

Desarrollo y análisis comparativo de una librería en R para la visualización de datos

Luis Laserna

1 de noviembre de 2019

Índice general

Índice general	1
1. Introduction	3
1.1. Abstrac	3
1.2. Resumen	4
1.3. Planificación del trabajo	5
1.4. Objetivos del trabajo	7
1.5. ¿Por qué R?	7
2. Visualización de datos	10
2.1. Diagramas	10
2.2. Mapas	27
3. Paquetes en R para la visualización de datos	32
3.1. Utilización de paquetes en R	32
3.2. Paquete base de R	32
4. Análisis comparativo de los paquetes	34
4.1. Diagramas	34
4.1.1. Árboles de Tallo y Hoja	34
4.1.2. Gráfico de Barras	35
4.1.3. Histograma	38
4.1.4. Boxplot o Caja de Turkey	40
4.1.5. Diagrama de pareto	42
4.1.6. Gráfico de densidad	43
4.1.7. Pie Charts	45
4.1.8. Gráfico de Puntos	46
4.1.9. Matriz de balón	47
4.1.10. Kernel	49
4.1.11. Sunflowerplot	50
4.1.12. smoothScatter	51
4.1.13. Scatterplot	53
4.1.14. Scatterplot3D	56
4.1.15. Diagrama de violín	58
4.1.16. Series de Tiempo	59
4.1.17. Asociacion Cohen-Friendly	61
4.1.18. Estrellas	62
4.1.19. Gráfico cuádruple/fourfold	64

4.1.20. Diagrama de mosaico	65
4.1.21. Diagrama de secuencia en cascada	67
4.1.22. Gráfica de Venn	69
4.1.23. Gráfica de nivel levelplot	72
4.1.24. Pairs	75
4.1.25. Series de tiempo múltiples	77
4.1.26. Coordenadas paralelas	77
4.2. Mapas	78
4.2.1. country_choropleth	78
4.2.2. GoogleMaps	79
4.2.3. Mapa de contornos	81
4.2.4. Mapa topográfico	82
4.2.5. Mapa 3d rejilla	84
4.2.6. Correlograma	86
4.2.7. Mapa de calor	87
4.2.8. Gráfico de Coordenadas Polares	89
4.2.9. Mapa geográfico leaflet	90
4.2.10. Proyecciones Cartográficas	90
5. Conclusiones y trabajos futuros	92
6. Glosario	94
Bibliografía	96

Capítulo 1

Introduction

1.1. Abstrac

The aim of this project is to develop a library containing the most relevant data visualization functions, as well as fulfilling a deep analysis and classification of the different available visualization techniques. To accomplish it, we have been working on a GNU environment called 'R project'. This environment provides a wide variety of statistical tools that can be integrated with different databases and moreover, allow users to create or complete functions. It is supported by a large community and it has an enormous potential both present and future.

We are now in the Big Data era, we have an overwhelming amount of information, turning the processing phase into a challenging task. At that point, data visualization turns out to be something imperative, essential to find qualifying information. Such way, making decisions and presenting information to other people turn to be an easy task.

Key Words: Visualization, Big Data, Data Science, Data Mining, Statistics.

1.2. Resumen

La razón de este proyecto, es la de desarrollar una librería que contenga las funciones más relevantes sobre la visualización de datos, así como realizar un análisis y clasificación de las diferentes técnicas de visualización, para ello se ha trabajado sobre un entorno GNU llamado R Project. Este entorno proporciona un amplio abanico de herramientas estadísticas que puede integrarse con diferentes bases de datos y además permite a los usuarios la creación y/o extensión de funciones, por ello cuenta con una numerosa comunidad y tiene un enorme potencial presente y futuro.

Actualmente nos encontramos en la era del Big Data, en la cual obtenemos más datos que nunca, siendo el tratamiento de los mismos complejo, por ello, deberemos poner límite a nuestros datos, realizando una correcta comprensión de los mismos. Ahí es donde la visualización de datos se hace imprescindible, utilizar la visualización de datos para descubrir información clarificatoria. De tal manera que podremos tomar decisiones de manera más sencilla y poder facilitar esa información al resto de personas integrantes en una organización. En este documento nos adentraremos en la visualización de datos y podremos aplicar directamente las funciones sobre los conjuntos de datos a estudiar.

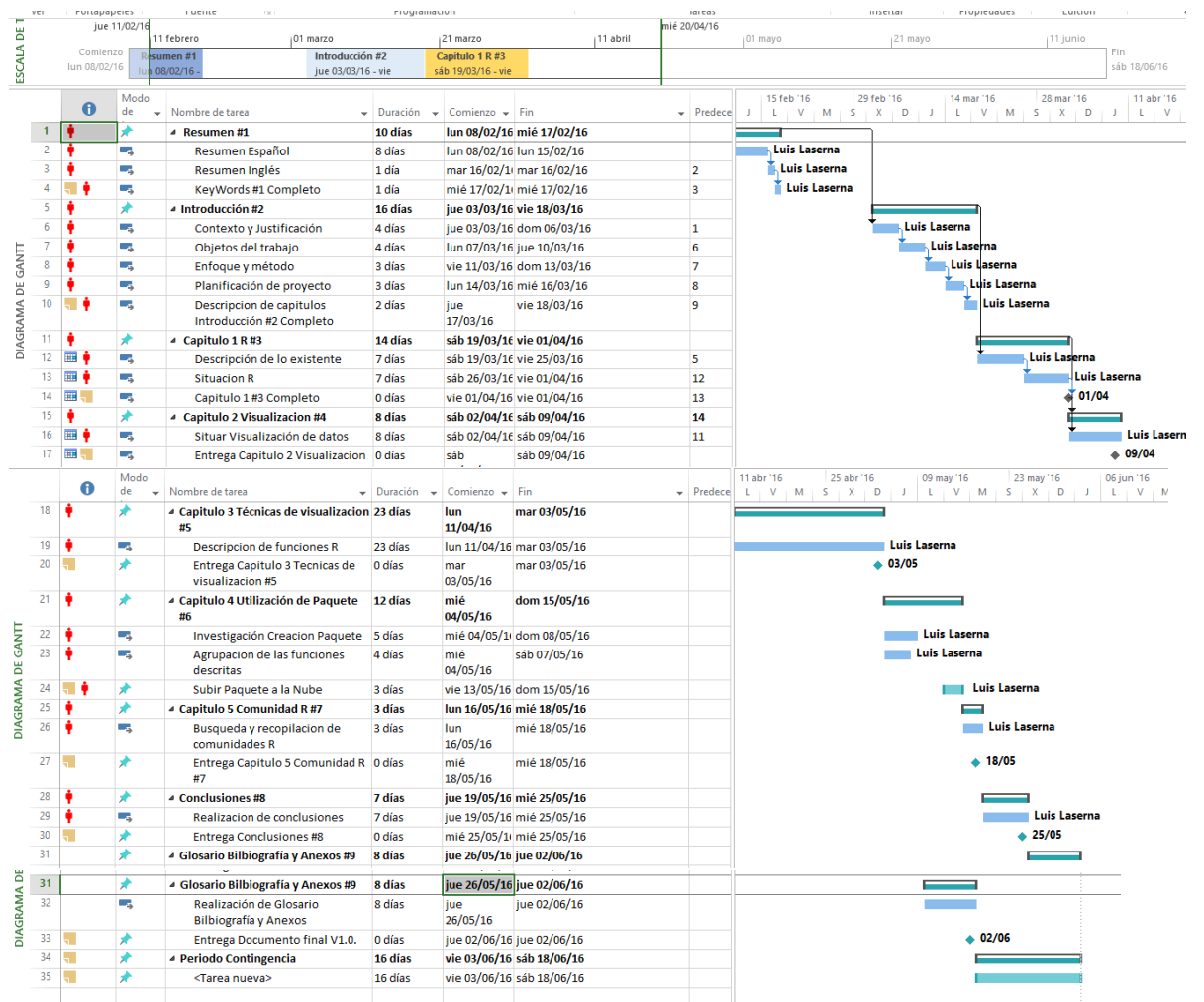
Palabras Clave: Visualización, Big Data, Data Science, Minería de datos, Estadística.

1.3. Planificación del trabajo

En este apartado realizaremos una descripción de los recursos necesarios para realizar el proyecto, las tareas a realizar y una planificación temporal de cada tarea utilizando un diagrama de Gantt. Esta planificación marca cuáles son los hitos parciales de cada una de las entregas.

Para ello, en primer lugar se ha dividido el proyecto en diferentes partes que se muestran a continuación representadas en un diagrama de Gantt. Siendo la lista de tareas principales las siguientes:

1. Desarrollo de Resumen. Que constará de un pequeño resumen de una hoja en español e inglés. Describirá la principal finalidad de este proyecto y ayudará a entender el objetivo.
2. Introducción. Estará dividido en la planificación y los objetivos.
3. Por qué la utilización de la herramienta R project. Se realizará una comparación entre los principales lenguajes y herramientas estadísticas utilizadas para la visualización de datos y mostraremos por qué lo hemos elegido.
4. Visualización de datos. En esta parte se realizará una descripción sobre la visualización de los datos y su utilización. Así como los diferentes tipos de diagramas y mapas que se utilizan y una breve descripción de lo mismo.
5. Paquetes R. Se realizará una breve descripción sobre la utilización de los paquetes en el entorno de R project.
6. Análisis comparativo de paquetes y funciones. Se hará un análisis de los paquetes y funciones que se utilizan para realizar las visualizaciones explicadas.
7. Conclusiones y trabajos futuros. Se explican las conclusiones del trabajo realizado y las futuras extensiones a raíz de haber elaborado dicho trabajo.
8. Bibliografía.



1.4. Objetivos del trabajo

El objetivo de este trabajo es la realización de un estudio el cual nos permita diseñar una clasificación de visualización de datos en R. Una vez realizado dicho estudio se deberá realizar un análisis comparativo entre las funciones que nos permite R para la creación y reproducción de estas visualizaciones.

Para realizar esta clasificación será necesario analizar todas y cada una de las técnicas de visualización de datos. Una vez obtenido el listado con todas ellas tendremos que clasificarlas por grupos. Para ello se realizará una primera segmentación en Diagramas, Mapas y Redes.

Una vez obtenido el agrupamiento de las diferentes visualizaciones se extraerá la información de cómo reproducirlas en nuestro entorno objeto 'R'. En este apartado dispondremos de toda la información sobre los paquetes requeridos y las funciones utilizadas para dicha representación.

Este proyecto está orientado a todas las personas que necesiten disponer de una clasificación de visualización de datos y su reproducción en R. De una manera rápida podrán disponer de las funciones y poder reproducirlas ellos mismos.

1.5. ¿Por qué R?

Hoy en día existen numerosos programas de computación que permiten realizar diversos análisis estadísticos. Entre los más conocidos están SPSS, Stata, SAS, Matlab o R.

El área de estudio, el nivel de conocimiento del usuario, el control o la exactitud que busquemos, así como también el presupuesto o tiempo del que dispongamos, son determinantes a la hora de recurrir a un programa u otro.

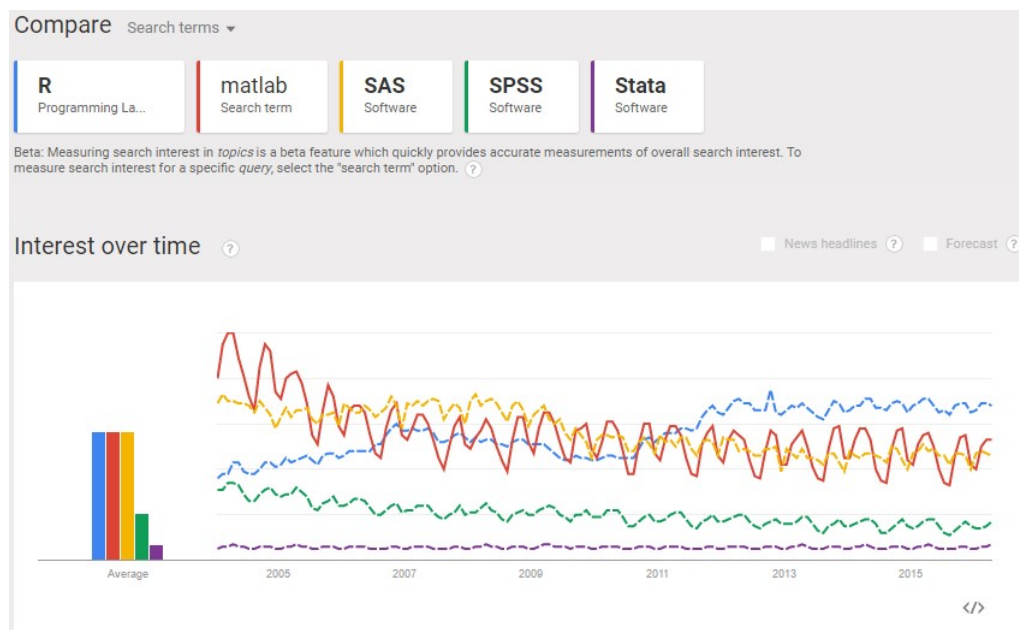
En este capítulo explicaremos las razones que nos han llevado a elegir R como nuestra herramienta de trabajo.

Listado programas estadísticos más relevantes:

Software	Interface	Curva de aprendizaje	Manejo de datos	Análisis estadístico	Gráficos
SPSS	Menús y sitaxis	Gradual	Moderado	Campo de aplicación moderado y baja versatilidad	Bien
Stata	Menús y sitaxis	Moderada	Potente	Campo de aplicación amplio y versatilidad media	Bien
SAS	Sitaxis	Empinada	Muy potente	Campo de aplicación muy amplio y alta versatilidad	Muy bien
R	Sitaxis	Empinada	Muy potente	Campo de aplicación muy amplio y alta versatilidad	Excelente
Matlab	Sitaxis	Empinada	Muy potente	Campo de aplicación muy limitado y alta versatilidad	Excelente

Cuadro 1.1: Cuadro comparativo Universidad de Nueva York (NYU) [1]

Una vez vista una breve descripción sobre las herramientas más utilizadas para realizar análisis estadísticos, vamos a realizar un estudio por búsquedas realizadas en Google sobre las diferentes herramietnas, esto nos dará una visión superficial o general sobre el volumen de comunidad que dispone cada una de ellas, así como las tendencias.



Se puede observar la evolución constante que muestra R respecto al resto de software. Empezó en el mercado en una posición inferior respecto a dos de sus competidores más fuertes (Matlab y SAS), la aparición de R como alternativa libre y la gran capacidad para manejar datos de la que dispone fue un golpe duro para sus competidores que hizo que bajasen, incluso en 2011 consiguió

superarlos. Esto es un indicativo de que R es una gran herramienta la cual cuenta con una numerosa comunidad estadística.

Por ello vamos a centrarnos en R y enumerar las siguientes cualidades más características que dispone:

1. Herramienta gratuita: R es totalmente gratuito y se puede descargar desde su página oficial(www.r-project.org). Si bien es cierto que la mayoría de los software propietario ofrecen grandes periodos de prueba gratuitos o incluso licencias sin costo a los universitarios, pensamos que R sigue siendo la opción más adecuada.
2. Compatibilidad: Todos los programas soportan su instalación en Windows, pero hay una gran cantidad de usuarios que utilizan otros sistemas operativos como Macintosh o S.O. Linux, que suelen tener problemas de compatibilidad con este tipo de software. R es el único que funciona de manera estable e íntegra en los tres sistemas operativos de mayor uso.
3. Alta escalabilidad: Muchos paquetes estadísticos propietario invierten buena parte de sus recursos en investigación y desarrollo. No es el caso de R, sin embargo, una de sus características más destacables es la ampliación de funcionalidades mediante la inclusión paquetes. Estos paquetes pueden ser desarrollados y puestos a disposición por cualquier persona. Por ello, la renovación e implementación de nuevos procedimientos en R es bastante rápida. Muchos de los principales paquetes de R han sido desarrollados por estadísticos que trabajan en diversas instituciones a nivel mundial, por tanto, podemos esperar que cuenten con algoritmos modernos y robustos.
4. Software libre: Esta característica esta muy ligada a la anterior. Gracias a que se trata de un software libre, todo el código, los procedimientos y los algoritmos están a disposición de todos los usuarios. Están expuestos a continuas revisiones, correcciones y refinaciones lo que asegura la calidad de los mismos, este es uno de los factores más clave con las que cuenta este producto, por este motivo cuenta con una gran comunidad de desarrolladores. R puede generar numerosos gráficos muy detallados, en el caso que no existir podremos desarrollarlo sin nada que nos lo impida.

Capítulo 2

Visualización de datos

La visualización de los datos permite a la gente analizar los datos cuando ellos no saben exáctamente qué preguntas necesitan responder por adelantado.[15]

Nuestra era de la información más a menudo se siente como una era de sobrecarga de información. Las excesivas cantidades de información son abrumadoras, los datos en bruto se convierten en útiles sólo cuando aplicamos métodos de conseguir la comprensión de la misma. Afortunadamente, los seres humanos son criaturas intensamente visuales. Pocos de nosotros podemos detectar patrones entre las filas de números, pero incluso los niños pequeños pueden interpretar los gráficos de barras, es decir, a partir de representaciones visuales los números. Por esa razón, la visualización de datos es un ejercicio de gran alcance. [16]

La visualización de datos es la manera más rápida para comunicarla a los demás. Por supuesto, las visualizaciones, como las palabras, se pueden usar para mentir, engañar, o distorsionar la verdad. Pero cuando se practica con honestidad y con cuidado, el proceso de visualización puede ayudarnos a ver el mundo de una manera nueva, revelando patrones y tendencias inesperadas en lo oculto de lo contrario a la información que nos rodea. En su mejor momento, la visualización de datos es la narración de expertos. La visualización es un proceso de mapeo de la información a efectos visuales. elaboramos reglas que interpretan los datos y expresan sus valores como propiedades visuales.[16]

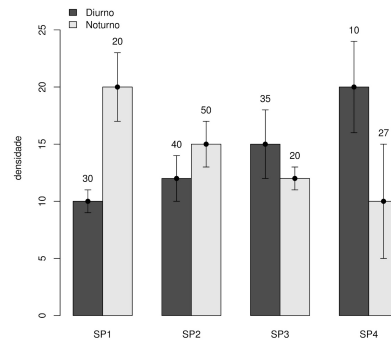
En este apartado mostraremos los diferentes tipos de visualizaciones aplicables a nuestro entorno objetivo, apoyados de una imagen como muestra de ejemplo y una breve descripción sobre la visualización.

2.1. Diagramas

Es un dibujo geométrico, muy utilizado en ciencia, en educación y en comunicación; con el que se obtiene la presentación gráfica de una proposición, de la resolución de un problema, de las relaciones entre las diferentes partes o elementos de un conjunto o sistema, o de la regularidad en la variación de un fenómeno que permite establecer algún tipo de ley.[2]

Gráfico de Barras

Imagen

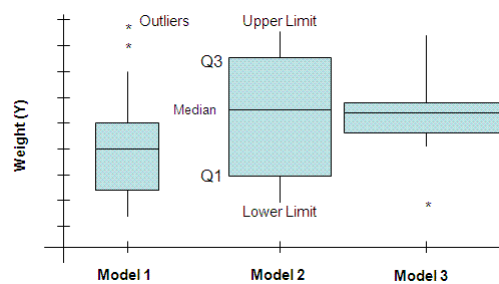


Descripción

También conocido como gráfico de barras o diagrama de columnas, es una forma de representar gráficamente un conjunto de datos o valores, y está conformado por barras rectangulares de longitudes proporcionales a los valores representados. Los gráficos de barras son usados para comparar dos o más valores. Las barras pueden orientarse verticalmente u horizontalmente.

