Crímenes de Chicago

Paula Carballo, Daniel Ciprián y Ester Cortés

16, julio, 2018

PRIMERA PARTE

Introducción

En esta primera parte de la práctica queremos determinar cuántas comisarías hay que colocar en Chicago y dónde hay que colocarlas

Carga de paquetes

En primer lugar, cargamos las librerías que vamos a necesitar para nuestro estudio.

```
library(tidyverse)
library(lubridate)
library(cowplot)
library(chron)
library(MASS)
library(imager)
library(ROCR)
library(cluster)
library(randomForest)
library(rpart)
library(e1071)
library(rattle)
```

Tratamiento de los datos

1. Carga de los datos

```
Chicago <- read.csv("D:/Usuarios/Ester/Documents/entrega/crimes-in-chicago/Chicago_Crime
```

```
df_Chicago <- Chicago
head(df_Chicago)
```

```
##      X       ID Case.Number                  Date                Block
## 1    3 10508693    HZ250496 05/03/2016 11:40:00 PM   013XX S SAWYER AVE
## 2   89 10508695    HZ250409 05/03/2016 09:40:00 PM   061XX S DREXEL AVE
```

```
## 3  197 10508697    HZ250503 05/03/2016 11:31:00 PM  053XX W CHICAGO AVE
## 4  673 10508698    HZ250424 05/03/2016 10:10:00 PM    049XX W FULTON ST
## 5  911 10508699    HZ250455 05/03/2016 10:00:00 PM    003XX N LOTUS AVE
## 6 1108 10508702    HZ250447 05/03/2016 10:35:00 PM 082XX S MARYLAND AVE
##   IUCR           Primary.Type             Description Location.Description
## 1 0486                BATTERY DOMESTIC BATTERY SIMPLE            APARTMENT
## 2 0486                BATTERY DOMESTIC BATTERY SIMPLE            RESIDENCE
## 3 0470 PUBLIC PEACE VIOLATION        RECKLESS CONDUCT               STREET
## 4 0460                BATTERY                  SIMPLE             SIDEWALK
## 5 0820                  THEFT          $500 AND UNDER            RESIDENCE
## 6 041A                BATTERY     AGGRAVATED: HANDGUN               STREET
##   Arrest Domestic Beat District Ward Community.Area FBI.Code X.Coordinate
## 1   True     True 1022       10   24             29      08B      1154907
## 2  False     True  313        3   20             42      08B      1183066
## 3  False    False 1524       15   37             25       24      1140789
## 4  False    False 1532       15   28             25      08B      1143223
## 5  False     True 1523       15   28             25       06      1139890
## 6  False    False  631        6    8             44      04B      1183336
##   Y.Coordinate Year            Updated.On Latitude Longitude
## 1      1893681 2016 05/10/2016 03:56:50 PM 41.86407 -87.70682
## 2      1864330 2016 05/10/2016 03:56:50 PM 41.78292 -87.60436
## 3      1904819 2016 05/10/2016 03:56:50 PM 41.89491 -87.75837
## 4      1901475 2016 05/10/2016 03:56:50 PM 41.88569 -87.74952
## 5      1901675 2016 05/10/2016 03:56:50 PM 41.88630 -87.76175
## 6      1850642 2016 05/10/2016 03:56:50 PM 41.74535 -87.60380
##                       Location
## 1 (41.864073157, -87.706818608)
## 2  (41.782921527, -87.60436317)
## 3 (41.894908283, -87.758371958)
## 4 (41.885686845, -87.749515983)
## 5 (41.886297242, -87.761750709)
## 6 (41.745354023, -87.603798903)
```

2. Resumen de las variables que conforman nuestros datos

1.- ID - Unique identifier for the record.

2.- Case Number - The Chicago Police Department RD Number (Records Division Number), which is unique to the incident

3.- Date - Date when the incident occurred. this is sometimes a best estimate.

4.- Block - The partially redacted address where the incident occurred, placing it on the same block as the actual address.

5.- IUCR - The Illinois Unifrom Crime Reporting code. This is directly linked to the Primary Type and Description.

6.- Primary Type - The primary description of the IUCR code.

7.- Description - The secondary description of the IUCR code, a subcategory of the primary description.

8.- Location Description - Description of the location where the incident occurred.

9.- Arrest - Indicates whether an arrest was made.

10.- Domestic - Indicates whether the incident was domestic-related as defined by the Illinois Domestic Violence Act.

11.- Beat - Indicates the beat where the incident occurred. A beat is the smallest police geographic area – each beat has a dedicated police beat car. Three to five beats make up a police sector, and three sectors make up a police district. The Chicago Police Department has 22 police districts.

12.- District - Indicates the police district where the incident occurred.

13.- Ward - The ward (City Council district) where the incident occurred.

14- Community Area - Indicates the community area where the incident occurred. Chicago has 77 community areas.

15.- FBI Code - Indicates the crime classification as outlined in the FBI's National Incident-Based Reporting System (NIBRS).

16.- X Coordinate - The x coordinate of the location where the incident occurred in State Plane Illinois East NAD 1983 projection. This location is shifted from the actual location for partial redaction but falls on the same block.

17.- Y Coordinate - The y coordinate of the location where the incident occurred in State Plane Illinois East NAD 1983 projection. This location is shifted from the actual location for partial redaction but falls on the same block.

18.- Year - Year the incident occurred.

19.- Updated On - Date and time the record was last updated.

20.- Latitude - The latitude of the location where the incident occurred. This location is shifted from the actual location for partial redaction but falls on the same block.

21.- Longitude - The longitude of the location where the incident occurred. This location is shifted from the actual location for partial redaction but falls on the same block.

22.- Location - The location where the incident occurred in a format that allows for creation of maps and other geographic operations on this data portal. This location is shifted from the actual location for partial redaction but falls on the same block.

3. Estudio de los datos

Antes de empezar a trabajar con los datos es importante saber cómo están organizados los mismos, qué variables tenemos y cómo se almacenan.

```r
summary(df_Chicago)
```

```
##        X                  ID             Case.Number
##  Min.   :      3   Min.   :   20224   HZ140230:      6
##  1st Qu.:2698636   1st Qu.: 9002709   HY346207:      4
##  Median :3063654   Median : 9605776   HZ403466:      4
##  Mean   :3308606   Mean   : 9597550   HZ554936:      4
##  3rd Qu.:3428849   3rd Qu.:10225766   HV217424:      3
##  Max.   :6253474   Max.   :10827880   HW486725:      3
##                                       (Other) :1456690
##                       Date
##  01/01/2012 12:01:00 AM:    166
##  01/01/2013 12:01:00 AM:    122
##  01/01/2012 12:00:00 AM:    115
##  01/01/2015 12:01:00 AM:    110
##  01/01/2014 12:01:00 AM:    104
##  01/01/2016 12:01:00 AM:    104
##  (Other)               :1455993
##                                    Block               IUCR
##  001XX N STATE ST                     :   3634   0820   :136036
##  0000X W TERMINAL ST                  :   2746   0486   :130700
##  008XX N MICHIGAN AVE                 :   2465   0460   : 88069
##  076XX S CICERO AVE                   :   2116   0810   : 74906
##  0000X N STATE ST                     :   1844   1320   : 72515
##  064XX S DR MARTIN LUTHER KING JR DR:   1349   1310   : 71694
##  (Other)                              :1442560   (Other):882794
##           Primary.Type                        Description
##  THEFT          :329460   SIMPLE                    :150600
##  BATTERY        :263700   $500 AND UNDER            :136036
##  CRIMINAL DAMAGE:155455   DOMESTIC BATTERY SIMPLE:130700
##  NARCOTICS      :135240   TO VEHICLE                : 75801
##  ASSAULT        : 91289   OVER $500                 : 74906
##  OTHER OFFENSE  : 87874   TO PROPERTY               : 71694
##  (Other)        :393696   (Other)                   :816977
##                       Location.Description   Arrest         Domestic
##  STREET                           :330471    False:1079242   False:1236660
##  RESIDENCE                        :233530    True : 377472   True : 220054
##  APARTMENT                        :185023
##  SIDEWALK                         :160891
##  OTHER                            : 55774
##  PARKING LOT/GARAGE(NON.RESID.): 41768
##  (Other)                          :449257
##       Beat          District          Ward       Community.Area
##  Min.   : 111   Min.   : 1.00   Min.   : 1.00   Min.   : 0.00
```

```
##  1st Qu.: 613   1st Qu.: 6.00   1st Qu.:10.00   1st Qu.:23.00
##  Median :1024   Median :10.00   Median :23.00   Median :32.00
##  Mean   :1151   Mean   :11.26   Mean   :22.87   Mean   :37.46
##  3rd Qu.:1711   3rd Qu.:17.00   3rd Qu.:34.00   3rd Qu.:56.00
##  Max.   :2535   Max.   :31.00   Max.   :50.00   Max.   :77.00
##                 NA's   :1       NA's   :14      NA's   :40
##     FBI.Code       X.Coordinate      Y.Coordinate          Year
##  06     :329460   Min.   :      0   Min.   :      0   Min.   :2012
##  08B    :227082   1st Qu.:1152544   1st Qu.:1858762   1st Qu.:2013
##  14     :155455   Median :1166021   Median :1891502   Median :2014
##  26     :137597   Mean   :1164398   Mean   :1885523   Mean   :2014
##  18     :129796   3rd Qu.:1176363   3rd Qu.:1908713   3rd Qu.:2015
##  05     : 83397   Max.   :1205119   Max.   :1951573   Max.   :2017
##  (Other):393927   NA's   :37083     NA's   :37083
##                   Updated.On        Latitude       Longitude
##  02/04/2016 06:33:39 AM:908366   Min.   :36.62   Min.   :-91.69
##  08/17/2015 03:03:40 PM:158320   1st Qu.:41.77   1st Qu.:-87.72
##  04/15/2016 03:49:27 PM:  7854   Median :41.86   Median :-87.67
##  09/10/2015 11:43:14 AM:  6479   Mean   :41.84   Mean   :-87.67
##  10/09/2015 03:58:54 PM:  6030   3rd Qu.:41.91   3rd Qu.:-87.63
##  08/31/2015 03:43:09 PM:  5063   Max.   :42.02   Max.   :-87.52
##  (Other)               :364602   NA's   :37083   NA's   :37083
##                          Location
##                                :  37083
##  (41.883500187, -87.627876698):   2096
##  (41.754592961, -87.741528537):   2084
##  (41.979006297, -87.906463155):   1338
##  (41.897895128, -87.624096605):   1320
##  (41.742710224, -87.634088181):   1122
##  (Other)                       :1411671
```

4. Observaciones

- La columna X la podemos eliminar porque no aporta información relevante para nuestro trabajo

```
df_Chicago$X <- NULL
```

- La variable "Case.Number" debe se única para cada incidente y, sin embargo, vemos que se repite en varias ocasiones, por lo que eliminamos las filas con número de caso repetido.

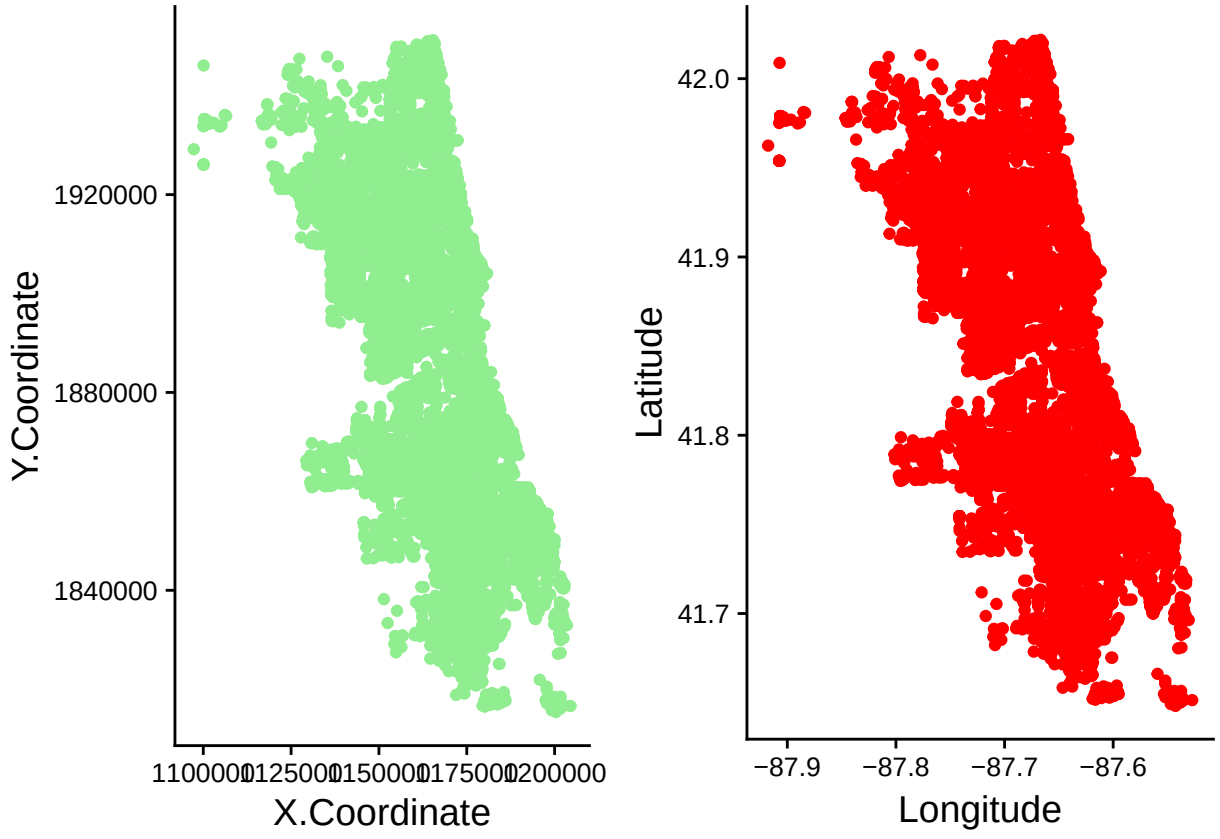
```
df_Chicago <- subset(df_Chicago, !duplicated(df_Chicago$Case.Number))
```

- Asimismo, la variable ID hace referencia al Case.Number por lo que podemos eliminarla. Lo mismo sucede con las variables IUCR y Primary.Type, por lo que nos quedaremos con Primary.Type y eliminaremos IUCR

```
df_Chicago$ID <- NULL
df_Chicago$IUCR <- NULL
```

- Las variables "X.Coordinate" e "Y.Coordinate" están relacionadas con "Latitude" y "Longitude", en tanto en cuanto que si realizamos una primera visualización de las mismas obtenemos un plano de Chicago.

```
plot1 <- ggplot(df_Chicago[1:10000,]) +
  geom_point(aes(x=df_Chicago$X.Coordinate[1:10000],
    y=df_Chicago$Y.Coordinate[1:10000]), color="light green")+
  xlab("X.Coordinate")+ ylab("Y.Coordinate")+
  theme(axis.text.x = element_text(size=10, angle=0),
    axis.text.y = element_text(size=10, angle=0))
plot2 <- ggplot(df_Chicago[1:10000,]) +
  geom_point(aes(x=df_Chicago$Longitude[1:10000],
    y=df_Chicago$Latitude[1:10000]),color="Red")+
  xlab("Longitude")+ ylab("Latitude")+
  theme(axis.text.x = element_text(size=10, angle=0),
    axis.text.y = element_text(size=10, angle=0))
plot_grid(plot1,plot2)
```



Por tanto, podemos trabajar con las variables "X.Coordinate" y "Y.Coordinate".

