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 este trabajo 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 Plu
s" "Personal Plus" ...
##  $ EDAD           : int  29 29 29 29 25 20 29 28 21 28 ...
##  $ INGRESOS       : int  1200000 5643700 1200000 1200000 0 4000000 210
0000 2000000 500000 4000000 ...
##  $ EGRESOS        : int  1200000 500000 1200000 1200000 0 2500000 3100
00 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
```

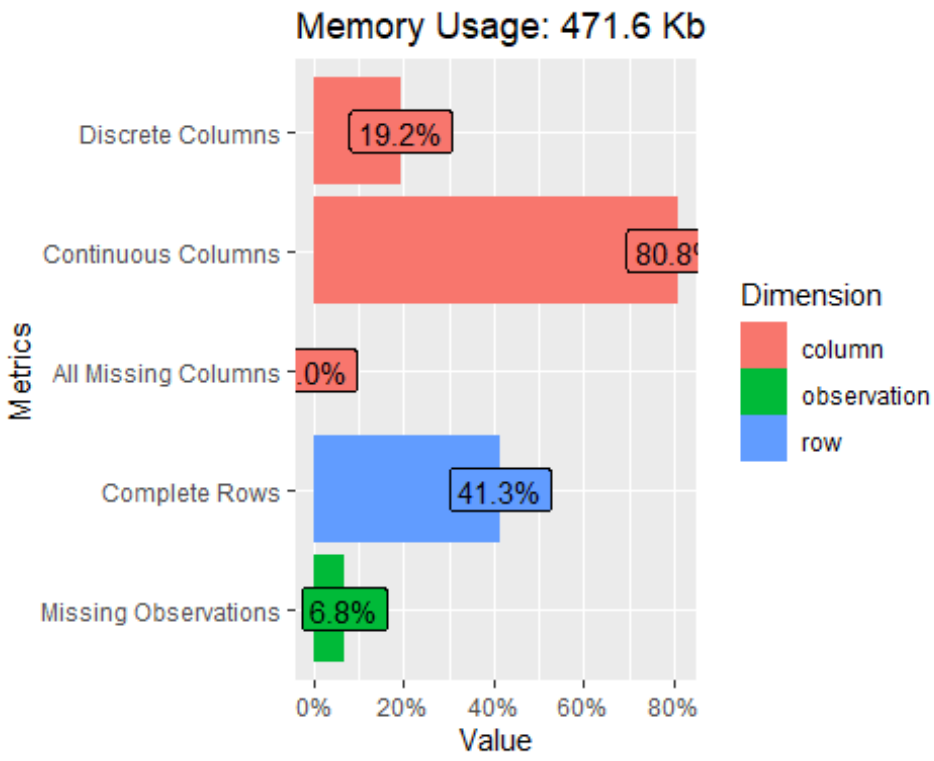
```
##      id FRAUDE VALOR HORA_AUX Dist_max_NAL  Canal1    FECHA COD_PAIS
CANAL
## 1 9e+09      1     0       13       659.13 ATM_INT 20150501       US A
TM_INT
```

```
## 2 9e+09      1     0       17       594.77 ATM_INT 20150515       US A
TM_INT
## 3 9e+09      1     0       13       659.13 ATM_INT 20150501       US A
TM_INT
## 4 9e+09      1     0       13       659.13 ATM_INT 20150501       US A
TM_INT
## 5 9e+09      1     0        0         1.00 ATM_INT 20150510       CR A
TM_INT
## 6 9e+09      1     0       13         1.00 ATM_INT 20150523       US A
TM_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)
```



Según esto, este dataset se compone de un 19.2% de variables continuas y un 80.8% de variables discretas. No hay columnas vacías, pero únicamente el 41.3% de ellas tiene todos sus campos completos. También se ve como hay un 6.8% de valores nulos.

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

```
#describe(df_Fraude)
#summary(df_Fraude)
```

Este es un análisis bastante completo, a diferencia del comando summary(), 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:

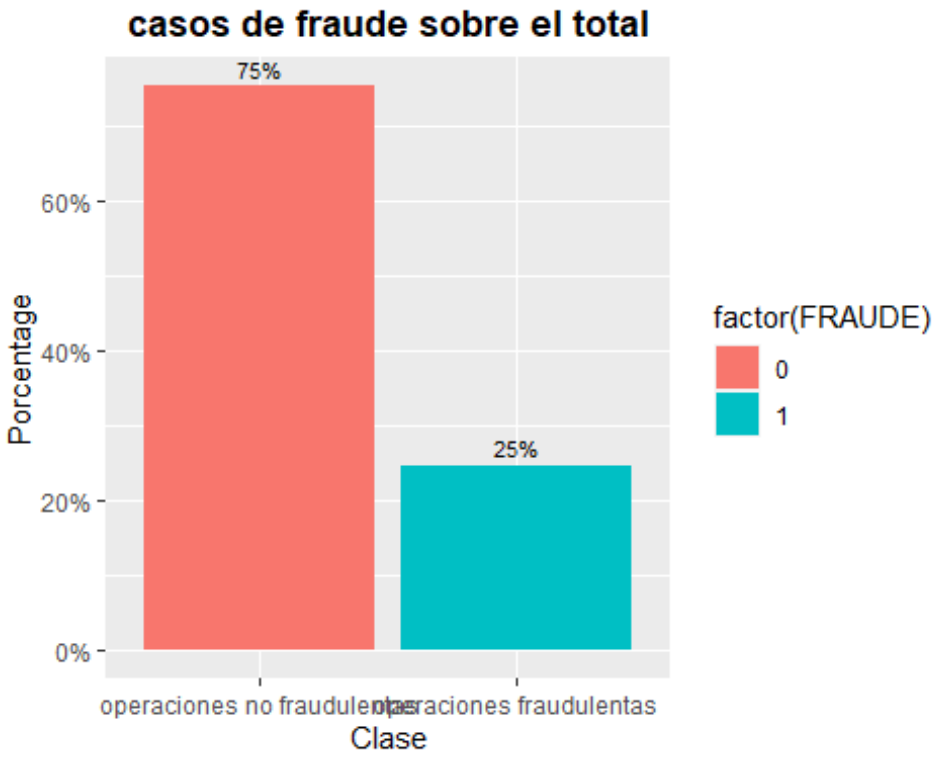
```
apply(is.na(df_Fraude), 2, sum)   #suma de valores NA por columna

##              id          FRAUDE           VALOR        HORA_AUX     Dis
t_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               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    Dis
t_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, 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(FRAU
DE),
                             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", "operacion
es fraudulentas"))+
    scale_y_continuous(labels = scales::percent)+
    labs(x = 'Clase', y = 'Porcentage') +
    ggtitle("casos de fraude sobre el total") +
    common_theme
```

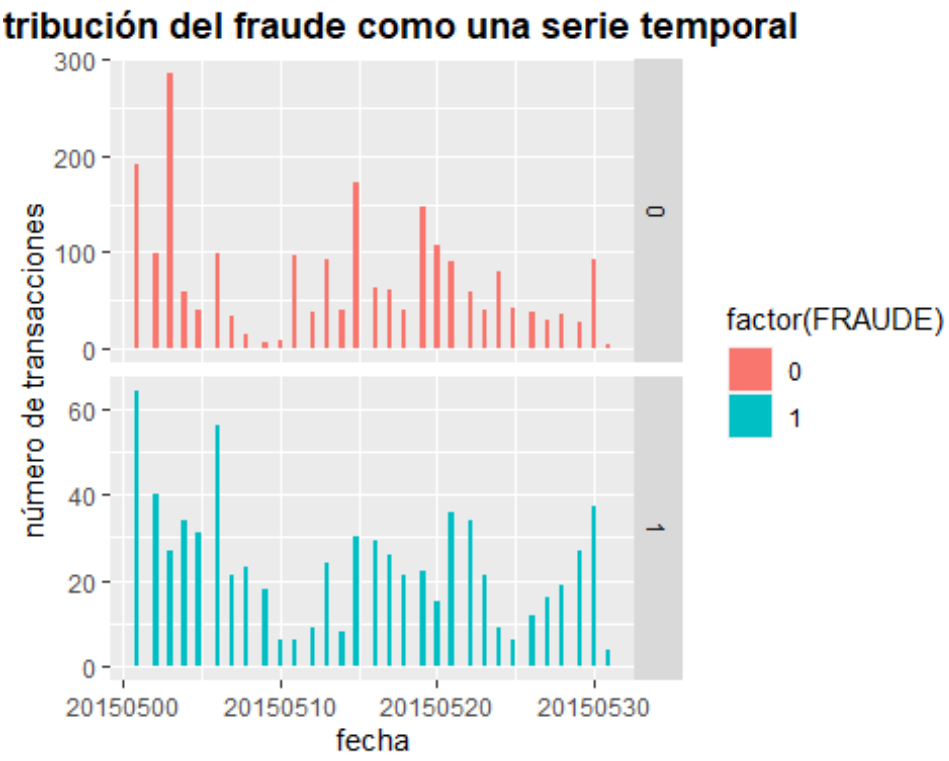


El 25% de las transacciones serían fraudulentas.

El siguiente paso sería estudiar la relación de la variable FRAUDE con el resto de variables, de forma individual.

Represento la posible relación con la variable FECHA:

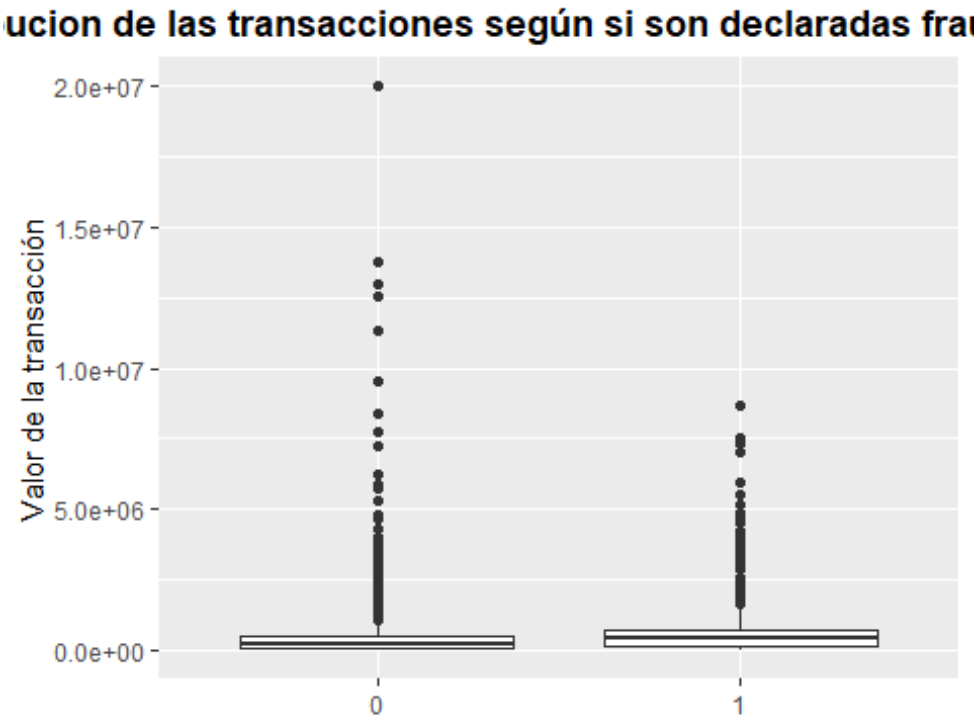
```
df_Fraude %>%
  ggplot(aes(x = FECHA, fill = factor(FRAUDE))) + geom_histogram(bins = 1
00)+
  labs(x = 'fecha', y = 'número de transacciones') +
  ggtitle('Distribución del fraude como una serie temporal') +
  facet_grid(FRAUDE ~ ., scales = 'free_y') + common_theme
```



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 fraudu
lentas o no") + common_theme
```




```
mean(df_Fraude[df_Fraude$FRAUDE==1,"VALOR"]) #la media de las transaccion
es fraudulentas es 666583.7
```

```
## [1] 666583.7
```

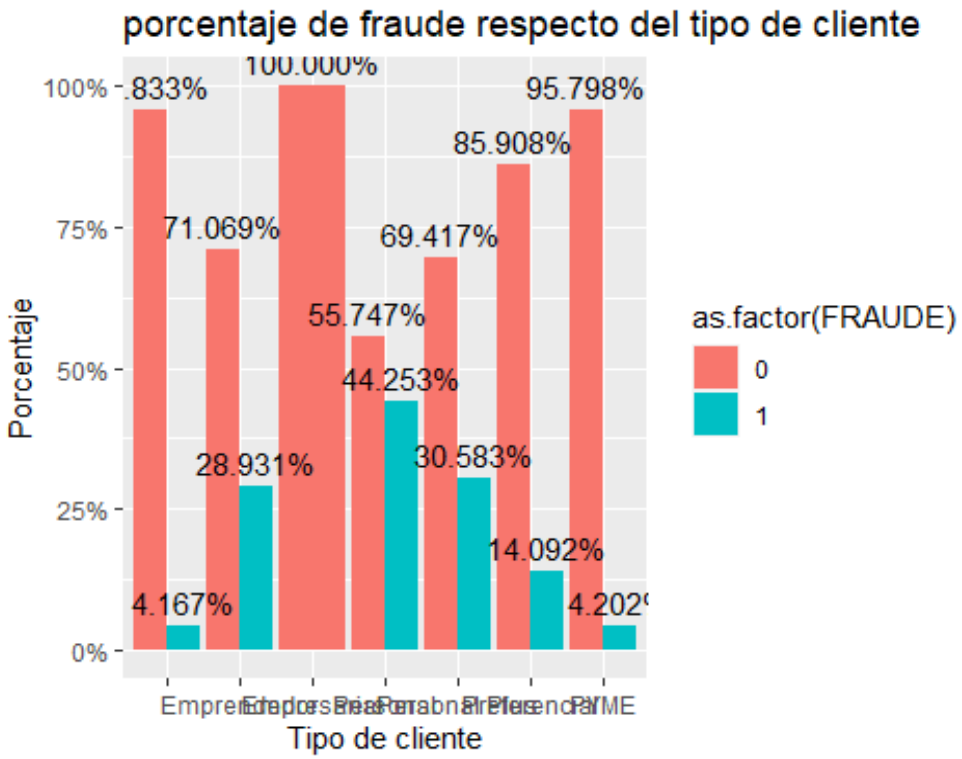
```
mean(df_Fraude[df_Fraude$FRAUDE==0,"VALOR"]) #la media de las transaccion
es legales es 450228.7
```

```
## [1] 450228.7
```

