Screening para detección rápida de impagos en créditos

[pedro.concejerocerezo@gmail.com](mailto:pedro.concejerocerezo@gmail.com)

17 de noviembre de 2016

# Introducción

Screening (se puede traducir como "cribado", o aunque sean dos palabras, "detección rápida") es una metodología ampliamente difundida en medicina y en otras ciencias. El objetivo es detectar cuanto antes la posibilidad de que aparezca un evento, que puede ser una enfermedad, un síntoma, u otros eventos que formen parte de un diagnóstico.

Screening incluye todo el conjunto de metodologías para calibrar estos instrumentos de detección, junto con la gestión de prácticas previas y posteriores a las pruebas concretas.

Quizás el mejor ejemplo de prueba de screening es el de detección del VIH, conocida como ELIZA. Aunque es poco sabido, se trata efectivamente de una prueba de screening, con un porcentaje considerable de falsos positivos. Es importante recalcar que aunque es un paso en el diagnóstico, el resultado del screening *no es el diagnóstico*. Precisamente el ELIZA, sólo en el caso de resultar positivo, requiere un segundo test denominado Western Blot que es mucho más costoso y que sí se considera diagnóstico.

De lo que se trata precisamente es de tener mecanismos rápidos baratos y lo más fácilmente aplicables posibles para filtrar aquellos elementos o individuos en riesgo y concentrar el esfuerzo diagnóstico en los que realmente lo necesitan.

El [screening en medicina](https://en.wikipedia.org/wiki/Screening_%28medicine%29) tiene infinidad de aplicaciones, y el propósito concreto de esta presentación es dara a conocer esta metodología para su aplicación en áreas de negocio. En concreto en la detección de impagos en créditos, una aplicación clásica de los modelos predictivos.

# Preparación del entorno R

## Loading required package: XLConnectJars

## XLConnect 0.2-12 by Mirai Solutions GmbH [aut],  
## Martin Studer [cre],  
## The Apache Software Foundation [ctb, cph] (Apache POI, Apache Commons  
## Codec),  
## Stephen Colebourne [ctb, cph] (Joda-Time Java library),  
## Graph Builder [ctb, cph] (Curvesapi Java library)

## http://www.mirai-solutions.com ,  
## http://miraisolutions.wordpress.com

## Loading required package: gplots

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

## Warning: package 'pROC' was built under R version 3.3.2

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

# Propósito

* Proponer metodología de detección rápida de variables asociadas a evento que se quiere predecir
* Aplicar, y en su caso adaptar, técnicas estadísticas del campo epidemiología, médico (screening)
* Prototipar estas metodologías con un caso de negocio, en concreto detectar impagos ("defaults") o retrasos en créditos

# Datos crediticios

Hay muchos datos de credit scoring, algunos relativamente simples (German credit, o los datos en UCI) y otros de competiciones limitadas (p.ej. [esta competición de kaggle](https://inclass.kaggle.com/c/to-loan-or-not-to-loan-that-is-the-question/data). O también estos [Datos de Prospectus](https://www.lendingclub.com/info/download-data.action).

Pero me ha parecido más interesante este conjunto de datos, de una startup del campo "fintech" llamada [LendingClub](https://www.lendingclub.com/public/how-peer-lending-works.action), muy interesante porque utiliza la filosofía del "crowdfunding" para facilitar y abaratar el crédito.

Extraído directamente de la descripción de sus actividades:

Lending Club uses technology to operate a credit marketplace at a lower cost than traditional bank loan programs, passing the savings on to borrowers in the form of lower rates and to investors in the form of solid returns. Borrowers who used a personal loan via Lending Club to consolidate debt or pay off high interest credit cards report in a survey that the interest rate on their loan was an average of 33% lower than they were paying on their outstanding debt or credit cards.1

By providing borrowers with better rates, and investors with attractive, risk-adjusted returns, Lending Club has earned among the highest satisfaction ratings in the financial services industry.2

Here's how it works:

Customers interested in a loan complete a simple application at LendingClub.com We leverage online data and technology to quickly assess risk, determine a credit rating and assign appropriate interest rates. Qualified applicants receive offers in just minutes and can evaluate loan options with no impact to their credit score Investors ranging from individuals to institutions select loans in which to invest and can earn monthly returns

# Descarga de datasets

De su propia descripción del dataset:

Estos ficheros contienen todos los datos de los préstamos emitidos en el periodo de tiempo establecido, incluyendo el último status del préstamos (Current, Late, Fully Paid, etc.) y la última información de pagos.

