

La Estadística en la Minería de Datos

Parte I

Víctor Guevara Ponce

Universidad Nacional Mayor de San Marcos

Marzo 2019

AGENDA

1 Introducción

- El mundo y los datos

2 Introducción a la minería de datos

- Disciplinas para análisis de datos
- Descubrimiento del Conocimiento en bases de datos
- La metodología CRISP-DM
- Tareas de minería de datos
- Análisis de datos

3 Análisis exploratorio de datos

- Introducción análisis exploratorio de datos

4 Introducción a R y R-Studio

- El entorno de R
- Instalación de R
- Instalación de R-Studio
- Navegando en R

El mundo de los datos...

- Gran cantidad de datos es coleccionada y almacenada.
- Las computadoras se han vuelto más baratas y poderosas
- La presión de la competencia es fuerte
- El 90 % de los datos del mundo se crearon en los últimos años.



Web data, e-commerce



Cyber Security



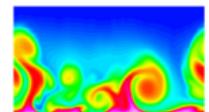
Traffic Patterns



Social Networking: Twitter



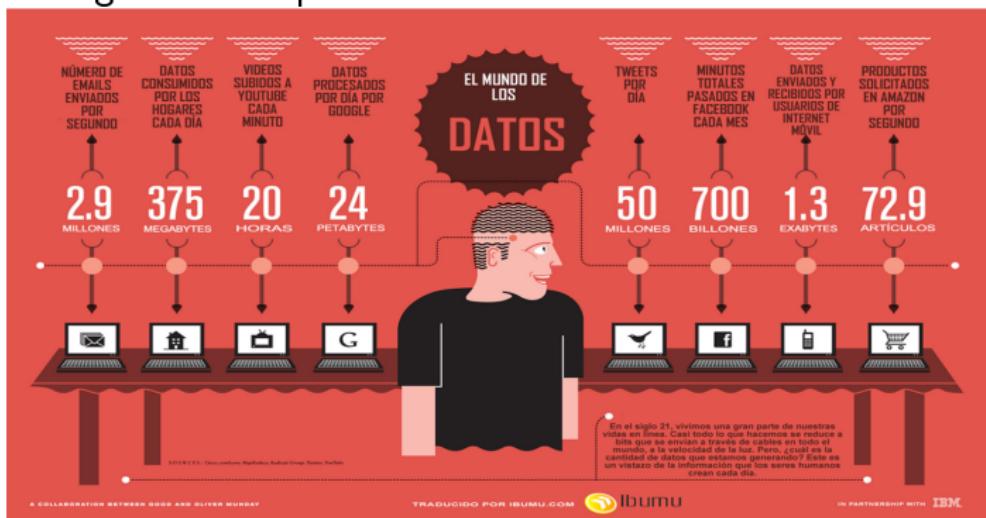
Sensor Networks



Computational Simulations

El mundo de los datos...

Ha habido un enorme crecimiento de datos en bases de datos comerciales y científicas debido a los avances en la generación de datos y tecnologías de recopilación



El mundo de los datos...

Explosión de Datos...

Las Herramientas y nuevas tecnologías automatizadas de almacenamiento de datos conllevan a que grandes cantidades de datos sean almacenados en bases de datos, data warehouses y otros repositorios de información.

¡Nos estamos ahogando en datos, pero estamos hambrientos de conocimiento!

El mundo de los datos...

- **Bases de Datos a ser extraídas**

Relacionales, transaccionales, orientada a objetos, espacial, series de tiempo, texto, multimedia, WWW, etc.

- **Conocimiento a ser descubierto**

Discriminación, asociación, clasificación, agrupamiento, análisis de tendencia, etc.

- **Técnicas Utilizadas**

Orientadas a datos, data warehouse (OLAP), machine learning, estadística, minería de datos visualización, redes neuronales, etc.

- **Aplicaciones Adaptadas**

Ventas, telecomunicaciones, banca, análisis de fraude, análisis de mercados, Web mining, etc.

El mundo de los datos...

¿Dónde están los datos?



kaggle.com



Banco Mundial

KDnuggets

El mundo de los datos...

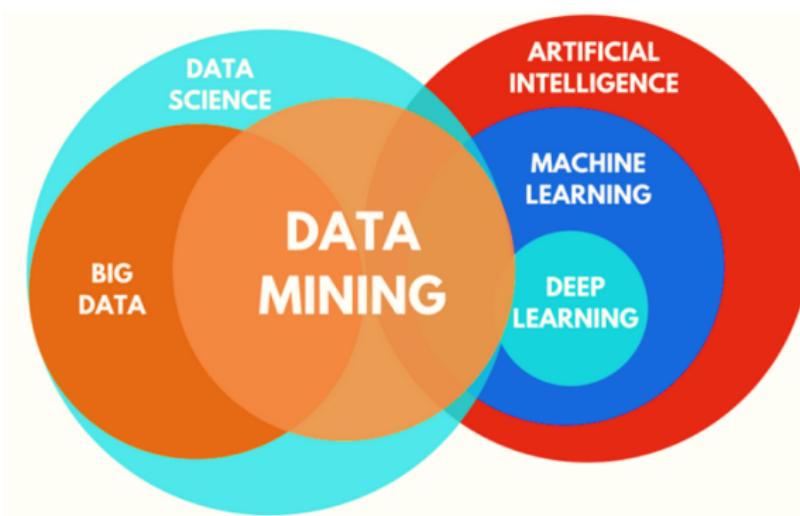
Las técnicas tradicionales pueden ser inadecuadas debido a los datos

- Los rápidos avances en la recolección de datos y la tecnología de almacenamiento han permitido a las organizaciones acumular grandes cantidades de datos.
- Sin embargo, extraer información útil ha resultado ser extremadamente desafiante.
- A menudo, las herramientas y técnicas de análisis de datos tradicionales no se pueden usar debido al tamaño masivo de un conjunto de datos.



