Desarrollo y análisis comparativo de una librería en R para la detección de Outliers

Diego Anas Rejas Haddioui

1 de noviembre de 2019

Índice general

Índice de figuras

Índice de cuadros

Listings

Capítulo 1

Introduction

1.1. Abstract

Text of Abstract

1.2. Resumen

Texto del Resumen

1.3. Preámbulo

En la detección de anomalías, el objetivo es encontrar objetos que son diferentes del resto de los demás objetos. A menudo, estos objetos anómalos son denominados con la palabra inglesa "outliers", dado que, en un gráfico de datos, se encuentran lejos del resto de los puntos de datos. La detección de anomalías también es conocida como detección de desviaciones, porque los objetos anómalos tienen atributos que se desvían significativamente de los atributos esperados o típicos, o como minería de excepciones, ya que las anomalías son excepcionales de algún modo.

En este trabajo se utilizaran los términos anomalía, excepción, valor aberrante u *outlier* de manera indistinta.

Existen variedad de enfoques para la detección de anomalías desde distintas áreas, incluyendo estadística, aprendizaje máquina y minería de datos. Todos intentan capturar la idea de que un objeto anómalo es inusual o inconsistente con respecto al resto de objetos. Aunque los objetos o eventos inusuales son, por definición, relativamente raros, esto no quiere decir que no ocurren frecuentemente en términos absolutos. Por ejemplo, un evento que

ocurre 1 de cada mil veces puede darse un millón de veces cuando miles de millones de eventos son considerados.

En el mundo natural, en la sociedad humana, o el dominio de los conjuntos de datos, la mayoría de eventos y objetos son, por definición, comunes u ordinarios, pero somos conscientes de la posibilidad de que se den objetos inusuales o extraordinarios. Esto incluye, por ejemplo, temporadas excepcionalmente secas o lluviosas, atletas famosos, o un valor de un atributo mucho más grande o pequeño que todos los demás. El interés en eventos y objetos anómalos nace del hecho que suelen ser de importancia: una sequía amenaza la agricultura, el rendimiento de un atleta puede llevarle a la victoria, y resultados anómalos en un experimento pueden indicar o bien un problema con el experimento o un nuevo fenómeno para ser investigado.

Los siguientes ejemplos ilustran diferentes aplicaciones para las cuales las anomalías son de considerable interés.

- Detección de fraude. El comportamiento de compra de alguien que ha robado una tarjeta de crédito es probablemente diferente al del propietario legítimo de la tarjeta. Las compañías de tarjetas de crédito intentan detectar un robo examinando el patrón de compre que caracteriza un robo o detectando un cambio en el comportamiento típico. Enfoques similares son utilizados en otro tipo de fraude.

- Detección de intrusiones. Los ataques a sistemas y redes informáticos son comunes. Aunque muchos de estos ataques son obvios, como los ataques de denegación de servicio, otros ataques, como aquellos diseñados para recabar información encubiertamente, son difíciles de detectar. Muchas de estas intrusiones pueden ser detectadas monitorizando los sistemas y redes en busca de comportamientos inusuales.

- Perturbaciones en ecosistemas. En el mundo natural, hay eventos atípicos que pueden tener efecto en las personas. Ejemplos de esto incluyen huracanes, inundaciones, sequías, olas de calor y fuegos. El objetivo es poder predecir estos hechos e identificar sus causas.

- Salud pública. Hospitales y clínicas médicas informan y envían estadísticas a organizaciones nacionales para análisis posteriores. De esta manera se pueden detectar brotes de enfermedades o problemas, por ejemplo, con programas de vacunación.

- Medicina. Para un paciente en particular, síntomas inusuales o resultados de pruebas pueden indicar un problema de salud importante. De cualquier manera, el hecho de que el resultado de una prueba sea anómalo depende de otras características del paciente, como la edad o el

género. Es de especial interés ya que la evaluación de una resultado como excepcional o no incurre un coste adicional – pruebas innecesarias a un paciente sano o graves daños a un paciente al que no se le ha diagnosticado la enfermedad que sufre.

Aunque el interés reciente en la detección de anomalías ha sido guiado por aplicaciones en las que las anomalías son el principal objetivo, históricamente, la detección (y eliminación) de anomalías ha sido vista como una técnica para mejorar el análisis típico de objetos de datos. Por ejemplo, un número relativamente pequeño de anomalías puede distorsionar la media y la desviación típica de un conjunto de valores o alterar el conjunto de agrupaciones detectadas por un algoritmo de agrupación. Por esto, la detección de valores aberrantes ha sido a menudo una parte del preprocesamiento de datos.

Después de algunos preliminares, se provee una discusión detallada de algunos enfoques a la detección de anomalías, ilustrándolas con ejemplos de técnicas específicas.

1.3.1. Preliminares

Antes de empezar con la discusión de algoritmos específicos, se provee de información adicional. Específicamente, exploramos la causa de las anomalías, consideramos distintos enfoques a la detección de anomalías, haremos distinción entre los enfoques dependiendo si utilizan información categórica, y describimos problemas comunes en las técnicas de detección de anomalías.

Causas

Las siguientes son algunas de las causas más comunes de las anomalías: datos de diferentes clases, variación natural, y errores de medición y recolección de datos.

Datos de diferentes clases. Un objeto puede ser distinto de otros, es decir, anómalo, porque son de distinto tipo o clase. Por ejemplo, alguien cometiendo un fraude con una tarjeta de crédito pertenece a una clase diferente de la clase de los usuarios legítimos de tarjetas. La mayoría de ejemplos presentados al principio de este capítulo, como fraude, intrusión, brotes de enfermedades, y resultados anormales en pruebas, son anomalías que representan a un clase diferente de objetos. Tales anomalías son comúnmente de interés particular y son el principal enfoque de detección de anomalías en el campo de la minería de datos.

La idea de que los objetos anómalos se originan de otra fuente que la de la mayoría de objetos es declarada en la famosa definición de valor aberrante por el estadístico Douglas Hawkins.

La definición de Hawkins de una anomalía es una observación que difiere tanto del resto de observaciones que levanta sospechas sobre si fue generada por un mecanismo diferente.

Variación natural. Muchos conjuntos de datos pueden ser modelados por distribuciones estadísticas, como la distribución normal (o gaussiana), donde la probabilidad de que se dé un objeto disminuye rápidamente según la distancia al centro de la distribución del objeto aumenta. En otras palabras, la mayoría de los datos están cerca de un centro (la media), y la probabilidad de que un objeto difiera significativamente de esta media es pequeña. Por ejemplo, una persona excepcionalmente alta no es anómala en el sentido de pertenencia a otra clase, pero sí sólo en el sentido de que tiene un valor extremo para una característica (en este caso la altura) poseída por todos los objetos. Las anomalías que representan variaciones extremas o improbables son a menudo interesantes.

Errores de medición y recolección de datos. Los errores en el proceso de recopilación de información son otra fuente de valores aberrantes. Por ejemplo, una medida puede ser registrada incorrectamente debido a un error humano, un problema con el equipo de medida, o la presencia de ruido. La meta es eliminar tales anomalías, ya que no son de ningún interés y sólo reducen la calidad de los datos y los posteriores análisis. Es por eso que la eliminación de este tipo de anomalías es el principal objetivo en el preprocesamiento de datos, en concreto en la limpieza de datos.

En resumen, una anomalía puede resultar de las anteriores causas u otras no consideradas. Las anomalías en un conjunto de datos pueden tener varias fuentes, y la causa subyacente de una anomalía en particular es a menudo desconocida. En la práctica, las técnicas de detección de anomalías se centran en encontrar valores que difieren substancialmente de la mayoría de objetos, y las propias técnicas no son afectadas por la causa de la anomalía. Por esto, la causa subyacente de la anomalía es sólo importante con respecto a la aplicación prevista.

Enfoques a la detección de anomalías

Se describirá algunas de las técnicas de detección de anomalías y sus definiciones de anomalía asociadas. Existe cierto solapamiento entre estas técnica:

Basada en modelos. Muchas técnicas de detección de anomalías construyen, en primer lugar, un modelo de los datos. Las anomalías son objetos

que no encajan bien en el modelo. Por ejemplo, un modelo de la distribución de los datos puede ser creado utilizando los datos para estimar los parámetros de una distribución de probabilidad. Un objeto no encaja en el modelo, es decir, es una anomalía, si es poco probable dada la distribución. Si el modelo es un conjunto de agrupaciones, entonces la anomalía es un objeto que no pertenece a una agrupación de manera significativa. Cuando es un modelo de regresión, una anomalía es un objeto que está relativamente lejos del valor previsto.

Dado que puede considerarse que objetos anómalos y normales definen clases distintas, se pueden utilizar técnicas de clasificación para construir modelos de estas dos clases. Por supuesto, las técnicas de clasificación sólo pueden ser utilizadas si se dispone de los atributos de clase de al menos un número de objetos para construir conjuntos de entrenamiento. También tiene que considerarse que las anomalías son relativamente raras al elegir la técnica de clasificación y las medidas que utilizar para su evaluación.

En algunos casos es difícil construir un modelo, porque la distribución estadística es desconocida o no se dispone de un conjunto de entrenamiento, por ejemplo. En estas situaciones se pueden utilizar otras técnicas como las siguientes.

Técnicas basadas en proximidad. A menudo es posible definir medidas de proximidad entre objetos, y ciertos enfoques en detección de anomalías se basan en la proximidad. Objetos anómalos son aquellos que se encuentran lejos del resto de objetos. Muchas de las técnicas en este área están basadas en la distancia. Cuando los datos pueden ser mostrados en diagramas de dos o tres dimensiones, las anomalías basadas en distancia pueden ser detectadas visualmente, observando aquellos puntos que se encuentran separados de la mayoría del resto de objetos.

Técnicas basadas en densidad. Las estimaciones de la densidad de los objetos son relativamente simples de computar, especialmente si se dispone una medida de distancia entre los objetos. Los objetos que están en regiones de baja densidad están relativamente lejos de sus vecinos, y pueden ser considerados aberrantes. Un enfoque más sofisticado incorpora el hecho de que los pueden tener regiones de diferente densidad, y clasifica un punto como anomalía si tiene una densidad local relativamente menor que la mayoría de sus vecinos.

Uso de atributos de clase

Básicamente, existen 3 maneras de abordar la detección de anomalías: no supervisada, supervisada, y semi-supervisada. La gran distinción es el grado en que los atributos de clase (anomalía o normal) están disponibles para al

menos un parte de los datos.

Detección de anomalías supervisada.Uso de atributos de clase Las técnicas para la detección de anomalías supervisada, requieren la existencia de un conjunto de entrenamiento con ambas clases, anomalías y objetos normales, presentes. (Puede haber más de una clase de objetos normales). Como se ha mencionado previamente, las técnicas que tratan el llamado problema de la clase aberrante son particularmente interesantes porque las anomalías son relativamente raras con respecto a los objetos normales.

Detección de anomalías no supervisada. En muchas situaciones prácticas, los atributos de clase no están disponibles. En estos casos, el objetivo es asignar una puntuación (o una categoría) a cada instancia de datos que refleje el grado en que el objeto es anómalo. Nótese que la presencia de muchas anomalías que son similares entre sí puede causar que todas ellas sean clasificadas como normales o que se las asigne una puntuación de anomalía baja. Por esto, para que la detección no supervisada sea fructuosa, las anomalías deben ser distintas las unas de las otras y de los objetos normales.

Detección de anomalías semi-supervisada. En ocasiones el conjunto de entrenamiento contiene objetos normales con categorías, pero no contiene información sobre los objetos anómalos. En este marco, la meta es encontrar una categoría anómala o puntuación para un conjunto de datos utilizando la información de los objetos normales ya clasificados. Nótese que, en este caso, la presencia de muchas anomalías relacionadas entre sí en el conjunto no afecta la evaluación de anomalías. A pesar de esto, en la práctica, puede resultar difícil dar con un conjunto de pequeño que represente los objetos normales.

Problemas

Existen variedad de problemas importantes que han de ser tratados a la hora de ocuparse de anomalías.

Número de atributos utilizados para definir una anomalía. La cuestión de si un objeto es anómalo basándose en un solo atributo es también cuestión de si ese valor para ese atributo es anómalo. Pero, ya que un objeto puede tener muchos atributos, puede tener valores anómalos para algunos de ellos pero valores ordinarios para otros atributos. Además, un objeto puede ser anómalo aunque ninguno de los valores para sus características sea anómalo. Por ejemplo, es común que haya personas que midan 1 metro (como niños) o que pesen 100 Kg, pero una persona de 1 metro que pese 100kg es bastante raro. Una definición general de anomalías debe especificar como los diferentes valores para los atributos son utilizados para determinar si un objeto es anómalo o no. Esto es particularmente importante cuando el número

de dimensiones del conjunto es grande.

Perspectiva local contra global. Un objeto puede ser considerado inusual comparado con el resto de objetos, pero no comparado con los objetos en su vecindad. Por ejemplo, una persona de 2 metros es inusualmente alta comparada con el resto de la población, pero no comparado con los jugadores profesionales de baloncesto.

Grado de anomalía. La evaluación de si un objeto es anómalo o no es tratada por algunas técnicas de manera binaria: o bien es una anomalía o no lo es. Pero, frecuentemente, esto no refleja la realidad subyacente de que algunos objetos son más anómalos que otros. Es por esto que disponer de una manera de evaluar el grado de anomalía de un objeto es conveniente. Esta evaluación es denominada como factor, grado o puntuación de anomalía.

Identificación de una o varias anomalías al mismo tiempo. En algunas técnicas, las anomalías son tratadas una por una, esto es, el objeto más anómalo es identificado y eliminado y el proceso se repite. En otras, un conjunto de anomalías es identificado a la vez. Las técnicas que intentan identificar una anomalía tras otra sufren a menudo el problema de enmascaramiento, donde la presencia de varias anomalías oculta la presencia de todas. Por otra parte, las técnicas que clasifican a la vez las anomalías pueden sufrir el problema de inundación, donde objetos normales son clasificados como anómalos. En enfoques basados en modelos, estos efectos pueden ocurrir porque la presencia de anomalías deforma el modelo de datos.

Evaluación. Si se dispone de atributos de categoría para la identificación de anomalías y objetos normales, la efectividad de una técnica de detección de anomalía puede ser evaluada mediante el uso de medidas de rendimiento discutidas en la sección [enlace]. Pero, como a menudo la clase anómala es mucho más pequeña que la clase normal, las medidas como precisión y falsos positivos son más apropiados que el rendimiento. Si las categorías de clase no están disponibles entonces la evaluación es difícil. Independientemente, en las técnicas basadas en modelos, se puede evaluar la efectividad con respecto a la mejora del modelo al eliminar las anomalías.

Eficiencia. Hay grandes diferencias entre los costes computacionales de los distintos esquemas de detección de anomalías. Enfoques basados en la clasificación pueden requerir gran cantidad de recursos para crear los modelos de clasificación, pero normalmente se pueden aplicar sin coste alguno. De manera parecida, los enfoques estadísticos crean modelos estadísticos y pueden clasificar un objeto en un tiempo constante. Los enfoques basados en distancia tienen normalmente una complejidad temporal $O(n^2)$, debido a que la información que requieren sólo puede ser obtenida mediante una matriz de distancias. Estos costes temporales pueden reducirse en algunos casos, con espacios de datos de pocas dimensiones, con el uso de estructuras de datos y

algoritmos especiales.

1.4. Planificación del trabajo

1.5. Objetivos del trabajo

1.6. ¿Por qué R?

Disponible como Free Software con Licencia Pública General GNU, por lo que disponemos del código fuente. Multiplataforma, compilable y ejecutable en gran variedad de plataformas UNIX, al igual que en Windows y MacOS.

Podemos extender o modificar las funcionalidades ofrecidas por R programando e implementando nuestros propios paquetes y subirlos a los repositorios.

A fecha actual, disponibles 8110 paquetes 'fuente?', de los cuales 8063 cuentan con el paquete binario disponible para Windows. Over 16 years or R Project history (Artículo que habla de la historia de R y su evolución, haciendo énfasis en el número de paquetes disponibles en el repositorio CRAN).

Alternativa de Software Libre a Matlab, SAS o SPSS.

Capítulo 2

Métodos Estadísticos

Son enfoques basados en modelos, esto es, un modelo es creado para los datos, y los objetos son evaluados con respecto a cómo de bien encajan en el modelo. La mayoría de enfoques estadísticos a la detección de anomalías están basados en crear un modelo de distribución de probabilidad y considerar cuan probable son los objetos dentro de ese modelo.

Definición probabilística de una anomalía: Una anomalía es un objeto que tiene una baja probabilidad con respecto al modelo de distribución de probabilidad del conjunto de datos.

Un modelo de distribución de probabilidad es construido a partir de los datos estimando los parámetros de una distribución específica. Si se asume que los datos tienen una distribución gaussiana, entonces la media y la desviación típica de la distribución subyacente puede ser aproximada con la media y desviación típica de los datos. La probabilidad de cada objeto bajo esa distribución puede entonces ser calculada.

Entre los problemas a afrontar al tratar el problema de la detección de anomalías desde un enfoque estadístico, destacan:

Identificación de la distribución específica del conjunto. Aunque muchos tipos de datos pueden ser descritos mediante un pequeño número de distribuciones más comunes, como la distribución normal (gaussiana), de Poisson, o binomial, otras distribuciones no estándar también se dan en conjuntos de datos. Por supuesto, si el modelo equivocado es el elegido entonces un objeto normal puede ser rechazado como excepcional. Por ejemplo, se puede interpretar que los datos provienen de una distribución normal, cuando en realidad se originan de otra distribución en la que valores alejados de la media son más probables. Distribuciones estadísticas de este tipo son conocidas como distribuciones de cola pesada.

El número de características utilizadas. La mayoría de métodos de detección de anomalías son aplicados a un solo atributo, aunque algunas han

sido definidas para datos multivariables.

Mezclas de distribuciones. Los datos pueden ser descritos como una mezcla de distribuciones, y las técnicas de detección pueden desarrollarse a partir de estos modelos. Aun siendo más potentes, estos modelos con más complicados de entender y de utilizar. Las distribuciones han de ser identificadas antes de que cualquier objeto pueda ser clasificado.

Los enfoques estadísticos para la detección de excepciones están desarrollados sobre una fuerte base de técnicas estadísticas, como la estimación de parámetros de una distribución. Cuando existe el conocimiento suficiente sobre los datos y el tipo de prueba que debe de ser aplicada, estos tests pueden ser muy efectivos. Hay gran variedad de pruebas estadísticas para un único atributo, pero hay menos opciones disponibles para datos multivariables, y estas pruebas no cumplen muy bien su objetivo en espacios de muchas dimensiones.

A continuación, distintos métodos gráficos y pruebas estadísticas basados en modelos, tanto univariables como multivariables, serán estudiados.

2.1. Histograma

El histograma, descrito por primera vez en 1895 por Karl Pearson [cita], es una representación gráfica de una variable numérica continua en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados en esa clase. Es una estimación de la distribución de probabilidad de una variable continua.

Los histogramas, junto a los diagramas de caja, son útiles a la hora de obtener una idea general de la distribución del conjunto. Sirven para obtener una "primera vista" general, o panorama, de la distribución de la población, o de la muestra, respecto a una característica, cuantitativa y continua (como la longitud o el peso). De esta manera ofrece una visión de grupo permitiendo observar una preferencia, o tendencia, por parte de la muestra o población por ubicarse hacia una determinada región de valores dentro del espectro de valores posibles (sean infinitos o no) que pueda adquirir la característica.

Así pues, podemos evidenciar comportamientos, observar el grado de homogeneidad, acuerdo o concisión entre los valores de todas las partes que componen la población o la muestra, o, en contraposición, poder observar el grado de variabilidad, y por ende, la dispersión de todos los valores que toman las partes. A partir del histograma podemos observar la desviación o inclinación de los datos, la existencia de anomalías o valores aberrantes, o la presencia de múltiples modelos en los datos. Por otra parte, es posible no evidenciar ninguna tendencia y obtener que cada miembro de la población toma por su lado y adquiere un valor de la característica aleatoriamente sin mostrar ninguna preferencia o tendencia, entre otras cosas.

Matemáticamente, un histograma es una función m_i que cuanta el número de observaciones que hay en cada categoría disjunta (clases o barras), donde el gráfico del histograma es sólo una manera de representar un histograma. Así, sea n el número total de observaciones o valores de una variable y sea k el número total de clases, el histograma m_i cumple:

$$n = \sum_{i=1}^{k} m_i$$

Hay distintos tipos de histograma. El más común y simple es el **histograma de frecuencias absolutas**. Para representar este histograma primero se debe dividir el rango de valores en clases o "barras" y tras eso contar cuantos valores hay en cada intervalo. El área de cada barra muestra la frecuencia de ocurrencia de cada intervalo, y ya que cada barra tiene el mismo ancho, la altura índica cuantas observaciones hay en esa categoría. Una variación común de este tipo es el **histograma de frecuencias relativas**: el histograma

es normalizado para mostrar la frecuencia relativa de cada clase, mostrando la proporción de casos correspondientes a cada clase, siendo la suma de las alturas igual a 1.

No existe el número perfecto de categorías, ya que diferentes tamaños de las barras pueden mostrar diferentes características de los datos. En la teoría se han realizado intentos de determinar un número óptimo de clases, pero estos métodos normalmente hacen importantes asunciones sobre el tipo de distribución que siguen los datos. Diferentes anchos de las barras pueden ser apropiados dependiendo de la distribución real de los datos y el objetivo del análisis, por lo que, a menudo, la experimentación es lo necesario para determinar una anchura o número de clases apropiados.

Utilizar clases más anchas donde la densidad es baja reduce el ruido causado por el muestreo aleatorio; mientras que utilizar clases más estrechas donde la densidad es alta (de manera que la señal enmudece el ruido) resulta en una mejor precisión en la estimación de densidad. Así pues, variar la anchura de clase en un histograma puede resultar beneficioso. De cualquier manera, lo más común es utilizar clases de la misma anchura.

2.1.1. hist{graphics}

2.1.2. hist{lattice}

2.1.3. hist{histogram}

2.2. Diagrama de caja y bigotes

El diagrama de caja y bigotes, o simplemente diagrama de caja, es una presentación visual la cual nos permite visualizar la distribución de un conjunto de datos numéricos mediante sus cuartiles. Este diagrama, denominado *"box-and-whisker plot"*, fue descrito por primera vez por John W. Tukey en *"Exploratory Data Analysis"*[cita].

Es la técnica más común para representar el *resumen de 5 números*, que consiste en el valor mínimo y máximo del rango, los cuartiles inferior y superior, y la mediana. Este conjunto de valores es una manera rápida de resumir la distribución de un conjunto de datos. Además, esta técnica reducida de representar el *resumen de 5 números* ofrece una manera de comparar conjuntos de manera más simple y directa, ya que sólo esas características han de ser analizadas. También informa sobre los valores atípicos y de la simetría de los datos del conjunto.

La construcción típica del diagrama de caja, divide la distribución de los datos en cuartiles, esto es, cuatro subconjuntos de igual tamaño. Una caja es dibujada para indicar la posición de los cuartiles superior e inferior; el interior de esta caja indica el rango intercuartílico, que es el área entre el cuartil inferior y el superior y abarca el 50 % de la distribución. Líneas, comúnmente denominadas bigotes, se extienden hasta el extremo de los datos, es decir, el mínimo y máximo valor del conjunto de datos, o:

- Los valores extremos dentro del rango de $1,5$ veces el RIC (Rango Intercuartílico)
- Una vez la desviación típica por encima y por debajo de la media aritmética de los datos
- Los percentiles 9º y 91º
- Los percentiles 2º y 92º

Aquellos datos que queden fuera de los bigotes han de ser tratados como valores atípicos y son normalmente representados con un punto, un círculo o un asterisco. Finalmente, una línea cruza la caja en el lugar de la mediana del conjunto.

En ocasiones estas 5 características no son suficientes, como al hacer comparaciones entre distintos conjuntos de datos, donde disponer información sobre el número de observaciones, al igual que información adicional, puede ser útil para no llegar a conclusiones erróneas. En *"Variations of Box Plots"* [cita] tres variaciones son descritas: ancho variable, con muesca, y la combinación de los dos: La primera variante es la de **caja de ancho variable**, donde el ancho de la caja es proporcional al tamaño de los datos, normalmente la raíz cuadrada de la cardinalidad. Esta guía visual indica al usuario la

distinción en el número de observaciones entre los conjuntos y puede ayudar a evitar interpretaciones erróneas. La segunda variante descrita es el **diagrama de caja con muesca**, dónde se añade una muesca o estrechamiento a la caja alrededor de la mediana. Las muescas son útiles pues ofrecen una indicación de la importancia de la diferencia entre las medianas de dos poblaciones (o gráficos): si las muescas de dos cajas no se solapan, esto quiere decir que existe una diferencia relevante entre las medianas. Su ancho es proporcional al RIC e inversamente proporcional a la raíz cuadrada de N: normalmente $\frac{1.58 \times RIC}{\sqrt{N}}$. La tercera variante descrita es el diagrama de caja de anchura variable con muesca, que combina las dos variantes anteriores. Estas variaciones permiten comparar unos diagramas con otros con mayor precisión.

Una de las mayores ventajas del diagrama de caja es la simplicidad de su diseño. La información más importante es expresada rápidamente, y la caja es una muestra de su distribución. Características generales como la simetría de la distribución, el lugar de su valor central, y la dispersión de las observaciones son inmediatamente aparentes. Esta representación permite al usuario añadir información extra sobre el conjunto y modificar el diagrama con propósitos específicos.

Por otra parte, una de las desventajas de su simple diseño es que puede ocultar características distinguibles de una distribución y se pueden generar diagramas similares para distribuciones muy distintas.

2.2.1. Boxplot del paquete Graphics

Este método se encuentra en uno de los paquetes que por defecto incluyen las distribuciones de R.

Listing 2.1: Uso por defecto del método boxplot

```
boxplot(x, ...)
# Método S3 para la clase 'formula'
boxplot(formula, data = NULL, ..., subset, na.action =
    NULL)

## Llamada S3 por defecto
boxplot(x, ..., range = 1.5, width = NULL, varwidth =
    FALSE, notch = FALSE, outline = TRUE, names, plot =
    TRUE)
```

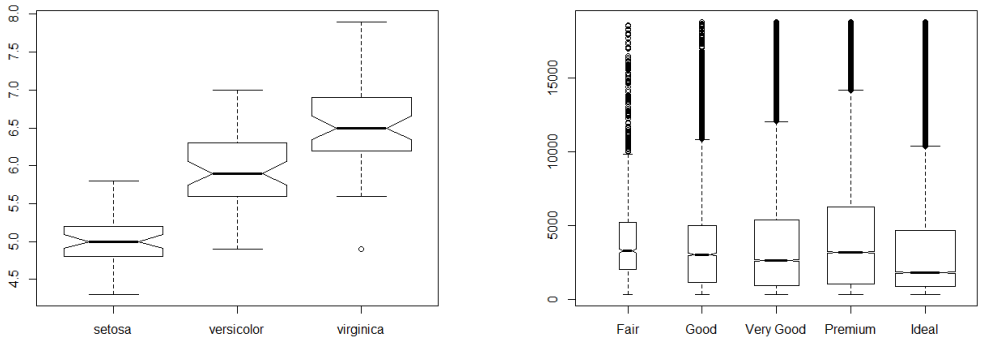
Argumentos

`formula`	Una formula del tipo `y grp`, donde `y` es un vector de valores numéricos a dividir en grupos de acuerdo a la variable `grp` (normalmente un factor numérico)
`data`	Un `data.frame` o lista del cual las variables de `formula` son extraídas
`subset`	Un vector opcional que especifica un subconjunto de valores utilizados para generar el gráfico
`na.action`	Función que determina la acción a tomar cuando el conjunto contiene valores nulos.
`x`	Especifica los datos de los que se van a generar el diagrama. Un vector numérico, o una lista contenedora de dichos vectores.
`range`	Determina la extensión de los bigotes desde la caja. Si el valor es positivo, los bigotes se extienden hasta el valor mas extremo no mayor que este valor multiplicado por el rango intercuartílico de la caja. Si es cero, los bigotes se abarcan todos los valores. El valor por defecto es $1,5$
`width`	Vector que determina el ancho relativo de las cajas que conforman el gráfico
`varwidth`	Si `TRUE`, las cajas son representadas con ancho variable proporcional a la raíz cuadrada del número de observaciones en el conjunto
`notch`	Si `TRUE`, el tipo de diagrama de caja será con muesca.
`outline`	Si `FALSE`, los datos atípicos no son representados
`boxwex`	Un factor de escala a aplicar a cada caja. Recomendado cuando únicamente hay pocas cajas ya que mejora la apariencia del gráfico haciendo las cajas más estrechas.
`plot`	Si `TRUE`, se genera el gráfico, si no, la función solo devuelve el resumen de los datos en los que los diagramas se han construido

Ejemplos

Listing 2.2: generación de dos diagramas de cajas

```
boxplot(Sepal.Length ~ Species, data=iris, notch=TRUE)
boxplot(price ~ cut, data=diamonds, notch=TRUE,
    varwidth=TRUE)
```



(a) Distribución de la longitud del sépalo por cada especie de la flor del Iris.

(b) Distribución de los precios por tipo de corte del diamante

Figura 2.1: Diagramas de caja de ancho variable y con muesca generados por boxplot con los conjuntos Iris y Diamonds.

2.2.2. Boxplot del paquete ggplot2

Dentro de esta librería se encuentra el metodo `geom_boxplot` para generar diagramas de cajas. Para producir un diagrama de caja, primero se debe crear un objeto `ggplot`, y luego invocar `geom_boxplot` para añadirle una capa al objeto, que contendría el diagrama de caja y bigotes.

Una vez invocado `ggplot`, se debe sumar al objeto una capa que contenga el diagrama de cajas mediante `geom_boxplot`, que hereda los argumentos de la llamada previa a `ggplot`.

Listing 2.3: llamadas por defecto de ggplot y geom_boxplot

```
ggplot(data = NULL, mapping = aes(), ..., environment =
    parent.frame())
geom_boxplot(mapping = NULL, data = NULL, stat = "
    boxplot", position = "dodge", ..., outlier.colour =
```

```
NULL, outlier.color = NULL, outlier.shape = 19,
outlier.size = 1.5, outlier.stroke = 0.5, notch =
FALSE, notchwidth = 0.5, varwidth = FALSE, na.rm =
FALSE, show.legend = NA, inherit.aes = TRUE)
```

Argumentos ggplot

`data`	Conjunto de datos utilizados para generar el gráfico. Si no es del tipo data.frame, será convertido. Si no es especificado, deberá serlo en cada capa a que quiera añadirse al gráfico
`mapping`	Lista por defecto de relaciones estéticas utilizadas para el gr
`...`	Otros argumentos comunicados a métodos adicionales.
`environment`	Entorno en el cual ggplot buscara una variable definida en la relaciones estéticas si no lo encuentra en los datos.

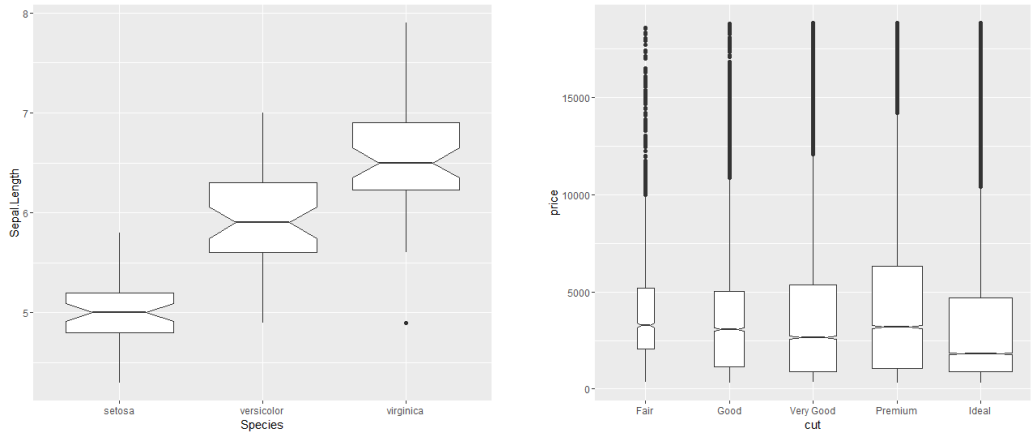
Argumentos geom_boxplot

`position`	Ajuste de posición
`...`	Argumentos transmitidos al método layer, generalmente estéticos.
`outlier.colour, outlier.color, outlier.shape, outlier.size, outlier.stroke`	Parámetros estéticos para representar los valores atípicos
`notch`	valor lógico que determina si el tipo de caja es con o sin muescas.
`notchwidth`	en los diagramas con muesca, ancho de la muesca relativo al del cuerpo de la caja (defecto 0,5)
`varwidth`	valor lógico que determina si el tipo de caja es de ancho variable o no
`na.rm`	valor logico que determina si muestra un aviso al eliminar valores nulos
`coef`	Longitud de los bigotes como producto del RIC, por defecto 1,5

Ejemplos

Listing 2.4: generación de dos diagramas de cajas

```
p <- ggplot(iris, aes(Species, Sepal.Length))
p + geom_boxplot(notch = TRUE)
p <- ggplot(data=diamonds, aes(cut, price))
p + geom_boxplot(varwidth = TRUE, notch = TRUE)
```



(a) Distribución de la longitud del sépalo por cada especie de la flor del Iris.

