

## Práctica Final

El archivo adjunto Fraude.csv contiene información sobre muchas transacciones con tarjetas de crédito y débito por diferentes canales. Para cada transacción se tiene el valor monetario de la misma y otras variables (ver el archivo diccionario\_variables.xlsx). De particular importancia es la variable FRAUDE en donde aparece 1 si la transacción constituyó un fraude o 0 si fue una transacción legítima.

El objetivo de esta práctica es desarrollar un modelo que permita, a partir de estos datos, predecir cuál será el valor de la variable FRAUDE para una transacción cualquiera

\*Librerías:

```
library(dplyr)
library(DataExplorer)
library(Hmisc)
library(ggplot2)
library(data.table)
library(stringr)
library(corrplot)
library(missForest)
library(randomForest)
library(tidyverse)

#Library(tidyr)
library(ROCR)
library(caret)
library(glmnet)
library(MASS)
library(pROC)
library(lattice)
library(e1071)
```

- Análisis exploratorio de los datos.

Antes de empezar con el análisis exploratorio en forma de gráficos muestro como se forma el dataframe para tener una idea de como son los datos.

```

df_Fraude <- read.csv("Fraude.csv")
str(df_Fraude) # tipo de variables que forman el dataframe

## 'data.frame': 2965 obs. of 26 variables:
##   $ id          : num  9e+09 9e+09 9e+09 9e+09 9e+09 ...
##   $ FRAUDE     : int  1 1 1 1 1 1 1 1 0 1 ...
##   $ VALOR       : num  0 0 0 0 0 0 0 0 0 0 ...
##   $ HORA_AUX    : int  13 17 13 13 0 13 14 18 16 15 ...
##   $ Dist_max_NAL: num  659 595 659 659 1 ...
##   $ Canal1      : chr  "ATM_INT" "ATM_INT" "ATM_INT" "ATM_INT" ...
##   $ FECHA       : int  20150501 20150515 20150501 20150501 20150510
20150523 20150526 20150502 20150501 20150502 ...
##   $ COD_PAIS    : chr  "US" "US" "US" "US" ...
##   $ CANAL       : chr  "ATM_INT" "ATM_INT" "ATM_INT" "ATM_INT" ...
##   $ DIASEM      : int  5 5 5 5 0 6 2 6 5 6 ...
##   $ DIAMES      : int  1 15 1 1 10 23 26 2 1 2 ...
##   $ FECHA_VIN   : int  20120306 20050415 20120306 20120306 20141009
20150220 20080409 20040520 20150110 20090330 ...
##   $ OFICINA_VIN : int  392 716 392 392 788 547 210 454 297 46 ...
##   $ SEXO         : chr  "M" "M" "M" "M" ...
##   $ SEGMENTO    : chr  "Personal Plus" "Personal Plus" "Personal
Plus" "Personal Plus" ...
##   $ EDAD         : int  29 29 29 29 25 20 29 28 21 28 ...
##   $ INGRESOS    : int  1200000 5643700 1200000 1200000 0 4000000
2100000 2000000 500000 4000000 ...
##   $ EGRESOS     : int  1200000 500000 1200000 1200000 0 2500000
310000 200000 300000 1500000 ...
##   $ NROPAISES   : int  1 1 1 1 1 1 2 1 2 1 ...
##   $ Dist_Sum_INTER: num  NA NA NA NA NA ...
##   $ Dist_Mean_INTER: num  NA NA NA NA NA ...
##   $ Dist_Max_INTER: num  NA NA NA NA NA ...
##   $ NROCIUDADES  : int  6 5 6 6 1 1 5 3 1 9 ...
##   $ Dist_Mean_NAL : num  475 290 475 475 NA ...
##   $ Dist_HOY      : num  4552 4552 4552 4552 1482 ...
##   $ Dist_sum_NAL  : num  5224 2030 5224 5224 1 ...

dim(df_Fraude) # 2965 filas y 26 columnas

## [1] 2965 26

head(df_Fraude) #visualización primeras 6 filas del dataframe

##      id FRAUDE VALOR HORA_AUX Dist_max_NAL Canal1    FECHA COD_PAIS
CANAL
## 1 9e+09      1     0      13      659.13 ATM_INT 20150501      US
ATM_INT
## 2 9e+09      1     0      17      594.77 ATM_INT 20150515      US
ATM_INT
## 3 9e+09      1     0      13      659.13 ATM_INT 20150501      US
ATM_INT
## 4 9e+09      1     0      13      659.13 ATM_INT 20150501      US

```

```

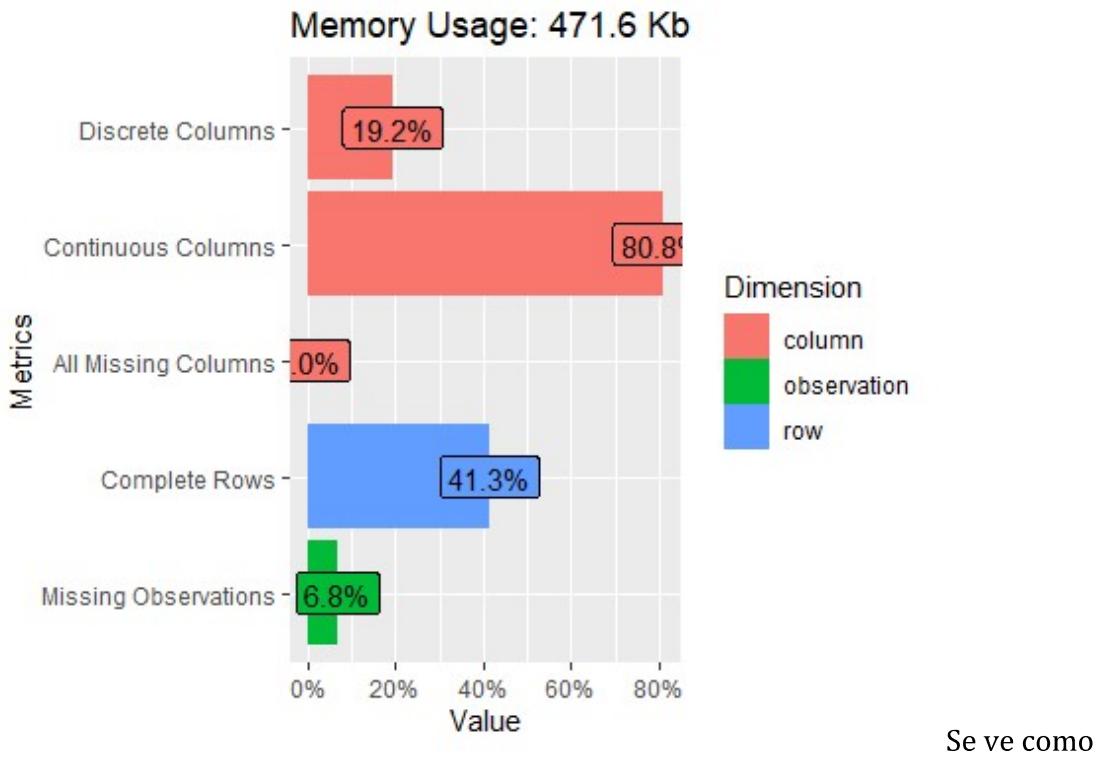
ATM_INT
## 5 9e+09      1     0      0      1.00 ATM_INT 20150510          CR
ATM_INT
## 6 9e+09      1     0     13      1.00 ATM_INT 20150523          US
ATM_INT
##  DIASEM DIAMES FECHA_VIN OFICINA_VIN SEXO      SEGMENTO EDAD INGRESOS
EGRESOS
## 1      5      1 20120306      392 M Personal Plus   29 1200000
1200000
## 2      5     15 20050415      716 M Personal Plus   29 5643700
500000
## 3      5      1 20120306      392 M Personal Plus   29 1200000
1200000
## 4      5      1 20120306      392 M Personal Plus   29 1200000
1200000
## 5      0     10 20141009      788 M Personal       25    0
0
## 6      6     23 20150220      547 M Emprendedor   20 4000000
2500000
##  NROPAISES Dist_Sum_INTER Dist_Mean_INTER Dist_Max_INTER NROCIUDADES
## 1      1           NA           NA           NA           6
## 2      1           NA           NA           NA           5
## 3      1           NA           NA           NA           6
## 4      1           NA           NA           NA           6
## 5      1           NA           NA           NA           1
## 6      1           NA           NA           NA           1
##  Dist_Mean_NAL Dist_HOY Dist_sum_NAL
## 1      474.94  4552.41  5224.36
## 2      289.99  4552.41  2029.90
## 3      474.94  4552.41  5224.36
## 4      474.94  4552.41  5224.36
## 5      NA      1482.35  1.00
## 6      NA      4552.41  1.00

```

El dataframe cuenta con 2965 observaciones y con 26 variables

También se puede estudiar la forma en la que se estructuran el dataset y los valores faltantes utilizando el siguiente gráfico.

```
plot_intro(df_Fraude)
```



Para realizar un análisis más completo aplico la función `describe()`. Una visualización estadística de las variables que forman el dataset.

```
describe(df_Fraude)

## df_Fraude
##
## 26 Variables      2965 Observations
## -----
## id
##   n    missing  distinct      Info      Mean      Gmd     .05
## .10
##   2965        0     2912      1 6.891e+09 6.276e+09 3.336e+08
## 7.363e+08
##   .25        .50     .75     .90     .95
## 2.553e+09 6.143e+09 9.000e+09 9.000e+09 9.550e+09
## 
## lowest : 2364560 2382108 4289561 4423003 12430697
## highest: 90900297404 91019048870 91032413560 91126594019 93300501209
## -----
## FRAUDE
##   n    missing  distinct      Info      Sum      Mean      Gmd
## 2965        0       2     0.557      731 0.2465 0.3716
## 
```

```

## -----
## VALOR
##      n  missing distinct      Info      Mean      Gmd      .05
.10
##    2965      0    2276      1 503570  638176  14705
33077
##    .25      .50      .75      .90      .95
##    90160  243591  505819 1087053 1690920
##
## lowest :      0.00    2410.09    2441.81    2529.01    2556.07
## highest: 11323766.46 12586915.88 13001774.37 13798933.28 20014064.66
## -----
## HORA_AUX
##      n  missing distinct      Info      Mean      Gmd      .05
.10
##    2965      0      24    0.996    14.96    6.905      1
3
##    .25      .50      .75      .90      .95
##    12      16      20      22      23
##
## lowest :  0  1  2  3  4, highest: 19 20 21 22 23
## -----
## Dist_max_NAL
##      n  missing distinct      Info      Mean      Gmd      .05
.10
##    2965      0    266    0.995    314.7    323.1    1.00
1.00
##    .25      .50      .75      .90      .95
##    24.83  243.62  594.77  698.52  741.95
##
## lowest :     1.00     4.48     7.57     7.94     9.23
## highest: 1182.29 1200.84 1217.57 1285.31 1310.46
## -----
## Canal1
##      n  missing distinct
##    2965      0      2
##
## Value      ATM_INT      POS
## Frequency    636    2329
## Proportion   0.215    0.785
## -----
## FECHA
##      n  missing distinct      Info      Mean      Gmd      .05
.10
##    2965      0      31    0.997 20150513    10.46 20150501

```

```

20150502
##      .25      .50      .75      .90      .95
## 20150504 20150515 20150521 20150526 20150529
##
## lowest : 20150501 20150502 20150503 20150504 20150505
## highest: 20150527 20150528 20150529 20150530 20150531
## -----
-----
## COD_PAIS
##      n missing distinct
##    2965      0      29
##
## lowest : AR AW BR CA CL, highest: PR PY SV US UY
## -----
##
## CANAL
##      n missing distinct
##    2965      0      3
##
## Value      ATM_INT      MCI      POS
## Frequency   636     1200     1129
## Proportion  0.215    0.405    0.381
## -----
##
## DIASEM
##      n missing distinct      Info      Mean      Gmd
##    2965      0      7     0.975     3.143     2.379
##
## lowest : 0 1 2 3 4, highest: 2 3 4 5 6
## 
## Value      0      1      2      3      4      5      6
## Frequency  512    305    337    439    284    641    447
## Proportion 0.173  0.103  0.114  0.148  0.096  0.216  0.151
## -----
##
## DIAMES
##      n missing distinct      Info      Mean      Gmd      .05
.10
##    2965      0      31     0.997    13.49    10.46      1
2
##      .25      .50      .75      .90      .95
##      4       15      21       26      29
##
## lowest : 1 2 3 4 5, highest: 27 28 29 30 31
## -----
##
## FECHA_VIN
##      n missing distinct      Info      Mean      Gmd      .05
.10
##    2941      24     1025      1 20009198    99239 19860710

```

```

19901023
##      .25      .50      .75      .90      .95
## 19951024 20011227 20080813 20120306 20130705
##
## lowest : 19111111 19721001 19740401 19760402 19780601
## highest: 20150220 20150224 20150226 20150302 20150427
## -----
## -----
## OFICINA_VIN
##      n missing distinct      Info      Mean      Gmd      .05
.10
##    2941      24      373       1    416.4    333.2      20
40
##      .25      .50      .75      .90      .95
##    168      360      659     841      929
##
## lowest :  1   2   4   5   6, highest: 944 945 961 963 967
## -----
## -----
## SEXO
##      n missing distinct
##    2910      55        2
##
## Value      F      M
## Frequency 1438 1472
## Proportion 0.494 0.506
## -----
## -----
## SEGMENTO
##      n missing distinct
##    2941      24        6
##
## lowest : Emprendedor      Empresarial      Personal      Personal Plus
## Preferencial
## highest: Empresarial      Personal      Personal Plus Preferencial  PYME
##
## Value      Emprendedor      Empresarial      Personal Personal Plus
## Frequency      159          4          174          1527
## Proportion      0.054      0.001      0.059      0.519
##
## Value      Preferencial      PYME
## Frequency      958          119
## Proportion      0.326      0.040
## -----
## -----
## EDAD
##      n missing distinct      Info      Mean      Gmd      .05
.10
##    2941      24      62    0.999    40.01    14.07      23
26

```

```

##      .25      .50      .75      .90      .95
##      31       38       47       57       61
##
## lowest :   0  18  19  20  21, highest:  77  78  80 115 133
## -----
-----
## INGRESOS
##      n  missing  distinct      Info      Mean      Gmd      .05
.10
##      2941       24       500      1 14491037 19675720  535600
1200000
##      .25      .50      .75      .90      .95
##  2500000  5800000 12740000 25496000 44000000
##
## lowest :          0           1           7000           62000           100000
## highest: 262476029 281000000 503916000 867000000 1940070000
##
## 0 (2048, 0.696), 2e+07 (652, 0.222), 4e+07 (128, 0.044), 6e+07 (64,
0.022),
## 8e+07 (10, 0.003), 1e+08 (7, 0.002), 1.2e+08 (4, 0.001), 1.4e+08 (2,
0.001),
## 1.6e+08 (1, 0.000), 1.8e+08 (2, 0.001), 2e+08 (6, 0.002), 2.2e+08 (1,
0.000),
## 2.6e+08 (1, 0.000), 2.8e+08 (4, 0.001), 5e+08 (6, 0.002), 8.6e+08 (4,
0.001),
## 1.94e+09 (1, 0.000)
##
## For the frequency table, variable is rounded to the nearest 20000000
## -----
-----
## EGRESOS
##      n  missing  distinct      Info      Mean      Gmd      .05
.10
##      2941       24       193     0.997  8506309 14251053  0.0e+00
0.0e+00
##      .25      .50      .75      .90      .95
##  5.0e+05 1.8e+06 4.5e+06 1.0e+07 1.9e+07
##
## lowest :          0           1           6000           50000           80000
## highest: 219600000 483083000 842000000 1400000000 1600000000
##
## Value      0.0e+00 2.0e+07 4.0e+07 6.0e+07 8.0e+07 1.0e+08 1.2e+08
1.6e+08
## Frequency    2678      190       23       14        5        2        4
3
## Proportion  0.911     0.065     0.008     0.005     0.002     0.001     0.001
0.001
##
## Value      1.8e+08 2.0e+08 2.2e+08 4.8e+08 8.4e+08 1.4e+09 1.6e+09
## Frequency    2         6         1         6         4         2         1

