Detección de Anomalías

1. Análisis de la base de datos

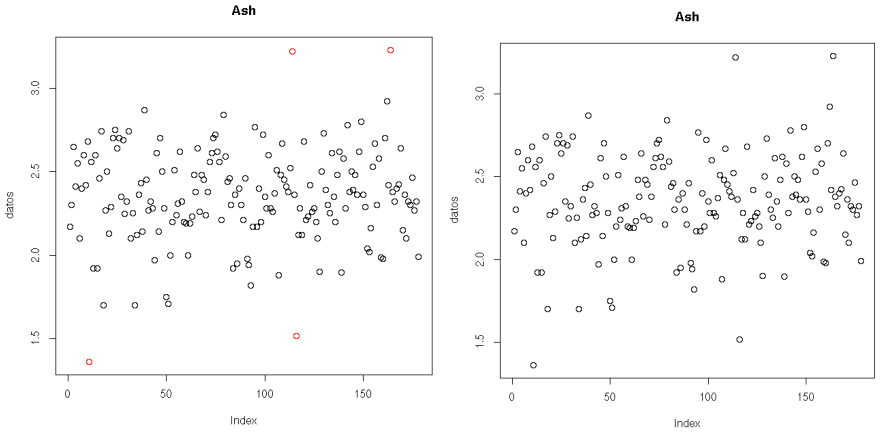
Para este trabajo sobre detección de anomalías se ha elegido la base de datos Wine, sacada de la web KEEL: http://sci2s.ugr.es/keel/dataset_smja.php?cod=594
Esta base de datos está compuesta de 13 variables de las cuales las 13 son de tipo real. Tiene un total de 178 instancias, y las instancias se clasifican en 3 clases. La base de datos no tiene valores perdidos, y se ha elegido esta base de datos ya que en la asignatura de Aprendizaje Automático del Grado de Ingeniería Informática tuve contacto con ella.
Los atributos son: Alcohol, Malic_acid, Ash, Alcalinity_of_ash, Magnesium, Total_phenols, Flavanoids, Nonflavanoid_phenols, Proanthocyanins, Color_intensity, Hue, OD280/OD315, Proline y Class.
Como bien se indica en esta web, la descripción del dataset es:
"These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.
In a classification context, this is a well posed problem with well behaved class structures. A good data set for first testing of a new classifier, but not very challenging."
Por lo tanto, vamos a pasar a la detección de las anomalías en nuestro dataset.

2. Univariate Statistical Outliers IQR

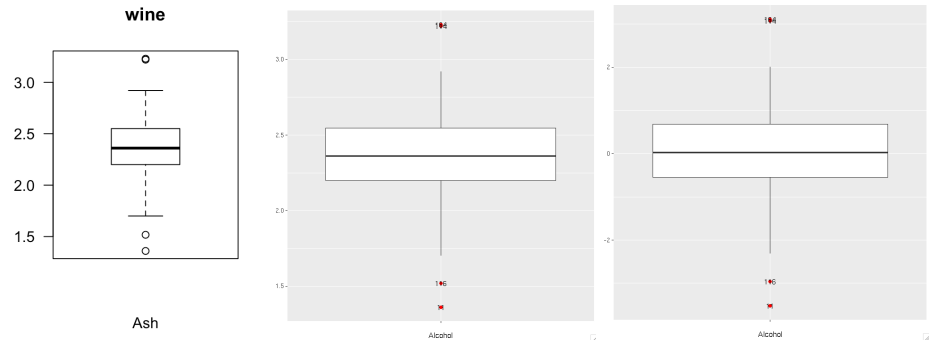
Siguiendo el modelo visto en prácticas, se ha realizado el Script en "B1Variate_IQR.R", para analizar los outliers del dataset.
Lo primero realizado ha sido dejar la variable mydata.numeric como todo el dataset sin la variable clasificadora de clase. La columna elegida ha sido la columna Ash, la 3.
Lo primero realizado ha sido el cómputo de los outliers IQR, para lo que se ha seguido la regla IQR, sin funciones propias. De esta forma, podemos obtener cuales son las claves de los outliers normales, que en nuestro caso han sido: 11, 114, 116 y 164. Aquí los podemos examinar:

```
> data.frame.outliers.normales
    Alcohol MalicAcid      Ash AlcalinityOfAsh Magnesium TotalPhenols flavanoids NonflavanoidsPhenols Proanthocyanins ColorIntensity  Hue OD280/OD315 Proline
11    12.37      0.94 1.360000           10.6        88         1.98       0.57             0.2800000           0.42        1.95000 1.05        1.82     520
114   13.05      2.05 3.220000           25.0       124         2.63       2.68             0.6459548           1.92       12.53849 1.13        3.20     830
116   13.76      1.53 1.517288           19.5       132         2.95       2.74             0.5000000           1.35        5.40000 1.25        3.00    1235
164   11.56      2.05 3.230000           28.5       119         3.18       5.08             0.4700000           1.87        6.00000 0.93        3.69     465
```

Si buscamos los outliers extremos vemos que para esta columna no encontramos ninguno. En las siguientes imágenes podemos ver representados los outliers, siendo la imagen izquierda los normales y la derecha los extremos.