(b) Distribución de los precios por tipo de corte del diamante

Figura 2.2: Diagramas de caja de ancho variable y con muesca generados por geom_boxplot con los conjuntos Iris y Diamonds.

2.3. Diagrama de bolsa

El diagrama de 'bolsa', en inglés *bagplot*, es una generalización bivariable del diagrama de caja y bigotes. Fue propuesto por Rousseeuw, Ruts, y Tukey en 1999[cita]. En esta forma bivariable del diagrama, la caja del diagrama pasa a ser una cascara/casco convexo, la 'bolsa'.

La implementación original del diagrama de caja solo abarcamos las distribuciones univariables. El resumen de 5 números es una descripción muy útil no solo para datos de una variable, sino también para distribuciones de datos bivariables. El mayor desafio al extender el diagrama de caja para el uso con más dimensiones es como traducir el resumen de 5 números, que en el caso bivariable son valores vectoriales, en guías visuales con posiciones espaciales significativas, a la vez que se mantiene la simplicidad del diagrama original. Un diagrama de caja bivariable no solo puede mostrar la localización y un resumen de la distribución, sino también su desviación, su extensión y su correlación.

El diagrama de bolsa permite la visualización de la localización (mediante la profundidad de Tukey), la extensión (el tamaño de la bolsa), correlación (la orientación de la bolsa), la asimetría (forma de la bolsa y el bucle), los extremos (puntos cercanos a la separación del bucle con las anomalías) y los elementos atípicos del conjunto de datos.

El diagrama de bolsa utiliza el concepto de profundidad semi-espacio ("*halfspace depth*") para construir una versión bivariable del diagrama de caja. La profundidad semi-espacio $ldepth(\theta|Z)$ de un punto θ es el menor número de puntos de datos $z_i \in Z = z_1, z_2, ..., z_n$ contenidos en cualquier semi-espacio cerrado con una línea limite a través de θ. La región de profundidad D_k, que es un polígono convexo, es el conjunto de todos los θ con $ldepth(\theta|Z) > k$ y $D_{k+1} \subset D_k$.

Para construir el diagrama de bolsa primero se crea un diagrama de dispersión. La "profundidad mediana" es entonces hallada, que es el punto θ con la mayor $ldepth(\theta|Z)$, si solo hay un solo θ que cumple esta condición, si no es el centro de gravedad de la región más profunda. Este punto, que está en el centro del diagrama, es representado con una cruz. La bolsa es definida como la región con la menor profundidad de Tukey que contiene al menos $n/2$ observaciones. Se dibuja de un color más oscuro. El perímetro que no es dibujado, se obtiene inflando la bolsa por un factor (normalmente 3). Los valores fuera de este perímetro son clasificados como valores atípicos, y se representan con otro símbolo. El bucle, una envolvente convexa, contiene los datos no etiquetados como anomalías, esto es, fuera de la bolsa pero dentro del perímetro.

La profundidad de cada punto es calculada mediante el algoritmo $LDEPTH$.

La profundidad mediana del conjunto (mediana bivariable de Tukey) es obtenida mediante el algoritmo $HALFMED$. Estos dos métodos son descritos en (Rousseeuw Ruts 1998, *Constructing the Bivariate Tukey Median* [cita]). Para obtener los contornos basados en profundidad se utiliza el algoritmo $ISODEPTH$ (Rousseuw Ruts 1996, *Computing depth contours of bivariate point clouds*).[cita].[cita].

2.3.1. ISODEPTH del paquete depth

ISODEPTH crea los contornos en función a la profundidad de Tukey. Calcula los contornos basados en profundidad mediante ISODEPTH. Determina qué versión de ISODEPTH utiliza dependiendo del factor introducido.

Listing 2.5: argumentos por defecto de isodepth

```
isodepth(x, dpth = NULL, output = FALSE, twodim = TRUE,
    mustdith = FALSE, maxdith = 50, dithfactor = 10,
   trace.errors = TRUE, eps = 1e-8, factor = 0.8, xlab
   = "X", ylab = "Y", zlab = "Tukey's depth",
   colcontours = NULL, ...)
```

Argumentos ISODEPTH

`x`	Conjunto de datos de dos variables como matriz, dataframe o lista.
`dpth`	Vector de enteros positivos para dibujar contornos determinados.
`output`	Determina si el resultado es un gráfico o una lista con los contornos y sus vértices.
`twodim`	Valor lógico, determina si el resultado es un
`mustdith`	Valor lógico, determina si se aplica tramado o ruido a los datos.
`maxdith`	Número máximo de iteraciones para tramado.
`dithfactor`	Factor de escala utilizado para añadir tramado.
`trace.errors`	Valor lógico. Determina si al encontrar un problema numérico al calcular un contorno este se descarta. Por defecto FALSE, los descarta.

`eps`	Tolerancia a errores para controlar el calculo.
`factor`	Proporción (de 0 a 1) de los contornos más exteriores calculados mediante la versión ISODEPTH de 1998; el resto son calculados mdeiante la versión de 1999.
`xlab`	Titulo para el eje x.
`ylab`	Titulo para el eje y.
`zlab`	Titulo para el eje z.
`colcontours`	Vector de nombres de colores para los contornos.
`...`	Parámetros gráficos adicionales.

2.3.2. Bagplot del paquete aplpack

Listing 2.6: argumentos por defecto de bagplot

```
bagplot(x, y, factor = 3, na.rm = FALSE, approx.limit =
    300, show.outlier = TRUE, show.whiskers = TRUE,
   show.looppoints = TRUE, show.bagpoints = TRUE, show.
   loophull = TRUE, show.baghull = TRUE,  create.plot =
    TRUE, add = FALSE, pch = 16, cex = 0.4, dkmethod =
   2, precision = 1, verbose = FALSE, debug.plots = "no
   ", col.loophull="#aaccff", col.looppoints="#3355ff",
    col.baghull="#7799ff", col.bagpoints="#000088",
   transparency=FALSE, ...)
```

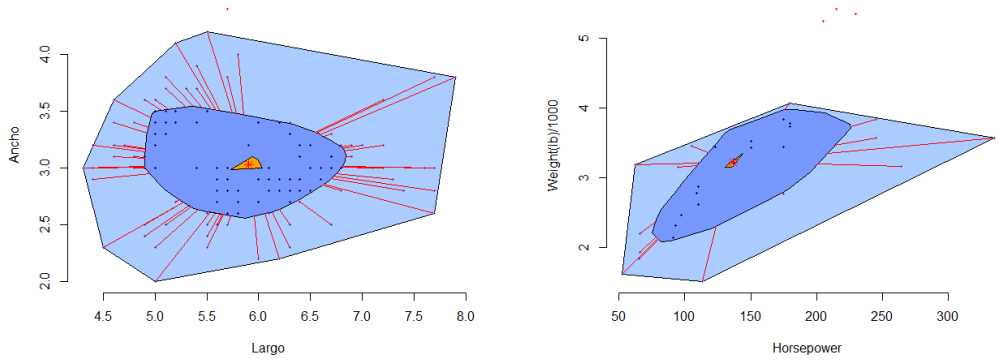
Argumentos bagplot

`x`	los valores x de un conjunto de datos.
`y`	valores y del conjunto de datos.
`factor`	factor que define el blucle
`na.rm`	valor lógico, que determina si los valores nulos son eliminados o sustituidos por la mediana.
`approx.limit`	si el número de valores en el conjunto exceed este valor, se utiliza una muestra para computar algunos valores del diagrama.

`show.outlier`	Muestra o no los outliers.
`show.whiskers`	Muestra o no los bigotes
`show.looppoints`	Muestra o no los puntos del bucle.
`show.bagpoints`	Muestra o no los puntos dentro de la bolsa
`show.loophull`	Muestra o no el bucle.
`show.baghull`	Muestra o no la bolsa.
`dkmethod`	Elige el método de aproximación (1 ó 2) para calcular la bolsa: el método 1 es muy rudimentario, solo basado en observaciones
`precision`	Precisión de la aproximación, 1 por defecto.
`verbose`	Mostrar comentarios de los cálculos.
`debug.plots`	Muestra o no diagramas adicionales explicativos de los resultados adicionales.
`...`	Argumentos gráficos adicionales.

Listing 2.7: generación de dos diagramas de cajas

```
bagplot(Sepal.Length, Sepal.Width)
bagplot(mtcars[,4], mtcars[,6], xlab="Horsepower", ylab="
    Weight(lb)/1000")
```



(a) Distribucion del ancho y largo del sépalo de 150 muestras de la flor del Iris

(b) Distribución de peso y potencia de 32 coches en 1974

Figura 2.3: Diagramas de bolsa generados por bagplot con los conjuntos iris y mtcars

2.4. Distancia de Mahalanobis

La distancia de Mahalanobis, propuesta por primera vez por P. C. Mahalanobis en 1936[cita], es una de las medidas de distancia más utilizada en estadística multivariable. Puede ser utilizada para determinar si un objeto es una anomalía, si un proceso está bajo control, o si una observación pertenece a un grupo o no.

La distancia de Mahalanobis es una medida de espacio entre un punto x y una distribuciónX. Es una generalización de la idea de medir cuantas veces la desviación típica se desvía x de la medía de X. La distancia es cero si x está en la media de X, y crece según x se aleja de la mediana: a lo largo de cada eje de componente principal, mide el número de veces la desviación típica P se aleja de la media.

Formalmente, sea $X = (x_1, ..., x_n)$ la población, con media μ y matriz de covarianza Σ. la distancia de Mahalanobis de un punto x a la media μ es:

$$d_M(x, \mu)^2 = (x - \mu)\Sigma^{-1}(x - \mu)^T \tag{2.1}$$

Propiedades de la distancia de Mahalanobis:

- d_M es escalar (o adimensional).
- d_M tiene en cuenta las diferentes variabilidades (varianzas de las variables)
- d_M tiene en cuenta las correlaciones entre las variables.
- para distribuciones normales, d_M sigue una distribución Chicuadrada χ^2 con d grados de libertad (d = dimensiones del conjunto)
- Si los datos son redimensionados para tener una varianza unitaria (estandarización), entonces d_M se corresponde con la distancia euclidiana en el espacio transformado.

Cuando sólo hay una variable, esta se simplifica a $d_M^2 = (x-\mu)v^{-1}(x-\mu)^T$, donde v es la varianza, o $d_M^2 = (x - \mu)^2/v$ resultando en $d_M = (x - \mu)/\sigma$ donde σ es la desviación típica de la muestra. De esta manera, la distancia de Mahalanobis para una variable equivale al número de veces la desviación típica una observación se aleja de la media.

Para más de una variable, la distancia de Mahalanobis puede ser visualizada como la distancia de un punto al centro del conjunto de datos, proyectado en una elipse cuyo eje principal es el de los datos.

Para la detección de anomalías, podemos rechazar todos los puntos con $d_M(x, \mu) > \chi^2(0, 975)$ (aproximadamente 3 veces la desviación típica σ). También se pueden aplicar técnicas vistas en este capítulo, como la prueba de Grubbs.

La distancia de Mahalanobis sufre la *maldición de la dimensión*, y cuanto mayor grado de libertad más similares son los valores d_M para todos los puntos. Otra desventaja es que es una medida poco robusta, ya que la media y la desviación típica, utilizadas en su calculo son sensibles a las anomalías.

2.4.1. mahalanobis{stats}

La función mahalanobis del paquete stats calcula la distancia a un conjunto de datos utilizando un vector de medias y la matriz covarianza.

Listing 2.8: argumentos por defecto de mahalanobis

```
mahalanobis(x, center, cov, inverted = FALSE, ...)
```

Devuelve el cuadrado de la distancia de Mahalanobis de las filas en `x` y el vector `mu = center` con respecto a `sigma=cov`. Para el vector `x` esto se define como:

$$d^2(x, \mu) := (x - \mu)^T \Sigma^{-1} (x - \mu) \tag{2.2}$$

Argumentos bagplot

`x`	vector o matriz de datos con p columnas.
`center`	μ, vector de medias de la distribución o segundo vector de datos de longitud p.
`cov`	Σ, matriz covarianza de dimensiones $p * p$ de la distribución.
`inverted`	valor lógico, índica si la matriz provista cov esta invertida.

2.4.2. aq.plot chisq.plot {mvoutlier}

2.5. Prueba C de Cochran

La prueba C de Cochran, prueba introducida por William G.Cochran en 1941 [cita], es una prueba de unilateral de límite de superior para varianzas atípicas. La prueba C es utilizada para evaluar si una estimación de varianza (o desviación típica) es relativamente mayor que un grupo de varianzas al que esta estimación supuestamente pertenece.

La prueba C de Cochran asume equilibrio en el conjunto, es decir, el conjunto de datos entero debe estar formado de muestras o series de datos con el mismo número de observaciones. Otra asunción de la prueba C es que cada muestra de datos sigue una distribución normal. çLa prueba C contrasta la hipótesis nula H_0 con la hipótesis alternativa H_a:

- H_0: Todas las varianzas son iguales.
- H_a: Al menos una varianza es significativamente mayor que las demás

La prueba de Cochran detecta una varianza excepcionalmente grande cada vez que se aplica. Tras esto, la muestra correspondiente es omitida de la colección completa. De acuerdo con el estándar ISO 5725, la prueba C puede ser realizada hasta que no se detecten varianzas anómalas en la colección, pero al hacerlo se puede correr el riesgo de llevar a cabo demasiados rechazos si las series de datos no siguen una distribución normal.

La estimación para la prueba es la relación entre la varianza máxima del conjunto de colecciones y la suma de todas las varianzas:

$$C_j = \frac{S_j^2}{\sum_{i=1}^{N} S_i^2} \tag{2.3}$$

donde:

C_j = estadística C de Cochran para la colección j
$S2j$ = varianza de la colección j
N= numero de colecciones restantes en el conjunto (N disminuye 1 a cada iteración de la prueba C)
Si = varianza de la colección i, $i \in [1, N]$

La varianza de una serie de datos j es considerada un valor anómalo en un nivel de confianza α si C_j sobrepasa el límite superior crítico CUL. CUL depende del nivel de confianza α, el número de series de datos N, el número de observaciones n por cada serie de datos, y el valor crítico del ratio *F de Fisher* F_c. Valores para este límite superior han sido tabulados para distintos niveles de confianza, y también se puede obtener mediante la

siguiente formula:

$$C_{\mathrm{UL}}(\alpha, n, N) = \left[1 + \frac{N-1}{F_{\mathrm{c}}(\alpha/N, (n-1), (N-1)(n-1))}\right]^{-1} \tag{2.4}$$

Donde:

C_{UL} = límite del valor critico superior para una prueba unilateral en un conjunto equilibrado.
α = nivel de confianza
n = número de datos por cada colección en el conjunto
F_c = valor crítico de la relacion F de Fisher (puede ser obtenido de tablas de la distribución F o calculando su valor)

2.5.1. cochran.test {outliers}

Esta función, también presente en el paquete outliers [cita], realiza la prueba C de Cochran para evaluar si la varianza mayor en distintos grupos de datos es anómala y este grupo debe ser rechazado. También ofrece la posibilidad de, si un grupo tiene una varianza muy pequeña, poder evaluarla como anómalamente pequeña.

Listing 2.9: argumentos por defecto de cochran.test

```
cochran.test(object, data, inlying = FALSE)
```

La función devuelve una lista de clase `htest` con el resultado de la prueba. Si la variable `p.value` es menor que el nivel de confianza elegido, se puede rechazar la hipótesis nula y evaluar la varianza como anómala.

Argumentos

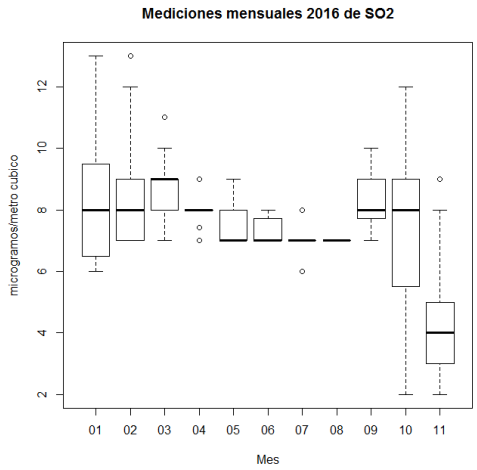
`object`	un vector con las varianzas o una formula.
`data`	Si `object` es un vector, data debe ser otro vector con la cardinalidad de cada conjunto correspondiente. Si es una formula, debe ser un objeto `dataframe` con las observaciones.
`inlying`	Valor lógico. Evalúa la varianza menor como anómalamente pequeña

Ejemplos

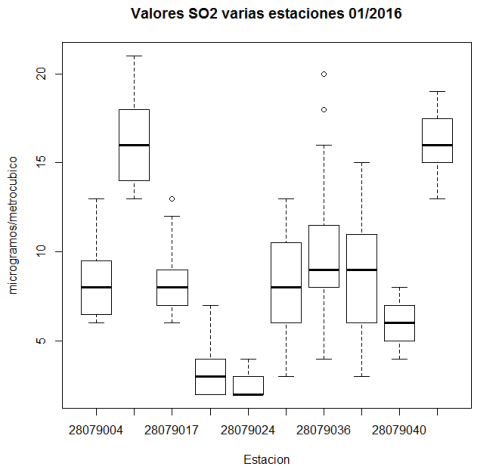
Ejemplo 1. En este ejemplo los datos obtenidos por una estación de medición de la contaminación durante el año 2016[cita] serán evaluados:

```
> 2016SO2$value[is.na(2016SO2$value)] <- mean(2016SO2$value, na.rm = TRUE)
> cochran.test(value~Mes, 2016SO2)

Cochran test for outlying variance

data:  value ~ Mes
C = 0.35479, df = 31, k = 11, p-value = 1.243e-14
alternative hypothesis: Group 10 has outlying variance
sample estimates:
01          02          03          04          05          06          07
4.10322581 2.91715661 1.11182796 0.40851495 0.36989247 0.19644627 0.09892473
08          09          10          11
0.0      0.63769165 7.57849462 3.93834970

> cochran.test(value~Mes,testa, inlying = TRUE)

Cochran test for inlying variance

data:  value ~ Mes
C = 0, df = 31, k = 11, p-value < 2.2e-16
alternative hypothesis: Group 08 has inlying variance
sample estimates:
01          02          03          04          05          06
4.10322581 2.91715661 1.11182796 0.40851495 0.36989247 0.19644627
07          08          09          10          11
0.09892473 0.00000000 0.63769165 7.57849462 3.93834970
> testa <- melt(subset(madrid2016, (Magnitud == '01' & Estacion == '28079004
    '), select = Mes:D31))
> boxplot(data =testa, value ~ Mes, main="Mediciones mensuales 2016 de SO2")
```

De esta manera, se puede detectar si en un mes las mediciones obtenidas son notablemente distan notoriamente más unas de otras que las de otros meses, y, con la opción `inlying`, si existe poca variación entre ellas. Nótese que, antes de realizar la prueba y debido al distinto número de días de los meses, los valores no presentes han sido reemplazados por la media mensual. De esta manera obtenemos que el mes de octubre tiene una varianza aberrante, mientras que la varianza del mes de agosto es evaluada como excepcionalmente pequeña (si consultamos los datos son 31 valores iguales). [enlace]



(a) Medidas de dióxido de azufre durante el año 2016



(b) Medidas de dióxido de azufre en el mes de enero de 2016

Figura 2.4: Diagramas de caja

Ejemplo 2. En este otro ejemplo los datos obtenidos de distintas estaciones durante enero de 2016 [cita] serán evaluados:

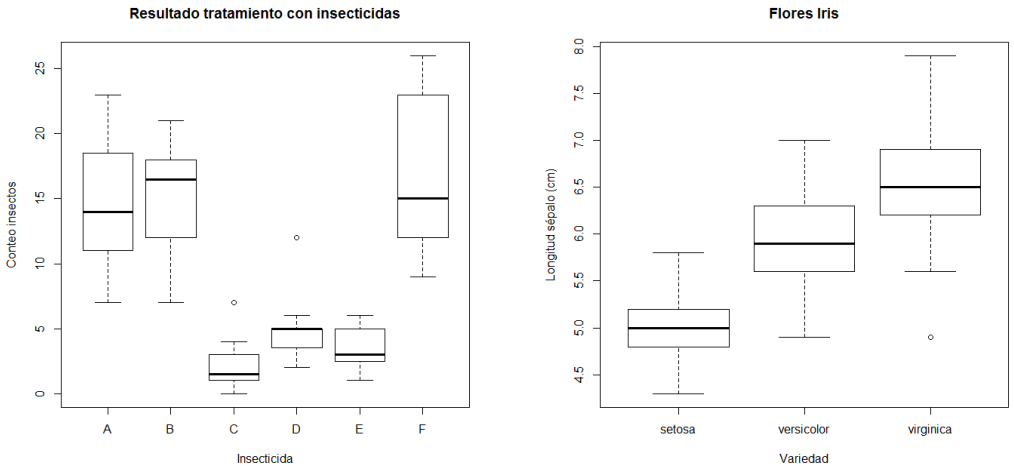
```
> SO2Estaciones <- subset(madrid2016, Magnitud == '01')
> SO2Estaciones1 <- subset(SO2Estaciones, Mes == '01', select = c(Estacion,
    D1:D31))
> testSO2E1<- melt(SO2Estaciones1, id.vars = "Estacion")
> cochran.test(value~Estacion,testSO2E1)

Cochran test for outlying variance

data: value ~ Estacion
C = 0.26268, df = 31, k = 10, p-value = 2.285e-06
alternative hypothesis: Group 28079036 has outlying variance
sample estimates:
28079004  28079008  28079017  28079018  28079024  28079035  28079036
4.1032258 5.3784946 3.3118280 1.7913978 0.3118280 8.6064516 13.1397849
28079038  28079040  28079057
9.0258065 0.9569892 3.3956989

> cochran.test(value~Estacion,testSO2E1, inlying = TRUE)

Cochran test for inlying variance

data: value ~ Estacion
C = 0.0062339, df = 31, k = 10, p-value < 2.2e-16
alternative hypothesis: Group 28079024 has inlying variance
sample estimates:
28079004  28079008  28079017  28079018  28079024  28079035  28079036
4.1032258 5.3784946 3.3118280 1.7913978 0.3118280 8.6064516 13.1397849
28079038  28079040  28079057
9.0258065 0.9569892 3.3956989

> boxplot(data = testSO2E1, value ~ Estacion, main="valores␣SO2␣varias␣
    estaciones␣01/2016", xlab = "Estacion", ylab="microgramos/metrocubico")
```

Mediante esta prueba, se puede detectar si las mediciones obtenidas por una estación distan notoriamente más unas de otras que las de otras estaciones, y, con la opción `inlying`, si existe poca variación entre ellas. Así, obtenemos que los datos obtenidos por la estación 28079036 tienen una varianza excepcionalmente grande, y aquellas mediciones obtenidas por la estación 28079024 tienen una varianza excepcionalmente pequeña. [enlace]

Ejemplo 3. La hipótesis nula será contrastada con los datos de insectos tratados con diferentes aerosoles insecticidas [cita]:

```
> cochran.test(count~spray,InsectSprays)

Cochran test for outlying variance

data:  count ~ spray
C = 0.41832, df = 12, k = 6, p-value = 0.004435
alternative hypothesis: Group F has outlying variance
sample estimates:
A         B         C         D         E         F
22.272727 18.242424  3.901515  6.265152  3.000000 38.606061
> boxplot(data=InsectSprays, count~spray, main="Resultado tratamiento con
    insecticidas", ylab="Conteo insectos",xlab="Insecticida")
```

El resultado obtenido rechaza la hipótesis nula clasificando el grupo F con una varianza excepcionalmente grande. Esto significa que con tal insecticida los resultados del conteo de insectos distan mucho comparados con el resto de los resultados. [enlace]



(a) Recuento de insectos por aerosol insecticida

(b) Diagrama de caja de la longitud del sépalo por especie

Figura 2.5

Ejemplo 4. En este ejemplo las varianzas de la longitud del sépalo según la especie de flores Iris [cita] son comparadas:

```
> cochran.test(Sepal.Length~Species,iris)

Cochran test for outlying variance

data:  Sepal.Length ~ Species
C = 0.50859, df = 50, k = 3, p-value = 0.003456
alternative hypothesis: Group virginica has outlying variance
sample estimates:
setosa versicolor  virginica
0.1242490  0.2664327  0.4043429
```

Muestra que las medidas de longitud del sépalo de las flores Iris de la variedad Virginica tienen una varianza excepcionalmente grande comparada con los otros dos grupos. [enlace]

2.5.2. Referencias

W.G. Cochran, The distribution of the largest of a set of estimated variances as a fraction of their total, Annals of Human Genetics (London) 11(1), 47–52 (January 1941). Using the Grubbs and Cochran tests to identify outliers, Analytical Methods Committee, AMCTB No. 69 Cochran's C Test Statistic – What, Why and How https://consultglp.com/2015/07/11/cochrans-c-test-for-outliers/

2.6. Prueba Chi-cuadrado

La prueba chi-cuadrado (o prueba χ^2), introducida por W.J. Dixon en 1950 [cita], está basada en las diferencias entre los objetos pertenecientes a un conjunto de datos y los correspondientes objetos de una supuesta distribución de probabilidad, en este caso, la distribución normal. Es una prueba bastante obsoleta, y las pruebas descritas por Dixon en la misma publicación son más potentes. La prueba chi-cuadrado contrasta la hipótesis nula H_0 con la hipótesis alternativa H_a:

- H_0: no hay anomalías en el conjunto
- H_a: el valor que más se aleja de la media es una anomalía

Requiere conocimiento del valor de la varianza (o una estimación de σ disponible) para calcularlo, según la formula en la publicación original:

$$\chi^2 = \frac{\sum(x-\mu)}{\sigma^2} \tag{2.5}$$

La prueba ofrecerá una indicación de una gran dispersión, y, ya que los valores extremos son los principales contribuidores a la suma de los cuadrados, es posible utilizar esta prueba como criterio para interpretar el valor o valores que están a mayor distancia de la media aritmética. Por tanto, χ^2 podrá ser utilizada, si el valor χ^2 es demasiado grande (mayor que un límite superior dado para la distribución χ^2), para clasificar el valor más distante de la media aritmética μ como outlier. Para los limites superiores, se podrá utilizar tablas de distribución χ^2. El nivel de confianza α recomendado es del $0,05$.

2.6.1. chisq.out.test {outliers}

Esta función, en el paquete `outliers` [cita], realiza una prueba chicuadrado para detectar un valor anómalo en un vector.

Listing 2.10: llamada por defecto a la función

```
chisq.out.test(x, variance=var(x), opposite = FALSE)
```

La función devuelve una lista de clase `htest` con el resultado de la prueba. Si la variable `p.value` es menor que el nivel de confianza elegido, se puede rechazar la hipótesis nula y evaluar el objeto como anómalo.

Argumentos

`x`	Un vector numérico para que contiene las observaciones a analizar.

`variance` La varianza conocida de la población o una estimación. Si no es especificada, una estimación de la muestra es utilizada.

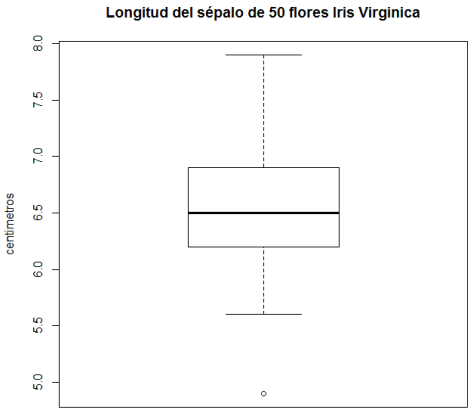
`opposite` Valor lógico, elige si el valor a evaluar no es aquel con mayor diferencia a la media, sino el opuesto (más bajo, si el más sospechoso es el mayor etc.)

Ejemplos

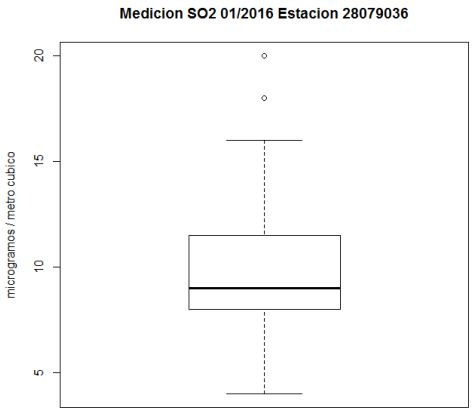
Ejemplo 1. Se probara si alguna de las observaciones extremas sobre la longitud del sépalo de las flores Iris Virginica [cita] puede ser considerada como anomalías:

```
> virginicaSepalo = sort(as.vector(t(subset(iris, select = Sepal.Length,
    Species == "virginica"))))
> chisq.out.test(virginicaSepalo)


chi-squared test for outlier

data:  virginicaSepalo
X-squared = 7.0469, p-value = 0.00794
alternative hypothesis: lowest value 4.9 is an outlier

> chisq.out.test(virginicaSepalo, opposite=TRUE)

chi-squared test for outlier

data:  virginicaSepalo
X-squared = 4.2571, p-value = 0.03909
alternative hypothesis: highest value 7.9 is an outlier

> chisq.out.test(virginicaSepalo [2:50])

chi-squared test for outlier

data:  virginicaSepalo[2:50]
X-squared = 4.6342, p-value = 0.03134
alternative hypothesis: highest value 7.9 is an outlier

##Generamos diagramas de caja
> boxplot(virginicaSepalo, main = "Longitud del sépalo de 50 flores Iris
    Virginica", ylab = "centimetros")
```

La prueba rechaza la observación menor de $4,9cm$ como una anomalía con un resultado muy por debajo del intervalo de confianza. También se puede rechazar la observación mayor de $7,9cm$ como valor aberrante, pero el resultado no se aleja mucho del límite marcado por el valor crítico. Para comprobar si la anomalía superior está afectando el resultado de la prueba para esta anomalía inferior se excluye de la prueba, y, de esta manera, la prueba rechaza la hipótesis nula con mayor confianza: esto se debe a que al eliminar la primera anomalía, la desviación típica disminuye y esta observación está más alejada (distancia se mide en veces la desviación típica).



(a) Diagrama de las longitudes del sépalo muestra la anomalía rechazada por la prueba



(b) Diagrama de las medidas de dioxido de azufre muestra 2 anomalías

Figura 2.6

Ejemplo 2. En este ejemplo se evaluara si las mediciones de dióxido de azufre más extremas de una estación de medida de contaminación del municipio de Madrid capital, durante el mes de enero de 2016, son anomalías [cita/enlace]:

```
> boxplot(E28079036$value, main= "Medicion SO2 01/2016 Estacion 28079036",
    ylab ="microgramos / metro cubico")
> chisq.out.test(E28079036$value)

chi-squared test for outlier

data:  E28079036$value
X-squared = 7.367, p-value = 0.006643
alternative hypothesis: highest value 20 is an outlier

>  chisq.out.test(E28079036$value, opposite=TRUE)

chi-squared test for outlier

data:  E28079036$value
X-squared = 2.8891, p-value = 0.08918
alternative hypothesis: lowest value 4 is an outlier
```

El resultado refleja que la medición mayor de $20\,\mu g$ correspondiente al día viernes 29 de enero es una anomalía. Por el contrario la medida más baja de $4\,\mu g$ no es rechazada. Si se genera un diagrama de caja del conjunto[enlace], se puede observar dos anomalías, y al eliminar la primera, el diagrama refleja de nuevo dos. Realizar el test de nuevo no es recomendable en este caso ya que se puede rechazar demasiadas observaciones: la prueba rechazaría 14

observaciones si se sigue repitiendo.

