
  geom_bar(stat="count")+
  labs(title = "Variable 'Years.in.current.job'",  

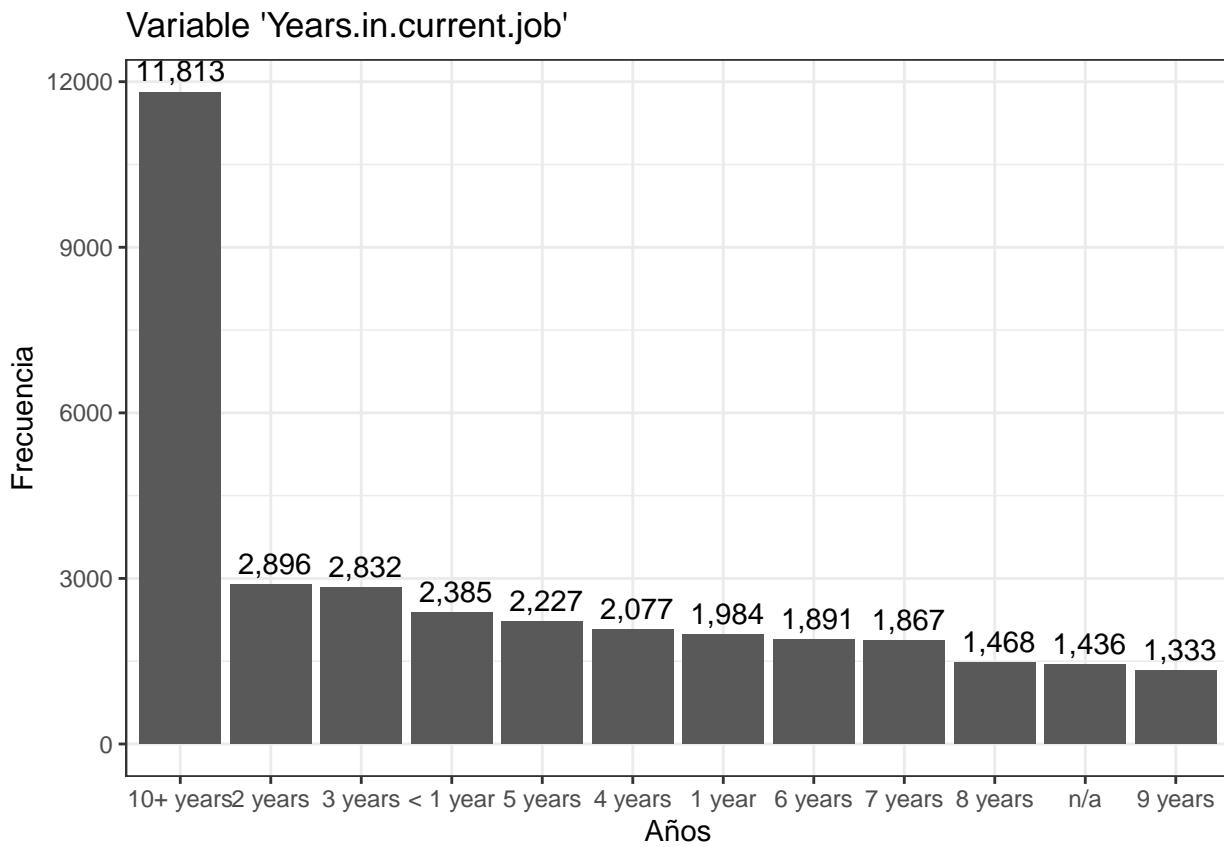
       x = "Años",  

       y = "Frecuencia") +  

  theme_bw() +  

  # Agregar el número de observaciones al filo de cada barra  

  geom_text(stat = "count", aes(label = after_stat(format(count, big.mark = ","))), vjust = -0.5)
```



c. Home.Ownership

```
Train_copy$Home.Ownership = as.factor(Train_copy$Home.Ownership)

ggplot(Train_copy, aes(fct_infreq(Home.Ownership)))+
  geom_bar()+
  labs(title = "Variable 'Home.Ownership'",  

       x = "Tipo de dueño",  

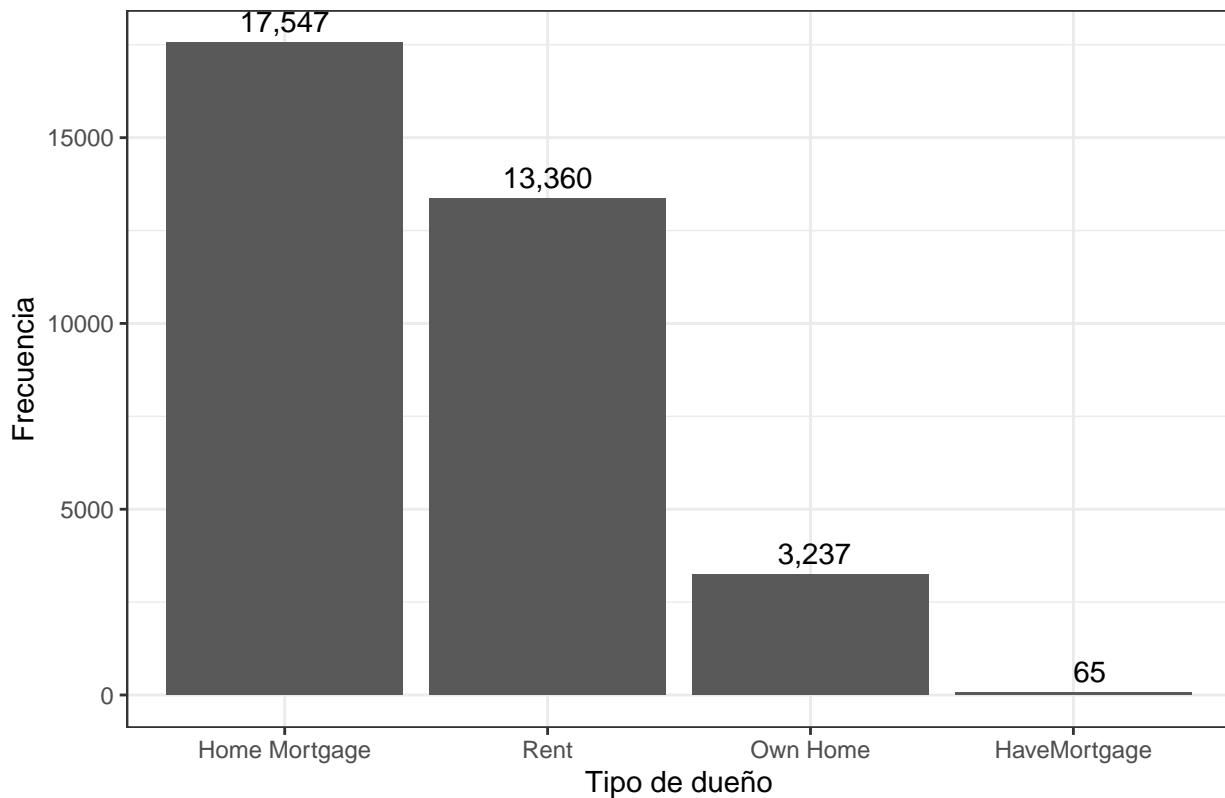
       y = "Frecuencia") +  

  theme_bw() +  

  # Agregar el número de observaciones al filo de cada barra  

  geom_text(stat = "count", aes(label = after_stat(format(count, big.mark = ","))), vjust = -0.5)
```

Variable 'Home.Ownership'



d. Purpose

Quizás valdría la pena reducir grupos

Por un lado, se unifica el valor de la variable "Other" y "other" en purpose con la función tolower(), y por otro, se valorará en el ajuste del modelo la reducción de dimensiones en Train_copy

```
unique(Train_copy$Purpose)
```

```
## [1] "Debt Consolidation"      "Home Improvements"       "other"
## [4] "Medical Bills"          "Other"                  "Take a Trip"
## [7] "Business Loan"           "Buy a Car"                "Buy House"
## [10] "major_purchase"          "small_business"          "moving"
## [13] "vacation"                 "wedding"                "Educational Expenses"
## [16] "renewable_energy"
```

```
Train_copy$Purpose = tolower(Train_copy$Purpose)
```

```
#unique(Train_copy$Purpose)
```

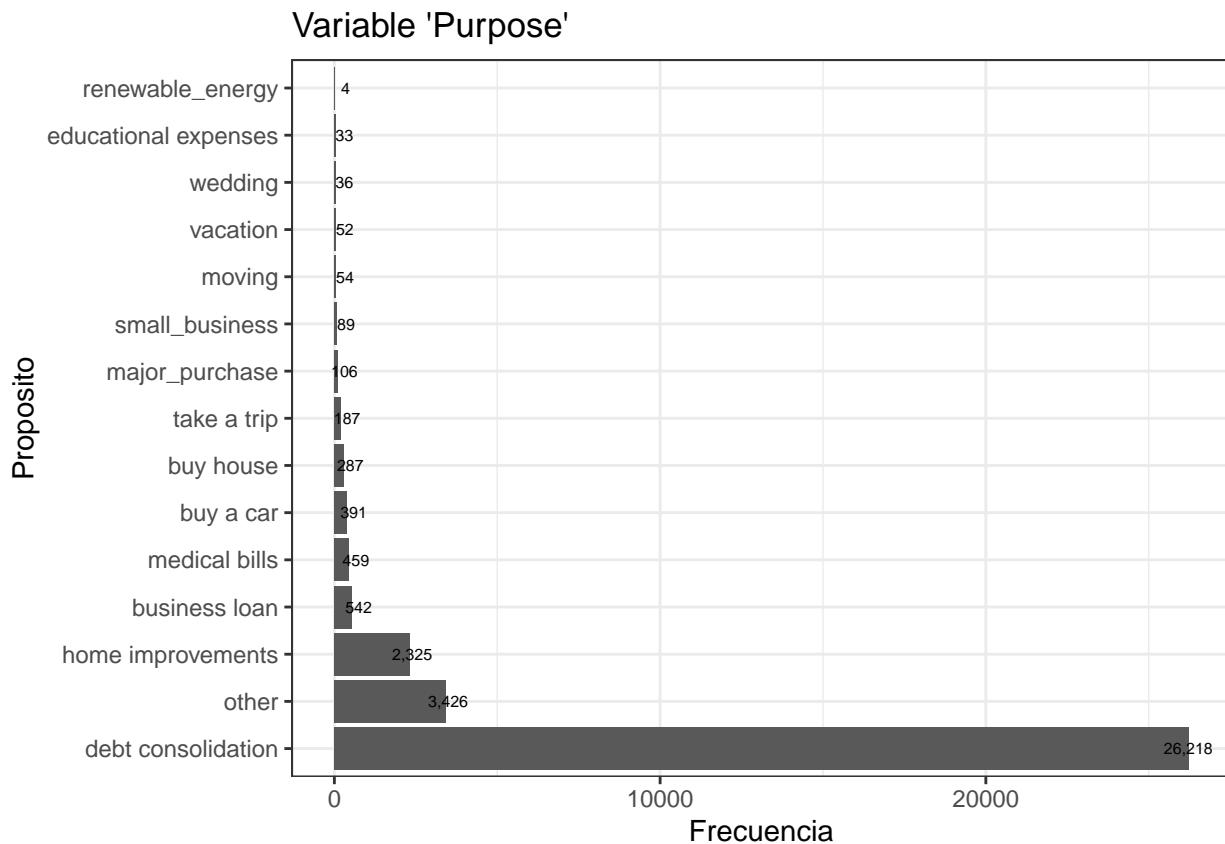
```
Train_copy$Purpose = as.factor(Train_copy$Purpose)
```

```
ggplot(Train_copy, aes(fct_infreq(Purpose)))+
  geom_bar(stat="count")+
  labs(title = "Variable 'Purpose'",  
x = "Propósito",
```

```

y = "Frecuencia") +
theme_bw() +
# Agregar el número de observaciones al filo de cada barra
geom_text(stat = "count", aes(label = after_stat(format(count, big.mark = ","))), size=2) +
coord_flip()

```



```

#unique(Train_copy$Purpose)
# # Cambiar "hombre" por 1 y "mujer" por 2
# nuevo_vector <- ifelse(observaciones == "hombre", 1, ifelse(observaciones == "mujer", 2, observacione

```

Análisis univariado y agrupación grafica de variables continuas Dado que son varias variables continuas y probablemente no tenga sentido meter todas en un solo gráfico de boxplot, buscamos rangos similares para segmentar en 2 o 3 grupos. Esto requiere inicialmente meter todas en un boxplot y manualmente separarlas

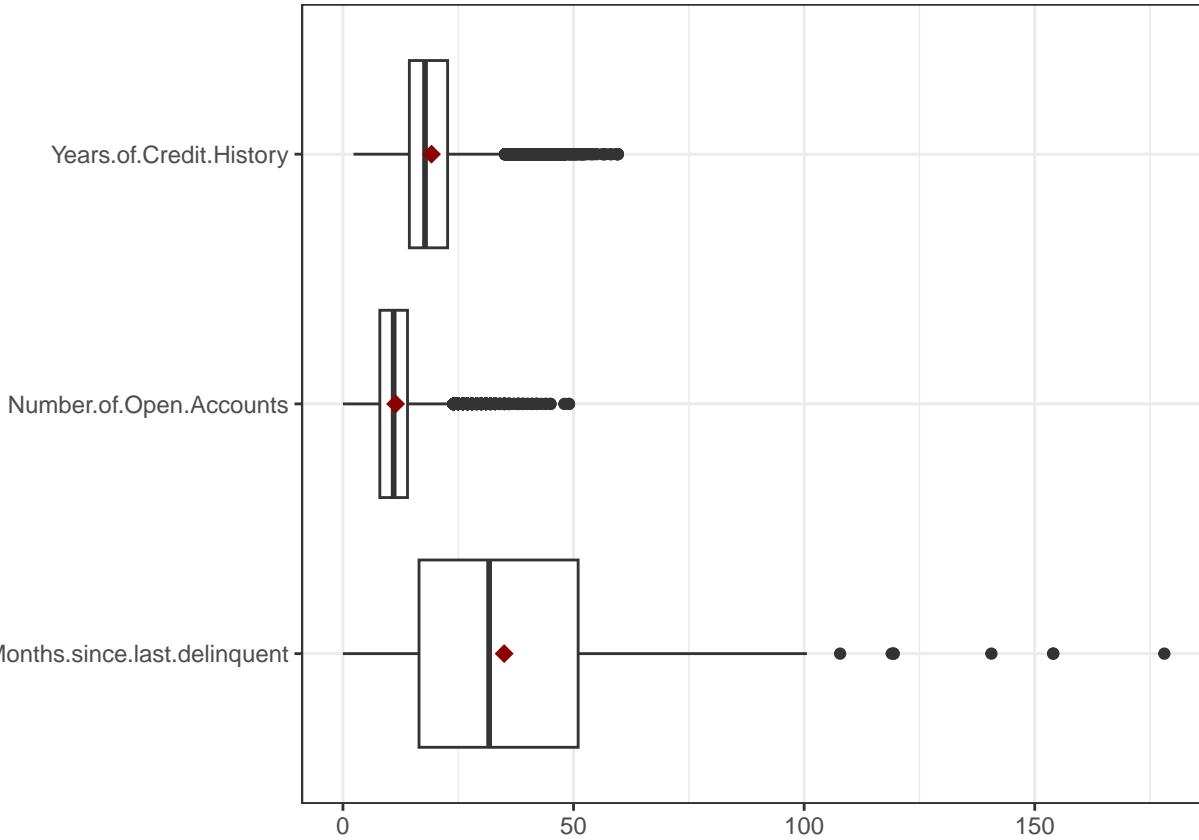
```

Train_copy_continuas = Train_copy %>% select(Current.Loan.Amount, Credit.Score, Annual.Income, Monthly.I
train_long_form = gather(select(Train_copy_continuas, -c(Maximum.Open.Credit, Current.Loan.Amount, Annual.
ggplot(train_long_form, aes(y=variable, x=valor))+
  geom_boxplot()+
  # geom_jitter()+
  stat_summary(fun=mean, colour="darkred", geom="point",
              shape=18, size=3, show.legend=FALSE)+
```

```

scale_x_continuous(NULL) +
scale_y_discrete(NULL) +
theme_bw()

```



2do grupo : - Maximum.Open.Credit - Current.Loan.Amount -

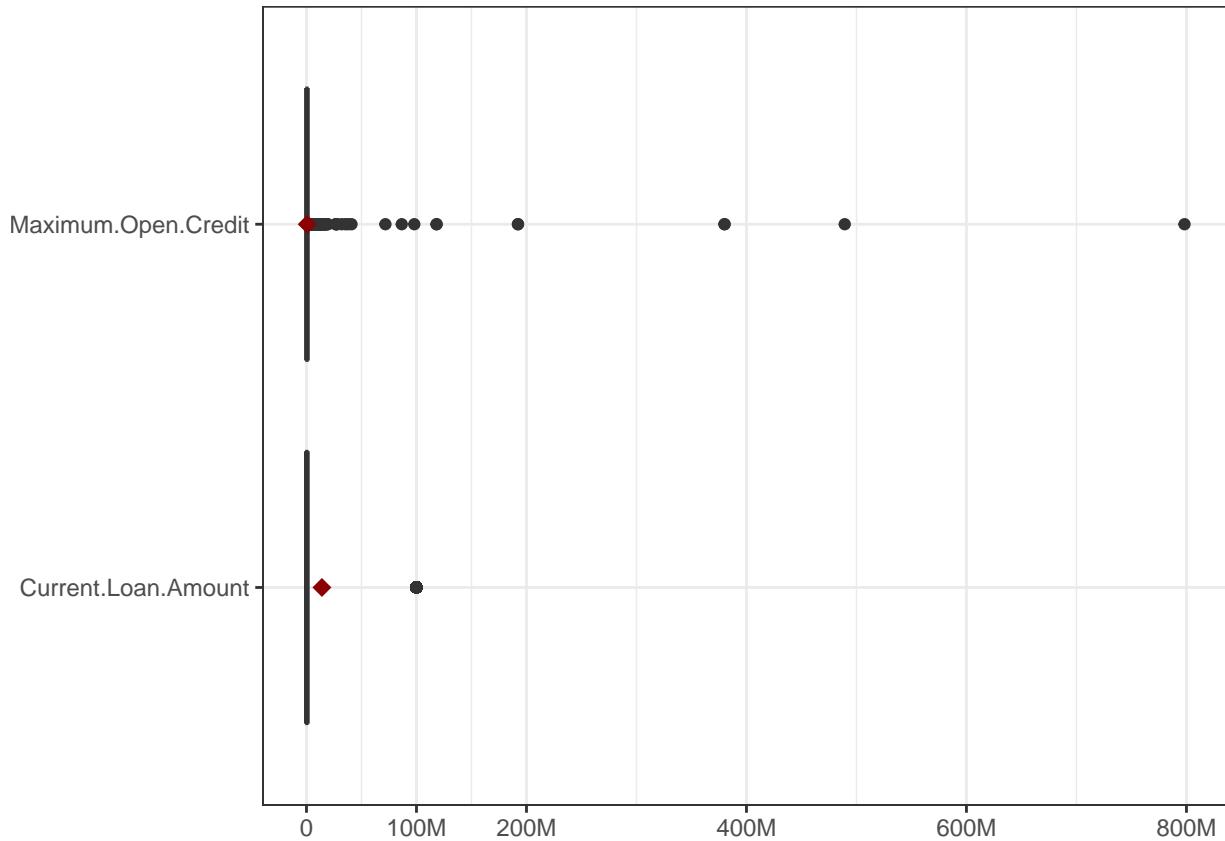
```

# nrow(Train_copy_continuas)
# df.temp = filter(Train_copy_continuas, Maximum.Open.Credit < 697531)
# nrow(df.temp)

train_selected = select(Train_copy_continuas, Maximum.Open.Credit, Current.Loan.Amount)
train_long_form1 = gather(train_selected, key = "variable", value = "valor")

ggplot(train_long_form1, aes(y=variable, x=valor))+
  geom_boxplot()+
  # geom_jitter()+
  stat_summary(fun=mean, colour="darkred", geom="point",
              shape=18, size=3, show.legend=FALSE)+
  scale_x_continuous(NULL, breaks = c(0, 100000000, 200000000, 400000000, 600000000, 800000000),
                     labels = c("0", "100M", "200M", "400M", "600M", "800M"))+
  scale_y_discrete(NULL) +
  theme_bw()

```

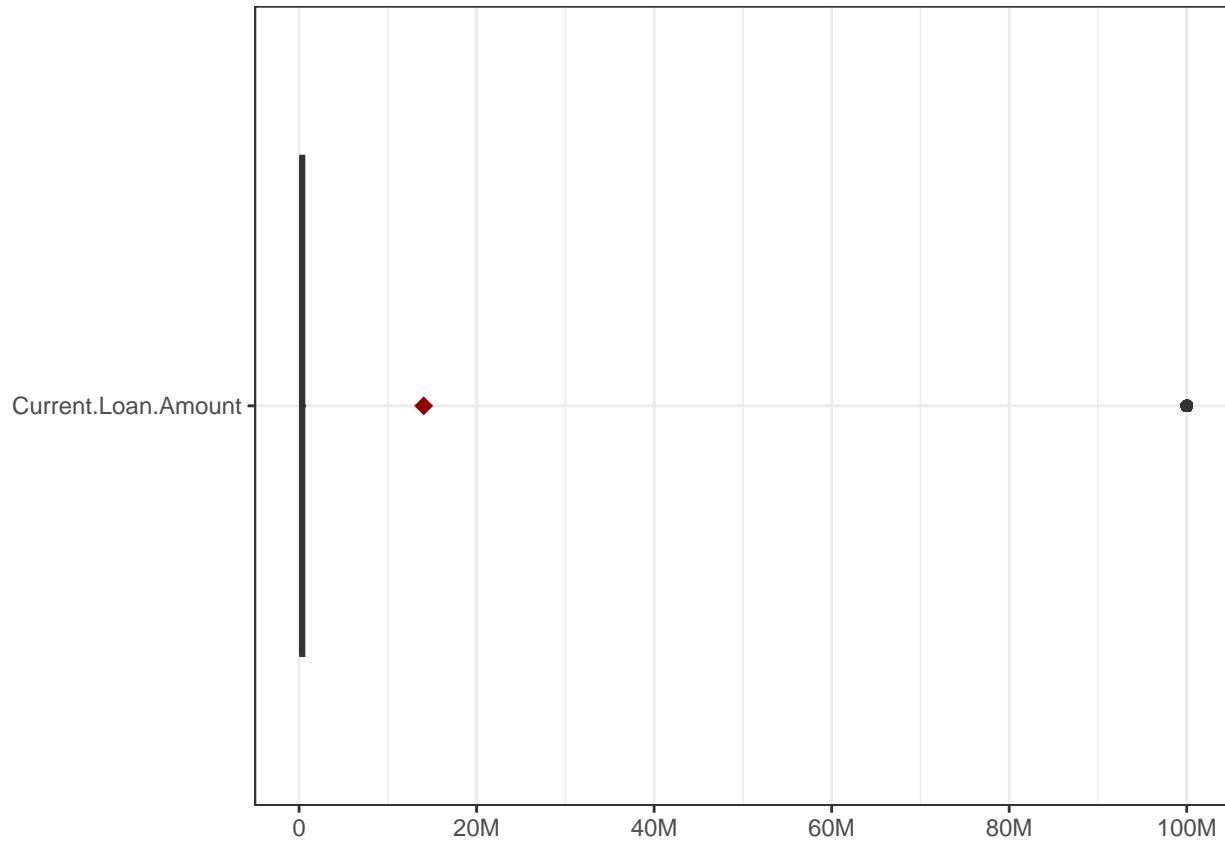


Se sospecha visualmente que en Current.Loan.Amount existe un dato outlier muy marcado cercano a los 100M, pero dicho outlier, no solo es un dato sino que pertenece a un subconjunto que representa el 4to cuartil con 4712 obs (13.77% del total).

Por lo tanto se vuelve caso de estudio analizar este subgrupo de outliers para evaluar la permanencia de estos registros en el ajuste del modelo.

