Examen I

Efrén Jiménez

28 de octubre de 2016

1

Ciencia de Datos: Es la disciplina que se encarga de los procesos para la extracción de conocimiento en grandes volúmenes de datos.

Minería de Datos: Es una disciplina que une los campos de la estadística y las ciencias de la computación para obtener conocimiento de los grandes volúmenes de datos.

Variable Cuantitativa: Es una variable que se puede expresar por medio de un numero puede ser entero o flotante

Variable Cualitativa: Es una variable que se refiere a una característica o atributo para que sea medida por números.

Estadística Descriptiva: Es una técnica en el ámbito matemático que intenta obtener, organizar, presentar y describir un conjunto de datos dados para su estudio.

Estadística Inferencial: Es una técnica en el ámbito matemático que intenta por medio de métodos y procedimientos determinar propiedades de una población, por medio de una muestra pequeña de los mismos.

Prueba de Hipótesis: Es una prueba estadistica que se utiliza para determinar si existe evidencia en una muestra de datos para inferir una condicion "X"" para una poblacion.

P-value: El valor p que se utiliza para definir una probabilidad contra una hipótesis nula con un valor que oscila entre el 0 y 1

Correlación: Indica la fuerza y la dirección en una relación lineal y su proporcionalidad entre las variables estadísticas.

Curva ROC: Es una representación grafica de la sensibilidad contra la especificidad para un sistema que clasifica en binario en un umbral de discriminación.

2

Análisis del Problema

Este famoso conjunto de datos del iris (Fisher's or Anderson's) da las medidas en centímetros de las variables longitud y ancho del sepal y longitud y ancho de los pétalos, respectivamente, para 50 flores de cada una de 3 especies de iris. Las especies son Iris setosa, versicolor y virginica.

¿Cuál es la principal conclusión que se puede sacar a partir del gráfico en el el conjunto de datos iris?

Entendimiento de los Datos

```
Dominio             Descripción
1. Sepal.Length:    Largo de sepa
2. Sepal.Width:     Ancho de sepa
3. Petal.Length:    Largo de sepa
4. Petal.Width:     Ancho del pétalo
5. Species:         (setosa,virginica,versicolor)
```

Exploración de los Datos

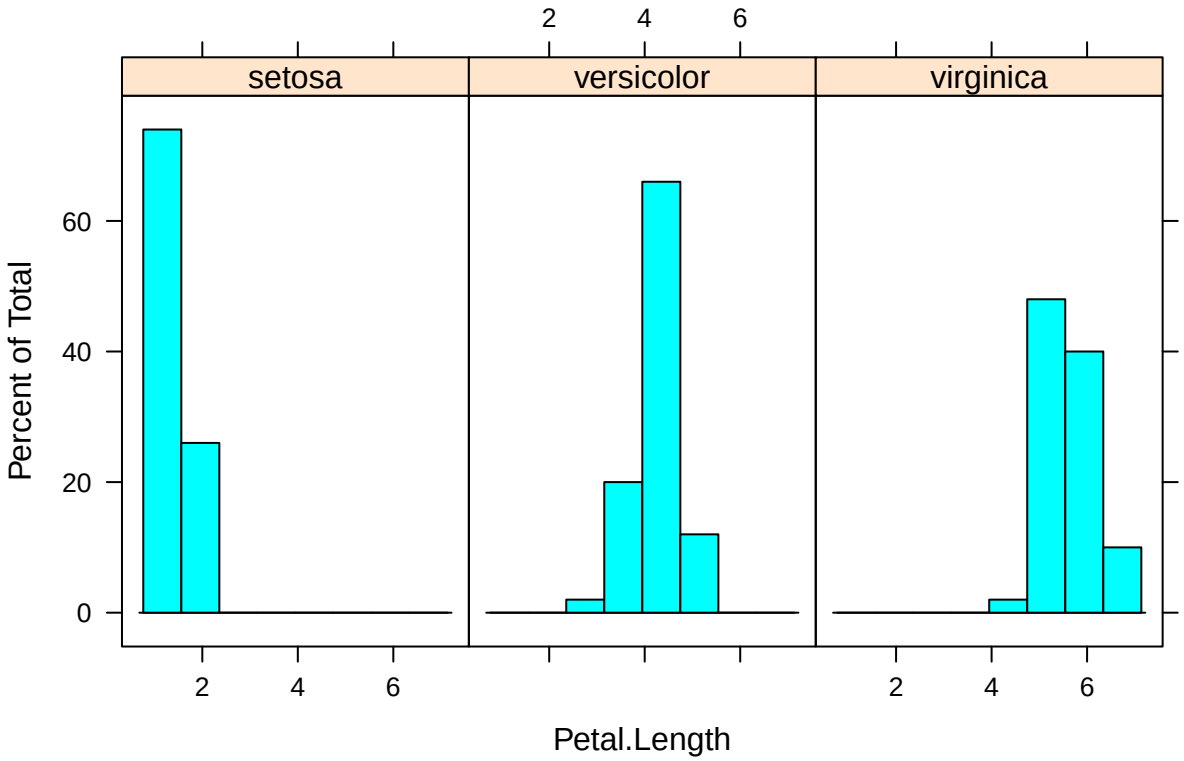
```
# librerías utiliadas
library(lattice)

# La estructura del conjunto de datos:
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Si analizamos con mayor detalle la variable Pedal.Length, su distribución entre la variable Species muestra que:

- La especie setosa se ditribuye mayormente dentro de 1 pedal al rededor del 80% y menormente entre el pedal 2 con alrededor del 20%.
- La especie versicolor se ditribuye mayormente dentro de los 4 y 5 pedales y alrededor de un 20% entre los 3 pedales.
- La especie virginica se ditribuye mayormente dentro de los 5 y 6 pedales un y al rededor del 15% entre los 7 pedales.

```
histogram(~Petal.Length | Species, data = iris, main = "Distribución de la variable Pedal.Length por la
```



Resultados

Los resueltados obtenidos desmuestran como las especies estan muy ligada a la cantidad de los petalos,ademas la distrubucion final fue la siguiente:

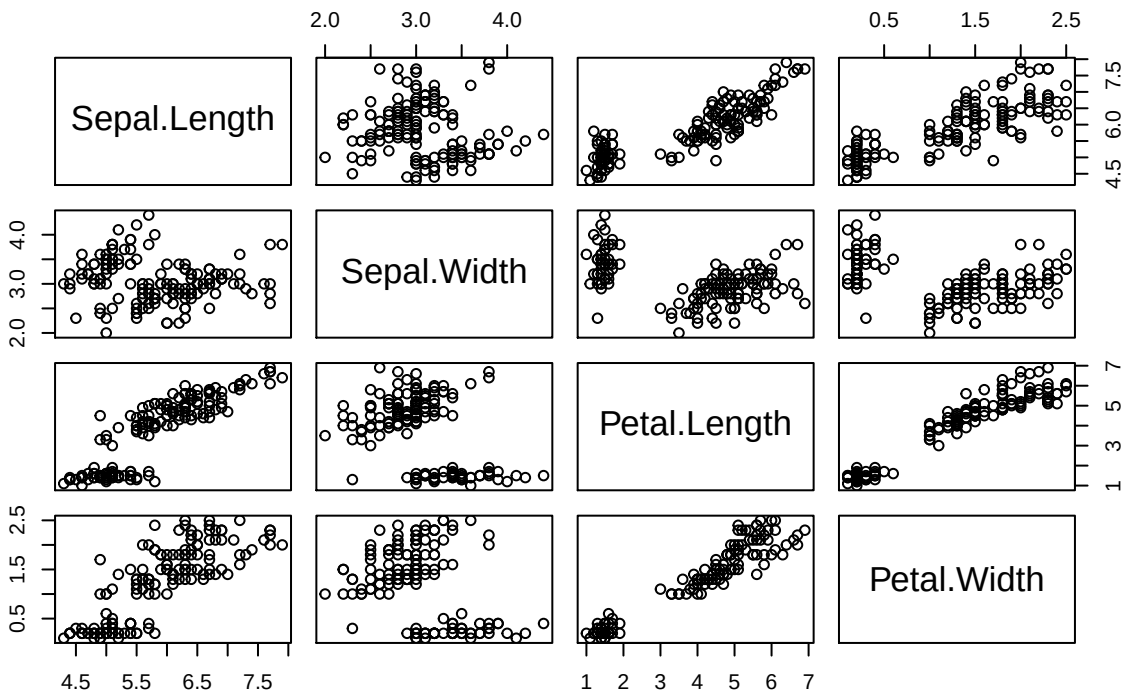
```
Setosa : Pétalos entre 1 y 2 pétalos.
Versicolor: Pétalos entre 3 y 5 pétalos.
Virginica: Pétalos entre 5 y 7 pétalos.
```

3

En esta matriz de correlacion podemos observar 3 relacion bastante fuertes entre las variables **Petal.Length** y **Petal.Width** con una correlacion de **0.96**, en la segunda relacion podemos encontrar un factor de coorelacion de **0.87** entre la variable **Sepal.Length** y **Petal.Length** y como tercera relacion podemos encontra con un factor de coorelacion de **0.81** entre las variables **Sepal.Length** y **Petal.Width**.

