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

Elsa Corral

1 de noviembre de 2019

Índice general

Capítulo 1

Introduction

1.1. Abstrac

The aim of this project is to develop a document containing the most relevant data visualization functions, as well as fulfilling a deep analisys and classification of the different availables visualization techniques.

To acomplish it, we have been working on a GNU enviroment called R project". This enviroment provides a wide variety of statistical tools that can be integrated with different databases and moreover, allow users to create or complete existing functions uploading their own packages. R is supported by a large and active community of users which provides R with a high potential.

We are now in the Big Data and Data Science era. In the technology, information and Social Media era, there is no limit generating and storing data. We have an overwhelming amount of information, thus, synthesising and processing phase turns out to be a challenging task. At that point, data visualization appears to be something imperative, essential to find qualifying information. This proyects is focused in that phase, Data Visualization is one of the most effective ways of trasmitting and perceiving information, more specifically, network data visualization will be discussed.

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

1.2. Resumen

La razón de este proyecto, es la de desarrollar un documento que contenga las técnicas de representación más relevantes sobre la visualización de datos, así como realizar un análisis y clasificación de las mismas, que pueda servir de guía o ayuda a los usuarios.

Para ello se ha trabajado sobre un entorno GNU llamado R Poject. 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 funcionalidades mediante la inclusión de sus propios paquetes. Cuenta con una numerosa y activa comunidad que proporciona un gran potencial a la herramienta.

Nos encontramos en el área del Big Data y del Data Science. En la era de la tecnología, de la información y de las redes sociales, no hay limite a la hora de generar y almacenar datos, pero sintetizar y extraer información útil de tal cantidad, se convierte en una imprescindible y laboriosa tarea. Este proyecto se centrará en la fase de visualización de datos, sin duda una de las formas mas eficaces tanto de percibir como de transmitir información, más en concreto, nos centraremos en la visualización de redes de datos.

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

1.3. Introducción

Como manera de introducción a este documento, me gustaría destacar la estrecha relación que guarda mi grado, sistemas de la información, con el presente proyecto. Durante los últimos años los sistemas de información constituyen uno de los principales ámbitos de estudio en el área de organización de empresas. El entorno donde las compañías desarrollan sus actividades se vuelve cada vez más complejo, si bien los recursos básicos analizados hasta ahora eran tierra, trabajo y capital, ahora la información aparece como otro componente fundamental a valorar en las empresas. Todo sistema de información utiliza como materia prima los datos, los cuales almacena, procesa y transforma para obtener como resultado final información útil.[1] [2] Justo en esta última parte, dónde los datos se transforman en información, es donde entra en juego la visualización de datos, en la cual profundizaremos a lo largo del documento.

Uno de los motivos que me llevaron a escoger la visualización de datos, es la relación que, para mi, mantiene con el arte. Al fin y al cabo, el objetivo de ambas disciplinas es el de trasmitir. Uno de los factores que facilita la comprensión, es la aceptación visual. Se trata de conseguir representaciones claras, limpias, estéticas y visualmente agradables. Las herramientas hoy en día son tan potentes, tienen tal capacidad gráfica y permiten tantas personalizaciones, que dan lugar a representaciones realmente impresionantes.

1.4. Objetivos del trabajo

Se desea realizar un análisis, clasificación y desarrollo de distintas visualizaciones de datos y sus respectivas funciones en R. A continuación, se llevará a cabo un desarrollo de dichas funciones indicando entre otras cosas modo de aplicación, argumentos y dónde se encuentra en R. Estarán dentro del alcance de este proyecto las visualizaciones de redes, también en forma de árboles y áreas.

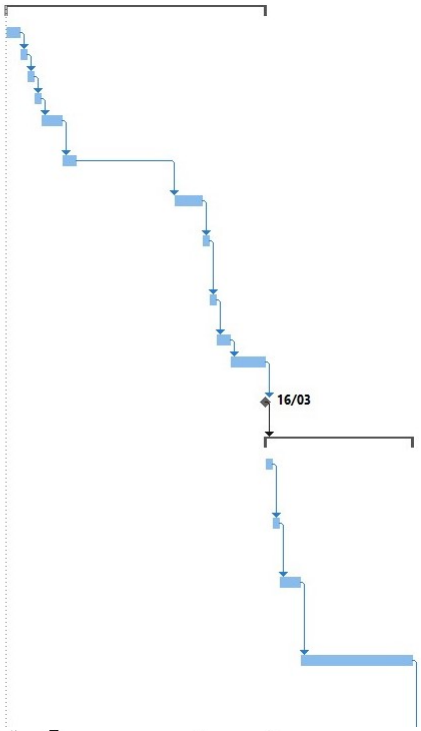
Este documento pretende ser de utilidad para toda la comunidad R, sirviendo de referencia y de ayuda para los usuarios que comiencen a adentrarse tanto en la herramienta como en el área de la visualización de datos.

1.5. Planificación del trabajo

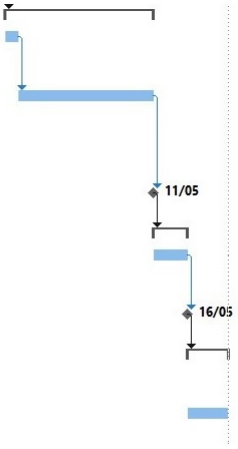
Este apartado explica la planificación seguida para llevar a cabo este proyecto, la división de tareas, la gestión del tiempo y los hitos de entrega planificados.

El primer esquema se realizó mediante MS Project en forma de diagrama de Gantt. Los diagramas que se muestran en esta sección hacen referencia a la planificación incial, realizada en febrero de 2016. El desarrollo real del proyecto sufrió algunas variaciones respecto a las fechas y plazos estimados, sin embargo tanto las tareas como sus duraciones han sido fieles a lo esperado.

	Primera Entrega	**19 días**	**lun 08/02/16**	**mié 16/03/1**	
	Resumen español	2 días	lun 08/02/16	mar 09/02/1	
	Resumen Inglés	1 día	mié 10/02/16	mié 10/02/1	3
	Palabras Clave	1 día	jue 11/02/16	jue 11/02/16	4
	Introducción	1 día	vie 12/02/16	vie 12/02/16	5
	Objetivos del trabajo	2 días	sáb 13/02/16	lun 15/02/16	6
	Enfoque y metodología	2 días	mar 16/02/16	mié 17/02/16	7
	Planificación del proyecto	3 días	jue 03/03/16	dom 06/03/16	8
	Descr. Siguientes capítulos de la memoria	1 día	lun 07/03/16	lun 07/03/16	9
	Estudiar posibles clasificaciones	1 día	mar 08/03/16	mar 08/03/16	10
	Estado del arte	2 días	mié 09/03/16	jue 10/03/16	11
	Primer capítulo:Sobre R	3 días	vie 11/03/16	mar 15/03/16	12
	Reunion revisión primera entrega	0 días	mié 16/03/16	mié 16/03/16	13
	Segunda entrega	**14 días**	**mié 16/03/16**	**mié 06/04/1**	14
	Correcciones de la primera entrega	1 día	mié 16/03/16	mié 16/03/16	
	División y estructuración de los capítulos	1 día	jue 17/03/16	jue 17/03/16	16
	Recopilar paquetes y documentación de internet	2 días	vie 18/03/16	dom 20/03/16	17
	Comenzar a clasificar funciones por capítulos	10 días	lun 21/03/16	mar 05/04/16	18



	Tercera Entrega	**19 días?**	**mié 20/04/16**	**mié 11/05/1**	20
	Correcciones de la segunda entrega	2 días	mié 20/04/16	jue 21/04/16	
	Completar capitulos restantes de clasificación de funciones.	17 días	vie 22/04/16	mié 11/05/16	22
	Revisión tercera entrega	0 días?	mié 11/05/16	mié 11/05/16	23
	Cuarta Entrega	**4 días?**	**jue 12/05/16**	**lun 16/05/16**	**24**
	Conclusiones, glosario y Bibliografía	4 días	jue 12/05/16	lun 16/05/16	
	Revisión cuarta entrega	0 días?	lun 16/05/16	lun 16/05/16	26
	Última entrega(Revisión final)	**5 días**	**mar 17/05/16**	**dom 22/05/16**	**27**
	Últimas correcciones	5 días	mar 17/05/16	dom 22/05/16	



1.5.1. Enfoque y metodología

El primer reto al que nos enfrentamos a la hora de comenzar a establecer las pautas para la realización del documento, fue la utilización de LaTeX como editor de texto. LaTeX tiene tanto ventajas como inconvenientes. Una de las ventajas más destacables es que se trata de una herramienta de estructuración multiplataforma, su esqueleto se mantiene estable, a veces ocurre que con otros editores, los formatos se descolocan al cambiar de versión, de ordenador o de extensión, LaTeX evita estos problemas. Su desventaja, es su curva de aprendizaje, se trata de un lenguaje complejo de manejar, y cuesta acostumbrarse, incluso para usuarios medios o avanzados es conveniente tener cerca un pequeño manual o resumen de ayuda.

Una vez decidido este punto, pasamos a construir la estructura del proyecto y a ordenar todos los puntos, construimos el índice. A continuación daremos una pequeña explicación de lo que podrás encontrar en los principales apartados del documento.

- Introducción: En esta sección se encuentra el resumen, la introduccón al proyecto, donde se explica la relación que guarda con el grado y las razones que me llevaron a escogerlo, los objetivos, donde se comenta la razón de este documento, esta sección de planificación y metodologías, y por último una sección en la que argumentamos por qué escogimos R como herramienta de trabajo.

- Visualización de Datos: En este capítulo introducimos y tratamos de explicar la visualización de los datos y concrétamente las redes. Realizamos una pequeña clasificacón donde incluimos diferentes tipos de visualizaciones incluyendo una pequeña imagen para facilitar su comprensión.

- Paquetes en R para la visualziación de Datos: Se describe el paquete base de R.

- Análisis comparativo de los paquetes: En esta sección incluimos las funciones que permiten realizar las visualizaciones de datos explicadas en el segundo capítulo, se incluye información como los argumentos o los paquetes donde encontrarlas.

- Conclusiones y trabajos futuros: Cconclusiones extraídas tras la realización de este proyecto y proposición de futuras mejoras y trabajos.

- Comunidad R: Filosofía de la comunidad R y recopilación de fuentes de información y ayuda sobre R.

- Glosario: Se incluyen palabras técnicas usadas a lo largo del documento.

- Bibliografía: Recopilación de todas las referencias bibliográficas.

1.6. ¿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 qué es R y las razones que nos han llevado a escogerlo como nuestra herramienta de trabajo.

En su definición más amplia, R es un lenguaje de programación que permite al usuario programar algoritmos y usar los que ya han sido programados por otros. Esta vaga descripción puede ser aplicada a todos los lenguajes de programación, sería más útil decir todo lo que R puede hacer. Algunos dicen que R puede hacer todo lo que imaginas, y esta lejos de ser una exageración. Con R puedes escribir funciones, hacer cálculos, aplicar miles de técnicas estadísticas, crear todo tipo de gráficos, incluso crear tus propias librerías de funciones. Muchos institutos de investigación, empresas y universidades ya emplean R. Cientos de libros con referencias y aplicaciones en R se han publicado en los últimos 10 años. A continuación analizamos en profundidad las características principales de R que nos han llevado a decantarnos.[3] [4]

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 clave con el 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.

5. Comunidad: Además de todo lo dicho anteriormente sobre su comunidad de usuarios, existen otras ventajas derivadas de esta característica. Muchos usuarios de R ayudan a otros usuarios a comenzar en el uso de R y lo recomiendan en sus entornos de trabajo y en sus círculos profesionales. Estos usuarios comienzan a ser activos en las listas de email de R (https://www.r-project.org/mail.html) y en las páginas de preguntas y respuestas como Stack Overflow o stackexchange. Ademas los usuarios son muy activos en redes sociales como Twitter y en conferencias de R.

