R
SOFTWARE

REALIZADA POR
LIZETH RONCANCIO

UNICISO
WWW.PORTALUNICISO.COM

UNICISO
WWW.PORTALUNICISO.COM

¿Qué es R?

R es un sistema para análisis estadísticos y gráficos creado por Ross Ihaka y Robert Gentleman. Este software tiene un lenguaje de programación y se distribuye gratuitamente bajo los términos establecidos; su desarrollo y distribución son llevados a cabo por varios estadísticos como son el Grupo Nuclear de Desarrollo de R.

Cabe destacar que posee muchas funciones, las cuales se pueden visualizar de manera inmediata en su propia ventana, y ser guardados en varios formatos **(jpg, png, bmp, ps, pdf, emf, pictex, xfig).** Los resultados de análisis estadísticos se muestran en la pantalla, y algunos otros resultados se pueden guardar, exportar a un archivo, o ser utilizados en análisis posteriores.

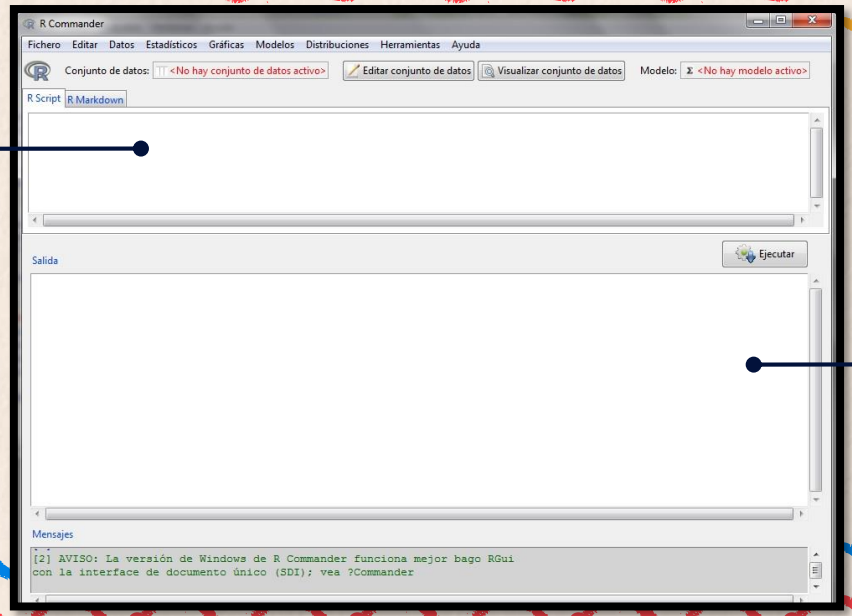
(Ahumada, 2003)

Una de las características más sobresalientes de R es su flexibilidad.

UNICISO
WWW.PORTALUNICISO.COM

Inicio

Ventana de instrucciones



Ventana de resultados

Una vez instale R en su computador, el programa se puede iniciar en el archivo correspondiente. **El cursor, que por defecto es el símbolo '>', indica que R está listo para recibir un comando.**

(Ahumada, 2003)

¿Cómo funciona?

R es un lenguaje orientado a objetos:

Significa que las variables, datos, funciones, resultados, etc., se guardan en **la memoria activa del computador en forma de objetos, con un nombre específico y sin usar archivos temporales.**

(Ahumada, 2003)

Debido a que los resultados mismos son objetos, pueden ser considerados como datos y analizados como tal.

La sintaxis de R es muy simple e intuitiva.

El usuario ejecuta las funciones con la ayuda de comandos definidos:

Para que una función sea ejecutada en R, debe estar siempre **acompañada de paréntesis**, inclusive en el caso que no haya nada dentro de los mismos. **Por ejemplo: ls().**

(Ahumada, 2003)

Los comandos escritos en el teclado son ejecutados directamente.

Por ejemplo, **una regresión lineal se puede ejecutar con el comando: lm(y ~x).**

UNICISO
WWW.PORTALUNICISO.COM

¿Dónde encuentro las funciones?