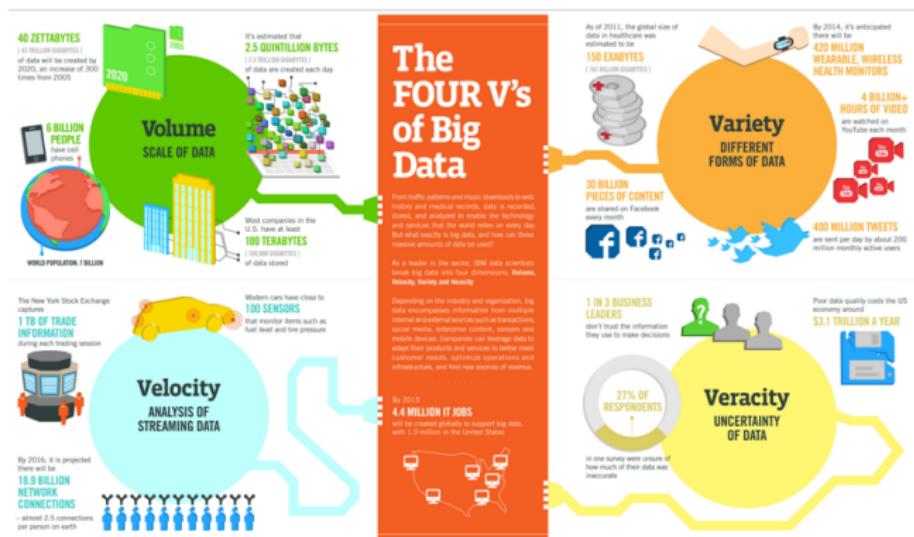
Introducción a la minería de datos

Disciplinas relacionadas al análisis de datos



Introducción a la minería de datos

BIG DATA

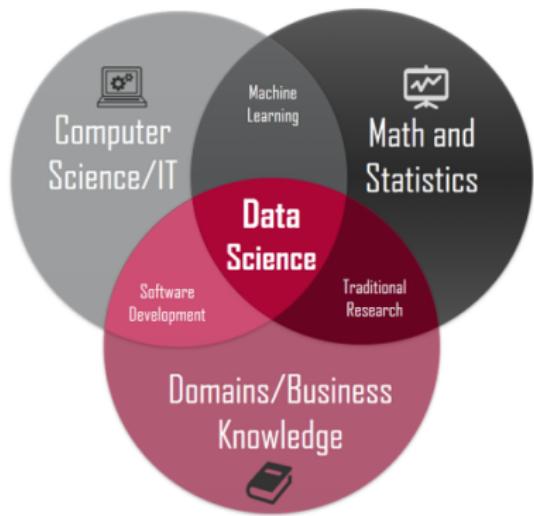


Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, NISTEC, Gartner

Introducción a la minería de datos

DATA SCIENCE

- La ciencia de datos es una ciencia, una disciplina rigurosa que combina elementos de estadística y ciencias de la computación, con raíces en las matemáticas.
- La ciencia de datos se aplica mejor en el contexto del conocimiento experto sobre el dominio del cual se originan los datos.



Introducción a la minería de datos

Descubrimiento del Conocimiento en bases de datos (KDD)

- Es un proceso automático en el que se combinan descubrimiento y análisis.
- El proceso consiste en extraer patrones en forma de reglas o funciones, a partir de los datos, para que el usuario los analice.
- Para ese fin se deben seguir un conjunto de procesos claramente establecidos

Introducción a la minería de datos

Proceso de extracción de conocimiento KDD

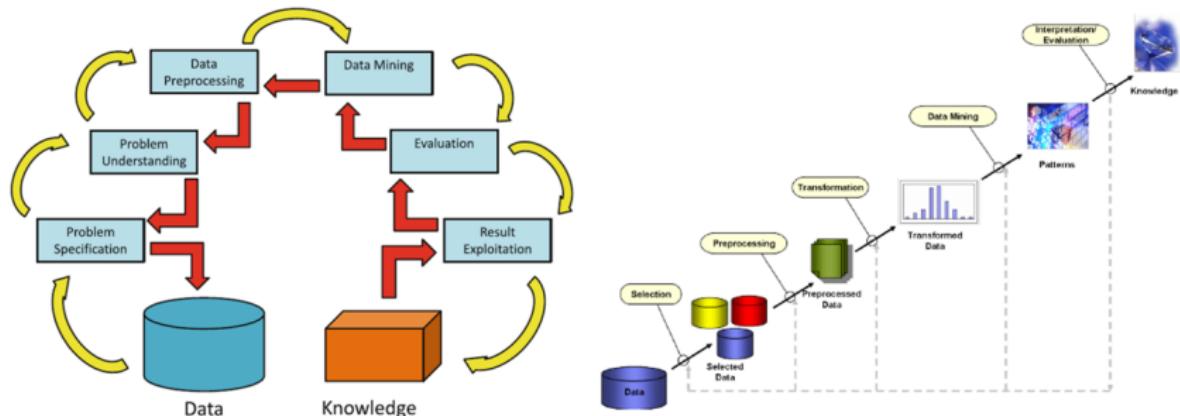


Fig. 1.1 KDD process

Resumen del proceso KDD

Introducción a la minería de datos

La metodología CRISP-DM

CRISP-DM (del inglés Cross Industry Standard Process for Data Mining). se trata de un modelo estándar abierto del proceso que describe los enfoques comunes que se utiliza para un proyecto de análisis de datos