6. Integración con otros lenguajes: A medida que los usuarios comienzan a decantarse por R, comienzan a intentar combinar sus antiguos procesos y métodos, lo que lleva a una cantidad enorme de paquetes para conectar R con sistemas de ficheros, bases de datos y muchas otras aplicaciones. Por ejemplo puede integrarse con Oracle, ODFBC,PostgreSQL,SPSS,SAS,JMP... muchos de ellos han sido incluidos a la instalación base de R, como formatos Excel o PDF.

Capítulo 2

Visualización de datos

Un paso muy importante en la metodología del Data Science es obtener una representación visual de los datos. Esto proporciona múltiples ventajas: primero, como humanos, extraemos de manera más eficaz la información de señales visuales, porque una representación visual es normalmente más intuitiva que una representación textual. Segundo, la visualización de los datos, en gran parte, también engloba una fase de tratamiento y síntesis. La visualización ofrece una instantánea muy concisa de los datos, y como dice el refrán, una imagen vale más que mil palabras".

El Objetivo de la visualización de datos es el de transmitir una historia al espectador. Crear visualizaciones efectivas es similar a ser un buen cuenta cuentos: consiste en ser capaz de transmitir información interesante a el nivel justo de detalle. Este hecho convierte la visualización de datos en un arte más que en una ciencia exacta. La presentación estética es igual de importante que los aspectos estadísticos que pretendemos mostrar. Esto no quiere decir que podamos paliar las lagunas de nuestro análisis con trucos de estilos de presentación. El objetivo es conseguir una análisis transparente, para que el espectador comprenda y aprecie la imagen al completo, mediante una presentación de los datos clara y efectiva. [5]

2.1. Redes

Desde cierto punto de vista todo forma parte de una red. Ya sea una red de relaciones interpersonales, de referencias bibliográficas, de referencias intertextuales, de ecosistemas, de células, de economías, de personas ... de casi cualquier cosa que podamos imaginar. Esas redes tienen una arquitectura y su estudio ha supuesto una especie de revolución para la compresión de muchas áreas de la ciencia.

Estamos en la era de la información, de la revolución informática, de las redes sociales, de compartir e intercambiar datos. Almacenamos cantidades inimaginables de información, pero surge la necesidad de hacer algo con ello, de aprovecharlo y de entenderlo. Las herramientas que nos permiten analizar esos datos, evolucionan rápidamente y cada vez nos permiten manejar cantidades mayores. Los diagramas y las representaciones de redes suelen ser una gran ayuda para lograr comprender de manera global este tipo de información. Los gráficos de

redes, comúnmente conocidos, son los que se suelen usar para describir sistemas comunicación, como redes telefónicas, sistemas informáticos o el propio internet. Abstrayéndose se pueden distinguir nodos, segmentos de conexiones y a veces incluso puede incluirse información espacial, pueden tener forma de árbol, pueden ser agrupaciones de datos, o pueden ser simples conexiones sin ninguna estructura ni patrón.

Una de las mayores complicaciones, surge a la hora de escoger que tipo de representación sería la más adecuada para nuestros datos. Lo primero es saber que aspecto es el que se quiere resaltar y estudiar. A lo largo de este capítulo haremos una recopilación de algunas funciones en R para realizar este tipo de visualizaciones. Para su clasificación, atenderemos a las características o las propiedades que se quieran destacar en cada uno de los gráficos. De esta manera, surgen tres grandes grupos: La visualización de redes, dónde se englobarán gráficos que pretendan destacar nodos claves o fuerza en las relacciones. La red como un mapa, dónde se incluyen árboles,dendrogramas, o redes representadas sobre planos, en esta categoría se pretende destacar propiedades estructurales y patrones de difusión. La última categoría es la visualización de redes en comunidades o áreas, se pretende enfatizar patrones de agrupación, la formación de comunidades o clusters. [6] [7]

2.1.1. Visualización de redes

En esta sección se incluyen visualizaciones de redes cuyo objetivo sea destacar nodos o actores clave, cantidad de conexiones o plasmar fuerza e intensidad en las relaciones.

Diagrama de conexiones

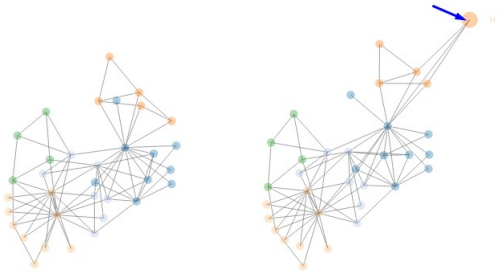
Imagen

Descripción

Este gráfico permite ver las conexiones existentes entre diferentes nodos. Se representa como una coleccin de objetos relacionados. Nos referimos a los objetos como nodos o vértices, y por lo general los dibujamos como puntos. Nos referimos a las conexiones entre los nodos como links, y por lo general los dibujamos como líneas entre los puntos.

Diagrama de conexiones Interactivo

Imagen

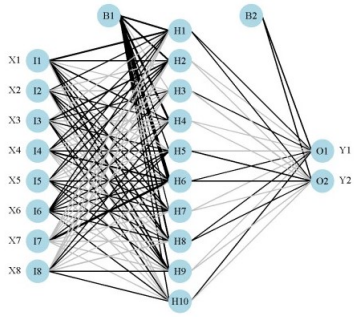


Descripción

Permite convertir el diagrama anterior en un gráfico interactivo. Se puede interactuar con los diagramas de redes moviendo la posición de los nodos con el cursor, lo que permite ver de una manera más clara las conexiones.

Redes neuronales

Imagen

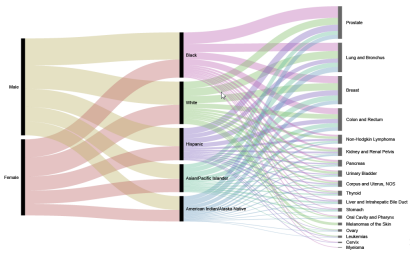


Descripción

Permite dibujar un grafo acíclico dirigido para representar redes neuronales. Un grafo acíclico dirigido puede reconocerse porque no tiene ciclos; esto significa que para cada vértice v, no hay un camino directo que empiece y termine en v.

Gráfico aluvial

Imagen

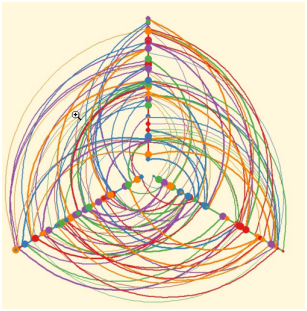


Descripción

En este tipo de diagramas, cada muestra es representada como un polígono lineal que cruza en horizontal una serie de ejes verticales que representan las variables. La anchura de los segmentos que cruzan el mapa horizontalmente representan la fuerza de la relacción.

Gráfico de colmena

Imagen

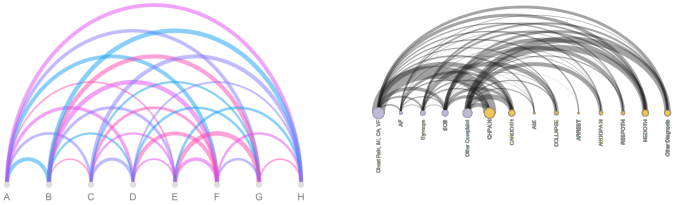


Descripción

Hive plot es un método de visualización racional para la representación de redes. Los nodos se asignan y se posicionan en ejes lineales distribuidos radialmente - esta asignación se basa en las propiedades estructurales de la red. Los bordes se dibujan como enlaces curvos.

Diagrama de Arcos

Imagen

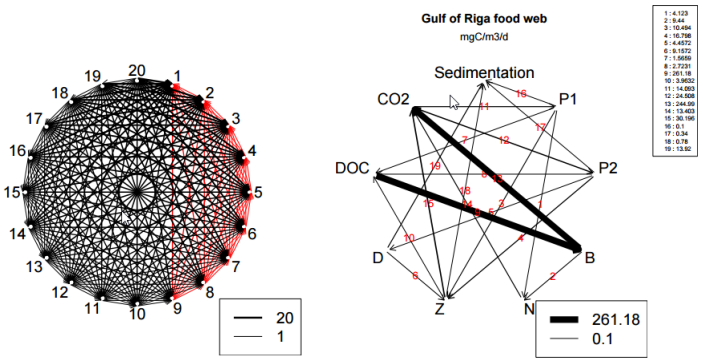


Descripción

Se especifican distintos nodos a lo largo de un eje horizontal, las relaciones y conexiones se representan con arcos de distintas medidas y colores enlazando nodos. Estos diagramas pueden ser efectivos para encontrar co-ocurrencias en los datos.

Diagrama de flujos entre conexiones

Imagen

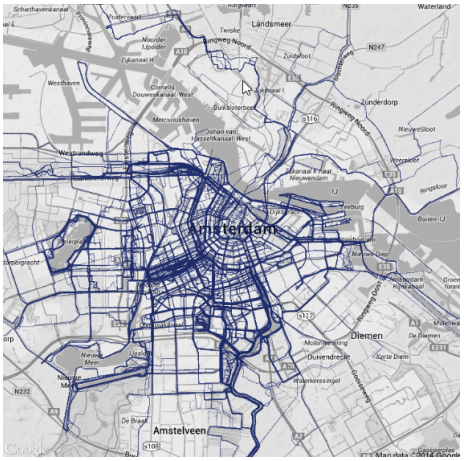


Descripción

Se representan los flujos entre las conexiones mediante segmentos que varían el tamaño atendiendo a las variables de entrada. Se distribuyen de manera circular.

Red de direcciones

Imagen



Descripción

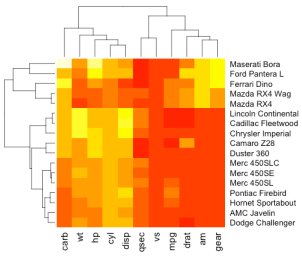
Esta visualización permite representar caminos,rutas o direcciones, se pueden representar sobre un plano vacío o sobre un mapa previamente plasmado.

2.1.2. Visualización de redes como árboles o mapas

En esta sección se incluirán visualizaciones de redes cuya intención sea destacar propiedades estructurales o patrones de difusión en los datos.

Mapa de calor

Imagen

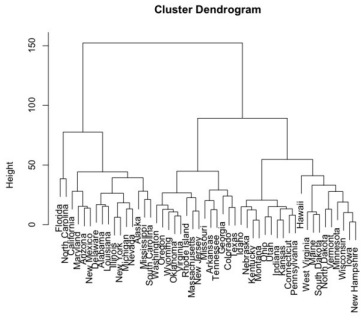


Descripción

Se trata de una representación gráfica de los datos donde los valores individuales que están contenidas en una matriz son representados como colores.

Dendrograma

Imagen



Descripción

Se trata de un diagrama que permite organizar los datos en subcategorías que se van dividiendo en otras hasta llegar al nivel de detalle deseado.

Árbol Radial

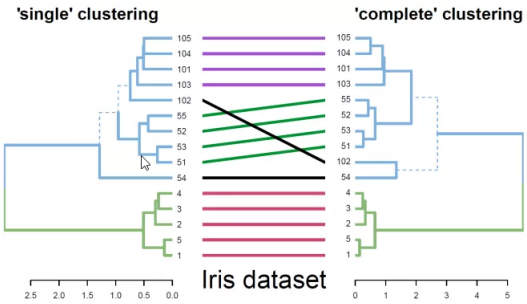
Imagen

Descripción

Se trata de un diagrama que permite representar árboles o dendrogramas disponiendo las hojas o nodos de manera circular alrededor de un punto central.

Comparación de dendrogramas

Imagen

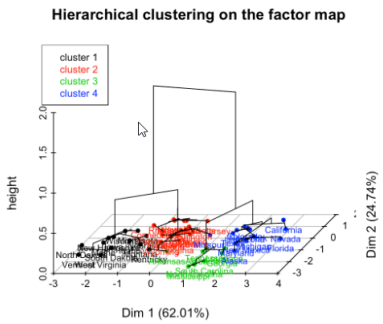


Descripción

Permite comparar dos dendrogramas visualmente. Se unirán los extremos o hijos que compartan etiqueta o valor.