Las funciones disponibles están guardadas en una librería localizada en el directorio **R HOME/library**, que se encuentra donde R está instalado. También podremos observar el paquete denominado **base**, que constituye el núcleo de R, además de contener las funciones básicas del lenguaje para leer y manipular datos, **algunas funciones gráficas, y algunas funciones estadísticas.**

Comando

El comando es básicamente escribir el nombre de un objeto, para visualizar su contenido. Por ejemplo, si escribimos el objeto **n**, se visualiza que contiene el valor **10**:

```
> n
[1] 10
```

El dígito 1 indica que la visualización del objeto, comienza con el primer elemento de n.

El nombre de un objeto debe comenzar con una letra (A-Z), o también puede incluir dígitos (0-9), y puntos (.). R discrimina entre letras mayúsculas y minúsculas.

(Ahumada, 2003)

Creación, listado y remoción de objetos en memoria

Un objeto puede ser creado con el operador **“asignar”**, el cual se denota como una flecha. Se utiliza el signo menos y el símbolo **“>” o “<”**, dependiendo de la dirección en que se asigna el objeto. Ejemplo:

```
> x <- 1
> x
[1] 1
```

(Ahumada, 2003)

Si el objeto ya existe, su valor anterior es borrado después de la asignación. **El valor asignado de esta manera puede ser el resultado de una operación y/o de una función**. Ejemplo:

```
> n <- 10 + 2
> n
[1] 12
```

UNICISO
WWW.PORTALUNICISO.COM

Ejemplo

Son las instrucciones que digitamos.

La función que queremos visualizar.

```
> name <- "Carmen"; n1 <- 10; n2 <- 100; m <- 0.5
> ls()
[1] "m"  "n1"  "n2"  "name"
```

Es el resultado que nos arroja R.

La función **ls** simplemente lista los objetos en memoria: solo se muestran los nombres de los mismos.

El uso del punto y coma sirve para separar comandos diferentes en la misma línea.

(Ahumada, 2003)

La ayuda en línea

Proporciona información muy útil de cómo utilizar las funciones. Funciona al escribir alguno de estos dos comandos:

> ?
> help

Posteriormente se abre una ventana o página con información general sobre la función de la primera línea. Dicha información incluye:

- Descripción
- Uso
- Argumentos
- Detalles
- Valor
- Ver también
- Ejemplos

(Ahumada, 2003)

UNICISO
WWW.PORTALUNICISO.COM

Datos con R

Objetos

Todo objeto tiene dos atributos intrínsecos: **tipo y longitud.** El tipo se refiere a la clase básica de los elementos en el objeto; existen cuatro tipos principales: **numérico, carácter, complejo, y lógico (FALSE or TRUE).** La **longitud es simplemente el número de elementos en el objeto.**

(Ahumada, 2003)

Para ver el tipo y la longitud de un objeto, se **pueden usar las funciones mode y length.** Por ejemplo:

```
> x <- 1

> mode(x)
[1] "numeric"

> length(x)
[1] 1
```

Cuando un dato no está disponible, se representa **como NA (not available).** Además, los datos numéricos que son muy grandes se expresan en notación exponencial.

Leyendo datos desde un archivo

R utiliza el directorio de trabajo para **leer y escribir archivos**. Para saber cual es este directorio puede utilizar el comando: getwd().

R puede leer datos guardados como archivos de texto con las siguientes funciones: **read.table, scan y read.fwf.**

R también puede leer archivos ´en otros formatos **(Excel, SAS, SPSS,** . . .)

Guardando datos

La función: write.table, guarda el contenido de un objeto en un archivo. Para guardar un grupo de objetos de cualquier tipo, se puede usar el **comando: save.**

(Ahumada, 2003)

Generación de datos

Secuencias regulares:

Son una **secuencia regular de números enteros**, por ejemplo, de 1 hasta 30 se pueden generar con:

```
> x <- 1:30
```