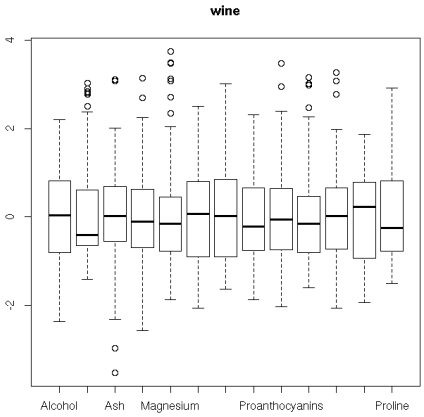


Si queremos ver el BoxPlot, debemos llamarlo con la función ggplot, ya que, como se ve en la imagen de la izquierda, no se muestran los outliers. Sin embargo, en las otras dos imágenes, si podemos apreciar en color rojo los outliers. La diferencia entre la imagen del centro y la de la parte derecha es, que en la de la izquierda se han normalizado los valores, viendo así que la normalización no afecta a la representación.

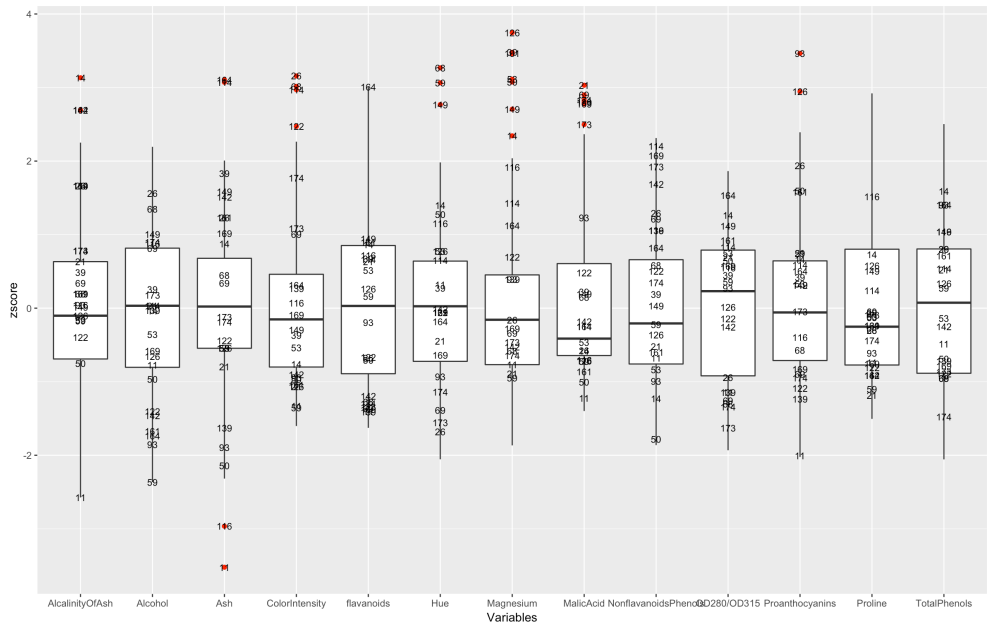


Seguidamente se han usado funciones propias para sacar directamente los outliers, como son las funciones: vector_es_outlier_IQR y vector_claves_outliers_IQR, que nos devuelven los mismos outliers que se han indicado anteriormente.

Como ya podemos obtener los outliers de una columna, vamos a buscar en todas las columnas, con las funciones, cuantos outliers podemos obtener. De esta forma vemos que en la columna 2 tenemos 6 outliers, en la 3 tenemos 4, en la 4 tenemos 3, en la 5 tenemos 7, en la 10 tenemos 4 y en la 11 tenemos 3. Los cuales los podemos representar con un boxplot normal:



O con una nueva función propia en la que le añadimos al boxplot las etiquetas de los outliers:

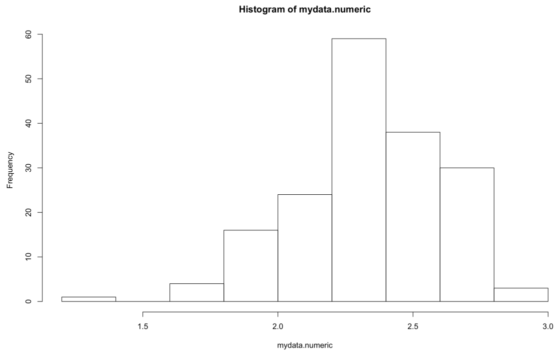


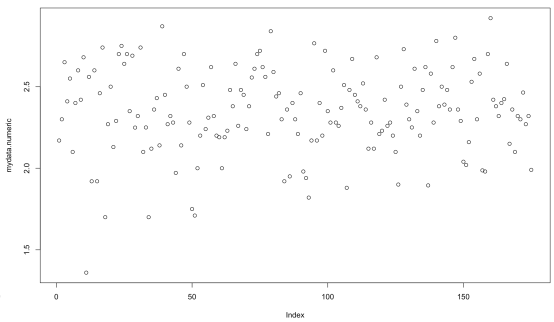
De esta forma tenemos representadas las variables y sus outliers.