Dendrogramas en 3D

Imagen

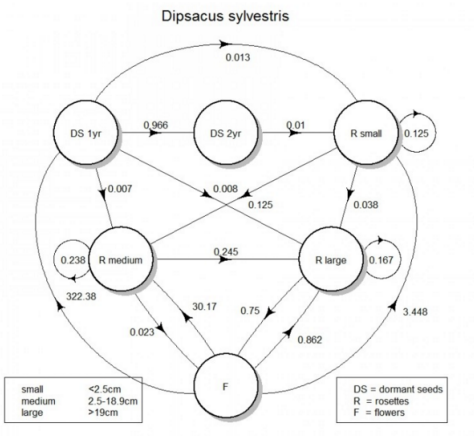


Descripción

Permite visualizar un dendrograma en tres dimensiones. Esto permite observar mejor las dependencias, la dispersión y otras propiedades estructurales.

Diagrama de transición

Imagen

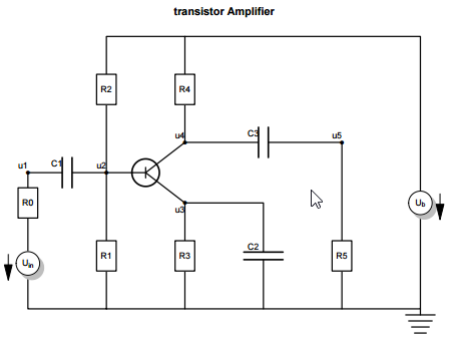


Descripción

Permite visualizar una matriz de transición como en forma de cajas o nodos conectados por links o flechas.

Circuito electrónicos

Imagen

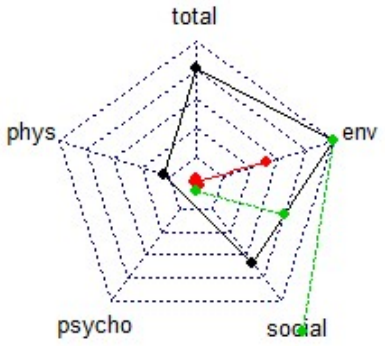


Descripción

Permite dibujar circuitos electrónicos añadiendo los elementos sobre un diagrama base.

Gráfico de araña

Imagen

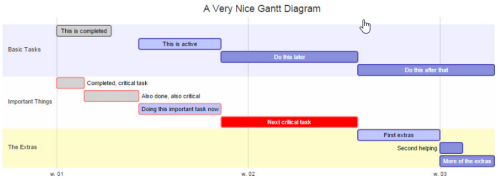


Descripción

Se trata de un gráfico radial, también conocido como un gráfico de estrella debido a su apariencia, traza los valores de cada categoría en un eje independiente que se inicia en el centro del gráfico y termina en el anillo exterior.

Diagrama de Gantt

Imagen



Descripción

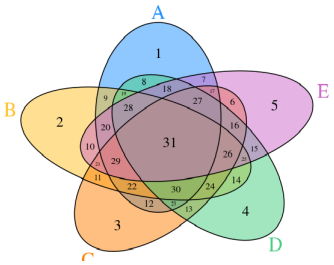
El objetivo de este diagrama es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado. Las tareas se representan en forma de cajas y se enlazan sucesivamente a lo largo de la línea temporal. Dentro de las cajas se puede especificar todo tipo de información, como recursos, posibles desviaciones y observaciones.

2.1.3. Visualización de redes en comunidades o áreas

En esta sección se incluirán visualizaciones de redes cuyo objetivo sea permitir detectar fácilmente comunidades, grupos o áreas entre los datos a estudiar.

Diagrama de Venn

Imagen

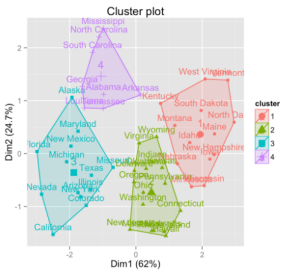


Descripción

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.

Agrupación en clusters

Imagen

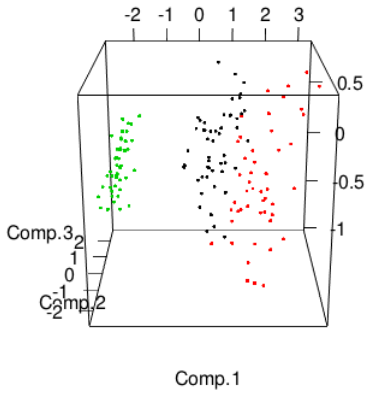


Descripción

Permite visualizar los datos clusterizados. Se realizan agrupaciones de datos con similares características en diferentes áreas o comunidades.

Clusterización en 3D e interactivo

Imagen

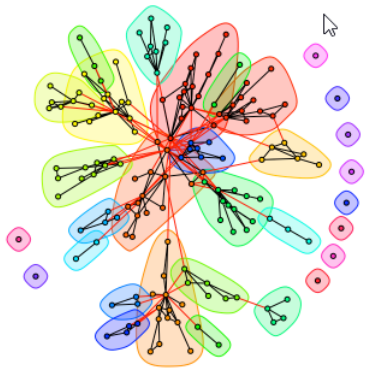


Descripción

Permite visualizar los datos clusterizados en 3 dimensiones y además permite interactuar con el gráfico.

Visualización para detectar comunidades

Imagen

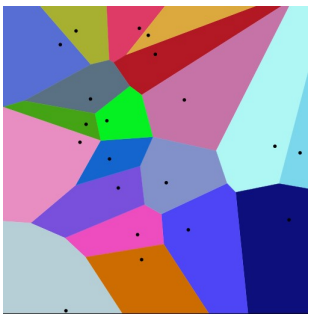


Descripción

Esta visualización permite detectar comunidades o subárboles densamente conectados dentro de un diagrama de red.

Diagrama de Voronoi

Imagen

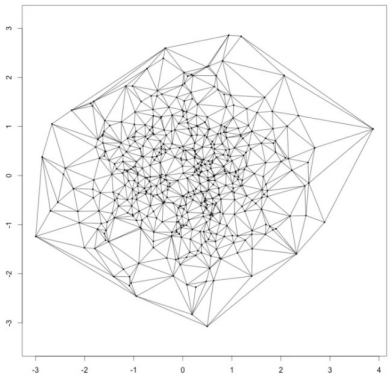


Descripción

Permite dibujar un diagrama de Voronoi, o también llamado polígonos de Thiessen. Los segmentos que conforman este diagrama se crean al unir los puntos entre sí, trazando las mediatrices de los segmentos de unión. Las intersecciones de estas mediatrices determinan una serie de polígonos en un espacio bidimensional alrededor de un conjunto de puntos de control, de manera que el perímetro de los polígonos generados sea equidistante a los puntos vecinos. Muy conocido y recurrido en los Sistemas de Información Geográfica. [9]

Triangulación de Delaunay

Imagen

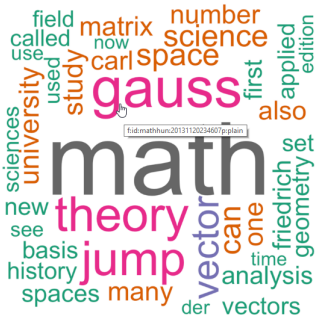


Descripción

Permite dibujar un gráfico que cumple la condición de Delaunay. Esta condición dice que la circunferencia circunscrita de cada triángulo de la red no debe contener ningún vértice de otro triángulo.

Nube de palabras

Imagen

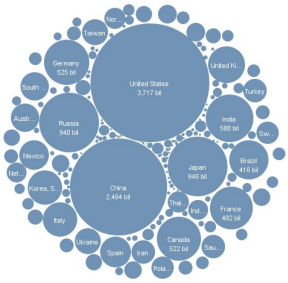


Descripción

Esta visualización permite representar la frecuencia con que las palabras aparecen en un texto. El diagrama es una nube de palabras, cuanto mayor es la frecuencia, mayor es el tamaño de la palabra, y más centrada aparece.

Diagrama de burbujas

Imagen

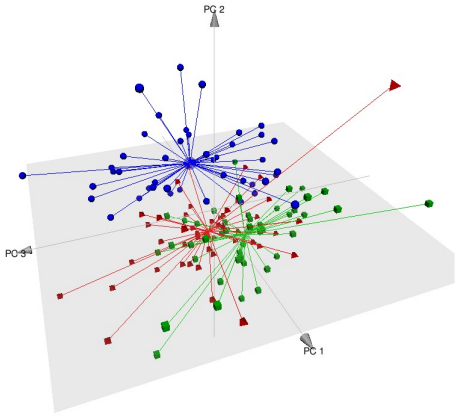


Descripción

Permite dibujar un gráfico de burbujas. Un gráfico de burbujas se utiliza para visualizar un conjunto de datos con de 2 a 4 dimensiones. Las dos primeras dimensiones se visualizan como coordenadas, la tercera es el color y la cuarta es el tamaño.

Diagrama de red de clusters 3D

Imagen



Descripción

Esta visualización permite representar los datos en un mapa 3D, permitiendo interactuar con el gráfico, consiguiendo que los clusters de datos y los patrones de difusión se puedan observar mejor.

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. Actualmente existen numerosas plataformas y lenguajes que tratan de resolver la problemática actual al tratamiento de los datos. Todas ellas están en continua actualización, incluyendo nuevos algoritmos, nuevas funciones y simplificando procesos. Como ya contábamos anteriormente, R es extensible mediante paquetes y su basta comunidad, contribuye a esta actualización continua, aportando sus propios contenidos y mejoras. A continuación analizaremos algunas funciones que nos permiten realizar los diferentes tipos de gráficos que describíamos al comienzo del documento, indicando a que paquete pertenecen y otras características.

3.1. Paquete base de R

El paquete base de R viene con la distibución del programa. Constituye el núcleo de R y contiene las funciones básicas del lenguaje para leer y manipular datos, algunas funciones gráficas y algunas funciones estadísticas (regresión lineal y análisis de varianza). Cada paquete contiene un directorio denominado R con un archivo con el mismo nombre del paquete(por ejemplo, para el paquete base, existe el archivo R HOME/library/base/R/base). Este archivo esta en formato ASCII y contiene todas las funciones del paquete. [8]

Capítulo 4

Análisis comparativo de los paquetes

4.1. Visualización de redes

4.1.1. Diagrama de conexiones

Paquete base de R

No lo tiene

Función:

forceNetwork()

Argumentos son:

(Links, Nodes, Source, Target, Value, NodeID, Nodesize, Group, height, width, colourScale, fontSize, fontFamily, linkDistance, linkWidth, radiusCalculation, charge, linkColour, opacity, zoom, legend, bounded, opacityNoHover, clickAction)

Links: un objeto de trama de datos con los enlaces entre los nodos. Debe incluir el origen y el destino de cada enlace. Estos deben ser numerados empezando por 0. Una variable de valor opcional se pueden incluir para especificar qué tan cerca están los nodos entre sí.

Nodes: una trama de datos que contiene el ID de nodo y las propiedades de los nodos. Si no se especifica ningún ID continuación, los nodos deben estar en el mismo orden que la columna de la variable Fuente en el marco de los enlaces de datos. Actualmente sólo se permite una variable de agrupación.

Source: cadena de caracteres nombrando la variable origen de la red en el marco de los enlaces de datos.

Target: cadena de caracteres nombrando la variable de destino de red en el marco de los enlaces de datos.

Value: cadena de caracteres nombrando la variable en el marco de enlaces de datos para cuán amplio los enlaces son.

NodeID: cadena de caracteres que especifica los identificadores de nodo en el marco de los nodos de datos.

Nodesize: cadena de caracteres que especifica una columna en la trama de datos de nodos en algún valor para variar el radio nodo. Ver también radiusCalculation.