La función: **seq**, puede generar secuencias de números reales. Por ejemplo:

```
> seq(1, 5, 0.5) [1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

Donde el primer número indica el principio de la secuencia, el segundo el final, y el tercero el ´ incremento que se debe usar para generar la secuencia.

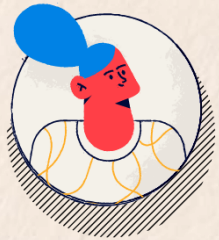
Secuencias aleatorias:

Estas funciones se generan con:

rfunc(n, p1, p2, ...)

Donde **func** indica la distribución, n es el número de datos generados, y p1, p2, son valores que toman los parámetros de la distribución.

(Ahumada, 2003)



Manipulación de objetos

Creación de objetos:

Vector

Tiene dos argumentos: **mode y length**, cuyos elementos pueden ser de tipo numérico, lógico o de carácter, dependiendo del argumento especificado en mode. Las siguientes funciones tienen exactamente el mismo efecto, y tienen un solo argumento **(la longitud del vector): numeric(), logical(), y character().**

Matriz

Una matriz es realmente un vector **con un atributo adicional (dim),** que define el número de filas y columnas del mismo. Una matriz se puede crear con la función matrix.

Marco de datos

Se crea de manera implícita con la función: **read.table,** también es posible hacerlo con la función: data.frame.

Lista

Se utiliza la función: **list**. No existe ninguna limitación en el tipo de objetos que se pueden incluir.

(Ahumada, 2003)

Series de tiempo

La función: ts, crea un objeto de clase "ts" (serie de tiempo), a partir de un vector (serie de tiempo única) o una matriz (serie multivariada). Por ejemplo:

> ts(1:10, start = 1959)

Time Series:

Start = 1959

End = 1968

Frequency = 1 [1] 1 2 3 4 5 6 7 8 9 10

Expresión

Todos los comandos válidos son expresiones. Cuando se escribe un comando directamente en el teclado, este es evaluado por **R y ejecutado si es válido.** En muchos casos, es útil construir una expresión sin evaluarla, esto es lo que hace la función: expression. Por supuesto, es posible evaluar la expresión posteriormente con eval().

Factor

Incluye no solo los valores correspondientes a una variable categórica, **también los diferentes niveles posibles de esta variable.** Por ejemplo:

> factor(1:3, levels=1:5)

[1] 1 2 3

Levels: 1 2 3 4 5

(Ahumada, 2003)

Conversión de objetos

Tal conversión se puede realizar usando la función: **as.algo.**

Conversión a	Función	Reglas
numérico	`as.numeric`	`FALSE` → 0
		`TRUE` → 1
		`"1", "2", ...` → 1, 2, ...
		"A", ... → `NA`
lógico	`as.logical`	0 → `FALSE`
		otros números → `TRUE`
		`"FALSE", "F"` → `FALSE`
		`"TRUE", "T"` → `TRUE`
		otros caracteres → `NA`
caracter	`as.character`	1, 2, ... → `"1", "2", ...`
		`FALSE` → `"FALSE"`
		`TRUE` → `"TRUE"`

Operadores

Los operadores aritméticos actúan sobre variables de tipo numérico o complejo, pero también lógico. Los operadores comparativos pueden actuar sobre cualquier tipo devolviendo uno o varios valores lógicos. Los operadores lógicos pueden actuar sobre uno o dos objetos de tipo lógico, y pueden devolver uno (o varios) valores lógicos.

Operadores

Aritméticos		Comparativos		Lógicos	
+	adición	<	menor que	`! x`	NO lógico
−	substracción	>	mayor que	`x & y`	Y lógico