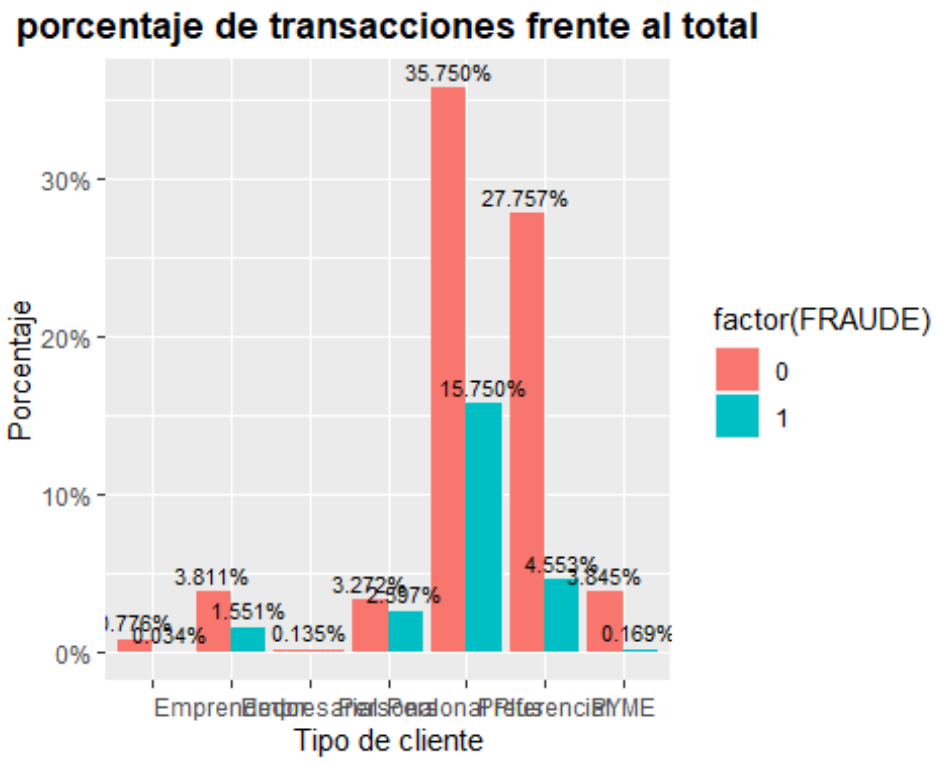
Existe una mayor variabilidad dentro de las transacciones no fraudulentas. Además, la media de las transaccines declaradas fraudulentas es superior a la de las transacciones legales.

A continuación respresento 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..]), positi
on="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")
```



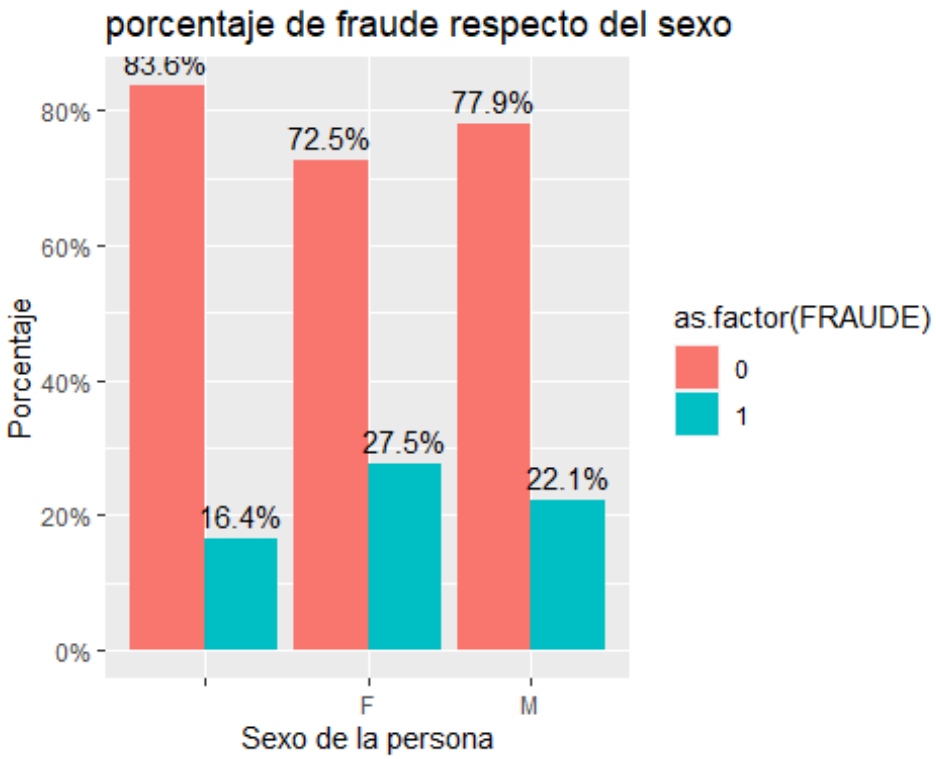

```
common_theme <- theme(plot.title = element_text(hjust = 0.5, face = "bold
"))

ggplot(data = df_Fraude, aes(x = factor(SEGMENTO),
                             y = prop.table(stat(count)), fill = factor(FRAU
DE),
                             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
```



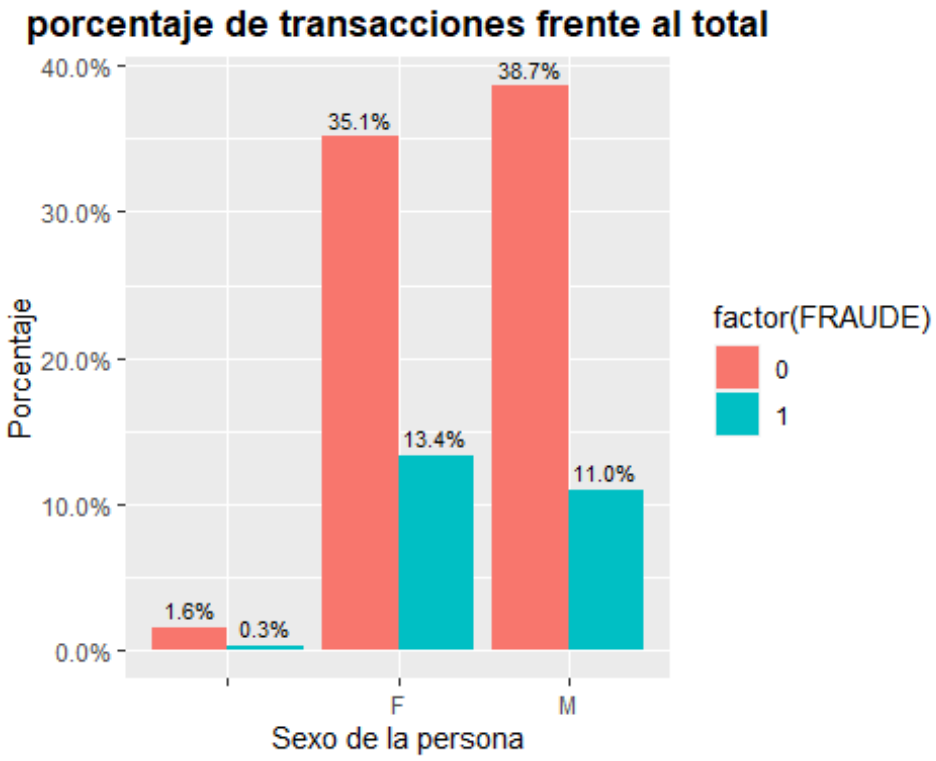
El primer gráfico muestra cómo las transacciones realizadas por el segmento personal tienen el mayor porcentaje de operaciones declaradas fraudulentas, con un 44.253%. Al contrario, el segmento Empresarial parece no tener intentos de fraude. Por optro lado, el segundo gráfico nos muestra como la mayoría de las transacciones se producen por el segmento personal, seguido del PusPreferencial.

De la misma forma, la relación entre FRAUDE y SEXO se muestra en el siguinete gráfico:

```
ggplot(df_Fraude, aes(x = factor(SEXO), fill=as.factor(FRAUDE)))+
  geom_bar(aes( y=..count../tapply(..count.., ..x.. ,sum)[..x..]), positi
on="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")
```



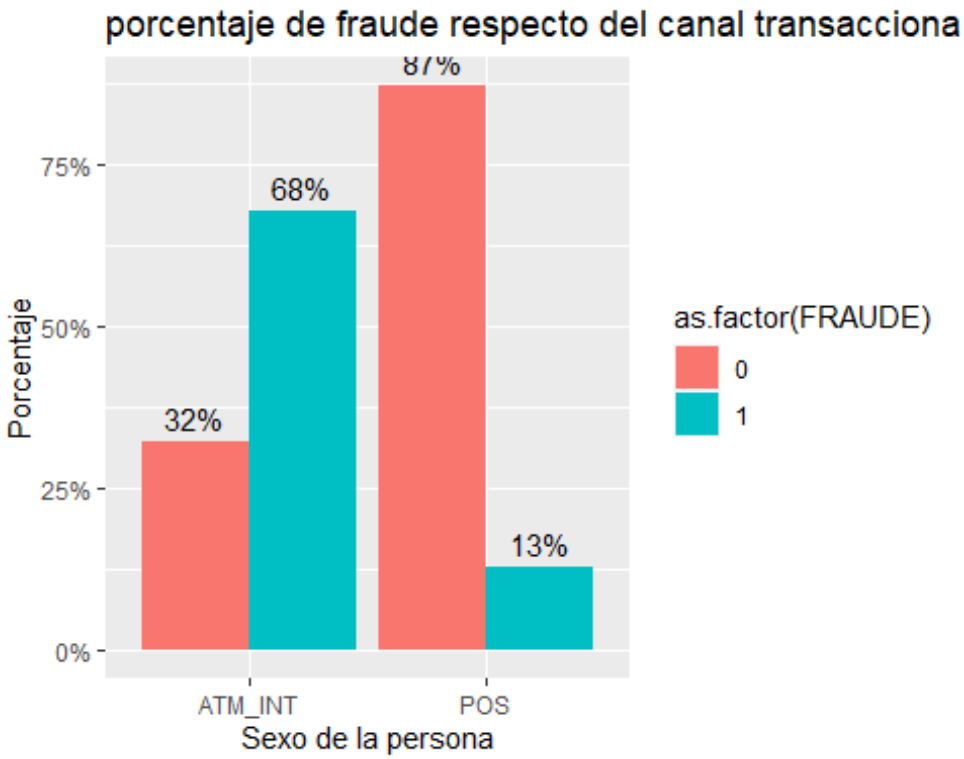

```
common_theme <- theme(plot.title = element_text(hjust = 0.5, face = "bold
"))


ggplot(data = df_Fraude, aes(x = factor(SEXO),
                             y = prop.table(stat(count)), fill = factor(FRAU
DE),
                             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
```



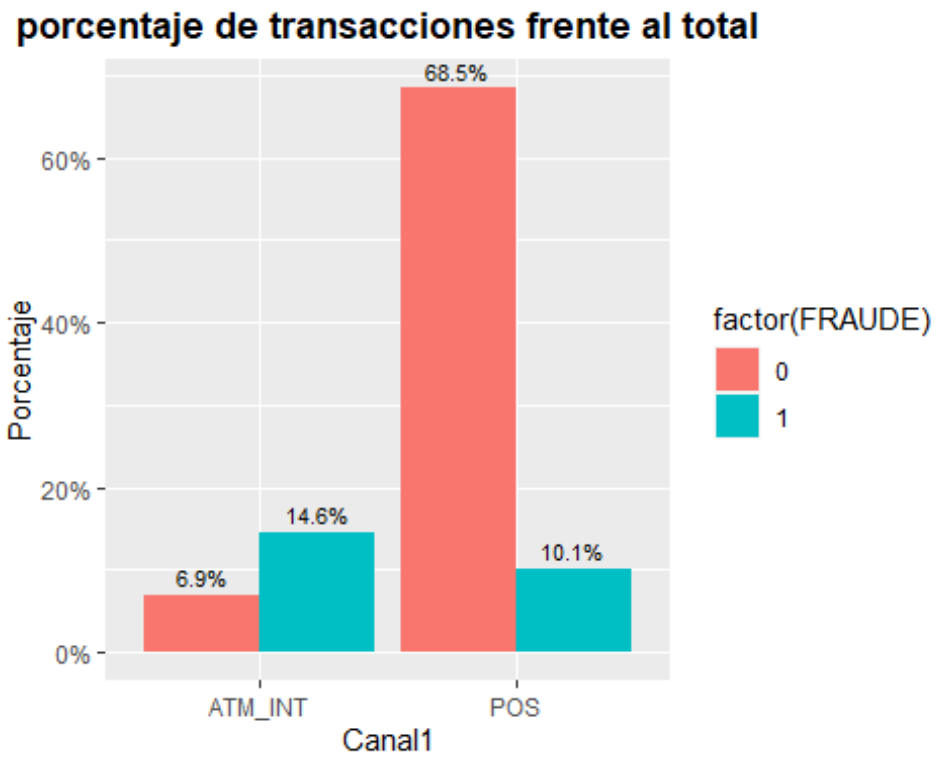
Se puede observar como el porcentaje de mujeres que cometen fraude es superior al de los hombres (27.5% frente al 22.1%). Además, el número de mujeres que han realizado operaciones declaradas fraudulentas es superior al de los hombres (13.4% frente al 11% en los hombres).

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..]), positi
on="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")
```