```
train_selected = select(Train_copy_continuas, Current.Loan.Amount)
train_long_form1 = gather(train_selected, key = "variable", value = "valor")

ggplot(train_long_form1, aes(y=variable, x=valor))+
  geom_boxplot()+
  # geom_jitter()+
  stat_summary(fun=mean, colour="darkred", geom="point",
              shape=18, size=3, show.legend=FALSE)+
  scale_x_continuous(NULL, breaks = c(0, 20000000, 40000000, 60000000, 80000000, 100000000),
                     labels = c("0", "20M", "40M", "60M", "80M", "100M"))+
  scale_y_discrete(NULL)+
  theme_bw()
```



Visualizamos los cuartiles

```
print("Primer cuartil")

## [1] "Primer cuartil"

quantile(Train_copy_continuas$Current.Loan.Amount, 0.25)

##      25%
## 186337.5

print("-----")

## [1] "-----"

print("Segundo cuartil")

## [1] "Segundo cuartil"

quantile(Train_copy_continuas$Current.Loan.Amount, 0.5)

##      50%
## 324302.7
```

```

print("-----")

## [1] "-----"

quantile(Train_copy_continuas$Current.Loan.Amount, 0.75)

##      75%
## 544016.2

print("Tercer cuartil")

## [1] "Tercer cuartil"

quantile(Train_copy_continuas$Current.Loan.Amount, 1)

## 100%
## 1e+08

```

En este punto cabe destacar que el 75% de los valores del vector Current.Loan.Amount son menores o iguales a 544016.2.

Dado que parece que los valores del cuarto cuartil son tan extremos para sugerirnos filtrarlos del estudio, pero en conteo representan 8,551 observaciones, se decide hacer un estudio de la distribución de los datos en el cuarto cuartil para evaluar si se pueden recuperar algunas observaciones. Es claro que un modelo no se va ajustar apropiadamente a registros con valores de cantidad de deuda actual de 500 mil y de 90 millones en una de las variables regresoras, y si se ajusta, no será el mejor modelo o la variable perderá significancia estadística en el modelo.

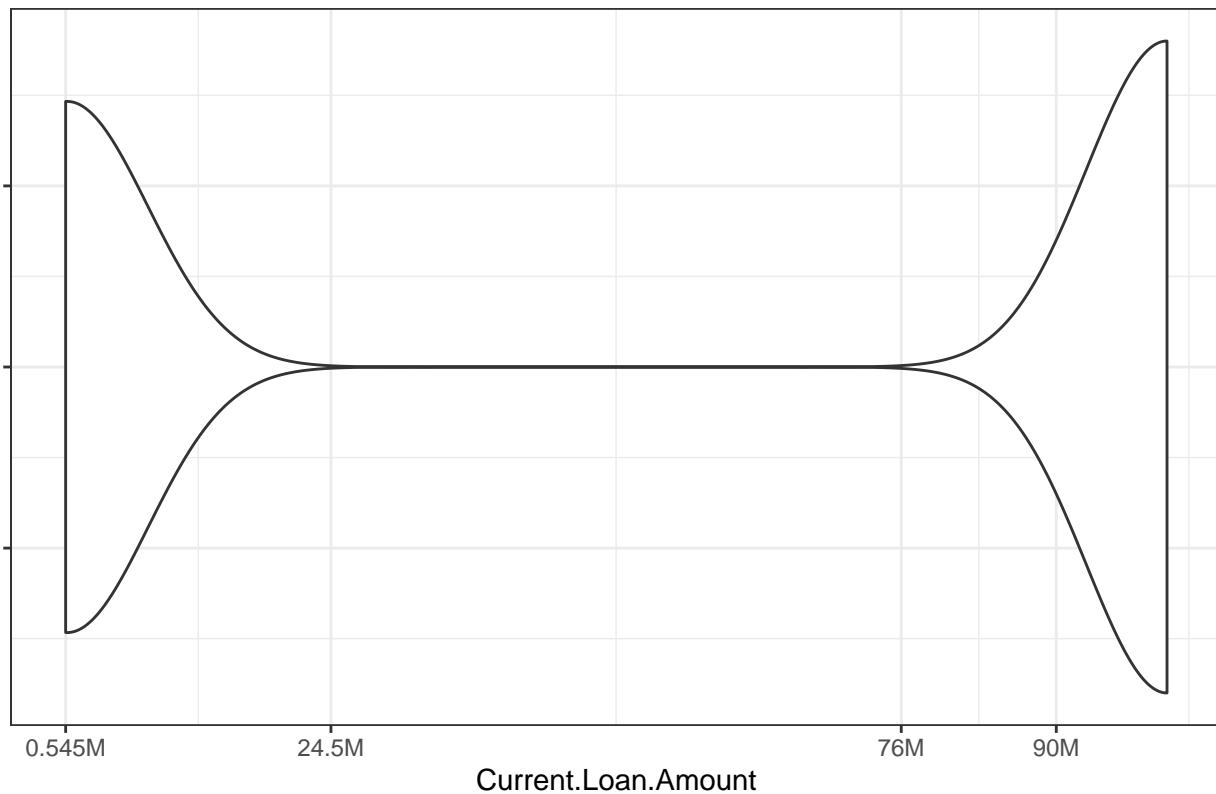
En el análisis del cuarto cuartil, que son valores mayores a 544016.2, se puede observar la siguiente distribución de los datos con ayuda de un gráfico de violin:

```

df_temp = filter(Train_copy_continuas, Current.Loan.Amount > 544016.2 )
ggplot(df_temp, aes(y=1,x=Current.Loan.Amount))+
  geom_violin()+
  scale_x_continuous(breaks = c(545000, 24500000, 76000000, 90000000),
                     labels = c("0.545M", "24.5M", "76M", "90M"))+
  scale_y_continuous(NULL, labels = NULL)+
  ggtitle("Análisis del 4to cuartil de la variable Current.Loan.Amount (valores > 544,016.2)")+
  theme_bw()

```

Análisis del 4to cuartil de la variable Current.Loan.Amount (valores > 544,016.2)

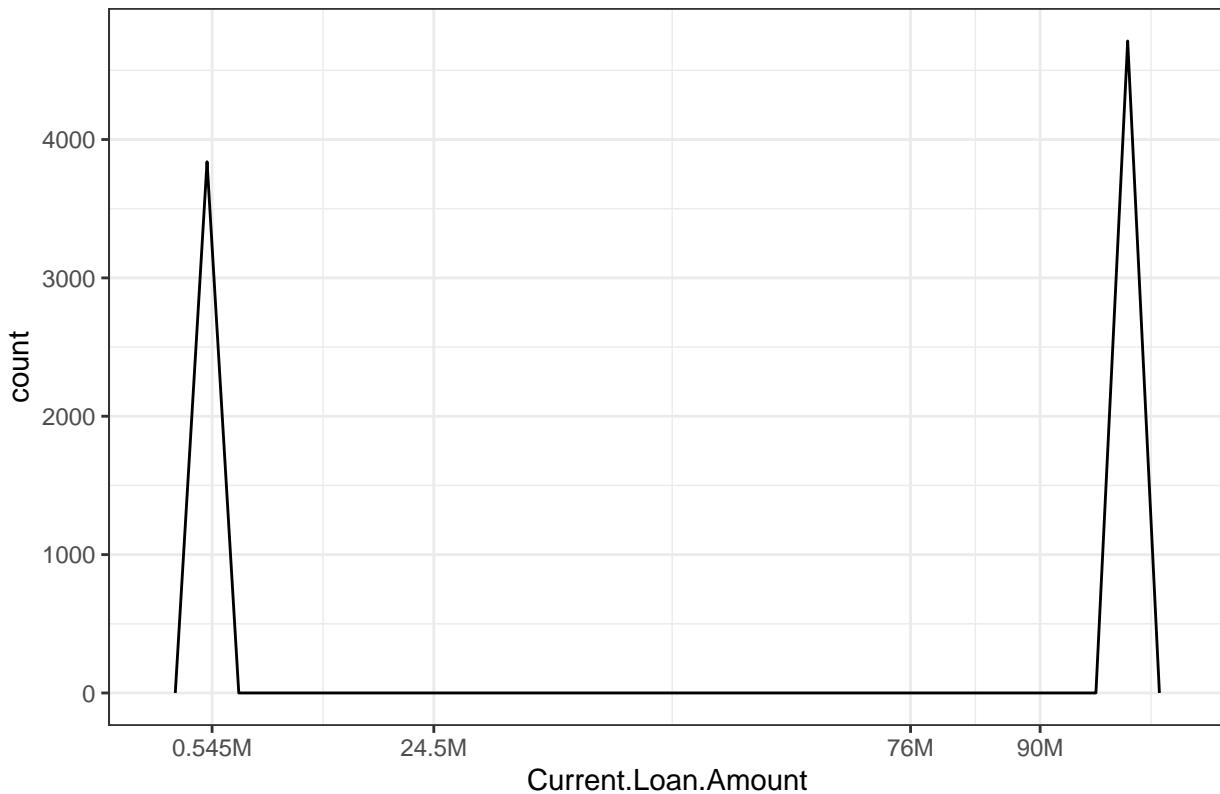


Se observa una distribución de datos que presenta un vacío entre 24.5M y 76M, también apreciable con un gráfico de densidad.

```
df_temp = filter(Train_copy_continuas, Current.Loan.Amount > 544016.2 )
ggplot(df_temp, aes(Current.Loan.Amount))+
  geom_freqpoly()+
  scale_x_continuous(breaks = c(545000, 24500000, 76000000, 90000000),
                     labels = c("0.545M", "24.5M", "76M", "90M"))+
  ggtitle("Análisis del 4to cuartil de la variable Current.Loan.Amount (valores > 544,016.2)")+
  theme_bw()

## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```

Análisis del 4to cuartil de la variable Current.Loan.Amount (valores > 544,

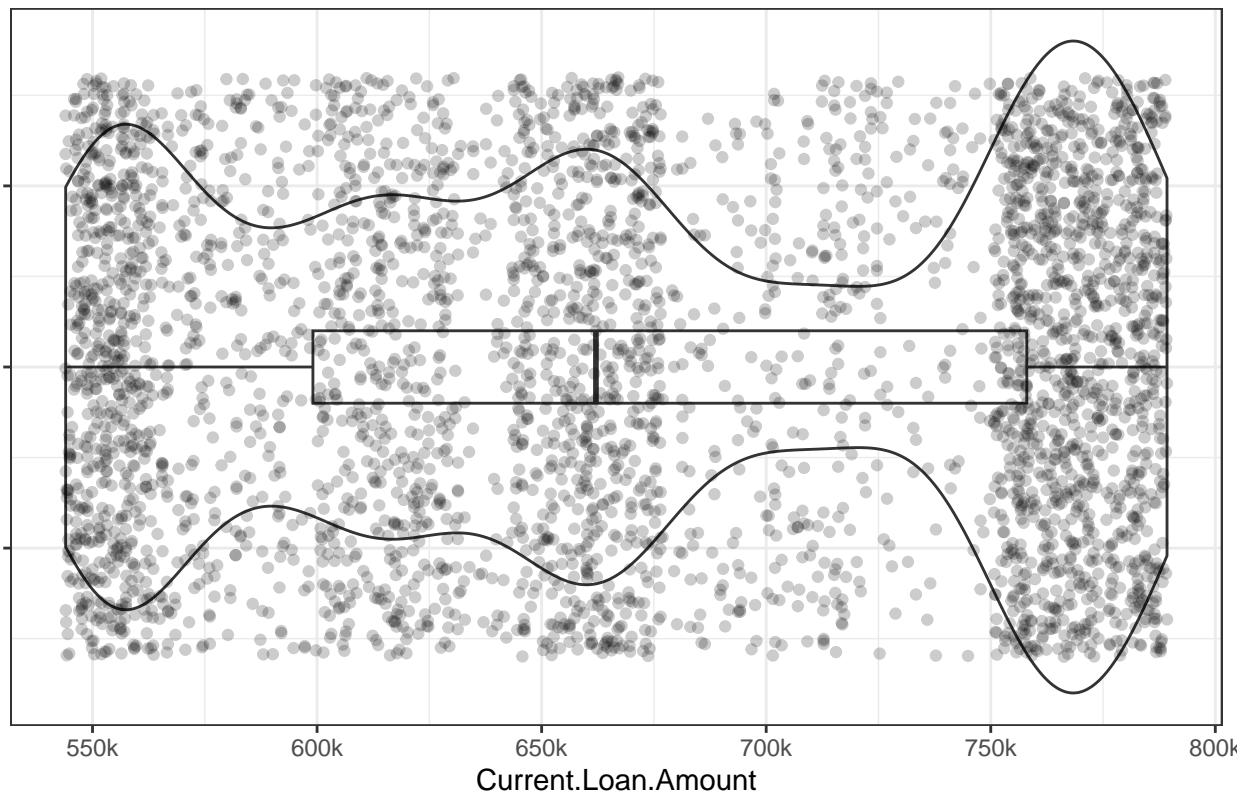


Y dado que los valores de la izquierda de la curva bimodal no son tan extremos como los de la parte derecha, nos planteamos la idea de tomar los de la izquierda y filtrar los valores extremos de la derecha. Dichos valores de la curva izquierda bimodal toman la siguiente distribución, con valor mínimo de 544,016.2 y máximo de 24.5 M, como no lo indica la anterior gráfica, que en conteo son 3,839 observaciones y tienen la siguiente distribución.

```
df_temp = filter(Train_copy_continuas, Current.Loan.Amount > 544016.2 )
df_temp = filter(df_temp, Current.Loan.Amount < 24500000 )

ggplot(df_temp, aes(y=1, x=Current.Loan.Amount))+
  geom_violin()+
  geom_boxplot(width = 0.1, fill = "white", alpha = 0.5)+
  geom_jitter(alpha=0.2)+
  scale_y_continuous(NULL, labels = NULL)+
  scale_x_continuous(breaks = c(550000, 600000, 650000, 700000, 750000, 800000),
                     labels = c("550k", "600k", "650k", "700k", "750k", "800k"))+
  ggtitle("Análisis del 4to cuartil de la variable Current.Loan.Amount (valores > 544,016.2)")+
  theme_bw()
```

Análisis del 4to cuartil de la variable Current.Loan.Amount (valores > 544,016.2)



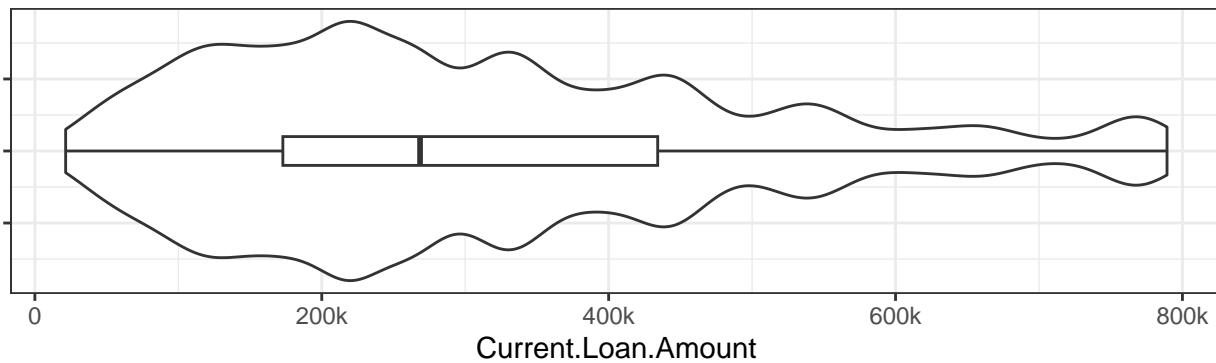
Por lo que en conclusión, en el análisis univariado de la variable `Current.Loan.Amount` y al poder observar la cota superior aproximada a la que debemos delimitarla para restringir valores extremos, el modelo se ajustara con todas aquellas observaciones que sean menores o iguales a 800,000 (por redondeo), que representaran un 86.2% de las observaciones de esta variable y que adoptan la siguiente distribución.

```
plot_1 = ggplot(filter(Train_copy_continuas, Current.Loan.Amount <= 80000000), aes(y=1, x=Current.Loan.Amount))+
  geom_violin() +
  geom_boxplot(width = 0.1, fill = "white", alpha = 0.5) +
  scale_y_continuous(NULL, labels = NULL) +
  scale_x_continuous(breaks = c(0, 200000, 400000, 600000, 800000),
                     labels = c("0", "200k", "400k", "600k", "800k")) +
  ggtitle("Variable Current.Loan.Amount posterior al filtro (valores < 80M)") +
  theme_bw()

plot_2 = ggplot(Train_copy_continuas, aes(y=1, x=Current.Loan.Amount)) +
  geom_violin() +
  geom_boxplot(width = 0.1, fill = "white", alpha = 0.5) +
  scale_y_continuous(NULL, labels = NULL) +
  scale_x_continuous(breaks = c(0, 20000000, 40000000, 60000000, 80000000, 100000000),
                     labels = c("0", "20M", "40M", "60M", "80M", "100M")) +
  ggtitle("Variable Current.Loan.Amount previo al filtro") +
  theme_bw()

grid.arrange(plot_1, plot_2, ncol = 1)
```

Variable Current.Loan.Amount posterior al filtro (valores < 80M)



Variable Current.Loan.Amount previo al filtro



Finalmente, se procede a recopilar los IDs de dichos registros para al finalizar el análisis exploratorio, hacer el filtro del dataset inicial ya que podría darse el caso que en otras variables se ajuste otro filtro.

Por otro y sin ahonda mucho en la variable `Maximum.Open.Credit`, para eliminar valores extremos en el mismo sentido de la variable previa `Current.Loan.Amount` se establece la cota superior en 80M. Esto bajo la lógica de que, si el tope máximo de la deuda actual es 80M es porque tienes un crédito otorgado máximo por esta cantidad. Este filtro posiblemente cambie en busqueda de un mejor ajuste del modelo, dado que el análisis de cuartiles nos indica que el 3er cuartil de `Maximum.Open.Credit` es un valor menor o igual a 697,531, sin embargo la lógica antes mencionada impera hasta este momento de análisis.

```
print("Primer cuartil")

## [1] "Primer cuartil"

quantile(Train_copy_continuas$Maximum.Open.Credit, 0.25)

##      25%
## 246619.9

print("-----")

## [1] "-----"

print("Segundo cuartil")
```

```

## [1] "Segundo cuartil"

quantile(Train_copy_continuas$Maximum.Open.Credit, 0.5)

##      50%
## 417515.8

print("-----")

## [1] "-----"

quantile(Train_copy_continuas$Maximum.Open.Credit, 0.75)

##      75%
## 697531

print("Tercer cuartil")

## [1] "Tercer cuartil"

quantile(Train_copy_continuas$Maximum.Open.Credit, 1)

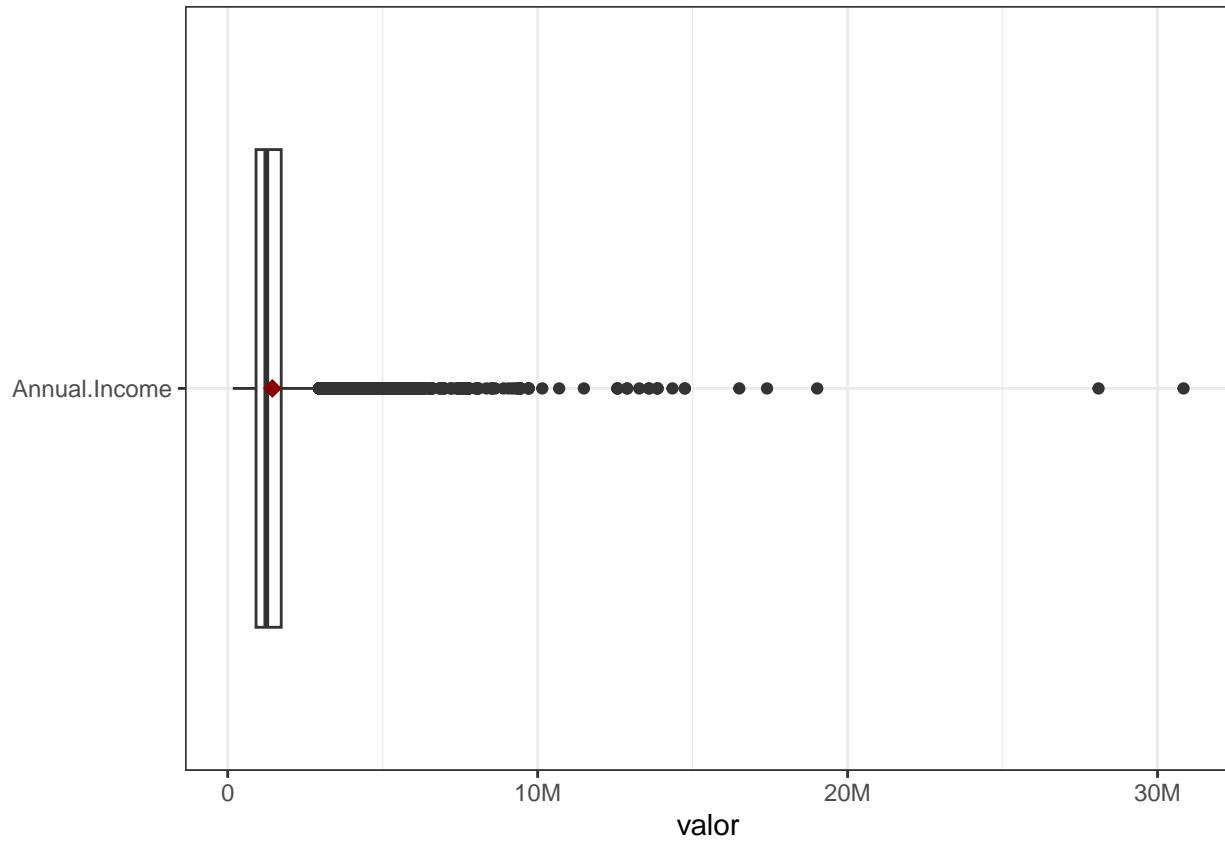
##      100%
## 798255370

3er grupo: - Annual.Income

train_selected = select(Train_copy_continuas, Annual.Income)
train_long_form2 = gather(train_selected, key = "variable", value = "valor")

ggplot(train_long_form2, aes(y=variable, x=valor))+
  geom_boxplot()+
  #geom_jitter(alpha=0.2)+
  stat_summary(fun=mean, colour="darkred", geom="point",
              shape=18, size=3, show.legend=FALSE)+
  scale_y_discrete(name=NULL)+
  scale_x_continuous(breaks = c(0, 10000000, 20000000, 30000000),
                     labels = c("0", "10M", "20M", "30M"))+
  theme_bw()