```

```

## Proportion 0.001 0.002 0.000 0.002 0.001 0.001 0.001 0.000
##
## For the frequency table, variable is rounded to the nearest 200000000
## -----
-----
## NROPAISES
##      n missing distinct      Info      Mean      Gmd
##    2965       0         8     0.829     1.766     1.001
##
## lowest : 1 2 3 4 5, highest: 4 5 6 7 9
##
## Value      1      2      3      4      5      6      7      9
## Frequency 1547   895   318   106   80    14    4    1
## Proportion 0.522 0.302 0.107 0.036 0.027 0.005 0.001 0.000
## -----
-----
## Dist_Sum_INTER
##      n missing distinct      Info      Mean      Gmd      .05
## .10
##    1418    1547    208    0.992    17355    16099    3387
## 4552
##    .25     .50     .75     .90     .95
##    6474    9105   21376   37021   49912
##
## lowest : 904.81 971.23 1043.91 1482.35 1618.55
## highest: 106664.24 113810.25 115833.16 139343.96 758837.94
##
## Value      0    10000   20000   30000   40000   50000   60000   70000
## 100000
## Frequency 315    522    276    137     52     83     23     1
## 1
## Proportion 0.222 0.368 0.195 0.097 0.037 0.059 0.016 0.001
## 0.001
##
## Value      110000 120000 140000 760000
## Frequency 3       3       1       1
## Proportion 0.002 0.002 0.001 0.001
##
## For the frequency table, variable is rounded to the nearest 10000
## -----
-----
## Dist_Mean_INTER
##      n missing distinct      Info      Mean      Gmd      .05
## .10
##    1418    1547    181    0.902    4144    1686    1266
## 1694
##    .25     .50     .75     .90     .95
##    3178    4552    4552    5231    7482
##
## lowest : 904.81 951.18 971.23 1009.14 1015.03

```

```

## highest: 11178.51 11742.08 12403.34 14317.70 16328.81
## -----
## Dist_Max_INTER
##      n   missing  distinct     Info      Mean      Gmd      .05
.10
##    1418     1547       61    0.805    4985    2358    1619
2371
##    .25     .50     .75     .90     .95
##    4552     4552     4552    8352    9794
##
## lowest :  904.81  971.23 1043.91 1162.89 1482.35
## highest: 15667.16 15703.93 16328.81 17020.18 17780.33
## -----
## NROCIUDADES
##      n   missing  distinct     Info      Mean      Gmd      .05
.10
##    2965       0       18    0.978    3.944    2.769      1
1
##    .25     .50     .75     .90     .95
##    2        3       5       7       9
##
## lowest :  1  2  3  4  5, highest: 14 15 17 19 20
## 
## Value      1     2     3     4     5     6     7     8     9     10
11
## Frequency  457   542   549   489   310   184   176   103   59    8
22
## Proportion 0.154 0.183 0.185 0.165 0.105 0.062 0.059 0.035 0.020 0.003
0.007
##
## Value      12    13    14    15    17    19    20
## Frequency  16    18     2     9     7     2    12
## Proportion 0.005 0.006 0.001 0.003 0.002 0.001 0.004
## -----
## Dist_Mean_NAL
##      n   missing  distinct     Info      Mean      Gmd      .05
.10
##    2508     457     774       1    196.6    197.8   10.92
19.59
##    .25     .50     .75     .90     .95
##    60.80   127.70   269.08   483.50   632.81
##
## lowest :  4.48    7.47    7.57    7.94    8.13
## highest: 733.11  769.76  777.18 1165.74 1217.57
## -----
## Dist_HOY

```

```

##          n    missing   distinct     Info      Mean      Gmd      .05
.10
##    2965        0        48    0.392    4380    989.7    1044
2241
##    .25        .50        .75        .90        .95
##    4552        4552        4552    4552    4552
##
## lowest :    0.00    904.81   971.23  1043.91  1482.35
## highest: 17020.18 17573.03 17592.42 20736.96 21991.20
## -----
-----
## Dist_sum_NAL
##          n    missing   distinct     Info      Mean      Gmd      .05
.10
##    2965        0        845    0.996    1765    2227      1.0
1.0
##    .25        .50        .75        .90        .95
##    139.9     836.1    2533.4    4775.6    6597.9
##
## lowest :    1.00      7.94      8.97      9.34     10.34
## highest: 13839.34 14434.25 15367.51 16840.17 18832.06
## -----
-----
```

Este es un análisis bastante completo que incluye el número de valores nulos por columna del dataframe. Si quisieramos centrarnos en el análisis de los valores nulos, existen otros indicadores más sencillos a la hora de visualizar, como por ejemplo:

```

apply(is.na(df_Fraude), 2, sum)  #suma de valores NA por columna

##          id      FRAUDE      VALOR      HORA_AUX
Dist_max_NAL
##        0        0        0        0
0
##      Canal1      FECHA      COD_PAIS      CANAL
DIASEM
##        0        0        0        0        0
0
##      DIAMES      FECHA_VIN      OFICINA_VIN      SEXO
SEGMENTO
##        0        24        24        24        0
0
##      EDAD      INGRESOS      EGRESOS      NROPAISES
Dist_Sum_INTER
##        24        24        24        0
1547
## Dist_Mean_INTER  Dist_Max_INTER      NROCIUDADES  Dist_Mean_NAL
Dist_HOY
##        1547        1547        0        457
0
```

```

##      Dist_sum_NAL
##                  0
apply(is.na(df_Fraude), 2, mean) # Porcentaje de NA por columna
##          id        FRAUDE       VALOR     HORA_AUX
Dist_max_NAL
## 0.000000000 0.000000000 0.000000000 0.000000000
0.000000000
##        Canal1        FECHA     COD_PAIS      CANAL
DIASEM
## 0.000000000 0.000000000 0.000000000 0.000000000
0.000000000
##        DIAMES        FECHA_VIN  OFICINA_VIN     SEXO
SEGMENTO
## 0.000000000 0.008094435 0.008094435 0.000000000
0.000000000
##        EDAD        INGRESOS     EGRESOS   NROPAISES
Dist_Sum_INTER
## 0.008094435 0.008094435 0.008094435 0.000000000
0.521753794
## Dist_Mean_INTER  Dist_Max_INTER  NROCIUDADES  Dist_Mean_NAL
Dist_HOY
## 0.521753794 0.521753794 0.000000000 0.154131535
0.000000000
##      Dist_sum_NAL
## 0.000000000

```

El objetivo final es realizar un modelo sobre la variable FRAUDE, por lo tanto, es importante centrar el análisis exploratorio sobre esa variable. En este sentido, continuando con el análisis exploratorio, el siguiente gráfico me permite representar el porcentaje de operaciones declaradas fraudulentas.

```

common_theme <- theme(plot.title = element_text(hjust = 0.5, face =
"bold"))

ggplot(data = df_Fraude, aes(x = factor(FRAUDE),
                               y = prop.table(stat(count)), fill =
factor(FRAUDE),
                               label =
scales::percent(prop.table(stat(count)))) + 
  geom_bar(position = "dodge") +
  geom_text(stat = 'count',
            position = position_dodge(.9),
            vjust = -0.5,
            size = 3) +
  scale_x_discrete(labels = c("operaciones no fraudulentas",
"operaciones fraudulentas"))+
  scale_y_continuous(labels = scales::percent)+ 
  labs(x = 'Clase', y = 'Porcentage') +

```

```
ggtitle("casos de fraude sobre el total") +  
common_theme
```



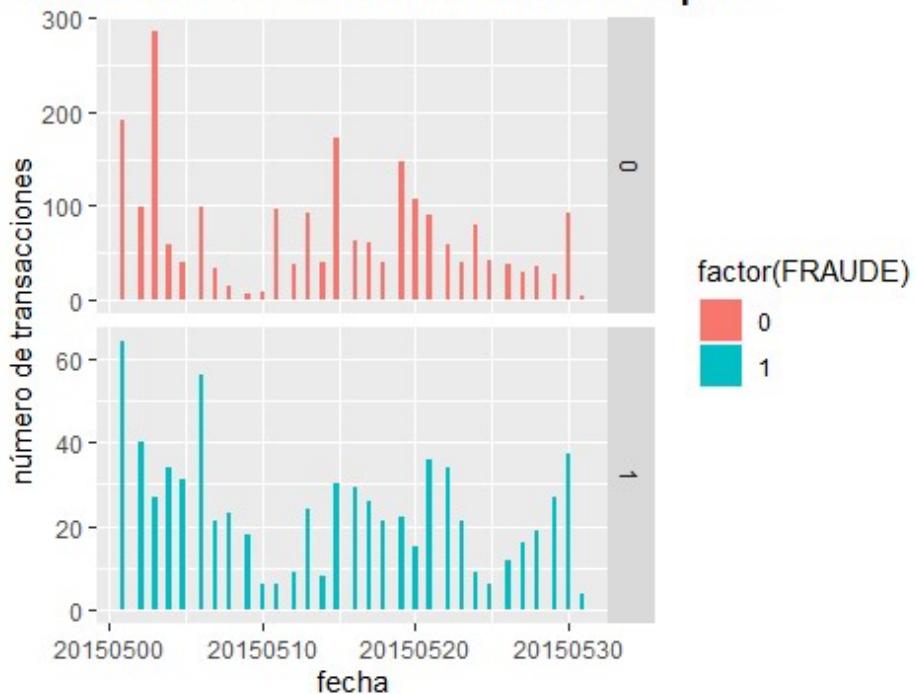
El 75% de las transacciones son no fraudulentas

El siguiente paso sería estudiar la relación de la variable FRAUDE con algunas de las variables que pienso que pueden estar relacionadas, de forma individual.

Represento la posible relación con la variable FECHA:

```
df_Fraude %>%  
  ggplot(aes(x = FECHA, fill = factor(FRAUDE))) + geom_histogram(bins =  
  100)+  
  labs(x = 'fecha', y = 'número de transacciones') +  
  ggtitle('Distribución del fraude como una serie temporal') +  
  facet_grid(FRAUDE ~ ., scales = 'free_y') + common_theme
```

## tribución del fraude como una serie temporal

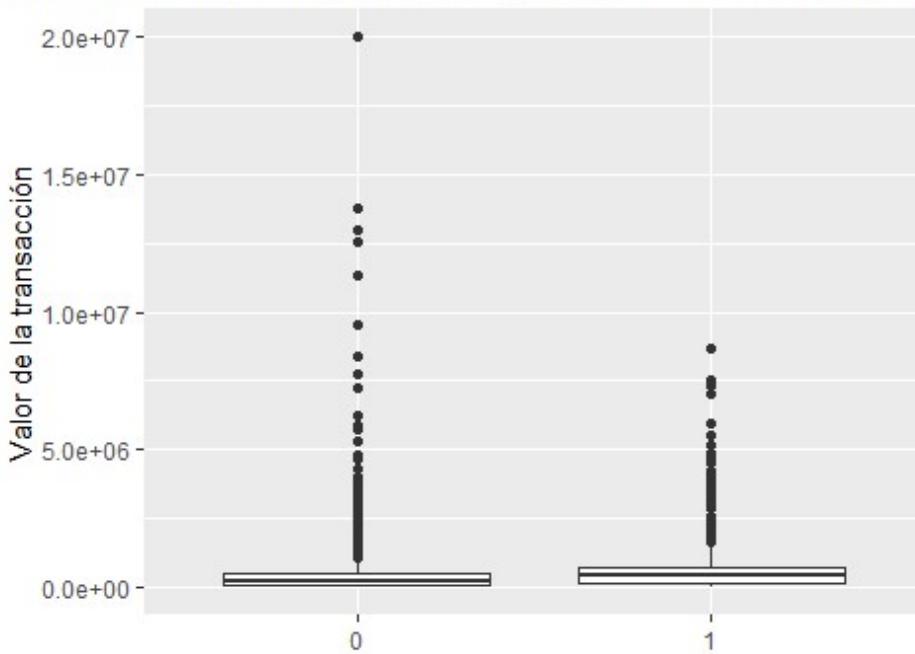


Con esta muestra de datos, parece que el fraude se da de forma bastante uniforme en el tiempo. A lo mejor con una muestra más extendida en el tiempo se podrían sacar conclusiones sobre si existe una tendencia o un comportamiento cíclico.

A continuación se estudia la relación con la variable VALOR (Valor de la transacción)

```
ggplot(df_Fraude, aes(x = factor(FRAUDE), y = VALOR)) + geom_boxplot() +  
  labs(x = ' ', y = 'Valor de la transacción') +  
  ggtitle("Distribucion de las transacciones según si son declaradas  
  fraudulentas o no") + common_theme
```

## Valor de la transacción según si son declaradas fraudulentas

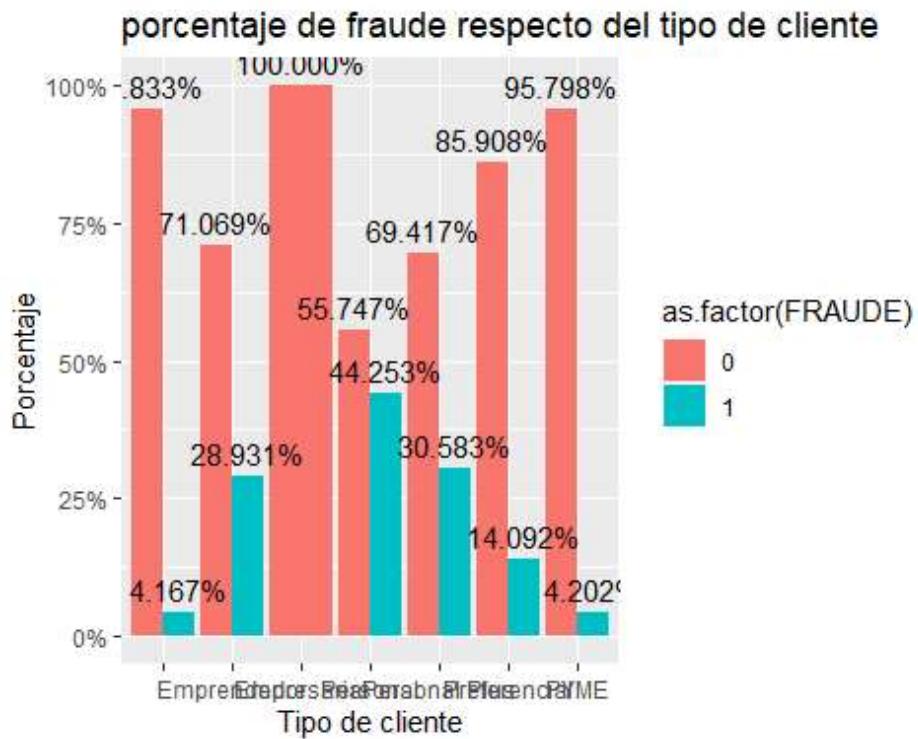


```
mean(df_Fraude[df_Fraude$FRAUDE==1,"VALOR"]) #La media de las transacciones fraudulentas es 666583.7  
## [1] 666583.7  
mean(df_Fraude[df_Fraude$FRAUDE==0,"VALOR"]) #la media de las transacciones legales es 450228.7  
## [1] 450228.7
```

Existe una mayor variabilidad dentro de las transacciones no fraudulentas. Además, la media de las transacciones declaradas fraudulentas es superior a la de las transacciones legales.

