Data Mining en Ciencia y Técnica - TP1

Ariel Aguirre, Miguel Barros, José Badillo, Diego Dell'Era

TP1

Cargamos el dataset:

```
glx <- read.csv("COMBO17.csv", header = T, stringsAsFactors = F)
```

Tarea 1

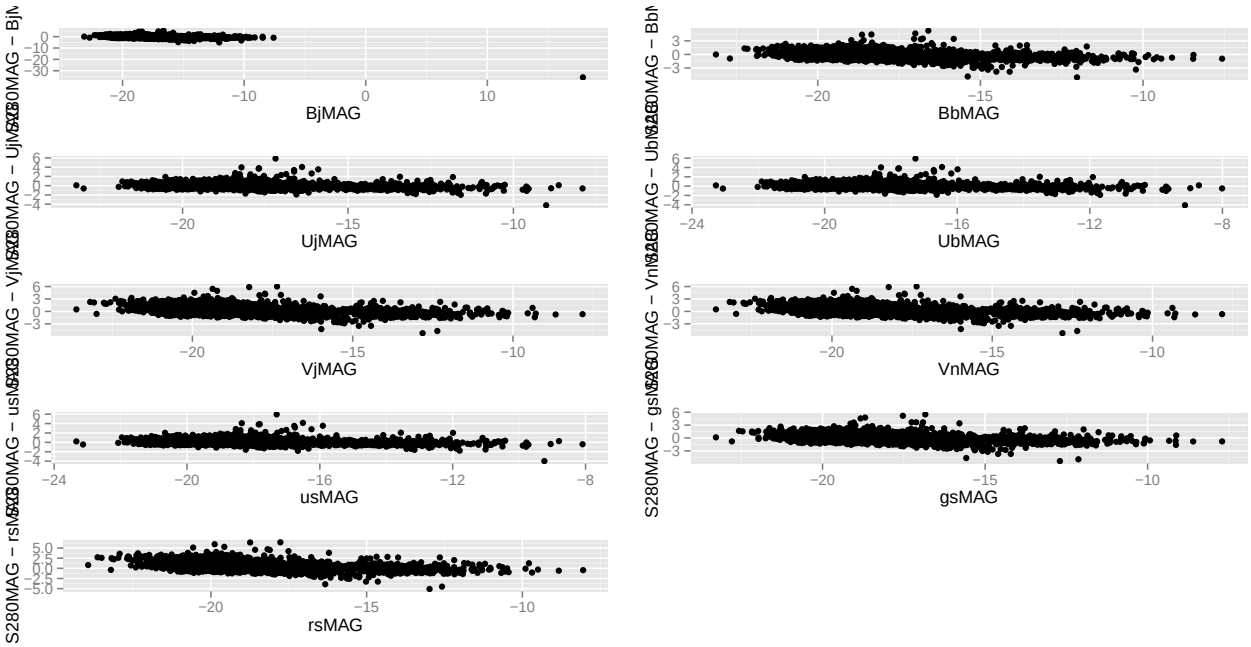
str(glx) -> problema: la variable e.W420FE es de tipo 'chr'. La convertimos a numérica:

```
glx$e.W420FE <- as.numeric(glx$e.W420FE)
```

Tarea 2

```
library(ggplot2)
library(gridExtra)

p1 <- qplot(BjMAG, S280MAG-BjMAG, data = glx)
p2 <- qplot(BbMAG, S280MAG-BbMAG, data = glx)
p3 <- qplot(UjMAG, S280MAG-UjMAG, data = glx)
p4 <- qplot(UbMAG, S280MAG-UbMAG, data = glx)
p5 <- qplot(VjMAG, S280MAG-VjMAG, data = glx)
p6 <- qplot(VnMAG, S280MAG-VnMAG, data = glx)
p7 <- qplot(usMAG, S280MAG-usMAG, data = glx)
p8 <- qplot(gsMAG, S280MAG-gsMAG, data = glx)
p9 <- qplot(rsMAG, S280MAG-rsMAG, data = glx)
grid.arrange(p1, p2, p3, p4, p5, p6, p7, p8, p9, ncol=2, nrow=5)
```