```
pairs(iris[, c(1, 2, 3, 4)], main = "Correlación de las variables Sepal.Length,Sepal.Width,Petal.Length
```

elación de las variables Sepal.Length,Sepal.Width,Petal.Length,Petal.W




```
cor(iris[, c(1, 2, 3, 4)])
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
## Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
## Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
## Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

4

Análisis del Problema

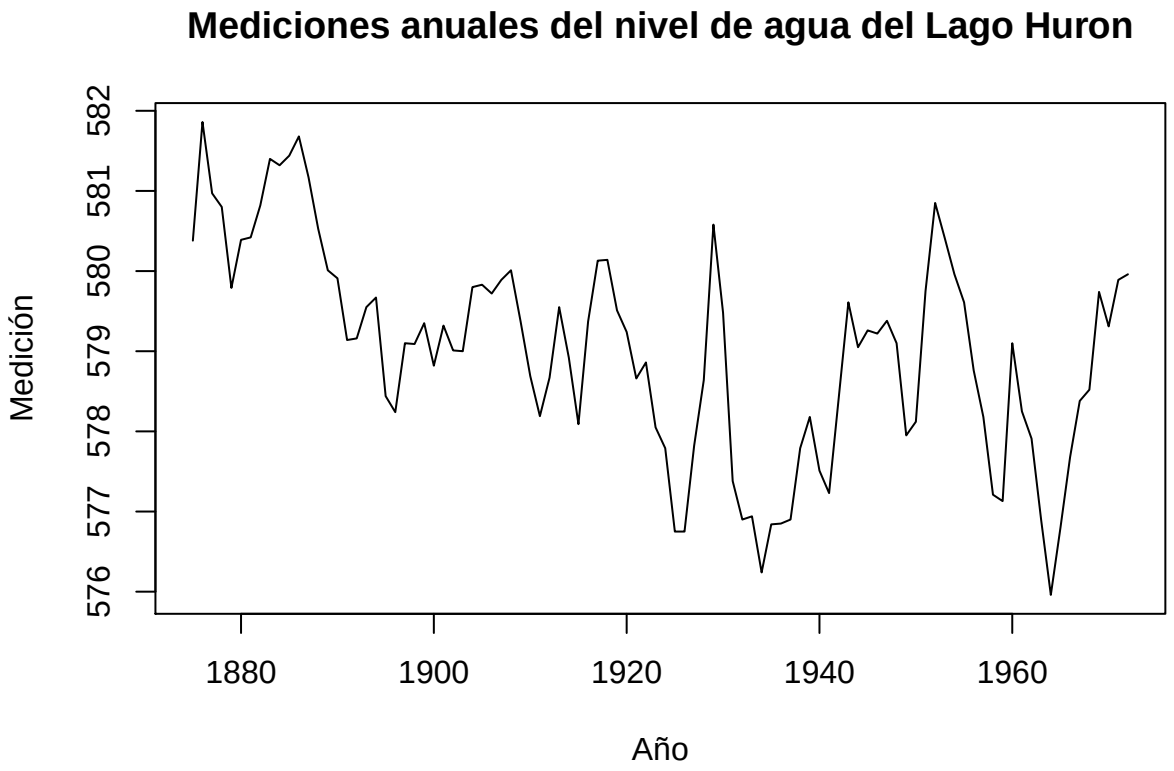
El nivel de agua de ríos, lagos y embalses se usa directamente para la predicción de crecidas, para la delimitación de zonas con riesgo de inundación y para el diseño de estructuras en cursos o masas de agua o cerca de ellas. Cuando se relaciona con los caudales de las corrientes o con el volumen de almacenamiento de embalses y lagos, el nivel de agua se utiliza como base para determinar el caudal o el volumen de agua almacenada.

Mediciones anuales del nivel, en pies, del lago Huron 1875-1972. Una serie temporal con 98 observaciones

Resultados

Los resultados obtenidos se observa que en el bloque de años de 1880 a 1900 fueron los mejores años en nivel de pies de agua en el lago Huron, luego existió una recaída importante hasta alrededor de 1930 y luego un aumento considerable antes de llegar a 1940 donde se puede observar la segunda peor recaída en el nivel de agua histórico, en el cual también se puede ver cerca de los 1960 la peor recaída con una medición de 576.0 y luego un aumento considerable hasta 1972.

```r
plot(LakeHuron, main = "Mediciones anuales del nivel de agua del Lago Huron",
    xlab = "Año", ylab = "Medición")
```




```r
summary(LakeHuron)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   576.0   578.1   579.1   579.0   579.9   581.9
```

5

Análisis del Problema

Los accidentes son eventos complejos y aleatorios en los que se involucra una variedad de factores ya sean éstos humanos, ambientales o propios de la mecánica de los vehículos involucrados. Por lo tanto, identificar los factores relevantes que influyen en los accidentes de tránsito y predecir la cantidad de éstos que ocurrirán durante una ventana de tiempo, resulta ser una herramienta de gran ayuda al momento de llevar a cabo planes de seguridad vial y evitar que este tipo de siniestros siga en aumento.

UKDriverDeaths es una serie cronológica que da los totales mensuales de conductores de automóviles en Gran Bretaña muertos o gravemente heridos entre enero de 1969 y diciembre de 1984. El uso obligatorio de los cinturones de seguridad fue introducido el 31 de enero de 1983.

Cinturones de seguridad es más información sobre el mismo problema.

Entendimiento de los Datos

```
Dominio             Descripción

1. DriversKilled:   Car drivers killed.

2. Drivers:         UKDriverDeaths.

3. Front:           Front-seat passengers killed or seriously injured.

4. Rear:            Rear-seat passengers killed or seriously injured.

5. Kms:             Distance driven.

6. PetrolPrice:     Petrol price.

7. VanKilled:       number of van ('light goods vehicle') drivers.

8. Law:             0/1: was the law in effect that month?
```

Exploración de los Datos

```r
# Librerías utilizadas
library(caTools)

# Establezca el directorio de trabajo
setwd("D:\\Drive\\Universidad\\Cenfotec\\MBD\\2016 Cuatrimestre 3\\MBD-305 Minería de datos 1\\Semana 7

datos <- data.frame(Seatbelts)
datos <- datos[, c(1:7)]

# La estructura del conjunto de datos:
str(datos)
```

```
## 'data.frame':    192 obs. of  7 variables:
##  $ DriversKilled: num  107 97 102 87 119 106 110 106 107 134 ...
##  $ drivers      : num  1687 1508 1507 1385 1632 ...
##  $ front        : num  867 825 806 814 991 ...
##  $ rear         : num  269 265 319 407 454 427 522 536 405 437 ...
##  $ kms          : num  9059 7685 9963 10955 11823 ...
##  $ PetrolPrice  : num  0.103 0.102 0.102 0.101 0.101 ...
##  $ VanKilled    : num  12 6 12 8 10 13 11 6 10 16 ...
```

```r
# Dividir el conjunto de datos en uno de entrenamiento y otro
# de pruebas:
set.seed(5768)
splt <- sample.split(datos$DriversKilled, SplitRatio = 0.7)
datos.entrenamiento <- datos[splt, ]
datos.prueba <- datos[!splt, ]
```

Es importante siempre validar los rangos de los conjuntos de datos creados, para evitar caer en extrapolación:

```r
summary(datos.entrenamiento)
```

```
##  DriversKilled      drivers         front             rear
##  Min.   : 60.0   Min.   :1057   Min.   : 426.0   Min.   :232.0
##  1st Qu.:104.8   1st Qu.:1462   1st Qu.: 715.5   1st Qu.:345.0
##  Median :117.5   Median :1635   Median : 837.0   Median :401.0
##  Mean   :123.4   Mean   :1674   Mean   : 839.1   Mean   :402.3
##  3rd Qu.:140.0   3rd Qu.:1822   3rd Qu.: 962.5   3rd Qu.:456.0
##  Max.   :198.0   Max.   :2654   Max.   :1299.0   Max.   :646.0
##       kms          PetrolPrice        VanKilled
##  Min.   : 8933   Min.   :0.08118   Min.   : 2.00
##  1st Qu.:12620   1st Qu.:0.09225   1st Qu.: 6.00
##  Median :14858   Median :0.10389   Median : 8.50
##  Mean   :14905   Mean   :0.10301   Mean   : 9.11
##  3rd Qu.:17301   3rd Qu.:0.11371   3rd Qu.:12.25
##  Max.   :21626   Max.   :0.13303   Max.   :17.00
```

```
summary(datos.prueba)
```

```
##  DriversKilled      drivers         front             rear
##  Min.   : 79.0   Min.   :1139   Min.   : 483.0   Min.   :224.0
##  1st Qu.:105.5   1st Qu.:1470   1st Qu.: 716.8   1st Qu.:324.8
##  Median :119.5   Median :1626   Median : 826.5   Median :408.0
##  Mean   :121.4   Mean   :1661   Mean   : 832.6   Mean   :398.6
##  3rd Qu.:136.2   3rd Qu.:1873   3rd Qu.: 922.8   3rd Qu.:458.2
##  Max.   :183.0   Max.   :2397   Max.   :1190.0   Max.   :600.0
##       kms          PetrolPrice        VanKilled
##  Min.   : 7685   Min.   :0.08275   Min.   : 4.000
##  1st Qu.:12866   1st Qu.:0.09596   1st Qu.: 6.750
##  Median :15420   Median :0.10553   Median : 8.000
##  Mean   :15209   Mean   :0.10512   Mean   : 8.929
##  3rd Qu.:16915   3rd Qu.:0.11490   3rd Qu.:11.000
##  Max.   :20705   Max.   :0.12449   Max.   :17.000
```

De acuerdo con los resúmenes anteriores, hay algunas observaciones en el conjunto de datos de prueba cuyo rango de la varibble kms se extiende más allá del rango en el conjunto de datos de entrenamiento, así que vamos a eliminar esas observaciones del conjunto de datos de prueba.

```
datos.TamanoInicial = nrow(datos.prueba)

datos.prueba <- datos.prueba[datos.prueba$kms >= 8933, ]
summary(datos.entrenamiento)
```

```
##  DriversKilled      drivers         front             rear
##  Min.   : 60.0   Min.   :1057   Min.   : 426.0   Min.   :232.0
##  1st Qu.:104.8   1st Qu.:1462   1st Qu.: 715.5   1st Qu.:345.0
##  Median :117.5   Median :1635   Median : 837.0   Median :401.0
##  Mean   :123.4   Mean   :1674   Mean   : 839.1   Mean   :402.3
##  3rd Qu.:140.0   3rd Qu.:1822   3rd Qu.: 962.5   3rd Qu.:456.0
##  Max.   :198.0   Max.   :2654   Max.   :1299.0   Max.   :646.0
##       kms          PetrolPrice        VanKilled
##  Min.   : 8933   Min.   :0.08118   Min.   : 2.00
##  1st Qu.:12620   1st Qu.:0.09225   1st Qu.: 6.00
##  Median :14858   Median :0.10389   Median : 8.50
```

```
##  Mean   :14905   Mean   :0.10301   Mean   : 9.11
##  3rd Qu.:17301   3rd Qu.:0.11371   3rd Qu.:12.25
##  Max.   :21626   Max.   :0.13303   Max.   :17.00
```

```
summary(datos.prueba)
```