*	multiplicación	<=	menor o igual que	`x && y`	id.
/	división	>=	mayor o igual que	`x | y`	O lógico
^	potencia	==	igual	`x || y`	id.
%%	módulo	!=	diferente de	`xor(x, y)`	O exclusivo
%/%	división de enteros				

(Ahumada, 2003)

Cómo acceder a los valores de un objeto

El sistema de indexación, es una manera eficiente y flexible de acceder selectivamente a los elementos de un objeto, y puede ser numérico o lógico. Por ejemplo, para acceder al tercer elemento de un vector x, simplemente se escribe: x[3].

El editor de datos

Es posible utilizar un editor gráfico similar a una hoja de cálculo para editar un objeto numérico. Por ejemplo, si X es una matriz, el comando: **data.entry(X)**, abrira un editor gráfico que le permitirá cambiar los valores en la matriz o adicionar nuevas columnas y/o filas.

Accediendo a los valores de un objeto con nombres

Los nombres son atributos, y existen diferentes tipos: **names, colnames, rownames, dimnames.**

(Ahumada, 2003)

Funciones aritméticas simples

Existen muchas funciones en R para manipular datos.
Algunos ejemplos son:

`sum(x)`	suma de los elementos de `x`
`prod(x)`	producto de los elementos de `x`
`max(x)`	valor máximo en el objeto `x`
`min(x)`	valor mínimo en el objeto `x`
`which.max(x)`	devuelve el índice del elemento máximo de `x`
`which.min(x)`	devuelve el índice del elemento mínimo de `x`
`range(x)`	rango de `x` o `c(min(x), max(x))`
`length(x)`	número de elementos en `x`
`mean(x)`	promedio de los elementos de `x`
`median(x)`	mediana de los elementos de `x`
`var(x)` o `cov(x)`	varianza de los elementos de `x` (calculada en $n-1$); si `x` es una matriz o un marco de datos, se calcula la matriz de varianza-covarianza
`cor(x)`	matriz de correlación de `x` si es una matriz o un marco de datos (1 si `x` es un vector)
`var(x, y)` o `cov(x, y)`	covarianza entre `x` y `y`, o entre las columnas de `x` y `y` si son matrices o marcos de datos
`cor(x, y)`	correlación lineal entre `x` y `y`, o la matriz de correlación si `x` y `y` son matrices o marcos de datos

(Ahumada, 2003)

Cálculos con Matrices

R posee facilidades para manipular y hacer operaciones con matrices. Las funciones: rbind() y cbind(), unen matrices con respecto a sus filas o columnas respectivamente.

UNICISO
WWW.PORTALUNICISO.COM

Gráficas en R

Cada función gráfica en R tiene un enorme número de opciones permitiendo una gran flexibilidad en la producción de un gráfico. Utilice el comando: **demo(graphics).**

Existen dos tipos de funciones gráficas: **las funciones de graficación de alto nivel,** que crean una nueva gráfica; y las **funciones de graficación de bajo nivel**, que agregan elementos a una gráfica ya existente.

Manejo de gráficos

Abriendo múltiples dispositivos gráficos: R abre una ventana para mostrar el gráfico si no hay ningún dispositivo abierto.

Disposición de una gráfica: La función: **split.screen**, divide el dispositivo gráfico activo.

(Ahumada, 2003)

Funciones gráficas

`plot(x)`	graficar los valores de `x` (en el eje *y*) ordenados en el eje *x*
`plot(x, y)`	gráfico bivariado de `x` (en el eje *x*) y `y` (en el eje *y*)
`sunflowerplot(x, y)`	igual a `plot()` pero los puntos con coordenadas similares se dibujan como flores con el número de pétalos igual al número de puntos