- La variable “Location” es un string, combinación de las variables “Latitude” y “Longitude”

```
df_Chicago$Latitude <- NULL
df_Chicago$Longitude <- NULL
df_Chicago$Location <- NULL
```

- Tras haber eliminado esas tres variables, tenemos datos faltantes en las siguientes variables:
  - District
  - Ward
  - Community.Area
  - X.Coordinate
  - Y.Coordinate

Es en éstas últimas en las que hay más NA (37077 en cada una). Dado que tenemos 1456714 datos, el porcentaje de NA’s está alrededor del 2%, que no es muy elevado por lo que podemos trabajar sin esos datos.

```
df_Chicago <- na.omit(df_Chicago)
```

- Tenemos varias variables de localización, por lo que “Block” y “Beat” podemos eliminarlas, pues la información es redundante

```
df_Chicago$Block <- NULL
df_Chicago$Beat <- NULL
```

- La variable “Updated.On” tampoco nos aporta información relevante para nuestro estudio, por lo que también podemos prescindir de ella.

```
df_Chicago$Updated.On <- NULL
```

- Unificar niveles dentro de la variable Primary.Type:

El delito NON - CRIMINAL aparece de varias formas(3):

```
df_Chicago$Primary.Type[which(df_Chicago$Primary.Type ==
  "NON - CRIMINAL")] <- "NON-CRIMINAL"
df_Chicago$Primary.Type[which(df_Chicago$Primary.Type ==
  "NON-CRIMINAL (SUBJECT SPECIFIED)")] <- "NON-CRIMINAL"

df_Chicago$Primary.Type <- factor(df_Chicago$Primary.Type,
  levels= c("ARSON","ASSAULT","BATTERY","BURGLARY",
  "CONCEALED CARRY LICENSE VIOLATION","CRIM SEXUAL ASSAULT",
  "CRIMINAL DAMAGE","CRIMINAL TRESPASS","DECEPTIVE PRACTICE","GAMBLING",
  "HOMICIDE","HUMAN TRAFFICKING","INTERFERENCE WITH PUBLIC OFFICER",
  "INTIMIDATION","KIDNAPPING","LIQUOR LAW VIOLATION","MOTOR VEHICLE THEFT",
  "NARCOTICS","NON-CRIMINAL","OBSCENITY","OFFENSE INVOLVING CHILDREN",
```

```
  "OTHER NARCOTIC VIOLATION","OTHER OFFENSE","PROSTITUTION",
  "PUBLIC INDECENCY","PUBLIC PEACE VIOLATION","ROBBERY","SEX OFFENSE",
  "STALKING","THEFT", "WEAPONS VIOLATION"))
```

- Asimismo, la variable FBI.Code no concuerda con la documentación e información proporcionada por el FBI(https://ucr.fbi.gov/nibrs/nibrs-user-manual), por lo que la eliminamos y creamos una nueva, según la citada documentación.

Podemos clasificar los crímenes en 2 grupos principales:

- Grupo A: Graves-muy graves
- Grupo B: Delitos leves

Analizamos la variable "Primary.Type" para ver qué tipos de delitos tenemos

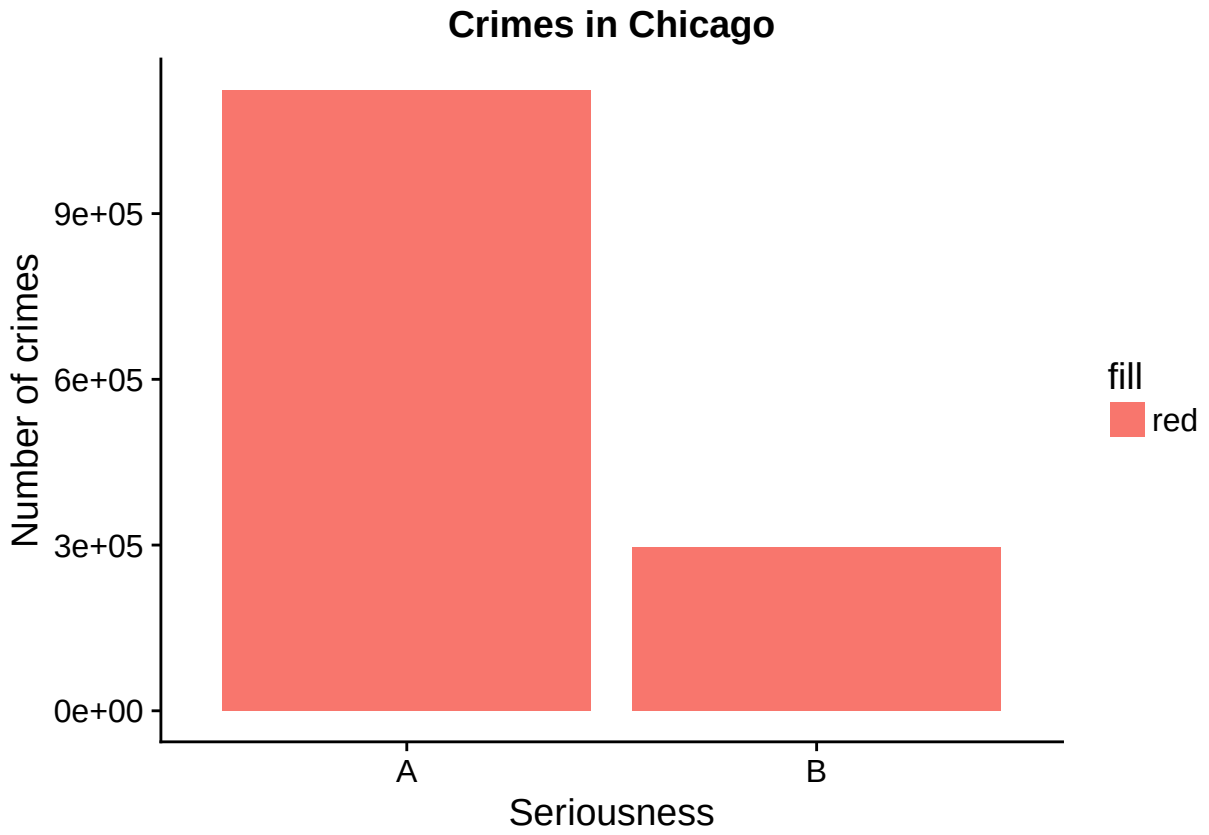
```
df_Chicago$FBI.Code <- NULL
length(unique(df_Chicago$Primary.Type))
```

```
## [1] 31
```

```
Group_A <- c("HOMICIDE","ROBBERY","CRIM SEXUAL ASSAULT","ASSAULT","BURGLARY",
  "SEX OFFENSE","BATTERY","THEFT","MOTOR VEHICLE THEFT",
  "OFFENSE INVOLVING CHILDREN","DECEPTIVE PRACTICE","NARCOTICS","ARSON",
  "WEAPONS VIOLATION","PROSTITUTION","KIDNAPPING","GAMBLING","STALKING",
  "INTIMIDATION","OBSCENITY","CONCEALED CARRY LICENSE VIOLATION",
  "HUMAN TRAFFICKING")
Group_B <- c("CRIMINAL TRESPASS","CRIMINAL DAMAGE","PUBLIC PEACE VIOLATION",
  "OTHER OFFENSE","INTERFERENCE WITH PUBLIC OFFICER","LIQUOR LAW VIOLATION",
  "PUBLIC INDECENCY","NON-CRIMINAL","OTHER NARCOTIC VIOLATION")
df_Chicago$Seriousness <- ifelse(df_Chicago$Primary.Type %in% Group_A, "A","B")
```

Hacemos una gráfica del número de crímenes de cada tipo

```
qplot(df_Chicago$Seriousness, xlab = 'Seriousness', main ='Crimes in Chicago',
  fill="red") + scale_y_continuous('Number of crimes')
```




```
df_Chicago$Seriousness <- as.numeric(factor(df_Chicago$Seriousness,
  levels=c("A","B")))
```

5.- Tratamiento de fechas y horas

La variable "Date" es de clase "factor" y el formato de en el que aparecen las fechas es del tipo "%m-%d-%Y %H:%M:%S AM/PM". Transformamos en una variable de clase Posixct y con un formato "%m-%d-%Y %H:%M:%S". Aplicamos a ambos conjuntos de datos, aunque a partir de ahora trabajemos sólo con el primer conjunto.

```
df_Chicago$Date <- mdy_hms(df_Chicago$Date)
```

Dividimos la columna "Date" en dos que serán "Day" y "Hour", y eliminamos la primera.

```
df_Chicago$Day <- as.Date(df_Chicago$Date)
df_Chicago$Hour <- times(format(df_Chicago$Date,"%H:%M:%S"))
df_Chicago$Date <- NULL
```

Creamos dos nuevas variables "Weekday" y "Time_slot", que nos permiten trabajar mejor para hacer el modelo.

```
df_Chicago$Nameday <- weekdays(df_Chicago$Day, abbreviate= FALSE)

time.tag <- chron(times= c('00:00:00', '08:00:00', '16:00:00','23:59:00'))
```

```
df_Chicago$Time_slot <- cut(df_Chicago$Hour, breaks= time.tag,
  labels= c('noche','mañana','tarde'), include.lowest=TRUE)
```

Determinación del número de comisarías

El primer objetivo de nuestro estudio es determinar el número de comisarías que es necesario en Chicago para absorber el número de delitos que se producen.

Puesto que tenemos el tipo de delito y sabemos si el delicuente ha sido o no arrestado, calculamos la probabilidad de arresto para cada tipo de delito.

```
z=table(df_Chicago$Primary.Type,df_Chicago$Arrest)
p=z[,2]/apply(z,1,sum)
sort(p)
```

```
##                          BURGLARY                   CRIMINAL DAMAGE
##                        0.05423044                        0.06594334
##                MOTOR VEHICLE THEFT                       KIDNAPPING
##                        0.07015170                        0.09116279
##                           ROBBERY                            ARSON
##                        0.09689093                        0.09885057
##                  HUMAN TRAFFICKING                             THEFT
##                        0.10000000                        0.11259122
##                      NON-CRIMINAL               CRIM SEXUAL ASSAULT
##                        0.11475410                        0.12099079
##                DECEPTIVE PRACTICE       OFFENSE INVOLVING CHILDREN
##                        0.12858062                        0.16572238
##                          STALKING                      INTIMIDATION
##                        0.17571059                        0.18040435
##                     OTHER OFFENSE                           BATTERY
##                        0.21459197                        0.23170826
##                           ASSAULT                       SEX OFFENSE
##                        0.23662689                        0.25673569
##                          HOMICIDE          OTHER NARCOTIC VIOLATION
##                        0.32501007                        0.70000000
##                  CRIMINAL TRESPASS            PUBLIC PEACE VIOLATION
##                        0.70586621                        0.76142912
##                         OBSCENITY                 WEAPONS VIOLATION
##                        0.79289941                        0.80038862
## CONCEALED CARRY LICENSE VIOLATION  INTERFERENCE WITH PUBLIC OFFICER
##                        0.85714286                        0.94784876
##               LIQUOR LAW VIOLATION                          NARCOTICS
##                        0.97873444                        0.99313904
##                      PROSTITUTION                          GAMBLING
```

```
##                         0.99550621                         0.99683401
##                 PUBLIC INDECENCY
##                         1.00000000
```

Dividimos en 3 categorías, según la probabilidad de ser arrestado en función del Primary.Type del delito (ej: Public Indecency te garantiza el arresto)

- Prob. baja <15%
- Prob. media 15%<p<70%
- Prob.alta >70%

```
length(unique(df_Chicago$Primary.Type))
```