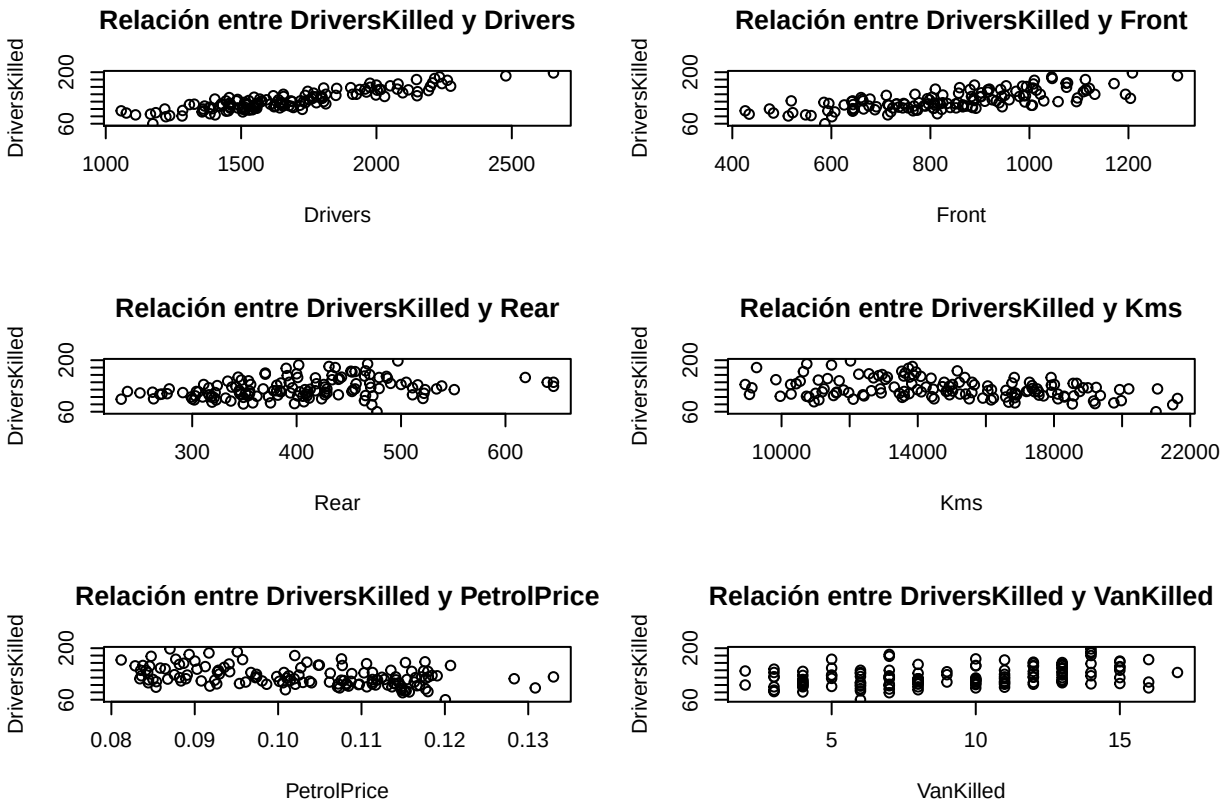
```
##  DriversKilled      drivers         front             rear
##  Min.   : 79.0   Min.   :1139   Min.   : 483.0   Min.   :224.0
##  1st Qu.:106.5   1st Qu.:1467   1st Qu.: 712.5   1st Qu.:344.0
##  Median :120.0   Median :1630   Median : 828.0   Median :411.0
##  Mean   :121.8   Mean   :1664   Mean   : 832.7   Mean   :401.1
##  3rd Qu.:136.5   3rd Qu.:1878   3rd Qu.: 925.5   3rd Qu.:458.5
##  Max.   :183.0   Max.   :2397   Max.   :1190.0   Max.   :600.0
##       kms          PetrolPrice        VanKilled
##  Min.   :10803   Min.   :0.08275   Min.   : 4.000
##  1st Qu.:12965   1st Qu.:0.09555   1st Qu.: 7.000
##  Median :15552   Median :0.10630   Median : 8.000
##  Mean   :15346   Mean   :0.10517   Mean   : 8.982
##  3rd Qu.:16921   3rd Qu.:0.11499   3rd Qu.:11.000
##  Max.   :20705   Max.   :0.12449   Max.   :17.000
```

```
paste("En total, se eliminaron ", datos.TamanoInicial - nrow(datos.prueba),
    " observaciones.")
```

```
## [1] "En total, se eliminaron  1  observaciones."
```

Necesitamos trabajar con regresiones lineales lo cual vamos a utilizar las variables cuantitativas para observar las relaciones entre estas. Vamos a utilizar las variables drivers, front, rear, kms, PetrolPrice, VanKilled relacionada con la variable DriversKilled

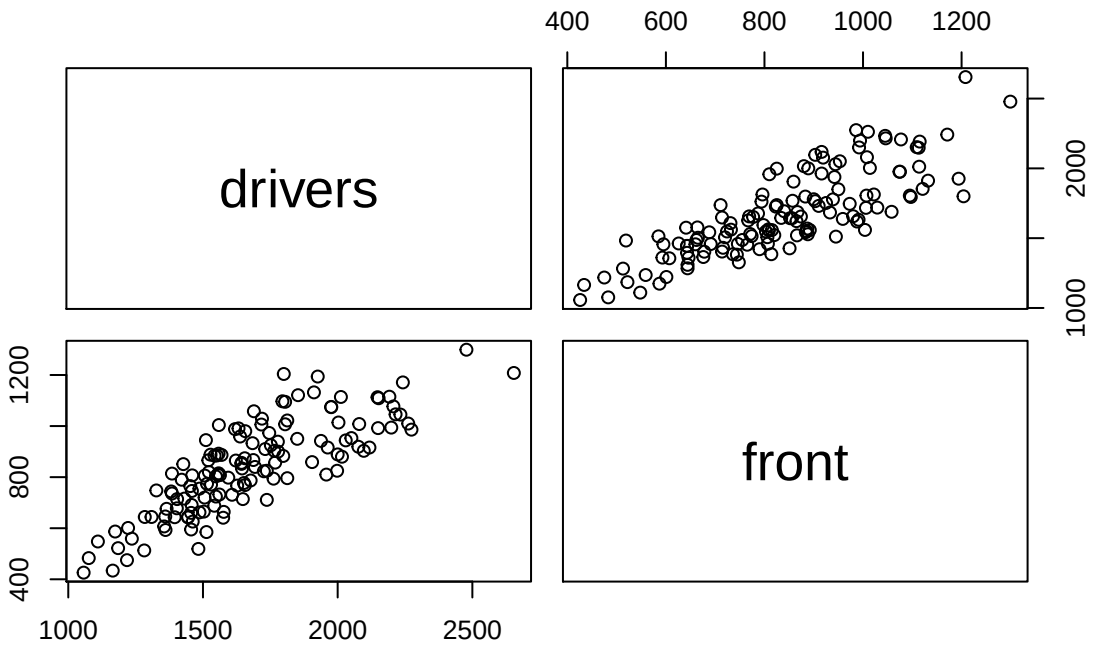
```
par(mfrow = c(3, 2))  #crear una cuadrícula de 3 columnas y 2 hileras para ver seis gráficos.
plot(x = datos.entrenamiento$drivers, y = datos.entrenamiento$DriversKilled,
    main = "Relación entre DriversKilled y Drivers", ylab = "DriversKilled",
    xlab = "Drivers")

plot(x = datos.entrenamiento$front, y = datos.entrenamiento$DriversKilled,
    main = "Relación entre DriversKilled y Front", ylab = "DriversKilled",
    xlab = "Front")

plot(x = datos.entrenamiento$rear, y = datos.entrenamiento$DriversKilled,
    main = "Relación entre DriversKilled y Rear", ylab = "DriversKilled",
    xlab = "Rear")

plot(x = datos.entrenamiento$kms, y = datos.entrenamiento$DriversKilled,
    main = "Relación entre DriversKilled y Kms", ylab = "DriversKilled",
    xlab = "Kms")

plot(x = datos.entrenamiento$PetrolPrice, y = datos.entrenamiento$DriversKilled,
    main = "Relación entre DriversKilled y PetrolPrice", ylab = "DriversKilled",
    xlab = "PetrolPrice")

plot(x = datos.entrenamiento$VanKilled, y = datos.entrenamiento$DriversKilled,
    main = "Relación entre DriversKilled y VanKilled", ylab = "DriversKilled",
    xlab = "VanKilled")
```



En los gráficos creados podemos observar un tipo de relación entre las variables DriversKilled y las variables drivers, front, rear, kms, PetrolPrice, VanKilled.Auque esta no sea lineal, si se muestra alguna relación

Necesitamos visualizar la relación entre las diferentes variables predictoras, para lo cual podemos crear una matriz de gráficos de dispersión:

```
par(mfrow = c(1, 1))  #volver a solo un gráfico por visualización.
pairs(datos.entrenamiento[!is.na(datos$DriversKilled), c(2:3)],
    main = "Relación entre predictores")
```



La información del gráfico anterior podemos complementarla con una matriz de correlación:

```
cor(datos.entrenamiento[!is.na(datos.entrenamiento$DriversKilled),
    c(1:7)])
```

```
##                DriversKilled    drivers      front       rear        kms
## DriversKilled      1.0000000  0.8926243  0.7235988  0.3355901 -0.3667119
## drivers            0.8926243  1.0000000  0.8187656  0.3273522 -0.4878644
## front              0.7235988  0.8187656  1.0000000  0.6075675 -0.3992908
## rear               0.3355901  0.3273522  0.6075675  1.0000000  0.3245412
## kms               -0.3667119 -0.4878644 -0.3992908  0.3245412  1.0000000
## PetrolPrice       -0.4022879 -0.4594991 -0.5570292 -0.1634557  0.3691116
## VanKilled          0.4107821  0.5254951  0.5582811  0.1792102 -0.5180936
##                PetrolPrice  VanKilled
## DriversKilled   -0.4022879  0.4107821
## drivers         -0.4594991  0.5254951
## front           -0.5570292  0.5582811
## rear            -0.1634557  0.1792102
## kms              0.3691116 -0.5180936
## PetrolPrice      1.0000000 -0.2838449
## VanKilled       -0.2838449  1.0000000
```

Se puede observar que la en la matriz de gráficos de dispersión, existe una correlación, significativa entre las variables DriversKilled y drivers, DriversKilled y front.

Vamos a crear una matriz de dispersión con las variables que poseen más correlación

```
cor(datos.entrenamiento[!is.na(datos.entrenamiento$DriversKilled),
    c(1:3)])
```

```
##                DriversKilled   drivers     front
## DriversKilled     1.0000000 0.8926243 0.7235988
## drivers           0.8926243 1.0000000 0.8187656
## front             0.7235988 0.8187656 1.0000000
```

Basándonos en la correlación absoluta, se va a escoger la variable drivers para ser incluida en el modelo.

Aunque la relación DriversKilled y front es significativa no llega a una correlación mayor a 0.75 por esta razón **vamos a utilizar un modelo de regresión simple**, dado que solo podemos utilizar una variable predictora como lo es drivers.