A continuación represento la variable SEGMENTO (Segmento del cliente) respecto de la variable FRAUDE utilizando dos gráficos de barras:

```
ggplot(df_Fraude, aes(x = factor(SEGMENTO), fill=as.factor(FRAUDE)))+  
  geom_bar(aes( y=..count../tapply(..count.., ..x.. ,sum)[..x..]),  
  position="dodge" ) +  
  geom_text(aes( y=..count../tapply(..count.., ..x.. ,sum)[..x..],  
  label=scales::percent(..count../tapply(..count.., ..x.. ,sum)[..x..]) ),  
  stat="count", position=position_dodge(0.9), vjust=-0.5)+  
  labs(x = 'Tipo de cliente', y = 'Porcentaje') +  
  scale_y_continuous(labels = scales::percent)+  
  ggtitle("porcentaje de fraude respecto del tipo de cliente")
```

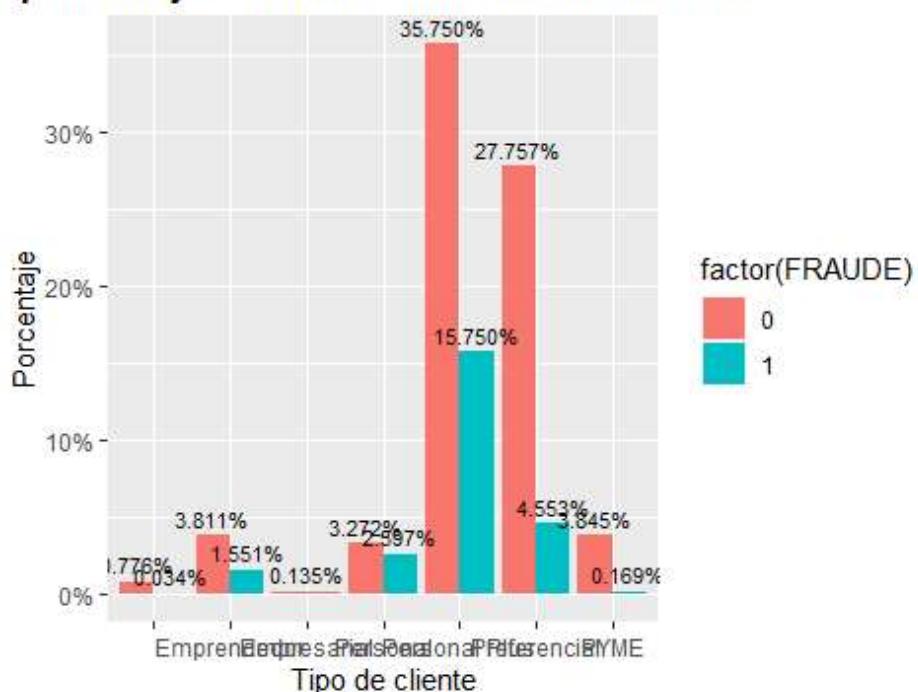


```

common_theme <- theme(plot.title = element_text(hjust = 0.5, face =
"bold"))

ggplot(data = df_Fraude, aes(x = factor(SEGMENTO),
                               y = prop.table(stat(count)), fill =
factor(FRAUDE),
                               label =
scales::percent(prop.table(stat(count)))) + 
  geom_bar(position = "dodge") +
  geom_text(stat = 'count',
            position = position_dodge(.9),
            vjust = -0.5,
            size = 3) +
  scale_y_continuous(labels = scales::percent)+ 
  labs(x = 'Tipo de cliente', y = 'Porcentaje') + 
  ggtitle("porcentaje de transacciones frente al total") + 
  common_theme
  
```

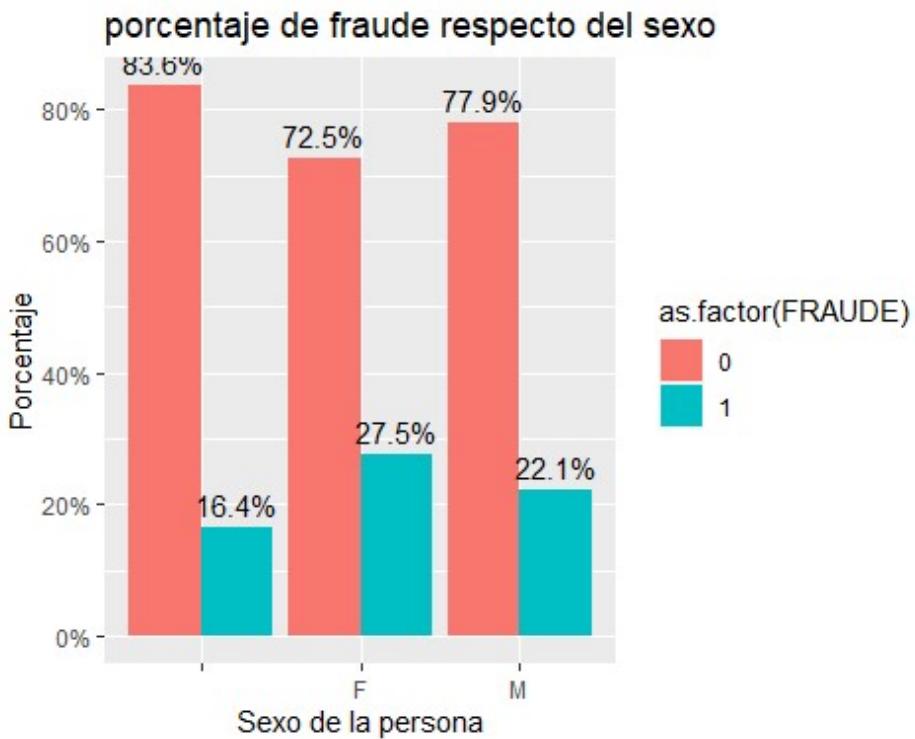
## porcentaje de transacciones frente al total



El primer gráfico muestra cómo las transacciones realizadas por el segmento personal tienen el mayor porcentaje de operaciones declaradas fraudulentas, con un 44.253%. Al contrario, el segmento Empresarial parece no tener intentos de fraude.

De la misma forma, la relación entre FRAUDE y SEXO se muestra en el siguiente gráfico.

```
ggplot(df_Fraude, aes(x = factor(SEXO), fill=as.factor(FRAUDE)))+
  geom_bar(aes( y=..count../tapply(..count.., ..x.. ,sum)[..x..]),
  position="dodge" ) +
  geom_text(aes( y=..count../tapply(..count.., ..x.. ,sum)[..x..],
  label=scales::percent(..count../tapply(..count.., ..x.. ,sum)[..x..]) ),
  stat="count", position=position_dodge(0.9), vjust=-0.5)+  
  labs(x = 'Sexo de la persona', y = 'Porcentaje') +
  scale_y_continuous(labels = scales::percent)+  
  ggtitle("porcentaje de fraude respecto del sexo")
```

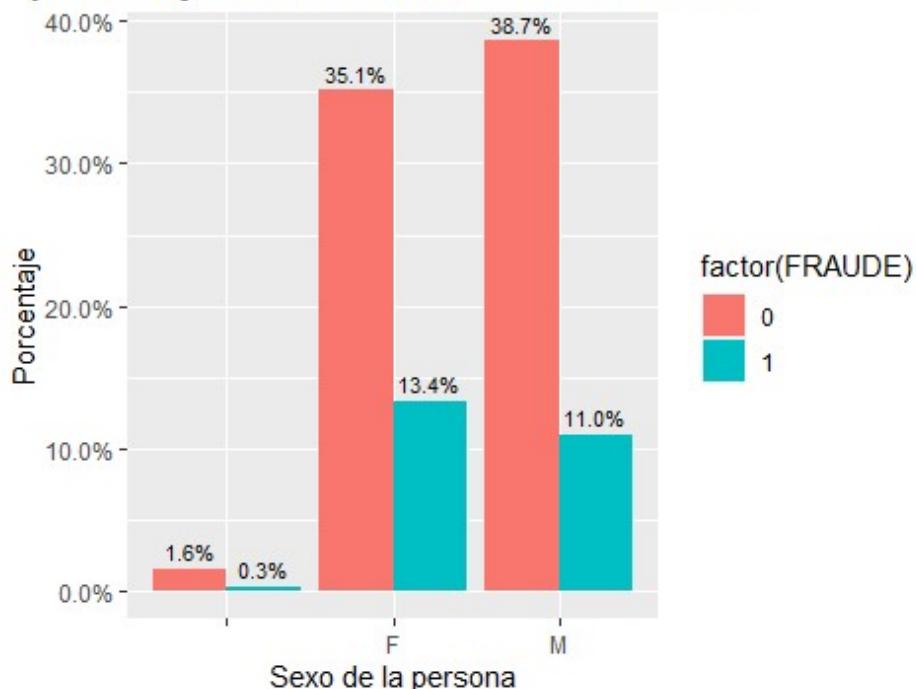


```

common_theme <- theme(plot.title = element_text(hjust = 0.5, face =
"bold"))

ggplot(data = df_Fraude, aes(x = factor(SEXO),
                               y = prop.table(stat(count)), fill =
factor(FRAUDE),
                               label =
scales::percent(prop.table(stat(count)))) + 
  geom_bar(position = "dodge") +
  geom_text(stat = 'count',
            position = position_dodge(.9),
            vjust = -0.5,
            size = 3) +
  scale_y_continuous(labels = scales::percent)+ 
  labs(x = 'Sexo de la persona', y = 'Porcentaje') +
  ggtitle("porcentaje de transacciones frente al total") +
  common_theme
  
```

## porcentaje de transacciones frente al total

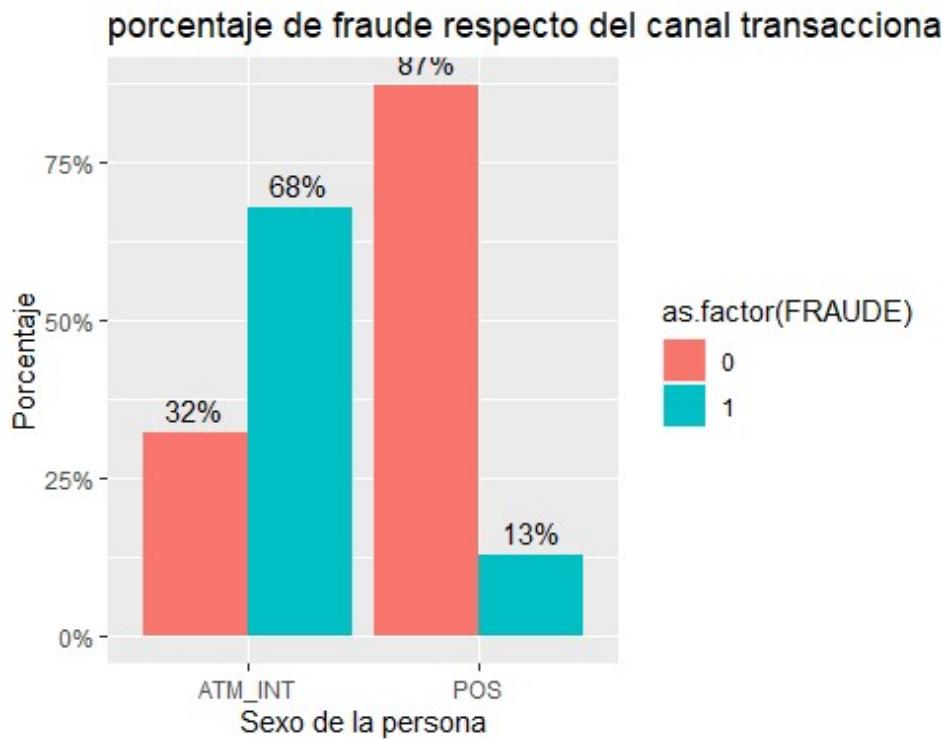


Se puede

observar como el porcentaje de mujeres que cometen fraude es superior al de los hombres (27.5% frente al 22.1%). Además, el número de mujeres que han realizado operaciones declaradas fraudulentas es superior al de los hombres (13.4% frente al 11% en los hombres).

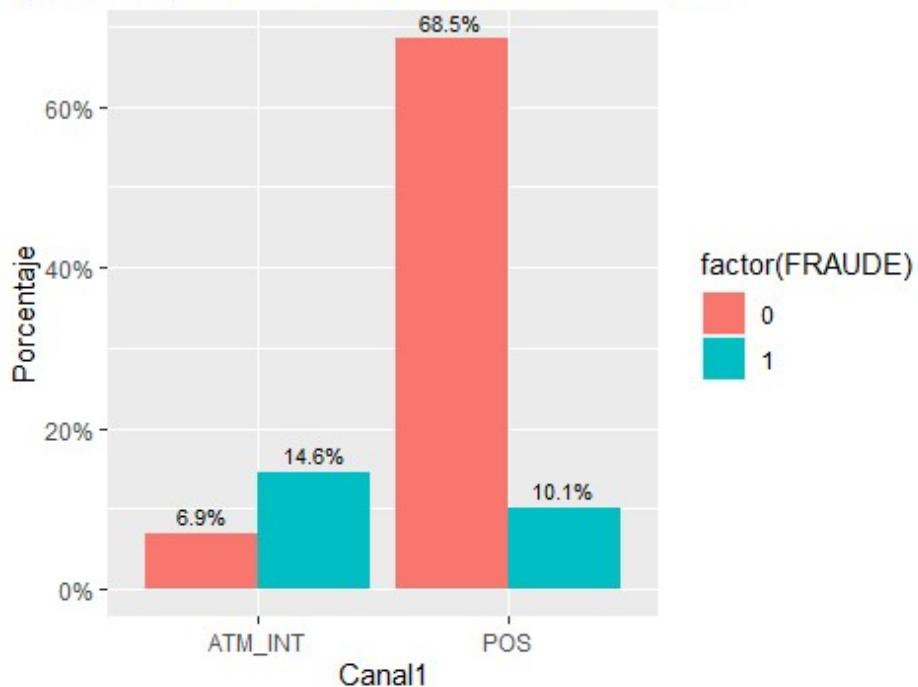
Si represento la relación entre la variable Canal1 y la variable objetivo:

```
ggplot(df_Fraude, aes(x = factor(Canal1), fill=as.factor(FRAUDE)))+
  geom_bar(aes( y=..count../tapply(..count.., ..x.. ,sum)[..x..]),  
position="dodge" ) +  
  geom_text(aes( y=..count../tapply(..count.., ..x.. ,sum)[..x..],  
label=scales::percent(..count../tapply(..count.., ..x.. ,sum)[..x..]) ),  
    stat="count", position=position_dodge(0.9), vjust=-0.5)+  
  labs(x = 'Sexo de la persona', y = 'Porcentaje') +  
  scale_y_continuous(labels = scales::percent)+  
  ggtitle("porcentaje de fraude respecto del canal transaccional")
```



```
ggplot(data = df_Fraude, aes(x = factor(Canal1),
                               y = prop.table(stat(count)), fill =
factor(FRAUDE),
                               label =
scales::percent(prop.table(stat(count)))) + 
  geom_bar(position = "dodge") +
  geom_text(stat = 'count',
            position = position_dodge(.9),
            vjust = -0.5,
            size = 3) +
  scale_y_continuous(labels = scales::percent)+ 
  labs(x = 'Canal1', y = 'Porcentaje') +
  ggtitle("porcentaje de transacciones frente al total") +
  common_theme
```

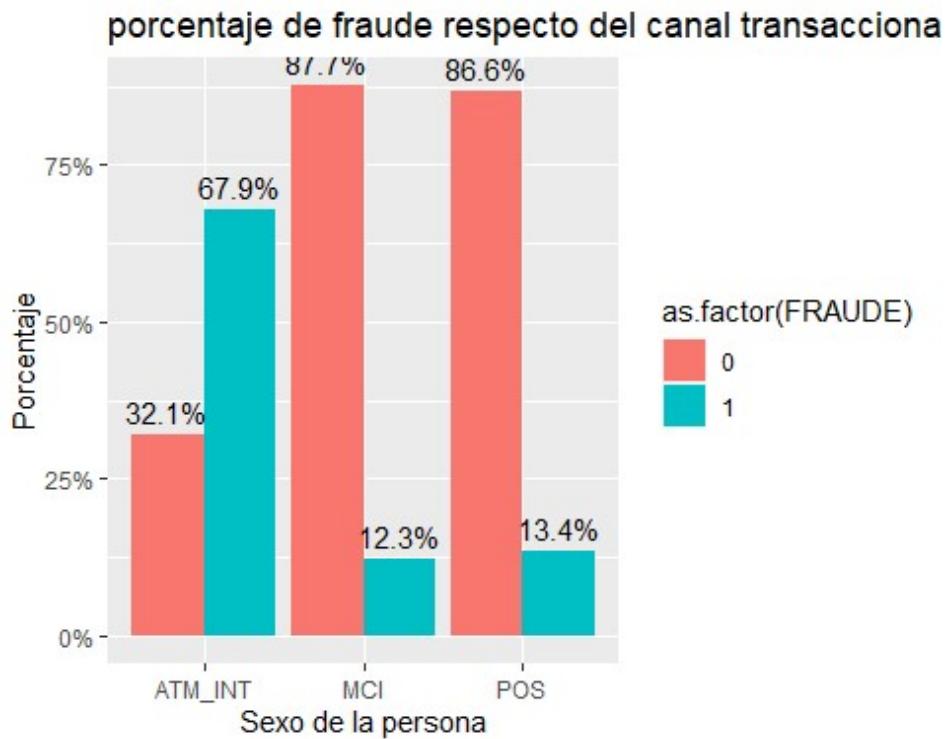
## porcentaje de transacciones frente al total



La Mayoría de las transacciones a través de ATM\_INT son fraudulentas, aunque la inmensa mayoría de transacciones se realice en POS, donde el fraude es menor.

