CRISP-DM E2 - Entendimiento de la data

amanosalva

19 de noviembre de 2018

Descripción del problea

Los modelos analíticos para el manejo de los seguros de accidentes se están usando por muchas instituciones y están dando resultados exitosos en todo el mundo. Los modelos analíticos de se pueden definir como un conjunto de métodos y técnicas cuantitativas usados para predecir la probabilidad de que un cliente falle (Sea siniestroso) y en consecuencia no se recupere el rédito otorgado por alguna institución.

Reto

Identifique los clientes que tienen una alta probabilidad de siniestro.

1. Instalación y activación de librerías necesarias:

```
#install.packages('DataExplorer', dependencies = T)
library(mlr)
library(dplyr)
library(DataExplorer)
```

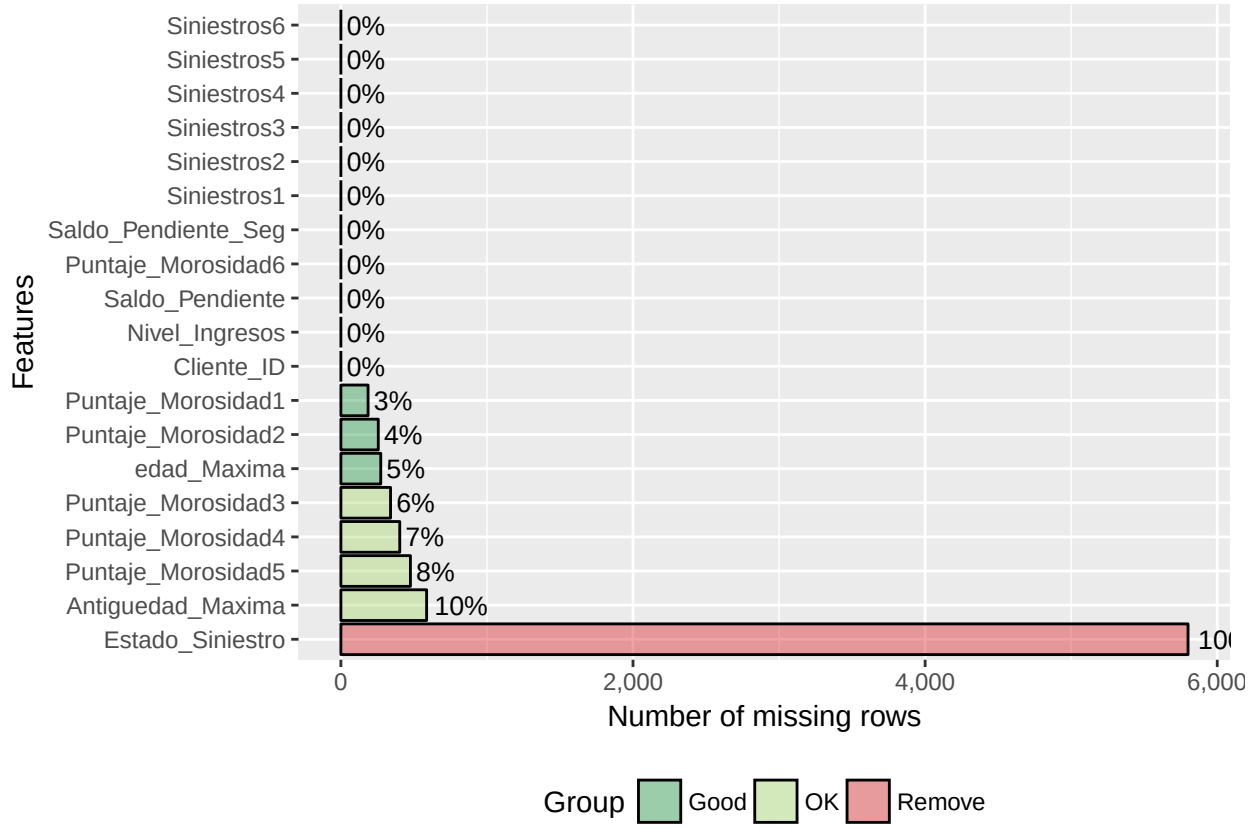
2. Ingesta de la data

```
train <- read.csv("train_seguros.csv", header = T)
```

3. Análisis exploratorio

Graficando los valores faltantes para una mejor visión global:

```
PlotMissing(train)
```