Group: cadena de caracteres que especifica el grupo de cada nodo en el marco de los nodos de datos.

height: altura numérico para el área de gráficos de redes en píxeles.

width: ancho numérico para el área de gráficos de redes en píxeles.

colourScale: cadena de caracteres que especifica la escala de color categórica para los nodos.

fontSize: tamaño de la fuente numérico en píxeles para las etiquetas de texto nodo.

fontFamily: familia de la fuente de las etiquetas de texto nodo.

linkDistance: cadena numérica o de caracteres. De cualquier distancia numérica fija entre los enlaces en píxeles (actualmente arbitraria en relación con la medida de presentación). O una función de JavaScript, posiblemente con el peso por valor. Por ejemplo: linkDistance = JS ("function(d) return d.value * 10").

linkWidth: cadena numérica o de caracteres. Puede ser un ancho fijo numérico en píxeles (arbitraria en relación con la medida de presentación). O una función de JavaScript, posiblemente con el peso por valor. El valor predeterminado es linkWidth = JS ("function(d) return Math.sqrt (d.value);").

radiusCalculation: cadena de caracteres. Una expresión matemática javascript, para ponderar la radio por Nodesize. El valor por defecto es el cálculo del radio = JS ("Math.sqrt (d.nodesize) 6").

charge: valor numérico que indica si la fuerza de la repulsión nodo (valor negativo) o atracción (valor positivo).

linkColour: vector de caracteres que especifica el color (s) desea que las líneas de enlace que sean. Múltiples formatos soportados (por ejemplo hexadecimal).

opacity: valor numérico de la proporción opaco que le gustan los elementos del gráfico para ser.

zoom: valor lógico para activar (TRUE) o desactivar (FALSE) zoom.

legend: valor lógico para habilitar leyendas de color de nodo.

bounded: valor lógico para activar (TRUE) o desactivar (FALSE) la caja de contorno que limita la extensión del gráfico.

opacityNoHover: valor numérico del porcentaje de opacidad para el nodo etiquetas de texto cuando el ratón no está sobre ellos.

clickAction: cadena de caracteres con una expresión JavaScript para evaluar cuando se hace clic en un nodo.

Paquetes requeridos

Paquete "network3D"

Referencias

https://cran.r-project.org/web/packages/networkD3/networkD3.pdf

http://christophergandrud.github.io/networkD3/

4.1.2. Diagrama de conexiones interactivo

Paquete base de R

No lo tiene

Función:

d3SimpleNetwork()

Argumentos son:

(Data, Source, Target, height, width, fontsize, linkDistance, charge, linkColour, nodeColour, nodeClickColour, textColour, opacity, parentElement, standAlone, , iframe, d3Script)

Data: a data frame object with three columns. The first two are the names of the linked units. The third records an edge value. (Currently the third column doesnt affect the graph.)

Source character string naming the network source variable in the data frame. If Source = NULL then the first column of the data frame is treated as the source.

Target character string naming the network target variable in the data frame. If Target = NULL then the second column of the data frame is treated as the target.

height numeric height for the network graphs frame area in pixels.

width numeric width for the network graphs frame area in pixels.

fontsize numeric font size in pixels for the node text labels.

linkDistance numeric distance between the links in pixels (actually arbitrary relative to the diagrams size).

charge numeric value indicating either the strength of the node repulsion (negative value) or attraction (positive value).

linkColour character string specifying the colour you want the link lines to be. Multiple formats supported (e.g. hexadecimal). 10 d3SimpleNetwork

nodeColour character string specifying the colour you want the node circles to be. Multiple formats supported (e.g. hexadecimal).

nodeClickColour character string specifying the colour you want the node circles to be when they are clicked. Also changes the colour of the text. Multiple formats supported (e.g. hexadecimal).

textColour character string specifying the colour you want the text to be before they are clicked. Multiple formats supported (e.g. hexadecimal).

opacity numeric value of the proportion opaque you would like the graph elements to be.

parentElement character string specifying the parent element for the resulting svg network graph. This effectively allows the user to specify where on the html page the graph will be placed. By default the parent element is body.

standAlone logical, whether or not to return a complete HTML document (with head and foot) or just the script. file a character string of the file name to save the resulting graph. If a file name is given a standalone webpage is created, i.e. with a header and footer. If file = NULL then result is returned to the console.

iframe logical. If iframe = TRUE then the graph is saved to an external file in the working directory and an HTML iframe linking to the file is printed to the console. This is useful if you are using Slidify and many other HTML slideshow framworks and want to include the graph in the resulting page. If you set the knitr code chunk results='asis' then the graph will be rendered in the output. Usually, you can use iframe = FALSE if you are creating simple knitr Markdown or HTML pages. Note: you do not need to specify the file name if iframe = TRUE, however if you do, do not include the file path.

d3Script a character string that allows you to specify the location of the d3.js script you would like to use. The default is http://d3js.org/d3.v3.min.js.

Paquetes requeridos

Paquete "network3D"

Referencias

https://cran.r-project.org/web/packages/networkD3/networkD3.pdf

http://christophergandrud.github.io/networkD3/

4.1.3. Gráfico aluvial

Paquete base de R

No lo tiene

Función:

alluvial()

Argumentos son:

(..., freq, col, border, layer, hide, alpha, gap.width, xw, cw)

...: vectors or data frames, all for the same number of observations

freq: Numérico, vector de frecuencias de la misma longitud que el número de observaciones.

col: Vector con los colores de las líneas.

border: Vector con los colores de los bordes de las líneas.

layer: Numérico, orden para representar las líneas.

hide: Logico, indica si la línea en particular se debería representar.

alpha: Numérico, vector con la transparencia de las líneas.

gap.width: Numérico, el espacio relativo de los huecos entre las categorías.

xw: Numérico, la distancia entre el eje a los puntos de control del xspline.

cw: Numérico, anchura del eje de categorias.

Paquetes requeridos

Paquete 'Alluvial'.

Referencias

http://www.rdocumentation.org/packages/alluvial/functions/alluvial

4.1.4. Gráfico de Colmena

Paquete base de R

No lo tiene

Función:

Plothive()

Argumentos son:

(HPD, ch, method, dr.nodes, bkgnd, axLabs, axLab.pos, axLab.gpar, anNodes, anNode.gpar, grInfo, arrow, np, anCoord, ...)

HPD: Un objeto de clase S3 HivePlotData.

Ch: Numérico; el tamaño del círculo central.

Method: Carácter. Se le pasa a manipAxis (ver manipAxis para comprobar valores válidos por defecto se emplean las coordenadas nativas o absolutas de los datos).

dr.nodes: Lógico; si es TRUE se dibujarán los nodos.

Bkgnd: Cualquier color válido. Usado como color de fondo.

axLabs: Un vector de cadenas de caracteres para las etiquetas de los ejes.

axLab.pos: Numérico; Un desplazamiento desde el extremo del eje para la colocación de etiquetas. Ya sea un solo valor o un vector de valores. Si se trata de un solo valor, todas las etiquetas estarán desplazadas la misma cantidad. Si es un vector de valores, debe haber un valor para cada eje. Las unidades dependen del método empleado.

axLab.gpar (aplicable solo a plotHive) Una lista de parejas nombre-valor válidas para gpar Esto controla las etiquetas y la disposición de las flechas.

anNodes (aplicable solo a plotHive) La ruta a un archivo csv que contenga información para las etiquetas de los nodos. Si se especifica, se dibujará un segmento desde el nodo al texto especificado. El texto se posicionará al final de dicho segmento. Las columnas del fichero csv deben ser nombradas como sigue (la descripción y el modo de uso entre paréntesis): node.lab (etiqueta del nodo de HPD*nodes*lab), node.text (el texto que será mostrado en la impresión), angle (coordenadas polares: ángulo en el cual se debe dibujar el segmento), radius (coordenadas polares: ángulo en el cual se debe mostrar el texto), offset (distancia adicional a lo largo del vector de radio para desplazar el textoo), hjust, vjust (justificación horizontal y vertical; nominal como [0. . . 1] pero números fraccionales y negativos también se admiten). Los primeros dos valores serán tratados como de tipo carácter, los otros como numéricos.

anNode.gpar (aplicable solo a plotHive): Una lista de parejas nombre-valor válidas para gpar. Esto controla tanto el texto usado para anotar los nodos como los segmentos que conectan esos textos a los nodos.

grInfo (aplicable solo a plotHive): La ruta a un archivo csv que contenga información para añadir decoraciones gráficas a la representación.Si se especifica, se incluirá una línea desde el nodo hasta la ubicación indicada en el gráfico. La ruta debe contener la extensión del fichero.

arrow (aplicable solo a plotHive): Un vector de 5 o 6 valores, una cadena de caracteres para etiquetar la flecha,4 valores numéricos que darán el ángulo de la flecha, y un valor para separar la etiqueta de la flecha.

np (aplicable solo a plotHive):Lógico, debe abrirse una nueva ventana al representar la gráfica?

... : Parámetros adicionales.

Paquetes requeridos

Paquete HiveR

Referencias

https://cran.r-project.org/web/packages/HiveR/HiveR.pdf

http://academic.depauw.edu/ hanson/HiveR/HiveR.html

4.1.5. Diagrama de Arcos

Paquete base de R

No lo tiene

Función:

arcplot()

Argumentos son:

(edgelist, nodes ordering, graphical attributes)

Lista de ejes: Una matriz de dos columnas con la lista de cada par de extremos de cada arco..

Nodos ordenados: Opcional, una lista de indices que permita ordenar los nodos, por numero de conexiones, por peso,etc.

Atributos gráficos:podemos modificar gráficamente los distintos elementos,labels,nodes,arcs...

Paquetes requeridos

Paquete "devtools".

Paquete .ªrcdiagram"

Referencias

http://gastonsanchez.com/software/arcdiagram_introduction.pdf

http://www.r-bloggers.com/arc-diagrams-in-r-les-miserables/

http://hint.fm/papers/arc-diagrams.pdf

4.1.6. Diagrama de flujos entre conexiones

Paquete base de R

No lo tiene.

Función:

plotweb()

Argumentos son:

(flowmat, names, lab.size, add, fig.size, main, sub, sub2, log, mar, nullflow, minflow, maxflow, legend, leg.digit, leg.title, lcol, arr.col, val, val.digit, val.size, val.col, val.title, val.ncol, budget, bud.digit, bud.size, bud.title, bud.ncol, maxarrow, minarrow, length, dcirc, bty, ..)

flowmat: Matriz de flujo, filas=flujo *desde*, columnas=flujo *hacia*.

names: Vector de cadenas de caracteres con el nombre de los componentes.

lab.size: Tamaño relativo de la etiqueta del nombre.

add: Comienza una nueva representación(FALSE) o añade a la actual(TRUE).

fig.size: Si add=FALSE: Indica el tamaño relativo de la figura.

main: Si add=FALSE: título principal.

sub: Si add=FALSE: Sub título.

sub2: Si add=FALSE: Título al final.

log: Valor lógico indicando si escalar los valores de flujo logarítmicamente.

mar: Los márgenes de la figura.

nullflow: O un valor o un vector de dos valores; Si flow ¡nullflow[1] or flow ¿nullflow[2] (si hay dos valores): flow se asume = 0 y la flecha no se dibuja.

minflow: Valor de flujo indicando la anchura mínima de la flecha.

maxflow Valor de flujo indicando la anchura máxima de la flecha.

legend: Valor lógico indicando si añadir una leyenda.

leg.digit: Número de dígitos a escribir en la leyenda, solo si legend=TRUE.

leg.title: Título para la leyenda de la flecha, solo si legend=TRUE.

lcol: Color de la línea de la flecha.

arr.col: Color de la flecha. Un valor o una matriz con las mismas dimensiones que flowmat, si se trata de una matriz cada flecha puede tener un color.

val: Valor lógico indicando si escribir los valores de los flujos como leyenda.

val.digit: Número de dígitos para representar los valores. Solo si val=TRUE.

val.size: Tamaño de los dígitos para representar los valores. Solo si val=TRUE.

val.col: Color para escribir valores. Solo si val=TRUE.

val.title: Título para los valores. Solo si val=TRUE.

val.ncol: Número de columnas para escribir valores. Solo si val=TRUE.

budget: Valor lógico indicando si calcular el resultado(suma de los flujos entrantes-suma de los flujos salientes) por componente.

bud.digit: Número de dígitos para escribir el resultado. Solo si budget=TRUE.

bud.size: Tamaño relativo para escribir el resultado. Solo si budget=TRUE.

bud.title: Título para la leyenda del resultado. Solo si budget=TRUE.

bud.ncol: Número de columnas para escribir el resultado. Solo si budget=TRUE.

maxarrow: Anchura máxima de la flecha.

minarrow: Anchura mínima de la flecha.

length: Longitud de las esquinas de la cabeza de la flecha (en pulgadas).

bty: El tipo de caja que será dibujada en las leyendas. Puede ser .º"(por defecto) y "n".