Modelo de Minería de Datos

Una vez seleccionadas las variables para incluir en el modelo de regresión, se procede a crearlo:

```
reg.DriversKilled <- lm(DriversKilled ~ drivers, data = datos.entrenamiento)
summary(reg.DriversKilled)
```

```
##
## Call:
## lm(formula = DriversKilled ~ drivers, data = datos.entrenamiento)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -28.4300  -9.1634  -0.4371   8.6563  24.7734
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.248203   5.789566  -1.252    0.213
## drivers      0.078029   0.003404  22.921   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.87 on 134 degrees of freedom
## Multiple R-squared:  0.7968, Adjusted R-squared:  0.7953
## F-statistic: 525.4 on 1 and 134 DF,  p-value: < 2.2e-16
```

En resumen el modelo, podemos ver que la variable drivers es significativa y que el modelo creado explica alrededor de un 80% de la variación en la variable de respuesta (DriversKilled). Asimismo, podemos ver que el modelo es mejor que un modelo sin variables. Con este modelo, procedemos a hacer las predicciones sobre el conjunto de datos de prueba.

```
datos.prueba$Prediccion <- predict(reg.DriversKilled, newdata = datos.prueba)
```

Evaluación

Para determinar qué tan bueno es el modelo, vamos a calcular dos métricas: primero la raíz cuadrada del promedio de los errores cuadrados (RMSE):

```r
sqrt(mean((datos.prueba$DriversKilled - datos.prueba$Prediccion)^2))
```

```
## [1] 11.12182
```

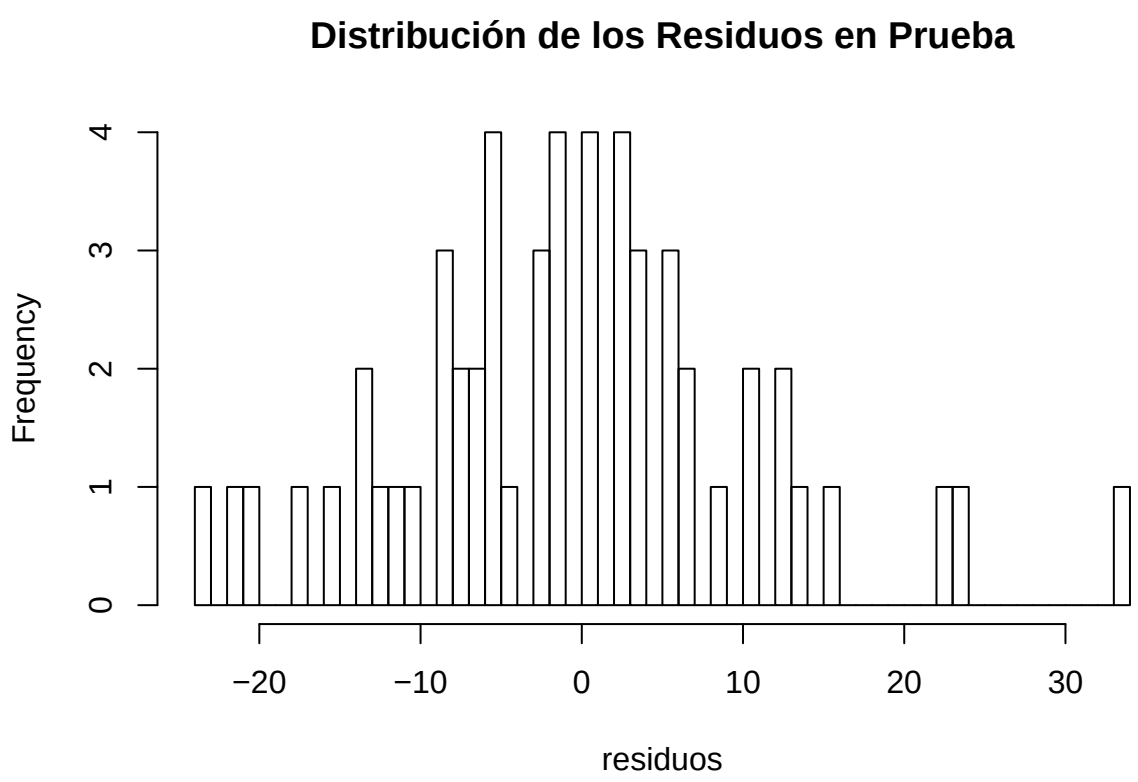
También es necesario calcular el r cuadrado:

```r
Suma.Total.Cuadrados <- sum((mean(datos.entrenamiento$DriversKilled) -
    datos.prueba$DriversKilled)^2)  #error total si usamos modelo ingenuo en prueba
Suma.Errores.Cuadrados <- sum((datos.prueba$Prediccion - datos.prueba$DriversKilled)^2)  #error total d
1 - (Suma.Errores.Cuadrados/Suma.Total.Cuadrados)
```

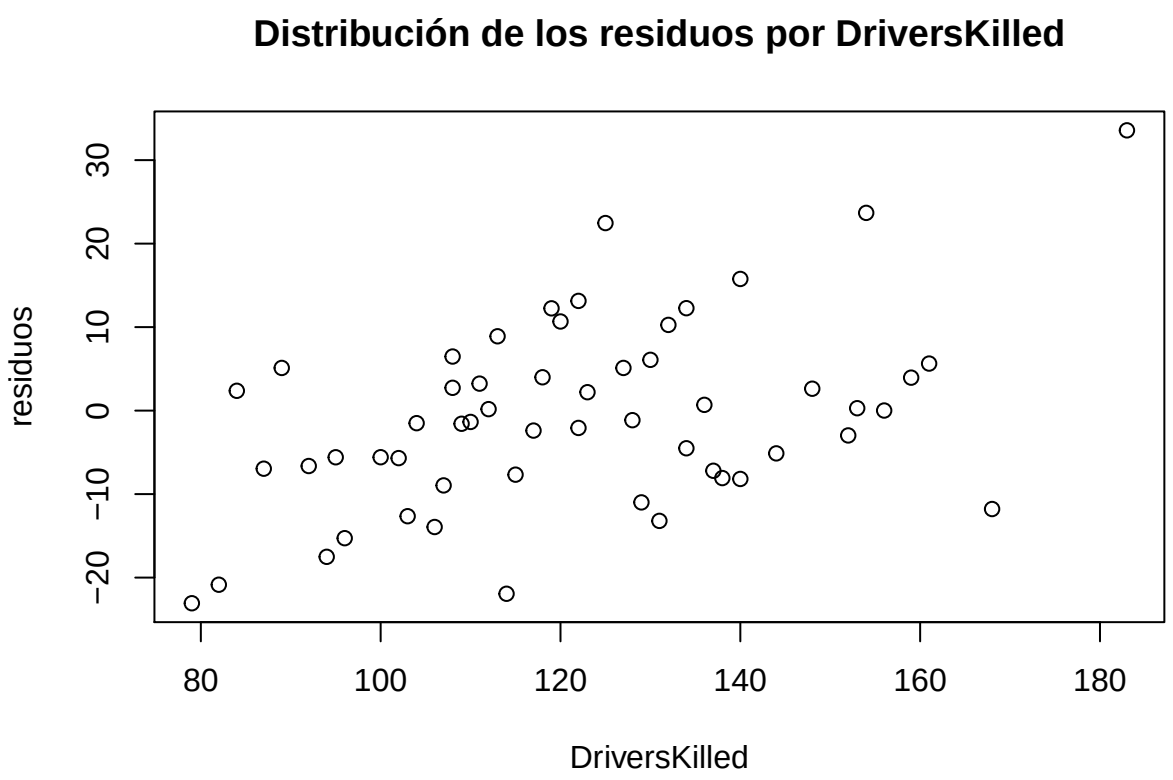
```
## [1] 0.7698508
```

Finalmente, procedemos a analizar la distribución de los residuos:

```r
hist(datos.prueba$DriversKilled - datos.prueba$Prediccion, breaks = 50,
    main = "Distribución de los Residuos en Prueba", xlab = "residuos")
```




```r
plot(y = datos.prueba$DriversKilled - datos.prueba$Prediccion,
    x = datos.prueba$DriversKilled, main = "Distribución de los residuos por DriversKilled",
    xlab = "DriversKilled", ylab = "residuos")
```



Resultados Podemos concluir con la evaluación hecha, que el modelo puede explicar cerca de un 78% de la variación de la variable DriversKilled en el conjunto de datos de prueba, y el error promedio es de alrededor de 13% DriversKilled para arriba o para abajo.

En resumen: el modelo se podría utilizar, pero se debe analizar más a fondo los datos para ver si se puede mejorar la entrada de datos , y con esto poder subir el porcentaje de predicción.

6

Análisis del Problema Hoy en día "El dinero plástico", como también se le conoce, es bienvenido en la mayoría de los establecimientos comerciales de todas las categorías: hoteles, restaurantes, agencias de viajes, entre otros. Esta herramienta de crédito es un convenio entre una institución financiera (banco u otro tipo de compañía) y el prestatario (la persona titular del crédito, o sea tú), mediante la cual se pone un cupo de dinero disponible para que lo use a través de una tarjeta de crédito, cuyo posterior pago mensual esta sujeto a los gasto que se efectúe y la cantidad de cuotas a la que sea diferida la compra.

Se quiere predecir cuales clientes son aptos para la aprobación de una tarjeta de crédito.

Este archivo se refiere a las aplicaciones de tarjetas de crédito. Los valores se han cambiado a símbolos sin sentido para proteger

Entendimiento de los Datos

```
Dominio   Descripción
A1:       b, a.
A2:       continuous.
A3:       continuous.
```

```
A4:       u, y, l, t.
A5:       g, p, gg.
A6:       c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff.
A7:       v, h, bb, j, n, z, dd, ff, o.
A8:       continuous.
A9:       t, f.
A10:        t, f.
A11:        continuous.
A12:        t, f.
A13:        g, p, s.
A14:        continuous.
A15:        continuous.
A16:      +,-         (class attribute)
```