Boxplot o Caja de Tukey

Imagen

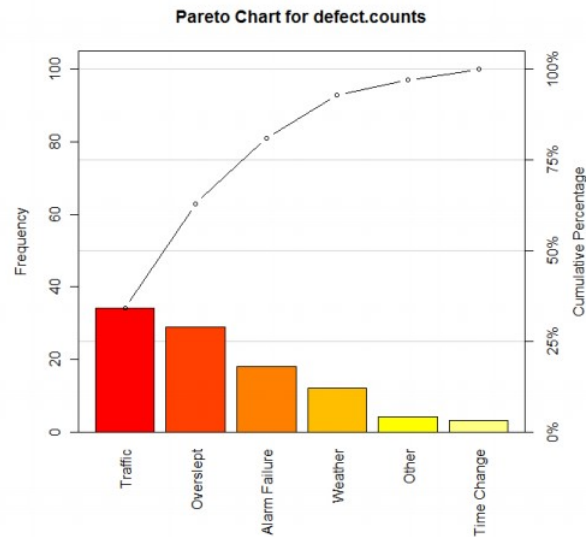


Descripción

Los diagramas de Caja-Bigotes (nombre original 'boxplots o box and whisker's) consiste en un rectángulo o caja del cual se extienden dos líneas o bigotes. Son una presentación visual que describe varias características importantes, al mismo tiempo, tales como la dispersión y simetría.[3]

Diagrama de pareto

Imagen



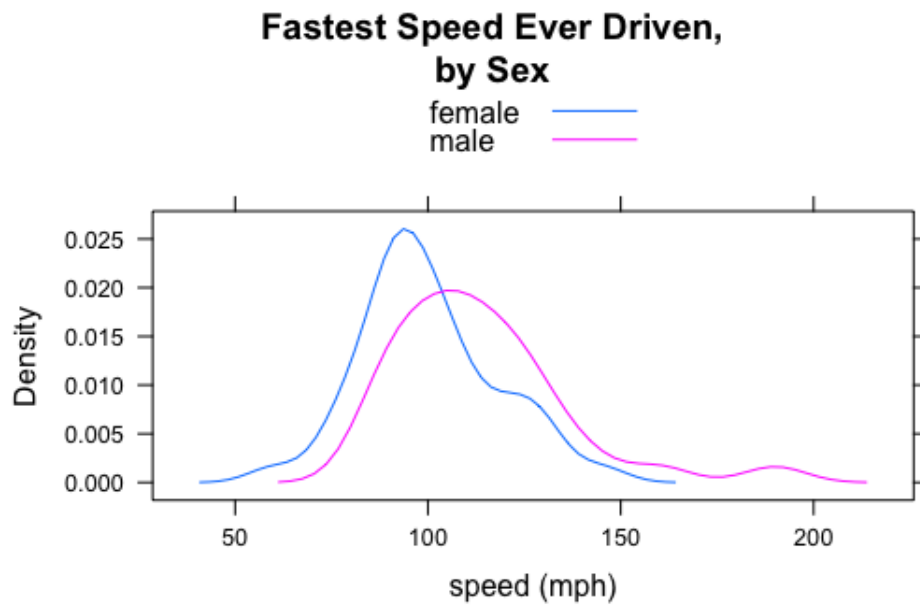
Descripción

El diagrama de Pareto, también llamado curva cerrada o Distribución A-B-C, es una gráfica para organizar datos de forma que estos queden en orden descendente, de izquierda a derecha y separados por barras. Permite asignar un orden de prioridades.[4]

El diagrama permite mostrar gráficamente el principio de Pareto (pocos vitales, muchos triviales), es decir, que hay muchos problemas sin importancia frente a unos pocos muy importantes. Mediante la gráfica colocamos los 'pocos que son vitales' a la izquierda y los 'muchos triviales' a la derecha.[4]

Gráfico de densidad

Imagen

**Descripción**

Se trata de una representación gráfica de la densidad estimada.

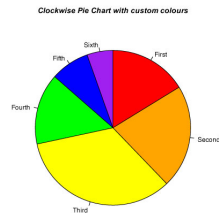
El algoritmo utilizado en `density.default` dispersa la masa de la función de distribución empírica sobre una rejilla regular de al menos 512 puntos y luego utiliza la transformada rápida de Fourier para convolucionar esta aproximación con una versión discretizada del núcleo y luego utiliza aproximación lineal para evaluar la densidad en los puntos especificados. [5]

Las propiedades estadísticas de un núcleo están determinadas por $\int \hat{K}(t) dt = 1$ que es siempre = 1 para nuestros núcleos (y por tanto el ancho de banda BW es la desviación estándar del núcleo) y $R(K) = \int (\hat{K}(t))^2 dt$. MSE-equivalente anchos de banda (para diferentes núcleos) son proporcionales a $\sqrt{R(K)}$ que es invariante escala y para nuestros núcleos igual a $R(K)$. Este valor se devuelve cuando `give.Rkern = true`. Vea los ejemplos para el uso de ancho de banda equivalente exacto. [5]

Valores infinitos en x se supone que corresponden a un punto de masa de $+/- \text{Inf}$ y la estimación de la densidad es de la sub-densidad en $(-\text{Inf}, +\text{Inf})$. [5]

Pie Charts

Imagen

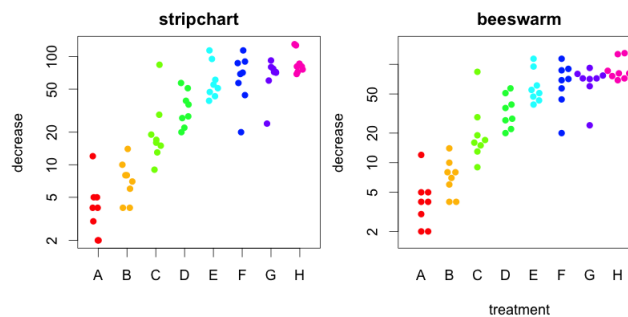


Descripción

Gráfico de sectores, también llamado gráfico de tarta o de pastel, el cual es una representación de datos en forma de segmentos o porciones dentro de una figura circular. Estos datos suelen ser porcentajes y porciones.[6]

Gráficos de Puntos

Imagen

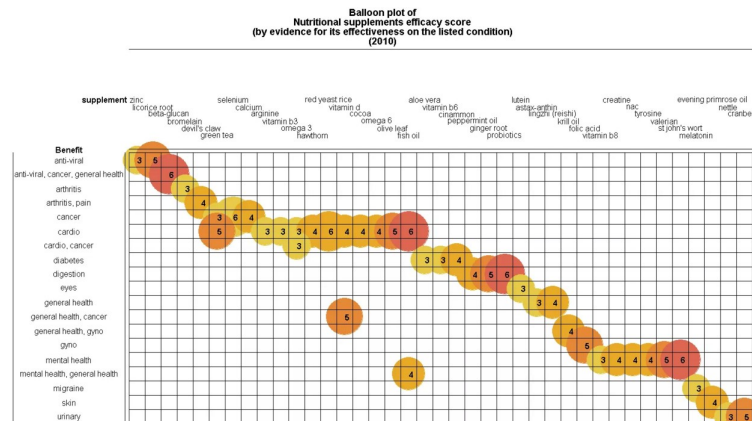


Descripción

Representación de gráficos de puntos unidimensionales. Es una alternativa a boxplot cuando tenemos muestras de datos reducidas y nos sirve para facilitar la localización de los datos así como su dispersión y variabilidad.

Matriz de balón

Imagen

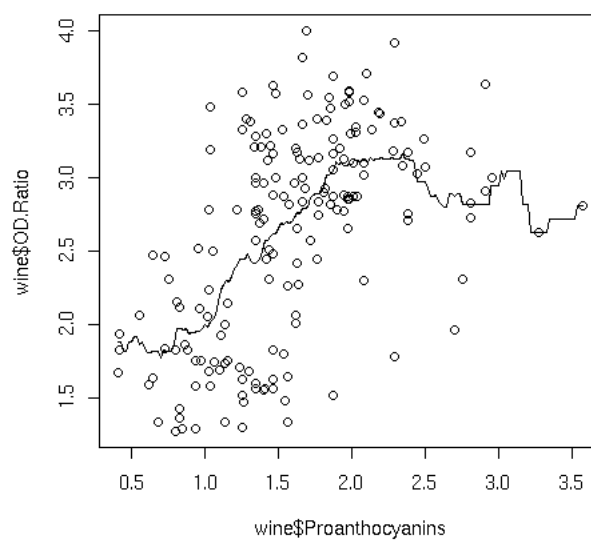


Descripción

Representación de una matriz gráfica donde cada celda contiene un punto cuyo tamaño refleja la magnitud relativa del componente correspondiente. Es una buena alternativa a Boxplot cuando se trata de pocos datos.

Kernel

Imagen

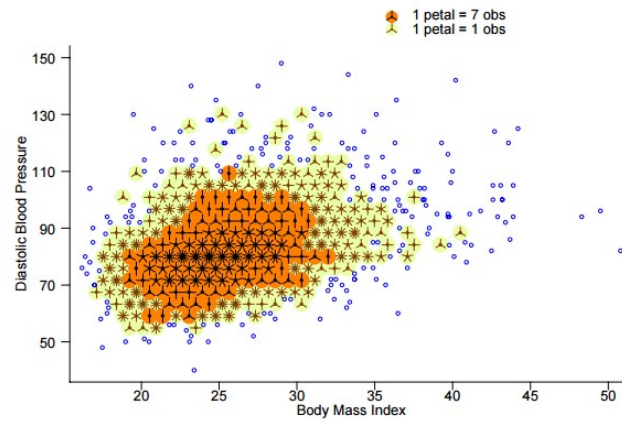


Descripción

Este diagrama representa la regresión Kernel que estima la variable dependiente continua a partir de un conjunto limitado de puntos de datos por convolución de las ubicaciones de los puntos de datos con una función kernel -aproximadamente hablando, la función de kernel especifica la forma de 'desenfoco' de la influencia de los puntos de datos, de modo que sus valores pueden ser utilizados para predecir el valor de localidades cercanas. [7]

Sunflower

Imagen

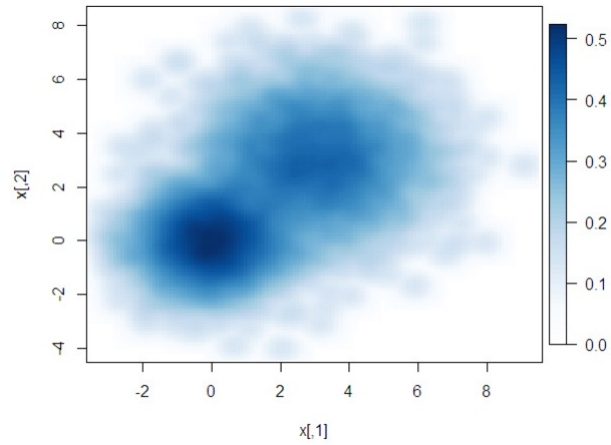


Descripción

Distribución que se utilizan para mostrar datos de dos variables de alta densidad. Son útiles para los datos en un gráfico de dispersión convencionales que son difíciles de leer debido a sobre-saturamiento del símbolo gráfico. [8]

SmoothScatter

Imagen

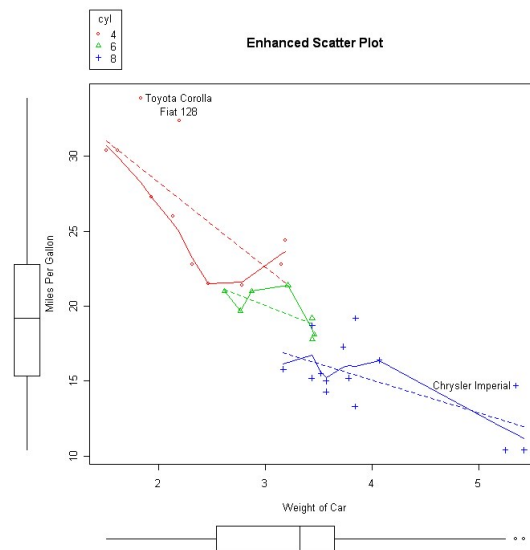


Descripción

Una representación de un gráfico de dispersión con densidad de color suavizado, obtenida a través de una estimación de densidad (2D) del núcleo esto nos facilita la visualización de la distribución.

Scatterplot

Imagen

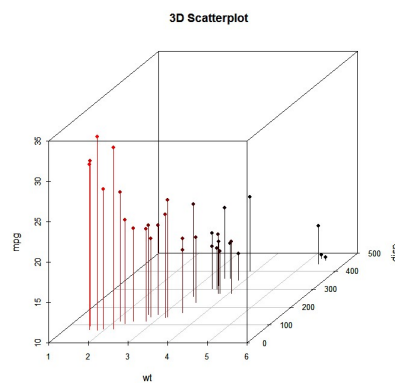


Descripción

Se trata de una mejora de los gráficos de dispersión, con diagramas de caja en los márgenes, una regresión no paramétrica suave, suavizado propagación condicional, la identificación de los outlier, y una línea de regresión; sp es una abreviatura de diagrama de dispersión.

Scatterplot3D

Imagen

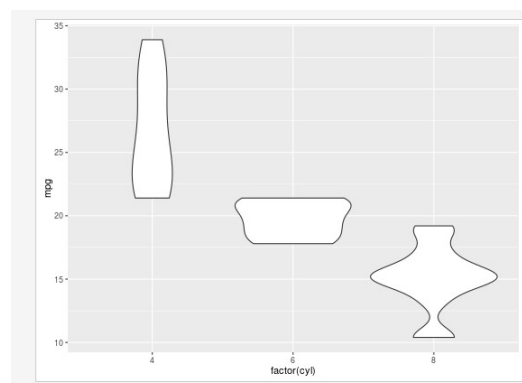


Descripción

Básicamente scatterplot3d genera un gráfico de dispersión en el espacio 3D usando una proyección paralela. dimensiones superiores (cuarta, quinta, etc.) de los datos pueden visualizarse en cierta medida utilizando, por ejemplo, diferentes colores, tipos de símbolos o tamaños de símbolos.

Diagrama de violín

Imagen

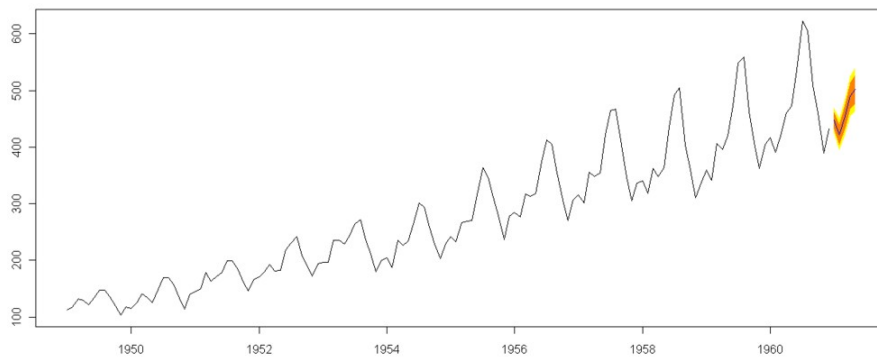


Descripción

Un diagrama de violín es un método de visualización de los datos numéricos. Es un diagrama de caja con una diagrama de densidad de núcleo rotado en cada lado. El diagrama de violín es similar a los diagramas de caja, excepto que también muestran la densidad de probabilidad de los datos a diferentes valores.[9]

Series de Tiempo

Imagen

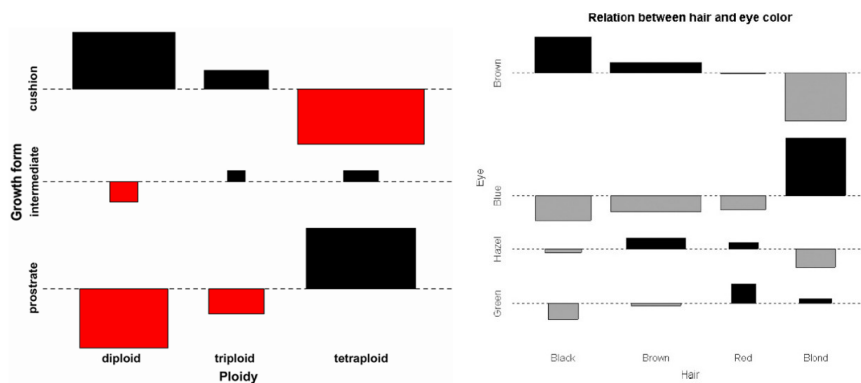


Descripción

Representación de una serie temporal o cronológica es una secuencia de datos, observaciones o valores, medidos en determinados momentos y ordenados cronológicamente.

Asociación Cohen-Friendly

Imagen

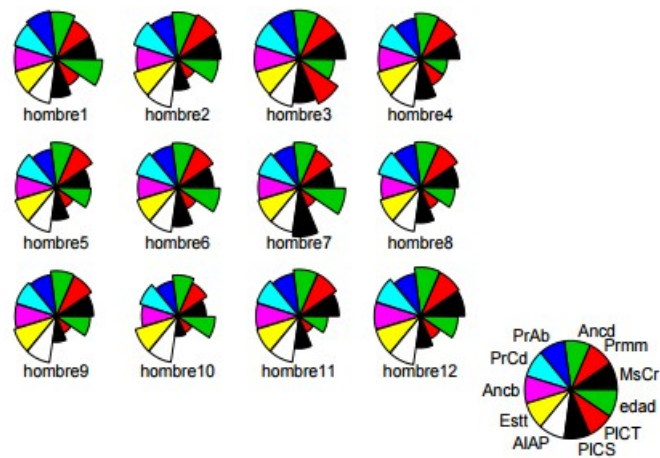


Descripción

Diagrama de asociación surgida por Cohen(1980) y ampliado por Friendly(1992) que indica las desviaciones de la independencia de filas y columnas en una tabla de contingencia de dos dimensiones. La visualización se ve reforzada por la utilización de colores.[10] [11]

Diagrama de Estrellas

Imagen

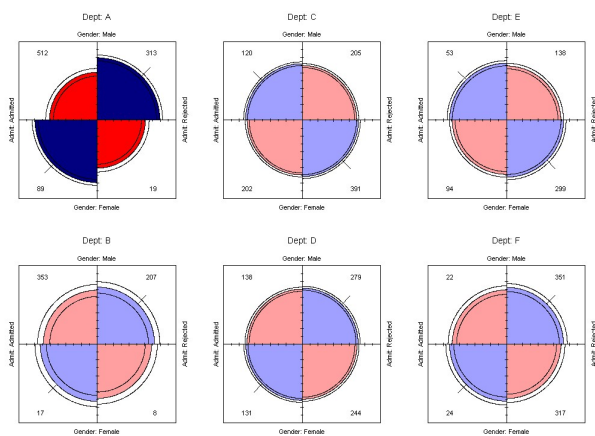


Descripción

Son una forma útil para representar observaciones multivariantes con un número arbitrario de variables. Cada observación se representa como una figura en forma de estrella con un rayo para cada variable. Para una observación dada, la longitud de cada rayo se hace proporcional al tamaño de esa variable.

Gráfico cuádruple/fourfold

Imagen

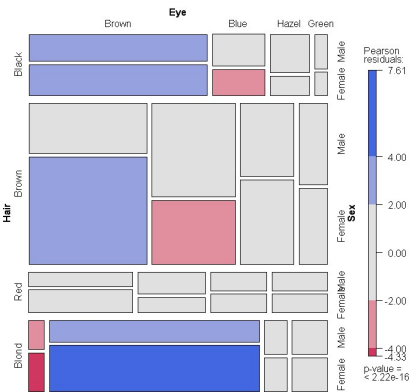


Descripción

Visualización cuádruple de una tabla de contingencia de 2 x 2 x k, lo que permite la inspección visual de la asociación entre dos variables dicotómicas en una o varias poblaciones.[12]

Diagrama de mosaico

Imagen

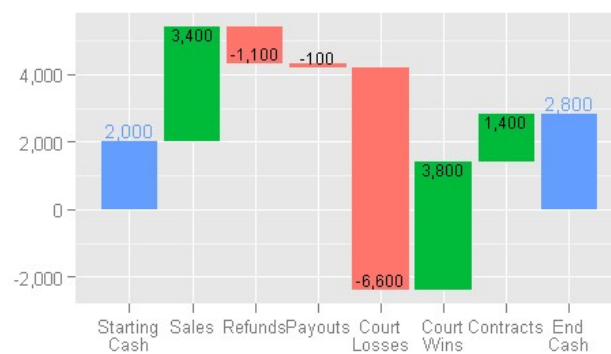


Descripción

Permite mostrar datos multivariantes en gráficos de dos dimensiones formados por tres o más ejes longitudinales, que representan variables cuantitativas, los cuales parten del mismo origen.