...: Argumentos extra.

Paquetes requeridos

Paquete 'Diagram'.

Referencias

http://www.inside-r.org/packages/cran/diagram/docs/plotweb

https://cran.r-project.org/web/packages/diagram/diagram.pdf

https://cran.r-project.org/web/packages/diagram/vignettes/diagram.pdf

4.1.7. Redes o direcciones sobre mapas

Paquete base de R

No lo tiene.

Preparación de los datos

Si se desea realizar esta visualización sobre un mapa previamente cargado, se recomienda usar la función qmap(), la aplicación puede ser la siguiente: qmap('amsterdam', zoom = 12, color = 'bw'). Después se deberá añadir nuestra función, de tal manera que la función quedaría: qmap() + geom_path() .

Función

geom_path()

Argumentos son

(mapping, data, stat, position, ..., lineend, linejoin, linemitre, arrow, na.rm, show.legend, inherit.aes)

mapping: Una serie de mapeos creados por aes o aes_. Si se especifica como inherit.aes = TRUE (por defecto), se combina con el mapeo por defecto a nivel superior del gráfico.

data: Los datos a ser representados en esta capa. Hay tres opciones: Si es NULL, por defecto, los datos son heredados de la representación de la llamada a ggplot. Un data.frame u otros objetos, sobrescribirán los datos representados. Todos los objetos serán forzados a producir un data.frame, ver fortify para saber que variables serán creadas. Será llamada una función con un solo argumento, los datos a representar.

stat: La transformación estadística a ser usada en los datos para esta capa, como una cadena de caracteres.

position: Ajuste de posición, una cadena de caracteres o una llamada a la función de ajuste de posición.

...: Otros argumentos pasados a la capa. Normalmente estéticos, por ejemplo colores o tamaños, color = red.º size = 3.

lineend: Estilo de la linea de fin(round, butt, square)

linejoin: Estilo de la linea de unión(round, mitre, bevel)

linemitre:Limite de línea en angulo(número mayor que 1)

arrow: Especificación de una flecha, por ejemplo creada con arrow.

na.rm: Si es FALSE(por defecto),elimina los valores que falten con un mensaje de error. Si es TRUE los elimina sin avisar.

show.legend: Valor lógico, si debe ser incluida esta capa en la leyenda, NA, por defecto, la incluye si se mapea algún elemento estético. FALSE nunca al incluye, TRUE siempre la incluye.

inherit.aes: Si es FALSE, sobrescribe los elementos estéticos por defecto, en vez de combinarse con ellos.

direction: Dirección de los escalones: 'vh' para vertical y luego horizontal, o 'hv' para horizontal y luego vertical.

Paquetes requeridos

Paquete "ggplot2".

Referencias

http://docs.ggplot2.org/current/geom_path.html

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

4.2. Visualización de redes como árboles y mapas

4.2.1. Mapa de calor

Paquete base de R

Si lo tiene.

Función:

heatmap()

Argumentos son:

(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, ...)

x: Matriz numérica con los valores a ser representados.

Rowv: Determina si y como la fila del dendrograma debe ser calculada y reordenada. 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.dedrogram podrá ser aplicado.

reorderfun: Función(d, w) de dedrograma 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.

Paquetes requeridos:

Paquete "stats".

Referencias:

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

4.2.2. Dendrograma

Paquete base de R

No lo tiene

Función:

ggdendrogram()

Argumentos son:

(data, segments, labels, leaf_labels, rotate, theme_dendro, ...)

data: Ya sea un objeto dendro o un objeto que puede ser forzado a dendro clase utilizando la función dendro_data, es decir, los objetos de la clase dendrograma, hclust o tree.

segments: Si es TRUE, muestran segmentos de línea.

labels: Si es verdad, muestra las etiquetas de segmentos.

leaf_labels: Si es verdad, muestra la hoja de etiquetas.

rotate: Si es verdad, gira el diagrama 90 grados.

theme_dendro: Si es verdad, se aplica un tema en blanco para la diagrama (ver theme_dendro)

...: otros parámetros pasados a geom_text

Paquetes requeridos

Paquete "ggdendro"

Referencias

https://cran.r-project.org/web/packages/ggdendro/ggdendro.pdf

https://cran.r-project.org/web/packages/ggdendro/vignettes/ggdendro.html

4.2.3. Árbol radial

Paquete base de R

No lo tiene

Función:

as.radialNetwork()

Argumentos son:

(d, root)

d: Un objeto de la clase R hclust o dendrograma.

root: Un nombre opcional para el nodo raíz. Si falta, utilizar el primer argumento nombre de la variable.

Paquetes requeridos

Paquete "network3D"

Referencias

https://cran.r-project.org/web/packages/networkD3/networkD3.pdf

http://christophergandrud.github.io/networkD3/

4.2.4. Comparación de Dendrogramas

Paquete base de R

No lo tiene.

Función:

tanglegram()

Argumentos son:

(dend1, dend2, which, sort, color_lines, lwd, edge.lwd, columns_width, margin_top, margin_bottom, margin_inner, margin_outer, left_dendo_mar, right_dendo_mar, intersecting, dLeaf, dLeaf_left, dLeaf_right, axes, type, lab.cex, remove_nodePar, main, main_left, main_right, sub, k_labels, k_branches, rank_branches, hang, match_order_by_labels, cex_main, cex_main_left, cex_main_right, cex_sub, highlight_distinct_edges, common_subtrees_color_lines, common_subtrees_color_branches)

dend1: Objeto árbol (dendrogram/dendlist/hclust/phylo), representado a la izquierda.

dend2: Objeto árbol (dendrogram/hclust/phylo), representado a la derecha.

which: Un vector de números de longitud 2, indicando cuales de los árboles especificados en el objeto dendlist debe ser representado.

sort: Lógico(defecto FALSE). Deben ser las etiquetas del dendrograma ordenadas? (Suele dar mejores resultados)

color_lines: Un vector de colores para las lineas que conectan los dendrogramas. Si el vector es menor que el número de elementos a conectar, los colores se reciclan y se avisa. Los colores son aplicados desde abajo hacia arriba.

lwd: Anchura de las lineas que conectan los gráficos. Por defecto 3.5.

edge.lwd: Anchura de las lineas de los dendrogramas.

columns_width: Un vector con tres elementos, especificando el tamaño relativo de las tres gráficas, el árbol izquierdo, las lineas de comparación y el árbol derecho. Por defecto c(5,3,5).

margin_top: El número de lineas de margen a ser especificados arriba de las representaciones.

margin_bottom: El número de lineas de margen a ser especificados debajo de las representaciones.

margin_inner: El número de lineas de margen a ser especificados entre la distancia interior de los árboles y las lineas de conexión.

margin_outer: El número de lineas de margen a ser especificados entre la distancia exterior de los árboles y las lineas de conexión.

left_dendo_mar: Parámetros mar para el árbol izquierdo.

right_dendo_mar: Parámetros mar para el árbol derecho.

intersecting: Lógico (TRUE). Deben ser las hojas de ambos dendrogramas podadas para que se igualen al número de etiquetas?

dLeaf: Un número especificando la distancia en coordenadas entre lapunta de la hoja y su etiqueta. Hay que tener en cuenta que si comparamos dos árboles con distitnas alturas, cambiar a mano este valor puede afectar a cada árbol de diferente manera y puede producir efectos no deseados. Se recomienda cambiar manualmente ambos, dLeaf_left y dLeaf_right.

dLeaf_left: dLeaf del dendrograma izquierdo, por defecto es igual que dLeaf (normalmente negativo).

dLeaf_right: dLeaf del dendrograma derecho, por defecto es igual que dLeaf (normalmente positivo).

axes: Lógico (TRUE). Deben ser los ejes representados?

type: Tipo de gráfico ("t/r- triángulo o rectángulo)

lab.cex: Numérico, el tamaño cex de las etiquetas.

remove_nodePar: Lógico (FALSE). Debe ser el nodePar de las hojas quitado? (útil cuando las hojas tienen demasiados parámetros especificados)

main: Caracter. Título encima de las lineas de conexión.

main_left: Caracter. Título a la izquierda del dendrograma.

main_right: Caracter. Título a la derecha del dendrograma.

sub: Caracter. Título debajo de las lineas de conexión.

k_labels: Número. Número de grupos por los que se deben colorear las lineas.

k_branches: Número. Número de grupos por los que se deben colorear las ramas.

rank_branches: Lógico (FALSE). Deben ser las alturas de las ramas ajustadas?

hang: Lógico (FALSE). Se deben colgar las hojas de los árboles?

match_order_by_labels: Lógico (TRUE). Debe ser el orden de los valores de las hojas combinado entre los dos árboles basándose en las etiquetas?

cex_main: Un valor numérico indicando la cantidad por la cual el título debe magnificarse relativamente al tamaño por defecto.

cex_main_left ver cex_main.

cex_main_right ver cex_main.

cex_sub ver cex_main.

highlight_distinct_edges: Lógico (por defecto TRUE). Si se deben resaltar extremos no coincidentes en cada árbol.

common_subtrees_color_lines: Logical (por defecto es TRUE). Colorea las lineas de conexión basado en los sub árboles en común.

common_subtrees_color_branches: Logical (por defecto es TRUE). Colorea las ramas basado en los sub árboles en común.

faster: Lógico por defecto FALSE. Si es TRUE, anula algunos otros parámetros para hacer la función un poco más rápida.

Paquetes requeridos:

Paquete "dendextend".

Referencias:

http://www.rdocumentation.org/packages/dendextend/functions/tanglegram

https://cran.r-project.org/web/packages/dendextend/dendextend.pdf

4.2.5. Dendrograma en 3D

Paquete base de R

No lo tiene.

Función:

plot.HCPC()

Argumentos son:

(x, axes, choice, draw.tree, ind.names, title, tree.barplot, centers.plot)

x: Un objeto de la clase HCPC.

axes: Los componentes principales a ser representados.

choice: Una cadena de caracteres.Debe ser 3D.map.

ind.names: Un valor lógico. Si TRUE, se mostrarán nombres individuales.

title: El título del gráfico.

tree.barplot: Valor lógico. Si TRUE, el árbol de perdidas de inercia será añadido al gráfico.

centers.plot: Valor lógico. Si TRUE, los centros de los clusters serán dibujados en el mapa.

Paquetes requeridos:

Paquete "FactoMineR".

Referencias:

http://www.sthda.com/english/wiki/hcpc-hierarchical-clustering-on-principal-components-hybrid-approach-2-2-unsupervised-machine-learning

http://www.inside-r.org/packages/cran/FactoMineR/docs/plot.HCPC

4.2.6. Diagrama de transición

Paquete base de R

No lo tiene

Función:

plotmat()

Argumentos son:

(A, pos, curve, name, absent, relsize, lwd, lcol, box.size, box.type, box.prop, box.col, box.lcol, box.lwd, shadow.size, shadow.col, dr, dtext, self.lwd, self.cex, self.shiftx, self.shifty, self.arrpos, arr.lwd, arr.lcol, arr.col, arr.type, arr.pos, arr.length, arr.width, endhead, mx, my, box.cex, txt.col, prefix , cex , cex.txt, add, main, cex.main, segment.from, segment.to, latex, ...)