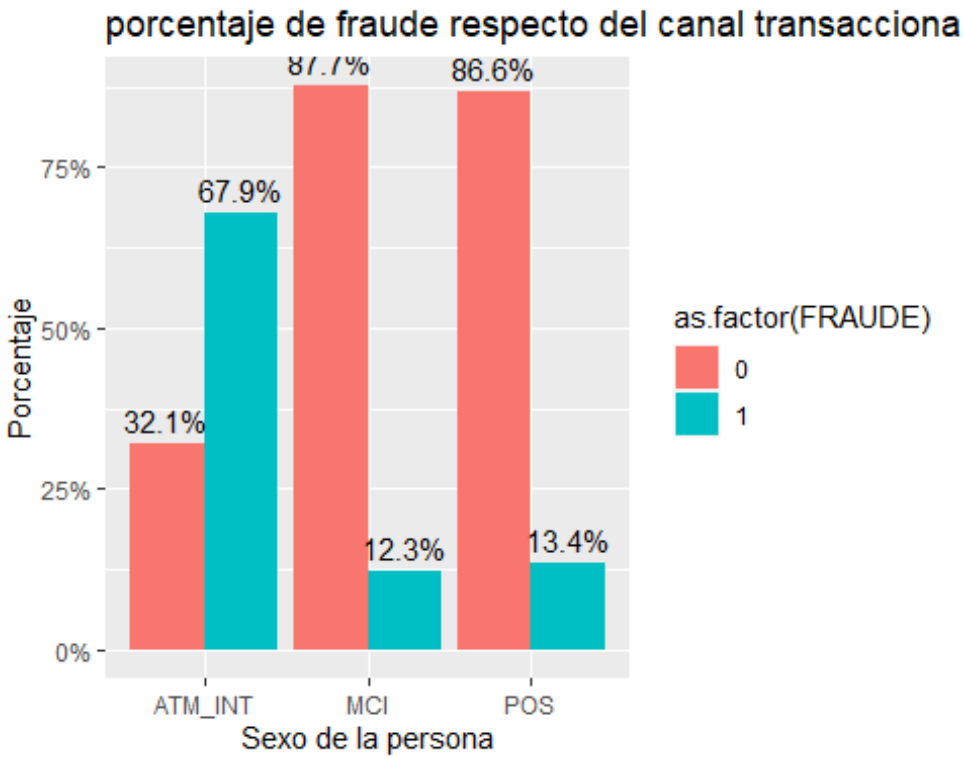

```
ggplot(data = df_Fraude, aes(x = factor(Canal1),
                             y = prop.table(stat(count)), fill = factor(FRAU
DE),
                             label = scales::percent(prop.table(stat(count))
))) +
    geom_bar(position = "dodge") +
    geom_text(stat = 'count',
              position = 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
```



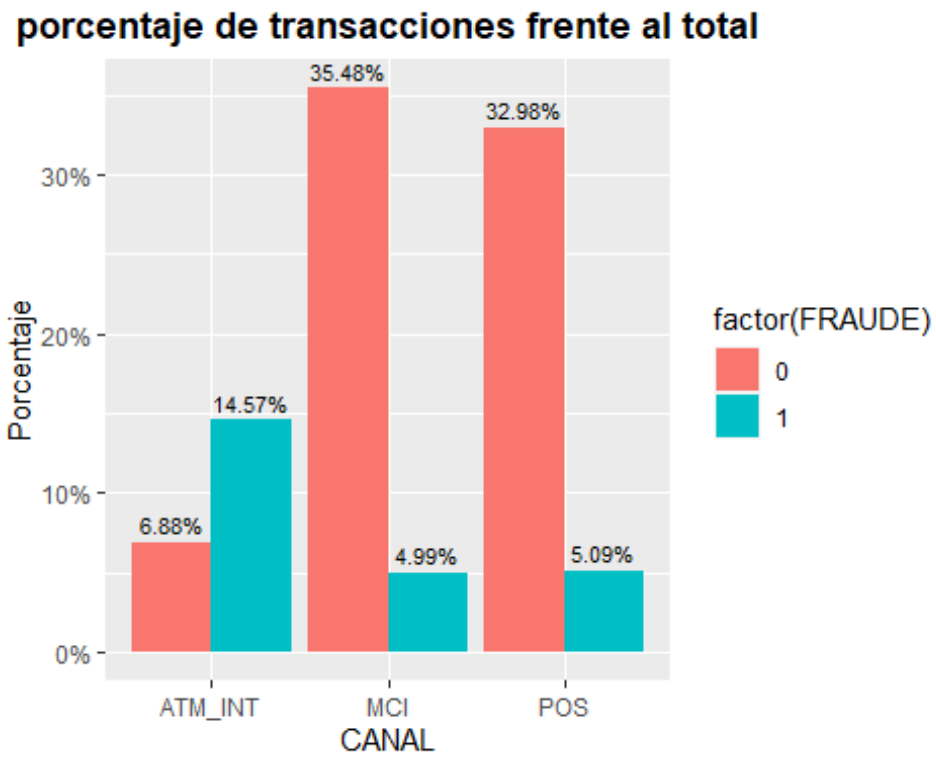
La Mayoría de las transacciones a través de ATM_INT son fraudulentas, aunque la inmensa mayoría de transacciones se realice en POS, donde el fraude es menor.

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..]), positi
on="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")
```



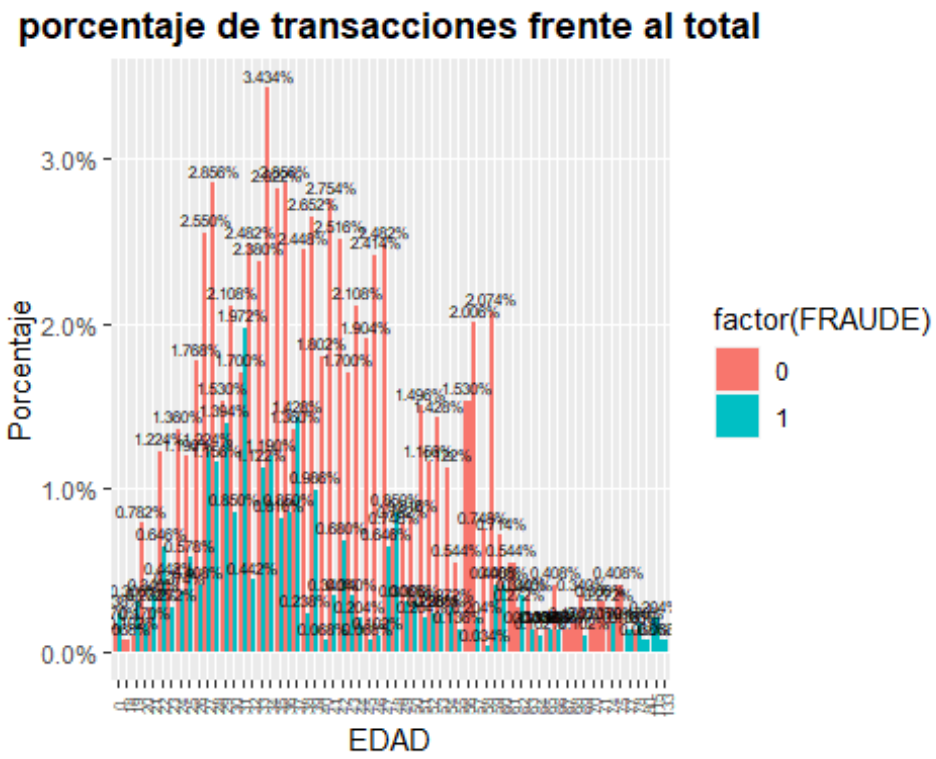

```
ggplot(data = df_Fraude, aes(x = factor(CANAL),
                             y = prop.table(stat(count)), fill = factor(FRAU
DE),
                             label = scales::percent(prop.table(stat(count))
))) +
    geom_bar(position = "dodge") +
    geom_text(stat = 'count',
              position = 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
```



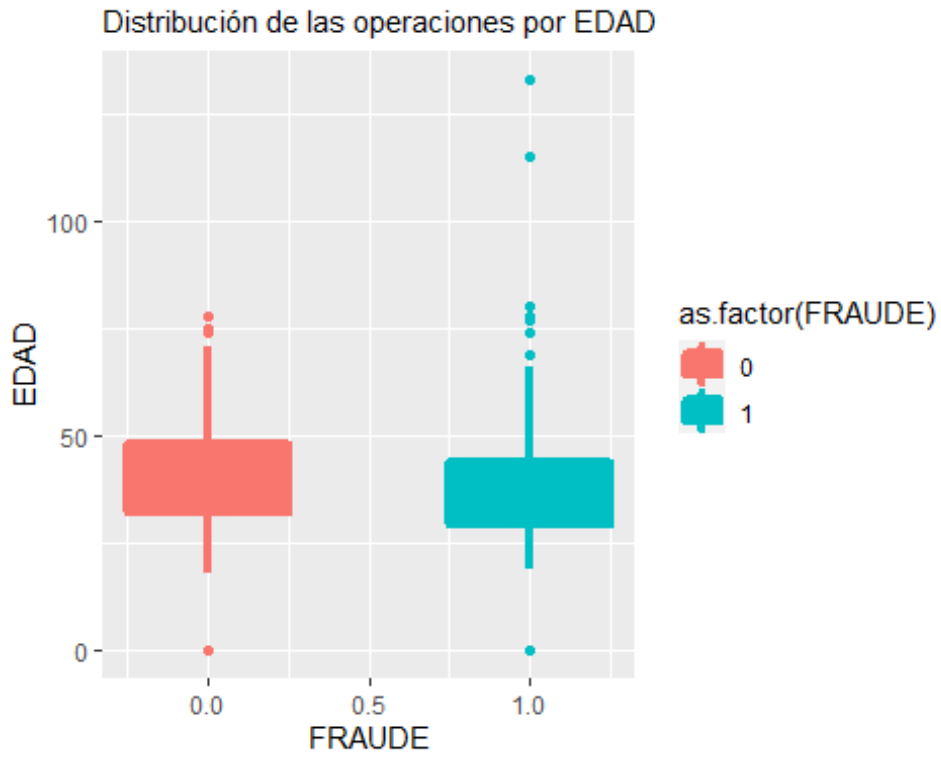
En este caso, El canal donde se comete más fraude sigue siento el ATM_INT, pero el más utilizado es MCI.

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 variab
le EDAD

ggplot(data = df_edad, aes(x = factor(EDAD),
                           y = prop.table(stat(count)), fill = factor(FRAU
DE),
                           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())
```



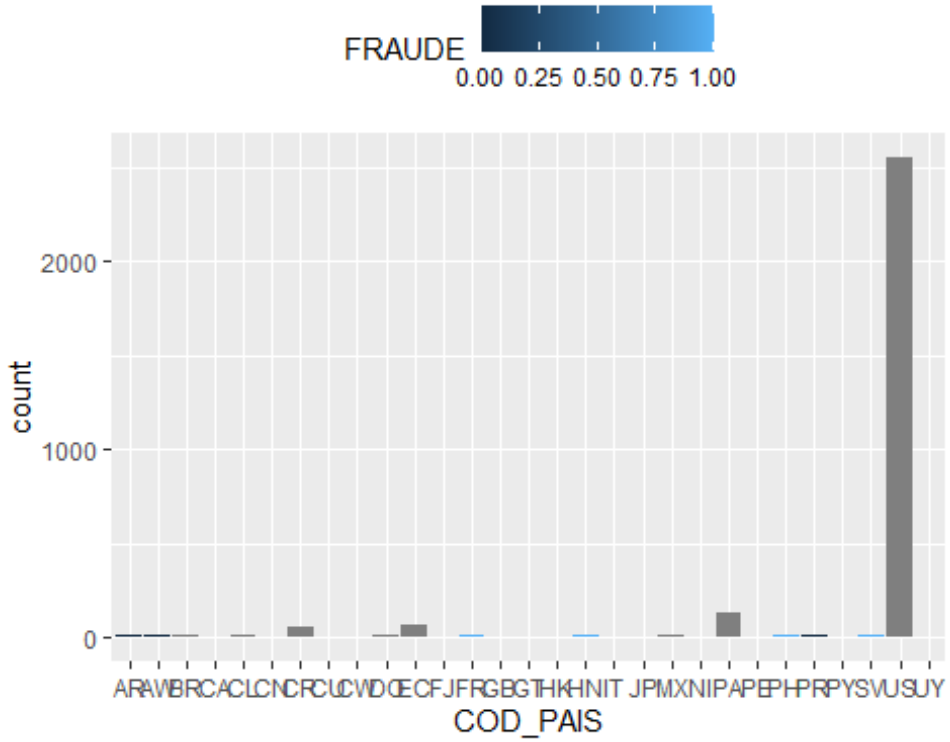

```
df_Fraude %>%
  ggplot(aes(x=FRAUDE, y=EDAD, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  #geom_jitter(width=0.15)+
  labs(subtitle="Distribución de las operaciones por EDAD")

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



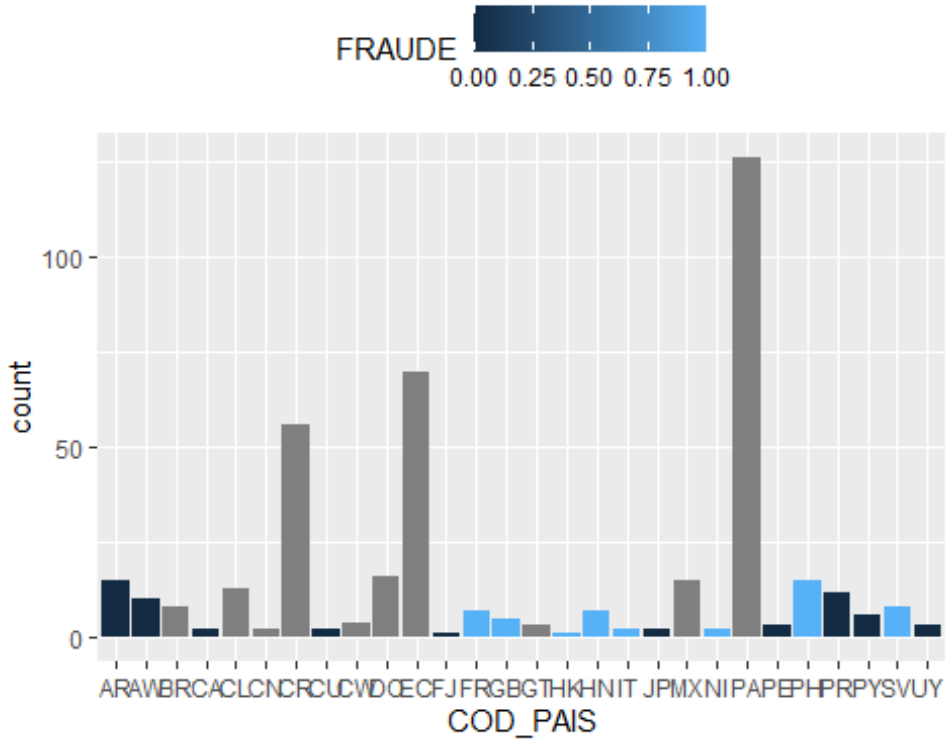
En los dos tipos de gráfico se pueden observar valores atípicos, en especial para las transacciones declaradas fraudulentas.

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 d
e las operaciones vienen de EEUU por lo que necesito eliminar "US" del gr
áfico 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 mayoria de las transacciones ocurrren 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
```

