

Modelo GLM de regresión logística

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.

La variable FRAUDE 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.

DICCIONARIO DE LAS VARIABLES

ID Id Cliente

FRAUDE 1= Fraude; 0=No fraude

VALOR Valor de la transacción

HORA_AUX Hora de la transacción, sin minutos ni segundos

Dist_max_NAL Dist maxima recorrida a nivel nacional (en millas)

Canal1 Canal transaccional de la transacción, incluido tipos de datafonos

FECHA Fecha de ocurrencia de la transacción

COD_PAIS Pais de ocurrencia de la transacción. Ver codigo ISO Internet

CANAL Canal transaccional de la transacción

DIASEM Día de la semana que se realizó la transacción

DIAMES Día del mes que se realizó la transacción (0= Domingo, 1= Lunes, 2= martes,... 3= miercoles...6= sábado)

FECHA_VIN Fecha de vinculacion del cliente

OFICINA_VIN Oficina de vinculacion del cliente

SEXO M=masculino, F= femenino

SEGMENTO Segmento del cliente

EDAD Edad del cliente

INGRESOS Ingresos del cliente

EGRESOS Egresos del cliente

NROPAISES # paises visitados

Dist_Sum_INTER Sumatoria de distancia recorrida a nivel internacional (en millas)

Dist_Mean_INTER Promedio de distancia recorrida a nivel internacional (en millas)

NROCIUDADES Numero de ciudades nacionales visitadas

Dist_Sum_NAL Distancia máxima recorrida a nivel nacional (en millas)

Dist_Mean_NAL Distancia máxima recorrida a nivel nacional (en millas)

Dist_HOY Diferencia entre la ultima transacción presente realizada y la transacción que esta realizando el dia de hoy

Dist_sum_NAL Sumatoria de distacia recorrida a nivel nacional (en millas)

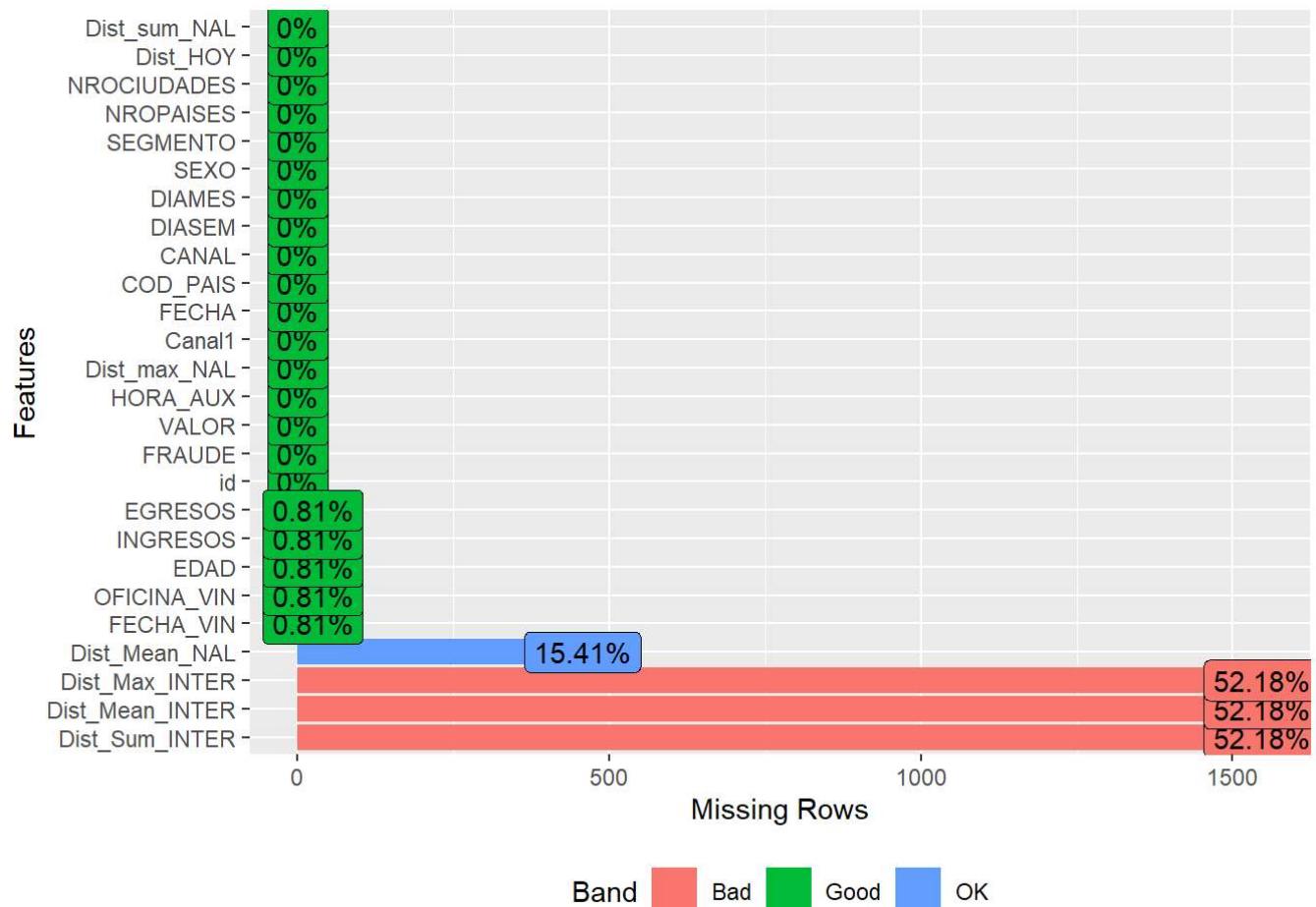
LIBRERIAS

```
library(tidyverse)
library(DataExplorer)
library(ggplot2)
library(kableExtra)
library(gridExtra)
library(missForest)
library(corrplot)
library(MASS)
library(pROC)
library(glmnet)
library(caret)
library(lattice)
library(e1071)
```

```
datos<-read.csv("Fraude.csv")
```

Análisis exploratorio

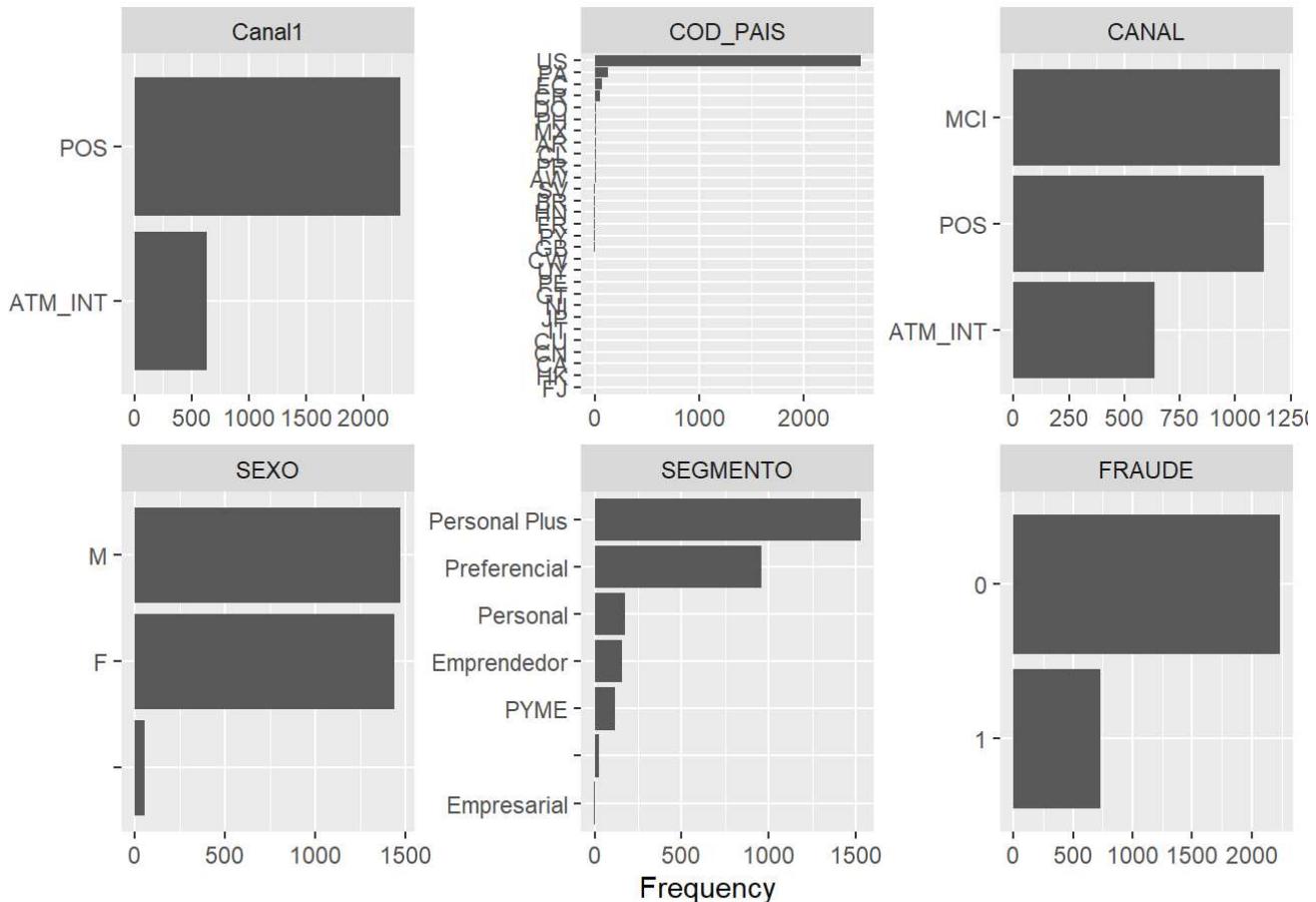
```
#Analisis de valores faltantes
datos %>% plot_missing()
```



Existe un gran número de valores faltantes en las variables “Dist_”, variables que miden de diferentes formas la distancia recorrida por el usuario de la tarjeta. Tendré esto en cuenta a la hora de descartar variables para el modelo.

```
#Análisis preliminar de Las variables categoricas:
```

```
datos %>% plot_bar()
```

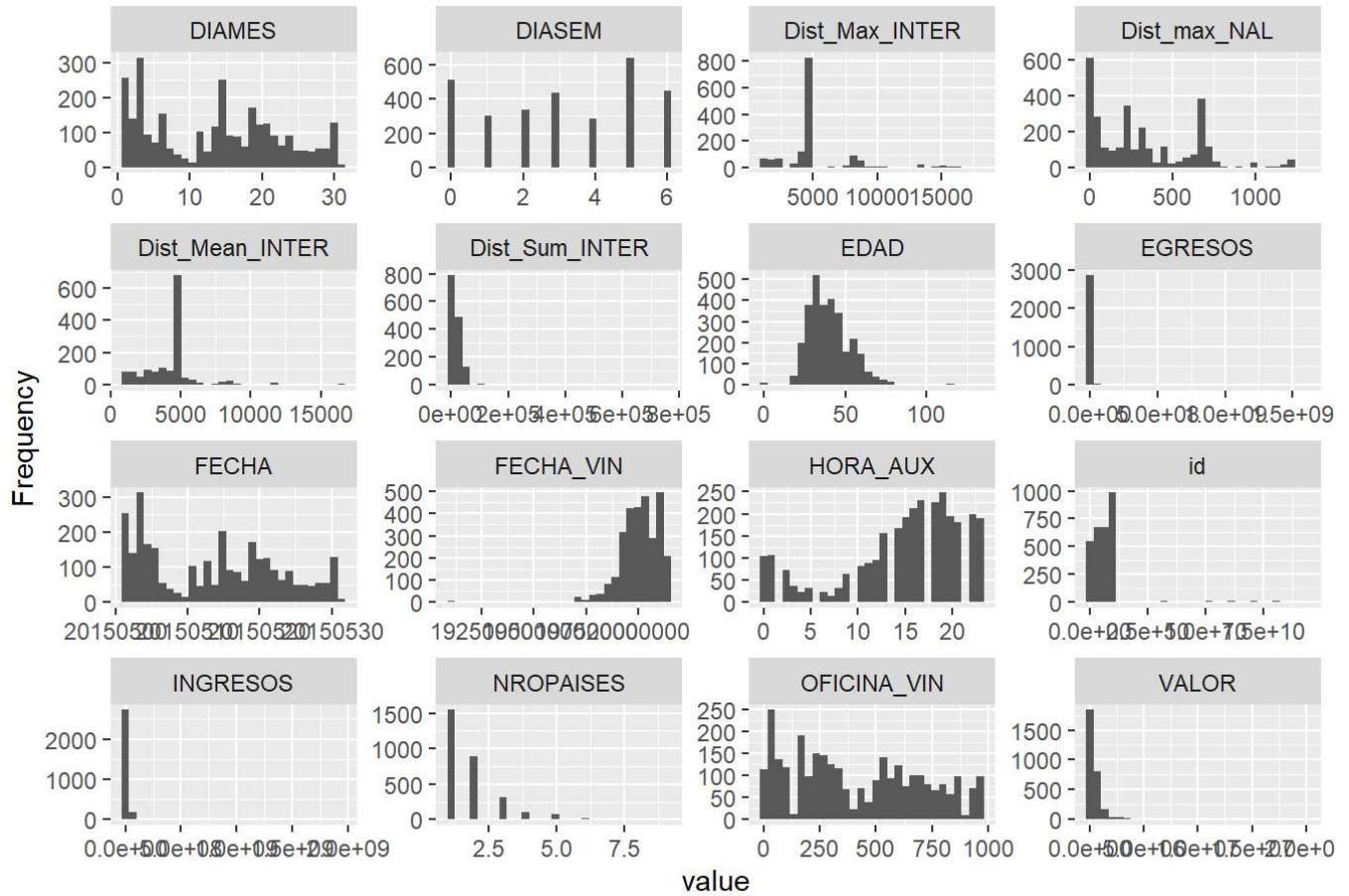


En un primer vistazo se observa cómo, en la variable COD_PAIS, la distribución de las operaciones entre los distintos países es muy desigual. En este sentido, la mayoría de las transacciones ocurren en Estados Unidos.

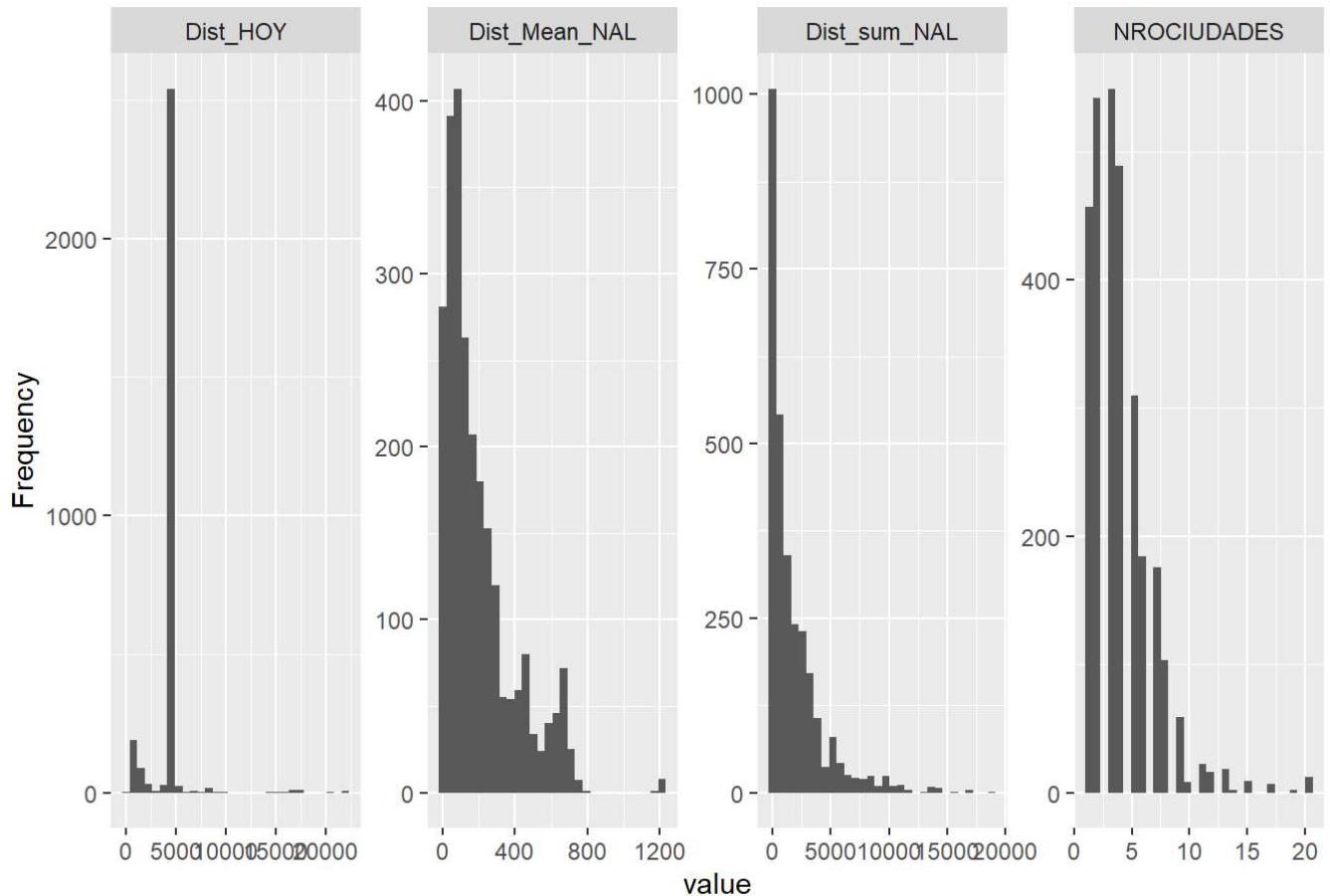
Esta combinación de gráficos permite tener una visión general. Me centrare en analizar cada una de las variables más adelante.

```
#Análisis preliminar de Las variables numéricas:
```

```
datos %>% plot_histogram()
```



Page 1



Page 2

De nuevo, con un primer vistazo se puede pensar en la existencia de posibles valores atípicos en algunas de las variables numéricas, como por ejemplo: variables EDAD o FECHA_VIN.

A continuación me centraré en estudiar cada una de las variables por separado. La variable FRAUDE es la variable target, es la variable más importante para el modelo, por lo tanto se analizará la relación de cada una de las variables con esta en específico.

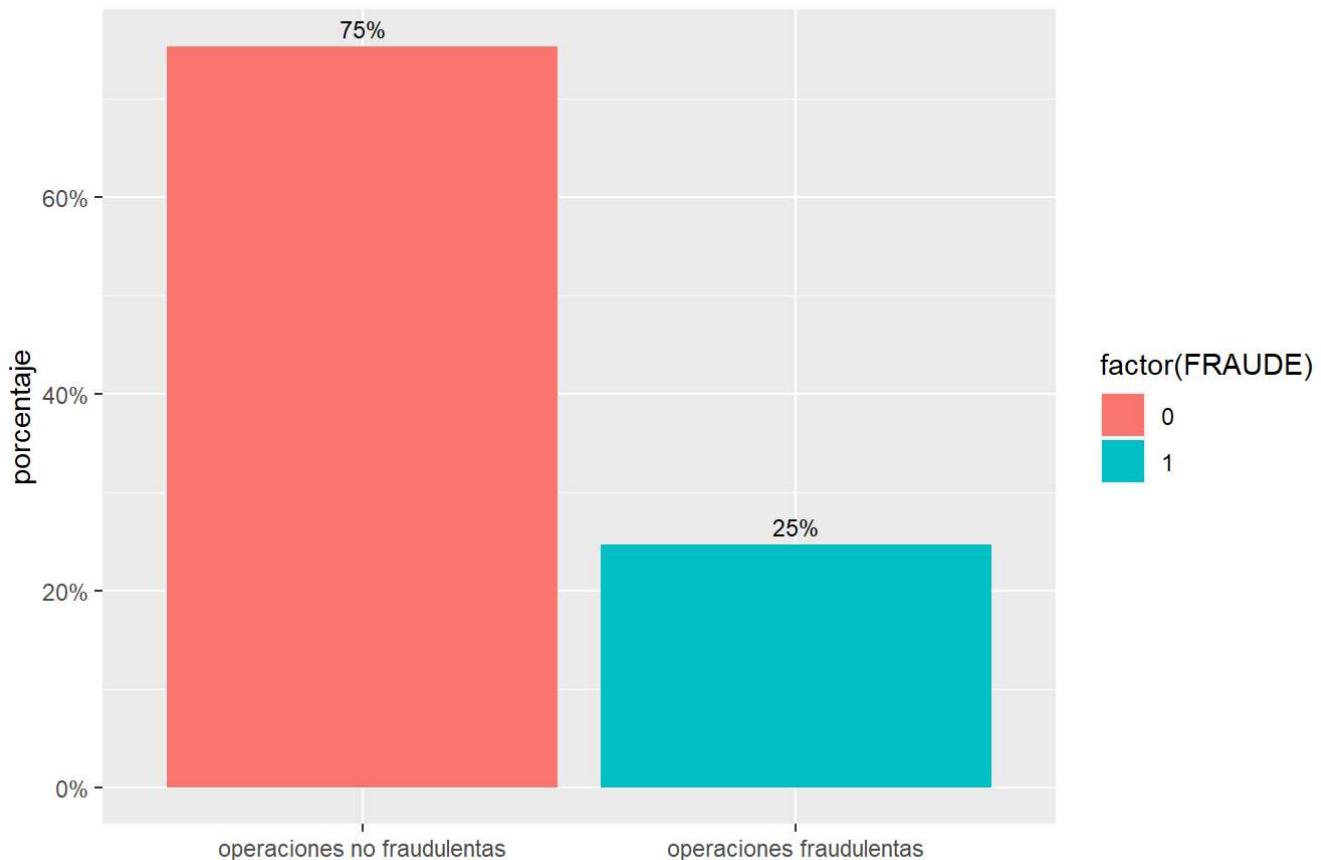
La variable id es un identificador único de cada titular de una tarjeta por lo tanto no se tendrá en cuenta.