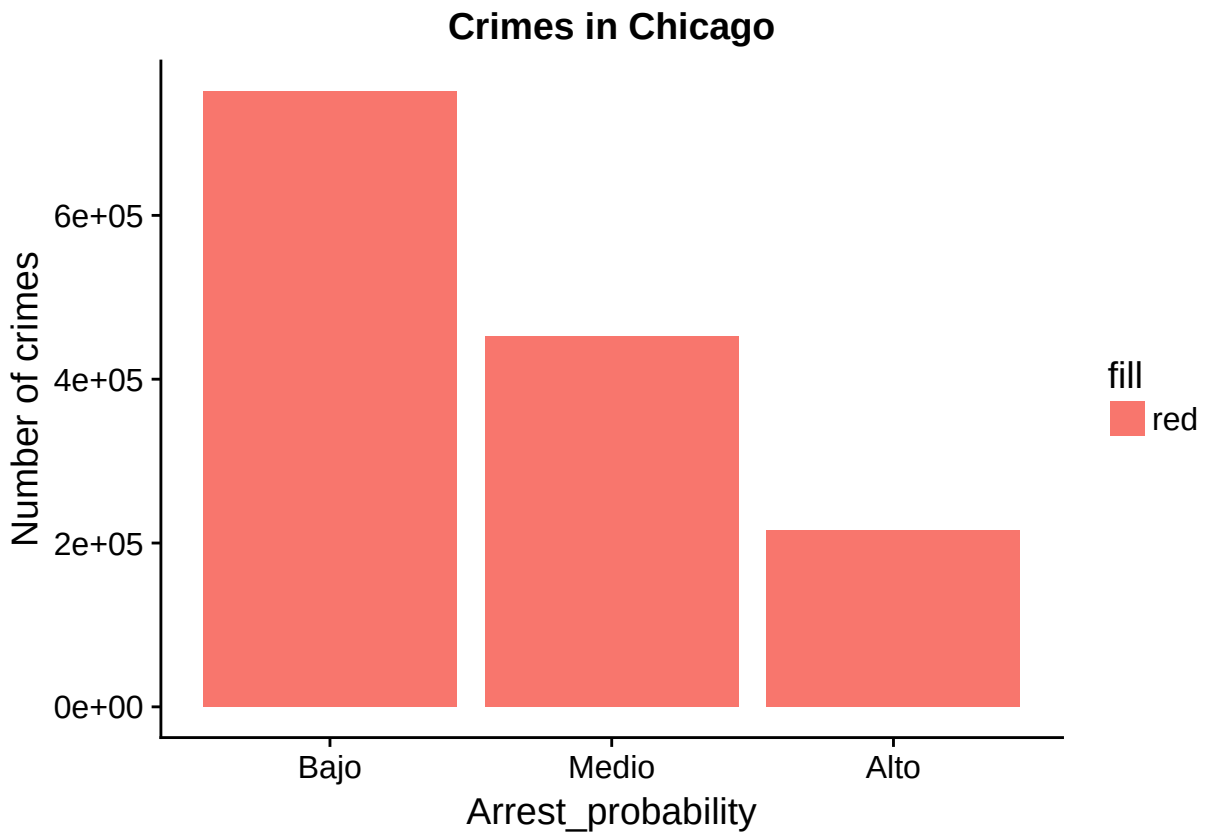
```
## [1] 31
```

```
Alto <- c("CONCEALED CARRY LICENSE VIOLATION","GAMBLING",
  "INTERFERENCE WITH PUBLIC OFFICER","LIQUOR LAW VIOLATION","NARCOTICS",
  "OBSCENITY","OTHER NARCOTIC VIOLATION","PUBLIC PEACE VIOLATION",
  "CRIMINAL TRESPASS","PROSTITUTION","PUBLIC INDECENCY","WEAPONS VIOLATION")

Medio <- c("ASSAULT","BATTERY","HOMICIDE","INTIMIDATION",
  "OFFENSE INVOLVING CHILDREN","OTHER OFFENSE","SEX OFFENSE","STALKING")

Bajo <- c("ARSON","BURGLARY","CRIM SEXUAL ASSAULT","CRIMINAL DAMAGE",
  "DECEPTIVE PRACTICE","HUMAN TRAFFICKING","KIDNAPPING","MOTOR VEHICLE THEFT",
  "NON-CRIMINAL","ROBBERY","THEFT")
```

Ahora creamos una nueva columna/variable que se llame "prob.arr_vs_type"

```
df_Chicago$Arrest_probability <- df_Chicago$Primary.Type

df_Chicago$Arrest_probability <-ifelse(df_Chicago$Primary.Type %in% Alto,
  "Alto",df_Chicago$Arrest_probability)
df_Chicago$Arrest_probability <-ifelse(df_Chicago$Primary.Type %in% Medio,
  "Medio",df_Chicago$Arrest_probability)
df_Chicago$Arrest_probability <-ifelse(df_Chicago$Primary.Type %in% Bajo,
  "Bajo",df_Chicago$Arrest_probability)
```

```
#Grafico Nº de Crimenes segun el % de ser arrestado
df_Chicago$Arrest_probability <- factor(df_Chicago$Arrest_probability,
  levels=c("Bajo","Medio","Alto"))
qplot(df_Chicago$Arrest_probability, xlab = 'Arrest_probability',
  main ='Crimes in Chicago', fill="red") + scale_y_continuous('Number of crimes')
```



Para nuestros cálculos, cambiamos los niveles de la probabilidad a numérico

```
df_Chicago$Arrest_probability <- as.numeric(factor(df_Chicago$Arrest_probability,
  levels=c("Bajo","Medio","Alto")))
```

Queremos determinar el número de comisarías que son necesarias en Chicago. Para ello utilizaremos un algoritmo de clúster jerárquico.

Vamos a realizar un muestreo con nuestros datos y crearemos 2 matrices de similaridad: una para las distancias y otra para la probabilidad de arresto.

Matriz de similaridad para las distancias

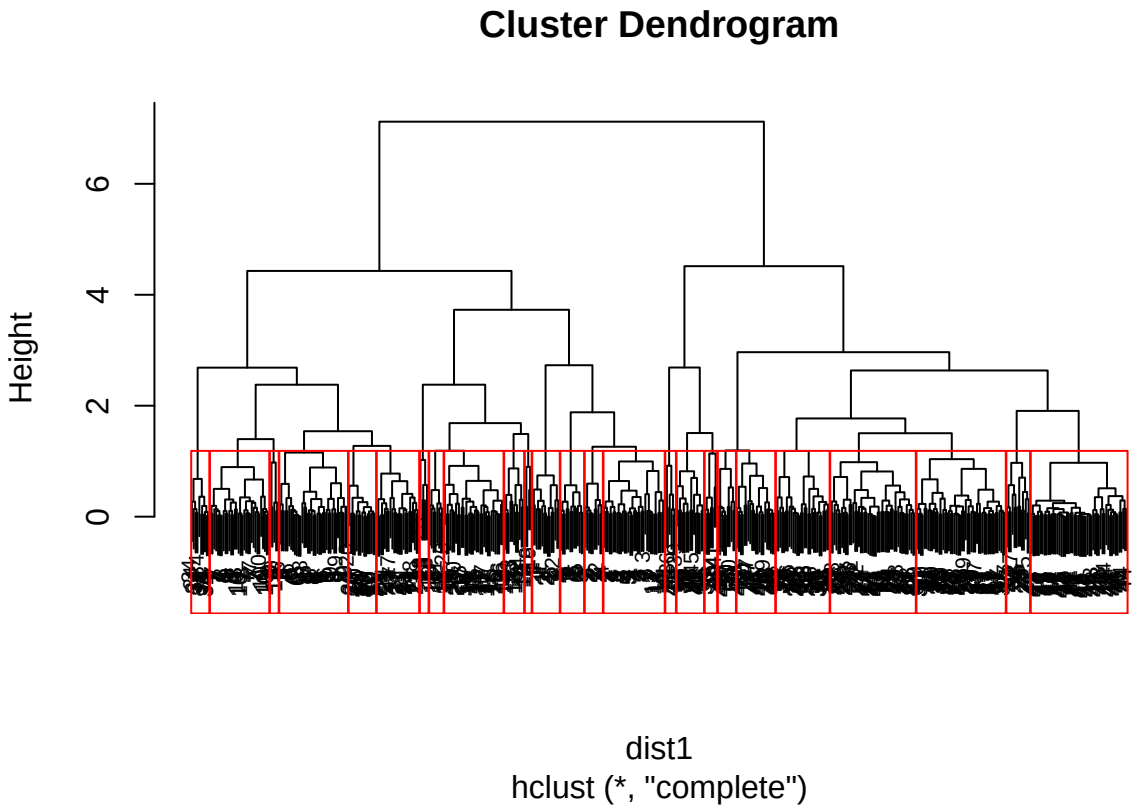
Nos quedamos con las variables X.Coordinate, Y.Coordinate y Ward. Esta última variable es una variable de localización geográfica (numérica) que hace referencia a una subdivisión municipal, independiente de las divisiones policiales. Empleamos esta para intentar obtener un clúsuter desvinculado de las localizaciones o divisiones policiales que nos proporciona el dataset.

```
dim(df_Chicago)
```

```
## [1] 1419482      18
```

```
set.seed(1)
ind=sample(1:1419482, 500)
```

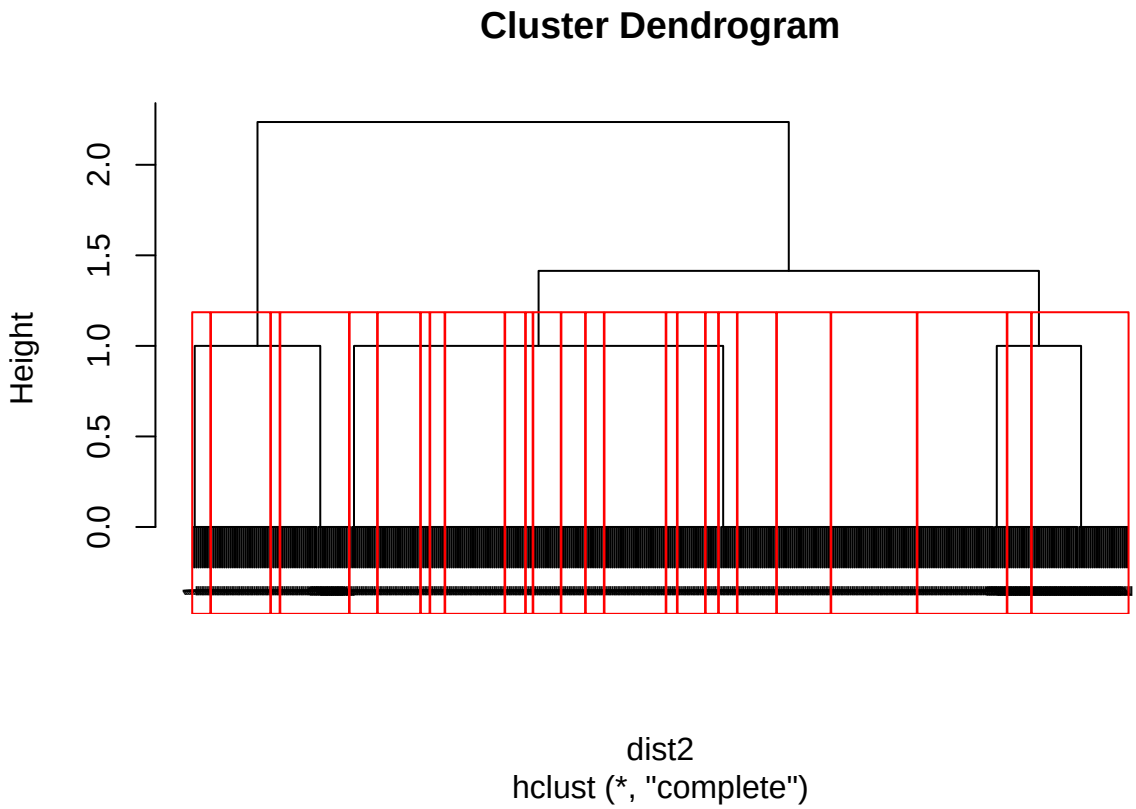
```
chicagotest <- df_Chicago[,c("Ward","X.Coordinate","Y.Coordinate")]
chicago.cl=chicagotest[ind,1:3]
chicago.cl <- scale(chicago.cl)
etiquetas=chicagotest[ind,1]
dist1 <- dist(chicago.cl,method = "euclidean")
h1 <- hclust(dist1,method = "complete")
plot(h1, labels=etiquetas, cex=0.7)
groups <- cutree(h1, k=25)
rect.hclust(h1, k=25, border="red")
```



Matriz de similaridad para la probabilidad de arresto En este caso nos quedamos con dos de las variables nuevas que hemos creado, la gravedad del delito de acuerdo a la clasificación del FBI ("Seriousness") y la probabilidad de arresto según el tipo de delito ("Arrest_probability").

```
ind=sample(1:1419482, 500)
chicagotest2 <- df_Chicago[,c("Seriousness", "Arrest_probability")]
chicago.cl2=chicagotest2[ind,1:2]
etiquetas2=chicagotest2[ind,1]
dist2 <- dist(chicago.cl2,method = "euclidean")
h2 <- hclust(dist2,method = "complete")
plot(h2, labels=etiquetas2, cex=0.7)
```

```
groups <- cutree(h1, k=25)
rect.hclust(h1, k=25, border="red")
```

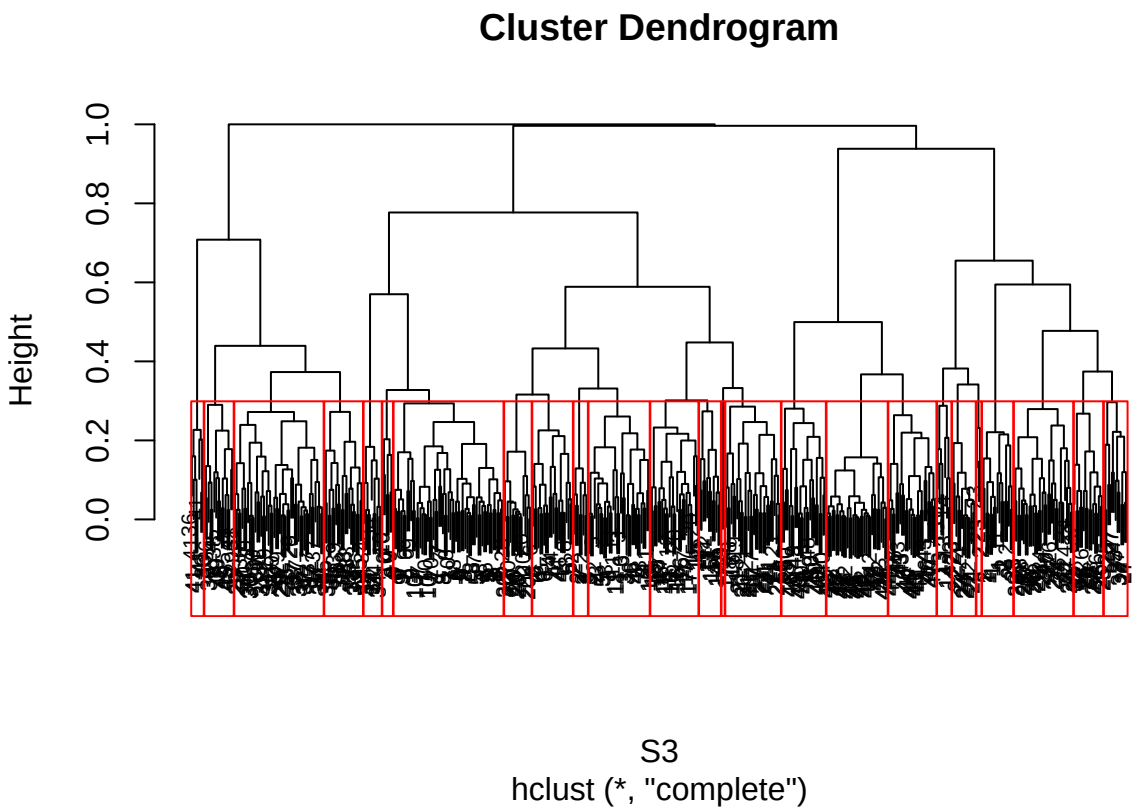


Normalizamos las matrices:

```
S <- function(x){(x-min(x))/(max(x)-min(x))}

S1 <- as.dist(apply(as.matrix(dist1),2 , S))

S2 <- as.dist(apply(as.matrix(dist2),2 , S))
```