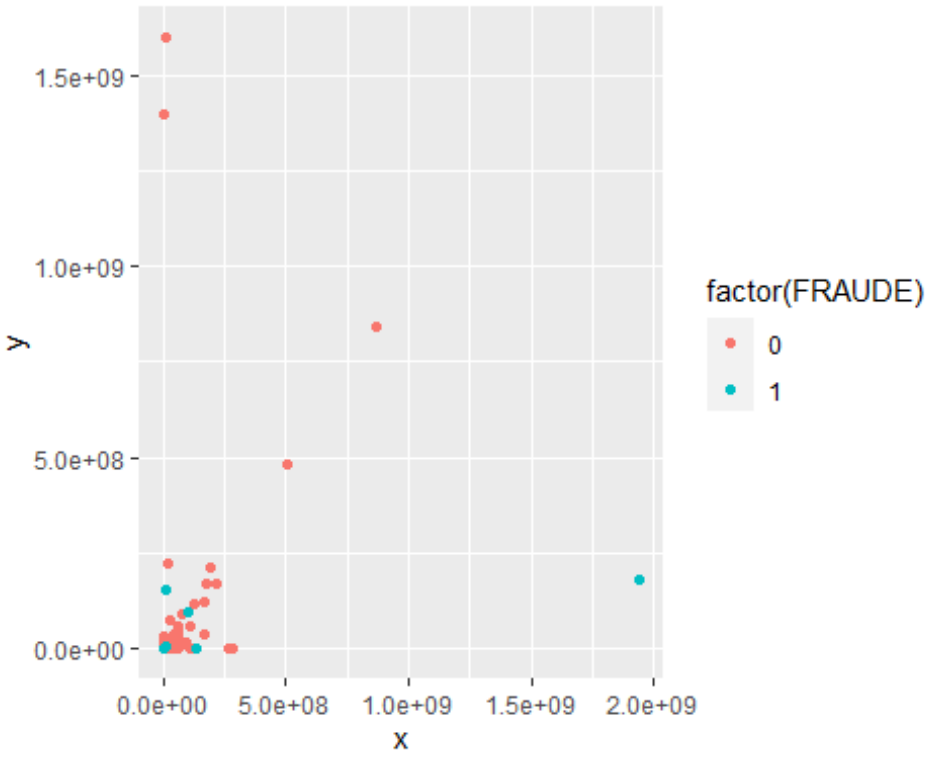
```
##    COD_PAIS    FRAUDE
## 13       FR 1.0000000
## 14       GB 1.0000000
## 16       HK 1.0000000
## 17       HN 1.0000000
## 18       IT 1.0000000
## 21       NI 1.0000000
## 24       PH 1.0000000
## 27       SV 1.0000000
## 7        CR 0.9642857
## 3        BR 0.8750000
## 20       MX 0.8666667
## 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
```

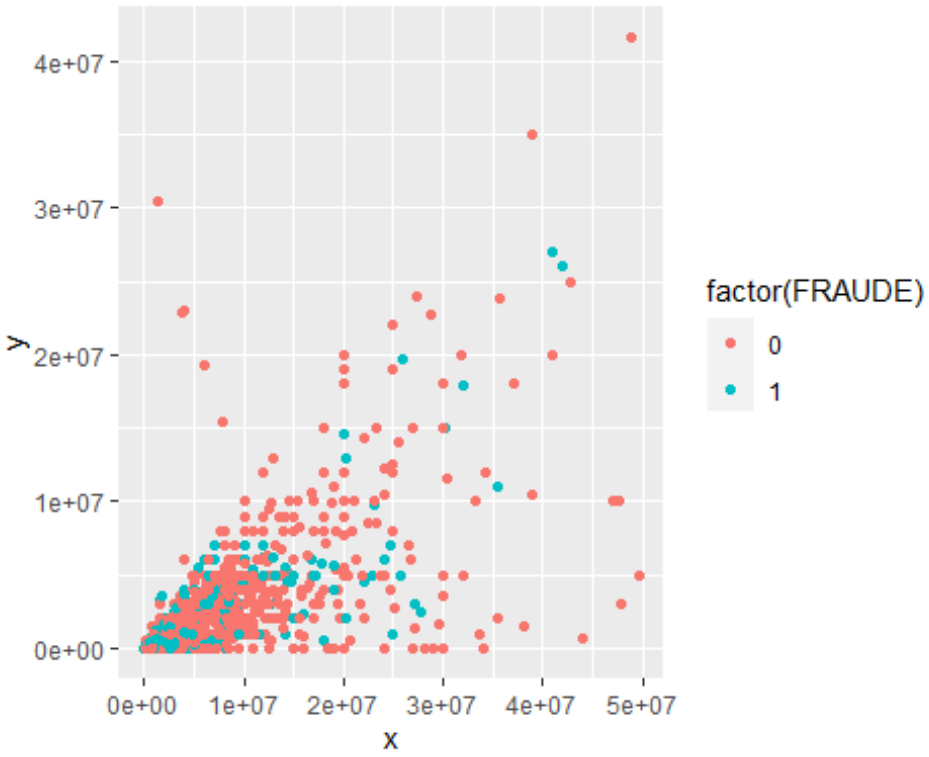
Sin embargo, como se ha dicho antes, la mayoría de observaciones provienen de unos pocos países por lo que los valores de la función de agregación anterior no son realmente representativos.

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

```
# Considero que el gáfico boxplot no poco explicativo en este caso en con
creto.

ggplot(df_Fraude, aes_(df_Fraude$INGRESOS,df_Fraude$EGRESOS)) + geom_poin
t(aes(color =factor(FRAUDE)))

## Warning: Removed 24 rows containing missing values (geom_point).
```



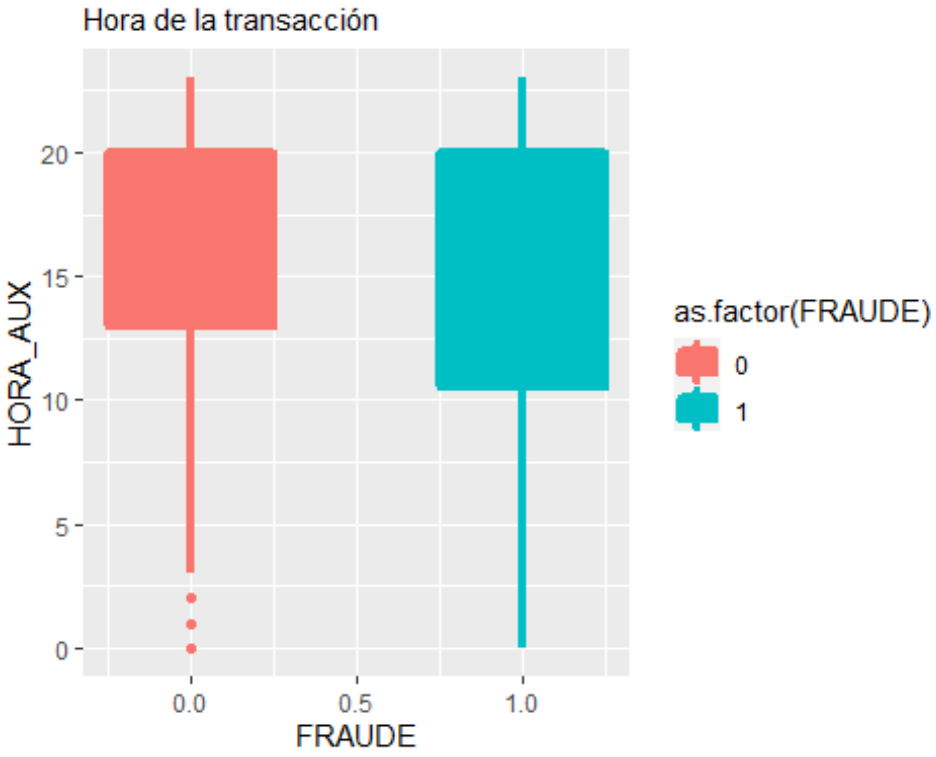

```
#Debido a la dispersión de esta variable sería necesario acotar los datos antes de realizar el mismo gráfico.

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 limites para el gráfico.
ggplot(p, aes_(p$INGRESOS,p$EGRESOS)) + geom_point(aes(color =factor(FRAUDE)))
```



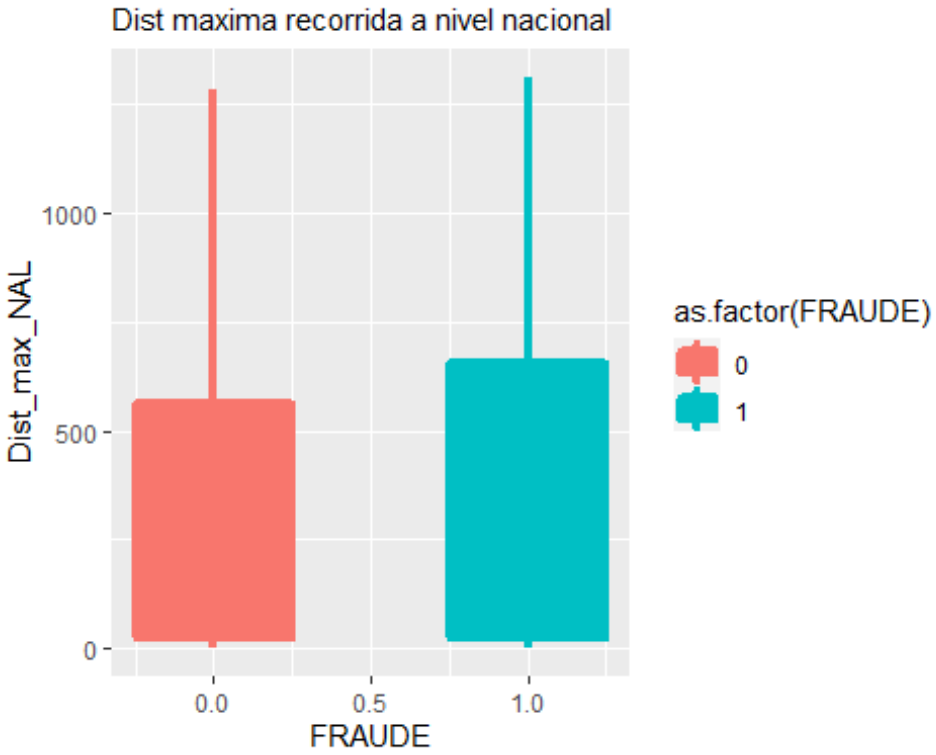
Se ve como las variables INGRESOS y ENGRESOS se distribuyen de la misma manera siendo casos de fraude o no. Por lo tanto, se puede deducir que la variable FRAUDE no estará muy correlacionada con ninguna de las dos variables.

Para el resto de variables he optado por utilizar graficos geom_boxplot como en el ejemplo anterior.

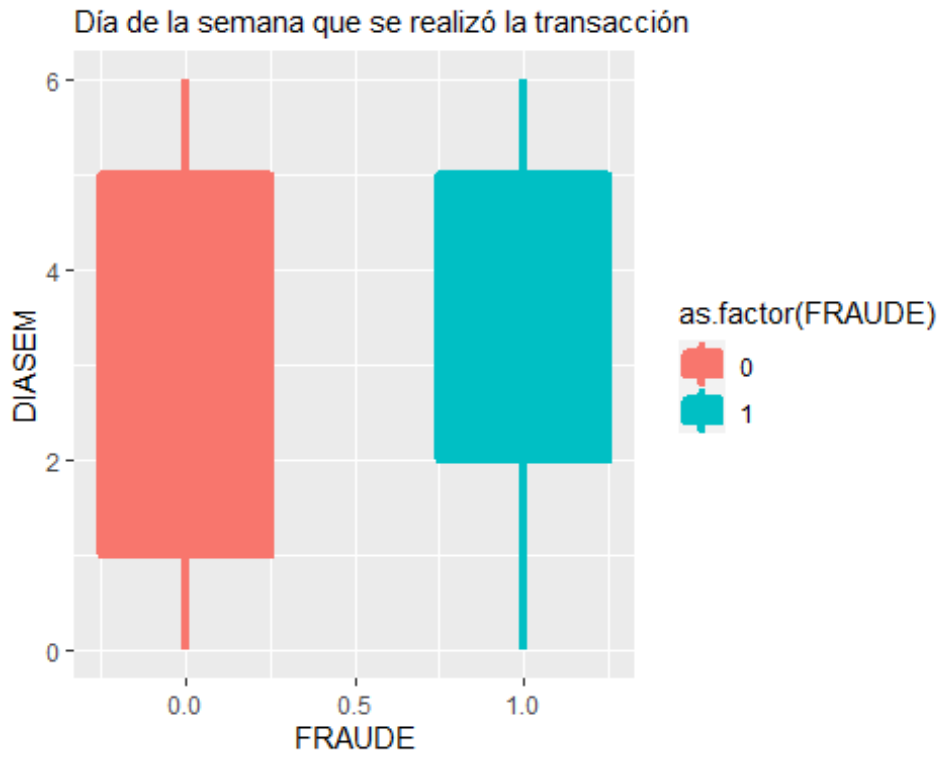
```
df_Fraude %>%
  ggplot(aes(x=FRAUDE, y=HORA_AUX, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  #geom_jitter(width=0.15)+
  labs(subtitle="Hora de la transacción")
```