Ejemplo 3. En este ejemplo se contrastaran las hipótesis con los datos de presión sanguínea de 270 pacientes [cita]:

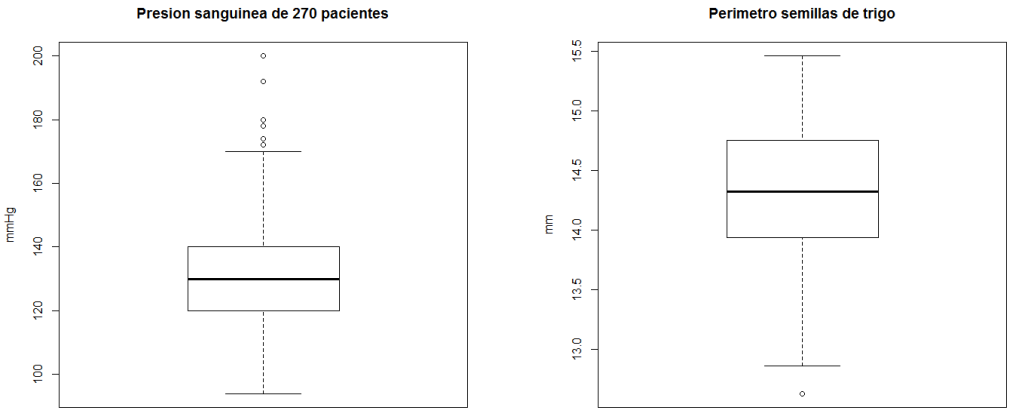
```
> chisq.out.test(heart$press)

chi-squared test for outlier

data:  heart$press
X-squared = 14.774, p-value = 0.0001212
alternative hypothesis: highest value 200 is an outlier

> chisq.out.test(heart$press, opposite=TRUE)

chi-squared test for outlier

data:  heart$press
X-squared = 4.3713, p-value = 0.03655
alternative hypothesis: lowest value 94 is an outlier

> boxplot(heart$press, main="Presion sanguinea de 270 pacientes", ylab="mmHg
    ")
> hist(heart1,breaks=21,main="Presion sanguinea de 270 pacientes", xlab="
    mmHg")
```

Se rechazan las observaciones más extremas. Nótese que un diagrama de caja no reflejara el valor mínimo como anomalía [enlace], y es difícil considerarlo como anomalía al observar el histograma de los datos, mientras que las dos observaciones extremas son rechazadas por un diagrama.

Ejemplo 4. La prueba chicuadrado se utilizar para hallar una posible anomalía entre las mediciones del perimetro de una variedad de semillas de trigo.

```
> boxplot(tr.kama$perimeter,main="Trigo variedad Kama",ylab="Perimetro (mm)"
    )
> chisq.out.test(tr.kama$perimeter)

chi-squared test for outlier

data:  tr.kama$perimeter
X-squared = 8.3317, p-value = 0.003896
alternative hypothesis: lowest value 12.63 is an outlier
```

La prueba rechaza la hipótesis nula y valora el valor menor como anómalo, al igual que se muestra en el diagrama de caja[enlace]



(a) El diagrama de caja solo muestra las observaciones superiores como anomalías

(b) Diagrama muestra como anomalía la observación menor

Figura 2.7

Ejemplo 5. Para este ejemplo se utilizaran observaciones del semi-diametro vertical de Venus hechas en 1846, al igual que en las publicaciones de Grubbs[cita] y de Tietjen-Moore[cita]:

```
> chisq.out.test(Venus)

chi-squared test for outlier

data:  Venus
X-squared = 6.6241, p-value = 0.01006
alternative hypothesis: lowest value -1.4 is an outlier

> chisq.out.test(Venus, opposite = TRUE)

chi-squared test for outlier

data:  Venus
X-squared = 3.2419, p-value = 0.07178
alternative hypothesis: highest value 1.01 is an outlier

> chisq.out.test(Venus[2:15])

chi-squared test for outlier

data:  Venus[2:15]
X-squared = 4.9224, p-value = 0.02651
alternative hypothesis: highest value 1.01 is an outlier
```

En este caso, sólo la observación menor es considerada como valor aberrante en un principio, y la más grande no puede ser rechazada con un 5 % de nivel de confianza. Si la prueba es realizada de nuevo excluyendo la primera

medida, entonces la observación mayor es considerada una anomalía. Esto es efecto del enmascaramiento: la presencia de la primera anomalía hace que la segunda no resalte tanto sobre las demás y que sea considerada una observación corriente.

2.6.2. Referencias

Dixon, W.J. (1950). Analysis of extreme values. Ann. Math. Stat. 21, 4, 488-506.

2.7. Prueba de Dixon

Esta prueba, al igual que la anterior, fue propuesta por Dixon (de ahí su nombre) en 1950[cita]. Esta prueba, utilizada para la identificación y rechazo de anomalías, asume que los datos a evaluar siguen una distribución normal y no requiere conocimiento previo del valor de la varianza σ.

Las pruebas diseñadas por Dixon están basadas en la distribución estadística de "ratios de rangos y subrangos" de muestras de datos ordenadas, tomadas de la misma población normal. Por esto es que se asume una distribución normal (de Gauss) al aplicar estas pruebas. En caso de detección y rechazo de un valor anómalo, las pruebas no pueden volver a llevarse a cabo en el conjunto de datos restante.

La prueba Dixon contrasta la hipótesis nula H_0 con la hipótesis alternativa H_a:

- H_0: no hay anomalías en el conjunto
- H_a: existe al menos una anomalía en el conjunto

Dependiendo de qué valor se quiere evaluar y de qué manera, se proponen seis formulas diferentes para cada prueba:

1. Para una única anomalía x_1

   $r_{10} = \frac{x_2 - x_1}{x_n - x_1} \quad \left(ó \quad \frac{x_n - x_{n-1}}{x_n - x_1}\right)$

2. Para una única anomalía x_1 no teniendo en cuenta x_n

   $r_{11} = \frac{x_2 - x_1}{x_{n-1} - x_1} \quad \left(ó \quad \frac{x_n - x_{n-1}}{x_n - x_2}\right)$

3. Para una única anomalía x_1 no teniendo en cuenta x_n, x_{n-1}

   $r_{12} = \frac{x_2 - x_1}{x_{n-2} - x_1} \quad \left(ó \quad \frac{x_n - x_{n-1}}{x_n - x_3}\right)$

4. Para una única anomalía x_1 no teniendo en cuenta x_2

   $r_{20} = \frac{x_3 - x_1}{x_n - x_1} \quad \left(ó \quad \frac{x_n - x_{n-2}}{x_n - x_1}\right)$

5. Para una única anomalía x_1 no teniendo en cuenta x_2 ni x_n

   $r_{21} = \frac{x_3 - x_1}{x_{n-1} - x_1} \quad \left(ó \quad \frac{x_n - x_{n-2}}{x_n - x_2}\right)$

6. Para una única anomalía x_1 no teniendo en cuenta x_2 ni x_n, x_{n-1}

   $r_{22} = \frac{x_3 - x_1}{x_{n-2} - x_1} \quad \left(ó \quad \frac{x_n - x_{n-2}}{x_n - x_3}\right)$

Una vez elegida la fórmula (se discutirá el criterio más adelante) para evaluar el objeto y obtenido el resultado, en el siguiente paso compararemos el resultado obtenido R con el valor crítico de la tabla r_{ij} según el intervalo de confianza α con el que se quiera realizar la prueba. Las tablas de valores críticos para los intervalos de confianza fueron determinados en una publicación posterior por Dixon [cita ratios].

Si el valor R calculado para una observación en particular es más grande que el valor crítico de la tabla r_{ij} podemos considerar que este objeto evaluado es una anomalía según la prueba Dixon.

En la notación de Dixon, el primer dígito del subíndice de cada ratio, r_{ij}, se refiere al número posible de valores anómalos sospechados en el mismo extremo del dato siendo evaluado, mientras el segundo dígito indica el número de posibles anomalías en el extremo opuesto de los datos del objeto evaluado. Así, el ratio r_{10} compara la diferencia de una única supuesta anomalía x_1 o x_n y su vecino más cercano entre el rango de valores de toda la muestra, es decir, determina la fracción del rango total que puede ser atribuido al supuesto valor aberrante. Los otros ratios son formulados de manera similar excepto que usan subrangos diseñados para evitar la influencia de anomalías adicionales en el extremo opuesto (r_{11} y r_{12}), en el mismo extremo (r_{20}), o en ambos (r_{21}, r_{22}).

Estos últimos ratios requieren una muestra de datos lo suficientemente grande para rendir adecuadamente. La recomendación de Dixon (basada en una combinación del rendimiento de cada ratio y su grado de independencia respecto a otros valores anómalos) es que, como regla general, los distintos ratios sean usados de esta manera: para $3 \leq n \leq 7$ se utiliza r_{10}, para $8 \leq n \leq 10$ se utiliza r_{11}, para $11 \leq n \leq 13$ se utiliza r_{21}, y para $n \geq 14$ se utiliza r_{22}.

El ratio r_{10} es el comúnmente designado Q y es, generalmente, considerado como la prueba estadística más adecuada y "legítima" para la evaluación y rechazo de valores anómalos en una muestra pequeña de distribución gaussiana (es igualmente adecuada para conjuntos de datos grandes si sólo hay un valor anómalo en la muestra).

La simplicidad de la prueba, en combinación con que los conjuntos de datos pequeños son comunes en procedimientos analíticos de control, explica el hecho de que la prueba Q sea incluida en la mayoría de tratados estadísticos y libros de texto diseñados para el uso en química analítica.

2.7.1. dixon.test {outliers}

Este método, presente en el paquete outliers [cita], realiza las distintas variantes de la prueba de Dixon para detectar un valor aberrante. La cardi-

nalidad de la muestre debe estar entre 3 y 30.

Listing 2.11: llamada por defecto

```
dixon.test(x, type = 0, opposite = FALSE, two.sided =
  TRUE)
```

La función devuelve una lista de clase `htest` con el resultado de la prueba. Si la variable `p.value` es menor que el nivel de confianza elegido, se puede rechazar la hipótesis nula y evaluar el objeto como anómalo.

Argumentos

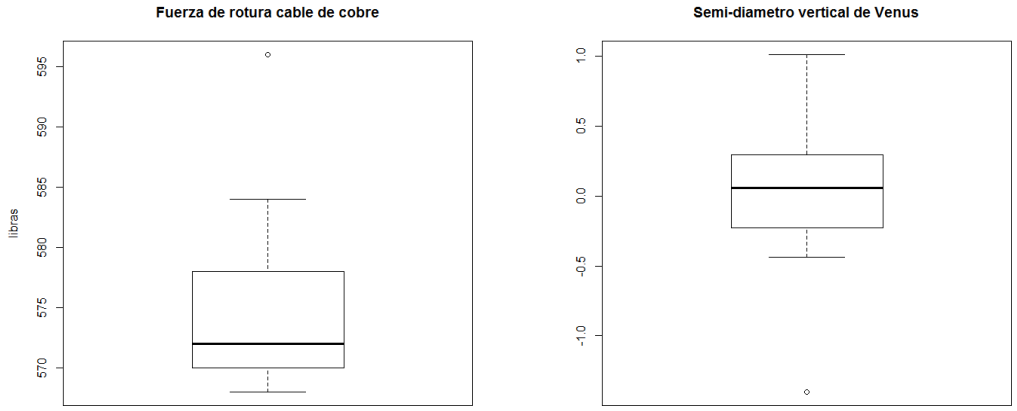
`x`	Un vector numérico para que contiene las observaciones a analizar.
`opposite`	Valor lógico, elige si el valor a evaluar no es aquel con mayor diferencia a la media, sino el opuesto (más bajo, si el más sospechoso es el mayor etc.)
`type`	Valor entero que especifica que variante de la prueba realizar. Los posibles valores, de acuerdo con aquellos dados por Dixon en 1950 [cita]: 10, 11, 12, 20, 21. Si este valor no es especificado, una variante de la prueba es elegida automáticamente de acuerdo al tamaño del vector (10 para 3-7, 11 para 8-10, 21 para 11-13, 22 para 14 y mayores). El valor mayor o menor es elegido automáticamente, y puede revertirse con el parámetro `opposite`.
`two.sided`	Valor lógico. Trata esta prueba como una prueba bilateral o unilateral.

Ejemplos

Ejemplo 1. Evaluaremos los siguientes datos que muestran la fuerza de rotura (en libras) de un cable de cobre [cita][Grubbs 1969]. En este ejemplo se utiliza la prueba de Dixon, basada simplemente en la relación entre los rangos, para evitar el cálculo de la varianza:

```
> cable = c(568,570,570,570,572,572,572,578,584,596)
> boxplot(cable,ylab="libras", main="Fuerza de rotura cable de cobre")
> dixon.test(cable)

Dixon test for outliers

data:  cable
Q = 0.46154, p-value = 0.1185
alternative hypothesis: highest value 596 is an outlier

> dixon.test(cable, type=10)

Dixon test for outliers

data:  cable
Q = 0.42857, p-value = 0.08148
alternative hypothesis: highest value 596 is an outlier
```

De esta manera, la prueba de Dixon determina que no se puede rechazar el valor 596 como una anomalía. Nótese que automáticamente, el método utiliza la formula r_{11} para evaluar el valor sospechoso. Si se utiliza el método r_{10} tampoco se rechaza la hipótesis nula aunque el *p-valor* disminuya. Esto se debe a que el método r_{10} tiene en cuenta todos los elementos, mientras que el r_{11} no toma en consideración el otro extremo del valor evaluado (en este caso 568).



(a) Diagrama de caja muestra como anomalía la observación superior

(b) Diagrama de caja muestra la observación menor como anomalía también

Figura 2.8

Ejemplo 2. Para este ejemplo se utilizaran observaciones del semi-diametro vertical de Venus hechas en 1846, al igual que en las publicaciones de Grubbs[cita] y de Tietjen-Moore[cita]:

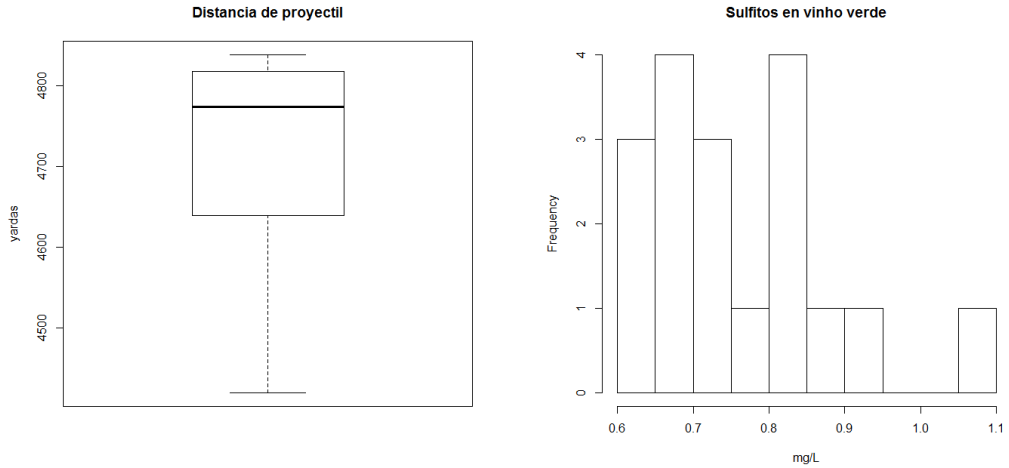
```
> Venus = c(-1.4, -0.24, -0.05, 0.18,0.48, -0.44, -0.22, 0.06, 0.2, 0.63,
    -0.3, -0.13, 0.1, 0.39, 1.01)
> boxplot(Venus, main="Semi-diametro vertical de Venus")
> dixon.test(Venus)
Dixon test for outliers

data: Venus
Q = 0.58511, p-value = 0.03515
alternative hypothesis: lowest value -1.4 is an outlier
> dixon.test(Venus, type = 11)
Dixon test for outliers

data: Venus
Q = 0.47291, p-value = 0.0253
alternative hypothesis: lowest value -1.4 is an outlier
> dixon.test(Venus, type = 12)
Dixon test for outliers

data: Venus
Q = 0.51064, p-value = 0.02497
alternative hypothesis: lowest value -1.4 is an outlier

> dixon.test(Venus, opposite = TRUE, type = 11)
Dixon test for outliers

data: Venus
Q = 0.26207, p-value = 0.3695
alternative hypothesis: highest value 1.01 is an outlier
```

Por defecto, elegido por el tamaño de la muestra, el método utiliza la formula r_{22}, y clasifica el valor menor como aberrante. Si elegimos la formula r_{11} o r_{12}, obtenemos una probabilidad menor ya que incluyen el segundo valor más extremo, y, para r_{11}, sólo un valor del extremo opuesto o, para r_{12}, los dos valores más lejanos del extremo opuesto. Si evaluamos el valor más grande, excluyendo el más pequeño, no rechazamos la hipótesis nula.

Ejemplo 3. En este ejemplo se analizara las distancias horizontales, en yardas, que alcanza el proyectil de un arma en distintas detonaciones con un mismo ángulo de elevación y una misma cantidad de pólvora, analizadas por Grubbs en 1969[cita]:

```
> dixon.test(proyectil, type=20)
Dixon test for outliers

data: proyectil
Q = 0.74163, p-value = 0.01105
alternative hypothesis: lowest value 4420 is an outlier

> dixon.test(proyectil2, type=10)
Dixon test for outliers

data: proyectil2
Q = 0.6263, p-value = 0.02333
alternative hypothesis: lowest value 4549 is an outlier

> boxplot(proyectil, main="Distancia de proyectil", ylab="yardas")
```

Si se observa los datos en un histograma, es evidente la distancia de las dos observaciones menores al resto. Es por eso que en este caso la fórmula más adecuada es la r_{20}. Esta fórmula evalúa la primera observación menor sin incluir la segunda menor en el cálculo. De esta manera rechazamos la observación menor como valor aberrante. Si se vuelve a realizar el test sin el primer valor aberrante, podemos rechazar de nuevo el valor menor del conjunto, esta vez mediante la fórmula r_{10} ya que el resto de datos deben ser incluidos en la prueba.



(a) Diagrama de las distancias no muestra la observación menor como anomalía

(b) Histograma muestra la observación mayor alejada del resto de datos

Figura 2.9

Ejemplo 4. Probaremos la existencia de anomalías en un conjunto de vinos [cita] respecto a la cantidad de sulfitos que contienen:

```
> dixon.test(subset(winequality_red,quality==8)$sulphates)

Dixon test for outliers

data:  subset(winequality_red, quality == 8)$sulphates
Q = 0.53333, p-value = 0.03498
alternative hypothesis: highest value 1.1 is an outlier

> dixon.test(subset(winequality_red,quality==8)$sulphates, type = 10)

Dixon test for outliers

data:  subset(winequality_red, quality == 8)$sulphates
Q = 0.38298, p-value = 0.03191
alternative hypothesis: highest value 1.1 is an outlier

> hist(subset(winequality_red, quality==8)$sulphates, main= "Sulfitos en
    vinho verde", xlab="mg/L")
```

Si no se especifica el tipo de test, el método elige el r_{22}, que en este caso no es el adecuado porque sólo sospechamos sobre el valor más grande, por eso debemos especificar la formula r_{10}: así incluimos todas las observaciones en la prueba y obtenemos un resultado con el que rechazar la hipótesis nula con mayor confianza.

2.7.2. Referencias

- Dixon, W.J. (1950). Analysis of extreme values. Ann. Math. Stat. 21, 4, 488-506.
- Dixon, W.J. (1951). Ratios involving extreme values. Ann. Math. Stat. 22, 1, 68-78.
- Rorabacher, D.B. (1991). Statistical Treatment for Rejection of Deviant Values: Critical Values of Dixon Q Parameter and Related Subrange Ratios at the 95 percent Confidence Level. Anal. Chem. 83, 2, 139-146.
- Dixon's Q test, Wikipedia, 2017, en.wikipedia.org/wiki/Dixon's_Q_test
- Dixon's Q test for outlier identification, Sebastian Raschka, 2017, sebastianraschka.com

2.8. Prueba de Grubbs

La prueba Grubbs, llamada así por Frank E. Grubbs, que publicó esta en 1950, es una prueba estadística para la detección de anomalías en un conjunto de datos de una única variable. Es también conocida como prueba de desviación extrema studentizada o prueba de residuos normalizados.

Esta prueba está basada en la suposición de que el conjunto de datos sigue una distribución normal. Detecta un valor anómalo cada vez que es llevada a cabo. Rechaza este valor anómalo y es repetido hasta que no se detectan más anomalías. Sin embargo, múltiples repeticiones puede cambiar la probabilidad de detección, y la prueba no debe realizarse con muestras de cardinalidad menor o igual a seis debido a que puede evaluar la mayoría de puntos como excepcionales. La prueba de Grubb contrasta la hipótesis nula H_0 con la hipótesis alternativa H_a:

- H_0: no hay anomalías en el conjunto
- H_a: existe al menos una anomalía en el conjunto

Definición: Sea un conjunto de datos de n observaciones, ordenado de menor a mayor tal que $x_1 < x_2 < ... < x_n$. Sea xn el valor dudoso de pertenencia, esto es, el valor más grande. El criterio para evaluar su importancia en el conjunto es

$$\frac{S_n^2}{S^2} = \frac{\sum_{i=1}^{n-1}(x_i - \overline{x}_n)^2}{\sum_{i=1}^{n}(x_i - \overline{x}_n)^2} \tag{2.6}$$

o, de manera más simple:

$$T_n = (x_n - \overline{x}/S)$$

donde

$\overline{x}$ = la media aritmética de todos los valores
S = la desviación típica de la muestra (o una estimación)
$\overline{x}_n = \frac{1}{n-1}\sum_{i=1}^{n-1} x^i$, esto es, la media aritmética sin el valor mayor.

Sí el primer valor, el más pequeño, es el que quiere evaluarse, un criterio similar S_1^2/S^2 puede ser utilizado:

$$T_1 = (\overline{x} - x_1/S)$$

Para evaluar si las dos observaciones más grandes son *demasiado* grandes, se utiliza la formula:

$$\frac{S^2_{n-1,n}}{S^2} = \frac{\sum_{i=1}^{n-2}(x_i - \overline{x}_{n-1,n})^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{2.7}$$

donde $\overline{x}_{n-1,n}$ no incluye los dos valores que se evalúan. De la misma manera que la anterior, para evaluar la importancia de los dos valores más pequeños, se utiliza una formula similar $S^2_{1,2}/S^2$.

Por último, para cuantificar la significación de los valores en cada extremo de los datos, esto es, la observación mayor y la observación menor, de manera simultanea, Grubbs propuso la fórmula:

$$\frac{S^2_{1,n}}{S^2} = \frac{\sum_{i=2}^{n-1}(x_i - \overline{x}_{1,2})^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{2.8}$$

donde $\overline{x}_{1,n} = \frac{1}{n-2}\sum_{i=2}^{n-1} x_i$.

Como curiosidad, la distribución de esta formula no fue estudiada debido a que la computación de los resultados hubiera resultado muy difícil por entonces.

La prueba compara la diferencia entre el resultado dudoso y la media aritmética de todo el conjunto con la desviación típica de la muestra. La prueba de Grubb es la desviación absoluta de la media aritmética más grande en unidades de desviación típica del conjunto.

La hipótesis nula de que no existen anomalías en el conjunto es rechazada con un nivel de confianza α si

$$\frac{S^2_n}{S^2} > \frac{N-1}{\sqrt{N}}\sqrt{\frac{t^2_{\alpha/(2N),N-2}}{N-2+t^2_{\alpha/(2N),N-2}}}$$

Donde t denota el valor critico superior de la distribución t *de Student* con $n-2$ grados de libertad y un nivel de confianza α/n. Para realizar la prueba para ambos extremos, se divide entre dos el nivel de confianza α.

2.8.1. grubbs.test {outliers}

Esta función, disponible en el paquete `outliers`[cita], realiza la prueba de Grubbs para una anomalía, dos anomalías en un mismo extremo, o dos anomalías en extremos opuestos, en muestras pequeñas (cardinalidad menor que 30).

Listing 2.12: llamada por defecto a la función

```
grubbs.test(x, type = 10, opposite = FALSE, two.sided =
    FALSE)
```

La función devuelve una lista de clase `htest` con el resultado de la prueba. Si la variable `p.value` es menor que el nivel de confianza elegido, se puede rechazar la hipótesis nula y evaluar el objeto u objetos como anómalo.

Argumentos

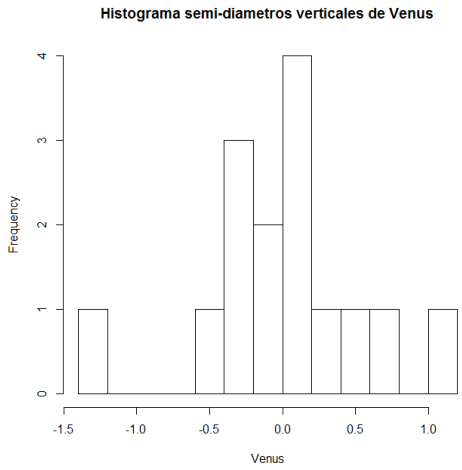
`x`	Un vector numérico para que contiene las observaciones a analizar.
`opposite`	Valor lógico, elige si el valor a evaluar no es aquel con mayor diferencia a la media, sino el opuesto (más bajo, si el más sospechoso es el mayor etc.)
`type`	Valor entero que especifica que variante de la prueba realizar. Para la prueba de una excepción 10 (el extremo es detectado automáticamente pero puede ser revertido con la función opposite), para evaluar la existencia de 2 anomalías, 11 evalúa si están en extremos opuestos, 20 evalúa si están en el mismo extremo.
`two.sided`	Valor lógico. Trata esta prueba como una prueba bilateral o unilateral.

Ejemplos

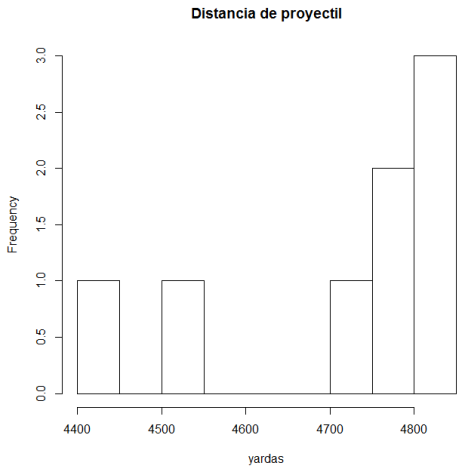
Ejemplo 1. Para este ejemplo se utilizaran observaciones del semi-diametro vertical de Venus hechas en 1846, al igual que en las publicaciones de Grubbs[cita] y de Tietjen-Moore[cita]:

```
> grubbs.test(Venus)

Grubbs test for one outlier

data:  Venus
G = 2.57370, U = 0.49305, p-value = 0.02178
alternative hypothesis: lowest value -1.4 is an outlier

> grubbs.test(Venus, opposite=TRUE)

Grubbs test for one outlier

data:  Venus
G = 1.8005, U = 0.7519, p-value = 0.4411
alternative hypothesis: highest value 1.01 is an outlier

> grubbs.test(Venus,type = 11,two.sided = TRUE)

Grubbs test for two opposite outliers

data:  Venus
G = 4.3743, U = 0.2920, p-value = 0.03039
alternative hypothesis: -1.4 and 1.01 are outliers

> hist(Venus, breaks = 11, main ="Histograma semi-diametros verticales de
    Venus")
```

El tipo de prueba por defecto evalúa el valor más alejado de la media, en este caso el menor, y lo rechaza como aberrante. Si evaluamos de la misma manera la observación mayor, la hipótesis nula no es rechazada. Sólo si evaluamos los dos extremos a la vez con la formula $S^2_{1,n}/S^2$ podemos rechazar ambos extremos como anomalías. En el siguiente histograma podemos observar la distribución con ambos extremos alejados del cuerpo del conjunto.



(a) El histograma muestra las observaciones extremas separadas del cuerpo de datos



(b) En el histograma las dos observaciones menores se separan del resto de datos

Figura 2.10

Ejemplo 2. En este ejemplo se analizara las distancias horizontales, en yardas, que alcanza el proyectil de un arma en distintas detonaciones con un mismo ángulo de elevación y una misma cantidad de pólvora, analizadas por Grubbs en 1969[cita]:

```
> proyectil = c(4782,4420,4838,4803,4765,4730,4549,4833)

> grubbs.test(proyectil)

Grubbs test for one outlier

data:  proyectil
G = 1.95990, U = 0.37287, p-value = 0.07663
alternative hypothesis: lowest value 4420 is an outlier

> grubbs.test(proyectil, type =20)

Grubbs test for two outliers

data:  proyectil
U = 0.054169, p-value = 0.001098
alternative hypothesis: lowest values 4420 , 4549 are outliers

> hist(proyectil, breaks=11, main="Distancia de proyectil", xlab="yardas")
```

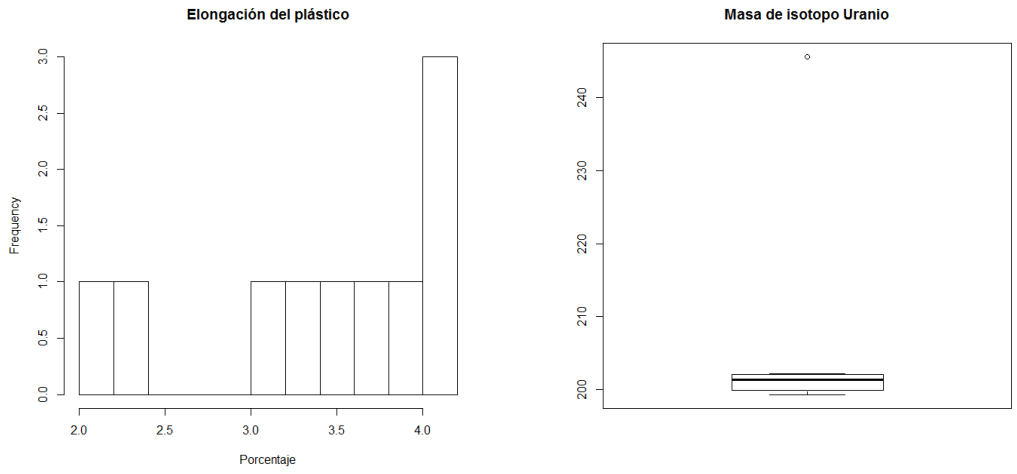
Al realizar la prueba para un solo valor aberrante, no podemos rechazar el valor menor con un nivel de confianza del 5 %. Solamente al realizar la prueba con la fórmula $S^2_{1,2}/S^2$ podemos rechazar ambos valores como anómalos. Nótese que al realizar la prueba de Dixon a estos valores, sólo se rechaza

la primera observación si la segunda es excluida de la prueba mediante la fórmula r_{20}, y tras eso se puede evaluar la siguiente anomalía mediante la fórmula r_{10}.

Ejemplo 3. Para el último ejemplo, se evaluara el porcentaje de elongación de un material plástico sujeto a un esfuerzo de tracción en el momento de rotura[cita]:

```
> plastico = c(3.73,3.59,3.94,4.13,3.04,2.22,3.23,4.05,4.11,2.02)
> grubbs.test(plastico, type=20)

Grubbs test for two outliers

data: plastico
U = 0.22361, p-value = 0.04532
alternative hypothesis: lowest values 2.02 , 2.22 are outliers

> hist(plastico,breaks = 10)
```

Si se observa el histograma de los datos, los dos datos menores se alejan notablemente del resto, por esto se utiliza $S^2_{1,2}/S^2$ para evaluar los datos de forma simultánea. Obtenemos un p-valor ligeramente inferior al nivel de confianza del 5 %, así que se pueden rechazar los valores como anomalías, pero con precaución. Nótese que la prueba de Dixon no rechaza la hipótesis nula de que no hay anomalías en los datos.



(a) Histograma muestra las dos observaciones menores separadas del resto

(b) La presencia de la anomalía distorsiona el diagrama de caja

Figura 2.11

Ejemplo 4. Se buscara valores anómalos entre las medidas de un espec-

trómetro de masa de un isotopo de uranio, ejemplo utilizado en la publicación de la prueba Tietjen-Moore [cita]:

```
> uranio2
[1] 199.31 199.53 200.19 200.82 201.92 201.95 202.18 245.57
> grubbs.test(uranio2,type=10)

Grubbs test for one outlier

data:  uranio2
G = 2.4688000, U = 0.0049308, p-value = 1.501e-07
alternative hypothesis: highest value 245.57 is an outlier

> grubbs.test(uranio2,type=20)

Grubbs test for two outliers

data:  uranio2
U = 0.003745, p-value < 2.2e-16
alternative hypothesis: highest values 202.18 , 245.57 are outliers

> grubbs.test(uranio2[1:7],type=10,opposite = TRUE)

Grubbs test for one outlier

data:  uranio2[1:7]
G = 1.11210, U = 0.75952, p-value = 0.9235
alternative hypothesis: highest value 202.18 is an outlier
```

En estos datos la observación sospechosa es la más grande. Al evaluarla como único outlier obtenemos el resultado esperado y la podemos rechazar. Pero si evaluamos la existencia de dos anomalías en un extremo con la fórmula $S^2_{n,n-1}/S^2$, también rechaza el segundo valor mayor $202,18$; cuando es evidente que no lo es. Esto es un claro ejemplo de saturación: el valor aberrante se aleja tanto que desvirtúa el resultado de la prueba de tal manera que se rechazan más valores de los que se debe. Si hacemos la prueba para una sola anomalía sin la presencia de este valor, se puede observar que $202,18$ ya no es rechazado como valor anómalo.