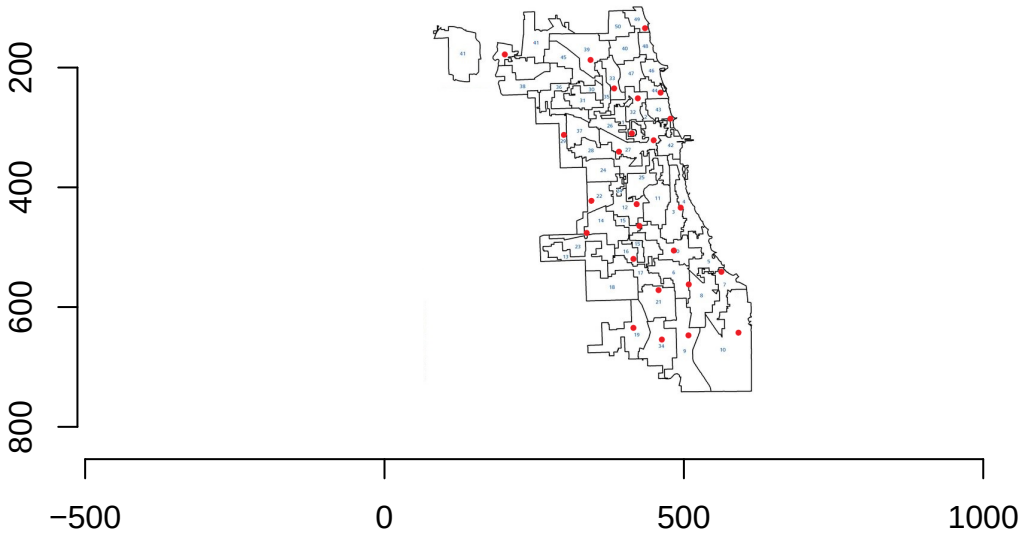
Una vez hemos normalizado las matrices, asignamos peso a cada una de ellas. Consideramos que la localización debe tener un mayor peso que la probabilidad de arresto.

```
S3 <- S1*0.9 + S2*0.1
h3 <- hclust(S3,method = "complete")
plot(h3, labels=etiquetas,cex=0.7)
groups <- cutree(h3, k=25)
rect.hclust(h3, k=25, border="red")
```

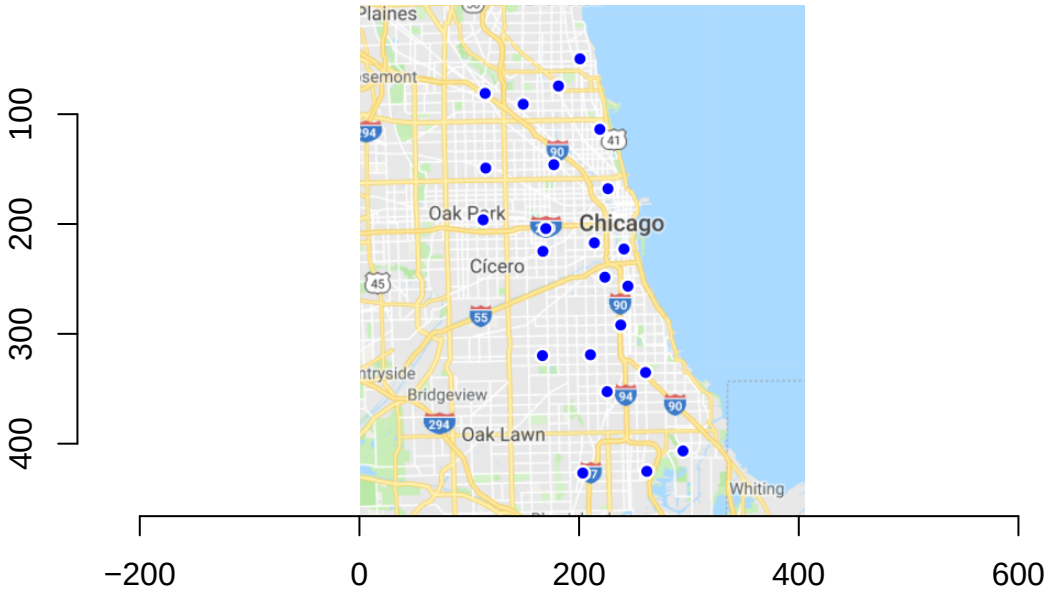


Distribuimos las 25 comisarías de acuerdo al clúster.

```
comisarias_estimadas <- load.image("comisarías_estimadas.png")
comisarias_reales <- load.image("comisarías_reales.PNG")

plot(comisarias_estimadas)
```




```
plot(comisarias_reales)
```



SEGUNDA PARTE

MODELO PARA LA PREDICCIÓN DEL ARRESTO

Dado que el conjunto de datos que tenemos es muuy grande, dIvidimos en dos grupos:

- años de 2012 a 2014
- años de 2015 a 2016

```
df_Chicago1 <- df_Chicago %>% filter(Year >= "2012" & Year <= "2014")
df_Chicago2 <- df_Chicago %>% filter(Year == "2015" | Year == "2016")
```

Y trabajaremos con el segundo de ellos, pues su peso es menor y es más cómodo para trabajar.

1. Limpieza de datos

Eliminamos las variables que no vamos a necesitar

```
drop2 <- c("Case.Number", "Primary.Type","Description","Location.Description",
           "Ward","Community.Area","X.Coordinate","Y.Coordinate",
           "Arrest_probability","Hour")
df_Chicago2 <- df_Chicago2[, !(names(df_Chicago2) %in% drop2)]
```

División según las comisarías:

```
North <- c("11","14","15","16","17","19","20","24","25","31")
Central <- c("1","2","3","8","9","10","12","18")
South <- c("4","5","6","7","22")

df_Chicago2$District<-ifelse(df_Chicago2$District  %in% North, "North",
                             df_Chicago2$District)
df_Chicago2$District<- ifelse(df_Chicago2$District  %in% Central, "Central",
                              df_Chicago2$District)
df_Chicago2$District<- ifelse(df_Chicago2$District  %in% South, "South",
                              df_Chicago2$District)
```

Trabajamos con meses y días:

```
Weekday <- c("lunes","martes","miércoles","jueves","viernes")
Weekend <- c("sábado","domingo")

df_Chicago2$Is.weekend <-ifelse(df_Chicago2$Nameday  %in% Weekday, "False","True")
```

```
df_Chicago2$Month <- months(df_Chicago2$Day)

primero <- c("enero","febrero","marzo")
segundo <- c("abril","mayo","junio")
tercero <- c("julio","agosto","septiembre")
cuarto <- c("octubre","noviembre","diciembre")

df_Chicago2$Trimestre <-ifelse(df_Chicago2$Month  %in% primero, "1º"," ")
df_Chicago2$Trimestre <-ifelse(df_Chicago2$Month  %in% segundo, "2º",
                               df_Chicago2$Trimestre)
df_Chicago2$Trimestre <-ifelse(df_Chicago2$Month  %in% tercero, "3º",
                               df_Chicago2$Trimestre)
df_Chicago2$Trimestre <-ifelse(df_Chicago2$Month  %in% cuarto, "4º",
                               df_Chicago2$Trimestre)
```

Ya tenemos nuestro dataframe definitivo, por lo que ya podemos empezar a aplicar modelos.

Análisis descriptivo de los datos.

En primer lugar, transformamos en factores aquellas variables que no lo son. Y con la variable Arrest, reordenamos los factores, de forma que TRUE sea "1" y FALSE sea "2".

```
levels(df_Chicago2$Arrest) <- c("2","1")
df_Chicago2$District <- as.factor(df_Chicago2$District)
df_Chicago2$Year <- as.factor(df_Chicago2$Year)
df_Chicago2$Seriousness <- as.factor(df_Chicago2$Seriousness)
df_Chicago2$Nameday <- as.factor(df_Chicago2$Nameday)
df_Chicago2$Is.weekend <- as.factor(df_Chicago2$Is.weekend)
```

```
df_Chicago2$Month <- as.factor(df_Chicago2$Month)
df_Chicago2$Trimestre <- as.factor(df_Chicago2$Trimestre)
summary(df_Chicago2)
```

```
##  Arrest      Domestic         District        Year        Seriousness
##  2:394496   False:428601   Central:200501   2015:259770   1:399555
##  1:116512   True : 82407   North  :182836   2016:251238   2:111453
##                            South  :127671
##
##
##
##
##       Day                   Nameday       Time_slot      Is.weekend
##  Min.   :2015-01-01   domingo  :71675   noche :108584   False:364792
##  1st Qu.:2015-07-04   jueves   :72845   mañana:193321   True :146216
##  Median :2015-12-24   lunes    :72367   tarde :209103
##  Mean   :2016-01-01   martes   :71523
##  3rd Qu.:2016-07-06   miércoles:71749
##  Max.   :2016-12-31   sábado   :74541
##                       viernes  :76308
##         Month        Trimestre
##  agosto    : 48054   1º:113004
##  julio     : 47390   2º:131962
##  mayo      : 45768   3º:140597
##  junio     : 45739   4º:125445
##  octubre   : 45215
##  septiembre: 45153
##  (Other)   :233689
```

Comprobamos si los datos están desequilibrados con respecto al arresto (TRUE=1 FALSE=2), que es la variable que queremos predecir.

```
table(df_Chicago2$District,df_Chicago2$Arrest)
```

```
##
##                2      1
##   Central 160043  40458
##   North   138552  44284
##   South    95901  31770
```

```
table(df_Chicago2$Domestic,df_Chicago2$Arrest)
```

```
##
##               2      1
##   False 327812 100789
##   True   66684  15723
```

```r
table(df_Chicago2$Year,df_Chicago2$Arrest)
```

```
##
##              2      1
##   2015 191464  68306
##   2016 203032  48206
```

```r
table(df_Chicago2$Seriousness,df_Chicago2$Arrest)
```

```
##
##          2      1
##   1 308291  91264
##   2  86205  25248
```

```r
table(df_Chicago2$Nameday,df_Chicago2$Arrest)
```

```
##
##                 2     1
##   domingo   55525 16150
##   jueves    56074 16771
##   lunes     56223 16144
##   martes    54949 16574
##   miércoles 55143 16606
##   sábado    57745 16796
##   viernes   58837 17471
```

```r
table(df_Chicago2$Time_slot,df_Chicago2$Arrest)
```

```
##
##                2      1
##   noche    89519  19065
##   mañana 151390  41931
##   tarde  153587  55516
```

```r
table(df_Chicago2$Is.weekend,df_Chicago2$Arrest)
```

```
##
##               2      1
##   False 281226  83566
##   True  113270  32946
```

```r
table(df_Chicago2$Month,df_Chicago2$Arrest)
```

```
##
##                  2     1
##   abril      30020 10435
##   agosto     38151  9903
##   diciembre  32447  7156
```

```
##   enero      29430  9961
##   febrero    24186  8948
##   julio      37008 10382
##   junio      35201 10538
##   marzo      29520 10959
##   mayo       34706 11062
##   noviembre  32397  8230
##   octubre    35776  9439
##   septiembre 35654  9499
```

```
table(df_Chicago2$Trimestre,df_Chicago2$Arrest)
```

```
##
##           2      1
##   1º  83136  29868
##   2º  99927  32035
##   3º 110813  29784
##   4º 100620  24825
```

Dividimos nuestro conjunto de datos en "train" y "test"

```
n_data=dim(df_Chicago2)[1]
n_train=round(0.7*n_data)
n_test=n_data-n_train

# Índices sobre los que vamos a muestrear

indices=1:n_data
indices_train= sample(indices,n_train)
indices_test=indices[-indices_train]
```

Construimos los dos conjuntos

```
Chicago_train=df_Chicago2[indices_train,]
Chicago_test=df_Chicago2[indices_test,]
```

Nuestro objetivo es crear un modelo de predicción del arresto.

```
summary(Chicago_train)
```

```
##  Arrest      Domestic         District        Year        Seriousness
##  2:276204   False:300026   Central:140271   2015:181967   1:279762
##  1: 81502   True : 57680   North  :127985   2016:175739   2: 77944
##                            South  : 89450
##
##
##
##
```

```
##       Day                  Nameday       Time_slot      Is.weekend
##  Min.   :2015-01-01   domingo  :50164   noche : 76101   False:255325
##  1st Qu.:2015-07-04   jueves   :50951   mañana:135540   True :102381
##  Median :2015-12-24   lunes    :50646   tarde :146065
##  Mean   :2016-01-01   martes   :50213
##  3rd Qu.:2016-07-05   miércoles:50169
##  Max.   :2016-12-31   sábado   :52217
##                       viernes  :53346
##         Month        Trimestre
##  agosto    : 33468   1º:79086
##  julio     : 33214   2º:92269
##  junio     : 31952   3º:98444
##  mayo      : 31921   4º:87907
##  septiembre: 31762
##  octubre   : 31707
##  (Other)   :163682
```

En primer lugar, antes de aplicar ningún modelo, definimos una función para dibujar la curva ROC.

```
rocplot = function(pred, truth, ...) {
predob = prediction(pred, truth)
perf = performance(predob, "tpr", "fpr")
auc = as.numeric(performance(predob,"auc")@y.values)
plot(perf, main= paste("Area=",round(auc,2),...))
}
```

2. Regresión logística

Creamos varios modelos, probando con distintas variables. Para cada uno de ellos dibujaremos su curva ROC correspondiente, lo que nos ayudará a determinar cuál es el más adecuado.