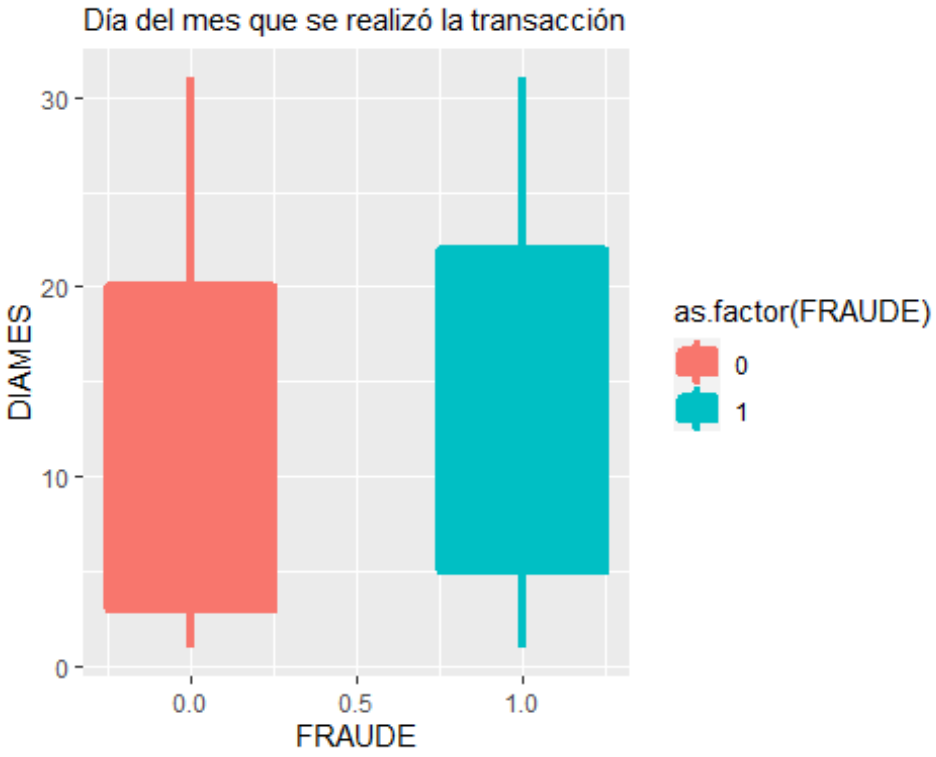

```
df_Fraude %>%
  ggplot(aes(x=FRAUDE, y=Dist_max_NAL, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  #geom_jitter(width=0.15)+
  labs(subtitle="Dist maxima recorrida a nivel nacional")
```



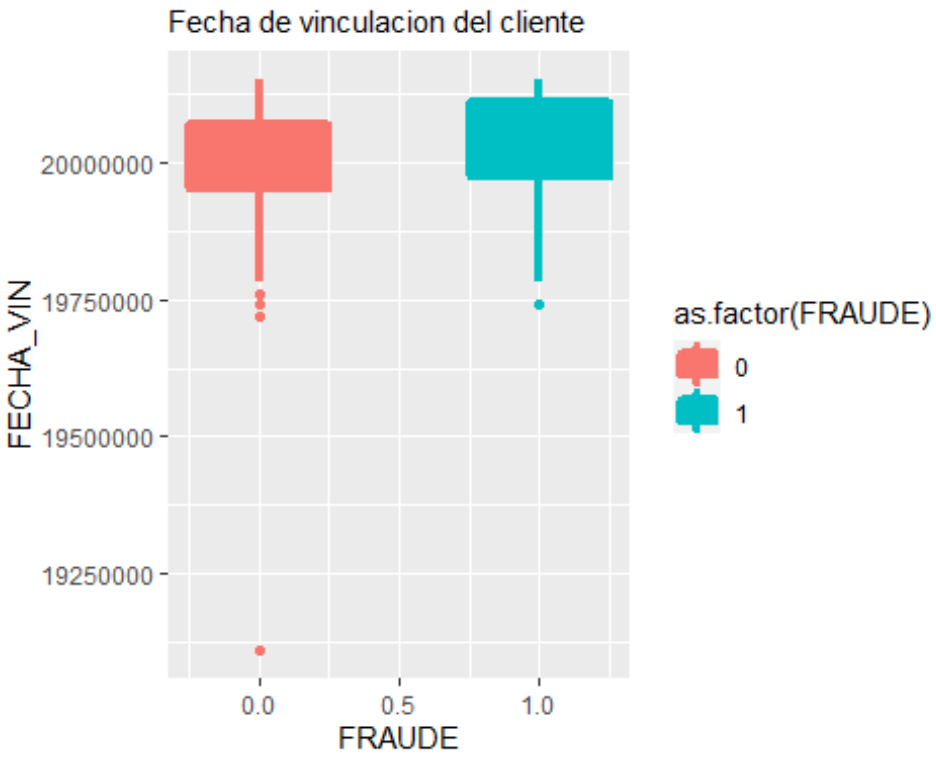

```
df_Fraude %>%
  ggplot(aes(x=FRAUDE, y=DIASEM, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  #geom_jitter(width=0.15)+
  labs(subtitle="Día de la semana que se realizó la transacción")
```



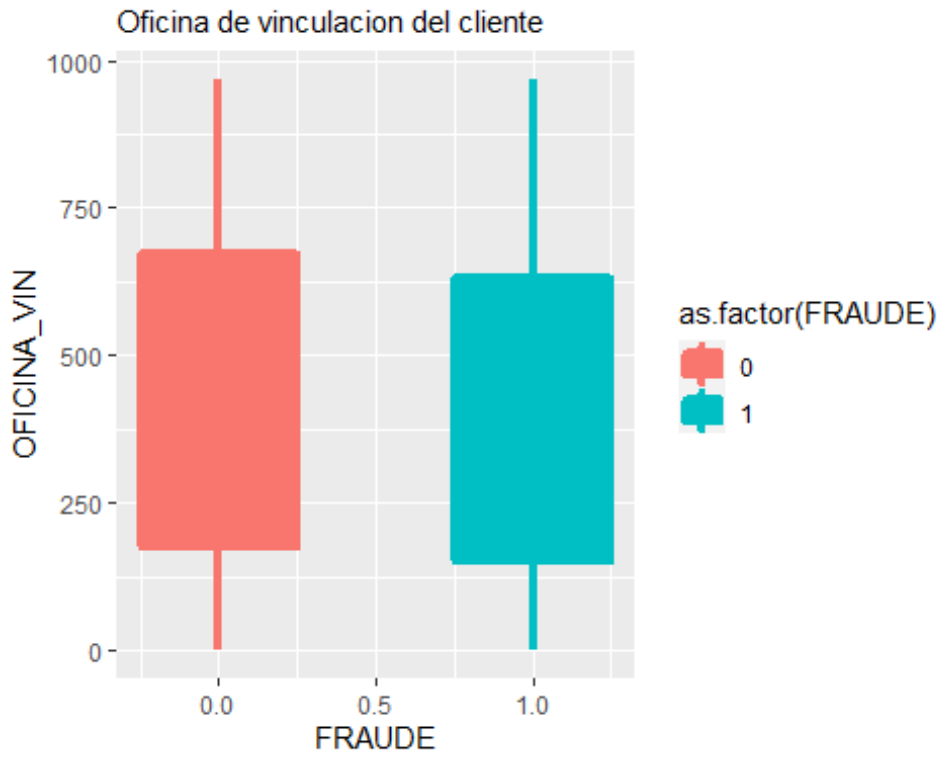

```
df_Fraude %>%
  ggplot(aes(x=FRAUDE, y=DIAMES, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  #geom_jitter(width=0.15)+
  labs(subtitle="Día del mes que se realizó la transacción")
```



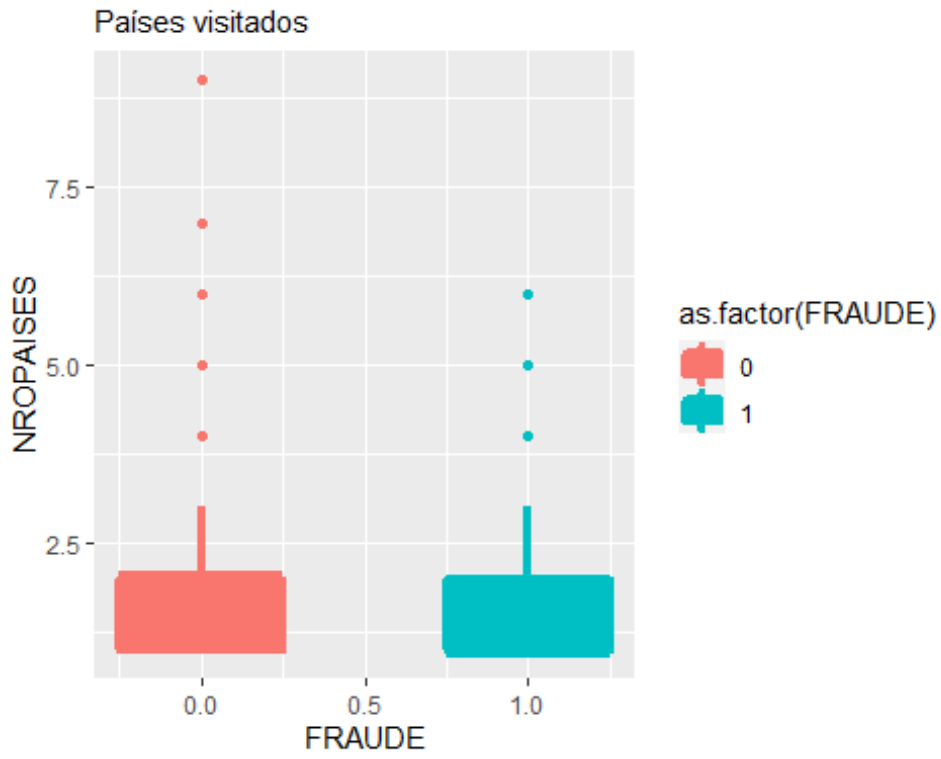

```
df_Fraude %>%
  ggplot(aes(x=FRAUDE, y=FECHA_VIN, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  #geom_jitter(width=0.15)+
  labs(subtitle="Fecha de vinculacion del cliente")

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



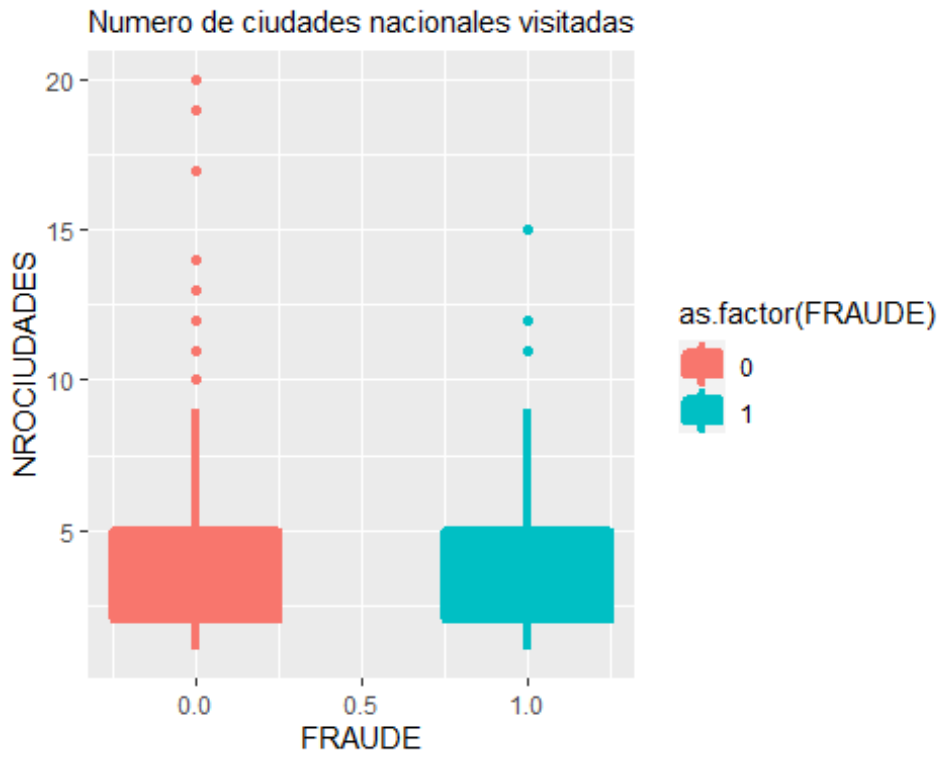

```
df_Fraude %>%
  ggplot(aes(x=FRAUDE, y=OFICINA_VIN, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  #geom_jitter(width=0.15)+
  labs(subtitle="Oficina de vinculacion del cliente")

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




```
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="Numero de ciudades nacionales visitadas")
```

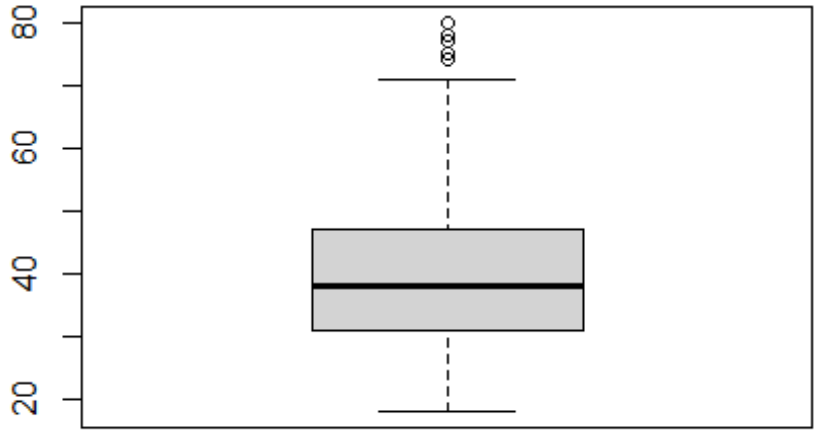


Estos gráficos son especialmente útiles como referencia a la hora de identificar posibles valores atípicos.