En caso de querer saber los valores exactos de los NA:

```
summarizeColumns(train)
```

```
##                   name    type   na         mean         disp     median
## 1           Cliente_ID integer    0 3.967334e+05 2.302859e+05 403865.000
## 2    Antiguedad_Maxima integer  587 4.984711e+01 4.705053e+01     34.000
## 3          edad_Maxima integer  273 1.071286e+00 1.158293e+00      1.000
## 4       Nivel_Ingresos numeric    0 3.240984e+03 6.157723e+03   1100.315
## 5      Saldo_Pendiente integer    0 1.993103e-01 6.790026e-01      0.000
## 6   Puntaje_Morosidad1 integer  186 2.434984e-01 8.912083e-01      0.000
## 7   Puntaje_Morosidad2 integer  256 2.202381e-01 8.424607e-01      0.000
## 8   Puntaje_Morosidad3 integer  340 1.935897e-01 7.810702e-01      0.000
## 9   Puntaje_Morosidad4 integer  403 1.854734e-01 7.443064e-01      0.000
## 10  Puntaje_Morosidad5 integer  476 1.722389e-01 7.073136e-01      0.000
## 11  Puntaje_Morosidad6 numeric    0 4.970516e+03 2.201641e+04      0.000
## 12 Saldo_Pendiente_Seg integer    0 7.837931e-01 4.890399e+00      0.000
## 13         Siniestros1 integer    0 6.960345e-01 4.746389e+00      0.000
## 14         Siniestros2 integer    0 6.458621e-01 4.661810e+00      0.000
## 15         Siniestros3 integer    0 5.943103e-01 4.575560e+00      0.000
## 16         Siniestros4 integer    0 5.456897e-01 4.484610e+00      0.000
## 17         Siniestros5 integer    0 5.044828e-01 4.401709e+00      0.000
## 18         Siniestros6  factor    0           NA 2.696552e-01         NA
## 19    Estado_Siniestro logical 5800           NA          NaN         NA
##            mad min      max nlevs
## 1  300364.3818 185 790771.0     0
## 2      37.0650   0    255.0     0
```

```
## 3       1.4826   0      5.0     0
## 4    1453.3335   0 124102.1     0
## 5       0.0000   0      6.0     0
## 6       0.0000   0      7.0     0
## 7       0.0000   0      7.0     0
## 8       0.0000   0      7.0     0
## 9       0.0000   0      7.0     0
## 10      0.0000   0      7.0     0
## 11      0.0000   0 442334.8     0
## 12      0.0000   0    289.0     0
## 13      0.0000   0    289.0     0
## 14      0.0000   0    289.0     0
## 15      0.0000   0    289.0     0
## 16      0.0000   0    289.0     0
## 17      0.0000   0    288.0     0
## 18          NA 496   4236.0     3
## 19          NA Inf     -Inf     0
```

Las siguientes columnas tienen valores NA: Antiguedad_Maxima - 587 edad_Maxima - 273 Puntaje_MorosidadX, donde 1 <= x <= 5

Viendo la estructura de la data

```
PlotStr(train)
```

Curiosamente vemos que la variable “Siniestro6” tiene tres niveles: "“,”si“,”no". Ese espacio en blanco debe ser considerado como ata faltante? Habría que ver su proporción o influencia.

```
levels(train$Siniestros6)
```

```
## [1] ""   "no" "si"
```

Distribuciones

Viendo las distribuciones numéricas:

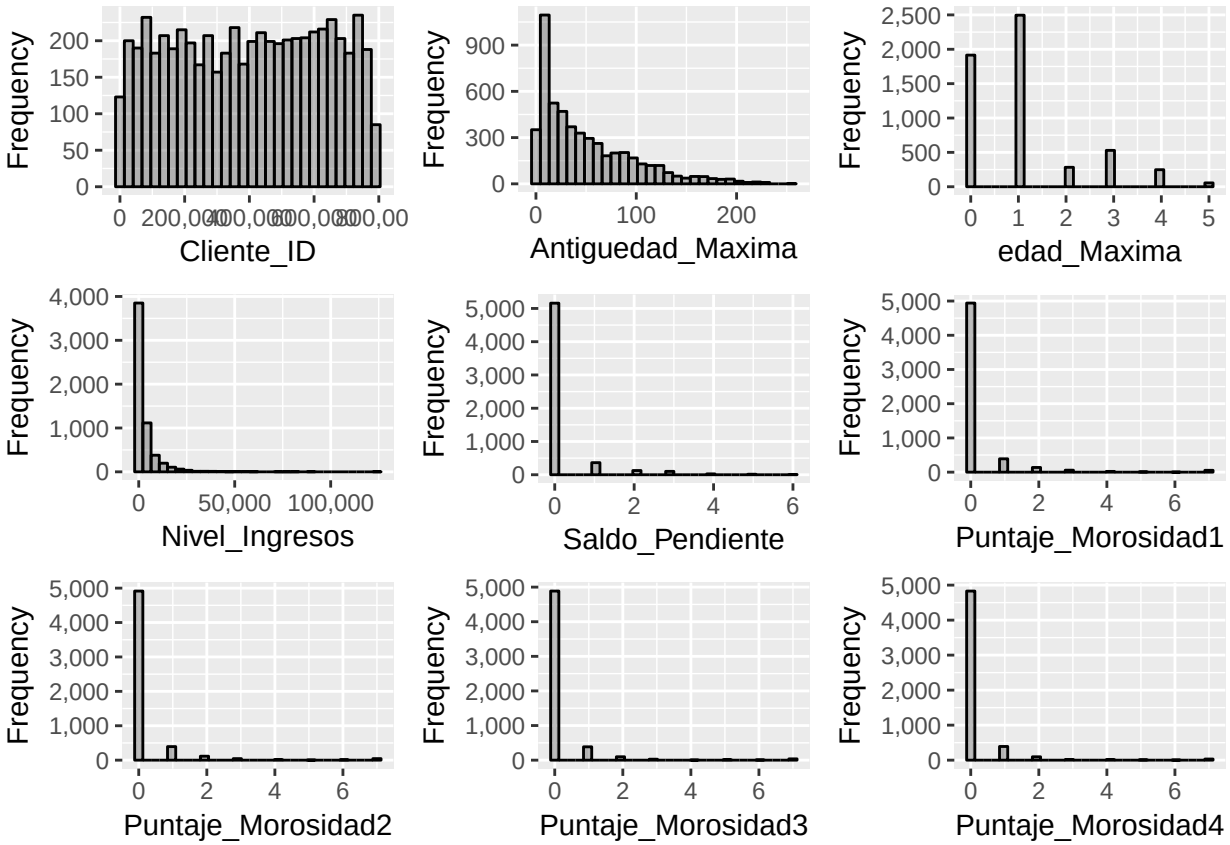
Separamos solo los numéricos

```
train_numericos <- select_if(train, is.numeric)
names(train_numericos)
```