Exploración de los Datos

```r
# Cargar las librerías necesarias
library(lattice)
library(caTools)
library(ROCR)
```

```
## Loading required package: gplots

##
## Attaching package: 'gplots'

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

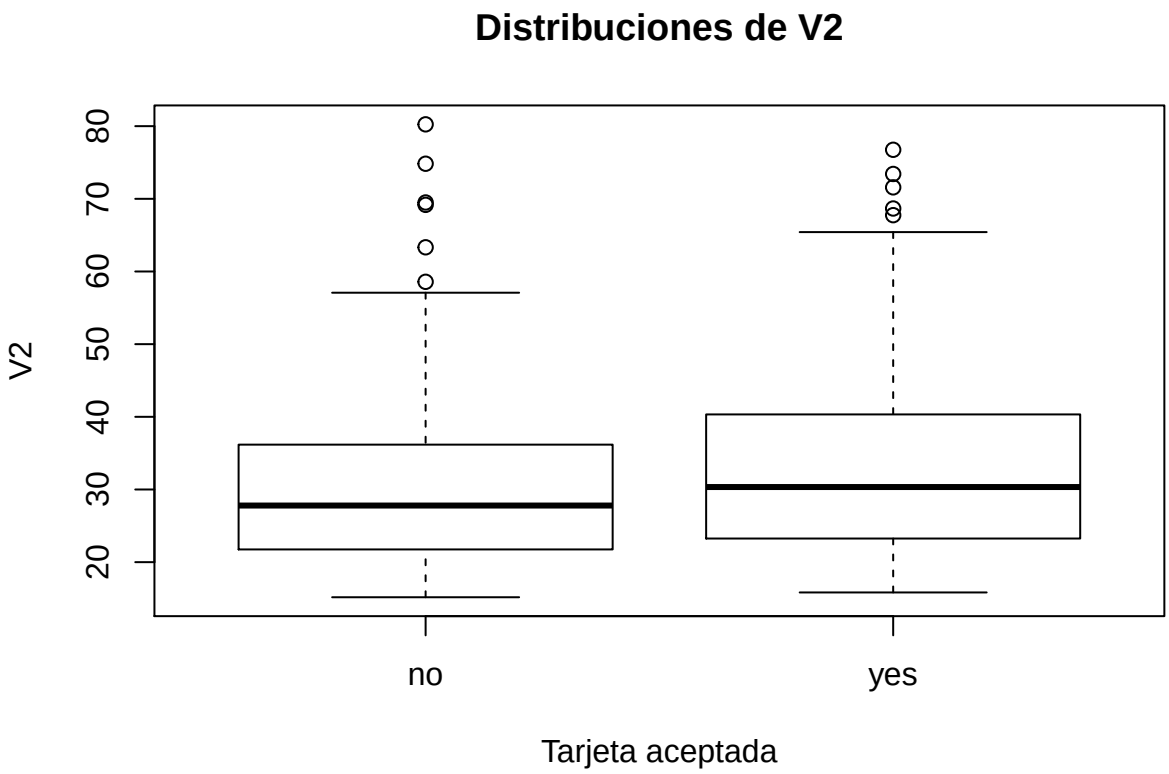
```r
# Establecer el directorio de trabajo
setwd("D:\\Drive\\Universidad\\Cenfotec\\MBD\\2016 Cuatrimestre 3\\MBD-305 Minería de datos 1\\Semana 7

# Cargar el archivo a una variable que se llame crx usando la
# función read.csv
crx = read.csv(file = "crx.data.txt", head = FALSE, sep = ",",
    na.strings = "?")

crx$V1 <- factor(crx$V1)
crx$V4 <- factor(crx$V4)
crx$V5 <- factor(crx$V5)
crx$V6 <- factor(crx$V6)
crx$V7 <- factor(crx$V7)
crx$V10 <- factor(crx$V10)
crx$V12 <- factor(crx$V12)
crx$V13 <- factor(crx$V13)

crx$V16 <- as.character(crx$V16)
crx[crx$V16 == "+", ]$V16 <- "yes"
crx[crx$V16 == "-", ]$V16 <- "no"
crx$V16 <- factor(crx$V16)

# La estructura del conjunto de datos:
str(crx)
```

```
## 'data.frame':    690 obs. of  16 variables:
##  $ V1 : Factor w/ 2 levels "a","b": 2 1 1 2 2 2 2 1 2 2 ...
##  $ V2 : num  30.8 58.7 24.5 27.8 20.2 ...
##  $ V3 : num  0 4.46 0.5 1.54 5.62 ...
##  $ V4 : Factor w/ 3 levels "l","u","y": 2 2 2 2 2 2 2 2 3 3 ...
##  $ V5 : Factor w/ 3 levels "g","gg","p": 1 1 1 1 1 1 1 1 3 3 ...
##  $ V6 : Factor w/ 14 levels "aa","c","cc",..: 13 11 11 13 13 10 12 3 9 13 ...
##  $ V7 : Factor w/ 9 levels "bb","dd","ff",..: 8 4 4 8 8 8 4 8 4 8 ...
##  $ V8 : num  1.25 3.04 1.5 3.75 1.71 ...
##  $ V9 : Factor w/ 2 levels "f","t": 2 2 2 2 2 2 2 2 2 2 ...
##  $ V10: Factor w/ 2 levels "f","t": 2 2 1 2 1 1 1 1 1 1 ...
##  $ V11: int  1 6 0 5 0 0 0 0 0 0 ...
##  $ V12: Factor w/ 2 levels "f","t": 1 1 1 2 1 2 2 1 1 2 ...
##  $ V13: Factor w/ 3 levels "g","p","s": 1 1 1 1 3 1 1 1 1 1 ...
##  $ V14: int  202 43 280 100 120 360 164 80 180 52 ...
##  $ V15: int  0 560 824 3 0 0 31285 1349 314 1442 ...
##  $ V16: Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
```

```r
# Dividir el conjunto de datos en uno de entrenamiento y otro
# de pruebas:
set.seed(5768)
splt <- sample.split(crx$V16, SplitRatio = 0.7)
crx.entrenamiento <- crx[splt, ]
crx.prueba <- crx[!splt, ]
```

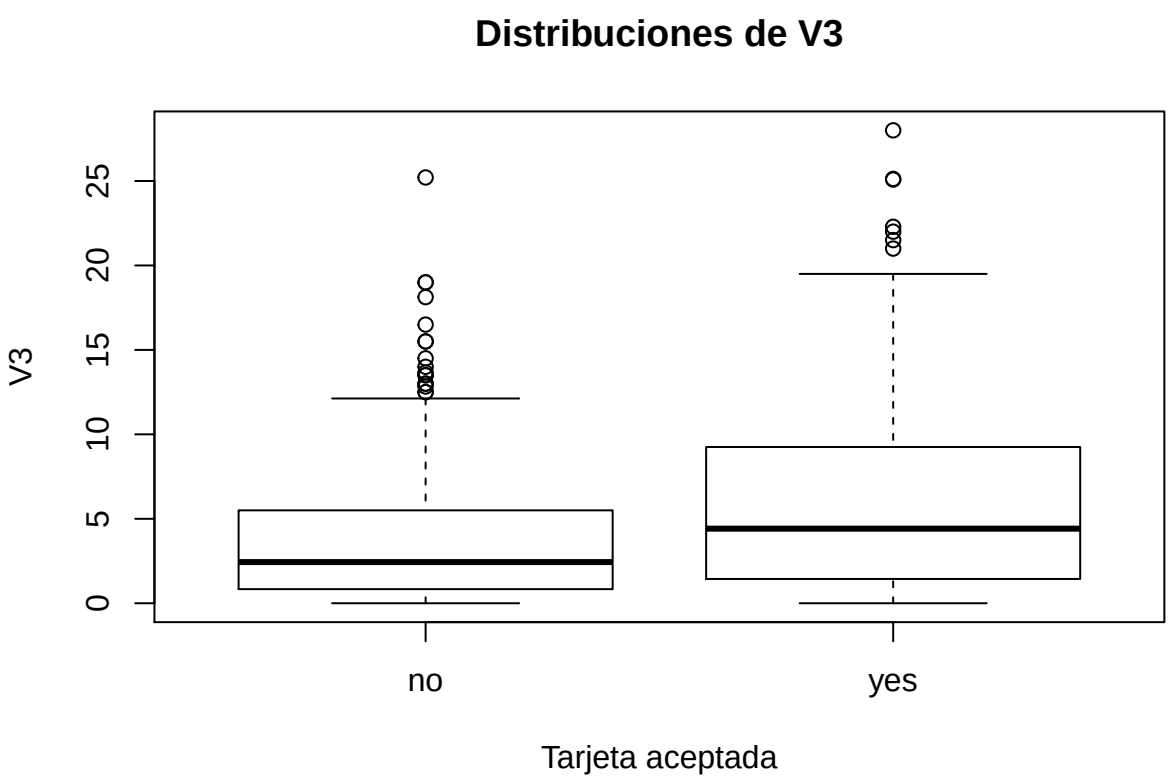
Una vez cargados los datos, podemos comenzar a explorarlos. Para comenzar, podemos analizar la distribución de la variable V16, en el contexto de si se puede aceptar una tarjeta de crédito o no:

```r
boxplot(crx.entrenamiento$V2 ~ crx.entrenamiento$V16, main = "Distribuciones de V2",
    ylab = "V2", xlab = "Tarjeta aceptada")
```



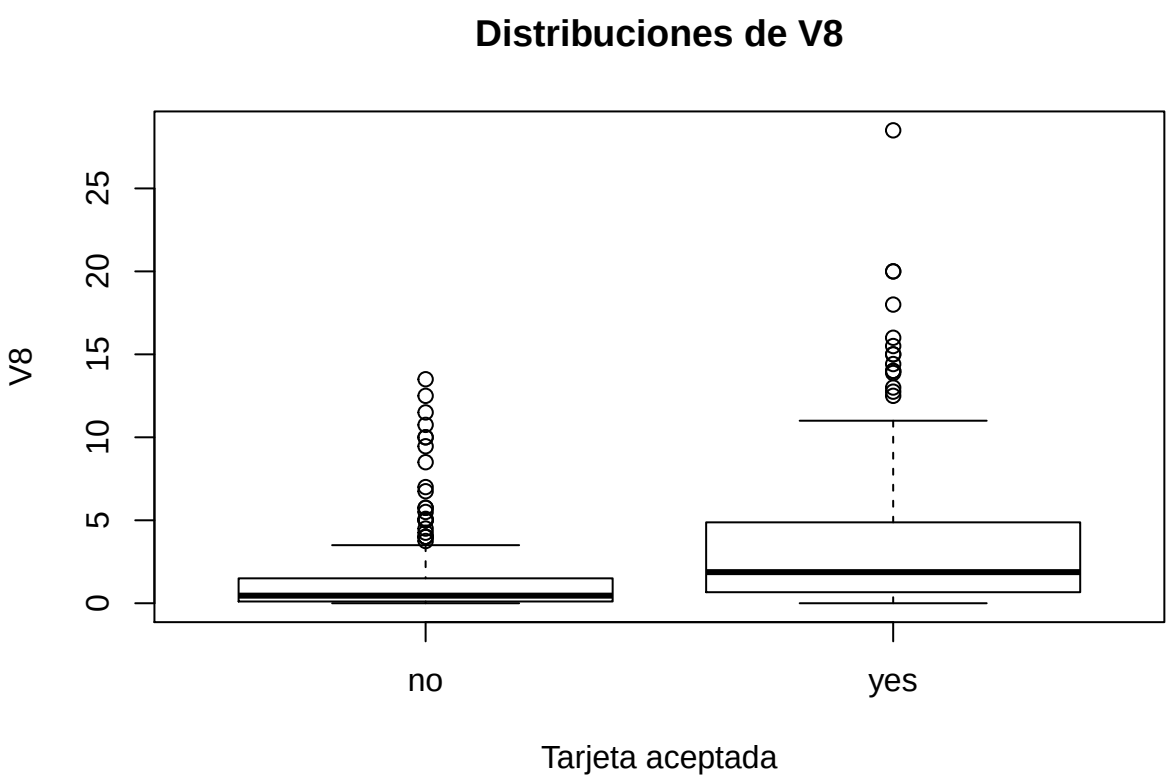
Del gráfico anterior, podemos concluir que hubo valores variados entre los diferentes rangos entre los 20 y 40 y unos valores que están generando bastante ruido arriba de los 60.