3. Test Estadísticos

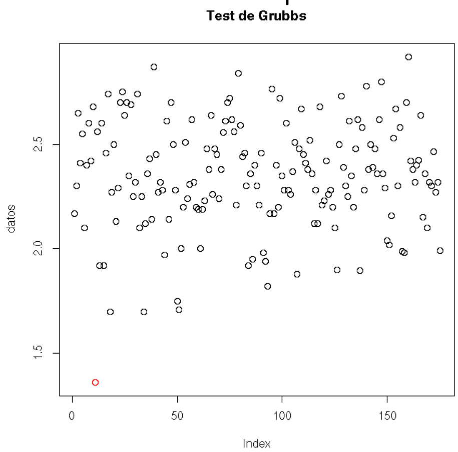
En este apartado, vamos a realizar el Script "B2Variate_TestsEstadisticos.R" en el que vamos a probar sobre nuestro conjunto de datos los tests de Grubbs, para un único outlier, y el test de Rosner para k outliers.

Comenzamos viendo, como nos quedamos en el Script anterior, una lista con los outliers de cada columna. Para la primera parte, es decir, el test de Grubbs, vamos a seleccionar la columna 3, que tiene 4 outliers, pero para probar este test, he decidido eliminar 3 filas, de los 4 outliers, para poder hacer un test correcto. Por lo que nos queda un conjunto de datos con estas representaciones:

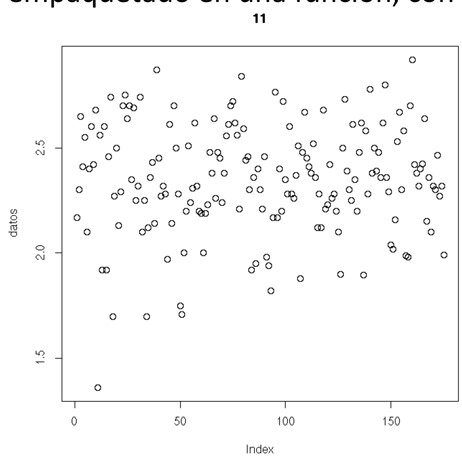




Si aplicamos el test de Grubbs, obtenemos un p.value de 0.019, por lo que será significativo, indicando que hay un outlier. Tendremos por lo tanto que calcular el outlier de forma manual. Una vez calculado tenemos que el outlier está en la fila 11 de esa columna, con un valor de 1.36. Si hacemos la representación nueva, obtenemos:

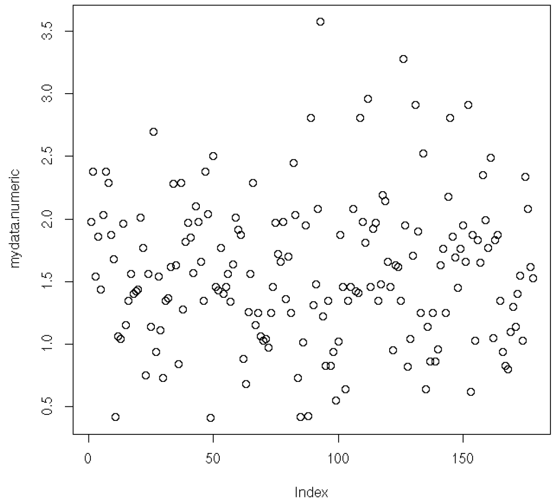


Donde se ve el outlier en la parte inferior izquierda. Además, todo este proceso se ha empaquetado en una función, con lo que obtenemos los resultados instantáneos:




```
> MiPlot_resultados_TestGrubbs(mydata.numeric)
p.value: 0.01962252
Índice de outlier: 11
Valor del outlier: 1.36
Numero de datos: 175
?Qui?n es outlier?: FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE F
ALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALS
E FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE F
ALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALS
E FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE F
ALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALS
E FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE F
ALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALS
E FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE F
ALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALS
E FALSE FALSE FALSE TRUE
```

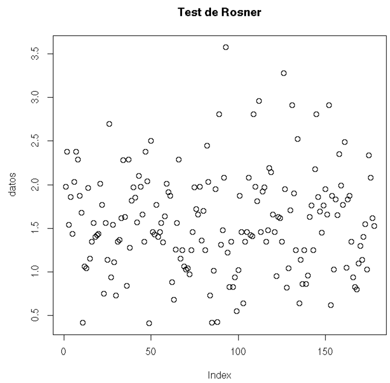
Vamos a pasar ahora a realizar el análisis con otra columna, la columna 9, donde tenemos dos outliers, y el test de Grubbs debería fallar al no ser un único outlier. El conjunto de datos sería:



Por lo que aplicando el test de Grubbs tenemos un p.value de 0.07, siendo no significativo. Si aplicamos la función anteriormente creada que hace el proceso automático, nos devuelve el mismo p.value indicando que no hay outliers, pero tenemos dos. Por lo tanto, pasaremos a aplicar el test de Rosner.

Si se aplica este test de Rosner con k=4, tenemos como salida que no hay ningún valor outlier, cosa que no era cierta según lo realizado en el anterior Script de IQR. Podemos ver los 4 valores que más se alejan de la media, siendo: 93, 126, 112 y 131, siendo 93 y 126 los que también aparecían como outlier con la regla IQR. Por lo tanto, este test, ha fallado en este caso.