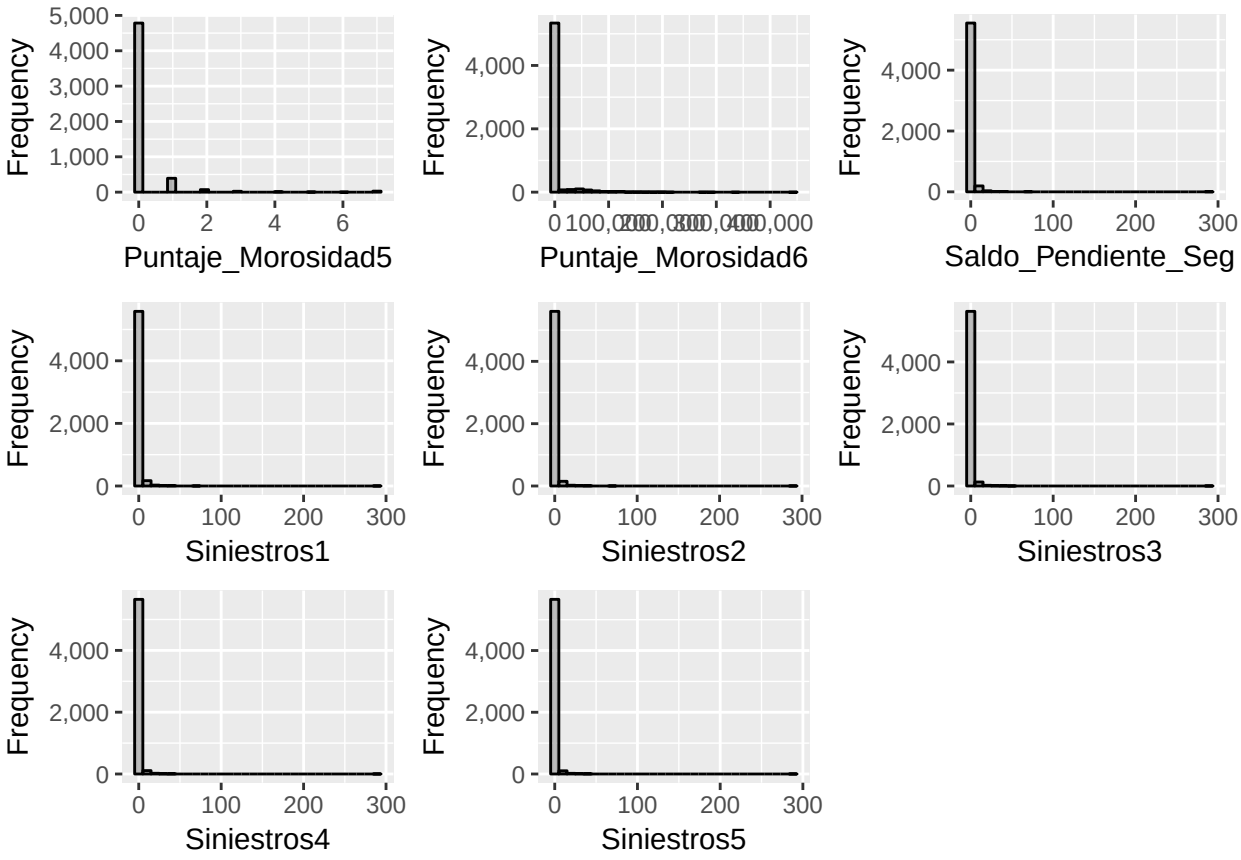
```
##  [1] "Cliente_ID"          "Antiguedad_Maxima"   "edad_Maxima"
##  [4] "Nivel_Ingresos"      "Saldo_Pendiente"     "Puntaje_Morosidad1"
##  [7] "Puntaje_Morosidad2"  "Puntaje_Morosidad3"  "Puntaje_Morosidad4"
## [10] "Puntaje_Morosidad5"  "Puntaje_Morosidad6"  "Saldo_Pendiente_Seg"
## [13] "Siniestros1"         "Siniestros2"         "Siniestros3"
## [16] "Siniestros4"         "Siniestros5"
```

Graficamos histogramas:

```
HistogramContinuous(train_numericos[1:9])
```




```
HistogramContinuous(train_numericos[10:length(train_numericos)])
```



Viendo las distribuciones categóricas:

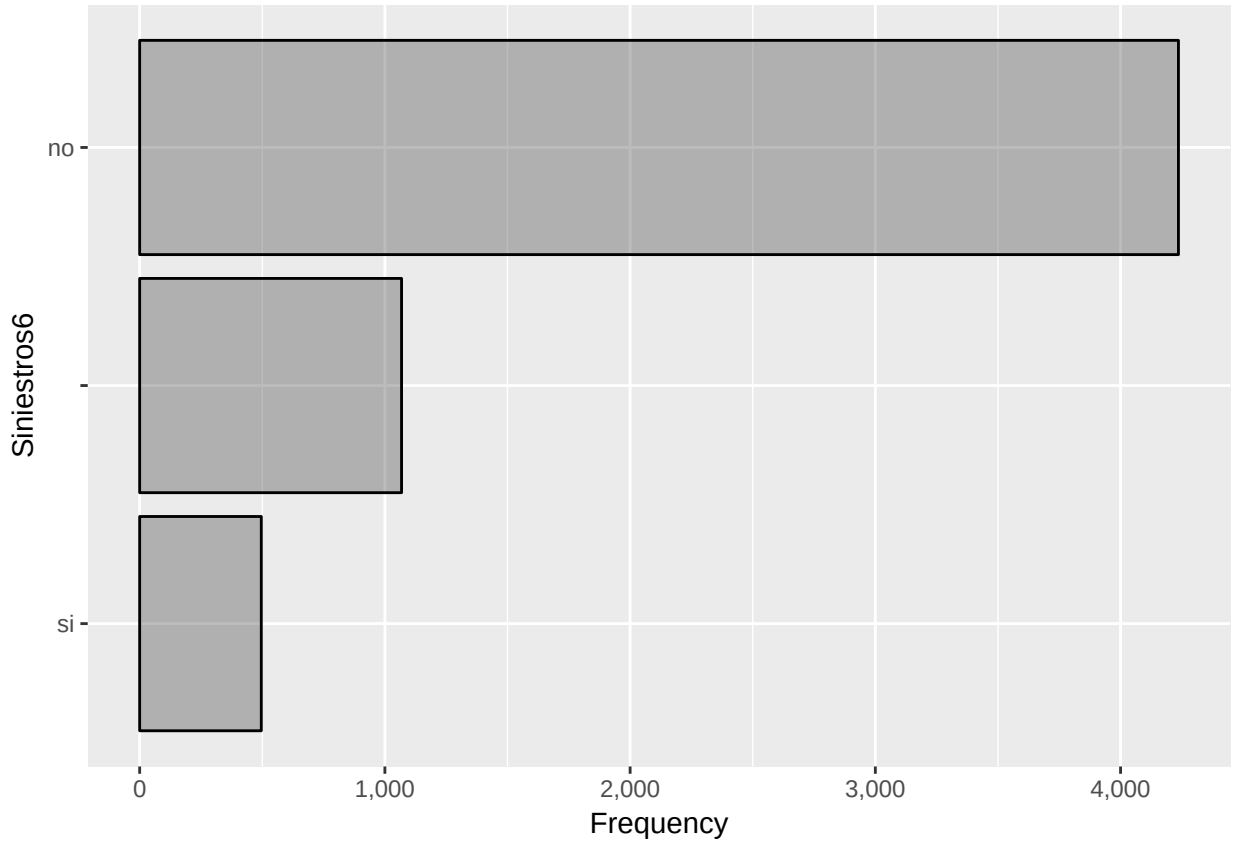
Separamos solo categóricos

```
train_factores <- select_if(train, is.factor)
names(train_factores)
```

```
## [1] "Siniestros6"
```