Diagrama de secuencia en cascada

Imagen

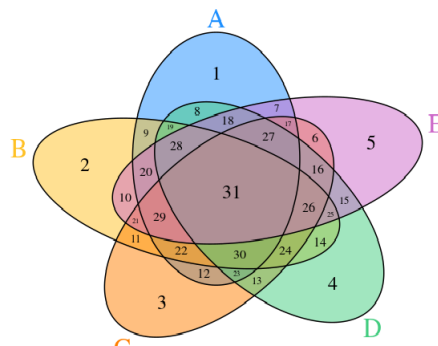


Descripción

Representa datos positivos o negativos, introducidos secuencialmente, y nos permite visualizar su efecto acumulativo a través de columnas partiendo de un eje horizontal a media altura. Son utilizados a menudo para fines de análisis en el entorno de negocios para mostrar el efecto de la secuencia introducido valores negativos y/ o positivos.

Diagrama de Venn

Imagen

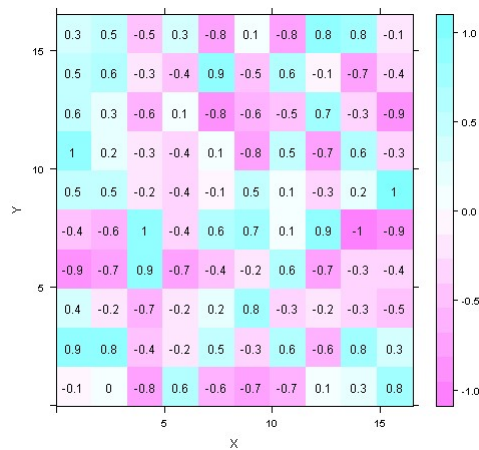


Descripción

Los diagramas de Venn son esquemas que se utilizan en la teoría de conjuntos, tema de interés en matemáticas, lógica de clases y razonamiento diagramático. Estos diagramas muestran colecciones (conjuntos) de cosas (elementos) por medio de líneas cerradas. La línea cerrada exterior abarca a todos los elementos bajo consideración, el conjunto universal U . [13]

Gráfica de nivel `levelplot`

Imagen

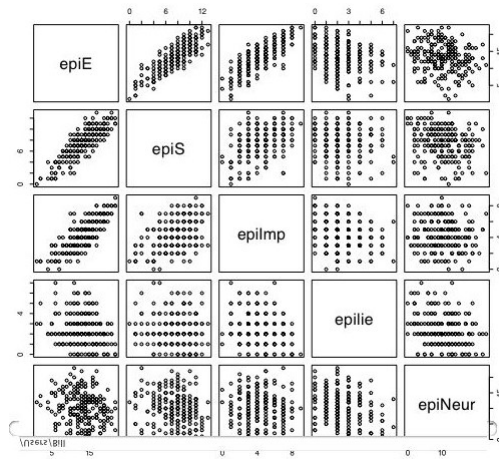


Descripción

Esta función nos permite realizar un gráfico de una función tridimensional a través de los niveles representados el plano por regiones con diferentes colores. Es una alternativa al gráfico de contorno.[14]

Pairs

Imagen



Descripción

Genera una matriz de diagramas de dispersión. Donde podemos observar de manera rápida las dispersiones entre dos variables, la distribución de las variables y las correlaciones entre dos variables.

Series de tiempo múltiples

Imagen

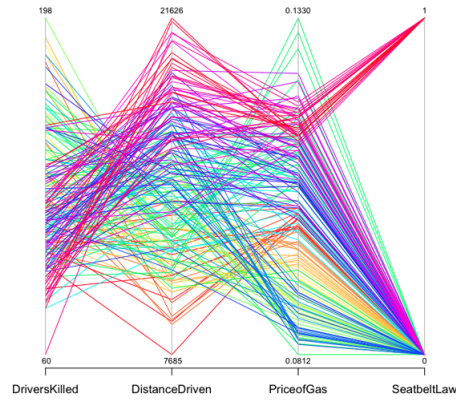


Descripción

Trazar varias series de tiempo en un diagrama común. Pueden tener diferentes bases de tiempo.

Coordenadas paralelas

Imagen



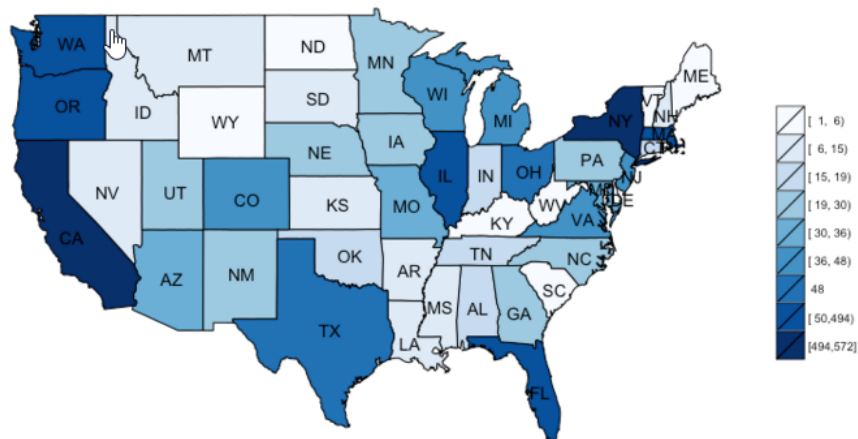
Descripción

Este diagrama nos permite representar n-dimensiones en un sistema bidimensional. Se trata de un trazado de coordenadas paralelas que asigna cada dato de la tabla como una línea, cada atributo de una fila es representado mediante un punto.

2.2. Mapas

Mapas coropléticos

Imagen



Descripción

Nos muestra un mapa temático en el que las regiones se les aplica un degradado de color de acuerdo a las medidas estadísticas que se especifiquen en los argumentos. Como por ejemplo podría ser la densidad de población o el ingreso por habitante. Esto nos permite hacer una fácil comparación utilizando una medida estadística entre las regiones.

GoogleMaps

Imagen

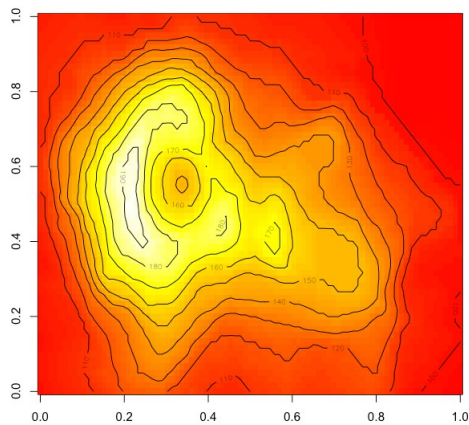


Descripción

Nos muestra el mapa de google mediante la función `gvisMap` lee un hoja de datos y crea una salida de texto que se envía a la API de visualización de Google, puede ser incluido en una página web, o como una página independiente.

Mapas de contorno

Imagen

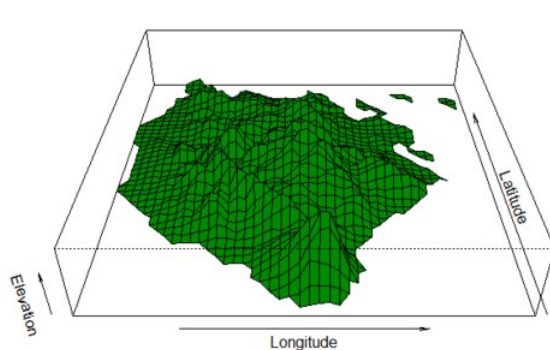


Descripción

Se trata de un mapa geográfico de contornos o también añade la líneas a un gráfico ya existente. Una curva de nivel o de contorno es aquella línea que en un mapa une todos los puntos que tienen igualdad de condiciones y de altitud.

Mapas topográficos

Imagen

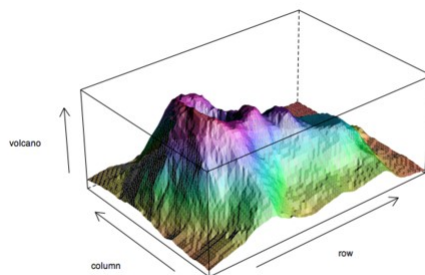


Descripción

Un mapa topográfico es una representación, generalmente parcial, del relieve de la superficie terrestre a una escala definida. A diferencia de los planos topográficos, los mapas topográficos representan amplias áreas del territorio: una zona provincial, una región, un país o el mundo.

Mapas 3d rejilla

Imagen

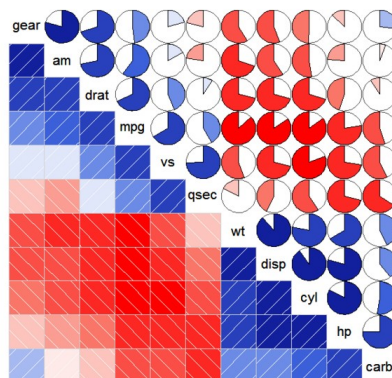


Descripción

Es un tipo de gráfico que se utiliza para mostrar una superficie de datos geográficos, por ejemplo este tipo de gráfico se utilizaría o podría ser utilizado para mostrar un modelo ajustado con más de una variable explicativa. Estas parcelas están relacionados con gráficos de contorno que son las dos dimensiones equivalentes.[14]

Correlograma

Imagen

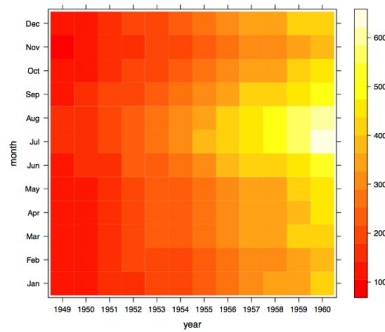


Descripción

Correlograms nos ayudan a visualizar los datos en matrices de correlación. Las celdas de la matriz pueden ser sombreados o de color para mostrar el valor de correlación.

Mapa de calor

Imagen



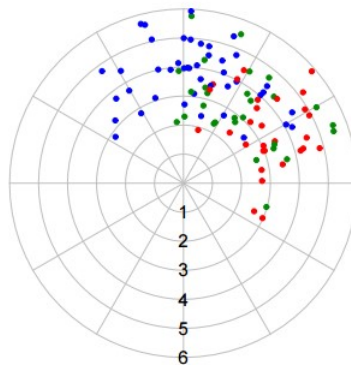
Descripción

Se trata de una representación gráfica de los datos donde los valores individuales que están contenidas en una matriz y son representados como colores. Por lo general, la reordenación de las filas y columnas se realiza acuerdo con un conjunto de valores (fila o columna)

Gráfico de Coordenadas Polares

Imagen

Gráfico de Coordenadas Polares
Datos Simulados

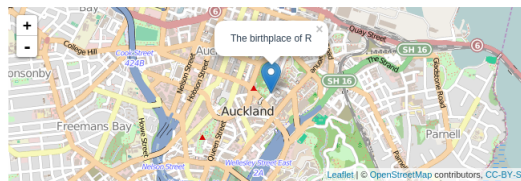


Descripción

Muestra un gráfico de líneas radiales, símbolos o un polígono con centro en el punto medio del gráfico en un gráfico circular de 0 a 360. Es un sistema de coordenadas bidimensional en el cual cada punto del plano se determina por una distancia y un ángulo, ampliamente utilizados en física y trigonometría.

Mapa geográfico leaflet

Imagen

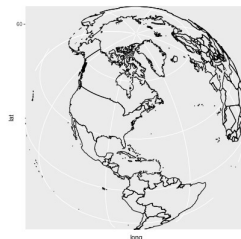


Descripción

Leaflet es una de las bibliotecas de JavaScript de código abierto más populares para crear los mapas interactivos. Es usado por los sitios web que van desde The New York Times y el Washington Post a GitHub y Flickr, así como especialistas en SIG como OpenStreetMap, Mapbox, y CartoDB.

Proyecciones cartográficas

Imagen



Descripción

La representación de una parte de la tierra, que es aproximadamente esférica, sobre un plano 2D plana requiere una proyección. Estas proyecciones representan el hecho de que la longitud real (en km) de un grado de longitud varía entre el ecuador y el polo.

Capítulo 3

Paquetes en R para la visualización de datos

En los últimos años la visualización de datos han ido adquiriendo una gran popularidad y el auge de las plataformas para su tratamiento ha supuesto un nuevo segmento de mercado en cuanto a Big Data se refiere. Por lo que actualmente existen numerosas plataformas y lenguajes que tratan de resolver la problemática actual al tratamiento de los datos. A continuación analizaremos la situación actual en base al lenguaje R. En cada sección se describirá un paquete que introduce funciones para la visualización de datos en R.

3.1. Utilización de paquetes en R

Gracias a que R forma parte de un proyecto colaborativo y abierto. Los usuarios publican paquetes que extienden su configuración básica. Existe un repositorio general en el cual van depositando todos los paquetes, actualmente existen 8594 paquetes. Dichos paquetes los podemos obtener en la siguiente página (<http://cran.r-project.org/>).

Para facilitar el desarrollo de nuevos paquetes, se ha puesto a servicio de la comunidad una forja de desarrollo que facilita las tareas relativas a dicho proceso (<http://r-forge.r-project.org/>).

Para la instalación de un paquete basta con escribir la siguiente sentencia en la consola de comandos `install.packages(nombre del paquete)`, automáticamente buscará el paquete a través de internet y se instalará.

3.2. Paquete base de R

El paquete base de R viene con la distribución del programa. Este paquete contiene las funciones básicas que permiten al entorno R trabajar como un idioma:, entrada/salida, soporte de programación básica aritmética, etc. Sus contenidos están disponibles a través de la herencia de cualquier entorno. Para obtener una lista completa sobre las funciones del paquete base basta con escribir

*CAPÍTULO 3. PAQUETES EN R PARA LA VISUALIZACIÓN DE DATOS*33

en la línea de comandos `library(help = 'base')`. Para el paquete base, existe el archivo R `HOME/library/base/R/base`. Este archivo está en formato ASCII y contiene todas las funciones del paquete. [17]

Capítulo 4

Análisis comparativo de los paquetes

En el presente capítulo vamos a indicar de qué manera podemos representar esas visualizaciones mediante el uso de las funciones en R. Para ello describiremos si dicha función se puede realizar si el requerimiento obligatorio de un paquete o por el contrario se encuentra en el paquete base de instalación de R.

Indicaremos si se encuentra en el paquete base o no, el nombre de la función, cuales son los argumentos que dispone así como su explicación de cada uno, los componentes en el caso de que existan, dónde podemos encontrarlo, si esa función necesita apoyarse de unos paquetes previos y por último la referencia en dónde se pueden encontrar esas funciones.

4.1. Diagramas

4.1.1. Árboles de Tallo y Hoja

Paquete base de R

Sí que lo tiene.

Función:

`stem()`

Argumentos son:

`(x, scale, width, atom)`

x: Vector numérico.

Scale: Esto controla la longitud de la trama.

Width: La anchura deseada de la trama.

Atom: Tolerancia.

Dónde encontrarlo en R

Paquete Graphics.

Paquetes requeridos:

N/A

Referencias:

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language. Wadsworth & Brooks/Cole.

<https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/stem.html>

4.1.2. Gráfico de Barras

Paquete base de R

Sí que lo tiene.

Función:

`barplot()`

Argumentos son:

(height,width, space, names.arg, legend.text, beside, horiz, density, angle, col , border, main , sub, xlab , ylab, xlim, ylim , xpd , log , axes, axisnames, cex.axis , cex.names, inside , plot , axis.lty , offset, add, args.legend, ...)

height: Ya sea un vector o matriz de valores que describen las barras que componen la trama. Si la altura es un vector, la trama consiste en una secuencia de barras rectangulares con alturas dadas por los valores en el vector. Si la altura es una matriz y al lado es FALSE entonces cada barra del gráfico corresponde a una columna de altura, con los valores de la columna dando las alturas de sub-barras apiladas que forman la barra. Si la altura es una matriz y al lado es TRUE, entonces los valores de cada columna se juxtaponen en vez de apilar.

width: Vector opcional de anchuras de barras. Reciclan a la longitud del número de barras dibujados. La especificación de un solo valor no tendrá ningún efecto visible a menos que se especifique xlim.

space: La cantidad de espacio (como una fracción de la anchura media de la barra) en la izquierda antes de cada barra. Se pueden administrar como un solo número o un número por barra. Si la altura es una matriz y al lado es TRUE, el espacio puede ser especificado por dos números, donde el primero es el espacio entre las barras en el mismo grupo, y el segundo el espacio entre los grupos. Si no se especifica explícitamente, el valor predeterminado es `c(0,1)` si la altura es una matriz y al lado es TRUE, y 0,2 al contrario.

`names.arg`: Un vector de nombres para ser trazada debajo de cada barra o un grupo de barras. Si se omite este argumento, entonces los nombres se toman del atributo `nombres` de altura si se trata de un vector, o los nombres de las columnas si se trata de una matriz.

`legend.text`: Un vector de texto utilizado para construir una leyenda para la trama, o indicando una lógica si una leyenda debe ser incluido. Esto sólo es útil cuando la altura es una matriz. En ese caso, las etiquetas de la leyenda dada deberían corresponder a las filas de altura; si `legend.text` es `TRUE`, los nombres de las filas de altura serán utilizados como etiquetas si son no nulo.

`beside`: Un valor lógico. Si es `FALSE`, las columnas de altura se presentan como barras apiladas, y si cumplen las columnas se presentan como barras yuxtapuestas.

`horiz`: Un valor lógico. Si es `FALSE`, las barras se dibujan verticalmente con la primera barra a la izquierda. Si es `TRUE`, las barras se dibujan horizontalmente con la primera en la parte inferior.

`density`: Un vector que da la densidad de las líneas de sombreado, en líneas por pulgada, para las barras o componentes de la barra. El valor predeterminado es `NULL`, significa que no hay líneas de sombreado que se dibujen. Los valores de no-positivos de la densidad también inhiben el dibujo de líneas de sombreado.

`angle`: La pendiente de las líneas de sombreado, dada como un ángulo en grados (hacia la izquierda), de los bares o componentes de la barra.

`col`: Un vector de colores para las barras o componentes de la barra. Por defecto, gris se utiliza si la altura es un vector, y una paleta de color gris corregida en gamma si la altura es una matriz.

`border`: El color que se utilizará para la frontera o borde de las barras. Utilice `border = NA` para omitir las fronteras o bordes. Si hay líneas de sombreado, `border = TRUE` significa que utilizan el mismo color tanto para el borde como para las líneas de sombreado.

`main,sub`: Título general y subtítulo de la trama.

`xlabs`: Una etiqueta para el eje X.

`ylabs`: Una etiqueta para el eje Y.

`xlim`: Límites para el eje X.

`ylim`: Límites para el eje Y.

`xpd`: Lógico, Se debe permitir barras fuera de la región

log: Lógico, cadena que especifica si escalas de los ejes deben ser logarítmica

axes: Lógico. Si es TRUE, una vertical (u horizontal, si es cierto horiz) se dibuja el eje.

axisnames: Lógico. Si es TRUE, y si hay names.arg (véase más arriba), se dibuja el otro eje (con lty = 0) y etiquetados.

cex.axis: Factor de expansión para las etiquetas de los ejes numéricos.

cex.names: Factor de expansión de nombres de eje (etiquetas de barras).

inside: Lógico. Si es TRUE, se dibujarán las líneas que dividen las barras adyacentes (no apilados!). Se aplica sólo cuando el espacio = 0 (que en parte es cuando al lado = VERDADERO).

plot: Lógico. Si es FALSO nada se muestra.

axis.lty: El parámetro gráfica lty aplicado al eje Y a las marcas del eje categórico (horizontal por defecto). Tenga en cuenta que por defecto el eje se suprime.

offset: Un vector que indica la cantidad de las barras que deben ser desplazadas con respecto al eje X.

add: Especificando una lógica, si las barras deben añadirse a una trama ya existente; por defecto es False.

args.legend: Lista de argumentos adicionales para pasar a la leyenda (); Los nombres de la lista se utilizan como nombres de argumentos. Sólo se utiliza si se suministra legend.text.

