

ANÁLISIS DE DATOS

BIBLIOGRAFÍA:

UNDERSTANDING ROBUST AND EXPLORATORY DATA ANALYSIS. Hoaglin, Mosteller, Tuckey. Wiley.

MODERN APPLIED STATISTICS WITH S-PLUS. Venables, Ripley.

SOFTWARE: S-PLUS, R

Página:

http://www.dm.uba.ar/materias/analisis_de_datos/2006/1/

¿POR QUÉ ANÁLISIS DE DATOS?

técnicas estadísticas clásicas

- **óptimas**- condiciones restrictivas
- **inadecuadas**- situación real alejamiento de los supuestos

técnicas robustas y exploratorias más recientes han ampliado la efectividad de los análisis estadísticos.

técnicas del análisis exploratorio de datos

- permiten dar un tratamiento informal a un conjunto de datos
- dan énfasis al estudio flexible de los datos antes de compararlos con cualquier modelo probabilístico.

S-PLUS - R

Ambos proveen

- entorno flexible para el análisis de datos.
- una colección extensa y coherente de herramientas estadísticas para análisis de datos,
- un lenguaje para expresar modelos estadísticos y herramientas para utilizar modelos estadísticos lineales y no lineales.
- facilidades para el análisis de datos y su presentación tanto en la computadora como en papel,
- un lenguaje de programación orientado a objetos que puede ser fácilmente extendido.

La mayoría de los objetos que se crean en **S-PLUS** son

permanentes,

en particular los datos, los resultados y las funciones.

En **R** el programa pregunta si se quiere guardar el espacio de trabajo - workspace - cada vez que se cierra la sesión.

Al **guardar** el espacio de trabajo, los objetos creados durante la sesión, quedan en forma **permanente** hasta que se los borre.

Un poco de historia

R es una implementación libre, independiente, “open-source” del lenguaje de programación **S** que actualmente es un producto comercial llamado **S-PLUS** y es distribuido por Insightful Corporation.

El lenguaje **S**, que fue escrito a mediados de los años 70 en Bell Labs (de AT&T y actualmente Lucent Technologies).

Originalmente un programa para el sistema operativo Unix, **R** ahora puede obtenerse también en versiones para Windows y Macintosh y Linux.

A pesar de que hay diferencias menores entre **R** y **S-PLUS** (la mayoría en la interfase gráfica), son esencialmente idénticos.

El proyecto **R** fue iniciado por Robert Gentleman y Ross Ihaka (de donde se deriva “**R**”) del Statistics Department in the University of Auckland en 1995.

Actualmente **R** es mantenido por un grupo internacional de desarrolladores *voluntarios*: Core development team.

La página web del proyecto **R** es <http://www.r-project.org>. Este es el sitio principal sobre in formación de **R**: documentación, FAQs (FAQ son las iniciales de Frequently Asked Questions, o sea preguntas más frecuentes).

Para bajar el software directamente se puede visitar el **Comprehensive R Archive Network (CRAN)**

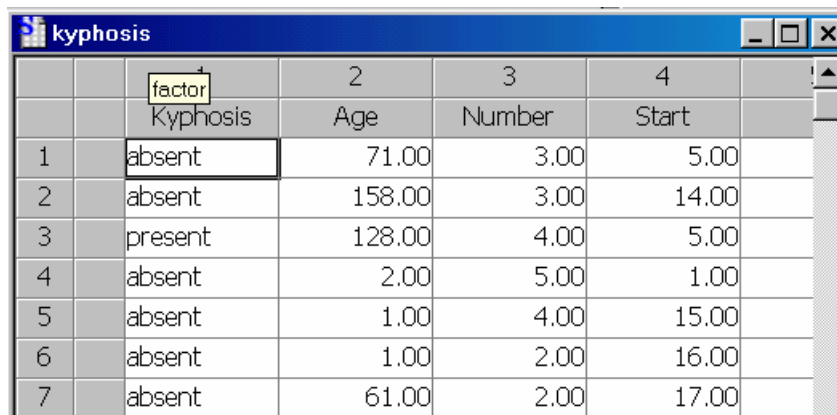
<http://cran.us.r-project.org/>

Tipos de datos

5 Tipos de objetos datos básicos:

data frames, matrices, vectores, listas y funciones.

Data frame: permite almacenar datos bidimensionales.