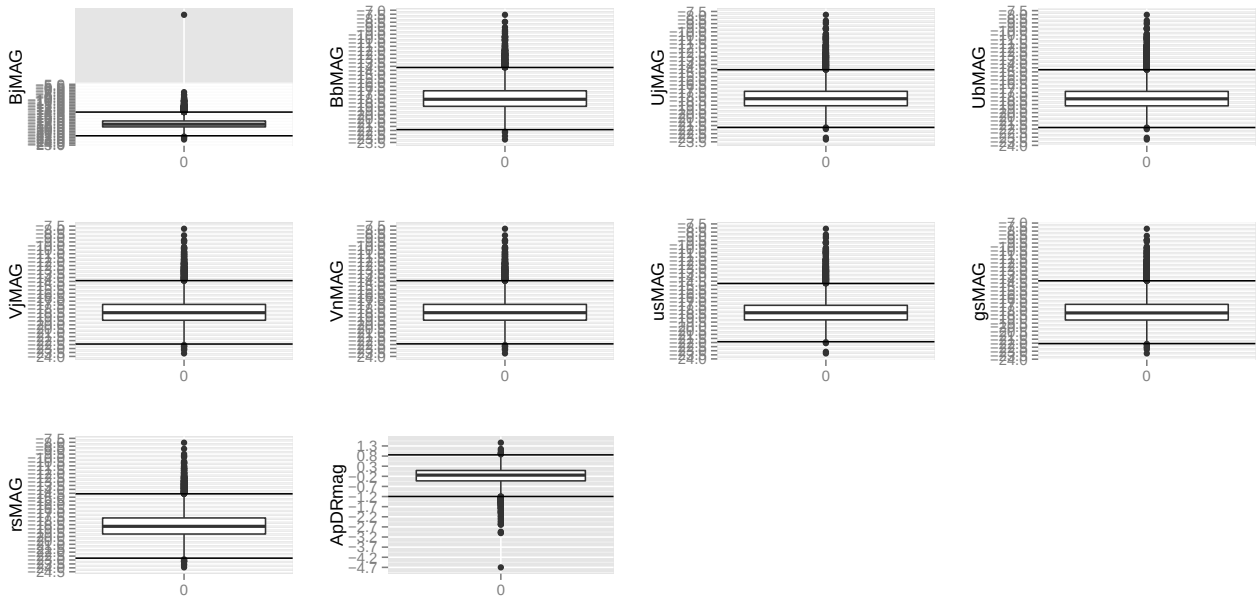


Para separar outliers, podemos empezar por mirar el criterio de los bigotes (i.e. definir un par de barras, arriba y abajo, a 1.5 * distancia intercuartil desde la caja), y quitar los que excedan esos límites:

```
limite_inferior_boxplot <- function(magnitud) { q <- quantile(magnitud, na.rm=TRUE); return (q[2] - (q[4
limite_superior_boxplot <- function(magnitud) { q <- quantile(magnitud, na.rm=TRUE); return (q[4] + (q[4

nros_ejes <- scale_y_continuous(breaks = round(seq(-30, -5, by = 0.5), 1))

b1 <-  qplot(factor(0), BjMAG,   geom = "boxplot", xlab="", data=glx) + geom_hline(yintercept=limite_ini
b2 <-  qplot(factor(0), BbMAG,   geom = "boxplot", xlab="", data=glx) + geom_hline(yintercept=limite_ini
b3 <-  qplot(factor(0), UjMAG,   geom = "boxplot", xlab="", data=glx) + geom_hline(yintercept=limite_ini
b4 <-  qplot(factor(0), UbMAG,   geom = "boxplot", xlab="", data=glx) + geom_hline(yintercept=limite_ini
b5 <-  qplot(factor(0), VjMAG,   geom = "boxplot", xlab="", data=glx) + geom_hline(yintercept=limite_ini
b6 <-  qplot(factor(0), VnMAG,   geom = "boxplot", xlab="", data=glx) + geom_hline(yintercept=limite_ini
b7 <-  qplot(factor(0), usMAG,   geom = "boxplot", xlab="", data=glx) + geom_hline(yintercept=limite_ini
b8 <-  qplot(factor(0), gsMAG,   geom = "boxplot", xlab="", data=glx) + geom_hline(yintercept=limite_ini
b9 <-  qplot(factor(0), rsMAG,   geom = "boxplot", xlab="", data=glx) + geom_hline(yintercept=limite_ini
b10 <- qplot(factor(0), ApDRmag, geom = "boxplot", xlab="", data=glx) + geom_hline(yintercept=limite_ini
grid.arrange(b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, ncol=4, nrow=3)
```



Pero quitaríamos demasiados puntos con ese criterio... Mejor quitamos sólo los que son claramente outliers, en las variables *ApRDmag* y de *BjMAG*:

```
# antes de quitar outliers
dim(glx)
```

```
## [1] 3462   65
```

```
glx <- subset(glx, ApDRmag > -3.2)
glx <- subset(glx, BjMAG < -7.0)

# después
dim(glx)
```

```
## [1] 3460   65
```