```

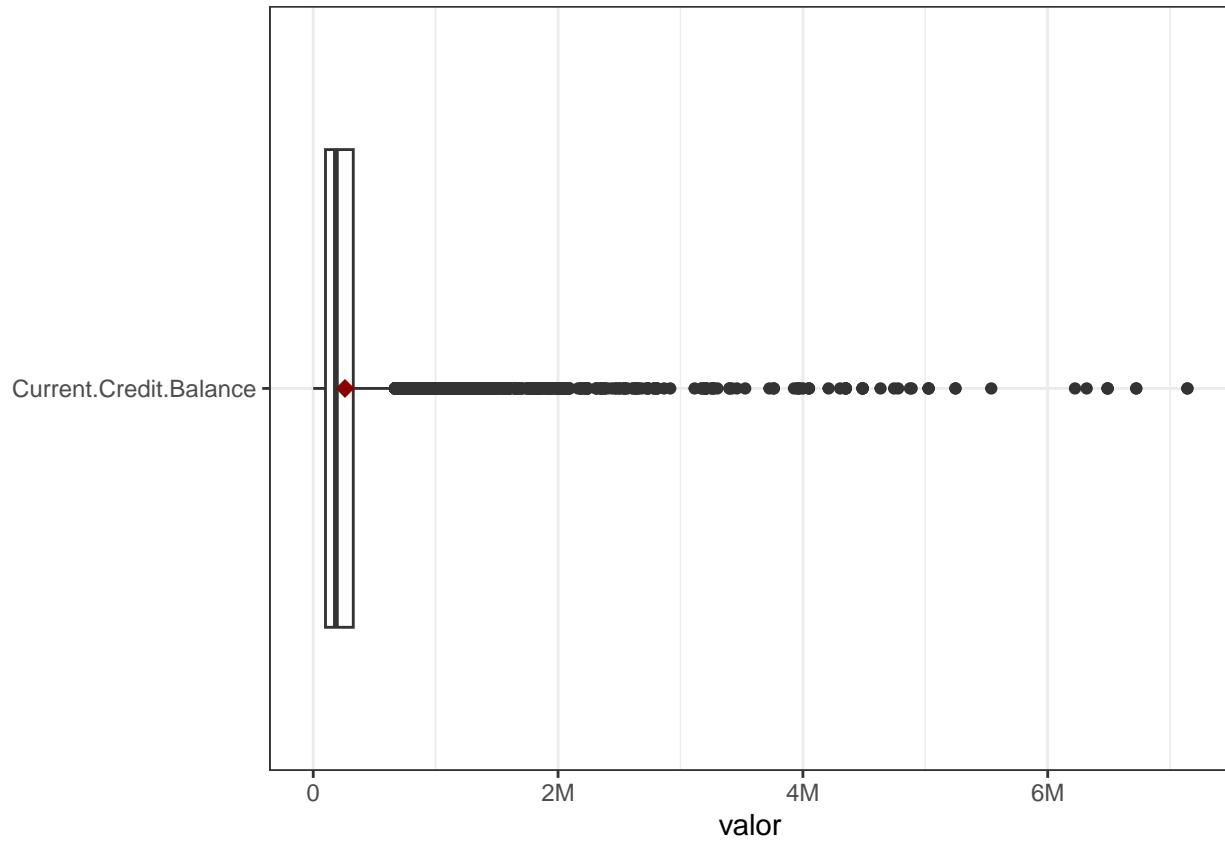


4to grupo:

- Current.Credit.Balance

```
train_selected = select(Train_copy_continuas, Current.Credit.Balance)
train_long_form3 = gather(train_selected, key = "variable", value = "valor")

ggplot(train_long_form3, aes(y=variable, x=valor))+
  geom_boxplot()+
  # geom_jitter()+
  stat_summary(fun=mean, colour="darkred", geom="point",
              shape=18, size=3, show.legend=FALSE)+
  scale_y_discrete(name=NULL)+
  scale_x_continuous(breaks = c(0, 2000000, 4000000, 6000000),
                     labels = c("0", "2M", "4M", "6M"))+
  theme_bw()
```

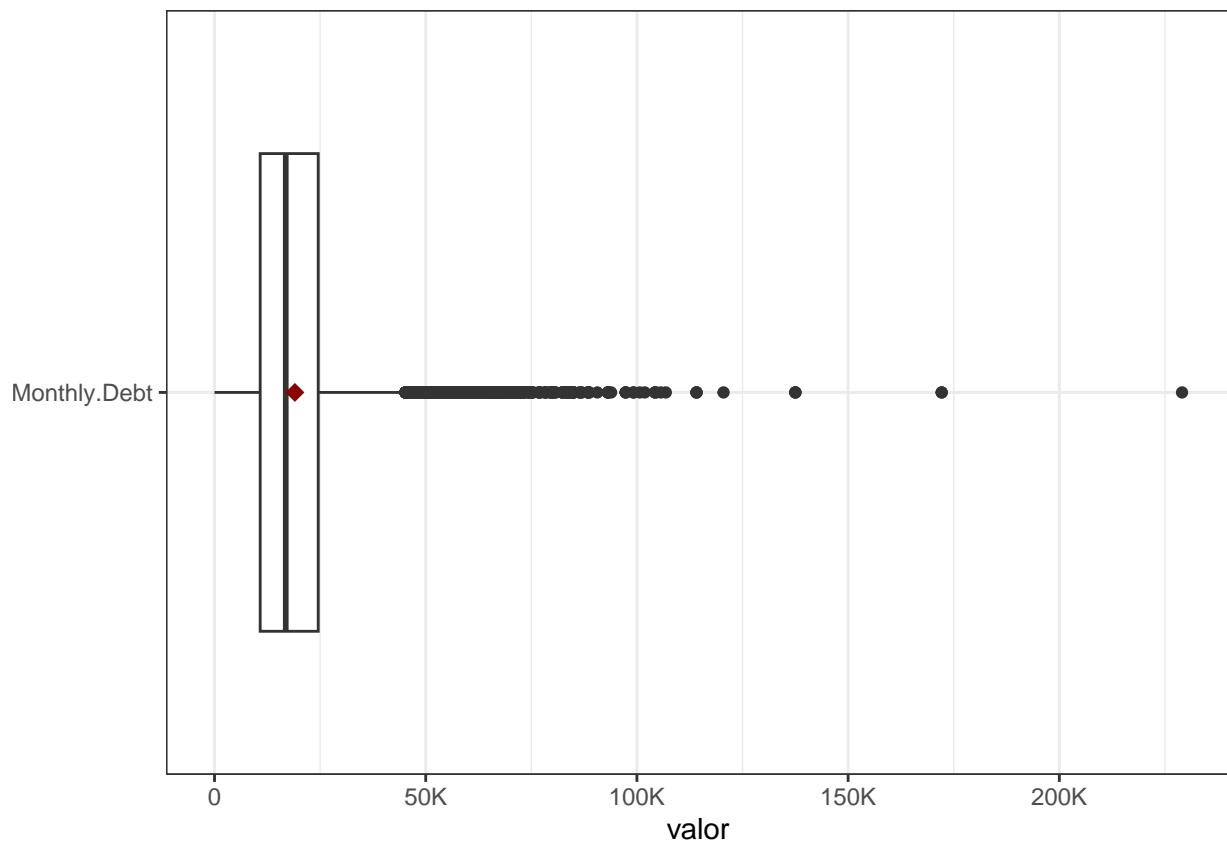


5to grupo:

- Monthly.Debt

```
train_selected = select(Train_copy_continuas, Monthly.Debt)
train_long_form4 = gather(train_selected, key = "variable", value = "valor")

ggplot(train_long_form4, aes(y=variable, x=valor))+
  geom_boxplot()+
  # geom_jitter()+
  stat_summary(fun=mean, colour="darkred", geom="point",
              shape=18, size=3, show.legend=FALSE)+
  scale_y_discrete(name=NULL)+
  scale_x_continuous(breaks = c(0, 50000, 100000, 150000, 200000),
                     labels = c("0", "50K", "100K", "150K", "200K"))+
  theme_bw()
```

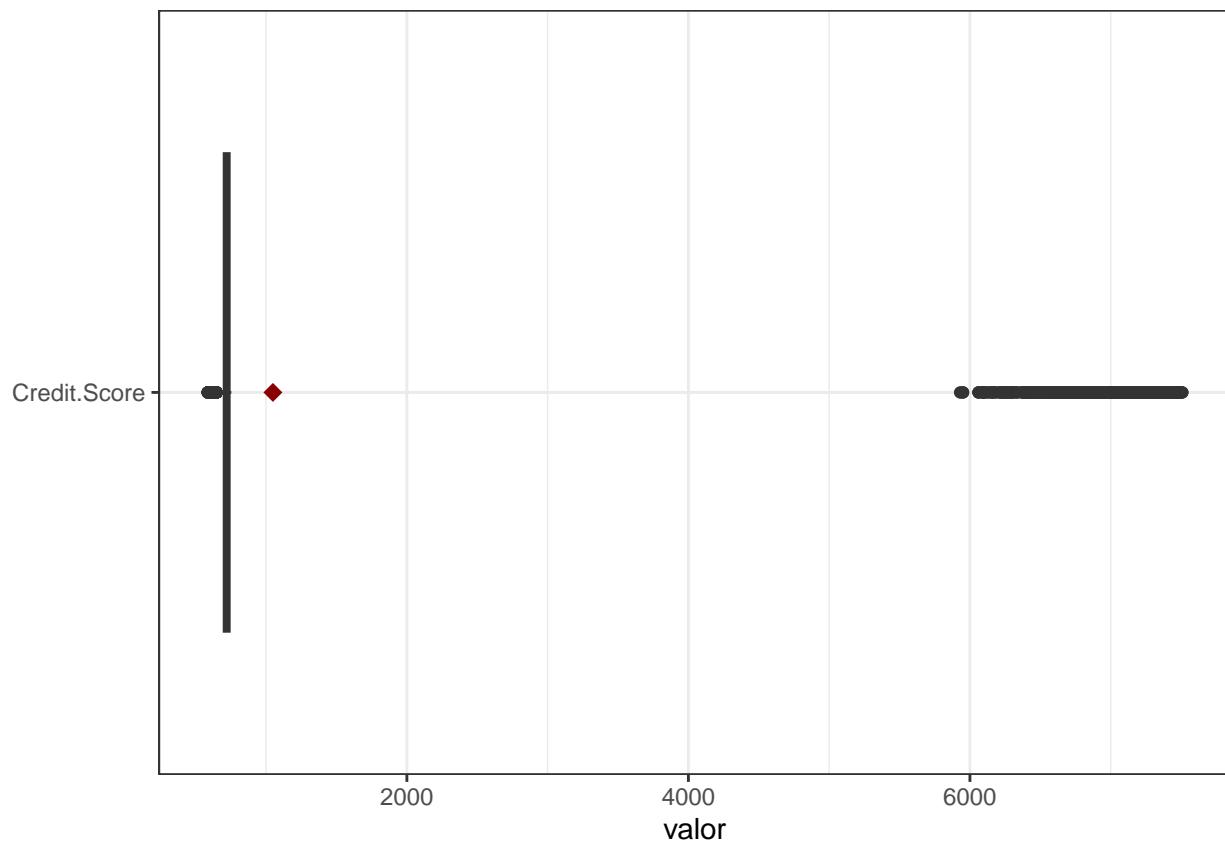


6to grupo: - Credit.Score

```

train_selected = select(Train_copy_continuas, Credit.Score)
train_long_form5 = gather(train_selected, key = "variable", value = "valor")

ggplot(train_long_form5, aes(y=variable, x=valor))+
  geom_boxplot()+
  # geom_jitter()+
  stat_summary(fun=mean, colour="darkred", geom="point",
              shape=18, size=3, show.legend=FALSE)+
  scale_y_discrete(name=NULL)+
  theme_bw()
  
```

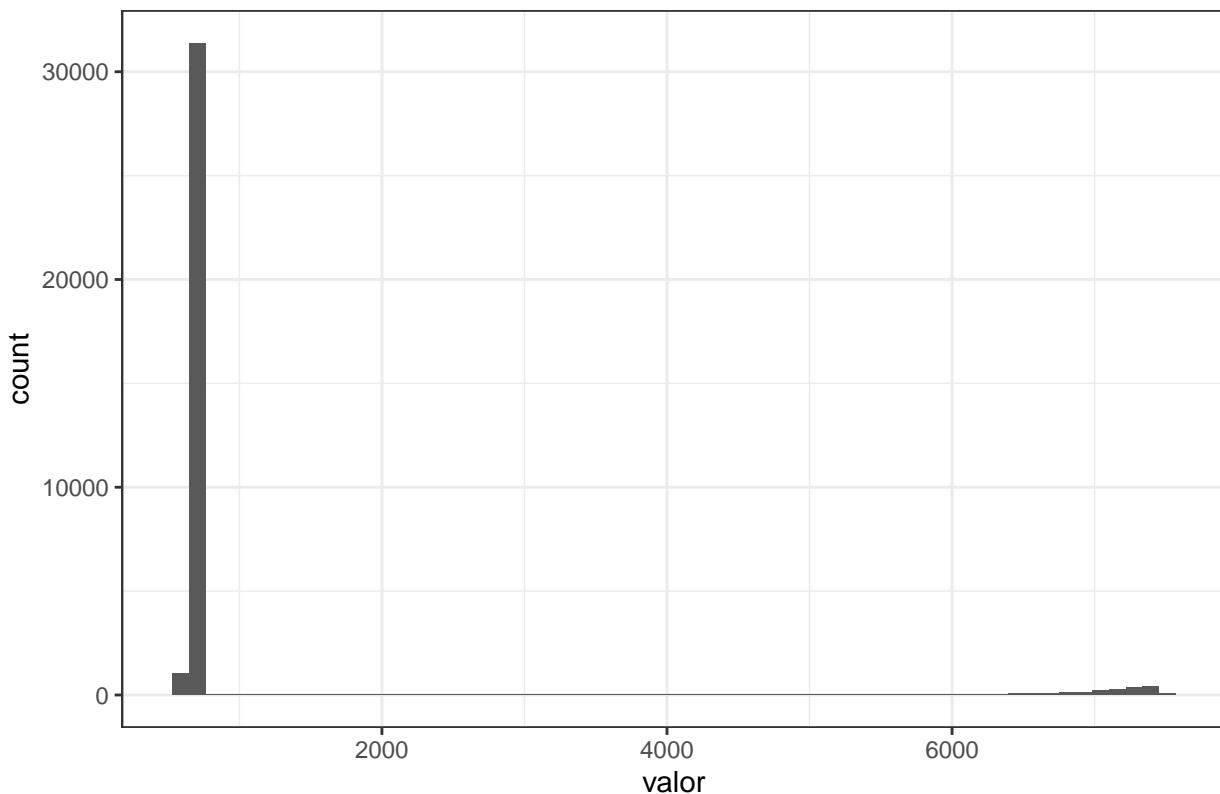


```
#  
# mean(Train_copy_continuas$Credit.Score)
```

Un análisis mediante un histógrafo nos indica que se presenta un subgrupo de valores extremos mayores a 2mil en la variable **Credit.Score**

```
ggplot(train_long_form5, aes(valor))+  
  geom_histogram(bins = 60)+  
  ggtitle("Histógrafo Credit.Score") +  
  theme_bw()
```

Histógrama Credit.Score



La situación de la distribución de los valores parece ser un poco similar al caso de la variable `Current.Loan.Amount`, por lo que se buscara filtrar los registros de aquellas observaciones extremas para la variable `Credit.Score` en estudio. Cuando la media el score crediticio se encuentra en 1048.483, los valores de esta variable mayores a 5mil empiezan a perder mucho sentido, generandnos dudas sobre si estos datos se recopilaron apropiadamente o son inexactos como para ponerles un tope.

Por un lado se analizan los cuartiles del vector y se elaboran dos gráficas de cajas y bigotes más una capa de violin, además de un histógrama: una para valores que se encuentre en el primer, segundo y tercer cuartil y otra para el cuarto cuartil.

```
print("Primer cuartil")

## [1] "Primer cuartil"

quantile(Train_copy_continuas$Credit.Score, 0.25)

## 25%
## 702

print("-----")

## [1] "-----"

print("Segundo cuartil")

## [1] "Segundo cuartil"
```

```

quantile(Train_copy_continuas$Credit.Score, 0.5)

## 50%
## 721

print("-----")

## [1] "-----"

quantile(Train_copy_continuas$Credit.Score, 0.75)

## 75%
## 738

print("Tercer cuartil")

## [1] "Tercer cuartil"

quantile(Train_copy_continuas$Credit.Score, 1)

## 100%
## 7509

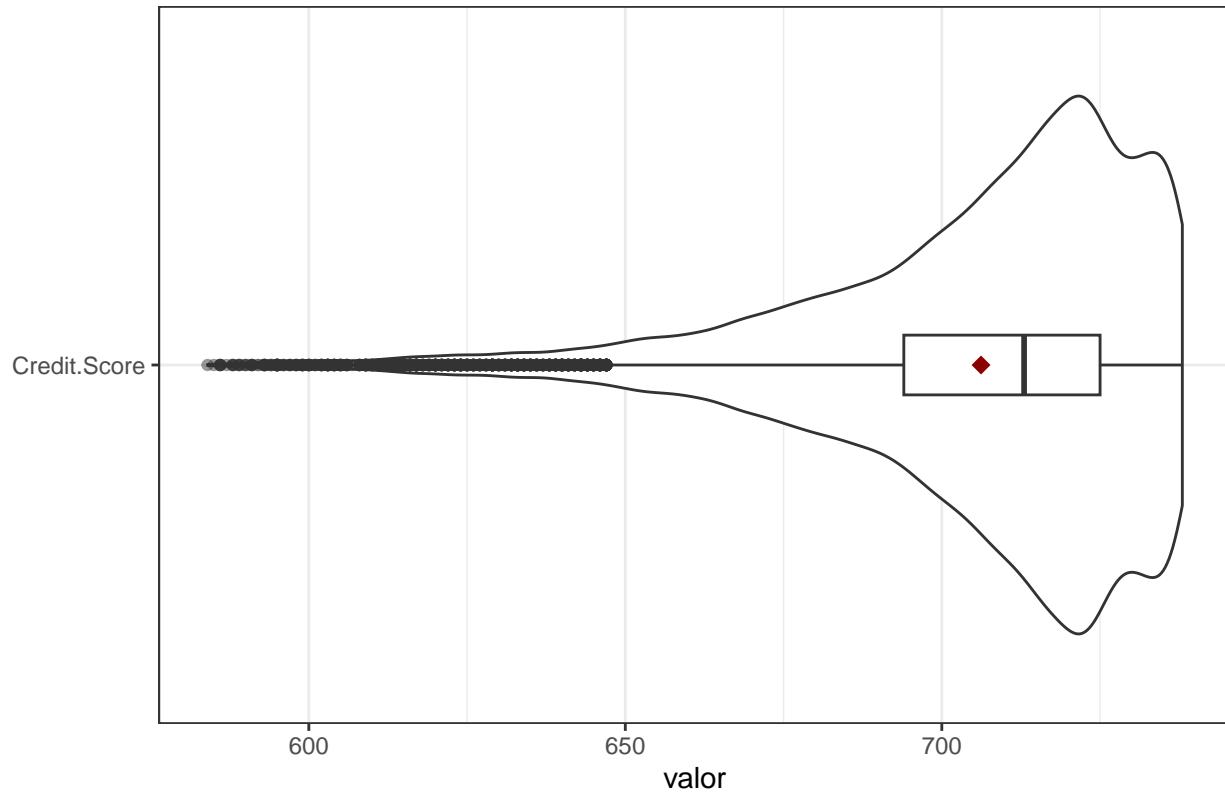
train_selected = Train_copy_continuas %>% filter(Credit.Score <= 738) %>% select(Credit.Score)

train_long_form6 = gather(train_selected, key = "variable", value = "valor")

ggplot(train_long_form6, aes(y=variable, x=valor))+
  geom_violin() +
  geom_boxplot(width = 0.1, fill = "white", alpha = 0.5)+
  # geom_jitter()+
  stat_summary(fun=mean, colour="darkred", geom="point",
              shape=18, size=3, show.legend=FALSE)+
  scale_y_discrete(NULL)+
  ggtitle(glue("Gráfico de Credit.Score <= 738 con {format(nrow(train_selected), big.mark=',')}"))
  theme_bw()

```

Gráfico de Credit.Score <= 738 con 26,176

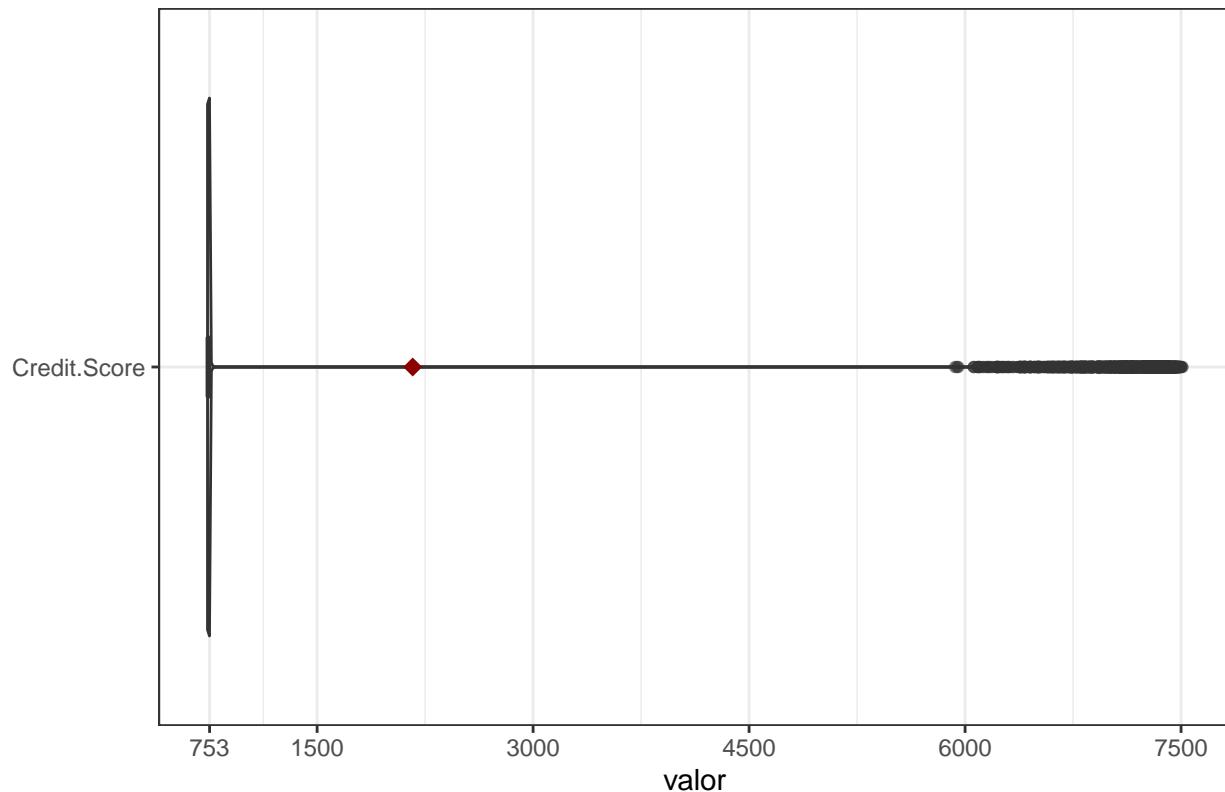


```
train_selected = Train_copy_continuas %>% filter(Credit.Score > 738) %>% select(Credit.Score)

train_long_form7 = gather(train_selected, key = "variable", value = "valor")

ggplot(train_long_form7, aes(y=variable, x=valor))+
  geom_violin() +
  geom_boxplot(width = 0.1, fill = "white", alpha = 0.5) +
  # geom_jitter()+
  stat_summary(fun=mean, colour="darkred", geom="point",
              shape=18, size=3, show.legend=FALSE) +
  scale_x_continuous(breaks = c(753, 1500, 3000, 4500, 6000, 7500)) +
  scale_y_discrete(NULL) +
  ggtitle(glue("Gráfico de Credit.Score > 738 con {format(nrow(train_selected), big.mark=',')}")) +
  theme_bw()
```