url <- "https://resources.lendingclub.com/LoanStats3a.csv.zip"  
  
descargado <- "LoanStats3a.csv.zip"  
  
if (file.exists(descargado)){  
 print("El fichero de datos ya está descargado")  
} else {  
 print("Vamos a descargar el fichero")  
 download.file(url,  
 destfile = descargado,  
 mode = "wb")  
  
}

## [1] "El fichero de datos ya está descargado"

descargado.unz <- unzip(descargado)

Probamos a leer directamente desde el zip

data <- fread(descargado.unz,  
 skip = 1,  
 header = T,   
 sep = ",")

## Warning in fread(descargado.unz, skip = 1, header = T, sep = ","): Stopped  
## reading at empty line 39789 but text exists afterwards (discarded): Loans  
## that do not meet the credit policy

## Warning in fread(descargado.unz, skip = 1, header = T, sep = ","): Read  
## less rows (39786) than were allocated (42542). Run again with verbose=TRUE  
## and please report.

Observamos que las variables aparecen como cadenas de caracteres, posteriormente las asignaremos clases.

También podemos descargarnos el diccionario

diccio.url <- "https://resources.lendingclub.com/LCDataDictionary.xlsx"  
  
descargado <- "LCDataDictionary.xlsx"  
  
if (file.exists(descargado)){  
 print("El fichero de datos ya está descargado")  
} else {  
 print("Vamos a descargar el fichero")  
 download.file(diccio.url,  
 destfile = descargado,  
 mode = "wb")  
}

## [1] "El fichero de datos ya está descargado"

diccio <- readWorksheetFromFile(descargado,  
 sheet = 1,  
 header = T,  
 startRow = 1)

Lamentablemente no contiene explícitas las clases de las variables descargadas. Lo malo: no está en el mismo orden que los datos, por lo que sólo lo usaremos para consultar el significado de columnas concretas.

View(diccio)

Y estas pueden ser las clases de las columnas. Convertimos en principio solo las numéricas

clases <- c(rep("character", 2), #los campos id   
 rep("numeric", 3), #cantidades concedidas en credito  
 rep("character", 2), #term = duración e int\_rate - habrá que convertir a numérico  
 rep("numeric", 1), #installment = The monthly payment owed by the borrower if the loan originates  
 rep("factor", 2), #grade and subgrade  
 rep("character", 3), #empleo y tiempo en empleo -habrá que convertirlo en numérico  
 rep("numeric", 1), #annual\_income, self-reported  
 rep("character",10), #empleo y tiempo en empleo -habrá que convertirlo en numérico  
 rep("numeric", 2), # dti ratio monthly debt payments on total debt, exc mortgage loan, divided by monthly income  
 rep("character", 1), #The month the borrower's earliest reported credit line was opened  
 rep("numeric", 6),  
 rep("character", 1), #porcentaje que habrá que convertir  
 rep("numeric", 1),  
 rep("character", 1),  
 rep("numeric", 9),  
 rep("character", 1), #una fecha  
 rep("numeric", 1),  
 rep("character", 2),  
 rep("numeric", 3),  
 rep("character", 1),  
 rep("numeric", 2),  
 rep("character", 1),  
 rep("numeric", 55)  
)  
  
numericas <- grep("numeric", clases)  
  
data <- as.data.frame(data)  
  
data[, numericas] <- lapply(data[, numericas], function(x) as.numeric(x))

La mitad de la tabla no tiene datos, columnas collections\_12\_mths\_ex\_med [50] y siguientes. Perdemos del orden de 50 variables.

#summary(data)  
dim(data)

## [1] 39786 111

data <- data[,colSums(is.na(data)) < nrow(data)]  
  
dim(data)

## [1] 39786 58

summary(data)