Tarea 3

Miramos si alguna variable (i.e., columna) tiene valores faltante:

```
apply(glx, 2, function(x) anyNA(x))
```

```
##        Nr      Rmag    e.Rmag   ApDRmag     mumax       Mcz     e.Mcz     MCzml
##     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE
##   chi2red     UjMAG  e.UjMAG     BjMAG  e.BjMAG     VjMAG  e.VjMAG     usMAG
##     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE
##   e.usMAG     gsMAG  e.gsMAG     rsMAG  e.rsMAG     UbMAG  e.UbMAG     BbMAG
##     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE
##   e.BbMAG     VnMAG  e.VbMAG  S280MAG e.S280MA   W420FE e.W420FE   W462FE
##     FALSE      TRUE      TRUE      TRUE      TRUE     FALSE      TRUE     FALSE
## e.W462FE   W485FD e.W485FD   W518FE e.W518FE   W571FS e.W571FS   W604FE
```

```
##    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
## e.W604FE   W646FD e.W646FD   W696FE e.W696FE   W753FE e.W753FE   W815FS
##    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
## e.W815FS   W856FD e.W856FD   W914FD e.W914FD   W914FE e.W914FE      UFS
##    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
##    e.UFS      BFS    e.BFS      VFD    e.VFD      RFS    e.RFS      IFD
##    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
##    e.IFD
##    FALSE
```

De las variables de interés, hay 2 con datos faltantes: *VnMAG*, *S280MAG*

```
faltantes_VnMAG <- which(is.na(glx$VnMAG))
faltantes_S280MAG <- which(is.na(glx$S280MAG))
faltantes_VnMAG
```

```
## [1] 3444
```

```
faltantes_S280MAG
```

```
##  [1]   22   40   89  159  363  385  415  492  576  969 1023 1426 1455 1529
## [15] 1530 1556 2264 2510 2815 2885 2889 2935 3422 3444
```

También hay valores faltantes en las variables asociadas de error, en los mismos registros:

```
faltantes_e.VbMAG <- which(is.na(glx$e.VbMAG))
faltantes_e.280MA <- which(is.na(glx$e.S280MA))
faltantes_e.VbMAG
```

```
## [1] 3444
```

```
faltantes_e.280MA
```

```
##  [1]   22   40   89  159  363  385  415  492  576  969 1023 1426 1455 1529
## [15] 1530 1556 2264 2510 2815 2885 2889 2935 3422 3444
```

Son 24 registros en total. Los borramos:

```
glx_sin_faltantes <- glx[complete.cases(glx[,26:29]),]
dim(glx)[1] - 24 == dim(glx_sin_faltantes)[1]
```