Introducción a la minería de datos

Comprendión del negocio (Business Understanding)

- El primer paso del proyecto es entender, desde una perspectiva de negocio, lo que el cliente quiere lograr.
- Identificar claramente el área del problema a solucionar (por ejemplo, control de fabricación, CRM, desarrollo de negocio...).
- Especificar todas las preguntas de negocio y cualquier otra exigencia tan precisamente como sea posible.
- Describir las salidas que se pretende conseguir en el proyecto que van a permitir el logro de los objetivos de negocio

Introducción a la minería de datos

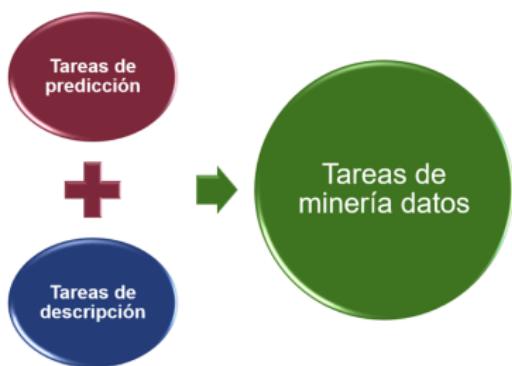
Definición de la variable objetivo (Target)

- Es el indicador que se desea predecir y cuyo valor calculado de forma anticipada permitirá optimizar las estrategias del negocio. También se conoce como Variable de respuesta.

Minería de datos para comprender diversas tareas en el negocio

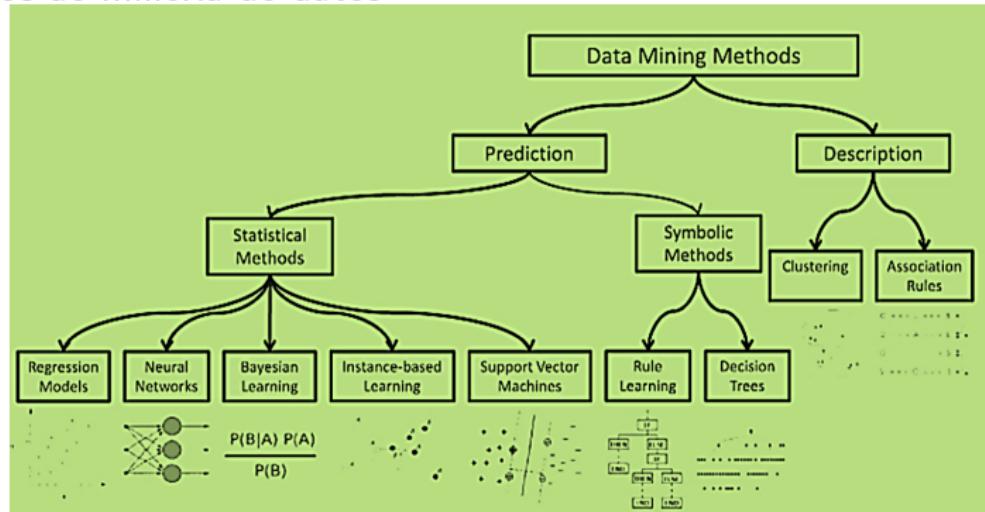
El objetivo es predecir el valor de un atributo particular (objetivo o variable dependiente) en base a los valores de los otros atributos (Variables independientes o explicativas).

El objetivo es detectar patrones (correlaciones, tendencias, grupos, trayectorias y anomalías), también llamadas tareas descriptivas que resumen las relaciones en los datos



Introducción a la minería de datos

Métodos de minería de datos



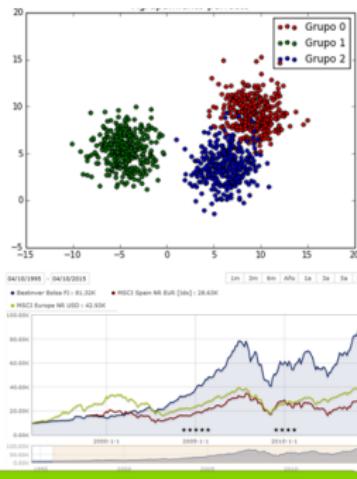
Introducción a la minería de datos

Tareas de predicción



- Se utiliza para variables objetivo discretas.
- Ejem: Si una transacción es fraudulenta o no.

- Se utiliza para variables objetivo continuas
- Ejem: Pronosticas el precio futuro de una acción.



El objetivo de ambas tareas es realizar un modelo que minimice el error entre los valores pronosticados y el verdadero valor de la variable objetivo

Entendimiento de la data

- Recolección de datos
 - Enumerar los conjuntos de datos adquiridos (lugares, los métodos utilizados para la adquisición, problemas encontrados y las soluciones alcanzadas).
- Descripción de los datos
 - Verificar el volumen de los datos y examinar sus propiedades.
 - Accesibilidad y disponibilidad de los atributos. Tipos de atributos, rango, correlaciones, identidades.
 - Comprender el significado de cada atributo y de los valores de los atributos en términos del negocio.
 - Para cada atributo, calcular estadísticos básicos (ej. promedio, desviación estándar, varianza, moda, sesgo, etc.)
- Exploración de datos
 - Analizar en detalle las propiedades de los atributos de interés.