Gráfico de Credit.Score > 738 con 8,033



```
ggplot(train_long_form7, aes(valor))+
  geom_histogram(bins = 60) +
  ggtitle("Histógrama Credit.Score (> 738)") +
  scale_x_continuous(NULL) +
  theme_bw()
```

Histógrama Credit.Score (> 738)



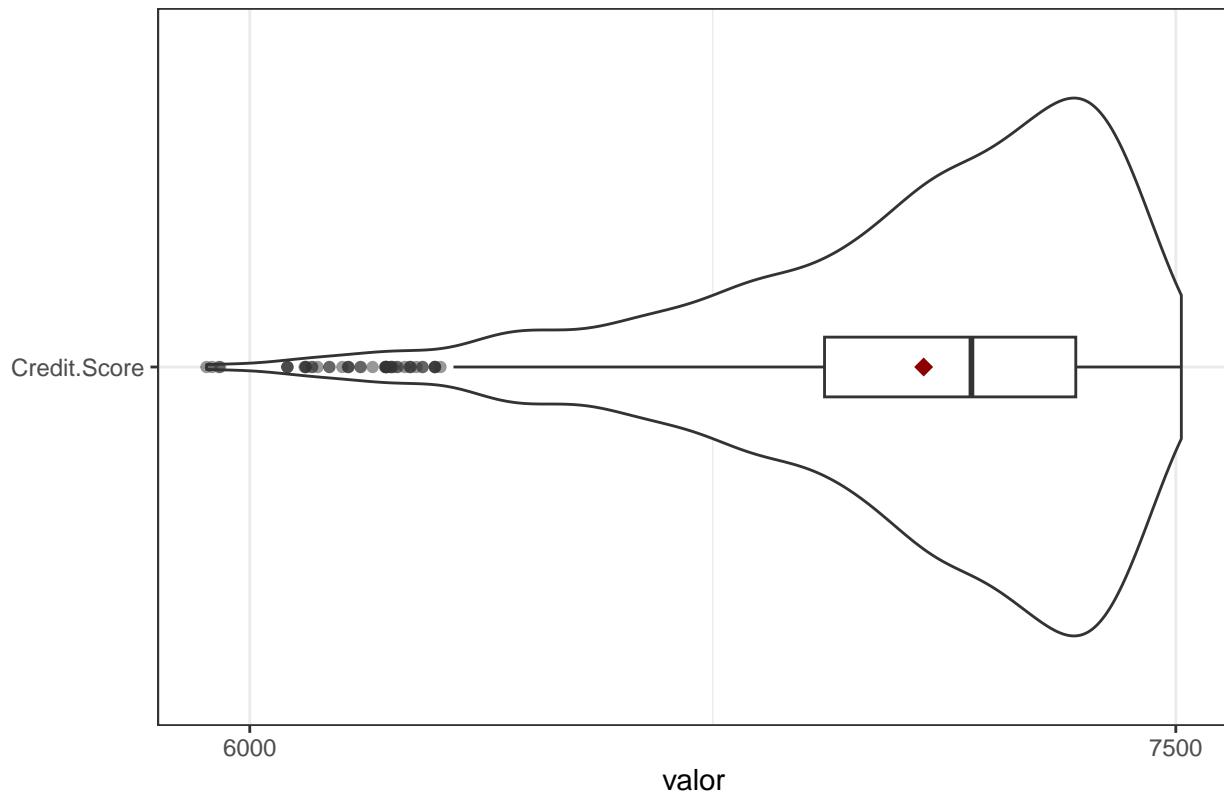
Con lo anterior, se decide filtrar todas aquellas observaciones mayores a 753, en donde ubicamos al valor 753 como el punto donde si se filtra los valores (que suponemos extremos en el vector) mayores a este, no se perderían tantas observaciones. De hecho toma sentido este subgrupo > 753 con un boxplot más violin con forma y sentido, pero se filtran dichas observaciones ya que el modelo no ajustaría apropiadamente para este subgrupo, y en proporción contra el total, un 5.25%, es desecharle.

```
train_selected = Train_copy_continuas %>% filter(Credit.Score > 753) %>% select(Credit.Score)

train_long_form7 = gather(train_selected, key = "variable", value = "valor")

ggplot(train_long_form7, aes(y=variable, x=valor))+
  geom_violin() +
  geom_boxplot(width = 0.1, fill = "white", alpha = 0.5) +
  # geom_jitter()+
  stat_summary(fun=mean, colour="darkred", geom="point",
              shape=18, size=3, show.legend=FALSE) +
  scale_x_continuous(breaks = c(753, 1500, 3000, 4500, 6000, 7500)) +
  scale_y_discrete(NULL) +
  ggtitle(glue("Gráfico de Credit.Score > 753 con {format(nrow(train_selected), big.mark=',')} observac"))
  theme_bw()
```

Gráfico de Credit.Score > 753 con 1,797 observaciones



```

train_selected = Train_copy_continuas %>% filter(Credit.Score <= 753) %>% select(Credit.Score)
train_long_form8 = gather(train_selected, key = "variable", value = "valor")

plot_1 = ggplot(train_long_form8, aes(y=variable, x=valor))+
  geom_violin() +
  geom_boxplot(width = 0.1, fill = "white", alpha = 0.5) +
  # geom_jitter() +
  stat_summary(fun=mean, colour="darkred", geom="point",
              shape=18, size=3, show.legend=FALSE) +
  scale_y_discrete(NULL) +
  ggtitle(glue("Credit.Score posterior al filtro (valores menores o iguales a 753)")) +
  theme_bw()

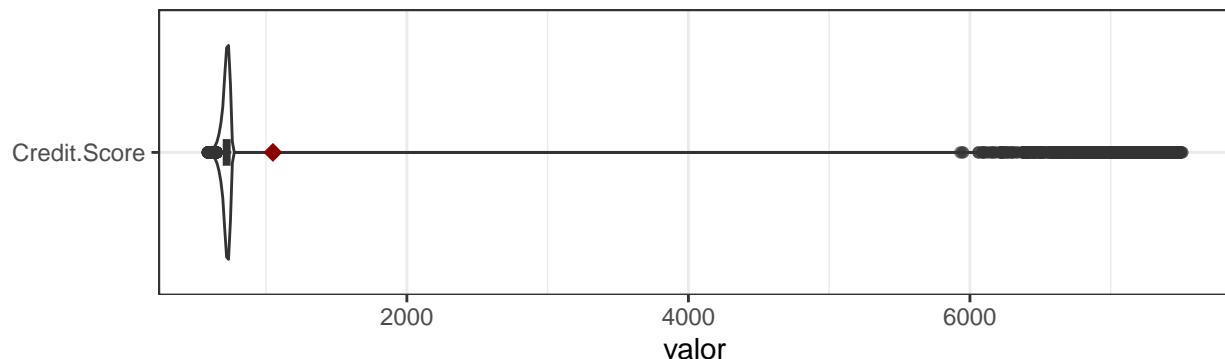
train_selected = Train_copy_continuas %>% select(Credit.Score)
train_long_form9 = gather(train_selected, key = "variable", value = "valor")

plot_2 = ggplot(train_long_form9, aes(y=variable, x=valor))+
  geom_violin() +
  geom_boxplot(width = 0.1, fill = "white", alpha = 0.5) +
  # geom_jitter() +
  stat_summary(fun=mean, colour="darkred", geom="point",
              shape=18, size=3, show.legend=FALSE) +
  scale_y_discrete(NULL) +
  ggtitle(glue("Credit.Score previo al filtro")) +
  theme_bw()

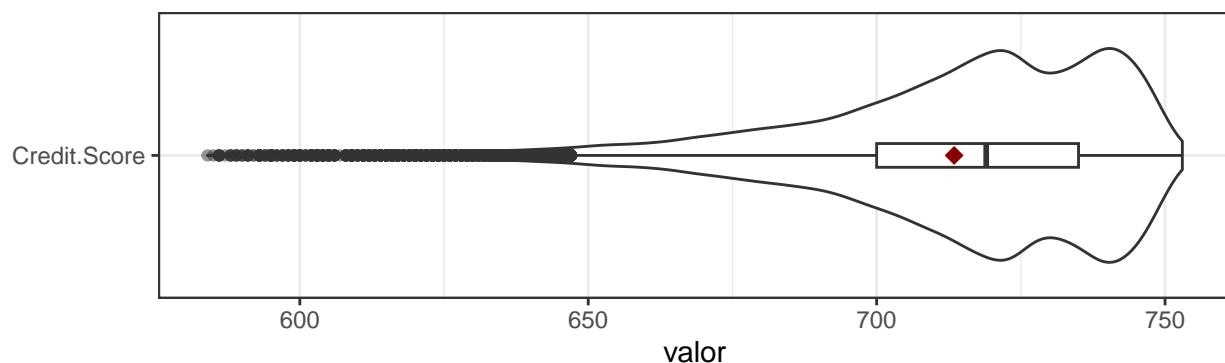
grid.arrange(plot_2, plot_1, ncol = 1)

```

Credit.Score previo al filtro



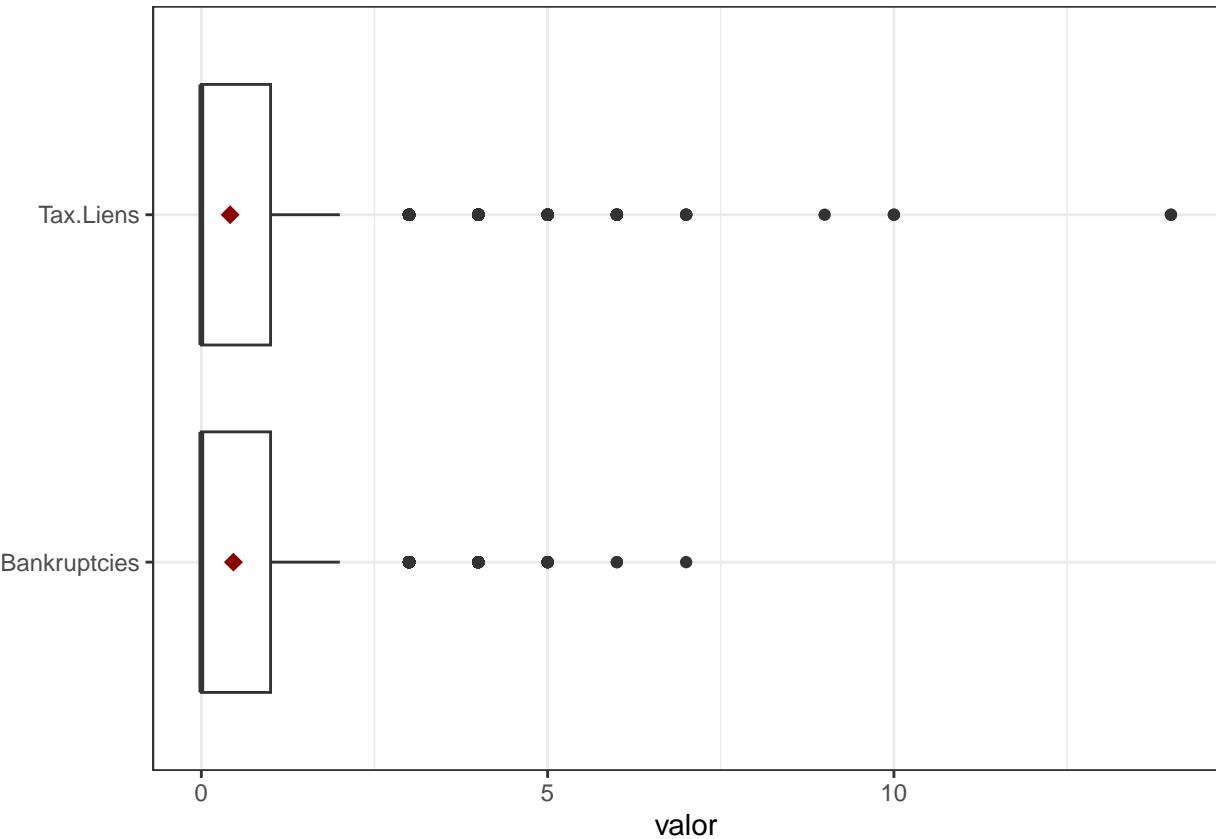
Credit.Score posterior al filtro (valores menores o iguales a 753)



7mo grupo: - Tax.Liens - Bankruptcies

```
train_selected = select(Train_copy_continuas, Tax.Liens, Bankruptcies)
train_long_form6 = gather(train_selected, key = "variable", value = "valor")

ggplot(train_long_form6, aes(y=variable, x=valor))+
  geom_boxplot()+
  # geom_jitter()+
  stat_summary(fun=mean, colour="darkred", geom="point",
              shape=18, size=3, show.legend=FALSE)+
  scale_y_discrete(NULL)+
  theme_bw()
```



```

rm(train_long_form, train_long_form1, train_long_form2, train_long_form3, train_long_form4, train_long_123)
rm(df_temp, df_temp2, df_temp3, df.temp)

## Warning in rm(df_temp, df_temp2, df_temp3, df.temp): objeto 'df_temp2' no
## encontrado

## Warning in rm(df_temp, df_temp2, df_temp3, df.temp): objeto 'df_temp3' no
## encontrado

## Warning in rm(df_temp, df_temp2, df_temp3, df.temp): objeto 'df.temp' no
## encontrado

rm(maximum_123,maximum_4)

## Warning in rm(maximum_123, maximum_4): objeto 'maximum_123' no encontrado

## Warning in rm(maximum_123, maximum_4): objeto 'maximum_4' no encontrado

rm(plot_1, plot_2)
rm(tc.copy, train_selected)

## Warning in rm(tc.copy, train_selected): objeto 'tc.copy' no encontrado

```

— Eliminar valores de variables que a criterio no ayudaran en el ajuste del modelo —

```
# Es aquí donde Train_copy sufrira los primeros filtros

Train_copy_1 = Train_copy %>% filter(Current.Loan.Amount <= 80000000, Credit.Score <= 753, Maximum.Open
print(glue("Train_copy en un inicio tuvo {format(nrow(Train_copy), big.mark = ',')} obs"))

## Train_copy en un inicio tuvo 34,209 obs

print(glue("Train_copy_1 ahora tiene {format(nrow(Train_copy_1), big.mark = ',')} obs"))

## Train_copy_1 ahora tiene 27,691 obs

print(glue("Se redujo el dataset en un {(1 - (nrow(Train_copy_1)/nrow(Train_copy)))*100 }%"))

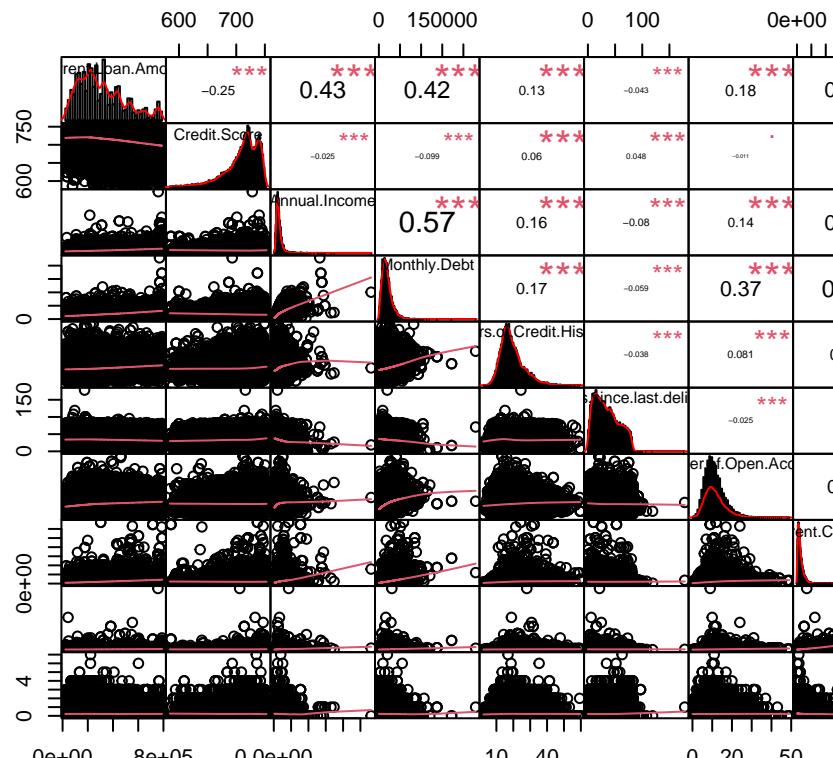
## Se redujo el dataset en un 19.0534654623052%
#(1 - (nrow(Train_copy_1)/nrow(Train_copy)))*100
```

———— ANALISIS BIVARIADO —————

```
describe(Train_copy_1)
```

```
## # A tibble: 18 x 8
##   variable      type    na na_pct unique   min    mean    max
##   <chr>        <chr>   <int> <dbl>   <int>   <dbl>   <dbl>
## 1 Loan.Status   fct     0     0     2 NA     NA     NA
## 2 Term          fct     0     0     2 NA     NA     NA
## 3 Years.in.current.job fct     0     0     12 NA     NA     NA
## 4 Home.Ownership fct     0     0     4 NA     NA     NA
## 5 Purpose        fct     0     0     15 NA     NA     NA
## 6 Current.Loan.Amount dbl     0     0     18300 2.14e4 3.13e+5 7.89e5
## 7 Credit.Score   dbl     0     0     170   5.84e2 7.12e+2 7.53e2
## 8 Annual.Income  dbl     0     0     18300 1.65e5 1.45e+6 2.81e7
## 9 Monthly.Debt   dbl     0     0     18298 0       1.90e+4 2.29e5
## 10 Years.of.Credit.History dbl    0     0     452   2.3 e0  1.92e+1 5.96e1
## 11 Months.since.last.delinqu~ dbl    0     0     855   0       3.48e+1 1.78e2
## 12 Number.of.Open.Accounts  dbl    0     0     48    0       1.14e+1 4.9 e1
## 13 Number.of.Credit.Problems dbl    0     0     11    0       5.3 e-1 1.5 e1
## 14 Current.Credit.Balance  dbl    0     0     18253 0       2.58e+5 6.72e6
## 15 Maximum.Open.Credit    dbl    0     0     18245 0       5.78e+5 7.17e7
## 16 Bankruptcies      dbl    0     0     8     0       4.6 e-1 7 e0
## 17 Tax.Liens         dbl    0     0     11   0       4.1 e-1 1.4 e1
## 18 ID                int    0     0     27691 1 e0   1.72e+4 3.42e4
```

```
Train_copy_1_continues = Train_copy_1 %>% select(Current.Loan.Amount, Credit.Score, Annual.Income, Mont
chart.Correlation(select(Train_copy_1_continues, -c(Tax.Liens)), histogram=TRUE, pch=19, main = "Matriz
```



Matriz de correlaciones e inferencia sobre ella

Con el análisis de la matriz de correlación podríamos plantearnos las siguientes interacciones:

- Current.Credit.Balance & Maximum.Open.Credit
- Annual.Income & Monthly.Debt

xray and Funnel chart

```
# Train_copy_1 %>%
#   make_xray() %>%
#   view_xray()
```

```
# Train_copy_1 %>% binarize() %>%
#   glimpse() %>%
#   correlate(target = Loan.Status__1 ) %>%
#   plot_correlation_funnel()
```

```
# Train_copy_1 %>% binarize() %>%
#   glimpse() %>%
#   correlate(target = Loan.Status__0 ) %>%
#   plot_correlation_funnel()
```

Análisis bivariado

- Exploratorio bivariado (shiny app)

```
# explore(Train_copy_1)
# Train %>% describe_cat(Current.Loan.Amount)
```

A resaltar que:

EN EL PERIODO DEL CREDITO - Son más los clientes que solicitan un crédito a corto plazo (70%) que los que solicitan a LP (30%) - Aquellos clientes con default son el grupo predominante en el gpo de clientes que solicitaron un crédito a largo plazo - La mediana del current loan es más alta para clientes con creditos CP que a LP, corroborar la media porque se notan outliers altos en CP. - Son más los clientes con una hipoteca que piden crédito a LP que a CP, que los clientes que rentan su vivienda con créditos a CP que a LP. (variacion > 10%) - La media de la deuda actual del cliente es mayor para los clientes que solicitan un crédito a LP, que uno a CP, lo cual tiene logica de negocio. - La media del credit score es mayor para los clientes que solicitan crédito a CP que a LP.

EN LOS AÑOS DE TRABAJO DEL CLIENTE: - El grupo que predomina es el de > 10 años de trabajo con un 35%, seguido de 2 años con un 8.5%

EN EL DUEÑO DE LA VIVIENDA: - Destaca el grupo de clientes con hipoteca con un 51%, seguido de Renta con un 39% y solo casa propia con un 9% - Hay dos factores que se podrían unificar: have mortgage & home mortgage, aunque la prop del primeor es de 0.2% por lo que no vale la pena

EN EL PROPOSITO DEL CRÉDITO: - Destaca con un 73% el proposito de consolidación de la deuda como proposito declarado.

EN LAS BANCARROTAS: - El 65% de los clientes declara nunca haber caído en bancarrota, le sigue el 25% con al menos una bancarrota. - El grupo, por proposito del crédito, que predomina en una bancarrota declarada es aquel grupo que declara solicitar el crédito para “moving”

EN LA DEUDA ACTUAL DEL CLIENTE: - La mediana de la deuda tiene una relación negativa con el aumento en el credit scoring, a mayor credit score menor es la mediana (falta corroborar la media dado outliers en altos credit scores) - la mediana sigue una relacion positiva con el ingreso anual (corroborar media) - la mediana sigue una relacion positiva con la deuda mensual (corroborar media) - la mediana sigue una relacion positiva con el balance crediticio (corroborar media)

EN LOS AÑOS DE HISTORIAL CREDITICIO:

- La mediana de los años mantiene una relacion positiva con el balance crediticio

EN EL BALANCE ACTUAL DEL CREDITO: - La mediana del balance mantiene una relacion positiva con el maximo de creditos abiertos.

EN EL NUMERO DE BANCARROTAS:

- La mediana del numero máximo de creditos abiertos se relaciona negativamente con el número de bancarrotas. Entre más bancarrotas es menor el numero de créditos abiertos.