```
boxplot(crx.entrenamiento$V3 ~ crx.entrenamiento$V16, main = "Distribuciones de V3",
    ylab = "V3", xlab = "Tarjeta aceptada")
```



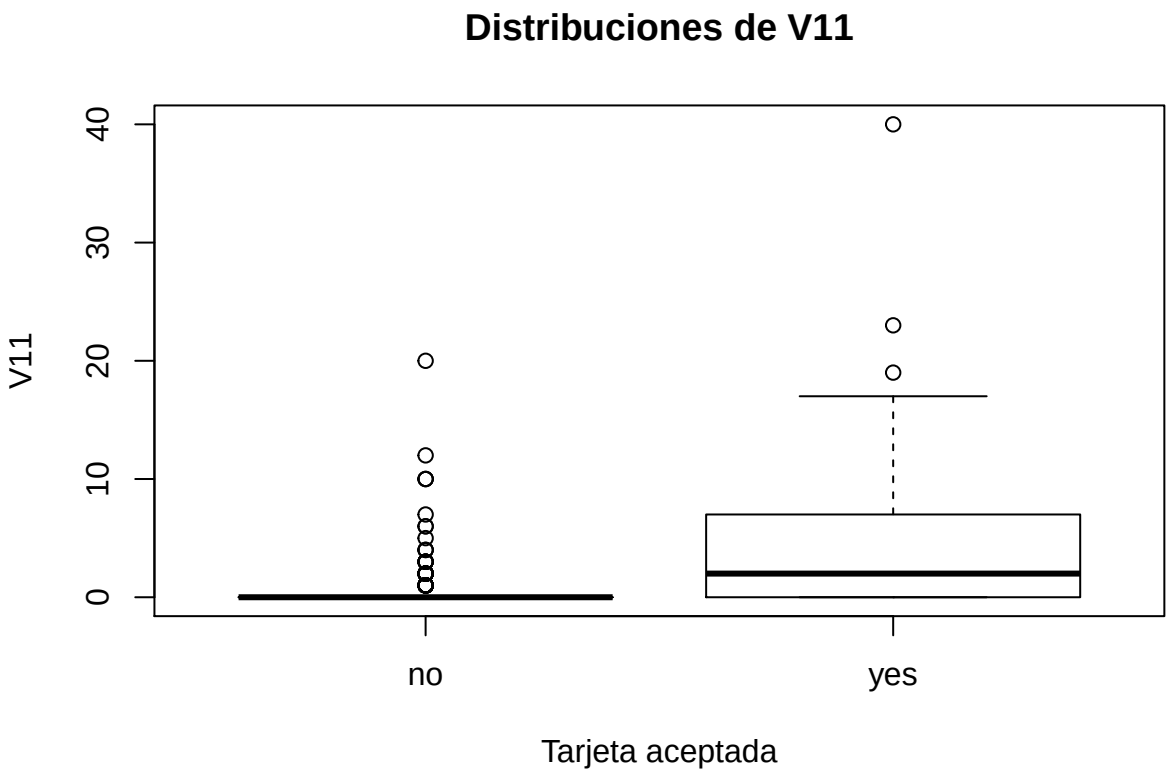
Del gráfico anterior, podemos concluir que hubo valores variados entre los diferentes rangos del sí entre 0 y 10 y unos valores que están generando bastante ruido arriba de los 15. Y valores poco variados en el no, y con valores arriba del 10 que generan bastante ruido en los datos.

```
boxplot(crx.entrenamiento$V8 ~ crx.entrenamiento$V16, main = "Distribuciones de V8",
    ylab = "V8", xlab = "Tarjeta aceptada")
```



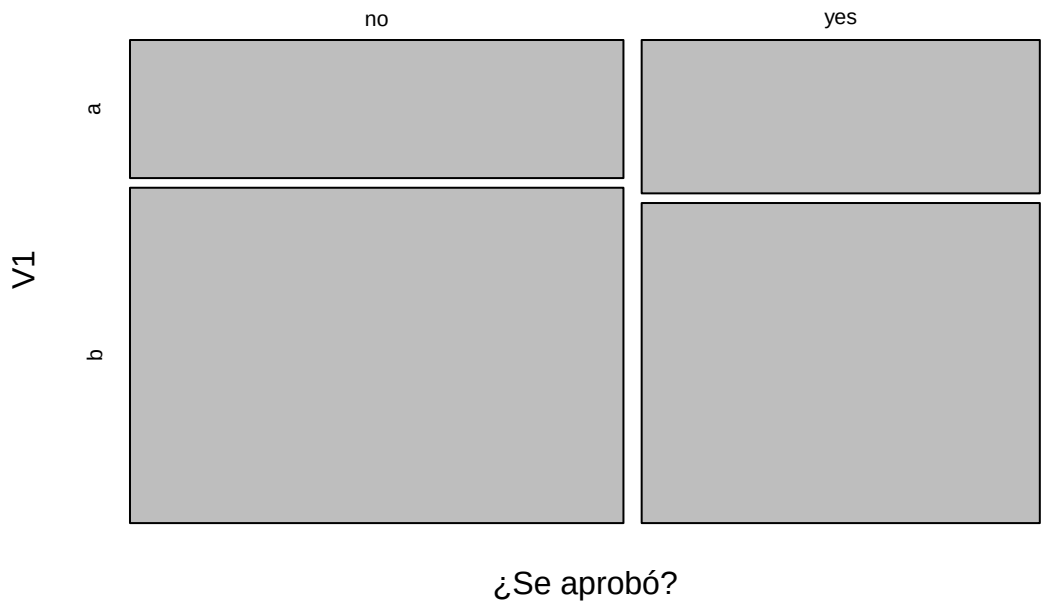
Del gráfico anterior, podemos concluir que casi no hubo valores variados entre los diferentes rangos del sí y del no y los valores arriba del 5 en no y 10 arriba del si están generando bastante ruido.

```
boxplot(crx.entrenamiento$V11 ~ crx.entrenamiento$V16, main = "Distribuciones de V11",
    ylab = "V11", xlab = "Tarjeta aceptada")
```



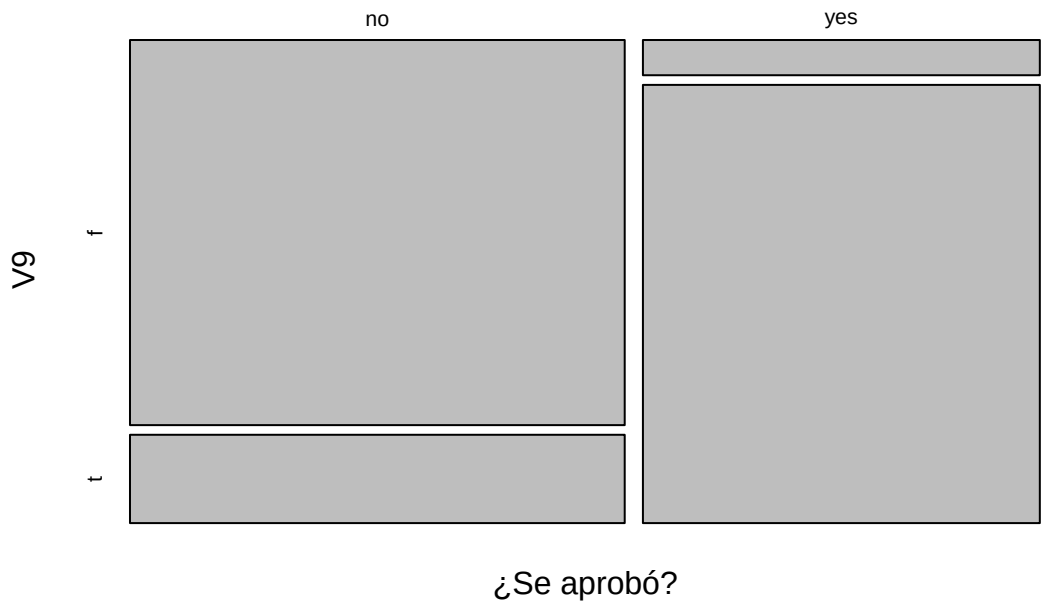
Del gráfico anterior, podemos concluir que casi no hubo valores variados entre los diferentes rangos del no, y variados en el si casi hasta el valor 10.

```
mosaicplot(~crx.entrenamiento$V16 + crx.entrenamiento$V1, main = "Proporción de aprobaciones por la vari
    ylab = "V1", xlab = "¿Se aprobó?")
```



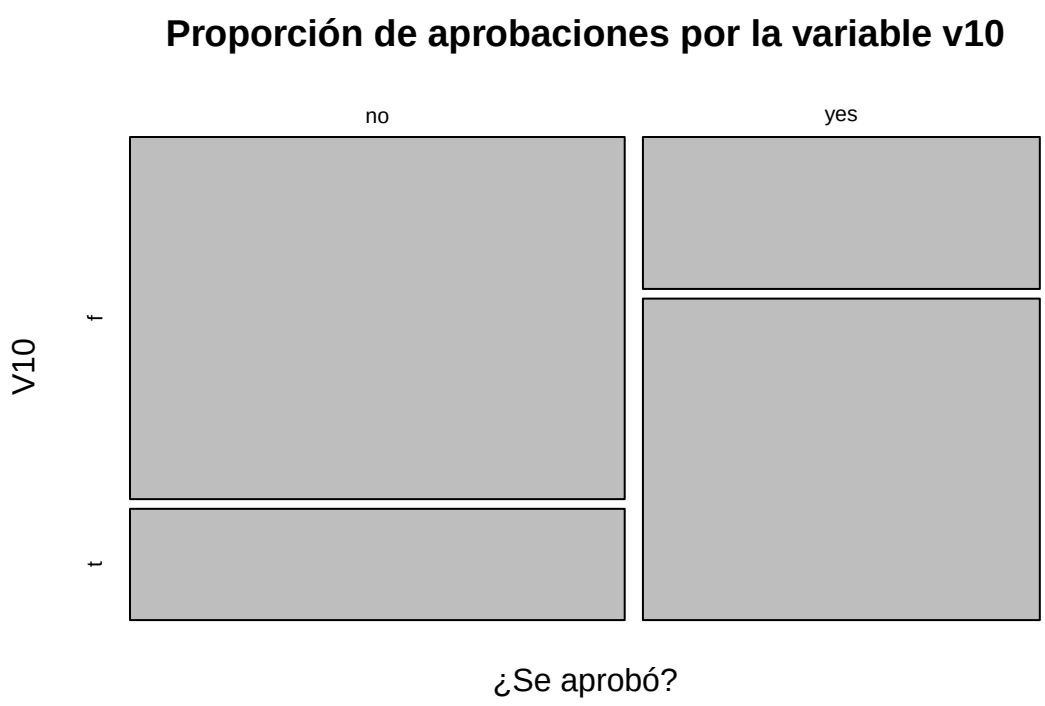
En el gráfico de mosaico arriba, podemos apreciar cómo hay un mayor número de observaciones de tipo B con la aprobación de la tarjeta de crédito, y como complemento el tipo A de menor aprobación