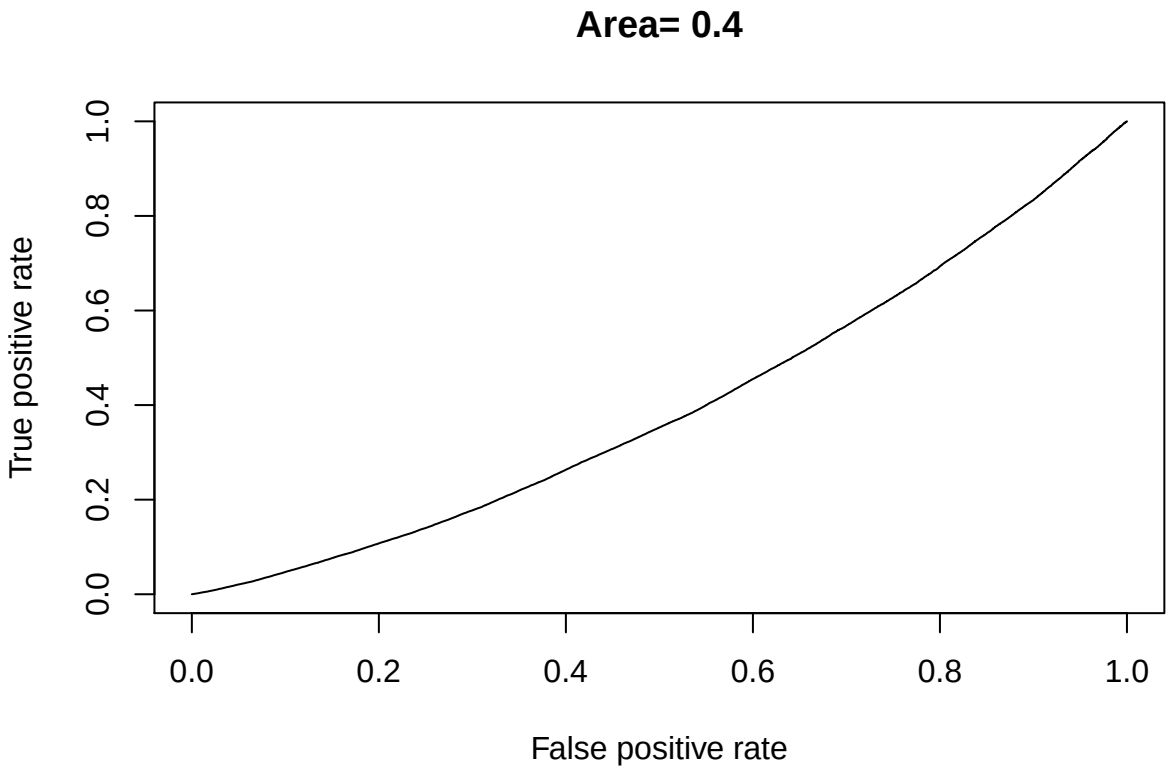
En primer lugar, creamos un modelo en el que usaremos las variables "Nameday" y "Month", sin tener en cuenta si son o no fin de semana y sin agrupar por trimestres.

```
modelarrest1 <- glm(Arrest ~ Domestic + District + Year + Seriousness +
                      Nameday + Time_slot + Month,
                      family = binomial,data = Chicago_train)
summary(modelarrest1)
```

```
##
## Call:
## glm(formula = Arrest ~ Domestic + District + Year + Seriousness +
##     Nameday + Time_slot + Month, family = binomial, data = Chicago_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0099  -0.7564  -0.6542  -0.5091   2.2672
```

```
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.2853382  0.0201839 -63.681  < 2e-16 ***
## DomesticTrue     -0.2733190  0.0116271 -23.507  < 2e-16 ***
## DistrictNorth     0.2286946  0.0094064  24.313  < 2e-16 ***
## DistrictSouth     0.2661855  0.0103492  25.720  < 2e-16 ***
## Year2016         -0.4021105  0.0081311 -49.453  < 2e-16 ***
## Seriousness2     -0.0281015  0.0098005  -2.867  0.00414 **
## Namedayjueves    -0.0002794  0.0152053  -0.018  0.98534
## Namedaylunes     -0.0439694  0.0153179  -2.870  0.00410 **
## Namedaymartes     0.0079804  0.0152474   0.523  0.60070
## Namedaymiércoles  0.0075988  0.0152549   0.498  0.61840
## Namedaysábado    -0.0123734  0.0151657  -0.816  0.41457
## Namedayviernes   -0.0141150  0.0150574  -0.937  0.34854
## Time_slotmañana   0.2424280  0.0117306  20.666  < 2e-16 ***
## Time_slottarde    0.5194735  0.0113322  45.840  < 2e-16 ***
## Monthagosto      -0.2979147  0.0193389 -15.405  < 2e-16 ***
## Monthdiciembre   -0.4576809  0.0208366 -21.965  < 2e-16 ***
## Monthenero       -0.0319301  0.0196060  -1.629  0.10340
## Monthfebrero      0.0840944  0.0202363   4.156 3.24e-05 ***
## Monthjulio       -0.1982225  0.0191169 -10.369  < 2e-16 ***
## Monthjunio       -0.1491767  0.0191751  -7.780 7.27e-15 ***
## Monthmarzo        0.0605446  0.0192250   3.149  0.00164 **
## Monthmayo        -0.0754909  0.0189994  -3.973 7.09e-05 ***
## Monthnoviembre   -0.3154911  0.0202368 -15.590  < 2e-16 ***
## Monthoctubre     -0.2768726  0.0195328 -14.175  < 2e-16 ***
## Monthseptiembre  -0.2663828  0.0195064 -13.656  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 383934  on 357705  degrees of freedom
## Residual deviance: 375959  on 357681  degrees of freedom
## AIC: 376009
##
## Number of Fisher Scoring iterations: 4
```

```
fitted_log1 <- predict(modelarrest1, Chicago_train, decision.values = TRUE)

rocplot(fitted_log1, Chicago_train[,"Arrest"])
```



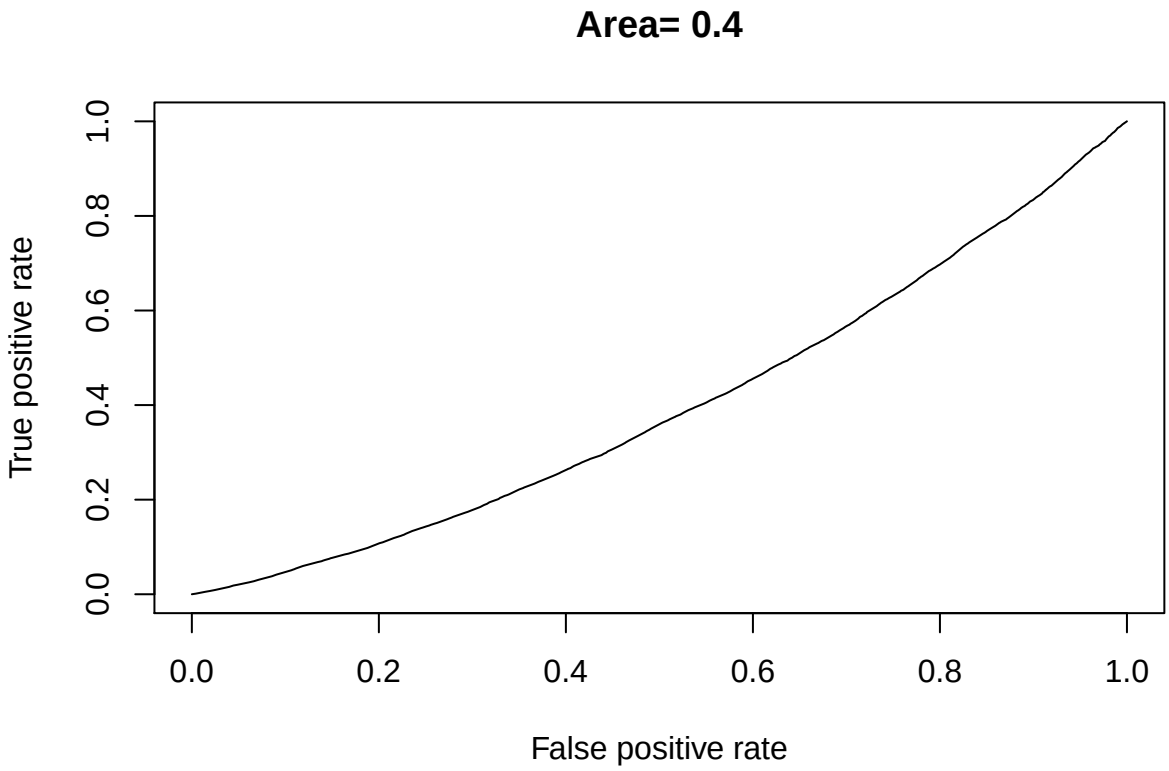
Probamos ahora con los meses agrupados por trimestre y los días de la semana en función de si es o no fin de semana.

```
modelarrest2 <- glm(Arrest ~ Domestic + District + Year + Seriousness +
                      Is.weekend + Time_slot + Trimestre,
                    family = binomial,data = Chicago_train)
summary(modelarrest2)
```

```
##
## Call:
## glm(formula = Arrest ~ Domestic + District + Year + Seriousness +
##     Is.weekend + Time_slot + Trimestre, family = binomial, data = Chicago_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9857  -0.7492  -0.6558  -0.5091   2.2068
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.259964   0.014292 -88.161  < 2e-16 ***
## DomesticTrue    -0.273978   0.011622 -23.575  < 2e-16 ***
## DistrictNorth    0.229332   0.009403  24.389  < 2e-16 ***
```

```
## DistrictSouth    0.267260   0.010346  25.833  < 2e-16 ***
## Year2016        -0.401735   0.008126 -49.436  < 2e-16 ***
## Seriousness2    -0.027068   0.009797  -2.763  0.00573 **
## Is.weekendTrue   0.003113   0.008981   0.347  0.72890
## Time_slotmañana  0.243533   0.011724  20.773  < 2e-16 ***
## Time_slottarde   0.520233   0.011327  45.927  < 2e-16 ***
## Trimestre2º     -0.112387   0.011245  -9.994  < 2e-16 ***
## Trimestre3º     -0.289250   0.011334 -25.521  < 2e-16 ***
## Trimestre4º     -0.380500   0.011806 -32.229  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 383934  on 357705  degrees of freedom
## Residual deviance: 376181  on 357694  degrees of freedom
## AIC: 376205
##
## Number of Fisher Scoring iterations: 4
```

```r
fitted_log2 <- predict(modelarrest2, Chicago_train, decision.values = TRUE)

rocplot(fitted_log2, Chicago_train[,"Arrest"])
```



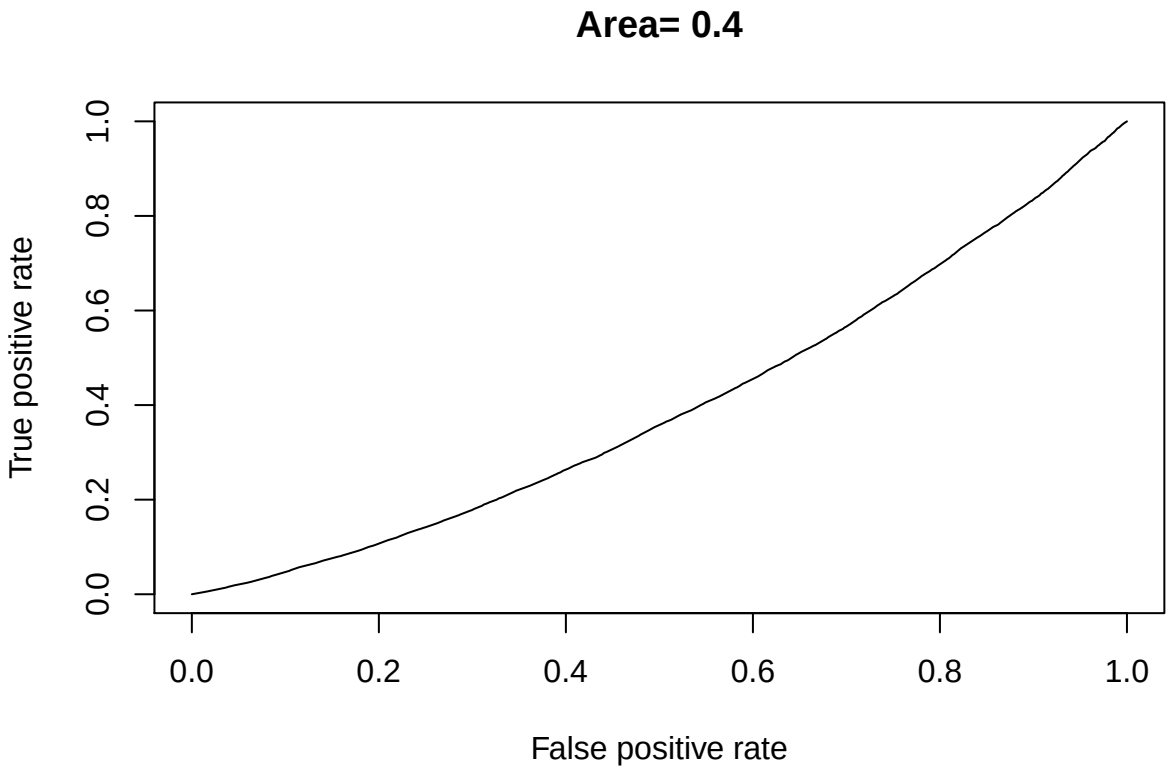
Puesto que en el primer modelo vemos que el lunes es un día significativo para el arresto, y que en el 2º modelo se ve que el fin de semana no lo es, hacemos un tercer modelo combinando ambos resultados.

```
modelarrest3 <- glm(Arrest ~ Domestic + District + Year + Seriousness +
                      (Nameday == "lunes") + Time_slot + Trimestre,
                    family = binomial,data = Chicago_train)
summary(modelarrest3)
```

```
##
## Call:
## glm(formula = Arrest ~ Domestic + District + Year + Seriousness +
##     (Nameday == "lunes") + Time_slot + Trimestre, family = binomial,
##     data = Chicago_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9870  -0.7508  -0.6568  -0.5099   2.2209
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -1.253552   0.014028 -89.359  < 2e-16 ***
```

```
## DomesticTrue           -0.273952   0.011614 -23.589  < 2e-16 ***
## DistrictNorth           0.229306   0.009403  24.385  < 2e-16 ***
## DistrictSouth           0.267364   0.010345  25.844  < 2e-16 ***
## Year2016               -0.401676   0.008126 -49.428  < 2e-16 ***
## Seriousness2           -0.027058   0.009797  -2.762 0.005745 **
## Nameday == "lunes"TRUE -0.040710   0.011646  -3.496 0.000473 ***
## Time_slotmañana         0.243738   0.011690  20.850  < 2e-16 ***
## Time_slottarde          0.520281   0.011308  46.011  < 2e-16 ***
## Trimestre2º            -0.112293   0.011245  -9.986  < 2e-16 ***
## Trimestre3º            -0.289269   0.011334 -25.522  < 2e-16 ***
## Trimestre4º            -0.380414   0.011806 -32.221  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 383934  on 357705  degrees of freedom
## Residual deviance: 376168  on 357694  degrees of freedom
## AIC: 376192
##
## Number of Fisher Scoring iterations: 4
```

```r
fitted_log3 <- predict(modelarrest3, Chicago_train, decision.values = TRUE)

rocplot(fitted_log3, Chicago_train[,"Arrest"])
```