##Eliminación de valores atípicos

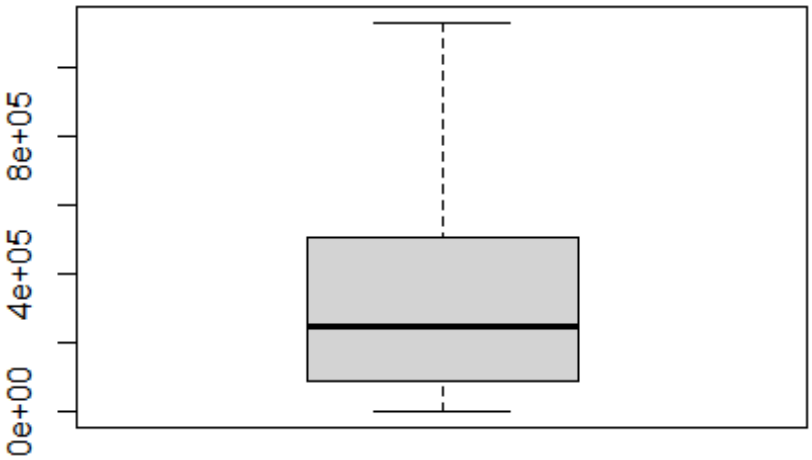
La variable EDAD incluye valores dificilmente creíbles, transacciones realizadas poar personas de 0 años o por personas que superan por mucho los 100 años. Consideraré esos valores como erroneos y los eliminaré de mi muestra.

```
df_Fraude$SEXO <- replace(df_Fraude$SEXO, df_Fraude$SEXO == "", NA) #Camb
io a NA los valores vacios

df_Fraude$EDAD <- replace( df_Fraude$EDAD, df_Fraude$EDAD == 0, mean(df_F
raude$EDAD))
df_Fraude$EDAD <- replace( df_Fraude$EDAD, df_Fraude$EDAD > 100, mean(df_
Fraude$EDAD))

boxplot(df_Fraude$EDAD)
```



Para el tratamiento de los valores atípicos de la variable VALOR, desconozco si la distribución de los valores se ajusta a la realidad, si los valores alejados de la media son realmente outliers. Pero como habíamos visto en el análisis exploratio inicial, es la variable con mayor dispersión de toda la muestra. Por lo tanto, he decidido utilizar el rango intercuartílico y eliminar los valores más alejados de la media para la realización del modelo.

```
quantile(df_Fraude$VALOR)

##          0%         25%         50%         75%        100% 
##        0.00    90160.01   243591.25   505819.01 20014064.66 

df_Fraude$VALOR <- replace(df_Fraude$VALOR, df_Fraude$VALOR >= as.numeric
( quantile(df_Fraude$VALOR)[4]) + 1.5*(quantile(df_Fraude$VALOR)[4]-quant
ile(df_Fraude$VALOR)[2]), as.numeric( quantile(df_Fraude$VALOR)[4]) + 1.5
*(quantile(df_Fraude$VALOR)[4]-quantile(df_Fraude$VALOR)[2]))

boxplot(df_Fraude$VALOR)
```

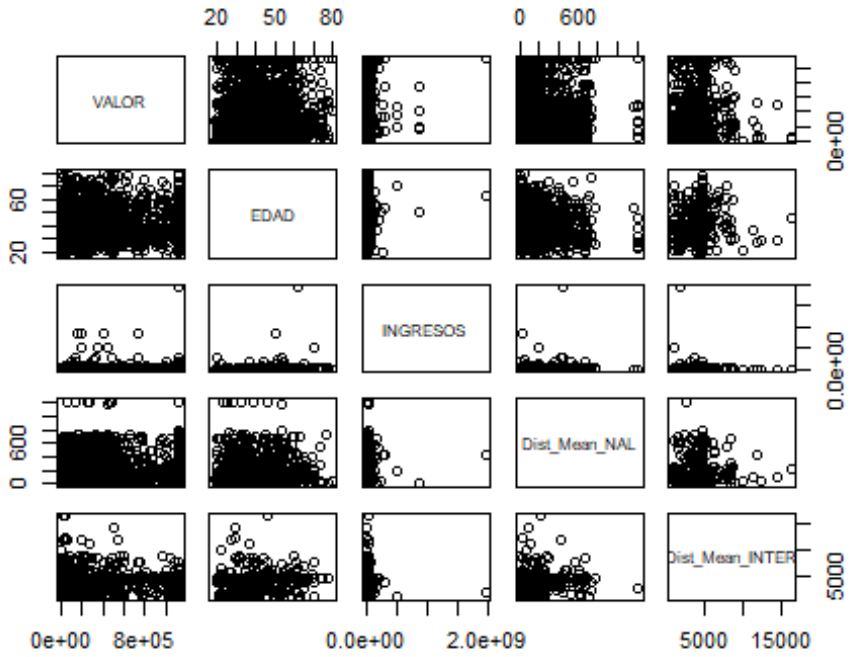


Podría realizar el mismo procedimiento con las variables con valores muy dispersos como INGRESOS o EGRESOS, pero al no conocer en profundidad si se tratan de valores con una alta probabilidad de ser erroneos, he decidido mantenerlos para el resto de variables por el momento.

Análisis de dependencia/independencia de variables

La primera opción para el estudio de las correlaciones entre las variables sería a través de el siguiente ejemplo de representación grafica:

```
#cor(df_Fraude)
pairs(~VALOR+EDAD+INGRESOS+Dist_Mean_NAL+Dist_Mean_INTER, data =df_Fraude
)
```



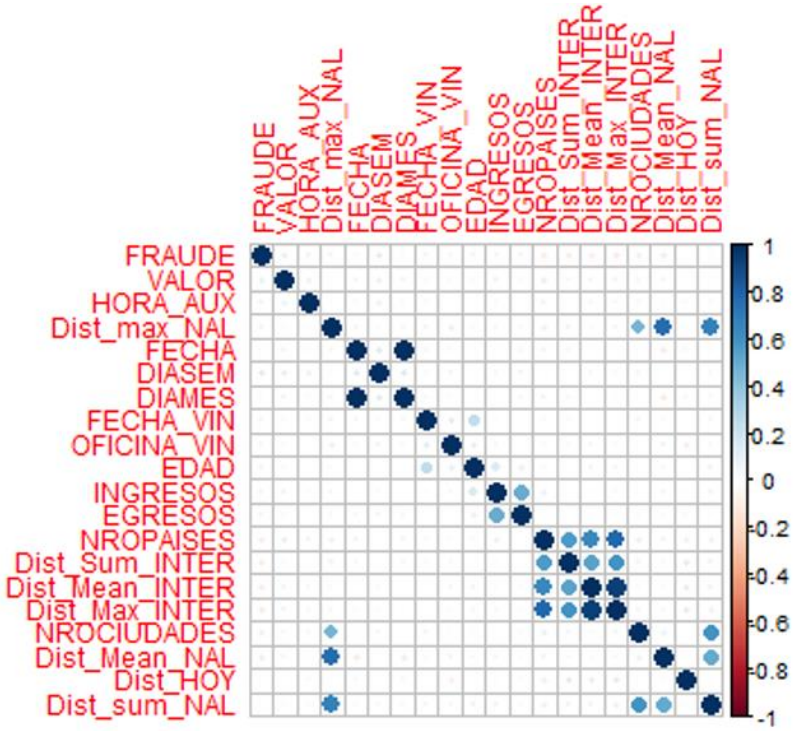
Sin embargo, este resulta poco explicativo.

He decidido generar una matriz de correlaciones en su lugar:

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

##      FRAUDE      VALOR HORA_AUX Dist_max_NAL    FECHA DIASEM DIAMES FE
CHA_VIN
## 1         1       0.00       13       659.13 20150501      5      1  2
0120306
## 2         1       0.00       17       594.77 20150515      5     15  2
0050415
```

```
## 3          1        0.00        13        659.13 20150501       5       1  2
0120306
```

```
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)
```

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)

 df_corr <- dplyr::select(df_Fraude, -Canal1, -COD_PAIS, -CANAL, -id, -SE
GMENTO,-Dist_Sum_INTER,-Dist_Max_INTER,-Dist_sum_NAL,-Dist_max_NAL,-DIAME
S)#elimino variables que considero que incurren en multicolinialidad.

#imputamos valores
df_impo <- missForest(df_corr)

##   missForest iteration 1 in progress...

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

## done!
##   missForest iteration 2 in progress...

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

## done!

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

##     NRMSE
## 0.5444478

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

##          FRAUDE           VALOR        HORA_AUX           FECHA
DIASEM
##    1.858222e-01    1.227283e+11    4.030481e+01    8.344166e+01    4.3
77654e+00
##       FECHA_VIN     OFICINA_VIN            SEXO            EDAD
INGRESOS
```

```
##    8.575551e+09    8.425289e+04    2.500518e-01    1.454834e+02    3.1
77928e+15
##         EGRESOS       NROPAISES Dist_Mean_INTER     NROCIUDADES   Dist
_Mean_NAL
##    3.818203e+15    1.086221e+00    3.221412e+06    7.562616e+00    3.6
87406e+04
##        Dist_HOY
##    3.167471e+06
```

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

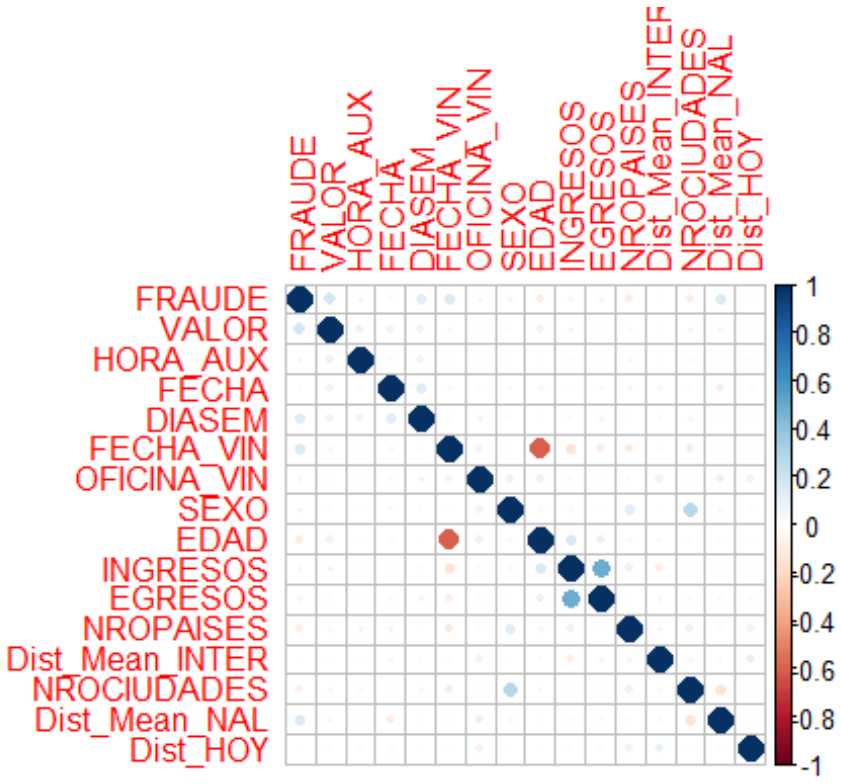
```
##          FRAUDE           VALOR        HORA_AUX           FECHA
DIASEM
##               0               0               0               0
0
##       FECHA_VIN     OFICINA_VIN            SEXO            EDAD
INGRESOS
##               0               0               0               0
0
##         EGRESOS       NROPAISES Dist_Mean_INTER     NROCIUDADES   Dist
_Mean_NAL
##               0               0               0               0
0
##        Dist_HOY
##               0
```

```
 df_corr <- df_impo$ximp
```

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

```
##          FRAUDE           VALOR        HORA_AUX           FECHA
DIASEM
##               0               0               0               0
0
##       FECHA_VIN     OFICINA_VIN            SEXO            EDAD
INGRESOS
##               0               0               0               0
0
##         EGRESOS       NROPAISES Dist_Mean_INTER     NROCIUDADES   Dist
_Mean_NAL
##               0               0               0               0
0
##        Dist_HOY
##               0
```

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



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, ENGRESOS E INGRESOS, NROCIUDADES Y SEXO, y Dist_Sum_INTER y NROPAISES.

Generación de modelos

Voy a proponer distintos modelos y comparar su acierto entre si.

Para empezar, 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)


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

##     NRMSE
## 0.5444478

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

##          FRAUDE           VALOR        HORA_AUX           FECHA
DIASEM
##    1.858222e-01    1.227283e+11    4.030481e+01    8.344166e+01    4.3
77654e+00
##       FECHA_VIN     OFICINA_VIN            SEXO            EDAD
INGRESOS
##    8.509965e+09    8.363659e+04    2.460647e-01    1.445542e+02    3.1
52249e+15
##         EGRESOS       NROPAISES Dist_Mean_INTER     NROCIUDADES   Dist
_Mean_NAL
##    3.788093e+15    1.086221e+00    1.620943e+06    7.562616e+00    3.2
12738e+04
##        Dist_HOY
##    3.167471e+06

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

##          FRAUDE           VALOR        HORA_AUX           FECHA
DIASEM
```

```
##               0               0               0               0
0
##       FECHA_VIN     OFICINA_VIN            SEXO            EDAD
INGRESOS
##               0               0               0               0
0
##         EGRESOS       NROPAISES Dist_Mean_INTER     NROCIUDADES   Dist
_Mean_NAL
##               0               0               0               0
0
##        Dist_HOY
##               0
```

