Anomalias_D1

Miguel Merelo Hernández

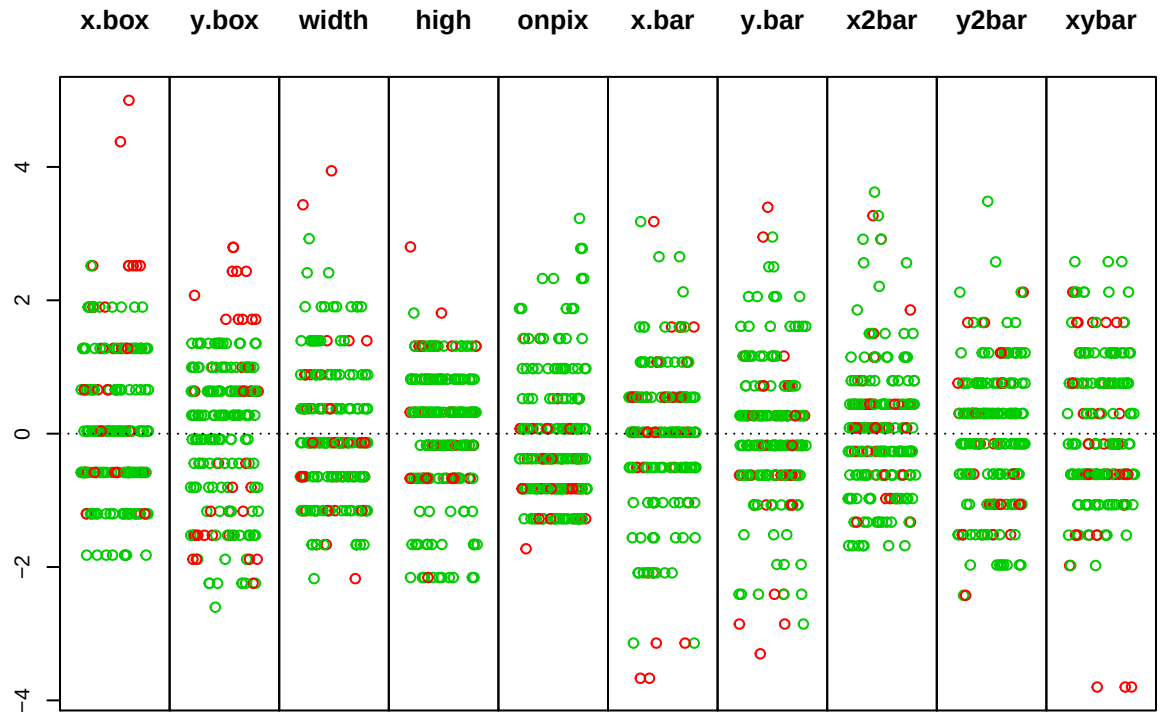
February 11, 2018

Crear datos

Utilizamos LetterRecognition del paquete mlbench. Solo utilizamos las primeras 200 líneas ya que la visualización de 20000 entradas hace imposible analizar el problema.