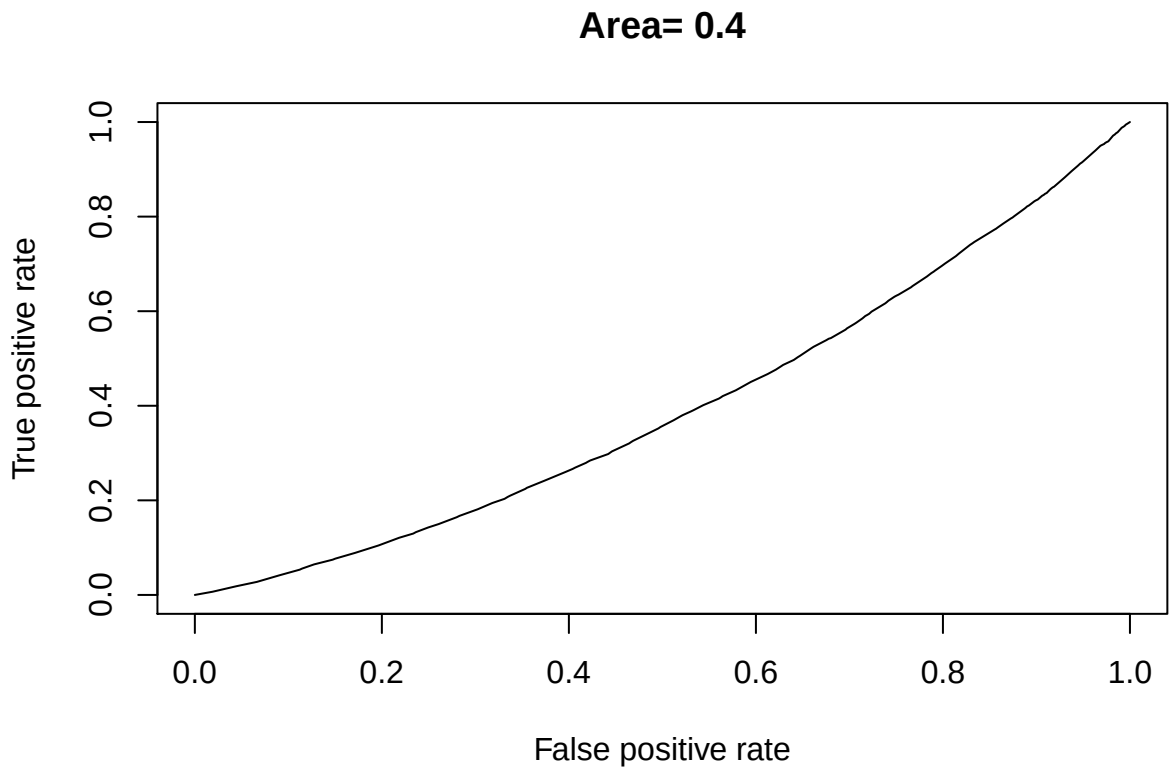
Probamos sin Seriousness y sin lunes

```
modelarrest4 <- glm(Arrest ~ Domestic + District + Year + + Time_slot +
                      Trimestre, family = binomial,data = Chicago_train)
summary(modelarrest4)
```

```
##
## Call:
## glm(formula = Arrest ~ Domestic + District + Year + +Time_slot +
##     Trimestre, family = binomial, data = Chicago_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9821  -0.7475  -0.6572  -0.5127   2.1974
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.264945   0.013774 -91.833   <2e-16 ***
## DomesticTrue    -0.273355   0.011612 -23.541   <2e-16 ***
## DistrictNorth    0.229246   0.009403  24.380   <2e-16 ***
## DistrictSouth    0.266399   0.010341  25.763   <2e-16 ***
## Year2016        -0.401826   0.008126 -49.448   <2e-16 ***
```

```
## Time_slotmañana  0.244058   0.011685  20.886   <2e-16 ***
## Time_slottarde   0.520120   0.011307  45.999   <2e-16 ***
## Trimestre2º     -0.112416   0.011245  -9.997   <2e-16 ***
## Trimestre3º     -0.289128   0.011334 -25.511   <2e-16 ***
## Trimestre4º     -0.380428   0.011806 -32.224   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 383934  on 357705  degrees of freedom
## Residual deviance: 376188  on 357696  degrees of freedom
## AIC: 376208
##
## Number of Fisher Scoring iterations: 4
```

```
fitted_log4 <- predict(modelarrest4, Chicago_train, decision.values = TRUE)

rocplot(fitted_log4, Chicago_train[,"Arrest"])
```

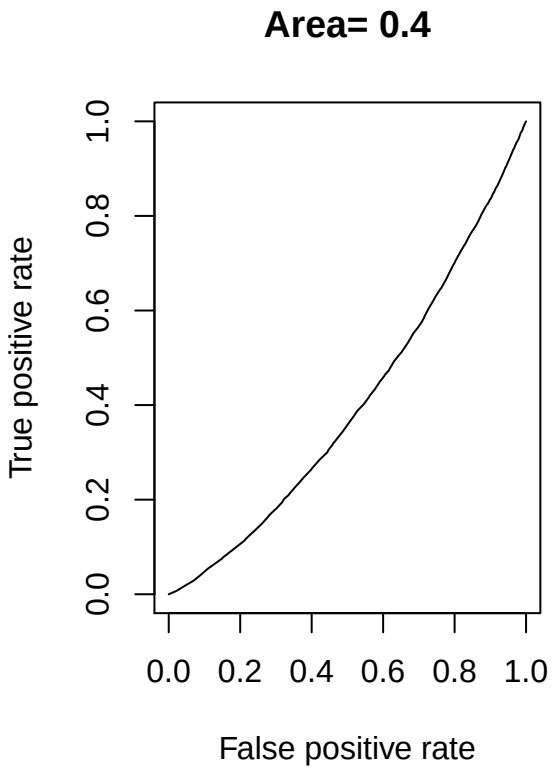


Comprobamos el último modelo con los datos de test

```
modelarrest_test <- glm(Arrest ~ Domestic + District + Year +
                     + Time_slot + Trimestre, family = binomial,data = Chicago_test)
summary(modelarrest_test)
```

```
##
## Call:
## glm(formula = Arrest ~ Domestic + District + Year + +Time_slot +
##     Trimestre, family = binomial, data = Chicago_test)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9893  -0.7494  -0.6574  -0.5131   2.1908
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.27091    0.02107 -60.304  < 2e-16 ***
## DomesticTrue    -0.25810    0.01766 -14.611  < 2e-16 ***
## DistrictNorth    0.22801    0.01438  15.860  < 2e-16 ***
## DistrictSouth    0.30020    0.01576  19.047  < 2e-16 ***
## Year2016        -0.40525    0.01240 -32.679  < 2e-16 ***
## Time_slotmañana  0.24940    0.01786  13.966  < 2e-16 ***
## Time_slottarde   0.51068    0.01727  29.573  < 2e-16 ***
## Trimestre2º     -0.11946    0.01716  -6.961 3.39e-12 ***
## Trimestre3º     -0.28489    0.01729 -16.474  < 2e-16 ***
## Trimestre4º     -0.37046    0.01801 -20.570  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 164739  on 153301  degrees of freedom
## Residual deviance: 161436  on 153292  degrees of freedom
## AIC: 161456
##
## Number of Fisher Scoring iterations: 4
```

```
fitted_logtest <- predict(modelarrest_test, Chicago_test, decision.values = TRUE)
par(mfrow = c(1, 2))
rocplot(fitted_logtest, Chicago_test[,"Arrest"])
```



Hemos localizado las variables más adecuadas para predecir el arresto. Sin embargo, como se puede ver en la curva ROC, el área bajo la curva es muy pequeña, por lo que no es el modelo más apropiado para nuestros datos.

*3. Modelo K-means y PCA

En primer lugar seleccionamos las variables con las que vamos a trabajar, eliminando nuestra variable objetivo, ya que posteriormente, lo interesante será observar cómo se distribuye ésta en cada uno de los clústers obtenidos. Previamente, hemos tenido que transformar todas nuestras variables cualitativas a cuantitativas debido a que k-means trabaja con este último tipo de variables.

```
df_Chicago3<- subset(df_Chicago2, select=c("District","Domestic","Seriousness",
                                           "Year","Time_slot","Is.weekend",
                                           "Trimestre"))
df_Chicago3$District<-as.numeric(df_Chicago3$District)
df_Chicago3$Domestic<-as.numeric(df_Chicago3$Domestic)
df_Chicago3$Year<-as.numeric(df_Chicago3$Year)
df_Chicago3$Time_slot<-as.numeric(df_Chicago3$Time_slot)
df_Chicago3$Is.weekend<-as.numeric(df_Chicago3$Is.weekend)
df_Chicago3$Trimestre<-as.numeric(df_Chicago3$Trimestre)
df_Chicago3$Seriousness<-as.numeric(df_Chicago3$Seriousness)
```

Además, conviene trabajar con una muestra (10k) ya que el dataset es muy pesado.

```
summary(df_Chicago3)
```

```
##     District        Domestic      Seriousness         Year
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :2.000   Median :1.000   Median :1.000   Median :1.000
##  Mean   :1.857   Mean   :1.161   Mean   :1.218   Mean   :1.492
##  3rd Qu.:2.000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:2.000
##  Max.   :3.000   Max.   :2.000   Max.   :2.000   Max.   :2.000
##    Time_slot       Is.weekend      Trimestre
##  Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:1.000   1st Qu.:2.000
##  Median :2.000   Median :1.000   Median :3.000
##  Mean   :2.197   Mean   :1.286   Mean   :2.545
##  3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:3.000
##  Max.   :3.000   Max.   :2.000   Max.   :4.000
```

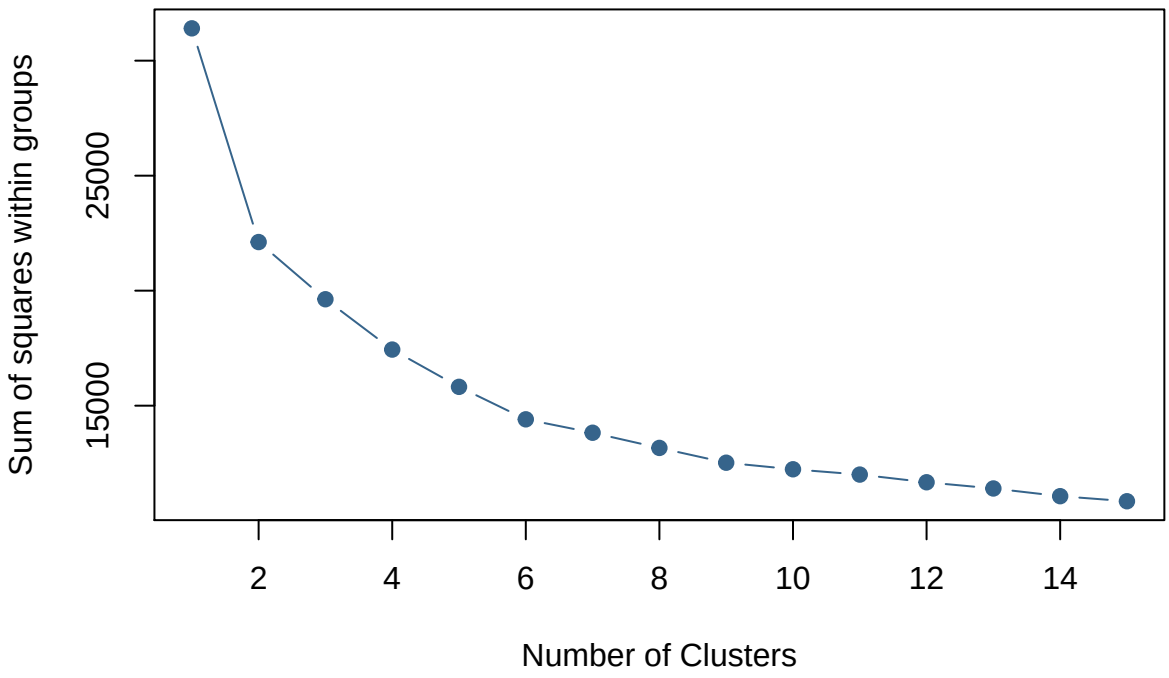
```
set.seed(1)
datos.st <- df_Chicago3[sample(1:nrow(df_Chicago3), 10000,replace=FALSE),]

dim(datos.st)
```

```
## [1] 10000     7
```

```
n = dim(datos.st)[1] #Número de Crímenes
p = dim(datos.st)[2] #Número de variables
```

Para elegir el número de clústers óptimo, lo que haremos será calcular la variabilidad dentro de los grupos para distintas ejecuciones de la función kmeans. En concreto, ejecutamos la función kmeans para un número de entre 2 y 15 clusters, y elegimos el número de clústers que proporcione descenso en la variabilidad y, a la vez, un número de clústers no demasiado grande. Para ello generamos un vector, que denominaremos SSW con las sumas de las varianzas dentro de los grupos que se obtienen después de cada ejecución del método, y lo representamos gráficamente.

```
#Inicializamos el vector
SSW <- vector(mode = "numeric", length = 15)
#Variabilidad de todos los datos
SSW[1] <- (n - 1) * sum(apply(datos.st,2,var))
#Variabilidad de cada modelo
for (i in 2:15) SSW[i] <- sum(kmeans(datos.st,centers=i,nstart=25)$withinss)
plot(1:15, SSW, type="b", xlab="Number of Clusters",
     ylab="Sum of squares within groups",pch=19, col="steelblue4")
```