```
mosaicplot(~crx.entrenamiento$V16 + crx.entrenamiento$V9, main = "Proporción de aprobaciones por la vari
    ylab = "V9", xlab = "¿Se aprobó?")
```



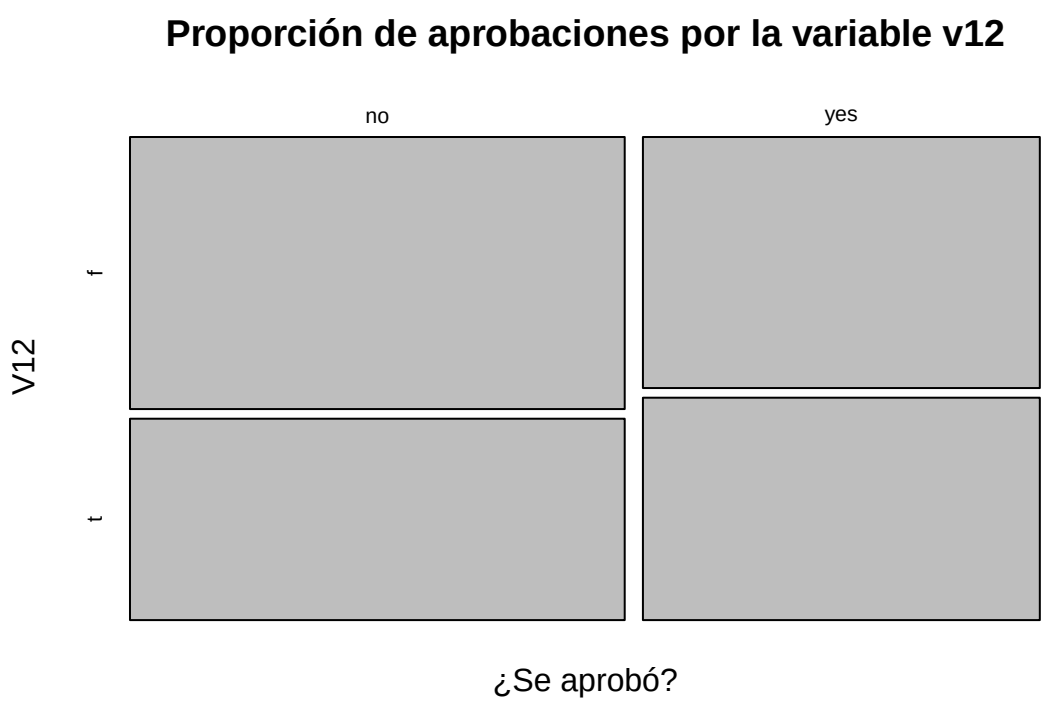
En el gráfico de mosaico arriba, podemos apreciar cómo hay un mayor número de observaciones de tipo F con la aprobación de la tarjeta de crédito, y como complemento el tipo T de menor aprobación

```
mosaicplot(~crx.entrenamiento$V16 + crx.entrenamiento$V10, main = "Proporción de aprobaciones por la va
    ylab = "V10", xlab = "¿Se aprobó?")
```



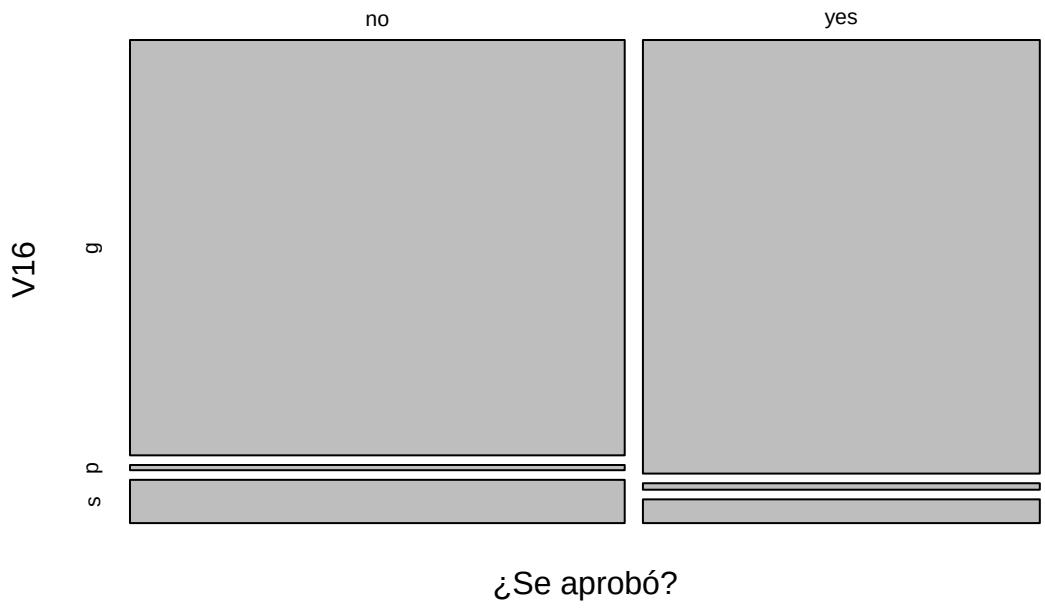
En el gráfico de mosaico arriba, podemos apreciar cómo hay un mayor número de observaciones de tipo F con la aprobación de la tarjeta de crédito, y como complemento el tipo T de menor aprobación

```
mosaicplot(~crx.entrenamiento$V16 + crx.entrenamiento$V12, main = "Proporción de aprobaciones por la va
    ylab = "V12", xlab = "¿Se aprobó?")
```



En el gráfico de mosaico arriba, podemos apreciar cómo hay un mayor número de observaciones de tipo F con la aprobación de la tarjeta de crédito, y como complemento el tipo T de menor aprobación

```
mosaicplot(~crx.entrenamiento$V16 + crx.entrenamiento$V13, main = "Proporción de aprobaciones por la va
    ylab = "V16", xlab = "¿Se aprobó?")
```



En el gráfico de mosaico arriba, podemos apreciar cómo hay un mayor número de observaciones de tipo G con la aprobación de la tarjeta de crédito, y como complemento el tipo S de menor aprobación y casi mínima aprobación en Tipo P.

Modelo de Minería de Datos

Para modelar este caso, se va a utilizar una regresión logística, en el primer modelo vamos a utilizar las variables V1 + V2 + V3 + V4 + V5 + V6 + V7+ V8+ V9+ V10+ V11+ V12+ V13+ V14+ V15:

```
crx.fit <- glm(V16 ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 +
    V9 + V10 + V11 + V12 + V13 + V14 + V15, data = crx.entrenamiento,
    family = binomial)
```

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

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

Al ver los detalles del modelo 1:

```
summary(crx.fit)
```

```
##
## Call:
## glm(formula = V16 ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 +
##     V10 + V11 + V12 + V13 + V14 + V15, family = binomial, data = crx.entrenamiento)
##
```

```
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.66130  -0.27437  -0.09946   0.36357   2.95172
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.196e+11  3.914e+14  -0.002  0.99874
## V1b         -3.226e-01  4.220e-01  -0.765  0.44452
## V2           7.694e-03  1.758e-02   0.438  0.66164
## V3          -3.309e-02  3.511e-02  -0.943  0.34593
## V4u          6.196e+11  3.914e+14   0.002  0.99874
## V4y          6.196e+11  3.914e+14   0.002  0.99874
## V5gg         6.196e+11  3.914e+14   0.002  0.99874
## V5p                 NA         NA      NA       NA
## V6c         -2.471e-01  7.471e-01  -0.331  0.74084
## V6cc         9.630e-01  9.882e-01   0.974  0.32982
## V6d          8.742e-01  1.113e+00   0.786  0.43205
## V6e          3.119e+00  1.368e+00   2.279  0.02264 *
## V6ff         2.002e+01  1.874e+05   0.000  0.99991
## V6i         -9.467e-01  9.455e-01  -1.001  0.31668
## V6j         -2.951e+01  1.320e+05   0.000  0.99982
## V6k         -5.444e-01  8.586e-01  -0.634  0.52610
## V6m         -5.417e-01  8.871e-01  -0.611  0.54144
## V6q         -2.532e-01  8.500e-01  -0.298  0.76582
## V6r          3.347e+10  6.727e+07 497.516  < 2e-16 ***
## V6w          1.925e-01  7.819e-01   0.246  0.80550
## V6x          1.984e+00  1.178e+00   1.685  0.09203 .
## V7dd        -3.319e+00  2.729e+00  -1.216  0.22382
## V7ff        -2.163e+01  1.874e+05   0.000  0.99991
## V7h          1.981e-01  7.334e-01   0.270  0.78705
## V7j          2.865e+01  1.320e+05   0.000  0.99983
## V7n          3.513e+00  1.694e+00   2.073  0.03817 *
## V7o         -8.451e+01  7.577e+05   0.000  0.99991
## V7v          4.385e-01  6.969e-01   0.629  0.52921
## V7z         -4.711e+00  1.987e+00  -2.371  0.01775 *
## V8           6.686e-02  5.951e-02   1.124  0.26122
## V9t          4.467e+00  4.948e-01   9.026  < 2e-16 ***
## V10t         6.705e-01  4.490e-01   1.493  0.13541
## V11          6.927e-02  6.369e-02   1.087  0.27682
## V12t        -3.599e-01  3.706e-01  -0.971  0.33152
## V13p        -2.361e+01  3.960e+05   0.000  0.99995
## V13s        -5.212e-01  6.643e-01  -0.785  0.43270
## V14         -3.191e-03  1.141e-03  -2.798  0.00514 **
## V15          6.919e-04  2.750e-04   2.516  0.01188 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 637.86  on 462  degrees of freedom
## Residual deviance: 239.98  on 426  degrees of freedom
##   (20 observations deleted due to missingness)
## AIC: 313.98
##
```