...: Argumentos que se pasan a/desde otros métodos. Para el método por defecto estos pueden incluir otros argumentos (tales como axes, asp y main) y los parámetros gráficos (véase par) que se pasan a plot.window(), title() y axis.

Dónde encontrarlo en R:

Paquete Graphics.

Paquetes requeridos:

N/A

Referencias:

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language. Wadsworth & Brooks/Cole.

Murrell, P. (2005) R Graphics. Chapman & Hall/CRC Press.

<https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/barplot.html>

4.1.3. Histograma

Paquete base de R

Sí que lo tiene.

Función:

`hist()`

Argumentos son:

`(x,Breaks,freq,probability,include.lowest,right,density,angle,col,border,main,xlab,ylab,xlim,ylim,axes,plot,`

`x`: Un vector de valores para el que se desea el histograma.

`Breaks`: Puede ser cualquiera de los siguientes; Un vector que da los puntos de ruptura entre las celdas del histograma. Una función para calcular el vector de puntos de corte. Un solo número que indica el número de celdas para el histograma. Una cadena de caracteres de nombres un algoritmo para calcular el número de celdas. En los tres últimos casos, el número es sólo una sugerencia; los puntos de corte se ajustan a los valores pretty. Si `breaks` es una función, el vector `X` se suministra a él como el único argumento.

`freq`: Si es TRUE, el gráfico del histograma es una representación de frecuencias, el componente de recuento de los resultados; si las densidades de probabilidad es FALSE, densidad de componentes, se representan gráficamente (de manera que el histograma tiene una superficie total de uno). El valor predeterminado es TRUE si y sólo si son equidistantes se rompe (y la probabilidad no se especifica).

`probability`: Un alias para `!freq`, para la compatibilidad de S.

`include.lowest`: Si es TRUE, una `x[i]` igual al valor de los breaks se incluirá en la primera (o última, por la derecha = FALSE) barra. Este será ignorado (con una advertencia) a menos que break sea un vector.

`right`: Si es TRUE, las celdas del histograma serán intervalos próximos a la derecha (izquierda abierta).

`density`: La densidad de líneas de sombreado, en líneas por pulgada. El valor predeterminado es NULL significa que no hay líneas de sombreado se dibujan. Los valores no positivos de la densidad también inhiben el dibujo de líneas de sombreado.

angle: La pendiente de las líneas de sombreado, dada como un ángulo en grados (sentido contrario a las agujas del reloj).

col: Un color que se utilizará para llenar las barras. El valor predeterminado es NULL produce barras sin rellenar.

border: El color del borde alrededor de las barras. El valor predeterminado es utilizar el color de primer plano estándar.

main, xlab, ylab: Estos argumentos del título son por defecto, muy útiles aquí.

xlim, ylim: El rango de valores de x e y con los parámetros por defecto. Tenga en cuenta que xlim no se utiliza para definir el histograma (breaks), pero sólo para el trazado (cuando plot= VERDADERO).

axes: Si es TRUE (default), los ejes se dibujan si se dibuja la trama.

plot: Si es TRUE (default), un histograma se representa, de lo contrario se devuelve una lista de los breaks y los counts. En este último caso, una advertencia se utiliza si no se especifica (por lo general gráficas) argumentos que sólo se aplica a la trama = TRUE.

labels: Cadena lógica o carácter. Además dibuja las etiquetas en la parte superior de las barras, si no es FALSE.

nclass: Numérico (entero). Para S (-PLUS) la compatibilidad única, NClass es equivalente a los breaks para un argumento escalar o carácter.

warn.unused: Si plot = FALSE y warn.unused = TRUE, una advertencia se emite cuando se pasan los parámetros gráficas para hist.default ().

... : Más argumentos y parámetros gráficos pasados a plot.histogram y de allí a title y axis (si plot = VERDADERO).

Componentes:

breaks: los límites $n + 1 =$ celdas fronteras(se rompe si eso era un vector). Estos son los descansos nominales, no con límite de la traza.

counts: n enteros; para cada célula, el número de $x[]$ en el interior.

mids: El punto medio de celdas n.

xname: Una cadena de caracteres con el actual nombre del argumento x.

equidist: Logica, lindicando si las distancias entre los breaks son todos iguales..

Dónde encontrarlo en R:

Paquete Graphics.

Paquetes requeridos:

N/A

Referencias:

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language. Wadsworth & Brooks/Cole.

Venables, W. N. and Ripley. B. D. (2002) Modern Applied Statistics with S. Springer.

<https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/hist.html>

4.1.4. Boxplot o Caja de Turkey**Paquete base de R**

Sí que lo tiene.

Función:

`boxplot()`

Argumentos: (x, range, width, varwidth, notch, outline, names, boxwex, staplex, outwex, plot, border, col, log, pars, horizontal, add, at)

x: Para especificar los datos en el que los diagramas de caja se van a producir. Ya sea un vector numérico o una lista única que contiene dichos vectores.

range: Esto determina hasta qué punto los bigotes de la trama se extienden hacia fuera de la caja. Si el rango es positivo, los bigotes se extienden hasta el punto de datos más extremo.

width: Indica la anchura relativa de las cajas que componen la trama.

varwidth: Si varwidth es TRUE, las cajas se dibujan con anchuras proporcional al cuadrado de base del número de observaciones en los grupos.

notch: Si notches TRUE, una muesca se dibuja en cada lado de las cajas.

outline: Si outline no es TRUE, los valores atípicos no se dibujan.

names: Etiquetas de grupo que se van a imprimir en cada diagrama de caja. Puede ser un vector de caracteres o una expresión.

boxwex: Un factor de escala que se aplica a todas las cajas. Cuando hay sólo unos pocos grupos, la aparición de la trama puede mejorarse haciendo las cajas más estrecho.

staplewex: Expansión de anchura de la línea de grapas, proporcional al ancho de la caja.

outwex: expansión de anchura de la línea de valores atípicos, proporcional al ancho de la caja.

plot: si es TRUE (por defecto) a continuación un gráfico de caja se mostrará. Si no es así, se devuelven los resúmenes de los diagramas de caja.

border: Un vector opcional de los colores para los contornos de los gráficos de caja. Los valores en la frontera o borde se reciclan si la longitud del borde es menor que el número de parcelas.

col: Si col no es nulo, se supone que contiene colores que se utilizarán para dar color a los cuerpos de los diagramas de caja. Por defecto están en el color de fondo.

log: Carácter que indica si X o Y o ambas coordenadas deben ser trazados en escala logarítmica.

pars: Una lista de muchos (potencialmente) más parámetros gráficos, por ejemplo, **boxwex** o **ouch**; éstos se pasan a **bxp** (si trama es cierto); para más detalles, véase allí.

horizontal: Lógica que indica si los diagramas de caja deben ser horizontal; FALSO por defecto significa cajas verticales.

add: Lógica, si es TRUE agregar gráfico de caja al gráfico actual.

at: Vector numérico dando a los lugares en los que los diagramas de caja deben ser dibujados, sobre todo cuando **add** = TRUE; por defecto es 1: n, donde n es el número de cajas.

Componentes:

stats: Una matriz, cada columna contiene el extremo del bigote inferior, la bisagra inferior, la mediana, la bisagra superior y el extremo de la barba superior para un grupo/trama. Si todas las entradas tienen el mismo atributo de clase, también lo hará este componente.

n: Un vector con el número de observaciones en cada grupo.

conf: Una matriz en la que cada columna contiene los extremos inferior y superior de la muesca.

out: Los valores de los puntos de datos que están más allá de los extremos de los bigotes.

group: Un vector de la misma longitud como out cuyos elementos indican a qué grupo pertenece el valor atípico.

names: Un vector de nombres para los grupos.

Dónde encontrarlo en R:

Paquete Graphics.

Paquetes requeridos:

N/A

Referencias:

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language. Wadsworth & Brooks/Cole.

Chambers, J. M., Cleveland, W. S., Kleiner, B. and Tukey, P. A. (1983) Graphical Methods for Data Analysis. Wadsworth & Brooks/Cole.

Murrell, P. (2005) R Graphics. Chapman & Hall/CRC Press.

<https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/boxplot.html>

4.1.5. Diagrama de pareto**Paquete base de R**

No lo tiene.

Función:

`pareto.chart()`

Argumentos son:

(x, ylab, ylab2, xlab, cumperc, ylim, main, col, plot, ...)

x: Un vector de valores. `names(x)` que se utilizan para etiquetar las barras.

ylab: Una cadena que especifica la etiqueta para el eje Y.

ylab2: Una cadena que especifica la etiqueta del segundo eje Y en el lado derecho.

xlab: Una cadena que especifica la etiqueta para el eje X.

cumperc: Un vector de valores porcentuales para ser utilizado como marcas de graduación para el segundo eje Y, en el lado derecho.

ylim: Un vector numérico que especifica los límites para el eje Y.

main: Una cadena que especifica el título principal que aparece en la trama.

col: Un valor para el color, un vector de colores, o una gama de colores para las barras. Consulte la ayuda para los colores y la paleta.

plot: Les proporcionará una lógica si debe proporcionarse un gráfico (TRUE, por defecto) o simplemente una tabla resumen deben ser devueltos.

...: otros argumentos, gráficas que se pasan a la función diagrama de barras.

Dónde encontrarlo en R:

Paquete 'qcc'.

Paquetes requeridos:

'qcc'.

Referencias:

<https://cran.r-project.org/web/packages/qcc/qcc.pdf>

4.1.6. Gráfico de densidad**Paquete base de R**

No lo tiene. .

Función:

density()

Argumentos son:

(x, bw, adjust, kernel, weights, window, width, give.Rkern, n, from, to, cut, na.rm, ...)

x: Los datos cuales la estimación se va a calcular.

bw: El ancho de banda de suavizado que se utilizará. Los núcleos se ajustan a escala de tal manera que esta es la desviación estándar del núcleo de suavizado. (Tenga en cuenta esta diferencia en los libros de referencia que se citan a continuación, y S-PLUS.). **bw** también puede ser una cadena de caracteres dando una regla para seleccionar el ancho de banda. Ver **bw.nrd**. El valor por defecto, **'nrd0'**, se ha mantenido el valor por defecto por razones históricas y de compatibilidad, más que como una recomendación general, en donde, por ejemplo, **'SJ'** preferiría ajuste, consulta Venables y Ripley (2002). El valor especificado (o computarizada) de **bw** se multiplica por **ajust**.

adjust: El ancho de banda utilizado es en realidad $\text{adjust} * \text{bw}$. Esto hace que sea fácil para especificar los valores de ancho de banda como 'la mitad el valor por defecto'.

kernel, window: Una cadena de caracteres que da el núcleo suavizado para ser utilizado. Este debe coincidir parcialmente uno de 'gaussiana', 'rectangular', 'triangular', 'Epanechnikov', 'biweight', 'cosine' o 'optcosine', con defecto 'gaussiana', y puede ser abreviada a un prefijo único (single carta). 'Cosine' es más suave que 'optcosine', que es el núcleo de costumbre 'Cosine' en la literatura y casi MSE-eficiente. Sin embargo, 'cosine' es la versión utilizada por S.

weights: Vector numérico de los pesos de observación no negativos, por lo tanto, de misma longitud que **X**. El valor por defecto es **NULL** equivalente a $\text{weights} = \text{rep}(1 / \text{nx}, \text{nx})$, donde **nx** es la longitud de (las entradas finitas de) **x[]**.

width: Esto existe para la compatibilidad con S; si se da, y **bw** no está, establecerá **bw** a la anchura si se trata de una cadena de caracteres, o a un múltiplo kernel-dependiente de la anchura si este es numérico.

give.Rkern: Lógico; si es **TRUE**, no se estima la densidad, y el 'ancho de banda canónica' del núcleo elegido se devuelve en su lugar.

n: El número de puntos igualmente espaciados en el que la densidad es a estimar. Cuando **n** es mayor a 512, se redondea a una potencia de 2 en los cálculos (como se usa FFT) y el resultado final se interpola en aprox. Por lo que casi siempre tiene sentido para especificar **n** como una potencia de dos.

from,to: Los puntos más a la derecha y la izquierda de la cuadrícula en la que la densidad ha de estimarse; los valores por defecto se cortan $* \text{bw}$ fuera de $\text{range}(\mathbf{x})$.

cut: Por defecto, los valores de desde y hacia allá se cortan los extremos anchos de banda de los datos. Esto permite que la densidad estimada caiga aproximadamente a cero en los extremos.

na.rm: Lógico; si es **TRUE**, los valores no se eliminan de **X**. Si **FALSE** valores que faltan causan un error.

...: Más argumentos para los métodos (no predeterminados).

Dónde encontrarlo en R:

Paquete 'stats'.

Paquetes requeridos:

'graphics'.

Referencias:

<https://stat.ethz.ch/R-manual/R-devel/library/stats/html/density.html>

4.1.7. Pie Charts**Paquete base de R**

Sí que lo tiene.

Función:

`pie()`

Argumentos son:

`(x, labels, edges, radius, clockwise, init.angle, col, main, ...)`

`x`: Un vector de números positivos. Estos valores representan el área de las secciones representadas.

`labels`: Una o más expresiones en forma de cadena de texto que dará nombre a cada variable.

`edges`: El perímetro del círculo es determinado por un polígono con la cantidad especificada de aristas.

`radius`: El círculo es centrado en un cuadrado cuyos lados están a un rango de -1 to 1. Si las etiquetas son demasiado largas, será conveniente especificar un radio menor.

`clockwise`: Operador lógico indicando si los segmentos se dibujaran en dirección contraria a las agujas del reloj o no.

`init.angle`: Número, especificando el ángulo de inicio en grados.

`col`: Un vector de colores que se usará para determinar el color de los sectores. Por defecto se usan 6 tonos pastel.

`main`: Un título general para la gráfica.

...: Parámetros adicionales para modificar la apariencia.

Dónde encontrarlo en R:

Paquete 'ggplot2'.

Paquetes requeridos:

'ggplot2'.

Referencias:

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language. Wadsworth & Brooks/Cole.

Cleveland, W. S. (1985) The Elements of Graphing Data. Wadsworth: Monterey, CA, USA.

4.1.8. Gráfico de Puntos**Paquete base de R**

Sí que lo tiene.

Función:

stripchart()

Argumentos son:

(x, data, dlab, ..., subset, na.action)

x: Los datos que representará la gráfica. Por defecto, las entradas de datos aceptadas son un vector numérico, o una lista de vectores numéricos, cada uno se corresponderá con un componente de la gráfica. Si no empleamos la entrada por defecto, podemos usar una fórmula, una especificación simbólica tal que y g. Esto indica que las observaciones en el vector y deben ser agrupadas teniendo en cuenta los niveles especificados en el factor g. Se permiten valores NAs.

data: Un data.frame (o list) de dónde las variables en X deben ser cogidas.

subset: Un vector opcional para especificar un subset de observaciones para ser representadas.

na.action: Una acción que debe indicar qué hacer en caso de que los datos contengan valores NAs(nulos o no válidos). Por defecto se ignoran dichos valores.

...: Parámetros adicionales que se incluyen para cambiar la estética de la gráfica.

method: El método que se usará para diferenciar puntos coincidentes. Por defecto se usa `overplot`, pero es posible especificar por ejemplo, `jitter` o `stack`.

vertical: Cuando `vertical` es `TRUE`, los puntos se representan verticalmente.

add: Valor booleano, si es verdadero, la representación se añade al gráfico actual.

Dónde encontrarlo en R:

Paquete `'ggplot2'`.

Paquetes requeridos:

`'ggplot2'`.

Referencias:

<https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/stripchart.html>

4.1.9. Matriz de balón

Paquete base de R

No lo tiene.

Función:

`balloonplot()`

Argumentos son:

(`x`, `y`, `z`, `xlab`, `ylab`, `zlab`, `dotsize`, `strheight`, `dotchar`, `dotcolor`, `text.size`, `text.color`, `main`, `label`, `label.digits`, `label.size`, `label.color`, `scale.method`, `scale.range`, `colsrt`, `rowsrt`, `colmar`, `rowmar`, `show.zeros`, `show.margins`, `cum.margins`, `sorted`, `label.lines`, `fun`, `hide.duplicates`, ...)

x: Un objeto tabla, un vector, o una lista de vectores categóricos conteniendo un grupo de variables para los primeros márgenes de X de la matriz representada.

y: Un vector o una lista de vectores para agrupar variables para la segunda dimensión y representada.

z: Un vector de valores para el tamaño de los puntos representados en la imagen.

xlab: Etiqueta de texto para la dimensión 'X'. Se mostrará en el eje X Y en el título.

ylab: Etiqueta de texto para la dimensión 'Y'. Se mostrará en el eje X Y en el título.

zlab: Etiqueta de texto para el punto. Se mostrará en el título.

dotsize: Tamaño máximo del punto.

dotchar: Carácter que será usado para representar el punto.

dotcolor: Vector especificando el color o colores que tomaran los puntos.

text.size, text.color: Tamaño y color del texto para las cabeceras de las filas y columnas.

label: Flag lógico que indica si los valores actuales de los elementos deben ser mostrados.

label.digits: Número de dígitos usados para formatear las etiquetas de los valores.

label.size, label.color: Tamaño y color del texto para las etiquetas de los valores.

scale.method: Método para escalar el tamaño de los puntos, bien por su volumen o por su diámetro.

rowsrt, colsrt: ángulo de rotación de las etiquetas de filas y columnas.

rowmar, colmar: Espacio reservado para las etiquetas de las filas y columnas. Cada unidad es la anchura/altura de una celda de la tabla.

show.zeros: Lógico. Si es FALSE, las entradas que estén a cero se dejarán blancas en la representación. Si es TRUE se representarán.

show.margins: Lógico. Si es TRUE, la suma de las filas y columnas se muestra en los bordes, debajo y a la derecha de la imagen.

sorted: Lógico. Si es TRUE, las filas serán ordenadas usando los niveles del primer factor 'y', después del segundo factor 'y'... De la misma manera para el factor 'x'.

label.lines: Lógico. Si TRUE, los bordes serán dibujados para las cabeceras de las filas y columnas.

hide.duplicates: Lógico. Si TRUE, las cabeceras de columnas y las filas omitirán duplicados. Por defecto a TRUE.

...: Argumentos adicionales.

Dónde encontrarlo en R:

Paquete 'gplots'.

Paquetes requeridos:

'gplots'.

Referencias:

<https://cran.r-project.org/web/packages/gplots/gplots.pdf>

4.1.10. Kernel**Paquete base de R**

No lo tiene.

Función:

`ksmooth()`

Argumentos son:

(x, y, kernel, bandwidth, range.x, n.points, x.points)

x: Entrada para valores de x. Se aceptan vectores largos.

y: Entrada para valores de y. Se aceptan vectores largos.

kernel: El kernel a usar. Puede ser abreviado.

bandwidth: El ancho de banda. Los kernels son escalados para que sus cuartiles (visto como probabilidad de densidades) estén al $\pm 0.25 \cdot \text{bandwidth}$.

range.x: El rango de puntos que será cubierto en el formato de salida.

n.points: El numero de puntos a los cuales se evaluará el ajuste.

x.points: Puntos a los cuales se evaluará el ajuste suavizado. Si no se especifica, n.points son leídos uniformemente para cubrir el range.x. Se soportan vectores largos.

Componentes:

x: Valores a los cuales se evaluará el ajuste del suavizado. En orden creciente.

y: Valores ajustados que corresponden a X.