## id member\_id loan\_amnt funded\_amnt   
## Length:39786 Length:39786 Min. : 500 Min. : 500   
## Class :character Class :character 1st Qu.: 5500 1st Qu.: 5400   
## Mode :character Mode :character Median :10000 Median : 9650   
## Mean :11231 Mean :10959   
## 3rd Qu.:15000 3rd Qu.:15000   
## Max. :35000 Max. :35000   
##   
## funded\_amnt\_inv term int\_rate installment   
## Min. : 0 Length:39786 Length:39786 Min. : 15.69   
## 1st Qu.: 5000 Class :character Class :character 1st Qu.: 167.08   
## Median : 8975 Mode :character Mode :character Median : 280.61   
## Mean :10409 Mean : 324.73   
## 3rd Qu.:14400 3rd Qu.: 430.78   
## Max. :35000 Max. :1305.19   
##   
## grade sub\_grade emp\_title   
## Length:39786 Length:39786 Length:39786   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## emp\_length home\_ownership annual\_inc   
## Length:39786 Length:39786 Min. : 4000   
## Class :character Class :character 1st Qu.: 40500   
## Mode :character Mode :character Median : 59000   
## Mean : 68979   
## 3rd Qu.: 82342   
## Max. :6000000   
##   
## verification\_status issue\_d loan\_status   
## Length:39786 Length:39786 Length:39786   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## pymnt\_plan url desc   
## Length:39786 Length:39786 Length:39786   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## purpose title zip\_code   
## Length:39786 Length:39786 Length:39786   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## addr\_state dti delinq\_2yrs earliest\_cr\_line   
## Length:39786 Min. : 0.00 Min. : 0.0000 Length:39786   
## Class :character 1st Qu.: 8.18 1st Qu.: 0.0000 Class :character   
## Mode :character Median :13.41 Median : 0.0000 Mode :character   
## Mean :13.32 Mean : 0.1465   
## 3rd Qu.:18.60 3rd Qu.: 0.0000   
## Max. :29.99 Max. :11.0000   
##   
## inq\_last\_6mths mths\_since\_last\_delinq mths\_since\_last\_record  
## Min. :0.000 Min. : 0.0 Min. : 0.00   
## 1st Qu.:0.000 1st Qu.: 18.0 1st Qu.: 22.50   
## Median :1.000 Median : 34.0 Median : 90.00   
## Mean :0.869 Mean : 35.9 Mean : 69.75   
## 3rd Qu.:1.000 3rd Qu.: 52.0 3rd Qu.:104.00   
## Max. :8.000 Max. :120.0 Max. :129.00   
## NA's :25727 NA's :36995   
## open\_acc pub\_rec revol\_bal revol\_util   
## Min. : 2.000 Min. :0.00000 Min. : 0 Length:39786   
## 1st Qu.: 6.000 1st Qu.:0.00000 1st Qu.: 3704 Class :character   
## Median : 9.000 Median :0.00000 Median : 8860 Mode :character   
## Mean : 9.294 Mean :0.05514 Mean : 13392   
## 3rd Qu.:12.000 3rd Qu.:0.00000 3rd Qu.: 17065   
## Max. :44.000 Max. :4.00000 Max. :149588   
##   
## total\_acc initial\_list\_status out\_prncp out\_prncp\_inv   
## Min. : 2.00 Length:39786 Min. : 0.00 Min. : 0.00   
## 1st Qu.:13.00 Class :character 1st Qu.: 0.00 1st Qu.: 0.00   
## Median :20.00 Mode :character Median : 0.00 Median : 0.00   
## Mean :22.09 Mean : 20.84 Mean : 20.77   
## 3rd Qu.:29.00 3rd Qu.: 0.00 3rd Qu.: 0.00   
## Max. :90.00 Max. :5413.85 Max. :5413.85   
##   
## total\_pymnt total\_pymnt\_inv total\_rec\_prncp total\_rec\_int   
## Min. : 0 Min. : 0 Min. : 0 Min. : 0   
## 1st Qu.: 5582 1st Qu.: 5120 1st Qu.: 4602 1st Qu.: 663   
## Median : 9931 Median : 9308 Median : 8000 Median : 1353   
## Mean :12208 Mean :11622 Mean : 9835 Mean : 2276   
## 3rd Qu.:16608 3rd Qu.:15865 3rd Qu.:13833 3rd Qu.: 2845   
## Max. :58564 Max. :58564 Max. :35000 Max. :23770   
##   
## total\_rec\_late\_fee recoveries collection\_recovery\_fee  
## Min. : 0.000 Min. : 0.00 Min. : 0.00   
## 1st Qu.: 0.000 1st Qu.: 0.00 1st Qu.: 0.00   
## Median : 0.000 Median : 0.00 Median : 0.00   
## Mean : 1.384 Mean : 95.93 Mean : 12.54   
## 3rd Qu.: 0.000 3rd Qu.: 0.00 3rd Qu.: 0.00   
## Max. :180.200 Max. :29623.35 Max. :7002.19   
##   
## last\_pymnt\_d last\_pymnt\_amnt next\_pymnt\_d   
## Length:39786 Min. : 0.0 Length:39786   
## Class :character 1st Qu.: 218.6 Class :character   
## Mode :character Median : 547.9 Mode :character   
## Mean : 2677.9   
## 3rd Qu.: 3291.7   
## Max. :36115.2   
##   
## last\_credit\_pull\_d collections\_12\_mths\_ex\_med policy\_code  
## Length:39786 Min. :0 Min. :1   
## Class :character 1st Qu.:0 1st Qu.:1   
## Mode :character Median :0 Median :1   
## Mean :0 Mean :1   
## 3rd Qu.:0 3rd Qu.:1   
## Max. :0 Max. :1   
## NA's :56   
## application\_type verification\_status\_joint acc\_now\_delinq  
## Length:39786 Length:39786 Min. :0   
## Class :character Class :character 1st Qu.:0   
## Mode :character Mode :character Median :0   
## Mean :0   
## 3rd Qu.:0   
## Max. :0   
##   
## chargeoff\_within\_12\_mths delinq\_amnt pub\_rec\_bankruptcies tax\_liens   
## Min. :0 Min. :0 Min. :0.0000 Min. :0   
## 1st Qu.:0 1st Qu.:0 1st Qu.:0.0000 1st Qu.:0   
## Median :0 Median :0 Median :0.0000 Median :0   
## Mean :0 Mean :0 Mean :0.0433 Mean :0   
## 3rd Qu.:0 3rd Qu.:0 3rd Qu.:0.0000 3rd Qu.:0   
## Max. :0 Max. :0 Max. :2.0000 Max. :0   
## NA's :56 NA's :697 NA's :39