La variable Canal se trata de la misma forma.

```
ggplot(df_Fraude, aes(x = factor(CANAL), fill=as.factor(FRAUDE)))+
  geom_bar(aes( y=..count../tapply(..count.., ..x.. ,sum)[..x..]),
  position="dodge" ) +
  geom_text(aes( y=..count../tapply(..count.., ..x.. ,sum)[..x..],
  label=scales::percent(..count../tapply(..count.., ..x.. ,sum)[..x..]) ),
  stat="count", position=position_dodge(0.9), vjust=-0.5)+  
  labs(x = 'Sexo de la persona', y = 'Porcentaje') +
  scale_y_continuous(labels = scales::percent)+  
  ggtitle("porcentaje de fraude respecto del canal transaccional")
```



```
ggplot(data = df_Fraude, aes(x = factor(CANAL),
                               y = prop.table(stat(count)), fill =
factor(FRAUDE),
                               label =
scales::percent(prop.table(stat(count)))) +  

  geom_bar(position = "dodge") +  

  geom_text(stat = 'count',
            position = position_dodge(.9),
            vjust = -0.5,
            size = 3) +  

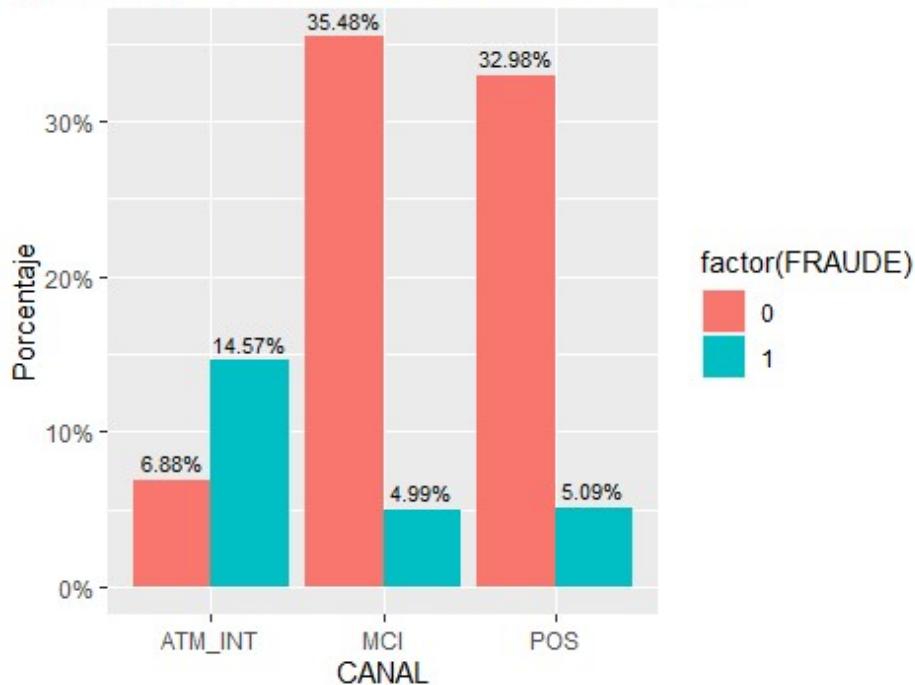
  scale_y_continuous(labels = scales::percent)+  

  labs(x = 'CANAL', y = 'Porcentaje') +  

  ggtitle("porcentaje de transacciones frente al total") +  

  common_theme
```

## porcentaje de transacciones frente al total



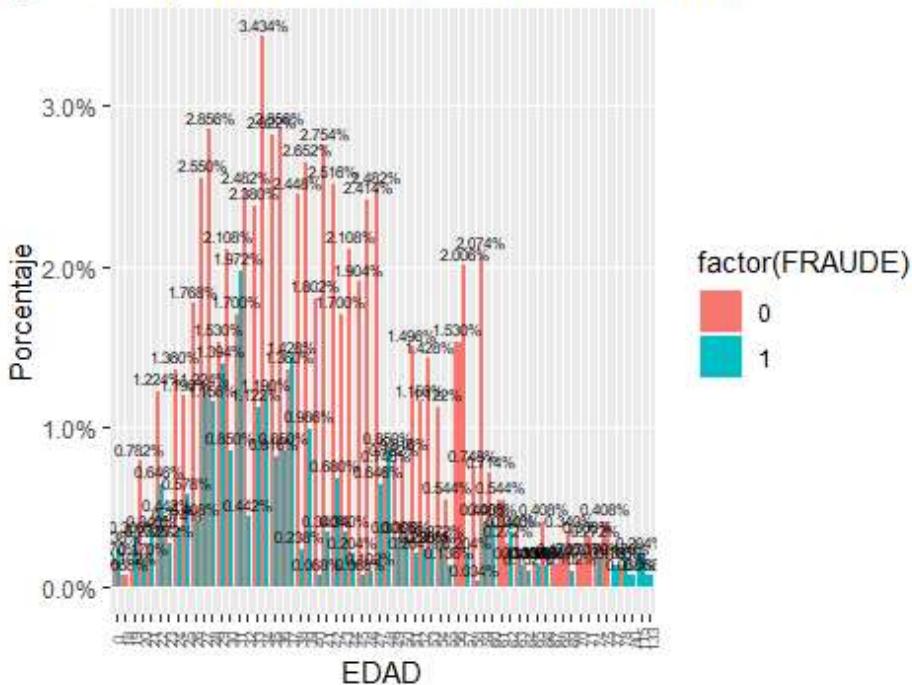
En este caso,

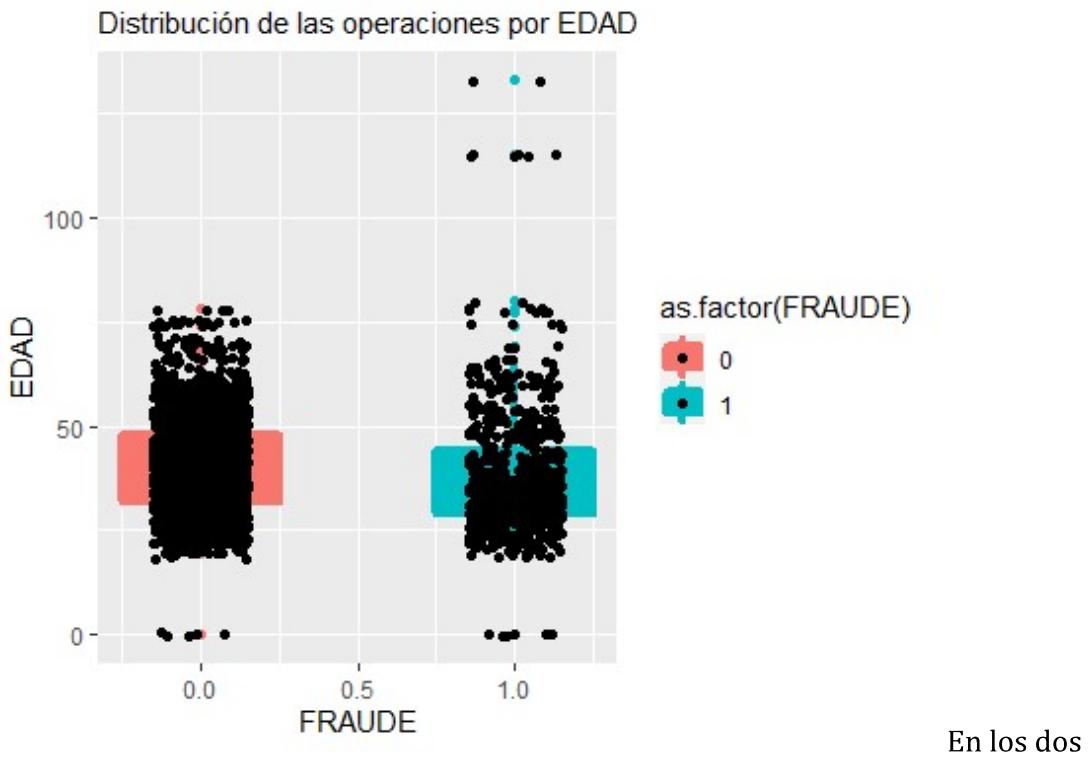
El canal donde se comete más fraude sigue siendo el ATM\_INT, pero el más utilizado es MCI.

Para estudiar la distribución de la variable EDAD voy a utilizar de nuevo dos gráficos.

```
df_edad<-df_Fraude %>% drop_na(EDAD) #Elimino Los valores NA en La  
variable EDAD  
  
ggplot(data = df_edad, aes(x = factor(EDAD),  
                           y = prop.table(stat(count)), fill =  
                           factor(FRAUDE),  
                           label =  
                           scales::percent(prop.table(stat(count)))) +  
  geom_bar(position = "dodge") +  
  geom_text(stat = 'count',  
           position = position_dodge(.5),  
           vjust = -0.5,  
           size = 2) +  
  
  scale_y_continuous(labels = scales::percent)+  
  labs(x = 'EDAD', y = 'Porcentaje') +  
  ggtitle("porcentaje de transacciones frente al total") +  
  common_theme +  
  theme(axis.text.x = element_text(angle = 90, vjust=0.5, size = 6),  
        panel.grid.minor = element_blank())
```

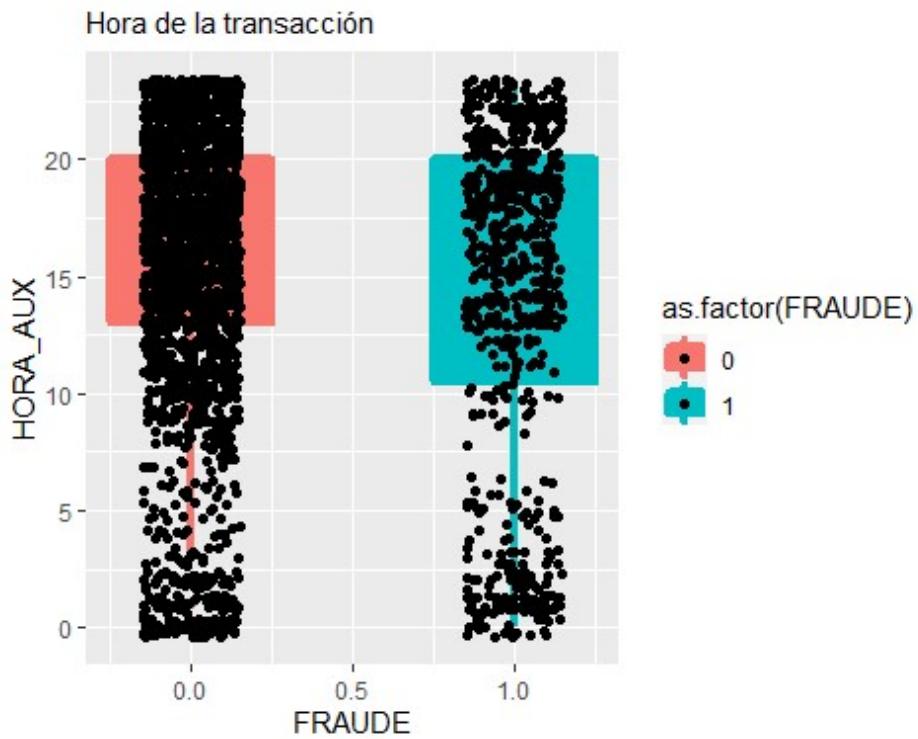
## porcentaje de transacciones frente al total





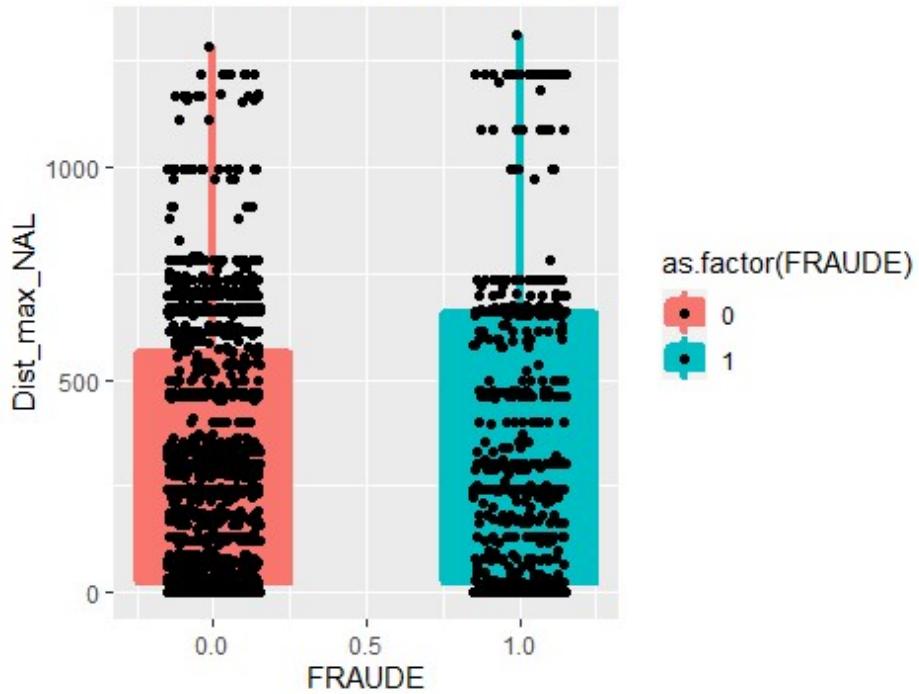
Para el resto de variables he optado por utilizar graficos geom\_boxplot como en el ejemplo anterior.

```
df_Fraude %>%
  ggplot(aes(x=FRAUDE, y=HORA_AUX, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5, lwd=1.5, aes(color=as.factor(FRAUDE))) +
  geom_jitter(width=0.15) +
  labs(subtitle="Hora de la transacción")
```

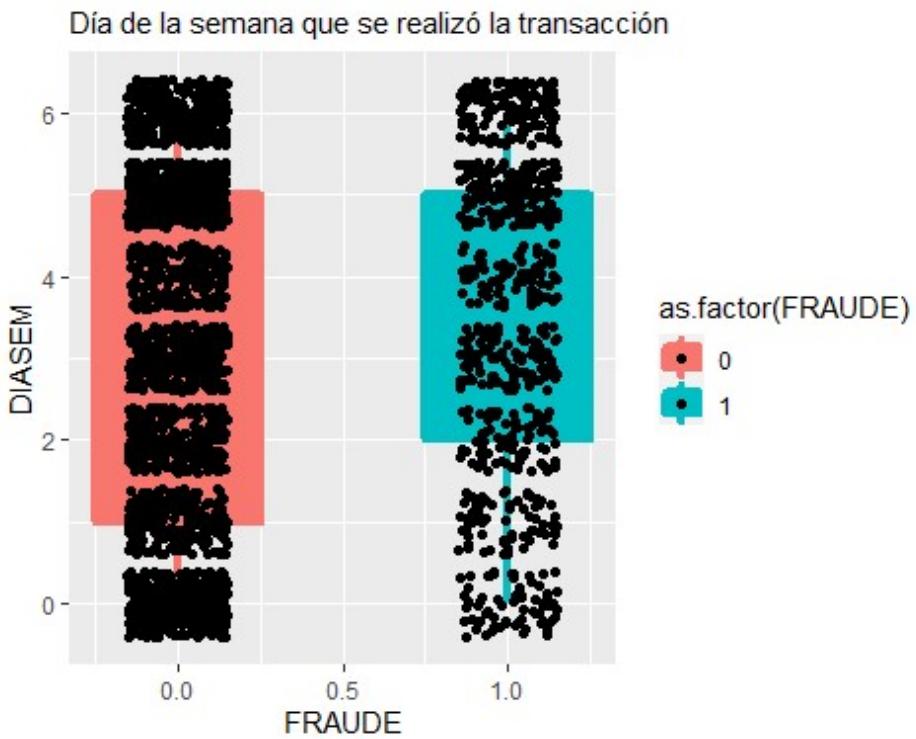


```
df_Fraude %>%
  ggplot(aes(x=FRAUDE, y=Dist_max_NAL, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5, lwd=1.5, aes(color=as.factor(FRAUDE))) +
  geom_jitter(width=0.15) +
  labs(subtitle="Dist maxima recorrida a nivel nacional")
```