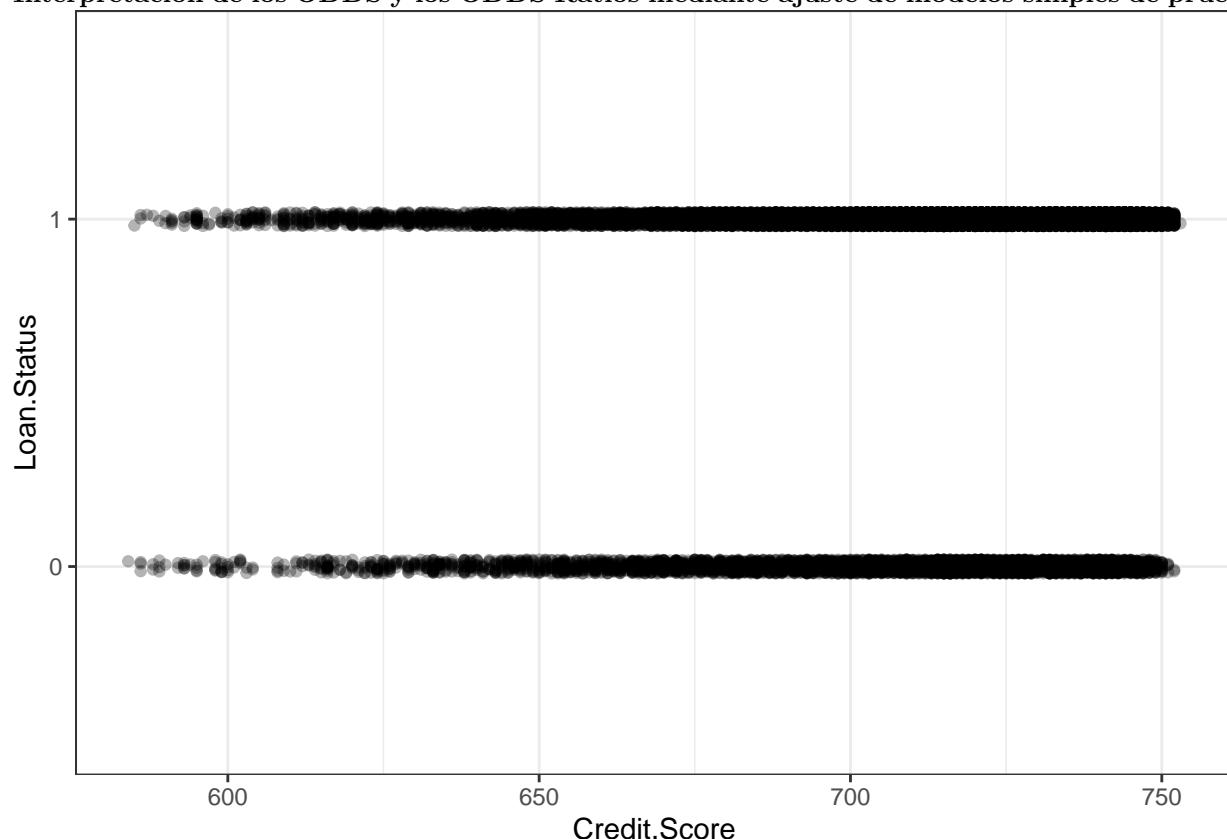
Análisis multivariado

Ajuste del modelo de regresión logística

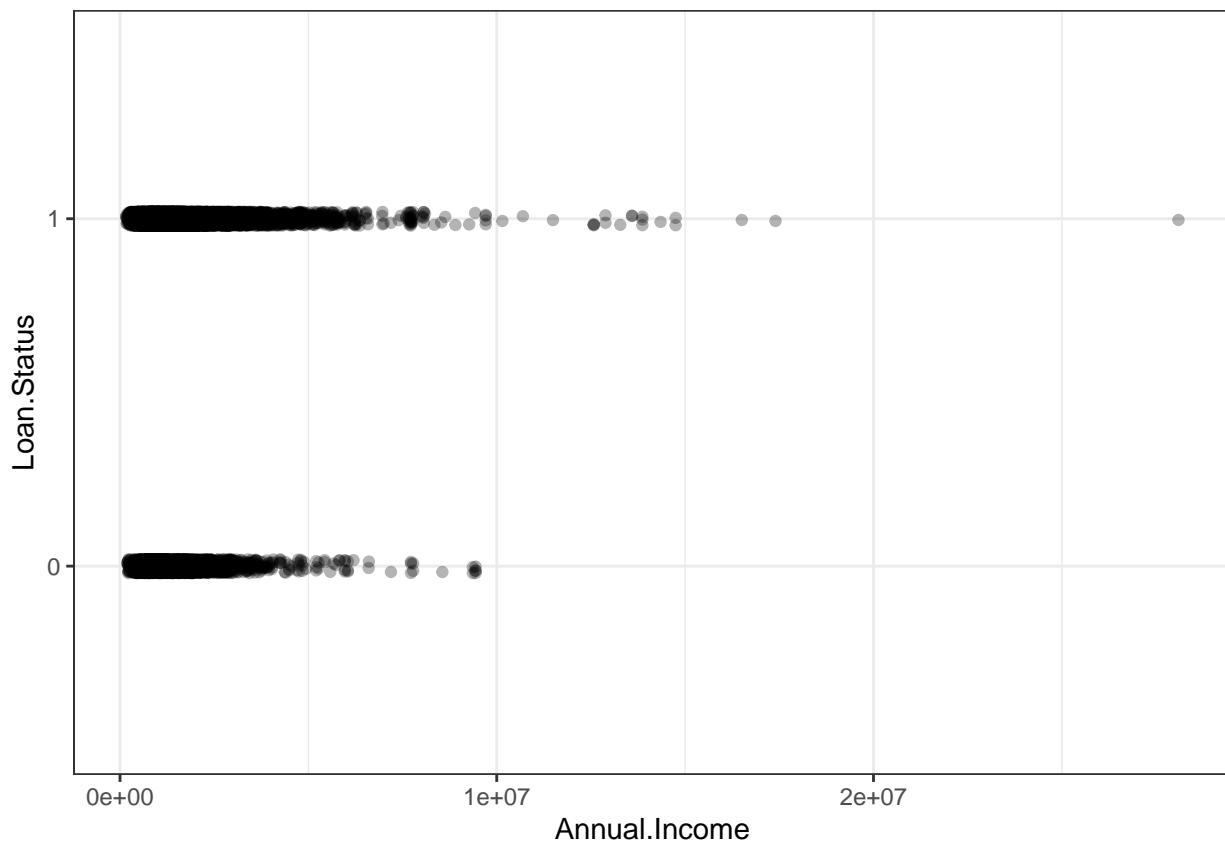
— AJUSTE INICIAL DE UN MODELO SIMPLE PARA LA INTERPRETACIÓN DE LOS ODDS Y LOS ODDS RATIOS —

```
##### GRAFICA DE LA OPCION BINOMINAL CON AJUSTE DE MODELO GLM
ggplot(Train_copy_1, aes(x=Credit.Score, y=Loan.Status)) +
  geom_jitter(height = .02, width = 0, alpha = 0.3) +
  theme_bw()
```

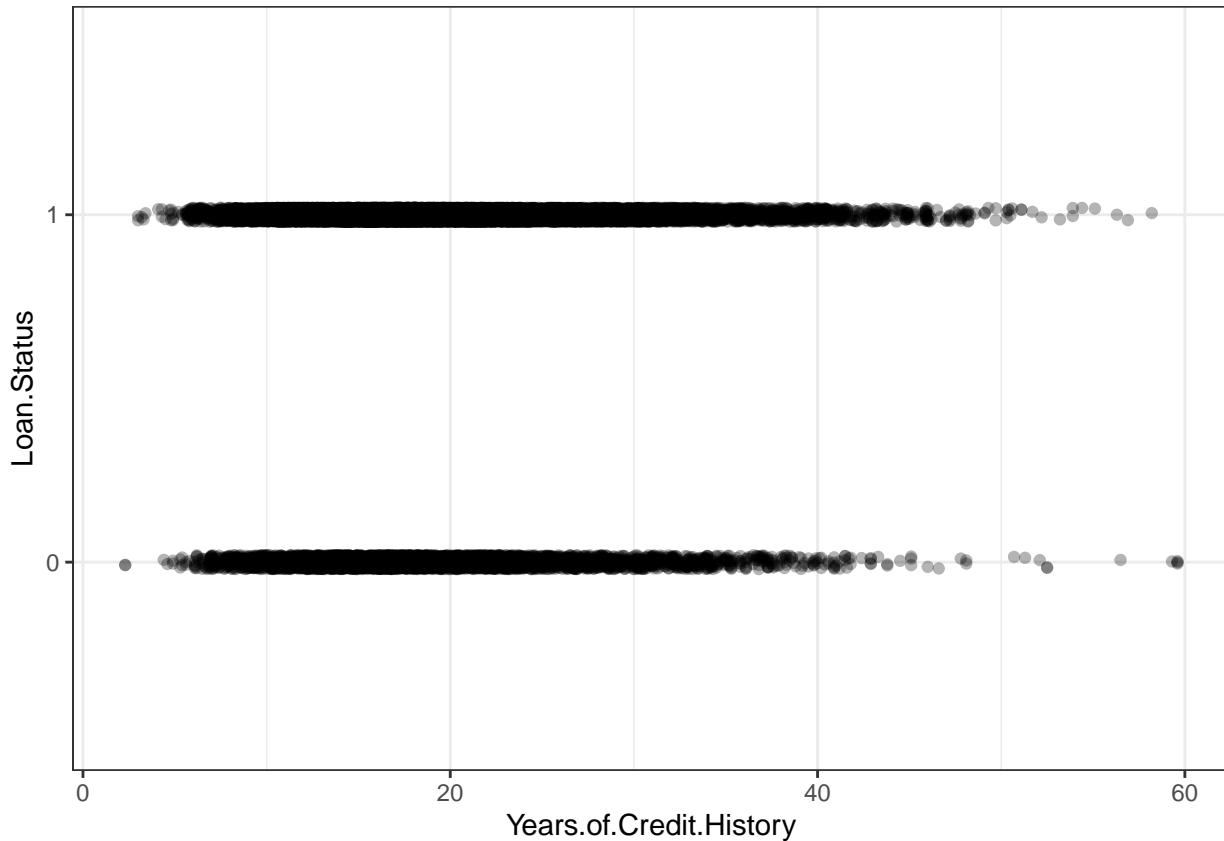
Interpretación de los ODDS y los ODDS Ratios mediante ajuste de modelos simples de prueba



```
ggplot(Train_copy_1, aes(x=Annual.Income, y=Loan.Status)) +  
  geom_jitter(height = .02, width = 0, alpha = 0.3) +  
  theme_bw()
```



```
ggplot(Train_copy_1, aes(x=Years.of.Credit.History, y=Loan.Status)) +  
  geom_jitter(height = .02, width = 0, alpha = 0.3) +  
  theme_bw()
```



```
### METER EN UN GRID VERTICAL
```

Inicialmente ajustamos dos modelos logit simples, es decir con solo una variable independiente en estudio. Para este caso primero analizaremos e interpretaremos la razón de momios o el odd ratio entre el modelo `Loan.Status ~ Credit.Score`, y repetiremos ese mismo ejercicio pero para `Loan.Status ~Annual.Income`, y `Loan.Status ~Years.of.Credit.History`.

- Iteración `Loan.Status ~ Credit.Score`

```
# AJUSTE DEL MODELO SIMPLE (Loan.Status & Credit.Score)
modelo_simple_1 = glm(data = Train_copy_1,
                      formula = Loan.Status~Credit.Score,
                      family = "binomial")
summary(modelo_simple_1)
```

```
##
## Call:
## glm(formula = Loan.Status ~ Credit.Score, family = "binomial",
##      data = Train_copy_1)
##
## Deviance Residuals:
##       Min        1Q     Median        3Q       Max
## -1.9446    0.5849    0.6248    0.6607    0.9432
##
## Coefficients:
```

```

##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.4410630  0.3582536 -9.605   <2e-16 ***
## Credit.Score  0.0068726  0.0005052 13.604   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 27054  on 27690  degrees of freedom
## Residual deviance: 26875  on 27689  degrees of freedom
## AIC: 26879
##
## Number of Fisher Scoring iterations: 4

```

```
print("Coeficientes como exponencial del num de euler")
```

```
## [1] "Coeficientes como exponencial del num de euler"
```

```
exp(modelo_simple_1$coefficients)
```

```

## (Intercept) Credit.Score
## 0.03203062 1.00689627

```

Del modelo de regresión logística `Loan.Status ~ Credit.Score` interpretamos que: - la independiente es estadísticamente significante dentro del modelo para explicar `Loan.Status`. - El logaritmo del odd (probabilidad de éxito sobre la probabilidad de fracaso) para `Credit.Score` es cercano a cero pero no igual a cero (`0.0068726`). - El odd ratio del `Credit.Score` $e^{0.0068726}$ es igual a `1.00689627`.

De lo que podemos concluir que con un aumento en una unidad del `Credit.Score` es en un 0.6% más probable que se dé un 1 que un 0, donde 1 es `Fully Paid` y 0 es `Charged Off`.

(es aquí donde se hacen las apuestas de 4 a 1 por ejemplo, cuando el OR es 4.00)

- Iteración `Loan.Status ~ Annual.Income`

```

# AJUSTE DEL MODELO SIMPLE (Loan.Status & Annual.Income)
modelo_simple_2 = glm(data = Train_copy_1,
                      formula = Loan.Status~Annual.Income,
                      family = "binomial")
summary(modelo_simple_2)

```

```

##
## Call:
## glm(formula = Loan.Status ~ Annual.Income, family = "binomial",
##      data = Train_copy_1)
##
## Deviance Residuals:
##       Min        1Q     Median        3Q       Max
## -2.5004    0.5807    0.6480    0.6761    0.7295
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)

```

```

## (Intercept) 1.154e+00 3.234e-02 35.691 <2e-16 ***
## Annual.Income 2.041e-07 2.102e-08 9.712 <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 27054 on 27690 degrees of freedom
## Residual deviance: 26947 on 27689 degrees of freedom
## AIC: 26951
##
## Number of Fisher Scoring iterations: 4

```

```
print("Coeficientes como exponencial del num de euler")
```

```
## [1] "Coeficientes como exponencial del num de euler"
```

```
exp(modelo_simple_2$coefficients)
```

```

## (Intercept) Annual.Income
## 3.172156 1.000000

```

Del modelo de regresión logística `Loan.Status ~ Annual.Income` interpretamos que: - la independiente es estadísticamente significante dentro del modelo para explicar `Loan.Status`. - El logaritmo del odd (probabilidad de éxito sobre la probabilidad de fracaso) para `Credit.Score` es 0, por lo que el odd ratio será igual a 1 ($e^0 = 1$), interpretando que un aumento o disminución en `Annual.Income` no explica ninguna razón de cambio en `Loan.Status`

Por lo que podríamos valorar eliminar dicha variable del modelo, pero ¿ entonces por qué resulta ser estadísticamente significativa en un inicio?

```
# AJUSTE DEL MODELO SIMPLE (Loan.Status & umber.of.Credit.Problems)
modelo_simple_3 = glm(data = Train_copy_1,
                      formula = Loan.Status~Years.of.Credit.History,
                      family = "binomial")
summary(modelo_simple_3)
```

```

##
## Call:
## glm(formula = Loan.Status ~ Years.of.Credit.History, family = "binomial",
##      data = Train_copy_1)
##
## Deviance Residuals:
##    Min      1Q   Median      3Q     Max
## -2.0046  0.6170  0.6495  0.6637  0.7030
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.240379  0.045618 27.191 < 2e-16 ***
## Years.of.Credit.History 0.010485  0.002278  4.602 4.18e-06 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

## 
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 27054 on 27690 degrees of freedom
## Residual deviance: 27033 on 27689 degrees of freedom
## AIC: 27037
##
## Number of Fisher Scoring iterations: 4

print("Coeficientes como exponencial del num de euler")

## [1] "Coeficientes como exponencial del num de euler"

exp(modelo_simple_3$coefficients)

##           (Intercept) Years.of.Credit.History
##            3.456924          1.010540

```

Del modelo de regresión logística `Loan.Status ~ Years.of.Credit.History` interpretamos que: - la independiente es estadísticamente significante dentro del modelo para explicar `Loan.Status`. - El logaritmo del odd (probabilidad de éxito sobre la probabilidad de fracaso) para `Years.of.Credit.History` es diferente a cero (0.010485). - El odd ratio del `Credit.Score` $e^{0.010485}$ es igual a 1.010540.

De lo que podemos concluir que con un aumento en una unidad de la variable `Years.of.Credit.History` es en un 1.05% más probable que se de un 1 que un 0, donde 1 es `Fully Paid` y 0 es `Charged Off`.

Ajuste del modelo de regresión logística base (modelo 1) ————— AJUSTE DEL MODELO DE REGRESIÓN LOGÍSTICA MULTIPLE —————

```
Train_copy_1_model = select(Train_copy_1, -c(ID))
```

Metemos todos los balones en una canasta y posteriormente sacamos los que no ajusten en el modelo

```

modelo_multiple_1 = glm(data = Train_copy_1_model,
                        formula = Loan.Status~., family = "binomial")
summary(modelo_multiple_1)

```

Ajuste no.1

```

## 
## Call:
## glm(formula = Loan.Status ~ ., family = "binomial", data = Train_copy_1_model)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.7224    0.4893   0.6014   0.6842   1.4496
##
## Coefficients:

```

```

##                                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -2.201e+00  6.589e-01 -3.340 0.000837 ***
## TermShort.Terms                3.668e-01  4.026e-02  9.110 < 2e-16 ***
## Years.in.current.job1.year    1.980e-01  8.656e-02  2.288 0.022165 *
## Years.in.current.job10+.years 1.414e-01  6.309e-02  2.241 0.025037 *
## Years.in.current.job2.years   1.501e-02  7.597e-02  0.198 0.843426
## Years.in.current.job3.years   2.277e-01  7.887e-02  2.887 0.003890 **
## Years.in.current.job4.years   1.985e-01  8.501e-02  2.335 0.019519 *
## Years.in.current.job5.years   1.952e-01  8.386e-02  2.328 0.019909 *
## Years.in.current.job6.years   1.769e-02  8.497e-02  0.208 0.835111
## Years.in.current.job7.years   5.044e-02  8.681e-02  0.581 0.561183
## Years.in.current.job8.years   6.337e-02  9.306e-02  0.681 0.495891
## Years.in.current.job9.years   2.008e-01  9.776e-02  2.054 0.039930 *
## Years.in.current.jobn/a      -8.801e-02  9.351e-02 -0.941 0.346577
## HomeOwnershipHomeMortgage     8.612e-01  4.940e-01  1.743 0.081297 .
## HomeOwnershipOwnHome          7.906e-01  4.956e-01  1.595 0.110654
## HomeOwnershipRent             5.967e-01  4.930e-01  1.210 0.226089
## Purposebuy_a_car              1.046e+00  2.175e-01  4.808 1.53e-06 ***
## Purposebuy_house               3.150e-01  1.935e-01  1.628 0.103510
## Purposedebt_consolidation    4.044e-01  1.106e-01  3.658 0.000255 ***
## Purposeeducational_expenses   3.380e-01  5.139e-01  0.658 0.510703
## Purposehome_improvements     4.513e-01  1.262e-01  3.576 0.000349 ***
## Purposemajor_purchase          -1.840e-01  2.689e-01 -0.684 0.493737
## Purposemedical_bills          3.759e-01  1.740e-01  2.161 0.030731 *
## Purposemoving                 -8.103e-02  3.595e-01 -0.225 0.821656
## Purposeother                  4.258e-01  1.189e-01  3.579 0.000344 ***
## Purposerenewable_energy       1.021e+01  9.839e+01  0.104 0.917387
## Purposesmall_business         -9.517e-01  2.703e-01 -3.521 0.000430 ***
## Purposetake_a_trip            1.010e+00  3.177e-01  3.179 0.001476 **
## Purposevacation               1.196e-01  3.787e-01  0.316 0.752173
## Purposewedding                -1.702e-02  4.557e-01 -0.037 0.970208
## Current.Loan.Amount           -5.383e-07  1.119e-07 -4.810 1.51e-06 ***
## Credit.Score                  2.603e-03  6.203e-04  4.197 2.70e-05 ***
## Annual.Income                 3.713e-07  3.024e-08 12.280 < 2e-16 ***
## Monthly.Debt                 -1.137e-05  1.917e-06 -5.934 2.96e-09 ***
## Years.of.Credit.History       6.189e-03  2.444e-03  2.532 0.011333 *
## Months.since.last.delinquent 1.622e-03  7.197e-04  2.254 0.024180 *
## Number.of.Open.Accounts        -4.583e-03  3.386e-03 -1.353 0.175919
## Number.of.Credit.Problems     -7.054e-02  2.883e-02 -2.447 0.014417 *
## Current.Credit.Balance        -4.420e-08  9.717e-08 -0.455 0.649182
## Maximum.Open.Credit            6.338e-08  3.926e-08  1.614 0.106454
## Bankruptcies                  8.237e-02  2.860e-02  2.880 0.003982 **
## Tax.Liens                      2.338e-02  2.778e-02  0.842 0.399981
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 27054 on 27690 degrees of freedom
## Residual deviance: 26394 on 27649 degrees of freedom
## AIC: 26478
##
## Number of Fisher Scoring iterations: 10

```

```

anova(modelo_multiple_1)

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Loan.Status
##
## Terms added sequentially (first to last)
##
##
##                                Df Deviance Resid. Df Resid. Dev
## NULL                           27690    27054
## Term                          1   203.475   27689    26851
## Years.in.current.job          11   40.308   27678    26810
## Home.Ownership                3   94.299   27675    26716
## Purpose                        14   65.279   27661    26651
## Current.Loan.Amount           1   0.921    27660    26650
## Credit.Score                  1   38.978   27659    26611
## Annual.Income                 1   145.411   27658    26466
## Monthly.Debt                  1   44.970   27657    26421
## Years.of.Credit.History       1    7.466   27656    26413
## Months.since.last.delinquent 1   4.737    27655    26408
## Number.of.Open.Accounts        1   0.848    27654    26408
## Number.of.Credit.Problems     1   1.147    27653    26406
## Current.Credit.Balance        1   1.133    27652    26405
## Maximum.Open.Credit            1   3.366    27651    26402
## Bankruptcies                  1   7.730    27650    26394
## Tax.Liens                      1   0.710    27649    26394

```

Dado que no resultan significantes, estas variables se eliminan del modelo:

Home.Ownership
Home Number.of.Open.Accounts
Current.Credit.Balance
Maximum.Open.Credit
Tax.Liens
ID

```

modelo_multiple_1 = update(object = modelo_multiple_1, formula. = .~.-Home.Ownership -Number.of.Open.Ac
summary(modelo_multiple_1)

```

Modelo 1 con Ajuste no.1

```

##
## Call:
## glm(formula = Loan.Status ~ Term + Years.in.current.job + Purpose +
##       Current.Loan.Amount + Credit.Score + Annual.Income + Monthly.Debt +
##       Years.of.Credit.History + Number.of.Credit.Problems + Bankruptcies,
##       family = "binomial", data = Train_copy_1_model)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q        Max
## -2.7849    0.4982   0.6049   0.6790   1.4265

```

```

## 
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.889e+00  4.312e-01 -4.381 1.18e-05 ***
## TermShort Term        3.360e-01  3.997e-02  8.407 < 2e-16 ***
## Years.in.current.job1 year 2.074e-01  8.643e-02  2.400 0.016397 *
## Years.in.current.job10+ years 1.927e-01  6.265e-02  3.076 0.002097 **
## Years.in.current.job2 years  2.069e-02  7.586e-02  0.273 0.785110
## Years.in.current.job3 years  2.466e-01  7.873e-02  3.132 0.001738 **
## Years.in.current.job4 years  2.157e-01  8.484e-02  2.542 0.011012 *
## Years.in.current.job5 years  2.201e-01  8.369e-02  2.630 0.008530 **
## Years.in.current.job6 years  5.715e-02  8.468e-02  0.675 0.499763
## Years.in.current.job7 years  8.579e-02  8.653e-02  0.991 0.321488
## Years.in.current.job8 years  1.120e-01  9.271e-02  1.208 0.226930
## Years.in.current.job9 years  2.461e-01  9.747e-02  2.525 0.011582 *
## Years.in.current.jobn/a    -2.904e-02  9.298e-02 -0.312 0.754793
## Purposebuy a car          1.019e+00  2.172e-01  4.693 2.69e-06 ***
## Purposebuy house           2.484e-01  1.932e-01  1.286 0.198409
## Purposedebt consolidation 3.764e-01  1.102e-01  3.415 0.000638 ***
## Purposeeducational expenses 2.607e-01  5.125e-01  0.509 0.611055
## Purposehome improvements   5.109e-01  1.257e-01  4.066 4.79e-05 ***
## Purposemajor_purchase      -1.695e-01  2.680e-01 -0.633 0.527038
## Purposemedical bills       3.452e-01  1.738e-01  1.986 0.047027 *
## Purposemoving               -1.764e-01  3.586e-01 -0.492 0.622873
## Purposeother                3.928e-01  1.187e-01  3.308 0.000940 ***
## Purposerenewable_energy   1.029e+01  9.843e+01  0.105 0.916745
## Purposesmall_business      -9.838e-01  2.702e-01 -3.641 0.000272 ***
## Purposetake a trip         6.941e-01  2.608e-01  2.661 0.007784 **
## Purposevacation             7.179e-02  3.778e-01  0.190 0.849296
## Purposewedding              -6.251e-02  4.555e-01 -0.137 0.890847
## Current.Loan.Amount        -4.641e-07  1.100e-07 -4.220 2.44e-05 ***
## Credit.Score                 3.162e-03  6.097e-04  5.186 2.15e-07 ***
## Annual.Income                3.893e-07  3.005e-08 12.958 < 2e-16 ***
## Monthly.Debt                -1.145e-05  1.743e-06 -6.573 4.93e-11 ***
## Years.of.Credit.History     7.961e-03  2.416e-03  3.295 0.000984 ***
## Number.of.Credit.Problems   -5.109e-02  2.247e-02 -2.273 0.023004 *
## Bankruptcies                  7.079e-02  2.630e-02  2.692 0.007104 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
## Null deviance: 27054  on 27690  degrees of freedom
## Residual deviance: 26463  on 27657  degrees of freedom
## AIC: 26531
## 
## Number of Fisher Scoring iterations: 10

```

```
confint(object = modelo_multiple_1, level = 0.95)
```

Intervalos de confianza del Ajuste no.1


```

## Years.of.Credit.History      3.242299e-03  1.271300e-02
## Number.of.Credit.Problems   -9.484768e-02 -6.720971e-03
## Bankruptcies                 1.936354e-02  1.224522e-01

```

Ecuación

- Identificando la ecuación

```

# Obtener el vector de coeficientes del modelo
coeficientes <- coef(modelo_multiple_1)

# Obtener los nombres de las variables del modelo
nombres_vars <- names(coeficientes)[-1]

# Crear una cadena con la ecuación del modelo
cadena_ecuacion <- paste(round(coeficientes[1], 4), "+",
                           paste(round(coeficientes[-1], 4), nombres_vars, sep = "*"), collapse = " + ")

# Imprimir la ecuación del modelo
cat("La ecuación del modelo es:\n", cadena_ecuacion, "\n")

## La ecuación del modelo es:
## -1.889 + 0.336*TermShort Term + -1.889 + 0.2074*Years.in.current.job1 year + -1.889 + 0.1927*Years...