2.8.2. FindOutliersGrubbsTwosided {climtrends}

Esta función, contenida en el paquete `climtrends` [cita] realiza la prueba de Grubbs. Esta función tiene, en el mismo paquete, otras dos variantes similares prácticamente iguales excepto por la observación que evalúan y por el nivel de confianza de la prueba:

`FindOutliersGrubbsTwosided` Evalúa la observación que más se aleja de la media, con un nivel de confianza $\alpha/2$

`FindOutliersGrubbsOnesidedMax / FindOutliersGrubbsOnesidedMin` evalúan el valor máximo y el valor mínimo, respectivamente, con

un nivel de confianza α, el doble de que se haría la prueba bilateral.

Listing 2.13: llamada por defecto

```
FindOutliersGrubbsTwosided(dataSeries,alpha=0.05,
    iterative=TRUE)
```

La función devuelve una matriz con 3 columnas. Cada fila contiene el resultado de la prueba g, el valor crítico p, y la posición del objeto en el vector `posG`. La hipótesis nula se rechaza cuando $G > p$

Argumentos

`dataSeries`	Conjunto de datos a evaluar
`alpha`	Nivel de confianza
`iterative`	Valor lógico, TRUE= iterativo, FALSE= evalua sólo el primer valor

Esta función tiene la desventaja, frente a la del paquete `outliers`, de no poder evaluar dos valores simultáneamente, ni de manera unilateral ni de manera bilateral. Si se quiere evaluar dos observaciones la evaluación es realizada iterativamente, es decir, una tras otra, con los problemas que eso conlleva.

Ejemplos

Ejemplo 1. Se analizaran las observaciones del semi-diámetro vertical de Venus [cita] como en pruebas anteriores:

```
> FindOutliersGrubbsOnesidedMax(Venus)
G          p posG
[1,] 1.800527 1.490551    15
> FindOutliersGrubbsOnesidedMin(Venus)
G           p posG
[1,] 2.573737 1.4905506     1
[2,] 1.123457 0.9123365    11
> FindOutliersGrubbsTwosided(Venus)
G          p posG
[1,] 2.573737 1.773457     1
[2,] 2.218645 1.741531    15
[3,] 1.801255 1.706998    14
[4,] 1.690693 1.669432    13
[5,] 1.718569 1.628300    12
[6,] 1.639318 1.582921     2
[7,] 1.512859 1.410940     5
[8,] 1.467630 1.336233     6
[9,] 1.471199 1.248505     7
```

Al realizar la prueba para el extremo mayor, evalúa el valor $1,01$ como anómalo. Nótese que al evaluar esta observación con la función del paquete outliers no era rechazada. Si realizamos la prueba por el extremo inferior, evalúa el valor menor $-1,40$ y, ya que lo hace de modo iterativo, realiza la prueba otra vez y rechaza el valor en la posición 11 del vector, esto es, rechaza $0,20$ como valor aberrante. La única conclusión que se puede obtener de este segundo resultado es que existe algún error en el código de la función, aunque la diferencia entre el valor crítico G y p no sea muy grande. Por último, al realizar la prueba bilateral (lo único que cambia en la función es el nivel de confianza al obtener el límite del intervalo de confianza), evalúa todos los valores que puede (la función se detiene si $n \leq 6$) como anomalías, comenzando por el valor menor $-1,40$, y luego por los valores mayores $1,01$; $0,63, \ldots$. Esto es una indicación de que realizar la prueba repetidas veces no es una buena idea, aunque podemos observar que, exceptuando ambos extremos $-1,40$ y $1,01$; las demás observaciones no tienen una evaluación G mucho mayor que el valor crítico p. Aun así, ya que no es una prueba simultanea de ambos extremos, como la que se puede realizar con la función del paquete outliers, no es recomendable rechazar la segunda anomalía.

Ejemplo 2. En este ejemplo se analizara las distancias horizontales, en yardas, que alcanza el proyectil de un arma en distintas detonaciones con un mismo ángulo de elevación y una misma cantidad de pólvora, analizadas por Grubbs en 1969[cita]:

```
> FindOutliersGrubbsOnesidedMin(proyectil)
G          p posG
[1,] 1.959884 1.1684848    1
[2,] 2.096599 1.0969396    2
[3,] 1.491729 1.0130791    3
[4,] 1.239179 0.9123365    4

> FindOutliersGrubbsOnesidedMax(proyectil, iterative = FALSE)
G         p posG
[1,] 0.817172 1.168485    8
```

En este caso las observaciones sospechosas son las dos menores. Si realizamos la prueba iterativamente por el extremo inferior de los datos obtenemos 4 anomalías. En este caso, ya que este paquete no permite la evaluación de dos valores en un extremo, sólo es recomendable rechazar el primero. Si realizamos la prueba por el extremo superior no obtenemos ningún valor aberrante como era de esperar (la opción iterativa debe ser desactivada ya que si no, no devuelve la estadística G).

Ejemplo 3. se evaluara el porcentaje de elongación de un material plástico sujeto a un esfuerzo de tracción en el momento de rotura[cita]:

```
> FindOutliersGrubbsOnesidedMin(plastico)
G          p posG
[1,] 1.797485 1.2856344    1
[2,] 2.113252 1.2307025    2
[3,] 1.662862 1.1684848    3
[4,] 1.801104 1.0969396    4
[5,] 1.521627 1.0130791    5
[6,] 1.596849 0.9123365    6
```

En estos datos, las dos observaciones menores son las sospechosas de ser aberrantes. De nuevo, al no existir opción de evaluarlas simultáneamente, debemos utilizar la opción iterativa y de esta manera obtenemos 6 anomalías de un vector de 10 observaciones, por lo que lo recomendable es sólo rechazar la primera (el valor menor $2,02$).

Ejemplo 4. Se realizara la búsqueda de valores anómalos en las mediciones de $SO2/m^3$ de la estación de medida 28079036 en Madrid en enero de 2016[cita]:

```
> FindOutliersGrubbsOnesidedMax(E28079036$value, iterative = FALSE)
G          p posG
[1,] 2.714214 1.831937   29
```

Se observa que el valor 20 correspondiente al día 29 de enero es rechazado como anomalía del conjunto.

2.8.3. Referencias

Grubbs, F.E. (1950). Sample Criteria for testing outlying observations. Ann. Math. Stat. 21, 1, 27-58. Grubbs, Frank (February 1969), Procedures for Detecting Outlying Observations in Samples, Technometrics, 11(1), pp. 1-21. NIST/SEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/hand 2017.

2.9. Prueba de Tietjen-Moore

La prueba Tietjen-Moore es utilizada para la detección de múltiples valores aberrantes en un conjunto de datos univariable que sigue, aproximadamente, una distribución normal. Es una generalización de la prueba de Grubb en el caso de anomalías múltiples. Es importante destacar que la prueba de Tietjen-Moore requiere que el número concreto de valores anómalos sospechosos sea especificado exactamente. Si se realiza la prueba para una única anomalía, la prueba es equivalente a la prueba de Grubb. Con el fin de abordar el problema de anomalías múltiples en conjuntos univariables, Tietjen y Moore proponen dos evaluaciones de "tipo Grubb", para la comúnmente dada situación de encontrar k valores (de una muestra de tamaño n) que son grandes o pequeños comparados con el $n-k$ resto de valores . La prueba de Tietjen-Moore esta definida para las hipótesis:

- H_0: No hay valores aberrantes en el conjunto de datos
- H_a: Existen exactamente k valores aberrantes en el conjunto

La prueba se realiza de la manera siguiente:
Dada la población $x_1, x_2, \ldots, x_n$. Se ordenan las observaciones de manera ascendente tal que $y_1 \leq y_2 \leq \ldots \leq y_i \leq \ldots \leq y_n$ es el conjunto ordenado de las observaciones. Sea la mayor K de estos valores y son valores sospechosos de ser anómalos. La estadística para evaluar la hipótesis nula de que todos estos valores provienen de una misma población de distribución normal es:

$$L_k = \frac{\sum_{i=1}^{n-k} (y_i - \bar{y}_k)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}, \tag{2.9}$$

donde

$$\bar{y}_k = \frac{\sum_{i=1}^{n-k} y_i}{n-k}$$

e $\bar{y}$ es la media aritmética de la muestra completa. Esta estadística L_k puede ser utilizada para examinar los menores k valores de y modificando el numerador para solo incluir los $n-k$ valores mayores; esto es, el numerador para L_k^* es:

$$\bar{y}_k = \sum_{i=k+1}^{n} (y_i - \bar{y}_k^*)^2.$$

Las estadísticas L_k y L_k^* resultan útiles para examinar k valores sospechosos que son más grandes o más pequeños que el cuerpo principal de la muestra. Algunas muestras, sin embargo, muestran valores sospechosos a ambos lados del grueso de los datos. Para tratar esta situación, en la que algunos de los k valores sospechosos son más grandes y otros son más pequeños que el resto de valores, otra estadística es propuesta:
De nuevo, sean los valores de la muestra $x_1, x_2, \ldots, x_n$. Se computa la media aritmética $\bar{x}$, y, tras esto, se calculan los siguientes n valores residuales:

$$r_i = |y_i - \bar{y}|,$$

las observaciones son renombradas como z de tal manera que z_i es la observación x cuyo valor residual r_i es el i valor mayor. Esto quiere decir que z_1 es la observación más cercana al valor de media aritmética $\bar{x}$ y que z_n la observación más lejana de $\bar{x}$. Para evaluar la hipótesis nula de que todos los valores provienen de la misma distribución normal, se calcula:

$$E_k = \frac{\sum_{i=1}^{n-k}(z_i - \bar{z}_k)^2}{\sum_{i=1}^{n}(z_i - \bar{z}_k)^2}, \tag{2.10}$$

donde

$$\bar{z}_k = \frac{\sum_{i=1}^{n-k} z_i}{n-k},$$

es la media de los $n - k$ valores menos alejados y $\bar{z}$ la media de la muestra completa.

Las estadísticas $L_k(L_k^*)$ y E_k son utilizadas de la siguiente manera: si, en una muestra de tamaño n, se decide evaluar si los k valores mayores (o menores) son anomalías, se calcula L_k; si este valor es más pequeño que el valor crítico deseado, se puede concluir que los k valores mayores son valores anómalos. Si, en cambio, se quiere comprobar si los k valores "más extremos" (medidos desde el valor de la media aritmética) son anomalías, se calcula E_k; si esta cantidad es menor que el valor crítico elegido, se concluye que los k valores sospechosos son anomalías. De esta manera solo se realiza una única prueba en el nivel de confianza especificado sin preocupaciones sobre el efecto de aplicar la prueba repetidas veces. Además, el efecto de enmascaramiento ya no resulta un impedimento. La región crítica para la prueba Tietjen-Moore es obtenida mediante simulación. La simulación se lleva a cabo generando una

muestra normal aleatoria de tamaño n y calculando la prueba de Tietjen-Moore. Normalmente, 10.000 muestras aleatorias son utilizadas. El valor de la prueba de Tietjen-Moore obtenido del conjunto de datos se compara a esta distribución de referencia.

Listing 2.14: obtención de la región critica en R

```
test = c(1:10000)
for (i in 1:10000){
xx = rnorm(length(x))
test[i] = tm(xx,k)}
quantile(test,0.05)
```

El valor de esta prueba estadística esta entre cero y uno $[0, 1]$. Si no hay anomalías en el conjunto, el resultado del valor de la prueba es cercano a 1. Si, por otra parte, existen valores aberrantes, el valor estará más cerca de cero. Esto es por lo que esta prueba siempre es una prueba inferior unilateral sin importar que formula para la prueba se elija, L_k o E_k.

2.9.1. FindOutliersTietjenMooreTest {climtrends}

Esta función, contenida en el paquete `climtrends` [cita], evalúa los outliers mediante la fórmula E_k.

Listing 2.15: llamada por defecto

```
FindOutliersTietjenMooreTest(dataSeries,k,alpha=0.05)
```

Devuelve una lista con el nivel de confianza utilizado para llevar a cabo la prueba, el valor obtenido de E_k (denominado `T`) y el valor de la región crítica (denominado `Talpha`). La hipótesis nula de que no hay anomalías en el conjunto es rechazada si `T < Talpha` ($E_k <$ Valor crítico).

Argumentos

`dataSeries`	Conjunto de datos a evaluar
`k`	Número de anomalías que se sospecha el conjunto contiene
`alpha`	Nivel de confianza

Ejemplos

Ejemplo 1. Para este ejemplo se utilizaran observaciones del semidiámetro vertical de Venus hechas en 1846, al igual que en las publicaciones de Grubbs[cita] y de Tietjen-Moore[cita]:

```
> FindOutliersTietjenMooreTest(Venus,1)
$T
[1] 0.4930518

$Talpha
5%
0.5040493

> FindOutliersTietjenMooreTest(Venus,2)
$T
[1] 0.2919994

$Talpha
5%
0.3162285
```

En este caso las observaciones más extremas que se sospecha de ser anómalas están en extremos opuestos. Para esta función este hecho no importa y simplemente se indica que se sospecha que existen 2 excepciones. Como `T<Talpha` ó $E_K < Valor\ critico$, podemos rechazar la hipótesis nula y considerar $-1,40$ y $1,01$ valores aberrantes.

Ejemplo 2. Se buscara valores anómalos entre las medidas de un espectrómetro de masa de un isotopo de uranio, ejemplo utilizado por Tietjen y Moore [cita]:

```
> uranio2
[1] 199.31 199.53 200.19 200.82 201.92 201.95 202.18 245.57
> FindOutliersTietjenMooreTest(uranio2,1)
$T
[1] 0.004930824

$Talpha
5%
0.2650087

> FindOutliersTietjenMooreTest(uranio2,2)
$T
[1] 0.003372516

$Talpha
5%
0.1004958

> FindOutliersTietjenMooreTest(uranio2[1:7],1)
$T
[1] 0.6839662

$Talpha
5%
0.2048402
```

En este caso es evidente que el valor mayor es el aberrante, y así se prueba si se evalúa la existencia de un único outlier. Pero si realizamos la prueba para dos anomalías, también rechazamos la hipótesis nula y declara el valor $202,18$ como anómalo, cuando no lo es. Esto es otro claro ejemplo de satu-

ración, ya que el valor 245, 57 desvía la prueba lo suficiente para declarar un valor normal como anómalo, y si realizamos la prueba sin él, la hipótesis nula no es rechazada.

Ejemplo 3. En este ejemplo se analizara las distancias horizontales, en yardas, que alcanza el proyectil de un arma en distintas detonaciones con un mismo ángulo de elevación y una misma cantidad de pólvora, analizadas por Grubbs en 1969[cita]:

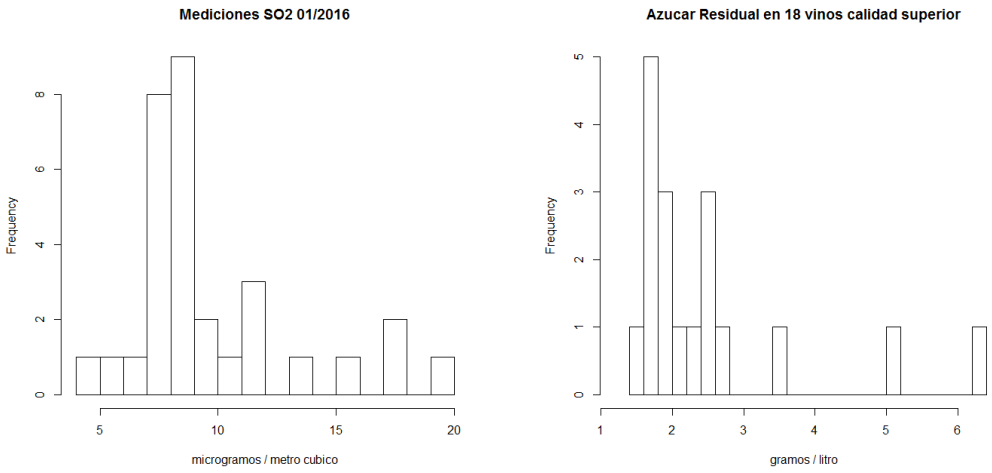
```
> proyectil
[1] 4420 4549 4730 4765 4782 4803 4833 4838
> FindOutliersTietjenMooreTest(proyectil, 1)
$T
[1] 0.3728741

$Talpha
5%
0.2585441

> FindOutliersTietjenMooreTest(proyectil, 2)
$T
[1] 0.0541694

$Talpha
5%
0.1005952

> FindOutliersTietjenMooreTest(proyectil[2:8], 1)
$T
[1] 0.1452753

$Talpha
5%
0.2186693

> FindOutliersTietjenMooreTest(c(proyectil[1], proyectil[3:8]), 1)
$T
[1] 0.06759144

$Talpha
5%
0.2080922
```

La prueba confirma que las dos observaciones menores corresponden a anomalías. En la primera prueba podemos observar el efecto de enmascaramiento entre los dos valores: en la ausencia de uno de ellos, la prueba de Tietjen-Moore rechaza la hipótesis nula y detecta un valor anómalo en el conjunto, mientras que, estando los dos valores, la prueba no rechaza el primero como excepcional.

Ejemplo 4. Se realizara la búsqueda de valores anómalos en las mediciones de $SO2/m^3$ de la estación de medida 28079036 en Madrid en enero de 2016[cita]:

```
> FindOutliersTietjenMooreTest( E28079036$value, 1)
$T
[1] 0.7462493

$Talpha
5%
0.7062538

> FindOutliersTietjenMooreTest( E28079036$value, 3)
$T
[1] 0.3836948

$Talpha
5%
0.4567217

> hist(E28079036$value, breaks = 15,xlab="microgramos / metro cubico", main=
    "Mediciones SO2 01/2016")
```

El resultado es que con un nivel de confianza del 5 % no se puede rechazar la hipótesis nula de que no existe ninguna anomalía frente a la existencia de un solo valor aberrante. En cambio, si realizamos la prueba para probar la existencia de tres anomalías, el test de Tietjen-Moore rechaza la hipótesis nula. Podemos observar la distribución en el histograma [enlace], donde vemos los 3 valores mayores alejados del cuerpo del conjunto.



(a) (b)

Figura 2.12

Ejemplo 5. Se probara la existencia de anomalías en un conjunto de vinos [cita] respecto a la cantidad de azúcar residual que contienen:

```
> FindOutliersTietjenMooreTest( subset(winequality_red, quality ==’8’)$‘
    residual sugar‘, 3)
$T
[1] 0.1329072

$Talpha
5%
0.2662221
> hist(subset(winequality_red, quality ==’8’)$‘residual sugar‘, breaks = 30,
     xlim = c(1.2,6.5), xlab = "gramos / litro", main = "Azucar Residual en
    18 vinos calidad superior")
```

Al observar el histograma [enlace], podemos observar que los dos valores mayores están notablemente alejados de los demás. Para confirmar si el tercero más grande también es un valor aberrante, realizamos la prueba para 3 anomalías, y, de esta manera, rechazamos la hipótesis nula.

Ejemplo 6. Se realizara la prueba para comprobar la existencia de valore anómalos entre las medidas del área de semillas de trigo de la variedad kama [enlace]:

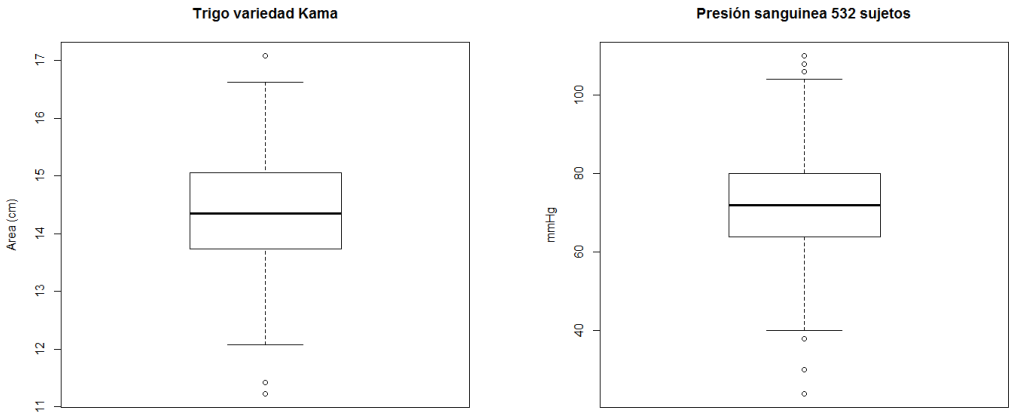
```
> boxplot(tr.kama$area, main="Trigo variedad Kama", ylab="Area (cm)")
> FindOutliersTietjenMooreTest(tr.kama$area, 3)
$T
[1] 0.7467143

$Talpha
5%
0.680032
```

Para confirmar los resultados obtenidos al generar un diagrama de caja [enlace], se prueba la existencia de 3 anomalías. Al contrario de lo que índica el diagrama, la prueba de Tietjen-Moore no rechaza la hipótesis nula y no podemos afirmar que esas tres observaciones son anómalas.

Ejemplo 7. Se realizara la prueba para comprobar la existencia de valore anómalos entre las medidas presión sanguínea en pacientes que se sospecha sufren de diabetes [enlace]:

```
> boxplot(diabetes$bp, main="Presión sanguinea 532 sujetos", ylab="mmHg")
> FindOutliersTietjenMooreTest(diabetes$bp,6)
$T
[1] 0.8738881

$Talpha
5%
0.890646
```

Para confirmar los resultados obtenidos al generar un diagrama de caja [enlace], se prueba la existencia de 3 anomalías. Al contrario de lo que índica el diagrama, la prueba de Tietjen-Moore no rechaza la hipótesis nula y no podemos afirmar que esas tres observaciones son anómalas.



(a) (b)

Figura 2.13

2.9.2. Referencias

Some Grubbs-Type Statistics for the Detection of Several Outliers, Gary L. Tietjen and Roger H. Moore, Technometrics, Vol. 14, No. 3 (Aug., 1972), pp. 583-597 NIST/SEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/d 2017. Testing For Normality, Henry C. Thode, 2002, CRC Press

2.10. Prueba de Desviación Extrema Studentizada general

La prueba de desviación extrema studentizada general (*generalized Extreme Studentized Deviation* o *generalized ESD*) es utilizada para detectar uno o más valores aberrantes en un conjunto de datos univariable, bajo la suposición de que el grueso de los datos proviene de una distribución normal.

La limitación principal de la prueba de Grubb y de la prueba de Tietjen-Moore es que el número de anomalías a evaluar, k, ha de ser especificado exactamente. Si k no es especificado correctamente, esto puede distorsionar los resultados de estas pruebas. Al contrario, la prueba de desviación extrema studentizada general solamente requiere que se especifique un límite superior de valores anómalos sospechosos.

Dado el límite superior, k, la prueba ESD general básicamente realiza k pruebas separadas: evalúa la presencia de una anomalía, evalúa la presencia de dos anomalías, y así hasta llegar a k valores anómalos. La prueba ESD general está definida para las hipótesis:

- H_0: No hay valores aberrantes en el conjunto de datos
- H_a: Existen hasta k anomalías en el conjunto de datos

Este procedimiento se basa en las estadísticas $R_1, \ldots, R_k$, que son las desviaciones extremas *studentizadas* calculadas de las sucesivas muestras reducidas de tamaño,$n, n-1 \ldots$ respectivamente. El procedimiento es el siguiente:
Se calcula

$$R_i = \frac{\max_i |x_i - \bar{x}|}{s} \tag{2.11}$$

donde $\bar{x}$ es la media y s la desviación típica. Tras esto, se calcula R_2, de la misma manera pero de la muestra reducida de tamaño $n-1$ obtenida de eliminar la observación correspondiente a $\max_i |x_i - \bar{x}|$ de la muestra completa. Así sucesivamente para $R_3, ..., R_k$.

Correspondientes a las k pruebas, se calculan los siguientes k valores críticos:

$$\lambda_i = \frac{(n-i)\, t_{p,\, n-i-1}}{\sqrt{(n-i-1+t^2_{p,\, n-i-1})(n-i+1)}}$$

Para $i = 1, 2, \ldots, k$; y donde $t_{p,v}$ es el centésimo punto porcentual de la distribución *t-student* con v grados de libertad y

$$p = 1 - \frac{\alpha}{2(n-i+1)}$$

El número de anomalías es determinado hallando el mayor valor para i donde $R_i > \lambda_i$

Estudios de simulaciones por Rosner indican que esta aproximación al valor crítico es muy precisa para conjuntos de cardinalidad mayor o igual a 25 y razonablemente precisa para $n \geq 15$ Nótese que, aunque la prueba de desviación extrema studentizada general es, en esencia, una prueba de Grubb aplicada sucesivamente, hay importantes diferencias:

- La prueba ESD general hace ajustes apropiados para los valores críticos basados en el número de anomalías que son evaluados, mientras al hacer repetidas pruebas de Grubb no.
- Si existe un efecto de enmascaramiento, aplicando sucesivas pruebas de Grubb puede que se detenga demasiado pronto.
- La prueba de Grubb permite tanto evaluaciones unilaterales (esto es, se puede especificar la prueba para el mínimo o el máximo) como evaluaciones bilaterales (evaluar el mínimo y el máximo al mismo tiempo). La prueba ESD general está restringida a evaluaciones bilaterales.

2.10.1. FindOutliersESDtest {climtrends}

Esta función, presente en el paquete `climtrends` [cita], realiza prueba de desviación extrema studentizada general (*generalized ESD*) a un conjunto de datos.

Listing 2.16: llamada por defecto

```
FindOutliersESDtest(dataSeries,k=10,alpha=0.05)
```

La función devuelve un conjunto de datos con el resultado para cada número de anomalías: el número de anomalías, el resultado de la prueba estadística y el valor crítico. El número de anomalías es determinado por el número mayor i para cual `Test.Stat > Critical.Val` ($R_i > \lambda_i$)

Argumentos

`dataSeries`	Conjunto de datos a evaluar.
`k`	Número de anomalías máximo que se sospecha que el conjunto tiene.
`alpha`	Nivel de confianza α

Nótese que la opción `alpha` no sirve para nada porque esta incrustada en el código R de la función: la prueba siempre se realiza con un nivel de confianza del 5 %.

Ejemplos

Ejemplo 1. Para este ejemplo se utilizaran observaciones del semi-diametro vertical de Venus hechas en 1846, al igual que en las publicaciones de Grubbs[cita] y de Tietjen-Moore[cita]:

```
> FindOutliersESDtest(Venus, k=5)
No.Outliers Test.Stat Critical.Val
1           1  2.573737     2.548308
2           2  2.218645     2.507321
3           3  1.801255     2.462033
4           4  1.690693     2.411560
5           5  1.718569     2.354730
```

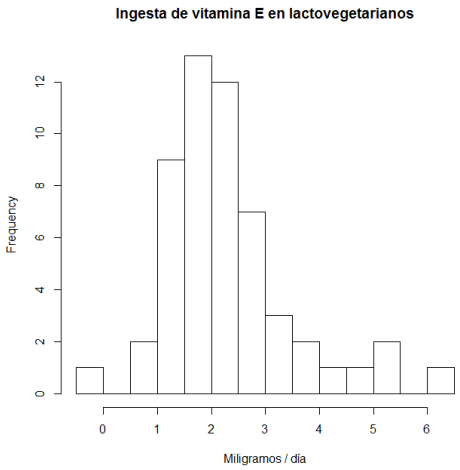
Aunque en pruebas anteriores ambos extremos del conjunto eran clasificados como anomalías, en este caso la prueba $GESD$ sólo considera que hay un valor aberrante.

Ejemplo 2. Este ejemplo lo realizaremos con el conjunto de datos probado por Rosner sobre la ingesta de vitamina E de lacto-vegetarianos[cita]:

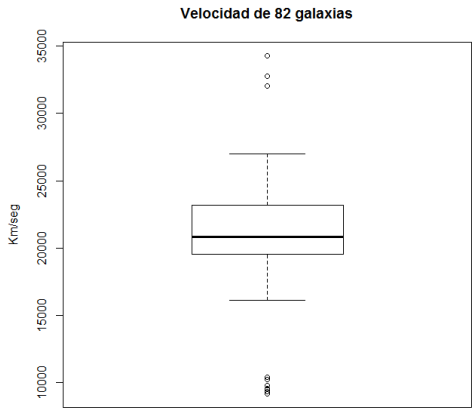
```
> FindOutliersESDtest(vitaminaE, k=8)
No.Outliers Test.Stat Critical.Val
1           1  3.118906     3.158794
2           2  2.942973     3.151430
3           3  3.179424     3.143890
4           4  2.810181     3.136165
5           5  2.815580     3.128247
6           6  2.848172     3.120128
7           7  2.279327     3.111796
8           8  2.310366     3.103243
> hist(vitaminaE,breaks=15, xlab = "Miligramos / día", main = "Ingesta de
    vitamina E en lactovegetarianos")
```

Para tres anomalías, podemos rechazar la hipótesis nula de la no existencia de valores aberrantes. Podemos observar el efecto del enmascaramiento cuando se evalúa la estadística para una o dos anomalías. En el histograma [enlace] podemos observar que los tres valores mayores se alejan del cuerpo principal de la distribución.

Ejemplo 3. Utilizaremos el conjunto de datos `galaxies` en el paquete `MASS` de R, que contiene la velocidad de 82 galaxias:



(a)



(b)

Figura 2.14

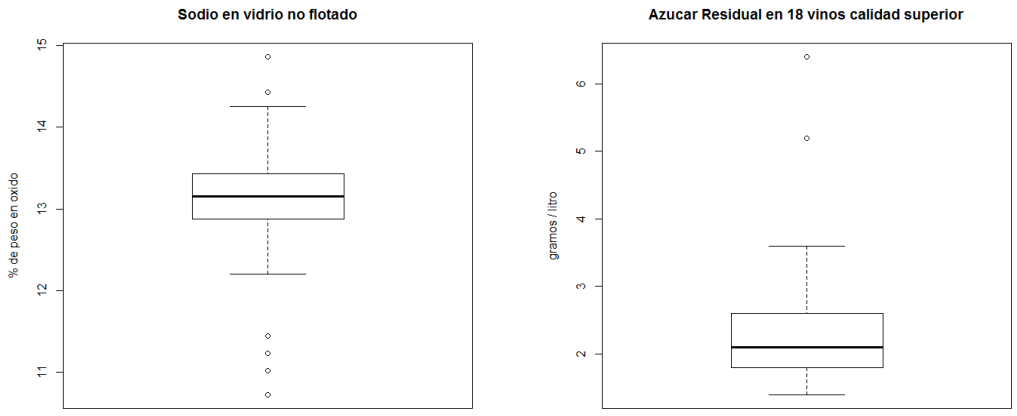
```
> FindOutliersESDtest(galaxies, k =20)
No.Outliers Test.Stat Critical.Val
1             1  2.947314     3.314935
2             2  2.796951     3.310562
3             3  2.789934     3.306121
4             4  2.830220     3.301613
5             5  2.961595     3.297033
6             6  3.131656     3.292382
7             7  3.358247     3.287656
8             8  3.601602     3.282852
9             9  3.838461     3.277970
10           10  4.262522     3.273006
11           11  2.531776     3.267957
12           12  2.530883     3.262821
13           13  2.535873     3.257596
14           14  2.639857     3.252277
15           15  2.278257     3.246863
16           16  2.033435     3.241349
17           17  1.961343     3.235733
18           18  1.833537     3.230010
19           19  1.853116     3.224177
20           20  1.919058     3.218230
> boxplot(galaxies, ylab = "Km/seg", main="Velocidad de 82 galaxias")
```

Para un máximo de 10 valores aberrantes, el valor de la prueba es mayor que el valor crítico, por lo que se rechaza la hipótesis nula de la no existencia de anomalías. Podemos observar la presencia de las anomalías en el diagrama de caja [enlace].

Ejemplo 4. Evaluaremos la presencia de anomalías en un conjunto de datos sobre vidrio no flotado[cita]. Para ello probaremos si existen anomalías en la cantidad de sodio que contienen los diferentes vidrios:

```
> FindOutliersESDtest( cristal2Na , 10)
No.Outliers Test.Stat Critical.Val
1             1  3.586053     3.287656
2             2  3.493836     3.282852
3             3  3.477879     3.277970
4             4  3.409531     3.273006
5             5  3.465599     3.267957
6             6  2.840758     3.262821
7             7  2.604980     3.257596
8             8  2.468698     3.252277
9             9  2.342131     3.246863
10           10  2.326122     3.241349
> boxplot(cristal2Na , ylab = "%de peso en oxido", main="Sodio en vidrio no
    flotado")
```

Mediante la prueba GESD se obtiene que para 5 anomalías el valor de la prueba es mayor que el valor crítico. Podemos observar la presencia de las anomalías en el diagrama de caja [enlace], aunque este indique 6 anomalías en lugar de 5.