También tenemos algunas variables constantes, las quitamos

dim(data)

## [1] 39786 58

data <- data[ ,!apply(data,   
 MARGIN = 2,   
 function(x) max(x, na.rm = TRUE) == min(x, na.rm = TRUE))]  
dim(data)

## [1] 39786 49

# Definición del objetivo de detección: impago

Lo importante ahora es definir adecuadamente el impago, y en concreto tenemos varias definiciones posibles de default

table(data$loan\_status)/nrow(data)\*100

##   
## Charged Off Current Default   
## 14.203488664 1.676469110 0.002513447   
## Fully Paid In Grace Period Late (16-30 days)   
## 84.019504348 0.052782386 0.015080682   
## Late (31-120 days)   
## 0.030161363

Debemos informarnos sobre el significado de los estados del crédito:

[Loan status explained](http://help.bitbond.com/knowledgebase/articles/515476-the-10-loan-status-variants-explained) [Understanding loan status](https://www.orchardplatform.com/blog/understanding-loan-statuses/)

kk <- data[, numericas[1:24]]  
  
kk <- cbind(kk, data$loan\_status)  
  
kk$loan\_status\_orig <- kk$loan\_status  
  
kk$loan\_status <- "Y"  
  
kk$loan\_status[kk$'data$loan\_status' %in% c("Default",  
 "Late (16-30 days)",  
 "Late (31-120 days)",  
 "Charged Off")] <- "N"  
  
  
kk$loan\_status <- as.factor(kk$loan\_status)  
  
summary(kk)