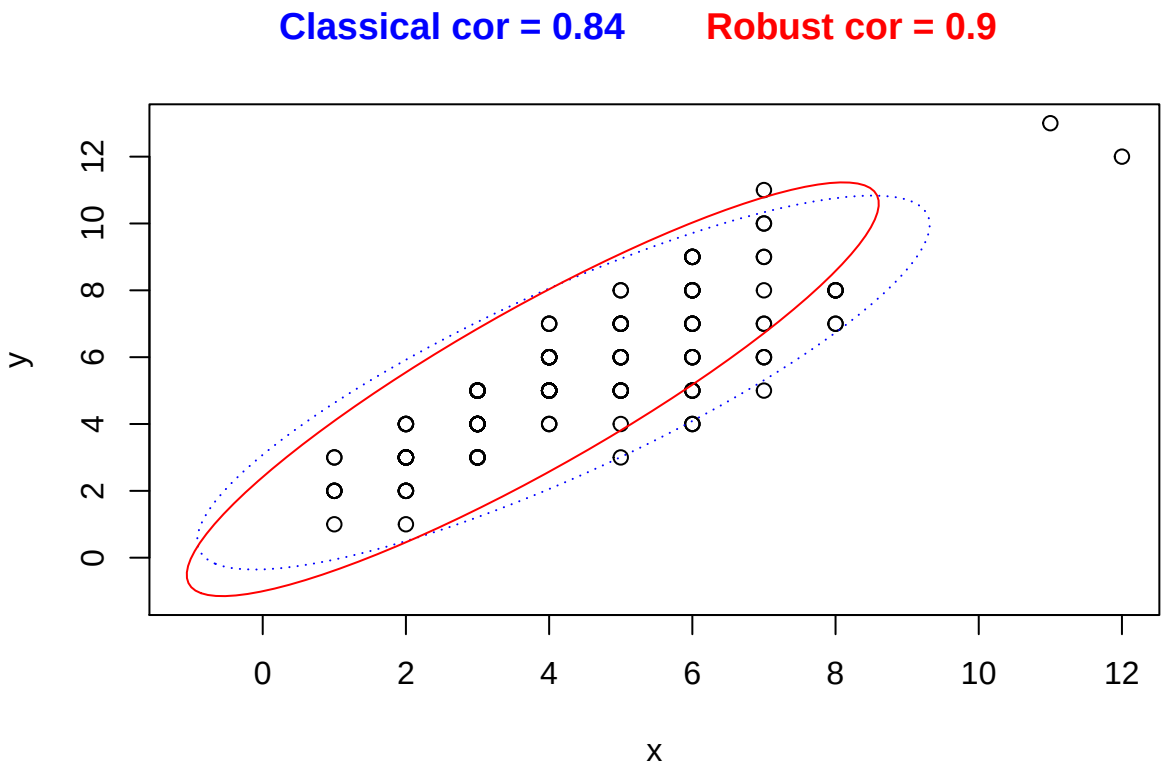
```
library(mlbench)
data("LetterRecognition")
mis.datos.numericos<-LetterRecognition[1:200,-c(1)]
mis.datos.numericos.normalizados<-scale(mis.datos.numericos)
row.names(mis.datos.numericos.normalizados)<-row.names(mis.datos.numericos)
```

```
alpha.value = 0.05
alpha.value.penalizado = 1 - ( 1 - alpha.value) ^ (1 / nrow(mis.datos.numericos))
set.seed(12)
mvoutlier.plot<-uni.plot(mis.datos.numericos[1:10],
                         symb=FALSE,
                         alpha=alpha.value.penalizado)
```



Tenemos el mismo gráfico ya analizado con los outliers multivariables marcados en rojo.

```
is.MCD.outlier<-mvoutlier.plot$outliers
numero.de.outliers.MCD<-sum(is.MCD.outlier)
corr.plot(mis.datos.numericos[,1], mis.datos.numericos[,3])
```