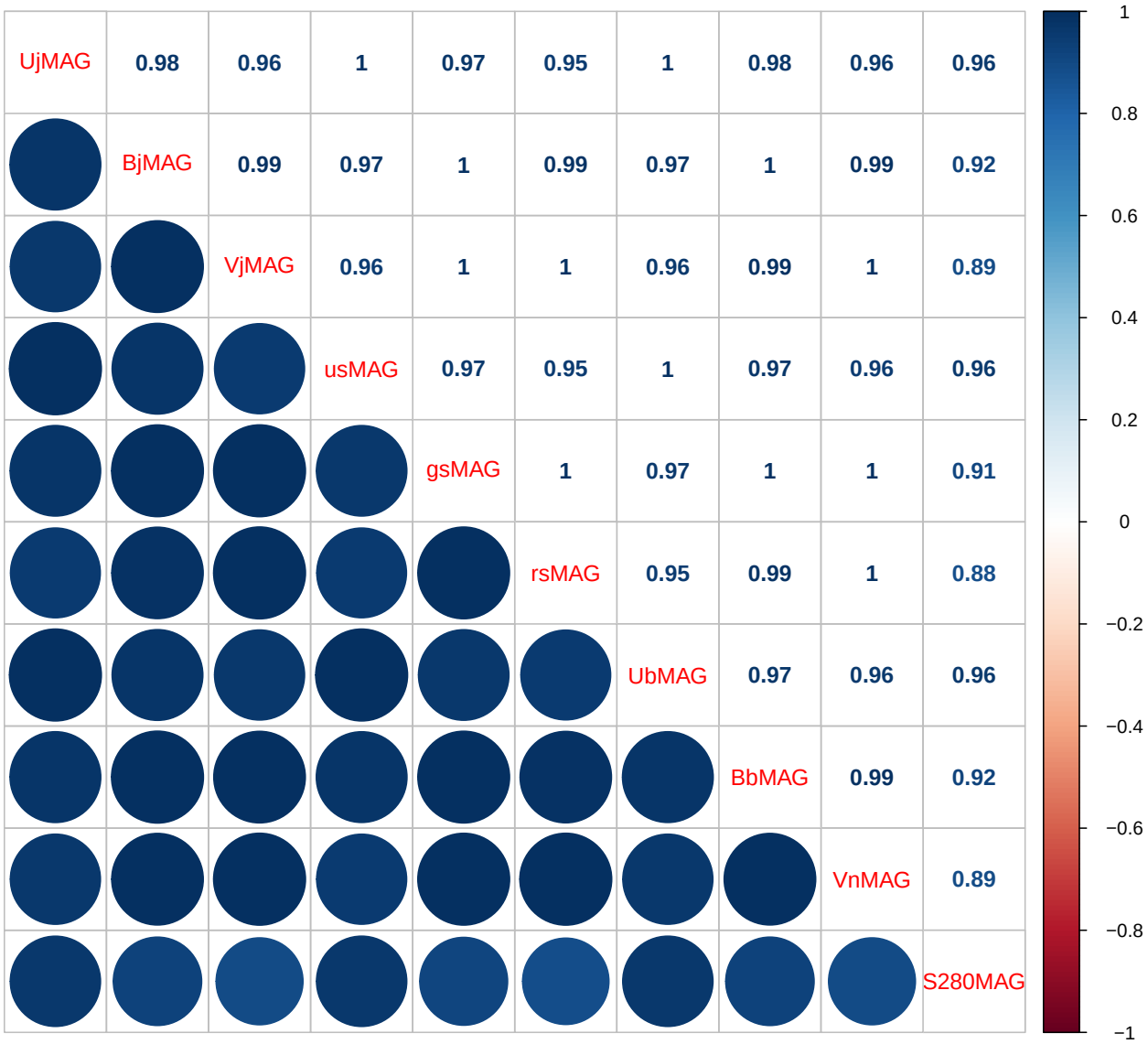
```
## [1] TRUE
```

Tarea 4

```
espectrales <- c(10,12,14,16,18,20,22,24,26,28)
variables_de_magnitud_absoluta_en_reposo <- glx_sin_faltantes[, espectrales]
head(variables_de_magnitud_absoluta_en_reposo)
```

```
##    UjMAG  BjMAG  VjMAG  usMAG  gsMAG  rsMAG  UbMAG  BbMAG  VnMAG S280MAG
## 1 -17.67 -17.54 -17.76 -17.83 -17.60 -17.97 -17.76 -17.53 -17.76  -18.22
## 3 -19.75 -19.91 -20.41 -19.87 -20.05 -20.71 -19.82 -19.89 -20.40  -19.77
## 4 -17.83 -17.39 -17.67 -17.98 -17.47 -17.89 -17.92 -17.38 -17.67  -18.12
## 5 -17.69 -18.40 -19.37 -17.81 -18.69 -19.88 -17.76 -18.35 -19.37  -13.93
## 6 -19.22 -18.11 -18.70 -19.34 -18.27 -19.05 -19.30 -18.08 -18.69  -19.18
## 7 -17.09 -16.06 -16.23 -17.26 -16.11 -16.39 -17.19 -16.05 -16.22  -17.81
```

```
library(corrplot)
correlaciones <- cor(variables_de_magnitud_absoluta_en_reposo)
# corrplot(correlaciones, method="circle", type="lower")
corrplot.mixed(correlaciones, lower="circle", upper="number")
```



A cada magnitud le restamos la magnitud a 280 nm:

```
variables_de_magnitud_absoluta_en_reposo_normalizadas <- sweep(variables_de_magnitud_absoluta_en_reposo
head(variables_de_magnitud_absoluta_en_reposo_normalizadas)
```

```
##   UjMAG BjMAG VjMAG usMAG gsMAG rsMAG UbMAG BbMAG VnMAG
## 1  0.55  0.68  0.46  0.39  0.62  0.25  0.46  0.69  0.46
## 3  0.02 -0.14 -0.64 -0.10 -0.28 -0.94 -0.05 -0.12 -0.63
## 4  0.29  0.73  0.45  0.14  0.65  0.23  0.20  0.74  0.45
## 5 -3.76 -4.47 -5.44 -3.88 -4.76 -5.95 -3.83 -4.42 -5.44
## 6 -0.04  1.07  0.48 -0.16  0.91  0.13 -0.12  1.10  0.49
## 7  0.72  1.75  1.58  0.55  1.70  1.42  0.62  1.76  1.59
```

```r
correlaciones_de_normalizadas <- cor(variables_de_magnitud_absoluta_en_reposo_normalizadas)
# corrplot(correlaciones_de_normalizadas, method="circle", type="lower")
corrplot.mixed(correlaciones_de_normalizadas, lower="circle", upper="number")
```