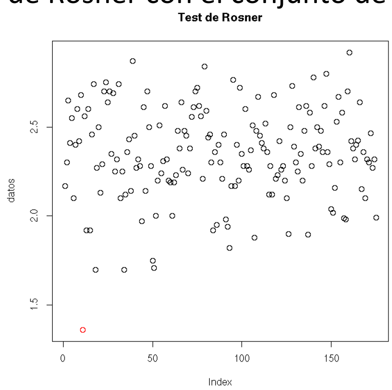
Si, como en el caso anterior, realizamos una función para realizar todo el proyecto:




```
> MiPlot_resultados_TestRosner(mydata.numeric)

Test de Rosner
Índices de las k-mayores desviaciones de la media: 93 126 112 131
De las k mayores desviaciones, ¿Quién es outlier? FALSE FALSE FALSE FALSE
Los índices de los outliers son:
Los valores de los outliers son:
Numero de datos: 178
?Qui?n es outlier?: FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE F
ALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALS
E FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE F
ALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALS
E FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE F
ALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALS
E FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE F
ALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALS
E FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE F
ALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALS
E FALSE FALSE FALSE FALSE FALSE FALSE
```

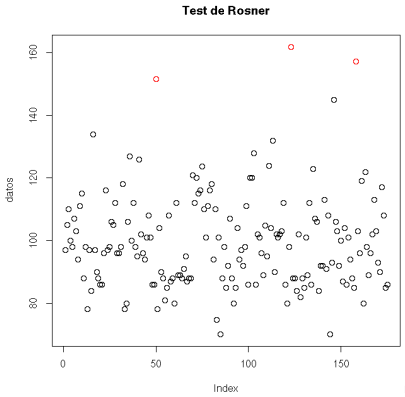
Donde se indica que no hay ningún valor que sea outlier. En cambio, si aplicamos el test de Rosner con el conjunto de un solo outlier:




```
> MiPlot_resultados_TestRosner(mydata.numeric)

Test de Rosner
Índices de las k-mayores desviaciones de la media: 11 18 34 51
De las k mayores desviaciones, ¿Quién es outlier? TRUE FALSE FALSE FALSE
Los índices de los outliers son: 11
Los valores de los outliers son: 1.36
Numero de datos: 175
?Qui?n es outlier?: FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FA
LSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FAL
SE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FAL
SE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FAL
SE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FAL
SE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE
```

Donde ahora si se indica que hay un solo outlier. Por lo que ahora voy a pasar a seleccionar la columna 5, que tiene 7 outlier, según IQR, pero eliminado 3 filas. Y el resultado del test de Rosner es:



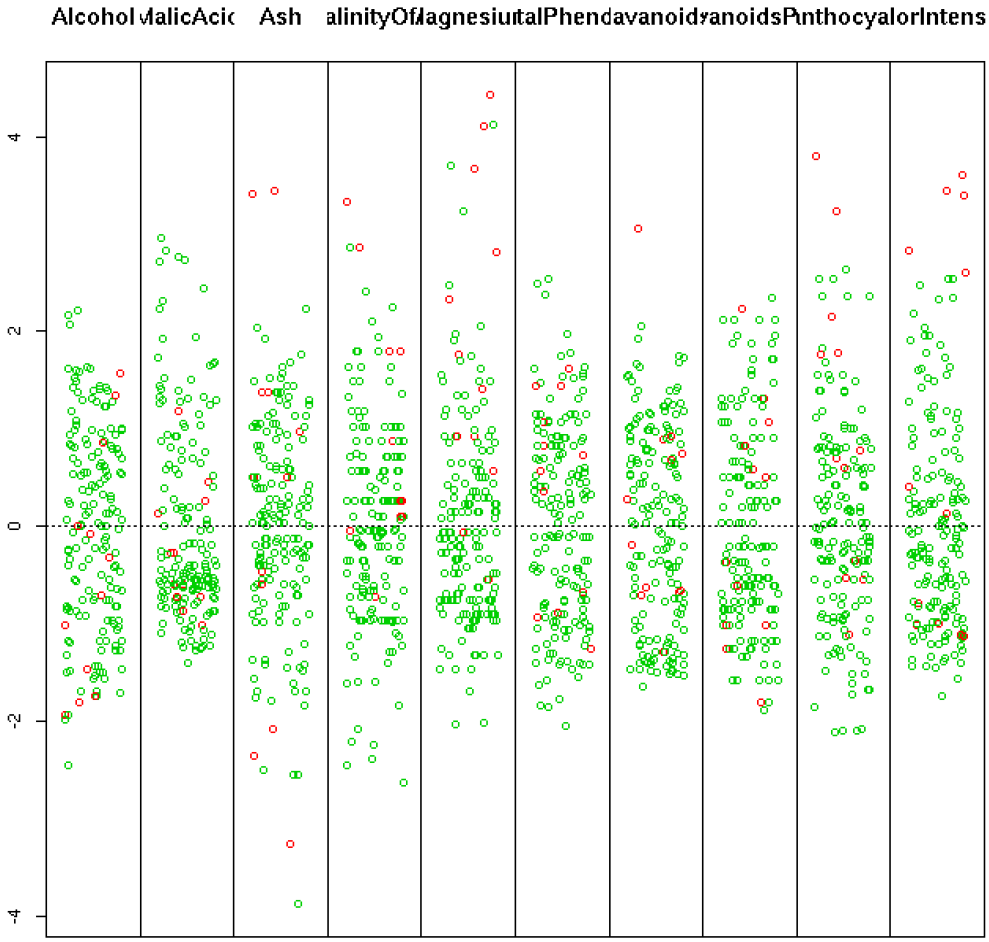

```
> MiPlot_resultados_TestRosner(mydata.numeric)

Test de Rosner
Índices de las k-mayores desviaciones de la media: 123 158 50 146
De las k mayores desviaciones, ¿Quién es outlier? TRUE TRUE TRUE FALSE
Los indices de los outliers son: 123 158 50
Los valores de los outliers son: 162 157.3309 151.6162
Numero de datos: 175
?Qui?n es outlier?: FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE F
ALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALS
E FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE T
RUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FAL
SE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FAL
SE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE F
ALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALS
E FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FA
LSE FALSE FALSE
```