#Variable Target

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

ggplot(data = datos, 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 = '', y = 'porcentaje') +
  ggtitle("casos de fraude sobre el total de las operaciones") +
  common_theme
```

casos de fraude sobre el total de las operaciones



```
datos %>% group_by(FRAUDE) %>% summarise(porcentaje_operaciones=round(n()*100/nrow(datos),2))
```

FRAUDE
<int>

porcentaje_operaciones
<dbl>

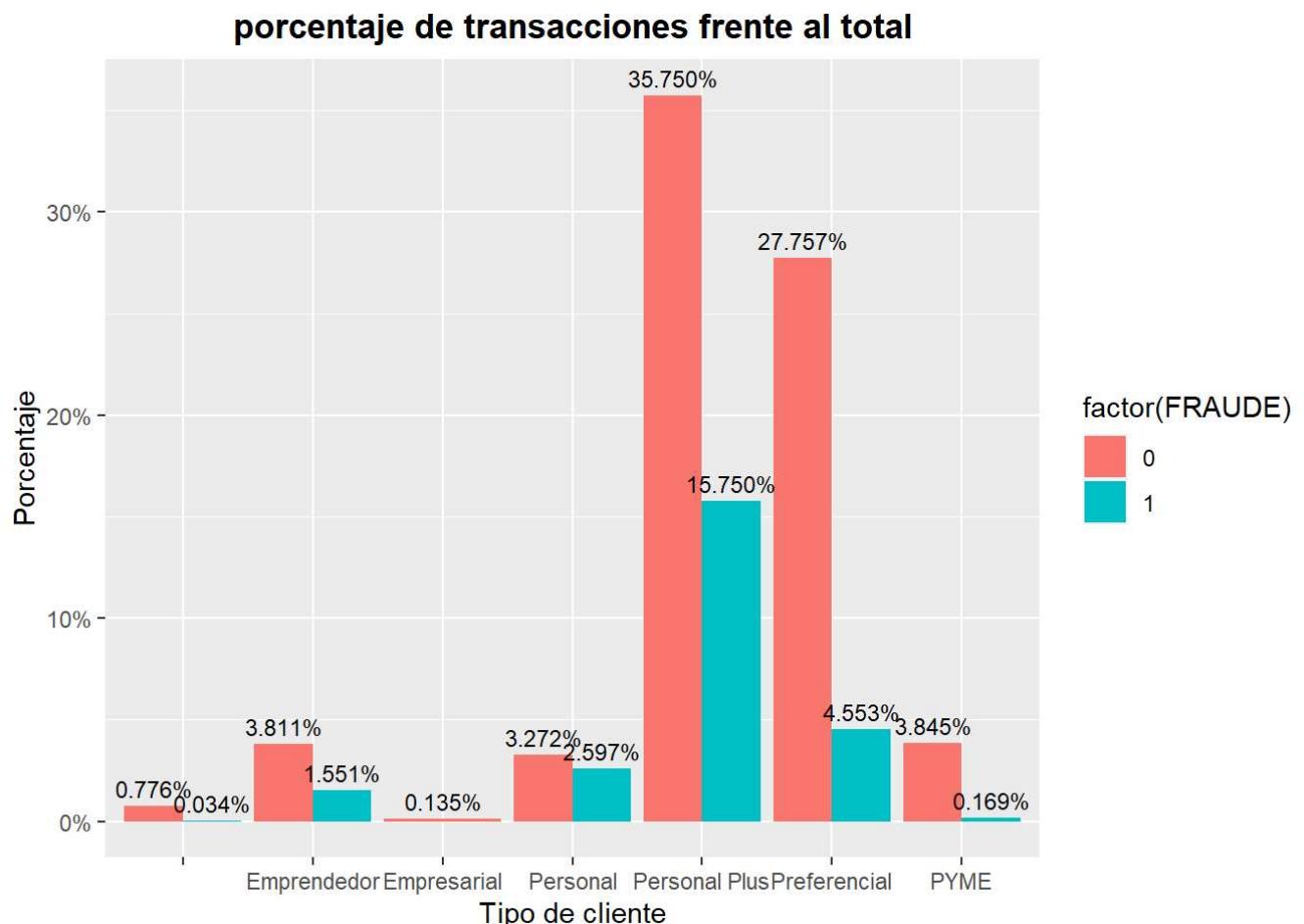
| FRAUDE | porcentaje_operaciones |
|--------|------------------------|
| <int> | <dbl> |
| 0 | 75.35 |
| 1 | 24.65 |

2 rows

A continuación representaré la relación de las variables categóricas con la variable target:

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

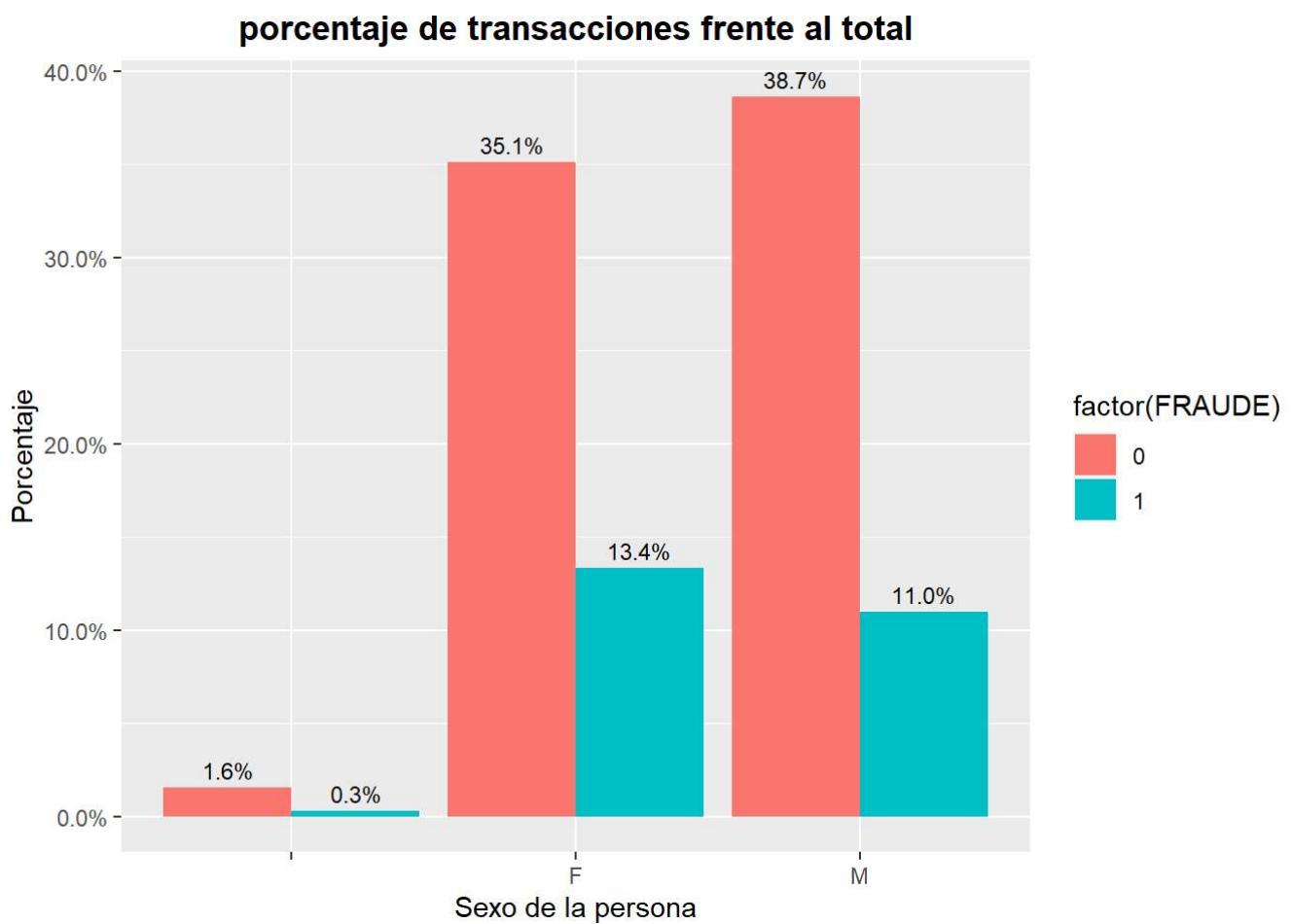
ggplot(data = datos, aes(x = factor(SEGMENTO),
                           y = prop.table(stat(count)), fill = factor(FRAUDE),
                           label = scales::percent(prop.table(stat(count)))) +
  geom_bar(position = "dodge") +
  geom_text(stat = 'count',
            position = position_dodge(.9),
            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
```



```

ggplot(data = datos, 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

```

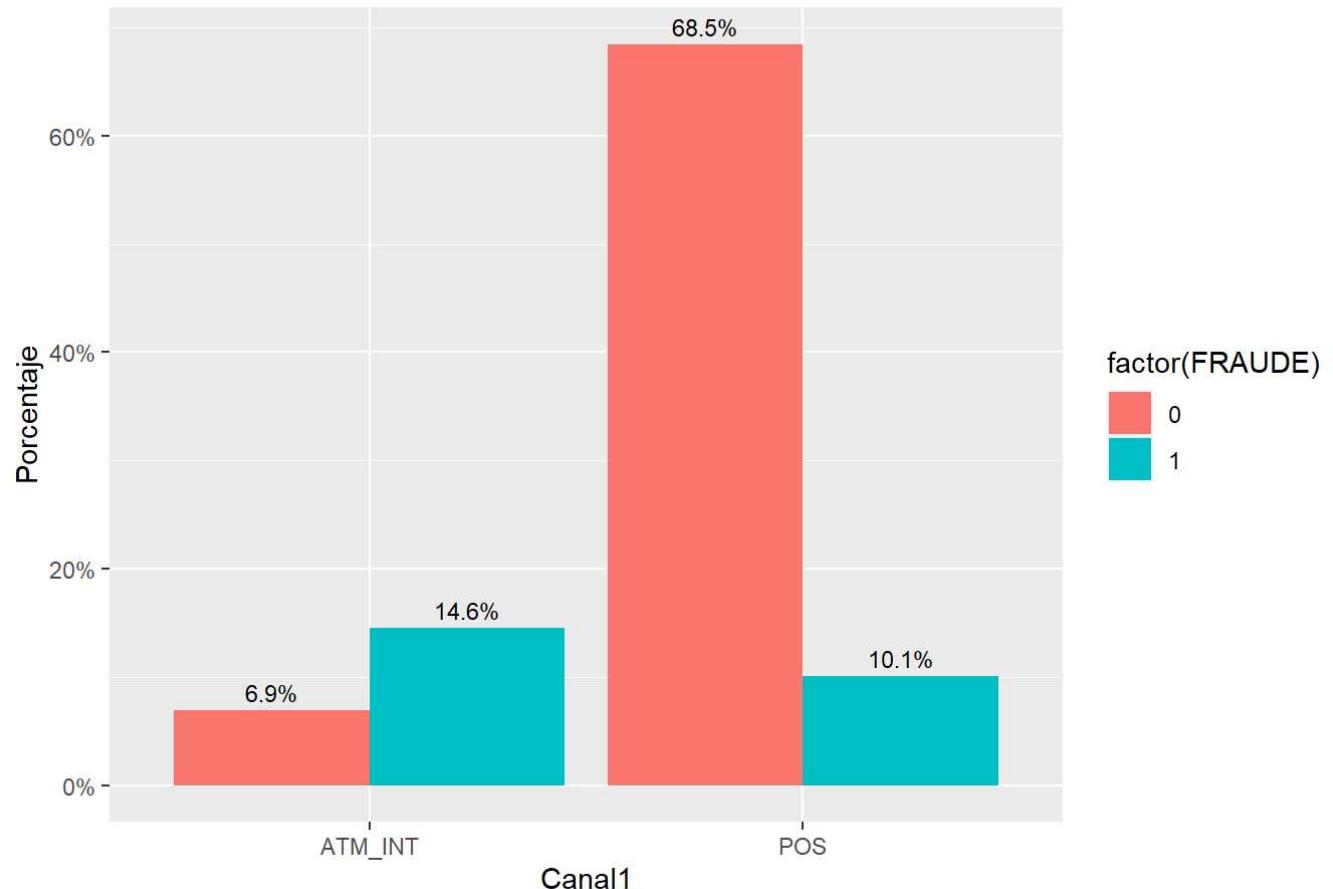


```

ggplot(data = datos, 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

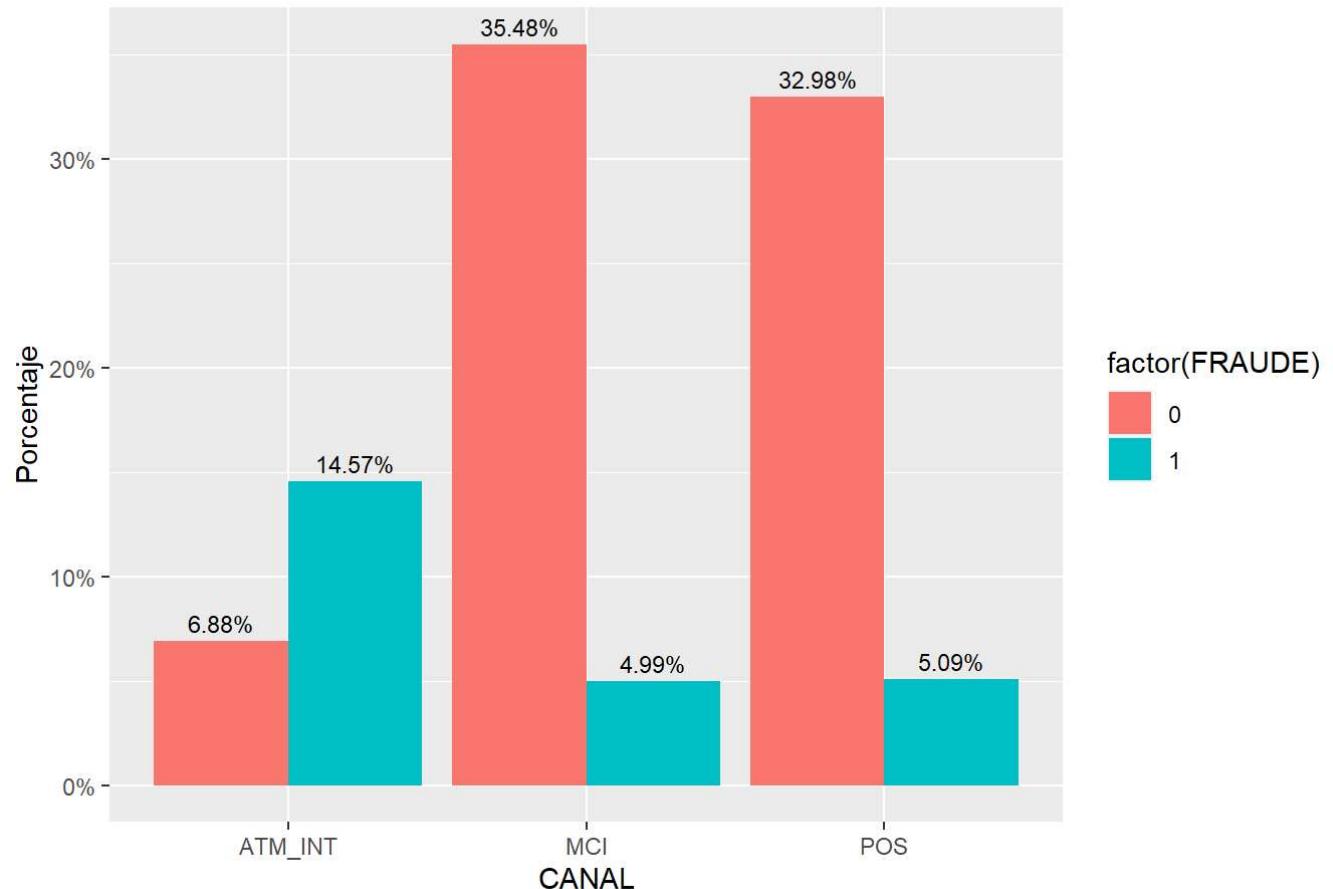
```

porcentaje de transacciones frente al total



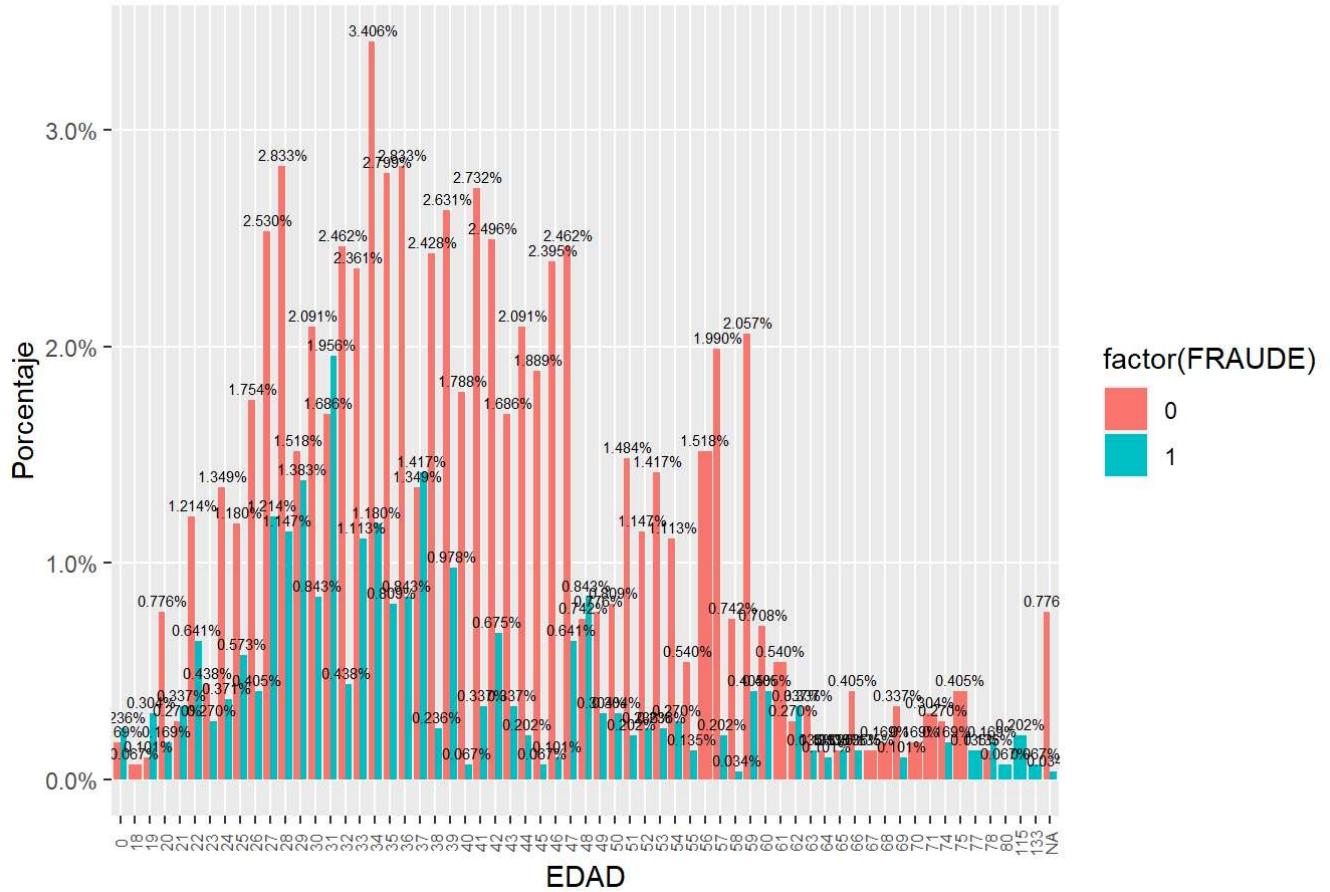
```
ggplot(data = datos, aes(x = factor(CANAL),
                           y = prop.table(stat(count)), fill = factor(FRAUDE),
                           label = scales::percent(prop.table(stat(count)))) +
  geom_bar(position = "dodge") +
  geom_text(stat = 'count',
            position = position_dodge(.9),
            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



```
ggplot(data = datos, 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

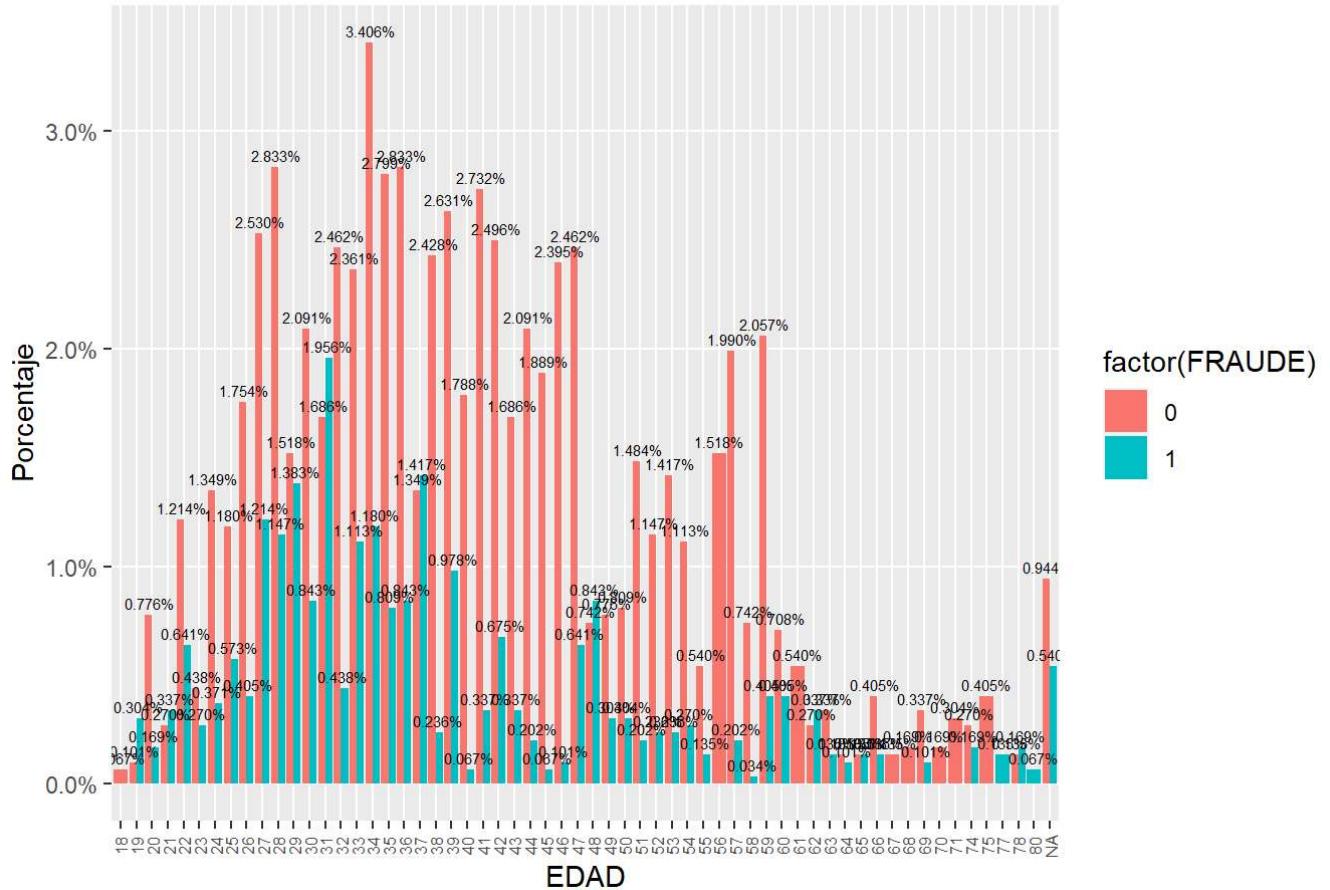


En el caso de la variable EDAD se incluyen valores difícilmente creíbles, transacciones realizadas por personas de 0 años o por personas que superan por mucho los 100 años. Consideraré esos valores como erroneos y los cambiaré por la media de mi muestra.

```
datos$EDAD <- replace( datos$EDAD, datos$EDAD == 0, mean(datos$EDAD))
datos$EDAD <- replace( datos$EDAD, datos$EDAD > 100, mean(datos$EDAD))

ggplot(data = datos, 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