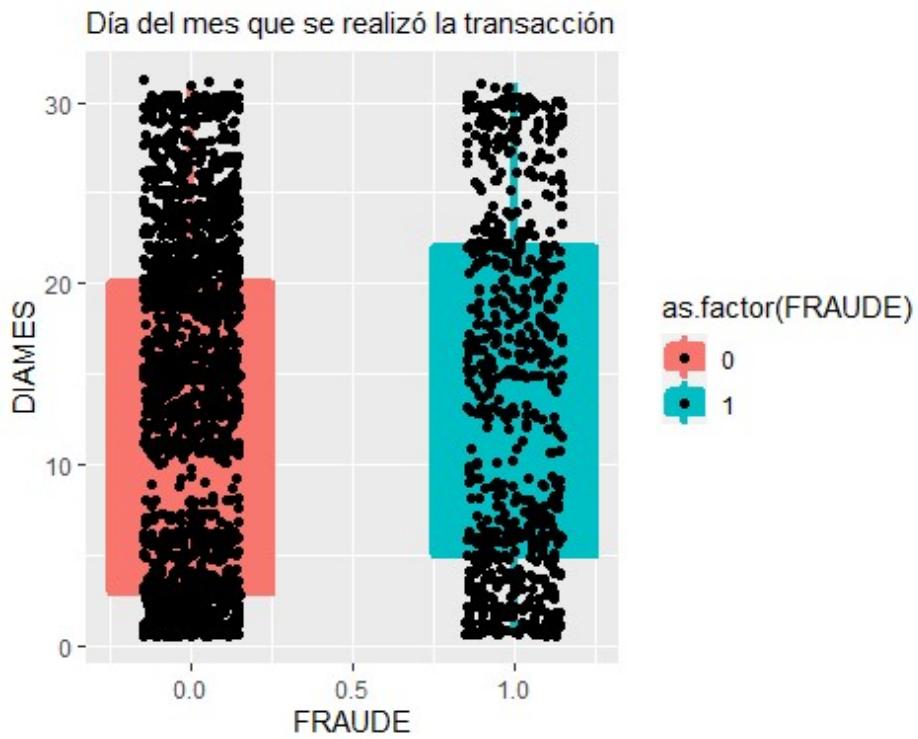
Dist maxima recorrida a nivel nacional



```
df_Fraude %>%
  ggplot(aes(x=FRAUDE, y=DIASEM, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  geom_jitter(width=0.15)+
  labs(subtitle="Día de la semana que se realizó la transacción")
```

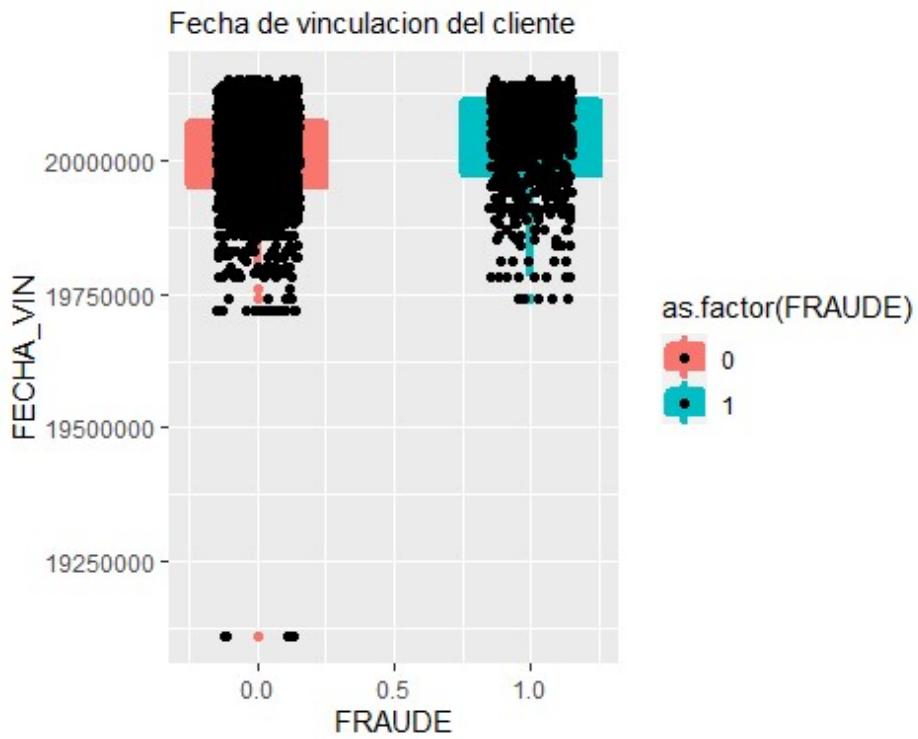


```
df_Fraude %>%
  ggplot(aes(x=FRAUDE, y=DIAMES, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5, lwd=1.5, aes(color=as.factor(FRAUDE))) +
  geom_jitter(width=0.15) +
  labs(subtitle="Día del mes que se realizó la transacción")
```



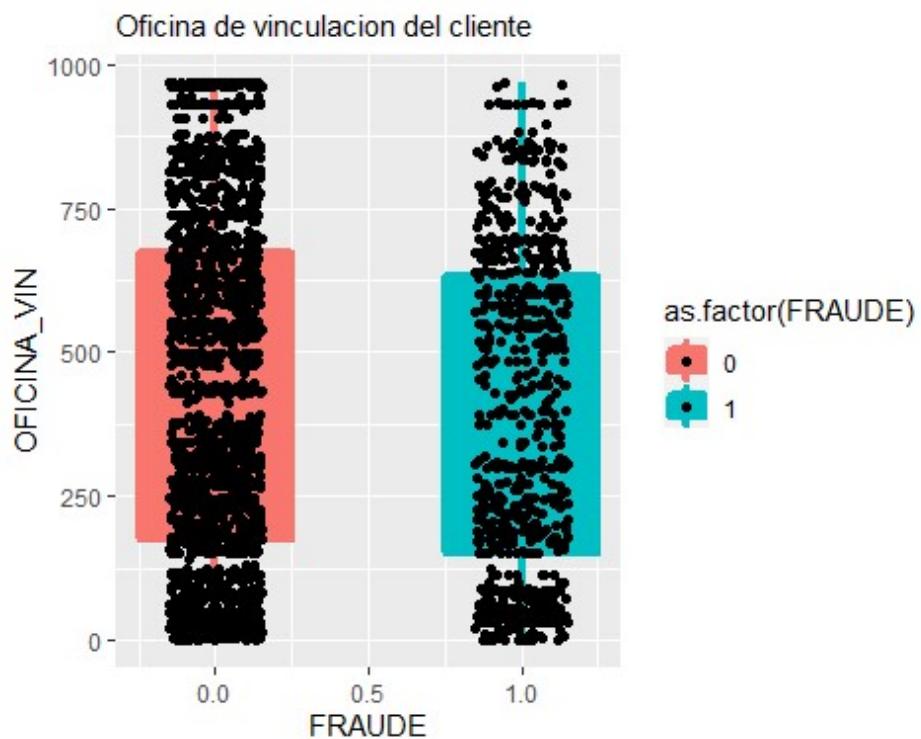
```
df_Fraude %>%
  ggplot(aes(x=FRAUDE, y=FECHA_VIN, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5, lwd=1.5, aes(color=as.factor(FRAUDE))) +
  geom_jitter(width=0.15) +
  labs(subtitle="Fecha de vinculacion del cliente")

## Warning: Removed 24 rows containing non-finite values (stat_boxplot).
## Warning: Removed 24 rows containing missing values (geom_point).
```

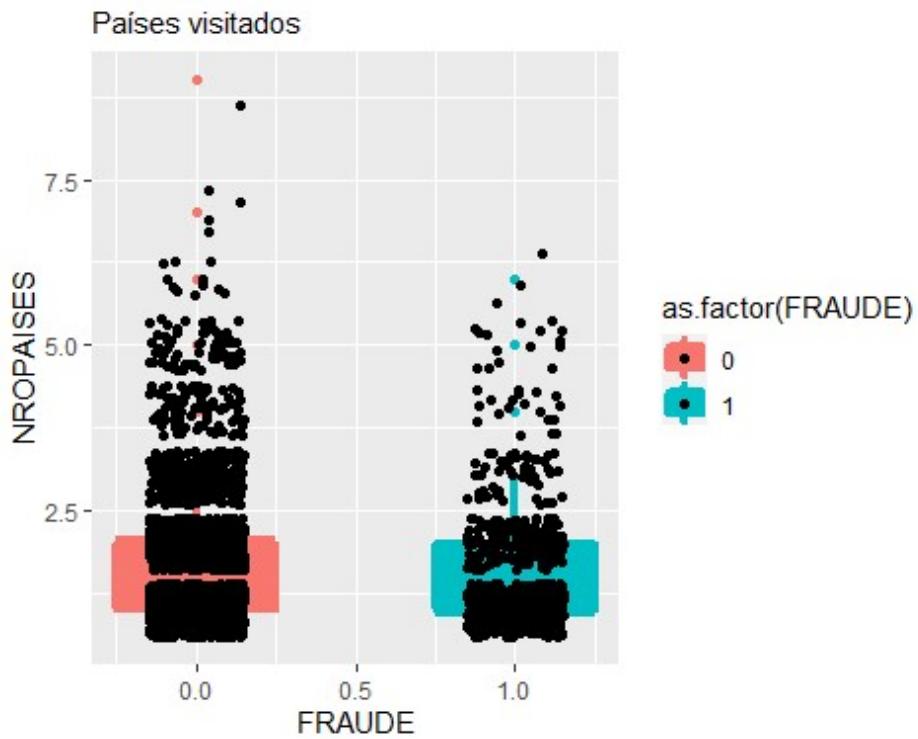


```
df_Fraude %>%
  ggplot(aes(x=FRAUDE, y=OFICINA_VIN, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  geom_jitter(width=0.15)+
  labs(subtitle="Oficina de vinculacion del cliente")

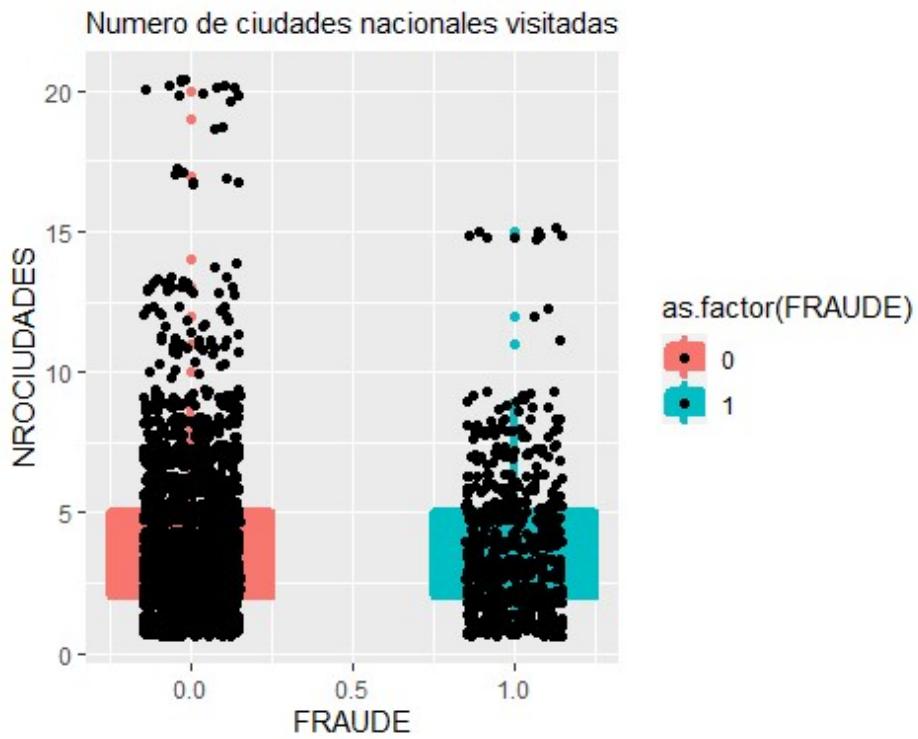
## Warning: Removed 24 rows containing non-finite values (stat_boxplot).
## Removed 24 rows containing missing values (geom_point).
```



```
df_Fraude %>%
  ggplot(aes(x=FRAUDE, y=NROPAISES, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  geom_jitter(width=0.15)+
  labs(subtitle="Países visitados")
```



```
df_Fraude %>%
  ggplot(aes(x=FRAUDE, y=NROCIUDADES, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5, lwd=1.5, aes(color=as.factor(FRAUDE))) +
  geom_jitter(width=0.15) +
  labs(subtitle="Número de ciudades nacionales visitadas")
```



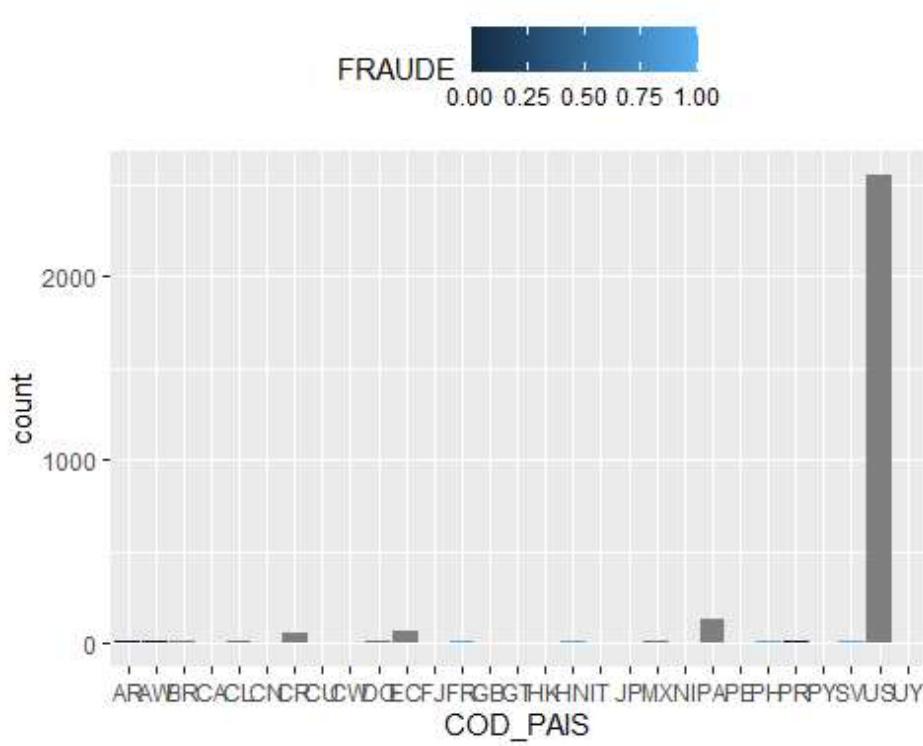
Estos gráficos son especialmente útiles como referencia a la hora de identificar valores atípicos.

Siguiendo con el análisis exploratorio de las variables, los siguientes dos gráficos de barras relacionan el fraude con el país en el que ocurre la transacción.

```
df_pais <- copy(df_Fraude)
df_pais <- as.data.frame((df_pais))

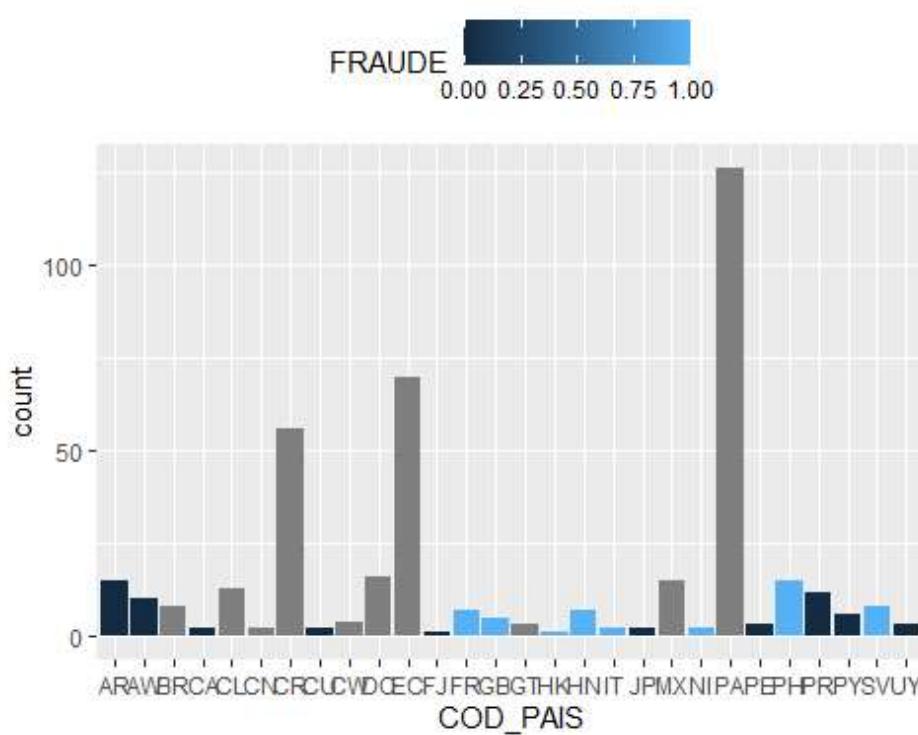
df_pais <- df_pais %>% filter(!str_detect( COD_PAIS, "NA")) # No tengo en cuenta los valores nulos para que no afecten al gráfico.

ggplot(df_pais, aes(x = COD_PAIS)) +
  geom_bar(aes(fill = FRAUDE), position = position_stack(reverse = TRUE))
  +
  theme(legend.position = "top")
```



`df_pais <- df_pais %>% filter(!str_detect( COD_PAIS, "US")) #La mayoría de las operaciones vienen de EEUU por lo que necesito eliminar "US" del gráfico para que se pueda apreciar mejor.`

```
ggplot(df_pais, aes(x = COD_PAIS)) +
  geom_bar(aes(fill = FRAUDE), position = position_stack(reverse = TRUE))
+ theme(legend.position = "top")
```



Estos dos gáficos nos permiten observar como la distribución de las operaciones entre los distintos países es muy desigual. En este sentido, la mayoría de las transacciones ocurren en Estados Unidos.