The screenshot shows an R console window with the title 'kyphosis'. It displays a data frame with 7 rows and 5 columns. The columns are labeled 'factor', 'Age', 'Number', and 'Start'. The 'factor' column contains the values 'absent' and 'present'. The other columns contain numerical values.

	factor	Age	Number	Start
1	absent	71.00	3.00	5.00
2	absent	158.00	3.00	14.00
3	present	128.00	4.00	5.00
4	absent	2.00	5.00	1.00
5	absent	1.00	4.00	15.00
6	absent	1.00	2.00	16.00
7	absent	61.00	2.00	17.00

kyphosis:

data frame, 81 filas (casos, niños sometidos a una cirugía espinal)
Variables: Kyphosis - factor - con 2 niveles presencia o ausencia de una deformidad post operatoria, Age, Number, Start son vectores numéricos.

Todas las columnas deben tener la misma longitud

Matrices: son similares a las data frames, salvo que sus elementos deben tener datos con el mismo modo (carácter, numérico, lógico). Las filas y las columnas pueden tener nombres.

Vectores: es un conjunto ordenado de elementos que tienen el mismo modo. Los elementos de un vector pueden tener nombres.

Listas: son colecciones de otros objetos. Sus componentes pueden ser data frames, matrices, vectores, otras listas, cualquier objeto de S-plus.

Funciones: existen gran cantidad de funciones incorporadas al S-plus. También es posible agregar funciones definidas por el usuario.

Más sobre **datos incorporados**

En S-plus

Object Explorer -> + search path -> Data
se encuentra “kyphosis”

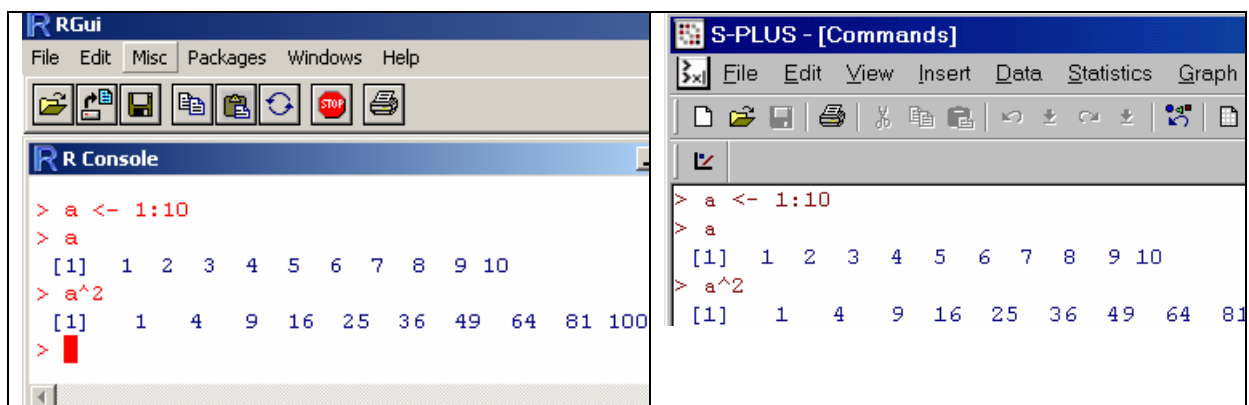
Data -> Select Data... en el menú principal

En la consola de comandos de R

```
> data()  
>
```

aparece una ventana con los nombres de los conjuntos de datos.

Ventana de Comandos



Lenguajes, R y S-plus: conceptos básicos

expresiones, asignaciones, funciones, tipos de datos

expresión simple

```
> 2+3    ↵    (enter) #expresión
[1] 5      #evaluación
```

expresión un poco más compleja

```
> sqrt(3/4)/(1/3 - 2/pi^2)
[1] 6.626513
```

El símbolo `>` (prompt) indica la línea de comandos y el `[1]` que la respuesta comienza en el primer elemento de un vector.

Si se escribe una expresión incompleta `> 2* ↵`
S-plus responde con un `+` que indica continuar

```
> 2* ↵
+ 5
[1] 10

> sqrt(3/4 ↵
+ )
[1] 0.8660254
```

La expresión más común es el llamado a una función, se escribe el nombre de la función seguida de sus argumentos entre paréntesis.

```
> sqrt(3/4)
[1] 0.8660254
```

```
> Sqrt(3/4)
Problem: Couldn't find a function
definition for "Sqrt"
```

Diferencian mayúsculas de minúsculas.

Evaluamos

```
> pi  
[1] 3.141593
```

Si se escribe una cadena de caracteres seguidos por un par de paréntesis el **S-PLUS** lo interpreta como el **nombre** de una **función**.

```
> pi()  
Error: couldn't find function "pi"  
In addition: Warning messages:  
Looking for object "pi" of mode  
"function", ignored one of mode  
"numeric"
```

y en R?

```
> pi()  
Error: attempt to apply non-function
```

Si la función existe y no requiere argumentos se ejecutará la función, como

```
q()
```

para irse de la sesión.