## loan\_amnt funded\_amnt funded\_amnt\_inv installment   
## Min. : 500 Min. : 500 Min. : 0 Min. : 15.69   
## 1st Qu.: 5500 1st Qu.: 5400 1st Qu.: 5000 1st Qu.: 167.08   
## Median :10000 Median : 9650 Median : 8975 Median : 280.61   
## Mean :11231 Mean :10959 Mean :10409 Mean : 324.73   
## 3rd Qu.:15000 3rd Qu.:15000 3rd Qu.:14400 3rd Qu.: 430.78   
## Max. :35000 Max. :35000 Max. :35000 Max. :1305.19   
##   
## annual\_inc dti delinq\_2yrs inq\_last\_6mths   
## Min. : 4000 Min. : 0.00 Min. : 0.0000 Min. :0.000   
## 1st Qu.: 40500 1st Qu.: 8.18 1st Qu.: 0.0000 1st Qu.:0.000   
## Median : 59000 Median :13.41 Median : 0.0000 Median :1.000   
## Mean : 68979 Mean :13.32 Mean : 0.1465 Mean :0.869   
## 3rd Qu.: 82342 3rd Qu.:18.60 3rd Qu.: 0.0000 3rd Qu.:1.000   
## Max. :6000000 Max. :29.99 Max. :11.0000 Max. :8.000   
##   
## mths\_since\_last\_delinq mths\_since\_last\_record open\_acc   
## Min. : 0.0 Min. : 0.00 Min. : 2.000   
## 1st Qu.: 18.0 1st Qu.: 22.50 1st Qu.: 6.000   
## Median : 34.0 Median : 90.00 Median : 9.000   
## Mean : 35.9 Mean : 69.75 Mean : 9.294   
## 3rd Qu.: 52.0 3rd Qu.:104.00 3rd Qu.:12.000   
## Max. :120.0 Max. :129.00 Max. :44.000   
## NA's :25727 NA's :36995   
## pub\_rec revol\_bal total\_acc out\_prncp\_inv   
## Min. :0.00000 Min. : 0 Min. : 2.00 Min. : 0.00   
## 1st Qu.:0.00000 1st Qu.: 3704 1st Qu.:13.00 1st Qu.: 0.00   
## Median :0.00000 Median : 8860 Median :20.00 Median : 0.00   
## Mean :0.05514 Mean : 13392 Mean :22.09 Mean : 20.77   
## 3rd Qu.:0.00000 3rd Qu.: 17065 3rd Qu.:29.00 3rd Qu.: 0.00   
## Max. :4.00000 Max. :149588 Max. :90.00 Max. :5413.85   
##   
## total\_pymnt total\_pymnt\_inv total\_rec\_prncp total\_rec\_int   
## Min. : 0 Min. : 0 Min. : 0 Min. : 0   
## 1st Qu.: 5582 1st Qu.: 5120 1st Qu.: 4602 1st Qu.: 663   
## Median : 9931 Median : 9308 Median : 8000 Median : 1353   
## Mean :12208 Mean :11622 Mean : 9835 Mean : 2276   
## 3rd Qu.:16608 3rd Qu.:15865 3rd Qu.:13833 3rd Qu.: 2845   
## Max. :58564 Max. :58564 Max. :35000 Max. :23770   
##   
## total\_rec\_late\_fee recoveries collection\_recovery\_fee  
## Min. : 0.000 Min. : 0.00 Min. : 0.00   
## 1st Qu.: 0.000 1st Qu.: 0.00 1st Qu.: 0.00   
## Median : 0.000 Median : 0.00 Median : 0.00   
## Mean : 1.384 Mean : 95.93 Mean : 12.54   
## 3rd Qu.: 0.000 3rd Qu.: 0.00 3rd Qu.: 0.00   
## Max. :180.200 Max. :29623.35 Max. :7002.19   
##   
## last\_pymnt\_d next\_pymnt\_d data$loan\_status  
## Length:39786 Length:39786 Charged Off : 5651   
## Class :character Class :character Current : 667   
## Mode :character Mode :character Default : 1   
## Fully Paid :33428   
## In Grace Period : 21   
## Late (16-30 days) : 6   
## Late (31-120 days): 12   
## loan\_status  
## N: 5670   
## Y:34116   
##   
##   
##   
##   
##

table(kk$loan\_status)/nrow(kk)\*100

##   
## N Y   
## 14.25124 85.74876

# Screening mediante Wilcoxon (ROCR)

Carguemos la función wilcoxon

# v.01 28/08/2012  
# v.02 16/10/2012 Absolute AUC value and decreasing ordenation   
# Purpose of program, inputs, and outputs   
# R code for computing ROC AUC's (same as wilcoxon)  
# source() and library() statements   
# Dependencies: ROCR package  
  
wmw <- function(object, ini.col, fin.col, num.target, write.out=NA,   
 write.name=NA){  
 options(warn=-1)  
 library(ROCR)  
 df <- data.frame()  
   
 for (i in ini.col:fin.col){  
 pred <- prediction(object[, c(i)], object[, c(num.target)])  
   
 #Calculate the area under curve in the diagram  
 perf <- performance(pred, "auc")  
 auc <- as.numeric(perf@y.values)  
   
 df[i, 1] <- names(object)[i]  
 if (auc < 0.500) {auc <- 1-auc}  
 df[i, 2] <- auc  
 }  
 names(df) <- c("var", "score")  
 df <- df[order(df$score, decreasing=TRUE, na.last=TRUE), ]  
 rm(pred, perf, auc)  
   
 if(write.out == 'TRUE'){  
 write.table(df, file=write.name, quote = FALSE)  
 }  
 return(df)  
 }

Aquí disponemos de una función que en realidad implementa un bucle, primero hay que definir el target, bueno en realidad el número de variable en la tabla

function(object, ini.col, fin.col, num.target, write.out=NA, write.name=NA)

Según parece, la función requiere columnas contiguas, así que vamos a extraer las columnas que nos interesan.