`piechart(x)`	gráfico circular tipo 'pie'
`boxplot(x)`	gráfico tipo 'box-and-whiskers'
`stripplot(x)`	gráfico de los valores de `x` en una línea (como alternativa a `boxplot()` para pequeños tamaños de muestra)
`coplot(x~y | z)`	gráfico bivariado de `x` y `y` para cada valor o intervalo de valores de `z`
`interaction.plot (f1, f2, y)`	si `f1` y `f2` son factores, grafica el promedio de `y` (en el eje *y*) con respecto a los valores de `f1` (en el eje *x*) y de `f2` (curvas diferentes); la opción `fun` permite escoger un estadístico de `y` (por defecto, el promedio: `fun=mean`)
`matplot(x,y)`	gráfica bivariada de la primera columna de `x` *vs.* la primera columna de `y`, la segunda columna de `x` *vs.* la segunda columna de `y`, etc.
`dotplot(x)`	si `x` es un marco de datos, hace un gráfico de puntos tipo Cleveland (gráficos apilados fila por fila y columna por columna)

(Ahumada, 2003)

UNICISO
WWW.PORTALUNICISO.COM

Comandos de graficación de bajo nivel

R posee un conjunto de funciones gráficas que afectan una gráfica ya existente, denominados: comandos de **graficación de bajo nivel**. Estos son los principales:

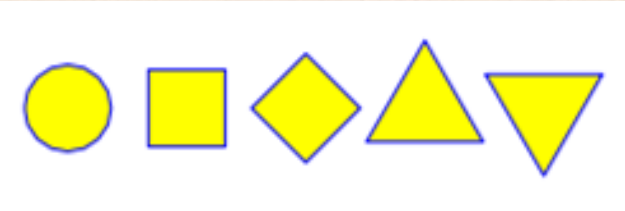
`points(x, y)`	agrega puntos (se puede usar la opción `type=`)
`lines(x, y)`	igual a la anterior pero con líneas
`text(x, y, labels, ...)`	agrega texto dado por `labels` en las coordenadas (x,y); un uso típico: `plot(x, y, type="n"); text(x, y, names)`
`mtext(text, side=3, line=0, ...)`	agrega texto dado por `text` en el margen especificado por `side` (ver `axis()` más abajo); `line` especifica la línea del área de graficado
`segments(x0, y0, x1, y1)`	dibuja una línea desde el punto (x0,y0) hasta el punto (x1,y1)
`arrows(x0, y0, x1, y1, angle= 30, code=2)`	igual al anterior pero con flechas desde (x0,y0) si `code=2`, al punto (x1,y1) si `code=1`, o en ambos si `code=3`; `angle` controla el ángulo desde la base de la flecha hasta la punta de la misma
`abline(a,b)`	dibuja una línea con pendiente `b` e intercepto `a`
`abline(h=y)`	dibuja una línea horizontal en la ordenada `y`
`abline(v=x)`	dibuja una línea vertical en la abcisa `x`
`abline(lm.obj)`	dibuja la línea de regresión dada por `lm.obj` (ver sección 5)
`rect(x1, y1, x2, y2)`	dibuja un rectángulo donde las esquinas izquierda, derecha, superior e inferior están dadas por `x1`, `x2`, `y1`, y `y2`, respectivamente
`polygon(x, y)`	dibuja un polígono uniendo los puntos dados por `x` y `y`
`legend(x, y, legend)`	agrega la leyenda en el punto (x,y) con símbolos dados por `legend`
`title()`	agrega un título y opcionalmente un sub-título
`axis(side, vect)`	agrega un eje en la parte inferior (`side=1`), izquierda (2), superior (3), o derecha (4); `vect` (opcional) da la abscisa (u ordenada) donde se deben dibujar los marcadores ('tick marks') del eje
`rug(x)`	dibuja los datos `x` en el eje *x* como pequeñas líneas verticales
`locator(n, type="n", ...)`	devuelve las coordenadas (*x*,*y*) después que el usuario a hecho click `n` veces en el gráfico con el ratón; también dibuja símbolos (`type="p"`) o líneas (`type="l"`) con respecto a parámetros gráficos opcionales (`...`); por defecto no se dibuja nada (`type="n"`)