Entendimiento de la data

- Verificar la calidad de los datos
 - Identificar valores especiales y catalogar su significado.
 - ¿Se cuenta con todos los casos requeridos? ¿Estos contienen errores y que tan comunes son?
 - Identificar atributos perdidos y en blanco. Significado de datos perdidos.
 - El significado de los atributos y los valores que contienen encajan?
 - Verificar la escritura de los valores (ej. un mismo valor pero a veces empieza con letras mayúsculas, otras con letra minúscula)
 - Verificar la factibilidad de los valores

Entendimiento de la data

- Selección de datos

- Reconsiderar el criterio de selección de los datos.
- Decidir el conjunto de datos que será usado.
- Recolectar data adicional que sea apropiada.
- Considerar el uso de técnicas de muestreo.
- Explicar por qué ciertos datos son incluidos o excluidos.

- Limpieza de datos

- Corregir, remover o ignorar ruido.
- Decidir como proceder con valores especiales y su significado (99 para estado civil)
- Niveles de totalización, valores perdidos, etc.
- outliers?

- Construcción de Datos

- ¿Los datos perdidos pueden imputarse o reconstruirse?
- Conocimiento previo.

Entendimiento de la data

● Formato de Datos

- Reordenamiento de los atributos (Algunas herramientas tienen requerimientos en relación al orden de los atributos, ej. el primer campo debe ser un identificador único para cada registro o el último campo debe ser la variable respuesta a ser predicha).
- Reordenamiento de registros (Puede ser que la herramienta de modelamiento requiera que los registros estén ordenados de acuerdo al valor de la variable respuesta)
- Reformateo de valores (Cambios puramente sintácticos para satisfacer los requerimientos de una herramienta específica de modelamiento, ej. NA para datos perdidos en vez de 99, remover caracteres ilegales, letras mayúsculas o minúsculas, etc.)

Entendimiento de la data

- Hay diferentes tipos de atributos.

Nominal

Ejemplos: números de identificación, color de ojos, códigos postales

Ordinal

Ejemplos: clasificaciones (por ejemplo, sabor de papas fritas en una escala del 1 al 10), calificaciones, altura alto, medio, corto

Intervalo

Ejemplos: fechas del calendario, temperatura.

Ratio

Ejemplos: Cantidad de personas, longitud, tiempo, cuentas

Datos y tipos de datos

- El tipo de un atributo depende de cuál de las siguientes propiedades/ operaciones posee:
Distinción: = o diferente
Orden: <>
Las diferencias son + - significativas:
Ratio son * / significativas
Atributo nominal: distinción
Atributo ordinal: distinción y orden
Atributo de intervalo: distinción, orden y diferencias significativas
Atributo de relación: las 4 propiedades / operaciones

Datos y tipos de datos

Operaciones según tipo de datos

Tipos de atributos		Descriptivos	Ejemplo	Operaciones
Categórico (Cualitativo)	Nominal	Los valores de un atributo nominal son solo nombres diferentes; es decir, los valores nominales solo proporcionan información suficiente para distinguir un objeto de otro. (=, ≠)	códigos postales, números de identificación de los empleados, color de ojos, género	moda, entropía, contingencia, correlación, prueba χ^2
	Ordinal	Los valores de un atributo ordinal proporcionan suficiente información para ordenar objetos. (<, >)	dureza de los minerales, {bueno, regular, bajo}, calificaciones, números de calles	mediana, percentiles, correlación de rangos, ejecutar pruebas, pruebas de signos
Numérico (Cuantitativo)	Intervalo	Para los atributos de intervalo, las diferencias entre los valores son significativas, es decir, existe una unidad de medida. (+, -)	Fechas calendario, temperatura en grados Celsius o Fahrenheit.	Media, desviación estándar, Correlación de Pearson, pruebas t y F
	Ratio	Para las variables de relación, tanto las diferencias como las proporciones son significativas. (*, /)	Temperatura en grados Kelvin, cantidades monetarias, conteos, edad, masa, longitud, corriente eléctrica.	significado geométrico, Significado armónico, variación porcentual

Análisis Exploratorio de Datos

AED

- La finalidad del AED es examinar a detalle los datos previamente a la aplicación de cualquier técnica o algoritmo.
- De esta forma el analista consigue un entendimiento básico de sus datos y de las relaciones existentes entre las variables analizadas.
- El examen previo de los datos es un paso necesario, que lleva tiempo, y que habitualmente se descuida por parte de los analistas de datos.

Análisis Exploratorio de Datos

AED

- El Análisis Exploratorio de Datos es un conjunto de técnicas estadísticas cuya finalidad es conseguir un entendimiento básico de los datos y de las relaciones existentes entre las variables analizadas.
- Para conseguir este objetivo el A.E.D. proporciona métodos sistemáticos sencillos para organizar y preparar los datos, detectar fallas en el diseño y recogida de los mismos, tratamiento y evaluación de datos ausentes (missing), identificación de casos atípicos (outliers) y comprobación de algunos supuestos.