```
 df_corr <- df_impo$ximp

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

```
##          FRAUDE           VALOR        HORA_AUX           FECHA
DIASEM
##               0               0               0               0
0
##       FECHA_VIN     OFICINA_VIN            SEXO            EDAD
INGRESOS
##               0               0               0               0
0
##         EGRESOS       NROPAISES Dist_Mean_INTER     NROCIUDADES   Dist
_Mean_NAL
##               0               0               0               0
0
##        Dist_HOY
##               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.

La idea es entrenar un modelo de regresión logística con regularización Ridge, Lasso, elastic net o stepAIC en train, seleccionando el que mejor AUC o metricas tenga.

Tendré en cuenta el criterio AUC, el área bajo la curva ROC como medida de acierto de la predicción de cada modelo, para los modelos Ridge, Lasso y elastic net. Además, también señalaré las medidas de accuracy y precisión para cada uno de los modelos propuestos.

##Aplico StepAIC

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

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

## 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

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

##                   Df Deviance    AIC
## + VALOR            1   2246.8 2250.8
## + FECHA_VIN        1   2277.6 2281.6
## + Dist_Mean_NAL    1   2289.7 2293.7
## + DIASEM           1   2296.3 2300.3
## + EDAD             1   2305.4 2309.4
## + EGRESOS          1   2310.2 2314.2
## + NROPAISES        1   2317.2 2321.2
## + NROCIUDADES      1   2319.8 2323.8
## + SEXO             1   2322.7 2326.7
## + OFICINA_VIN      1   2324.9 2328.9
## + INGRESOS         1   2325.9 2329.9
## + HORA_AUX         1   2329.3 2333.3
## + FECHA            1   2330.8 2334.8
## <none>                 2333.3 2335.3
## + Dist_HOY         1   2332.5 2336.5
## + Dist_Mean_INTER  1   2333.2 2337.2
##
## Step:  AIC=2250.79
## FRAUDE ~ VALOR

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

##                   Df Deviance    AIC
## + FECHA_VIN        1   2178.6 2184.6
## + Dist_Mean_NAL    1   2209.2 2215.2
## + EDAD             1   2209.3 2215.3
## + EGRESOS          1   2211.8 2217.8
## + DIASEM           1   2218.2 2224.2
## + NROPAISES        1   2226.3 2232.3
## + INGRESOS         1   2232.8 2238.8
## + SEXO             1   2234.1 2240.1
```

```
## + OFICINA_VIN      1   2236.1 2242.1
## + NROCIUDADES      1   2236.7 2242.7
## + HORA_AUX         1   2237.8 2243.8
## <none>                 2246.8 2250.8
## + FECHA            1   2246.1 2252.1
## + Dist_HOY         1   2246.3 2252.3
## + Dist_Mean_INTER  1   2246.6 2252.6
## - VALOR            1   2333.3 2335.3
##
## Step:  AIC=2184.6
## FRAUDE ~ VALOR + FECHA_VIN

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

##                    Df Deviance    AIC
## + Dist_Mean_NAL     1   2147.6 2155.6
## + DIASEM            1   2151.9 2159.9
## + EGRESOS           1   2160.6 2168.6
## + SEXO              1   2167.0 2175.0
## + NROPAISES         1   2167.8 2175.8
## + HORA_AUX          1   2168.6 2176.6
## + OFICINA_VIN       1   2171.9 2179.9
## + NROCIUDADES       1   2173.8 2181.8
## + INGRESOS          1   2174.4 2182.4
## <none>                  2178.6 2184.6
## + EDAD              1   2177.1 2185.1
## + Dist_HOY          1   2178.1 2186.1
## + Dist_Mean_INTER   1   2178.3 2186.3
## + FECHA             1   2178.4 2186.4
## - FECHA_VIN         1   2246.8 2250.8
## - VALOR             1   2277.6 2281.6
##
## Step:  AIC=2155.58
## FRAUDE ~ VALOR + FECHA_VIN + Dist_Mean_NAL

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

##                    Df Deviance    AIC
## + DIASEM            1   2121.7 2131.7
## + EGRESOS           1   2128.3 2138.3
## + NROPAISES         1   2135.4 2145.4
## + HORA_AUX          1   2136.5 2146.5
## + SEXO              1   2138.0 2148.0
## + OFICINA_VIN       1   2139.1 2149.1
## + INGRESOS          1   2141.6 2151.6
## + NROCIUDADES       1   2145.4 2155.4
## <none>                  2147.6 2155.6
## + EDAD              1   2145.9 2155.9
## + FECHA             1   2146.4 2156.4
## + Dist_Mean_INTER   1   2146.9 2156.9
## + Dist_HOY          1   2147.0 2157.0
```

```
## - Dist_Mean_NAL    1   2178.6 2184.6
## - FECHA_VIN        1   2209.2 2215.2
## - VALOR            1   2238.9 2244.9
##
## Step:  AIC=2131.68
## FRAUDE ~ VALOR + FECHA_VIN + Dist_Mean_NAL + DIASEM

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

##                   Df Deviance    AIC
## + EGRESOS          1   2102.9 2114.9
## + HORA_AUX         1   2107.7 2119.7
## + NROPAISES        1   2109.2 2121.2
## + OFICINA_VIN      1   2109.8 2121.8
## + SEXO             1   2112.3 2124.3
## + INGRESOS         1   2115.0 2127.0
## + NROCIUDADES      1   2119.5 2131.5
## <none>                 2121.7 2131.7
## + EDAD             1   2119.7 2131.7
## + Dist_Mean_INTER  1   2120.9 2132.9
## + Dist_HOY         1   2121.1 2133.1
## + FECHA            1   2121.4 2133.4
## - DIASEM           1   2147.6 2155.6
## - Dist_Mean_NAL    1   2151.9 2159.9
## - FECHA_VIN        1   2181.0 2189.0
## - VALOR            1   2205.7 2213.7

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

##
## Step:  AIC=2114.9
## FRAUDE ~ VALOR + FECHA_VIN + Dist_Mean_NAL + DIASEM + EGRESOS

## 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

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

##                    Df Deviance    AIC
## + HORA_AUX          1   2089.1 2103.1
## + OFICINA_VIN       1   2091.6 2105.6
## + NROPAISES         1   2092.3 2106.3
## + SEXO              1   2094.1 2108.1
## + NROCIUDADES       1   2100.7 2114.7
## <none>                  2102.9 2114.9
## + Dist_Mean_INTER   1   2101.1 2115.1
## + EDAD              1   2101.9 2115.9
## + Dist_HOY          1   2102.4 2116.4
## + FECHA             1   2102.6 2116.6
## + INGRESOS          1   2102.7 2116.7
## - EGRESOS           1   2121.7 2131.7
## - DIASEM            1   2128.3 2138.3
## - Dist_Mean_NAL     1   2134.4 2144.4
## - FECHA_VIN         1   2147.5 2157.5
## - VALOR             1   2193.4 2203.4

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

##
## Step:  AIC=2103.14
## FRAUDE ~ VALOR + FECHA_VIN + Dist_Mean_NAL + DIASEM + EGRESOS +
##     HORA_AUX

## 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

##                    Df Deviance    AIC
## + OFICINA_VIN       1   2076.7 2092.7
## + NROPAISES         1   2079.3 2095.3
## + SEXO              1   2079.9 2095.9
## + NROCIUDADES       1   2086.5 2102.5
## <none>                  2089.1 2103.1
## + Dist_Mean_INTER   1   2087.4 2103.4
## + EDAD              1   2088.1 2104.1
## + Dist_HOY          1   2088.7 2104.7
## + INGRESOS          1   2088.9 2104.9
## + FECHA             1   2088.9 2104.9
## - HORA_AUX          1   2102.9 2114.9
## - EGRESOS           1   2107.7 2119.7
## - DIASEM            1   2117.6 2129.6
## - Dist_Mean_NAL     1   2121.6 2133.6
## - FECHA_VIN         1   2135.3 2147.3
## - VALOR             1   2186.1 2198.1
##
## Step:  AIC=2092.69
## FRAUDE ~ VALOR + FECHA_VIN + Dist_Mean_NAL + DIASEM + EGRESOS +
##     HORA_AUX + OFICINA_VIN

## 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

##                    Df Deviance    AIC
## + NROPAISES         1   2067.7 2085.7
## + SEXO              1   2068.6 2086.6
## + Dist_Mean_INTER   1   2073.8 2091.8
## + NROCIUDADES       1   2074.0 2092.0
## <none>                  2076.7 2092.7
## + EDAD              1   2075.8 2093.8
## + INGRESOS          1   2076.3 2094.3
## + FECHA             1   2076.6 2094.6
## + Dist_HOY          1   2076.6 2094.6
## - OFICINA_VIN       1   2089.1 2103.1
## - HORA_AUX          1   2091.6 2105.6
## - EGRESOS           1   2094.5 2108.5
## - DIASEM            1   2108.8 2122.8
```

```
## - Dist_Mean_NAL    1   2111.8 2125.8
## - FECHA_VIN        1   2119.1 2133.1
## - VALOR            1   2175.2 2189.2
##
## Step:  AIC=2085.67
## FRAUDE ~ VALOR + FECHA_VIN + Dist_Mean_NAL + DIASEM + EGRESOS +
##     HORA_AUX + OFICINA_VIN + NROPAISES

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

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

##                    Df Deviance    AIC
## + SEXO              1   2061.1 2081.1
## + Dist_Mean_INTER   1   2064.0 2084.0
## + NROCIUDADES       1   2065.5 2085.5
## <none>                  2067.7 2085.7
## + EDAD              1   2066.6 2086.6
## + INGRESOS          1   2066.9 2086.9
## + Dist_HOY          1   2067.3 2087.3
## + FECHA             1   2067.4 2087.4
## - NROPAISES         1   2076.7 2092.7
## - OFICINA_VIN       1   2079.3 2095.3
## - HORA_AUX          1   2081.7 2097.7
## - EGRESOS           1   2083.9 2099.9
## - DIASEM            1   2099.9 2115.9
## - Dist_Mean_NAL     1   2103.2 2119.2
## - FECHA_VIN         1   2105.4 2121.4
## - VALOR             1   2166.8 2182.8
##
## Step:  AIC=2081.14
## FRAUDE ~ VALOR + FECHA_VIN + Dist_Mean_NAL + DIASEM + EGRESOS +
##     HORA_AUX + OFICINA_VIN + NROPAISES + SEXO

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

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

##                    Df Deviance    AIC
## + Dist_Mean_INTER   1   2057.3 2079.3
## <none>                  2061.1 2081.1
## + INGRESOS          1   2060.2 2082.2
## + EDAD              1   2060.4 2082.4
## + NROCIUDADES       1   2060.5 2082.5
## + FECHA             1   2060.7 2082.7
## + Dist_HOY          1   2060.8 2082.8
## - SEXO              1   2067.7 2085.7
## - NROPAISES         1   2068.6 2086.6
## - OFICINA_VIN       1   2071.8 2089.8
## - HORA_AUX          1   2075.6 2093.6
## - EGRESOS           1   2077.2 2095.2
```

```
## - DIASEM           1   2093.1 2111.1
## - Dist_Mean_NAL    1   2094.8 2112.8
## - FECHA_VIN        1   2099.7 2117.7
## - VALOR            1   2161.6 2179.6
##
## Step:  AIC=2079.3
## FRAUDE ~ VALOR + FECHA_VIN + Dist_Mean_NAL + DIASEM + EGRESOS +
##     HORA_AUX + OFICINA_VIN + NROPAISES + SEXO + Dist_Mean_INTER

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

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

##                   Df Deviance    AIC
## <none>                 2057.3 2079.3
## + NROCIUDADES      1   2056.4 2080.4
## + INGRESOS         1   2056.5 2080.5
## + EDAD             1   2056.8 2080.8
## + Dist_HOY         1   2056.9 2080.9
## + FECHA            1   2057.1 2081.1
## - Dist_Mean_INTER  1   2061.1 2081.1
## - SEXO             1   2064.0 2084.0
## - NROPAISES        1   2065.5 2085.5
## - OFICINA_VIN      1   2069.3 2089.3
## - HORA_AUX         1   2071.8 2091.8
## - EGRESOS          1   2074.8 2094.8
## - DIASEM           1   2089.9 2109.9
## - Dist_Mean_NAL    1   2092.2 2112.2
## - FECHA_VIN        1   2094.6 2114.6
## - VALOR            1   2158.6 2178.6

summary(step)

##
## Call:
## glm(formula = FRAUDE ~ VALOR + FECHA_VIN + Dist_Mean_NAL + DIASEM +
##     EGRESOS + HORA_AUX + OFICINA_VIN + NROPAISES + SEXO + Dist_Mean_IN
TER,
##     family = binomial, data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.84562  -0.74520  -0.53600  -0.00066   2.63361
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -8.100e+01  1.377e+01  -5.883 4.02e-09 ***
## VALOR            1.510e-06  1.511e-07   9.993  < 2e-16 ***
## FECHA_VIN        4.016e-06  6.863e-07   5.851 4.88e-09 ***
## Dist_Mean_NAL    1.724e-03  2.927e-04   5.890 3.86e-09 ***
## DIASEM           1.538e-01  2.734e-02   5.625 1.86e-08 ***
```

```
## EGRESOS         -1.642e-08  6.238e-09  -2.632  0.00850 **
## HORA_AUX        -3.274e-02  8.555e-03  -3.827  0.00013 ***
## OFICINA_VIN     -6.774e-04  1.974e-04  -3.432  0.00060 ***
## NROPAISES       -1.638e-01  5.870e-02  -2.791  0.00526 **
## SEXO            -2.877e-01  1.114e-01  -2.584  0.00978 **
## Dist_Mean_INTER -1.045e-04  5.427e-05  -1.925  0.05426 .
## ---
## 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: 2057.3  on 2070  degrees of freedom
## AIC: 2079.3
##
## Number of Fisher Scoring iterations: 7
```

En este caso, el modelo propuesto sería el siguiente: FRAUDE ~ VALOR + FECHA_VIN + Dist_Mean_NAL + DIASEM + EGRESOS + HORA_AUX + OFICINA_VIN + NROPAISES + SEXO + Dist_Mean_INTER

El modelo con menor AIC de los propuestos.

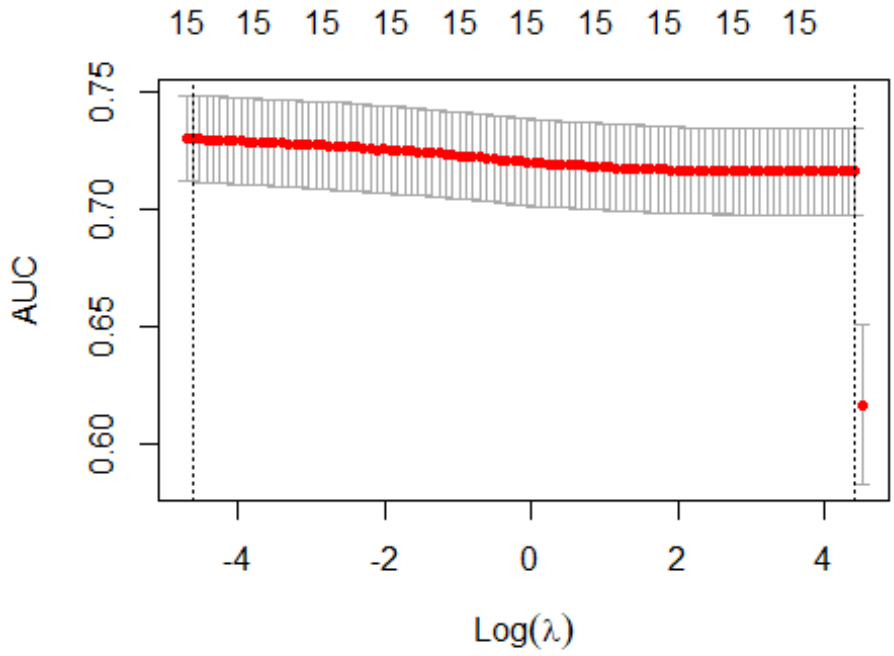
Damos métricas:

```
y_pred <- as.numeric(predict(step, y_test, type="response"))

## Error in model.frame.default(Terms, newdata, na.action = na.action, xl
ev = object$xlevels): 'data' must be a data.frame, environment, or list

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

table(no_fraud,y_test)

## Error in table(no_fraud, y_test): objeto 'no_fraud' no encontrado
```

Tendríamos un Accuracy del 0.765 y una precisión del 0.98

##Aplico Ridge:

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

plot(cv.ridge)
```




```
cv.ridge$lambda.min

## [1] 0.009953723

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

## [1] 0.7300243
```

Observamos que el modelo regularizado de Ridge con λ óptimo cuenta con AUC de 0.8713678

Vemos los coeficientes