Dónde encontrarlo en R:

Paquete 'Stats'.

Paquetes requeridos:

'stats', 'graphics'.

Referencias:

<https://cran.r-project.org/doc/contrib/grafi3.pdf>

4.1.11. Sunflowerplot**Paquete base de R**

Si lo tiene

Función:

`sunflowerplot()`

Argumentos son:

(x, y, number, log, digits, xlab, ylab, xlim, ylim, add, rotate, pch, cex, cex.fact, col, bg, size, seg.col, seg.lwd, ...)

x: Vector numérico de coordenadas X de longitud n, por ejemplo, u otra estructura válida, como para `plot.default`, ver también `xy.coords`.

y: Vector numérico de coordenadas de longitud n.

number: Vector de enteros de longitud n. Número [i] = número de repeticiones para (x[i], y[i]), puede ser 0. Por defecto (`missing(number)`): calcular la multiplicidad exacta de los puntos x[], y[], a través de la tabla `XY()`.

log: Carácter de registro que indica la coordenada escala, ver `plot.default`.

digits: Cuando se calcula el número (ejemplo, no especificado), X e Y se redondean a los dígitos significativos antes de calcular las multiplicidades.

xlab, ylab: Etiqueta de caracteres para X, o eje Y, respectivamente.

xlim, ylim: Numérico (2) que limita la extensión de X, o el eje Y.

add: Lógica; debe añadirse el diagrama en una anterior. El valor predeterminado es `FALSE`.

rotate: Lógica; si es `TRUE`, gira aleatoriamente los girasoles (Impidiendo artefactos).

pch: Carácter mostrado que se utilizará para los puntos (número [i] == 1) y el centro de girasoles.

cex: Numérico; el tamaño de expansión del carácter de los puntos centrales (similar a pch).

cex.fact: Factor de encogimiento numérico que se utilizará para los puntos del centro cuando hay hojas de flores, es decir, $cex / cex.fact$ se utiliza para esto.

col, bg: Colores para los símbolos impresos, pasados por plot.default.

size: Tamaño de los girasoles en pulgadas, 1[in] := 2.54[cm]. Default: 1/8; aproximadamente 3.2mm.

seg.col: Color para ser utilizado en los segmentos que forman las hojas de los girasoles, ver par (col =); col = 'gold2' recuerda a girasoles reales.

seg.lwd: Numérico; el ancho de línea para los segmentos de las hojas.

...: Más argumentos para mostrar [si add = FALSE], o para ser transmitidos hacia o desde otro método.

Dónde encontrarlo en R:

Paquete 'graphics'.

Paquetes requeridos:

'graphics'.

Referencias:

<https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/sunflowerplot.html>

Chambers, J. M., Cleveland, W. S., Kleiner, B. and Tukey, P. A. (1983) Graphical Methods for Data Analysis. Wadsworth.

Schilling, M. F. and Watkins, A. E. (1994) A suggestion for sunflower plots. The American Statistician, 48, 303305.

Murrell, P. (2005) R Graphics. Chapman & Hall/CRC Press.

4.1.12. smoothScatter

Paquete base de R

Si lo tiene

Función:

`smoothScatter()`

Argumentos son:

(`x`, `y`, `nbin`, `bandwidth`, `colramp`, `nrpoints`, `ret.selection`, `pch`, `cex`, `col`, `transformation`, `postPlotHook`, `xlab`, `ylab`, `xlim`, `ylim`, `xaxs`, `yaxs`, ...)

`x`, `y`: Argumentos que proporcionan la coordenadas X e Y de la trama. Cualquier forma razonable de la definición de las coordenadas es aceptable. Ver las `xy.coords` de función para obtener más detalles. Si se suministra por separado, deben ser de la misma longitud.

`nbin`: Vector numérico de longitud uno (en ambos sentidos) o dos (para X e Y por separado) que especifica el número de puntos de cuadrícula igualmente espaciados para la estimación de la densidad

`bandwidth`: Vector numérico (longitud 1 o 2) de ancho de banda suavizado(s). Si es ausente, es usado uno por defecto. el ancho de banda después es pasado a la función `bkde2D`.

`colramp`: Función de la aceptación de un entero `n` como argumento y devuelve `n` colores.

`nrpoints`: Número de puntos que se superpone sobre la imagen de la densidad. Los primeros puntos de aquellas áreas de densidades más bajas regionales serán impresos. Añadir puntos a la impresión permite la identificación de los valores atípicos. Si todos los puntos deben ser impresos, elija `nrpoints = Inf`.

`pch`, `cex`, `col`: Argumentos que se pasan a los puntos cuando `NPOINTS` > 0: Símbolo de punto, factor de expansión de carácter y color, consulte también `par`.

`transformation`: Función de mapeo de la escala de densidad de la escala de colores.

`postPlotHook`: NULL o una función que será llamada (sin argumentos) después de la imagen.

`xlab`, `ylab`: Cadenas de caracteres que se utilizan como etiquetas de los ejes, pasaron a la imagen.

`xlim`, `ylim`: Vectores numéricos de los límites de longitud de 2 ejes especificando.

`xaxs`, `yaxs`, ...: Más argumentos, pasados a la imagen.

Dónde encontrarlo en R:

Paquete 'graphics'.

Paquetes requeridos:

'graphics'.

Referencias:

<https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/smoothScatter.html>

4.1.13. Scatterplot**Paquete base de R**

No lo tiene.

Función:

Scatterplot()

Argumentos son:

(x, y ,)

x: Vector de coordenadas horizontales (o primer argumento de la función genérica).

y: Vector de coordenadas verticales.

formula: Una formula 'modelo', con una estructura $Y \sim X$ o (para representar por grupos) $y \sim x \mid z$, donde Z hace referencia a un factor u otra variable que permite dividir los datos en grupos. Si X es un factor, entonces se producen varios diagramas de cajas paralelos utilizando la función Boxplot.

data: Data frame con el que se evalúa la fórmula.

subset: Expresión que define un subconjunto de observaciones.

smoother: Una funcion para dibujar una regresión no paramétrica; por defecto es loessLine , lo que hace el suavizado de loess . La función gamLine ajusta un modelo aditivo generalizado y permite incluir un enlace y una función de error. Ver ScatterplotSmoothers. Al establecer este argumento para algo que no sea una función, por ejemplo, FALSE suprime el suavizador.

smoother.args: Una lista de valores con nombre que se pasa a la función; éstos dependerán del factor smoother (ver ScatterplotSmoothers).

`smooth`, `span`: Estos argumentos se incluyen por motivos de compatibilidad : si `smooth = TRUE` entonces `smoother` se aplica a `loessLine` , y si se especifica rango, se añade a `smoother.args` .

`spread`: Si es `TRUE`, estima la raíz cuadrada de la función de varianza. Para `LoessLine` y `gamLine`, esto se hace suavizando los cuadrados de los residuos tanto positivos como negativos de la media de ajuste por separado y luego suma la raíz cuadrada de los valores ajustados a el ajuste medio.

`reg.line`: Función para dibujar una linea de regresión en la impresión o ponerla a `FALSE` para que no se represente.

`boxplots`: Si se especifica 'X' se crea un boxplot para X debajo de la impresión; si se especifica 'Y' se crea un boxplot para Y a la izquierda de la impresión; si se especifican ambos, 'xy', se imprimen ambos boxplot; si se deja como " o `FALSE` se suprimirían.

`xlab`: Etiqueta para el eje horizontal.

`ylab`: Etiqueta para el eje vertical.

`las`: Si es 0, las etiquetas de ayuda son dibujadas paralelas al eje, si se especifica como 1 se mostrarán etiquetas horizontales (ver par).

`lwd`: Anchura de las lineas de la regresión lineal (por defecto es 1).

`lty`: Tipo de las lineas de la regresión lineal (default 1, solid line).

`id.method`, `id.n`, `id.cex`, `id.col`: Argumentos para etiquetar los puntos. Por defecto está como `id.n=0` para no etiquetar ningún punto. Véase `showLabels` para más detalles sobre este argumento. Si la impresión utiliza varios colores para representar los grupos, el argumento `id.col` es ignorado y las etiquetas de colores se especifican con el argumento `col`.

`labels`: Un vector con las etiquetas de los puntos; si no se especifica, la función intenta determinar unas etiquetas acordes, y, si no lo consigue, emplea números de la observación.

`jitter`: Una lista con elementos X e Y , especificando valores de fluctuación para las coordenadas horizontales y verticales de los puntos en el scatterplot. La función de fluctuación es empleada para distorsionar los puntos aleatoriamente; por defecto 1. Las lineas ajustadas no son afectadas por este argumento.

`xlim`: Los límites de x(min, max) de la impresión; si es `NULL`, será determinado por los datos.

`ylim`: Los límites de y(min, max) de la impresión; si es `NULL`, será determinado por los datos.

`groups`: Un factor u otra variable que divide los datos en grupos; los grupos son impresos con diferentes colores y diferentes símbolos.

`by.groups`: Si es TRUE, Las líneas de regresión son ajustadas por grupos.

`legend.title`: El título de la leyenda; por defecto es el nombre de la variable de grupo.

`legend.coords`: Coordenadas para emplazar al leyenda; puede ser una lista con componentes X e Y que especifiquen las coordenadas de la esquina superior izquierda de la leyenda; O una palabra clave entrecomillada, como por ejemplo 'topleft'.

`legend.columns`: Número de columnas que tendrá la leyenda; Si no se especifica se representará horizontalmente encima del gráfico.

`ellipse`: Si es TRUE, se muestran elipses con concentraciones de datos.

`levels`: Nivel o niveles en los que las elipses de concentración son impresas, por defecto es c(.5, .95).

`col`: Colores para las líneas y puntos; los colores por defecto son tomados de la paleta de colores estándar, con `palette()[3]` para las líneas de regresiones lineales, `palette()[2]` para regresiones lineales no paramétricas, y `palette()[1]` para puntos si no hay grupos, y así sucesivamente para los colores e los grupos en caso de que los hubiere.

`cex`, `cex.axis`, `cex.lab`, `cex.main`, `cex.sub`: Establece las medidas de varios elementos gráficos; (ver par).

`legend.plot`: Si es TRUE, se mostrará una leyenda de los grupos impresos en la parte superior.

... otros argumentos que pueden ser indicados para la impresión.

`grid`: Si es TRUE, el por defecto, una cuadrilla gris claro, es desplegada en el fondo de la impresión.

Dónde encontrarlo en R:

Paquete 'car'.

Paquetes requeridos:

'car'.

Referencias:

<https://cran.r-project.org/web/packages/car/car.pdf>

4.1.14. Scatterplot3D

Paquete base de R

No lo tiene.

Función:

Scatterplot3D()

Argumentos son:

scatterplot3d(x, y, z, color, pch, main, sub, xlim=NULL, ylim, zlim, xlab, ylab, zlab, scale.y, angle, axis, tick.marks, label.tick.marks=TRUE, x.ticklabs, y.ticklabsL, z.ticklabs, y.margin.add, grid, box, lab, lab.z, type, highlight.3d, mar, bg, col.axis, col.grid, col.lab, cex.symbols, cex.axis, cex.lab, font.axis, font.lab, lty.axis, lty.grid, lty.hide, lty.hplot, log, ...)

x: Las coordenadas de puntos X en la impresión.

y: Las coordenadas Y de los puntos en la trama, opcional si x es una estructura adecuada.

z: Las coordenadas de los puntos en la trama, opcional si x es una estructura apropiada.

color: Colores de los puntos del diagrama, opcional si x es una estructura apropiada. Será ignorado si highlight.3d = TRUE.

pch: Impresión del carácter que se va a utilizar.

main: Un título global para el diagrama.

sub: Subtítulo para el diagrama.

xlim, ylim, zlim: Las x, y, z límites(min, max) del diagrama. Tenga en cuenta que establecer límites ampliados pueden no funcionar exactamente como se esperaba (un conocido, pero no fijado error).

xlab, ylab, zlab: Títulos para los ejes X, Y y Z

scale.y: Escala del eje Y en relación con X y eje Z.

angle: ángulo entre los ejes X e Y (Atención: resultado depende de la escala).

axis: Un valor lógico que indica si los ejes debes constituirse sobre el diagrama.

tick.marks: Un valor lógico que indica si las marcas de graduación debe constituirse sobre el diagrama (sólo si el eje = VERDADERO).

`label.tick.marks`: Un valor lógico que indica si las marcas de graduación deben ser etiquetados en el gráfico (sólo si `eje = TRUE` y `tick.marks = VERDADERO`).

`x.ticklabs`, `y.ticklabs`, `z.ticklabs`: Vector de etiquetas de marcas de graduación.

`y.margin.add`: Añadir espacio adicional entre las etiquetas de marcas de graduación y etiqueta del eje del eje Y.

`grid`: un valor lógico que indica si una cuadrícula debe constituirse sobre el diagrama.

`box`: Un valor lógico que indica si una caja debe ser dibujado alrededor del gráfico.

`lab`: Un vector numérico de la forma `c(x, y, len)`. Los valores de `x` e `y` dar el (Aproximado) número de marcas en los ejes `x` e `y`.

`lab.z`: Lo mismo que `lab`, pero para el eje `z`.

`type`: Carácter que indica el tipo de diagrama: `'p'` por los puntos, `'l'` para las líneas, `'h'` para las líneas verticales a `x-y-plane`, etc.

`highlight.3d`: Los puntos serán dibujados en diferentes colores relacionados con coordenadas `y` (sólo si `type = 'p'` o `type = 'h'`, serán utilizados el color más). En algunos dispositivos no todos los colores se pueden mostrar. En este caso probar el dispositivo PostScript o utilizar `highlight.3d = FALSO`.

`mar`: Un vector numérico de la forma `c(bottom, left, top, right)` que da a las líneas de margen es específica en los cuatro lados de el diagrama.

`bg`: Fondo (relleno) de color para los símbolos diagrama abiertas dadas por `PCH = 21:25`.

`col.axis`, `col.grid`, `col.lab`: El color que se utilizará para las etiquetas de eje / cuadrícula / eje.

`cex.symbols`, `cex.axis`, `cex.lab`: El aumento que se utiliza para los símbolos de puntos, la anotación de eje, etiquetas con respecto a la corriente.

`font.axis`, `font.lab`: El tipo de letra que se utilizará para el eje de anotación / etiquetas.

`lty.axis`, `lty.grid`: El tipo de línea que se utilizará para el eje / cuadrícula.

`lty.hide`: Estilo de línea utilizado para trazar los bordes 'no visibles' (por defecto del estilo `lty.axis`)

lty.hplot: El tipo de línea que se utilizará para segmentos verticales con type = 'h'.

log: Aun no implementado! Una cadena de caracteres que contiene 'x' (si el eje x es ser logarítmica), 'y', 'z', 'XY', 'xz', 'yz', 'XYZ'.

... Más parámetros gráficos pueden darse como argumentos, pch = 16 o pch = 20 puede ser vistoso.

Dónde encontrarlo en R:

Paquete 'Scatterplot3D'.

Paquetes requeridos:

'Scatterplot3D'.

Referencias:

<https://cran.r-project.org/web/packages/scatterplot3d/scatterplot3d.pdf>

4.1.15. Diagrama de violín

Paquete base de R

No lo tiene.

Función:

geom_violin()

Argumentos son:

(mapping, data, stat, position, ..., draw_quantiles, trim, scale, na.rm, show.legend, inherit.aes)

mapping: Conjunto de asignaciones estéticas creadas por los aes o aes.. Si se especifica y inherit.aes = TRUE (por defecto), se combina con la asignación pre-determinada en el nivel superior del diagrama. Debe proporcionar esta función si no hay ninguna asignación trama.

data: Los datos que se muestran en esta capa. Hay tres opciones: Si es NULL, el valor por defecto, los datos se heredan de los datos del diagrama según lo especificado en la llamada a ggplot. Un data.frame, u otro objeto, tienen prioridad sobre los datos del gráfico. Todos los objetos se fortalecerán para producir una trama de datos. Ver fortify por el que se crearán las variables. Una función será llamada con un solo argumento, los datos del gráfico. El valor de retorno debe ser un data.frame., Y será utilizado como la capa de datos.

position: Ajuste de la posición, ya sea en forma de cadena, o el resultado de una llamada a una función de ajuste de posición.

...: otros argumentos pasados a la capa. Estos son a menudo la estética, se utiliza para establecer una estética a un valor fijo, como el `color = 'red'` o `size = 3`. También puede ser parámetros para el emparejado `geom/stat`.

`draw_quantiles`: Si no se introduce es (NULL) (por defecto), dibuja líneas horizontales en los cuartiles dados de la estimación de la densidad.

`trim`: Si es TRUE (por defecto), recortar las colas de los violines al rango de los datos. Si es FALSE, no recorta las colas.

`scale`: si `'area'` (por defecto), todos los violines tienen la misma área (antes de recortar las colas). Si `'Count'`, las áreas se ajustan a escala en proporción al número de observaciones. Si `'width'`, todos los violines tienen la misma anchura máxima.

`na.rm`: Si es FALSO (por defecto), elimina los valores ausentes con una advertencia. Si es TRUE eliminar los valores que faltan sin mostrar advertencia.

`show.legend`: Lógico. En caso de que esta capa esté incluida en las leyendas. NA, el valor por defecto, si se asignan incluye cualquier estética. FALSE nunca incluye, y TRUE incluye siempre.

`inherit.aes`: Si es FALSE, anula la estética por defecto, en lugar de combinar con ellos. Esto es muy útil para funciones auxiliares que definen tanto los datos como la estética y no debe heredar el comportamiento de la especificación de gráfico predeterminado, por ejemplo, bordes.

`geom, stat`: Utilizar para anular la conexión predeterminada entre `geom_violin` y `stat_ydensity`.

Dónde encontrarlo en R:

Paquete `'ggplot2'`.

Paquetes requeridos:

`'ggplot2'`.

Referencias:

<https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf>

4.1.16. Series de Tiempo

Paquete base de R

No lo tiene.

Función:

`ts()`

Argumentos son:

(data, start, end, frequency, deltat, ts.eps, class, names)

data: Un vector o matriz de los valores de series temporales observadas. Una trama de datos será forzado a través de una matriz numérica data.matrix. (Ver también 'Details'.)

start: El momento de la primera observación. Cualquiera de un número único o un vector de dos números enteros.

end: El momento de la última observación, se especifica en la misma forma que start.

frequency: El número de observaciones por unidad de tiempo.

deltat: La fracción del periodo de muestreo entre las observaciones sucesivas; por ejemplo, 1/12 para los datos mensuales. Sólo uno de frequency o deltat debe ser proporcionado.

ts.eps: Tolerancia de comparación de serie de tiempo. Las frecuencias se consideran iguales si su diferencia absoluta es menor que ts.eps.

class: La clase que ha de darse al resultado, o ninguno si NULL o 'none'. El valor por defecto es 'ts' para una sola serie, C ('MTS', 'ts', 'Matrix') para múltiples series.

names: Un vector de caracteres de los nombres de la serie en una serie múltiple: por defecto las columnas de datos, o la Serie 1, Serie 2,

x: Un objeto R arbitrario.

...: Argumentos que se pasan a los métodos (no utilizados para el método por defecto).

Dónde encontrarlo en R:

Paquete 'stats'.

Paquetes requeridos:

'stats'.

Referencias:

<https://stat.ethz.ch/R-manual/R-devel/library/stats/html/ts.html>

4.1.17. Asociación Cohen-Friendly

Paquete base de R

Si lo tiene

Función:

`assocplot()`

Argumentos son:

`(x, col, space, main, xlab, ylab)`

`x`: Una tabla de contingencia de dos dimensiones en forma de matriz.

`col`: Un vector de caracteres de longitud dos que dan los colores utilizados para la elaboración de residuos positivos y negativos de Pearson, respectivamente.

`space`: La cantidad de espacio (como una fracción de la anchura media rectángulo y altura) que queda entre cada rectángulo.

`main`: Título general del diagrama.

`xlab`: Una etiqueta para el eje X. El valor predeterminado es el nombre (si lo hay) de la dimensión de las filas de X.