(a) (b)

Figura 2.15

Ejemplo 5. La existencia de anomalías en un conjunto de vinos [cita] respecto a la cantidad de azúcar residual que contienen será probada:

```
> FindOutliersESDtest(subset(winequality_red, quality =='8')$`residual sugar
    `)
No.Outliers Test.Stat Critical.Val
1             1  2.951436     2.651599
2             2  3.153396     2.619964
3             3  2.622157     2.585676
4             4  1.790434     2.548308
5             5  1.733843     2.507321
6             6  1.599668     2.462033
7             7  2.054137     2.411560
8             8  1.900841     2.354730
9             9  2.024684     2.289954
10           10  1.403122     2.215004
> boxplot(subset(winequality_red, quality =='8')$`residual sugar`, ylab = "
    gramos / litro", main ="Azucar Residual en 18 vinos calidad superior")
```

La prueba índica que existen 3 anomalías en la cantidad de azúcar de los vinos. Clasifica los valores de azúcar de los 3 vinos con mayor cantidad como valores aberrantes. Si se muestran los datos en un diagrama de caja [enlace], el tercer vino con mayor cantidad no habría sido detectado.

2.10.2. Referencias

Percentage Points for a Generalized ESD Many-Outlier Procedure, Bernard Rosner, Technometrics, Vol. 25, No. 2 (May, 1983), pp. 165-172

NIST/SEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/h
2017.

Capítulo 3

Métodos de minería de datos

Preliminares

A diferencia de los métodos estadísticos anteriores, los métodos relacionados con la minería de datos no requieren parámetros, así, no asumen ningún modelo o mecanismo generador de los datos subyacente. Estos métodos están diseñados para poder gestionar un gran número de datos y de dimensiones.

Antes de discutir los diferentes métodos, se describen las modificaciones realizadas a los conjuntos datos previamente a la aplicación de los métodos, y las medidas para evaluar el rendimiento y la precisión de cada técnica aplicada a los diferentes conjuntos de datos.

Tras esto, se describirán cuatro métodos populares para la detección de anomalías en conjuntos de datos con gran número de dimensiones. Las cuatro clasifican cada objeto presente en el conjunto con un factor de anomalía basado en diferentes conceptos: **factor local de anomalía**, basado en el concepto de densidad local; **factor angular de anomalía**, clasificación basada en la varianza de los ángulos de los vectores a otros puntos; **grado de anomalía subespacio**, clasifica los obejtos en subespacios del espacio de datos original; y **empaquetado de características para detección de anomalías**, una técnica de ensamblado de múltiples iteraciones.

Preprocesamiento de datos

Medidas de Evaluación

Los métodos analizados en este capítulo devuelven una clasificación completa de todos los objetos del conjunto de datos basada en su grado de anomalía, puntuados de acuerdo al modelo de anomalía en el que el método está basado.

En la práctica, en cambio, el usuario de un método para la detección de anomalías está interesado en obtener un subconjunto de los datos relativamente pequeño formado por los objetos con mayor grado de anomalía. Para esos casos, donde el objetivo es un número n de candidatos a anomalía especificado con anterioridad, la medida de rendimiento más simple es la **precisión en *n*** (denotado como $P@n$), definida como la proporción de resultados correctos en los n puestos superiores. Para una conjunto de datos CD de tamaño N, constituido por anomalías (*outliers*) $O \subset CD$ y objetos corrientes (*inliers*) $I \subseteq CD$ ($CD = O \bigcup I$), $P@n$ es formulada como:

$$P@n = \frac{|\{o \in O | rango(o) \leq n\}|}{n} \tag{3.1}$$

Aquí se asume que el grado de anomalía es único y que si dos objetos tienen la misma puntuación, el empate se soluciona de la misma manera arbitraria en todos los casos.

Surge un problema al utilizar $P@n$ para evaluar los resultados de un método de detección de anomalía, ya que no está aclaro que valor elegir para el parámetro n. Elegir el número de anomalías para el parámetro, $n = |O|$, devuelve la conocida medida R-Precisión. Cuando el numero de anomalías $n = |O|$ es muy pequeño en relación a N, los valores de $P@n$ pueden ser engañosamente bajos y poco informativos. Por otra parte, si el número de anomalías $n = |O|$ es relativamente grande (en el mismo orden que N), se pueden obtener valores altos de $P@n$ simplemente por la pequeña cantidad de valores corrientes disponibles. Para hacer de la comparación de resultados más fácil, la precisión en n se puede ajustar por probabilidad.

El principal motivo del ajuste por probabilidad es permitir la comparación entre distintos ajustes donde los valores esperados de las puntuaciones varíen, alineando aquellas puntuaciones que están cerca de su valor esperado. Para la medida $P@n$, el valor máximo posible es $|O|/n$ si $n > |O|$, y 1 en cualquier otro caso. El valor esperado de un hipotético método de detección de anomalías aleatorio es $|O|/N$, que no depende de n. Si $n \leq |O|$, obtenemos el ajuste de la formula:

$$P@n_{ajustada} = \frac{P@n - |O|/N}{1 - |O|/N} \tag{3.2}$$

Para valores mayores de n, el máximo O/n debe utilizarse en lugar de 1 en la ecuación

Un desafió importante tanto en el diseño como en la evaluación de métodos de anomalía es lidiar con el desequilibrio entre el número de datos corrientes y anómalos: normalmente se espera que $i \gg o$, y que $I \approx N$.

Aunque las medidas $P@n$ y $P@n$ *ajustada* son fácilmente interpretadas, son sensibles a la elección de n, en especial cuando n es pequeño. Por ejemplo, para un conjunto de datos con 10 anomalías y 1 millón de datos normales, un algoritmo que asigna los verdaderos valores aberrantes en los (bastante altos) rangos 11-20 tendrá $P@10 = 0$, pero $P@20 = 0,5$. La medida $P@n$ ajustada puede igualmente verse afectada por la elección de n. Para evitar este inconveniente, las siguientes medidas utilizan la media de diferentes valores de n. El uso de la evaluación R-Precisión ($P@|O|$) asume implícitamente que el usuario tiene algún conocimiento del número de anomalías en el conjunto. Este no es siempre el caso, por lo que se ha dado mayor importancia a medidas que incluyen el rendimiento en un amplio rango de posibles valores de n. Una de estas medidas, popular en el contexto de búsqueda de información, es la **precisión media** (*Average Precision* o AP):

$$AP = \frac{1}{|O|} \sum_{o \in O} P@rango(o) \tag{3.3}$$

En lugar de evaluar un solo valor de n, se realiza la media con los valores de $P@n$ en los rangos de todos los valore anómalos $o \in |O|$. Existen otras variantes, como evaluar a lo largo de una muestra de los rangos en intervalos fijados. De cualquier manera, la evaluación en rangos de anomalías parece tanto popular como bien adaptada para conjuntos de datos desequilibrados. La precisión media también puede ser ajustada de la misma manera que $P@n$. Una clasificación perfecta devuelve un máximo valor de 1, y el valor esperado de una clasificación aleatoria es $|O|/N$:

$$AP_{ajustada} = \frac{AP - |O|/N}{1 - |O|/N)} \tag{3.4}$$

Para la *precisión en n* y la *precisión media*, el ajuste no es estrictamente necesario cuando los rendimientos de dos métodos en el mismo conjunto de datos (es decir, con la misma proporción de anomalías) son comparados en términos relativos. El ajuste por probabilidad es útil si la medida se va a interpretar en valores absolutos. El ajuste por probabilidad es estrictamente necesario si el rendimiento se va a comparar con distintos conjuntos de datos con distinta proporción de anomalías. Estas comparaciones pueden llevar a conclusiones incorrectas si no están ajustadas.

3.1. LOF: Local Outlier Factor

Desarrollado por Markus Breunig, Hans-Peter Kriegel, Raymond Ng y Jörg Sander [cita], este algoritmo asigna un grado de outlier a cada objeto en el conjunto de datos: el factor local de anomalía. Lo que lo hace local es que el grado de outlier depende de cómo aislado está el objeto con respecto a la vecindad alrededor, ya que limita los objetos de su vecindad con los que se calcula el factor.

El enfoque de este algoritmo está relacionado con la agrupación basada en densidad. En la primera publicación el algoritmo era denominado OPTICS-OF ("*OPTICS with Outlier Factors*") [cita] debido a la relación con el algoritmo de agrupación OPTICS ("*Ordering Points To Identify the Clustering Structure*")[cita] cuyo funcionamiento se basa en el agrupamiento por densidad.

Para entender cómo funciona el algoritmo, se debe definir los términos involucrados en el cálculo. Sea p un objeto de un conjunto de datos D, sea ε un valor de distancia, sea k un número natural y sea d una medida de distancia en D:

Vecindad ε de un objeto p: conjunto que contiene aquellos objetos x a una distancia de p menor o igual a ε:

$$N_\varepsilon(p) = \{x \in D | d(p,x) \leq \varepsilon\}$$

K-distancia de un objeto p: es la distancia $d(p,o)$ entre p y un objeto $o \in D$ tal que, para al menos k objetos, $o' \in D|\{p\}]$, se cumple que $d(p,o') \leq d(p,o)$, y que, para máximo $k-1$ objetos $o' \in D|\{p\}$, se cumple que $d(p,o') < d(p,o)$.

Vecindad k-distancia de p: Dada la k-distancia del objeto p, la vecindad k-distancia de p contiene todos los objetos cuya distancia a p es menor o igual que la k-distancia:

$$N_{k\text{-}distancia(p)}(p) = \{\ q\ \in D \setminus \{p\} \mid d(p,q) \leq k\text{-}distancia(p)\} \tag{3.5}$$

Pudiendo abreviarse como $N_k(p)$, y comúnmente denominada como los *k-vecinos más cercanos* o *k-nn* por sus siglas en inglés. Nótese que $N_k(p)$ puede ser mayor que k ya que más de un objeto puede estar a la misma distancia.

En este enfoque, una agrupación o clúster son formalmente definidos como conjuntos máximos de objetos conectados por densidad. Un objeto p esta conectado por densidad a otro objeto q si existe un objeto o tal que ambos p y q son alcanzables por densidad desde o (tanto directamente como transitivamente). Un objeto p es directamente alcanzable por densidad desde o si p se encuentra en la vecindad de o y o es un objeto núcleo.

Distancia de alcance de p con respecto de o: es la distancia menor tal que p es alcanzable por densidad desde o.

$$dist\text{-}alcance_k(p,o) = max\{k\text{-}distancia(o), d(p,o)\}. \tag{3.6}$$

Es la distancia entre dos objetos, pero con la diferencia de que si están lo suficientemente cerca la distancia es reemplazada por la k-distancia de o.

Un objeto núcleo es aquel cuya ε-*vecindad* contiene al menos $MinPts$ objetos. El concepto de distancia de alcance por densidad es en lo que se basa el algoritmo de agrupación en el que esta basado LOF, y cuyo resultado sólo depende de este último parámetro $MinPts$, que es el número de vecinos más cercanos utilizado para definir la vecindad local de un objeto, y utilizado para obtener la *distancia de alcance de un objeto p*.

Densidad local de alcance de p: intuitivamente, es la inversa de la media de distancias de alcance basadas en los $MinPts$-vecinos más cercanos de p.

$$lrd_{MinPts}(p) = 1/\left(\frac{\sum_{o \in N_{MinPts}(p)} dist\text{-}alcance_{MinPts}(p,o)}{\mid N_{MinPts}(p) \mid}\right) \tag{3.7}$$

La razón para utilizar la distancia de alcance en lugar de simplemente la distancia entre p y sus vecinos o es que debilita significativamente la fluctuación estadística de las distancias entre los objetos interiores: lrds para objetos cerca los unos de los otros sera igualado ya que es, como mínimo, tan grande como la $k - distancia$.

Factor Local de Outlier: el factor outlier de un objeto p es el grado de anomalía de p como:

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{\mid N_{MinPts}(p) \mid} \tag{3.8}$$

Es la media de los ratios de la densidad local de alcance de p y de aquellos puntos en la vecindad $k - distancia$ de p ($k = MinPts$).

Un resultado de $LOF \leq 1$ indica un claro dato corriente dentro de una agrupación, esto es, la densidad de la vecindad del objeto es comparable a la de los objetos cercanos. El problema de LOF es que no hay un criterio claro para cuando un punto es una anomalía: en un conjunto de datos, un valor $LOF = 1.1$ ya puede indicar una excepción, pero en otro conjunto de datos, con otra parametrización (con fluctuaciones locales importantes), una observación con valor $LOF = 2$ puede todavía pertenecer a una agrupación.

En la publicación [cita], los autores discuten los límites de los valores obtenidos mediante este método dependiendo de si el objeto está en el interior de una agrupación, o se encuentra cerca de su periferia, o si este está fuera y es una anomalía. Estos valores varían también según la elección del parámetro $MinPts$.

El parámetro $MinPts$ define, como se ha descrito anteriormente, el número de objetos más cercanos utilizados para definir la vecindad local de un objeto. La elección de este parámetro no es trivial, ya que, si es muy grande, el calculo de LOF para objetos en una agrupación incluirá otros de otras agrupaciones u objetos aberrantes; y, si es muy pequeño, las fluctuaciones estadísticas serán muy grandes. Por estas razones, la recomendación de los autores es la de definir un rango entre el limite inferior $MinPtsLI$ y el límite superior $MinPtsLS$.

Para elegir $MinPtsLI$, este valor debe ser considerado como el mínimo numero de objetos que una agrupación debe contener para que otros objetos puedan ser anomalías locales en relación a este grupo, y lo suficientemente grande para debilitar las fluctuaciones estadísticas. Los autores recomiendan un valor entre 10 y 20 según las pruebas realizadas es sus experimentos. Para $MinPtsLS$, debe ser considerado como el máximo numero de objetos cercanos que tengan potencial para ser anomalías locales.

Una vez determinado el rango $[MinPtsLI - MinPtsLS]$, el factor local de anomalía LOF de un objeto será el valor máximo obtenido en este rango para el parámetro $MinPts$ tal que:

$$LOF(p) = max\{LOF_{MinPts}(p)|MinPtsLI \leq MinPts \leq MinPtsLs\}$$

El algoritmo consta de **dos fases** para calcular los valores LOF dentro de un rango $[MinPtsLI - MinPtsLS]$ para todos los n objetos de un conjunto de datos D. En la primera fase se encuentran los $MinPtsLS$-vecinos más cercanos y, en la segunda fase, se calcula el LOF de cada objeto:

En el primer paso, se encuentran los $MinPtsLS$-vecinos más cercanos para cada objeto p, a la vez que se calcula su distancia a p. De esta manera se obtiene una base de datos M de tamaño $n * MinPtsLS$ distancias. La complejidad computacional de este paso es $O(n * tiempo\ consulta\ k - nn)$. Para las consultas de vecino más cercano, hay diferentes métodos disponibles, siendo la de índices (árbol-X, árbol-R, árbol-M) la recomendada para espacios dimensionales de no muy alto número dimensiones, que resulta en una complejidad para el primer paso $O(n \log n)$.

En el segundo paso los valores LOF se calculan utilizando la base de datos M, prescindiendo del conjunto original de los datos D. La base de datos M es escaneada dos veces para cada valor $MinPts$ en el rango $[MinPtsLI -$

$MinPtsLS$], una vez para obtener la *densidad local de alcance* de cada objeto, y otra para finalmente calcular los valores LOF. La complejidad computacional de este paso es de $O(n)$.

Ankerst M., Breunig M. M., Kriegel H.-P., Sander J.: "OPTICS: Ordering Points To Identify the Clustering Structure", Proc. ACM SIGMOD Int. Conf. on Management of Data, Philadelphia, PA, 1999.

OPTICS-OF: Identifying Local Outliers Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng1, Jörg Sander

"LOF: Identifying Density-Based Local Outliers" Markus M. Breunig†, Hans-Peter Kriegel†, Raymond T. Ng‡, Jörg Sander†

3.1.1. lof {dbscan}

Esta función, contenida en el paquete `dbscan` [cita], calcula el valor del factor local de anomalía LOF de cada objeto, utilizando para ello un árbol-kd (árbol k-dimensional) para las consultas de vecinos más cercanos.

Listing 3.1: Llamada por defecto

```
lof(x, k = 4, ...)
```

La función devuelve un vector numérico que contiene los valores LOF de cada observación en el conjunto de datos.

Argumentos

`x`	el conjunto de datos o un objeto `dist`.
`k`	número de vecinos o elementos próximos a encontrar.

Estos son los dos argumentos esenciales, pudiendo utilizar otros para la función `knn`, entre los que cabe destacar:

`search`	Mediante este argumento se índica que método utilizar para la búsqueda de los vecinos más cercanos. Tres valores posibles: `kdtree` para utilizar árbol-kd, `linear` para búsqueda lineal, o `dist` para precalcular la distancia euclidiana con R.

Ejemplos

Ejemplo 1. En este ejemplo se probara el método con un conjunto de datos de flores Iris [enlace], que contiene 50 observaciones de la variedad *versicolor* y 3 de la variedad *virginica*:

```
#Construcción del conjunto de datos
> iris.test3 <- rbind(versicolor,  virginica[sample(50, 3),])
#Método con 4 valores diferentes knn
> iris.test3[,"LOF3"] <- lof(iris.test3[ ,1:4], k=3)
> iris.test3[,"LOF5"] <- lof(iris.test3[,1:4], k=5)
> iris.test3[,"LOF10"] <- lof(iris.test3[,1:4], k=10)
> iris.test3[,"LOF15"] <- lof(iris.test3[,1:4], k=15)
#Generamos los diagramas de barras
> par(mfrow=c(2,2), oma=c(0,0,2,0))
> barplot(iris.test3$LOF3, main="knn=3", ylab="LOF", space=0, col = ifelse(
    iris.test3$Species=="virginica", "red", "grey"))
> barplot(iris.test3$LOF5, main="knn=5", ylab="LOF", space=0, col = ifelse(
    iris.test3$Species=="virginica","red","grey"))
> barplot(iris.test3$LOF10, main="knn=10", ylab="LOF", space=0, col = ifelse
    (iris.test3$Species=="virginica", "red", "grey"))
> barplot(iris.test3$LOF15, main="knn=15", ylab="LOF", space=0, col = ifelse
    ( iris.test3$Species=="virginica", "red", "grey"))
> mtext("LOF trigo.test3 con dbscan::lof", outer=TRUE, cex=1.5)
```

Knn	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
3	0,6666667	0.6466667	0.6464646	0.6252525
5	0,6666667	0.6466667	0.6031746	0.5793651
10	0,3333333	0.2933333	0.497076	0.4669006
15	0,3333333	0.2933333	0.4944444	0.4641111

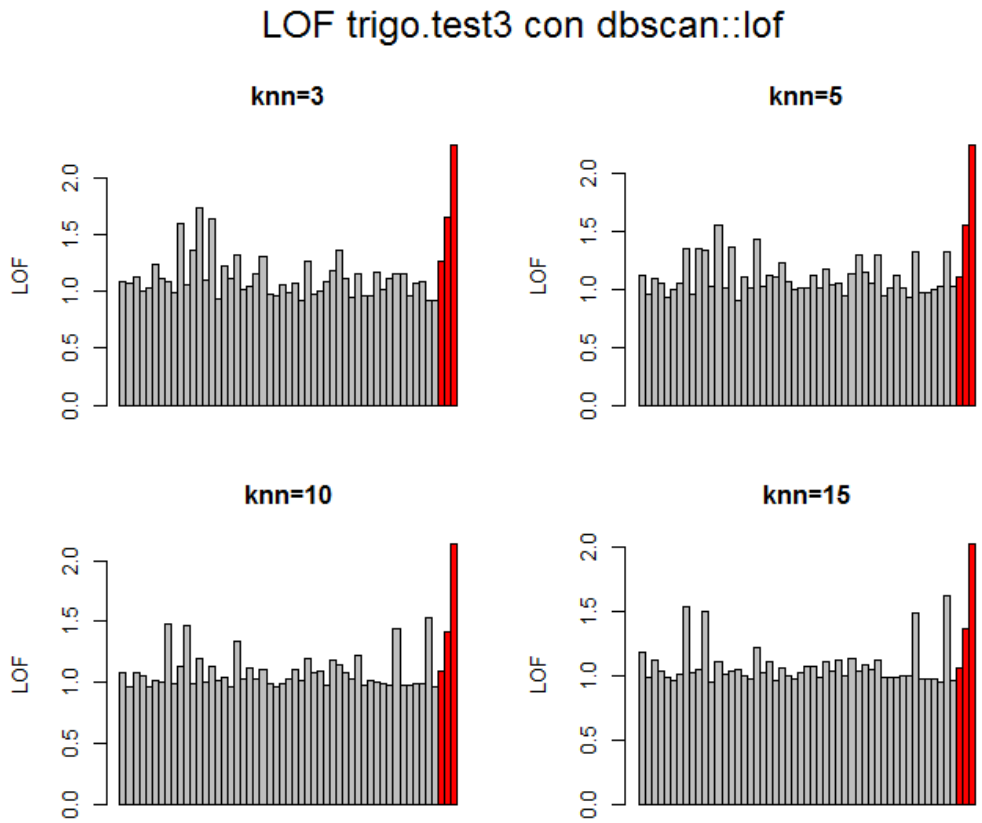


Figura 3.1: Clasificación *LOF* para semillas de trigo diferentes variedades

Ejemplo 2. En este ejemplo se analizara un conjunto de datos de lecturas de radar sobre señales retornadas de la ionosfera [cita]. Este conjunto de datos diferencia entre "buenas" lecturas, las cuales muestran evidencia de algún tipo de estructura en la ionosfera; y "malas" lecturas, las cuales atraviesan la ionosfera. Contiene 351 observaciones, de las cuales 126 son malas (35,9 %) y 225 son buenas (64,1 %):

```
> ionosphere[,"lof10"] <- lof(ionosphere[3:34], k=10)
> ionosphere[,"lof15"] <- lof(ionosphere[3:34], k=15)
> ionosphere[,"lof20"] <- lof(ionosphere[3:34], k=20)
> ionosphere[,"lof30"] <- lof(ionosphere[3:34], k=30)
> ionosphere[,"lof40"] <- lof(ionosphere[3:34], k=40)
> par(mfrow=c(2,2), oma=c(0,0,2,0))
> barplot(ionosphere$lof10, main="knn=10", ylab="LOF", space=0, border= NA,
    col = ifelse(ionosphere$X35 == "b", "red", "black"))
> barplot(ionosphere$lof15, main="knn=15", ylab="LOF", space=0, border= NA,
    col = ifelse(ionosphere$X35 == "b", "red", "black"))
> barplot(ionosphere$lof40, main="knn=40", ylab="LOF", space=0, border= NA,
    col = ifelse(ionosphere$X35 == "b", "red", "black"))
> barplot(ionosphere$lof80, main="knn=80", ylab="LOF", space=0, border= NA,
    col = ifelse(ionosphere$X35 == "b", "red", "black"))
> mtext("LOF trigo.test3 con dbscan::lof", outer=TRUE, cex=1.5)
```

Knn	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
10	0.8253968	0.727619	0.8681743	0.7943519
15	0.7857143	0.6657143	0.8325965	0.7388506
20	0.7698413	0.6409524	0.8276966	0.7312067
30	0.7301587	0.5790476	0.8322785	0.7383544
40	0.7301587	0.5790476	0.8378014	0.7469701
80	0.7301587	0.5790476	0.8563133	0.7758488

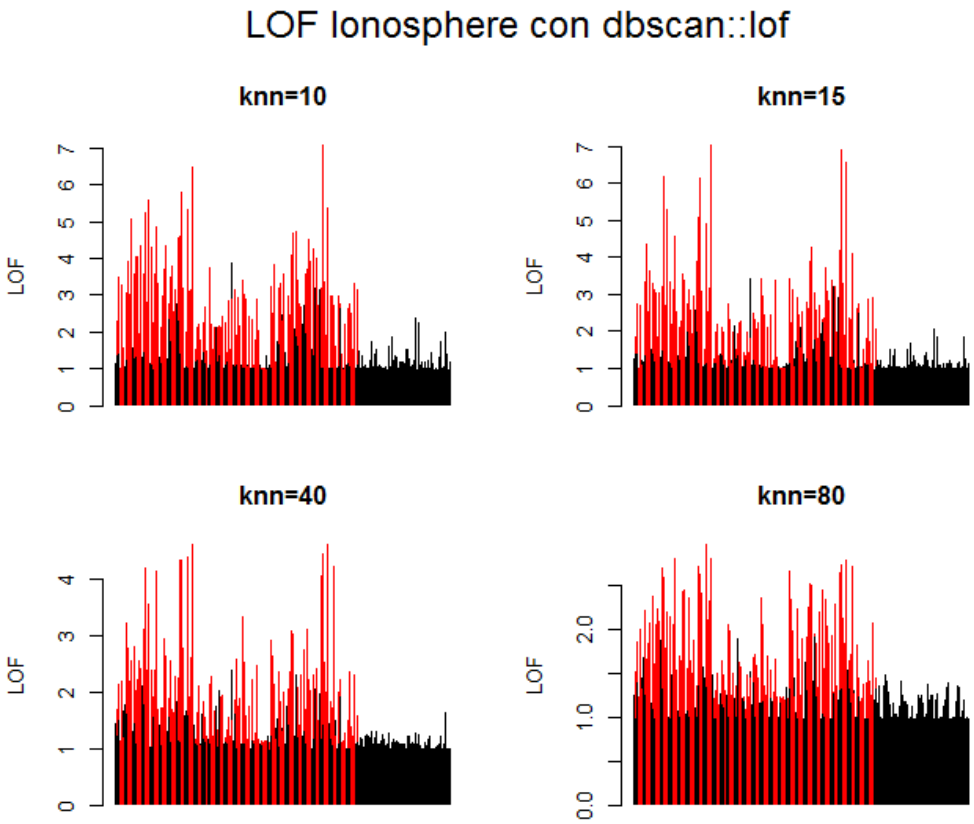


Figura 3.2: Clasificación de observaciones según parámetro de vecinos cercanos

Ejemplo 3. En este ejemplo utilizaremos una versión de un conjunto de datos sobre diferentes tipos de cáncer, benignos o malignos [cita original]. En esta versión [enlace versión], se ha reducido el número de outliers a 10 y se han eliminado observaciones duplicadas. El resultado es este conjunto con 223 objetos, 10 de ellos (2,2 %) anomalías:

```
> WBC.norm.v5[,"LOF10"] <- lof(WBC.norm.v5[,1:9],10)
> WBC.norm.v5[,"LOF20"] <- lof(WBC.norm.v5[,1:9],20)
> WBC.norm.v5[,"LOF30"] <- lof(WBC.norm.v5[,1:9],30)
> WBC.norm.v5[,"LOF40"] <- lof(WBC.norm.v5[,1:9],40)
> WBC.norm.v5[,"LOF60"] <- lof(WBC.norm.v5[,1:9],60)
> WBC.norm.v5[,"LOF80"] <- lof(WBC.norm.v5[,1:9],80)
> barplot(WBC.norm.v5$LOF20, main ="k=20", ylab = "LOF", col = ifelse(WBC.
    norm.v5$outlier == "yes", "red", "black"), border = NA)
> barplot(WBC.norm.v5$LOF60, main ="k=60", ylab = "LOF", col = ifelse(WBC.
    norm.v5$outlier == "yes", "red", "black"), border = NA)
> barplot(WBC.norm.v5$LOF100, main ="k=100", ylab = "LOF", col = ifelse(WBC.
    norm.v5$outlier == "yes", "red", "black"), border = NA)
> barplot(WBC.norm.v5$LOF120, main ="k=120", ylab = "LOF", col = ifelse(WBC.
    norm.v5$outlier == "yes", "red", "black"), border = NA)
> mtext("LOF␣WBC.v5␣con␣dbscan::lof", outer=TRUE, cex=1.5)
```

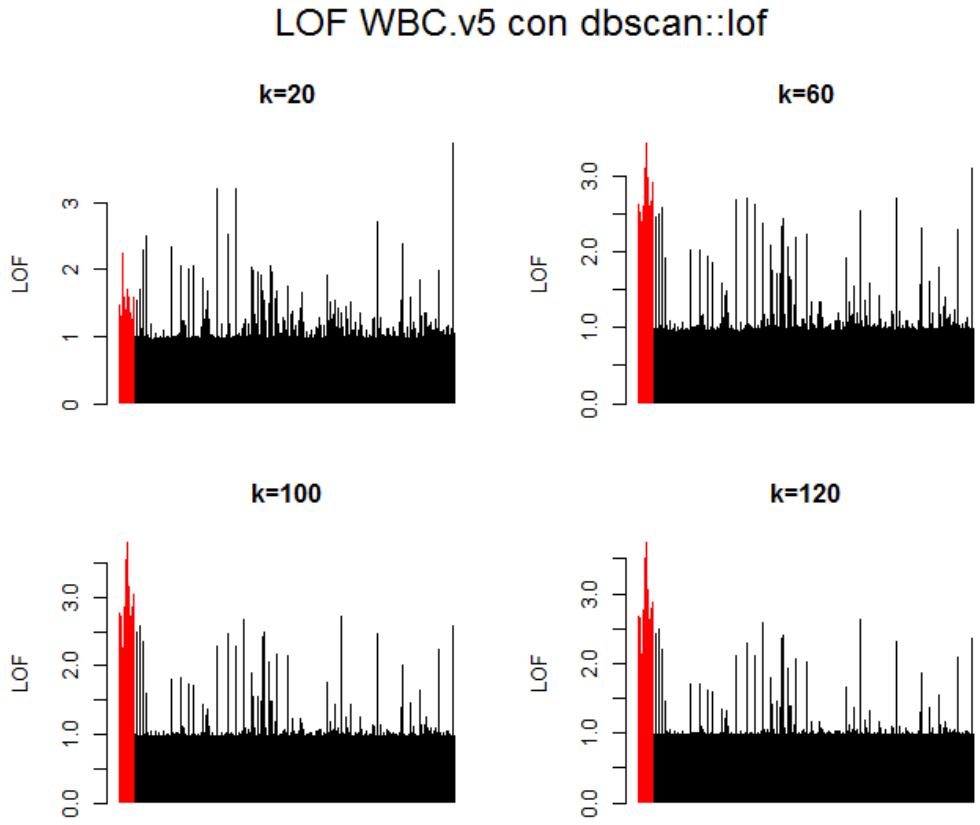


Figura 3.3: Clasificación *LOF* con diferentes parámetros *KNN*

Knn	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
10	0.8253968	0.727619	0.8681743	0.7943519
15	0.7857143	0.6657143	0.8325965	0.7388506
20	0.7698413	0.6409524	0.8276966	0.7312067
30	0.7301587	0.5790476	0.8322785	0.7383544
40	0.7301587	0.5790476	0.8378014	0.7469701
80	0.7301587	0.5790476	0.8563133	0.7758488

Ejemplo 4. Este conjunto contiene 70 medidas de semillas de trigo de la variedad *kama* y 6 medidas de la variedad *canadian* [cita]:

```
> trigo.test <- rbind(tr.kama, tr.canadian[sample(70,6),])
> trigo.test[,"LOF3"] <- lof(trigo.test[,1:7],k = 3)
> trigo.test[,"LOF5"] <- lof(trigo.test[,1:7],k = 5)
> trigo.test[,"LOF10"] <- lof(trigo.test[,1:7],k = 10)
> trigo.test[,"LOF15"] <- lof(trigo.test[,1:7],k = 15)
> trigo.test[,"LOF20"] <- lof(trigo.test[,1:7],k = 20)
> trigo.test[,"LOF30"] <- lof(trigo.test[,1:7],k = 30)
> trigo.test[,"LOF40"] <- lof(trigo.test[,1:7],k = 40)
> barplot(trigo.test$LOF10, main ="k=10", ylab = "LOF", col = ifelse(trigo.
    test$type == "3", "red", "grey"), space = 0 )
> barplot(trigo.test$LOF20, main ="k=20", ylab = "LOF", col = ifelse(trigo.
    test$type == "3", "red", "grey"), space = 0)
> barplot(trigo.test$LOF30, main ="k=30", ylab = "LOF", col = ifelse(trigo.
    test$type == "3", "red", "grey"), space = 0)
> barplot(trigo.test$LOF40, main ="k=40", ylab = "LOF", col = ifelse(trigo.
    test$type == "3", "red", "grey"), space = 0)
> mtext("LOF trigo.test con dbscan::lof", outer=TRUE, cex=1.5)
```

Knn	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
3	0	-0.08571429	0.1234028	0.04826588
5	0.1666667	0.0952381	0.2277293	0.1615347
10	0.1666667	0.0952381	0.2687354	0.2060556
15	0.3333333	0.2761905	0.3123545	0.2534135
20	0.3333333	0.2761905	0.3679563	0.3137812
30	0.5	0.4571429	0.4507797	0.4037037
40	0.5	0.4571429	0.4588235	0.412437

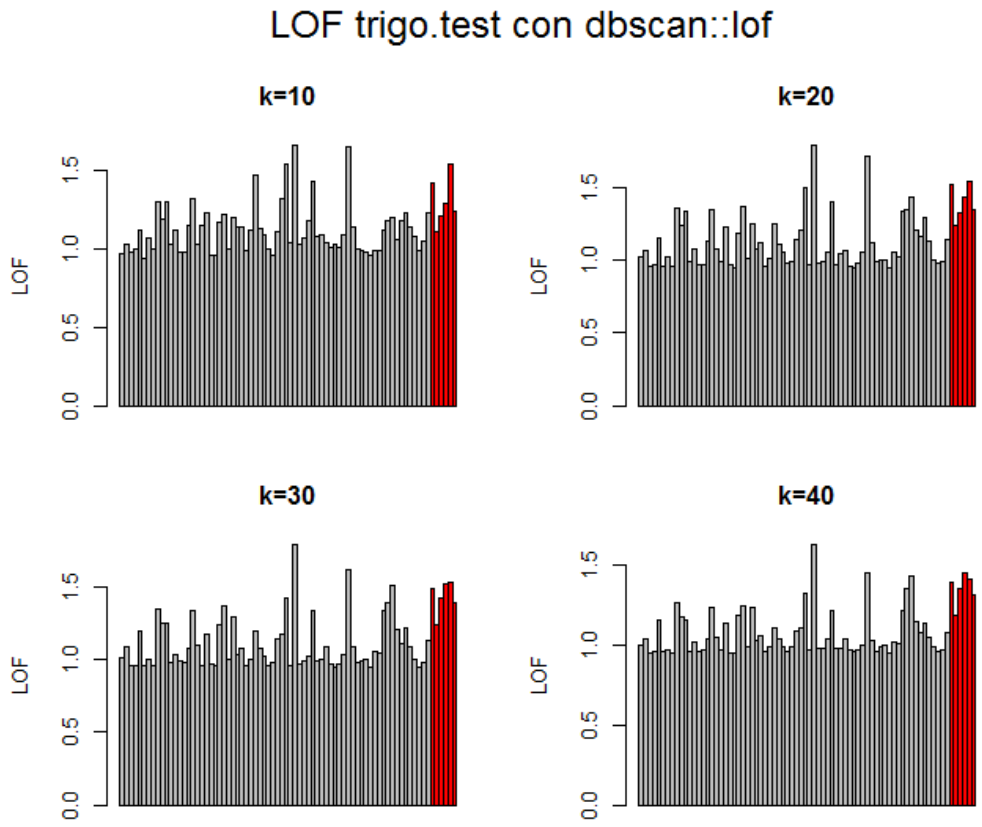


Figura 3.4: Clasificación LOF con diferentes parámetros KNN

3.1.2. lofactor {DMwR}

Esta función obtiene el factor local de anomalía LOF de cada objeto en un conjunto. La función, contenida en el paquete `DMwR`, es una reimplementación de la función con el mismo nombre originalmente en el paquete `dprep`, pero este ya no es soportado por las versiones actuales de R.

```
lofactor(data, k)
```

La función devuelve un vector numérico que contiene los valores LOF de cada observación en el conjunto de datos.