```

- Identificando los coeficientes

```

nombres_vars = names(coef(modelo_multiple_1))
resumen = summary(modelo_multiple_1)

# Iterar por cada variable e imprimir su nombre y coeficiente
for (i in seq_along(nombres_vars)) {
  cat(paste(nombres_vars[i], ":", resumen$coefficients[i, "Estimate"], "\n"))
}

## (Intercept) : -1.88904640521648
## TermShort Term : 0.336005681325557
## Years.in.current.job1 year : 0.207427315140393
## Years.in.current.job10+ years : 0.192710310158611
## Years.in.current.job2 years : 0.0206852991976141
## Years.in.current.job3 years : 0.246564634659547
## Years.in.current.job4 years : 0.215698169357802
## Years.in.current.job5 years : 0.220134498583248
## Years.in.current.job6 years : 0.0571501498700893
## Years.in.current.job7 years : 0.0857891535241224
## Years.in.current.job8 years : 0.112025671799244
## Years.in.current.job9 years : 0.246074498471009
## Years.in.current.jobn/a : -0.0290385773218456
## Purposebuy a car : 1.01941888127771
## Purposebuy house : 0.24841077862314
## Purposedebt consolidation : 0.376396738932033
## Purposeeducational expenses : 0.260663408270432
## Purposehome improvements : 0.510942167631994

```

```

## Purposemajor_purchase : -0.169498709668742
## Purposemedical bills : 0.345192876999899
## Purposemoving : -0.176357862906933
## Purposeother : 0.392793397152014
## Purposerenewable_energy : 10.2891846462885
## Purposesmall_business : -0.983784893147889
## Purposetake a trip : 0.694073475673608
## Purposevacation : 0.0717878744057671
## Purposewedding : -0.0625094433157294
## Current.Loan.Amount : -4.64082754784547e-07
## Credit.Score : 0.00316211573805694
## Annual.Income : 3.89323525110045e-07
## Monthly.Debt : -1.14541428144143e-05
## Years.of.Credit.History : 0.0079607202690665
## Number.of.Credit.Problems : -0.0510894136792419
## Bankruptcies : 0.0707855501736036

```

ODDS Ratios del ajuste no.1

- Identificando los odds ratios e^{B_k}

```

nombres_vars = names(coef(modelo_multiple_1))
resumen = summary(modelo_multiple_1)

# Iterar por cada variable e imprimir su nombre y el num de euler elevado al coeficiente
for (i in seq_along(nombres_vars)) {
  cat(paste(nombres_vars[i], ":", exp(resumen$coefficients[i, "Estimate"])), "\n")
}

## (Intercept) : 0.1512159388351
## TermShort Term : 1.39934697493408
## Years.in.current.job1 year : 1.23050827426841
## Years.in.current.job10+ years : 1.21253148458281
## Years.in.current.job2 years : 1.02090072280234
## Years.in.current.job3 years : 1.27962188846987
## Years.in.current.job4 years : 1.24072783280019
## Years.in.current.job5 years : 1.24624433741345
## Years.in.current.job6 years : 1.05881477936219
## Years.in.current.job7 years : 1.08957657070406
## Years.in.current.job8 years : 1.11854157525123
## Years.in.current.job9 years : 1.27899485315359
## Years.in.current.jobn/a : 0.971378990543835
## Purposebuy a car : 2.77158367672821
## Purposebuy house : 1.28198643667157
## Purposedebt consolidation : 1.45702507769035
## Purposeeducational expenses : 1.29779076627086
## Purposehome improvements : 1.66686091776251
## Purposemajor_purchase : 0.844087843632589
## Purposemedical bills : 1.41226228631304
## Purposemoving : 0.838317926760513
## Purposeother : 1.48111235568577
## Purposerenewable_energy : 29412.7827511589
## Purposesmall_business : 0.373893271300502

```

```

## Purposetake a trip : 2.00185344851495
## Purposevacation : 1.07442740633693
## Purposewedding : 0.93940419168121
## Current.Loa n.Amount : 0.999999535917353
## Credit.Score : 1.00316712049985
## Annual.Income : 1.0000003893236
## Monthly.Debt : 0.999988545922784
## Years.of.Credit.History : 1.00799249105282
## Number.of.Credit.Problems : 0.950193706418723
## Bankruptcies : 1.0733510212454

```

Predicción y Matriz de confusión para el modelo no. 1

- Predicción

Estableciendo una predicción con punto de corte arbitrario de 0.85

```

prediccion_1 = predict(object = modelo_multiple_1,
                      newdata = Train_copy_1_model, type = "response")

```

```

Prediccion_1<-as.factor(ifelse(prediccion_1<=0.85,yes = 1, no = 0))

```

```

table(Prediccion_1)

```

```

## Prediccion_1
##      0      1
## 5935 21756

```

```

print("Valores reales")

```

```

## [1] "Valores reales"

```

```

table(Train_copy_1_model$Loan.Status)

```

```

##
##      0      1
## 5305 22386

```

- Matriz de confusión con punto de corte de 0.85 (arbitrario)

```

confusionMatrix(reference = as.factor(Train_copy_1_model$Loan.Status), Prediccion_1)

```

```

## Confusion Matrix and Statistics
##
##                  Reference
## Prediction      0      1
##      0    712   5223
##      1   4593  17163
##
##                  Accuracy : 0.6455

```

```

##                               95% CI : (0.6398, 0.6512)
##      No Information Rate : 0.8084
##      P-Value [Acc > NIR] : 1
##
##                           Kappa : -0.0948
##
## McNemar's Test P-Value : 2.172e-10
##
##                           Sensitivity : 0.13421
##                           Specificity : 0.76668
##      Pos Pred Value : 0.11997
##      Neg Pred Value : 0.78889
##                           Prevalence : 0.19158
##                           Detection Rate : 0.02571
##      Detection Prevalence : 0.21433
##      Balanced Accuracy : 0.45045
##
##      'Positive' Class : 0
##
```

```

#confusionMatrix(data = as.factor(Train_copy_1_model$Loan.Status),
#                  Prediccion)

```

```

Exactitud_1 <- vector()
Corte_1 <- seq(0.05, 0.95, by = 0.001)

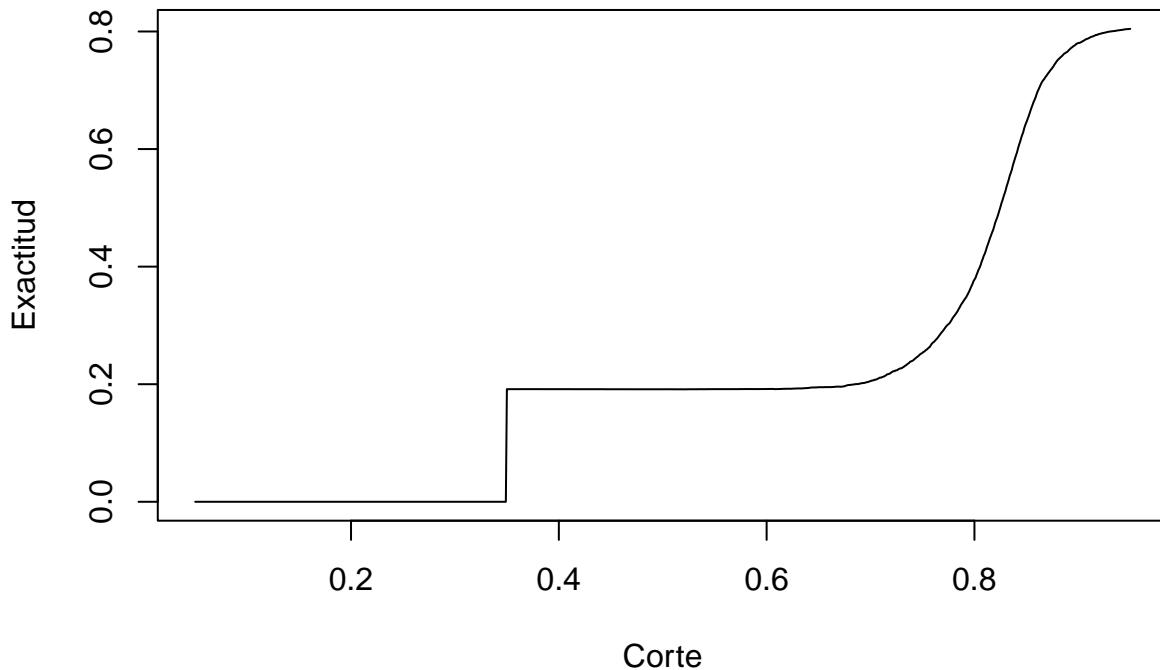
for (i in 1:length(Corte_1)) {
  Prediccion <- as.factor(ifelse(prediccion_1 <= Corte_1[i], yes = 1, no = 0))

  Exactitud_1[i] <- tryCatch(
    {
      confusionMatrix(data = as.factor(Train_copy_1_model$Loan.Status), Prediccion)$overall[1]
    },
    error = function(e) {
      0 # O cualquier valor predeterminado que deseas asignar
    }
  )
}

plot(x=Corte_1, y = Exactitud_1, type = "l", main ="Punto de corte óptimo", xlab="Corte", ylab="Exactitud")

```

Modelo optimizado con el punto máximo de la curva de exactitud (punto de corte ideal)
Punto de corte óptimo



ubicando tal punto de corte óptimo

```
corte_optimo_1 = which(Exactitud_1==max(Exactitud_1))
Corte_1[corte_optimo_1]
```

```
## [1] 0.95
```

Predicción y matriz de confusión para el modelo no. 1 ajustado con punto de corte óptimo

- Generando una nueva predicción con el punto de corte óptimo para el modelo ajustado

```
Prediccion_1_2<-as.factor(ifelse(prediccion_1<=Corte_1[corte_optimo_1],yes = 1, no = 0))
confusionMatrix(data = as.factor(Train_copy_1_model$Loan.Status),
                 Prediccion_1_2)
```

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction    0     1
##       0     12   5293
##       1    122  22264
##
##                  Accuracy : 0.8044
##                  95% CI : (0.7997, 0.8091)
##      No Information Rate : 0.9952
##      P-Value [Acc > NIR] : 1
##
```

```

##                               Kappa : -0.0051
##
##  Mcnemar's Test P-Value : <2e-16
##
##                               Sensitivity : 0.0895522
##                               Specificity : 0.8079254
##                               Pos Pred Value : 0.0022620
##                               Neg Pred Value : 0.9945502
##                               Prevalence : 0.0048391
##                               Detection Rate : 0.0004334
##  Detection Prevalence : 0.1915785
##                               Balanced Accuracy : 0.4487388
##
##                               'Positive' Class : 0
##

```

```
table(Train_copy_1_model$Loan.Status)
```

```

##                               0      1
##  5305  22386

```

```
table(Prediccion_1_2)
```

```

##  Prediccion_1_2
##                               0      1
##  134   27557

```

```

##### Comparar modelo ya ajustado (no.1) con uno nuevo y muy simple
Prediccion_1_roc<-predict(object = modelo_multiple_1,
                           newdata = Train_copy_1_model, type = "response",
                           se.fit = T)

predicciones_roc<-prediction(Prediccion_1_roc$fit,Train_copy_1$Loan.Status)
Desempeno_roc_1<-performance(predicciones_roc,'tpr','fpr')

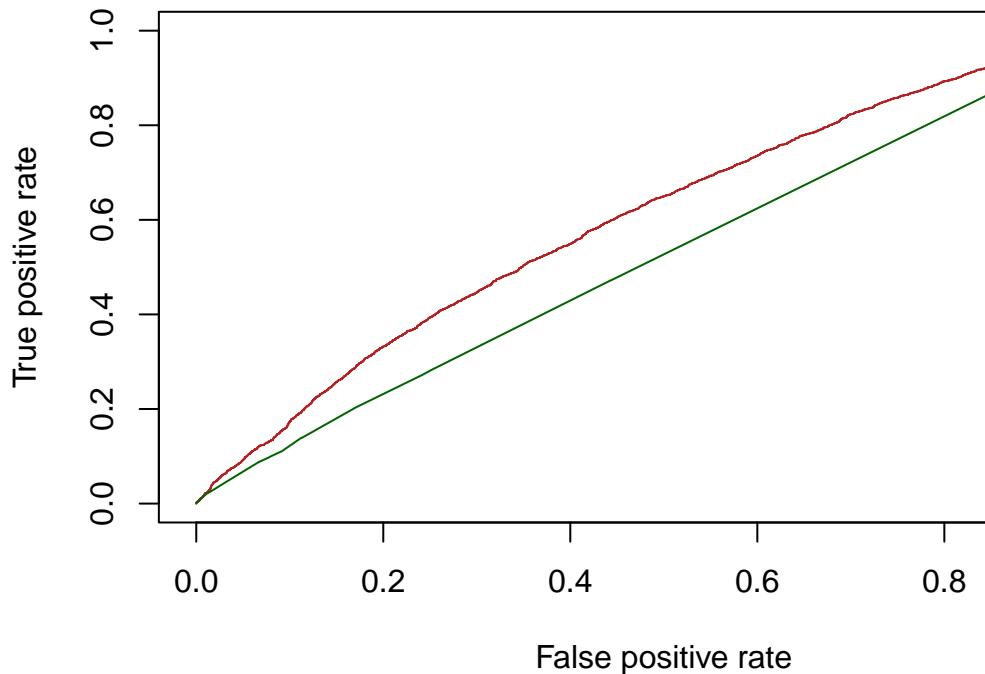
## Cgenerando nuevo modelo simple
modelo_simple_4 <- glm(data = Train_copy_1,
                        formula = Loan.Status~Purpose+Bankruptcies, family = "binomial")

Prediccion_3_simple<-predict(object = modelo_simple_4,
                               newdata = Train_copy_1_model, type = "response",
                               se.fit = T)

predicciones3_simple<-prediction(Prediccion_3_simple$fit,Train_copy_1_model$Loan.Status)
Desempeno3_simple<-performance(predicciones3_simple,'tpr','fpr')

#ejecutar en conjunto
plot(Desempeno_roc_1,col="firebrick")
plot(Desempeno3_simple, col = "darkgreen", add = T)

```



Curva ROC modelo ajustado no.1

False positive rate

El primer modelo ajustado, aunque en la matriz de confusión arroja un 80% de accuracy, apenas es mejor que uno simple. - hay que estudiar cómo se compone una matriz de confusión para ver en qué apartados puede estar fallando.

Interacciones Interacción a

- Annual.Income & Monthly.Debt

Ajuste no. 1 plus (con interacciones previamente propuestas)

```
modelo_multiple_1_plus = glm(data = Train_copy_1_model,
                             formula = Loan.Status~Term+Years.in.current.job+Purpose+
                                         Current.Loan.Amount+Credit.Score+Annual.Income*Monthly.Debt+Years.of.Credit.History+Num
summary(modelo_multiple_1_plus)
```

```
##
## Call:
## glm(formula = Loan.Status ~ Term + Years.in.current.job + Purpose +
##       Current.Loan.Amount + Credit.Score + Annual.Income * Monthly.Debt +
##       Years.of.Credit.History + Number.of.Credit.Problems + Maximum.Open.Credit +
##       Bankruptcies, family = "binomial", data = Train_copy_1_model)
##
## Deviance Residuals:
##      Min        1Q     Median        3Q       Max
## -2.7576    0.4916    0.6050    0.6804    1.8512
##
## Coefficients:
## (Intercept)          Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.820e+00  4.349e-01 -4.185 2.85e-05 ***
## TermShort Term      3.426e-01  4.000e-02  8.563 < 2e-16 ***
## Years.in.current.job1 year 2.089e-01  8.646e-02  2.416 0.015697 *
```

```

## Years.in.current.job10+ years 1.863e-01 6.271e-02 2.970 0.002976 **
## Years.in.current.job2 years 2.096e-02 7.590e-02 0.276 0.782439
## Years.in.current.job3 years 2.475e-01 7.879e-02 3.141 0.001683 **
## Years.in.current.job4 years 2.165e-01 8.488e-02 2.550 0.010764 *
## Years.in.current.job5 years 2.160e-01 8.374e-02 2.580 0.009876 **
## Years.in.current.job6 years 5.645e-02 8.471e-02 0.666 0.505186
## Years.in.current.job7 years 8.293e-02 8.657e-02 0.958 0.338058
## Years.in.current.job8 years 1.071e-01 9.276e-02 1.154 0.248401
## Years.in.current.job9 years 2.461e-01 9.750e-02 2.524 0.011605 *
## Years.in.current.jobn/a -2.143e-02 9.309e-02 -0.230 0.817967
## Purposebuy a car 1.027e+00 2.173e-01 4.725 2.30e-06 ***
## Purposebuy house 2.566e-01 1.932e-01 1.328 0.184098
## Purposedebt consolidation 3.836e-01 1.103e-01 3.478 0.000505 ***
## Purposeeducational expenses 2.840e-01 5.129e-01 0.554 0.579731
## Purposehome improvements 5.151e-01 1.258e-01 4.096 4.20e-05 ***
## Purposemajor_purchase -1.645e-01 2.683e-01 -0.613 0.539652
## Purposemedical bills 3.460e-01 1.739e-01 1.990 0.046602 *
## Purposemoving -1.765e-01 3.588e-01 -0.492 0.622822
## Purposeother 3.976e-01 1.188e-01 3.345 0.000821 ***
## Purposerenewable_energy 1.030e+01 9.841e+01 0.105 0.916612
## Purposesmall_business -9.912e-01 2.702e-01 -3.668 0.000244 ***
## Purposetake a trip 7.019e-01 2.609e-01 2.690 0.007146 **
## Purposevacation 6.200e-02 3.782e-01 0.164 0.869798
## Purposewedding -4.217e-02 4.556e-01 -0.093 0.926241
## Current.Loan.Amount -5.525e-07 1.123e-07 -4.921 8.63e-07 ***
## Credit.Score 2.927e-03 6.151e-04 4.759 1.95e-06 ***
## Annual.Income 4.574e-07 3.636e-08 12.578 < 2e-16 ***
## Monthly.Debt -7.600e-06 2.104e-06 -3.612 0.000304 ***
## Years.of.Credit.History 7.320e-03 2.430e-03 3.012 0.002594 **
## Number.of.Credit.Problems -4.872e-02 2.249e-02 -2.167 0.030252 *
## Maximum.Open.Credit 5.743e-08 2.632e-08 2.182 0.029105 *
## Bankruptcies 7.040e-02 2.630e-02 2.676 0.007443 **
## Annual.Income:Monthly.Debt -2.229e-12 5.790e-13 -3.850 0.000118 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 27054 on 27690 degrees of freedom
## Residual deviance: 26445 on 27655 degrees of freedom
## AIC: 26517
##
## Number of Fisher Scoring iterations: 10

anova(modelo_multiple_1_plus)

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Loan.Status
##
## Terms added sequentially (first to last)
##

```

```

##                                     Df Deviance Resid. Df Resid. Dev
## NULL                               27690      27054
## Term                                1  203.475   27689    26851
## Years.in.current.job                11   40.308   27678    26810
## Purpose                               14   70.684   27664    26740
## Current.Loan.Amount                  1    0.050   27663    26740
## Credit.Score                            1   50.218   27662    26690
## Annual.Income                           1  166.382   27661    26523
## Monthly.Debt                            1   40.685   27660    26482
## Years.of.Credit.History                 1   11.451   27659    26471
## Number.of.Credit.Problems                1    0.763   27658    26470
## Maximum.Open.Credit                      1    5.071   27657    26465
## Bankruptcies                            1    7.318   27656    26458
## Annual.Income:Monthly.Debt               1  12.504   27655    26445

```

- Destacar con el modelo plus de interacción que el AIC para el modelo 1 plus a tiene un menor AIC que el modelo 1.

```
modelo_multiple_1_plus$aic
```

```
## [1] 26517.4
```

```
modelo_multiple_1$aic
```

```
## [1] 26531.01
```

Por lo que: - Hay que evaluar el PRESS y el MRSPE - Hay que evaluar en la validación cruzada qué versión del modelo 1 es la mejor para predecir.

Aunque es de mencionar que la variable Maximum.Open.Credit con esta interacción recobró significancia.

Adicionalmente se ha probado la interacción de: Current.Credit.Balance & Maximum.Open.Credit pero no se han encontrado resultados que la favorezcan.