`ylab`: Una etiqueta para el eje Y. El valor predeterminado es el nombre (si lo hay) de la dimensión de la columna en X.

Dónde encontrarlo en R:

Paquete 'graphics'.

Paquetes requeridos:

'graphics'.

Referencias:

Cohen, A. (1980), On the graphical display of the significant components in a two-way contingency table. *Communications in Statistics Theory and Methods*, A9, 10251041.

Friendly, M. (1992), Graphical methods for categorical data. SAS User Group International Conference Proceedings, 17, 190200. <http://www.math.yorku.ca/SCS/sugi/sugi17-paper.html>

<https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/assocplot.html>

4.1.18. Estrellas

Paquete base de R

Si lo tiene

Función:

`stars()`

Argumentos son:

(`x`, `full`, `scale`, `radius`, `labels`, `locations`, `nrow`, `ncol`, `len`, `key.loc`, `key.labels`, `key.xpd`, `xlim`, `ylim`, `flip.labels`, `draw.segments`, `col.segments`, `col.stars`, `col.lines`, `axes`, `frame.plot`, `main`, `sub`, `xlab`, `ylab`, `cex`, `lwd`, `lty`, `xpd`, `mar`, `add`, `plot`, ...)

`x`: Matriz o data frame de los datos. Una o parcela segmento serán producidos para cada fila de `x`. Se permiten valores que faltan (NA), pero son tratados como si fueran 0 (después de la escala, si es relevante).

`full`: Flag lógica: si es TRUE, los diagramas del segmento ocuparán un círculo completo. De lo contrario, ocupan sólo el semicírculo (superior).

`scale`: Flag lógica: si es TRUE, las columnas de la matriz de datos se escalan de forma independiente de manera que el valor máximo en cada columna es 1 y el mínimo es 0. Si es FALSE, la presunción es que los datos se han reducido por algún otro algoritmo para la gama [0, 1].

`radius`: Flag lógica: en TRUE, se elaborará el radio correspondiente a cada variable en los datos.

`labels`: Vector de cadenas de caracteres para etiquetar las diagramas. A diferencia de las estrellas de la función `S`, no se hace ningún intento de construir etiquetas si las etiquetas = NULL.

`locations`: De cualquier matriz de dos columnas con la coordenadas X e Y se utilizan para colocar cada una de las diagramas del segmento; o numérico de longitud 2 cuando todas las parcelas deben ser superpuestas (para una 'trama de araña '). De forma predeterminada, las ubicaciones = NULL, las diagramas del segmento serán colocados en una cuadrícula rectangular.

`nrow`, `ncol`: Enteros que proporcionan el número de filas y columnas para usar cuando las ubicaciones es NULL. De forma predeterminada, `nrow == ncol`, se utilizará un diseño cuadrado.

`len`: Factor de escala para la longitud del radio o segmentos.

`key.loc`: Vector con las coordenadas X e Y de la unidad clave.

key.labels: Vector de cadenas de caracteres para el etiquetado de los segmentos de la unidad clave. Si se omite, se utiliza el segundo componente de dimnames (x), si está disponible.

key.xpd: Interruptor de recorte para la unidad clave (dibujo y etiquetado), véase el par ('XPD').

xlim: Vector con el rango de coordenadas de X para mostrar.

ylim: Vector con el rango de coordenadas de Y para mostrar.

flip.labels: Lógica que indica si las ubicaciones de las etiquetas deben voltear hacia arriba y hacia abajo a partir del diagrama de diagrama. El valor predeterminado es una heurística un poco inteligente.

draw.segments: Lógica. Si es TRUE dibuja un segmento.

col.segments: Vector de color (número entero o carácter, ver par), especificando cada uno un color para uno de los segmentos (variables). Se ignora si draw.segments = FALSE.

col.stars: Vector de color (número entero o carácter, ver par), especificando cada uno un color para una de las estrellas (casos). Se ignora si draw.segments = TRUE.

col.lines: Vector de color (número entero o carácter, ver par), especificando cada uno un color para una de las líneas (casos). Se ignora si draw.segments = TRUE.

axes: Flag lógica: si es TRUE los ejes son añadidos al diagrama.

frame.plot: Flag lógica: Si es TRUE, la zona del diagrama se enmarca.

main: Título principal del diagrama.

sub: Subtítulo del diagrama.

xlab: Una etiqueta para el eje X.

ylab: Una etiqueta para el eje Y.

cex: Factor de expansión de caracteres para las etiquetas.

lwd: Ancho de línea utilizado para el dibujo.

lty: Tipo de línea utilizada por el dibujo.

xpd: Lógica o NA que indica si se debe hacer recortes, ver par (XPD =.).

mar: Argumento a la altura (Mar = *), por lo general la elección de los márgenes más pequeños que de forma predeterminada.

...: Más argumentos, se pasan a la primera llamada de plot(), ver plot.default y box() si frame.plot es TRUE.

add: Lógica, si es TRUE añade estrellas al diagrama actual.

plot: Lógica, si es FALSE, nada será mostrado.

Dónde encontrarlo en R:

Paquete 'graphics'.

Paquetes requeridos:

'graphics'.

Referencias:

<https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/stars.html>

<https://cran.r-project.org/doc/contrib/grafi3.pdf>

4.1.19. Gráfico cuádruple/fourfold**Paquete base de R**

Si lo tiene

Función:

fourfoldplot()

Argumentos son:

(x, color, conf.level, std, margin, space, main, mfrow, mfcoll)

x: Una tabla de contingencia 2 por 2 por k en forma de matriz, o como una matriz de 2 por 2 si k es 1.

color: Un vector de longitud 2 especificando los colores a utilizar para las diagonales más pequeños y más grandes de cada 2 por 2 tabla.

conf.level: El nivel de confianza utilizado en los anillos de confianza del índice de probabilidades. Debe ser un solo número no negativo menor que 1; si está en 0, los anillos de confianza se suprimen.

`std`: Una cadena de caracteres, que especifica cómo normalizar la tabla. Debe coincidir con uno de los `'margins'`, `'ind.max'`, o `'all.max'`, y puede ser abreviado a la letra inicial. Si se establece en `'margins'`, cada tabla de 2 por 2 ha sido estandarizada para equiparar los márgenes especificados por el margen, preservando al mismo tiempo el índice de probabilidades. Si `'ind.max'` o `'all.max'`, las tablas son o bien individualmente o simultáneamente estandarizada a una frecuencia máxima de la célula 1.

`margin`: Un vector numérico con los márgenes de equiparar. Debe ser uno de 1, 2 ó `c(1, 2)` (por defecto), que corresponde a la normalización de la fila, columna o ambos márgenes en cada tabla 2 por 2. Sólo se utiliza si `std` es igual a `'margins'`.

`space`: La cantidad de espacio (como una fracción del radio máximo de los cuartos de círculo) que se utiliza para las etiquetas de fila y columna.

`main`: Cadena de caracteres para el título cuatro veces.

`mfrow`: Un vector numérico de la forma `c(nr,nc)`, que indica que las pantallas de los 2 por 2 tablas, deben estar dispuestos en un diseño `nr` por `nc`, relleno por filas.

`mfcol`: Un vector numérico de la forma `c(nr,nc)`, lo que indica que las pantallas de los 2 por 2 tablas deben estar dispuestos en un diseño `nr` por `nc`, llenado por columnas.

Dónde encontrarlo en R:

Paquete `'graphics'`.

Paquetes requeridos:

`'graphics'`.

Referencias:

<https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/fourfoldplot.html>

4.1.20. Diagrama de mosaico

Paquete base de R

Si lo tiene

Función:

`mosaicplot()`

Argumentos son:

(x, main, sub, xlab, ylab, sort, off, dir, color, shade, margin, cex.axis, las, border, type, ..., formula, subset, na.action)

x: Una tabla de contingencia en forma de matriz, con las etiquetas de categoría opcionales especificadas en el atributo `dimnames(x)`. La tabla está mejor creada por el comando de la `table()`.

main: Cadena de caracteres para el título del mosaico.

sub: Cadena de caracteres para el sub-título del mosaico (En la parte inferior).

xlab, ylab: Etiquetas de X e Y utilizadas para el gráfico; por defecto, el primero y el segundo elemento del nombre (`dimnames (X)`) (es decir, el nombre de la primera y la segunda variable en X).

sort: Vector de ordenamiento de las variables, que contiene una permutación de los números enteros 1: `length(dim (x))` (por defecto).

off: Vector de desplazamientos para calcular el espacio existente de porcentaje en cada nivel del mosaico (valores apropiados son entre 0 y 20), y el valor predeterminado es 20 veces el número de divisiones para las tablas de 2 dimensiones, y 10 lo contrario. Re-escalado a un máximo de 50, y se recicla si necesario.

dir: Vector de direcciones de división ('v' para vertical y 'h' para horizontal) para cada nivel del mosaico, una dirección para cada dimensión de la tabla de contingencia. El valor predeterminado consiste en alternar direcciones, comenzando con una división vertical.

color: Vector lógico (reciclaje) de los colores para el sombreado de color, usado sólo cuando `shade` es FALSE, o NULL (predeterminado). Por defecto, las cajas grises se dibujan. `color = TRUE` utiliza una paleta de color gris con corrección gamma. `color = FALSE` da cajas vacías sin sombreado.

shade: Una lógica que indica si produce parcelas de mosaico extendidos, o un vector numérico de como máximo 5 números positivos distintos de los valores absolutos de los puntos de corte para los residuos. Por defecto, `shade` es FALSE y mosaicos simples son creados. El uso de `shade = TRUE` corta valores absolutos en 2 y 4.

margin: Una lista de vectores con los totales marginales para estar en forma en el modelo log-lineal. Por defecto, un modelo de independencia está equipado. Ver `loglin` para obtener más información.

cex.axis: La ampliación que se utilizará para la anotación eje, como un múltiplo de par ('cex').

las: numérico; el estilo de la etiquetas de los ejes, ver par.

border: Color de los bordes de celdas: véase el polygon.

type: Una cadena de caracteres que indica el tipo de residuo a ser representado. Debe ser uno de 'Pearson' (dando a los componentes de chi-cuadrado de Pearson), 'deviance' (dando a los componentes de la relación de probabilidad chi-cuadrado), o 'FT' corresponde a los residuos de Freeman-Tukey. El valor de este argumento puede ser abreviado.

formula: Una fórmula, tales como $Y \sim x$.

data: Una trama de datos (o lista), o una tabla de contingencia de la que se deben tomar las variables en la fórmula.

...: Más argumentos que se pasan hacia o desde métodos.

subset: Un vector opcional, que especifica un subconjunto de observaciones en la trama de datos que se utiliza para el trazado.

na.action: Una función que indica lo que debe suceder cuando los datos contienen variables para ser cruzada tabulados, y estas variables contienen NAs(valores ausentes). El valor por defecto es omitir los casos que tienen una NA en alguna variable. Desde la tabulación omitirá todos los casos que contienen valores que faltan, esto sólo será útil si la función na.action reemplaza los valores que faltan.

Dónde encontrarlo en R:

Paquete Graphics.

Paquetes requeridos:

'Graphics'.

Referencias:

<https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/mosaicplot.html>

4.1.21. Diagrama de secuencia en cascada

Paquete base de R

No lo tiene.

Función:

waterfallchart()

Argumentos son:

(x, data, groups, horizontal, panel, prepanel, summaryname, box.ratio, origin, level.lines, ...)

x: Una fórmula que describe la forma de condicionamiento del diagrama. La fórmula general es de la forma $y \sim x \sim g1 * g2 * \dots$, lo que indica que las parcelas de y (en el eje y) frente a 'x' (en el eje x) se deben producir condicionada a las variables 'G1, G2, ...'. Sin embargo, las variables condicionantes 'G1, G2, ...' se puede omitir.

data: Una trama de datos que contiene los valores (o más precisamente, todo lo que es un argumento válido 'envir' en 'eval', por ejemplo, una lista o un entorno) de las variables de la fórmula, así como 'groups' y 'subset' si fuera aplicable. Si no se encuentra en 'data', o si 'data' no se especifica, las variables se buscan en el entorno de la fórmula.

groups: Un vector esperado para actuar como una variable de agrupación dentro de cada panel, normalmente se utiliza para distinguir los diferentes grupos mediante la variación de los parámetros de gráficos como el color y el tipo de línea. A diferencia de la función de diagrama de barras, 'groups' especifica dónde, deben aparecer subtotales columnas. Hay un subtotal creado para cada grupo especificado. Si no se dan grupos, una columna resumen será reportada.

horizontal: Este argumento se utiliza para procesar los argumentos de estas funciones de alto nivel, pero lo más importante, se pasa como argumento a la función del panel.

panel: Esto atrae la trama actual después de que el diagrama de caja haya hecho el trabajo difícil de procesar la fórmula.

prepanel: Esta función devuelve la información bwplot del número de columnas para mostrar y dónde colocar etiquetas.

summaryname: Nombre de la columna de resumen, por lo general 'total'.

box.ratio: Especifica la relación de la anchura de los rectángulos para el espacio inter rectángulo.

origin: Inicial de desplazamiento con respecto al eje X. El valor sirve como punto de partida lógico para la primera columna y la columna resumen. El valor predeterminado es 0.

level.lines: Si es FALSE, las líneas que conectan las cajas adyacentes se omiten de la pantalla.

... : más argumentos.

Dónde encontrarlo en R:

Paquete 'waterfall'.

Paquetes requeridos:

'waterfall'.

Referencias:

<http://www.inside-r.org/packages/cran/waterfall/docs/waterfallchart>

Andrew Jaquith, Security Metrics: Replacing Fear, Uncertainty, and Doubt (Boston: Addison-Wesley Professional, 2007), 170-172.

Ethan M. Rasiel, The McKinsey Way: Using the Techniques of the World's Top Strategic Consultants to Help You and Your Business (New York: McGraw-Hill, 1999), 113-118.

4.1.22. Gráfica de Venn**Paquete base de R**

No lo tiene.

Función:

Dependiendo del número de conjuntos que queramos realizar usamos uno y otro. Dependerá de cual elijamos el número de áreas que incorporaremos en sus argumentos. Además las longitudes de los vectores dependerá del número de conjuntos que utilizemos.

```
draw.single.venn ()
```

```
draw.pairwise.venn()
```

```
draw.triple.venn()
```

```
draw.quad.venn()
```

Argumentos son:

(area1, area2 , cross.area, category, euler.d, scaled , inverted, ext.text, ext.percent, lwd , lty , col, fill, alpha, label.col, cex), fontface , fontfamily, cat.pos, cat.dist, cat.cex, cat.col, cat.fontface, cat.fontfamily, cat.just, cat.default.pos, cat.prompts, ext.pos, ext.dist, ext.line.lty, ext.length, ext.line.lwd1, rotation.degree, rotation.centre, ind, sep.dist, offset, cex.prop, print.mode , sigdigs, ...)

area1: El tamaño del primer conjunto.

area2: El tamaño del segundo conjunto.

cross.area: El tamaño de la intersección entre los conjuntos.

category: Un vector (longitud 2) de cadenas dando el nombre de categoría de los conjuntos.

euler.d: Booleano, que indica si desea dibujar diagramas de Euler cuando se cumplen las condiciones o no (diagramas de Venn con círculos móviles).

scaled: Booleano, que indica si debe escalar tamaños circulares en el diagrama de acuerdo configurar tamaños o no (euler.d debe ser verdad para permitir esto).

inverted: Booleano, que indica si el diagrama debe ser reflejado a lo largo el eje vertical o no.

ext.text: Booleano, indica si desea colocar etiquetas en la zona fuera de los círculos en el caso de las zonas parciales pequeñas o no.

ext.percent: Un vector (longitud 3) indica la proporción de una zona parcial que tiene que ser menor para activar la colocación del texto externo. Los elementos permiten el control individual de las áreas en el orden de Area1, Area2 y el área se cruzan.

lwd: Un vector (longitud 2) de números que dan el ancho de línea de circunferencias de los círculos.

lty: Un vector (longitud 2) que da el patrón de línea de trazos de las circunferencias de los círculos

col: Un vector (longitud 2) que da los colores de las circunferencias de los círculos.

fill: Un vector (longitud 2) que da los colores de las áreas de los círculos.

alpha: Un vector (longitud 2), dando la transparencia alfa de las áreas de los círculos.

label.col: Un vector (longitud 3) que da los colores de las áreas de 'etiquetado'.

cex: Un vector (longitud 3) que da el tamaño de las áreas 'etiquetado'.

fontface: Un vector (longitud 3), dando la tipo de fuente de la áreas 'etiquetado'.

fontfamily: Un vector (longitud 3) dando a la familia de fuente de la áreas 'etiquetado'.

`cat.pos`: Un vector (longitud 2) que da las posiciones (en grados) de los nombres de las categorías a lo largo de los círculos, con 0 (por defecto) en la ubicación 12:00 según el reloj.

`cat.dist`: Un vector (longitud 2) indicando las distancias (en unidades npc) de los nombres de las categorías de los bordes de los círculos (puede ser negativo).

`cat.cex`: Un vector (longitud 2) que da el tamaño de los nombres de las categorías.

`cat.col`: Un vector (longitud 2) que da los colores de los nombres de las categorías.

`cat.fontface`: Un vector (longitud 2), dando la tipo de fuente de los nombres de las categorías.

`cat.fontfamily`: Un vector (longitud 2) dando la familia de fuente de los nombres de las categorías.

`cat.just`: Lista de 2 vectores de longitud 2 que indica la justificación horizontal y vertical de cada nombre de categoría.

`cat.default.pos`: Uno similar a `c('outer', 'text')` para especificar la ubicación predeterminada de los nombres de las categorías (`cat.pros` y `cat.dist` se manejan de forma diferente).

`cat.prompts`: Booleano que indica si se debe mostrar el texto de ayuda en nombre de la categoría de posicionamiento o no.

`ext.pos`: Un vector (longitud 1 o 2) que da las posiciones (en grados) de las etiquetas de área externa a lo largo de los círculos, con 0 (por defecto) a las 12 horas.

`ext.dist`: Un vector (longitud de 1 o 2), dando hasta dónde coloca el área externa de etiquetas con respecto a su punto de anclaje.

`ext.line.lty`: Un vector (longitud 1 o 2) que da el patrón de discontinuidad de las líneas que unen las etiquetas área externa de sus puntos de anclaje.

`ext.length`: Un vector (longitud de 1 o 2), dando la proporción de las líneas de conexión de las etiquetas área externa a sus puntos de anclaje actualmente dibujada.

`ext.line.lwd`: Un vector (longitud de 1 o 2), dando la anchura de las líneas que unen las etiquetas área externa a sus puntos de anclaje.

`rotation.degree`: Número de grados para girar todo el diagrama.

- `rotation.centre`: Un vector (longitud 2) que indica (x, y) del centro de rotación.
- `ind`: Booleano que indica si la función es dibujar automáticamente el diagrama antes de devolver el objeto de lista o no.
- `sep.dist`: Número que da la distancia entre los círculos en el caso de un diagrama de Euler que muestra conjuntos mutuamente excluyentes.
- `offset`: Número entre 0 y 1 que da la cantidad de desplazamiento desde el centro en caso de un diagrama que muestra los conjuntos de Euler incluido.
- `cex.prop`: Una función o cadena que se utiliza para cambiar la escala de las zonas.
- `print.mode`: Puede ser o bien 'raw' o 'percent'. Este es el formato que los números se imprimirán. Puede pasar en un vector con el segundo elemento que se está imprimiendo debajo del primero.
- `sigdigs`: Si uno de los elementos en `print.mode` es 'percent', entonces este es el número de dígitos significativos que deberá mantenerse.
- ...: Los argumentos adicionales que se deben pasar, incluyendo el margen, lo que indica la cantidad de espacio en blanco alrededor del diagrama final en unidades de NPC.

Dónde encontrarlo en R:

Paquete 'VennDiagram'.

Paquetes requeridos:

'VennDiagram'.