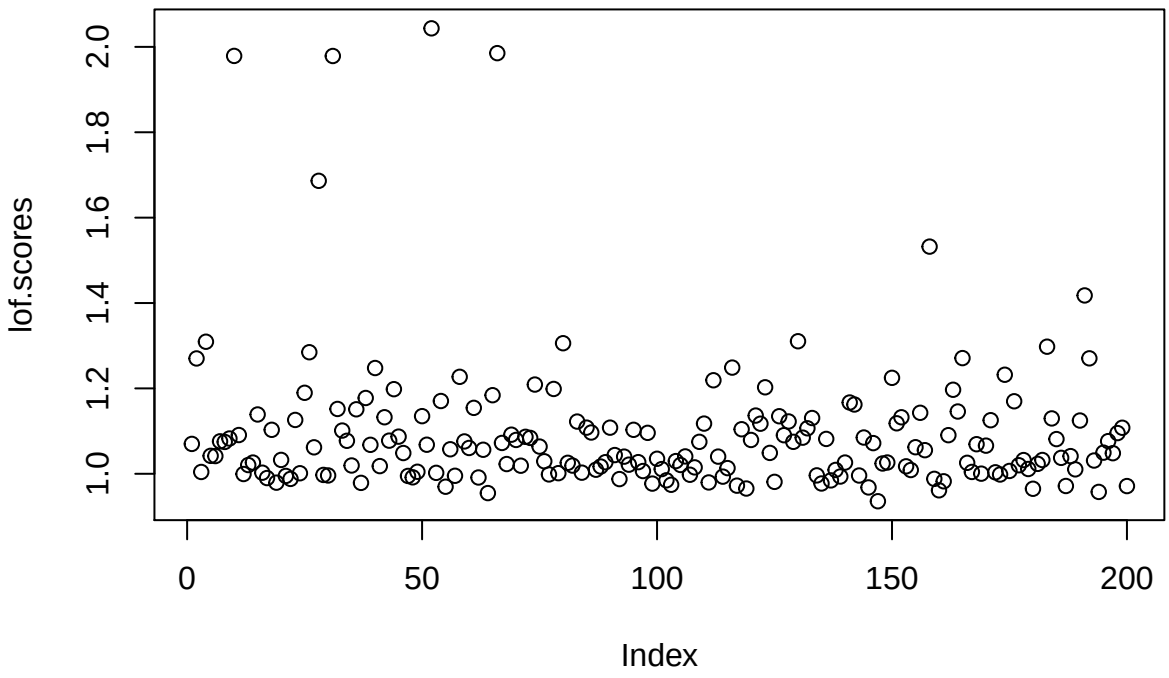



```
## $cor.cla
## [1] 0.8436275
##
## $cor.rob
## [1] 0.8972285
```

Entre estas dos variables podemos ver como hay dos outliers bastante claros con valores 11 y 12 en X y 13 y 12 en Y.

1. DISTANCE BASED OUTLIERS (LOF)

```
numero.de.vecinos.lof = 5
set.seed(12)
lof.scores<-lofactor(mis.datos.numericos.normalizados,numero.de.vecinos.lof)
plot(lof.scores)
```