Intentemos ahora la función wilcoxon

wmw(kk, #el objeto sobre el que se va a calcular el wilcoxon  
 1, 22, #columnas contiguas  
 26, #num de columna target  
 write.out = "TRUE",  
 write.name = "primera\_prueba.txt"  
)

## var score  
## 21 recoveries 0.8742504  
## 22 collection\_recovery\_fee 0.8358025  
## 18 total\_rec\_prncp 0.8314364  
## 16 total\_pymnt 0.7396008  
## 17 total\_pymnt\_inv 0.7381785  
## 20 total\_rec\_late\_fee 0.5592749  
## 5 annual\_inc 0.5580997  
## 8 inq\_last\_6mths 0.5550193  
## 6 dti 0.5362873  
## 10 mths\_since\_last\_record 0.5346797  
## 1 loan\_amnt 0.5330777  
## 2 funded\_amnt 0.5317769  
## 14 total\_acc 0.5208061  
## 19 total\_rec\_int 0.5183198  
## 12 pub\_rec 0.5173325  
## 4 installment 0.5168920  
## 3 funded\_amnt\_inv 0.5154973  
## 13 revol\_bal 0.5123279  
## 11 open\_acc 0.5108786  
## 7 delinq\_2yrs 0.5087369  
## 15 out\_prncp\_inv 0.5084110  
## 9 mths\_since\_last\_delinq 0.5009599

Hay variables que tienen una relación muy fuerte con el target, porque dependen de él. Es el caso de las recuperaciones después de "Charge Off"

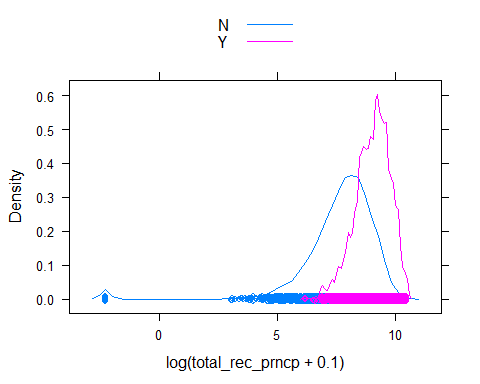
22 recoveries 0.8742504 23 collection\_recovery\_fee 0.8358025

Aquí me doy cuenta de una cosa: aquellos "loans" con "charge off" es porque ya están vencidos. En las páginas que he consultado sobre el estado de préstamos, éstos se "dan por perdidos", se plantea incluso su venta a recuperadores -con su coste adicional.

Pero también tenemos variables que no 19 total\_rec\_prncp 0.8314364

Intentaremos responder a las siguientes preguntas: - ¿Tenemos alguna manera de descartar las predictoras que son estrictamente inútiles? (contraste estadístico de capacidad predictiva cero) - Una vez elegida una variable, ¿qué rango / valores seleccionamos para una observación más detallada? - ¿Cómo podemos dibujar las ROC de la mejor manera para tomar decisiones basadas en ellas? - ¿Qué ventajas tiene este enfoque frente al tradicional del modelo? (Del que por otro lado es totalmente complementario) - extraer rápidamente óptimos de punto corte a partir de cruce sensib. especif. (posteriormente modificarlos con prevalencia)

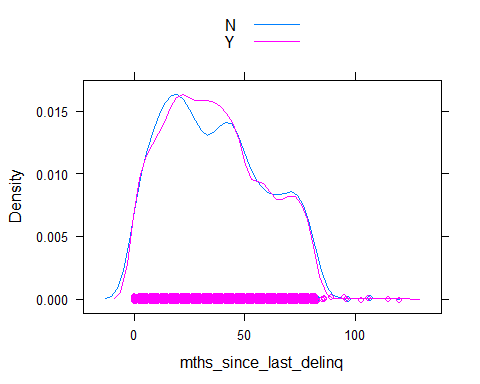
d1 <- densityplot( ~ log(total\_rec\_prncp + 0.1), #Para evitar tener los 0's originales como NAs  
 data = kk,  
 groups = loan\_status,  
 auto.key = TRUE)  
  
print(d1)



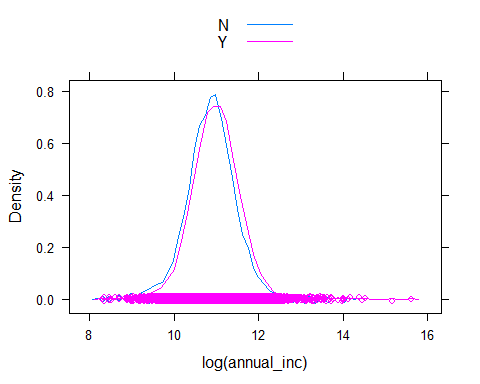
# Variables sin apenas capacidad predictiva