Argumentos

`x`	El conjunto de datos (internamente será convertido a una matriz)
`k`	Número de vecinos o elementos próximos utilizados en el cálculo de LOF

Ejemplos

Ejemplo 1. Este conjunto contiene 70 medidas de semillas de trigo de la variedad *canadian* y 3 medidas de la variedad *kama* [cita]:

```
> trigo.test2 <- rbind(tr.canadian, tr.kama[sample(1:70,3),])
> trigo.test2[,"lofactor10"] <- lofactor(trigo.test2[1:7], 10)
> trigo.test2[,"lofactor15"] <- lofactor(trigo.test2[1:7], 15)
> trigo.test2[,"lofactor20"] <- lofactor(trigo.test2[1:7], 20)
> trigo.test2[,"lofactor30"] <- lofactor(trigo.test2[1:7], 30)
> par(mfrow=c(2,2), oma=c(0,0,2,0))
> barplot(trigo.test2$lofactor10, main ="k=10", ylab = "LOF")
> barplot(trigo.test2$lofactor15, main ="k=15", ylab = "LOF")
> barplot(trigo.test2$lofactor20, main ="k=20", ylab = "LOF")
> barplot(trigo.test2$lofactor30, main ="k=30", ylab = "LOF")
> mtext("LOF trigo.test2 con lofactor()", outer=TRUE, cex=1.5)
```

Knn	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
5	0.6666667	0.652381	0.8666667	0.8609524
10	0.6666667	0.652381	0.8055556	0.7972222
15	0.6666667	0.652381	0.6984127	0.6854875
20	0.6666667	0.652381	0.7916667	0.7827381
30	0.6666667	0.652381	0.7777778	0.768254

Ejemplo 2. En este ejemplo se probara el método con un conjunto de datos de flores Iris [enlace], que contiene 50 observaciones de la variedad *setosa* y 3 de la variedad *virginica*:

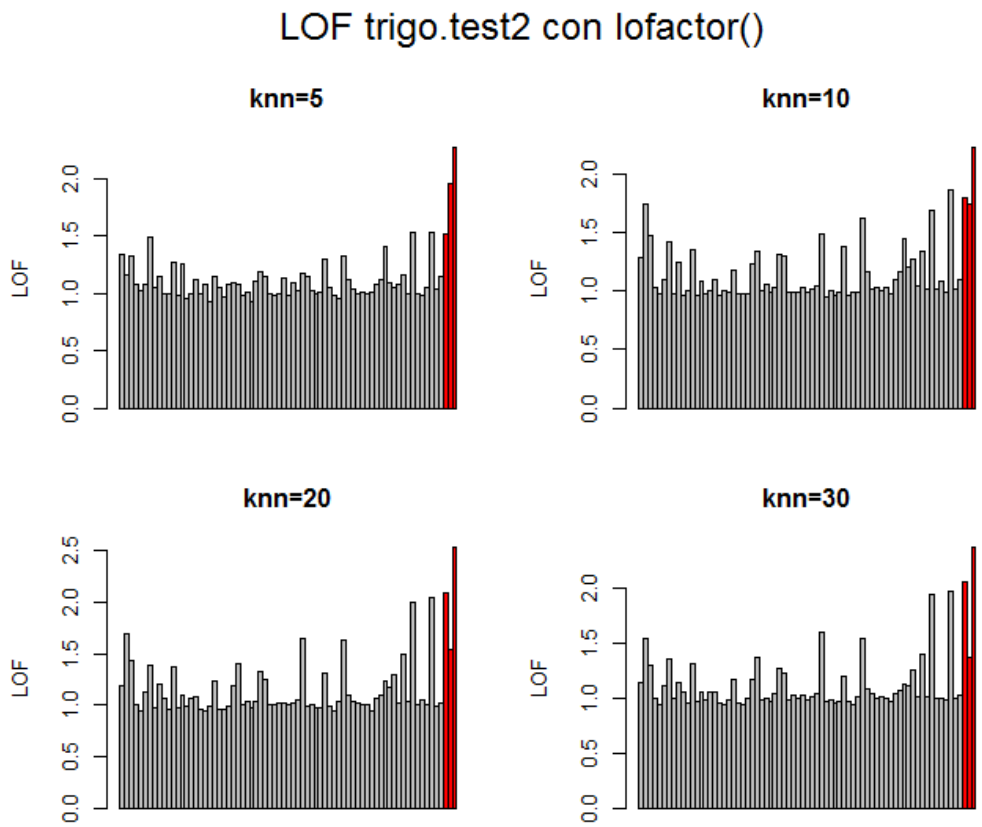


Figura 3.5: Clasificación semillas de trigo según parámetro `k`

```
> iris.test <- rbind(setosa, virginica[sample(1:50,3),])
> iris.test[,"lofactor10"] <- lofactor(iris.test[,1:4],10)
> iris.test[,"lofactor15"] <- lofactor(iris.test[,1:4],15)
> iris.test[,"lofactor20"] <- lofactor(iris.test[,1:4],20)
> iris.test[,"lofactor30"] <- lofactor(iris.test[,1:4],30)
> par(mfrow=c(2,2), oma=c(0,0,2,0))
> barplot(iris.test$lofactor10, main ="k=10", ylab = "LOF")
> barplot(iris.test$lofactor15, main ="k=15", ylab = "LOF")
> barplot(iris.test$lofactor20, main ="k=20", ylab = "LOF")
> barplot(iris.test$lofactor30, main ="k=30", ylab = "LOF")
> mtext("LOF iris.test con lofactor()", outer=TRUE, cex=1.5)
```

Knn	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
5	1	1	1	1
10	1	1	1	1
15	1	1	1	1
20	1	1	1	1
30	1	1	1	1
50	0.6666667	0.6466667	0.7142857	0.6971429

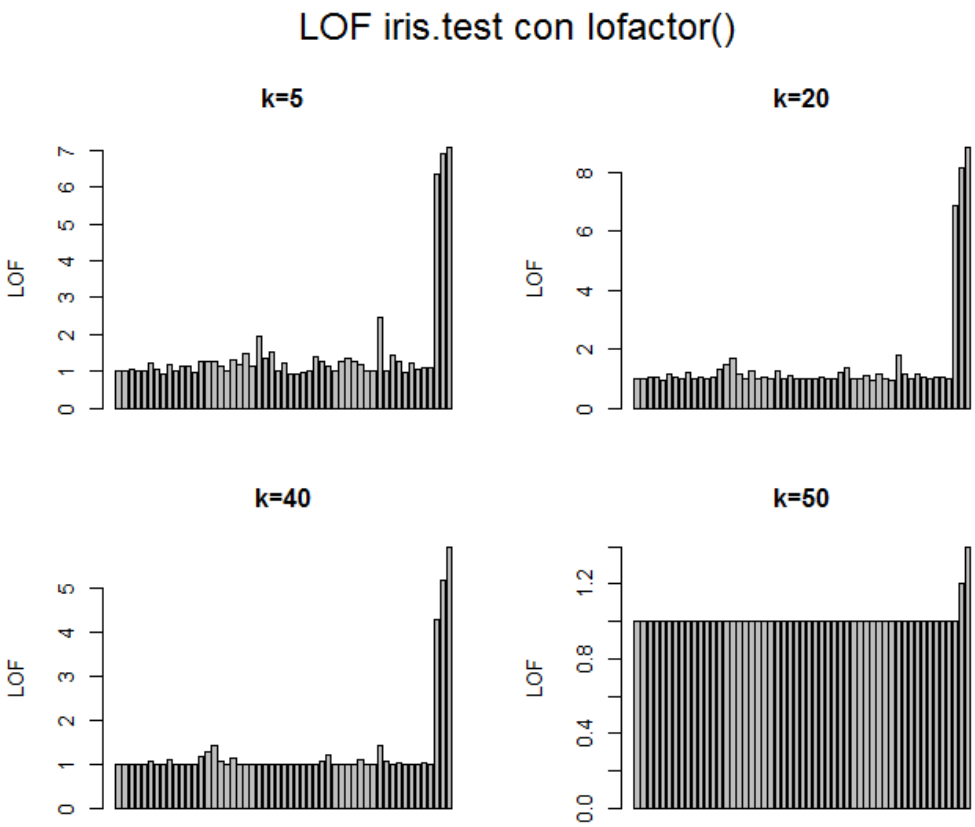


Figura 3.6: Clasificación de variedades de Iris Setosa y Virginica

Ejemplo 3. En este ejemplo utilizaremos una versión de un conjunto de datos sobre diferentes tipos de cáncer, benignos o malignos [cita original]. En esta versión [enlace versión], se ha reducido el número de outliers a 10 y se han eliminado observaciones duplicadas. El resultado es este conjunto con 223 objetos, 10 de ellos (2,2 %) anomalías:

```
> WBC.norm.v1[, "lofactor20"] <- lofactor(WBC.norm.v1[, 1:9], 20)
> WBC.norm.v1[, "lofactor40"] <- lofactor(WBC.norm.v1[, 1:9], 40)
> WBC.norm.v1[, "lofactor80"] <- lofactor(WBC.norm.v1[, 1:9], 80)
> WBC.norm.v1[, "lofactor110"] <- lofactor(WBC.norm.v1[, 1:9], 110)
> barplot(WBC.norm.v1$lofactor20, main ="knn=20", ylab = "LOF", col = ifelse
    (WBC.norm.v1$outlier=="yes", "red", "black"), border = NA, space = 0)
> barplot(WBC.norm.v1$lofactor40, main ="knn=40", ylab = "LOF", col = ifelse
    (WBC.norm.v1$outlier=="yes", "red", "black"), border = NA, space = 0)
> barplot(WBC.norm.v1$lofactor80, main ="knn=90", ylab = "LOF", col = ifelse
    (WBC.norm.v1$outlier=="yes", "red", "black"), border = NA, space = 0)
> barplot(WBC.norm.v1$lofactor110, main ="knn=110", ylab = "LOF", col =
    ifelse(WBC.norm.v1$outlier=="yes", "red", "black"), border = NA, space =
    0)
> mtext("LOF␣WBC.v1␣con␣lofactor()", outer=TRUE, cex=1.5)
```

Knn	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
20	0.1	0.05774648	0.1231274	0.08195971
30	0.2	0.1624413	0.2491161	0.2138634
40	0.4	0.371831	0.4390092	0.4126717
50	0.5	0.4765258	0.5786332	0.5588507
60	0.6	0.5812207	0.7273684	0.7145688
70	0.7	0.6859155	0.7991925	0.789765
80	0.7	0.6859155	0.8522084	0.8452698
90	0.7	0.6859155	0.8571535	0.8504471
100	0.8	0.7906103	0.8841529	0.878714
110	0.8	0.7906103	0.8966529	0.8918009
120	0.8	0.7906103	0.8863889	0.881055

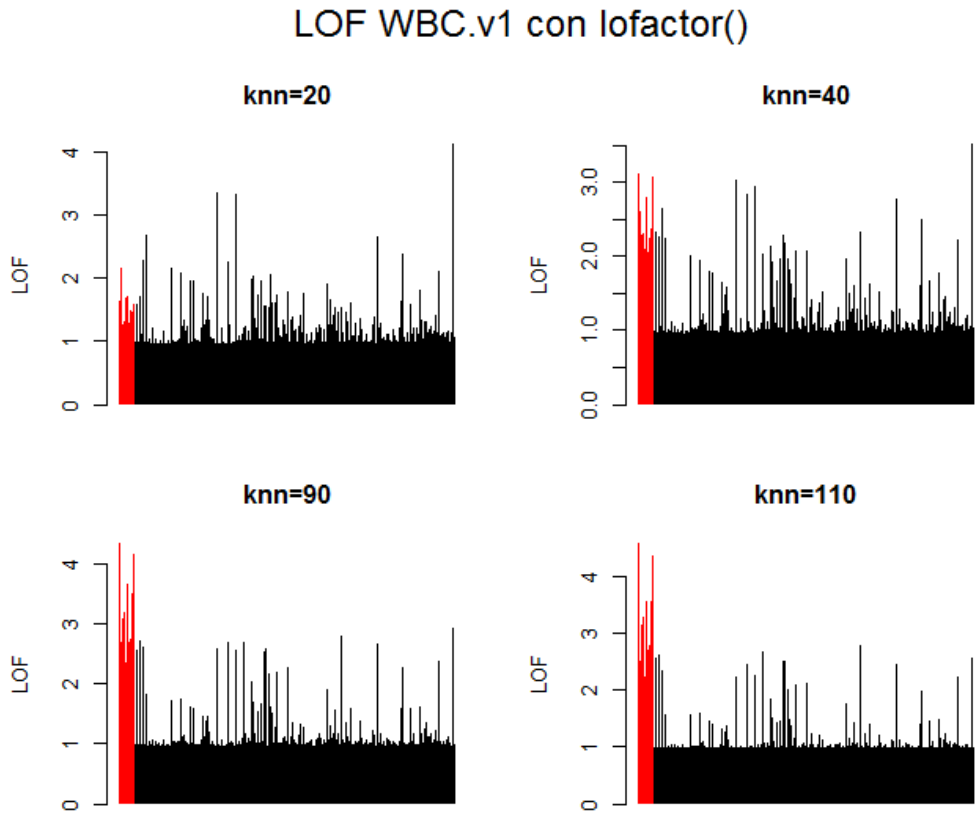


Figura 3.7: Factores de anomalía de diferente observaciones de cáncer

3.1.3. lof {RLOF}

Contenida en el paquete `Rlof`, la función es una implementación del algoritmo *LOF* que calcula los valores en base a múltiples valores k de forma paralela y también en base a distintas medidas de distancia aparte de la distancia euclidiana por defecto. Esta implementación es más rápida que la implementación de `lofactor`, originaria del paquete `dprep`, ya que hace uso de una estructura de datos y un cálculo de la distancia diferente.

```
lof(data, k, cores = NULL, ...)
```

La función devuelve una matriz `lof` que contiene los valores *LOF* de cada observación en cada fila para los diferentes valores k en cada columna.

Argumentos

`data`	El conjunto de datos, bien data.frame o una matriz.
`k`	Número de vecinos o elementos próximos utilizados en el cálculo de *LOF*. Puede ser un vector que contenga los múltiples valores k en base a los que *LOF* será calculado
`cores`	Este parámetro opcional especifica el número de núcleos a utilizar para el cálculo en paralelo. Si no es especificado, el máximo número de núcleos disponible serán utilizados.
`...`	Lista de parámetros transmitidos a la función `distmc()` del mismo paquete, que especifican la medida de distancia (`"euclidean"`, `"maximum"`, `"manhattan"`, `"canberra"`, `"binary"` or `"minkowski"`).

Ejemplos

Ejemplo 1. Para este ejemplo se analizara la presencia de anomalías en un conjunto de diferentes vidrios [cita]. El conjunto contiene 70 muestras de vidrio flotado y 9 de vidrio "de mesa":

```
> glass.test <- subset(glass.norm.v,type=="1"|type=="6")
> glass.test <- cbind(glass.test, lof(glass.test[,2:10], k=c(10, 15, 20, 25,
    30)))
> par(mfrow=c(2,2), oma=c(0,0,2,0))
> barplot(glass.test$`10`, main ="knn=10", ylab = "LOF",space=0)
> barplot(glass.test$`15`, main ="knn=15", ylab = "LOF",space=0)
> barplot(glass.test$`20`, main ="knn=20", ylab = "LOF",space=0)
> barplot(glass.test$`30`, main ="knn=30", ylab = "LOF",space=0)
> mtext("LOF glass.test con Rlof::lof()", outer=TRUE,cex=1.5)
```

Knn	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
10	0.5555556	0.4984127	0.7945437	0.7681279
15	0.8888889	0.8746032	0.9513889	0.9451389
20	0.8888889	0.8746032	0.9888889	0.9874603
25	0.7777778	0.7492063	0.8878205	0.8733974
30	0.5555556	0.4984127	0.7928673	0.7662359

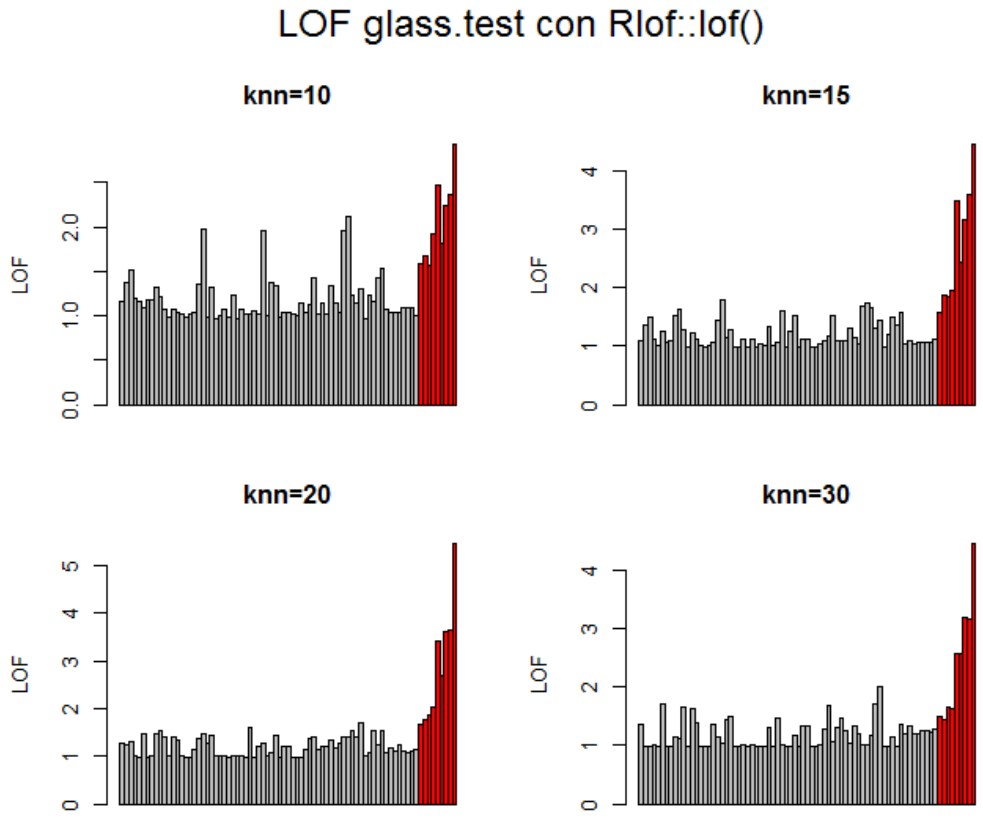


Figura 3.8: Factores LOF para diferentes parámetros KNN con `Rlof`

Ejemplo 2. Este conjunto contiene 70 medidas de semillas de trigo de la variedad *rosa* y 6 medidas de la variedad *kama* [cita]:

```
> trigo.test3 <- rbind(tr.rosa, tr.kama[sample(1:70,6),])
> trigo.test3 <- cbind(trigo.test3, lof(trigo.test3[,1:7],k=c
    (5,10,15,20,25,30)))
> par(mfrow=c(2,2), oma=c(0,0,2,0))
> barplot(trigo.test3$`5`, main ="k=5", ylab = "LOF", space = 0, col =
    ifelse(trigo.test3$type == "1", "red", "grey"))
> barplot(trigo.test3$`10`, main ="k=10", ylab = "LOF", space = 0, col =
    ifelse(trigo.test3$type == "1", "red", "grey"))
> barplot(trigo.test3$`15`, main ="k=15", ylab = "LOF", space = 0, col =
    ifelse(trigo.test3$type == "1", "red", "grey"))
> barplot(trigo.test3$`30`, main ="k=30", ylab = "LOF", space = 0, col =
    ifelse(trigo.test3$type == "1", "red", "grey"))
> mtext("LOF trigo.test3 con Rlof::lof()", outer=TRUE, cex=1.5)
```

Knn	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
5	0.5	0.4571429	0.6843585	0.6573035
10	0.6666667	0.6380952	0.7365544	0.7139733
15	0.5	0.4571429	0.6948052	0.6686456
20	0.6666667	0.6380952	0.6559163	0.6264234
25	0.6666667	0.6380952	0.6388889	0.6079365
30	0.6666667	0.6380952	0.7546569	0.7336275

Ejemplo 3. En este ejemplo utilizaremos una versión de un conjunto de datos sobre diferentes tipos de cáncer, benignos o malignos [cita original]. En esta versión [enlace versión], se ha reducido el número de outliers a 10 y se han eliminado observaciones duplicadas. El resultado es este conjunto con 223 objetos, 10 de ellos (2,2 %) anomalías:

```
> WBC.norm.v2 <- cbind(WBC.norm.v2, lof(WBC.norm.v2[,1:9],k=c(20,40,60,80)))
> par(mfrow=c(2,2), oma=c(0,0,2,0))
> barplot(WBC.norm.v2$`20`, main ="knn = 20", ylab = "LOF", col = ifelse(WBC
    .norm.v1$outlier=="yes","red","black"), border = NA, space=0)
> barplot(WBC.norm.v2$`60`, main ="knn = 60", ylab = "LOF", col = ifelse(WBC
    .norm.v1$outlier=="yes","red","black"), border = NA, space=0)
> barplot(WBC.norm.v2$`80`, main ="knn = 80", ylab = "LOF", col = ifelse(WBC
    .norm.v1$outlier=="yes","red","black"), border = NA, space=0)
> barplot(WBC.norm.v2$`120`, main ="knn = 120", ylab = "LOF", col = ifelse(
    WBC.norm.v1$outlier=="yes","red","black"), border = NA, space=0)
> mtext("LOF WBC.norm.v2 con Rlof::lof", outer=TRUE, cex=1.5)
```

Knn	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
20	0.1	0.05774648	0.2193434	0.1826928
40	0.6	0.5812207	0.6835839	0.6687287
60	0.7	0.6859155	0.8577273	0.8510478
80	0.7	0.6859155	0.8557018	0.8489272
100	0.7	0.6859155	0.8446078	0.8373124
110	0.7	0.6859155	0.8517507	0.8447906
120	0.7	0.6859155	0.8568711	0.8501515

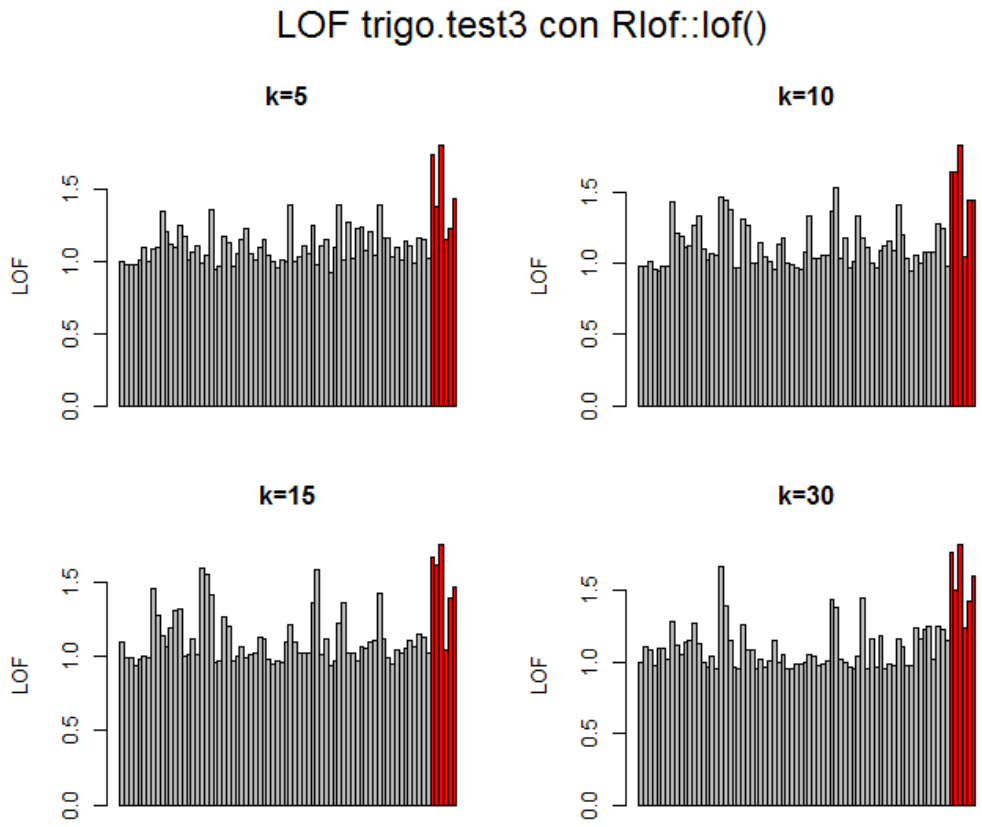


Figura 3.9: Factores de anomalía para diferentes parámetros KNN con `Rlof`

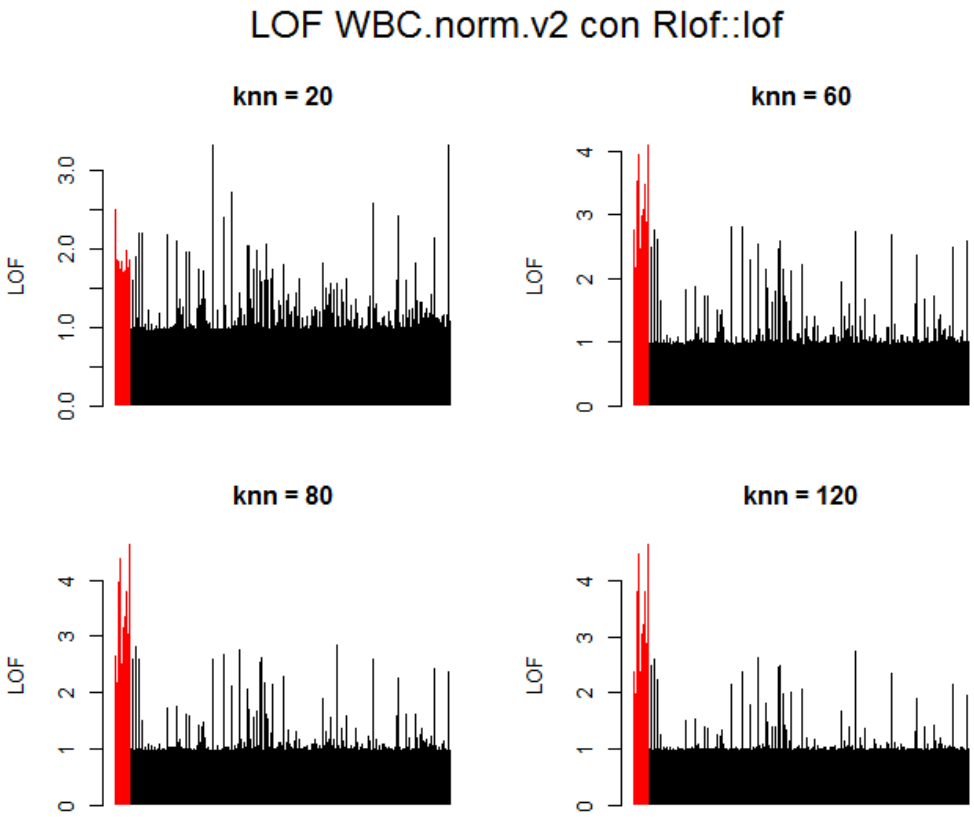


Figura 3.10: Factores de anomalía para diferentes parámetros KNN con `Rlof`

3.2. ABOD: Angle Based Outlier Detection

ABOD (*Angle Based Outlier Detection*) es un algoritmo de enfoque global que asigna a cada objeto un factor de outlier obtenido mediante la evaluación de la varianza de los ángulos entre los vectores de un punto a otros puntos y, en menor manera, a partir de la distancia que los separa. Comparando los ángulos entre pares de vectores distancia a otros objetos se logra discernir entre objetos pertenecientes a un grupo y objetos anómalos.

La necesidad de un enfoque diferente a los típicos, basados en la evaluación de las diferentes distancias entre los objetos, o basados en consultas de ε-rango o consultas de k-vecinos-cercanos para métodos locales; viene de que estos no se adaptan bien a conjuntos de datos con un gran número de dimensiones, el conocido problema de "*la maldición de la dimensión*". Al aumentar el número de dimensiones d, el contraste relativo entre el punto más alejado y el más cercano converge a0. Esto quiere decir que las funciones que miden la distancia entre dos puntos pierden su utilidad en espacios con gran número de dimensiones.

Con este enfoque, la distancia sigue siendo incluida en el cálculo, pero solo como medida secundaria con el fin de normalizar el resultado. El concepto básico se basa en que, para un punto dentro de una agrupación, los ángulos entre pares de vectores a otros puntos difieren notablemente, aunque la varianza de los ángulos en los puntos de los bordes de la agrupación aumente. En cambio, la varianza para un punto realmente anómalo aumenta drásticamente. En estos, los ángulos a la mayoría de pares de puntos serán pequeños porque estarán agrupados en la misma dirección.

Si el rango de ángulos observados para un punto es amplio, el punto estará rodeado de otros puntos en todas las direcciones posibles, lo que significa que está situado dentro de un clúster. Si, al contrario, el rango de ángulos observados es pequeño, los otros puntos se encuentran sólo en ciertas direcciones, lo que significa que el punto está situado fuera de conjuntos de puntos que están agrupados entre ellos.

Para asignar el $ABOF$ a cualquier objeto en conjunto de datos D, calculamos el producto escalar de los vectores diferencia para cualquier tripleta de puntos (esto es, para un punto $\vec{A} \in D$ y todos los pares $(\vec{B}, \vec{C})$ para todos los puntos restantes en $D \setminus \{\vec{A}\}$) normalizado por el producto cuadrático de la longitud de los vectores diferencia. Esto quiere decir que el ángulo es ponderado de menor manera si los puntos correspondientes están lejos del punto de la consulta. Mediante este factor de ponderación, la distancia influye en el valor a pesar de todo, pero de una manera mucho menos significativa. De cualquier modo, esta ponderación de la varianza es importante ya que el ángulo para un par de puntos varía de manera más significativa para distancias

más grandes.