```
data %>%
  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")
```

```
## Error in `ggplot()`:
## ! You're passing a function as global data.
## Have you misspelled the `data` argument in `ggplot()`?
```

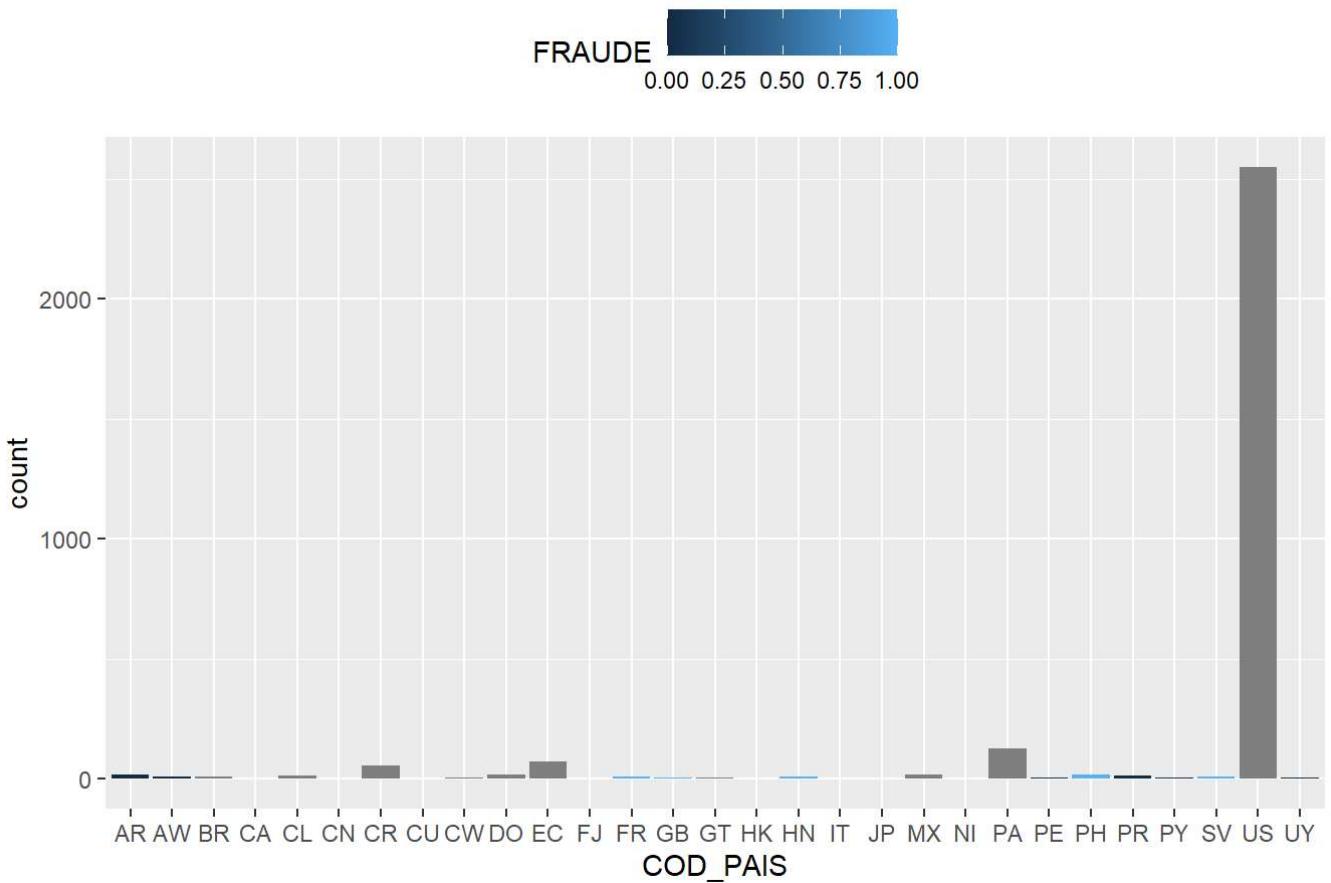
El caso de los valores NA y otros valores atípicos en el resto de variables se tratará más adelante.

El caso particular de la variable COD_PAIS:

```
df_pais <- datos %>% 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)) +
  ggtitle("Número de transacciones por países")+
  geom_bar(aes(fill = FRAUDE), position = position_stack(reverse = TRUE)) +
  theme(legend.position = "top")
```

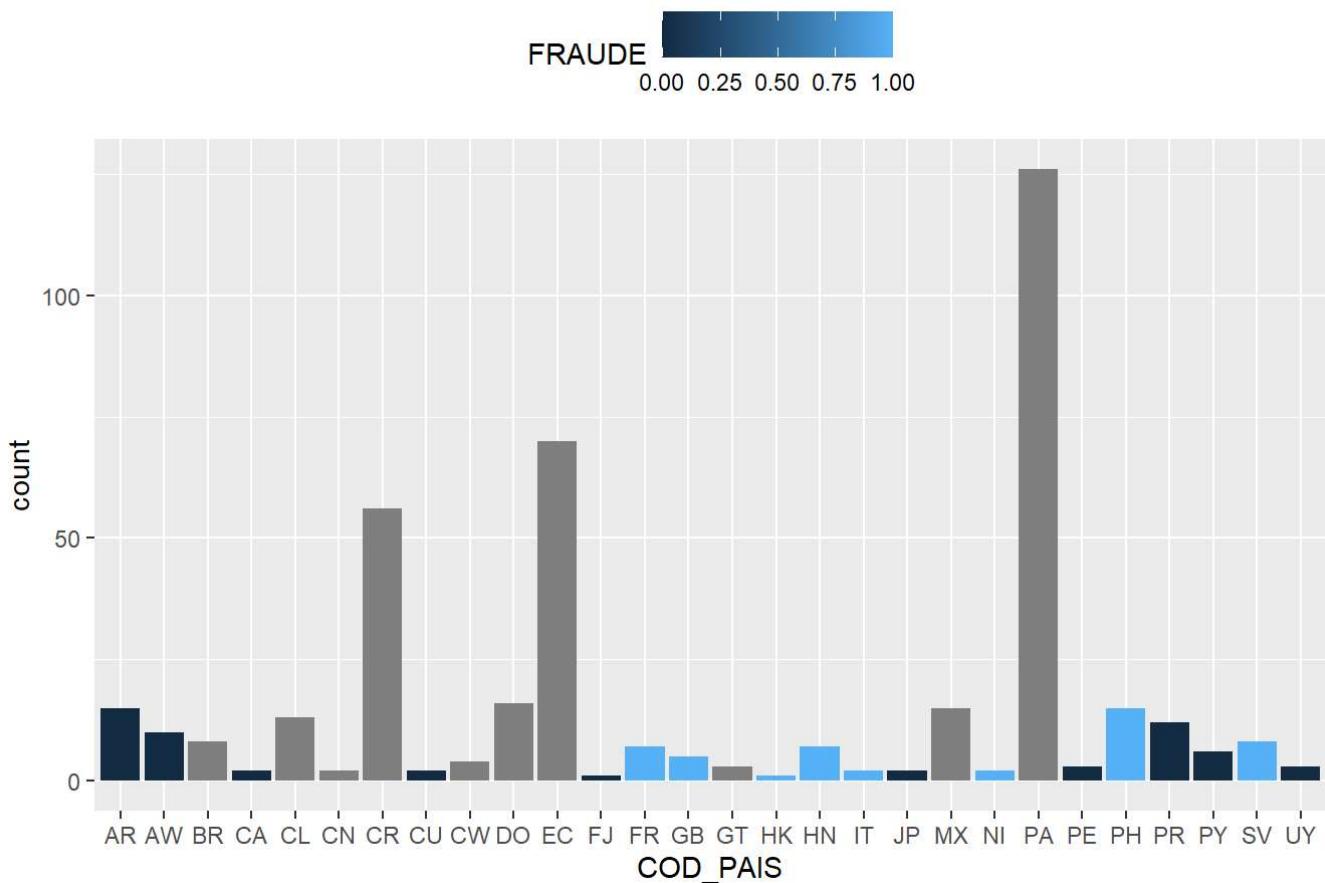
Número de transacciones por países



```
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)) +  
  ggtitle("Número de transacciones por países (SIN US)") +  
  geom_bar(aes(fill = FRAUDE), position = position_stack(reverse = TRUE)) +  
  theme(legend.position = "top")
```

Número de transacciones por países (SIN US)

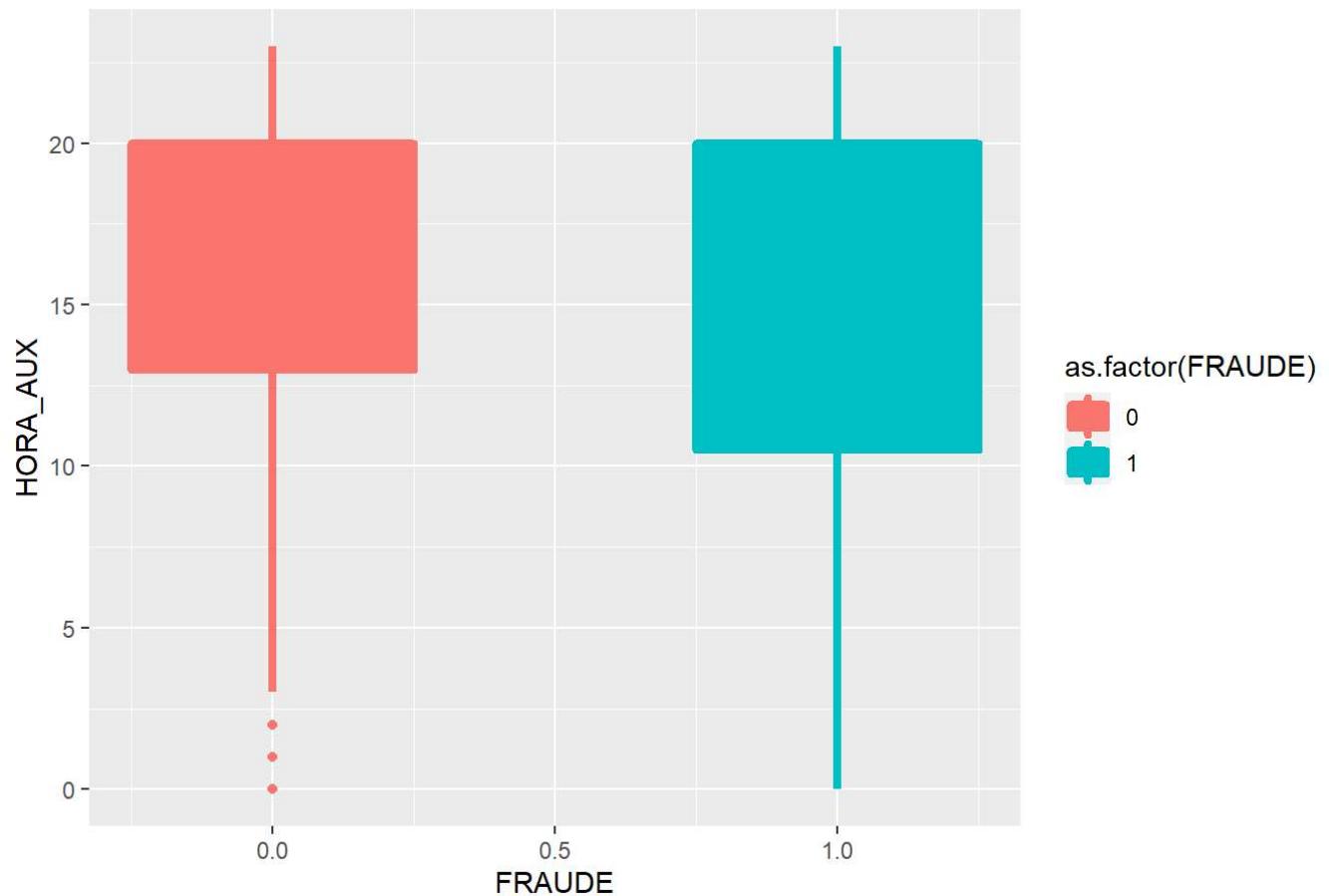


Estos dos gráficos nos permiten observar como la distribución de las operaciones entre los distintos países es muy desigual. En este sentido, la mayoría de las transacciones ocurren en Estados Unidos. Por lo tanto, no tendrá en cuenta esta variable dentro de mi modelo.

Para mostrar la relación y distribución de las variables numéricas respecto a la variable target utilizaré el gráfico boxplot para cada una de ellas.

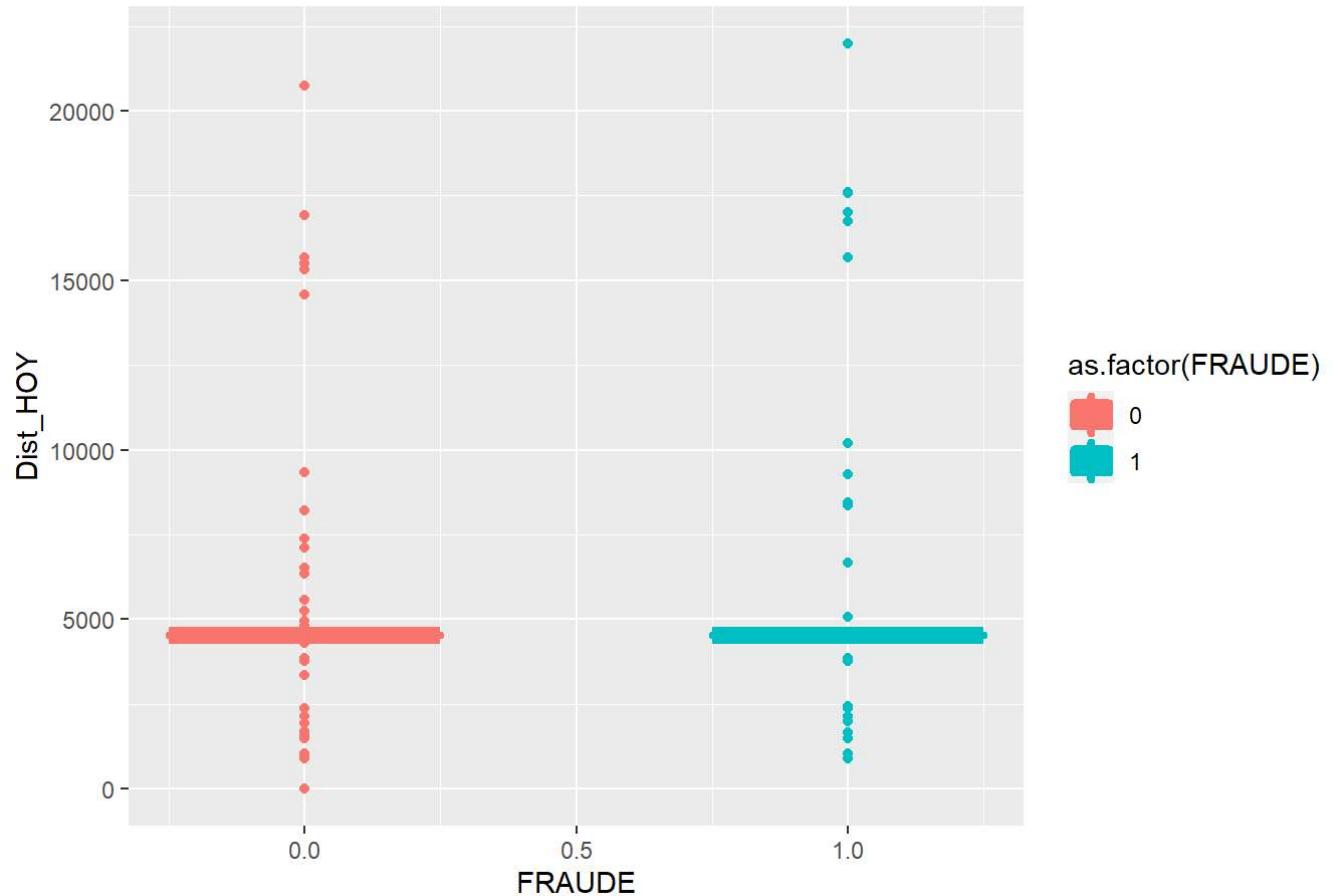
```
datos %>%
  ggplot(aes(x=FRAUDE, y=HORA_AUX, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  #geom_jitter(width=0.15)+
  labs(subtitle="Hora de la transacción")
```

Hora de la transacción



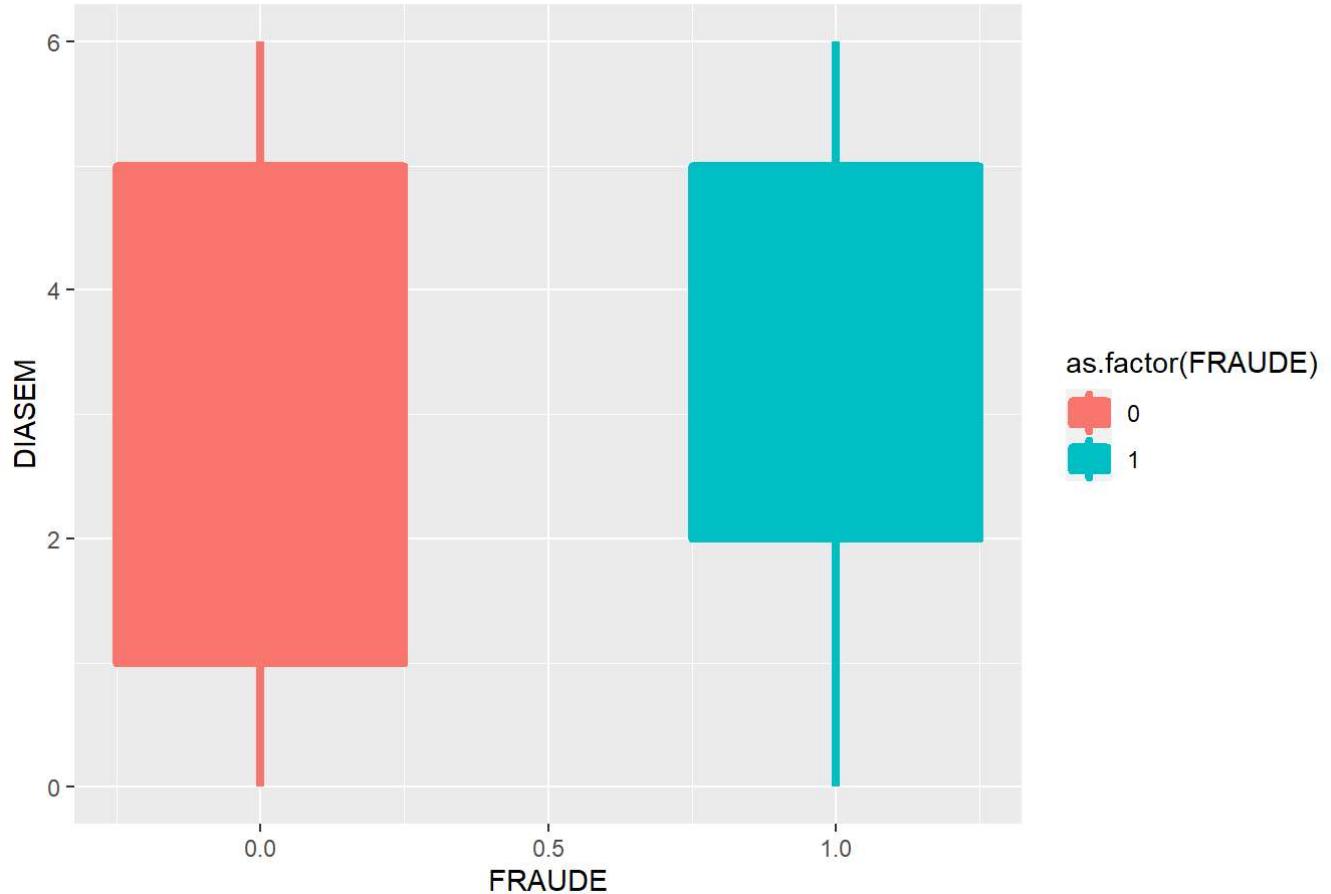
```
datos %>%
  ggplot(aes(x=FRAUDE, y=Dist_HOY, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  #geom_jitter(width=0.15) +
  labs(subtitle="Dist maxima recorrida a nivel nacional")
```

Dist maxima recorrida a nivel nacional



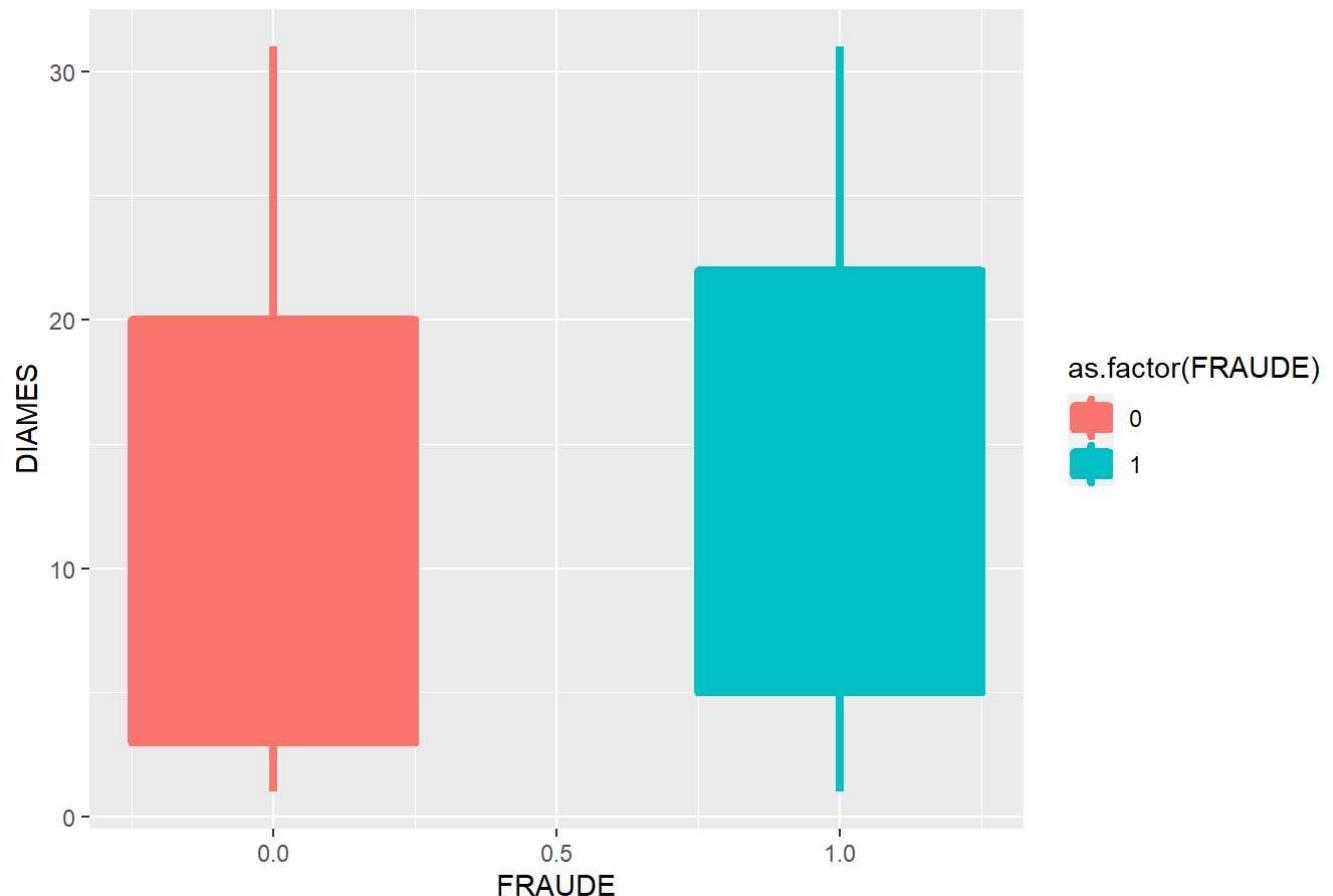
```
datos %>%
  ggplot(aes(x=FRAUDE, y=DIASEM, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  #geom_jitter(width=0.15)+
  labs(subtitle="Día de la semana que se realizó la transacción")
```

Día de la semana que se realizó la transacción