```
## Number of Fisher Scoring iterations: 25
```

Se puede observar que hay muchas variables que no son significativas: V1,V2,V3,V4,V5,V8,V10,V11,V12,V13,así que se procede a hacer un segundo modelo sin estas variables. El AIC presento un numero de 327.06.

Para modelar el caso dos, se va a utilizar una regresión logística, vamos a utilizar las variables V6 +V7+V9 + V14 +V15:

```
crx.fit <- glm(V16 ~ V6 + V7 + V9 + V14 + V15, data = crx.entrenamiento,
    family = binomial)
```

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

Al ver los detalles del modelo 2:

```
summary(crx.fit)
```

```
##
## Call:
## glm(formula = V16 ~ V6 + V7 + V9 + V14 + V15, family = binomial,
##     data = crx.entrenamiento)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4505  -0.2838  -0.1324   0.5057   2.8975
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.464e+00  9.237e-01  -3.750 0.000177 ***
## V6c         -3.085e-02  6.762e-01  -0.046 0.963610
## V6cc         1.638e+00  8.734e-01   1.875 0.060798 .
## V6d          2.337e-01  9.957e-01   0.235 0.814423
## V6e          3.160e+00  1.352e+00   2.337 0.019415 *
## V6ff        -3.871e+00  1.669e+00  -2.319 0.020380 *
## V6i         -5.432e-01  8.620e-01  -0.630 0.528619
## V6j         -1.884e+01  1.020e+03  -0.018 0.985261
## V6k         -4.360e-01  7.645e-01  -0.570 0.568505
## V6m         -1.895e-01  7.784e-01  -0.243 0.807666
## V6q          8.396e-01  7.840e-01   1.071 0.284238
## V6r         -9.649e+00  2.400e+03  -0.004 0.996792
## V6w          5.078e-01  7.273e-01   0.698 0.485071
## V6x          2.083e+00  1.023e+00   2.036 0.041735 *
## V7dd        -2.644e+00  2.318e+00  -1.140 0.254109
## V7ff         2.985e+00  1.530e+00   1.951 0.051074 .
## V7h          3.150e-01  6.274e-01   0.502 0.615631
## V7j          1.836e+01  1.020e+03   0.018 0.985637
## V7n          3.364e+00  1.588e+00   2.119 0.034069 *
## V7o         -5.560e+01  2.400e+03  -0.023 0.981514
## V7v          3.851e-01  5.948e-01   0.647 0.517323
## V7z         -2.919e+00  1.773e+00  -1.646 0.099756 .
## V9t          4.564e+00  4.312e-01  10.584  < 2e-16 ***
## V14         -3.104e-03  1.026e-03  -3.026 0.002478 **
## V15          8.090e-04  2.202e-04   3.674 0.000239 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 652.23  on 473  degrees of freedom
## Residual deviance: 277.06  on 449  degrees of freedom
##   (9 observations deleted due to missingness)
## AIC: 327.06
##
## Number of Fisher Scoring iterations: 15
```

En este segundo modelo, las variables significativas son V9, V14, V15, y el AIC subió de 313.98 a 327.06 pero las variables son mucho más significativas que todas las utilizadas en el primer modelo. Se puede observar que hay muchas variables que no son significativas: V6,V7 así que se procede a hacer un tercer modelo sin estas variables:

```
crx.fit <- glm(V16 ~ V9 + V15, data = crx.entrenamiento, family = binomial)
```

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

Al ver los detalles del modelo 3:

```
summary(crx.fit)
```

```
##
## Call:
## glm(formula = V16 ~ V9 + V15, family = binomial, data = crx.entrenamiento)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2081  -0.3286  -0.3251   0.7107   2.4359
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.9139393  0.2902036 -10.041  < 2e-16 ***
## V9t          3.9756282  0.3207915  12.393  < 2e-16 ***
## V15          0.0005848  0.0001669   3.505 0.000457 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 663.75  on 482  degrees of freedom
## Residual deviance: 341.57  on 480  degrees of freedom
## AIC: 347.57
##
## Number of Fisher Scoring iterations: 7
```

En este tercer modelo, las variables significativas son V9,V15, y el AIC subió de 327.06 a 347.57 en comparacion al segundo, pero las variables son mucho más significativas que todas las utilizadas en el segundo modelo.

Con respecto a la interpretación de coeficientes, se puede decir que:

*El logaritmo de las posibilidades de los tipos F de acceder a una tarjeta de crédito es mayor que la del tipo T

*La probabilidad de acceder a una tarjeta de crédito es mayor conforme crece la variable V15.

Evaluación

A manera de modelo ingenuo, podemos tener un modelo que prediga que nadie acedio a una tarjeta de credito, pues es el resultado más frecuente. Dicho modelo tendría una exactitud del 43.47% (92 aciertos de 207 en el conjunto de pruebas).

```
table(crx.entrenamiento$V16)
```

```
##
##  no yes
## 268 215
```

```
table(crx.prueba$V16, rep("yes", nrow(crx.prueba)))
```

```
##
##       yes
##   no  115
##   yes  92
```

Al generar las predicciones del modelo sobre el conjunto de pruebas, tenemos las siguientes métricas según la tabla abajo (usando 0.5 como umbral de discriminación):

. Exactitud: 83%

. Sensibilidad: 93%

. Especificidad: 75%

. Área bajo la curva: 90%

```
predicciones <- predict(crx.fit, newdata = crx.prueba, type = "response")

table(crx.prueba$V16, predicciones >= 0.5)
```

```
##
##       FALSE TRUE
##   no     86   29
##   yes     7   85
```

```
# Exactitud:
(86 + 85)/nrow(crx.prueba)
```

```
## [1] 0.826087
```

```
# Sensibilidad:
85/(85 + 7)
```

```
## [1] 0.923913
```

```r
# Especificidad:
86/(86 + 29)
```

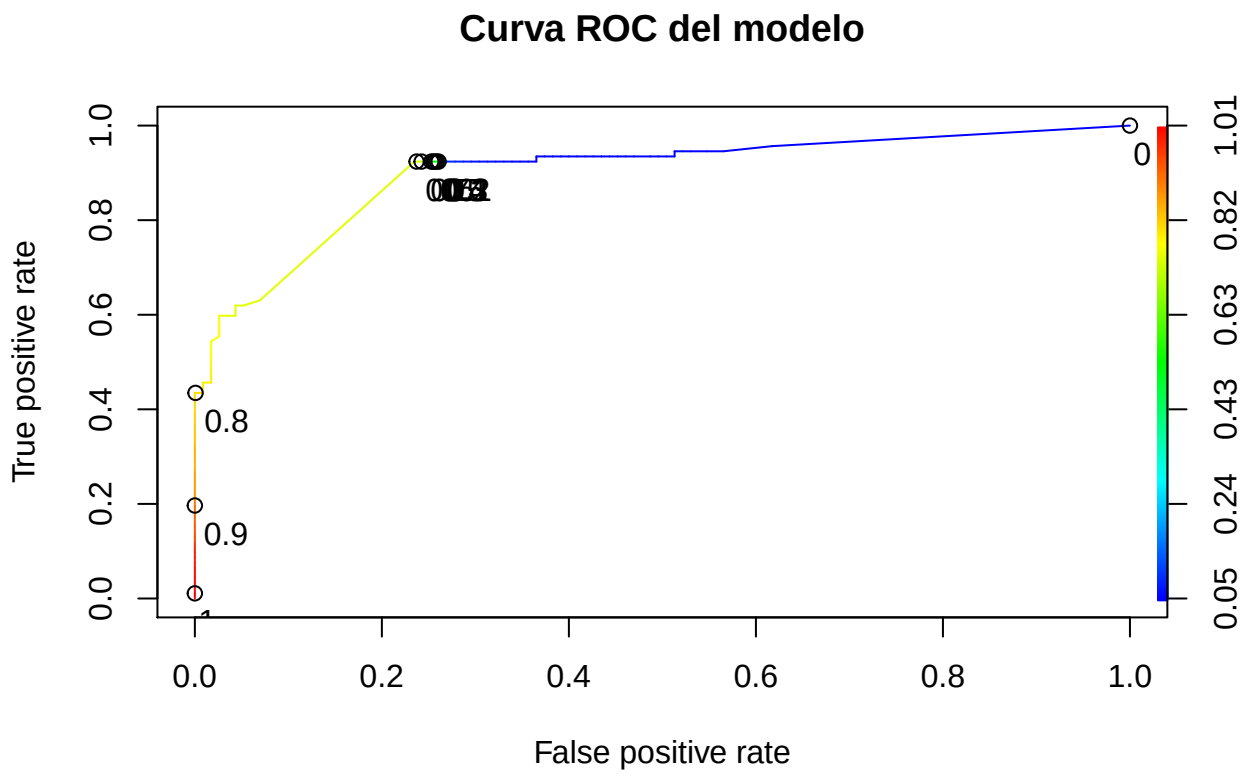
```
## [1] 0.7478261
```

```r
# Área bajo la curva:
prediccionesROC <- prediction(predicciones, crx.prueba$V16)
as.numeric(performance(prediccionesROC, "auc")@y.values)
```

```
## [1] 0.8995274
```

Según la curva ROC, al intentar aumentar el porcentaje de verdaderos positivos (sensibilidad) o de verdaderos negativos (especificdad) del modelo, estaría aumentando significativamente el porcentaje de falsos positivos y falsos negativos, respectivamente:

```r
plot(performance(prediccionesROC, "tpr", "fpr"), colorize = T,
    print.cutoffs.at = seq(0, 1, by = 0.1), text.adj = c(-0.2,
        1.7), main = "Curva ROC del modelo")
```



Resultados Se puede decir que este modelo es mejor que el modelo ingenuo a nivel de exactitud. Tenemos el porcentaje del 90% de la aceptación de tarjetas de crédito. Lo cual puede ser bastante bueno si queremos un mayor porcentaje de tarjetas aceptadas

Cambiaremos el umbral de discriminación para saber con mayor exactitud las tarjetas aceptadas:

```r
table(crx.prueba$V16, predicciones >= 0.1)
```

```
##
##       FALSE TRUE
##   no     86   29
##   yes     7   85
```

Con este cambio, la sensibilidad del modelo sube a 0.923913% (se identificaron correctamente al 93% de la aceptación de las tarjetas de crédito). Además, la especificidad se mantuvo al 75% por lo cual se puede negar el 75 % de las tarjetas de crédito que no contiene los requisitos necesarios para obtenerla.El 25% de las tarjetas fueron negadas, aunque los clientes cumplieran los requisitos.

El modelo es bastante valido para obtener la mayor cantidad de verdaderos positivos, dado que es mejor obtener negación a clientes de tarjetas de crédito, que aceptar tarjeras de crédito a clientes que no pueden hacerse cargo de estas.