Las siguientes variables

d1 <- densityplot( ~ mths\_since\_last\_delinq,   
 data = kk,  
 groups = loan\_status,  
 auto.key = TRUE)  
  
print(d1)

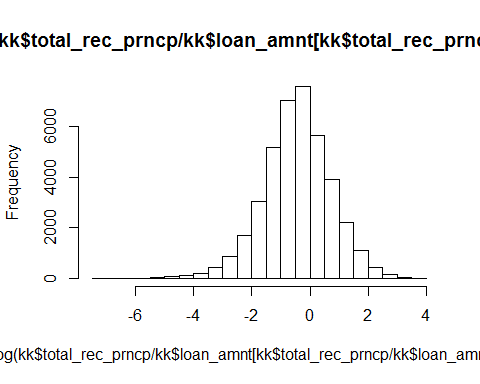


d1 <- densityplot( ~ log(annual\_inc),   
 data = kk,  
 groups = loan\_status,  
 auto.key = TRUE)  
  
print(d1)



Veamos a partir de lo que ya sabemos, si podemos detectar los créditos vivos. Estos serán aquellos cuya cantidad recuperada (total\_rec\_prncp) sea inferior al total del préstamo (loan\_amnt), y veamos la distribución en aquellos cuya razón entre ambas sea menor a 1 (créditos en principio vivos).

hist(log(kk$total\_rec\_prncp/ kk$loan\_amnt[kk$total\_rec\_prncp/ kk$loan\_amnt < 1]))



summary(log((kk$total\_rec\_prncp/ kk$loan\_amnt[kk$total\_rec\_prncp/ kk$loan\_amnt < 1])+ 0.01))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -4.6050 -1.1090 -0.3906 -0.4176 0.2952 3.5560

Tenemos una bonita curva normal. Veamos qué tal distingue entre loan\_status Y ó N.

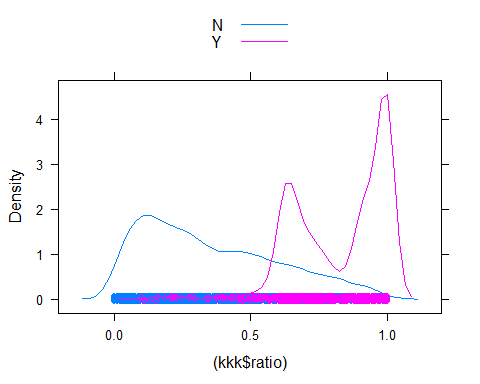
kkk <- kk[kk$total\_rec\_prncp/ kk$loan\_amnt < 1,]  
  
dim(kkk)

## [1] 8740 26

kkk$ratio <- (kkk$total\_rec\_prncp/ kkk$loan\_amnt)   
  
table(kkk$loan\_status)

##   
## N Y   
## 5670 3070

d1 <- densityplot( ~ (kkk$ratio),   
 data = kkk,  
 groups = loan\_status,  
 auto.key = TRUE)  
  
print(d1)



Veamos el funcionamiento de WmW

# Necesitamos columnas contiguas  
  
k4 <- kkk[, c(1:20,  
 27,  
 26,  
 25)]  
  
wmw(k4, #el objeto sobre el que se va a calcular el wilcoxon  
 1, 21, #columnas contiguas  
 22, #num de columna target  
 write.out = "TRUE",  
 write.name = "primera\_prueba.txt"  
)

## var score  
## 21 ratio 0.9295398  
## 18 total\_rec\_prncp 0.9145937  
## 16 total\_pymnt 0.8560074  
## 17 total\_pymnt\_inv 0.8141361  
## 19 total\_rec\_int 0.6958168  
## 1 loan\_amnt 0.6881520  
## 5 annual\_inc 0.6438779  
## 2 funded\_amnt 0.6192293  
## 15 out\_prncp\_inv 0.6104129  
## 3 funded\_amnt\_inv 0.6040166  
## 4 installment 0.5995331  
## 13 revol\_bal 0.5958787  
## 14 total\_acc 0.5822597  
## 11 open\_acc 0.5621334  
## 20 total\_rec\_late\_fee 0.5558994  
## 8 inq\_last\_6mths 0.5486853  
## 10 mths\_since\_last\_record 0.5411684  
## 12 pub\_rec 0.5222824  
## 9 mths\_since\_last\_delinq 0.5181727  
## 6 dti 0.5097818  
## 7 delinq\_2yrs 0.5053577

#table(k4$`data$loan\_status`)

# CONTRASTE ESTADÍSTICO DE CURVAS ROC

El paquete [pROC](https://cran.r-project.org/web/packages/pROC/index.html) proporciona métodos para el contraste estadístico de curvas ROC. Probemos por ejemplo si con nuestra proporción de pagado versus obtenido ("ratio") mejoramos la capacidad predictiva de uno de sus componentes ("total\_rec\_prncp").