A: Matriz de coeficiente de segundo grado, especificando los links (filas=a, columnas=desde).

pos: Vector, especificando el número de elementos en cada fila o una matriz de dos columnas con la posición de los elementos, o NULL. Si es una matriz de dos columnas los valores suelen estar entre 0 y 1.

curve: Un valor, o una matriz, de las mismas dimensiones que A, especificando la curvatura de la flecha, 0 para que sea recta, NA para dejarla por defecto.

name: Un vector de cadenas, especificando los nombres de los elementos.

absent: Todos los elementos en A diferentes de este valor, son conectados.

relsize: Factor para escalar el gráfico.

lwd: La anchura por defecto de la flecha y la caja.

lcol: Color por defecto de la flecha y la linea de la caja.

box.size: Tamaño de la caja de la etiquetas.

box.type: La forma de la caja de etiquetas (rect, ellipse, diamond, round, hexa, multi).

box.col: Color con el que rellenar la caja de etiquetas.

box.lcol: Color de la línea de la caja de etiquetas.

box.lwd: Line width of the box, one value or a vector with dimension = number of rows of A.

dr: Tamaño de los segmentos, en radianes, para dibujar la elipse.

dtext: Controla la posición del texto de la flecha, es relativa a la cabeza de la flecha.

self.lwd: Anchura de la linea de una auto flecha, del x a x.

self.cex: Tamaño relativo de la auto flecha.

self.arrpos: Posición de la auto flecha , el ángulo en radianes relativo a la dirección de x.

arr.lwd: Anchura de la flecha que conecta dos valores diferentes.

arr.lcol: Color de la línea de la flecha.

arr.col: Color de la cabeza de la flecha.

arr.type: Tipo de flecha (çurved", "triangle", çircle", .elipse", "T", "simple").

arr.pos: Posición relativa de la flecha.

arr.length: Longitud de la flecha.

arr.width: Anchura de la flecha.

endhead: Si es TRUE:la linea de la flecha se para en la cabeza; por defecto = FALSE la línea continúa después de la cabeza.

mx: Posición horizontal de las cajas.

my: Posición vertical de las cajas.

box.cex: Tamaño relativo del texto en las cajas.

main: Título principal, solo es efectivo si add=False;

cex.main: Tamaño relativo del titulo.

Dónde encontrarlo en R

Paquete Diagram

Referencias

https://cran.r-project.org/web/packages/diagram/diagram.pdf

http://www.inside-r.org/packages/cran/diagram/docs/plotmat

4.2.7. Circuitos electrónicos

Paquete base de R

No lo tiene

Función:

en.Resistor (), en.Capacitator (), en.Transistor (), en.Node(), en.Amplifier(), en.Signal(), en.Ground(),

Argumentos son:

(mid, width, length, lab, pos, dtext, vert, ...)

mid: El punto medio(x,y) de el símbolo.

width: La anchura del símbolo.

length: La longitud del símbolo.

lab: Etiqueta que será añadida al símbolo.

pos: Posición de la etiqueta del símbolo; 1 = abajo; 2 = izquierda; 3 = arriba, 4 = derecha; 1.5 = abajo-izquierda, ...

dtext: Cambio de dirección de x- y/o y- para el texto.

vert: Si TRUE; se alinean verticalmente.

gate: Posición (x,y) de la puerta del en.Transistor.

drain: Posición (x,y)de la toma tierra del en.Transistor.

source: Posición (x,y) de la fuente de en.Transistor.

r: Radio de en.Signal y en.Amplifier

cex: Tamaño del nodo pch (en.Node)

n: Número de líneas horizontales en (en.Ground)

dx: Reducción de tamaño de las líneas horizonales en (en.Ground)

...: Otros argumentos.

Paquetes requeridos

Paquete 'Diagram'.

Referencias

https://cran.r-project.org/web/packages/diagram/diagram.pdf

4.2.8. Gráfico de araña

Paquete base de R

No lo tiene

Función:

radarchart()

Argumentos son:

(df, axistype, seg, pty, pcol, plty, plwd, pdensity, pangle, pfcol, cglty, cglwd, cglcol, axislabcol, title, maxmin, na.itp, centerzero, vlabels, vlcex, caxislabels, calcex, paxislabels, palcex, ...)

df: La trama de datos es usado para dibujar radarchart. Si maxmin es TRUE, esto debe incluir los valores máximos como la fila 1 y los valores mínimos como la fila 2 para cada variable, y los datos reales se debe administrar como la fila 3 y las filas inferiores. El número de columnas (variables) debe ser superior a 2.

axistype: El tipo de ejes, especificado por alguna de 0: 5. 0 significa que no hay etiqueta del eje. 1 significa que sólo etiqueta del eje central. 2 significa única etiqueta alrededor del gráfico. 3 significa tanto etiquetas alrededor del gráfico (periféricos) y el centro. 4 es *. ** Formato de 1, 5 es *. ** Formato de 3. El valor predeterminado es 0.

seg: El número de segmentos para cada eje (por defecto 4).

pty: Un vector para especificar símbolo de punto: por defecto 16 (círculo cerrado), si tu no trazas puntos de datos, debe ser 32. Esto se usa repetidamente para la serie de datos.

pcol: Un vector de códigos de color para los datos del diagrama: Por defecto 1: 8, que se utilizan de forma repetida.

plty: Un vector de tipos de línea para los datos diagrama: Por defecto 1: 6, que se utilizan de forma repetida.

plwd: Un vector de anchos de línea para los datos de diagrama: Por defecto 1, que se utiliza en repetidas ocasiones.

pdensity: Un vector de densidad de relleno de polígonos: NULL por defecto, que se utiliza en repetidas ocasiones.

pangle: Un vector de los ángulos de las líneas utilizadas como relleno polígonos: Por defecto 45, que se utiliza repetidamente.

pfcol: Un vector de códigos de color para el llenado de polígonos: Defecto NA, que es varias veces usd.

cglty: tipo de línea para redes de radar: 3 normal, lo que significa que línea de puntos.

cglwd: Ancho de línea para las redes de radar: Por defecto 1, lo que significa línea más fina.

cglcol: Color de línea para las redes de radar: por defecto "navy"

axislabcol: Color de la etiqueta del eje y números: Default .ªzul"

title: en su caso, el título debe ser escrito.

maxmin: Lógico. Si es TRUE, trama de datos incluye valores máximos posibles como la fila 1 y los posibles valores mínimos, ya la fila 2. Si es FALSE, el valor máximo y mínimo para cada eje se calcularán como máximo real y el mínimo de los datos. Por defecto TRUE.

na.itp: Lógico. Si es TRUE, elementos con los valores de NA se interpolan a partir de su vecino más cercano y las conectan. Si es FALSE , elementos con NA son tratados como el origen (pero no señalado, solamente conectados con las líneas). Por defecto FALSO.

centerzero: Lógico. Si es TRUE, esta función dibuja gráficos con escalamiento originado a partir de (0,0). Si es FALSE, tablas generado por (1 / segmentos). Por defecto FALSO.

vlabels: vector de caracteres para los nombres de las variables. Si es NULL, se utilizan los nombres de las variables como COLNAMES (DF). NULL por defecto.

vlcex: magnificación tamaño de letra para etiquetas. Si es NULL, el tamaño de fuente se fija en el text() por defecto. NULL por defecto.

caxislabels: vector de caracteres para las etiquetas de eje central, sobrescribiendo los valores especificados en la opción de tipo de eje. Si es NULL, se utilizan los valores especificados por el eje opción de tipo. El valor predeterminado es NULL.

calcex: magnificación tamaño de letra para caxilabels. Si es NULL, el tamaño de fuente se fija en el text() por defecto. NULL por defecto.

paxislabels: vector de caracteres para las etiquetas alrededor del gráfico (periféricos), los valores especificados en la opción de tipo de eje sobrescribir. Si es NULL, se utilizan los valores especificados por el eje opción de tipo. El valor predeterminado es NULL.

palcex: magnificación tamaño de letra para paxislabels. Si es NULL, el tamaño de fuente se fija en el text() por defecto. NULL por defecto.

... : Varios argumentos que se dan para plot.default ().

Paquetes requeridos

Paquete "fmsb".

Referencias

http://www.inside-r.org/packages/cran/fmsb/docs/radarchart

4.2.9. Diagrama de Gantt

Paquete base de R

No lo tiene

Función:

mermaid()

Argumentos son:

(diagram, ...)

diagram: Diagrama en formato mermaid o un fichero (como una conexión o nombre de fichero) conteniendo la especificación de un diagrama. Si no se especifica ningún diagrama, entonces se asume que el diagrama vendrá dado por etiquetas y que DiagrammeR esta siendo usado solo para inyección de dependencia.

...: Otros argumentos y parámetros que se pasarán a Javascript

Paquetes requeridos

Paquete 'DiagrammeR'

Referencias

https://cran.r-project.org/web/packages/DiagrammeR/DiagrammeR.pdf

4.3. Visualización de redes en comunidades o áreas

4.3.1. Diagrama de Voronoi

Paquete base de R

No lo tiene

Funciónes:

deldir()
plot()

Preparación de los Datos:
Antes de la visualización debemos obtener los datos que queremos representar en el formato adecuado. Para ello usaremos la función deldir(), esta función calcula la triangulación de un set de puntos sobre un plano según el algoritmo de Lee y Schacter. Para más información visitar las referencias.

Argumentos son:

(data, wlines, add)

data: El objeto de la clase 'tess' a ser representado.

wlines: Debe ser 'tess'.También toma los valores 'triang' y 'both'.

add: Lógico.Permite añadir la representación a un gráfico existente, generalmente la representación de los puntos.

Paquetes requeridos

Paquete 'Deldir'
Paquete 'spatstat'

Referencias

https://cran.r-project.org/web/packages/deldir/deldir.pdf

http://www.inside-r.org/packages/cran/spatstat/docs/plot.tess

http://civilgeeks.com/2011/09/24/poligonos-de-thiessen/

4.3.2. Triangulación de Delaunay

Paquete base de R

No lo tiene

Función es:

deldir()

plot()

Preparación de los Datos:
Antes de la visualización debemos obtener los datos que queremos representar en el formato adecuado. Para ello usaremos la función deldir(), esta función calcula la triangulación de un set de puntos sobre un plano según el algoritmo de Lee y Schacter. Para más información visitar las referencias.

Argumentos son:

(data, wlines, add)

data: El objeto de la clase 'tess' a ser representado.

wlines: Debe ser 'triang'. También toma los valores 'tess' y 'both'.

add: Lógico.Permite añadir la representación a un gráfico existente, generalmente la representación de los puntos.

Paquetes requeridos

Paquete 'Deldir'

Paquete 'spatstat'

Referencias

https://cran.r-project.org/web/packages/deldir/deldir.pdf http://search.r-project.org/library/deldir/html

4.3.3. Redes Neuronales

Paquete base de R

No lo tiene

Función es:

neuralnet()

plot.nn()

Preparación de los Datos:
Antes de la visualización debemos obtener los datos que queremos representar en el formato adecuado.Necesitamos un objeto 'nn' Para ello usaremos la función neuralnet(), esta función es usada para entrenar redes neuronales, permite emplear diversos algoritmos. Para más información visitar las referencias.

Argumentos son:

(x, rep, x.entry, x.out, radius, arrow.length, intercept, intercept.factor, information, information.pos, col.entry.synapse, col.entry, col.hidden, col.hidden.synapse , col.out, col.out.synapse, col.intercept, fontsize, dimension, show.weights, file, ...)

x: Un objeto de clase nn

rep: Repeticiones de la red neuronal. Si rep="best", se representará la repetición con menos errores. Si no se especifica todas las repeticiones serán plasmadas, cada una en una ventana por separado.

x.entry: Coordenada x de la capa del centro. Depende de arrow.length por defecto.

x.out: Coordenada x de la capa de salida.

radius: Radio de las neuronas.

arrow.length: Longitud de las flechas de entrada y salida.

intercept: Valor lógico indicando si pintar la intersección

intercept.factor: Posición x de la intersección. Cuanto más cerca este de 0, más cerca estará la intersección de la neurona izquierda.

information: Un valor lógico indicando si añadir el error y los pasos a la representación.

information.pos: Posición y de la información.

col.entry.synapse: Color de la sinapsis que conduce a las neuronas de entrada.

col.entry: Color de las neuronas de entrada.

col.hidden: Color de las neuronas en la capa oculta.

col.hidden.synapse: Color de las sinapsis con pesos.

col.out: Color de las neuronas de salida.

col.out.synapse: Color de las sinapsis que conducen fuera de las neuronas de salida.

col.intercept: Color de la intersección.

fontsize: Fuente del texto.

dimension: Tamaño de la representación en pulgadas.

show.weights: Valor lógico indicando si mostrar los pesos calculados encima de las sinapsis.

file: Una cadena de caracteres indicando como debe ser nombrada la representación, si no se especifica la representación no será guardada.