Análisis Exploratorio de Datos

AED

- Entendimiento del contexto del problema que uno pretende resolver.
- Preparación y Valor Agregado de los Datos.
- Examen gráfico para evaluar la naturaleza de las variables individuales y un análisis descriptivo numérico que permita cuantificar algunos aspectos de los datos.
- Examen gráfico de las asociaciones entre las variables analizadas.
- Identificar los posibles casos atípicos (outliers) y evaluar el impacto potencial que puedan ejercer en análisis estadísticos posteriores.
- Evaluar, si fuera necesario, el impacto potencial que pueden tener los datos ausentes (missing) sobre la representatividad de los datos analizados.

Análisis Exploratorio de Datos

AED univariado

- Algunos de los gráficos considerados en el AED son:
- Gráficos de histogramas
- Grafica de Intervalos.
- Diagrama de Tallos y hojas.
- Diagrama de Cajas.
- Diagrama de Densidad de Kernel.
- Diagrama de Violín.
- Gráfica de Probabilidad.

Análisis Exploratorio de Datos

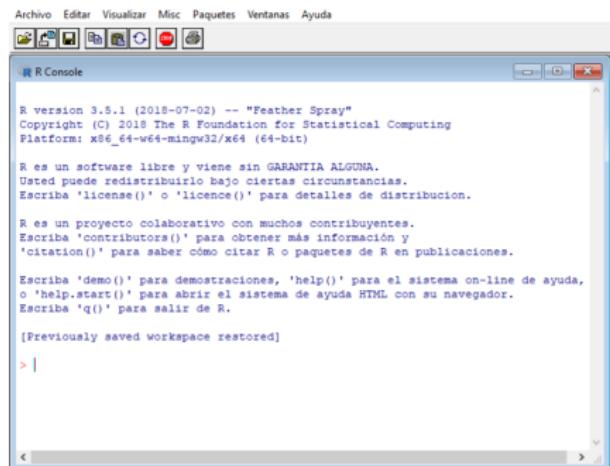
AED multivariante

Algunos de los gráficos considerados en el AED son:

- Diagrama de dispersión 2D.
- Matriz de dispersión .
- Diagrama de dispersión 3D.
- Diagrama marginal.
- La caras de Chernoff.
- Gráfico de Estrellas.
- Gráfico de coordenadas paralelas.

Introducción a R

¿Qué es R?



¿Qué es R?

- R es un lenguaje computacional de alto nivel y un programa para realizar análisis estadístico y gráficos.
- Es una solución de código abierto para el análisis de datos que es compatible con una comunidad de investigación mundial grande y activa
- R es una plataforma poderosa para el análisis y exploración de datos interactivos.
- R se ejecuta en una amplia gama de plataformas, incluidas Windows, Unix y Mac
- El sitio web oficial de R es: <http://www.R-project.org>

Instalación de R

Ingresar a <https://cloud.r-project.org/>

Seleccionar el instalador según el sistema operativo

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages. **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for Mac OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2018-12-20, Eggshell Igloo) [R-3.5.2.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

Instalación de R-Studio

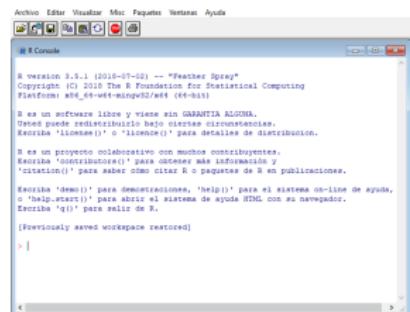
Ingresar a <https://www.rstudio.com/products/rstudio/download/>
Seleccionar el instalador según el sistema operativo

The screenshot shows the RStudio download page. At the top, there's a navigation bar with links for Products, Recursos, Precios, Sobre nosotros, Blogs, and a search icon. Below the navigation, a message says "RStudio requiere R 3.0.1+. Si aún no tienes R, descárgalo aquí." Another message below it states, "Es posible que los usuarios de Linux deban importar la clave de firma de código pública de RStudio antes de la instalación, dependiendo de la política de seguridad del sistema operativo." A section titled "Instaladores para plataformas soportadas" lists various RStudio versions for different operating systems. One row is highlighted with a red border: "RStudio 1.1.463 - Windows Vista / 7/8/10" with a file size of 85.8 MB, a date of 2018-10-29, and a MD5 hash of 58b3d796d8cf96fb8580c62f46ab64d4.

Instaladores	tamaño	Fecha	MD5
RStudio 1.1.463 - Windows Vista / 7/8/10	85.8 MB	2018-10-29	58b3d796d8cf96fb8580c62f46ab64d4
RStudio 1.1.463 - Mac OS X 10.6+ (64 bits)	74.5 MB	2018-10-29	a79032ba4d7daaa8ea0da01948278bd94
RStudio 1.1.463 - Ubuntu 12.04-15.10 / Debian 8 (32 bits)	89.3 MB	2018-10-29	8a6755fa9fae2bafce289fd3358aaef63
RStudio 1.1.463 - Ubuntu 12.04-15.10 / Debian 8 (64 bits)	97.4 MB	2018-10-29	bc50d6bd34926c1cc3ae4a209d67d649
RStudio 1.1.463 - Ubuntu 16.04 / Debian 9+ (64 bits)	65 MB	2018-10-29	cfd659db18619cc78d1592fefa7c753
RStudio 1.1.463 - Fedora 19+ / RedHat 7+ / openSUSE 13.1+ (32 bits)	88.1 MB	2018-10-29	742f0bad60fffea3281576e14ad6699e
RStudio 1.1.463 - Fedora 19+ / RedHat 7+ / openSUSE 13.1+ (64 bits)	90.6 MB	2018-10-29	c7303067a0ca99deeae427b856952d1