```
datos %>%
  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")
```

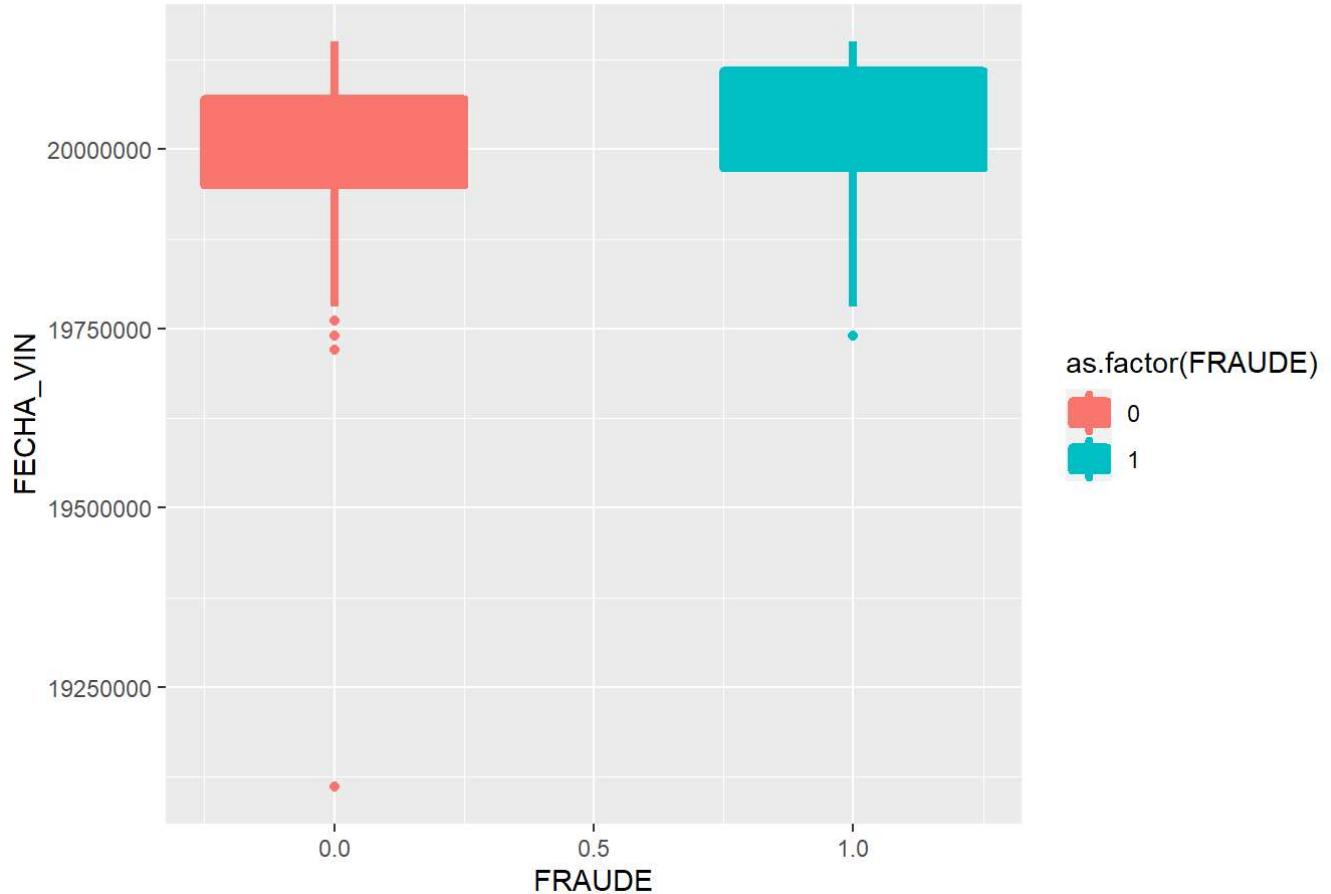
Día del mes que se realizó la transacción



```
datos %>%
  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).
```

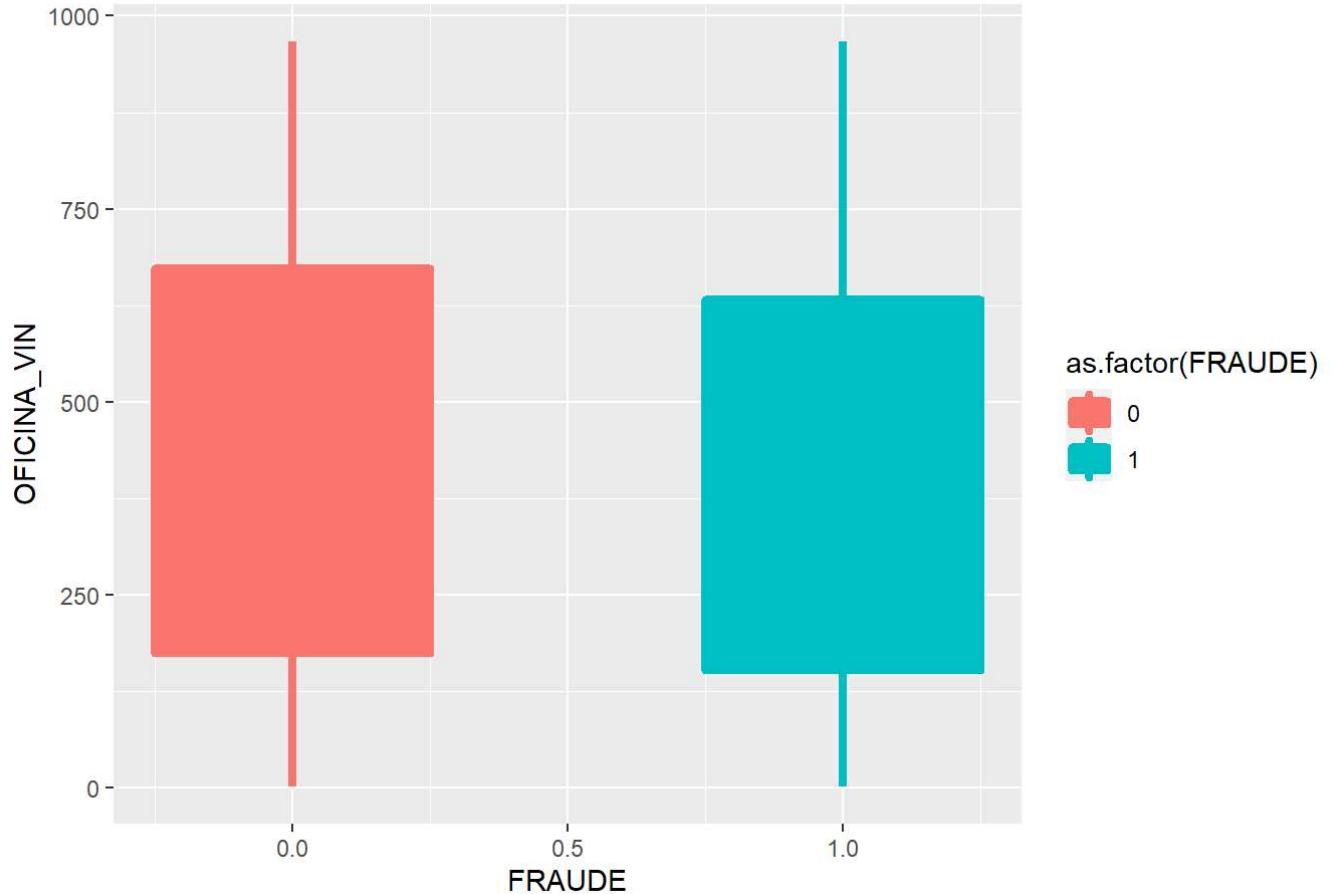
Fecha de vinculacion del cliente



```
datos %>%
  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")
```

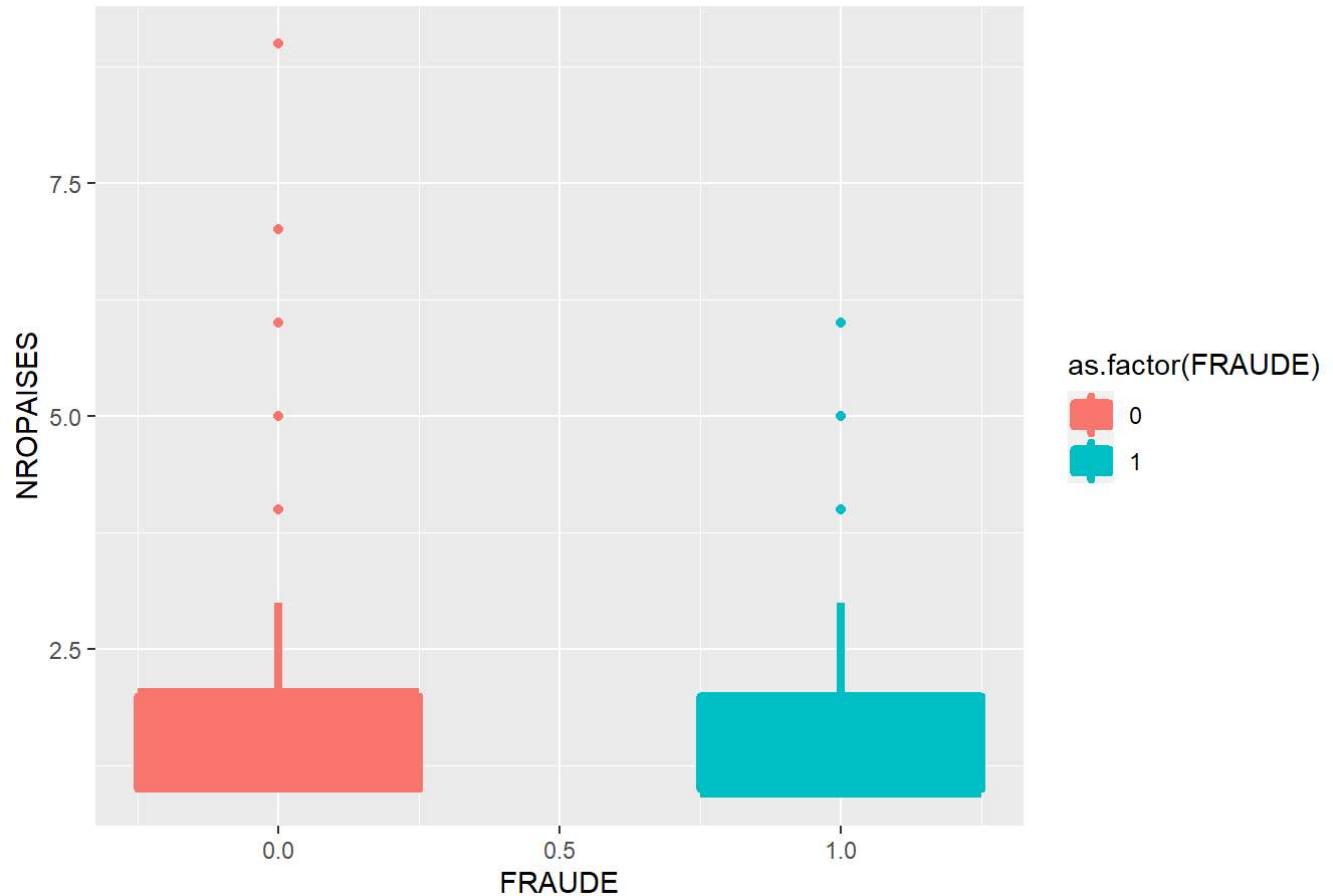
```
## Warning: Removed 24 rows containing non-finite values (stat_boxplot).
```

Oficina de vinculacion del cliente



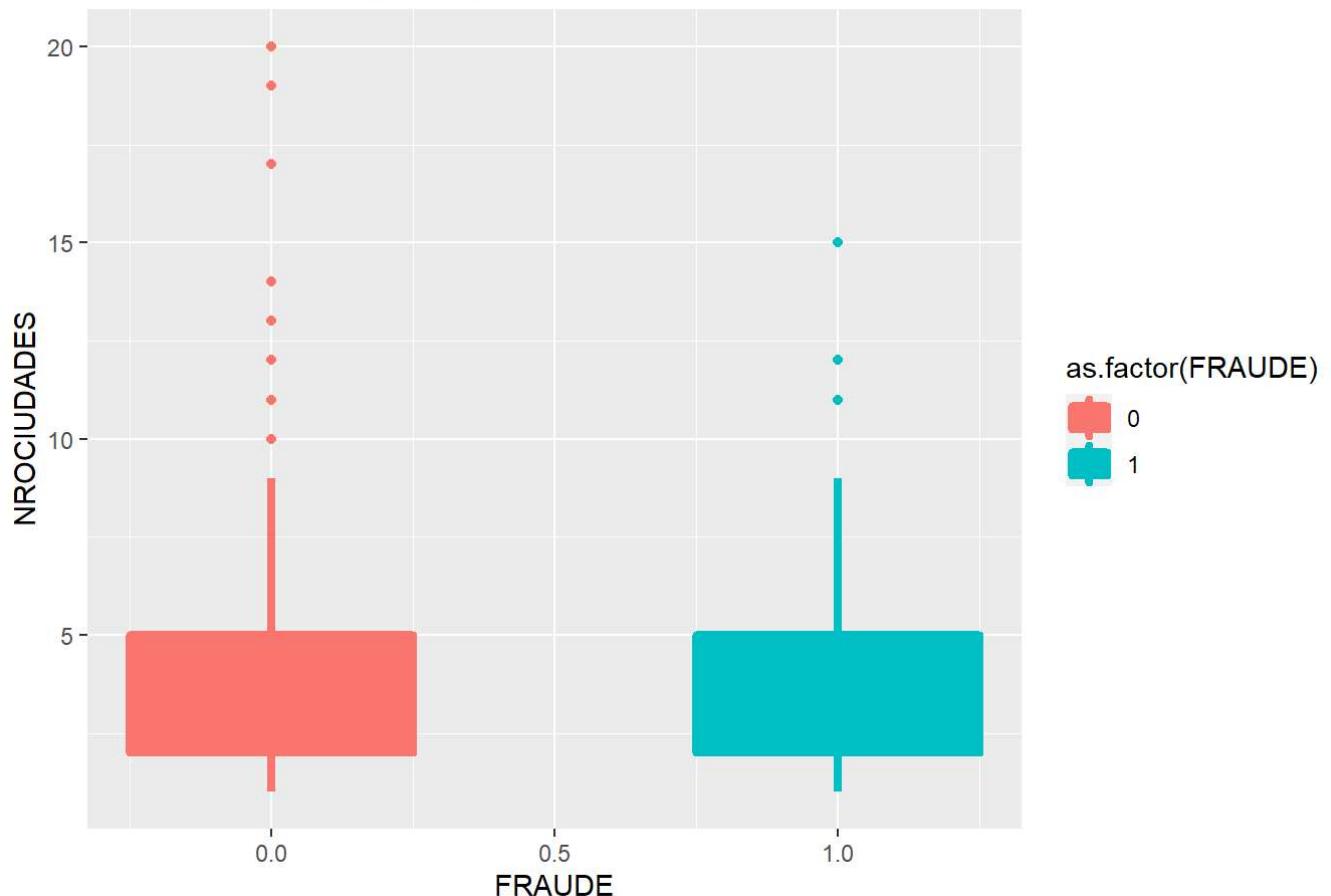
```
datos %>%
  ggplot(aes(x=FRAUDE, y=NROPAISES, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  #geom_jitter(width=0.15) +
  labs(subtitle="Países visitados")
```

Países visitados



```
datos %>%
  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")
```

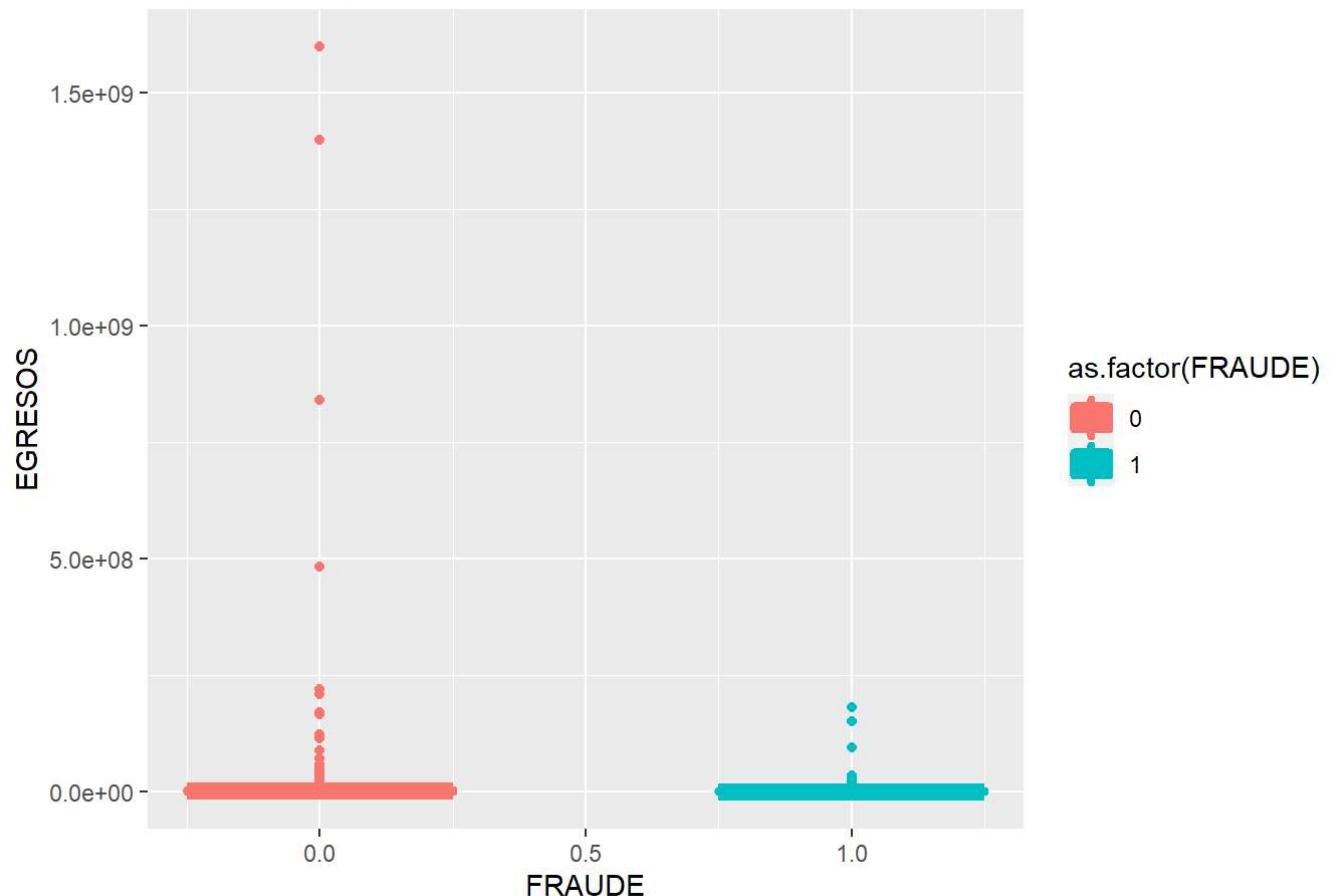
Numero de ciudades nacionales visitadas