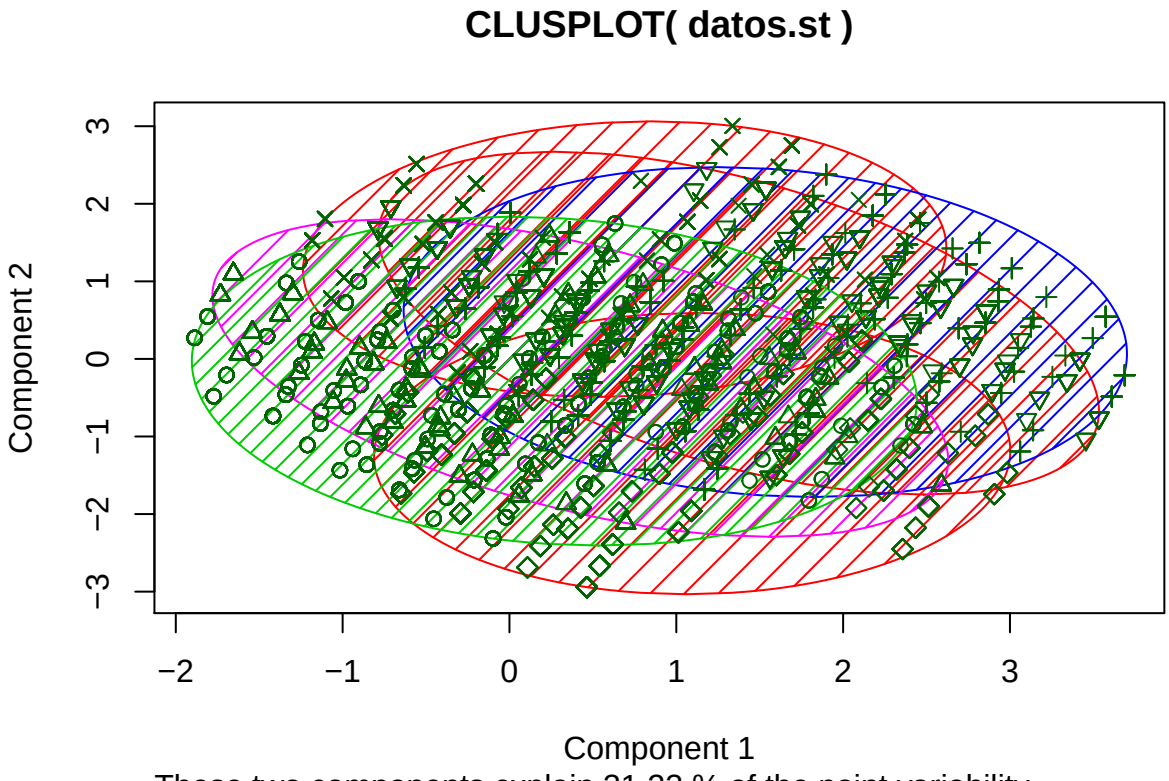
A continuación, tras utilizar el método de Elbow, gráficamente no observamos una elección clara, por lo que deberíamos probar con varias opciones. Probamos con 6 grupos y 25 arranques:

```
clusters6.datos <- kmeans(datos.st, 6, nstart = 25)
centroides<-aggregate(datos.st,by=list(clusters6.datos$cluster),FUN=mean)
```

Reducción de la dimensionalidad:

PCA y gráfico con las dos primeras componentes

```
 # Guardamos el vector con cada cluster
datos.clusters6 <- clusters6.datos$cluster
# PCA
clusplot(datos.st, datos.clusters6, color=TRUE, shade=TRUE,labels=0,lines=0)
```



Filtros Tras trabajar con los datos en bruto, hemos realizado unos filtros según los inconvenientes que nos hemos encontrado:

- Selección de una muestra: la distancia euclídea no es método adecuado para agrupar en nuestro caso y acumula la gran mayoría de datos en un solo clúster.
- Selección de la Zona Norte: buscamos para un área más específica, posibles resultados interesantes.

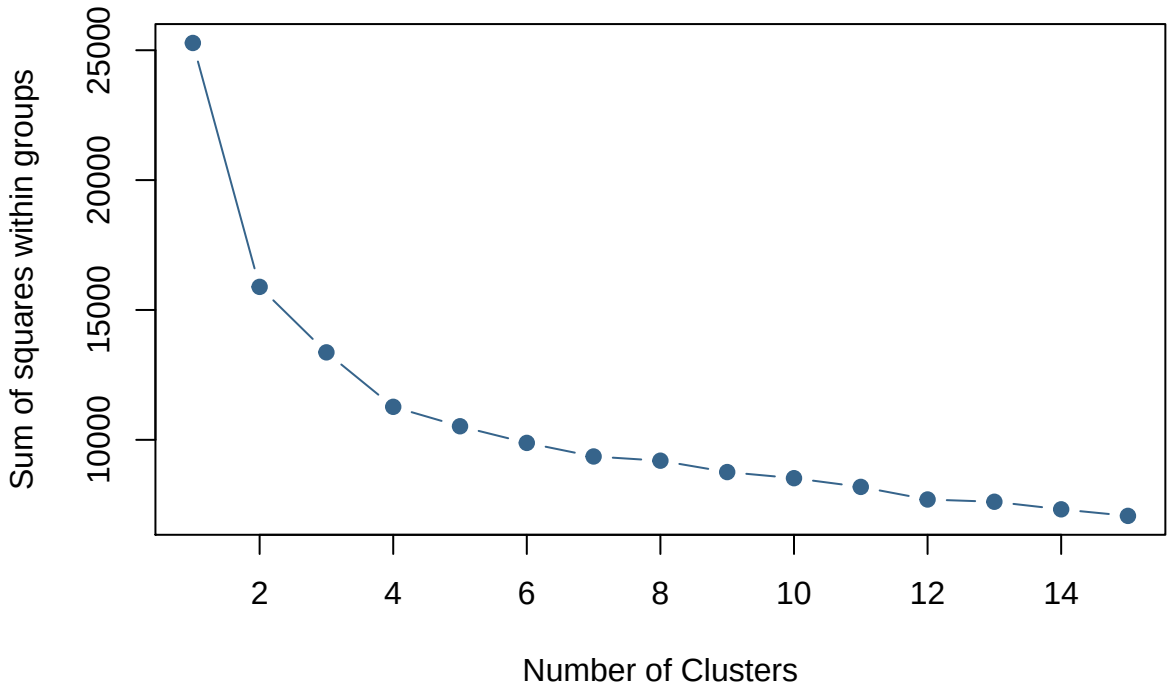
```
set.seed(1)
df_chicnorte <- df_Chicago3 %>% filter(District == "2")
df_chicnorte <- subset(df_chicnorte, select=c("Domestic","Seriousness",
                                              "Year","Time_slot","Is.weekend",
                                              "Trimestre"))
datos.st.norte <- df_chicnorte[sample(1:nrow(df_chicnorte), 10000,replace=FALSE),]

dim(datos.st.norte)
```

```
## [1] 10000     6
```

```
n2 = dim(datos.st.norte)[1] #Número de Crímenes
p2 = dim(datos.st.norte)[2] #Número de variables
```

Gráfico de Elbow II

```
 #Inicializamos el vector
SSW <- vector(mode = "numeric", length = 15)
#Variabilidad de todos los datos
SSW[1] <- (n2 - 1) * sum(apply(datos.st.norte,2,var))
#Variabilidad de cada modelo
for (i in 2:15) SSW[i] <- sum(kmeans(datos.st.norte,centers=i,nstart=25)$withinss)
plot(1:15, SSW, type="b", xlab="Number of Clusters",
     ylab="Sum of squares within groups",pch=19, col="steelblue4")
```



Cluster

Se escogen 6 grupos y 25 arranques diferentes. El modelo y los centroides correspondientes se pueden ver a continuación:

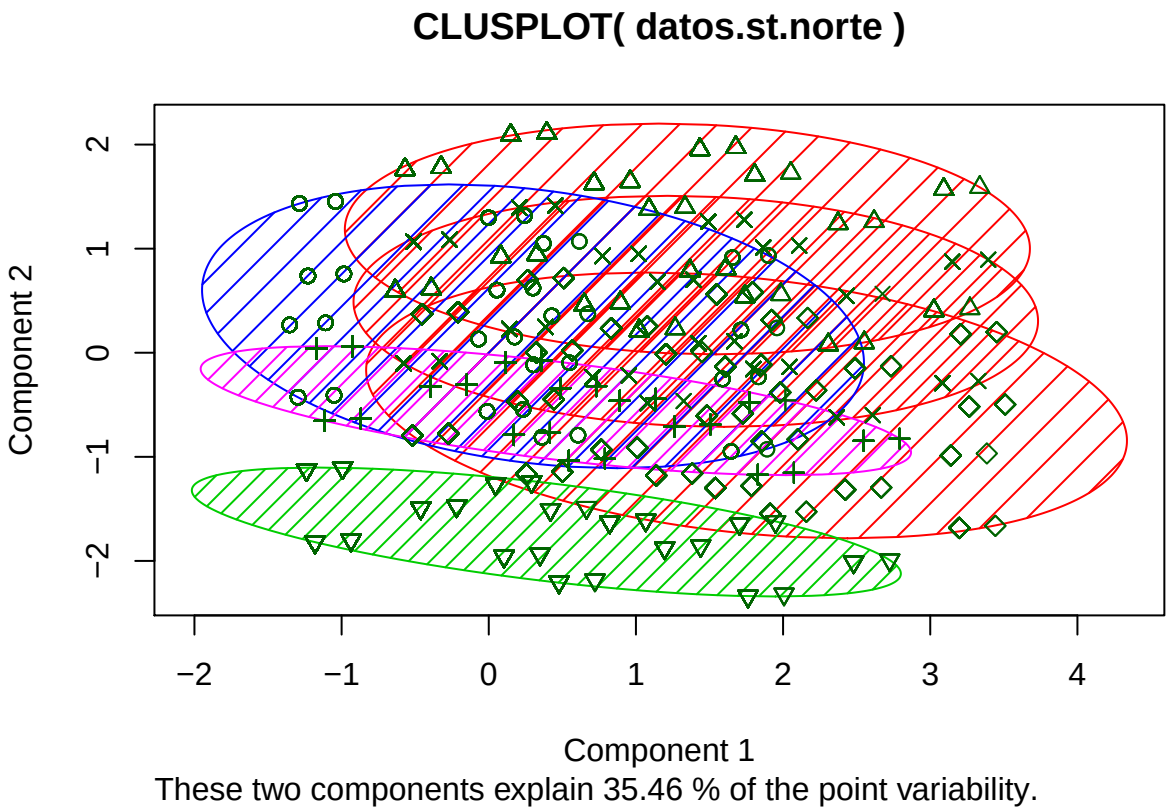
```
clusters6.datos.norte <- kmeans(datos.st.norte, 6, nstart = 25)
centroides <-aggregate(datos.st.norte,by=list(clusters6.datos.norte$cluster),
                       FUN=mean)
```

Reducción de la dimensionalidad II

Aplicamos PCA para el caso de 6 Clusters y los filtros anteriores aplicados, con el objetivo de describir los datos en términos de nuevas variables. Se observa claramente como hay un clúster (verde) que destaca en la parte superior, pero los demás también quedan definidos de

forma apilada (grafico para representar las dos primeras componentes agrupando según los clusters obtenidos previamente).

```
datos.clusters6.norte <- clusters6.datos.norte$cluster
clusplot(datos.st.norte, datos.clusters6.norte, color=TRUE, shade=TRUE,
         labels=0,lines=0)
```



En general, esto nos aporta un abanico enorme de líneas de investigación. - Deberiamos aplicar los métodos anteriores por cada una de las áreas definidas (Norte, Sur, o bien por distrito) - Ademas, conviene realizar el punto anterior segmentando por año.

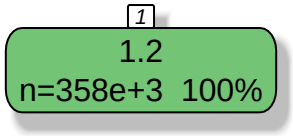
4. Árbol de decisión

```
forest_arrest1 <- rpart(as.numeric(Arrest) ~ ., data = Chicago_train)
summary(forest_arrest1)
```

```
## Call:
## rpart(formula = as.numeric(Arrest) ~ ., data = Chicago_train)
##   n= 357706
##
##            CP nsplit rel error xerror xstd
## 1 0.008225057      0         1      0    0
##
## Node number 1: 357706 observations
```

```
##   mean=1.227846, MSE=0.1759324
```

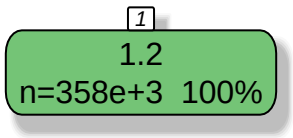
```
par(mfrow = c(1, 2))
fancyRpartPlot(forest_arrest1, sub = "")
```




```
forest_arrest2 <- rpart(as.numeric(Arrest) ~  Month, data = Chicago_train)
summary(forest_arrest2)
```

```
## Call:
## rpart(formula = as.numeric(Arrest) ~ Month, data = Chicago_train)
##   n= 357706
##
##            CP nsplit rel error xerror xstd
## 1 0.003341244      0         1      0    0
##
## Node number 1: 357706 observations
##   mean=1.227846, MSE=0.1759324
```

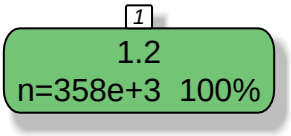
```
par(mfrow = c(1, 2))
fancyRpartPlot(forest_arrest2, sub = "")
```




```
forest_arrest3 <- rpart(as.numeric(Arrest) ~ Domestic + District + Year +
                      + Time_slot + Trimestre, data = Chicago_train)
summary(forest_arrest3)
```

```
## Call:
## rpart(formula = as.numeric(Arrest) ~ Domestic + District + Year +
##     +Time_slot + Trimestre, data = Chicago_train)
##   n= 357706
##
##            CP nsplit rel error xerror xstd
## 1 0.007096097      0         1      0    0
##
## Node number 1: 357706 observations
##   mean=1.227846, MSE=0.1759324
```

```
par(mfrow = c(1, 2))
fancyRpartPlot(forest_arrest3, sub = "")
```



Como se ve claramente en los 3 modelos, este método no es adecuado para el dataset que tenemos, pues en todos ellos se genera un único nodo. Puede deberse a que los datos no están desequilibrados.