```
confint(object = modelo_multiple_1_plus, level = 0.95)
```

Intervalos de confianza del modelo no. 1 plus

```
## Waiting for profiling to be done...
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                               2.5 %      97.5 %
## (Intercept)           -2.670785e+00 -9.657792e-01
## TermShort Term        2.640798e-01  4.209000e-01
## Years.in.current.job1 year 3.987987e-02  3.789038e-01
## Years.in.current.job10+ years 6.248871e-02  3.083469e-01
## Years.in.current.job2 years -1.280135e-01  1.695639e-01
## Years.in.current.job3 years 9.305955e-02  4.019857e-01
## Years.in.current.job4 years 5.046206e-02  3.832680e-01
## Years.in.current.job5 years 5.222581e-02  3.805586e-01
## Years.in.current.job6 years -1.093122e-01  2.228576e-01
## Years.in.current.job7 years -8.632984e-02  2.531208e-01
## Years.in.current.job8 years -7.392758e-02  2.898148e-01
## Years.in.current.job9 years 5.619938e-02  4.385441e-01
## Years.in.current.jobn/a -2.032669e-01  1.617749e-01
## Purposebuy a car       6.120016e-01  1.466353e+00
## Purposebuy house       -1.175986e-01  6.410141e-01
## Purposedebt consolidation 1.639608e-01  5.967406e-01
## Purposeeducational expenses -6.481732e-01  1.404635e+00
## Purposehome improvements 2.663511e-01  7.597037e-01
## Purposemajor_purchase -6.797581e-01  3.757869e-01
## Purposemedical bills   7.825816e-03  6.902826e-01
## Purposemoving           -8.559888e-01  5.623702e-01
## Purposeother            1.617816e-01  6.279571e-01
## Purposerenewable_energy -9.780108e-01      NA
## Purposesmall_business  -1.518703e+00 -4.561932e-01
## Purposetake a trip     2.085287e-01  1.235939e+00
## Purposevacation         -6.456007e-01  8.520661e-01
## Purposewedding          -8.923345e-01  9.206103e-01
## Current.Loan.Amount    -7.723633e-07 -3.322319e-07
## Credit.Score            1.719135e-03  4.130627e-03
## Annual.Income           3.868310e-07  5.293386e-07
## Monthly.Debt           -1.174319e-05 -3.496140e-06
## Years.of.Credit.History 2.573299e-03  1.209922e-02
## Number.of.Credit.Problems -9.250859e-02 -4.328206e-03
## Maximum.Open.Credit     9.151167e-09  1.118807e-07
## Bankruptcies            1.895929e-02  1.220785e-01
## Annual.Income:Monthly.Debt -3.340114e-12 -1.069546e-12

```

Ecuación para el modelo no. 1 plus

- Identificando la ecuación

```

# Obtener el vector de coeficientes del modelo
coeficientes <- coef(modelo_multiple_1_plus)

# Obtener los nombres de las variables del modelo
nombres_vars <- names(coeficientes)[-1]

# Crear una cadena con la ecuación del modelo
cadena_ecuacion <- paste(round(coeficientes[1], 4), "+",
                           paste(round(coeficientes[-1], 4), nombres_vars, sep = "*"), collapse = " + ")

# Imprimir la ecuación del modelo
cat("La ecuación del modelo es:\n", cadena_ecuacion, "\n")

```

```

## La ecuación del modelo es:
## -1.8202 + 0.3426*TermShort Term + -1.8202 + 0.2089*Years.in.current.job1 year + -1.8202 + 0.1863*Ye

```

- Identificando los coeficientes para el modelo no. 1 plus

```

nombres_vars = names(coef(modelo_multiple_1_plus))
resumen = summary(modelo_multiple_1_plus)

# Iterar por cada variable e imprimir su nombre y coeficiente
for (i in seq_along(nombres_vars)) {
  cat(paste(nombres_vars[i], ":", resumen$coefficients[i, "Estimate"], "\n"))
}

```

```

## (Intercept) : -1.82020596582749
## TermShort Term : 0.342560190091518
## Years.in.current.job1 year : 0.208876834249251
## Years.in.current.job10+ years : 0.186251609218009
## Years.in.current.job2 years : 0.0209582545461086
## Years.in.current.job3 years : 0.247479344332605
## Years.in.current.job4 years : 0.216455325156521
## Years.in.current.job5 years : 0.21604984735363
## Years.in.current.job6 years : 0.0564497480382222
## Years.in.current.job7 years : 0.0829345765882696
## Years.in.current.job8 years : 0.107065866836408
## Years.in.current.job9 years : 0.246071775144147
## Years.in.current.jobn/a : -0.0214266449498968
## Purposebuy a car : 1.02687380591018
## Purposebuy house : 0.25664586147506
## Purposedebt consolidation : 0.383636666346523
## Purposeeducational expenses : 0.284022362533694
## Purposehome improvements : 0.515144717187252
## Purposemajor_purchase : -0.16454679313635
## Purposemedical bills : 0.346022589992417
## Purposemoving : -0.176489509407758
## Purposeother : 0.397555736722623
## Purposerenewable_energy : 10.3042285221475
## Purposesmall_business : -0.991244676348497
## Purposetake a trip : 0.701911845089237
## Purposevacation : 0.0619959847067464

```

```

## Purposewedding : -0.0421744623777086
## Current.Loan.Amount : -5.52453840957194e-07
## Credit.Score : 0.00292723660847249
## Annual.Income : 4.57350612721595e-07
## Monthly.Debt : -7.59957227571742e-06
## Years.of.Credit.History : 0.00731955989906187
## Number.of.Credit.Problems : -0.0487241767585226
## Maximum.Open.Credit : 5.74288102085127e-08
## Bankruptcies : 0.070397028029003
## Annual.Income:Monthly.Debt : -2.228844079432e-12

```

ODDS Ratios para el modelo no. 1 plus

- Identificando los odds ratios e^{B_k}

```

nombres_vars = names(coef(modelo_multiple_1_plus))
resumen = summary(modelo_multiple_1_plus)

# Iterar por cada variable e imprimir su nombre y el num de euler elevado al coeficiente
for (i in seq_along(nombres_vars)) {
  cat(paste(nombres_vars[i], ":", exp(resumen$coefficients[i, "Estimate"])), "\n")
}

```

```

## (Intercept) : 0.161992382602502
## TermShort Term : 1.40854913181195
## Years.in.current.job1 year : 1.23229321286434
## Years.in.current.job10+ years : 1.20472534224821
## Years.in.current.job2 years : 1.02117942114928
## Years.in.current.job3 years : 1.28079290647821
## Years.in.current.job4 years : 1.24166761280847
## Years.in.current.job5 years : 1.24116424621179
## Years.in.current.job6 years : 1.05807344319802
## Years.in.current.job7 years : 1.08647072560141
## Years.in.current.job8 years : 1.11300756235205
## Years.in.current.job9 years : 1.27899137003729
## Years.in.current.jobn/a : 0.978801274852323
## Purposebuy a car : 2.79232283252283
## Purposebuy house : 1.29258729073895
## Purposedebt consolidation : 1.46761211192902
## Purposeeducational expenses : 1.32846263821113
## Purposehome improvements : 1.67388072358039
## Purposemajor_purchase : 0.848278062424593
## Purposemedical bills : 1.41343454493269
## Purposemoving : 0.838207572402917
## Purposeother : 1.48818273808376
## Purposerenewable_energy : 29858.6100808317
## Purposesmall_business : 0.371114486010572
## Purposetake a trip : 2.01760637332745
## Purposevacation : 1.06395807261579
## Purposewedding : 0.958702508462422
## Current.Loan.Amount : 0.999999447546312
## Credit.Score : 1.00293152514906
## Annual.Income : 1.00000045735072

```

```

## Monthly.Debt : 0.999992400456601
## Years.of.Credit.History : 1.00734641335613
## Number.of.Credit.Problems : 0.95244379960774
## Maximum.Open.Credit : 1.00000005742881
## Bankruptcies : 1.0729340816051
## Annual.Income:Monthly.Debt : 0.999999999997771

```

Predicción y Matriz de confusión para el modelo no. 1 plus

- Predicción

Estableciendo una predicción con punto de corte arbitrario de 0.85

```

prediccion_1_plus = predict(object = modelo_multiple_1_plus,
                            newdata = Train_copy_1_model, type = "response")

Prediccion_1_plus<-as.factor(ifelse(prediccion_1_plus<=0.85, yes = 1, no = 0))

table(Prediccion_1_plus)

```

```

## Prediccion_1_plus
##      0      1
## 6086 21605

```

```

print("Valores reales")

## [1] "Valores reales"

table(Train_copy_1_model$Loan.Status)

```

```

##
##      0      1
## 5305 22386

```

- Matriz de confusión para el modelo 1 plus con punto de corte de 0.85 (arbitrario)

```

confusionMatrix(reference = as.factor(Train_copy_1_model$Loan.Status), Prediccion_1_plus)

## Confusion Matrix and Statistics
##
##          Reference
## Prediction    0     1
##           0 726 5360
##           1 4579 17026
##
##          Accuracy : 0.6411
##             95% CI : (0.6354, 0.6467)
##   No Information Rate : 0.8084
##   P-Value [Acc > NIR] : 1
##

```

```

##          Kappa : -0.0971
##
##  Mcnemar's Test P-Value : 5.121e-15
##
##          Sensitivity : 0.13685
##          Specificity : 0.76056
##          Pos Pred Value : 0.11929
##          Neg Pred Value : 0.78806
##          Prevalence : 0.19158
##          Detection Rate : 0.02622
##          Detection Prevalence : 0.21978
##          Balanced Accuracy : 0.44871
##
##          'Positive' Class : 0
##

```

```

Exactitud_1_plus <- vector()
Corte_1_plus <- seq(0.05, 0.95, by = 0.001)

for (i in 1:length(Corte_1_plus)) {
  Prediccion <- as.factor(ifelse(prediccion_1_plus <= Corte_1_plus[i], yes = 1, no = 0))

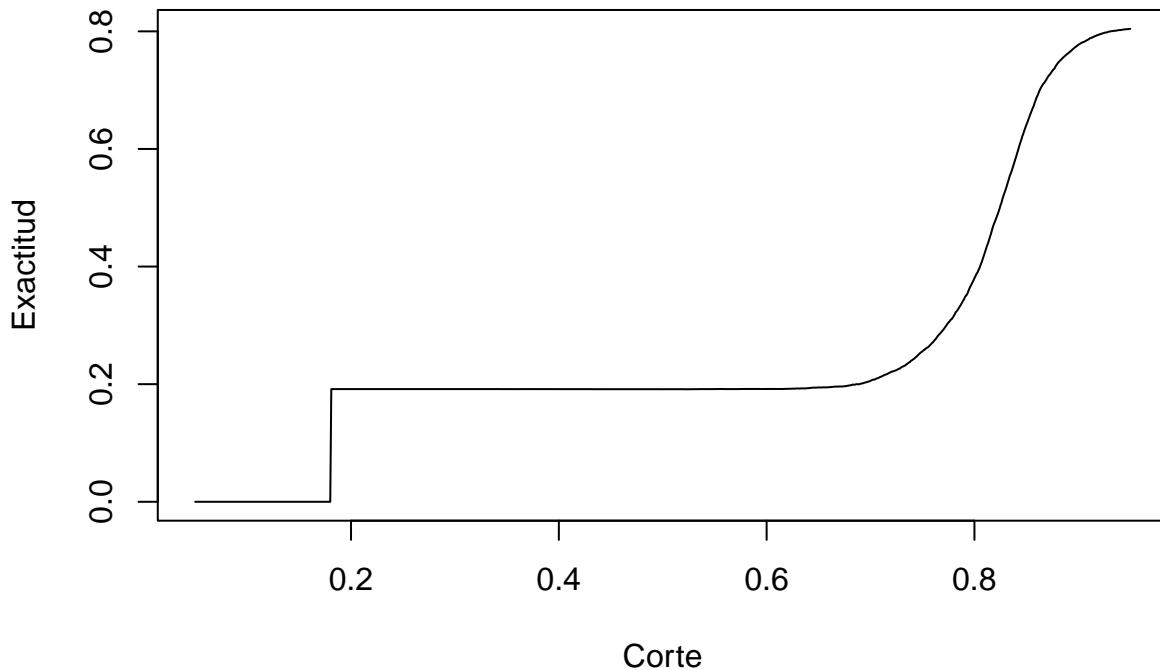
  Exactitud_1_plus[i] <- tryCatch(
    {
      confusionMatrix(data = as.factor(Train_copy_1_model$Loan.Status), Prediccion)$overall[1]
    },
    error = function(e) {
      0 # O cualquier valor predeterminado que deseas asignar
    }
  )
}

plot(x=Corte_1_plus, y = Exactitud_1_plus, type = "l", main ="Punto de corte óptimo", xlab="Corte", yla

```

Modelo optimizado con el punto máximo de la curva de exactitud (punto de corte ideal)

Punto de corte óptimo



ubicando tal punto de corte óptimo

```
corte_optimo_1_plus = which(Exactitud_1_plus==max(Exactitud_1_plus))
Corte_1_plus[corte_optimo_1_plus]
```

```
## [1] 0.95
```

Predicción y matriz de confusión para el modelo no. 1 plus con punto de corte optimo

- Generando una nueva predicción con el punto de corte óptimo para el modelo ajustado

```
Prediccion_1_2_plus<-as.factor(ifelse(prediccion_1_plus<=Corte_1_plus[corte_optimo_1_plus],yes = 1, no =
confusionMatrix(data = as.factor(Train_copy_1_model$Loan.Status),
Prediccion_1_2_plus)
```

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction      0      1
##       0     15   5290
##       1    132  22254
##
##                  Accuracy : 0.8042
##                  95% CI : (0.7995, 0.8089)
##      No Information Rate : 0.9947
##      P-Value [Acc > NIR] : 1
##
```

```

##                               Kappa : -0.0049
##
##  McNemar's Test P-Value : <2e-16
##
##                               Sensitivity : 0.1020408
##                               Specificity  : 0.8079437
##                               Pos Pred Value : 0.0028275
##                               Neg Pred Value : 0.9941035
##                               Prevalence   : 0.0053086
##                               Detection Rate : 0.0005417
##  Detection Prevalence : 0.1915785
##                               Balanced Accuracy : 0.4549922
##
##                               'Positive' Class : 0
##

```

```

##### Comparar modelo ya ajustado (no.1) con uno nuevo y muy simple
Prediccion_1_plus_roc<-predict(object = modelo_multiple_1_plus,
                                 newdata = Train_copy_1_model, type = "response",
                                 se.fit = T)

predicciones_roc_plus<-prediction(Prediccion_1_plus_roc$fit,Train_copy_1$Loan.Status)
Desempeno_roc_1_plus<-performance(predicciones_roc_plus,'tpr','fpr')

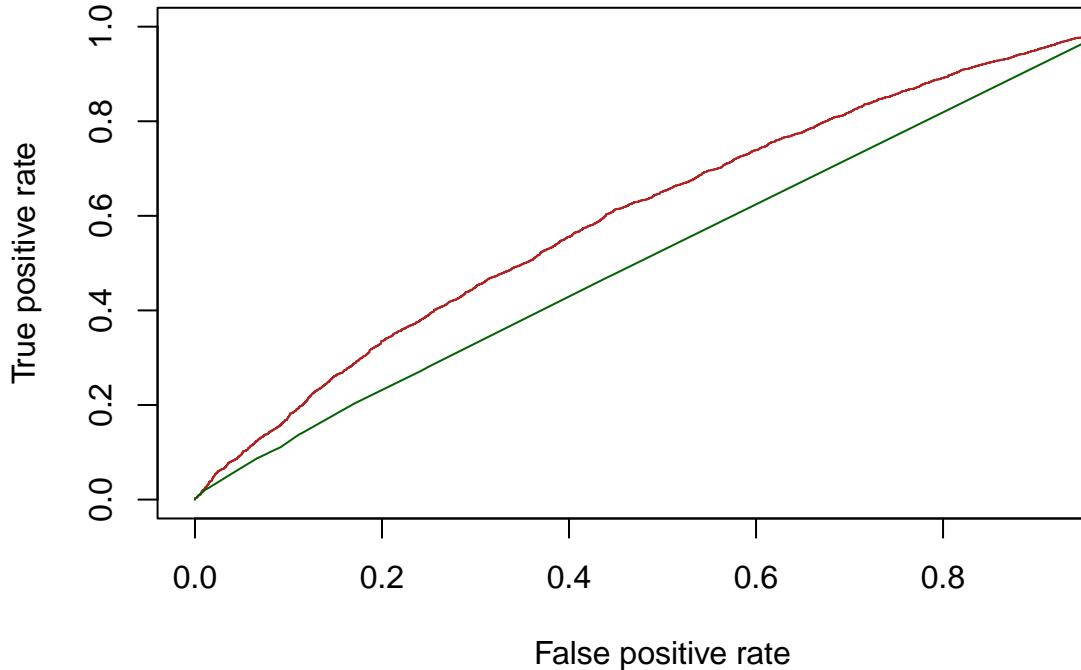
## Cgenerando nuevo modelo simple
modelo_simple_4 <- glm(data = Train_copy_1,
                         formula = Loan.Status~Purpose+Bankruptcies, family = "binomial")

Prediccion_3_simple<-predict(object = modelo_simple_4,
                               newdata = Train_copy_1_model, type = "response",
                               se.fit = T)

predicciones3_simple<-prediction(Prediccion_3_simple$fit,Train_copy_1_model$Loan.Status)
Desempeno3_simple<-performance(predicciones3_simple,'tpr','fpr')

#ejecutar en conjunto
plot(Desempeno_roc_1_plus,col="firebrick")
plot(Desempeno3_simple, col = "darkgreen", add = T)

```



Curva ROC modelo 1 plus

False positive rate

En este modelo con interacción, si bien el AIC disminuyó, también el accuracy que nos brinda la matriz de confusión. Se recomienda evaluar el PRESS y el MRPS para evaluar su viabilidad frente al modelo 1.

Reducción de dimensiones En este apartado se pretende evaluar el modelo no.1 con la finalidad de obtener el numero optimo de variables y que variables son las que conforman el modelo optimo.

```
### Con la libreria OLSRR no es posible trabajar modelos glm
# Todos_1 = ols_step_all_possible(modelo_multiple_1)
# plot(Todos_1)
```

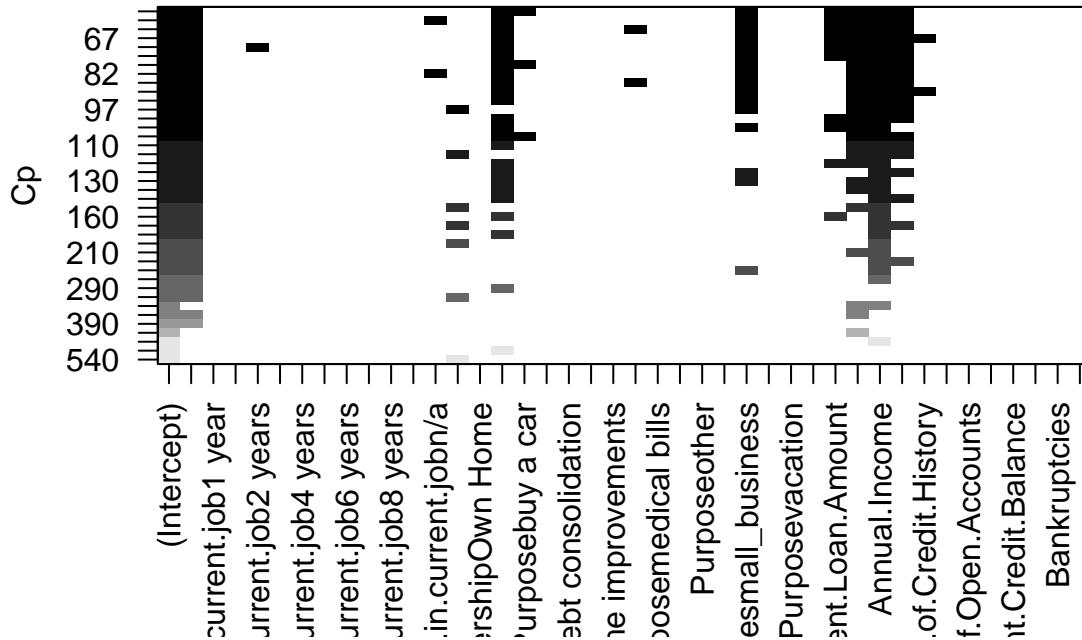
Selección de variables

Modelo 2 Selección de variables por medio de stepwise & estadístico Cp de mallow

```
Variables = colnames(Train_copy_1_model)
#####Exhaustivo
Subs<-regsubsets(data = Train_copy_1_model, x = Loan.Status~, nbest=5,
                   method = "exhaustive")#intensivo

plot(Subs, scale = c( "Cp"), main="Cp de Mallow")#Cp de mallow
```

Cp de Mallow



- Ajustando el modelo propuesto por el cp de mallow

```
modelo_multiple_2 = glm(data = Train_copy_1_model,
                        formula = Loan.Status~Term+Home.Ownership+Purpose+Current.Loan.Amount+
                        Credit.Score+Annual.Income+Monthly.Debt, family = "binomial")
summary(modelo_multiple_2)
```

```
##
## Call:
## glm(formula = Loan.Status ~ Term + Home.Ownership + Purpose +
##       Current.Loan.Amount + Credit.Score + Annual.Income + Monthly.Debt,
##       family = "binomial", data = Train_copy_1_model)
##
## Deviance Residuals:
##      Min        1Q     Median        3Q       Max
## -2.7317    0.4966   0.6038   0.6823   1.4785
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -2.199e+00  6.519e-01 -3.373 0.000743 ***
## TermShort Term            3.572e-01  4.007e-02  8.915 < 2e-16 ***
## Home.OwnershipHome Mortgage 8.504e-01  4.936e-01  1.723 0.084887 .
## Home.OwnershipOwn Home   7.764e-01  4.951e-01  1.568 0.116844
## Home.OwnershipRent       5.791e-01  4.925e-01  1.176 0.239689
## Purposebuy a car          1.055e+00  2.172e-01  4.858 1.19e-06 ***
## Purposebuy house           2.813e-01  1.929e-01  1.458 0.144729
## Purposedebt consolidation 3.961e-01  1.100e-01  3.601 0.000317 ***
## Purposeeducational expenses 3.348e-01  5.120e-01  0.654 0.513189
```

```

## Purposehome_improvements    4.438e-01   1.258e-01   3.528 0.000419 ***
## Purposemajor_purchase      -2.026e-01   2.681e-01  -0.756 0.449647
## Purposemedical_bills       3.640e-01   1.735e-01   2.099 0.035844 *
## Purposemoving                -8.311e-02   3.589e-01  -0.232 0.816867
## Purposeother                  4.201e-01   1.186e-01   3.542 0.000397 ***
## Purposerenewable_energy     1.029e+01   9.838e+01   0.105 0.916731
## Purposesmall_business      -9.131e-01   2.695e-01  -3.389 0.000702 ***
## Purposetake_a_trip          9.960e-01   3.173e-01   3.139 0.001697 **
## Purposevacation              1.399e-01   3.782e-01   0.370 0.711491
## Purposewedding                -4.133e-02   4.559e-01  -0.091 0.927771
## Current.Loan.Amount        -4.945e-07   1.098e-07  -4.502 6.73e-06 ***
## Credit.Score                  2.974e-03   6.093e-04   4.882 1.05e-06 ***
## Annual.Income                 3.840e-07   2.988e-08  12.851 < 2e-16 ***
## Monthly.Debt                  -1.135e-05   1.739e-06  -6.529 6.61e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 27054  on 27690  degrees of freedom
## Residual deviance: 26449  on 27668  degrees of freedom
## AIC: 26495
##
## Number of Fisher Scoring iterations: 10

```

Dicho modelo presenta un AIC menor que el modelo no. 1 (26494.82 vs 26517.4)

modelo_multiple_2\$aic

```
## [1] 26494.82
```

```
confint(object = modelo_multiple_2, level = 0.95)
```

Waiting for profiling to be done...