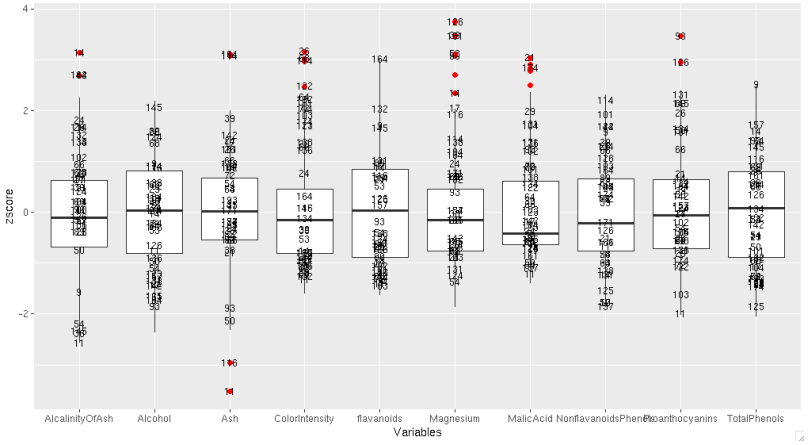
Indicando que hay 3 outliers, donde en realidad, según IQR, había 4 outliers.

4. Mahalanobis

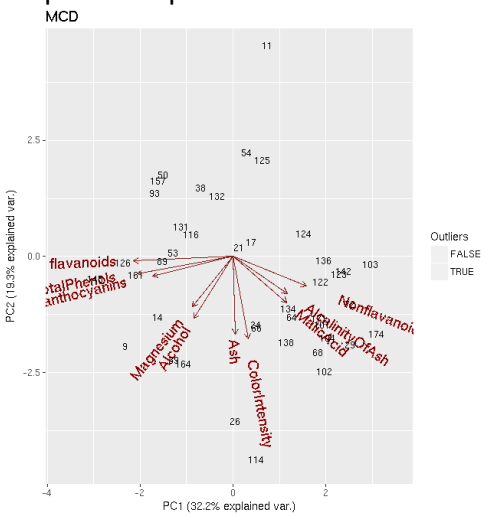
Lo primero, en el Script “C1MultiVariate_Mahalanobis.R”, será eliminar 3 variables más además de la de clase, ya que la función mvoutlier no permite más de 10 variables. Llamamos a esta función para calcular los outliers Multivariantes según la distancia de Mahalanobis considerando la estimación de la matriz de covarianzas MCD y la estimación de la media de cada variable:



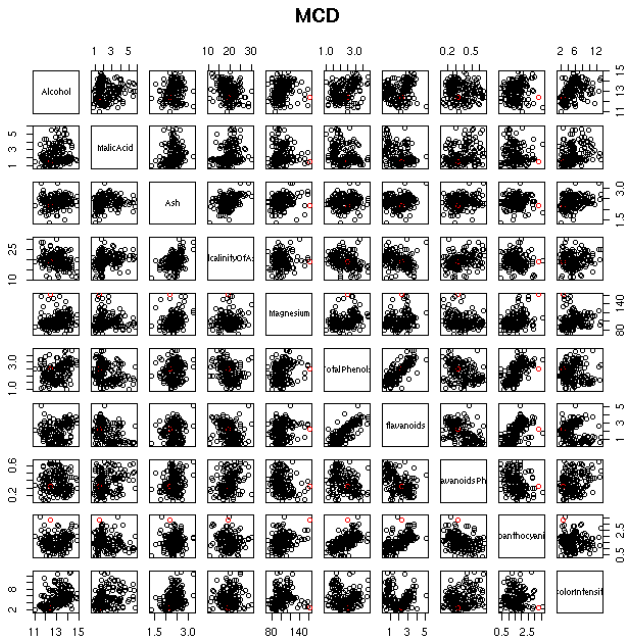
Pasamos por lo tanto a analizar los outliers. Podemos contar el número total de outliers que es, para nuestro dataset con 10 variables: 42. Si mostramos los boxplots de forma conjunta con las etiquetas de los outliers tenemos:



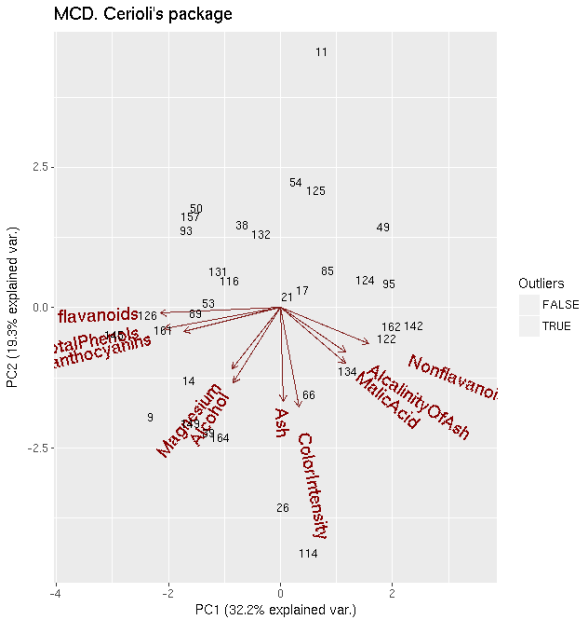
Si queremos hacer un BiPlot para ver que otuliers son multivariantes:



Si elegimos una instancia para analizarla, podemos construir una matriz con los gráficos de dispersión obtenidos al cruzar todas las variables, indicando con rojo la instancia 126 seleccionada:

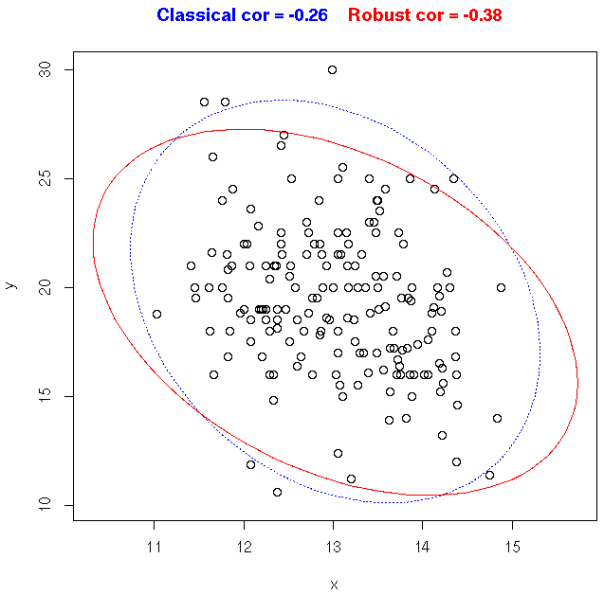


Si hacemos uso del paquete CerioliOutlierDetection para realizar lo anterior, podemos ver que el número de outliers que saca es 31, siendo anteriormente 42. La representación del BiPlot con esta función nos quedaría:



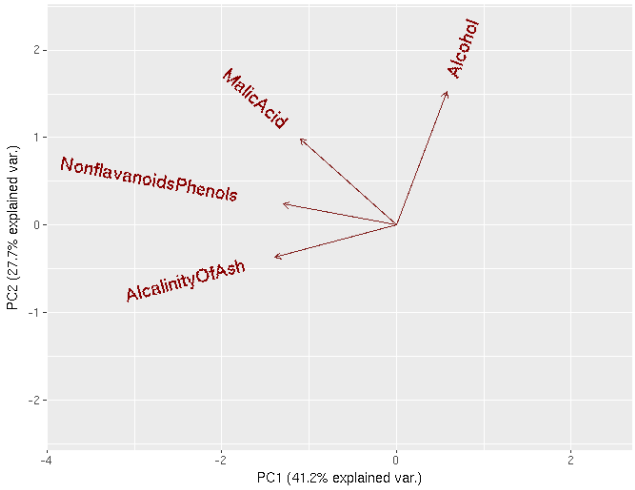
5. LOF

Para realizar esta parte, he hecho uso del Script “D1LOF.R”. Primeramente, selecciono las variables: “Alcohol”, “MalicAcid”, “AlcalinityOfAsh” y NonflavonoidsPhenols”.
Para comprobar que el método de Mahalanobis no es aplicable:



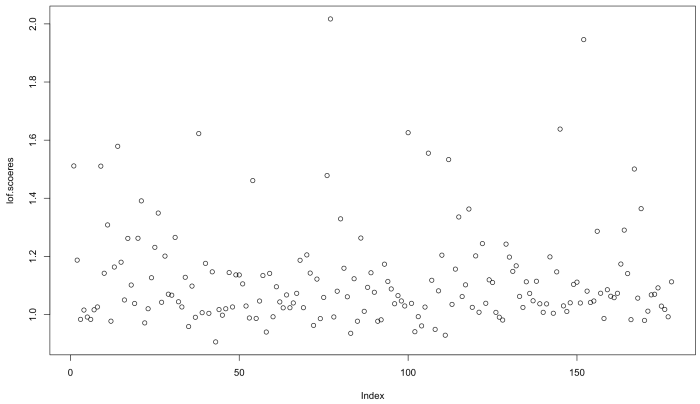
El gráfico nos muestra la dispersión al cruzar las variables 1 y 3, viendo que hay dos grupos bien definidos de datos. Usando la distancia de Mahalanobis clásica (azul) el elipsoide contiene a ambos grupos por lo que los puntos que hubiese entre ellos no serán outliers. Usando la distancia de Mahalanobis construida con la estimación robusta de la matriz de covarianzas y las correspondientes medias, el elipsoide (rojo) se construye con el grupo de datos más numeroso y todos los datos del otro grupo se marcan como outliers.