5. SVM

Puesto que nuestro dataset es muy grande, tenemos que coger una muestra. Lo que haremos será coger una muestra del conjunto de datos de entrenamiento, y después validaremos el modelo con una muestra del conjunto de datos de test.

```
set.seed(1)
muestra_modelo <- Chicago_train[sample(1:nrow(Chicago_train),10000,
                                       replace = FALSE),]
```

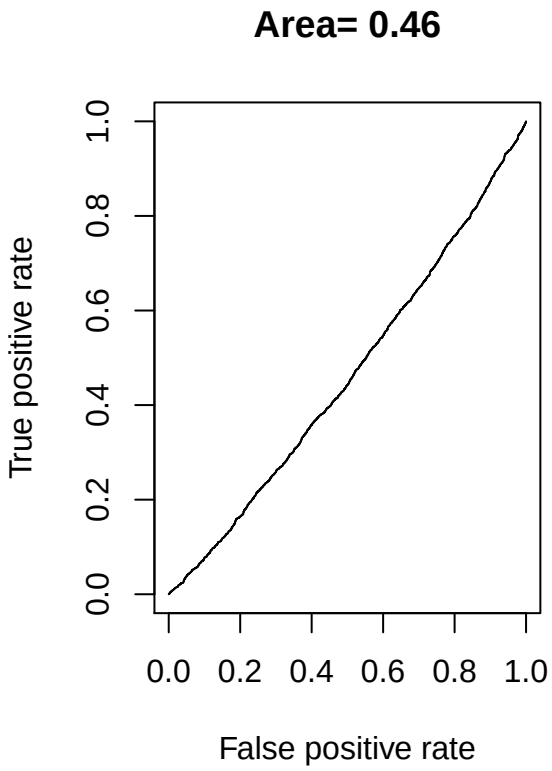
Realizamos un primer modelo de acuerdo a los que hemos aplicado con los métodos anteriores.

```
svmfit.opt = svm(Arrest ~ ., data =  muestra_modelo, kernel = "linear",
                 gamma = 1, cost = 1, decision.values = T,
                 type = "C-classification")
fitted = attributes(predict(svmfit.opt, muestra_modelo,
                            decision.values = TRUE))$decision
summary(svmfit.opt)
```

```
##
```

```
## Call:
## svm(formula = Arrest ~ ., data = muestra_modelo, kernel = "linear",
##     gamma = 1, cost = 1, decision.values = T, type = "C-classification")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##       gamma:  1
##
## Number of Support Vectors:  4769
##
##  ( 2526 2243 )
##
##
## Number of Classes:  2
##
## Levels:
##  2 1
```

```
par(mfrow = c(1, 2))
rocplot(fitted, muestra_modelo[,"Arrest"])
```



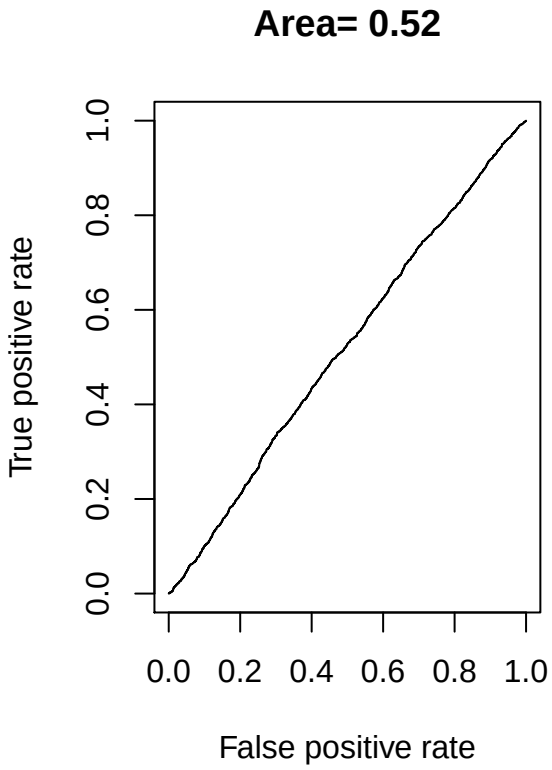
Cambiamos el valor de gamma y de cost

```
svmfit.opt2 = svm(Arrest ~ ., data =  muestra_modelo, kernel = "linear",
                  gamma = 10, cost = 10, decision.values = T,
                  type = "C-classification")
fitted2 = attributes(predict(svmfit.opt2, muestra_modelo,
                             decision.values = TRUE))$decision
summary(svmfit.opt2)
```

```
##
## Call:
## svm(formula = Arrest ~ ., data = muestra_modelo, kernel = "linear",
##     gamma = 10, cost = 10, decision.values = T, type = "C-classification")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  10
##       gamma:  10
##
## Number of Support Vectors:  6000
```

```
##
##  ( 3757 2243 )
##
##
## Number of Classes:  2
##
## Levels:
##  2 1
```

```r
par(mfrow = c(1, 2))
rocplot(fitted2, muestra_modelo[,"Arrest"])
```



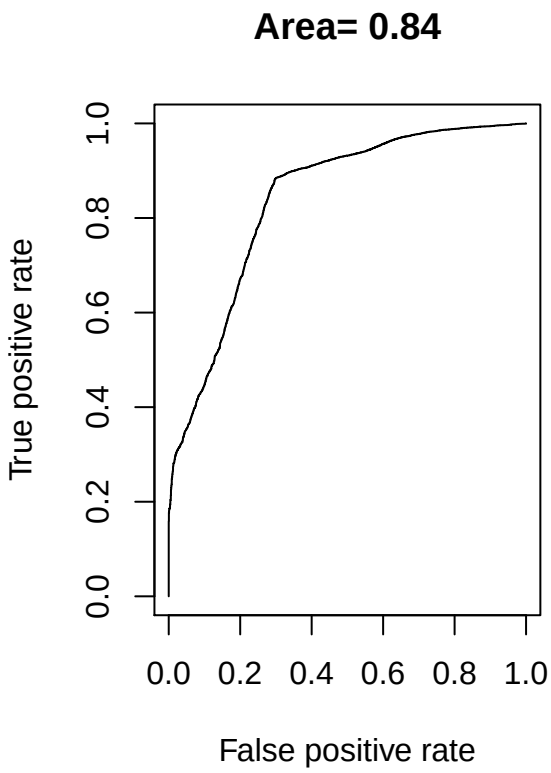
Modificamos ahora el tipo de kernel

```r
svmfit.opt3 = svm(Arrest ~ ., data =  muestra_modelo, kernel = "radial"
                  ,gamma = 1, cost = 1, decision.values = T,
                  type = "C-classification")
fitted3 = attributes(predict(svmfit.opt3, muestra_modelo,
                             decision.values = TRUE))$decision
summary(svmfit.opt3)
```

```
##
## Call:
```

```
## svm(formula = Arrest ~ ., data = muestra_modelo, kernel = "radial",
##     gamma = 1, cost = 1, decision.values = T, type = "C-classification")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  1
##
## Number of Support Vectors:  6268
##
##  ( 4029 2239 )
##
##
## Number of Classes:  2
##
## Levels:
##  2 1
```

```
par(mfrow = c(1, 2))
rocplot(fitted3, muestra_modelo[,"Arrest"])
```

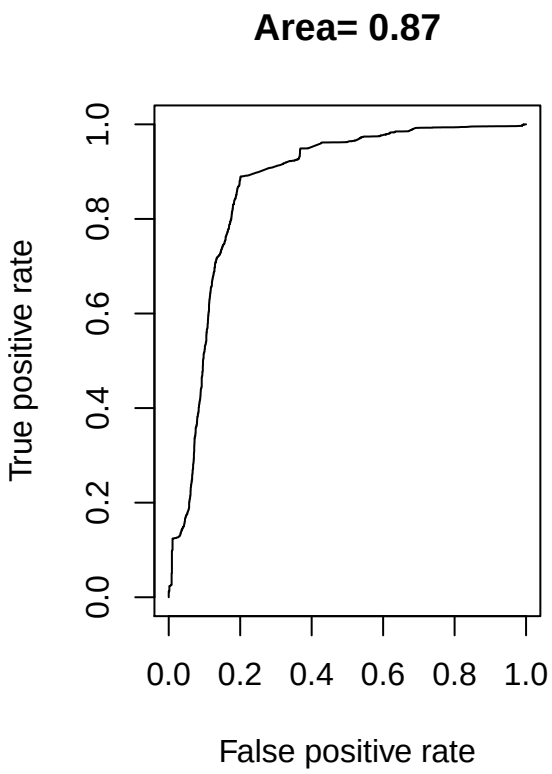


Al igual que en el caso anterior, cambiamos los valores de gamma y cost

```
svmfit.opt4 = svm(Arrest ~ ., data =  muestra_modelo, kernel = "radial",
                  gamma = 10, cost = 10, decision.values = T,
                  type = "C-classification")
fitted4 = attributes(predict(svmfit.opt4, muestra_modelo,
                             decision.values = TRUE))$decision
summary(svmfit.opt4)
```

```
##
## Call:
## svm(formula = Arrest ~ ., data = muestra_modelo, kernel = "radial",
##     gamma = 10, cost = 10, decision.values = T, type = "C-classification")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  10
##       gamma:  10
##
## Number of Support Vectors:  7780
##
##  ( 5579 2201 )
##
##
## Number of Classes:  2
##
## Levels:
##  2 1
```

```
par(mfrow = c(1, 2))
rocplot(fitted4, muestra_modelo[,"Arrest"])
```



Vemos que, en ambos casos, funciona mejor gamma = 10 y cost = 10, que cuando su valor es 1.

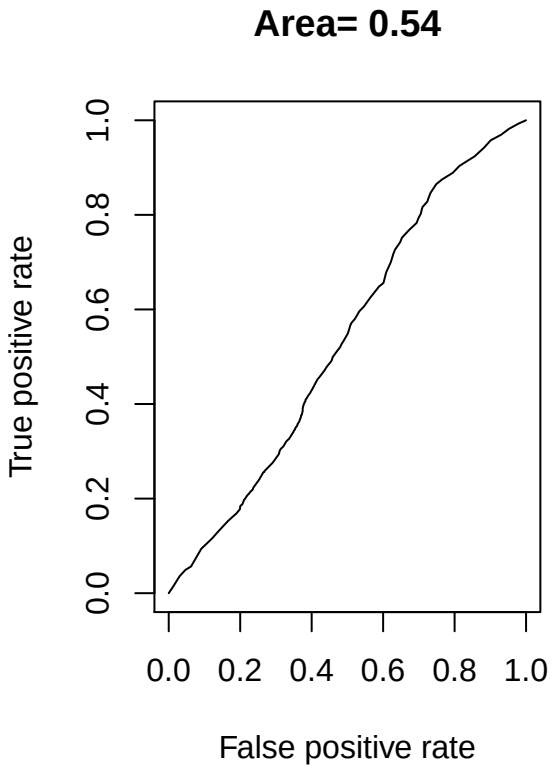
Por otro lado, a la hora de hacer este modelo, estamos considerando todas las variables. Realizamos ahora el modelo con las variables elegidas en el modelo de regresión logística que mejor funcionaba.

```
svmfit.opt5 = svm(Arrest ~ Domestic + District + Year ++ Time_slot + Trimestre,
                  data =  muestra_modelo, kernel = "radial",gamma = 10,
                  cost = 10, decision.values = T,type = "C-classification")
fitted5 = attributes(predict(svmfit.opt5, muestra_modelo,
                             decision.values = TRUE))$decision
summary(svmfit.opt5)
```

```
##
## Call:
## svm(formula = Arrest ~ Domestic + District + Year + +Time_slot +
##     Trimestre, data = muestra_modelo, kernel = "radial", gamma = 10,
##     cost = 10, decision.values = T, type = "C-classification")
##
##
## Parameters:
##    SVM-Type:  C-classification
```

```
##  SVM-Kernel:  radial
##        cost:  10
##       gamma:  10
##
## Number of Support Vectors:  4487
##
##  ( 2244 2243 )
##
##
## Number of Classes:  2
##
## Levels:
##  2 1
```

```r
par(mfrow = c(1, 2))
rocplot(fitted5, muestra_modelo[,"Arrest"])
```



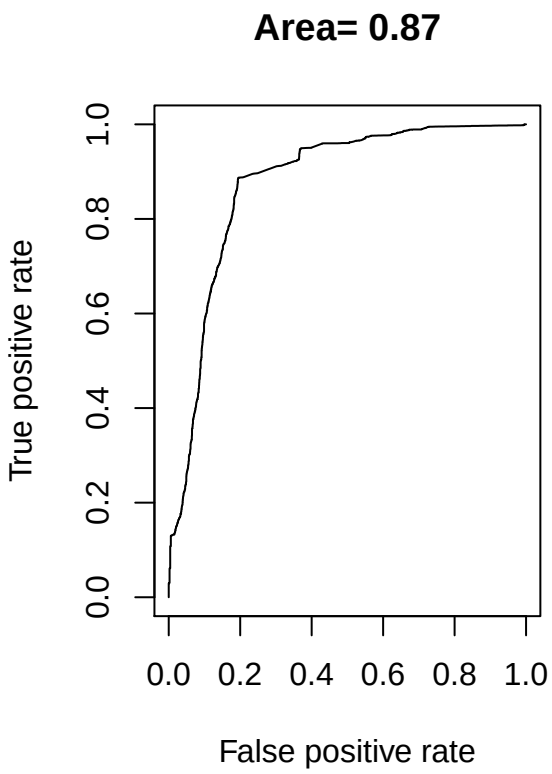
Vemos que con éste último no mejoramos los resultados obtenidos con el anterior. Lo probamos, por tanto, con una muestra del conjunto de test.

```r
set.seed(1)
muestra_modelo_test <- Chicago_test[sample(1:nrow(Chicago_test),10000,
                                           replace = FALSE),]
```

```r
svmfit.opt_test = svm(Arrest ~ ., data =  muestra_modelo_test,
                      kernel = "radial",gamma = 10, cost = 10,
                      decision.values = T,type = "C-classification")
fitted_test = attributes(predict(svmfit.opt_test, muestra_modelo_test,
                                 decision.values = TRUE))$decision
summary(svmfit.opt_test)
```

```
##
## Call:
## svm(formula = Arrest ~ ., data = muestra_modelo_test, kernel = "radial",
##     gamma = 10, cost = 10, decision.values = T, type = "C-classification")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  10
##       gamma:  10
##
## Number of Support Vectors:  7717
##
##  ( 5546 2171 )
##
##
## Number of Classes:  2
##
## Levels:
##  2 1
```

```r
par(mfrow = c(1, 2))
rocplot(fitted_test, muestra_modelo_test[,"Arrest"])
```



CONCLUSIONES

A la vista de los resultados obtenidos al aplicar distintos modelos de predicción a nuestros datos, podemos que el modelo que mejor se adapta son los SVM. No es posible aplicar otros modelos, como random forest, y otros nos dan unos resultados poco favorables para nuestra predicción del arresto.