... Otros argumentos a ser pasados a otros métodos, como parámetros gráficos.

Paquetes requeridos

Pauqete 'Deldir'

Paquete 'spatstat'

Referencias

http://www.inside-r.org/packages/cran/neuralnet/docs/plot.nn

https://beckmw.wordpress.com/tag/neuralnet/

http://www.inside-r.org/packages/cran/neuralnet/docs/neuralnet

4.3.4. 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 utilicemos.
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.

Paquetes requeridos:

Paquete '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

4.3.5. Nube de palabras

Paquete base de R

No lo tiene

Función:

wordcloud()

Argumentos son:

(words,freq,scale,min.freq,max.words, random.order, random.color, rot.per, colors,ordered.colors,use.r.layout, fixed.asp, ...)

words: Vector conteniendo las palabras.

freq: Vector conteniendo sus frecuencias.

scale: Un vector de longitud 2 indicando el rango de tamaños de las palabras.

min.freq: Las palabras cuya frecuencia este por debajo de este parámetro no serán representadas.

max.words: Número máximo de palabras a ser representadas.

random.order: Representar palabras en orden aleatorio, si es FALSE, serán representadas en orden de frecuencia decreciente.

random.color: Elije los colores aleatorios para las palabras. Si es FALSE, los colores se asignan según las frecuencias.

rot.per: Proporción de las palabras con una rotación de 90 grados.

colors: Colorea las palabras desde la frecuencia más baja hasta la más alta.

ordered.colors: Si es TRUE, entonces los colores son asignados en orden.

use.r.layout: Si es FALSE,entonces código c++ es usado para detectar colisiones, en otro caso se empleará R.

fixed.asp: Si es TRUE, el ratio de exposición se fija. Solo se puede variar el ratio si rot.per==0.

...: Parámetros adicionales a ser pasados a las palabras como strheight,strwidth.

Paquetes requeridos

Paquete 'wordcloud'

Referencias

https://cran.r-project.org/web/packages/wordcloud/wordcloud.pdf

4.3.6. Visualización en clusters

Paquete base de R

No lo tiene

Función:

fviz_cluster()

Argumentos son:

(object, data, stand, geom, repel, show.clust.cent, frame, frame.type, frame.level, frame.alpha, pointsize, labelsize, title, jitter, outlier.color, outlier.shape)

object: Un objeto de la clase "partitionçreado por la dunción pam(), clara() o fanny() en el paquete cluster; "kmeans"[en el paquete stats]; "dbscan"[en el paquete fpc]; "Mclust"[en el paquete mclust]; "hkmeans", .[ec]lust"[en el paquete factoextra]. Otros posibles valores son también listas de objetos con datos y componentes clusters (p.e; object = list(data = mydata, cluster = myclust)).

data: Los datos que han sido usados para clusterizar. Requerido solo cuando el objeto es de clase kmeans o dbscan.

stand: Valor lógico; Si TRUE, los datos son estandarizados antes del análisis principal del componente.

geom: Un texto especificando la geometría a ser usada para el gráfico. Los valores permitidos son la combinación de c("point", "text"). Usar "point"(para mostrar solo puntos); "text"para mostrar solo etiquetas; c("point", "text") para mostrar ambos.

repel: Un booleano, O usar ggrepel para evitar sobreponerse o no.

show.clust.cent: Lógico; si es TRUE; enseña los centros de los clusters.

frame logical value: Si es TRUE, Dibuja una lína exterior a los puntos de cada cluster.

frame.type: Caracterer especificando Character especificando el tipo de marco. Los valores posibles son convex o los tipos soportados por ggplot2.

frame.level: Pasado para el nivel ggplot2::stat_ellipse. Ignorado en convex. El valor por defecto es 0.95.

frame.alpha: Alpha para el marco especificando la transparencia del color de relleno.

pointsize: El tamaño de los puntos.

labelsize: Tipo de fuente para las etiquetas.

title: El título dle gráfico.

jitter: Un parámetro usado para 'estresar' los puntos para reducir el solapamiento. Es una lista conteniendo los objetos, la anchura y la altura.(jitter = list(what, width, height)).

- what: El elemento a ser estresado. Los valores posibles son "point.º "p"; "label.º "l"; "both.º "b".
- width: degree of jitter in x direction
- height: degree of jitter in y direction

outlier.color, outlier.shape: El color y la forma de los outliers. Los outliers son detectados solo por el clustering DBSCAN.

Paquetes requeridos

Paquete 'factoextra'

Referencias

https://cran.r-project.org/web/packages/factoextra/factoextra.pdf

4.3.7. Clusterización en 3D e interactivo

Paquete base de R

No lo tiene.

Función:

plot3d()

Argumentos son:

d(x, y, z, xlab, ylab, zlab, type, col, size, lwd, radius, add, aspect, xlim, ylim, zlim, forceClipregion, ...)

x, y, z: Vector de puntos a ser representados. Cualquier forma razonable de especificarlos es válida, para más información ver la función xyz.cords.

xlab, ylab, zlab: Etiquetas de las coordenadas.

type: Para el método por defecto, un solo carácter indicando el tipo del item a ser representado. Los tipos soportados son: p para puntos, s para esferas, l para lineas, h para segmentos desde z = 0, y n para nada. Para el método mesh3d, shade, wire, o dots, la concordancia parcial es usada.

col: El color a ser usado para representar los items.

size: El tamaño de los puntos representados.

lwd: La anchura de la linea para representar items.

radius: El radio de las esferas.

add: Si se debería añadir los puntos a la gráfica existente.

aspect: O un valor lógico indicando si se deben ajustar los ratios, o un nuevo ratio.

expand: Cuanto se debe expandir la caja alrededor de los datos, si es dibujada.

xlim, ylim, zlim: En plot3d, si no es NULL, establece los límites del recorte de la gráfica. Son usadas para las etiquetas.

forceClipregion: Fuerza a una region recortada a ser usada, se den o no los límites.

...: Parámetros adicionales que serán pasados a par3d, material3d o decorate3d.

Paquetes requeridos:

Paquete rgl".

Referencias:

http://planspace.org/2013/02/03/pca-3d-visualization-and-clustering-in-r/

https://cran.r-project.org/web/packages/rgl/rgl.pdf

4.3.8. Visualización para detectar comunidades

Paquete base de R

No lo tiene.

Función:

walktrap.community()

Argumentos son:

d(graph, weights, steps, merges, modularity, membership)

graph: El gráfico de entrada, las direcciones de los ejes son ignoradas en gráficos dirigidos.

weights: Los pesos de las aristas.

steps: La longitud delos caminos aleatorios a realizar.

merges: Escalar lógico, si se incluye la matriz de unión en el resultado o no.

modularity: Lógico, si incluir el vector de las puntuaciones de modularidad en el resultado o no. Si es TRUE, siempre será calculado.

membership: Lógico, si calcular el vector de membresía para las divisiones correspondiendo al valor modular más alto.

Paquetes requeridos

Paquete ïgraph".

Referencias

http://www.inside-r.org/packages/cran/igraph/docs/walktrap.community

https://cran.r-project.org/web/packages/rgl/rgl.pdf

4.3.9. Diagrama de burbujas

Paquete base de R

No lo tiene.

Función

gvisBubbleChart()

Argumentos son

(data, idvar, xvar, yvar, colorvar, sizevar, options, chartid)

data: Un data frame que será representado como un gráfico de burbujas. Los datos deben tener al menos tres columnas para idvar, xvar, and yvar.

idvar: Nombre de columna de los datos.

xvar: Nombre de columna de un vector numérico de los datos a ser dibujados en el eje x.

yvar: Nombre de columna de un vector numérico de los datos a ser dibujados en el eje y.

colorvar: Nombre de columna de los datos que identifican las burbujas en las miasmas series. Se usa el mismo color para identificar a las burbujas que pertenecen a la misma serie. Las series se configuran usando la opción 'series'.

sizevar: Los valores en esta columna son mapeados a los valores de los píxeles actuales usando la opción sizeAxis.

options: Lista de opciones de configuración. Ver referencias para más información.

chartid: Tipo carácter. Si no se especifica (por defecto) se genera un ID aleatorio basado en el tipo de gráfico.

Paquetes requeridos

Paquete "googleVis".

Referencias

https://developers.google.com/chart/interactive/docs/gallery/bubblechart#Configuration_Options

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

http://www.inside-r.org/packages/cran/googlevis/docs/gvisbubblechart

4.3.10. Diagrama de red de clusters 3D

Paquete base de R

No lo tiene.

Función

pca3d()

Argumentos son

(pca, components, col, title, new, axes.color, bg, radius, group, shape, palette, fancy, biplot, biplot.vars, show.scale, show.labels, labels.col, show.axes, show.axe.titles, show.plane, show.shadows, show.centroids, show.group.labels)

pca Either a prcomp object or a matrix with at least three columns

components Vector of length 3 (pca3d) or 2 (pca2d) containing the components to be shown

col Either a single value or a vector of length equal to number of rows, containing color definitions for the plot points to be shown

title Window title

new Use TRUE to open a new window

axes.color Axis color This option has no effect in pca2d.

bg Background color

palette Specifies the color palette when colors are automatically assigned to the groups. See Details.

fancy set show.labels, show.shadows, show.centroids and show.group.labels to TRUE.

radius Scaling item for the size of points to be shown. In pca2d, this corresponds to the cex parameter.

biplot Specify whether to show a biplot (see section biplots below)

biplot.vars Specify which loading to show on the biplot (see section biplots below)

group either NULL or a factor of length equal to number of rows. Factor levels can be used to automatically generate symbols and colors for the points shown

shape Either a single value or a character vector describing the shapes to be used when drawing data points. Allowed shapes are: sphere, tetrahaedron and cube, and may be abbreviated. In pca2d, the parameter is passed directly on to the pch option of the points() function.

show.labels TRUE for showing labels (taken from the coordinate matrix or the prcomp object). Alternatively, a vector with labels to be shown next to the data points.

labels.col Single value or vector describing the colors of the labels.

show.scale TRUE for showing a numeric scale at the edges of the plot. This option has no effect in pca2d.

show.axes TRUE to show the axes. This option has no effect in pca2d.

show.axe.titles If TRUE, show axe titles (PC 1, PC 2 etc.) This option has no effect in pca2d.

show.plane If TRUE, show a grey horizontal plane at y = 0. This option has no effect in pca2d.

show.shadows If TRUE, show a "lollipoprepresentation of the points on the y = 0 plane: a vertical line joining the data point with the plane and a shadow. In pca2d, for each sample at (x,y), a grey line is drawn from (x,y) to (x,0).

show.centroids If TRUE and the group variable is defined, show cluster centroids (using apropriate group symbols) and lines from each data point to the corresponding centroid.

show.group.labels Either TRUTH/FALSE or a vector equal to the number of unique values in the group parameter. If set, labels for each of the defined group will be shown at the group's centroid. If the value of the parameter is TRUE, then the group names will be taken from the group parameter. Otherwise, the values from this parameter will be used.

... For pca2d, any further argument will be passed on to the points() function.

Paquetes requeridos

Paquete "pca3d".

Referencias

http://www.inside-r.org/packages/cran/pca3d/docs/pca3d

https://cran.r-project.org/web/packages/pca3d/pca3d.pdf

https://cran.r-project.org/web/packages/rgl/rgl.pdf

Capítulo 5

Conclusiones y trabajos futuros

A lo largo de este documento hemos discutido la importancia que esta adquiriendo, desde hace algunos años, el área del Big Data y del Data Science. En la era de la tecnología, de la información y de las redes sociales, no hay limite a la hora de generar y almacenar datos, pero con tales cantidades, sintetizar y extraer información útil se convierte en todo un reto. Existen numerosas aplicaciones y lenguajes que tratan de simplificar esta fase de tratamiento de datos.