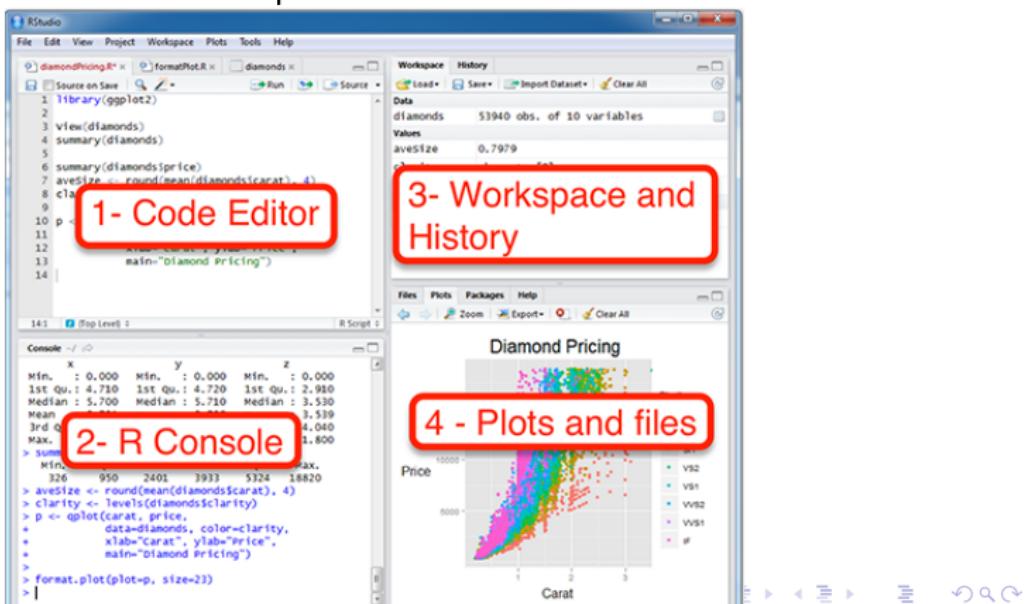
Consola de R

- En la consola del R es donde se realizan los cálculos.
- Cuando una expresión se introduce en la consola, es posteriormente evaluada. Dependiendo de la expresión, el sistema puede responder mediante la salida de resultados a la consola o la creación de un gráfico en una ventana nueva. Luego otra expresión es ingresada y evaluada.



Consola de R

- RStudio es un conjunto de herramientas integradas diseñadas para ayudarlo a ser más productivo con R.



Operaciones básicas en R

R como calculadora

```
> 2+3  
[1] 5  
> 5/9  
[1] 0.5555556  
> log(5)  
[1] 1.609438  
> pi  
[1] 3.141593  
> exp(2)  
[1] 7.389056  
> sin(pi)  
[1] 1.224606e-16
```

Operaciones básicas en R

- Operadores en R

Binary Operators	+	-	*	/	^	%%
Math Functions	abs	sqrt	log	exp	log10	factorial
Trig Functions	sin	cos	tan	asin	acos	atan
Rounding	round	ceiling	floor	trunc	signif	zapsmall
Math Quantities	Inf	-Inf	NaN	pi	exp(1)	ii
%% is for modular arithmetic						

Aritméticos	Comparativos	Lógicos
+ Suma	== igualdad	& Y lógico
- Resta	!= Diferente de	! No lógico
* Multiplicación	< Menor que	O lógico
/ División	> Mayor que	all
^ Potencia	<= Menor o igual	any
%% residuo	>= Mayor o igual	
	%in% igual grupos	
	%like%	
	igualdad no exacta	

Objetos y funciones

- Para comentar se utiliza el signo de numeral o hash-tag # (# es ignorado por el intérprete de R, esto es útil para documentar el código en lenguaje natural)
- Lo que normalmente hacemos es crear objetos y aplicar funciones a los objetos (las funciones también se consideran objetos).
- En la práctica, casi todas las expresiones compuestas se dividen con valores intermedios asignados a variables que, cuando se usan en expresiones futuras, son como sustituir la variable (Objeto) con el valor que se le asignó
- las asignaciones siguen el formulario VARIABLE <- VALOR (También se puede utilizar =)

```
> variable <- 10
```

Objetos y funciones

Operaciones con objetos

```
> objeto1 <- 10
> objeto1
[1] 10
> objeto2 <- 20
> objeto2
[1] 20
> objeto1 + objeto2
[1] 30
> objeto1*objeto2
[1] 200
> objeto1^2
[1] 100
```

Objetos y funciones

Funciones

- Una función, es un conjunto de instrucciones que se desarrollan para evitar repetir una misma tarea o reducir la complejidad.
- Es construida con el fin de llevar a cabo una tarea específica.
- R cuenta con múltiples funciones
- Cada función tiene un conjunto formal de argumentos con algunos valores por defecto. Estos pueden ser encontrados en la documentación de la función.
- El llamado a una función puede incluir cualquier subconjunto de la lista completa de argumentos.
- Para especificar un argumento en particular usar el nombre del argumento.

Objetos y funciones

Funciones

- R distingue MAYÚSCULAS y minuscúlas.
- Los argumentos no tienen que ser nombrados si están inscritos en el mismo orden que la lista de argumentos formales de la función. Sin embargo, para que su código sea más fácil de entender por lo general es una buena idea nombrar a sus argumentos.