La varianza de este valor sobre todos los pares para el objeto A de la consulta constituye el factor de anomalía basado en ángulo de A:

Dada una base de datos $\mathcal{D} \subseteq \mathbb{R}^d$ *un punto* $\overrightarrow{A} \in \mathcal{D}$ *y una norma* $\|.\| : \mathbb{R}^d \to R_0^+$. *El producto escalar denotado por* $\langle .,. \rangle$*:* $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. *Para dos puntos* $\overrightarrow{B}, \overrightarrow{C} \in \mathcal{D}$, $\overline{BC}$ *denota el vector diferencia* $\overrightarrow{C} - \overrightarrow{B}$. *El* factor angular de anomalía $ABOF(\overrightarrow{A})$ *es la varianza de los ángulos entre los diferentes vectores de* $\overrightarrow{A}$ *a todos los puntos en* $\mathcal{D}$ *ponderada por la distancia a los puntos:*

$$\begin{aligned} ABOF(\vec{A}) &= VAR_{\vec{B},\vec{C} \in D} \left(\frac{\langle \overline{AB}, \overline{AC} \rangle}{\|\overline{AB}\|^2 \cdot \|\overline{AC}\|^2} \right) \\ &= \frac{\sum\limits_{\vec{B} \in D} \sum\limits_{\vec{C} \in D} \left(\frac{1}{\|\overline{AB}\| \cdot \|\overline{AC}\|} \frac{\langle \overline{AB}, \overline{AC} \rangle}{\|\overline{AB}\|^2 \cdot \|\overline{AC}\|^2} \right)^2}{\sum\limits_{\vec{B} \in D} \sum\limits_{\vec{C} \in D} \frac{1}{\|\overline{AB}\| \cdot \|\overline{AC}\|}} \\ &\quad - \left(\frac{\sum\limits_{\vec{B} \in D} \sum\limits_{\vec{C} \in D} \frac{1}{\|\overline{AB}\| \cdot \|\overline{AC}\|} \frac{\langle \overline{AB}, \overline{AC} \rangle}{\|\overline{AB}\|^2 \cdot \|\overline{AC}\|^2}}{\sum\limits_{\vec{B} \in D} \sum\limits_{\vec{C} \in D} \frac{1}{\|\overline{AB}\| \cdot \|\overline{AC}\|}} \right)^2 \end{aligned} \tag{3.9}$$

El algoritmo $ABOD$ asigna el factor angular de anomalía a cada punto en el conjunto de datos y devuelve una lista ordenada con todos los puntos ordenados por su $ABOF$. Ya que la distancia es sólo es considerada como una ponderación del criterio principal, la varianza de los ángulos, $ABOD$ es capaz de detectar anomalías incluso es espacios de gran dimensionalidad donde otros enfoques basados o relacionados con la distancia, como LOF, pueden perder precisión. Además, $ABOD$ no requiere de parámetros especificados por el usuario que sean cruciales para el resultado del algoritmo.

El problema con este algoritmo $ABOD$ es su complejidad computacional. Ya que por cada punto en el conjunto de datos $\mathcal{D}$ todos los pares de puntos deben ser tenidos en cuenta, la complejidad temporal del algoritmo se eleva a $O(n^3)$, que resulta muy grande comparada, por ejemplo, con la complejidad temporal de LOF que es $O(n^2 \cdot k)$ en el peor de los casos. Para resolver este problema los autores proponen tres variantes de este algoritmo para aproximar el valor de $ABOF$, pero no se discutirán ya que no están disponibles en ninguna librería de R.

Para resolver este problema, los autores proponen un algoritmo para aproximar el factor de anomalía angular $ABOF$, mediante una muestra del conjunto, llamado $FastABOD$. Para ello, se utilizan los pares de puntos con mayor importancia en la varianza, por ejemplo, pares de objetos entre los k-vecinos-cercanos, aunque también se puede utilizar un conjunto aleatorio de$k$ objetos; pero en el primer conjunto tenemos los objetos con mayor ponderación en el $ABOF$.

El factor angular de anomalía quedaría como:
Dada una base de datos $\mathcal{D} \subseteq \mathbb{R}^d$ *un punto* $\overrightarrow{A} \in \mathcal{D}$ *y una norma* $\|.\| : \mathbb{R}^d \to R_0^+$. *El producto escalar denotado por* $\langle .,. \rangle$*:* $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. *Para dos puntos* $\overrightarrow{B}, \overrightarrow{C} \in \mathcal{D}$, $\overline{BC}$ *denota el vector diferencia* $\overrightarrow{C} - \overrightarrow{B}$. $\mathcal{N}_k(\overrightarrow{A}) \subseteq \mathcal{D}$ *denota el conjunto de k vecinos más cercanos de* $\overrightarrow{A}$. *El* factor angular de anomalía aproximado $aproxABOF(\overrightarrow{A})$ *es la varianza de los ángulos entre los diferentes vectores de* $\overrightarrow{A}$ *a todos los puntos en* $\mathcal{N}_k(\overrightarrow{A})$ *ponderada por la distancia a los puntos:*

$$
\begin{aligned}
aproxABOF_k(\vec{A}) &= VAR_{\vec{B},\vec{C} \in \mathcal{N}_k(\overrightarrow{A})} \left(\frac{\langle \overline{AB}, \overline{AC} \rangle}{\|\overline{AB}\|^2 \cdot \|\overline{AC}\|^2} \right) \\
&= \frac{\displaystyle\sum_{\vec{B} \in \mathcal{N}_k(\overrightarrow{A})} \sum_{\vec{C} \in \mathcal{N}_k(\overrightarrow{A})} \left(\frac{1}{\|\overline{AB}\| \cdot \|\overline{AC}\|} \frac{\langle \overline{AB}, \overline{AC} \rangle}{\|\overline{AB}\|^2 \cdot \|\overline{AC}\|^2} \right)^2}{\displaystyle\sum_{\vec{B} \in \mathcal{N}_k(\overrightarrow{A})} \sum_{\vec{C} \in \mathcal{N}_k(\overrightarrow{A})} \frac{1}{\|\overline{AB}\| \cdot \|\overline{AC}\|}} \\
&\quad - \left(\frac{\displaystyle\sum_{\vec{B} \in \mathcal{N}_k(\overrightarrow{A})} \sum_{\vec{C} \in \mathcal{N}_k(\overrightarrow{A})} \frac{1}{\|\overline{AB}\| \cdot \|\overline{AC}\|} \frac{\langle \overline{AB}, \overline{AC} \rangle}{\|\overline{AB}\|^2 \cdot \|\overline{AC}\|^2}}{\displaystyle\sum_{\vec{B} \in \mathcal{N}_k(\overrightarrow{A})} \sum_{\vec{C} \in \mathcal{N}_k(\overrightarrow{A})} \frac{1}{\|\overline{AB}\| \cdot \|\overline{AC}\|}} \right)^2
\end{aligned}
\tag{3.10}
$$

Mediante esta aproximación se obtienen resultados con una velocidad un orden de magnitud menor. El algoritmo resultante $FastABOD$ tiene una complejidad temporal de $O(n^2 + n \cdot k^2)$, que lo hace apropiado a conjuntos con muchos puntos. Por otra parte, la calidad de la aproximación dependerá del conjunto elegido para cada punto, es decir, del número k de vecinos más cercanos y la calidad de esta selección de objetos próximos, o del tamaño y objetos presentes en la muestra aleatoria.

"Angle-Based Outlier Detection in High-dimensional Data" Hans-Peter Kriegel, Peer Kroger, Erich Schubert, and Arthur Zimek

3.2.1. FUNC.ABOD {HighDimOut}

3.2.2. abod {abodOutlier}

3.3. SOD Subspace Outlier Degree

El algoritmo *SOD* ("*Subspace Outlier Degree*" o *grado de anomalía subespacio*), propuesto en "*Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data*" [cita], es una propuesta para la búsqueda de valores anómalos en espacios de datos con gran número de dimensiones, enfocada a la búsqueda de anomalías

El algoritmo intenta solucionar dos problemas de la búsqueda de objetos aberrantes relacionados con la maldición de la dimensión. Primero, en espacios de muchas dimensiones, las distancias euclidianas (y otras medidas L_p-norma) no pueden ser utilizadas para diferenciar puntos de otros, porque todos son más o menos equidistantes entre ellos, y, a consecuencia, es difícil detectar una anomalía que se aleje considerablemente de la mayoría de puntos. Segundo, puede haber mecanismos concretos que hayan generado los datos pero, normalmente, para cada mecanismo sólo un subconjunto de características son importantes (este problema se conoce como *relevancia local de la característica* o "*local feature relevance*"), y por lo tanto la detección de anomalías tiene sentido sólo cuando se consideran los subconjuntos de características relevantes de estos mecanismos generadores, es decir, subespacios del espacio de datos original.

La idea del algoritmo *SOD* es la de realizar la búsqueda sólo en subconjuntos de datos que incluyan las características más importantes de los mecanismos que han generado los datos. La búsqueda de anomalías en subespacios es interesante en conjuntos de gran número de dimensiones donde podemos prever una distribución uniforme teniendo en cuenta todas las dimensiones pero distribuciones más peculiares, incluyendo anomalías, en subespacios. Ya que estos subespacios de características significativas no son conocidos a priori, la búsqueda de anomalías debe ser precedida por la búsqueda de subespacios relevantes.

Para el modelo de anomalías en subespacios, la idea es analizar cada punto, y de qué manera encaja en el subespacio que es generado por un conjunto de puntos de referencia. El subespacio generado es simplemente un hiperplano de ejes paralelos de cualquier dimensionalidad $l < d$ (donde d es el número de dimensiones original del espacio de datos) tal que todos los puntos del conjunto de referencia se encuentren cercanos a este hiperplano. Si un punto se aleja significativamente del hiperplano de referencia, es considerado una anomalía en el subespacio que es perpendicular a ese hiperplano.

A continuación, se considera que $\mathcal{D} \subseteq \mathbb{R}^d$ es una base de datos de n puntos en un espacio característico de d-dimensiones y $dist$ es una función de medida de distancia en los puntos en $\mathcal{D}$, por ejemplo la similitud coseno o una L_p-norma. Para cualquier punto $p \in \mathbb{R}^d$, denominamos la proyección

de p en i con p_i

El *subespacio hiperplano* de un conjunto de puntos S (el conjunto de referencia) captura el subespacio en el cual la varianza de los puntos de S es alta, mientras que en el subespacio perpendicular, la varianza de los puntos en S es baja. La *varianza* $VAR^S \in \mathbb{R}$ de S es la media de las distancias al cuadrado de los puntos de S al valor medio ν^S, esto es $VAR^S = \frac{\sum_{p \in S} dist(p,\mu^S)^2}{Card(S)}$, donde $Card(S)$ es la cardinalidad del conjunto S. La varianza a lo largo de un atributo i, denominada como var_i^S de S es definida como $var_i^S = \frac{\sum_{p \in S}(dist(p_i,\mu_i^S))^2}{Card(S)}$

Sea $R(p) \subseteq D$ un conjunto de puntos de referencia para $p \in \mathcal{D}$ con respecto al cual se medirá la discrepancia de p. El vector subespacio $v^{R(p)} \in \mathbb{R}^d$ de un conjunto de referencia $R(p)$ especifica los atributos relevantes del subespacio definido por el conjunto, es decir, los atributos donde los puntos del conjunto $R(p)$ exhiben una varianza pequeña. Para diferenciar entre una varianza pequeña y grande, se aplica el siguiente criterio: en los d atributos, los puntos tienen una varianza total de $VAR^{R(p)}$, por lo que la varianza esperada a lo largo de cada atributo i es $1/d \cdot VAR^{R(p)}$. Evaluamos la varianza en ese atributo como pequeña si $var_i^{R(p)}$ es menor que la varianza esperada en un factor predefinido α. Por cada atributo donde $R(p)$ exhibe una varianza pequeña, el correspondiente valor del vector subespacio $v^{R(p)}$ es puesto a 1, y para el resto de atributos a 0. Formalmente:

$$v_i^{R(p)} = \begin{cases} 1 & \text{if} \quad var_i^{R(p)} < \alpha \frac{VAR^{R(p)}}{d} \\ 0 & \text{else.} \end{cases} \tag{3.11}$$

El *subespacio hiperplano* $\mathcal{H}(R(p))$ de $R(p)$ es definido por una tupla del valor de la media $\mu^{R(p)}$ de $R(p)$ y el vector subespacio $v^{R(p)}$ de $R(p)$, tal que $\mathcal{H}(R(p)) = (\mu^{R(p)}, v^{R(p)})$. De esta manera es posible medir cuanto se desvía p del hiperplano generado por su conjunto de referencia. La desviación de cualquier punto o a un hiperplano $\mathcal{H}(S)$ es definida por la distancia euclidiana en el subespacio que es perpendicular al hiperplano. Esta se puede computar simplemente utilizando la distancia euclidiana ponderada entre o y μ^S utilizando el vector subespacio v^S como valor de ponderación, esto es:

$$dist(o, \mathcal{H}(S)) = \sqrt{\sum_{i=1}^{d} v_i^S \cdot (o_i, \mu_i^S)^2} \tag{3.12}$$

Este valor de distancia una medida intuitiva como factor de anomalía de cualquier punto $p \in \mathcal{D}$ con respecto a un conjunto de referencia. Un valor cercano a 0 indica que el punto en cuestión encaja muy bien en el hiperplano

$\mathcal{H}(R(p))$, es decir, que no es una *outlier*, mientras que un valor considerablemente mayor indica que p es un outlier.

Finalmente, la definición del factor subespacio de anomalía es:

Sea $R(p)$ un conjunto de objetos de referencia para el objeto $p \in \mathcal{D}$. El factor subespacio de anomalía (SOD) de p con respecto a $R(p)$, escrito $SOD_{R(p)}(p)$, viene dado por

$$SOD_{R(p)}(p) := \frac{dist(o, \mathcal{H}(R(p)))}{\|v^{R(p)}\|} \tag{3.13}$$

Esto es, la distancia entre el punto p y su conjunto de referencia $R(p)$ de acuerdo con la ecuación [ecuacion], normalizado por el número de dimensiones relevantes dado por el número de elementos $v_i^{R(p)} = 1$ en el factor de ponderación definido en [ecuacion]

Para la elección del conjunto de referencia relevante para un punto dado $p \in \mathcal{D}$ se utiliza la *similitud de vecinos más cercanos SNN*, debido a los problemas ya mencionados con distintas medidas de distancia por el número de dimensiones. Un enfoque SNN normalmente mide la similaridad de dos objetos basándose en el número de vecinos más cercanos en común. Formalmente, sea $N_k(p) \subseteq \mathcal{D}$ el conjunto k-vecinos cercanos de $p \in \mathcal{D}$ con respecto a la medida de distancia $dist$. La *similitud de vecinos mas cercanos compartidos* entre dos puntos $p, q \in \mathcal{D}$ es definida como $sim_{SNN}(p, q) = Card(N_k(p) \bigcap N_k(q))$. Con esto, el conjunto de referencia $R(p)$ de p es el conjunto de l-vecinos más cercanos de p utilizando $simSNN$, esto es, un subconjunto de $\mathcal{D}$ que contiene l puntos de acuerdo con la siguiente condición: $\forall o \in R(p), \forall \hat{o} \in \mathcal{D} \setminus R(p) : sim_{SNN}(\hat{o}, p) \leq sim_{SNN}(o, p)$.

El algoritmo SOD depende de dos parámetros de entrada. Primeramente, k específica el número de vecinos más cercanos que son incluidos para calcular la similitud de vecinos más cercanos. No es un parámetro crítico pero debe ser lo suficientemente grande para abarcar los puntos de un mismo mecanismo generador.

Segundo, l especifica el tamaño del conjunto de referencia. Debería ser suficientemente grande por la misma razón y, obviamente, menor o igual a k. Podemos considerar un tercer parámetro, α, que especifica el límite para decidir sobre la importancia de un atributo. En los experimentos de la publicación original, los autores recomiendan $\alpha = 0.8$ debido a los buenos resultados obtenidos de manera consistente.

Para calcular el SOD, primero se debe encontrar el conjunto de k-vecinos-cercanos de cada uno de los n puntos de la base de datos, lo que requiere $O(d \cdot n^2)$ en el peor de los casos. Esto puede ser reducido a $O(d \cdot n \log n)$ si se utiliza un índice para realizar las consultas de vecinos cercanos. Después,

para cada punto p, el conjunto de referencia de p formado por los l vecinos más cercanos de p con respecto a la similitud SNN ha de ser hallado, lo que lleva $O(k \cdot n)$, la media y la varianza de este conjunto de referencia han de ser calculadas, lo que lleva $O(d \cdot l)$, y, por último, se calcula el grado de anomalía SOD. En resumen, como $k \ll n$ y $l \ll n$, la complejidad temporal de la ejecución del algoritmo en general es de $O(d \cdot n^2)$, complejidad comparable a la mayoría de algoritmos de detección de anomalías.

3.3.1. Func.SOD {HighDimOut}

Esta función, del paquete `HighDimOut`, lleva a cabo el algoritmo de detección de anomalías en subespacios SOD.

Listing 3.2: llamada por defecto

```
Func.SOD(data, k.nn, k.sel, alpha = 0.8)
```

La función devuelve un vector numérico que contiene los valores grado de anomalía subespacio SOD para cada objeto.

Argumentos

`data`	El conjunto de datos `data.frame`
`k.nn`	Valor numérico especifica el número de vecinos cercanos para calcular SNN. Debe ser mayor que `k.sel`
`k.sel`	Valor numérico especifica el número de vecinos cercanos compartidos. Puede interpretarse como el número de conjuntos de referencia para construir el hiperplano subespacio
`alpha`	Especifica el límite inferior para elegir un subespacio. El valor 0,8 por defecto es el sugerido en la publicación original

Ejemplos

Ejemplo 1. En este ejemplo se analizara un conjunto de datos de lecturas de radar sobre señales retornadas de la ionosfera [cita]. Este conjunto de datos diferencia entre "buenas" lecturas, las cuales muestran evidencia de algún tipo de estructura en la ionosfera; y "malas" lecturas, las cuales atraviesan la ionosfera. Contiene 351 observaciones, de las cuales 126 son malas (35,9 %) y 225 son buenas (64,1 %):

```
> ionosphere.SOD[,"SOD10"] <- Func.SOD(ionosphere.SOD[,3:34], k.nn=10, k.sel
    =5)
> ionosphere.SOD[,"SOD15"] <- Func.SOD(ionosphere.SOD[,3:34], k.nn=15, k.sel
    =5)
> ionosphere.SOD[,"SOD15v"] <- Func.SOD(ionosphere.SOD[,3:34], k.nn=15, k.
    sel=8)
> ionosphere.SOD[,"SOD20"] <- Func.SOD(ionosphere.SOD[,3:34], k.nn=20, k.sel
    =5)
> ionosphere.SOD[,"SOD20v"] <- Func.SOD(ionosphere.SOD[,3:34], k.nn=20, k.
    sel=10)
> ionosphere.SOD[,"SOD30"] <- Func.SOD(ionosphere.SOD[,3:34], k.nn=30, k.sel
    =5)
> barplot(ionosphere.SOD$SOD10, main = "knn = 10", ylab = "Factor Anomalía",
     space = 0, border = NA, col = ifelse(ionosphere.SOD$X35 == "b", "red",
    "black"))
> barplot(ionosphere.SOD$SOD15, main = "knn = 15", ylab = "Factor Anomalía",
     space = 0, border = NA, col = ifelse(ionosphere.SOD$X35 == "b", "red",
    "black"))
> barplot(ionosphere.SOD$SOD20, main = "knn = 20", ylab = "Factor Anomalía",
     space = 0, border = NA, col = ifelse(ionosphere.SOD$X35 == "b", "red",
    "black"))
> barplot(ionosphere.SOD$SOD30, main = "knn = 30", ylab = "Factor Anomalía",
     space = 0, border = NA, col = ifelse(ionosphere.SOD$X35 == "b", "red",
    "black"))
```

k.nn	k.sel	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
10	5	0.7698413	0.6409524	0.8805208	0.8136125
15	5	0.7857143	0.6657143	0.8817474	0.8155259
15	8	0.7777778	0.6533333	0.8821721	0.8161884
20	5	0.7777778	0.6533333	0.8883668	0.8258523
20	10	0.7936508	0.6780952	0.885884	0.8219791
30	15	0.7539683	0.6161905	0.8607437	0.7827602

Ejemplo 2. Para este ejemplo se analizara la presencia de anomalías en un conjunto de diferentes vidrios [cita]. El conjunto contiene 70 muestras de vidrio flotado y 9 de vidrio "de mesa":

```
> barplot(glass.norm.test$SOD10, main = "knn = 10", ylab = "Factor Anomalía"
    , space = 0, col = ifelse(glass.norm.test$type == "6", "red", "grey"))
> barplot(glass.norm.test$SOD20, main = "knn = 20", ylab = "Factor Anomalía"
    , space = 0, col = ifelse(glass.norm.test$type == "6", "red", "grey"))
> barplot(glass.norm.test$SOD25v, main = "knn = 25", ylab = "Factor Anomalía
    ", space = 0, col = ifelse(glass.norm.test$type == "6", "red", "grey"))
> barplot(glass.norm.test$SOD30v, main = "knn = 30", ylab = "Factor Anomalía
    ", space = 0, col = ifelse(glass.norm.test$type == "6", "red", "grey"))
> mtext("Factor anomalía glass.test con Func.SOD", outer=TRUE)
```

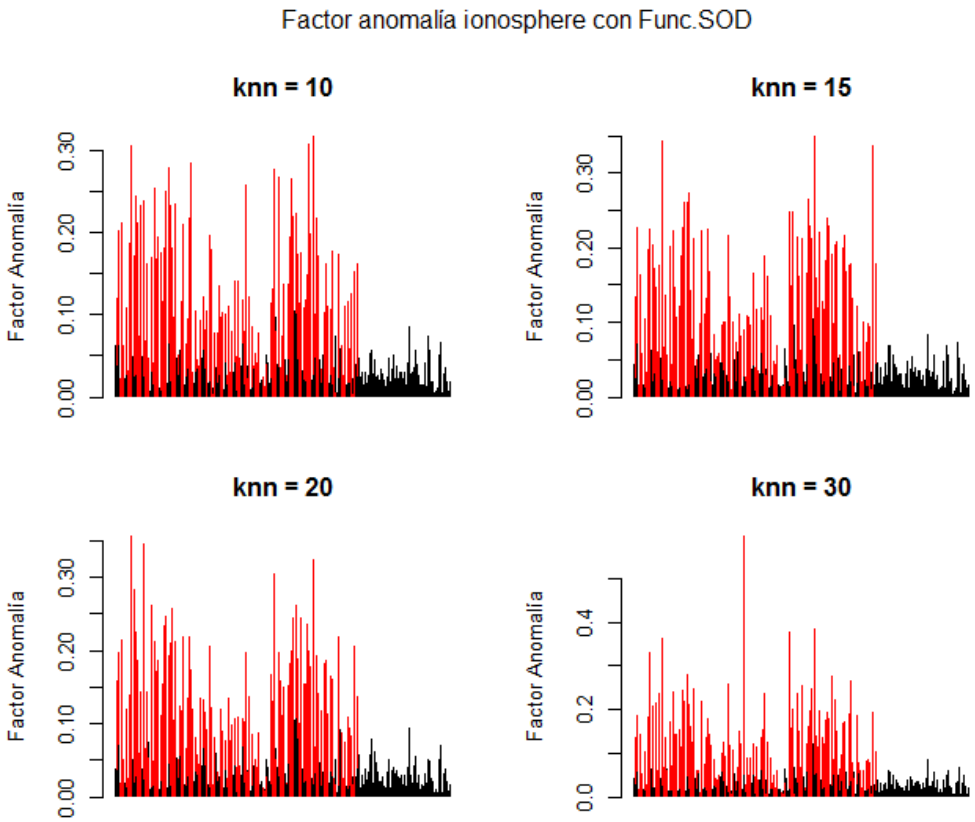


Figura 3.11: Clasificación mediante *SOD* con diferentes parámetros `k.nn`

`k.nn`	`k.sel`	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
10	5	0.3333333	0.247619	0.3024182	0.2127291
15	7	0.1111111	-0.003174603	0.2008967	0.09815483
15	5	0.1111111	-0.003174603	0.2415986	0.1440898
20	10	0.4444444	0.3730159	0.5356267	0.4759216
20	12	0.4444444	0.3730159	0.4707703	0.4027265
20	5	0.2222222	0.1222222	0.3285927	0.2422689
25	12	0.4444444	0.3730159	0.4212859	0.3468798
25	17	0.4444444	0.3730159	0.6527273	0.6080779
25	20	0.5555556	0.4984127	0.6494529	0.6043825
30	15	0.4444444	0.3730159	0.5425019	0.4836807
30	20	0.5555556	0.4984127	0.6550265	0.6106727

Ejemplo 3. Este conjunto contiene 70 medidas de semillas de trigo de la variedad *canadian* y 3 medidas de la variedad *kama* [cita]:

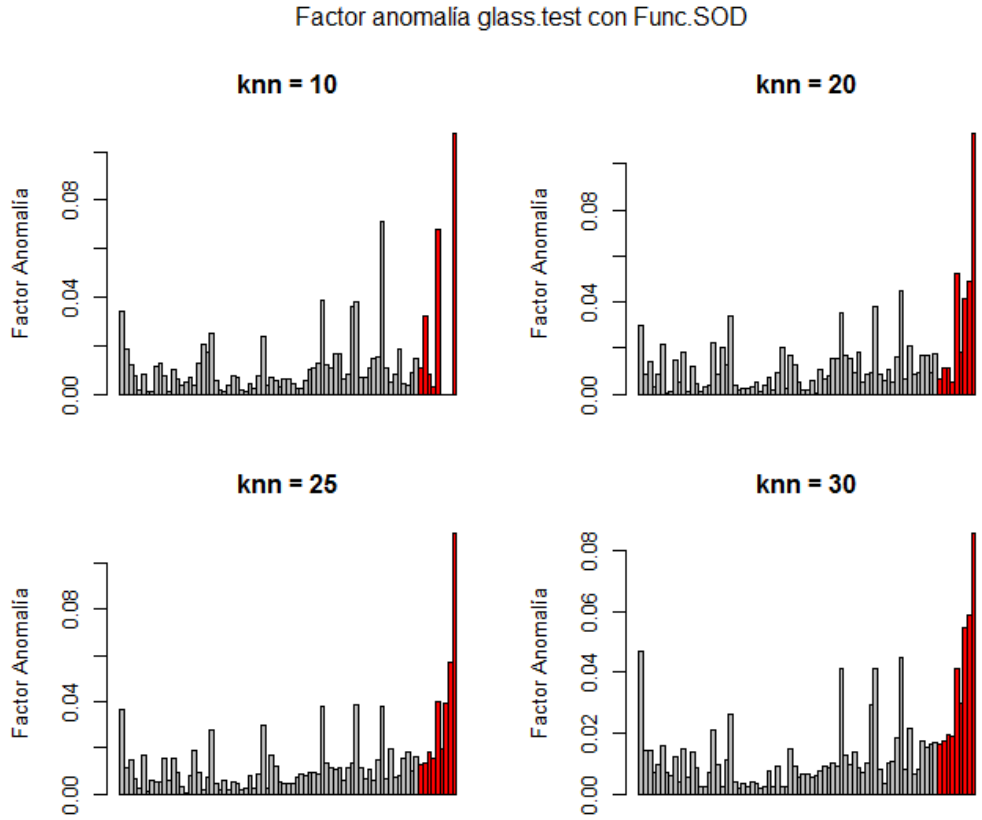


Figura 3.12: Clasificación mediante *SOD* con diferentes parámetros `k.nn` y `k.sel`

```
> trigo.test2 <- rbind(tr.canadian, tr.kama[sample(1:70,3),])
> trigo.test2["SOD10"] <- Func.SOD(trigo.test2[1:7], k.nn = 10, k.sel = 3)
> trigo.test2["SOD15v2"] <- Func.SOD(trigo.test2[1:7], k.nn = 15, k.sel = 5)
> trigo.test2["SOD20v2"] <- Func.SOD(trigo.test2[1:7], k.nn = 20, k.sel = 5)
> trigo.test2["SOD25"] <- Func.SOD(trigo.test2[1:7], k.nn = 20, k.sel = 5)
> barplot(trigo.test2$SOD10, main = "knn = 10", ylab = "Factor Anomalía",
    space = 0, col = ifelse(trigo.test2$type == "1", "red", "grey"))
> barplot(trigo.test2$SOD15v2, main = "knn = 15", ylab = "Factor Anomalía",
    space = 0, col = ifelse(trigo.test2$type == "1", "red", "grey"))
> barplot(trigo.test2$SOD20v2, main = "knn = 20", ylab = "Factor Anomalía",
    space = 0, col = ifelse(trigo.test2$type == "1", "red", "grey"))
> barplot(trigo.test2$SOD25, main = "knn = 25", ylab = "Factor Anomalía",
    space = 0, col = ifelse(trigo.test2$type == "1", "red", "grey"))
> mtext("Factor anomalía trigo.test2 con Func.SOD", outer=TRUE)
```

k.nn	k.sel	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
5	2	0.3333333	0.3047619	0.5434783	0.523913
10	3	0.3333333	0.3047619	0.5	0.4785714
10	5	0.6666667	0.652381	0.6899225	0.6766334
15	5	1	1	1	1
15	7	0.6666667	0.652381	0.5833333	0.5654762
15	10	0.3333333	0.3047619	0.5263158	0.506015
20	5	0.6666667	0.652381	0.8666667	0.8609524
20	10	0.3333333	0.3047619	0.4936937	0.4719949
20	15	0.3333333	0.3047619	0.5322581	0.512212
25	5	0.6666667	0.652381	0.8055556	0.7972222

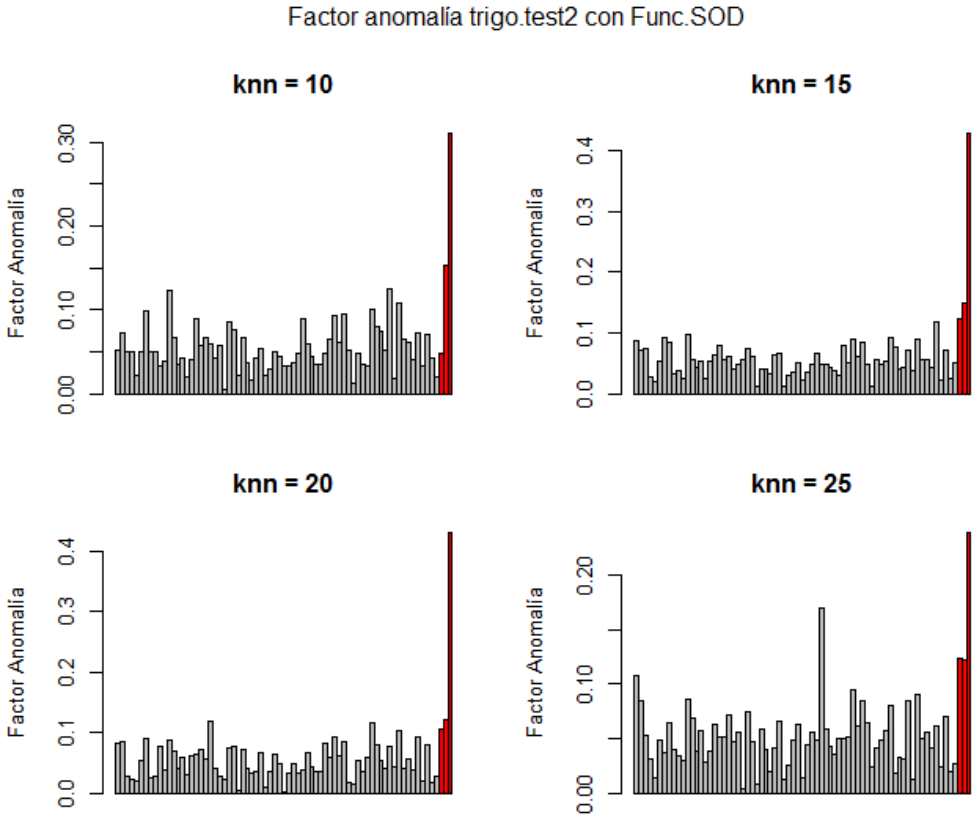


Figura 3.13: Clasificación por *SOD* con diferentes `k.nn` y `k.sel`

Ejemplo 4. En este ejemplo utilizaremos una versión de un conjunto de datos sobre diferentes tipos de cáncer, benignos o malignos [cita original]. En esta versión [enlace versión], se ha reducido el número de outliers a 10 y

se han eliminado observaciones duplicadas. El resultado es este conjunto con 223 objetos, 10 de ellos (2,2 %) anomalías:

```
> WBC.norm.v5[,"SOD15v"] <- Func.SOD(WBC.norm.v5[1:9], k.nn = 15, k.sel =
    10)
> WBC.norm.v5[,"SOD30v2"] <- Func.SOD(WBC.norm.v5[1:9], k.nn = 30, k.sel =
    15)
> WBC.norm.v5[,"SOD50v"] <- Func.SOD(WBC.norm.v5[1:9], k.nn = 50, k.sel =
    15)
> WBC.norm.v5[,"SOD80"] <- Func.SOD(WBC.norm.v5[1:9], k.nn = 80, k.sel = 10)
```

k.nn	k.sel	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
10	5	0.7	0.6859155	0.6184708	0.6005587
15	5	0.5	0.4765258	0.6036111	0.5850013
15	10	0.7	0.6859155	0.6882826	0.673648
20	5	0.4	0.371831	0.3810677	0.3520098
20	10	0.5	0.4765258	0.5355503	0.5137452
20	15	0.5	0.4765258	0.5783876	0.5585936
30	5	0.7	0.6859155	0.667712	0.6521116
30	10	0.6	0.5812207	0.705841	0.6920307
30	15	0.6	0.5812207	0.7475444	0.735692
50	10	0.6	0.5812207	0.6377132	0.6207045
50	15	0.7	0.6859155	0.809143	0.8001826
50	25	0.6	0.5812207	0.7070887	0.693337
80	10	0.7	0.6859155	0.7861111	0.7760694