En este proyecto nos hemos centrado en el lenguaje y en la herramienta R, en otros capítulos explicábamos las razones las cuales nos llevaron a decantarnos por ello frente a las demás opciones. Entre sus principales características destacan su alta compatibilidad, su escalabilidad, su gran comunidad y su gratuidad. Por esas y muchas otras, pensamos que R tiene mucho potencial y sin duda es una herramienta excelente para el tratamiento estadístico de los datos.

Más concrétamente, este proyecto se ha centrado en la visualización de datos. Se trata de una parte muy importante del Data Science, es una fase que engloba tratamiento y síntesis de datos. Sin duda es una de las maneras más eficaces tanto de percibir como de transmitir información, como es bien sabido 'una imagen vale más que mil palabras'.

Una vez finalizado el proyecto, se plantean varios posibles objetivos o ampliaciones. Primeramente, debido a la cantidad de algoritmos y diferentes tipos de visualización existentes, se podría seguir ampliando este documento incluyendo más funciones y más paquetes. Se podría abarcar también otro tipo de áreas aparte de redes.

También sería interesante crear un nuevo paquete para consolidar todas estas funciones en un único repositorio, de tal manera que quedaría una muy valiosa fuente o herramienta de trabajo, para que los usuarios de R que necesiten realizar visualizaciones de diferentes tipos tengan todo lo necesario en un único paquete y documento.

Capítulo 6

Comunidad R

R cuenta una comunidad global de más de 2 millones de usuarios y desarrolladores que contribuyen voluntariamente con su tiempo y experiencia técnica a mantener, a apoyar y a extender el lenguaje R, su entorno, herramientas e infraestructura.

En el centro de la Comunidad R esta el grupo R Core, cuenta con una veintena de desarrolladores que mantienen R y guían su evolución. La estructura pública oficial de la Comunidad R esta representada por la Fundación R, una organización sin ánimo de lucro que asegura la estabilidad financiera del proyecto-R y mantiene y administra los derechos de autor tanto del software R como de su documentación.

La comunidad R encarna todas las virtudes del movimiento entorno al código abierto. Es una comunidad abierta, dispuesta a acoger a nuevos usuarios y a ayudarles a introducirse en ella, involucrada, y como regla general, la gente es muy activa y siempre predispuesta a dar información y ser de utilidad. La comunidad hace especial hincapié en la transparencia, el intercambio y la creación de software de alta calidad.

Existen cantidad de blogs, páginas web, grupos de usuarios, cuentas en redes sociales, asociaciones, congresos... A continuación indicaremos algunos sitios por los que poder comenzar a adentrarse en la comunidad o simplemente para curiosear o pedir ayuda.

Para estar al tanto de las últimas tendencias, novedades o aportes de otros usuarios, recomendamos tres referencias webs para comenzar.

- R-Bloggers: Una página web que recoge cantidad de noticias y tutoriales, tiene servicio de suscripción a su canal RSS para no perderte nada: http://www.r-bloggers.com/
- inside-R: Otra página web con toda clase de información, diariamente publican la función, el paquete y la pregunta del día: http://www.inside-r.org/

- Twitter: Si se es usuario de las rede sociales, sería interesante estar al tanto del hastag #rstats, con él se publican todas las novedades.
- Revolution Analytics: Otro blog con actualizaciones semanales, suelen incluir interesantes entrevistas a grandes analistas de datos: http://blog.revolutionanalytics.com/

Para pedir ayuda, preguntar o buscar preguntas y respuestas de otros usuarios sobre ciertos temas, existen otro tipo de webs enfocadas a ello específicamente.

- Lista de Emails de R: Se pueden formular preguntas concretas al equipo de R, tratarán de dar respuesta lo antes posible, se debe dejar información de contacto y la propia consulta: https://stat.ethz.ch/mailman/listinfo/r-help
- StackOverflow: Quizá sea la página más famosa de ayuda, tiene secciones para todos los lenguajes de programación, en ella todos los usuarios tienen acceso a tu pregunta y puedes responder: http://stackoverflow.com/tags/r
- StackExchange: Otra página web con similar funcionamiento a la anterior y con sección específica de R: http://stats.stackexchange.com/questions/tagged/r

Para obtener otro tipo de información útil, como próximas conferencias, cursos, eventos, trabajos relacionados, grupos nacionales etc, podemos indagar en las siguientes webs.

- Lista de grupos de usuarios nacionales: En la siguiente web se encuentran recopilados la mayoría de grupos de usuarios a nive nacional: http://blog.revolutionanalytics.com/local-r-groups.html
- Comunidad de usuarios de R España: Asociación Española con sede en Madrid y muy activa: http://r-es.org/quenes-somos/

Capítulo 7

Glosario

A continuación se muestran una selección de palabras recurridas a lo largo del documento con una breve descripción tratando de evitar jerga técnica. Primero se muestran las palabras más generales y a continuación entramos en términos de R.

Algoritmo: Una fórmula matemática que realiza un análisis de un conjunto de datos

Inteligencia artificial: El desarrollo de máquinas inteligentes y software que es capaces de percibir el entorno y tomar las medidas correspondientes cuando sea necesario e incluso aprender de esas acciones.

Big Data Scientist: Alguien que es capaz de desarrollar algoritmos que den sentido a grandes volúmenes de datos.

Inteligencia de negocios (BI): El término general que se utiliza para la identificación, la extracción y el análisis de datos.

Cloud computing: Un sistema informático distribuido a través de una red, que se utiliza para almacenar datos fuera de las instalaciones propias.

Análisis de Clusters: El proceso de identificación de objetos que son similares entre sí y se agrupan con el fin de entender las diferencias, así como las similitudes dentro del conjunto de datos.

Cloud o Nube: Un término general que se refiere a cualquier aplicación o servicio basado en Internet que está alojado de forma remota.

Base de Datos: Una colección digital de datos y la estructura sobre la cual se organizan los datos. Los datos se introducen y se accede a través de un sistema de gestión de base de datos (DBMS).

Data science: Un término reciente que tiene múltiples definiciones, pero generalmente se acepta como una disciplina que incorpora estadística, visualización de datos, programación informática, minería de datos, aprendizaje automático, e ingeniería de base de datos para resolver problemas complejos.

Data scientist: Un practicante del data science.

Data set: Una colección de datos, por lo general en forma de tabla.

Data source o fuente de datos: Cualquier proveedor de datos, por ejemplo, una base de datos o un flujo de datos.

Visualización de Datos: Una abstracción visual de los datos diseñados con el fin de transmitir significado o comunicar información de manera más eficaz.

Machine learning o Inteligencia Artificial: El uso de algoritmos para permitir que una computadora analize datos con el fin de .ªprender"que acción debe tomar cuando se produce un patrón específico o sucede algún evento .

Social Media o Medios sociales: Los social media son instrumentos online de comunicación de masas, que incluyen blogs, microblogs, redes sociales y podcasts.

Área de trabajo: Un área de trabajo temporal en la que suceden todos los cálculos de R. Los datos con los que se trabajan, desaparecerán si no se guardan en el disco duro antes de salir de R.

Argumento: Las opciones que controlan lo que hacen los comandos y las funciones. Confuso porque en R, las funciones hacen lo que, ambos, comandos y funciones hacen en Stata. Más formalmente, son las entradas a una función y que controlan el funcionamiento de dicha función.

Array o Colección de Datos: Una matriz con más de dos dimensiones. Todas las variables deben ser solamente de un tipo (por ejemplo, todas numéricas o todas caracteres). Más formalmente, un vector con un atributo dim. El dim controla el número y el tamaño de las dimensiones.

Atributos: Rasgos de un conjunto de datos como sus nombres de variables y etiquetas.

Clase: Un atributo de una variable o conjunto de datos que utiliza un comando para adaptar sus opciones automáticamente.

Componente: Un elemento en una lista. La longitud de una lista es igual al número de componentes que tenga.

CRAN: La Red Integral de Archivos R en http://cran.r-project. org/. Un archivo en Internet como el de Statistical Software Components(SSC). Consiste en un conjunto de sitios en todo el mundo llamados espejo o mirrors que proporcionan las descarga e instalación de R y de todos sus paquetes.

Cuadro de datos: Un conjunto de datos. Más formalmente, un conjunto de vectores unidos en una lista. Pueden ser de diferentes modos o clases (por ejemplo, carácter y numérico), pero deben tener la misma longitud.

Dim: Una variable cuyos valores son el número de filas y columnas en un conjunto de datos. Se almacena en el conjunto de datos en sí. Además, es un procedimiento que imprime o establece estos valores. Más formalmente, es el atributo que describe las dimensiones de una matriz.

Elemento: Un valor específico para una variable.

Factor: Una variable categórica y los valores de sus etiquetas. Las etiquetas de valor no pueden ser mayores que "1", "2". . . , Si no se asignan de forma explícita.

Función: Un programa de R que se almacena como un objeto.

Etiqueta o 'label': Un procedimiento que crea las etiquetas de las variables. Además, un parámetro que establece el valor de las etiquetas.

Longitud: El número de observaciones/casos en una variable, incluidos los valores que faltan, o el número de variables de un conjunto de datos.

Librería: Se trata de donde una determinada versión de R almacena sus paquetes base y los complementos que ha instalado. También es un procedimiento que carga un paquete de la biblioteca en la memoria de trabajo. Se debe hacer en cada sesión de R antes de usar un paquete.

Lista: Como una colección de conjuntos de datos comprimidos que se puede analizar fácilmente sin descompresión. Más formalmente, un conjunto de objetos de cualquier clase. Puede contener vectores, matrices e incluso otras listas.

Matriz: Un conjunto de datos que debe contener sólo un tipo de variable, por ejemplo, todas numéricas o carácter. Más formalmente, una matriz de dos dimensiones; es decir, un vector con un atributo dim de longitud 2.

Método: Los análisis y/o gráficos que un procedimiento realizará de forma predeterminada, que será diferente para diferentes tipos de variables. La configuración predeterminada para algunos comandos depende de la escala de las variables suministradas por el usuario. P.ej. summary(temperatura) proporciona la temperatura media, summary (género)cuenta machos y hembras. Más formalmente, se define como una función que proporciona el cálculo de una función genérica para una clase específica de objeto.

NA: Un valor que falta. Significa Not Availible (no disponible).

NULL: Un objeto que puede utilizar para eliminar o desechar variables o valores. Más formalmente, NULL tiene una longitud cero y no pertenece a ningún modo en particular. Asignar NULL a un objeto lo elimina.

Objeto: Casi todo en R. Si tiene un modo, es un objeto. Incluye tramas de datos, vectores, matrices, arrays, listas y funciones.

Paquete: Un conjunto archivos relacionados, puede contener funciones y los ficheros de ayuda. Puede venir con R o puede ser creado por sus usuarios.

Vector: Una variable. Puede existir por sí mismo en memoria o puede ser parte de un conjunto de datos.

Bibliografía

[1] Fernando Giner de la Fuente, María de los Ángeles Gil Estallo, *Los sistemas de información en la sociedad del conocimiento*, ESIC Editorial, 2004.

[2] Alejandro Hernandez trasobares, *LOS SISTEMAS DE INFORMACIóN: EVOLUCIóN Y DESARROLLO* , Universidad de Zaragoza.

[3] A.Ohri, *R for Business Analytics*, Springer, 1-2.

[4] Andrie de Vries, Joris Meys, *R for Dummies*, Making Everything Easier!, 31-33, 248-250.

[5] Manas A. Pathak, *Beginning Data Science with R*, Springen International Publishing Switzerland 2014, 31-32.

[6] Richard A. Becker, Stephen G. Eick, Allan R. Wilks, *VISUALIZING NETWORK DATA.*, Morgan Kauffman Publisher,1-3.

[7] Stuart K. Card, Jock D. Mackinlay, Ben Shneiderman, *Readings in Information Visualization: Using Vision to Think*, Morgan Kauffman Publisher 186-188.

[8] Emmanuel Paradis, *R para Principiantes*, Institut des Sciences de l'Evolution, 5-6.

[9] Marina L. Gavrilova, *Generalized Voronoi Diagram: A Geometry-Based Approach to Computational Intelligence*, Springer.