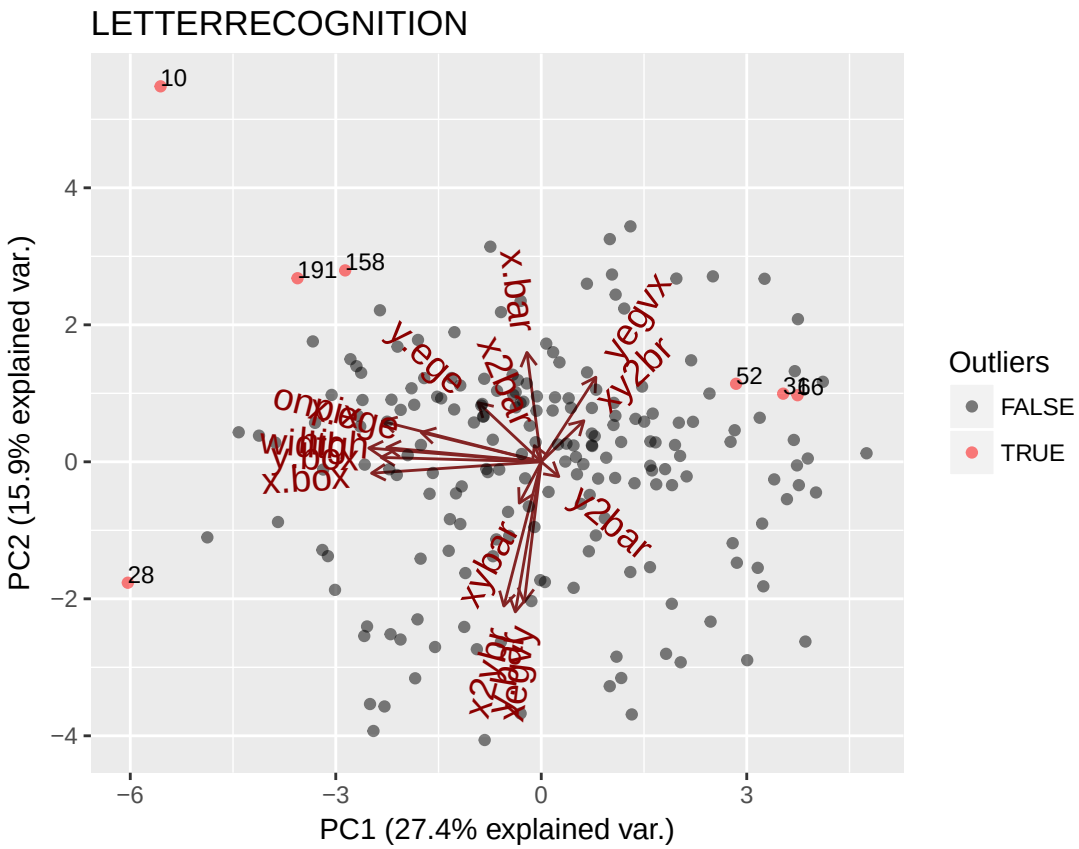


Vemos que hay 5 puntos con un lof claramente más alto que los demás, por encima de 1.6 y otros 2 que destacan entre 1.4 y 1.6 por lo que fijaremos el número de outliers en 7.

```
numero.de.outliers = 7
indices.de.lof.outliers.ordenados<-order(lof.scores,decreasing=TRUE)
indices.de.lof.top.outliers<-indices.de.lof.outliers.ordenados[1:numero.de.outliers]
is.lof.outlier<-row.names(mis.datos.numericos) %in% indices.de.lof.top.outliers
indices.de.lof.top.outliers
```

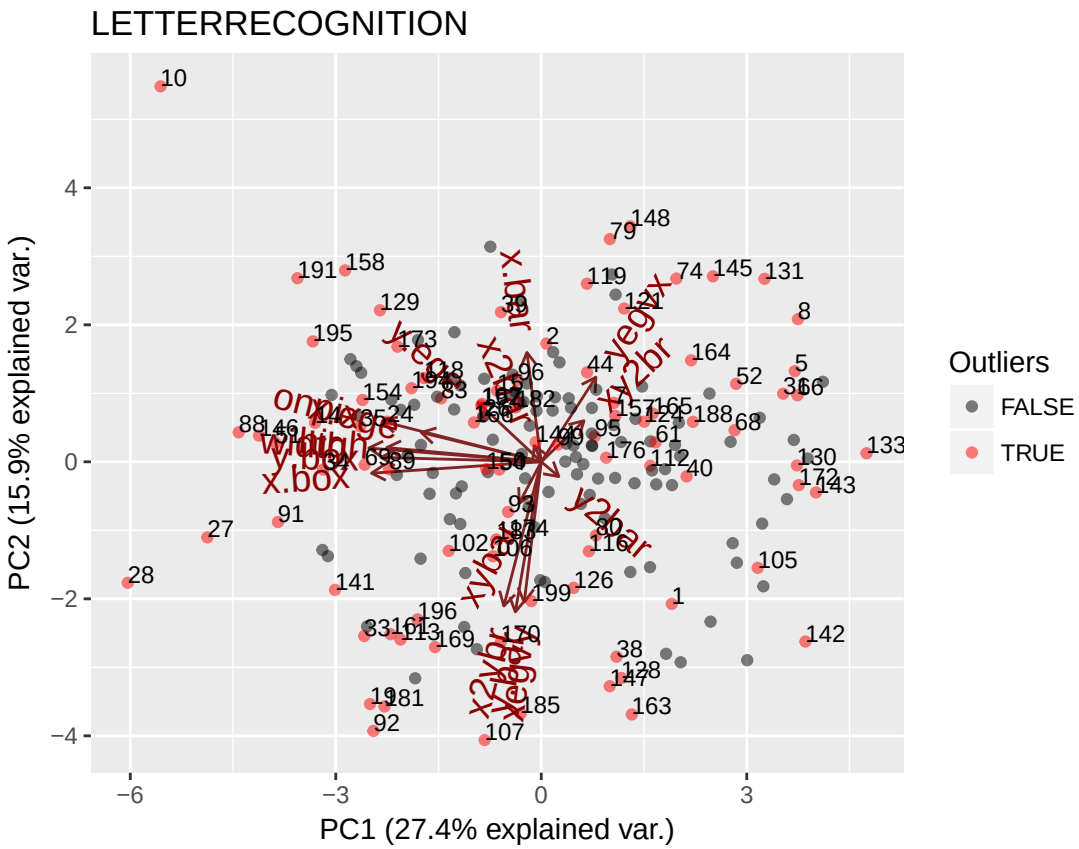
```
## [1]  52  66  10  31  28 158 191
```

```
MiBiPlot_Multivariate_Outliers(mis.datos.numericos,
                               is.lof.outlier,
                               "LETTERRECOGNITION")
```