Podemos primero crear dos objetos ROC:

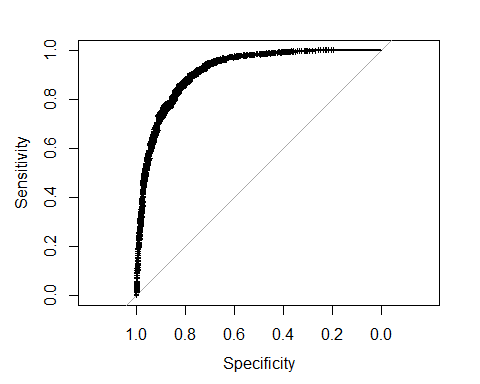
roc1 <- pROC::roc(response = k4$loan\_status,  
 predictor = k4$total\_rec\_prncp,  
 auc = TRUE,  
 ci = TRUE)

Con pROC podemos dibujar fácilmente la curva y además sus intervalos de confianza mediante "bootstrap"

plot(roc1)

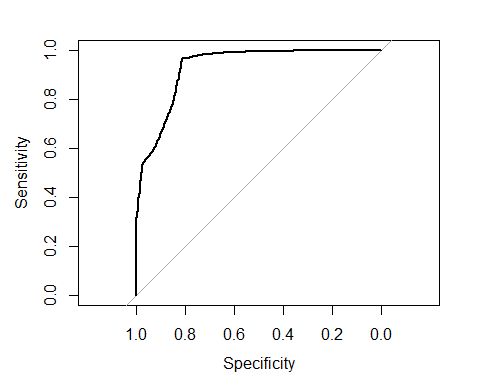
##   
## Call:  
## roc.default(response = k4$loan\_status, predictor = k4$total\_rec\_prncp, auc = TRUE, ci = TRUE)  
##   
## Data: k4$total\_rec\_prncp in 5670 controls (k4$loan\_status N) < 3070 cases (k4$loan\_status Y).  
## Area under the curve: 0.9146  
## 95% CI: 0.9088-0.9204 (DeLong)

ci.threshodls.obj <- ci.thresholds(roc1)  
plot(ci.threshodls.obj)



Segundo objeto ROC

roc2 <- pROC::roc(response = k4$loan\_status,  
 predictor = k4$ratio,  
 auc = TRUE,  
 ci = TRUE)  
  
plot(roc2)



##   
## Call:  
## roc.default(response = k4$loan\_status, predictor = k4$ratio, auc = TRUE, ci = TRUE)  
##   
## Data: k4$ratio in 5670 controls (k4$loan\_status N) < 3070 cases (k4$loan\_status Y).  
## Area under the curve: 0.9295  
## 95% CI: 0.9245-0.9345 (DeLong)

Y la comparación estadística entre ambas:

pROC::roc.test(roc1, roc2,  
 alternative = "two.sided")

##   
## DeLong's test for two correlated ROC curves  
##   
## data: roc1 and roc2  
## Z = -4.7206, p-value = 2.352e-06  
## alternative hypothesis: true difference in AUC is not equal to 0  
## sample estimates:  
## AUC of roc1 AUC of roc2   
## 0.9145937 0.9295398

# CONCLUSIONES

* screening puede ser mucho más rápido y más manejable que hacer modelos estadísticos
* ROC es una metodología muy avanzada y sin ningún lugar a dudas la mejor implementación es en librerías R
* limitaciones: objetivo de predicción binario y predictores cuantitativos
* ventajas: muy rápido y no paramétrico

# REFERENCIAS

* Pepe, Margaret: [procedimientos roccurve, comproc, rocreg y predcurve](http://research.fhcrc.org/diagnostic-biomarkers-center/en/software/rocbasic.html)
* Robin, Xavier, et al. (2011): pROC: an open-source package for R and S+ to analyze and compare ROC curves. Robin et al . BMC Bioinformatics 2011, 12:77. [enlace](http://www.biomedcentral.com/1471-2105/12/77)
* Tobias Sing, Oliver Sander, Niko Beerenwinkel, Thomas Lengauer (2005): ROCR: visualizing classifier performance in R. Bioinformatics 21(20):3940-3941. [sitio web de ROCR con toda la información](http://rocr.bioinf.mpi-sb.mpg.de/)