```
coef(cv.ridge, s=cv.ridge$lambda.min)

## 17 x 1 sparse Matrix of class "dgCMatrix"
##                               s1
## (Intercept)     -5.718237e+04
## VALOR            1.370420e-06
```

```
## HORA_AUX         -2.982954e-02
## FECHA             2.834438e-03
## DIASEM            1.432893e-01
## FECHA_VIN         3.321799e-06
## OFICINA_VIN      -6.059492e-04
## SEXO             -2.302732e-01
## EDAD             -6.328131e-03
## INGRESOS         -2.377064e-10
## EGRESOS          -6.114515e-09
## NROPAISES        -1.637599e-01
## Dist_Mean_INTER -8.603256e-05
## NROCIUDADES      -2.477723e-02
## Dist_Mean_NAL     1.591696e-03
## Dist_HOY          1.628803e-05
## random            .
```

Damos métricas en el test

```
y_pred <- as.numeric(predict.glmnet(cv.ridge$glmnet.fit, newx=x_test, s=c
v.ridge$lambda.min)>.5)
y_pred <- as.factor(y_pred)
y_test <- as.factor(y_test)


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

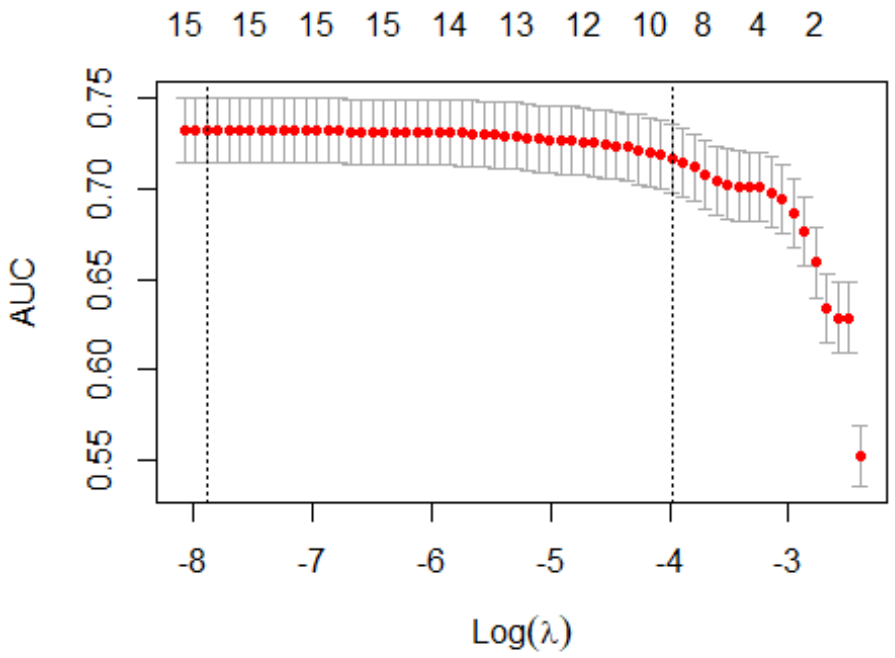
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 665   5
##          1 203  11
##
##                Accuracy : 0.7647
##                  95% CI : (0.7353, 0.7923)
##     No Information Rate : 0.9819
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0641
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7661
##             Specificity : 0.6875
##          Pos Pred Value : 0.9925
##          Neg Pred Value : 0.0514
##               Precision : 0.9925
##                  Recall : 0.7661
##                      F1 : 0.8648
##              Prevalence : 0.9819
```

```
##          Detection Rate : 0.7523
##    Detection Prevalence : 0.7579
##       Balanced Accuracy : 0.7268
##
##        'Positive' Class : 0
##
```

Ridge proporciona un modelo con un accuracy de 0.7647 y una precisión del 0.9925.

##Aplico Lasso

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

plot(cv.lasso)
```




```
cv.lasso$lambda.min

## [1] 0.0003747513

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

## [1] 0.7328629
```

Observamos que el modelo regularizado de Lasso con λ óptimo cuenta con AUC de 0.8710428

Vemos los coeficientes

```
coef(cv.lasso, s=cv.lasso$lambda.min)

## 17 x 1 sparse Matrix of class "dgCMatrix"
##                               s1
## (Intercept)     -5.185860e+04
## VALOR            1.482043e-06
```

```
## HORA_AUX         -3.247399e-02
## FECHA             2.570068e-03
## DIASEM            1.512802e-01
## FECHA_VIN         3.494246e-06
## OFICINA_VIN      -6.547189e-04
## SEXO             -2.467645e-01
## EDAD             -5.177498e-03
## INGRESOS          1.157295e-09
## EGRESOS          -1.703294e-08
## NROPAISES        -1.707426e-01
## Dist_Mean_INTER -9.977521e-05
## NROCIUDADES      -2.391222e-02
## Dist_Mean_NAL     1.677767e-03
## Dist_HOY          1.638266e-05
## random            .
```

Hace una selección bastante exhaustiva de los mismos, poniendo en valor una de las características principales de las regularizaciones Lasso: la obtención de modelos sparse o huecos.

```
y_pred <- as.numeric(predict.glmnet(cv.lasso$glmnet.fit, newx=x_test, s=c
v.lasso$lambda.min)>.5)
y_pred <- as.factor(y_pred)
y_test <- as.factor(y_test)

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

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 663   7
##          1 200  14
##
##                Accuracy : 0.7658
##                  95% CI : (0.7365, 0.7934)
##     No Information Rate : 0.9762
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0793
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.76825
##             Specificity : 0.66667
##          Pos Pred Value : 0.98955
##          Neg Pred Value : 0.06542
##               Precision : 0.98955
##                  Recall : 0.76825
##                      F1 : 0.86497
```

```
##              Prevalence : 0.97624
##          Detection Rate : 0.75000
##    Detection Prevalence : 0.75792
##       Balanced Accuracy : 0.71746
##
##        'Positive' Class : 0
##
```

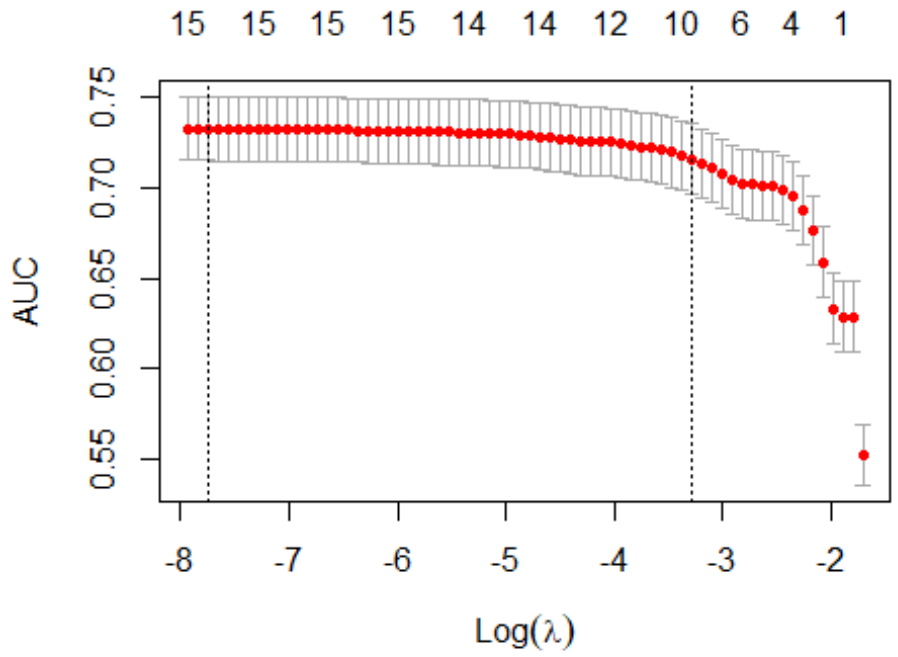
En este caso el accuracy es del 0.7658 y la precisión del modelo del 0.98955.

Además, el AUC del modelo de Ridge es mayor al de Lasso. Por lo tanto preferiremos Ridge antes que Lasso.

Aplico 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, t
ype.measure='auc')
# Resultados
plot(cv.elastic)
```




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

## [1] 0.0004288929

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

## [1] 0.7328303
```

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)
```

```r
y_pred <- as.factor(y_pred)
y_test <- as.factor(y_test)

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

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 663   7
##          1 200  14
##
##                Accuracy : 0.7658
##                  95% CI : (0.7365, 0.7934)
##     No Information Rate : 0.9762
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0793
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.76825
##             Specificity : 0.66667
##          Pos Pred Value : 0.98955
##          Neg Pred Value : 0.06542
##               Precision : 0.98955
##                  Recall : 0.76825
##                      F1 : 0.86497
##              Prevalence : 0.97624
##          Detection Rate : 0.75000
##    Detection Prevalence : 0.75792
##       Balanced Accuracy : 0.71746
##
##        'Positive' Class : 0
##
```

Para este modelo el accuracy sería del 0.7658 y la precisión del 0.98955. Mejoraría levemente el accuracy y el AUC, perjudincando la precisión del modelo.

##Modelo propio

Por último, de manera adicional, voy a tratar de crear mi propio modelo de otra manera. Primero voy a generar un modelo con todas las variables excepto “id”.

```r
#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: 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
## -1.83789  -0.74660  -0.52904  -0.00033   2.66765
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -5.630e+04  1.206e+05  -0.467  0.64068
## VALOR            1.495e-06  1.519e-07   9.842  < 2e-16 ***
## HORA_AUX        -3.299e-02  8.576e-03  -3.847  0.00012 ***
## FECHA            2.790e-03  5.986e-03   0.466  0.64110
## DIASEM           1.525e-01  2.752e-02   5.540 3.03e-08 ***
## FECHA_VIN        3.473e-06  8.705e-07   3.990 6.61e-05 ***
## OFICINA_VIN     -6.656e-04  1.987e-04  -3.349  0.00081 ***
## SEXO            -2.498e-01  1.164e-01  -2.145  0.03195 *
## EDAD            -5.326e-03  6.082e-03  -0.876  0.38121
## INGRESOS         1.505e-09  1.342e-09   1.121  0.26217
## EGRESOS         -1.970e-08  7.520e-09  -2.620  0.00879 **
## NROPAISES       -1.739e-01  5.958e-02  -2.919  0.00352 **
## Dist_Mean_INTER -1.040e-04  5.547e-05  -1.875  0.06074 .
## NROCIUDADES     -2.494e-02  2.316e-02  -1.077  0.28146
## Dist_Mean_NAL    1.689e-03  2.968e-04   5.693 1.25e-08 ***
## Dist_HOY         1.758e-05  2.761e-05   0.637  0.52422
## ---
## 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: 2054.1  on 2065  degrees of freedom
## AIC: 2086.1
##
## Number of Fisher Scoring iterations: 7
```

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

```
a_mantener <- c("VALOR","HORA_AUX","DIASEM","FECHA_VIN","OFICINA_VIN","Di
st_Mean_NAL") #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
## -1.7162  -0.7511  -0.5592  -0.2663   2.5099
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -9.766e+01  1.347e+01  -7.249 4.21e-13 ***
## VALOR          1.407e-06  1.473e-07   9.548  < 2e-16 ***
## HORA_AUX      -3.318e-02  8.442e-03  -3.930 8.50e-05 ***
## DIASEM         1.525e-01  2.704e-02   5.638 1.72e-08 ***
## FECHA_VIN      4.790e-06  6.725e-07   7.123 1.05e-12 ***
## OFICINA_VIN   -6.969e-04  1.933e-04  -3.605 0.000313 ***
## Dist_Mean_NAL  1.671e-03  2.876e-04   5.812 6.18e-09 ***
## ---
## 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: 2094.5  on 2074  degrees of freedom
## AIC: 2108.5
##
## Number of Fisher Scoring iterations: 4
```

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

```
## Start:  AIC=2108.52
## FRAUDE ~ VALOR + HORA_AUX + DIASEM + FECHA_VIN + OFICINA_VIN +
##     Dist_Mean_NAL
##
##                 Df Deviance    AIC
## <none>               2094.5 2108.5
## - OFICINA_VIN    1   2107.7 2119.7
## - HORA_AUX       1   2109.8 2121.8
```

```
## - DIASEM         1   2127.3 2139.3
## - Dist_Mean_NAL  1   2128.3 2140.3
## - FECHA_VIN      1   2150.6 2162.6
## - VALOR          1   2186.3 2198.3

summary(step)

##
## Call:
## glm(formula = FRAUDE ~ VALOR + HORA_AUX + DIASEM + FECHA_VIN +
##     OFICINA_VIN + Dist_Mean_NAL, family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7162  -0.7511  -0.5592  -0.2663   2.5099
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -9.766e+01  1.347e+01  -7.249 4.21e-13 ***
## VALOR          1.407e-06  1.473e-07   9.548  < 2e-16 ***
## HORA_AUX      -3.318e-02  8.442e-03  -3.930 8.50e-05 ***
## DIASEM         1.525e-01  2.704e-02   5.638 1.72e-08 ***
## FECHA_VIN      4.790e-06  6.725e-07   7.123 1.05e-12 ***
## OFICINA_VIN   -6.969e-04  1.933e-04  -3.605 0.000313 ***
## Dist_Mean_NAL  1.671e-03  2.876e-04   5.812 6.18e-09 ***
## ---
## 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: 2094.5  on 2074  degrees of freedom
## AIC: 2108.5
##
## Number of Fisher Scoring iterations: 4
```

Veo que el criterio AIC no ha mejorado si no que ha empeorado.

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] 0.1023243
```

En este cado no he logrado mejorar los modelos propuestos anteriormente.

###Conclusiones

Antes de generar los distintos modelos he relaizado un pequeño anális exploratorio de la muestra de datos, limpiado mi dataset de posible multicolinealidad y de valores atípicos. Una vez hecho esto, he generado distintos modelos y he estudiado su eficacia a través de la partición de mi dataset test.

Todos los modelos propuestos prentan un accuracy relativamente alto, entorno al 0.76. Si tubieramos que decantarnos por uno de los modelos sería por del de elactic net, al tener un AUC (acierto en la predicción), un accuracy y un recall, ligeramente superior al de Ridge y Lasso, aunque su precisión si sea un poco más pequeña que la del modelo Ridge.

En cuanto al modelo generado a través de stepwise, su desempeño era parecido pero no mejora el modelo elastic net.