```
datos %>%
  ggplot(aes(x=FRAUDE, y=EGRESOS, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  #geom_jitter(width=0.15) +
  labs(subtitle="Engresos del cliente")
```

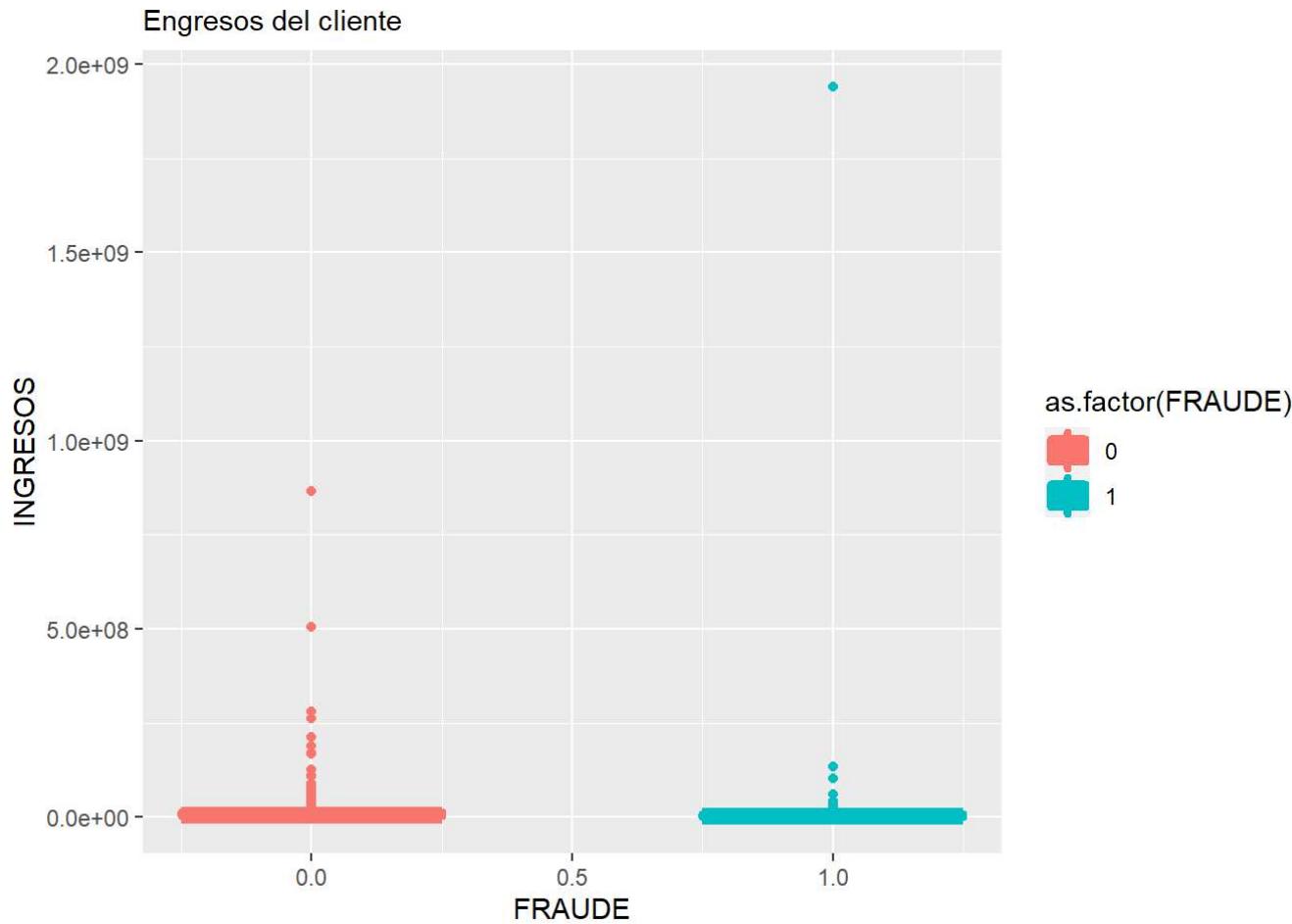
```
## Warning: Removed 24 rows containing non-finite values (stat_boxplot).
```

Engresos del cliente



```
datos %>%
  ggplot(aes(x=FRAUDE, y=INGRESOS, fill=as.factor(FRAUDE))) +
  geom_boxplot(width=0.5,lwd=1.5,aes(color=as.factor(FRAUDE))) +
  #geom_jitter(width=0.15) +
  labs(subtitle="Engresos del cliente")
```

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



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

Eliminación de valores atípicos

En el caso de las variables INGRESOS Y EGRESOS, podría parecer que los valores más alejados de la mediana son atípicos. Sin embargo, por la naturaleza de estas variables, voy a mantener esos valores de momento.

En el caso de la variable FECHA_VIN si voy a reemplazar por la media aquellos valores atípicos que estén por debajo del umbral inferior.

```
#summary(datos$FECHA_VIN)

#primer cuartil = 19951024
#Max = 20150427

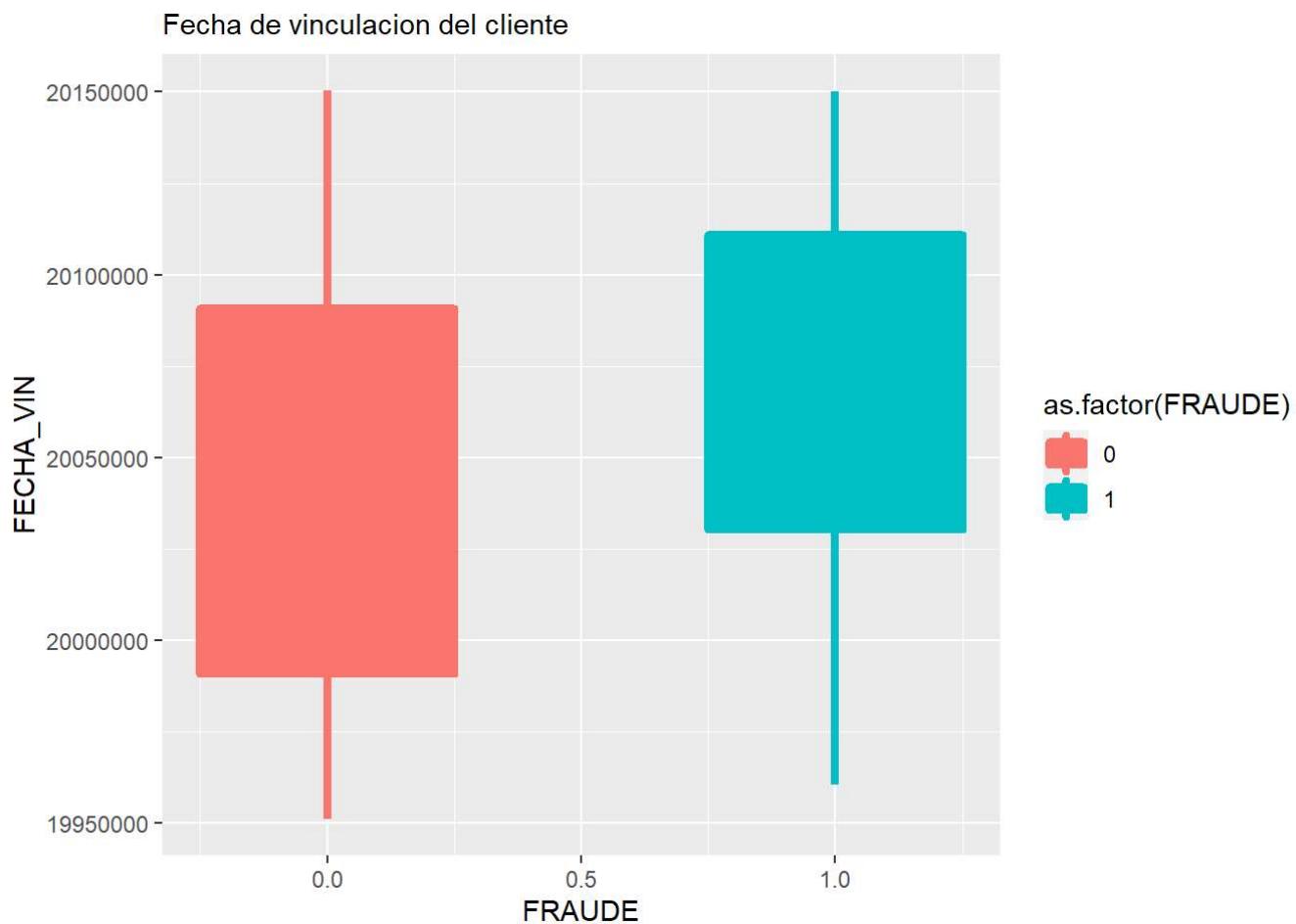
outliersReplace <- function(data, lowLimit, highLimit){
  data[data < lowLimit] <- mean(data)
  data[data > highLimit] <- median(data)
  data      #devolvemos el dato
}

datos$FECHA_VIN<- outliersReplace(datos$FECHA_VIN, 19951024, 20150427)
```

Comprobamos cómo se han remplazado los valores “atípicos”:

```
datos %>%
  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 759 rows containing non-finite values (stat_boxplot).



**Sustitución de valores NA con

```
apply(is.na(datos), 2, sum) #Vuelvo a consultar Los valores NA de cada variable
```

| | 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 | 759 | 24 | 0 | 0 |
| ## | EDAD | INGRESOS | EGRESOS | NROPAISES | Dist_Sum_INTER |
| ## | 44 | 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 | | | | |

Para no tener que transformar los valores nulos asumiendo su valor, voy a utilizar la librería missForest para imputación de valores.

```

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

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

datos <- dplyr::select(datos, -id,-COD_PAIS,-Dist_Sum_INTER,-Dist_Max_INTER,-Dist_sum_NAL,-Di
st_max_NAL)#elimino variables que no voy a utilizar en mi modelo

#imputar valores
df_impo <- missForest(datos)

```

```
## 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!
## missForest iteration 2 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!
## missForest iteration 3 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!
```

```

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

```

```

##      NRMSE      PFC
## 0.5545487 0.0000000

```

```

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      Canal1      FECHA
##          0           0           0           0           0
##        CANAL      DIASEM     DIAMES    FECHA_VIN  OFICINA_VIN
##          0           0           0           0           0
##        SEXO      SEGMENTO     EDAD      INGRESOS     EGRESOS
##          0           0           0           0           0
## NROPAISES Dist_Mean_INTER  NROCIUDADES Dist_Mean_NAL  Dist_HOY
##          0           0           0           0           0

```

```
df <- df_impo$ximp
```

#Me aseguro de que todos Los valores NA han sido remplazados

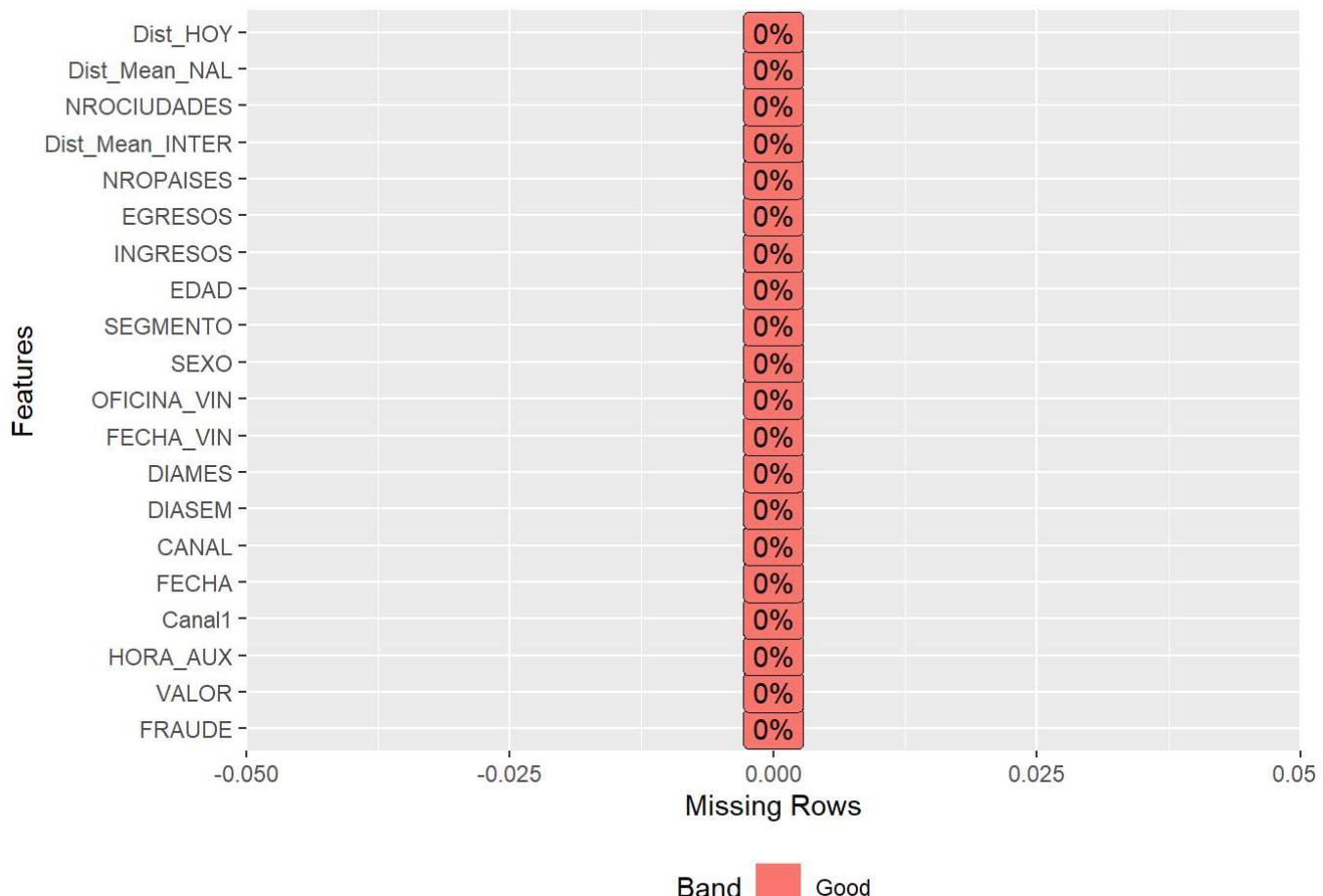
```

##          FRAUDE      VALOR     HORA_AUX      Canal1      FECHA
##          0           0           0           0           0
##        CANAL      DIASEM     DIAMES    FECHA_VIN  OFICINA_VIN
##          0           0           0           0           0
##        SEXO      SEGMENTO     EDAD      INGRESOS     EGRESOS
##          0           0           0           0           0
## NROPAISES Dist_Mean_INTER  NROCIUDADES Dist_Mean_NAL  Dist_HOY
##          0           0           0           0           0

```

#Me aseguro de que todos Los valores NA han sido remplazados

```
df %>% plot_missing()
```



Ya no tenemos valores faltantes en el dataset, continuo con el análisis de dependencias antes de modelizar.

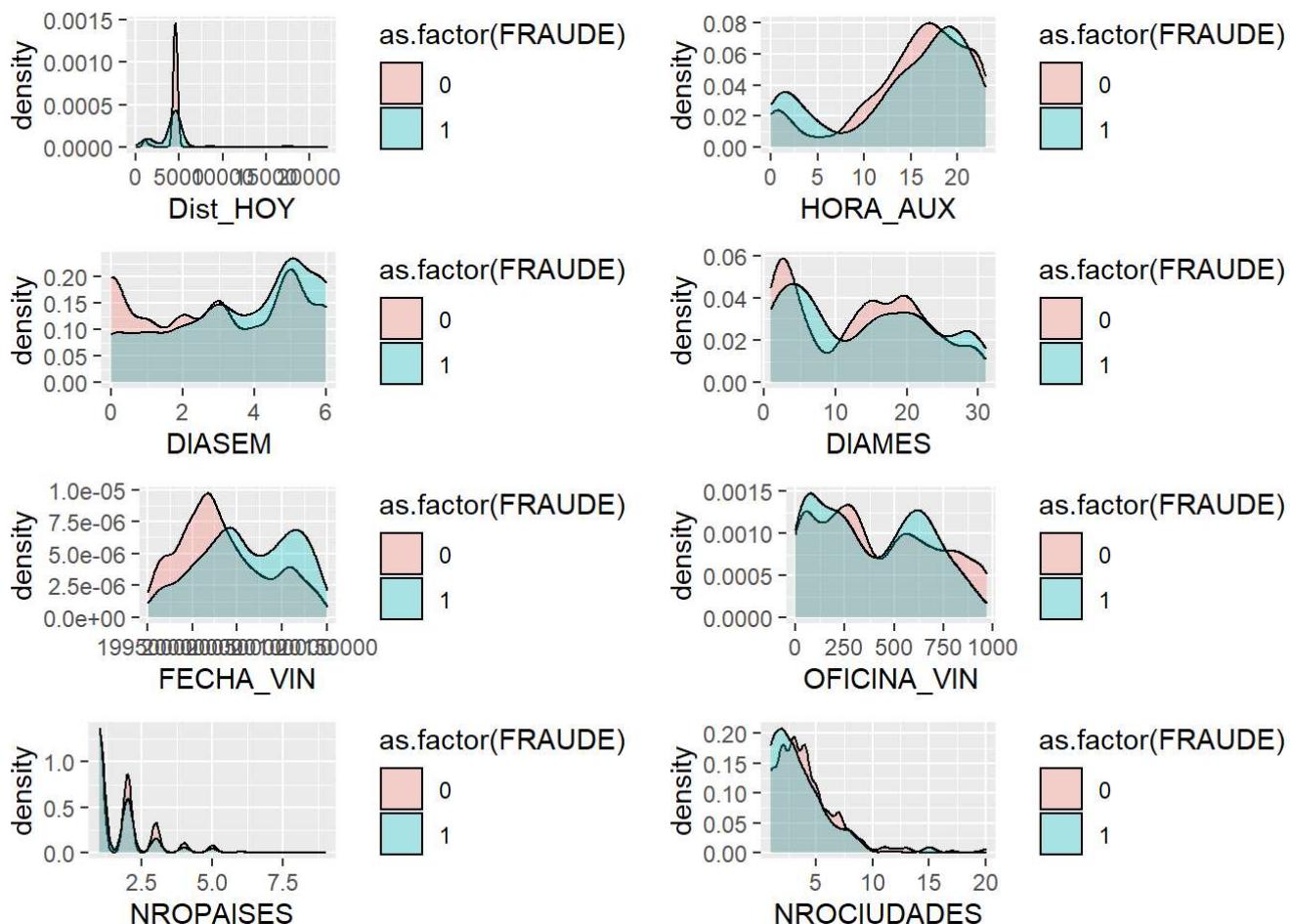
Análisis de dependencias

Antes de empezar a analizar las correlaciones entre variables voy a mostrar algunos gráficos que ayudan a complementar el análisis exploratorio inicial.

Para complementar los boxplot anteriores muestro gráficos de densidad para las variables numéricas respecto de la variable target.

```
p1 = ggplot(df, aes(Dist_HOY, group=FRAUDE, fill=as.factor(FRAUDE))) + geom_density( alpha=.3 )
p2 = ggplot(df, aes(HORA_AUX, group=FRAUDE, fill=as.factor(FRAUDE))) + geom_density( alpha=.3 )
p3 = ggplot(df, aes(DIASEM, group=FRAUDE, fill=as.factor(FRAUDE))) + geom_density( alpha=.3 )
p4 = ggplot(df, aes(DIAMES, group=FRAUDE, fill=as.factor(FRAUDE))) + geom_density( alpha=.3 )
p5 = ggplot(df, aes(FECHA_VIN, group=FRAUDE, fill=as.factor(FRAUDE))) + geom_density( alpha=.3 )
p6 = ggplot(df, aes(OFICINA_VIN, group=FRAUDE, fill=as.factor(FRAUDE))) + geom_density( alpha=.3 )
p7 = ggplot(df, aes(NROPAISES, group=FRAUDE, fill=as.factor(FRAUDE))) + geom_density( alpha=.3 )
p8 = ggplot(df, aes(NROCIUDADES, group=FRAUDE, fill=as.factor(FRAUDE))) + geom_density( alpha=.3 )

grid.arrange(p1,p2,p3,p4,p5,p6,p7,p8, ncol=2)
```



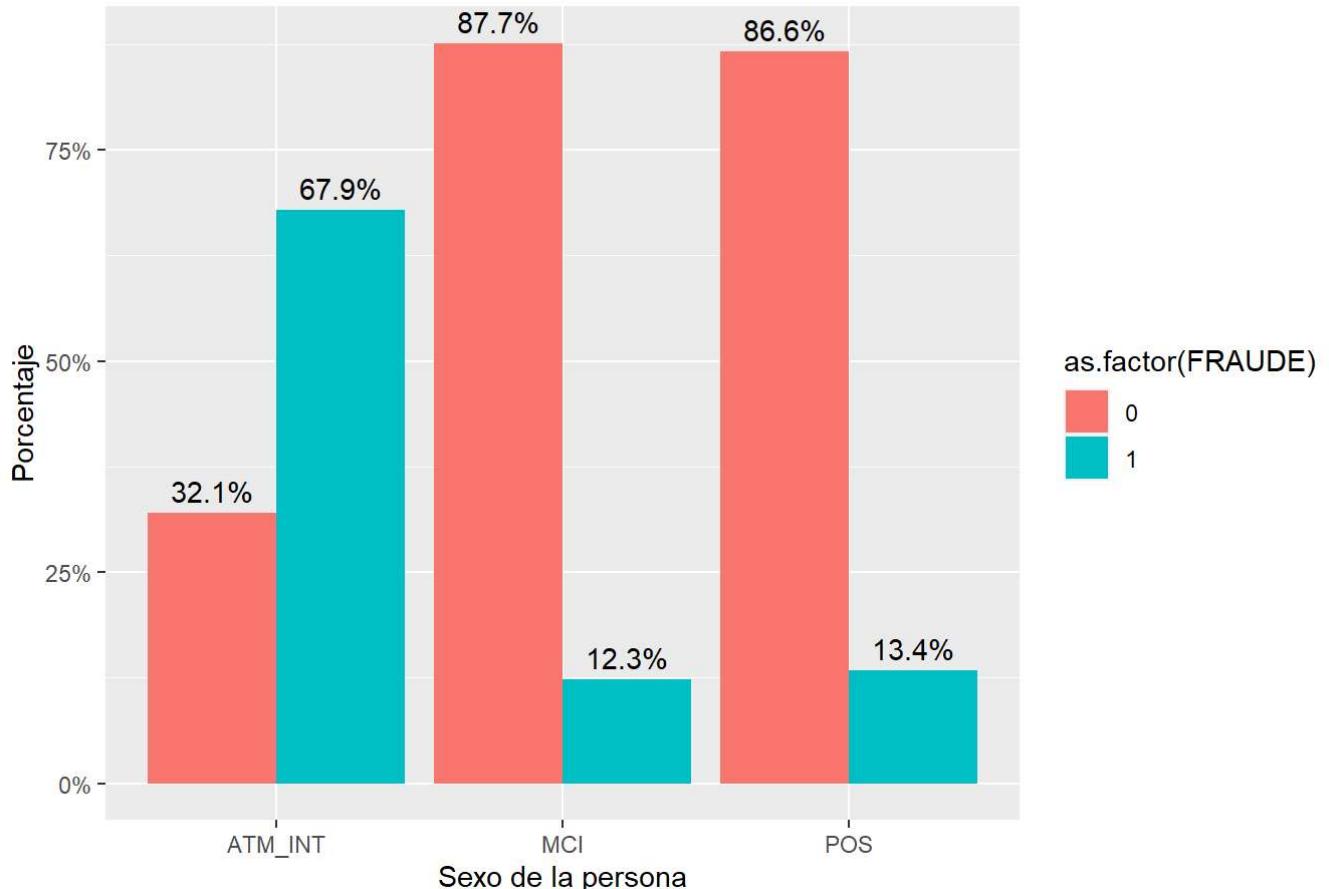
Ahora muestro la relación entre algunas de las variables de forma gráfica:

```

ggplot(df, 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(..co
unt../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")