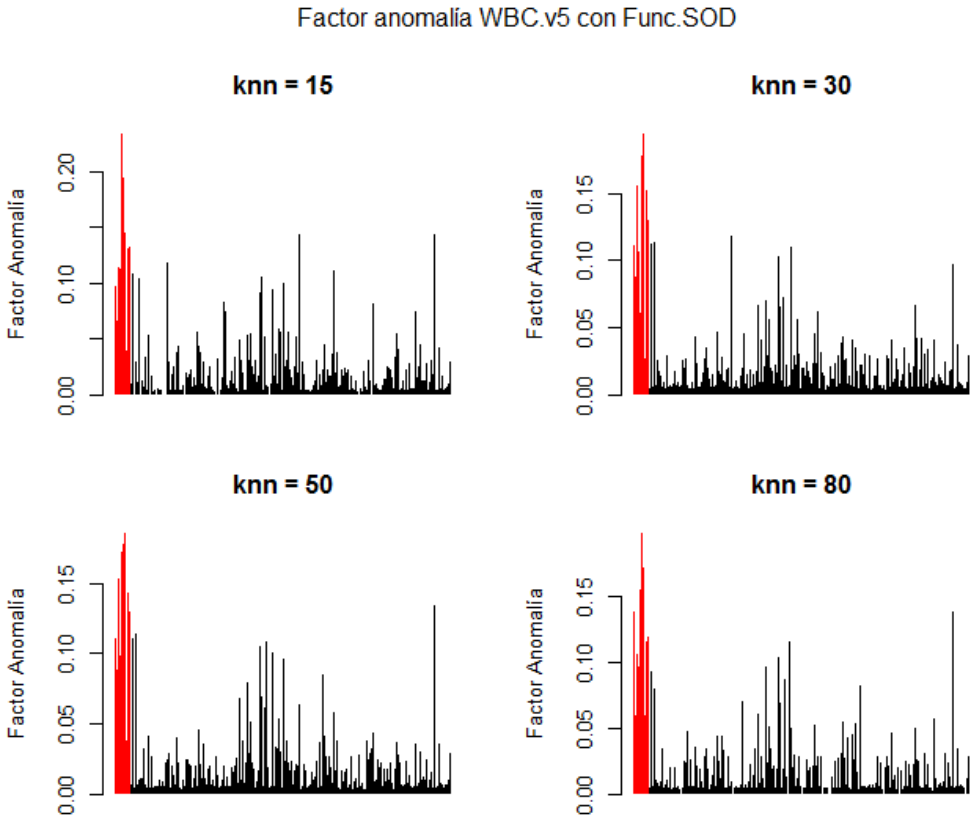


Figura 3.14: Clasificación por *SOD* con diferentes `k.nn` y `k.sel`

3.4. FBOD Feature Bagging for Outlier Detection

Propuesto por A. Lazarevic y V. Kumar en *Feature Bagging for Outlier Detection* [cita], este enfoque aborda el problema de la detección de anomalías mediante la combinación de múltiples algoritmos para detectar anomalías en conjuntos de datos de alta dimensionalidad y con gran cantidad de ruido. A diferencia de los enfoques estándar de captación, donde los modelos de clasificación y regresión que son combinados utilizan distribuciones de datos muestreadas aleatoriamente, en esta propuesta los algoritmos de detección son combinados y su diversidad es mejorada mediante el muestreo aleatorio de subconjuntos de características del conjunto de características completo original. Debido a la sensibilidad de los algoritmos de detección de anomalías a la selección de características utilizadas en el cálculo de distancias, cada detector identifica diferentes anomalías y asigna diferentes puntuaciones de anomalía para cada registro. Las puntuaciones de anomalía son después combinadas con tal de encontrar aquellos outliers de "mejor calidad" que los *outliers* identificados mediante un solo algoritmo de detección.

En esta publicación los autores eligieron, debido a los resultados de estudios anteriores de estos sobre numerosos algoritmos de detección de outliers, algoritmos basados en la computación de distancias de todas las dimensiones entre unos puntos y otros a la vez que en la densidad de las vecindades locales. Por esto, eligieron el *Factor Local de Outlier* (LOF) como algoritmo para demostrar sus avances.

El procedimiento para combinar técnicas de detección de anomalías es llevado a cabo en una serie de T rondas, aunque estas rondas puedan ser ejecutadas en paralelo para un procesado de los datos más rápido. En cada ronda t, el algoritmo de detección es invocado con un conjunto de diferentes características F_t que es utilizado en el cálculo de la distancia. El conjunto de características F_t es seleccionado aleatoriamente del conjunto de datos original, de manera que el número de características en F_t es también seleccionado aleatoriamente entre $\lfloor d/2 \rfloor$ y $(d-1)$, donde d es el número de dimensiones del conjunto de datos original. Cuando el número de características N_t en F_t es elegido, N_t características son seleccionadas aleatoriamente sin sustitución del conjunto de datos original.

Como resultado, cada algoritmo de detección devuelve un *vector de factores de outlier* AS_t que refleja la probabilidad de que cada objeto en el conjunto de datos original $\mathcal{S}$ sea un outlier. Por ejemplo, si $AS_t(i) > AS_t(j)$, el dato x_i tiene mayor probabilidad de ser un outlier que el dato x_j. Al final del procedimiento, tras T rondas, hay T *vectores de factores de outlier* que

corresponden con cada técnica de detección de anomalía llevada a cabo.

- Dado un conjunto S $\{(x_1, y_1), ..., (x_m, y_m)\}$ $x_i \in X^d$ donde d corresponde al número de dimensiones (número de características) del vector
- Normalizar el conjunto de datos S
- Para $t = 1, 2, 3, 4, \ ... \ T$
  1. Elegir aleatoriamente el tamaño del subconjunto de características N_t de una distribución uniforme entre $\lfloor d/2 \rfloor$ y $d(-1)$
  2. Elegir aleatoriamente, sin sustitución, N_t características para crear un subconjunto de características F_t
  3. Ejecutar el algoritmo de detección de anomalías O_t empleando el subconjunto de características F_t
  4. El resultado del algoritmo de detección de anomalías O_t es un vector de factores de anomalía AS_t
- Combinar los vectores de factores de anomalía y obtener un vector final de factores de anomalía AS_{FINAL} como:

$AS_{FINAL} = COMBINE(AS_t), \ t = 1, ..., T$

La función COMBINE es entonces utilizada para unir los T vectores de factores de outlier $AS_t, \ t = \overline{1, T}$ en un único vector de factores de outlier AS_{final}, el cual es utilizado para asignar a cada objeto del conjunto la probabilidad correspondiente de ser una anomalía.

En el momento de utilizar la función para combinar los vectores de factores de outlier surge un problema complejo, similar al problema de combinar resultados de distintos motores de búsqueda para obtener mejores resultados para los *metabuscadores*, ya que no hay etiquetas que ayuden a entender cuán importante son los resultados de la detección de cada algoritmo y los factores de los resultados de cada algoritmo son importantes en el momento de la combinación, ya que proveen la noción de la relevancia del resultado. Debido a esto, los autores proponen dos variantes de la función COMBINE para integrar los resultados de los múltiples algoritmos de detección, basadas en los enfoques utilizados en los *metabuscadores*.

La primera versión realiza la integración por anchura. Este método primero ordena todos los vectores AS_t en los vectores SAS_t, y devuelve índices Ind_t que relacionan los elementos ordenados de los vectores de factores y los elementos originales de los vectores ordenados. Por ejemplo, $Ind_t(1) = k$

significa que en el t vector de factores AS_t, el dato x_k tiene el factor de anomalía más alto $AS_t(k)$.

Dados AS_t, $t = 1, ..., T$, y sea m el tamaño del conjunto de datos S y el de cada vector AS_t

Ordenar todos los vectores de factores de anomalía AS_t en los vectores SAS_t y obtener los índices Ind_t de los vectores ordenados, tal que $SAS_t(l)$ tenga el mayor factor de anomalía y $Ind_t(l)$ es el indice del objeto en S con el mayor factor en $SAS_t(l)$

Teniendo AS_{FINAL} y Ind_{FINAL} vectores vacíos.

Para $i = 1$ to m

Para $t = 1$ to T

Si el indice $Ind_t(i)$ del objeto en el situado en el lugar i por el algoritmo de detección t y que tiene como factor de outlier $AS_t(i)$ no existe en el vector Ind_{FINAL}

Insertar $Ind_t(i)$ al final del vector Ind_{FINAL}

Insertar $AS_t(i)$ al final del vector AS_{FINAL}

Retornar Ind_{FINAL} y AS_{FINAL}

Después de ordenar todos los vectores de factores AS_t, el método por anchura simplemente toma los objetos con mayor puntuación de anomalía de todos los algoritmos de detección e inserta sus índices en el vector Ind_{FINAL}, para después tomar los objetos con el segundo mayor factor de anomalía y los agrega al vector Ind_{FINAL}, y así sucesivamente. Si el índice del objeto actual ya se encuentra en el vector Ind_{FINAL}, no se hace nada. Al final del método por anchura, el vector Ind_{FINAL} contiene índices de cada objeto que están ordenados de acuerdo con la probabilidad que tienen de ser un outlier, y un vector AS_{FINAL} con esa probabilidad.

Los resultados finales del método por anchura son, en general, sensibles al orden de los algoritmos de detección de outliers. De cualquier manera, las diferencias son menores ya que las variaciones podrán ocurrir como mucho en T factores (T es generalmente mucho más pequeño que el número total de objetos del conjunto total), ya que a cada pasada i el método recorre T índices para objetos clasificados en el lugar i del vector de factores de outliers AS_t.

La segunda variación de la función COMBINE, utiliza el método de la Suma Acumulada como muestra el pseudocódigo:

Dado $AS_t,\ t = 1, ..., T$, y m el tamaño de cada vector AS_t
Sumar todos los factores de anomalía AS_t de todas las T iteraciones según:
Para todo $i = 1$ a m

$$AS_{FINAL}(i) = \sum_{t=1}^{T} AS_t(i)$$

Se obtiene el vector AS_{FINAL}

Este método de combinación crea, en primer lugar, un vector AS_{FINAL} con las puntuaciones de outlier de cada objeto sumando todas las puntuaciones asignadas por cada algoritmo de detección en los vectores AS_t de todas las T iteraciones, y luego ordena el vector AS_{FINAL} para, finalmente, identificar como outliers a aquellos objetos con la puntuación más alta.

Es importante darse cuenta que este método ofrece algo más de flexibilidad que en los métodos utilizados en los *metabuscadores*, donde un resultado importante en un motor de búsqueda puede ser ocultado por resultados mediocres en otros buscadores. En este caso al hacer la suma acumulada, un objeto que sea asignado una puntuación considerablemente alta por un algoritmo en concreto podrá tener una puntuación suficientemente alta tras todas las sumas realizadas que todavía le haga ser detectado como una anomalía. Este caso es de vital importancia en los escenarios en que los outliers sean sólo visibles en ciertas dimensiones, ya que en este caso es suficiente con elegir las características importantes sólo en un pequeño número de las iteraciones, calcular factores de outliers altos en estos subconjuntos de dimensiones y con esto hacer que el objeto sea tenido en cuenta como anomalía en las puntuaciones finales.

3.4.1. Func.FBOD {HighDimOut}

Presente en el paquete `HighDimOut`, este función lleva a cabo el algoritmo de detección de outliers mediante selección de atributos.

Listing 3.3: llamada por defecto

```
Func.FBOD(data, iter, k.nn)
```

Para cada `iter` iteración, un subconjunto aleatorio de características, entre $d/2$ y d, es seleccionado para aplicar el método LOF. La función devuelve un vector que contiene las puntuaciones de anomalía finales $FBOD$ de cada observación, que son la suma acumulativa de cada iteración.

Argumentos

`data`	Objeto `dataframe` que contiene las observaciones
`iter`	Número de iteraciones llevadas a cabo
`k.nn`	Número de vecinos más cercanos utilizados para calcular la puntuación *LOF*

Ejemplos

Ejemplo 1. Este conjunto de datos representa pacientes divididos en cuatro clases de acuerdo con el resultado de un examen radiológico. El conjunto original tiene 18 atributos y contiene 148 observaciones de las cuales 6 (4,05 %) son consideradas anómalas. En este caso se ha analizado un conjunto de datos en el que los atributos categóricos han sido codificados *1-a-n*:

```
> lymphography.1ofn.results[,"FBOD10"] <- Func.FBOD(lymphography.1ofn.
    results[, 1:47], iter = 10, k.nn = 10)
> lymphography.1ofn.results[,"FBOD15"] <- Func.FBOD(lymphography.1ofn.
    results[, 1:47], iter = 10, k.nn = 15)
> lymphography.1ofn.results[,"FBOD20"] <- Func.FBOD(lymphography.1ofn.
    results[, 1:47], iter = 10, k.nn = 20)
> lymphography.1ofn.results[,"FBOD30"] <- Func.FBOD(lymphography.1ofn.
    results[, 1:47], iter = 10, k.nn = 30)
>  par(mfrow=c(2,2), oma=c(0,0,2,0))
> barplot(lymphography.1ofn.results$FBOD, ylab = "Puntuación Anomalia", main
    = "k.nn = 10", col = ifelse(lymphography.1ofn.results$Outlier == "yes",
    "red", "grey"), space = 0, border = NA)
> barplot(lymphography.1ofn.results$FBOD15, ylab = "Puntuación Anomalia",
    main = "k.nn = 15", col = ifelse(lymphography.1ofn.results$Outlier == "
    yes", "red", "grey"), space = 0, border=NA)
> barplot(lymphography.1ofn.results$FBOD20, ylab = "Puntuación Anomalia",
    main = "k.nn = 20", col = ifelse(lymphography.1ofn.results$Outlier == "
    yes", "red", "grey"), space=0, border=NA)
> barplot(lymphography.1ofn.results$FBOD30, ylab = "Puntuación Anomalia",
    main = "k.nn = 30", col = ifelse(lymphography.1ofn.results$Outlier == "
    yes", "red", "grey"), space=0, border=NA)
> mtext("Anomalias de Lymphografy.1ofn con FBOD", outer=TRUE)
```

`k.nn`	`iter`	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
10	10	0.8333333	0.8262911	0.9333333	0.9305164
10	5	0.8333333	0.8262911	0.9055556	0.9015649
10	15	0.8333333	0.8262911	0.9150794	0.9114912
15	10	0.6666667	0.6525822	0.8819444	0.8769562
20	10	0.6666667	0.6525822	0.786014	0.7769723
30	10	0.6666667	0.6525822	0.7986111	0.7901017

Ejemplo 2. En este ejemplo se analizara un conjunto de datos de lecturas de radar sobre señales retornadas de la ionosfera [cita]. Este conjunto de datos

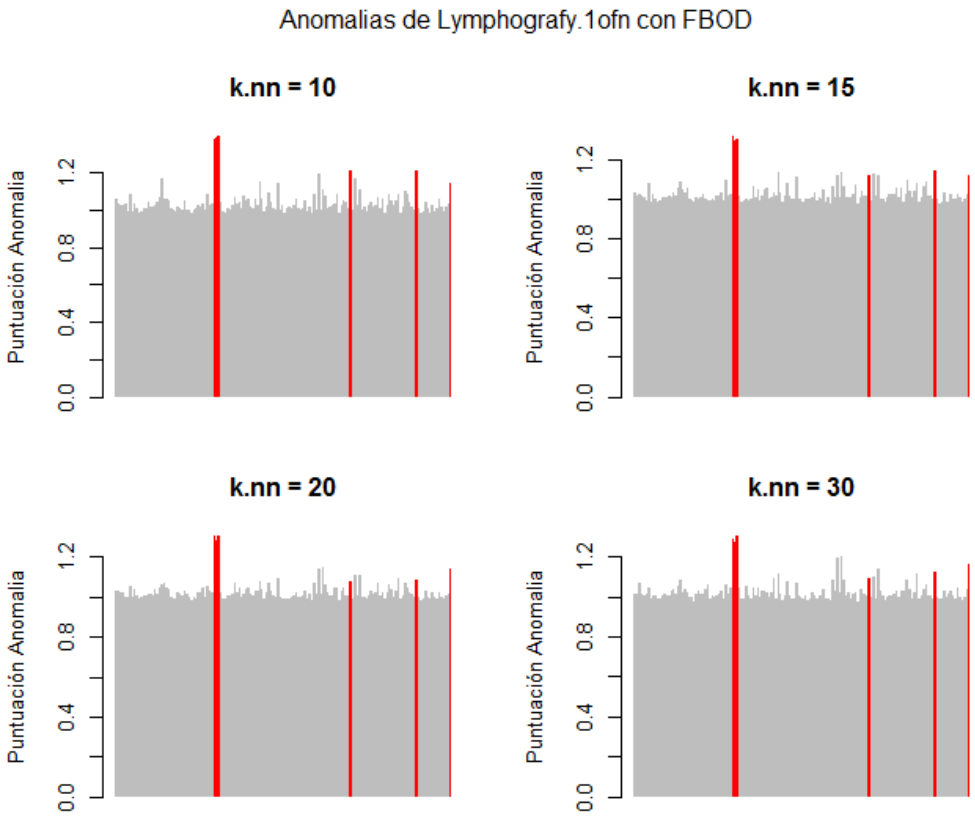


Figura 3.15

diferencia entre "buenas" lecturas, las cuales muestran evidencia de algún tipo de estructura en la ionosfera; y "malas" lecturas, las cuales atraviesan la ionosfera. Contiene 351 observaciones, de las cuales 126 son malas (35,9 %) y 225 son buenas (64,1 %):

```
> ionosphere.FBOD[,"FBOD10"] <- Func.FBOD(ionosphere.FBOD[,3:34],iter = 20,
    k.nn=10)
> ionosphere.FBOD[,"FBOD15"] <- Func.FBOD(ionosphere.FBOD[,3:34],iter = 20,
    k.nn=15)
> ionosphere.FBOD[,"FBOD20"] <- Func.FBOD(ionosphere.FBOD[,3:34],iter = 20,
    k.nn=20)
> ionosphere.FBOD[, "FBOD30"] <- Func.FBOD(ionosphere.FBOD[,3:34], iter =
    20, k.nn=30)
>  par(mfrow=c(2,2), oma=c(0,0,2,0))
> barplot(ionosphere.FBOD$FBOD10, ylab = "Puntuación Anomalia", main= "k.nn
    = 10", col = ifelse(ionosphere.FBOD$X35 == "b", "red", "black"), space
    =0, border=NA)
> barplot(ionosphere.FBOD$FBOD15, ylab= "Puntuación Anomalia", main= "k.nn =
    15", col = ifelse(ionosphere.FBOD$X35== "b", "red", "black"), space=0,
    border=NA)
> barplot(ionosphere.FBOD$FBOD20, ylab = "Puntuación Anomalia", main= "k.nn
    = 20", col = ifelse(ionosphere.FBOD$X35 == "b", "red", "black"), space
    =0, border=NA)
> barplot(ionosphere.FBOD$FBOD30, ylab="Puntuación Anomalia", main = "k.nn =
    30", col = ifelse(ionosphere.FBOD$X35 == "b", "red", "black"), space=0,
     border=NA)
> mtext("Anomalias de Ionosphere con FBOD", outer=TRUE)
```

k.nn	iter	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
10	10	0.8333333	0.74	0.871008	0.7987725
15	10	0.8015873	0.6904762	0.839022	0.7488744
20	10	0.7698413	0.6409524	0.8315605	0.7372343
30	10	0.7460317	0.6038095	0.8368785	0.7455305

Ejemplo 3. En este ejemplo utilizaremos una versión de un conjunto de datos sobre diferentes tipos de cáncer, benignos o malignos [cita original]. En esta versión [enlace versión], se ha reducido el número de outliers a 10 y se han eliminado observaciones duplicadas. El resultado es este conjunto con 223 objetos, 10 de ellos (2,2 %) anomalías:

```
> WBC.norm.v4[,"FBOD20"] <- Func.FBOD(WBC.norm.v4[,1:9],iter=5,k.nn=20)
> WBC.norm.v4[,"FBOD40"] <- Func.FBOD(WBC.norm.v4[,1:9],iter=5,k.nn=40)
> WBC.norm.v4[,"FBOD60"] <- Func.FBOD(WBC.norm.v4[,1:9],iter=5,k.nn=60)
> WBC.norm.v4[,"FBOD80"] <- Func.FBOD(WBC.norm.v4[,1:9],iter=5,k.nn=80)
> par(mfrow=c(2,2), oma=c(0,0,2,0))
> barplot(WBC.norm.v4$FBOD20, main ="k=20", ylab = "Puntuación Anomalia",
    col = ifelse(WBC.norm.v4$outlier=="yes","red","black"), border = NA,
    space=0)
> barplot(WBC.norm.v4$FBOD40, main ="k=40", ylab = "Puntuación Anomalia",
    col = ifelse(WBC.norm.v4$outlier=="yes","red","black"), border = NA,
    space=0)
> barplot(WBC.norm.v4$FBOD60, main ="k=60", ylab = "Puntuación Anomalia",
    col = ifelse(WBC.norm.v4$outlier=="yes","red","black"), border = NA,
    space=0)
> barplot(WBC.norm.v4$FBOD80, main ="k=80", ylab = "Puntuación Anomalia",
    col = ifelse(WBC.norm.v4$outlier=="yes","red","black"), border = NA,
    space=0)
> mtext("Anomalias de WBC.v4 con FBOD", outer=TRUE)
```

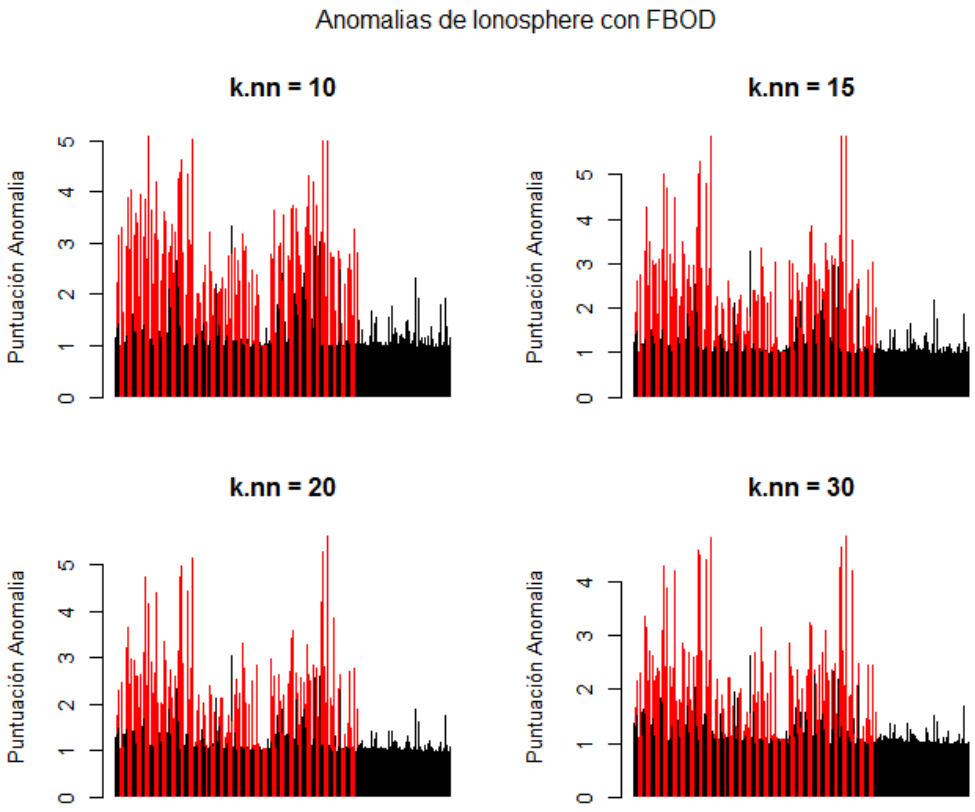


Figura 3.16

k.nn	iter	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
20	5	0.1	0.05774648	0.1756917	0.1369918
20	10	0.2	0.1624413	0.3341417	0.3028808
40	10	0.5	0.4765258	0.5898892	0.5706352
60	10	0.6	0.5812207	0.6357701	0.6186701
80	5	0.9	0.8953052	0.94	0.9371831
80	10	0.9	0.8953052	0.94	0.9371831

Ejemplo 2. En este ejemplo se probara el método con un conjunto de datos de flores Iris [enlace], que contiene 50 observaciones de la variedad *virginica* y 6 de la variedad *versicolor*:

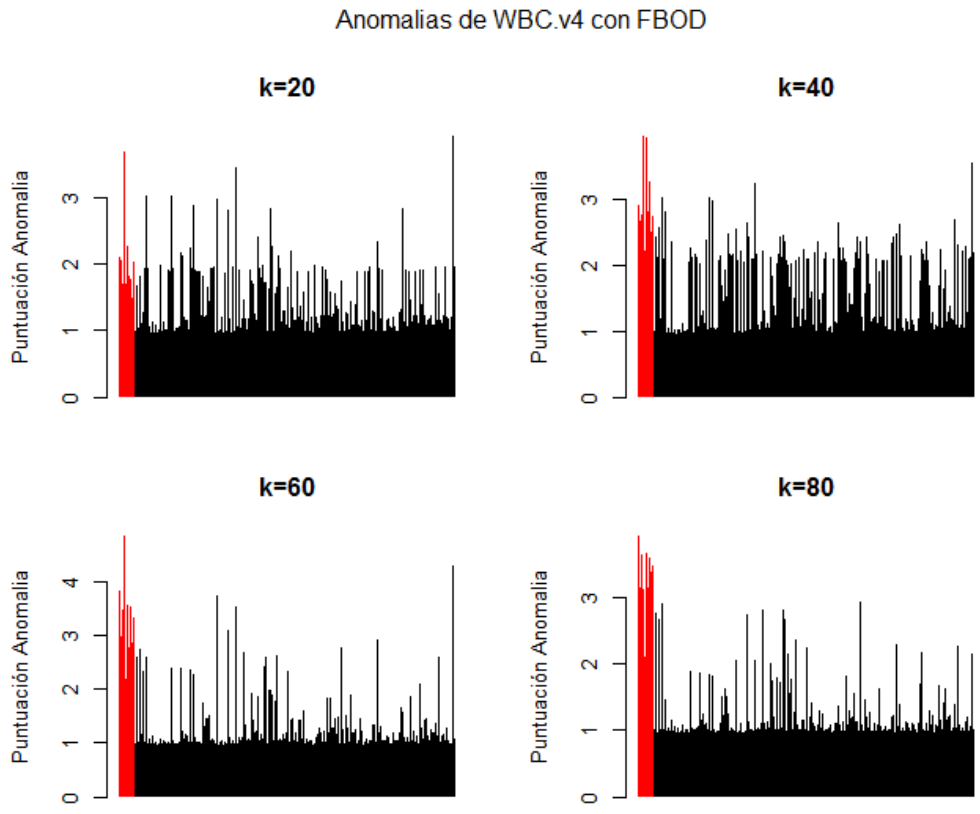


Figura 3.17

```
> iris.test2 <- rbind(virginica, versicolor[sample(1:50,6),])
> iris.test2[,"FBOD5"] <- Func.FBOD(iris.test2[,1:4], iter = 10, k.nn=5)
> iris.test2[,"FBOD10"] <- Func.FBOD(iris.test2[,1:4], iter = 10, k.nn=10)
> iris.test2[,"FBOD15"] <- Func.FBOD(iris.test2[,1:4], iter = 10, k.nn=15)
> iris.test2[,"FBOD20"] <- Func.FBOD(iris.test2[,1:4], iter = 10, k.nn=20)
> par(mfrow=c(2,2), oma=c(0,0,2,0))
> barplot(iris.test2$FBOD5, main = "knn=5", ylab="Puntuación anomalía", col
    = ifelse(iris.test2$Species == "versicolor", "red", "grey"))
> barplot(iris.test2$FBOD10, main = "knn=10", ylab="Puntuación anomalía",
    col = ifelse(iris.test2$Species == "versicolor", "red", "grey"))
> barplot(iris.test2$FBOD15, main = "knn=15", ylab="Puntuación anomalía",
    col = ifelse(iris.test2$Species == "versicolor", "red", "grey"))
> barplot(iris.test2$FBOD20, main = "knn=20", ylab="Puntuación anomalía",
    col = ifelse(iris.test2$Species == "versicolor", "red", "grey"))
> mtext("Anomalias de iris.test2 con FBOD", outer=TRUE)
```

k.nn	iter	$P@n$	$P@n_{ajustada}$	Precisión Media (AP)	$AP_{ajustada}$
3	10	0.3333333	0.2533333	0.4369164	0.3693464
5	10	0.6666667	0.6266667	0.6412805	0.5982341
10	10	0.3333333	0.2533333	0.4663781	0.4023434
15	10	0.3333333	0.2533333	0.4236111	0.3544444
20	10	0.1666667	0.06666667	0.3741653	0.2990651
30	10	0.1666667	0.06666667	0.370197	0.2946207

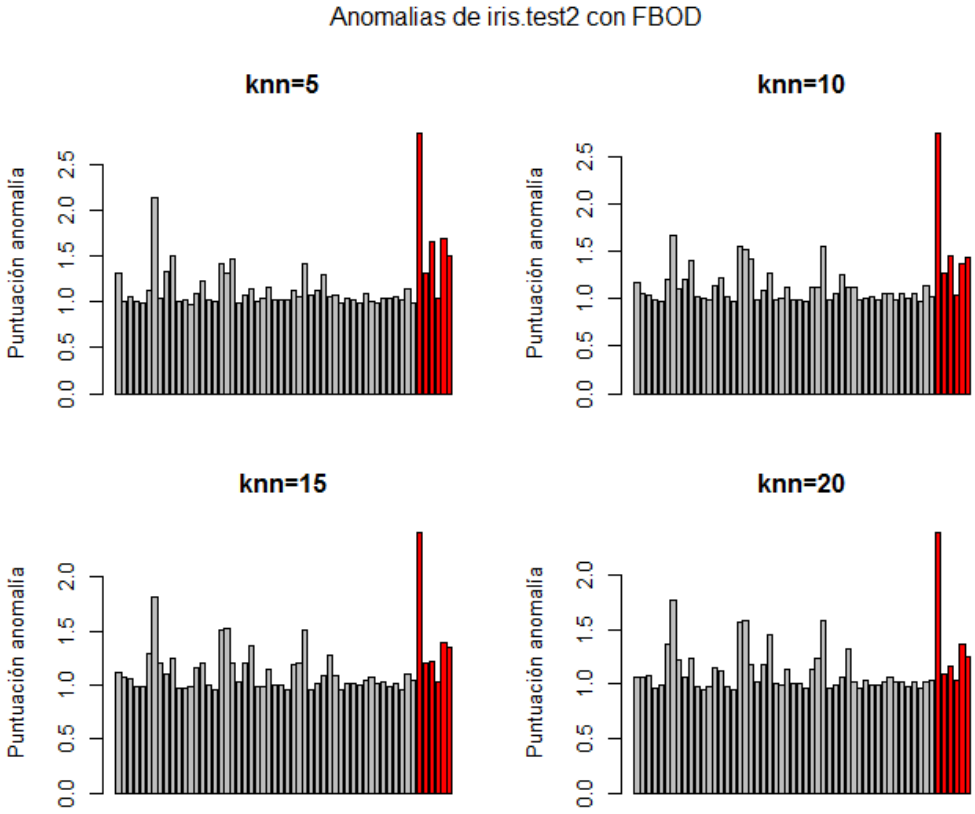


Figura 3.18