Referencias:

- <https://cran.r-project.org/web/packages/VennDiagram/VennDiagram.pdf>
- <http://finzi.psych.upenn.edu/library/VennDiagram/html/draw.pairwise.venn.html>
- <http://www.inside-r.org/node/161236>

4.1.23. Gráfica de nivel levelplot**Paquete base de R**

No lo tiene.

Función:

`levelplot()`

Argumentos son:

(x, data, allow.multiple, outer, aspect, panel, prepanel, scales, strip, groups = NULL, xlab, lim, ylab, ylim, at, cuts, pretty, region, drop.unused.levels, ..., useRaster, lattice.options, default.scales, default.prepanel, colorkey = region, col.regions, alpha.regions, subset)

x: Para el método de la fórmula, una fórmula de la forma $z \sim x * y \sim g1 * g2 * \dots$, donde Z es una respuesta numérica, y X e Y son los valores numéricos evaluados en una celda rectangular. g1, g2, ... son variables condicionales opcionales, y debe ser o bien factores o tejas si está presente. Los cálculos se basan en la suposición de que todos los valores X e Y son evaluadas en una cuadrícula (definidos por sus valores únicos). La función no devolverá un error si esto no es cierto, pero el diagrama puede que no sea significativo. Sin embargo, los valores de X e Y no necesitan ser equidistantes. Tanto levelplot y wireframe tienen métodos para objetos de matriz, array, y tabla, en cuyo caso X proporciona el vector Z descrito anteriormente, mientras que sus filas y columnas se interpretan como la X e Y respectivamente vectores. Esto es similar a la forma usada en filled.contour e image. Para los arrays y las tablas de dimensiones superiores, otras dimensiones se utilizan como variables condicionantes. Tenga en cuenta que los dimnames pueden duplicarse; esto es manejado por llamada make.unique para hacer los nombres únicos (aunque se utilizan las etiquetas originales para la X e Y ejes).

data: Para los métodos de fórmula, una trama de datos opcional en la que las variables en la fórmula (así como grupos y subgrupos, en su caso) han de ser evaluados. Por lo general es ignorado con una advertencia en otros casos.

row.values, column.values: Vectores opcionales de valores que definen la red cuando X es una matriz. row.values y column.values deben tener las mismas longitudes como nrow (x) y ncol (x), respectivamente. Por números predeterminados, fila y columna.

panel: Función de panel utilizado para crear el diagrama, como se describe en xyplot.

aspect: Para los métodos de matriz, la relación de aspecto por defecto se elige para cada cuadrado de celda. El valor por defecto habitual es de aspecto = 'fill', como se describe en xyplot.

at: Un vector numérico dando puntos de interrupción a lo largo del rango de Z. Las curvas de nivel (si hay) serán elegidos a esas alturas, y las regiones intermedias se colorean utilizando col.regions. En este último caso, los valores fuera del rango no serán dibujados en absoluto. Esto sirve como una manera para limitar el rango de los datos mostrados, similar a lo que un argumento zlim podría haber sido utilizado. Sin embargo, esto también significa que cuando se suministra en forma explícita, uno tiene que tener cuidado de incluir valores fuera del rango de Z para asegurarse de que se muestran todos los datos.

at: Puede tener longitud uno, sólo si region = FALSE.

`col.regions`: Vector de color para ser utilizado si las regiones son TRUE. La idea general es que este debe ser un vector de color de la longitud moderadamente grande (mayor que el número de regiones. Por defecto es 100). Se espera que este vector fuera variando gradualmente en el color (de modo que los colores cercanos serían similares). Cuando los colores son en realidad elegidos, se eligen para estar espaciados igualmente a lo largo de este vector. Cuando hay más regiones que en colores en `col.regions`, los colores se reciclan. La asignación de color real es realizada por `level.colors`, que se documenta por separado.

`alpha.regions`: Numérica, especificando la transparencia alfa (sólo funciona en algunos dispositivos)

`colorkey`: Lógico que especifica si un botón de color se va a dibujar junto con el diagrama, o una lista que describe la clave de color.

`contour`: Un indicador lógico, indicando si se dibujan las curvas de nivel.

`cuts`: El número de niveles de la gama de Z sería dividido dentro.

`labels`: Normalmente, una lógica que indica si las curvas de nivel deben ser etiquetados, pero existe otras posibilidades de control más sofisticados. Los detalles se documentan en la página de ayuda para `panel.levelplot`, a la que este argumento se pasa sin cambios. Esa página de ayuda también documenta el argumento `label.style`, que afecta a cómo se generan las etiquetas.

`pretty`: Un indicador lógico, que indica si se utiliza cut en ubicaciones y etiquetas.

`region`: Un indicador lógico, indicando si las regiones entre las curvas de nivel se deben llenar como en una parcela de nivel.

`allow.multiple`, `outer`, `prepanel`, `scales`, `strip`, `groups`, `xlab`, `xlim`, `ylab`, `ylim`, `drop.unused.levels`, `lattice.options`, `default.scales`, `subset`: Estos argumentos se describen en la página de ayuda para `ggplot`.

`default.prepanel`: Función del panel antes de repliegue. Ver `xyplot`.

...: Otros argumentos pueden ser suministrados. Algunos son procesados por el gráfico de contorno `levelplot` r, y los que son reconocidos puedan ser traspasados a la función del panel.

`useRaster`: Un indicador lógico que indica si las representaciones de mapa de bits deben ser usadas, tanto la imagen de color falso y la clave de color (si está presente). Efectivamente, configurando este valor a TRUE cambia la función del panel predeterminado de `panel.levelplot` a `panel.levelplot.raster`, y establece el valor predeterminado de la clave de `colorkey$raster` en TRUE. Tenga en cuenta que `panel.levelplot.raster` proporciona sólo un subconjunto de las características

de `panel.levelplot`, pero el establecimiento de `useRaster = TRUE` no comprobará si cualquiera de las características adicionales que se han solicitado. No todos los dispositivos son compatibles con imágenes de mapa de bits. Para los dispositivos que parecen carecer de apoyo, `useRaster = TRUE` se ignorará con una advertencia.

Dónde encontrarlo en R:

Paquete 'lattice'.

Paquetes requeridos:

'lattice', 'ggplot'.

Referencias:

<https://stat.ethz.ch/R-manual/R-devel/library/lattice/html/levelplot.html>

Sarkar, Deepayan (2008) Lattice: Multivariate Data Visualization with R, Springer. <http://lmdvr.r-forge.r-project.org/>

4.1.24. Pairs**Paquete base de R**

Si lo tiene.

Función:

`pairs()`

Argumentos son:

(`x`, `formula`, `data`, `subset`, `na.action`, `labels`, `panel`, ..., `horInd`, `verInd`, `lower.panel`, `upper.panel`, `diag.panel`, `text.panel`, `label.pos`, `line.main`, `cex.labels`, `font.labels`, `row1atop`, `gap`, `log`)

`x`: Las coordenadas de puntos dados como columnas numéricas de un marco de matriz o de datos. columnas lógicas y de factores se convierten en numérico de la misma manera que `data.matrix` hace.

`formula`: Una fórmula, como `x + y + z`. Cada término dará una variable independiente en la impresión de pares, por lo que debe ser vectores términos numéricos. (Una respuesta será interpretada como otra variable, pero no trata de forma especial, por lo que es confuso para usar una.)

`data`: Un `data.frame` (o lista) a partir del cual se deben tomar las variables en la fórmula.

`subset`: Un vector opcional que especifica un subconjunto de las observaciones que se utiliza para la impresión.

`na.action`: una función que indica lo que debe ocurrir cuando los datos contienen NAs (valores nulos o no válidos). El valor por defecto es pasar los valores que faltan a las funciones del panel, pero `na.action = na.omit` causará que los casos con valores perdidos en cualquiera de las variables sea omitida por completo.

`labels`: El nombre de las variables.

`panel`: `function(x, y, ...)` que se utiliza para representar los contenidos de cada panel de la pantalla.

`...`: Argumentos que se pasan hacia o desde métodos. También parámetros gráficos pueden darse como argumentos que pueden graficar como principal. El par (`'OMA'`) será un valor apropiado a menos que se especifique.

`horInd`, `verInd`: Los índices (numéricos) de las variables que se representan en los ejes horizontal y vertical, respectivamente.

`lower.panel`, `upper.panel`: Separa las funciones `panel` (o `NULL`) que se utilizarán por debajo y por encima de la diagonal respectivamente.

`diag.panel`: Opcional `function(x, ...)` aplicado en las diagonales.

`text.panel`: Opcional `function(x, y, labels, cex, font, ...)` aplicado en las diagonales.

`label.pos`: Y posición de las etiquetas en el panel de texto.

`line.main`: Si se especifica principal, `line.main` da el argumento de línea de `mtext()`, que dibuja el título. Es posible que desee especificar o cuando se cambia `line.main`.

`cex.labels`, `font.labels`: Parámetros de gráficos para el panel de texto.

`row1attop`: Lógica. En caso de que el diseño sea en forma de matriz con la fila 1 en la parte superior, o un gráfico similar con la fila 1 en la parte inferior.

`gap`: Distancia entre subgráficas, líneas de margen.

`log`: una cadena de caracteres que indica si son ejes logarítmicos a utilizar: ver `plot.default`. `log = 'xy'` especifica ejes logarítmicos para todas las variables.

Dónde encontrarlo en R:

Paquete `'graphics'`.

Paquetes requeridos:

`'graphics'`.

Referencias:

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language. Wadsworth & Brooks/Cole.

<https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/pairs.html>

4.1.25. Series de tiempo múltiples**Paquete base de R**

No lo tiene.

Función:

`ts.plot()`

Argumentos son:

`(..., gpars)`

...: Una o más series de tiempo univariante o multivariante.

`gpars`: Lista de parámetros gráficos con nombre para pasar a las funciones de impresión. Los que se utilizan comúnmente pueden ser suministrados directamente en

Dónde encontrarlo en R:

Paquete 'stats'.

Paquetes requeridos:

'stats', 'graphics'.

Referencias:

<https://stat.ethz.ch/R-manual/R-devel/library/stats/html/ts.plot.html>

4.1.26. Coordenadas paralelas**Paquete base de R**

No lo tiene.

Función:

`parcood()`

Argumentos son:

`(x, col, lty, var.label, ...)`

x: Una matriz o trama de datos que indica las columnas que se utilizarán. Se permiten valores ausentes.

col: Un vector de colores, reciclado como necesario para cada observación.

lty: Un vector de tipos de línea, reciclados como necesario para cada observación.

var.label: Si es TRUE, el eje de cada variable se marca con los valores máximos y mínimos.

...: Otros parámetros gráficos que se pasan a `matplot`.

Dónde encontrarlo en R:

Paquete 'MASS'.

Paquetes requeridos:

'MASS'.

Referencias:

Wegman, E. J. (1990) Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association* 85, 664-675.

Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S*. Fourth edition. Springer.

4.2. Mapas

4.2.1. `country_choropleth`

Paquete base de R

No lo tiene.

Función:

`country_choropleth()`

Argumentos son:

(df, title, legend, num_colors, zoom)

df: Un `data.frame` con una columna llamada 'region' y una columna llamada 'value'. Los elementos de la columna 'region' deben coincidir con el nombre de las columnas de `country.map`.

title: Especifica un título opcional para el diagrama.

legend: Un nombre opcional para la leyenda.

num.colors: El numero de colores que se usarán en el mapa. Un valor de 1 usará una misma escala de color, y un valor entre [2, 9] usara todos esos colores.

zoom: Un vector opcional con las regiones a las que hacer zoom. Los elementos de este vector deben ser exactamente los que aparezcan en la columna 'region' de country.regions.

Dónde encontrarlo en R:

Paquete choroplethr.

Paquetes requeridos:

'choroplethr'.

Referencias:

<https://cran.r-project.org/web/packages/choroplethr/choroplethr.pdf>

4.2.2. GoogleMaps

Paquete base de R

No lo tiene.

Función:

`gvisMap()`

Argumentos son:

(data, locationvar, tipvar, options, chartid)

data: Un objeto data.frame. Los datos deben tener al menos dos columnas con los nombres de las ubicaciones(locationvar) y las variables para representar el texto en el icono de ayuda (tipvar).

locationvar: Columna de nombres con las geolocalizaciones para ser analizadas. Las localizaciones pueden estar proporcionadas en dos formatos: Formato 1 latitud:longitud. Formato 2 La primera columna debería ser una cadena que contenga una dirección. Esta dirección debería ser tan completa como tu puedas hacerla.

tipvar: Nombre de la columna de datos con el texto de la cadena sobre el icono extremo.

options: Lista de opciones de configuración de Google Map.

Para más detalle: <https://developers.google.com/chart/interactive/docs/gallery/map>

Opciones de configuración Los parámetros se pueden ajustar a través de una listas indicadas. Los parámetros tienen que asignar aquellos de la documentación oficial de Google.

1. Argumentos booleanos, se establece en TRUE o FALSE, usando la sintaxis R.
2. Parámetros de la API de Google con un solo valor y con nombres que no incluyen un '.' se establece como uno lo haría en R, es decir `options = list(width = 200, height = 300)`. Las excepciones a esta regla son las opciones de anchura y altura para `gvisAnnotatedTimeLine` y `gvisAnnotationChart`. Para esas dos funciones, la anchura y la altura debe ser cadenas de caracteres del formato 'Xpx', donde X es un número, o 'automático'. Por ejemplo, las opciones de la lista = `(width = '200px', height = '300px')`.
3. Parámetros de la API de Google con nombres que no incluyen un '.', pero requieren múltiples valores se ajustan como un carácter, envuelto en '[]' y separados por comas, por ejemplo, `options = list(colors = '['#cbb69d', '#603913', '#c69c6e']')`
4. Parámetros de la API de Google con nombres que sí incluyen un '.' presentan parámetros con varias sub-opciones y tienen que ser establecido como un carácter envuelto en ' '. Los valores de estas sub-opciones se establecen a través del parámetro: `value`. Los valores booleanos tienen que ser declarado como 'TRUE' o 'FALSE'. Por ejemplo, la documentación Google establece las opciones de formateo para el eje vertical y establece el parámetro como `vAxis.format`. A continuación, este parametro se puede ajustar en R como: `options = lista (vAxis = '{formato: '#, # # # %' }')`.
5. Si varias sub-opciones se han establecido, por ejemplo, `titleTextStyle.color`, `titleTextStyle.fontName` y `titleTextStyle.fontSize`, entonces aquellas se pueden combinar en un solo elemento de la lista, tales como: `options = list(titleTextStyle = '{color:'red', fontName: 'Courier', fontSize: 16}')`
6. Parámetros que pueden tener más de un valor por cada sub-opcion están envueltos en '[]'. Por ejemplo, para configurar las etiquetas de uso eje de la izquierda y la derecha: `options = list(VAX = '[title: 'val1', title: 'val2']')`
7. `gvis.editor` una etiqueta de caracteres para un botón de encendido-página que se abre un cuadro de diálogo en la página permitiendo a los usuarios editar, cambiar y personalizar el gráfico. Por defecto se da ningún valor y por lo tanto no se muestra ningún botón.

`chartid`: Carácter. Si es ausente (por defecto) un identificador gráfico al azar será generado basado en el tipo de gráfico y el archivo temporal.

Dónde encontrarlo en R:

Paquete 'gvis'.

Paquetes requeridos:

'gvis'.

Referencias:

<https://cran.r-project.org/web/packages/googleVis/googleVis.pdf>

4.2.3. Mapa de contornos**Paquete base de R**

Si lo tiene.

Función:

`contour()`

Argumentos son:

(`x`, `y`, `z`, `nlevels`, `levels`, `labels`, `labcex`, `drawlabels`, `logical`, `method`, `vfont`, `xlim`, `ylim`, `zlim`, `axes`, `frame.plot`, `col`, `lty`, `lwd`, `add`, ...).

`x`, `y`: La ubicación de las líneas de cuadrícula en la que se miden los valores de `Z`. Estos deben estar en orden ascendente. Por defecto, se utilizan los valores equidistantes de 0 a 1. Si `X` es una lista, sus componentes `x$X` y `X$y` se utilizan para `x` e `y`, respectivamente. Si la lista tiene `z` componente Esto se utiliza para `z`.

`z`: Una matriz que contiene los valores a ser representados (NAs (valores ausentes o inválidos) están permitidos). Tenga en cuenta que `X` puede ser utilizado en lugar de `Z` para mayor comodidad.

`nlevels`: Número de niveles de contorno deseados, fuera de los niveles no se suministra.

`levels`: Vector numérico de niveles en los que dibujar los contornos.

`labels`: Un vector que da las etiquetas de los contornos. Si se utilizan `NULL` entonces los niveles son usados como etiquetas, por lo demás este es coaccionado por `as.character`.

`labcex`: Cex para el etiquetado de contorno. Se trata de un tamaño absoluto, no es un múltiplo de par ('CeX').

`drawlabels`: Valor lógico. Los contornos se etiquetan si es `TRUE`.

`method`: Cadena de caracteres que especifica dónde se ubicarán las etiquetas. Los valores posibles son 'simple', 'edge' y 'flattest' (por defecto).

`vfont`: Si es `NULL`, la familia de fuentes actual y la fachada se utilizan para las etiquetas de contorno. Si un vector de caracteres de longitud 2 entonces, las fuentes vectoriales Hershey se utilizan para las etiquetas de contorno. El primer elemento del vector selecciona un tipo de letra y el segundo elemento selecciona un `fontindex` (véase el texto para más información). El valor predeterminado es `NULL` en los dispositivos gráficos con la rotación de alta calidad del texto y `c` (`'sans serif'`, `'plain'`) en caso contrario.

`xlim, ylim, zlim`: `x, y, z` límites de la parcela.

`axes, frame.plot`: Lógica que indica si los ejes o una caja deben ser dibujados, ver `plot.default`.

`col`: El color de las líneas dibujadas.

`lty`: tipo de línea para las líneas dibujadas.

`lwd`: Ancho de línea de las líneas dibujadas.

`add`: Lógico. Si es `TRUE`, se suman a una trama actual.

`...` : Los argumentos adicionales a `plot.window`, título, eje y la caja, por lo general los parámetros gráficos tales como `cex.axis`.

Dónde encontrarlo en R:

Paquete `'graphics'`.

Paquetes requeridos:

`'graphics'`.

Referencias:

<https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/contour.html>

4.2.4. Mapa topográfico

Paquete base de R

si lo tiene

Función:

`persp()`

Argumentos son:

(`x, y, z, xlim, ylim, range, zlim, xlab, ylab, zlab, main, sub, theta, phi, r, d, scale, expand, col, border, ltheta, lphi, shade, box, axes, nticks, ticktype, ...`)

x, y: Ubicación de las líneas de la rejilla sobre la cual los valores en Z son medidos. Estas deben estar en orden ascendente. Por defecto, se usan valores igualmente espaciados en el intervalo de 0 a 1. Si X es una lista, sus componentes `xy` son usados para x e y respectivamente.

z: Una matriz que contiene los valores a ser representados (valores NA son permitidos). Notar que X puede ser usada en vez de Z por conveniencia.

xlim, ylim, zlim: Límites x, y y z para el gráfico. El gráfico es producido de manera que el volumen rectangular definido por estos límites sea visible.

xlab, ylab, zlab: Títulos para los ejes. Estos deben ser cadenas de caracteres; no se permiten expresiones.

main, sub: Título y subtítulo.

theta, phi: Ángulos que definen la dirección de las vistas theta da la dirección azimutal y phi la colatitud.

r: La distancia del punto de mira desde el centro de la caja de graficación.

d: Un valor que puede usarse para variar la longitud de la transformación de la perspectiva. Valores mayores que 1 reducirán el efecto perspectiva y los valores menores o iguales a 1 lo exagerarán.

scale: Antes de visualizar las coordenadas x, y y z de los puntos que definen la superficie éstas son transformadas al intervalo [0, 1]. Si scale es TRUE las coordenadas son transformadas separadamente. Si es FALSE las coordenadas son escaladas de manera que el aspecto de las razones son mantenidas. Esto es útil para representar cosas como información DEM.

expand: Un factor de expansión aplicado a las coordenadas z. A menudo es usado con expand entre (0,1) para contraer el gráfico en la dirección z.

col: El color de las facetas de la superficie.

border: El color de la línea dibujada alrededor de las facetas de la superficie. Un valor NA inhabilitará el trazo de bordes. Esto resulta útil cuando la superficie es sombreada.

ltheta, lphi: Si se especifican valores finitos para estos, la superficie es sombreada como si fuera iluminada desde la dirección especificada por el azimut ltheta y l colatitud lphi.

shade: La sombra en un faceta de la superficie es calculada como $((1 + d)/2)$ shade, donde d es el producto punto de un vector unitario normal a la faceta y un vector unitario en la dirección de una fuente de luz. Los valores de shade cercanos a 1 producen sombreado similar a un modelo de fuente de luz puntual y valores cercanos a 0 no producen sombra. Los valores en el rango de 0.5 a 0.75 proporcionan una aproximación a la iluminación diurna.

box: Traza un caja para la superficie. Por defecto es 'TRUE'.

axes: Agrega ticks y etiquetas a la caja. Por defecto es 'TRUE'. Si 'box' es 'FALSE' las etiquetas y los ticks no son dibujados.

ticktype: Carácter: simple dibuja un flecha paralela al eje para indicar la dirección de aumento; detailed dibuja los ticks normales como en gráficos 2D.

nticks: El número aproximado de marcas ticks a dibujar en los ejes. No tiene efecto si 'ticktype' es simple.