```

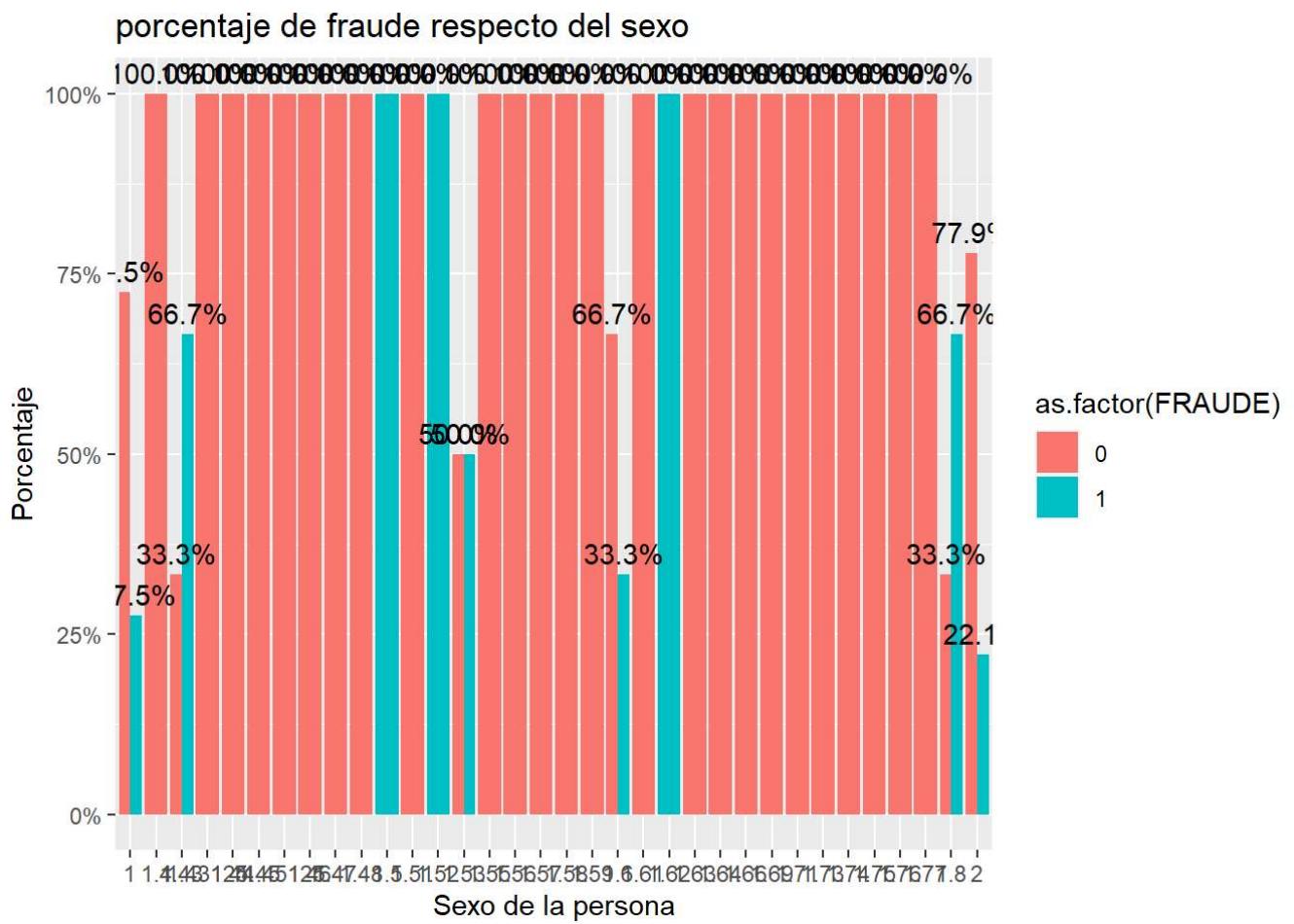
porcentaje de fraude respecto del canal transaccional



```

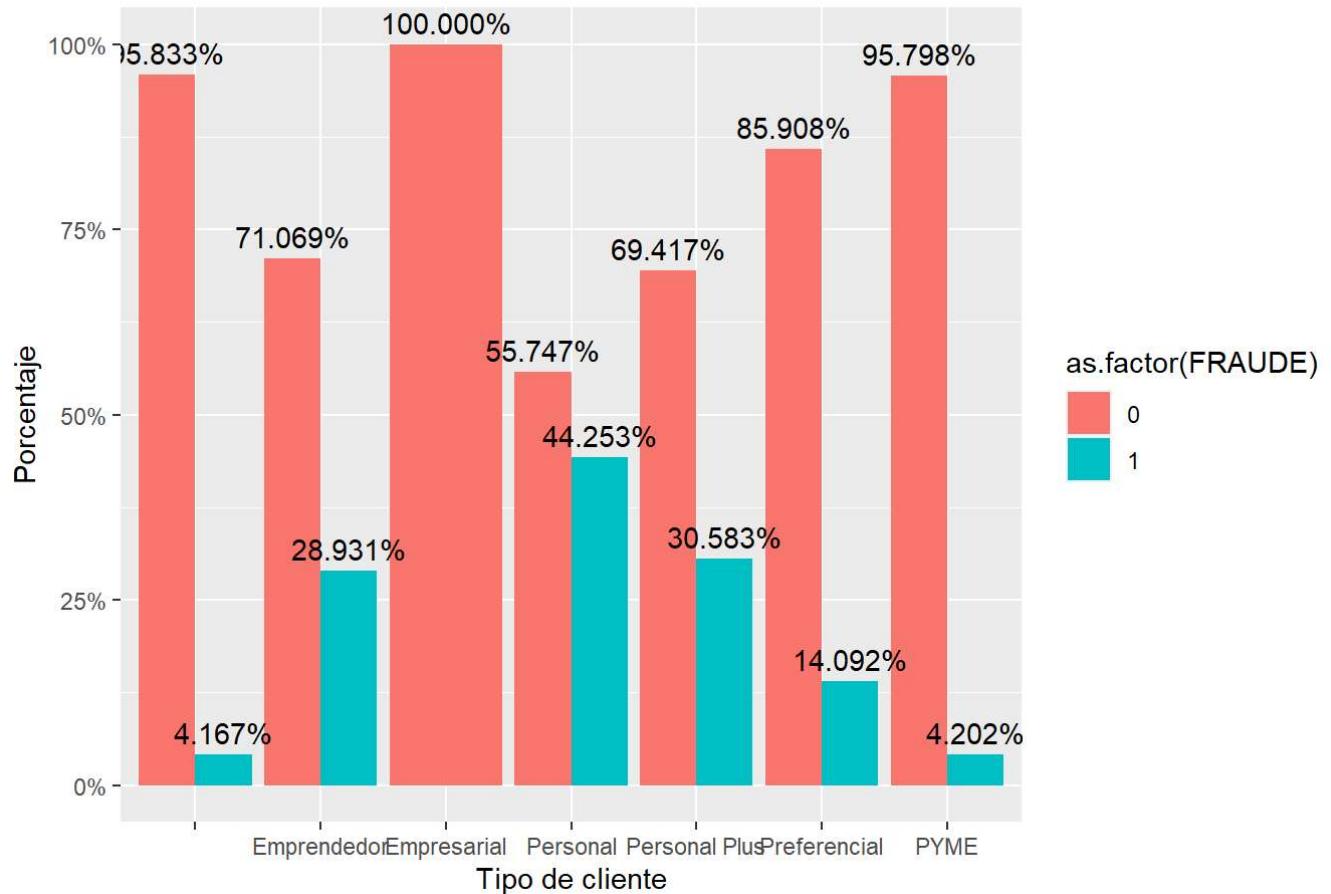
ggplot(df, aes(x = factor(SEXO), fill=as.factor(FRAUDE)))+
  geom_bar(aes( y=..count../tapply(..count.., ..x.. ,sum)[..x..]), position="dodge" ) +
  geom_text(aes( y=..count../tapply(..count.., ..x.. ,sum)[..x..], label=scales::percent(..co
unt../tapply(..count.., ..x.. ,sum)[..x..]) ),
            stat="count", 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")

```



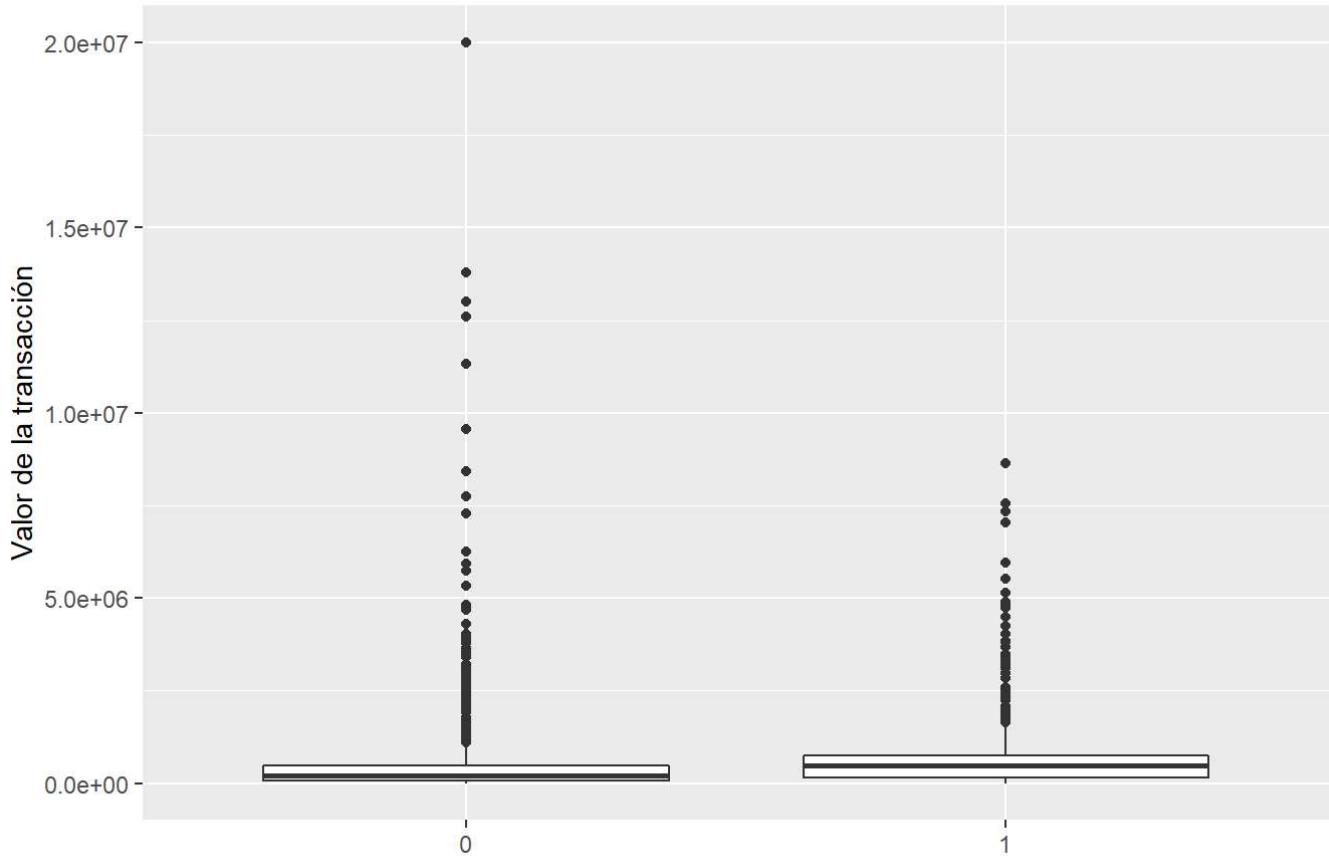
```
ggplot(df, aes(x = factor(SEGMENTO), fill=as.factor(FRAUDE)))+
  geom_bar(aes( y=..count../tapply(..count.., ..x.. ,sum)[..x..]), position="dodge" ) +
  geom_text(aes( y=..count../tapply(..count.., ..x.. ,sum)[..x..], label=scales::percent(..count../tapply(..count.., ..x.. ,sum)[..x..]) ),
            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")
```

porcentaje de fraude respecto del tipo de cliente



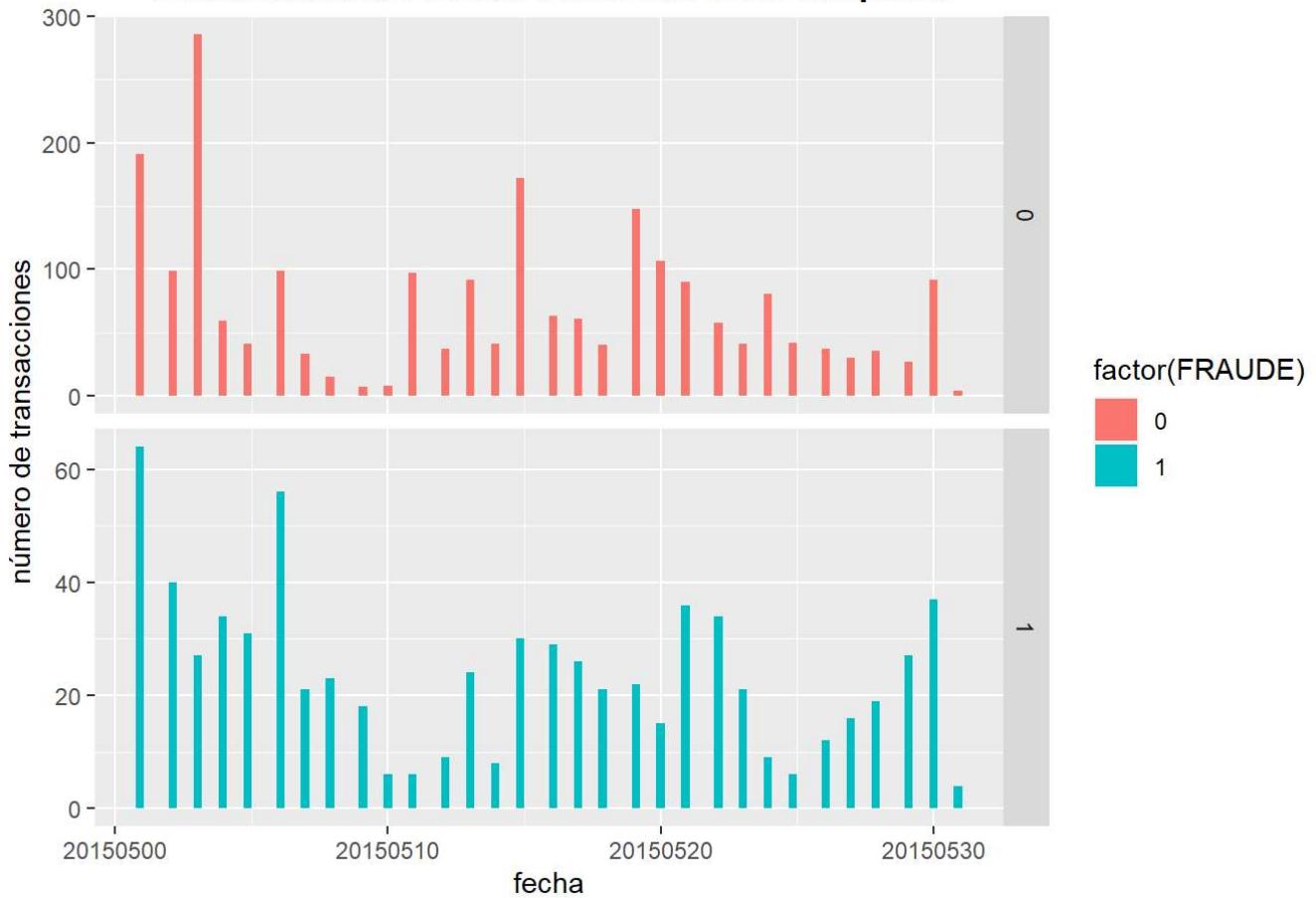
```
ggplot(df, aes(x = factor(FRAUDE), y = VALOR)) + geom_boxplot() +  
  labs(x = ' ',x='1= Fraude; 0=No fraude', y = 'Valor de la transacción') +  
  ggtitle("Distribucion de las transacciones según si son declaradas fraudulentas o no") + co  
mmon_theme
```

Distribucion de las transacciones según si son declaradas fraudulentas o no



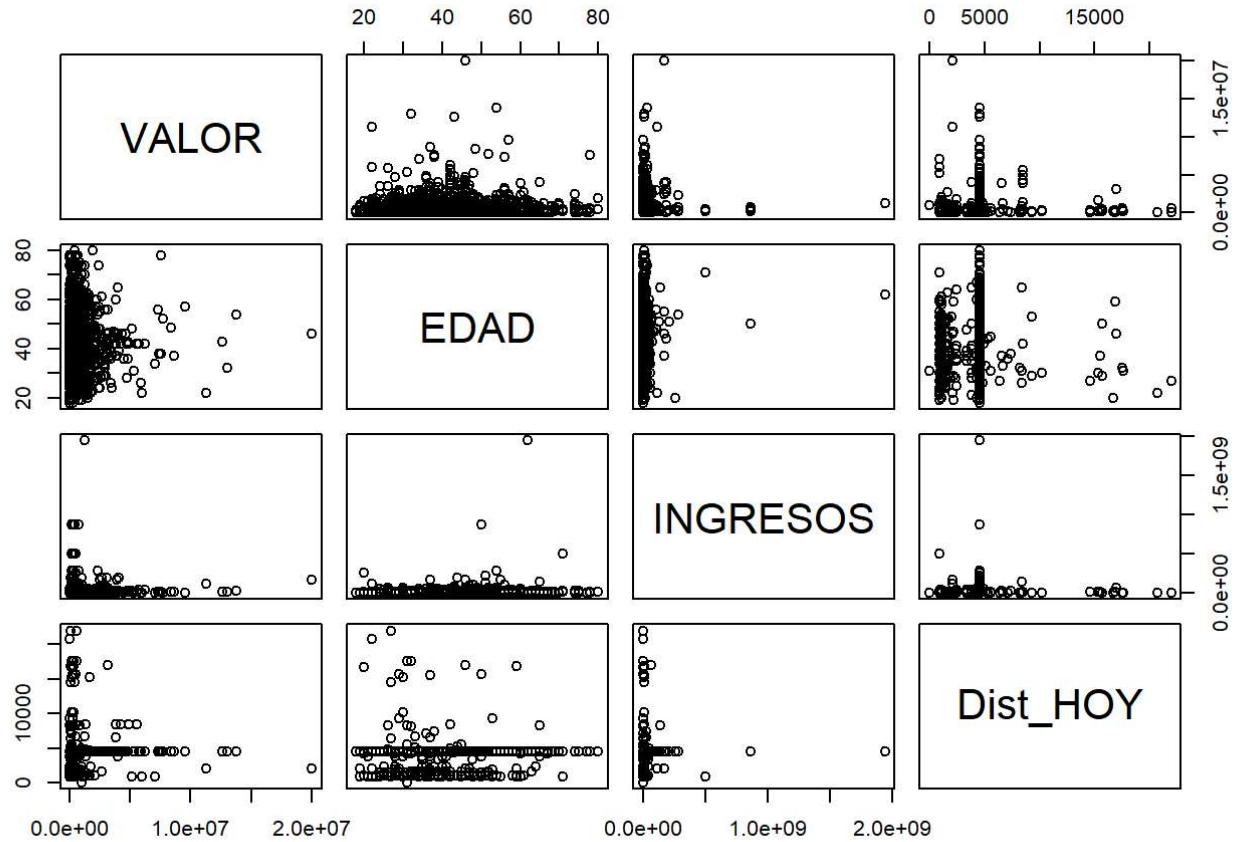
```
df %>%
  ggplot(aes(x = FECHA, fill = factor(FRAUDE))) + geom_histogram(bins = 100) +
  labs(x = 'fecha', y = 'número de transacciones') +
  ggtitle('Distribución del fraude como una serie temporal') +
  facet_grid(FRAUDE ~ ., scales = 'free_y') + common_theme
```

Distribución del fraude como una serie temporal



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

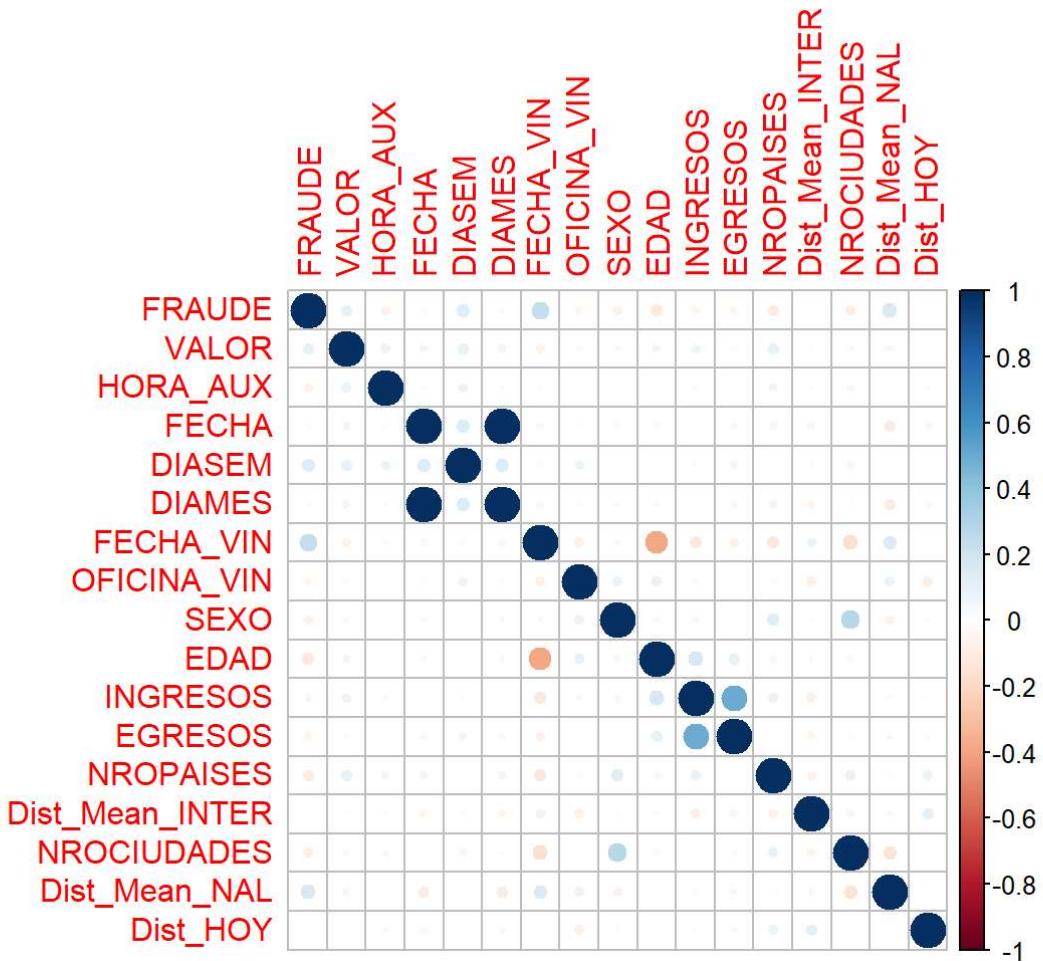
```
#cor(df)
pairs(~VALOR+EDAD+INGRESOS+Dist_HOY, data =df)
```



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, -Canal1, -CANAL, -SEGMENTO) #SElecciono únicamente variables numéricas.
cor.table = cor(df_corr)
corrplot(cor.table, method = "circle")
```



Las variables DIAMES Y FECHA están directamente relacionadas. Para evitar la multicolinealidad en el modelo voy a eliminar la variable DIAMES

```
df<-select(df,-DIAMES)
```

```
## Error in select(df, -DIAMES): unused argument (-DIAMES)
```

```
df<-select(df,-DIAMES)
```

```
## Error in select(df, -DIAMES): unused argument (-DIAMES)
```

```
df<-select(df,-DIAMES)
```

```
## Error in select(df, -DIAMES): unused argument (-DIAMES)
```

Modelización En primer lugar voy a generar una regresión logistica con la totalidad de las variables seleccionadas anteriormente.