Sintaxis de una función

```
function.nombre <- function(argumentos)
{
    cálculos sobre los argumentos
    algun otro código
}
```

Datos y tipos de datos

la función `class` devuelve el tipo de dato para R

Datos numéricos

```
> class(10)
[1] "numeric"
> class(1.5)
[1] "numeric"
> class(5L)
[1] "integer"
> objeto1<- 10
> class(objeto1)
[1] "numeric"
```

Datos no numéricos

```
> class('análisis de datos')
[1] "character"
> obj<- "Curso"
> class(obj)
[1] "character"
```

Clases y tipos de datos

Valores lógicos

```
> class(TRUE)  
[1] "logical"  
> class(NA)  
[1] "logical"  
> T  
[1] TRUE  
> 4 < 2  
[1] FALSE
```

Cambiando la clase de un dato

```
> obj1<-10  
> class(obj1)  
[1] "numeric"  
> obj1<-as.character(obj1)  
> class(obj1)  
[1] "character"
```

Coerción

- Las funciones muestran error cuando el tipo de dato que esperan no coincide con los que colocamos en los argumentos.
- Cuando llama a una función con un argumento del tipo incorrecto, R intentará forzar valores a un tipo diferente para que la función funcione.
- En R, los datos pueden ser coercionados, es decir, forzados, para transformarlos de un tipo a otro.
- En ocasiones el resultado puede ser error, esto ocurre porque no todos los tipos de datos pueden ser transformados a los demás, para ello se sigue una regla general.

Coerción

Casos de datos que permiten coerción

Tipo	Comprobación	Coerción
array	is.array()	as.array()
character	is.character()	as.character()
complex	is.complex()	as.complex()
double	is.double()	as.double()
factor	is.factor()	as.factor()
integer	is.integer()	as.integer()
list	is.list()	as.list()
logical	is.logical()	as.logical()
matrix	is.matrix()	as.matrix()
NA	is.na()	-
NaN	is.nan()	-
NULL	is.null()	as.null()
numeric	is.numeric()	as.numeric()
ts	is.ts()	as.ts()
vector	is.vector()	as.vector()

Vectores

- El vector es el objeto más básico en R, tiene como restricción el solo poseer una dimensión, y solo puede almacenar un tipo de dato, tales como: integer, boolean, numeric, character, complex.
- La función `c(...)` concatena argumentos para formar un vector.

```
> vec<-10
> is.vector(vec)
[1] TRUE
> #Creamos un vector con varios datos
> vec1<-c(2,4,6,7,10,12)
> vec1
[1]  2  4  6  7 10 12
> vec2<-c("a","b","c","d","e")
> vec2
```

Vectores

- Para crear un vector con un patrón determinado:
 - : Secuencia de enteros.
 - seq() Secuencia en general.
 - rep() Vector de elementos repetidos.

```
> vec4<-c(3:8)
> vec4
[1] 3 4 5 6 7 8
> seq(0, 4, by=0.5)
[1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
> seq(0, 6, len=6)
[1] 0.0 1.2 2.4 3.6 4.8 6.0
> rep(1:8, each=2)
> rep(1:6, times=3)
```

Vectores

Usar [] con un vector/escalar de posiciones para acceder a los elementos de un vector.

Acceder a los elementos de un vector

```
> vec1<-c(2,4,6,7,10,12)  
> vec1  
[1] 2 4 6 7 10 12  
> vec1[3]  
[1] 6  
> vec1[2:4]  
[1] 4 6 7  
> vec1[c(2,4,6)]  
[1] 4 7 12  
> vec1[vec1>6]
```

eliminar un elemento de un vector

```
> class('análisis de datos')  
  
#se debe incluir el signo negativo  
> vec1[-5]  
[1] 2 4 6 7 12  
#Reemplazar un elemento  
> vec1[2]<-100  
> vec1  
[1] 2 100 6 7 10 12  
\
```

Operaciones con Vectores

Cuando los vectores son usados en expresiones matemáticas, las operaciones son realizadas con cada elemento.

```
> #Operaciones con vectores
> vec6<-c(1,3,5,7,9)
> vec6
[1] 1 3 5 7 9
> vec6*vec6
[1] 1 9 25 49 81
> vec6^2
[1] 1 9 25 49 81
> sqrt(vec6)
[1] 1.000000 1.732051 2.236068 2.645751 3.000000
> sum(vec6)
[1] 25
```

Funciones con Vectores

Las funciones por defecto más utilizadas en R son:

<code>sum(x)</code>	<code>prod(x)</code>	Sum/product of the elements of x
<code>cumsum(x)</code>	<code>cumprod(x)</code>	Cumulative sum/product of the elements of x
<code>min(x)</code>	<code>max(x)</code>	Minimum/Maximum element of x
<code>mean(x)</code>	<code>median(x)</code>	Mean/median of x
<code>var(x)</code>	<code>sd(x)</code>	Variance/standard deviation of x
<code>cov(x,y)</code>	<code>cor(x,y)</code>	Covariance/correlation of x and y
<code>range(x)</code>		Range of x
<code>quantile(x)</code>		Quantiles of x for the given probabilities
<code>fivenum(x)</code>		Five number summary of x
<code>length(x)</code>		Number of elements in x
<code>unique(x)</code>		Unique elements of x
<code>rev(x)</code>		Reverse the elements of x
<code>sort(x)</code>		Sort the elements of x
<code>which()</code>		Indices of TRUEs in a logical vector
<code>which.max(x)</code>	<code>which.min(x)</code>	Index of the max/min element of x
<code>match()</code>		First position of an element in a vector
<code>union(x, y)</code>		Union of x and y
<code>intersect(x, y)</code>		Intersection of x and y
<code>setdiff(x, y)</code>		Elements of x that are not in y
<code>setequal(x, y)</code>		Do x and y contain the same elements?