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

Unknowns: all sites visited, probabilities summed over Oct 1, assumed

```

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

##                               2.5 %      97.5 %
## (Intercept)           -3.467323e+00 -9.011824e-01
## TermShort Term        2.786079e-01  4.356726e-01
## HomeOwnershipHome Mortgage -1.532728e-01 1.804839e+00
## HomeOwnershipOwn Home -2.300372e-01 1.733943e+00
## HomeOwnershipRent     -4.227244e-01 1.531458e+00
## Purposebuy a car       6.405532e-01 1.494459e+00
## Purposebuy house      -9.226115e-02 6.649472e-01
## Purposedebt consolidation 1.769887e-01 6.085260e-01
## Purposeeducational expenses -5.954735e-01 1.453975e+00
## Purposehome improvements 1.949434e-01 6.884150e-01
## Purposemajor_purchase   -7.174100e-01 3.372450e-01
## Purposemedical bills    2.669834e-02 7.074801e-01
## Purposemoving            -7.625726e-01 6.559944e-01
## Purposeother             1.847710e-01 6.499623e-01
## Purposerenewable_energy -9.928442e-01 NA
## Purposesmall_business   -1.439092e+00 -3.795108e-01
## Purposetake a trip       4.062209e-01 1.659619e+00
## Purposevacation          -5.675853e-01 9.298875e-01
## Purposewedding            -8.925065e-01 9.219148e-01
## Current.Loan.Amount      -7.095918e-07 -2.789976e-07
## Credit.Score              1.777891e-03 4.166454e-03
## Annual.Income              3.260880e-07 4.432245e-07
## Monthly.Debt              -1.475756e-05 -7.942124e-06

```

Ecuación para el modelo no. 2

- Identificando la ecuación

```

# Obtener el vector de coeficientes del modelo
coeficientes <- coef(modelo_multiple_2)

# Obtener los nombres de las variables del modelo
nombres_vars <- names(coeficientes)[-1]

# Crear una cadena con la ecuación del modelo
cadena_ecuacion <- paste(round(coeficientes[1], 4), "+",
                           paste(round(coeficientes[-1], 4), nombres_vars, sep = "*"), collapse = " + ")

# Imprimir la ecuación del modelo
cat("La ecuación del modelo es:\n", cadena_ecuacion, "\n")

## La ecuación del modelo es:
## -2.199 + 0.3572*TermShort Term + -2.199 + 0.8504*HomeOwnershipHome Mortgage + -2.199 + 0.7764*Home

```

- Identificando los coeficientes para el modelo no. 1 plus

```

nombres_vars = names(coef(modelo_multiple_2))
resumen = summary(modelo_multiple_2)

# Iterar por cada variable e imprimir su nombre y coeficiente
for (i in seq_along(nombres_vars)) {
  cat(paste(nombres_vars[i], ":", resumen$coefficients[i, "Estimate"], "\n"))
}

## (Intercept) : -2.19903318371546
## TermShort Term : 0.357207543809835
## Home.OwnershipHome Mortgage : 0.850409614169555
## Home.OwnershipOwn Home : 0.776406423785975
## Home.OwnershipRent : 0.579092664061572
## Purposebuy a car : 1.05518793620508
## Purposebuy house : 0.28127114189497
## Purposedebt consolidation : 0.396063005876388
## Purposeeducational expenses : 0.334781675771329
## Purposehome improvements : 0.443802853281349
## Purposemajor_purchase : -0.202647749809607
## Purposemedical bills : 0.364042141921484
## Purposemoving : -0.0831113304273208
## Purposeother : 0.420059979727
## Purposerenewable_energy : 10.2854378239952
## Purposesmall_business : -0.913114823529781
## Purposetake a trip : 0.996032770758514
## Purposevacation : 0.139868850921692
## Purposewedding : -0.0413300643715761
## Current.Loan.Amount : -4.94513623000562e-07
## Credit.Score : 0.00297442027087073
## Annual.Income : 3.84037139755036e-07
## Monthly.Debt : -1.13512091963771e-05

```

ODDS Ratios para el modelo no. 1 plus

- Identificando los odds ratios e^{B_k}

```

nombres_vars = names(coef(modelo_multiple_2))
resumen = summary(modelo_multiple_2)

# Iterar por cada variable e imprimir su nombre y el num de euler elevado al coeficiente
for (i in seq_along(nombres_vars)) {
  cat(paste(nombres_vars[i], ":", exp(resumen$coefficients[i, "Estimate"])), "\n"))

## (Intercept) : 0.110910336462635
## TermShort Term : 1.42933248818228
## Home.OwnershipHome Mortgage : 2.34060540073248
## Home.OwnershipOwn Home : 2.17364704733338
## Home.OwnershipRent : 1.7844186288578
## Purposebuy a car : 2.87251495273515
## Purposebuy house : 1.32481276748517
## Purposedebt consolidation : 1.48596293899926

```

```

## Purposeeducational expenses : 1.39763521428045
## Purposehome improvements : 1.55862317788542
## Purposemajor_purchase : 0.816565826239495
## Purposemedical bills : 1.43913486078846
## Purposemoving : 0.920248689837037
## Purposeother : 1.52205284519498
## Purposerenewable_energy : 29302.7844817085
## Purposesmall_business : 0.401272382753773
## Purposetake a trip : 2.70751914443604
## Purposevacation : 1.15012295140067
## Purposewedding : 0.959512376826538
## Current.Loan.Amount : 0.999999505486499
## Credit.Score : 1.00297884824798
## Annual.Income : 1.00000038403721
## Monthly.Debt : 0.999988648855228

```

Predicción y Matriz de confusión para el modelo no. 1 plus

- Predicción

Estableciendo una predicción con punto de corte arbitrario de 0.85

```

prediccion_2 = predict(object = modelo_multiple_2,
                      newdata = Train_copy_1_model, type = "response")

Prediccion_2<-as.factor(ifelse(prediccion_2<=0.85,yes = 1, no = 0))

table(Prediccion_2)

## Prediccion_2
##      0      1
## 6285 21406

print("Valores reales")

## [1] "Valores reales"

table(Train_copy_1_model$Loan.Status)

##
##      0      1
## 5305 22386

```

- Matriz de confusión para el modelo 1 plus con punto de corte de 0.85 (arbitrario)

```

confusionMatrix(reference = as.factor(Train_copy_1_model$Loan.Status), Prediccion_2)

## Confusion Matrix and Statistics
##
##          Reference

```

```

## Prediction      0      1
##             0    770  5515
##             1   4535 16871
##
##                  Accuracy : 0.6371
##                  95% CI : (0.6314, 0.6427)
##      No Information Rate : 0.8084
##      P-Value [Acc > NIR] : 1
##
##                  Kappa : -0.0945
##
## McNemar's Test P-Value : <2e-16
##
##                  Sensitivity : 0.14515
##                  Specificity : 0.75364
##      Pos Pred Value : 0.12251
##      Neg Pred Value : 0.78814
##      Prevalence : 0.19158
##      Detection Rate : 0.02781
##      Detection Prevalence : 0.22697
##      Balanced Accuracy : 0.44939
##
##      'Positive' Class : 0
##

```

```

Exactitud_2 <- vector()
Corte_2 <- seq(0.05, 0.95, by = 0.001)

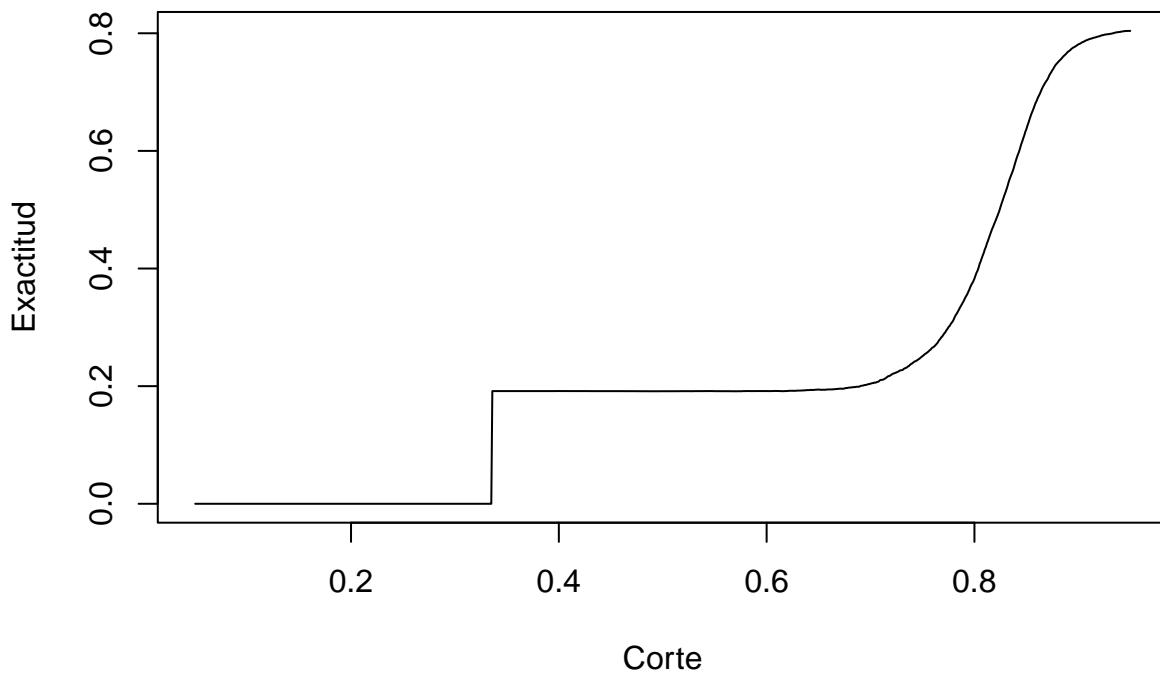
for (i in 1:length(Corte_2)) {
  Prediccion <- as.factor(ifelse(prediccion_2 <= Corte_2[i], yes = 1, no = 0))

  Exactitud_2[i] <- tryCatch(
    {
      confusionMatrix(data = as.factor(Train_copy_1_model$Loan.Status), Prediccion)$overall[1]
    },
    error = function(e) {
      0 # O cualquier valor predeterminado
    }
  )
}

plot(x=Corte_2, y = Exactitud_2, type = "l", main ="Punto de corte óptimo", xlab="Corte", ylab="Exactitud")

```

Modelo optimizado con el punto máximo de la curva de exactitud (punto de corte ideal)
Punto de corte óptimo



ubicando tal punto de corte óptimo

```
corte_optimo_2 = which(Exactitud_2==max(Exactitud_2))
Corte_2[corte_optimo_2]
```

```
## [1] 0.95
```

Predicción y matriz de confusión para el modelo no. 1 plus con punto de corte optimo

- Generando una nueva predicción con el punto de corte óptimo para el modelo ajustado

```
Prediccion_2_1<-as.factor(ifelse(prediccion_2<=Corte_2[corte_optimo_2],yes = 1, no = 0))
confusionMatrix(data = as.factor(Train_copy_1_model$Loan.Status),
                 Prediccion_2_1)
```

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction    0     1
##       0     12   5293
##       1    133  22253
##
##                  Accuracy : 0.8041
##                  95% CI : (0.7993, 0.8087)
##      No Information Rate : 0.9948
##      P-Value [Acc > NIR] : 1
##
```

```

##                               Kappa : -0.0059
##
##  McNemar's Test P-Value : <2e-16
##
##                               Sensitivity : 0.0827586
##                               Specificity : 0.8078487
##                               Pos Pred Value : 0.0022620
##                               Neg Pred Value : 0.9940588
##                               Prevalence : 0.0052364
##                               Detection Rate : 0.0004334
##  Detection Prevalence : 0.1915785
##                               Balanced Accuracy : 0.4453037
##
##                               'Positive' Class : 0
##

```

```

##### Comparar modelo ya ajustado (no.1) con uno nuevo y muy simple
Prediccion_2_roc<-predict(object = modelo_multiple_2,
                           newdata = Train_copy_1_model, type = "response",
                           se.fit = T)

predicciones_roc_2=prediction(Prediccion_2_roc$fit,Train_copy_1$Loan.Status)
Desempeno_roc_2<-performance(predicciones_roc_2,'tpr','fpr')

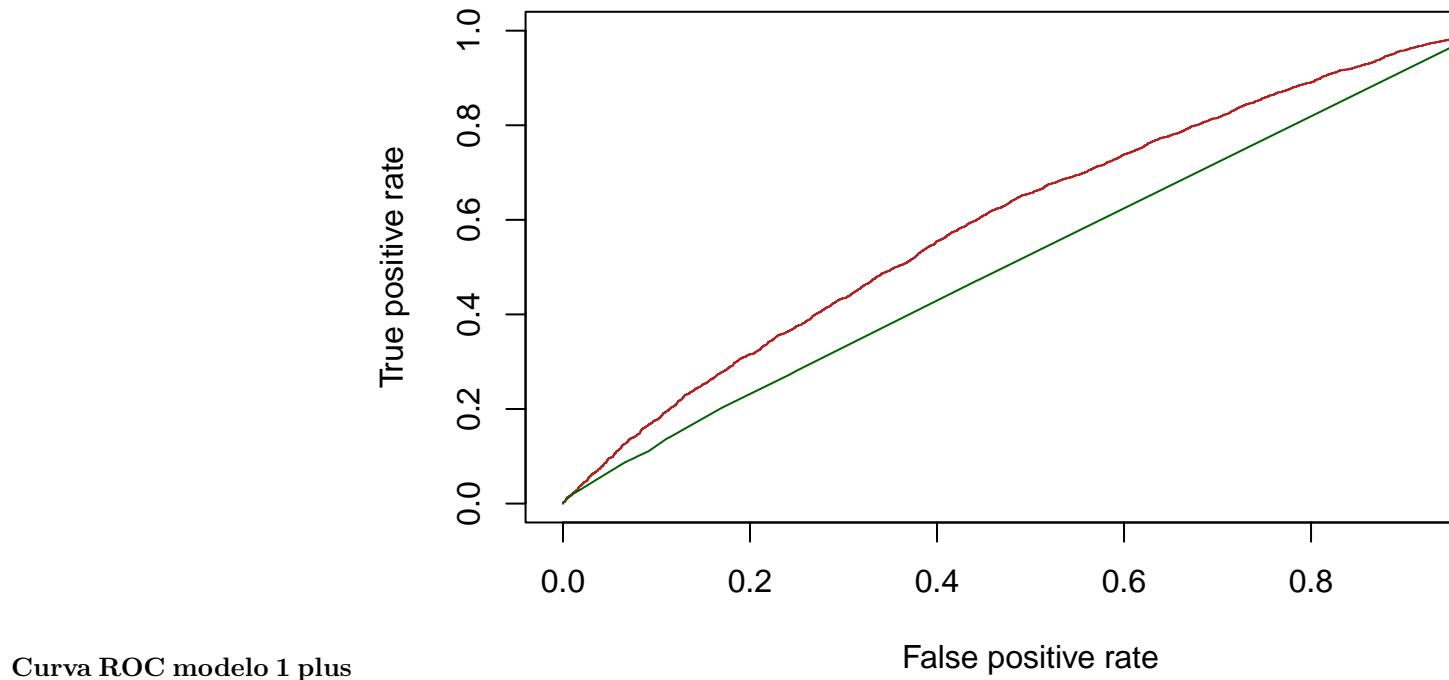
## Cgenerando nuevo modelo simple
modelo_simple_4 <- glm(data = Train_copy_1,
                        formula = Loan.Status~Purpose+Bankruptcies, family = "binomial")

Prediccion_3_simple<-predict(object = modelo_simple_4,
                               newdata = Train_copy_1_model, type = "response",
                               se.fit = T)

predicciones3_simple<-prediction(Prediccion_3_simple$fit,Train_copy_1_model$Loan.Status)
Desempeno3_simple<-performance(predicciones3_simple,'tpr','fpr')

#ejecutar en conjunto
plot(Desempeno_roc_2,col="firebrick")
plot(Desempeno3_simple, col = "darkgreen", add = T)

```



Aunque el accuracy del modelo no.2 bajo en un 0.01 punto, preferimos un modelo más ligero y más manejable, por lo que nos quedamos con este último modelo.

Cumplimiento de supuestos del modelo de regresión logística

Multicolinealidad (VIF) y ajuste del modelo • Multicolinealidad: se requiere de muy poca a ninguna multicolinealidad entre los predictores (para regresión logística múltiple).

Respuesta binaria: La variable dependiente ha de ser binaria.

Independencia: las observaciones han de ser independientes.

Linealidad entre la variable independiente y el logaritmo natural de odds.

Cross validation El modelo actual seleccionado (modelo no.2) podría sufrir de sobre-ajuste dado que se empleo toda la muestra para su modelado. Por lo que procedemos a hacer la validación cruzada del modelo para obtener un modelo que no solo sea efectivo en la muestr actual, si no en otras muestras no estudiadas antes.

Validación cruzada para los modelos: modelo no. 1, modelo no.1 plus y modelo no. 2 k pliegues

\$delta returns a vector of length two. The first component is the raw cross-validation estimate of prediction error. The second component is the adjusted cross-validation estimate. The adjustment is designed to compensate for the bias introduced by not using leave-one-out cross-validation.

```

#####1.3 k pliegues #####
MSE_kf_1 <- cv.glm(data = Train_copy_1_model,
                     glmfit = modelo_multiple_1,
                     K = 100)
MSE_kf_1$delta

## [1] 0.1518267 0.1518247

#####1.3 k pliegues #####
MSE_kf_1_plus <- cv.glm(data = Train_copy_1_model,
                         glmfit = modelo_multiple_1_plus,
                         K = 100)
MSE_kf_1_plus$delta

## [1] 0.1517703 0.1517682

#####1.3 k pliegues #####
MSE_kf_2 <- cv.glm(data = Train_copy_1_model,
                     glmfit = modelo_multiple_2,
                     K = 100)
MSE_kf_2$delta

## [1] 0.1515597 0.1515583

```

En la validación cruzada por k pliegues evaluada para los 3 modelos disponibles se puede interpretar que el error cuadrático medio (MSE, por sus siglas en inglés) del modelo es menor, lo cual es la intención, para el modelo multiple no.2, que a su vez resulta ser el que elegimos previamente porque tenía prácticamente el mismo accuracy en la matriz de confusión que el modelo no.1 plus (que mostró superioridad respecto al modelo no.1 en este mismo apartado), y además se reducían las variables del modelo.

MSPR

MSPR para el modelo multiple no. 1

```

set.seed(123)
indices <- createDataPartition(Train_copy_1_model$Loan.Status, p = 0.7, list = FALSE)
Train <- Train_copy_1_model[indices, ]
Test <- Train_copy_1_model[-indices, ]

# Obtener las predicciones en el conjunto de prueba
predicciones <- predict(modelo_multiple_1, newdata = Test, type = "response")

# Calcular el MSPR con un modelo de clasificación binaria
MSPR <- mean((Test$Loan.Status == 1) - predicciones)^2

# Imprimir el resultado
print(paste0("MSPR = ", MSPR))

## [1] "MSPR = 5.97616197972532e-08"

```

MSPR para el modelo multiple no. 1 plus

```

set.seed(123)
indices <- createDataPartition(Train_copy_1_model$Loan.Status, p = 0.7, list = FALSE)
Train <- Train_copy_1_model[indices, ]
Test <- Train_copy_1_model[-indices, ]

# Obtener las predicciones en el conjunto de prueba
predicciones <- predict(modelo_multiple_1_plus, newdata = Test, type = "response")

# Calcular el MSPR
MSPR <- mean((Test$Loan.Status == 1) - predicciones)^2

# Imprimir el resultado
print(paste0("MSPR = ", MSPR))

## [1] "MSPR = 5.3875591801177e-08"

#unique(Test$Loan)

```

MSPR para el modelo multiple no. 2

```

set.seed(123)
indices <- createDataPartition(Train_copy_1_model$Loan.Status, p = 0.7, list = FALSE)
Train <- Train_copy_1_model[indices, ]
Test <- Train_copy_1_model[-indices, ]

# Obtener las predicciones en el conjunto de prueba
predicciones <- predict(modelo_multiple_2, newdata = Test, type = "response")

# Calcular el MSPR
MSPR <- mean((Test$Loan.Status == 1) - predicciones)^2

# Imprimir el resultado
print(paste0("MSPR = ", MSPR))

## [1] "MSPR = 6.44167382157586e-10"

#unique(Test$Loan)

```

Bootstraping o remuestreo. Una vez que la validación cruzada ha demostrado que el mejor modelo es el modelo no. 2

Predicción de fully paid or Charged off para casos específicos Una vez que el modelo se ha ajustado, podemos emplearlo para predecir resultados individuales

```

#define two individuals
new <- data.frame(balance = 1400, income = 2000, student = c("Yes", "No"))
#
# #predict probability of defaulting
# predict(model, new, type="response")

#           1           2
# 0.02732106 0.04397747

```

Outliers y valores leverage en coeficientes (también distancia de cook)

Pruebas en Test

Aspectos a tomar en cuenta

- Al igual que en el caso de la regresión lineal, los resultados obtenidos usando solo un predictor pueden diferir respecto a aquellos obtenidos usando múltiples predictores, especialmente cuando existe correlación entre ellos. Este fenómeno se conoce como confusión (confounding).
- CONDICIONES DEL MODELO LOGÍSTICO • Respuesta binaria: La variable dependiente ha de ser binaria.
- Independencia: las observaciones han de ser independientes.
- Multicolinealidad: se requiere de muy poca a ninguna multicolinealidad entre los predictores (para regresión logística múltiple).
- Linealidad entre la variable independiente y el logaritmo natural de odds.

Apartado de dudas:

1. • En el caso de que se tenga más de un nivel en una variable ordinal o nominal en la ecuación del modelo, cómo se interpreta el valor de la primera variable que debería situarse en el intercepto, pero ahora hay más de una de estas variables representadas en el impacto del intercepto o constante.
— ejemplo: variable cualitativa termino y variable cualitativa proposito, en ambas se pierde un nivel representado en el intercepto, cómo se encuentra el impacto aislado de cada una?
2. • ¿Cómo comparar odds ratios entre si?

Caso de ejemplo: Purposebuy a car : 2.77158367672821 Purposebuy house : 1.28198643667157

¿cómo puedo explicar y en qué medida es más probable que alguien que pida el crédito para comprar un carro pague el crédito comparado con alguien que lo pida para casa?

3. • ¿Por qué el coeficiente y el odds ratio para Purposerenewable_energy es muy alto?
4. • EL modelo OLSRR que nos indica por estadístico las variables ideales, toma en cuenta interacciones?
5. En la selección de variables se recurrió al estadístico cp de mallow porque no encontramos escala del PRESS ¿Cómo emplear el PRESS? `plot(Subs, scale = c("Cp"), main="Cp de Mallow")#Cp de mallow`
6. ¿El calculo del MSPR se hace apropiadamente para un modelo de respuesta binaria?
7. ¿Cómo manejar la multicolinealidad en un modelo de regresión lineal?
8. Si los estadísticos de predicción arrojan buenos valores, ¿por qué la curva ROC no es la mejor?