Debido a la desigual distribución de las operaciones, la mayoría de países carecen de la suficiente información para sacar conclusiones. Aún así, si quisieramos saber que países tienen un ratio de fraude más alto según nuestro dataset, podemos recurrir a la siguiente función de aggregación:

```
res<-aggregate(FRAUDE~COD_PAIS, df_Fraude, mean)
res[order(-res$FRAUDE), ] #porcentaje de fraude por país

##      COD_PAIS      FRAUDE
## 13      FR 1.0000000
## 14      GB 1.0000000
## 16      HK 1.0000000
## 17      HN 1.0000000
## 18      IT 1.0000000
## 21      NI 1.0000000
## 24      PH 1.0000000
## 27      SV 1.0000000
## 7       CR 0.9642857
## 3       BR 0.8750000
## 20     MX 0.8666667
## 11     EC 0.7857143
## 9       CW 0.7500000
## 5       CL 0.6923077
```

```

## 15      GT 0.6666667
## 6       CN 0.5000000
## 10     DO 0.3750000
## 28     US 0.2020400
## 22     PA 0.1507937
## 1      AR 0.0000000
## 2      AW 0.0000000
## 4      CA 0.0000000
## 8      CU 0.0000000
## 12    FJ 0.0000000
## 19    JP 0.0000000
## 23    PE 0.0000000
## 25    PR 0.0000000
## 26    PY 0.0000000
## 29    UY 0.0000000

```

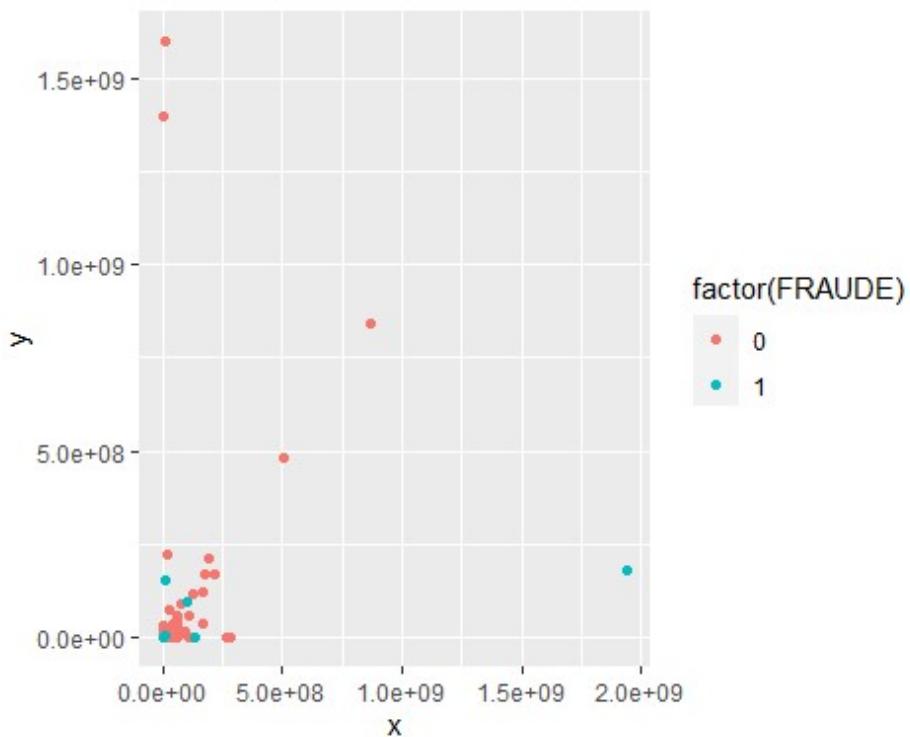
Sin embargo, como se ha dicho antes, la mayoría de observaciones provienen de unos pocos países por lo que los valores de la función de agregación anterior no son realmente representativos.

Por último, represento como se distribuyen las variables INGRESOS Y EGRESOS respecto de la variable factor FRAUDE:

```

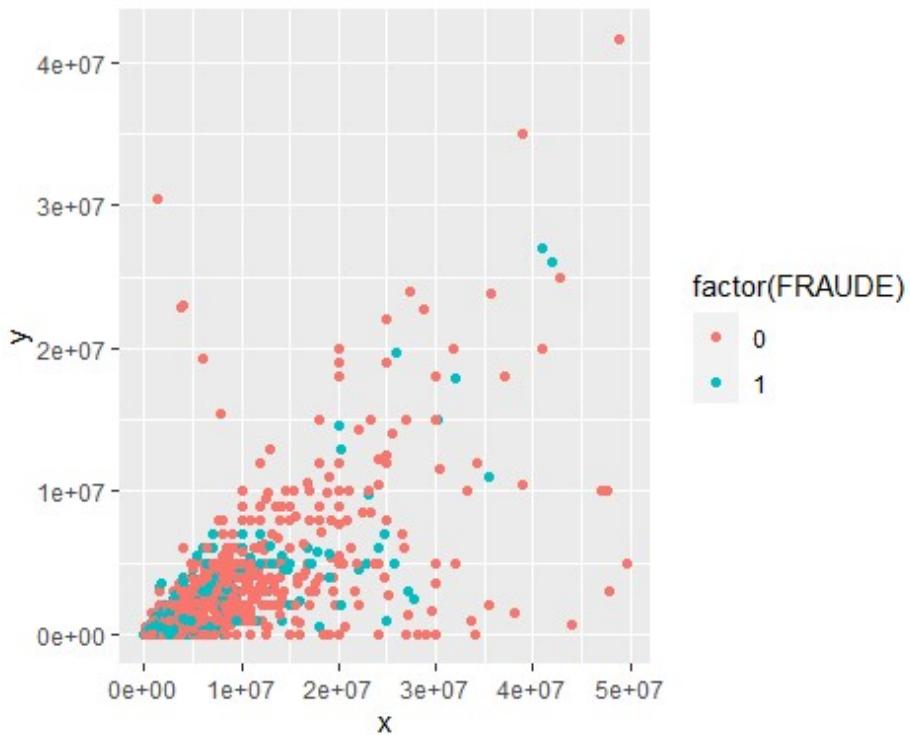
ggplot(df_Fraude, aes_(df_Fraude$INGRESOS,df_Fraude$EGRESOS)) +
  geom_point(aes(color =factor(FRAUDE)))
## Warning: Removed 24 rows containing missing values (geom_point).

```



#Debido a la dispersión de esta variable sería necesario acotar los datos antes de realizar el mismo gráfico.

```
p<-df_Fraude %>% filter(INGRESOS<50000000) %>% filter(EGRESOS<50000000) #  
Me fijo en el percentil 90 y la media de ambas variables en el describe()  
del principio para fijar los límites para el gráfico.  
ggplot(p, aes_(p$INGRESOS,p$EGRESOS)) + geom_point(aes(color  
=factor(FRAUDE)))
```



Se ve como las

variables INGRESOS y EGRESOS se distribuyen de la misma manera siendo casos de fraude o no. Por lo tanto, se puede deducir que la variable FRAUDE no estará muy correlacionada con ninguna de las dos variables.

#### -Análisis de dependencia/independencia de variables

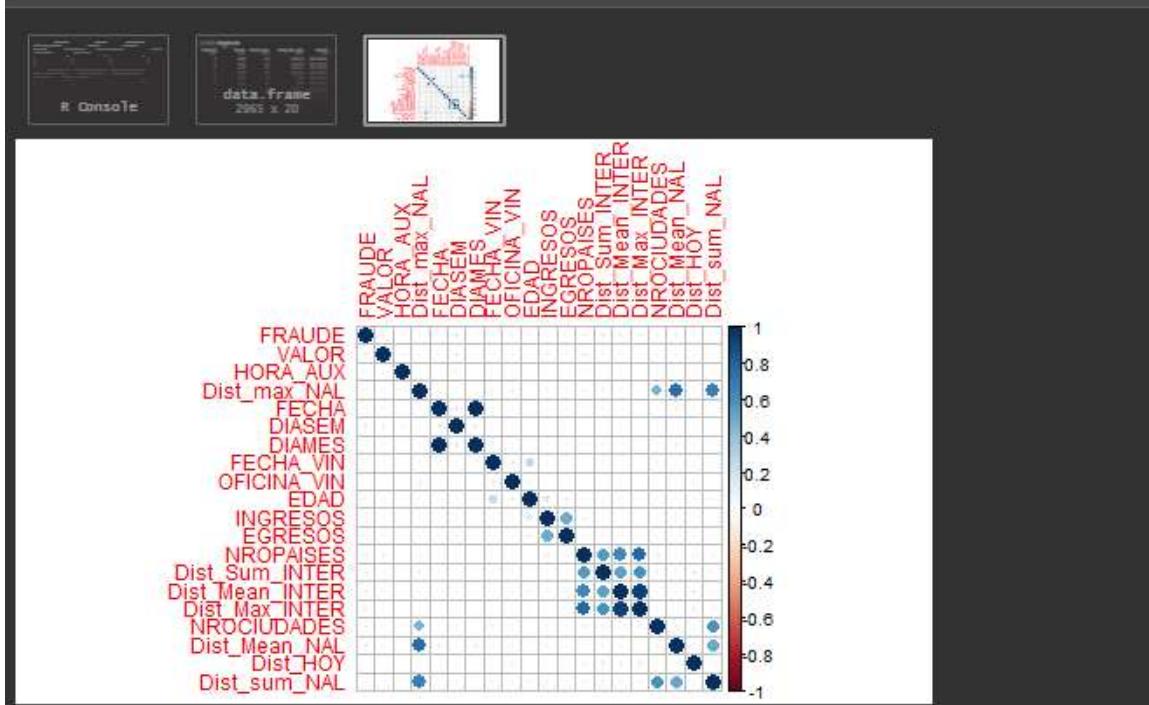
El primer paso es generar una matriz de correlaciones:

```
# La matriz de correlación sólo admite variables numéricas  
df_corr <- select(df_Fraude, -Canal1, -COD_PAIS, -CANAL, -id, -SEGMENTO,  
-SEXO ) #SElecciono únicamente variables numéricas.  
  
df_corr[is.na(df_corr)]<- 0 # Supongo que los valores NA de variables  
#numéricas son igual a 0  
  
apply(is.na(df_corr), 2, sum) #Comprobar que ya no existen nulos  
  
transform(df_corr,FRAUDE= as.numeric(FRAUDE))  
  
cor.table = cor(df_corr)
```

```
corrplot(cor.table, method = "circle")
```

El primer paso es generar una matriz de correlaciones:

```
# La matriz de correlación sólo admite variables numéricas
df_corr <- select(df_Fraude, -Canal, -COD_PAIS, -CANAL, -id, -SEGMENTO, -SEXO) #Se eligen únicamente variables numéricas
df_corr[is.na(df_corr)] <- 0 #Supongo que los valores NA de variables numéricas son igual a 0
apply(is.na(df_corr), 2, sum) #Comprobar que ya no existen nulos
transform(df_corr, FRAUDE= as.numeric(FRAUDE))
cor.table = cor(df_corr)
corrplot(cor.table, method = "circle")
```



La matriz de correlaciones no muestra ninguna relación de dependencia relevante entre las variables, excepto para DIASEM Y FECHA (DIASEM depende directamente de la variable FECHA) y en el caso de las variables Dist, NROCIUDADES y NROPAISES, en menor medida.

Para que no exista multicolinealidad en el modelo, seleccionaré únicamente la variable Dist\_Sum\_INTER(Sumatoria de distancia recorrida a nivel internacional), que es la que dentro de las variables de distancia tiene una mayor correlación con la variable objetivo (FRAUDE). De la misma manera, excluiré la variable DIAMES para solo quedarme con la variable FECHA.

Que no exista dentro de las variables seleccionadas ninguna con un nivel de correlación respecto a la variable objetivo relevante puede no ser buena señal a priori.

Si se quiere consultar el resultado numérico de las correlaciones se puede recurrir a la siguiente gráfica.

```
round(cor(df_corr), 2)
```

```
## Error in is.data.frame(x): objeto 'df_corr' no encontrado
```

Para no tener que transformar los valores nulos asumiendo su valor, voy a utilizar la librería missForest para imputación de valores. Además, esta vez si añadiré la variable SEXO.

```
#transformo la variable SEXO en numérica
df_Fraude$SEXO <- replace(df_Fraude$SEXO, df_Fraude$SEXO == "", NA)
df_Fraude$SEXO <- factor(df_Fraude$SEXO, labels=c("F", "M"))
df_Fraude$SEXO <- as.numeric(df_Fraude$SEXO, labels=c("F", "M"))
str(df_Fraude$SEXO)

##  num [1:2965] 2 2 2 2 2 2 2 1 2 2 ...

df_corr <- select(df_Fraude, -Canal1, -COD_PAIS, -CANAL, -id, -SEGMENTO,
-Dist_sum_NAL, -Dist_HOY, -Dist_Mean_NAL, -Dist_Max_INTER, -
Dist_Mean_INTER, -Dist_max_NAL, -DIAMES) #elimino variables no numéricas
o con posible multicolinealidad.

## Error in select(df_Fraude, -Canal1, -COD_PAIS, -CANAL, -id, -SEGMENTO,
: unused arguments (-Canal1, -COD_PAIS, -CANAL, -id, -SEGMENTO, -
Dist_sum_NAL, -Dist_HOY, -Dist_Mean_NAL, -Dist_Max_INTER, -
Dist_Mean_INTER, -Dist_max_NAL, -DIAMES)

#imputamos valores

help(missForest)

## starting httpd help server ... done

df_impo <- missForest(df_corr)

## Error in nrow(xmis): objeto 'df_corr' no encontrado

df_impo$OOBerror # errores asociados a cada variable (MSE para continuas
(error cuadrático medio) y PFC(proporción de mala clasificación)
categoricas)

## Error in eval(expr, envir, enclos): objeto 'df_impo' no encontrado

#calculo de varianzas quitando na
apply(df_corr,2,var,na.rm=TRUE)

## Error in apply(df_corr, 2, var, na.rm = TRUE): objeto 'df_corr' no
encontrado

apply(is.na(df_impo$ximp),2,sum) #me indica nº de na en la BBDD imputada,
comprobamos que lo hemos hecho bien.

## Error in apply(is.na(df_impo$ximp), 2, sum): objeto 'df_impo' no
encontrado

df_corr <- df_impo$ximp

## Error in eval(expr, envir, enclos): objeto 'df_impo' no encontrado
```

```

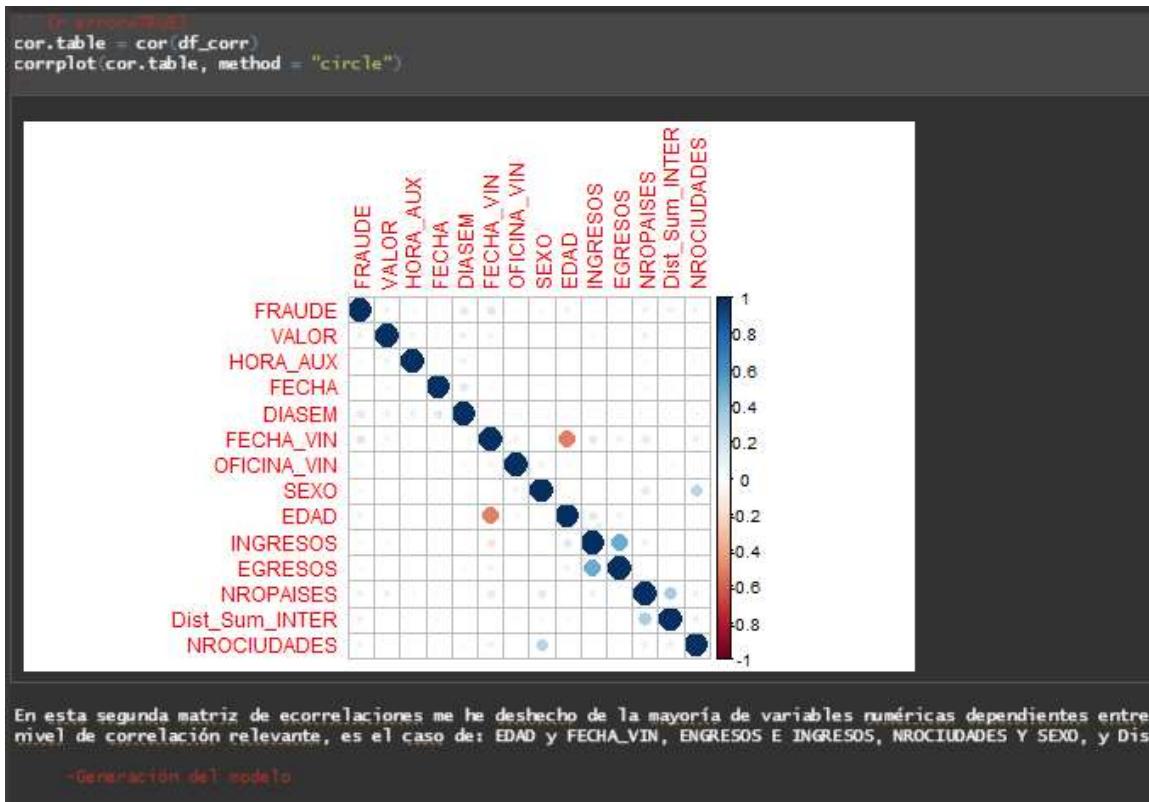
apply(is.na(df_corr), 2, sum) #Me aseguro de que todos los valores NA han
sido remplazados

## Error in apply(is.na(df_corr), 2, sum): objeto 'df_corr' no encontrado
cor.table = cor(df_corr)

## Error in is.data.frame(x): objeto 'df_corr' no encontrado
corrplot(cor.table, method = "circle")

## Error in corrplot(cor.table, method = "circle"): objeto 'cor.table' no
encontrado

```



En esta segunda matriz de ecorrelaciones me he deshecho de la mayoría de variables numéricas dependientes entre si, además de haber añadido la variable SEXO. Sin embargo, aún existen algunas variables con un nivel de correlación relevante, es el caso de: EDAD y FECHA\_VIN, INGRESOS E INGRESOS, NROCIUDADES Y SEXO, y Dist\_Sum\_INTER y NROPAISES.