También podemos mostrar el BiPlot:

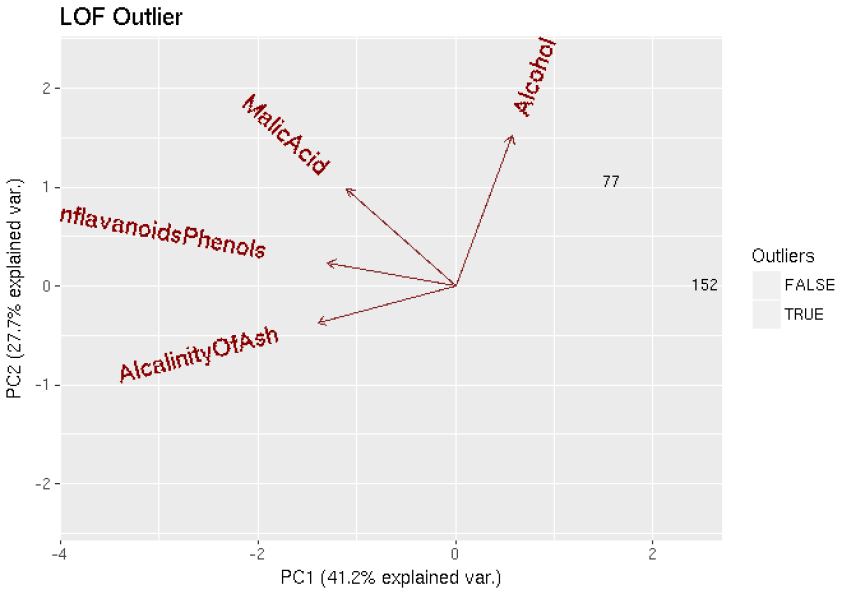


Así pues, el método de detección de outliers usando la distancia de Mahalanobis no es adecuado.

Por lo tanto, vamos a usar LOF, estableciendo primero el número de vecinos a considerar y llamando a la función “lofactor”. Teniendo como resultado gráfico:



Como se puede apreciar que hay dos valores de lof notablemente más altos que el resto, vamos a extraerlos. Los índices de los elementos son: 77 y 152. Ahora mostramos en un BiPlot con los outliers, quedando:



6. Clustering

En este apartado vamos a trabajar con las mismas columnas que en el apartado anterior, todo en el Script "D2ClusterBasedOutliers.R".

Lo primero, será realizar k-Means con 3 cluster. Con kmeans, obtenemos para cada índice de las instancias, a que cluster pertenecen. Además, obtenemos los datos de los centroides:

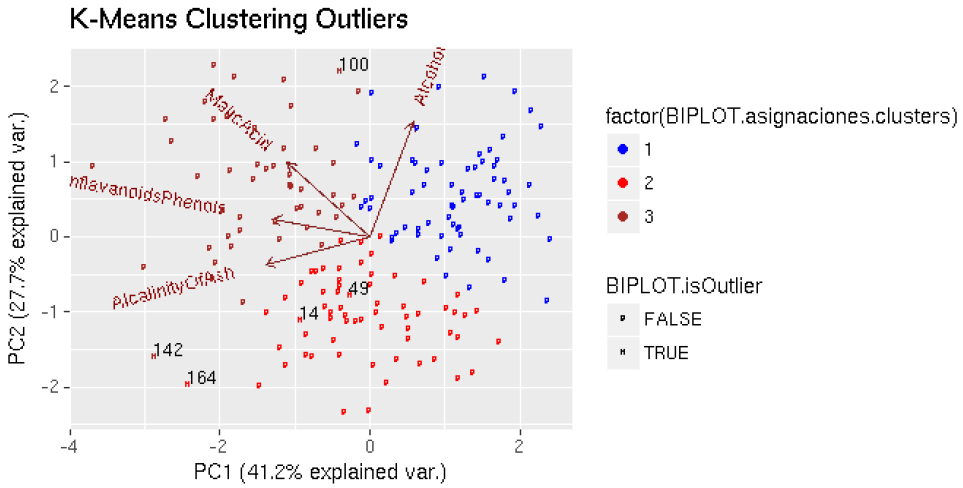
```
> centroides.normalizados
     Alcohol  MalicAcid AlcalinityOfAsh NonflavanoidsPhenols
1  0.8474526 -0.3983927      -0.6874059           -0.5390918
2 -0.9819602 -0.4177951       0.1991495           -0.1863460
3  0.1440329  1.1048502       0.6796504            0.9897126
```

Seguidamente calculamos la distancia euclídea de cada dato a su centroide. Ordenamos dichas distancias y obtenemos los índices correspondientes a los outliers: 14, 142, 164, 49 y 100.