`identify(x, ...)`	similar a `locator()` con la diferencia que imprime en la gráfica el valor de x (u opcionalmente de una leyenda especificada en la opción `labels=`) más cercano al punto donde se hizo click. Util para identificar puntos en la gráfica que están asociados con nombres.

(Ahumada, 2003)

Parametros gráficos

Estos se pueden utilizar como opciones de las funciones gráficas (pero no funciona para todas), usando la función: **par**, para cambiar de manera permanente parámetros gráficos, es decir gráficas subsecuentes se dibujarán con respecto a los parámetros especificados por el usuario. Por ejemplo, el siguiente comando:

> par(bg="yellow"), dará como resultado que todos los gráficos subsecuentes tendrán el fondo de color amarillo.

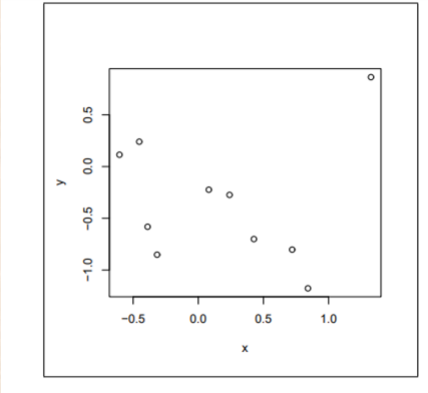


(Ahumada, 2003)

Los paquetes grid y lattice

Son **gráficas condicionales múltiples**: una gráfica bivariada se divide en varias gráficas con respecto a los valores de una tercera variable. **Grid**, contiene todas las funciones necesarias para el modo gráfico, mientras que lattice, contiene las funciones más comúnmente usadas.

La gráfica que queremos visualizar se puede obtener con: plot()



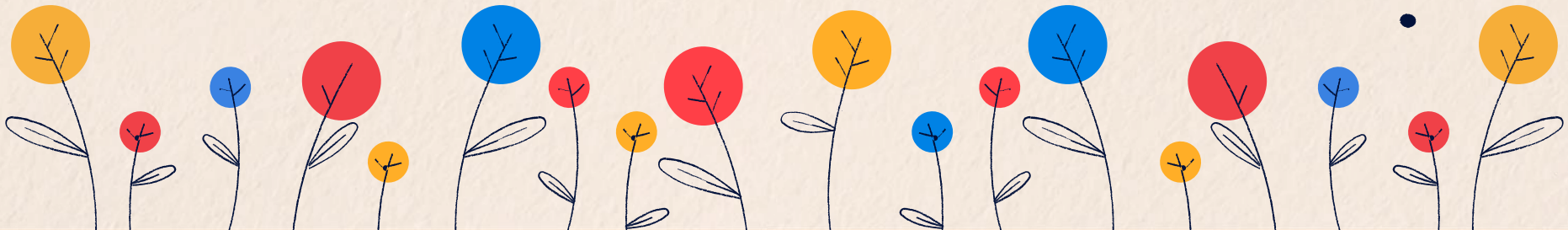
(Ahumada, 2003)

UNICISO
WWW.PORTALUNICISO.COM

Análisis estadísticos con R

Los comandos y conceptos analizados son una parte de las caracterısticas de R **para realizar análisis de todo tipo, debido a que este contiene una gran diversidad de funciones.**

(Ahumada, 2003)



Fórmulas

El uso de fórmulas es un elemento clave en el análisis estadístico con R: la nomenclatura utilizada es la misma para casi todas las funciones. Una fórmula se escribe típicamente como: **y ~modelo, donde (y) es la variable dependiente o de respuesta, y (modelo) es un conjunto de términos para los cuales es necesario estimar una serie de parámetros.**

`a+b`	efectos de `a` y `b`
`X`	si `X` es una matriz, especifica un efecto aditivo para cada una de las columnas; por ejemplo `X[,1]+X[,2]+...+X[,ncol(X)]`; algunas de las columnas se pueden seleccionar con índices numéricos (por ej., `X[,2:4]`)
`a:b`	efecto interactivo entre `a` y `b`
`a*b`	efectos aditivos e interactivos entre `a` y `b` (idéntico a `a+b+a:b`)
`poly(a, n)`	polinomios de `a` hasta grado *n*
`^n`	incluye todas las itnteracciones hasta el nivel *n*, por ej., `(a+b+c)^2` es idéntico a `a+b+c+a:b+a:c+b:c`
`b%in% a`	los efectos de `b` están anidados en `a` (idéntico a `a+a:b` o `a/b`)
`a-b`	remueve el efecto de `b`, por ejemplo: `(a+b+c)^2-a:b` es idéntico a `a+b+c+a:c+b:c`
`-1`	`y~x-1` regresión a través del origen (igual para `y~x+0` o `0+y~x`)
`1`	`y~1` ajusta un modelo sin efectos (solo el intercepto)
`offset(...)`	agrega un efecto al modelo sin estimar los parámetros (e.g., `offset(3*x)`)

(Ahumada, 2003)

Funciones genéricas

Los objetos resultantes de un análisis poseen un atributo particular denominado clase, que contiene la "firma" de la función usada para el análisis. **Las funciones que se usan posteriormente para extraer los resultados del análisis actuarán específicamente con respecto a la clase del objeto.** Por ejemplo, la función más utilizada para extraer resultados de otros análisis es: summary, que muestra los resultados detallados de los mismos.