...: Parámetros gráficos adicionales.

Dónde encontrarlo en R:

Paquete 'graphics'.

Paquetes requeridos:

'graphics'.

Referencias:

<https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/persp.html>

4.2.5. Mapa 3d rejilla

Paquete base de R

No lo tiene.

Función:

wireframe()

Argumentos son:

(x, data, panel,default.prepanel, ...)

`x`: El objeto en el que se lleva a cabo el envío del método. Para los métodos de 'fórmula', una fórmula de la forma $z \sim x * y \sim g1 * g2 * \dots$, donde z es una respuesta numérica, y x , y son los valores numéricos. $G1$, $G2$, ..., si están presentes, son variables condicionantes utilizado para el acondicionamiento, y debe ser tanto factores o tejas. En el caso de `wireframe`, los cálculos se basan en la suposición de que los valores de X y se evalúan en una cuadrícula rectangular definida por sus valores únicos. Los puntos de la rejilla no necesitan ser equidistantes. Para `wireframe`, x , y y z puede ser también matrices (de la misma dimensión), en cuyo caso se toman para representar una superficie 3-D parametrizada en una cuadrícula de 2-D (Por ejemplo, una esfera). Condicionante no es posible con esta característica. Los valores ausentes son permitidos, ya sea como valores de 'NA' en el vector Z , o falta filas de la trama de datos (sin embargo, que en ese caso la rejilla de X e Y serán determinadas solamente por los valores disponibles). Para una representación agrupada (produciendo múltiples superficies), las filas que faltan no están permitidos. `Wireframe` tiene métodos para objetos de matriz, en cuyo caso X proporciona el vector z que se ha descrito anteriormente, mientras que sus filas y columnas se interpretan como la X y vectores Y respectivamente. Esto es similar a la forma usada en `persp`.

`data`: Para los métodos 'fórmula', una trama de datos opcional en la que las variables en la fórmula (así como grupos y subgrupos, en su caso) han de ser evaluados. Los datos no deben ser especificados, excepto cuando se utiliza el método de la 'fórmula'.

`panel`: función de panel utilizada para crear la pantalla. Ver `panel.cloud` para más detalles (no triviales).

`default.prepanel`: función del panel antes de repliegue. Ver `xyplot`.

...: Cualquier número de otros argumentos puede ser especificado, y son pasados a la función del panel. En particular, los argumentos `distancia`, `perspectiva`, `pantalla` y `R.mat` son muy importantes en la determinación de la pantalla 3-D. El sombreado del argumento puede ser útil para las llamadas de la estructura `wireframe`, y controla el sombreado de la superficie representada. Estos argumentos se describen en detalle en la página de ayuda para `panel.cloud`. Además, un argumento denominado `zoom` puede ser especificado, el cual debe ser un escalar numérico que debe interpretarse como un factor de escala por el que se amplía la proyección. Esto puede ser útil para obtención de los nombres de las variables en la trama. Este argumento es en realidad utilizado por la función `prepanel` predeterminado.

Dónde encontrarlo en R:

Paquete 'lattice'.

Paquetes requeridos:

'lattice'.

Referencias:

<https://cran.r-project.org/web/packages/lattice/lattice.pdf>

4.2.6. Correlograma**Paquete base de R**

No lo tiene.

Función:

`corrgram()`

Argumentos son:

(x, type, order, labels, panel, lower.panel, upper.panel, diag.panel, text.panel, label.pos, label.srt, cex.labels, font.labels, rowlattice, dir, gap, abs, col.regions, cor.method, ...)

x: Un data frame con una observación por línea, o una matriz de correlación.

type: Se usa data o cor/corr para especificar explícitamente que 'X' son datos o una matriz de correlación. Raramente usado.

order: Deberían las variables ser reordenadas? Usar TRUE/'PCA' para una ordenación basada en PCA.

labels: Etiquetas a usar (En vez de los nombres de las variables del data-frame) para los paneles diagonales.

panel: Función usada para representar los contenidos de cada panel.

lower.panel, upper.panel: Separa las funciones del panel usadas por encima y por debajo de la diagonal.

diag.panel, text.panel: Función del panel usada en la diagonal.

label.pos: Posicionamiento horizontal y vertical de las etiquetas en los paneles diagonales.

label.srt: Rotación de la palabra para etiquetas diagonales.

cex.labels, font.labels: Parámetros gráficos para paneles diagonales.

gap: Distancia entre paneles.

abs: Usar el valor absoluto de las correlaciones para clusterizar? Por defecto FALSE.

col.regions: Una función devolviendo un vector de colores.

cor.method: Método de correlación para usar en las funciones de los paneles. Por defecto 'Pearson'.

... Parámetros adicionales.

Dónde encontrarlo en R:

Paquete 'corrgram'.

Paquetes requeridos:

'corrgram'.

Referencias:

<http://www.statmethods.net/advgraphs/correlograms.html>

<https://cran.r-project.org/web/packages/corrgram/corrgram.pdf>

4.2.7. Mapa de calor

Paquete base de R

Si lo tiene.

Función:

heatmap (x, Rowv, Colv, distfun, hclustfun, reorderfun, add.expr, symm , revC, scale, na.rm, margins , ColSideColors, RowSideColors, cexRow, cexCol, labRow, labCol, main, xlab, ylab, keep.dendro, verbose, ...)

Argumentos son:

(x, y ,)

x: Matriz numérica con los valores a ser representados.

Rowv: Determina si y como la fila del dendrograma debe ser calculada y re-ordenada. Debe ser o un dendrograma o un vector de valores ordenados que se usaran para re-ordenar el dendrograma o se usa 'N/A' para eliminar cualquier fila dendrograma.

Colv: Determina si y como la columna del dendrograma debe ser re-ordenada. Tiene las mismas opciones que el argumento Rowv. Colv=Rowv indicará que las columnas deberán ser tratadas igual que las filas.

distfun: Función usada para calcular la distancia entre columnas y filas.

`hclustfun`: Función usada para calcular el clustering hereditario cuando `Rowv` o `Colv` no son dendrogramas. Por defecto `hclust`. Debe introducirse como argumento el resultado de la función `distfun` y devolver un objeto al cual `as.dendrogram` podrá ser aplicado.

`reorderfun`: Función(`d`, `w`) de dendrograma y pesos para re-ordenar las filas y las columnas. Por defecto usa `reorder.dendrogram`.

`add.expr`: Expresión que será evaluada después de la llamada a la imagen. Puede ser usada para añadir componentes a la gráfica.

`symm`: Valor lógico, indicando si debe ser tratada simétricamente. Solo puede ser `True` si `X` es una raíz cuadrada.

`revC`: Valor lógico, indicando si el orden de la columna debe ser tratado al revés al dibujarse.

`scale`: Carácter indicando si los valores deben ser centrados y escalados en la dirección o bien de la fila o bien de la columnas, o ninguna.

`na.rm`: Valor lógico, indicando si los NAs(valores nulos o inválidos) deben ser quitados.

`margins`: Vector numérico de longitud 2 conteniendo los márgenes (ver `par(mar = *)`) de las etiquetas de columnas y filas, respectivamente.

`ColSideColors`: (opcional), vector de caracteres de longitud `ncol(x)` conteniendo los nombres de los colores para la barra horizontal del lado que sera usada para las columnas de `X`.

`RowSideColors`: (opcional), vector de caracteres de longitud `ncol(x)` conteniendo los nombres de los colores para la barra vertical del lado que sera usada para las filas de `X`.

`labRow`, `labCol`: Vector de caracteres con los nombres que deben usarse para las etiquetas de filas y columnas.

`main`, `xlab`, `ylab`: Títulos de `main`, `x-` y `y-axis`; por defecto ninguno.

`...`: parámetros adicionales para dibujar en la imagen, por ejemplo, columnas especificando los colores.

Dónde encontrarlo en R:

Paquete `'stats'`.

Paquetes requeridos:

`'stats'`.

Referencias:

<https://stat.ethz.ch/R-manual/R-devel/library/stats/html/heatmap.html>

4.2.8. Gráfico de Coordenadas Polares**Paquete base de R**

No lo tiene.

Función:

`polar.plot()`

Argumentos son:

`(lengths, polar.pos, labels, label.pos, start, clockwise, rp.type, ...)`

`lengths`: Vector de datos numérico. Las magnitudes se representan como las posiciones radiales de símbolos, líneas de extremos o vértices del polígono.

`polar.pos`: Vector numérico de posiciones en un círculo 0:360 grados. Estos se pueden convertir en radianes cuando se pasa a `'radial.plot'`.

`labels`: Etiquetas de texto para colocar en la periferia del círculo. Por defecto, etiquetas cada 20 grados. Para omitir las etiquetas, pasar una cadena vacía.

`label.pos`: Posiciones de las etiquetas periféricas al comienzo de los grados. La posición de cero grados en la trama en grados.

`clockwise`: Ya sea para aumentar los ángulos de las agujas del reloj en lugar de la opción predeterminada en sentido contrario a las agujas del reloj.

`rp.type`: Ya sea para trazar líneas radiales, símbolos o un polígono.

`...`: argumentos adicionales pasados a `'radial.plot'` y luego a `'trama'`.

Dónde encontrarlo en R:

Paquete `'plotrix'`.

Paquetes requeridos:

`'plotrix'`.

Referencias:

<https://cran.r-project.org/doc/contrib/grafi3.pdf>

4.2.9. Mapa geográfico leaflet

Paquete base de R

No lo tiene.

Función:

`leaflet()`

Argumentos son:

`(data, width, height, padding)`

`data`: Un objeto (hasta ahora los soportados son matrices, data frames, objetos espaciales del paquete `sp` de la clase `SpatialPoints`, `SpatialPointsDataFrame`, `Polygon`, `Polygons`, `SpatialPolygons`, `SpatialPolygonsDataFrame`, `Line`, `Lines`, `SpatialLines`, y `SpatialLinesDataFrame`).

`width`: La anchura del mapa.

`height`: La altura del mapa.

`padding`: El espaciado del mapa.

Dónde encontrarlo en R:

Paquete 'leaflet'.

Paquetes requeridos:

'leaflet'.

Referencias:

<https://cran.r-project.org/web/packages/leaflet/leaflet.pdf>

4.2.10. Proyecciones Cartográficas

Paquete base de R

No lo tiene.

Función:

`coord_map()`

Argumentos son:

`(projection , ..., orientation, xlim, ylim)`

`projection`: Proyección para usar, ver `mapproject` para la lista

...: Otros argumentos que se pasan a la orientación mapproject proyección de orientación, que por defecto es C(90, 0, significa (range (x))). Esto no es óptimo para muchas proyecciones, por lo que tendrá que proporcionar la suya propia. Ver mapproject para más información.

xlim: Especificar manualmente los límites de x (en grados de longitud).

ylim: Especificar manualmente los límites de Y (en grados de latitud).

Dónde encontrarlo en R:

Paquete 'ggplot2'.

Paquetes requeridos:

'ggplot2'.

Referencias:

<https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf>

Capítulo 5

Conclusiones y trabajos futuros

En el presente trabajo hemos aprendido sobre los diferentes entornos existentes para la aplicación de modelos estadísticos y la realización de análisis, además se han analizado las diferentes visualizaciones existentes en el entorno de programación R. Por ello a continuación presentamos las conclusiones obtenidas y posibles trabajos futuros a realizar.

Como conclusiones generales podemos afirmar que nos encontramos ante la era de los datos en la que almacenamos cada vez más datos. Por lo que las corporaciones necesitan realizar la importante tarea de extraer esos datos y aplicar modelos estadísticos, y por ello, no es menos importante el paso final donde se encuentra nuestro objeto estudiado, la visualización de los datos.

Para resolver esta problemática las corporaciones necesitan el uso de herramientas para el tratamiento de los datos y posterior visualización. Estas herramientas les permite analizar grandes volúmenes de datos de una manera más rápida y flexible, permitiendo a los expertos poder realizar diferentes análisis y comparación de datos así como aplicar diferentes modelos estadísticos.

Todo esto nos permite reducir tiempos y costos en la fase de interpretación de los datos pudiendo anticipar o predecir a través de los mismos. La representación de los datos nos facilita la comprensión de la información, y para ello es imprescindible una representación de esa información mediante la utilización de sofisticadas técnicas de visualización.

Este potencial se encuentra en pleno auge y las organizaciones se han dado cuenta más que nunca de la importancia de la correcta presentación de los datos, aportándoles una importante ventaja competitiva. Por ello el lenguaje R y su herramienta cuenta una numerosa comunidad de colaboradores la cual sigue creciendo permitiendo el desarrollo continuo de nuevas funcionalidades y mejoras en los algoritmos.

Como posible trabajo futuro a realizar, dada la numerosa comunidad y el continuo desarrollo permite que sigan apareciendo nuevos paquetes, funciones y nuevos tipos de visualizaciones, por lo que es posible la ampliación de este documento mediante la adhesión de todas esas nuevas formas de visualización.

Otro posible trabajo también podría ser la generación de un nuevo paquete en el cual concentre todas las funciones que aparezcan para la visualización de datos. De esta manera facilitará al desarrollador el evitar que tenga que descargarse numerosos paquetes para realizar las distintas visualizaciones.

Capítulo 6

Glosario

API: La interfaz de programación de aplicaciones, (del inglés: Application Programming Interface), es el conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

ASCII: (acrónimo inglés de American Standard Code for Information Interchange. Código Estándar Estadounidense para el Intercambio de Información), es un código de caracteres basado en el alfabeto latino, tal como se usa en inglés moderno.

Big Data: Es un concepto que hace referencia al almacenamiento de grandes cantidades de datos y a los procedimientos usados para encontrar patrones repetitivos dentro de esos datos. El fenómeno del Big Data también es llamado datos a gran escala.

Clase: Es una plantilla para la creación de objetos de datos según un modelo predefinido. Las clases se utilizan para representar entidades o conceptos, como los sustantivos en el lenguaje.

CRAN: CRAN (Comprehensive R Archive Network) R-Wiki Interfaz Web para R, R Graph Gallery, una colección de gráficos creados con R.

Dicotómica: se denomina variable dicotómica o binaria a aquella que tiene solo dos formas de presentarse es decir, que puede asumir solo dos valores posibles); por ejemplo la variable sexo con sus dos formas mujer y hombre.

GNU: Es un sistema operativo de tipo Unix desarrollado por y para el Proyecto GNU y auspiciado por la Free Software Foundation. Está formado en su totalidad por software libre, mayoritariamente bajo términos de copyleft. GNU es un acrónimo recursivo de 'GNU's Not Unix' (en español: GNU no es Unix)

Lenguaje R: Es un entorno y lenguaje de programación con un enfoque al análisis estadístico.

Linux: es uno de los términos empleados para referirse a la combinación del núcleo o kernel libre similar a Unix denominado Linux con el sistema operativo GNU.

Macintosh: Abreviado como Mac, es la línea de computadoras personales diseñada, desarrollada y comercializada por Apple Inc.

Matlab: (abreviatura de MATrix LABoratory, 'laboratorio de matrices') es una herramienta de software matemático que ofrece un entorno de desarrollo integrado.

NA: Valor ausente, proviene de Not Available.

NULL: El término null o nulo o DG es a menudo utilizado en la computación, haciendo referencia a la nada.

SPSS: Es un programa estadístico informático muy usado en las ciencias exactas, sociales y aplicadas, además de las empresas de investigación de mercado. Originalmente SPSS fue creado como el acrónimo de Statistical Package for the Social Sciences.

SAS: Es un lenguaje de programación desarrollado por SAS Institute a finales de los años sesenta.

Outlier: un valor atípico, una observación que es numéricamente distante del resto de los datos.

Paquete: Un paquete es un espacio de nombre que organiza un conjunto de clases.

Razonamiento diagramático: Es el que se lleva adelante haciendo uso de representaciones visuales de los conceptos. En esta técnica, los diagramas y los gráficos son más importantes que las palabras y las expresiones matemáticas.

Stata: Es un paquete de software estadístico creado en 1985 por StataCorp. Es utilizado principalmente por instituciones académicas y empresariales dedicadas a la investigación, especialmente en economía, sociología, ciencias políticas, biomedicina y epidemiología.

S.O.: Sistema operativo.

Bibliografía

- [1] [HTTP://GUIDES.NYU.EDU/QUANT/STATSOFT](http://guides.nyu.edu/quant/statsoft)
- [2] REAL ACADEMIA ESPAÑOLA *diagrama. Diccionario de la lengua española (23. edición)*. Madrid: Espasa.(2014)
- [3] HILDEBRAND, K; OTT, L. *Estadística Aplicada a la Administración y a la Economía* Wilmington, Addison- Wesley Iberoamericana. (1988)
- [4] PARETO, V. I. L. F. R. E. D. O. *Diagrama de pareto*.(1848)
- [5] SILVERMAN, B. W. *Density Estimation*. London: Chapman and Hall. (1986)
- [6] CLEVELAND, WILLIAM S. *The Elements of Graphing Data*. Pacific Grove, Wadsworth & Advanced Book Program. (1985)
- [7] HOROVÁ, I.; KOLÁEK, J.; ZELINKA, J. *Kernel Smoothing in MATLAB: Theory and Practice of Kernel Smoothing*. Singapore: World Scientific Publishing. (2012)
- [8] SCHILLING, M. F. AND WATKINS, A. E. *A suggestion for sunflower plots*. The American Statistician, 48, 303-305. (1994)
- [9] HINTZE, J. L., NELSON, R. D. *Violin Plots: A Box Plot-Density Trace Synergism*. The American Statistician 52, 181-184. (1998)
- [10] COHEN, A. *On the graphical display of the significant components in a two-way contingency table*. Communications in Statistics—Theory and Methods. (1980)
- [11] FRIENDLY, M. *Graphical methods for categorical data*. SAS User Group International Conference Proceedings, 17, 190–200. (1992)
- [12] FRIENDLY, M. *A Fourfold Display for 2 by 2 by k Tables* Psychology Department. York University. Report Number: 217. (1995)
- [13] VENN, J. *On the employment of geometrical diagrams for the sensible representations of logical propositions* Proceedings of the Cambridge Philosophical Society, 4 : 47-59. (1880)
- [14] SARKAR, DEEPAYAN *Lattice: Multivariate Data Visualization with R*, Springer. (2008)
- [15] MUNZNER, T., *Visualization Analysis and Design*. AK Peters, 2014

- [16] MURRAY, S., *Interactive Data Visualization for the Web*. First edition, O'Reilly 2013.
- [17] IHAKA, ROSS, AND ROBERT GENTLEMAN., *R: a language for data analysis and graphics*. Journal of computational and graphical statistics 5.3 (1996): 299-314.