-Generación del modelo

Vuelvo a aplicar missForest, esta vez sobre la totalidad de las variables.

```
df_corr <- df_Fraude
```

```

#transformo todas las variables ch a factor
df_Fraude$SEGMENTO <- as.factor(df_Fraude$SEGMENTO)
df_Fraude$CANAL <- as.factor(df_Fraude$CANAL)
df_Fraude$COD_PAIS <- as.factor(df_Fraude$COD_PAIS)
df_Fraude$Canal1 <- as.factor(df_Fraude$Canal1)

#imputamos valores

help(missForest)
df_impo <- missForest(df_corr)

## Warning in mean.default(xmis[, t.co], na.rm = TRUE): argument is not
numeric or
## logical: returning NA

## Warning in mean.default(xmis[, t.co], na.rm = TRUE): argument is not
numeric or
## logical: returning NA

## Warning in mean.default(xmis[, t.co], na.rm = TRUE): argument is not
numeric or
## logical: returning NA

## Warning in mean.default(xmis[, t.co], na.rm = TRUE): argument is not
numeric or
## logical: returning NA

## missForest iteration 1 in progress...

## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree,
mtry =
## mtry, : The response has five or fewer unique values. Are you sure you
want to
## do regression?

## done!

## Error in FUN(left, right): argumento no-numérico para operador binario

df_impo$OOBerror # errores asociados a cada variable (MSE para continuas
(error cuadrático medio) y PFC(proporción de mala clasificación)
categoricas)

## Error in eval(expr, envir, enclos): objeto 'df_impo' no encontrado

#calculo de varianzas quitando na
apply(df_corr,2,var,na.rm=TRUE)

## Warning in stats::var(...): NAs introducidos por coerción

```

```

## Warning in stats::var(...): NAs introducidos por coerción
## Warning in stats::var(...): NAs introducidos por coerción
## Warning in stats::var(...): NAs introducidos por coerción

##           id      FRAUDE      VALOR      HORA_AUX
Dist_max_NAL
## 9.486176e+19 1.858222e-01 9.720967e+11 4.030481e+01
8.710920e+04
##      Canal1      FECHA      COD_PAIS      CANAL
DIASEM
##      NA 8.344166e+01      NA      NA
4.377654e+00
##      DIAMES      FECHA_VIN      OFICINA_VIN      SEXO
SEGMENTO
## 8.344166e+01 8.575551e+09 8.425289e+04 2.500518e-01
NA
##      EDAD      INGRESOS      EGRESOS      NROPAISES
Dist_Sum_INTER
## 1.683893e+02 3.177928e+15 3.818203e+15 1.086221e+00
6.334293e+08
## Dist_Mean_INTER  Dist_Max_INTER      NROCIUDADES      Dist_Mean_NAL
Dist_HOY
## 3.221412e+06 7.049459e+06 7.562616e+00 3.687406e+04
3.167471e+06
##      Dist_sum_NAL
## 5.753603e+06

apply(is.na(df_impo$ximp), 2, sum) #me indica nº de na en la BBDD imputada,
comprobamos que lo hemos hecho bien.

## Error in apply(is.na(df_impo$ximp), 2, sum): objeto 'df_impo' no
encontrado

df_corr <- df_impo$ximp

## Error in eval(expr, envir, enclos): objeto 'df_impo' no encontrado

apply(is.na(df_corr), 2, sum) #Me aseguro de que todos los valores NA han
sido remplazados

##           id      FRAUDE      VALOR      HORA_AUX
Dist_max_NAL
## 0          0          0          0          0
##      Canal1      FECHA      COD_PAIS      CANAL
DIASEM
## 0          0          0          0          0
##      DIAMES      FECHA_VIN      OFICINA_VIN      SEXO

```

```

SEGMENTO
##          0          24          24          55
0
##          EDAD      INGRESOS      EGRESOS      NROPAISES
Dist_Sum_INTER
##          24          24          24          0
1547
## Dist_Mean_INTER  Dist_Max_INTER  NROCIUDADES  Dist_Mean_NAL
Dist_HOY
##          1547        1547          0          457
0
##  Dist_sum_NAL
##          0

```

Divido el dataset creando las particiones de training (70%) y test (30%)

```

#establezco una semilla para que me salga el mismo resultado siempre que ejecute este código.
set.seed(1)
#Generamos una variable aleatoria con una distribución 70-30
df_corr$random<-sample(0:1,size = nrow(df_corr),replace = T,prob =
c(0.3,0.7))
#Creo dos dataframes
train<-filter(df_corr,random==1)
test<-filter(df_corr,random==0)
df_corr$random <- NULL #Elimino random para que no moleste

# Matrices de entrenamiento y test
#
===== =====
x_train <- model.matrix(FRAUDE~., data = train)[, -1]
y_train <- train$FRAUDE

x_test <- model.matrix(FRAUDE~., data = test)[, -1]
y_test <- test$FRAUDE

```

Una vez tengo mis dataframes de training y test voy a buscar el mejor modelo posible. Mi idea es entrenar un modelo de regresión logística con regularización Ridge o Lasso en train seleccionando el que mejor AUC tenga

Empezaré utilizando el criterio stepAIC para buscar el mejor modelo posible:

```

fit1 <- glm(FRAUDE~., data=train, family=binomial)

## Warning: glm.fit: algorithm did not converge

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

```

```

fit0 <- glm(FRAUDE~1, data=train, family=binomial)

#Aplico both de stepwise

step <-stepAIC(fit0,direction="both",scope=list(upper=fit1,lower=fit0))

## Start: AIC=2335.27
## FRAUDE ~ 1

## Error in x[good, , drop = FALSE]: (subscript) subscripto lógico muy
largo

summary(step)

## Error in object[[i]]: objeto de tipo 'closure' no es subconjunto

```

En este caso, no es un idicador apropiado para seleccionar un modelo ya que los modelos propuestos mejoran su AIC cuanto menos variables explicativas contengan.

Estudiaré diferentes modelos regularizados. Estos modelos regularizados pretenden generar modelos menos sensibles a los datos, añadiendo así sesgo a los modelos, con el objetivo de que estos no caigan en sobreajustes.

*##Aplico Ridge:*

```

set.seed(4) #Semilla
cv.ridge <- cv.glmnet(x_train, y_train, family='binomial', alpha=0,
type.measure='auc')

## Error in glmnet(x, y, weights = weights, offset = offset, lambda =
lambda, : number of observations in y (2081) not equal to the number of
rows of x (838)

plot(cv.ridge)

## Error in h(simpleError(msg, call)): error in evaluating the argument
'x' in selecting a method for function 'plot': objeto 'cv.ridge' no
encontrado

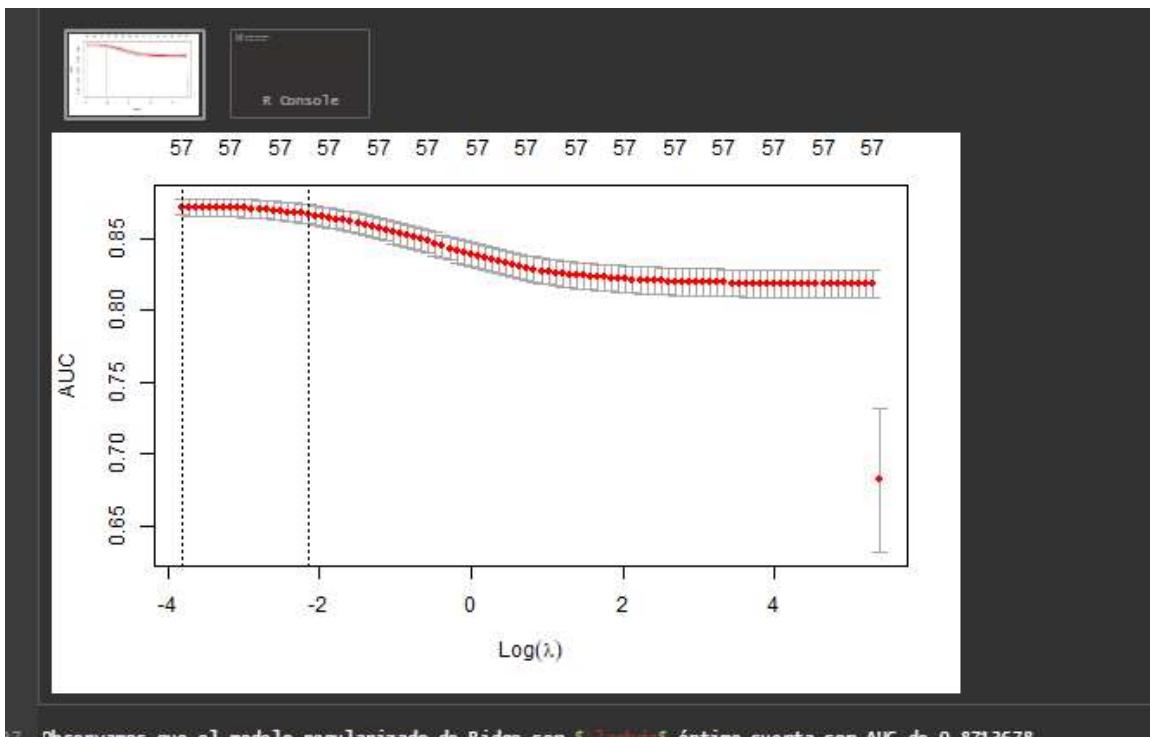
cv.ridge$lambda.min

## Error in eval(expr, envir, enclos): objeto 'cv.ridge' no encontrado

#este es el valor del error que se estima para ese valor Lambda mínimo
dado en AUC
max(cv.ridge$cvm)

## Error in eval(expr, envir, enclos): objeto 'cv.ridge' no encontrado

```



Observamos que el modelo regularizado de Ridge con  $\lambda$  óptimo cuenta con AUC de 0.8713678

Observamos que el modelo regularizado de Ridge con  $\lambda$  óptimo cuenta con AUC de 0.8713678

##Aplico Lasso

```
set.seed(4)
cv.lasso <- cv.glmnet(x_train, y_train, family='binomial', alpha=1,
type.measure='auc')

## Error in glmnet(x, y, weights = weights, offset = offset, lambda =
lambda, : number of observations in y (2081) not equal to the number of
rows of x (838)

plot(cv.lasso)

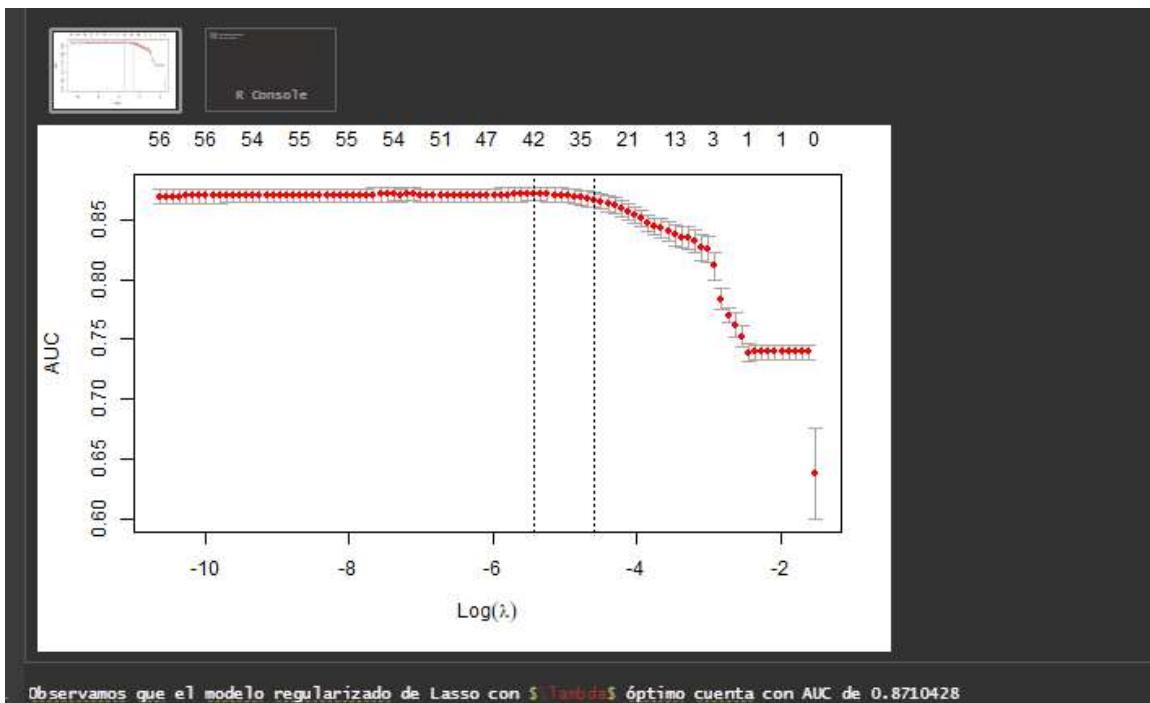
## Error in h(simpleError(msg, call)): error in evaluating the argument
'x' in selecting a method for function 'plot': objeto 'cv.lasso' no
encontrado

cv.lasso$lambda.min

## Error in eval(expr, envir, enclos): objeto 'cv.lasso' no encontrado

#este es el valor del error que se estima para ese valor Lambda mínimo
#dado en AUC
max(cv.lasso$cvm)

## Error in eval(expr, envir, enclos): objeto 'cv.lasso' no encontrado
```



Observamos que el modelo regularizado de Lasso con  $\lambda$  óptimo cuenta con AUC de 0.8710428

Como el AUC del modelo de Ridge es mayor al de Lasso, nos quedamos con el modelo obtenido por la regularización de Ridge.

Vemos los coeficientes del modelo óptimo obtenido por la regularización de Lasso.

```
coef(cv.ridge, s=cv.ridge$lambda.min)
## Error in coef(cv.ridge, s = cv.ridge$lambda.min): objeto 'cv.ridge' no
encontrado
```

Calculo la predicción en test y sus métricas

Muestro las seis primeras predicciones y los correspondientes valores reales.

```
y_pred <- as.numeric(predict.glmnet(cv.ridge$glmnet.fit, newx=x_test,
s=cv.ridge$lambda.min)>.5)
## Error in h(simpleError(msg, call)): error in evaluating the argument
'x' in selecting a method for function 't': error in evaluating the
argument 'x' in selecting a method for function 'as.matrix': objeto
'cv.ridge' no encontrado
y_pred <- as.factor(y_pred)
## Error in is.factor(x): objeto 'y_pred' no encontrado
y_test <- as.factor(y_test)
```

```

head(y_pred)

## Error in h(simpleError(msg, call)): error in evaluating the argument
'x' in selecting a method for function 'head': objeto 'y_pred' no
encontrado

head(y_test)

## [1] 1 1 1 0 0 0
## Levels: 0 1

```

Doy métricas en el test

```

confusionMatrix(y_test, y_pred, mode="everything")

## Error in is.factor(reference): objeto 'y_pred' no encontrado

```

Observamos que el modelo Ridge muestra un accuracy alto del 8326% y una precisión del 96.57%.