```
mod <- glm(FRAUDE~., data=df, family=binomial)
```

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

```
mod
```

```

## 
## Call: glm(formula = FRAUDE ~ ., family = binomial, data = df)
## 
## Coefficients:
##             (Intercept)          VALOR          HORA_AUX
##                 -1.918e+13      3.498e-07     -4.185e-02
##             Canal1POS           FECHA          CANALMCI
##                 -2.652e+00      9.518e+05    -1.262e-01
##             CANALPOS            DIASEM          DIAMES
##                  NA      1.517e-01     -9.518e+05
##             FECHA_VIN        OFICINA_VIN          SEXO
##                 3.379e-06     -2.616e-04    -1.446e-01
## SEGMENTOEmprendedor SEGMENTOEmpresarial SEGMENTOPersonal
##                 1.996e+00     -2.199e+01      2.774e+00
## SEGMENTOPersonal Plus SEGMENTOPreferencial SEGMENTOPYME
##                 2.794e+00      2.742e+00      9.180e-01
##                  EDAD           INGRESOS          EGRESOS
##                 1.416e-03      1.579e-09    -1.992e-09
##             NROPAISES       Dist_Mean_INTER  NROCIUDADES
##                 -5.936e-02     -4.672e-05      5.493e-02
##             Dist_Mean_NAL          Dist_HOY
##                 2.019e-03      3.454e-05
## 
## Degrees of Freedom: 2964 Total (i.e. Null); 2940 Residual
## Null Deviance: 3312
## Residual Deviance: 2361 AIC: 2411

```

#Se muestra el término independiente y Los coeficientes de Las variables predictoras

los exponenciales de los coeficientes, necesarios para la interpretación del odds ratio son los siguientes:

```
exp(mod$coefficients)
```

```

##             (Intercept)          VALOR          HORA_AUX
##                 0.000000e+00      1.000000e+00     9.590108e-01
##             Canal1POS           FECHA          CANALMCI
##                 7.050052e-02           Inf      8.814401e-01
##             CANALPOS            DIASEM          DIAMES
##                  NA      1.163755e+00      0.000000e+00
##             FECHA_VIN        OFICINA_VIN          SEXO
##                 1.000003e+00      9.997385e-01      8.653524e-01
## SEGMENTOEmprendedor SEGMENTOEmpresarial SEGMENTOPersonal
##                 7.362791e+00      2.825806e-10      1.602469e+01
## SEGMENTOPersonal Plus SEGMENTOPreferencial SEGMENTOPYME
##                 1.635180e+01      1.552275e+01      2.504383e+00
##                  EDAD           INGRESOS          EGRESOS
##                 1.001417e+00      1.000000e+00      1.000000e+00
##             NROPAISES       Dist_Mean_INTER  NROCIUDADES
##                 9.423684e-01      9.999533e-01      1.056467e+00
##             Dist_Mean_NAL          Dist_HOY
##                 1.002021e+00      1.000035e+00

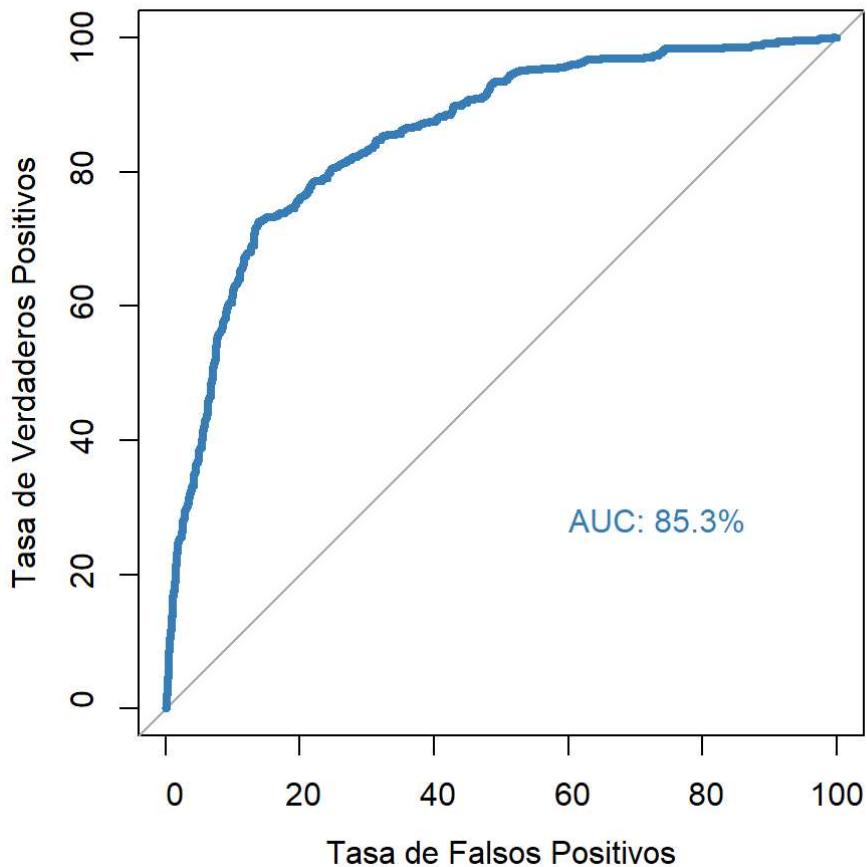
```

Antes de realizar clasificaciones con el modelo logístico, calculo su curva ROC y su métrica AUC, calculando las correspondientes tasas de verdaderos y negativos verdaderos para todos los posibles umbrales que sean significativos.

```
par(pty = "s") # para que el gráfico sea cuadrado  
roc(df$FRAUDE,mod$fitted.values, plot=TRUE, legacy.axes=TRUE, percent=TRUE, xlab="Tasa de Falsos Positivos", ylab="Tasa de Verdaderos Positivos",col="#377eb8",lwd=4, print.auc = TRUE, print.auc.x=40,print.auc.y=30 )
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```



```
##  
## Call:  
## roc.default(response = df$FRAUDE, predictor = mod$fitted.values, percent = TRUE, plot = TRUE, legacy.axes = TRUE, xlab = "Tasa de Falsos Positivos", ylab = "Tasa de Verdaderos Positivos", col = "#377eb8", lwd = 4, print.auc = TRUE, print.auc.x = 40, print.auc.y = 30)  
##  
## Data: mod$fitted.values in 2234 controls (df$FRAUDE 0) < 731 cases (df$FRAUDE 1).  
## Area under the curve: 85.29%
```

Visualizo las primeras predicciones que arroja este primer modelo sobre los primeros registros.

```
y_pred <- predict(mod, subset(df,select = -FRAUDE ), type="response")
```

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

```
head(y_pred) #Probabilidades de que La transacción no sea un fraude
```

```
## 1 2 3 4 5 6  
## 0.8390562 0.7205643 0.8390562 0.8390562 0.6933176 0.6627727
```

parece coherente teniendo en cuenta que el 75% de las transacciones no son declaradas fraudulentas.

Para comprobar si es correcto selecciono un umbral del 0.5 y paso a predecir. Si la probabilidad predicha es superior a 0.5 asignaremos la clase 1 y, en caso contrario, la clase 0.

```
y_pred <- as.numeric(predict(mod, subset(df,select = -FRAUDE ), type="response"  
)>.5)
```

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

```
y <- as.numeric(df$FRAUDE)  
y_pred <- as.factor(y_pred)  
y <- as.factor(y)  
head(y_pred) # observamos Las primeras 5 clases predichas
```

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

```
head(y) # observamos Las primeras 5 clases reales
```

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

El modelo acierta en sus primeras 6 predicciones.

Utilización de el algoritmo stepAIC

Ahora buscaré un modelo logístico haciendo uso de la métrica AIC (maximizar la verosimilitud) a través del método stepwise.

Utilizaré la dirección both en este caso.

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

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

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

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

```
## Start: AIC=3313.94  
## FRAUDE ~ 1
```

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

```

##                                     Df Deviance    AIC
## + Canal1                         1  2583.5 2587.5
## + CANAL                          2  2582.9 2588.9
## + SEGMENTO                       6  3139.5 3153.5
## + FECHA_VIN                      1  3154.6 3158.6
## + Dist_Mean_NAL                  1  3245.5 3249.5
## + DIASEM                         1  3261.3 3265.3
## + EDAD                           1  3272.7 3276.7
## + EGRESOS                        1  3274.0 3278.0
## + NROPAISES                      1  3282.3 3286.3
## + NROCIUDADES                    1  3287.2 3291.2
## + VALOR                          1  3288.5 3292.5
## + INGRESOS                        1  3292.1 3296.1
## + HORA_AUX                        1  3299.7 3303.7
## + SEXO                            1  3300.3 3304.3
## + OFICINA_VIN                     1  3304.2 3308.2
## <none>                           3311.9 3313.9
## + DIAMES                         1  3310.7 3314.7
## + FECHA                           1  3310.7 3314.7
## + Dist_HOY                        1  3311.8 3315.8
## + Dist_Mean_INTER                 1  3311.8 3315.8
##
## Step:  AIC=2587.5
## FRAUDE ~ Canal1
##
##                                     Df Deviance    AIC
## + Dist_Mean_NAL                  1  2538.9 2544.9
## + VALOR                          1  2540.9 2546.9
## + DIASEM                         1  2547.3 2553.3
## + SEGMENTO                       6  2542.7 2558.7
## + HORA_AUX                        1  2566.4 2572.4
## + FECHA_VIN                      1  2570.6 2576.6
## + DIAMES                         1  2577.1 2583.1
## + FECHA                           1  2577.1 2583.1
## + EGRESOS                        1  2579.6 2585.6
## <none>                           2583.5 2587.5
## + EDAD                           1  2581.7 2587.7
## + NROCIUDADES                    1  2582.4 2588.4
## + SEXO                           1  2582.5 2588.5
## + Dist_HOY                        1  2582.6 2588.6
## + CANAL                          1  2582.9 2588.9
## + OFICINA_VIN                     1  2583.0 2589.0
## + Dist_Mean_INTER                 1  2583.1 2589.1
## + NROPAISES                      1  2583.5 2589.5
## + INGRESOS                        1  2583.5 2589.5
## - Canal1                         1  3311.9 3313.9
##
## Step:  AIC=2544.95
## FRAUDE ~ Canal1 + Dist_Mean_NAL
##
##                                     Df Deviance    AIC
## + VALOR                          1  2499.2 2507.2
## + DIASEM                         1  2502.6 2510.6
## + SEGMENTO                       6  2496.0 2514.0
## + HORA_AUX                        1  2521.2 2529.2

```

```

## + DIAMES           1  2528.1 2536.1
## + FECHA            1  2528.1 2536.1
## + FECHA_VIN        1  2531.4 2539.4
## + NROCIUDADES      1  2534.8 2542.8
## + EGRESOS          1  2535.1 2543.1
## <none>             2538.9 2544.9
## + EDAD              1  2537.2 2545.2
## + OFICINA_VIN       1  2537.4 2545.4
## + CANAL             1  2537.8 2545.8
## + Dist_HOY           1  2538.0 2546.0
## + Dist_Mean_INTER    1  2538.3 2546.3
## + SEXO               1  2538.6 2546.6
## + INGRESOS          1  2538.9 2546.9
## + NROPAISES         1  2538.9 2546.9
## - Dist_Mean_NAL      1  2583.5 2587.5
## - Canal1             1  3245.5 3249.5
##
## Step: AIC=2507.24
## FRAUDE ~ Canal1 + Dist_Mean_NAL + VALOR
##
##                         Df Deviance   AIC
## + SEGMENTO            6  2446.8 2466.8
## + DIASEM              1  2468.9 2478.9
## + HORA_AUX             1  2476.6 2486.6
## + FECHA_VIN           1  2489.5 2499.5
## + DIAMES               1  2490.7 2500.7
## + FECHA               1  2490.7 2500.7
## + EGRESOS              1  2492.6 2502.6
## + NROCIUDADES          1  2494.0 2504.0
## + CANAL                1  2495.8 2505.8
## + EDAD                 1  2496.6 2506.6
## + OFICINA_VIN          1  2497.2 2507.2
## <none>                  2499.2 2507.2
## + Dist_HOY              1  2498.4 2508.4
## + SEXO                  1  2498.6 2508.6
## + NROPAISES             1  2498.6 2508.6
## + Dist_Mean_INTER        1  2498.7 2508.7
## + INGRESOS              1  2498.8 2508.8
## - VALOR                  1  2538.9 2544.9
## - Dist_Mean_NAL          1  2540.9 2546.9
## - Canal1                 1  3224.7 3230.7
##
## Step: AIC=2466.85
## FRAUDE ~ Canal1 + Dist_Mean_NAL + VALOR + SEGMENTO
##
##                         Df Deviance   AIC
## + DIASEM              1  2419.6 2441.6
## + HORA_AUX             1  2425.0 2447.0
## + FECHA                1  2436.4 2458.4
## + DIAMES               1  2436.4 2458.4
## + FECHA_VIN            1  2438.0 2460.0
## + NROCIUDADES          1  2441.6 2463.6
## <none>                  2446.8 2466.8
## + Dist_HOY              1  2445.4 2467.4
## + OFICINA_VIN           1  2445.5 2467.5
## + CANAL                 1  2445.7 2467.7

```

```

## + NROPAISES      1  2445.8 2467.8
## + EDAD          1  2446.1 2468.1
## + INGRESOS      1  2446.1 2468.1
## + Dist_Mean_INTER 1  2446.2 2468.2
## + SEXO          1  2446.5 2468.5
## + EGRESOS       1  2446.6 2468.6
## - SEGMENTO      6   2499.2 2507.2
## - Dist_Mean_NAL 1  2491.6 2509.6
## - VALOR         1  2496.0 2514.0
## - Canal1        1  3020.4 3038.4
##
## Step: AIC=2441.56
## FRAUDE ~ Canal1 + Dist_Mean_NAL + VALOR + SEGMENTO + DIASEM
##
##             Df Deviance    AIC
## + HORA_AUX      1  2391.5 2415.5
## + FECHA_VIN     1  2410.0 2434.0
## + FECHA         1  2412.2 2436.2
## + DIAMES        1  2412.2 2436.2
## + NROCIUDADES   1  2413.1 2437.1
## + OFICINA_VIN    1  2417.0 2441.0
## <none>           2419.6 2441.6
## + Dist_HOY       1  2418.2 2442.2
## + NROPAISES     1  2418.4 2442.4
## + Dist_Mean_INTER 1  2418.5 2442.5
## + EDAD          1  2418.6 2442.6
## + INGRESOS      1  2418.7 2442.7
## + CANAL         1  2418.8 2442.8
## + EGRESOS       1  2419.1 2443.1
## + SEXO          1  2419.2 2443.2
## - DIASEM        1  2446.8 2466.8
## - SEGMENTO      6   2468.9 2478.9
## - VALOR         1  2461.7 2481.7
## - Dist_Mean_NAL 1  2463.9 2483.9
## - Canal1        1  2981.7 3001.7
##
## Step: AIC=2415.5
## FRAUDE ~ Canal1 + Dist_Mean_NAL + VALOR + SEGMENTO + DIASEM +
##      HORA_AUX
##
##             Df Deviance    AIC
## + FECHA_VIN     1  2382.5 2408.5
## + DIAMES        1  2384.7 2410.7
## + FECHA         1  2384.7 2410.7
## + NROCIUDADES   1  2386.0 2412.0
## + OFICINA_VIN    1  2388.4 2414.4
## <none>           2391.5 2415.5
## + Dist_HOY       1  2389.7 2415.7
## + INGRESOS      1  2390.2 2416.2
## + Dist_Mean_INTER 1  2390.3 2416.3
## + EDAD          1  2390.5 2416.5
## + NROPAISES     1  2390.8 2416.8
## + CANAL         1  2390.9 2416.9
## + EGRESOS       1  2391.1 2417.1
## + SEXO          1  2391.2 2417.2
## - HORA_AUX       1  2419.6 2441.6

```

```

## - DIASEM           1  2425.0 2447.0
## - SEGMENTO         6  2439.6 2451.6
## - Dist_Mean_NAL   1  2437.4 2459.4
## - VALOR            1  2439.0 2461.0
## - Canal1           1  2958.2 2980.2
##
## Step: AIC=2408.49
## FRAUDE ~ Canal1 + Dist_Mean_NAL + VALOR + SEGMENTO + DIASEM +
##      HORA_AUX + FECHA_VIN
##
##          Df Deviance    AIC
## + DIAMES           1  2376.5 2404.5
## + FECHA            1  2376.5 2404.5
## + NROCIUDADES     1  2376.9 2404.9
## + OFICINA_VIN      1  2379.7 2407.7
## <none>             2382.5 2408.5
## + Dist_HOY          1  2381.0 2409.0
## + INGRESOS          1  2381.1 2409.1
## + Dist_Mean_INTER   1  2381.2 2409.2
## + NROPAISES         1  2381.7 2409.7
## + SEXO              1  2381.9 2409.9
## + CANAL             1  2382.0 2410.0
## + EGRESOS           1  2382.2 2410.2
## + EDAD              1  2382.5 2410.5
## - FECHA_VIN         1  2391.5 2415.5
## - HORA_AUX          1  2410.0 2434.0
## - DIASEM            1  2416.8 2440.8
## - SEGMENTO          6  2429.8 2443.8
## - Dist_Mean_NAL     1  2421.7 2445.7
## - VALOR             1  2431.3 2455.3
## - Canal1            1  2903.7 2927.7
##
## Step: AIC=2404.48
## FRAUDE ~ Canal1 + Dist_Mean_NAL + VALOR + SEGMENTO + DIASEM +
##      HORA_AUX + FECHA_VIN + DIAMES
##
##          Df Deviance    AIC
## + NROCIUDADES     1  2371.2 2401.2
## + OFICINA_VIN      1  2374.0 2404.0
## <none>             2376.5 2404.5
## + INGRESOS          1  2375.1 2405.1
## + Dist_HOY          1  2375.2 2405.2
## + NROPAISES         1  2375.4 2405.4
## + CANAL             1  2375.6 2405.6
## + Dist_Mean_INTER   1  2375.7 2405.7
## + SEXO              1  2375.7 2405.7
## + EGRESOS           1  2376.3 2406.3
## + EDAD              1  2376.5 2406.5
## - DIAMES            1  2382.5 2408.5
## - FECHA_VIN         1  2384.7 2410.7
## - HORA_AUX          1  2403.6 2429.6
## - DIASEM            1  2407.4 2433.4
## - SEGMENTO          6  2425.1 2441.1
## - Dist_Mean_NAL     1  2419.4 2445.4
## - VALOR             1  2423.5 2449.5
## - Canal1            1  2903.2 2929.2

```

```

## Step: AIC=2401.17
## FRAUDE ~ Canal1 + Dist_Mean_NAL + VALOR + SEGMENTO + DIASEM +
##      HORA_AUX + FECHA_VIN + DIAMES + NROCIUDADES
##
##          Df Deviance    AIC
## + OFICINA_VIN     1  2368.7 2400.7
## + SEXO            1  2369.1 2401.1
## <none>           2371.2 2401.2
## + INGRESOS       1  2369.7 2401.7
## + Dist_HOY        1  2369.8 2401.8
## + NROPAISES      1  2370.1 2402.1
## + CANAL          1  2370.1 2402.1
## + Dist_Mean_INTER 1  2370.6 2402.6
## + EGRESOS         1  2371.0 2403.0
## + EDAD            1  2371.1 2403.1
## - NROCIUDADES    1  2376.5 2404.5
## - DIAMES          1  2376.9 2404.9
## - FECHA_VIN       1  2379.5 2407.5
## - HORA_AUX        1  2397.4 2425.4
## - DIASEM          1  2402.8 2430.8
## - SEGMENTO        6   2419.3 2437.3
## - Dist_Mean_NAL   1  2417.5 2445.5
## - VALOR           1  2419.5 2447.5
## - Canal1          1  2903.2 2931.2
##
## Step: AIC=2400.72
## FRAUDE ~ Canal1 + Dist_Mean_NAL + VALOR + SEGMENTO + DIASEM +
##      HORA_AUX + FECHA_VIN + DIAMES + NROCIUDADES + OFICINA_VIN
##
##          Df Deviance    AIC
## <none>           2368.7 2400.7
## + SEXO            1  2366.9 2400.9
## - OFICINA_VIN     1  2371.2 2401.2
## + INGRESOS       1  2367.3 2401.3
## + NROPAISES      1  2367.7 2401.7
## + Dist_HOY        1  2367.7 2401.7
## + Dist_Mean_INTER 1  2367.8 2401.8
## + CANAL          1  2367.9 2401.9
## + EGRESOS         1  2368.6 2402.6
## + EDAD            1  2368.6 2402.6
## - NROCIUDADES    1  2374.0 2404.0
## - DIAMES          1  2374.2 2404.2
## - FECHA_VIN       1  2376.7 2406.7
## - HORA_AUX        1  2395.5 2425.5
## - DIASEM          1  2401.7 2431.7
## - SEGMENTO        6   2416.2 2436.2
## - Dist_Mean_NAL   1  2416.4 2446.4
## - VALOR           1  2417.2 2447.2
## - Canal1          1  2898.2 2928.2