Si creamos una función para automatizar todo el proceso, obtenemos el siguiente resultado:

```
Distancias a sus centroides de los top k clustering outliers (k-means,
usando distancia euclídea)
> top.outliers.kmeans$distancias
      14      142      164       49      100
3.258797 2.961047 2.795709 2.644486 2.633266
```

Podemos realizar un BiPlot de los outliers para los clusters:



Si queremos pasar de los centroides normalizados, y revertir la operación z-score, debemos aplicar la fórmula:

```
z-score = (dato - media.columna) / sd.columna
dato = z-score * sd.columna + media.columna
```

Pasando de:

```
> centroides.normalizados
     Alcohol  MalicAcid AlcalinityOfAsh NonflavanoidsPhenols
1  0.8474526 -0.3983927      -0.6874059           -0.5390918
2 -0.9819602 -0.4177951       0.1991495           -0.1863460
3  0.1440329  1.1048502       0.6796504            0.9897126
```

a no estar normalizados:

```
> centroides.valores
   Alcohol MalicAcid AlcalinityOfAsh NonflavanoidsPhenols
1 13.73639  1.882554        17.01116            0.2979263
2 12.19513  1.860391        20.02513            0.3427021
3 13.14376  3.599690        21.65866            0.4919851
```

Ahora vamos a usar el método de clustering PAM (Partition around medoids). Primero sacamos la distancia entre los datos. Aplicamos el método pam, y podemos obtener los índices de los medoides que nos dan: 74, 121 y 170. Los valores normalizados de los medoides son:

```
> medoides.valores.normalizados
      Alcohol  MalicAcid AlcalinityOfAsh NonflavanoidsPhenols
74   0.7211751 -0.4619091      -0.6318577          -0.52275486
121 -0.8693494 -0.6369963       0.3094167           0.02870832
170  0.1633046  1.0788587       0.4859057           0.50139104
```

y los no normalizados son:

```
> medoides.valores
    Alcohol MalicAcid AlcalinityOfAsh NonflavanoidsPhenols
74    13.63      1.81            17.2                 0.30
121   12.29      1.61            20.4                 0.37
170   13.16      3.57            21.0                 0.43
```

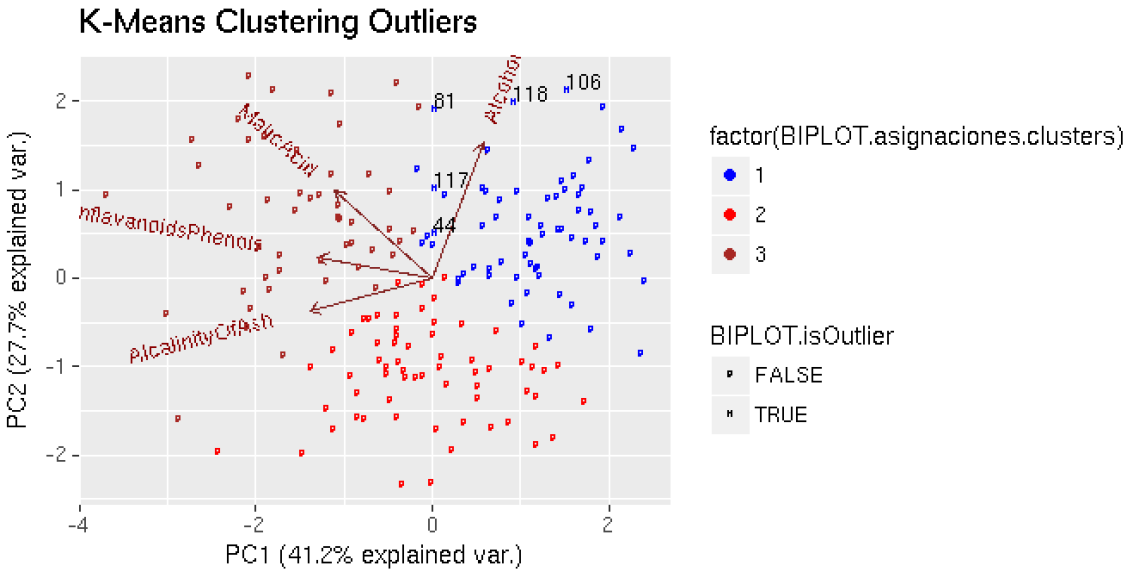
Si llamamos a la función “top_clustering_outliers” podemos mostrar los índices de los outliers:

```
> top.outliers.pam
$distancias
      14      142      169      164       26
3.202485 2.995537 2.703789 2.682645 2.589273

$indices
[1]  14 142 169 164  26
```

Partes de ampliación:

Calcular la distancia de cada punto a su centoride usando la distancia de Mahalanobis:



Partes de ampliación:

Calcular la distancia de forma relativa obteniendo como resultado:

```
Índices de los top k clustering outliers (k-means, usando distancia relativa)
> top.outliers.kmeans.distancia.relativa$indices
[1] 142 100  14  21  26
Distancias a sus centroides de los top k clustering outliers (k-means, usando
distancia relativa)
> top.outliers.kmeans.distancia.relativa$distancias
     142      100       14       21       26
4.549693 4.046052 3.876359 3.767593 3.730932
```