Los 5 puntos con mayor lof son 10, 28, 52, 31 y 66 mientras que los dos que consideramos outliers por superar el valor 1.4 de lof son 191 y 158.

```
vector.claves.outliers.IQR.en.alguna.columna<-
  vector_claves_outliers_IQR_en_alguna_columna(mis.datos.numericos)
vector.es.outlier.IQR.en.alguna.columna<-vector_es_outlier_IQR_en_alguna_columna(mis.datos.numericos)
MiBiPlot_Multivariate_Outliers(mis.datos.numericos,vector.es.outlier.IQR.en.alguna.columna,"LETTERRECOG
```



Vemos que los puntos con mayor lof que habíamos seleccionado son outliers por columna.

```
indices.de.outliers.multivariantes.LOF.pero.no.1variantes<-setdiff(vector.claves.outliers.IQR.en.alguna
sort(indices.de.outliers.multivariantes.LOF.pero.no.1variantes)
```

```
##  [1]   1   2   5   7   8  14  15  19  24  26  27  33  34  35  38  39  40
## [18]  44  51  54  61  68  69  74  79  80  83  88  89  91  92  93  95  96
## [35]  99 102 105 106 107 112 113 116 118 119 121 124 126 128 129 130 131
## [52] 133 141 142 143 144 145 146 147 148 150 154 157 161 163 164 165 166
## [69] 167 169 170 172 173 174 176 181 182 183 185 188 192 194 195 196 199
```

Confirmamos la conclusión del gráfico anterior por lo que podemos decir que, para nuestro caso, usando distancia de Mahalanobis o usando LOF llegamos al mismo resultado.

```
data.frame.numeric<-LetterRecognition[sapply(LetterRecognition,is.numeric)]
head(data.frame.numeric)
```

```
##   x.box y.box width high onpix x.bar y.bar x2bar y2bar xybar x2ybr xy2br
## 1     2     8     3    5     1     8    13     0     6     6    10     8
## 2     5    12     3    7     2    10     5     5     4    13     3     9
## 3     4    11     6    8     6    10     6     2     6    10     3     7
## 4     7    11     6    6     3     5     9     4     6     4     4    10
## 5     2     1     3    1     1     8     6     6     6     6     5     9
## 6     4    11     5    8     3     8     8     6     9     5     6     6
##   x.ege xegvy y.ege yegvx
## 1     0     8     0     8
## 2     2     8     4    10
## 3     3     7     3     9
## 4     6    10     2     8
```

```
## 5     1     7     5    10
## 6     0     8     9     7
```