Matrices y arrays

- Las matrices y arrays pueden ser descritas como vectores multidimensionales. Al igual que un vector, únicamente pueden contener datos de un sólo tipo.
- Una matriz es una generalización, en dos dimensiones, de un vector.
- Los arrays, pueden tener un número arbitrario de dimensiones. Pueden ser cubos, hipercubos y otras formas.
- las matrices son una caso especial de un array, que se distingue por tener específicamente dos dimensiones, un “largo” y un “alto”.
- Las matrices son, por lo tanto, una estructura con forma rectangular, con renglones y columnas.

Matrices

- Para crear una matriz:

```
matrix(data=NA, nrow=1, ncol=1, byrow = FALSE,  
dimnames = NULL)
```

data: Un vector con los datos para llenar la matriz; si los datos no contienen elementos suficientes para llenar la matriz, entonces estos son reciclados.

nrow: Número deseado de filas.

ncol: Número deseado de columnas.

byrow: Si es FALSE (defecto) la matriz es llenada por columnas, caso contrario por filas.

dimnames: (opcional) lista de longitud 2 dando los nombres de filas y columnas respectivamente, los nombres de la lista serán usados como nombres de las dimensiones.

Matrices

```
> matriz1 <- matrix(c(1,2,3,4,5,6,7,8,9,10,11,12),  
nrow=4, ncol=3)  
> matriz1  
 [,1] [,2] [,3]  
[1,] 1 5 9  
[2,] 2 6 10  
[3,] 3 7 11  
[4,] 4 8 12  
> matriz1 <- matrix(c(1,2,3,4,5,6,7,8,9,10,11,12),  
nrow=4, ncol=3, byrow = T)  
> colnames(matriz1)<-c("A","B","C")  
> rownames(matriz1)<-c("F1","F2","F3","F4")
```

Operaciones con Matrices

- Cuando se usan matrices en expresiones matemáticas se realizan las operaciones elemento por elemento.
- Para multiplicación matricial usar el operador `%*%`.

```
> A <- matrix(1:4, nrow=2)
> B <- matrix(1, nrow=2, ncol=2)
> A*B
```

Funciones con Matrices

Las funciones por defecto más utilizadas en R son:

<code>t(A)</code>	Transpose of A
<code>det(A)</code>	Determinate of A
<code>solve(A, b)</code>	Solves the equation $Ax=b$ for x
<code>solve(A)</code>	Matrix inverse of A
<code>MASS::ginv(A)</code>	Generalized inverse of A (MASS package)
<code>eigen(A)</code>	Eigenvalues and eigenvectors of A
<code>chol(A)</code>	Choleski factorization of A
<code>diag(n)</code>	Create a $n \times n$ identity matrix
<code>diag(A)</code>	Returns the diagonal elements of a matrix A
<code>diag(x)</code>	Create a diagonal matrix from a vector x
<code>lower.tri(A), upper.tri(A)</code>	Matrix of logicals indicating lower/upper triangular matrix
<code>apply()</code>	Apply a function to the margins of a matrix
<code>rbind(...)</code>	Combines arguments by rows
<code>cbind(...)</code>	Combines arguments by columns and
<code>dim(A)</code>	Dimensions of A
<code>nrow(A), ncol(A)</code>	Number of rows/columns of A
<code>colnames(A), rownames(A)</code>	Get or set the column/row names of A
<code>dimnames(A)</code>	Get or set the dimension names of A

Data Frame (Marco de datos)

- R denomina data frames a los conjuntos de datos (base de datos o datasets).
- Los data frames son estructuras de datos de dos dimensiones (rectangulares) que pueden contener datos de diferentes tipos, por lo tanto, son heterogéneas.
- Un dataframe tiene una estructura similar a una matriz, donde las columnas pueden ser de diversos tipos.
- El dataframe es la estructura de datos fundamental usada en R para realizar análisis estadístico.
- Se crea automáticamente cuando se leen datos desde un archivo.

Data Frame (Marco de datos)

Un data frame está compuesto por vectores.

```
> datafram<- data.frame(  
+   "primero" = 2:5,  
+   "segundo" = c("A", "B", "C", "D"),  
+   "tercero" = c(1.1, 1.2, 1.3, 1.7),  
+   "cuarto" = as.character(c("a", "b", "c", "d")))  
> datafram  
  primero segundo tercero cuarto  
1       2        A     1.1      a  
2       3        B     1.2      b  
3       4        C     1.3      c  
4       5        D     1.7      d
```

Funciones de Prueba y Conversión

Las funciones por defecto más utilizadas en R son:

Type	Testing	Coercing
Array	is.array()	as.array()
Character	is.character()	as.character()
Dataframe	is.data.frame()	as.data.frame()
Factor	is.factor()	as.factor()
List	is.list()	as.list()
Logical	is.logical()	as.logical()
Matrix	is.matrix()	as.matrix()
Numeric	is.numeric()	as.numeric()
Vector	is.vector()	as.vector()

... Ahora practicamos con base de datos