`print`	devuelve un corto resúmen
`summary`	devuelve un resúmen detallado
`df.residual`	devuelve el número de grados de libertad
`coef`	devuelve los coeficientes estimados (algunas veces con sus errores estándar)
`residuals`	devuelve los residuales
`deviance`	devuelve la devianza
`fitted`	devuelve los valores ajustados
`logLik`	calcula el logaritmo de la verosimilitud y el número de parámetros
`AIC`	calcula el criterio de información de Akaike o AIC (depende de `logLik()`)

(Ahumada, 2003)

PAQUETES

El primer paso es cargar: **> library(eda).** Estos son algunos de los paquetes disponibles:

Paquete	Descripción
ctest	pruebas clásicas (Fisher, 'Student', Wilcoxon, Pearson, Bartlett, Kolmogorov-Smirnov, ...)
eda	métodos descritos en "Exploratory Data Analysis" por Tukey (solo ajuste lineal robusto y ajuste de medianas)
lqs	regresión resistente y estimación de covarianza
methods	definición de métodos y clases para objetos en R y herramientas de programación
modreg	regresión moderna (alisamiento y regresión local)
mva	análisis multivariado
nls	regresión no-lineal
splines	representaciones polinómicas
stepfun	funciones de distribución empíricas
tcltk	funciones para hacer interfase desde R a elementos de interfase gráfica Tcl/Tk
tools	herramientas para desarrollo y administración de paquetes
ts	análisis de series temporales

(Ahumada, 2003)

UNICISO
WWW.PORTALUNICISO.COM

Programación práctica con R

Escribiendo un programa en R

Los programas en R se escriben en un archivo que **se guarda en formato ASCII, con terminación R**. Un programa se usa típicamente cuando uno quiere hacer una o varias operaciones muchas veces.

Bucles y Vectorización

Posibilidad de programar de una manera muy sencilla, una serie de **análisis que se puedan ejecutar de manera sucesiva.**

Creando sus propias funciones

R permite **al usuario escribir sus propias funciones,** y estas tendrán las mismas propiedades que las demás funciones.

(Ahumada, 2003)

Literatura adicional sobre R

- Manuales: R trae varios manuales que se instalan por defecto en R HOME/doc/manual/.
- FAQ: R también viene con su propio FAQ (Preguntas más frecuentes), localizadas en el directorio R HOME/doc/html.
- Recursos en línea: el sitio CRAN y la página web de R contienen varios documentos, recursos bibliográficos y enlaces a otros sitios.
- Listas de correo: existen tres listas de discusión en R; para suscribirse, mande un mensaje o lea los archivos en http://www.R-project.org/mail.html.
- R News: la revista electrónica R News tiene como objetivo llenar un vacío entre las listas de discusión electrónicas y publicaciones científicas tradicionales.

(Ahumada, 2003)

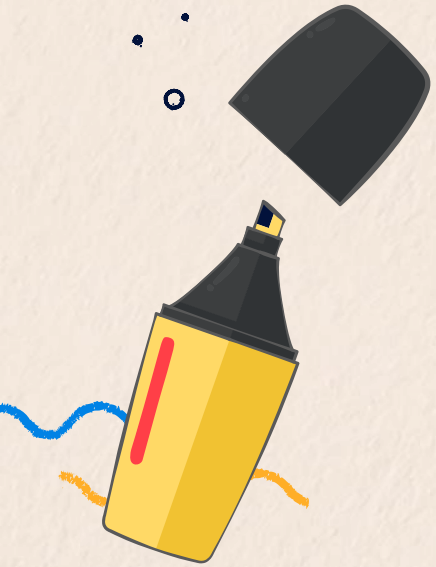
Referencias

- Ahumada, J. A. (2003). R para Principiantes. University of Hawaii.
- Ihaka R. & Gentleman R. 1996. R: a language for data analysis and graphics. Journal of Computational and Graphical Statistics 5: 299–314.

UNICISO
WWW.PORTALUNICISO.COM

CITA DE LA GUÍA

Roncancio, L. (2020). R. Software. UNICISO. Disponible en: www.portaluniciso.com



UNICISO
WWW.PORTALUNICISO.COM