```

Según el modelo AIC el mejor modelo es el que considera las siguientes variables predictoras:

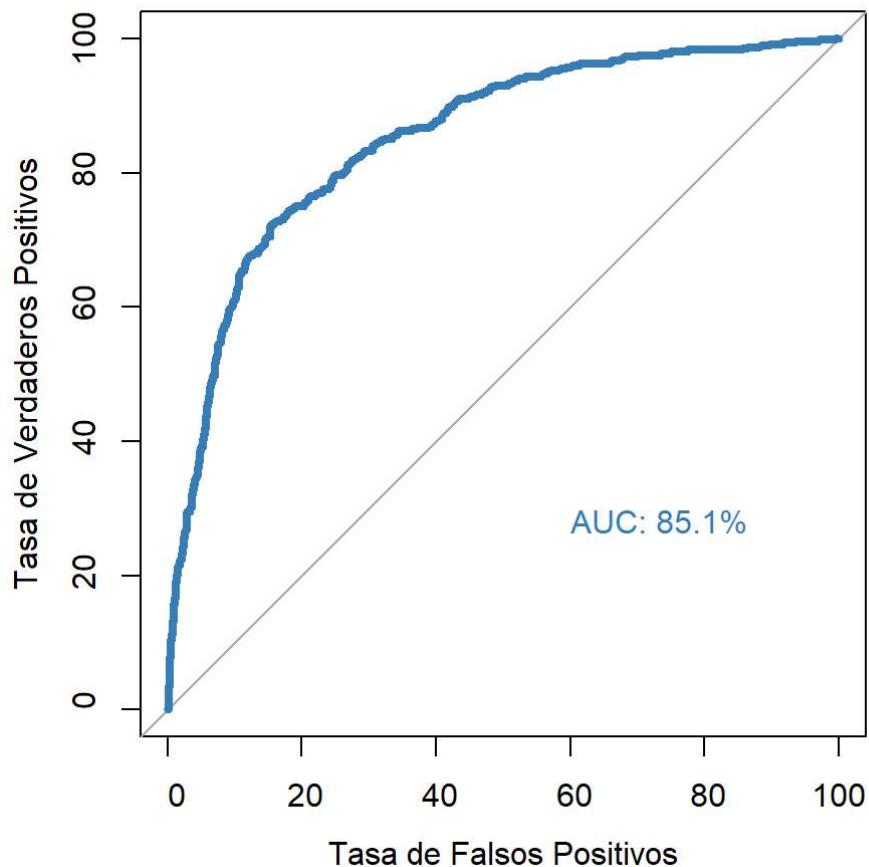
FRAUDE ~ Canal1 + Dist_Mean_NAL + VALOR + SEGMENTO + DIASEM + HORA_AUX + FECHA_VIN + FECHA + NROCIUDADES + OFICINA_VIN

A continuación muestro la curva ROC y su métrica AUC.

```
par(pty = "s") # para que el gráfico sea cuadrado  
roc(df$FRAUDE,step$fitted.values, plot=TRUE, legacy.axes=TRUE, percent=TRUE, xlab="Tasa de Falsos Positivos", ylab="Tasa de Verdaderos Positivos", col="#377eb8", lwd=4, print.auc = TRUE, print.auc.x=40,print.auc.y=30 )
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```



```
##  
## Call:  
## roc.default(response = df$FRAUDE, predictor = step$fitted.values, percent = TRUE, plot = TRUE, legacy.axes = TRUE, xlab = "Tasa de Falsos Positivos", ylab = "Tasa de Verdaderos Positivos", col = "#377eb8", lwd = 4, print.auc = TRUE, print.auc.x = 40, print.auc.y = 30)  
##  
## Data: step$fitted.values in 2234 controls (df$FRAUDE 0) < 731 cases (df$FRAUDE 1).  
## Area under the curve: 85.13%
```

El siguiente paso es predecir utilizando el modelo. Como en el caso anterior, defino el umbral en 0.5.

```
y_pred <- as.numeric(predict(step, subset(df,select = -FRAUDE ), type="response")>.5)
y <- as.numeric(df$FRAUDE)
y_pred <- as.factor(y_pred)
y <- as.factor(y)
head(y_pred) # primeras 6 clases predichas
```

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

```
#primeras 6 clases reales
head(y)
```

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

El modelo acierta en las primeras 5 transacciones del dataframe.

Modelos restringidos Ridge, Lasso y Elastic El objetivo es presentar estos tres tipos de modelos. Estas regularizaciones tienen la ventaja de añadir sesgo al modelo de forma que disminuyen la varianza de los modelos a la hora de hacer predicciones.

Creo las matrices de predictores y variable objetivo:

```
X <- data.matrix(subset(df , select= - FRAUDE))
y <- as.double(as.matrix(df$FRAUDE ))
str(X)
```

```
##  num [1:2965, 1:19] 0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "dimnames")=List of 2
##    ..$ : chr [1:2965] "1" "2" "3" "4" ...
##    ..$ : chr [1:19] "VALOR" "HORA_AUX" "Canal1" "FECHA" ...
```

Creo un conjunto de entrenamiento y otro de test

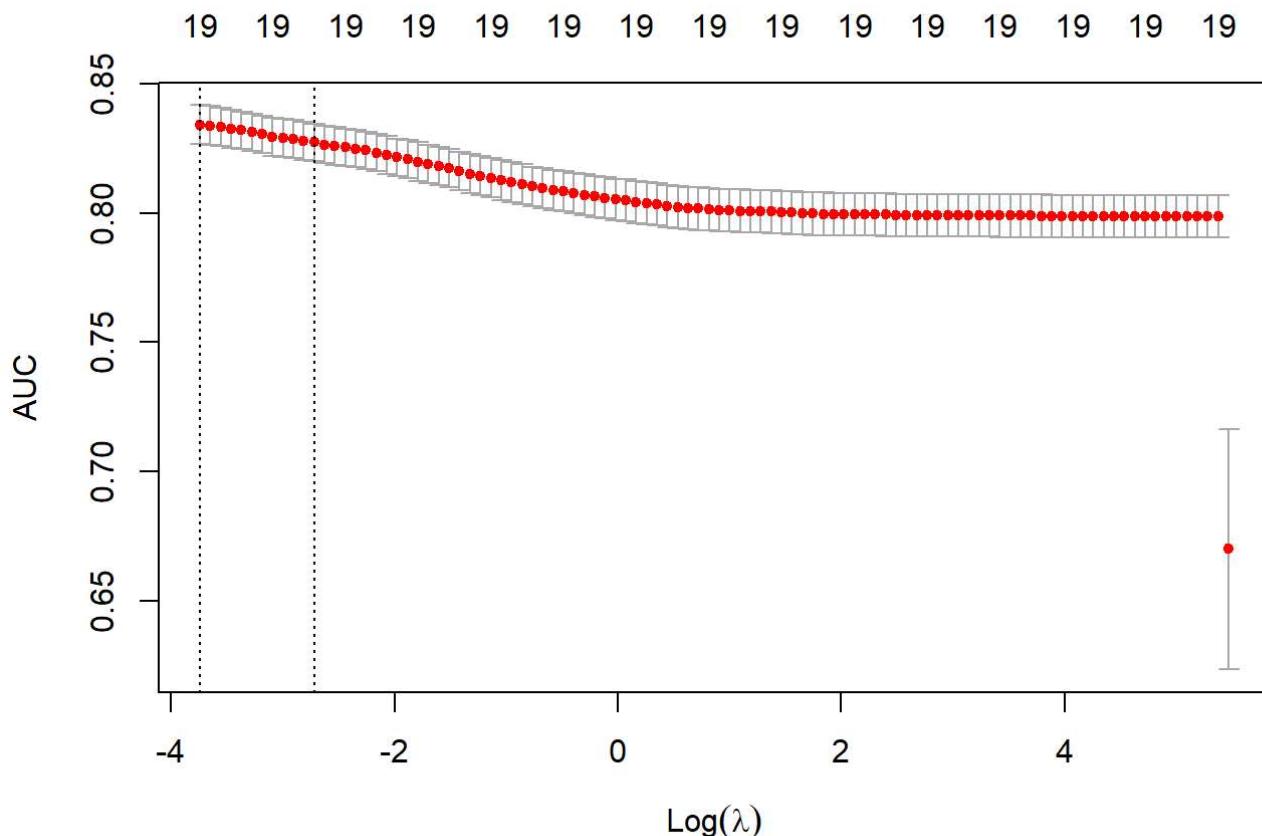
```
dim(X)
```

```
## [1] 2965 19
```

```
X_train <- X[1:2372,]
y_train <- y[1:2372]
X_test <- X[2372:2965,]
y_test <- y[2372:2965]
```

RIDGE

```
set.seed(5)#semilla
cv.ridge <- cv.glmnet(X_train, y_train, family='binomial', alpha=0, type.measure='auc')
# Resultados
plot(cv.ridge)
```



```
#el mejor valor de Lambda:  
cv.ridge$lambda.min
```

```
## [1] 0.02368716
```

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

```
## [1] 0.8340125
```

Muestro los coeficientes de Ridge

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

```

## 20 x 1 sparse Matrix of class "dgCMatrix"
##
## (Intercept) -8.308676e+04
## VALOR        2.387866e-07
## HORA_AUX     -2.726565e-02
## Canal1       -2.066043e+00
## FECHA        4.119394e-03
## CANAL        -1.923325e-01
## DIASEM        1.032465e-01
## DIAMES        4.067958e-03
## FECHA_VIN    4.060006e-06
## OFICINA_VIN   -3.187222e-04
## SEXO          -1.769605e-01
## SEGMENTO      4.183958e-02
## EDAD          -6.519445e-04
## INGRESOS      3.724301e-10
## EGRESOS       -2.193164e-09
## NROPAISES     -1.151311e-02
## Dist_Mean_INTER -2.127349e-05
## NROCIUDADES   2.650338e-02
## Dist_Mean_NAL  1.459463e-03
## Dist_HOY       1.266828e-05

```

Las métricas del modelo Ridge:

```

y_pred <- as.numeric(predict.glmnet(cv.ridge$glmnet.fit, newx=X_test, s=cv.ridge$lambda.min))
.5) # fijamos el umbral en 0.5
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
##           0 457  22
##           1  83  32
##
##                 Accuracy : 0.8232
##                   95% CI : (0.7901, 0.8531)
##       No Information Rate : 0.9091
##     P-Value [Acc > NIR] : 1
##
##                 Kappa : 0.291
##
## McNemar's Test P-Value : 4.759e-09
##
##                 Sensitivity : 0.8463
##                 Specificity  : 0.5926
##      Pos Pred Value : 0.9541
##      Neg Pred Value : 0.2783
##                 Precision  : 0.9541
##                 Recall    : 0.8463
##                  F1 : 0.8970
##                 Prevalence : 0.9091
##      Detection Rate : 0.7694
## Detection Prevalence : 0.8064
##   Balanced Accuracy : 0.7194
##
## 'Positive' Class : 0
##

```

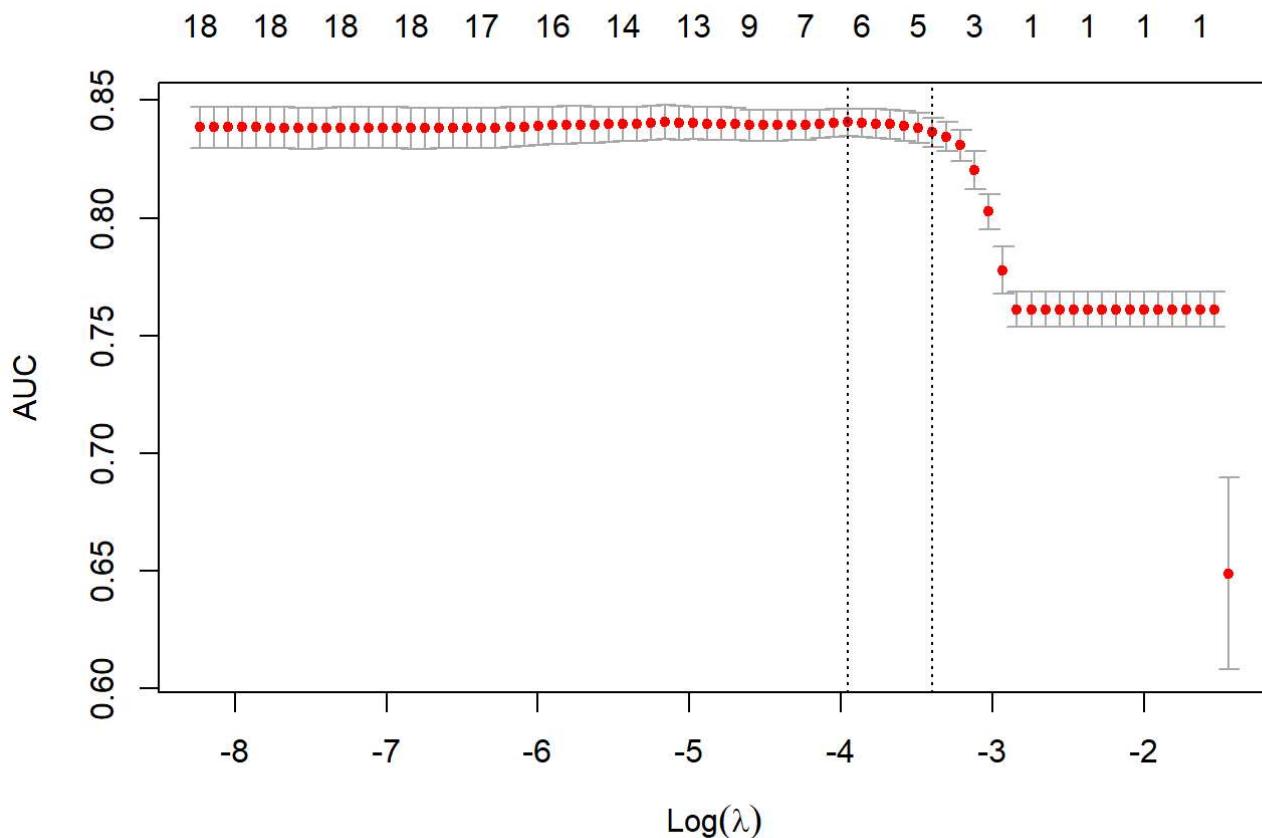
Se puede observar un Accuracy de 0.8215

LASSO

```

set.seed(5)
cv.lasso <- cv.glmnet(X_train, y_train, family='binomial', alpha=1, type.measure='auc')
# Resultados
plot(cv.lasso)

```



```
#mejor valor de Lambda
cv.lasso$lambda.min
```

```
## [1] 0.01921338
```

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

```
## [1] 0.8407111
```

los coeficientes son los siguientes:

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

```

## 20 x 1 sparse Matrix of class "dgCMatrix"
##
## (Intercept) -3.465408e+01
## VALOR        1.564950e-07
## HORA_AUX     -9.347483e-03
## Canal1       -2.379647e+00
## FECHA        .
## CANAL        .
## DIASEM       4.386000e-02
## DIAMES        .
## FECHA_VIN   1.861772e-06
## OFICINA_VIN .
## SEXO          .
## SEGMENTO     .
## EDAD          .
## INGRESOS     .
## EGRESOS      .
## NROPAISES    .
## Dist_Mean_INTER .
## NROCIUDADES .
## Dist_Mean_NAL  9.431200e-04
## Dist_HOY      .

```

Las métricas:

```

y_pred <- as.numeric(predict.glmnet(cv.lasso$glmnet.fit, newx=X_test, s=cv.lasso$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
##           0 455  24
##           1  81  34
##
##                 Accuracy : 0.8232
##                   95% CI : (0.7901, 0.8531)
##       No Information Rate : 0.9024
##     P-Value [Acc > NIR] : 1
##
##                 Kappa : 0.3025
##
## McNemar's Test P-Value : 4.628e-08
##
##                 Sensitivity : 0.8489
##                 Specificity  : 0.5862
##      Pos Pred Value : 0.9499
##      Neg Pred Value : 0.2957
##                 Precision  : 0.9499
##                 Recall    : 0.8489
##                 F1       : 0.8966
##                 Prevalence : 0.9024
##      Detection Rate : 0.7660
## Detection Prevalence : 0.8064
##   Balanced Accuracy : 0.7175
##
## 'Positive' Class : 0
##

```

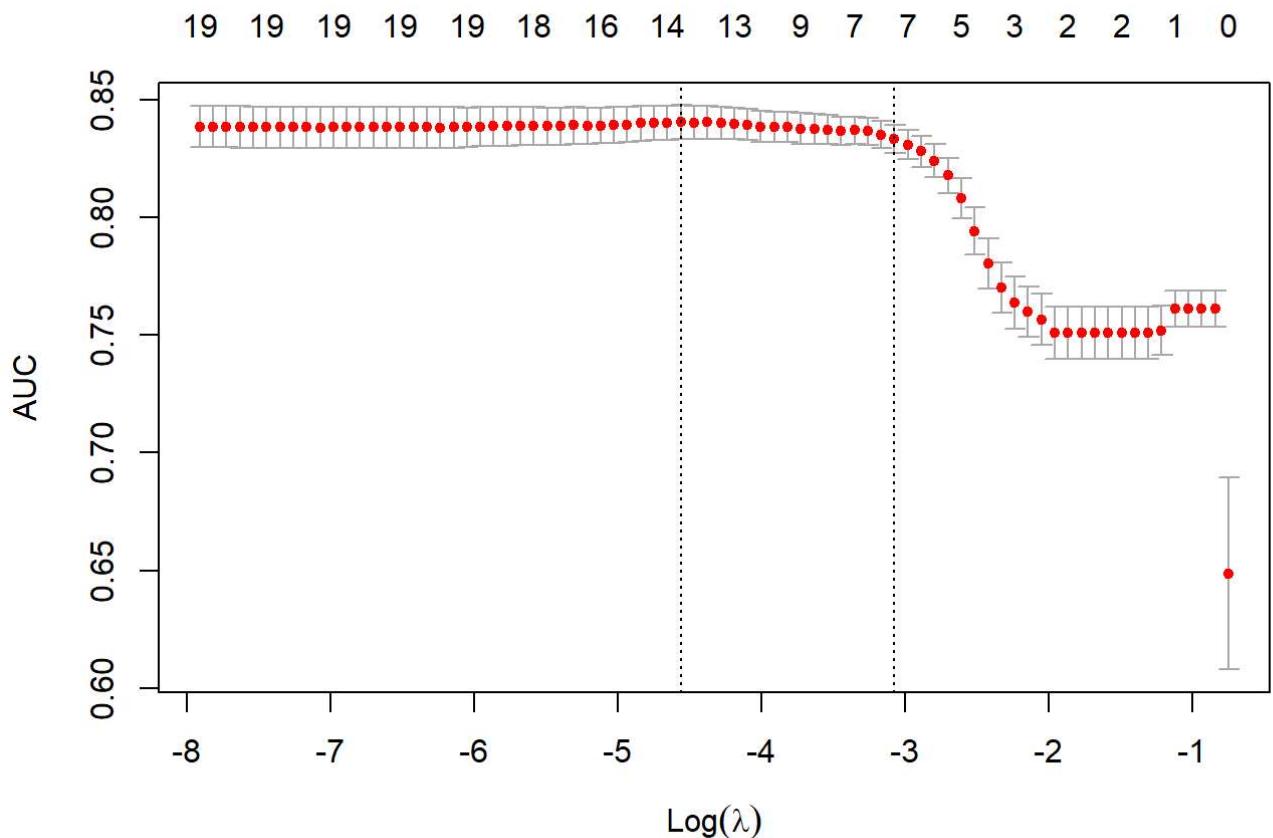
Accuracy : 0.8266

ELASTIC NET

```

set.seed(5)
cv.elastic <- cv.glmnet(X_train, y_train, family='binomial', alpha=0.5, type.measure='auc')
# Resultados
plot(cv.elastic)

```



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

```
## [1] 0.01044666
```

```
#valor del error que se estima para ese valor Lambda mínimo dado en AUC
max(cv.elastic$cvm)
```

```
## [1] 0.8404193
```

Coeficientes:

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

```

## 20 x 1 sparse Matrix of class "dgCMatrix"
##                                     s1
## (Intercept) -8.184305e+04
## VALOR        2.420363e-07
## HORA_AUX    -2.584295e-02
## Canal1      -2.504716e+00
## FECHA       4.058326e-03
## CANAL        .
## DIASEM       9.557490e-02
## DIAMES       2.944096e-03
## FECHA_VIN   3.411738e-06
## OFICINA_VIN -2.063714e-04
## SEXO         -9.633005e-02
## SEGMENTO     3.316026e-02
## EDAD         .
## INGRESOS     .
## EGRESOS      -7.767106e-10
## NROPAISES    .
## Dist_Mean_INTER .
## NROCIUDADES  2.040721e-02
## Dist_Mean_NAL  1.462851e-03
## Dist_HOY      2.949478e-06

```

Métricas:

```

y_pred <- as.numeric(predict.glmnet(cv.elastic$glmnet.fit, newx=X_test, s=cv.elastic$lambda.in)>.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
##           0 455  24
##           1  79  36
##
##                 Accuracy : 0.8266
##                   95% CI : (0.7937, 0.8562)
##   No Information Rate : 0.899
## P-Value [Acc > NIR] : 1
##
##                 Kappa : 0.3213
##
## Mcnemar's Test P-Value : 1.033e-07
##
##                 Sensitivity : 0.8521
##                 Specificity  : 0.6000
##   Pos Pred Value : 0.9499
##   Neg Pred Value : 0.3130
##             Precision : 0.9499
##             Recall    : 0.8521
##             F1       : 0.8983
##             Prevalence : 0.8990
##   Detection Rate : 0.7660
## Detection Prevalence : 0.8064
## Balanced Accuracy : 0.7260
##
## 'Positive' Class : 0
```

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

```

## Confusion Matrix and Statistics
##
##             Reference
## Prediction    0    1
##           0 455  24
##           1  79  36
##
##                 Accuracy : 0.8266
##                   95% CI : (0.7937, 0.8562)
##   No Information Rate : 0.899
## P-Value [Acc > NIR] : 1
##
##                 Kappa : 0.3213
##
## McNemar's Test P-Value : 1.033e-07
##
##                 Sensitivity : 0.8521
##                 Specificity  : 0.6000
##   Pos Pred Value : 0.9499
##   Neg Pred Value : 0.3130
##                 Precision  : 0.9499
##                 Recall    : 0.8521
##                 F1       : 0.8983
##                 Prevalence : 0.8990
##   Detection Rate : 0.7660
## Detection Prevalence : 0.8064
##   Balanced Accuracy : 0.7260
##
## 'Positive' Class : 0
##

```

Accuracy : 0.8283

CONCLUSIONES Entre los modelos mod y step el modelo con el mejor AUC es el realizado mediante stepwise.

Luego, el mejor modelo regularizado es el Elactic net.