## Elastic net en regresión lineal

Finalmente implementamos una regularización Elastic net con una combinación de ambos métodos a partes iguales, por lo que  $\alpha = 0.5$ .

```

set.seed(4)
cv.elastic <- cv.glmnet(x_train, y_train, family='binomial', alpha=0.5,
type.measure='auc')

## Error in glmnet(x, y, weights = weights, offset = offset, lambda =
lambda, : number of observations in y (2081) not equal to the number of
rows of x (838)

# Resultados
plot(cv.elastic)

## Error in h(simpleError(msg, call)): error in evaluating the argument
'x' in selecting a method for function 'plot': objeto 'cv.elastic' no
encontrado

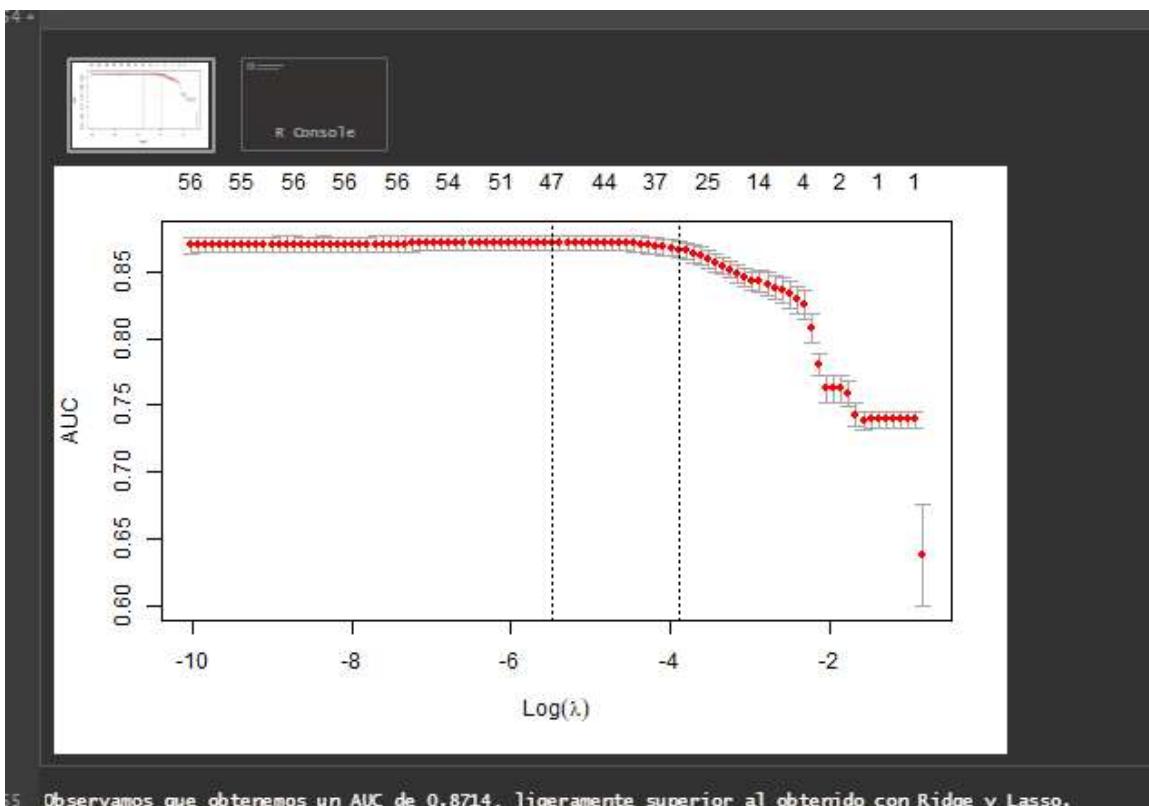
#este es el mejor valor de Lambda
cv.elastic$lambda.min

## Error in eval(expr, envir, enclos): objeto 'cv.elastic' no encontrado

#este es el valor del error que se estima para ese valor Lambda mínimo
# dado en AUC
max(cv.elastic$cvm) # recordemos que el máximo valor del AUC es el mejor
# de los resultados

## Error in eval(expr, envir, enclos): objeto 'cv.elastic' no encontrado

```



55 Observamos que obtenemos un AUC de 0.8714, ligeramente superior al obtenido con Ridge y Lasso.

Observamos que obtenemos un AUC de 0.8714, ligeramente superior al obtenido con Ridge y Lasso.

```
y_pred <- as.numeric(predict.glmnet(cv.elastic$glmnet.fit, newx=x_test,
s=cv.elastic$lambda.min)>.5)

## Error in h(simpleError(msg, call)): error in evaluating the argument
'x' in selecting a method for function 't': error in evaluating the
argument 'x' in selecting a method for function 'as.matrix': objeto
'cv.elastic' no encontrado

y_pred <- as.factor(y_pred)

## Error in is.factor(x): objeto 'y_pred' no encontrado

y_test <- as.factor(y_test)

confusionMatrix(y_test, y_pred, mode="everything")

## Error in is.factor(reference): objeto 'y_pred' no encontrado
```

Voy a tratar de crear mi propio modelo de otra manera, un modelo con todas las variables excepto "id".

#Señalo Las variables independientes y la target del modelo

```

independientes <- setdiff(names(df_corr),c( "id","FRAUDE"))#Las variables
independientes son todas menos id y la variable objetivo
target <- 'FRAUDE'

# Creo La formula para usar en el modelo
formula <- reformulate(independientes,target)

```

Modelizo con regresión logística

```

formula_rl <- formula
rl<- glm(formula_rl,train,family=binomial(link='logit'))

## Warning: glm.fit: algorithm did not converge

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

summary(rl)

##
## Call:
## glm(formula = formula_rl, family = binomial(link = "logit"),
##      data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -3.9679 -0.5083 -0.2933 -0.0958  3.1206
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 6.924e+12 1.024e+14  0.068 0.946089
## VALOR       6.444e-07 1.102e-07  5.848 4.96e-09 ***
## HORA_AUX    3.229e-03 2.086e-02  0.155 0.876994
## Dist_max_NAL 1.218e-03 8.411e-04  1.448 0.147545
## Canal1POS   -3.061e+00 5.033e-01 -6.082 1.19e-09 ***
## FECHA       -3.436e+05 5.082e+06 -0.068 0.946089
## COD_PAISAW  3.792e+00 2.466e+05  0.000 0.999988
## COD_PAISBR  5.665e+01 2.872e+05  0.000 0.999843
## COD_PAISCL  2.283e+01 1.357e+05  0.000 0.999866
## COD_PAISCN  4.886e+01 3.760e+05  0.000 0.999896
## COD_PAISCR  3.308e+06 2.023e+07  0.164 0.870119
## COD_PAISCU  -1.094e+01 3.744e+05  0.000 0.999977
## COD_PAISDO  2.644e+01 1.357e+05  0.000 0.999845
## COD_PAISEC  2.783e+01 1.357e+05  0.000 0.999836
## COD_PAISFR  5.323e+01 3.755e+05  0.000 0.999887
## COD_PAISGB  5.131e+01 2.980e+05  0.000 0.999863
## COD_PAISHK  4.717e+01 3.879e+05  0.000 0.999903
## COD_PAISIT  5.336e+01 3.757e+05  0.000 0.999887
## COD_PAISMX  5.203e+01 1.964e+05  0.000 0.999789
## COD_PAISNI  4.561e+01 2.798e+05  0.000 0.999870
## COD_PAISPA  2.685e+01 1.357e+05  0.000 0.999842
## COD_PAISPH  4.536e+01 2.865e+05  0.000 0.999874

```

```

## COD_PAISPR      2.540e+00  1.955e+05  0.000 0.999990
## COD_PAISUS     2.511e+01  1.357e+05  0.000 0.999852
## COD_PAISUY     3.230e+00  3.887e+05  0.000 0.999993
## CANALMCI      -9.818e-01  2.833e-01  -3.466 0.000528 ***
## CANALPOS        NA          NA          NA          NA
## DIASEM         1.483e-01  6.450e-02  2.299 0.021510 *
## DIAMES         3.436e+05  5.082e+06  0.068 0.946089
## FECHA_VIN     -1.119e-06  1.912e-06  -0.585 0.558520
## OFICINA_VIN    8.038e-05  4.266e-04  0.188 0.850533
## SEXO            -1.721e-01  2.509e-01  -0.686 0.492689
## SEGMENTOEmpresarial -2.088e+01  2.029e+05  0.000 0.999918
## SEGMENTOPersonal   4.458e+00  1.033e+00  4.314 1.60e-05 ***
## SEGMENTOPersonal Plus 2.725e+00  8.844e-01  3.081 0.002062 **
## SEGMENTOPreferencial 2.159e+00  9.015e-01  2.395 0.016643 *
## SEGMENTOPYME      1.056e+00  1.355e+00  0.779 0.436103
## EDAD             1.703e-02  1.545e-02  1.102 0.270478
## INGRESOS         1.783e-09  1.471e-09  1.212 0.225452
## EGRESOS          -4.392e-09  4.912e-09  -0.894 0.371227
## NROPAISES        7.759e-02  2.671e-01  0.291 0.771416
## Dist_Sum_INTER   -4.348e-05  1.557e-05  -2.792 0.005237 **
## Dist_Mean_INTER  1.956e-04  2.106e-04  0.929 0.352945
## Dist_Max_INTER   -1.374e-04  1.716e-04  -0.801 0.423336
## NROCIUDADES      1.392e-01  7.323e-02  1.901 0.057336 .
## Dist_Mean_NAL    1.490e-03  1.308e-03  1.139 0.254618
## Dist_HOY          3.814e-04  1.314e-04  2.903 0.003701 **
## Dist_sum_NAL     -4.135e-05  6.339e-05  -0.652 0.514192
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 850.73 on 837 degrees of freedom
## Residual deviance: 503.64 on 791 degrees of freedom
## (1243 observations deleted due to missingness)
## AIC: 597.64
##
## Number of Fisher Scoring iterations: 25

```

mantengo todas las variables con alta significatividad (que tengan tres estrellas) y lanzo un segundo modelo con estas.

```
a_mantener <- c("VALOR", "Canal1", "CANAL", "SEGMENTO") #mantengo solo Las variables con una alta significatividad.
```

Modelizo de nuevo con las 4 variables seleccionadas.

```
formula_rl <- reformulate(a_mantener,target)
rl<- glm(formula_rl,train,family=binomial(link='logit'))
summary(rl)
```

```

## 
## Call:
## glm(formula = formula_rl, family = binomial(link = "logit"),
##      data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -3.3201 -0.5307 -0.5094 -0.1251  2.6251
##
## Coefficients: (1 not defined because of singularities)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -2.160e+00 1.067e+00 -2.025 0.04287 *
## VALOR                  3.756e-07 6.036e-08  6.223 4.88e-10 ***
## Canal1POS             -2.664e+00 1.585e-01 -16.809 < 2e-16 ***
## CANALMCI              -2.848e-02 1.546e-01 -0.184  0.85386
## CANALPOS                NA        NA       NA      NA
## SEGMENTOEmprendedor   2.113e+00 1.092e+00  1.934 0.05307 .
## SEGMENTOEmpresarial  -8.830e+00 3.089e+02 -0.029 0.97720
## SEGMENTOPersonal      2.831e+00 1.088e+00  2.601 0.00931 **
## SEGMENTOPersonal Plus 2.843e+00 1.071e+00  2.654 0.00797 **
## SEGMENTOPreferencial 2.795e+00 1.079e+00  2.591 0.00958 **
## SEGMENTOPYME            1.404e+00 1.184e+00  1.186 0.23547
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 2333.3 on 2080 degrees of freedom
## Residual deviance: 1783.9 on 2071 degrees of freedom
## AIC: 1803.9
##
## Number of Fisher Scoring iterations: 12

step <- stepAIC(rl, trace=TRUE, direction="both")

## Start:  AIC=1803.88
## FRAUDE ~ VALOR + Canal1 + CANAL + SEGMENTO
##
##
## Step:  AIC=1803.88
## FRAUDE ~ VALOR + CANAL + SEGMENTO
##
##          Df Deviance    AIC
## <none>      1783.9 1803.9
## - SEGMENTO  6   1815.5 1823.5
## - VALOR    1   1829.4 1847.4
## - CANAL    2   2189.6 2205.6

summary(step)

```

```

## 
## Call:
## glm(formula = FRAUDE ~ VALOR + CANAL + SEGMENTO, family =
## binomial(link = "logit"),
##       data = train)
##
## Deviance Residuals:
##      Min        1Q     Median        3Q       Max
## -3.3201  -0.5307  -0.5094  -0.1251   2.6251
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -2.160e+00  1.067e+00 -2.025  0.04287 *
## VALOR                  3.756e-07  6.036e-08  6.223 4.88e-10 ***
## CANALMCI              -2.693e+00  1.709e-01 -15.751 < 2e-16 ***
## CANALPOS              -2.664e+00  1.585e-01 -16.809 < 2e-16 ***
## SEGMENTOEmprendedor    2.113e+00  1.092e+00  1.934  0.05307 .
## SEGMENTOEmpresarial   -8.830e+00  3.089e+02 -0.029  0.97720
## SEGMENTOPersonal       2.831e+00  1.088e+00  2.601  0.00931 **
## SEGMENTOPersonal Plus  2.843e+00  1.071e+00  2.654  0.00797 **
## SEGMENTOPreferencial   2.795e+00  1.079e+00  2.591  0.00958 **
## SEGMENTOPYME            1.404e+00  1.184e+00  1.186  0.23547
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 2333.3 on 2080 degrees of freedom
## Residual deviance: 1783.9 on 2071 degrees of freedom
## AIC: 1803.9
##
## Number of Fisher Scoring iterations: 12

```

Veo que el criterio AIC ha mejorado significativamente.

Calculo el pseudo R cuadrado de McFadden para esta última modelización:

```

#Los resultados entre 0,2 y 0,4 indican un buen ajuste del modelo.
pr2_rl <- 1 -(rl$deviance / rl>null.deviance)
pr2_rl

## [1] 0.2354577

```

Aplicamos el modelo al conjunto de test, generando un vector con las probabilidades

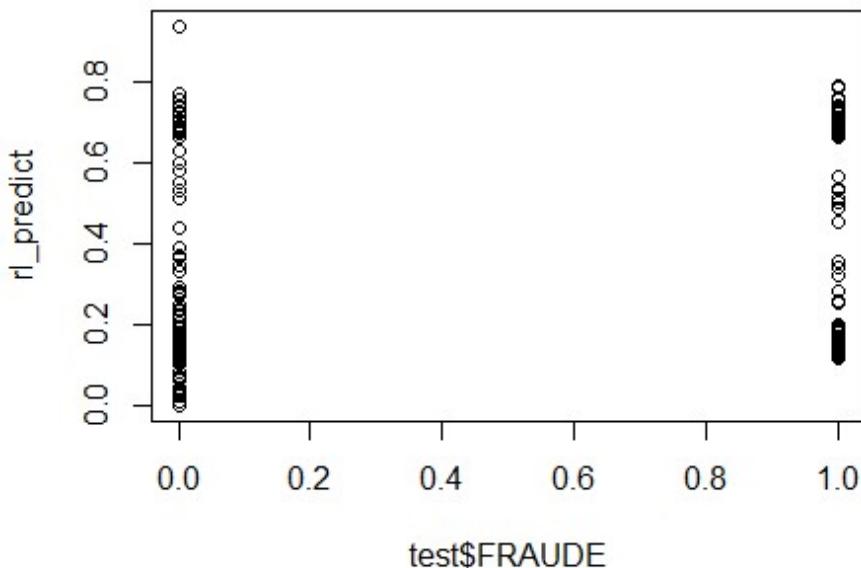
```

rl_predict<-predict(rl,test,type = 'response') #'response' para que nos
diga La probabilidad no la predicción

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if
(type == :
## prediction from a rank-deficient fit may be misleading

```

```
head(r1_predict)
##      1      2      3      4      5      6
## 0.6642690 0.4881496 0.6642690 0.6615465 0.6615465 0.6615465
plot(r1_predict~test$FRAUDE)
```



```
mean(r1_predict)
## [1] 0.2526783
```

Al lanzar un “head” podemos ver las probabilidades para los 6 primeros sujetos. Por ejemplo: el sujeto 1 tendría una probabilidad de cometer fraude del 66%, mientras que el segundo sujeto tiene una probabilidad del 48%.

```
tinytex::install_tinytex()
```