Graficamos barras:

```
BarDiscrete(train_factores)
```

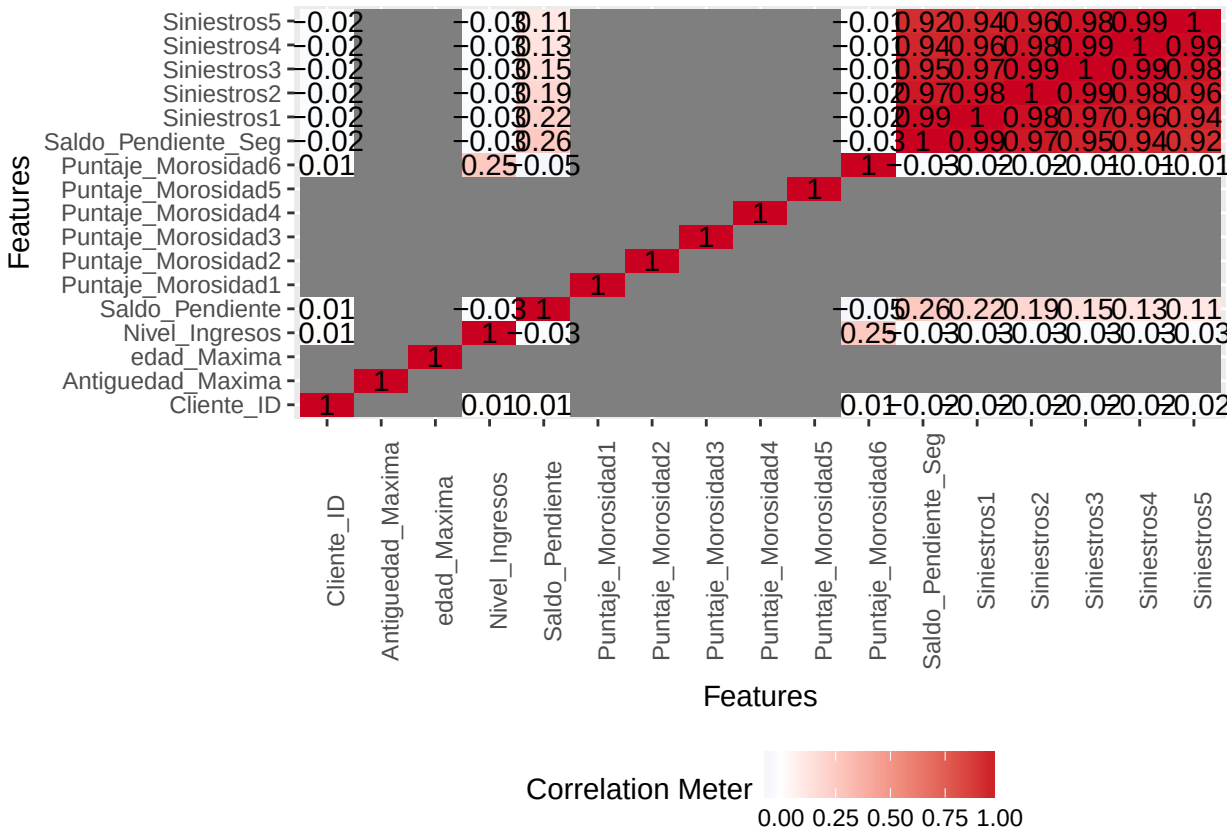


Correlaciones

Viendo las correlaciones continuas:

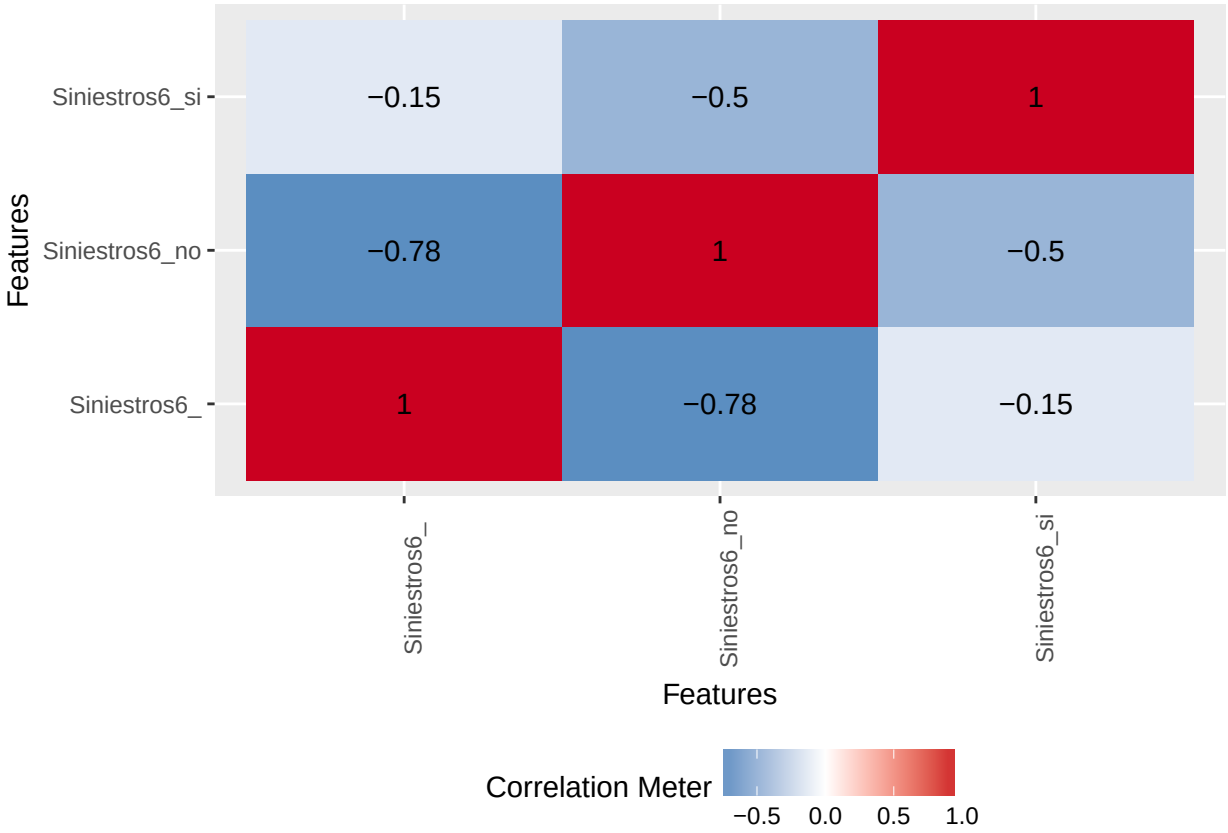
```
CorrelationContinuous(train_numericos)
```

```
## Warning: Removed 182 rows containing missing values (geom_text).
```



Viendo las correlaciones discretas:

```
CorrelationDiscrete(train_factores)
```


Siniestros6_si
Siniestros6_no
Siniestros6_
Features
−0.15
−0.5
1
−0.78
1
−0.5
1
−0.78
−0.15
Siniestros6_
Siniestros6_no
Siniestros6_si
Features
Correlation Meter
−0.5
0.0
0.5
1.0