Si la función existe y requiere argumentos dará un mensaje de error

```
> sqrt()      # en S-plus  
Problem in sqrt(): argument "x" is missing  
with no default
```

Use `traceback()` to see the call stack

`> sqrt()` `# en R`

Error: 0 arguments passed to 'sqrt' which requires 1

Asignaciones

Hay varios operadores con los que es posible realizar asignaciones

`"< -"` signo menor seguido del signo menos, sin espacios

`"="`,

`"_"`, subguión `#` es confuso y por suerte R no lo tiene!

En S-plus	En R
<pre>> a <- 2 > a [1] 2 > b_-2 > b [1] -2 > c <- pi > c [1] 3.141593</pre>	<pre>> a <- 2 > a [1] 2 > b_-2 Error: object "b_" not found > b Error: object "b" not found > c <- pi > c [1] 3.141593</pre>

¡SON MUY PARECIDOS!

Enteros consecutivos

`> a <- 2:6` `# crea el vector (2,3,4,5,6)`

```
> a
[1] 2 3 4 5 6
```

Aritmética

```
> b <- 2*a+1
> b
[1] 5 7 9 11 13
```

```
> b <- a/2          # división
> b
[1] 1.0 1.5 2.0 2.5 3.0
```

```
> b <- a^3.7         # elevar a la potencia 3.1
> b
[1] 12.99604 58.25707 168.89701
[4] 385.64616 757.11112
```

```
> b <- log(a)        # logaritmo natural
> b
[1] 0.6931472 1.0986123 1.3862944
[4] 1.6094379 1.7917595
```

```
> b <- log10(a)      # asignación
> log10(a)           # evaluación
[1] 0.3010300 0.4771213 0.6020600
[4] 0.6989700 0.7781513
```

```
> b <- logb(a, base=2) # logaritmo base 2
```

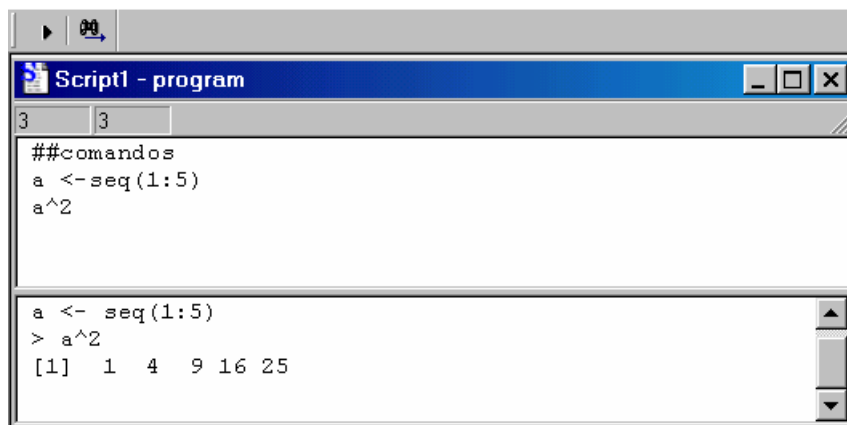
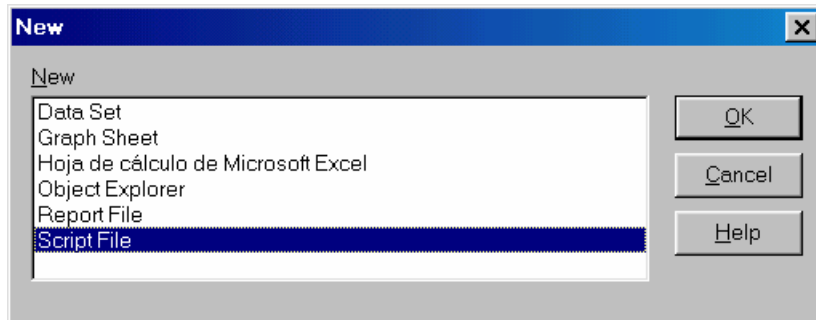
```
> b <- logb(a, 2)     # idem
```

```
> help(logb)          # se abre una ventana de
                        #ayuda
```

Ventana de Escritura (Script)

En S-plus.

File -> New (ó )



En R

File -> New script



Es útil acomodar las dos ventanas, la de comandos y la del editor, para poder verlas simultáneamente **Windows -> Tile**